



# ***GE Fanuc Automation***

---

***Программируемые Устройства Управления***

***Модуль Motion Mate™ DSM314  
для ПЛК Series 90™-30***

***Руководство пользователя***

GFK-1742B\_RU

Март 2003

## *Предупреждения и замечания встречающиеся в руководстве*

### **Предупреждение**

**Предупреждения используются в этом документе, чтобы уведомить об опасном напряжении, токе, температуре, или других факторах, которые могут нанести вред здоровью при работе с оборудованием.**

**Предупредительные надписи применяют, когда невнимательность может привести к повреждению здоровья и оборудования.**

### **Внимание**

**Внимание обращается на действия, которые могут привести к повреждению оборудования.**

**Примечание:** Примечания привлекают внимание к информации, которая важна для понимания принципов функционирования оборудования.

Документ содержит информацию, доступную на момент публикации. Информация может быть уточнена. Документ не описывает все возможные варианты аппаратного и программного обеспечения, а также способы их установки, функционирования и обслуживания. Описываемые функции могут присутствовать не во всех видах оборудования. Компания GE Fanuc Automation оставляет за собой право вносить изменения в документ без дополнительного уведомления.

Компания GE Fanuc Automation не дает явных либо подразумеваемых гарантий точности, полноты, достаточности, или полезности информации, содержащейся в документе. Не предоставляются также гарантии коммерческой ценности или соответствия целям пользователя.

Ниже перечислены товарные знаки, принадлежащие GE Fanuc Automation North America, Inc.

Alarm Master	Genius	PowerMotion	VersaMax
CIMPLICITY	Helpmate	PowerTRAC	VersaPro
CIMPLICITY 90-ADS	Logicmaster	Series 90	VuMaster
CIMSTAR	Modelmaster	Series Five	Workmaster
FrameworkX	Motion Mate	Series One	
Field Control	ProLoop	Series Six	
GENet	PROMACRO	Series Three	

©Copyright 1999 - 2002 GE Fanuc Automation North America, Inc.

**Всё права защищены**

## Содержание данного руководства

В данном руководстве описывается модуль Motion Mate DSM314, полноценная система управления движением в форме интеллектуального программируемого модуля для ПЛК Series 90-30.

## Документы, связанные с этим руководством

За дополнительной информацией обращайтесь к следующим руководствам:

*Руководство по установке и аппаратному обеспечению ПЛК Series 90™-30*, GFK-0356\_RU

*Installation Requirements for Conformance to Standards*, GFK-1179

*α Series Servo Amplifier (SVU) Descriptions Manual*, GFZ-65192EN

*Control Motor Amplifier α Series (SVM)*, GFZ-65162E

*α Series Servo Motor Manuals*, GFZ-65165E, GFZ-65150E, GFZ-65142E

*β Series Servo Product Specification Guide*, GFH-001

*SL Series Servo User's Manual*; GFK-1581

*VersaPro™ Software User's Manual*, GFK-1670

*CIMPLICITY® Machine Edition Logic Developer-PLC Getting Started*, GFK-1918



<b>Глава 1. Краткое описание устройства .....</b>	<b>1-1</b>
Поддерживаемые типы сервоустройств .....	1-1
<b>Характеристики модуля Motion Mate DSM314.....</b>	<b>1-1</b>
Высокая производительность.....	1-1
Простота использования.....	1-2
Универсальность и гибкость входов /выходов.....	1-2
<b>Раздел 1: Краткое описание системы движения.....</b>	<b>1-3</b>
<b>ПЛК Series 90-30 и модуль DSM314.....</b>	<b>1-4</b>
Задержка данных ПЛК и задержки DSM314.....	1-4
Передача данных от ПЛК к DSM.....	1-4
Программа движения / CTL-входы лицевой панели.....	1-5
Local Logic.....	1-5
Интервалы обновления контура сервопривода в DSM314.....	1-5
DSM314 - стробы позиции .....	1-5
Увеличение длительности цикла ПЛК при работе с DSM314.....	1-6
Программное обеспечение.....	1-7
Интерфейсы оператора .....	1-7
Разъемы сервопривода и станка.....	1-7
<b>Раздел 2: Краткое описание работы модуля DSM314.....</b>	<b>1-8</b>
Стандартный режим ( <i>следающий контур управления</i> = выключен).....	1-8
Режим слежения ( <i>следающий контур управления</i> = включен).....	1-8
Работа в стандартном режиме .....	1-9
Работа в режиме слежения .....	1-10
<b>Раздел 3: Сервоприводы <math>\alpha</math> Series (цифровой режим) .....</b>	<b>1-11</b>
Интегральный цифровой усилитель $\alpha$ Series (SVU).....	1-11
Серводвигатели $\alpha$ Series .....	1-12
<b>Раздел 4: Сервоприводы <math>\beta</math> Series (цифровой режим).....</b>	<b>1-13</b>
Цифровые усилители $\beta$ Series.....	1-13
Серводвигатели $\beta$ Series.....	1-14
<b>Раздел 5: Сервоприводы SL-Series (аналоговый режим скорости) .....</b>	<b>1-15</b>
<b>Глава 2. Начало работы .....</b>	<b>2-1</b>
<b>Раздел 1: Распаковка системы .....</b>	<b>2-3</b>
Распаковка DSM314 .....	2-3
Распаковка Цифрового Сервоусилителя .....	2-3
Распаковка Двигателя FANUC.....	2-3
<b>Раздел 2: Сборка Системы Motion Mate DSM314.....</b>	<b>2-4</b>
Общие Указания .....	2-4
Соединения Motion Mate DSM314.....	2-4

Подключение Цифрового Сервоусилителя SVU $\alpha$ Series .....	2-5
Переключатели Каналов Усилителя SVU .....	2-6
Установка и Подключение DSM314 для Аналогового Режима .....	2-21
Заземление Системы Управления Движением Motion Mate DSM314.....	2-22
<b>Раздел 3: Включение Motion Mate DSM314 .....</b>	<b>2-24</b>
<b>Раздел 4: Конфигурация Motion Mate DSM314 (вводное руководство)....</b>	<b>2-25</b>
Подключение Программатора к ПЛК.....	2-25
Конфигурация VersaPro .....	2-26
Конфигурация при помощи программы SIMPLICITY Machine Edition .....	2-31
Данные по Конфигурации Модуля (Только как Вводное руководство по Началу Работы).....	2-34
Тип Двигателя (Цифровой Режим) .....	2-37
Сохранение Конфигурации на ПЛК .....	2-39
<b>Раздел 5: Проверка системы.....</b>	<b>2-40</b>
Формирование Движение .....	2-40
Режим Разрешения Работы на ПЛК .....	2-40
VersaPro.....	2-40
SIMPLICITY Machine Edition .....	2-41
Толчковое перемещение при помощи Motion Mate DSM314.....	2-42
<b>Раздел 6: Поиск и устранение неисправностей Системы Движения.....</b>	<b>2-43</b>
Сигналы тревоги.....	2-43
Установки Конфигурации.....	2-43
Техническая Поддержка .....	2-43
Веб-сайт GE Fanuc .....	2-43
Телефонные Номера GE Fanuc .....	2-43
<b>Раздел 7: Дальнейшие действия.....</b>	<b>2-44</b>
<b>Глава 3. Монтаж и подключение модуля DSM314.....</b>	<b>3-1</b>
<b>Раздел 1: Описание аппаратуры .....</b>	<b>3-1</b>
Светодиодные индикаторы.....	3-2
Разъем DSM COMM (последовательные связи).....	3-3
Разъемы В/В .....	3-3
Подсоединение заземления экрана .....	3-4
<b>Раздел 2: Установка модуля DSM314.....</b>	<b>3-5</b>
Стандартные источники питания Series 90-30 .....	3-6
Источники питания Series 90-30 повышенной мощности .....	3-6
<b>Раздел 3: Разводка и подключения Вводов/Выводов .....</b>	<b>3-8</b>
<b>Типы цепей Ввода/Вывода (В/В).....</b>	<b>3-8</b>
<b>Клеммные колодки.....</b>	<b>3-8</b>

<b>Клеммная колодка цифровой сервооси - IC693ACC335 .....</b>	<b>3-10</b>
Описание .....	3-10
Монтажные размеры .....	3-11
Переход от монтажа на DIN-шине к монтажу на панели .....	3-12
<b>Вспомогательная клеммная колодка - IC693ACC336 .....</b>	<b>3-14</b>
Описание и монтажные размеры .....	3-14
Переход от монтажа на DIN-шине к монтажу на панели .....	3-15
<b>Кабели .....</b>	<b>3-17</b>
Заземление кабеля В/В .....	3-20
Кабель от DSM до цифрового сервоусилителя $\alpha$ или $\beta$ Series – заземление сигнального кабеля .....	3-20
Идентификаторы цепей В/В и обозначения сигналов .....	3-23
Функции цепей В/В и назначения контактов .....	3-23
Назначение цепей и контактов для цифровых сервоосей 1, 2 .....	3-24
Назначение цепей и контактов для аналоговых сервоосей 1-4 .....	3-26
Назначение цепей и контактов вспомогательных осей 2-4 .....	3-28
Схемы подключения В/В .....	3-30
Технические характеристики В/В .....	3-38
Дифференциальные / однополярные 5В-входы .....	3-39
Однополярный 5В вход-приемник .....	3-40
Оптически изолированные входы-источники/ входы-приемники 24 В .....	3-41
Однополярные 5В-вводы / выводы .....	3-42
Дифференциальные 5В-выходы .....	3-42
Оптически изолированный выход, 24 В пост. тока .....	3-44
Оптически изолированный включающий релейный выход .....	3-45
Дифференциальные +/- 10 В аналоговые входы .....	3-46
Однополярный аналоговый выход +/- 10 В .....	3-47
Питание +5 В .....	3-48
 <b>Глава 4. Конфигурация .....</b>	 <b>4-1</b>
<b>Конфигурация крейта / слота .....</b>	<b>4-1</b>
<b>Конфигурация модуля .....</b>	<b>4-2</b>
Задание параметров конфигурации .....	4-2
Установочные параметры .....	4-3
Данные конфигурации порта последовательной связи .....	4-8
Управляющие биты (CTL) .....	4-9
Output Bits (Выходные биты) .....	4-10
Данные конфигурации оси .....	4-10
Tuning Data (Данные настройки) .....	4-26
$\alpha$ Series FANUC серводвигатели .....	4-27
$\alpha$ L Series FANUC серводвигатели .....	4-27
$\alpha$ C Series FANUC серводвигатели .....	4-27
$\alpha$ HV Series FANUC серводвигатели .....	4-28
$\alpha$ M Series FANUC серводвигатели .....	4-28
$\beta$ Series FANUC серводвигатели .....	4-28
$\beta$ M Series FANUC серводвигатели .....	4-28
Расчет переменных, определяющих предельные значения данных .....	4-33

Дополнительные табличные данные .....	4-33
Параметры настройки, поддерживаемые в версии 1.0 .....	4-33
Параметры настройки, поддерживаемые в версии 3.0 .....	4-34
Ввод номеров параметров настройки и данных в ячейки расширенной таблицы .....	4-35
Данные по энергопотреблению .....	4-35
<b>Глава 5. Взаимодействие модуля Motion Mate DSM314 с ПЛК.....</b>	<b>5-1</b>
<b>Раздел 1: Биты статуса %I.....</b>	<b>5-2</b>
<b>Раздел 2: %AI слова статуса .....</b>	<b>5-8</b>
<b>Раздел 3: Дискретные команды %Q .....</b>	<b>5-11</b>
<b>Раздел 4: Команды немедленного действия %AQ .....</b>	<b>5-17</b>
Цифровой режим.....	5-22
Пример использования команды Force Analog Output (Цифровой режим) .....	5-23
Аналоговый режим скорости .....	5-23
Аналоговый режим момента вращения .....	5-24
<b>Глава 6. Непрограммируемое движение .....</b>	<b>6-1</b>
<b>Цикл начальной позиции модуля DSM314.....</b>	<b>6-1</b>
Режим переключателя начальной позиции .....	6-2
Программа Find Home (Начальное позиционирование) для Датчика начальной позиции .....	6-2
Пример переключателя начальной позиции.....	6-3
Режимы Move+ (Движение+) и Move- (Движение-) .....	6-4
Пример цикла начальной позиции Move – (минус).....	6-4
Программа Find Home (Начальное позиционирование) для Move (Движение) + или Move (Движение) – .....	6-5
<b>Толчковое передвижение при помощи DSM314 .....</b>	<b>6-6</b>
<b>Команда Move at Velocity (Движение на скорости) .....</b>	<b>6-6</b>
<b>Команда Force Servo Velocity (Форсировка скорость сервоустройства) (ЦИФРОВЫЕ сервоустройства; Аналоговый режим момента вращения) .....</b>	<b>6-7</b>
<b>Команда Force Analog Output (Форсировка аналогового выхода) (АНАЛОГОВЫЕ сервоустройства с интерфейсом скорости).....</b>	<b>6-7</b>
<b>Команды Position Increment (Приращение позиции).....</b>	<b>6-8</b>
<b>Дополнительная информация .....</b>	<b>6-8</b>
<b>Глава 7. Программируемое движение .....</b>	<b>7-1</b>
Типы команд программ движения.....	7-3
<b>Предварительные условия для программируемого движения.....</b>	<b>7-5</b>
<b>Условия, приводящие к остановке программы движения.....</b>	<b>7-5</b>
<b>Основы программы движения .....</b>	<b>7-6</b>
Количество программ, подпрограмм и операторов .....	7-6

Формат .....	7-6
Одно- и многоосные программы и подпрограммы .....	7-6
Операторы определения программы и подпрограммы .....	7-6
Номера блоков и блоки синхронизации .....	7-6
Команды и синтаксис языка движения .....	7-7
Пустые пространства .....	7-7
Числовые константы .....	7-7
Комментарии .....	7-7
Ключевые слова программы движения .....	7-7
Переменные .....	7-8
Разделители .....	7-8
<b>Команды программы движения .....</b>	<b>7-9</b>
ACCEL .....	7-9
Номер блока .....	7-10
CALL .....	7-10
CMOVE .....	7-11
DWELL .....	7-11
ENDPROG .....	7-12
ENDSUB .....	7-12
JUMP .....	7-12
LOAD .....	7-12
PMOVE .....	7-13
PROGRAM .....	7-13
SUBROUTINE .....	7-14
Sync Block .....	7-15
VELOC .....	7-15
WAIT .....	7-16
<b>Структура программы и подпрограммы .....</b>	<b>7-17</b>
Структура одноосной программы .....	7-17
Пример одноосной программы .....	7-17
Структура многоосной программы .....	7-18
Пример многоосной программы .....	7-18
Структура одноосной подпрограммы .....	7-19
Пример одноосной подпрограммы .....	7-19
Структура многоосной подпрограммы .....	7-19
Пример многоосной подпрограммы .....	7-20
<b>Примеры использования команд .....</b>	<b>7-21</b>
Абсолютное или инкрементное позиционирование .....	7-21
Абсолютное позиционирование .....	7-21
Инкрементное позиционирование .....	7-21
Типы ускорения .....	7-22
Линейное ускорение .....	7-22
Ускорение по S-кривой .....	7-22
<b>Типы команд программируемого движения .....</b>	<b>7-23</b>
Движение позиционирования (PMOVE) .....	7-23
Непрерывное движение (CMOVE) .....	7-23
Программируемые движения .....	7-25
Пример 1: комбинирование PMOVE и CMOVE .....	7-25

Пример 2: Изменение режима ускорения на профиле.....	7-26
Пример 3: не хватает расстояния, чтобы выйти на заданную скорость .....	7-26
Пример 4: Зависание движения при отсутствии расстояния .....	7-26
Команда DWELL .....	7-27
Пример 5: DWELL .....	7-27
Команда Wait .....	7-28
Подпрограммы .....	7-28
Номера блоков и переходы .....	7-28
Безусловные переходы .....	7-29
Пример 6: Безусловный переход .....	7-29
Условные переходы .....	7-29
Пример 1 условного перехода: .....	7-30
Пример 2 условного перехода: .....	7-30
Пример 3 условного перехода: .....	7-30
Проверка условия перехода .....	7-31
Пример 7: Проверка условия перехода .....	7-31
Нормальный останов перед командой JUMP .....	7-32
Пример 8: Нормальный останов перед командой JUMP .....	7-32
Переход без остановки .....	7-33
Пример 9: Выполнение команды JUMP без остановки .....	7-33
Остановка перехода .....	7-34
Пример 10: Остановка перехода .....	7-34
Пример 11: Команда перехода с последующей командой PMOVE .....	7-35
Переходы с S-CURVE (S-кривая) .....	7-35
Переходы на S-кривой после средней точки ускорения или замедления .....	7-35
Пример 12: Переходы на S-кривой после средней точки ускорения или замедления .....	7-36
Переходы на S-кривой до средней точки ускорения или замедления .....	7-36
Пример 13: Переходы на S-кривой до средней точки ускорения или замедления .....	7-36
Переходы на S-кривой к более высокому значению ускорения при ускорении или к более низкому значению замедления при замедлении .....	7-37
Пример 14: Переходы на S-кривой к более высокому значению скорости при ускорении или к более низкому значению скорости при замедлении .....	7-37
<b>Дополнительная информация по программируемому движению .....</b>	<b>7-38</b>
Максимальная длительность ускорения .....	7-38
Пример 15: Максимальная длительность ускорения .....	7-38
<b>Остановка подачи модулем DSM314 .....</b>	<b>7-40</b>
Пример 16: Остановка подачи .....	7-40
<b>Подстановка скорости подачи .....</b>	<b>7-41</b>
Пример 17: Подстановка (коррекция) скорости подачи .....	7-41
<b>Многоосное программирование .....</b>	<b>7-42</b>
Пример 18: Многоосное программирование .....	7-42
<b>Параметры (P0-P255) в модуле DSM314 .....</b>	<b>7-44</b>
<b>Расчет значений ускорения, скорости и позиции .....</b>	<b>7-46</b>

<b>Кинематические уравнения.....</b>	<b>7-46</b>
<b>Сообщения об ошибках программы Motion Editor и предупреждающие сообщения.....</b>	<b>7-49</b>
Синтаксические ошибки.....	7-49
Семантические ошибки.....	7-49
Предупреждения.....	7-52
Использование сообщений об ошибках для поиска ошибок в программах движения.....	7-53
<b>Глава 8. Движение слежения.....</b>	<b>8-1</b>
Ведущие источники.....	8-2
Внешние ведущие входы.....	8-2
Пример 1: Отслеживание ведущего входа с текущей позицией оси Z.....	8-2
Внутренние генераторы команд ведущей оси.....	8-3
Пример 2: Отслеживание внутренней ведущей команды.....	8-3
<b>Отношение А:В.....</b>	<b>8-3</b>
Пример 3: Примеры отношений А:В.....	8-4
Пример 4: Изменение отношения А:В.....	8-5
<b>Фиксация скорости.....</b>	<b>8-5</b>
Пример 5: Фиксация скорости.....	8-5
<b>Работа в одном направлении.....</b>	<b>8-6</b>
Пример 9: Работа в одном направлении.....	8-6
<b>Включение слежения при помощи внешнего входа.....</b>	<b>8-6</b>
<b>Выключение слежения при помощи внешнего входа.....</b>	<b>8-7</b>
<b>Операция выключения слежения, сконфигурированная для инкрементного позиционирования.....</b>	<b>8-7</b>
<b>Управление линейным изменением скорости следящей оси.....</b>	<b>8-7</b>
Заданный источник режима слежения и опции подключения.....	8-11
Блок-схема контура управления следящего устройства.....	8-16
<b>Глава 9. Комбинация движения слежения и движения, задаваемого командами.....</b>	<b>9-1</b>
Пример 1: Движение слежения в комбинации с Jog (Толчок).....	9-1
Движение слежения в комбинации с программами движения.....	9-2
Пример 2: Движение слежения в комбинации с программой движения..	9-5
Последовательность управления:.....	9-5
<b>Глава 10. Введение в программирование Local Logic.....</b>	<b>10-1</b>
Введение в программирование Local Logic.....	10-1
Local Logic или логика ПЛК?.....	10-4
<b>Основы программирования программ Local Logic и движения.....</b>	<b>10-4</b>
Требования.....	10-4
Создание программы Local Logic.....	10-5
Программное обеспечение VersaPro.....	10-5
Программное обеспечение SIMPLICITY Machine Edition.....	10-8

<b>Таблица переменных Local Logic.....</b>	<b>10-9</b>
<b>Подключение редактора Local Logic к модулю DSM.....</b>	<b>10-11</b>
<b>Построение программы Local Logic .....</b>	<b>10-12</b>
Создание программы Local Logic .....	10-12
Пример программы Local Logic.....	10-14
Проверка синтаксиса Local Logic .....	10-16
VersaPro.....	10-16
SIMPLICITY Machine Edition .....	10-18
Установка параметров конфигурации аппаратного обеспечения для Local Logic.....	10-19
<b>Загрузка вашей программы Local Logic.....</b>	<b>10-22</b>
<b>Выполнение вашей первой программы Local Logic .....</b>	<b>10-25</b>
<b>Использование редактора программы движения .....</b>	<b>10-26</b>
Выполнение вашей первой программы движения .....	10-36
<b>Глава 11. Вводное руководство по Local Logic .....</b>	<b>11-1</b>
<b>Введение.....</b>	<b>11-1</b>
<b>Операторы.....</b>	<b>11-1</b>
<b>Комментарии .....</b>	<b>11-2</b>
<b>Переменные.....</b>	<b>11-2</b>
<b>Операторы.....</b>	<b>11-3</b>
Арифметические операторы.....	11-3
Операторы отношения .....	11-4
Побитовые логические операторы.....	11-4
<b>Связь "программа Local Logic / ПЛК / программа движения" .....</b>	<b>11-5</b>
<b>Примеры программирования Local Logic .....</b>	<b>11-5</b>
<b>Пример программы по ограничению момента вращения.....</b>	<b>11-6</b>
<b>Пример программы планировщика коэффициента усиления .....</b>	<b>11-7</b>
<b>Пример программы программируемого концевого переключателя .....</b>	<b>11-8</b>
<b>Пример программы триггерного выхода, основанного на позиции .....</b>	<b>11-9</b>
<b>Пример программы с кадрирующими стробами .....</b>	<b>11-11</b>
Локальная логическая программа кадрирующих стробов .....	11-11
<b>Глава 12. Синтаксис языка Local Logic .....</b>	<b>12-1</b>
<b>Введение.....</b>	<b>12-1</b>
<b>Синтаксические элементы.....</b>	<b>12-1</b>
Числовые константы .....	12-1
Локальные логические переменные .....	12-2
Локальные логические операторы.....	12-2
Операторы присваивания Local Logic.....	12-3
Условные операторы Local Logic .....	12-4
Пустое пространство .....	12-4

Комментарии.....	12-5
Директива PRAGMA.....	12-5
Локальные логические ключевые слова и операторы .....	12-6
<b>Включение и выключение Local Logic .....</b>	<b>12-6</b>
<b>Выходы / команды Local Logic .....</b>	<b>12-7</b>
<b>Арифметические операторы Local Logic.....</b>	<b>12-8</b>
Оператор + .....	12-8
Оператор - .....	12-9
Оператор * .....	12-9
Оператор /.....	12-10
Оператор MOD .....	12-11
Функция ABS .....	12-12
<b>Побитовые логические операторы Local Logic.....</b>	<b>12-13</b>
Оператор BWAND.....	12-13
Оператор BWOR.....	12-14
Оператор BWXOR.....	12-14
Оператор BWNOT .....	12-15
<b>Операторы сравнения.....</b>	<b>12-16</b>
<b>Ошибки выполнения программы Local Logic .....</b>	<b>12-17</b>
Статус переполнения .....	12-17
Деление на ноль .....	12-18
Предупреждение / ошибка - окончание времени ожидания .....	12-18
<b>Сообщения об ошибках Local Logic.....</b>	<b>12-19</b>
Сообщения об ошибках построения Local Logic.....	12-19
Синтаксические ошибки Local Logic.....	12-19
Пример: .....	12-19
Семантические ошибки Local Logic .....	12-20
Примеры: .....	12-20
Семантические предупреждения Local Logic .....	12-22
Сообщения об ошибках загрузки программ Local Logic .....	12-23
Ошибки выполнения программы Local Logic.....	12-25
<b>Глава 13. Переменные Local Logic .....</b>	<b>13-1</b>
<b>Типы переменных Local Logic.....</b>	<b>13-1</b>
<b>Системные переменные Local Logic.....</b>	<b>13-2</b>
Переменная First_Local_Logic_Sweep.....	13-2
Переменная Overflow (Переполнение).....	13-2
Переменная System_Halt (Системный_останов).....	13-3
<b>64-битовые регистры двойной точности .....</b>	<b>13-3</b>
<b>Цифровые выходы / CTL-переменные .....</b>	<b>13-4</b>
<b>Глава 14. Конфигурация Local Logic .....</b>	<b>14-1</b>
<b>Конфигурация CTL-битов .....</b>	<b>14-1</b>
<b>Новые CTL-биты CTL01-CTL32.....</b>	<b>14-2</b>

Варианты выбора конфигурации битов CTL01-CTL24.....	14-4
Функция FBSA и назначения CTL-битов .....	14-5
Конфигурация битов выходов лицевой панели.....	14-5
<b>Глава 15. Использование VersaPro с модулем DSM314 .....</b>	<b>15-1</b>
<b>Начало использования.....</b>	<b>15-1</b>
Начало работы с VersaPro .....	15-1
Изменение структуры окна VersaPro .....	15-3
<b>Начало процесса конфигурации .....</b>	<b>15-4</b>
<b>Конфигурация DSM314 .....</b>	<b>15-6</b>
Запись установок конфигурации на диск.....	15-8
<b>Подсоединение к ПЛК и сохранение на нем конфигурации .....</b>	<b>15-9</b>
Полезные пиктограммы панели инструментов .....	15-9
Подсоединение к ПЛК.....	15-9
Останов ПЛК .....	15-10
Операция сохранения .....	15-10
<b>Использование редактора Motion (Движение).....</b>	<b>15-11</b>
Выход на окно редактора Motion (Движение).....	15-11
Запись программы движения на диск .....	15-12
Сохранение программ и подпрограмм движения на ПЛК .....	15-12
Распечатка программ и подпрограмм движения.....	15-13
Печать .....	15-13
Печать отчета.....	15-13
<b>Выход в окно редактора Local Logic .....</b>	<b>15-15</b>
Запись на диск программы Local Logic.....	15-16
Сохранение программы Local Logic на ПЛК.....	15-16
Распечатка программы Local Logic .....	15-17
Печать .....	15-17
Печать отчета.....	15-17
Просмотр таблицы переменных Local Logic .....	15-18
<b>Глава 16. Использование функции электронного кулачка (CAM) .....</b>	<b>16-1</b>
<b>Раздел 1:        Введение.....</b>	<b>16-1</b>
<b>Краткое описание электронного САМ .....</b>	<b>16-1</b>
<b>Основные элементы формы кулачка (САМ) / определения .....</b>	<b>16-3</b>
<b>Раздел 2:        Синтаксис САМ.....</b>	<b>16-4</b>
<b>Типы САМ.....</b>	<b>16-4</b>
Нециклический САМ .....	16-4
Линейный циклический САМ .....	16-4

Круговой циклический САМ.....	16-5
<b>Интерполяция и сглаживание.....</b>	<b>16-7</b>
Сопрягающиеся сектора .....	16-8
1-й порядок с 1-ым порядком.....	16-8
1-й порядок с 2-ым порядком.....	16-8
2-й порядок с 1-ым порядком.....	16-8
2-й порядок с 2-ым порядком.....	16-8
2-й порядок с 3-им порядком .....	16-8
3-й порядок с 2-ым порядком.....	16-8
3-й порядок с 3-им порядком .....	16-9
Граничные условия .....	16-9
<b>Взаимодействие программ движения с САМ.....</b>	<b>16-10</b>
<b>САМ-команда .....</b>	<b>16-10</b>
<b>Команда САМ-LOAD .....</b>	<b>16-12</b>
<b>Команда САМ-PHASE .....</b>	<b>16-13</b>
<b>Команды САМ и MOVE .....</b>	<b>16-13</b>
<b>Движение САМ, контролируемое по времени .....</b>	<b>16-13</b>
<b>Редактор масштабирования САМ и конфигурация аппаратного обеспечения .....</b>	<b>16-14</b>
<b>Синхронизация САМ-движения с внешними событиями.....</b>	<b>16-18</b>
<b>Коды ошибок модуля DSM, специфические для САМ.....</b>	<b>16-19</b>
<b>Раздел 3:        Основы программирования электронного САМ.....</b>	<b>16-21</b>
<b>Требования .....</b>	<b>16-21</b>
<b>Введение в программирование электронного САМ .....</b>	<b>16-21</b>
Пример создания САМ .....	16-21
Базовые этапы.....	16-21
VersaPro.....	16-22
Этап 1: создание новой папки.....	16-22
Этап 2: создание блока САМ с использованием редактора САМ.....	16-23
Этап 3: Создание профиля САМ .....	16-24
Этап 4: Связывание профиля САМ с блоком САМ.....	16-26
Этап 5: конфигурация точек данных профиля САМ.....	16-27
Этап 6: задание типа САМ.....	16-29
Этап 7: Задание характеристики коррекции.....	16-30
Этап 8: сохранение на диске профиля САМ.....	16-31
Этап 9: создание программ движения и Local Logic.....	16-31
Этап 10: задание установочных параметров в конфигурации аппаратного обеспечения в VersaPro.....	16-34
Этап 11: выполнение (тестирование) вашей программы движения на базе САМ.....	16-44
<b>Приложение А.  Сообщения об ошибках .....</b>	<b>А-1</b>
<b>Коды ошибок модуля DSM314 .....</b>	<b>А-1</b>
Слово кода статуса модуля .....	А-1
Слова кода ошибки оси .....	А-1
<b>Формат кода ошибки.....</b>	<b>А-2</b>
<b>Методы реагирования .....</b>	<b>А-2</b>

Аварийные сообщения модуля DSM цифрового сервопривода (B0–BE).....	A-16
Аварийные сообщения цифрового сервопривода – поиск неисправностей.....	A-17
Светодиодные индикаторы.....	A-20
<b>Приложение В. Команды запросов на обмен данными модуля DSM314.....</b>	<b>B-1</b>
<b>Раздел 1: Краткое описание команд запросов на обмен данными.....</b>	<b>B-2</b>
Структура запроса на обмен данными .....	B-2
Корректирующие действия .....	B-3
Мониторинг слова статуса .....	B-4
Обнаружение ошибок и работа с ними .....	B-4
Проверка получения модулем DSM правильных данных .....	B-4
<b>Раздел 2: Команда COMM REQ Ladder .....</b>	<b>B-6</b>
Требования и рекомендации по программированию DSM COMM REQ.....	B-7
<b>Раздел 3: Команда COMM REQ "Таблица пользовательских данных" (UDT) .....</b>	<b>B-8</b>
Характеристики команды COMM REQ "Таблица пользовательских данных" и информация по использованию .....	B-8
Командный блок UDT COMM REQ.....	B-9
Пример команды COMM REQ "Таблица пользовательских данных" .....	B-11
Пример команды COMM REQ "Таблица пользовательских данных" .....	B-13
<b>Раздел 4: Команда COMM REQ "Загрузка параметров" .....</b>	<b>B-14</b>
Командный блок.....	B-14
Пример команды COMM REQ "Загрузка параметров" модуля DSM .....	B-17
<b>Раздел 5: Пример команды COMM REQ Ladder Logic.....</b>	<b>B-19</b>
Установка значений командного блока COMM REQ.....	B-19
Логика для данных параметров (не показана).....	B-20
Обработка значений параметров в виде двойных целых и масштабирование входных значений .....	B-20
Команда запроса на обмен данными .....	B-21
<b>Приложение С. Устройства обратной связи по положению .....</b>	<b>C-1</b>
Режимы цифрового последовательного датчика положения .....	C-1
<b>Информация по режиму инкрементного датчика положения.....</b>	<b>C-1</b>
<b>Информация по режиму абсолютного датчика положения .....</b>	<b>C-2</b>
Абсолютный датчик положения – первое использование или использование после отключения питания от батареи .....	C-2
Режим абсолютного датчика – инициализация позиции .....	C-2
Цикл начального позиционирования – режим абсолютного датчика положения .....	C-2
Команда установки позиции – режим абсолютного датчика положения.....	C-3

Режим абсолютного датчика положения – включение питания модуля DSM314 .....	C-3
Инкрементный импульсный датчик положения.....	C-4
<b>Приложение D. Пуск и настройка цифровых и аналоговых сервосистем GE Fanuc .....</b>	<b>D-1</b>
<b>Информация о запуске и настройке цифровых сервосистем.....</b>	<b>D-1</b>
<b>Проверка правильности работы датчика начальной позиции, входов перебега и направления вращения двигателя .....</b>	<b>D-1</b>
Рекомендации по поиску неисправностей при пуске цифровой сервосистемы. D-3	
<b>Настройка цифрового сервопривода GE Fanuc.....</b>	<b>D-5</b>
Требования к настройке .....	D-5
Настройка ЦИФРОВОГО РЕЖИМА контура скорости.....	D-6
Метод #1.....	D-6
Метод #2.....	D-6
Уравнение 1 .....	D-6
Пример проведения настройки ЦИФРОВОГО РЕЖИМА контура скорости... D-8	
Настройка контура позиционирования .....	D-13
Терминология .....	D-13
Широкая полоса пропускания .....	D-14
<b>Процедуры запуска системы интерфейса режима аналоговой скорости .....</b>	<b>D-15</b>
Процедуры запуска.....	D-15
<b>Процедуры запуска системы интерфейса режима аналогового момента вращения.....</b>	<b>D-17</b>
Процедуры запуска.....	D-17
Настройка контура скорости в режиме момента вращения.....	D-20
Метод #1.....	D-20
Метод #2.....	D-20
Уравнение 2 .....	D-22
Пример настройки контура скорости.....	D-24
<b>Рекомендации по поиску и устранению неисправностей системы (аналоговый режим) .....</b>	<b>D-31</b>
<b>Приложение E. Длительность выполнения программ Local Logic.....</b>	<b>E-1</b>
<b>Информация о длительности выполнения программ Local Logic.....</b>	<b>E-1</b>
<b>Пример 1 .....</b>	<b>E-2</b>
<b>Пример 2 .....</b>	<b>E-3</b>
<b>Приложение F. Обновление микропрограммы в модуле DSM314.....</b>	<b>F-1</b>
<b>Чтобы установить новую микропрограмму, выполните следующие действия:.....</b>	<b>F-1</b>
Обновление в DOS .....	F-1
Обновление под Windows (для Windows 95/NT/98, <u>НЕ ПРЕДНАЗНАЧЕНО ДЛЯ Windows 3.1</u> ).....	F-2
<b>Повторный запуск прерванного обновления микропрограммы .....</b>	<b>F-3</b>

<b>Приложение G. Расчеты точности стробов .....</b>	<b>G-1</b>
<b>Аналоговый режим.....</b>	<b>G-1</b>
<b>Цифровой режим.....</b>	<b>G-1</b>

# Глава 1

## Краткое описание устройства

Модуль Motion Mate DSM314 является высокопроизводительным, простым в эксплуатации модулем управления движением по нескольким осям, который тесно связан с логикой и коммуникационными функциями ПЛК Series 90-30.

DSM314 поддерживает две конфигурации первичного контура управления:

- Стандартный режим (Следящий Контур Управления Выключен)
- Режим Слежения (Следящий Контур Управления Включен)

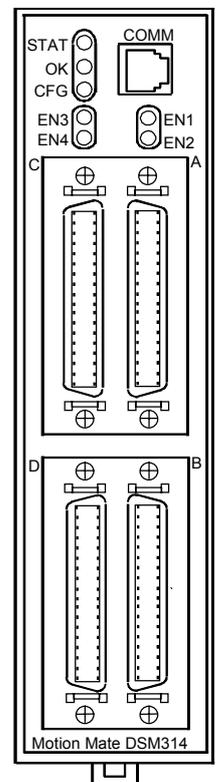
### Поддерживаемые типы сервоустройств

- Цифровые: цифровые сервоусилители и сервоприводы GE Fanuc  $\alpha$  Series и  $\beta$  Series. Эти изделия описаны в руководстве GFK-001.
- Аналоговые: аналоговые сервоустройства GE Fanuc SL-Series, а также аналоговые сервоустройства командного интерфейса управления скоростью и моментом, выпускаемые другими организациями. Сервоустройства GE Fanuc SL-Series описаны в руководстве пользователя "Сервоустройства SL-Series", GFK-1581.

## Характеристики модуля Motion Mate DSM314

### Высокая производительность

- Управление сервоустройством GE Fanuc от цифрового процессора обработки сигналов (DSP)
- Время обработки сигналов менее 5 мсек
- Коэффициент упреждения по скорости и Интегратор Ошибки позиционирования для улучшения точности отслеживания
- Высокое разрешение программных устройств
  - Позиция: -536 870 912...+536 870 911 пользовательских единиц
  - Скорость: 1 ... 8 388 607 пользовательских единиц / секунда
  - Ускорение: 1 ... 1 073 741 823 пользовательских единиц / секунда<sup>2</sup>



## Простота использования

- Простая и мощная система команд для программирования движения
- Простые программы движения по осям от 1 до 4-х. Многоосные программы, использующие Оси 1 и 2, допускают синхронный запуск блока.
- Энергонезависимое ЗУ на 10 программ и 40 подпрограмм, создаваемых при помощи соответствующего программного обеспечения.
- Совместим с центральными процессорами (ЦП) Series 90-30, оснащенными микропрограммой версии 10.0 или более поздней (не работает с ЦП 311 – 341 и 351). Более детальную информацию по ЦП см. в "Руководстве по установке и по аппаратному обеспечению ПЛК Series 90-30", GFK-0356P или более позднее.
- Единая точка подключения (входа) для всех задач программирования и конфигурации, включая создание программ движения (Программы движения 1 – 10) и программирование Local Logic. Все компоненты программ и конфигурации загружаются через программируемый коммуникационный порт ПЛК. В свою очередь, ЦП загружает в модуль DSM314 все компоненты конфигурации и программ движения, а также программы Local Logic через объединительную плату ПЛК.
- Пользовательское масштабирование программируемых единиц (Пользовательские единицы) как для Стандартного режима, так и для режима Слежения.
- Микропрограммное обеспечение DSM314, находящееся во флэш-памяти, обновляется через СОММ-порт на передней панели. Комплекты обновления микропрограммного обеспечения предоставляют собой микропрограммное и загрузочное (Loader) ПО на флоппи-диске. Микропрограммное обеспечение может быть также загружено с веб-сайта GE Fanuc (<http://www.gefanuc.com/support>).
- Программирование сценариев, использующее параметры команд в качестве операндов для команд Ускорение (Acceleration), Скорость (Velocity), Движение (Move) и Задержка (Dwell).
- Автоматическая Передача Данных между таблицами ПЛК и DSM314 без использования пользовательского программирования.
- Простота подключения входов / выходов при помощи заводских кабелей и разъемов.
- Возможность электронной САМ, действующая с Микропрограммой версии 2.0.

## Универсальность и гибкость входов / выходов

- Управление цифровыми сервоприводами GE Fanuc  $\alpha$  Series и  $\beta$  Series, сервоприводами SL-Series, а также сервоприводами третьих фирм с интерфейсом на основе команд аналогового управления скоростью или команд аналогового управления вращающим моментом.
- Входные сигналы перехода в исходную позицию и перерегулирования для каждой Оси Сервопривода.
- Два положения входного строба захвата для каждой оси может использоваться при захвате оси и/или позиционирование ведущего, с точность  $\pm 2$  импульса плюс колебания в 10 мсек.
- 5 В, 24 В и аналоговый вход / выход для использования на ПЛК
- Вход Инкрементного Импульсного Датчика положения по каждой оси для Датчика положения / Аналоговый режим
- Вход Импульсного Датчика положения для Ведущей оси Слежения
- 13 битовый Аналоговый Выход может управляться при помощи ПЛК или использоваться как Цифровой монитор Настройки Сервопривода
- Высокоскоростной цифровой выход (четыре по 24 В и четыре по 5 В) через встроенное управление Local Logic

## Раздел 1: Краткое описание системы движения

Модуль DSM314 является микропроцессорным, полностью программируемым модулем управления движением для Программируемого Логического Контроллера (ПЛК) Series 90-30. DSM314 позволяет пользователям ПЛК объединять в единой системе возможности высокопроизводительного управления движением и возможности программирования Local Logic с функциями управляющей программы ПЛК. На рисунке внизу показано аппаратное и программное обеспечение, используемое для установки и эксплуатации сервосистемы. В данном разделе дано краткое описание каждого элемента системы, направленное на общее понимание ее работы.

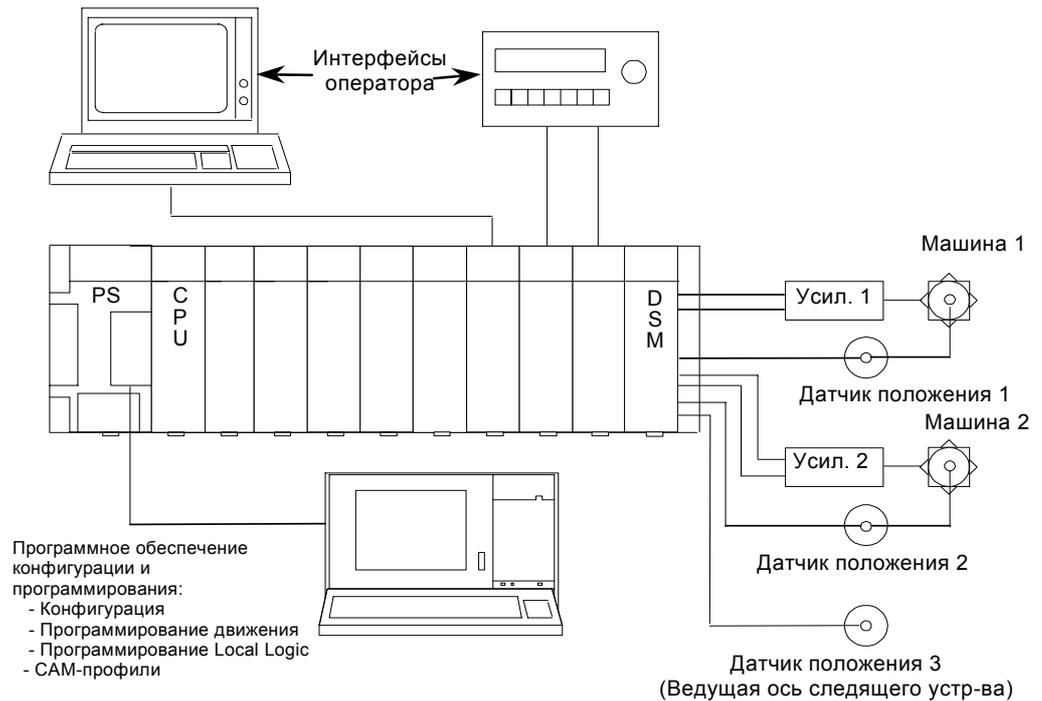


Рис. 1-1. Аппаратно-программное обеспечение, используемое для конфигурации, программирования и эксплуатации сервосистемы DSM314.

## ПЛК Series 90-30 и модуль DSM314

DSM314 и ПЛК Series 90-30 работают вместе как единая система управления движением. DSM314 обменивается данными с ПЛК через системную шину. На каждом цикле ПЛК происходит преобразование таких данных, как *Commanded Velocity (Заданная Скорость)* и *Actual Position (Текущая Позиция)*, внутри модуля DSM314 в %I и %AI данные ПЛК. Аналогично на каждом цикле ПЛК происходит передача данных %Q и %AQ от ПЛК к DSM314. Данные %Q и %AQ используются для управления устройством DSM314. Биты %Q выполняют такие функции, как инициация движения, прекращение движения и очистка стробовых флажков. Команды %AQ выполняют такие функции, как инициализация регистров параметров позиции и загрузки.

Помимо использования адресов %I, %AI, %Q и %AQ имеется дополнительный путь отправки параметров от ПЛК к DSM314 при помощи команды COMM\_REQ в Ladder-программе. Более детальная информация об использовании команды COMM\_REQ с устройством DSM содержится в Приложении В, команды DSM314 COMM\_REQ.

### Задержка данных ПЛК и задержки DSM314

DSM314 является микропроцессорным модулем, работающим асинхронно с модулем ЦП Series 90-30. Обмен данными между ЦП и DSM314 происходит автоматически. Информация о цикле работы Series 90-30 содержится в "Справочном руководстве по системе команд ЦП ПЛК Series 90-30" GFK-0467. Далее дан анализ временных интервалов, применимый к модулю DSM314.

#### Передача данных от ПЛК к DSM

- Функции, которыми обладает ПЛК, позволяют ему извлекать информацию о статусе модуля DSM (%I и %AI) асинхронно из памяти данных DSM. DSM внутри обновляет все данные о статусе, кроме Actual Velocity (Текущая Скорость), с частотой контура позиционирования (один раз каждые 0,5 – 2 мс). Информация о Текущей Скорости обновляется в памяти данных DSM каждые 128 миллисекунд. DSM выполняет усреднение для получения точного значения Текущей Скорости; поэтому значение Текущей Скорости не предназначено для высокоскоростного управления.
- ПЛК требуется, примерно, 2 – 4 миллисекунды на служебные операции на объединительной плате при чтении данных (%I и %AI) с внутренней памяти модуля DSM и записи данных (%Q и %AQ) туда же, если DSM расположен в крейте ЦП. ПЛК обычно считывает входные данные из DSM и записывает выходные данные на это устройство один раз за цикл ПЛК. В худшем случае обновление внутренних данных DSM (которое занимает от 0,5 до 2 мс) происходит сразу же после того, как произошло обновление входных данных в ходе сканирования ПЛК. В этом случае ПЛК не обращается к данным DSM до следующего цикла сканирования; любые изменения в состоянии DSM становятся доступными в ПЛК либо через 4 – 6 мс, либо, примерно, через один цикл ПЛК, в зависимости от того, какой интервал длиннее.
- Программа конфигурации автоматически выбирает длину данных %AI и %AQ в зависимости от количества конфигурируемых осей. Для ЦП ПЛК требуется время, чтобы считать и записать данные через системную шину, связывающую его с DSM314. "Справочное руководство по системе команд ПЛК Series 90-30", GFK-0467 версии M или более поздней, описывает длительность цикла для группы моделей ЦП, если выбраны различные конфигурации осей. Кроме того, дополнительная информация содержится в документе "Важная информация об изделии", который находится в упаковке модуля DSM.
- Команды от ПЛК к DSM (%Q, %AQ) являются выходами на DSM в конце цикла управляющей программы ПЛК. DSM обрабатывает эти команды в течение 4 миллисекунд после получения.

## Программа движения / СТЛ-входы лицевой панели

- Задержки, связанные с управлением программой движения или ветвлением через СТЛ-входы лицевой панели, определяются интервалами обновления контура позиционирования (от 0,5 до 2 мс) плюс задержка входного фильтра (обычно 5 мс для СТЛ-входов на 24 В или 10 мс для СТЛ-входов на 5 В). См. таблицы 1-1, 1-2 и 1-3 для различных интервалов обновления контура позиционирования.

## Local Logic

- Задержки, связанные с обновлением данных Local Logic, определяются интервалом обновления контура позиционирования (см. следующий раздел "Интервалы Обновления Контура Сервопривода в DSM314") и не связаны со сканированием ПЛК. Благодаря этому программы Local Logic могут быстро использовать изменяющиеся внутренние данные модуля DSM, что не может быть выполнено при помощи ЦП ПЛК из-за наличия времени переноса данных от ПЛК к DSM и из-за большего интервала сканирования у ПЛК.

## Интервалы обновления контура сервопривода в DSM314

При управлении цифровым сервоприводом переменного тока GE Fanuc модуль DSM314 использует интервалы обновления контура управления, показанные в таблице 1-1.

**Таблица 1-1. Интервалы обновления контура цифрового сервопривода GE Fanuc**

Контур Тока Двигателя / Момент вращения	250 микросекунд
Контур Скорости Двигателя	1 миллисекунда
Контур Позиционирования Двигателя	2 миллисекунды

При управлении Аналоговым сервоприводом модуль DSM314 **без** Local Logic использует интервалы обновления контура управления, показанные в таблице 1-2.

**Таблица 1-2. Интервалы обновления контура аналогового сервопривода без Local Logic**

1 (одна) Ось - Контур Позиционирования без Local Logic:	0,5 миллисекунды
2 Оси - Контур Позиционирования без Local Logic:	1 миллисекунда
3 - 4 Оси - Контур Позиционирования без Local Logic:	2 миллисекунды

При управлении Аналоговым сервоприводом модуль DSM314 **с** Local Logic использует интервалы обновления контура управления, показанные в таблице 1-3. Интервалы обновления контура управления с Local Logic больше, т.к. используется временной интервал Axis #4 для расчета функции Local Logic.

**Таблица 1-3. Интервалы обновления контура аналогового сервопривода с Local Logic**

1 (одна) Ось - Контур Позиционирования с Local Logic:	1 миллисекунда
2 -3 Оси - Контур Позиционирования с Local Logic:	2 миллисекунды

Аналоговый режим Момент вращения включает в себя регулятор скорости в дополнение к упомянутым выше регуляторам позиции. Для оси в Аналоговом режиме Момент вращения регулятор скорости запускается каждые 0,5 миллисекунды.

## DSM314 - стробы позиции

Каждый разъем оси на лицевой панели DSM314 имеет два входа Стробов Позиции. Нарастающий фронт импульса на входе "Строб" вызывает захват *Текущей позиции* оси. Разрешение захвата позиции составляет +/- 2 отсчета с дополнительным отклонением 10

микросекунд для задержки на входном фильтре stroba. Фактическая наблюдаемая ошибка зависит от ускорения сервопривода и входной фильтрации / дискретизации stroba. Формулы расчета точности stroba приведены в Приложении G.

Данные по strobu обновляются в течение одного интервала обновления контура позиционирования (0,5 – 2 мс) в соответствующем регистре данных *Strobe Position* (*Позиция stroba*) в памяти %AI. Данные по *Позиции stroba* сохраняются также в регистре DSM-параметров и могут быть использованы в качестве операнда для команд PMOVE и CMOVE Программы движения, а также в Local Logic. Обновление данных *Позиция stroba* в ПЛК зависит от длительности цикла ПЛК и может быть более 2 мс.

В Цифровом режиме эти stroбы представляют собой 5В несимметричные / дифференциальные входы (IN1-IN2).

В Аналоговом режиме эти stroбы являются только 5В несимметричными (IO5-IO6). Только в Аналоговом режиме эти stroбовые входы (как видно на битах статуса Stroba %I ПЛК) "загоняются вверх", если они не подключены физически к устройству.

## Увеличение длительности цикла ПЛК при работе с DSM314

Приведенная ниже таблица показывает, сколько миллисекунд добавляет DSM314 к длительности сканирования ПЛК. Величина этого времени зависит (1) от числа сконфигурированных осей DSM314 и (2) от типа крейта (главный крейт, расширительный крейт или дистанционный крейт), где смонтировано устройство DSM314.

Число Сконфигурированных Осей	Увеличение длительности цикла ПЛК (в миллисекундах)		
	Главный крейт	Расширительный крейт	Дистанционный крейт
1	1.6	2.6	6.9
2	2.2	3.8	9.9
3	2.8	4.3	13.0
4	3.3	5.2	15.9

### Примечания:

1. Не забывайте, что внутренний процессор Local Logic DSM314 имеет максимальную длительность сканирования 2 мс, которая не зависит от сканирования ПЛК. Это дает возможность пользователю управлять задачами движения, критическими ко времени, в программе Local Logic. Более детальную информацию по программированию Local Logic см. в соответствующих главах данного руководства.
2. В приложениях, в которых указанное выше увеличение длительности сканирования влияет на работу станка, может оказаться необходимым использовать такие особенности, как "подвешенный вход/выход", функции "DOIO" и "SNAP", чтобы избирательно переносить требуемые данные к DSM314 и от него. Эти особенности позволяют устранить перенос всех данных %I, %Q, %AI, %AQ при каждом сканировании, если эти данные не требуются часто, что благоприятно сказывается на длительности цикла сканирования.

## Программное обеспечение

Модуль DSM314 требует один из следующих пакетов программного обеспечения (ПО) конфигурации и программирования:

- SIMPLICITY Machine Edition Logic Developer PLC-версия 2.1 или более поздняя.
- VersaPro, версия 1.1 или более поздняя.

Пакет программ для программирования / конфигурации используется для решения указанных далее задач. Информация, создаваемая этими задачами, направляется в DSM314 через объединительную плату ПЛК при каждом его включении.

- Конфигурация. Позволяет пользователю выбрать установки модуля и рабочие параметры по умолчанию.
- Создание программы движения. Допускается создание до 10-ти программ движения и 40 подпрограмм.
- Создание программы Local Logic. Программа Local Logic работает синхронно с программой движения, но не зависит от сканирования ЦП ПЛК. Это позволяет модулю DSM314 быстро взаимодействовать с сигналами входа / выхода движения на разъемах его лицевой панели. Время внутренней реакции на сигналы входа / выхода движения намного меньше, чем время, требующееся для обработки логики таких сигналов основной программе лестничной логике, работающей в ПЛК. Последнее связано с (1) задержкой в обмене сигналами через системную плату и (2) с более длительным циклом в ПЛК.
- Создание САМ-профиля. САМ-профиль определяет реакцию следящего сервопривода на показатель точности ведущего. Ссылка на САМ-профили осуществляется по имени в соответствующей программе движения.

**Примечание:** Редактор САМ полностью интегрирован в программу Logic Developer PLC. Для программирования электронных САМ-характеристик требуется программа VersaPro версии 1.5 или более поздней с встроенным Редактором САМ.

## Интерфейсы оператора

Интерфейсы оператора обеспечивают возможность оператору управлять сервосистемой и контролировать при помощи панели управления или ЭЛТ-дисплея. Эти интерфейсы взаимодействуют с ПЛК через цифровые модули входа / выхода или через программируемые последовательные линии связи или через модули сетевой связи.

Информация оператора автоматически передается между ПЛК и DSM314 при помощи обращений к памяти %I, %AI, %Q и %AQ, задаваемых при конфигурации модуля. Эта автоматическая передача данных обеспечивает гибкий и простой интерфейс для нескольких операторских интерфейсов, которые могут взаимодействовать с ПЛК Series 90-30.

## Разъемы сервопривода и станка

Разъем сервопривода и станка представляет собой 36-контактный разъем для каждой оси. По этому разъему передаются сигналы управления положением оси, такие как Широтно-Импульсная Модуляция (PWM) на усилитель, сигналы Цифровой Последовательной Обратной связи от Датчика положения или Обратная связь по Командам Аналогового Сервопривода и Импульсного Датчика положения. Разъем обеспечивает, кроме того, входы *Датчик начальной позиции* и *Пере регулирование оси*, а также входы и выходы общего назначения на ПЛК.

Стандартные кабели, подключаемые непосредственно к оконечному устройству, смонтированному на пользовательской DIN-рейке или на панели, упрощают пользовательскую разводку; кабели могут быть поставлены GE Fanuc. Оконечные блоки обеспечивают резьбовые точки подсоединения для временного подключения к модулю DSM314. Дополнительную информацию по кабелям и оконечным блокам, используемым с модулем DSM314, см. в главе 3.

## Раздел 2: Краткое описание работы модуля DSM314

Каждая ось модуля DSM314 может работать с включенным или выключенным Следящим Контуром Управления (Follower Control Loop).

### Стандартный режим (следающий контур управления = выключен)

- \* В Цифровом Стандартном Режиме модуль обеспечивает управление позиционированием, скоростью и моментом вращения по замкнутому контуру одним или двумя сервоприводами GE Fanuc  $\alpha$  Series или  $\beta$  Series на Оси 1 и Оси 2. Ось 3 может использоваться как интерфейсная команда аналоговой скорости сервопривода оси или ведущую ось «Aux».(дополнительной) или дополнительная ведущая ось
- \* В Аналоговом Стандартном Режиме модуль обеспечивает управление по замкнутому контуру позиционирования сервоприводами количеством до 4-х. Также, в зависимости от конфигурации оси, модуль DSM обеспечивается контур управления по скорости для Аналогового режима Момент вращающ. Когда модуль DSM используется с сервоприводами, имеющими интерфейс аналоговой скорости, то контуры управления по скорости и вращающему моменту замыкаются в сервоусилителе, тогда как сам DSM замыкает контур позиционирования. Когда DSM используется с сервоприводами, имеющими интерфейс аналогового момента вращения, то контур управления по вращающему моменту замыкается в сервоусилителе, тогда как сам DSM замыкает контура скорости и позиционирования.
- \* Как для цифровых, так и для аналоговых применений, программируемые пользовательские единицы могут быть настроены при конфигурировании соотношения конфигурационных параметров Пользовательских Единиц и Отчетов. Команды Jog (Толчок), Move at Velocity (Движение на Скорости) и Execute Motion Program (Программа Выполнения Движения) позволяют использовать Стандартный режим в широком диапазоне применения при позиционировании.

### Режим слежения (следающий контур управления = включен)

- \* В Цифровом Режиме Слежения модуль обеспечивает управление позиционированием, скоростью и моментом вращения по замкнутому циклу одним или двумя сервоприводами GE Fanuc  $\alpha$  Series или  $\beta$  Series на Оси 1 и Оси 2. Ось 3 может быть использована как сервоось Командного Интерфейса Аналоговой Скорости или как ведущая ось «Aux».
- \* В Аналоговом Режиме Слежения модуль обеспечивает управление по замкнутому контуру позиционирования сервоприводами количеством до 4-х (одна или две из четырех имеющихся осей могут быть использованы вместо этого в качестве ведущей оси Aux). Также в зависимости от конфигурации оси модуль обеспечивает контур управления по скорости для Аналогового режима Момент вращающ. Когда модуль DSM используется с сервоприводами, имеющими интерфейс аналоговой скорости, то контура управления по скорости и вращающему моменту замыкаются в сервоусилителе, тогда как сам DSM замыкает контур позиционирования. Когда модуль DSM используется с сервоприводами, имеющими интерфейс аналогового момента вращения, то контур управления по вращающему моменту замыкается в сервоусилителе, тогда как сам DSM замыкает контура скорости и позиционирования.
- \* Как для цифровых, так и для аналоговых применений, модуль обеспечивает те же характеристики, что и в Стандартном режиме, включая конфигурируемое соотношение между Пользовательскими Единицами и Счетами.
- \* Кроме того, может быть сконфигурирован вход позиционирования Ведущей Оси. Каждая ось Следящего устройства отслеживает вход Ведущей Оси при программно задаваемом отношении (A:B). Движение, определяемое командами Толчок, Движение на Скорости и Программа Выполнения Движения, может быть объединены с последовательным движением от ведущей оси.
- \* Возможности следящего устройства:

- Набор параметров Ведущей Оси может быть сконфигурирован как Текущая или Заданная Позиция из любой другой оси
- Бит Выбора Ведущего Набора параметров %Q переключает между двумя предварительно сконфигурированными наборами параметров Ведущей Оси.
- Изменение Ускорения позволяет плавно разгонять ведомую ось, пока ее позиция и скорость не синхронизируются с параметрами ведущей оси.
- Наборы параметров следящего устройства могут включаться и выключаться по отдельности.

Примечание: Режим Намотки (Winder) не поддерживается в модуле DSM314. Он поддерживается в модуле DSM302.

## Работа в стандартном режиме

На рис. 1-2 представлено упрощенное изображение Контра Позирования в Стандартном режиме. Генератор Команд внутреннего движения выдает Заданную Позицию и Заданную Скорость на Контур Позирования. В Контуре Позирования происходит вычитание Текущей Позиции (Обратная связь по Положению) из Заданной Позиции и выдается Ошибка по Положению (рассогласование). Значение Ошибки по Положению умножается на Коэффициент усиления Контра Позирования для выработки Команды Скорости Сервопривода. Для снижения Ошибки по Положению, возникающей из-за движения сервопривода, Заданная Скорость, выдаваемая Генератором Команд, суммируется в виде слагаемого *Коэффициент упреждения по скорости* и дает на выход Команду Скорости Сервопривода.

В данные, передаваемые от DSM314 на ПЛК, входит следующее:

<b>Заданная Скорость</b>	- мгновенная скорость, вырабатываемая в DSM314 внутренним генератором пути
<b>Заданная Позиция</b>	- мгновенная позиция, вырабатываемая в DSM314 внутренним генератором пути
<b>Текущая Скорость</b>	- скорость оси, определяемая по обратной связи
<b>Текущая Позиция</b>	- позиция оси, определяемая по обратной связи
<b>Ошибка по Положению</b>	- разница между <i>Заданной Позицией</i> и <i>Текущей Позицией</i>

Модуль DSM314 позволяет запрограммировать *Постоянную Времени Контра Позирования* (в единицах 0,1 миллисекунды) и *Коэффициент упреждения по скорости* (в единицах 0,01 процента). *Постоянная Времени Контра Позирования* задает Коэффициент Усиления Контра Позирования и определяет скорость реакции замкнутого Контра Позирования. Значение процента *Коэффициент упреждения по скорости* определяет ту часть *Заданной Скорости*, которая вводится в Команду Скорости Сервопривода.

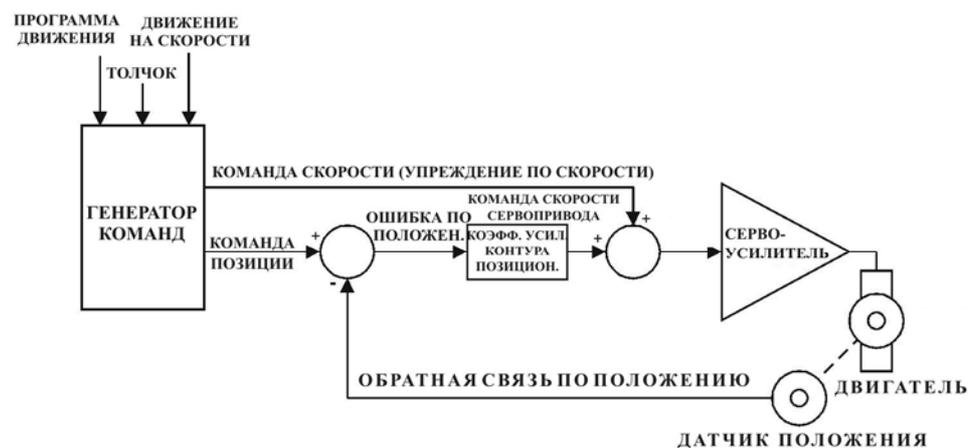


Рис. 1-2. Упрощенное изображение контра позиционирования в стандартном режиме с коэффициентом упреждения по скорости (аналоговый интерфейс скорости)

## Работа в режиме слежения

На рис. 1-3 представлено упрощенное изображение Контура Positionирования в режиме *Слежение*. Он похож на Контур Positionирования в *Стандартном* режиме (см. предыдущую страницу) с добавлением входа Ведущей Оси. Вход Ведущей Оси представляет собой дополнительный источник команд, выдающий Позицию Ведущей Оси и Скорость Ведущей Оси. Позиция Ведущей Оси суммируется с *Заданной Позицией*, поступающей от Генератора Команд оси. Скорость Ведущей Оси суммируется с выходом *Заданной Скорости (Коэффициент упреждения по скорости)* от Генератора Команд оси. Таким образом, позиция и скорость серводвигателя определяются суммой выхода Генератора Команд и входа Ведущей Оси. Чтобы создать движение Оси Сервопривода, Генератор Команд и Ведущая Ось могут работать одновременно или независимо.

Модуль DSM314 допускает иметь несколько источников для входа Ведущей Оси:

- Ось 1 Заданная Позиция
- Ось 1 Текущая Позиция (Датчик положения Оси 1)
- Ось 2 Заданная Позиция
- Ось 2 Текущая Позиция (Датчик положения Оси 2)
- Ось 3 Заданная Позиция
- Ось 3 Текущая Позиция (Датчик положения Оси 3)
- Ось 4 Заданная Позиция
- Ось 4 Текущая Позиция (Датчик положения Оси 4)

Коэффициент, в соответствии с которым Ось Сервопривода отслеживает Ведущую Ось, программируется как отношение двух целых чисел. Например, можно запрограммировать перемещение Оси Сервопривода на 125 единиц Обратной связи по Положению на каждые 25 единиц позиции Ведущей Оси. Тогда при каждом изменении позиции Ведущей Оси на 1 будет происходить перемещение Оси Сервопривода на  $(125 / 25) = 5$  единиц Обратной связи по Положению.

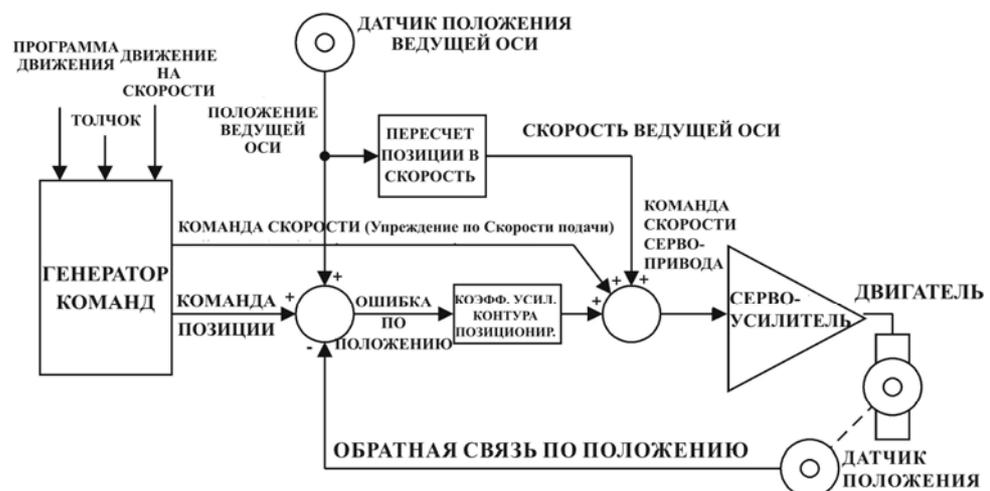


Рис. 1-3. Упрощенное изображение контура позиционирования в режиме слежения с входом ведущей оси (аналоговый интерфейс скорости)

## Раздел 3: Сервоприводы $\alpha$ Series (цифровой режим)

Особенности Цифрового Сервопривода  $\alpha$  Series (произносится как "альфа") GE Fanuc:

- Самая высокая на мировом рынке надежность
- Малые затраты на техническое обслуживание, отсутствие дрейфа, отсутствие коллекторных щеток
- Все параметры задаются в цифровом виде, не требуется повторная настройка
- При применении абсолютного датчика положения не требуется установка начального положения (требуется дополнительно комплект аккумуляторов)
- По спецзаказу возможна установка тормоза двигателя
- Для большинства двигателей возможно по спецзаказу выполнение в классе защиты IP67
- Обратная связь при помощи датчика положения с высоким разрешением - 64К импульсов на оборот (инкрементный или абсолютный)

Серводвигатели доказали свои свойства на более чем трех миллионах осей по всему миру; они демонстрируют свою высочайшую надежность и работоспособность. Новейшие технологические достижения, такие как высокоскоростные последовательные датчики положения и высокоэффективные Интегрированные Модули Питания (IPMs), еще более расширяют возможности для пользователей.

Сервосистема GE Fanuc является уникальной в том, что все контуры управления – по току, скорости и позиции – собраны в контроллере движения. Такой подход уменьшает время на установку и дает значительный выигрыш в производительности даже в самых трудных приложениях.

Эти сервоприводы требуют меньше расходов на их встраивание и техническое обслуживание. Цепи управления не подвержены температурным воздействиям. Специализированные модули-приставки отсутствуют. Эти сервоприводы имеют широкую область применимости, т.е. широкий диапазон инерционности нагрузок, гибкость ускорения и торможения, конфигураций обратной связи по положению и т.д.

Для оптимизации работы и преодоления машинных ограничений имеется значительный инструментарий настройки для пользователя. Сервоусилители на базе Интегрированных Модулей Питания (IPM) требуют на 60% меньше габаритного пространства, чем обычные усилители; они также излучают на 30% меньше тепла.

### Интегральный цифровой усилитель $\alpha$ Series (SVU)

Интегральный Сервоусилитель  $\alpha$  Series (SVU) представляет собой усилитель с встроенным источником питания, выполненные в виде автономного блока. Этот блок имеет те же габариты и посадочные места, что и предыдущая "С"-серия Сервоусилителей GE Fanuc.

Интегральные SVU Усилители  $\alpha$  Series используют те же контакты, что и усилители "С" серии, за исключением того, что здесь цепь Аварийного Останова использует внутренний источник 24В, поэтому более не требуется источник питания на 100В.

На конструкции SVU тепло выделяется через панель, что позволяет выводить тепло за пределы корпуса.

Поскольку Усилители  $\alpha$  SVU не обеспечивают регенеративный режим работы, могут понадобиться тормозные резисторы. Имеется несколько типоразмеров таких резисторов.

Сервоусилители  $\alpha$  Series типа SVU поставляются в пяти типоразмерах с ограничением максимального номинального пикового тока от 12 до 130 А. (Примечание: В данный момент GE Fanuc NA предлагает модели только на 80 и 130 А).

Имеются кабели различной длины для подключения Усилителей SVU к DSM314 и к двигателям.

Более подробную информацию о сервоустройствах  $\alpha$  Series см. в издании GFH-001 "Справочник по Техническим Характеристикам Сервоустройств").

## Серводвигатели $\alpha$ Series

Серводвигатели  $\alpha$  Series имеют улучшения конструкции, которые обеспечивают максимально возможные эксплуатационные характеристики. Предлагаются номиналы до 56 нм. Эти двигатели на 15% короче и легче серводвигателей предыдущей S-серии. Новая изоляция в обмотке и общее изоляционное покрытие защищает двигатель от окружающей среды.

Стандартный датчик положения, поставляемый вместе с двигателем, является абсолютным датчиком на 64К. Дополнительно могут быть установлены удерживающие тормоза (90 В пост. тока) и выполнена герметизация IP67. Серводвигатели  $\alpha$  Series сертифицированы на соответствие международным стандартам для ЕС (по электромагнитной совместимости (EMC) и по низковольтной аппаратуре), IEC и UL/CUL. В приведенной таблице даны примеры имеющихся двигателей  $\alpha$  Series (имеются также некоторые  $\alpha$ L,  $\alpha$ C,  $\alpha$ HV и  $\alpha$ M).

Более детальную информацию см. в главе 4 данного руководства "Конфигурация модуля DSM314" в разделе "Тип двигателя". См. также следующие публикации:

- GFH-001, *"Справочник по Техническим Характеристикам Сервоустройств"*
- GFZ-65142E, *"Справочное Руководство по Двигателям переменного тока  $\alpha$  Series"*

**Таблица 1-1. Некоторые модели серводвигателей  $\alpha$  Series**

Номер $\alpha$ -Модели	Момент вращения нм	Выход кВт	Максим. скорость (об./мин)
$\alpha$ 1	1	0,3	3000
$\alpha$ 2	2	0,4	2000
$\alpha$ 2	2	0,5	3000
$\alpha$ 3	3	0,9	3000
$\alpha$ 6	6	1,0	2000
$\alpha$ 6	6	1,4	3000
$\alpha$ 12	12	2,1	2000
$\alpha$ 12	12	2,8	3000
$\alpha$ 22	22	3,8	2000
$\alpha$ 22	22	4,4	3000
$\alpha$ 30	30	3,3	1200
$\alpha$ 30	30	4,5	2000
$\alpha$ 30	30	4,8	3000
$\alpha$ 40	38	5,9	2000
$\alpha$ 40/Fan	56	7,3	2000

## Раздел 4: Сервоприводы $\beta$ Series (цифровой режим)

Особенности Цифрового Сервопривода  $\beta$  Series (произносится как "бета") GE Fanuc:

- Самая высокая на мировом рынке надежность
- Не большие затраты на техническое обслуживание, отсутствие дрейфа, отсутствие коллекторных щеток
- Все параметры задаются в цифровом виде, не требуется повторная настройка
- При применении абсолютного датчика положения не требуется установка в исходное положение (требуется дополнительно комплект аккумуляторов)
- Возможна дополнительно установка тормоза двигателя
- Датчик положения с высоким разрешением (32 К (Beta) (64К – Beta M) импульсов на оборот)

Сервоприводы GE Fanuc  $\beta$  Series обеспечивают высочайшие эксплуатационные характеристики и надежность работы. Новейшие технологические достижения, такие как высокоскоростные последовательные датчики положения и высокоэффективные Интегрированные Модули Питания (IPMs), еще более расширяют эксплуатационные возможности сервосистемы. Будучи сконструированными, для рынка управления движением, Сервоприводы  $\beta$  Series идеально подходят для задач упаковки, погрузочно-разгрузочные операций, обработки материалов и для машиностроения.

Сервосистема GE Fanuc является уникальной в том, что все контуры управления – по току, скорости и позиции – собраны в контроллере движения. Такой подход уменьшает время на установку и дает значительный выигрыш в производительности даже в самых трудных приложениях.

Эти сервоприводы требуют меньше расходов на их встраивание и техническое обслуживание. Цепи управления не подвержены температурным воздействиям. Специализированные модули-приставки отсутствуют. Эти сервоприводы имеют широкую область применения, включая широкий диапазон инерционных нагрузок, настраиваемые ускорения и торможения и конфигурируемые обратные связи по положению. Для оптимизации работы и преодоления машинных ограничений имеется значительный программный инструментарий настройки для пользователя.

### Цифровые усилители $\beta$ Series

Сервоусилитель  $\beta$  Series имеет встроенный источник питания с коммутационными схемами. Благодаря этому GE Fanuc может предоставить компактный усилитель, который на 60% меньше распространенных моделей. Фактически усилитель  $\beta$  Series имеет ту же высоту и толщину, что и модуль ПЛК GE Fanuc Series 90-30. Благодаря этому удается эффективно заполнять панель при использовании контроллера движения DSM314.

Усилитель отвечает требованиям международных стандартов.

GE Fanuc предлагает три интерфейса связи для усилителей  $\beta$  Series: широтно-импульсная модуляция (PWM), Последовательная Шина Сервопривода Fanuc (FSSB) и Интерфейс Связи входа / выхода. С модулем DSM314 может быть использован только интерфейс модулирования ширины импульсов (PWM). Интерфейс PWM использует стандартный протокол связи сервоприводов GE Fanuc. Обратная связь по положению обеспечивается последовательно между контроллером DSM и встроенным в двигатель последовательным датчиком положения.

## Серводвигатели $\beta$ Series

Серводвигатели  $\beta$  Series основаны на первоклассной технологии сервоприводов  $\alpha$  Series. Они содержат несколько конструктивных новшеств, которые обеспечивают улучшенное сочетание высоких эксплуатационных характеристик, низкой стоимости и компактности. Предлагаются номинальные значения до 0,5 до 12 нм.

Эти двигатели на 15% короче и легче, чем аналогичные сервомеханизмы. Новая изоляция в обмотках и общее изоляционное покрытие защищает двигатель от окружающей среды.

Двигатели  $\beta$  Series отвечают международным стандартам (IEC). Класс защиты двигателя - IP65 (по спецзаказу может быть IP67).

В каждом серводвигателе  $\beta$  Series стандартно имеется абсолютный датчик положения (32К – Beta) (64 К – Beta M). Дополнительно каждая модель может быть снабжена удерживающим тормозом 90 В пост. тока.

Более детальную информацию см. в главе 4 данного руководства "Конфигурация DSM314" в разделе "Тип двигателя". См. также следующие публикации:

- GFH-001, *"Справочник по Техническим Характеристикам Сервоустройств"*
- GFZ-65232E, *"Справочное Руководство по Двигателям переменного тока  $\beta$  Series"*

**Таблица 1-2. Некоторые модели серводвигателей  $\beta$  Series**

Номер $\alpha$ -Модели	Момент вращения <sup>1</sup> нм	Выход кВт	Максим. скорость (об./мин)
$\beta$ 0.5	0.5	0.2	3000
$\beta$ M0.5	0.65	0.2	5000
$\beta$ 1	1	0.3	3000
$\beta$ M1	1.2	0.4	5000
$\beta$ 2	2	0.5	3000
$\beta$ 3	3	0.5	3000
$\beta$ 6	6	0.9	2000
$\alpha$ C12	12	1.4	2000

<sup>1</sup> Означает непрерывный режим работы, 100% рабочий цикл

**Примечание:** В списке  $\beta$ -двигателей находится двигатель  $\alpha$ C12 из-за одинаковых характеристик и серии усилителя.

---

## *Раздел 5: Сервоприводы SL-Series (аналоговый режим скорости)*

Модуль DSM314 поддерживает все модели сервоприводов GE Fanuc SL-Series. Детальное описание сервоусилителей SL-Series, двигателей и принадлежностей см. в "*Руководстве Пользователя по Сервоприводам SL-Series*", GFK-1581.



Данная глава содержит семь разделов, в которых кратко описана процедура установки системы Motion Mate DSM314 и проверки ее связей и функций.

1. Распаковка Системы ..... стр. 2-3
2. Сборка Системы ..... стр. 2-3
3. Включение Системы ..... стр. 2-23
4. Конфигурация Motion Mate DSM314 (вводное руководство)..... стр. 2-24
5. Тестирование Системы ..... стр. 2-39
6. Поиск и устранение неисправностей Системы Управления Движением ..... стр. 2-42
7. Дальнейшие действия ..... стр. 2-44

Данная глава является вводной для тех лиц, которые не знакомы с системой управления движением Motion Mate DSM314. Выполняя приведенные здесь действия, оператор может быстро начать работу или привести в движение контроллер движения и сервопривод. Требуется минимальный уровень знаний по одному из указанных ниже пакетов программ конфигурации / программирования.

- Программа Разработки Логики SIMPLICITY, Machine Edition, - ПЛК-версия 2.1 или более поздняя.
- VersaPro, версия 1.1 или более поздняя.

### Предупреждение

**Для проведения данного обучения не разрешается подсоединять механические устройства к валу двигателя. Серводвигатель должен быть надежно закреплен на неподвижной поверхности.**

Типичная система управления движением DSM314 включает в себя контроллер движения DSM314, Программируемый Логический Контроллер (ПЛК) Series 90-30, двигатель (двигатели), сервоусилитель (сервоусилители), Входы / Выходы (I/O) и Человеко-Машинный Интерфейс (HMI).

Система управления DSM314 состоит из двух частей: сервоуправление и управление станком.

**Сервоуправление** преобразует команды движения в сигналы, которые передаются на сервоусилитель. И запускает на выполнение программы Local Logic и Движения. Сервоусилитель получает сигналы управления с сервоуправления, усиливает их до уровня, определяемого мощностью двигателя. Сервоуправление выполняется модулем DSM314.

**Управление станком** (ПЛК Series 90-30) осуществляется при помощи модуля DSM314 и модулей Ввода / Вывода (I/O). Управление станком реализует заданную пользователем логику управления (но не Local Logic). Подсистемы управления станком (ПЛК) и сервоуправления (DSM314) обмениваются данными через системную плату Series 90-30.

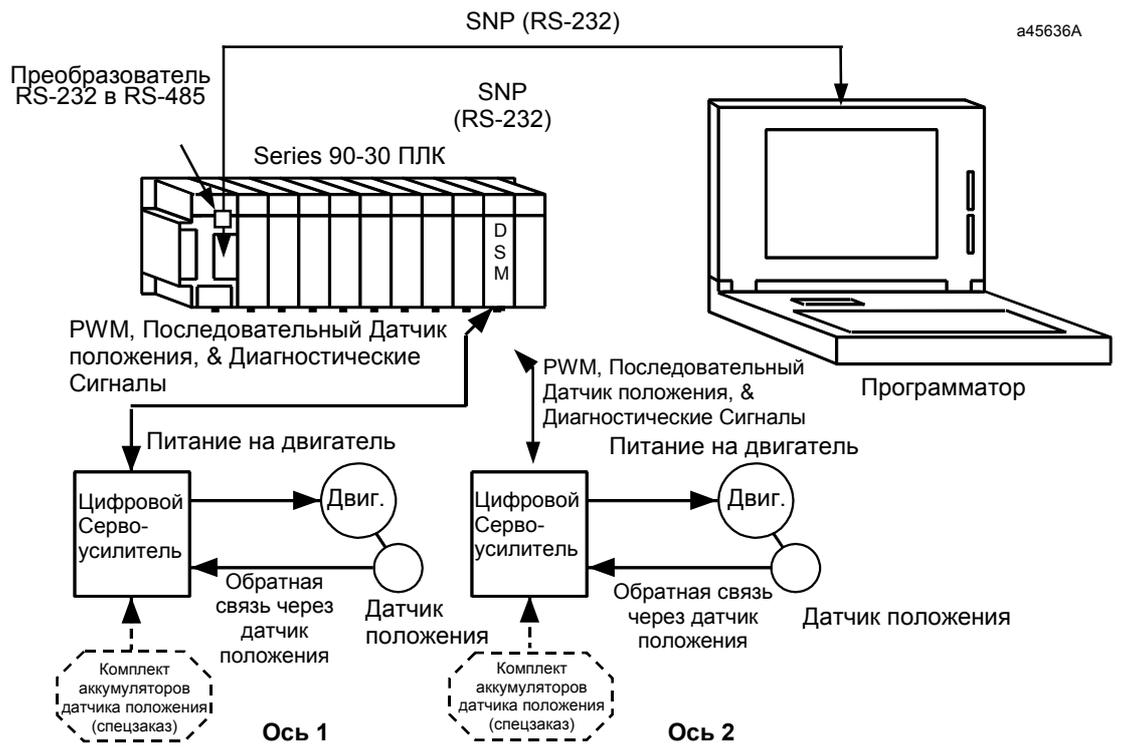


Рис. 2-1. Типовая Цифровая Система Управления Движением по Двум Осям Motion Mate DSM314

## Раздел 1: Распаковка системы

Модуль DSM314, Цифровые Сервоусилители и Двигатели поставляются упакованными по отдельности. Данная глава описывает процедуру распаковки оборудования и предварительной проверки его компонентов.

### Распаковка DSM314

Осторожно распакуйте компоненты системы DSM314 и ПЛК. Убедитесь, что имеются все компоненты, указанные в накладной. Сохраните все сопроводительные и транспортные документы, поставляемые вместе с системой управления движением DSM314.

### Распаковка Цифрового Сервоусилителя

Для использования с DSM314 имеются два упаковочных места подсистемы цифрового усилителя и подсистемы сервоуправления  $\alpha$  Series или  $\beta$  Series.

Цифровой сервоусилитель поставляется в ящике с двойными стенками. Снимите верхний слой упаковочного материала, чтобы получить доступ к усилителю. Затем осторожно извлеките внутренний ящик из внешней упаковки. Извлеките усилитель из внутреннего ящика. Сохраните все незакрепленные детали или прокладочные материалы, упакованные вместе с усилителем. Выполните визуальную проверку усилителя на предмет повреждений при транспортировке.

**Примечание:** Не пытайтесь в это время изменить каким-либо образом имеющееся положение перемычек или переключателей на усилителе.

### Распаковка Двигателя FANUC

Двигатели FANUC упаковываются двумя различными способами в зависимости от их габаритов. Крупногабаритные двигатели поставляются на деревянных поддонах в картонной упаковке. Однако большинство двигателей упаковано в картонные ящики.

1. Инструкции по распаковке:

- У двигателя FANUC, упакованного в ящик, откройте ящик сверху. Двигатель находится в двух держателях по форме двигателя. Осторожно извлеките верхний держатель из ящика. В результате появится достаточно места, чтобы извлечь сам двигатель.
- Если двигатель закреплен на поддоне, снимите картонную упаковку. Это даст доступ к болтам, удерживающим двигатель на поддоне. Удалите болты, чтобы снять двигатель с поддона.

2. Проверьте двигатель на предмет повреждений.

3. Убедитесь, что вал двигателя проворачивается рукой.

**Примечание:** Если был заказан двигатель с дополнительным удерживающим тормозом, то вал не будет вращаться, пока не подано питание на тормоз.

Следующий шаг

**Сборка Системы Motion Mate DSM314**

## Раздел 2: Сборка Системы Motion Mate DSM314

### Общие Указания

- Всегда проверяйте надежность подключения электрических разъемов. Разъемы выполнены так, что они могут быть установлены только одним единственным образом. Не прикладывайте к ним значительных усилий.
- Обращайте внимание на правильное заземление компонентов системы DSM314, в т.ч. и экранирующего заземляющего провода лицевой панели DSM314. Информация по заземлению содержится в данном разделе.

Все пользовательские подсоединения (за исключением вывода заземления) находятся на передней части модуля DSM314. Вывод заземления расположен на нижней части модуля. См. рис. ниже.

Инструкции по установке DSM314 с соблюдением IEC и других стандартов см. в "Рекомендации по Установке с Соблюдением Стандартов", GFK-1179.

### Соединения Motion Mate DSM314

На рис. 2-2 представлено упрощенное изображение лицевой панели и надписей на модуле DSM314. Дополнительная информация и полная схема соединений находятся в главе 3 "Установка и Разводка Motion Mate DSM314".

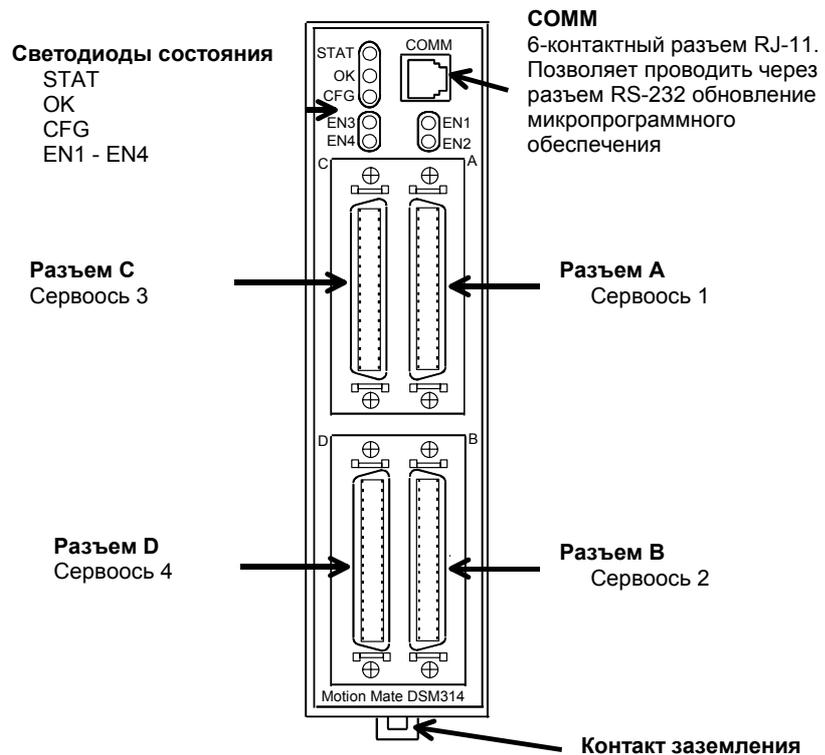


Рис. 2-2. Разъемы, расположенные на Лицевой Панели Системы Управления Движением Motion Mate DSM314

## Подключение Цифрового Сервоусилителя SVU $\alpha$ Series

*Если должен быть подключен усилитель  $\beta$  Series, перейдите к следующему разделу.*

Цифровой Сервоусилитель  $\alpha$  Series не требует дополнительной настройки при первом запуске или при замене компонентов. Ему также не требуется подстройка при изменении условий окружающей среды.

Для подключения Цифрового Сервоусилителя  $\alpha$  Series выполните перечисленные ниже действия.

### 1. Подключение Сервоусилителя $\alpha$ Series к DSM314.

- A. Перед подключением кабеля сервокоманд убедитесь, что экранирующий заземляющий провод лицевой панели DSM314 подсоединен. Этот провод поставляется вместе с модулем DSM314, и он должен быть закреплен между клеммой 1/4" на нижней части модуля и соответствующей точкой заземления на панели.
- B. Кабель сервокоманд передает выходной сигнал широтно-импульсной модуляции (PWM) от DSM, последовательные данные от датчика положения двигателя и диагностические сигналы от усилителя. Сигналы, передаваемые по этому кабелю, имеют уровень напряжения, предназначенный для передачи данных; кабель должен располагаться вдали от других кабелей, особенно от силовых.
- C. Расположите кабель сервокоманд **IC800CBL001** (1 метр) или **IC800CBL002** (3 метра). Вставьте разъем этого кабеля в разъем **JS1B** в нижней части Сервоусилителя (см. рис. 2-4).
- D. Если щиток разъемов оси IC693ACC335 не используется для отключения пользовательских Вводов / Выводов, таких как пере регулирование или датчика начальной позиции, то вставьте другой конец кабеля в разъем, обозначенный как **A** для оси 1 или как **B** для оси 2, на передней части DSM314. Если щиток разъемов используется, то вставьте другой конец кабеля в разъем оконечного устройства, помеченного как **SERVO**. Затем расположите кабель подключения щитка разъемов **IC693CBL324** (1 метр) или **IC693CBL325** (3 метра). Вставьте один конец этого кабеля в разъем щитка, помеченного как **DSM**. А другой конец кабеля в разъем, обозначенный как **A** для сервооси 1 или как **B** для сервооси 2, на передней части модуля DSM314.

**Примечание:** Информация о пользовательских I/O, доступных для подключений оконечного устройства IC693ACC335, находится в разделе "Подключения Вводов / Выводов (I/O)" в главе 3.

## Переключатели Каналов Усилителя SVU

Убедитесь, что Переключатели Каналов (DIP-переключатели), расположенные за дверцей усилителя, установлены так, как показано в таблицах 2-1. Учтите, что положение **OFF (ВЫКЛ.)** - слева, а положение **ON (ВКЛ.)** - справа. Учтите также, что нумерация переключателей идет снизу вверх (Переключатель 1 – самый нижний). Например, на рис. 2-3 Переключатели 1, 3 и 4 находятся в положении ON, а переключатель 2 – в OFF.

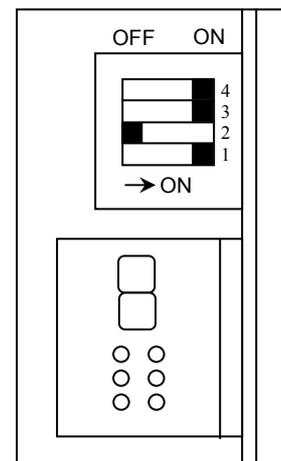


Рис. 2-3. Переключатели Каналов Усилителя SVU

Таблица 2-1. Установки Переключателей Каналов Усилителя SVU

Усилитель SVU1-80				
Регенеративный (тормозной) Блок	SW1	SW2	SW3	SW4
Встроенный (100 Вт)	ВКЛ.	ВЫКЛ.	ВКЛ.	ВКЛ.
Отдельный А06В-6089-Н500 (200 Вт)	ВКЛ.	ВЫКЛ.	ВКЛ.	ВЫКЛ.
Отдельный А06В-6089-Н713 (800 Вт)	ВКЛ.	ВЫКЛ.	ВЫКЛ.	ВЫКЛ.

Усилитель SVU1-130				
Регенеративный (тормозной) Блок	SW1	SW2	SW3	SW4
Встроенный (400 Вт)	ВКЛ.	ВЫКЛ.	ВКЛ.	ВКЛ.
Отдельный А06В-6089-Н711 (800 Вт)	ВКЛ.	ВЫКЛ.	ВКЛ.	ВЫКЛ.

(Чтобы подключить дополнительные усилители, повторите приведенные выше шаги В, С и D для каждого дополнительного усилителя).

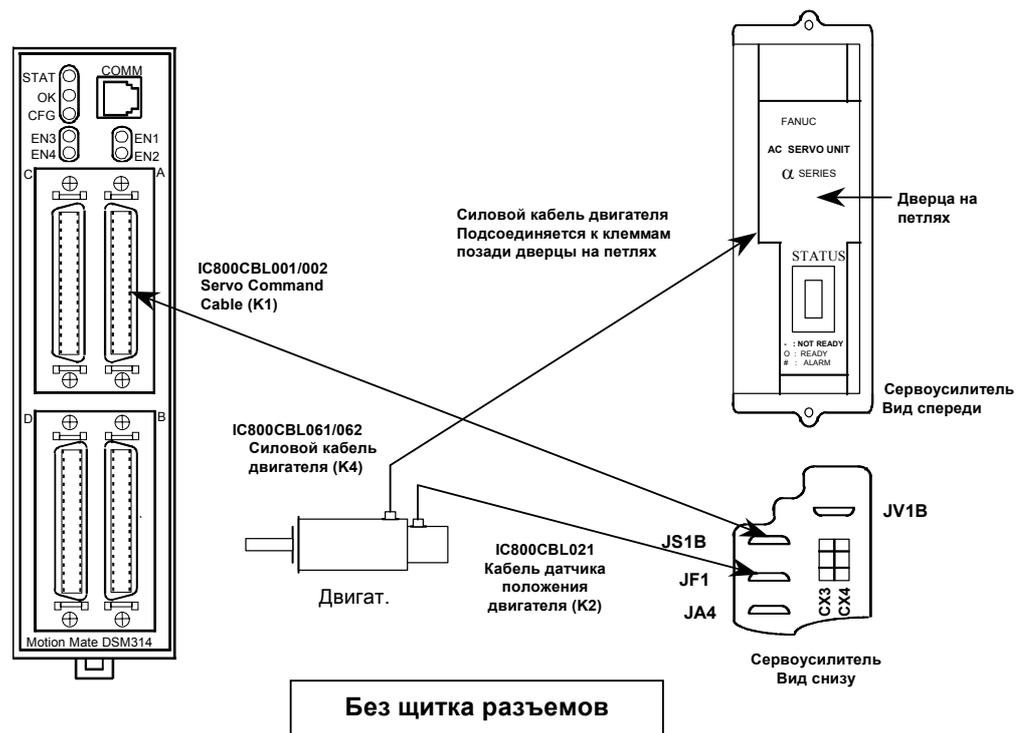
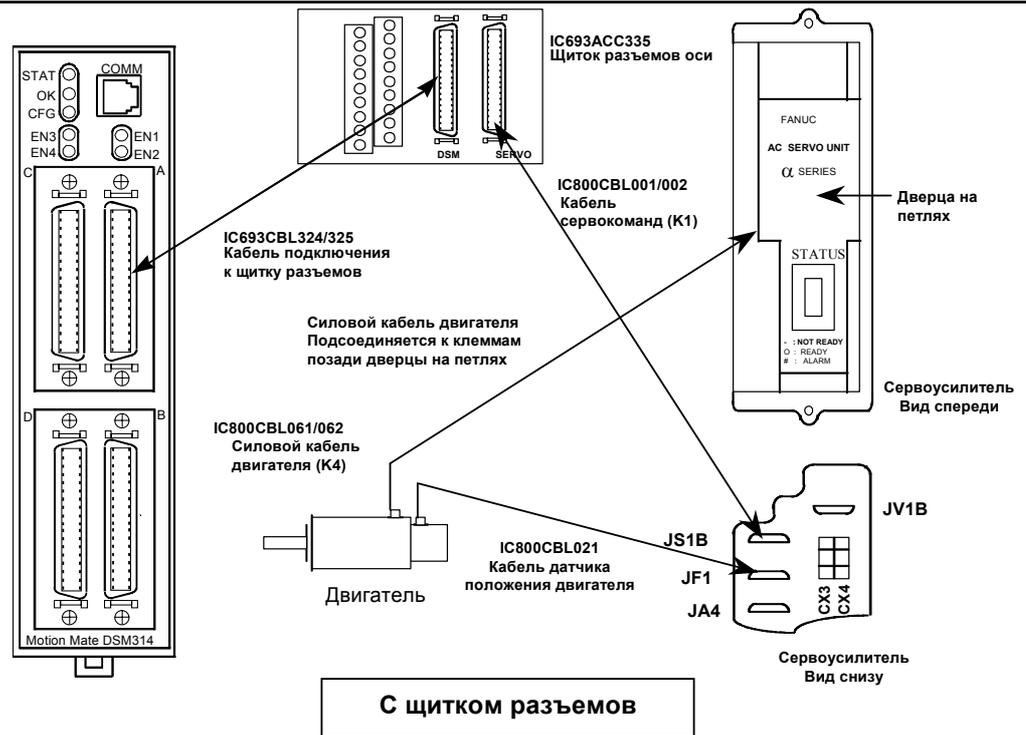


Рис. 2-4. Подключение Цифрового Сервоусилителя  $\alpha$  Series к модулю Motion Mate DSM314

## 2. Подключение Силовой Кабеля Двигателя к Цифровому Сервоусилителю $\alpha$ Series

- A. Размер двигателя для системы определяет тип и размер силового кабеля K4 двигателя, если вместе с системой заказываются кабели заводского изготовления.

Двигатели в приведенной ниже таблице сгруппированы так, чтобы использовать один из кабелей заводского изготовления, которые могут быть заказаны через GE Fanuc. В таблице приведен список не всех силовых кабелей для серводвигателей  $\alpha$  Series, а лишь широко используемых. Полный список находится в "Справочнике по техническим характеристикам Сервоустройств", GFH-001.

**Таблица 2-2. Примеры Номеров Заказа Силового Кабеля Заводского изготовления (K4) для  $\alpha$ -Серводвигателя**

Тип Двигателя	Номера в Каталоге для Кабеля, работающего в Тяжелом Режиме	Описание Кабеля	Длина Кабеля
$\alpha 3/3000$ $\alpha 6/3000$	IC800CBL061	Изогнутый соединительный элемент MS	14 метров
$\alpha 12/3000$ $\alpha 22/2000$ $\alpha 30/1200$	IC800CBL062	Изогнутый соединительный элемент MS	14 метров
$\alpha 30/3000$ $\alpha 40/2000$	IC800CBL063	Изогнутый соединительный элемент MS	14 метров

- В. С одной стороны этого кабеля имеются четыре провода, обозначенные как U, V, W и GND, которые подсоединяются к зажимным контактам 9 – 12 на сервоусилителе. Подсоедините эти четыре провода к клеммной колодке, как показано на рис. 2-5.
- С. Снимите пластиковую крышку, защищающую разъем на двигателе, и подсоедините к нему другой конец кабеля. Обратите внимание на то, что этот кабель имеет соответствующий ключ и может быть установлен на одну из точек подключения двигателя только одним единственным способом.

(Повторите эту процедуру, если требуется, для любой другой оси системы).

Для получения самой последней информации по силовым кабелям двигателей или по изготовлению силовых кабелей по спецзаказу см. последнюю версию "Справочного Руководства по Серводвигателям  $\alpha$  Series", GFZ-65142E.

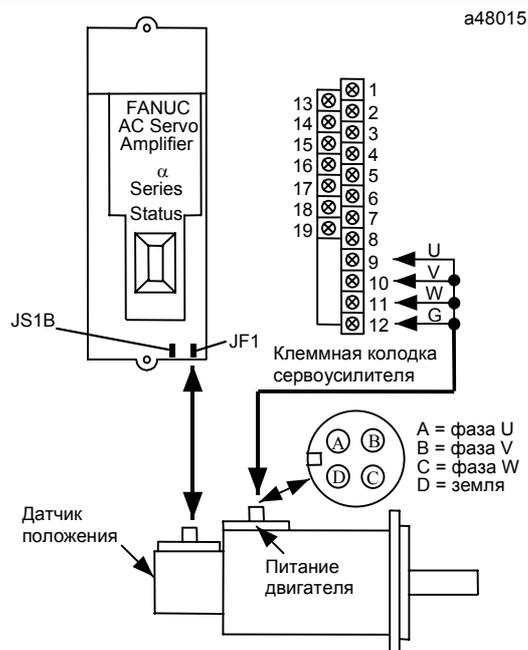


Рис. 2-5. Подключение Двигателя к Контактной Колодке Сервоусилителя  $\alpha$  Series

### 3. Подключение Датчика Положения Двигателя к Цифровому Сервоусилителю $\alpha$ Series

- А. Снимите защитную пластиковую крышку с разъема датчика положения на двигателе и подключите кабель обратной связи K2 IC800CBL021. Кабель выполнен так, что он может быть подсоединен к двигателю только в одном месте
- В. Вставьте противоположный конец кабеля в гнездо, обозначенное как **JF1**, в нижней части сервоусилителя  $\alpha$  Series (см. рис. 2-6)

Повторите эту процедуру для всех осей в системе.

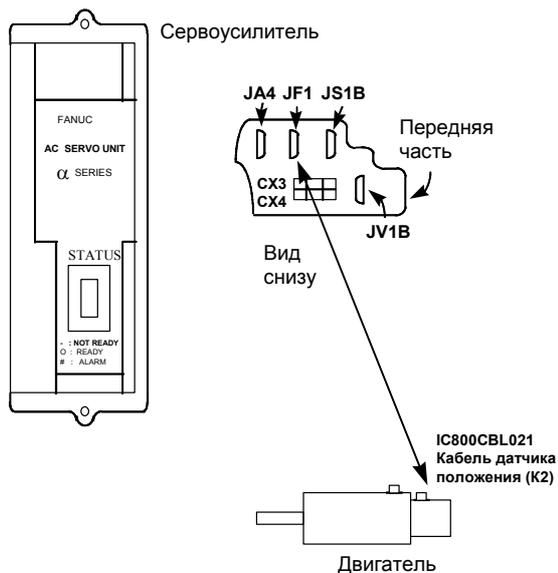


Рис. 2-6. Подключение Датчика положения Двигателя α Series

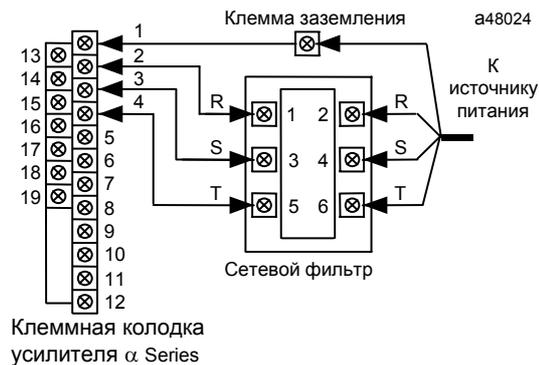
Таблица 2-3. Кабель (K2) Датчика положения Серводвигателя α для Моделей от α3 до α40

Модель Двигателя	Кабель для Работы в Тяжелом режиме	Длина Кабеля
От α3 до α40	IC800CBL021	14 метров

**Примечание:** Подробности о кабелях серии α имеются в "Справочном Руководстве для Серводвигателей переменного тока α Series", GFZ-65142E, и в "Справочнике по Техническим Характеристикам Сервоустройств α Series и β Series" GFH-001.

#### 4. Подключение 3-фазного электропитания 220 В к Цифровому Усилителю $\alpha$ Series

Фильтр сети переменного тока снижает влияние гармонических помех на источник питания; рекомендуется использовать такой фильтр. На один фильтр сети переменного тока может быть установлено два или более усилителей, если не превышает его допустимая нагрузка по мощности. На рис. 2-6 показано подключение усилителя к такому сетевому фильтру.



**Примечание:** Требуется трехфазная линия ~ 220 В.

Рис. 2-7. Подключение Сервоусилителя к Сетевому Фильтру и к Источнику Питания

**Примечание:** Заказчик самостоятельно обеспечивает кабель для обоих подключений: сетевого фильтра к сервоусилителю и сетевого фильтра к источнику питания. Для линии между сетевым фильтром и сервоусилителем следует использовать сертифицированный 4-жильный кабель UL или CSA на 600 В, 60°C (140°F).

Сечение проводников, используемых для соединения сетевого фильтра с источником питания, выбирается, исходя из тока отключения между источником питания и сетевым фильтром и из количества сервоустройств, подключаемых к сетевому фильтру.

Если для питания усилителей переменным током используется отдельный развязывающий трансформатор, то сетевой фильтр не требуется.

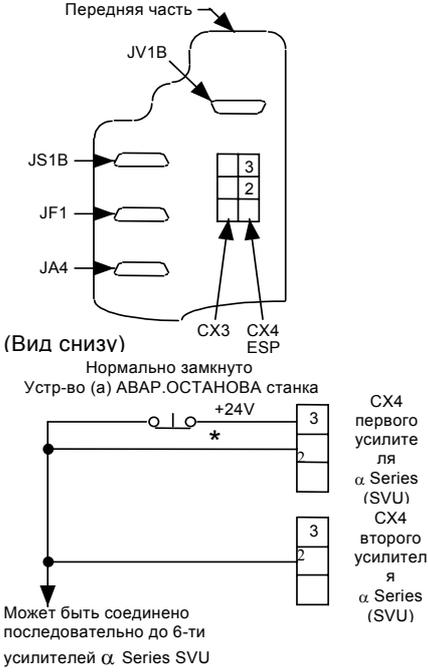
#### 5. Подключение Аварийного Останов Станка к Цифровому Сервоусилителю $\alpha$ Series

Контакт 3 разъема CX4, расположенного в нижней части усилителя  $\alpha$  Series (SVU), выдает +24 В постоянного тока на цепь E-STOP (АВАРИЙНЫЙ ОСТАНОВ). Соберите схему АВАРИЙНОГО ОСТАНОВА таким образом, что бы на контакте 2 было +24 В постоянного тока (=) **при отсутствии** АВАРИЙНОГО ОСТАНОВА. Если выключатель АВАРИЙНЫЙ ОСТАНОВ не используется, то такое соединение **должно** быть выполнено перемычкой

**Примечание:** Кабель для этого соединения обеспечивает Заказчик. В комплект усилителя входят снабженные ключами штекерные гнезда, обозначенные как X, и соединительные штекеры. Чтобы усилитель работал, они должны быть установлены в виде переключателя или перемычки.

**Предостережение**

Запрещается подавать на этот разъем внешнее напряжение.



**Рис. 2-8. Подключение Аварийного Останова к Сервоусилителю  $\alpha$  Series**

Дополнительную информацию см. в "Справочном руководстве по сервоусилителям (SVU)  $\alpha$  Series", GFZ-65192EN.

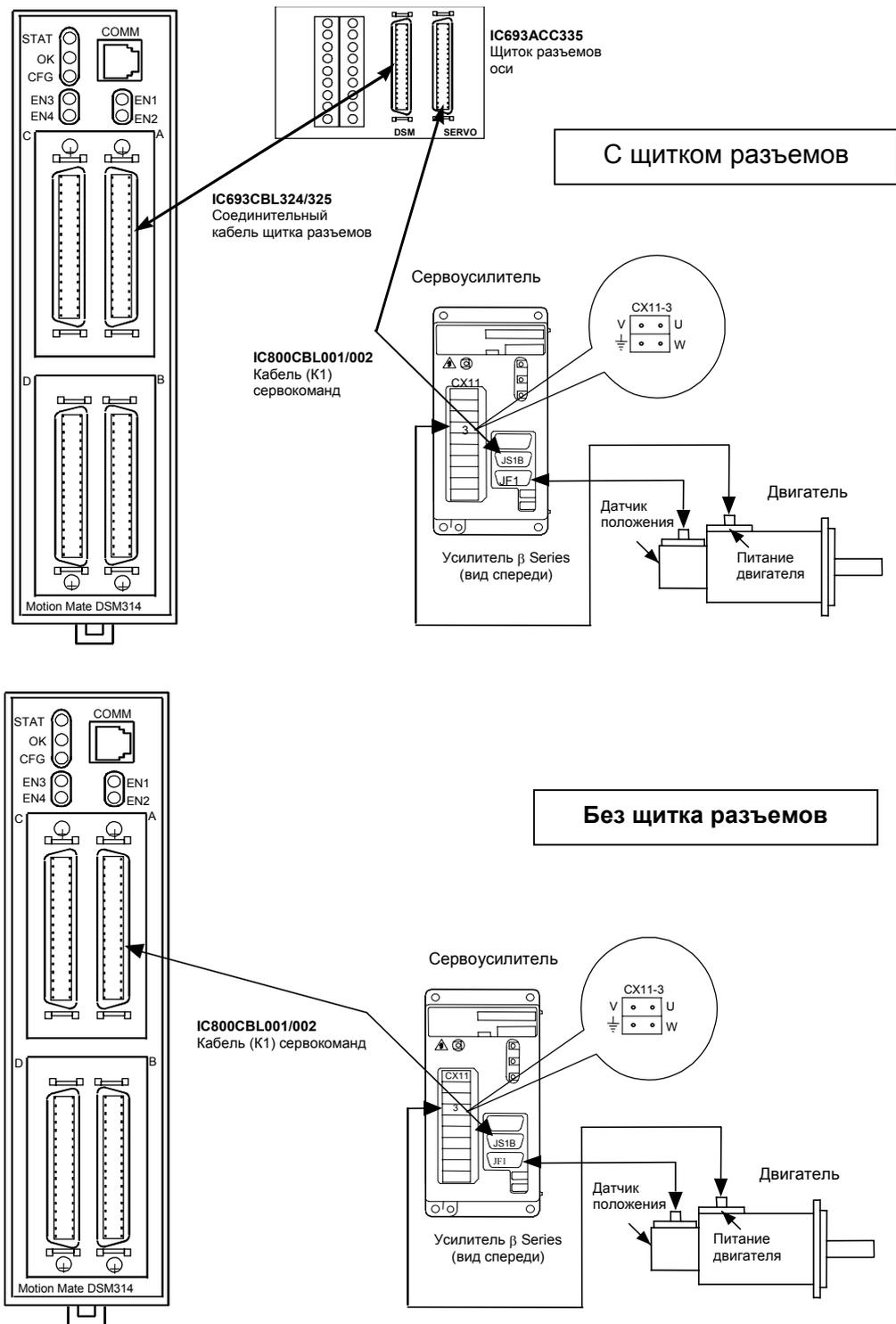
## Подключение Цифрового Сервоусилителя SVU $\beta$ Series

Цифровой Сервоусилитель  $\beta$  Series не требует какой-либо пользовательской регулировки. Для подключения Цифрового Сервоусилителя  $\beta$  Series выполните перечисленные ниже действия. *Информацию по усилителям  $\alpha$  Series см. в предыдущем разделе.*

### 1. Подключение Цифрового Сервоусилителя $\beta$ Series к DSM314

- A. Перед подключение кабеля сервокоманд убедитесь, что экранирующий заземляющий провод лицевой панели DSM314 подсоединен. Этот провод поставляется вместе с модулем DSM314, и он должен быть закреплен между клеммой 1/4" на нижней части модуля и соответствующей точкой заземления на панели.
- B. Кабель сервокоманд передает выходной сигнал широко-импульсной модуляции (PWM) от контроллера движения, последовательные данные от датчика положения двигателя и диагностические сигналы от усилителя. Сигналы, передаваемые по этому кабелю, имеют уровень напряжения, предназначенный для передачи данных; кабель должен быть отнесен от других кабелей, особенно от силовых.
- C. Установите кабель сервокоманд **IC800CBL001** (1 метр) или **IC800CBL002** (3 метра). Установите разъем этого кабеля в разъем **JS1B** на передней части Сервоусилителя (см. рис. 2-9)
- D. Этот шаг зависит от того, используется ли щиток разъемов или нет.
  - Если щиток разъемов **IC693ACC335** оси не используется для отключения пользовательских Вводов / Выводов, таких как пере регулирование или ограничения возврата в начальную позицию, то вставьте другой конец кабеля в разъем, обозначенный как **A** для сервооси 1 или как **B** для сервооси 2, на передней части DSM314.
  - Если щиток разъемов **IC693ACC335** оси используется, то вставьте другой конец кабеля в разъем этого щитка, обозначенного как **SERVO**. Затем расположите кабель сервокоманд **IC693CBL324** (1 метр) или **IC693CBL325** (3 метра). Вставьте один конец этого кабеля в разъем щитка, обозначенного как **DSM**. Вставьте другой конец кабеля в разъем, обозначенный как **A** для сервооси 1 или как **B** для сервооси 2, на передней части модуля DSM314.

Чтобы подключить дополнительные усилители, повторите приведенные выше шаги B - D для каждого дополнительного усилителя.



**Рис. 2-9. Подключение Сервоусилителя  $\beta$  Series**

Дополнительную информацию см. в разделе о подключениях "Справочника по Техническим Характеристикам Сервоустройств", GFH-001.

## 2. Подключение Силового Кабеля Двигателя (K4) к Цифровому Сервоусилителю $\beta$ Series

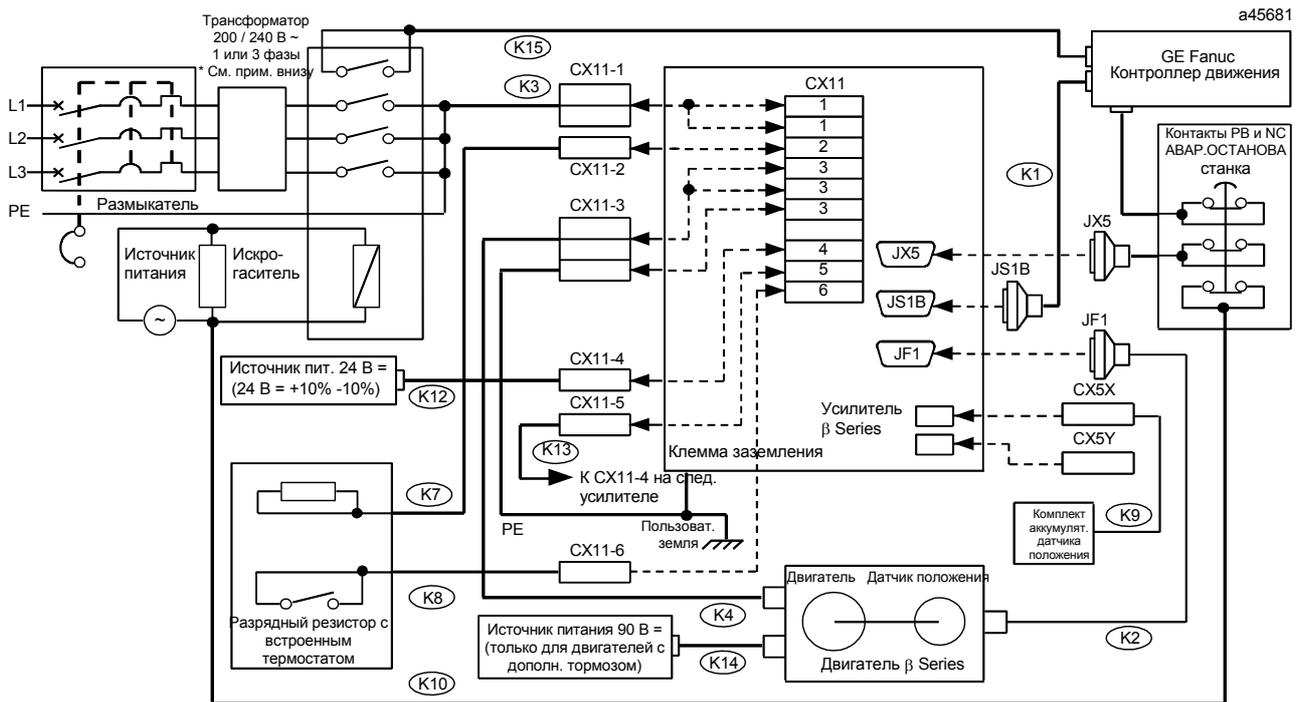
### Предостережение

Подключение разъема CX-11 следует выполнять осторожно. Этот разъем не имеет ключа. Проверьте выполненное соединение несколько раз, прежде чем подавать питание. Неправильное соединение может привести к неправильной работе оборудования или к его повреждению. Модификации усилителя более поздние, чем G, имеют разъемы, защищенные ключами.

- A. Размер двигателя для системы определяет тип и размер силового кабеля (K4), необходимого для двигателя. Можно приобрести кабели заводского изготовления или изготовить кабели самостоятельно. Информацию по соответствию изготовленных самостоятельно кабелей и их установки знаку CE см. в "*Справочном Руководстве по Управляющим Двигателям  $\beta$  Series*", GFZ-65232EN. Конец силового кабеля (для двигателя) заводского изготовления, подключаемый к усилителю, выполнен так, чтобы его можно было подключить к разъему CX11-3 на усилителе.

Таблица 2-4. Кабель K4 - Примеры Кабелей Двигателей  $\beta$  Series

Тип Серводвигателя	Номер Заказа Кабеля K-4 Двигателя	Описание Кабеля
$\beta$ 0.5/3000	IC800CBL067	14 метров
$\beta$ 1/3000, $\beta$ 2/3000, $\beta$ 3/3000 и $\beta$ 6/2000	IC800CBL068	14 метров
$\alpha$ C12/2000	IC800CBL069	14 метров
$\beta$ M 0.5/5000	CP8B-OWPB-0140-AA	14 метров
$\beta$ M 1/5000	CP8B-OWPB-0140-AA	14 метров



\* **Примечания:**

1. Сетевой фильтр и защита от атмосферных разрядов могут использоваться вместо трансформатора, если к распределительному шкафу подведено напряжение 200-240В.
2. При работе с одной фазой линия L3 не подсоединяется. Информацию по снижению выходного номинального тока см. в *Технических Характеристиках Сервосистемы* в "Справочнике по Техническим Характеристикам Сервоустройств", GFH-001.

**Рис. 2-10. Подключение к Контактной Колодке Цифрового Сервоусилителя beta Series**

- В. Снимите пластиковую крышку, защищающую разъем на двигателе, и подсоедините к нему конец силового кабеля двигателя. Обратите внимание на то, что этот кабель имеет соответствующий ключ и может быть установлен на одну из точек подключения двигателя только одним единственным способом.
- С. В поставку силовых кабелей двигателей, приобретаемых в GE Fanuc, входит однометровый одножильный проводник с разъемом CX11-3 на одном конце и кольцевым наконечником – с другой. Этот кабель обеспечивает заземляющие подключения для корпуса двигателя, и он должен быть постоянно подсоединен. Изготовители кабелей для Заказчика должны всегда включать в поставку этот кабель. Для правильного подключения к усилителю см. схему соединений выше.

(Повторите эту процедуру, если требуется, для другой оси системы).

Дополнительную информацию см. в "Справочнике по Техническим Характеристикам Сервоустройств", GFH-001.

### 3. Подключение Кабеля Датчика Положения (K2) Двигателя к Цифровому Сервоусилителю beta Series

Размер двигателя для системы определяет тип и размер силового кабеля K4 двигателя, если вместе с системой заказываются кабели заводского изготовления. Пожалуйста, обратитесь к приведенной ниже таблице, для определения правильного каталожного номера кабеля датчика положения,

- А. Снимите защитную пластиковую крышку с разъема на двигателе и подключите кабель датчика положения K2 (см. табл. 2-5). Этот кабель имеет два различных разъема.
- В. Вставьте один конец кабеля с разъемом D-типа в гнездо, обозначенное как **JF1**, на сервоусилителе (см. рис. 2-9).

С. Другой разъем кабеля выполнен так, что он может быть вставлен только в одно гнездо датчика положения на двигателе (крышка с красным верхом).

(Повторите эту процедуру для всех осей в системе).

Таблица 2-5. К2 Кабель – Примеры кабелей β Series Датчика положения

Тип Двигателя	Номер Заказа Кабеля К2 Датчика положения	Описание Кабеля
β 0.5/3000	IC800CBL022	14 метров
β1/3000, β 2/3000, β3/3000 и β6/2000	IC800CBL023	14 метров
α C12/2000	IC800CBL021	14 метров
βM 0.5/5000	CFBA-OWPB-0140-AA	14 метров
βM 1/5000	CFBA-OWPB-0140-AA	14 метров

#### 4. Подключение Силового Кабеля ~ 220 В (К3) к Цифровому Усилителю β Series

Кабель питания от сети переменного тока поставляет Заказчик; кабель подсоединяется к **CX11-1** на передней части усилителя β Series. Разъем для подключения кабеля на стороне усилителя является частью комплекта A06B-6093-K305, поставляемого вместе с каждым усилителем. Более подробную информацию см. в "Справочнике по Техническим Характеристикам Сервоустройств", GFH-001.

Фильтр сети переменного тока снижает влияние гармонического шума на источник питания; рекомендуется использовать такой фильтр. Сетевой фильтр не требуется, если используется изолирующий трансформатор или отдельный силовой трансформатор. На один фильтр сети переменного тока может быть установлено два или более усилителей, если не превышает его допустимая нагрузка по мощности. На рис. 2-11 показано подключение усилителя к такому сетевому фильтру

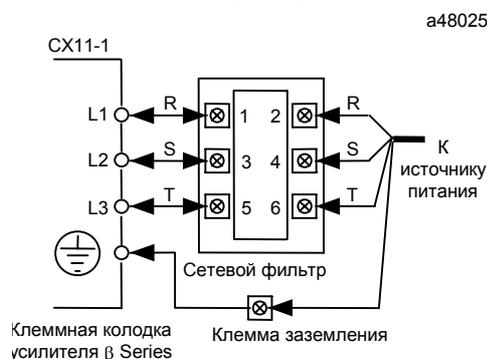


Рис. 2-11. Подключение Сервоусилителя β Series к Сетевому Фильтру и к Источнику Питания

**Примечание:** Заказчик предоставляет кабель для соединения между сетевым фильтром и источником питания. Для линии между сетевым фильтром и сервоусилителем следует использовать сертифицированный 4-жильный кабель UL или CSA на 600 В, 60°C (140°F). Сечение проводников, используемых для соединения сетевого фильтра с источником питания, выбирается, исходя из характеристик тока отключения цепи между источником питания и сетевым фильтром и из количества сервоустройств, подключаемых к сетевому фильтру. Силовые разъемы и клеммные подключения входят в комплект поставки усилителя.

## 5. Подключение Аварийного Останов Станка к Цифровому Сервоусилителю $\beta$ Series

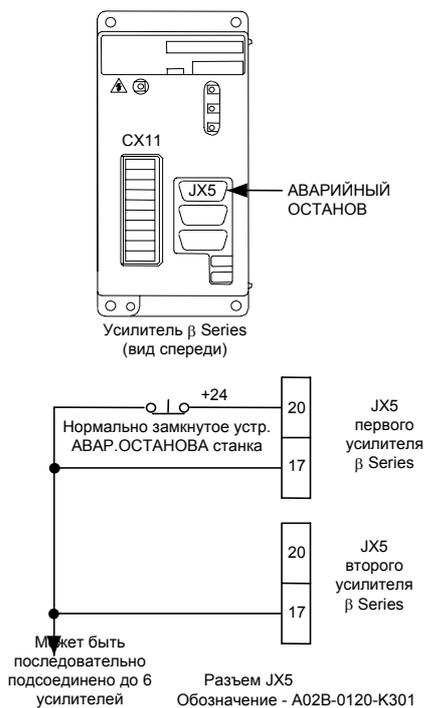


Рис. 2-12. Подключение АВАРИЙНОГО ОСТАНОВА к Сервоусилителю  $\beta$  Series

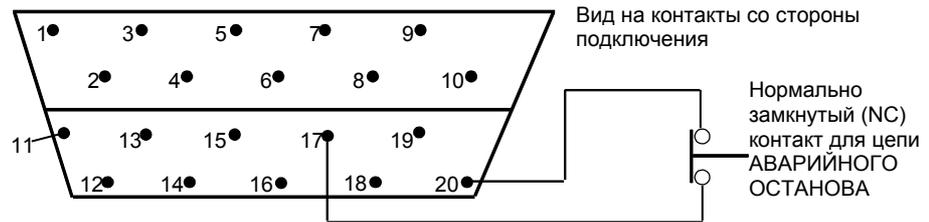
**Примечание:** Кабель для этого соединения обеспечивает Заказчик. Разъем JX5 и крышка разъема входят в поставку усилителя под номером A02B-0120-K301. Если цепь АВАРИЙНОГО ОСТАНОВА не требуется, это соединение должно быть выполнено при помощи перемычки или же усилитель будет заблокирован.

На контакте 20 разъема JX5 имеется +24 В постоянного тока для цепи АВАРИЙНОГО ОСТАНОВА. Это напряжение с контакта 20 проходит через нормально замкнутые контакты или переключатели так, что на контакте 17 JX5 имеется +24 В постоянного тока при отсутствии АВАРИЙНОГО ОСТАНОВА. GE Fanuc использует два класса разъемов для разъема JX5. См. рис. 2-13, чтобы правильно подключить тот или иной тип.

### Предостережение

Запрещается подавать на этот разъем какое-либо внешнее напряжение.

### Расположение Выводов Разъема HIROSE PCR тип на 20 контактов



### Расположение Выводов Разъема HONDA PCR тип на 20 контактов

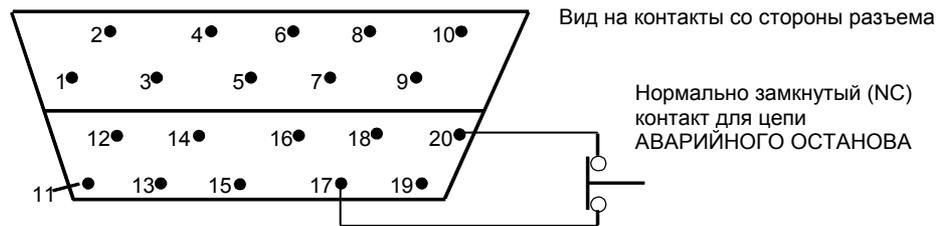


Рис. 2-13. Расположение Выводов Разъема PCR на 20 контактов

## 6. Подключение Кабеля 24 В постоянного тока (K12) к Цифровому Сервоусилителю $\beta$ Series

Разъем для источника внешнего напряжения 24 В постоянного тока входит в комплект усилителя как часть набора A06B-6093-K305; этот разъем подключается к **СХ11-4**. Другой конец кабеля должен быть подсоединен к источнику постоянного тока 24 В, способному обеспечить ток не менее 450 мА для каждого усилителя  $\beta$  Series. Рекомендуется использовать источник питания GE Fanuc IC690PWR024. **Не подавайте напряжение во время работы!**

## 7. Подключение Кабеля K8 – Перемычка или Внешний Регенеративный (тормозной) Резистор к Цифровому Сервоусилителю $\beta$ Series

### Без Внешнего Регенеративного (тормозного) Резистора (Использование Перемычки)

Если внешний регенеративный (тормозной) резистор отсутствует, то соединения на СХ11-2 (DCP и DCC) должны быть разомкнутыми. Но требуется поставить перемычку на клеммы СХ11-6 (ТН1 и ТН2), как показано на рисунке ниже. (Эта перемычка замыкает цепь, которая в противном случае замыкалась бы через нормально замкнутый переключатель теплового реле во внешнем блоке регенеративного (тормозного) резистора). Если эта перемычка не установлена, то усилитель не будет работать. Перемычка и ее разъем входят в состав комплекта разъема A06B-6093-K305, который поставляется в комплекте каждого усилителя.

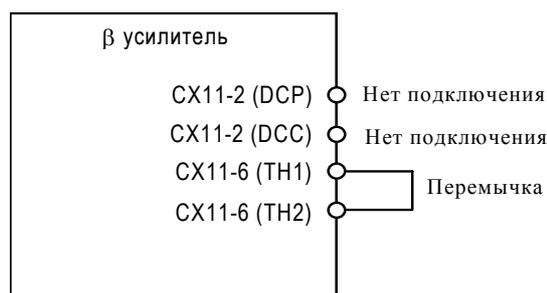


Рис. 2-14. Установка Перемычки, если не используется Внешний Регенеративный Резистор

### При наличии Внешнего Регенеративного (тормозного) Резистора

Если имеется внешний регенеративный (тормозной) резистор, то у него есть 4 провода. Два провода меньшего размера (K8) принадлежат внутреннему нормально замкнутому контакту теплового реле. Этот переключатель размыкается и отключает усилитель, если температура резистора становится слишком высокой. Два провода большего размера (K7) подключаются к резистору. Все соединительные элементы, требуемые для подключения этого резисторного блока к усилителю, имеются в комплекте усилителя.

Подключите два провода теплового реле (K8) к клеммам TH1 и TH2 на CX11-6. (Эти подключения нейтральны к полярности).

Подсоедините два резисторных провода (K7) к CX11-2 – клеммы DCP и DCC. (Эти подключения нейтральны к полярности).

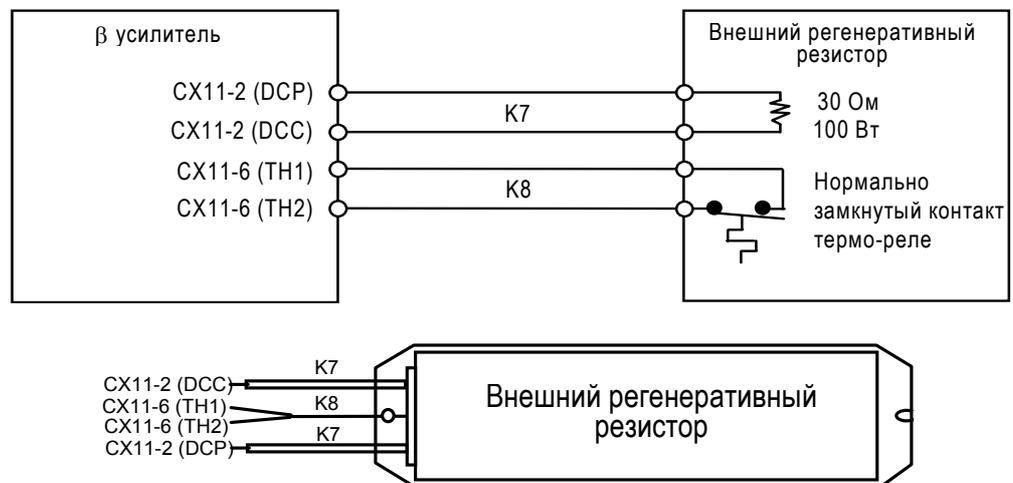


Рис. 2-15. Подключение Внешнего Регенеративного Резистора

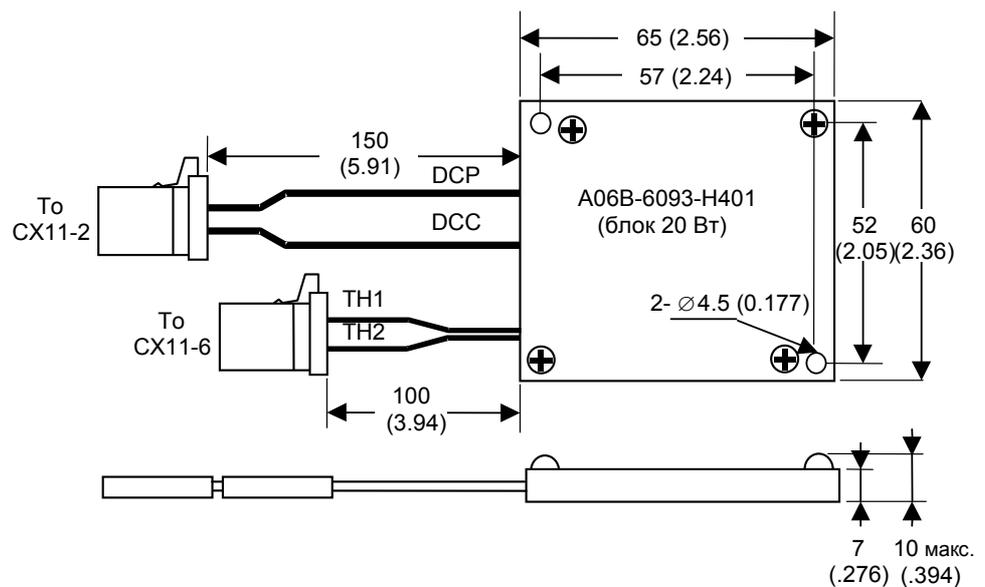


Рис. 2-1. Регенеративный (тормозной) Резистор 20 Вт A06B-6093-H401

## Установка и Подключение DSM314 для Аналогового Режима

Важные замечания по аналоговым сервоустройствам:

1. Выход режима Команды Аналоговой Скорости Сервопривода или Аналогового Моментa вращения являются несимметричными сигналами на контакте 6 Вспомогательной Клеммной Колодки. Этот сигнал определяется по отношению к 0 В на модуле DSM и на ПЛК. Этот сигнал должен быть подключен к входу команды скорости или команды момента вращения сервоусилителя.

**Примечание:** Представляется важным правильно сконфигурировать DSM либо для режима Аналогового Моментa вращения, либо для режима Аналоговой Скорости. Выбор режима зависит от типа используемого сервоусилителя.

2. DSM314 имеет слаботочный (30 мА) выход твердотельного реле на контакте 15 Вспомогательной Клеммной Колодки для подключения к входу сигнала разрешения сервоусилителя.
3. В аналоговом режиме для DSM314 требуется вход Drive Ready (Готов к Пуску) (сигнал IN\_4) на контакте 5 Вспомогательной Клеммной Колодки. **Этот сигнал должен стать равным 0 В при готовности усилителя управлять сервоприводом. DSM начинает проверять вход Drive Ready раз в секунду после того, как реле Drive Enable (Разрешить Пуск) включилось в ответ на бит %Q Enable Drive (Разрешить Пуск). Если сервоусилитель не может обеспечить соответствующий выход, то вход IN\_4 на DSM314 может быть подключен к 0 В или при конфигурации модуля эта функция может быть запрещена. Подробности смотри в Главе 4.**
4. Обратная связь через импульсный датчик положения используется в аналоговом режиме. Подключения разводки датчика положения показаны на рис. 3-19 – 3-23.
5. Рис. 3-19 – 3-23 являются общими схемами аналоговой разводки для DSM.
6. Детальнее о связи DSM314 с SL-сервоустройствами GE Fanuc см. в "*Руководстве Пользователя по Сервоустройствам SL Series*", GFK-1581.

## Заземление Системы Управления Движением Motion Mate DSM314

Система DSM314 должна быть правильно заземлена. Много проблем возникает просто потому, что это требование не соблюдается на практике. Чтобы правильно заземлить систему Motion Mate DSM314, следует выполнять приведенные ниже правила:

- Сопротивление заземления системной "земли" должно быть 100 Ом или менее (заземление класса 3).
- Должен быть закреплен экранирующий заземляющий провод лицевой панели DSM314 (входит в комплект модуля) между клеммой 1/4" на нижней части модуля и соответствующей точкой заземления на панели.
- Если используется щиток разъемов оси, то должны быть обеспечены две точки подключения экрана ("S"), одна из которых должна быть подключена к земле рамы панели.
- Кабель системной земли должен иметь достаточно большое сечение, чтобы быть в состоянии надежно пропустить возможный большой ток на землю, если произойдет, например, короткое замыкание. Обычно минимальное сечение должно быть таким же, как у силового кабеля цепи питания переменного тока. На рис. 2-16 показана система заземления.
- Чтобы соответствовать действующим на месте использования правилам электромонтажа, всегда должно быть выполнено подключение земли усилителя, силовой цепи (PE) и корпуса двигателя. При монтаже в соответствии с предписаниями стандарта СЕ требуется шина заземления и зажим (зажимы) (заказывается отдельно) для перехода от клеммной колодки к кабелю усилителя. Дополнительную информацию см. в разделе "Заземление кабелей Ввода / Вывода" главы 3 "Установка и разводка DSM314".

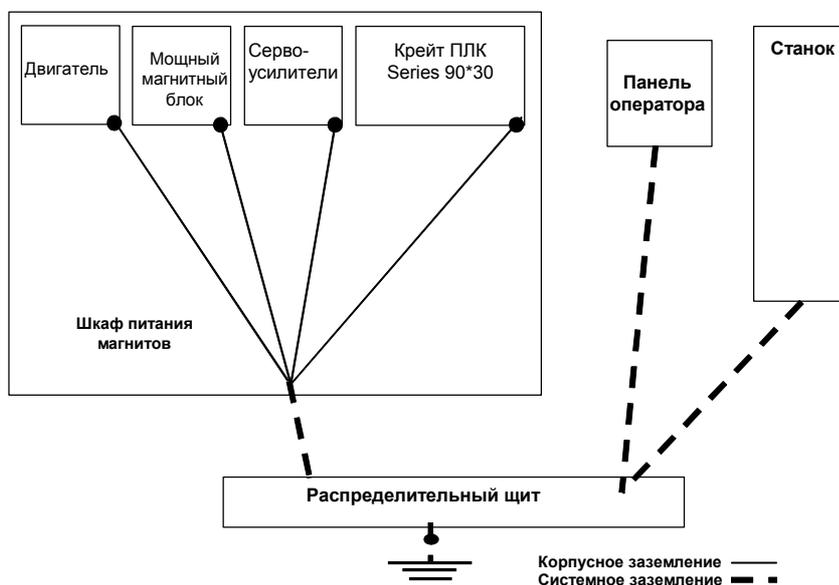


Рис. 2-16. Точки подключения заземления системы Motion Mate DSM314

Таблица 2-6. Системы Заземления

Система Заземления	Описание
Система "Корпусная Земля"	Система "корпусной земли" используется для обеспечения безопасности и для подавления внешних и внутренних помех. В системе ""корпусной земли" выполняется соединение рам, корпусов блоков, панелей и экранов кабелей между собой.
Система "Системной Земли"	Система "системной земли" используется для соединения с землей систем "корпусной земли", действующей между приборами или блоками.

Таким образом, завершается установка системы Motion Mate DSM314.

Устанавливая связи станка, необходимо выполнить еще некоторые действия, однако в данный момент можно перейти к следующему этапу . . .

+

### **Включение Системы Motion Mate DSM314**



## Раздел 3: Включение Motion Mate DSM314

До того как подать питание, необходимо выполнить следующее:

- Убедитесь, что кабели, входящие в комплект поставки, подключены должным образом к соответствующим разъемам.
- Убедитесь, что все подключение к источникам питания выполнено правильно.
- Убедитесь, что обеспечена надлежащая защита двигателей.
- Проверьте правильность заземления всех компонентов, включая экран лицевой панели.
- Если используется более одного двигателя, то убедитесь, что подключения сервоусилителей и кабелей обратной связи не пересекаются между двигателями.

Для подачи питания на Систему Управления DSM314 имеется определенная последовательность. Выполните следующие шаги **по порядку, указанному в списке**:

1. Подайте 220 В ~ на Цифровой сервопривод. Убедитесь, что светодиодный индикатор *наличия напряжения* на усилителе горит.
2. Для Цифровых усилителей  $\beta$  Series включите источник питания 24 В =. Убедитесь, что индикатор *Питание* на усилителе горит.
3. Включите питание на ПЛК Series 90-30 и проверьте, что горит светодиод PWR (Питание) на Источнике Питания Series 90-30.
4. Используя соответствующий кабель связи, подключите к ПЛК Ручной Программатор (Hand Held Programmer = ННР) или персональный компьютер (PC), содержащий программное обеспечение конфигурации / программирования. Более детальную информацию см. в "*Руководстве по Установке и по Аппаратному обеспечению ПЛК Series 90-30*", GFK-0356.
5. Установите на ПЛК режим **СТОП / Запрещено**. Это может быть выполнено при помощи программного обеспечения конфигурации / программирования на PC или при помощи ключей Ручного Программатора (Режим Run –).
6. Если используется удерживающий тормоз, устанавливаемый на двигателе по спецзаказу, то необходимо подать соответствующее напряжение (90 В постоянного тока для двигателей  $\alpha$  Series и  $\beta$  Series и 24 В пост. тока для двигателей SL Series), чтобы удерживающий тормоз освободил вал двигателя.

Следующий шаг . . .

+ . . . . . **Конфигурация Motion Mate DSM314**

## Раздел 4: Конфигурация Motion Mate DSM314 (вводное руководство)

Конфигурация Контроллера DSM314 выполняется с использованием программного обеспечения VersaPro для конфигурации (версия 1.1 или выше) или с использованием Программы Разработки Логики (Machine Edition Logic Developer) SIMPLICITY для ПЛК (версия 2.1 или выше).

Модуль DSM314 имеет широкий набор параметров, который позволяет адаптировать это устройства к самым разным приложениям. Можно легко выполнить настройку на конкретную систему движения. Регистры параметров в памяти DSM314 позволяют использовать переменные в программах движения DSM314.

DSM314 имеет множество параметров конфигурации; однако для большинства приложений достаточно установить лишь некоторые из них. Остальные параметры конфигурации устанавливаются, как правило, по умолчанию.

В данном разделе дано краткое описание полей конфигурации DSM314, а также изложен порядок просмотра и задания конфигурационных параметров, требуемых для перемещения осей.

### Подключение Программатора к ПЛК

Все программирование DSM314 выполняется через интерфейс программного обеспечения конфигурации / программирования, обеспечивая единственную точку входа в программирование модуля. Более детальную информацию см. в "Руководстве по Установке и по Аппаратному Обеспечению ПЛК Series 90-30", GFK-0356. Программная среда поддерживает несколько интерфейсов связи, имеет несколько возможностей связи. Одним из возможных вариантов подключения является подсоединение программатора непосредственно к последовательному порту ПЛК, как показано на рис. 2-17. Обмен информацией идет по протоколу SNP. Дополнительные методы связи см. в документации по программному обеспечению.

**Примечание:** DSM314 имеет также последовательный порт на лицевой плате модуля. Этот последовательный порт используется только для обновления микропрограмм DSM314.

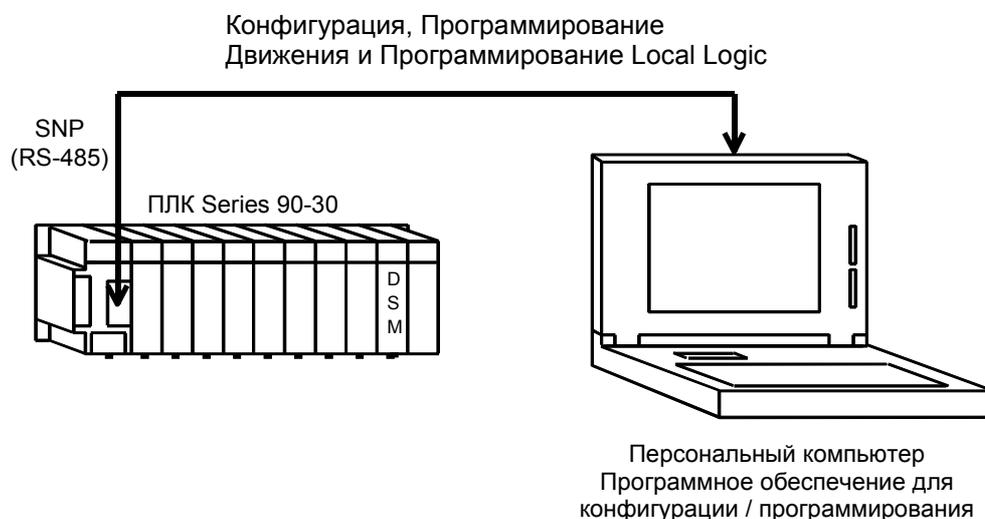


Рис. 2-17. Схема Подключения Программатора DSM

## Конфигурация VersaPro

Конфигурация ПЛК и логическая программа, а также конфигурация DSM314 хранятся в каталоге программ VersaPro на жестком или гибком диске. Пользовательская среда VersaPro представляет собой автономную среду, которая позволяет выполнять все действия, необходимые для конфигурации, создания, редактирования и загрузки программ в ПЛК и DSM.

1. Запуск ПО VersaPro.
2. Находясь в VersaPro, создайте новую папку (каталог) VersaPro.
  - A. Чтобы создать папку VersaPro, выберите меню **F**ile (Файл), а затем пункт меню **N**ew Folder (Новая Папка). В результате появляется Мастер по Созданию Новой Папки. Требуется ввести Folder **N**ame (Название Папки), **L**ocation (Местонахождение) и Folder **D**escription (Описание Папки). В данном примере введите текст, как показано в диалоговом окне. (Рис. 2-1).

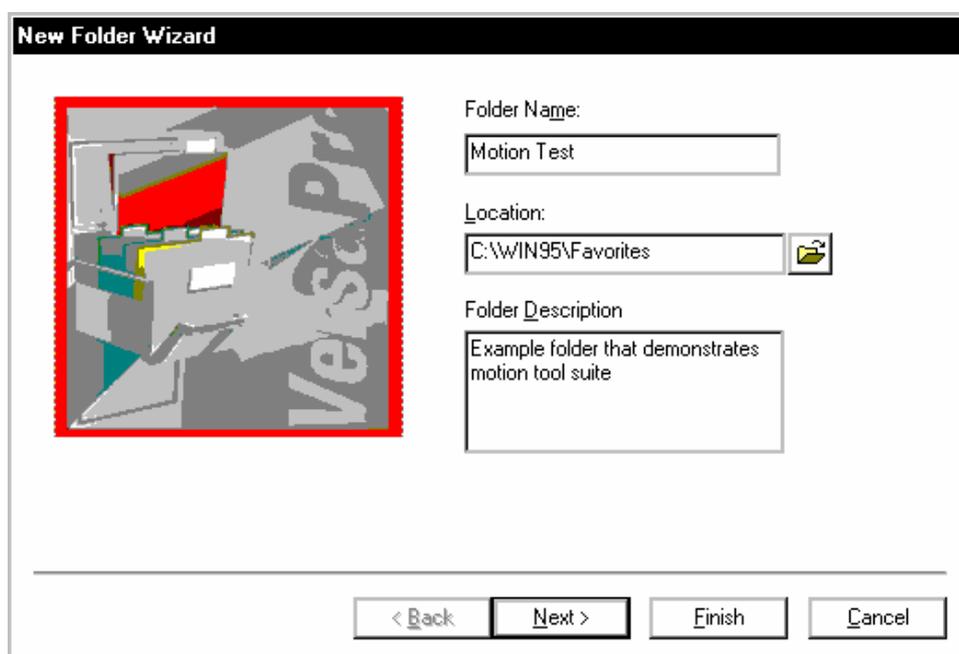


Рис. 2-18. Мастер по Созданию Новой Папки, Стр. 1

- B. Щелкните кнопку **N**ext (Далее). Следующее диалоговое окно позволяет начать с пустой папки или импортировать ее содержимое из другого источника. В данном примере выбрана пустая папка. (Рис. 2-3).

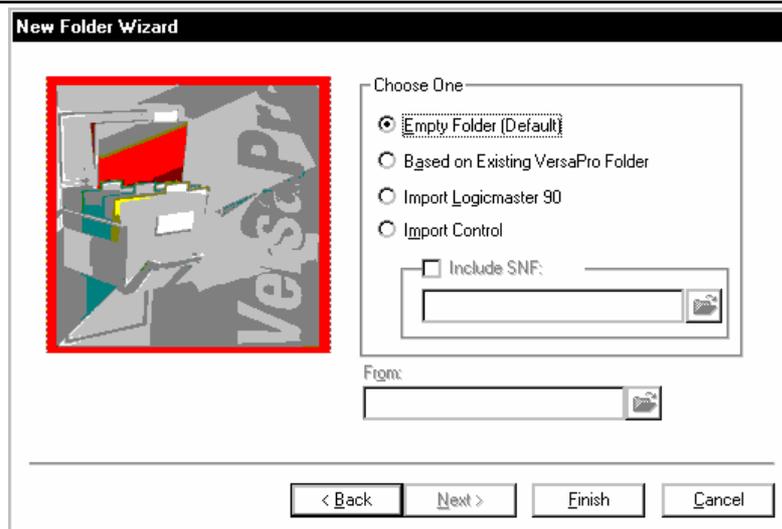


Рис. 2-19. Мастер по Созданию Новой Папки, Стр. 2

- С. Щелкните кнопку Finish (Завершить), чтобы создать новую папку. Результирующее окно VersaPro показано на рис. 2-20.

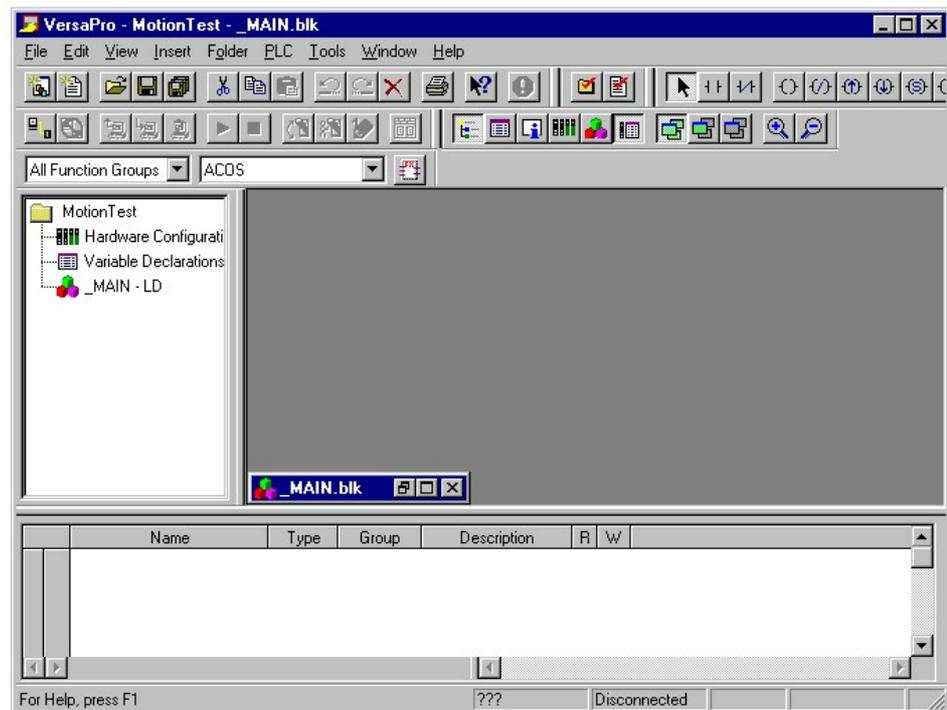


Рис. 2-20. Главное Окно Новой Папки VersaPro

3. Запустите инструментарий конфигурации аппаратного обеспечения. Для этого имеется несколько возможностей. (Подробности см. в документации по VersaPro). Двумя методами являются следующие:
  - В меню View (Вид) выберите пункт Hardware Configuration (Конфигурация Аппаратного обеспечения).
  - Щелкните значок конфигурации аппаратного обеспечения в окне просмотра папки.
4. Может понадобиться изменить конфигурацию, устанавливаемую по умолчанию, на конфигурацию Series 90-30. Чтобы сделать это, выберите на линейке команд команду File (Файл), затем Convert To (Преобразовать В) и затем Series 90-30. Это изменит

конфигурацию ПЛК, устанавливаемую по умолчанию, на конфигурацию Series 90-30. Эта последовательность действий с меню показана на рис. 2-4.

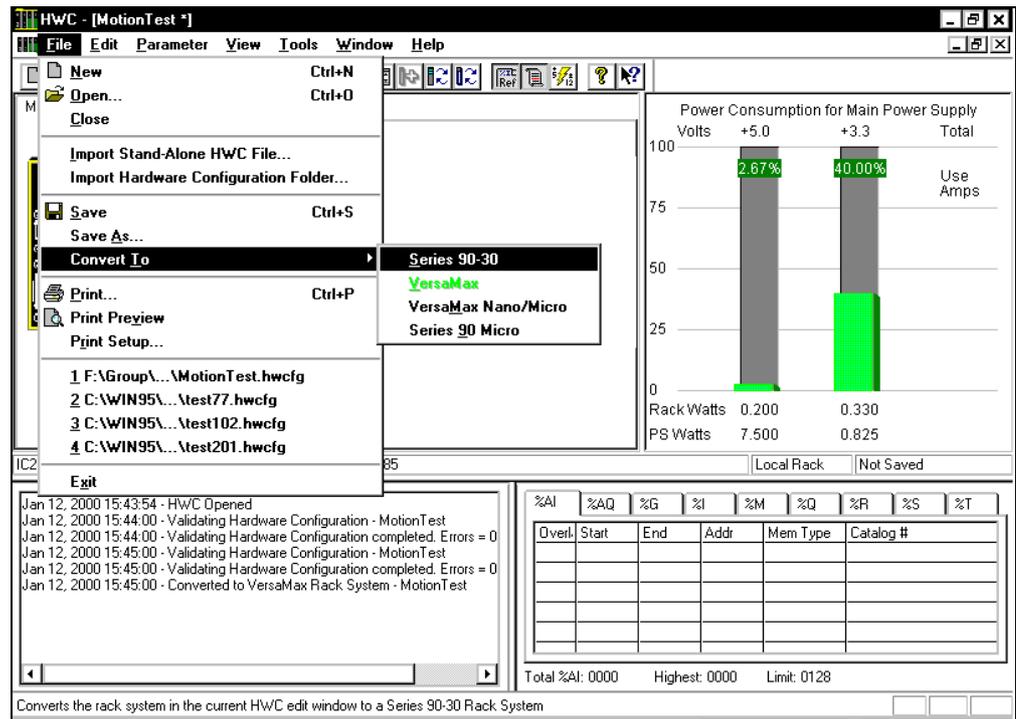


Рис. 2-21. Выбор Стойки при Конфигурации Аппаратного обеспечения

Диалоговое окно предупреждает об удалении информации. Поскольку папка, созданная в данном примере, новая, то потери информации не произойдет. Если новая папка не создавалась, то помните, что при этой операции произойдет потеря информации о конфигурации. Рекомендуется использовать в данном примере новую пустую папку. Щелкните Yes (Да) в диалоговом окне (рис. 2-5).

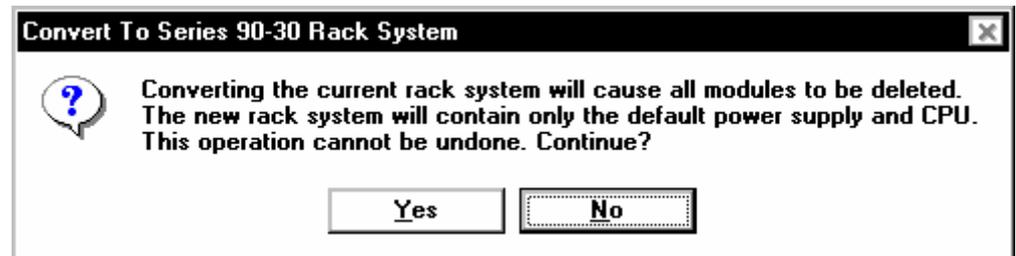


Рис. 2-22. Диалоговое Окно Преобразования Стойки при Конфигурации Аппаратного обеспечения

После завершения этой процедуры получается незаполненная стойка 90-30, которую необходимо сконфигурировать. Получающееся окно Конфигурации Аппаратного обеспечения должно выглядеть так, как показано на 2-6.

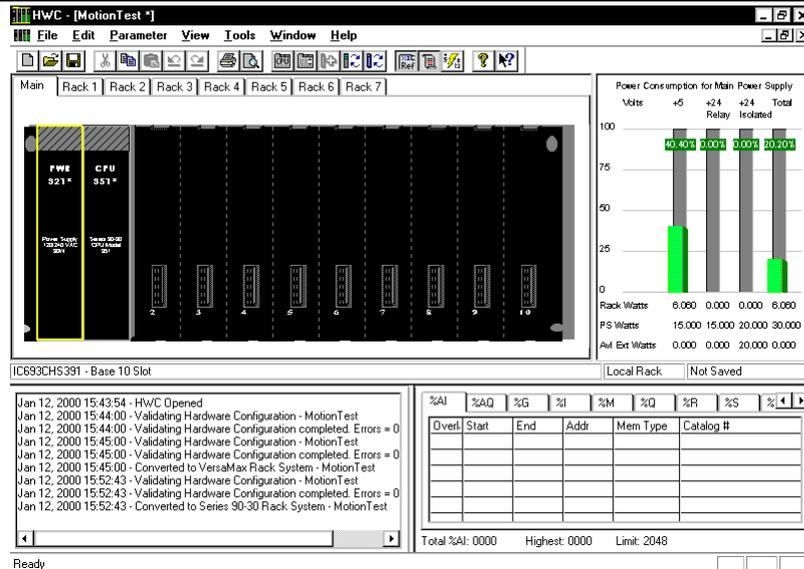


Рис. 2-23. Конфигурация Аппаратного обеспечения стойки 90-30 с ЦП

5. Выберите источник питания и ЦП, соответствующие имеющемуся оборудованию. Учтите, что DSM314 требует микропрограммного обеспечения ЦП версии 10 или выше. ЦП по умолчанию – "CPU351" – не поддерживает версию 10.0 микропрограмм. В документации на ЦП имеется список ЦП, поддерживающий функции версии 10.0. В данном примере используется CPU364.

Чтобы изменить модель ЦП, щелкните правой кнопкой CPU (ЦП) и выберите Replace Module (Заменить Модуль). Появится диалоговое окно Module Catalog (Каталог Модулей). Выберите модель ЦП, установленный в рассматриваемой системе, и щелкните ОК. Щелкните в Yes в появившемся предупреждающем диалоговом окне. Результирующее окно выглядит так, как показано на рис. 2-7.

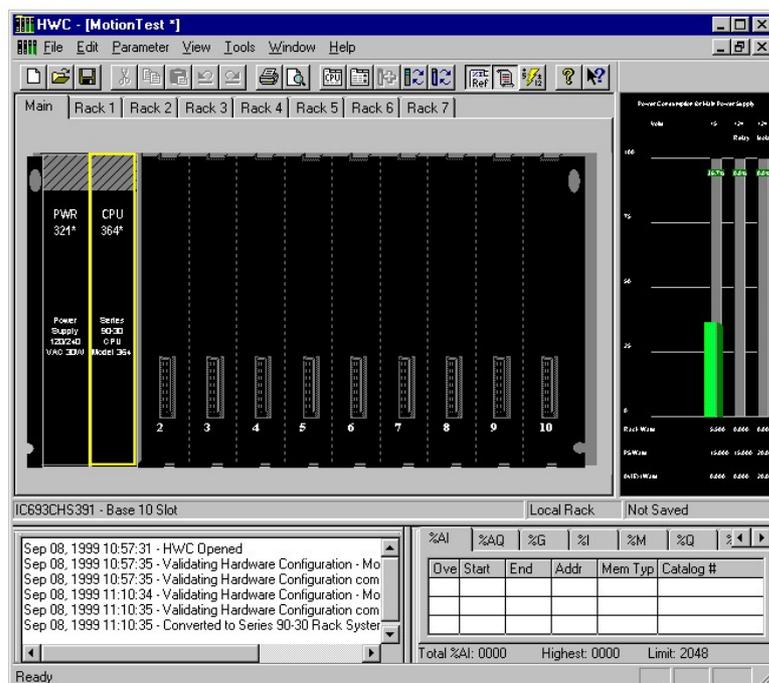


Рис. 2-24. Конфигурация Аппаратного обеспечения стойки 90-30 с CPU364

6. Добавьте в стойку DSM314.

Выберите тот слот стойки, куда должен быть установлен модуль DSM314 (в данном примере Слот 2). Откройте диалоговое окно Module Catalog (Каталог Модулей), используя один из следующих двух методов:

- Дважды щелкните требуемый слот (в данном случае слот №2).
- Выберите на строке меню команду Edit (Редактировать), затем из подменю Edit выберите Module Operations (Действия с Модулями), и в нем выберите Add Module (Добавить Модуль).

7. В диалоговом окне Module Catalog (Каталог Модулей) выберите закладку Motion, а на ней выберите из списка модуль DSM314. Результирующее окно выглядит так, как показано на рис. 2-8.

Эта операция добавляет модуль DSM314 в стойку и отображает окна конфигурации DSM314, которые позволяют настроить DSM314 на конкретное приложение. В разделе "Данные по Конфигурации Модуля (Только как Вводное руководство по Началу Работы)" на стр. 2-35 дан пример конфигурации. См. гл. 4 о деталях параметров конфигурации DSM314.

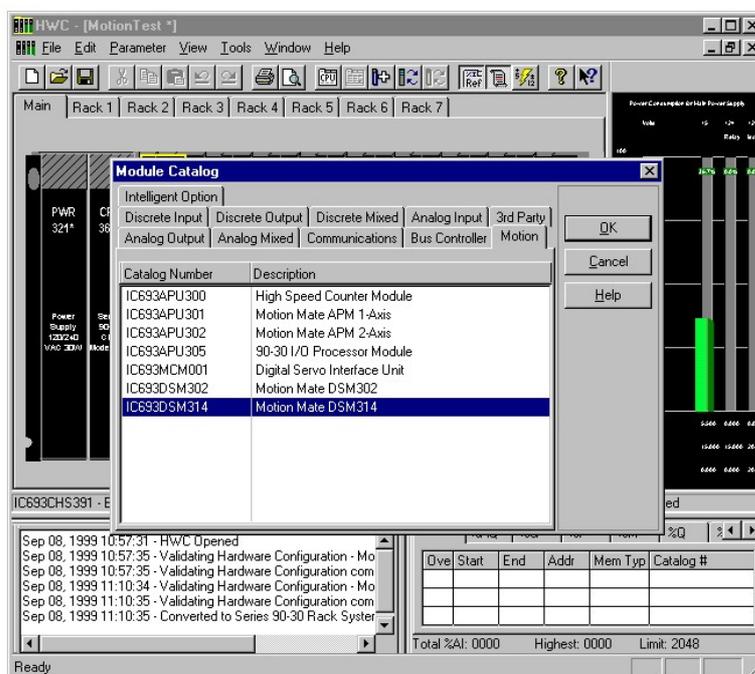
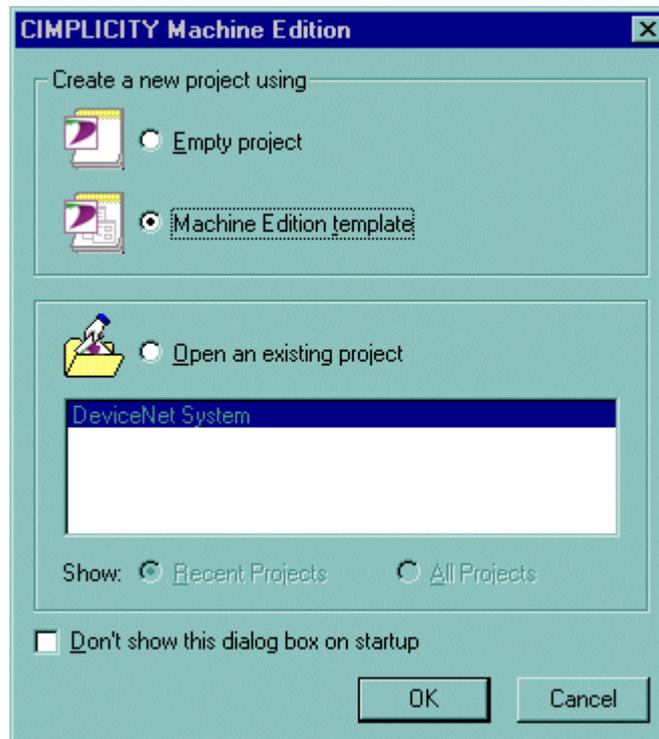


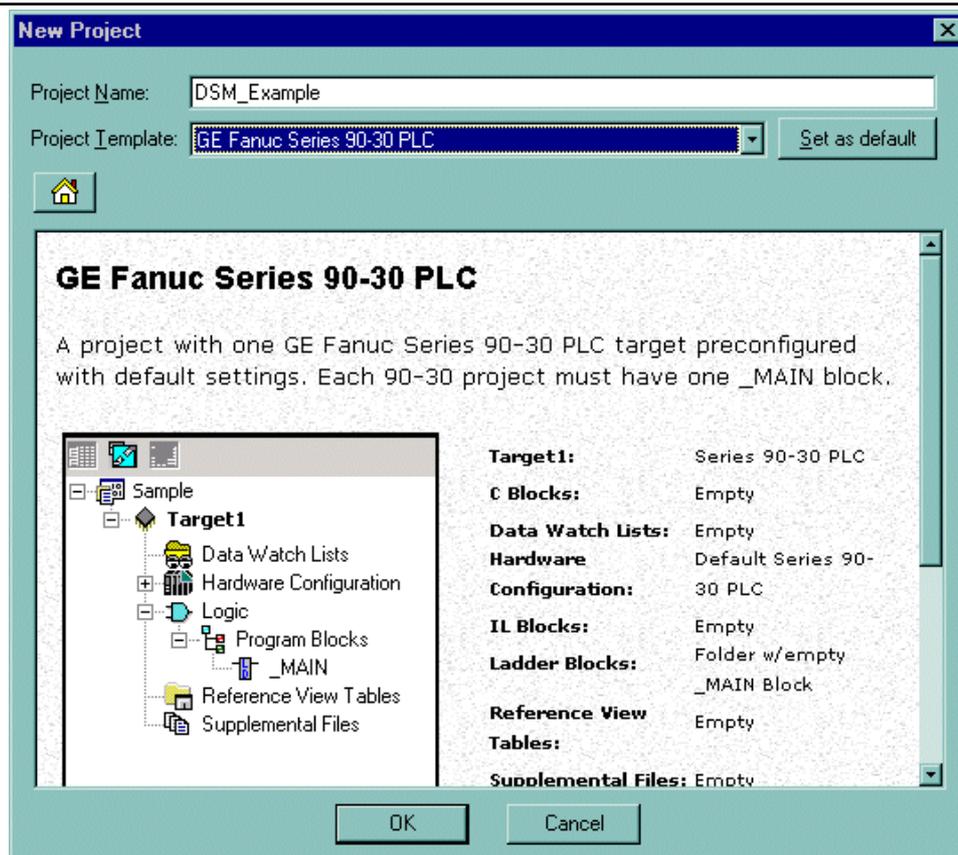
Рис. 2-25. Выбор DSM314 в ходе Конфигурации Аппаратного обеспечения Стойки 90-30

## Конфигурация при помощи программы SIMPLICITY Machine Edition

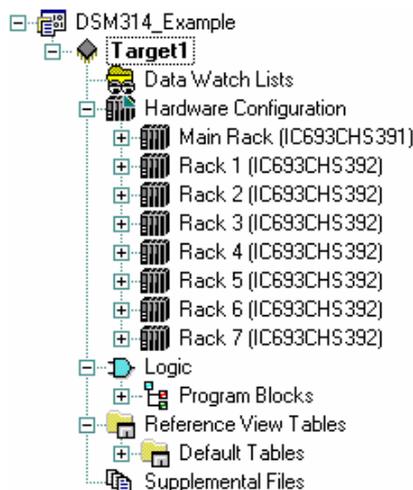
1. Запустите программу Разработки Логки SIMPLICITY, Machine Edition, для ПЛК. Появляется диалоговое окно SIMPLICITY Machine Edition.



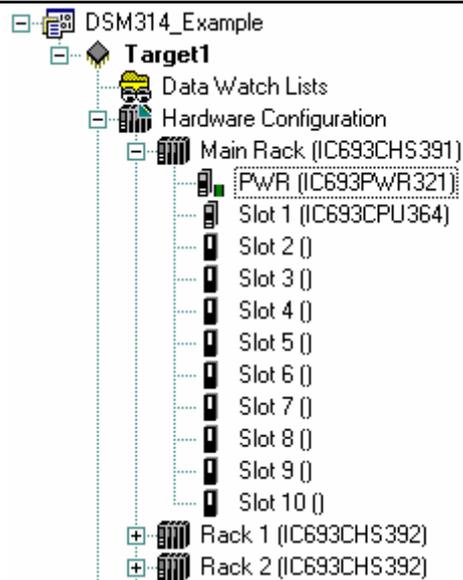
2. В окне Create a New Project (Создать Новый Проект) выберите Machine Edition Template (Шаблон Machine Edition) и щелкните OK. Откроется диалоговое окно New Project (Новый Проект).
3. Введите Project Name (Названия Проекта) (любое). В выпадающем списке Project Template (Шаблон Проекта) выберите ПЛК GE Fanuc Series 90-30. Щелкните OK.



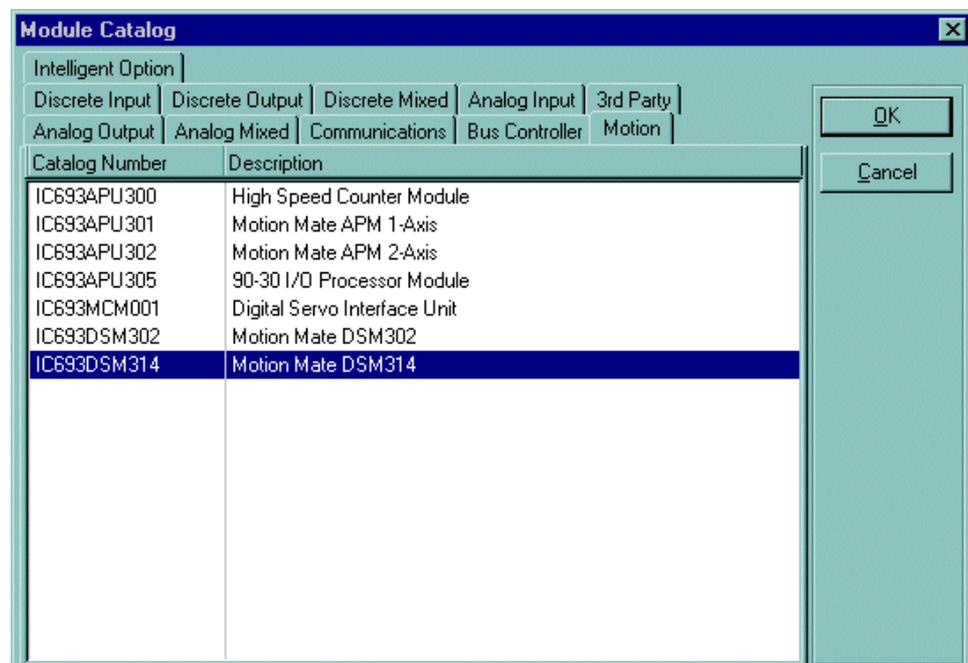
Проект появится в окне Navigator (Навигатор), как показано на рисунке ниже.



4. Щелкните узел Главной Стойки, где содержится источник питания по умолчанию и ЦП.



5. При необходимости замените источник питания и / или ЦП теми моделями, которые будут использоваться в данном приложении. Для замены модуля щелкните правой клавишей и выберите Replace Module (Заменить Модуль).
6. Добавьте DSM314 в конфигурацию стойки.
  - A. Щелкните правой клавишей пустой слот и выберите Add Module (Добавить Модуль). Появится диалоговое окно Module Catalog (Каталог Модулей).
  - B. Выберите закладку Motion (Движение), выделите DSM314 и щелкните ОК.



Эта операция добавляет модуль DSM314 в стойку и отображает окна конфигурации DSM314, которые позволяют настроить DSM314 на конкретное приложение. В разделе "Данные по Конфигурации Модуля (Только как Вводное руководство по Началу Работы)" на стр. 2-35 дан пример конфигурации. См. гл. 4 о деталях параметров конфигурации DSM314.

## Данные по Конфигурации Модуля (Только как Вводное руководство по Началу Работы)

Если выполнены описанные выше действия, то появляются окна конфигурации DSM314. Если вы начинаете с этого этапа, то щелкните дважды тот модуль DSM314, который требуется редактировать. Это действие откроет окна конфигурации. После этого выберите или введите значения из приведенных далее таблиц для того типа сервосистемы, цифровой или аналоговой, которую предполагается использовать. В главе 4 данного руководства даны дополнительные сведения по каждому параметру конфигурации, что позволяет задать параметры применительно к требованиям конкретного приложения.

**Таблица 2-7. Параметры Установки (только как Вводное Руководство для Начала Работы)**

Параметр Конфигурации	Описание	Для Цифрового Режима	Для Аналогового Режима	Значения параметров по умолчанию
Number of Axes	Количество Осей	2	4	4
%I Reference	Начальный адрес для ссылки типа %I (80 бит)	%I00001	%I00001	Следующая наибольшая имеющаяся %I ссылка
%I Length	Длина адреса ссылки %I	48	80	80
%Q Reference	Начальный адрес для ссылки типа %Q (80 бит)	%Q00001	%Q00001	Следующая наибольшая имеющаяся %Q ссылка
%Q Length	Длина адреса ссылки %Q	48	80	80
%AI reference	Начальный адрес для ссылки типа %AI (84 бита)	%AI00001	%AI00001	Следующая наибольшая имеющаяся %AI ссылка
%AI Length	Длина адреса ссылки %AI	44	84	84
%AQ reference	Начальный адрес для ссылки типа %AQ (12 бит)	%AQ00001	%AQ00001	Следующая наибольшая имеющаяся %AQ ссылка
%AQ Length	Длина адреса ссылки %AQ	6	12	12
Axis 1 Mode	Режим Управления Оси 1	<b>Цифровой Сервопривод</b>	Аналоговый Сервопривод	Аналоговый Сервопривод
Axis 2 Mode	Режим Управления Оси 2	<b>Цифровой Сервопривод</b>	Аналоговый Сервопривод	Аналоговый Сервопривод
Axis 3 Mode	Режим Управления Оси 3	<b>Вспомогат. Ось</b>	Аналоговый Сервопривод	Вспомогательная Ось
Axis 4 Mode	Режим Управления Оси 4	<b>Отключено</b>	Аналоговый Сервопривод	Отключено
Local Logic Mode	Режим Local Logic Engine	Отключено	Отключено	Отключено
Total Encoder Power	Требования к мощности датчика положения	0	**	0
Motion Program Block Name	Имя программы движения, выполняемой на модуле	<пусто>	<пусто>	<пусто>
Local Logic Block Name	Имя блока локальной программы Local Logic, выполняемой на модуле	<пусто>	<пусто>	<пусто>
Cam Block Name	Имя блока САМ-программы, выполняемой на модуле	<пусто>	<пусто>	<пусто>

Таблица 2-8. Параметры Оси (только как Вводное Руководство для Начала Работы)

Параметр Конфигурации	Описание	Для Цифрового Режима	Для Аналогового Режима	Значения параметров по умолчанию
User Units	Значение Пользовательских Единиц	1	1	1
Counts	Единицы счета Обратной связи	1	1	1
OT Limit Sw	Включатель выхода за границу	<b>Отключено</b>	<b>Отключено</b>	Включено
Drive Ready Input	Вход готовности устройства	Включено	<i>Зависит от Сервоусилителя</i>	Включено
High Position Limit	Верхний предел положения	+8388607	+8388607	+8388607
Low Position Limit	Нижний предел положения	-8388608	-8388608	-8388608
High Software EOT Limit	Верхний программный ограничитель движения	+8388607	+8388607	+8388607
Low Software EOT Limit	Нижний программный ограничитель движения	-8388608	-8388608	-8388608
Software End of Travel	Программный ограничитель движения	Отключено	Отключено	Отключено
Velocity Limit	Ограничение скорости	1,000,000	1,000,000	1,000,000
Command Direction	Команда направления	Два направления	Два направления	Два направления
Axis Direction	Направление Оси	Нормальное	Нормальное	Нормальное
Feedback Source	Обратная связь	По умолчанию	По умолчанию	По умолчанию
Feedback Mode (только Цифровой Режим)	Режим обратной связи	Инкрементный	<b>нет</b>	Инкрементный
Reversal Compensation	Компенсация Реверсирования	0	0	0
Drive Disable Delay	Задержка выключения привода	100	100	100
Jog Velocity	Скорость передвижения	+1000	+1000	+1000
Jog Acceleration	Ускорение передвижения	+10,000	+10,000	+10,000
Jog Acceleration Mode	Тип ускорения передвижения	Линейный	Линейный	Линейный
Home Position	Начальная позиция	0	0	0
Home Offset	Начальное смещение	0	0	0
Final Home Velocity	Скорость подхода к начальной позиции	+500	+500	+500
Find Home Velocity	Скорость поиска начальной позиции	+2000	+2000	+2000
Home Mode		<b>Движение +</b>	<b>Движение +</b>	Переключатель Исходной Позиции
Return Data 1 Mode	Режим возврата данных 1	0	0	0
Return Data 1 Offset	Смещение возврата данных 1	0	0	0
Return Data 2 Mode	Режим возврата данных 2	0	0	0
Return Data 2 Offset	Смещение возврата данных 2	0	0	0
Cam Master Source	Мастер САМ	Текущая Позиция 3	Текущая Позиция 3	Текущая Позиция 3
Follower Control Loop	Связанный способ управления	Отключено	Отключено	Отключено
Ratio A Value	Значение коэффициента А	1	1	1
Ratio B Value	Значение коэффициента В	1	1	1
Follower Master Source 1	Связанный мастер 1	Нет	Нет	Нет
Follower Master Source 2	Связанный мастер 2	Нет	Нет	Нет
Follower Enable	Связанный триггер	Нет	Нет	Нет

Trigger	разрешения			
Follower Disable Trigger	Связанный триггер запрещения	Нет	Нет	Нет
Follower Disable Action	Связанное действие запрещения	Стоп	Стоп	Стоп
Ramp Makeup Acceleration	Формирование угла ускорения	10,000	10,000	10,000
Ramp Makeup Mode	Режим формирования угла	Время нарастания	Время нарастания	Время нарастания
Ramp Makeup Time	Время формирования угла	0	0	0
Ramp Makeup Velocity	Скорость формирования угла	+1000	+1000	+1000

**Таблица 2-9. Параметры Настройки (только как Вводное Руководство для Начала Работы)**

Параметр Конфигурации	Описание	Для Цифрового Режима	Для Аналогового Режима Скорости	Значения параметров по умолчанию
Motor Type	Тип Двигателя	Из табл. 2-1	0	0
Analog Servo Command	Команда Аналогового Сервопривода	Скорость	Скорость	Скорость
Position Error Limit	Предел Ошибки позиционирования	60,000	60,000	60,000
In Position Zone	В зоне позиционирования	10	10	10
Pos Loop Time Constant (0.1 ms)	Постоянная Времени Позиционирования	600*	1000*	1000
Velocity at MaxCmd	Скорость при Максимальном значении	нет	100,000*	100,000
Velocity Feed Forward Percentage	Коэффициент упреждения скорости	0	0	0
Acceleration Feed Forward Percentage	Коэффициент упреждения ускорения	0	0	0
Integrator Mode	Режим Интегратора Контра Позиционирования	Выкл.	Выкл.	Выкл.
Integrator Time Constant	Постоянная Времени интегрирования Контра Позиционирования	0	0	0
Velocity Loop Gain	Управляющий коэффициент скорости	16	16	16

\* Правильные значения параметров должны быть определены в ходе запуска и настройки системы. См. стр. "Информация о Запуске и Настройке для Аналоговых Сервосистем" в Приложении D.

\*\* Правильные значения параметров должны быть определены при требованиях к питанию 5 В.

**Примечание:** Повторите приведенные выше установки для конфигурации осей 2, 3 и 4, если требуется.

## Тип Двигателя (Цифровой Режим)

Выберите тип серводвигателя переменного тока FANUC, который должен быть использован вместе с DSM314. DSM314 хранит в своей памяти таблицы установочных параметров, принимаемых по умолчанию, для всех поддерживаемых сервоприводов GE Fanuc. Конкретный двигатель для отображаемой оси выбирается при помощи полей конфигурации *Тип Двигателя 1* (для оси 1) или *Тип Двигателя 2* (для оси 2). Поддерживаемые типы двигателей приведены в таблице внизу.

**Модель Двигателя FANUC.** Информация о модели двигателя дается в форме "*серия, значение постоянного момента вращения в ньютон x метр / максимальное число оборотов в минуту*". Пример:  $\beta 2/3000$  обозначает двигатель серии "бета" ( $\beta$ ) с величиной постоянного момента вращения 2 нм и с максимальным числом оборотов в минуту, равным 3000.

**Спецификация Двигателя FANUC.** Информация о конфигурации для поля типа двигателя определяется по информации на заводской табличке двигателя. Номера компонентов двигателя имеют вид A06B-xxxx-уууу, где xxxx представляет собой поле Спецификации Двигателя. Например: Соответствующая последовательность цифр 0032 на заводской табличке двигателя A06B-0032-B078 обозначает модель  $\beta 2/3000$ . Таблица внизу показывает, что "36" является правильным Кодом Типа Двигателя для Спецификации Двигателя "0032".

**Таблица 2-10. Выбор Кода Типа Двигателя FANUC**

Код Типа Двигателя	Модель Двигателя	Спецификация Двигателя
61	$\alpha 1/3000$	0371
46	$\alpha 2/2000$	0372
62	$\alpha 2/3000$	0373
15	$\alpha 3/3000$	0123
16	$\alpha 6/2000$	0127
17	$\alpha 6/3000$	0128
18	$\alpha 12/2000$	0142
19	$\alpha 12/3000$	0143
27	$\alpha 22/1500$	0146
20	$\alpha 22/2000$	0147
21	$\alpha 22/3000$	0148
28	$\alpha 30/1200$	0151
22	$\alpha 30/2000$	0152
23	$\alpha 30/3000$	0153
30	$\alpha 40/2000$	0157
29	$\alpha 40/FAN$	0158
56	$\alpha L 3/3000$	0561
57	$\alpha L 6/3000$	0562
58	$\alpha L 9/3000$	0564
59	$\alpha L 25/3000$	0571
60	$\alpha L 50/2000$	0572

Продолжение на следующей странице

Таблица 2-10, продолжение

Код Типа	Модель	Спецификация
7	$\alpha$ C 3/2000	0121
8	$\alpha$ C 6/2000	0126
9	$\alpha$ C 12/2000	0141
10	$\alpha$ C 22/1500	0145
3	$\alpha$ 12HV/3000	0176
4	$\alpha$ 22HV/3000	0177
5	$\alpha$ 30HV/3000	0178
24	$\alpha$ M 3/3000	0161
25	$\alpha$ M 6/3000	0162
26	$\alpha$ M 9/3000	0163
13	$\beta$ 0.5/3000	0113
115	$\beta$ M 0.5/5000	0115
35	$\beta$ 1/3000	0031
116	$\beta$ M1/5000	0116
36	$\beta$ 2/3000	0032
33	$\beta$ 3/3000	0033
34	$\beta$ 6/2000	0034

## Сохранение Конфигурации на ПЛК

Необходимо провести до конца конфигурацию системы Series 90-30, включив в нее Источник Питания, Стойку, ЦП и дополнительные модули, чтобы соответствовать используемой системе. Обращайтесь за помощью к Руководству пользователя программным обеспечением и по он-лайн линии, если требуется.

### ВАЖНО!

Законченная конфигурация должна быть сохранена в ПЛК. Инструкции по тому, как выполнить это, см. в главе 15 "Подключение к ПЛК и Сохранение Конфигурации". Дополнительную информацию можно получить в Руководстве пользователя программным обеспечением и по он-лайн линии.

Если все в порядке (т.е. после Сохранения нет сообщений об ошибках статуса ПЛК или Ввода / Вывода и т.п.), то следующим шагом должно быть следующее: **Проверка системы.**

**Примечание:** Ошибка статуса ПЛК из "Несоответствия Конфигурации Системы" с тем же самым расположением стойки / слота, что и модуль DSM314, указывает на то, что был сконфигурирован и отправлен в DSM314 параметр, который был не принят этим модулем. Тщательно сверьте каждый параметр конфигурации DSM314 на предмет несоответствия с установками конфигурации, приведенной в данном руководстве. Устраните несоответствие, очистите таблицу сбоя ПЛК и снова сохраните конфигурацию. Убедитесь, что ошибка исправлена. Действия по просмотру и очистке таблицы сбоя ПЛК см. в следующем разделе "Режим Разрешения Работы на ПЛК".

Модуль DSM314 может обнаружить многие типичные ошибки конфигурации. Они отображаются, как коды ошибки, в виде Dxxx (шестнадцатеричный) в слове *Код Статуса Модуля %AI* или в словах *Код Ошибки Оси %AI*. Эти ошибки не влияют на статус ПЛК в "Несоответствии Конфигурации Системы". См. в Приложении А описание этих ошибок. Исправьте все ошибки конфигурации и сохраните конфигурацию ПЛК в Стоп-режиме.

## Раздел 5: Проверка системы.

### Формирование Движение

#### Предостережение

Для правильной работы станка необходимо выполнить рекомендуемую процедуру пуска. Она включает в себя операцию проверки Границ пере регулирования и Переключателя Исходной Позиции, проверку правильности направления вращения двигателя и настройки контуров скорости и позиционирования. Все это должно быть выполнено опытным персоналом. Подробные инструкции по пуску см. в Приложении D "Пуск и Настройка Цифровых или Аналоговых Сервосистем GE Fanuc".

### Режим Разрешения Работы на ПЛК

Следующим шагом в работе с системой DSM314 является задание режима RUN (РАБОТА) для ПЛК.

#### VersaPro

1. Из главного меню VersaPro выберите меню PLC (ПЛК). В подменю PLC выберите пункт Connect (Соединить). Появится меню Connect. В этом меню выберите тот метод / порт связи, который предполагается использовать для связи с ПЛК. Если этот порт связи еще не сконфигурирован, то обратитесь к документации GFK-1670 VersaPro или по он-лайн линии помощи по VersaPro за дальнейшей информацией.
2. Из главного меню VersaPro выберите меню Tools (Инструменты). В подменю Tools выберите вход Fault Table (Таблица Сбоев). Появится таблица сбоев ПЛК. Просмотрите таблицу сбоев на предмет любых проблем, а затем используйте функциональную клавишу F9 Clear (Очистить), чтобы устранить любые имеющиеся сбои ПЛК. Выберите закладку таблицы сбоев I/O (Ввода / Вывода) и выполните эти же операции по тем же сбоям. Закройте таблицу сбоев и вернитесь в главное окно VersaPro.
3. Из главного меню VersaPro выберите меню PLC (ПЛК). В подменю PLC выберите пункт Run (Работа). ПЛК перейдет в режим работы.

#### Предупреждение

**Убедитесь, что двигатель закреплен надлежащим образом. При несоблюдении этого может произойти повреждение двигателя или травмирование персонала.**

4. Убедитесь, что светодиоды ПЛК, обозначенные как PWR, OK и RUN горят. Убедитесь, что светодиоды модуля DSM314, обозначенные как STAT, OK и CFG горят. Для Цифровых Усилителей  $\alpha$  Series убедитесь, что на индикации статуса усилителя виден знак "минус" (-), что означает Режим Ожидания. Для Цифровых Усилителей  $\beta$  Series убедитесь, что светодиод Power (Питание) горит, и что светодиоды ALM не горят. Для усилителей SL Series проверьте дисплей передней панели каждого усилителя на предмет каких-либо ошибок, (это отображается миганием всех цифр на дисплее).
5. Если усилитель демонстрирует наличие неисправности (сбоя), то необходимо полностью снять питание с него и с ПЛК и повторить последовательность включения, описанную в разделе 3.

6. При допущении, что ошибки (неисправности) отсутствуют, появляется возможность осуществить толчковое перемещение соответствующих осей. Для ранее описанной конфигурации пуска следует, чтобы устранить любые сбои, индицируемые на DSM314 миганием светодиода STAT, переключить (т.е. сначала включить, затем выключить) бит *Clear Error %Q* (%Q1 для данной конфигурации) в таблице данных ПЛК. Необходимо устранить причины появления сбоя так, чтобы очистить индикатор состояния. Для получения информации по Статусу Ошибок DSM см. Приложение А "Коды Ошибок".

### CIMPLICITY Machine Edition

1. Чтобы сконфигурировать связи в окне Навигатора CIMPLICITY Machine Edition, щелкните правой кнопкой узел Target (Цель) и выберите Properties (Свойства). Открывается окно Inspector (Инспектор), показывающее свойства целевого объекта, в которые входят параметры конфигурации канала связи. Подробнее см. на он-лайн поддержки или в "Руководстве по началу работы с программой Разработки Логики ПЛК CIMPLICITY® Machine Edition", GFK-1918.
2. Щелкните правой кнопкой узел Target и выберите Go Online (Выйти на он-лайн). Появится диалоговое окно Connecting (Соединение). Если программатор успешно подключился к ПЛК, то значок узла Target изменяется, показывая либо Online (Он-лайн) и Equal (Равно) , либо Online и Не Равно . (Если подключение Online и Не Равно, то логика и/или конфигурация на ПЛК не соответствует тому, что имеется в программаторе. В рамках цели данного вводного курса может оказаться необходимым загрузить вашу конфигурацию.)
3. Чтобы открыть Fault Table Viewer (Просмотр Таблиц Сбоев), щелкните правой клавишей узел Target и выберите Diagnostics (Диагностика). Просмотрите таблицы сбоев ПЛК на предмет каких-либо проблем, а затем устраните все имеющиеся сбои ПЛК. Выполните ту же операцию по сбоям Ввода / Вывода.
4. В окне Navigation (Навигация) щелкните правой клавишей узел Target и выберите Online Commands (Он-лайн-Команды), затем Start PLC (Пуск ПЛК). ПЛК перейдет в режим работы.

### Предупреждение

**Убедитесь, что двигатель закреплен надлежащим образом. При несоблюдении этого может произойти повреждение двигателя или травмирование персонала.**

5. Убедитесь, что светодиоды ПЛК, обозначенные как PWR, OK и RUN горят. Убедитесь, что светодиоды модуля DSM314, обозначенные как STAT, OK и CFG горят. Для Цифровых Усилителей  $\alpha$  Series убедитесь, что на индикации статуса усилителя виден знак "минус" (-), что означает Режим Ожидания. Для Цифровых Усилителей  $\beta$  Series убедитесь, что светодиод Power (Питание) горит и что светодиоды ALM не горят. Для усилителей SL Series проверьте дисплей передней панели каждого усилителя на предмет каких-либо ошибок (это отображается миганием всех цифр на дисплее).
6. Если усилитель демонстрирует наличие неисправности (сбоя), то необходимо полностью снять питание с него и с ПЛК и повторить последовательность включения, описанную в разделе 3.
7. При допущении, что ошибки (неисправности) отсутствуют, появляется возможность осуществить толчковое перемещение соответствующих осей. Для ранее описанной конфигурации пуска следует, чтобы устранить любые сбои, индицируемые на DSM314 миганием светодиода STAT, переключить (т.е. сначала включить, затем выключить) бит *Clear Error %Q* (%Q1 для данной конфигурации) в таблице данных ПЛК. Необходимо устранить причины появления сбоя так, чтобы очистить индикатор состояния. Для получения информации по Статусу Ошибок DSM см. Приложение А "Коды Ошибок".

## Толчковое перемещение при помощи Motion Mate DSM314

*Скорость Толчка, Ускорение Толчка и Режим Ускорения Толчка* конфигурируются в модуле DSM314. Эти значения используются всякий раз, когда бит *Jog Plus* или *Jog Minus* %Q переводится в состояние ON (ВКЛ). Обратите внимание, что если оба бита Толчка находятся в состоянии ON, то движение не происходит. Значения по умолчанию для скорости толчка были заданы в ходе конфигурации.

*Jog (Толчок)* может быть выполнен в том случае, если отсутствуют другие команды движения. Для *Jog* не требуется нахождение бита *Enable Drive* %Q в состоянии ON. Включение бита *Jog* %Q приведет к автоматическому включению сервопривода. Будет слышно включение усилителя, а на валу двигателя появится момент вращения. И наоборот, привод автоматически выключится, если выключить бит толчка.

Чтобы выполнить толчок оси в положительном направлении, переключите в ON %Q0021 для оси 1 или %Q0037 для оси 2. Обратите внимание на то, что ссылки, использованные здесь, действуют для ссылок %Q, начинающихся с %Q0001. Начальная ссылка %Q может быть сконфигурирована.

Если толчковое движение двигателя не выполняется, то см. раздел 6 "Поиск и устранение неисправностей Системы Движения".

## Раздел 6: Поиск и устранение неисправностей Системы Движения

### Сигналы тревоги

Первым шагом на пути устранения проблемы является выяснение, имеется ли какое-либо тревожное сообщение. Тревожные сообщения или сообщения об ошибках можно просмотреть в таблице сбоя ПЛК. Тревожные сообщения по подсистемам сервопривода и движения можно просмотреть в слове DSM314 *Module Status Code (Код Статуса Модуля) %AI* или в словах *Axis Error Code (Код Ошибки Оси) %AI*. Дополнительную информацию по ошибкам, проявляющимся через данные %AI, см. в главе 5.

Дополнительную информацию по тревожным сообщениям Motion Mate DSM314 см. в Приложении А "Коды Ошибок", содержащем список кодов тревожных сообщений и их описаний.

Дополнительную информацию по поиску и устранению неисправностей см. в Приложении D "Пуск и Настройка Цифровых или Аналоговых Сервосистем GE Fanuc".

### Установки Конфигурации

Если при подаче питания на систему появляются тревожные сообщения, то это может быть вызвано неправильной установкой при конфигурации.

Конфигурация ПЛК Series 90-30 должна быть сохранена в ЦП ПЛК, а сам ПЛК должен находиться в режиме Run/Output Enabled (Работа / Вывод Включена).

Если не удастся выполнить движение или толчок оси, то проверьте, выполнены ли все условия, необходимые для проведения этих операций. Для этого см. соответствующие разделы данного руководства.

### Техническая Поддержка

#### Веб-сайт GE Fanuc

<http://www.gefanuc.com/support>

#### Телефонные Номера GE Fanuc

Если проблему все же не удастся решить, то можно обратиться по номеру GE Fanuc по месту использования, согласно приведенному ниже списку.

Местонахождение	Номер
Северная Америка, Канада, Мексика (Горячая Линия Технической Поддержки)	Бесплатная Поддержка: 1-800 GE Fanuc Прямой Набор: (780) 420-2000
Латинская Америка (для Мексики см. выше)	Прямой Набор: (780) 420-2000
Франция, Германия, Люксембург, Швейцария и Великобритания	Бесплатная Поддержка: 00800 433 268 23
Италия	Бесплатная Поддержка: 16 77 80 596
Другие страны Европы	352 (72) 79 79 309
Азия / Тихоокеанский бассейн - Сингапур	65-566 4918
Индия	91-80-552 0107

## Раздел 7: Дальнейшие действия

Что делать дальше, если движение оси было выполнено успешно?

Данная глава, посвященная пуску, не перекрывает все аспекты системы управления движением DSM314. Поэтому необходимо ознакомиться с информацией, содержащейся во всех других руководствах (см. "Взаимосвязанные Публикации" в Предисловии к данному руководству). Кроме того, GE Fanuc предлагает соответствующие курсы обучения. Если приложение требует специализированного машинного интерфейса, то необходимо также пройти курс программирования GE Fanuc. Для получения информации по обучению звоните 1-800-GE FANUC или обращайтесь к торговым представителям GE Fanuc.

Глава настоящего руководства, посвященная установке и подключению, дает инструкции по завершению установки аппаратного обеспечения DSM.

Глава о конфигурации дает подробную информацию, требуемую для конфигурации модуля DSM для конкретного приложения. Следует начать с просмотра раздела "*Параметры Конфигурации*" в главе 4.

⇒ . . . . . **Две заключительные рекомендации:**

- Сохраните техническую документацию, поступившую вместе с системой.

Документ "*Важная информация об изделии*" содержит самые последние сведения о данном изделии, которые могут отсутствовать в настоящем руководстве.

- Выполняйте резервное копирование папки управляющей программы.

**Внимание!** . . . . При разработке приложения делайте это достаточно часто.

Раздел 1: Описание аппаратуры

В данном разделе описаны основные свойства аппаратной части модуля. На передней панели модуля находится семь светодиодов индикации состояния, одно гнездо порта связи RJ-11 и четыре пользовательских разъема В/В (36 контактов). Вывод заземления в нижней части модуля обеспечивает удобный способ подключения заземления лицевой панели к земле.

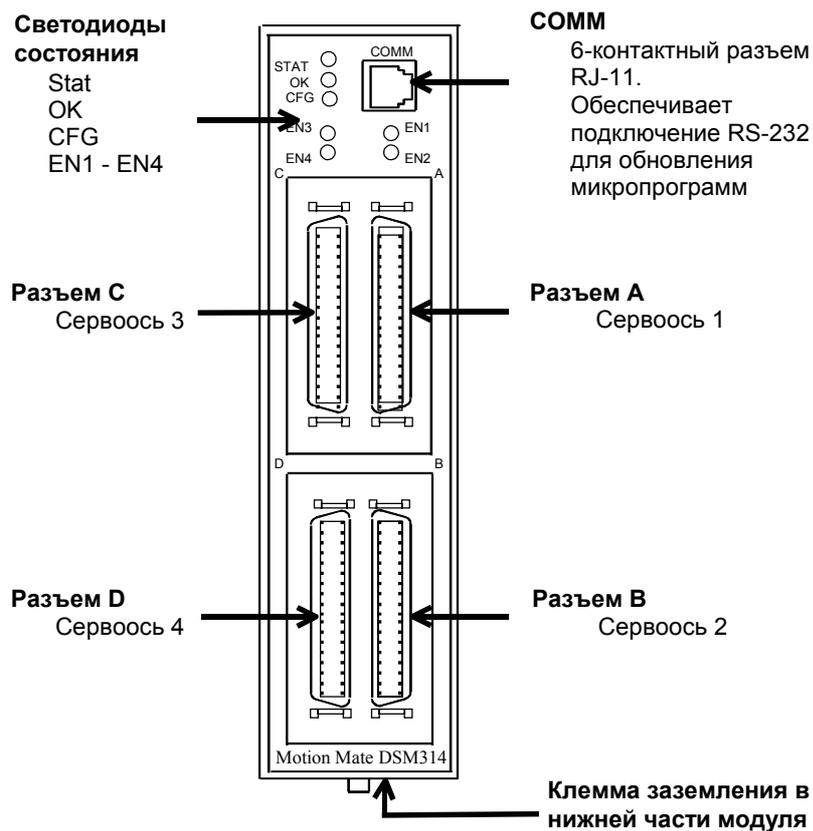


Рис. 3-1. Модуль DSM314

## Светодиодные индикаторы

В модуле DSM314 имеется семь светодиодных индикатора состояния, описанных ниже:

**STAT** Нормально ВКЛ. МИГАЕТ для индикации ошибок при работе. *Медленное* мигание (четыре раза в секунду) – при ошибках только статуса. *Быстрое* мигание (восемь раз в секунду) – при ошибках, которые вызывают прекращение работы сервоуправления.

**ВКЛ.:** Если светодиод постоянно в состоянии ВКЛ., то это значит, что модуль DSM314 функционирует правильно. В нормальной ситуации этот светодиод должен быть всегда в состоянии ВКЛ.

**ВЫКЛ.:** Если светодиод в состоянии ВЫКЛ., то это значит, что модуль DSM314 не функционирует. Это является результатом неисправности аппаратуры или программного обеспечения, которые защищают модуль от перегрузки.

**Мигание:** Если светодиод МИГАЕТ, это означает состояние неисправности (ошибки).

### Постоянная частота мигания, светодиод CFG в состоянии ВКЛ.:

Светодиод мигает медленно (четыре раза в секунду) при ошибках только статуса и быстро (восемь раз в секунду) при ошибках, которые вызывают прекращение работы сервоуправления. Бит статуса *Module Error Present (Имеется ошибка модуля) %I* будет ВКЛ. Код ошибки (шестнадцатеричный формат) помещается в слово *Module Status Code (Код Статуса Модуля) %AI* или в одно из слов *Axis Error Code (Код Ошибки Оси) %AI*.

### Постоянная частота мигания, светодиод CFG мигает:

Если светодиоды STAT и CFG оба мигают **вместе** с постоянной скоростью, то это значит, что модуль DSM314 в режиме начальной загрузки ожидает загрузки новой микропрограммы. Если светодиоды STAT и CFG оба мигают **попеременно** с постоянной скоростью, то это значит, что микропрограмма DSM314 обнаружила истечение времени ожидания при контроле программного обеспечения, как результат неправильного функционирования аппаратуры или программ.

### Меняющаяся нерегулярным образом частота мигания, светодиод CFG в состоянии ВЫКЛ.:

Если это происходит при включении, то это означает, что обнаружена неисправность аппаратуры или ошибка программы. Светодиод STAT на модуле будет мигать, отображая два номера ошибки, разделенных короткой задержкой. Эти номера определяются подсчетом миганий в обеих последовательностях. Запишите эти номера и обратитесь в GE Fanuc за информацией по решению проблемы.

**ОК** Светодиод ОК показывает текущее состояние (статус) модуля DSM314.

**ВКЛ.:** Если светодиод постоянно в состоянии ВКЛ., то это значит, что модуль DSM314 функционирует правильно. В нормальной ситуации этот светодиод должен быть всегда в состоянии ВКЛ.

**ВЫКЛ.:** Если светодиод в состоянии ВЫКЛ., то это значит, что модуль DSM314 не функционирует. Это является результатом неисправности аппаратуры или программного обеспечения, которые защищают модуль от перегрузки.

**CFG** Этот светодиод ВКЛ., когда конфигурация модуля получена от ПЛК.

**EN1** Если этот светодиод ВКЛ., то это значит, что включен выход реле Включения привода оси 1.

**EN2** Если этот светодиод ВКЛ., то это значит, что включен выход реле Включения привода оси 2.

**EN3** Если этот светодиод ВКЛ., то это значит, что включен выход реле Включения привода оси 3.

**EN4** Если этот светодиод ВКЛ., то это значит, что включен выход реле Включения привода оси 4.

### Разъем DSM COMM (последовательные связи)

На передней панели модуля имеется один разъем RJ-11 для последовательной связи, обозначенный как "COMM". Он используется для загрузки обновлений микропрограмм в модуль DSM с персонального компьютера, на котором запущена программная утилита GE Fanuc PC Loader или Win Loader. (Смотри приложение F подробнее).

Этот последовательный COMM-порт подсоединяется к последовательному порту ПК и использует SNP-протокол GE Fanuc и стандарт последовательной связи GE Fanuc RS-232. Скорость в бодах конфигурируется от 300 до 19200 бод. Этот COMM-порт сконфигурирован с использованием программы конфигурации.

Для подключения COMM-порта к персональному компьютеру в комплект поставки от GE Fanuc входит кабель IC693CBL316 длиной 1 метр. У этого кабеля 9-контактный D-разъем гнездового типа со стороны компьютера и 6-контактный разъем RJ-11 со стороны модуля DSM314. **При использовании более длинного кабеля максимальная рекомендуемая длина 50 футов.**

Таблица 3-1. Назначение контактов COMM-порта DSM314

Номер контакта RJ-11	Номер контакта 9-контактного (гнездового) разъема	Обозначение контакта	Описание
1	7	CTS	Готовность к приёму
2	2	TXD	Передача данных
3	5	0V	Сигнальная земля
4	5	0V	Сигнальная земля
5	3	RXD	Приём данных
6	8	RTS	Готовность к передаче

**Примечание:** Контакт 1 находится в нижней части разъема, если смотреть на него со стороны передней панели модуля.

### Разъемы В/В

Модуль DSM314 представляет собой цифровой сервопривод для двух осей / аналоговый интерфейс скорости для одной оси или аналоговый сервоконтроллер (режим управления по моменту вращения или режим управления по скорости); модуль имеет четыре разъема В/В по 36 контактов, маркированных как А, В, С и D. Назначение разъемов следующее:

Таблица 3-2. Назначение разъемов В/В осей

Разъем	Номер оси	Тип оси	Использование В/В
A	1	Сервоось	Цифровое или аналоговое сервоуправление по замкнутому контуру
B	2	Сервоось Вспомогат. ось	Цифровое или аналоговое сервоуправление по замкнутому контуру или обратная связь по положению и вспомогательные аналоговые / цифровые В/В
C	3	Сервоось Вспомогат. ось	Аналоговое сервоуправление по замкнутому контуру или обратная связь по положению и вспомогательные аналоговые / цифровые В/В
D	4	Сервоось Вспомогат. ось	Аналоговое сервоуправление по замкнутому контуру или обратная связь по положению и вспомогательные аналоговые / цифровые В/В

Все четыре разъема обеспечивают схожие аналоговые и цифровые цепи В/В. **Только** Ось 1 и Ось 2 могут быть сконфигурированы так, чтобы управлять цифровыми сервоприводами.

**Если используются цифровые сервоприводы, то как Ось 1, так и Ось 2 должны быть**  
Глава 3 Монтаж и подключение модуля DSM314

сконфигурированы для режима Цифрового сервопривода. Если Ось 1 и Ось 2 сконфигурированы для цифровых сервоприводов, то Ось 3 может быть использована как сервоось аналогового интерфейса скорости или для управления вспомогательной осью. **Если Оси 1 и 2 сконфигурированы для цифровых сервоприводов, то Ось 4 не может быть использована как сервоось аналогового интерфейса скорости, сервоось интерфейса момента вращения или для управления вспомогательной осью.**

Если Ось 1 сконфигурирована для Аналогового сервоуправления (интерфейс момента вращения или интерфейс скорости), то Оси 2 – 4 могут быть использованы для Аналогового сервоуправления (интерфейс момента вращения или интерфейс скорости) или для управления вспомогательной осью. Функции вспомогательной оси включают в себя ввод позиционирования для ведущих осей следящего устройства и генерирование внутренних команд (виртуальный мастер).

Любой из этих четырех разъемов, используемых в системе, обычно подсоединяется кабелем к соответствующей клеммной колодке, используются кабели IC693CBL324 (1 метр) или IC693CBL325 (3 метра). Три различные клеммные колодки обеспечивают подключение внешних устройств к винтовым клеммам. Клеммные колодки описываются далее в разделе "Клеммные колодки" данной главы.

## Подсоединение заземления экрана

Экран передней панели модуля DSM314 должен быть подсоединен к земле рамы. Соединение от модуля DSM314 до заземления рамы может быть выполнено при помощи зеленого заземляющего провода (каталожный номер 44A735970-001R01), поставляемого вместе с модулем. Этот провод имеет вставной разъем на одном конце для подключения к клемме 1/4 дюйма в нижней части модуля DSM314 и клемму с другого конца для подключения к заземленному корпусу.

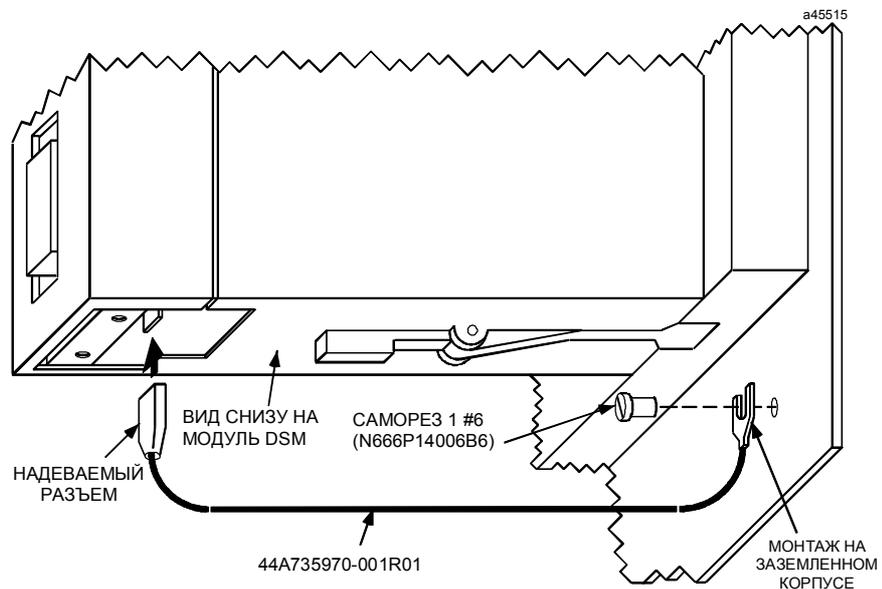


Рис. 3-2. Подключение заземления экрана

## Раздел 2: Установка модуля DSM314

Модуль Motion Mate DSM314 может работать с любой модификацией Series 90-30 или с дистанционной базовой платой (Series 90-30 версия 6.50 или более поздняя). Файл конфигурации, создаваемый программой конфигурации, должен соответствовать физической конфигурации модулей.

**Примечание:** Общие положения по установке и по окружающей среде Series 90-30 см. в "Руководстве по Установке и по Аппаратному обеспечению ПЛК Series 90-30", GFK-0356.

Чтобы установить DSM314 на базовую плату, следует выполнить следующие действия:

1. Используя программу конфигурации или ручной программатор остановить ПЛК. Это предотвратит выдачу программой локального приложения, если таковое имеется, какой-либо команды, которая может вызвать работу модуля при последующем включении питания.
2. Отключите питание с системы ПЛК Series 90-30.
3. Выровняйте модуль по нужному базовому слоту и разъёму. Наклоните модуль вверх так, чтобы верхний задний зацеп модуль вошел в слот на верхнем углу базовой платы.
4. Наклоняйте модуль вниз до тех пор, пока соединительный элемент и рычаг-защелка в нижней части модуля не защелкнутся, войдя в вырез базовой платы.
5. Соедините провод экрана лицевой панели между лепестком 1/4" на нижней части модуля и соответствующей точкой заземления на панели.
6. См. рис. 3-10 – 3-23 и таблицы 3-7 – 3-14 по требованиям к подключению В/В.
7. Подайте питание на крейт ПЛК. После того как контроллер пройдет диагностику включения, загорится светодиод статуса на модуле Motion Mate DSM314.
8. Повторите эту процедуру для каждого модуля DSM314 в данной системе ПЛК.
9. Сконфигурируйте модуль (модули) DSM314 как описано в главе 4.

Приведенная ниже таблица показывает потребление электроэнергии модулей DSM314 и приводит число модулей, которое может быть установлено в конкретной системе ПЛК.

Количество модулей в системе может быть ограничено следующими обстоятельствами:

- Ограничение по мощности для крейта ПЛК
- Место в таблице В/В ПЛК. Модуль DSM требует использования памяти %I, %Q, %AI и %AQ в таблице В/В ПЛК, где обычно более всего из четырех ограничены типы %I и %Q. %AI ограничено также на тех ЦП, которые не поддерживают конфигурируемую память %AQ (такие как ЦП 350). Количество доступной памяти зависит от используемой модели ЦП ПЛК.
- Емкость памяти данных конфигурации ПЛК
- Имеющаяся в распоряжении память ПЛК

Нельзя превышать абсолютные границы для каждого типа ПЛК, поскольку в некоторых случаях они базируются на таблице В/В ПЛК и на емкости памяти данных конфигурации.

Практическое число осей должно учитывать использование В/В и время цикла всей системы.

Таблица 3-3. Максимальное число DSM-модулей для различных типов базовой платы и модулей питания

<p><b>Напряжение источника питания:</b>  <b>Ток, потребляемый модулем DSM от источника питания:</b></p> <p><b>Имеющийся ток от источника +5В / модуль для питания внешнего датчика положения, если он используется:</b></p>	<p>5 В постоянного тока от системной платой ПЛК  800 мА плюс ток питания датчика положения (см. следующий пункт).</p> <p>500 мА (если используется, то должен быть добавлен к потреблению тока от модуля +5В)</p>
<p><b>Модели ПЛК 350, 352, 360, 363, 364:</b>  <b>(базовые платы ЦП на 5 и 10 слотов, платы расширения или дистанционные базовые платы на 5 и 10 слотов – всего 8 базовых плат на систему)</b></p>	<p>2 модуля DSM314 на базовую плату ЦП с PWR321/322/328  5 модулей DSM314 на базовую плату ЦП с PW330/331  3 модуля DSM314 на базовую плату расширения / дистанционную базовую плату с PWR321/322/328  6 модулей DSM314 на дистанционную базовую плату с PW330/331  7 модуля DSM314 на базовую плату расширения с PW330/331  <b>Всего 20 модулей DSM314 на ПЛК-систему с PWR321/322/328*</b>  <b>Всего 20 модулей DSM314 на ПЛК-систему с PWR330/331*</b></p>

**\*Примечание:** Series 90-30 обычно поддерживает 20 модулей DSM314 в системе. Это количество может быть уменьшено при использовании других модулей в системе, таких как модули APM и GBC. Оно дополнительно уменьшается также при наличии установки дейтаграмм, которые считывают таблицы ссылок или сбоев. Если конфигурация и пользовательская программа сохраняются одновременно, то присутствие С-блоков внутри LD-программы или программы С-логики может также повлиять на количество модулей DSM314, которое может быть включено в систему. Если указанное сохранение провести не удалось, то может оказаться возможным ввести заданную конфигурацию в систему, сохранив сначала логическую программу, а затем по отдельному запросу – конфигурацию.

Значения, приведенные в таблице выше, являются теоретическим максимумом. Однако важным фактором определения состава модулей на любой базовой плате является общее потребление мощности всеми модулями. Оно не должно превышать полной допустимой нагрузки источника питания. Вполне возможно, что из-за ограничений источника питания в конкретной ситуации количество модулей будет меньше максимального числа модулей DSM314, которое могло бы быть установлено на базовой плате. Программа конфигурации имеет блок автоматического определения загрузки источника питания, который может быть использован для тестирования. Этот расчет может быть сделан также вручную. Процедура объяснена в "Руководстве по установке ПЛК Series 90-30" (GFK-0356, версия Р или более поздняя), где приведены требования по нагрузке для модулей Series 90-30. Имеются следующие источники питания:

### Стандартные источники питания Series 90-30

- IC693PWR321 – стандартный источник питания постоянного/переменного тока – выдает 15 Вт (3000 мА), +5 В пост. тока
- IC693PWR322 – источник питания с входом постоянного тока 24/48 В – выдает 15 Вт (3000 мА), +5 В пост. тока
- IC693PWR328 – источник питания с входом постоянного тока 48 В – выдает 15 Вт (3000 мА), +5 В пост. тока

### Источники питания Series 90-30 повышенной мощности

- IC693PWR330 – источник питания постоянного/переменного тока повышенной мощности – выдает 30 Вт (6000 мА), +5 В пост. тока
- IC693PWR331 – источник питания повышенной мощности с входом постоянного тока – выдает 30 Вт (6000 мА), +5 В пост. тока

**Примечания:** Стандарты, действующие для изделий, и общие технические характеристики см. в GFK-0867B ("Соглашения, стандарты и общие спецификации, применяемые к изделиям GE Fanuc") или в более поздних изданиях

Инструкции по установке, приведенные в данном руководстве, предназначены для установки, которая не требует специальных процедур в особо шумном или опасном окружении. В случае если необходимо соблюдение более жестких требований при установке (например, действующих для знака CE), смотри GFK-1179, "Требования к установке в соответствии со стандартами".

## Раздел 3: Разводка и подключения Вводов/Выводов

### Типы цепей Ввода/Вывода (В/В)

Каждый из четырех разъемов модуля (разъемы А, В, С и D) обеспечивает работу следующих типов цепей В/В:

- Три дифференциальных / однополярных 5В-входа (IN1-IN3)
- Питание датчика положения 5 В постоянного тока (P5V)
- Один однополярный 5В-вход (IN4)
- Четыре однополярных 5В-цепи входа / выхода (IO5-IO8)
- Три входа 24 В (IN9-IN11)
- Один выход твердотельного реле 24 В 125 мА (OUT1)
- Два дифференциальных 5В-выхода драйвера линии (OUT2-OUT3)
- Один выход твердотельного реле 24 В 30 мА (OUT4)
- Два дифференциальных Аналоговых входа +/- 10 В (AIN1-AIN2)
- Один однополярный Аналоговый выход +/- 10 В (AOUT1)

Не все из этих цепей В/В имеются в распоряжении для пользовательского подсоединения. Некоторые из этих цепей используются для управления цифровым сервоусилителем GE Fanuc. См. дополнительную информацию в таблицах 3-11 – 3-14.

### Клеммные колодки

- **Клеммная колодка оси, каталожный № IC693ACC335** – используется только в цифровом режиме. Она соединяет разъем А или В модуля DSM с цифровым сервоусилителем GE Fanuc серии  $\alpha$  или  $\beta$ . Она имеет также винтовые клеммы для подсоединения устройств В/В. Данная клеммная колодка включает в себя два разъема по 36 контактов. Один подсоединяется к модулю DSM через кабель IC693CBL324/325, а другой подключается к цифровому сервоусилителю GE Fanuc через кабель сервокоманд IC800CBL001 / 002. См. рис. 3-10, 3-16, 3-17 и 3-18.

**Примечание:** Для тех приложений цифрового сервоуправления, для которых не требуется использование разъемов А или В сигналов В/В, разъем DSM может быть подсоединен непосредственно к цифровому сервоусилителю GE Fanuc. Дополнительную информацию см. в разделе 3 "Разводка и подключение Вводов/Выводов" далее в данной главе.

- **Вспомогательная клеммная колодка, каталожный № IC693ACC336** - эта клеммная колодка имеет один разъем на 36 контактов, подключаемый к модулю DSM314. Эта колодка используется в двух случаях (см. рис. 3-10 и 3-11):
  1. Для систем аналогового сервоуправления она подсоединяет к DSM-разъемам А, В, С или D винтовые клеммы для подключения третьего аналогового сервоусилителя и устройств В/В. См. рис. 3-19 – 3-23.
  2. Для вспомогательных осей она подсоединяет к DSM-разъемам А, В, С или D винтовыми клеммами для подключения внешних устройств, таких как стробирующие датчики, переключатели исходной позиции и выключатели выхода за границу. **Примечание:** См. рис. 3-23.
- **Клеммная колодка соединения сервоустройства SL-Series и APM/DSM, каталожный № IC800SLT001** – используется для подсоединения DSM-разъемов А, В, С или D к сервоусилителю аналогового интерфейса скорости, а также предоставляет винтовые клеммы для подключения устройств В/В. Она имеет два разъема. Один подсоединяется к модулю DSM, а другой – к сервоусилителю SL-Series. Дополнительную информацию см. в "Руководстве Пользователя по Сервоустройствам SL Series", GFK-1581.

Таблица быстрого подбора клеммной колодки DSM			
Применение DSM	Разъем DSM	Режим оси DSM	Требуемая клеммная колодка
Подсоединение к цифровому сервоустройству α-Series или β-Series и к устройствам В/В.	А или В	Цифровой	IC693ACC335
Подсоединение непосредственно к цифровому сервоустройству α-Series или β-Series. Подсоединения В/В не требуются.	А или В	Цифровой	Не требуется
Подсоединение к третьей части аналогового сервоустройства и к устройствам В/В.	А, В, С или D	Аналоговый	IC693ACC336
Подсоединение к аналоговым сервоустройствам SL-Series и устройствам В/В.	А, В, С или D	Аналоговый интерфейс скорости	IC800SLT001
Подсоединение В/В вспомогательной оси к DSM-разъему В, С или D.	В, С или D	Аналоговый или Вспомогательный	IC693ACC336

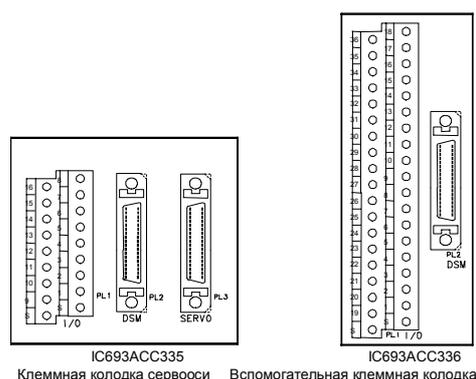


Рис. 3-3. Узлы клеммных колодок, предназначенные для осей и для вспомогательных целей

**Примечание:** Каждая клеммная колодка оборудована монтажным креплением для DIN-шины. В данной главе имеются инструкции по подготовке клеммной колодки для монтажа на панели.

## Клеммная колодка цифровой сервооси - IC693ACC335

### Описание

Клеммная колодка IC693ACC335 цифровой сервооси используется для подключения модуля DSM314 к цифровому сервоусилителю GE Fanuc. Колодка имеет два разъема по 36 контактов, маркированных как **DSM** и **SERVO**. Кабель IC693CBL324 (1 метр) или IC693CBL325 (3 метра) соединяет разъем **DSM** (PL2) с разъемом А или В на лицевой панели DSM324. Кабель сервокоманд IC800CBL001 (1 метр) или IC800CBL002 (3 метра) соединяет разъем **SERVO** (PL3) с разъемом JS1В на цифровом сервоусилителе GE Fanuc  $\alpha$  Series или  $\beta$  Series.

На клеммной колодке цифровой сервооси имеется 18 винтовых клемм для подсоединения пользовательских устройств. Эти клеммы имеют следующее назначение:

Таблица 3-4. Назначение контактов клеммной колодки IC693ACC335 цифровой оси

Винтовая клемма В/В клеммной колодки оси	Контакт лицевой панели DSM314	Идентификатор цепи	Тип цепи	Функция цепи сервооси 1, 2	Название сигнала (указана ось 1)*	Максим. напряжение
1 9	1 19	IN1	Однополярные / дифференц. 5В входы	Вход строба 1 (+) Вход строба 1 (-)	IN1P_A IN1M_A	5 В =
2 10	2 20	IN2		Вход строба 2 (+) Вход строба 2 (-)	IN2P_A IN2M_A	5 В =
3	4	P5V	Питание 5 В	Питание 5 В	P5V_A	5 В =
11	22	0V	0 В	0 В	0V_A	5 В =
6	16	IN9	Оптически изолированные входы 24 В	Перебег (+)	IN9_A	30 В =
14	34	IN10		Перебег (-)	IN10_A	30 В =
7	17	IN11		Переключатель Исходной позиции	IN11_A	30 В =
15	35	INCOM	24 В вход общий	24 В вход общий	INCOM_A	30 В =
8 16	18 36	OUT1	24 В 125 мА Выход DC SSR	ПЛК 24 В выход (+) ПЛК 24 В выход (-)	OUT1P_A OUT1M_A	30 В =
5 13	14 32	OUT3	Дифференциальный Выход 5 В	ПЛК 5 В выход (+) ПЛК 5 В выход (-)	OUT3P_A OUT3M_A	5 В =
4	6	AOUT	+/- 10 В аналоговый выход	Аналоговый выход ПЛК	AOUT_A	5 В =
12	24	ACOM	Аналоговый выход общий	Аналоговый выход общий	ACOM_A	5 В =
S (2 pins)		SHIELD	Экран кабеля	Экран кабеля	SHIELD_A	5 В =

\* Для названия сигналов, относящихся к оси 2, замените все "\_A" на "\_B".

**Шесть варисторов на 130 В (130V MOV) установлено между выбранными точками В/В и экраном (заземление рамы) для подавления шумов. Клеммами В/В, подсоединенными таким образом, являются 6, 7, 8, 14, 15 и 16.**

**Клеммы В/В поддерживают калибр провода 14-28 AWG. Максимальный разрешенный для винтов момент вращения составляет 5 дюймов x фунт.**

**Примечание:** Две винтовые клеммы отмаркированы как S (Shield) для экрана. Короткий провод заземления должен соединить одну из S-клемм непосредственно с землей панели. Кабельные экраны всех экранированных кабелей от пользовательских устройств должны быть присоединены к одной из S-клемм.

Монтажные размеры

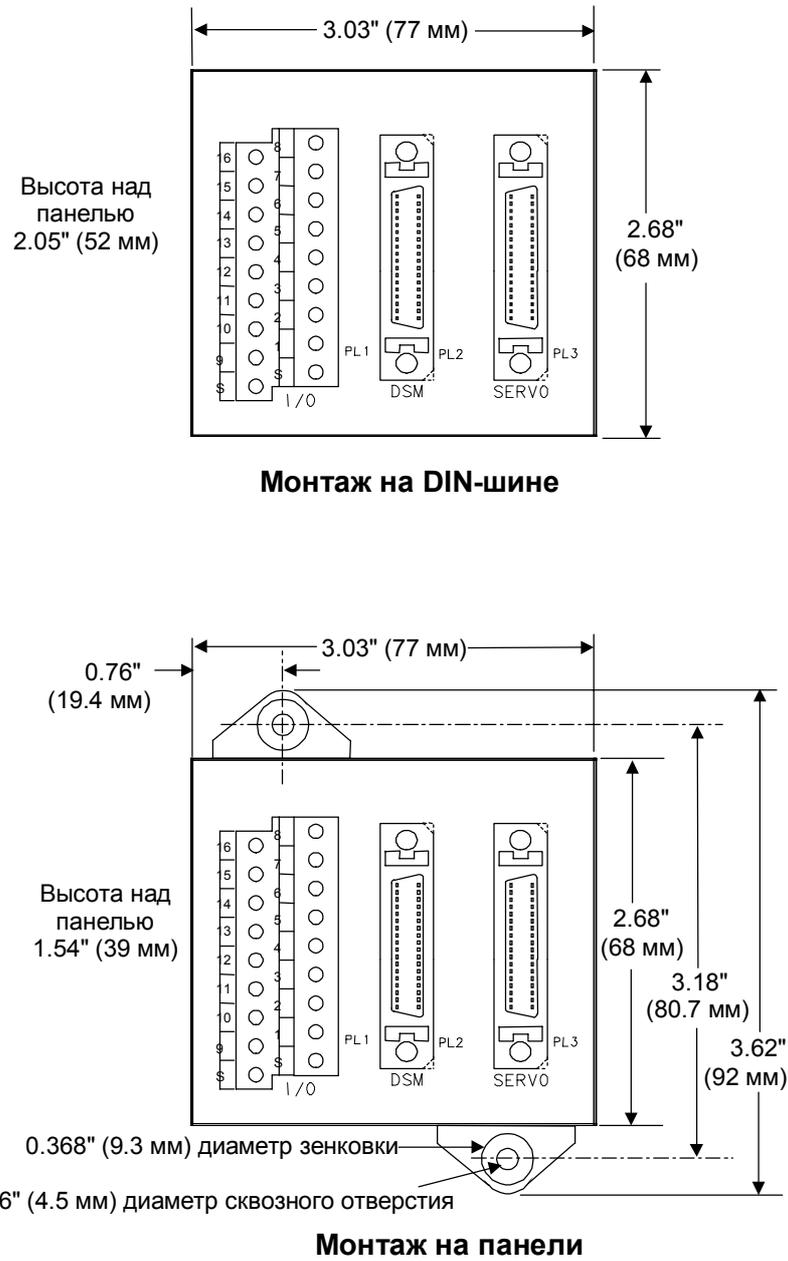


Рис. 3-4. Монтажные размеры клеммной колодки IC693ACC335 цифровой оси

## Переход от монтажа на DIN-шине к монтажу на панели

Приведенные ниже части используются в монтажных узлах при монтаже, как на DIN-шине, так и на панели. Клеммная колодка оси поставляется в конфигурации монтажа на DIN-шине. Указания в данном разделе показывают, что надо сделать, чтобы смонтировать клеммную колодку в панельной конфигурации.

В таблице ниже и на рисунках дано описание различных пластиковых деталей, из которых состоит клеммная колодка, и приведен вид сбоку, сконфигурированную для монтажа на DIN-шине.

Таблица 3-5. Детали клеммной колодки оси

Номер пластиковой детали	Описание	Количество	Тип используемой монтажной конструкции
UMK-BE 45	Основание	1	DIN, панель
UMK-SE 11.25-1	Боковой элемент	2	DIN, панель
UMK-FE	Нижний элемент	2	DIN
UMK-BF*	Монтажный зажим	2	Панель

\* Детали, поставляемые с клеммной колодкой оси для монтажа на панели (дополнительно).

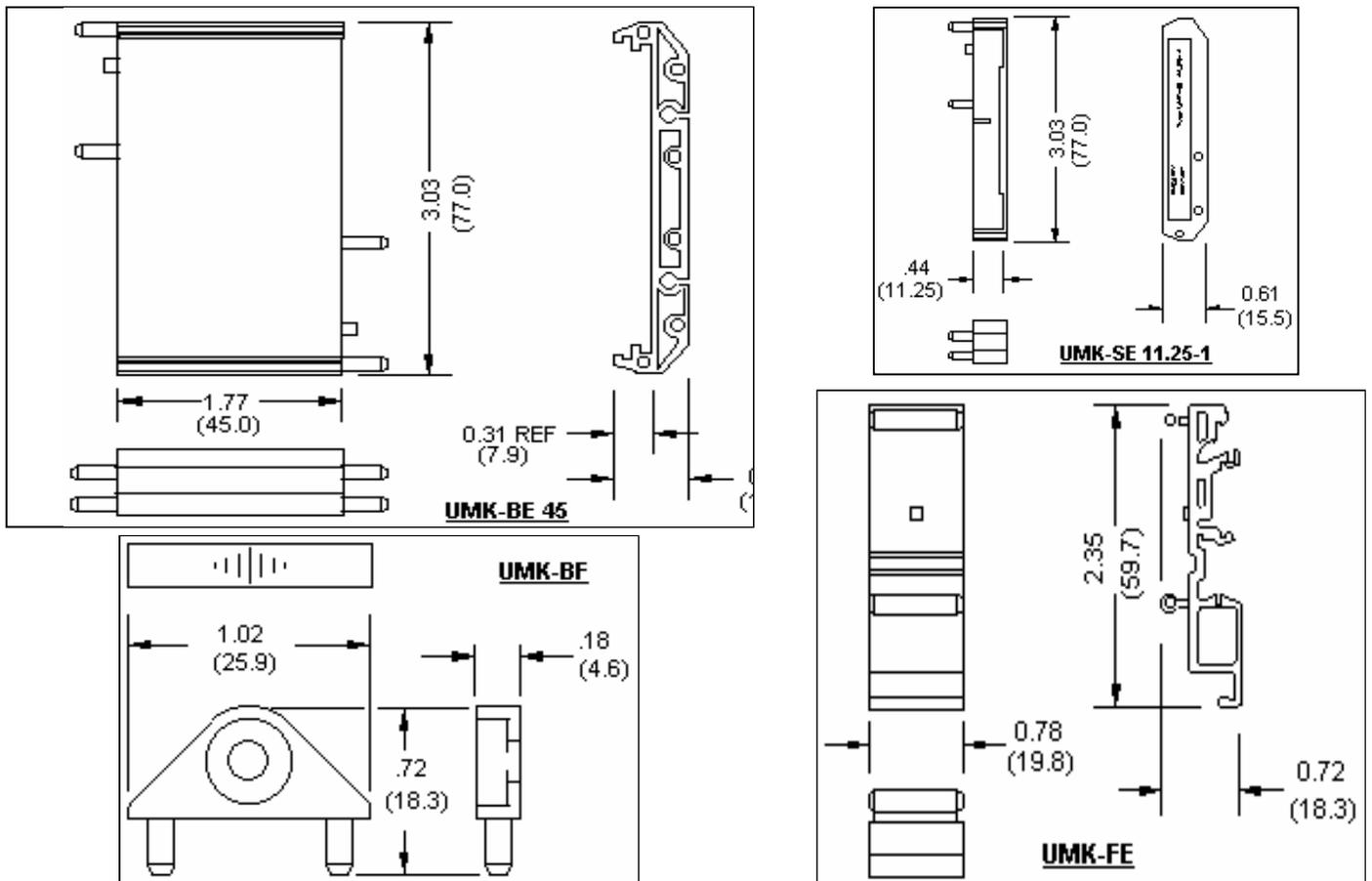
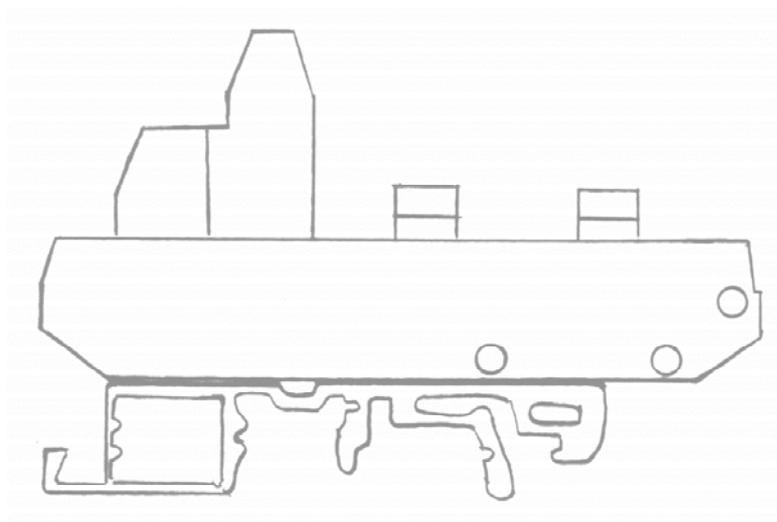


Рис. 3-5. Сборочные чертежи клеммной колодки цифровой сервооси



**Рис. 3-6. Вид сбоку на узел клеммной колодки цифровой сервооси**

Для подготовки клеммной колодки цифровой сервооси для монтажа на панели должны быть проделаны описанные ниже действия. Не забудьте сохранить все снятые части для возможной позднее переделки на монтаж на DIN-шине.

1. Осторожно снимите один боковой элемент UMK-SE 11.25-1 с основания UMK-BE 45. Если используется отвертка или иной инструмент, будьте предельно осторожны, чтобы не повредить пластиковые детали или монтажную плату.
2. Вытяните нижний элемент UMK-FE из основания. Сохраните эту деталь для возможного использования в будущем при переделке клеммной колодки обратно для монтажа на DIN-шине.
3. Защелкните боковой элемент, снятый на этапе 1, обратно на основании.
4. Вставьте один монтажный зажим UMK-BF в соответствующие два отверстия на боковом элементе. Обратите внимание, что монтажный зажим имеет утопленное отверстие, чтобы в него можно было, затем вставить монтажный винт (предоставляет пользователь). Утопленное отверстие должно смотреть вверх, чтобы соответствовать монтажному винту.
5. Повторите вышеприведенные шаги 1 – 4 для другой стороны клеммной колодки.

## Вспомогательная клеммная колодка - IC693ACC336

### Описание и монтажные размеры

Вспомогательная клеммная колодка IC693ACC336 используется для подключения DSM314 к аналоговым сервоосям и к вспомогательным устройствам, таким как инкрементные импульсные датчики положения, стробовые детекторы и внешние переключатели. Колодка имеет один разъем на 36 контактов, маркированный как **DSM**. Кабель IC693CBL324 (1 метр) или IC693CBL325 (3 метра) соединяет разъем **DSM** (PL2) с лицевой панелью DSM324.

На вспомогательной клеммной колодке имеется тридцать восемь клемм для подключения пользовательских устройств. Эти винтовые клеммы имеют ту же маркировку контактов, что и разъем лицевой платы на 36 контактов модуля DSM314. Более детальную информацию по подключению см. в главе 3.

**Максимальное допустимое напряжение на клеммах 16-18 и 34-36 составляет 30 В постоянного тока. Максимальное напряжение для любой другой клеммы входа составляет 5 В пост. тока.**

**Шесть варисторов на 130 В (130V MOV) установлено между выбранными точками В/В и экраном (заземление рамы) для подавления шумов. Клеммами В/В, подсоединенными таким образом, являются 16, 17, 18, 34, 35 и 36.**

**Клеммы В/В поддерживают калибр провода 14-28 AWG. Максимальный разрешенный для винтов момент вращения составляет 5 дюймов х фунт.**

**Примечание:** Две винтовые клеммы отмаркированы как **S** (Shield) для экрана. Короткий провод заземления должен соединить одну из **S**-клемм непосредственно с землей панели. Кабельные экраны всех экранированных кабелей от пользовательских устройств должны быть присоединены к одной из **S**-клемм.

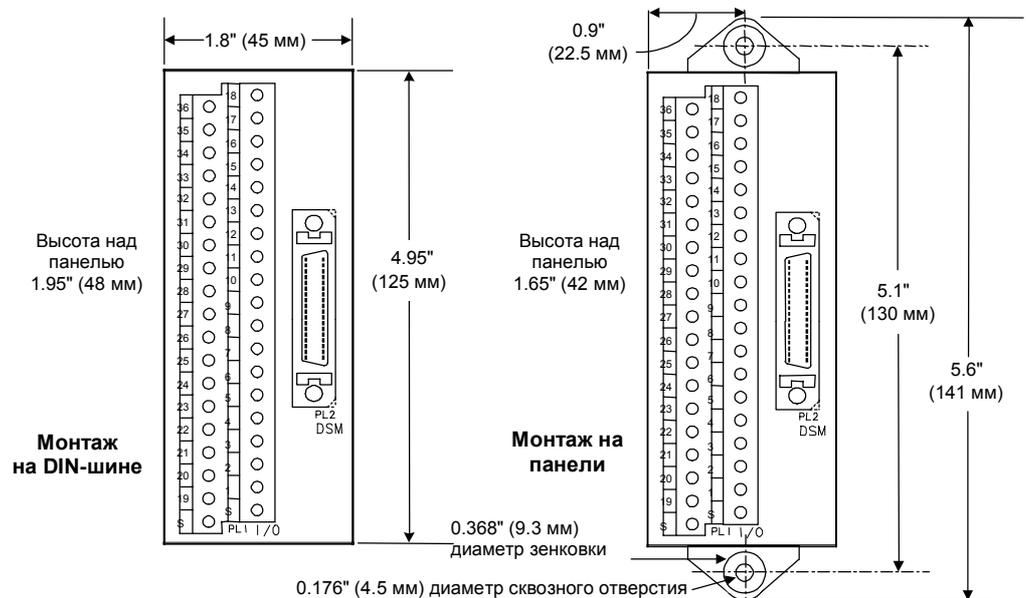


Рис. 3-7. Монтажные размеры клеммной колодки IC693ACC336

### Переход от монтажа на DIN-шине к монтажу на панели

Приведенные ниже части используются в монтажных узлах при монтаже как на DIN-шине, так и на панели. Вспомогательная клеммная колодка поставляется в конфигурации монтажа на DIN-шине. Указания в данном разделе показывают, что надо сделать, чтобы смонтировать клеммную колодку в панельной конфигурации.

В таблице ниже и на рисунках дано описание различных пластиковых деталей, из которых состоит вспомогательная клеммная колодка, и приведен вид сбоку на колодку, сконфигурированную для монтажа на DIN-шине.

Таблица 3-6. Детали вспомогательной клеммной колодки

Контактный номер детали Phoenix	Описание	Количество
UM45 профиль 105.25	Держатель печатной платы	1
UM 45-SEFE с 2 винтами	Боковой элемент с нижней частью	2
UMK 45-SES с 2 винтами*	Боковой элемент	2
UMK-BF*	Монтажный зажим	2

\* Детали, поставляемые с вспомогательной клеммной колодкой для монтажа на панели (дополнительно).

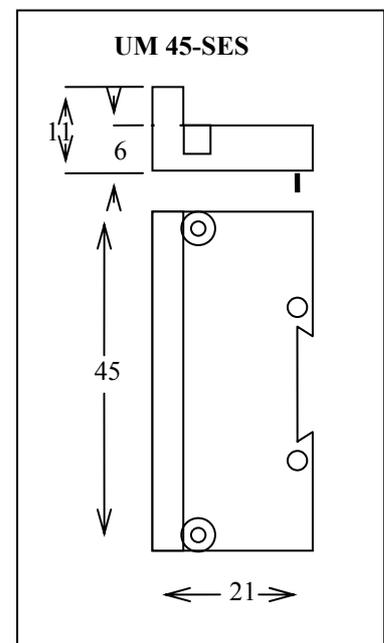
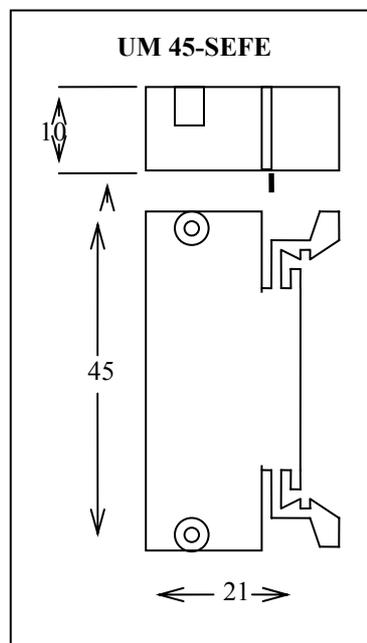
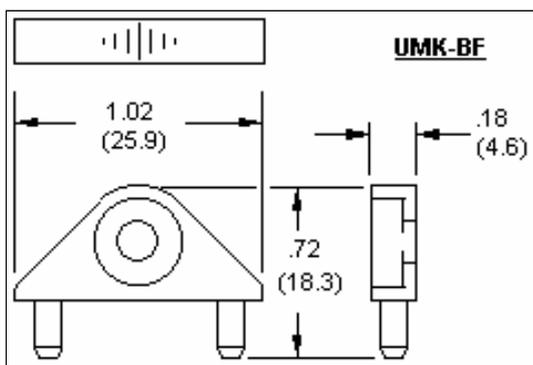
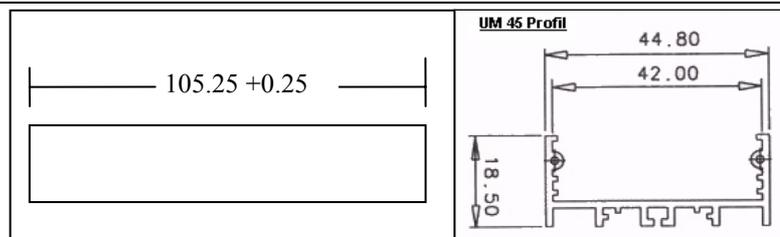
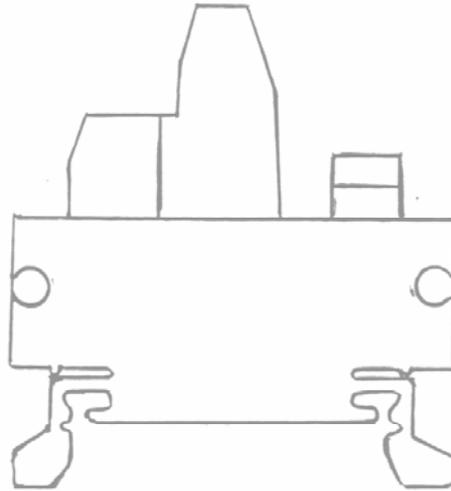


Рис. 3-8. Сборочные чертежи вспомогательной клеммной колодки



**Рис. 3-9. Вид сбоку на вспомогательную клеммную колодку**

Если требуется установить вспомогательную клеммную колодку непосредственно на панель, а не на DIN-шину, необходимо выполнить приведенные ниже действия. Не забудьте сохранить все снятые части для возможной позднее переделки на монтаж на DIN-шине.

1. Используя маленькую простую отвертку Phillips, осторожно удалите два винта, удерживающие один боковой элемент UM-45 SEFE с нижней частью на профильном держателе печатной платы UM 45. Сохраните эту деталь для возможного использования в будущем при переделке клеммной колодки обратно для монтажа на DIN-шине.
2. Закрепите на держателе печатной платы боковой элемент UMK 45-SES вместо того, который был снят на шаге 1; используйте для этого те же два винта. Соблюдайте осторожность, чтобы не перекрутить эти винты.
3. Вставьте один монтажный зажим UMK-BF в соответствующие два отверстия на боковом элементе. Обратите внимание, что монтажный зажим имеет утопленное отверстие, чтобы в него можно было, затем вставить монтажный винт (предоставляется пользователем). Утопленное отверстие должно смотреть вверх, чтобы соответствовать монтажному винту.
4. Повторите вышеприведенные шаги 1 -3 для другой стороны клеммной колодки.

## Кабели

Для модуля DSM314 имеется пять кабелей:

Таблица 3-7. Кабели для DSM314

Кабель	Описание	Длина	Приложение
IC693CBL316	Station Manager Cable (Кабель управления станцией)	1 метр	DSM314 Comm для обновления программ
IC693CBL324	Соединительный кабель клеммной колодки	1 метр	Подключение модуля DSM314 к клеммной колодке сервооси или к вспомогательной клеммной колодке
IC693CBL325	Соединительный кабель клеммной колодки	3 метра	Подключение модуля DSM314 к клеммной колодке сервооси или к вспомогательной клеммной колодке
IC800CBL001	Кабель цифровых сервокоманд	1 метр	Подключение клеммной колодки цифровой сервооси или DSM к цифровому сервоусилителю
IC800CBL002	Кабель цифровых сервокоманд	3 метра	Подключение клеммной колодки цифровой сервооси или DSM к цифровому сервоусилителю

Для заказа специальной клеммной колодки и сервокабелей большей длины обращайтесь к соответствующему представителю GE Fanuc. Максимальная рекомендуемая длина кабеля для подключения DSM к сервоусилителю серии  $\alpha$  и  $\beta$  составляет 50 метров.

**Кабели имеют специальную конструкцию и экранирования, чтобы обеспечить надежность команд сервоуправления. GE Fanuc рекомендует пользователям не пытаться проводить какие-либо изменения кабелей или разъемов на месте.**

**Примечание:** Если цифровая сервоось не использует устройства, которые обычно подключаются к винтовым клеммам клеммной колодки цифрового сервоуправления IC693ACC335, то клеммная колодка и ее кабель IC693CBL324/325 не требуются. Вместо этого кабель цифровых сервокоманд IC800CBL001/002 может быть подключен непосредственно от цифрового сервоусилителя к разъему A или B на лицевой панели DSM314. Если это сделано, то в программе конфигурации конфигурационный параметр **OT Limit Sw** должен быть установлен в положение **Disabled (Выключено)**; иначе DSM не будет работать.

Рисунок ниже показывает клеммную колодку цифровой сервооси и кабели, связанные с модулем DSM314.

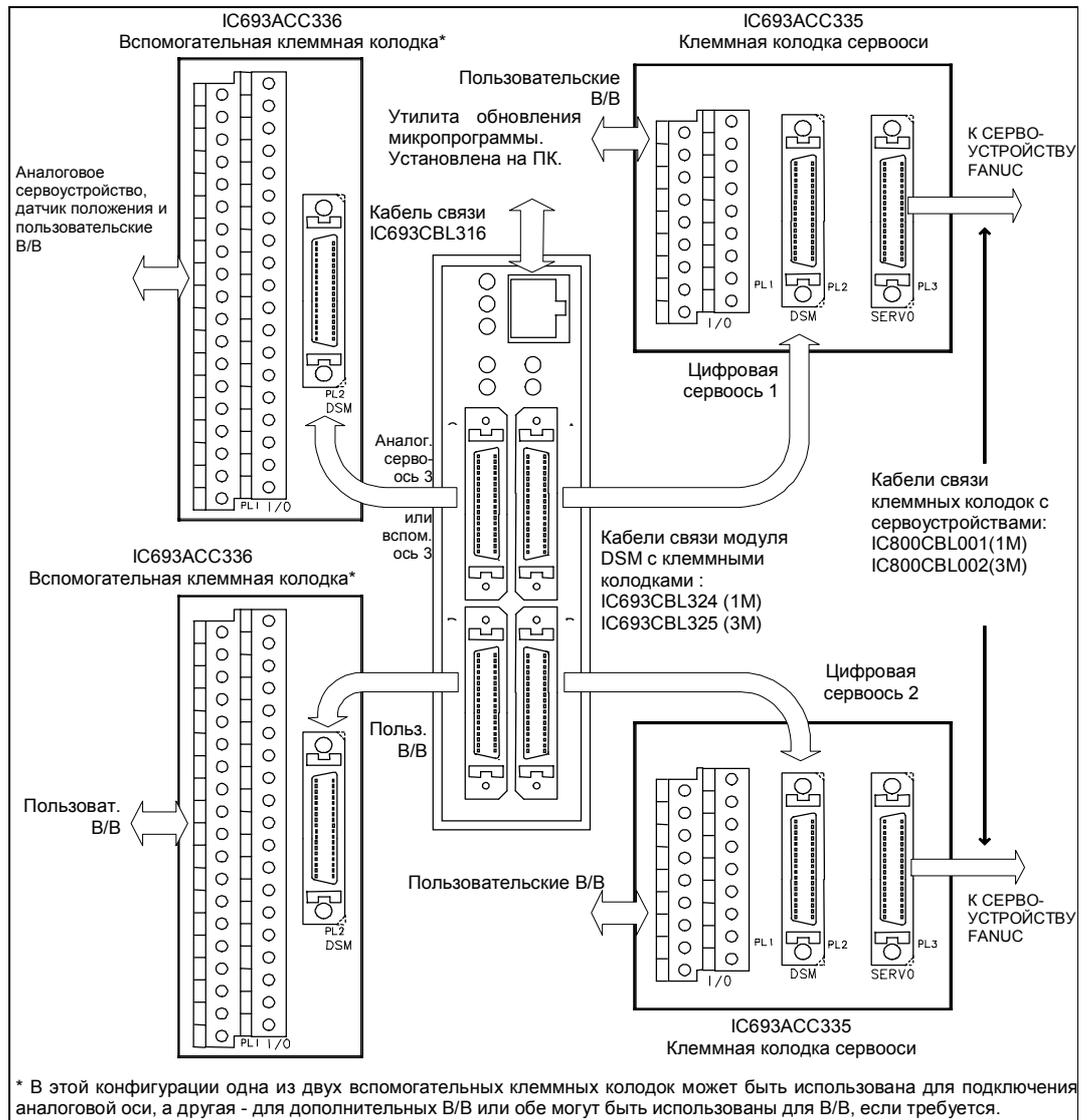


Рис. 3-10. Модуль DSM314: клеммные колодки цифрового сервоустройства и подключения

Рисунок ниже показывает клеммную колодку аналогового сервоустройства и кабели, связанные с модулем DSM314.

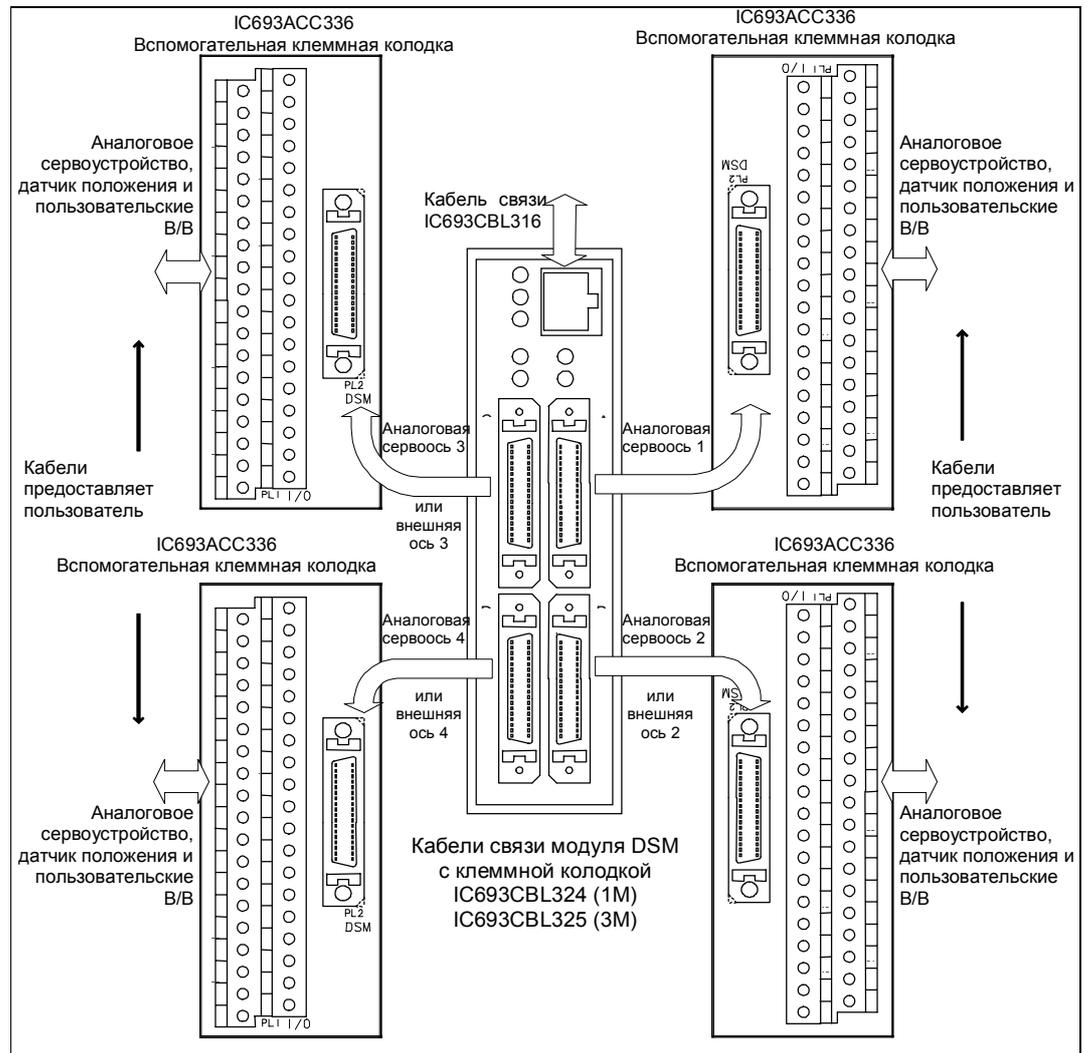


Рис. 3-11. Модуль DSM314: клеммные колодки и подключения для сторонних аналоговых сервоустройств (см. GFK-1581 для SL-сервоустройств)

## Заземление кабеля В/В

Правильная прокладка сигнальных кабелей, кабелей питания усилителя и силовых кабелей двигателя наряду с устройством заземления класса 3 в состоянии обеспечить надежную работу. Обычно заземление класса 3 подразумевает проводник заземления с минимальным диаметром не менее диаметра входного силового кабеля, соединенный через сопротивление не более 100 ом с землей. Необходимо просмотреть правила, действующие на месте использования, и выполнить работу в соответствии с ними.

Технические характеристики для установки и подключения цифровых сервоусилителей  $\alpha$  и  $\beta$  Series, в т.ч. их заземление, полностью описаны в руководстве "*Справочнике по Техническим Характеристикам Сервоустройств*", GFH-001.

При прокладке сигнальных кабелей, входных кабелей питания усилителя и силовых кабелей двигателя необходимо разделить сигнальные от силовых кабелей. Приведенная ниже таблица показывает, как это должно быть сделано.

**Таблица 3-8. Отделение сигнальных линий**

Группа	Сигнал	Действие
A	Вход питания усилителя Питание двигателя Катушка пуска главного контактора управления (MCC) Вход питания усилителя переключения MCC.	Должно находиться на расстоянии не менее 10 см от сигналов группы "B" в виде отдельного жгута или должен быть поставлен экран (заземленная стальная пластина). Для MCC используйте защиту от шумов.
B	Кабель от DSM до клеммной колодки оси. Кабель от клеммной колодки оси до усилителя. Кабель от DSM до вспомогательной клеммной колодки. Кабель обратной связи датчика положения	Должно находиться на расстоянии не менее 10 см от сигналов группы "A" в виде отдельного жгута или должен быть поставлен экран (заземленная стальная пластина). Используйте все требующиеся заземления экранов отдельных кабелей и подключения шины заземления.

### Кабель от DSM до цифрового сервоусилителя $\alpha$ или $\beta$ Series – заземление сигнального кабеля

Сигнальные кабели, используемые вместе с модулем DSM314, имеют экраны, которые должны быть правильно заземлены, чтобы обеспечить надежную работу. На рисунке внизу показаны рекомендации по заземлению кабеля при стандартной установке. Необходимо обратить внимание на следующее:

1. Провод заземления лицевой панели модуля DSM314 должен быть подключен к надежной точке заземления панели.
2. Клеммная колодка цифровой сервооси и вспомогательная клеммная колодка имеют по две винтовых клеммы, маркированных как S. Короткий провод заземления должен соединить одну из S-клемм с надежной землей панели.

Кабелю обратной связи датчика положения, подходящему к цифровому сервоусилителю  $\alpha$  и  $\beta$  Series, требуется зажим заземления экрана кабеля A99L-0035-0001 и один из 11

имеющихся слотов на шине заземления 44В295864-001 на стороне усилителя. Эта конструкция с зажимом работает как на снятие механических нагрузок, так и как заземление кабеля. Внешняя изоляция кабеля цифрового сервоусилителя должна быть снята, чтобы привести кабель в контакт с контактной поверхностью зажима.

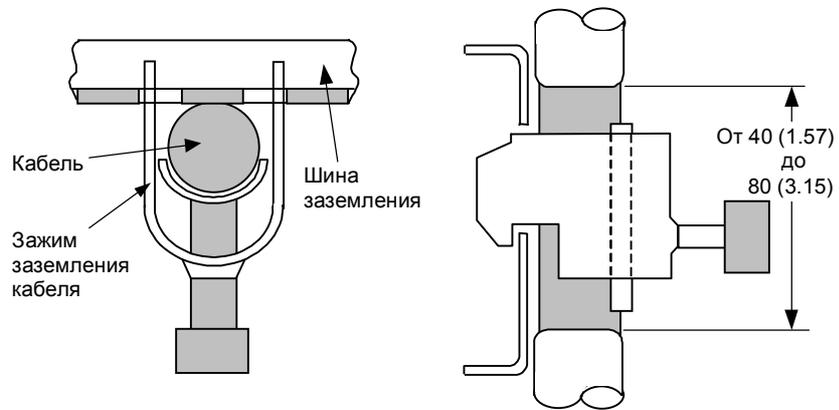


Рис. 3-12. Конструкция зажима для заземления кабеля A99L-0035-0001

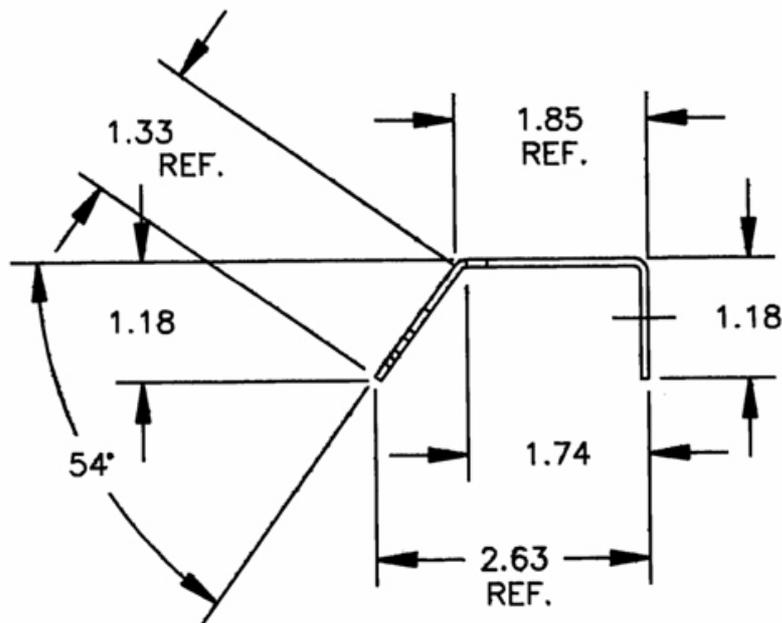


Рис. 3-13. Шина заземления 44В295864-001, размеры, вид сбоку

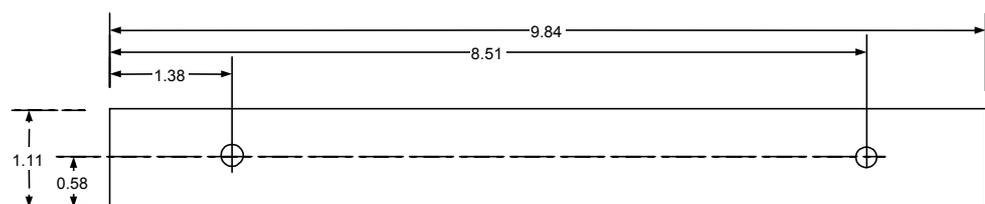


Рис. 3-14. Размеры шины заземления 44В295864-001, вид сзади с монтажными отверстиями

3. Для монтажа, который должен соответствовать стандартам ИЕС по электрической помехозащищенности, зажим заземления экрана кабеля A99L-0035-0001 и один из имеющихся в наличии 11 слотов на шине заземления 44B295864-001, и кабель сервоусилителя IC800CBL001/002 со стороны клеммной колодки цифровой сервооси. Если кабель цифрового сервоусилителя подключается непосредственно к лицевой панели DSM314 (не используется клеммная колодка цифровой сервооси), то не требуется заземление кабеля со стороны лицевой панели .

Дополнительную информацию см. "Требования к установке в соответствии со стандартами", GFK-1179.



Рис. 3-15. Модуль DSM314: заземление кабеля В/В

## Идентификаторы цепей В/В и обозначения сигналов

Идентификаторы цепей В/В обеспечивают последовательный подход к обозначению цепей В/В. Например, IN1 относится к первому из трех дифференциальных / однополярных 5В-входов для каждой оси.

Обозначения сигналов назначаются идентификаторам цепей для каждой оси. *Обозначение сигнала состоит из идентификатора цепи с прибавлением индекса A-D для идентификации разъема оси.* Дифференциальные цепи имеют также индексы P (положительный) и M (минус), чтобы идентифицировать сигнал (+) и (-) для каждой дифференциальной пары.

*Пример:* OUT2 является идентификатором цепи для первого дифференциального 5В-выхода на каждом разъеме. Обозначения сигналов, связанных с цепью OUT2, являются следующими:

Таблица 3-9. Обозначения сигналов, связанных с OUT2

Ось:	Ось 1	Ось 2	Ось 3	Ось 4
Разъем:	A	B	C	D
(+) выходной сигнал:	OUT2P_A	OUT2P_B	OUT2P_C	OUT2P_D
(-) выходной сигнал:	OUT2M_A	OUT2M_B	OUT2M_C	OUT2M_D

## Функции цепей В/В и назначения контактов

Следующие три таблицы показывают функциональное назначение цепей В/В, а также назначение контактов разъема и клеммной колодки для каждого разъема оси. Хотя каждый разъем имеет одинаковые цепи В/В, их функциональное назначение зависит от оси:

Таблица 3-10. Назначение и функции разъемов осей

Разъем	Номер оси	Тип оси	Использование В/В
A	1	Сервоось	Цифровое / аналоговое сервоуправление по замкнутому контуру и пользовательские В/В
B	2	Сервоось	Цифровое / аналоговое сервоуправление по замкнутому контуру или вспомогательные аналоговые и цифровые В/В
C	3	Сервоось	Аналоговое сервоуправление по замкнутому контуру или вспомогательные аналоговые и цифровые В/В
D	4	Сервоось	Аналоговое сервоуправление по замкнутому контуру или вспомогательные аналоговые и цифровые В/В

## Назначение цепей и контактов для цифровых сервоосей 1, 2

Данная таблица представляет все назначения цепей и контактов для цифровых сервоосей 1 и 2. Затененные области показывают сигналы, которые подходят к сервоусилителю и не предназначены для подключений пользователя.

Таблица 3-11. Назначение цепей и контактов для цифровых сервоосей 1 и 2

Обозначение цепи	Тип цепи	Функция цепей сервоосей 1, 2	Обозначение сигнала оси 1	Обозначение сигнала оси 2	Контакт лицевой панели	Клемма клеммной колодки оси
IN1	Однополярные / дифференциальные 5В входы	Вход строба 1 (+)	IN1P_A	IN1P_B	1	1
		Вход строба 1 (-)	IN1M_A	IN1M_B	19	9
IN2		Вход строба 2 (+)	IN2P_A	IN2P_B	2	2
		Вход строба 2 (-)	IN2M_A	IN2M_B	20	10
IN3		Данные последовательного датчика положения (+)	IN3P_A	IN3P_B	3	
			Данные последовательного датчика положения (-)	IN3M_A	IN3M_B	21
P5V	Питание 5 В	Питание 5 В	P5V_A	P5V_B	4	3
0V	0 В	0 В	0V_A	0V_B	22,23	11
IN4	Однополярный 5В-вход	Вход готовности сервоуправления	IN4_A	IN4_B	5	
IO5	Однополярные 5В-входы / выходы	Серво ШИМ / тревога	IO5_A	IO5_B	9	
IO6		Серво ШИМ / тревога	IO6_A	IO6_B	10	
IO7		Серво ШИМ / тревога	IO7_A	IO7_B	11	
IO8		Серво ШИМ / тревога	IO8_A	IO8_B	12	
0V	0 В	0 В	0V_A	0V_B	27-30	
IN9	Оптически изолированные входы 24 В	Перебег (+)	IN9_A	IN9_B	16	6
IN10		Перебег (-)	IN10_A	IN10_B	34	14
IN11		Переключатель Исходной позиции	IN11_A	IN11_B	17	7
INCOM	24 В вход общий	24 В вход общий	INCOM_A	INCOM_B	35	15
OUT1	24 В, 125 мА DC SSR выход	ПЛК 24 В выход (+)	OUT1P_A	OUT1P_B	18	8
		ПЛК 24 В выход (-)	OUT1M_A	OUT1M_B	36	16
OUT2	Дифференциальные 5В-выходы	Запрос последовательного датчика положения (+)	OUT2P_A	OUT2P_B	13	
		Запрос последовательного датчика положения (-)	OUT2M_A	OUT2M_B	31	
OUT3		ПЛК 5В-выход (+)	OUT3P_A	OUT3P_B	14	5
		ПЛК 5В-выход (-)	OUT3M_A	OUT3M_B	32	13
ENBL	24 В, 30 мА SSR выход	Серво MCON (+)	ENBL1_A	ENBL1_B	15	
		Серво MCON 0 В	ENBL2_A	ENBL2_B	33	
AIN1	Дифференциаль-	IR ток фазы (+)	AIN1P_A	AIN1P_B	7	

	ные +/- 10 В	IR ток фазы (-)	AIN1M_A	AIN1M_B	25	
AIN2	аналоговые входы	IS ток фазы (+)	AIN2P_A	AIN2P_B	8	
		IS ток фазы (-)	AIN2M_A	AIN2M_B	26	
AOUT1	+/- 10 В аналоговый выход	Аналоговый выход ПЛК	AOUT_A	AOUT_B	6	4
ACOM	Аналоговый выход общий	Аналоговый выход общий	ACOM_A	ACOM_B	24	12
SHIELD	Экран кабеля	Экран кабеля	SHIELD_A	SHIELD_B		S

## Назначение цепей и контактов для аналоговых сервоосей 1-4

Данная таблица показывает назначение всех цепей и контактов для аналоговых сервоосей 1 – 4. Затененные области обозначают сигналы, которые не используются и не предназначены для пользовательских подключений.

Таблица 3-12. Назначение цепей и контактов для аналоговых сервоосей 1 - 4

Обозначение цепи	Тип цепи	Функция цепей аналоговых сервоосей 1 - 4	Обозначение сигнала оси 1	Обозначение сигнала оси 2	Обозначение сигнала оси 3	Обозначение сигнала оси 4	Контакт лицевой панели	Клемма вспомогательной клеммной колодки
IN1	Однополярные / дифференциальные 5В входы	Канал датчика положения А (+)	IN1P_A	IN1P_B	IN1P_C	IN1P_D	1	1
		Канал датчика положения А (-)	IN1M_A	IN1M_B	IN1M_C	IN1M_D	19	19
Канал датчика положения В (+)		IN2P_A	IN2P_B	IN2P_C	IN2P_D	2	2	
Канал датчика положения В (-)		IN2M_A	IN2M_B	IN2M_C	IN2M_D	20	20	
IN3		Маркер датчика положения (+)	IN3P_A	IN3P_B	IN3P_C	IN3P_D	3	3
		Маркер датчика положения (-)	IN3M_A	IN3M_B	IN3M_C	IN3M_D	21	21
P5V	Питание 5 В	Питание 5 В датчика положения	P5V_A	P5V_B	P5V_C	P5V_D	4	4
0V	0 В	0 В	0V_A	0V_B	0V_C	0V_D	22,23	22,23
IN4	Однополярный 5В-вход	Вход готовности сервоуправления	IN4_A	IN4_B	IN4_C	IN4_D	5	5
IO5	Однополярные 5В-входы / выходы	Вход строба 1	IO5_A	IO5_B	IO5_C	IO5_D	9	9
IO6		Вход строба 2	IO6_A	IO6_B	IO6_C	IO6_D	10	10
IO7		Не используется	IO7_A	IO7_B	IO7_C	IO7_D	11	11
IO8		Не используется	IO8_A	IO8_B	IO8_C	IO8_D	12	12
0V	0 В	0 В	0V_A	0V_B	0V_C	0V_D	27-30	27-30
IN9	Оптически изолированные входы 24 В	Перебег (+)	IN9_A	IN9_B	IN9_C	IN9_D	16	16
IN10		Перебег (-)	IN10_A	IN10_B	IN10_C	IN10_D	34	34
IN11		Датчик начальной позиции	IN11_A	IN11_B	IN11_C	IN11_D	17	17
INCOM	24 В вход общий	24 В вход общий	INCOM_A	INCOM_B	INCOM_C	INCOM_D	35	35
OUT1	24 В, 125 мА Выход DC SSR	ПЛК 24 В выход (+)	OUT1P_A	OUT1P_B	OUT1P_C	OUT1P_D	18	18
		ПЛК 24 В выход (-)	OUT1M_A	OUT1M_B	OUT1M_C	OUT1M_D	36	36
OUT2	Дифференциальные 5В-выходы	Не используется	OUT2P_A	OUT2P_B	OUT2P_C	OUT2P_D	13	13
		Не используется	OUT2M_A	OUT2M_B	OUT2M_C	OUT2M_D	31	31
ПЛК 5 В выход (+)		OUT3P_A	OUT3P_B	OUT3P_C	OUT3P_D	14	14	
ПЛК 5 В выход (-)		OUT3M_A	OUT3M_B	OUT3M_C	OUT3M_D	32	32	
ENBL	24 В, 30 мА Выход SSR	Серво вкл. (+)	ENBL1_A	ENBL1_B	ENBL1_C	ENBL1_D	15	15
		Серво вкл. (-)	ENBL2_A	ENBL2_B	ENBL2_C	ENBL2_D	33	33
AIN1	Дифференциальные +/- 10 В аналоговые входы	Аналоговый вход ПЛК (+)	AIN1P_A	AIN1P_B	AIN1P_C	AIN1P_D	7	7
		Аналоговый вход ПЛК (-)	AIN1M_A	AIN1M_B	AIN1M_C	AIN1M_D	25	25
Аналоговый вход ПЛК (+)		AIN2P_A	AIN2P_B	AIN2P_C	AIN2P_D	8	8	
Аналоговый вход ПЛК (-)		AIN2M_A	AIN2M_B	AIN2M_C	AIN2M_D	26	26	

AOUT1	+/- 10 В аналоговый выход	Сервокоманда скорости (+) или сервокоманда момента вращения (+)	AOUT_A	AOUT_B	AOUT_C	AOUT_D	6	6
ACOM	Аналоговый выход общий	Сервокоманда скорости общая или момент вращения общий	ACOM_A	ACOM_B	ACOM_C	ACOM_D	24	24
SHIELD	Экран кабеля	Экран кабеля	SHIELD_A	SHIELD_B	SHIELD_C	SHIELD_D		S

## Назначение цепей и контактов вспомогательных осей 2-4

Данная таблица показывает назначение всех цепей и контактов для вспомогательных осей 2 – 4. Затененные области обозначают сигналы, которые не используются и не предназначены для пользовательских подключений.

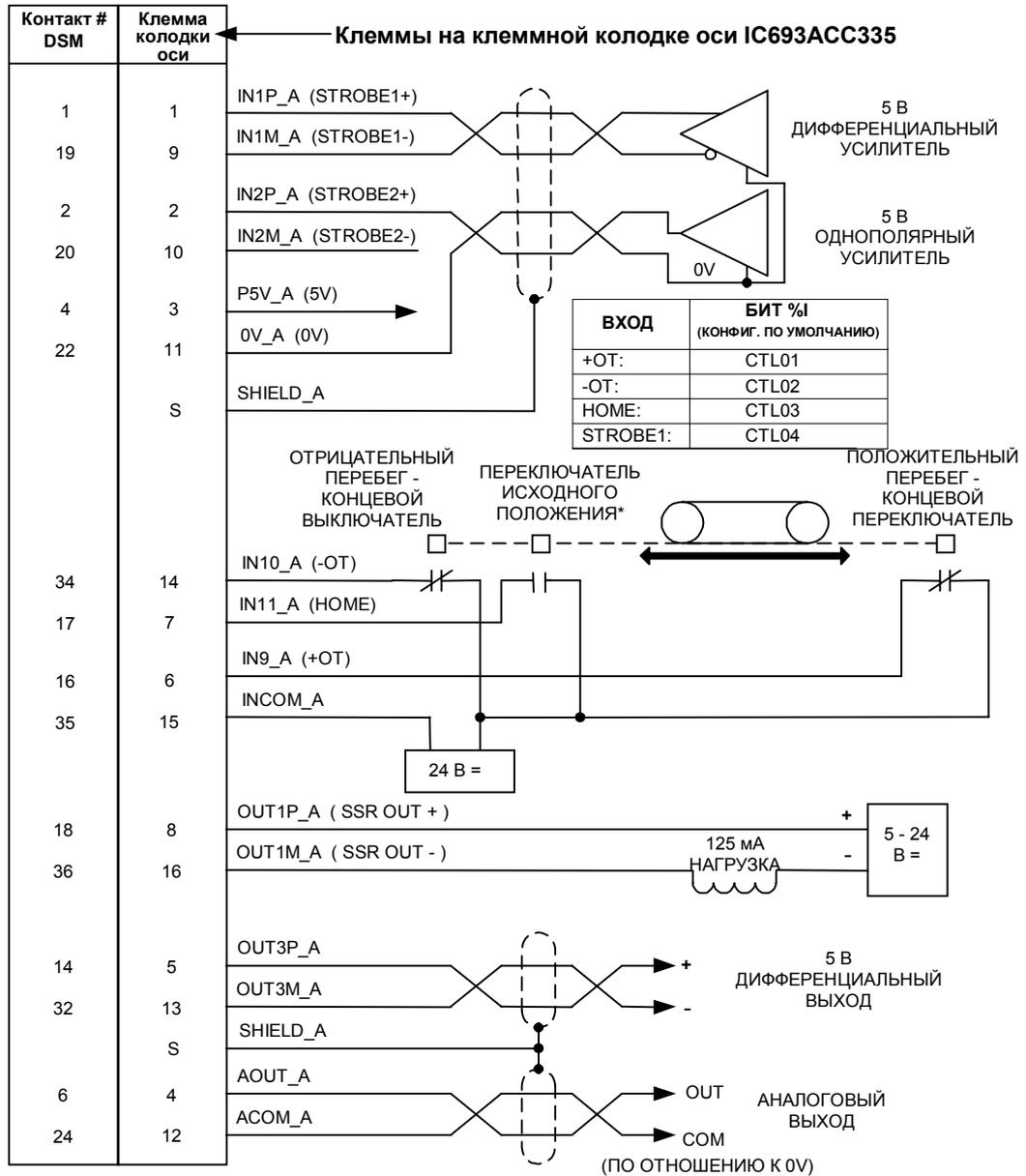
Таблица 3-13. Назначения цепей и контактов для вспомогательной оси 3 (разъем С)

Обозначение цепи	Тип цепи	Функция цепи вспомогательных осей 2 - 4	Обозначение сигнала оси 2	Обозначение сигнала оси 3	Обозначение сигнала оси 4	Контакт лицевой панели	Клемма вспомогательной клеммной колодки	
IN1	Однополярные / дифференциальные 5В входы	Канал датчика положения А (+)	IN1P_B	IN1P_C	IN1P_D	1	1	
		Канал датчика положения А (-)	IN1M_B	IN1M_C	IN1M_D	19	19	
Канал датчика положения В (+)		IN2P_B	IN2P_C	IN2P_D	2	2		
Канал датчика положения В (-)		IN2M_B	IN2M_C	IN2M_D	20	20		
IN3		Маркер датчика положения (+)	IN3P_B	IN3P_C	IN3P_D	3	3	
		Маркер датчика положения (-)	IN3M_B	IN3M_C	IN3M_D	21	21	
P5V		5 В от ПЛК	Питание 5 В датчика положения	P5V_B	P5V_C	P5V_D	4	4
0V		0 В	0 В	0V_B	0V_C	0V_D	22,23	22,23
IN4		Однополярный 5В-вход	Вход 5 В ПЛК	IN4_B	IN4_C	IN4_D	5	5
IO5	Однополярные 5В-входы / выходы	Вход строба 1	IO5_B	IO5_C	IO5_D	9	9	
IO6		Вход строба 2	IO6_B	IO6_C	IO6_D	10	10	
IO7		Не используется	IO7_B	IO7_C	IO7_D	11	11	
IO8		Не используется	IO8_B	IO8_C	IO8_D	12	12	
0V	0 В	0 В	0V_B	0V_C	0V_D	27-30	27-30	
IN9	Оптически изолированные входы 24 В	Вход 24 В ПЛК	IN9_B	IN9_C	IN9_D	16	16	
IN10		Вход 24 В ПЛК	IN10_B	IN10_C	IN10_D	34	34	
IN11		Датчик начальной позиции	IN11_B	IN11_C	IN11_D	17	17	
INCOM	24 В вход общий	24 В вход общий	INCOM_B	INCOM_C	INCOM_D	35	35	
OUT1	24 В, 125 мА Выход DC SSR	ПЛК 24 В выход (+)	OUT1P_B	OUT1P_C	OUT1P_D	18	18	
		ПЛК 24 В выход (-)	OUT1M_B	OUT1M_C	OUT1M_D	36	36	
OUT2	Дифференциальные 5В-выходы	Не используется	OUT2P_B	OUT2P_C	OUT2P_D	13	13	
		Не используется	OUT2M_B	OUT2M_C	OUT2M_D	31	31	
ПЛК 5 В выход (+)		OUT3P_B	OUT3P_C	OUT3P_D	14	14		
ПЛК 5 В выход (-)		OUT3M_B	OUT3M_C	OUT3M_D	32	32		
ENBL	24 В, 30 мА Выход SSR	ВКЛ., когда действует команда Включить Аналоговый выход %AQ	ENBL1_B ENBL2_B	ENBL1_C ENBL2_C	ENBL1_D ENBL2_D	15 33	15 33	
		Аналоговый вход ПЛК (+)	AIN1P_B	AIN1P_C	AIN1P_D	7	7	
AIN1	Дифференциальные +/- 10 В аналоговые входы	Аналоговый вход ПЛК (-)	AIN1M_B	AIN1M_C	AIN1M_D	25	25	
AIN2		Аналоговый вход ПЛК (+)	AIN2P_B	AIN2P_C	AIN2P_D	8	8	

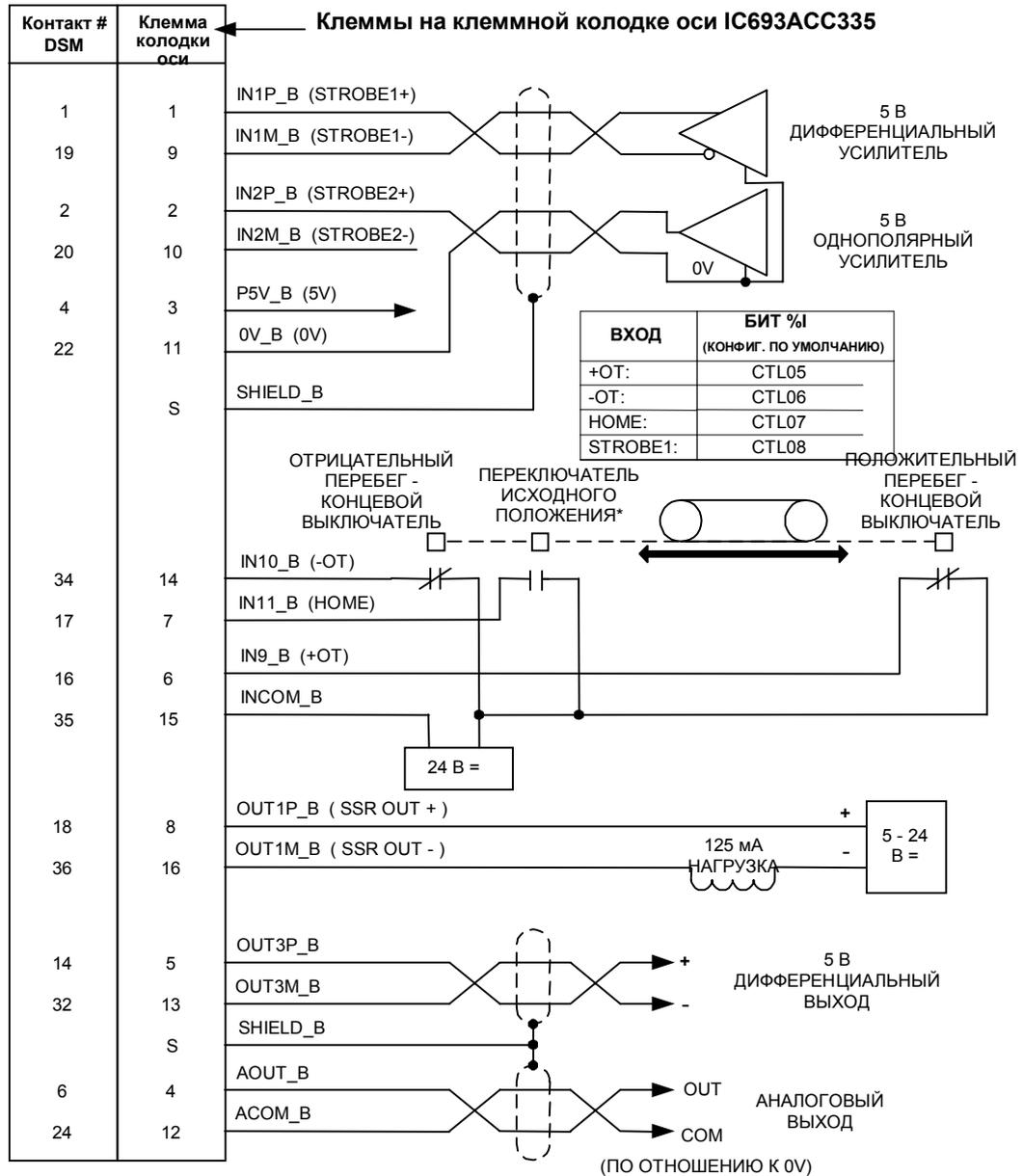
		Аналоговый вход ПЛК (-)	AIN2M_B	AIN2M_C	AIN2M_D	26	26
AOUT1	+/- 10 В аналоговый выход	Аналоговый выход ПЛК	AOUT_B	AOUT_C	AOUT_D	6	6
ACOM	Аналоговый выход общий	Аналоговый выход общий	ACOM_B	ACOM_C	ACOM_D	24	24
SHIELD	Экран кабеля	Экран кабеля	SHIELD_B	SHIELD_C	SHIELD_D		S

### Схемы подключения В/В

Приведенные ниже схемы показывают типичные пользовательские подключения к модулю DSM314.

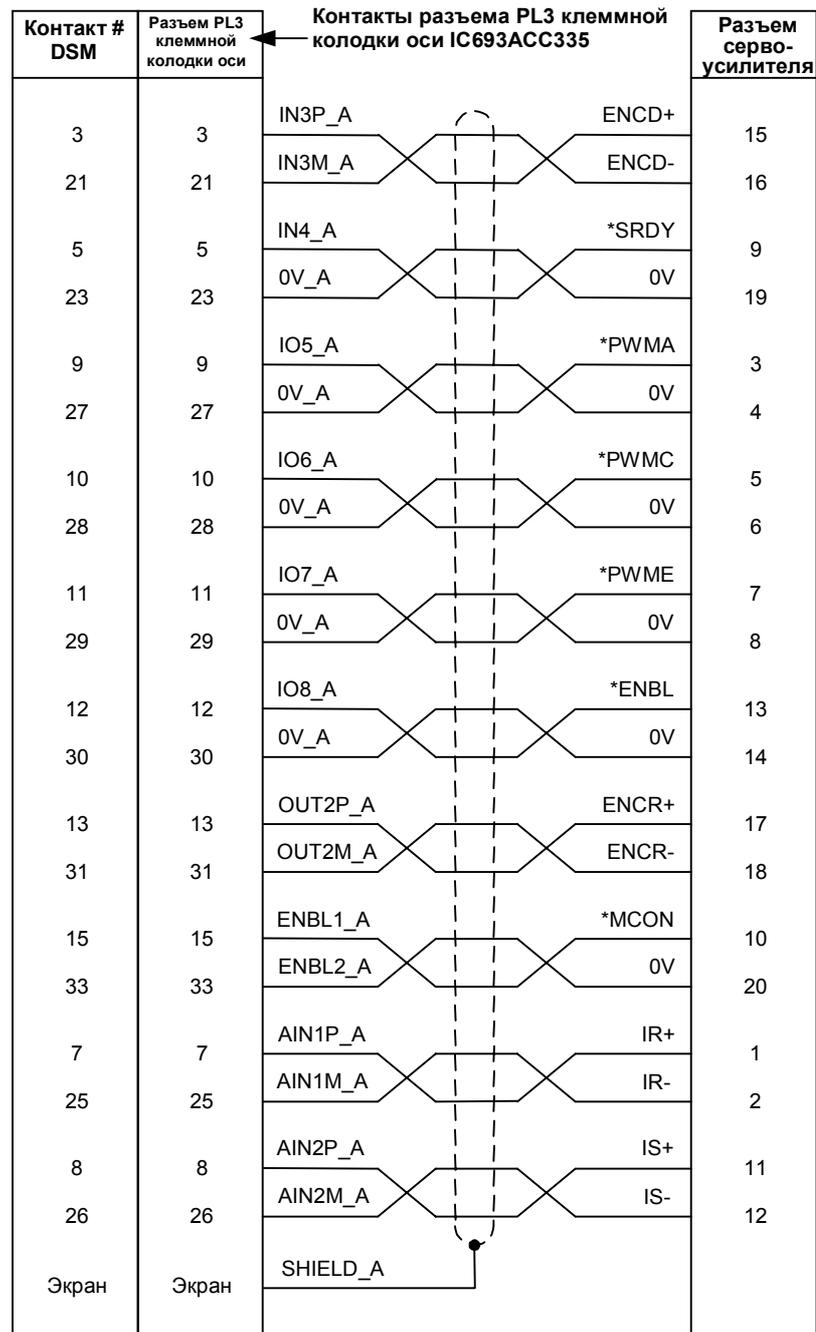


**Рис. 3-16. Подключение цифровой сервооси 1**



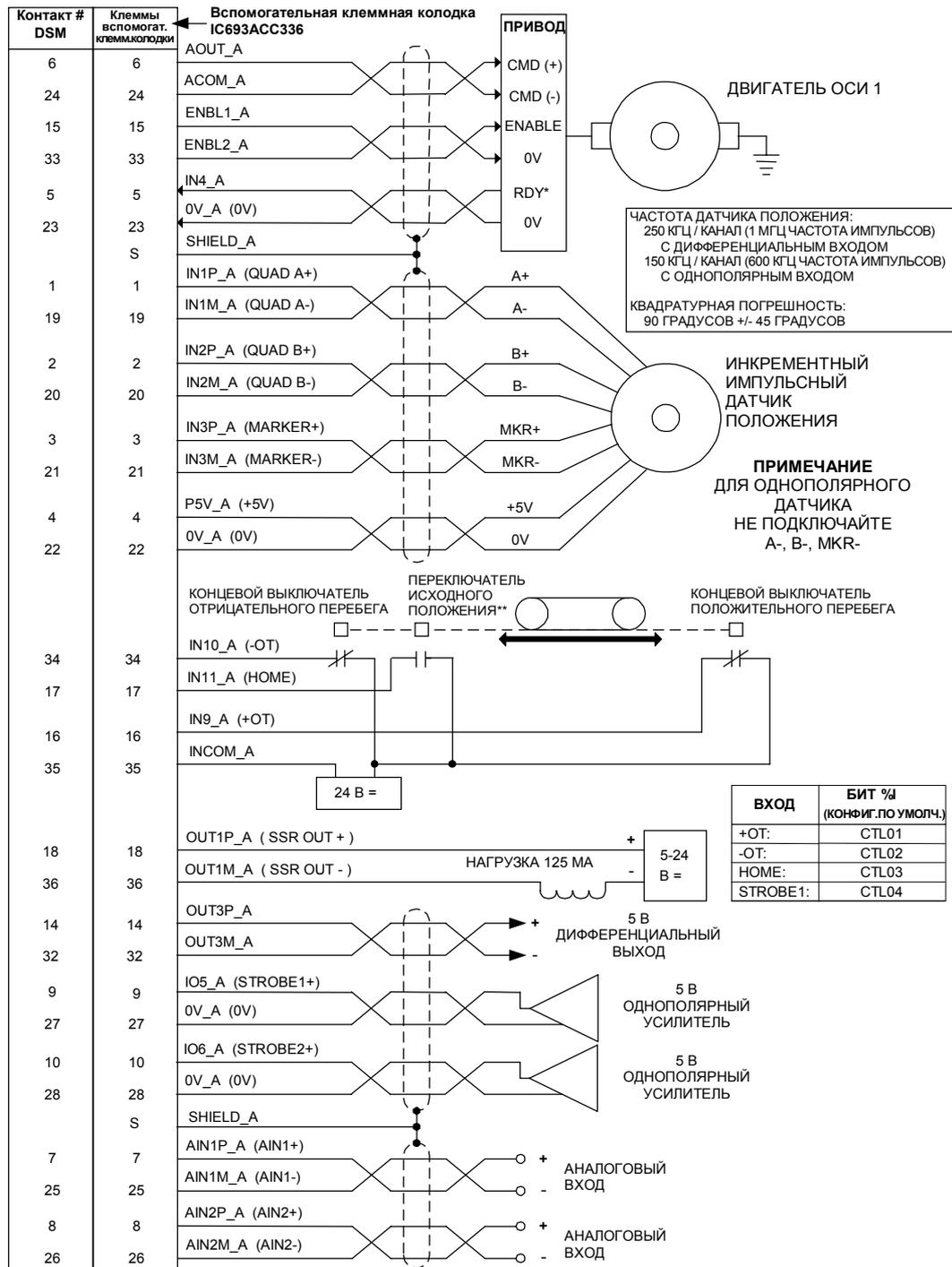
\* Примечание: информацию по переключателю исходного положения см. гл. 6.

**Рис. 3-17. Подключение цифровой сервооси 2**



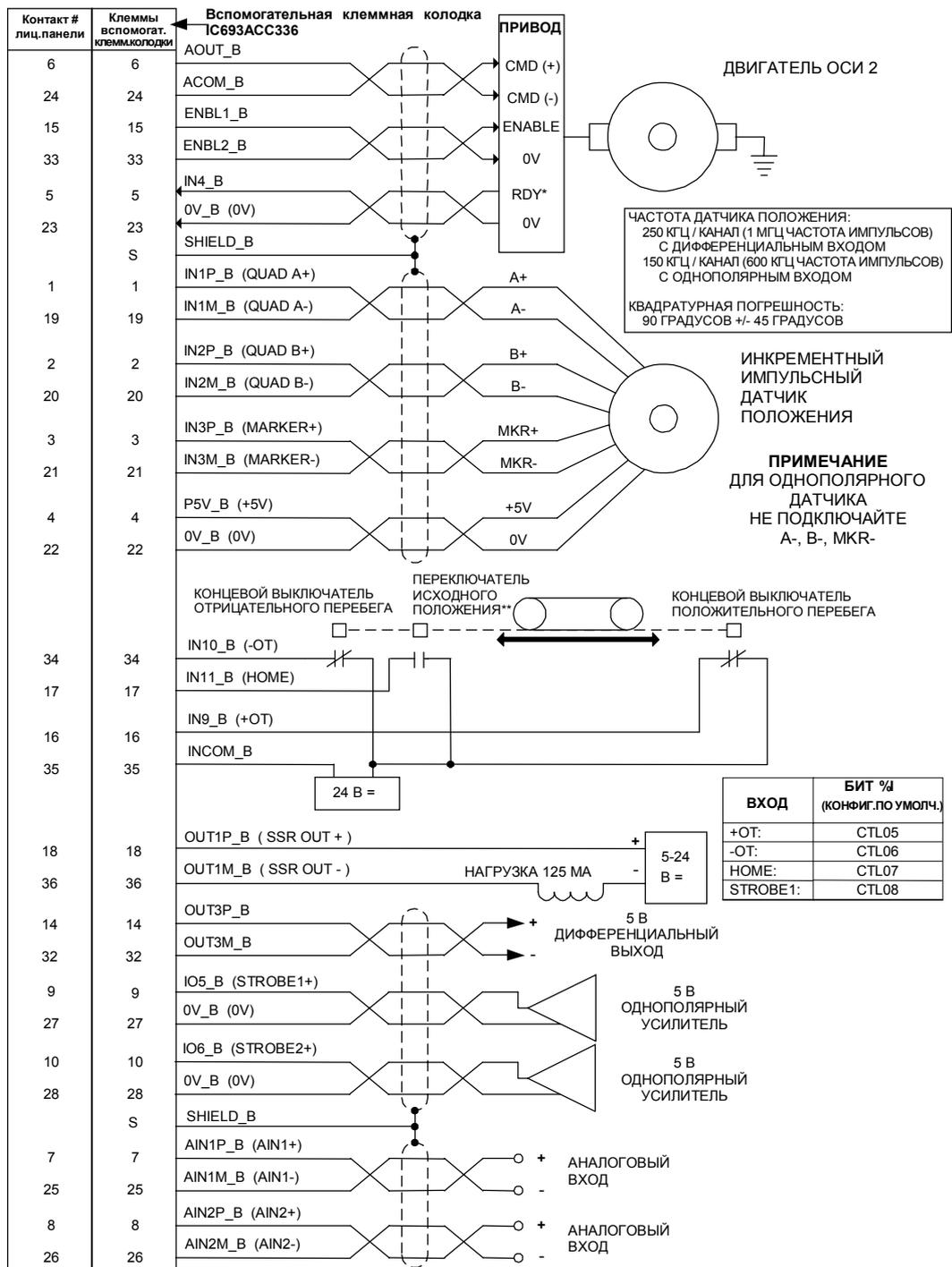
\* Обозначает инвертированный сигнал

Рис. 3-18. Подключение кабеля цифровых сервокоманд  $\alpha$  и  $\beta$ Series (IC800CBL001/002)



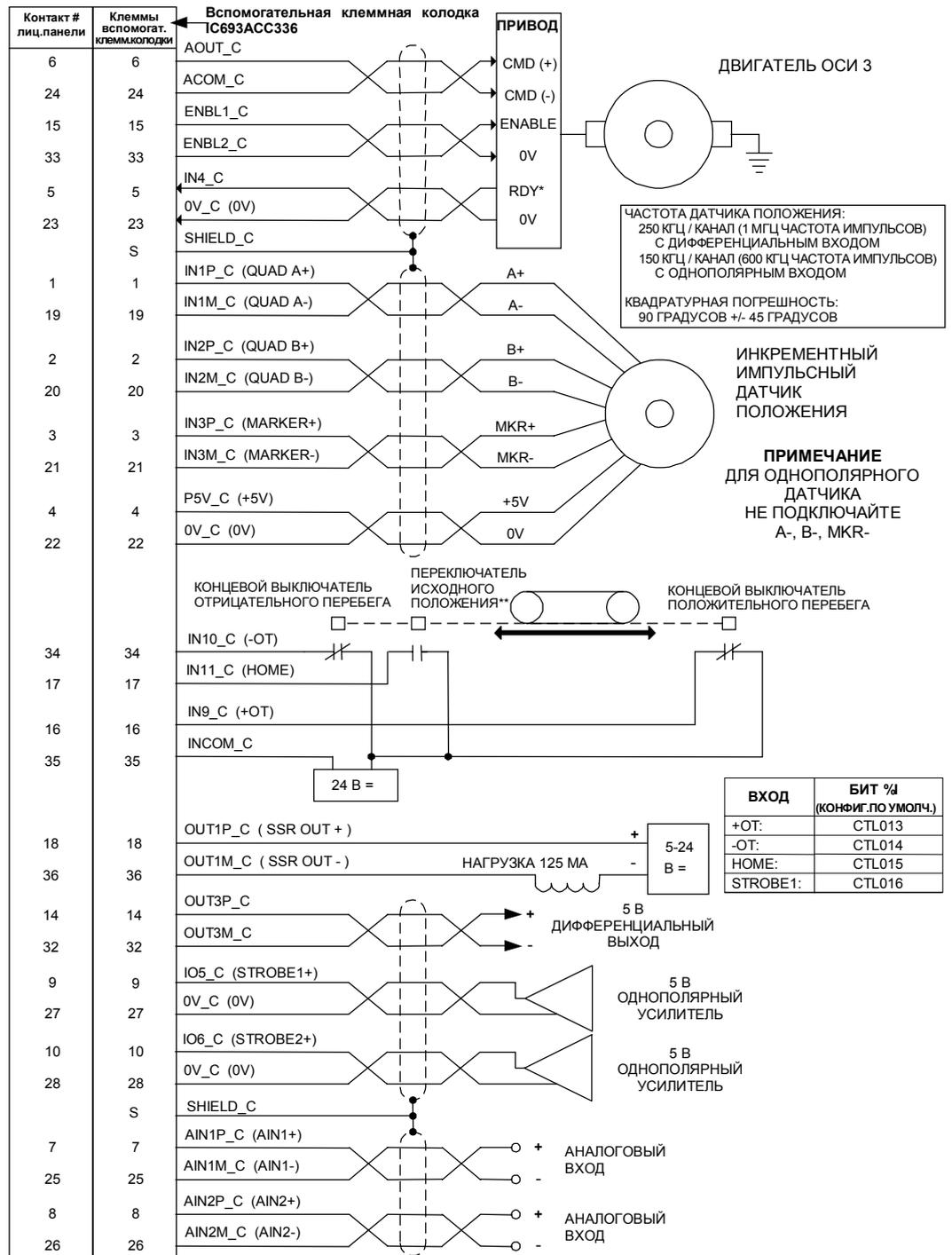
ПРИМЕЧАНИЯ: \* Означает инвертированный сигнал  
 \*\* Информацию по переключателю исходного положения см. в гл. 6.

Рис. 3-19. Подключение аналоговой сервооси 1



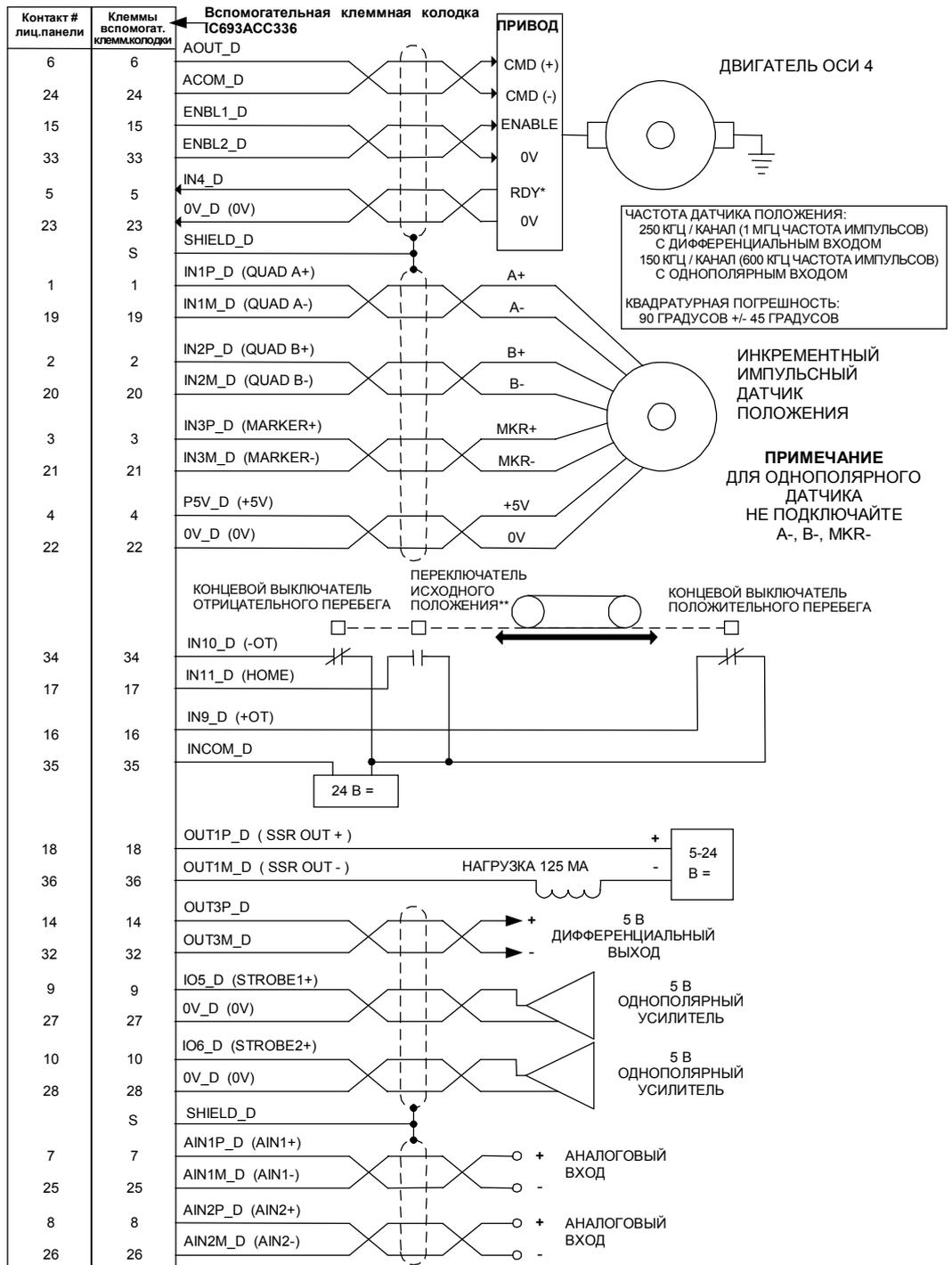
ПРИМЕЧАНИЯ: \* Означает инвертированный сигнал  
 \*\* Информацию по переключателю исходного положения см. в гл. 6.

Рис. 3-20. Подключение аналоговой сервоси 2



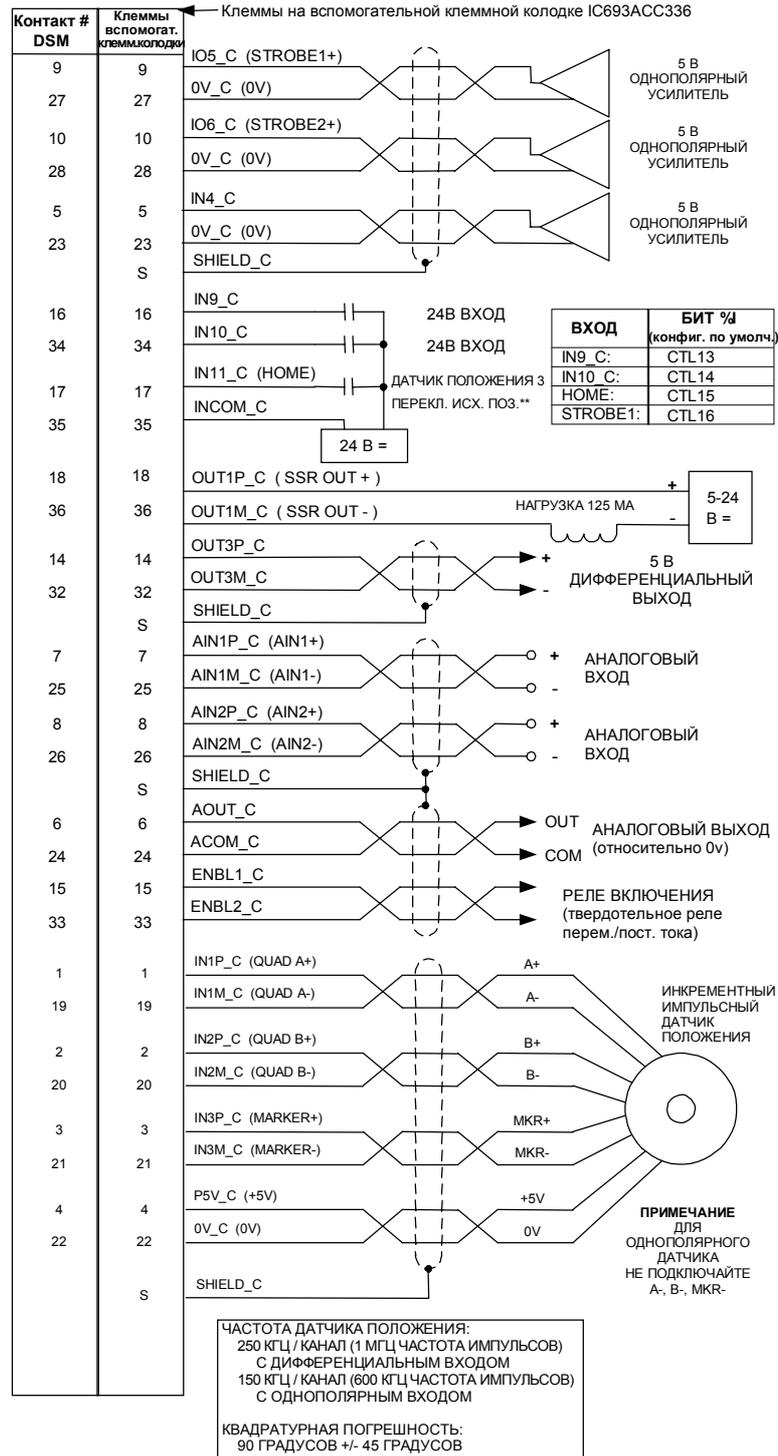
ПРИМЕЧАНИЯ: \* Означает инвертированный сигнал  
 \*\* Информацию по переключателю исходного положения см. в гл. 6.

Рис. 3-21. Подключение аналоговой сервооси 3



ПРИМЕЧАНИЯ: \* Означает инвертированный сигнал  
\*\* Информацию по переключателю исходного положения см. в гл. 6.

Рис. 3-22. Подключение аналоговой сервооси 4



Примечание: \*\* Информацию по переключателю исходного положения см. в гл. 6.

Рис. 3-23. Подключение вспомогательной оси (показана ось 3)

## Технические характеристики В/В

На следующих страницах даны технические характеристики и упрощенные схемы цепей В/В модуля. Описанные цепи В/В представляют собой следующее:

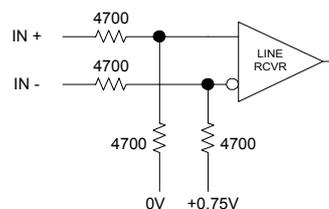
- Дифференциальные/однополярные 5В-входы (IN1, IN2, IN3)
- Однополярный 5В вход-приемник (IN4)
- Оптически изолированные входы-источники/ входы-приемники 24 В (IN9, IN10, IN11, INCOM)
- Однополярные 5В входы/выходы (IO5, IO6, IO7, IO8)
- Дифференциальные выходы 5В (OUT2, OUT3)
- Оптически изолированный выход 24 В пост. тока (OUT1)
- Оптически изолированный Разрешающий релейный выход (OUT4)
- Дифференциальные аналоговые входы +/- 10 В (AIN1, AIN2)
- Однополярные аналоговые выходы +/- 10 В (AOOUT1, ACOM)
- Питание +5 В (P5V, 0V)

## Дифференциальные / однополярные 5В-входы

Обозначение цепи	Функция цепей сервоосей 1, 2	Функция цепей аналоговых сервоосей 1-4 и вспомогат. осей 2-4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
IN1	Вход строба 1 (+)	Канал датчика положения A (+)	IN1P_X	1	1	1
	Вход строба 1 (-)	Канал датчика положения A (-)	IN1M_X	19	19	9
IN2	Вход строба 2 (+)	Канал датчика положения B (+)	IN2P_X	2	2	2
	Вход строба 2 (-)	Канал датчика положения B (-)	IN2M_X	20	20	10
IN3	Данные последовательного датчика положения (+)	Маркер датчика положения (+)	IN3P_X	3	3	N/C
	Данные последовательного датчика положения (-)	Маркер датчика положения (-)	IN3M_X	21	21	N/C

Тип В/В:	Дифференциальные / однополярные 5В-входы
Тип цепи:	Вход источника (согласующий резистор 9,4 кОм на 0 В)
Полное входное сопротивление: вход (+) или (-)	9,5 кОм в обычном режиме на 0 В 18,8 кОм дифференциальное
Максимальное входное напряжение:	+/- 15 В обычный режим +/- 20 В дифференциальный
Порог логического 0:	+ 0,8 В макс. однополярный + 0,4 В макс. дифференциальный
Порог логической 1:	+2,0 В мин. однополярный + 1,5 В мин. дифференциальный
Входная фильтрация:	типично 0,5 микросекунды
Частота импульсного датчика положения:	250 кГц/канал (1МГц частота импульсов) макс. с дифференциальными входами 150 кГц/канал (600 кГц частота импульсов) макс. с однополярными входами Допуск сдвига по фазе: 90 градусов +/- 45 градусов
Характеристика строба:	Минимальная ширина импульса: 3 микросекунды Точность позиции захвата: +/- 2 отсчета с дополнительными 10 микросекундами отклонения

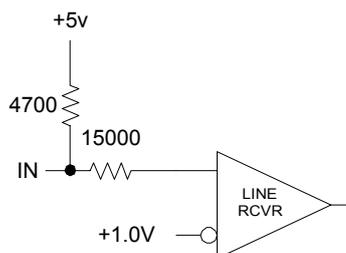
**Примечания:** Используйте (+) вход для однополярного режима и оставьте (-) вход плавающим. Используйте контакт 0В лицевой панели для опорного уровня обычного режима или для возврата однополярного сигнала. Управление входами может осуществляться при помощи 5В TTL или CMOS логики.



## Однополярный 5В вход-приемник

Обозначение цепи	Функция цепи сервооси 1 - 4	Функция цепи вспомогательной оси 2 - 4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
IN4	Вход готовности устройства	Вход 5В лицевой панели	IN4_X	5	5	N/C

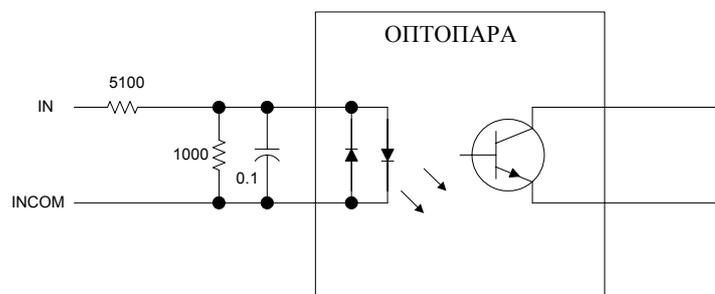
Тип В/В:	Однополярный 5В вход-приемник
Тип цепи:	Вход-приемник (согласующий резистор 4,5 кОм на внутренние +5 В)
Полное входное сопротивление:	4,7 кОм на +5 В
Максимальное входное напряжение:	+/- 10.0 В
Порог логического 0:	+0.8 В макс.
Порог логической 1:	+2.0 В мин
Входная фильтрация:	1,0 микросекунда (типично) аппаратный фильтр + частота опроса контура позиции (0,5, 1,0 или 2,0 миллисекунды).
<b>Примечания:</b>	Чтобы включить, этот вход должен быть установлен на 0 В.



## Оптически изолированные входы-источники/ входы-приемники 24 В

Обозначение цепи	Функция цепи сервооси 1 - 4	Функция цепи вспомогательной оси 2 - 4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
IN9	Перебег (+)	Вход 24 В лицевой панели	IN9_X	16	16	6
IN10	Перебег (-)	Вход 24 В лицевой панели	IN10_X	34	34	14
IN11	Переключатель Исходной позиции	Переключатель Исходной позиции	IN11_X	17	17	7
INCOM	24 В вход общий	24 В вход общий	INCOM_X	35	35	15

- Тип В/В: Оптически изолированные входы-источники/ входы-приемники 24 В
- Тип цепи: Источник / приемник (резистор 5К к INCOM)
- Полное входное сопротивление: 5.4 кОм к INCOM (@ 24 В пост. тока)
- Максимальное входное напряжение: +/- 30.0 В (по отношению к INCOM)
- Порог логического 0: +/-6.0 В макс. (по отношению к INCOM)
- Порог логической 1: +/-18.0 В мин. (по отношению к INCOM)
- Входная фильтрация: 5 миллисекунд типично
- Примечания:** Эти входы используют двунаправленные оптопары и могут быть включены положительным или отрицательным входом по отношению к INCOM.

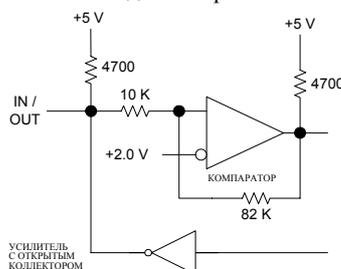


## Однополярные 5В-вводы / выводы

Обозначение цепи	Функция цепей сервоосей 1, 2	Функция цепей аналоговых сервоосей 1-4 и вспомогательных осей 2-4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
IO5 0V	Серво ШИМ / тревога 0 В	Вход строба 1 0 В	IO5_X / IN5_X 0V_X	9 27	9 27	N/C N/C
IO6 0V	Серво ШИМ / тревога 0 В	Вход строба 2 0 В	IO6_X / IN6_X 0V_X	10 28	10 28	N/C N/C
IO7 0V	Серво ШИМ / тревога 0v	Не используется 0v	IO7_X / IN7_X 0V_X	11 29	11 29	N/C N/C
IO8 0V	Серво ВКЛ. / тревога 0 В	Не используется 0 В	IO8_X / IN8_X 0V_X	12 30	12 30	N/C N/C

Тип В/В:	Однополярные 5В-вводы / выводы
Тип цепи:	Приемник (согласующий резистор 4,7 кОм на внутренние +5 В)
Полное входное сопротивление:	4,7 кОм на внутренние +5 В
Максимальное входное напряжение:	-1.0 В , +7.0 В
Порог логического 0:	+0.8 В макс.
Порог логической 1:	+2.4 В мин
Входная фильтрация:	типично 10 микросекунд
Ток выхода приемника	10 мА макс.
Выходное напряжение в состоянии включения	+0.5 В при 10 мА
Характеристика строба:	Минимальная ширина импульса: 10 микросекунд. Точность позиции захвата: +/- 2 отсчета с дополнительными 10 микросекундами отклонения

**Примечания:** Для цифровых сервосистем эти точки действуют как выходы ШИМ / ВКЛ. и аварийные входы. Для аналоговых сервосистем и для вспомогательных осей эти точки являются только входами. Указанные контакты 0v должны, в общем случае, использоваться для возврата сигнала.



## Дифференциальные 5В-выходы

Обозначение цепи	Функция цепей сервоосей 1, 2	Функция цепей аналоговых сервоосей 1-4 и вспомогат. осей 2-4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
OUT2	Запрос последовательного датчика положения (+)	Не используется	OUT2P_X	13	13	N/C
	Запрос последовательного датчика положения (-)	Не используется	OUT2M_X	31	31	N/C
OUT3	Лицевая панель, 5 В, выход (+)	Лицевая панель, 5 В, выход (+)	OUT3P_X	14	14	5
	Лицевая панель, 5 В, выход (-)	Лицевая панель, 5 В, выход (-)	OUT3M_X	32	32	13

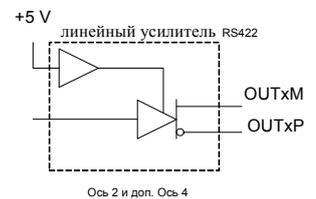
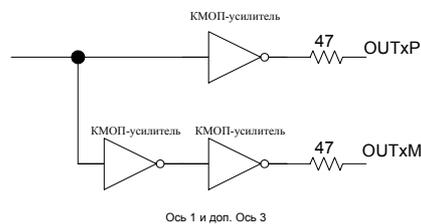
Тип В/В: Дифференциальные 5В-выходы

Тип цепи: Дифференциальный каскадный выход (источник / приемник)

Выходной ток источника / приемника: 20 мА макс.

Выходное напряжение: +/- 1.5 В на 120 Ом дифференциальной нагрузки

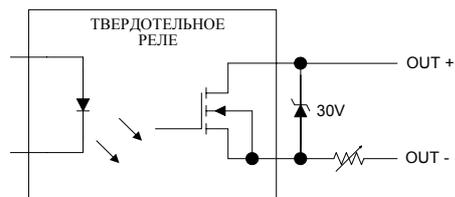
**Примечания:** Ось 1 и ось 3 используют КМОП-усилители с последовательными резисторами 47 ом, ось 2 и ось 4 используют линейные усилители RS-422



## Оптически изолированный выход, 24 В пост. тока

Обозначение цепи	Функция цепи сервооси 1 - 4	Функция цепи вспомогат. оси 2 - 4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
OUT1	Лицевая панель 24 В выход (+)	Лицевая панель 24 В выход (+)	OUT1P_X	18	18	8
	Лицевая панель 24 В выход (-)	Лицевая панель 24 В выход (-)	OUT1M_X	36	36	16

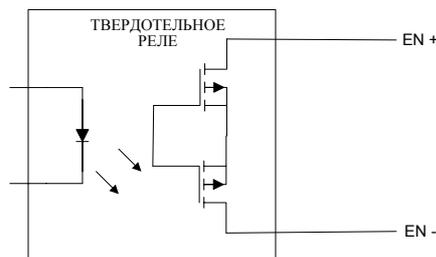
Тип В/В:	Оптически изолированный выход, 24 В пост. тока
Тип цепи:	Изолированное твердотельное реле (SSR)
Выходной ток:	125 мА непрерывно, 500 мА в течение 10 мс (резистивный или индуктивный)
Падение напряжения на выводе:	1.0 В макс. при 0.125 А
<b>Примечания:</b>	Выход защищен диодным ограничителем на 30 В и самовосстанавливающимся полимерным предохранителем на 0.2 А Polyswitch. Если происходит короткое замыкание, то выход автоматически переключается на высокое полное сопротивление до тех пор, пока не будет снята нагрузка. Не допускается новая подача нагрузки в течение 60 секунд. Этот выход является выходом постоянного тока и постоянно находится в состоянии ВКЛ., если полярность подключения к нему противоположна.



### Оптически изолированный включающий релейный выход

Обозначение цепи	Функция цепей сервоосей 1, 2	Функция цепей аналоговых сервоосей 1 - 4	Функция цепи вспомогательной оси 2 - 4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
ENBL	Серво MCON (+)	Привод вкл. (+)	Привод вкл. (+)	ENBL1_X	15	15	N/C
	Серво MCON 0 В	Привод вкл. (-)	Привод вкл. (-)	ENBL2_X	33	33	N/C

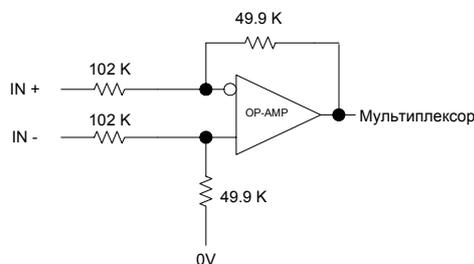
- Тип В/В: Оптически изолированный включающий релейный выход
- Тип цепи: Изолированное твердотельное реле (SSR) переменного тока
- Выходной ток: 30 мА постоянно, 50 мА в течение 10 мс
- Падение напряжения на выводе: 1.0 В макс. при 10 мА
- Примечания:** Этот выход является слаботочным выходом SSR. Выход находится в состоянии ВКЛ., если горит соответствующий светодиод на лицевой панели "Ось Включена". Это имеет место в следующих ситуациях:
- Разрешена работа сервосистемы
  - Используется команда *Force Digital Servo Velocity (Включить цифровое сервоуправление скоростью) %AQ* (ось 1, 2)
  - Используется команда *Force Analog Output (Включить аналоговый выход) %AQ*



## Дифференциальные +/- 10 В аналоговые входы

Обозначение цепи	Функция цепей сервоосей 1, 2	Функция цепей аналоговых сервоосей 1-4 и вспомогательных осей 2-4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
AIN1	IR ток фазы (+)	Аналоговый вход лицевой панели (+)	AIN1P_X	7	7	N/C
	IR ток фазы (-)	Аналоговый вход лицевой панели (-)	AIN1M_X	25	25	N/C
AIN2	IS ток фазы (+)	Аналоговый вход лицевой панели (+)	AIN2P_X	8	8	N/C
	IS ток фазы (-)	Аналоговый вход лицевой панели (-)	AIN2M_X	26	26	N/C

Тип В/В:	Дифференциальные +/- 10 В аналоговые входы
Тип цепи:	Дифференциальный вход
Полное входное сопротивление:	Обычный режим 102 кОм по отношению к разъему 0v лицевой панели 204 кОм дифференциальное
Максимальное входное напряжение:	Обычный режим +/- 15 В по отношению к разъему 0v лицевой панели +/- 20 В дифференциальный
Разрешение:	15 бит
Линейность	13 бит
Входное смещение:	+/- 1.0 милливольт
Коэффициент преобразования:	+/- 10.0 В = +/- 32000 импульсов
Точность преобразования:	+/- 0.5 %
Период обновления:	2 миллисекунды + время цикла ПЛК, если данные передаются в %AI таблицу ПЛК.
<b>Примечания:</b>	Используйте контакты 0v лицевой панели для опорного уровня обычного режима



## Однополярный аналоговый выход +/- 10 В

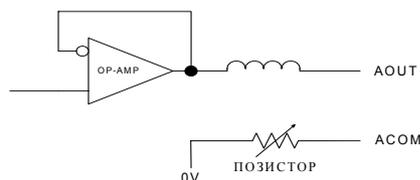
Обозначение цепи	Функция цепей аналоговых сервоосей 1 - 4	Функция цепей аналоговых сервоосей 1, 2 и вспомогательных осей 2-4	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
AOUT1	Команда Аналоговой Скорости Сервопривода или Команда Аналогового Момент вращающия Сервопривода	Аналоговый выход лицевой панели	AOUT_X	6	6	4
АСОМ	Аналоговый выход общий	Аналоговый выход общий	АСОМ_X	24	24	12

Тип В/В:	Однополярный аналоговый выход
Тип цепи:	Выход следящего устройства Op Amp Voltage
Полное сопротивление нагрузки:	Минимум 2 кОм
Выходной ток:	5 мА макс.
Разрешение:	13 бит
Линейность	13 бит
Напряжение выходного смещения:	+/- 500 микровольт макс.
Включить Коэффициент передачи цифро-аналогового преобразования:	+/- 10.0 В = +/-32000 импульсов
Точность преобразования:	+/- 1.0%
Включить период обновления аналогового выхода:	1. Цикл ПЛК, если используется команда <i>Force Analog Output %AQ</i> 2. 250 микросекунд, если используется выход настройки цифрового сервоустройства.

### Примечания:

Поскольку это – однополярный выход, то он должен, как правило, работать вместе с пользовательским устройством, имеющим дифференциальный вход, чтобы предупредить проблемы с шумами, возникающими в обычном (синфазном) режиме. Положительный дифференциальный вход должен быть подключен к AOUT, а отрицательный – к АСОМ.

Для выбора источника аналогового выхода может быть использована команда *Select Analog Output Mode (Режим выбора аналогового выхода) %AQ* Более детальную информацию см. в главе 5.



## Питание +5 В

Обозначение цепи	Функция цепи сервооси	Функция цепи вспомогательной оси	Обозначение сигнала (X = разъем A, B, C или D)	Контакт лицевой панели	Вспомогательная клеммная колодка	Клеммная колодка сервоуправления
P5V	Питание 5 В	Питание 5 В	P5V_X	4	4	3
0V	0 В	0 В	0V_X	22	22	11

Тип В/В:	Питание +5 В датчика положения
Тип цепи:	Питание +5 В с электронной защитой от короткого замыкания
Выходное напряжение:	4.70 - 5.20 В при 0.5 А
Выходной ток:	0.5 А макс. (вместе для всех разъемов)
<b>Примечания:</b>	<p>Этот выход предназначен для питания внешних устройств, таких как инкрементные импульсные датчики, для которых в общей сложности требуется менее 0.5 А от всех 4 разъемов осей. Выходной ток обеспечивается источником питания +5 В на системной плате ПЛК; в модуле DSM314 установлена электронная защита от короткого замыкания.</p> <p>Общий ток, потребляемый внешними устройствами от этой цепи +5 В, должен быть прибавлен к значению потребления в окне конфигурации модуля DSM314 программы конфигурации; он должен быть учтен при ручном расчете загрузки источника питания.</p> <p>Указанный контакт 0v должен, в общем случае, использоваться как сигнал обратной связи по питанию.</p>

Данная глава описывает все детали конфигурации, требуемые для установки модуля DSM314 в конкретном приложении. В главе 2 описываются указания поконфигурированию системы, для установки команду *Jog (Толчок)* на модуль DSM с целью проверки работоспособности компонентов системы. Информацию по конфигурации электронной САМ см. в гл. 16.

Модуль DSM314 требует **микропрограмму ЦП версии 10.00 или выше**. См. документацию по ЦП ПЛК, чтобы согласовать аппаратную платформу ЦП с указанной версией микропрограммы. Модуль DSM314 конфигурируется с использованием одного из следующих программных инструментов конфигурации / программирования.

- Пакет SIMPLICITY Machine Edition Logic Developer PLC, версия 2.1 или более поздняя.
- Пакет VersaPro, версия 1.1 или более поздняя.

Конфигурация является двухступенчатым процессом:

1. Конфигурация крейта / слота
2. Конфигурация модуля

## Конфигурация крейта / слота

Конфигурация оборудования определяет тип и расположение всех модулей, размещаемых в крейте ПЛК. Это осуществляется на первом этапе настройки, отображающей модули на базовой плате, с последующим сохранением информации в файле конфигурации, загружаемом затем в ЦП ПЛК.

Детальное описание выбора базовой платы DSM314 и расположения слотов в крейте ПЛК см. в гл. 2 раздел 4 "Конфигурация модуля Motion Mate DSM314 (вводное руководство)". После выполнения конфигурации крейта / слота проводится вторая часть процесса конфигурации – конфигурация модуля, чтобы определить параметры DSM314 под конкретные требования приложения.

## Конфигурация модуля

### Задание параметров конфигурации

Так же как конфигурация крейта В/В конфигурация модуля выполняется в рабочих окнах программы конфигурации оборудования. Данные по конфигурации оборудования присутствуют в табличном формате. Таблицы соответствуют группам, показанным ниже. Название таблицы и / или таблиц, которые соответствуют данной группе, приведено в скобках после названия группы. Обратите внимание, чтоб таблицы появляются и исчезают в зависимости от установок конфигурации, сделанных в таблице Установки. Например, если отключена ось 4, то таблица Axis (Ось) #4 не выводится.

- Данные конфигурации модуля (Установки, Биты CTL, Биты выхода)
- Последовательное соединение (порт SNP)
- Данные конфигурации оси
- Настройка оси
- Расширенный набор установочных параметров
- Потребление электроэнергии

Основное содержание каждой таблицы показано ниже:

Название таблицы	Функция или описание
<b>Settings (Установочные параметры)</b>	Содержит назначения и длины ссылок ПЛК, установочные параметры оси DSM и другие глобальные данные
<b>SNP</b>	Установочные параметры порта SNP на передней панели DSM
<b>CTL Bits (Биты CTL)</b>	Конфигурация 24 контрольных битов DSM
<b>Output Bits (Биты выхода)</b>	Конфигурация 8 цифровых выходов лицевой панели DSM
<b>Axis #1 - Axis #4 (Ось #1 - Ось #4)</b>	Конфигурация параметров оси, таких как Границы позиции, Скорость поиска исходной позиции и Ускорение толчка
<b>Tuning #1 – Tuning #4 (Настройка #1 – Настройка #4)</b>	Конфигурация параметров настройки сервоконтура, таких как Тип двигателя, Постоянная времени контура позиционирования и Упреждение по скорости.
<b>Advanced (Расширенный набор)</b>	Позволяет пользователю ввести пользовательские наборы параметров настройки для любой оси
<b>Power Consumption (Потребление электроэнергии)</b>	Даются значения энергопотребления модулем DSM от источника питания на объединительной плате (4,0 Вт + питание датчика положения)

За дополнительной информацией по работе программы конфигурации обращайтесь в онлайн-службу поддержки руководства вашего ПО или соответствующего пользовательского ПО:

*Руководство пользователя по программному обеспечению VersaPro™, GFK-1670*

Основы использования пакета SIMPLICITY® Machine Edition Logic Developer-PLC, GFK-1918

## Установочные параметры

Таблица Settings (Установочные параметры) содержит информацию о конфигурации, которая позволяет задать основные операции модуля. Эти параметры действуют на все управляемые оси, режимы работы осей и т.п. Выбор, осуществляемый в данных таблицах, может приводить к появлению или исчезновению других таблиц в ходе конфигурации. Например, если отключается ось #4, то таблицы Axis (Ось) и Tuning (Настройка), относящиеся к оси #4, не появляются.

Обратите внимание, что в таблице Settings должны быть заданы имена программы Motion (Движение), Local Logic и Cam-блока. Эти имена определяют, какие программы, сохраняемые в ЦП, будут переданы на соответствующий DSM при включении питания системы. Дополнительную информацию по этому направлению см. в соответствующих главах данного руководства.

На каждом цикле ЦП происходит автоматический обмен данными между DSM314 и ЦП. Таблица Settings содержит ссылки на данные интерфейса ЦП и начальные адреса для автоматической передачи данных. Параметры конфигурации, приведенные в таблице Settings, описаны в табл. 4-1. Все обозначения в колонке "Ссылки" относятся к таблицам данной главы.

**Таблица 4-1. Таблица Settings (Установочные параметры)**

Параметр конфигурации	Описание	Значения	По умолчанию (Стандарт, параметры)	Ед.	Ссылки
Number of Axes	Количество Осей	1 2 3 4	4	-	1.01
%I Reference	Память %I (80 бит)	Зависит от ЦП	%I00001 или следующая более высокая ссылка	-	1.02
%I Length	Длина %I	32 = 1 Ось 48 = 2 Ось 64 = 3 Ось 80 = 4 Ось	- Длина задается автоматически параметром Number of Axes (Количество осей)	-	1.02
%Q Reference	Память %Q (80 бит)	Зависит от ЦП	%Q00001 или следующая более высокая ссылка	-	1.02
%Q Length	Длина %Q	32 = 1 Ось 48 = 2 Ось 64 = 3 Ось 80 = 4 Ось	- Длина задается автоматически параметром Number of Axes (Количество осей)	-	1.02
%AI reference	Память %AI (84 бита)	Зависит от ЦП	%AI00001 или следующая более высокая ссылка	-	1.02
%AI Length	Длина %AI	24 = 1 Ось 44 = 2 Ось 64 = 3 Ось 84 = 4 Ось	- Длина задается автоматически параметром Number of Axes (Количество осей)	-	1.02
%AQ reference	Память %AQ (12 бит)	Зависит от ЦП	%AQ00001 или следующая более высокая ссылка	-	1.02
%AQ Length	Длина %AQ	3 = 1 Ось 6 = 2 Ось 9 = 3 Ось 12 = 4 Ось	- Длина задается автоматически параметром Number of Axes (Количество осей)	-	1.02
Axis 1 Mode	Режим Оси 1	Аналоговый Сервопривод Цифровой Сервопривод	Аналоговый Сервопривод	-	1.03
Axis 2 Mode	Режим Оси 2	Аналоговый Сервопривод Цифровой Сервопривод Вспомогательная Ось	Аналоговый Сервопривод	-	1.03

Axis 3 Mode	Режим Оси 3	Аналоговый Сервопривод Вспомогательная Ось	Вспомогательная Ось	-	1.03
Axis 4 Mode	Режим Оси 4	Выключено Аналоговый Сервопривод Вспомогательная Ось	Выключено	-	1.03
Local Logic Mode	Режим локальной логике	Выключено Включено	Выключено	-	1.04
Total Encoder Power	Общая мощность энкодера	Числовое поле	0	Вт	1.05
Motion Program Block Name	Имя блока программы управления	ASCII-20	<пробел>	-	1.06
Local Logic Block Name	Имя блока локальной логике	ASCII-20	<пробел>	-	1.07
CAM Block Name	Имя блока CAM	ASCII-20	<пробел>	-	1.08

**1.01 Number of Axes (Количество Осей)** Этот параметр позволяет выбрать количество осей, которыми будет управлять модуль DSM314, и размер пакета при автоматическом обмене данными между ПЛК и DSM. (По умолчанию = 4). В приведенных ниже таблицах даны возможные комбинации осей для Аналогового и Цифрового режимов. Оси, названные как **Limited Aux Axis (Ограниченная вспомогательная ось)**, обеспечивают обратную связь по положению, но не позволяют генерировать внутренние команды движения.

Таблица 4-2. Количество Осей

Поз.	# Осей	Ось 1	Ось 2	Ось 3	Ось 4	Local Logic	Частота опроса (мс)
1.	4	Аналоговое сервоуст-во (Аналог.серв.)	Аналоговое сервоустройство (Аналоговое серв.)	Аналоговое сервоуст-во (Аналог.серв.)	Аналоговое сервоустройство (Аналог.серв.)	Выключ. (Disabled)	2.0
2.	4	Аналог.серв.	Аналоговое серв.	Аналог.серв.	Вспомогат. ось	Выключ.	2.0
3.	4	Аналог.серв.	Аналоговое серв.	Аналог.серв.	Выключено	Выключ.	2.0
4.	4	Аналог.серв.	Аналоговое серв.	Аналог.серв.	Выключено	Включ. (Enabled)	2.0
5.	4	Аналог.серв.	Аналоговое серв.	Аналог.серв.	Ограниченная вспомогат. ось	Включ.	2.0
6.	4	Аналог.серв.	Аналоговое серв.	Вспомогат. ось	Аналоговое серв.	Выключ.	2.0
7.	4	Аналог.серв.	Аналоговое серв.	Вспомогат. ось	Вспомогат. ось	Выключ.	2.0
8.	4	Аналог.серв.	Аналоговое серв.	Вспомогат. ось	Выключено	Выключ.	2.0
9.	4	Аналог.серв.	Аналоговое серв.	Вспомогат. ось	Выключено	Включ.	2.0
10.	4	Аналог.серв.	Аналоговое серв.	Вспомогат. ось	Ограниченная вспомогат. ось	Включ.	2.0
11.	4	Аналог.серв.	Вспомогат. ось	Аналог.серв.	Аналоговое серв.	Выключ.	2.0
12.	4	Аналог.серв.	Вспомогат. ось	Аналог.серв.	Вспомогат. ось	Выключ.	2.0
13.	4	Аналог.серв.	Вспомогат. ось	Аналог.серв.	Выключено	Выключ.	2.0
14.	4	Аналог.серв.	Вспомогат. ось	Аналог.серв.	Выключено	Включ.	2.0

Поз.	# Осей	Ось 1	Ось 2	Ось 3	Ось 4	Local Logic	Частота опроса (мс)
15.	4	Аналог.серв.	Вспомогат. ось	Аналог.серв.	Ограниченная вспомогат. ось	Включ.	2.0
16.	4	Аналог.серв.	Вспомогат. ось	Вспомогат. ось	Аналоговое серв.	Выключ.	2.0
17.	4	Аналог.серв.	Вспомогат. ось	Вспомогат. ось	Вспомогат. ось	Выключ.	2.0
18.	4	Аналог.серв.	Вспомогат. ось	Вспомогат. ось	Выключено	Выключ.	2.0
19.	4	Аналог.серв.	Вспомогат. ось	Вспомогат. ось	Выключено	Включ.	2.0
20.	4	Аналог.серв.	Вспомогат. ось	Вспомогат. ось	Ограниченная вспомогат. ось	Включ.	2.0
21.	3	Аналог.серв.	Аналоговое серв.	Аналог.серв.	-	Выключ.	2.0
22.	3	Аналог.серв.	Аналоговое серв.	Аналог.серв.	-	Включ.	2.0
23.	3	Аналог.серв.	Аналоговое серв.	Вспомогат. ось	-	Выключ.	2.0
24.	3	Аналог.серв.	Аналоговое серв.	Вспомогат. ось	-	Включ.	2.0
25.	3	Аналог.серв.	Вспомогат. ось	Аналог.серв.	-	Выключ.	2.0
26.	3	Аналог.серв.	Вспомогат. ось	Аналог.серв.	-	Включ.	2.0
27.	3	Аналог.серв.	Вспомогат. ось	Вспомогат. ось	-	Выключ.	2.0
28.	3	Аналог.серв.	Вспомогат. ось	Вспомогат. ось	-	Включ.	2.0
30.	2	Аналог.серв.	Аналоговое серв.	-	-	Выключ.	1.0
31.	2	Аналог.серв.	Аналоговое серв.	-	-	Включ.	2.0
32.	2	Аналог.серв.	Вспомогат. ось	-	-	Выключ.	1.0
33.	2	Аналог.серв.	Limited Aux Axis (Ограниченная вспомогат. ось)	-	-	Включ.	1.0
34.	1	Аналог.серв.	-	-	-	Выключ.	0.5
35.	1	Аналог.серв.	-	-	-	Включ.	1.0

Таблица 4-3. Конфигурации цифровых осей

Поз.	# Осей	Ось 1	Ось 2	Ось 3	Ось 4	Local Logic	Частота опроса (мс)
1.	4	Цифров.серв.	Цифровое серв.	Аналог.серв.	Выключено	Выключ.	2.0
2.	4	Цифров.серв.	Цифровое серв.	Аналог.серв.	Выключено	Включ.	2.0
3.	4	Цифров.серв.	Цифровое серв.	Вспомогат. ось	Выключено	Выключ.	2.0
4.	4	Цифров.серв.	Цифровое серв.	Вспомогат. ось	Выключено	Включ.	2.0
5.	3	Цифров.серв.	Цифровое серв.	Аналог.серв.	-	Выключ.	2.0
6.	3	Цифров.серв.	Цифровое серв.	Аналог.серв.	-	Включ.	2.0
7.	3	Цифров.серв.	Цифровое серв.	Вспомогат. ось	-	Выключ.	2.0
8.	3	Цифров.серв.	Цифровое серв.	Вспомогат. ось	-	Включ.	2.0
9.	2	Цифров.серв.	Цифровое серв.	-	-	Выключ.	2.0
10.	2	Цифров.серв.	Цифровое серв.	-	-	Включ.	2.0
11.	1	Цифров.серв.	-	-	-	Выключ.	2.0
12.	1	Цифров.серв.	-	-	-	Включ.	2.0

- 1.02 I/Q/AI/AQ Len (Длины I/Q/AI/AQ).** Отображает начальные адреса и количество ссылок %I, %Q, %AI и %AQ, назначенных модулю DSM314. Размеры ссылок задаются, когда пользователь конфигурирует количество осей.
- 1.03 Axis n Mode (Режим Оси n).** Эти параметры определяют типы выходных команд, обеспечиваемых подсистемами сервоуправления. *Digital Servo (Цифровое сервоустройство)* позволяет выбрать специальный цифровой выход для Цифровых сервоприводов GE Fanuc. Если выбрано **Digital Servo**, то Оси 1 и 2 должны быть цифровыми. Выбор **Analog Servo (Аналоговое сервоустройство)** определяет команду скорости +/-10 В или команду момента вращения +/-10 В для стандартного аналогового сервопривода. Интерфейс момента вращения или скорости конфигурируется при задании Analog Servo Command (Команда аналогового сервоустройства) в конфигурации модуля. **Auxiliary Axis (Вспомогательная ось)** отключает контур позиционирования; в результате внутренний генератор команд и вход позиции датчика положения могут быть использованы для функций следящего устройства или САМ. Если какой-либо разъем оси используется как вход ведущего источника для режима слежения, то он должен быть сконфигурирован как **Вспомогательная ось**. **Вспомогательная ось** будет выдавать аналоговое напряжение, пропорциональное *Commanded Velocity (Заданной скорости)*, если значение *Коэффициента упреждения по скорости* отлично от нуля. Если ось сконфигурирована как **Вспомогательная ось** и обозначена как **Limited Aux Axis (Ограниченная вспомогательная ось)** в таблице 4-2, то обратная связь по положению имеется, но внутренняя генерация команд движения отсутствует. Ось, сконфигурированная как **Disabled (Выключено)** (применяется только к оси 4) обеспечивает аналоговый и цифровой В/В, но обратная связь по положению или внутренняя генерация команд движения отсутствуют. (По умолчанию = *Аналоговое сервоустройство* (Оси 1-2), *Вспомогательная ось* (Ось 3), *Выключено* (Ось 4)).
- 1.04 Local Logic Mode (Режим Local Logic).** Этот параметр определяет статус процессора Local Logic. Чтобы включить Local Logic, этот параметр должен быть установлен на Включено. Если Local Logic включено, то максимальное число имеющихся в распоряжении сервоосей составляет 3. Если *Local Logic Mode* = Enabled (Включено), то тогда должно быть также введено *Local Logic Block Name (Имя блока Local Logic)*. (По умолчанию = *Выключено*)
- 1.05 Total Encoder Power (Полная мощность датчиков положения).** Этот параметр определяет полное энергопотребление всех датчиков положения, закрепленных за модулем DSM. (По умолчанию = 0). Этот параметр должен быть задан для всех ведущих датчиков положения и датчиков положения аналоговых осей; он используется для обновления окна Потребления мощности в ПО конфигурации.
- 1.06 Motion Program Block Name (Имя блока Программы движения).** Этот параметр задает дополнительное имя блока Программы движения для выполнения на модуле DSM. Если имя не введено, то DSM считает, что блоки Программы движения не используются. Если имя введено, то блок Программы движения с таким именем должен существовать в активном каталоге. Ввод неправильного имени вызывает появление ошибки при сохранении конфигурации аппаратного обеспечения в ПЛК. Имя может содержать до 31 символа, однако в нем не должно быть пробелов, хотя разрешается использовать подчеркнутые символы. Допускаются символы как верхнего, так и нижнего регистров. (По умолчанию = *<пробел>*).
- 1.07 Local Logic Block Name (Имя блока Local Logic).** Этот параметр задает дополнительное имя блока Local Logic для выполнения на модуле DSM. Если имя не введено, то DSM считает, что блоки Local Logic не используются. Если имя введено, то блок Local Logic с таким именем должен существовать в активном каталоге. Ввод неправильного имени вызывает появление ошибки при сохранении конфигурации аппаратного обеспечения в ПЛК. Имя может содержать до 31 символа, однако в нем не

должно быть пробелов, хотя разрешается использовать подчеркнутые символы. Допускаются символы как верхнего, так и нижнего регистров. Чтобы работал Local Logic, требуется, чтобы **Local Logic Mode** был также установлен на Включено. (По умолчанию = <пробел>).

**1.08 SAM Block Name (Имя САМ-блока).** Определяет дополнительное имя САМ-блока, выполняемого на модуле. Если имя не введено, то DSM считает, что САМ-блоки не используются. Если имя введено, то САМ-блок с таким именем должен существовать в активном каталоге. Ввод неправильного имени вызывает появление ошибки при сохранении конфигурации аппаратного обеспечения в ПЛК. Правилами для имен САМ-блоков являются следующие:

- Разрешается использование только следующих символов: A-Z, a-z, 0-9 и \_ (символ подчеркивания). Непрерывно следующие друг за другом символы подчеркивания и пробела не допускаются.
- Имя САМ-блока должно начинаться с буквы или символа подчеркивания.
- Два блока в одном открытом каталоге не могут иметь одно и то же имя.
- Имя САМ-блока может содержать не более семи символов.

Это свойство впервые было обеспечено в ПО DSM314 версии 2.00; оно требует использования одного из следующих программных инструментов:

- Пакет SIMPLICITY Machine Edition Logic Developer PLC, версия 2.1 или более поздняя.
- Пакет VersaPro 1.5 или более поздний и САМ Editor 1.0 или более поздний.

Более детальную информацию об особенностях САМ см. в главе 16. (По умолчанию = <пробел>).

## Данные конфигурации порта последовательной связи

Порт последовательной связи модуля DSM314 использует разъем RJ-11, маркированный как СОММ, на лицевой панели модуля и поддерживает протокол RS-232. Он используется для обновления микропрограммы на флэш-памяти и должен быть правильно сконфигурирован, чтобы взаимодействовать с ПО обновления, работающим на программаторе. Следует убедиться, что параметры конфигурации программатора и параметры конфигурации Порты последовательной связи DSM314 соответствуют друг другу. Эти параметры конфигурации описаны в таблице 4-4.

Таблица 4-4. Таблица порта SNP (последовательной связи)

Параметр конфигурации	Описание	Значения	По умолч.	Ед.	Ссылка
Baud Rate (Скорость в бодах)	Скорость обмена x порта SNP (бит/сек)	300, 600, 1200, 2400, 4800, 9600, 19200	19200	-	2.01
Stop Bits (Стоповые биты)	Число стоповых битов	1 или 2	1	-	2.02
Parity (Четность)	Паритет	ODD, EVEN, NONE	ODD	-	2.03
Idle Time (Время ожидания)	Время ожидания	1...255	10	секунды	2.04
Modem Turnaround Time (Длительность цикла модема)	Время переключения модема	0...255	0	.01 секунды / отсчет	2.05
SNP ID	SNP ID	Семь символов из A-F и 0-9. Первым символом должно быть A-F.	A000001	-	2.06

- 2.01 Baud Rate (Скорость в бодах).** Параметр определяет скорость передачи данных через последовательный порт в битах в секунду.
- 2.02 Stop Bits (Стоповые биты).** Все устройства последовательной связи используют, по крайней мере, один (1) стоповый бит. Для устройств с меньшей скоростью следует установить два (2) стоповых бита.
- 2.03 Parity (Четность).** Задается, должен ли использоваться бит четности (**NONE**, если не надо) и, если используется, то должен он быть **ODD (НЕЧЕТНЫЙ)** или **EVEN (ЧЕТНЫЙ)**.
- 2.04 Idle Time (Время ожидания).** Задаёт время в секундах, которое DSM314 будет ждать поступления нового сообщения от ведущего устройства, прежде чем принять решение, что связь потеряна или закончена. В этом случае DSM314 заново инициализируется, чтобы ожидать запуска нового сеанса последовательной связи.
- 2.05 Modem Turnaround Time (Длительность цикла модема).** Для использования модема требуется задать Длительность цикла модема. Оно представляет собой тот промежуток времени, который требуется модему для начала передачи данных после получения запроса на передачу. Если модем не используется, то здесь должен быть задан 0. Если модем используется, то здесь должно быть задано значение, превышающее 0.
- 2.06 SNP ID.** Идентификатор, содержащий от 0 до 7 символов из A-F и 0-9. Первый задаваемый символ должен быть A-F. Этот идентификатор должен использоваться для много абонентских сетей. DSM314 поддерживает много абонентские соединения только в том случае, когда соединение RS232 преобразуется в RS422/485.

**Примечание:** Поскольку этот Порт последовательной связи используется только для обновления микропрограммы DSM314, рекомендуется оставить его стандартные установочные параметры. Для соединения этого порта с последовательным портом ПК, на котором работает ПО обновления микропрограммы, используйте кабель IC693CBL316.

## Управляющие биты (CTL)

Конфигурация битов CTL позволяет пользователю сконфигурировать входные источники для Управляющих битов (CTL01-CTL24). Окно конфигурации позволяет пользователю выбрать такую конфигурацию CTL-битов, которая соответствует требованиям программы Motion (Движения) и программы Local Logic. Параметры конфигурации CTL-битов описаны в таблице 4-5. Дополнительную информацию по конфигурации CTL-битов см. в главе 14.

**Таблица 4-5. Таблица CTL-битов**

Параметр конфигурации	Описание	По умолчанию	Ссылка
CTL01 Config	Конфигурация бита CTL01	IN9_A (Ось 1 +OT)	Гл. 14
CTL02 Config	Конфигурация бита CTL02	IN10_A (Ось 1 -OT)	Гл. 14
CTL03 Config	Конфигурация бита CTL03	IN11_A (Переключатель исходного положения Оси 1)	Гл. 14
CTL04 Config	Конфигурация бита CTL04	Strobe1 Level Ось 1)	Гл. 14
CTL05 Config	Конфигурация бита CTL05	IN9_B (Ось 2 +OT)	Гл. 14
CTL06 Config	Конфигурация бита CTL06	IN10_B (Ось 2 -OT)	Гл. 14
CTL07 Config	Конфигурация бита CTL07	IN11_B (Переключатель исходного положения Оси 2)	Гл. 14
CTL08 Config	Конфигурация бита CTL08	Strobe1 Level (Ось 2)	Гл. 14
CTL09 Config	Конфигурация бита CTL09	%Q бит Смещение 12	Гл. 14
CTL10 Config	Конфигурация бита CTL10	%Q бит Смещение 13	Гл. 14
CTL11 Config	Конфигурация бита CTL11	%Q бит Смещение 14	Гл. 14
CTL12 Config	Конфигурация бита CTL12	%Q бит Смещение 15	Гл. 14
CTL13 Config	Конфигурация бита CTL13	IN9_C (Ось 3 +OT)	Гл. 14
CTL14 Config	Конфигурация бита CTL14	IN10_C (Ось 3 -OT)	Гл. 14
CTL15 Config	Конфигурация бита CTL15	IN11_C (Переключатель исходного положения Оси 3)	Гл. 14
CTL16 Config	Конфигурация бита CTL16	Strobe1 Level (Ось 3)	Гл. 14
CTL17 Config	Конфигурация бита CTL17	%Q бит Смещение 24	Гл. 14
CTL18 Config	Конфигурация бита CTL18	%Q бит Смещение 25	Гл. 14
CTL19 Config	Конфигурация бита CTL19	%Q бит Смещение 40	Гл. 14
CTL20 Config	Конфигурация бита CTL20	%Q бит Смещение 41	Гл. 14
CTL21 Config	Конфигурация бита CTL21	%Q бит Смещение 56	Гл. 14
CTL22 Config	Конфигурация бита CTL22	%Q бит Смещение 57	Гл. 14
CTL23 Config	Конфигурация бита CTL23	%Q бит Смещение 72	Гл. 14
CTL24 Config	Конфигурация бита CTL24	%Q бит Смещение 73	Гл. 14

Каждый CTL-бит, показанный в предыдущей таблице, может быть сконфигурирован на одно из значений, приведенных в следующей далее таблице.

Таблица 4-6. Допускаемые значения для табличных STL-битов

Управляемый Local Logic	IN9_D (Ось 4 +ОТ)	Strobe2 Level (Ось 4)	%Q бит Смещение 57
IN9_A (Ось 1 +ОТ)	IN10_D (Ось 4 -ОТ)	%Q бит Смещение 12	%Q бит Смещение 72
IN10_A (Ось 1 -ОТ)	IN11_D (Переключатель исходного положения Оси 4)	%Q бит Смещение 13	%Q бит Смещение 73
IN11_A (Переключатель исходного положения Оси 1)	Strobe1 Level (Ось 1)	%Q бит Смещение 14	FBSA* Бит записи 1
IN9_B (Ось 2 +ОТ)	Strobe2 Level (Ось 1)	%Q бит Смещение 15	FBSA* Бит записи 2
IN10_B (Ось 2 -ОТ)	Strobe1 Level (Ось 2)	%Q бит Смещение 24	FBSA* Бит записи 3
IN11_B (Переключатель исходного положения Оси 2)	Strobe2 Level (Ось 2)	%Q бит Смещение 25	FBSA* Бит записи 4
IN9_C (Ось 3 +ОТ)	Strobe1 Level (Ось 3)	%Q бит Смещение 40	Активный флажок Local Logic
IN10_C (Ось 3 -ОТ)	Strobe2 Level (Ось 3)	%Q бит Смещение 41	
IN11_C (Переключатель исходного положения Оси 3)	Strobe1 Level (Ось 4)	%Q бит Смещение 56	

\* FBSA является сокращение "Fast Backplane Status Access" ("Быстрый доступ к состоянию базовой платы") (Запрос на обслуживание #46). См. детали в GFK-0467L или более позднем издании.

## Output Bits (Выходные биты)

Таблица конфигурации Битов выхода позволяет пользователю сконфигурировать цифровые выходы лицевой панели DSM314 либо для управления программы Local Logic, либо для управления программы ПЛК. Параметры Битов выхода описаны в таблице 4-7. Дополнительную информацию по конфигурации Битов выхода см. в гл. 14.

Таблица 4-7. Таблица Битов выхода

Параметр конфигурации	Описание	Значения	По умолчанию	Ссылка
Out1_A Config	Источник управления Out1_A	Упр-ние ПЛК (%Q bit Offset 24) Упр-ние DSM (Digital Output1_1)	Упр-ние ПЛК	Гл. 14
Out3_A Config	Источник управления Out3_A	Упр-ние ПЛК (%Q bit Offset 25) Упр-ние DSM (Digital Output3_1)	Упр-ние ПЛК	Гл. 14
Out1_B Config	Источник управления Out1_B	Упр-ние ПЛК (%Q bit Offset 40) Упр-ние DSM (Digital Output1_2)	Упр-ние ПЛК	Гл. 14
Out3_B Config	Источник управления Out3_B	Упр-ние ПЛК (%Q bit Offset 41) Упр-ние DSM (Digital Output3_2)	Упр-ние ПЛК	Гл. 14
Out1_C Config	Источник управления Out1_C	Упр-ние ПЛК (%Q bit Offset 56) Упр-ние DSM (Digital Output1_3)	Упр-ние ПЛК	Гл. 14
Out3_C Config	Источник управления Out3_C	Упр-ние ПЛК (%Q bit Offset 57) Упр-ние DSM (Digital Output3_3)	Упр-ние ПЛК	Гл. 14
Out1_D Config	Источник управления Out1_D	Упр-ние ПЛК (%Q bit Offset 72) Упр-ние DSM (Digital Output1_4)	Упр-ние ПЛК	Гл. 14
Out3_D Config	Источник управления Out3_D	Упр-ние ПЛК (%Q bit Offset 73) Упр-ние DSM (Digital Output3_4)	Упр-ние ПЛК	Гл. 14

## Данные конфигурации оси

Параметры конфигурации Оси DSM314 определяют такие параметры, как отношение пользовательских единиц к количеству импульсов, Скорость толчка, Ускорение толчка, Конец перемещения и Пределы скорости. Ниже определены и кратко описаны параметры конфигурации для каждого режима контура управления. Информация в

столбце "Ссылка" указывает номера разделов в данной главе. Значения MaxPosnUu, MaxVelUu и MaxAccUu в приведенной ниже таблице могут быть рассчитаны с использованием формул таблицы 4-12 ("Расчет переменных, определяющих предельные значения данных").

Таблица 4-8. Данные конфигурации оси

Параметр конфигурации	Описание	Значения	По умолчанию	Ед.	Ссылка
User Units	Значение Пользовательских Единиц	1...65535	1	-	5.01
Counts	Единицы счета Обратной связи	1...65535	1	-	5.01
OT Limit Sw	Включатель выхода за границу	Включено Выключено	Включено	-	5.02
Drive Ready Input	Вход готовности устройства	Включено Выключено	Включено	-	5.03
High Position Limit	Верхний предел положения	-MaxPosnUu ...+MaxPosnUu-1*	+8388607	Пользоват. единицы	5.04
Low Position Limit	Нижний предел положения	-MaxPosnUu ...+MaxPosnUu-1*	-8388608	Пользоват. единицы	5.05
High Software EOT Limit	Верхний программный ограничитель движения	-MaxPosnUu ...+MaxPosnUu-1*	+8388607	Пользоват. единицы	5.06
Low Software EOT Limit	Нижний программный ограничитель движения	-MaxPosnUu ...+MaxPosnUu-1*	-8388608	Пользоват. единицы	5.07
Software End of Travel	Программный ограничитель движения	Выключено Включено	Выключено	-	5.08
Velocity Limit	Ограничение скорости	1...MaxvelUu	1000000	Пользоват. единицы / с	5.09
Command Direction	Команда направления	Два направления Только положительное Только отрицательное	Два направления	-	5.10
Axis Direction	Направление Оси	Нормальное Назад	Нормальное	-	5.11
Feedback Source	Обратная связь	По умолчанию Внешний импульсный датчик положения Внешний последовательный датчик положения	По умолчанию	-	5.12
Feedback Mode (Только Цифровой режим)	Режим обратной связи	Инкрементный Абсолютный	Инкрементный	-	5.13
Reversal Compensation	Компенсация Реверсирования	0...255	0	Пользоват. единицы	5.14
Drive Disable Delay	Задержка выключения привода	0...60000	100	мс	5.15
Jog Velocity	Скорость передвижения	1...MaxVelUu	+1000	Польз. ед. / с	5.16
Jog Acceleration	Ускорение передвижения	1...MaxAccUu*	+10000	Польз. ед. / с <sup>2</sup>	5.17
Jog Acceleration Mode	Тип ускорения передвижения	Линейный S-кривая	Линейный	-	5.18
Home Position	Начальная позиция	Нижняя граница позиции ... верхняя граница	0	Пользоват. единицы	5.19

Home Offset	Начальное смещение	-32768...+32767	0	Пользоват. единицы	5.20
Final Home Velocity	Скорость подхода к начальной позиции	1...MaxVelUu*	+500	Польз. ед. / с	5.21
Find Home Velocity	Скорость поиска начальной позиции	1...MaxVelUu*	+2000	Польз. ед. / с	5.22
Home Mode	Режим поиска начальной позиции	Переключатель Исходной позиции Движение + Движение -	Переключатель Исходной позиции	-	5.23
Return Data 1 Mode	Режим возврата данных 1	0...FF	0		5.24
Return Data 1 Offset	Смещение возврата данных 1	От -2 147 483 648 до 2 147 483 647	0		5.24
Return Data 2 Mode	Режим возврата данных 2	0...FF	0		5.24
Return Data 2 Offset	Смещение возврата данных 2	От -2 147 483 648 до 2 147 483 647	0		5.24
Cam Master Source	Мастер САМ	Команд. позиция 1 Текущая Позиция 1 Команд. позиция 2 Текущая Позиция 2 Команд. позиция 3 Текущая Позиция 3 Команд. позиция 4 Текущая Позиция 4	Текущая Позиция 3	-	5.25
Follower Control Loop	Связанный способ управления	Выключено Включено	Выключено	-	5.26
Ratio A Value	Значение коэффициента А	-32768...+32767	1	-	5.27
Ratio B Value	Значение коэффициента В	1...32767	1	-	5.27
Follower Master Source 1	Связанный мастер 1	Нет Команд. позиция 1 Текущая Позиция 1 Команд. позиция 2 Текущая Позиция 2 Команд. позиция 3 Текущая Позиция 3 Команд. позиция 4 Текущая Позиция 4	Нет	-	5.28
Follower Master Source 2	Связанный мастер 2	Нет Команд. позиция 1 Текущая Позиция 1 Команд. позиция 2 Текущая Позиция 2 Команд. позиция 3 Текущая Позиция 3 Команд. позиция 4 Текущая Позиция 4	Нет	-	5.29
Follower Enable Trigger	Связанный триггер разрешения	Нет CTL01-CTL32	Нет	-	5.30
Follower Disable Trigger	Связанный триггер запрещения	Нет	Нет	-	5.31

		CTL01-CTL32			
Follower Disable Action	Связанное действие запрещения	Стоп Инкрементная позиция Абсолютная позиция	Стоп	-	5.32
Ramp Makeup Acceleration	Формирование угла ускорения	1...MaxAccUu*	10 000	Польз. ед. / с <sup>2</sup>	5.33
Ramp Makeup Mode	Режим формирования угла	Интервал компенсации Скорость компенсации	Интервал компенсации	-	5.34
Ramp Makeup Time	Время формирования угла	0...32000	0	мс	5.35
Ramp Makeup Velocity	Скорость формирования угла	1...MaxVelUu*	+100,000	Польз. ед. / с	5.36

\* Для расчета значений MaxAccUu, MaxPosUu и MaxVelUu см. табл. 4-8.

**5.01 Пользовательские единицы отсчета.** Параметр "Число пользовательских единиц на единицу отсчета" задает число программных единиц измерения для каждого отсчета обратной связи по положению. Это дает возможность пользователю запрограммировать DSM314 в единицах измерения, специфических для приложения. Количество пользовательских единиц и число отсчетов должны быть в диапазоне от 1 до 65 535. Отношение пользовательских единиц к числу отсчетов должно быть в диапазоне от 8:1 до 1:32. Если, например, 10000 дюймов перемещения соответствует 8192 отсчетам обратной связи, то отношение пользовательских единиц к числу отсчетов, задаваемое как 1000:8192, устанавливает 1 пользовательскую единицу равной 0.001 дюйма. По умолчанию имеем 1:1.

Параметр "Число пользовательских единиц на единицу отсчета" задает число программных единиц измерения для каждого отсчета обратной связи по положению. Для механических систем, связанных с осью, это значение должно быть задано правильно; в противном случае движение будет происходить неточно и со сбоями.

**Примечание:** *Важно задать это соотношение в начале процесса конфигурации;* многие другие поля конфигурации определяются в пользовательских единицах измерения.

Например, Скорость определяется количеством пользовательских единиц измерения в секунду, а Ускорение – в секунду<sup>2</sup>.

Это отношение является очень удобным средством масштабирования. Отношение пользовательских единиц к отсчету может быть задано так, чтобы выполнить программирование для другого количества отсчетов, чем устанавливается по умолчанию. В упрощенном примере предположим, что приложение, использующее для обратной связи датчик положения, имеет такой датчик, который выдает 1000 импульсов отсчета на один оборот (250 линий); применение связано с устройством, проходящим один дюйм за оборот. Значением по умолчанию является одна тысячная дюйма на отсчет. Однако требуется написать программы и использовать модуль DSM300 Series с метрическими единицами измерения. Чтобы сделать это возможным, при конфигурации должно быть задано отношение 2540 пользовательских единиц на 1000 отсчетов. При таком отношении одна пользовательская единица будет равна 0.01 мм. 2540 пользовательских единиц дадут 25.40 мм (один дюйм) перемещения.

Следующий пример показывает, как следует рассчитывать значения чтобы соответствовать требованиям: числопользовательских единиц и число отсчетов находились в диапазоне от 1 до 65 535, а их отношение – в диапазоне от 8:1 до 1:32.

Основным уравнением, для соответствовать этому, следующее:

$$\frac{\text{Число пользовательских единиц}}{\text{Количество отсчетов}} = \frac{(\text{Перемещение нагрузки на оборот двигателя}) \div (\text{Требуемое разрешение в пользовательских единицах})}{\text{Количество отсчетов датчика положения на оборот двигателя}}$$

Числитель и знаменатель должны быть в пределах RANGE (ДИОПАЗОН). Полученное отношение должно соответствовать границам для RATIO (ОТНОШЕНИЕ). Десятичная точка всегда опускается; она не используется. Отношение пользовательских единиц к отсчетам всегда выражается отношение целых чисел.

**Пример применения** При использовании в конфигурации DSM314 отношения пользовательских единиц к отсчетам можно провести программирование в технических единицах, а не в отсчетах датчика положения. Как пример, предположим, что машина имеет двигатель с встроенным импульсным датчиком положения; двигатель через шестеренчатый редуктор соединен с цилиндрическим прямозубым колесом. Это колесо установлено на конце вала с жесткими прижимными роликами. Прижимной ролик перемещает листовой материал в устройстве резания. Программа движения определяет длину отрезаемой полосы. Требуется запрограммировать разрешение 0.01 дюйма.

Имеются следующие данные:

- Датчик разрешения на 2000 линий (x4 = 8000 отсчетов на оборот датчика)
- Понижающая передача 20:1
- Диаметр окружности цилиндрического прямозубого колеса составляет 14.336"
- Требуемая программная единица измерения в дюймах (0.01)

Хотя возможно использование нескольких подходов, но наиболее прямым являются расчеты на базе одного оборота цилиндрического прямозубого колеса.

1. Сначала определим число пользовательских единиц на оборот цилиндрического прямозубого колеса:

14.336" диаметра окружности колеса \*  $\pi$  (pi) = 45.0378" длина окружности

45.0378" / 0.01" требуемой пользовательской единицы = 4503.78 пользовательских единиц на оборот цилиндрического прямозубого колеса

2. Теперь определим отсчетов датчика положения на оборот цилиндрического прямозубого колеса:

$$2000 \text{ линий} * \frac{4 \text{ отсчета}}{\text{линия}} * \frac{20 \text{ оборотов двигателя}}{1 \text{ оборот цил. колеса}} = 160\,000 \text{ отсчетов датчика положения на оборот цилиндр. прямозубого колеса}$$

3. Проверим значение отношения числа пользовательских единиц к числу отсчетов. Это отношение должно быть от 8:1 до 1:32 (от 8 до 0.03125), а два числа должны быть в диапазоне от 1 до 65535.

4503.78 пользовательских единиц / 160 000 отсчетов датчика положения  
= 0.02815 или 1:35.5.

Такое отношение является слишком малым; поэтому требуется что-то изменить. Для решения проблемы можно изменить любой из указанных ниже компонентов системы:

- Увеличить диаметр прямозубого цилиндрического колеса до 15.92" или более.
- Уменьшить число линий датчика положения на оборот до 1800 или менее.
- Уменьшить коэффициент передачи редуктора до 18:1 или меньше.
- Изменить требуемую программную единицу измерения на 0.001 дюйма.

Совершенно очевидно, что наиболее легким путем является изменение требуемой программной единицы измерения на 0.001 дюйма.

4. Теперь пересчитаем число пользовательских единиц измерения на оборот, используя программную единицу измерения 0.001 дюйма.  $14.336" \text{ диаметр} * \pi = 45.0378" \text{ длина окружности}$   $45.0378" / 0.001" \text{ программируемой пользовательской единицы} = 45\,037.8 \text{ пользовательских единиц на оборот цилиндрического прямозубого колеса}$

Таким образом, отношение пользовательских единиц к числу отсчетов составляет  $45\,038 / 160\,000 = 0.2815$  или около 1:3.6, что находится в пределах допустимого диапазона.

Можно было бы использовать отношение  $45\,038 / 160\,000$ , однако  $160\,000$  превышает допустимое значение  $65\,535$ . Деление обоих чисел на 10 решает эту проблему, приводя к отношению  $4\,504 / 16\,000$ . Обратите внимание, что в этом примере мы просто убрали дробь и пренебрегли небольшой ошибкой округления.

Одним из методов устранения "округления" является запись числового отношения в виде дроби. В приведенном выше примере может быть использовано любое число, дающее отношение 0.2815. Например,  $2815 / 10000$ .

Другим подходом является рационализация дроби (сведение ее к наименьшим членам). Это выполняется последовательным делением как числителя, так и знаменателя на последовательно уменьшающиеся простые числа, начиная с простого наибольшего числа, на которое делится как числитель, так и знаменатель, и заканчивая таким, когда деление без остатка дальше невозможно.

При конфигурации отношения числа пользовательских единиц к числу отсчетов следует с целью увеличения точности всегда сохранять дробь из целых чисел, десятичное отношение, выраженное в виде дроби, или рационализированную дробь. Пользователь должен определить, насколько существенна ошибка округления, если таковая имеет место. Приложение, работающее в режиме вращения, которое всегда действует в одном направлении, будет накапливать ошибки округления со временем и "дрейфовать". Линейное приложение будет накапливать только ошибку для длины перемещения, после чего "смаывать" ее при реверсе оси.

- 5.02 Overtravel Limit Switch (Включатель выхода за границу).** Задаст, использует ли модуль DSM300 Series входы аппаратного концевого выключателя перебега.

**ВЫКЛЮЧЕНО**, входы выключателя выхода за границу на лицевой плате (IN09 и IN10) могут быть использованы как входы общего назначения для контроля работы программы движения и для ветвления программ (назначаются на CTL01-CTL24).

**ВКЛЮЧЕНО**, показывает, что DSM300 будет проверять входы выключателя выхода за границу постоянно, каждые 10 мс до тех пор, пока на входе %I *Drive Enabled* имеется логическая единица. Если какой-либо из концевых выключателей разомкнется (на входе появится логический ноль, "Выкл."), то на все движения будет немедленно выдана команда останова. Управление замедлением отключено; выдается команда нулевой скорости сервопривода. Твердотельное реле включения оси не разомкнется до тех пор, пока после %Q *Enable Drive* команды не выставлено на ноль. Код ошибки, указывающий, какой концевой выключатель сработал, передается в %AI *Axis Error Code* (*Код ошибки оси*). В этом момент разрешено только одно действие DSM314: могут быть одновременно использованы, биты %Q *Jog* (*Толчок*) и %Q *Clear Error* (*Удалить ошибку*), чтобы отойти от концевого выключателя. Бит %Q *Clear Error* должен быть в состоянии ON (ВКЛ.), чтобы иметь возможность толчком отойти от концевого выключателя. Пользователь может также вручную отвести выключенную ось от концевого выключателя. Нормальная работа может быть продолжена после удаления аварийного сообщения.

### Предостережение

**Включающие цифровые и аналоговые (Ц/А) команды игнорируют концевые выключатели и должны использоваться с осторожностью.**

- 5.03 Вход Drive Ready (Вход готовности устройства).** Включает или выключает вход Drive Ready (Вход готовности устройства) для аналогового сервоустройства. Этот пункт конфигурации пропускается для цифрового сервоустройства или для вспомогательной оси. Если вход Drive Ready включен, то сигнал входа лицевой панели Drive Ready (IN4) должен быть включен (выставлен на 0 В) **в течение 1 секунды** после включения бита *Enable Drive* %Q. Если вход лицевой панели Drive Ready выключается, в то время как бит *Drive Enabled* %I включен, выдается код ошибки C0h, и ось останавливается. Уставка входа Drive Ready должна быть выставлена на "Выключено" для аналогового сервоустройства, которое не обеспечивает совместимый выходной сигнал Drive Ready.
- 5.04 High Position Limit (Верхний предел положения).** (Пользовательские единицы). При движении в положительном направлении при достижении нижней границы произойдет выход значения Текущей позиции за ее пределы. **Параметры "предел положения" могут быть использованы для приложений, в которых имеет место постоянное вращение, если параметр конфигурации Software End of Travel (Программный ограничитель движения) установлен на Выключено.** Верхний предел положения всегда должна быть установлена в пользовательских единицах на один цикл меньше, чем требуется. Например, машина с 360° должна была бы иметь уставку верхней границы позиции на 359. На следующем отсчете после 359 отсчет перейдет на значение, заданное в параметре Нижний предел положения (0 в данном примере). **Для правильной работы абсолютное значение перехода (Верхний предел положения – Нижний предел положения + 1) всегда должно быть больше расстояния, проходимого осью за один интервал дискретизации контура позиционирования (обычно 2 мс).** См. в приложении информацию по использованию датчика положения в абсолютном режиме. Значение по умолчанию: 8 388 607.
- 5.05 Low Position Limit (Нижний предел положения).** Нижний предел положения (пользовательские единицы). При движении в отрицательном направлении при достижении Верхнего предела положения произойдет выход значения Текущей позиции за ее пределы. **Параметры "предел положения" могут быть использованы для приложений, в которых имеет место постоянное вращение, если параметр**

конфигурации **Software End of Travel** (Программный ограничитель движения) установлен на **Выключено**. Для правильной работы абсолютное значение перехода (Верхний предел положения – Нижний предел положения + 1) всегда должно быть больше расстояния, проходимого осью за один интервал дискретизации контура позиционирования (обычно 2 мс). См. в приложении С информацию по использованию датчика положения в абсолютном режиме. Значение по умолчанию: - 8 388 608.

**5.06 High Software EOT Limit (Верхний программный ограничитель движения).** Верхний программный ограничитель движения (пользовательские единицы). Если эта граница включена, а DSM314 запрограммирован так, чтобы выйти на позицию, превышающую Верхний программный ограничитель движения, то будет выдана ошибка, и DSM314 не сможет осуществить движение оси. Если Следящий контур управления включен, то Верхний программный ограничитель движения игнорируется для движения подчиненной оси, возникающего под действием команд ведущей оси. Эта граница действует только по отношению к движению подчиненной оси, возникающему под действием внутренних программных команд толчка и движения. **Эта граница всегда игнорируется для команд *Move at Velocity %AQ* (Движение на скорости).** Значение по умолчанию: +8 388 607.

**В аналоговом или цифровом серворежиме** верхний программный ограничитель движения используется только тогда, когда параметр конфигурации **Software End of Travel** (Программный ограничитель движения) установлен на **Включено**. Если Верхний программный ограничитель движения включен и если его значение больше, чем Верхний предел положения, то внутренними средствами происходит изменение первой на значение второй. При этом выдается код ошибки 17h, указывающий на подстройку границы. **Верхний Программный ограничитель движения игнорируется для команд *Jog*, если бит *Position Valid %I* выключен.**

**В режиме Вспомогательной оси** верхний программный ограничитель движения имеет особое назначение, зависящее от установки параметра **Software End of Travel** (Программный ограничитель движения):

Параметр **Software End of Travel** установлен на **Включено** – программы движения и команды *Jog* (Толчок) ограничены значением Верхний Программный ограничитель движения. Команда *Move at Velocity %AQ* может создать *Commanded Position* (Заданную позицию), выходящую за указанную границу. *Заданная позиция* может выйти за максимально положительные и отрицательные значения позиции (- 2 147 483 648 ... +2 147 483 647 при масштабе 1:1).

Параметр **Software End of Travel** установлен на **Выключено** – Верхний Программный ограничитель движения используется как значение перехода для *Commanded Position* (Заданная позиция). Программа движения, команды *Jog* (Толчок) и *Move at Velocity* (Движение на скорости) - все они могут создавать переход *Заданной позиции на Верхний программный ограничитель движения*.

**5.07 Low Software EOT Limit (Нижний программный ограничитель движения).** Нижний программный ограничитель движения (пользовательские единицы). Если эта граница включена, а DSM314 запрограммирован так, чтобы выйти на позицию, находящуюся ниже Нижний программный ограничитель движения, то будет выдана ошибка, и DSM314 не сможет осуществить движение оси. Если Следящий контур управления включен, то Верхний программный ограничитель движения игнорируется для движения подчиненной оси, возникающего под действием команд ведущей оси. Эта граница действует только по отношению к движению подчиненной оси, возникающему под действием внутренних программных команд толчка и движения. **Эта граница всегда игнорируется для команд *Move at Velocity %AQ* (Движение на скорости).** Значение по умолчанию: -8 388 608

**В аналоговом или цифровом серворежимах** Нижний программный ограничитель движения используется только тогда, когда параметр конфигурации **Software End of Travel (Программный ограничитель движения)** установлен на Включено. Если Нижний программный ограничитель движения включена и если ее значение меньше, чем Нижний предел положения, то внутренними средствами происходит изменение первой на значение второй. При этом выдается код ошибки 17h, указывающий на подстройку границы. **Нижний программный ограничитель движения игнорируется для команд Jog, если бит Position Valid %I выключен.**

**В режиме Вспомогательной оси** Нижний программный ограничитель движения имеет особое назначение, зависящее от установки параметра **Software End of Travel (Программного ограничитель движения)**:

Параметр **Software End of Travel** установлен на Включено – программы движения и команды *Jog (Толчок)* ограничены значением Нижнего программного ограничителя движения. Команда *Move at Velocity %AQ* может создать *Commanded Position (Заданную позицию)*, выходящую за указанную границу. *Заданная позиция* может выйти за максимально положительные и отрицательные значения позиции (- 2 147 483 648 ... +2 147 483 647 при масштабе 1:1).

Параметр **Software End of Travel** установлен на Выключено – Нижний программный ограничитель движения используется как значение перехода для *Commanded Position (Заданная позиция)*. Программа движения, команды *Jog (Толчок)* и *Move at Velocity (Движение на скорости)* - все они могут создавать переход *Заданной позиции* на Нижний программный ограничитель движения.

#### **5.08 Software End of Travel (Программный ограничитель движения).**

Включает или выключает Верхний и Нижний программный ограничитель движения. Значение по умолчанию: Выключено

#### **5.09 Velocity Limit (Ограничение скорости скорости).**

Ограничение скорости оси (пользовательские единицы/с). Параметр Ограничение скорости действует на сумму всех источников команд скорости для оси, включая команды внутреннего генератора траектории и команды внешней ведущей оси слежения. Если команда скорости сервопривода превысит данную границу, то **будет выдан код ошибки F2h**, а сервокоманда будет внутренними средствами установлена на заданное предельное значение. Значение по умолчанию: 1 000 000

#### **5.10 Command Direction (Команда направления).**

Позволяет сконфигурировать ось для работы в одном или в двух направлениях. Если выбрана работа в одном направлении (только положительное или только отрицательное), то сервокоманды противоположного направления не будут направляться на контур позиционирования сервоустройства. Значение по умолчанию: два направления

#### **5.11 Axis Direction (Направление Оси).**

Сконфигурированное направление оси при Нормальной ситуации для всех цифровых сервоустройств GE Fanuc определяется как положительное при вращении вала двигателя против часовой стрелки (CCW), если смотреть на вал двигателя. Сконфигурированное направление оси при Реверсе определяется как положительное при вращении вала двигателя по часовой стрелке (CW).

Сконфигурированное направление оси при Нормальной ситуации для аналоговых сервоустройств определяется как положительное, если канал А датчика положения опережает канал В. Сконфигурированное направление оси при реверсе определяется как положительное, если канал В датчика положения опережает канал А. На практике конфигурация направления оси дает возможность пользователю легко изменять направление движения, создаваемое всеми командами, без изменения программы движения. Значение по умолчанию: Нормальное

- 5.12 Feedback Source (Обратная связь).** Этот параметр конфигурации не используется в данной микропрограмме модуля DSM314. Его следует установить на "По умолчанию".
- 5.13 Feedback Mode (Режим обратной связи).** Используется только в том случае, если Axis Mode (Режим Оси) установлен на Digital Servo (Цифровой Сервопривод). Этот параметр позволяет задать Инкрементный или Абсолютный тип обратной связи для последовательного датчика положения GE Fanuc. "Инкрементный" означает, что последовательный датчик положения используется как инкрементный и что аварийные сигналы по состоянию аккумулятора датчика положения не выдаются. "Абсолютный" означает, что последовательный датчик положения используется как абсолютный (установлен аккумулятор сохранения информации), который сохраняет позицию при отключении питания системы. В Абсолютном режиме происходит выдача аварийных сигналов по состоянию аккумулятора. Дополнительную информацию см. в приложении С "Устройства обратной связи по положению". Значение по умолчанию: Инкрементный
- 5.14 Reversal Compensation (Компенсация Реверсирования).** Параметр компенсации, который в системах, имеющих люфты, позволяет сервоприводу изменять направление и в то же время обеспечивать точность позиционирования. Люфт проявляется для серводвигателя в том, что при изменении направления движения серводвигатель должен пройти некоторое небольшое расстояние ("мертвый" ход) до того, как нагрузка начнет перемещаться. Например, рассмотрим дверной замок с засовом, открываемым ключом. Представим себе, что сервопривод приводит в движение ключ в замке, а по цепи обратной связи поступает информация о перемещении засова. Когда сервопривод вращает ключ против часовой стрелки, то засов двигается влево. Однако когда сервопривод вращает ключ по часовой стрелке, засов не двигается до тех пор, пока ключ не дойдет до определенной точки. Параметр Reversal Compensation (Компенсация реверсирования) добавляет необходимый "холостой" ход для быстрого перемещения сервопривода в точку, где начинается движение, обнаруживаемое через устройство обратной связи. Модуль DSM314 отнимает участок компенсации, если поступает команда на движение в отрицательном направлении, и добавляет участок компенсации, прежде чем начать движение в положительном направлении.  
Значение по умолчанию: 0.

**Примечание:** Компенсация реверсирования не может быть использована, если параметр конфигурации *Follower Control Loop (Следящий Контур Управления)* установлен на Включено.

- 5.15 Drive Disable Delay (Задержка выключения привода Привода).** Задержка выключения сервопривода (миллисекунды). Эта задержка представляет собой интервал времени с момента получения команды нулевой скорости до момента выключения сигнала, разрешающего работу привода (цифровое сервоустройство MCON). Задержка выключения является эффективной, если выключен бит *Enable Drive %Q* или если возникла какая-то ошибочная ситуация (Стоп-режим). Величина задержки выключения должна быть больше, чем длительность торможения сервопривода от максимальной скорости в наиболее неблагоприятной ситуации. Находясь в состоянии OFF (ВЫКЛ.), бит *Enable Drive %Q* останавливает управление сервоприводом со стороны DSM314; но бывают ситуации, когда сигнал разрешения работы привода должен оставаться в состоянии ON (ВКЛ.). Например, если сервопривод приближается к границе конца передвижения, а сигнал разрешения работы привода в этот момент выключается из-за ошибки, то сервопривод может продолжить движение, пока не упрется в упор. Тогда, чтобы дать возможность модулю DSM314 выдать команды и обеспечить быструю остановку, длительность задержки выключения привода должна быть больше длительность торможения сервопривода с максимальной скорости.

Задержка выключения может быть использована для управления и в том случае, когда с двигателя вала снимается момент вращения. Приложения, использующие

электромеханический тормоз, требуют, в общем случае, время для срабатывания тормоза, прежде чем произойдет снятие момента вращения сервопривода. Величина задержки в этом случае должна быть установлена больше, чем время срабатывания тормоза. Значение по умолчанию: 100.

**5.16 Jog Velocity (Скорость передвижения).** Скорость передвижения (пользовательские единицы/секунды) Скорость, с которой двигается сервопривод при операции *Толчок*. *Jog Velocity (Скорость передвижения)* используется в программах движения, если в программу не включена команда Velocity (Скорость). *Jog Velocity* всегда используется командой %AQ Move (27h). Значение по умолчанию: 1000.

**5.17 Jog Acceleration (Ускорение передвижения).** Величина Ускорения передвижения (пользовательские единицы/секунды<sup>2</sup>) Значение ускорения и торможения используется в операциях *Jog (Толчок)*, *Find Home (Поиск Исходной Позиции)*, *Move at Velocity (Движение на скорости)*, *Abort All Moves (Прекратить все движения)* и **Normal Stop (Нормальный останов)**. **Normal Stop (Нормальный останов)** происходит, когда ПЛК переключается от Run на Stop или после определенных программных ошибок (см. Приложение А). *Jog Acceleration (Ускорение Толчка)* используется в программах движения, если в программу не включена команда Acceleration (Ускорение). *Jog Acceleration* всегда используется командой %AQ Move (27h). Значение *Jog Acceleration* должно быть задано достаточно большим, чтобы операции *Abort all Moves* и **Normal Stop** выполнялись удовлетворительно. Значение по умолчанию: 10000.

**Примечание:** В модуле DSM314 используется минимальное значение после масштабирования. Это значение определяется следующим правилом: Jog Acc \* (пользовательские единицы/отсчет) >= 32 отсчета/секунда<sup>2</sup>.

**5.18 Jog Acceleration Mode (Тип ускорения передвижения).** Тип ускорения передвижения (ЛИНЕЙНЫЙ или S-КРИВАЯ). Тип ускорения для операций *Jog (Толчок)*, *Find Home (Поиск Исходной Позиции)*, *Move at Velocity (Движение на скорости)*, *Abort All Moves (Прекратить все движения)* и **Normal Stop (Нормальный останов)**. **Normal Stop (Нормальный останов)** происходит, когда ПЛК переключается от Run на Stop или после определенных программных ошибок (см. Приложение А). ЛИНЕЙНЫЙ тип (постоянное ускорение) вызывает изменение скорости линейно по времени. Тип S-КРИВОЙ (ускорение, ограничивающее величину рывка) характеризуется меньшими значениями изменения скорости в начале и в конце интервала ускорения, чем в линейном режиме. **Движения, использующие S-кривую для ускорения, требуют вдвое больше времени и расстояния для изменения скорости по сравнению с движениями, использующими то же значение ускорения в линейном режиме.** Чтобы сохранить время цикла машины неизменным, профили движения с S-кривой требуют наличия значения ускорения (и пикового момента вращения двигателя), вдвое превышающего значение для эквивалентного режима движения с линейным ускорением. Поэтому может оказаться необходимым выбор оптимального соотношения между стоимостью двигателя и длительностью машинного цикла. Значение по умолчанию: ЛИНЕЙНЫЙ.

**5.19 Home Position (Начальная позиция).** Начальная позиция (Пользовательские единицы). Это значение приписывается *Commanded Position (Заданная позиция)*, когда заканчивается цикл *Find Home (Поиск Начальная позиция)*.

**5.20 Home Offset (Начальное смещение).** Начальное смещение (Пользовательские единицы). Значение, прибавляемое к конечной точке остановки сервопривода после завершения цикла *Find Home* или отнимаемое от нее. **Home Offset (Начальное смещение)** позволяет подстроить конечную точку остановки сервопривода относительно маркера датчика положения. Более детальную информацию по циклу исходной позиции см. в гл. 6. Значение по умолчанию: 0.

- 5.21 Find Home Velocity (Скорость поиска начальной позиции).** Скорость поиска начальной позиции (пользовательские единицы/секунды). Скорость, с которой сервопривод ищет начальные переходы Home Switch (Переключатель Исходной позиции) во время цикла *Find Home*, если Home Mode (Режим поиска начальной позиции) сконфигурирован на HOMESW. Если требуется, может быть установлено большее значение Скорости поиска исходной позиции, чтобы позволить сервоприводу быстро обнаружить Датчик Исходной позиции. Значение по умолчанию: 2000
- 5.22 Final Home Velocity (Скорость подхода к начальной позиции)** Конечная Скорость подхода к начальной позиции (пользовательские единицы/секунды). Это скорость, с которой сервопривод ищет конечный переход датчика исходной позиции и импульс Маркера Датчика положения в конце цикла Find Home. Эта скорость используется также для режимов цикла исходной позиции MOVE+ и MOVE-. Более детальную информацию по циклу исходной позиции см. в гл. 6. Конечная Скорость Прихода в Исходную позицию должна быть достаточно малой, чтобы сделать возможной задержку 10 миллисекунд (время фильтрации) между конечным переходом Переключателя Исходной позиции и импульсом Маркера Датчика положения. Значение по умолчанию: 500
- 5.23 Home Mode (Режим поиска начальной позиции).** Find Home Mode (Режим поиска начальной позиции). Этот метод используется, чтобы найти исходную позицию во время цикла *Find Home (Поиск начальной Позиции)*. HOME SWITCH (ДАТЧИК НАЧАЛЬНОЙ ПОЗИЦИИ) указывает, что для Поиска Исходной позиции должен быть выведен на монитор датчик начальной позиции. MOVE+ и MOVE- определяют прямое положительное или отрицательное перемещение к следующему маркеру датчика положения на Скорости подхода к начальной позиции. Более детальную информацию по Циклу исходной позиции, Переключателю исходной позиции, по режимам Move+ и Move- см. в гл. 6. Значение по умолчанию: HOMESW.
- 5.24 Режим 1 Возврата Данных 1 и Смещение, Режим 2 Возврата Данных и Смещение.** Эти параметры конфигурации позволяют выводить в отчет альтернативные данные, подлежащие выводу в отчет в раздел *User Selected Data 1 (Данные 1, выбранные пользователем)* и *User Selected Data 2 (Данные 2, выбранные пользователем)* %AI для каждой оси. Альтернативные данные содержат такую информацию, как содержание памяти Параметров и редакция микропрограммы DSM314.
- Имеются два параметра конфигурации *Return Data (Возвращаемые данные)*: выбор режима и выбор смещения. Параметр режима позволяет выбрать тип Возвращаемых данных. Параметр смещения используется только в том случае, если выбран режим Parameter Data (Данные Параметра) (18h). Режим по умолчанию = 0 (Команда Момент вращения). Смещение по умолчанию = 0. Разрешается следующий выбор Возвращаемых данных:

Таблица 4-9. Возвращаемые данные, выбираемые пользователем

Цифровой	Аналог. момент вращения	Аналог. скорость	Выбираемые Возвращаемые данные	Режим данных	Смещение данных
Да	Да	Нет	Команда Момент вращения	00h	Не используется
Да	Да	Да	Редакция микропрограммы DSM	10h	Не используется
Да	Да	Да	Встроенный идентификационный номер микропрограммы DSM (шестнадцатеричный)	11h	Не используется
Да	Нет	Нет	Абсолютное смещение обратной связи (в отсчетах)	17h	Не используется

Да	Да	Да	Данные параметра	18h	Номер параметра (0-255)
Да	Да	Да	Биты CTL 1-32	19h	Не используется
Да	Да	Да	Аналоговые входы – Ось 1	1Ch	Не используется
Да	Да	Да	Аналоговые входы – Ось 2	1Dh	Не используется
Да	Да	Да	Аналоговые входы – Вспомогательная ось 3	1Eh	Не используется
Да	Да	Да	Аналоговые входы – Вспомогательная ось 4	1Fh	Не используется
Да	Да	Да	Заданная Позиция (пользовательские единицы)	20h	Не используется
Да	Да	Да	Заданная позиция программы слежения (в отсчетах)	21h	Не используется
Да	Да	Да	Не скорректированная Текущая позиция (в отсчетах)	28h	Не используется
Да	Да	Да	Не скорректированная позиция строба 1 (в отсчетах)	29h	Не используется
Да	Да	Да	Не скорректированная позиция строба 2 (в отсчетах)	2Ah	Не используется

**Команда Момент вращении** масштабирована так, что  $\pm 10000 = \pm 100\%$  момента вращения.

**Контроль микропрограммы DSM** понимается как два отдельных слова для старшего и младшего кодов контроля.

**Встроенный идентификационный номер микропрограммы DSM** понимается как одно шестнадцатеричное слово.

**Абсолютное смещение обратной связи** представляет собой смещение (в отсчетах), которое используется для инициализации Текущей позиции, если применяется цифровой Абсолютный Датчик положения GE Fanuc. Текущая позиция = Данные Абсолютного датчика + Абсолютное смещение обратной связи

**Аналоговые входы** обеспечивают два слова данных для каждой оси: младшее слово = AIN1 и старшее слово = AIN2. Эти данные масштабированы так, что  $\pm 32000 = \pm 10.0$  В.

**Заданная позиция (пользовательские единицы)** является копией данных *Commanded Position (Заданная позиция) %AI*, представляемых для каждой оси. См. раздел 2.04 в главе 5.

**Заданная позиция программы слежения** (в отсчетах) представляет собой действующую заданную позицию (в отсчетах обратной связи), обновляемую и используемую внутренним генератором команд движения. См. гл. 9 – Комбинированное движение под действием команд движения и команд следящего устройства.

**Нескорректированная текущая позиция** представляет собой накопленную текущую позицию (в отсчетах, не в пользовательских единицах) с 32-битовым двоичным значением от  $-2\ 147\ 483\ 648$  до  $+2\ 147\ 483\ 647$ .

**Нескорректированная позиция строба 1** представляет собой *Нескорректированную текущую позицию*, захваченную Стробом 1.

**Нескорректированная позиция строба 2** представляет собой *Нескорректированную текущую позицию*, захваченную Стробом 2.

Должно пройти не менее трех циклов ПЛК или 10 миллисекунд (в зависимости оттого, что больше), прежде чем новые Выбранные Возвращаемые данные станут доступными в ПЛК.

- 5.25 Cam Master Source (Мастер CAM).** Этот параметр конфигурации не используется в данной микропрограмме модуля DSM314.
- 5.26 Follower Control Loop (Связанный способ управления).** Если этот параметр конфигурации установлен на Включено, то сервоось будет отслеживать вход ведущей оси дополнительно к стандартным функциям движения, создаваемые внутренним генератором. Значение по умолчанию: Выключено
- 5.27 Ratio A Value and Ratio B Value (Значение параметра А и Значение параметра В).** (Следящий способ управления должен быть Включен). Отношение А к В задает передаточное число для следящей подчиненной оси по отношению к ведущей.

$$\text{Движение следящей оси (отсчеты)} = \frac{\text{А * Отсчеты ведущей опорной оси}}{\text{В}}$$

Диапазон для А - от -32 768 до +32 767, а для В от 1 до +32 767. Если А отрицательное, то подчиненная ось будет двигаться в направлении, противоположном движению ведущей оси. Микропрограмма DSM поддерживает отношения слежения подчиненная / ведущая ось (А/В) в диапазоне от 32:1 до 1:10 000. Значение по умолчанию: 1:1.

- 5.28 Follower Master Source 1 (Связанный мастер 1).** (Связанный способ управления должен быть Включен). Конфигурирует источник 1 ведущей оси следящего устройства. Разрешенным выбором являются Заданная или Текущая позиция для любой из 4 осей (если эта ось является конфигурируемой). Ведущий источник 1 следящего устройства является активным, если бит *Follower Master Source Select (Выбор Ведущего Источника Следящего Устройства) %Q* установлен в положение OFF (ВЫКЛ.).

**Заданная позиция или Текущая позиция подчиненной оси не должны выбираться в качестве ведущего источника для этой оси.** Если как Ведущий Источник 1 следящего устройства выбрана неконфигурированная ось, то это игнорируется. Значение по умолчанию: Нет. Информацию по режиму слежения см. в гл. 8.

- 5.29 Follower Master Source 2 (Ведущий Источник 2 Следящего Устройства).** (Следящий контур управления должен быть Включен). Конфигурирует источник 2 ведущей оси следящего устройства. Разрешенным выбором являются Заданная или Текущая позиция для любой из 4 осей (если эта ось является конфигурируемой). Источник 2 следящего устройства является активным, если бит *Follower Master Source Select (Выбор Ведущего Источника Следящего Устройства) %Q* установлен в положение ON (ВКЛ.).

**Заданная позиция или Текущая позиция подчиненной оси не должны выбираться в качестве ведущего источника для этой оси.** Если как Ведущий Источник 2 следящего устройства выбрана не сконфигурированная ось, то это игнорируется. Значение по умолчанию: Нет. Информацию по режиму слежения см. в гл. 8.

- 5.30 Follower Enable Trigger (Связанный триггер разрешения).** Вход Триггера Включения Связанного Устройства. Выбирается управляющим битом, CTL01-CTL32, который должен быть использован в качестве Входа Связанного триггера разрешения. Ось следящего устройства включена, когда выбранные переходы входа триггера ON (ВКЛ.) и бит *Enable Follower (Включение Следящего Устройства) %Q* также в положении ON (ВКЛ.). После того как Следящее устройство включено, бит ПЛК *Enable Follower %Q* и дополнительный бит триггера Выключения следящего устройства

управляют активным состоянием следующей далее функции. Наличие "Нет" означает, что ось следящего устройства выключается только битом *Enable Follower %Q*. Значение по умолчанию: Нет.

**5.31 Follower Disable Trigger (Триггер Выключения Следящего Устройства).** Вход Триггера Выключения Следящего Устройства. Выбирается управляющий бит, STL01-STL32, который должен быть использован в качестве Входа Триггера Выключения Следящего Устройства. Вход триггера проверяется только в том случае, когда бит *Enable Follower %Q* находится в состоянии ON (ВКЛ.). Если бит *Enable Follower %Q* находится в состоянии ON (ВКЛ.), то переход бита триггера от OFF (ВЫКЛ.) к ON (ВКЛ.) выключает следящее устройство. Установка бита *Enable Follower %Q* в состояние OFF (ВЫКЛ.) приводит к немедленному выключению следящего устройства, независимо от выключающей конфигурации триггера. Значение по умолчанию: Нет.

**5.32 Follower Disable Action (Связанное действие запрещения).** "Stop" ("Стоп") означает немедленное торможение следящего устройства до нулевой скорости в соответствии с заданным при конфигурации изменением ускорения. "Inc Position" ("Инкрементная Позиция") означает, что следящее устройство продолжает двигаться со скоростью, действующей в данный момент, затем тормозится и останавливается после прохождения определенного расстояния. Приращение позиции определено в регистре параметров для каждой оси:

P227 = Приращение позиции по Оси 1  
 P235 = Приращение позиции по Оси 2  
 P242 = Приращение позиции по Оси 3  
 P250 = Приращение позиции по Оси 4

Приращение позиции представляет собой полное изменение текущей позиции с точки, когда следящее устройство было выключено, до точки, когда оно остановилось.

Конфигурация "Abs Position" ("Абсолютная позиция") не поддерживается в данной микропрограмме DSM314. Значение по умолчанию: Stop (Стоп)

**5.33 Ramp Makeup Acceleration (Формирование угла ускорения).** Линейное формирование угла ускорения устройства (пользовательские единицы/с<sup>2</sup>) Определяет изменение скорости (ускорение или замедление), используемое для следующих целей:

- Ускорение следящей оси для достижения скорости ведущей оси после включения следящего устройства (участок АВ на рис. 4-1)
- Компенсация отсчетов команды ведущей оси, потерянных во время разгона следящего устройства (участки ВС и DE на рис. 4-1)
- Торможение для остановки после выключения следящего устройства (участок FG на рис. 4-1).

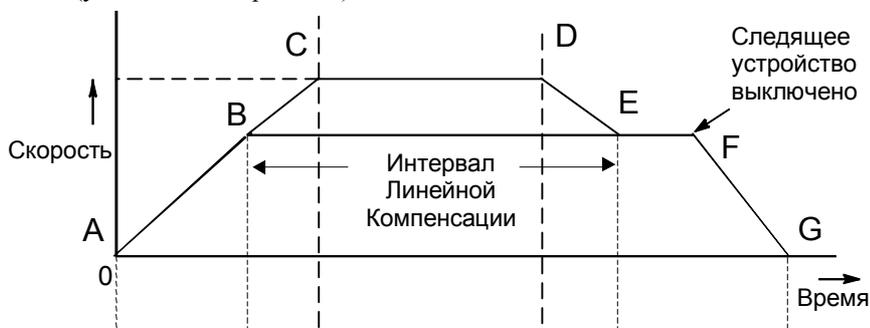


Рис. 4-1. График скорости во время линейного цикла следящего устройства

**5.34 Ramp Makeup Mode (Режим формирования угла).** Можно выбрать Makeup Time (Интервал компенсации) или Makeup Velocity (Скорость компенсации), объяснение которых дано далее.

- **Режим Интервала Компенсации** – в этом режиме длительность процесса компенсации равна интервалу времени, заданному параметром Ramp Makeup Time (Интервал Линейной Компенсации) (см. рис. 4-1). Этот режим является режимом по умолчанию.
- **Режим Скорости Компенсации** – этот режим зарезервирован для будущего использования.

**5.35 Ramp Makeup Time (Время формирования угла).** Время формирования угла Изменения Скорости Следящего Устройства (миллисекунды). Задаёт длительность промежутка времени в миллисекундах, используемого для компенсации отсчетов команды ведущей оси, потерянных во время линейного ускорения следящего устройства. Если коррекция расстояния оказывается невозможной в течение сконфигурированного интервала компенсации (из-за малости заданного значения), то время коррекции будет больше и выдается предупреждающее сообщение об ошибке. Этот установочный параметр имеет влияние только в том случае, если Ramp Makeup Mode (Режим Линейной Компенсации) установлен на Makeup Time (Интервал Компенсации).

Если требуется линейное ускорение без какой-либо коррекции потерянных отсчетов, то Makeup Time должно быть установлено на 0. В этом случае двигатель синхронизирует скорость по отношению к ведущей оси, но не будет корректировать какое-либо отклонение позиции, которое возникнет в ходе ускорения следящей оси.

Минимальное значение для Makeup time (Интервал компенсации) составляет 10; если вводится значение в диапазоне от 1 до 10, то вместо него используется значение 10.

Значение по умолчанию: 0.

Более детальное обсуждение этой функции дано в гл. 8 "Движение следящего устройства" раздел "Управление линейным изменением скорости следящей оси".

**5.36 Ramp Makeup Velocity (Скорость Формирования угла).** Это поле зарезервировано.

## Tuning Data (Данные настройки)

Таблицы настройки DSM314 используются для конфигурации данных настройки сервооси. В этих таблицах приводятся данные по конфигурации таких параметров, как Motor Type (Тип двигателя), Velocity at Max Cmd (Скорость при максимальной команде), Velocity Feed Forward Percentage (Процент упреждения по скорости подачи), and Position Loop Time Constant (Постоянная Времени Контур Позиционирования). В окне конфигурации DSM314 может присутствовать от одной до четырех таблиц настройки – по одной таблице для каждой оси **Сервопривода**, конфигурируемой в данной таблице установочных параметров.

Информация в столбце "Ссылка" приведенной ниже таблицы указывает номера разделов в данной главе.

Таблица 4-10. Параметры таблицы настройки

Параметр конфигурации	Описание	Значения	По умолчанию	Ед.	Ссылка
Motor Type	Тип двигателя	0...65 535	0	-	6.01
Analog Servo Command	Команда Аналогового Сервопривода	Скорость Момент (Примечание 1)	Скорость	-	6.02
Position Error Limit	Предел Ошибки позиционирования	100...60 000	60 000	Число пользовательских единиц	6.03
In Position Zone	В зоне позиционирования	1...60 000	10	Число пользоват. единиц	6.04
Pos Loop Time Constant	Постоянная Времени Позиционирования	0...65 535	1 000	0.1 мс	6.05
Velocity at MaxCmd	Скорость при Максимальном значении	256.. MaxVelUu (Прим. 2)	100 000	Число пользоват. единиц	6.06
Velocity Feed Forward Percentage	Коэффициент упреждения скорости	0...12 000	0	0.01%	6.07
Acceleration Feed Forward Percentage	Коэффициент упреждения ускорения	0...12 000	0	0.01%	6.08
Integrator Mode	Режим Интегратора Контур Позиционирования	Off (Выкл.) Continuous (Постоянно) Servo Null (Нуль Сервопривода)	Off (Выкл.)	-	6.09
Integrator Time Constant	Постоянная Времени интегрирования Контур Позиционирования	0...10 000	0	мс	6.10
Velocity Loop Gain	Управляющий коэффициент скорости	0...65 535	16	-	6.11

**Примечание 1:** Режим Момент вращение поддерживается в микропрограмме DSM версии 3.0 или более поздней.

**Примечание 2:** По расчету MaxVelUu см. таблицу 4-8.

**6.01 Motor Type (Тип двигателя).** Выбирается тип серводвигателя переменного тока FANUC, который должен быть использован вместе с DSM314 ТОЛЬКО в Цифровом режиме. DSM314 хранит в своей памяти таблицы установочных параметров для всех поддерживаемых сервоприводов FANUC. Выбор типа двигателя 0 выключает цифровое сервоуправление со стороны DSM314 для цифровой сервооси. **Тип двигателя должен быть установлен на 0, когда нет прикрепленного цифрового сервопривода, если на ось посылаются какие-либо битовые %Q команды или %AQ команды данных.** Поддерживаемые типы двигателей FANUC приведены в таблице внизу.

Параметр Motor Type должен быть установлен на 0 для АНАЛОГОВОГО режима или при отсутствии прикрепления двигателя к оси. Значение по умолчанию: 0.

Номера двигателей FANUC используются для определения правильного кода типа двигателя FANUC; они даются в форме A06B-xxxx-уууу, где xxxx представляет собой поле спецификации двигателя. Например: В номере двигателя A06B-0032-B078 на его табличке цифры 0032 показывают модель двигателя  $\beta$ 2/3000. Таблица  $\beta$  Series приводит Код типа двигателя (36), требуемый для поля конфигурации. Поддерживаемые типы двигателей FANUC приведены в таблице внизу. В дальнейших изданиях список поддерживаемых двигателей может быть расширен.

#### $\alpha$ Series FANUC серводвигатели

Код типа двигателя	Модель двигателя	Спецификация двигателя
61	$\alpha$ 1/3000	0371
46	$\alpha$ 2/2000	0372
62	$\alpha$ 2/3000	0373
15	$\alpha$ 3/3000	0123
16	$\alpha$ 6/2000	0127
17	$\alpha$ 6/3000	0128
18	$\alpha$ 12/2000	0142
19	$\alpha$ 12/3000	0143
27	$\alpha$ 22/1500	0146
20	$\alpha$ 22/2000	0147
21	$\alpha$ 22/3000	0148
28	$\alpha$ 30/1200	0151
22	$\alpha$ 30/2000	0152
23	$\alpha$ 30/3000	0153
30	$\alpha$ 40/2000	0157
29	$\alpha$ 40/FAN	0158

#### $\alpha$ L Series FANUC серводвигатели

Код типа двигателя	Модель двигателя	Спецификация двигателя
56	$\alpha$ L3/3000	0561
57	$\alpha$ L6/3000	0562
58	$\alpha$ L9/3000	0564
59	$\alpha$ L25/3000	0571
60	$\alpha$ L50/2000	0572

#### $\alpha$ C Series FANUC серводвигатели

Код типа двигателя	Модель двигателя	Спецификация двигателя
7	$\alpha$ C3/2000	0121

8	$\alpha$ C6/2000	0126
9	$\alpha$ C12/2000	0141
10	$\alpha$ C22/1500	0145

#### $\alpha$ HV Series FANUC серводвигатели

Код типа двигателя	Модель двигателя	Спецификация двигателя
3	$\alpha$ 12HV/3000	0176
4	$\alpha$ 22HV/3000	0177
5	$\alpha$ 30HV/3000	0178

#### $\alpha$ M Series FANUC серводвигатели

Код типа двигателя	Модель двигателя	Спецификация двигателя
24	$\alpha$ M3/3000	0161
25	$\alpha$ M6/3000	0162
26	$\alpha$ M9/3000	0163

#### $\beta$ Series FANUC серводвигатели

Код типа двигателя	Модель двигателя	Спецификация двигателя
13	$\beta$ 0.5/3000	0013
35	$\beta$ 1/3000	0031
36	$\beta$ 2/3000	0032
33	$\beta$ 3/3000	0033
34	$\beta$ 6/2000	0034

#### $\beta$ M Series FANUC серводвигатели

Код типа двигателя	Модель двигателя	Спецификация двигателя
115	$\beta$ M 0.5/5000	0115
116	$\beta$ M 1/5000	0116

#### 6.02 Analog Servo Command (Команда Аналогового Сервопривода).

Параметр Analog Servo Command (Команда Аналогового Сервопривода) определяет, является ли аналоговая команда, выдаваемая модулем DSM300 series командой скорости или момента вращения. Выбор команды момента вращения поддерживается в микропрограмме модуля DSM314 версии 3.0 или более поздней. Значение по умолчанию: Скорость

#### 6.03 Position Error Limit (Предел Ошибки позиционирования).

Предел Ошибки позиционирования (пользовательские единицы). Параметр Position Error Limit (Предел Ошибки позиционирования) является максимальным значением *Ошибки позиционированию (Заданная позиция – Текущая позиция)*, допускаемым, когда модуль DSM314 управляет сервоприводом. Этот предел следует, как правило, устанавливать на 10 – 20% выше, чем наибольшая *Ошибка по положению*, встречающаяся при нормальной работе сервопривода. Значение по умолчанию: 60000.

Формула для диапазона Предел Ошибки позиционирования:

$$256 \times (\text{пользоват. единицы/отсчеты}) \leq \text{Предел Ошибки позиционирования} \leq 60\,000 \times (\text{пользоват. единицы/отсчеты})$$

Если Коэффициент упреждения по скорости подачи не используется, то Предел ошибки по положению может быть установлен на значение, превышающее, примерно, на 20% *Ошибку позиционирования*, требуемую для выдачи команды 400 об./мин. *Ошибка позиционирования* (пользовательские единицы), требуемая для выдачи команды 4000 об./мин. с 0% Коэффициент упреждения по скорости, определяется следующим образом:

Ошибка позиционирования (пользовательские единицы) =	$\frac{\text{Постоянная Времени Контура Позиционирования (мс)} \times \text{Сервоскорость @ 4000 об./мин. (пользовательские единицы/секунды)}}{1000}$
------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

### Пример

Отношение пользовательских единиц к отсчетам равно 2:1; Постоянная Времени Контура Позиционирования составляет 50 мс.

*Этап 1:*

Расчет сервоскорости при 4000 об./мин. =	$\frac{(2 \text{ пользовательские единицы/отсчет}) \times (8192 \text{ отсчетов/оборот}) \times (4000 \text{ оборотов/минута})}{(60 \text{ секунд/минута})}$
	= 1 092 266 пользовательские единицы/секунды

*Этап 2:*

Расчет <i>Ошибки по положению</i> при 4000 об./мин. =	$\frac{(50 \text{ миллисекунд}) \times (1\,092\,266 \text{ пользовательские единицы/секунды})}{1000 \text{ миллисекунд/секунды}}$
	= 54 613 пользовательских единиц

Если для уменьшения ошибки слежения используется Коэффициент упреждения по скорости, то может быть задано меньшее предельное значение; однако, в общем случае, значение предела ошибки должно быть на 10-20% выше максимального ожидаемого значения ошибки слежения.

**Примечание:** Если значение *Предела ошибки позиционирования* превышено более чем на 1000 отсчетов, то выдается ошибка *Out of Sync* и происходит быстрый останов. Модуль DSM314 пытается предотвратить появление ошибки *Out of Sync*, приостанавливая на время работу внутреннего генератора команд всякий раз, когда ошибка по положению превышает *Предел ошибки позиционирования*. Приостановка генератора команд дает возможность обратной связи по положению догнать позицию и сделать ошибку меньше предельно допустимого значения.

Если системе обратной связи не удастся отработать позицию и ошибка по положению продолжает расти, то выполняется условие для появления *Out of Sync (Потеря синхронизации)*. Возможными причинами этого являются следующие:

1. Неправильное подключение обратной связи.

2. Проскальзывание соединительного устройства системы обратной связи.
3. Неисправность сервопривода.
4. Механическое усилие на валу двигателя/датчика положения, превышающее возможности сервопривода по моменту вращения.
5. Заданное значение ускорения или торможения двигателя превышает возможности системы.

**6.04 In Position Zone (В зоне позиционирования).** В зоне позиционирования (Пользовательские единицы). Если ошибка по положению меньше или равна текущему значению *In Position Zone (В зоне позиционирования)*, то бит *In Zone %I* становится в ON (ВКЛ.). Значение по умолчанию: 10.

**6.05 Pos Loop Time Constant (Постоянная Времени Позиционирования) (0.1 мс).** Постоянная Времени Позиционирования (единицы = 0.1 миллисекунды). Параметр задает требуемое значение постоянной времени контура позиционирования сервопривода. Это значение определяет интервал времени, требуемый для того, чтобы скорость сервопривода достигла 63% своего конечного значения при команде *Velocity (Скорость)*, имеющей вид ступенчатого воздействия. Чем меньше это значение, тем быстрее реагирует система. Однако слишком малые значения вызывают неустойчивость системы и возникновение колебаний. Значение по умолчанию: 1000 = 100 мс.

**Примечание:** Чтобы точно отслеживать заданную траекторию скорости, параметр *Pos Loop Time Constant* должен быть от 1/4 до 1/2 МИНИМАЛЬНОГО времени ускорения или торможения системы. Например, если самый быстрый разгон должен происходить за время 100 мс, то *Pos Loop Time Constant* должна быть от 25 до 50 мс. Чтобы поддержать устойчивость системы, можно использовать повышенные значения.

*Для пользователей, знакомых с понятием полосы пропускания сервоустройства, выраженной в рад/с:*

Ширина полосы пропускания (рад/с) = 1000 / Постоянная Времени Контура Позиционирования (мс)

*Для пользователей, знакомых с коэффициентом усиления сервоустройства, выраженным через число прерываний в минуту/мс:*

Коэффициент усиления (число прерываний в минуту/мс) = 60 / Постоянная Времени Контура Позиционирования (мс)

**Таблица 4-11. Коэффициент усиления / Ширина полосы / Постоянная Времени Контура Позиционирования**

Коэффициент передачи (число прерываний в минуту/мил)	Ширина полосы (рад/с)	Постоянная Времени Контура Позиционирования (мс)
0.5	8.5	120
0.75	12.5	80
1.0	16.6	60
1.5	25.1	40
2.0	33.4	30
2.5	41.8	24
3.0	50	20

Для приложений, которые не требуют управления по обратной связи или используют очень грубые системы позиционирования, существует *Open Loop Mode (Режим*

*разомкнутого контура*). Выбор этого режима осуществляется заданием нуля для Постоянной Времени Позиционирования, что указывает на выключение контура позиционирования. Следует отметить, что в Режиме разомкнутого контура единственным способом создать движение является программирование ненулевого Упреждения по скорости подачи. Параметр Ошибка по положению более не используется для создания движения, поскольку Ошибка по положению основана на обратной связи по положению, а в Режиме разомкнутого контура все обратные связи не учитываются.

### Предостережение

**Для Аналоговых Осей Постоянная времени контура позиционирования будет неточной до тех пор, пока не будет правильно установлено значение параметра *Velocity at Max Cmd*.**

#### 6.06 **Velocity at MaxCmd (Скорость при Максимальном значении)**

(пользовательские единицы/секунды). Все аналоговые сервофункции модуля DSM314 зависят от этого значения, которое должно быть правильно установлено, чтобы обеспечить правильное функционирование.

Для **Digital Servo Mode (Режим Цифрового Сервопривода)** поле конфигурации *Velocity at Max Cmd* не используется.

Для **Режима Аналогового Сервопривода в Режиме Скорости** значением поля конфигурации *Velocity at Max Cmd* является Текущая скорость сервопривода (пользовательские единицы/секунда), требуемая при 10 В, выдаваемых на сервопривод с выхода задаваемой аналоговой скорости DSM314. Команда *Force D/A Output (Включить Цифровой/Аналоговый Выход) %AQ* немедленного действия и слово состояния *Actual Velocity (Текущая Скорость) %AI* могут быть использованы при необходимости для задания напряжения с целью проверки правильности сконфигурированного значения.

Для **Режима Аналогового Сервопривода в Режиме Моменты вращения** поле конфигурации *Velocity at Max Cmd* представляет собой максимальную скорость, с которой пользователь предполагает работать на сервоприводе. Это значение определяется свойствами управляемой сервосистемы и приводимой в действие нагрузки.

Только в Цифровом режиме, если пользователь направляет на DSM314 значение скорости, которое превышает возможности сервосистемы, модуль DSM314 *уменьшает* это заданное значение до соответствующего предела максимальной скорости двигателя.

**Обратите внимание, что обратно на DSM314 не выдается сигнал ошибки.**

Более детальную информацию о задании правильного значения см. в приложении D "Пуск и Настройка Цифровых или Аналоговых Сервосистем GE Fanuc".

Значение по умолчанию: 100000.

### Предостережение

**Значение *Velocity at 10V (Скорость при 10 В)* должно быть сконфигурировано правильно, чтобы параметры *Pos Loop Time Constant (Постоянная времени контура позиционирования)* и *Velocity Feedforward (Упреждение по скорости подачи)* были точными.**

#### 6.07 **Velocity Feed Forward (Коэффициент упреждения скорости) (0.01%).**

Величина Коэффициента упреждения скорости подачи (единицы = 0.01%). Доля Заданной Скорости в %, которая прибавляется к выходу задания скорости контура позиционирования DSM314. Возрастающее Коэффициент упреждения скорости увеличивает быстродействие сервопривода и уменьшает ошибку по положению. Оптимальное значение для каждой системы должно быть определено индивидуально.

Для Цифровых сервоприводов значение 95% для **Коэффициента упреждения скорости подачи** является хорошей исходной точкой. Для аналоговых сервоприводов такой хорошей стартовой точкой являются 70%. Характеристики сервосистемы определяют оптимальное значение. Если изменяется значение **Velocity Feed Forward**, то может понадобиться подстройка значения **Pos Err Limit**. Значение по умолчанию: 0.

### Предостережение

Для Аналоговых Осей величина Коэффициент упреждения скорости подачи в % не является точной до тех пор, пока сначала правильно не установлено значение Velocity at MaxCmd (Скорость при максимальной команде).

- 6.08 Acceleration Feed Forward Percentage (Коэффициент упреждения ускорения).** Этот параметр конфигурации не используется в данной микропрограмме модуля DSM314.
- 6.09 Integrator Mode (Режим Интегратора Контура Позиционирования).** Integrator Mode (Режим Интегратора). Режим работы с интегрированием ошибки по положению контура позиционирования. Значение Off (Выкл.) означает, что интегратор не используется. Значение "Continuous" ("Постоянно") означает, что интегратор работает постоянно, в т.ч. и во время движения сервопривода. Servo Null (Нуль Сервопривода) означает, что интегратор работает только тогда, когда бит статуса *Moving (Передвижение)* %I находится в состоянии OFF (ВЫКЛ.). **Режим интегратора** должен быть в обычной ситуации установлен на Off (Выкл.). Режим Continuous разрешается использовать только для таких операций слежения, когда ожидается постоянная или медленно изменяющаяся скорость ведущей оси. Этот параметр не следует использовать для сглаживания возмущений в контуре обратной связи по положению. Никогда не используйте режим Continuous для приложений поточечного позиционирования. Значение по умолчанию: OFF (ВЫКЛ.).
- 6.10 Integrator Time Constant (Постоянная Времени интегрирования Контура Позиционирования).** Постоянная времени интегратора (миллисекунды). Этот параметр определяет постоянную времени интегратора ошибки по положению контура позиционирования. Постоянная времени обозначает интервал времени, требуемый для снижения ошибки по положению на 63%. Например, если постоянная времени интегратора равна 1000 (1 секунда), то *Ошибка по положению* уменьшится до 37% своей первоначальной величины через 1 секунду. Задание нуля выключает интегратор. **Если она используется, то величина постоянной времени интегратора должна быть в 5 – 10 раз больше значения Постоянной времени контура позиционирования, чтобы предупредить возникновение неустойчивости и колебаний.** Значение по умолчанию: 0.
- 6.11 Velocity Loop Gain (Управляющий коэффициент скорости).** Используется для задания коэффициента обратной связи по скорости. Это применимо только к Цифровым Сервоприводам и к Аналоговым Сервоприводам в Режиме Момент вращающего момента GE Fanuc. Этот параметр не используется для Аналоговых сервоприводов в Режиме Скорости.

$$\text{Управляющий коэффициент скорости} = \frac{\text{Инерционность нагрузки (J}_L\text{)}}{\text{Инерционность двигателя (J}_M\text{)}} \times 16$$

Формула может быть использована для выбора начального значения управляющего коэффициента скорости. Допустимый диапазон значений лежит от 0 до 255. Значение 0 используется, если вал двигателя не нагружен.

Значение по умолчанию: 16 (Инерционность нагрузки равна инерционности двигателя).

## Расчет переменных, определяющих предельные значения данных

Предельные значения данных для параметров MaxPosnUu, MaxVelUu и MaxAccUu, приведенных в некоторых из таблиц в данной главе, могут быть рассчитаны с использованием следующих формул:

Таблица 4-12. Расчет переменных, определяющих предельные значения данных

Формулы для расчета переменных, определяющих предельные значения данных		
Граница позиции MaxPosnUu	Граница скорости MaxVelUu	Граница ускорения MaxAccUu
Если отношение uu:cts ("пользовательские единицы/отсчеты") $\geq 1:1$ , MaxPosnUu = 536 870 912 Иначе (если uu:cts < 1:1) MaxPosnUu = 536 870 912 * пользоват. единицы/отсчеты	MaxVelUu = 1 000 000* пользоват. единицы/отсчеты	Если отношение uu:cts ("пользовательские единицы/отсчеты") $\geq 1:1$ , MaxAccUu = 1 073 741 823 Иначе (если uu:cts < 1:1) MaxAccUu = 1 073 741 823* пользоват. единицы/отсчеты

## Дополнительные табличные данные

Хотя дополнительная таблица имеет 16 столбцов для ввода данных параметров настройки осей, микропрограмма модуля DSM314, версия 1.0 позволяет использовать только строки 1 и 2. На рис. внизу показаны данные в ячейках для оси 1 во Входных строки 1 и 2. В версии 3.0 и в более поздних версиях микропрограммы DSM это ограничение устранено.

Settings	SNP Port	CTL Bits	Output Bits	Axis #1	Axis #2	Axis #3	Tuning #1	Tuning #2	Advanced	Power Consumption
Entry	Axis 1 Par #	Axis 1 Data	Axis 2 Par #	Axis 2 Data	Axis 3 Par #	Axis 3 Data	Axis 1	Axis 2	Axis 3	Axis 4
→	1	2	0	0	0	0	0	0	0	0
→	3	10	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0
→	0	0	0	0	0	0	0	0	0	0

Рис. 4-2. Расширенная таблица

## Параметры настройки, поддерживаемые в версии 1.0

Модуль DSM314 версии 1.0 поддерживает только параметры настройки 1 и 3:

**Параметр настройки 1:** Задаёт Разрешение Цифрового датчика положения (только для цифрового сервопривода GE Fanuc). Установочные параметры, отличные от 0, ведут к уменьшению максимальной поддерживаемой двигателем скорости. Обратите внимание, что для установочных параметров 0 и 1 некоторые значения максимальной скорости двигателя ниже максимальной поддерживаемой скорости, показанной в таблице. Диапазон допустимых установок: 0 – 3. На рис. 4.2 вверху параметр настройки 1 установлен на значение 2 для Оси 1.

Таблица 4-13. Значения параметра настройки 1

Значения параметра настройки 1	Отсчеты / обороты	Максим. поддерживаемая скорость двигателя
0	8192	4400 <sup>1,2</sup>
1	16384	3662 <sup>2</sup>
2	32768	1831
3	65536	915

Примечание 1: Параметры по умолчанию

Примечание 2: Некоторые значения максимальной скорости двигателя ниже значения в таблице

**Параметр настройки 3:** Задаёт выход минимальной скорости (милливольт) для аналогового сервопривода. Допустимый диапазон данных от 0 до 1000 мВ. Рекомендуемое значение 5 – 10 мВ или ровно столько, чтобы выполнить перемещение сервопривода на +/- 1 отсчет ошибки по положению. На рис. 4.2 вверху параметр настройки 3 установлен на значение 10 для Оси 1.

### Параметры настройки, поддерживаемые в версии 3.0

Версия 3.0 микропрограммы для модуля DSM314 поддерживает дополнительные параметры настройки, предназначенные для Режимы момента вращения.

**Параметр настройки 6:** задает разрешение датчика положения. Этот параметр используется только в режиме момента вращения. Чтобы работа в режиме момента вращения происходила правильно, это значение должно быть равно числу отсчетов импульсного датчика положения (4 x число линий датчика положения), создаваемых устройством обратной связи двигателя за один его оборот. Пользователь может определить это число из спецификации устройства обратной связи. В качестве дополнительной проверки пользователь может подсоединить устройство обратной связи к DSM и вручную повернуть вал двигателя на один оборот. Значение в данных DSM %AI для текущей позиции должно приблизительно соответствовать значению данного параметра (некоторое различие может быть из-за точности поворота вала вручную на один оборот). Допустимый диапазон составляет 100-32767 отсчетов на оборот. Значение по умолчанию - 4096 отсчетов на оборот.

**Параметр настройки 7:** Задаёт коэффициент передачи пропорционального регулятора скорости. Этот параметр используется только в режиме момента вращения. Коэффициент передачи пропорционального регулятора умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к пропорциональному члену. Правильное задание этого значения определяет, насколько хорошо регулятор скорости будет работать в системе управления. В приложении D описан метод правильной настройки этого параметра. Допустимым диапазоном для члена коэффициента передачи пропорционального регулятора контура скорости является 0 – 32767. Значение по умолчанию: 1500.

**Параметр настройки 8:** Задаёт коэффициент передачи интегрального регулятора скорости. Этот параметр используется только в режиме момента вращения. Коэффициент передачи интегрального регулятора умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к интегральному члену. Правильное задание этого значения определяет, насколько хорошо регулятор скорости будет работать в системе управления. В приложении D описан метод правильной настройки этого параметра. Допустимым диапазоном для члена коэффициента передачи пропорционального регулятора контура скорости является 0 – 32767. Значение по умолчанию: 0.

**Параметр настройки 10:** Задаёт установку фильтра уставок момента вращения. Фильтр уставки момента вращения даёт возможность пользователю активировать фильтр нижних частот на выходе регулятора скорости. Этот фильтр обычно используется для того, чтобы предотвратить появление на машине резонансных колебаний. Допустимая область установочных значений фильтра момента вращения 0 – 3. Значение по умолчанию: 0.

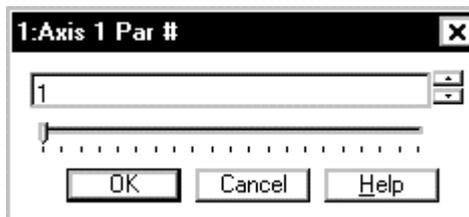
Таблица 4-14. Значения параметра настройки 10

Значения параметра настройки 10	Установочные параметры низкочастотного фильтра заданного момента вращения
0	ВЫКЛ. <sup>1</sup>
1	Узкополосный фильтр (150 Гц на уровне 3 дБ)
2	Среднеполосный фильтр (250 Гц на уровне 3 дБ)
3	Широкополосный фильтр (350 Гц на уровне 3 дБ)

Примечание 1: Установка по умолчанию

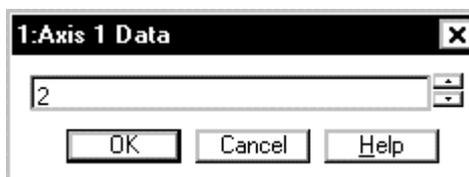
### Ввод номеров параметров настройки и данных в ячейки расширенной таблицы

Для начала работы дважды щелкните требуемую ячейку. Например, если дважды щелкнуть строку ввода 1 в столбце Axis 1 Par # (Ось 1 Параметр #), то появится диалоговое окно 1:Axis 1 Par #, показанное на рисунке внизу. Введите номер требуемого параметра настройки. Помните, что микропрограмма версии 1.0 модуля DSM314 поддерживает параметры настройки только 1 и 3. В примере ниже введено число 1, представляющее параметр настройки 1.



Щелкните кнопку ОК, чтобы установить 1 в строке ввода 1 в ячейке Axis 1 Par #. (Показано на рис. 4-2).

Затем дважды щелкните ячейку справа – строка ввода 1, данные оси 1. Появится следующее диалоговое окно. В данном примере введено число 2. Помните, что для параметра настройки 1 могут быть использованы только значения от 0 до 3 (см. табл. 4-13).



Щелкните кнопку ОК, чтобы установить 2 в строке ввода 1 в ячейке Axis 1 Data. (Показано на рис. 4-2).

### Данные по энергопотреблению

Эта таблица предназначена только для просмотра и показывает мощность, требуемую модулем DSM314.



В данной главе описаны данные, которые передаются между ЦП и модулем Motion Mate DSM314 автоматически на каждом цикле ПЛК без пользовательского программирования. Эти данные подразделяются следующим образом:

- Данные о состоянии входов (передаются от Motion Mate DSM314 на ЦП)
  - Биты статуса: 32 (1 ось), 48 (2 оси), 64 (3 оси), 80 (4 оси) битов данных %I
  - Слова статуса: 24 (1 ось), 44 (2 оси), 64 (3 оси), 84 (4 оси) бита данных % AI
- Данные о выходных командах (передаются от ЦП на Motion Mate DSM314)
  - Дискретные команды: 32 (1 ось), 48 (2 оси), 64 (3 оси), 80 (4 оси) битов данных %Q
  - Команды прямого действия: 3 (1 ось), 6 (2 оси), 9 (3 оси), 12 (4 оси) слов данных %AQ

**Примечание:** Во всей данной главе слова, выделенные *курсивом*, относятся к ссылкам на текущие данные ПЛК станка (%I, %A, %AI, %AQ).

## Раздел 1: Биты статуса %I

Приведенные ниже биты статуса %I пересылаются на каждом цикле автоматически от DSM314 к ЦП. Текущие адреса битов статуса определяются начальным адресом, сконфигурированным для ссылок %I (см. таблицу 4-1 "Таблица установочных параметров"). Смещения битов, приведенные в таблице ниже, являются смещениями для данного начального адреса. Все обозначения в столбце ссылок относятся к настоящей главе.

Таблица 5-1. Биты статуса %I

Смещение бита	Описание	Ось	Ссылка	Смещение бита	Описание	Ось	Ссылка
00	Имеется ошибка модуля	-	1.01	40	Предел Ошибки по Положению	Серво 2	1.12
01	Действует Local Logic	-	1.02	41	Предел по моменту вращения	Серво 2	1.13
02	Получена новая конфигурация	-	1.03	42	Сервоустройство готово / IN4_B (5 В)	Серво 2	1.14
03	<i>Зарезервировано</i>			43	<i>Зарезервировано</i>		
04	STL01 (функция выбрана при конфигурации)	-	1.04	44	Слежение включено	Серво 2	1.15
05	STL02 (функция выбрана при конфигурации)	-	1.04	45	Граница скорости	Серво 2	1.16
06	STL03 (функция выбрана при конфигурации)	-	1.04	46	Действует линейное ускорение / замедление следящего устройства	Серво 2	1.17
07	STL04 (функция выбрана при конфигурации)	-	1.04	47	<i>Зарезервировано</i>		
08	STL05 (функция выбрана при конфигурации)	-	1.04	48	Ось в норме	Серво 3	1.05
09	STL06 (функция выбрана при конфигурации)	-	1.04	49	Позиция достигнута	Серво 3	1.06
10	STL07 (функция выбрана при конфигурации)	-	1.04	50	Привод включен	Серво 3	1.07
11	STL08 (функция выбрана при конфигурации)	-	1.04	51	Программа активна	Серво 3	1.08
12	STL13 (функция выбрана при конфигурации)	-	1.04	52	Передвижение	Серво 3	1.09
13	STL14 (функция выбрана при конфигурации)	-	1.04	53	В зоне	Серво 3	1.10
14	STL15 (функция выбрана при конфигурации)	-	1.04	54	Флаг строба 1 (5 В)	Серво 3	1.11
15	STL16 (функция выбрана при конфигурации)	-	1.04	55	Флаг строба 2 (5 В)	Серво 3	1.11
16	Ось в норме	Серво 1	1.05	56	Предел Ошибки по Положению	Серво 3	1.12
17	Позиция достигнута	Серво 1	1.06	57	<i>Зарезервировано</i>		
18	Привод включен	Серво 1	1.07	58	Сервоустройство готово / IN4_C вход	Серво 3	1.14
19	Программа активна	Серво 1	1.08	59	<i>Зарезервировано</i>		
20	Передвижение	Серво 1	1.09	60	Слежение включено	Серво 3	1.15
21	В зоне	Серво 1	1.10	61	Граница скорости	Серво 3	1.16

Смещение бита	Описание	Ось	Ссылка	Смещение бита	Описание	Ось	Ссылка
22	Флаг строба 1 (5 В)	Серво 1	1.11	62	Действует линейное ускорение / замедление следящего устройства	Серво 3	1.17
23	Флаг строба 2 (5 В)	Серво 1	1.11	63	<i>Зарезервировано</i>	Серво 3	
24	Предел Ошибки по Положению	Серво 1	1.12	64	Ось в норме	Серво 4	1.05
25	Предел по моменту вращения	Серво 1	1.13	65	Позиция достигнута	Серво 4	1.06
26	Сервоустройство готово / IN4_A (5 В)	Серво 1	1.14	66	Привод включен	Серво 4	1.07
27	<i>Зарезервировано</i>			67	Программа активна	Серво 4	1.08
28	Слежение включено	Серво 1	1.15	68	Передвижение	Серво 4	1.09
29	Граница скорости	Серво 1	1.16	69	В зоне	Серво 4	1.10
30	Действует линейное ускорение / замедление следящего устройства	Серво 1	1.17	70	Флаг строба 1 (5 В)	Серво 4	1.11
31	<i>Зарезервировано</i>			71	Флаг строба 2 (5 В)	Серво 4	1.11
32	Ось в норме	Серво 2	1.05	72	Предел Ошибки по Положению	Серво 4	1.12
33	Позиция достигнута	Серво 2	1.06	73	<i>Зарезервировано</i>	Серво 4	
34	Привод включен	Серво 2	1.07	74	Сервоустройство готово / IN4_D (5 В)	Серво 4	1.14
35	Программа активна	Серво 2	1.08	75	<i>Зарезервировано</i>	Серво 4	
36	Передвижение	Серво 2	1.09	76	Слежение включено	Серво 4	1.15
37	В зоне	Серво 2	1.10	77	Граница скорости	Серво 4	1.16
38	Флаг строба 1 (5 В)	Серво 2	1.11	78	Действует линейное ускорение / замедление следящего устройства	Серво 4	1.17
39	Флаг строба 2 (5 В)	Серво 2	1.11	79	<i>Зарезервировано</i>	Серво 4	

**1.01 Имеется ошибка модуля.** Этот бит статуса устанавливается, когда DSM314 обнаруживает какую-либо ошибку. Ошибки, относящиеся к конкретной серво- или вспомогательной оси, идентифицируются в связанном слове *Axis n Error Code (Код Ошибки Оси n) %AI*. Ошибки модуля, не относящиеся к конкретной оси, идентифицируются в слове *Module Status Code (Код Статуса Модуля) %AI*. Подробности см. в разделе 2 "Слова статуса %AI". Бит *Clear Error (Сброс Ошибки) %Q* является единственной командой, которая очищает (сбрасывает) бит статуса *Module Error Present %I* и связанное(ые) слово(а) *Module Status Code* и *Axis n Error Code %AI*. Если условие, вызывающее ошибку продолжает действовать, то бит статуса *Module Error Present %I* не сбрасывается.

**1.02 Действует Local Logic.** Если этот бит статуса ВКЛ., то это значит, что выполняется программа Local Logic.

**1.03 Получена новая конфигурация.** Бит статуса *%I New Configuration Received (Получена новая конфигурация)* устанавливается каждый раз, когда ПЛК посылает на модуль DSM314 команду сброса или новую конфигурацию. Бит *New Configuration Received* должен быть очищен программой ПЛК перед тем, как посылать на DSM314 какие-либо команды немедленного действия *%AQ*, такие как *In Position Zone (Зона прихода в позицию)* или *Position Loop Time Constant (Постоянная времени контура позиционирования)*. После этого статус бита может отслеживаться ПЛК. Если бит установлен, то это значит, что DSM314 был сброшен или переконфигурирован. ПЛК должен очистить бит, а затем направить все необходимые команды *%AQ*. Бит очищается командой прямого действия *%AQ 49h*. Более детальную информацию об

интерфейсе команд немедленного действия см. в разделе 4 "Команды немедленного действия %AQ" данной главы.

**1.04 Конфигурируемые биты статуса %I.** Эти входы показывают состояние конфигурируемых CTL-битов CTL01-CTL08 и CTL13-CTL16. Назначения CTL-битов, осуществляемые по умолчанию, дают информацию об уровне внешних входных устройств, подключенных к сигналам лицевой панели. Все CTL-биты могут быть проверены в ходе выполнения программы *Wait (Ожидание)* и команд *Conditional Jump (Условный переход)*. CTL-биты могут быть также использованы для перехода между функциями включения/выключения линейного ускорения / замедления следящего устройства. Назначения CTL-битов выбираются при конфигурации. Дополнительную информацию см. в главах 4 и 14. Значения битов CTL01-CTL08 и CTL13-CTL16 приведены в табл. 5-2.

**Таблица 5-2. Значения по умолчанию конфигурируемых битов статуса %I**

Обозначение бита	Обозначение сигнала	Использование сигнала	Тип входа	Контакт разъема лицевой панели	Контакт клеммной колодки цифрового сервоустр.	Контакт клеммной колодки аналог. сервоустр. / вспомогат. оси
CTL01	IN9_A	Сервоось 1 (+) (ОТ)перебег	24 В	A-16	6	16
CTL02	IN10_A	Сервоось 1 (-)(ОТ) перебег	24 В	A-34	14	34
CTL03	IN11_A	Сервоось 1 - переключатель исходной позиции	24 В	A-17	7	17
CTL04	IN1_A	Сервоось 1 – уровень строба 1	5 В	A-1,19	1,9	9
CTL05	IN9_B	Сервоось 2 (+)(ОТ) перебег	24 В	B-16	6	16
CTL06	IN10_B	Сервоось 2 (-)(ОТ) перебег	24 В	B-34	14	34
CTL07	IN11_B	Сервоось 2 - переключатель исходной позиции	24 В	B-17	7	17
CTL08	IN1_B	Сервоось 2 – уровень строба 1	5 В	B-1,19	1,9	9
CTL13	IN9_C	Сервоось 3 (+) (ОТ)перебег	24 В	C-16	-	16
CTL14	IN10_C	Сервоось 3 (-) (ОТ)перебег	24 В	C-34	-	34
CTL15	IN11_C	Сервоось 3 - переключатель исходной позиции	24 В	C-17	-	17
CTL16	IN5_C	Сервоось 3 – уровень строба 1	5 В	C-9	-	9

**1.05 Ось в норме.** Бит статуса *Axis OK (Ось в норме)* находится в состоянии ON (ВКЛ.), когда модуль DSM314 готов к приему команд и управлению сервоприводом. При возникновении Сбойной ситуации, которая останавливает сервопривод, переводит бит *Axis OK* в состояние OFF (ВЫКЛ.). Если бит *Axis OK* находится в состоянии OFF, то ось не может принять никакую команду, кроме бита *Clear Error %Q*.

- 1.06 Позиция достигнута.** Для сервооси бит статуса *Position Valid* (*Позиция достигнута*) указывает на то, что команда *Set Position* (*Установить позицию*) или успешное завершение цикла *Find Home* (*Поиск Исходной Позиции*) инициализировали значение позиции в слове статуса *Actual Position* (*Текущая позиция*) %AI. Для сервооси бит *Position Valid* должен быть в состоянии ON, чтобы можно было выполнить программу движения.
- Для вспомогательной оси бит статуса *Position Valid* (*Позиция достигнута*) указывает на то, что команда *Aux Encoder Set Position* (*Установить позицию вспомогательного датчика положения*) или успешное завершение цикла *Find Home* (*Поиск Исходной Позиции*) инициализировали значение позиции в слове статуса *Actual Position* (*Текущая позиция*) %AI. Для вспомогательной оси **не** требуется нахождение бита *Position Valid* в состоянии ON, чтобы можно было выполнить программу движения.
- Если модуль DSM314 сконфигурирован с использованием абсолютной цифровой датчика положения обратной связи (сервопривод GE Fanuc  $\alpha$  или  $\beta$  Series с дополнительным аккумулятором датчика положения), то *Position Valid* устанавливается автоматически каждый раз, когда цифровой датчик положения сообщает о достижении абсолютной позиции. См. Приложение С подробности по работе с использованием абсолютного режима цифровой датчик положения.
- 1.07 Привод включен.** Бит статуса *Drive Enabled* (*Привод включен*) показывает состояние бита *Enable Drive* %Q и выхода твердотельного реле, питание на которое подается от DSM314. Состояние ON (ВКЛ.) бита *Drive Enabled* %I соответствует состоянию ЗАМКНУТО на выходе реле и состоянию ON (ВКЛ.) соответствующего индикатора EN на лицевой панели. В цифровом режиме твердотельное реле обеспечивает сигнал MCON на цифровой сервопривод GE Fanuc через кабель сервокоманд. Бит *Drive Enabled* очищается при включении питания или при сбое, при котором происходит остановка сервопривода.
- 1.08 Программа активна.** Бит статуса *Program Active* (*Программа активна*) для каждой оси показывает, что на данной оси выполняется программа Motion (*Движение*) (1-10) или команда *Move* %AQ (27h). Выполнение многоосной программы устанавливает биты *Program Active* для обеих осей 1 и 2.
- 1.09 Передвижение.** Бит статуса *Moving* (*Передвижение*) находится в состоянии ON (ВКЛ.), когда Заданная скорость не равна нулю, иначе – в состоянии OFF (ВЫКЛ.). Все команды *Move*, *Jog* и *Move at Velocity* будут вызывать установку бита *Moving* (*Передвижение*) в состоянии ON (ВКЛ.). Команда *Force Servo Velocity* (*Включить скорость сервопривода*) %AQ и линейное ускорение следящего устройства не вызывают установки бита *Moving*.
- В режиме слежения бит *Moving* находится в состоянии ON в ситуациях, указанных выше; на него не влияет состояние "включено" или "выключено" ведущего входа следящего устройства. Если действует линейное ускорение / замедление следящего устройства, то в состоянии ON (ВКЛ.) находится специальный бит %I, *Follower Ramp Active* (*Действует линейное ускорение / замедление следящего устройства*). Дополнительную информацию по линейному изменению скорости следящего устройства см. в главе 8 "Движение следящего устройства".
- 1.10 В зоне.** Действие бита *In Zone* (*В зоне*) зависит только от значения *Position Error* (*Ошибка по положению*) и не связано с состоянием бита *Moving*. Бит *In Zone* устанавливается в состояние ON, когда Ошибка по положению оказывается меньше или равна сконфигурированному значению *In Position Zone* (*Зона позиционирования*). Бит *In Zone* (ON) может быть использован в комбинации с битом *Moving* (OFF), чтобы определить момент, когда ось пришла в точку назначения.

Таблица 5-3. Действие бита *In Zone*

Действует генератор команд (бит <i>Moving %I</i> в состоянии ON)	Ошибка по положению $\leq$ Зона позиционирования	Бит <i>In Zone</i>	Ось в точке назначения
Нет	Нет	OFF	Нет
Нет	Да	ON	Да
Да	Нет	OFF	Нет
Да	Да	ON	Нет

- 1.11 Флаг строба 1, флаг строба 2.** Биты статуса *Strobe 1 Flag* (Флаг строба 1) и *Strobe 2 Flag* (Флаг строба 2) показывают, что произошел переход от OFF к ON на соответствующем Входе строба на лицевой панели. Когда это происходит, то позиция оси захватывается и передается в слово статуса *Strobe n Position* (Позиция строба n) %AI, где "n" означает Ось 1 – Ось 4. Бит *Strobe n Flag* (Флаг строба n) %I очищается соответствующим битом *Reset Strobe n* (Сброс строба n) %Q. **После того как бит *Reset Strobe n %Q* был установлен в состояние ON, требуется не более 2 циклов ПЛК, чтобы бит *Strobe n Flag %I* был очищен в ПЛК.** После очистки бита *Strobe n Flag* могут быть захвачены новые данные другим Импульсом строба. Разрешение захвата позиции составляет +/- 2 отсчета с дополнительным отклонением 10 микросекунд для задержки на входном фильтре строба.

**Примечание:** Биты *Strobe n Flag* не показывают логический уровень входа лицевой панели; они только показывают, что на входе произошел переход от OFF к ON.

- 1.12 Предел Ошибки по Положению.** Установка бита статуса *Position Error Limit* (Предел Ошибки по Положению) происходит в том случае, если абсолютное значение ошибки по положению превышает сконфигурированное значение *Position Error Limit* (Предел Ошибки по Положению). Если установлен бит статуса *Position Error Limit*, то значения *Commanded Velocity* (Заданная скорость) и *Commanded Position* (Заданная позиция) перестают изменяться ("замораживаются"), чтобы дать возможность оси "быть захваченной" в Заданной позиции.
- 1.13 Предел по моменту вращения.** Бит статуса *Torque Limit* (Предел по моменту вращения) устанавливается в том случае, если заданный момент вращения превышает установку предела момента вращения для сконфигурированного типа двигателя.
- 1.14 Сервоустройство готово.** Этот бит статуса устанавливается, когда сигнал лицевой панели IN4 соответствующего разъема (A, B, C или D) находится в состоянии ON (активный низкий уровень: ON = 0 В, OFF = +5 В). Для каждой сервооси этот вход показывает состояние *Servo Ready* (Сервоустройство готово) сервоусилителя.
- 1.15 Слежение включено.** Этот бит статуса показывает, когда слежение включено для оси. Бит *Enable Follower* (Включить слежение) % Q и дополнительный триггерный вход лицевой панели STL01-CTL32 включают функцию слежения. Если управление изменением скорости следящего устройства действует, когда бит *Follower Enabled* (Слежение включено) устанавливается в состояние ON, то ось будет ускоряться до заданной скорости ведущего устройства, а когда оно выключается, то ось будет замедляться до нулевой заданной скорости ведущего устройства. Как ускорение, так и замедление в ходе процесса изменения скорости будут использовать сконфигурированный закон изменения скорости следящего устройства.
- 1.16 Граница скорости.** Бит статуса *Velocity Limit* (Граница скорости) устанавливается в том случае, если скорость, задаваемая какой-либо командой оси (внутренний генератор траектории или внутренний / внешний источник следящего устройства), превышает сконфигурированную границу скорости. Следовательно, бит *Velocity Limit* индицирует, что ось более не привязана к своей команде позиционирования. Если Слежение

включено, то при установленном бите *Velocity Limit* в соответствующую переменную кода ошибки выдается код ошибки.

Исключение имеет место, когда сконфигурировано однонаправленное движение заданием *Command Direction* (Заданное направление) *Positive Only* (Только положительное) или *Negative Only* (Только отрицательное). Задание *Positive Only* (Только положительное) означает, что для движения в отрицательном направлении границей скорости является ноль. Задание *Negative Only* (Только отрицательное) означает, что для движения в положительном направлении границей скорости является ноль. Для границы, установленной на ноль, ошибка не выдается. Например, если Заданное направление установлено на Только отрицательное, а выдана команда + Counts (Отсчеты), то происходит установка бита статуса *Velocity Limit* (*Граница скорости*), но код ошибки статуса не выдается.

#### **1.17 Действует линейное ускорение / замедление следящего устройства.**

Если слежение включено, то бит статуса *Follower Ramp Active* (*Действует линейное ускорение / замедление следящего устройства*) находится в состоянии ON во время начального ускорения и компенсации расстояния. Если слежение выключено, то бит статуса *Follower Ramp Active* находится в состоянии ON до тех пор, пока идет приращение позиции в соответствии с *Follower Disable Action* (*Действие операции Выключения Следящего Устройства*) и следящее устройство замедляется до нулевой скорости.

## Раздел 2: %AI слова статуса

Приведенные ниже слова статуса %AI передаются на каждом цикле автоматически от модуля DSM314 к ЦП. Все слова статуса %AI конфигурируются при помощи ПО конфигурации на длину 22, 44, 64 или 84. Текущие адреса слов статуса зависят от начального адреса, сконфигурированного для ссылок %AI. См. таблицу 4-1 "Таблица установочных параметров". Номера слов, приведенные в таблице ниже, являются смещениями для данного начального адреса. Все обозначения в столбце ссылок относятся к настоящей главе. Все данные %AI за исключением *Actual Velocity (Текущая скорость)* обновляются внутри DSM314 с частотой опроса контура позиции (2 мс для цифрового сервопривода, 0,5 мс или 1,0 мс для некоторых аналоговых сервоконфигураций). *Текущая скорость* обновляется через каждые 128 миллисекунд.

Таблица 5-4. %AI слова статуса

Смещ. слова	Описание	Ось	Ссылка	Смещ. слова	Описание	Ось	Ссылка
00	Код статуса модуля	-	2.01				
01-03	<i>Зарезервировано</i>						
04	Код ошибки оси 1	Серво 1	2.02	44	Код ошибки оси 3	Серво 3	2.02
05	Номер блока команды	Серво 1	2.03	45	Номер блока команды	Серво 3	2.03
06-07	Заданная Позиция	Серво 1	2.04	46-47	Заданная Позиция	Серво 3	2.04
08-09	Текущая Позиция	Серво 1	2.05	48-49	Текущая Позиция	Серво 3	2.05
10-11	Позиция строба 1	Серво 1	2.06	50-51	Позиция строба 1	Серво 3	2.06
12-13	Позиция строба 2	Серво 1	2.06	52-53	Позиция строба 2	Серво 3	2.06
14-15	Ошибка по Положению	Серво 1	2.07	54-55	Ошибка по Положению	Серво 3	2.07
16-17	Заданная Скорость	Серво 1	2.08	56-57	Заданная Скорость	Серво 3	2.08
18-19	Текущая Скорость	Серво 1	2.09	58-59	Текущая Скорость	Серво 3	2.09
20-21	Данные 1, выбранные пользователем	Серво 1	2.10	60-61	Данные 1, выбранные пользователем	Серво 3	2.10
22-23	Данные 2, выбранные пользователем	Серво 1	2.11	62-63	Данные 2, выбранные пользователем	Серво 3	2.11
24	Код ошибки оси 2	Серво 2	2.02	64	Код ошибки оси 4	Серво 4	2.02
25	Номер заданного блока	Серво 2	2.03	65	Номер блока команды	Серво 4	2.03
26-27	Заданная Позиция	Серво 2	2.04	66-67	Заданная Позиция	Серво 4	2.04
28-29	Текущая Позиция	Серво 2	2.05	68-69	Текущая Позиция	Серво 4	2.05
30-31	Позиция строба 1	Серво 2	2.06	70-71	Позиция строба 1	Серво 4	2.06
32-33	Позиция строба 2	Серво 2	2.06	72-73	Позиция строба 2	Серво 4	2.06
34-35	Ошибка по Положению	Серво 2	2.07	74-75	Ошибка по Положению	Серво 4	2.07
36-37	Заданная Скорость	Серво 2	2.08	76-77	Заданная Скорость	Серво 4	2.08
38-39	Текущая Скорость	Серво 2	2.09	78-79	Текущая Скорость	Серво 4	2.09
40-41	Данные 1, выбранные пользователем	Серво 2	2.10	80-81	Данные 1, выбранные пользователем	Серво 4	2.10
42-43	Данные 2, выбранные пользователем	Серво 2	2.11	82-83	Данные 2, выбранные пользователем	Серво 4	2.11

- 2.01 Код статуса модуля.** *Module Status Code (Код статуса модуля)* показывает рабочее состояние модуля DSM314. Если установлен флаг *Module Error Present (Имеется ошибка модуля)* %I и эта ошибка не относится к какой-то конкретной оси, то в Код статуса модуля передается номер кода ошибки, который описывает причину ошибки. Новый *Код статуса модуля* **не замещает** предыдущий *Код статуса модуля*, если только новый код не имеет приоритета Fast Stop (Быстрый останов) или System Error (Системная ошибка).

Слово *Код статуса модуля* используется также для передачи *Ошибок статуса системы*. Они имеют формат Dxxx, Exxx и Fxxx. Более детальную информацию по кодам *Ошибок статуса системы* см. в приложении А.

*Список кодов ошибок модуля Motion Mate DSM314 см. в приложении А.*

- 2.02 Код ошибки Оси 1 – Оси 4.** *Servo Axis n Error Code (Код ошибки сервооси n)*, где n = Ось 1 – Ось 4, показывает текущий рабочий статус каждой оси. Если установлен флаг *Module Error Present (Имеется ошибка модуля)* %I и эта ошибка относится к какой-то конкретной оси, то передается номер кода ошибки, который описывает причину ошибки. Новый *Axis Error Code (Код ошибки оси)* замещает предыдущий *Код ошибки оси*, если он имеет такой же или более высокий приоритет (Предупреждение, Нормальный останов, Быстрый останов) по сравнению с предыдущим *Кодом ошибки оси*.

*Список кодов ошибок модуля Motion Mate DSM314 см. в приложении А.*

- 2.03 Номер блока команды.** *Command Block Number (Номер блока команды)* показывает номер блока той команды, которая выполняется в данный момент в активной программе или подпрограмме. Он изменяется при запуске каждого нового блока при выполнении команд и, таким образом, идентифицирует текущее рабочее положение внутри программы. Номера блоков отображаются только в том случае, если программа движения использует их. Кроме того, номер самого последнего использованного блока отображается на экране до тех пор, пока не будет заменен новым значением. *Номер блока команды* устанавливается на ноль при вкл./выкл. питания или при перезапуске.

- 2.04 Заданная Позиция.** *Commanded Position (Заданная позиция)* (пользовательские единицы) – это позиция, где ось должна быть через некоторое время. Для сервооси разница между *Заданной позицией* и *Текущей позицией* является *Ошибкой по положению*, которая вырабатывает команду скорости, чтобы приводить в движение ось. Скорость, с которой изменяется *Заданная позиция*, определяет скорость движения оси.

Если *Заданная позиция* перемещается за пределы отсчета, то она переходит к другой границе и продолжается в направлении движения оси.

- 2.05 Текущая Позиция.** *Actual Position (Текущая позиция)* (пользовательские единицы) – это значение, обрабатываемое модулем DSM314, для предоставления физического положения оси. Оно устанавливается на начальное значение командой немедленного действия *Set Position (Установить позицию)* %AQ или на Исходную позицию в цикле *Find Home (Поиск Исходной Позиции)*. Если используются цифровые абсолютные датчики положения, то *Текущая позиция* устанавливается автоматически каждый раз, когда датчик положения сообщает о достижении позиции. Движение устройства обратной связи оси непрерывно обновляет *Текущую позицию* оси.

Если *Текущая позиция* перемещается за пределы отсчета, то она переходит к другой границе и продолжается в направлении движения оси.

- 2.06 Позиция строба 1, 2.** *Strobe 1 Position (Позиция строба 1)* и *Strobe 2 Position (Позиция строба 2)* (пользовательские единицы) содержит текущую позицию оси, если имеет место сигнал Strobe 1 Input (Вход строба 1) или Strobe 2 Input (Вход строба 2). Если появляется Импульс строба, то устанавливается бит Флага строба 1 или Флага строба 2, сообщающая на ПЛК, что в соответствующем слове статуса *Strobe 1 Position* или

*Strobe 2 Position* имеются новые данные строба. ПЛК должен установить надлежащий бит флага *Reset Strobe 1 (Сброс строба 1)* или *Reset Strobe 2 (Сброс строба 2)* %Q, чтобы очистить соответствующий бит Флага строба 1, 2 %I.

*Strobe 1, 2 Position* (Позиция строба 1, 2) будет сохраняться и не будет перекрываться дополнительными Импульсами строба до тех пор, пока не будет очищен бит Флага строба 1, 2 %I. Если бит *Reset Strobe Flag* (Флаг Сброса строба) %Q оставлен в состоянии ON (таким образом, бит Флага строба 1, 2 %I удерживается в очищенном состоянии), то каждый появляющийся Импульс строба будет вызывать захват позиции оси в *Strobe 1, 2 Position* (Позиция строба 1, 2).

Значения текущей позиции *Strobe 1, 2 Position* помещаются также в регистры параметров данных для использования с командами программ движения. Регистр параметров данных имеет следующие назначения:

	Сервоось 1	Сервоось 2	Сервоось 3	Сервоось 4
Позиция строба 1	P224	P232	P240	P248
Позиция строба 2	P225	P233	P241	P249

Это свойство позволяет импульсу строба переключать Conditional JUMP (Условный ПЕРЕХОД) в программном блоке, используя *Strobe 1 Position* или *Strobe 2 Position* в качестве назначения команды CMOVE или PMOVE.

**Информацию по задержке строба и по длительности обработки см. в гл. 1 "Краткое описание изделия, стробы позиции DSM314".**

- 2.07 Ошибка по Положению.** *Position Error* (Ошибка по положению) (пользовательские единицы) является разность между Заданной и Текущей позициями. В контуре управления сервоустройства Ошибка по положению умножается на постоянный коэффициент, чтобы обеспечить команду скорости сервопривода.
- 2.08 Заданная Скорость.** *Commanded Velocity* (Заданная скорость) (пользовательские единицы в секунду) представляет собой значение, создаваемое генератором команд оси в модуле DSM314. Заданная скорость показывает текущую команду скорости, которая вызывает движение оси. В начале движения скорость возрастает с заданным ускорением; после достижения запрограммированной скорости она стабилизируется на заданном уровне.
- В режиме Слежения *Заданная скорость* представляет только выход генератора команд оси. Вход ведущей оси следящего устройства или контроллер изменения скорости следящего устройства не влияют на *Заданную скорость*.
- 2.09 Текущая Скорость.** *Actual Velocity* (Текущая скорость) (пользовательские единицы в секунду) представляет собой скорость оси, получаемую по данным устройства обратной связи; она обновляется модулем DSM314 один раз в 128 миллисекунд.
- 2.10 Данные 1, выбранные пользователем.** Имеется по одному из этих слов для каждой из четырех осей. Информация, передаваемая в *User Selected Data 1* (Данные 1, выбранные пользователем) определяется при конфигурации модуля (см. гл. 4) или командой *Select Return Data 1* (Выбрать возвращаемые данные 1) %AQ (см. раздел 4 "Команды немедленного действия %AQ" в данной главе).
- 2.11 Данные 2, выбранные пользователем.** Имеется по одному из этих слов для каждой из четырех осей. Информация, передаваемая в *User Selected Data 2* (Данные 2, выбранные пользователем) определяется при конфигурации модуля (см. гл. 4) или командой *Select Return Data 1* (Выбрать возвращаемые данные 1) %AQ (см. раздел 4 "Команды немедленного действия %AQ" в данной главе). Дополнительную информацию см. в разделе 4 "Команды немедленного действия %AQ".

### Раздел 3: Дискретные команды %Q

Приведенные далее выходы %Q представляют Дискретные команды, посылаемые автоматически на модуль DSM314 от ЦП на каждом цикле ПЛК. Команда выполняется переключением в состояние ON (ВКЛ.) соответствующего Бита выхода. Текущие адреса битов дискретной команды зависят от начального адреса, сконфигурированного для ссылок %Q. См. таблицу 4-1 "Таблица установочных параметров". Смещения битов, приведенные в таблице ниже, являются смещениями для данного начального адреса. Номера в столбце "Ссылки" относятся к разделам данной главы.

Таблица 5-5. Дискретные команды %Q

Смещение бита	Описание	Ось	Ссылка	Смещение бита	Описание	Ось	Ссылка
00	Сбросить ошибку	-	3.01	40	OUT1_В / Конфигурируемый источник для STL-битов	Серво 2	3.12
01	Включить Local Logic	-	3.02	41	OUT3_В / Конфигурируемый источник для STL-битов	Серво 2	3.13
02	Выполнить программу Движения 1	-	3.03	42	Зарезервировано		
03	Выполнить программу Движения 2	-	3.03	43	Зарезервировано		
04	Выполнить программу Движения 3	-	3.03	44	Включить слежение	Серво 2	3.14
05	Выполнить программу Движения 4	-	3.03	45	Выбрать ведущий источник следящего устройства	Серво 2	3.15
06	Выполнить программу Движения 5	-	3.03	46	Зарезервировано		
07	Выполнить программу Движения 6	-	3.03	47	Зарезервировано		
08	Выполнить программу Движения 7	-	3.03	48	Прекратить все движения	Серво 3	3.05
09	Выполнить программу Движения 8	-	3.03	49	Остановить подачу (программа Пауза)	Серво 3	3.06
10	Выполнить программу Движения 9	-	3.03	50	Включить привод / MCON	Серво 3	3.07
11	Выполнить программу Движения 10	-	3.03	51	Поиск Исходной Позиции	Серво 3	3.08
12	Конфигурируемый источник STL-битов	-	3.04	52	Плюс Толчка	Серво 3	3.09
13	Конфигурируемый источник STL-битов	-	3.04	53	Минус Толчка	Серво 3	3.10
14	Конфигурируемый источник STL-битов	-	3.04	54	Сбросить строб 1	Серво 3	3.11
15	Конфигурируемый источник STL-битов	-	3.04	55	Сбросить строб 2	Серво 3	3.11
16	Прекратить все движения	Серво 1	3.05	56	OUT1_С / Конфигурируемый источник для STL-битов	Серво 3	3.12
17	Остановить подачу (программа Пауза)	Серво 1	3.06	57	OUT3_С / Конфигурируемый источник для STL-битов	Серво 3	3.13
18	Включить привод / MCON	Серво 1	3.07	58	Зарезервировано		
19	Поиск Исходной Позиции	Серво 1	3.08	59	Зарезервировано		
20	Плюс Толчка	Серво 1	3.09	60	Включить слежение	Серво 3	3.14
21	Минус Толчка	Серво 1	3.10	61	Выбрать ведущий источник следящего устройства	Серво 3	3.15
22	Сбросить строб 1	Серво 1	3.11	62	Зарезервировано		
23	Сбросить строб 2	Серво 1	3.11	63	Зарезервировано		
24	OUT1_А / Конфигурируемый	Серво 1	3.12	64	Прекратить все движения	Серво 4	3.05

	источник для STL-битов						
25	OUT3_A / Конфигурируемый источник для STL-битов	Серво 1	3.13	65	Остановить подачу (программа Пауза)	Серво 4	3.06
26	Зарезервировано			66	Включить привод / MCON	Серво 4	3.07
27	Зарезервировано			67	Поиск Исходной Позиции	Серво 4	3.08
28	Включить слежение	Серво 1	3.14	68	Плюс Толчка	Серво 4	3.09
29	Выбрать ведущий источник следящего устройства	Серво 1	3.15	69	Минус Толчка	Серво 4	3.10
30	Зарезервировано			70	Сбросить строб 1	Серво 4	3.11
31	Зарезервировано			71	Сбросить строб 2	Серво 4	3.11
32	Прекратить все движения	Серво 2	3.05	72	OUT1_V / Конфигурируемый источник для STL-битов	Серво 4	3.12
33	Остановить подачу (программа Пауза)	Серво 2	3.06	73	OUT3_V / Конфигурируемый источник для STL-битов	Серво 4	3.13
34	Включить привод / MCON	Серво 2	3.07	74	Зарезервировано	Серво 4	
35	Поиск Исходной Позиции	Серво 2	3.08	75	Зарезервировано	Серво 4	
36	Плюс Толчка	Серво 2	3.09	76	Включить слежение	Серво 4	3.14
37	Минус Толчка	Серво 2	3.10	77	Выбрать ведущий источник следящего устройства	Серво 4	3.15
38	Сбросить строб 1	Серво 2	3.11	78	Зарезервировано	Серво 4	
39	Сбросить строб 2	Серво 2	3.11	79	Зарезервировано	Серво 4	

**3.01 Сбросить ошибку.** Если возникает сбойная ситуация, то эта команда используется для очистки бита статуса *Module Error Present (Имеется ошибка модуля) %I*, а также связанных *%AI* слов статуса *кода статуса модуля* и *кода ошибки осей 1 – 4*. Условия ошибки, которые продолжают действовать (такие как ошибка концевого переключателя *End of Travel (Конец передвижения)*), не сбрасываются; их сброс выполняется другими корректирующими действиями. Если бит *Clear Error* находится в состоянии ON, то может быть использована команда *Jog (Толчок)* для увода от разомкнутого концевого переключателя перебега.

**3.02 Включить Local Logic.** Эта команда позволяет выполнить текущую программу Local Logic в DSM. См. в гл. 4 информацию по сконфигурированному имени программы Local Logic.

**3.03 Выполнить программу Движения 1 - 10.** Эти команды используются для выбора сохраненных программ движения для немедленного выполнения. Каждая команда использует одно короткое действие; поэтому командный бит должен переходить от OFF к ON каждый раз, когда программа должна быть исполнена. Программы можно останавливать на время командой *Feed Hold (Остановить подачу)*.

Когда начинается выполнение программы, *Rate Override (Подстановка скорости)* всегда устанавливается на 100%. Команда *Rate Override (Подстановка скорости) %AQ* может быть послана в том же цикле, что и бит *Execute Motion Program n (Выполнить программу движения n) %Q*, и будет действовать при запуске программы.

Одновременно может выполняться только одна программа Движения на ось. Бит статуса *Program Active (Программа активна) %I* должен быть в состоянии OFF (ВЫКЛ.); иначе не будет разрешен запуск программы Движение на выполнение. Многоосная программа Движение использует как ось 1, так и ось 2, поэтому оба бита *Program Active* должны быть в состоянии OFF, чтобы могла стартовать многоосная программа Движение.

**3.04 Конфигурируемые источники CTL-битов.** Смещения 12-15 битов %Q конфигурируются как источники для CTL-битов CTL01-CTL24. Дополнительную информацию см. в главе 4. Конфигурация по умолчанию:

%Q бит смещение 12: CTL09

%Q бит смещение 13: CTL10

%Q бит смещение 14: CTL11

%Q бит смещение 15: CTL12

**3.05 Прекратить все движения.** Эта команда вызывает прекращение любого совершаемого движения при текущем значении *Jog Acceleration* (*Изменение скорости при Толчке*) и при сконфигурированном режиме *Jog Acceleration* (*Изменение скорости при Толчке*). Поэтому важно использовать такое *Jog Acceleration*, которое обеспечит замедление на надлежащем отрезке пути. Любая ожидающая запрограммированная команда или команда немедленного действия удаляется и поэтому не может начать работать. Ситуация прекращения действует до тех пор, пока команда прекращения включена. Если движение происходило в тот момент, когда была получена данная команда, то бит статуса *Moving* будет оставаться установленным до тех пор, пока заданная скорость не достигнет нуля.

**3.06 Остановить подачу (включение команды).** Эта команда вызывает прекращение любого совершающегося движения с таким изменением скорости, которое заложено в текущую программу. Команда *Feed Hold* (*Остановить подачу*) не прекращает движение, задаваемое ведущим источником в режиме включения слежения. Когда движение прекращается, бит статуса *Moving* (*Передвижение*) очищается, а бит статуса *In Zone* (*В зоне*) устанавливается, если выполняется условие *In Zone*. В ситуации *Feed Hold* команды *Jog* (*Толчок*) разрешены. После срабатывания команды *Feed Hold* (*Остановить подачу*) программное движение прекращается, даже если переходы командных битов были переведены в состояние OFF до того, как движение прекратилось.

**Остановить подачу (выключение команды).** Эта команда вызывает продолжение работы любых программ движения, прерванных командой *Feed Hold*, при программных значениях ускорения и скорости. При этом будут выполнены дополнительные программные движения, и будет продолжено нормальное выполнение программы. Команда *Feed Hold OFF* (*Выключить остановку подачи*) действует аналогично команде Выполнить программу за исключением того, что ПО для выработки траектории движения использует в программе только остающуюся часть пути.

Если в то время как действовала команда *Feed Hold* (*Остановить подачу*) был выполнен толчок, то прерванная команда *Move* будет восстановлена с того места, где ось была оставлена после *Jog* (*Толчок*). Эта команда *Move* выйдет на правильную программную скорость и продолжит с исходно запрограммированной позиции так, как будто не было смещения под действием Толчка.

**3.07 Включить привод / MCON.** Если биты статуса *Module Error Present* (*Имеется ошибка модуля*) и *Drive Enabled* (*Привод включен*) %I очищены, то эта команда *Enable Drive / MCON* (*Включить привод / MCON*) вызывает замыкание контактов реле Включения привода и установку бита *Drive Enabled* %I. Если бит *Drive Enabled* %I установлен, то оказываются включенными функции генерации траектории и управления позицией; можно выполнять управление сервоприводом. Сигнал направляется (*MCON*) к цифровому сервоустройству, включая привод. Чтобы сделать возможным нормальное движение сервопривода, *Enable Drive* должен быть в состоянии ON (за исключением использования команд *Jog*). Если используется команда немедленного действия *Force Analog Output* (*Включить аналоговый выход*) (см. раздел 4.06 "Включить аналоговый выход"), то должен быть включен соответствующий сигнал *Enable Drive*, чтобы эта команда могла создать сигнал на аналоговом выходе.

- 3.08 Поиск Исходной позиции.** Команда Find Home (Поиск Исходной позиции) вызывает установку модуля DSM314 в *Home Position (Исходная позиция)*. Вход переключателя Исходной позиции на разьеме В/В ориентировочно показывает положение Исходной позиции; следующий маркер датчик положения показывает ее точное положение. Если конфигурация оси *Home Mode (Режим исходной позиции)* установлена на MOVE+ или MOVE-, то вход переключателя Исходной позиции игнорируется. Для сервооси сконфигурированное *Home Offset (Смещение исходной позиции)* определяет *Home Position (Исходную позицию)* как отрезок смещения от маркера исходной позиции. Индикация бита *Position Valid (Позиция достигнута) %I* устанавливается по завершении Home Cycle (Цикл исходной позиции). *Дополнительную информацию по Циклу исходной позиции см. в гл. 6. Информацию по абсолютному датчику положения см. в приложении С.*
- 3.09 Плюс Толчка.** Когда этот командный бит *Jog Plus (Плюс Толчка)* установлен в ON, то ось двигается в положительном направлении при сконфигурированных уставках *Jog Acceleration (Ускорение Толчка)* и *Jog Velocity (Скорость Толчка)*. Установка *Jog Plus* в положение OFF вызывает торможение и остановку оси. Если *Jog Plus* выключается на короткое время, хотя бы на один цикл ПЛК, то ось замедляется до остановки, а затем ускоряется и продолжает движение толчка. Ось будет продолжаться двигаться до тех пор, пока действует команда *Jog Plus* и пока не достигнута сконфигурированная программная граница *Positive End Of Travel (Положительный конец передвижения)* или переключатель Положительного перебега. Входы переключателя перебега могут быть выключены при помощи конфигурационного параметра *OT Limit*. Команда *Jog Plus* может быть использована для отхода толчком от переключателя Отрицательного перебега, если бит *Clear Error (Сбросить ошибку) %Q* также сохраняется во включенном состоянии. *Дополнительную информацию по Толчковому передвижению при помощи модуля DSM314 см. в гл. 6 "Непрограммируемое движение".*
- 3.10 Минус Толчка.** Когда этот командный бит *Jog Minus (Минус Толчка)* установлен в ON, то ось двигается в отрицательном направлении при сконфигурированных уставках *Jog Acceleration (Ускорение Толчка)* и *Jog Velocity (Скорость Толчка)*. Установка *Jog Minus* в положение OFF вызывает торможение и остановку оси. Если *Jog Minus* выключается на короткое время, хотя бы на один цикл ПЛК, то ось замедляется до остановки, а затем ускоряется и продолжает движение толчка. Ось будет продолжать двигаться до тех пор, пока действует команда *Jog Minus* и пока не достигнута сконфигурированная программная граница *Positive End Of Travel (Отрицательный конец передвижения)* или переключатель Отрицательного перебега. Входы переключателя перебега могут быть выключены при помощи конфигурационного параметра *OT Limit*. Команда *Jog Minus* может быть использована для отхода толчком от переключателя Положительного перебега, если бит *Clear Error (Сбросить ошибку) %Q* также сохраняется во включенном состоянии. *Дополнительную информацию по Толчковому передвижению при помощи модуля DSM314 см. в гл. 6 "Непрограммируемое движение".*
- 3.11 Сброс флага строба 1, 2.** Флаг бита статуса *Strobe n Flag (Флаг строба n) %I* информирует ПЛК о том, что Импульс строба захватил позицию оси, которая теперь сохранена в соответствующем слове статуса *Strobe n Position (Позиция строба n) %AI*. После того как ПЛК подтвердил эти данные, он может использовать бит команды *Reset Strobe n Flag (Сброс флага строба n) %Q*, чтобы очистить флаг бита статуса *Strobe n Flag (Флаг строба n) %I*. До тех пор пока установлен бит *Strobe n Flag %I*, дополнительные импульсы строба не вызывают захвата новых данных. Флаг должен быть удален (очищен), чтобы можно было выполнить захват другой позиции строба. До тех пор пока командный бит *Reset Strobe n Flag %Q* остается во включенном состоянии, бит *Strobe n Flag* будет находиться в очищенном состоянии. В этой ситуации последняя позиция импульса строба будет зафиксирована в слове статуса *Strobe n Position (Позиция строба n)*, однако флаг не может быть использован в ПЛК, чтобы информировать о присутствии новых данных.

- 3.12 OUT1\_A, B, C, D управление выходом / Конфигурируемый источник для CTL-битов.** Разъем каждой оси имеет выход твердотельного реле (SSR) 24 В пост. тока с номинальным током 125 мА. Биты *OUT1\_A*, *OUT1\_B*, *OUT1\_C* и *OUT1\_D Output Control (Управление выходом) %Q* могут управлять состоянием соответствующего выхода, но только в том случае, если соответствующая конфигурация Битов выхода установлена для управления ПЛК. Информацию по конфигурации см. в главе 4.

Для каждой оси имеются следующие контакты разъемов:

	Контакт разъема лицевой панели	Контакт вспомогательной клеммной колодки IC693ACC336	Контакт клеммной колодки цифрового сервоустройства IC693ACC335
OUT1 SSR (+) контакт	18	18	8
OUT1 SSR (-) контакт	36	36	16

Эти биты %Q действуют также как источники для конфигурируемых CTL-битов независимо от конфигурации Битов выхода. Информацию о конфигурировании источников битов CTL01-CTL24 см. в гл. 4.

**Примечание:** Биты OUT\_1A, B, C, D не управляют выходами лицевой панели, пока соответствующая конфигурация Битов выхода не установлена на управление ПЛК. Информацию по конфигурации см. в главе 4.

- 3.13 OUT3\_A, B, C, D управление выходом / Конфигурируемый источник для CTL-битов.** Разъем каждой оси имеет дифференциальный выход 5 В пост. тока, который пригоден для приведения в действие 5 В ТТЛ- или КМОП-нагрузок. Биты *OUT3\_A*, *OUT3\_B*, *OUT3\_C* и *OUT3\_D Output Control (Управление выходом) %Q* могут управлять состоянием соответствующего выхода, но только в том случае, если соответствующая конфигурация Битов выхода установлена для управления ПЛК. Информацию по конфигурации см. в главе 4.

Для каждой оси имеются следующие контакты разъемов:

	Контакт разъема лицевой панели	Контакт вспомогательной клеммной колодки IC693ACC336	Контакт клеммной колодки цифрового сервоустройства IC693ACC335
OUT1 SSR (+) контакт	14	14	5
OUT3 (-) контакт	32	32	13

**Примечание:** Биты OUT\_3A, B, C, D не управляют выходами лицевой панели, пока соответствующая конфигурация Битов выхода не установлена на управление ПЛК. Информацию по конфигурации см. в главе 4.

Эти биты %Q действуют также как источники для конфигурируемых CTL-битов независимо от конфигурации Битов выхода. Информацию о конфигурировании источников битов CTL01-CTL24 см. в гл. 4.

- 3.14 Включить слежение.** Если этот бит *Enable Follower* (Включить слежение) установлен и бит статуса *Follower Enabled* (*Слежение включено*) %I показывает, что слежение включено, то движение, задаваемое внешним или внутренним ведущим устройством, будет действовать как вход на контур следящего устройства. Для инициации движения слежения может быть сконфигурирован дополнительный бит Триггера слежения. Если используется Триггер слежения, то *Enable Follower* должен быть в состоянии ON (ВКЛ.), чтобы можно было проверить состояние триггера. Очистка *Enable Follower* вызывает отсоединение контура слежения от ведущего источника. Команды *Jog*, *Move at Velocity* и *Execute Program n* разрешаются независимо от состояния *Enable Follower*. Если Слежение включено, то команды *Jog*, *Move at Velocity* или *Execute Program n* накладываются на команды скорости или позиции ведущего устройства. *Find Home* (*Поиск Исходной позиции*) не разрешается, пока бит *Enable Follower* не очищен. Дополнительную информацию см. в главе 8. Этот бит используется только в режиме слежения.
- 3.15 Выбрать ведущий источник следящего устройства.** Этот бит *Select Follower Internal Master* (*Выбрать ведущий внутренний источник следящего устройства*) переключает источник ведущей оси с Источник 1 (бит OFF) на Источник 2 (бит ON). Ведущие источники следящего устройства могут быть отконфигурированы как Заданная позиция или Текущая позиция с любой из 4 осей.

## Раздел 4: Команды немедленного действия %AQ

Приведенные ниже слова команд немедленного действия %AQ передаются на каждом цикле ПЛК из %AQ-данных ЦП на модуль DSM314. Количество сконфигурированных %AQ-слов (6, 9 или 12) зависит от количества сконфигурированных управляемых осей. Текущие адреса слов команд немедленного действия зависят от начального адреса, сконфигурированного для слов %AQ. См. таблицу 4-1 "Таблица установочных параметров". Номера смещения слов, приведенные в таблице ниже, являются смещениями для данного начального адреса. Назначение слов следующее:

Таблица 5-6. Назначения слов %AQ

Смещение слова	Описание	Ось
00	Слово команды немедленного действия	Серво 1
01-02	Данные команды	Серво 1
03	Слово команды немедленного действия	Серво 2
04-05	Данные команды	Серво 2
06	Слово команды немедленного действия	Серво 3
07-08	Данные команды	Серво 3
09	Слово команды немедленного действия	Серво 4
10-11	Данные команды	Серво 4

**На каждом цикле ПЛК на модуль DSM314 может быть послана только одна команда немедленного действия %AQ; единственным исключением является команда *Load Parameter Immediate (Загрузить параметр немедленно)*, которая не зависит от оси. Количество команд *Load Parameter Immediate (Загрузить параметр немедленно)*, которое может быть отправлено в одном цикле, зависит от количества сконфигурированных слов %AQ (см. дополнительную информацию в табл. 5-8).**

Однако, несмотря на то, что команды могут отправляться на каждом цикле, модуль DSM314 будет реагировать на команду ТОЛЬКО В ТОМ СЛУЧАЕ, если она изменилась после последнего цикла. Если любое слово из 3 изменилось, то DSM314 воспримет данные как новую команду и соответственно отреагирует.

**Бит *Axis OK (Ось в норме)* %I должен быть установлен для оси в состояние ON, чтобы можно было принять новую команду немедленного действия %AQ.** При определенных условиях, таких как отключенный цифровой датчик положения, незапитанный сервоусилитель или неочищенная ошибка, *Axis OK* будет в состоянии OFF; выполнение команды %AQ для такой оси будет отключено. Если Цифровая сервоось 1 или 2 не используется для управления двигателем, то конфигурируемый ***Motor Type (Тип двигателя)*** должен быть установлен на 0; в противном случае будет выдано сообщение об ошибке, и бит *Axis OK* будет находиться в состоянии OFF.

Шестибайтовый формат для команд немедленного действия определен в таблице 5-7. Текущие адреса слов команд немедленного действия зависят от начального адреса, сконфигурированного для ссылок %AQ. **Номера слов, приведенные в таблице ниже, являются смещениями для данного начального адреса.**

Смещения слов показаны в обратном порядке и в шестнадцатиричном формате, чтобы упростить ввод данных. В следующем примере показана отправка команды Set Position (Установить позицию) на ось 1. Первое слово (слово 0) содержит номер текущей команды. Для команды Set Position таким номером команды является 0023h. Второе и третье слова содержат данные для команды Set Position, относящиеся к позиции. Второе слово (слово 1) является младшим словом позиции, а третье (слово 2) – старшим.

**Пример:**

Чтобы задать позицию 3 400 250, следует сначала преобразовать значение в шестнадцатиричный формат. Десятичное число 3 400 250 соответствует шестнадцатиричному числу 0033E23A. Для данного числа 0033 является младшим словом, а E23A – старшим. Данные, подлежащие отправке на модуль DSM314, выглядят следующим образом:

<i>Слово 2</i>	<i>Слово 1</i>	<i>Слово 0</i>	<i>Команда</i>
0033	E23A	0023	Set Position (Установить позицию) 3 400 250

Задание слова 0 в виде шестнадцатиричного слова, а слов 1 и 2 в виде двойного целого в представлении таблицы просмотра ссылок упрощает ввод команд немедленного действия.

**Предельные значения данных для MaxPosnUu, MaxVelUu и MaxAccUu могут быть рассчитаны следующим образом:**

<b>Формулы для расчета переменных, определяющих предельные значения данных</b>		
<b>Граница позиции MaxPosnUu</b>	<b>Граница скорости MaxVelUu</b>	<b>Граница ускорения MaxAccUu</b>
Если отношение uu:cts ("пользовательские единицы/отсчеты") $\geq 1:1$ , MaxPosnUu = 536 870 912	MaxVelUu = 1,000,000* пользовательские единицы/отсчеты	Если отношение uu:cts ("пользовательские единицы/отсчеты") $\geq 1:1$ , MaxAccUu = 1 073 741 823
Иначе (если uu:cts < 1:1) MaxPosnUu = 536 870 912* пользовательские единицы/отсчеты		Иначе (если uu:cts < 1:1) MaxAccUu = 1 073 741 823* пользовательские единицы/отсчеты

В приведенной далее таблице команд %AQ указаны смещения слов только для сервооси 1. Смещения слов для других осей рассчитываются добавлением 3 (сервоось 2), 6 (сервоось 3) или 9 (сервоось 4) к указанным смещениям слов. Номера в столбце "Ссылки" относятся к разделам данной главы.

Таблица 5-7. Команды немедленного действия %AQ, использующие шестибайтовый формат

Слово 2		Слово 1		Слово 0		Определение команды немедленного действия	Ссылка
Байт 5	Байт 4	Байт 3	Байт 2	Байт 1	Байт 0		
xx	xx	xx	xx	00	00h	Нуль	4.01
xx	xx	xx	RO%	00	20h	Подстановка скорости RO% = 0 ... 120%	4.02
xx	xx	*	Инкр.	00	21h	Приращение позиции без обновления позиции Инкр. = -128 ... +127 пользовательских единиц	4.03
Скорость				00	22h	Движение на скорости Скорость = -MaxVelUu ... +MaxVelUu	4.04
Позиция				00	23h	Установить позицию Поз. = -MaxPosnUu ... + MaxPosnUu-1	4.05
xx	xx	Аналоговый выход		00	24h	Включить аналоговый выход Аналоговый выход = -32 000 ... + 32 000	4.06
xx	xx	*	Инкр.	00	25h	Приращение позиции с обновлением позиции Инкр. = -128 ... +127 пользовательских единиц	4.07
xx	xx	xx	Зона прихода в позиц.	00	26h	Зона Прихода в Позицию Диапазон = 0 ... 255	4.08
Позиция или параметр #				Тип движения	27h	Команда движения Поз. = -MaxPosnUu ... + MaxPosnUu-1 Параметр # = 0 ... 255	4.09
Скорость				00	28h	Скорость Толчка Скорость = +1 ... +MaxVelUu	4.10
Ускорение				00	29h	Ускорение Толчка Ускорение = +1 ... + MaxAccUu	4.11
xx	xx	Постоянная времени (единицы 0.1 мс)		00	2Ah	Постоянная Времени Контура Позиционирования Постоянная Времени = 0 - 65535 (единицы 0.1 мс)	4.12
xx	xx	VFF (0.01% единицы)		00	2Bh	Упреждение по скорости подачи (VFF) VFF = 0 ... 12000 (0.01% единицы)	4.13
xx	xx	Постоянная времени интегратора		00	2Ch	Постоянная времени интегратора Постоянная времени = 0,10 ... 10 000 мс	4.14
Отношение B		Отношение A		00	2Dh	A/B Отношение Следящего Устройства Отношение A = -32 768 ... +32 767 Отношение B = +1 ... +32 767	4.15
xx	xx	xx	VLGN	00	2Eh	Коэффициент обратной связи по скорости (VLGN) (только цифровой режим) VLGN = 0 ... 255	4.16
xx	xx	Предел по моменту вращения (0.01% единицы)		00	2Fh	Предел по моменту вращения (только цифровой режим и аналоговый режим момента вращения) Диапазон = 0-10000 (0.01% единицы)	4.17

Таблица 5-7. - Продолжение - Команды немедленного действия %AQ, использующие шестибайтовый формат

Слово 2		Слово 1		Слово 0		Определение команды немедленного действия	Ссылка
Байт 5	Байт 4	Байт 3	Байт 2	Байт 1	Байт 0		
Позиция				00	31h	Установить позицию вспомогательного датчика положения Позиция = -MaxPosnUu ... + MaxPosnUu-1	4.18
xx	xx	Команда скорости сервоустройства (Примечание: Не используется в Аналоговом режиме скорости – см. команду Force D/A Output (Включить Цифровой / Аналоговый Выход))		00	34h	Включить скорость сервоустройства Команда скорости сервоустройства = -4 095 ... +4 095 об./мин.	4.19
xx	xx	Смещение		Режим	40h	Выбрать возвращаемые данные 1	4.20
xx	xx	Смещение		Режим	41h	Выбрать возвращаемые данные 2	4.21
xx	xx	Длительность компенсации		00	42h	Длительность участка линейного компенсирующего изменения скорости следящего устройства Активный диапазон = 0,10 ... 32000 мс	4.22
xx	xx	KpVel		07	46h	Коэффициент передачи пропорционального регулятора скорости (KpVel) (только аналоговый режим момента вращения) KpVel = 0 - 32767	4.23
xx	xx	KiVel		08	46h	Коэффициент передачи интегрального регулятора скорости (KiVel) (только аналоговый режим момента вращения) KiVel = 0 - 32767	4.24
xx	xx	Режим TqFilt		0A	46h	Фильтр уставок момента вращения (TqFilt) (только аналоговый режим момента вращения) TqFilt = 0 - 3	4.25
xx	xx	Режим		Ось	47h	Выбрать режим аналогового выхода (только цифровой режим)	4.26
xx	xx	xx	xx	00	49h	Удалить новую полученную конфигурацию	4.27
Данные параметра				Par #h	50h	Загрузить параметр немедленно Par # = 0 ... 255 Данные параметра = диапазон зависит от использования параметра.	4.28

\* = Допускается только 00 или FFh.

xx = не имеет значения

**4.01 Нуль.** Эта команда Null (Нуль) является командой немедленного действия %AQ по умолчанию. Поскольку слова %AQ автоматически передаются на каждом цикле ПЛК, то команда *Null* должна использоваться всегда, чтобы предотвратить несанкционированное выполнение другой команды немедленного действия %AQ.

- 4.02 Подстановка скорости.** Эта команда *Rate Override (Подстановка скорости)* немедленно изменяет значение % подстановки скорости подачи, которое изменяет заданные скорости для всех последующих программных движений. Это новое значение начинает действовать сразу же при получении его модулем DSM314. Оно запоминается и продолжает действовать до тех пор, пока не будет перезаписано другим значением. Подстановка скорости не действует на непрограммируемое движение или на ускорение. Значение *Rate Override (Подстановка скорости)* устанавливается на 100% при каждой инициализации программы. Команда *Rate Override* может быть направлено в том же цикле ПЛК, что и бит *Execute Program (Выполнить программу) %Q*; значение подстановки окажет немедленное воздействие. Команда *Rate Override* может быть использована для настройки запрограммированной скорости (не ускорения) специфического движения или набора движения для любой данной оси.
- 4.03 Приращение позиции без обновления позиции.** (Пользовательские единицы). Эта команда *Position Increment Without Position Update (Приращение позиции без обновления позиции)* смещает позицию оси в диапазоне от – 128 до + 127 пользовательских единиц без обновления *Текущей позиции* или *Заданной позиции*. Модуль DSM314 немедленно передвинет ось на заданное приращение, если сервоустройство включено. Команда Приращения позиции может быть использована для выполнения небольших коррекций позиции на станке, чтобы скомпенсировать изменение текущих условий. **Дополнительную информацию по командам приращения позиции при помощи модуля DSM314 см. в гл. 6 "Непрограммируемое движение".**
- 4.04 Движение на скорости.** (Пользовательские единицы в секунду). Эта команда *Move at Velocity (Движение на скорости)* выдается с ПЛК, чтобы перемещать ось с постоянной скоростью. Для команды *Move at Velocity* используются текущие данные *Jog Acceleration (Ускорение толчка)* и сконфигурированный *Jog Acceleration Mode (Режим Ускорения толчка)*. Данные текущей позиции оси в ходе такого перемещения переходят при сконфигурированной верхней или нижней границе, достигая ее. **Дополнительную информацию по команде Move at Velocity см. в гл. 6 "Непрограммируемое движение".**
- 4.05 Установить позицию.** (Пользовательские единицы). Команда *Set Position (Установить позицию)* изменяет значения регистра позиции оси без перемещения самой оси. Действие команды зависит от конфигурации оси.

**Сервоось.** Изменяется как *Заданная позиция*, так и *Текущая позиция*, в результате чего команда движения не вырабатывается. Для *Текущей позиции* устанавливается предписанное значение, а для *Заданной позиции* – это значение + *Ошибка по положению*. Команда *Set Position* не может быть выполнена, если бит *Moving %I* или бит *Program Active %I* находится в состоянии ON. Команда *Set Position* допускается, если бит *In Zone (В зоне) %I* находится в состоянии OFF, тогда как *Actual Velocity (Текущая скорость) ≤ 100* отсчетов / секунда. Значение позиции должно быть в пределах конца передвижения и в пределах отсчетов; в противном случае будет выдана ошибка статуса. Бит *Position Valid (Позиция достигнута) %I* устанавливается после успешного выполнения команды *Set Position*. См. в приложении С информацию по использованию датчиков положения в абсолютном режиме. Команда *Set Position* обычно используется для установки исходной точки стартовой позиции на ноль (или другое значение) без возврата оси в исходное положение.

**Вспомогательная ось.** *Commanded Position (Заданная позиция)* устанавливается на предписанное значение. Для вспомогательной оси *Actual Position (Текущая позиция)* не зависит от *Заданной позиции* и на нее не влияет команда *Set Position*. **Порядок установки Текущей позиции для датчика положения вспомогательной оси см. в параграфе 4.18 "Установить позицию датчика положения вспомогательной оси".** Команда *Set Position* не может быть выполнена, если бит *Moving %I* или бит *Program*

*Active %I* находится в состоянии ON. Значение позиции должно быть в пределах конца передвижения; в противном случае будет выдана ошибка статуса.

**Примечание:** Если на цифровую сервосистему GE Fanuc с абсолютным датчиком положения (*Feedback Mode* (Режим обратной связи) = абсолютный) питание подается первый раз после удаления или замены его аккумулятора, то следует повернуть датчик положения за его внутреннюю опорную точку. Если это не сделано, то команда *Set Position* будет проигнорирована, и будет выдан код ошибки 53h (Set Position до того, как датчик положения прошел опорную точку).

**4.06 Включить аналоговый выход.** Разъем каждой оси поддерживает один аналоговый выходной сигнал. Команда немедленного действия *Force Analog Output* (Включить аналоговый выход) может быть использована в прикладной программе ПЛК, чтобы задать значение напряжения этого выхода постоянного тока. Команда *Force Analog Output* воздействует на один из аналоговых выходов на разъеме С или D лицевой панели DSM в Цифровом режиме или в Аналоговом режиме скорости на разъеме А, В, С или D. Множественные команды *Force Analog Output* могут быть использованы, чтобы управлять выходами на различных разъемах при использовании соответствующих смещений слова %AQ (см. параграф перед таблицей 5-7). Команда *Force Analog Output* имеет диапазон от +32000 (+10.00 В пост. тока) до -32000 (-10.00 В пост. тока). Если ось сконфигурирована для Аналогового режима момента вращения, то команда *Force Analog output* ОТСУТСТВУЕТ.

Чтобы активировать значение аналогового выхода, заданное этой командой, необходимо включить соответствующий бит %Q “Enable Drive” (имеется по одному такому биту для каждой оси).

**Примечание:** Это не соответствует функциональным возможностям IC693DSM302.

Имеются два требования, направленные на поддержание включенного напряжения аналогового выхода: (1) команда *Force Analog Output* и значение должны присутствовать постоянно в данных %AQ, и (2) Соответствующий бит %Q “Enable Drive” должен быть в состоянии ON. Бит %Q “Enable Drive” может быть использован для включения и выключения напряжения на аналоговом выходе.

Если для данной оси действует команда *Force Analog Output*, то любая другая %AQ команда немедленного действия для этой оси удалит команду *Force Analog Output* и выключит соответствующий аналоговый выход.

Есть некоторое различие в использовании этой команды в Цифровом и Аналоговом режимах оси; это различие поясняется далее.

### Цифровой режим

- Команда *Force Analog Output* (Включить аналоговый выход) в Цифровом режиме может быть использована только на разъемах С и D (в Цифровом режиме обе оси 1 и 2 на разъемах А и В, соответственно, должны быть цифровыми). Фактически *Force Analog Output* является выходом по умолчанию на разъемах С и D в Цифровом режиме.
- Если оси 1 и 2 (разъемы А и В) сконфигурированы для цифрового сервоустройства, то их аналоговые выходы используются только для сервонастройки, и эта функция не может быть перекрыта командой *Force Analog Output*. Выдача команды *Force*

*Analog Output* на цифровую ось (разъем А или В) не будет иметь влияния; сообщение об ошибке не появится.

- В Цифровом режим сигнал *Force Analog Output* может быть перекрыт, если другой сигнал направлен на разъем С или D командой *Select Analog Output Mode* (*Выбрать режим аналогового выхода*). Если команда по умолчанию *Force Analog Output* (*Включить аналоговый выход*) была перекрыта на разъемах С или D, она может быть переустановлена либо (1) путем выдачи команды немедленного действия *Select Analog Output* (*Выбрать аналоговый выход*) (код сигнала 00) на каждую задействованную ось, либо (2) выключением и включением модуля DSM314. См. раздел 4.25 "Выбрать режим аналогового выхода".

### Пример использования команды Force Analog Output (Цифровой режим)

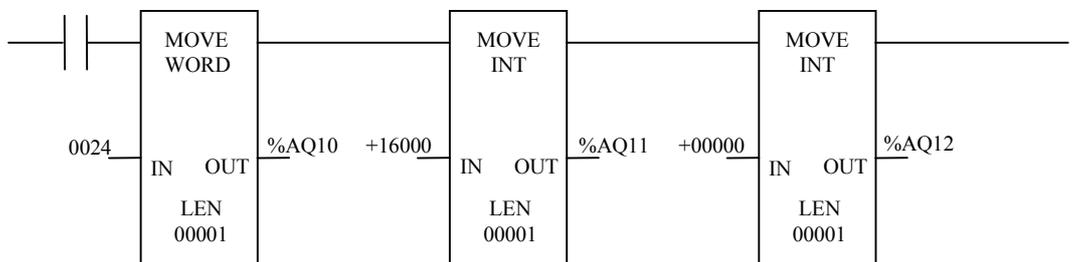
В данном примере оси 1 и 2 сконфигурированы как цифровые, начальный адрес DSM314 %Q сконфигурирован как %Q1, а начальный адрес %AQ сконфигурирован как %AQ1. Разъемы С и D на их условия аналогового выхода по умолчанию (*Force Analog Output* (*Включить аналоговый выход*)).

Чтобы включить аналоговый выход +5 В постоянного тока на разъеме D, должна быть выдана в программе Ladder Logic команда немедленного действия *Force Analog Output*. После того как первое слово %AQ было сконфигурировано как %AQ1, тремя словами, которые применяются к разъему D ("Ось 4"), являются %AQ10, %AQ11 и %AQ12 (подробности см. в параграфе перед таблицей 5-7). Поскольку %Q1 было сконфигурировано как первый %Q-бит, то битом *Enable Drive* (*Включить привод*) (*Серво 4*) является %Q67 (см. таблицу 5-5 "Дискретные команды").

Таким образом, в соответствующие слова должны быть перемещены приведенные ниже значения с использованием инструкций Move в ladder logic (использование инструкции Move, имеющей тип WORD'a, облегчает перемещение шестнадцатиричного числа):

- %AQ10 Установить 24h (это определяет команду *Force Analog Output* (*Включить аналоговый выход*))
- %AQ11 Установить +16000 (это равно +5 В постоянного тока)
- %AQ12 Установить 0 (это слово не используется для перемещения существенных данных)

Кроме того, в бите %Q67 (*Enable Drive*) должна быть установлена логическая 1.



### Аналоговый режим скорости

- В Аналоговом режиме скорости команда *Force Analog Output* может быть использована на всех четырех разъемах для включения выходного напряжения.

- Команда *Select Analog Output* (*Выбрать аналоговый выход*), обсуждавшаяся в разделе "Цифровой режим" выше, не работает в Аналоговом режиме.

### Аналоговый режим момента вращения

- В Аналоговом режиме момента вращения команда *Force Analog Output* ОТСУТСТВУЕТ.

**4.07 Приращение позиции с обновлением позиции.** (Пользовательские единицы). Команда *Position Increment with Position Update* (*Приращение позиции с обновлением позиции*) аналогична команде *Position Increment Without Position Update* (*Приращение позиции без обновления позиции*) (#21h) за исключением того, что *Текущая* и *Заданная позиции* (возвращаемые в данных %AI) обновляются на величину приращения. Если сервоустройство включено, модуль DSM314 немедленно передвинет ось на значение приращения. Команда Приращения позиции может быть использована для выполнения небольших коррекций позиции на станке, чтобы скомпенсировать изменение текущих условий. **Дополнительную информацию по командам приращения позиции при помощи модуля DSM314 см. в гл. 6 "Непрограммируемое движение".**

**4.08 Зона Прихода в Позицию.** (Пользовательские единицы). Команда *In Position Zone* (*Зона прихода в позицию*) может быть использована для задания активной **Зоне прихода в позицию** значения, отличного от сконфигурированного.

Модуль DSM314 сравнивает **In Position Zone** с Ошибкой по положению, чтобы управлять битом *In Zone %I*. Когда Ошибка по положению  $\leq$  **In Position Zone**, тогда бит *In Zone %I* находится в состоянии ON.

Если происходит выключение-включение питания DSM314 или перезапуск ЦП ПЛК по каким-либо причинам, то значение, установленное данной командой, теряется и устанавливается то значение *In Position zone*, которое было задано при конфигурации.

**4.09 Команда движения.** Эта команда создает одиночный профиль движения, в соответствии с которым будет перемещаться ось в заданную позицию каждый раз, когда выдается эта команда. Для такого движения будет использоваться *Jog Acceleration (Ускорение толчка)* и *Jog Velocity (Скорость толчка)* (которые также могут быть изменены командами %AQ). Выполнение команды PMOVE считается незавершенным (бит *Program Active (Программа активна)* %I остается в состоянии ON) до тех пор, пока *Commanded Position (Заданная позиция)* не достигнет запрограммированной точки и бит *In Zone (В зоне)* %I не перейдет в состояние ON. Выполнение команды CMOVE считается завершенным (бит *Program Active (Программа активна)* %I переходит в состояние OFF), когда *Commanded Position (Заданная позиция)* достигнет запрограммированной точки, даже если бит *In Zone* остается в состоянии OFF. Поэтому выполнение команды CMOVE будет прекращаться, даже если Текущая позиция еще не достигла запрограммированной точки CMOVE. Наблюдая за битом *Program Active* %I, можно определить, когда команда AQ Move активна.

Поле данных для этой команды может содержать позицию движения или расстояние в байтах 2-5 с типом команды (в шестнадцатиричном формате), как показано ниже:

Тип движения (байт 1):

00h = Абсолютный, Pmove, линейный  
01h = Абсолютный, Cmove, линейный  
10h = Абсолютный, Pmove, S-кривая  
11h = Абсолютный, Cmove, S-кривая  
40h = Инкремент., Pmove, линейный  
41h = Инкремент., Cmove, линейный  
50h = Инкремент., Pmove, S-кривая  
51h = Инкремент., Cmove, S-кривая

Поле данных для этой команды может содержать номер параметра в байте 2 (байты 3-5 не используются) с типом команды, как показано ниже:

Тип движения (байт 1):

80h = Абсолютный, Pmove, линейный  
81h = Абсолютный, Cmove, линейный  
90h = Абсолютный, Pmove, S-кривая  
91h = Абсолютный, Cmove, S-кривая  
C0h = Инкремент., Pmove, линейный  
C1h = Инкремент., Cmove, линейный  
D0h = Инкремент., Pmove, S-кривая  
D1h = Инкремент., Cmove, S-кривая

Команда *Move* выполняется как одноразовая программа передвижения. Поэтому все ограничения, применимые к выполнению программы движения, действуют также на *Move Command*. Например, если какая-то программа уже активна на оси 1, то попытка направить эту команду для оси 1 приведет к сообщению о ситуации сбоя.

- 4.10 Скорость Толчка.** (Пользовательские единицы в секунду). Команда *Jog Velocity* (*Скорость Толчка*) задает скорость, используемую для толчка в положительном или отрицательном направлении, когда используется бит *Jog %Q*. *Jog Velocity* (*Скорость Толчка*) используется в программах движения, если в программу не включена команда *Velocity* (*Скорость*). *Jog Velocity* всегда используется командой *%AQ Move* (27h). Перезапуск ПЛК или цикл выключения – включения устанавливает это значение на сконфигурированное.
- 4.11 Ускорение Толчка.** (Пользовательские единицы в секунду за секунду). Команда *Jog Acceleration* (*Ускорение Толчка*) задает значение ускорения, используемое в операциях *Jog* (*Толчок*), *Find Home* (*Поиск Исходной Позиции*), *Move at Velocity* (*Движение на скорости*), *Abort All Moves* (*Прекратить все движения*) и **Normal Stop (Нормальный останов)**. **Normal Stop (Нормальный останов)** происходит, когда ПЛК переключается от Run на Stop или после определенных программных ошибок (см. Приложение А). *Jog Acceleration* (*Ускорение Толчка*) используется в программах движения, если в программу не включена команда *Acceleration* (*Ускорение*). *Jog Acceleration* всегда используется командой *%AQ Move* (27h). Перезапуск ПЛК или цикл выключения – включения устанавливает это значение на сконфигурированное.
- Примечание:** В модуле DSM314 используется минимальное значение после масштабирования. Это значение определяется следующим правилом:  

$$\text{Jog Acc} * (\text{пользовательские единицы/отсчет}) \geq 32 \text{ отсчета/секунда/секунда.}$$
- 4.12 Постоянная Времени Контура Позиционирования.** (0.1 миллисекунды). Команда *Position Loop Time Constant* (*Постоянная времени контура позиционирования*) позволяет изменить постоянную времени контура позиционирования сервопривода, заданную при конфигурации. Чем меньше это значение, тем быстрее реагирует система. Однако слишком малые значения вызывают неустойчивость системы и возникновение колебаний. Чтобы точно отслеживать заданный профиль скорости, *Position Loop Time Constant* должна быть от 1/4 до 1/2 МИНИМАЛЬНОГО времени ускорения или замедления системы. В Аналоговом режиме для правильного действия *Position Loop Time Constant* следует правильно задать значение конфигурации "Vel at Max Cmd". Перезапуск ПЛК или цикл выключения – включения устанавливает это значение на сконфигурированное.
- 4.13 Упреждение по скорости.** Команда *Velocity Feedforward* (*Упреждение по скорости*) задает величину *Velocity Feedforward* (0.01%). Эта величина представляет собой процент от Заданной скорости, который добавляется к выходной команде скорости модуля DSM314. Увеличение значения *Velocity Feedforward* (*Упреждение по скорости*) увеличивает быстродействие сервопривода и уменьшает ошибку по положению. Оптимум значений *Velocity Feedforward* лежит около 90-100%. Для аналоговых сервоприводов для правильного действия установленной величины *Velocity Feedforward* следует правильно задать значение конфигурации "Vel at Max Cmd" ("Скорость при максимальной команде"). Перезапуск ПЛК или цикл выключения – включения устанавливает это значение на сконфигурированное.
- 4.14 Постоянная времени интегратора.** (Миллисекунды). Команда *Integrator Time Constant* (*Постоянная времени интегратора*) задает величину *Постоянной времени* для интегратора ошибки по положению. Эта величина представляет собой интервал времени, в течение которого убирается 63% Ошибки по положению. *Integrator Time Constant* (*Постоянная времени интегратора*) должна быть в 5 – 10 раз больше значения *Position Loop Time Constant* (*Постоянная времени контура позиционирования*), чтобы предупредить возникновение неустойчивости и колебаний. **Рекомендуется использовать интегратор ошибки по положению только в длительно работающих следящих системах. Использование интегратора в**

приложениях с поточечным перемещением может вызывать перерегулирование по положению при остановке.

- 4.15 A/B Отношение Следящего Устройства.** Команда *Follower A/B Ratio (A/B Отношение Следящего Устройства)* позволяет ПЛК изменять отношение A/B подчиненный / ведущий, используемое в каждом контуре обратной связи. "А" является 16-битовым целым числом со знаком в диапазоне от -32 768 до +32 767. "В" является 16-битовым целым числом в диапазоне от 1 до 32 767. Отношение A/B должно быть в диапазоне от 32:1 до 1:10 000; в противном случае будет выдано сообщение об ошибке статуса. Дополнительную информацию об отношении A/B см. в главе 8.
- 4.16 Коэффициент обратной связи по скорости. (VLGN).** Только в Цифровом режиме и в Аналоговом режиме момента вращения. Коэффициент обратной связи по скорости для цифровой сервооси и для сервоустройства в Аналоговом режиме момента вращения GE Fanuc может быть задан командой *Velocity Loop Gain (Коэффициент обратной связи по скорости)*. Изменение значения VLGN используется для согласования инерционности нагрузки ( $J_L$ ) с инерционностью двигателя ( $J_M$ ). VLGN имеет значение по умолчанию, равное 16, что представляет отношение инерционностей 1:1. Это значение VLGN рассчитывается в предположении, что нагрузка жестко связана с двигателем. Однако в реальной ситуации требуемое значение может существенно отличаться от расчетного из-за различного сочетания жесткости, трения, люфтов и т.п. Сброс или выключение-включение ПЛК устанавливают сконфигурированное значение VLGN. Предлагаемая исходная точка для значения *Velocity Loop Gain*:

$$\text{Коэффициент Обратной связи по Скорости} = \frac{\text{Инерционность нагрузки } (J_L)}{\text{Инерционность двигателя } (J_M)} \times 16$$

Допустимый диапазон *Velocity Loop Gain* составляет 0 – 255.

*Пример:* Момент инерции двигателя ( $J_M$ ) некоторого сервоустройства составляет 0.10 фунт x фут x с<sup>2</sup>. Момент инерции нагрузки ( $J_L$ ) для данного приложения составляет 0.05 фунт x фут x с<sup>2</sup>.  $VLGN = (0.05 / 0.10) * 16 = 8$

Значение этого коэффициента по умолчанию задается в установочном параметре *Velocity Loop Gain (Коэффициент обратной связи по скорости)* при конфигурации.

### Предостережение

**Неправильное значение VLGN может вызвать неустойчивость оси. Любые изменения значения VLGN следует выполнять с осторожностью.**

- 4.17 Предел по моменту вращения.** (0.01%). Только в Цифровом режиме и в Аналоговом режиме момента вращения. Команда *Torque Limit (Предел по моменту вращения)* дает метод ограничения момента вращения, создаваемого серводвигателем GE Fanuc. В Аналоговом режиме момента вращения значение предела по моменту вращения ограничивает заданный момент процентом от заданного полного значения момента вращения. В частности происходит ограничение полного диапазона аналогового выхода, когда полный диапазон равен 10 В. Модуль DSM314 устанавливает *Torque Limit (Предел по моменту вращения)* на значение по умолчанию 10000 (100%), если происходит выключение – включение питания или сброс. Прикладная логика ПЛК должна установить любое другое значение для требуемого *Предела по моменту вращения*. Разрешенным диапазоном для *Предела по моменту вращения* является диапазон от 0 до 10000 в единицах 0.01%. Это соответствует 0 – 100% максимального момента вращения при заданной скорости. Если задается значение в диапазоне 10001 – 65535, то предел по моменту вращения устанавливается на 10000. Значение *Предела по моменту вращения* может быть изменено во время движения оси; это изменение

начинает действовать сразу же. Для определения текущего значения момента вращения, имеющегося для данной скорости, см. кривую изменения крутящего момента двигателя в соответствующем руководстве по серводвигателю. Простым примером использования *Предела по моменту вращения* может быть защита от чрезмерного усилия затягивания на станке.

#### 4.18 Установить позицию вспомогательного датчика положения.

(Пользовательские единицы). Команда *Set Aux Encoder Position* (*Установить позицию вспомогательного датчика положения*) задает значение *Actual Position* (*Текущая позиция*) вспомогательного датчика положения без использования операции Find Home (Поиск исходной позиции). Бит *Position Valid* (*Позиция достигнута*) %I для вспомогательной оси устанавливается при получении команды.

#### 4.19 Включить скорость сервоустройства. (RPM). Только в Цифровом режиме и в Аналоговом режиме момента вращения.

Команда *Force Servo Velocity* (*Включить скорость сервоустройства*) обходит контур позиционирования и включает заданную скорость на цифровом сервоустройстве с целью подстройки. В Аналоговом режиме момента вращения эта команда обходит контур позиционирования и включает заданную скорость на регуляторе скорости. Управление ускорением не производится, и изменения скорости действуют немедленно. Значение +4095 команды *Force Servo Velocity* приведет к +4 095 об./мин. двигателя, а значение -4095 – к -4 095 об./мин. Контур управления цифровым сервоустройством могут ограничить текущую скорость двигателя меньшим значением. Следует быть внимательным, чтобы не использовать серводвигатель при нагрузках, превышающих его номинал.

Чтобы команда *Force Servo Velocity* могла быть выполнена, бит *Enable Drive* %Q должен быть установлен при отсутствии других заданных команд движения. Для правильной работы эта команда должна быть в течение длительного времени в данных %AQ. Если для данной оси действует команда *Force Servo Velocity*, то любая другая %AQ команда немедленного действия для этой оси удалит команду *Force Servo Velocity* и остановит сервопривод. *Дополнительная информация по команде Force Servo Velocity имеется также в главе 6 "Непрограммируемое движение".*

#### 4.20 Выбрать возвращаемые данные 1.

Команда *Select Return Data 1* (*Выбрать возвращаемые данные 1*) позволяет изменить данные, выводимые для каждой оси в отчет в раздел *User Selected Data 1* (*Данные 1, выбранные пользователем*) %AI. Альтернативные данные содержат такую информацию, как содержание памяти Параметров и редакция микропрограммы DSM314.

Команда *Select Return Data 1* использует выбор режима и выбор смещения. Выбор режима (смещение байта + 1 шестибайтовой команды) определяет тип возвращаемых данных. Выбор смещения (смещения байтов +2, +3 шестибайтовой команды) выбирает раздел индивидуальных данных для некоторых режимов. Установка режима 00h выводит сообщение о команде момента вращения, действующей по умолчанию. **Режим по умолчанию и смещение для Данных 1, выбранных пользователем, могут быть заданы в программе конфигурации модуля.**

#### 4.21 Выбрать возвращаемые данные 2.

Команда *Select Return Data 2* (*Выбрать возвращаемые данные 2*) позволяет изменить данные, выводимые для каждой оси в отчет в раздел *User Selected Data 2* (*Данные 2, выбранные пользователем*) %AI. Альтернативные данные содержат такую информацию, как содержание памяти Параметров и редакция микропрограммы DSM314.

Команда *Select Return Data 2* использует выбор режима и выбор смещения. Выбор режима (смещение байта + 1 шестибайтовой команды) определяет тип возвращаемых данных. Выбор смещения (смещения байтов +2, +3 шестибайтовой команды) выбирает раздел индивидуальных данных для некоторых режимов. Установка режима 00h выводит сообщение о команде момента вращения, действующей по умолчанию. **Режим**

по умолчанию и смещение для Данных 2, выбранных пользователем, могут быть заданы в программе конфигурации модуля.

Допускаются следующие комбинации для команд Select Return Data 1 и Select Return Data 2:

Цифр.	Аналог. момент вращения	Аналог. скорость	Выбираемые Возвращаемые данные	Режим данных	Смещение данных
Да	Да	Нет	Команда Момент вращения	00h	не используется
Да	Да	Да	Редакция микропрограммы DSM	10h	не используется
Да	Да	Да	Встроенный идентификационный номер микропрограммы DSM (шестнадцатиричный)	11h	не используется
Да	Нет	Нет	Абсолютное смещение обратной связи (в отсчетах)	17h	не используется
Да	Да	Да	Данные параметра	18h	номер параметра (0–255)
Да	Да	Да	CTL-биты 1-32	19h	не используется
Да	Да	Да	Аналоговые входы – Ось 1	1Ch	не используется
Да	Да	Да	Аналоговые входы – Ось 2	1Dh	не используется
Да	Да	Да	Аналоговые входы – вспомогательная ось 3	1Eh	не используется
Да	Да	Да	Аналоговые входы – вспомогательная ось 4	1Fh	не используется
Да	Да	Да	Заданная Позиция (пользовательские единицы)	20h	не используется
Да	Да	Да	Заданная позиция программы слежения (в отсчетах)	21h	не используется
Да	Да	Да	Нескорректированная Текущая позиция (в отсчетах)	28h	не используется
Да	Да	Да	Нескорректированная позиция строба 1 (в отсчетах)	29h	не используется
Да	Да	Да	Нескорректированная позиция строба 2 (в отсчетах)	2Ah	не используется

**Команда Момент вращения** смасштабирована так, что +/- 10000 = +/- 100% момента вращения.

**Редакция микропрограммы DSM** интерпретируется как два отдельных слова для старшего-младшего кодов редакции.

**Встроенный идентификационный номер микропрограммы DSM** интерпретируется как единое шестнадцатиричное слово.

**Абсолютное смещение обратной связи** представляет собой смещение (в отсчетах), которое используется для инициализации Текущей позиции, если применяется цифровой Абсолютный Датчик положения GE Fanuc. Текущая позиция = Данные Абсолютного датчика + Абсолютное смещение обратной связи

**Аналоговые входы** обеспечивают два слова данных для каждой оси: младшее слово = AIN1 и старшее слово = AIN2. Эти данные смасштабированы так, что +/- 32000 = +/- 10.0 В.

**Заданная позиция (пользовательские единицы)** является копией данных *Commanded Position (Заданная позиция) %AI*, представляемых для каждой оси. См. раздел 2.04 в главе 5.

**Заданная позиция программы слежения** (в отсчетах) представляет собой действующую заданную позицию (в отсчетах обратной связи), обновляемую и используемую

внутренним генератором команд движения. См. гл. 9 – Комбинированное движение под действием команд движения и команд следящего устройства.

**Нескорректированная текущая позиция** представляет собой накопленную текущую позицию (в отсчетах, не в пользовательских единицах) с 32-битовым двоичным значением перехода от  $-2\ 147\ 483\ 648$  до  $+2\ 147\ 483\ 647$ .

**Нескорректированная позиция строба 1** представляет собой *Нескорректированную текущую позицию*, захваченную Стробом 1.

**Нескорректированная позиция строба 2** представляет собой *Нескорректированную текущую позицию*, захваченную Стробом 2.

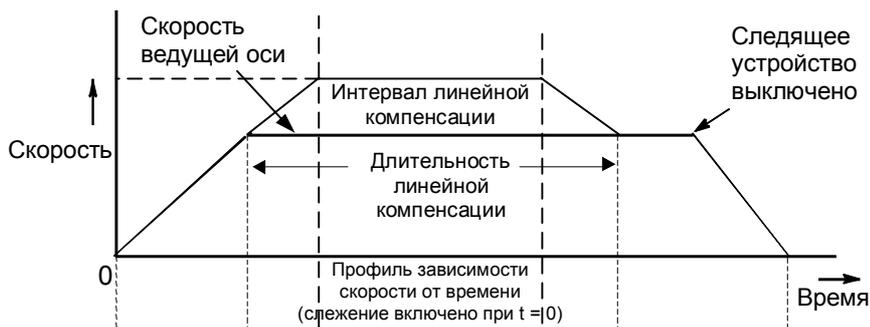
**Примечание:** Должно пройти не менее трех циклов ПЛК или 10 миллисекунд (в зависимости от того, что больше), прежде чем новые Выбранные Возвращаемые данные станут доступными в ПЛК.

**4.22 Длительность участка линейного компенсирующего изменения скорости следящего устройства.** Если выбрано свойство *Follower Ramp* (Линейное изменение скорости следящего устройства) и слежение включено, то ведомая ось выводится на скорость ведущей оси при сконфигурированном значении ***Follower Ramp Acceleration (Ускорение линейного изменения скорости)***, если значение скорости ведущей оси не равно нулю в момент включения слежения. Отсчеты ведущей оси, которые накапливаются в процессе ускорения ведомой оси, запоминаются. В этом режиме ведомая ось будет разгоняться до скорости, превышающей скорость ведущей оси, чтобы скомпенсировать ошибку по положению, которая накопилась в процессе разгона ведомой оси до скорости ведущей. Эта компенсирующая коррекция расстояния имеет трапецеидальный профиль скорости, определяемый ***Follower Ramp Distance Make-Up Time (Длительность участка линейного компенсирующего изменения скорости следящего устройства)*** и ***Ramp Makeup Acceleration (Линейное Компенсирующее Изменение Скорости)*** в начале коррекции. Этот режим используется, если требуется засинхронизировать ведомую ось по позиции и по скорости с ведущей осью в то время, когда включен режим слежения.

Если ***Длительность участка линейного компенсирующего изменения скорости следящего устройства*** слишком мала, то профиль скорости становится треугольным. Если в процессе коррекции расстояния скорость превышает 80% границы скорости, то автоматически рассчитываемая скорость задерживается на 80% сконфигурированной границы скорости. В обоих случаях появляется предупреждающее сообщение, а фактическая длительность участка компенсации оказывается превышающей запрограммированную; однако расстояние в обоих случаях корректируется правильно.

Установка ***Follower Ramp Distance Make-Up Time*** на ноль позволяет разогнать ось без компенсации накопленных отсчетов. В этом случае скорость ведомой оси не будет превышать скорость ведущей. Для приложений, в которых требуется только синхронизация скоростей ведомой и ведущей осей, а потеря отсчетов не имеет значения, следует установить длительность компенсации равной 0.

Типичный профиль скорости в цикле линейного изменения скорости при слежении показан ниже.



Более детальное обсуждение этой функции дано в гл. 8 "Движение следящего устройства" раздел "Управление линейным изменением скорости следящей оси".

**4.23 Пропорциональный коэффициент обратной связи по скорости.**

**Только аналоговый режим момента вращения.** Команда *Velocity Loop Proportional Gain* (Пропорциональный коэффициент обратной связи по скорости) AQ позволяет пользователю задать пропорциональный коэффициент регулятора скорости в Аналоговом режиме момента вращения. Коэффициент передачи пропорционального регулятора умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к пропорциональному члену. Правильное задание этого значения определяет, насколько хорошо регулятор скорости будет работать в системе управления. В приложении D описан метод правильной настройки этого параметра. Допустимым диапазоном для члена пропорционального коэффициента контура скорости является 0 – 32767. Значение по умолчанию: 1500.

**4.24 Интегральный коэффициент обратной связи по скорости.**

**Только аналоговый режим момента вращения.** Команда *Velocity Loop Integral Gain* (Интегральный коэффициент обратной связи по скорости) AQ позволяет пользователю задать интегральный коэффициент регулятора скорости в Аналоговом режиме момента вращения. Интегральный коэффициент умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к интегральному члену. Правильное задание этого значения определяет, насколько хорошо регулятор скорости будет работать в системе управления. В приложении D описан метод правильной настройки этого параметра. Допустимым диапазоном для члена пропорционального коэффициента контура скорости является 0 – 32767. Значение по умолчанию: 0.

**4.25 Фильтр уставок момента вращения.**

**Только аналоговый режим момента вращения.** Фильтр уставок момента вращения дает возможность пользователю активировать фильтр нижних частот для выхода регулятора скорости (Команда момента вращения). Этот фильтр обычно используется для того, чтобы предотвратить резонансные явления на станке. Имеющиеся в распоряжении установочные параметры для фильтра уставок момента вращения показаны в таблице 5-9.

Таблица 5-8. Команды фильтра момента вращения

Режим TqFilt (Фильтр момента вращения)	Установочные параметры низкочастотного фильтра заданного момента вращения
0	ВЫКЛ <sup>1</sup>
1	Узкополосный фильтр (150 Гц на уровне 3 дБ)
2	Среднеполосный фильтр (250 Гц на уровне 3 дБ)
3	Широкополосный фильтр (350 Гц на уровне 3 дБ)

Примечание 1: Установка по умолчанию

**4.26 Выбрать режим аналогового выхода.** **Только Цифровой Режим.** Для цифровых сервоустройств GE Fanuc команда *Select Analog Output Mode* (Выбрать режим аналогового выхода) дает возможность выбрать, какие аналоговые сигналы

направить на контакты аналоговых выходов (контакты 6 и 24) четырех разъемов лицевой панели DSM. Команда *Select Analog Output Mode* использует Код сигнала для задания того сигнала, который должен быть отправлен, и Код разъема для задания того разъема DSM, который должен получить сигнал. Эта команда особенно полезна при настройке сервопривода. Эта команда может быть отправлена с регистров команд на любую ось (1 – 4).

Для задания шестибайтовой команды немедленного действия %AQ используется следующая структура (описанная в табл. 5-7):

- Байт 0 содержит код команды *Select Analog Output Mode* (Выбрать режим аналогового выхода) (47h).
- Байт 1 содержит код разъема, шестнадцатиричное число.
- Байты 2-3 содержат код сигнала, десятичное число.
- Байты 4-5 не используются и должны содержать 0.

### Коды разъемов

Код разъема	Выбранный разъем	Контакты разъемов
01h	Разъем А	Контакт 6 = OUT Контакт 24 = COM (относит. 0 В) См. схемы подключений В/В в гл. 3 для подключений клеммной колодки.
02h	Разъем В	
03h	Разъем С	
04h	Разъем D	

### Коды сигналов

Обратите внимание в приведенной ниже таблице, что лишь некоторые из сигналов имеют выход по умолчанию.

Код сигнала	Описание сигнала	Выход по умолчанию на:
00 десятичный*	Данные %AQ <i>Force Analog Output</i> (Включить аналоговый выход) *	Разъем С или D
10 десятичный	Команда момента вращения сервооси 1	Нет
15 десятичный	Текущая скорость сервооси 1	Разъем А
20 десятичный	Команда момента вращения сервооси 2	Нет
25 десятичный	Текущая скорость сервооси 2	Разъем В

\* Направление не может быть изменено. Этот код сигнала может быть использован только для того, чтобы восстановить сигнал обратно на его значение по умолчанию.

**Примечание:** На осях с цифровым управлением аналоговый выход для управления со стороны пользователя отсутствует. Выдача на цифровые оси команд *Force Analog Output* (Включить аналоговый выход) или *Select Analog Output* (Выбрать аналоговый выход) не окажет влияние на эти аналоговые выходы.

Команда *Select Analog Output Mode* (Выбрать режим аналогового выхода) имеет три основных направления использования:

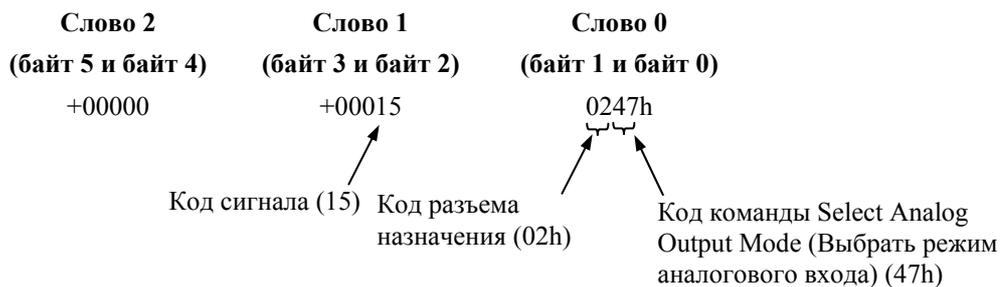
- (1) Перенаправление *Servo Axis 1 Actual Velocity* (Текущая скорость сервооси 1) или *Servo Axis 2 Actual Velocity* с их выхода по умолчанию на другой выход. Сигнал данных %AQ *Force Analog Output* (Включить аналоговый выход) не может быть перенаправлен на другой разъем; однако он может быть заменен на своем выходном

разъеме по умолчанию (С или D) другим сигналом, который может быть направлен туда командой *Select Analog Output Mode* (Выбрать режим аналогового выхода).

- (2) Направление одного из двух сигналов, не имеющих выхода по умолчанию, *Servo Axis 1 Torque Command* (Команда момента вращения сервооси 1) и *Servo Axis 2 Torque Command*, на один из выходов, замещая, таким образом, предыдущий сигнал на этом выходе. Это показано в примере 2 ниже.
- (3) Восстановление сигналов с выходами по умолчанию, которые были заменены перенаправленными сигналами. В примере 2 сигнал *%AQ Force Analog Output* (Включить аналоговый выход), который, как правило, по умолчанию находится на разъеме D, заменен сигналом *Servo Axis 1 Torque Command* (Команда момента вращения сервооси 1), который был направлен на разъем D командой *Select Analog Output Mode* (Выбрать режим аналогового выхода). В примере 3 сигнал *%AQ Force Analog Output* восстановлен на разъеме D при помощи команды *Select Analog Output Mode*.

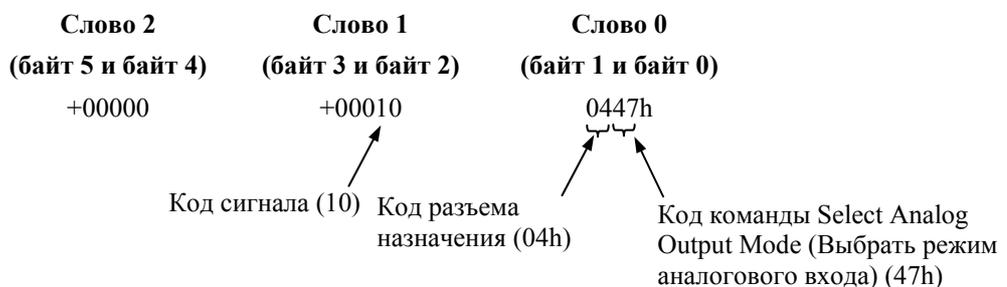
### Пример 1:

В данном примере сигнал *Servo Axis 1 Actual Velocity* (Текущая скорость сервооси 1) (код сигнала = 15) перенаправлен на разъем В (код разъема= 02h) с его выхода по умолчанию на разъеме А, заменив, тем самым, любые предыдущие сигналы на разъеме В. Это сопровождается помещением следующих данных в словах команды немедленного действия %AQ:



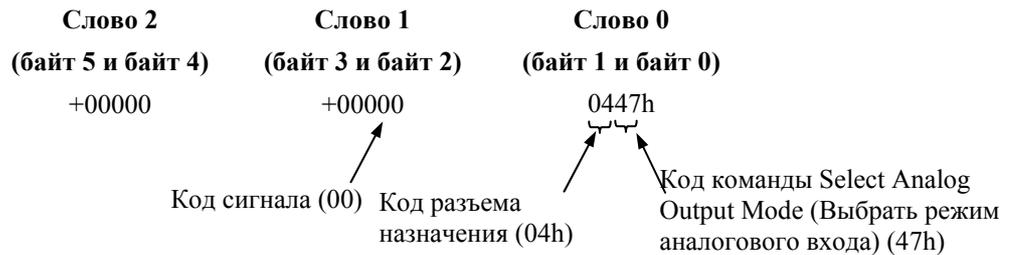
### Пример 2:

В данном примере сигнал *Servo Axis 1 Torque Command* (Команда момента вращения сервооси 1) (код сигнала =10) выбран в качестве аналогового выхода на разъеме D (код разъема=04h), заменив, тем самым, любые предыдущие сигналы на разъеме D. Это сопровождается помещением следующих данных в словах команды немедленного действия %AQ:



### Пример 3:

В примере 2 сигнал по умолчанию *%AQ Force Analog Output* (Включить аналоговый выход) был заменен на аналоговом выходе разъема D сигналом *Servo Axis 1 Torque Command* (Команда момента вращения сервооси 1). Для восстановления сигнала *%AQ Force Analog Output* (код сигнала=00) на разъеме D (код разъема = 04h) следует поместить следующие данные в словах команды немедленного действия %AQ:



**4.27 Удалить новую полученную конфигурацию.** Команда *Clear New Configuration Received* (Удалить новую полученную конфигурацию) очищает бит *New Configuration Received* (Получена новая конфигурация) %I. Будучи очищенным, бит *Configuration Complete* (Конфигурация закончена) устанавливается только в том случае, если перезапускается ПЛК или реконфигурируется модуль. ПЛК может отслеживать состояние бита, чтобы определить, не следует ли послать другие команды %AQ, такие как *In Position Zone* (Зона прихода в позицию) или *Jog Acceleration* (Ускорение толчка). Это может оказаться необходимым только в том случае, если %AQ команды были использованы для перезаписи данных конфигурации модуля DSM314 при помощи соответствующего ПО ПЛК. Эта команда может быть отправлена с регистров команд на любую ось (1 – 4).

**4.28 Загрузить параметр немедленно.** Команда *Load Parameter Immediate* (Загрузить параметр немедленно) выполняется ПЛК, чтобы немедленно изменить значение параметра данных на модуле DSM314. Эта команда может быть отправлена с регистров команд на любую ось (1 – 4). Параметры данных используются только программами движения. Для каждого изменения параметра требуется команда. Байт 1 слова 0 содержит номер параметра (в шестнадцатиричном формате), который должен быть изменен. Модуль DSM314 содержит 256 параметров из двух слов; параметры пронумерованы от 0 до 255 (десятичный формат). Более детальную информацию см. в главе 7 раздел "Параметры (P0-P255) в DSM314".

Таблица 5-9. Количество команд *Load Parameter Immediate* (Загрузить параметр немедленно), допускаемых в одном цикле

Количество сконфигурированных осей	Количество слов %AQ	Количество команд <i>Load Parameter Immediate</i> (Загрузить параметр немедленно), допускаемых в одном цикле
2	6	2
3	9	3
4	12	4

Модуль DSM314 может оси несколькими способами без использования программы движения.

- *Find Home (Начальное позиционирование)* и *Jog Plus/Minus (Толчок Плюс/Минус)* используют биты %Q для команд движения.
- *Move at Velocity (Движение на скорости)*, *Move (Движение)*, *Force Servo Velocity (Форсировать скорость сервоустройства)*, *Force Analog Output (Включить аналоговый выход)* и *Position Increment (Приращение позиции)* используют команды %AQ прямого действия.

Если во время выполнения *Jog*, *Find Home*, *Move at Velocity*, *Move* и *Force Servo Velocity* будет дана какая-либо другая команда движения, программная или непрограммная, то возникнет ошибка. Единственным исключением является команда *Position Increment* %AQ, которая может быть выдана в любое время. Более детальную информацию о движении *Position Increment* см. в описании ниже.

Непрограммируемые движения (*Abort All Moves (Прекратить все движения)*, *Jog Plus/Minus (Толчок Плюс/Минус)*, *Move at Velocity (Движение на скорости)*, *AQ Move Cmd (Команда AQ движения)* и *Normal Stop (Нормальный останов)*) используют **Jog Acceleration (Ускорение перемещения)** и **Jog Acceleration Mode (Режим ускорения толчка)**. Команда *Feed Hold (Остановить подачу)* %Q использует программное ускорение и режим ускорения.

## Цикл начальной позиции модуля DSM314

Цикл начальной позиции может быть использован для задания правильной Текущей позиции относительно опорной точки станка. Сконфигурированное значение **Home Offset (Начальное смещение)** определяет *Home Position (Исходную позицию)* как отрезок смещения от Маркера начальной позиции.

В течение всего цикла начальной позиции бит *Enable Drive (Включить привод)* %Q должен быть в состоянии ON. Однако не требуется, чтобы бит *Find Home (Начальное позиционирование)* %Q удерживался в состоянии ON; он может быть включен на короткое время однократно. Следует отметить, что установка в состояние ON бита *Find Home (Начальное позиционирование)* %Q сразу же переводит в состояние OFF бит *Position Valid (Позиция достигнута)* %I до конца цикла начальной позиции. Бит *Abort All Moves (Прекратить все движения)* %Q останавливает цикл начальной позиции, но бит *Position Valid* не возвращается в состояние ON. Программы движения не могут выполняться, пока бит *Position Valid* не перейдет в состояние ON.

## Режим переключателя начальной позиции

Если **Find Home Mode (Скорость поиска начальной позиции)** сконфигурирован как HOMESW (HOME Switch – Датчик начальной позиции), то вход Home Switch (Датчик начальной позиции) с разъема В/В оси используется сначала, для того, чтобы ориентировочно определить опорную позицию для начальной позиции. Затем следующий маркер датчика положения, поступающий при движении в отрицательном направлении, показывает точное положение. Разомкнутый вход Переключателя начальной позиции показывает, что сервоустройство находится с положительной стороны переключателя, а замкнутый – с отрицательной. Переход от OFF к ON команды **Find Home (Начальное позиционирование) %Q** дает следующий цикл. Если не определено иное, ускорение соответствует текущему **Jog Acceleration (Ускорение передвижения)** и сконфигурированному **Jog Acceleration Mode (Режим ускорения толчка)**.

### Программа Find Home (Начальное позиционирование) для Датчика начальной позиции

Если инициация проведена с позиции на положительной стороне переключателя начальной позиции, когда этот переключатель должен быть в состоянии OPEN (РАЗОМКНУТО) (логический 0), то программа Поиска начальной позиции начинается с шага 1, указанного далее. (Все первые несколько шагов из приведенной далее программы являются необходимыми, чтобы позволить работу с различными конструкциями переключателей начальной позиции и начальных позиций). Если инициация проведена с позиции на отрицательной стороне переключателя начальной позиции, когда этот переключатель должен быть в состоянии CLOSED (ЗАМКНУТО) (логическая 1), то программа Поиска начальной позиции начинается с шага 3, указанного далее.

1. Ось перемещается в отрицательном направлении при сконфигурированной **Find Home Velocity (Скорость поиска начальной позиции)**, пока Датчик начальной позиции не замкнется.
2. Ось тормозится и останавливается.
3. Ось ускоряется в положительном направлении и двигается при сконфигурированной **Find Home Velocity (Скорость начального позиционирования)**, пока Датчик начальной позиции не разомкнется.
4. Ось тормозится и останавливается.
5. Ось ускоряется в отрицательном направлении и двигается при сконфигурированной **Final Home Velocity (Скорость подхода к начальной позиции)**, пока Датчик начальной позиции не замкнется.
6. Ось продолжает движение в отрицательном направлении при сконфигурированной **Final Home Velocity (Скорость подхода к начальной позиции)**, пока не появится импульс маркера. Маркер показывает исходную опорную позицию.
7. Ось тормозится и останавливается (в позиции за импульсом маркера).
8. Ось перемещается при текущей **Jog Velocity (Скорость передвижения)** на то количество пользовательских единиц, которое задано значением **Home Offset (Начальное смещение)**, от исходной опорной позиции. Если **Home Offset = 0**, то ось перемещается назад в позицию импульса маркера.
9. Ось тормозится и останавливается.
1. Модуль DSM314 устанавливает слова статуса **Commanded Position (Заданная позиция)** и **Actual Position (Текущая позиция) %AI** на сконфигурированное значение **Home Position (Начальная позиция)**. В завершение модуль DSM314 устанавливает

бит *Position Valid (Позиция достигнута)* %I, показывая, что цикл начальной позиции завершен.

### Пример переключателя начальной позиции

Возможны различные конструкции переключателя начальной позиции. Переключатель может быть нормально разомкнутым или нормально замкнутым; он может быть смонтирован в разных местах. Пример, приведенный в данном разделе, показывает наиболее распространенное расположение, используемое для линейных осей. На рисунке ниже датчик начальной позиции представляет собой нормально разомкнутый бесконтактный переключатель, смонтированный вблизи конца участка перемещения направляющей станка (в отрицательном направлении). Воображаемой линией, которая разделяет положительную и отрицательную стороны переключателя начальной позиции, является рабочая точка переключателя, расположенная примерно на его центральной линии. Когда направляющая станка переместится в отрицательном направлении достаточно далеко, так что кулачок переключателя вызовет его замыкание, то можно говорить, что направляющая станка перешла на "отрицательную сторону" переключателя начальной позиции. **Кулачок переключателя имеет достаточно большую длину, и пока направляющая находится на отрицательной стороне переключателя, кулачок будет держать переключатель в замкнутом состоянии.**

Обратите внимание на соотношение позиций начальной позиции, отрицательного перебега и положительного останова. Между начальной позицией и позицией отрицательного перебега имеется небольшое расстояние в отрицательном направлении. Это дает некоторое "рабочее пространство" для настройки и установки этих позиций и для программы "Начальное позиционирование", для которой требуется завершающее движение в отрицательном направлении.

Имеется также некоторое расстояние между границей позиции перебега и позицией положительного останова. Это расстояние должно быть достаточно большим, чтобы предотвратить случайное срабатывание положительного останова от направляющей станка. Правильное расстояние должно быть больше того, которое может понадобиться направляющей станка в худшем случае для остановки после ее выхода на позицию границы перебега.

В данном примере рабочая граница направляющей станка находится на положительной стороне переключателя начальной позиции. Если параметр *Home Position (Начальная позиция)* модуля DSM был установлен на 0, то это упрощает программирование команд абсолютного позиционирования, поскольку используются только положительные числа.

Часто требуется установить начальную позицию на точное расстояние от опорной точки станка. Чтобы облегчить эту настройку, датчик начальной позиции может быть сделан с вырезанными монтажными отверстиями, которые позволят выполнить грубую установку кулачка так, чтобы калибровка могла быть осуществлена за один оборот датчика положения. После этого может быть проведено точное измерение оставшегося расстояния и полученное значение введено в качестве параметра *Home Offset* модуля DSM.

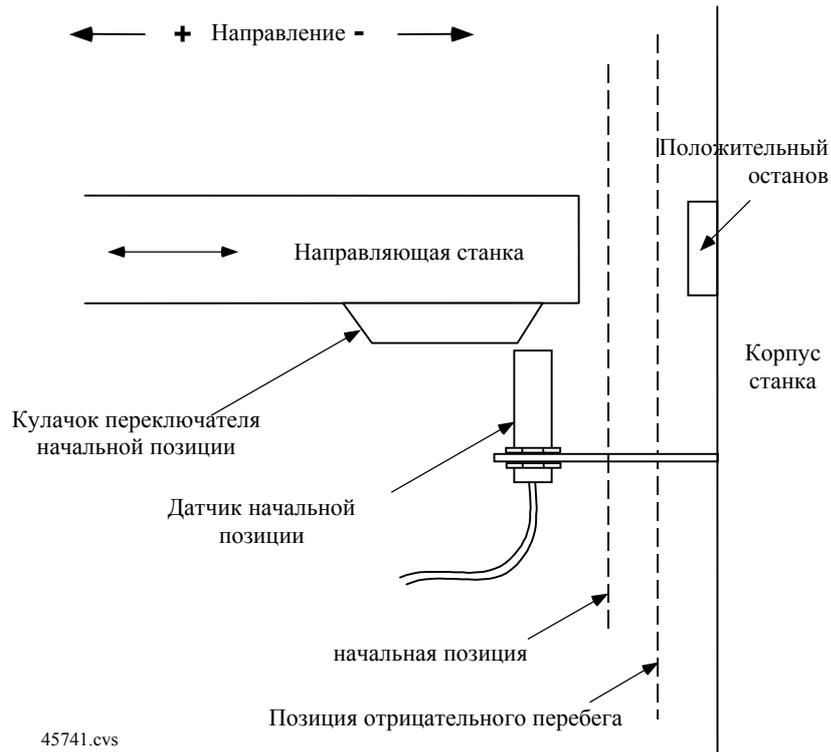


Рис. 6-1. Пример датчика начальной позиции

## Режимы Move+ (Движение+) и Move- (Движение-)

Если **Find Home Mode (Скорость поиска начальной позиции)** сконфигурирован как MOVE+ (ДВИЖЕНИЕ+) или MOVE- (ДВИЖЕНИЕ-), то первый импульс маркера датчика положения, возникающий при движении в соответствующем направлении (положительное для MOVE+, отрицательное для MOVE-) после выдачи команды "Начальное позиционирование", используется для задания точного положения. В этом режиме оператор обычно сначала переводит толчком ось в положение, близкое к начальной позиции (в пределах одного оборота датчика положения), а затем запускает команду начального позиционирования. На станке и на оси станка иногда помещают несколько меток совмещения, показывающих близость к начальной позиции, чтобы помочь оператору подойти к правильной позиции.

### Пример цикла начальной позиции Move – (минус)

Следующий рисунок показывает пример параметра Home Position (Начальная позиция), установленного на Move (Движение) – (минус). В данном примере оператор перемещает толчком ось так, чтобы подвижная метка на направляющей станка совпала с неподвижной меткой на регулировочной плите, смонтированной на корпусе станка. (Обратите внимание, что метки располагаются на положительной стороне начальной позиции, поскольку параметр Home Position установлен на Move –). Затем оператор запускает программу поиска начальной позиции, которая перемещает ось в отрицательном направлении, пока не появится импульс маркера.

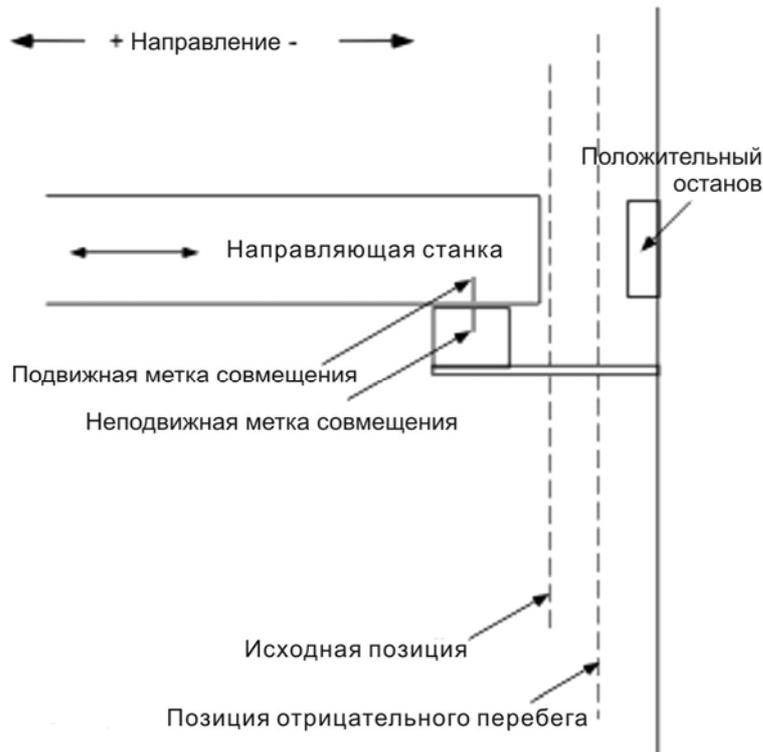


Рис. 6-2. Пример начальной позиции Move (Движение) – (минус)

### Программа Find Home (Начальное позиционирование) для Move (Движение) + или Move (Движение) –

Если запущена команда поиска начальной позиции (бит *Find Home (Начальное позиционирование)* %Q переключается с OFF на ON), то происходит следующая последовательность действий:

1. Ось ускоряется при значении *Jog Acceleration (Ускорение перемещения)* и движется при сконфигурированном значении *Final Home Velocity (Скорость подхода к начальной позиции)* (в положительном направлении для MOVE+, в отрицательном – для MOVE–), пока не появится импульс маркера. Этот импульс маркера показывает исходную опорную позицию.
2. Ось останавливается (в позиции после импульса маркера), используя сконфигурированное значение *Jog Acceleration (Ускорение перемещения)* и сконфигурированный режим *Jog Acceleration Mode (Режим ускорения толчка)*.
3. Ось перемещается при сконфигурированном значении *Jog Velocity (Скорость толчка)* и при сконфигурированных значениях *Jog Acceleration (Ускорение перемещения)* и режиме *Jog Acceleration Mode (Режим ускорения толчка)* на то количество пользовательских единиц, которое задано значением (*Начальное смещение*), от исходной опорной позиции. Если *Home Offset = 0*, то ось перемещается назад в позицию импульса маркера.
4. Ось останавливается при сконфигурированных значениях *Jog Acceleration (Ускорение перемещения)* и режиме *Jog Acceleration Mode (Режим ускорения толчка)*.
5. Модуль DSM314 устанавливает слова статуса *Commanded Position (Заданная позиция)* и *Actual Position (Текущая позиция)* %AI на сконфигурированное значение *Home Position (Начальная позиция)*; в завершение модуль DSM314 устанавливает бит *Position Valid (Позиция достигнута)* %I, показывая, что цикл начальной позиции завершен.

## Толчковое передвижение при помощи DSM314

*Скорость передвижения*, *Ускорение передвижения* и *Режим ускорения толчка* являются параметрами конфигурации в модуле DSM314. Эти значения используются всякий раз, когда бит *Jog Plus (Толчок плюс)* или *Jog Minus (Толчок минус)* %Q переводится в состояние ON (ВКЛ.). Обратите внимание, что если оба бита находятся в состоянии ON, то движение не происходит. *Ускорение передвижения* и *Режим ускорения толчка* используется также в *Find Home (Начальное позиционирование)*, *Move at Velocity (Движение на скорости)*, *Abort All Moves (Прекратить все движения)* и *Normal Stop (Нормальный останов)*. Программные движения используют *Скорость передвижения* и *Ускорение передвижения* по умолчанию.

Команда *Jog Plus/Minus (Толчок плюс/минус)* %Q может быть выполнена, когда нет команд на какое-либо другое движение или если программное движение временно остановлено командой *Feed Hold (Остановить подачу)* %Q. Для толчка не требуется нахождение бита *Enable Drive* %Q в состоянии ON, но он может быть в этом состоянии. Включение бита *Jog Plus/Minus* %Q автоматически приводит к замыканию Реле включения и включает бит *Drive Enabled (Привод включен)* %I. Если концевой переключатель перебега находится в состоянии OFF, то биты *Jog Plus/Minus (Толчок плюс/минус)* и *Clear Error (Сбросить ошибку)* %Q могут быть включены одновременно, чтобы выполнить отход от разомкнутого концевого переключателя. При этом команда *Jog Plus (Толчок плюс)* %Q не будет работать, пока разомкнут концевой переключатель положительного перемещения, а команда *Jog Minus (Толчок минус)* не будет работать, пока разомкнут концевой переключатель отрицательного перемещения. Установка бита *Jog* %Q в положение OFF вызывает торможение и остановку оси. Если бит *Jog* %Q выключается на короткое время, хотя бы на один цикл ПЛК, то ось замедляется до остановки, а затем ускоряется и продолжает движение толчка.

## Команда Move at Velocity (Движение на скорости)

Команда *Move at Velocity (Движение на скорости)* %AQ создается при помещении значения 22h в первом слове данных %AQ, назначенных оси. Второе и третье слова вместе представляют 32-битовую скорость со знаком. Обратите внимание, что третье слово является старшим словом скорости. После того как команда выдана, данные %AQ могут быть удалены посылкой команды *NULL* или изменены нужным образом. *Move at Velocity (Движение на скорости)* не будет работать до тех пор, пока не будет включен сервопривод (должна быть выдана команда *Enable Drive (Включить привод)* %Q и установлен бит статуса *Drive Enabled (Привод включен)* %I).

Список команд прямого действия %AQ показывает слова в обратном порядке, чтобы облегчить понимание. Например, чтобы выдать команду скорости 512 пользовательских единиц в секунду на модуль DSM314, сконфигурированный с данными, начинающимися на %AQ1, должны быть использованы следующие значения: 0022h (34 в десятичном формате) в %AQ1, 0200h (512 в десятичном формате) в %AQ2 и 0 в %AQ3. Когда модуль DSM314 получает эти значения и *Drive Enabled %I* находится в состоянии ON, *Abort All Moves %Q* - в состоянии OFF, а другая команда движения отсутствует, то модуль начинает перемещать ось со скоростью 512 пользовательских единиц в секунду в положительном направлении, используя текущие *Jog Acceleration (Ускорение перемещения)* и *Acceleration Mode (Режим ускорения)*.

Бит *Drive Enabled %I* должен быть в состоянии ON до того, как DSM314 получит команду прямого действия; в противном случае будет выдано сообщение об ошибке. Также, если команда *Move at Velocity (Движение на скорости)* уже имеется в данных %AQ, то значение скорости должно быть изменено, пока бит *Drive Enabled (Привод включен)* находится в состоянии ON для DSM314, чтобы принять его. Модуль DSM314

определяет команду *Move at Velocity* (*Движение на скорости*), когда происходит изменение значений %AQ.

Когда модуль DSM314 выполняет команду *Move at Velocity*, он игнорирует программные границы перемещения (*Pos EOT* и *Neg EOT*). Аппаратные границы перебега должны быть в состоянии ON, если они включены.

Команда *Move at Velocity* (*Движение на скорости*) может быть остановлена без выдачи ошибки двумя путями: выдача команды *Move at Velocity* с нулевой скоростью или переводом бита *Abort All Moves* (*Прекратить все движения*) %Q в состояние ON на хотя бы один цикл ПЛК.

## Команда Force Servo Velocity (Форсировка скорость сервоустройства) (ЦИФРОВЫЕ сервоустройства; Аналоговый режим момента вращения)

Команда *Force Servo Velocity* (*Форсировка скорость сервоустройства*) обходит контур позиционирования и включает заданную скорость в об./мин. на цифровом сервоустройстве или аналоговом интерфейсе момента вращения с целью подстройки. Управление ускорением не производится, и изменения скорости действуют немедленно. Значение +4095 команды *Force Servo Velocity* приведет к + 4 095 об./мин. двигателя, а значение -4095 – к -4 095 об./мин. Контур управления цифровым сервоустройством могут ограничить текущую скорость двигателя меньшим значением.

### ПРЕДОСТЕРЕЖЕНИЕ!

**Следует быть внимательным, чтобы не использовать серводвигатель при нагрузках, превышающих его номинал.**

Чтобы команда *Force Servo Velocity* могла быть выполнена, бит *Enable Drive* %Q должен быть установлен при отсутствии другой заданной команды движения. Для правильной работы эта команда должна быть в течение длительного времени в данных %AQ. Если для данной оси действует команда *Force Servo Velocity*, то любая другая %AQ команда прямого действия для этой оси удалит команду *Force Servo Velocity* и остановит сервопривод. Одноразовая команда *Force Servo Velocity* будет действовать, только в течение того цикла, в котором она появилась.

Более детальную информацию по этой команде см. в главе 5 "Взаимодействие модуля Motion Mate DSM314 с ПЛК".

**Примечание:** Команда *Force Analog Output* (Включить аналоговый выход), описанная ниже, используется для аналоговых сервоустройств с интерфейсом команд скорости.

## Команда Force Analog Output (Форсировка аналогового выхода) (АНАЛОГОВЫЕ сервоустройства с интерфейсом скорости)

В режиме Аналогового интерфейса скорости команда прямого действия *Force Analog Output* (*Форсировка аналогового выхода*) %AQ выводит аналоговый выход на разъемы

A, B, C или D лицевой панели DSM. Значение команды *Force Analog Output* +32000 создаст +10.00 В постоянного тока, а значение -32000 создаст -10.00 В пост. тока.

*Force Analog Output* действует только тогда, когда данные %AQ являются активными. Если для данной оси действует команда *Force Analog Output*, то любая другая %AQ команда прямого действия для этой оси удалит команду *Force Analog Output* и выключит соответствующий аналоговый выход.

Более детальную информацию по этой команде см. в главе 5 "Взаимодействие модуля Motion Mate DSM314 с ПЛК".

## Команды Position Increment (Приращение позиции)

Чтобы делать небольшие исправления позиции оси и отслеживания модулем DSM314, могут быть использованы команды *Position Increment (Приращение позиции)* %AQ, позволяющие выполнить смещение *Текущей позиции* на определенное число пользовательских единиц. Если бит *Drive Enabled (Привод включен)* %I находится в состоянии ON, то ось немедленно переместится на величину приращения. Если используется приращение позиции без ее обновления (%AQ команда 21h), то слово статуса *Actual Position (Текущая позиция)* %AI, передаваемое модулем DSM314, останется без изменения. Если используется *Position Increment With Position Update (Приращение позиции с обновлением позиции)* (%AQ команда 25h), то слова статуса *Actual Position (Текущая позиция)* и *Commanded Position (Заданная позиция)* %AI, передаваемые модулем DSM314, будут изменены на значение приращения. *Position Increment (Приращение позиции)* может быть использовано в любое время, хотя одновременное использование с командой *Force Servo Velocity (Форсировать скорость сервоустройства)* невозможно, поскольку команда *Force Servo Velocity* должна оставаться в области данных команд %AQ или сервоустройство остановится.

## Дополнительная информация

Дополнительная информация по использованию непрограммного движения:

- Если бит *Abort All Moves (Прекратить все движения)* %Q находится в состоянии ON, то это исключает любое не программное движение с самого начала.
- Переход бита *Abort All Moves (Прекратить все движения)* %Q остановит любое совершающееся не программное движение при текущем значении *Jog Acceleration (Ускорение перемещения)*.
- Команда *Set Position (Установить позицию)* %AQ, выданная при совершающемся не программном движении, создаст ошибку статуса.
- Перевод бита *Enable Drive (Включить привод)* %Q в состояние OFF при выполнении цикла начальной позиции или при выполнении команды *Move at Velocity (Движение на скорости)* %AQ вызовет ошибку останова.
- Бит *Feed Hold (Остановить подачу)* %Q не влияет на непрограммируемое движение.
- Команда *Rate Override (Подстановка скорости)* %AQ не влияет на непрограммируемое движение.
- Изменение *Jog Velocity (Скорость передвижения)* или *Jog Acceleration (Ускорение перемещения)* не влияет на совершающееся движение.

Любая программа движения состоит из группы запрограммированных пользователем командных операторов движения, которая хранится и выполняется в модуле DSM314. Модуль DSM314 выполняет команды программы движения последовательно блок за блоком после запуска программы на исполнение. Программа движения выполняется независимо от ПЛК. ПЛК запускает саму программу движения в DSM314 и может взаимодействовать с нею в ходе исполнения через параметры и определенные команды. Кроме того, чтобы задерживать или изменять ход выполнения программы движения, могут быть использованы внешние входы (CTL-биты), подключенные непосредственно к лицевой панели DSM314 или управляемые программой Local Logic. В ходе выполнения программы ПЛК получает от DSM314 информацию о статусе (такую как позиция, скорость и номер командного блока). Программы движения 1 – 10 и подпрограммы 1 – 40 создаются с использованием ПО ПЛК; они сохраняются с установочными параметрами конфигурации модуля в DSM314 через базовую плату ПЛК.

Дополнительную информацию см. в интерактивной справке по программному обеспечению или в соответствующих руководствах:

*Руководство пользователя по программному обеспечению VersaPro™, GFK-1670*

*Основы использования пакета SIMPLICITY® Machine Edition Logic Developer-PLC, GFK-1918*

### **Программы и подпрограммы движения одной оси (одноосные)**

Одноосная программа содержит командные операторы только для одной оси. Программируемая ось определяется в первой строке программы, например: PROGRAM 1 AXIS1. Число одноосных программ, с которыми может работать модуль DSM314, составляет 4. Эти программы могут действовать независимо или одновременно. Например, программа движения 1 может быть написана для оси 1, а программа 2 – для оси 2. Каждая ось может быть привязана к начальной позиции, и программа движения для каждой оси может выполняться независимо без учета состояния другой оси. Как альтернатива, программы 1 и 2 могут быть запущены одновременно (битами запуска программы %Q) в одном и том же цикле ПЛК.

Программы движения модуля DSM314 поддерживают функцию "подпрограмм", которая позволяет включать все имеющиеся команды программ движения, в т.ч. и команду CALL. Команда SYNC Block зарезервирована для многоосных программ (оси 1 и 2) и подпрограмм. Подпрограмма "вложение", использующая операторы CALL, поддерживает до 8 уровней вложения. Одноосные подпрограммы, так же как и программы движения, содержат команды только для одной оси. Различие лишь в том, что в одноосной подпрограмме не задан номер оси. Одноосная программа движения может CALL (ВЫЗВАТЬ) любую одноосную подпрограмму, хранящуюся в памяти модуля. Например, одноосная программа движения 1, работающая с осью 1, может содержать оператор CALL для одноосной подпрограммы 1. В то же время одноосная программа движения 2, работающая с осью 2, может также содержать оператор CALL для одноосной подпрограммы 1. Одноосные программы движения не могут CALL (ВЫЗВАТЬ) многоосные подпрограммы.

Программа движения и структура подпрограмм делает возможными гибкость выполнения и управления осями в модуле DSM314. Практическим ограничением является то, что каждая ось может в данный момент выполнять только одну программу. Например, если программа 1 включена для приведения в действие оси 1, то она должна завершиться или быть прервана, чтобы программа 2 могла начать работать с осью 1.

### **Программы и подпрограммы движения нескольких осей (многоосные)**

Термин "многоосная" задается в операторе определения (в первой строке) программы или подпрограммы, например: PROGRAM 2 MULTI-AXIS или SUBROUTINE 7 MULTI-AXIS. **Ось 1 и ось 2 являются единственными двумя номерами осей, которые разрешены в многоосной программе или подпрограмме.** Чтобы программа была выполнена, обе оси должны быть приведены в начальную позицию; они должны также удовлетворять некоторым предварительным условиям (см. раздел "Предварительные условия для программируемого движения" в данной главе). Многоосная программа движения может *CALL (ВЫЗВАТЬ)* только многоосную подпрограмму. Одна команда программы движения *SYNC Block* доступна только в многоосной программе или подпрограмме движения. Подпрограмма "вложение" имеет те же ограничения, что и одноосная программа движения. В многоосной программе имеются две категории движений: движения 1-й оси и движения 2-й оси.

**Движения 1-й оси:** Если запрограммированы два последовательных движения 1-й оси, то выполнение второго движения начинается через 2 миллисекунды после окончания первого.

**Движения 2-й оси:** Движение 2-й оси программируется тремя последовательными блоками. Первый из этих трех блоков должен содержать команду *SYNC Block*. Следующие два блока содержат команды движения: один - для оси 1, другой – для оси 2. Когда команда *SYNC Block* выполнена, два движения запускаются "вместе" (в пределах 2 миллисекунд). Следует отметить, что синхронизирован только запуск этих движений.

Более детальная информация о программировании многоосного движения, о структуре блоков программы, об управлении потоками (JUMP) и о команде *SYNC Block* дается в этой главе далее.

## Типы команд программ движения

Команды программ движения делятся на четыре группы:

### Команды типа 1

- CALL (подпрограмма)
- JUMP

### Команды типа 2

- Номер блока
- SYNC (Синхронизация блока)
- LOAD (ЗАГРУЗИТЬ) (Параметр)
- ACCEL (Ускорение)
- VELOC (Скорость)

### Команды типа 3

- PMOVE (Движение позиционирования)
- CMOVE (Непрерывное движение)
- DWELL (ЗАДЕРЖКА)
- WAIT (ОЖИДАНИЕ)

### Команды определения (задания) программы / подпрограммы

- PROGRAM
- ENDPROG
- SUBROUTINE
- ENDSUB

*Команды типа 1* могут пере направлять выполнение программы, но не могут прямо воздействовать на позиционирование.

- Call (подпрограмма) выполняет подпрограмму, прежде чем перейти к выполнению другой команды.
- Jumps (Переходы) могут быть условными или безусловными. Безусловный переход всегда пере направляет выполнение в определенную точку программы. Условный переход связан с проверкой STL-бита. Если STL-бит находится в состоянии ON, то команда условного перехода направляет выполнение в определенное место программы. Если STL-бит находится в состоянии OFF, то переход игнорируется.

*Команды типа 2* также не влияют на позицию.

- Номера блоков обеспечивают идентификацию или метку для следующей далее команды типа 3. Номера блоков требуются для команд JUMP (ПЕРЕХОД); в противном случае они являются необязательными. Если блок программы не имеет номера блока, то действует номер предыдущего блока, если такой номер имеется.
- Команда SYNC (синхронизировать блок) является командой синхронизации двух осей (она может либо задержать движение по одной оси, либо обойтись без этого).
- Команда Load Parameter (Загрузить параметр) позволяет пользователю загрузить какое-то значение в регистр параметров.
- Команды Скорости (VELOC) и Ускорения (ACCEL) задают значения скорости и ускорения для команды типа 3 MOVE или для команд, следующих далее. Команды Скорости и Ускорения действуют до тех пор, пока они не изменены.

*Команды типа 3* запускают или останавливают движение и, таким образом, влияют на управление позиционированием.

- Команды движения позиционирования (PMOVE) и непрерывного движения (CMOVE) вызывают движение.
- Команды Dwell, Wait и End of Program останавливают движение.

### **Обработка программных блоков и команд движения**

"Программный блок" содержит одну (и только одну) команду типа 3 и определяется именно так; он может содержать любое количество любой комбинации предшествующих команд типа 1 и 2.

Номер блока имеет два основных назначения: (1) он определяет идентификацию точки перехода (метку), и (2) он идентифицирует раздел программы, который в данный момент выполняется при помощи слов статуса *Block Number (Номер блока) %AI* для каждой оси. Команды типа 2 являются необязательными; блок программы может содержать единственную команду типа 3. Команды типа 2 и условные переходы не оказывают влияния до тех пор, пока модуль DSM не выполнит следующую команду типа 3.

В то время как модуль DSM314 выполняет блок программы, следующий блок программы обрабатывается в буферной области команд. Такая буферизация минимизирует время перехода между блоками. Поэтому параметры, используемые в движении, должны быть загружены до того, как команда движения, которая была запрограммирована двумя блоками ранее, завершит выполнение. Другими словами, чтобы минимизировать время перехода между блоками, предварительная обработка нового блока должна происходить во время выполнения предыдущего. Параметры блока программы должны быть загружены до того, как начнется выполнение предыдущего блока.

Когда модуль DSM314 выполняет многоосную программу, то команды программ просматриваются независимо по каждой оси и на выполнение направляются только те данные, которые предназначены для той оси. Обратите внимание, что некоторые многоосные команды

не определяют ось (Block number, Jump, Call и End) и потому могут быть применены к обеим осям.

Многоосные программы могут содержать команды *SYNC*, синхронизирующие оси в назначенной точке. Когда первая ось достигнет блока *SYNC* (блок, содержащий команду *SYNC*), то следующий блок не будет выполняться до тех пор, пока другая ось так же не достигнет блока *SYNC*. Пример этого см. далее в данной главе в примере 18 "Многоосное программирование".

## Предварительные условия для программируемого движения

Для инициации программы движения должны быть выполнены следующие условия (для многоосной программы эти условия должны быть выполнены для обеих осей):

- Бит Enable Drive (Включить привод) %Q должен быть в состоянии ON.
- Бит Drive Enabled (Привод включен) %I должен быть в состоянии ON.
- Бит Position Valid (Позиция достигнута) %I должен быть в состоянии ON.
- Бит Moving (Движение) %I должен быть в состоянии OFF.
- Бит Program Active (Программа активна) %I должен быть в состоянии OFF.
- Бит Abort All Moves (Прекратить все движения) %Q должен быть в состоянии OFF.
- Положение оси должно быть в сконфигурированных границах конца передвижения (Верхняя и Нижняя программные границы конца передвижения), кроме случая, когда режим программного конца передвижения отключен.
- Входы концевого выключателя перебега, если они включены, должны быть в состоянии ON (верхний уровень входа 24 В).
- Команда Force Digital Servo Velocity (Форсировка скорость цифрового сервоустройства) %AQ не должна быть в активном состоянии.
- Предназначенная для выполнения программа должна быть полноценной программой, хранимой в DSM314.

## Условия, приводящие к остановке программы движения

Выполнение программы движения немедленно прекращается, если имеет место любое из приведенных ниже условий:

- Бит *Abort All Moves (Прекратить все движения)* %Q переходит в состояние ON.
- Бит *Enable Drive (Включить привод)* %Q переходит в состояние OFF.
- **Выключатель выхода за границу** переходит в состоянии OFF, если этот Выключатель выхода за границу ВКЛЮЧЕН при конфигурации.
- Следующее программируемое движение, PMOVE или CMOVE, доходит до программной границы конца передвижения (если режим *Software End of Travel (Программный ограничитель движения)* не выключен при конфигурации).
- Появляется ошибка метода ответной реакции на Stop Normal (Нормальный останов) или Stop Fast (Быстрый останов). См. приложение А "Сообщения об ошибках".

## Основы программы движения

### Количество программ, подпрограмм и операторов

Модуль DSM314 поддерживает 10 программ движения, 40 подпрограмм и, в общей сложности, 1000 операторов программы движения.

### Формат

- Программы движения и подпрограммы написаны с использованием текста ASCII.
- Допускается только один оператор движения в строке, и оператор движения не может занимать более одной строки. Обычные комментарии могут занимать несколько строк.
- Для улучшения читаемости и для отделения разделов можно использовать пробелы и пустые строки.
- Редактор программ движения Motion Editor не различает регистров букв.
- Все программы движения и подпрограммы должны быть в одном файле.

### Одно- и многоосные программы и подпрограммы

Одноосная программа должна быть в состоянии работать на любой одной оси, заданной в операторе определения программы. Поэтому команды движения в одноосных программах и подпрограммах не задают какую-либо ось. В то же время ось, заданная в операторе ПРОГРАММЫ, используется для всех команд движения в программе. Многоосные программы и подпрограммы могут вызывать только многоосные подпрограммы. Аналогично, одноосные программы и подпрограммы могут вызывать только одноосные подпрограммы.

### Операторы определения программы и подпрограммы

Motion Editor требует операторов определения "Программа" и "Подпрограмма", которые задают номер программы / подпрограммы и конфигурацию оси (PROGRAM 1 AXIS2 или SUBROUTINE 2 MULTI-AXIS). Эти операторы помещаются в первой строке программы или подпрограммы. Программы заканчиваются оператором ENDPROG; подпрограммы заканчиваются оператором ENDSUB. Эти операторы выступают в качестве разделителей между программами и подпрограммами; они идентифицируют номера программ и подпрограмм и показывают тип программы (одноосная или многоосная).

### Номера блоков и блоки синхронизации

За номером блока следует двоеточие (например, 1:). Блоки синхронизации идентифицируются строкой с номером блока с последующей командой SYNC (например, 2: SYNC). Номера блоков могут стоять отдельно на строке или сопровождаться последующей командой движения на этой строке.

## Команды и синтаксис языка движения

### Пустые пространства

Пустые пространства не имеют значения и игнорируются, за исключением тех случаев, когда они используются в качестве разделителей. Например, в "CMOVE AXIS1 50000,ABS,S-CURVE" пробел требуется в качестве разделителя между CMOVE и AXIS1, но он не требуется далее в 50000,ABS, поскольку здесь параметры разделяет запятая. Пробелы, пустые строки и табуляторы рассматриваются как пустые пространства.

### Числовые константы

Числовые константы ограничены 32-битовыми целыми числами со знаком или без него в зависимости от используемого контекста. Все указанные далее команды ограничивают этот диапазон. Числовые константы могут быть введены как десятичные, шестнадцатеричные или двоичные значения. Шестнадцатеричные и двоичные константы идентифицируются префиксом 16# и 2#, соответственно (не используйте пробел между префиксом и самим числом). У шестнадцатеричных и двоичных констант не может быть префикса с отрицательным знаком. Поэтому отрицательные значения следует вводить во второй дополнительной форме. Числовые константы могут содержать одиночные подчеркнутые символы (например, 5\_000\_000) между цифрами, чтобы улучшить читаемости больших чисел или чтобы представить подразумеваемые десятичные точки в числах с фиксированной точкой.

### Комментарии

Пара символов "(" вводит обычный комментарий, который заканчивается парой символов "\*")". Эти комментарии могут появиться везде, где может быть поставлен пробел, например, внутри оператора программы движения или за ним, отдельно в строке или занять несколько строк. Эти комментарии не вкладываются. Пара символов "/" вводит комментарий в одиночной строке. Весь текст, расположенный за // до конца строки, игнорируется в Motion Editor. Однако при использовании // не разрешается применять разрыв строки (т.е. Return) для перехода к следующей строке; в противном случае будет выдано сообщение об ошибке. Если требуется сделать длинный комментарий, читаемый в окне Motion Editor без необходимости скроллинга вправо, то можно использовать символы (\*and \*) (требуемые для многострочных комментариев) вместе с переводом строки Return (создаваемым нажатием клавиши Enter); это вызывает переход текста на следующую строку.

### Ключевые слова программы движения

Следующие слова имеют специальное значение в языке программирования движения.

ABS	AXIS3	ENDSUB	MULTI-AXIS	SUB
ABSOLUTE	AXIS4	ENDS	PMOVE	SYNC
ACCEL	CALL	INCR	PROGRAM	VELOC
ACC	CMOVE	INCREMENTAL	PROG	VEL
ACCELERATION	DWELL	JUMP	S-CURVE	VELOCITY
AXIS1	ENDP	LINEAR	SINGLE-AXIS	WAIT
AXIS2	ENDPROG	LOAD	SUBROUTINE	

## Переменные

Программы движения поддерживают ограниченный набор предварительно определенных переменных: регистры данных параметров и STL-биты. В таблице ниже X представляет собой десятичное число в указанном диапазоне. Величина X интерпретируется на базе его числового значения. Поэтому ссылка на данную переменную может быть сделана различными способами. Например, как P1, так и P001 ссылаются на регистр 1 данных параметра; обе ссылки принимаются в Motion Editor.

Переменная	Ограничения
Px	$0 \leq x \leq 255$
STLx	$01 \leq x \leq 32$

## Разделители

Разделители используются для разделения элементов или для добавления к элементам с целью указания на то, что они выполняют особую функцию.

Разделитель	Функция
,	Разделяет параметры команды
:	Идентифицирует константу как номер блока

## Команды программы движения

Этот раздел описывает команды движения. Большинство команд движения имеет две формы: одноосная и многоосная. Многоосная форма используется в многоосных программах и подпрограммах; она требует, чтобы ось была определена как параметр в определенных командах (например: VELOC AXIS1 5000). В одноосных программах номер оси задается в заголовке программы (например: PROGRAM 2 AXIS1). Номер оси не нужно задавать внутри программы.

Некоторые из командных ключевых слов имеют альтернативные имена. Эти альтернативные ключевые слова команд являются функционально эквивалентными основным ключевым словам. Использование альтернативных имен не является обязательным, будучи делом личных предпочтений.

Элементы, появляющиеся в угловых скобках (“<”, “>”), представляют собой классы элементов, и описываются более детально. Элементы, появляющиеся в квадратных скобках (“[”, “]”), являются не обязательными. Элементы, появляющиеся в фигурных скобках (“{”, “}”), требуются для многоосных программ и подпрограмм, однако не допускаются в одноосных.

Общим форматом команд языка движения является следующее: КЛЮЧЕВОЕ СЛОВО {ось} <параметр [, параметр]>. Если ось задана, она следует непосредственно за ключевым словом команды. Параметр(ы) команды следует(ют) за осью, если она задана. Если имеется несколько параметров, они разделяются запятыми.

**Примечание:** Модуль DSM314 не поддерживает команду NULL (НУЛЬ) или Program Zero (Программный ноль).

## ACCEL

Оператор ACCEL задает ускорение оси для последующих движений и действует в данной программе до тех пор, пока не будет изменен. Если оператор ACCEL в программе не задан, то ускорение будет равно значению текущего *Jog Acceleration* (*Ускорение передвижения*). Движения, запрограммированные до первого оператора ACCEL, будут использовать значение ускорения текущего *Jog Acceleration* (*Ускорение передвижения*). Движения, запрограммированные после оператора ACCEL, будут использовать значение оператора ACCEL.

**Примечание:** Команды ACCEL для данной оси в программе или подпрограмме должны быть разделены оператором PMOVE, оператором CMOVE или безусловным переходом.

Синтаксис:

ACCEL {<ось>} <ускорение>

Параметр	Описание
<ось>	Номер оси может быть задан только в многоосной программе или подпрограмме. Ось может быть задана при помощи ключевых слов или констант.
<ускорение>	Ускорение задается при помощи либо константы без знака в диапазоне 1 – 1 073 741 823, либо регистра данных параметра.

**Альтернативные названия:**

ACC, ACCELERATION

**Ошибки**

1. Команды ACCEL должны быть разделены, по крайней мере, одной командой движения.
2. Заданная константа для ускорения выходит из диапазона 1 – 1 073 741 823.
3. Регистр данных параметра не в диапазоне 0 - 255.
4. Ось задана в одноосной программе.
5. Не задана ось в многоосной программе.
6. Заданная ось не поддерживает программируемое движение.

**Номер блока**

Номера блоков могут быть использованы как назначения в командах JUMP. Они могут стоять отдельно на строке или предшествовать команде.

**Синтаксис:**

<номер блока>: [<команда>]

Параметр	Описание
<номер блока>	Номер блока должен быть в диапазоне 1 – 65535.
<команда>	После номера блока на той же строке может следовать любая команда, кроме следующих: PROGRAM, SUBROUTINE, ENDPROG, ENDSUB или другой номер блока.

**Альтернативные названия:**

Нет

**Ошибки**

1. Все номера блоков и номера блоков синхронизации должны быть уникальными в рамках программы или подпрограммы.
2. Номер блока должен быть в диапазоне 1 – 65535.

**CALL**

Команда CALL вызывает подпрограмму из программы или подпрограммы.

**Синтаксис:**

CALL <обозначение подпрограммы>

Параметр	Описание
<обозначение подпрограммы>	Обозначение подпрограммы, заданное как константа (1 – 40) или как регистр данных параметра.

**Альтернативные названия:**

Нет

**Ошибки**

1. Номер подпрограммы должен быть в диапазоне от 1 до 40 или регистр данных параметра должен быть в диапазоне от 0 до 255.
2. Если вызывающей стороной является подпрограмма, то она не может вызывать сама себя (рекурсия невозможна) или другую подпрограмму, которая прямо или косвенно ссылается на нее.
3. Вызываемая подпрограмма должна быть определена в том же файле.
4. Одноосные программы и подпрограммы могут вызывать только одноосные подпрограммы. Многоосные программы и подпрограммы могут вызывать только многоосные подпрограммы.

## SMOVE

Команда SMOVE вызывает непрерывное движение с использованием заданной позиции и режима ускорения.

**Синтаксис:**

SMOVE {<ось>} <позиция>, <режим позиционирования>, <режим ускорения>

Параметр	Описание
<ось>	Ось может быть задана только в многоосной программе или подпрограмме. Ось может быть задана при помощи ключевых слов AXIS или констант.
<позиция>	Заданная позиция. Может быть константой или регистром данного параметра.
<режим позиционирования>	Задаёт инкрементное (INCR) или абсолютное (ABS) позиционирование.
<режим ускорения>	Задаёт для движения линейное (LINEAR) ускорение или S-кривую (S-CURVE).

**Альтернативные названия:**

Нет

**Ошибки**

1. Ось задана в одноосной программе.
2. Не задана ось в многоосной программе.
3. Позиция должна быть в диапазоне от -536 870 912 до -536 870 911 или регистр данных параметра должен быть в диапазоне от 0 до 255.
4. Режим позиционирования должен быть INCR или ABS.
5. Режим ускорения должен быть LINEAR или S-CURVE.
6. Заданная ось не поддерживает программируемое движение.

## DWELL

Команда DWELL вызывает прекращение движения в течение определенного интервала времени до начала обработки следующей команды. Задание значения задержки равным нулю (либо как константы, либо как значения в регистре данных параметра) вызывает отсутствие задержки, (это представляет собой отличие от свойств APM и DSM302).

Одиночная команда DWELL применима только к одной оси. Поэтому в многоосной программе следует назначить номер оси для каждой команды DWELL. Например: DWELL AXIS1 2000. Если требуется ввести паузу для обеих осей в многоосной программе, то надо использовать команду DWELL для каждой оси.

**Синтаксис:**

DWELL {<ось>} <задержка>

Параметр	Описание
<ось>	Ось может быть задана только в многоосной программе или подпрограмме. Ось может быть задана при помощи ключевых слов AXIS или констант.
<задержка>	Задержка в миллисекундах определяется как константа или как регистр данных параметра. Диапазон от 0 до 60 000 мс. Значение 0 интерпретируется как нулевая команда.

**Альтернативные названия:**

Нет

**Ошибки**

1. Ось задана в одноосной программе.
2. Не задана ось в многоосной программе.

3. Задержка должна быть в диапазоне от 0 до 60 000 или регистр данных параметра должен быть в диапазоне от 0 до 255.
4. Заданная ось не поддерживает программируемое движение.

## ENDPROG

Оператор ENDPROG заканчивает задание программы движения.

**Синтаксис:**

ENDPROG

**Альтернативные названия:**

ENDP

## ENDSUB

Оператор ENDSUB заканчивает задание подпрограммы движения.

**Синтаксис:**

ENDSUB

**Альтернативные названия:**

ENDS

## JUMP

Выполняется переход на номер блока или на блок синхронизации внутри выполняемой программы или подпрограммы. Переход может быть условным на основе состояния CTL-бита или безусловным.

**Синтаксис:**

JUMP <условие>, <место назначения>

Параметр	Описание
<условие>	Условие перехода должно быть определено CTL01 – CTL32 или UNCOND
<место назначения>	Номер блока назначения или блока синхронизации

**Альтернативные названия:**

Нет

**Ошибки**

1. В условии перехода должен быть CTL в диапазоне 1 – 32 или ключевое слово UNCOND.
2. Блок назначения должен быть в диапазоне от 1 до 65535 и должен быть определен внутри той же программы или подпрограммы, что и оператор JUMP.

## LOAD

Инициализирует или изменяет регистр данных параметра при помощи 32-битового двойного дополнительного целого значения.

**Синтаксис:**

LOAD <регистр данных параметра>, <загружаемое значение>

Параметр	Описание
<регистр данных параметра>	Инициализируется регистр данных параметра. Ограничено регистрами P000 – P255.
<загружаемое значение>	32-битовая числовая константа.

**Альтернативные названия:**

Нет

**Ошибки**

1. Регистр данных параметра должен быть в диапазоне P000 – P255.
2. Загружаемое значение должно быть в диапазоне 32-битового двойного дополнительного целого значения.

**RMOVE**

Команда RMOVE вызывает движение позиционирования с использованием заданной позиции и режима ускорения.

**Синтаксис:**

RMOVE {<ось>} <позиция>, <режим позиционирования>, <режим ускорения>

Параметр	Описание
<ось>	Ось может быть задана только в многоосной программе или подпрограмме. Ось может быть задана при помощи ключевых слов AXIS или констант.
<позиция>	Заданная позиция. Может быть константой или регистром данных параметра.
<режим позиционирования>	Задаёт инкрементное (INCR) или абсолютное (ABS) позиционирование.
<режим ускорения>	Задаёт для движения линейное (LINEAR) ускорение или S-кривую (S-CURVE).

**Альтернативные названия:**

Нет

**Ошибки**

1. Ось задана в одноосной программе.
2. Не задана ось в многоосной программе.
3. Позиция должна быть в диапазоне от –536 870 912 до –536 870 911 или регистр данных параметра должен быть в диапазоне от 0 до 255.
4. Режим позиционирования должен быть INCR или ABS.
5. Режим ускорения должен быть LINEAR или S-CURVE.
6. Заданная ось не поддерживает программируемое движение.

**PROGRAM**

Оператор PROGRAM является первым оператором в программе движения. Оператор программы идентифицирует номер программы (1 – 10) и конфигурацию оси. Программы не могут быть вложенными.

Имеются два типа программ движения: одноосные программы, в которых все команды действуют на одну ось, и многоосные, которые могут содержать команды для оси 1 и оси 2. Тип программы определяется оператором PROGRAM. Одноосная программа идентифицируется заданием оси вслед за номером программы (например, PROGRAM 3 AXIS1). Многоосная программа идентифицируется словом MULTI-AXIS вслед за номером программы (например, PROGRAM 4 MULTI-AXIS).

Конфигурация программной оси используется для того, чтобы задать, должны или нет использоваться параметры оси в командах программы движения. Таким образом, вводится также ограничение для многоосных программ на вызов только многоосных подпрограмм, а для одноосных – только одноосных. Ось, определенная в одноосной программе, используется любой вызываемой подпрограммой; таким образом, номер оси не требуется задавать где-нибудь в одноосной подпрограмме.

**Синтаксис:**

PROGRAM &lt;номер программы&gt; &lt;конфигурация оси&gt;

Параметр	Описание
<номер программы>	Номер программы должен быть десятичным числом в диапазоне 1 – 10. В пределах исходного файла каждая определенная PROGRAM должна иметь <b>уникальный номер</b> .
<конфигурация оси>	Конфигурация оси должна иметь значение MULTI-AXIS для многоосных программ или обозначение оси (например, AXIS1) для одноосных программ. Оси могут быть заданы при помощи ключевых слов AXISx или констант, где x = 1 – 4.

**Альтернативные названия:**

PROG

**SUBROUTINE**

Оператор SUBROUTINE является первым оператором в подпрограмме движения.

Оператор подпрограммы идентифицирует номер подпрограммы (1 - 40) и конфигурацию оси. Определения подпрограммы нельзя вкладывать.

Имеются два типа подпрограмм движения: одноосные подпрограммы, в которых все команды действуют на одну ось, и многоосные, которые могут содержать команды для оси 1 и оси 2. Тип подпрограммы определяется оператором SUBROUTINE. Одноосные подпрограммы идентифицируются словом SINGLE-AXIS, следующим за номером подпрограммы. Многоосные подпрограммы идентифицируются словом MULTI-AXIS, следующим за номером подпрограммы.

Конфигурация оси подпрограммы используется для того, чтобы задать, должны или нет использоваться параметры оси в командах подпрограммы движения. Таким образом, вводится также ограничение для многоосных подпрограмм на вызов только многоосных подпрограмм, а для одноосных – только одноосных. Одноосная подпрограмма использует номер оси, заданный в вызывающей программе.

**Синтаксис:**

SUBROUTINE &lt;номер подпрограммы&gt; &lt;конфигурация оси&gt;

Параметр	Описание
<номер подпрограммы>	Номер подпрограммы должен быть десятичным числом в диапазоне 1 - 40. В пределах исходного файла каждая определенная подпрограмма должна иметь <b>уникальный номер</b> .
<конфигурация оси>	Конфигурация оси должна иметь значение MULTI-AXIS или SINGLE-AXIS.

**Альтернативные названия:**

SUB

## Sync Block

Блок синхронизации является особым случае номера блока. Блок синхронизации может быть использован только в многоосной программе.

Блок синхронизации идентифицируется номером блока, стоящим за командой SYNC. Команда SYNC должна быть в той же строке, что и номер блока.

### Синтаксис:

<номер блока>: SYNC

Параметр	Описание
<номер блока>	Номер блока должен быть в диапазоне 1 – 65535.

### Альтернативные названия:

Нет

### Ошибки

1. Блоки синхронизации имеют место только в многоосных программах.
2. Все номера блоков и номера блоков синхронизации должны быть уникальными в рамках программы или подпрограммы.
3. Блоки синхронизации и номера блоков не могут появиться в следующих друг за другом операторах без промежуточной команды.
4. Номера блоков синхронизации должны быть в диапазоне 1 – 65535.

## VELOC

Команда задает скорость оси, используемую следующими далее командами движения; заданное значение скорости действует до тех пор, пока не будет изменено другим оператором VELOC. Если оператор VELOC в программе не задан, то скорость будет равна текущему значению *Jog Velocity* (*Скорость передвижения*). Движения, запрограммированные до первого оператора VELOC, также используют текущее значение *Jog Velocity*.

**Примечание:** Команды VELOC для данной оси в программе или подпрограмме должны быть разделены оператором PMOVE, оператором CMOVE или безусловным переходом.

### Синтаксис:

VELOC {<ось>} <скорость>

Параметр	Описание
<ось>	Ось может быть задана только в многоосной программе или подпрограмме. Ось может быть задана при помощи ключевых слов AXIS или констант.
<скорость>	Требуемая скорость. Может быть константа или регистр данных параметра.

### Альтернативные названия:

VEL, VELOCITY

### Ошибки

1. Ось задана в одноосной программе.
2. Не задана ось в многоосной программе.
3. Скорость должна быть константой в диапазоне 1 – 8388607.
4. Команды VELOC должны быть разделены, по крайней мере, одной командой движения.
5. Заданная ось не поддерживает программируемое движение.

## WAIT

Команда выполняет синхронизацию с некоторыми внешними событиями при помощи CTL-битов. Выполнение следующей команды задерживается до тех пор, пока не будет установлен соответствующий CTL-бит.

Одиночная команда WAIT применима только к одной оси. Поэтому в многоосной программе следует назначить номер оси, к которой должна быть применена программа WAIT. Например: WAIT AXIS1 CTL01. Если требуется ввести ожидание для обеих осей в многоосной программе, то надо использовать отдельную команду WAIT для каждой оси.

### Синтаксис:

WAIT {<ось>} <ctl>

Параметр	Описание
<ось>	Ось может быть задана только в многоосной программе или подпрограмме. Ось может быть задана при помощи ключевых слов AXIS или констант.
<ctl>	Задаёт CTL01 – CTL32.

### Альтернативные названия:

Нет

### Ошибки

1. Ось задана в одноосной программе.
2. Не задана ось в многоосной программе.
3. CTL должен быть в диапазоне 1 - 32.
4. Заданная ось не поддерживает программируемое движение.

## Структура программы и подпрограммы

### Структура одноосной программы

- **Оператор определения PROGRAM.** Этот оператор должен быть в первой строке программы. Он должен определять номер программы и номер оси. Между номером программы и ключевым словом PROGRAM должен быть пробел. Номер оси, напротив, должен быть без пробела. Например:

```
PROGRAM 1 AXIS3
```

- **Тело программы.** Тело программы содержит команды текущей программы. Следует отметить, что в одноосной программе не требуется задавать номер оси в какой-либо команде. Если это сделать, то будет выдано сообщение об ошибке. Пример правильного синтаксиса для одноосной программы:

```
ACCEL 50000
```

- **Конец программы.** Используется оператор ENDPROG. Этот оператор ясно идентифицирует конец программы и помогает отделить одну программу или подпрограмму от другой. Оператор ENDPROG должен быть единственным оператором в последней строке любой программы:

```
ENDPROG
```

### Пример одноосной программы

Следует отметить, что номер оси задается в первой строке и не задается в теле программы. Обратите также внимание, что в определении AXIS1 отсутствует пробел.

```
PROGRAM 2 AXIS1
  ACCEL 50000
  VELOC 5000
  PMOVE 10000, ABS, LINEAR
  DWELL 6000
  PMOVE 5000, ABS, LINEAR
ENDPROG
```

## Структура многоосной программы

- **Оператор определения PROGRAM.** Этот оператор должен быть в первой строке программы. Он должен идентифицировать номер программы; тот факт, что программа является многоосной, определяется использованием слова MULTI-AXIS. Например:

```
PROGRAM 3 MULTI-AXIS
```

- **Тело программы.** Тело программы содержит команды текущей программы. Следует отметить, что в многоосной программе следует задавать номер оси во многих командах. Невыполнение этого ведет к ошибке. Еще раз отмечаем, что определение номера оси не должно иметь пробела внутри. Пример правильного синтаксиса для многоосной программы:

```
ACCEL AXIS1 50000
```

- **Конец программы.** Используется оператор ENDPROG. Этот оператор ясно идентифицирует конец программы и помогает отделить одну программу или подпрограмму от другой. Оператор ENDPROG должен быть единственным оператором в последней строке любой программы:

```
ENDPROG
```

## Пример многоосной программы

Следует отметить, что определение MULTI-AXIS должно быть использовано в операторе PROGRAM в первой строке и что номера осей определяются в соответствующих командах в теле программы.

```
PROGRAM 1 MULTI-AXIS
  ACCEL AXIS1 500000
  VELOC AXIS1 5000
1:  CMOVE AXIS2 -100000, ABS, LINEAR
    DWELL AXIS2 6000
    JUMP CTL31, 1
    CALL P255
    LOAD P215, 2000
    PMOVE AXIS1 8388607, INCR, S-CURVE
ENDPROG
```

## Структура одноосной подпрограммы

- **Оператор определения SUBROUTINE.** Этот оператор должен быть в первой строке подпрограммы. Он должен идентифицировать номер подпрограммы и содержать оператор SINGLE-AXIS. Например:

```
SUBROUTINE 3 SINGLE-AXIS
```

- **Тело подпрограммы.** Тело подпрограммы содержит команды текущей подпрограммы. Следует отметить, что в одноосной подпрограмме не требуется задавать номер оси в какой-либо команде. Если это сделать, то будет выдано сообщение об ошибке. Пример правильного синтаксиса для команды одноосной подпрограммы:

```
ACCEL 50000
```

- **Конец подпрограммы.** Используется оператор ENDSUB. Этот оператор ясно идентифицирует конец подпрограммы и помогает отделить одну подпрограмму или программу от другой. Оператор ENDSUB должен быть единственным оператором в последней строке любой подпрограммы:

```
ENDSUB
```

## Пример одноосной подпрограммы

Номер оси не требуется задавать где-нибудь в одноосной подпрограмме. Это связано с тем, что одноосная подпрограмма действует на ось, определенную в одноосной вызывающей программе. Это позволяет использовать подпрограмму в различных одноосных программах, независимо от конкретного номера оси, заданного в этих программах.

```
SUBROUTINE 15 SINGLE-AXIS
  ACCEL 50000
  VELOC 10000
  PMOVE 200000, ABS, LINEAR
  DWELL 3000
  PMOVE 50000, ABS, LINEAR
ENDSUB
```

## Структура многоосной подпрограммы

- **Оператор определения SUBROUTINE.** Этот оператор должен быть в первой строке подпрограммы. Он должен идентифицировать номер подпрограммы; тот факт, что программа является многоосной, определяется использованием слова MULTI-AXIS. Например:

```
SUBROUTINE 7 MULTI-AXIS
```

- **Тело программы.** Тело подпрограммы содержит команды текущей подпрограммы. Следует отметить, что в многоосной подпрограмме следует задавать номер оси во многих командах. Невыполнение этого ведет к ошибке. Пример правильного синтаксиса для команды многоосной подпрограммы:

```
ACCEL AXIS2 50000
```

- **Конец подпрограммы.** Используется оператор ENDSUB. Этот оператор ясно идентифицирует конец подпрограммы и помогает отделить одну подпрограмму или программу от другой. Оператор ENDSUB должен быть единственным оператором в последней строке любой подпрограммы:

---

ENDSUB

**Пример многоосной подпрограммы**

```
SUBROUTINE 2 MULTI-AXIS
  ACCEL AXIS2 P100
  VELOC AXIS2 P105
2:  SYNC
    CMOVE AXIS2 P001, INCR, S-CURVE
    DWELL AXIS2 P001
    JUMP  CTL01, 2
    PMOVE AXIS2 P214, ABS, LINEAR
ENDSUB
```

## Примеры использования команд

Приведенные примеры не являются полноценными программами. Например, во многих случаях операторы PROGRAM и ENDPROG не показаны. Чтобы сделать компиляцию программы успешной, требуется добавить эти операторы (с правильным синтаксисом).

Программируемые движения имеют три параметра:

1. *Расстояние* (данные), на которое надо передвинуться, или *позиция*, в которую надо придти,
2. *Тип задатчика позиционирования* (модификатор команды), используемой для движения, и
3. *Тип ускорения* (модификатор команды), используемый для движения.

**Примечание:** Программы движения могут содержать операторы, которые используют константы как данные, связанные с командами или переменными, которые также относятся к параметрам (P0-P255).

## Абсолютное или инкрементное позиционирование

### Абсолютное позиционирование

При движении абсолютного позиционирования первым параметром является позиция, к которой следует осуществить передвижение. Ниже дан пример движения с абсолютным позиционированием.

```
PMOVE 5000, ABS, LINEAR
```

В данном примере ось будет двигаться от ее текущей позиции, где бы она ни находилась, в позицию 5000. Таким образом, расстояние, которое должно пройдено, зависит от текущей позиции оси в момент получения команды движения. Если начальная позиция оси 0, то ось должна будет пройти 5000 пользовательских единиц в положительном направлении. Если начальная позиция оси 8000, то ось должна будет пройти 3000 пользовательских единиц в отрицательном направлении. Если начальная позиция равна 5000, то ось двигаться не будет.

### Инкрементное позиционирование

При инкрементном движении первый параметр определяет расстояние, которое надо пройти от текущей позиции. Модуль DSM314 преобразует инкрементную дистанцию передвижения в абсолютную позицию передвижения. Это исключает накопление ошибки. Ниже дан пример движения с инкрементным позиционированием.

```
PMOVE 5000, INCR, LINEAR
```

В данном примере ось будет передвигаться с ее текущей позиции на позицию, расположенную на 5000 пользовательских единиц дальше. При инкрементном движении первый параметр задает текущее количество пользовательских единиц, на которое следует передвинуть ось.

## Типы ускорения

### Линейное ускорение

Пример линейного графика зависимости скорости от времени показан на рис. 7-1. Как видно, линейное движение использует постоянное (линейное) ускорение. Площадь под рисунком представляет собой пройденное расстояние.

```
ACCEL 1000
VELOC 2000
PMOVE 6000, INCR, LINEAR
```

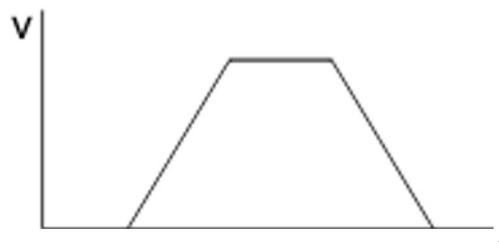


Рис. 7-1. Пример линейного движения

### Ускорение по S-кривой

Пример движения по S-кривой с представлением зависимости скорости от времени показан ниже. Как видно, ускорение по S-кривой является нелинейным. В начале движения ускорение мало и нарастает, пока не достигнет запрограммированного значения. Это имеет место в середине процесса ускорения. Затем ускорение уменьшается до нуля, когда достигается заданная скорость. Движение по S-кривой требует вдвое больше времени и расстояния для ускорения и замедления по сравнению с соответствующим линейным движением. Площадь под рисунком представляет собой пройденное расстояние.

```
ACCEL 2000
VELOC 2000
PMOVE 8000, INCR, S-CURVE
```

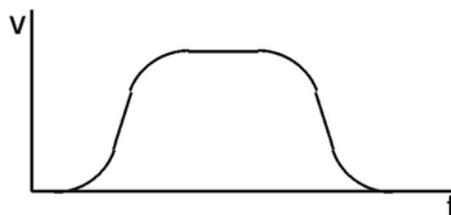


Рис. 7-2. Пример движения по S-кривой

## Типы команд программируемого движения

Приведенные примеры не являются полноценными программами. Например, во многих случаях операторы PROGRAM и ENDPROG не показаны. Чтобы сделать компиляцию программы успешной, требуется добавить эти операторы (с правильным синтаксисом).

### Движение позиционирования (PMOVE)

PMOVE должно всегда заканчиваться полной остановкой. Длительность этого останова должна быть достаточно большой, чтобы бит *In Zone (В зоне) %I* мог перейти в состояние ON до того, как сможет начаться следующее движение.

PMOVE использует последние значения запрограммированной скорости и ускорения. Если в программе движения отсутствует команда VELOC, то по умолчанию используется **Jog Velocity (Скорость передвижения)**. Если в программе движения отсутствует команда ACCEL, то по умолчанию используется **Jog Acceleration (Ускорение передвижения)**.

### Непрерывное движение (SMOVE)

SMOVE не останавливается при завершении, если после этой команды не следует команда DWELL или WAIT или если следующая программируемая скорость не задана равной нулю или если это не последняя команда программы. Переход к следующему движению осуществляется без ожидания установки бита *In Zone (В зоне) %I* в состояние ON. Нормальная команда SMOVE является командой, которая достигает своей запрограммированной позиции за то же самое время, за которое она достигает скорости расположенной далее вслед за нею команды Move.

SMOVE использует последние значения запрограммированной скорости и ускорения. Если в программе движения отсутствует команда VELOC, то по умолчанию используется **Jog Velocity (Скорость передвижения)**. Если в программе движения отсутствует команда ACCEL, то по умолчанию используется **Jog Acceleration (Ускорение передвижения)**.

Может быть использована специальная форма команды SMOVE, чтобы DSM314 достиг запрограммированной позиции SMOVE до начала изменения скорости, связанного со следующей командой движения (т.е. вся команда SMOVE выполняется при постоянной скорости). **Программирование инкрементной команды SMOVE с операндом 0 (например: SMOVE 0, INCR, LINEAR) задержит изменение скорости сервопривода до следующей, расположенной далее команды движения.** Приведенная ниже последовательность команд показывает это (предположим, что ускорения заданы, чтобы выполнить движения нормально):

Команда	Данные	Комментарии
VELOC	10000	//Задается скорость первого движения = 10000
CMOVE	15000, ABS, LINEAR	//Достигается скорость второго движения (20000) к позиции = 15000
VELOC	20000	//Задается скорость второго движения = 20000
CMOVE	30000, ABS, LINEAR	(*Скорость сохраняется = 20000 до достижения позиции = 30000, затем скорость становится = 5000*)
CMOVE	0, INCR, LINEAR	(*Модуль DSM314 информируется, что следует подождать следующего движения, прежде чем перейти к следующей скорости*)
VELOC	5000	//Задается скорость третьего движения = 5000
PMOVE	40000, ABS, LINEAR	//Позиция окончательного останова = 40000

**Примечание:** Символы пустого пространства (пробелы, табуляторы и т.п.) используются в приведенной выше программе для улучшения читаемости.

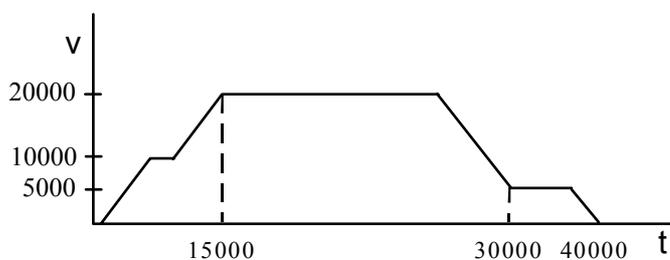


Рис. 7-3. Пример 1: До использования CMOVE (0)

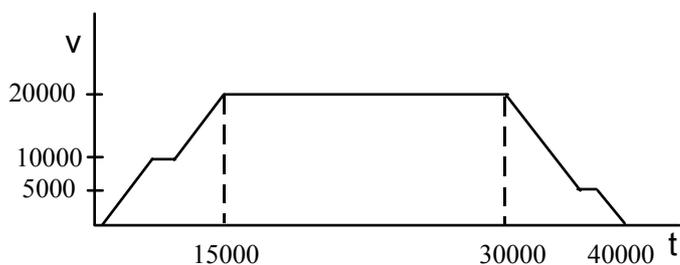


Рис. 7-4. Пример 2: При использовании CMOVE (0)

## Программируемые движения

Комбинируя команды CMOVE и PMOVES, абсолютные и инкрементные движения, линейное изменение скорости и изменение по S-кривой можно создать практически любой график движения. Приведенные далее примеры показывают несколько простых графиков движения, а также некоторые распространенные ошибки программирования в программах движения.

### Пример 1: комбинирование PMOVE и CMOVE

Данный пример показывает, как, просто комбинируя команды PMOVE и CMOVE, создавать графики движения.

```
ACCEL 1000
VELOC 2000
PMOVE 5000, ABS, LINEAR
VELOC 1200
PMOVE 10000, ABS, S-CURVE
ACCEL 1500
VELOC 2800
CMOVE 6000, INCR, LINEAR
VELOC 1200
CMOVE 23000, ABS, S-CURVE
ACCEL 1000
VELOC 2800
PMOVE 5000, INCR, LINEAR
```

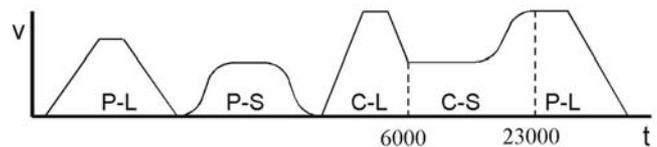


Рис. 7-5. Комбинирование PMOVE и CMOVE

Первая команда PMOVE выполняет ускорение до заданной скорости, перемещение на определенное расстояние и замедление до остановки. Это определяется тем, что после любой команды PMOVE происходит останов. Когда первое движение прекращается, то это значит, что заданное расстояние пройдено.

Вторым движением является S-кривая PMOVE. Как и в первой команде PMOVE, здесь происходит ускорение до заданной скорости, движение в течение некоторого времени и замедление до нулевой скорости.

Следующим движением является линейное CMOVE. Происходит ускорение до запрограммированной скорости, движение в течение некоторого времени и линейное замедление до меньшей скорости. Когда команда CMOVE заканчивает свое действие, имеется заданная позиция и скорость следующего движения. Поэтому, когда начинается четвертое движение, оно уже имеет свою запрограммированную скорость.

Четвертым движением является CMOVE, поэтому, когда происходит приближение к его конечной позиции, выполняется ускорение до скорости пятого движения к моменту завершения четвертого. На рисунке видно, что ускорение четвертого движения представляет собой S-кривую.

В завершение совершается пятое движение на запрограммированной скорости в течение некоторого времени, после чего происходит замедление до нуля. Любые последующие движения после пятого будут начинаться с нулевой скорости, поскольку пятым является PMOVE.

## Пример 2: Изменение режима ускорения на профиле

Приведенный ниже пример показывает, как можно использовать различные ускорения и даже режимы ускорений на профиле, применяя команды CMOVE. Первая команда CMOVE выполняет линейное ускорение до заданной скорости. Поскольку скорость второй команды CMOVE равна первой, то первая команда CMOVE завершает свое движение без изменения скорости. Изменение скорости второго движения происходит по S-кривой; скорость снижается до нуля.

```
ACCEL 2000
VELOC 6000
CMOVE 13000, ABS, LINEAR
ACCEL 4000
CMOVE 15000, INCR, S-CURVE
```

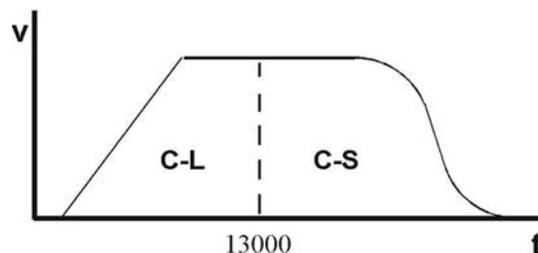


Рис. 7-6. Изменение режима ускорения на профиле

## Пример 3: не хватает расстояния, чтобы выйти на заданную скорость

Команды CMOVE и PMOVE могут быть запрограммированы и при нехватке расстояния для выхода на заданную скорость. Рисунок далее показывает выполнение команды CMOVE, которая не может достичь заданной скорости. Модуль DSM314 выполняет ускорение до точки, где он должен начать замедление, чтобы выйти на запрограммированную позицию C1 при скорости второй команды CMOVE.

```
ACCEL 2000
VELOC 8000
CMOVE 7000, INCR, LINEAR
ACCEL 10000
VELOC 2000
CMOVE 4400, INCR, LINEAR
```

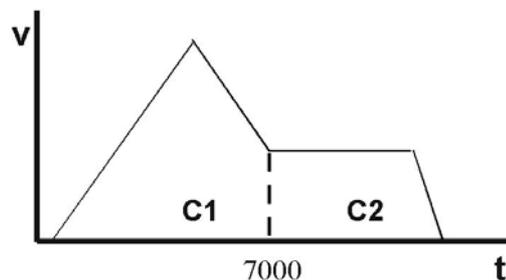


Рис. 7-7. Ситуация, когда не хватает расстояния, чтобы выйти на заданную скорость

## Пример 4: Зависание движения при отсутствии расстояния

Серьезной ошибкой программирования является "зависание" (т.е. возникновение ситуации, когда у генератора команд нет подходящих возможностей) движения на высокой скорости, когда расстояние передвижения заканчивается. В приведенном ниже примере первая команда CMOVE выполняет ускорение до большой скорости. Вторая команда CMOVE имеет ту же скорость. Однако расстояние, заданное для второй команды CMOVE, очень мало. Поэтому ось, двигающаяся на очень большой скорости, должна остановиться на коротком отрезке. Если запрограммированное ускорение недостаточно велико, то график скорости может стать таким, как он показан на рисунке. Модуль DSM314 пытается предотвратить перебег конечной позиции, давая команду нулевой скорости. Такое быстрое изменение скорости является нежелательным; оно может вызвать повреждение станка.

```
ACCEL 500
VELOC 3000
CMOVE 9000, ABS, LINEAR
ACCEL 600
CMOVE 4800, INCR, LINEAR
```

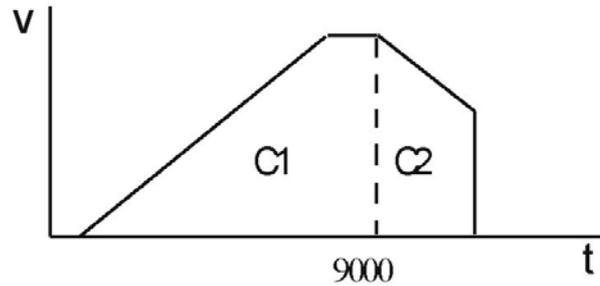


Рис. 7-8. Зависание модуля DSM314 при отсутствии расстояния

## Команда DWELL

Команда DWELL используется для обеспечения отсутствия движения в течение заданного интервала времени (в миллисекундах). Команда DWELL может использовать значение, находящееся в назначенном параметре.

Команда DWELL, следующая за командой CMOVE, вызывает остановку CMOVE перед следующим движением, если только в качестве длительности задержки не указано ноль миллисекунд. Команда DWELL рассматривается как "нулевая" команда и пропускается (CMOVE переходит к следующей команде Move, расположенной за DWELL), если команда DWELL имеет значение "нуль" или если она ссылается на регистр параметра, имеющий значение "нуль".

Одиночная команда DWELL применима только к одной оси. Поэтому в многоосной программе следует назначить номер оси для каждой команды DWELL. Например: DWELL AXIS1 2000. Если требуется ввести паузу для обеих осей в многоосной программе, то надо использовать команду DWELL для каждой оси.

## Пример 5: DWELL

Ниже показан простой график движения, который выполняет перемещение к определенной точке, выжидает там некоторое время, а затем возвращается в исходную точку.

```
ACCEL 30000
VELOC 15000
PMOVE 120000, ABS, LINEAR
DWELL 4000
PMOVE 0, ABS, LINEAR
```

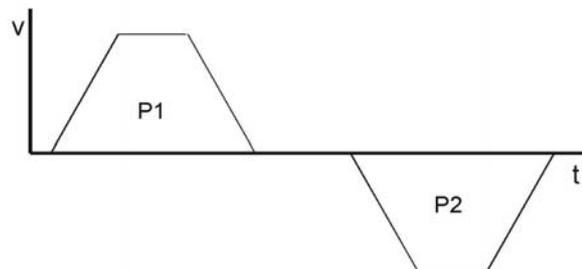


Рис. 7-9. Пример команды Dwell

## Команда Wait

Команда WAIT похожа на команду DWELL. Вместо прекращения движения на определенный промежуток времени команда WAIT останавливает движение до перехода соответствующего CTL-бита в состояние ON. Таким образом, движение прекращается всякий раз, когда встречается команда WAIT, даже если CTL-бит находился в состоянии ON при появлении этой команды. Роль переключателя для продолжения программы может выполнять любой из двенадцати CTL-битов.

Если в предыдущем примере команду WAIT заменить командой DWELL, то график движения будет тем же, за исключением того, что вторая команда PMOVE не будет запущена, пока CTL-бит не окажется в состоянии ON. Если CTL-бит был в состоянии ON, когда программа достигла команды WAIT, то выполнение второй команды PMOVE начнется немедленно после завершения первой команды PMOVE.

Таким образом, если бы в предыдущем примере была использована вместо команды DWELL команда WAIT, то команды CMOVE и PMOVE создали бы аналогичные график скорости. Команда WAIT прекращает движение независимо от того, какое движение ей предшествовало - CMOVE или PMOVE.

Одиночная команда WAIT применима только к одной оси. Поэтому в многоосной программе следует назначить номер оси для каждой команды DWELL. Например: WAIT AXIS1 CTL001. Если требуется ввести ожидание для обеих осей в многоосной программе, то надо использовать отдельную команду WAIT для каждой оси.

## Подпрограммы

Модуль DSM314 может хранить до десяти отдельных программ и до сорока подпрограмм. Подпрограммы разделяются на два типа: *одноосные* и *многоосные*. Подпрограммы могут быть использованы всеми программами движения, создаваемыми при помощи Motion Editor. Команды одноосной подпрограммы не должны содержать номер оси; соблюдение этого позволяет вызывать одноосные подпрограммы из любой одноосной программы, (команды подпрограммы используют номер оси, заданный в вызывающей программе). Команды в многоосной подпрограмме содержат номера осей аналогично командам в многоосной программе. Многоосные подпрограммы могут быть вызваны только многоосными программами или подпрограммами. Одноосные подпрограммы могут быть вызваны только одноосными программами или подпрограммами. Например, одноосная программа для оси 1 и одноосная программа для оси 2 могут вызвать одну и ту же одноосную подпрограмму одновременно. Каждой подпрограмме должен быть назначен уникальный номер от 1 до 40.

Вызов подпрограммы осуществляется командой CALL, которая задает номер вызываемой подпрограммы. Если в ходе выполнения программы появляется команда CALL, то выполнение программы перенаправляется на подпрограмму. После завершения подпрограммы происходит возврат в программу на команду, следующую за командой CALL. Подпрограммы можно вызывать из другой подпрограммы; однако, если подпрограмма запущена, она должна быть завершена до того, как она будет вызвана снова для той же оси. Иначе говоря, рекурсия не разрешается.

## Номера блоков и переходы

Номера блоков используются как опорные точки в программе движения и для управления при тестировании переходов. Слово данных %AI отображает номер текущего блока, который можно отслеживать, чтобы обеспечить правильное выполнение программы или чтобы определить срок наступления того или иного события. Номер блока может также выступать как назначение команды JUMP. Jumps (Переходы) могут быть безусловными или условными. Команда безусловного перехода просто указывает модулю DSM314

продолжить выполнение программы с блока с указанным номером. Условный переход происходит только в том случае, если выполнено определенное условие. Далее приведены примеры обоих типов переходов.

## Безусловные переходы

### Пример 6: Безусловный переход

В приведенной программе выполняется команда PMOVE, делается задержка на 2 секунды, а затем происходит безусловный переход назад на начало программы – на блок 1. Таким образом, повторяется команда PMOVE до тех пор, пока не будет достигнута граница передвижения (*Верхний* или *Нижний ограничитель программного движения*) или выключатель выхода за границу. Для остановки программы может быть также использована битовая команда *Abort All Moves (Прекратить все движения) %Q*.

```

ACCEL 10000
VELOC 30000
1:
PMOVE 200000, INCR, LINEAR
DWELL 2000
JUMP UNCOND, 1
    
```

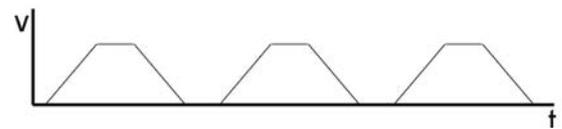


Рис. 7-10. Безусловный переход

## Условные переходы

Условным переходом является команда JUMP с STL-битом, определенным в команде. Условные переходы являются командами типа 1 в том, что они влияют на ход программы, но они также похожи на команды типа 2, поскольку они не оказывают воздействия до тех пор, пока выполняется команда типа 3, расположенная далее. Когда выполняется команда условного перехода JUMP, модуль DSM314 проверяет определенный STL-бит. Если бит в состоянии ON, то выполнение программы переходит на блок с указанным номером; если бит находится в состоянии OFF, то программа переходит к команде, расположенной за командой JUMP. Следует отметить, что команда типа 3 после условного перехода и в точке назначения перехода влияет на характер перехода.

Команды условного перехода не следует использовать в многоосных программах, содержащих блоки SYNC, за исключением тех случаев, когда JUMP включается, поскольку обе оси проверяют одну и ту же команду JUMP. Несоблюдение этой рекомендации может привести к непредсказуемым последствиям.

Проверка условного перехода **запускается**, когда активизируется команда PMOVE, CMOVE, DWELL или WAIT, расположенная за командой условного перехода JUMP.

Когда проверка условного перехода действует, указанный STL-бит проверяется с интервалом обновления контура позиционирования (0.5, 1.0 или 2.0 миллисекунды в зависимости от конфигурации).

Проверка условного перехода **заканчивается**, когда указанный STL-бит оказывается в состоянии ON (происходит включение перехода) или когда активизируется новый номер блока.

Если запрограммировано несколько условных переходов без промежуточной команды PMOVE, CMOVE, DWELL или WAIT, то определяется только последний условный переход.

Условный переход не может быть использован в качестве последней строки подпрограммы (или на строке, предшествующей безусловному переходу в конце подпрограммы),

поскольку проверка условия перехода заканчивается, когда обрабатывается команда конца подпрограммы.

Говоря коротко, условный переход передает управление, новому блоку программы, основываясь на состоянии ON одного из внешних входных CTL-битов. Проверка статуса CTL-бита может быть выполнена один раз или проводиться непрерывно в течение действия следующей команды типа 3, если она находится в том же блоке программы. Многократные условные переходы не поддерживаются в одном блоке программы (в примере ниже показано такое некорректное использование команды условного перехода).

### Пример 1 условного перехода:

```
PROGRAM 1 MULTI-AXIS
  VELOC  AXIS1 10000
  ACCEL  AXIS1 10000
1:  JUMP  CTL01, 2           //Эта команда перехода JUMP
                             игнорируется
JUMP  CTL02, 3           //Эта команда перехода JUMP
                             принимается
  CMOVE  AXIS1 +40000, INCR, LINEAR
2:  CMOVE  AXIS2 +20000, INCR, LINEAR
3:  PMOVE  AXIS2 +100000, ABS, LINEAR
4:  DWELL  AXIS2 100
ENDPROG
```

Первая команда JUMP запрограммирована некорректно, поскольку (1) за ней не следует промежуточная команда типа 3, и (2) она находится в том же блоке, что и другая команда JUMP. Если новый номер блока активизируется ПОСЛЕ команды условного перехода JUMP, то проверка условия перехода проводится лишь один раз.

### Пример 2 условного перехода:

```
PROGRAM 2 AXIS1
  VELOC  10000
  ACCEL  10000
1:  CMOVE  20000, ABS, LINEAR
    JUMP  CTL01, 3
2:  PMOVE  40000, ABS, LINEAR //CTL01 проверяется только один раз
3:  DWELL  100
ENDPROG
```

В примере выше бит *CTL01* проверяется лишь один раз, поскольку команда PMOVE, следующая за командой JUMP, имеет новый номер блока (2). Однако изменение расположения номера блока 2 вызывает проверку CTL-бита в течение всей команды PMOVE, расположенной за командой JUMP, как видно в следующем примере:

### Пример 3 условного перехода:

```
PROGRAM 3 AXIS1
  VELOC  10000
  ACCEL  10000
1:  CMOVE  20000, ABS, LINEAR
2:  JUMP  CTL01, 3
    PMOVE  40000, ABS, LINEAR //CTL01 проверяется в течение всего
                             действия команды PMOVE
3:  DWELL  100
ENDPROG
```

В данном примере бит CTL01 проверяется в течение всего действия команды PMOVE, поскольку команды PMOVE и JUMP находятся в одном и том же блоке.

Модуль DSM314 может выполнить условный переход из активной команды CMOVE в программный блок, содержащий CMOVE или PMOVE, без остановки. **Чтобы ось могла выполнить такой переход без остановки, расстояние, заданное командой CMOVE или PMOVE в блоке перехода, должно быть больше, чем интервал остановки сервоустройства.** Интервал остановки сервоустройства рассчитывается с использованием имеющейся заданной скорости и параметров изменения скорости, которые начинают действовать при активизации блока перехода.

Ось ОСТАНОВИТСЯ перед переходом, если произойдет включение условного перехода, при наличии любого из приведенных ниже условий:

- Если активна команда PMOVE
- Если активна команда CMOVE, а блок назначения перехода содержит команду CMOVE или PMOVE, определяющую движение в противоположном направлении.
- Если активна команда CMOVE и блок назначения перехода содержит команду CMOVE или PMOVE, определяющую движение в том же направлении, но расстояния недостаточно для остановки оси.
- Если активна команда CMOVE, но блок назначения перехода содержит команду DWELL, WAIT или END (программы).

**Если ось не останавливается перед условным переходом, то используются текущее заданное значение и режим ускорения.**

Безусловный переход не вызывает остановки оси перед переходом на новый программный блок. Например, команда CMOVE, за которой следует команда безусловного перехода JUMP к следующей команде CMOVE, действует так, как две последовательно расположенные команды CMOVE без промежуточного безусловного перехода.

Если выполняется проверка условного перехода, когда процессор команд DSM314 получает команду CALL SUBROUTINE (ВЫЗОВ ПОДПРОГРАММЫ), то ось **останавливается** и заканчивает проверку условия перехода до того, как будет выполнена команда CALL.

Если выполняется проверка условного перехода, когда процессор команд DSM314 получает команду END SUBROUTINE (КОНЕЦ ПОДПРОГРАММЫ), то ось **останавливается** и заканчивает проверку условия перехода до того, как будет выполнена команда END SUBROUTINE.

## Проверка условия перехода

Команда условного перехода делает проверку условия перехода. Если CTL-бит находится в состоянии ON, то переход немедленно выполняется. Если CTL-бит находится в состоянии OFF, то модуль DSM314 следит за состоянием CTL-бита и отслеживает конечную точку перехода. Такой мониторинг CTL-бита называется проверкой условия перехода. Если в процессе проверки условия перехода CTL-бит принимает состояние ON до того, как встретится команда BLOCK, другая команда JUMP или команда CALL, то выполняется переход. Указанные команды прекращают проверку условия перехода.

### Пример 7: Проверка условия перехода

Рассмотрим приведенные ниже два примера сегментов одноосной программы. В примере 1 движение в позицию 2000 выполняется до начала проверки условия перехода. Номер блока, стоящий непосредственно за командой JUMP, заканчивает проверку условия перехода. Таким образом, интервал времени, в течение которого отслеживается, CTL-бит, очень мал. Однако в примере 2 команда JUMP стоит перед командой CMOVE. Поэтому

проверка условия перехода начинается до начала движения и продолжается в течение всего того времени, которое занимает движение. Если в процессе совершающегося движения CTL-бит переходит в состояние ON, то выполняется переход. После завершения движения появляется номер следующего блока, который заканчивает проверку условия перехода, и выполнение программы продолжается обычным образом. Если до номера следующего блока было запрограммировано дополнительное движение, то проверка условия перехода будет продолжаться также в течение этого движения до тех пор, пока не появится номер следующего блока.

**Пример 1:**

```

ACCEL 5000
VELOC 1000
1:
  CMOVE 2000, ABS, LINEAR
  JUMP  CTL01, 3
2:

```

**Пример 2:**

```

ACCEL 5000
VELOC 1000
1:
  JUMP  CTL01, 3
  CMOVE 2000, ABS, LINEAR
2:

```

## Нормальный останов перед командой JUMP

Команда условного перехода похожа на команды типа 2 в том, что проверка условия перехода не запускается до тех пор, пока выполняется команда типа 3, расположенная сразу же за командой JUMP. Если эта команда типа 3 нормальным образом прекращает движение, то движение прекращается до того, как начнется проверка условия перехода. Командами типа 3, которые прекращают движение, являются следующие: DWELL, WAIT, ENDPROG и движения, совершаемые в противоположном направлении.

Таким образом, хотя CTL-бит может быть в состоянии ON до того, как начнет выполняться блок с командой условного перехода и командой типа 3, движение оси прекратится до того, как выполнение программы продолжится в конечной точке перехода. Такая остановка не является Jump Stop (Остановка перехода), которая описана в примере 10.

### Пример 8: Нормальный останов перед командой JUMP

В приведенном далее примере за командой перехода следует команда DWELL. Модуль DSM314, работающий с упреждением, знает, что он должен сделать остановку после команды CMOVE. Поэтому он выполняет остановку до выполнения команды DWELL. Поскольку проверка условия перехода не начинается до тех пор, пока не выполнена команда DWELL, то проверка начинается после того, как движение прекратится. Проверка условия перехода заканчивается под действием соответствующей команды BLOCK, когда запускается следующая далее команда CMOVE. Штриховые линии на графике скорости показывают, когда имеет место проверка условия перехода. В этом примере бит CTL03 не включается в состояние ON во время выполнения программы.

```

1: ACCEL 5000
   VELOC 10000
   CMOVE 60000, INCR, LINEAR
2: JUMP  CTL03, 4
   DWELL 4000
3: ACCEL 10000
   VELOC 5000
   CMOVE 15000, INCR, LINEAR
4:

```



Рис. 7-11. Нормальный останов перед командой JUMP

## Переход без остановки

Если командой типа 3, которая следует за командой условного перехода, является CMOVE, а командой типа 3 в точке назначения перехода является команда движения с достаточным расстоянием, чтобы можно было полностью погасить скорость до нуля после завершения, то переход выполняется без остановки. Это единственная возможность поддержать движение при выполнении перехода.

### Пример 9: Выполнение команды JUMP без остановки

Здесь дан простой пример условного перехода от одной команды CMOVE к другой. В то время, когда происходит проверка условия перехода на бите *CTL03*, первая команда CMOVE разгоняет ось до заданной скорости. До штриховой линии бит *CTL03* находится в состоянии OFF, но после этой линии – в состоянии ON. Выполнение программы немедленно передается на блок 3, и там запускается команда CMOVE. Поскольку скорость в точке назначения иная, то происходит изменение скорости с ускорением, заданным для блока назначения перехода. В конце, когда заканчивается выполнение второй команды CMOVE, скорость снижается до нуля, и программа завершается.

```

1: ACCEL 2000
   VELOC 10000
   JUMP CTL03, 3
   CMOVE 120000, INCR, LINEAR
3: ACCEL 20000
   VELOC 5000
   CMOVE 15000, INCR, LINEAR
    
```

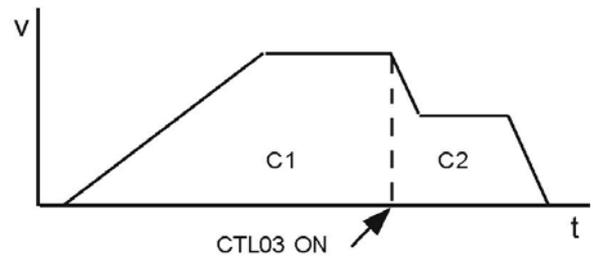


Рис. 7-12. Выполнение команды JUMP без остановки

## Остановка перехода

Остановка перехода – это такая остановка, которая вызвана переходом. **Когда происходит остановка перехода, то используются текущие заданные значение ускорение и режим ускорения.** Следует отметить, что движение по S-кривой выходит на постоянную скорость, прежде чем начать торможение. Более детальную информацию см. в примере перехода по S-кривой. Имеются два пути создания остановки перехода, описанные далее.

Условный переход, включаемый во время действия команды PMOVE, будет всегда порождать остановку перехода. Поскольку PMOVE всегда вызывает остановку перед последующим движением, то остановка перехода всегда происходит, если в ходе выполнения PMOVE возникает переход.

Однако если условный переход включается во время CMOVE, то остановка перехода не происходит, если командами движения в точке назначения перехода являются PMOVE или CMOVE, имеющие достаточное расстояние в том же направлении. Остановка перехода происходит, если PMOVE или CMOVE в точке назначения перехода не имеют достаточного расстояния или если они представляют движение в противоположном направлении.

При движении по S-кривой будет реализовываться один из двух сценариев остановки перехода. Если переход происходит после средней точки процесса изменения скорости, то ускорение или замедление завершаются без инициации остановки перехода. Если переход происходит до средней точки процесса изменения скорости, то график немедленно начинает выравниваться. Когда значение ускорения или замедления становится равным нулю, начинается процесс остановки перехода. См. примеры S-кривых при переходе.

### Пример 10: Остановка перехода

Ниже дан пример условного перехода с остановкой перехода. Расширением примера 5 (DWELL) является введение отслеживания внешнего CTL-бита, который будет информировать о проблеме с положительным движением. Если этот CTL-бит так и не включится, то график для приведенной ниже программы будет идентичным тому, что показан в примере DWELL. Если произойдет включение этого CTL-бита в ходе выполнения первой команды PMOVE или при DWELL, то немедленно начнется движение в обратном направлении.

Если этот CTL-бит включится на первой команде PMOVE (штриховая линия), то будет иметь место график, показанный на рисунке. Поскольку первое движение закончилось раньше из-за включения CTL-бита, то второе движение не должно перемещать ось на такое же расстояние, как в примере с DWELL, чтобы выйти обратно на позицию 0. Следует отметить, что поскольку движение, запрограммированное в точке назначения перехода, осуществляется в направлении, противоположном исходному, то график будет идентичным тому, который получился бы, если бы вместо команд PMOVE были команды CMOVE.

```

ACCEL    30000
VELOC    15000
1: JUMP   CTL09, 2
   PMOVE  120000, ABS, LINEAR
   DWELL  4000
2: PMOVE  0, ABS, LINEAR

```

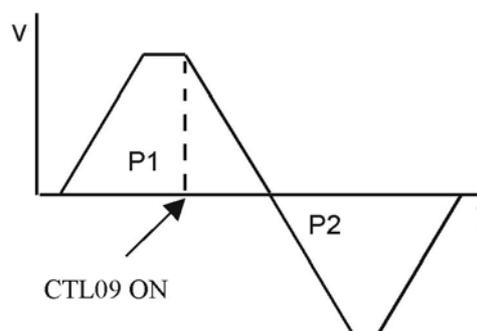


Рис. 7-13. Остановка перехода

### Пример 11: Команда перехода с последующей командой PMOVE

В данном примере с командой JUMP после нее следует команда PMOVE, обеспечивающая движение в том же направлении. График скорости на рисунке ниже показывает ускорение и перемещение для первой команды CMOVE и замедление до скорости PMOVE. Бит CTL01, находящийся в состоянии OFF, когда начинает работать команда PMOVE, включается в состояние ON в момент времени, отмеченный второй штриховой линией. Движение прекращается после PMOVE, даже если происходит условный переход на другой блок. Таким образом, бит CTL01 включает замедление до нуля до того, как начнет работать завершающая команда CMOVE.

```

1: ACCEL    2000
   VELOC    8000
   CMOVE    76000, INCR, LINEAR
2: ACCEL    1000
   VELOC    4000
   JUMP     CTL01, 3
   PMOVE    50000, INCR, LINEAR
3: ACCEL    6000
   VELOC    6000
   CMOVE    6000, INCR, LINEAR
    
```

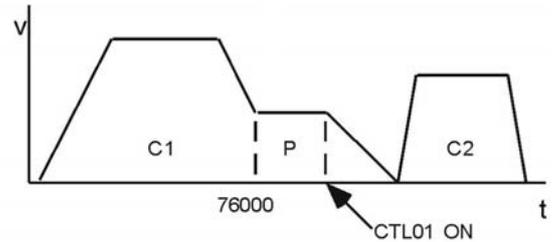


Рис. 7-14. Команда перехода с последующей командой PMOVE

### Переходы с S-CURVE (S-кривая)

Переходы, происходящие во время линейного движения или во время движения с S-кривой при постоянных скоростях, немедленно начинают изменение скорости на новое значение (ускорение или замедление). Однако переходы во время ускорения или замедления по S-кривой требуют соблюдения различающихся правил, чтобы сохранить S-профиль. То, что происходит, когда переход возникает при движении по S-кривой во время изменения скорости, зависит от двух обстоятельств: а) происходит переход до или после средней точки (точка, где величина изменения скорости (ускорения или замедления) имеет наибольшее значение), б) выше или ниже скорость в точке назначения перехода текущей скорости.

#### Переходы на S-кривой после средней точки ускорения или замедления

Если переход возникает после средней точки изменения скорости, то изменение продолжается нормальным образом, пока не будет достигнуто постоянное значение скорости; затем скорость будет изменяться на новое значение с использованием режима изменения скорости движения в точке назначения перехода.

## Пример 12: Переходы на S-кривой после средней точки ускорения или замедления

В следующем примере переход возникает в конце замедления (штриховая линия). Процесс замедления продолжается до тех пор, пока не будет достигнута постоянная скорость, после чего начинается ускорение до более высокого значения скорости.

```

ACCEL 50000
VELOC 100000
1: JUMP CTL01, 3
   CMOVE 500000, ABS, S-CURVE
2: VELOC 60000
   CMOVE 500000, INCR, S-CURVE
3: VELOC 85000
   ACCEL 100000
   CMOVE 250000, INCR, S-CURVE

```

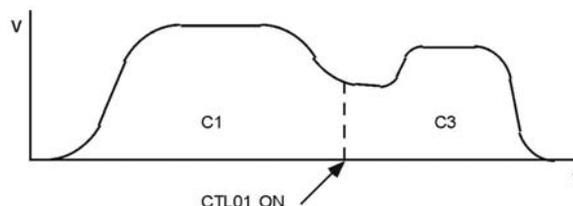


Рис. 7-15. Переход после средней точки ускорения или замедления

## Переходы на S-кривой до средней точки ускорения или замедления

Если переход возникает до средней точки ускорения или замедления, то результат зависит от того, выше или ниже скорость в точке назначения перехода той скорости, которая имеется к моменту возникновения перехода. В том случае, если идет ускорение, но новая скорость ниже, или идет замедление, но новая скорость выше, модуль DSM314 немедленно начинает сводить ускорение или замедление на нуль. При нулевой скорости DSM314 использует для выхода на новую скорость значения ускорения и скорости, заданные для точки назначения перехода.

## Пример 13: Переходы на S-кривой до средней точки ускорения или замедления

В приведенном ниже примере переход возникает в процессе ускорения при выполнении первой команды CMOVE (первая штриховая линия). Поскольку скорость в точке назначения перехода ниже скорости первой команды CMOVE, то модуль DSM314 снижает ускорение до нуля. Постоянная скорость при нулевом ускорении достигается на второй штриховой линии. После этого модуль DSM314 начинает замедление до нового значения скорости, используя ускорение, заданное для точки назначения перехода. В завершение вторая команда CMOVE заканчивается.

```

ACCEL 1000
VELOC 50000
1: JUMP CTL01, 3
   CMOVE 50000, INCR, S-CURVE
3: VELOC 5000
   ACCEL 10000
   CMOVE 15000, INCR, S-CURVE

```

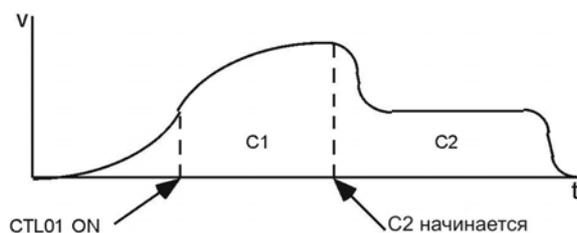


Рис. 7-16. Переход до средней точки ускорения или замедления

## Переходы на S-кривой к более высокому значению ускорения при ускорении или к более низкому значению замедления при замедлении

Второй случай рассматривает переход к более высокой скорости при ускорении или к более низкой скорости при замедлении. Если имеет место такая ситуация, то модуль DSM314 продолжает ускорение или замедление, осуществлявшееся при первом движении. Такое ускорение или замедление продолжается (так же как при линейном изменении скорости), пока ось не приблизится к новой скорости. После этого для вывода ускорения или замедления на ноль используется нормальная S-кривая.

### Пример 14: Переходы на S-кривой к более высокому значению скорости при ускорении или к более низкому значению скорости при замедлении

В данном примере команда JUMP включается на начальной фазе ускорения (первая штриховая линия); скорость в точке назначения перехода выше, чем скорость текущего движения. Первая штриховая линия показывает максимальное ускорение первой команды CMOVE. Это значение сохраняется, и ось продолжает ускоряться до тех пор, пока она по S-кривой не выйдет на постоянную скорость. Постоянная скорость (вторая штриховая линия) показывает начало второй команды CMOVE. Это движение продолжается, пока не произойдет замедление до нуля в конце программы.

```

ACCEL 50000
VELOC 30000
1: JUMP CTL02, 2
   CMOVE 150000, INCR, S-CURVE
2: VELOC 90000
   ACCEL 25000
   CMOVE 500000, INCR, S-CURVE

```

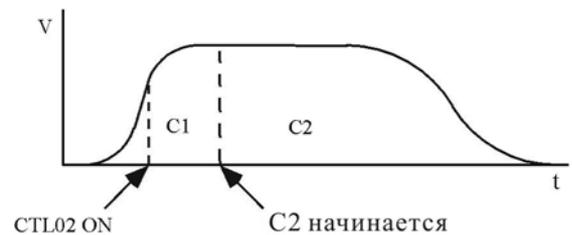


Рис. 7-17. Переходы к более высокому значению скорости при ускорении или к более низкому значению скорости при замедлении

## Дополнительная информация по программируемому движению

Приведенные далее примеры не являются полноценными программами. Например, во многих случаях операторы PROGRAM и ENDPORG не показаны. Чтобы сделать компиляцию программы успешной, требуется добавить эти операторы (с правильным синтаксисом).

### Максимальная длительность ускорения

Максимальная длительность программируемого ускорения или замедления составляет 131 секунду. Если расчетная длительность ускорения или замедления превышает это время, то модуль DSM314 исходит из того, что процесс изменения скорости будет продолжаться 131 секунду. Чтобы получить большее значение длительности ускорения (замедления), следует использовать несколько команд CMOVE с увеличивающимися или уменьшающимися значениями скорости.

### Пример 15: Максимальная длительность ускорения

Приведенные ниже два примера программ действительны только для модуля DSM314, использующего длительность цикла обновления контура позиционирования 2 мс. В примере 1 показана возможная проблема, связанная с очень большой длительностью ускорения; пример 2 показывает возможное решение этой проблемы. В примере 1 видно, что для выхода на заданную скорость 24 000 при ускорении 100 требуется 240 секунд ( $24000 \div 100 = 240$ ). Поскольку это превышает границу в 131 секунду, заложенную в DSM для процесса изменения скорости, то модуль DSM проведет расчет, используя эту границу. В данном случае расчет DSM покажет, что для достижения скорости 24 000 за 131 секунду требуется ускорение 183. В примере 1 график из сплошной линии показывает более высокое значение ускорения (183), которое использует DSM. График из штриховой линии показывает требуемый (программируемый) уровень ускорения и график скорости, который не реализуется.

```
ACCEL 100
VELOC 24000
PMOVE 8000000, INCR, LINEAR
```

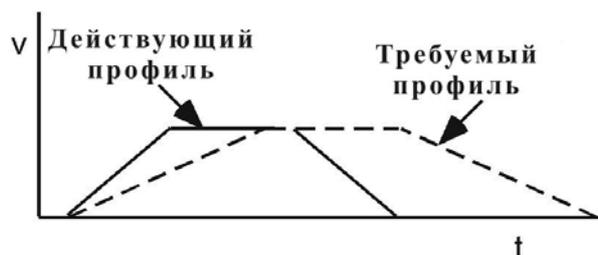


Рис. 7-18. Пример 1 максимальной длительности ускорения

Одним из возможных решений (которое требует некоторых специальных расчетов) для получения низкого значения ускорения, действующего в течение длительного времени, является разделение движения на отдельные непрерывные участки (используя команды CMOVE), на каждом из которых длительность изменения скорости составляет менее 131 секунды. В задаче, демонстрируемой примером 1, запрограммированное движение требует 240 секунд, как на ускорение, так и на замедление. Деление этого интервала времени пополам путем использования двух движений с длительности процесса изменения скорости 120 секунды в каждом интервале, позволяет провести движения в пределах границы DSM в 131 секунду. Такая схема использована в примере ниже.

Пример 2 показывает, как можно получить результат, требуемый в примере 1, заменой одного движения из примера 1 четырьмя. Четыре движения требуются потому, что участок графика, как ускорения, так и замедления должен быть разделен на два движения. Чтобы поделить полную длительность ускорения (или замедления) пополам, следует рассчитать расстояние до средней точки каждого наклонного участка, где скорость составляет 12 000; это расстояние составляет 720 000 пользовательских единиц.

Расстояние, проходимое при ускорении или замедлении, рассчитывается по формуле:

$$\text{Пройденное расстояние} = \frac{\text{Изменения скорости} \times \text{требуемое время}}{2}$$

$$720\,000 = \frac{12\,000 \times 120}{2}$$

(Поскольку 240 секунд требуются для достижения скорости 24 000, то скорость 12 000 может быть получена за 120 секунд). Как исходная команда CMOVE, так и завершающая команда PMOVE используют это расстояние. Вторая команда CMOVE "вступает в дело" в средней точке наклонного участка ускорения вслед за первой командой CMOVE и ускоряет ось до требуемой скорости 24 000. Третья команда CMOVE требуется для деления участка замедления графика скорости. Завершающее движение PMOVE "вступает в дело" после третьей команды CMOVE в средней точке участка замедления (720 000 пользовательских единиц от конечной позиции). Третья команда CMOVE, приближаясь к своей конечной точке, автоматически замедляется до скорости PMOVE 12 000. Штриховые линии примера 2 показывают четыре отдельных участка. Чтобы рассчитать расстояния для второй и третьей команд CMOVE, следует вычесть расстояния, рассчитанные для первой команды CMOVE и завершающей команды PMOVE (по 720 000 для каждой команды – всего 1 440 000), из полного расстояния 8 000 000. Это дает расстояние 6 560 000, которое следует поделить пополам между второй и третьей командами CMOVE (по 3 280 000).

```

ACCEL 100
VELOC 12000
CMOVE 720000, INCR, LINEAR
VELOC 12000
CMOVE 3280000, INCR, LINEAR
VELOC 24000
CMOVE 3280000, INCR, LINEAR
VELOC 12000
PMOVE 720000, INCR, LINEAR
    
```



Рис. 7-19. Пример 2 максимальной длительности ускорения

## Остановка подачи модулем DSM314

Остановка подачи используется для временной приостановки исполнения программы без ее завершения; часто это используется для проверки каких-то аспектов работы системы. Такая остановка подачи завершает все движения оси при заданном значении изменения скорости. Когда остановка подачи заканчивается, выполнение программы возобновляется. Прерванное движение восстанавливается при заданных значениях ускорения и скорости.

Остановка подачи инициируется переводом бита *Feed Hold (Остановить подачу) %Q* в состояние ON, и длится до тех пор, пока бит %Q не будет переведен в состояние OFF. Переход бита *Abort All Moves (Прекратить все движения) %Q* в состояние ON или появление ошибки, которая вызывает ошибку останова, заканчивает остановку подачи, а также завершает программу. При действии остановки подачи допускаются положительные и отрицательные толчки, но никаких других движений. Когда остановка подачи закончена, и выполнение программы возобновлено, то модуль DSM314 извлекает из памяти действовавшие установки и выполняет передвижение на них.

### Пример 16: Остановка подачи

Приведенный ниже пример показывает график движения при действии остановки подачи. Показан линейный разгон до заданной скорости при заданном ускорении. В момент времени, показанный штриховой линией, начинает действовать остановка подачи; скорость уменьшается до нуля при заданном значении замедления. Затем выполняется толчок с использованием бита *Jog Minus (Толчок минус) %Q*. Это очевидно, поскольку скорость толчка отрицательная. Следует обратить внимание на то, что ускорение, используемое при толчке, является текущим *Jog Acceleration (Ускорение передвижения)*, которое отличается от запрограммированного ускорения. Следует отметить также, что команда *Feed Hold (Остановить подачу) %Q* должна действовать в течение всего толчка. После прекращения движения толчка заканчивается остановка подачи, и программа продолжается до завершения.

ACCEL 1000  
VELOC 2000  
PMOVE 12000, INCR, LINEAR

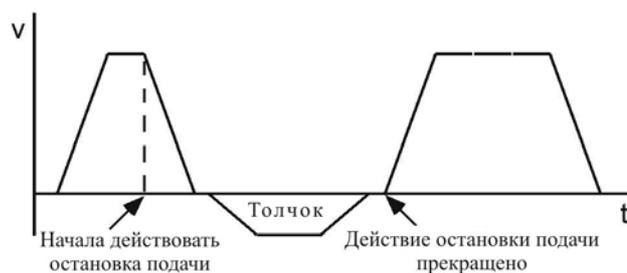


Рис. 7-20. Пример остановки подачи

## Подстановка скорости подачи

Некоторые применения требуют проведения небольших изменений запрограммированной скорости, чтобы реагировать на внешние воздействия. Команда прямого действия *Rate Override (Подстановка скорости) %AQ*, направляемая на DSM через ladder logic, позволяет выполнять изменения запрограммированной скорости подачи (скорости) во время выполнения программы (Более детальная информация о команде *Rate Override (Подстановка скорости)* находится в главе 5). Когда начинается выполнение программы, величина коррекции всегда устанавливается на 100%. Таким образом, изменения скорости подачи, сделанные до того, как бит выполнения программы перешел в состояние ON, игнорируются. Однако коррекция скорости, выданная в том же цикле, что и бит выполнения программы, *начинает* действовать.

Значение коррекции скорости подачи может меняться от 0 до 120%. Когда выдается команда *Rate Override*, то модуль DSM314 умножает процент коррекции на значение запрограммированной скорости, чтобы получить новую скорость. Если ось движется, то для изменения ее скорости на новое значение используется текущий *Jog Acceleration Mode (Режим ускорения передвижения)*. Введенное изменение скорости подачи влияет на все дальнейшие значения скорости движения. Следует отметить, что если задана скорость подачи 0%, то движение отсутствует до тех пор, не будет задано новое значение скорости подачи. Также следует обратить внимание, что бит *Moving (Движение) %I* остается в состоянии ON, когда скорость подачи 0%.

Подстановка (коррекция) скорости не действует на не программные движения, такие как *Jog (Толчок)*, *Find Home (Начальное позиционирование)* или *Move at Velocity (Движение на скорости)*.

### Пример 17: Подстановка (коррекция) скорости подачи

При выполнении данной программы выдаются изменения скорости + или -10%. Пунктирные линии показывают -10%, штриховые линии показывают +10%.

```
ACCEL 1000
VELOC 6000
PMOVE I10000, INCR, LINEAR
```

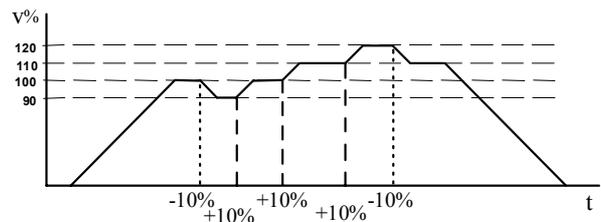


Рис. 7-21. Пример подстановки (коррекции) скорости подачи

## Многоосное программирование

Для синхронизации команд движения осей в точках, где согласование по времени является критическим, в многоосных программах могут быть использованы блоки Sync (Синхронизации).

### Пример 18: Многоосное программирование

В данном примере принимается, что ось 1 управляет вертикальным перемещением, а ось 2 – горизонтальным. Задачей данного движения является максимально быстрое перемещение предмета из точки А в точку Е, обходя препятствие, которое мешает прямому передвижению из точки в точку.

Наиболее простым путем было бы перемещение из точки А в точку С, а затем из С в Е. Однако этот путь ведет к потере времени. Более подходящим представляется начать горизонтальное движение до достижения точки С. Можно видеть, что когда ось 1 передвинется на позиции 30 000 пользовательских единиц (точка В), то ось 2 может начать двигаться так, чтобы не задеть препятствие. Этот сегмент программы можно записать следующим образом:

```
10: CMOVE  AXIS1 30000, INCR, LINEAR
20: SYNC
   PMOVE  AXIS1 50000, INCR, LINEAR
   PMOVE  AXIS2 120000, INCR, LINEAR
```

Когда выполняется блок 10, ось 1 перемещается на 30 000 единиц, тогда как ось 2 в это время стоит. Когда данное движение оси 1 заканчивается, происходит следующее: ось 1 начинает перемещение на 50 000 единиц по команде PMOVE из блока 20 (SYNC) без остановки (поскольку первое движение было CMOVE), а ось 2 начинает свое перемещение на 120 000 единиц. На рисунке внизу видно, что первое движение оси 1 перемещает предмет из точки А в точку В. В точке В ось 1 продолжает движение (выполняя свое второе движение), и начинается движение оси 2, что перемещает предмет в точку D. Ось 1 завершает свое второе движение в точке D и останавливается; однако ось 2 продолжает двигаться и перемещает предмет в точку Е.

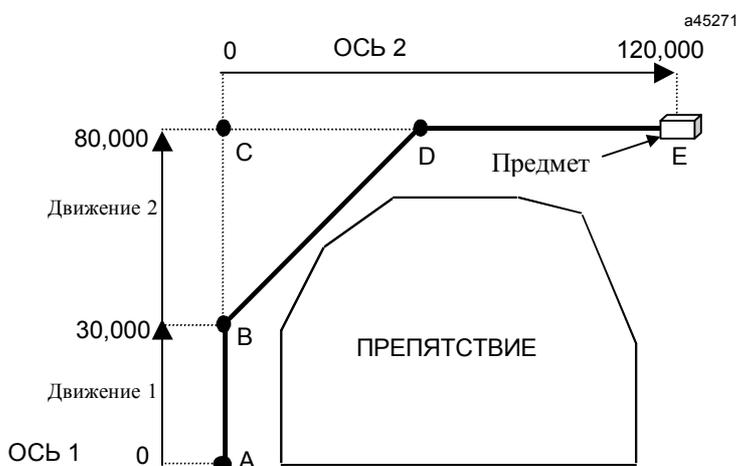


Рис. 7-22. Пример многоосного программирования

Если данный сегмент программы находится не в начале программы и если по каким-то причинам ось 2 еще не достигла блока 20, когда ось 1 переместилась на 30 000 импульсов, то возникает ситуация сбоя. Ось 1 переместится на 80 000 импульсов, а модуль DSM314 передаст в код статуса сообщение "Ошибка синхронизации блока при выполнении CMOVE".

Если синхронизация осей на блоке 20 является обязательной, то изменение блока 10 на команду PMOVE обеспечит такую синхронизацию, но тогда ось 1 остановится при значении 30 000 импульсов.

## Параметры (P0-P255) в модуле DSM314

Модуль DSM314 хранит в памяти 256 параметров (от 0 до 255), представленных двойными словами. Эти параметры могут быть использованы в качестве переменных в командах движения ACCEL, VELOC, DWELL, PMOVE и CMOVE. Следует помнить, что значения параметров имеют ограничения, и что при выходе параметра за соответствующие границы может возникнуть сбойная ситуация. Параметры 216 – 255 являются параметрами специального назначения. Некоторые из этих параметров автоматически записываются модулем DSM314. Например, P224 автоматически обновляется, когда на оси 1 появляется строб позиционирования 1. Приведенная ниже таблица описывает функции параметров специального назначения.

**Таблица 7-1. Параметры специального назначения**

Номер параметра	Функция специального назначения	Ось	Единицы
216-223	<i>Зарезервировано</i>		
224	<i>Строб 1 позиции</i>	Ось 1	Пользоват. единицы
225	<i>Строб 2 позиции</i>	Ось 1	Пользоват. единицы
226	<i>Заданная позиция в Follower Enable Trigger (Триггер включения следящего устройства)</i>	Ось 1	Пользоват. единицы
227	<i>Инкрементная дистанция останова следящего устройства</i>	Ось 1	Пользоват. единицы
228-231	<i>Зарезервировано</i>		
232	<i>Строб 1 позиции</i>	Ось 2	Пользоват. единицы
233	<i>Строб 2 позиции</i>	Ось 2	Пользоват. единицы
234	<i>Заданная позиция в Follower Enable Trigger (Триггер включения следящего устройства)</i>	Ось 2	Пользоват. единицы
235	<i>Инкрементная дистанция останова следящего устройства</i>	Ось 2	Пользоват. единицы
236-239	<i>Зарезервировано</i>		
240	<i>Строб 1 позиции</i>	Ось 3	Пользоват. единицы
241	<i>Строб 2 позиции</i>	Ось 3	Пользоват. единицы
242	<i>Заданная позиция в Follower Enable Trigger (Триггер включения следящего устройства)</i>	Ось 3	Пользоват. единицы
243	<i>Инкрементная дистанция останова следящего устройства</i>	Ось 3	Пользоват. единицы
244-247	<i>Зарезервировано</i>		
248	<i>Строб 1 позиции</i>	Ось 4	Пользоват. единицы
249	<i>Строб 2 позиции</i>	Ось 4	Пользоват. единицы
250	<i>Заданная позиция в Follower Enable Trigger (Триггер включения следящего устройства)</i>	Ось 4	Пользоват. единицы
251	<i>Инкрементная дистанция останова следящего устройства</i>	Ось 4	Пользоват. единицы
252-255	<i>Зарезервировано</i>		

Значения всех параметров сбрасываются в ноль при выключении-включении или после того, как конфигурация DSM314 сохранена в ПЛК. Есть три пути задать значения параметрам:

- Команда LOAD (ЗАГРУЗИТЬ) программы движения.
- Команда *Load Parameter Immediate (Загрузить параметр немедленно) %AQ*.
- Функциональный блок COMM\_REQ. Это наиболее предпочтительный путь, если требуется передать несколько параметров за один раз. Функциональный блок COMM\_REQ описан в приложении В.

Задание параметра удаляет любое его предыдущее значение. Значения параметров можно изменять по ходу выполнения программы, однако такое изменение следует предпринимать до того, как модуль DSM314 начнет выполнять команду типа 3 (Move, Wait или Dwell), предшествующую той команде типа 3, которая использует данный параметр. Это связано с предварительной обработкой команд типа 3, которая происходит в модуле DSM314. Следует отметить, что команда JUMP прекращает предварительную обработку и вызывает обработку команд программы в точке назначения перехода.

Ниже дан пример программы движения, использующей параметры. Значения параметров 1 – 5 загружены предварительно командой COMREQ из ПЛК, по крайней мере, за два блока программы до их использования.

(Напоминаем, что "блоки программы" не то же самое, что секции программы движения, которые отмечены командой BLOCK #).

Блок / Команда / Данные	Комментарии
1: VELOC P001	// Задается скорость первого движения = значение в параметре 1
ACCEL P002	// Задается ускорение первого движения = значение в параметре 2
CMOVE P003, ABS, LINEAR	// Достигается скорость второго движения (20000) к позиции = пар. 3
2: VELOC 20000	// Задается скорость второго движения = 20000
PMOVE 20000, INCR, LINEAR	// Нормальная команда PMOVE
DWELL P004	// Задержка на время, заданное в параметре 4
PMOVE P005, INCR, LINEAR	// PMOVE к значению в параметре 5
(*Строб #1 появляется на оси 1 во время движения к позиции, заданной в параметре 5*)	
DWELL 1000	// Задержка на одну секунду
LOAD P006,2000	// Загрузить параметр, находящийся в параметре 6
3: PMOVE P224, INCR, LINEAR	// Двигаться на позицию, захваченную стробом #1 на оси 1
DWELL 2000	// Задержка на две секунды
PMOVE P006, ABS, S-CURVE	// Конечная позиция останова = значение в параметре 6

## Расчет значений ускорения, скорости и позиции

Одним из методов определения значения переменных для программ движения АРМ или DSM, таких как *Acceleration (Ускорение)*, *Velocity (Скорость)* или *Position (Позиция)*, является представление требуемого движения или участка движения в виде графика скорости. При таком представлении время откладывается по горизонтальной оси, а скорость – по вертикальной. Ключом к пониманию создания графика является разбиение всего движения на небольшие сегменты, которые можно проанализировать геометрически. Большинство применений использует экономичные трапециевидные графики скорости, показанные далее. Для максимально быстрого передвижения используются треугольные графики скорости, если сервоустройства имеет достаточный диапазон скорости. При треугольном движении в течение половины расстояния действует ускорение, а в течение оставшейся половины – торможение. Другой альтернативой является использование трапециевидного графика с укороченным сегментом поворота.

### Кинематические уравнения

Кинематика – это направление механики, которое изучает движение тела или системы тел без учета их массы или действующих на них сил. Приведенная ниже таблица показывает преобразования основных линейных уравнений, применяемых к участку ускорения графика движения. Следует использовать эти формулы для расчета скорости и ускорения на участке изменения скорости при движении.

Таблица 7-2. Преобразования линейных уравнений

<del>Пересчет для</del>	Дано	A, X	A, V	A, t	V, t	V, X	X, t
<i>Acceleration (Ускорение)</i>					V/t	V <sup>2</sup> /2X	2X/t <sup>2</sup>
<i>Velocity (Скорость)</i>		$\sqrt{2AX}$		At			2X/t
<i>X (Расстояние)</i>			V <sup>2</sup> /2A	At <sup>2</sup> /2	Vt/2		
<i>t (время)</i>		$\sqrt{2X/A}$	V/A			2X/V	

На рис. 7-23 дан пример трапециевидного движения. Начиная с нулевой скорости, ось ускоряется в положительном направлении ( $t_a$ ), перемещается (поворачивается) при некоторой скорости в течение некоторого времени ( $t_s$ ), а затем замедляется до нулевой скорости ( $t_d$ ). Это может быть полное движение или его часть. При рассмотрении данной фигуры легко выделяются ее различающиеся участки. **Общепринятым эмпирическим правилом является разделение трапециевидного графика движения на три участка: одна треть для ускорения, одна треть на фиксированной скорости и последняя треть для замедления.** Все передвижение разделяется следующим образом: участок равномерного движения ( $X_s$ ) составляет половину от всего трапециевидного графика скорости, а участки ускорения и замедления составляют по  $\frac{1}{4}$  от трапециевидного графика скорости. Правило "третей" минимизирует ток нагрузки двигателя и является наиболее экономичным с позиции энергопотребления.

### Трапецевидный профиль

- Ограничена максим. скоростью двигателя
- Более высокое значение ускорения чем для треугольного профиля
- Симметричный профиль (1/3, 1/3, 1/3) улучшает передачу мощности к нагрузке
- Наиболее распространенное при длительном перемещении

A = ускорение

D = замедление

X = расстояние

$V_{pk}$  = максим. скорость

$t_a$  = длительность ускорения

$t_s$  = длительность движения на пост. скорости

$t_d$  = длительность замедления

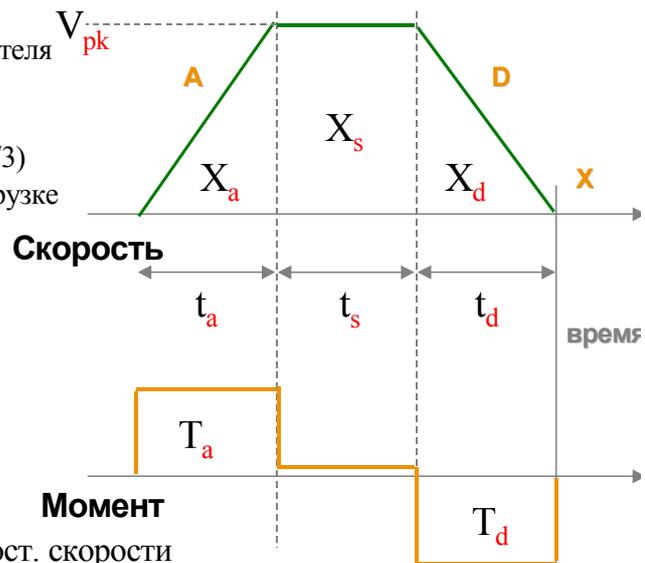
$T_a$  = момент вращения при ускорении

$T_d$  = момент вращения при замедлении

$X_a$  = участок ускорения

$X_s$  = участок постоянной скорости

$X_d$  = участок замедления



#### Уравнения

$$x = V_{pk} (0.5t_a + t_s + 0.5t_d)$$

$$V_{pk} = \frac{x}{(0.5t_a + t_s + 0.5t_d)}$$

$$a = \frac{V_{pk}}{t_a} \Rightarrow D = \frac{V_{pk}}{t_d}$$

Рис. 7-23. Трапецевидный график скорости движения

После того как контур участка движения подготовлен, требуется определить технические характеристики или физические ограничения, применимые к данному движению. Например, может оказаться необходимым выполнить движение в течение заданного интервала времени ( $t_a + t_s + t_d$ ) или на фиксированном отрезке (X). Максимальная скорость серводвигателя ( $V_{pk}$ ) при этом является примером физического ограничения. Задав любые два известных значения на участке ускорения данного сегмента движения можно определить оставшиеся переменные, используя кинематические уравнения, как показано в примере ниже.

### Пример применения с трапецевидным графиком скорости

В данном примере принимается, что в течение трех секунд должно быть выполнено полное передвижение на 16 дюймов и что максимальная скорость двигателя после редуктора составляет 16 дюймов в секунду. Используя эмпирическое правило для трапецевидного движения, делим весь временной интервал на три равных участка:  $t_a = 1$  секунда (с),  $t_s = 1$  с и  $t_d = 1$  с. Можно разделить все расстояние 16" также на три участка. Участок равномерного движения ( $X_s$ ) таким образом разделенного на равные участки трапецевидного графика скорости представляет половину расстояния движения, а участки ускорения ( $X_a$ ) и замедления ( $X_d$ ) представляют по  $1/4$  полного расстояния:  $X_a = 4"$ ,  $X_s = 8"$  и  $X_d = 4"$ .

Чтобы рассчитать максимальную скорость ( $V_{pk}$ ), следует исходить из того, что первый участок (участок ускорения) ( $X_a$ ) должен быть пройден за заданное время ( $t_a$ ). Из вышеприведенной кинематической формулы для скорости ( $2X/t$ ), используя  $X_a = 4"$  и  $t_a = 1$  с, получаем:  $(2 \cdot 4") / 1 \text{ секунду} = 8"/\text{с}$ .

Для расчета ускорения имеется простейшая формула  $(V/T) = (8"/\text{с} / 1 \text{ с}) = 8"/\text{с}^2$ .

Позиция (расстояние = X) представляет собой все расстояние передвижения ( $X_a + X_s + X_d$ ) или 16 дюймов.

### Треугольный график скорости

Треугольный график скорости позволяет уменьшить значение ускорения сервоустройства, но требует от серводвигателя более высокой скорости по сравнению с трапецевидным графиком для того же самого расстояния и времени. Следует использовать треугольный график для быстрых коротких движений.

#### Уравнения

$$x = \frac{1}{2} V_{pk} (t_a + t_d)$$

$$V_{pk} = \frac{2(x)}{(t_a + t_d)}$$

$$a = \frac{V_{pk}}{t_a}$$

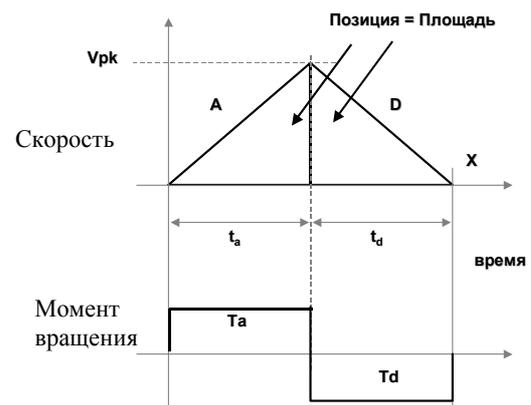


Рис. 7-24. Треугольный график скорости

### Нелинейное изменение скорости или изменение скорости по S-кривой

Расчет изменения скорости по S-кривой (такого изменения скорости, которое ограничивает рывки при движении) выполняется просто, если предварительно сделан линейный расчет. Контроллеры движения APM и DSM используют 100%-ограничение рывков. Для преобразования линейного изменения скорости в такое, которое на 100% ограничивает рывки, следует либо вдвое поднять значение ускорения ( $2 \cdot A$ ), либо вдвое увеличить время, используемое для ускорения ( $2t_a$ ). Использование S-кривой при том же значении ускорения (замедления) ( $A$ ), что и при линейном ускорении (замедлении), требует вдвое больше времени для достижения той же скорости ( $t_a$ ). Если длительность движения должна остаться без изменения, а сервопривод имеет достаточное значение пикового момента, то следует использовать удвоенное значение ускорения ( $2 \cdot A$ ), чтобы достичь заданной скорости за то же самое время.

Уравнения:

$$X_a = X_d = \frac{V_{pk}^2}{a}$$

$$t_a = t_d = \frac{2V_{pk}}{a}$$

$$t_s = \frac{X_s}{V_{pk}}$$

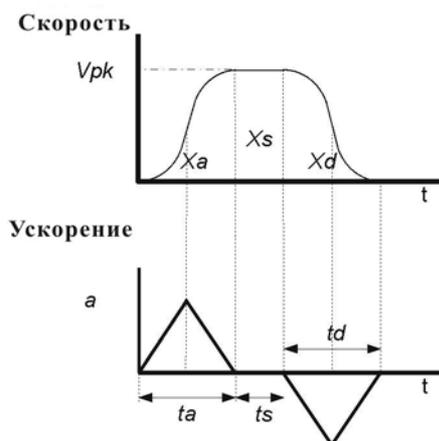


Рис. 7-25. S-кривая, ускорение

## Сообщения об ошибках программы Motion Editor и предупреждающие сообщения

Программа-редактор создает три типа сообщений об ошибках: синтаксические ошибки, семантические ошибки и предупреждения. Все они объяснены ниже.

Если исходная программа движения не имеет синтаксических или семантических ошибок, то редактор выдает только программный код. Если редактор обнаруживает синтаксические или семантические ошибки, он выдает сообщение об ошибке, которое может быть использовано для поиска ошибки в программе. На последней странице данной главы обсуждается данный предмет ("Использование сообщений об ошибках для поиска ошибок в программах движения").

Сообщения об ошибках, появляющиеся в окне статуса, содержат числовой код ошибки. Приведенный далее список содержит коды ошибок, их описания и информацию об общих причинах.

Программа Motion Editor содержит максимальные границы для позиции, скорости и ускорения на базе масштаба 8:1 пользовательских единиц на импульс.

### Синтаксические ошибки

ПО редактора программ движения транслирует программы в код, используемый модулем DSM314. Если исходный код нарушает синтаксические правила, то редактор не может распознать код и выдает сообщение о синтаксической ошибке. В этом сообщении делается попытка описать источник ошибки.

### Семантические ошибки

Этот раздел описывает синтаксические ошибки, выдаваемые синтаксическим анализатором, и их типичные причины.

#### (M200) Неопределенный идентификатор

Текстовая строка не распознана как переменная или ключевое слово программы движения.

#### (M201) Регистр параметров должен быть в диапазоне P000 – P255.

Программа движения ссылается на регистр параметров, выходящий за диапазон P000 - P255.

**(M203) STL-переменная должна быть в диапазоне CTL01 - CTL32 (DSM314)**

Программа движения ссылается на STL-бит, находящийся за пределами действующего диапазона.

**(M204) Неправильный вход программы движения**

Файл программы движения содержит неправильный символ. Файлы программы движения должны содержать только текст ASCII или пустые пространства.

**(M210) Шестнадцатеричные константы должны быть в диапазоне 16#0 - 16#FFFFFF**

Программа движения содержит шестнадцатеричное число, находящееся за пределами действующего диапазона.

**(M211) Двоичные константы должны быть в диапазоне от 0 до (2<sup>32</sup>)-1**

Программа движения содержит двоичное число, находящееся за пределами действующего диапазона. Двоичное число не может содержать более 32 двоичных разрядов.

**(M212) Целые числа должны быть в диапазоне от 0 до 4294967294**

Программа движения содержит целое число без знака, которое не может быть представлено 32 битами.

**(M213) Целые числа со знаком должны быть в диапазоне от -2147483648 до 2147483647**

Программа движения содержит целое число со знаком, которое не может быть представлено 32 битами.

**(M214) Оператор SYNC действителен только в многоосных программах и подпрограммах**

Одноосная программа или подпрограмма попыталась определить блок синхронизации.

**(M215) Многоосные программы не поддерживают ось 3 или 4**

Команды в многоосных программах могут ссылаться только на оси 1 или 2.

**(M220) Заданная ось за пределами действующего диапазона**

Одноосная программа движения может ссылаться только на ось 1, 2, 3 или 4.

**(M221) Ускорение должно быть в диапазоне 1 – 1073741823**

Команда ACCEL задала значение ускорения, находящееся за пределами действующего диапазона.

**(M222) Скорость должна быть в диапазоне 1 - 8388607**

Команда VELOC задала значение скорости, находящееся за пределами действующего диапазона.

**(M223) Позиция должна быть в диапазоне -536870912 - 536870911**

Команда CMOVE или PMOVE задала значение позиции, находящееся за пределами действующего диапазона.

**(M224) Задержка должна быть в диапазоне 0 - 60000**

Команда DWELL задала значение задержки, находящееся за пределами действующего диапазона.

**(M225) Номер блока должен быть в диапазоне 1 – 65535.**

Программа или подпрограмма движения сделала попытку задать номер блока, находящийся за пределами действующего диапазона.

**(M230) Следует задать ось в многоосной программе**

Команды ACCEL, VELOC, CMOVE, PMOVE, DWELL и WAIT должны задать ось при использовании в многоосной программе или подпрограмме.

**(M231) Невозможно задать ось в одноосной программе**

При использовании в одноосной программе или подпрограмме команд ACCEL, VELOC, CMOVE, PMOVE, DWELL и WAIT ось для них задавать *нельзя*.

**(M233) Переназначение ускорения без промежуточной команды движения**

Нельзя изменять ускорение для данной оси, если отсутствует промежуточная команда PMOVE или CMOVE.

**(M234) Переназначение скорости без промежуточной команды движения**

Нельзя изменять скорость для данной оси, если отсутствует промежуточная команда PMOVE или CMOVE.

**(M235) Номер блока уже определен в этом блоке программы**

Программа или подпрограмма движения попыталась задать номер блока, который уже задан.

**(M236) Не задан блок назначения перехода**

Программа или подпрограмма движения имеет оператор перехода JUMP на номер блока, который не задан.

**(M237) Вызов подпрограммы назначения не определен**

Программа или подпрограмма движения содержат вызов подпрограммы, которая не задана.

**(M238) Программа должна быть в диапазоне 1 - 10**

Оператор PROGRAM попытался задать номер программы, который находится за пределами действующего диапазона.

**(M239) Попытка переопределить программу. Программа уже задана.**

Оператор PROGRAM попытался задать программу, используя уже заданный номер программы.

**(M240) Конец программы задается оператором ENDPROG**

PROGRAM была завершена оператором ENDSUB, или оператор ENDSUB был установлен внутри программы.

**(M242) Отсутствует оператор ENDPROG**

До самого конца файла с PROGRAM не обнаружен оператор ENDPROG.

**(M243) Подпрограмма должна быть в диапазоне 1 - 40**

Оператор SUBROUTINE попытался задать номер подпрограммы, который находится за пределами действующего диапазона.

**(M244) Попытка переопределить подпрограмму. Подпрограмма уже задана.**

Оператор SUBROUTINE попытался задать программу, используя уже заданный номер подпрограммы.

**(M245) Конец подпрограммы задается оператором ENDSUB**

SUBROUTINE была завершена оператором ENDPROG, или оператор ENDPROG был установлен внутри подпрограммы.

**(M246) Подпрограмма не определена**

Блок программы содержит команду ENDSUB, но нет открытой SUBROUTINE.

**(M247) Подпрограмма не может вызвать сама себя**

DSM не поддерживает любой вид рекурсии. Будучи вызванной, подпрограмма не может вызвать сама себя или быть вызванной подпрограммой, которую она вызвала.

**(M248) Задание оси подпрограммы должно соответствовать вызывающей программе**

Сделана попытка вызвать одноосную подпрограмму из многоосной программы или подпрограммы или вызвать многоосную подпрограмму из одноосной программы или подпрограммы.

**(M249) Программа или подпрограмма уже задана**

Внутри не завершившейся PROGRAM или SUBROUTINE встретился оператор PROGRAM или SUBROUTINE.

**(M280) Превышено граница количества операторов, макс. 1000**

Блок программы движения может содержать не более 1000 операторов программы. Это сообщение об ошибке появляется в том случае, если число операторов превышает эту границу.

**(M281) Файл должен содержать, как минимум, одну программу**

Блок программы движения должен содержать, как минимум, одну PROGRAM; в противном случае, его невозможно вызвать. Такое сообщение об ошибке появляется, если блок программы движения не имеет ни одного оператора PROGRAM.

**(M282) Оператор должен быть в теле программы или подпрограммы**

Это сообщение об ошибке выдается, если обнаруживаются команды программы движения за пределами PROGRAM или SUBROUTINE.

**(M283) Этот оператор недействителен для заданного типа модуля**

Блок программы движения содержит оператор, недействительный для модуля назначения.

**(M293) Превышена максимальная граница счетчика числа ошибок**

Синтаксический анализатор программы движения выдает сообщения, максимум, о 30 ошибках, анализируя блок программы движения. При достижении этой границы выдается указанная ошибка, и информация о дальнейших ошибках не сообщается.

**(M300) Директивы синтаксического анализа должны предшествовать выполняемым операторам**

В начале блока программы движения должна быть задана директива #указание транслятору, т.е. она должна предшествовать любым операторам программы движения.

**(M301) Неправильное задание директивы**

Была задана неправильная директива #указание транслятору.

**(M302) Неправильный параметр директивы**

Неправильная опция была задана в качестве параметра директивы #указание транслятору.

## Предупреждения

Предупреждения выдаются для таких кодов, которые представляются неясными, но не вызывают конкретную ошибку. Этот раздел описывает синтаксические предупреждения, выдаваемые синтаксическим анализатором, и их типичные причины.

**(M482) Неожиданный конец программы: незакрытый комментарий**

Комментарий был не закрыт, когда был обнаружен конец файла.

**(M483) Вложенные комментарии**

Синтаксический анализатор движения не поддерживает вложенные комментарии. При обнаружении комментария внутри комментария выдается предупреждение.

**(M490) Программа не содержит исполняемых операторов**

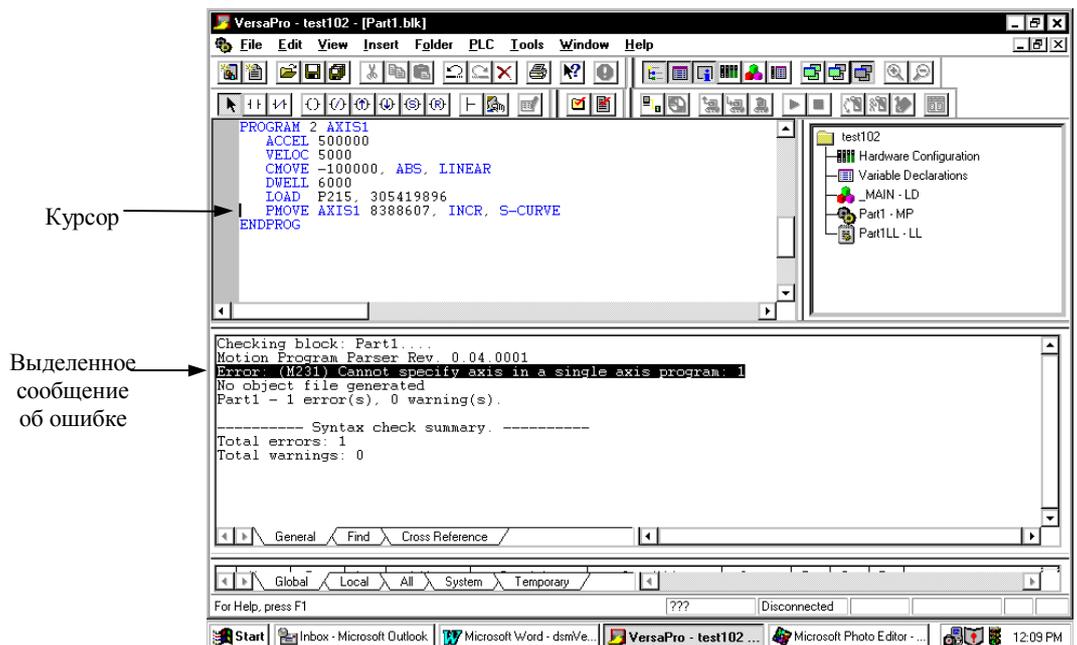
Такое предупреждение выдается, когда блок программы не содержит исполняемых операторов.

**Использование сообщений об ошибках для поиска ошибок в программах движения**

После создания в окне Motion Editor программ или подпрограмм движения можно проверить их на основные ошибки, щелкнув значок Block Check (Проверка блока) на панели инструментов.



Редактор проверяет блоки программ движения и сообщает обо всех ошибках, которые он обнаруживает, в информационном окне. Следующий рисунок показывает пример обнаружения ошибок при такой проверке.



**Рис. 7-26. Использование сообщений об ошибках для поиска ошибок в программах движения**

Ошибка, показанная на рисунке выше "Ошибка: (M231) Невозможно задать ось в одноосной программе: 1", относится к последней строке программы непосредственно перед оператором ENDPROG. Сообщение показывает, что в этой строке обнаружено AXIS1. Поскольку PROGRAM 2 является одноосной программой, то использование номера оси внутри программы не разрешается; это стало причиной выдачи сообщения об ошибке.

В примере выше по сообщению об ошибке был проведен двойной щелчок, о чем говорит тот факт, что оно негативно выделено. Если это сделано, то курсор перепрыгивает в то место программы, которое порождает ошибку. Здесь можно видеть курсор в начале строки, содержащей оператор AXIS1.

Для дальнейшей помощи при поиске ошибок см. раздел "Сообщения об ошибках программы Motion Editor и предупреждающие сообщения" данной главы, где даны наиболее распространенные причины появления различных кодов ошибок. Например, листинг для ошибки (M231), показанной в примере выше, содержит следующее:

**(M231) Невозможно задать ось в одноосной программе**

При использовании в одноосной программе или подпрограмме команд ACCEL, VELOC, CMOVE, PMOVE, DWELL и WAIT ось для них задавать *нельзя*.



Если при конфигурации модуля DSM314 задать *Follower Control Loop* (*Связанный способ управления*) = Включен (в программе конфигурации, таблица конфигурации оси), то это позволяет привязать каждую сервоось (ведомую) к ведущей оси при заданном отношении ведомая / ведущая. Модуль DSM314 задает отношение ведомая / ведущая как отношение двух целых чисел А и В. Основная формула для расчета движения слежения:

**Движение сервооси следящего устройства (ведомая ось) = Движение ведущей оси x (A/B)**

или

**отношение (ведомая : ведущая) = отношение (А : В)**

Если инициируется также команда *Jog* (*Толчок*), *Move at Velocity* (*Движение на скорости*) или *Execute Motion Program* (*Выполнить программу движения*), то движение оси представляет собой комбинацию движения ведущей оси и движения под действием внутренней команды. Данная глава представляет описание движения сервоустройства, связанного с ведущей осью. Дополнительная информация по комбинированному движению слежения и от внутренних команд см. в гл. 9.

Если бит *Enable Follower* (*Включить слежение*) %Q установлен в состояние ON, то ось немедленно начинает отслеживание выбранного *Master Source* (*Ведущий источник*), кроме того случая, когда был выбран вход *Follower Enable Trigger* (*Триггер Включения Следящего Устройства*). Если был выбран вход *Follower Enable Trigger* (*Триггер Включения Следящего Устройства*), то бит *Enable Follower* (*Включить слежение*) %Q должен быть в состоянии ON и должен произойти переход входа триггера из состояния OFF в состояние ON. Внешний вход CTL01 - CTL032 триггера выбирается в программе конфигурации.

Модуль DSM314 всегда управляет следящей осью в режиме "линейной компенсации". Если ведущая ось имеет ненулевую скорость в момент включения слежения, то ведомая ось ускоряется при сконфигурированном *Ramp Makeup Acceleration* (*Линейное Компенсирующее ускорение*) до скорости, которая позволяет начать отслеживание ведущей оси.

## Ведущие источники

Сервоось модуля DSM314 может быть сконфигурирована так, чтобы отслеживать любые два из восьми возможных входных ведущих источников. Эти два источника идентифицируются как Источник 1 и Источник 2. Бит *Follower Master Source Select* (*Выбор Ведущего Источника Следящего Устройства*) %Q определяет, какой из источников – 1 или 2 – будет активным. Допускаемые параметры для Источников 1 и 2 являются следующие:

- Ось 1 Заданная Позиция
- Ось 1 Текущая Позиция
- Ось 2 Заданная Позиция
- Ось 2 Текущая Позиция
- Ось 3 Заданная Позиция
- Ось 3 Текущая Позиция
- Ось 4 Заданная Позиция
- Ось 4 Текущая Позиция

Следует заметить, что движение слежения суммируется с *Jog* (*Толчок*), *Move at Velocity* (*Движение на скорости*) или с программами движения. Если ведомая ось отслеживает ведущий вход на скорости  $V1$ , а команда *Jog* задает скорость  $V2$ , то ось будет перемещаться со скоростью  $V1 + V2$ .

## Внешние ведущие входы

*Текущая позиция* для осей 1 – 4 представляет внешние источники ведущей оси. Датчик положения, подключенный к такой оси, или обратная связь сервосистемы может быть использована в качестве источника текущей позиции. Контур слежения модуля DSM314 позволяет ведомой оси отслеживать выбранный внешний источник, как показано в примере ниже:

### Пример 1: Отслеживание ведущего входа с текущей позицией оси 3

В данном примере зависимость скорости ( $v$ ) от времени ( $t$ ) показывает скорости ведущего входа (текущая позиция 3) и ведомой оси, которая отслеживает ведущую. Модуль DSM314 сконфигурирован так, что ***Follower Master Source 1* (Связанный мастер 1) = Actual Position 3 (Текущая позиция 3)**, а бит *Select Master Source* (*Выбрать ведущий источник*) %Q установлен в состояние OFF. Отношение A:B составляет 1:1. График скорости следящей (ведомой) оси идентичен графику ведущего входа.

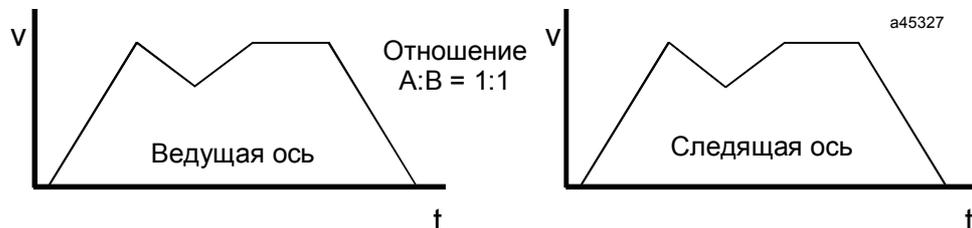


Рис. 8-1. Ведущий вход от датчика положения 3

## Внутренние генераторы команд ведущей оси

Заданная позиция для осей 1 – 4 представляет внутренние источники ведущей оси. Контур слежения модуля DSM314 позволяет ведомой оси отслеживать выбранный внутренний источник команд, как показано в примере ниже:

### Пример 2: Отслеживание внутренней ведущей команды

В этом примере ось 1 модуля DSM314 сконфигурирована так, что *Follower Master Source 2 (Связанный мастер 2) = Commanded Position 2 (Заданная позиция 2)*, а бит *Select Internal Master (Выбрать ведущий внутренний источник) %Q* установлен в состояние ON. Отношение A:B составляет 1:2. На ось 2 выдана команда *Move at Velocity (Двигаться на скорости)* 12000, а затем 0. Ось 1 отслеживает ось 2 при уменьшенных вдвое скорости и ускорении оси 2; ось 1 проходит только половину расстояния, на которое перемещается ось 2.

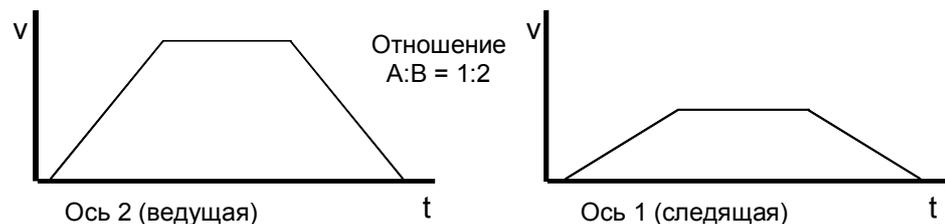


Рис. 8-2. Отслеживание датчика положения сервооси 2

## Отношение A:B

Ось модуля DSM314, отслеживающая ведущий вход, может делать это в широком диапазоне отношений ведомая / ведущая (A:B). Значением "A" может быть любое число от -32768 до 32767. Значением "B" может быть число в диапазоне от 1 до 32767. Диапазон отношения A:B может быть от 1:10 000 до 32:1. Таким образом, можно использовать очень точные соотношения, такие как, например, 12 356:12 354 или 32 000:1024.

Команда *Follower A/B Ratio (A/B Отношение Следящего Устройства) %AQ* может быть использована для изменения отношения A:B в любой момент времени, даже в процессе отслеживания. Однако неправильное отношение приведет к ошибке статуса и будет проигнорировано. Неправильным отношением является такое, в котором B равно нулю или меньше нуля или в котором отношение A:B превышает 32:1 или меньше 1:10 000.

Если отслеживание происходит при отношении, отличающемся от 1:1, то графики скоростей ведущей оси и ведомой будут выглядеть по-разному. Горизонтальные линии, показывающие постоянную скорость, и наклонные линии, показывающие ускорение и замедление, будут различными. **Если отношение A:B меньше 1:1, то скорость и ускорение следящего устройства будут меньше, чем у ведущего. Если отношение A:B больше 1:1, то скорость и ускорение следящего устройства будут больше, чем у ведущего.** Длительность движения и время, в течение которого ведомая ось ускоряется, замедляется или движется на постоянной скорости, являются одинаковыми для ведущей и ведомой осей.

Пройденное ведомой осью расстояние, которое для графика скорости представляет собой площадь участка, заключенного между графиком скорости и осью времени, получается из расстояния, пройденного ведущей осью, умножением на отношение A:B. Если A равно нулю, то движение слежения отсутствует. Если A – отрицательное, то ведомая ось двигается в направлении, противоположном по отношению к ведущей.

### Пример 3: Примеры отношений A:B

На приведенных ниже примерах показано отслеживание входа ведущего источника при различных отношениях A:B.

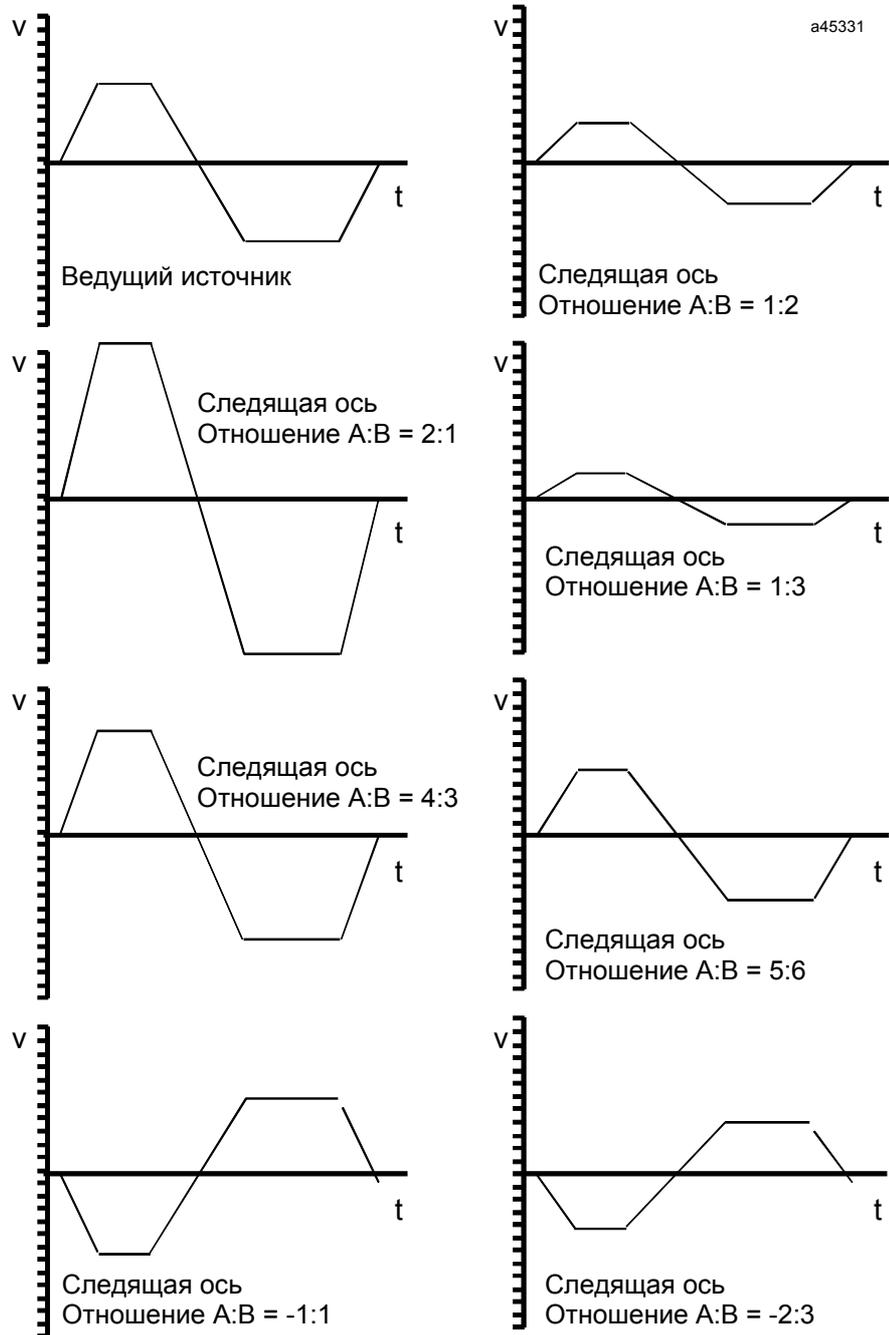


Рис. 8-3. Примеры отношений A:B

### Пример 4: Изменение отношения А:В

Одним из примеров переменного отношения А:В является использование одного отношения при положительном движении и другого – при отрицательном. Следует отметить, что задание положительной и отрицательной скорости и обновление отношения А:В должно выполняться в ПЛК или в программе Local Logic модуля DSM314. В примере ниже ведомая ось использует отношение 2:1 при положительном движении и отношение 1:2 – при отрицательном.

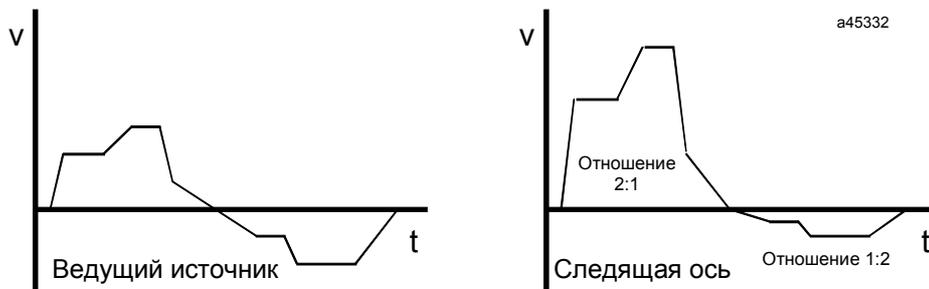


Рис. 8-4. Изменение отношения А:В

### Фиксация скорости

Фиксация скорости имеется в распоряжении при использовании установки *Velocity Limit (Ограничение скорости)* в ПО конфигурации. Если скорость ведущей оси превышает сконфигурированную границу, то ведомая ось продолжает движение при скорости, равной предельному значению, умноженному на отношение А:В. При этом происходит установка бита *Velocity Limit (Ограничение скорости) %I* и выдается ошибка статуса, которая показывает, что ведомая ось более не отслеживает позиционирование ведущего входа. Ведомая ось начинает отставать от ведущего входа.

### Пример 5: Фиксация скорости

В данном примере *Velocity Limit (Ограничение скорости)* установлена на значение 100 000. Таким образом, скорость ведомой оси ограничена значением 100 000 пользовательских единиц в секунду в любом направлении. Когда ведущая ось превышает эту границу, ведомая ось сохраняет эту скорость. Когда скорость ведущей оси снижается ниже этой границы, ведомая ось продолжает отслеживать ее скорость. Импульсы, выдаваемые в области превышения **Границы скорости**, теряются для отслеживания. На рисунке горизонтальные линии показывают границы скорости. Затененные области показывают участки, где бит *In Velocity Limit (В области ограничения скорости)* находится в состоянии ON и ведомая ось отстает от ведущей.

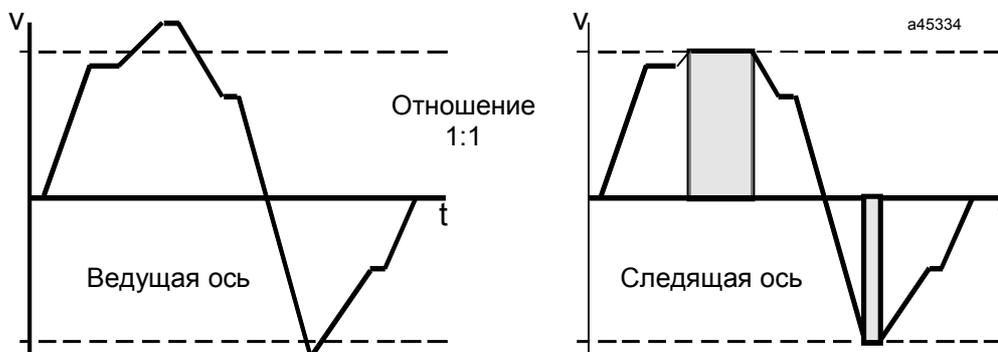


Рис. 8-5. Фиксация скорости

## Работа в одном направлении

Если при конфигурации оси задать *Command Direction (Заданное направление)* Positive Only (Только положительное) или Negative Only (Только отрицательное), то это выразится в однонаправленном движении отслеживания. Любые импульсы ведущей оси в направлении, ограниченном нулем, будут игнорироваться. При импульсах в этом направлении сообщение об ошибке не выдается. Однако бит *In Velocity Limit (В области ограничения скорости) %I* будет показывать наличие ведущей команды в направлении, ограниченном нулем.

### Пример 9: Работа в одном направлении

В данном примере при конфигурации параметр *Command Direction (Заданное направление)* установлен на Positive Only (Только положительное). Как можно видеть из графика скорости ниже, ведомая ось отслеживает положительные импульсы, но игнорирует отрицательные. Следует отметить, что когда ведущая ось движется в отрицательном направлении, бит *In Velocity Limit (В области ограничения скорости) %I* находится в состоянии ON, но ошибка статуса не возникает.

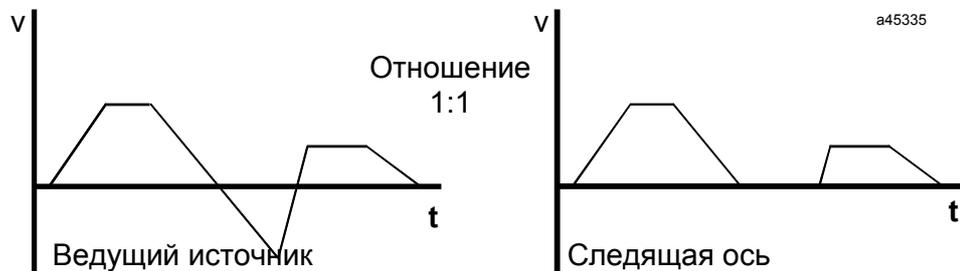


Рис. 8-6. Работа в одном направлении

## Включение слежения при помощи внешнего входа

Любой CTL-бит CTL01 - CTL32 может быть сконфигурирован как триггер включения для следящей оси. Если источник CTL-битов сконфигурирован как внешний вход лицевой панели, то этот вход может быть использован для запуска слежения. Если вход не выбран, то слежение включается и выключается непосредственно битом *Enable Follower (Включить слежение) %Q*. Если вход для триггера включения выбран и бит *Enable Follower (Включить слежение) %Q* установлен, то следующий положительный переход назначенного входа немедленно включит слежение. Слежение будет оставаться включенным до тех пор, пока не будет очищен, бит *Enable Follower (Включить слежение) %Q*. Входы 24 В лицевой панели имеют фильтры на 5 мс, что приводит к времени реакции триггера включения слежения 5 – 7 миллисекунд. Входы 5 В лицевой панели не имеют таких фильтров и обеспечивают время реакции триггера включения слежения 2 миллисекунды или меньше.

Когда срабатывает триггер *Enable Follower (Включить слежение)*, то происходит захват *Заданной позиции* в регистр параметра, который может использоваться в команде *Программируемого движения*. Позиция захватывается параметром 226 (для сервооси 1), параметром 234 (для сервооси 2), параметром 242 (для сервооси 3) или параметром 250 (для сервооси 4).

Статус *Follower Enabled (Слежение включено)* возвращается в бит %I для каждой оси.

## Выключение слежения при помощи внешнего входа

Любой CTL-бит CTL01 - CTL32 может быть сконфигурирован как триггер выключения для следящей оси. Вход триггера проверяется только в том случае, когда бит *Enable Follower (Включить слежение) %Q* находится в состоянии ON (ВКЛ.). Если бит *Enable Follower (Включить слежение) %Q* находится в состоянии ON, то переход бита триггера от OFF (ВЫКЛ.) к ON (ВКЛ.) выключает слежение. Установка бита *Enable Follower (Включить слежение) %Q* в состояние OFF (ВЫКЛ.) приводит к немедленному выключению слежения независимо от выключающей конфигурации триггера.

## Операция выключения слежения, сконфигурированная для инкрементного позиционирования

Конфигурация *Follower Disable Action (Операция выключения слежения)* для инкрементного позиционирования позволяет следящей оси выполнить **Incremental Registration Move (Движение с регистрацией приращения)**. Выключение слежения при помощи бита *Enable Follower (Включить слежение) %Q* или дополнительного *Disable Trigger (Триггер выключения)* имеет следующие последствия: ось продолжает двигаться с ее текущей скоростью, затем тормозится и останавливается после прохождения определенного расстояния). Приращение позиции определено в регистре параметров для каждой оси:

P227 = Приращение позиции по Оси 1

P235 = Приращение позиции по Оси 2

P242 = Приращение позиции по Оси 3

P250 = Приращение позиции по Оси 4

Приращение позиции представляет собой полное изменение текущей позиции с точки, когда следящее устройство было выключено, до точки, когда оно остановилось. Команды, вызывающие движения, (*Jog, Move at Velocity* или программы движения) не действуют во время *Follower Registration Move (Движение слежения с регистрацией)*.

## Управление линейным изменением скорости следящей оси

В применениях, где слежение включается, когда ведущая ось уже движется, функция *Follower Ramp (Линейное изменение скорости следящего устройства)* может быть использована для управления значением ускорения при разгоне следящей оси. Это может быть сделано без потери каких-либо импульсов ведущей оси с момента включения слежения. Во время автоматически создаваемого компенсирующего движения с линейным изменением скорости слежения значение ускорения/замедления не превышает значения, сконфигурированного в параметре *Follower Ramp Acceleration (Ускорение при линейном изменении скорости)*. Таким образом, обеспечивается плавное движение. Когда включено слежение, то ведомая ось разгоняется до скорости ведущей оси при значении ускорения, сконфигурированном в *Follower Ramp Acceleration*. Эта функция особенно полезна, когда ведущий источник движется при включении слежения. В дополнение к биту ПЛК *Enable Follower (Включить слежение) %Q* может быть сконфигурирован, CTL-бит (CTL01-CTL32) в качестве сигнала включения слежения для функций регистрации позиции. Если бит *Enable Follower (Включить слежение) %Q* находится в состоянии ON, то выбранный CTL-бит действует как триггер, срабатывающий по нарастающему фронту, чтобы включать режим слежения. После того как слежение включено, только бит ПЛК *Enable Follower (Включить слежение) %Q* управляет

активным состоянием функции слежения. Когда следящая ось включена для движения от ведущего источника, часть импульсов счета ведущего источника не может быть использована немедленно. Импульсы ведущей оси, которые накапливаются в процессе ускорения ведомой оси, запоминаются. Когда следящая ось достигает скорости ведущей оси, эти импульсы учитываются в ходе движения компенсирующей коррекции расстояния. Это движение имеет автоматически рассчитываемый трапециевидный график, определяемый параметром ***Follower Ramp Distance Makeup Time*** (***Длительность участка линейного компенсирующего изменения скорости следящего устройства***), числом накопленных импульсов и сконфигурированным значением ***Follower Ramp Acceleration*** (***Ускорение при линейном изменении скорости следящего устройства***). Следует установить требуемое значение времени в параметр ***Follower Ramp Distance Make-up Time*** (***Длительность участка линейного компенсирующего изменения скорости следящего устройства***) в программе конфигурации или изменить его при помощи команды ПЛК 42h %AQ.

Если ***Длительность участка линейного компенсирующего изменения скорости следящего устройства*** слишком мала, то автоматически создаваемый график скорости становится треугольным. Если в ходе коррекции расстояния скорость превышает 80% сконфигурированного значения ***Velocity Limit*** (***Ограничение скорости***), то автоматически рассчитанная скорость движения будет зафиксирована на 80% этого предельного значения. Фиксация скорости компенсирующего движения на 80% предельного значения дает системе некоторый резерв скорости для дальнейшего отслеживания скорости ведущего источника. В обоих случаях появляется предупреждающее сообщение. А фактическая длительность участка компенсации оказывается больше запрограммированной, однако расстояние в обоих случаях корректируется правильно.

Установка ***Follower Ramp Distance Make-Up Time*** (***Линейное компенсирующее изменение скорости следящего устройства***) на ноль позволяет разогнать ось без компенсации накопленных импульсов. В этом случае скорость ведомой оси не будет превышать скорость ведущей. Для приложений, в которых потерянные импульсы не имеют значения, следует установить длительность участка компенсации = 0.

По умолчанию накладываемый график движения, автоматически создаваемый функцией линейного изменения скорости слежения (с ненулевым временем компенсации), является трапециевидным; этот график использует значение ускорения ***Follower Ramp Acceleration*** (***Ускорение при линейном изменении скорости следящего устройства***) и значение расстояния, полученное из действующей ***Длительности интервала линейной компенсации***.

Значение ***Velocity Limit*** (***Ограничение скорости***) может влиять на функции различным образом в зависимости от соотношений скорости ведущего источника. Следующие примеры иллюстрируют данное положение.

Ситуация 1: Скорость ведущего источника менее 80% сконфигурированной ***Velocity Limit*** (***Ограничение скорости***), а длительность компенсации (Mkup Time) достаточно велика; в результате скорость остается все время меньше 80% предельного значения. Такая работа является наиболее предпочтительной; при этом не появляются сообщения об ошибках, а движение на повышенной скорости при разгоне имеет место только в пределах заданного интервала времени компенсации. Скорость следящей оси не превышает 80% границы скорости, если скорость ведущей оси не возрастает.

Ситуация 2: Скорость ведущего источника ниже 80% сконфигурированного значения ***Velocity Limit*** (***Ограничение скорости***), однако длительность интервала компенсации слишком мала, чтобы сделать возможной работу, как в ситуации 1. Когда скорость следящей оси будет соответствовать заданной скорости ведущего источника, то будет выдана ошибка только статуса (ECh). Движение компенсации будет

выполнять ускорение, используя действующее значение *Follower Ramp Acceleration* (*Ускорение при линейном изменении скорости следящего устройства*), до 80% предельного значения скорости (*Vlim*). Будет осуществляться движение компенсации, и все накопленные импульсы, сохраненные при начальном ускорении, будут использованы.

Ситуация 3: Скорость ведущего источника превышает 80% сконфигурированного значения *Velocity Limit* (*Ограничение скорости*), когда скорость следящей оси соответствует скорости ведущей. Выдается ошибка только статуса (EAh); движение компенсирующей коррекции не производится.

Ситуация 4: В тот момент, когда скорость следящей оси соответствует скорости ведущей и должно начаться компенсирующее движение и когда условия такие же, как в ситуации 1 или 2, и компенсирующее движение инициировано, скорость ведущего источника возрастает до значения >80% значения *Velocity Limit* (*Ограничение скорости*). Количество накопленных импульсов и действующее значение длительности интервала компенсации будут определять, сможет ли завершиться движение компенсации в заданное время. Если суммарная заданная ведущая скорость и скорость движения компенсации достигнет 100% предельной скорости, то будет выдана ошибка только статуса (F2h). Заданная ведущая скорость не будет превышать 100% значения *Velocity Limit* (*Ограничение скорости*). Накопленные импульсы могут быть потеряны, движение компенсации не завершится.

Индикация бита *Follower Ramp Active* (*Действует линейное ускорение / замедление следящего устройства*) %I включена, поскольку контроль участка изменения скорости действует как при нарастании скорости, так и при ее уменьшении.

Можно при помощи ПЛК отслеживать биты *Follower Enabled* (*Слежение включено*) и *Follower Ramp Active* (*Действует линейное ускорение / замедление следящего устройства*) %I, чтобы определять, какая часть цикла ускорения / замедления активна. Приведенный ниже рисунок показывает состояние *Follower Enabled* (*Слежение включено*) и *Follower Ramp Active* (*Действует линейное ускорение / замедление следящего устройства*) во время цикла слежения.

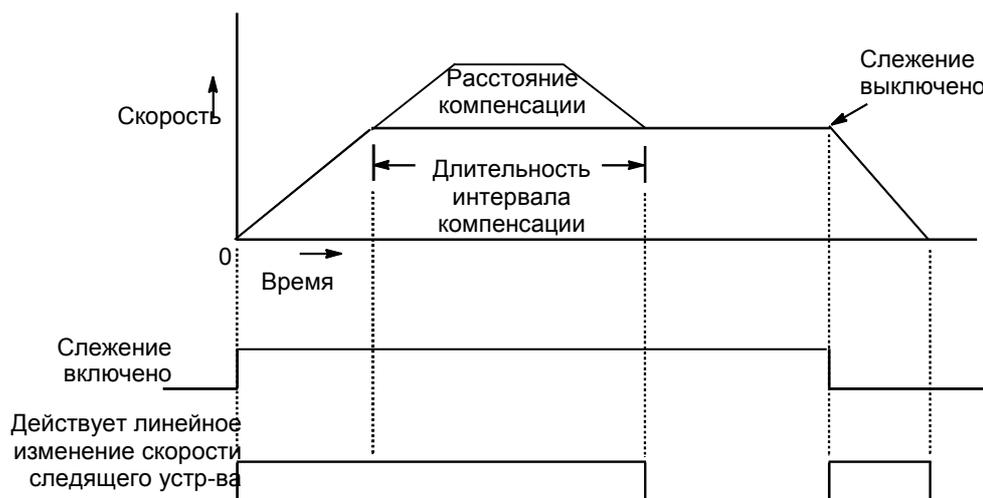
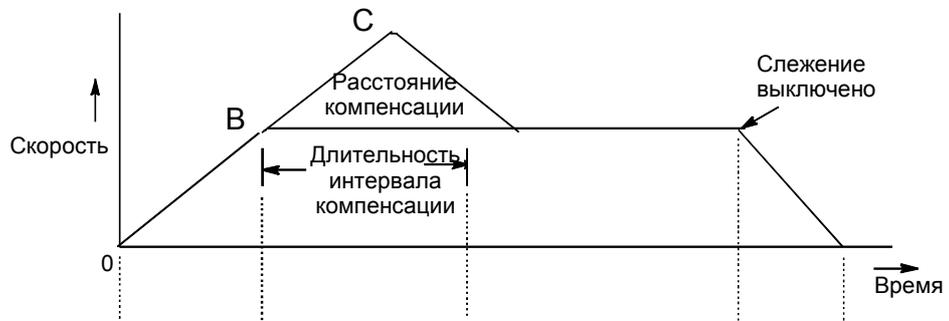


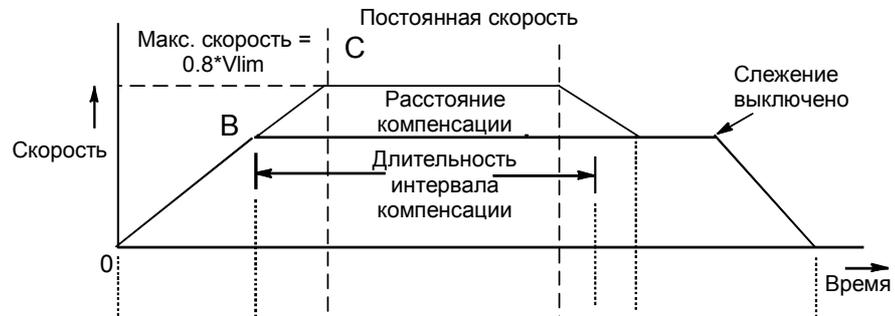
Рис. 8-7. Цикл ускорения / замедления при слежении (ситуация 2)

Запрограммированная длительность интервала компенсации может оказаться слишком малой для требуемой коррекции расстояния. В этом случае выдается предупреждающее сообщение (в точке В траектории), но система продолжает ускорение до скорости, обеспечивающей минимально возможное время коррекции расстояния. График скорости для такой ситуации показан на рис. 8-12.



**Рис. 8-8. Цикл ускорения / замедления при слежении - ситуация 2, когда длительность интервала компенсации слишком мала.**

На фазе нарастания скорости при коррекции расстояния значение скорости контролируется. Если рассчитанная скорость становится слишком большой, то ее значение фиксируется, и выдается код предупреждающего сообщения (в точке С траектории). Рис. 8-13 показывает график скорости во время такого цикла.



**Рис. 8-9. Цикл ускорения / замедления при слежении – ситуация с действующим ограничением скорости.**

Если длительность участка ускорения (сектор ВС траектории на рис. 8-13) превышает 128 секунд, то выдается другое предупреждающее сообщение. В этом случае расстояние также корректируется правильно.

### Заданный источник режима слежения и опции подключения

Рисунки далее показывают различные возможности подключения контуров ведущей оси и следящей ведомой оси.

Рисунок ниже показывает три аналоговые оси модуля DSM314, включенные в параллель с Текущей позицией для оси #4. Следует учитывать, что при такой конфигурации *может* работать функция Local Logic. Это связано с тем, что генератор команд для оси #4 не требуется для этой конфигурации. Позиции конфигурации ведущего источника все установлены на Текущую позицию оси #4. Это не является обязательным. Однако это позволяет исключить источник ошибки, если бит выбора ведущего источника установлен неправильно.

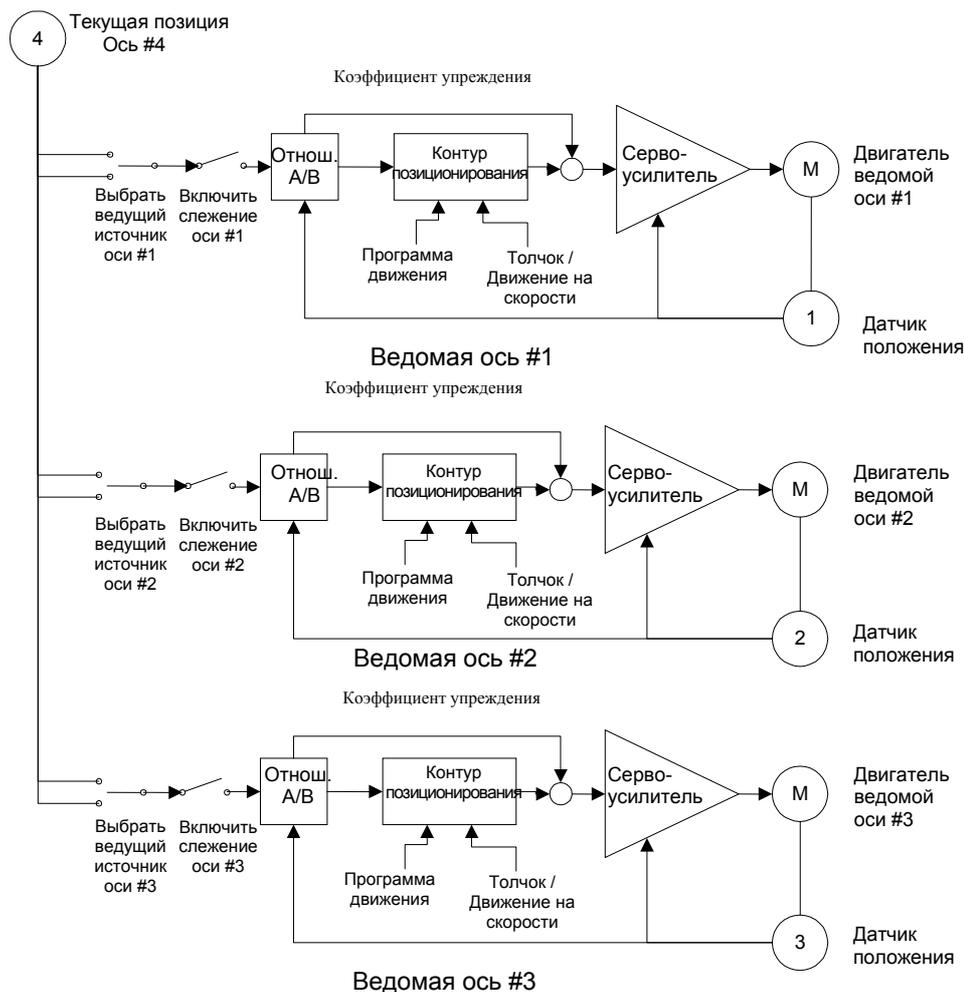


Рис. 8-10. 3-осное Аналоговое следящее устройство / Параллельная структура / Источник следящего устройства = Текущая позиция 4

Ниже рисунок показывает три аналоговые оси модуля DSM314, включенные в параллель с Заданной позицией для оси #4. Следует учитывать, что при такой конфигурации *не может* работать функция Local Logic. Это связано с тем, что генератор команд для оси #4 требуется для этой конфигурации. Позиции конфигурации ведущего источника все установлены на текущую позицию оси #4. Это не является обязательным. Однако это позволяет исключить источник ошибки, если бит выбора ведущего источника установлен неправильно.

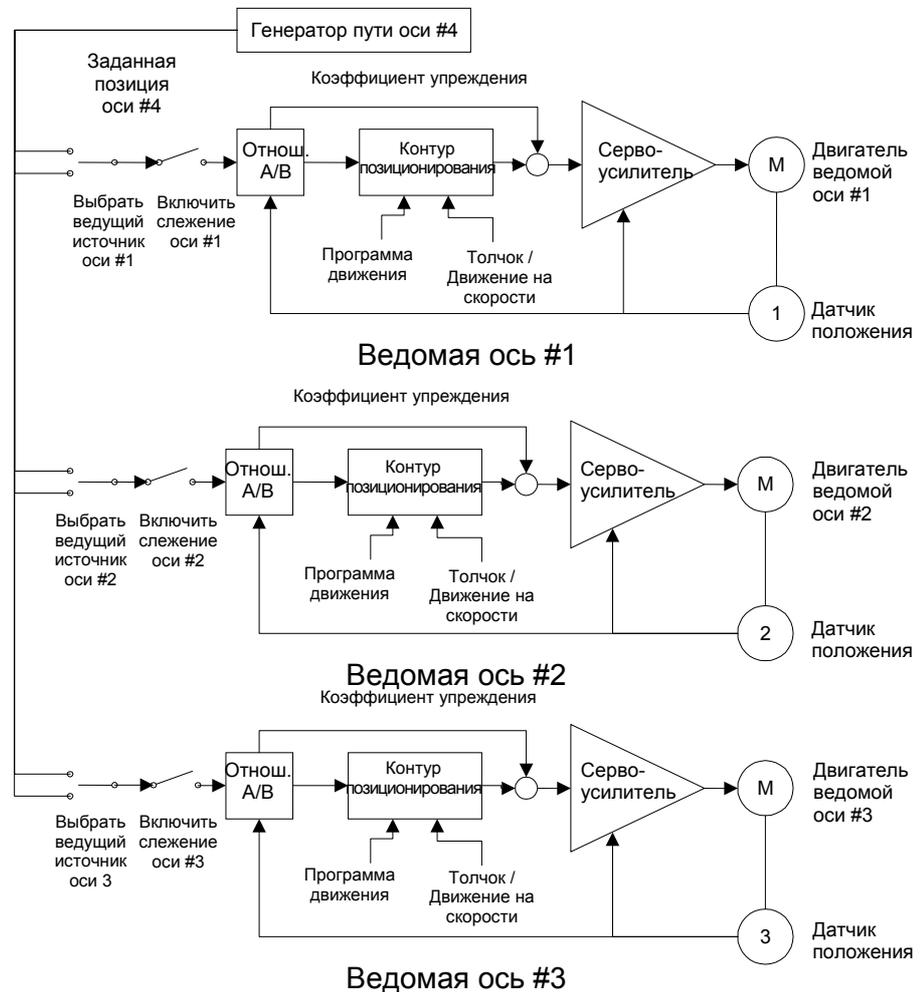


Рис. 8-11. 3-осное Аналоговое следящее устройство / Параллельная структура / Источник = Заданная позиция

Рисунок ниже показывает две цифровые оси модуля DSM314, включенные в параллель с Заданной или Текущей позицией для оси #3. Следует учитывать, что при такой конфигурации *может* работать функция Local Logic. Это связано с тем, что генератор команд для оси #4 не требуется для этой конфигурации.

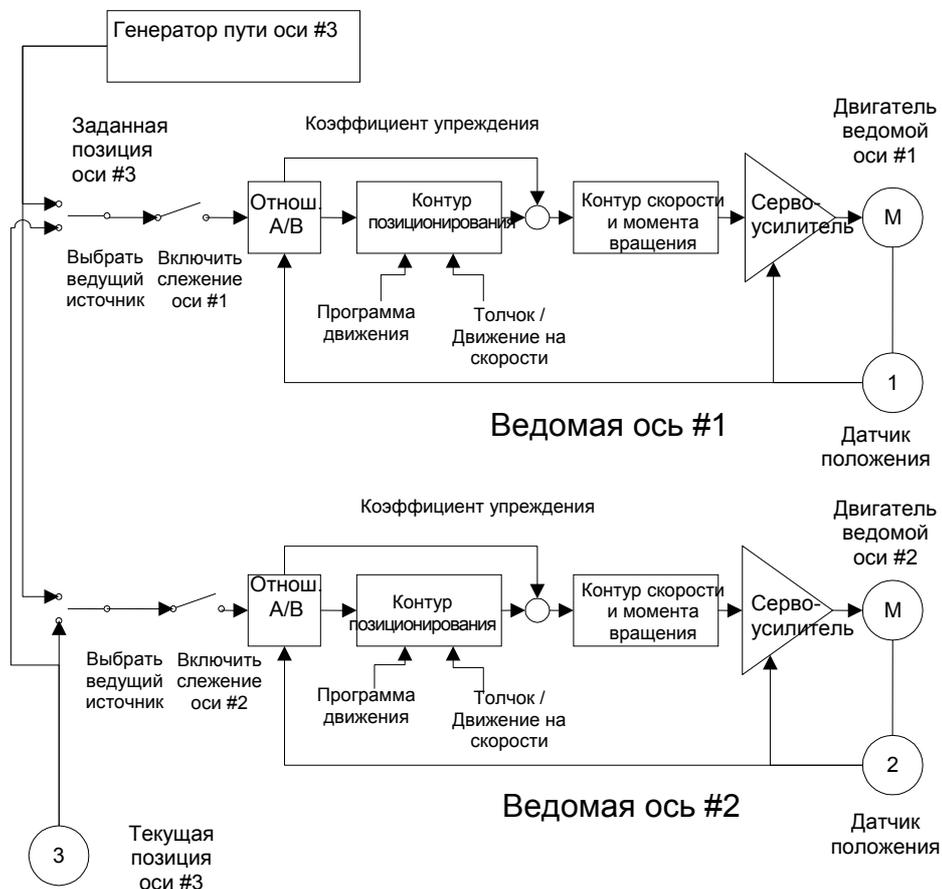


Рис. 8-12. 2-осное Цифровое следящее устройство / Параллельная структура / Источник = Заданная или Текущая позиция 3

Рисунок внизу показывает две цифровые оси модуля DSM314, подключенные в параллель с заданной позицией сервоконтролей, приводимых в действие от оси 1, для оси 1 и оси 2. Это позволяет запускать обе оси одной последовательностью команд. Следует отметить, что ось 1 сконфигурирована с ***Follower Control Loop (Связанный способ управления)*** = Выключен. Эта конфигурация не позволяет распределять нагрузку между осями, которые тесно связаны. Следует учитывать, что при такой конфигурации *может* работать функция Local Logic. Это связано с тем, что генератор команд для оси #4 не требуется для этой конфигурации.



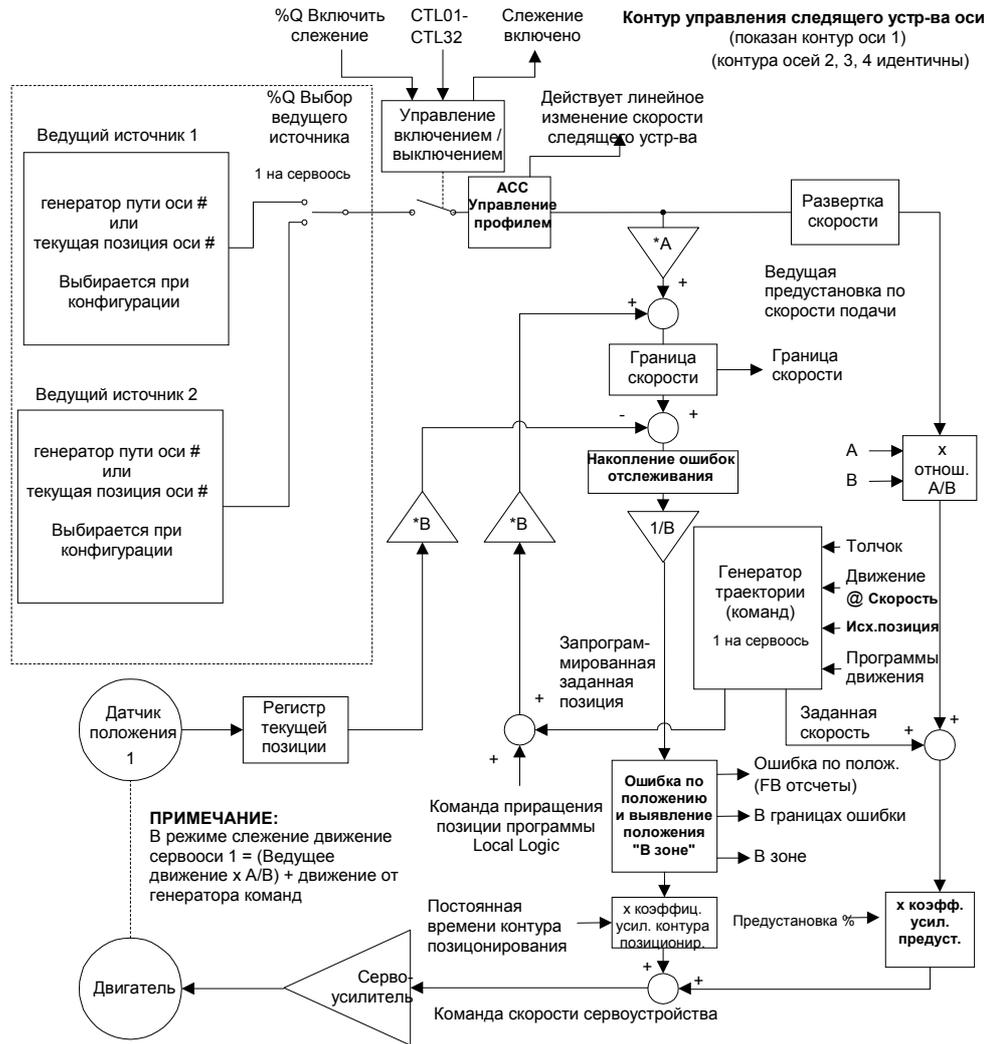
Рис. 8-13. 2-осное Аналоговое следящее устройство / Параллельная структура / Источник = Заданная позиция 1

Рисунок внизу показывает четыре аналоговые оси модуля DSM314, соединенные в две параллельные пары. Следует учитывать, что при такой конфигурации *не может* работать функция Local Logic. Это связано с тем, что для этой конфигурации для оси #4 требуется контур позиционирования сервоустройства.



Рис. 8-14. 4-осное Аналоговое следящее устройство / Параллельная структура / Источник = Заданная позиция 1 и Заданная позиция 3

**Блок-схема контура управления следящего устройства**



**Рис. 8-15. Блок-схема контура управления осью следящего устройства**

# Глава 9

## Комбинация движения слежения и движения, задаваемого командами

Комбинируемое движение представляет собой движение слежения, задаваемое ведущей осью, на которое накладывается движение, создаваемое одной из приведенных ниже внутренних команд движения:

- Команда *Jog Plus/Minus* (Толчок плюс/минус) %Q
- Команда *Move at Velocity* (Движение на скорости) %AQ
- Команда *Move* (Движение) %AQ
- Программа движения, хранящаяся в памяти

Комбинируемые движения складываются. Движение ведомой оси представляет собой сумму движения, задаваемого ведущей осью, и движения, задаваемого внутренней командой.

### Пример 1: Движение слежения в комбинации с Jog (Толчок)

В данном примере установлен бит *Enable Follower* (Включить слежение) %Q, что вызывает отслеживание ведомой осью ведущего входа. В то время как ведомая ось выполняет слежение, происходит установка бита *Jog Plus* (Толчок плюс) %Q. Следящая ось ускоряется до значения, равного скорости ведущей оси плюс текущее значение *Jog Velocity* (Скорость толчка). Величина ускорения такова, как если бы ось не отслеживала в данный момент ведущий источник. Когда бит *Jog Plus* (Толчок плюс) %Q очищается, ведомая ось затормаживается до ведущей скорости. На графике скорости внизу штриховая линия показывает точки включения и выключения бита *Jog Plus* (Толчок плюс) %Q.

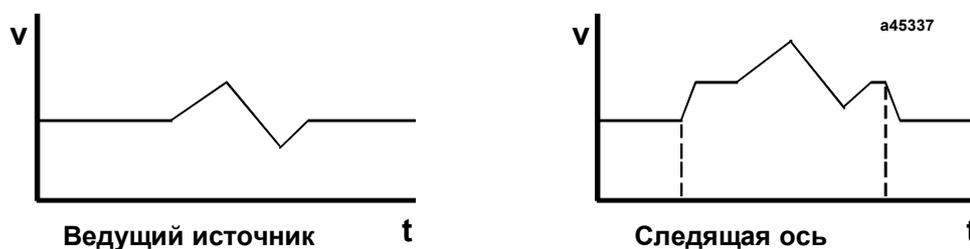


Рис. 9-1. Комбинируемое движение (слежение + толчок)

## Движение слежения в комбинации с программами движения

Для приведения в движение ведомой оси на команды ведущей оси могут накладываться движения от сохраняемых в памяти программ или команда *Move (Движение) %AQ*. Такие команды поточечного движения могут поступить с одной из хранящихся в памяти программ 1 – 10 и с любой сохраняемой в памяти подпрограммы, которую эта программа может вызвать. Команда *Move (Движение) %AQ* обрабатывается как однострочная программа движения, которая использует текущие значения **Jog Velocity (Скорость толчка)** и **Jog Acceleration (Ускорение толчка)**. Выполнение программы начинается с установки на ПЛК бита *Execute Program n (Выполнить программу n) %Q* или с посылки команды *Move (Движение) %AQ*.

Если ведущая команда отсутствует, управление осью осуществляется только данными программы движения, хранящимися в памяти. Таким образом, если отсутствует ведущий вход на сервоось 2 и если Заданная позиция 2 не выбрана в качестве ведущего источника для сервооси 1, то хранимая программа может быть использована для управления сервоосью 2 со стороны сервооси 1 в соответствии с заданным отношением.

Если при выключенном слежении выполняются команды *PMOVE*, то в конце движения должен быть установлен бит *In Zone (В зоне) %I* до того, как будет продолжено запрограммированное движение. Если слежение включено, то бит *In Zone (В зоне)* может быть не включен, в то время как ведущая команда будет отслеживаться; индикация состояния *In Zone (В зоне)* не является обязательной для продолжения. Выполнение следующей команды движения начнется после того, как будет завершено прохождение заданного расстояния предыдущего движения. Бит *In Zone (В зоне) %I* всегда показывает фактическое состояние нахождения в зоне.

Действующая заданная позиция, обновленная и используемая программой движения, хранимой в памяти, рассматривается как Программная Заданная позиция. Каждый раз, когда программа выбирается для исполнения, данный регистр позиции инициализируется одним из двух способов, указанных ниже.

1. Если слежение **не включено**, то Программная Заданная позиция устанавливается на текущее значение *Заданной позиции = Текущая позиция + Ошибка по положению*.
2. Если слежение **включено**, то Программная Заданная позиция устанавливается на Программную Начальную позицию (**0**). Поскольку Программная Заданная позиция обновляется только внутри формируемых командами (а не ведущей командой), то она будет представлять собой позицию, создаваемую хранимой программой. Команды абсолютного движения, получаемые от хранимой программы, привязываются к Программной Начальной позиции.

Диапазоны позиции (в импульсах) для регистров Текущей и Программной Заданной позиции показаны на рисунке ниже.



При длительном заданном движении в одном направлении Программная Заданная позиция может достигать значения +2 147 483 647 или -2 147 483 64 импульсов.

Однако Текущая позиция ограничивается значениями, сконфигурированными в параметрах *High Position Limit (Верхняя Граница Позиции)* и *Low Position Limit (Нижняя Граница Позиции)*.

Таблица 9-1 показывает, какие исходные команды воздействуют на эти регистры позиции и на текущие и заданные скорости. Программная Заданная позиция изменяется только под действием формируемых внутри команд (команды программ, *Jog Plus Minus (Толчок плюс минус)*, *Find Home (Начальное позиционирование)* и *Move at Velocity (Движение на скорости)*). Заданная скорость (возвращаемая в данных %AI) также показывает только ту скорость, которая задана формируемыми внутри командами. Данные %AI, возвращаемые в *Actual Position (Текущая позиция)* и в *Actual Velocity (Текущая скорость)*, отражают комбинацию ведущего входа и команд движения. Другими словами, импульсы, поступающие от ведущего источника, влияют **только** на *Actual Position (Текущая позиция)* и *Actual Velocity (Текущая скорость)*. Если формируемые внутри команды отсутствуют, то *Commanded Velocity (Заданная скорость)* будет равна 0, и Программная Заданная позиция изменяться не будет.

Таблица 9-1. Влияние входов команд на регистры позиции

Вход КОМАНДЫ	Слежение включено	Регистры слежения, на которые влияет данный вход
Ведущие команды (от выбранного ведущего источника)	Нет	Не влияет
	Да	Изменяется слово статуса <i>Actual Position (Текущая позиция)</i> %AI Изменяется слово статуса <i>Commanded Position (Заданная позиция)</i> %AI ( <i>Текущая позиция + Ошибка по положению</i> ) Программная Заданная позиция <b>не</b> меняется Изменяется слово статуса <i>Actual Velocity (Текущая скорость)</i> %AI Слово статуса <i>Commanded Velocity (Заданная скорость)</i> %AI <b>не</b> меняется
Команды программы	Нет	Изменяется слово статуса <i>Actual Position (Текущая позиция)</i> %AI Изменяется слово статуса <i>Commanded Position (Заданная позиция)</i> %AI ( <i>Текущая позиция + Ошибка по положению</i> ) Программная Заданная позиция изменяется Изменяется слово статуса <i>Actual Velocity (Текущая скорость)</i> %AI Изменяется слово статуса <i>Commanded Velocity (Заданная скорость)</i> %AI (только для скорости, задаваемой программно)
	Да	Изменяется слово статуса <i>Actual Position (Текущая позиция)</i> %AI (для программной и ведущей команд) Изменяется слово статуса <i>Commanded Position (Заданная позиция)</i> %AI ( <i>Текущая позиция + Ошибка по положению</i> ) Программная Заданная позиция изменяется (только для программной команды) Изменяется слово статуса <i>Actual Velocity (Текущая скорость)</i> %AI (для скорости, задаваемой командой программы, и скорости, задаваемой ведущей командой) Изменяется слово статуса <i>Commanded Velocity (Заданная скорость)</i> %AI (только для скорости, задаваемой программно)
Другие формируемые	Нет	Изменяется слово статуса <i>Actual Position (Текущая позиция)</i> %AI Изменяется слово статуса <i>Commanded Position (Заданная позиция)</i> %AI

внутри команды движения (Начальное позиционирование, Толчок и Движение на скорости)		(Текущая позиция + Ошибка по положению) Программная заданная позиция изменяется, но не используется. Изменяется слово статуса <i>Actual Velocity</i> (Текущая скорость) %AI. Изменяется слово статуса <i>Commanded Velocity</i> (Заданная скорость) %AI (только для скорости, задаваемой внутренней командой)
	Да (Find Home (Начальное позиционирование) выключено)	Изменяется слово статуса <i>Actual Position</i> (Текущая позиция) %AI (для внутренней и ведущей команд) Изменяется слово статуса <i>Commanded Position</i> (Заданная позиция) %AI (Текущая позиция + Ошибка по положению) Программная заданная позиция изменяется, но не используется. Изменяется слово статуса <i>Actual Velocity</i> (Текущая скорость) %AI (для скорости, задаваемой внутренней командой, и скорости, задаваемой ведущей командой) Изменяется слово статуса <i>Commanded Velocity</i> (Заданная скорость) %AI (только для скорости, задаваемой внутренней командой)

Программная Заданная позиция может быть засинхронизирована со значением Текущей позиции %AI тремя способами:

- команда *Find Home* (Начальное позиционирование) %Q
- команда *Set Position* (Установить позицию) %AQ
- команда *Execute Motion Program n* (Выполнить программу движения n) %Q (если слежение не включено).

Воздействие этих команд показано в таблице 9-2 далее.

**Таблица 9-2. Действия, влияющие на Программную Заданную позицию**

ДЕЙСТВИЕ	Слежение включено	Результирующие изменения в позиционных регистрах слежения
Начальное позиционирование выполнено	Нет	Слово статуса <i>Actual Position</i> (Текущая позиция) %AI установлено на значение Начальной позиции. Программная заданная позиция установлена на значение <i>Actual Position</i> (Текущая позиция) + <i>Position Error</i> (Ошибка по положению)
	Да	Команда <i>Find Home</i> (Начальное позиционирование) %Q не допускается. Возвращается Ошибка по положению.
Команда <i>Set Position</i> (Установить позицию) %AQ	Не применимо	Слово статуса <i>Actual Position</i> (Текущая позиция) %AI установлено на значение %AQ. Программная заданная позиция установлена на значение <i>Actual Position</i> (Текущая позиция) + <i>Position Error</i> (Ошибка по положению).
		<b>Примечание:</b> Команда <i>Set Position</i> (Установить позицию) не допускается, если бит <i>Moving</i> %I находится в состоянии ON.
Выполнить программу	Нет	Слово статуса <i>Actual Position</i> (Текущая позиция) %AI НЕ изменяется. Программная заданная позиция установлена на значение <i>Actual Position</i> (Текущая позиция) + <i>Position Error</i> (Ошибка по положению).
	Да	Слово статуса <i>Actual Position</i> (Текущая позиция) %AI НЕ изменяется. Программная заданная позиция установлена на Reference Position (Начальная позиция) (0).

Программные движения будут выполняться непрерывным образом так, что инкрементные команды PMOVE или CMOVE будут переходить через границы и продолжаться. Абсолютные команды PMOVE или CMOVE также могут быть

использованы для применений, которые не требуют выхода за пределы верхней или нижней границы импульсов.

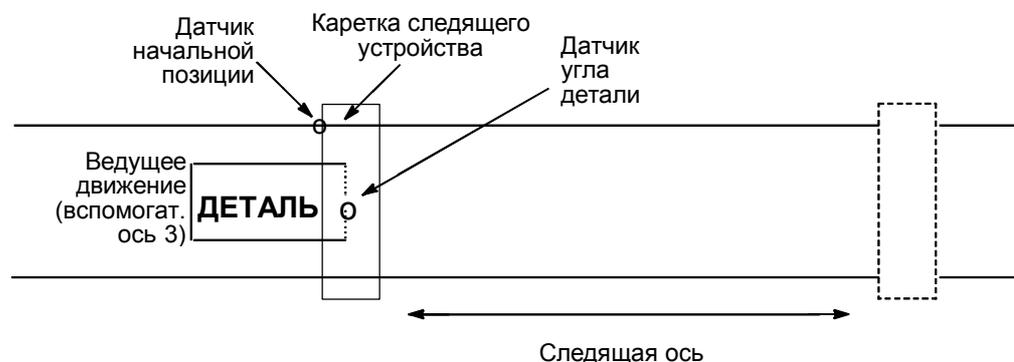
Любая команда движения, формируемая внутри, может быть немедленно завершена командой *Abort All Moves (Прекратить все движения) %Q*.

Слово статуса *User Selected Data (Данные, выбранные пользователем) %AI* может быть изменено так, чтобы получить информацию о Программной Заданной позиции при использовании команды *Select Return Data (Выбрать возвращаемые данные) %AQ*. Более детальную информацию см. в главе 5.

Приведенный далее пример применения показывает, как может быть использована программа, сохраняемая в памяти, для операций управления позиционированием, относительно обнаруженного угла перемещаемого объекта, когда он двигается со скоростью, определяемой входом датчика положения ведущей оси (вспомогательная ось 3).

## Пример 2: Движение слежения в комбинации с программой движения

Применения, которые требуют проводить операции с перемещаемыми деталями (такие операции, как надрезание, маркировка, клепка, точечная сварка, точечное приклеивание и т.п.) могли бы использовать поточечные движения, накладываемые на движение отслеживания, и включать отслеживание при определенных условиях на входе. Типичная конфигурация и последовательность управления, требуемые для таких применений, показана ниже.



### Последовательность управления:

1. При бите *Enable Follower (Включить слежение) %Q*, находящемся в состоянии ВЫКЛ., ПЛК дает команду следящей оси на выход в начальную позицию, где Текущая позиция и Программная командная позиция синхронизируются и устанавливаются на значение Начальной позиции. Бит *Position Valid (Позиция достигнута) %I* показывает, когда этот шаг завершен.
2. ПЛК устанавливает битовую команду *Enable Follower (Включить слежение) %Q*.

**Примечание:** Бит CTL01 - CTL24, к которому подключен датчик угла детали, должен быть уже сконфигурирован в параметре конфигурации *Follower Enable Trigger (Триггер Включения Следящего Устройства)*.

3. Когда происходит срабатывание датчика угла детали, модуль DSM314 включает отслеживание следящей осью входов ведущего датчика положения (вспомогательная ось 3). Бит *Follower Enabled (Слежение включено)* %I показывает, когда ось отслеживает ведущую команду. Следует отметить, что функции *Accel Ramp (Линейное изменение скорости)* и *Make-Up Time (Длительность компенсации)* могут быть использованы при необходимости, чтобы следящая ось могла догнать ведущую ось.
4. Когда слежение включено, ПЛК посылает бит *Execute Motion Program n (Выполнить программу движения n)* %Q, чтобы начать выполнение выбранной программы для следящей оси. Когда программа выбрана, Программная Заданная позиция устанавливается в программную начальную позицию (0), поскольку слежение включено. Выполнение программы затем осуществляется относительно двигающегося угла детали, т.к. следящая ось отслеживает деталь. Программная Заданная позиция теперь получает значение позиции следящей оси относительно угла детали, а Текущая позиция показывает полное расстояние, которое следящая ось прошла от Начальной точки (команды ведущего устройства +/- команды программы).
5. В конце программы ПЛК переводит бит *Enable Follower (Включить слежение)* %Q в положение OFF (ВЫКЛ.) и возвращается назад к шагу 1, чтобы повторить операции со следующей деталью.

**Примечание:** Поскольку модуль DSM314 сохранил *Заданную позицию* входного триггера включения слежения в регистре параметров (#226 для оси 1, #234 для оси 2), то в этот раз шаг 1 мог бы быть использован, чтобы выполнить другую программу с абсолютной командой движения назад к значению позиции в параметре и продолжить шаг 2. В этом случае состояние битов *Moving (Движение)* и *In Zone (В зоне)* %I может быть использовано для индикации завершения шага 1.

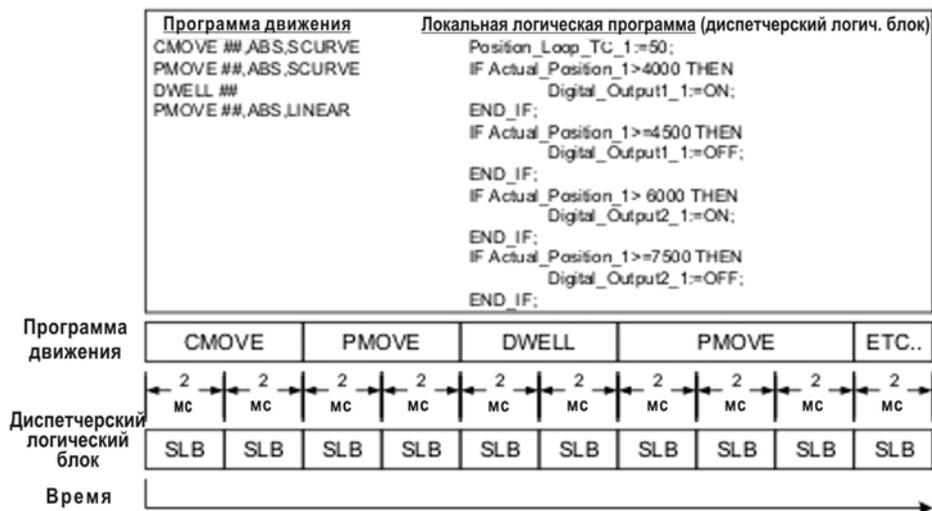
Этот метод возможен, поскольку Программная Заданная позиция устанавливается на *Actual Position (Текущая позиция) + Position Error (Ошибка по положению)*, когда запускается выполнение программы движения с выключенным слежением.

Данная глава представляет собой введение в базовые концепции локального логического программирования (local logic). Сам модуль DSM и язык программирования движения, используемый в модуле DSM, детально в данной главе не обсуждаются. Эти разделы обсуждаются в других главах настоящего руководства.

## **Введение в программирование Local Logic**

Локальная логическая программа работает совместно с логической программой ПЛК и с программой движения, создавая, таким образом, гибкую программную среду. В частности, локальные логические программы обеспечивают пользователю возможность реализовывать такую математику и логику, которая является детерминированной и синхронизированной с работой контура позиционирования модуля DSM. Эта возможность является чрезвычайно важной во многих применениях, где точность и/или скорость требуют наличия такой тесной синхронизации.

Наличие функции локальной логики модуля DSM обеспечивает пользователю возможность выполнять базовые логические и математические функции внутри DSM-модуля. Дополнительно локальная логика позволяет выполнять быстрый доступ записи / чтения к локальным цифровым и аналоговым В/В модуля DSM. Полный список доступных В/В имеется в главах 14 и 15. Метод выполнения локальной логической программы обеспечивает то, что локальная логическая программа выполняется на частоте дискретизации данного контура позиционирования и завершается на каждом периоде выборки. Примечание: Если модуль не может завершить выполнение локальной логической программы в течение заданного времени, то модуль выдает сообщение об ошибке. В главе 13 и в приложении Е дана более детальная информация о времени выполнения программы. Кроме того, локальная логическая программа работает в параллель с нормальными программами движения модуля DSM. Такая параллельная работа программ позволяет локальной логической программе контролировать программу движения. Поэтому локальные логические программы называют также диспетчерскими логическими блоками (supervisory logic block = SLB). Выполнение локальной логической программы совместно с выполнением программы движения показано на рис. 10-1.



**Рис. 10-1. Сопоставление выполнения локальной логической программы и программы движения**

Прежде чем приступить к написанию локальных логических программ, необходимо понять концепцию, демонстрируемую рисунком 10-1. Полное выполнение локальной логической программы происходит в течение каждого цикла дискретизации контура позиционирования. Программа полностью воспроизводит локальную логическую программу на каждом следующем цикле контура позиционирования. Этот метод выполнения отличается от метода выполнения программы движения. Программы движения выполняют каждую команду последовательно до ее завершения без каких-либо ограничений по времени. Эта концепция иллюстрируется таблицей 10-1, которая показывает первые четыре цикла выполнения для локальной логической программы и для программы движения, изображенных на рис. 10-1. В этом примере обратите внимание, что локальная логическая программа выполняется полностью на каждом цикле позиционирования. Операторы программы движения выполняются до тех пор, пока при движении не будет получен требуемый результат. Дополнительно по выполнению операторов программы движения см. гл. 7.

**Таблица 10-1. Пример выполнения локальной логической программы и программы движения**

Номер цикла позиционирования	Действующий оператор программы движения	Операторы локальной логической программы
n	CMOVE ##,ABS,S-CURVE	Position_Loop_TC_1:=50; IF Actual_Position_1>4000 THEN Digital_Output1_1:=ON; END IF; IF Actual_Position_1>=4500 THEN Digital_Output1_1:=OFF; END IF; IF Actual_Position_1> 6000 THEN Digital_Output3_1:=ON; END IF; IF Actual_Position_1>=7500 THEN Digital_Output3_1:=OFF; END IF;
n+1	CMOVE ##,ABS,SCURVE	Position_Loop_TC_1:=50; IF Actual_Position_1>4000 THEN Digital_Output1_1:=ON; END IF; IF Actual_Position_1>=4500 THEN

Номер цикла позиционирования	Действующий оператор программы движения	Операторы локальной логической программы
		Digital Output1 1:=OFF;
		END_IF;
		IF Actual Position 1> 6000 THEN
		Digital Output3 1:=ON;
		END_IF;
		IF Actual Position 1>=7500 THEN
		Digital Output3 1:=OFF;
		END_IF;
n+2	CMOVE ##,ABS,SCURVE	Position Loop_TC 1:=50;
		IF Actual Position 1>4000 THEN
		Digital Output1 1:=ON;
		END_IF;
		IF Actual Position 1>=4500 THEN
		Digital Output1 1:=OFF;
		END_IF;
		IF Actual Position 1> 6000 THEN
		Digital Output3 1:=ON;
		END_IF;
		IF Actual Position 1>=7500 THEN
		Digital Output3 1:=OFF;
		END_IF;
n+3	CMOVE ##,ABS,SCURVE	Position Loop_TC 1:=50;
		IF Actual Position 1>4000 THEN
		Digital Output1 1:=ON;
		END_IF;
		IF Actual Position 1>=4500 THEN
		Digital Output1 1:=OFF;
		END_IF;
		IF Actual Position 1> 6000 THEN
		Digital Output3 1:=ON;
		END_IF;
		IF Actual Position 1>=7500 THEN
		Digital Output3 1:=OFF;
		END_IF;

## Local Logic или логика ПЛК?

Язык локального логического программирования содержит базовые математические и логические конструкции. Этот аппарат не предназначен для замены логических возможностей ПЛК. Более того, локальная логика разработана для дополнения логических и математических возможностей ПЛК. В частности, локальная логика предназначена для решения небольших логических и математических задач, которые требуют тесную синхронизацию с управляемым движением. Полное выполнение локальной логической программы должно происходить в течение каждого цикла дискретизации контура позиционирования. Поэтому локальные логические программы ограничены по размерам. Граница длины локальной логической программы по умолчанию составляет 150 строк. Компоновщик Local Logic выдаст сообщение об ошибке, если будет превышена граница в 150 строк. Предупредительное сообщение выдается при превышении 100 строк. Если эта программа очень большая и требует интенсивных вычислений, то она может превысить выделенное время исполнения; сторожевой таймер выдаст предупредительное сообщение или сообщение об ошибке (см. приложение E). В отличие от этого размер программы ПЛК ограничен только имеющейся в распоряжении памятью. Однако если возрастают размеры программ ПЛК, то возрастают длительности цикла. (Дополнительную информацию по длительностям циклов ПЛК см. в "*Справочном руководстве по системе команд ЦП ПЛК Series 90-30/20/Micro*", GFK-0467). Для локальных логических программ дело обстоит иначе. Программы Local Logic всегда полностью выполняются в течение каждого цикла дискретизации контура позиционирования. Если используется ПЛК-логика, то добавленная задержка, связанная с длительностями циклов ПЛК, для ограниченных во времени логических операций, которые тесно связаны с движением, может оказаться неприемлемой или снижающей производительность процесса. Такие жестко связанные и ограниченные во времени процессы являются потенциальной областью использования программ Local Logic. Следует оценивать каждый процесс индивидуально, чтобы определить, какие разделы следует писать в ПЛК-логике, а какие – в Local Logic.

## Основы программирования программ Local Logic и движения

Приведенные далее разделы дают информацию по основам работы с редакторами программ Local Logic и движения. Данные разделы концентрируются на использовании программ с акцентом на их создание, проверку синтаксиса и загрузку программ.

### Требования

Редакторы программ Local Logic и движения являются составной частью программной среды. Для работы с ними требуется один из указанных ниже пакетов программ. Инструкции по установке находятся в документации по ПО.

- Пакет SIMPLICITY Machine Edition Logic Developer PLC, версия 2.1 или более поздняя.
- Пакет VersaPro, версия 1.1 или более поздняя.

Набор функциональных возможностей модуля DSM314 требует также наличия ЦП-микропрограммы 90-30, версия 10.0 или более поздняя.

## Создание программы Local Logic

Редактор Local Logic является составной частью программной среды. Этот редактор позволяет легко создавать, редактировать, сохранять и загружать программы Local Logic. Программа Local Logic создается в папке VersaPro или в проекте SIMPLICITY Machine Edition. Подробную информацию по тому, как создать или открыть проект, см. в документации по ПО.

### Программное обеспечение VersaPro

Детальную информацию по началу работы с ПО VersaPro см. в разделе "Конфигурация VersaPro" в главе 2.

1. Чтобы создать программу Local Logic, откройте папку VersaPro, выберите команду **File** (Файл), затем выберите пункт меню **New Motion** (Новое движение), а затем - **Local Logic**. (Рис. 10-2).

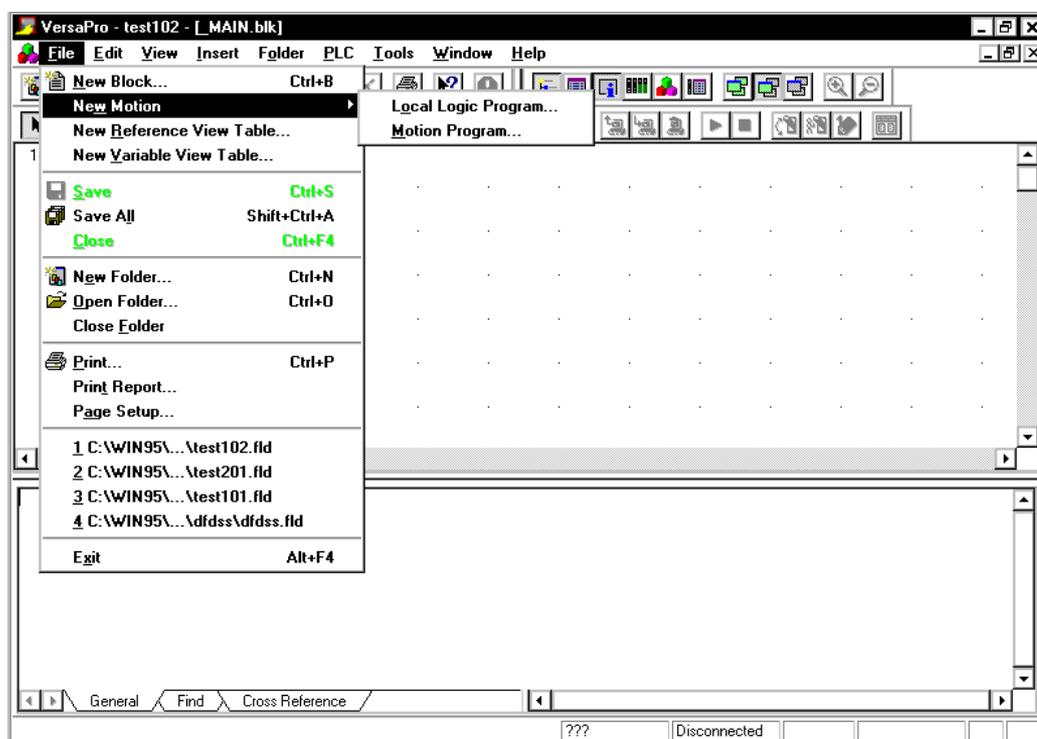


Рис. 10-2. Создание программы Local Logic в VersaPro

2. В окне диалога Create New Local Logic Program (Создать новую программу Local Logic) введите имя и, если требуется, краткий комментарий. В настоящее время Local Logic поддерживается только на модуле DSM314. Поэтому не следует изменять выбор **Motion Module Type** (Тип модуля движения), предлагаемый по умолчанию.

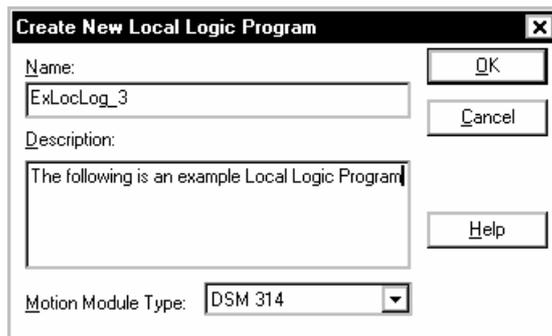


Рис. 10-3. Создание новой программы Local Logic

- Щелкните кнопку ОК, чтобы создать программу Local Logic. Появится окно редактора Local Logic.

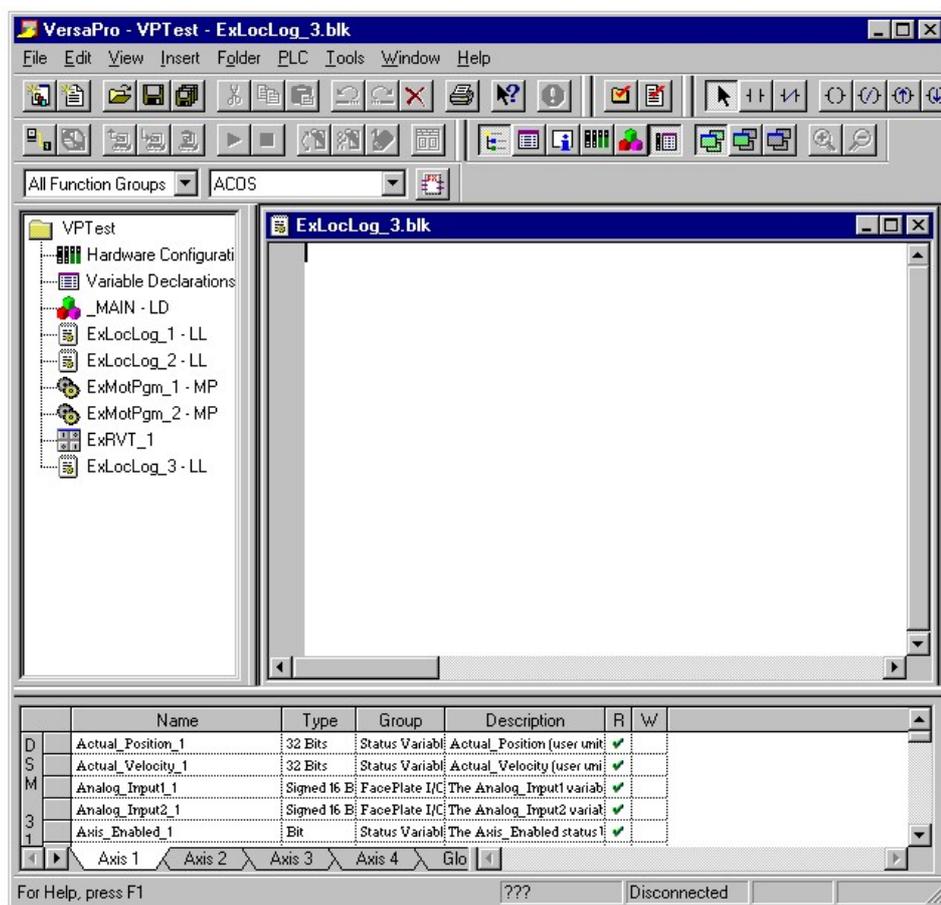


Рис. 10-4. Незаполненный редактор Local Logic

Структура редактора VersaPro / Local Logic показана на рис. 10-5.

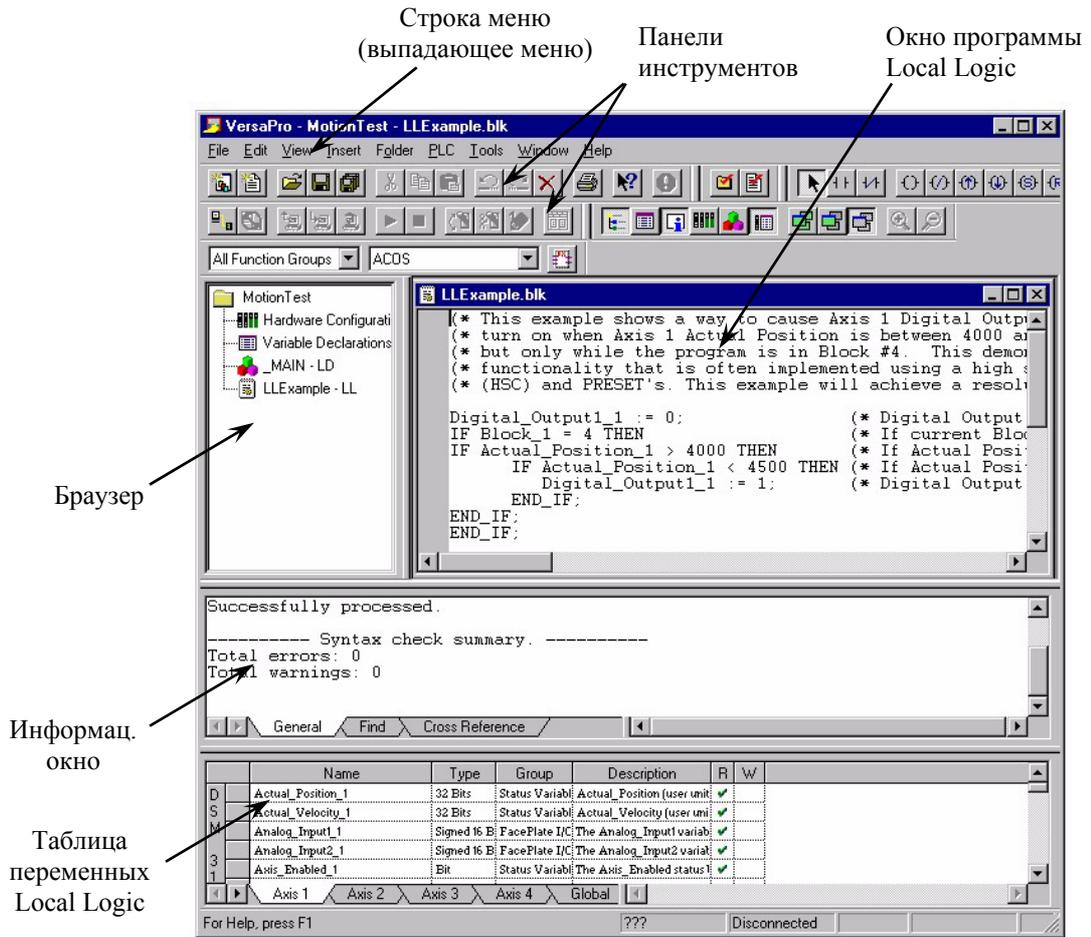
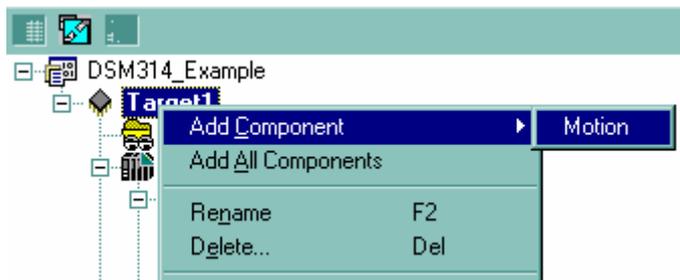


Рис. 10-5. Структура главного окна редактора Local Logic, VersaPro

## Программное обеспечение SIMPLICITY Machine Edition

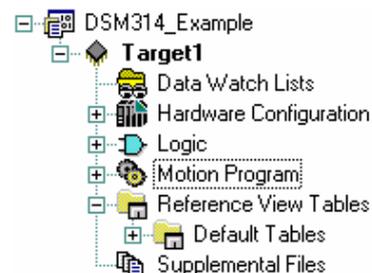
Детальную информацию по началу работы с ПО SIMPLICITY Machine Edition см. в разделе "Конфигурация SIMPLICITY Machine Edition" в главе 2.

1. Чтобы создать локальную логическую программу, откройте ваш проект в SIMPLICITY Machine Edition.
2. На закладке Project (Проект) окна Навигатора щелкните правой клавишей пункт Target (Задача) в разделе DSM314, выберите пункт Add Component (Добавить компоненту), а затем выберите Motion (Движение).



В Навигаторе появится папка Motion Program (Программа движения).

3. Откройте папку Motion Program (Программа движения). Выберите Local Logic, а затем – New (Новая). В папке Local Logic создается локальный логический блок и открывается редактор локальных логических программ.



4. Чтобы изменить название локального логического блока, отредактируйте имя в разделе Block Properties (Свойства блока), который выводится в окне Inspector (Инспектор).

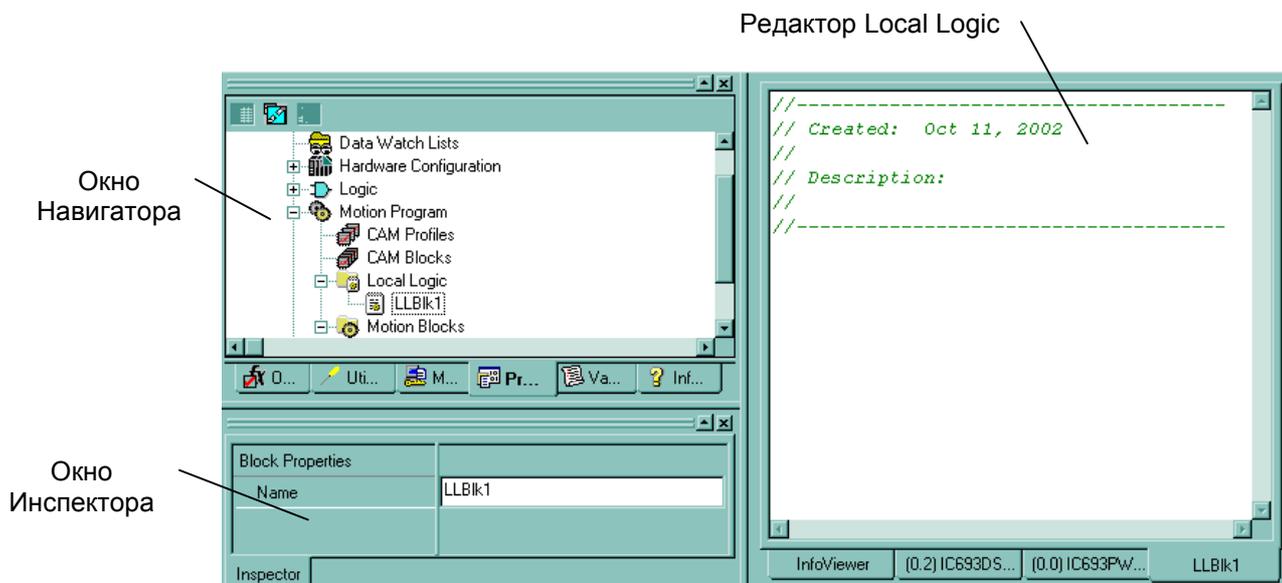


Рис. 10-6. Структура главного окна редактора Local Logic, SIMPLICITY Machine Edition

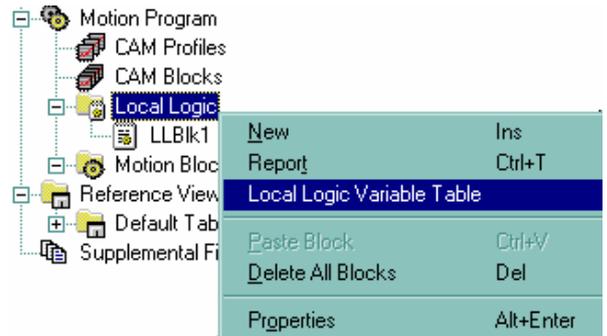
## Таблица переменных Local Logic

Программная среда включает в себя окно, которое содержит переменные Local Logic. Таблица переменных Local Logic (Local Logic Variable table = LLVT) позволяет перемещать или вырезать и вставлять текст из таблицы в программу. (См. рис. 10-7).

- Чтобы открыть таблицу LLVT в **VersaPro**, выберите пункт Local Logic Variable Table (Таблица переменных Local Logic) в команде View (Вид), нажмите Alt + 6 или щелкните кнопку Toggle Local Logic Variable Table (Включить Таблицу переменных Local Logic) на панели инструментов.



- Чтобы открыть таблицу LLVT в **SIMPLICITY Machine Edition**, щелкните правой кнопкой папку Local Logic в Навигаторе и выберите пункт Local Logic Variable Table (Таблица переменных Local Logic).



Эта таблица содержит несколько страниц-закладок, которые делят переменные на категории. Такими категориями являются следующими:

- Axis 1 (Ось 1) – Переменные, относящиеся к оси номер 1
- Axis 2 (Ось 2) – Переменные, относящиеся к оси номер 2
- Axis 3 (Ось 3) – Переменные, относящиеся к оси номер 3
- Axis 4 (Ось 4) – Переменные, относящиеся к оси номер 4
- Global (Глобальные) – Глобальные данные, такие как Module Status Code (Код статуса модуля)
- CTL Bits (Биты CTL) – Биты управления / статуса, общего назначения, модуль DSM
- Регистры параметров - Данные параметров модуля DSM

	Name	Type	Group	Description	R	W
D S M  3 1 4	Actual_Position_1	32 Bits	Status Variables	Actual_Position (user units) is a value maintained by the DSM to represent the physical position of the axis.	✓	✓
	Actual_Velocity_1	32 Bits	Status Variables	Actual_Velocity (user units/sec) represents the axis velocity derived from the position.	✓	✓
	Analog_Input1_1	Signed 16 Bits	FacePlate I/O	The Analog_Input1 variable reports the input value for the first analog input of the axis.	✓	✓
	Analog_Input2_1	Signed 16 Bits	FacePlate I/O	The Analog_Input2 variable reports the input value for the second analog input of the axis.	✓	✓
	Axis_Enabled_1	Bit	Status Variables	The Axis_Enabled status bit is ON when the DSM is ready to receive commands and the axis is enabled.	✓	✓
	Block_1	Unsigned 16 Bits	Status Variables	Block is the present command block number reported by the motion program.	✓	✓
	Commanded_Position_1	32 Bits	Status Variables	Commanded_Position (user units) is the instantaneous axis position command. The position is the sum of the position command and the position offset.	✓	✓
	Commanded_Torque_1	32 Bits	Status Variables	The Commanded_Torque variable reports the present digital servo torque command.	✓	✓
	Commanded_Velocity_1	32 Bits	Status Variables	Commanded_Velocity (user units/sec) is generated by the DSM axis command generator.	✓	✓
	Digital_Output1_1	Bit	FacePlate I/O	The Digital_Output1 bit controls the axis faceplate digital OUT_1 signal. This bit can be used to control the axis digital output.	✓	✓
	Digital_Output3_1	Bit	FacePlate I/O	The Digital_Output3 bit controls the axis faceplate digital OUT_3 signal. This bit can be used to control the axis digital output.	✓	✓
	Drive_Enabled_1	Bit	Status Variables	The Drive_Enabled status bit indicates the state of the Enable Drive %Q bit and the Drive_Enabled bit.	✓	✓
	Enable_Follower_1	Bit	Control Variables	When the Enable_Follower bit is set, motion commanded by the follower master will be executed.	✓	✓

Рис. 10-7. Таблица просмотра переменных Local Logic

Таблица имеет шесть колонок. Колонки следующие:

- **Name (Имя)** – Колонка содержит имя переменной, которое действует внутри локальной логической программы.
- **Type (Тип)** – Тип данных для этой переменной. Например, 32 Bits означает, что данная переменная является 32-битовой.
- **Group (Группа)** – Группа, в которой находится данная переменная. Например, FacePlate I/O (В/В лицевой панели) означает, что данная переменная относится к точке на лицевой панели модуля.
- **Description (Описание)** – Эта колонка содержит текстовое описание переменной. Если курсор поместить над описанием, то появится подсказка, позволяющая легко читать описание.
- **R** – Эта колонка показывает, может ли данная переменная быть считана программой Local Logic.
- **W** – Эта колонка показывает, может ли данная переменная быть записана программой Local Logic.

В VersaPro, чтобы изменить размер колонок в рассматриваемой таблице, следует поместить курсор в верхней части правого угла колонки. Курсор превратится в вертикальную полосу со стрелками. Нажмите первую клавишу мыши и перетащите ширину колонки до нужного размера.

Таблица имеет всплывающее меню, которое выводится щелчком правой клавиши мыши, когда курсор находится над таблицей. Всплывающее меню показано на рис. 10-8. Чтобы выполнить сортировку таблицы в SIMPLICITY Machine Edition, щелкните заголовок колонки, которую требуется отсортировать.

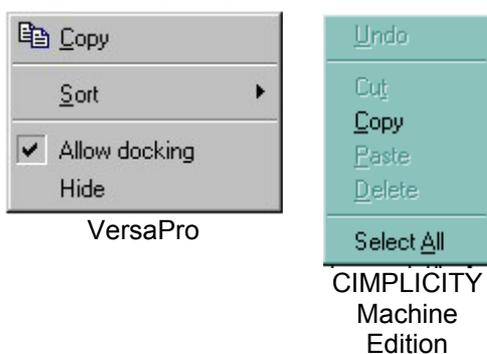


Рис. 10-8. Всплывающее меню LLVT

## Подключение редактора Local Logic к модулю DSM

Программное обеспечение конфигурации / программирования имеет несколько возможностей создания связи. Наиболее распространенным способом является прямое подключение к последовательному порту ПЛК, показанному на рис. 10-14 внизу. Имеются также опции Ethernet. Все программирование модуля DSM314 выполняется через интерфейс программного обеспечения, предоставляющий единую точку входа программирования для модуля. (Модуль DSM314 имеет также последовательный порт на лицевой панели модуля, который используется только для обновления микропрограммы модуля DSM314).

Программы Local Logic и движения сохраняются пакетом VersaPro в предназначенной для этого памяти в ЦП ПЛК. Модуль DSM314 запрашивает затем эти программы по имени из ЦП в ходе конфигурации. Ссылка на программы, которые модуль DSM314 запрашивает из ЦП, содержится в конфигурации аппаратного обеспечения для крейта ПЛК. Преимуществом такого подхода является то, что программы являются специфическими не для модуля, а для крейта / слота. Поэтому если требуется переставить модули DSM314 внутри ПЛК или заменить какой-либо модуль DSM314, то от пользователя требуется выполнить только следующие три действия: (1) снять питание с ПЛК, (2) заменить модули DSM314 и (3) снова подать питание на ПЛК. После подачи питания ПЛК направит на модули DSM314 правильные программы и установочные параметры конфигурации.

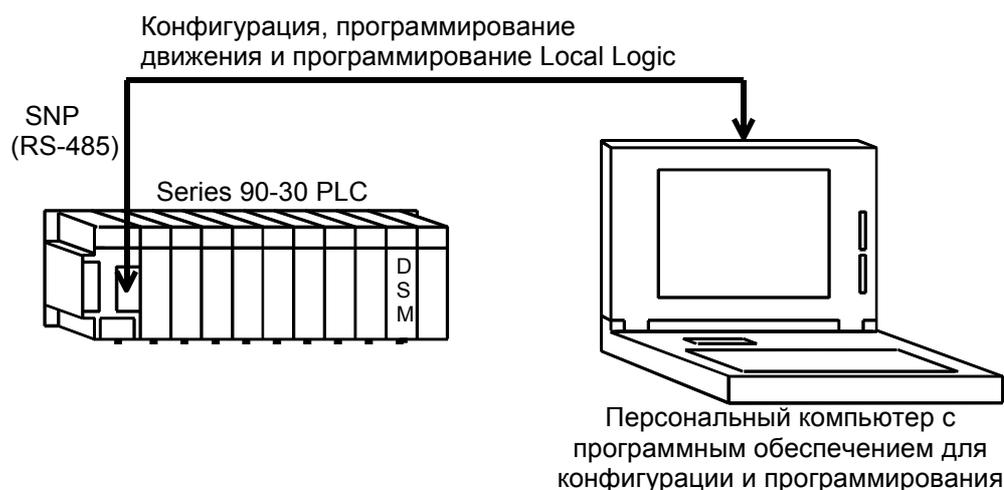


Рис. 10-9. Подключение программатора

## Построение программы Local Logic

Программное обеспечение для программирования предоставляет самодостаточную среду, которая позволяет пользователю выполнять все действия, необходимые для создания, редактирования и загрузки локальной логической программы в модуль DSM314.

### Создание программы Local Logic

Создадим программу Local Logic с именем LLExample. Подробности о том, как это сделать, см.:

"VersaPro" стр. 10-7

"SIMPLICITY Machine Edition" стр. 10-10

Получающееся окно изображенное, на рис. 10-16 и 10-17.

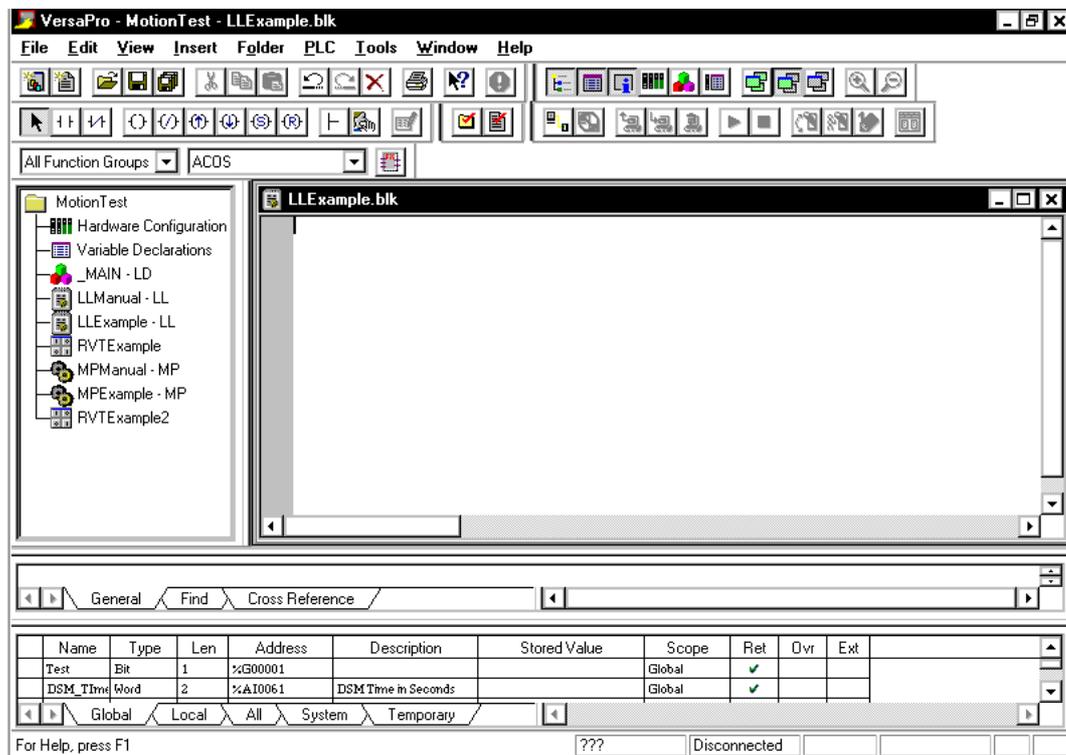


Рис. 10-10. Новая программа Local Logic в VersaPro

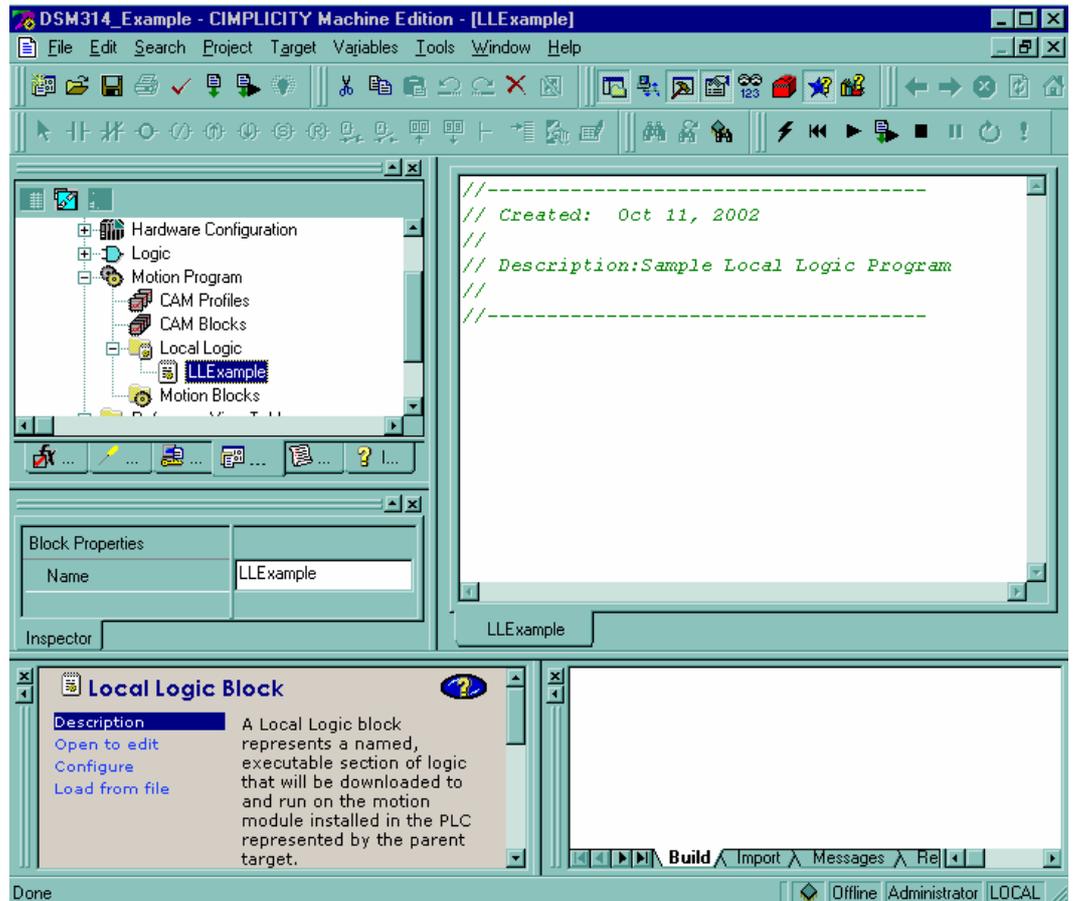


Рис. 10-11. Новая программа Local Logic в CIMPLICITY Machine Edition New

Редактор Local Logic является редактором с текстом в свободной форме; он позволяет вводить программы в том стиле, который предпочитает пользователь. Данный пример является очень простой программой Local Logic, которая не представляет собой полностью функциональное приложение, поскольку она предназначена только для целей обучения. Данный пример программы является простым таймерным применением, которое базируется на длительности цикла контура позиционирования цифрового сервоустройства (2 миллисекунды). См. главу 1 по циклам дискретизации контура позиционирования для других конфигураций.

## Пример программы Local Logic

```

(*****)
(* Название программы: LLExample *)
(* Описание: Данный пример локальной логической программы является *)
(* программой, которая создает обычный таймер. Таймер начинает *)
(* работу после того, как произойдет первый Local_Logic_Sweep *)
(* (Локальный Логический Цикл). Программа использует тот факт, что *)
(* Local Logic запускается модулем DSM каждые 2 мс. Таким образом, *)
(* программа считает локальные логические циклы. Программа имеет *)
(* три счетчика. Первый счетчик считает миллисекунды, а два других *)
(* - секунды и минуты. Счетчики секунд и минут переходят на ноль *)
(* при значении 59. Программа также устанавливает каждую секунду *)
(* бит STL01. *)
(*****)
(* Переменные *)
(* P001 = Счетчик миллисекунд *)
(* P003 = Счетчик секунд *)
(* P004 = Счетчик минут *)
(* STL01 = Сигнал секунд *)
(* P100 = Используется для проверки того, прошла ли 1 секунда *)
(* P102 = Используется для проверки того, прошла ли 1 минута *)
(*****)
IF First_Local_Logic_Sweep THEN (* Первый цикл выполнения *)
    P001 := 0; (* Установка P001 на 0 *)
    P003 := 0; (* Установка P003 на 0 *)
    P004 := 0; (* Установка P004 на 0 *)
END_IF;

P001:=P001+2; (* Время в миллисекундах *)

P100:= P001 MOD 1000; (* Проверка, прошла ли 1 секунда (1000 мс) *)
IF P100 = 0 THEN (* Если остаток от операции MOD =0, 1 секунда прошла *)
    P003:=P003+1; (* Время в секундах *)
    STL01 := 1;
    IF P003 = 60 THEN (* Если секунды = 60, то начинается с 0 *)
        P003:=0;
    END_IF;
END_IF;
IF P100 <> 0 THEN
    STL01 :=0; (* STL01=0 Если счетчик секунд не инкрементный *)
END_IF;

P101:=P001 MOD 60000; (* Проверка, прошла ли 1 минута (60000 мс) *)
IF P101 = 0 THEN (* Если остаток от операции MOD =0, 1 минута прошла *)
    P004:=P004+1; (* Время в минутах*)
    IF P004 = 60 THEN (* Если минуты = 60, то начинается с 0 *)
        P004:=0;
    END_IF;
END_IF;

```

Если ввести приведенную выше программу в текстовый редактор, то окно редактора будет выглядеть так, как на рис. 10 – 12.

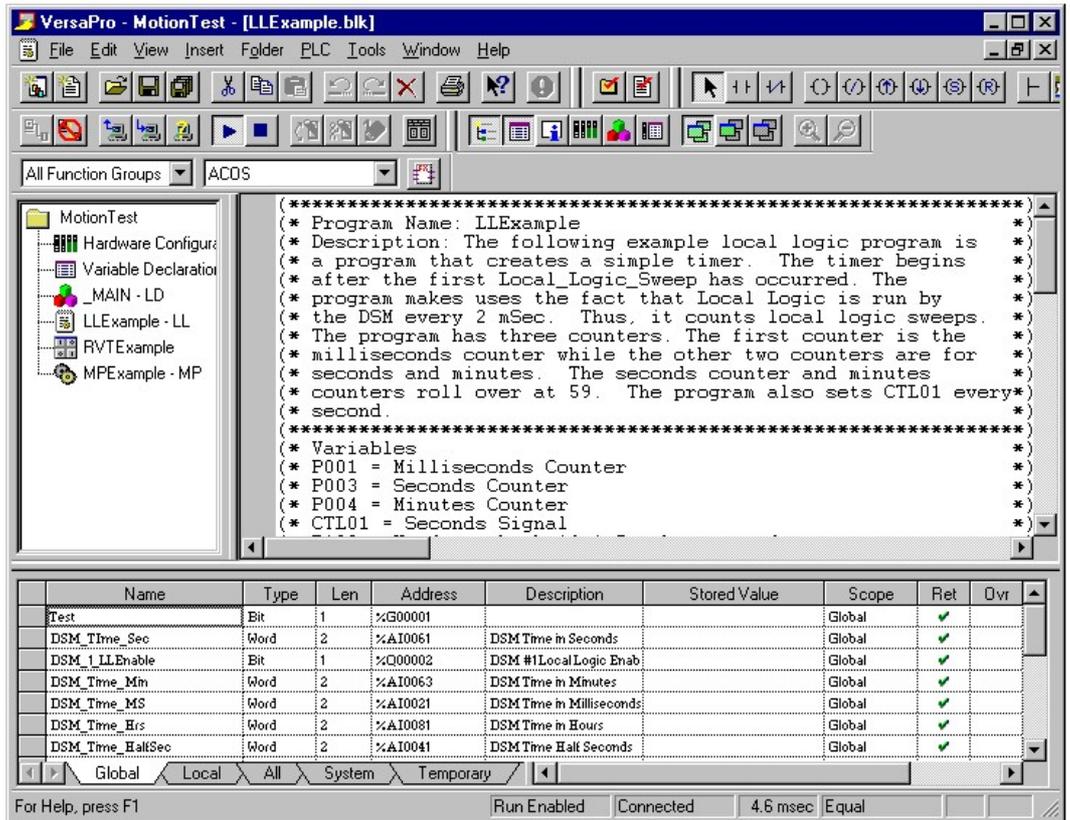


Рис. 10-12. Local Logic (LLEExample)

## Проверка синтаксиса Local Logic

Здесь проверяется программа на предмет правильности синтаксиса языка.

### VersaPro

Чтобы проверить синтаксис в VersaPro, выберите команду Folder (Папка) из главного меню, затем в подменю выберите Check Block 'LLExample' (Проверить блок "LLExample"). Это ведет к началу работы подпрограммы проверки синтаксиса на указанной программе Local Logic. **Примечание:** Чтобы проверить все блоки в папке, выберите Check All (Проверить все). (Рис. 10-13)

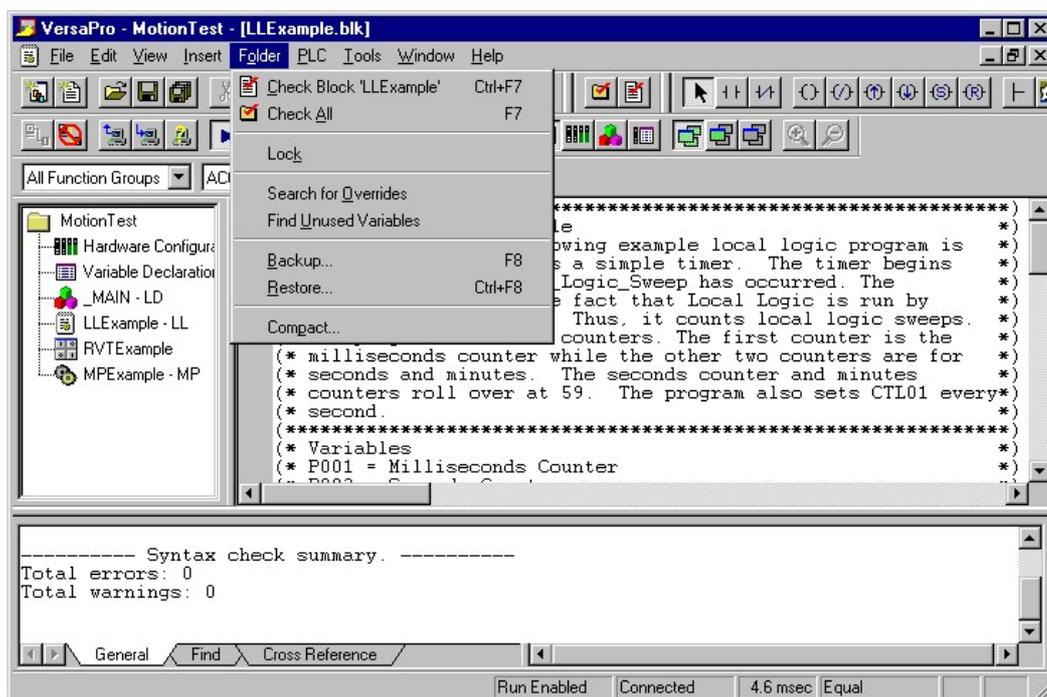


Рис. 10-13. Команда проверки синтаксиса LLExample

Проверка блоков вызывает появление информационного окна, если его не было до этого. Информационное окно может быть включено и выключено также нажатием информационной кнопки .

Информационное окно отображает результаты операции проверки синтаксиса. Если программа примера была введена правильно, то должно появиться сообщение, информирующее об отсутствии ошибок и предупреждений.

Если информационное окно показывает наличие синтаксической ошибки, то можно переместить это окно на строку, которая содержит сообщение об ошибке. После того как информационное окно установлено на нужную строку, щелкните дважды по сообщению об ошибке. В результате окно редактора автоматически переместится на строку программы, где обнаружена данная ошибка. Например, если в программе неправильно введено "First\_Local\_Logic\_Swee" вместо "First\_Local\_Logic\_Sweep", то появится сообщение о синтаксической ошибке. Оно выглядит следующим образом:

```
Error: (P200) Undefined identifier: First_Local_Logic_Swee
(Ошибка: (P200) Неопределенный идентификатор:
First_Local_Logic_Swee)
```

Можно дважды щелкнуть по этой строке в информационном окне. Редактор Local Logic автоматически перейдет на начало строки, вызвавшей сообщение об ошибке, благодаря чему пользователь может устранить ошибку, как показано на рис. 10-31.

Если выполняется активное редактирование локальной логической программы и требуется определить примерный номер строки, то следует щелкнуть вертикальный бегунок с правой стороны редактора. Появится номер строки, расположенной в самой верхней части окна. Если курсор находится несколькими строками ниже верха окна, то либо прокручивайте окно, пока текущая строка не окажется на его самом верху, либо отсчитайте количество строк от верха и сложите их с текущим номером строки.

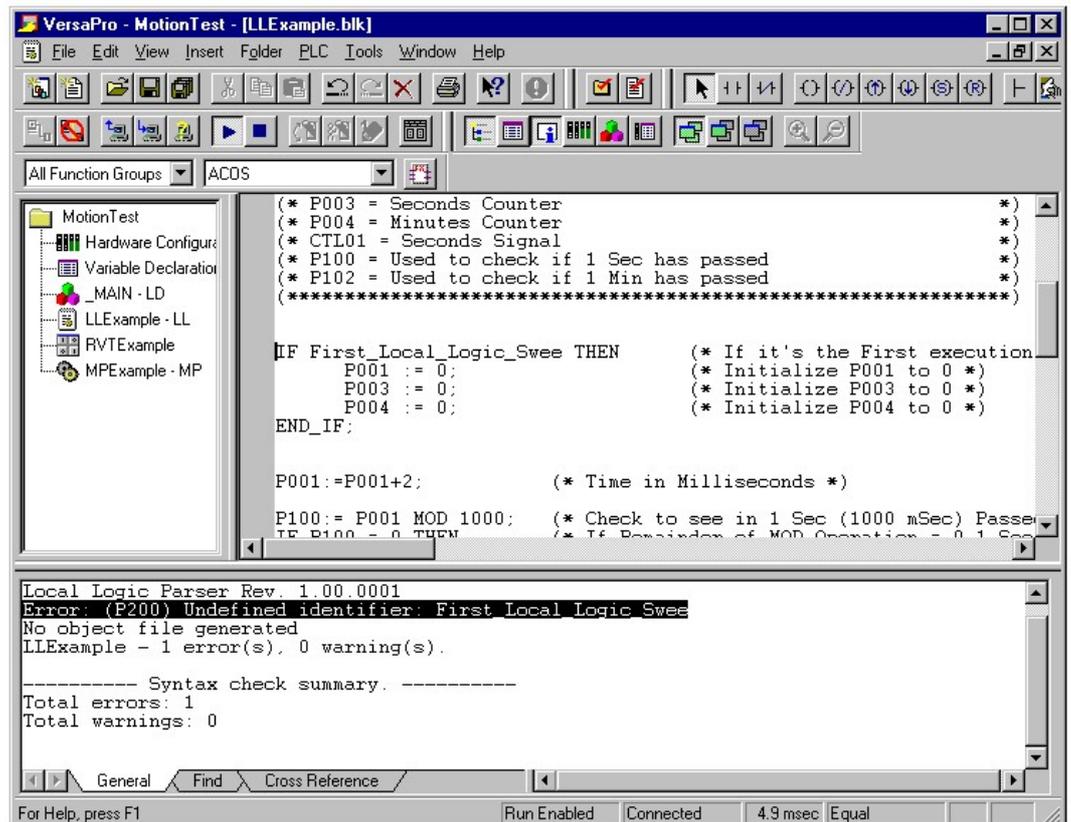


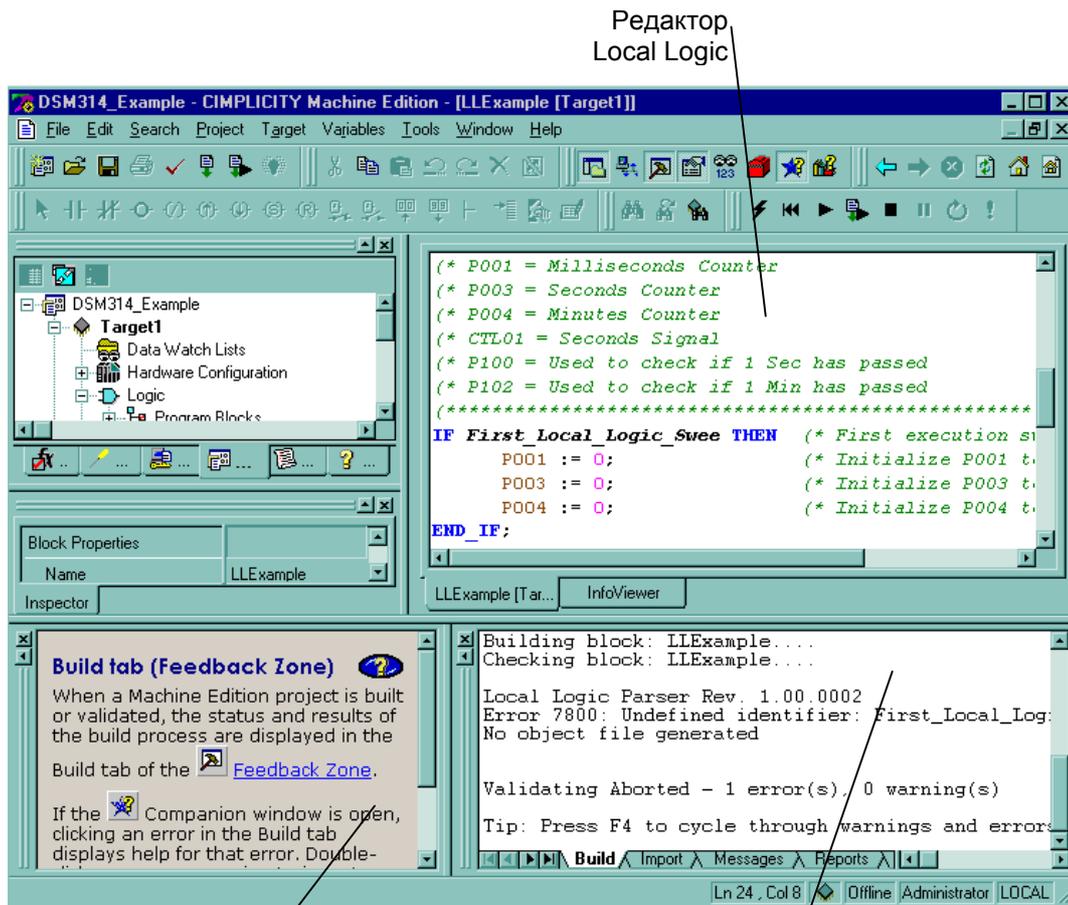
Рис. 10-14. Пример проверки синтаксиса LLExample при наличии ошибки

В главе 12 имеется дополнительная информация, касающаяся неуспешной проверки синтаксиса и корректирующих действий.

## CIMPLICITY Machine Edition

Чтобы проверить синтаксис языка, выберите Target (Задача), затем Validate (Проверить) <Имя задачи>. Можно также нажать F7, находясь в любом месте окна CIMPLICITY Machine Edition. Будет проведена проверка всех логических блоков активной задачи. Результаты проверки синтаксиса отображаются в разделе Feedback Zone (Зона обратной связи). (Если раздел Feedback Zone (Зона обратной связи) еще не открыт, то запуск процесса Validate (Проверить) откроет его).

В приведенном выше примере строка "First\_Local\_Logic\_Sweep" введена с ошибкой, как "First\_Local\_Logic\_Swee".



Справка по  
ошибке

Зона обратной  
связи

**Советы:** Чтобы перемещаться в Зоне обратной связи по предупредительным сообщениям и сообщениям об ошибках, нажимайте F4.

Чтобы перейти к той строке в локальной логической программе, которая создала ошибку, следует дважды щелкнуть описание ошибки в Зоне обратной связи. Курсор переместится в окно редактора Local Logic на начало строки, содержащей ошибку.

В главе 12 имеется дополнительная информация, касающаяся неуспешной проверки синтаксиса и корректирующих действий.

## Установка параметров конфигурации аппаратного обеспечения для Local Logic

Если проверка синтаксиса прошла успешно, то требуется установить конфигурацию аппаратного обеспечения, что должно позволить загрузить программу примера в надлежащий модуль DSM314. Следует отметить, что это - нетипичный порядок действий, в котором обычно выполняются данные шаги. Большинство пользователей сначала задает конфигурацию своего аппаратного обеспечения, а затем генерирует операторы программирования. Однако данный порядок в настоящем примере используется для того, чтобы лучше показать связь между конфигурацией аппаратного обеспечения и именем программы Local Logic в конфигурации аппаратного обеспечения модуля DSM314.

Более детальная информация о проведении шагов 1 и 2 находится в следующих разделах главы 2:

- "Конфигурация VersaPro"
  - "Конфигурация SIMPLICITY Machine Edition"
1. Если это еще не сделано, то следует открыть конфигурацию аппаратного обеспечения и сконфигурировать ЦП, поддерживающий микропрограмму версии 10.0 или более позднюю, и соответствующий источник питания для вашего применения. Добавьте модуль DSM314 в конфигурацию крейта. Эта операция добавляет модуль DSM314 в крейт и открывает окна конфигурации DSM314, которые позволяют настроить модуль DSM314 на конкретное применение.

**Примечание:** См. гл. 4 о деталях установочных параметров конфигурации модуля DSM314.

2. В таблице "Установочные параметры" установите параметр "Local Logic Mode" ("Режим Local Logic") на Enabled (Включено) и введите название программы примера "LLExample" в поле "Local Logic Block Name" ("Имя блока Local Logic"). Получающееся окно Конфигурации Аппаратного обеспечения будет выглядеть так, как показано на рис. 10-15.

**Примеч.:** Этот метод связывания модуля DSM314 с программой Local Logic позволяет легко задавать различные модули DSM314, которые используют одну и ту же программу Local Logic. Данный пример имеет только один модуль DSM314. Однако, если имеется несколько модулей DSM314, которые должны работать с одной и той же программой Local Logic, то следует просто указать это при конфигурации каждого модуля DSM314, который должен выполнять данную программу. Это позволяет программатору иметь один исходный файл Local Logic для нескольких модулей DSM314. Следует также отметить, что это не исключает выполнение модулями DSM314 различных программ.

Parameters	Values
Number of Axes:	4
%I Reference:	%I00081
%I Length:	80
%Q Reference:	%Q00001
%Q Length:	80
%AI Reference:	%AI0001
%AI Length:	84
%AQ Reference:	%AQ0001
%AQ Length:	12
Axis 1 Mode:	Analog Servo
Axis 2 Mode:	Analog Servo
Axis 3 Mode:	Auxiliary Axis
Axis 4 Mode:	Disabled
Local Logic Mode:	Enabled
Total Encoder Power (Watts)	0
Motion Program Block Name:	
Local Logic Block Name:	LLExample
Cam Block Name:	

Рис. 10-15. Таблица установочных параметров модуля DSM314 при аппаратной конфигурации крейта 90-30

3. Конфигурация возвращаемых данных

Пример программы Local Logic, показанный на стр. 10-23, использует регистры параметров P001, P003 и P004 в качестве счетчиков, которые содержат значения, представляющие время. Чтобы просмотреть эти регистры параметров в регистрах возвращаемых данных модуля DSM314, требуется сконфигурировать возвращаемые данные. Для конфигурации возвращаемых данных:

- A. Выберите страницу-закладку Axis (Ось) #1 и вход 18 в Return Data 1 Mode (Режим Возвращаемых Данных 1). Таким образом, модуль DSM информируется о необходимости возврата регистров параметров. В Return Data 1 Offset (Смещение Возвращаемых Данных 1) введите 1. Этим модуль DSM информируется о необходимости возврата параметра P001.

Программа LLExample возвращает P001, P003 и P004. Однако группировать лучше, если P003 и P004 возвращаются в Axis (Ось) #2. Поэтому можно либо оставить Return Data 2 Mode (Режим Возвращаемых Данных 2) и Return Data 2 Offset (Смещение Возвращаемых Данных 2) на их значениях по умолчанию, либо ввести 18 в Return Data 2 Mode и 2 – в Return Data 2 Offset, чтобы сообщить модулю DSM о необходимости возврата P002. Заметьте, что Select Return Data 1 (Выбрать возвращаемые данные 1) Оси 1 возвращается в смещение памяти %AI 21, а Return Data 2 (Возвращаемые данные 2) для оси 1 возвращается в смещение %AI 23.

Parameters	Values
Drive Disable Delay (ms):	100
Jog Velocity:	1000
Jog Acceleration:	10000
Jog Acceleration Mode:	Linear
Home Position:	0
Home Offset:	0
Find Home Velocity:	2000
Final Home Velocity:	500
Home Mode:	Home Switch
Return Data 1 Mode:	18
Return Data 1 Offset:	1
Return Data 2 Mode:	18
Return Data 2 Offset:	2
Cam Master Source:	Actual Position 3
Followers Control Logic:	Disabled
Ratio A Value:	1

Рис. 10-16. Таблица Axis#1 (Ось#1) модуля DSM314 при аппаратной конфигурации крейта 90-30

Указанные выше шаги должны быть повторены для P003 и P004.

- B. Выберите страницу-закладку Axis (Ось) #2 и вход 18 в Return Data 1 Mode (Режим Возвращаемых Данных 1). Таким образом, модуль DSM информируется о необходимости возврата регистров параметров. В Return Data 1 Offset (Смещение Возвращаемых Данных 1) введите 3. Этим модуль DSM информируется о необходимости возврата параметра P003.
- C. В таблице Axis (Ось) #2 введите 18 в Return Data 2 Mode (Режим Возвращаемых Данных 2) и введите 4 в Return Data 2 Offset (Смещение Возвращаемых Данных 2), чтобы сообщить модулю DSM о необходимости возврата P004.

**Примечание:** Select Return Data 2 (Выбрать возвращаемые данные 2) Оси 2 возвращается в смещение памяти %AI 41, а Return Data 2 (Возвращаемые данные 2) для оси 2 возвращается в смещение %AI 43.

Parameters	Values
Drive Disable Delay (ms):	100
Jog Velocity:	1000
Jog Acceleration:	10000
Jog Acceleration Mode:	Linear
Home Position:	0
Home Offset:	0
Find Home Velocity:	2000
Final Home Velocity:	500
Home Mode:	Home Switch
Return Data 1 Mode:	18
Return Data 1 Offset:	3
Return Data 2 Mode:	18
Return Data 2 Offset:	4
Cam Master Source:	Actual Position 3
Follower Control Logic:	Disabled
Ratio A Value:	1

Рис. 10-17. Таблица Axis#2 (Ось#2) модуля DSM314 при аппаратной конфигурации крейта 90-30

#### 4. Конфигурация CTL-бита.

Пример программы Local Logic, показанный на стр. 10-23, управляет битом CTL01, который используется для информирования программы движения о том, что секунда прошла. Этот CTL-бит должен быть сконфигурирован так, чтобы быть управлением Local Logic. Чтобы сделать это, надо войти в закладку "CTL Bits" ("CTL-биты") при конфигурации аппаратного обеспечения. Выберите "CTL01 Config" ("Конфигурация CTL01") и задайте Local\_Logic\_Controlled. Результирующее окно CTL01 выглядит так, как показано на рис. 10-18.

Parameters	Values
CTL01 Config:	Local Logic Controlled
CTL02 Config:	IN10_A (Axis 1 - OT)
CTL03 Config:	IN11_A (Axis 1 Home SW)
CTL04 Config:	Strobe 1 Level (Axis 1)
CTL05 Config:	IN9_B (Axis 2 + OT)
CTL06 Config:	IN10_B (Axis 2 - OT)
CTL07 Config:	IN11_B (Axis 2 Home SW)
CTL08 Config:	Strobe 1 Level (Axis 2)
CTL09 Config:	%Q Bit Offset 12
CTL10 Config:	%Q Bit Offset 13
CTL11 Config:	%Q Bit Offset 14
CTL12 Config:	%Q Bit Offset 15
CTL13 Config:	IN9_C (Axis 3 + OT)
CTL14 Config:	IN10_C (Axis 3 - OT)
CTL15 Config:	IN11_C (Axis 3 Home SW)
CTL16 Config:	Strobe 1 Level (Axis 3)
CTL17 Config:	%Q Bit Offset 24

Рис. 10-18. Таблица CTL Bits (CTL-биты) модуля DSM314 при аппаратной конфигурации крейта 90-30

- Это завершает изменения конфигурации, требуемые в данном примере. Закройте средства конфигурации аппаратного обеспечения и сохраните папку. Связь между примером программы Local Logic и модулем DSM314 теперь закончена. Можно теперь создать любую требуемую ранжированную многоступенчатую логику ПЛК и затем выполнить Check All (Проверить все) на этих программах.

## Загрузка вашей программы Local Logic

Чтобы выполнить операцию загрузки, сначала следует убедиться, что порт связи правильно сконфигурирован. Для входа в установку связи выберите команду Tools (Инструменты) в главном меню, а затем выберите подменю Communications Setup (Установка связи). Данный пункт меню вызывает пакет конфигурации связи. Первое диалоговое окно запрашивает у пользователя пароль. Паролем по умолчанию (при поставке) является **netutil** (рис. 10-19).

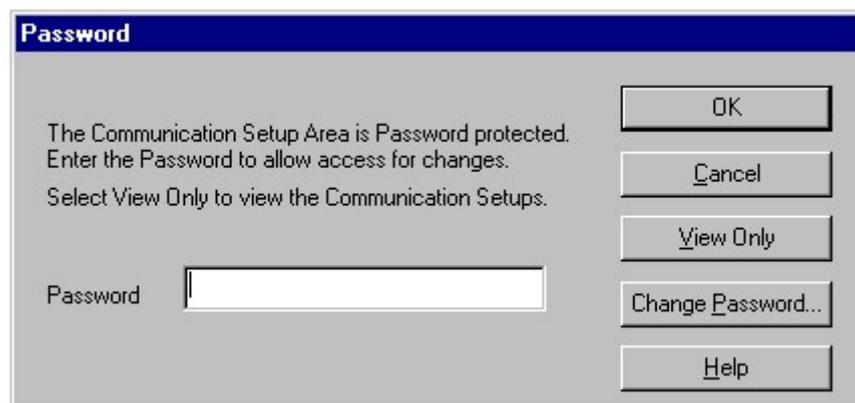


Рис. 10-19. Диалоговое окно пароля при установке связи

Ввод пароля вызывает утилиту конфигурации связи (рис. 10-20). VersaPro использует Host Communications Toolkit (HCT) (Набор средств связи с главным компьютером) для обеспечения драйверов связи. Дополнительная информация по правильному использованию данной утилиты содержится в руководстве "GFK-1026 - GE Fanuc Утилита конфигурации связи". По этой утилите может быть получена интерактивная помощь, которая в состоянии дать ответы на большинство наиболее часто встречающихся вопросов.

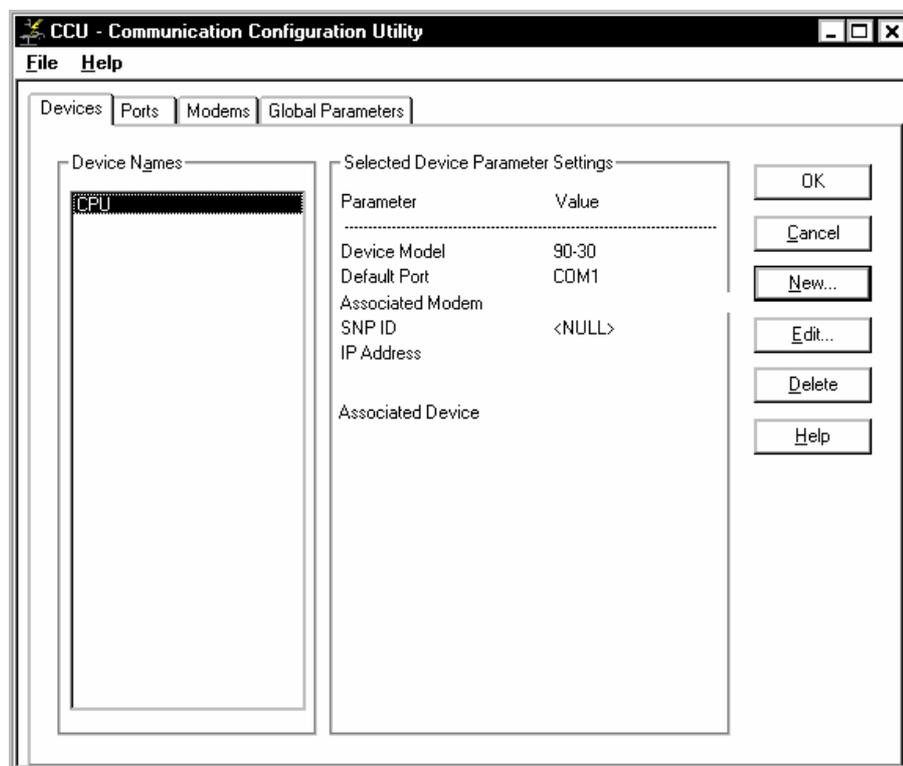


Рис. 10-20. Утилита конфигурации связи

После конфигурации порта связи в ЦП ПЛК может быть загружена локальная логическая программа. Чтобы сохранить текущую папку в ПЛК, выберите **P**LC (ПЛК) в меню, а затем пункт **C**onnect (Соединить) в подменю. После установления соединения следует выбрать **P**LC (ПЛК) в меню, а затем пункт **S**to**r**e (Сохранить) в подменю. Операция сохранения начинает процесс передачи папки из программатора в ЦП ПЛК. Когда пользователь инициирует операцию сохранения, то появляется диалоговое окно, которое позволяет пользователю выбрать то, что требуется сохранить в ПЛК. В данном случае выбрано сохранение программы Local Logic, конфигурации аппаратного обеспечения и всей логики ПЛК. Чтобы выполнить эту операцию, выберите в диалоговом окне *Store hardware configuration and motion to the PLC (Сохранить в ПЛК конфигурацию аппаратного обеспечения и движение)* и *Store logic to PLC (Сохранить в ПЛК логику)* (рис. 10-21).

**Примечание:** Программы Local Logic и программы движения передаются как часть процесса конфигурации аппаратного обеспечения. Поэтому при каждой загрузке измененной программы Local Logic и/или программы движения следует выбирать в диалоговом окне сохранения пункт *Store hardware configuration and motion to the PLC (Сохранить в ПЛК конфигурацию аппаратного обеспечения и движение)*. (Рис. 10-21).

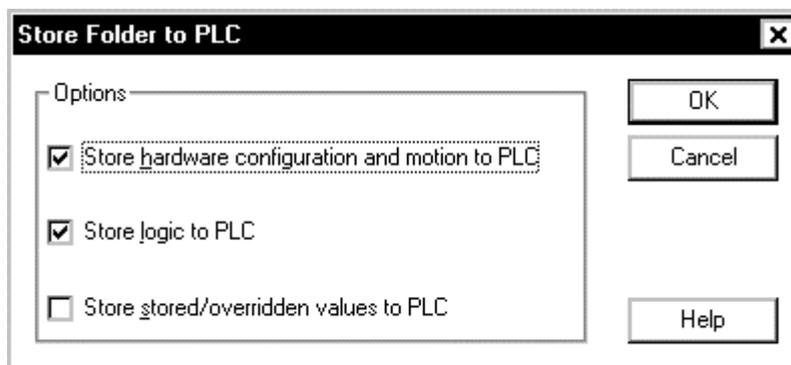


Рис. 10-21. Диалоговое окно сохранения в VersaPro

VersaPro затем проверяет все блоки, которые были изменены. Если процедура построения проходит успешно, то файлы загружаются в ПЛК. VersaPro показывает успешное проведение загрузки индикацией в диалоговом окне "Store to PLC Successful" ("Сохранение в ПЛК прошло успешно").

Теперь программы загружены в ПЛК. Можно направить запросы в DSM, чтобы убедиться в правильности работы программы Local Logic. Для этой операции может быть использовано окно Reference View Table (RVT) (Таблица просмотра ссылок). Чтобы создать RVT, войдите в пункт меню **F**ile (Файл) и выберите пункт **N**ew **R**eference View Table (Новая таблица просмотра ссылок). (Рис. 10-21). Это приведет к появлению диалогового окна New Reference View Table (Новая таблица просмотра ссылок). Диалоговое окно позволяет пользователю дать имя данной таблице RVT. В настоящем примере задано имя RVTExample.

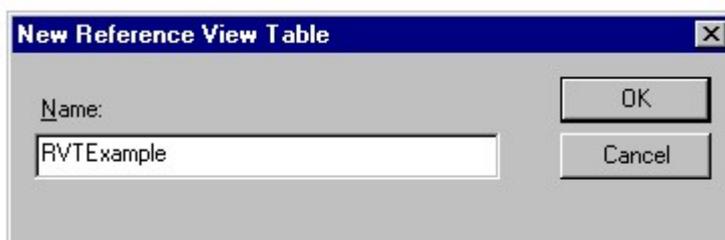


Рис. 10-22. Диалоговое окно "Новая таблица просмотра ссылок"

Теперь пользователь может вставлять переменные, определять форматы отображения переменных, точки переключения данных и посылать AQ-команды наряду с другими действиями. Более детальную информацию по строению RVT см. в документации по VersaPro. Один пример RVT, полезный для данной программы, показан на рис. 10-23.

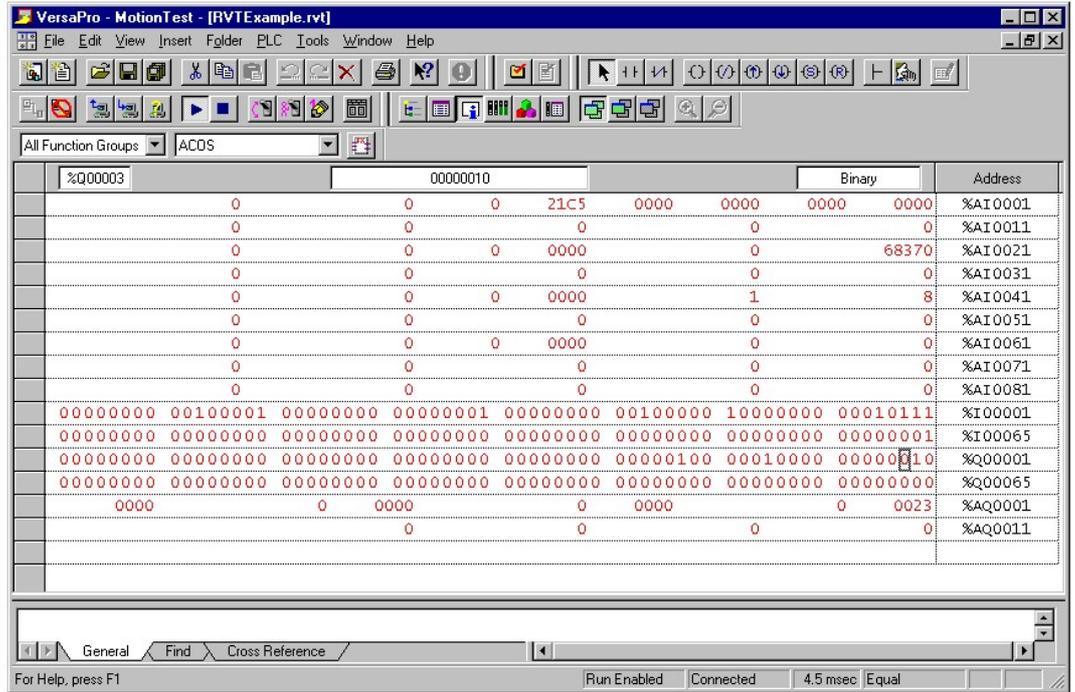


Рис. 10-23. Таблица просмотра ссылок

## Выполнение вашей первой программы Local Logic

После завершения операции загрузки модуль готов выполнить локальную логическую программу. Чтобы модуль DSM начал выполнять локальную логическую программу, должно быть установлено смещение 1 Q-бита от ПЛК в то время, когда ПЛК находится в режиме RUN (ВЫПОЛНИТЬ). В этот момент происходит активация локальной логической программы, и она начинает работать в модуле DSM.

**Примечание:** Пример *LLExample* программы является простым применением счетчика. Пользователь может использовать RVT для просмотра соответствующих параметров, чтобы удостовериться в активности программы и в ее правильном функционировании. Из RVT (рис. 10-23) можно видеть, что после запуска Local Logic прошла 1 минута и 8 секунд (см., соответственно, %AI0043 и %AI0041). Кроме того, прошло 68370 миллисекунд, как видно в %AI0021. Дополнительная информация, касающаяся интерфейса ПЛК между DSM и ПЛК, имеется в главе 5. После проверки программы на правильность работы следует сохранить папку.

## Использование редактора программы движения

Теперь после проверки нормальной работы программы Local Logic следует связать ее с программой движения. Вход в редактор программ движения выполняется способом, очень похожим на вход в редактор Local Logic. Этот редактор позволяет легко создавать, редактировать, сохранять и загружать программы движения. Чтобы создать программу движения, папка VersaPro должна быть открыта. В настоящем примере открытая папка VersaPro должна быть папкой, созданной как часть примера Local Logic в предыдущих разделах. Именем папки в данном примере является "MotionTest" ("Тест движения"). Процедуру открытия папки см. в документации по VersaPro. Если папка VersaPro открыта, выберите в меню команду File (Файл), затем New Program (Новая программа), после чего Motion... (Движение...) (рис. 10-23).



Рис. 10-24. Меню создания программы движения VersaPro

Выбор приведенных ниже меню вызывает диалоговое окно, которое позволяет пользователю задать имя программы движения, дать описательный комментарий и выбрать тип модуля движения. В настоящее время программы движения, использующие данный редактор, поддерживаются только на модуле DSM314. Поэтому не следует изменять выбор Motion Module Type (Тип модуля движения), предлагаемый по умолчанию. (См. рис. 10-24).

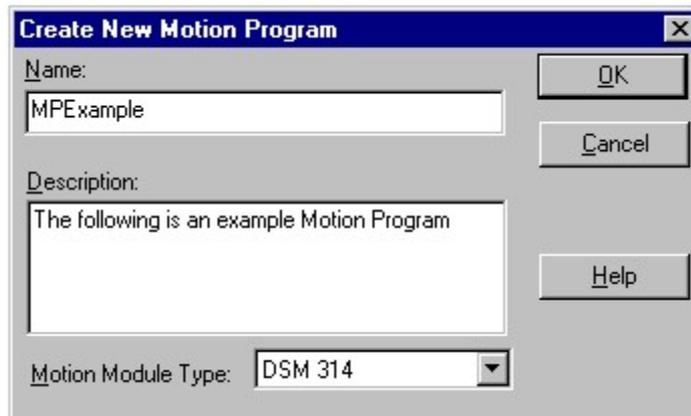


Рис. 10-25. Диалоговое окно Create New Motion Program (Создать новую программу движения)

Для создание программы движения нажмите кнопку ОК.

После завершения данного действия VersaPro выводит окно редактора движения. (Рис. 10-26).

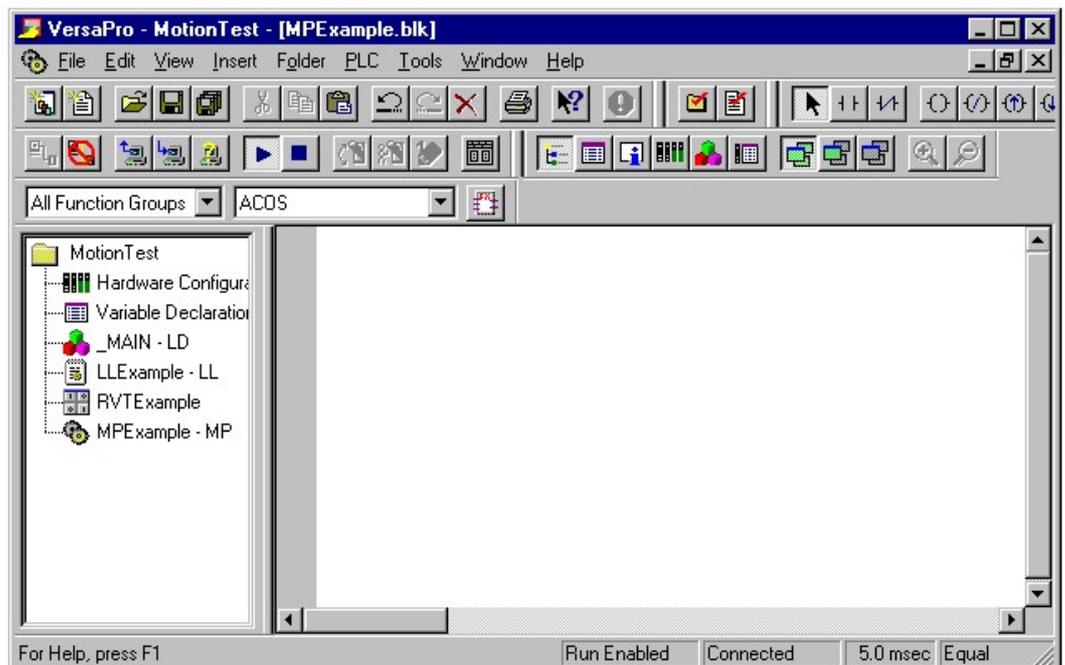


Рис. 10-26. Незаполненный редактор программ движения в VersaPro

В предыдущих разделах работало меню файла; это не повторится в данном разделе. Ссылки на эту информацию см. в предыдущих разделах. Редактор программ движения является редактором с текстом в свободной форме; он позволяет вводить программы в том стиле, который предпочитает пользователь. Пример использует очень простую программу движения. Этот пример не представляет собой функциональное приложение и предназначен только для целей обучения. Этот пример связан с программой Local Logic, введенной в разделе о начале работы с Local Logic. Для напоминания ниже приведена программа Local Logic из предыдущих разделов:

```

(*****)
(* Название программы: LLExample *)
(* Описание: Данный пример локальной логической программы является *)
(* программой, которая создает обычный таймер. Таймер начинает *)
(* работу после того, как произойдет первый Local_Logic_Sweep *)
(* (Локальный Логический Цикл). Программа использует тот факт, что *)
(* Local Logic запускается модулем DSM каждые 2 мс. Таким образом, *)
(* программа считает локальные логические циклы. Программа имеет *)
(* три счетчика. Первый счетчик считает миллисекунды, а два других *)
(* - секунды и минуты. Счетчики секунд и минут переходят на ноль *)
(* при значении 59. Программа также устанавливает каждую секунду *)
(* бит STL01. *)
(*****)
(* Переменные *)
(* P001 = Счетчик миллисекунд *)
(* P003 = Счетчик секунд *)
(* P004 = Счетчик минут *)
(* STL01 = Сигнал секунд *)
(* P100 = Используется для проверки того, прошла ли 1 секунда *)
(* P102 = Используется для проверки того, прошла ли 1 минута *)
(*****)
IF First_Local_Logic_Sweep THEN (* Первый цикл выполнения *)
    P001 := 0; (* Установка P001 на 0 *)
    P003 := 0; (* Установка P003 на 0 *)
    P004 := 0; (* Установка P004 на 0 *)
END_IF;

P001:=P001+2; (* Время в миллисекундах *)

P100:= P001 MOD 1000; (* Проверка, прошла ли 1 секунда (1000 мс) *)
IF P100 = 0 THEN (* Если остаток от операции MOD =0, 1 секунда прошла *)
    P003:=P003+1; (* Время в секундах *)
    STL01 := 1;
    IF P003 = 60 THEN (* Если секунды = 60, то начинается с 0 *)
        P003:=0;
    END_IF;
END_IF;
IF P100 <> 0 THEN
    STL01 :=0; (* STL01=0 Если счетчик секунд не инкрементный *)
END_IF;

P101:=P001 MOD 60000; (* Проверка, прошла ли 1 минута (60000 мс) *)
IF P101 = 0 THEN (* Если остаток от операции MOD =0, 1 минута прошла *)
    P004:=P004+1; (* Время в минутах*)
    IF P004 = 60 THEN (* Если минуты = 60, то начинается с 0 *)
        P004:=0;
    END_IF;
END_IF;

```

Программа Local Logic вызывает переход CTL01 каждую секунду из состояния логического 0 в состояние логической 1. В данном простом примере программы движения вал двигателя поворачивается на 1/60 оборота при каждом переходе CTL01. Таким образом, программа движения будет вызывать движение вала двигателя, аналогичное движению секундной стрелки на кварцевых часах. Это не самое элегантное применение, но оно демонстрирует некоторые концепции.

До того как писать программу движения, следует определить масштаб оси. Первой переменной, которую требуется определить, является отношение числа пользовательских единиц к числу импульсов. Параметр "Отношение числа пользовательских единиц к числу импульсов" задает число программных единиц измерения, приходящихся на каждый импульс обратной связи по положению. Это дает возможность пользователю программировать модуль DSM314 в единицах измерения, специфических для конкретного применения. Количество пользовательских единиц и число импульсов должны быть в диапазоне от 1 до 65 535. Отношение числа пользовательских единиц к числу импульсов должно быть в диапазоне от 8:1 до 1:32. Если, например, 1 000 дюймов перемещения соответствует 8192 импульсам обратной связи, то отношение числа пользовательских единиц к числу импульсов, задаваемое как 1000:8192, устанавливает 1 пользовательскую единицу равной 0.001 дюйма.

Чтобы задать отношение числа пользовательских единиц к числу импульсов, требуется, прежде всего, информация о количестве импульсов, приходящемся на один оборот устройства обратной связи. В данном примере используется двигатель Beta 0.5. Beta 0.5 имеет разрешение обратной связи 8192 импульсов на оборот. Теперь выполним небольшой расчет, чтобы определить отношение. Базовым уравнением является следующее:

$$\frac{\text{Польз.ед.}}{\text{Импульсы}} = \left( \frac{\text{Передвижение нагрузки на оборот двигателя}}{\text{Требуемое разрешение}} \right) \cdot \frac{1}{\text{Кол. - во импульсов датчика на оборот двиг.}}$$

Для данного примера:

$$\frac{\text{Польз.ед. - цы}}{\text{Импульсы}} = \left( \frac{1}{\frac{1}{60}} \right) \cdot \frac{1}{8192}$$

$$\frac{\text{Польз.ед. - цы}}{\text{Импульсы}} = \frac{60}{8192}$$

Такое отношение является проблематичным, поскольку оно нарушает правило, что минимальное значение отношения числа пользовательских единиц к числу импульсов равно 1/32. Эта проблема легко решается: следует изменить программные единицы с 60 на оборот до 600 на один оборот. Это дает 1 программную единицу, равную 1/600 оборота. Повторим вышеприведенный расчет:

$$\frac{\text{Польз.ед. - цы}}{\text{Импульсы}} = \left( \frac{1}{\frac{1}{600}} \right) \cdot \frac{1}{8192}$$

$$\frac{\text{Польз.ед. - цы}}{\text{Импульсы}} = \frac{600}{8192}$$

Таким образом, чтобы получить перемещение двигателя на 1/60 оборота, следует ввести 10 единиц в программу движения. Дополнительная информация по заданию отношения числа пользовательских единиц к числу импульсов содержится в главе 4.

Следующей позицией, которую следует определить, является максимальная скорость двигателя. Это вычисление является сравнительно простым.

$$\begin{aligned} \text{Макс. скор.} \cdot \left( \frac{\text{польз.ед.}}{\text{секунды}} \right) &= \text{Макс.ск.двиг.} \cdot \left( \frac{\text{об.}}{\text{с}} \right) \cdot \text{Число\_имп.\_датч.\_на\_об.} \cdot \left( \frac{\text{имп.}}{\text{об.}} \right) \cdot \frac{\text{Польз.ед.}}{\text{Имп.}} \cdot \left( \frac{\text{польз.ед.}}{\text{имп.}} \right) \\ \text{Макс. скор.} \cdot \left( \frac{\text{польз.ед.}}{\text{секунды}} \right) &= \frac{3000}{60} \cdot \left( \frac{\text{об.}}{\text{с}} \right) \cdot 8192 \cdot \left( \frac{\text{имп.}}{\text{об.}} \right) \cdot \frac{600}{8192} \cdot \left( \frac{\text{польз.ед.}}{\text{имп.}} \right) \\ \text{Макс. скор.} \cdot \left( \frac{\text{польз.ед.}}{\text{секунды}} \right) &= 3000 \cdot 10 \cdot \left( \frac{\text{польз.ед.}}{\text{секунды}} \right) \\ \text{Макс. скор.} \cdot \left( \frac{\text{польз.ед.}}{\text{секунды}} \right) &= 30000 \cdot \left( \frac{\text{польз.ед.}}{\text{секунды}} \right) \end{aligned}$$

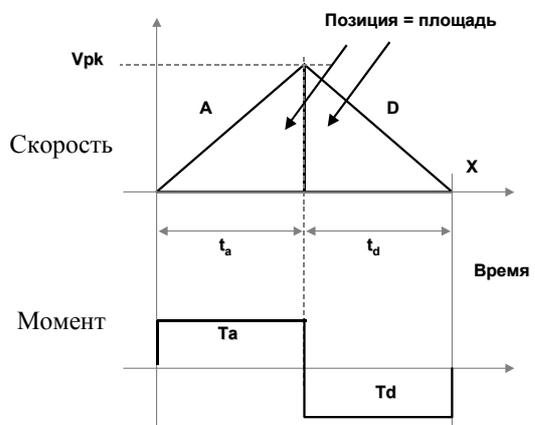
Затем следует рассчитать скорость и ускорение, требуемые для движения. В данном примере выбирается треугольный график скорости, чтобы минимизировать время. Уравнения, предназначенные для расчета параметров, показаны ниже.

**Уравнения:**

$$x = \frac{1}{2} V_{pk} (t_a + t_d)$$

$$V_{pk} = \frac{2(x)}{(t_a + t_d)}$$

$$a = \frac{V_{pk}}{t_a}$$



Использование значений из данного примера к уравнению треугольного графика скорости дает следующее:

Дано :

$$t_a = 0.01 \cdot c$$

$$t_d = 0.01 \cdot c$$

$$x = \frac{1}{60} \cdot \text{оборота}$$

$$V_{pk} = \frac{2 \cdot \frac{1}{60} \cdot \text{об.}}{0.01 \cdot c + 0.01 \cdot c}$$

$$V_{pk} = 1.6667 \cdot \frac{\text{оборота}}{\text{секунда}}$$

$$V_{pk} = 1.6667 \cdot \frac{\text{об.}}{c} \cdot 8192 \cdot \frac{\text{имп.}}{\text{об.}} \cdot \frac{600}{8192} \cdot \frac{\text{польз.ед.}}{\text{имп.}}$$

$$V_{pk} = 1000 \cdot \frac{\text{польз.ед.}}{c}$$

$$a = \frac{V_{pk}}{t_a}$$

$$a = \frac{1000 \cdot \frac{\text{польз.ед.}}{c}}{0.01 \cdot c}$$

$$a = 10000 \cdot \frac{\text{польз.ед.}}{c^2}$$

Теперь можно писать программу движения. Код программы движения приведен ниже.

```

(*****)
(* Имя программы: MPEexample *)
(* Описание: Приведенная далее программа движения вызывает *)
(* поворот двигателя на 10 единиц (величина перемещения *)
(* определяется масштабом) каждый раз, когда происходит *)
(* переход STL01 из 0 в 1. *)
(*****)
(* Переменные *)
(* STL01 = Триггер выполнения программы *)
(*****)

PROGRAM 1 AXIS1 (* Номер программы 1 для Оси 1 *)

    ACCEL 10000 (* 10000 польз.ед./секунда^2 *)
    VELOC 1000 (* 100 польз.ед./секунда *)

1:
    WAIT STL01 (* Ждать сигнал от STL01 *)
    PMOVE 10, INCR, LINEAR (* Линейное изменение позиции на 10*)
    JUMP UNCOND, 1 (* Переход назад на начало *)

ENDPROG (* Конец программы *)

```

Пользователь должен ввести вышеприведенную программу в текстовый редактор. После завершения этого процесса редактор будет выглядеть, как показано на рис. 10-27.

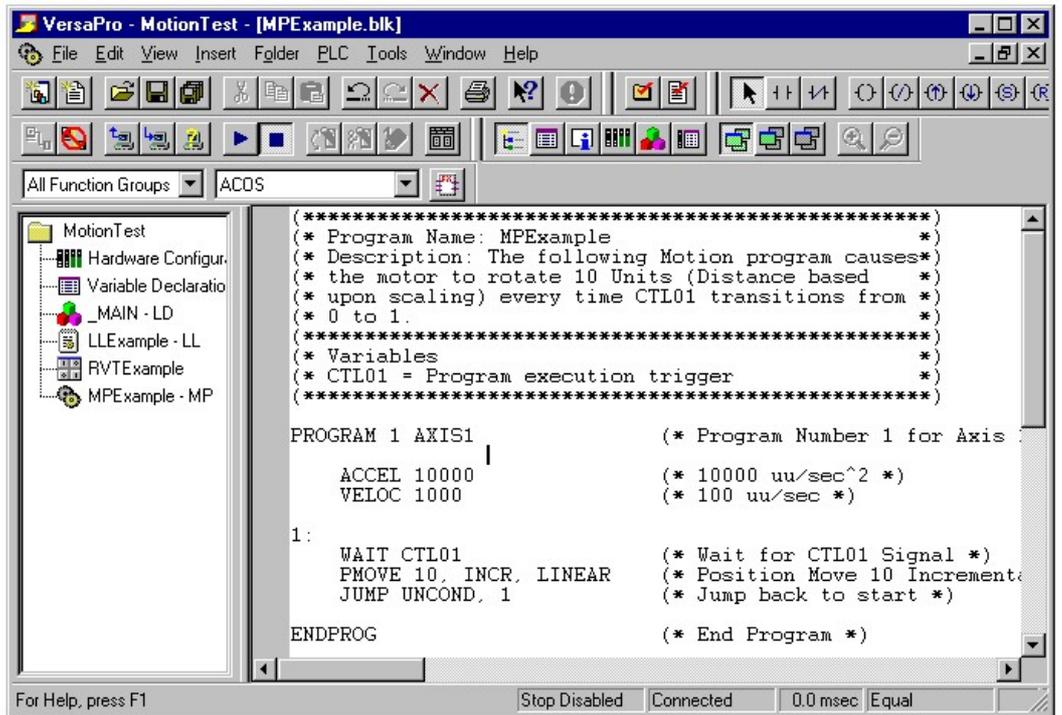


Рис. 10-27. Редактор программы VersaPro с примером программы MPEexample

Здесь пользователь должен проверить программу на предмет правильности синтаксиса языка. Проверка синтаксиса осуществляется выбором из главного меню команды Folder (Папка), а затем подменю Check Block 'MPEexample' (Проверить блок "MPEexample"). Это ведет к началу работы подпрограммы проверки синтаксиса на указанной программе движения. Примечание: Можно также выбрать Check All (Проверить все), чтобы задать все блоки в папке. (См. рис. 10-27).

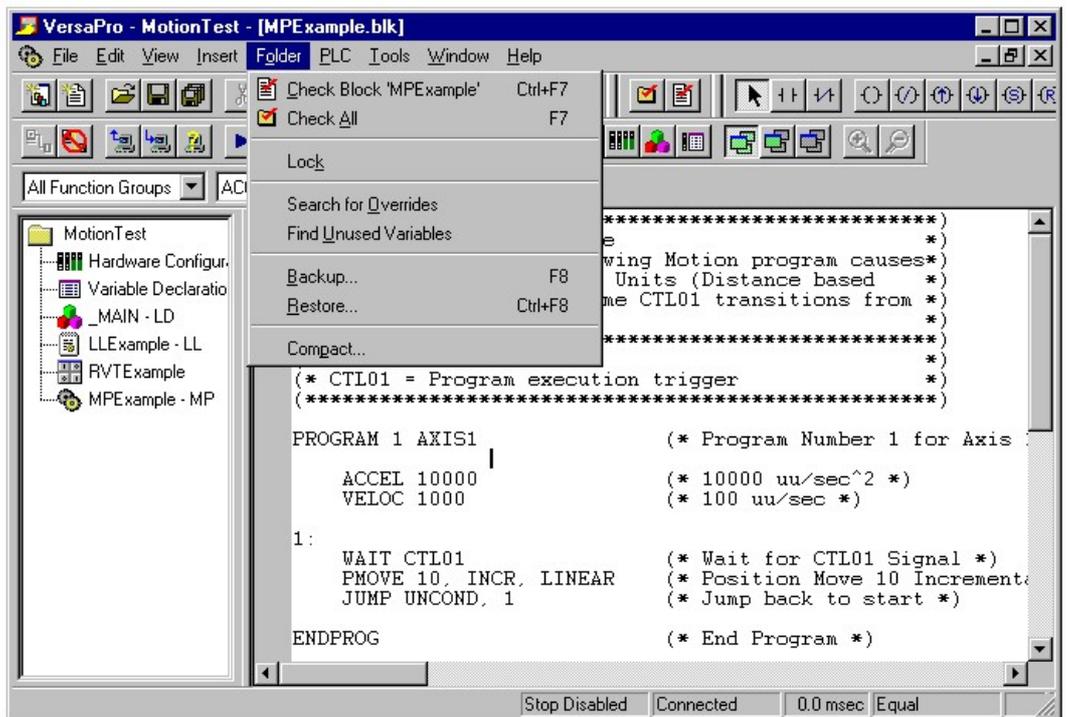


Рис. 10-28. Меню проверки блока MPEexample

Проверка блоков вызывает появление информационного окна, если его не было до этого. Обратите внимание, что информационное окно может быть включено и выключено

также нажатием информационной кнопки .

Информационное окно отображает результаты операции проверки синтаксиса. Если программа примера была введена правильно, то должно появиться сообщение, информирующее об отсутствии ошибок и предупреждений. Примечание: Информационное окно является прокручиваемым. Пользователь может пользоваться лифтом с правой стороны окна, чтобы видеть информацию, которая не отображается в данный момент на имеющемся пространстве окна.

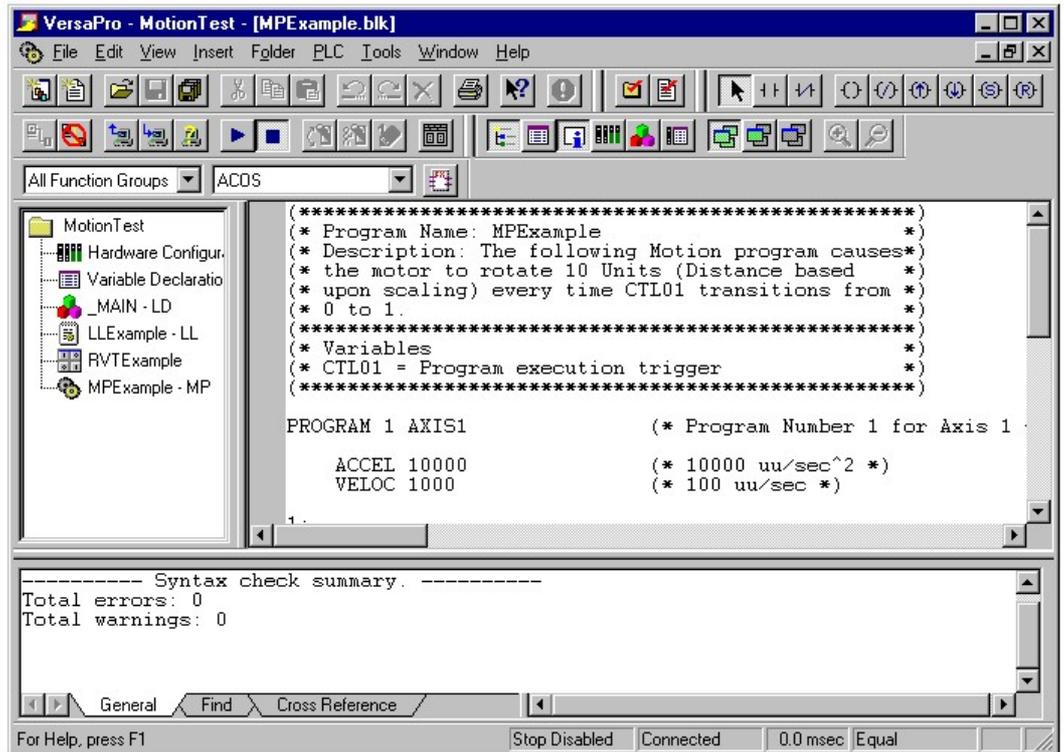


Рис. 10-29. Успешное завершение проверки синтаксиса в примере MPEExample

Если информационное окно показывает наличие синтаксической ошибки, то можно переместить это окно на строку, которая содержит сообщение об ошибке. После того как информационное окно установлено на нужную строку, щелкните дважды по сообщению об ошибке. В результате окно редактора автоматически переместится на строку программы, где обнаружена данная ошибка.

Если выполняется активное редактирование программы движение и требуется определить примерный номер строки, то следует щелкнуть вертикальный бегунок с правой стороны редактора. Появится номер строки, расположенной в самой верхней части окна. Если курсор находится несколькими строками ниже верха окна, то либо прокручивайте окно, пока текущая строка не окажется на самом верху окна, либо отсчитайте количество строк от верха и сложите их с текущим номером строки.

В главе 12 имеется дополнительная информация, касающаяся неуспешной проверки синтаксиса и корректирующих действий. Если проверка синтаксиса прошла успешно, то требуется установить конфигурацию аппаратного обеспечения, что должно позволить загрузить программу примера в надлежащий модуль DSM314. Порядок, использованный в примере, является нетипичным для большинства применений. Большинство пользователей сначала задает конфигурацию своего аппаратного обеспечения, а затем генерирует операторы программирования. Однако данный пример имеет характер иллюстрации программы движения; в нем изменена последовательность, чтобы лучше

показать связь между конфигурацией аппаратного обеспечения и именем программы движения в конфигурации аппаратного обеспечения модуля DSM314. Поэтому следующим шагом в данном примере является проведение конфигурации аппаратного обеспечения. Имеется несколько путей вывода на экран окон конфигурации аппаратного обеспечения. Подробности см. в документации по VersaPro. В предыдущих разделах данной главы показан последовательно метод активации этой конфигурации. Поэтому эта информация не повторяется в данном разделе. В настоящем разделе внимание концентрируется на дополнительной информации, которая должна быть введена в конфигурацию аппаратного обеспечения, чтобы дать возможность работать программе движения. См. гл. 4 о деталях установочных параметров конфигурации модуля DSM314. Первым полем, подлежащем редактированию, является поле "Motion Program Block Name" ("Имя блока программы движения") на странице-закладке Settings (Установочные параметры). Это поле определяет для модуля DSM314 имя программы движения, которая должна быть загружена в модуль. Введите в поле имя примера программы "MPEexample".

**Примечание:** Этот метод связывания модуля DSM314 с программой движения был выбран, чтобы позволить пользователю легко задавать различные модули DSM314, которые должны использовать одну и ту же программу движения. Данный пример имеет только один модуль DSM314. Однако, если имеется несколько модулей DSM314, которые должны работать с одной и той же программой движения, то конфигурация конкретного модуля DSM314 обеспечивает это. Программатор может иметь один исходный файл программы движения для нескольких модулей DSM314. Следует отметить, что это не исключает выполнение модулями DSM314 различных программ.

Поскольку данный пример использует Beta 0.5, то Axis1 Mode (Режим Оси 1) следует установить на Digital Servo (Цифровой сервопривод).

Получающееся окно Конфигурации Аппаратного обеспечения будет выглядеть так, как показано на рис. 10-30.

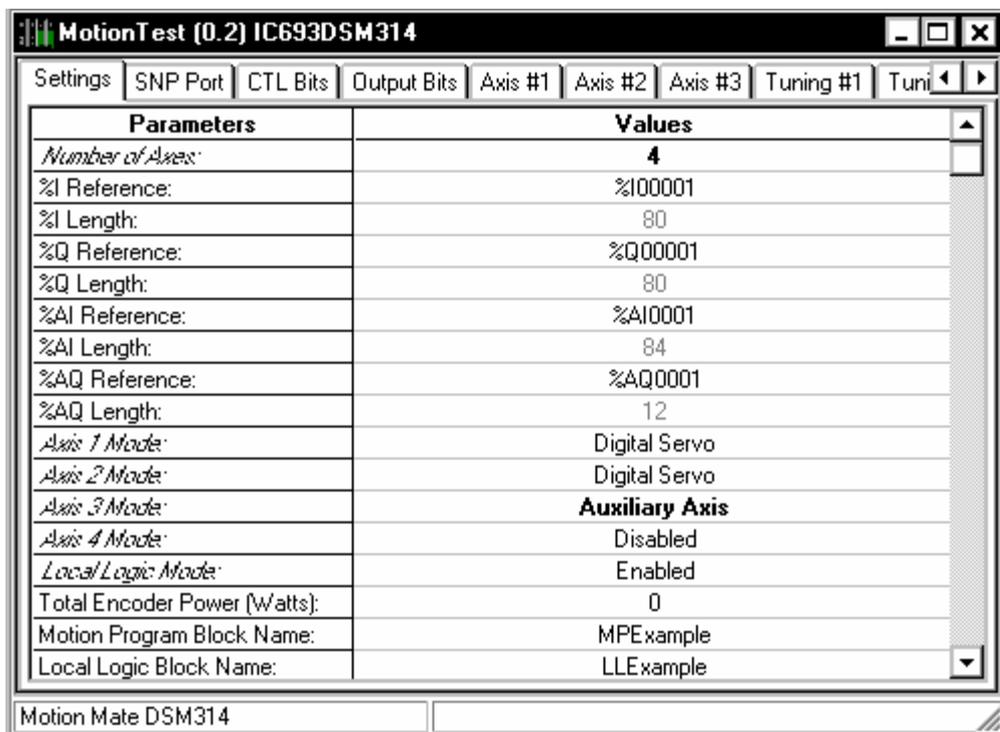


Рис. 10-30. Таблица установочных параметров модуля DSM314 при конфигурации аппаратного обеспечения

При конфигурации DSM требуется также ввести значения, рассчитанные выше, для отношения пользовательских единиц к импульсам и для максимальной скорости. В примере также конфигурируется направление оси и верхняя граница позиции. Они являются необязательными параметрами. Информацию по этим полям конфигурации см. в главе 4. Чтобы добавить эти значения, введите приведенные ниже значения в поля в таблице Axis#1.

UserUnits (Пользовательские единицы): 600

Counts (Импульсы): 8192

High Position Limit (Верхняя Граница Позиции): 599 (Необязательный параметр, вызывает переход позиции на нуль при каждом обороте)

Positive Velocity (Положительная скорость): 30000

Axis Direction (Направление Оси): Обратное (необязательный параметр, вызывает вращение сервопривода по часовой стрелке)

Parameters	Values
User Units:	600
Counts:	8192
Over Travel Limit Switch:	Disabled
Drive Ready Input:	Enabled
High Position Limit:	599
Low Position Limit:	0
High Software EOT Limit:	8388607
Low Software EOT Limit:	-8388608
Software End of Travel:	Disabled
Velocity Limit:	30000
Command Direction:	Bidirectional
Axis Direction:	Reverse
Feedback Source:	Default
Feedback Mode:	Incremental
Reversal Compensation:	0
Drive Disable Delay (ms):	100
Jog Velocity:	1000
Jog Acceleration:	10000
Jog Acceleration Mode:	Linear
Home Position:	0
Home Offset:	0
Find Home Velocity:	2000
Final Home Velocity:	500

Рис. 10-31. Таблица Axis#1 модуля DSM314 при конфигурации аппаратного обеспечения

Чтобы закончить конфигурацию, необходимо перейти к таблице Tuning#1 (Настройка#1) и ввести приведенные ниже данные.

Motor Type (Тип двигателя): 13

Position Error Limit (Предел Ошибки по Положению): 200 (необязательный параметр; см. дополнительную информацию в разделе по конфигурации)

In Position Zone (Область позиционирования): 5 (необязательный параметр; см. дополнительную информацию в разделе по конфигурации)

Pos Loop Time Const (Постоянная времени контура позиционирования): 200 (Примечание: базируется на данных по применению и по механическим узлам, см. гл. 4 и Приложение D)

Velocity FeedForward (Предустановленная скорость): 9000 (Примечание: базируется на данных по применению и по механическим узлам, см. гл. 4 и Приложение D)

Результирующее окно будет похоже на то, что представлено на рис. 10-32.

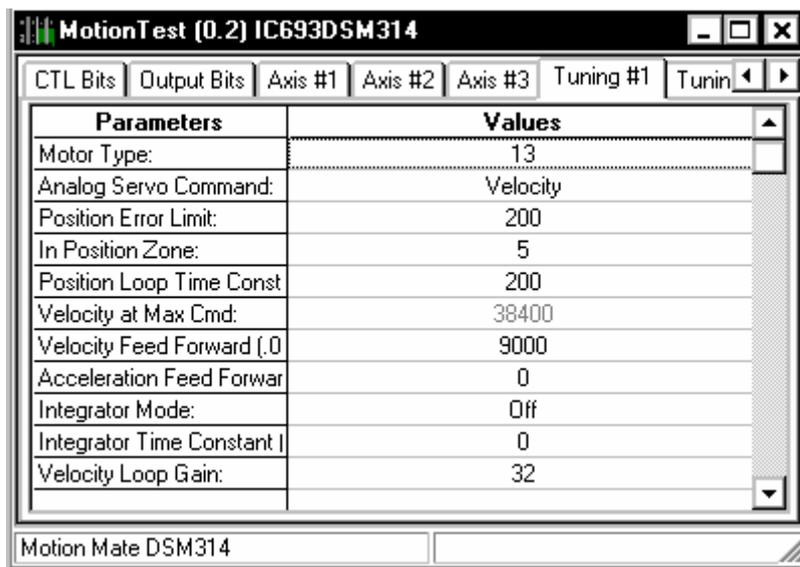


Рис. 10-32. Таблица Tuning#1 (Настройка#1) при конфигурации аппаратного обеспечения

Здесь можно закрыть диалоговое окно конфигурации модуля. Рекомендуется сохранить результаты работы. Для этого выберите команду File (Файл) из главного меню, а затем в выпавшем меню выберите пункт Save All (Сохранить все).

Связь между примером программы движения, программой Local Logic и модулем DSM314 теперь установлена. Можно теперь создать любую требуемую ранжированную многоступенчатую логику ПЛК и затем выполнить Check All (Проверить все) на этих программах, после чего загрузить их в ПЛК. Дополнительная информация, касающаяся операции загрузки, имеется в документации GFK-1670 по VersaPro или может быть получена на линии интерактивной поддержки VersaPro 1.1.

## Выполнение вашей первой программы движения

После завершения операции загрузки модуль готов выполнить программу движения и локальную логическую программу. Чтобы модуль DSM начал выполнять локальную логическую программу, пользователь должен установить смещение 1 Q-бита от ПЛК в то время, когда ПЛК находится в режиме RUN (ВЫПОЛНИТЬ). Это активизирует программу Local Logic в модуле DSM. Следующее, что требуется сделать, это выполнить команду установки позиции. Это дает модулю точку отсчета и позволяет выполнить требуемую программу движения. Чтобы выполнить это действие, следует открыть RVT (RVTEexample), созданную в разделе Local Logic и ввести 0023 шестнадцатеричное в AQ смещение 1. Таким образом, вводится команда установки позиции. Затем введите 0 в AQ смещение 2. Дополнительную информацию по вводу команд AQ см. в гл. 5. Результирующее окно будет похоже на то, что представлено на рис. 10-33.

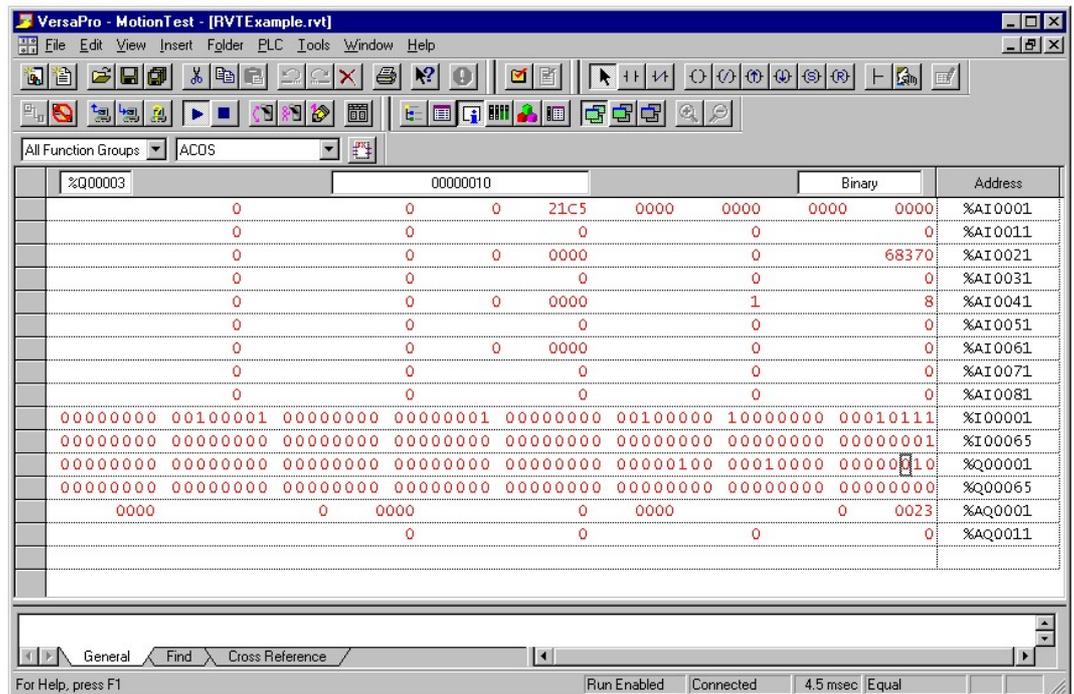


Рис. 10-33. Окно RVTEExample

В этом месте, если нет ошибок, то можно выполнить программу движения. Введите 1 (или переключите) Q бит смещение 2 (%Q00003). Двигатель должен затем начать выполнять программу движения и перемещаться каждую секунду на 1/60 полного оборота.

Дополнительная информация, касающаяся интерфейса между DSM и ПЛК, имеется в главе 5.



## Введение

Язык программирования Local Logic поддерживает операторы присваивания, условные операторы, арифметические и логические операции, а также операции отношения. Программа Local Logic работает синхронно с контуром позиционирования модуля движения, и потому она является детерминистской. Этот язык включает в себя конструкции, которые позволяют программе Local Logic обмениваться информацией между программой Local Logic, программой движения и основным ПЛК. Настоящее руководство акцентировано на локальном логическом языке и на его связи с программами движения. Дополнительная информация по языку программирования движения находится в главе 7.

## Операторы

Язык программирования Local Logic поддерживает операторы присваивания и условные операторы. Операторы присваивания позволяют назначать переменным арифметические результаты и побитовые логические операции. Условные операторы позволяют выполнять условные локальные логические программы. Условное исполнение базируется на значении постоянной или переменной или на результате выражения отношения или побитового логического выражения.

Операторы присваивания используют оператор ":=". Приведенное ниже выражение умножает два регистра параметров и назначает результат другому регистру параметров.

```
P001 := P210 * P107;
```

Примечание: Операторы присваивания должны заканчиваться точкой с запятой, как показано выше.

Условные операторы используют комбинацию ключевых слов IF-THEN-END\_IF. Ключевое слово END\_IF завершает условный оператор. В приведенном ниже примере проводится проверка значения переменной в Block\_1 и в зависимости от результата установка значения в регистре параметров. В частности, если значение переменной Block\_1 равно 5, то параметр P010 принимает значение 100.

```
IF Block_1 = 5 THEN  
    P010 := 100;  
END_IF;
```

Ключевые слова IF, THEN и END\_IF различают регистр букв; оператор END\_IF должен заканчиваться точкой с запятой. Операторы IF могут вкладываться до восьми уровней; тело операторов IF может содержать один или несколько операторов. Детальное описание этих операторов см. в главе 12.

## Комментарии

Комментарии позволяют программисту описывать работу программы или вводить любую информацию, которую программист хотел бы добавить в программу. Текст комментария начинается двумя символами (\* и заканчивается двумя символами \*); комментарий может располагаться в любом месте программы. Например:

(\* Правильная структура комментария \*)

Модуль DSM игнорирует комментарии при выполнении программы. Поэтому строки комментариев не учитываются при определении длины локальной логической программы.

## Переменные

Local Logic обеспечивает пользователю доступ к данным контроллера движения, к битам управления и статуса и к параметрам, использующим фиксированный набор переменных. Данный язык поддерживает также десятичные, шестнадцатеричные и двоичные константы. Представления шестнадцатеричных и двоичных значений являются в программных операторах константами без знака, однако в математических выражениях они всегда интерпретируются как второе дополнение со знаком. Чтобы присвоить значение переменной, пользователь должен ввести следующее:

Torque\_Limit\_1 :=5000; (\* Задание предела момента вращения для оси 1 50% \*)

или в шестнадцатеричной форме

Torque\_Limit\_1:=16#1388;(\* Задание предела момента вращения для оси 1 50% \*)

Когда каким-то переменным назначается числовое значение, то они автоматически ограничиваются, исходя из 32-битового значения со знаком. Приведенные ниже примеры показывают наибольшие, имеющиеся в распоряжении положительные и отрицательные значения.

P001:=16#7FFFFFFF;(\* P001=2147483647 \*)

или в десятичной форме

P001:=2147483647; (\* P001=16#7FFFFFFF \*)

Чтобы назначить максимальное отрицательное значение, пользователь должен ввести:

P002:=16#80000000; (\* P002=-2147483648 \*)

или в десятичной форме

P002:=-2147483648;(\* P002=16#80000000 \*)

Если пользователь вводит число, превышающее указанные выше числовые границы, то будет выдано сообщение об ошибке, показывающее, что константа находится за допустимыми пределами.

Переменные Local Logic имеют "руководящие" атрибуты чтения, записи или чтения/записи. (Дополнительная информация по переменным и их типам содержится в главе 13). Например, переменная Positive End of Travel (Положительная граница завершения движения) для Оси 1 (Positive\_EOT\_1) является переменной, доступной только для чтения. Поэтому приведенная ниже конструкция выполнена правильно:

P001:=Positive\_EOT\_1;(\* P001 = Positive End of Travel Axis 1 \*)

Однако следующая конструкция является неправильной:

```
Positive_EOT_1:=1;
```

Синтаксический анализатор Local Logic выдает сообщение об ошибке, если программа пытается выполнить запись в переменную, доступную только для чтения, или пытается прочитать переменную, доступную только для записи.

Кроме того, переменные Local Logic имеют атрибут размера от логического (булева) (1 бит) до двойного целого (64 бита). Синтаксический анализатор Local Logic выдает предупредительное сообщение, если логической (булевой) переменной присваивается небулево значение. Данное сообщение информирует, что данные могут быть потеряны из-за усечения, если такое назначение произойдет. Следует учитывать, что двойные целые переменные (64 бита) могут быть использованы только для назначения результата операции умножения или для числителя операции деления или получения модуля.

Дополнительная информация по переменным Local Logic содержится в главе 13. Кроме того, Таблица переменных Local Logic (LLVT) в ПО программирования содержит информацию по размеру переменных, их типу и по свойствам чтение / запись.

## Операторы

Local Logic предоставляет три класса операторов. Это операторы арифметические, операторы отношения и побитовые логические операторы. Ниже дано краткое описание каждого оператора. Более детальное обсуждение операторов содержится в главе 12.

### Арифметические операторы

Local Logic предоставляет пользователю возможность выполнять базовые арифметические операции. Этот язык поддерживает операции с 32-битовыми целыми числами и ограниченное использование 64/32-битовых операций так, где представляется возможным сохранить точность. Все арифметические функции, кроме функции ABS, требуют два операнда.

Local Logic поддерживает операции сложения, вычитания, умножения, деления, получения абсолютного значения и модуля.

Примеры конструкций:

```
P010 := Commanded_Velocity_1 - P009; (* P010=Заданная скорость оси 1 – P009*)
```

Следует учесть, что следующая конструкция является неправильной:

```
Commanded_Velocity_1 := P010 - P009; (* Commanded_Velocity_1=P010-P009*)
```

Ошибка конструкции в том, что математическое выражение пытается приписать результат (P010-P009) переменной Commanded\_Velocity\_1, которая доступна только для чтения.

Сохранение промежуточных результатов в регистрах параметров обеспечивает гибкость, требуемую для работы со сложными математическими выражениями.

Например, следующая конструкция является неправильной, поскольку она содержит более одной операции (умножение и вычитание):

```
P005:= Torque_Limit_1 *( P001 – P010);
```

Чтобы получить тот же результат, пользователь должен ввести следующее:

```
P004:= P001 – P010;  
P005:= Torque_Limit_1 * P004;
```

## Операторы отношения

Операторы отношения сравнивают два операнда в условном операторе. Соответствующими операторами отношения являются следующие: < (меньше чем), > (больше чем), <= (меньше чем или равно), >= (больше чем или равно), = (равно) и <> (не равно). Выполнение тела оператора IF происходит в том случае, если условное выражение становится "истинным". В примере переменная Torque\_Limit\_1 принимает значение 10000, если переменная Block\_1 равна 3. Если значение Block\_1 не равно 3, то выражение расценивается как "ложь", и выполнение программы продолжается после оператора END\_IF.

### Пример:

```
IF Block_1 = 3 THEN
  Torque_Limit_1 := 10000;    (* Задание предела момента вращения = 100% @ Block
3 *)
END_IF;
```

Вложение IF-операторов позволяет реализовать сложные выражения. Например, чтобы установить предел момента вращения оси 1 (Torque\_Limit\_1) на 10000=100% (т.е. тот же масштаб, что при обработке команды AQ), когда действует блок 3 программы движения и заданная скорость оси 1 (Commanded\_Velocity\_1) меньше 1000, следует задать следующую конструкцию:

```
IF Block_1 = 3 THEN
  IF Commanded_Velocity_1 < 1000 THEN
    Torque_Limit_1 := 10000;    (* Задание предела момента вращения = 100% @
Block 3 *)
  END_IF;
END_IF;
```

## Побитовые логические операторы

Побитовые логические операторы накладывают маску или инвертируют отдельный бит или группу битов. Действующими конструкциями являются операторы BWAND (и), BWOR (или), BWXOR (исключающее или) и BWNOT (одинарное дополнительное). BWAND, BWOR и BWXOR требуют два операнда. Оператор BWNOT требует один операнд.

В примере ниже соответствующий сегмент кода снимает копию нескольких битов в слове CTL\_1\_to\_32 и назначает ее регистру параметров.

Затем происходит проверка последних четырех значащих битов этого значения; параметру P002 назначается значение 4985, если какие-нибудь биты установлены.

```
P001 := CTL_1_to_32 BWAND 16#0000A005;
IF P001 BWAND 16#F THEN
  P002 := 4985;
END_IF;
```

В частности, данные операторы, выполняют следующие действия. Первый оператор использует 16#0000A005 в качестве маски. Эта маска соответствует двоичному выражению, показанному ниже:

```
16# 0000A005 = 2#0000 0000 0000 0000 1010 0000 0000 0101
```

Таким образом, оператор

```
P001:=CTL_1_to_32 BWAND 16#0000A005
```

выделяет биты 1, 3, 14 и 16 из CTL\_1\_to\_32 и помещает результат в P001.

Следующий оператор предпринимает побитовую проверку, установлены ли какие-нибудь биты среди последних значащих битов. Значение проверки соответствует двоичному выражению, показанному ниже:

```
16#F = 2#1111
```

Таким образом, оператор

```
IF P001 BWAND 16#F THEN
```

выполняет побитовую проверку последнего значащего байта в P001, и, если какой-нибудь из битов последнего значащего байта установлен в логическую единицу (значение = 1), то происходит обработка операторов в блоке IF.

В данном примере, поскольку CTL\_1\_to\_32 было закрыто маской в предыдущем операторе, то IF-условие проверяет только биты 1 и 3 из CTL\_1\_to\_32.

## Связь "программа Local Logic / ПЛК / программа движения"

Программа Local Logic или основной ПЛК обмениваются данными с программой движения, используя параметры, CTL-биты и номера блоков программы движения. Использование этих методов:

- **Данные параметров** – Данные параметров могут быть получены из программ Local Logic, с основного ПЛК и из программ движения. Данные параметров схожи с переменными в программах. Например, программа движения может использовать для команды DWELL (ЗАДЕРЖКА) интервал времени, который задается параметром. Программа Local Logic или основной ПЛК могут записать этот параметр, который будет определять время DWELL (ЗАДЕРЖКА) в программе движения.
- **CTL-биты** – CTL-биты позволяют программе Local Logic или основному ПЛК выдать сигнал программе движения на начало какого-либо действия. Например, CTL-биты используются для того, чтобы управлять ходом программы движения при помощи команды JUMP (ПЕРЕХОД).
- **Номера блоков программы движения** – Программа движения (если номера блоков используются внутри программы движения) предоставляет номер текущего блока программе Local Logic или основному ПЛК. Этот текущий номер блока может быть использован в программе Local Logic или в основном ПЛК для того, чтобы выполнить определенное действие только в пределах определенной секции программы движения.

Сигнализирующие конструкции между программами (ПЛК, движения и Local Logic) позволяют им взаимодействовать и выполнять совместные действия. Такие сигнализирующие конструкции играют важную роль в примерах программирования, приведенных ниже. Дополнительная информация по связи ПЛК и программ движения и по взаимодействию программ см. в гл. 5 и 7.

## Примеры программирования Local Logic

В предыдущих разделах коротко описаны базовые конструкции локального логического языка. Чтобы проиллюстрировать эти положения, далее дано несколько примеров программ. Эти программы предназначены только для иллюстрации и не предоставляют все необходимое для функционального применения. Дополнительная информация по имеющимся локальным логическим операторам, по переменным и конструкциям содержится в главах 12 и 13.

## Пример программы по ограничению момента вращения

Приведенный ниже пример иллюстрирует метод использования локальной логики совместно с программой движения, чтобы выполнить ограничение момента вращения, основываясь на номере блока в программе движения. В данном примере сервоось 1 накручивает гайку на резьбовой стержень. Вначале ось двигается слегка назад, чтобы улучшить прилегание резьбы гайки и стержня. Это движение имеет границу момента вращения, установленную на максимум. Затем гайка закручивается до затягивания с моментом, ограниченным 30% от максимального значения. Во время этой операции точка назначения команды движения обычно не достигается, и ось останавливается, когда сила трения превышает предел по моменту вращения. Затем, чтобы устранить все натяжения в механических узлах, момент вращения устанавливается на 0; через 0,1 секунды включается сигнал "операция закручивания завершена". Когда ПЛК включит сигнал "захват гайки открыт", ось двигается к начальной позиции при полном моменте вращения.

### Локальная логическая программа ограничения момента вращения

```
(*НАЗНАЧЕНИЕ БИТОВ УПРАВЛЕНИЯ*)
(*STL01 - НАЧАЛО ОПЕРАЦИИ ЗАКРУЧИВАНИЯ - СИГНАЛ ОТ ПЛК*)
(*STL02 - КОНТУР ПРОГРАММЫ ДВИЖЕНИЯ 1*)
(*STL03 - ЗАХВАТ ГАЙКИ ОТКРЫТ - СИГНАЛ ОТ ПЛК*)
(*STL04 - ОПЕРАЦИЯ ЗАКРУЧИВАНИЯ ЗАВЕРШЕНА - СИГНАЛ НА ПЛК*)

(*НАЗНАЧЕНИЕ ДАННЫХ ПАРАМЕТРОВ*)
(*P001 - УСКОРЕНИЕ ОБРАТНОГО ДВИЖЕНИЯ*)
(*P002 - СКОРОСТЬ ОБРАТНОГО ДВИЖЕНИЯ*)
(*P003 - РАССТОЯНИЕ ОБРАТНОГО ДВИЖЕНИЯ*)
(*P004 - УСКОРЕНИЕ ВРАЩАТЕЛЬНОГО ДВИЖЕНИЯ ГАЙКИ*)
(*P005 - СКОРОСТЬ ВРАЩАТЕЛЬНОГО ДВИЖЕНИЯ ГАЙКИ*)
(*P006 - РАССТОЯНИЕ ВРАЩАТЕЛЬНОГО ДВИЖЕНИЯ ГАЙКИ*)
(*P007 - СЧЕТЧИК, ИСПОЛЬЗУЕМЫЙ ДЛЯ ТАЙМЕРА ВЫДЕРЖКИ ПОСЛЕ СНЯТИЯ
НАТЯЖЕНИЯ*)

(*ЛОКАЛЬНЫЙ ЛОГИЧЕСКИЙ КОД*)
IF Program_Active_1 = 1 THEN (*программа движения оси 1 активна*)
  IF Block_1 = 10 THEN (*программа движения ожидает команду
пуска*)
    P007:=0; (*сброс счетчика*)
    STL04:=0; (*очищает "операция закручивания
выполнена"*)
    Torque_Limit_1:= 10000; (*устанавливает границу момента вращения
на максимальное значение 100%*)
  END_IF;
  IF Block_1=30 THEN (*основное закручивающее движение гайки*)
    Torque_Limit_1:= 3000; (*установка границы момента вращения
30%*)
  END_IF;
  IF Block_1=40 THEN (*программа движения ждет сигнал
отпускания*)
    Torque_Limit_1:= 0; (*устанавливает границу момента вращения
=0*)
    P007:=P007 + 1; (*обновление счетчика времени*)
    IF P007=50 THEN (*ожидание 100 мс*)
      STL04:=1; (*устанавливает "операция закручивания
выполнена"*)
    END_IF;
  END_IF;
  IF Block_1=50 THEN
    Torque_Limit_1:= 10000; (*устанавливает границу момента вращения
на максимальное значение 100%*)
  END_IF;
END_IF;
```

**Программа движения для ограничения момента вращения**

(\* Программа движения для ограничения момента вращения \*)

```

Program 1 AXIS1
10:WAIT   STL01          (* Ожидание команды пуска *)
    ACCEL P001          (* Устанавливает ускорение движения *)
    VELOC P002          (* Устанавливает скорость движения *)
20:PMOVE  P003,INCR,LINEAR (* Обратное движение *)
    ACCEL P004          (* Устанавливает ускорение движения *)
    VELOC P005          (* Устанавливает скорость движения *)
30:CMOVE  P006,INCR,LINEAR (* Основное закручивающее движение гайки *)
40:WAIT   STL03          (* Ожидание сигнала отпускания *)
50:PMOVE  0,ABS,LINEAR  (* Движение к начальной позиции *)
60:JUMP   STL02,10      (* Переход к началу, если программа
циклическая *)

EndProg

```

**Пример программы планировщика коэффициента усиления**

Приведенный ниже пример показывает метод использования локальной логики для улучшения простого алгоритма планировщика коэффициента усиления. При изменении алгоритма, который динамически изменяет характеристики управления, следует соблюдать осторожность. Во многих случаях динамическое изменение характеристик управления может привести к неустойчивости управляемого процесса. Следует отметить, что в приведенной ниже программе управляющая переменная *Velocity\_Loop\_Gain* (*Коэффициент обратной связи по скорости*) может быть записана неоднократно в одном и том же цикле. Однако активным является последнее значение данного цикла, поскольку переменные обновляются при завершении выполнения Local Logic. Детальное описание управляющих переменных и выходов Local Logic см. в гл. 12 и 13.

**Локальная логическая программа планировщика коэффициента усиления**

```

(* Данный пример показывает путь улучшения простейшего *)
(* алгоритма планирования коэффициента усиления. Базовый *)
(* коэффициент обратной связи по скорости установлен равным 12. *)
(* Однако этот коэффициент изменяется в зависимости от позиции *)
(* и заданной скорости. *)

Velocity_Loop_Gain_1 := 12;      (* Базовый коэффициент обратной связи по
скорости *)
IF Actual_Position_1 > 500 THEN  (* Если текущая позиция оси 1 > 500 *)
    IF Commanded_Velocity_1 < 10 (* Если Заданная скорость оси 1 < 10 *)
        Velocity_Loop_Gain_1 := 16; (* Коэффициент обратной связи по скорости оси 1
= 16 *)
    END_IF;
END_IF;

IF Actual_Position_1 > 1000 THEN (* Если текущая позиция оси 1 > 1000 *)
    IF Commanded_Velocity_1 < 10 (* Если Заданная скорость оси 1 < 10 *)
        Velocity_Loop_Gain_1 := 20; (* Коэффициент обратной связи по скорости оси 1
= 20 *)
    END_IF;
END_IF;

IF Actual_Position_1 > 1500 THEN (* Если текущая позиция оси 1 > 1500 *)
    IF Commanded_Velocity_1 < 10 THEN (* Если Заданная скорость оси 1 < 10 *)
        Velocity_Loop_Gain_1 := 24; (* Коэффициент обратной связи по скорости оси
1 = 24 *)
    END_IF;
END_IF;

```

## Пример программы программируемого концевого переключателя

Приведенный ниже пример показывает метод использования локальной логики для выполнения функции программируемого концевого переключателя. Этот программируемый конечной переключатель включает / выключает выход, основываясь на текущей позиции двигателя и на номере блока в программе движения.

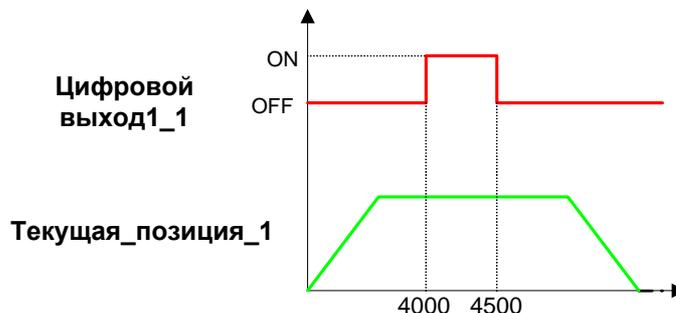


Рис. 11-1. Пример программируемого концевого переключателя

### Локальная логическая программа программируемого концевого переключателя

```
(* Данный пример показывает, как включить Цифровой выход # 1 оси 1, *)
(* когда текущая позиция оси 1 оказывается между 4000 и 4500, *)
(* но чтобы при этом программа находилась на блоке #4. *)
(* Пример демонстрирует функциональные качества, которые часто *)
(* дополняются использованием быстродействующего счетчика (HSC) *)
(* и PRESET (ПРЕДВАРИТЕЛЬНЫЕ УСТАНОВКИ). Данный пример дает *)
(* разрешение 2 мс. *)
Digital_Output1_1 := 0; (* Цифровой выход оси 1 выключен *)
IF Block_1 = 4 THEN      (* Если текущий номер блока для оси 1 равен 4 *)
  IF Actual_Position_1 > 4000 THEN      (* Если текущая позиция оси 1 > *)
    4000 *)
    IF Actual_Position_1 < 4500 THEN      (* Если текущая позиция оси 1 < *)
      4500 *)
      Digital_Output1_1 := 1; (* Цифровой выход оси 1 включен *)
    END_IF;
  END_IF;
END_IF;
```

Сегмент программы движения, соответствующий данной локальной логической программе, показан ниже.

### Сегмент программы движения для примера программируемого концевого переключателя

```
Program 1 Axis1          (* Программа номер 1 для оси 1 *)
3: PMOVE 0,ABS,LINEAR    (* Движение к начальной позиции *)
4: PMOVE 8000,ABS,LINEAR (* Движение к позиции 8000 *)
5: PMOVE -8000,ABS,LINEAR (* Движение к позиции -8000 *)
EndProg
```

## Пример программы триггерного выхода, основанного на позиции

Приведенный далее пример иллюстрирует метод использования Local Logic для переключения таймерного выхода, основываясь на текущей позиции двигателя. Обратите внимание, что таймерная реализация использует счетчик внутри программы. Счетчик отсчитывает количество выполнения программы с момента последнего сброса счетчика. Поскольку локальные логические программы выполняются в течение каждого цикла дискретизации контура позиционирования, то период счетчика базируется на этом цикле. Данный пример использует цифровой сервопривод с циклом дискретизации контура позиционирования 2 мс. Поэтому счетчик будет работать с приращением 2 мс. Длительности цикла дискретизации контура позиционирования для других конфигураций см. в гл. 1. Дополнительно Local Logic позволяет программе записывать переменную неоднократно внутри программы. На выход выдается последнее состояние, которое переменная получила перед завершением программы (см. гл. 12 раздел "Выходы и команды Local Logic"). Это является важным для приведенной ниже программы. Второй блок IF-THEN-END\_IF включает цифровой выход для оси 1 (Digital\_Output1\_1), когда текущая позиция оси 1 (Actual\_Position\_1) превышает 4000, независимо от текущего значения таймера (P008). Однако последний блок IF-THEN-END\_IF проверяет текущее значение таймера (P008) и выключает цифровой выход 1 для оси (Digital\_Output1\_1), если значение таймера превышает 500. Данное применение показано графически на рис. 11-2.

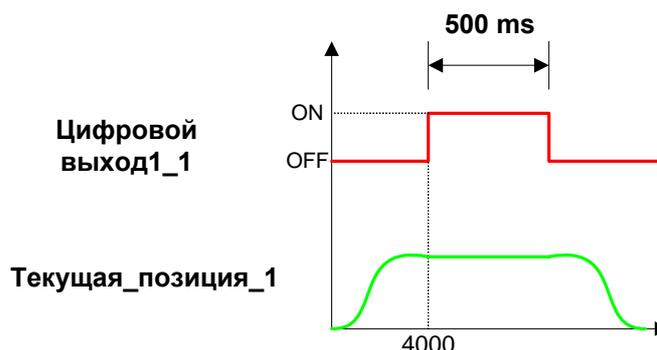


Рис. 11-2. Пример таймерного выхода, базирующегося на позиции

### Локальная логическая программа таймерного выхода, базирующаяся на позиции

```
(* Данный пример показывает способ включения таймерного выхода, *)
(* основываясь на текущей позиции оси 1. Когда Actual_Position_1 *)
(* (Текущая позиция 1) превышает 4000, происходит включение *)
(* Digital_Output1_1 (Цифровой выход 1) на промежуток времени *)
(* длительностью 500 мс. *)

IF Block_1 = 4 THEN (* Блок 4 в программе движения сбрасывает таймер *)
*)
  P008 := 0;          (* Сброс таймера (P008) *)
  Digital_Output1_1 := 0; (* следует убедиться, что выход выключен,
  прежде чем начать движение *)
  P009 := 0;          (* P009 отслеживает состояние цифрового выхода *)

END_IF;

IF Block_1=5 THEN
  IF Actual_Position_1 > 4000 THEN (* Actual_Position_1 превышает
  4000 *)
    Digital_Output1_1 := 1;          (* переход Digital_Output1_1
    (Цифровой выход 1_1) в состояние ON (ВКЛ.) *)
    P009 := 1;
  END_IF;
END_IF;

IF P009 = 1 THEN(* если Digital_Output1_1 в состоянии ON (ВКЛ.) *)
  P008 := P008 + 2;          (* приращение таймера на 2 мс *)
  IF P008 > 500 THEN          (* если таймер превышает 500 мс *)
    Digital_Output1_1 := 0; (* переход Digital_Output1_1 (Цифровой
    выход 1) в состояние OFF (ВЫКЛ.) *)
    P009 := 0;          (* помните, что только последняя запись в цикле
    Local Logic имеет силу для цифрового выхода *)
  END_IF;
END_IF;
```

Сегмент программы движения, соответствующий данной локальной логической программе, показан ниже.

### Сегмент программы движения для примера программы с таймерным выходом, основанном на позиции

```
Program 1 Axis1          (* Программа номер 1 для оси 1 *)
4: PMOVE 0, ABS, LINEAR  (* Движение к начальной позиции *)
5: PMOVE 12000, ABS, S-CURVE (* Движение к позиции 12000 *)
EndProg
```

## Пример программы с кадрюющими stroбaми

Приведенный ниже пример показывает метод использования локальной логики для выполнения функции кадрюющих stroбa. В примере игнорируется команда stroба, пока текущая позиция двигателя не окажется в пределах окна (Текущая позиция больше 4000, но меньше 5000). Если позиция двигателя оказывается внутри данного окна, то первое же появление stroба вызывает захват текущей позиции двигателя в регистре stroбa. Это применение показано графически на рис. 11-3.

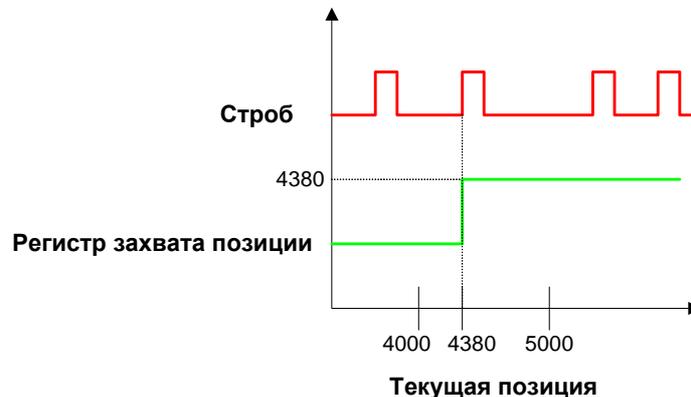


Рис. 11-3. Пример кадрюющих stroбa

### Локальная логическая программа кадрюющих stroбa

(\* Кадрюющие stroбы - Пример локальной логической программы \*)  
 (\* P009 представляет собой позицию stroба, считанную из Strobel\_Position\_1 \*)

(\* STL13 используется в качестве флага появления stroба - он должен быть сконфигурирован для управления Local Logic в конфигурации аппаратного обеспечения VersaPro \*)

```
IF Block_1 = 6 THEN
  P009 := 0; (* Установка P009 на 0 *)
  STL13 := 0; (* сброс флага появления stroба *)
END_IF;
Reset_Strobel_1 := 0; (* Установка бита Reset Strobe (Сбросить stroб) в
состояние off (выкл.) по умолчанию *)
IF Strobel_Flag_1 = 1 THEN (* Если stroб появляется в области "окна",
*)
  IF Actual_Position_1 > 4000 THEN (* установка P009 = позиция stroба
*)
    IF Actual_Position_1 < 5000 THEN
      P009 := Strobel_Position_1;
      STL13 := 1; (* Установка флага появления stroба в
состояние ON (ВКЛ.) *)
    END_IF;
  END_IF;
  Reset_Strobel_1 := 1; (* Если появляется stroб, происходит
установка бита Reset Strobe (Сбросить stroб). *)
END_IF;
```

Сегмент программы движения, соответствующий данной локальной логической программе, показан ниже.

### Сегмент программы движения для примера с кадрюющими stroбaми

```
Program 1 AXIS1 (* Программа номер 1 для оси 1 *)
  PMOVE 0,ABS,LINEAR (* Движение к начальной позиции *)
6: DWELL 10
7: PMOVE 10000,ABS,S-CURVE (* Движение к позиции 10000 *)
EndProg
```



## Введение

Данная глава описывает синтаксис языка программирования Local Logic, правила и языковые элементы. Этот язык использует конструкции, базирующиеся на тексте свободного формата, полученные из структурированного текстового стандарта IEC 1131. В разделах дано описание имеющихся команд и их синтаксиса.

## Синтаксические элементы

Синтаксис локального логического языка описывается в расположенных далее разделах. Этот синтаксис легко изучается; он обеспечивает богатый набор возможностей, позволяющих пользователю решать задачи программирования. В главе 11 дано несколько примеров, которые способствуют пониманию синтаксиса и его применения. В первое время пользователь может также использовать раздел "Построение вашей первой программы Local Logic" из главы 10 и примеры программ из главы 11 в качестве дополнительной помощи.

## Числовые константы

Язык локального логического программирования поддерживает десятичные, шестнадцатеричные и двоичные константы. Модуль DSM обрабатывает все константы как 32-битовые двойные, дополнительные целые значения со знаком. Между цифрами могут быть вставлены одиночные символы подчеркивания (т.е. 16#7fff\_ffff), чтобы улучшить читаемость больших чисел.

Десятичные константы должны быть в диапазоне от -2147483648 до 2147483647. Поддерживаются только целые значения, поэтому константы не должны иметь десятичной точки. Таким образом, как и во всех системах, основанных на целых числах, десятичные точки здесь подразумеваются, и программист должен отслеживать это, если требуются расчеты с дробными числами.

### Примеры:

523 Положительная десятичная константа  
-1048 Отрицательная десятичная константа  
1\_745\_245 Положительная десятичная константа с подчеркиванием

Шестнадцатеричные константы (основание 16) идентифицируются приставкой 16#; они должны иметь значение, которое может быть представлено 32 битами (8 шестнадцатеричных знаков). Шестнадцатеричные константы не могут иметь знак (+/-). Шестнадцатеричные знаки A-F не чувствительны к регистру, для них можно использовать как большие, так и малые буквы.

### Примеры:

16#FFFF Шестнадцатеричная константа  
16#7fff\_ffff Шестнадцатеричная константа с подчеркиванием

Двоичные константы (основание 2) идентифицируются приставкой 2#, они должны иметь значение, которое может быть представлено 32 битами (32 двоичных знака). Двоичные константы не могут иметь знак (+/-).

**Примеры:**

2#1010      Двоичная константа

2#11111110\_11101101\_10111110\_11101111      Двоичная константа с подчеркиванием

Локальная логическая программа может иметь не более 50 *уникальных* констант, значение которых более 2047 или менее – 2048. Если локальная логическая программа объявляет более 50 уникальных констант, то построитель программы выдает сообщение об ошибке. Большинство программ использует менее 50 констант, поэтому, в общем случае, это не является ограничением.

## Локальные логические переменные

Локальный логический язык поддерживает множество предварительно определенных переменных, которые позволяют осуществлять доступ к данным В/В модуля DSM, к STL-битам и к другой информации по статусу и управлению. Подробное описание набора локальных логических переменных содержится в главе 13. Каждая переменная имеет два атрибута – размер и направление. Переменные Local Logic имеют размер от 1 бита (битовые операнды) до 64 битов.

Все регистры параметров Local Logic относятся к одному из следующих типов:

- **Double integer (Двойные целые)** переменные содержат 32-битовые значения со знаком (от –2147483648 до 2147483647). Имеется 256 регистров параметров (P000-P255).
- **Long integer (Длинные целые)** переменные содержат 64-битовые значения со знаком ( $\pm 9.22 \times 10^{18}$ ). Длинные целые переменные являются уникальными в том, что они могут быть использованы только для результата умножения или в качестве числителя в операции деления или получения модуля. Имеется 8 регистров длинных целых (D00-D07).

Все переменные Local Logic имеют один из следующих атрибутов направления:

- Переменные **Read-only (Только для чтения)** не могут быть использованы в качестве адресата операции присваивания.
- Переменные **Write-only (Только для записи)** могут быть использованы *только* в качестве адресата оператора присваивания.
- Переменные **Read-write (чтение-запись)** могут быть использованы как источник и как адресат.

Список все атрибутов размера и направления, переменных Local Logic см. в главе 13.

## Локальные логические операторы

Язык Local Logic поддерживает два типа операторов: операторы присваивания и условные операторы. Программа Local Logic поддерживает 150 операторов. Блок проверки Local Logic выдает сообщение об ошибке, если будет превышена граница в 150 строк. Предупреждение выдается при превышении программой Local Logic размера в 100 строк. Предупредительное сообщение может быть выключено директивой #pragma. См. дополнительную информацию в разделах #pragma. Операторы программы разделяются точками с запятыми.

## Операторы присваивания Local Logic

Операторы присваивания позволяют осуществлять простые арифметические и битовые операции с результатом, назначаемым переменной. Оператор присваивания имеет приведенный ниже формат.

**<адресат> := <выражение>;**

Оператор **<адресат>** может представлять собой любую переменную типа "чтение-запись" или "только запись". **<выражение>** может быть простой константой или переменной, математической или побитовой логической операцией с двумя операндами, функцией ABS или побитовой NOT-операцией. Переменные типа "только запись" не могут быть **выражением** для операции присваивания.

### Примеры:

P032 := Strobe1\_Position\_1 + 5000;      ← *Правильная конструкция.*

P001 := ABS(Analog\_Input1\_1);      ← *Правильная конструкция.*

Reset\_Strobe1\_1 := BWNOT Strobe1\_Flag\_1;      ← *Правильная конструкция.*

P040 := 2#11111010\_1011000;      ← *Правильная конструкция.*

P011 := 3 \* Strobe1\_Position\_1 + 20;      ← *НЕПРАВИЛЬНАЯ конструкция – слишком много операций.*

Если требуются сложные операции, то следует использовать последовательность операций, использующих регистры параметров для сохранения промежуточных результатов.

### Примеры:

Чтобы задать Velocity\_Loop\_Gain\_2 равной  $(1+75000/Actual\_Velocity\_2)$ , программист использует последовательность операторов, аналогичную показанной ниже:

```
P012 := 75000 / Actual_Velocity_2;  
Velocity_Loop_Gain_2 := 1 + P012;
```

Компоновщик выдаст предупредительное сообщение, если булева (логическая), переменная использована в качестве адресата для выражения, содержащего не булевы переменные или константы со значением, отличающимся от нуля или единицы. Предупреждение выдается потому, что модуль DSM назначит логической переменной значение последнего значащего бита выражения.

## Условные операторы Local Logic

Условные операторы позволяют выполнять условные программы, основанные на простых операциях отношения и побитовых логических операциях. Условный оператор имеет приведенный ниже формат.

**IF <выражение> THEN**

Локальные логические операторы

**END\_IF;**

<выражение> может содержать константу, переменную, операцию отношения или побитовую логическую операцию с двумя переменными или побитовое дополнение константы или переменной. Переменные типа "только запись" не разрешаются в данном выражении. Если выражение отношения приобретает значение "логической единицы" ("истина") или если побитовая операция, переменная или константа имеет значение, отличное от нуля, то происходит выполнение операторов Local Logic в теле оператора IF. В теле оператора IF может находиться любое число программных операторов (действует общее ограничение). Каждый оператор IF-THEN должен иметь соответствующий END\_IF.

### Примеры:

IF P226 THEN ← Правильная конструкция.

IF CTL\_1\_to\_32 BWAND 2#1010 THEN ← Правильная конструкция.

IF Strobe1\_Level\_1 = TRUE THEN ← Правильная конструкция.

IF BWNOT P100 THEN ← Правильная конструкция.

IF BWNOT P001 <> P002 THEN ← НЕПРАВИЛЬНАЯ конструкция – слишком много операций.

Операторы IF могут иметь до 8 уровней вложения. При подсчете количества программных операторов операторы IF-THEN и END\_IF считаются как два отдельных оператора.

**Таблица 12-1. Действующие операторы**

Тип оператора	Действующие операторы	
Условные операторы	Операторы отношения	<, >, <=, >=, <>, =
	Побитовые логические операторы	BWAND, BWOR, BWXOR, BWNOT
Операторы присваивания	Арифметические	+, -, /, *, MOD
	Побитовые логические операторы	BWAND, BWOR, BWXOR, BWNOT
	Функция получения абсолютного значения	ABS ()

## Пустое пространство

Пропуски символов, знак конца строки и табуляторы рассматриваются как пустое пространство. Пустые пространства игнорируются, за исключением тех случаев, когда они разделяют соответствующие синтаксические элементы; эти пространства могут быть использованы для улучшения читаемости программ за счет отступов и пустых строк.

## Комментарии

Комментарии могут быть использованы для добавления в программу такой информации, которая не будет учитываться при выполнении программы Local Logic. Поддерживаются два типа комментариев.

Пара символов "(" вводит обычный комментарий, который заканчивается парой символов ")". Эти комментарии могут появиться везде, где может быть поставлено пустое пространство, например, внутри локального логического оператора или за ним, отдельно в строке или занять несколько строк. Эти комментарии не вкладываются.

Пара символов "//" вводит комментарий в одиночной строке. Весь текст, расположенный за // до конца строки, игнорируется при выполнении программы Local Logic.

**Примечание:** Следует отдавать себе отчет, что можно ввести локальную логическую программу и непреднамеренно включить в комментарий исполняемый код. Наиболее распространенный сценарий этого следующий:

(\* Данный пример показывает, как включить Цифровой выход # 1 оси 1, когда \*)  
 (\* текущая позиции оси 1 оказывается между 4000 и 4500, но чтобы при этом \*)  
 (\* программа находилась на блоке #4. Пример демонстрирует функциональные \*)  
 (\* качества, которые часто дополняются использованием быстродействующего \*)  
 (\* счетчика (*HSC*) и *PRESET* (**ПРЕДВАРИТЕЛЬНЫЕ УСТАНОВКИ**). \*)  
 (\* Данный пример дает разрешение 2 мс. )

```
Digital_Output1_1 := 0;          (* Цифровой выход оси 1 выключен *)
IF Block_1 = 4 THEN             (* Если текущий номер блока для оси 1 равен 1 *)
```

В приведенном выше сегменте программы структура конца комментария в строке, показанной в целях иллюстрации курсивом и жирным шрифтом, неправильная, т.к. отсутствует звездочка в структуре завершения строки. Эта ошибка вызывает рассмотрение следующей строки также как комментария. Таким образом, оператор Digital\_Output\_1:=0 рассматривается как комментарий и не выполняется. Цветная схема в редакторе Local Logic может быть очень полезной для нахождения ошибок такого типа. Цветовая схема по умолчанию окрашивает комментарии в иной цвет, чем операторы программирования. Таким образом, пользователь получает визуальный метод обнаружения таких ошибок. Информацию по изменению цветов по умолчанию в редакторе см. в главе 2.

## Директива PRAGMA

Директива #pragma используется для конфигурации синтаксического анализатора Local Logic. Эта директива НЕ требуется для работы синтаксического анализатора. Однако если пользователь желает выключить предупреждающие сообщения, то директива #pragma позволяет сделать это. Директива #pragma ДОЛЖНА быть первой строкой программы. Кроме того, перед директивой не должно быть пустого пространства.

Чтобы выключить ВСЕ предупреждения Local Logic, выдается следующая команда:

```
#pragma errorsonly 1
или
#pragma errorsonly ON
```

Чтобы снова включить предупреждающие сообщения, следует либо удалить директиву, либо изменить ее следующим образом:

```
#pragma errorsonly 0
или
#pragma errorsonly OFF
```

## Локальные логические ключевые слова и операторы

Приведенные далее ключевые слова и операторы имеют особое значение в языке программирования Local Logic. Ключевые слова являются чувствительными к регистру клавиатуры и используют только большие буквы. Они описываются подробнее в следующих разделах.

**Таблица 12-2. Ключевые слова Local Logic**

ABS	TRUE	+	>
BWAND	FALSE	-	<
BWOR	IF	/	>=
BWXOR	THEN	*	<=
BWNOT	END_IF	16#	=
ON	MOD	2#	<>
OFF	:	:=	

## Включение и выключение Local Logic

Выполнение Local Logic включается Q-битом ПЛК. Например, если DSM сконфигурирован с начальной ссылкой %Q как %Q0001, то битом включения Local Logic является %Q0002 (начальная ссылка + смещение 1). Имя программы Local Logic должно быть задано в программе конфигурации аппаратного обеспечения, а в поле Enabled/Disabled (Включение / Выключение) Local Logic должно быть задано Enabled. Более детальное описание конфигурации Local Logic при конфигурации аппаратного обеспечения см. в гл. 10.

Программа Local Logic выполняется только в том случае, если ПЛК находится в режиме RUN. Если ПЛК находится в режиме STOP или если Q-бит включения Local Logic выключен, то выполнение программы Local Logic прекращается, и все цифровые выходы, биты управления (Толчок, Остановка подачи, Сбросы строга, Включение слежения) и STL-биты, управляемые со стороны Local Logic, выключаются.

Попытка выполнить Local Logic в первом цикле ПЛК ведет к сообщению об ошибке. Например, переключение из режима Stop (Останов) в режим Run (Выполнить) при включенном бите Local Logic Enable (Включить Local Logic) ведет к сообщению об ошибке, и программа Local Logic не выполняется. Для выполнения программы Local Logic следует переключить Q-бит Enable.

### Примечание

Процессор Local Logic не будет действовать, если включены какие-либо *заказные* функции Local Logic через раздел Advanced Parameters (Расширенные параметры) в ходе конфигурации аппаратного обеспечения VersaPro. Заказная функция обычно отсутствует; она разрабатывается для специального использования в GE Fanuc.

## Выходы / команды Local Logic

Битовые выходы команд DSM (**Толчок, Остановка подачи, Включение слежения и Сбросы строба**) подчиняются командам ПЛК или командам Local Logic. Поэтому этими битами управляют как ПЛК, так и Local Logic, т.е. управляющий битовый выход активен, если он включен либо со стороны ПЛК, либо со стороны Local Logic.

Команды AQ принимаются по принципу "действует последняя записанная". Например, если как ПЛК (%AQ), так и Local Logic выдадут команду Follower Ratio (Отношение слежения), то действовать будет последнее записанное значение.

Цифровые выходы лицевой панели DSM (реальные выходы, переключаемые DSM) конфигурируются индивидуально, для управления от Local Logic, либо от управления ПЛК, но не от обоих одновременно. Подробное описание конфигурации цифровых выходов см. в гл. 14.

Цифровые выходы Local Logic, команды прямого действия и командные биты обновляются в конце каждого цикла Local Logic (см. в гл. 13 список командных переменных и переменных цифровых выходов). Поэтому, если программа Local Logic проводит несколько раз запись в одном и том же цикле в одну и ту же командную переменную или в переменную цифрового выхода, то действовать будет последняя запись.

Пример программы внизу показывает неоднократную запись в одном цикле переменных *Jog\_Plus*, *Strobe\_Reset* и *Follower\_Ratio*. Во всех случаях активным является последнее значение.

### Пример:

```
Jog_Plus_1 := TRUE;    (* Включает Jog Plus (Толчок плюс) для оси 1 *)
Strobe_Reset1_3 := 0;  (* Выключает бит сброса строба 1 для оси 3 *)
(* Здесь находятся еще какие-то части программы *)
Follower_Ratio_A_1 := 10; (* Устанавливает Follower Ratio (Отношение
слежения) A на 10 *)
Jog_Plus_1 := FALSE;  (* Выключает Jog Plus для оси 1 *)
Strobe_Reset1_3 := 1;  (* Включает бит сброса строба 1 для оси 3 *)
Follower_Ratio_A_1 := 20;    (* Устанавливает Follower Ratio
(Отношение слежения) A на 20 *)
```

Для каждой из команд выхода, показанных выше, логический процессор действует в соответствии с последним значением, записанным в конце каждого цикла. Таким образом, *Jog\_Plus\_1* переходит в состояние OFF, *Strobe\_Reset1\_3* – в состояние ON, а *Follower\_Ratio\_A\_1* принимает значение 20.

## Арифметические операторы Local Logic

Язык Local Logic содержит привычные конструкции для проведения основных арифметических расчетов с целыми числами со знаком. Этот язык поддерживает 32-битовую арифметику в программе Local Logic и ограниченно - 64/32-битовую арифметику. Все операции требуют наличия двух операндов, кроме функции ABS, которая возвращает абсолютное значение переменной или числовой константы.

Таблица 12-3. Арифметические операторы

Оператор	Значение
+	Сложение
-	Вычитание
*	Умножение
/	Целочисленное деление
MOD	Модуль
ABS	Абсолютное значение

Арифметические выражения могут быть использованы только в операторах присваивания с одной операцией на оператора.

Арифметические операции не требуют функций преобразования типа данных, поскольку модуль движения автоматически делает эту операцию.

### Оператор +

Прибавляет источник 1 к источнику 2 и сохраняет результат в адресате.

#### Синтаксис

адресат := источник1 + источник2;

Синтаксис оператора + состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
Источник 1	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.
Источник 2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

**Переполнение** – Устанавливается, если результат сложения больше 2 147 483 647 или меньше -2 147 483 648. Module\_Status\_Code принимает значение 16#0095, которое показывает ошибку "только статус".

## Оператор -

Вычитает источник 2 из источника 1 и сохраняет результат в адресате.

адресат := источник1 - источник2;

Синтаксис оператора - состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
Источник 1	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.
Источник 2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

**Переполнение** – Устанавливается, если результат вычитания больше 2 147 483 647 или меньше -2 147 483 648. Module\_Status\_Code принимает значение 16#0095, которое показывает ошибку "только статус".

### Примечания

Оператор – не может быть использован как унарный (одноместный) оператор за исключением случаев с десятичной (основание 10) константой (т.е. P001 := -P003; является неправильным). Чтобы сделать переменную отрицательной, ее надо вычесть из нуля, т.е. P001 := 0 – P003;.

## Оператор \*

Оператор выполняет умножение со знаком источника 1 и источника 2, создавая 64-битовый результат со знаком. Результат может быть сохранен в 32- или в 64-битовом адресате.

### Синтаксис 1

адресат := источник1 \* источник2;

### Синтаксис 2

двойной адресат := источник1 \* источник2;

Синтаксис оператора \* состоит из следующих частей:

Часть	Описание
адресат	Любая логическая переменная, в которую может быть проведена запись.
двойной адресат	Любая из 64-битовых локальных логических переменных параметров (Dxx регистры).
Источник 1	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.
Источник 2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

**Переполнение** – Никогда не устанавливается.

### Примечания

Если результат назначен 32-битовой переменной, то значение сохраняется в последних 32 битах. Любое превышение отсекается.

Второй синтаксис может быть использован для операций умножения там, где результат может выходить за границы  $\pm 2$  млрд.

## Оператор /

Оператор производит целочисленное деление источника 1 на источник 2 и возвращает частное адресату. Регистр параметров двойной точности (64 бита) может быть использован как числитель.

### Синтаксис 1

адресат := источник1 / источник2;

### Синтаксис 2

адресат := двойной источник1 / источник2;

Синтаксис оператора / состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
Источник 1	Любая локальная логическая переменная, которая может быть прочитана, или числовая константа.
двойной источник 1	Любая из 64-битовых локальных логических переменных параметров (Dxx регистры).
Источник 2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

**Переполнение** – См. **примечания** ниже.

### Примечания

При делении на ноль Module\_Status\_Code устанавливается на 16#2093. При переполнении результата деления Module\_Status\_Code устанавливается на 16#2094.

Переполнение результата деления или деление на ноль являются ошибками **Stop Fast (Быстрый Останов)**. Выполнение программы Local Logic немедленно прекращается; прекращается также движение заданием команды нулевой скорости сервоустройства.

Переполнение результата деления происходит в том случае, если частное от операции деления не может быть правильно представлено 32-битовым значением со знаком. Это может произойти *только* при использовании двойного операнда (Dxx регистры) в качестве числителя. Деление на ноль происходит в том случае, когда знаменатель дроби равен нулю.

## Оператор MOD

Оператор MOD возвращает остаток, получающийся при целочисленном делении со знаком источника 1 на источник 2. Регистр параметров двойной точности (64 бита) может быть использован как числитель.

### Синтаксис 1

адресат := источник1 MOD источник2;

### Синтаксис 2

адресат := двойной источник1 MOD источник2;

Синтаксис оператора MOD состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
источник 1	Любая локальная логическая переменная, которая может быть прочитана, или числовая константа.
двойной источник 1	Любая из 64-битовых локальных логических переменных параметров (Dxx регистры).
источник 2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

**Переполнение** - См. примечания ниже.

### Примечания

При делении на ноль Module\_Status\_Code устанавливается на 16#2093. При переполнении результата деления Module\_Status\_Code устанавливается на 16#2094.

Модуль (остаток) рассчитывается при выполнении целочисленного деления, поэтому оператор MOD имеет те же условия ошибки, что и оператор деления.

Переполнение результата деления происходит в том случае, если частное от операции деления не может быть правильно представлено 32-битовым значением со знаком. Это может произойти *только* при использовании двойного операнда в качестве числителя. Деление на ноль происходит в том случае, когда знаменатель дроби равен нулю.

Переполнение результата деления или деление на ноль являются ошибками **Stop Fast (Быстрый Останов)**. Выполнение программы Local Logic немедленно прекращается; прекращается также движение заданием команды нулевой скорости сервоустройства.

## Функция ABS

Функция ABS возвращает значение без знака для переменного или постоянного параметра.

### Синтаксис

адресат := ABS(параметр);

Синтаксис оператора ABS состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
Параметр	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

**Переполнение** – Устанавливается, если операнд имеет значение – 2 147 483 648. Module\_Status\_Code принимает значение 16#0096, которое показывает ошибку "только статус".

## Побитовые логические операторы Local Logic

Все логические операции выполняются на побитовой основе; например, результат операции BWAND складывается из 32 "И"-операций между каждым из соответствующих битов операндов. Логические операторы определяются буквами "BW", чтобы показать, что они не булевы операторы.

Таблица 12-4. Побитовые логические операторы

Оператор	Значение
BWAND	Побитовое логическое "И"
BWOR	Побитовое логическое "ИЛИ"
BWXOR	Побитовое логическое исключающее "ИЛИ"
BWNOT	Побитовое логическое "НЕ" (одинарное дополнительное)

Выражения, использующие побитовые логические операторы, могут быть использованы в операторах присвоения или в условных операторах. В каждом выражении может быть использован только один побитовый логический оператор.

### Оператор BWAND

Выполняет по битово операцию "И" для источника 1 и источника 2.

#### Синтаксис 1

адресат := источник1 BWAND источник2;

#### Синтаксис 2

IF источник1 BWAND источник2 THEN

Синтаксис оператора BWAND состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
источник1	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.
источник2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

#### Примечания

Синтаксис 1 используется для присваивания (назначения); синтаксис 2 используется в условных выражениях.

## Оператор BWOR

Оператор BWOR возвращает результат побитового "ИЛИ" источника1 и источника2.

### Синтаксис 1

адресат := источник1 BWOR источник2;

### Синтаксис 2

IF источник1 BWOR источник2 THEN

Синтаксис оператора BWOR состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
источник1	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.
источник2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

### Примечания

Синтаксис 1 используется для присваивания (назначения); синтаксис 2 используется в условных выражениях.

## Оператор BWXOR

Оператор BWXOR возвращает результат побитового исключающего "ИЛИ" источника1 и источника2.

### Синтаксис 1

адресат := источник1 BWXOR источник2;

### Синтаксис 2

IF источник1 BWXOR источник2 THEN

Синтаксис оператора BWXOR состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
источник1	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.
источник2	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

### Примечания

Синтаксис 1 используется для присваивания (назначения); синтаксис 2 используется в условных выражениях.

## Оператор BWNOT

Оператор BWNOT возвращает одинарное дополнение исходного параметра.

### Синтаксис 1

адресат :=BWNOT источник;

### Синтаксис 2

IF BWNOT источник THEN

Синтаксис оператора BWNOT состоит из следующих частей:

Часть	Описание
адресат	Любая локальная логическая переменная, в которую может быть проведена запись, кроме регистров Dxx.
источник1	Любая локальная логическая переменная / константа, которая может быть прочитана, кроме регистров Dxx.

### Примечания

Синтаксис 1 используется для присваивания (назначения); синтаксис 2 используется в условных выражениях.

## Операторы сравнения

Операторы сравнения устанавливают относительное соответствие между двумя операндами. Выражение сравнения оценивает условные целые значения со знаком, базирующиеся на операндах.

Таблица 12-5. Операторы отношения

Оператор	Значение
<	Меньше чем
>	Больше чем
<=	Меньше чем или равно
>=	Больше чем или равно
=	Равно
<>	Не равно

Операторы сравнения могут быть использованы только в выражениях условных операторов; в каждом выражении может быть использован только один оператор сравнения.

IF источник1 Оператор Сравнения источник2 THEN

Операторы сравнения состоят из следующих частей:

Часть	Описание
источник1	Любая булева или 32-битовая локальная логическая переменная, которая может быть прочитана, или числовая константа.
источник2	Любая булева или 32-битовая локальная логическая переменная, которая может быть прочитана, или числовая константа.
Оператор Сравнения	Любой оператор отношения / логический оператор.

### Примечания

Приведенная далее таблица содержит список операторов сравнения и условия, которые определяют оценку результата – "истина" или "ложь":

Таблица 12-6. Операторы сравнения Local Logic

Операторы отношения	Оператор	"Истина", если	"Ложь", если
Меньше чем	<	источник1 < источник2	источник1 ≥ источник2
Меньше чем или равно	<=	источник1 ≤ источник2	источник1 > источник2
Больше чем	>	источник1 > источник2	источник1 ≤ источник2
Больше чем или равно	>=	источник1 ≥ источник2	источник1 < источник2
Равенство	=	источник1 = источник2	источник1 <> источник2
Неравенство	<>	источник1 <> источник2	источник1 = источник2
Нуль-оператор (IF Переменная THEN....)	нет	Переменная ненулевая	Переменная равна нулю
Побитовые логические операторы	BWAND, BWOR, BWXOR, BWNOT	Результат ненулевой	Результат нулевой

## Ошибки выполнения программы Local Logic

### Статус переполнения

Некоторые арифметические операции могут давать результаты, которые не могут быть правильно представлены в виде 32-битовых значений со знаком. Пример этого дан в приведенном ниже сегменте программы.

```
P001 := 16#7FFF_FFFF; // (1) P001 ◀ 2 147 483 647 (макс.полож.32-бит. знач.)  
P003 := P001 + 1; // (2) ! Переполнение !
```

В первой строке в P001 загружено 2 147 483 647, т.е. наибольшее значение, которое может быть представлено как 32-битовое одинарное дополнительное число со знаком. Во второй строке к этому значению прибавляется единица. Сумма, представляемая в шестнадцатеричном виде, составляет 16#8000\_0000. Однако это значение, интерпретируемое как 32-битовое одинарное дополнительное число со знаком, представляет собой отрицательное число -2 147 483 648, а не положительное 2 147 483 648! Во многих ситуациях такой результат является неожиданным и имеет нежелательные последствия в расположенных далее программных операторах.

Программа Local Logic имеет в своем распоряжении три переменные, чтобы обнаруживать переполнения. Переменная *Overflow* (*Переполнение*) является булевой переменной чтения-записи, имеющейся в распоряжении только локальной логической программы (см. гл. 13, раздел "Локальные логические системные переменные"). Когда происходит ошибка переполнения и переменная Переполнение не очищена перед окончанием цикла Local Logic, то устанавливается слово %AI кода статуса модуля DSM (локальная логическая переменная *Module\_Status\_Code*). Код ошибки указывает тип переполнения, и устанавливается %I бит Module Error Present (Имеется ошибка модуля) (локальная логическая переменная *Module\_Error\_Present*). Слово Module Status Code (Код статуса модуля) %AI и бит %I Module Error Present (Имеется ошибка модуля) не устанавливаются до тех пор, пока не закончится выполнение текущего цикла Local Logic. Переменная *Overflow* (*Переполнение*), напротив, устанавливается немедленно сразу же за операцией, вызвавшей переполнение. Программа Local Logic может очистить переменную *Переполнение*, назначив ей значение нуль. Код статуса модуля должен быть очищен со стороны ПЛК установкой бита модуля Clear Error (Сбросить ошибку) %Q.

Ошибки переполнения и расчета являются ошибками "только статус" за двумя исключениями. Деление на нуль или переполнение при делении (частное не может быть представлено 32 битами) являются ошибками *Stop Fast* (*Быстрый останов*). В случае ошибок "только статус" обработка программы Local Logic и формирование пути продолжают нормальным образом. Ошибка *stop fast* вызывает прекращение работы Local Logic до перехода к следующей операции; любое движение прекращается в результате выдачи сервоустройству команды нулевой скорости. **Примечание:** Очистка переменной Переполнение не влияет на ошибки Stop Fast.

См. таблицу 12-10 с листингом всех кодов ошибок работы Local Logic.

## Деление на ноль

Процессор Logic определяет операции деления на ноль как ошибки Fast Stop (Быстрый останов), поскольку результат операции является неопределенным. Выполнение программы Local Logic и движение сервопривода прекращаются. В код статуса модуля выдается код ошибки 16#2093 и 16#2х9А в коды ошибок каждой оси, если приводы были включены.

## Предупреждение / ошибка - окончание времени ожидания

Программы Local Logic ограничены в процессоре Logic длительностью 300 микросекунд, предоставляемых на полное выполнение программы. Это требуется для того, чтобы обеспечить достаточно времени обработки в модуле для формирования пути и для других задач. В Приложении Е дан подробный перечень времен выполнения для всех действующих операций Local Logic. Пользователь может рассчитать время исполнения данной программы, используя таблицы данных из Приложения Е. Процессор Logic выдает код ошибки предупреждения ("только статус"), если время выполнения составляет более 275 микросекунд, но менее 300. Ошибка Fast Stop (Быстрый останов) вырабатывается, если время выполнения программы превышает 300 микросекунд. См. таблицу 12-10 со списком всех кодов предупреждений и ошибок работы.

**Примечание:** Работа Local Logic прекращается, если возникают любые ошибки *Fast Stop (Быстрый останов)* Local Logic (см. табл. 12-10) и если выключены все биты цифровых выходов, управления (Толчок, Останов подачи, Сбросы строба, Включение слежения) и STL-биты, находящиеся под управлением Local Logic. Выполнение Local Logic начинается с начала, когда пользователь очищает ошибку (при помощи бита %Q сброса ошибки).

## Сообщения об ошибках Local Logic

### Сообщения об ошибках построения Local Logic

Компоновщик локальной логической программы взаимодействует с состоянием процесса построения через окно локальной логики редактора журнала ошибок. При обнаружении ошибки компоновщик сообщает об ошибке и пытается продолжить процесс.

Сообщения об ошибках, выдаваемые локальным логическим компоновщиком разделяются на три категории: синтаксические ошибки, семантические ошибки и семантические предупреждения.

Сообщения об ошибках, выдаваемые синтаксическим анализатором, имеют несколько общих элементов:

**Имя файла (Строка): [Важность] [сообщение об ошибке]**

*Имя файла* – имя текущего обрабатываемого файла.

*Строка* – номер строки в файле, где обнаружена ошибка.

*Важность* – описывает важность (серьезность) ошибки. Ошибки не допускают двойного толкования. Предупреждения являются информационными.

*Сообщение об ошибке* является коротким общим описанием ошибки.

### Синтаксические ошибки Local Logic

Компоновщик проводит проверку локального логического синтаксиса. Если исходная программа не соответствует критериям, то компоновщик сообщает о синтаксической ошибке. Сообщение об ошибке идентифицирует ошибку как синтаксическую. Сообщение об ошибке содержит обнаруженный синтаксический тип элемента, за которым следует один или несколько синтаксических элементов, которые, как полагает анализатор, должны быть здесь. Принято давать сообщение о синтаксической ошибке в строке, следующей за строкой с фактической ошибкой. Типичным примером является отсутствующая точка с запятой.

#### Пример:

```
scratch.llp (3): Error :syntax error  
actual: IF expecting: ;
```

В данном случае в строке 2 фактически отсутствует точка с запятой. Поскольку точка с запятой может оказаться на другой строке, то синтаксический анализатор не сообщает об ошибке до тех пор, пока он не обнаружит значащий синтаксический элемент, который не является точкой с запятой.

В силу своей природы одиночная синтаксическая ошибка может породить "каскадные ошибки". Исправление одной синтаксической ошибки может исключить несколько сообщений о синтаксических ошибках. Чтобы избежать путаницы при отладке программ Local Logic, имеющих синтаксические ошибки, следует исправить первую ошибку и провести новую компоновку программы, чтобы обновить список ошибок, прежде чем продолжить.

## Семантические ошибки Local Logic

Сообщения о семантических ошибках выдаются в том случае, если синтаксис правильный, но имеются семантические проблемы. Например, неправильно приписывание какое-то значение к двойной переменной за исключением случаев, когда это результат операции умножения.

### Примеры:

Ошибка (P203) Неправильное приписывание к двойной переменной:  
D00

В этом случае за сообщением об ошибке следует строка, которая идентифицирует набор символов, породивший ошибку. Далее приведен список семантических ошибок и типичные случаи.

**Таблица 12-7. Семантические ошибки Local Logic**

Номер ошибки	Описание ошибки
(P200)	Неопределенный идентификатор Программа содержит нераспознанную переменную или ключевое слово. Проверьте орфографию и синтаксис команды.
(P201)	Регистр параметров должен быть в диапазоне P000 – P255. Это сообщение об ошибке выдается, когда программа задает неопределенный регистр параметров, например P278.
(P202)	CTL-переменная должна быть в диапазоне CTL01 - CTL32 Это сообщение об ошибке выдается, когда программа задает неопределенную CTL-переменную, например CTL35.
(P203)	Неправильное назначение переменной двойной точности Программа попыталась неправильно использовать один из регистров двойной точности. Регистры двойной точности могут быть назначены только таким значениям, которые являются результатом операции умножения.
(P204)	Неправильное использование переменной двойной точности Программа попыталась неправильно использовать один из регистров двойной точности. Регистры двойной точности в выражениях могут быть использованы только в качестве делителя в операции деления или MOD.
(P205)	Присвоение переменной, предназначенной только для чтения Программа сделала попытку присвоить значение переменной, предназначенной только для чтения.
(P206)	Попытка прочитать переменную, разрешенную только для записи Программа попыталась использовать переменную, разрешенную только для записи, в одном из операндов в арифметическом или в логическом выражении или в выражении отношения.
(P207)	Переменные с индексами не поддерживаются Переменные вида Data_Table_Int[xx] не поддерживаются. Операции с таблицами данных требуют использования переменной Data_Table_Ptr.
(P208)	Имя идентификатора превышает 50 знаков Программа попыталась сослаться на переменную с длиной идентификатора, превышающей 50 знаков.
(P209)	Регистр двойной точности должен быть в диапазоне D00 - D07 Это сообщение об ошибке выдается, когда программа задает неопределенный регистр двойной точности, например D08.
(P220)	Шестнадцатеричные константы должны быть в диапазоне 16#0 - 16#FFFFFFF Программа задала шестнадцатеричную константу, которая не может быть представлена 32 битами.
(P221)	Двоичные константы должны быть в диапазоне от 0 до (2 <sup>32</sup> )-1 Программа задала двоичную константу, которая не может быть представлена 32 битами.

Номер ошибки	Описание ошибки
(P222)	Целые числа должны быть в диапазоне от -2147483648 до 2147483647 Программа задала десятичную константу, которая не может быть представлена 32 битами.
(P223)	Переполнение таблицы констант Программа может содержать не более 50 уникальных констант, которые больше 2047 или меньше -2048 (т.е. числа для своего представления требуют не менее 12 бит). Программа может содержать любое количество констант в диапазоне от -2048 до 2047.
(P230)	Превышено предельное количество уровней вложения 8 для IF. Операторы IF не могут иметь более 8 уровней вложения.
(P231)	Неправильный член в операторе IF Программа имеет арифметический оператор в операторе IF.
(P232)	Отсутствует оператор END_IF Имеется оператор IF, для которого отсутствует соответствующий оператор END_IF. Эта ошибка обнаруживается лишь в конце программы.
(P233)	Обнаружен ненадлежащий оператор END_IF Обнаружен оператор END_IF, для которого отсутствует соответствующий оператор IF.
(P240)	Назначение константе Программа попыталась использовать константу в качестве адресата оператора присвоения.
(P241)	Неправильный оператор, ожидается присвоение В том месте, где ожидался оператор присвоения (:=), был обнаружен другой оператор.
(P242)	Оператор отношения не разрешен в операторе присвоения Сделана попытка провести операцию сравнения в операторе присвоения. Присвоение, базирующееся на отношениях, разрешается выполнять при назначении булева значения в операторе IF.
(P260)	Неправильный логический оператор. Используйте BWAND, BWOR, BWXOR или BWNOT - <оператор> Программа использовала ключевые слова AND, OR, XOR или NOT, а не BWAND, BWOR, BWXOR или BWNOT, соответственно.
(P280)	Превышено граница количества операторов, макс. 150 Локальная логическая программа может иметь максимальную длину в 150 операторов. Данное сообщение об ошибке выдается, когда программа превышает указанную длину.
(P290)	Адрес находится за пределами диапазона в ссылке прямого доступа к памяти Переменная с прямым доступом к памяти задала неправильное смещение.
(P291)	Неправильная переменная с прямой адресацией памяти Была задана неправильная переменная с прямым доступом к памяти.
(P292)	Переменная с прямым доступом к памяти требует подстрочного индекса Программа сделала ссылку на переменную с прямым доступом к памяти без указания смещения.
(P293)	Превышена максимальная граница счетчика числа ошибок Анализатор Local Logic сообщает не более чем о 30 ошибках. Если эта граница превышена, то появляется данное сообщение, и сообщения о других ошибках более не выводятся.
298- (P299)	Внутренняя ошибка. Обратитесь в службу технической поддержки GE Fanuc. Если анализатор сообщает об ошибке 298 или 299 для пользовательской программы, то обратитесь за поддержкой в службу технической поддержки GE Fanuc. Предоставьте копию программы и журнал регистрации ошибок.
(P300)	Директивы семантического анализа должны предшествовать любым исполняемым операторам. Директивы #pragma (указание транслятору) должны предшествовать любым исполняемым операторам в блоке программы Local Logic.
(P301)	Неправильное задание директивы Заданная директива #pragma не опознана анализатором Local Logic.
(P302)	Неправильный параметр директивы Был задан неправильный аргумент для директивы только ошибки #pragma. Этим аргументом должно быть 1, ON, 0 или OFF.

## Семантические предупреждения Local Logic

Семантические предупреждения выводятся для условий, которые могут давать неожиданные результаты или указывать на возможные недочеты в программе Local Logic.

**Таблица 12-8. Семантические предупреждения Local Logic**

Номер ошибки	Описание ошибки
(P400)	Присваивание двоичное переменной может привести к потере данных Это сообщение выдается, когда булевой переменной назначается не булева переменная или константа или выражение, содержащее не булевы переменные.
(P410)	Проверьте длительность выполнения программы Это предупреждение появляется для программ, содержащих более 100 операторов. Поскольку максимум для количества операторов составляет 150 операторов, то можно написать такую программу Local Logic, которая потребует слишком много времени для выполнения. Время, требуемое для выполнения программ, см. в приложении А в " <i>Справочном руководстве по ЦП Series 90-30</i> ", GFK-0467.
(P481)	Устаревший синтаксис: параметр функции требует круглые скобки Параметр вызова функции ABS не заключен в круглые скобки.
(P482)	Неожиданный конец программы: незакрытый комментарий Комментарий, начавшийся двумя символами "(*", не был закрыт, когда был обнаружен конец программы.
(P483)	Вложенные комментарии Это предупреждение появляется, когда программа Local Logic задает комментирующий текст внутри другого комментария.
(P490)	Программа не содержит исполняемых операторов Программа содержит только пустое пространство и / или комментарии.

## Сообщения об ошибках загрузки программ Local Logic

При загрузке программы Local Logic могут быть выданы в код статуса модуля указанные далее ошибки.

**Таблица 12-9. Коды ошибок конфигурации Local Logic**

Код ошибки (шестнадцатиричный)	Реакция	Описание	Тип ошибки
0A	Системная ошибка	Неправильная конфигурация цифрового выхода	Модуль
0B	Системная ошибка	Неправильная конфигурация CTL-бита	Модуль

**Примечание:** См. в главе 14 подробное описание по конфигурированию CTL-битов и цифровых выходов для Local Logic

**Таблица 12-10. Коды ошибок предварительной обработки Local Logic**

Код ошибки (шестнадцатиричный)	Реакция	Описание	Тип ошибки
F0A0	Системная ошибка	Ошибка заголовка программы Local Logic	Модуль
F0A1	Системная ошибка	Ошибка в указателе конца программы Local Logic	Модуль
F0A2	Системная ошибка	Ошибка константы в заголовке программы Local Logic	Модуль
F0A3	Системная ошибка	Ошибка константы в указателе конца программы Local Logic	Модуль
F0A4	Системная ошибка	Ошибка ссылки на константу в программе Local Logic	Модуль
F0A5	Системная ошибка	Превышен предел для оттранслированной программы Local Logic	Модуль
F0A6	Системная ошибка	Ошибка несогласованного оператора IF_THEN в программе Local Logic	Модуль
F0A7	Системная ошибка	Ошибка несогласованного оператора END_IF в программе Local Logic	Модуль
F0A8	Системная ошибка	Превышена граница по количеству уровней вложения в программе Local Logic	Модуль
F0A9	Системная ошибка	Ошибка прохода программы Local Logic	Модуль
F0AA	Системная ошибка	Ошибка зарезервированного класса программы Local Logic	Модуль
F0AB	Системная ошибка	Неправильный регистр параметра программы Local Logic	Модуль
F0AC	Системная ошибка	Неправильный регистр двойной точности программы Local Logic	Модуль
F0AD	Системная ошибка	Ошибка цифрового выхода программы Local Logic	Модуль
F0AE	Системная ошибка	Ошибка CTL-бита программы Local Logic	Модуль
F0B0	Системная ошибка	Неправильный первичный оператор программы Local Logic	Модуль
F0B1	Системная ошибка	Неправильный вторичный оператор программы Local Logic	Модуль
F0B2	Системная ошибка	Неправильный вторичный источник программы Local Logic	Модуль
F0B3	Системная ошибка	Неправильный первичный источник программы Local Logic	Модуль

F0B4	Системная ошибка	Неправильный источник программы Local Logic	Модуль
F0B5	Системная ошибка	Ошибка источника "только запись" программы Local Logic	Модуль
F0B6	Системная ошибка	Ошибка прямой адресации памяти программы Local Logic	Модуль
F0B7	Системная ошибка	Неправильное назначение в программе Local Logic	Модуль
F0B8	Системная ошибка	Назначение при "только чтение" в программе Local Logic	Модуль

## Ошибки выполнения программы Local Logic

При выполнении в модуле программы Local Logic могут появиться приведенные ниже сообщения об ошибках и предупреждения.

**Таблица 12-11. Ошибки выполнения программы Local Logic**

Код ошибки (шестнадцатичный)	Реакция	Описание	Тип ошибки
91	Быстрый останов	Выдается команда системного останова в программе Local Logic	Модуль
92	Быстрый останов	Превышена граница длительности выполнения программы Local Logic	Модуль
93	Быстрый останов	Деление на ноль в Local Logic	Модуль
94	Быстрый останов	Переполнение при делении в Local Logic	Модуль
95	Только статус	Переполнение при сложении / вычитании в Local Logic	Модуль
96	Только статус	Переполнение при получении абсолютного значения (ABS) в Local Logic	Модуль
97	Только статус	Предупреждение о границе длительности выполнения программы Local Logic	Модуль
98	Только статус	Ошибка при выполнении первого цикла программы Local Logic	Модуль
99	Только статус	Неправильное имя программы в Local Logic или оно не разрешено при конфигурации аппаратного обеспечения	Модуль
9A	Быстрый останов	Ошибка останова Local Logic	Каждая ось

## Типы переменных Local Logic

Программа Local Logic получает доступ к переменным контроллера движения и к регистрам параметров, используя предварительно выделенные имена переменных. Полный список всех переменных Local Logic см. в таблицах 13-1 – 13-6.

Примеры:

```
IF Actual_Position_2 > 5000 THEN ...;
```

```
IF Strobe1_Level_2 = ON THEN ...;
```

Сохранение значений в переменных осуществляется оператором присваивания ":=".

Примеры:

```
Torque_Limit_2 := 8500; (* Устанавливает границу момента вращения на 85% *)
```

```
Position_Loop_TC_1 := 2500 / Actual_Velocity_1;
```

Переменные Local Logic разделяются на две категории: глобальные переменные и осевые переменные. Имеется четыре набора осевых переменных (Ось 1 – ось 4). Каждый набор переменных разделяется на **Управляющие переменные, переменные Состояния (Статуса)** и **Вводы/Выводы лицевой панели** (см. таблицы 13-1 – 13-6). Описание терминов, используемых в таблицах переменных, следующее:

### Атрибут переменной

Атрибут для каждой переменной Local Logic указан в таблицах 13-1 – 13-6. Переменные могут быть **Read-Only (Только для чтения)**, **Write-Only (Только для записи)** или **Read-Write (Для чтения и записи)**. Синтаксический анализатор выдает сообщение об ошибке, когда пользователь пытается выполнить запись в переменную с атрибутом "Только для чтения" или прочитать переменную с атрибутом "Только для записи".

### Размер переменной

Переменные Local Logic имеют размер от 1 бита (битовые операнды) до 64 битов (для регистров двойной точности Dxx). Размер каждой переменной Local Logic см. в таблицах 13-1 – 13-6. Попытка записать значение, превышающее данный размер переменной, ведет к усечению этого значения. Например, если результат математической операции имеет длину 32 бита и назначен переменной с длиной 16 битов, то будут сохранены только младшие 16 битов. Анализатор выдает предупреждение, если битовый операнд используется в качестве адресуемой переменной в небулевой математической операции (будет сохранен только последний значащий бит результата).

**Примечание:** Переменные команд AQ (Момент вращения, Коэффициент обратной связи по скорости, Отношение слежения, Приращение позиции и Постоянная времени контура позиционирования) могут иметь разрешенный диапазон меньше, чем размер переменной Local Logic. Модуль выдает предупреждающее сообщение об ошибке и отвергает любые неправильные значения, если программа пытается записать значение, находящееся вне заданного диапазона команды AQ. Описание разрешенного диапазона команд %AQ см. в гл. 4.

### Знак переменной

Переменные Local Logic, длина которых менее 32 битов, могут иметь или не иметь знак (кроме битовых операндов, которые всегда без знака). Все математические / логические операции в процессоре Logic являются 32-битовыми операциями со знаком (кроме 64-битовых операций со знаком деления и получения модуля). Переменные со знаком, длина которых менее 32 битов, автоматически получают знаковый разряд, расширяясь до 32 битов, когда они загружаются в процессор Logic. Переменные без знака не получают дополнительный знаковый разряд. Таким образом, процессор Logic автоматически выполняет все преобразование данных и проверку границ.

## Системные переменные Local Logic

Переменные *First\_Local\_Logic\_Sweep* (Первый цикл Local Logic), *Overflow* (Переполнение) и *System\_Halt* (Системный останов) используются только в процессоре Logic; они описываются ниже.

### Переменная *First\_Local\_Logic\_Sweep*

Переменная *First\_Local\_Logic\_Sweep* является битовым операндом **Read-Only (Только для чтения)** (см. таблицу 13-5). Она устанавливается процессором Logic в ходе первого цикла исполнения, когда включена Local Logic и ПЛК находится в режиме RUN. Переменная сбрасывается на ноль для последующих циклов. Поэтому она может быть использована в программе Local Logic, чтобы инициализировать некоторые переменные. Например, приведенная ниже программа инициализирует некоторые регистры параметров и управляющие переменные на первом цикле, используя переменную *First\_Local\_Logic\_Sweep*.

```
IF First_Local_Logic_Sweep THEN (* Если это первый цикл исполнения, то *)
    P001 := 0; (* Установка P001 на 0 *)
    P015 := 1000; (* Установка P015 на 1000 *)
    Velocity_Loop_Gain_1 := 20; (* Установка коэффициента усиления обратной
    связи по скорости для оси 1 на 20 *)
END_IF;
```

### Переменная *Overflow* (Переполнение)

Переменная *Overflow* (Переполнение) является битовым операндом **Read-Write (Чтение-запись)** (см. таблицу 13-5). Она устанавливается процессором Logic, если происходит переполнение при операции сложения, вычитания или получения абсолютного значения (ABS). Когда устанавливается флаг переполнения и возникает ошибка переполнения, в код статуса модуля выдается также код предупреждающего сообщения об ошибке (см. гл. 12). Пользователь может отказаться от выдачи предупреждений о переполнении в операциях сложения, вычитания и ABS, если установить переменную *Overflow* на ноль в конце программы Local Logic. Аналогично пользователь может протестировать ошибки переполнения в программе Local Logic путем чтения переменной *Overflow* и

осуществления некоторых соответствующих действий. Очистка переменной *Overflow* выполняется при следующих обстоятельствах:

- 1) Она автоматически очищается при запуске программы Local Logic перед первым циклом исполнения.
- 2) Она может быть очищена пользователем в программе Local Logic (установкой *Overflow := 0;* ).
- 3) Она очищается, когда пользователь переключает Q-бит сброса ошибки.

### Переменная *System\_Halt* (Системный\_останов)

Переменная *System\_Halt* (*Системный\_останов*) является битовым операндом **Write-Only (Только запись)** (см. таблицу 13-5). Если программа Local Logic записывает 1 в переменную *System\_Halt*, то движение сервоустройства и выполнение программы Local Logic прекращаются. В код статуса модуля выдается также код ошибки (см. гл. 12). Таким образом, переменная *System\_Halt* может быть использована для выявления ситуации фатальной ошибки и для восстановления после ошибки. Пример программы ниже показывает возможный сценарий, в котором использована переменная *System\_Halt*:

```
IF Overflow THEN          (* Выявление ситуации переполнения *)
    System_Halt := TRUE;  (* Остановка выполнения программы Local Logic и
движения сервоустройства *)
END_IF;
```

## 64-битовые регистры двойной точности

Local Logic предоставляет дополнительно к 255-ти 32-битовым регистрам еще 8 64-битовых регистра (D00-D07) (см. таблицу 13-5). Это требуется для того, чтобы позволить пользователю сохранить результат умножения двух 32-битовых чисел в регистре Dxx, а затем выполнить с результатом операцию деления или получения модуля. Таким образом, 64-битовые регистры могут быть использованы в следующих условиях:

1. Как адресный регистр для операции умножения.
2. Как делимое (числитель) в операции деления или получения модуля.

Анализатор выдает сообщение об ошибке, если регистр используется в других операциях. Программа примера использования регистров Dxx показана ниже:

```
D01:= P001 * 2147483647; (* Выполняется операция умножения и сохранение в
64-битовом регистре *)

P010:= D01 / 12500;     (* Разделить результат и сохранить в 32-битовом
регистре *)
```

Следует отметить, что указанный выше сценарий может привести к **Divide Overflow (Переполнение при делении)**, если результат не будет помещаться в 32-битовый регистр. Переполнение при делении вызовет остановку выполнения программы Local Logic и остановку движения сервоустройства, поскольку результат операции является неопределенным (см. гл. 12). В код статуса модуля будет также выдан код ошибки.

### Примечание

При запуске Local Logic автоматически не происходит инициализация содержания 64-битовых регистров данных (D00-D07) и 32-битовых записываемых регистров данных (P000-P255). Пользователь должен инициализировать все требуемые переменные, используя либо отдельную программу Local Logic, либо переменную *First\_Local\_Logic\_Sweep*, либо ПЛК-ladder (загрузчик).

## Таблица пользовательских данных Local Logic

Local Logic использует **кольцевой буфер на 8192 байта**, который может быть использован для сохранения и восстановления, данных программой Local Logic. Список переменных *Data\_Table* см. в таблице 13-5. Доступ к таблице данных происходит через косвенную адресацию памяти. Переменная *Data\_Table\_Ptr* ("Pointer" = "Указатель") используется для указания правильного положения байта в буфере на 8192 байта. Поэтому размер переменной *Data\_Table\_Ptr* составляет 13 бит (допустимый диапазон от 0 до 8191). Указатель автоматически увеличивается, когда происходит считывание значения из кольцевого буфера или запись в него. Величина этого приращения зависит от размера переменной, к которой осуществляется доступ. *Data\_Table\_Ptr* автоматически устанавливается на 0 при запуске Local Logic до начала первого цикла выполнения. Таким образом, переменные таблицы данных могут быть использованы для доступа к большим предварительно загруженным блокам данных в программе Local Logic. Для доступа к кольцевому буферу используются приведенные ниже переменные:

<i>Data_Table_Ptr</i>	: Указатель таблицы данных - разрешенный диапазон от 0 до 8191.
<i>Data_Table_sint</i>	: 8 битов со знаком (Указатель автоматически возрастает на 1 для чтения / записи)
<i>Data_Table_usint</i>	: 8 битов без знака (Указатель автоматически возрастает на 1 для чтения / записи)
<i>Data_Table_int</i>	: 16 битов со знаком (Указатель автоматически возрастает на 2 для чтения / записи)
<i>Data_Table_uint</i>	: 16 битов без знака (Указатель автоматически возрастает на 2 для чтения / записи)
<i>Data_Table_dint</i>	: 32 бита (Указатель автоматически возрастает на 4 для чтения / записи)

Пример программы далее показывает, как можно получить доступ к определенному участку памяти в круговом буфере:

```
Data_Table_Ptr := 100; (* Указывает на смещение 100 байта в буфере *)
P001 := Data_Table_int; (* Считывает 16-битовое число со знаком из буфера *)
(* Data_Table_Ptr автоматически возрастает до 102 *)
Data_Table_sint := -120; (* Выполняется запись 8-битового числа со знаком в байт 102 *)
Data_Table_Ptr := 0; (* Указывает на смещение 0 байта в буфере *)
```

## Цифровые выходы / STL-переменные

Восемь (8) цифровых выходов модуля (по 2 на ось) конфигурируются индивидуально так, чтобы находиться под управлением либо ПЛК (по умолчанию), либо Local Logic (DSM). Если программа Local Logic выполняет запись в определенную переменную *Digital\_Output* (*Цифровой выход*) (см. табл. 13-1 – 13-6), то последняя должна быть сконфигурирована под управление DSM. Модуль DSM будет отвергать любые программы Local Logic, которые загружены с неправильной конфигурацией цифрового выхода. Подробное описание конфигурации цифровых выходов см. в гл. 14.

STL01-STL24 также конфигурируются индивидуально, чтобы иметь различные источники входа. Детальное описание опций конфигурации см. в главе 14. STL25 - STL32 не конфигурируются и всегда находятся под управлением Local Logic. Модуль DSM будет отвергать любые программы Local Logic, которые загружены с неправильной конфигурацией STL. Например, если программа Local Logic имеет оператор, который записывает в STL16 (т.е. *CTL16 := I;*), то STL16 должен быть сконфигурирован как "Управляемый от Local Logic" в конфигурации аппаратного обеспечения VersaPro. Для синхронизации программы движения и программы Local Logic могут быть использованы, биты STL01 - STL32 и номера блоков программы движения (переменные *Block\_1*, *Block\_2*, *Block\_3*, *Block\_4*).

Таблица 13-1. Переменные оси 1

Имя переменной Local Logic	Атрибут	Размер
<b>В/В лицевой панели</b>		
Strobe1_Level_1	Только чтение	Битовый операнд
Strobe2_Level_1	Только чтение	Битовый операнд
Positive_EOT_1	Только чтение	Битовый операнд
Negative_EOT_1	Только чтение	Битовый операнд
Home_Switch_1	Только чтение	Битовый операнд
Digital_Output1_1 <sup>(1)</sup>	Только запись	Битовый операнд
Digital_Output3_1 <sup>(1)</sup>	Только запись	Битовый операнд
Analog_Input1_1	Только чтение	16 битов со знаком
Analog_Input2_1	Только чтение	16 битов со знаком
<b>Управляющие переменные</b>		
Velocity_Loop_Gain_1	Чтение / Запись	8 битов без знака
Position_Loop_TC_1	Только запись	16 битов без знака
Torque_Limit_1	Только запись	16 битов без знака
Follower_Ratio_A_1	Только запись	16 битов со знаком
Follower_Ratio_B_1	Только запись	16 битов со знаком
Position_Increment_Cts_1 <sup>(2)</sup>	Только запись	16 битов со знаком
Reset_Strobe1_1	Только запись	Битовый операнд
Reset_Strobe2_1	Только запись	Битовый операнд
Enable_Follower_1	Только запись	Битовый операнд
Jog_Plus_1	Только запись	Битовый операнд
Jog_Minus_1	Только запись	Битовый операнд
FeedHold_1	Только запись	Битовый операнд
<b>Переменные статуса</b>		
Error_Code_1	Только чтение	16 битов без знака
Block_1	Только чтение	16 битов без знака
Actual_Position_1	Только чтение	32 бита
Commanded_Position_1	Только чтение	32 бита
Position_Error_1	Только чтение	32 бита
Strobe1_Position_1	Только чтение	32 бита
Strobe2_Position_1	Только чтение	32 бита
Actual_Velocity_1	Только чтение	32 бита
Commanded_Velocity_1	Только чтение	32 бита
Commanded_Torque_1	Только чтение	32 бита
User_Selected_Data1_1	Только чтение	32 бита
User_Selected_Data2_1	Только чтение	32 бита
UnAdjusted_Actual_Position_Cts_1	Только чтение	32 бита
UnAdjusted_Strobe1_Position_Cts_1	Только чтение	32 бита
UnAdjusted_Strobe2_Position_Cts_1	Только чтение	32 бита
Axis_OK_1	Только чтение	Битовый операнд
Position_Valid_1	Только чтение	Битовый операнд
Strobe1_Flag_1	Только чтение	Битовый операнд
Strobe2_Flag_1	Только чтение	Битовый операнд
Drive_Enabled_1	Только чтение	Битовый операнд
Program_Active_1	Только чтение	Битовый операнд
Moving_1	Только чтение	Битовый операнд
In_Zone_1	Только чтение	Битовый операнд
Position_Error_Limit_1	Только чтение	Битовый операнд
Torque_Limited_1	Только чтение	Битовый операнд
Servo_Ready_1	Только чтение	Битовый операнд
Follower_Enabled_1	Только чтение	Битовый операнд
Follower_Velocity_Limit_1	Только чтение	Битовый операнд
Follower_Ramp_Active_1	Только чтение	Битовый операнд

**Примечания:**

- (1) Эти цифровые выходы должны быть сконфигурированы для управления Local Logic при конфигурации аппаратного обеспечения VersaPro, чтобы Local Logic могла осуществлять запись.
- (2) Переменная Position\_Increment\_Cnts\_n имеет максимальный диапазон  $\pm 1023$  импульса.

Таблица 13-2. Переменные оси 2

Имя переменной Local Logic	Атрибут	Размер
В/В лицевой панели		
Strobe1_Level_2	Только чтение	Битовый операнд
Strobe2_Level_2	Только чтение	Битовый операнд
Positive_EOT_2	Только чтение	Битовый операнд
Negative_EOT_2	Только чтение	Битовый операнд
Home_Switch_2	Только чтение	Битовый операнд
Digital_Output1_2 <sup>(1)</sup>	Только запись	Битовый операнд
Digital_Output3_2 <sup>(1)</sup>	Только запись	Битовый операнд
Analog_Input1_2	Только чтение	16 битов со знаком
Analog_Input2_2	Только чтение	16 битов со знаком
Управляющие переменные		
Velocity_Loop_Gain_2	Чтение / Запись	8 битов без знака
Position_Loop_TC_2	Только запись	16 битов без знака
Torque_Limit_2	Только запись	16 битов без знака
Follower_Ratio_A_2	Только запись	16 битов со знаком
Follower_Ratio_B_2	Только запись	16 битов со знаком
Position_Increment_Cts_2 <sup>(2)</sup>	Только запись	16 битов со знаком
Reset_Strobe1_2	Только запись	Битовый операнд
Reset_Strobe2_2	Только запись	Битовый операнд
Enable_Follower_2	Только запись	Битовый операнд
Jog_Plus_2	Только запись	Битовый операнд
Jog_Minus_2	Только запись	Битовый операнд
FeedHold_2	Только запись	Битовый операнд
Переменные статуса		
Error_Code_2	Только чтение	16 битов без знака
Block_2	Только чтение	16 битов без знака
Actual_Position_2	Только чтение	32 бита
Commanded_Position_2	Только чтение	32 бита
Position_Error_2	Только чтение	32 бита
Strobe1_Position_2	Только чтение	32 бита
Strobe2_Position_2	Только чтение	32 бита
Actual_Velocity_2	Только чтение	32 бита
Commanded_Velocity_2	Только чтение	32 бита
Commanded_Torque_2	Только чтение	32 бита
User_Selected_Data1_2	Только чтение	32 бита
User_Selected_Data2_2	Только чтение	32 бита
UnAdjusted_Actual_Position_Cts_2	Только чтение	32 бита
UnAdjusted_Strobe1_Position_Cts_2	Только чтение	32 бита
UnAdjusted_Strobe2_Position_Cts_2	Только чтение	32 бита
Axis_OK_2	Только чтение	Битовый операнд
Position_Valid_2	Только чтение	Битовый операнд
Strobe1_Flag_2	Только чтение	Битовый операнд
Strobe2_Flag_2	Только чтение	Битовый операнд

Имя переменной Local Logic	Атрибут	Размер
Drive_Enabled_2	Только чтение	Битовый операнд
Program_Active_2	Только чтение	Битовый операнд
Moving_2	Только чтение	Битовый операнд
In_Zone_2	Только чтение	Битовый операнд
Position_Error_Limit_2	Только чтение	Битовый операнд
Torque_Limited_2	Только чтение	Битовый операнд
Servo_Ready_2	Только чтение	Битовый операнд
Follower_Enabled_2	Только чтение	Битовый операнд
Follower_Velocity_Limit_2	Только чтение	Битовый операнд
Follower_Ramp_Active_2	Только чтение	Битовый операнд

**Примечания:**

- (1) Эти цифровые выходы должны быть сконфигурированы для управления Local Logic при конфигурации аппаратного обеспечения VersaPro, чтобы Local Logic могла осуществлять запись.
- (2) Переменная Position\_Increment\_Cnts\_n имеет максимальный диапазон  $\pm 1023$  импульса.

**Таблица 13-3. Переменные оси 3**

Имя переменной Local Logic	Атрибут	Размер
<b>В/В лицевой панели</b>		
Strobe1_Level_3	Только чтение	Битовый операнд
Strobe2_Level_3	Только чтение	Битовый операнд
Positive_EOT_3	Только чтение	Битовый операнд
Negative_EOT_3	Только чтение	Битовый операнд
Home_Switch_3	Только чтение	Битовый операнд
Digital_Output1_3 *	Только запись	Битовый операнд
Digital_Output3_3 *	Только запись	Битовый операнд
Analog_Input1_3	Только чтение	16 битов со знаком
Analog_Input2_3	Только чтение	16 битов со знаком
<b>Управляющие переменные</b>		
Reset_Strobe1_3	Только запись	Битовый операнд
Reset_Strobe2_3	Только запись	Битовый операнд
<b>Переменные статуса</b>		
Error_Code_3	Только чтение	16 битов без знака
Actual_Position_3	Только чтение	32 бита
Strobe1_Position_3	Только чтение	32 бита
Strobe2_Position_3	Только чтение	32 бита
Actual_Velocity_3	Только чтение	32 бита
Axis_OK_3	Только чтение	Битовый операнд
Position_Valid_3	Только чтение	Битовый операнд
Strobe1_Flag_3	Только чтение	Битовый операнд
Strobe2_Flag_3	Только чтение	Битовый операнд

\* **Примечание:** Эти цифровые выходы должны быть сконфигурированы для управления Local Logic при конфигурации аппаратного обеспечения VersaPro, чтобы Local Logic могла осуществлять запись.

**Примечание**

Модуль DSM314, версия 2.0, поддерживает для оси 3 только переменные из таблицы 13-3.

Таблица 13-4. Переменные оси 4

Имя переменной Local Logic	Атрибут	Размер
	<b>В/В лицевой панели</b>	
Strobe1_Level_4	Только чтение	Битовый операнд
Strobe2_Level_4	Только чтение	Битовый операнд
Positive_EOT_4	Только чтение	Битовый операнд
Negative_EOT_4	Только чтение	Битовый операнд
Home_Switch_4	Только чтение	Битовый операнд
Digital_Output1_4 *	Только запись	Битовый операнд
Digital_Output3_4 *	Только запись	Битовый операнд
Analog_Input1_4	Только чтение	16 битов со знаком
Analog_Input2_4	Только чтение	16 битов со знаком

\* **Примечание:** Эти цифровые выходы должны быть сконфигурированы для управления Local Logic при конфигурации аппаратного обеспечения VersaPro, чтобы Local Logic могла осуществлять запись.

### Примечание

Модуль DSM314, версия 2.0, поддерживает для оси 4 только переменные из таблицы 13-4.

Таблица 13-5. Глобальные переменные

Имя переменной Local Logic	Атрибут	Размер
Overflow <sup>(1)</sup>	Чтение / Запись	Битовый операнд
System_Halt <sup>(1)</sup>	Только запись	Битовый операнд
Data_Table_Ptr <sup>(2)</sup>	Чтение / Запись	13 битов
Data_Table_sint <sup>(2)</sup>	Чтение / Запись	8 битов со знаком
Data_Table_usint <sup>(2)</sup>	Чтение / Запись	8 битов без знака
Data_Table_int <sup>(2)</sup>	Чтение / Запись	16 битов со знаком
Data_Table_uint <sup>(2)</sup>	Чтение / Запись	16 битов без знака
Data_Table_dint <sup>(2)</sup>	Чтение / Запись	32 бита
Module_Error_Present	Только чтение	Битовый операнд
New_Configuration_Received	Только чтение	Битовый операнд
First_Local_Logic_Sweep <sup>(1)</sup>	Только чтение	Битовый операнд
Module_Status_Code	Только чтение	16 битов без знака
CTL_1_to_32 <sup>(3)</sup>	Только чтение	32 бита
P000-P255	Чтение / Запись	32 бита
D00-D07 <sup>(4)</sup>	Чтение / Запись	64 бита

### Примечания:

- (1) См. раздел в "Системные переменные Local Logic".
- (2) См. раздел в "Таблица пользовательских данных Local Logic".
- (3) Переменная *CTL\_1\_to\_32* может быть использована для считывания всех 32 CTL-битов в регистр.
- (4) См. раздел в "64-битовые регистры с двойной точностью".

Таблица 13-6. STL-биты

Имя переменной Local Logic	Атрибут	Размер
CTL01 **	Чтение / Запись	Битовый операнд
CTL02 **	Чтение / Запись	Битовый операнд
CTL03 **	Чтение / Запись	Битовый операнд
CTL04 **	Чтение / Запись	Битовый операнд
CTL05 **	Чтение / Запись	Битовый операнд
CTL06 **	Чтение / Запись	Битовый операнд
CTL07 **	Чтение / Запись	Битовый операнд
CTL08 **	Чтение / Запись	Битовый операнд
CTL09 **	Чтение / Запись	Битовый операнд
CTL10 **	Чтение / Запись	Битовый операнд
CTL11 **	Чтение / Запись	Битовый операнд
CTL12 **	Чтение / Запись	Битовый операнд
CTL13 **	Чтение / Запись	Битовый операнд
CTL14 **	Чтение / Запись	Битовый операнд
CTL15 **	Чтение / Запись	Битовый операнд
CTL16 **	Чтение / Запись	Битовый операнд
CTL17 **	Чтение / Запись	Битовый операнд
CTL18 **	Чтение / Запись	Битовый операнд
CTL19 **	Чтение / Запись	Битовый операнд
CTL20 **	Чтение / Запись	Битовый операнд
CTL21 **	Чтение / Запись	Битовый операнд
CTL22 **	Чтение / Запись	Битовый операнд
CTL23 **	Чтение / Запись	Битовый операнд
CTL24 **	Чтение / Запись	Битовый операнд
CTL25	Чтение / Запись	Битовый операнд
CTL26	Чтение / Запись	Битовый операнд
CTL27	Чтение / Запись	Битовый операнд
CTL28	Чтение / Запись	Битовый операнд
CTL29	Чтение / Запись	Битовый операнд
CTL30	Чтение / Запись	Битовый операнд
CTL31	Чтение / Запись	Битовый операнд
CTL32	Чтение / Запись	Битовый операнд

**\*\* Примечание:** STL-биты от 1 до 24 конфигурируются индивидуально при конфигурации аппаратного обеспечения VersaPro (см. гл. 14 "Конфигурация Local Logic"). Local Logic может выполнять запись в CTL01-24 только в том случае, если биты были сконфигурированы при конфигурации аппаратного обеспечения как "Управляемые от Local Logic".



## Конфигурация CTL-битов

Программная среда VersaPro позволяет пользователю конфигурировать входной источник для CTL-битов (CTL01 - CTL24), используя окно конфигурации аппаратного обеспечения. В окне конфигурации аппаратного обеспечения следует выбрать модуль DSM314, который надо сконфигурировать. Правильная последовательность выбора меню и кнопок для выхода на конфигурацию аппаратного обеспечения дана в главе 10. Окна конфигурации DSM314 содержат закладку с названием "CTL Bits" ("CTL-биты"). Выбор этой закладки выводит окно, похожее на изображенное на рис. 14-1.

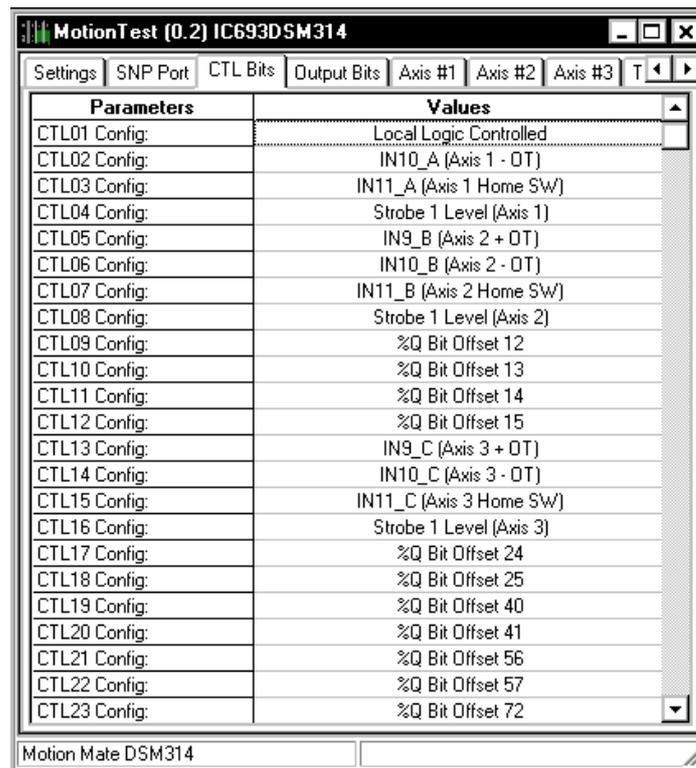


Рис. 14-1. Диалоговое окно конфигурации CTL-битов

Окно конфигурации позволяет пользователю выбрать такую конфигурацию CTL-битов, которая соответствует данной программе движения и программе Local Logic. Приведенный далее раздел дает дополнительную информацию по процессу конфигурации CTL-битов.

## Новые CTL-биты CTL01-CTL32

- CTL01 - CTL24 являются конфигурируемыми CTL-битами.
- CTL25-CTL32 являются неконфигурируемыми CTL-битами, обеспечивающими чтение Local Logic и запись Local Logic.

**Таблица 14-1. Общее представление CTL-битов**

Общее представление битов CTL01-CTL32 для DSM314							
Идентификатор	Бит %I	Входы лицевой панели	Бит %Q	Чтение Local Logic	Запись Local Logic	Запись FBSA <sup>1</sup>	Чтение FBSA <sup>1</sup>
CTL01-CTL08	X	Конфиг.	Конфиг.	X	Конфиг.	Конфиг.	X
CTL09-CTL12		Конфиг.	Конфиг.	X	Конфиг.	Конфиг.	
CTL13-CTL16	X	Конфиг.	Конфиг.	X	Конфиг.	Конфиг.	
CTL17-CTL24		Конфиг.	Конфиг.	X	Конфиг.	Конфиг.	
CTL25-CTL32				X	X		

<sup>1</sup>FBSA обозначает "Быстрый доступ к статусу системной шины". См. подробности в GFK-0467L (или в более позднем издании).

Рисунок далее показывает источники, которые выполняют запись в CTL-биты и адресаты, которые считывают CTL-биты:

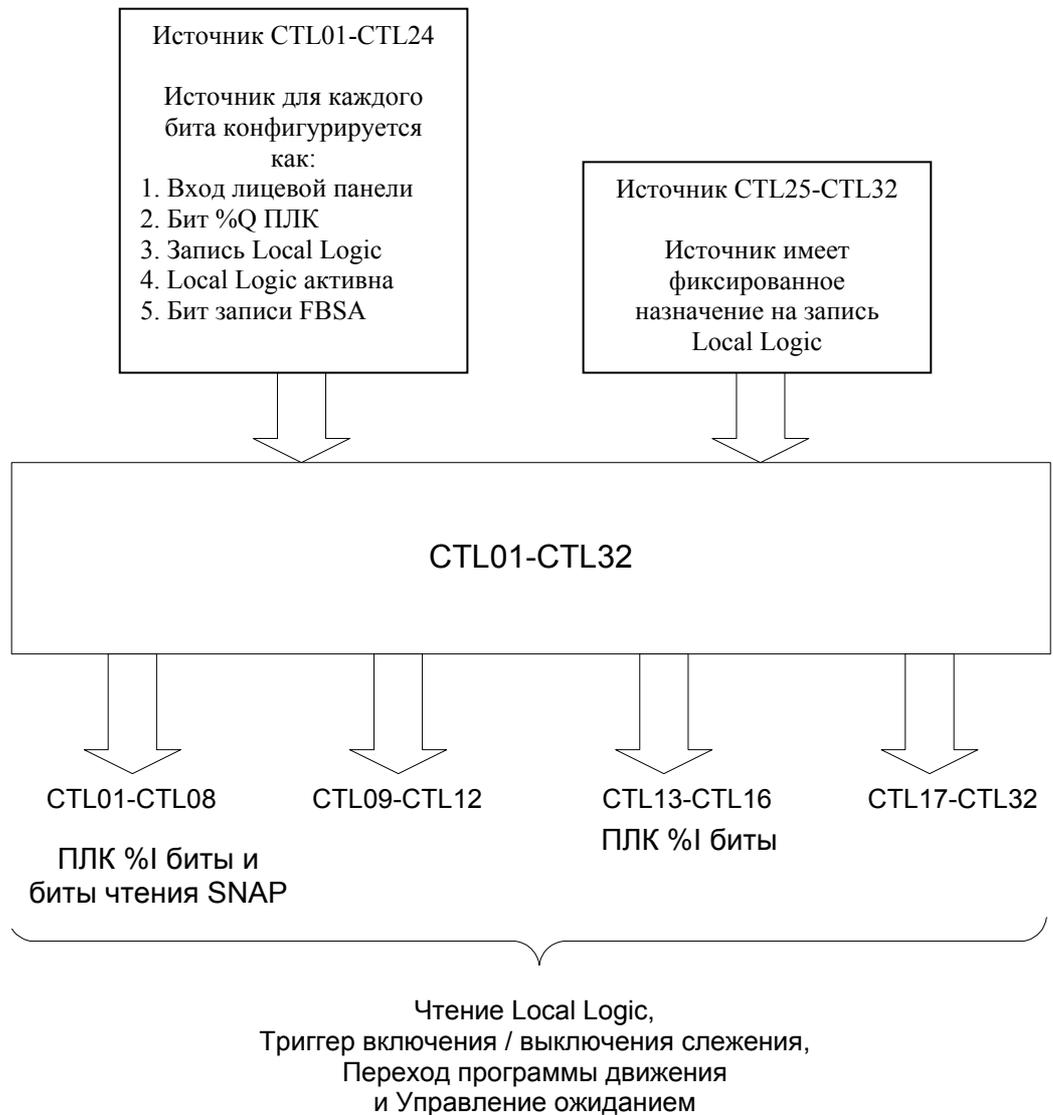


Рис. 14-2. Источники / адреса для CTL-битов

## Варианты выбора конфигурации битов CTL01-CTL24

Каждый из битов CTL01-CTL24 является индивидуально конфигурируемым. CTL17-CTL22 относятся по умолчанию к управляющим битам цифровых выходов %Q для осей 1 – 3. CTL23-CTL24 относятся по умолчанию к битам записи 1-2 Быстрого доступа к статусу системной шины (FBSA). Варианты конфигурации показаны в таблице ниже.

**Таблица 14-2. Варианты конфигурации CTL-битов**

CTL-биты	Допустимые значения конфигурации для источника бита	Описание
CTL01-CTL24	IN9_A	Перерегулирование (+) оси 1
	IN10_A	Перерегулирование (-) оси 1
	IN11_A	Переключатель начальной позиции оси 1
	IN9_B	Перерегулирование (+) оси 2
	IN10_B	Перерегулирование (-) оси 2
	IN11_B	Переключатель начальной позиции оси 2
	IN9_C	Лицевая панель, вход 24 В, ось 3
	IN10_C	Лицевая панель, вход 24 В, ось 3
	IN11_C	Переключатель начальной позиции оси 3
	IN9_D	Лицевая панель, вход 24 В, ось 4
	IN10_D	Лицевая панель, вход 24 В, ось 4
	IN11_D	Лицевая панель, вход 24 В, ось 4
	Strobe1 Level Axis1	Уровень входного строба 1, ось 1
	Strobe2 Level Axis1	Уровень входного строба 2, ось 1
	Strobe1 Level Axis2	Уровень входного строба 1, ось 2
	Strobe2 Level Axis2	Уровень входного строба 2, ось 2
	Strobe1 Level Axis3	Уровень входного строба 1, ось 3
	Strobe2 Level Axis3	Уровень входного строба 2, ось 3
	IN5_D	Лицевая панель, вход 5 В, ось 4
	IN6_D	Лицевая панель, вход 5 В, ось 4
	Local Logic Write	CTL-бит под управлением Local Logic
	Local Logic Active Flag	Действует программа Local Logic
	FBSA Write Bit 1	Последовательный неподтвержденный протокол (FBSA), бит 1
	FBSA Write Bit 2	Последовательный неподтвержденный протокол (FBSA), бит 2
	FBSA Write Bit 3	Последовательный неподтвержденный протокол (FBSA), бит 3
	FBSA Write Bit 4	Последовательный неподтвержденный протокол (FBSA), бит 4
	%Q bit Offset 12	CTL09, управление программы
	%Q bit Offset 13	CTL10, управление программы
	%Q bit Offset 14	CTL11, управление программы
	%Q bit Offset 15	CTL12, управление программы
	%Q bit Offset 24	Лиц. панель, 24 В, выход, упр-ние, ось 1 (OUT1_A)
	%Q bit Offset 25	Лиц. панель, 5 В, выход, упр-ние, ось 1 (OUT3_A)
	%Q bit Offset 40	Лиц. панель, 24 В, выход, упр-ние, ось 2 (OUT1_B)
%Q bit Offset 41	Лиц. панель, 5 В, выход, упр-ние, ось 2 (OUT3_B)	
%Q bit Offset 56	Лиц. панель, 24 В, выход, упр-ние, ось 3 (OUT1_C)	
%Q bit Offset 57	Лиц. панель, 5 В, выход, упр-ние, ось 3 (OUT3_C)	

## Функция FBSA и назначения CTL-битов

Функция системной шины Fast Backplane Status Access (Быстрый доступ к статусу системной шины) (FBSA) записывает 4 бита в DSM и считывает 8 битов. Расположение функции FBSA показано в приведенной далее таблице. Информацию по обслуживающим запросам FBSA см. в GFK-0467L (или в более позднем издании) в "Справочном руководстве по системе команд ЦП ПЛК Series 90-30/20/Micro".

Таблица 14-3. Назначения бита FBSA CTL-битам

Назначения бита FBSA		
FBSA чтение	CTL01-CTL08	Биты CTL01 - CTL08 будут иметь индивидуально конфигурируемый источник, который содержит Local Logic или любой вход лицевой панели DSM. Эти биты всегда считываются как ПЛК-биты %I и входы FBSA.
FBSA запись	CTL01-CTL24 (конфигурируемые)	Биты записи FBSA 1-4 могут быть сконфигурированы как источники для любого из битов CTL01-CTL24. Биты записи FBSA 1-2 являются источниками по умолчанию для CTL23-24.

## Конфигурация битов выходов лицевой панели

Среда программирования VersaPro при помощи конфигурации аппаратного обеспечения позволяет пользователю сконфигурировать цифровые выходы лицевой панели DSM314 либо для управления программы Local Logic, либо для управления программы ПЛК. Чтобы получить доступ к окну конфигурации битов выхода, вызовите конфигурацию аппаратного обеспечения из VersaPro. В окне конфигурации аппаратного обеспечения следует выбрать модуль DSM314, который надо сконфигурировать. Правильную последовательность меню и кнопок для выхода в конфигурацию аппаратного обеспечения см. в GFK-1670 или в интерактивной помощи по VersaPro 1.1. Окна конфигурации DSM314 содержат закладку с названием "Output Bits" ("Биты выхода"). Выбор этой закладки выводит окно, похожее на изображенное, на рис. 14-2.

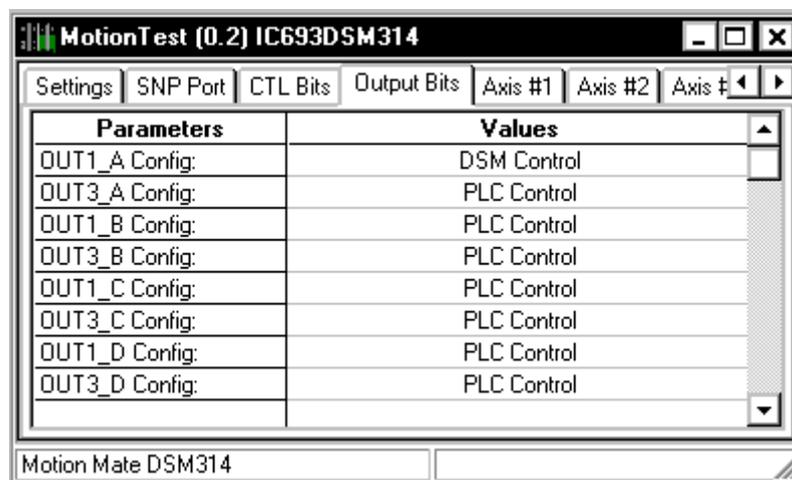


Рис. 14-3 Конфигурация битов выхода

Приведенная далее таблица описывает выходы лицевой панели, управление которыми может осуществляться либо от Local Logic, либо от ПЛК.

**Таблица 14-4. Описание битов выхода лицевой панели**

<b>Обозначение сигнала</b>	<b>Описание</b>
OUT1_A	Лицевая панель, 24 В (твердотельное реле), выход, ось 1
OUT3_A	Лицевая панель, 5 В, выход, ось 1
OUT1_B	Лицевая панель, 24 В (твердотельное реле), выход, ось 2
OUT3_B	Лицевая панель, 5 В, выход, ось 2
OUT1_C	Лицевая панель, 24 В (твердотельное реле), выход, ось 3
OUT3_C	Лицевая панель, 5 В, выход, ось 3
OUT1_D	Лицевая панель, 24 В (твердотельное реле), выход, ось 4
OUT3_D	Лицевая панель, 5 В, выход, ось 4

Конфигурация, показанная в данной главе, является специфической для программного обеспечения VersaPro. Пользователи SIMPLICITY Machine Edition должны использовать интерактивную помощь для получения детальных инструкций по конфигурации контроллера DSM314.

## Начало использования

**Примечание:** Для использования с модулем DSM314 требуется VersaPro, версия 1.1 или более поздняя. В данном документе обсуждается, как следует использовать программное обеспечение VersaPro, чтобы выйти на окна конфигурации DSM314, программирования движения и программирования Local Logic. Здесь не идет речь о том, какие значения должны быть сконфигурированы или какие команды следует использовать в программах движения или Local Logic. Такая информация содержится в других местах данного руководства. Дополнительная информация по VersaPro может быть получена в "Руководстве пользователя по пакету программ VersaPro", GFK-1670, а также в рамках интерактивной помощи по VersaPro.

## Начало работы с VersaPro

Для запуска программы щелкните дважды по значку VersaPro на панели Windows. VersaPro запускается с пустым экраном, называемым "Workbench" ("Рабочее место").

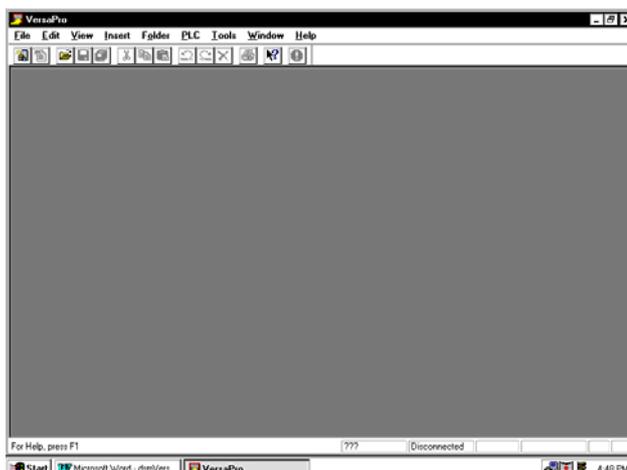


Рис. 15-1. Окно "Рабочее место" начала работы с пакетом VersaPro

- Прежде чем создавать файл, проверьте установки рабочего места, действующие по умолчанию, чтобы убедиться, что Series 90-30 является ПЛК по умолчанию. Для этого щелкните пункт Tools (Инструменты) в меню (см. рис. 15-4), затем щелкните пункт Options (Опции). Появится диалоговое окно Options, как показано ниже:

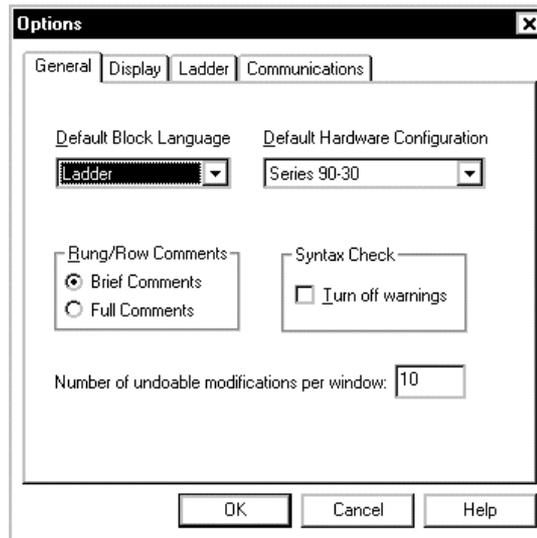


Рис. 15-2. Проверка установок по умолчанию VersaPro в диалоговом окне Tools/Options.

- Убедитесь, что Series 90-30 установлен в окне Default Hardware Configuration (Конфигурация аппаратного обеспечения по умолчанию), затем щелкните кнопку ОК.
- Чтобы открыть папку, щелкните команду File (Файл) в строке меню, затем либо щелкните Open Folder (Открыть папку), чтобы открыть существующую, либо щелкните New Folder (Новая папка), чтобы создать новую. Можно также импортировать существующую папку Series 90-30, созданную первоначально в Logicmaster или в Control. Подробности см. в разделе "Операции с папками" в главе 2 "Руководства пользователя по VersaPro". В данном примере щелкните New Folder (Новая папка). Появится диалоговое окно New Folder Wizard (Мастер создания новой папки), как показано на рисунке внизу.

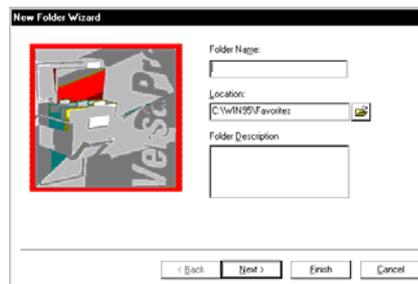


Рис. 15-3. Диалоговое окно New Folder Wizard (Мастер создания новой папки)

- Введите имя вашей папки. Хотя для имени папки может быть использовано до 255 символов, только последние 7 составляют имя папки в ПЛК. Последние 7 символов имени папки называются мнемоническим именем. Представляется правильным так выбрать имя папки, чтобы его мнемоническое имя было содержательным. Например, если папка называется "Pumphouse\_Number\_1", то сокращение, которое сохранится в ПЛК, будет "umber\_1"; оно не передает желаемый смысл. Более подходящим именем могло бы быть что-то вроде "PHouse1" или "PH1". Глава 2 "Руководства пользователя по VersaPro" (GFK-1670) имеет раздел, который поясняет правила создания имен папок, включая разрешенные символы.
- Можно также изменить место расположения папки, если требуется хранить ее не в том директории, который показан по умолчанию в поле Location (Расположение). Кроме того, имеется поле Description (Описание), которое позволяет ввести описательную информацию размером до 64 символов. После окончания ввода информации в данном диалоговом окне, щелкните кнопку Finish (Готово). (Если

требуется импортировать папку Logicmaster или Control, то щелкните Next (Далее); появится следующее диалоговое окно с возможностями импорта). Теперь появится главное окно LD (Ladder Diagram = Ladder-схема).

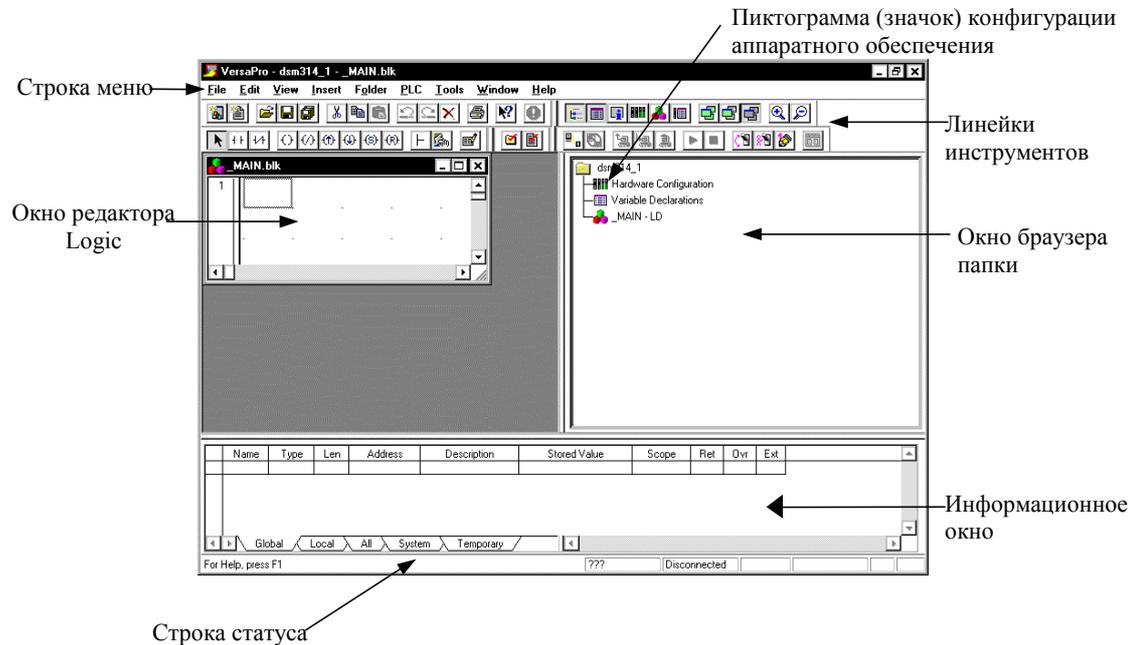


Рис. 15-4. Главное окно Ladder-схемы (LD) VersaPro

## Изменение структуры окна VersaPro

Если структура окна чем-то не устраивает, имеется возможность изменить ее. Например, некоторые пользователи предпочитают, чтобы окно браузера папки располагалось слева или чтобы окно редактора Local Logic было больше, чем задано по умолчанию. Имеются два главных способа изменения структуры окна: перемещение и изменение размеров.

**Перемещение окна:** Щелкните непосредственно у внутренней границы окна левой кнопкой мыши. В окне появится прямоугольный контур. Удерживая кнопку мыши, переместите контур в требуемое положение и отпустите кнопку мыши. Иногда при движении окна его очертания меняются. Если требуется, размер можно изменить, как описано далее.

**Изменение размеров окна:** Поместите указатель мыши на соответствующую границу окна, положение которой надо изменить. Указатель мыши изменится на короткую двойную линию со стрелками:



Нажмите левую клавишу мыши, и, удерживая ее, переместите границу в требуемое положение.

## Начало процесса конфигурации

Конфигуратор является фактически отдельной программой, которую можно запускать из главного окна (показано на предыдущем рисунке). Чтобы начать, щелкните дважды значок конфигурации аппаратного обеспечения; это приведет к запуску программы HWC (Hardware Configuration = Конфигурация аппаратного обеспечения).



Окно HWC может появиться внутри или в верхней части окна рабочего места VersaPro, как показано ниже. При этом можно щелкнуть кнопку Expand (Развернуть), чтобы раскрыть окно полностью. (Можно также щелкнуть кнопку разворачивания в малом окне, чтобы развернуть его тоже).

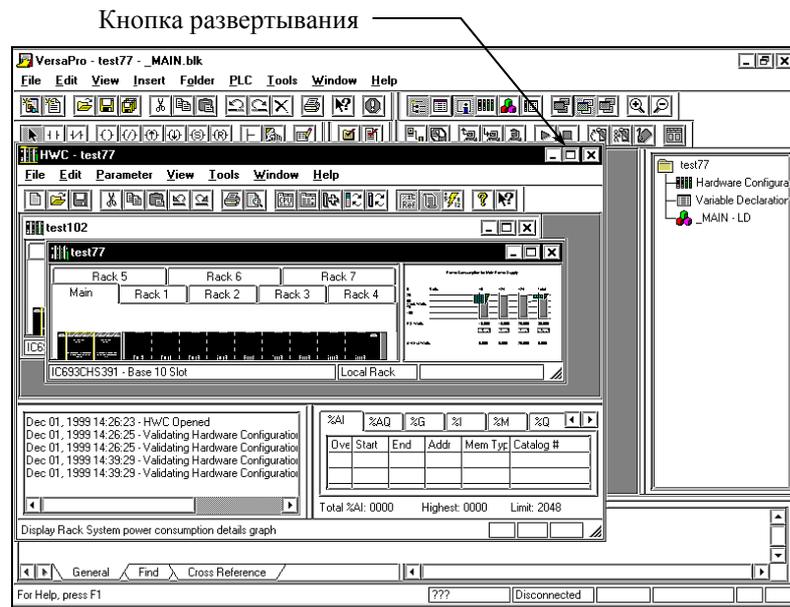


Рис. 15-5. Окно запуска конфигурации аппаратного обеспечения (HWC)

Окно конфигурации развернется в полный размер:

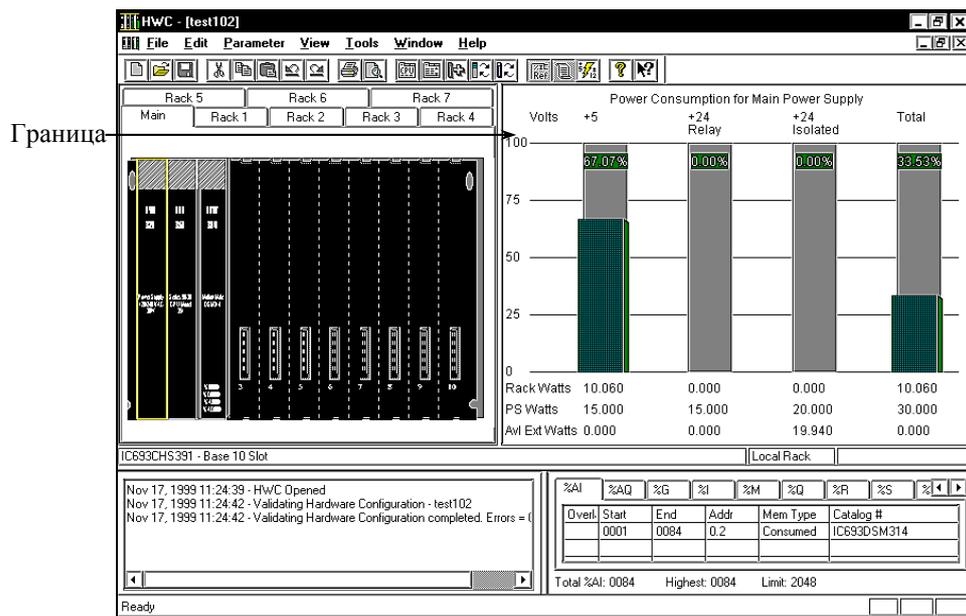


Рис. 15-6. Развернутое окно конфигурации аппаратного обеспечения

- Если не удастся прочитать номера модулей, то можно увеличить левое окно, перемещая его границу (показано на предыдущем рисунке) вправо. После этого экран будет выглядеть следующим образом:

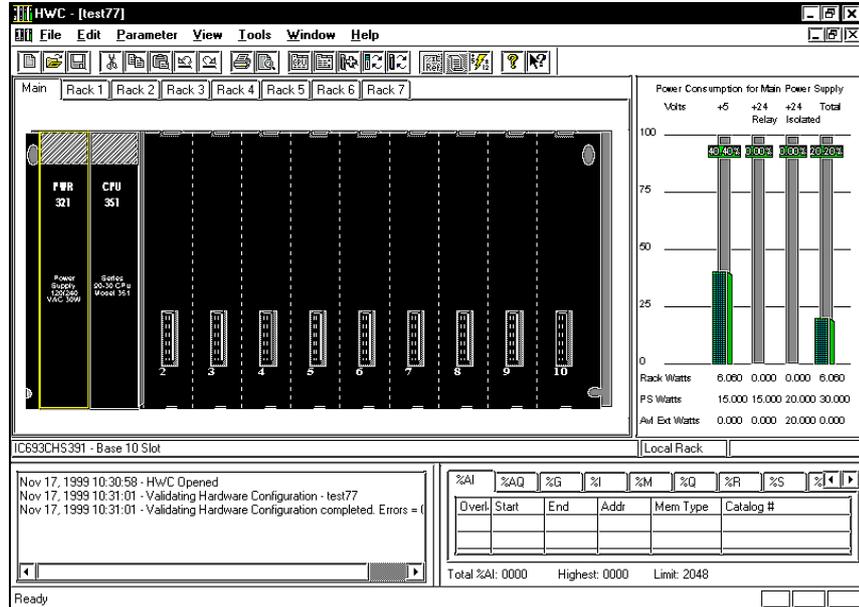


Рис. 15-7. Увеличенный экран конфигурации аппаратного обеспечения

Теперь все готово для начала самого процесса конфигурации, описанного в следующем разделе "Конфигурация DSM314".

## Конфигурация DSM314

В данном разделе обсуждается конфигурация модуля DSM314. Для конфигурации прочего аппаратного обеспечения обращайтесь к "Руководству пользователя VersaPro", GFK-1670, и к интерактивной помощи по VersaPro.

- При открытом окне конфигурации, как показано на предыдущем рисунке, щелкните дважды по пустому слоту, где должен быть установлен DSM314. Появится окно Module Catalog (Каталог модулей) со списком категорий модулей:

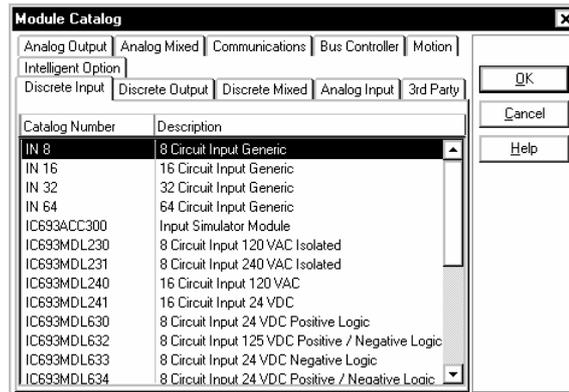


Рис. 15-8. Окно Module Catalog (Каталог модулей) для конфигурации аппаратного обеспечения

- Щелкните закладку Motion (Движение), чтобы выйти на список модулей движения:

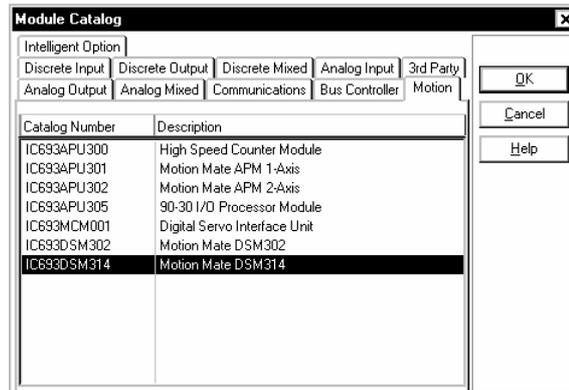


Рис. 15-9. Закладка Motion (Движение) для конфигурации аппаратного обеспечения

- Дважды щелкните IC693DSM314 или выделите его, как показано на рисунке, и щелкните кнопку ОК. Модуль DSM314 будет добавлен в экранный крейт, и появится его окно конфигурации:

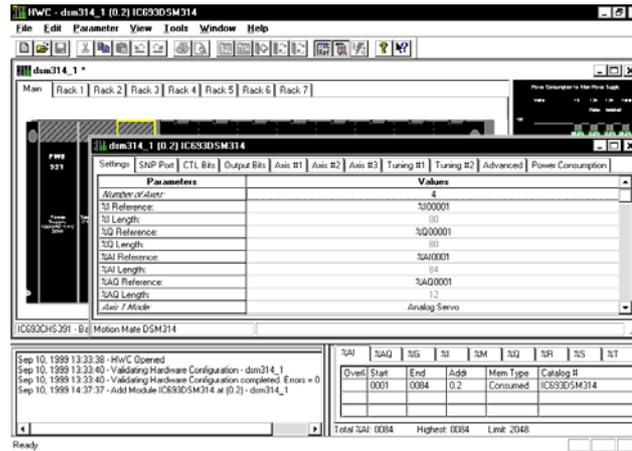


Рис. 15-10. Окно DSM314 при конфигурации аппаратного обеспечения

На рисунке сверху показаны установочные параметры конфигурации модуля DSM314, действующие по умолчанию. Видны только 11 из закладок выбора. Другие закладки, которых в данный момент не видно, появятся при выборе связанных с ними параметров. См. гл. 4 о подробностях установочных параметров конфигурации. Здесь приведена лишь общая информация об этих закладках.

Таблица 15-1. Закладки выбора в окне модуля DSM314 при конфигурации аппаратного обеспечения

Название закладки	Описание
Settings (Установочные параметры)	Содержит назначения и длины ссылок ПЛК, установочные параметры оси DSM и другие глобальные данные
SNP Port (Последовательный порт)	Установка последовательного порта на передней панели модуля DSM (обозначено как COMM).
CTL Bits (CTL-биты)	Конфигурация 24 управляющих битов, используемых внутри DSM.
Output Bits (Биты выхода)	Конфигурация восьми (8) цифровых выходов на лицевой панели DSM.
Axis #1 (Ось #1)	Конфигурация параметров оси, таких как Границы позиции, Начальная скорость и Ускорение толчка
Axis #2 (Ось #2)	
Axis #3 (Ось #3)	
Axis #4 (Ось #4)	
Tuning #1 (Настройка #1)	Конфигурация параметров настройки сервоконтура, таких как Тип двигателя, Постоянная времени контура позиционирования и Предустановленная скорость.
Tuning #2 (Настройка #2)	
Tuning #3 (Настройка #3)	
Tuning #4 (Настройка #4)	
Advanced (Расширенный набор)	Позволяет пользователю ввести пользовательские наборы параметров настройки для любой оси
Power Consumption (Потребление электроэнергии)	Даются значения энергопотребления модулем DSM, требующиеся от источника питания на системной шине (4,0 Вт + питание датчика положения).

- Когда конфигурация модуля завершена, щелкните кнопку закрывания окна конфигурации модуля DSM314 (эта кнопка находится в верхнем правом углу окна конфигурации и обозначена как "X"), чтобы вернуться в окно "Rack View" ("Просмотр крейта"). В это время установки конфигурации еще не сохранены на диске. Они пока находятся в энергозависимой памяти ОЗУ.

### **Запись установок конфигурации на диск**

- Щелкните команду File (Файл) в строке меню, затем в выпавшем меню щелкните пункт Save (Записать). Установочные параметры конфигурации будут записаны в соответствующий файл папки программы. После сохранения файла пункт Save (Записать) из меню файла становится неактивным (цвет его изменяется с черного на слегка зеленоватый). Если в конфигурации будут сделаны какие-то изменения, то пункт Save (Записать) из меню File (Файл) снова вернется в свое активное состояние (его цвет вернется к черному).
- После сохранения файла конфигурации щелкните пункт File (Файл) в строке меню, затем щелкните Exit (Выход) для возврата в главное окно LD.

## Подсоединение к ПЛК и сохранение на нем конфигурации

### Примечание

**Нельзя сохранить конфигурацию на ПЛК, находясь в программе конфигуратора. Чтобы выполнить сохранение в ПЛК, следует быть в главном окне LD VersaPro.**

### Полезные пиктограммы панели инструментов

На следующих шагах для инициации таких операций, как Connect (Соединить), Stop the PLC (Остановить ПЛК) и Store (Сохранить), будет использовано несколько пиктограмм с панели инструментов. Приведенные ниже рисунки показывают эти пиктограммы (значки):

1. Соединить
2. Отсоединить
3. Сохранить в ПЛК
4. Загрузить из ПЛК
5. Свернуть с ПЛК
6. Запустить ПЛК
7. Остановить ПЛК
8. Переключить ссылку
9. Подставить ссылку
10. Записать значение в ссылку
11. Просмотреть статус ПЛК

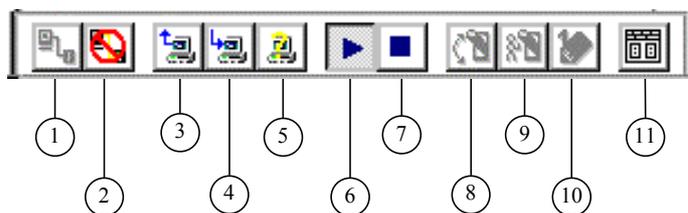


Рис. 15-11. Пиктограммы (значки) панели инструментов VersaPro

### Подсоединение к ПЛК

- Находясь в главном окне VersaPro, щелкните значок "Соединить" на панели инструментов. Появится диалоговое окно Connect (Соединить).

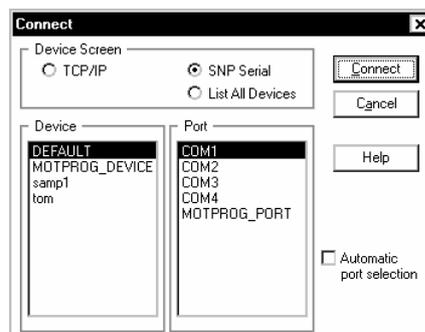


Рис. 15-12. Диалоговое окно Connect (Соединить)

- Если подсоединение выполняется непосредственно к порту программатора ПЛК с последовательного порта COM1 на вашем компьютере, то используйте установки DEFAULT (ПО УМОЛЧАНИЮ), как показано на рисунке выше.
- Убедитесь, что компьютер и последовательный порт на ПЛК соединены кабелем последовательной связи. Щелкните кнопку Connect (Соединить) в диалоговом окне Connect, чтобы установить соединение с ПЛК. На линейке сообщений в нижней части окна VersaPro появится сообщение "Connecting" ("Соединение") с горизонтальной полосой. После установления соединения сообщение в строке статуса изменится с Disconnected (Нет связи) на Connected (Соединение установлено).

## Останов ПЛК

- Чтобы сохранить файлы конфигурации, ПЛК должен быть остановлен; для этого щелкните значок "Стоп" на линейке инструментов. Появится диалоговое окно Stop Execution (Выполнение останова).

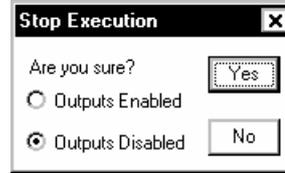


Рис. 15-13. Диалоговое окно выполнения останова

- Для останова ПЛК щелкните "Yes" ("Да"). Сообщение в строке статуса в нижней части окна изменится с Run Enabled (Работа включена) на Stop Disabled (Останов выключен).

## Операция сохранения

- Щелкните значок ПЛК "Сохранить" на линейке инструментов. Появится диалоговое окно Store Folder to PLC (Сохранить папку на ПЛК).

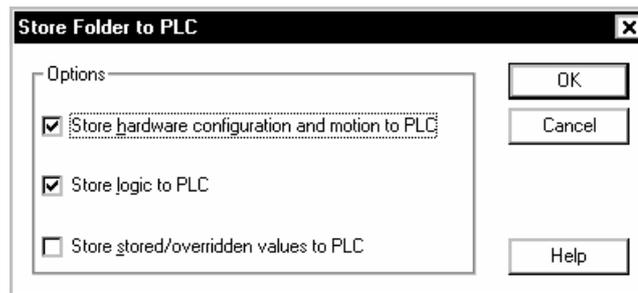


Рис. 15-14. Диалоговое окно сохранения папки на ПЛК

- Убедитесь, что флажок поставлен в пункте "Store hardware configuration and motion to PLC" ("Сохранить конфигурацию аппаратного обеспечения и движение на ПЛК"); щелкните кнопку ОК, чтобы выполнить сохранение на ПЛК. Когда сохранение завершено, сообщение в строке статуса в нижней части окна изменится с Not Equal (Не равно) на Equal (Равно).

## Использование редактора Motion (Движение)

### Выход на окно редактора Motion (Движение)

Выход как на редактор Motion, так и на редактор Local Logic выполняется из окна браузера папки VersaPro. Однако, будучи созданными и сохраненными, программы движения и программы Local Logic становятся частью конфигурации аппаратного обеспечения ЦП ПЛК; они сохраняются в ПЛК вместе с другой информацией конфигурации.

- В главном окне LD щелкните пункт File (Файл) в строке меню и затем выберите New Motion (Новое движение). Затем в меню, появившемся сбоку, щелкните позицию Motion Program (Программа движения) (см. рисунок далее).

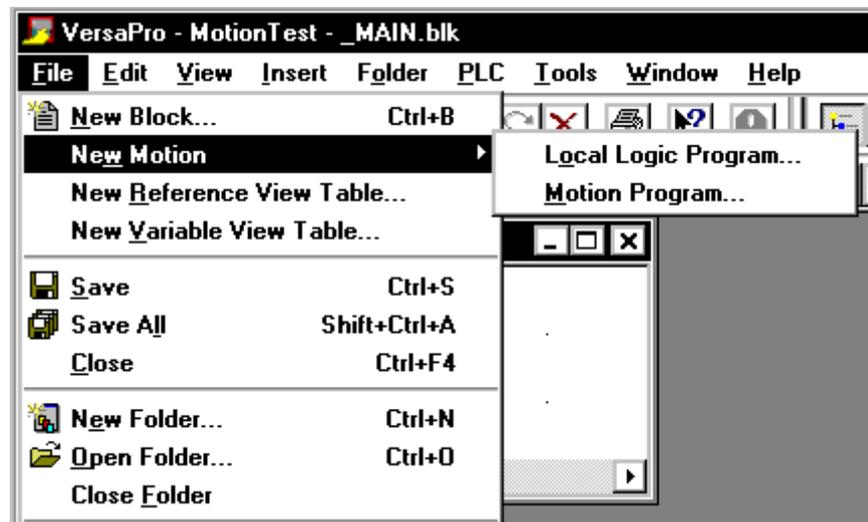


Рис. 15-15. Создание новой программы движения из меню File (Файл)

- Появится диалоговое окно Create New Motion Program (Создать новую программу движения).

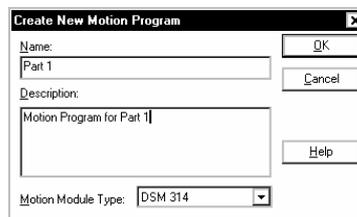


Рис. 15-16. Диалоговое окно создания новой программы движения

- Введите имя и описание программы движения, затем щелкните кнопку ОК (оставьте в поле Motion Module Type (Тип модуля движения) значение по умолчанию DSM314). Откроется окно для блока новой программы движения. Как показано на рисунке ниже, название окна базируется на имени папки (в данном случае Test102) и имени программы движения (в данном случае Part1). Обратите внимание также на рисунке внизу, что в окне браузера папки появляется значок новой программы движения с именем "Part1 – MP" (MP = Motion Program = Программа движения).

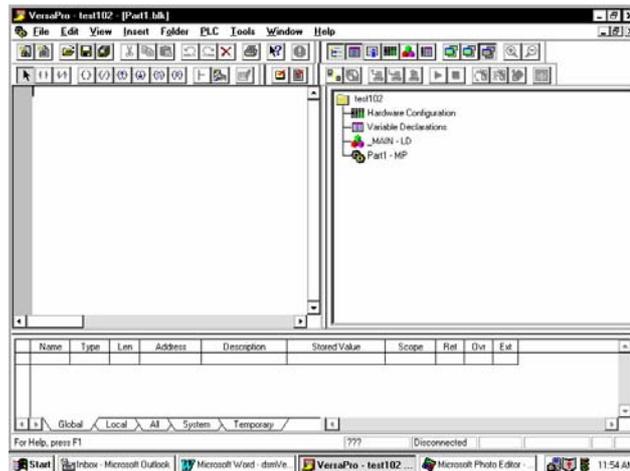


Рис. 15-17. Окно редактора новой программы движения

- В окне редактора Motion создаются текстовые программы и подпрограммы движения, как показано на следующих далее рисунках. В этом окне может быть создано до 10 программ движения и до 40 подпрограмм, которые разделяются их заголовками-идентификаторами (например, "PROGRAM 1 MULTI-AXIS"); все они сохраняются в одном файле. Подробная информация по командам программ движения и по синтаксису находится в главе 7.

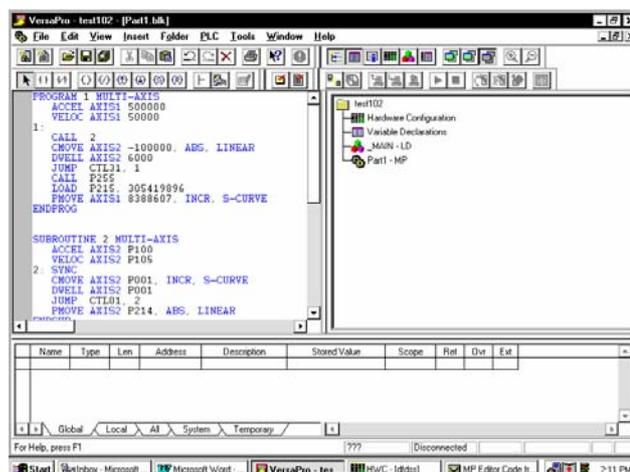


Рис. 15-18. Окно редактора Motion с кодом программы

## Запись программы движения на диск

- Когда файл, содержащий программу или подпрограмму движения, готов к сохранению на жестком диске компьютера, следует щелкнуть либо значок Save (Записать) на линейке инструментов (выглядит как флоппи-диск), либо команду File (Файл) в строке меню и затем позицию Save (Записать).

## Сохранение программ и подпрограмм движения на ПЛК

Поскольку файл программ/подпрограмм движения рассматривается, как часть группы файла конфигурации, то следует использовать процедуру под заголовком "Подсоединение к ПЛК и сохранение на нем конфигурации", описанную выше в данной главе.

## Распечатка программ и подпрограмм движения

В меню File имеются две возможности для печати: Print (Печать) и Print Report (Печать отчета).

### Печать

- В данном пункте описано, как распечатать весь файл (блок) программы движения. В то время как редактор Motion работает, щелкните File (Файл) в строке меню и выберите Print (Печать). Появится диалоговое окно Printer (Принтер). Выполните требуемые установки принтера, затем щелкните кнопку ОК.

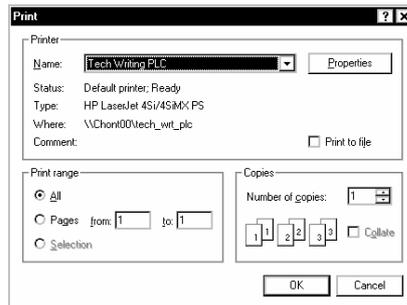


Рис. 15-19. Диалоговое окно печати

- В данном пункте описано, как распечатать выбранную часть файла программы (подпрограммы) движения. В окне редактора Motion выберите при помощи мыши участок, подлежащий распечатке, щелкните File (Файл) в строке меню, затем выберите Print (Печать). Убедитесь, что в диалоговом окне Print (Печать) (показано вверху) в разделе Print range (Диапазон печати) установлен переключатель Selection (Выбор) (в переключателе стоит точка в центре). Щелкните кнопку ОК.

### Печать отчета

- Чтобы распечатать все блоки программы движения (если их более одного), как часть отчета с другой информацией папки, щелкните File (Файл) в строке меню, затем выберите Print Report (Печать отчета). Появится диалоговое окно Print Report (Печать отчета). Щелкните флаговую кнопку Blocks (Блоки) в этом диалоговом окне. Убедитесь, что переключатель All (Все) установлен. (Для отчета могут быть выбраны также другие позиции и функции, как, например, оглавление, перекрестные ссылки, переменные и т.п.). Чтобы начать печать, щелкните ОК. Программа движения, блоки Local Logic и Ladder Diagram будут распечатаны как часть этого отчета.
- Чтобы распечатать только выбранные блоки, выделите их в окне браузера папки. Щелкните File (Файл) в строке меню и выберите Print Report (Печать отчета). Щелкните флаговую кнопку Blocks (Блоки), затем выберите переключатель Selected (Выбрано). Это ограничивает отчеты только теми блоками, которые выделены в окне браузера папки. Пример этого см. на рисунке ниже.

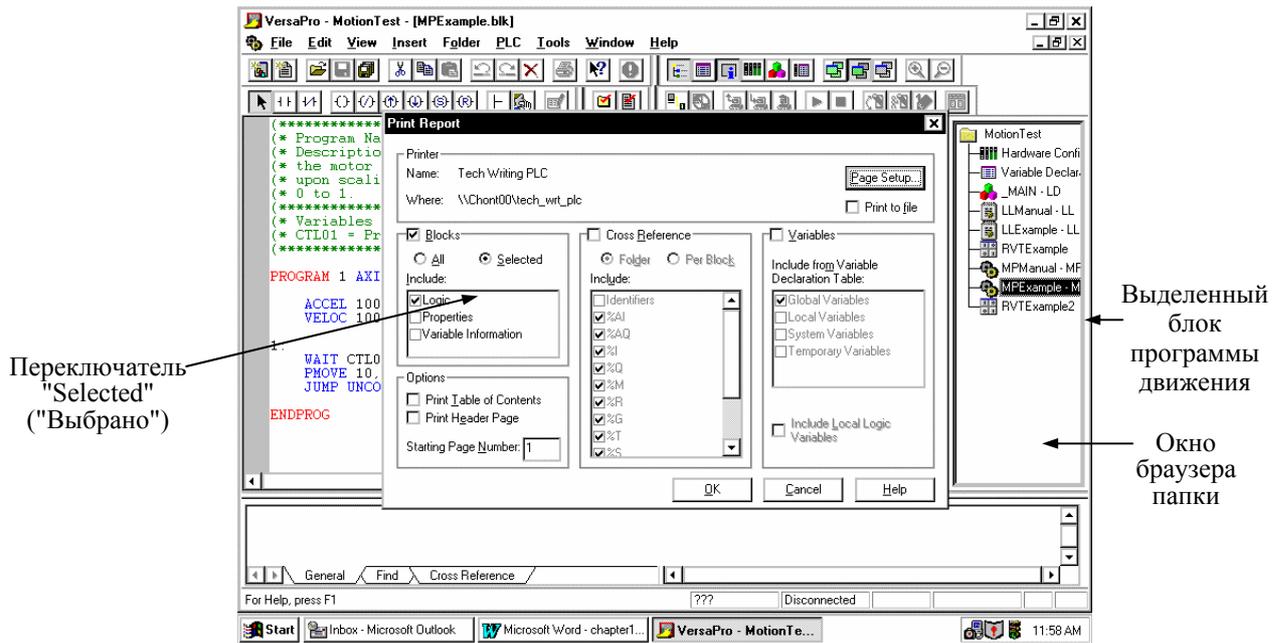


Рис. 15-20. Диалоговое окно печати отчета

## Выход в окно редактора Local Logic

Выход как на редактор Motion, так и на редактор Local Logic выполняется из окна браузера папки VersaPro. Однако, будучи созданными и сохраненными, программы движения и программы Local Logic становятся частью конфигурации аппаратного обеспечения ЦП ПЛК; они сохраняются в ПЛК вместе с другой информацией конфигурации.

- В главном окне LD щелкните пункт File (Файл) в строке меню и затем выберите New Motion (Новое движение). Затем в меню сбоку щелкните строку Local Logic Program (Программа Local Logic).

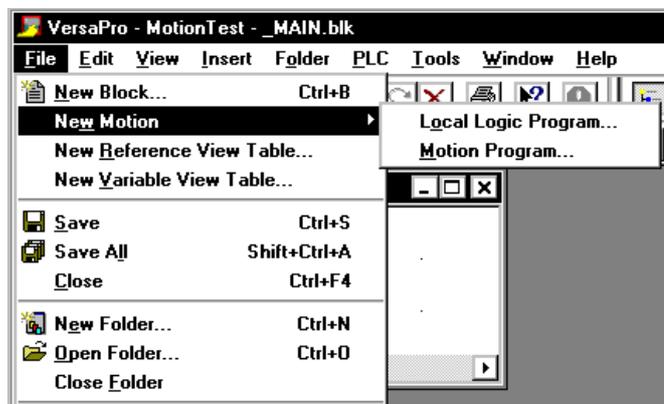


Рис. 15-21. Создание новой программы Local Logic

Появится диалоговое окно Create New Local Logic (Создать новую программу Local Logic).

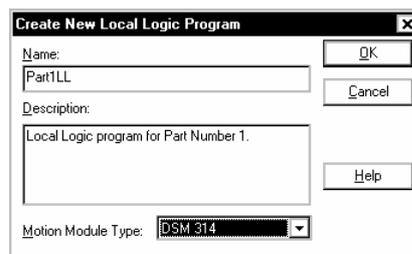
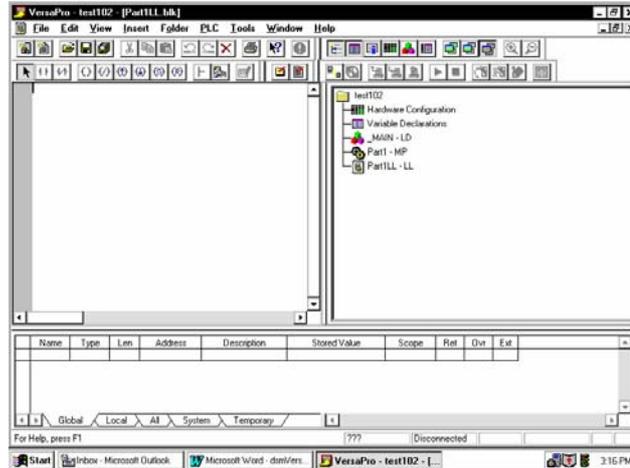


Рис. 15-22. Диалоговое окно создания новой программы Local Logic

- Введите имя и описание программы Local Logic, затем щелкните кнопку ОК (оставьте в поле Motion Module Type (Тип модуля движения) значение по умолчанию DSM314). Откроется окно для блока новой программы Local Logic. Как показано на рисунке ниже, название окна базируется на имени папки (в данном случае Test102) и имени программы Local Logic (в данном случае Part1LL). Программа Local Logic в данном примере не может быть названа "Part1", т.к. это имя уже использовано в имени блока программы движения. Обратите внимание также на рисунок внизу, что в окне браузера папки с правой стороны экрана появляется значок новой программы движения с именем "Part1LL – LL" (Local Logic).



**Рис. 15-23. Окно новой программы Local Logic**

Текстовая программа Local Logic создается в левом окне (окно редактора Local Logic). Подробная информация по командам Local Logic и по синтаксису находится в главах 10 - 14.

## Запись на диск программы Local Logic

Когда файл Local Logic готов к сохранению на жестком диске компьютера, следует щелкнуть либо значок Save (Записать) на линейке инструментов (выглядит как флоппи-диск), либо команду File (Файл) в строке меню и затем позицию Save (Записать).

## Сохранение программы Local Logic на ПЛК

Поскольку файл Local Logic рассматривается как часть группы файла конфигурации, то следует использовать процедуру под заголовком "Подсоединение к ПЛК и сохранение на нем конфигурации", описанную выше в данной главе.

## Распечатка программы Local Logic

В меню File имеются две возможности для печати: Print (Печать) и Print Report (Печать отчета).

### Печать

- **Распечатка всего файла (блока) Local Logic:** В то время как редактор Local Logic работает, щелкните File (Файл) в строке меню и выберите Print (Печать). Появится диалоговое окно Printer (Принтер). Выполните требуемые установки принтера, затем щелкните кнопку ОК.

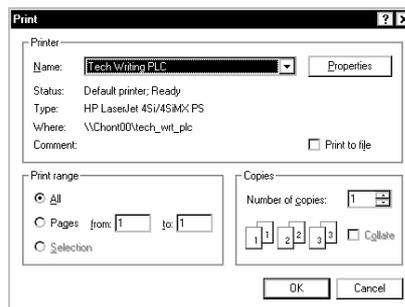


Рис. 15-24. Диалоговое окно печати

- **Печать выбранной части файла Local Logic:** В окне редактора Local Logic выберите при помощи мыши участок, подлежащий распечатке, щелкните File (Файл) в строке меню, затем выберите Print (Печать). Убедитесь, что в диалоговом окне Print (Печать) (показано вверху) в разделе Print range (Диапазон печати) установлен переключатель Selection (Выбор) (в переключателе стоит точка в центре). Щелкните кнопку ОК.

### Печать отчета

- **Чтобы распечатать все блоки Local Logic, как часть отчета с другой информацией папки,** щелкните File (Файл) в строке меню, затем выберите Print Report (Печать отчета). Появится диалоговое окно Print Report (Печать отчета). Щелкните флаговую кнопку Blocks (Блоки) в этом диалоговом окне. Убедитесь, что переключатель All (Все) установлен. (Для отчета могут быть выбраны также другие позиции и функции, как, например, оглавление, перекрестные ссылки, переменные и т.п.). Чтобы начать печать, щелкните ОК. Программа движения, блоки Local Logic и Ladder Diagram будут распечатаны как часть этого отчета.
- **Чтобы распечатать только выбранные блоки Local Logic, как часть отчета,** выделите требуемые блоки Local Logic в окне браузера папки. Щелкните File (Файл) в строке меню и выберите Print Report (Печать отчета). Щелкните флаговую кнопку Blocks (Блоки), затем выберите переключатель Selected (Выбрано). Это ограничивает отчеты только теми блоками, которые выделены в окне браузера папки. Пример этого см. на рисунке ниже.

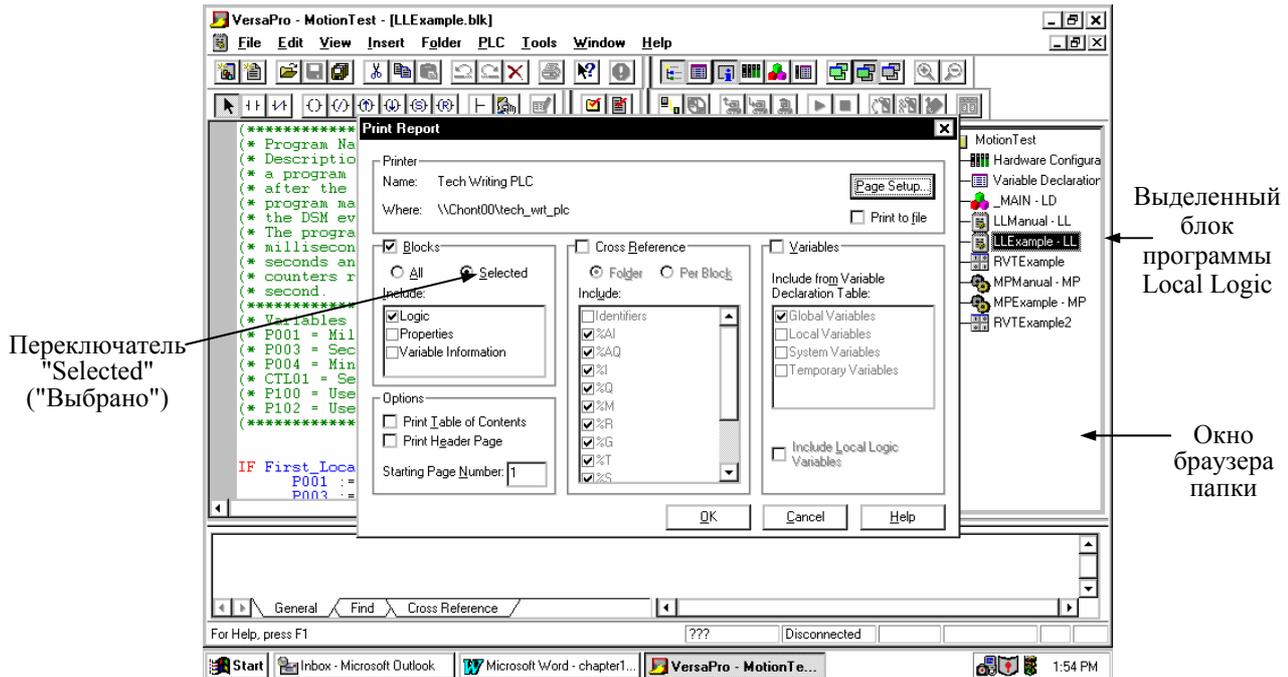


Рис. 15-25. Диалоговое окно печати отчета

## Просмотр таблицы переменных Local Logic

Таблица переменных Local Logic появляется в информационном окне экрана; она содержит информацию о переменных, используемых в программе Local Logic. Чтобы использовать эту функцию, должен существовать блок Local Logic. Если он отсутствует, то его надо создать. Чтобы вывести таблицу на экран, щелкните команду View (Вид) в строке меню, затем в выпадающем меню, показанном на рисунке далее, выберите пункт Local Logic Variable Table (Таблица переменных Local Logic).

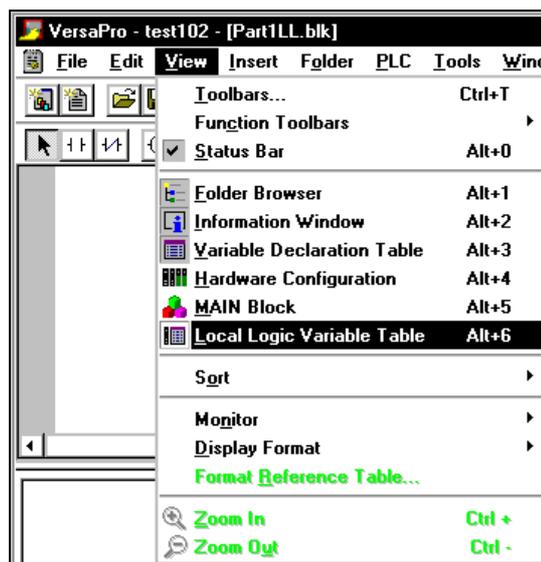


Рис. 15-26. Выбора пункта Local Logic Variable Table (Таблица переменных Local Logic) из меню View (Вид).

Когда данная таблица появится в нижней части окна VersaPro, можно по своему усмотрению перемещать ее верхнюю границу или границы колонок. Сммотри следующий рисунок.

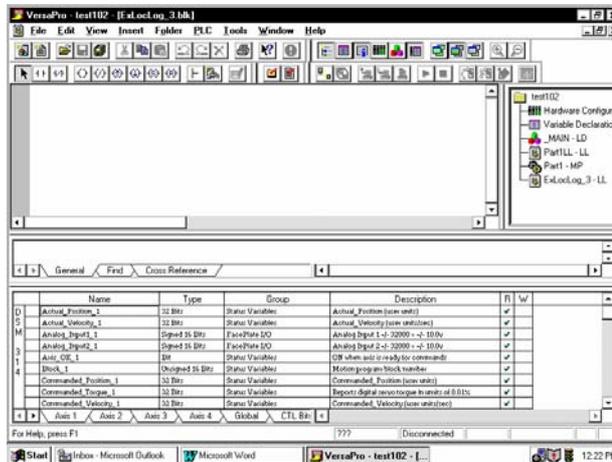


Рис. 15-27. Таблица переменных Local Logic в нижней части экрана

Эта таблица полезна при создании программы Local Logic, поскольку она позволяет копировать и вставлять имена переменных, такие как "Actual\_Position\_1", в вашу программу. Это устраняет необходимость запоминать точные имена переменных, выискивать их в данном руководстве или вводить их с клавиатуры.



## Раздел 1: Введение

Настоящая глава описывает функцию автоматизированного управления (САМ), имеющуюся в версии 2 DSM314. Электронный САМ аналогичен механическому САМ. В большинстве случаев электронный САМ может не только заменить традиционный механический САМ, но и может выполнить множество таких функций, которые отсутствуют у механического аналога. Например, у электронного САМ полностью отсутствует механический износ.

## Краткое описание электронного САМ

Электронные САМ используются в машиностроении, чтобы реализовать сложные движения, которые требуют тесного взаимодействия осей. Имеется много примеров применений, которые предъявляют такие требования. Одним из примеров является использование вращающегося ножа, показанного на рис. 16-1 и 16-2. В данном случае позиция ленты конвейера является ведущей позицией, а вращающийся нож – ведомым. Поскольку позиция ножа связана с ведущей позицией, то нож всегда отслеживает ее при всех ускорениях или замедлениях ленты.

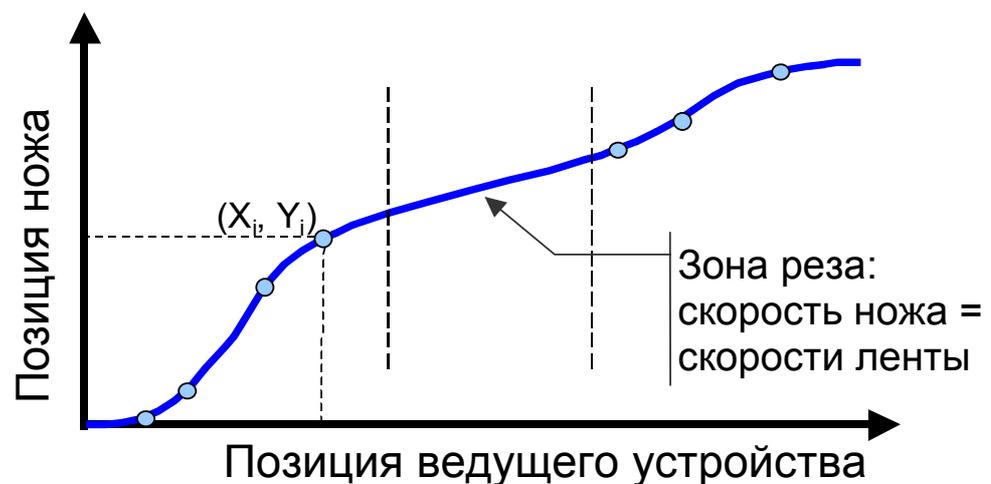


Рис. 16-1. Соответствие позиций для вращающегося ножа

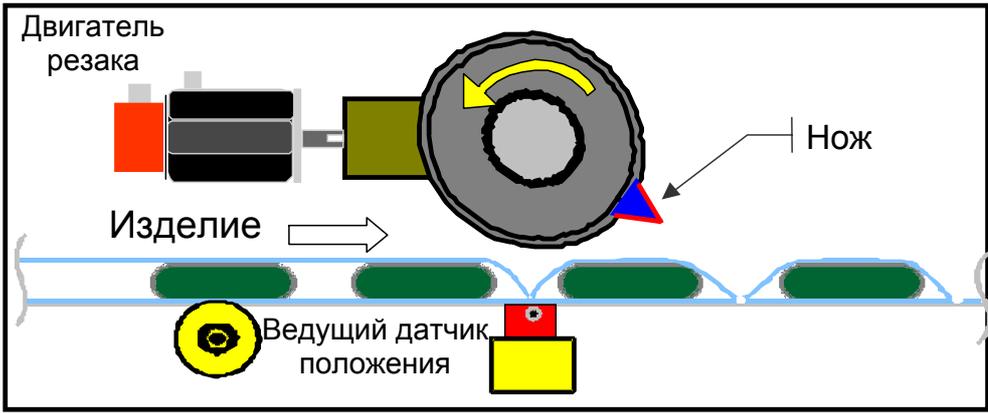


Рис. 16-2. Использование вращающегося ножа

Другим примером использования является линия заполнения бутылок (см. рис. 16-3 и 16-4). В этом случае устройство, которое поднимает и опускает бутылки, выступает в качестве ведущего устройства САМ. Ведомым является поршень, который создает давления для разлива жидкости в бутылки. В этом примере бутылки имеют изогнутую форму. Поэтому скорость подачи жидкости должна меняться, чтобы соответствовать этой форме.

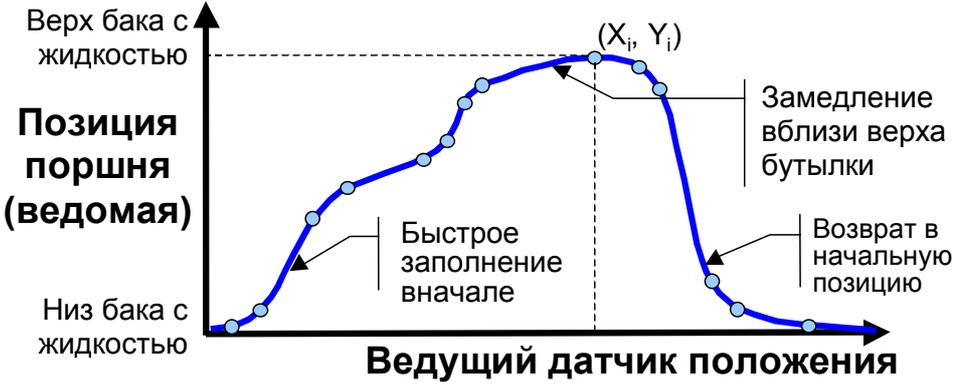


Рис. 16-3. Соответствие позиций при заполнении бутылок

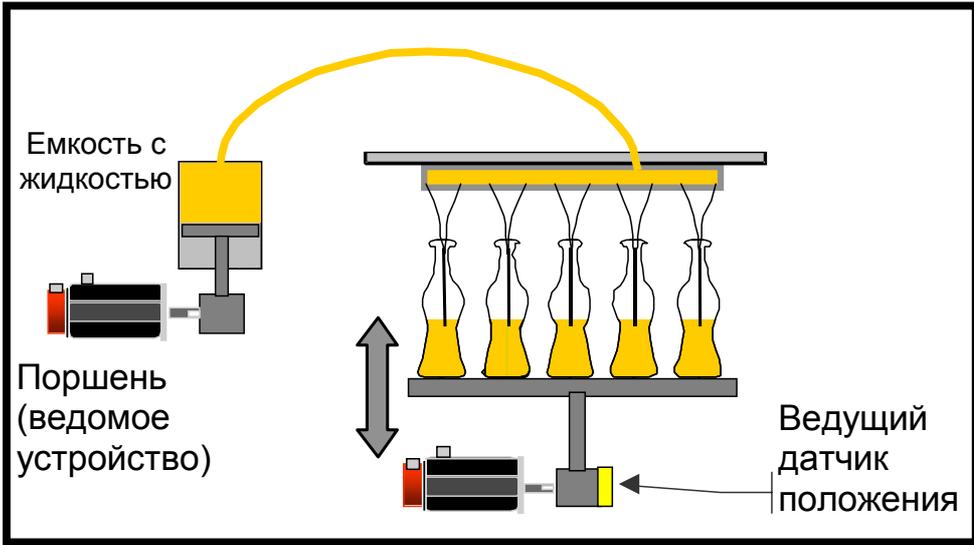


Рис. 16-4. Пример с заполнением бутылок

## Основные элементы формы кулачка (САМ) / определения

Электронные "кулачки" действуют подобно своим механическим аналогам. На рисунке далее показаны элементы базовой механической системы с кулачком и показана ведомая позиция для двух позиций ведущего кулачка. Когда ведущий вал вращается, то ведущий кулачок, который связан с ведущим валом, вращается тоже. Кулачковый следящий элемент (который представляет собой шаровую опору, смонтированную на кулисе) катится по ведущему кулачку при его повороте. Кулачковый следящий элемент толкает вверх или тянет вниз кулису в зависимости от положения ведущего кулачка. Основной рычаг, который прикреплен к кулисе, под ее действием, в свою очередь, движется вверх - вниз, качаясь на опорной призме. Помимо формы ведущего кулачка дополнительными параметрами, которые определяют ведомое движение, являются значение фазы кулачка (интервал, на который позиция ведущего кулачка сдвинута по отношению к позиции ведущего вала), длина рычага кулисы и амплитуда следящего элемента (определяется положением опорной призмы). Все эти механические параметры имеют своих аналогов у электронного кулачка.

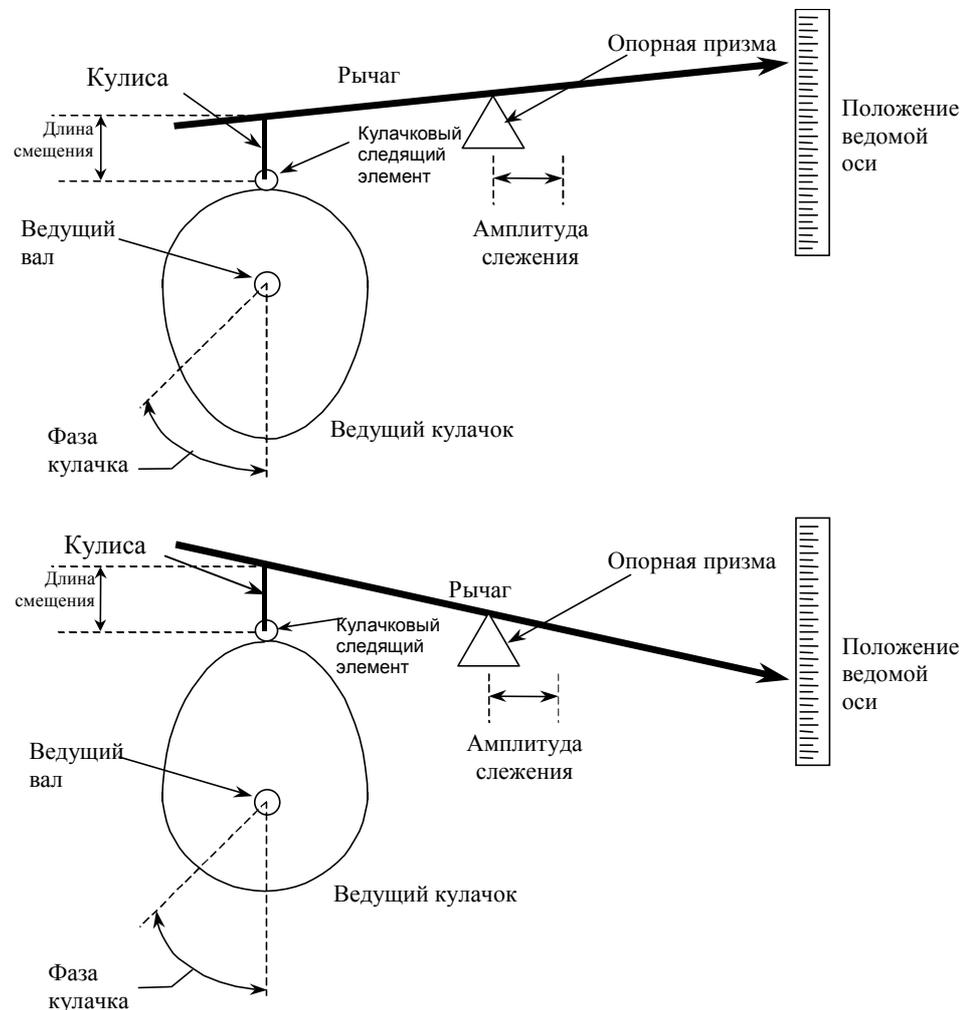


Рис. 16-5. Модель САМ

## Раздел 2: Синтаксис САМ

В данном разделе описаны некоторые важные свойства функции, САМ, даны операторы программы САМ-движения и коды ошибок.

### Типы САМ

Важной концепцией, касающейся функции САМ, является наличие различных типов САМ. Профили САМ могут относиться к одному из следующих типов:

- 1) Нециклический
- 2) Линейный циклический
- 3) Круговой циклический

В разделах далее описывается каждый из этих типов САМ.

#### Нециклический САМ

Нециклический САМ имеет уникальный неповторяющийся профиль для всего диапазона значений ведущей позиции. Выход из САМ происходит при достижении какой-либо из границ САМ-профиля. Выход из САМ происходит также в том случае, если какое-либо внешнее событие сконфигурировано так, чтобы включить условный переход. Отношение числа пользовательских единиц к числу импульсов, заданное для ведущей и ведомой осей, при конфигурировании нециклического САМ должно совпадать с отношением числа пользовательских единиц к числу импульсов, заданным для соответствующих осей в конфигурации аппаратного обеспечения. Значения максимальной и минимальной позиции для ведомой и ведущей осей также должны быть в пределах Верхней / Нижней границы позиции, заданной для соответствующих осей в конфигурации аппаратного обеспечения.

#### Линейный циклический САМ

Линейный циклический САМ имеет профиль, который повторяется до тех пор, пока какое-либо событие не станет причиной выхода. Кроме того, числовые и физические конечные точки ведомой оси САМ являются теми же, что начальная точка цикла. Коленчатый вал, совершающий возвратно-поступательное движение, является примером линейного циклического САМ. Отношение числа пользовательских единиц к числу импульсов, заданное для ведущей и ведомой осей, при конфигурировании САМ-профиля должно совпадать с отношением числа пользовательских единиц к числу импульсов, заданным для соответствующих осей в конфигурации аппаратного обеспечения. На рисунках 16-1 и 16-2 показаны примеры применения линейного циклического САМ.

**Ограничение:** Для линейного циклического САМ первая и последняя ведомая точки должны быть одной и той же точкой. Редактор САМ не покажет опцию для "Linear Cyclic" ("Линейный циклический") в поле "Cam Type" ("Тип кулачка"), если данные в таблице САМ не удовлетворяют этому ограничению.

**Примечания:**

1. Для любого циклического САМ ведущие Верхняя / Нижняя границы в конфигурации аппаратного обеспечения должны быть установлены в соответствии с ведущими точками перехода в профиле САМ. Нижняя граница ведущей оси должна быть равна первой ведущей позиции на профиле. Верхняя граница ведущей оси должна быть равна (Последняя ведущая позиция – 1) в пользовательских единицах. Это определяется тем, что первая и последняя точки циклического профиля являются одной и той же точкой на физическом устройстве.
2. Для линейного циклического САМ происходит переход ведущей оси в конечных точках профиля, однако ведомая ось этого не делает. Значения максимальной и минимальной позиции для ведомой оси должны быть в пределах Верхней / Нижней границы, заданной для соответствующей оси в конфигурации аппаратного обеспечения.

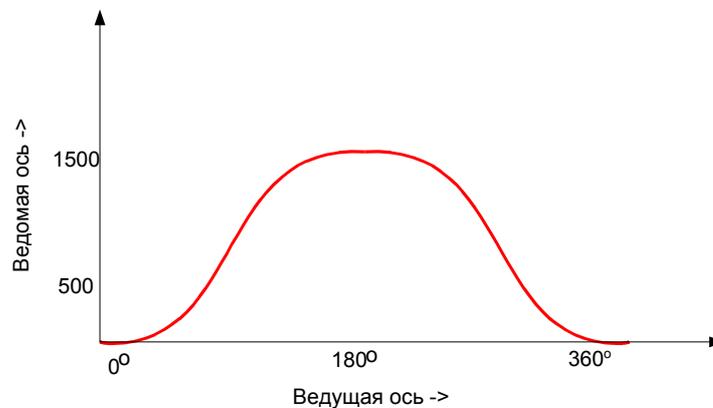


Рис. 16-6. Линейный циклический САМ

**Круговой циклический САМ**

Круговой циклический САМ имеет профиль, который повторяется до тех пор, пока какое-либо событие не станет причиной выхода. Кроме того, круговой циклический САМ имеет различные числовые позиции начала и конца для ведомой оси (см. рис. 16-1). Как ведущая ось, так и ведомая осуществляют переход в конечных точках профиля. Вращающийся нож является примером кругового циклического САМ.

**Ограничение:** Весь ведомый профиль (включая интерполированные значения) должен находиться между минимальной и максимальной границами ведомой позиции; при этом минимальная и максимальная ведомые границы определяются следующим образом:

Минимальное ведомое значение	Максимальное ведомое значение	Условие
Первая ведомая точка	Последняя ведомая точка	Последняя точка > первой точки
Последняя ведомая точка	Первая ведомая точка	Последняя точка < первой точки

### Примечания

1. Редактор не покажет опцию "Circular Cyclic" ("Кольцевой циклический") в поле "CAM Type" ("Тип САМ"), пока не будет выполнено описанное выше ограничение.
2. Для любого циклического САМ ведущая Верхняя и Нижняя границы в конфигурации аппаратного обеспечения должны быть установлены в соответствии с ведущими точками перехода в профиле САМ. Нижняя граница ведущей оси должна быть равна первой ведущей позиции на профиле. Верхняя граница ведущей оси должна быть равна (Последняя ведущая позиция – 1) в пользовательских единицах. Это определяется тем, что первая и последняя точки циклического профиля являются одной и той же точкой на физическом устройстве (например,  $0^\circ$  и  $360^\circ$  для кольцевого ножа).
3. Для кольцевого циклического САМ как ведущая ось, так и ведомая осуществляют переход в конечных точках профиля. Верхняя и Нижняя границы позиции для ведомой оси устанавливаются (в конфигурации аппаратного обеспечения) следующим образом:
  - Если минимальная ведомая позиция является первой точкой профиля, а максимальная ведомая позиция – последней, то следует установить Нижнюю границу позиции на ведомое значение первой точки, а Верхнюю границу позиции – на (ведомое значение последней точки – 1).
  - Если минимальная ведомая позиция является последней точкой профиля, а максимальная ведомая позиция – первой, то следует установить Нижнюю границу позиции на (ведомое значение последней точки + 1), а Верхнюю границу позиции – на ведомое значение первой точки.

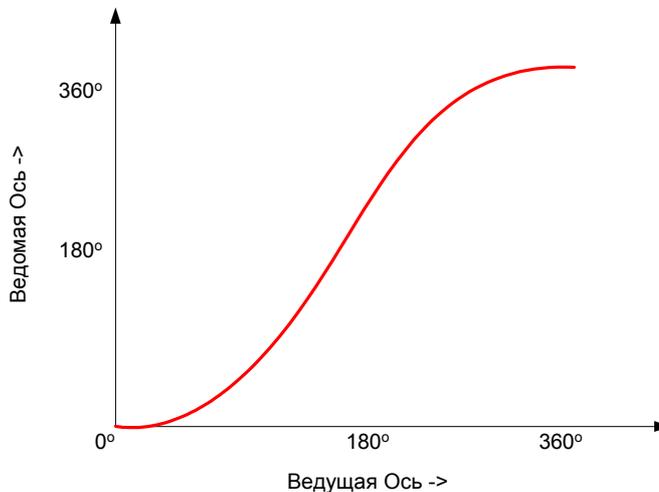


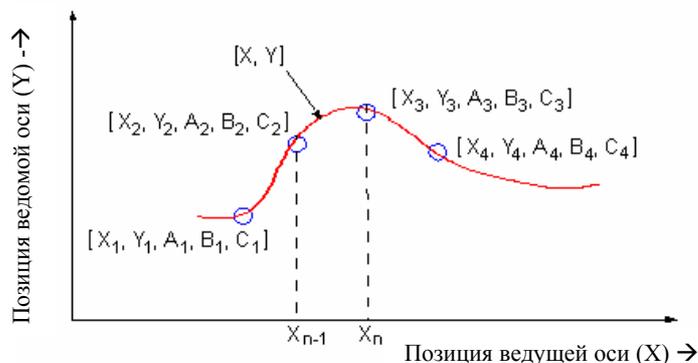
Рис. 16-7. Круговой циклический САМ

## Интерполяция и сглаживание

Одной из ключевых характеристик САМ является схема интерполяции, используемая для задания САМ-профилей. Приведенный ниже текст является перепечаткой раздела из системы справочной информации редактора САМ. Этот текст включен в данный раздел не только для того, чтобы изложить эти важные концепции, но и чтобы подтолкнуть пользователя использовать встроенную систему оперативной консультативной информации редактора САМ для получения дополнительной информации.

Редактор САМ использует полиномиальную интерполяцию с помощью сплайн-функции, чтобы задать профиль движения между точками, определенными пользователем. Такой подход снижает объем памяти, требуемой для хранения профиля на целевом модуле движения, обеспечивая в то же время точную и плавную траекторию движения. Без использования такой интерполяционной схемы потребовалось бы большое количество точек данных для задания каждого профиля, и, соответственно, большой объем памяти.

Профиль САМ задается минимальным количеством текущих точек данных. После задания этих точек они группируются в сектора; профиль составляется из одного или нескольких секторов. Для каждого сектора задается порядок кривой интерполяции (1, 2 или 3). Чем выше порядок, тем плавней выполняется интерполяция. Порядок интерполяционной кривой является порядком полиномиальной кривой, используемой для задания той области сектора, которая не описывается точками, задаваемыми пользователем. Уникальные полиномиальные коэффициенты интерполяционных кривых рассчитываются для каждого сегмента сектора (т.е., между каждой парой точек, определенных пользователем). Коэффициенты полиномов рассчитываются так, чтобы они включали в себя точки, заданные пользователем, и обеспечивали бы совпадение наклона профиля с любой стороны от заданной пользователем точки (за исключением секторов 1-го порядка).



### Примечание

$[X_n, Y_n]$  = координаты ведущей / ведомой оси в точке "n"

$[A_n, B_n, C_n]$  = полиномиальные коэффициенты интерполяции для точек между  $X_{n-1}$  (включительно) и  $X_n$  (не входит)

Полиномиальные кривые для профиля позиции описываются следующей функцией:

$$Y(X) = A_{n-1}(X_n - X_{n-1})^3 + B_{n-1}(X_n - X_{n-1})^2 + C_{n-1}(X_n - X_{n-1}) + Y_{n-1}$$

Где:

$Y$  = значение ведомой позиции при ведущей позиции  $X$ .

$X_{n-1}$  = значение ведущей позиции в точке n-1.

$A_{n-1}, B_{n-1}, C_{n-1}$  = полиномиальные коэффициенты в точке n-1.

### Примечания:

- Для данной ведущей позиции  $X$ , находящейся между  $X_{n-1}$  и  $X_n$ , коэффициенты  $A$ ,  $B$  и  $C$  выбираются такими, которые действуют для точки, соответствующей  $X_{n-1}$ .
- Для кривой второго порядка коэффициент  $A$  всегда равен нулю, а при интерполяции первого порядка всегда равны нулю коэффициенты  $A$  и  $B$ .

## Сопрягающиеся сектора

Процессы, используемые для сопряжения соседних секторов, зависят от порядка действующих на них кривых. Приведенные ниже описания рассматривают возможные сценарии.

### 1-й порядок с 1-ым порядком

Сглаживание перехода между последовательными линейными секторами (т.е. с порядком, равным 1) не делается. Профиль просто соединят прямой линией конечную точку одного сектора с начальной точкой следующего.

### 1-й порядок с 2-ым порядком

Если квадратичный (2-й порядок) сектор следует за линейным сектором (1-й порядок), то полиномиальные коэффициенты для первого сегмента квадратичного сектора рассчитываются так, чтобы наклон профиля был одинаковым с каждой стороны начальной точки этого сектора. Это значит, что начальный наклон квадратичного сектора равен конечному наклону линейного сектора.

### 2-й порядок с 1-ым порядком

Если линейный (1-й порядок) сектор следует за квадратичным сектором (2-й порядок), то какие-либо действия по сглаживанию перехода не выполняются. Такой тип перехода не рекомендуется (если его можно избежать), поскольку он приводит к резкому изменению скорости или ускорения на управляемом сервоустройстве.

### 2-й порядок с 2-ым порядком

Если квадратичный (2-й порядок) сектор следует за другим квадратичным (2-й порядок) сектором, то полиномиальные коэффициенты для первого сегмента второго квадратичного сектора рассчитываются так, чтобы наклон профиля был одинаковым с каждой стороны начальной точки этого сектора. Это значит, что начальный наклон второго квадратичного сектора равен конечному наклону первого квадратичного сектора.

### 2-й порядок с 3-им порядком

Если кубический (3-й порядок) сектор следует за квадратичным (2-й порядок) сектором, то полиномиальные коэффициенты для первого сегмента кубического сектора рассчитываются так, чтобы наклон профиля был одинаковым с каждой стороны начальной точки этого сектора. Это значит, что начальный наклон кубического сектора равен конечному наклону квадратичного сектора.

### 3-й порядок с 2-ым порядком

Если квадратичный (2-й порядок) сектор следует за кубическим (3-й порядок) сектором, то полиномиальные коэффициенты для первого сегмента квадратичного сектора рассчитываются так, чтобы наклон профиля был одинаковым с каждой стороны

начальной точки этого сектора. Это значит, что начальный наклон квадратичного сектора равен конечному наклону кубического сектора.

### 3-й порядок с 3-им порядком

Если сопрягаются два кубических (3-й порядок) сектора, то наклоны профиля до и после точки их стыковки делаются равными. Т.е., вторые производные профилей до и после точки, где сектора сопрягаются, должны быть одинаковыми. Полиномиальные коэффициенты интерполяционных кривых для двух сопрягающихся сегментов рассчитываются одновременно.

## Граничные условия

Для нециклических профилей необходимо задать определенное условие в начальной и конечной точке профиля, чтобы рассчитать полиномиальные коэффициенты интерполирующих кривых. Начальным или конечным граничным условием может быть следующее:

- Числовое значение 1-й производной профиля (наклон).
- Числовое значение 2-й производной профиля.
- Условия расчета, принимаемые по умолчанию.

Условиями расчета по умолчанию являются следующие условия:

- **Начальная граница.** Наклон в начальной точке профиля рассчитывается в результате промежуточной подгонки полиномиальной кривой к первым трем (сектор 2-го порядка) или четырем (сектор 3-го порядка) точкам с последующим расчетом наклона такого промежуточного полинома в первой точке.
- **Конечная граница** Наклон в конечной точке профиля рассчитывается в результате промежуточной подгонки полиномиальной кривой к последним четырем точкам (сектор 3-го порядка) с последующим расчетом наклона такого промежуточного полинома в конечной точке.

## Взаимодействие программ движения с САМ

САМ-движение может быть инициировано в DSM314 командами в программе движения. Для поддержки программирования САМ-движения требуются следующие новые команды движения:

- 1) **САМ**: используется в программе движения для запуска САМ-движения и задания условий выхода.
- 2) **САМ-LOAD**: используется для загрузки начального положения ведомой САМ-оси в регистр параметров. Вместе с командой САМ-LOAD может быть использована команда PMOVE, чтобы переместить ведомую ось в начальную точку.
- 3) **САМ-PHASE**: используется для задания фазы САМ-команды. Значение фазы может быть задано либо через регистр параметров, либо в виде константы.

Приведенные далее секции описывают подробно синтаксис и функции каждой из указанных выше команд. Для синтаксиса команд действует следующее соглашение:

Скобки "<>" - указывают требуемое поле.

Скобки "[" "]" - указывают дополнительное поле.

Скобки "{" "}" - указывают поле, которое требуется для многоосных программ и подпрограмм, но которое не разрешено для программ и подпрограмм с одной осью.

### САМ-команда

САМ-команда используется для программирования САМ-движения на базе заданного САМ-профиля.

#### Синтаксис:

САМ <"Имя профиля САМ">, <расстояние>, <режим ведущего>, [Условие циклического выхода]

Параметр	Описание
<"Имя профиля САМ">	Имя профиля САМ из САМ-библиотеки (этот профиль должен быть связан с блоком загрузки САМ). Это имя ограничено 20-ю знаками (см. также примечание 3 внизу). <b>Обратите внимание, что имя должно быть в кавычках.</b>
<расстояние>	Максимальное расстояние, на которое может переместиться ведущая ось после включения САМ (в пользовательских единицах). В качестве расстояния может выступать константа, параметр или ключевое слово NONE. Допустимый диапазон: $-\text{MaxPosn} \dots (\text{MaxPosn}-1)$ пользовательские единицы / импульсы
<режим ведущего>	Режим ведущего может быть задан как ABS (абсолютный) или INCR (инкрементный); он указывает на то, как следует интерпретировать данные позиции ведущего. В режиме ABS абсолютная позиция ведущей оси используется для определения ведомого значения из таблицы САМ. В режиме INCR значение ведущего в начальной точке САМ-команды принимается равным значению "САМ-фазы"; ведомые значения, рассчитанные при САМ-движении, являются относительными по отношению к данному начальному значению ведущего.
[Условие циклического выхода]	Определяет условия выхода для циклических САМ. Разрешенным диапазоном является STL01-STL32. Если этот STL-бит принимает значение логической единицы ("истина"), то происходит выход из САМ в конце текущего цикла. <b>Примечание: этот параметр не следует использовать для не циклического САМ-профиля.</b>

**Примечания:**

1. Не разрешается использование САМ-команды в многоосной программе движения.
2. САМ-команда считается как две команды в рамках ограничения (1000 команд), действующего для программы движения.
3. Имена профилей **UDT\_CAM\_1**, **UDT\_CAM\_2**, **UDT\_CAM\_3** и **UDT\_CAM\_4** зарезервированы для будущего использования.

Аргумент **<расстояние>** в САМ-команде задает максимальное расстояние, которое ведущая ось может пройти по профилю до выхода. Для циклических САМ это расстояние может быть больше или меньше длины САМ-таблицы (определяется как абсолютная разность между первым и последним значениями позиции ведущего в САМ-таблице). Если указанное расстояние меньше, чем длина таблицы, то САМ-команда выполняется один раз, пока проходит это расстояние. Если расстояние больше длины таблицы, то САМ будет действовать по циклу таблицы САМ, пока не будет пройдено все расстояние. Таким образом, пользователь может задать количество циклов, которое должно быть выполнено циклическим САМ. Например, задание расстояния в 2.5 раза больше, чем длина ведущей позиции из САМ-таблицы, приведет к тому, что САМ-профиль будет "отработан" два с половиной раза, после чего произойдет выход. Для нециклических САМ заданное расстояние не может быть более длины таблицы.

Расстояние может, быть задано как "NONE". Для циклического САМ это будет означать непрерывное САМ-движение до тех пор, пока STL-бит не вызовет переключения на выход или пока движение не будет прекращено. Для нециклического САМ задание "NONE" означает выход САМ при достижении минимальной или максимальной ведущей позиции профиля.

**[режим ведущего]** используется для задания абсолютного или инкрементного режима ведущей оси. Ведущая ось может работать как в абсолютном, так и в инкрементном режиме в циклических, и в нециклических САМ. В абсолютном режиме ведущие позиции в таблице представляют собой абсолютные позиции ведущей оси. В инкрементном режиме позиции ведомой оси в таблице являются относительными по отношению к позиции ведущей оси, когда выдается САМ-команда.

**[Условие циклического выхода]** задает условие выхода для циклического САМ-профиля. Если этот STL-бит принимает значение логической единицы ("ИСТИНА"), то происходит выход из САМ в конце текущего цикла.

**Двунаправленные** и **однонаправленные** САМ могут быть заданы при использовании параметров ограничения скорости ведущей оси  $+V_{lim}$  и  $-V_{lim}$ . Для однонаправленной работы соответствующая граница скорости должна быть установлена на ноль для того движения, движение в котором запрещается. Например, если движение в отрицательном направлении запрещено, то  $-V_{lim}$  должно быть установлено на ноль.

## Команда CAM-LOAD

Команда CAM-LOAD используется для того, чтобы загрузить позицию ведомой оси в регистр параметров. Затем может быть использована стандартная команда PMOVE, чтобы переместить ведомую ось в загруженную позицию. Команда CAM-LOAD использует имя профиля CAM, текущую позицию и фазу (задаваемую командой CAM-PHASE) ведущей оси, чтобы определить начальную точку ведомой оси.

### Синтаксис:

**CAM-LOAD** <"Имя профиля CAM">, <Номер параметра>, <режим ведущего>

Параметр	Описание
<"Имя профиля CAM">	Имя профиля CAM из CAM - библиотеки (этот профиль должен быть связан с блоком загрузки CAM). Это имя ограничено 20-ю знаками (см. также примечание 3 внизу). <b>Обратите внимание, что имя должно быть в кавычках.</b>
<Номер параметра>	Определяет подлежащий загрузке номер параметра
<режим ведущего>	Режим ведущего может быть задан как ABS (абсолютный) или INCR (инкрементный); он указывает на то, как следует интерпретировать данные позиции ведущего при расчете начальной позиции ведомой оси.  В режиме ABS абсолютная позиция ведущей оси используется для определения соответствующего значения начальной позиции ведомой оси из таблицы CAM. В режиме INCR ведущее значение принимается в расчетах равное значению CAM-Phase.

### Примечания:

1. Команда CAM-LOAD считается как две команды в рамках ограничения (1000 команд), действующего для программы движения.
2. Когда выполняется команда CAM-LOAD, то происходит следующая последовательность действий:
  - A. Считывается текущая позиция ведущей оси.
  - B. Используя ведущую позицию, значение CAM-Phase, таблицу CAM-профиля и таблицу конфигурации CAM, рассчитывается соответствующая позиция ведомой оси; эта позиция загружается в назначенный регистр параметров.
  - C. Программа движения может использовать команду PMOVE для перемещения ведомой оси в позицию, рассчитанную на шаге B.
3. Имена UDT\_CAM\_1, UDT\_CAM\_2, UDT\_CAM\_3 и UDT\_CAM\_4 зарезервированы для будущего использования и не могут быть использованы как имена профилей CAM.

## Команда САМ-PHASE

Команда САМ-PHASE (САМ-ФАЗА) задает фазу команд САМ. Эта команда позволяет выполнить сдвиг или смещение фазового отношения между позициями ведущей и следящей осей. Значение фазы может быть задано либо через регистр параметров, либо в виде константы. Обратите внимание, что значение фазы действительно для всех команд САМ, которые следуют за нею, до тех пор, пока оно не будет изменено другой командой САМ-PHASE. Значение Cam Phase (Фаза кулачка) по умолчанию для программы движения равно 0.

**Синтаксис:**

САМ-PHASE <Фаза>

Параметр	Описание
<Фаза>	Значение фазы САМ задается как константа или как регистр параметров. Допустимый диапазон для константы: $-\text{MaxPosn} \dots (\text{MaxPosn}-1)$

## Команды САМ и MOVE

Последовательность команд САМ может быть выполнена без какой-либо задержки или прерывания. Чтобы получить плавное движение, следует обеспечить, чтобы начальная точка на каждом последующем профиле САМ была конечной точкой предшествующего САМ-профиля. Это обеспечивает непрерывность траектории по позиции и скорости. Для последовательности не циклических САМ начальная и конечная точки могут быть согласованы в редакторе САМ, чтобы получить плавные переходы. В настоящее время переходы между командами САМ и MOVE во время движения ведомой оси не разрешаются. Поэтому ведомая ось должна иметь начальную скорость, равную 0, в точке перехода между командами САМ и MOVE. Когда появляется команда САМ, то – если сразу же за ней не следует другая команда САМ – ось будет использовать запрограммированное изменение скорости, чтобы остановиться.

## Движение САМ, контролируемое по времени

Использование САМ-профиля, контролируемого во времени, связано с тем же механизмом, что и использование стандартного САМ (ведущая ось, контролируемая по положению). Чтобы запрограммировать профиль САМ, контролируемый по времени, ведущий источник САМ должен быть сконфигурирован как "Commanded Position" ("Заданная позиция") оси 3 в конфигурации аппаратного обеспечения модуля DSM314, а режим оси 3 должен быть установлен на "Auxiliary Axis" ("Вспомогательная ось"). Затем на оси 3 инициируется команда постоянной скорости. Эффективная временная шкала движения САМ определяется масштабированием ведущей оси в исходном файле профиля и коэффициентом пересчета пользовательских единиц к импульсам, заданным в конфигурации аппаратного обеспечения. САМ-команда движения, контролируемого по времени, может выполняться одновременно на нескольких осях.

## Редактор масштабирования САМ и конфигурация аппаратного обеспечения

Модуль DSM позволяет пользователю масштабировать разрешение устройства обратной связи по положению в зависимости от единиц программирования модуля. Например, предположим, что 1 оборот двигателя соответствует 1 дюйму перемещения приводимой в движение нагрузки. В данном примере двигатель, связанный с нагрузкой, имеет датчик положения, который выдает 8192 импульса на один оборот двигателя. Эти 8192 импульса обратной связи соответствуют 1 дюйму перемещения нагрузки. Некоторым пользователям легче программировать движение в дюймах, а не в импульсах обратной связи. В этом случае следует установить масштаб для программного движения в тысячных долях дюйма. Чтобы получить такой результат, надо в конфигурации аппаратного обеспечения модуля DSM задать отношение числа пользовательских единиц к числу импульсов как 1000 к 8192. Дополнительная информация по заданию этих значений находится в главе 5.

Функция САМ поддерживает также специфические для применения единицы. Однако значения, введенные в конфигурации аппаратного обеспечения, должны быть перенесены вручную в соответствующие поля в редакторе САМ.

**Примечание:** Эти значения должны быть перенесены как для ведущей, так и для ведомой оси. Основываясь на предыдущем примере, предположим, что ведущая и ведомая ось имеют одинаковые двигатели. Поэтому каждое устройство обратной связи выдает одни и те же 8192 импульса на оборот. Однако для ведущей оси один оборот двигателя равен одному дюйму перемещения нагрузки, тогда как для ведомой оси один оборот двигателя равен половине дюйма перемещения нагрузки. Чтобы сделать программы более легкими для понимания, ведущую и ведомую оси следует программировать в одних и тех же единицах. В настоящем примере используются единицы, равные 0.001 дюйма. Чтобы получить этот результат, определим сначала правильные отношения числа пользовательских единиц к числу импульсов для ведущей и ведомой осей.

Чтобы определить это отношение, используем следующее уравнение

$$\frac{\text{Пользов. ед.}}{\text{Импульсы}} = \left( \frac{\text{Перемещение нагрузки на обор. двиг.}}{\text{Требуемое разрешение}} \right) \cdot \frac{1}{\text{Число импульсов обратной связи на обор. двиг.}}$$

Для ведущей оси в данном примере:

$$\frac{\text{Пользов. ед.}}{\text{Импульсы}} = \left( \frac{1 \text{ дюйм}}{1} \right) \cdot \frac{1}{8192 \text{ импульса}}$$

$$\frac{\text{Пользов. ед.}}{\text{Импульсы}} = \left( \frac{1000}{8192} \right) \cdot \frac{\text{дюймы}}{\text{Импульсы}}$$

Для ведомой оси в данном примере:

$$\frac{\text{Пользов. ед.}}{\text{Импульсы}} = \left( \frac{0.5 \text{ дюйма}}{1} \right) \cdot \frac{1}{8192 \text{ импульса}}$$

$$\frac{\text{Пользов. ед.}}{\text{Импульсы}} = \left( \frac{500}{8192} \right) \cdot \frac{\text{дюймы}}{\text{импульсы}}$$

Полученные значения следует ввести в соответствующие поля при конфигурации аппаратного обеспечения. В данном примере Axis #2 является ведущей, а Axis #1 – ведомой. Тогда введите значения пользовательских единиц и импульсов в конфигурации аппаратного обеспечения для ведомой оси так, как показано на рис. 16-1.

Parameters	Values
User Units:	500
Counts:	8192
Over Travel Limit Switch:	Disabled
Drive Ready Input:	Enabled

Рис. 16-8. Ввод значений пользовательских единиц и импульсов при конфигурации аппаратного обеспечения ведомой оси

Значения пользовательских единиц и импульсов для ведущей оси вводятся, как показано на рис. 16-2.

Parameters	Values
User Units:	1000
Counts:	8192
Over Travel Limit Switch:	Disabled
Drive Ready Input:	Disabled
High Position Limit:	49999

Рис. 16-9. Ввод значений пользовательских единиц и импульсов при конфигурации аппаратного обеспечения ведущей оси

Это дает модулю правильное масштабирование при работе с программами движения. Однако правильное масштабирование должно быть доведено также до редактора САМ, чтобы выполнять правильное преобразование пользовательских единиц в импульсы. Для данного примера эти данные должны быть введены в любой из САМ-профилей, который предполагается использовать с рассматриваемыми осями. Пример показан на рисунке 16-3.

- Profile Link	
Profile Name	Scale_Non_Cyc
CAM Type	Non-Cyclic CAM
Start Type	Ignored
End Type	Ignored
Master Device Counts	8192
Master User Counts	1000
Slave Device Counts	8192
Slave User Counts	500

Рис. 16-10. Ввод числа пользовательских единиц и числа импульсов для ведомой и ведущей осей в редакторе САМ

Рекомендуется выполнить рассмотренные выше операции до начала программирования любого САМ, если только он не имеет масштаб "один к одному". Причина такого предложения состоит в том, чтобы устранить повторный ввод данных. Редактор САМ отображает данные ведущей и ведомой осей в пользовательских единицах. Поэтому, если масштаб не определен и выполнен ввод всех САМ-данных по умолчанию, то это значит, что выбран масштаб 1:1. Если в завершение эта ошибка исправлена и введен правильный масштаб, то все ненулевые значения в таблицах САМ-данных изменятся, чтобы отразить новое значение пользовательских единиц.

В следующей секции рассматривается, как редактор САМ округляет значения, когда пользователь вводит данные. Эта функция осуществляется автоматически; не требуется, чтобы пользователь для этого осуществлял какие-то действия или конфигурировал специально редактор. Этот раздел предлагается для окончания темы масштабирования, рассмотренной в предыдущих параграфах.

Следует отметить, что внутри себя модуль DSM работает в естественных единицах обратной связи и преобразует эти естественные единицы в пользовательские единицы автоматически для пользователя. Модуль выполняет эту операцию, чтобы полностью использовать преимущества всего доступного разрешения обратной связи. Это относится также к случаю, когда пользователь выбрал программное движение в единицах, которые не используют полное разрешение устройства обратной связи. Редактор САМ также стремится сохранить все доступное разрешение (без показа неправильного разрешения) для данного комплекта двигателя и устройства обратной связи. Поэтому при задании масштаба в редакторе САМ редактор в некоторых случаях может добавить десятичные значения в таблице данных. Кроме того, он автоматически округляет числа до значений, которые могут быть представлены как целые числа импульсов обратной связи. Таблица 16-1 с примером САМ базируется на рассмотренном выше случае. Следует отметить, что редактор САМ отображает значения в пользовательских единицах, но всегда округляет их до целого значения в импульсах.

**Таблица 16-1. Пример данных САМ, масштабированных в дюймах**

Позиция ведущей оси (дюймы)	Позиция ведомой оси (дюймы)
0	0
0.075	0.075
0.5	0.25
1	0.5

Таблица вверху показывает значения в дюймах. В данном примере САМ запрограммировано в тысячных долях дюйма. Поэтому следует преобразовать эти значения и ввести данные в редактор САМ, как показано в таблице 16-2

**Таблица 16-2. Пример данных САМ, масштабированных в 0.001 дюйма**

Позиция ведущей оси (тысячные доли дюйма)	Позиция ведомой оси (тысячные доли дюйма)
0 дюймов = 0 тысячных долей дюйма	0 дюймов = 0 тысячных долей дюйма
0.075 дюйма = 75 тысячных долей дюйма	0.075 дюйма = 75 тысячных долей дюйма
0.5 дюйма = 500 тысячных долей дюйма	0.25 дюйма = 250 тысячных долей дюйма
1 дюйм = 1000 тысячных долей дюйма	0.5 дюйма = 500 тысячных долей дюйма

При вводе данных в редактор САМ (рис. 16-4) некоторые значение редактором САМ изменяются автоматически.

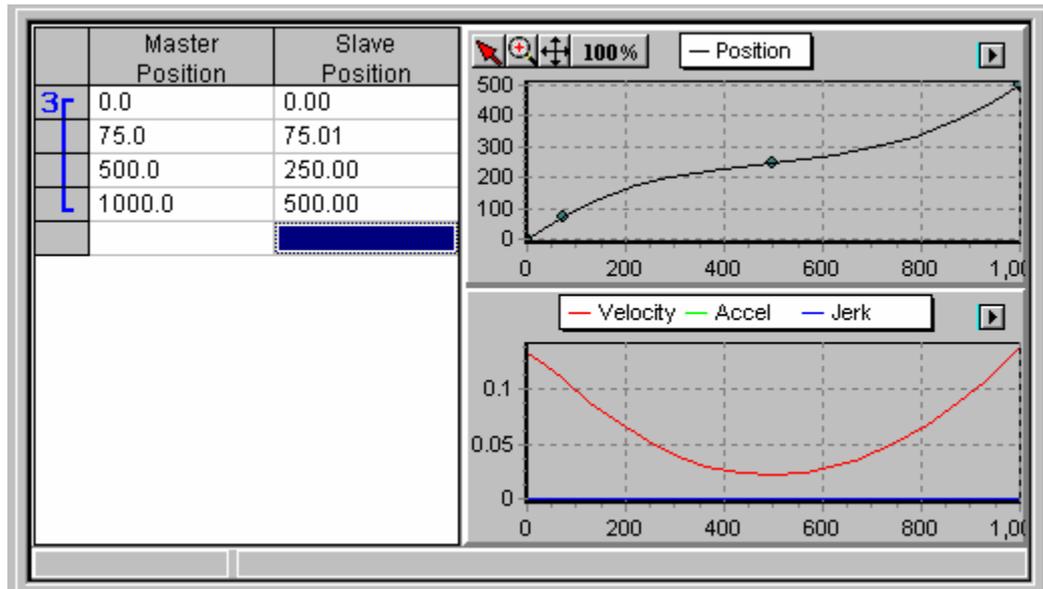


Рис. 16-11. Пример пользовательских единиц в таблице данных САМ

Редактор САМ автоматически изменяет значения так, чтобы они соответствовали целому числу импульсов обратной связи. Редактор округляет отображаемые значения также для того, чтобы ограничить "шумы" внутри таблицы. Следует отметить, что редактор сохраняет точность переменных, а округление отображаются только на дисплее. Функции, которые автоматически выполняет редактор САМ, проиллюстрированы ниже. Прежде всего, он определяет целое число импульсов обратной связи, которое ближе всего к требуемому значению. В данном примере только два числа не могут быть представлены точно. Это позиция 75 ведущей оси и позиция 75 ведомой оси. Наиболее близкими целыми значениями числа импульсов для этих величин являются 614 и 1229 импульсов для ведущей и ведомой оси, соответственно. Соотношение между ведущей и ведомой позицией в пользовательских единицах и в импульсах показано в табл. 16-3.

Ведущая позиция (пользовательские единицы)	Ведущая позиция (импульсы)	Ведомая позиция (пользовательские единицы)	Ведомая позиция (импульсы)
0	0	0	0
74.951171875	614	75.01000000000001	1229
500	4096	250	4096
1000	8192	500	8192

Таблица 16-3. Соотношение между пользовательскими единицами и импульсами в примере масштабирования

Это соответствует тем функциям, которые были автоматически выполнены редактором. Следует отметить, что для ведущей позиции 74.951171875, редактор производит округление до 75.0. Ведомую позицию 75.01000000000001 редактор округляет до 75.01.

## Синхронизация САМ-движения с внешними событиями

Описанные ниже механизмы позволяют программисту синхронизировать САМ-движение с внешними событиями:

- Запуск САМ-движения может быть синхронизирован с внешним событием при помощи существующей команды WAIT в программе движения.
- Циклический САМ может быть синхронизирован со стробом при помощи переменных Local Logic. Дополнительную информацию по Local Logic см. в главах 11 – 14.

## Коды ошибок модуля DSM, специфические для САМ

Таблица 16-4. Коды ошибок, специфические для САМ

<b>Коды ошибок программ САМ</b>				
<b>Код ошибки (шестнадцатичный)</b>	<b>Реакция</b>	<b>Описание</b>	<b>Тип ошибки</b>	<b>Возможная причина</b>
2A	Нормальный останов	CTL-условие выхода циклического САМ, заданное для не циклического САМ	Ось	CTL-условия выхода разрешены только для циклического САМ. Программа движения получает команду не циклического САМ, которая содержит CTL-условие выхода.
2B	Нормальный останов	Значение фазы САМ вне диапазона	Ось	Значение САМ PHASE находится за пределами диапазона позиционирования оси.
<b>Коды ошибок конфигурации САМ</b>				
2D	Нормальный останов	Ошибка конфигурации ведущей оси САМ – ведущий профиль не соответствует конфигурации ведущей оси.	Ось	Отношение числа пользовательских единиц к числу импульсов, заданное для ведущей оси в редакторе и в конфигурации аппаратного обеспечения, не совпадают и / или Верхняя / Нижняя границы позиции, заданные для ведущей оси в конфигурации аппаратного обеспечения, не соответствуют профилю. См. раздел по типам САМ для подробного описания установок Верхней / Нижней границ позиционирования.
2E	Нормальный останов	Ошибка конфигурации ведомой оси САМ – ведомый профиль не соответствует конфигурации ведомой оси.	Ось	Отношение числа пользовательских единиц к числу импульсов, заданное для ведомой оси в редакторе и в конфигурации аппаратного обеспечения, не совпадают и / или Верхняя / Нижняя границы позиции, заданные для ведомой оси в конфигурации аппаратного обеспечения, не соответствуют профилю. См. раздел по типам САМ с подробным описанием установок Верхней / Нижней границ позиционирования.
2F	Нормальный останов	Режим SW EOT ведомой оси САМ не может быть разрешен для кругового циклического САМ	Ось	

<b>Коды ошибок параметров конфигурации</b>				
<b>Код ошибки (шестнадцатичный)</b>	<b>Реакция</b>	<b>Описание</b>	<b>Тип ошибки</b>	<b>Возможная причина</b>
1D	Нормальный останов	Попытка использовать команды CAM, CAM-Load или CAM-Phase с режимом Слежения.	Ось	Если используется CAM, то следует убедиться, что Follower Mode (режим слежения) не сконфигурирован (режим слежения не может быть использован, если используется CAM). Если используется Follower Mode (режим слежения), следует убедиться, что CAM-команды отсутствуют в программе движения (CAM не может быть использован, если сконфигурирован режим слежения).
<b>Коды ошибок выполнения CAM</b>				
66	Нормальный останов	CAM-профиль не обнаружен в блоке загрузки CAM	Ось	CAM-профиль не был связан с блоком загрузки CAM в редакторе CAM и / или имя блока загрузки CAM не было задано в конфигурации аппаратного обеспечения.
67	Нормальный останов	Траектория выхода CAM вне диапазона (не циклического CAM)	Ось	Расстояние выхода для нециклического CAM было больше, чем модуль для CAM.
68	Только статус	(Коррекция разрешена) Заданная скорость ограничена из-за нарушения границы скорости или нарушения предела ошибки по положению	Ось	
68	Нормальный останов	(Коррекция запрещена) Заданная скорость CAM превышает границу скорости сконфигурированной оси	Ось	
6A	Нормальный останов	Нарушение предела ошибки CAM по положению (с запрещенной коррекцией)	Ось	
6B	Только статус	Заданная CAM-позиция при выходе отличается от значения CAM-профиля из-за ошибки по положению или ограничения скорости	Ось	
6C	Нормальный останов	Значение ведущей оси CAM находится за пределами диапазона профиля ведущей оси для нециклического профиля (команды CAM и CAM-LOAD)	Ось	
6D	Нормальный останов	Абсолютный режим CAM последовательно с инкрементальным режимом CAM	Ось	
6F	Быстрый останов	Ошибка расчета траектории CAM	Ось	Обратитесь в GE Fanuc Automation

## Раздел 3: Основы программирования электронного САМ

Данный раздел представляет собой введение в базовые концепции программирования электронного САМ. Функции Local Logic и программирование движения не обсуждаются детально в данном разделе, поскольку они описаны в других главах данного руководства.

### Требования

Редакторы программ Local Logic, САМ и движения являются составной частью программной среды. Для работы с ними требуется один из указанных ниже пакетов программ. Инструкции по установке находятся в документации по ПО.

Модуль DSM314 требует один из следующих пакетов программного обеспечения (ПО) конфигурации и программирования:

- Пакет SIMPLICITY Machine Edition Logic Developer PLC, версия 2.1 или более поздняя.
- Пакет VersaPro, версия 1.1 или более поздняя.

Набор функциональных возможностей модуля DSM314 требует наличия ЦП-микропрограммы 90-30, версия 10.0 или более поздняя. Кроме того, функция САМ требует наличия на модуле DSM314 микропрограммы версии 2.0 или более поздней.

## Введение в программирование электронного САМ

Функция электронного САМ работает вместе с программой движения модуля DSM314, программой Local Logic DSM314 и программной средой ПЛК, обеспечивая гибкое программное окружение. В частности, функция электронного САМ позволяет пользователю задавать прецизионную поточечную связь между ведущей и ведомой осями. Эта особенность является критичной для многих применений, где чрезвычайно жесткая синхронизация между осями является абсолютно необходимой.

Электронная САМ-функция модуля DSM позволяет пользователю задавать поточечные связи, используя встроенные программные инструменты. Средства редактора САМ дают возможность пользователю ввести эти соотношения графически, в табличной форме или в их комбинации. Эти САМ-профили затем сохраняются в модуле DSM, где к ним можно получить доступ через программы движения DSM. Базовые концепции САМ являются простыми и наилучшим образом иллюстрируются простым примером.

### Пример создания САМ

#### Базовые этапы

1. Открывание папки проекта или создание новой
2. Создание блока САМ
3. Создание профиля САМ
4. Связывание профиля САМ с блоком САМ
5. Конфигурирование профиля САМ
6. Задание типа САМ
7. Задание характеристики коррекции
8. Сохранение на диске профиля САМ
9. Создание программы движения и Local Logic
10. Проведение конфигурации аппаратного обеспечения в программе конфигурации / программирования
11. Выполнение (тестирование) задачи

## VersaPro

### Этап 1: создание новой папки

Работа начинается с запуска VersaPro. Эта процедура описана в главе 15 данного руководства, во встроенной системе оперативной консультативной информации по VersaPro и в руководстве по VersaPro, GFK-1670. После того как программа VersaPro запущена, следует создать новую папку; для этого откройте **File** меню (**Файл**) и выберите подменю **New Folder (Новая папка)**. Появится диалоговое окно, которое позволяет создать новую папку. В данном примере имя папки "CAMExample", как показано на рис. 16-1.

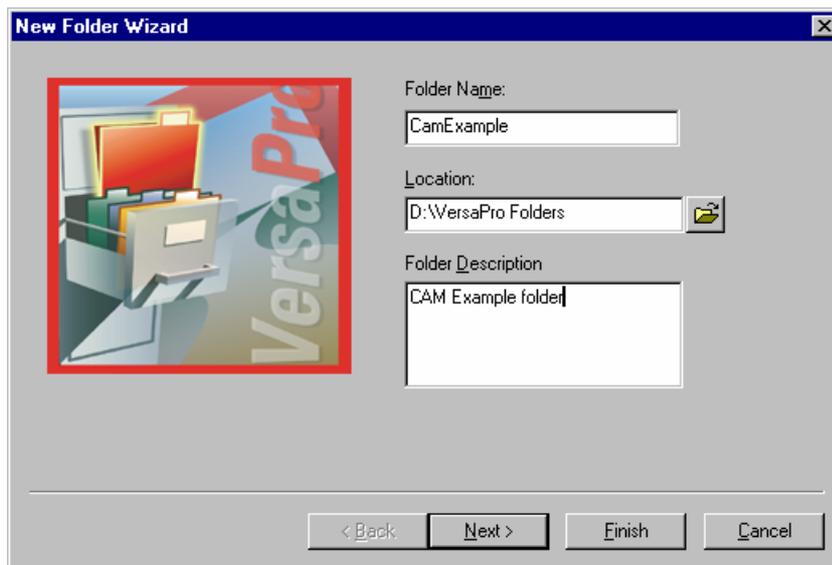


Рис. 16-12. Мастер по созданию новой папки, стр. 1

Щелчок по кнопке "Next" ("Далее") вызывает вопрос VersaPro, требуется ли создать каталог (если он не существует). После того как папка создана, появляется следующее диалоговое окно, которое предлагает на выбор несколько источников для новой папки. Настоящий пример начинается с пустой папки, которая является папкой по умолчанию (рис. 16-2).

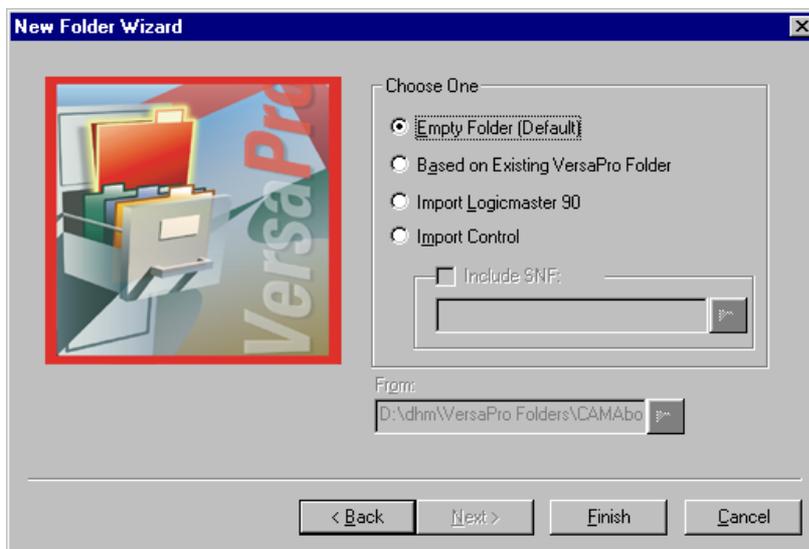


Рис. 16-13. Мастер по созданию новой папки, стр. 2

После щелчка по кнопке Finish (Завершить) программа VersaPro создает папку и запускает основную программу.

## Этап 2: создание блока САМ с использованием редактора САМ

Редактор САМ является частью пакета программ VersaPro. Этот редактор позволяет легко создавать, редактировать, сохранять и загружать блоки САМ. Чтобы создать блок САМ, следует открыть или создать новую папку VersaPro (см. этап 1). Информацию о создании или открывании папки см. в главе 15. Когда папка VersaPro открыта, выберите в меню команду File (Файл), затем New Program (Новая программа), после чего САМ Program... (Программа САМ...). (Рис. 16-1).

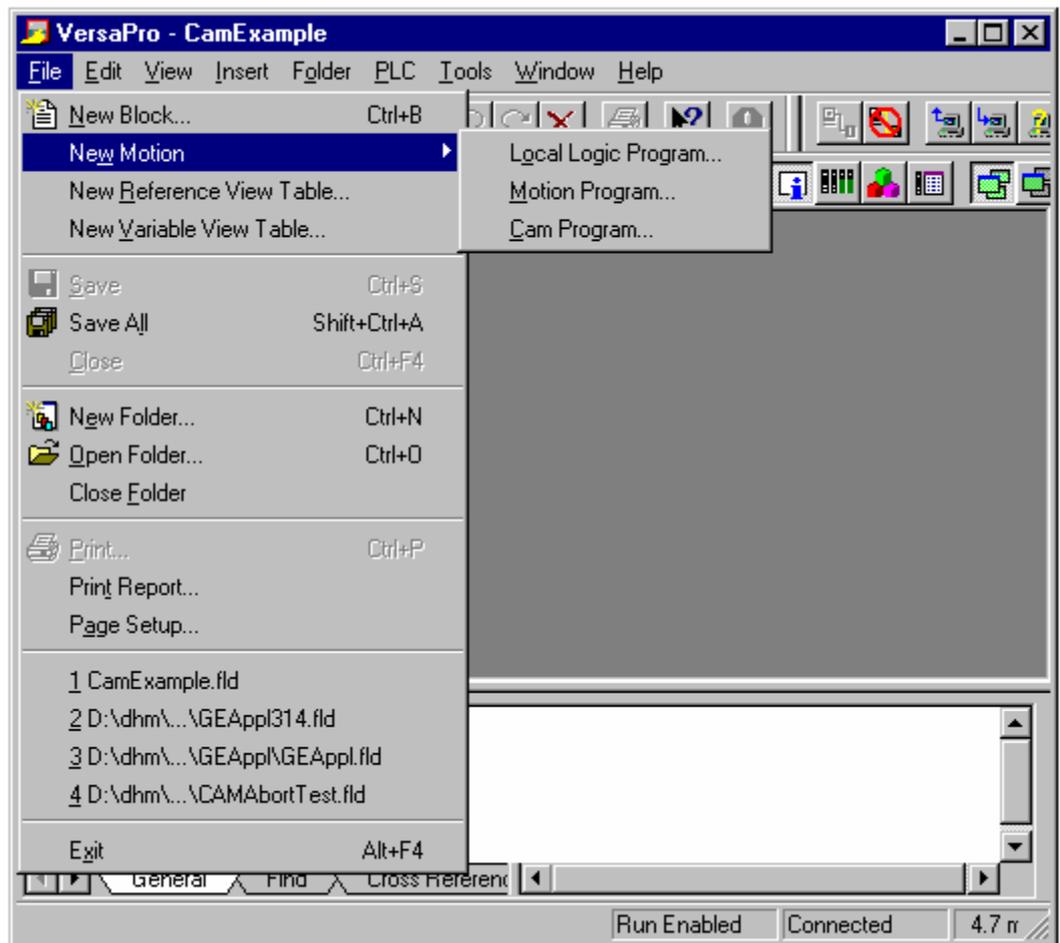


Рис. 16-14. Создание программы САМ

Появится диалоговое окно "Create New Program" ("Создать новую программу"). Дайте имя блоку САМ; дополнительно можно ввести описательный комментарий. В настоящее время функция САМ поддерживается только у модуля DSM314 (версия 2.0 или более поздняя). Поэтому не следует изменять выбор Motion Module Type (Тип модуля движения), предлагаемый по умолчанию. (Рис. 16-2). Следующие правила для имен САМ-блоков:

- Разрешается использование только следующие символы: A-Z, a-z, 0-9 и \_ (символ подчеркивания). Последовательно расположенные подчеркивание не разрешаются.
- Имя блока должно начинаться с буквы или символа подчеркивания.
- Два блока в одной открытой папке не могут иметь одно и то же имя.
- Имя блока САМ может иметь до двадцати символов.

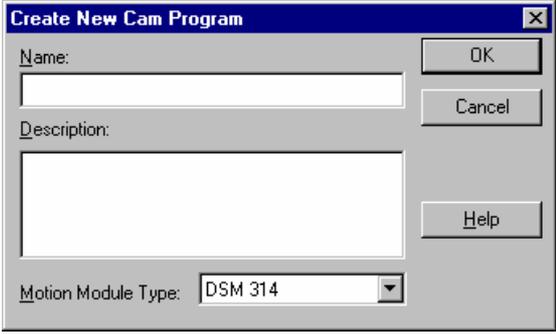


Рис. 16-15. Создание новой программы CAM

Введите данные в окно "Create New CAM Program" ("Создание новой программы CAM"), затем щелкните кнопку ОК, чтобы был создан блок CAM. В результате программа VersaPro запустит редактор CAM. (Рис. 16-3).

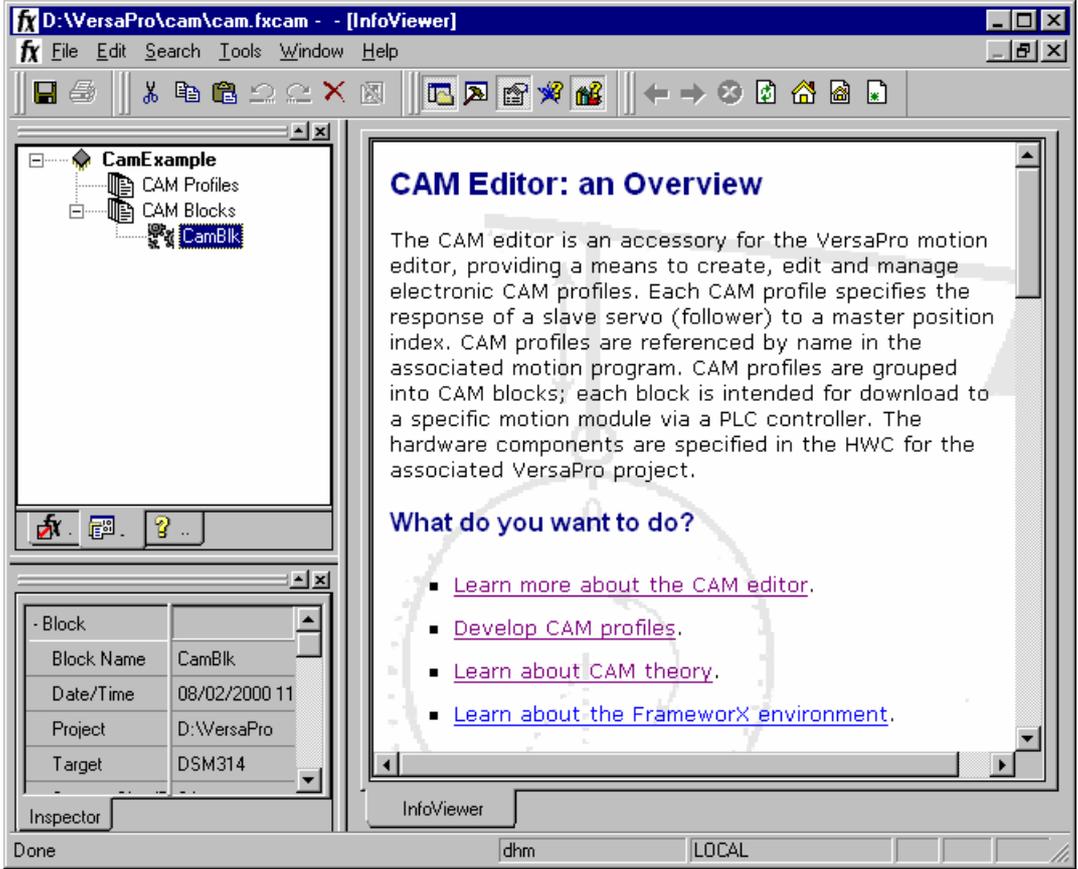


Рис. 16-16. Исходное окно редактора CAM с окном просмотра информации

Редактор CAM содержит обширную гипертекстовую интерактивную справочную информацию поддержки. В данном руководстве делается лишь попытка дать вводную информацию по некоторым из аспектов. Настоящее руководство не пытается описать все функции редактора; поэтому настоятельно рекомендуется просмотреть справочный материал по программированию.

**Этап 3: Создание профиля CAM**

Следующим этапом является создание простого профиля CAM в редакторе CAM. Редактор CAM содержит библиотеку CAM-профилей, которая создается пользователем. Профили CAM из библиотеки связаны с блоками CAM. Дополнительная информация по осуществлению такой связи содержится во встроенной системе оперативной консультативной информации (онлайн-справка). Для данного примера следует сначала

создать профиль в библиотеке. Одним из методов осуществления этого является щелчок правой кнопкой значка "CAM Profiles" ("Профили САМ") в окне Навигатора. Появляется локальное меню. Выберите пункт New Profile (Новый профиль), как показано на рис. 16-1.

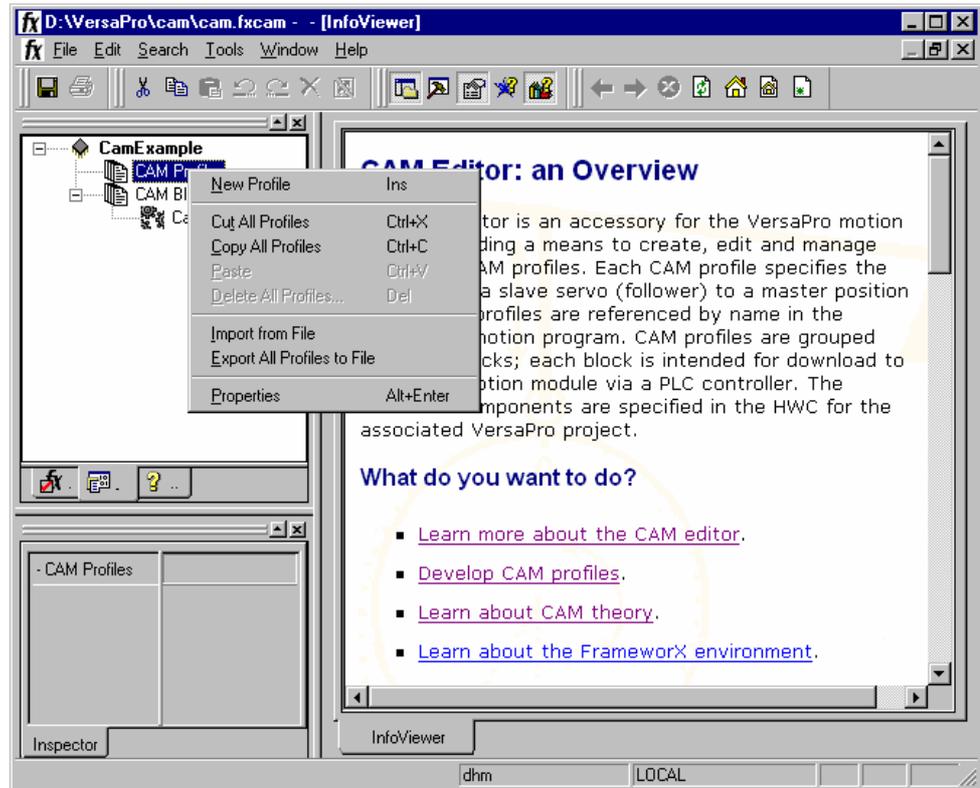


Рис. 16-17. Создание нового профиля САМ

В библиотеку будет введен новый профиль под именем "profile1", как показано на рис. 16-2.

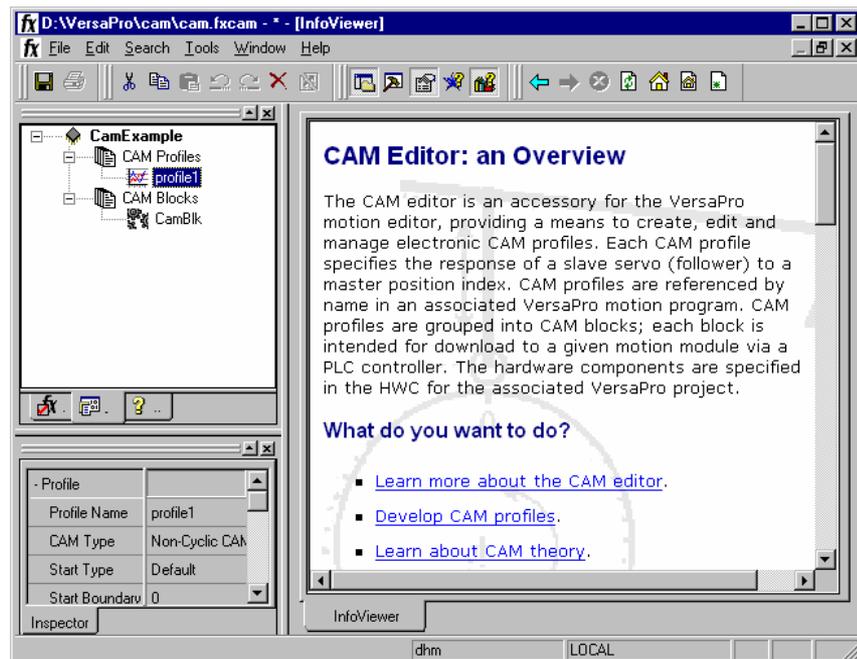


Рис. 16-18. Создание нового профиля

Позднее, если требуется, можно дать этому профилю имя, более подходящее для конкретного применения. Для имен действуют следующие правила:

- Может быть использован любой буквенно-цифровой символ или символ подчеркивания (\_).
- Первым символом в имени профиля должна быть буква.
- Имя профиля должно содержать не более 20 символов.
- Ссылка на профиль осуществляется по имени в программе движения VersaPro.  
ПРИМЕЧАНИЕ: VersaPro не восприимчив к регистру букв при ссылке на имя профиля.

Чтобы переименовать профиль, можно щелкнуть правой кнопкой имя профиля в окне Навигатора и выбрать в локальном меню пункт Rename Profile (Переименовать профиль) (рис. 16-3). Введите имя профиля и нажмите ENTER для завершения операции. Произойдет переименование профиля и всех, связанных с профилем CAM-связей. В данном случае новым именем профиля является ExCamProfile. Дополнительную информацию см. в справке.

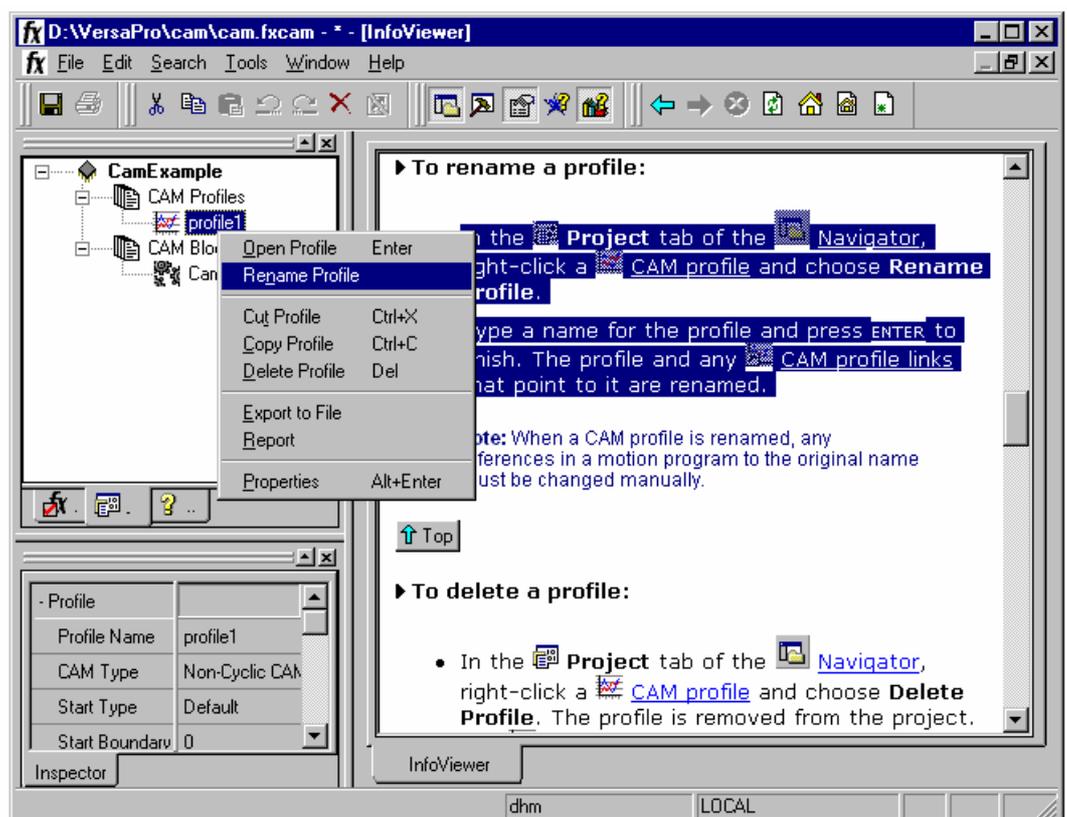


Рис. 16-19. Переименование профиля

#### Этап 4: Связывание профиля CAM с блоком CAM

- CAM-профили должны быть связаны с соответствующим блоком CAM. CAM-блок может содержать различные CAM-профили. Модуль DSM имеет два ограничения, которые влияют на количество профилей. Такими ограничениями являются максимальный размер блока CAM, равный 50К, и максимальное число профилей, которое может быть связано с отдельным блоком, равное 100. Библиотека CAM-профилей ограничена только имеющимся дисковым пространством на основном компьютере. CAM-блок затем связывается с модулем DSM через вход CAM-блока в конфигурации аппаратного обеспечения.

Хотя имеется несколько путей связи САМ-профиля с САМ-блоком, но наиболее простым методом является просто щелкнуть требуемый САМ-профиль, затем перенести и опустить его на соответствующий САМ-блок. Результат показан на следующем рисунке.

### Примечание

VersaPro ограничивает полный размер загружаемого блока (движение, Local Logic и САМ – все вместе) до 32 К.

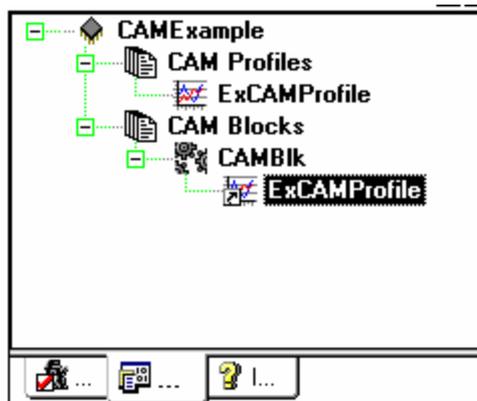


Рис. 16-20. Связывание профиля с САМ-блоком

### Этап 5: конфигурация точек данных профиля САМ

После завершения указанных выше операций необходимо сконфигурировать профиль САМ. Этот профиль представляет собой соотношение между позициями ведущей и ведомой осей. САМ-профиль состоит из последовательности точек. Каждая точка определяется двумя координатами. Если рассматривать графическое представление, то ведущая координата откладывается по горизонтальной оси, а ведомая – по вертикальной, как показано на рисунке ниже.

Двойной щелчок по профилю открывает его в окне редактора профиля (см. рис. ниже), которое состоит из двух редакторов:

- **Табличный редактор** аналогичен электронной таблице. В этой таблице каждая точка имеет свою строку с двумя колонками, одна из которых предназначена для ведущей позиции, а другая – для соответствующей ведомой позиции. Когда открывается новый профиль, то в нем по умолчанию имеются только две точки – начальная и конечная. Начальная точка является самой верхней точкой таблицы, а конечная – самой нижней.
- **Редактирование точек в графическом редакторе** состоит в том, что требуется щелкнуть точку на графике и перетащить ее в требуемое положение. (ПРИМЕЧАНИЕ: Данные, относящиеся к точке, будут при этом в табличном редакторе изменены так, чтобы соответствовать новой позиции). Чтобы выполнить другие задачи в графическом редакторе, надо щелкнуть правой кнопкой по графику и выбрать соответствующий пункт в локальном меню.

Следующим шагом является редактирование конечной точки (самая нижняя точка в таблице) для ведущей и ведомой осей. В табличном редакторе щелкните колонку ведущей оси конечной точки и введите значение 50000; затем щелкните колонку ведомой оси конечной точки и введите значение 0. (ПРИМЕЧАНИЕ: Когда происходит добавление или изменение точек в табличном редакторе, изменяется, соответственно, график в графическом редакторе).

Вставка дополнительной точки в таблицу редактора. Щелкните правой кнопкой по колонке ведущей оси конечной точки и выберите пункт Insert Point (Вставить точку) в

локальном меню (показано на рисунке далее). Над строкой конечной точки будет вставлена новая строка, определяющая новую точку, со значениями ведущей и ведомой осей, которые представляют собой по умолчанию среднее значение (в данном случае, 25000 и 0, соответственно) двух существующих соседних точек (сверху и снизу). Измените значения для этой точки на 47500 для ведущей оси и на 11000 - для ведомой. Чтобы изменить значение в точке, щелкните его, введите новое число, а затем нажмите клавишу Enter или щелкните за пределами таблицы.

Чтобы изменить порядок интерполирующей кривой, щелкните колонку Curve-Fit (Интерполирующая кривая), затем выберите Curve-Fit Order (Порядок интерполирующей кривой) в окне Property Inspector (Инспектор свойств). Профиль может быть также разделен на несколько секций или несколько секций могут быть объединены в одну; для этого надо щелкнуть изображение Curve-Fit (Интерполирующая кривая) правой кнопкой и выбрать из локального меню соответствующий пункт.

### Примечание

САМ-профиль ограничен 400 точками, если он содержит секторы второго или третьего порядка. САМ-профиль ограничен 5000 точками, если он содержит секторы только первого порядка.

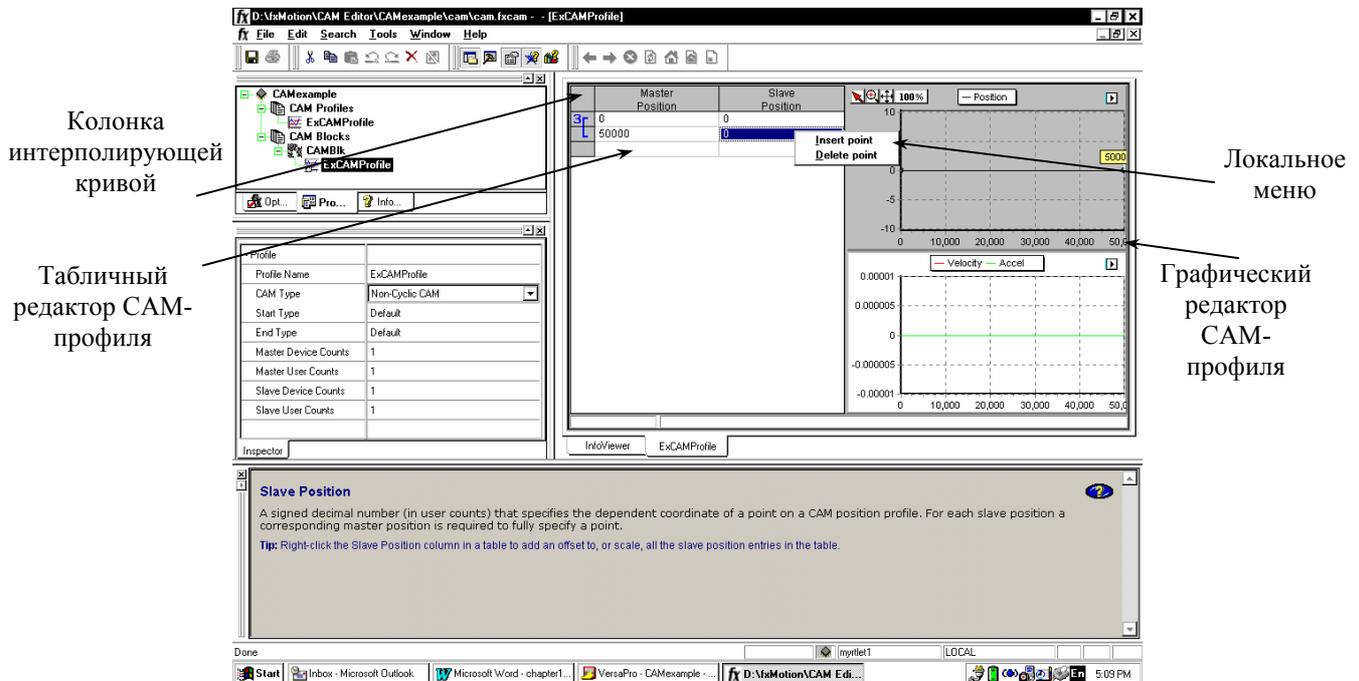


Рис. 16-21. Вставка точки в окне редактора профиля

Поскольку конечная точка позиционирования ведомой оси имеет то же значение (0), что и начальная ее точка, то этот САМ удовлетворяет требованиям линейного циклического САМ. (Если требуется, см. в разделе 2 более детальную информацию по различным типам САМ). Обратите внимание, что редактор САМ имеет несколько "интеллектуальных" полей редактирования, которые отображают только те возможности выбора, которые имеют силу для имеющегося набора данных. Например, поскольку требованием для линейного циклического САМ является равенство начальной и конечной точек позиционирования ведомой оси, то редактор позволяет выбрать Linear Cyclic CAM (Линейный циклический САМ) только в случае выполнения этого требования.

Далее следует вставить новую точку в профиль и отредактировать эту точку. Данная точка может быть отредактирована либо в таблице профиля, либо графически на изображении. Вставьте точку, как показано выше и на рис. 16-2, 16-3, щелкнув правой

клавишей по точке, находящейся под позицией вставки, и выбрав в меню пункт Insert Point (Вставить точку). Затем измените значение по умолчанию на 25000 для ведущей оси и на 10000 для ведомой.

	Master Position	Slave Position
3r	0	0
	25000	10000
	47500	11000
	50000	0

Рис. 16-22. Данные таблицы САМ-профиля

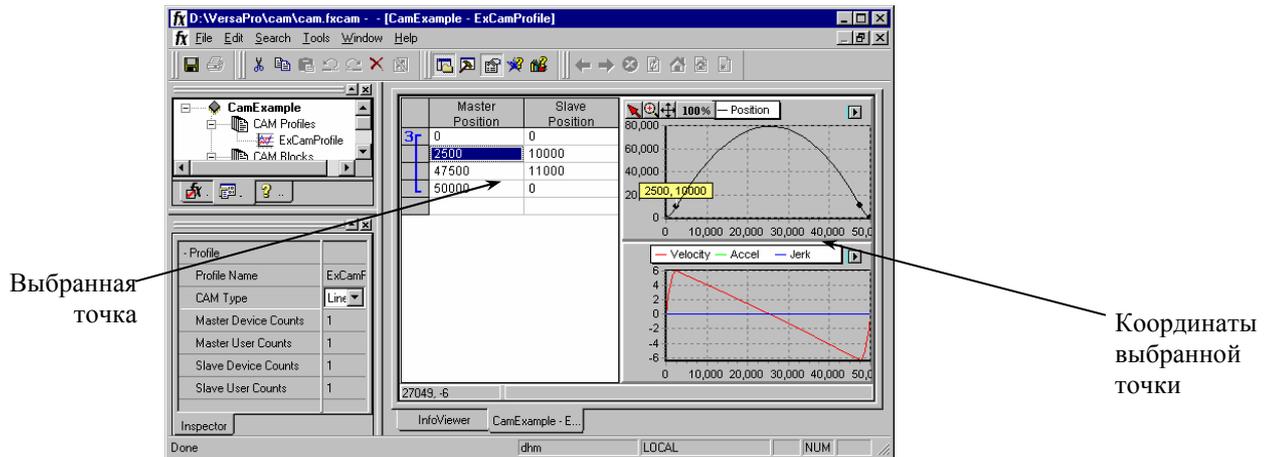


Рис. 16-23. Пример редактора САМ

Редактор имеет много других функций. В частности, он позволяет задать дополнительные секторы с различающимися методами интерполяции. Эти функции редактора обсуждаются во встроенной системе оперативной консультативной информации по программированию. См. дополнительную информацию в этой справке.

### Этап 6: задание типа САМ

В данном примере САМ является линейным циклическим, как рассматривалось выше. Используется следующая процедура:

- В закладке Project (Проект) Навигатора щелкните правой клавишей CAM profile (Профиль САМ). Появляется локальное меню.
- Выберите в локальном меню пункт Properties (Свойства). Откроется окно Inspector (Инспектор), показывая свойства САМ-профиля.
- В Инспекторе щелкните стрелку в поле CAM Type (Тип САМ). Появится выпадающий список типов САМ.
- Выберите из списка "Linear Cyclic CAM" (рис. 16-1).

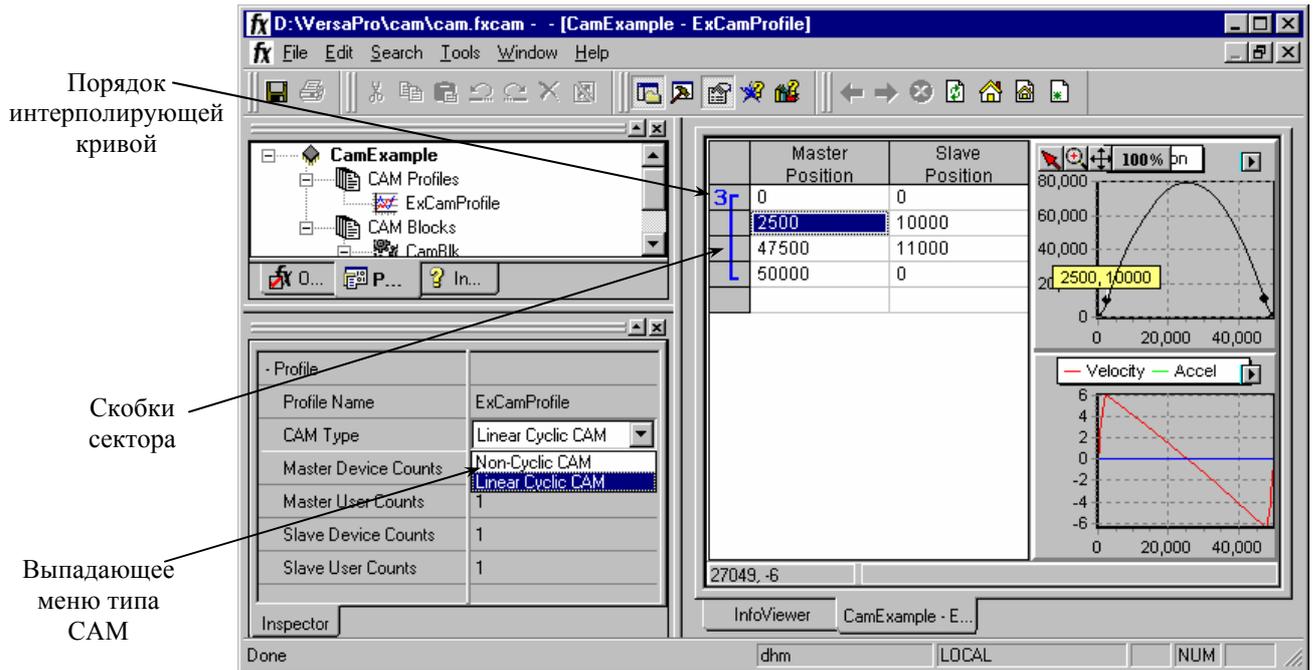


Рис. 16-24. Выбор типа CAM в CAM-редакторе

### Этап 7: Задание характеристики коррекции

Последней позицией, которую требуется задать в данном примере, является статус коррекции. Характеристика коррекции определяет, разрешает ли модуль движения оперативную коррекцию конкретного сектора. Сектором является участок САМ-профиля, определенный, по крайней мере, двумя соседними заданными пользователем точками. Сектор содержит точки, заданные пользователем, и кривую, соединяющую их и также не входящую в данный сектор первую точку, заданную для следующего смежного сектора. Точки, включенные в данный сектор, заключены в скобки, показанные на рисунке выше. Каждому сектору присвоено число, обозначающее порядок интерполирующей кривой; число также показано на рисунке выше. Сегменты профиля между точками, определенными пользователем, задаются полиномами заданного порядка. Интерполяция между каждой парой смежных точек, определенных пользователем, выполняется при помощи уникальных полиномов. Хотя действующие полиномиальные коэффициенты могут быть различными для каждого сегмента, порядок кривой сохраняется для всего сектора. Сектор отображается в таблице САМ-профиля как столбик, стягивающий значения позиции ведущей оси, включенные в этот сектор. Первоначально все точки, заданные в профиле, включены в единый сектор. Этот единый начальный сектор может быть поделен требуемым образом, чтобы облегчить сглаживание САМ-профиля. Когда атрибут Correction (Коррекция) установлен на Enabled (Разрешена), то модуль движения выдает предупреждение, если происходит нарушение ограничения скорости. Когда атрибут Correction (Коррекция) установлен на Disabled (Запрещена), то при таких нарушениях модуль движения выдает сообщение об ошибке и останавливает ведомую ось.

Для этого примера коррекция должна быть разрешена. Чтобы разрешить коррекцию, выберите сектор в таблице профиля САМ, щелкнув ее. Появится окно Инспектора, отображающее свойства сектора и позволяющее провести его редактирование. Выберите окно Correction (Коррекция) с выпадающим меню и выберите в этом меню Enabled (Разрешена) (рис. 16-1).

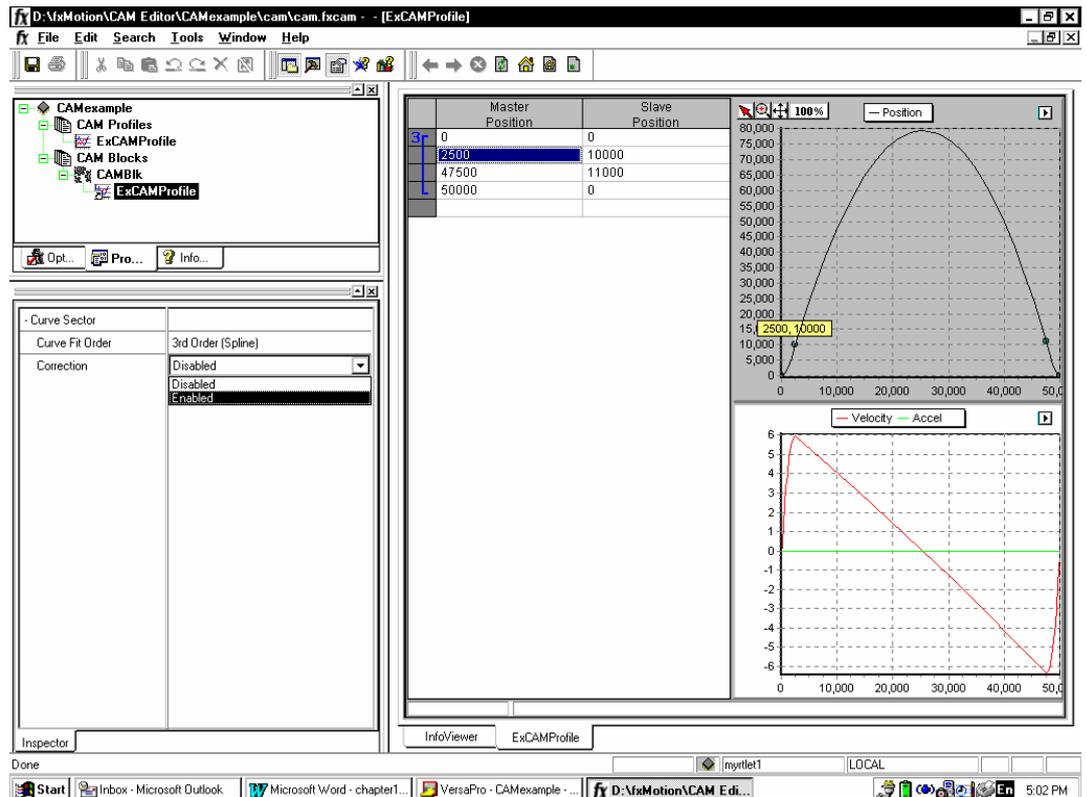


Рис. 16-25. Разрешение коррекции в редакторе САМ

### Этап 8: сохранение на диске профиля САМ

К этому моменту задание простого САМ-профиля сделано. Чтобы сохранить на диске блоки / профили САМ, выберите пункт меню File (Файл), а затем выберите в подменю пункт Save Project (Сохранить проект). Можно также выбрать Exit (Выход), что приведет к автоматическому сохранению. САМ-редактор имеет множество дополнительных особенностей и функций. Детальное описание этих свойств см. справочной документации.

### Этап 9: создание программ движения и Local Logic

Следующими позициями, которые должны быть созданы, являются программы движения и Local Logic, которые будут работать с этим САМ-профилем. В настоящем примере логика должна работать с модулем DSM314, управляющим двумя осями. Ось #1 является ведомой, а ось #2 – ведущей. Поэтому будут две программы движения. Программа оси 1 (для ведомой оси) должна выполнить некоторую базовую инициализацию, загрузить начальную ведомую точку для данного САМ-профиля, а затем выполнить САМ-команду. Программа оси 2 (для ведущего источника) является простой программой, которая должна провести инициализацию, а затем ждать готовности ведомой программы. Она должна выполнить последовательность движений. Программа должна выполнить остановку в точках, заданных в ведущей САМ-программе так, чтобы можно было легко проверить, что ведомая ось правильно реализует САМ-профиль. Данный пример требует наличия программы Local Logic. В этом примере программа Local Logic контролирует работу САМ-программ движения ведомой и ведущей осей. Таким образом, Local Logic синхронизирует эти две программы.

Дополнительную информацию по этим функциям см. в соответствующих главах данного руководства. Программа движения и программы Local Logic для данного примера показаны далее:

```

//      Программа движения для примера САМ-блока

//      Ведомая ось
Program 1 AXIS1
VELOC 10000           //      Задание скорости
ACCEL 10000           //      Задание ускорения
100: WAIT CTL01       //      Ожидание LL, означающего
                       //      готовность ведущей оси
110: CAM-LOAD "ExCamProfile", P006, ABS //      Загрузка в регистр параметров
                       //      ведомой точки, которая соответствует
                       //      текущей позиции ведущей оси
120: PMOVE P006, ABS, LINEAR //      Перемещение ведомой оси в
                       //      позицию, которая соответствует началу
                       //      таблицы.
130: CAM "ExCamProfile", 50000, ABS //      Выполнение оператора САМ
140: PMOVE 0,ABS,S-CURVE //      Возврат к нулю
150:
      ENDPROG

//      Программа ведущей оси
Program 2 AXIS2
      VELOC 10000           //      Задание скорости
      ACCEL 10000           //      Задание ускорения
200: PMOVE 0 ,ABS,S-Curve //      Запуск при нуле
210: WAIT CTL08       //      Ведущая ось ждет, когда закончится
                       //      позиционирование ведомой оси
220: PMOVE 2500,ABS,LINEAR //      Перемещение ведущей оси к первой точке
                       //      таблицы
230: DWELL 5000        //      Ожидание 5 секунд
240: PMOVE 47500,ABS,LINEAR //      Перемещение ко второй точке
250: DWELL 5000        //      Ожидание 5 секунд
260: PMOVE 2500,INC,LINEAR //      Пройти до конца расстояние, заданное в
                       //      САМ-команде, 1-го комплекта САМ
270: PMOVE 0,ABS,LINEAR //      Возврат к нулю
280:
      ENDPROG

```

// **Программа Local Logic для примера САМ**

```

CTL01 := 0; // Выходы, записанные, когда логика
              завершила инициализировать на нуль, чтобы
              позволить переключение на логическую
              "единицу" ("истина").

CTL08 := 0; // Выходы, записанные, когда логика
              завершила инициализировать на нуль, чтобы
              позволить переключение на логическую
              "единицу" ("истина").

//      Управляющая логика для
//      программ 1 и 2
//      Программа 1 = ведомая ось
//      Программа 2 = ведущая ось
IF PROGRAM_ACTIVE_2=1 THEN // Убедитесь, что программа 2 активна
  IF BLOCK_2=210 THEN // Показывает, что ведущая ось готова к
                      // запуску САМ
    IF PROGRAM_ACTIVE_1=1 THEN // Проверка того, что программа на оси 1
                                // активна
      IF BLOCK_1=100 THEN // Блок показывает готовность ведомой
                          // оси к выполнению САМ-Load
        CTL01:=1; // Сигнал для ведомой оси начать
                  // выполнение САМ Load

      END_IF;
      IF BLOCK_1=130 THEN // Блок показывает, что ведомая ось
                          // завершила инициализацию и выполнение САМ-
                          // Load
        CTL08:=1; // Сигнал ведущей и ведомой осей, что
                  // обе оси готовы к запуску выполнения САМ

      END_IF;
    END_IF;
  END_IF;
END_IF;
//      Конец управляющей логики для программ 1 и 2

```

После завершения входа в программу результирующее окно VersaPro должно выглядеть примерно так, как показано на рисунке внизу (рис. 16-1).

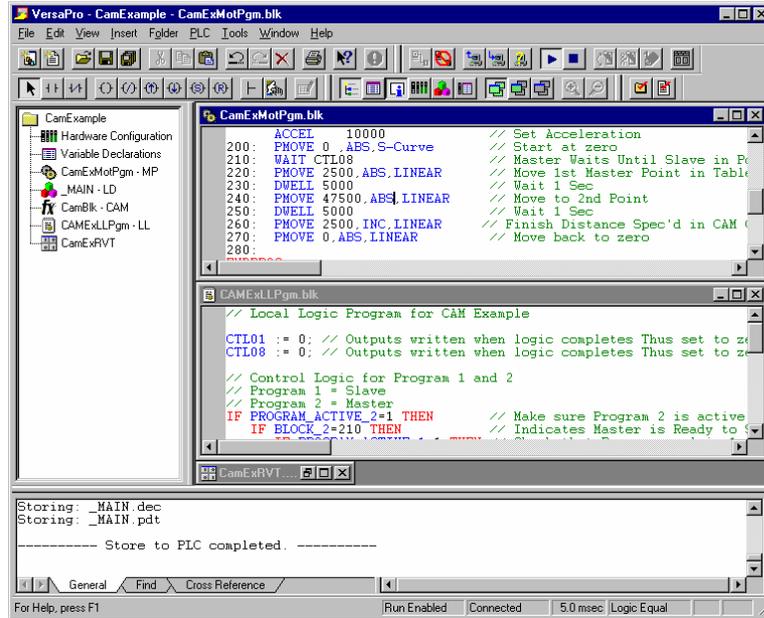


Рис. 16-26. Окна VersaPro в примере САМ-программы

## Этап 10: задание установочных параметров в конфигурации аппаратного обеспечения в VersaPro

Если проверка синтаксиса для локальной логической программы и программы движения прошла успешно, то требуется установить конфигурацию аппаратного обеспечения, что должно позволить загрузить программу примера в модуль DSM314. Последовательность действий в данном примере не является типичной для большинства процессов установки. Большинство пользователей сначала задает конфигурацию своего аппаратного обеспечения, а затем создает операторы программирования. Однако данный пример имеет своей целью проиллюстрировать функцию САМ. Поэтому последовательность действий в настоящем примере "перевернута", чтобы лучше показать связь между конфигурацией аппаратного обеспечения и именем САМ-блока в конфигурации аппаратного обеспечения модуля DSM314. Следующим шагом в данном примере является запуск конфигурации аппаратного обеспечения. Для этого имеется несколько возможностей. Здесь представлены два метода (дополнительную информацию см. в документации по VersaPro). Чтобы выполнить это, следует либо (1) выбрать команду View (Вид) в главном меню, а затем выбрать пункт Hardware Configuration (Конфигурация аппаратного обеспечения), либо (2) щелкнуть значок Конфигурации аппаратного обеспечения в строке инструментов. Это приведет к запуску инструментария конфигурации аппаратного обеспечения. Окна конфигурации аппаратного обеспечения по умолчанию предназначены для изделий VersaMax. Поэтому первым действием должен быть выбор Series 90-30. Чтобы сделать это, выберите в основном меню команду File (Файл), затем Convert To (Преобразовать В), а затем в подменю Convert To выберите Series 90-30, как показано на рис. 16-1.

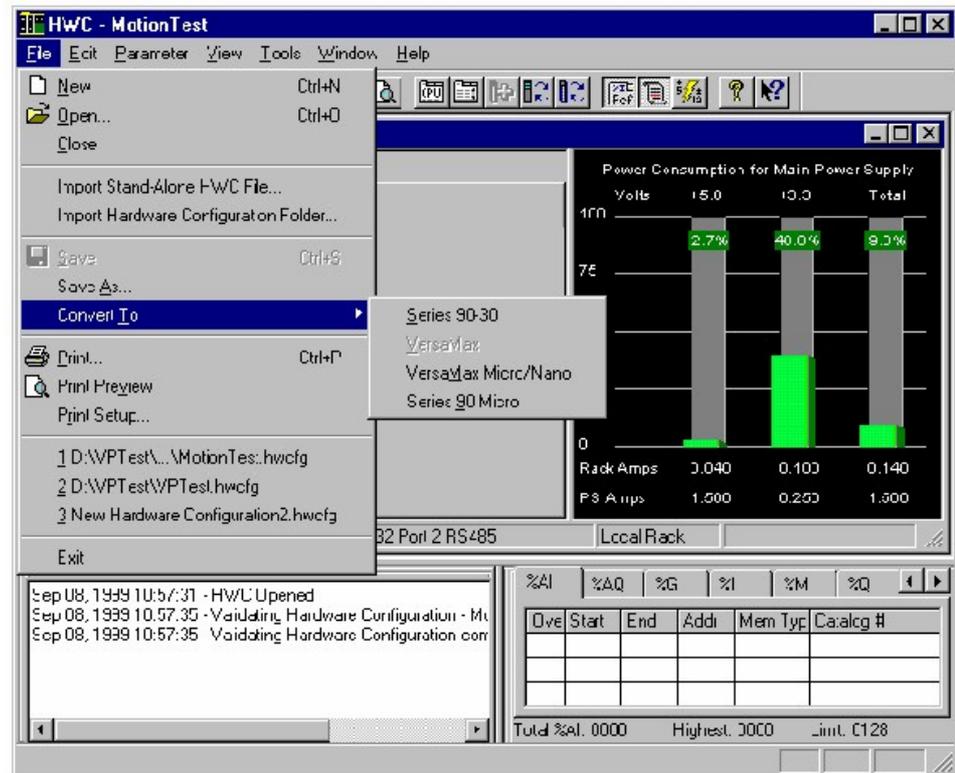


Рис. 16-27. Выбор крейта при конфигурации аппаратного обеспечения

Появляется диалоговое окно, которое предупреждает об удалении информации. Поскольку папка, созданная в данном примере, новая, то потери информации не произойдет. Если новая папка не создавалась, то помните, что при этой операции произойдет потеря информации о конфигурации. Рекомендуется использовать в данном примере новую пустую папку. Щелкните Yes (Да) в диалоговом окне, показанном на рис. 16-2.

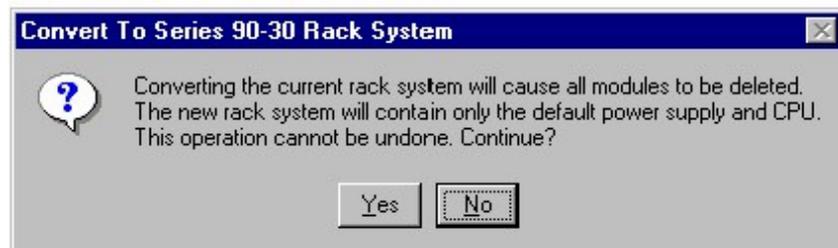


Рис. 16-28. Диалоговое окно преобразования крейта при конфигурации аппаратного обеспечения

После завершения этой процедуры получается не заполненный крейт 90-30, который необходимо сконфигурировать. Получающиеся окна конфигурации аппаратного обеспечения показаны на рис. 16-3.

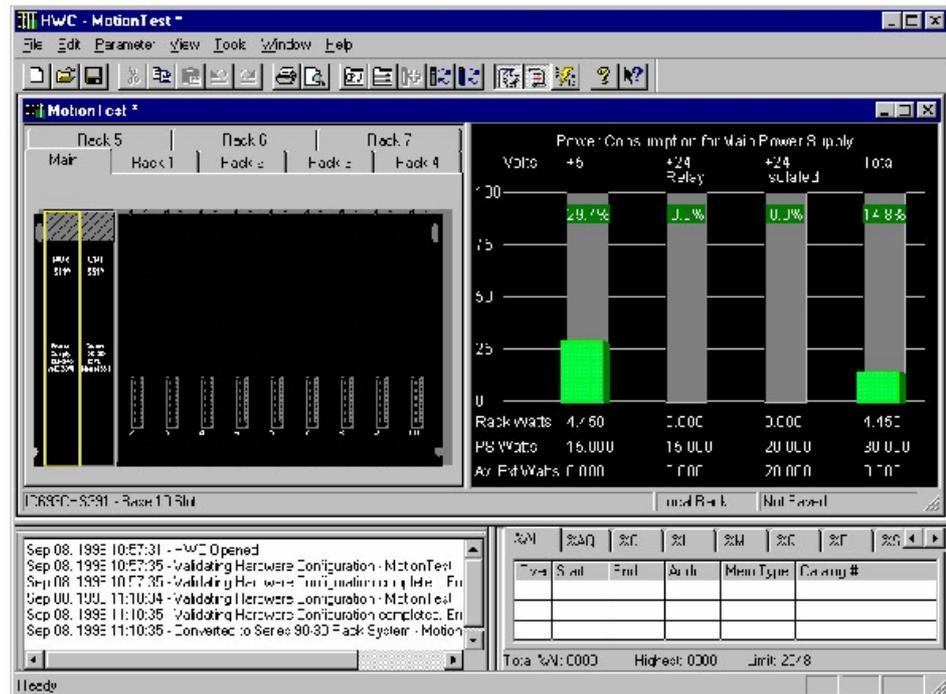


Рис. 16-29. Конфигурация крейта 90-30 с ЦП

Далее следует выбрать модуль питания и ЦП, соответствующие имеющемуся оборудованию. Обратите внимание, что Local Logic требует наличия у ЦП микропрограммы версии 10.00 или более поздней. ЦП по умолчанию – "CPU351" – не поддерживает микропрограмму версии 10.00, поэтому требуется заменить ЦП на модель, которая поддерживает версию 10.00, например, на модель CPU363. Для этого надо сделать следующее:

- Щелкните правой кнопкой CPU в окне конфигурации аппаратного обеспечения и выберите в локальном меню пункт Replace CPU (Заменить ЦП). Появится окно выбора модуля.
- Щелкните позицию IC693CPU363, затем щелкните кнопку ОК.
- Появятся два сообщения, предупреждающих о том, что (1) замена ЦП не может быть отменена и что (2) SNP и другие параметры уже перенесены и т.п. В обоих случаях щелкните Yes (Да) для продолжения.
- Появится окно конфигурации CPU363. Выполните все требуемые изменения конфигурации, после чего закройте окно конфигурации.

Если требуется, дополнительная информация может быть получена в документации по VersaPro или в справке. Появится окно, изображенное на рис. 16-4.

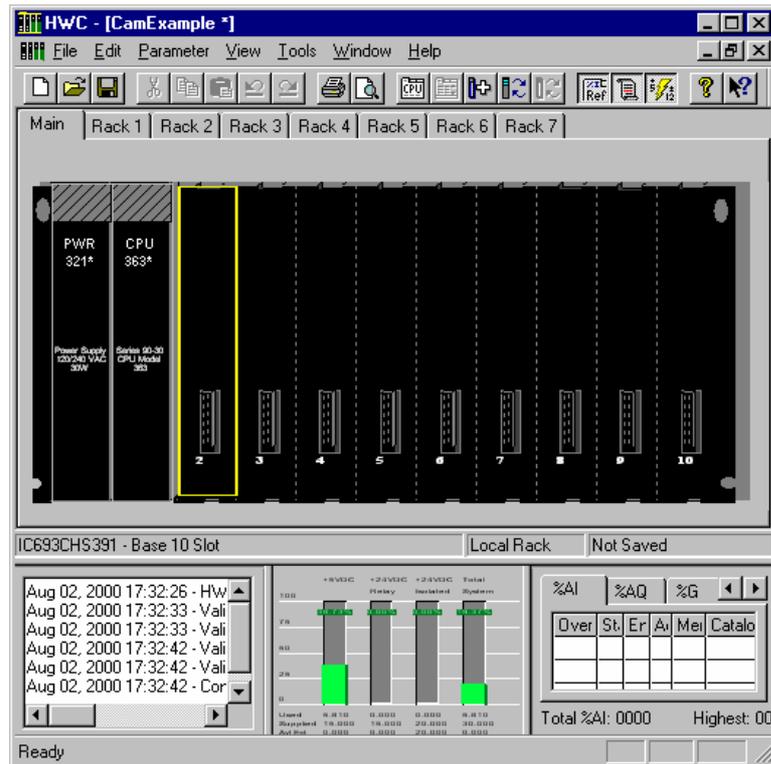


Рис. 16-30. Конфигурация крейта 90-30 с CPU363

На этом этапе требуется добавить модуль DSM314 в крейт. Чтобы выполнить этот шаг, выберите слот крейта, куда должен быть установлен модуль DSM314. В этом примере DSM314 устанавливается в слот номер 2. Имеется несколько возможностей добавить модули в слот крейта. Два метода добавления модуля представлены здесь (см. документацию VersaPro для дополнительной информации по процедурам).

- Можно (1) дважды щелкнуть требуемый слот (в данном случае это слот номер 2) или (2) щелкнуть правой клавишей требуемый слот (2), а затем выбрать пункт Add Module (Добавить модуль).
- При этом появляется диалоговое окно, которое позволяет выбрать тип модуля. Выберите в диалоговом окне закладку "Motion" ("Движение").
- Выберите в списке модуль DSM314. Результирующее окно показано на рис. 16-5.

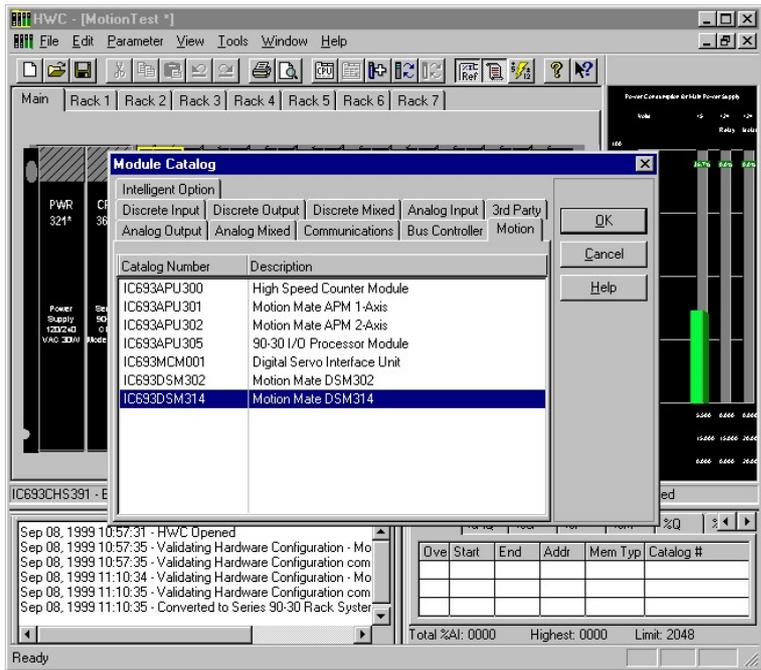


Рис. 16-31. Выбор DSM314 в ходе аппаратной конфигурации крейта 90-30

Эта операция добавит DSM314 в крейт и выведет окно конфигурации DSM314. С его помощью можно настроить DSM314 на конкретное применение. См. в гл. 4 подробности, касающиеся установок конфигурации DSM314. В данном примере подлежат изменению имена блока Local Logic, блока программы движения, блока CAM и режим Local Logic. Эти поля находятся под закладкой "Settings" ("Установочные параметры"). Local Logic Mode (Режим Local Logic) должен быть установлен на Enabled (Включен). В поле "Local Logic Block Name" ("Имя блока Local Logic") введите имя программы примера "CamExLLPgm". В поле "CAM Block name" ("Имя блока CAM") введите "CamBlk", а в поле "Motion Program Block name" введите "CamExMotPgm".

### Примечание

Этот метод связывания модуля DSM314 с блоками программ и CAM был выбран для того, чтобы позволить легко задавать разные модули DSM314, которые используют одни и те же блоки. Данный пример использует только один модуль DSM314. Если имеется несколько модулей DSM314, которым требуется использовать одну и ту же программу Local Logic, одни и те же программы движения или блоки CAM, укажите их в конфигурации для конкретного модуля DSM314, который будет выполнять данную программу. Это позволяет иметь один исходный файл для нескольких модулей DSM314. Следует отметить, что это не исключает выполнение модулями DSM314 различных программ. Таким образом, модуль DSM314 будет выполнять те файлы (программу CAM, Local Logic и движения), которые указаны при конфигурации. Получающееся окно конфигурации аппаратного обеспечения будет выглядеть так, как показано на рис. 16-6.

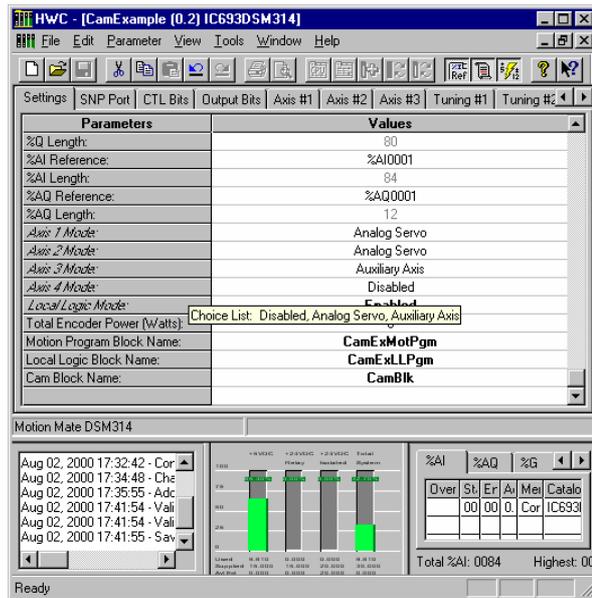


Рис. 16-32. Таблица установочных параметров модуля DSM314 при аппаратной конфигурации крейта 90-30

В данном примере программа Local Logic управляет битами CTL01 и CTL08. Поскольку биты CTL01 и CTL08 используются для передачи сигнала программам движения, то эти CTL-биты должны быть сконфигурированы под управление Local Logic. Чтобы сделать это, надо войти в закладку "CTL Bits" ("CTL-биты") при конфигурации аппаратного обеспечения в VersaPro. Выберите "CTL01 Config" ("Конфигурация CTL01") и задайте Local\_Logic\_Controlled. Повторите эту процедуру для CTL08. Получающиеся окна конфигурации аппаратного обеспечения показаны на рис. 16-7.

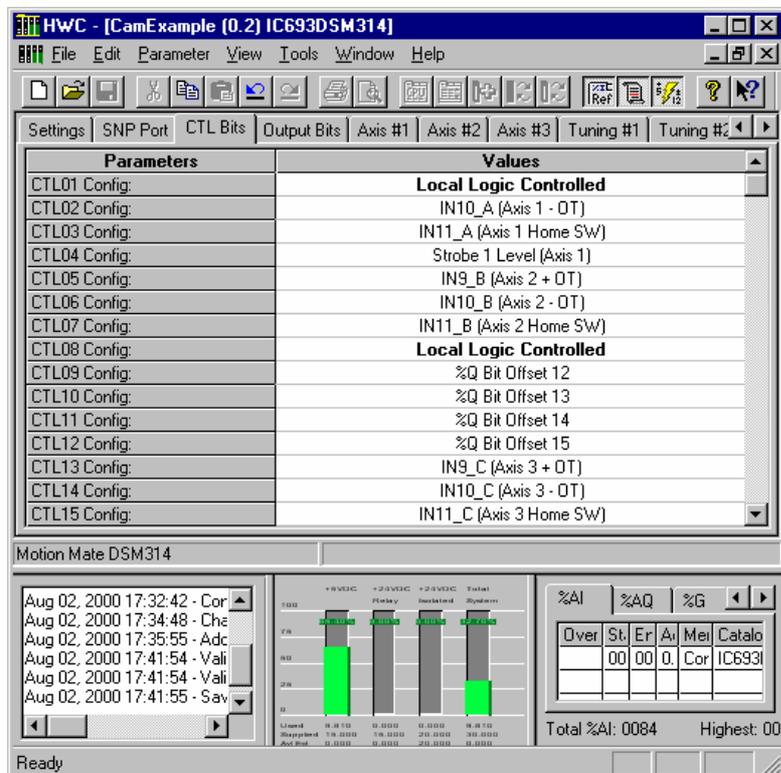


Рис. 16-33. Таблица CTL Bits (CTL-биты) модуля DSM314 при аппаратной конфигурации крейта 90-30

Поскольку данный пример использует цифровой сервопривод Beta 0.5, то режим оси 1 и оси 2 должен быть установлен на Digital Servo (Цифровой сервопривод).

Получающиеся окна конфигурации аппаратного обеспечения показаны на рис. 16-8.

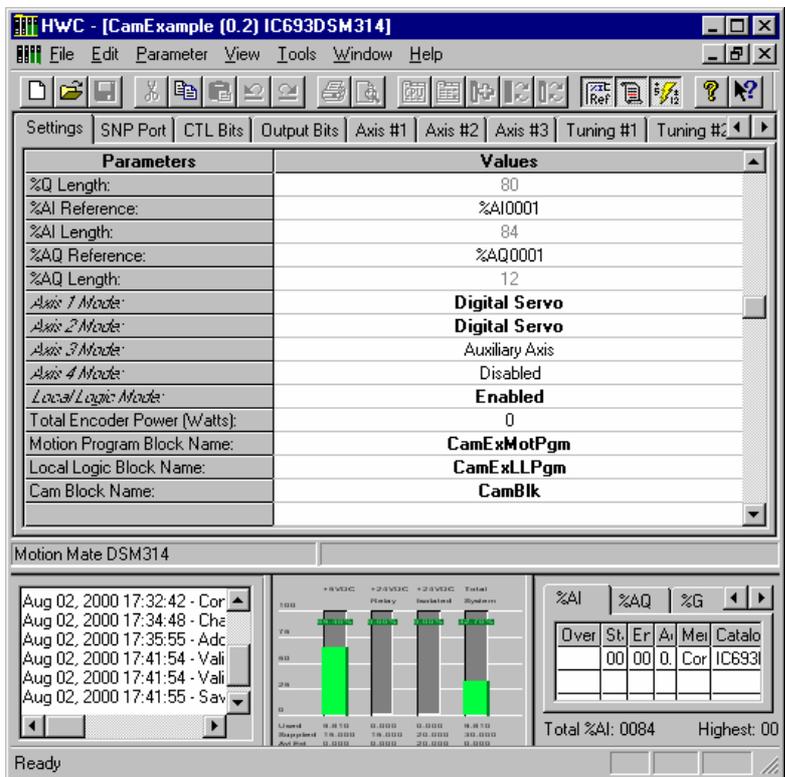


Рис. 16-34. Таблица установочных параметров модуля DSM314 при конфигурации аппаратного обеспечения

Необходимо также указать оси #1, что она будет использовать заданные позиции оси #2 в качестве ведущего источника САМ. Чтобы сделать это, надо войти в закладку "Axis #1" ("Ось #1") при конфигурации аппаратного обеспечения. Перейдите в поле ввода данных САМ Master Source (Ведущий источник САМ). В выпадающем списке выберите Cmd Position 2. Это задает оси #1 возможность использовать заданную позицию оси #2 в качестве ее ведущего источника САМ (рис. 16-9). Находясь на этой закладке, установите поле Home Mode (Режим позиционирования) на Move + (Движение +), а поле OverTravel Switch (Выключатель пере регулирования) – на Disabled (Отключен).

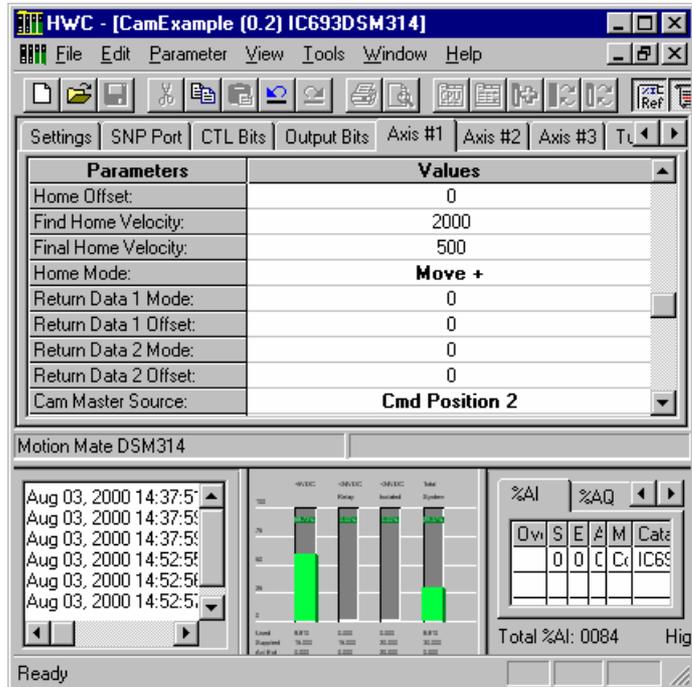


Рис. 16-35. Выбор ведущего источника для ведомой оси САМ

Требуется также указать оси #2 точки перехода для ссылки на позицию ведущей оси. Чтобы сделать это, надо войти в закладку "Axis #2" ("Ось #2") при конфигурации аппаратного обеспечения. Введите 49999 в поле High Position Limit (Верхняя граница позиции) и 0 - в поле Low Position Limit (Нижняя граница позиции) (рис. 16-10). Обратите внимание, что, поскольку здесь циклический САМ, то верхняя граница ведущего источника, по определению, должна быть на единицу меньше, чем последняя точка в ведущей таблице данных. В данном примере это точка 50000. Таким образом, верхняя граница равна 49999. Одним из путей представить эту ситуацию может быть изображение циклического САМ-ведущего в виде непрерывной круговой ленты, где первая точка ленты является одновременно ее последней точкой. Поэтому в данном примере 50000 это то же самое, что ноль. Находясь на этой закладке, установите поле Home Mode (Режим позиционирования) на Move + (Движение +), а поле OverTravel Switch (Включатель пере регулирования) – на Disabled (Отключен).

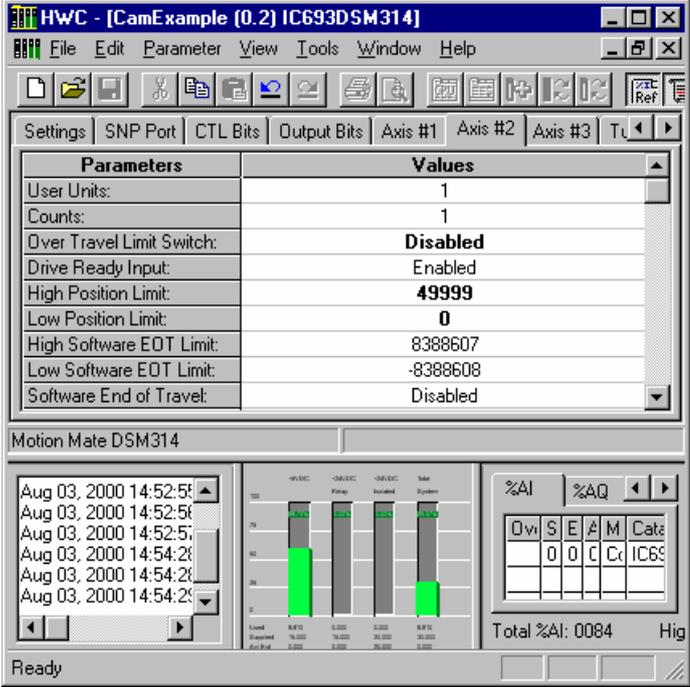


Рис. 16-36. Масштабирование ведущей оси САМ

Чтобы закончить конфигурацию, необходимо перейти к закладкам Tuning#1 (Настройка#1) и Tuning#2 (Настройка#2) и ввести приведенные ниже данные:

- **Motor Type (Тип двигателя):** 13
- **Position Error Limit (Предел Ошибки по Положению):** 200 (необязательный параметр; см. дополнительную информацию в разделе по конфигурации)
- **In Position Zone (Область позиционирования):** 20 (необязательный параметр; см. дополнительную информацию в разделе по конфигурации)
- **Pos Loop Time Const (Постоянная времени контура позиционирования):** 200 (Примечание: базируется на данных по применению и по механическим узлам, см. гл. 4 и Приложение D)
- **Velocity FeedForward (Коэффициент упреждения по скорости):** 9000 (Примечание: базируется на данных по применению и по механическим узлам, см. гл. 4 и Приложение D)
- **Vel Loop Gain (Коэффициент обратной связи по скорости):** 32 (Примечание: базируется на инерционных свойствах двигателя. Стандартная конструкция Beta Demo имеет закрепленное индикаторное колесо, которое создает почти весь момент инерции у Beta 0.5).

Результирующее окно будет похоже на то, что представлено на рис. 16-11.

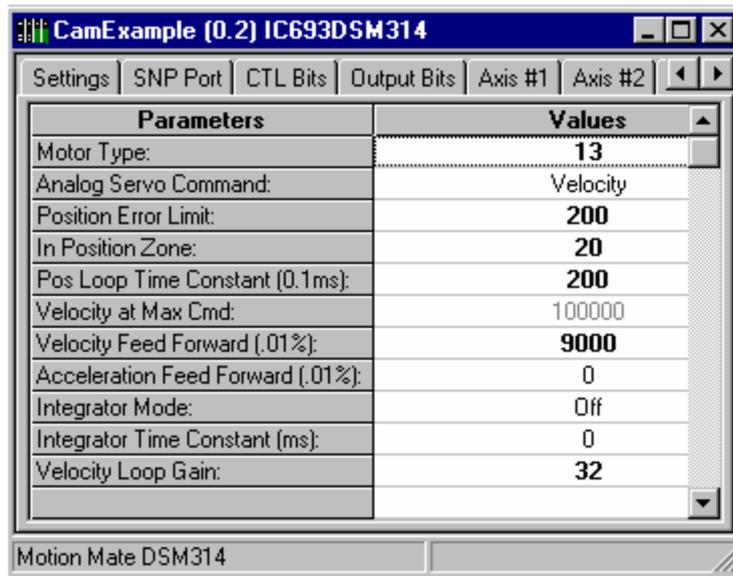


Рис. 16-37. Закладка Tuning#1 (Настройка#1) при конфигурации аппаратного обеспечения

Установки на закладке Tuning (Настройка) для Axis #2 должны быть заданы так же, как для оси #1 на рис. 2-26.

На этом этапе можно закрыть диалоговое окно конфигурации модуля и сохранить результаты работы на диске. Для этого выберите команду **F**ile (Файл) из главного меню, а затем в выпавшем меню выберите пункт **S**ave **A**ll (Сохранить все).

Это завершает изменения конфигурации, требуемые в данном примере. Закройте инструментарий конфигурации аппаратного обеспечения и сохраните папку, выбрав **F**ile (Файл) в главном меню, а затем, выбрав пункт **S**ave **A**ll (Сохранить все).

Связь между примером блока САМ, программы движения, программы Local Logic и модулем DSM314 теперь установлена. Теперь можно выполнить любое требуемое многоступенчатое логическое программирование ПЛК, затем провести Check All (Проверить все) на этих программах, после чего загрузить их в ПЛК. Дополнительная информация, касающаяся операции загрузки, имеется в руководстве GFK-1670 по VersaPro или в справке по VersaPro 1.5.

### Этап 11: выполнение (тестирование) вашей программы движения на базе CAM

#### Предупреждение

Прежде чем приступить к тестированию вашей прикладной программы на реальном оборудовании, следует сначала убедиться, что при тестировании обеспечена безопасность. Это подразумевает то, что, все устройства установлены в соответствии с правилами техники безопасности, что все приборы безопасности смонтированы и работают и что персонал в рабочей зоне предупрежден. Пренебрежение вопросами, связанными с безопасностью, может привести к травмированию персонала и к повреждению оборудования.

После завершения операции загрузки модуль готов выполнить блоки CAM, программы движения и локальную логическую программу. Используется следующая процедура:

1. Переведите ПЛК в режим работы.
2. Включите сервоприводы. Чтобы включить ось #1, переключите бит %Q смещение 18. Чтобы включить ось #2, переключите бит %Q смещение 34. В зависимости от текущего статуса ошибки модуля может также понадобиться инициировать подпрограмму очистки ошибки переключением бита %Q смещение 0.
3. Запустите для обеих осей подпрограмму начального позиционирования; для этого переключите бит %Q смещение 19 (начальное позиционирование оси #1) и бит %Q смещение 35 (начальное позиционирование оси #2). На этом этапе обе оси выполняют цикл начального позиционирования. Подождите, пока это не будет выполнено для обеих осей и пока не произойдет включение бита Position Valid (Позиция достигнута) %I. Таким битом Position Valid (Позиция достигнута) %I для оси 1 является бит %I смещение 17 (18-й бит %I), а для оси 2 - бит %I смещение 33 (34-й бит %I). Результирующее окно показано на рис. 16-1.

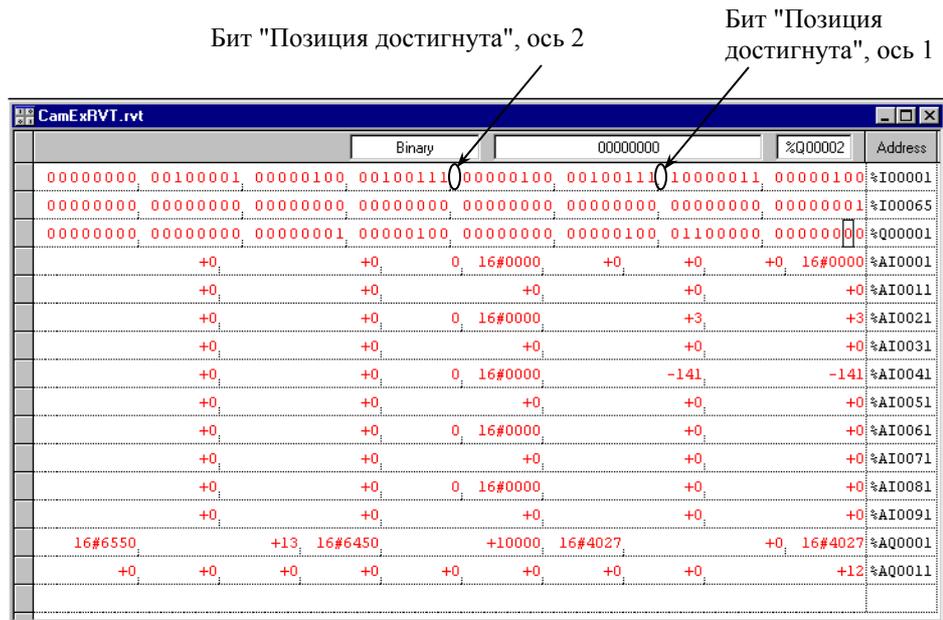


Рис. 16-38. Окно RVTEExample

4. Включите Local Logic установкой бита %Q смещение 1 от ПЛК. Если ошибок нет, то теперь можно выполнить программы движения.
5. Выполните программу 1 переключением бита %Q смещение 2. Двигатель, подключенный к оси #1, должен начать выполнять программу движения #1.
6. Выполните программу 2 переключением бита %Q смещение 3. Двигатель, подключенный к оси #2, должен начать выполнять программу движения #2.
7. Двигатели будут выполнять команды до тех пор, пока они не дойдут до первой команды DWELL (ЗАДЕРЖКА), где можно визуальнo проверить правильность отработки профиля САМ. Окно будет похоже на то, что представлено на рис. 16-2. Обратите внимание на то, что заданная позиция для Axis#2 (Ось #2) равна 2500, в то время как заданная позиция для ведомой оси соответствует САМ-таблице и имеет значение 10000.

Текущая позиция, ось 1      Заданная позиция, ось 1

Signed Decimal		000000000000000000010011100010000		%AI0007	Address			
00000000	00100001	00000100	00101111	00000100	00111111	10000011	00000110	%I0001
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000001	%I00065
00000000	00000000	00000001	00000100	00000000	00000100	01100000	00001110	%Q00001
+10000	+10000	130	16#0000	+0	+0	+0	16#0000	%AI0001
+0	+0	+0	+0	+0	+0	+0	+0	%AI0011
+2500	+2500	230	16#0000	-22	-22	-22	-22	%AI0021
-4	-4	-2	-2	+0	+0	+0	+0	%AI0031
+0	+0	0	16#0000	-214	-214	-214	-214	%AI0041
+0	+0	+0	+0	+0	+0	+0	+0	%AI0051
+0	+0	0	16#0000	+0	+0	+0	+0	%AI0061
+0	+0	+0	+0	+0	+0	+0	+0	%AI0071
+0	+0	0	16#0000	+0	+0	+0	+0	%AI0081
+0	+0	+0	+0	+0	+0	+0	+0	%AI0091
16#6550	+13	16#6450	+10000	16#4027	+0	16#4027	+0	%AQ0001
+0	+0	+0	+0	+0	+0	+0	+0	+12 %AQ0011

Текущая позиция, ось 2      Заданная позиция, ось 2

**Рис. 16-39. Окно первой задержки RVTEExample**

По истечении времени задержки двигатели продолжают выполнение команд до тех пор, пока не появится вторая команда DWELL, где можно визуальнo проверить правильность отработки. Окно будет похоже на то, что представлено на рис. 16-3. Обратите внимание на то, что заданная позиция для Axis#2 (Ось #2) равна 47500, в то время как заданная позиция для ведомой оси соответствует САМ-таблице и имеет значение 11000.

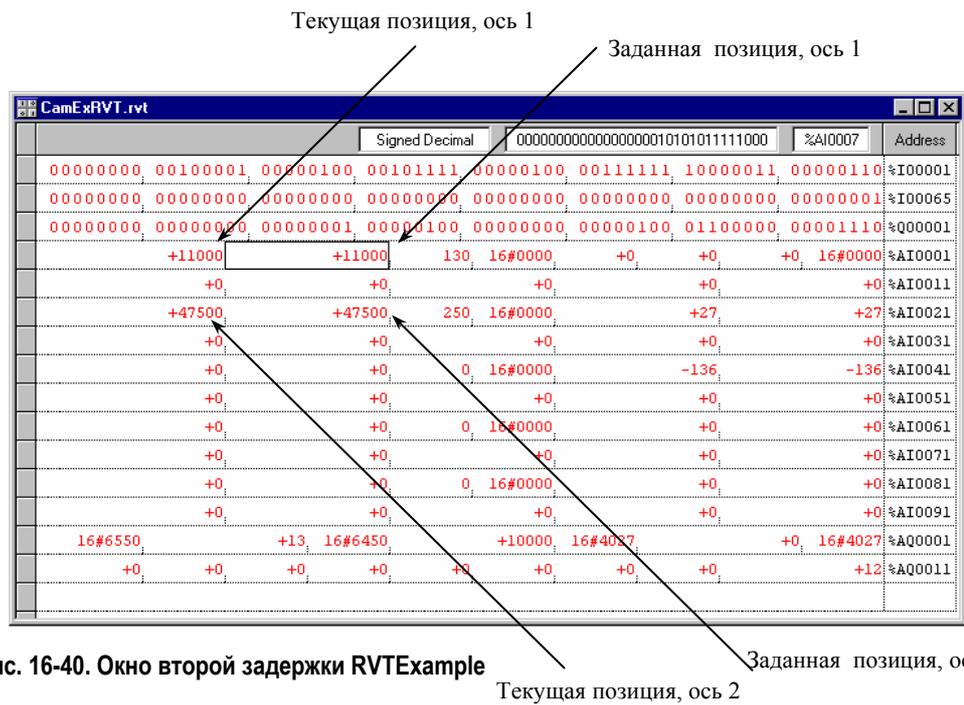


Рис. 16-40. Окно второй задержки RVTEExample

Когда ведущая ось достигнет 50000 (47500 +2500), то произойдет выход команды САМ, ведомая ось затормозится при запрограммированном значении изменения скорости и остановится; обе оси вернуться в ноль.

Подробная информация о памяти DSM %AI, %AQ, %I и %Q находится в главе 5.

## Коды ошибок модуля DSM314

Модуль DSM314 выдает коды ошибок в следующие места таблицы %AI:

%AI положение в таблице	Выдаваемые данные	Применение
00	Код статуса модуля	Ошибки не привязаны к конкретной оси
04	Код ошибки оси 1	Ошибки относятся к оси 1
24	Код ошибки оси 2	Ошибки относятся к оси 2
44	Код ошибки оси 3	Ошибки относятся к оси 3
64	Код ошибки оси 4	Ошибки относятся к оси 4

Каждый код ошибки представляет собой шестнадцатеричное слово, которое описывает указанную ошибку, когда установлен бит статуса *Module Error Present* (*Имеется ошибка модуля*) %I.

### Слово кода статуса модуля

Слово статуса *Module Status Code* (*Код статуса модуля*) %AI сообщает о следующих двух категориях ошибок:

- Ошибки модуля, которые не связаны с конкретной осью. Примерами таких ошибок являются аппаратные неисправности, выявляемые при самотестировании, или запрос на запуск пустой или дефектной программы. Новый *Module Status Code* (*Код статуса модуля*) не замещает предыдущий *Код статуса модуля*, если только новый *Код статуса модуля* не имеет приоритета Fast Stop (Быстрый останов) или System Error (Системная ошибка). Эти ошибки могут быть очищены битом %Q Clear Error (Сбросить ошибку).
- Ошибки статуса системы. Они имеют формат Dxxx, Exxx и Fxxx. Если имеется хотя бы один из этих кодов, то модуль работать не будет; бит %Q Clear Error (Сбросить ошибку) не может сбросить (очистить) эту ошибку. См. подробности в данном приложении в разделе "Ошибки статуса системы".

### Слова кода ошибки оси

Все ошибки, связанные с движением конкретных осей, выдаются в соответствующем слове статуса *Axis Error Code* (*Код ошибки оси*) %AI. Всякий раз, когда устанавливается бит статуса *Module Error Present* (*Имеется ошибка модуля*) %I, должны быть проверены все слова ошибок (включая *Module Status Code* (*Код статуса модуля*)) на предмет сообщения об ошибке. Новый *Axis Error Code* (*Код ошибки оси*) замещает предыдущий *Код ошибки оси*, если он имеет такой же или более высокий приоритет (Предупреждение, Нормальный останов, Быстрый останов) по сравнению с предыдущим *Кодом ошибки оси*.

Коды ошибок, которые останавливают ось, сбрасывают бит Axis OK (Ось в норме) %I для этой оси. Пользовательская логика, которая направляет команды %Q или %AQ на ось, должна проверять соответствующий бит %I Axis OK (Ось в норме). Если бит Axis OK (Ось в норме) находится в выключенном состоянии, то ось не будет реагировать на любой бит %Q или команды %AQ, кроме Clear Error (Сбросить ошибку) или Load Parameter (Загрузить параметр). Бит %Q Clear Error (Сбросить ошибку) будет всегда очищать Axis Error Code (Код ошибки оси); однако, если условие, которое вызвало ошибку, продолжает существовать, то сообщение об ошибке сразу же появится снова.

**Примечание:** Индикатор STAT на лицевой панели модуля мигает медленно (четыре раза в секунду) при ошибках только статуса и быстро (восемь раз в секунду) при ошибках, которые вызывают прекращение работы сервопривода. В случае фатальной аппаратной ошибки, обнаруженной при включении питания, индикатор своим миганием будет выдавать код ошибки, который следует сообщить в GE Fanuc. См. более детальную информацию в разделе "Светодиодные индикаторы" далее в данной главе.

### Формат кода ошибки

Все коды ошибок представляются шестнадцатеричными данными в следующем формате:

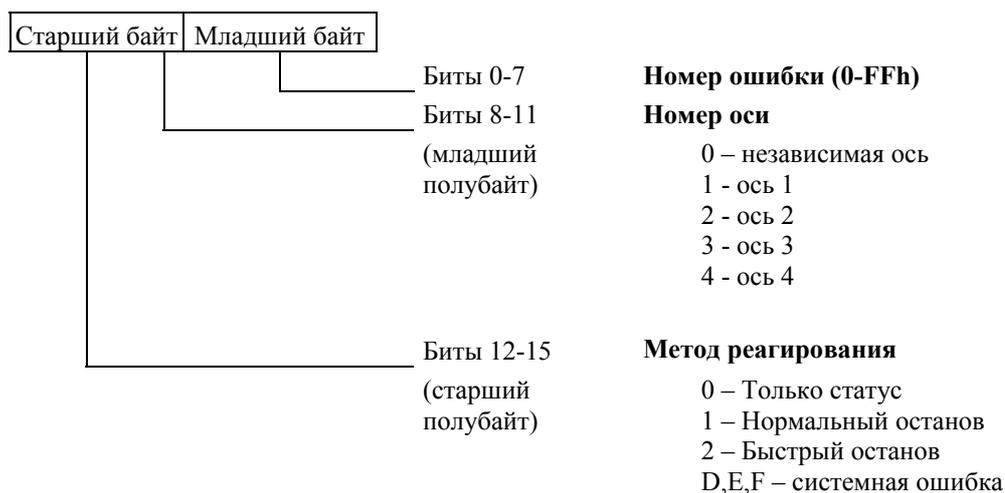


Рис. А-1. Структура кода статуса

### Методы реагирования

- Ошибки Status Only (Только статус):** Происходит установка бита *Module Error Present* (Имеется ошибка модуля) %I и слова *Module Status Code* (Код статуса модуля) или *Axis Error Code* (Код ошибки оси) %AI, но на движении это не сказывается.  
**Примечание:** Если не указано иное, то любая команда, которая вызывает ошибку Status Only (Только статус), игнорируется.
- Ошибки Stop Normal (Нормальный останов):** Выполняется внутреннее завершение любого текущего движения с использованием *Jog Acceleration* (Ускорение толчка) и *Jog Acceleration Mode* (Режим ускорения толчка) (LINEAR (ЛИНЕЙНЫЙ) или S-CURVE (S-КРИВАЯ)). Оба бита *Drive Enabled* (Привод включен) и *Axis Enabled* (Ось включена) %I переходят в состояние OFF (ВЫКЛ.) через сконфигурированное время *Drive Disable Delay* (Задержка отключения привода).
- Ошибки Stop Fast (Быстрый останов):** Происходит немедленное прекращение всех движений установкой заданной скорости на ноль. Оба бита *Drive Enabled* (Привод

включен) и *Axis Enabled* (Ось включена) %I переходят в состояние OFF (ВЫКЛ.) через сконфигурированное время *Drive Disable Delay* (Задержка отключения привода).

**Системные ошибки (отображаются только в коде статуса модуля):** DSM отключается и не реагирует на команды управления ПЛК. Системные ошибки не могут быть сброшены до тех пор, пока на DSM не будет направлена новая конфигурация.

Таблица А-1. Коды ошибок модуля DSM314

Код ошибки (шестнадцатеричный)	Реакция	Описание	Тип ошибки	Возможная причина
00	Нет	Нет ошибки	Все	
<b>Ошибки конфигурации</b>				
02	Только статус	Масштабированные данные имеют слишком большую величину, использовано максимальное значение диапазона	Ось	Проверьте конфигурацию оси модуля DSM в конфигурации аппаратного обеспечения
03	Только статус	Начальная позиция > значения положительного конца передвижения, использовано значение положительного конца передвижения	Ось	Проверьте конфигурацию оси модуля DSM в конфигурации аппаратного обеспечения
04	Только статус	Начальная позиция < значения отрицательного конца передвижения, использовано значение отрицательного конца передвижения	Ось	Проверьте конфигурацию оси модуля DSM в конфигурации аппаратного обеспечения
05	Только статус	Неправильная строка 1 параметров настройки; данные проигнорированы	Ось	Проверьте конфигурацию настройки DSM в конфигурации аппаратного обеспечения, закладка Advanced (Расширенные возможности), строка 1
06	Только статус	Неправильная строка 2 параметров настройки; данные проигнорированы	Ось	Проверьте конфигурацию настройки DSM в конфигурации аппаратного обеспечения, закладка Advanced (Расширенные возможности), строка 2
07	Только статус	Ошибка в строке 3 – 16 параметров настройки; данные проигнорированы	Ось	Проверьте конфигурацию настройки DSM в конфигурации аппаратного обеспечения, закладка Advanced (Расширенные возможности). Один или несколько параметров в строке 3 – 16 являются неправильными.
0A	Системная ошибка	Выход, в который Local Logic произвела запись, не сконфигурирован для управления от Local Logic.	Модуль	Имя блока Local Logic задано в конфигурации, а конфигурация аппаратного обеспечения (таблица Output Bits (Биты выхода)) для этого модуля не имеет требуемого выхода, сконфигурированного для управления от Local Logic.
0B	Системная ошибка	CTL-бит, в который Local Logic произвела запись, не сконфигурирован для управления от Local Logic.	Модуль	Имя блока Local Logic задано в конфигурации, а конфигурация аппаратного обеспечения (таблица CTL Bits (Биты CTL)) для этого модуля не имеет требуемого CTL-бита, сконфигурированного для управления от Local Logic.
<b>Ошибки параметров конфигурации</b>				
17	Только статус	Ошибка настройки EOT (конца передвижения)	Ось	Программный конец передвижения (EOT) включен в конфигурации, но значения верхнего или нижнего конца передвижения установлены за пределами верхней или нижней границ импульсов. Конфигурация должна быть изменена так, чтобы либо запретить использование программного конца передвижения, либо установить значения конца передвижения в границах импульсов.
18	Только статус	(Только вспомогательная) Масштабированное значение конца передвижения не является целым числом.	Ось	Проверьте конфигурацию оси модуля DSM в конфигурации аппаратного обеспечения.
19	Только статус	Модуль импульсов масштабированной верхней / нижней границы вращения не является целым числом.	Ось	Проверьте конфигурацию оси модуля DSM в конфигурации аппаратного обеспечения.
1C	Только	Неподдерживаемый режим команды AQ	Ось	Команды AQ, которые конфигурируют переменные

	статус			Torque Mode (Режим момента вращения) отсутствуют в Аналоговом режиме скорости.
<b>1D</b>	Нормальный останов	Попытка использовать команды CAM, CAM-Load или CAM-Phase с режимом Слежения.	Ось	Если используется CAM, то следует убедиться, что Follower Mode (режим слежения) не сконфигурирован (режим слежения не может быть использован, если используется CAM). Если используется Follower Mode (режим слежения), следует убедиться, что CAM-команды отсутствуют в программе движения (CAM не может быть использован, если сконфигурирован режим слежения).
<b>1E</b>	Только статус	Команда прямого действия Jog Velocity (Скорость толчка) находится за пределами диапазона, команда игнорируется.	Ось	Выданная AQ-команда прямого действия Jog Velocity (Скорость толчка) имеет слишком большое значение. Введите команду снова, используя меньшее значение.
<b>1F</b>	Только статус	Команда прямого действия Jog Acceleration (Ускорение толчка) находится за пределами диапазона, команда игнорируется.	Ось	Выданная AQ-команда прямого действия Jog Acceleration (Ускорение толчка) имеет слишком большое значение. Введите команду снова, используя меньшее значение.
<b>Ошибки программы</b>				
<b>20</b>	Только статус	Программное ускорение выходит за пределы диапазона, ускорение установлено на максимальное значение	Ось	Величина ускорения, запрограммированная в выполняемой в настоящий момент программе, слишком велика. В программе движения используется максимальное значение (1 073 741 823 имп./с <sup>2</sup> при масштабе 1:1).
<b>21</b>	Только статус	Программное ускорение слишком мало, по умолчанию 32 имп./с <sup>2</sup>	Ось	Величина ускорения, запрограммированная в выполняемой в настоящий момент программе, слишком мала. В программе движения используется значение по умолчанию (32 имп./с <sup>2</sup> ).
<b>22</b>	Только статус	Масштабированная скорость превышает или равна 1 млн. имп./с	Ось	Проверьте масштабирование в конфигурации, скорость в программе
<b>23</b>	Только статус	Программная скорость равна нулю, установлено минимальное значение 1 импульс/с	Ось	Программная скорость в программе, выполняемой в настоящий момент, равна нулю. Используется минимальное значение (1 импульс/с).
<b>24</b>	Только статус	Скорость в программе движения > сконфигурированной границы скорости, используется значение границы	Ось	Программная скорость в программе, выполняемой в настоящий момент, превышает значение границы скорости, установленное в конфигурации оси.
<b>25</b>		Зарезервировано – не используется в DSM314	Ось	
<b>26</b>	Нормальный останов	Ошибка Jump Mask (Маска перехода)	Ось	Обратитесь в GE Fanuc Automation
<b>27</b>	Нормальный останов	Ошибка Wait Mask (Маска ожидания)	Ось	Обратитесь в GE Fanuc Automation
<b>28</b>	Нормальный останов	Позиция в параметре слишком велика	Ось	Значение позиции, содержащееся в параметре, на который выполнена ссылка в исполняемой команде PMOVE или CMOVE выходит за пределы диапазона (от - 536 870 912 до +536 870 911 при масштабе 1:1)
<b>29</b>	Только статус	Время задержки превышает 60 секунд, разрешается 5 секунд	Ось	Выполняемая программа движения обнаружила оператор DWELL, где длительность задержки превышает 60 секунд. Это значение превышает разрешенное. Длительность задержки, разрешенная для команды DWELL, составляет 5 секунд. Пользователь должен войти в программу движения и исправить время у оператора DWELL так, чтобы оно было меньше 60 секунд. Если требуется большее время задержки, то используйте несколько операторов DWELL.
<b>2A</b>	Нормальный останов	CTL-условие выхода циклического CAM задано для не циклического CAM	Ось	CTL-условия выхода разрешены только для циклических CAM. Программа движения содержит команду не циклического CAM с CTL-условием выхода.
<b>2B</b>	Нормальный останов	Значение фазы CAM вне диапазона	Ось	Значение CAM PHASE находится за пределами диапазона позиционирования оси.
<b>Ошибки приращения позиции</b>				
<b>2C</b>	Только статус	Ошибка превышения значения приращения позиции, приращение игнорируется	Ось	Приращение позиции в команде AQ должно быть в диапазоне от – 128 до 128 пользовательских единиц.
<b>2D</b>	Нормальный останов	Ошибка конфигурации ведущей оси CAM – ведущий профиль не соответствует конфигурации ведущей оси.	Ось	1) Отношение числа пользовательских единиц к числу импульсов, заданное для ведущей оси в редакторе и в конфигурации аппаратного обеспечения, не совпадают и / или 2) Верхняя / Нижняя граница позиции, заданная для

				ведущей оси в конфигурации аппаратного обеспечения, не соответствует профилю. См. раздел по типам САМ с подробным описанием установок Верхней / Нижней границ позиционирования.
2E	Нормальный останов	Ошибка конфигурации ведомой оси САМ – ведомый профиль не соответствует конфигурации ведомой оси.	Ось	1) Отношение числа пользовательских единиц к числу импульсов, заданное для ведомой оси в редакторе и в конфигурации аппаратного обеспечения, не совпадают и / или 2) Верхняя / Нижняя граница позиции, заданная для ведомой оси в конфигурации аппаратного обеспечения, не соответствует профилю. См. раздел по типам САМ с подробным описанием установок Верхней / Нижней границ позиционирования.
2F	Нормальный останов	Режим SW EOT ведомой оси САМ не может быть разрешен для кругового циклического САМ	Ось	
<b>Ошибки начального позиционирования</b>				
30	Только статус	Начальное позиционирование в то время, когда привод выключен	Ось	Была выполнена команда Find Home (Поиска начальной позиции) в то время как бит Drive Enable (Включить привод) не был установлен. Пользователь должен включить привод и снова выполнить команду.
31	Только статус	Начальное позиционирование в то время, когда была выбрана программа	Ось	Была выполнена команда Find Home (Поиска начальной позиции) в то время как была выбрана для выполнения программа движения. Программа движения должна быть остановлена (1-бит Program Active (Программа активна) должен быть сброшен) до того, как выполнять команду начального позиционирования.
32	Только статус	Начальное позиционирование в то время, когда включена форсировка скорости цифрового сервопривода или включен аналоговый выход	Ось	Была выполнена команда Find Home (Поиска начальной позиции) в то время как пользователь выдал AQ-команду Force Digital Servo Velocity (Включить форсировку скорости цифрового сервопривода) (34h) или Force Analog Output (Включить аналоговый выход) (24h) AQ. Пользователь должен сбросить эту команду до выполнения команды Find Home (Поиска начальной позиции).
33	Только статус	Начальное позиционирование во время толчка	Ось	Пользователь выдал команду Find Home (Поиска начальной позиции), когда сервопривод выполнял толчковое передвижение. Пользователь должен сбросить команду Jog (Толчок) до выполнения команды Find Home (Поиска начальной позиции).
34	Только статус	(1) Начальное позиционирование во время движения на скорости или (2) Начальное позиционирование в то время, когда еще продолжает действовать другой цикл начального позиционирования	Ось	Пользователь выдал команду Find Home (Поиска начальной позиции), когда (1) выполнялась AQ-команда Move at Velocity (Движение на скорости) (22h) или (2) действовал другой цикл Find Home (Поиска начальной позиции). В (1) необходимо остановить операцию Move at Velocity (Движение на скорости) (сбросить бит движения I) перед тем, как выполнять команду Find Home (Поиска начальной позиции). Во (2) следует убедиться, что ось находится в зоне и не двигается, прежде чем выполнять команду Find Home (Поиска начальной позиции).
35	Только статус	Начальное позиционирование, в то время как включено слежение	Ось	Пользователь выдал команду Find Home (Поиска начальной позиции), когда была включена функция слежения. Пользователь должен выключить слежение (сбросить I-бит Follower Enabled (Слежение включено)), прежде чем выполнять команду Find Home (Поиска начальной позиции).
36	Только статус	Начальное позиционирование в то время, как установлен, бит Abort (Прекратить)	Ось	Пользователь выдал команду Find Home (Поиска начальной позиции) в то время как был установлен, бит Abort (Прекратить). Пользователь должен сбросить команду Abort (Прекратить) до выполнения команды Find Home (Поиска начальной позиции).
37	Только статус	Начальное позиционирование на первом цикле ПЛК	Ось	Q-бит Find Home (Поиска начальной позиции) был установлен на первом цикле ПЛК. Программа ПЛК должна быть откорректирована так, чтобы предотвратить выдачу этой команды на первом цикле ПЛК.
<b>Ошибки движения на скорости</b>				
38	Только статус	Команда движения на скорости дана на первом цикле ПЛК	Ось	Команда Move at Velocity (Движение на скорости) (22h) была выдана во время первого цикла ПЛК. Программа ПЛК должна быть откорректирована так, чтобы

				предотвратить выдачу этой команды на первом цикле ПЛК.
<b>39</b>	Только статус	Команда движения на скорости дана при не включенном приводе	Ось	Команда Move at Velocity (Движение на скорости) (22h) была выдана при не установленном бите Drive Enable (Включить привод). Пользователь должен включить привод и снова выполнить команду.
<b>3A</b>	Только статус	Команда движения на скорости дана при выбранной программе	Ось	Команда Move at Velocity (Движение на скорости) (22h) была выдана, в то время как была выбрана для выполнения команда движения. Программа движения должна быть остановлена (1-бит Program Active (Программа активна) должен быть сброшен) до того, как выполнять команду Move at Velocity (Движение на скорости).
<b>3B</b>	Только статус	Команда движения на скорости дана при активном начальном позиционировании	Ось	Команда Move at Velocity (Движение на скорости) (22h) была выдана, в то время как модуль выполнял Home Cycle (Цикл начального позиционирования). Пользователь должен прекратить выполнение цикла начального позиционирования, либо подождать его завершения прежде, чем выдавать команду Move at Velocity (Движение на скорости).
<b>3C</b>	Только статус	Команда движения на скорости дана во время толчка	Ось	Команда Move at Velocity (Движение на скорости) (22h) была выдана при активном Q-бите Jog (Толчок). Пользователь должен сбросить команду Jog (Толчок) прежде, чем выдавать команду Move at Velocity (Движение на скорости).
<b>3D</b>	Только статус	Команда движения на скорости дана при установленном бите Прекратить все движения	Ось	Команда Move at Velocity (Движение на скорости) (22h) была выдана при установленном Q-бите Abort All Moves (Прекратить все движения). Пользователь должен сбросить команду Abort All Moves (Прекратить все движения) прежде, чем выдавать команду Move at Velocity (Движение на скорости).
<b>3E</b>	Только статус	Значение команды движения на скорости превышает 8 388 607 пользовательских единиц в секунду	Ось	Пользователь выдал команду Move at Velocity (Движение на скорости) (22h), в которой заданная скорость превышает 8 388 607 пользовательских единиц в секунду. Пользователь должен уменьшить заданную скорость, прежде чем снова выдать команду.
<b>3F</b>	Только статус	Значение команды движения на скорости превышает 1 млн. импульсов в секунду	Ось	Пользователь выдал команду Move at Velocity (Движение на скорости) (22h), в которой заданная скорость превышает 1 млн. импульсов в секунду. Пользователь должен уменьшить заданную скорость, прежде чем снова выдать команду. Проверьте масштабирование.
<b>Ошибки толчка</b>				
<b>40</b>	Только статус	Толчок во время начального позиционирования	Ось	Пользователь выдал команду Jog (Толчок) в то время как модуль выполнял функцию начального позиционирования. Следует либо прекратить выполнение функции начального позиционирования, либо подождать, пока позиционирование закончится, прежде чем давать команду Jog (Толчок).
<b>41</b>	Только статус	Толчок во время движения на скорости	Ось	Пользователь установил Q-бит Jog (Толчок) в то время как модуль выполнял команду Move at Velocity (Движение на скорости) (22h). Необходимо остановить действие команды Move at Velocity (Движение на скорости) до того, как выполнять Jog (Толчок).
<b>42</b>	Только статус	Толчок во время включения скорости цифрового сервопривода	Ось	Пользователь установил Q-бит Jog (Толчок) в то время как модуль выполнял AQ-команду Force Digital Velocity (Включить скорость цифрового сервопривода) (34h) или Force Analog Output (Включить аналоговый выход) (24h). AQ-команда должна быть удалена до того, как выполнять Jog (Толчок).
<b>43</b>	Только статус	Толчок, в то время как выбрана программа и не действует остановка подачи	Ось	Во время работы программы модуль DSM может выполнить Jog (Толчок) только в том случае, если установлен, Q-бит Feedhold (Остановка подачи).
<b>Ошибки, связанные с командой включения скорости цифрового сервопривода</b>				
<b>47</b>	Только статус	Команда включения скорости цифрового сервопривода или включения аналогового выхода во время толчка	Ось	Пользователь выдал AQ-команду Force Digital Servo Velocity (Включить форсировку скорости цифрового сервопривода) (34h) или Force Analog Output (Включить аналоговый выход) (24h) во время выполнения модулем

				функции Jog (Толчок). Перед выполнением этих команд функция Jog (Толчок) должна быть остановлена.
48	Только статус	Команда включения скорости цифрового сервопривода или включения аналогового выхода во время движения на скорости	Ось	Пользователь выдал AQ-команду Force Digital Servo Velocity (Включить форсировку скорости цифрового сервопривода) (34h) или Force Analog Output (Включить аналоговый выход) (24h) во время выполнения модулем функции Move at Velocity (Движение на скорости). Перед выполнением этих команд функция Move at Velocity (Движение на скорости) должна быть остановлена.
49	Только статус	Команда включения скорости цифрового сервопривода или включения аналогового выхода в то время, как выбрана программа	Ось	Пользователь выдал AQ-команду Force Digital Servo Velocity (Включить форсировку скорости цифрового сервопривода) (34h) или Force Analog Output (Включить аналоговый выход) (24h) во время выполнения модулем программы движения. Перед выполнением этих команд программа движения должна быть остановлена (1-бит Program Active (Программа активна) сброшен).
4A	Только статус	Команда включения скорости цифрового сервопривода или включения аналогового выхода при включенном слежении	Ось	Пользователь выдал AQ-команду Force Digital Servo Velocity (Включить форсировку скорости цифрового сервопривода) (34h) или Force Analog Output (Включить аналоговый выход) (24h) при включенном слежении. Перед выполнением этих команд слежение должно быть выключено (1-бит Follower Enabled (Слежение включено) сброшен).
4B	Только статус	Включен аналоговый выход в то время, как действует аналоговый режим момента вращения	Ось	Пользователь выдал AQ-команду Force D/A (Включить цифровой / аналоговый выход) (24h) в то время, как сервопривод был сконфигурирован для аналогового режима момента вращения. Команда Force Analog Output (Включить аналоговый выход) не поддерживается в аналоговом режиме момента вращения.
<b>Ошибки позиционирования</b>				
50	Только статус	Установка позиции при выбранной программе	Ось	Пользователь выдал команду Set Position (Установить позицию) в то время, как была выбрана для выполнения программа движения. Программа движения должна быть остановлена (1-бит Program Active (Программа активна) должен быть сброшен) до того, как выполнять команду Set Position (Установить позицию).
51	Только статус	Данные установки позиции за пределами диапазона	Ось	Пользователь выдал команду Set Position (Установить позицию) со значением, превышающим максимальный диапазон позиционирования (от -536 870 912 до +536 870 911 при масштабе 1:1).
52	Только статус	Установка позиции в то время, как происходит движение, или установка позиции при отсутствии движения, при нахождении вне зоны, но скорость > 100 импульсов в секунду.	Ось	Команда Set Position (Установить позицию) не допускается, если бит Moving (Движение) %I включен. Если бит Moving (Движение) выключен и бит %I In Zone (В зоне) также выключен, то Текущая скорость должна быть менее 100 импульсов в секунду.
53	Только статус	Попытка инициализировать позицию до того, как цифровой датчик положения достигнет начальной точки.	Ось	Абсолютный цифровой датчик положения не прошел через нулевую опорную точку после первой подачи питания. В абсолютном режиме выдача команды Set Position (Установить позицию) разрешена после того, как датчик положения при повороте пройдет через опорную точку (до 1 оборота).
54	Только статус	Неправильная позиция цифрового датчика положения, следует использовать команду начального позиционирования или установки позиции	Ось	<ol style="list-style-type: none"> <li>1. Не была инициализирована абсолютная позиция датчика положения после первого включения питания.</li> <li>2. Конфигурация для режима датчика положения была изменена с инкрементной на абсолютную.</li> <li>3. Была изменена конфигурация направления оси (нормальная или обратная).</li> <li>4. Было изменено разрешение датчика положения (задается параметром конфигурации в расширенной таблице)</li> <li>5. Было выдано аварийное сообщение датчика положения</li> </ol>
55	Только статус	Цифровой датчик положения переместился слишком далеко при отключении питания	Ось	Цифровой абсолютный датчик положения переместился более чем на 16383 оборота после отключения питания.
<b>Ошибки, связанные с окончанием передвижения и ограничением числа импульсов</b>				
56	Только статус	Заданная позиция > положительного значения конца передвижения или верхней	Ось	Пользователь выдал команду, которая привела к тому, что заданная позиция сервопривода превысила

		границы числа импульсов		положительное значение конца передвижения или верхнюю границу числа импульсов. Следует либо так изменить команду, чтобы заданная позиция оказалась меньше этих значений, либо увеличить эти значения в конфигурации аппаратного обеспечения.
57	Только статус	Заданная позиция < отрицательного значения конца передвижения или нижней границы числа импульсов	Ось	Пользователь выдал команду, которая привела к тому, что заданная позиция сервопривода превысила отрицательное значение конца передвижения или нижнюю границу числа импульсов. Следует либо так изменить команду, чтобы заданная позиция оказалась больше этих значений, либо сдвинуть эти значения в отрицательную сторону в конфигурации аппаратного обеспечения.
58	Только статус	Позиция абсолютного датчика положения > верхней программной границы конца передвижения	Ось	Это сообщение об ошибке выдается при включении питания или при реконфигурации, если абсолютный цифровой датчик положения был сдвинут за верхнюю программную границу конца передвижения.
59	Только статус	Позиция абсолютного датчика положения < нижней программной границы конца передвижения	Ось	Это сообщение об ошибке выдается при включении питания или при реконфигурации, если абсолютный цифровой датчик положения был сдвинут за нижнюю программную границу конца передвижения.
<b>Ошибки выключения привода</b>				
5B	Нормальный останов	Привод выключен во время движения	Ось	Q-бит Enable Drive (Включить привод) был выключен, когда сервопривод выполнял толчок или движение на скорости (был установлен I-бит Moving (Движение)). Должна быть откорректирована программа ПЛК, чтобы устранить эту ошибку. Следует рассмотреть, используя бит Moving (Движение) логику, которая отключает привод.
5C	Нормальный останов	Привод выключен в то время, как программа активна	Ось	Q-бит Enable Drive (Включить привод) был выключен, когда сервопривод выполнял программу движения (был установлен I-бит Program Active (Программа активна)). Должна быть откорректирована программа ПЛК, чтобы устранить эту ошибку. Следует рассмотреть, используя бит Program Active (Программа активна) логику, которая отключает привод.
<b>Ошибки программного обеспечения</b>				
5F	Только статус	Ошибка программного обеспечения (звоните в службу поддержки GE Fanuc)	Ось	Обратитесь в GE Fanuc Automation
60	Только статус	Ошибка расчета круговой позиции абсолютного датчика положения	Ось	Обратитесь в GE Fanuc Automation
<b>Ошибки программ и подпрограмм</b>				
61	Нормальный останов	Неправильный номер подпрограммы	Ось	Программа движения вызвала подпрограмму, которая отсутствует среди программ модуля. Если команда вызова ссылается на параметр, который содержит номер подпрограммы, убедитесь, что данные параметра правильные.
62	Нормальный останов	Ошибка вызова (подпрограмма уже действует на оси)	Ось	Подпрограмма движения вызывает сама себя или вызывает другую подпрограмму, которая вызывает исходную.
63	Нормальный останов	В программе обнаружена команда конца подпрограммы	Ось	Программа движения содержит неправильную команду конца подпрограммы внутри основной программы движения (программы 1 – 10). Откорректируйте программу движения, чтобы удалить этот оператор.
64	Нормальный останов	В подпрограмме обнаружена команда конца программы	Ось	Подпрограмма движения содержит неправильную команду конца программы внутри подпрограммы движения (программы 1 - 40). Откорректируйте подпрограмму так, чтобы удалить этот оператор.
65	Нормальный останов	В программе, не требующей синхронизации, обнаружена подпрограмма синхронизации	Ось	Программа движения обнаружила блок синхронизации в программе, которая не является многоосной и которая не была установлена для использования блоков синхронизации.
66	Нормальный останов	SAM-профиль не обнаружен в блоке загрузки SAM	Ось	SAM-профиль не был связан с блоком загрузки SAM в редакторе SAM и / или имя блока загрузки SAM не было задано в конфигурации аппаратного обеспечения.
67	Нормальный останов	Расстояние выхода SAM вне диапазона (нециклические SAM)	Ось	Расстояние выхода для нециклического SAM было больше, чем модуль для этого SAM.
68	Только статус	(Коррекция разрешена) Заданная скорость ограничена из-за нарушения	Ось	

		границы скорости или нарушения предела ошибки по положению		
69	Нормальный останов	(Коррекция запрещена) Заданная скорость САМ превышает границу скорости сконфигурированной оси	Ось	
6A	Нормальный останов	Нарушение предела ошибки САМ по положению (с запрещенной коррекцией)	Ось	
6B	Только статус	Заданная САМ-позиция при выходе отличается от значения САМ-профиля из-за ошибки по положению или ограничения скорости	Ось	
6C	Нормальный останов	Значение ведущей оси САМ находится за пределами диапазона профиля ведущей оси для нециклического профиля (команды САМ и САМ-LOAD)	Ось	
6D	Нормальный останов	Абсолютный режим САМ последовательно с инкрементным режимом САМ	Ось	
6E	Быстрый останов	Ошибка расчета траектории САМ	Ось	Обратитесь в GE Fanuc Automation
<b>Ошибки выполнения программы</b>				
70	Только статус	Выполнение программы на первом цикле ПЛК	Модуль	Q-бит Execute Program (Выполнить программу) был установлен на первом цикле ПЛК. Программа ПЛК должна быть откорректирована так, чтобы предотвратить установку этого Q-бита от установки на первом цикле.
71	Только статус	Слишком много программ запрошено в одном цикле ПЛК	Модуль	Количество Q-битов Execute Program (Выполнить программу), которые были включены на первом цикле, превышает сконфигурированное число осей.
72	Только статус	Запрос на выполнение программы для оси 1 или 2 в то время, как работает многоосная программа	Модуль	Q-бит Execute Program (Выполнить программу) был установлен для программы оси 1 или оси 2 в то время, когда уже действовала многоосная программа.
73	Только статус	Запрос на выполнение программы для оси, сконфигурированной как ограниченная вспомогательная ось	Модуль	Программы движения не могут быть выполнены на оси, сконфигурированной как Limited Aux (Ограниченная вспомогательная). Limited Aux (Ограниченная вспомогательная) ось выполняет только обработку позиции обратной связи и не имеет внутреннего формирователя пути.
74		Зарезервировано – не используется в DSM314		
75	Только статус	Запрошена пустая или неправильная программа	Модуль	Q-бит Execute Program (Выполнить программу) был установлен для номера программы, который не задан в сконфигурированном блоке программы движения. Проверьте конфигурацию на правильность имени блока программы движения. Убедитесь, что запрошенный номер программы задан в сконфигурированном блоке программы.
76	Только статус	Позиция AQ-команды движения находится за пределами диапазона	Ось	Пользователь выдал AQ-команду движения (27h) со значением позиции, превышающим максимальное. (от -536 870 912 до +536 870 911 при масштабе 1:1)
77	Только статус	AQ-команда движения на первом цикле ПЛК	Ось	AQ-команда движения (27h) была выдана на первом цикле ПЛК. Программа ПЛК должна быть откорректирована так, чтобы предотвратить выдачу этой команды на первом цикле.
<b>Ошибки условий выполнения программы</b>				
80	Только статус	Запрос на выполнение программы во время работы цикла начального позиционирования.	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) в то время, как модуль выполнял цикл начального позиционирования. Пользователь до выполнения программы движения должен либо подождать завершения цикла начального позиционирования, либо прекратить этот цикл.
81	Только статус	Запрос на выполнение программы во время толчка	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) в то время, как модуль выполнял операцию толчка. До начала выполнения программы движения биты Jog (Толчок) (от ПЛК или локальные логические) должны быть выключены.
82	Только статус	Запрос на выполнение программы во время движения на скорости	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) в то время, как модуль выполнял команду Move at Velocity (Движение на скорости) (22h). Команда

				Move at Velocity (Движение на скорости) должна быть снята до выполнения программы движения.
83	Только статус	Запрос на выполнение программы в то время, когда включена скорость цифрового сервопривода или включен аналоговый выход	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) в то время, как модуль выполнял AQ-команду Force Digital Velocity (Включить скорость цифрового сервопривода) (34h) или Force Analog Output (Включить аналоговый выход) (24h). Команда Force Digital Velocity (Включить скорость цифрового сервопривода) или Force Analog Output (Включить аналоговый выход) должна быть снята до выполнения программы движения.
84	Только статус	Запрос на выполнение программы в то время, как программа активна	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) для оси, когда та уже выполняла программу движения. Текущая программа должна быть завершена (I-бит Program Active (Программа активна) выключен) до начала выполнения другой программы на той же оси.
85	Только статус	Запрос на выполнение программы в то время, как установлен бит Abort All Moves (Прекратить все движения)	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) в то время, как модуль выполнял команду Abort All Moves (Прекратить все движения). До начала выполнения программы должны быть выключены Q-бит Abort (Прекратить), I-бит Moving (Движение) и I-бит Program Active (Программа активна).
86	Только статус	Запрос на выполнение программы в то время, как не установлен бит Position Valid (Позиция достигнута)	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) в то время, как I-бит Position Valid (Позиция достигнута) был выключен. I-бит Position Valid (Позиция достигнута) должен быть установлен в результате выполнения цикла Find Home (Поиска начальной позиции) или команды Set Position (Установить позицию).
87	Только статус	Запрос на выполнение программы в то время, как не установлен бит Drive Enabled (Привод включен)	Ось	ПЛК установил Q-бит Execute Program (Выполнить программу) в то время, как привод был не включен (I-бит Drive Enabled (Привод включен) был не установлен). Q-бит Enable Drive (Включить привод) должен быть установлен, чтобы включить привод.
<b>Ошибки блока синхронизации программ</b>				
8C	Только статус	Ошибка блока синхронизации во время CMOVE	Ось	При выполнении программы поступила команда CMOVE, идентифицированная блоком синхронизации, хотя другая ось еще не достигла блока синхронизации.
8D	Только статус	Ошибка блока синхронизации во время Jump (Переход)	Ось	При выполнении программы произошел переход на команду CMOVE или PMOVE, идентифицированную блоком синхронизации, хотя другая ось еще не достигла блока синхронизации.
<b>Ошибки EEPROM (электрически стираемое программируемое ПЗУ)</b>				
90	Только статус	Ошибка программирования флэш-памяти EEPROM	Модуль	Обратитесь в GE Fanuc Automation
<b>Ошибки Local Logic</b>				
91	Быстрый останов	Системный останов Local Logic	Модуль	Программа Local Logic выполнила оператор, который записал переменную System_Halt (Системный_Останов) (например, System_Halt := 1;).
92	Быстрый останов	Ошибка ограничения по времени Local Logic	Модуль	Программа Local Logic превысила выделенное значение длительности выполнения 300 микросекунд. Следует уменьшить длительность выполнения программы Local Logic путем уменьшения количества операторов Local Logic или изменения структуры программы. Более подробную информацию по длительности выполнения локальных логических программ см. в Приложении E.
93	Быстрый останов	Ошибка деления на ноль в Local Logic	Модуль	Программа Local Logic предприняла деление на ноль или операцию получения модуля с нулем. Проверьте операторы деления программы Local Logic для обнаружения источника ошибки. Возможными источниками этих ошибок являются регистры параметров, содержащие нулевые значения.
94	Быстрый останов	Ошибка переполнения при операциях Local Logic деление / получение модуля	Модуль	Программа Local Logic предприняла деление (или операцию получения модуля) с целым 64-битовым числом, и результат не может быть описан 32-битовым целым числом. Проверьте операторы деления программы Local Logic для обнаружения источника

				ошибки.
95	Только статус	Предупреждение о переполнении при сложении / вычитании в Local Logic	Модуль	Программа Local Logic выполнила сложение или вычитание чисел, что вызвало переполнение. Допустимый диапазон значений лежит от -2 147 483 648 до +2 147 483 647. Измените локальную логическую программу так, чтобы исключить переполнение, или устанавливайте переменную Overflow (Переполнение) на 0 в конце каждого локального логического цикла.
96	Только статус	Предупреждение о переполнении при операции получения абсолютного значения (ABS) в Local Logic	Модуль	Программа Local Logic попыталась выполнить операцию получения абсолютного значения (ABS) на -2 147 483 648, что привело к переполнению.
97	Только статус	Предупреждение об ограничении времени в Local Logic	Модуль	Длительность выполнения программы Local Logic приближается (составляет более 275 мкс) к максимально допустимому значению длительность выполнения (300 микросекунд). Следует уменьшить длительность выполнения программы Local Logic путем уменьшения количества операторов Local Logic или изменения структуры программы. Более подробную информацию по длительности выполнения локальных логических программ см. в Приложении E.
98	Только статус	Ошибка - выполнение программы Local Logic на первом цикле	Модуль	Пользователь сделал попытку выполнить программу Local Logic на первом цикле ПЛК (т.е. Q-бит Local Logic enable (Включить Local Logic) оказался во включенном состоянии, когда ПЛК переключился с режима Стоп на режим Работа).
99	Только статус	Неправильное имя программы в Local Logic или Local Logic не разрешена при конфигурации	Модуль	Имя программы Local Logic, заданное в конфигурации аппаратного обеспечения, неправильное (или отсутствует), или Local Logic не разрешена в конфигурации аппаратного обеспечения.
9A	Быстрый останов	Ошибка останова Local Logic (для оси)	Ось	Произошла ошибка быстрого останова Local Logic (коды ошибки 91 – 94).
<b>Ошибки конечных переключателей аппаратного обеспечения</b>				
A0	Быстрый останов	Ошибка конечного переключателя (+)	Ось	Вход положительного конечного переключателя находится перерегулирования в состоянии "выкл.". Если конечные переключатели перерегулирования не используются, то конфигурацию конечного переключателя перерегулирования следует установить в состоянии Disabled (Отключено).
A1	Быстрый останов	Ошибка конечного переключателя (-)	Ось	Вход отрицательного конечного переключателя перерегулирования находится в состоянии "выкл.". Если конечные переключатели перерегулирования не используются, то конфигурацию конечного переключателя перерегулирования следует установить в состоянии Disabled (Отключено).
<b>Ошибки аппаратного обеспечения</b>				
A8	Быстрый останов	Ошибка потери синхронизации	Ось	Ошибка по положению превысила предел ошибки по положению. Возможными источниками этой ошибки являются следующие: 1. Предел ошибки по положению был задан очень малым для данного применения. 2. Произошло отсоединение устройства обратной связи от управляемого устройством или имеет место проскальзывание. 3. Неправильное подключение устройства обратной связи, (т.е. положительное направление вращения в этом устройстве воспринимается как отрицательное).
A9	Только статус	Потеря обратной связи по положению	Ось	На инкрементном импульсном датчике положения обнаружена квадратурная ошибка. Проверьте подключение датчика положения и убедитесь, что датчик используется при скорости, не превышающей его номинальную.
B0- BE		См. таблицу A-3		Аварийные сообщения цифрового сервопривода описаны в таблице A-3.

Аварийные сообщения датчика положения				
<b>C0</b>	Быстрый останов	Сервоустройство не готово	Ось	<p>Для аналогового сервоустройства вход на лицевой панели Drive Ready (Готов к запуску) должен быть установлен в состояние "вкл." (0 В) в течение 1 секунды после включения Q-бита Enable Drive (Включить привод). Если вход Drive Ready (Готов к запуску) не используется для аналогового сервоустройства, то входная конфигурация должна быть установлена на Disabled (Отключено).</p> <p>Для цифровых сервоустройств FANUC разрешается активировать вход E-стоп усилителя; в противном случае может произойти отказ усилителя.</p>
<b>C1</b>	Только статус	Разряжена батарея последовательного датчика положения	Ось	Мало напряжение батареи последовательного датчика положения. Батарея должна быть заменена, или датчик положения должен быть сконфигурирован для инкрементной (вместо абсолютной) работы.
<b>C2</b>	Нормальный останов	Батарея последовательного датчика положения неисправна	Ось	Батарея последовательного датчика положения неисправна. Батарея должна быть заменена, или датчик положения должен быть сконфигурирован для инкрементной (вместо абсолютной) работы.
<b>C3</b>	Нормальный останов	Превышение температуры серводвигателя	Ось	Серводвигатель или управляющая микропрограмма выдала сообщение о превышении температуры. Пользователь должен проверить программу движения, чтобы убедиться, что для мотора не превышено значение цикла нагрузки. Необходимо проверить также установку двигателя, чтобы убедиться в правильности действия теплоотвода и вентиляции двигателя.
<b>C4</b>	Нет	Не используется.	Нет	
<b>C5</b>	Быстрый останов	Датчик положения потерян	Ось	Модуль не имеет связи с датчиком положения. Убедитесь, что сервоусилитель включен. Проверьте подключение датчика положения, чтобы убедиться, что кабели подключены. Проверьте также правильность выполнения заземления.
<b>C6</b>	Быстрый останов	Ошибка в обнаружении импульсов датчика положения	Ось	Схема обнаружения импульсов датчика положения обнаружила ошибку. Убедитесь, что двигатель заземлен должным образом. Если ошибка сохранилась, то обратитесь к изготовителю.
<b>C7</b>	Быстрый останов	Ошибка счетчика датчика положения	Ось	Схема счетчика импульсов датчика положения обнаружила ошибку. Убедитесь, что двигатель заземлен должным образом. Если ошибка сохранилась, то обратитесь к изготовителю.
<b>C8</b>	Быстрый останов	Отсоединен индикатор датчика положения	Ось	Отсоединен индикатор датчика положения. Обратитесь к изготовителю.
<b>C9</b>	Быстрый останов	Ошибка контрольной суммы CRC датчика положения	Ось	Схема связи датчика положения обнаружила ошибку CRC (контроль при помощи циклического избыточного кода). Проверьте заземление кабеля датчика положения и заземление двигателя, как возможные источники ошибки. Проверьте возможность наличия других источников электрических помех вблизи кабелей двигателя и датчика положения. Заизолируйте эти источники от кабелей двигателя и датчика положения, если это возможно. Если ошибка сохранилась, то обратитесь к изготовителю.
<b>CA</b>	Быстрый останов	Неподдерживаемый датчик положения, линейный или тип A	Ось	Встроенный в двигатель датчик положения, подключенный к модулю, не поддерживается. Двигатель либо не поддерживается модулем DSM, либо имеет неподлежащий датчик положения, встроенный в двигатель. По табличке двигателя проверьте, поддерживается ли эта модель. Если проблема продолжает существовать, то обратитесь к изготовителю.
<b>CB</b>	Быстрый останов	Неподдерживаемый датчик положения, тип C	Ось	Встроенный в двигатель датчик положения, подключенный к модулю, не поддерживается. Двигатель либо не поддерживается модулем DSM, либо имеет неподлежащий датчик положения, встроенный в двигатель. По табличке двигателя проверьте,

				поддерживается ли эта модель. Если проблема продолжает существовать, то обратитесь к изготовителю.
CC	Нормальный останов	Отсутствует импульс DZ, когда происходит переход DS от 1 к 0	Ось	Данные позиционирования могут быть неправильными. Выключите и снова включите двигатель и усилитель. Если проблема продолжает существовать, то обратитесь к изготовителю.
<b>Аварийные сообщения DSP (цифровая обработка сигналов)</b>				
D1	Быстрый останов	Обнаружено повышенное значение тока	Ось	Микропрограмма управления двигателем обнаружила повышенное значение тока. Возможными источниками этой ошибки являются следующие: <ul style="list-style-type: none"> <li>- При конфигурации аппаратного обеспечения задан неправильный тип двигателя.</li> <li>- Станок чрезмерно нагружает приводной двигатель.</li> <li>- Имеют место условия цикла перегрузки</li> </ul>
D2	Нет	Не используется		
D3	Быстрый останов	Обнаружено повышенное значение ускорения	Ось	Микропрограмма управления двигателем выявила наличие ускорения, которое превышает разрешенное значение. Эта ошибка не появляется при нормальных условиях работы. Источниками ошибки могут быть неисправность датчика положения, его проскальзывание, неправильная позиция, сообщаемая датчиком положения. Если ошибку не удастся объяснить аппаратным обеспечением, то следует обратиться к производителю.
D4	Быстрый останов	Обнаружено повышенное значение скорости	Ось	Микропрограмма управления двигателем выявила наличие скорости, которая превышает разрешенное значение. Эта ошибка не появляется при нормальных условиях работы. Источниками ошибки могут быть неисправность датчика положения, его проскальзывание, неправильная позиция, сообщаемая датчиком положения. Если ошибку не удастся объяснить аппаратным обеспечением, то следует обратиться к производителю.
D5	Только статус	Коэффициент обратной связи по скорости для Kp слишком велик	Ось	Пропорциональный коэффициент обратной связи по скорости превысил заданное значение. Значение ограничено допустимым диапазоном. Эта ошибка не должна появляться при нормальной работе. Источниками ошибки могут быть задание неправильного типа двигателя при конфигурации аппаратного обеспечения или слишком большие значения коэффициента обратной связи по скорости. Если тип двигателя в конфигурации задан правильно, то следует уменьшить коэффициент обратной связи по скорости. Если проблема продолжает существовать или если коэффициент обратной связи по скорости становится слишком малым для данного применения, то следует обратиться к изготовителю.
D6	Только статус	Интегральный коэффициент слишком велик	Ось	Интегральный коэффициент обратной связи по скорости превысил заданное значение. Значение ограничено допустимым диапазоном. Эта ошибка не должна появляться при нормальной работе. Источниками ошибки могут быть задание неправильного типа двигателя при конфигурации аппаратного обеспечения или слишком большие значения коэффициента обратной связи по скорости. Если тип двигателя в конфигурации задан правильно, то следует уменьшить коэффициент обратной связи по скорости. Если проблема продолжает существовать или если коэффициент обратной связи по скорости становится слишком малым для данного применения, то следует обратиться к изготовителю.
D7	Только статус	Переполнение G.S. при Alpha-расчете	Ось	Внутренний расчет обратной связи по скорости превысил заданное значение. Значение ограничено допустимым диапазоном. Эта ошибка не должна появляться при нормальной работе. Уменьшите коэффициент обратной связи по скорости. Если проблема продолжает существовать, то обратитесь к изготовителю.
D8	Только статус	Переполнение при расчете интегрального коэффициента	Ось	Интегральный коэффициент обратной связи по току превысил заданное значение. Расчет ограничен допустимым диапазоном. Эта ошибка не должна появляться при нормальной работе. Если ошибка сохранилась, то обратитесь к изготовителю.

D9	Только статус	Переполнение при расчете Kp	Ось	Пропорциональный коэффициент обратной связи по току вышел за пределы заданного диапазона. Расчет ограничен допустимым диапазоном. Эта ошибка не должна появляться при нормальной работе. Если ошибка сохранилась, то обратитесь к изготовителю.
DA	Быстрый останов	Обнаружена ошибка FPGA (вентильная матрица с эксплуатационным программированием)	Ось	Была обнаружена ошибка при инициализации FPGA (вентильная матрица с эксплуатационным программированием). Эта ошибка не должна появляться при нормальной работе. Если ошибка сохранилась, то обратитесь к изготовителю.
<b>Специфические ошибки</b>				
E2	Быстрый останов	Ошибка прерывания цифровой обработки сигналов	Модуль	Обратитесь в GE Fanuc Automation
<b>Ошибки линейного ускорения / замедления следящего устройства</b>				
E8	Только статус	Расстояние регистрации слежения (из регистра параметров) выходит за пределы разрешенного диапазона – следящее устройство останавливается с использованием линейного изменения скорости.	Ось	Если задано Follower Disable Action (Операция выключения слежения) = Incremental Position (Инкрементная позиция), то инкрементное расстояние (расстояние регистрации), заданное в соответствующем регистре параметров, должно быть больше расстояния остановки. Расстояние остановки зависит от действующей в данный момент скорости ведомой оси и от значения изменения скорости следящего устройства. <b>Движение ведомой оси в отрицательном направлении требует задания отрицательных значений расстояния регистрации.</b>
E9		Зарезервировано – не используется в DSM314		
EA	Только статус	Ведущая скорость превышает 0.8 x "предел скорости" – компенсация расстояния не происходит	Ось	Ведущая скорость, будучи преобразованной в единицы ведомой оси, превышает 0.8 x "skonфигурированный предел скорости". Следует либо увеличить предел скорости, либо уменьшить значение ведущей скорости.
EB	Быстрый останов	Ошибка в расчете в ходе изменения скорости следящего устройства	Ось	Обратитесь в GE Fanuc Automation
EC	Только статус	Длительность интервала линейной компенсации следящего устройства недостаточно велика	Ось	Сконфигурированное значение длительности интервала линейной компенсации следящего устройства недостаточно велико, поэтому фактическое время компенсации оказывается больше. Необходимо увеличить время компенсации изменения скорости следящего устройства.
ED	Только статус	Нарушение предела скорости при линейном изменении скорости следящего устройства.	Ось	Линейная компенсация следящего устройства требует наличия скорости, превышающей 0.8 x "skonфигурированный предел скорости оси"; поэтому фактическая длительность компенсации превышает ее skonфигурированное значение. Следует либо увеличить предел скорости, длительность компенсации или величину линейного изменения скорости.
EE	Только статус	Нарушение ограничения по времени на участке ускорения при коррекции расстояния следящего устройства	Ось	Линейная компенсация потребовала длительности действия ускорения более 64000 интервалов дискретизации контура позиционирования. Необходимо увеличить значение линейного ускорения следящего устройства.
<b>Ошибки контура позиционирования</b>				
F1	Только статус	Достигнут предел ошибки по положению следящего устройства	Ось	Ошибка по положению достигла своего предела, и контур слежения оказывается более не привязанным с ведущей оси. Следует либо увеличить значение предела ошибки по положению, либо использовать предустановку по скорости.
F2	Только статус	Достигнута граница скорости	Ось	Сумма всех задающих входов (внутренние команды + ведущее следящее устройство + локальная логика) контура позиционирования превысила skonфигурированную границу скорости. Ось более не является привязанной по положению к командам. Следует либо уменьшить значения задаваемых скоростей, либо увеличить границу скорости.
F3	Только статус	Значение отношения слежения B = 0	Ось	Не допускаются значения отношения слежения B < 0.
F4	Только статус	Значение отношения слежения B < 0	Ось	Не допускается использование значения отношения

	статус			слежения B = 0.
<b>F5</b>	Только статус	Отношение слежения A:B > 32:1 или < 1:10000	Ось	Значения отношения слежения A:B должны быть от 32:1 до 1:10000.
<b>Внутренние ошибки</b>				
<b>FB</b>	Только статус	Длительность выполнения в контуре управления > 500 микросекунд	Ось	Обратитесь в GE Fanuc Automation
<b>FC</b>	Только статус	Длительность выполнения в контуре управления > 400 микросекунд более 5 раз подряд	Ось	Обратитесь в GE Fanuc Automation
<b>FD</b>	Быстрый останов	Ошибка системного программного управления	Ось	Обратитесь в GE Fanuc Automation
<b>FE</b>	Быстрый останов	Нераспознанный датчик положения, не поддерживается	Ось	Ошибка может свидетельствовать о неисправном кабеле датчика положения – проверьте кабель. Если с кабелем все в порядке, обратитесь в GE Fanuc Automation.

## Коды системных ошибок

Если модуль DSM обнаруживает ошибки конфигурации, программы движения или локального логического блока, то он помещает код системной ошибки в регистр кода статуса модуля (первое AI-слово). Если происходит системная ошибка, то DSM не обновляет какие-либо биты %I или %AI и не реагирует на какой бы то ни было бит %Q или на любые команды %AQ.

Так, например, бит %Q Clear Error (Сбросить ошибку) не действует на системную ошибку. Системная ошибка может быть удалена только посылкой новой конфигурации на DSM.

Приведенные далее коды системных ошибок показывают, что пользователь ввел неправильную конфигурацию DSM в программном обеспечении конфигурации/программирования. Если появляется какая-либо из этих ошибок, то следует изменить конфигурацию и сохранить новую конфигурацию в ПЛК. Любые другие ошибки формата Dxxx, Exxx или Fxxx, отличающиеся от тех, что приведены в таблице, не ожидаются; если они появятся, то о них следует сообщить в GE Fanuc Automation.

Таблица A-2. Коды системных ошибок

Код ошибки (шестнадцатеричный) (x = номер оси)	Тип системной ошибки	Описание
<b>D008</b>	Модуль	Не допускается ось 4, если оси 1, 2 = цифровой сервопривод
<b>Dx65</b>	Ось	Источник обратной связи неправильный или не поддерживается.
<b>Dx68</b>	Ось	Операция выключения слежения не поддерживается
<b>Dx69</b>	Ось	Режим Линейной Компенсации Следящего Устройства не поддерживается
<b>Dx71</b>	Ось	Неправильный тип двигателя цифрового сервопривода
<b>Dx81</b>	Ось	Командный режим аналогового сервоустройства (режим момента вращения) не поддерживается.  <b>Примечание:</b> DSM314 версии 3.0 или более поздней поддерживает режим момента вращения.

## Аварийные сообщения модуля DSM цифрового сервопривода (B0–BE)

Цифровые сервосистемы GE Fanuc  $\alpha$  и  $\beta$  имеют встроенные схемы обнаружения и защитного отключения для многих потенциально опасных условий. Таблица внизу показывает, что с модулем DSM могут быть использованы три различные модели сервоусилителей: серия  $\beta$ , серия  $\alpha$  SVU и серия  $\alpha$  SVM. Эта таблица содержит аварийные сообщения, выдаваемые конкретным сервоусилителем и соответствующий код ошибки DSM. *Ячейка таблицы, остающаяся пустой в колонке усилителя, показывает, что данное аварийное сообщение не поддерживается в рассматриваемой серии усилителей.* Чтобы сбросить аварийные сообщения сервосистемы, выполните цикл выключения – включения усилителя. Кроме того для очистки кода ошибки DSM требуется отдельная команда %Q "Clear Error" ("Сбросить ошибку"). Аварийные сообщения, которые оказались не сброшенными при цикле выключения-включения питания, будут присутствовать на модуле DSM. В конце таблицы ошибок имеется небольшой раздел по поиску и устранению неисправностей при аварийных сообщениях сервосистемы.

Таблица А-3. Аварийные сообщения цифрового сервопривода DSM

Номер ошибки (шестнадцатеричный)	Название аварийного сообщения сервосистемы	Описание	Отображение аварийного сообщения усилителя		
			SVM 7 SEG	SVU 7 SEG	$\beta$ ALM LED
B0	HV	Перенапряжение DC LINK	07 <sup>†</sup>	1	ON
B1	LV	Пониженное напряжение питания в цепи управления	06 <sup>†</sup>	2	
B2	DBRLY	Неисправность динамической цепи торможения † SVM PSM DC LINK низкий заряд	05 <sup>†</sup>	7	
B3	LVDC	Пониженное напряжение DC LINK	04 <sup>†</sup>	3	ON
B4	OH	Перегрев усилителя	03 <sup>†</sup>		ON
B5	FAL	Неисправность вентилятора охлаждения	02 <sup>†</sup>		ON
B6		† SVM PSM IPM тревога или повышенный ток	01 <sup>†</sup>		
B7	DCSW DCOH	Цепь рекуперации – аварийное сообщение о неисправности Цепь рекуперации – аварийное сообщение о торможении	08 <sup>†</sup>	4 5	ON
B9	LV5V	SVM модуль сервопривода +5 В низкое	2		
BA	IPML IPMM IPMN IPMLM IPMMN IPMNL IPMLMN	IPM – повышенный ток, высокая температура или низкое напряжение  (L-ось, M-ось, N-ось, L & M-оси, M & N-оси, N & L-оси или L & M & N-оси)	8. 9. A. b. C. d. E.	8. 9. A. b. C. d. E.	
BB	LVDC	SVM модуль сервопривода, низкое DC LINK	5		
BD	FAL	SVM модуль сервопривода, неисправность вентилятора	1		
BE	HCL HCM HCN HCLM HCMN HCNL HCLMN	Аномально высокий ток двигателя  (L-ось, M-ось, N-ось, L & M-оси, M & N-оси, N & L-оси или L & M & N-оси)	8 9 A b C d E	8 9 A b C d E	ON

<sup>†</sup> Дисплей на модуле источника питания SVM показывает состояние его аварийных сообщений.

## Аварийные сообщения цифрового сервопривода – поиск неисправностей

Приведенные ниже рекомендации предназначены для помощи в решении проблем, связанных с различными аварийными сообщениями сервоприводов. Если позиции, указанные далее, не описывают ваш случай или не решают проблему с аварийным сообщением, то замените сервоусилитель или позвоните по горячей линии GE Fanuc, что получить поддержку. Соответствующие руководства по техническому обслуживанию и техническое описание для усилителя и двигателя более детально представляют операции поиска неисправностей.

**Аварийные сообщения, связанные с высоким напряжением.** Это аварийное сообщение появляется, когда уровень высокого постоянного напряжения (DC LINK) чрезмерно возрастает.

1. Переменное напряжение, подаваемое на усилитель, может быть выше номинального входного напряжения. Трехфазное напряжение питания усилителя серии  $\beta$  должно быть 200 – 240 В переменного тока.

Внешний регенеративный резистор может оказаться подключенным неправильно. Внимательно проверьте его подключение к усилителю. Проверьте, что сопротивление регенеративного резистора не отличается от номинала более чем на 20%. Замените узел регенерации, если сопротивление выходит за указанные границы.

2. Регенеративный резистор может оказаться не в состоянии рассеять избыток напряжения. Если требуется, проведите расчеты для выбора регенеративного тормозного блока и поставьте резистор большей номинальной мощности. Уменьшение значений ускорения / торможения и коэффициентов усиления контура позиционирования (более высокое значение *постоянной времени контура позиционирования*) позволяет дополнительно снизить уровни вырабатываемого напряжения.

**LVDC (пониженное напряжение связи постоянного тока):** Это аварийное сообщение появляется, когда уровень высокого постоянного напряжения (DC LINK) становится слишком низким.

1. Переменное напряжение, подаваемое на усилитель, может отсутствовать или быть меньше номинального входного напряжения. Трехфазное напряжение питания усилителя серии  $\beta$  должно быть 200 – 240 В переменного тока. Убедитесь, что на входных клеммах усилителя (L1, L2 и L3) имеется должное переменное напряжение.

**DCOH или DCSW (аварийное сообщение регенерации):** Аварийное сообщение DCOH выдается, когда температура регенерационного резистора становится слишком высокой. Аварийное сообщение DCSW информирует о проблемах на участке переключения схемы регенерации.

1. Если внешний резистор регенерации *не* используется, то проверьте, что вход датчика температуры на усилителе замкнут накоротко или на него установлена перемычка. На разъем CX11-6 должна быть установлена перемычка T604 для усилителя серии  $\beta$ .
2. Внешний регенеративный резистор может оказаться подключенным неправильно. Внимательно проверьте его подключение к усилителю. Проверьте, что сопротивление температурного датчика регенерационного резистора составляет почти ноль ом при комнатной температуре. Замените регенеративный резистор, если датчик температуры показывает разомкнутое состояние.
3. Регенеративный резистор может оказаться не в состоянии рассеять избыток напряжения. Если требуется, проведите расчеты для выбора регенеративного тормозного блока и поставьте резистор большей номинальной мощности.

Уменьшение значений ускорения / торможения и коэффициентов усиления контура позиционирования (более высокое значение *постоянной времени контура позиционирования*) позволяет дополнительно снизить уровни вырабатываемого напряжения.

**ОН (аварийное сообщение перегрева):** Повышенная температура теплоотвода усилителя или двигателя.

1. Окружающая температура может быть слишком высокой, в этом случае следует использовать охлаждающий вентилятор для серводвигателя. GE Fanuc поставляет вентиляторы для установки на большинство двигателей FANUC.
2. Эксплуатация двигателя может происходить с нарушением ограничений рабочего цикла. Рассчитайте длительность охлаждения, основываясь на кривых рабочего цикла, имеющихся для данного конкретного двигателя.
3. Двигатель может быть перегружен. Проверьте наличие избыточного трения или каких-либо препятствий на станке.
4. При всех вышеприведенных проблемах следует дать остыть усилителю в течение десяти минут с минимальной нагрузкой двигателя или вообще без нагрузки, затем выключить и включить усилитель.

**FAL (аварийное сообщение вентилятора):** Неисправность вентилятора охлаждения.

1. Проверьте вентилятор на посторонние предметы или грязь. Отключив напряжение питания от усилителя, попробуйте провернуть вентилятор вручную.
2. В усилительных системах типа SVM по охлаждающему вентилятору имеется как в модуле питания (PSM), так и в модуле усилителя сервопривода. Код аварийного сообщения показывает, какой блок неисправен.
3. Некоторые усилители имеют вентиляторные блоки, которые можно заменять прямо в рабочих условиях. Если блок вентилятора на замену отсутствует, то следует заменить весь усилитель.

**НС, НСL и т.д. (аварийное сообщение большого тока):** Чрезмерно большой ток двигателя. Для усилителей серии  $\alpha$  добавка (L, M, N и т.д.) показывает, для какой оси выдано аварийное сообщение.

1. Может иметь место замыкание обмотки двигателя (U, V, W) на землю или неправильное подключение к фазе. Проверьте обмотки и соединения. Проверьте серводвигатель на предмет короткого замыкания на корпус двигателя. Если короткое замыкание имеется, то замените двигатель.
2. Может быть сконфигурирован неправильный код типа двигателя или заданы чрезмерные значения для параметров настройки. Убедитесь, что сконфигурирован правильный двигатель и уменьшите значения коэффициентов.
3. Руководство по эксплуатации усилителя описывает процедуру контроля сигналов тока двигателя (IR и IS). Если сигналы имеют аномальную форму, замените усилитель. Если имеет место повышенный уровень шумов, проверьте заземление и, в частности, заземление экрана кабеля команд (K1) к усилителю.
4. Эксплуатация двигателя может происходить с нарушением ограничений рабочего цикла. Рассчитайте длительность охлаждения, основываясь на кривых рабочего цикла, имеющихся для данного конкретного двигателя.
5. Двигатель может быть перегружен. Проверьте наличие избыточного трения или каких-либо препятствий на станке.
6. При всех вышеприведенных проблемах следует дать остыть усилителю в течение десяти минут с минимальной нагрузкой двигателя или вообще без нагрузки, затем выключить и включить усилитель.

**LV (аварийное сообщение пониженного напряжения питания в цепи управления):** Слишком низкое напряжение, используемое для питания низковольтных цепей усилителя.

1. Усилители серии  $\alpha$  типа SVU поставляются с установленными по умолчанию переключателями так, чтобы использовать для питания усилителя однофазное переменное напряжение 220 В. Дополнительно пользователь может удалить переключатели и подключить 220 В питающего переменного напряжения отдельно. Проверьте, что на клеммах L1C и L2C при установке по умолчанию или на разъеме CX3 (Y-ключ) при отдельной подаче напряжения питания имеется, как минимум, 200 В переменного напряжения.
2. Проверьте предохранитель усилителя. Если предохранитель вышел из строя, замените его новым после проверки напряжения питания. Если второй предохранитель перегорает, то замените усилитель.

**DBRLY (неисправность реле динамического торможения):** Это аварийное сообщение показывает, что контакты тормозного реле сварились. Немедленно замените усилитель.

**IPML, IPMM и т.д. (аварийное сообщение IPM):** Микропроцессорный модуль питания (IPM) представляет собой сильноточное коммутационное устройство, установленное в усилителе. Модуль IPM может обнаружить повышенный ток, перегрев или пониженное напряжение в цепях электропитания. Добавка (L, M, N и т.д.) показывает, для какой оси выдано аварийное сообщение.

1. Может иметь место замыкание обмотки двигателя (U, V, W) на землю или неправильное подключение к фазе. Проверьте обмотки и соединения. Проверьте серводвигатель на предмет короткого замыкания на корпус двигателя. Если короткое замыкание имеется, то замените двигатель.
2. Может быть сконфигурирован неправильный код типа двигателя или заданы чрезмерные значения для параметров настройки. Убедитесь, что сконфигурирован правильный двигатель и уменьшите значения коэффициентов.
3. Руководство по эксплуатации усилителя описывает процедуру контроля сигналов тока двигателя (IR и IS). Если сигналы имеют аномальную форму, замените усилитель. Если имеет место повышенный уровень шумов, проверьте заземление и, в частности, заземление экрана кабеля команд (K1) к усилителю.
4. Эксплуатация двигателя может происходить с нарушением ограничений рабочего цикла. Рассчитайте длительность охлаждения, основываясь на кривых рабочего цикла, имеющихся для данного конкретного двигателя.
5. Двигатель может быть перегружен. Проверьте наличие избыточного трения или каких-либо препятствий на станке.
6. При всех вышеприведенных проблемах следует дать остыть усилителю в течение десяти минут с минимальной нагрузкой двигателя или вообще без нагрузки, затем выключить и включить усилитель.

## Светодиодные индикаторы

На модуле DSM314 имеется семь светодиодных индикаторов, которые обеспечивают индикацию статуса (состояния) модуля. Эти индикаторы описаны ниже.

**STAT** Нормально ВКЛ. МИГАЕТ для индикации ошибок при работе. Мигает *медленно* (четыре раза в секунду) – при ошибках только статуса. *Быстрое* мигание (восемь раз в секунду) – при ошибках, которые вызывают прекращение работы сервопривода.

**ON (ВКЛ.):** Если этот светодиодный индикатор постоянно в состоянии ВКЛ., то это значит, что модуль DSM314 функционирует правильно. В нормальной ситуации этот индикатор должен быть всегда в состоянии ВКЛ.

**OFF (ВЫКЛ.):** Если светодиодный индикатор в состоянии ВЫКЛ., то это значит, что модуль DSM314 не функционирует. Это является результатом неисправности аппаратуры или программного обеспечения, что не позволяет включить модуль.

**Мигание:** Если светодиодный индикатор МИГАЕТ, это означает состояние неисправности (ошибки).

**Постоянная частота мигания, индикатор CFG в состоянии ВКЛ.:**

Индикатор мигает медленно (четыре раза в секунду) при ошибках только статуса и быстро (восемь раз в секунду) при ошибках, которые вызывают прекращение работы сервопривода. Бит статуса *Module Error Present (Имеется ошибка модуля) %I* будет в состоянии ВКЛ. Код ошибки (шестнадцатеричный формат) помещается в слово *Module Status Code (Код Статуса Модуля) %AI* или в одно из слов *Axis Error Code (Код Ошибки Оси) %AI*.

**Постоянная частота мигания, индикатор CFG мигает:**

Если светодиодные индикаторы STAT и CFG оба мигают **вместе** с постоянной скоростью, то это значит, что модуль DSM314 в режиме начальной загрузки ожидает загрузку новой микропрограммы. Если индикаторы STAT и CFG оба мигают **попеременно** с постоянной скоростью, то это значит, что микропрограмма DSM314 обнаружила истечение времени ожидания при контроле программного обеспечения, как результат неправильного функционирования аппаратуры или программ.

**Меняющаяся нерегулярным образом частота мигания, индикатор CFG в состоянии ВЫКЛ.:**

Если это происходит сразу же при включении, то это означает, что обнаружена неисправность аппаратуры или ошибка программы. Индикатор STAT на модуле будет мигать, отображая два номера ошибки, разделенных короткой задержкой. Эти номера определяются подсчетом миганий в обеих последовательностях. Запишите эти номера и обратитесь в GE Fanuc за информацией по решению проблемы.

**OK** Светодиодный индикатор ОК показывает текущее состояние (статус) модуля DSM314.

**ON (ВКЛ.):** Если индикатор постоянно в состоянии ВКЛ., то это значит, что модуль DSM314 функционирует правильно. В нормальной ситуации этот светодиод должен быть всегда в состоянии ВКЛ.

**OFF (ВЫКЛ.):** Если светодиодный индикатор в состоянии ВЫКЛ., то это значит, что модуль DSM314 не функционирует. Это является результатом неисправности аппаратуры или программного обеспечения, что не позволяет включить модуль.

**CFG** Этот индикатор ВКЛ., когда конфигурация модуля получена от ПЛК.

- EN1** Если этот индикатор ВКЛ., то это значит, что включен выход реле Включения привода оси 1.
- EN2** Если этот индикатор ВКЛ., то это значит, что включен выход реле Включения привода оси 2.
- EN3** Если этот индикатор ВКЛ., то это значит, что включен выход реле Включения привода оси 3.
- EN4** Если этот индикатор ВКЛ., то это значит, что включен выход реле Включения привода оси 4.



## Команды запросов на обмен данными модуля DSM314

---

---

Данное приложение описывает два типа коммуникационных запросов (далее в данном приложении используется сокращение COMM REQ) инструкции лестничной логике используемых с модулем DSM314.

- **Тип "Загрузка параметров".** Используется для загрузки параметров в память модуля DSM. Достоинством команды COMM REQ является то, что каждая команда может загрузить до 16 параметров и что в одном цикле ПЛК может быть использовано несколько команд COMM REQ. Для сравнения: каждая команда немедленного действия Load Parameter (Загрузить параметр) может загрузить за один цикл ПЛК только один параметр; при этом в одном цикле ПЛК может быть от одной до четырех таких команд в зависимости от количества сконфигурированных слов %AQ (которое, в свою очередь, зависит от количества сконфигурированных осей – см. табл. 5-8). Поэтому команда COMM REQ является чрезвычайно полезной для загрузки множества параметров, а команда немедленного действия Load Parameter (Загрузить параметр) – для загрузки небольшого количества параметров (от одного до четырех).
- **Тип "Таблица пользовательских данных" (UDT)** Используется для доступа к Таблице пользовательских данных Local Logic модуля DSM314. Таблицей пользовательских данных является область памяти размером 8192 байта, которую программы Local Logic могут использовать для хранения и извлечения данных. UDT COMM REQ может копировать данные либо из памяти с пословной выборкой ПЛК в UDT, либо из UDT в эту память.

В общем случае, COMM REQ используется в программе лестничной логике ПЛК Series 90-30 для обмена данными с множеством многопроцессорных модулей. В данном приложении в разделах 1 и 2 обсуждаются общие вопросы, касающиеся команды COMM REQ. Затем в разделах 3 – 5 обсуждается конкретное использование этой команды в модуле DSM314. Приложение имеет следующую структуру:

- Раздел 1: Краткое описание команд запросов на обмен данными
- Раздел 2: Команда COMM REQ Ladder
- Раздел 3: Команда COMM REQ "Таблица пользовательских данных" (UDT)
- Раздел 4: Команда COMM REQ "Загрузка параметров"
- Раздел 5: Пример COMM REQ Ladder Logic (используется COMM REQ "Загрузка параметров")

## Раздел 1: Краткое описание команд запросов на обмен данными

Запрос на обмен данными использует параметры команды *COMM REQ Ladder* и связанный командный блок, чтобы задать характеристики запроса. Связанное *Status Word* (Слово статуса) содержит результаты каждого запроса.

### Структура запроса на обмен данными

Запрос на обмен данными состоит из трех главных частей:

- Команда *COMM REQ Ladder*
- Командный блок, который является блоком памяти ПЛК (обычно память %R); этот блок содержит команды и данные для *COMM REQ*.
- Слово статуса, которое представляет собой одно слово памяти, в которое записываются коды статуса / ошибки.

Рисунок далее показывает взаимодействие этих частей:

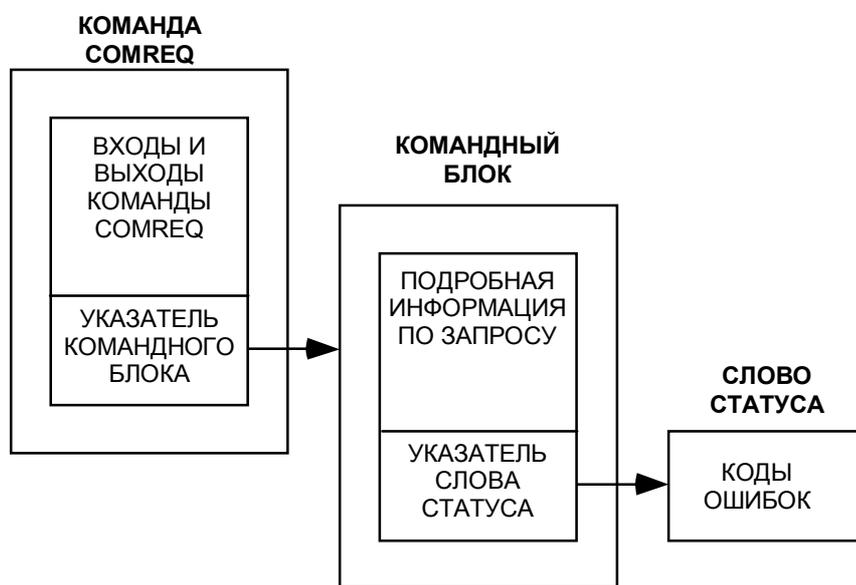


Рис. В-1. Структура *COMM REQ*

**Команда *COMM REQ Ladder*.** Команда *COMM REQ Ladder* является главной структурой, используемой для задания конкретной информации о запросе на обмен данными. Эта информация включает в себя расположение крейта и слота модуля DSM, связанного с данным запросом, и параметр, указывающий начальный адрес командного блока. Следует отметить, что при программировании этой команды данные командного блока должны быть инициализированы в программе лестничной логики до того, как будет выполнено звено, содержащее команду *COMM REQ*.

**Командный блок.** Командный блок состоит из нескольких слов памяти ПЛК, которые содержат дополнительную информацию о запросе на обмен данными. Эта информация включает в себя временные параметры, указатель на слово статуса, блок данных, типы и размеры памяти, а также специфические командные коды. Блок данных задает направление передачи данных (через код команды), а также расположение и тип передаваемых данных.

**Слово статуса.** Слово статуса является отдельной зоной памяти ПЛК, куда ЦП передает результат запроса на обмен данными. Адрес слова статуса задан в командном блоке пользователем. Приведенная далее таблица содержит коды статуса, передаваемые в слово статуса.

Таблица В-1. Коды слова статуса DSM COMM REQ

Коды слова статуса DSM COMM REQ			
Название кода	Номер кода	Описание	Возможные действия по исправлению ситуации
IOB_SUCCESS	1	Весь обмен данными прошел нормально.	Не требуется.
IOB_PARITY_ERR	-1	Произошла ошибка четности во время обмена данными с крейтом расширения.	Сделайте еще одну попытку. Проверьте оборудование – кабели, модуль DSM и т.д.
IOB_NOT_COMPL	-2	После завершения обмена данными модуль не показал его окончание.	Сделайте еще одну попытку. Проверьте параметры COMM REQ.
IOB_MOD_ABORT	-3	Модуль оборвал обмен данными.	Сделайте еще одну попытку. Проверьте параметры COMM REQ.
IOB_MOD_SYNTAX	-4	Модуль показал, что отправленные данные имеют неправильную последовательность.	Проверьте параметры COMM REQ.
IOB_NOT_RDY	-5	Бит RDY в статусе модуля был не включен.	Сделайте еще одну попытку. Проверьте модуль DSM.
IOB_TIMEOUT	-6	Прошел максимальный интервал времени для получения отклика от модуля, но отклик не получен.	Проверьте модуль DSM. Проверьте параметры COMM REQ.
IOB_BAD_PARAM	-7	Один из переданных параметров был неправильным.	Проверьте параметры COMM REQ.
IOB_BAD_CSUM	-8	Контрольная сумма, полученная из модуля протокола DMA, не соответствует полученным данным.	Сделайте еще одну попытку. Проверьте правильность заземления, экранирования, подавление шума и т.п.
IOB_OUT_LEN_CHGD	-9	Выходная длина для модуля была изменена, поэтому нормальная обработка записи отклика не может быть выполнена.	Проверьте параметры COMM REQ.

#### Корректирующие действия

Тип корректирующих действий, которые следует предпринять, зависит от конкретного применения. Если ошибка возникает на стадии запуска или отладки при разработке программы лестничной логике, то представляется уместной рекомендация "Проверьте параметры COMM REQ". То же справедливо, если ошибка происходит сразу же после изменения программы. Однако, если ошибка возникает на проверенном применении, которое до этого успешно работало, то наиболее вероятно, что ошибка связана с оборудованием. Для получения возможной дополнительной информации следует проверить таблицы неисправностей ПЛК, когда происходит поиск причин ошибок, отображаемых в слове статуса.

## Мониторинг слова статуса

### Обнаружение ошибок и работа с ними

Как показано в таблице вверху, при нормальном проведении обмена данными в слово статуса возвращается 1; однако при появлении любой ошибки возвращается отрицательное значение. Если в программе лестничной логике требуется обнаружение ошибки, то следует использовать команду сравнения Less Than (LT) (Меньше чем), чтобы определить, когда слово статуса станет отрицательным (меньше нуля). Пример этого показан на рисунке далее. При возникновении ошибки на выходе Less Than (LT) (Меньше чем), (Q) появляется сигнал. Для подключения дальнейшей обработки ошибки или логики сообщения об ошибке может быть использовано реле или подобный элемент, срабатывающий от этого выхода.



На FT-выход COMM REQ, описанный далее в данном приложении, при определенных ошибках подается напряжение, которое также может быть использовано для индикации этих ошибок. Кроме того, может осуществляться просмотр слова статуса логикой обнаружения сообщений об ошибках для отображения этих ошибок на интерфейсном устройстве оператора; в этом случае коды слова статуса будут преобразовываться в соответствующие сообщения об ошибках, выводимые на экран оператора. Например, если в слове статуса обнаруживается -1, то на экране может появиться сообщение вида "Ошибка обмена данными с DSM-модулем в крейте расширения".

Для динамической проверки слова статуса в него записывается не несущее значения положительное число (обычно используется 0 или 99) каждый раз перед выполнением соответствующей команды COMM REQ. Затем, если команда выполнена успешно, то ЦП запишет туда число 1. Этот метод позволяет быть уверенным, что последняя команда COMM REQ совершенно определенно была выполнена успешно, если обнаружено число 1, т.е. это число не осталось от предыдущей команды. В примере в конце данного приложения число 99 помещается в слово статуса (%R0195) на этапе, предшествующем выполнению команды COMM REQ.

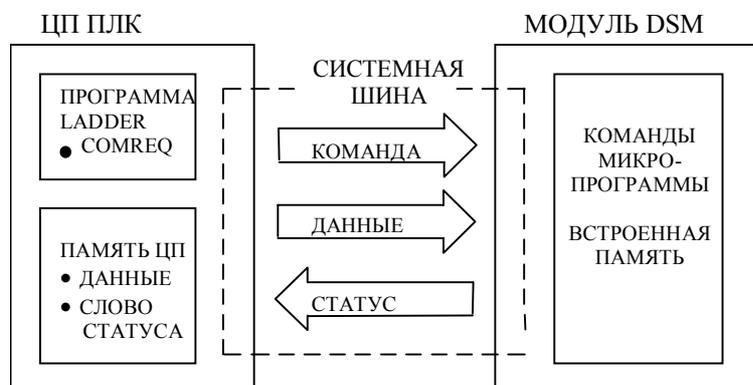
Если используется несколько DSM COMM REQ, то рекомендуется проверять успешность выполнения каждой из них до того, как перейти к следующей команде. Мониторинг слова статуса является одним из путей выполнения этого.

### Проверка получения модулем DSM правильных данных

Для критических применений можно порекомендовать проверять, что значения определенных параметров правильно переданы модулю DSM, прежде чем продолжать работу. Чтобы реализовать это, необходимо сначала запрограммировать команду немедленного действия *Select Return Data (Выбрать возвращаемые данные)* %AQ, чтобы задать номер параметра DSM, который должен быть считан в соответствующее двойное слово *User Selected Data (Данные, выбранные пользователем)* %AI (для каждой оси имеется одно двойное слово *User Selected Data* %AI). **Отметьте, что должно пройти не менее трех циклов ПЛК или 20 миллисекунд (в зависимости от того, что больше), прежде чем новые Данные, выбранные пользователем, станут доступными в ПЛК.** Это обстоятельство требует программирования некоторой логики задержки по времени с тем, чтобы обеспечить выполнение этих условий. Затем программируется команда Equal (Равно) для двойных целых чисел, чтобы сравнить значение, возвращаемое в двойном слове *User Selected Data (Данные, выбранные пользователем)*, с отправленным значением. В разделе 5 данного приложения показан пример этого. Дополнительную информацию по слову *User Selected Data (Данные, выбранные пользователем)* и по команде *Select Return Data (Выбрать возвращаемые данные)* см. в главе 5.

## Действие запроса на обмен данными

На рисунке ниже проиллюстрирован обмен информацией между ЦП ПЛК и модулем DSM.



**Рис. В-2. Действие запроса на обмен данными модуля DSM**

Запрос на обмен данными инициируется, когда при проходе ПЛК активируется команда COMM REQ. При этом элементы запроса на обмен данными, состоящие из команды и данных, передаются от ЦП ПЛК на модуль DSM.

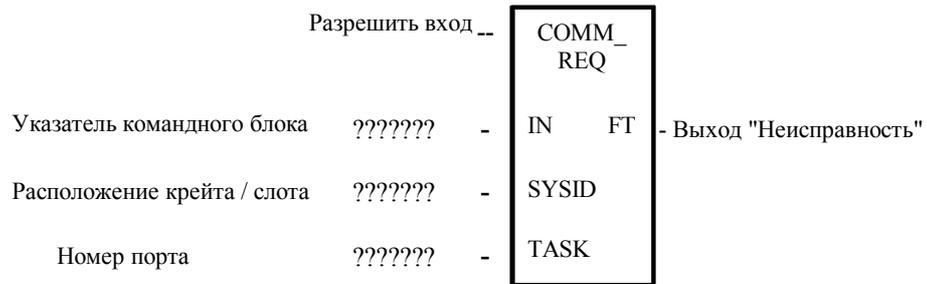
- Если имеет место COMM REQ "Загрузка параметров", то установки команды определяют, что данные должны быть считаны из памяти ПЛК и скопированы в соответствующие места памяти параметров DSM.
- Если имеет место UDT COMM REQ, то установки команды определяют, что данные должны быть считаны из памяти ПЛК и скопированы в определенный сегмент памяти UDT или что они должны быть считаны из определенного сегмента памяти UDT и скопированы в память ПЛК.

Порядок отправки этих команд имеет значение, поэтому командный блок для каждого типа COMM REQ должен быть запрограммирован точно так, как описано далее в данном приложении. На рисунке выше модуль DSM показан в крейте ЦП, и обмен данными происходит через системную шину ПЛК. Если модуль DSM располагается в расширительном или удаленном крейте, то команды и данные посылаются через системную шину крейта ЦП по кабелю удаленного управления или управления расширением на крейт, содержащий данный DSM, и далее по его системной шине на этот модуль DSM.

В завершение каждого запроса ЦП ПЛК передает статус данного запроса в слово статуса, которое является участком памяти ПЛК, назначенным указателем слова статуса в командном блоке.

## Раздел 2: Команда COMM REQ Ladder

В данном разделе обсуждаются общие вопросы, касающиеся команды COMM REQ. Дополнительная информация имеется в "Справочном руководстве по системе команд ЦП ПЛК Series 90-30/20/Micro", издание GFK-0467L или более позднее. Запрос на обмен данными начинается при активации команды COMM REQ Ladder. Команда лестничной логике COMM REQ имеет четыре входа и один выход.



**Рис. В-3. Команда COMM REQ Ladder**

Ниже дано детальное описание всех входов и выходов. Важно понимать, что указатель командного блока указывает на другую область памяти, куда требуется ввести дополнительную информацию о запросе на обмен данными.

**Enable Input (Разрешить ввод):** чтобы разрешить команду COMM REQ, должна быть логическая 1. Рекомендуется, чтобы разрешающей логикой был контакт переходной катушки ("одновибратор").

**IN:** Нахождение в памяти первого слова командного блока. Это может быть любой действительный адрес памяти с пословным обращением (%R, %AI или %AQ).

**SYSID:** Шестнадцатеричное значение, которое дает положение крейта и слота модуля, на который направлена команда COMM REQ. Старший байт (первые две цифры шестнадцатеричного номера) содержит номер крейта, а младший байт содержит номер слота. В таблице внизу даны некоторые примеры этого:

### Примеры SYSID

<u>Крейт</u>	<u>Слот</u>	<u>Значение шестнадцатеричного слова</u>
0	4	0004h
3	4	0304h
2	9	0209h

**TASK (ЗАДАЧА):** Для DSM-модуля здесь всегда должен быть установлен 0.

**FT-выход:** Функциональный FT-выход может обеспечить выход на дополнительную логику, которая может подтвердить успешное завершение запроса на обмен данными. FT-выход может иметь следующие состояния:

Таблица В-2. Таблица состояний FT-выхода команды COMM REQ

FT-выход		
Состояние Enable Input (Разрешить ввод)	Ошибка существует?	FT-выход
Активное	Нет	Низкий
Активное	Да	Высокий
Неактивное	Не исполняется	Низкий

- FT-выход устанавливается на "Высокое значение", если:
  - Указанный адрес назначения отсутствует (например, задан крейт 1, в то время как система использует только крейт 0).
  - Указанный номер задачи недействителен для устройства (для DSM номер TASK (ЗАДАЧА) всегда должен быть 0).
  - Длина данных установлена на 0.

#### Требования и рекомендации по программированию DSM COMM REQ

- Рекомендуется включать (разрешать) команды DSM COMM REQ контактом одновибратора.
- Если в программе лестничной логике используется более одной команды DSM COMM REQ, то перед началом выполнения такой команды необходимо проверять, что предыдущая была выполнена успешно. Это может быть сделано проверкой слова статуса и выхода FT (неисправность), как описано выше в данном приложении в разделе "Мониторинг слова статуса".
- Как можно видеть в таблице выше, FT-выход имеет значение логического 0 ("ложь"), если Enable Input неактивен. Это значит, что если команда COMM REQ включена контактом одновибратора и происходит неисправность (сбой, ошибка), то на FT-выходе будет высокое значение только в течение одного сканирования ПЛК. Поэтому, чтобы "захватить" сбой, FT-выход должен быть запрограммирован как "катушка с запоминанием", которая не сбрасывается автоматически в конце сканирования. В этом случае требуется дополнительная логика для сброса выхода этой катушки после квитирования сообщения о неисправности.
- Требуется дополнительное программирование устройства, такого как "катушка с запоминанием" на FT-выходе COMM REQ.
- Обратите внимание на то, что команда Series 90-30 COMM REQ (в отличие от многих других команд ПЛК Series 90-30) не имеет выхода ОК.
- До начала выполнения команды COMM REQ необходимо инициализировать данные в командном блоке. Поскольку нормальным порядком цикла ПЛК является движение сверху вниз, то инициализация командного блока на более раннем этапе (или этапах), чем этап, содержащий команду COMM REQ, облегчает выполнение этого требования. См. пример в конце данного приложения.
- Рекомендация: Если команды MOVE используются для загрузки значений в регистры командного блока, то используйте пословную команду MOVE для загрузки шестнадцатеричного номера и целочисленную команду MOVE – для загрузки десятичного номера. Пример этого можно видеть в конце данного приложения для команды COMM REQ "Загрузка параметров", где код E501h загружается при помощи пословной команды MOVE, а остальные десятичные значения загружаются при помощи целочисленных команд MOVE.

## Раздел 3: Команда *COMM REQ* "Таблица пользовательских данных" (*UDT*)

Модуль DSM314 имеет область памяти размером 8192 байта, называемую Таблицей пользовательских данных (*UDT*), которая предназначена для использования с программами Local Logic (*LL*). Программы Local Logic могут иметь доступ ко всей этой памяти или к части ее, чтобы записывать туда данные и извлекать их. *UDT* полезна для хранения и передачи больших массивов данных, таких как, например, большие пакеты установочных данных.

ЦП ПЛК может записать в *UDT* или считать из нее, используя команду запроса на обмен данными "Таблица пользовательских данных" (*UDT COMM REQ*) в многозвенной программе ПЛК. Одна команда *UDT COMM REQ* может считать или записать 2048 байтов за один раз. Поэтому *UDT* логически разделена на четыре сегмента по 2048 байт, называемых сегментами 1 – 4, доступ к которым может быть организован по отдельности через команду *UDT COMM REQ*. Для каждого из этих четырех сегментов имеется своя уникальная команда чтения и команда записи, давая, таким образом, 8 возможных команд *UDT COMM REQ*.

### **Характеристики команды *COMM REQ* "Таблица пользовательских данных" и информация по использованию**

- Чтение или запись 2 К (2048) байт за один раз в Таблице пользовательских данных Local Logic. Иное значение не разрешается.
- Работа возможна только с модулем DSM314 (с модулем DSM302 не работает)
- Не может использоваться для загрузки параметров в модуль DSM314.

Эта команда добавляет около 15 мс к длительности сканирования (к циклу) ПЛК на одно сканирование, если управляющий параметр окна цикла обмена данными ПЛК установлен на *COMPLETE* (Выполнять до завершения). Если этот параметр установлен на *LIMITED* (ОГРАНИЧЕНО), то команда *COMM REQ* будет выполнена в течение нескольких циклов сканирования так, чтобы минимально влиять на время сканирования. Однако эта команда *COMM REQ*, скорее всего, не будет выполнена повторно – она будет выполнена только в том случае, если имеется потребность изменить данные. Поэтому, если она была направлена на первом проходе сканирования или во время установки параметров работы, то она не окажет воздействия, пока задача работает.

- Чтобы избежать конфликтов доступа к памяти, рекомендуется не использовать периодическую подпрограмму в то время, когда активна команда *COMM REQ*.
- Эта команда *COMM REQ* не поддерживает дискретную память для своего типа данных ПЛК.

## Командный блок UDT COMM REQ

Таблица В-3. Командный блок Таблицы пользовательских данных

Командный блок команды COMM REQ "Таблица пользовательских данных" для модуля DSM314		
Описание	Смещение адреса	Номер слова и значение
Длина заголовка блока данных	Адрес + 0	Слово 1, всегда устанавливается на 4
Флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ)	Адрес + 1	Слово 2, всегда устанавливается на 0
Тип памяти слова статуса (см. таблицу кодов типов памяти слова статуса далее)	Адрес + 2	Слово 3, выбирается пользователем (см. таблицу кодов типов памяти далее)
Смещение указателя статуса	Адрес + 3	Слово 4, выбирается пользователем
Значение времени ожидания	Адрес + 4	Слово 5, всегда устанавливается на 0
Максимальная длительность обмена данными	Адрес + 5	Слово 6, всегда устанавливается на 0
Код команды	Адрес + 6	Слово 7, см. таблицу кодов команд
Размер данных параметров, байт	Адрес + 7	Слово 8, всегда 2048.
Тип памяти для данных ПЛК	Адрес + 8	Слово 9, выбирается пользователем (см. таблицу кодов типов памяти далее)
Запуск данных ПЛК (Смещение данных)	Адрес + 9	Слово 10, выбирается пользователем

**Длина блока данных (слово 1).** Длина части заголовка блока данных в командном блоке. Его следует установить на 4. Заголовок блока данных хранится в словах 7 – 10 командного блока.

**Флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ) (слово 2).** Для DSM он всегда должен быть установлен на логический **ноль**.

**Тип памяти слова статуса (слово 3).** Это слово определяет тип памяти, который будет использован для слова статуса. Каждый тип памяти имеет свой собственный специфический номер кода, показанный в таблице кодов типов памяти далее. Так, например, если есть желание использовать для слова статуса память %R, то следует ввести в это слово либо десятичный номер кода 8, либо шестнадцатеричный номер кода 08h.

Следует отметить, что если выбирается дискретный тип памяти (%I или %Q), то для слова памяти назначается 16 последовательных битов, начиная с адреса, заданного в слове смещения указателя слова статуса, приведенного ниже.

**Таблица В-4. Коды типов памяти слова статуса**

Коды типов памяти слова статуса COMM REQ			
Краткое обозначение типа памяти	Тип памяти	Номер кода, подлежащий вводу	
		Десятичный	Шестнадцатеричный
%I	Таблица дискретных входов	70	46h
%Q	Таблица дискретных выходов	72	48h
%R	Память регистров	8	08h
%AI	Таблица аналоговых входов	10	0Ah
%AQ	Таблица аналоговых выходов	12	0Ch

**Смещение указателя слова статуса (слово 4).** Это слово содержит смещение для выбора типа памяти. *Примечание: смещение указателя слова статуса является числом, базирующимся на нуле.* С практической позиции это означает, что требуется вычитать единицу из номера адреса, который требуется задать. Например, чтобы выбрать %R0001, следует ввести нуль ( $1 - 1 = 0$ ). Или, если требуется задать %R0100, то введите 99 ( $100 - 1 = 99$ ). Следует заметить, что тип памяти (в данном примере это %R) задан в предыдущем слове (см. пояснение выше "Тип памяти указателя слова статуса").

**Значение времени ожидания (слово 5).** Поскольку DSM всегда использует режим NOWAIT (НЕ ЖДАТЬ) (флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ) всегда установлен на ноль), то этот параметр (Значение времени ожидания) не работает в DSM. Установите его в нуль.

**Максимальная длительность обмена данными (слово 6).** Поскольку DSM всегда использует режим NOWAIT (НЕ ЖДАТЬ) (флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ) всегда установлен на ноль), то этот параметр (Максимальная длительность обмена данными) не работает в DSM. Установите его в нуль.

**Код команды (слово 7).** Используйте один из восьми кодов команды из таблицы ниже. Коды команд даны в шестнадцатеричном виде.

**Таблица В-5. Коды команд UDT COMM REQ**

Команды COMM REQ "Таблица пользовательских данных" (UDT)	
Код команды	Описание команды
D001h	Выполнить запись в сегмент 1 UDT
D101h	Выполнить запись в сегмент 2 UDT
D201h	Выполнить запись в сегмент 3 UDT
D301h	Выполнить запись в сегмент 4 UDT
D804h	Выполнить чтение из сегмента 1 UDT
D904h	Выполнить чтение из сегмента 2 UDT
DA04h	Выполнить чтение из сегмента 3 UDT
DB04h	Выполнить чтение из сегмента 4 UDT

**Размер сегмента данных UDT (слово 8).** Определяет размер памяти в байтах сегмента UTP, к которому обеспечивается доступ. Это значение всегда должно быть равно 2048 байт (800h в шестнадцатеричном виде).

**Тип памяти данных ПЛК (слово 9).** Это слово определяет тип памяти, который будет использован для данных ПЛК. Каждый тип памяти имеет уникальный номер

кода, показанный в таблице кодов типов памяти далее. Так, например, для задания памяти %R, следует ввести в это слово либо десятичный номер кода 8, либо шестнадцатеричный номер кода 08h.

### Примечание

UDT COMM REQ не поддерживает дискретную память (%I или %Q) для типа памяти данных ПЛК.

Таблица В-6. Коды типов памяти данных ПЛК для UDT COMM REQ

Коды типов памяти данных ПЛК для UDT COMM REQ			
Краткое обозначение типа памяти	Тип памяти	Номер кода, подлежащий вводу	
		Десятичный	Шестнадцатеричный
%R	Память регистров	8	08h
%AI	Таблица аналоговых входов	10	0Ah
%AQ	Таблица аналоговых выходов	12	0Ch

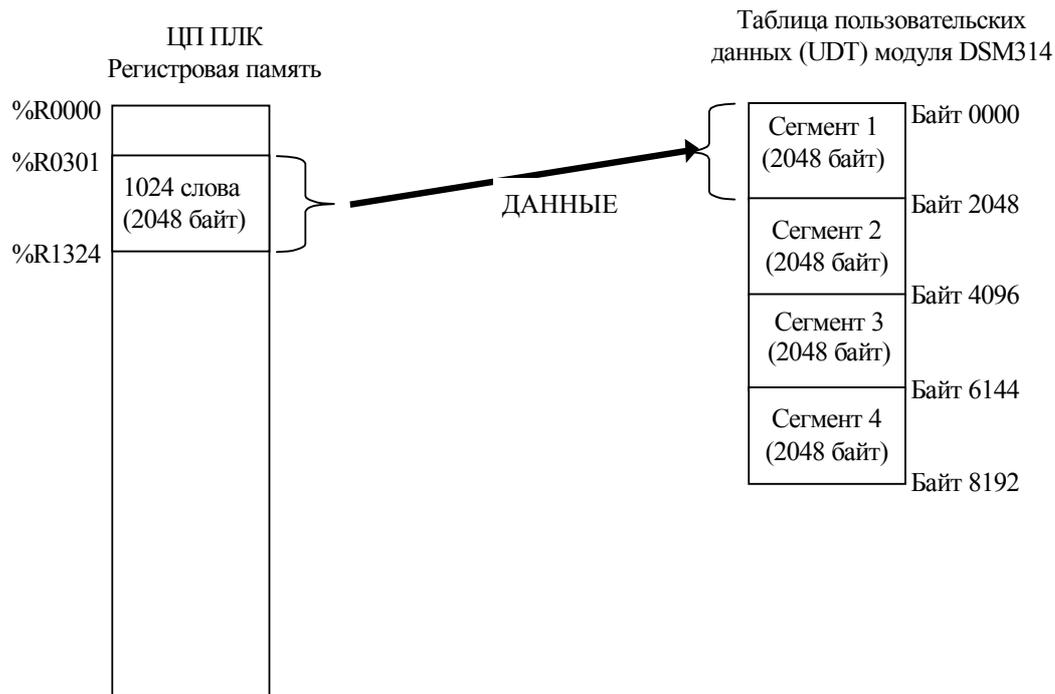
**Смещение начального указателя данных ПЛК (слово 10).** Это слово содержит смещение для типа памяти, выбранного в слове типа памяти данных ПЛК (слово 9).

**Примечание:** смещение начального указателя данных ПЛК является числом, базирующимся на нуле. С практической позиции это означает, что требуется вычитать единицу из номера адреса, который требуется задать. Например, чтобы выбрать %R0001 в качестве начальной точки данных ПЛК, следует ввести нуль ( $1 - 1 = 0$ ). Или, чтобы выбрать %R0100, введите 99 ( $100 - 1 = 99$ ). Следует заметить, что тип памяти (в данном примере это %R) задан в предыдущем слове. Начальным адресом, назначенным этим словом, будет первое из 1024 следующих друг за другом слов памяти ПЛК, используемой в COMM REQ.

### Пример команды COMM REQ "Таблица пользовательских данных"

В данном примере используются следующие характеристики:

- Модуль DSM314 смонтирован в крейте 0, слот 7 ПЛК.
- Начальный адрес командного блока %R0196.
- Слово статуса располагается в %R0195.
- FT-выход (неисправность) команды COMM REQ вызывает срабатывание "катушки с запоминанием".
- Запись должна осуществляться в сегмент 1 Таблицы пользовательских данных DSM314. Это определяется кодом команды D001 в слове 7 командного блока.
- Данные размером 1024 слова (2048 байт) регистровой памяти ПЛК (с %R0301 по %R1324) копируются и записываются в сегмент 1 (2048 байт) Таблицы пользовательских данных. (Обратите внимание, что каждое слово %R имеет длину два байта). Эта передача данных иллюстрируется рисунком далее.



**Рис. В-4. Передача данных для кода команды D001 (запись в сегмент 1)**

### Пример команды COMM REQ "Таблица пользовательских данных"

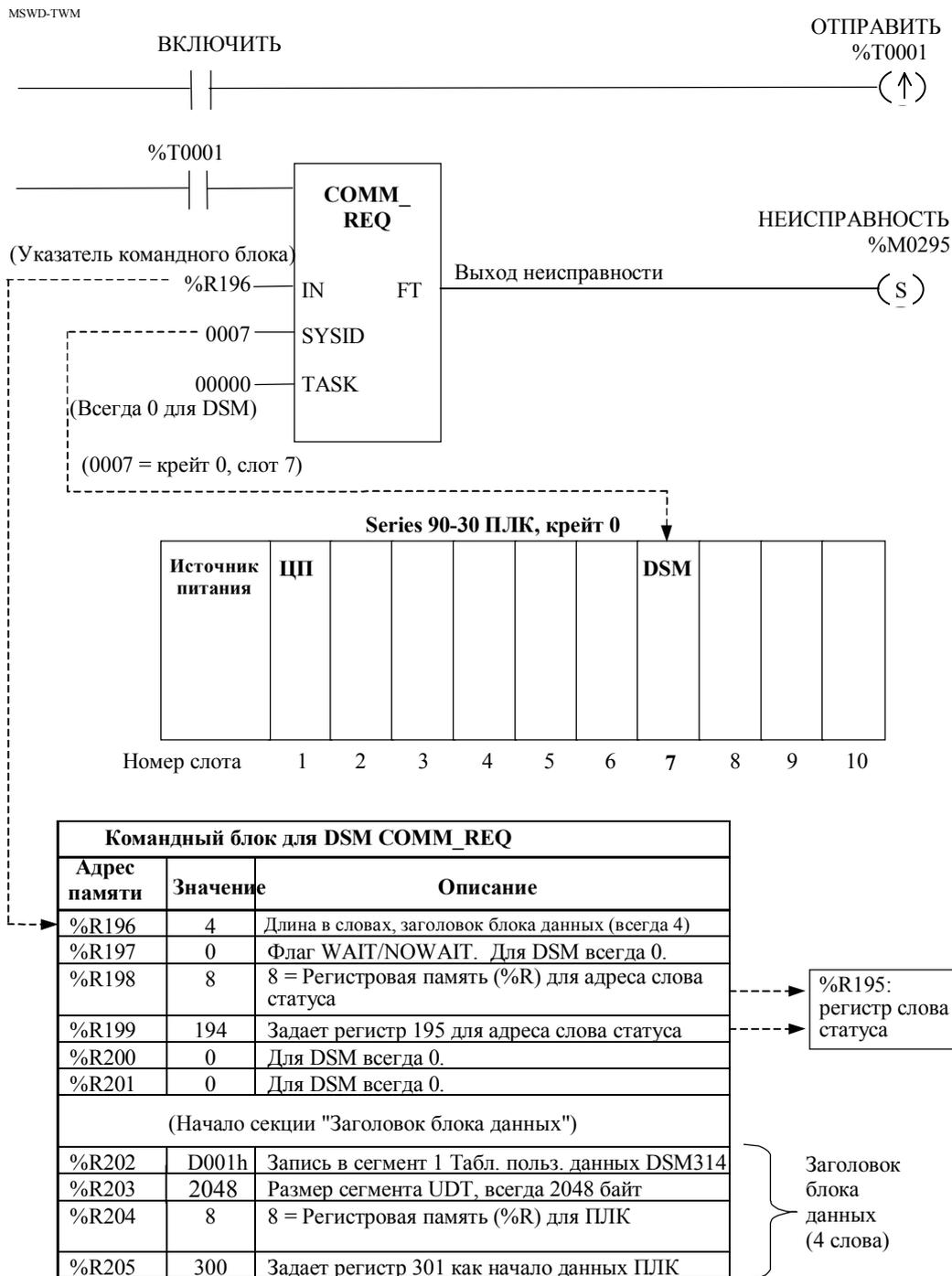


Рис. В-5. Пример UDT COMM REQ для модуля DSM314

## Раздел 4: Команда COMM REQ "Загрузка параметров"

### Командный блок

Командный блок содержит подробную информацию о запросе на обмен данными. Первый адрес командного блока определен во входе IN команды COMM REQ Ladder. Этот адрес может быть в любой области памяти с пословным обращением (%R, %AI или %AQ). Структура командного блока может быть расположена в назначенной области памяти при помощи соответствующей команды программирования (рекомендуется команда BLOCK MOVE). Командный блок модуля DSM имеет следующую структуру:

**Таблица В-7. Командный блок команды COMM REQ "Загрузка параметров" модуля DSM**

Командный блок команды COMM REQ "Загрузка параметров" модуля DSM		
Описание	Адрес + Смещение	Номер слова и значение
Длина заголовка блока данных	Адрес + 0	Слово 1, всегда устанавливается на 4
Флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ)	Адрес + 1	Слово 2, всегда устанавливается на 0
Тип памяти указателя статуса (см. таблицу кодов типов памяти далее)	Адрес + 2	Слово 3, выбирается пользователем (см. таблицу кодов типов памяти)
Смещение указателя статуса	Адрес + 3	Слово 4, выбирается пользователем
Значение времени ожидания	Адрес + 4	Слово 5, всегда устанавливается на 0
Максимальная длительность обмена данными	Адрес + 5	Слово 6, всегда устанавливается на 0
Код команды	Адрес + 6	Слово 7, всегда E501 (шестнадцатеричное)
Размер блока данных параметра, байты (должен включать в себя 4 байта для слов спецификаторов параметров) *	Адрес + 7	Слово 8 (размер зависит от значения в слове 12)
Тип памяти данных параметров (для слова 11)	Адрес + 8	Слово 9, выбирается пользователем (см. таблицу кодов типов памяти)
Смещение данных параметров (для слова 11)	Адрес + 9	Слово 10, выбирается пользователем
Слова спецификатора параметров	Номер начального параметра (0 – 255)	Адрес хуz (адрес задается в словах 9 и 10)
	Число параметров, подлежащих загрузке	Адрес хуz + 1
Данные параметров	Данные 1-го параметра	Адрес хуz + 2/3
	Данные 2-го параметра	Адрес хуz + 4/5
	...	Адрес хуz + ...
	Данные 16-го параметра (4 байта)	Адрес хуz + 32/33

\* Размер блока данных параметров состоит из 4 байт для слов спецификаторов параметров плюс 4 байта для каждого параметра.

\*\* Использование этих слов зависит от значения в слове 12.

**Длина блока данных (слово 1).** Длина части заголовка блока данных в командном блоке. Для модуля DSM ее следует установить на 4. Заголовок блока данных хранится в словах 7 – 10 командного блока.

**Флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ) (слово 2).** Для DSM он всегда должен быть установлен на логический ноль.

**Тип памяти указателя слова статуса (слово 3).** Это слово определяет тип памяти, который будет использован для слова статуса. Каждый тип памяти имеет свой собственный специфический номер кода, показанный в таблице кодов типов памяти далее. Так, например, если есть желание использовать для слова статуса память %R, то следует ввести в это слово либо десятичный номер кода 8, либо шестнадцатеричный номер кода 08h.

Следует отметить, что если выбирается дискретный тип памяти (%I или %Q), то для слова памяти назначается 16 последовательных битов, начиная с адреса, заданного в слове смещения указателя слова статуса, приведенного ниже.

**Смещение указателя слова статуса (слово 4).** Это слово содержит смещение для выбора типа памяти. *Примечание: смещение указателя слова статуса является числом, базирующимся на нуле.* С практической позиции это означает, что требуется вычитать единицу из номера адреса, который требуется задать. Например, чтобы выбрать %R0001, следует ввести ноль ( $1 - 1 = 0$ ). Или, если требуется задать %R0100, то введите 99 ( $100 - 1 = 99$ ). Следует заметить, что тип памяти (в данном примере это %R) задан в предыдущем слове (см. пояснение выше "Тип памяти указателя слова статуса").

**Значение времени ожидания (слово 5).** Поскольку DSM всегда использует режим NOWAIT (НЕ ЖДАТЬ) (флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ) всегда установлен на ноль), то этот параметр (Значение времени ожидания) не работает в DSM. Установите его в ноль.

**Максимальная длительность обмена данными (слово 6).** Поскольку DSM всегда использует режим NOWAIT (НЕ ЖДАТЬ) (флаг WAIT/NOWAIT (ЖДАТЬ / НЕ ЖДАТЬ) всегда установлен на ноль), то этот параметр (Максимальная длительность обмена данными) не работает в DSM. Установите его в ноль.

**Командный код (слово 7).** Для модуля DSM он всегда равен E501 (шестнадцатеричный). Чтобы ввести это число непосредственно в виде шестнадцатеричного значения, используйте пословную команду MOVE. Тогда, поскольку это значение равно 58 625 в десятичном виде, то целочисленная команда MOVE (ограниченная максимальным десятичным значением 32 767, т.к. бит 16 используется для знака) не имеет возможности запомнить его. Пословная команда MOVE может содержать десятичные числа до 65 535 (FFFF в шестнадцатеричном виде).

**Размер данных параметров (слово 8).** Определяет размер данных параметров в байтах. Это значение зависит от значения в слове 12, которое задает количество подлежащих загрузке параметров. Это значение может быть от 8 до 68. Оно складывается из 4 байтов (для первых двух слов в разделе данных параметров) и дополнительных 4 байтов для каждого загружаемого параметра. Например, если требуется загрузить 16 параметров (максимум для COMM REQ), то умножение 4 на 16 дает 64. Сложение 4 с 64 дает полный размер 68 байтов.

**Тип памяти данных параметров (слово 9).** Это слово определяет тип памяти, который будет использован для данных параметров. Каждый тип памяти имеет уникальный номер кода, показанный в таблице кодов типов памяти далее. Так, например, для задания памяти %R, следует ввести в это слово либо десятичный номер кода 8, либо шестнадцатеричный номер кода 08h.

Обратите внимание, что если выбран тип дискретной памяти (%I или %Q), то для каждого параметра потребуется группа из 32 последовательных битов, а также по группе из 16 последовательных битов для слов 11 и 12.

**Смещение начального указателя данных параметров (слово 10).** Это слово содержит смещение для типа памяти, выбранного в параметре типа памяти данных параметров. *Примечание: смещение указателя данных параметров является числом, базирующимся на нуле.* С практической позиции это означает, что требуется вычитать единицу из номера адреса, который требуется задать. Например, чтобы выбрать %R0001 в качестве начальной точки данных параметров, следует ввести нуль ( $1 - 1 = 0$ ). Или, чтобы выбрать %R0100, введите 99 ( $100 - 1 = 99$ ). Следует заметить, что тип памяти (в данном примере это %R) задан в предыдущем слове.

**Номер начального параметра (слово 11).** Задаёт номер первого загружаемого параметра. Действительными являются значения от 0 до 255. Однако этот номер должен учитывать значение в слове 12. Например, если слово 12 задаёт 10 параметров, подлежащих загрузке, то номер начального параметра должен быть меньше 247; в противном случае номер последнего параметра, подлежащего загрузке, выйдет из разрешенного диапазона (будет больше 255).

**Количество параметров, подлежащих отправке, (слово 12).** Задаёт, сколько параметров будет загружено. Действительными являются значения от 1 до 16. Значение в этом слове воздействует на значение в слове 8.

**Данные параметров (слова 13 – 44).** Размер этой области данных параметров зависит от значения в слове 12 (количество параметров, подлежащих отправке). Для каждого параметра требуется два слова (4 байта). Поскольку действительными числами для параметров, представляющих собой двойные целые, являются числа от 1 до 16, то область данных параметров может быть от 2 до 32 слов.

**Коды типов памяти COMM REQ.** Коды, указанные в приведенной ниже таблице, используются в слове 3 (тип памяти указателя слова статуса) и в слове 9 (тип памяти данных параметров).

**Таблица В-8. Коды типов памяти для команды COMM REQ "Загрузка параметров"**

Коды типов памяти для команды COMM REQ "Загрузка параметров"			
Краткое обозначение типа памяти	Тип памяти	Номер кода, подлежащий вводу	
		Десятичный	Шестнадцатеричный
%I	Таблица дискретных входов	70	46h
%Q	Таблица дискретных выходов	72	48h
%R	Память регистров	8	08h
%AI	Таблица аналоговых входов	10	0Ah
%AQ	Таблица аналоговых выходов	12	0Ch

## Пример команды COMM REQ "Загрузка параметров" модуля DSM

Этот пример используется как основа для следующего раздела "Раздел 5: пример COMM REQ Ladder Logic". В данном примере используются следующие характеристики:

- Модуль DSM смонтирован в крейте 0, слот 7 ПЛК.
- Начальный адрес командного блока %R0196.
- Слово статуса располагается в %R0195.
- Подлежат отправке 16 параметров.
- FT-выход (неисправность) команды COMM REQ вызывает срабатывание "катушки с запоминанием".
- Параметр 1 модуля DSM в данном примере приложения рассматривается как критический. Последние два этапа "Примера COMM REQ Ladder Logic" (см. раздел 5) проверяют, что параметр получил правильное значение при помощи команды COMM REQ.
- Данные, находящиеся в 32 словах памяти ПЛК (16 двойных слов) (с %R0208 до %R0239), копируются в 16 двойных слов регистров параметров (с P001 до P016) в памяти параметров модуля DSM314. Эта передача данных иллюстрируется рисунком далее.

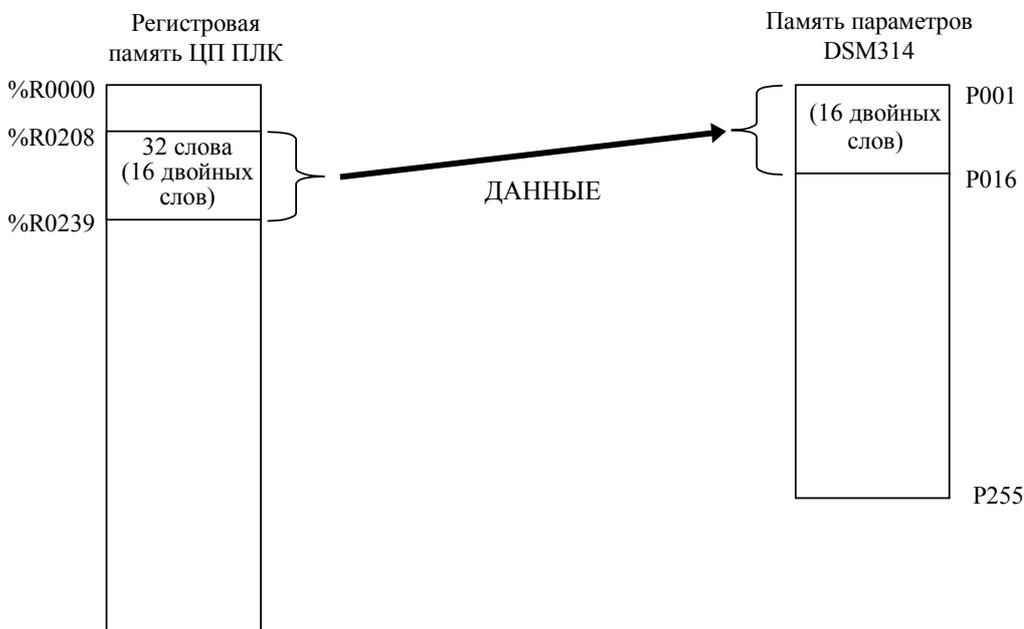
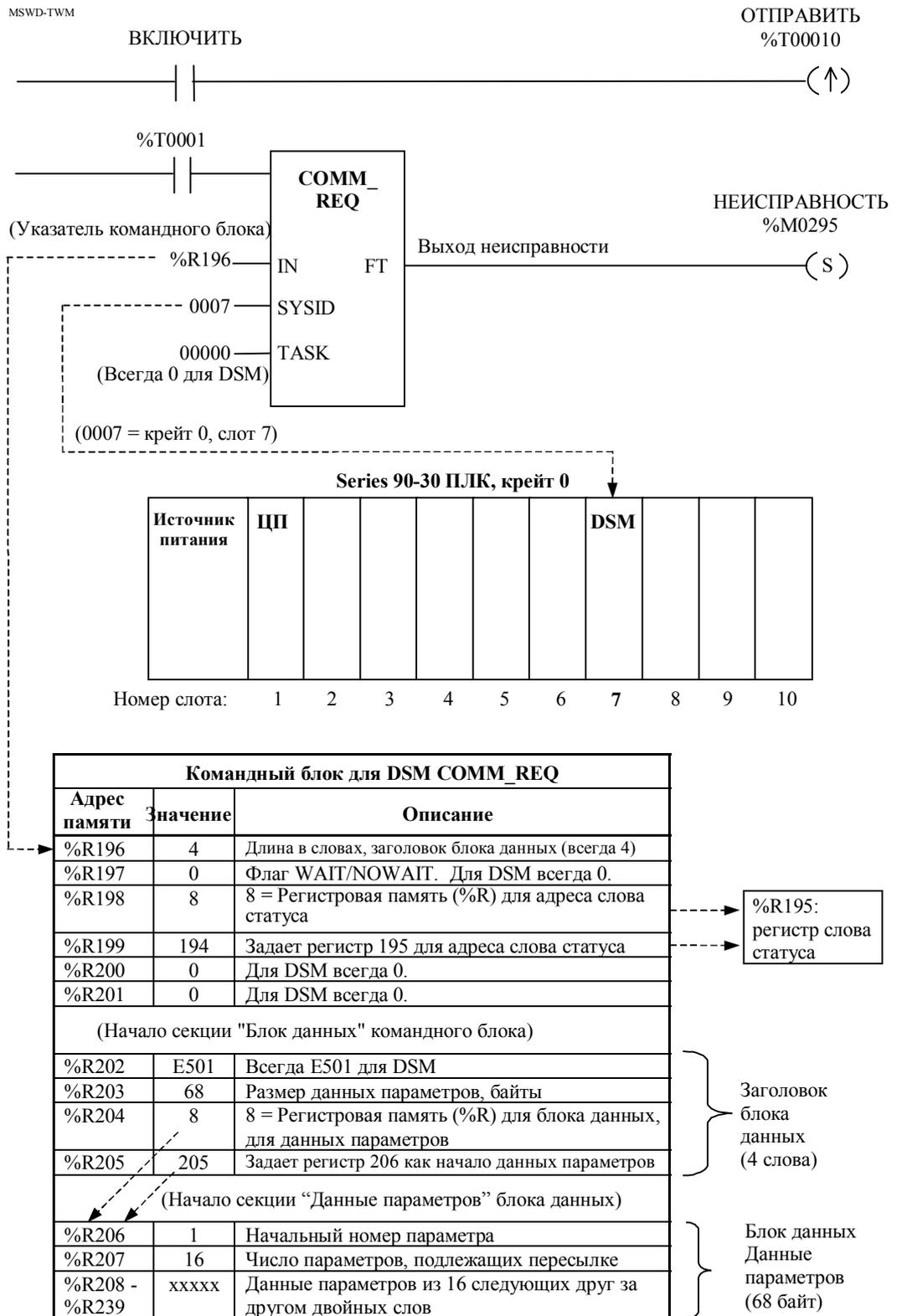


Рис. В-6. Передача данных в примере команды COMM REQ "Загрузка параметров"



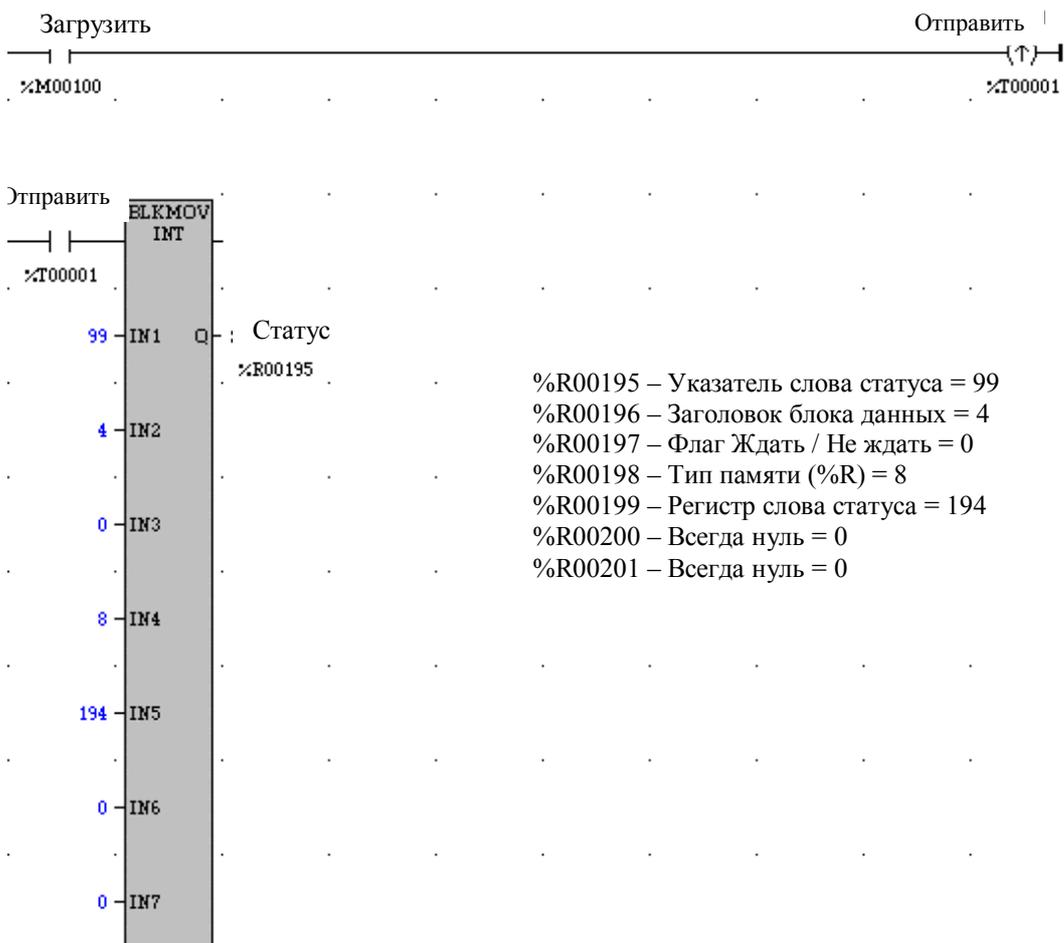
**Рис. В-7. Краткое описание примера команды COMM REQ "Загрузка параметров"**

## Раздел 5: Пример команды COMM REQ Ladder Logic

Следующий далее пример лестничной логики базируется на примере команды COMM REQ "Загрузка параметров" в предыдущем разделе. Листинг командного блока дан в таблице на предыдущей странице.

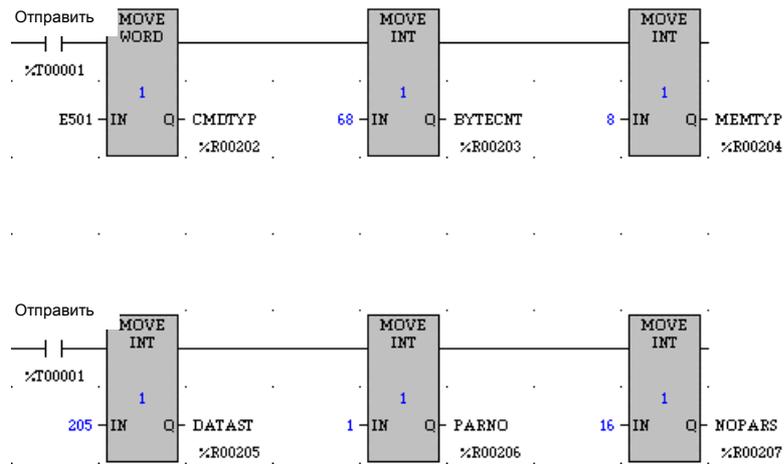
### Установка значений командного блока COMM REQ

На приведенных далее двух этапах происходит загрузка соответствующих значений в первые семь слов командного блока COMM REQ.



На следующих двух этапах происходит загрузка остающихся данных командного блока. Эти данные перечислены ниже:

- %R00202 – Команда. Для модуля DSM это всегда = E501 (шестнадцатеричное)
- %R00203 – Размер данных параметров, байт = 68
- %R00204 – Код типа памяти для памяти %R = 8
- %R00205 – Начальный регистр для данных параметров (смещение на 1) = 205
- %R00206 – Номер начального параметра = 1
- %R00207 – Число параметров, подлежащих отправке = 16
- %R00208 - %R00239 – Данные параметров, подлежащие отправке



**Логика для данных параметров (не показана)**

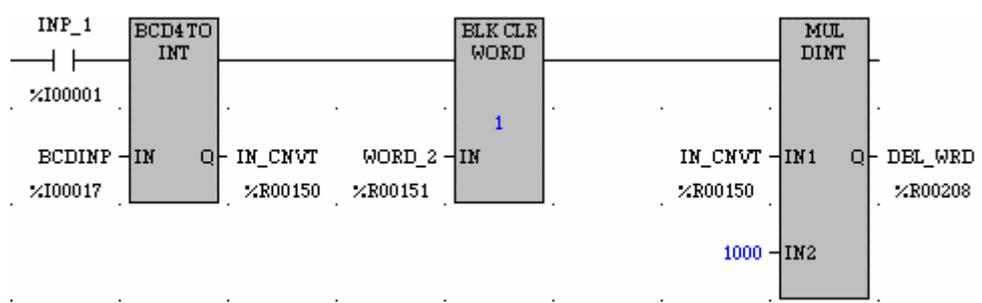
Для загрузки данных в регистры %R00208 - %R00239 так, чтобы они могли быть отправлены в параметры DSM314, требуется дополнительная логика. (Значение в двойном слове %R00208/%R00209 передается в параметр 1, значение в %R00210/%R00211 – в параметр 2 и т.д. до конца, где значение в %R00238/%R00239 направляется в параметр 16). Метод, подлежащий использованию для загрузки данных в эти регистры, зависит от применения. Если эти значения данных не будут меняться, то константы могут быть переданы в регистры с использованием команд Block Move и / или Move. Если значения могут изменяться, то их следует передавать в регистры с интерфейсного устройства оператора.

**Обработка значений параметров в виде двойных целых и масштабирование входных значений**

Данные в регистрах одинарной точности (16 бит) подлежат преобразованию в двойные целые числа (32 бита), т.к. параметры модуля DSM имеют размер двойных целых. Обычным путем сделать это является использование команды Double Integer Multiply (Умножение двойных целых) (MUL DINT), чтобы переместить входные данные в те регистры ПЛК, содержание которых будет направлено в модуль DSM. Имеются два потенциальных достоинства такого подхода:

- Это простой путь преобразовать регистры одинарной точности в форму двойных целых чисел.
- Он позволяет легко масштабировать входные значения, если это требуется. Под термином "масштабирование" здесь понимается умножение и / или деление значения, чтобы получить новое, пропорциональное исходному. Например, умножение входного значения на два, а затем деление его на 3 дает выходное значение, которое составляет всегда 2/3 входной величины. Масштабирование часто требуется в сервосистемах, чтобы подстроить фактическое расстояние перемещения к заданному расстоянию. Эта операция реализует функцию "электронной коробки передач". Она может быть использована для выставления значения передаточного числа, шага шариковой винтовой пары, разрешения датчика положения и пользовательских настроек входного значения.

В примере ниже целое число от входного устройства оператора (4-знаковый BCD (двоично-кодированный десятичный) координатный манипулятор) умножается на 1000, а затем помещается в целочисленное слово двойной точности %R00208/%R00209 (для параметра 1).



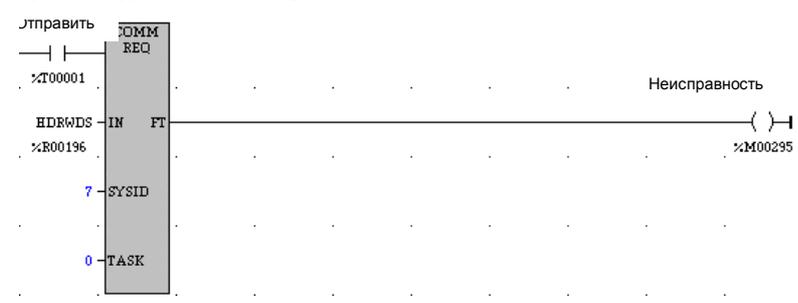
В примере выше, когда переключатель %I00001 замкнут, двоично-кодированное десятичное (BCD) значение в BCDINP (%I00017-%I00032) из входного BCD-устройства оператора преобразуется в целое значение (командой BDC4 TO INT); это целое значение помещается в регистр %R00150. Затем %R00151 сбрасывается на нуль командой BLK CLR. Обратите внимание, что на выходе команды BCD4 TO INT в %R00150 находится число одинарной точности. Однако, когда %R00150 используется как вход (IN1) для команды умножения двойной точности (MUL DINT), то ЦП автоматически объединяет его со следующим адресом %R (%R00151), чтобы сформировать значение двойной точности. В этом слове двойной точности слово %R00150 становится младшим, а слово %R00151 – старшим. Команда MUL DINT умножает значение в %R00150/%R00151 на 1000 и помещает результат в двойное слово %R00208/%R00209 (DBL\_WRD).

При использовании этого метода %R00151 называется "неявным адресом", поскольку он не появляется на экране. Примите во внимание, что нельзя использовать %R00151 для каких-либо иных целей (в нем должно быть значение нуля); в противном случае значение, помещенное в %R00150 из BDC4 командой INT будет изменено. То же самое применимо для двойного слова %R00208/R00209. Здесь использование %R00209 следует из того факта, что %R00208 отображается как выход команды (Умножение двойных целых) (MUL DINT). Поэтому %R00209 должен быть зарезервирован только для такого использования.

На этом этапе команда MUL DINT выполняет две функции: (1) преобразует значение в %R00150 из формы одинарной точности в форму двойной точности, и (2) масштабирует значение в %R00150/%R00151, умножая его на 1000. Если масштабирование не требуется, то в IN2 команды MUL DINT используется 1 вместо 1000; это обеспечит преобразование в форму двойной точности без изменения (масштабирования) значения.

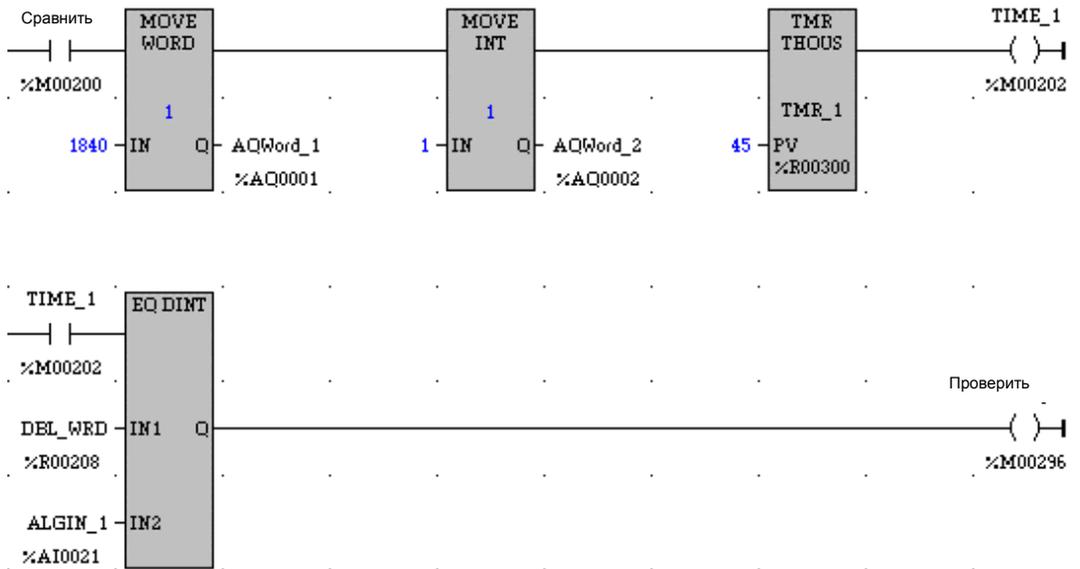
**Команда запроса на обмен данными**

На следующем рисунке показана команда запроса на обмен данными (COMM REQ). Вход IN содержит адрес первого слова командного блока. Вход SYSID содержит номер крейта и слота (крейт 00, слот 07) модуля DSM314, на который направлена команда COMM REQ. Вход TASK для модуля DSM314 всегда равен нулю. FT-выход подсоединен к исполнительному устройству (%M00295), на которое подается питание при обнаружении неисправности.



## Проверка передачи данных в параметр 1

В данном примере значение параметра 1 модуля DSM является критическим, поскольку этот параметр определяет расстояние перемещения, неправильная установка которого может привести к повреждению станка. Поэтому логика на следующих двух этапах проверяет правильность значения, полученного параметром 1. Если значение неправильное, то контакты (здесь не показано) выходной катушки "ПРОВЕРКА" на втором этапе отключают модуль DSM от выдачи команд движения.



**Первый этап.** Команда MOVE WORD перемещает шестнадцатеричное число 1840 в %AQ0001, первое слово команды немедленного действия. Значение младшего байта (40) этого числа определяет команду немедленного действия *Select Return Data (Выбрать возвращаемые данные)*. Значение старшего байта (18) определяет выбор режима для *Parameter Data (Данные параметров)*.

Команда MOVE INT перемещает десятичное значение 1, указывающее на параметр 1, в %AQ0002. Это означает, что значение, находящееся в параметре 1 модуля DSM, должно быть записано в двойное слово *User Selected Data (Данные, выбранные пользователем)* для оси 1, т.е. в данном примере в %AI00021/AI00022.

**Примечание:** Текущие адреса %AI, используемые для любого модуля DSM, задаются при конфигурации модуля.

Таймерная команда TMR THOUS (тысячные) создает временную задержку длительностью 45 миллисекунд после отправки команды немедленного действия *Select Return Data (Выбрать возвращаемые данные)*. Это требуется потому, что *User Selected Data (Данные, выбранные пользователем)* отсутствуют в схеме до тех пор, пока не пройдет 3 цикла или 20 миллисекунд (в зависимости от того, что больше) после отправки команды немедленного действия *Select Return Data (Выбрать возвращаемые данные)*. Поскольку длительность цикла в данном примере составляет 14 миллисекунд, то задержка 45 мс обеспечивает наличие данных параметра 1 в двойном слове *User Selected Data (Данные, выбранные пользователем)* до того, как начнется выполнение команды Equal (Равно) на следующем этапе. Обратите внимание, что контакт %M00200 должен оставаться в состоянии ON (ВКЛ.) достаточно долго, чтобы таймер TMR мог сработать и включить второй этап.

---

**Второй этап.** После окончания задержки 45 миллисекунд на предыдущем этапе срабатывает контакт %M00202 и включает данный этап. На этом этапе команда двойной точности EQUAL (РАВНО) сравнивает значение в %R00208/R00209 (источник значения, отправленного командой COMM REQ в параметр 1 модуля DSM) со значением, возвращенным от параметра 1 в %AI00021/AI00022. Если эти значения равны, то сработает исполнительное устройство "Проверка".



С модулем DSM314 работают четыре модели цифровых последовательных датчиков положения GE Fanuc серии  $\alpha$  и  $\beta$ .

**Таблица С-1. Разрешения цифровых последовательных датчиков положения**

8K	(8,192 импульсов / оборот)	- В новых двигателях отсутствует
32K	(32,768 импульсов / оборот)	- Стандарт для двигателей $\beta$ Series
64K	(65,536 импульсов / оборот)	- Стандарт для двигателей $\alpha$ Series
1000K	(1,048,576 импульсов / оборот)	- По заказу для двигателей $\alpha$ Series

**Примечание:** Последовательные датчики положения более ранних серий "А" или "С" на млн. импульсов не работают с модулем DSM314. Если подключить такой датчик положения, то будет выдано сообщение об ошибке.

Для целей управления положением модуль DSM314 рассматривает все датчики положения, по умолчанию, как выдающие 8192 импульса на оборот. Датчики положения с дополнительным разрешением 32K, 64K и 1000K используются в цифровом сервоконтроллере скорости для обеспечения плавной работы при малых скоростях. Чтобы использовать увеличенное разрешение обратной связи по положению, необходимо сделать установки в разделе "Параметры настройки" главы "Конфигурация".

## Режимы цифрового последовательного датчика положения

Цифровые последовательные датчики положения GE Fanuc могут работать либо в **инкрементном**, либо в **абсолютном** режиме. Режим конфигурируется выбором **Feedback Mode (Режим обратной связи)** в программе конфигурации. Правильная работа в **абсолютном** режиме требует наличия внешнего комплекта батарей, который должен быть подсоединен к сервоусилителю. Порядок выбора и установки комплекта батарей см. в соответствующем руководстве по усилителю.

## Информация по режиму инкрементного датчика положения

Цифровой последовательный датчик положения может быть использован как инкрементный датчик положения, выдающий 8192 импульса на оборот вала, без запоминания числа оборотов во время цикла выключения - подачи питания. Эквивалент маркерного импульса выдается один раз за оборот вала двигателя. Все **режимы позиционирования** (Home Switch, Move+, Move-) и команды %AQ Set Position (Установить позицию) используют в качестве опорной позицию оси и устанавливают бит %I Position Valid (Позиция достигнута) после успешного завершения. Здесь действуют сконфигурированные **Верхняя** и **Нижняя границы позиции**; слово статуса %AI Actual Position (Текущая позиция), сообщаемое модулем DSM314, будет переходить

от верхнего к нижнему или от нижнего к верхнему значению импульсов. Этот режим прекрасно подходит для применений с непрерывным инкрементным движением в одном направлении. Пункты конфигурации **Home Offset (Начальное смещение)** и **Home Position (Начальная позиция)** позволяют просто осуществлять установку в требуемое положение.

## Информация по режиму абсолютного датчика положения

Цифровой последовательный датчик положения GE Fanuc может быть использован как датчик абсолютного типа при использовании комплекта батарей, чтобы сохранить информацию о позиции сервопривода при выключении питания. Должен быть выполнен цикл начального позиционирования или команда **%AQ Set Position (Установить позицию)** в начале работы или при каждом отключении батареи и при обесточенном состоянии сервоусилителя. Для правильной работы комплекта батарей **Feedback Mode (Режим обратной связи)** должен быть установлен в положение АБСОЛЮТНЫЙ.

### Абсолютный датчик положения – первое использование или использование после отключения питания от батареи

Абсолютный датчик положения в течение некоторого времени при первом использовании или после восстановления питания датчика от батареи выдает инкрементные данные. Инкрементные данные сбрасываются, когда датчик положения при вращении вала двигателя проходит через опорную точку (аналогично маркерному сигналу) во время одного оборота вала двигателя. После нового подключения батареи абсолютного режима к усилителю необходимо, чтобы цифровой абсолютный последовательный датчик положения повернулся на один полный оборот. Датчик положения определит начало отсчета в течение одного оборота и передаст информацию о состоянии, привязанном к опорной точке, в модуль DSM314.

### Режим абсолютного датчика – инициализация позиции

Когда система первый раз включается в режиме абсолютного датчика положения, должно быть задано смещение позиции для датчика положения. Это достигается использованием цикла **%Q Find Home (Начальное позиционирование)** или команды **%AQ Set Position (Установить позицию)**.

### Цикл начального позиционирования – режим абсолютного датчика положения

**Find Home Mode (Режим начального позиционирования)** может быть сконфигурирован для операции Move (Движение) (+), Move (Движение) (-) или Home Switch (Датчик начальной позиции). См. в главе 4 дополнительную информацию по операции начальной позиции. Пункты конфигурации **Home Offset (Смещение начальной позиции)** и **Home Position (Начальная позиция)** выполняются так же, как в режиме инкрементного датчика положения. При завершении цикла начальной позиции слово статуса **Actual Position (Текущая позиция) %AI** устанавливается на сконфигурированное значение **Home Position (Начальная позиция)**. Модуль DSM314 рассчитывает внутри абсолютное смещение обратной связи датчика положения, требуемое для выхода на сконфигурированное значение **Home Position (Начальная позиция)** после завершения цикла начальной позиции. Это абсолютное смещение обратной связи сразу же сохраняется в энергонезависимой (емкостное резервирование) памяти модуля DSM314.

После того как абсолютная позиция задана при успешном завершении цикла начального позиционирования, модуль DSM314 автоматически инициализирует слово статуса %AI *Actual Position (Текущая позиция)* при включении питания и устанавливает бит %I *Position Valid (Позиция достигнута)*.

**Примечание:** Если бит %I *Position Valid (Позиция достигнута)* установлен до инициации цикла начальной позиции, то этот цикл очищает бит *Position Valid (Позиция достигнута)*, а затем устанавливает его снова при завершении цикла. Если цикл начальной позиции прерывается битовой командой %Q *Abort All Moves (Прекратить все движения)*, то бит *Position Valid (Позиция достигнута)* остается неустановленным. Однако выключение – включение питания восстанавливает правильную текущую позицию, и происходит автоматическая установка бита *Position Valid (Позиция достигнута)*.

## Команда установки позиции – режим абсолютного датчика положения

Команда %AQ *Set Position (Установить позицию)* работает так же, как в режиме инкрементного датчика положения. При завершении операции *Set Position (Установить позицию)* *Actual Position (Текущая позиция)* устанавливается на значение, заданное в *Set Position*. Модуль DSM314 рассчитывает внутри абсолютное смещение обратной связи датчика положения, требуемое для выхода на значение, заданное в *Set Position (Установить позицию)*. Это абсолютное смещение обратной связи сразу же сохраняется в энергонезависимой (емкостное резервирование) памяти модуля DSM314.

Если команда AQ *Set Position (Установить позицию)* получена до того, как датчик положения был привязан к опорной точке, то выдается код ошибки 53 (шестнадцатеричный) "Попытка инициализировать позицию до того, как цифровой датчик положения прошел опорную точку". Этот код ошибки выдается только в том случае, если режим обратной связи установлен как **Абсолютный**. Последовательные датчики положения, сконфигурированные в режиме **Инкрементный**, не имеют этого ограничения.

Если абсолютная позиция установлена командой *Set Position (Установить позицию)*, то модуль DSM314 автоматически инициализирует *Actual Position (Текущая позиция)* после подачи питания и устанавливает бит %I *Position Valid (Позиция достигнута)*.

## Режим абсолютного датчика положения – включение питания модуля DSM314

Батарея, устанавливаемая в сервосистему, сохраняет питание для счетных схем датчика положения. После того как датчик положения был привязан к опорной точке при первом включении, он автоматически поддерживает информацию о текущей позиции, в т.ч. и при движении оси, когда питание системы выключено. Датчик положения отслеживает состояние батареи и сообщает модулю DSM314 об отсутствии или пониженном напряжении батареи.

Модуль DSM314 выполняет диагностику питания и, если он сконфигурирован в режиме абсолютного датчика положения, запрашивает состояние цифрового последовательного датчика положения, привязанное к опорной точке. Правильное привязанное к опорной точке состояние, полученное от датчика положения, вызывает считывание модулем DSM314 абсолютной позиции датчика. Модуль DSM314 выдает состояние %AI *Actual Position (Текущая позиция)* как сумму позиции датчика положения и абсолютного

смещения обратной связи, установленного при исходном цикле начального позиционирования или командой *%AQ Set Position (Установить позицию)*.

## Инкрементный импульсный датчик положения

Инкрементный импульсный датчик положения обеспечивает три выходных сигнала для модуля DSM314: канал А, канал В и маркер. Канал А и канал В выдают импульсы при вращении датчика положения, что позволяет модулю DSM314 считать число этих импульсов и определять последнее изменение позиции датчика положения и направление вращения.

Инкрементные импульсные датчики положения являются инкрементными устройствами обратной связи; они не обеспечивают непрерывную индикацию абсолютного угла вала при его вращении. Поэтому слово статуса *%AI Actual Position (Текущая позиция)* модуля DSM314 должно быть инициализировано при известном физическом положении до того, как будет разрешено управление позицией. Это выставление позиции может быть выполнено при помощи команды немедленного действия *%AQ Set Position (Установить позицию)* или цикла *%Q Find Home (Начальное позиционирование)*. Цикл начальной позиции использует маркерный канал датчика положения, который выдает один импульс на оборот при известном угле датчика положения. Успешное выполнение цикла *%Q Find Home (Начальное позиционирование)* или команды *%AQ Set Position (Установить позицию)* вызывает установку модулем DSM314 бита оси *%I Position Valid (Позиция достигнута)*. Бит *Position Valid (Позиция достигнута)* должен быть установлен до того, как программы движения получат разрешение на выполнение. Бит *Position Valid (Позиция достигнута)* сбрасывается только при возникновении квадратурной ошибки датчика положения (каналы А и В переключаются одновременно) или при одновременном включении битов *Find Home (Начальное позиционирование)* и *Abort (Прекратить) %Q*.

**Примечание:** В цифровом режиме для ведущей оси в режиме слежения поддерживаются только инкрементные импульсные датчики положения.

Настоящее приложение описывает процедуру запуска и настройки цифровой или аналоговой сервосистемы GE Fanuc. Для цифровых сервосистем в модуле DSM314 имеются два контура управления, которые требуют настройки: контур скорости и контур позиционирования. Следует всегда начинать с конфигурации модуля; затем выполняется установка контура скорости, а в завершение – контура позиционирования. Для аналоговых сервосистем имеется последовательность действий запуска, которая должна быть выполнена.

## Информация о запуске и настройке цифровых сервосистем

Здесь рассматриваются три основных раздела:

- Проверка правильности работы датчика начальной позиции, входов переключения и направления вращения двигателя
- Настройка контура скорости
- Настройка контура позиционирования

## Проверка правильности работы датчика начальной позиции, входов перебега и направления вращения двигателя

1. Подключите двигатель, усилитель и модуль DSM314, как описано в главе 2.
2. Если используются концевые выключатели выхода за границу (в конфигурации задано **Overtravel Limit Switch (Выключатель выхода за границу)** = Enabled (Включено)), то подключите их к соответствующим клеммам клеммной колодки 24В (см. гл. 3). Входы выключателя выхода за границу работают в безопасном режиме, т.е. используются нормально замкнутые или типа PNP устройства переключения. Через вход должен протекать ток, обеспечивающий логическую единицу на входе, если ось НЕ находится в позиции выхода за границу; в противном случае возвращается код ошибки (ошибка A9). Чтобы этого не было, **Overtravel Limit Switch (Выключатель выхода за границу)** должен быть установлен на Disabled (Отключено) при конфигурации.
3. Если используется переключатель начальной позиции (в конфигурации задано Home Mode (Режим начальной позиции) = Home Switch (Переключатель начальной позиции)), то подключите его к соответствующим клеммам клеммной колодки 24В (см. гл. 3). Этот переключатель начальной позиции должен быть подключен и запитан так, чтобы он был в состоянии ALWAYS ON (замкнут), когда ось находится

с отрицательной стороны начальной позиции, и в состоянии ALWAYS OFF (разомкнут), когда ось находится с положительной стороны начальной позиции. Обычно переключатель начальной позиции на конце участка перемещения оси или вблизи него. Необходимо проверить работу переключателя начальной позиции перед тем, как запустить цикл начальной позиции. Может оказаться необходимым изменить направление вращения двигателя (Motor1 или Motor2 Dir = POS/NEG) в конфигурации модуля.

4. Используйте программное обеспечение конфигурации для задания требуемых пользовательских коэффициентов масштаба и других конфигурируемых параметров. ДОЛЖНЫ быть изменены следующие позиции по сравнению с установками конфигурации по умолчанию:

<u>Позиция конфигурации</u>	<u>Установка</u>
Axis 1 Mode (Режим оси 1):	Цифровой Сервопривод
Motor Type (Тип двигателя):	Выберите в таблице в главе 4
Position Loop Time Constant (Постоянная времени позиционирования):	60 мс
Velocity Loop Gain (Управляющий коэффициент скорости):	(Момент инерции нагрузки / момент инерции двигателя) * 16
User Units (Пользовательские единицы): Counts (Счетчик) (только стандартный режим)	См. главу 3
Position Error Limit (Предел ошибки позиционирования):	30000 x Пользовательские единицы / Импульсы

**Задайте параметры конфигурации в порядке, указанном выше.**

5. Сохраните конфигурацию на ПЛК.
6. Сбросьте программу от ПЛК, выключите все %Q биты модуля DSM314 и установите ПЛК в режим RUN (ВЫПОЛНИТЬ). Просмотрите STL-биты %I для Home Switch (датчик начальной позиции), (+) Overtravel (Перебер), (-) Overtravel и убедитесь, что каждый бит соответствует правильному переключателю (см. в главе 5 определения битов %I).
7. Установите бит %Q *Enable Drive (Включить привод)* и убедитесь, что сервоусилитель включен. Если на серводвигателе используется тормоз, то он должен быть в это время отпущен.
8. Выдайте код команды %AQ *Force Digital Servo Velocity (Включить скорость цифрового сервопривода)* 100 (об./мин). Убедитесь, что двигатель вращается в требуемом ПОЛОЖИТЕЛЬНОМ направлении и что *Actual Velocity (Текущая скорость)*, приведенная в таблице %AI, ПОЛОЖИТЕЛЬНАЯ. Если двигатель вращается в неправильном направлении, то используйте параметр **Axis Direction (Направление оси)** в программе конфигурации, чтобы поменять положительное и отрицательное направления оси.
9. Удалите команду *Force Digital Servo Velocity (Включить скорость цифрового сервопривода)* из таблицы %AQ. Используйте малые значения **Jog Velocity (Скорость толчка)** и **Jog Acceleration (Ускорение толчка)** в конфигурации; значения могут быть увеличены позднее. Установите %Q бит *Jog Plus (Толчок плюс)*. Убедитесь, что сервопривод двигается в правильном направлении и что значение *Actual Velocity (Текущая скорость)*, передаваемое модулем DSM314 в %AI

таблицу, соответствует сконфигурированной *Jog Velocity* (*Скорости передвижения*). Если программы движения будут использовать ускорение, превышающее данное *Jog Acceleration* (*Ускорение передвижения*), то может оказаться необходимым увеличить *Jog Acceleration* так, чтобы команды *Abort All Moves* (*Прекратить все движения*) и Normal Stop (Нормальный останов) могли действовать, как положено.

10. Используйте малые значения для *Find Home Velocity* (*Скорость поиска начальной позиции*) и *Final Home Velocity* (*Скорость подхода к начальной позиции*) в конфигурации модуля; значения могут быть увеличены позднее. Проверьте правильность работы цикла *Find Home* (*Начальное позиционирование*) включением на короткое время бита %Q *Find Home* (бит %Q *Drive Enabled* (*Привод включен*)) также должен быть установлен). Ось должна двигаться в сторону переключателя начальной позиции при сконфигурированном значении *Find Home Velocity* (*Скорость поиска начальной позиции*), затем подходить к опорной точке датчика положения при сконфигурированном значении *Final Home Velocity* (*Скорость подхода к начальной позиции*). Если требуется, подстройте сконфигурированные скорости и положение переключателя начальной позиции, чтобы обеспечить согласованные действия. Конечный переключатель начальной позиции ДОЛЖЕН сработать, по крайней мере, за 10 миллисекунд до того, как произойдет выход в опорную точку датчика положения. Физическое положение *Home Position* (*Начальная позиция*) может быть подстроено изменением значения *Home Offset* (*Смещение начальной позиции*) в программе конфигурации.
11. Проследите работу сервопривода и используйте %Q-биты *Jog Plus* (*Толчок плюс*) и *Jog Minus* (*Толчок минус*) для перемещения серводвигателя в каждом направлении. Помещение правильного командного кода в таблице %AQ может временно изменить *Position Loop Time Constant* (Постоянная времени позиционирования). Для большинства систем эту *Постоянная времени позиционирования* можно уменьшать до тех пор, пока не будет замечена некоторая нестабильность сервопривода; после этого следует увеличить ее значение на, примерно, 50%. После задания правильной постоянной времени необходимо обновить конфигурацию DSM314, используя программное обеспечение конфигурации. Для оптимальной реакции сервопривода можно также задать ненулевое значение для *Velocity Feedforward* (*Коэффициент упреждения по скорости*) (обычно 90 – 100%). Информацию по установке *Velocity Loop Gain* (*Управляющий коэффициент скорости*) для цифрового сервопривода см. в разделе "Настройка цифрового сервопривода GE Fanuc".
12. Если режим слежения используется с инкрементным импульсным датчиком, то следует убедиться, что *Actual Position* (*Текущая позиция*) (вспомогательная ось Z) отображает позицию датчика положения. **Убедитесь, что при помощи программного обеспечения конфигурации запрограммировано требуемое для осей следящего устройства отношение "ведомая ось : ведущая ось", как отношение "A:B".**

## Рекомендации по поиску неисправностей при пуске цифровой сервосистемы

1. Модуль DSM314 требует наличия микропрограммы ПЛК версии 10.0 или более поздней и одного из следующих программных пакетов конфигурации / программирования:
  - Пакет SIMPLICITY Machine Edition Logic Developer PLC, версия 2.1 или более поздняя.
  - Пакет VersaPro, версия 1.1 или более поздняя.

2. Поддержка модулем DSM двигателей Beta M1 и Beta M0.5 требует микропрограммы DSM версии 3.0 или более поздней.
3. Конфигурацией модуля DSM314 для входов *Overtravel Limit Switch (Выключатель выхода за границу)* по умолчанию является ENABLED (ВКЛЮЧЕНО). Поэтому на входы перебега должно быть подано 24 В постоянного тока; в противном случае модуль DSM314 работать не будет. Если входы перебега не используются, то при конфигурации модуля DSM314 для входов *Overtravel Limit Switch (Выключатель выхода за границу)* должно быть установлено DISABLED (ОТКЛЮЧЕНО).

Если бит *%I Axis Enabled (Ось включена)* находится в состоянии OFF (ВЫКЛ.), то ось не будет реагировать на какой бы то ни было бит *%Q* или команды *%AQ*. Если серводвигатель не используется с сервоосью, то *Motor Type (Тип двигателя)* должен быть установлен на 0, или бит *Axis Enabled (Ось включена)* будет оставаться в состоянии OFF (ВЫКЛ.). Установка 0 в *Motor Type (Тип двигателя)* отключает работу сервоконтура оси и устанавливает *Axis Enabled (Ось включена)* в состояние ON (ВКЛ.), позволяя, таким образом, оси воспринимать команды, такие как *Load Parameter Immediate (Загрузить параметр немедленно)* и *Set Analog Output Mode (Установить режим аналогового выхода)*.

4. **Управляющий бит *%Q Enable Drive (Включить привод)* должен быть постоянно в состоянии ON (ВКЛ.); в противном случае может быть выполнено только движение толчка.** Если не возникают ошибки STOP (СТОП), то бит статуса *%I Drive Enabled (Привод включен)* будет зеркально отражать статус *%Q*-бита *Enable Drive (Включить привод)*. Ошибка STOP (СТОП) выключает бит *Drive Enabled (Привод включен)*, несмотря на то, что бит *Enable Drive (Включить привод)* остается в состоянии ON (ВКЛ.). Необходимо устранить состояние ошибки и установить управляющий бит *Clear Error (Сбросить ошибку)* *%Q* на один цикл ПЛК, чтобы снова включить привод.
5. Если *%I*-бит статуса *Module Error Present (Имеется ошибка модуля)* находится в состоянии ON (ВКЛ.), а *%I*-биты статуса *Axis Enabled (Ось включена)* и *Drive Enabled (Привод включен)* – в состоянии OFF (ВЫКЛ.), то это значит, что возникла ошибка STOP (СТОП) (светодиодный индикатор статуса быстро мигает). **В этом состоянии сервоось не будет реагировать на какие бы то ни было *%Q*-биты или команды *%AQ*, кроме *%Q*-бита *Clear Error (Сбросить ошибку)*.**
6. Управляющий *%Q*-бит *Clear Error (Сбросить ошибку)* является битом одноразового действия. При каждом возникновении ошибки этот бит должен быть установлен в состояние OFF (ВЫКЛ.), а затем в ON (ВКЛ.), по крайней мере, в течение одного цикла ПЛК, чтобы сбросить ошибку.
7. Индикатор CFG ОК должен быть в состоянии ON (ВКЛ.); в противном случае модуль DSM314 не будет реагировать на команды ПЛК. Если этот индикатор находится в состоянии OFF (ВЫКЛ.), то это значит, что от ПЛК не получена правильная конфигурация модуля DSM314 или что была обнаружена ошибка конфигурации. Проверьте слова кода ошибки *%AI* на ошибки Dxxx, которые приведены в разделе "Коды системных ошибок" в Приложении А. Также проверьте таблицы сбоя ПЛК на предмет ошибок конфигурации, сообщения о которых были выданы.

## Настройка цифрового сервопривода GE Fanuc

Приведенное далее описание дает введение в основные знания, требующиеся для настройки цифрового сервопривода GE Fanuc. Это введение показывает один метод для настройки сервопривода. Этот метод не может быть использован для всех применений; его подход следует изменять в зависимости от конкретного применения. Чтобы отобразить и измерить форму требуемого сигнала, аналоговые выходы модуля DSM314 должны быть подключены к осциллографу. Без осциллографа невозможно измерять сигналы и настраивать сервопривод, используя приведенный далее подход. Команда *Select Analog Output Mode (Выбрать режим аналогового выхода)* %AQ (47h) используется для того, чтобы выбрать данные, которые направляются на аналоговые выходы в ходе настройки сервопривода. Описание этой команды %AQ дано в главе 5.

### Требования к настройке

Данный модуль имеет три основных параметра, которые регулируются в ходе настройки. Этими параметрами являются *Position Loop Time Constant (Постоянная времени позиционирования)*, *Velocity Feed Forward Gain (Коэффициент упреждения по скорости)* и *Velocity Loop Gain (Управляющий коэффициент скорости)*. *Position Loop Integrator Time Constant (Постоянная времени интегратора контура позиционирования)* дает контуру позиционирования дополнительную степень свободы, но в типичных применениях она не требуется.

Подходом, используемым здесь для настройки контуров управления, является первоначальная настройка внутренних контуров управления. В настоящем примере внутренним контуром управления, который требует настройки, является контур скорости. Как видно на рисунке ниже, контур позиционирования является внешним контуром; он выдает команды скорости на контур скорости.

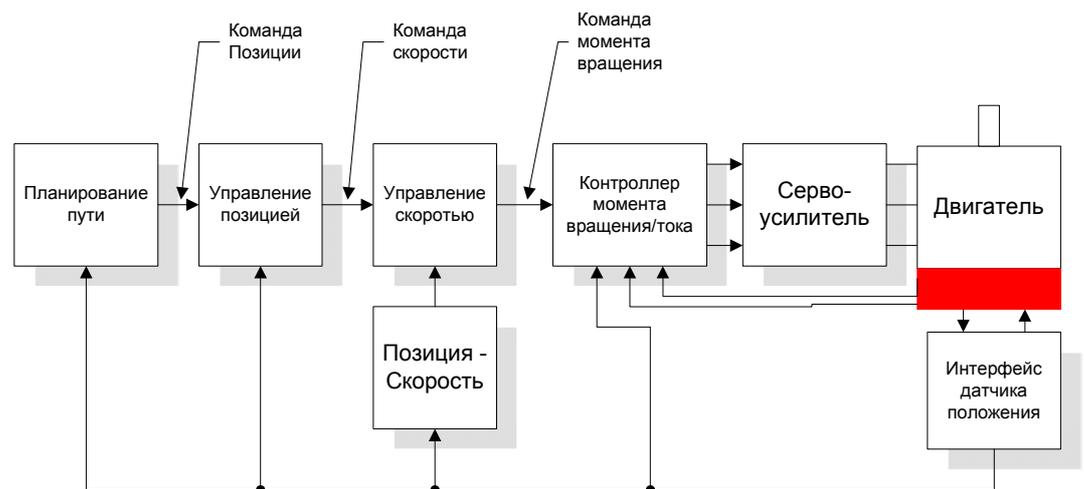


Рис. D-1. Блок-схема контуров управления

## Настройка ЦИФРОВОГО РЕЖИМА контура скорости

Правильным методом настройки контура скорости является разделение контуров скорости и позиционирования. Для такого разделения должен быть использован метод прямой отправки команд скорости без использования управления контуром позиционирования. Модуль DSM имеет несколько режимов, которые позволяют пользователю направлять команду скорости непосредственно на контур скорости. Ниже дано описание этих методов.

### Метод #1

Команда немедленного действия *%AQ Force Digital Servo Velocity (Включить скорость цифрового сервопривода)* (34h) выдает команду скорости непосредственно в контур скорости. Эта команда отличается от команды *Move at Velocity (Движение на скорости)*, которая для генерации команды использует контур позиционирования. Это представляется важным, поскольку на данном этапе процесса настройки контур позиционирования не должен взаимодействовать с контуром скорости. Команда *%AQ Force Digital Servo Velocity (Включить скорость цифрового сервопривода)* позволяет пользователю выдать ступенчатое изменение скорости. Задание ступенчатого изменения скорости используется для выработки реакции контура скорости на ступенчатое воздействие. Пользователь должен учесть, что при задании ступенчатого изменения скорости величина ускорения ограничена только полосой пропускания контура скорости. В некоторых применениях это может привести к повреждению управляемого устройства из-за большого ускорения.

### Метод #2

В некоторых применениях метод #1 приводит к сильному удару на управляемом устройстве. В этих случаях требуется другой метод генерации команды скорости. Этот метод требует, чтобы пользователь установил контур позиционирования в конфигурацию открытого контура. Это достигается установкой *Position Loop Time Constant (Постоянная времени позиционирования)* на нуль, а *Velocity Feedforward Gain (Коэффициент упреждения по скорости)* – на 100%. После этого можно использовать команду *Move at Velocity (Движение на скорости)* или программу движения, чтобы выдавать команды скорости на сервопривод.

Первым параметром, требующим настройки, является *Velocity Loop Gain (Управляющий коэффициент скорости)*. Этот параметр задает ширину полосы пропускания контура скорости. В качестве начальной точки следует использовать приведенную ниже формулу (см. также раздел о коэффициенте обратной связи по скорости):

### Уравнение 1

$$\text{Управляющий коэффициент скорости} = \frac{J_1}{J_m} \cdot 16$$

где:

$J_1$  = момент инерции нагрузки

$J_m$  = момент инерции двигателя

Во многих случаях не требуется изменять **Управляющий коэффициент скорости**, рассчитанный в уравнении 1. Однако при некоторых применениях (например, когда возникает резонанс на машине) может оказаться необходимым подстроить это значение. Для изменения **Управляющий коэффициент скорости** может быть использована описанная далее процедура.

1. Выберите метод, позволяющий ввести команду скорости в контур скорости. Методы #1 и #2 (см. выше) являются примерами методов, предназначенных для такой задачи.
2. Подсоедините осциллограф к аналоговым выходам обратной связи по скорости двигателя и команд момента вращения. Команды конфигурации аналоговых выходов см. в разделе 4.25 главы 5.
3. Установите коэффициент обратной связи по скорости на ноль. Этот подход является "осторожным". Если о применении известно, что оно не имеет резонансных частот от нуля до, примерно, 250 Гц, то можно начинать с более высокого значения, но не выше значения, рассчитанного в уравнении 1 в данном разделе.
4. Выдайте ступенчатое изменение скорости. На этом этапе ступенчатое изменение должно быть относительно небольшим по сравнению с полной скоростью машины. Хорошим началом могло бы быть 10 – 20% номинальной скорости машины.
5. Наблюдайте на осциллографе изменение скорости двигателя и команды момента вращения. Целью является получение реакции критического затухания в контуре скорости. Обратите особое внимание на любые колебания, которые могут происходить в сигнале обратной связи по скорости.
6. Увеличивайте понемногу **Управляющий коэффициент скорости** и повторяйте шаги 4 и 5 до тех пор, пока в сигнале обратной связи по скорости двигателя не появится некоторая неустойчивость. При достижении этого значения уменьшите **Управляющий коэффициент скорости**, по крайней мере, на 15%. Общим правилом является следующее: чем ниже значение **Управляющий коэффициент скорости** при удовлетворении требований к системе, тем надежнее работает управление. Следует внимательно наблюдать сигнал обратной связи по скорости. В некоторых применениях подъем **Управляющий коэффициент скорости** до достаточно большого значения, чтобы создать неустойчивость, может привести к повреждению машины. При опасениях установите **Управляющий коэффициент скорости** не более значения, рассчитанного в уравнении 1. Если в сигнале обратной связи по скорости двигателя начнутся колебания до достижения этой точки, то следует уменьшить величину **Управляющий коэффициент скорости** и перейти к шагу 7.
7. На этом этапе контур скорости можно считать настроенным. Однако должна быть проверена робастность контура. Для такого теста введите ступенчатое изменение скорости с величиной ступеньки 20, 40, 60, 80 и 100% номинальной скорости машины. Наблюдайте сигналы скорости двигателя и команды момента вращения на предмет неустойчивости. При проявлении неустойчивости или резонанса следует уменьшить **Управляющий коэффициент скорости** и повторить тест.

**Примечание:** Для цифровых сервоприводов команда Force Analog Output (Включить аналоговый выход) %AQ может обеспечить уставку момента вращения или заданную скорость двигателя. Скорость = 750 обор. в мин./Вольт и %TqCmd = (100/1.111111 В)\*X Вольт  
или Torque Cmd = 100% Уставка момента вращения = 1.111 Вольт,  
100%TqCmd = MaxCur усилителя. Например: Beta 0.5  
MaxCurAmp = 12 ампер => 1.111111 Вольт = 12 ампер.

## Пример проведения настройки ЦИФРОВОГО РЕЖИМА контура скорости

На рисунках ниже показан пример проведения настройки контура скорости.

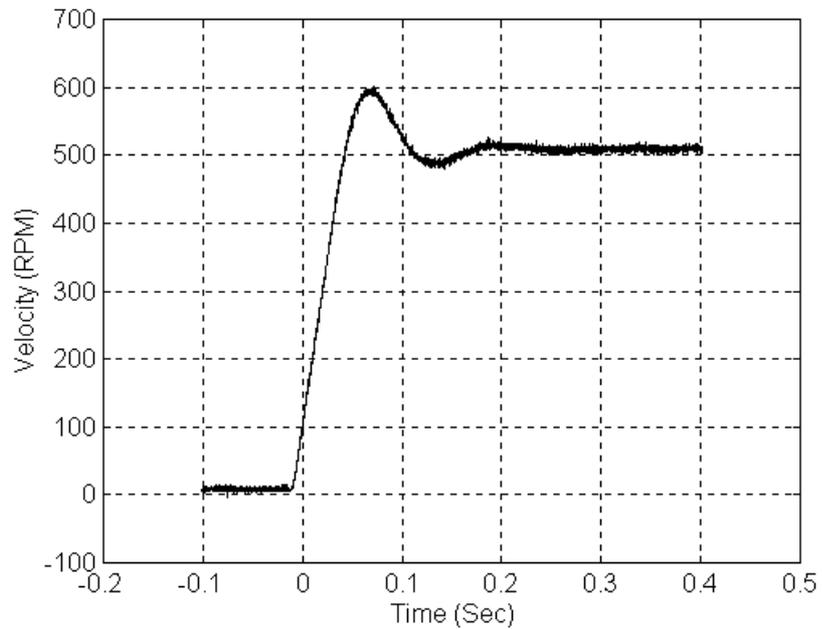


Рис. D-2. Реакция при ступенчатом изменении скорости, зависимость скорости от времени, VLGN (управляющий коэффициент скорости) = 0

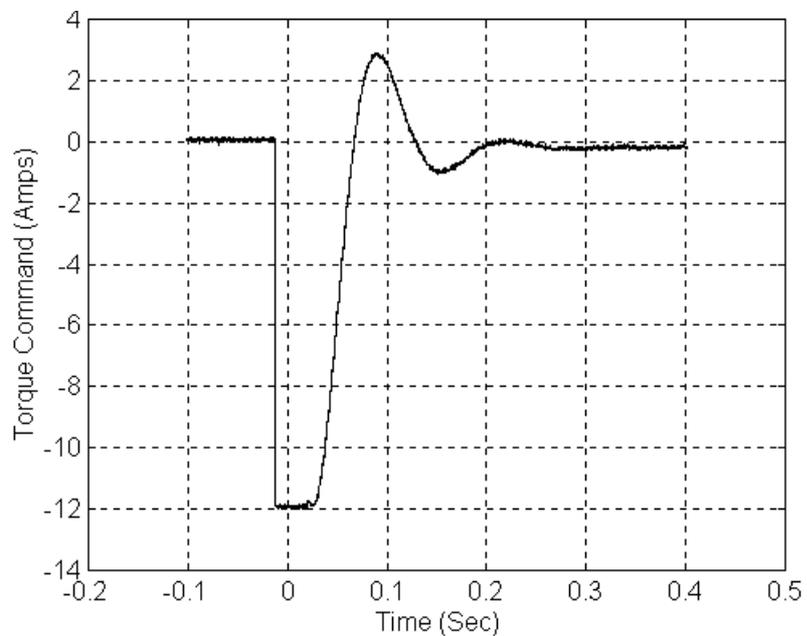


Рис. D-3. Реакция при ступенчатом изменении скорости, зависимость уставки момента вращения от времени, VLGN (управляющий коэффициент скорости) = 0

Обратите внимание, что на рисунках D-2 и D-3 система имеет недостаточное затухание. В этом случае ширина полосы пропускания контроллера меньше требуемой; коэффициент обратной связи по скорости должен быть увеличен.

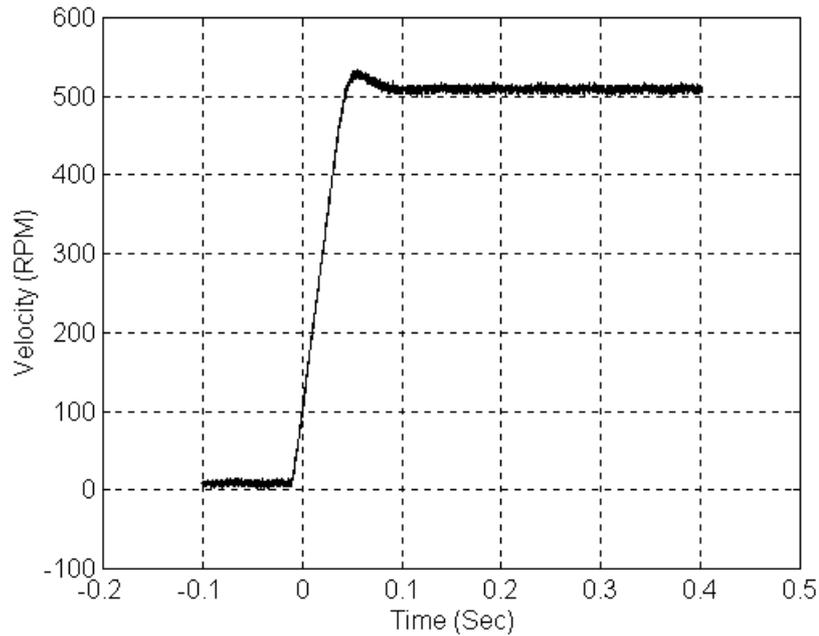


Рис. D-4. Реакция при ступенчатом изменении скорости, зависимость скорости от времени, VLGN (управляющий коэффициент скорости) = 24

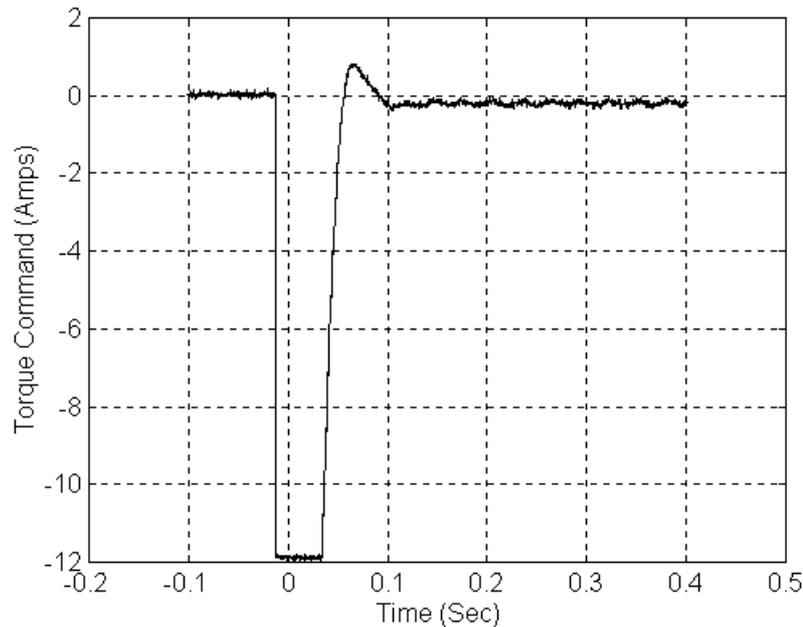


Рис. D-5. Реакция при ступенчатом изменении скорости, зависимость уставки момента вращения от времени, VLGN (управляющий коэффициент скорости) = 24

Обратите внимание, что на рисунках D-4 и D-5 система начинает выглядеть приемлемо. Единственной проблемой является перерегулирование по скорости.

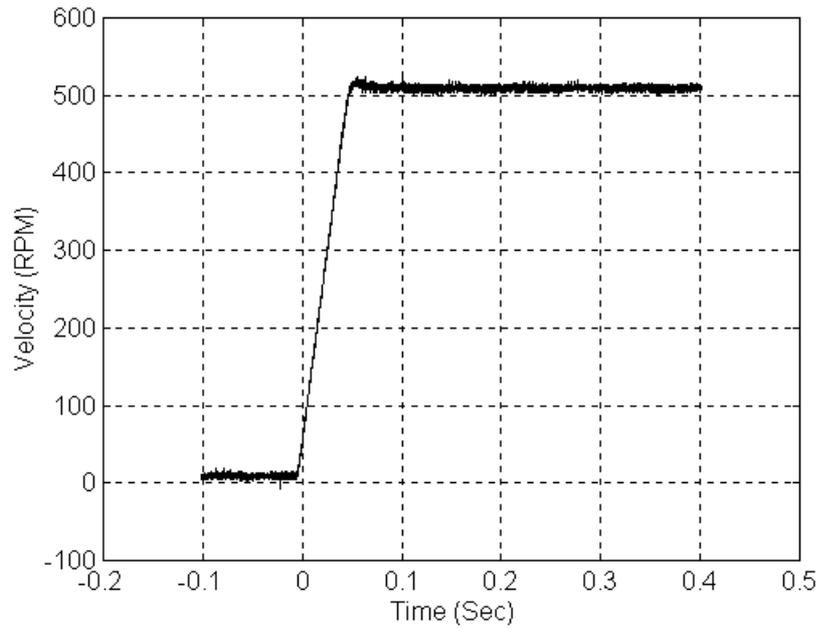


Рис. D-6. Реакция при ступенчатом изменении скорости, зависимость скорости от времени, VLGN (управляющий коэффициент скорости) = 48

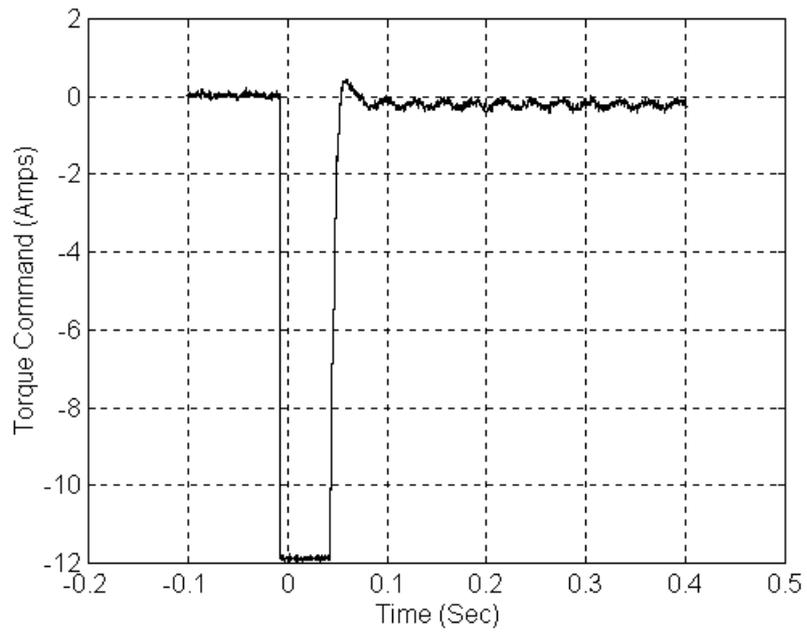


Рис. D-7. Реакция при ступенчатом изменении скорости, зависимость уставки момента вращения от времени, VLGN (управляющий коэффициент скорости) = 48

Реакция, показанная на рис. D-6 и D-7, является хорошей.

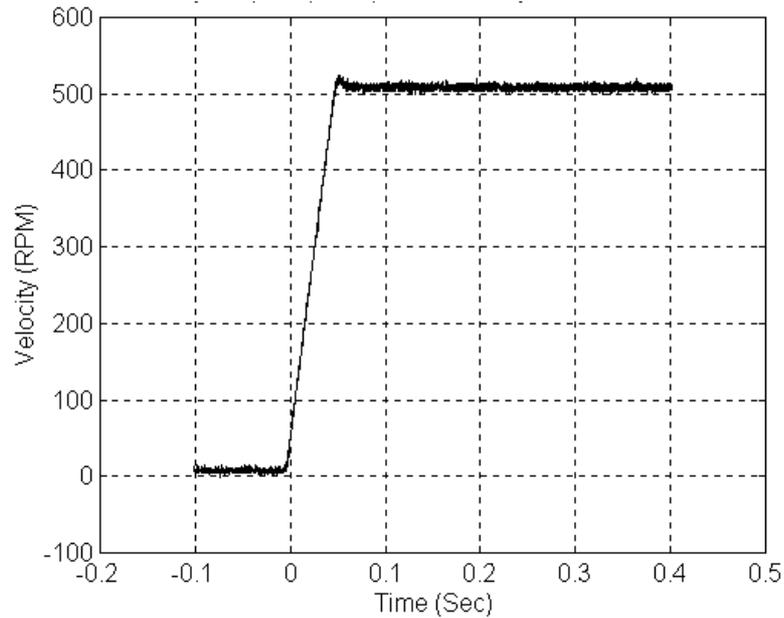


Рис. D-8. Реакция при ступенчатом изменении скорости, зависимость скорости от времени, VLGN (управляющий коэффициент скорости) = 64

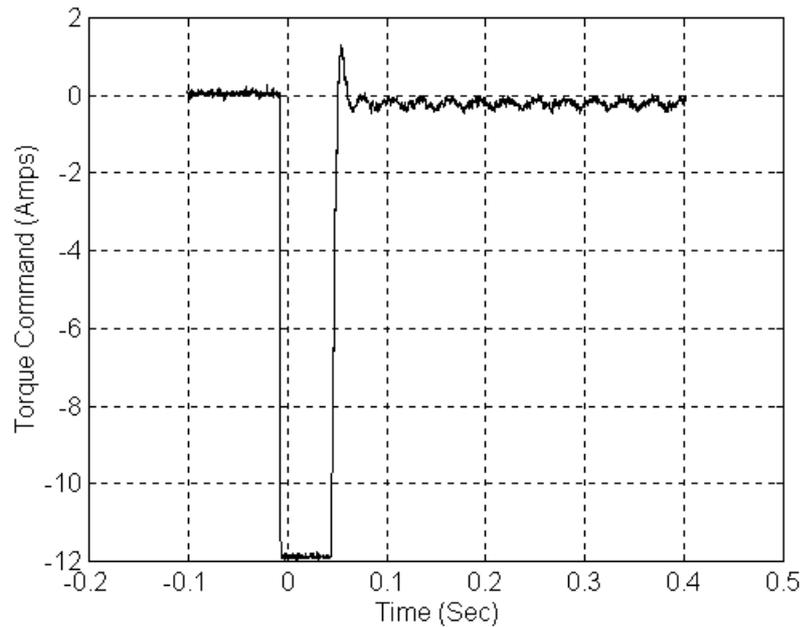
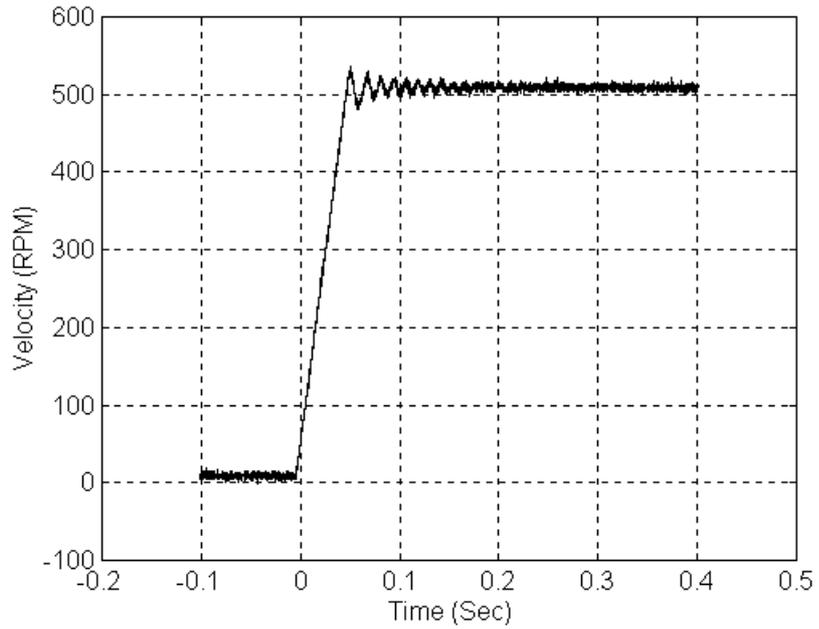
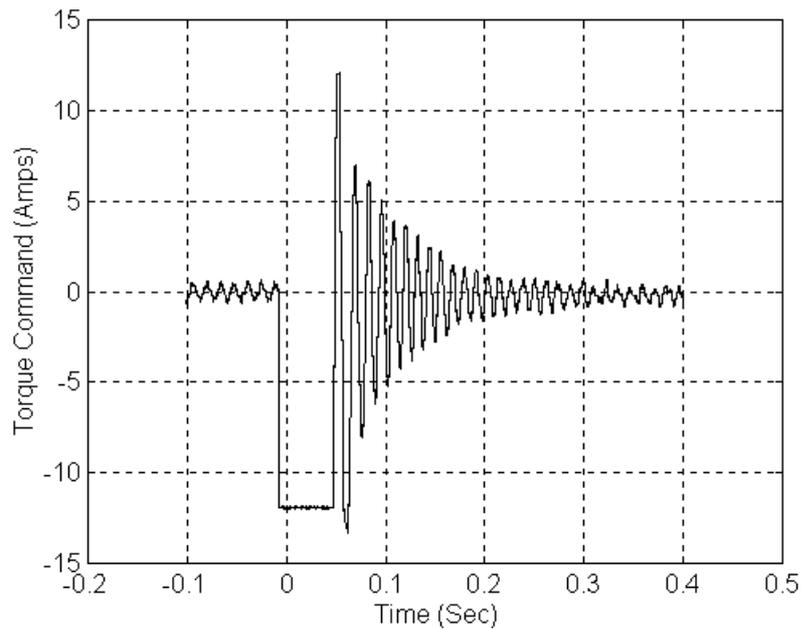


Рис. D-9. Реакция при ступенчатом изменении скорости, зависимость уставки момента вращения от времени, VLGN (управляющий коэффициент скорости) = 64

Реакция, показанная на рис. D-8 и D-9, является приемлемой.



**Рис. D-10. Реакция при ступенчатом изменении скорости, зависимость скорости от времени, VLGN (управляющий коэффициент скорости) = 208**



**Рис. D-11. Реакция при ступенчатом изменении скорости, зависимость уставки момента вращения от времени, VLGN (управляющий коэффициент скорости) = 208**

Реакция, показанная на рис. D-10 и D-11, является лишь в минимальной степени устойчивой и неприемлема для большинства применений. Все графики показаны только как примеры.

## Настройка контура позиционирования

Самым первым шагом при настройке контура позиционирования является обеспечение устойчивости контура скорости и его реакции, приемлемой для применения. Методы настройки контура скорости см. в предыдущем разделе.

### Предварительные установки контура позиционирования для проведения настройки

1. Если используются установки контура управления Standard Mode (Стандартный режим), то необходимо задать конфигурацию User Unit (Пользовательские единицы) и Counts (счет) так, чтобы она соответствовала механической конфигурации оси. См. более детальную информацию в описании и в примерах главы о конфигурации.
2. Установите Velocity (Скорость) на 10В, как описано в главе о конфигурации.
3. Установите выбор Integrator Mode (Режим интегратора) на "OFF" ("ВЫКЛ.").
4. Установите Feed Forward % (% предустановленной скорости) на нуль.
5. Установите Position Error Limit (Предел ошибки по положению) близко к максимуму. Максимум составляет 60 000 (Пользовательские единицы / импульсы).

### Установка Position Loop Gain (Коэффициент усиления позиционирования)

Контур управления позиционированием первоначально представляет собой алгоритм "PI" регулятора (пропорциональный, интегральный) с дополнительной пред установкой подачи. Настройка контура позиционирования начинается с установки пропорционального коэффициента (Pos Loop TC), чтобы обеспечить устойчивую реакцию с достаточным коэффициентом усиления (полоса пропускания), отвечающим требованиям по профилю движения. Установив Integrator Mode (Режим интегратора) на "OFF" ("ВЫКЛ."), как описано в предыдущем разделе, получаем чисто пропорциональный контур управления. Предлагается два метода установки пропорционального коэффициента усиления (Pos Loop TC).

### Метод 1 получения пропорционального коэффициента контура позиционирования

Расчет пропорционального коэффициента контура позиционирования предполагает, что механическая конструкция машины имеет достаточно широкую полосу пропускания, чтобы быть устойчивой, и что все резонансные частоты лежат выше полосы пропускания, требуемой для профиля движения.

### Терминология

Значительное несоответствие между моментами инерции нагрузки и двигателя может стать причиной **РЕЗОНАНСА** в системе. Резонанс представляет собой колебательное движение, вызываемое механическими ограничениями; его появлению способствуют люфты передач или вращательные колебания механических узлов, таких как муфты или валы. Резонанс устраняют, улучшая механические характеристики, уменьшая расхождение моментов инерции нагрузки и двигателя или снижая коэффициенты усиления сервопривода (снижаются рабочие характеристики).

**ПОЛОСА ПРОПУСКАНИЯ** является характеристикой качества, используемой для сравнения системы управления или механической эффективности. Если частота команд возрастает, то отклик системы начинает запаздывать. Полоса пропускания определяется как диапазон частот, выше которого реакция системы (коэффициент передачи) составляет не менее 70% (-3 децибела) требуемой уставки.

## Широкая полоса пропускания

- Позволяет сервоприводу с большей точностью воспроизводить требуемое движение.
- Позволяет точно отслеживать острые углы пути движения и использовать высокие скорости цикла
- Позволяет устранять искажения момента вращения, создаваемые механикой или внешними воздействиями, повышая точность системы
- Делает систему чувствительной к резонансу на частотах вблизи или ниже полосы пропускания

Реакцией чисто пропорциональной системы, которая возникает при установке параметра *Integrator Mode (Режим интегратора)* на "OFF", является экспонента. Постоянная времени экспоненты определяет 68% остатка. Например, начиная с нулевой скорости, контуру позиционирования, реагирующему на команду изменения, требуется одна постоянная времени, чтобы достичь 68% заданного значения скорости. В течение второй постоянной времени остаток уменьшится еще на 68%. Последующие постоянные времени уменьшают остаток на 68%. Например,  $100\% - 68\%$  (одна постоянная времени)  $= 32\%$ ,  $32\%(68\%)=21.8\%$ ,  $68\%$  (первая постоянная времени)  $+ 21.8\%$  (вторая постоянная времени)  $= 89.8\%$ . Две постоянные времени дают 89.8% требуемого значения команды. Три постоянные времени определяют 96.7% уставки. Четыре постоянные времени дают 98.9%. Как правило, трех постоянных времени достаточно для большинства применений, связанных с движением.

Знание постоянных времени можно использовать для описания требуемой реакции системы. Например, если наибольшее ускорение, требуемое в профиле скорости, должно происходить в течение 200 мс, то отклик длительностью 200 мс на изменение уставки будет представлять собой 98.9% уставки за три постоянных времени. Деление 200 мс на 3 дает постоянную времени около 67 мс. **Поле конфигурации Pos Loop TC представляет собой одну постоянную времени в мс.** В приведенном выше примере одна постоянная времени равна 67 мс.

## Метод 2 получения пропорционального коэффициента контура позиционирования

Этот метод аналогичен методу настройку контура скорости, приведенному выше. Используйте осциллограф и постепенно уменьшайте значение Pos Loop TC (постоянная времени позиционирования). Отслеживание рабочих характеристик осуществляется через аналоговый выход *Motor Velocity (Скорость двигателя)*.

## Информация о запуске и настройке аналоговых сервосистем

Здесь рассматриваются два основных раздела:

- Проверка правильности работы датчика начальной позиции, входов пере регулирования и направления вращения двигателя
- Скорость при Max Cmd, постоянная времени позиционирования и коэффициент упреждения по скорости.

## Процедуры запуска системы интерфейса режима аналоговой скорости

### Процедуры запуска

1. Подсоедините двигатель к сервоусилителю интерфейса аналоговой скорости в соответствии с рекомендациями изготовителя.
2. Подсоедините реле **Drive Enable (Привод включить)** и выходы **Velocity Command (Команда скорости)** модуля DSM314 к сервоусилителю. Подключите устройство обратной связи (инкрементный импульсный датчик положения) к входам датчика положения на модуле Motion Mate DSM314.

**Примечание:** Если это подключение выполнено неправильно или если имеет место люфт в подсоединении к устройству обратной связи, то при подаче команды движения может возникнуть ошибка **Out of Sync (Потеря синхронизации)**.

3. Подключите выход сервоусилителя Ready (Готовность) (если имеется) к входу модуля DSM314 Drive Ready (Готов к запуску) (IN\_4). Этот сигнал должен стать равным 0 В при готовности усилителя управлять сервоприводом. **DSM начинает проверять вход Drive Ready (Готов к запуску) раз в секунду после того, как реле включения привода включилось в ответ на бит %Q Enable Drive (Включить привод)**. Если сервоусилитель не может обеспечить соответствующий выход готовности, то этот вход на DSM314 должен быть подключен к 0 В, или при конфигурации модуля вход Drive Ready (Готов к запуску) может быть отключен. Если используется датчик **начальной** позиции (24 В пост. тока), то подключите его к соответствующему входу модуля DSM314. Этот датчик **начальной** позиции должен быть подключен так, чтобы он был в состоянии ALWAYS ON (ВСЕГДА ЗАМКНУТ), когда ось находится с отрицательной стороны начальной позиции, и в состоянии ALWAYS OFF (ВСЕГДА РАЗОМКНУТ), когда ось находится с положительной стороны начальной позиции.
4. Используйте программное обеспечение конфигурации для задания требуемых конфигурируемых параметров. Сохраните конфигурацию на ПЛК.
5. Включите бит **%Q Enable Drive (Включить привод)** и выставьте код команды для **Force D/A Output (Включить Цифровой / Аналоговый Выход)** равным нулю в таблице %AQ. Убедитесь, что сервоусилитель включен (двигатель должен проявлять удерживающий момент вращения). Если двигатель вращается, то отрегулируйте смещение уставки усилителя так, чтобы движение прекратилось. Примечание: Бит **%Q Enable Drive (Включить привод)** должен быть в состоянии ON (ВКЛ.), чтобы действовала команда **Force D/A Output (Включить Цифровой / Аналоговый Выход)**.
6. Выдайте командный код для **Force D/A Output (Включить Цифровой / Аналоговый Выход)**, равный +3200 (+1.0 В). Убедитесь, что двигатель вращается в требуемом ПОЛОЖИТЕЛЬНОМ направлении (на базе установки параметра конфигурации **Axis Direction (Направление оси)**) и что **Actual Velocity (Текущая скорость)**, приведенная в таблице %AI модуля DSM314, ПОЛОЖИТЕЛЬНАЯ. Если направление вращения

двигателя неправильное, то обратитесь к производителю сервоусилителя, чтобы получить инструкции по изменению ситуации. Параметр *Axis Direction (Направление оси)* в программе конфигурации может быть также использован для обмена местами положительного и отрицательного направлений оси. Если двигатель вращается в ПОЛОЖИТЕЛЬНОМ направлении, но модуль DSM314 сообщает, что *Actual Velocity (Текущая скорость)* является ОТРИЦАТЕЛЬНОЙ, то необходимо поменять местами входы каналов А и В датчика положения.

7. Запишите значение текущей скорости двигателя, выданное модулем Motion Mate DSM314 при уставке скорости 1.0 В. Умножьте эту скорость на 10 и обновите вход *Velocity at Max Cmd (Скорость при максимальной команде)* в конфигурации модуля DSM314, если требуется. Первоначально установите параметр конфигурации *Pos Loop Time Constant (Постоянная времени позиционирования) (0.1 мс)* на верхнее значение (обычно это 100 мс или значение 1000 при конфигурации).
8. Установите %Q бит *Jog Plus (Толчок плюс)*. Убедитесь, что сервопривод двигается в правильном направлении и что значение *Actual Velocity (Текущая скорость)*, передаваемое модулем Motion Mate DSM314 в %AI таблицу, соответствует сконфигурированной *Jog Velocity (Скорость передвижения)*. Если программы движения будут использовать ускорение, превышающее *данное Jog Acceleration (Ускорение передвижение)*, то может оказаться необходимым увеличить *Jog Acceleration* так, чтобы команды *Abort All Moves (Прекратить все движения)* и *Normal Stop (Нормальный останов)* могли действовать, как положено.
9. При бите *Drive Enabled (Привод включен)* %Q, установленном в состояние ON (ВКЛ.), и при отсутствии задания серводвижения отрегулируйте смещение команды сервопривода для нулевой *Position Error (Ошибка по положению)*. Интегратор в ходе этой процедуры должен быть выключен.
10. Проверьте правильность работы цикла *Find Home (Начальное позиционирование)* включением на короткое время бита %Q *Find Home* (бит %Q *Drive Enabled (Привод включен)*) также должен быть установлен). Ось должна двигаться в сторону переключателя начальной позиции при сконфигурированном значении *Find Home Velocity (Скорость поиска начальной позиции)*, затем подходить к опорной точке датчика положения при сконфигурированном значении *Final Home Velocity (Скорость подхода к начальной позиции)*. Если требуется, подстройте сконфигурированные скорости и положение *датчика начальной позиции*, чтобы обеспечить согласованные действия. Последний импульс *датчика начальной позиции* ДОЛЖЕН иметь место не менее, чем за 10 мс до появления импульса маркера датчика положения. Физическое положение *Home Position (Начальная позиция)* может быть затем подстроено изменением значения *Home Offset* (начальное смещение) в программе конфигурации.
11. Проследите работу сервопривода и используйте %Q-биты *Jog Plus (Толчок плюс)* и *Jog Minus (Толчок минус)* для перемещения аналогового серводвигателя в каждом направлении. *Постоянная времени позиционирования* может быть на время изменена, если в таблицу %AQ ввести правильный командный код. Для большинства систем эту *Постоянную времени позиционирования* можно уменьшать до тех пор, пока не будет замечена некоторая нестабильность сервопривода; после этого следует увеличить ее значение на, примерно, 50%. После задания правильной постоянной времени необходимо обновить конфигурацию модуля DSM314, используя программное обеспечение конфигурации. Для оптимальной реакции сервопривода можно также задать ненулевое значение для *Velocity Feedforward (Коэффициент упреждения по скорости)* (обычно 90 - 100%).

**Примечание:** Для правильной работы сервопривода вход конфигурации для *Velocity at Max Cmd (Скорость при максимальной команде)* должен быть установлен на значение текущей скорости сервопривода (в пользовательские единицы/секунды), которое имеет место при команде скорости 10 В на усилителе.

# Процедуры запуска системы интерфейса режима аналогового момента вращения

## Процедуры запуска

1. Подсоедините двигатель к сервоусилителю интерфейса аналогового момента вращения в соответствии с рекомендациями изготовителя. **Примечание:** Усилитель должен быть сконфигурирован так, чтобы работать с напряжением ( $\pm 10$  В), которое соответствует моменту вращения двигателя.
2. Подсоедините реле **Drive Enable (Привод включить)** и выходы **Torque Command (Команда момента вращения)** модуля DSM314 к сервоусилителю. Подключите устройство обратной связи (инкрементный импульсный датчик положения) к входам датчика положения на модуле Motion Mate DSM314.

**Примечание:** Если это подключение выполнено неправильно или если имеет место проскальзывание в подсоединении к устройству обратной связи, то при подаче команды движения может возникнуть ошибка **Out of Sync (Потеря синхронизации)**.

3. Подключите выход сервоусилителя Ready (Готовность) (если имеется) к входу модуля DSM314 Drive Ready (Готов к запуску) (IN\_4). Этот сигнал должен стать равным 0 В при готовности усилителя управлять сервоприводом. **DSM начинает проверять вход Drive Ready (Готов к запуску) раз в секунду после того, как реле включения привода включилось в ответ на бит %Q Enable Drive (Включить привод).** Если сервоусилитель не может обеспечить соответствующий выход готовности, то этот вход на DSM314 должен быть подключен к 0 В, или при конфигурации модуля вход Drive Ready (Готов к запуску) может быть отключен. Если используется датчик **начальной** позиции (24 В пост. тока), то подключите его к соответствующему входу модуля DSM314. Этот датчик **начальной позиции** должен быть подключен так, чтобы он был в состоянии ALWAYS ON (ВСЕГДА ЗАМКНУТ), когда ось находится с отрицательной стороны начальной позиции, и в состоянии ALWAYS OFF (ВСЕГДА РАЗОМКНУТ), когда ось находится с положительной стороны начальной позиции.
4. Используйте программное обеспечение конфигурации для задания требуемых конфигурируемых параметров. Сохраните конфигурацию на ПЛК. Специфическими параметрами, которые требуются пользователю, являются следующие:

**Analog Servo Command (Аналоговая Сервокоманда).** При конфигурации следует установить на "Момент вращения". Это не является значением по умолчанию. Этот параметр конфигурации определяет то, что модуль выдает на аналоговый выход команду момента вращения. **Примечание:** Для работы режима аналогового момента вращения требуется микропрограмма модуля DSM версии 3.0 или более поздняя.

**Velocity at Max Command (Скорость при Максимальной Команде).** Значение скорости при максимальной команде (уставки), задаваемое при конфигурации, определяет максимальную скорость, которая может быть дана на выполнение сервоприводу. На ранних этапах настройки рекомендуется задать это значение сравнительно небольшим. Это позволит увеличивать его в системе постепенно. После того как основная работа и настройка проведены, максимальное значение может быть установлено на такое значение, которое определяется либо ограничениями процесса, либо установкой усилителя / двигателя.

**Torque Limit (Предел по моменту вращения).** Предел по моменту вращения определяет команду максимального аналогового момента вращения, которая может быть направлена на сервоусилитель. На ранних этапах настройки рекомендуется задать это значение сравнительно небольшим. Этот предел по моменту вращения устанавливается с использованием команды %AQ. Информацию по этой команде см. в главе 5. После проверки основной работы значение момента вращения может быть установлено на то, что требуется для применения.

**Advanced Tuning Parameters (Расширенные параметры настройки).** Раздел расширенных параметров настройки содержит много параметров, которые используются для конфигурации режима момента вращения с целью обеспечения правильной работы. Расширенные параметры настройки подробно обсуждаются в главе конфигурации. Чтобы получить полную информацию, см. эту главу. Параметрами настройки, представляющими особый интерес, являются следующие:

**Параметр настройки 6.** Задает разрешение датчика положения. Этот параметр используется только в режиме момента вращения. Чтобы работа в режиме момента вращения происходила правильно, это значение должно быть равно числу импульсов импульсного датчика положения, создаваемых устройством обратной связи двигателя за оборот. Пользователь может определить это число из спецификации устройства обратной связи. В качестве дополнительной проверки пользователь может подсоединить устройство обратной связи к DSM и вручную повернуть вал двигателя на один оборот. Значение в данных DSM %AI для текущей позиции должно приблизительно соответствовать значению данного параметра (некоторое различие может быть из-за точности поворота вала вручную на один оборот). Допустимый диапазон составляет 100 - 32767 импульсов на оборот. Значение по умолчанию - 4096 импульсов на оборот.

**Параметр настройки 7.** Задает коэффициент передачи пропорционального регулятора скорости. Этот параметр используется только в режиме момента вращения. Коэффициент передачи пропорционального регулятора умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к пропорциональному члену. Правильное задание этого значения определяет, насколько хорошо регулятор скорости будет работать в системе управления. В приведенном далее разделе обсуждается установка этого значения.

**Параметр настройки 8.** Задает коэффициент передачи интегрального регулятора скорости. Этот параметр используется только в режиме момента вращения. Интегральный коэффициент умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к интегральному члену. Правильное задание этого значения определяет, насколько хорошо регулятор скорости будет работать в системе управления. В приведенном далее разделе обсуждается установка этого значения.

**Примечание:** Для правильной работы сервопривода вход конфигурации для **Encoder Resolution (Разрешение датчика положения)** ДОЛЖЕН быть установлен на правильное значение комплекса сервоусилитель / двигатель. Если это значение задано неправильно, то может возникнуть неустойчивость.

5. Включите бит %Q *Enable Drive (Включить привод)* и выставьте код команды для *Force Servo Velocity (Включить скорость сервопривода)* равным нулю в таблице %AQ. Убедитесь, что сервоусилитель включен (двигатель должен проявлять удерживающий момент вращения). Если двигатель вращается, то отрегулируйте усилитель так, чтобы движение прекратилось.
6. При первом проведении указанной ниже операции убедитесь, что вал двигателя не подсоединен к нагрузке. Пользователю требуется проверить базовые функции управления. Выдайте командный код *Force Servo Velocity (Включить скорость сервопривода)* на 10 обор. в мин. Убедитесь, что двигатель вращается в требуемом ПОЛОЖИТЕЛЬНОМ направлении (на базе установки параметра конфигурации **Axis Direction (Направление оси)**) и что *Actual Velocity (Текущая скорость)*, приведенная в таблице %AI модуля DSM314, ПОЛОЖИТЕЛЬНАЯ. Если направление вращения двигателя неправильное, то обратитесь к производителю сервоусилителя, чтобы получить инструкции по изменению ситуации. Параметр **Axis Direction (Направление оси)** в программе конфигурации может быть также использован для обмена местами положительного и отрицательного направлений оси. Если двигатель вращается в ПОЛОЖИТЕЛЬНОМ направлении, но модуль DSM314 сообщает, что *Actual Velocity (Текущая скорость)* является ОТРИЦАТЕЛЬНОЙ, то необходимо поменять местами входы каналов А и В датчика положения.

7. При %Q-бите *Drive Enabled (Привод включен)*, установленном в состояние ON (ВКЛ.), и при отсутствии задания серводвижения отрегулируйте сервопривод так, чтобы движение отсутствовало. Интегральная часть контура скорости должна быть установлена на 0, чтобы правильно выполнить этот этап.
8. После получения правильной базовой операции требуется настройка контура скорости. В разделе "Настройка контура скорости в режиме момента вращения" описана основная процедура настройки контура. ПРИМЕЧАНИЕ: Процедура настройки для регуляторов скорости в режиме момента вращения ОТЛИЧАЕТСЯ от процедуры для регуляторов скорости в цифровом режиме. Пользователь НЕ должен делать настройку контура позиционирования до тех пор, пока не завершена настройка контура скорости.
9. После завершения настройки регуляторов скорости может быть закончена настройка и установка контура позиционирования. Первоначально установите параметр конфигурации *Pos Loop Time Constant (Постоянная времени позиционирования) (0.1 мс)* на верхнее значение (обычно это 100 мс или значение 1000 при конфигурации).
10. Установите %Q бит *Jog Plus (Толчок плюс)*. Убедитесь, что сервопривод двигается в правильном направлении и что значение *Actual Velocity (Текущая скорость)*, передаваемое модулем Motion Mate DSM314 в %AI таблицу, соответствует сконфигурированной *Jog Velocity (Скорость передвижения)*. Если программы движения будут использовать ускорение, превышающее *данное Jog Acceleration (Ускорение передвижения)*, то может оказаться необходимым увеличить *Jog Acceleration* так, чтобы команды *Abort All Moves (Прекратить все движения)* и *Normal Stop (Нормальный останов)* могли действовать, как положено.
11. Проверьте правильность работы цикла *Find Home (Начальное позиционирование)* включением на короткое время бита %Q *Find Home* (бит %Q *Drive Enabled (Привод включен)*) также должен быть установлен). Ось должна двигаться в сторону переключателя начальной позиции при сконфигурированном значении *Find Home Velocity (Скорость поиска начальной позиции)*, затем подходить к опорной точке датчика положения при сконфигурированном значении *Final Home Velocity (Скорость подхода к начальной позиции)*. Если требуется, подстройте сконфигурированные скорости и положение *датчик начальной позиции*, чтобы обеспечить согласованные действия. Последний импульс *датчик начальной позиции* должен иметь место не менее, чем за 10 мс до появления импульса маркера датчика положения. Физическое положение *Home Position (Начальная позиция)* может быть затем подстроено изменением значения *Home Offset (начальное смещение)* в программе конфигурации.
12. Проследите работу сервопривода и используйте %Q-биты *Jog Plus (Толчок плюс)* и *Jog Minus (Толчок минус)* для перемещения аналогового серводвигателя в каждом направлении. *Постоянная времени позиционирования* может быть на время изменена, если в таблицу %AQ ввести правильный командный код. Для большинства систем эту *Постоянную времени позиционирования* можно уменьшать до тех пор, пока не будет замечена некоторая нестабильность сервопривода; после этого следует увеличить ее значение на, примерно, 50%. После задания правильной постоянной времени необходимо обновить конфигурацию модуля DSM314, используя программное обеспечение конфигурации. Для оптимальной реакции сервопривода можно также задать ненулевое значение для *Velocity Feedforward (Коэффициент упреждения по скорости)* (обычно 90 - 100%).

**Примечание:** Для правильной работы сервопривода вход конфигурации для *Velocity at Max Cmd (Скорость при максимальной команде)* должен быть установлен на максимальную скорость сервопривода (пользовательские единицы / секунда), которую позволяет система или процесс.

## Настройка контура скорости в режиме момента вращения

Правильным методом настройки контура скорости является разделение контуров скорости и позиции. Для такого разделения должен быть использован метод прямой отправки команд скорости без использования управления контуром позиции. Модуль DSM имеет несколько режимов, которые позволяют пользователю направлять команду скорости непосредственно на контур скорости. Ниже дано описание двух методов.

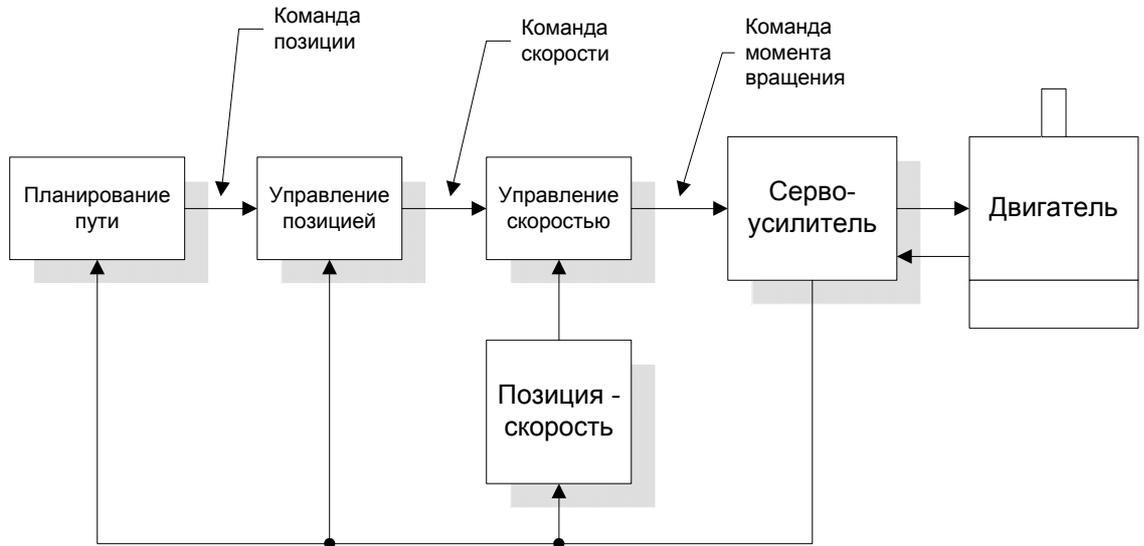


Рис. D-12. Блок-схема контуров управления интерфейса момента вращения в аналоговом режиме

### Метод #1

Команда немедленного действия `%AQ Force Servo Velocity (Включить скорость сервопривода)` (34h) выдает команду скорости непосредственно в контур скорости. Эта команда отличается от команды `Move at Velocity (Движение на скорости)`, которая для генерации команды использует контур позиционирования. Это представляется важным, поскольку на данном этапе процесса настройки контур позиционирования не должен взаимодействовать с контуром скорости. Команда `%AQ Force Servo Velocity (Включить скорость сервопривода)` позволяет пользователю выдать ступенчатое изменение скорости. Задание ступенчатого изменения скорости используется для выработки реакции контура скорости на ступенчатое воздействие. Пользователь должен учесть, что при задании ступенчатого изменения скорости величина ускорения ограничена только полосой пропускания контура скорости. В некоторых применениях это может привести к повреждению управляемого устройства из-за высокого значения ускорения.

### Метод #2

В некоторых применениях метод #1 приводит к сильному удару на управляемом устройстве. В этих случаях требуется другой метод генерации команды скорости. Этот метод требует, чтобы пользователь установил контур позиционирования в конфигурацию открытого контура. Это достигается установкой ***Position Loop Time Constant (Постоянная времени позиционирования)*** на нуль, а ***Velocity Feedforward Gain (Коэффициент упреждения по скорости)*** – на 100%. После этого можно использовать команду `Move at Velocity (Движение на скорости)` или программу движения, чтобы выдавать команды скорости на сервопривод.

1. Описанная далее процедура позволяет настроить регулятор скорости. Предполагается, что первоначально это выполняется с двигателем без подсоединения приводимой в движение нагрузки. Настройка при подсоединенной нагрузке проводится позднее. Первым параметром, требующим настройки, является ***Velocity Loop Proportional Gain (Пропорциональный коэффициент обратной связи по скорости)***. Пропорциональный коэффициент обратной связи по скорости умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к пропорциональному члену. В начале процесса пропорциональный член должен быть установлен небольшим. Хотя это значение зависит от полосы пропускания управляемого сервоусилителя, но значение по умолчанию, равное 1500, представляется хорошей исходной точкой. Тем не менее, если сервоусилитель имеет узкую полосу пропускания или очень чувствителен к изменениям уставки момента вращения, то может оказаться необходимым установить меньшее начальное значение. Процедура настройки позволяет пользователю итерационным путем придти к конечному значению. Таким образом, если существуют какие-либо опасения, то следует начинать с очень малого значения (например, 100).
2. Выберите метод, позволяющий ввести команду скорости в контур скорости. Методы #1 и #2 (см. выше) являются примерами методов, предназначенных для такой задачи.
3. Подсоедините осциллограф к аналоговым выходам для скорости двигателя от сервоусилителя.
4. Как обсуждалось выше, установите начальное значение пропорционального коэффициента обратной связи по скорости.
5. Выдайте ступенчатое изменение скорости. На этом этапе ступенчатое изменение должно быть относительно небольшим по сравнению с полной скоростью машины. Хорошим началом могло бы быть 10 – 20% номинальной скорости машины.
6. Наблюдайте скорость двигателя на осциллографе. Целью является получение реакции критического затухания в контуре скорости. На данном этапе наиболее вероятно будет установившаяся ошибка скорости. Это ожидаемый результат на данном этапе процесса настройки. Интегральный член скорости вводится на этапах далее, чтобы устранить эту ошибку. Обратите особое внимание на 1-й возникающий пик и на любые колебания, возникающие в сигнале скорости.
7. Увеличивайте понемногу ***Пропорциональный коэффициент обратной связи по скорости*** и повторяйте шаги 5 и 6 до тех пор, пока не будет получена требуемая реакция. В зависимости от применения это может быть система с критическим затуханием или с небольшим пере регулированием. Общим правилом является следующее: чем ниже значение ***пропорционального коэффициента обратной связи по скорости*** при удовлетворении требований к системе, тем надежнее работает управление. Следует внимательно наблюдать сигнал обратной связи по скорости. В некоторых применениях подъем ***коэффициента обратной связи по скорости*** до достаточно большого значения, чтобы создать неустойчивость, может привести к повреждению машины. Если в сигнале обратной связи по скорости двигателя начнутся колебания до достижения этой точки, то следует уменьшить величину ***пропорционального коэффициента обратной связи по скорости***.
8. Следующим параметром, который должен быть настроен, является ***постоянная времени интегратора контура позиционирования***. Интегральный коэффициент обратной связи по скорости умножается на ошибку по скорости (заданная скорость – фактическая скорость), чтобы создать часть задания момента вращения, относящегося к интегральному члену. Интегральный член коэффициента обычно используется для компенсации установившейся ошибки по скорости. Для начала процесса настройки необходимо установить ***постоянную времени интегратора***

**контура позиционирования** на ноль. Процедура настройки состоит в медленном увеличении этого значения до тех пор, пока не будет устранена установившаяся ошибка без возникновения значительного пере регулирования или чрезмерного колебательного процесса в реакции системы.

9. Выберите метод, позволяющий ввести команду скорости в контур скорости. Методы #1 и #2 (см. выше) являются примерами методов, предназначенных для такой задачи.
10. Подсоедините осциллограф к аналоговым выходам для скорости двигателя от сервоусилителя.
11. Как обсуждалось выше, установите начальное значение **постоянной времени интегратора контура позиционирования**.
12. Выдайте ступенчатое изменение скорости. На этом этапе ступенчатое изменение должно быть относительно небольшим по сравнению с полной скоростью машины. Хорошим началом могло бы быть 10 – 20% номинальной скорости машины.
13. Наблюдайте скорость двигателя на осциллографе. Целью является устранение установившейся ошибки без возникновения значительного перерегулирования или колебательного процесса. При настройке интегрального члена обращайтесь особое внимание на любые возникающие колебания в реакции системы. Чрезмерные колебания являются признаком нестабильности контура управления из-за слишком высокого значения интегрального члена.
14. Увеличивайте понемногу **постоянную времени интегратора контура позиционирования** и повторяйте шаги 12 и 13 до тех пор, пока не будет получена требуемая реакция. В зависимости от применения это может быть система с критическим затуханием или с небольшим пере регулированием. Общим правилом является следующее: чем ниже значение **постоянной времени интегратора контура позиционирования** при удовлетворении требований к системе, тем надежнее работает управление. Следует внимательно наблюдать сигнал обратной связи по скорости. В некоторых применениях увеличение **постоянной времени интегратора контура позиционирования** до достаточно большого значения, может привести к неустойчивости системы, может привести к повреждению машины. Если в сигнале обратной связи по скорости двигателя начнутся колебания до достижения этой точки, то следует уменьшить величину **постоянной времени интегратора контура позиционирования**. На этом этапе основной контур скорости можно считать настроенным. Следующим шагом является подсоединение двигателя к нагрузке и настройка параметра **Velocity Loop Gain (управляющий коэффициент скорости)** совместно с нагрузкой.
15. При настроенном основном контуре скорости подсоедините двигатель к нагрузке. Параметр, **управляющий коэффициент скорости**, позволяет настроить реакцию контура скорости так, чтобы скомпенсировать нагрузку. В частности этот параметр изменяет полосу пропускания контура скорости. В качестве начальной точки используется формула, приведенная ниже.

## Уравнение 2

$$\text{управляющий коэффициент скорости} = \frac{J_1}{J_m} \cdot 16$$

где:

$J_1$  = момент инерции нагрузки

$J_m$  = момент инерции двигателя

Во многих случаях не требуется изменять *управляющий коэффициент скорости*, рассчитанный выше. Однако при некоторых применениях (например, когда возникает резонанс на машине) может оказаться необходимым подстроить это значение. Для изменения *управляющий коэффициент скорости* может быть использована описанная далее процедура.

16. Выберите метод, позволяющий ввести команду скорости в контур скорости. Методы #1 и #2 (см. выше) являются примерами методов, предназначенных для такой задачи.
17. Подсоедините осциллограф к аналоговым выходам для обратной связи по скорости двигателя от сервоусилителя.
18. Установите коэффициент обратной связи по скорости на ноль. Этот подход является "осторожным". Если о применении известно, что оно не имеет резонансных частот от нуля до, примерно, 250 Гц, то можно начинать с более высокого значения, но не выше значения, рассчитанного в уравнении 2 в данном разделе.
19. Выдайте ступенчатое изменение скорости. На этом этапе ступенчатое изменение должно быть относительно небольшим по сравнению с полной скоростью машины. Хорошим началом могло бы быть 10 – 20% номинальной скорости машины.
20. Наблюдайте скорость двигателя на осциллографе. Целью является получение реакции критического затухания в контуре скорости. Обратите особое внимание на любые колебания, которые могут происходить в сигнале обратной связи по скорости.
21. Увеличивайте понемногу *управляющий коэффициент скорости* и повторяйте шаги 19 и 20 до тех пор, пока в сигнале обратной связи по скорости двигателя не появится некоторая неустойчивость. **Примечание:** На этом этапе должна быть проявлена осторожность, чтобы возникающая неустойчивость не привела к повреждению машины. При достижении этого значения уменьшите *управляющий коэффициент скорости*, по крайней мере, на 15%. Общим правилом является следующее: чем ниже значение *управляющий коэффициент скорости* при удовлетворении требований к системе, тем надежнее работает управление. Следует внимательно наблюдать сигнал обратной связи по скорости. В некоторых применениях подъем *управляющий коэффициент скорости* до достаточно большого значения, чтобы создать неустойчивость, может привести к повреждению машины. При опасениях установите *управляющий коэффициент скорости* на более значения, рассчитанного в уравнении 1. Если в сигнале обратной связи по скорости двигателя начнутся колебания до достижения этой точки, то следует уменьшить величину *управляющий коэффициент скорости* и перейти к шагу 22.
22. На этом этапе контур скорости можно считать настроенным. Однако должна быть проверена робастность контура. Для такого теста введите ступенчатое изменение скорости с величиной ступеньки 20, 40, 60, 80 и 100% номинальной скорости машины. Наблюдайте сигналы скорости двигателя и команды момента вращения на предмет неустойчивости. При проявлении неустойчивости или резонанса следует уменьшить *управляющий коэффициент скорости* и повторить тест.

## Пример настройки контура скорости

На рисунках ниже показан пример настройки контура скорости. Чтобы начать этот процесс, должен быть настроен *пропорциональный коэффициент обратной связи по скорости*.

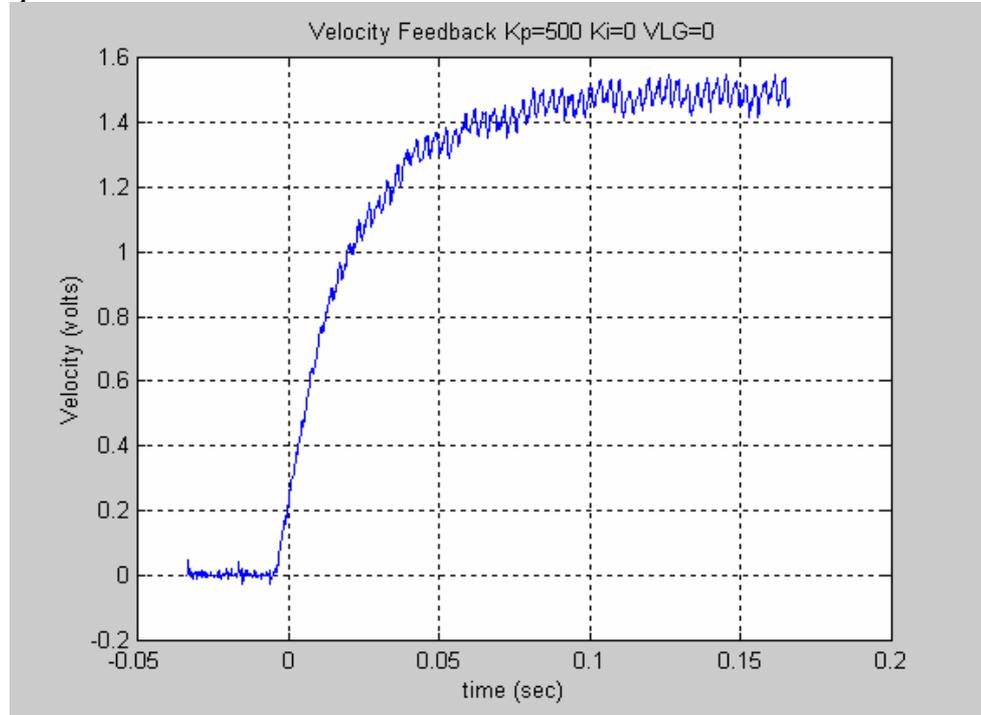


Рис. D-13. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 500$ ,  $K_i = 0$ ,  $VLGN$  (управляющий коэффициент скорости) = 0

Обратите внимание на то, что система имеет сравнительно медленный отклик. Кроме того, по отношению к требуемой скорости имеется установившаяся ошибка. Для более быстрой реакции можно увеличить *пропорциональный коэффициент обратной связи по скорости*.

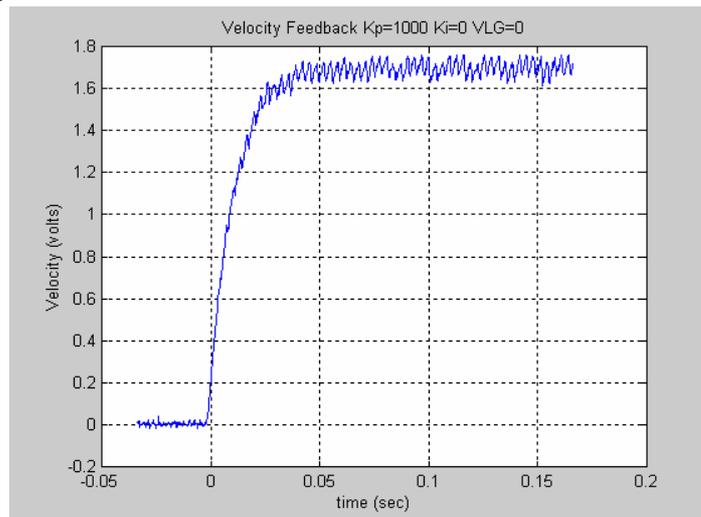


Рис. D-14. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 1000$ ,  $K_i = 0$ ,  $VLGN = 0$

*Пропорциональный коэффициент обратной связи по скорости* был увеличен на рисунке выше. Время нарастания уменьшилось. Однако система может быть еще улучшена добавлением дополнительного *пропорционального коэффициента обратной связи по скорости*. Установившаяся ошибка продолжает существовать.

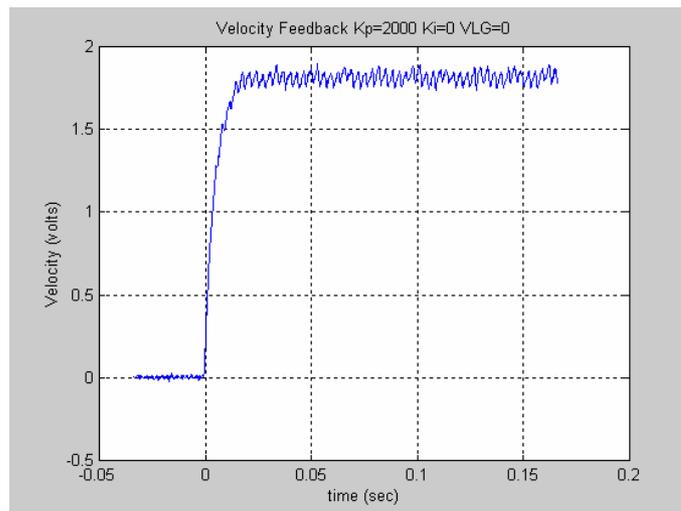


Рис. D-15. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  $K_p = 2000$ ,  $K_i = 0$ ,  $VLGN = 0$

*Пропорциональный коэффициент обратной связи по скорости* был снова увеличен. Реакция начинает выглядеть вполне приемлемо. Однако время нарастания может быть еще улучшено.

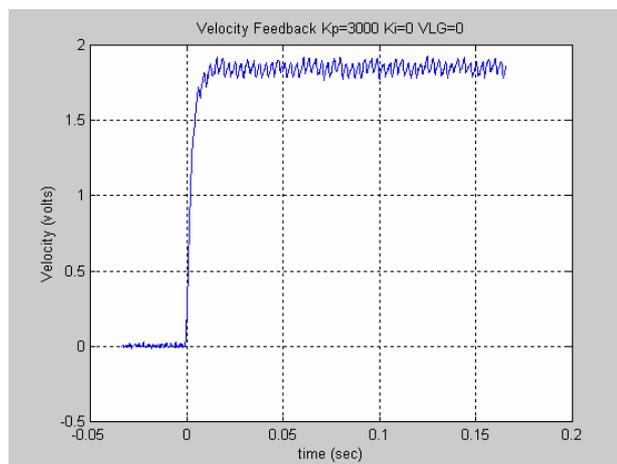


Рис. D-16. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  $K_p = 3000$ ,  $K_i = 0$ ,  $VLGN = 0$

Показанная на рисунке реакция представляется очень хорошей. Обратите внимание на небольшой пик в отклике. Продолжая экспериментировать с реакцией, увеличим еще раз *пропорциональный коэффициент обратной связи по скорости*.

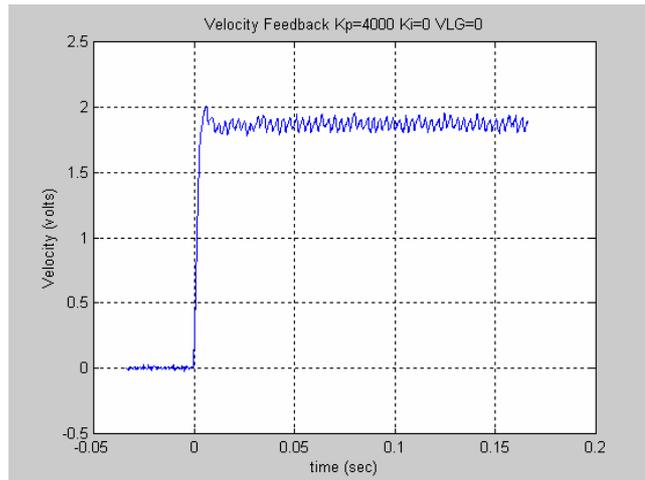


Рис. D-17. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 4000$ ,  $K_i = 0$ ,  $VLGN = 0$

Реакция, показанная на рисунке, демонстрирует небольшое перерегулирование. Такая реакция или та, что показана на предыдущем рисунке, являются приемлемыми для большинства применений. Однако настройка должна быть проведена с использованием всех возможностей машины. Все графики показаны только как примеры.

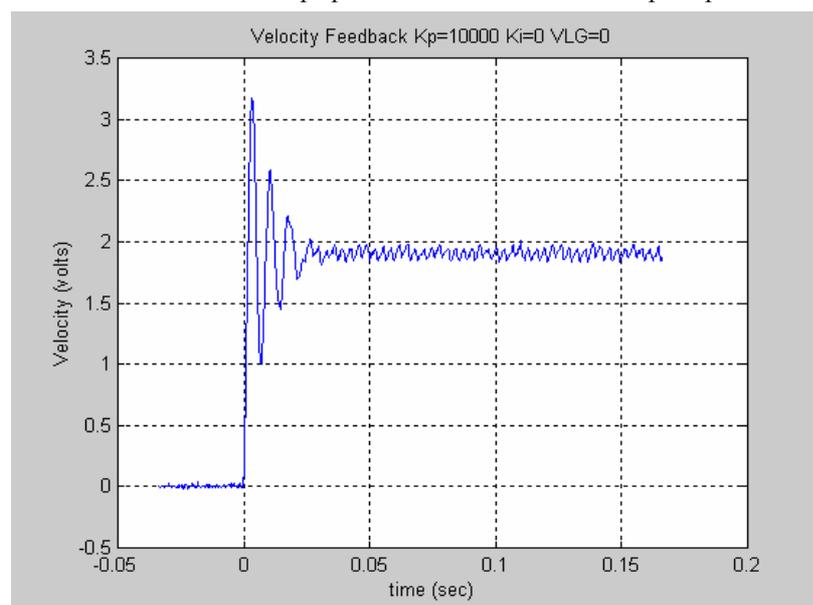


Рис. D-18. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 10000$ ,  $K_i = 0$ ,  $VLGN = 0$

На данном рисунке показана неприемлемая реакция системы. Контур проявляет признаки неустойчивости. Обратите внимание на перерегулирование и на колебательный процесс, следующий за первым пиком. Для получения более устойчивого отклика необходимо значительно уменьшить **пропорциональный коэффициент обратной связи по скорости**.

В данном примере в качестве нужного отклика системы выбирается **пропорциональный коэффициент обратной связи по скорости** ( $K_p$ ) = 4000. Это значение будет использоваться при настройке **интегрального коэффициента обратной связи по скорости**.

Первоначально **интегральный коэффициент обратной связи по скорости** равен нулю. Можно делать небольшие изменения с ним и наблюдать за откликом (реакцией).

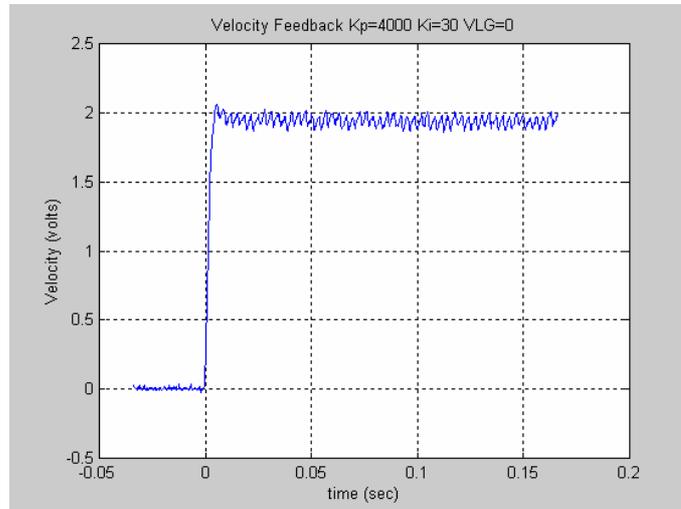


Рис. D-19. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  $K_p = 4000$ ,  $K_i = 30$ ,  $VLGN = 0$

Реакция выше показывает, что введение **интегрального коэффициента обратной связи по скорости** улучшило реакцию системы. В частности, установившаяся ошибка уменьшилась.

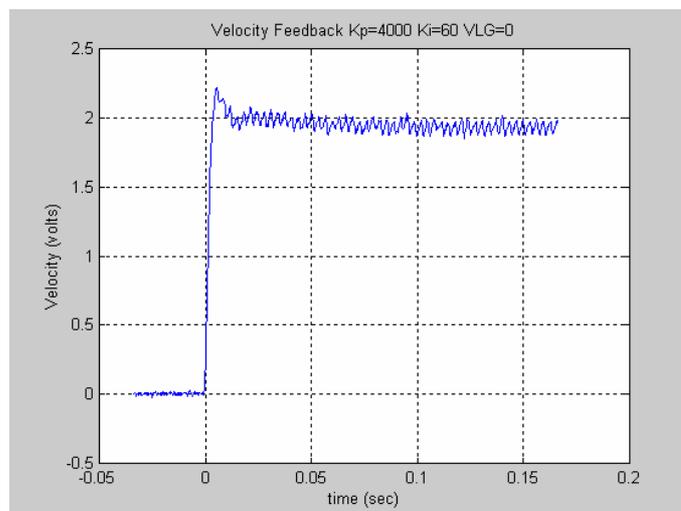


Рис. D-20. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  $K_p = 4000$ ,  $K_i = 60$ ,  $VLGN = 0$

При дальнейшем увеличении **интегрального коэффициента обратной связи по скорости** можно видеть появление перерегулирования, вызванного интегральным коэффициентом. Однако отклик, показанный на обоих вышеприведенных рисунках, считается приемлемым. Окончательное значение зависит от возможностей приводимой в движение нагрузки. В общем случае, чем ниже значение как **пропорционального**, так и **интегрального коэффициентов обратной связи по скорости** при удовлетворении требований к системе, тем надежнее работает управление.

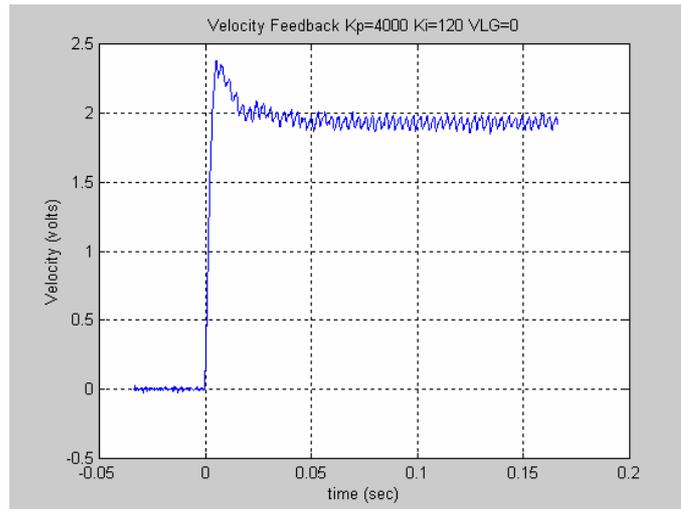


Рис. D-21. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  $K_p = 4000$ ,  $K_i = 120$ ,  $VLGN = 0$

Показанный выше отклик демонстрирует слишком большое значение *интегрального коэффициента обратной связи по скорости*; для большинства применений такая реакция считается неприемлемой.

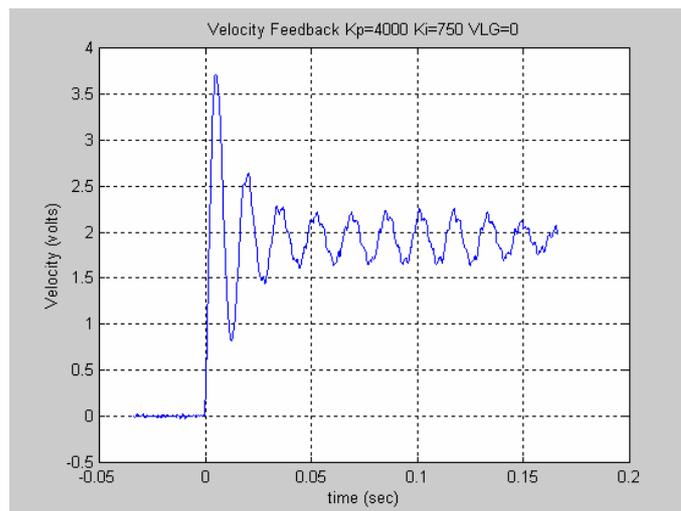


Рис. D-22. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  $K_p = 4000$ ,  $K_i = 750$ ,  $VLGN = 0$

На рисунке показана система с минимальной устойчивостью. В ее реакции присутствует не только значительное перерегулирование, но и слабо затухающий колебательный процесс. Такая реакция считается неприемлемой.

Следующим шагом процесса настройки является подсоединение к двигателю нагрузки и регулировка управления для получения требуемой реакции. Параметр *Velocity Loop Gain* (Коэффициент обратной связи по скорости) позволяет пользователю выполнить регулировку параметров контроллера с учетом нагрузки двигателя. Так же как в процедуре выше начинать следует с нулевого значения *Velocity Loop Gain* (Коэффициент обратной связи по скорости).

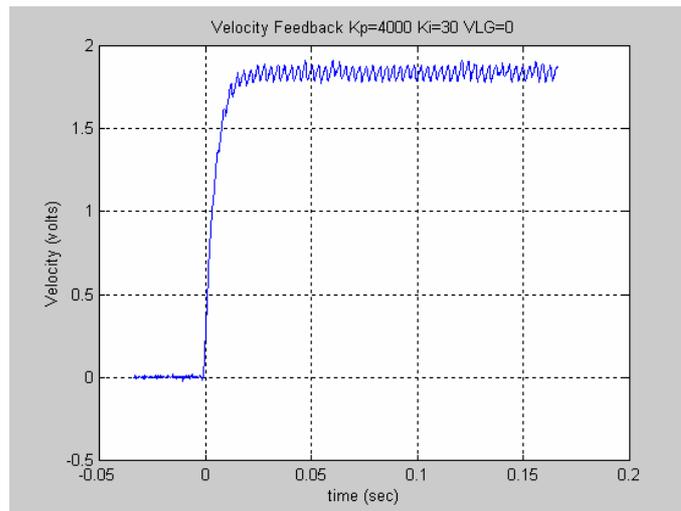


Рис. D-23. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 4000$ ,  $K_i = 30$ ,  $VLGN = 0$

На рисунке выше показана реакция скорости двигателя с подсоединенной нагрузкой при параметрах системы, установленных из приведенного ранее примера. Отклик системы является приемлемым, однако увеличение *управляющий коэффициент скорости* может уменьшить время нарастания.

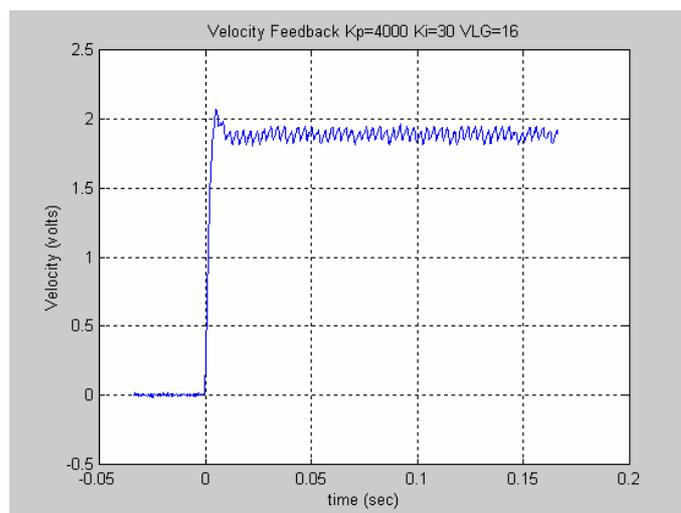


Рис. D-24. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 4000$ ,  $K_i = 30$ ,  $VLGN = 16$

Показанный на рисунке выше отклик является приемлемым. Он имеет небольшое перерегулирование, но сколько-нибудь продолжительные колебания отсутствуют.

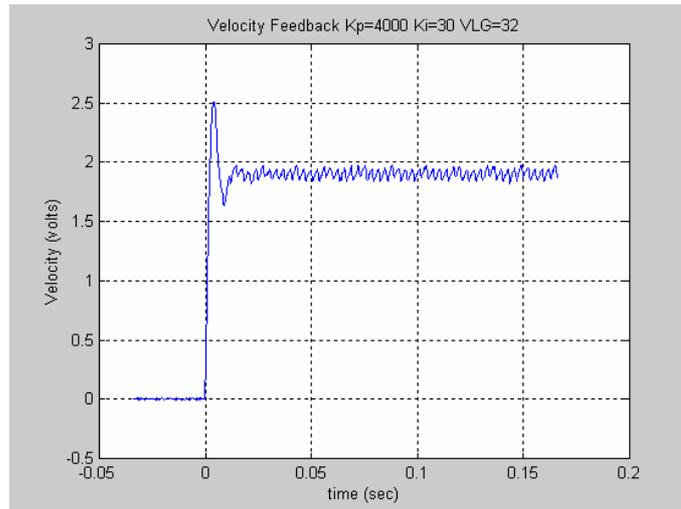


Рис. D-25. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 4000$ ,  $K_i = 30$ ,  $VLGN = 32$

Показанный здесь отклик имеет довольно большое перерегулирование, однако другие неблагоприятные эффекты, кроме начального перерегулирования и одного колебания, отсутствуют. Это перерегулирование показывает, что для пользователя может оказаться желательным снизить *управляющий коэффициент скорости*.

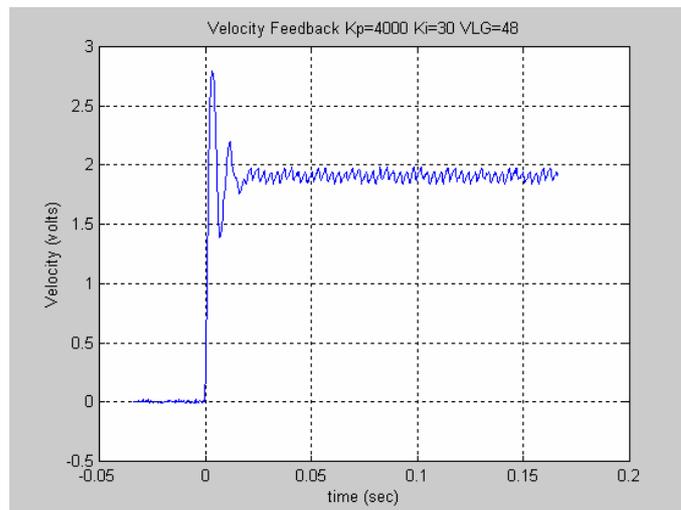


Рис. D-26. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  
 $K_p = 4000$ ,  $K_i = 30$ ,  $VLGN = 48$

Показанная на рисунке реакция имеет пере регулирование и заметный колебательный процесс. Такая реакция показывает, что коэффициент обратной связи по скорости превышает требуемое значение.

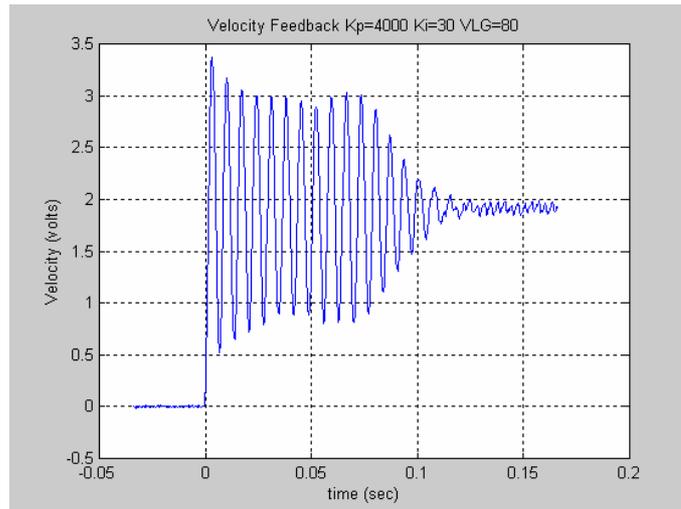


Рис. D-27. Реакция при ступенчатом изменении скорости, зависимость скорости от времени.  $K_p = 4000$ ,  $K_i = 30$ ,  $VLGN = 80$

На рисунке показана система с минимальной устойчивостью. *Управляющий коэффициент скорости* имеет слишком большое значение. Реакция демонстрирует значительное пере регулирование и продолжительный колебательный процесс. Такая реакция считается неприемлемой.

## Рекомендации по поиску и устранению неисправностей системы (аналоговый режим)

1. Модуль DSM314 требует наличия микропрограммы ПЛК версии 10.0 или более поздней и одного из следующих программных пакетов конфигурации / программирования:
  - Пакет SIMPLICITY Machine Edition Logic Developer PLC, версия 2.1 или более поздняя.
  - Пакет VersaPro, версия 1.1 или более поздняя.
2. Функция режима момента вращения модуля DSM требует наличия микропрограммы DSM версии 3.0 или более поздней.
3. Если в конфигурации модуля включен вход Drive Ready (Готов к Запуску), то этот вход должен быть подсоединен к 0 В в течение 1 секунды после включения реле Drive Enable (Привод включить); в противном случае модуль Motion Mate DSM314 работать не будет. Неправильная конфигурация Drive Ready (Готов к Запуску) или неправильное подсоединение приведет к выдаче кода ошибки C0h в %AI-данные *Axis Error Code (Код ошибки оси)*.
4. Управляющий бит %Q *ENABLE DRIVE (ВКЛЮЧИТЬ ПРИВОД)* должен быть постоянно в состоянии 1; в противном случае может быть выполнено только движение толчка. Если не возникают ошибки STOP (см. в Приложении А коды ошибок), то %I-бит статуса DRIVE ENABLED (ПРИВОД ВКЛЮЧЕН) будет зеркально отражать статус %Q-бита ENABLE DRIVE (ВКЛЮЧИТЬ ПРИВОД). Ошибка STOP сбрасывает выходной бит DRIVE ENABLED (ПРИВОД ВКЛЮЧЕН), несмотря на то что бит ENABLE DRIVE (ВКЛЮЧИТЬ ПРИВОД) остается в состоянии 1. Необходимо устранить состояние ошибки и установить управляющий

%Q-бит CLEAR ERROR (СБРОСИТЬ ОШИБКУ) на один цикл ПЛК, чтобы снова включить привод.

5. Если %I-бит статуса ERROR (ОШИБКА) находится в состоянии **1**, а %I-биты статуса AXIS ENABLED (ОСЬ ВКЛЮЧЕНА) и DRIVE ENABLED (ПРИВОД ВКЛЮЧЕН) – в состоянии **0**, то это значит, что возникла ошибка STOP (СТОП) (светодиодный индикатор статуса быстро мигает). В этом состоянии модуль DSM314 не реагирует ни на какие команды, кроме управляющего %Q-бита CLEAR ERROR (СБРОСИТЬ ОШИБКУ).
6. Управляющий %Q-бит CLEAR ERROR (СБРОСИТЬ ОШИБКУ) является битом одноразового действия. При каждом возникновении ошибки этот бит должен быть установлен в **0**, а затем в **1**, по крайней мере, в течение одного цикла ПЛК, чтобы сбросить ошибку.
7. Индикатор CFG ОК должен быть в состоянии ON (ВКЛ.); в противном случае модуль DSM314 не будет реагировать на команды ПЛК. Если этот индикатор находится в состоянии OFF (ВЫКЛ.), то это значит, что от ПЛК не получена правильная конфигурация модуля DSM314 или что была обнаружена ошибка конфигурации. Проверьте слова кода ошибки %AI на ошибки Dxxx, которые приведены в разделе "Коды системных ошибок" в Приложении А. Также проверьте таблицы сбоя ПЛК на предмет ошибок конфигурации, сообщения о которых были выданы.
8. Схема ПЛК не должна направлять следующие битовые %Q-команды на модуль DSM314 на первом цикле ПЛК: *Find Home* (Начальное позиционирование), *Execute Motion Program* (Выполнить программу движения), *Execute Local Logic* (Выполнить программу Local Logic). Если эти команды передаются на первом цикле ПЛК, то выдается сообщение об ошибке, и действие не выполняется.
9. Схема ПЛК не должна направлять следующие %AQ-команды на модуль DSM314 на первом цикле ПЛК: *Move at Velocity* (Движение на скорости), *Move Command* (Команда движения). Если эти команды передаются на первом цикле ПЛК, то выдается сообщение об ошибке, и действие не выполняется.

Настоящее приложение содержит информацию, необходимую для определения длительности выполнения локальной логической программы.

## Информация о длительности выполнения программ Local Logic

Программа Local Logic в модуле DSM ограничена длительностью выполнения **300 микросекунд**. Превышение предела длительности выполнения приводит к срабатыванию контроля времени и выдаче сообщения об ошибке. Ошибка контроля времени влечет прекращение движения оси и выполнения программы Local Logic. Длительности, приведенные в таблицах далее, позволяют программисту рассчитать длительность выполнения программы в наиболее неблагоприятном случае. Обратите внимание, что данные, приведенные ниже, представляют собой время **выполнения**, а не время отклика (реакции). Например, время выполнения, требуемое для записи значения в переменные отношения слежения, составляет 0.30 микросекунды; однако время, требуемое для проявления результирующего изменения движения оси, будет от 2 до 5 миллисекунд. Аналогично для цифровых входов при расчете времени отклика следует учитывать задержки в аппаратных фильтрах. **Примечание:** Если длительность выполнения программы лежит в пределах от 300 до 350 микросекунд, то контроль времени может не сработать; это зависит от задачи, загруженной в модуль. Пользователь должен удерживать длительность выполнения программы в пределах 300 микросекунд, чтобы обеспечить выполнение программы без каких-либо остановок из-за контроля времени.

**Приведенные далее таблицы могут быть использованы для расчета длительности выполнения программы в самом неблагоприятном случае; таким образом, можно предварительно обеспечить, что не произойдет останов программы из-за контроля времени. Примеры ниже показывают расчет длительности выполнения программы.**

## Пример 1

```

P001 := P002 + 3500; (* командная строка 1 *)
IF P001 > 5000 THEN (* командная строка 2 *)
    Torque_Limit_1 := 75; (* командная строка 3 *)
    Jog_Plus_1 := Strobe1_Level_1; (* командная строка 4 *)
END_IF; (* командная строка 5 *)

```

**Длительность выполнения для командной строки 1 =>**

(длительность загрузки P002) + (длительность загрузки константы) + (длительность выполнения операции сложения) + (длительность записи P001)

=> 0.60 (из таблицы E-7) + 0.50 (из таблицы E-7) + 0.90 (из таблицы E-1) + 0.60 (из таблицы E-7)

=> **2.60 микросекунды**

**Длительность выполнения для командной строки 2 =>**

(длительность загрузки P001) + (длительность загрузки константы) + (длительность выполнения > (условного перехода))

=> 0.60 (из таблицы E-7) + 0.50 (из таблицы E-7) + 2.50 (из таблицы E-2)

=> **3.60 микросекунды**

**Длительность выполнения для командной строки 3** (в предположении, что операция условного перехода выдала логическую единицу ("истина")) =>

(длительность загрузки константы) + (длительность записи Torque\_Limit\_1)

=> 0.50 (из таблицы E-7) + 0.30 (из таблицы E-3)

=> **0.80 микросекунды**

**Длительность выполнения для командной строки 4** (в предположении, что операция условного перехода выдала логическую единицу ("истина")) =>

(длительность загрузки Strobe1\_Level\_1) + (длительность записи Jog\_Plus\_1)

=> 1.40 (из таблицы E-3) + 1.70 (из таблицы E-3)

=> **3.10 микросекунды**

**Длительность выполнения для командной строки 5 => 0.0 микросекунды**  
(из таблицы E-2)

**Полная длительность выполнения => 2.60 + 3.60 + 0.80 + 3.10 + 0.0 = 10.10 микросекунды**

## Пример 2

```
D00 := P100 * 1000;          (* командная строка 1 *)
P101 := D00 / 55;           (* командная строка 2 *)
  Enable_Follower_1 := CTL01 BWAND CTL02;  (* командная строка 3 *)
    Follower_Ratio_A_1 := P101;          (* командная строка 4 *)
```

### Длительность выполнения для командной строки 1 =>

(длительность загрузки P100) + (длительность загрузки константы) + (длительность выполнения операции умножения) + (длительность записи D00)

=> 0.60 (из таблицы E-7) + 0.50 (из таблицы E-7) + 1.30 (из таблицы E-1) + 0.70 (из таблицы E-7)

=> **3.10 микросекунды**

### Длительность выполнения для командной строки 2 =>

(длительность загрузки D00) + (длительность загрузки константы) + (длительность выполнения операции деления) + (длительность записи P101)

=> 0.70 (из таблицы E-7) + 0.50 (из таблицы E-7) + 2.90 (из таблицы E-1) + 0.60 (из таблицы E-7)

=> **4.70 микросекунды**

### Длительность выполнения для командной строки 3 =>

(длительность загрузки CTL01) + (длительность загрузки CTL02) + (длительность выполнения операции BWAND) + (длительность операции сохранения Enable\_Follower\_1)

=> 1.40 (из таблицы E-7) + 1.40 (из таблицы E-7) + 0.20 (из таблицы E-1) + 1.70 (из таблицы E-3)

=> **4.70 микросекунды**

### Длительность выполнения для командной строки 4 =>

(длительность загрузки P101) + (длительность записи Follower\_Ratio\_A\_1)

=> 0.60 (из таблицы E-7) + 0.30 (из таблицы E-3)

=> **0.90 микросекунды**

**Полная длительность выполнения** => 3.10 + 4.70 + 4.70 + 0.90 = **13.40 микросекунды**

Таблица E-1. Длительности выполнения математических и логических операций в программах Local Logic

Математические и логические операции в программах Local Logic (назначение, :=)	Длительность выполнения в программе Local Logic (в микросекундах)
Сложение (+)	0.90**
Вычитание (-)	0.90**
Умножение (*)	1.30
Деление (/)	2.90
Получение модуля (MOD)	2.90
Получение абсолютного значения (ABS)	1.70**
BWAND	0.20
BWOR	0.30
BWXOR	0.20
BWNOT	0.50

**\*\* Примечание:** Длительности выполнения для операций сложения, вычитания и получения абсолютного значения (ABS) предполагают отсутствие переполнения при расчете.

Таблица E-2. Длительности выполнения операций условного перехода в программах Local Logic

Операции условного перехода в программах Local Logic (IF...THEN)	Длительность выполнения в программе Local Logic (в микросекундах)
Больше чем (>)	2.50
Меньше чем (<)	2.50
Больше или равно (>=)	2.50
Меньше или равно (<=)	2.50
Равно (=)	2.30
Не равно (<>)	2.30
BWAND	1.40
BWOR	1.40
BWXOR	1.40
BWNOT	1.60
Нуль-оператор ( <i>IF переменная THEN</i> )	1.10
END IF	0.00

**Примечание:** Длительность выполнения условных переходов дана для случая, когда результатом операции IF...THEN является FALSE (ЛОЖЬ). Это дает более неблагоприятный случай выполнения, поскольку длительность операции условного перехода с TRUE (ИСТИНА) на выходе меньше. Обратите внимание, что команда END\_IF совсем не требует времени на исполнение.

Таблица E-3. Длительности, связанные с работой с переменными Local Logic оси 1

X – не применимо.

Имя переменной Local Logic	Длительность выполнения в программе Local Logic (в микросекундах)	
	Чтение	Запись
Strobe1_Level_1	1.40	X
Strobe2_Level_1	1.40	X
Positive_EOT_1	1.40	X
Negative_EOT_1	1.40	X
Home_Switch_1	1.40	X
Digital_Output1_1	X	1.80
Digital_Output3_1	X	1.80
Analog_Input1_1	0.80	X
Analog_Input2_1	0.80	X
Position_Loop_TC_1	X	0.30
Follower_Ratio_A_1	X	0.30
Follower_Ratio_B_1	X	0.30
Torque_Limit_1	X	0.30
Position_Increment_Cts_1	X	0.30
Velocity_Loop_Gain_1	0.80	0.20
Reset_Strobe1_1	X	1.70
Reset_Strobe2_1	X	1.70
Enable_Follower_1	X	1.70
Jog_Plus_1	X	1.70
Jog_Minus_1	X	1.70
FeedHold_1	X	1.70
Error_Code_1	0.80	X
Actual_Position_1	0.70	X
Strobe1_Position_1	0.80	X
Strobe2_Position_1	0.80	X
Actual_Velocity_1	0.80	X
Block_1	0.90	X
Commanded_Position_1	0.60	X
Position_Error_1	0.60	X
Commanded_Velocity_1	0.60	X
User_Selected_Data1_1	0.60	X
User_Selected_Data2_1	0.60	X
UnAdjusted_Actual_Position_Cts_1	0.80	X
UnAdjusted_Strobe1_Position_Cts_1	0.80	X
UnAdjusted_Strobe2_Position_Cts_1	0.80	X
Commanded_Torque_1	0.80	X
Axis_OK_1	1.40	X
Position_Valid_1	1.40	X
Strobe1_Flag_1	1.40	X
Strobe2_Flag_1	1.40	X
Drive_Enabled_1	1.40	X
Program_Active_1	1.40	X
Moving_1	1.40	X
In_Zone_1	1.40	X
Position_Error_Limit_1	1.40	X
Torque_Limited_1	1.40	X
Servo_Ready_1	1.40	X
Follower_Enabled_1	1.40	X
Follower_Ramp_Active_1	1.40	X
Follower_Velocity_Limit_1	1.40	X

**Таблица Е-4. Длительности, связанные с работой с переменными Local Logic оси 2**

X – не применимо.

Имя переменной Local Logic	Длительность выполнения в программе Local Logic (в микросекундах)	
	Чтение	Запись
Strobel_Level_2	1.40	X
Strobe2_Level_2	1.40	X
Positive_EOT_2	1.40	X
Negative_EOT_2	1.40	X
Home_Switch_2	1.40	X
Digital_Output1_2	X	1.80
Digital_Output3_2	X	1.80
Analog_Input1_2	0.80	X
Analog_Input2_2	0.80	X
Position_Loop_TC_2	X	0.30
Follower_Ratio_A_2	X	0.30
Follower_Ratio_B_2	X	0.30
Torque_Limit_2	X	0.30
Position_Increment_Cts_2	X	0.30
Velocity_Loop_Gain_2	0.80	0.20
Reset_Strobel_2	X	1.70
Reset_Strobe2_2	X	1.70
Enable_Follower_2	X	1.70
Jog_Plus_2	X	1.70
Jog_Minus_2	X	1.70
FeedHold_2	X	1.70
Error_Code_2	0.80	X
Actual_Position_2	0.70	X
Strobel_Position_2	0.80	X
Strobe2_Position_2	0.80	X
Actual_Velocity_2	0.80	X
Block_2	0.90	X
Commanded_Position_2	0.60	X
Position_Error_2	0.60	X
Commanded_Velocity_2	0.60	X
User_Selected_Data1_2	0.60	X
User_Selected_Data2_2	0.60	X
UnAdjusted_Actual_Position_Cts_2	0.80	X
UnAdjusted_Strobel_Position_Cts_2	0.80	X
UnAdjusted_Strobe2_Position_Cts_2	0.80	X
Commanded_Torque_2	0.80	X
Axis_OK_2	1.40	X
Position_Valid_2	1.40	X
Strobel_Flag_2	1.40	X
Strobe2_Flag_2	1.40	X
Drive_Enabled_2	1.40	X
Program_Active_2	1.40	X
Moving_2	1.40	X
In_Zone_2	1.40	X
Position_Error_Limit_2	1.40	X
Torque_Limited_2	1.40	X
Servo_Ready_2	1.40	X
Follower_Enabled_2	1.40	X
Follower_Ramp_Active_2	1.40	X
Follower_Velocity_Limit_2	1.40	X

Таблица Е-5. Длительности, связанные с работой с переменными Local Logic оси 3

X – не применимо.

Имя переменной Local Logic	Длительность выполнения в программе Local Logic (в микросекундах)	
	Чтение	Запись
Strobe1_Level_3	1.40	X
Strobe2_Level_3	1.40	X
Positive_EOT_3	1.40	X
Negative_EOT_3	1.40	X
Home_Switch_3	1.40	X
Digital_Output1_3	X	1.80
Digital_Output3_3	X	1.80
Analog_Input1_3	0.80	X
Analog_Input2_3	0.80	X
Reset_Strobe1_3	X	1.70
Reset_Strobe2_3	X	1.70
Error_Code_3	0.80	X
Actual_Position_3	0.70	X
Strobe1_Position_3	0.80	X
Strobe2_Position_3	0.80	X
Actual_Velocity_3	0.80	X
Axis_OK_3	1.40	X
Position_Valid_3	1.40	X
Strobe1_Flag_3	1.40	X
Strobe2_Flag_3	1.40	X

Таблица Е-6. Длительности, связанные с работой с переменными Local Logic оси 4

X – не применимо.

Имя переменной Local Logic	Длительность выполнения в программе Local Logic (в микросекундах)	
	Чтение	Запись
Strobe1_Level_4	1.40	X
Strobe2_Level_4	1.40	X
Positive_EOT_4	1.40	X
Negative_EOT_4	1.40	X
Home_Switch_4	1.40	X
Digital_Output1_4	X	1.80
Digital_Output3_4	X	1.80
Analog_Input1_4	0.80	X
Analog_Input2_4	0.80	X

Таблица Е-7. Длительности, связанные с работой с глобальными переменными Local Logic  
X – не применимо.

Имя переменной Local Logic	Длительность выполнения в программе Local Logic (в микросекундах)	
	Чтение	Запись
<i>Константы программ Local Logic</i>	0.50	X
Overflow (Переполнение)	2.40	1.30
System_Halt	X	1.80
Data_Table_Ptr	0.60	0.70
Data_Table_sint	2.10	1.70
Data_Table_usint	1.80	1.70
Data_Table_int	2.30	2.20
Data_Table_uint	2.30	2.20
Data_Table_dint	3.80	4.00
Module_Error_Present	1.40	X
New_Configuration_Received	1.40	X
First_Local_Logic_Sweep	1.40	X
Module_Status_Code	0.50	X
CTL_1_to_32	0.50	X
P000-P255	0.60	0.60
D00-D07	0.70	0.70
CTL01-CTL32	1.40	1.80

## Обновление микропрограммы в модуле DSM314

---

---

Рабочая микропрограмма модуля DSM314 хранится во встроенной флэш-памяти. Обновление микропрограммы предоставляется на дискете. Утилита загрузки "PC Loader" управляет загрузкой новой микропрограммы с дискеты во флэш-память модуля DSM314; утилита работает под DOS. Имеется загрузочная программа для Windows (Winloader). Для работы утилиты загрузки "PC Loader" требуется компьютер, совместимый с IBM AT/PC, с ОЗУ не менее 640 Кб, один дисковод, операционная система MS-DOS® 3.3 (или выше) и один последовательный порт RS-232. Чтобы утилита могла работать в окне MS-DOS под Windows® 3.1, Windows® 95 или Windows NT®, процессор должен быть, по крайней мере, Pentium™ 133. Если это не выполняется, то следует перезагрузить компьютер в режиме MS-DOS. Утилита PC Loader оптимально работает с жестким диском, на котором имеется не менее 1 МБ свободного пространства. Утилита обновления Winloader требует для своей работы Windows 95, Windows NT или Windows 98. Аппаратное обеспечение, требующееся для работы этих операционных систем, является достаточным также для работы Winloader. Утилита Winloader требует около 500 КБ пространства на жестком диске.

### Предупреждение

**Прежде чем перейти к режиму загрузки, пользователь обязан убедиться, что PC подключен к модулю DSM (а не к ЦП ПЛК или к другому модулю, который поддерживает обновления микропрограмм во флэш-памяти). Несоблюдение этого может вызвать потерю программы ЦП ПЛК и конфигурации.**

### Чтобы установить новую микропрограмму, выполните следующие действия:

1. Перед началом действий по обновлению сохраните или зарезервируйте все программы и данные, находящиеся в модуле.
2. Переведите ПЛК в режим работы STOP/NOIO. (Устраните все сбои и неисправности).
3. Проверьте, что выставлена скорость 19200 бод на последовательном SNP порту модуля.
4. Используя кабель "Station Manager to PC", IC693CBL316, подсоедините соответствующий последовательный порт вашего компьютера (ведущий) с модулем DSM302, на котором должна быть обновлена микропрограмма, (ведомый).

### Обновление в DOS

**Примечание:** Этот раздел действителен только для программы DOS Loader (загрузчик), запущенной из DOS. Для программы, использующей Windows, см. следующий раздел "Обновление в Windows"

5. (DOS) Вставьте дискету с меткой в дисковод A: или B:. Убедитесь, что защита записи на дискете снята. Запустите самораспаковывающийся архив, задав

- привод A: или B: как место назначения в ответ на сообщение "Unzip to folder" ("Распаковать в папку").
6. (DOS) На подсказку C:\> введите A:install (или B:install, если дисководом является B:). Программа установки скопирует несколько файлов на жесткий диск, затем запустит программу PC Loader Установка может быть проведена также непосредственно с дисковода, если на жестком диске недостаточно места. Для этого введите install в ответ на подсказку A:\> или B:\>.
  7. (DOS) В главном меню нажмите клавишу F3, чтобы сконфигурировать правильный последовательный порт, если кабель не подключен к COM1. Нажимайте клавишу TAB, чтобы переключаться между опциями, и клавишу ENTER, чтобы выполнять выбор.
  8. (DOS) В главном меню нажмите клавишу F1, чтобы прикрепить модуль DSM302 как ведомое устройство.
  9. (DOS) Когда ведомое устройство назначено, то появится меню режима загрузки; нажмите F1, чтобы ввести BOOT MODE (РЕЖИМ ЗАГРУЗКИ), и нажмите клавишу "Y", чтобы подтвердить эту операцию. Индикаторы STAT и CFG на передней панели модуля начинают мигать синхронно.
  10. (DOS) Находясь в режиме загрузки, нажмите клавишу F1, чтобы загрузить новую микропрограмму.
  11. (DOS) нажмите клавишу Y, чтобы подтвердить эту операцию. Загрузка занимает около 4 минут. Если загрузка прерывается, см. далее раздел "Повторный запуск прерванного обновления микропрограммы".
  12. (DOS) После завершения загрузки программа PC loader сообщает о необходимости выключить и включить модуль. Выполните этот цикл с модулем. Если модуль установлен на расширительном или удаленном крейте, то необходимо выполнить операцию выключения-включения с основным крейтом также.
  13. (DOS) Поместите на узел метку с указанием установленной версии микропрограммы. Если эта программа является бета-версией или наладочной, укажите это на метке.

### **Обновление под Windows (для Windows 95/NT/98, НЕ ПРЕДНАЗНАЧЕНО ДЛЯ Windows 3.1)**

**Примечание:** Этот раздел предназначен только для тех, кто использует программу обновления Winloader с Windows 95, NT или 98. Если используется операционная система DOS, то см. раздел "Обновление под DOS".

5. (WIN) Вставьте дискету с меткой в дисковод A: или B:. Убедитесь, что защита записи на дискете снята. Запустите самораспаковывающийся архив, задав привод A: или B: как место назначения в ответ на сообщение "Unzip to folder" ("Распаковать в папку").
6. (WIN) Запустите программный пакет Winloader двойным щелчком по его значку на приводе A: или B: (в зависимости от обозначения привода для дискет 3.5") в Проводнике Windows или просто запустите на выполнение, войдя в стартовое меню и выбрав RUN (ВЫПОЛНИТЬ). В окне RUN введите A(или B):winloader.exe.
7. (WIN) Щелкните кнопку "Update" ("Обновить"); начинается загрузка микропрограммы.
8. (WIN) После завершения обновления окно развернется, показывая статус обновления. Если обновление прошло успешно, то выключите и включите ПЛК, а также укажите что другое устройство НЕ должно обновляться, для чего щелкните левой кнопкой "No". Если обновление не было нормально завершено, то попытайтесь изменить скорость обновления в установках с 38400 на 19200 и повторите процедуру.

## Повторный запуск прерванного обновления микропрограммы

- A. Подсоедините все кабели, как описано на шаге 4 выше.
- B. Выключите и включите кейт, в котором установлен модуль. Если была выполнена частичная или ошибочная загрузка, то на включившемся модуле индикаторы STAT и CFG будут мигать синхронно.
- C. Если на вашем компьютере продолжает работать программа PC Loader или Winloader, перейдите к шагу D, указанному ниже; в противном случае выполните шаги 5 и 6, указанные выше.
- D. Выполните шаг 7, указанный выше. Обратите внимание, что происходит автоматический вход в BOOT MODE (РЕЖИМ ЗАГРУЗКИ).
- E. Выполните шаги 9 – 12, указанные выше.
- F. Если обновление все же не удалось, что выполните процесс с меньшим значением скорости в бодах.
- G. Поместите на узел метку с указанием установленной версии микропрограммы.

*MS-MS-DOS, Windows и Windows NT являются зарегистрированными торговыми знаками корпорации Майкрософт; Pentium является зарегистрированной торговой маркой корпорации Intel; IBM-AT и IBM-PC являются зарегистрированными торговыми знаками корпорации International Business Machines.*



В общем случае точность значения позиции строба может быть охарактеризована как  $\pm 2$  импульса (отсчета) с дополнительным отклонением 10 микросекунд. Однако фактическая точность значения позиции строба может быть лучше в зависимости от конфигурации оси, ускорения двигателя при стробировании и от числа импульсов на оборот используемого датчика положения. Прежде всего, имеет значение, является ли конфигурация оси цифровой или аналоговой.

## Аналоговый режим

В аналоговом режиме при стробировании значение импульсного счетчика сразу же захватывается в регистр временного хранения. Это значит, что погрешность захвата позиции определяется, в первую очередь, входной фильтрацией и задержкой дискретизации для входа строба, что может составлять в общей сложности до 10 микросекунд (или то количество импульсов, которое может быть выдано за 10 микросекунд). Обратите внимание, что это значение может быть на один импульс меньше, в зависимости от того, когда был выдан строб по отношению к тому, когда изменилось значение импульса.

## Цифровой режим

В цифровом режиме датчик положения считывается в виде последовательности данных. Поскольку эти данные запрашиваются один раз в 250 микросекунд, то фиксирование считывания значения позиции от датчика положения может быть осуществлено только с точностью 250 микросекунд. Чтобы преодолеть это ограничение, стробирование привязывается по времени к последнему считыванию позиции датчиком положения; это осуществляется в самом модуле DSM314. Это значение используется для того, чтобы оценить позицию оси на тот момент, когда был выдан строб, основываясь на текущей скорости сервооси на момент строба. Скорость, используемая для расчета, определяется по разности двух отсчетов позиции датчика положения вокруг строба (см. формулу ниже).

$$\text{Скорость} = \frac{(\text{Отсчет позиции после строба}) - (\text{Отсчет позиции перед стробом})}{250 \text{ микросекунд}}$$

Таким образом, изменение скорости (например, ускорение или торможение двигателя) между отсчетами позиции здесь не учитывается, что вносит погрешность в значение захваченной стробом позиции. Для стробирования, осуществляемого при постоянной скорости за период считывания, алгоритм интерполяции дает точность до одного импульса; погрешность определения позиции здесь определяется, прежде всего, фильтрацией и задержками дискретизации.

Приведенный ниже пример может быть использован для расчета погрешностей в наиболее неблагоприятном случае при ускорении на конкретном серводвигателе.

В примере даны следующие значения и константы:

Разрешение датчика положения = 8192 импульсов на оборот

$A$  = Ускорение / замедление во время стробирования составляет 250 000 000 имп./с<sup>2</sup> (принимается постоянным на всем интервале 250 мс; использование большего значения ускорения будет увеличивать значение ошибки в расчете).

$T_p$  = Период дискретизации позиции, составляющий 250 микросекунд

$V_1$  = Начальная скорость непосредственно перед стробированием, которая принимается равной 0 в данном примере.

Изменение количества импульсов датчика положения (Cnts) для данного промежутка времени (t) может быть рассчитано с использованием следующей формулы:

$$P_{act} = V_1 t + \frac{1}{2} A t^2$$

Следовательно, полное количество импульсов, выдаваемых за период дискретизации в данном примере, составляет, примерно, 8 (фактическое рассчитанное значение равно 7.8125) или 0.343 градуса.

Средняя скорость за период выборки, определяющая изменение позиции, рассчитывается следующим образом:

$$V_{avg} = \frac{\text{Изменение числа импульсов}}{\text{Интервал выборки}} = \frac{7.8125 \text{ имп.}}{250 \text{ мкс}} = 31250 \text{ имп./с}$$

Для оценки позиции строба с использованием полученной выше скорости может быть применена следующая формула:

$$P_{est} = V_1 t + V_{avg} t$$

Таким образом, ошибка между расчетной позицией строба и фактической его позицией определяется как:

$$\text{Ошибка} = P_{act} - P_{est}$$

Графики ниже показывают изменение фактической и расчетной позиции строба, а также результирующей ошибки отсчета строба на периоде выборки 250 микросекунд. График показывает, что наибольшее значение ошибки имеет место в середине периода (т.е. при 125 мкс).

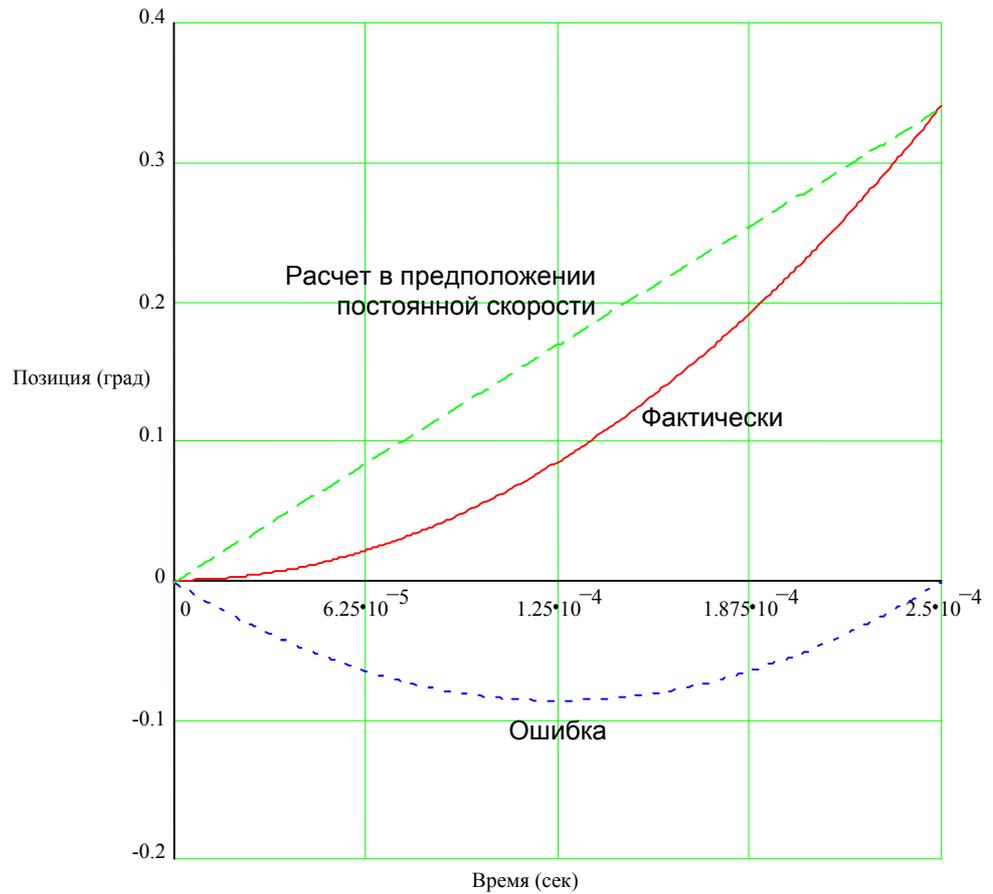


Рис. G-1: Пример ошибки определения позиции при ускорении

Поскольку начальная скорость здесь равна 0, то формулу для расчета  $P_{act}$  можно преобразовать так, чтобы определить время, когда импульс фактически имел место ( $T_{act}$ ):

$$T_{act} = \sqrt{\frac{2P_{act}}{A}}$$

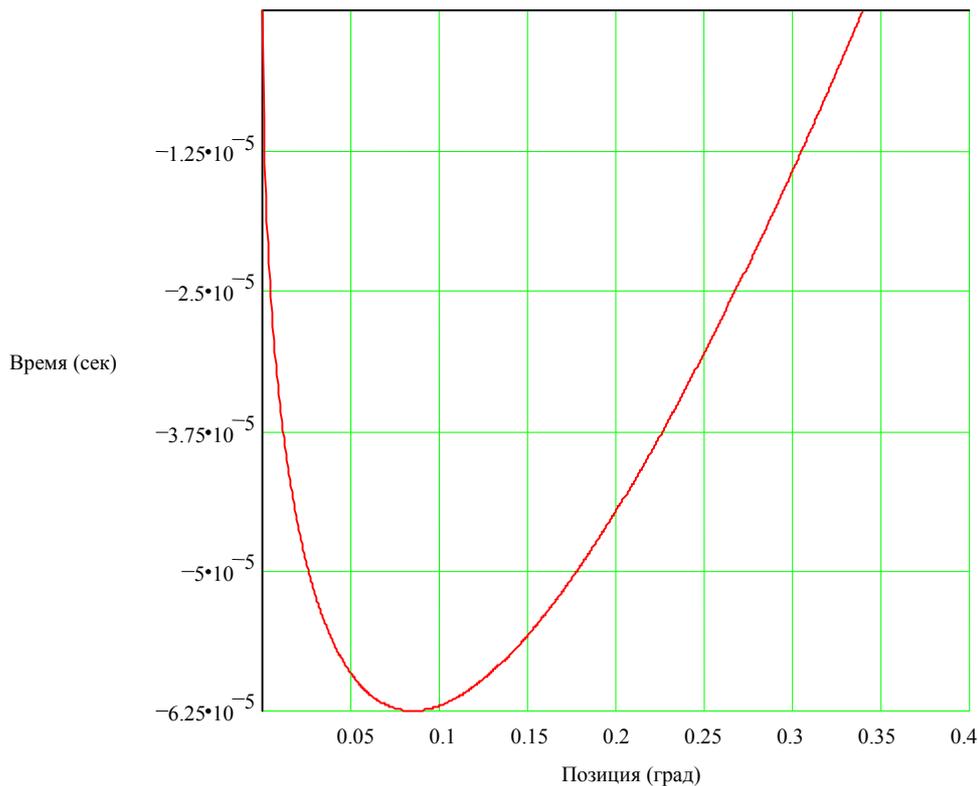
Аналогично формула для расчета позиции строба ( $P_{est}$ ) может быть преобразована относительно времени ( $T_{est}$ ) (предполагается, что начальная скорость равна 0):

$$T_{est} = P_{est} / V_{avg}$$

Используя эти формулы, можно рассчитать разницу по времени между моментом появления строба и моментом выдаваемого импульса (т.е. эффективную задержку):

$$\text{Эффективная задержка} = T_{est} - T_{act}$$

Эффективная задержка при максимальном значении ошибки позиции строба (т.е. при 125 микросекундах) составляет – 62.5 микросекунды. Это значение является отрицательным, поскольку расчетная / сообщаемая позиция строба появляется до фактической позиции при стробировании. Приведенный далее рисунок показывает эту эффективную задержку в зависимости от изменения позиции на интервале выборки в данном примере.



**Рис. G-2: Эффективная задержка времени отклика**

Таким образом, в вышеприведенном примере наибольшая ошибка, возникающая из-за ускорения/замедления, может быть выражена как  $\pm 0.086$  градуса (примерно, 2 импульса) позиции или как 62.5 микросекунды задержки (при условии нулевой начальной скорости). **Учтите, что DSM не работает с дробными числами; поэтому ошибка будет округлена до ближайшего значения в импульсах или пользовательских единицах.**

Формула для определения ошибки строба из-за ускорения/замедления цифровой оси:

Импульсы\_ошибки =

$$\text{Эффективная\_задержка} = \frac{T_p(V_1 + A T_p)}{4(2 V_1 + A T_p)}$$

Где:

- A = Ускорение / замедление во время стробирования
- $T_p$  = Период дискретизации позиции, составляющий 250 микросекунд
- $V_1$  = Скорость непосредственно перед поступлением строба

Обратите внимание, что указанные выше формулы предполагают постоянное значение ускорения на интервале выборки. Формулы для определения величины ошибки для случаев, когда ускорение изменяется на интервале выборки, являются слишком сложными для данного руководства.

Обратите внимание, что может иметь место также дополнительная ошибка порядка 10 микросекунд (или соответствующее количество градусов или импульсов позиции, вмещающиеся в эти 10 микросекунд) в результате входного фильтрования / задержкам дискретизации в аппаратуре.

**Предупреждение**

**Учтите, что подключение и тип устройства, используемого для входа строба, могут также внести погрешность в значение строба.**



## #

### α Series

двигатели, 1-12, 1-14  
сервоприводы, 1-11, 1-13  
таблица двигателей, 1-14  
усилители, 1-11, 1-13

### α Series C12

двигатель, 1-14

## %

### %AI слова статуса

Заданная позиция, 5-9  
Заданная скорость, 5-10  
номер блока команды, 5-9  
позиция строба 1, 2, 5-9  
Текущая позиция, 5-9  
Текущая скорость, 5-10

### %AI слово статуса

Данные, выбранные пользователем, 5-10  
код статуса модуля, 5-9  
Ось 1, 2, Код ошибки ведущего устройства  
слежения, 5-9

### %AQ команда немедленного действия

Загрузить параметр немедленно, 5-34

### %AQ команды немедленного действия

Включить цифровой / аналоговый выход,  
5-22, 5-28  
Выбрать возвращаемые данные, 5-28  
Движение, 5-25  
Движение на скорости, 5-21  
Длительность участка линейного  
компенсирующего изменения скорости  
слеящего устройства, 5-30  
Зона прихода в позицию, 5-24  
Интегральный коэффициент обратной  
связи по скорости, 5-31  
Коэффициент обратной связи по скорости,  
5-27  
Ноль, 5-20  
Подстановка скорости, 5-21  
Постоянная времени контура  
позиционирования, 5-26  
Приращение позиции, 5-21  
Приращение позиции с обновлением  
позиции, 5-24  
Пропорциональный коэффициент  
обратной связи по скорости, 5-31  
Скорость Толчка, 5-26  
Упреждение по скорости, 5-26  
Ускорение Толчка, 5-26  
Установить бит Configuration Complete  
(Конфигурация закончена), 5-34  
Установить позицию, 5-21  
Фильтр уставок момента вращения, 5-31

### %AQ Команды немедленного действия

выбор Return Data (Возвращаемые Данные)  
%AQ, 4-21

### %AQ команды прямого действия

Включить цифровой / аналоговый выход,  
6-7

### %I бит статуса

Axis Enabled (Ось включена), 5-4  
Drive Enabled (Привод включен), 5-5  
Follower Enabled (Слежение включено), 5-6  
In Zone (В зоне), 5-5  
Position Error Limit (Предел Ошибки по  
Положению), 5-6  
Position Valid (Позиция достигнута), 5-5  
имеется ошибка модуля, 5-3  
ПЛК-управление действует, 5-3  
получена новая конфигурация, 5-3  
предел по моменту вращения, 5-6  
привод в норме (вход 4), 5-6  
программа активна, 5-5

### %I биты статуса

позиция строба 1, 2, 5-6

### %Q бит выхода

Выход 5 В, 5-15

### %Q дискретная команда

Включить привод / MCON, 5-13  
Включить слежение, 5-16  
Выполнить программу Движения 0 - 10, 5-  
12  
Минус Толчка, 5-14  
Остановить подачу (включение команды),  
5-13  
Остановить подачу (выключение  
команды), 5-13  
Плюс Толчка, 5-14  
Поиск Исходной позиции, 5-14  
Сброс флагов стробов 1, 2, 5-14  
Сбросить ошибку, 5-12

### %Q дискретная команд

Выбрать ведущий внутренний источник  
слеящего устройства, 5-16

### %Q дискретные команды

CTL09 - CTL12 управление выходами, 5-13  
Прекратить все движения, 5-13

## 1

1-й порядок с 1-ым порядком, 16-8  
1-й порядок с 2-ым порядком, 16-8

## 2

2-й порядок с 1-ым порядком, 16-8  
2-й порядок с 2-ым порядком, 16-8  
2-й порядок с 3-им порядком, 16-8

## 3

3-й порядок с 2-ым порядком, 16-8  
3-й порядок с 3-им порядком, 16-9

## 6

64-битовые регистры двойной точности,  
13-3

## A

A:В отношение, 8-3

ACCEL команда  
программирование движения, 7-9

## C

CFG LED, A-20  
SIMPLICITY Machine Edition -  
Конфигурация, 2-31  
SMOVE, 7-23  
SMOVE команда  
программирование движения, 7-11  
COMM\_REQ  
обнаружение ошибок и работа с  
ними, B-4  
FT-выход (неисправность), B-6  
UDT, B-8  
коды слова статуса, B-3  
командный блок, B-14  
мониторинг слова статуса, B-4  
пример, B-17  
пример проверки, B-22  
проверка, B-4  
сравнение с командой немедленного  
действия Load Param. (Загрузить  
параметр), B-1  
таблица кодов памяти, B-16  
COMM\_REQ пример  
UDT, B-11  
COMM\_REQ таблица памяти  
тип Загрузкопараметров, B-16  
COMMREQ Ladder команда, B-2, B-6  
COMMREQ командный блок, B-2, B-14  
CTL09 - CTL12 управление выходами %Q  
дискретная команда, 5-13

## D

DSM302 модуль  
установка, 3-5  
DSM302-модули на систему, 3-5  
DWELL команда  
программирование движения, 7-11  
Dwell, команда, 7-27

## E

EN1 LED, A-21  
EN2 LED, A-21  
EN3 LED, A-21  
EN4 LED, A-21  
ENDPROG команда  
программирование движения, 7-12  
ENDSUB команда  
программирование движения, 7-12

## F

Find Home Vel (Конечная Скорость  
Прихода в Исходную позицию), 4-21

Find Home Vel (Скорость поиска  
исходной позиции), 4-21  
Follower Enabled (Слежение включено) %I  
бит статуса, 5-6  
FollowrEnInp (Триггер Включения  
Следящего Устройства), 4-23, 4-24

## H

Hirose  
20 контактов PRC разъем, 2-19  
Home Positn (Исходная Позиция), 4-20  
Home Positn (Смещение Исходной  
Позиции), 4-20

## I

IC693ACC335  
клеммная колодка, 3-10  
IC693ACC336  
вспомогательная клеммная колодка оси, 3-  
14  
IC693CBL316  
кабель для обновления микропрограммы,  
4-9  
IC800CBL001/002  
подключение кабеля, 3-32  
In Zone (В зоне) %I бит статуса, 5-5  
Intgr Mode (Режим Интегратора), 4-32  
Intgr TC (Постоянная времени  
интегратора), 4-32

## J

JF1 – Соединительный элемент, 2-9  
JS1B Разъем, 2-5, 2-13  
JUMP команда  
программирование движения, 7-12

## K

K1 Кабель к DSM314  
α Series - Подключение, 2-13

## L

LOAD команда  
программирование движения, 7-12  
Local Logic, длительности выполнения  
математических и логических  
операций в программах, E-4  
Local Logic, длительности выполнения  
операций условного перехода в  
программах, E-4  
Local Logic, информация о длительности  
выполнения программ, E-1

## М

- MaxAccUu  
расчет, 5-18
- MaxPosnUu  
расчет, 5-18
- MaxVelUu  
расчет, 5-18
- MkipTime (Длительность  
компенсации)**, 8-8
- Motion Mate DSM314  
включение питания, С-3  
пояснения, 1-1  
характеристики, 1-1
- Motion Mate DSM314 – Данные  
Конфигурации  
Модель Двигателя, 2-37
- Move+ и Move – режимы, 6-4

## Р

- PMOVE, 7-23
- PMOVE команда  
программирование движения, 7-13
- Pos Err Lim (Предел Ошибки по  
Положению), 4-28
- Pos Loop TC (Постоянная Времени  
Контура Позиционирования), 4-30
- Position Error Limit (Предел Ошибки по  
Положению) %I бит статуса, 5-6
- Position Valid (Позиция достигнута) %I  
бит статуса, 5-5
- PROGRAM команда  
программирование движения, 7-13

## S

- S-CURVE (S-кривая), переходы, 7-35, 7-  
36, 7-37
- SL-Series  
сервоприводы, 1-15
- SUBROUTINE команда  
программирование движения, 7-14
- SVU-усилитель, 1-11
- SYNC block команда  
программирование движения, 7-15
- S-кривая, уравнения изменения  
скорости**, 7-48
- S-кривая, ускорение, 7-22

## U

- UDT COMM\_REQ, В-8  
пример, В-11

## V

- Vel Lp Gain (Коэффициент Обратной  
связи по Скорости), 4-32

- VELOC команда  
программирование движения, 7-15
- VersaPro конфигурация, 2-26

## W

- WAIT команда  
программирование движения, 7-16
- Wait, команда, 7-28

## A

- Абсолютное позиционирование, 7-21
- Абсолютный датчик положения  
использование после отключения питания  
от батареи, С-2  
первое использование, С-2
- Абсолютный датчик положения, режим,  
информация, С-2  
инициализация позиции, С-2  
команда установки позиции, С-3
- АВАРИЙНЫЙ ОСТАНОВ**  
α Series - Подключение, 2-11, 2-18
- АВАРИЙНЫЙ ОСТАНОВ СТАНКА**, 2-11
- Автоматическая передача данных  
данные о выходных командах, 5-1  
данные о состоянии входов, 5-1
- Аппаратное обеспечение - конфигурация,  
2-25  
Machine Edition, 2-31
- Аппаратное обеспечение – конфигурация  
VersaPro, 2-26
- Арифметические операторы, 11-3, 12-8

## Б

- Безусловные переходы, 7-29
- Бит статуса Axis Enabled (Ось включена)  
%I, 5-4
- Бит статуса Drive Enabled (Привод  
включен) %I, 5-5
- Блок #, 7-3

## В

- В/В, технические характеристики, 3-38
- Варианты выбора конфигурации битов  
CTL01-CTL24, 14-4
- Введение в программирование  
электронного САМ, 16-21
- Ведущие  
источники, 8-2
- Взаимодействие программ движения с  
САМ, 16-10
- Включение и выключение Local Logic, 12-  
6
- Включение слежения при помощи  
внешнего входа, 8-6, 8-7

- Включить привод / MCON %Q  
дискретная команда, 5-13
- Включить слежение %Q дискретная команда, 5-16
- Включить Ц/А, команда, 6-7
- Включить цифровой / аналоговый выход %AQ команда немедленного действия, 5-22, 5-28
- Включить цифровой / аналоговый выход %AQ команда прямого действия, 6-7
- Внешний вход для включения слежения, 8-6, 8-7
- Внутренний генератор ведущей скорости, 8-3
- Вспомогательная клеммная колодка, 3-14  
вид сбоку, 3-16  
описание и монтажные размеры, 3-14  
сборочные чертежи, 3-15
- Вход Drive Ready  
параметр включено/выключено, 4-16
- Входы перебега  
пусковая проверка правильности работы, D-1
- Выбор Команды немедленного действия Return Data (Возвращаемые Данные) %AQ), 4-21
- Выбрать ведущий внутренний источник следящего устройства %Q дискретная команда, 5-16
- Выбрать возвращаемые данные %AQ команда немедленного действия, 5-28
- Вызов подпрограммы, 7-3
- Выполнение (тестирование) вашей программы движения на базе CAM, 16-44
- Выполнение вашей первой программы Local Logic, 10-25, 10-36
- Выполнить программу Движения 0 – 10 %Q дискретная команда, 5-12
- Выход 5 В %Q бит выхода, 5-15
- Выходы / команды Local Logic, 12-7
- Г**
- Глобальные переменные, 13-8, 13-9
- Глобальные переменные Local Logic.  
Длительности, связанные с работой с ними., E-8
- Граничные условия, 16-9
- Д**
- Данные конфигурации модуля DSM314, 4-3
- Данные конфигурации оси, 4-10, 4-33, 4-35  
Fnl Home Vel (Конечная Скорость Прихода в Исходную позицию), 4-21
- FollwrEnInp, 4-23, 4-24
- Intgr Mode (Режим Интегратора), 4-32
- Intgr TC (Постоянная времени интегратора), 4-32
- MkupTime (Длительность компенсации)**, 8-8
- Pos Err Lim, 4-28
- Pos Loop TC, 4-30
- Vel Lp Gain, 4-32
- Исходная Позиция, 4-20
- концевой выключатель перебега, 4-15
- отрицательный конец передвижения, 4-17
- положительный конец передвижения, 4-17
- Режим Исходной позиции, 4-21
- Скорость поиска исходной позиции, 4-21
- Смещение Исходной позиции, 4-20
- Данные Конфигурации Оси  
Скорость при 10 В, 4-31
- Данные конфигурации порта последовательной связи, 4-8
- Данные, выбранные пользователем, %AI слово статуса, 5-10
- Датчик положения 3  
ведущий вход, 8-2
- Датчик положения, инкрементный импульсный, C-4
- Двигатели  
β Series, 1-14  
α-серия, 1-12
- Движение %AQ команда немедленного действия, 5-25
- Движение на скорости %AQ команда немедленного действия, 5-21
- Движение на скорости, команда, 6-6
- Движение позиционирования, 7-3
- Движение CAM, контролируемое по времени, 16-13
- Движение слежения в комбинации с программами движения, 9-2
- Движение, непрерывное, (CMOVE), 7-23
- Движение, подпрограмма структура, 7-17
- Движение, позиционирование (PMOVE), 7-23
- Движения, программируемые, 7-25
- Деление на ноль, 12-18
- Длительность сканирования  
увеличение, 1-6
- Дополнительная информация, 6-8
- З**
- Загрузить параметр, 7-3
- Загрузить параметр немедленно %AQ команда немедленного действия, 5-34
- Загрузка вашей первой программы Local Logic, 10-22
- Задание параметров конфигурации, 4-2
- Заданная позиция %AI слово статуса, 5-9

Заданная скорость %AI слово статуса, 5-10

Задержка, 7-3

Зажим заземления экрана, 3-22

Заземление

зажим заземления экрана, 3-22

кабель В/В, 3-20

система DSM314, 2-22

Запрос на обмен данными. *See*

COMM\_REQ

Запрос на обмен данными (COMMREQ),

В-2

Запуск

интерфейс аналогового момента вращения,

сервосистема, D-17

интерфейс аналоговой скорости,

сервосистема, D-15

Зона прихода в позицию %AQ команда

немедленного действия, 5-24

## И

Идентификаторы цепей В/В и

обозначения сигналов, 3-23

Изменение скорости

управление изменением, следящая ось, 8-7

Имеется ошибка модуля %I бит статуса,

5-3

Индикаторы светодиодные, А-20

Инициализация позиции, С-2

Инкрементное позиционирование, 7-21

Инкрементный датчик положения, режим,

информация, С-1

Инкрементный импульсный датчик

положения, С-4

Интегральный коэффициент обратной

связи по скорости %AQ команда

немедленного действия, 5-31

Интервал обновления

контур сервопривода, 1-5

Интерполяция и сглаживание, 16-7

Интерфейс

оператор, 1-7

Информация, режим абсолютного датчика

положения

инициализация позиции, С-2

Информация, режим инкрементного

датчика положения, С-1

Информация. Режим абсолютного

датчика

команда установки позиции, С-3

Информация. Режим абсолютного

датчика положения, С-2

цикл начального позиционирования, С-2

Использование редактора Local Logic, 10-

5, 10-26

Источник Ведущей оси, 4-23

Источники, ведущие, 8-2

## К

К12 Кабель

β Series, 24 В постоянного тока на

Сервоусилитель, 2-19

К2 Кабель

α Series - Подключение, 2-9, 2-16

Номера для Заказа Заводского Кабеля, 2-17

К3 Кабель

β Series - Подключение, 2-17

К8 Кабель

Внешний Регенеративный Резистор или

Переключатель для β Series, 2-19

Кабели для DSM302, 3-17

Кабель К1

α Series - Подключение, 2-5, 2-13

Кабель К1 к DSM314

α Series - Подключение, 2-5

Кабель К2

α Series - Подключение, 2-9

Кабель В/В

заземление, 3-20

Кабель К12

24 В постоянного тока на Сервоусилитель,

2-19

Кабель К2

β Series - Подключение, 2-16

Кабель К3

220 В перем. тока к Усилителю β Series, 2-

17

Кабель К4

Питание Двигателя α Series, 2-8

Кабель К4, Питание Двигателя

Номера компонентов для Заказа, 2-15

Подключение α Series, 2-8, 2-15

Кабель К8

Внешний Регенеративный Резистор или

Переключатель для β Series, 2-19

*Кинематические уравнения*, 7-46

Клеммная колодка

назначение контактов, 3-10

таблица быстрого подбора, 3-9

Клеммная колодка и подключения

кабелей

рисунки, 3-18

Клеммная колодка оси

детали, 3-15

компоненты, 3-12

сборочные чертежи, 3-12

Клеммная колодка, вспомогательная, 3-14

Клеммная колодка, Ось, 3-10

Клеммные колодки, 3-8

Клеммные колодки, вспомогательная

клеммная колодка сервоустройства,

3-14

Клеммные коробки

переход от монтажа на DIN-шине к

монтажу на панели, 3-12, 3-15

Ключевые слова и операторы, 12-5, 12-6

- Код статуса модуля %AI слово статуса, 5-9
  - Коды ошибок модуля DSM, специфические для CAM, 16-19
  - Коды слова статуса
    - таблица кодов, В-3
  - Количество модулей в системе, ограничения, 3-5
  - Команда CALL
    - программирование движения, 7-10
  - Команда CAM-LOAD, 16-12
  - Команда CAM-PHASE, 16-13
  - Команда номера блока
    - программирование движения, 7-10
  - Команда установки позиции, С-3
  - Команда DWELL, 7-27
  - Команда Wait, 7-28
  - Команда, Включить Ц/А, 6-7
  - Команда, движение
    - примеры использования, 7-21
  - Команда, скорость, 6-6
  - Команды
    - программа движения, 7-9
  - Команды САМ и MOVE, 16-13
  - Команды немедленного действия, использующие шестибайтовый формат, 5-19, 5-20
  - Команды программ движения
    - Тип 1
      - Вызов подпрограммы, 7-3
      - Переход, 7-3
    - Тип 2
      - Блок #**, 7-3
      - Загрузить параметр**, 7-3
      - Нуль**, 7-3
      - Скорость**, 7-3
      - Ускорение**, 7-3
    - Тип 3
      - Движение позиционирования, 7-3
      - Задержка, 7-3
      - Конец программы, 7-3
      - Непрерывное движение, 7-3
      - Ожидание, 7-3
  - Команды, Приращение позиции, 6-8
  - Комментарии, 11-2, 12-5
  - Конец программы, 7-3
  - Контактная Колодка, Сервоусилитель  $\alpha$  Series, 2-9, 2-17
  - Контрольный Список перед Включением Питания, 2-24
  - Конфигурация
    - аппаратное обеспечение, 2-25
    - Machine Edition, 2-31
    - VersaPro, 2-26
    - модуль DSM314, 4-1
  - Конфигурация - параметры, 4-2
  - Конфигурация STL-битов, 14-1
  - Конфигурация Logicmaster 90-30
    - модуль контроллера, 4-1
  - Конфигурация битов выходов лицевой панели, 14-5
  - Конфигурация данных оси
    - скорость FF%, 4-31
  - Конфигурация крейта / слота, 4-1
  - Конфигурация модуля DSM314, 4-1
  - Конфигурация модуля Motion Mate DSM314, 4-2
  - Коэффициент обратной связи по скорости %AQ команда немедленного действия, 5-27
  - Краткое описание электронного САМ, 16-1
  - Крейт / слот - конфигурация, 4-1
  - Круговой циклический САМ, 16-5
- ## Л
- Линейное изменение скорости следящего устройства
    - длительность участка компенсации, 8-8
  - Линейное ускорение, 7-22
  - Линейный циклический САМ, 16-4
- ## М
- Максимальная длительность ускорения, 7-38
  - Масштабирование
    - пример, 4-13
  - Методы реагирования, А-2
  - Минус Толчка %Q дискретная команда, 5-14
  - Многоосное программирование, 7-42
- ## Н
- Назначение контактов
    - клеммные колодки, 3-10
  - Назначения контактов сервооси 1 и 2, 3-24, 3-26
  - Назначения контактов для разъемов осей, 3-23
  - Направление вращения двигателя
    - пусковая проверка правильности работы, D-1
  - Настройка
    - аналоговая сервосистема, D-15
    - кривые реакции, D-8, D-24
    - цифровая сервосистема, D-1, D-5
  - Настройка цифровой сервосистемы GE Fanuc, D-5
  - Неисправности системы, поиск и устранение, рекомендации, D-31
  - Непрерывное движение, 7-3
  - Непрерывное движение (СMOVE), 7-23
  - Нециклический САМ, 16-4
  - Новые STL-биты STL17-CTL32, 14-2

Номер блока команды %AI слово статуса, 5-9  
 Номера блоков и переходы, 7-28  
 Номера в Каталоге  
     Силовые Кабели Двигателей для β Series, 2-17  
     Силовые Кабели Двигателей, β Series, 2-15  
     Силовые Кабели для Двигателей, α Series, 2-8  
 Нормальный останов перед командой JUMP, 7-32  
 Нуль %AQ команда немедленного действия, 5-20

## O

Обновление микропрограммы кабель IC693CBL316, 4-9  
 Обозначения сигналов, 3-23  
 Одно направление, работа, 8-6  
 Ожидание, 7-3  
 Окно  
     изменение структуры, 15-3  
 Оператор -, 12-9  
 Оператор \*, 12-9  
 Оператор /, 12-10  
 Оператор +, 12-8  
 Оператор BWAND, 12-13  
 Оператор BWNOT, 12-15  
 Оператор BWOR, 12-14  
 Оператор BWXOR, 12-14  
 Оператор MOD, 12-11  
 Операторы, 11-1, 11-3, 12-2  
 Операторы отношения, 11-4, 12-16  
 Операторы присваивания, 12-3  
 Оси  
     количество, таблица, 4-4  
 Основные элементы формы кулачка (CAM) / определения, 16-3  
 Особенности DSM314  
     простота использования, 1-2  
     универсальность и гибкость входов /выходов, 1-2  
 Останов перед командой JUMP, нормальный, 7-32  
 Остановить подачу (включение команды) %Q дискретная команда, 5-13  
 Остановить подачу (выключение команды) %Q дискретная команда, 5-13  
 Остановка перехода, 7-34  
 Остановка подачи модулем Motion Mate DSM314, 7-40  
 Ось 1, 2, Код ошибки ведущего устройства слежения %AI слово статуса, 5-9  
**Ось 1, длительности, связанные с работой с переменными Local Logic, E-5**

**Ось 2, длительности, связанные с работой с переменными Local Logic, E-6**  
 Ось 3, длительности, связанные с работой с переменными Local Logic, E-7  
 Ось 4, длительности, связанные с работой с переменными Local Logic, E-7  
 Ось, клеммная колодка, 3-10  
 Отключение питания от батареи, C-2  
 Отношение  
     A:B, 8-3  
 Отрицательный программный конец передвижения, 4-17  
 Ошибки выполнения программы Local Logic, 12-17

## П

Параметр Overtravel Limit Switch (Концевой выключатель перебега), 4-15  
 Параметры в модуле Motion Mate DSM314, 7-44  
 Параметры настройки, 4-33  
     ввод в ячейки расширенной таблицы, 4-35  
     поддерживаемые в версии 1.0, 4-33  
     поддерживаемые в версии 3.0, 4-34  
 Первое использование, абсолютный датчик положения, C-2  
 Передача данных  
     ПЛК - DSM, 1-4  
 Переключатель начальной позиции  
     пусковая проверка правильности работы, D-1  
 Переключатель начальной позиции, режим, 6-2  
 Переменная System\_Halt (Системный\_останов), 13-3  
 Переменные, 11-2, 12-2  
 Переменные оси 1, 13-5  
 Переменные оси 2, 13-6  
 Переменные оси 3, 13-7  
 Переменные оси 4, 13-8  
 Переменные, определяющие предельные значения данных  
     расчет, 4-33  
 Переменный ток - Сетевой Фильтр, 2-11, 2-17  
 Переход, 7-3  
 Переход без остановки, 7-33  
 Переход, проверка, 7-31  
 Переходы и номера блоков, 7-28  
 Переходы, S-CURVE (S-кривая), 7-35  
 Переходы, S-CURVES (S-кривая), 7-35, 7-36, 7-37  
 Переходы, безусловные, 7-29  
 Переходы, условные, 7-29  
 ПЛК – DSM  
     передача данных, 1-4

- ПЛК-управление действует – бит статуса %I, 5-3
- Плюс Толчка %Q дискретная команда, 5-14
- Побитовые логические операторы, 11-4, 12-13
- Поддержка
  - GE Fanuc Веб-сайт, 2-43
- Подключение Двигателя к Контактной Колодке Сервоусилителя
  - $\alpha$  Series, 2-9
  - $\beta$  Series, 2-16
- Подключение редактора Local Logic к модулю DSM, 10-11
- Подключения, пользователь, 3-30
- Подпрограмма, движение структура, 7-17
- Подпрограммы, 7-28
  - многоосные, 7-2
  - одноосные, 7-1
- Подсоединение заземления экрана, 3-4
- Подсоединение заземления, экран лицевой панели, 3-4
- Подстановка скорости %AQ команда немедленного действия, 5-21
- Подстановка, скорость подачи, 7-41
- Позиционирование, абсолютное, 7-21
- Позиционирование, движение (PMOVE), 7-23
- Позиционирование, инкрементное, 7-21
- Позиция
  - расчет значений, 7-46
- Позиция строба 1, 2 %AI слово статуса, 5-9
- Позиция строба 1, 2 %I бит статуса, 5-6
- Поиск Исходной позиции %Q дискретная команда, 5-14
- Положительный программный конец передвижения, 4-17
- Полоса пропускания
  - определение, D-13
- Получена новая конфигурация - бит статуса %I, 5-3
- Пользовательские подключения, 3-30
- Последовательность Подачи Питания, 2-24
- Последовательность управления, 9-5
- Постоянная времени контура позиционирования %AQ команда немедленного действия, 5-26
- Построение вашей первой программы Local Logic, 10-12
- Потеря синхронизации, 4-29
- Пояснения к модулю DSM314, 1-1
- Предварительные условия для программируемого движения, 7-5
- Предел по моменту вращения %I бит статуса, 5-6
- Предупреждающие сообщения
  - Motion Editor, 7-49
- Предупреждающие сообщения, движение поиск ошибок, 7-53
- Предупреждение / ошибка - окончание времени ожидания, 12-18
- Прекратить все движения %Q дискретная команда, 5-13
- Привод в норме (вход 4) %I бит статуса, 5-6
- Приложение D
  - устройства обратной связи по положению, C-1
- Приложение F
  - настройка цифровой сервосистемы GE Fanuc, D-1
- Приложение A
  - Выдача сообщений об ошибках, A-1
- Пример программы планировщика коэффициента усиления, 11-7
- Пример программы по ограничению момента вращения, 11-6
- Пример программы программируемого концевого переключателя, 11-8
- Пример программы с кадрирующими стробами, 11-11
- Пример программы триггерного выхода, основанного на позиции, 11-9
- Пример, движение слежения
  - изменение отношения A
    - B, 8-5
    - отслеживание ведущего входа датчика положения 3, 8-2
    - отслеживание внутреннего источника, 8-3
    - примеры отношений A
      - B, 8-4
      - работа в одном направлении, 8-6
      - фиксация скорости, 8-5
  - Пример, движение слежения в комбинации с программой движения, 9-5
- Пример, стандартный режим
  - Dwell, 7-27
  - Feedhold (Остановка подачи), 7-40
  - JUMP без остановки, 7-33
  - S-Curve, переходы до средней точки ускорения или замедления, 7-36
  - S-Curve, переходы после средней точки ускорения или замедления, 7-36
  - S-кривая, переходы к более высокому значению скорости при ускорении или к более низкому значению скорости при замедлении, 7-37
  - Безусловный переход, 7-29
  - Зависание модуля Motion Mate DSM314 при отсутствии расстояния, 7-26
  - изменение режима ускорения на профиле, 7-26
  - команда перехода с последующей командой PMOVE, 7-35
  - комбинирование PMOVE и CMOVE, 7-25

- максимальная длительность ускорения, 7-38  
 Многоосное программирование, 7-42  
 не хватает расстояния, чтобы выйти на заданную скорость, 7-26  
 нормальный останов перед командой JUMP, 7-32  
 остановка перехода, 7-34  
 подстановка (коррекция) скорости подачи, 7-41  
 проверка условия перехода, 7-31  
 Пример, Движение слежения в комбинации с Jog (Толчок), 9-1  
 Примеры программирования Local Logic, 11-5  
 Приращение позиции %AQ команда немедленного действия, 5-21, 5-24  
 Приращение позиции, команда, 6-8  
 Проверка, переход, 7-31  
 Программа Find Home (Начальное позиционирование), 6-2  
 Программа активна - %I бит статуса, 5-5  
 Программа движения  
   ключевые слова, 7-7  
   команды, 7-9  
   переменные, 7-8  
   синтаксис и команды, 7-7  
   структура, 7-17  
   типы команд, 7-3  
   формат, 7-6  
 Программа движения, условия остановки, 7-5  
 Программа, движение  
   структура, 7-17  
 Программирование, многоосное, 7-42  
 Программируемые движения, 7-25  
 Программное обеспечение  
   конфигурация и программирование, 1-7, 2-1  
 Программы движения  
   многоосные, 7-2  
   одноосные, 7-1  
 Пропорциональный коэффициент обратной связи по скорости %AQ команда немедленного действия, 5-31  
 Процедура настройки цифрового сервопривода, D-6  
 Пуск  
   поиск неисправностей, D-3  
 Пустое пространство, 12-4
- Р**
- Работа  
   краткое описание, 1-8  
 Работа в одном направлении, 8-6  
 Разъем  
   CX11, 2-19  
   Hirose 20 контактов PCR, 2-19  
   JS1B, 2-5, 2-13  
   JX5, 2-18  
   K12,  $\beta$  Series, 2-19  
   станок, 1-7  
 Разъем JX5, 2-18  
 Разъем последовательной связи, 3-3  
 Разъемы В/В, 3-3  
 Разъемы осей  
   назначения контактов, 3-23  
 Расчеты трапецевидного графика скорости, 7-47  
 Редактор масштабирования САМ и конфигурация аппаратного обеспечения, 16-14  
 Режим  
   абсолютный датчик положения, C-2  
   аналоговая разводка, 2-21  
   инкрементный датчик положения, C-1  
   слежение, 1-8, 1-10  
   стандартный, 1-8, 1-9  
   цифровой последовательный датчик положения, C-1  
 Режим абсолютного датчика положения  
   включение питания модуля Motion Mate DSM314, C-3  
 Режим абсолютного датчика положения, информация  
   цикл начального позиционирования, C-2  
 Режим исходной позиции (Режим Поиска Исходной Позиции), 4-21  
 Режим переключателя начальной позиции, 6-2  
 Режимы Move+ и Move-, 6-4  
 Режимы, последовательный датчик положения, C-1  
 Резонанс  
   определение, D-13  
 Рекомендации, поиск и устранение неисправностей системы, D-31
- С**
- САМ, типы, 16-4  
 САМ-команда, 16-10  
 Сброс флагов стробов 1, 2 %Q дискретная команда, 5-14  
 Сбросить ошибку %Q дискретная команда, 5-12  
 Светодиод CFG, 3-2  
 Светодиод EN1, 3-2  
 Светодиод EN2, 3-2  
 Светодиод EN3, 3-2  
 Светодиод EN4, 3-2  
 Светодиод ОК, 3-2  
 Светодиод состояния, 3-2  
 Светодиодные индикаторы  
   модуль DSM302, A-20  
 Светодиодный индикатор ОК, A-20  
 Светодиодный индикатор статуса (состояния), A-20  
 Семантические ошибки, 12-20

- Семантические предупреждения, 12-22
  - Серводвигатели цифровые, 4-27
  - Сервооси 1 и 2
    - назначение контактов, 3-24, 3-26
  - Сервоуправление, 2-1
  - Сервоусилитель,  $\alpha$  Series
    - Подключение 24 В постоянного тока, 2-19
    - подключение 3-фазного электропитания ~ 220 В, 2-11
    - Подключение АВАРИЙНОГО ОСТАНОВА, 2-18
    - Подключение Аварийного Останов Станка, 2-11, 2-18
    - Подключение Датчика положения Двигателя, 2-9, 2-16
    - Подключение к Сетевому Фильтру, 2-17
    - Подключение Питания 220 В ~ 3 фазы, 2-17
    - Подключение Силового Кабеля Двигателя, 2-7
  - Сервоусилитель,  $\beta$  Series
    - Подключение Внешнего Регенеративного Резистора или Перемычки, 2-19
    - Подключению к Motion Mate DSM314, 2-13
  - Сетевой Фильтр, переменный ток, 2-11, 2-17
  - Силовой Кабель для Двигателей
    - Заводское изготовление, Номера для Заказа
    - для  $\alpha$  Series, 2-8, 2-15
  - Синтаксические ошибки, 12-19
  - Синхронизация САМ-движения с внешними событиями, 16-18
  - Синхронизация, потеря, 4-29
  - Система Motion Mate DSM314
    - НМИ (Человеко-Машинный Интерфейс), 2-1
    - Включение Питания, 2-24
    - Входы / Выходы, 2-1
    - Двигатели, 2-1
    - Иллюстрация, 2-2
    - Модуль Контроллера, 2-4
    - Рекомендации, 2-44
    - Сервоусилители, 2-1
  - Система, рекомендации по поиску и устранению неисправностей, D-31
  - Системные переменные Local Logic, 13-2
  - Системы Заземления
    - Корпусная Земля, 2-23
    - Системная Земля, 2-23
  - Скорость, 7-3
    - генератор, внутренний источник, 8-3
    - расчет значений, 7-46
    - фиксация, 8-5
  - Скорость FF% (Значение Упреждения по скорости подачи, 4-31
  - Скорость подачи, подстановка, 7-41
  - Скорость при 10 В, 4-31
  - Скорость Толчка %AQ команда немедленного действия, 5-26
  - Скорость, команда, движение, 6-6
  - Слежение
    - Длительность участка линейного компенсирующего изменения скорости %AQ команда немедленного действия, 5-30
    - управление линейным изменением скорости оси, 8-7
  - Соединительный элемент CX4, 2-11
  - Соединительный элемент JF1, 2-9
  - Сообщения об ошибках Motion Editor, 7-49
  - Сообщения об ошибках, движение поиск ошибок, 7-53
  - Сопрягающиеся сектора, 16-8
  - Статус переполнения, 12-17
  - Стробы позиции, 1-5
  - Структура
    - изменение окна, 15-3
  - Структура окна редактора Local Logic SIMPLICITY Machine Edition, 10-8**
  - CX4 - Соединительный элемент, 2-11
  - Схемы упрощенные, 3-38
- ## Т
- Таблица переменных Local Logic, 10-9
  - Таблица пользовательских данных Local Logic, 13-4**
  - Текущая позиция %AI слово статуса, 5-9
  - Текущая скорость %AI слово статуса, 5-10
  - Терминология, Определения
    - Заданная Позиция, 1-9
    - Заданная Скорость, 1-9
    - Ошибка по Положению, 1-9
    - Текущая Позиция, 1-9
    - Текущая Скорость, 1-9
  - Технические характеристика В/В
    - оптически изолированный включающий релейный выход, 3-45
  - Технические характеристики В/В, 3-38
    - дифференциальные / однополярные 5В-входы, 3-39
    - дифференциальные +/- 10 В аналоговые входы, 3-46
    - дифференциальные 5В-выходы, 3-42
    - однополярные 5В-входы / выводы, 3-42
    - однополярный +/- 10 В аналоговый выход, 3-47
    - однополярный 5В-вход-приемник, 3-40
    - оптически изолированные входы-источники/ входы-приемники 24 В, 3-41
    - оптически изолированный выход, 24 В пост. тока, 3-44
    - питание +5 В, 3-48
  - Типы команд
    - программа движения, 7-3
  - Типы ускорения, 7-22
  - Типы цепей Ввода/Вывода (В/В), 3-8

Толчковое передвижение при помощи  
DSM302, 6-6  
Толчок  
пример, 2-42  
Требования, 16-21  
Требования к системе, 10-4

## У

Управление Станком, 2-1  
Упреждение по скорости %AQ команда  
немедленного действия, 5-26  
**Уравнения изменения скорости,  
ограничивающие рывки, 7-48**  
Уравнения треугольного графика  
скорости, 7-48  
Усилители  
цифровые, 1-13  
Усилитель  
SVU цифровой, 1-11  
Ускорение, 7-3  
длительность, максимальная, 7-38  
линейное, 7-22  
расчет значений, 7-46  
типы, 7-22  
Ускорение Толчка %AQ команда  
немедленного действия, 5-26  
Ускорение, S-кривая, 7-22  
Условия, приводящие к остановке  
программы движения, 7-5  
Условные операторы, 12-4  
Условные переходы, 7-29  
Установить бит Configuration Complete  
(Конфигурация закончена) %AQ  
команда немедленного действия, 5-  
34  
Установить позицию %AQ команда  
немедленного действия, 5-21  
Установка модуля Motion Mate DSM302  
рекомендуемая процедура, 3-5  
Устройства обратной связи, типы  
инкрементный импульсный датчик  
положения, С-4

## Ф

Фиксация, скорость, 8-5  
Фильтр уставок момента вращения %AQ  
команда немедленного действия, 5-  
31  
Формат кода ошибки, А-2, А-15  
Функции цепей В/В и назначения  
контактов, 3-23  
Функциональные блок-схемы для 2-й оси  
DSM302  
ось 2 аналоговый вход от ведущего  
источника, 1-10  
Функциональные блок-схемы для 2-й оси  
DSM314

ось 1 датчик положения 3 от ведущего  
источника / внутреннее ведущее  
устройство, 1-10  
Функция ABS, 12-12  
Функция SNAP и назначения CTL-битов,  
14-5

## Х

Характеристики DSM314, 1-1  
высокая производительность, 1-1

## Ц

Ц/А команда, включить, 6-7  
Цикл начального позиционирования, С-2  
Цикл начальной позиции, 6-1  
Цифровой последовательный датчик  
положения, режимы, С-1  
Цифровые выходы / CTL-переменные, 13-  
4  
Цифровые серводвигатели, список, 4-27

## Ч

Числовые константы, 12-1

## Э

Электромонтажные подключения, 3-30