

Семейства MELSEC FX

Программируемые контроллеры

Пособие для начинающего программиста

**FX1S, FX1N,
FX2N, FX2NC,
FX3G, FX3U, FX3UC**

Об этом руководстве

Имеющиеся в этом руководстве тексты, изображения, диаграммы и примеры служат исключительно для разъяснения установки, обслуживания, программирования и применения программируемых контроллеров MELSEC серий FX1S, FX1N, FX2N, FX2NC, FX3G, FX3U и FX3UC.

Если у вас возникнут вопросы по монтажу и эксплуатации описываемых в этом руководстве приборов, не колеблясь обратитесь в ваше региональное торговое представительство или к одному из региональных партнеров по сбыту (см. обложку).

Актуальную информацию и ответы на часто задаваемые вопросы вы можете найти в интернете (www.mitsubishiautomation.de).

MITSUBISHI ELECTRIC EUROPE B.V. сохраняет за собой право в любое время совершать технические изменения в этом руководстве без особых указаний.

© 11/2005–07/2009

Указания по безопасности

Кому адресовано это руководство

Это руководство адресовано только квалифицированным электрикам, получившим признанное образование и знающим стандарты безопасности в технике автоматизации. Проектировать, устанавливать, вводить в эксплуатацию, обслуживать и проверять приборы разрешается только квалифицированным электрикам, получившим признанное образование и знающим стандарты безопасности в технике автоматизации. Вмешательства в аппаратуру и программное обеспечение нашей продукции, не описанные в этом руководстве, разрешены только нашему квалифицированному персоналу

Использование по назначению

Программируемые контроллеры серий FX1S, FX1N, FX2N, FX2NC, FX3G, FX3U и FX3UC предназначены только для тех областей применения, которые описаны в этом руководстве. Соблюдайте все содержащиеся в руководстве характеристики. Продукция разработана, изготовлена, проверена и задокументирована с соблюдением норм безопасности. Неквалифицированные вмешательства в аппаратуру или программное обеспечение или несоблюдение содержащихся в этом руководстве или нанесенных на саму продукцию предупреждений могут привести к тяжелым травмам или материальному ущербу. В сочетании с программируемыми контроллерами серий FX1S, FX1N, FX2N, FX2NC, FX3G, FX3U и FX3UC разрешается использовать приборы и модули расширения, рекомендуемые фирмой МИЦУБИСИ ЭЛЕКТРИК.

Любое иное использование, выходящее за рамки сказанного, считается использованием не по назначению.

Предписания, относящиеся к безопасности

При проектировании, установке, вводе в эксплуатацию, техническом обслуживании и проверке приборов должны соблюдаться предписания по технике безопасности и охране труда, относящиеся к специфическому случаю применения. В особенности должны соблюдаться следующие предписания (без претензии этого перечня на полноту):

- Предписания электротехнического союза (VDE)
 - VDE 0100
Правила возведения силовых электроустановок с номинальным напряжением до 1000 В
 - VDE 0105
Эксплуатация силовых электроустановок
 - VDE 0113
Электроустановки с электронными компонентами оборудования
 - VDE 0160
Оборудование силовых электроустановок и электрических компонентов оборудования
 - VDE 0550/0551
Правила установки трансформаторов
 - VDE 0700
Безопасность электрических приборов, предназначенных для домашнего пользования и подобных целей
 - VDE 0860
Правила безопасности для электронных приборов и их принадлежностей, работающих от сети и предназначенных для домашнего пользования и подобных целей.
- Правила противопожарной безопасности

- Правила предотвращения несчастных случаев
 - VBG № 4
Электроустановки и электрические компоненты оборудования

Предупреждения об опасностях

Отдельные указания имеют следующее значение:



ОПАСНО:

Означает, что непринятие соответствующих мер предосторожности опасно для жизни и здоровья пользователя.



ВНИМАНИЕ:

Означает, что непринятие соответствующих мер предосторожности может привести к повреждению прибора или иного имущества.

Общие предупреждения об опасностях и профилактические меры безопасности

Нижеследующие предупреждения об опасностях следует рассматривать как общие правила обращения с программируемым контроллером в сочетании с другими приборами. Эти указания вы должны обязательно соблюдать при проектировании, монтаже и эксплуатации управляющей установки.

**ОПАСНО**

- *Соблюдать предписания по технике безопасности и охране труда, относящиеся к специфическому случаю применения. Выполнять монтаж, работать с электропроводкой и открывать блоки, компоненты и приборы необходимо в их обесточенном состоянии.*
- *Блоки, компоненты и приборы должны быть установлены в безопасном для прикосновения корпусе, оснащенном надлежащей крышкой и защитным устройством.*
- *Если приборы подключаются к сети постоянной проводкой, в оборудование здания должен быть встроен выключатель для отделения от сети по всем полюсам и предохранитель.*
- *Регулярно проверяйте на отсутствие дефектов изоляции или мест обрыва токоведущие кабели и провода, которыми соединены приборы. При обнаружении неисправностей в соединениях следует сразу обесточить приборы и соединение, а затем заменить дефектный кабель.*
- *Перед вводом в эксплуатацию проверьте, совпадает ли допустимый диапазон сетевого напряжения с местным сетевым напряжением.*
- *Необходимо принять требуемые профилактические меры безопасности, чтобы обрыв провода или жилы на сигнальной стороне не мог привести к неопределенным состояниям.*
- *Примите требуемые профилактические меры, чтобы можно было надлежащим образом возобновлять работу прерванной программы после провалов и падений напряжения. При этом опасные рабочие состояния не должны возникать даже на короткое время.*
- *Для программируемых контроллеров недостаточно использовать устройства защиты от токов повреждения по DIN VDE 0641, часть 1-3, в качестве единственной защиты при косвенных прикосновениях. Для программируемых контроллеров должны быть приняты дополнительные или иные меры защиты.*
- *Устройства аварийного выключения в соответствии со стандартом EN60204 / IEC 204 VDE 0113 должны оставаться работоспособными во всех рабочих режимах программируемого контроллера. Деблокировка устройства аварийного выключения не должна вызывать неконтролируемого или неопределенного повторного запуска.*
- *Чтобы обрыв провода или жилы на сигнальной стороне не мог привести к неопределенным состояниям в системе управления, в аппаратуре и программном обеспечении должны быть приняты соответствующие профилактические меры безопасности.*
- *При использовании модулей следует всегда строго соблюдать расчетные электрические и физические параметры.*

Содержание

| | | |
|----------|--|------|
| 1 | Введение | |
| 1.1 | Это руководство... | 1-1 |
| 1.2 | Дополнительная информация... | 1-1 |
| 2 | Программируемые контроллеры | |
| 2.1 | Что такое программируемый контроллер? | 2-1 |
| 2.2 | Выполнение программы в контроллере | 2-2 |
| 2.3 | Семейство MELSEC FX... | 2-4 |
| 2.4 | Выбор контроллера | 2-5 |
| 2.5 | Конструкция контроллеров | 2-6 |
| 2.5.1 | Входные и выходные контуры | 2-6 |
| 2.5.2 | Описание базовых блоков MELSEC FX1S... | 2-6 |
| 2.5.3 | Описание базовых блоков MELSEC FX1N | 2-7 |
| 2.5.4 | Описание базовых блоков MELSEC FX2N | 2-7 |
| 2.5.5 | Описание базовых блоков MELSEC FX2NC | 2-8 |
| 2.5.6 | Описание базовых блоков MELSEC FX3G | 2-8 |
| 2.5.7 | Описание базовых блоков MELSEC FX3U | 2-9 |
| 2.5.8 | Описание базовых блоков MELSEC FX3UC | 2-9 |
| 2.5.9 | Глоссарий функциональных компонентов | 2-10 |
| 3 | Основы программирования | |
| 3.1 | Структура управляющей команды | 3-1 |
| 3.2 | Биты, байты и слова | 3-2 |
| 3.3 | Системы счисления | 3-2 |
| 3.4 | Базовый набор команд | 3-5 |
| 3.4.1 | Начало логических операций | 3-6 |
| 3.4.2 | Вывод или присвоение результата логической операции | 3-6 |
| 3.4.3 | Учет датчиков | 3-8 |
| 3.4.4 | Операции AND | 3-9 |
| 3.4.5 | Операции OR | 3-11 |
| 3.4.6 | Команды соединения логических операций | 3-12 |
| 3.4.7 | Выполнение логических операций в зависимости от фронта сигнала | 3-14 |

| | | |
|----------|---|------|
| 3.4.8 | Установка и сброс | 3-15 |
| 3.4.9 | Сохранение, считывание и стирание результата логической операции ... | 3-17 |
| 3.4.10 | Выработка импульса | 3-18 |
| 3.4.11 | Функция главного выключателя (команды MC и MCR) | 3-19 |
| 3.4.12 | Инвертирование результата логической операции | 3-20 |
| 3.5 | Безопасность превыше всего! | 3-21 |
| 3.6 | Реализация задач управления | 3-23 |
| 3.6.1 | Установка охранной сигнализации | 3-23 |
| 3.6.2 | Шторные ворота | 3-28 |
| 4 | Подробное описание операндов | |
| 4.1 | Входы и выходы | 4-1 |
| 4.2 | Маркеры | 4-3 |
| 4.2.1 | Специальные маркеры | 4-4 |
| 4.3 | Таймеры | 4-5 |
| 4.4 | Счетчики | 4-8 |
| 4.5 | Регистры | 4-11 |
| 4.5.1 | Регистры данных | 4-11 |
| 4.5.2 | Специальные регистры | 4-12 |
| 4.5.3 | Регистры файлов | 4-13 |
| 4.6 | Советы по программированию таймеров и счетчиков | 4-14 |
| 4.6.1 | Косвенное задание значений для таймеров и счетчиков | 4-14 |
| 4.6.2 | Задержка выключения | 4-17 |
| 4.6.3 | Задержка включения и выключения | 4-18 |
| 4.6.4 | Датчик тактовых импульсов | 4-19 |
| 5 | Продвинутое программирование | |
| 5.1 | Обзор прикладных команд | 5-1 |
| 5.1.1 | Ввод прикладных инструкций | 5-6 |
| 5.2 | Команды передачи данных | 5-7 |
| 5.2.1 | Передача отдельных данных с помощью команды MOV | 5-7 |
| 5.2.2 | Групповая передача битовых операндов | 5-9 |
| 5.2.3 | Передача взаимосвязанных данных командой BMOV | 5-10 |
| 5.2.4 | Передача одних и тех же данных в несколько целевых операндов (FMOV) | 5-11 |
| 5.2.5 | Обмен данными со специальными модулями | 5-12 |

| | | |
|-------|--|------|
| 5.3 | Операторы сравнения..... | 5-15 |
| 5.3.1 | Команда CMP | 5-15 |
| 5.3.2 | Сравнения в рамках логических операций | 5-17 |
| 5.4 | Арифметические операции | 5-20 |
| 5.4.1 | Сложение..... | 5-21 |
| 5.4.2 | Вычитание..... | 5-22 |
| 5.4.3 | Умножение | 5-23 |
| 5.4.4 | Деление | 5-24 |
| 5.4.5 | Комбинирование арифметических операций | 5-25 |

6 **Возможности расширения**

| | | |
|-------|--|-----|
| 6.1 | Общие сведения | 6-1 |
| 6.2 | Обзор..... | 6-1 |
| 6.2.1 | Модули расширения для дополнительных дискретных входов и выходов ... | 6-1 |
| 6.2.2 | Аналоговые модули ввода-вывода | 6-1 |
| 6.2.3 | Коммуникационные модули | 6-2 |
| 6.2.4 | Модули позиционирования | 6-2 |
| 6.2.5 | Панели интерфейса “человек-машина” | 6-2 |

7 **Обработка аналоговых значений**

| | | |
|-------|--|-----|
| 7.1 | Аналоговые модули | 7-1 |
| 7.1.1 | Критерии выбора аналоговых модулей..... | 7-3 |
| 7.1.2 | Адаптеры, адаптерные модули и специальные модули | 7-4 |
| 7.2 | Обзор аналоговых модулей | 7-5 |

Указатель ключевых слов

1 Введение

1.1 Это руководство...

...призвано облегчить первые шаги при работе с программируемыми контроллерами MELSEC семейства FX. В особенности оно адресовано пользователям, которые еще не имеют опыта в программировании контроллеров.

Однако это руководство облегчит “переход” на контроллеры MELSEC семейства FX и программистам, которые до сих пор работали с контроллерами других изготовителей.

Для обозначения различных приборов одной серии в этом руководстве применяется безразличный символ “о”. Так, например, под обозначением “FX1S-10о-оо” следует понимать все контроллеры, начинающиеся с “FX1S-10”, т. е. FX1S-10 MR-DS, FX1S-10 MR-ES/UL, FX1S-10 MT-DSS и FX1S-10 MT-ESS/UL.

1.2 Дополнительная информация...

...и более подробные описания отдельных приборов содержатся руководствах по эксплуатации или монтажу отдельных модулей.

Технический каталог MELSEC FX, артикул 147540, содержит обзорную информацию о контроллерах MELSEC семейства FX. Кроме того, в нем имеется информация о возможностях расширения и поставляемых принадлежностях.

Первые шаги при работе со средой программирования облегчит пособие для начинающего программиста GX Developer FX.

Подробное описание всех программных команд вы найдете в руководстве по программированию MELSEC семейства FX, артикул 154315.

Руководства и каталоги можно бесплатно скачать с сайта Мицубиси (www.mitsubishielectric.ru).

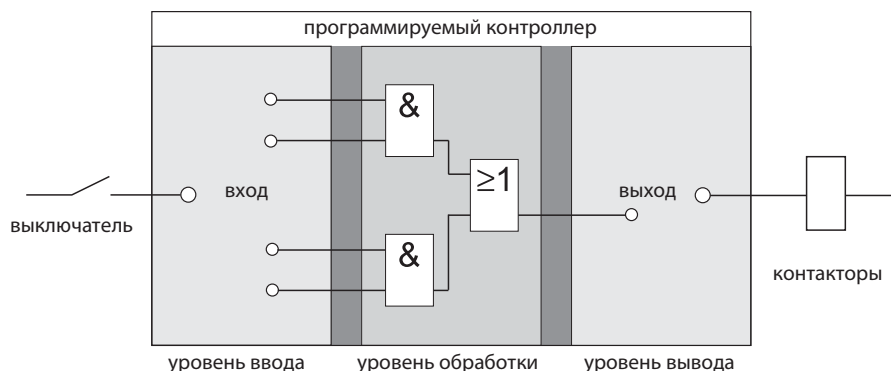
2 Программируемые контроллеры

2.1 Что такое программируемый контроллер?

В отличие от системы управления, функционирование которой зависит только от электрического монтажа, работа программируемого контроллера определяется его программой. Конечно, для соединения с внешним миром электропроводка нужна и программируемому контроллеру. Однако его принципиальное отличие заключается в том, что содержание памяти для хранения программы можно в любое время изменить и тем самым приспособить программу к различным задачам управления.

При работе программируемых контроллеров данные вводятся, обрабатываются, а затем снова выводятся как результаты обработки. Этот процесс подразделяется на следующие уровни:

- уровень ввода,
 - уровень обработки
- и
- уровень вывода.



Уровень ввода

Уровень ввода служит для того, чтобы передавать на уровень обработки управляющие сигналы, поступающие от выключателей, кнопок или датчиков.

Сигналы этих компонентов возникают по ходу процесса управления и подаются на входы контроллера в виде логических состояний. С уровня ввода уже предварительно подготовленные сигналы передаются на уровень обработки.

Уровень обработки

На уровне обработки сигналы, принятые и подготовленные на уровне ввода, обрабатываются хранящейся в памяти программой. При этом между ними устанавливаются логические взаимосвязи. Память для программы на уровне обработки свободно программируется. Имеется возможность в любое время изменить ход обработки, изменив или заменив хранящуюся в памяти программу.

Уровень вывода

На выходном уровне результаты, полученные при обработке входных сигналов программой, влияют на подключенные к выходам устройства, например, контакторы, сигнальные лампы, электромагнитные клапаны и т. п.

2.2 Выполнение программы в контроллере

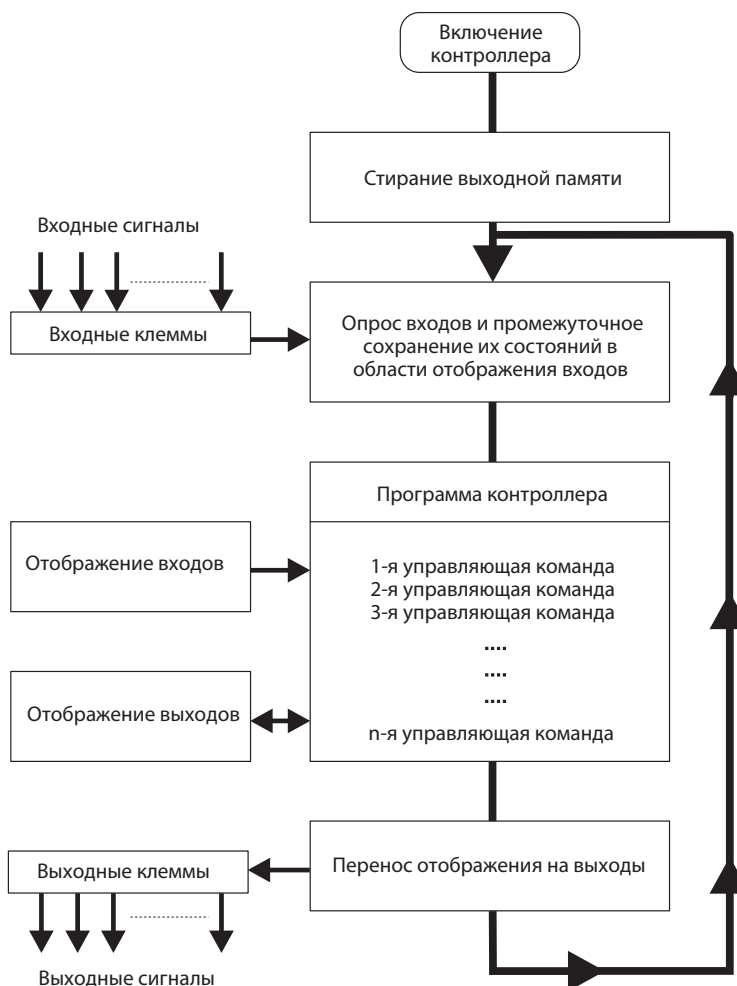
Программируемый контроллер работает по заданной программе, которая, как правило, создается вне контроллера, а затем передается в контроллер и хранится в его памяти. Для программирования важно знать, как контроллер обрабатывает программу.

Программа состоит из череды отдельных команд, определяющих функционирование контроллера. Контроллер одну за другой обрабатывает управляющие команды в запрограммированной последовательности.

Выполнение всей программы постоянно повторяется, т. е. происходит ее циклическое выполнение. Время, необходимое для выполнения программы, называется “временем цикла программы”.

Отображение процесса

При обработке программы контроллер обращается не непосредственно ко входам и выходам, а к их отображению:



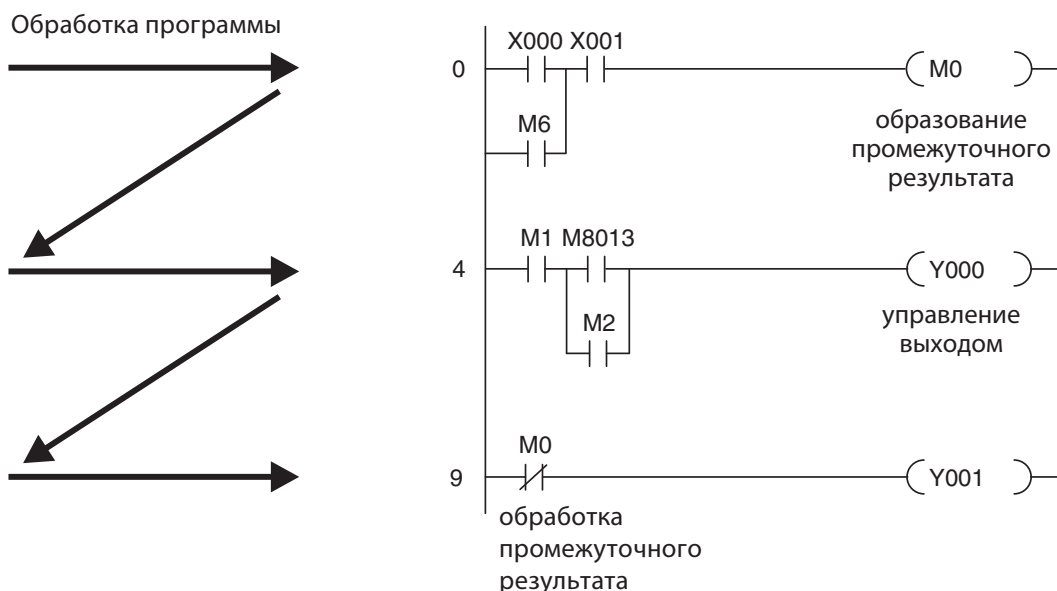
Отображение входов

В начале программного цикла состояния входов опрашиваются и сохраняются в промежуточной памяти: создается так называемое отображение входов.

Выполнение программы

Во время последующего прохождения программы контроллер обращается к состояниям входов, хранящимся в области отображения. Поэтому изменения сигналов на входах распознаются лишь при следующем программном цикле.

Программа обрабатывается сверху вниз, в последовательности ее ввода. Промежуточные результаты можно использовать уже в том же программном цикле.



Отображение выходов

Результаты логических операций, относящиеся к выходам, передаются в выходную буферную память (область отображения выходов). Лишь по окончании выполнения программы промежуточные результаты передаются на выходы. В выходной буферной памяти отображение выходов сохраняется до очередной перезаписи. После присвоения значений выходам программный цикл повторяется.

Обработка сигналов в программируемом контроллере в отличие от системы управления, запрограммированной путем электромонтажа

В случае системы управления, запрограммированной путем электромонтажа, программа задана типом функциональных звеньев и соединениями между ними (электропроводкой). Все процессы управления выполняются одновременно (параллельно). Любое изменение состояний входных сигналов сразу вызывает изменение состояний выходных сигналов.

В программируемом контроллере, если во время выполнения программы изменились состояния входных сигналов, эти изменения могут быть учтены лишь при следующем программном цикле. Этот недостаток в значительной степени компенсируется малыми значениями времени программного цикла. Время цикла программы зависит от количества и типа управляющих команд.

2.3 Семейство MELSEC FX

Компактные малые контроллеры MELSEC серии FX являются экономичным решением для небольших и средних задач управления и позиционирования в промышленности, ремесле и технике зданий. Эти контроллеры имеют от 10 до 256 встроенных входов и выходов.

Если в установке понадобится сделать изменения, все контроллеры серии FX (кроме FX1S) можно расширять и наращивать в зависимости от потребностей.

Имеются также возможности привязки к сетям обмена данными. Таким образом, контроллеры семейства FX могут коммуницировать с другими программируемыми контроллерами, а также с регулирующими системами и интерфейсами "человек-машина". Для этого контроллеры можно, во-первых, встраивать в сети МИЦУБИСИ в качестве локальных станций и, во-вторых, применять в качестве подчиненных устройств в открытых сетях (например, PROFIBUS/DP).

Кроме того, семейство MELSEC FX предоставляет возможность создания многоточечных и одноранговых сетей.

Для пользователя, желающего решить сложные задачи управления и при этом использовать многие особые функции (например, аналогово-цифровое и цифро-аналоговое преобразование, сетевая коммуникация), оптимальным выбором являются контроллеры FX1N, FX2N, FX3G, FX3U и FX3UC, имеющие возможность модульного расширения.

Контроллеры всех типов являются составной частью большого семейства MELSEC FX и совместимы между собой.

| Данные | FX1S | FX1N | FX2N | FX2NC | FX3G | FX3U | FX3UC |
|--|-------------|-------------|--------------|--------------|---|--|---|
| Макс. количество встроенных адресов входов-выходов | 30 | 60 | 128 | 96 | 60 | 128 | 96 |
| Расширяемость (максимальное количество входов-выходов) | 34 | 132 | 256 | 256 | 256 | 384 | 384 |
| Память для программы (шагов) | 2000 | 8000 | 16000 | 16000 | 32000 | 64000 | 64000 |
| Время цикла на каждую лог. инструкцию (мкс) | 0,55 – 0,7 | 0,55 – 0,7 | 0,08 | 0,08 | 0,21 / 0,42 | 0,065 | 0,065 |
| Количество инструкций (стандартных команд / команд состояния шага / особых команд) | 27 / 2 / 85 | 27 / 2 / 89 | 27 / 2 / 107 | 27 / 2 / 107 | 29 / 2 / 123 | 27 / 2 / 209 | 29 / 2 / 209 |
| Макс. число подключаемых специальных модулей | — | 2 | 8 | 4 | 8 с правой стороны 4 с левой стороны | 8 с правой стороны 10 с левой стороны | 8 с правой стороны 6 с левой стороны |

2.4 Выбор контроллера

Базовые блоки серий MELSEC FX1S, FX1N, FX2N(C) и FX3U имеют в различных исполнениях, различающихся по электропитанию и типу выходов. Имеется выбор между приборами с электропитанием 100...240 В пер. или 24 В пост. или 12...24 В пост., а также между вариантами выходов “реле” и “транзистор”.

| Серия | Входы-выходы | Тип | Кол-во входов | Кол-во выходов | Электропитание | Тип выходов |
|-------|--------------|-----------------|---------------|----------------|--|--|
| FX1S | 10 | FX1S-10 M□-□□ | 6 | 8 | по выбору 24 В пост. или 100...240 В пер. | по выбору транзисторные или релейные |
| | 14 | FX1S-14 M□-□□ | 8 | 6 | | |
| | 20 | FX1S-20 M□-□□ | 12 | 8 | | |
| | 30 | FX1S-30 M□-□□ | 16 | 14 | | |
| FX1N | 14 | FX1N-14 M□-□□ | 8 | 6 | по выбору 12...24 В пост. или 100...240 В пер. | по выбору транзисторные или релейные |
| | 24 | FX1N-24 M□-□□ | 14 | 10 | | |
| | 40 | FX1N-40 M□-□□ | 24 | 16 | | |
| | 60 | FX1N-60 M□-□□ | 36 | 24 | | |
| FX2N | 16 | FX2N-16 M□-□□ | 8 | 8 | по выбору 24 В пост. или 100...240 В пер. | по выбору транзисторные или релейные |
| | 32 | FX2N-32 M□-□□ | 16 | 16 | | |
| | 48 | FX2N-48 M□-□□I | 24 | 24 | | |
| | 64 | FX2N-64 M□-□□ | 32 | 32 | | |
| | 80 | FX2N-80 M□-□□ | 40 | 40 | | |
| | 128 | FX2N-128 M□-□□ | 64 | 64 | | |
| FX2NC | 16 | FX2NC-16 M□-□□ | 8 | 8 | 24 В пост. | по выбору транзисторные или релейные |
| | 32 | FX2NC-32 M□-□□ | 16 | 16 | | |
| | 64 | FX2NC-64 M□-□□ | 32 | 32 | | |
| | 96 | FX2NC-96 M□-□□ | 48 | 48 | | |
| FX3G | 14 | FX3G-14 M□/□□□ | 8 | 6 | 100...240 В пер. | по выбору транзисторные или релейные |
| | 24 | FX3G-24 M□/□□□ | 14 | 10 | | |
| | 40 | FX3G-40 M□/□□□ | 24 | 16 | | |
| | 60 | FX3G-60 M□/□□□ | 36 | 24 | | |
| FX3U | 16 | FX3U-16 M□-□□ | 8 | 8 | по выбору 24 В пост. или 100...240 В пер. | по выбору транзисторные или релейные |
| | 32 | FX3U-32 M□-□□ | 16 | 16 | | |
| | 48 | FX3U-48 M□-□□ | 24 | 24 | | |
| | 64 | FX3U-64 M□-□□ | 32 | 32 | | |
| | 80 | FX3U-80 M□-□□ | 40 | 40 | | |
| | 128 | FX3U-128 M□-□□ | 64 | 64 | 100...240 В пер. | по выбору транзисторные или релейные |
| FX3UC | 16 | FX3UC-16 M□/□□□ | 8 | 8 | 24 В пост. | транзистор |
| | 32 | FX3UC-32 M□/□□□ | 16 | 16 | | |
| | 64 | FX3UC-64 M□/□□□ | 32 | 32 | | |
| | 96 | FX3UC-96 M□/□□□ | 48 | 48 | | |

Для правильного выбора контроллера необходимо учитывать следующие критерии:

- Сколько сигналов должно приниматься (т. е. от внешних выключателей, кнопок и датчиков)?
- Какие и сколько функций должны коммутироваться?
- Какое электропитание имеется в моем распоряжении?
- Какие нагрузки коммутируются выходами? Если требуется коммутировать высокие нагрузки, следует применять релейные выходы. Для быстрых, бестриггерных процессов переключения используются транзисторные выходы.

2.5 Конструкция контроллеров

Все приборы, в принципе, имеют одинаковую конструкцию. В разделе 2.5.7 дан обзор наиболее важных функциональных элементов и узлов.

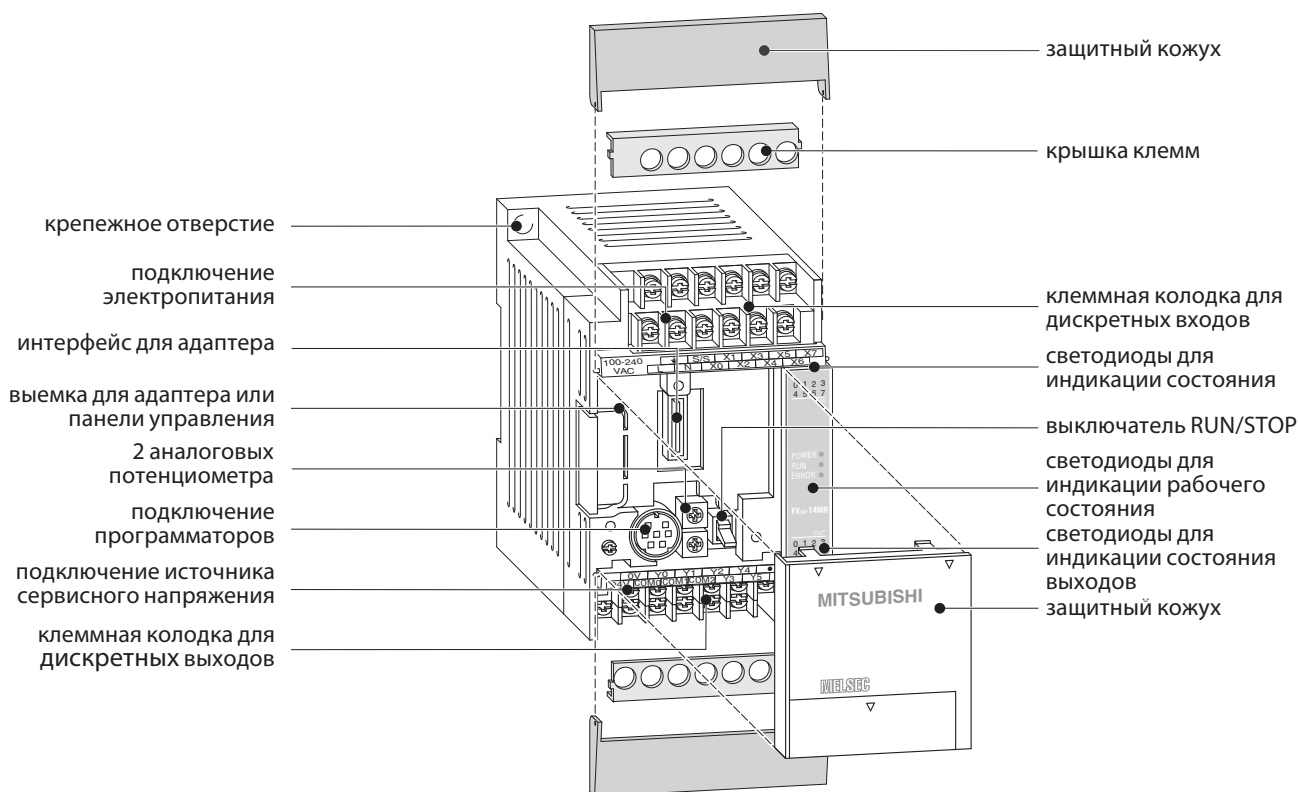
2.5.1 Входные и выходные контуры

Входные контуры выполнены в виде бесконтактных входов. Для изоляции электрических контуров в контроллере используется гальваническая развязка с помощью оптического соединителя. **Выходные контуры** выполнены либо в виде релейных, либо в виде транзисторных выходов. Для изоляции электрических контуров в контроллере в случае транзисторных модулей также используется гальваническая развязка с помощью оптического соединителя.

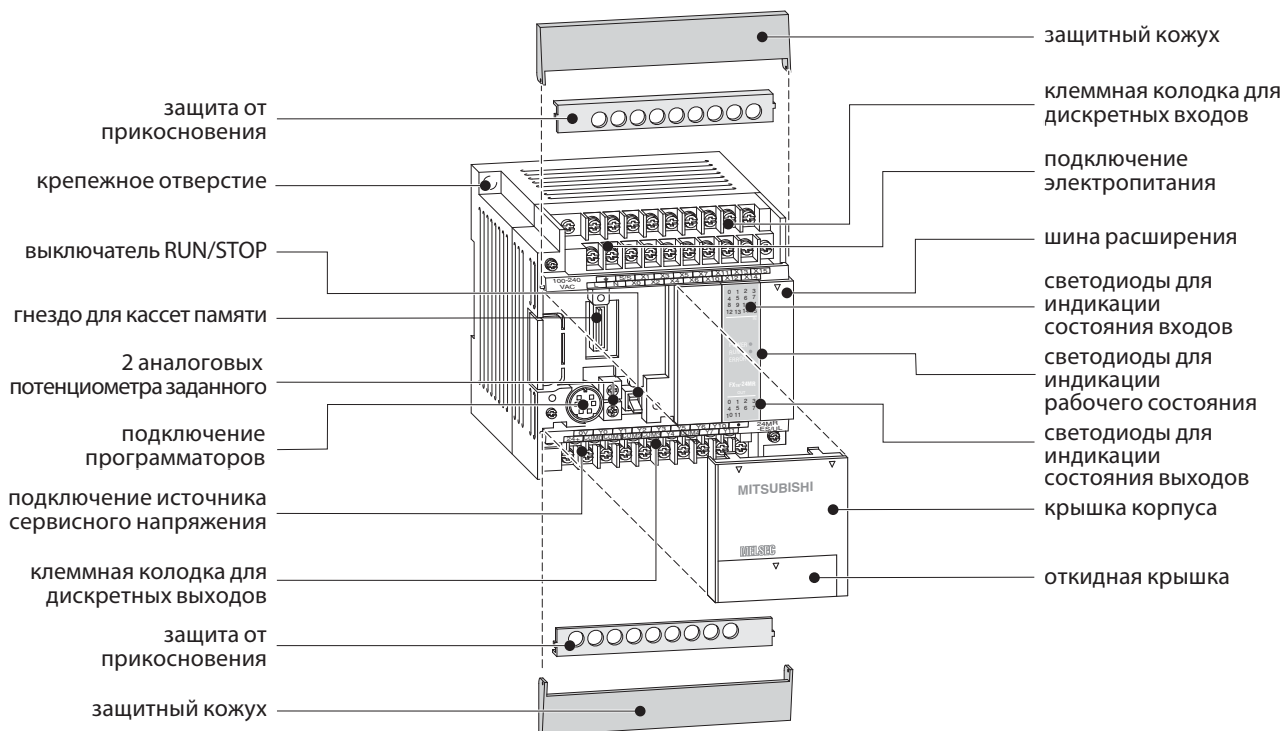
Для всех дискретных входов необходимо определенное коммутируемое входное напряжение (например, постоянное 24 В). Это напряжение можно снимать с встроенного сетевого блока контроллера. Если коммутируемое напряжение на входе ниже указанного номинального значения (< 24 В), вход не обрабатывается.

Максимальный выходной ток составляет: в случае релейных модулей 2 А при переменном напряжении 250 В и омической нагрузке; в случае транзисторных модулей 0,5 А при постоянном напряжении 24 В и омической нагрузке.

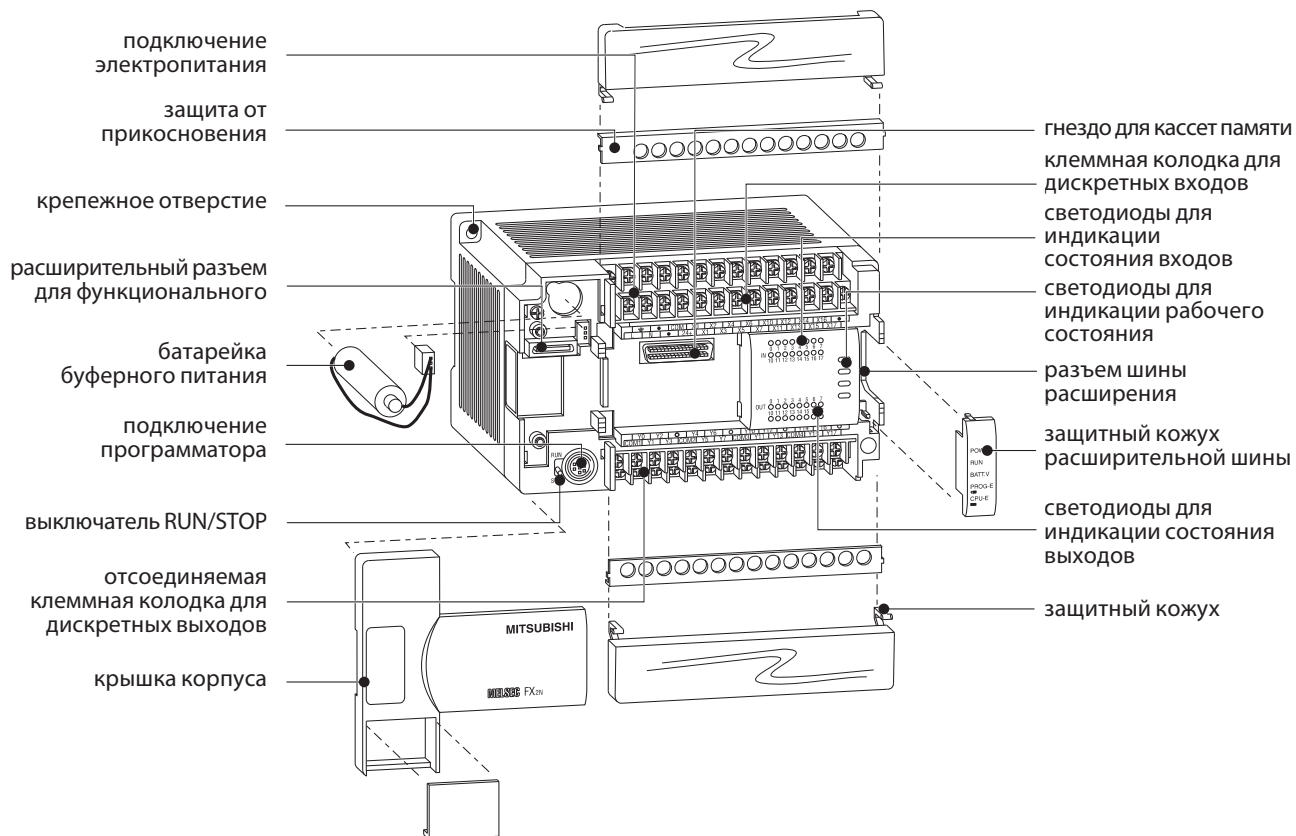
2.5.2 Описание базовых блоков MELSEC FX1S



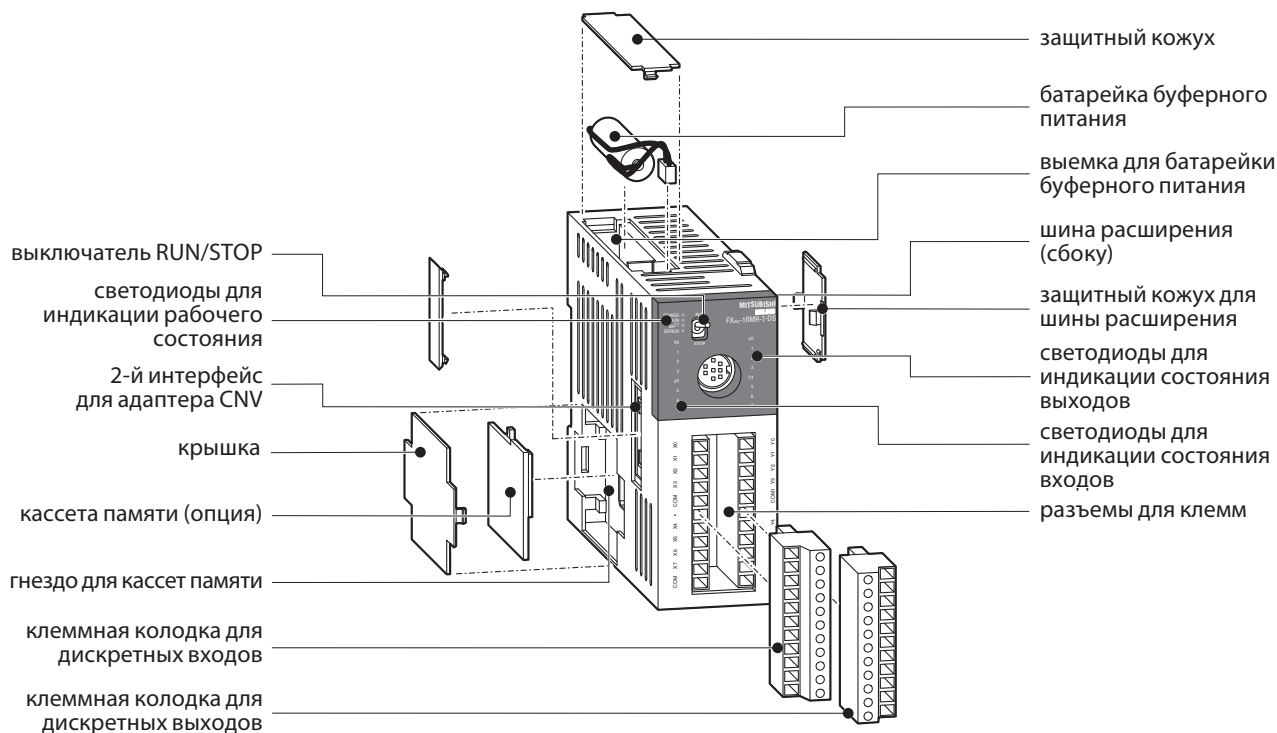
2.5.3 Описание базовых блоков MELSEC FX1N



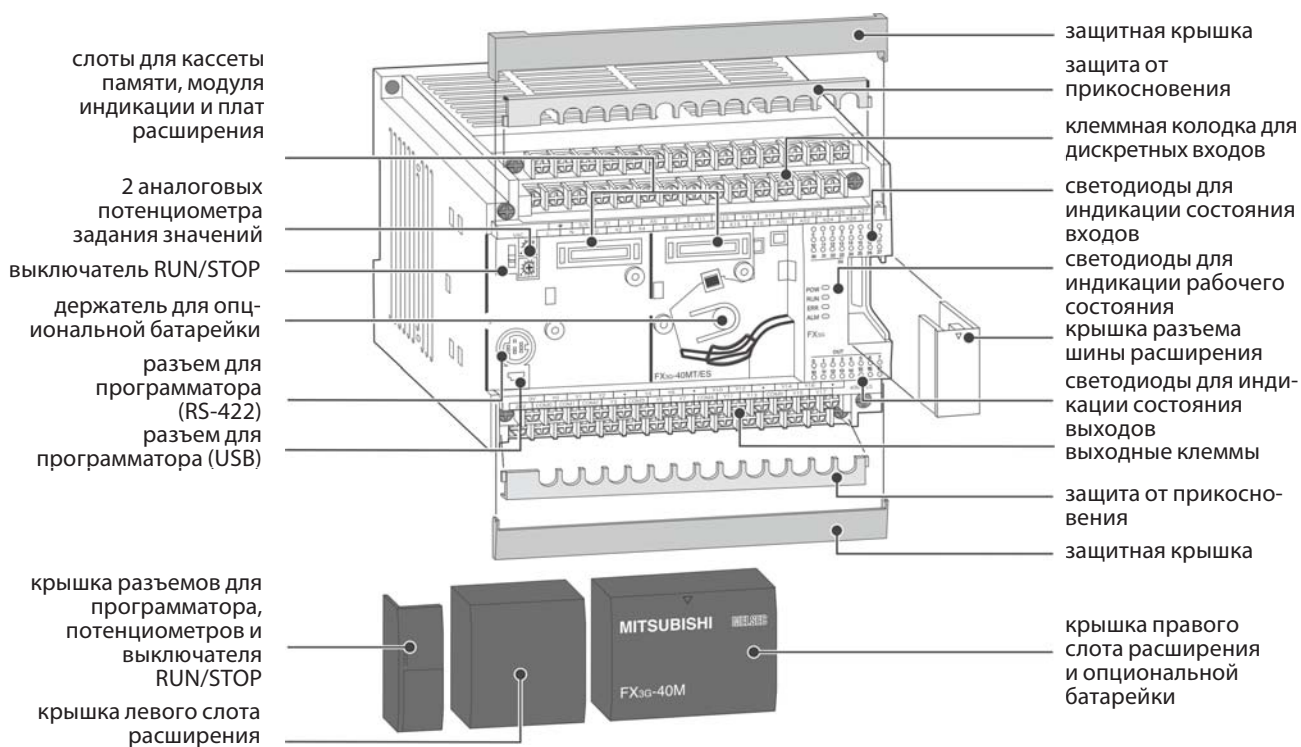
2.5.4 Описание базовых блоков MELSEC FX2N



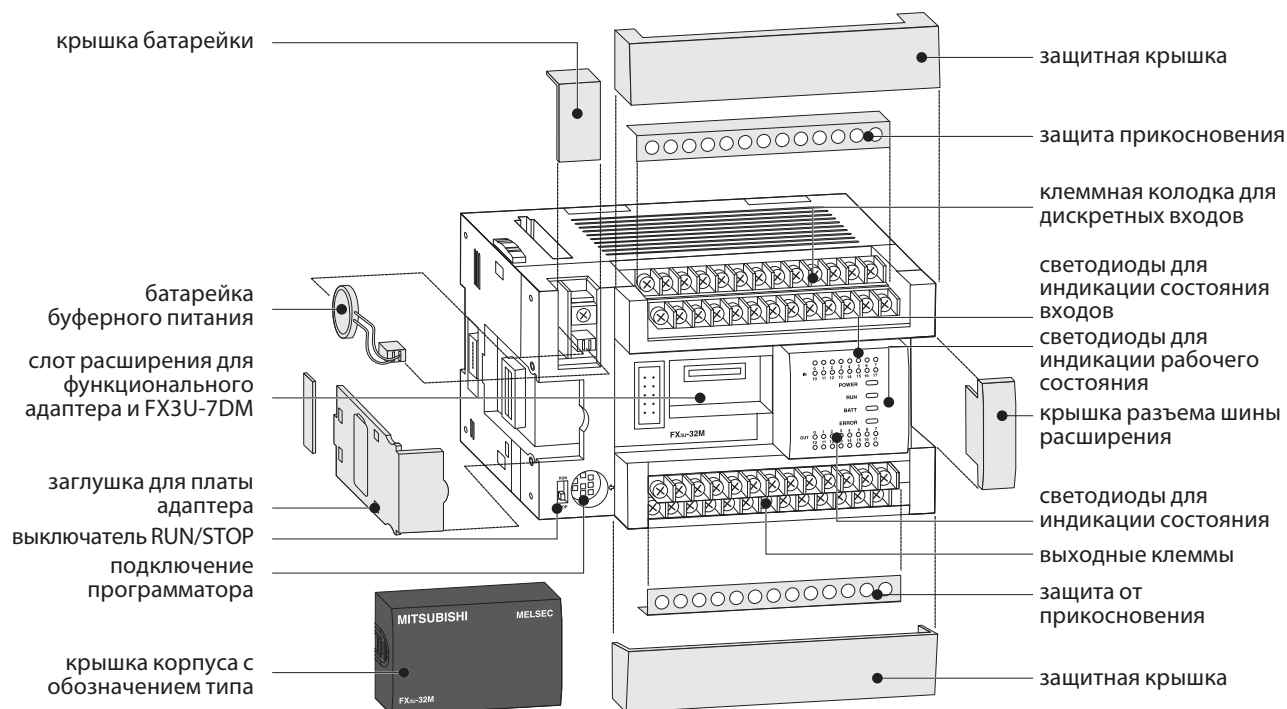
2.5.5 Описание базовых блоков MELSEC FX2NC



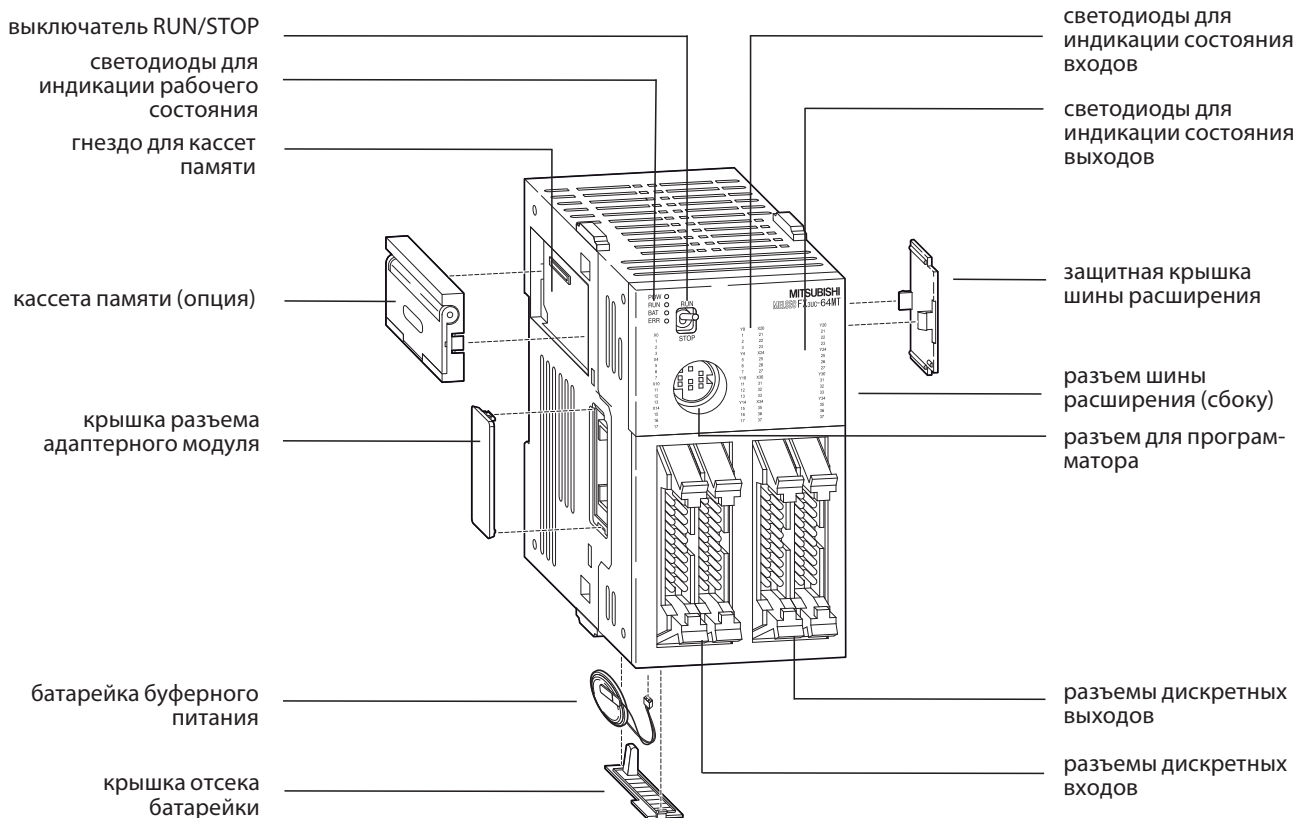
2.5.6 Описание базовых блоков MELSEC FX3G



2.5.7 Описание базовых блоков MELSEC FX3U



2.5.8 Описание базовых блоков MELSEC FX3S



2.5.9 Глоссарий функциональных компонентов

В следующей таблице разъяснено назначение и принцип работы отдельных компонентов и узлов программируемого контроллера.

| Компонент | Описание |
|---|--|
| Разъем для плат адаптеров | К этому интерфейсу можно подсоединять опциональные расширительные адаптеры. Адаптеры имеются в различных исполнениях для всех серий FX (кроме FX2NC). Они позволяют оснастить базовый блок дополнительными расширениями или коммуникационными интерфейсами. Адаптеры можно вставлять непосредственно в выемку. |
| Разъем для программаторов | К этому разъему можно подключить ручной программатор FX-20P-E, а также внешний персональный компьютер или ноутбук со средой программирования (например, GX Developer/FX). |
| EEPROM | Запоминающие устройства двустороннего действия, в которые записывается рабочая программа из среды программирования и из которых она считывается. Эти запоминающие устройства являются постоянными запоминающими устройствами, т. е. они сохраняют информацию даже при выпадении напряжения и поэтому не нуждаются в буферном питании от батарейки. |
| Гнездо для кассет памяти | В этот разъем можно вставлять опциональные кассеты памяти. При вставлении этих кассет внутренняя память контроллера отключается, после чего обрабатывается только программа, находящаяся во вставленной кассете памяти. |
| Шина расширения | Помимо дополнительных модулей ввода-вывода, к этой шине расширения можно подключить специальные модули для иных расширений программируемого контроллера. Обзор таких устройств содержится в разделе 6 этого руководства. |
| Аналоговые потенциометры | С помощью аналоговых потенциометров можно устанавливать заданные значения. Соответствующую настройку можно опрашивать с помощью программы и использовать для таймеров, вывода импульсов или подобных целей (см. раздел 4.6.1) |
| Источник сервисного напряжения | Источник сервисного напряжения (не имеется у FX2NC, FX3UC) поставляет регулируемое постоянное напряжение 24 В для питания входных сигналов и датчиков. Допустимая нагрузка этого источника напряжения зависит от типа контроллера (например, FX1S, FX1N и FX3G: 400 мА, от FX2N-16M□-□□ до FX2N-32M□-□□: 250 мА, от FX2N-48M□-□□ до FX2N-64M□-□□: 460 мА) |
| Дискретные входы | Через дискретные входы принимаются управляющие сигналы от подключенных к ним выключателей, кнопок или датчиков. Могут определяться состояния "ВКЛ." (напряжение приложено) или ВыКЛ. (напряжения не имеется). |
| Дискретные выходы | В зависимости от применения и типа выхода, к дискретным выходам можно подключать сервоэлементы и исполнительные устройства, например, контакторы. |
| Светодиоды состояния входов | Светодиоды состояний входов показывают, к какому входу приложен сигнал, т. е. определенное напряжение. Если соответствующие светодиод горит, то это означает, что напряжение имеется, т. е. ко входу приложен управляющий сигнал и вход включен. |
| Светодиоды состояния выходов | Состояния выходов (т. е. включен выход или выключен) показываются с помощью светодиодов. При этом выходы контроллера могут коммутировать различные напряжения в зависимости от их типа. |
| Светодиоды индикации рабочего состояния | Светодиоды "RUN", "POWER" и "ERROR" показывают текущее рабочее состояние программируемого контроллера. "POWER" означает наличие напряжения питания, "RUN" означает выполнение программы, находящейся в памяти контроллера, а "ERROR" наличие сбоя. |
| Батарейка | При выпадении напряжения батарейка обеспечивает буферное питание внутренних запоминающих устройств типа RAM в контроллере MELSEC (только у моделей FX2N, FX2NC, FX3U и FX3UC). Батарейка служит для буферизации областей фиксации таймеров, счетчиков и маркеров. Кроме этого, она питает встроенные часы, когда питание программируемого контроллера выключено. |
| Выключатель RUN/STOP | Контроллеры MELSEC имеют два рабочих режима: "RUN" и "STOP". С помощью выключателя RUN/STOP можно переключать контроллер между этими двумя режимами. В режиме "RUN" контроллер выполняет заданную программу. В режиме "STOP" выполнение программы не происходит. В этом режиме контроллер можно программировать. |

3 Основы программирования

Программа состоит из череды отдельных управляющих команд, определяющих работу контроллера и выполняемых контроллером в запрограммированной последовательности. Поэтому при программировании сам процесс управления требуется разложить на отдельные команды. Управляющая команда является самой маленькой единицей прикладной программы контроллера.

3.1 Структура управляющей команды

Управляющая команда состоит из инструкции (команды) и операнда (в случае прикладных инструкций операндов может быть несколько). Некоторые управляющие команды могут обходиться и без операндов. Эти команды управляют выполнением программы в контроллере.

При программировании каждой управляющей команде автоматически присваивается номер шага, в результате чего однозначно устанавливается ее местонахождение в программе, так как одинаковые команды с одним и тем же операндом могут использоваться в программе многократно.

Изображение команды на языке релейных диаграмм (слева) и языке инструкций (справа):



Команда указывает, что следует сделать, т. е. какую функцию должен выполнить контроллер. Операнд указывает, с чем требуется это сделать. Обозначение операнда состоит из буквенного признака типа и адреса.



Примеры буквенных обозначений операндов:

| Буквенное обозначение операнда | Тип | Значение |
|--------------------------------|----------------|--|
| X | вход | Входная клемма программируемого контроллера (например, выключатель) |
| Y | выход | Выходная клемма программируемого контроллера (например, контактор или лампа) |
| M | маркер | Промежуточная память в контроллере, способная принимать два состояния ("Вкл." или "Выкл.") |
| T | таймер | "Реле времени" для реализации функций, зависящих от времени |
| C | счетчик | Счетчик |
| D | регистр данных | Память в контроллере, в которой можно хранить, например, измеренные значения или результаты вычислений |

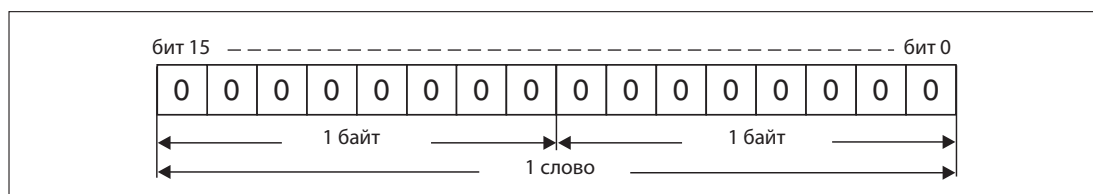
Операнды подробно описаны в разделе 4.

Так как, например, входов можно быть несколько, указание адреса операнда позволяет идентифицировать определенный вход.

3.2 Биты, байты и слова

Самой мелкой единицей информации в программируемом контроллере (и в цифровой технике вообще) является "бит". Бит может принимать только два состояния: "0" (выключено или ложно) и "1" (включено или истинно). В контроллере биты встречаются, например, в виде входов, выходов и маркеров - так называемых битовых операндов.

8 битов образуют байт, а два байта образуют слово. В контроллере MELSEC семейства FX словными операндам являются, например, регистры данных.



Благодаря 16-битному размеру в каждом регистре можно сохранять значения в диапазоне от -32768 до 32767 (см. раздел 3.3). Если этого не достаточно, два слова можно объединить в двойное 32-битное слово, в котором можно сохранять значения от -2 147 483 648 до 2 147 483 647. Эта возможность используется, например, для счетчиков.

3.3 Системы счисления

В контроллере MELSEC семейства FX используются различные системы счисления. Они служат для ввода или индикации значений и для указания адресов операндов.

Десятичные числа

С десятичными числами мы имеем дело в повседневной жизни. Их основой является число "10", т. е. после отсчета до 9 дальнейший счет переносится в следующий десятичный разряд (9>10, 19>20, 29>30 и т. д.).

- основание: 10
- цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

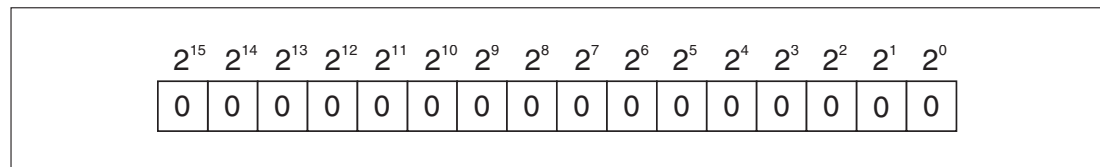
Десятичными числами в программируемых контроллерах MELSEC семейства FX указываются константы и заданные значения таймеров и счетчиков. Кроме того, в десятичном формате указываются адреса операндов (кроме входов и выходов).

Двоичные числа (двоичная система счисления)

Как и любой компьютер, программируемый контроллер оперирует только информацией типа ВКЛ./ВЫКЛ. или 0/1, содержащейся в отдельных битах (т. е. бинарной или двоичной информацией). При вводе или индикации чисел в других форматах среда программирования автоматически конвертирует различные системы счисления.

- основание: 2
- цифры: 0 и 1

Если двоичные числа сохраняются в виде слова, отдельным битам присваивается определенная значимость:



| Представление по основанию 2 | Десятичное значение | Представление по основанию 2 | Десятичное значение |
|------------------------------|---------------------|------------------------------|---------------------|
| 2 | 1 | 2 | 256 |
| 2 | 2 | 2 | 512 |
| 2 | 4 | 2 | 1024 |
| 2 | 8 | 2 | 2048 |
| 2 | 16 | 2 | 4096 |
| 2 | 32 | 2 | 8192 |
| 2 | 64 | 2 | 16384 |
| 2 | 128 | 2 | 32768* |

* В двоичных значениях бит 15 применяется для указания арифметического знака. (бит 15 = 0 означает положительное значение, а бит 15 = 1 отрицательное значение)

Чтобы преобразовать двоичное число в десятичное, следует преобразовать в десятичное значение биты, имеющие значение "1", в соответствии с их значимостью, а затем сложить эти отдельные значения.

например 00000010 00011001 (двоичное число)

$$\begin{aligned} 00000010\ 00011001\ (\text{двоичное число}) &= 1 \times 2^9 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0 \\ 00000010\ 00011001\ (\text{двоичное число}) &= 512 + 16 + 8 + 1 \\ 00000010\ 00011001\ (\text{двоичное число}) &= 537\ (\text{десятичное число}) \end{aligned}$$

Шестнадцатиричная система счисления

Шестнадцатиричные числа можно легко получить из двоичных. Поэтому они часто используются в цифровой технике и программируемых контроллерах. В контроллерах MELSEC семейства FX шестнадцатиричные числа используются для указания констант. В руководстве по программированию и справочниках по модулям шестнадцатиричные числа всегда помечаются дополнительной буквой "H", чтобы их было невозможно перепутать с десятичными (например, 12345H)

- основание: 16
- цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (буквы A, B, C, D, E и F соответствуют десятичным значениям 10, 11, 12, 13, 14 и 15)

В шестнадцатиричной системе после отсчета до Fh дальнейший счет продолжается путем переноса в следующий разряд (Fh → 10H, 1Fh → 20H, 2Fh → 30H). Каждый разряд имеет значимость с основанием 16.

| | | | | | | | | | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-----------|---------------------------|-------------------------|--|
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 16 = 1 | (в этом примере: 15 x 1 | = 15) | |
| | | | | | | | | | | | | | | | | 16 = 16 | (в этом примере: 7 x 16 | = 112) | |
| | | | | | | | | | | | | | | | | 16 = 256 | (в этом примере: 10 x 256 | = 2560) | |
| | | | | | | | | | | | | | | | | 16 = 4096 | (в этом примере: 1 x 4096 | = 4096) | |
| | | | | | | | | | | | | | | | | | | 6783 (десятичное число) | |

Вышеупомянутое простое преобразование двоичных чисел в шестнадцатиричные и наоборот можно продемонстрировать на следующем примере:

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|----|---|---|---|---|---|---|---|-------------------|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | двоичное |
| | | | | | | | | | | | | | | | | |
| 15 | | | | 5 | | | | 11 | | | | 9 | | | | десятичное* |
| | | | | | | | | | | | | | | | | |
| F | | | | 5 | | | | B | | | | 9 | | | | шестнадцатиричное |

* В десятичные значения преобразование осуществляется группами по 4 бита. Полученное таким образом десятичное число не соответствует значению полного 16-битного двоичного числа!

Восьмеричная система счисления

У базовых блоков семейства FX не имеется, например, входов X8 и X9, а также выходов Y8 и Y9. Причина заключается в том, что входы и выходы контроллеров MELSEC пронумерованы в восьмеричной системе счисления. Так как при этом в качестве основания используется "8", числа 8 и 9 не существуют. После отсчета до 8 дальнейший счет переносится в следующий разряд (от 0 до 7, от 10 до 17 от 70 до 77, от 100 до 107 и т. д.).

- основание: 8
- цифры: 0, 1, 2, 3, 4, 5, 6, 7

Обобщение

В следующей таблице еще раз сопоставлены вышеописанные четыре системы счисления:

| Десятичное число | Восьмеричное число | Шестнадцатиричное число | Двоичное число |
|------------------|--------------------|-------------------------|---------------------|
| 0 | 0 | 0 | 0000 0000 0000 0000 |
| 1 | 1 | 1 | 0000 0000 0000 0001 |
| 2 | 2 | 2 | 0000 0000 0000 0010 |
| 3 | 3 | 3 | 0000 0000 0000 0011 |
| 4 | 4 | 4 | 0000 0000 0000 0100 |
| 5 | 5 | 5 | 0000 0000 0000 0101 |
| 6 | 6 | 6 | 0000 0000 0000 0110 |
| 7 | 7 | 7 | 0000 0000 0000 0111 |
| 8 | 10 | 8 | 0000 0000 0000 1000 |
| 9 | 11 | 9 | 0000 0000 0000 1001 |
| 10 | 12 | A | 0000 0000 0000 1010 |
| 11 | 13 | B | 0000 0000 0000 1011 |
| 12 | 14 | C | 0000 0000 0000 1100 |
| 13 | 15 | D | 0000 0000 0000 1101 |
| 14 | 16 | E | 0000 0000 0000 1110 |
| 15 | 17 | F | 0000 0000 0000 1111 |
| 16 | 20 | 10 | 0000 0000 0001 0000 |
| : | : | : | : |
| 99 | 143 | 63 | 0000 0000 0110 0011 |
| : | : | : | : |

3.4 Базовый набор команд

Команды контроллеров MELSEC семейства FX можно подразделить на базовый набор команд и так называемые прикладные команды.

Функции команд базового набора можно сравнить с функциями обычных электросхем, зависящими от электрического монтажа. Базовый набор команд способен выполнять все контроллеры MELSEC семейства FX, а в отношении прикладных команд необходимо учитывать ограничения (см. раздел 5).

Обзорный перечень базового набора команд

| Команда | Значение | Описание | Ссылка |
|-------------|--|---|---------------|
| LD | Загрузить | Начало логической операции с опросом на состояние сигнала "1" | раздел 3.4.1 |
| LDI | Загрузить инверсно | Начало логической операции с опросом на состояние сигнала "0" | |
| OUT | Команда вывода | Присвоение результата логической операции | раздел 3.4.2 |
| AND | И | Операция AND с опросом на состояние сигнала "1" | раздел 3.4.4 |
| ANI | И-Не | Операция AND с опросом на состояние сигнала "0" | |
| OR | ИЛИ | Операция OR с опросом на состояние сигнала "1" | раздел 3.4.5 |
| ORI | ИЛИ-Не | Операция OR с опросом на состояние сигнала "0" | |
| ANB | И-блок | Последовательное соединение параллельных логических связей | раздел 3.4.6 |
| ORB | ИЛИ-блок | Параллельное соединение последовательных логических связей | |
| LDP | Логические операции в зависимости от фронта сигнала | Команда загрузки при восходящем фронте операнда | раздел 3.4.7 |
| LDF | | Команда загрузки при нисходящем фронте операнда | |
| ANDP | | Операция AND при восходящем фронте операнда | |
| ANDF | | Операция AND при нисходящем фронте операнда | |
| ORP | | Операция OR при восходящем фронте операнда | |
| ORF | | Операция OR при нисходящем фронте операнда | |
| SET | Установка операнда | Присвоение состояния, которое сохраняется и после того, как условие входа более не выполняется. | раздел 3.4.8 |
| RST | Сброс операнда | | |
| MPS | Сохранение, считывание и стирание промежуточных результатов при логических операциях | Сохранение результата логической операции | раздел 3.4.9 |
| MRD | | Считывание сохраненного результата логической операции | |
| MPP | | Считывание и стирание сохраненного результата логической операции | |
| PLS | Выработка импульса | Установка операнда* на время одного программного цикла при восходящем фронте условия входа | раздел 3.4.10 |
| PLF | | Установка операнда* на время одного программного цикла при нисходящем фронте условия входа | |
| MC | Главный выключатель | Активация обработки частей программы | раздел 3.4.11 |
| MCR | Сброс главного выключателя | | |
| INV | Инверсия | | |

3.4.1 Начало логических операций

| Команда | Значение | Символ | GX Developer FX |
|------------|---|--------|-----------------|
| LD | Команда загрузки, начало логической операции с опросом на состояние сигнала "1" | | |
| LDI | Команда загрузки, начало логической операции с опросом на состояние сигнала "0" | | |

Цепь тока всегда начинается с команды LD или LDI. В качестве операндов можно указывать входы, маркеры, датчики времени и счетчики.

Примеры применения этих команд вы найдете в следующем разделе при описании команды OUT.

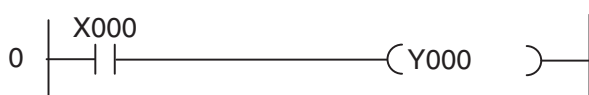
3.4.2 Вывод или присвоение результата логической операции

| Команда | Значение | Символ | GX Developer FX |
|------------|---|--------|-----------------|
| OUT | Команда вывода, присвоение результата логической операции | | |

С помощью команды OUT можно завершить цепь тока. В качестве результата логической операции можно также запрограммировать несколько команд OUT. Результат логической операции, присвоенный операнду с помощью команды OUT, можно использовать в следующих шагах программы в качестве состояния входного сигнала.

Пример (команды LD и OUT)

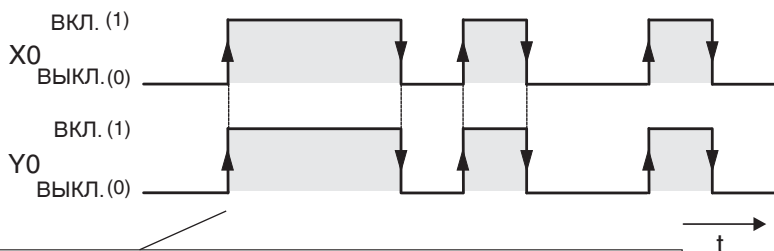
Релейная диаграмма



Список инструкций

```
0 LD X000
1 OUT Y000
```

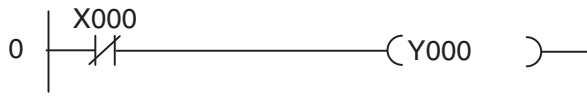
С помощью этих двух команд образуется следующая диаграмма изменения сигналов:



Условие команды LD (опрос на состояние "1") выполнено, поэтому результат логической операции также соответствует "1" и выход включается.

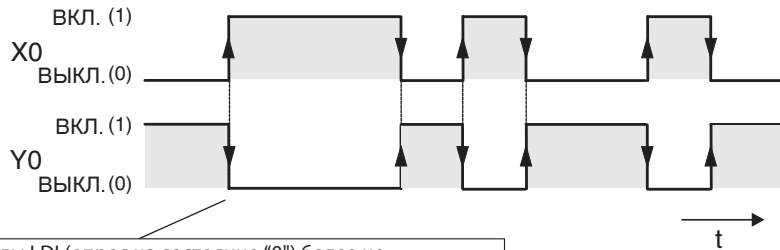
Пример (команды LDI и OUT)

Релейная диаграмма



Список инструкций

```
0 LDI X000
1 OUT Y000
```

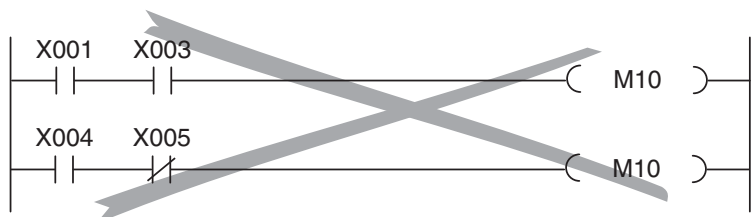


Условие команды LDI (опрос на состояние "0") более не выполняется, выход выключается.

Двойное присвоение маркеров или выходов

Присваивать результат логической операции операнду следует только в одном месте программы.

По мере отработки программы "сверху вниз" первое присвоение для M10 перезаписывается вторым присвоением.



Видоизменив эту часть программы, можно учесть все входные логические связи.



3.4.3 Учет датчиков

Прежде чем описывать прочие команды, рассмотрим вкратце действие сигналов датчиков.

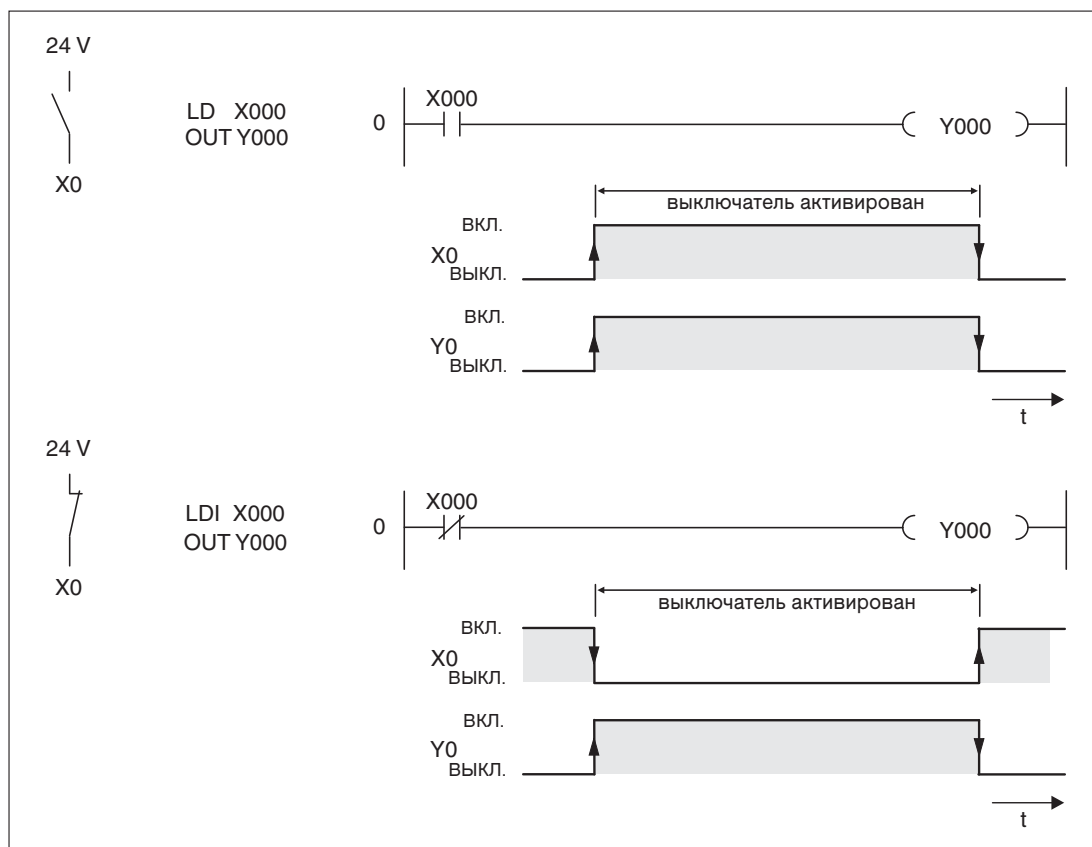
Чтобы получить требуемую функцию при программировании контроллера, необходимо учитывать принцип действия выключателей, кнопок и датчиков. Управляющая команда проверяет только состояние сигнала указанного входа (независимо от того, каким способом управляется этот вход).

| | | |
|--|---------------------|---|
| | Замыкающий контакт | При активации замыкающего контакта вход включается (состояние сигнала "1"). |
| | Размыкающий контакт | При активации размыкающего контакта вход выключается (состояние сигнала "0"). |



Поэтому уже при программировании должно быть известно, что представляет собой датчик, подключенный ко входу контроллера – размыкающий или замыкающий контакт. Вход, к которому подключен замыкающий контакт, должен обрабатываться иначе, чем вход, к которому подключен размыкающий контакт. Это можно наглядно продемонстрировать на следующем

В большинстве случаев используются датчики с замыкающими контактами. Однако по соображениям безопасности в некоторых случаях (например, для выключения приводов) применяются размыкающие контакты (см. раздел 3.5).

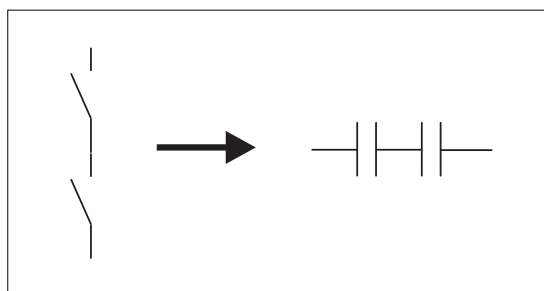
На рисунке ниже изображены два фрагмента программы, при которых несмотря на использование различных датчиков достигается одинаковый результат: при воздействии на выключатель выход включается.



3.4.4 Операции AND

| Команда | Значение | Символ | GX Developer FX |
|---------|--|--------|---|
| AND | И (операция AND с опросом на состояние сигнала "1") | |  |
| ANI | И-Не (операция AND с опросом на состояние сигнала "0") | |  |

Операция AND соответствует последовательному соединению нескольких выключателей (по меньшей мере двух). Только если все контакты замкнуты, ток течет. Если один или несколько контактов разомкнуты, функция AND не выполняется и ток не течет.

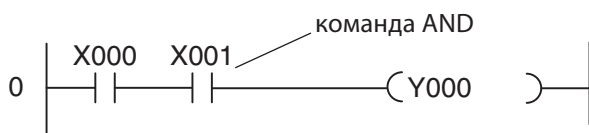


Между прочим, в среде программирования для команд AND и ANI используются те же экранные кнопки и функциональные клавиши, что и для команд LD и LDI. При программировании на языке релейных диаграмм программное обеспечение автоматически присваивает команды в зависимости от места их ввода.

Если вы программируете на языке инструкций, учитывайте, что команды AND и ANI нельзя программировать в начале цепи тока. Начало логической связи программируется с использованием команд LD или LDI (раздел 3.4.1).

Пример команды AND

Релейная диаграмма

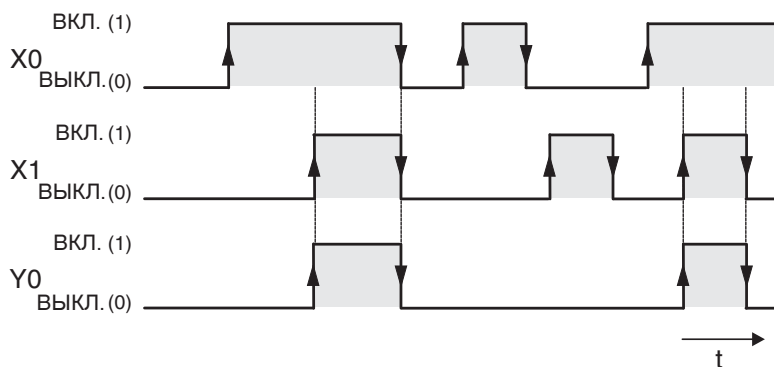


Список инструкций

```

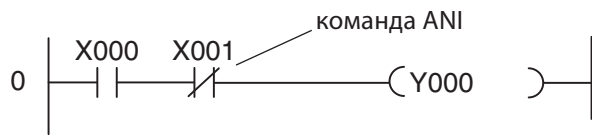
0 LD X000
1 AND X001
2 OUT Y000
    
```

Выход Y0 включается только в том случае, если X0 и X1 включены:



Пример команды ANI

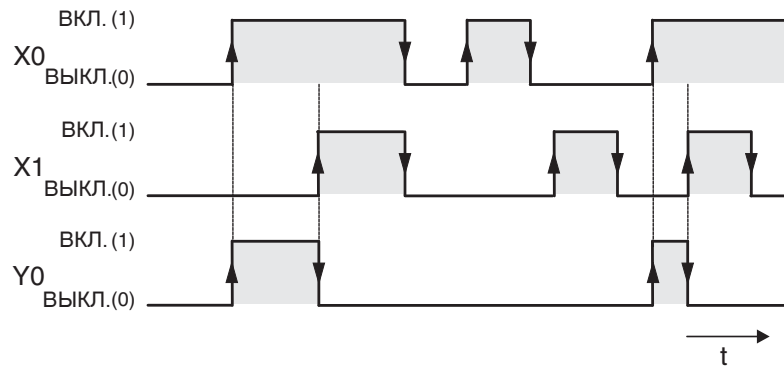
Релейная диаграмма



Список инструкций

| | | |
|---|-----|------|
| 0 | LD | X000 |
| 1 | ANI | X001 |
| 2 | OUT | Y000 |

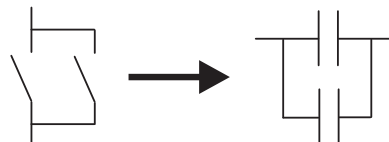
Выход Y0 включается только в том случае, если X0 включен и X1 выключен:



3.4.5 Операции OR

| Команда | Значение | Символ | GX Developer FX |
|---------|---|--------|-----------------|
| OR | ИЛИ (операция OR с опросом на состояние сигнала "1") | | |
| ORI | ИЛИ-Не (операция OR с опросом на состояние сигнала "0") | | |

Операция OR в коммутационной технике соответствует параллельному соединению нескольких выключателей. Если один из контактов замкнут, ток течет. Только если ни один из контактов не замкнут, ток не течет.



Пример команды OR

Релейная диаграмма

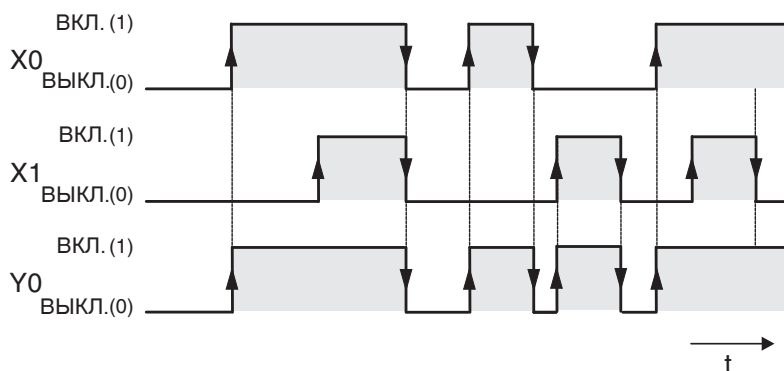


Список инструкций

```

0 LD X000
1 OR X001
2 OUT Y000
    
```

В этом примере выход Y0 включается, если включен X0 или X1:



Пример команды ORI

Релейная диаграмма

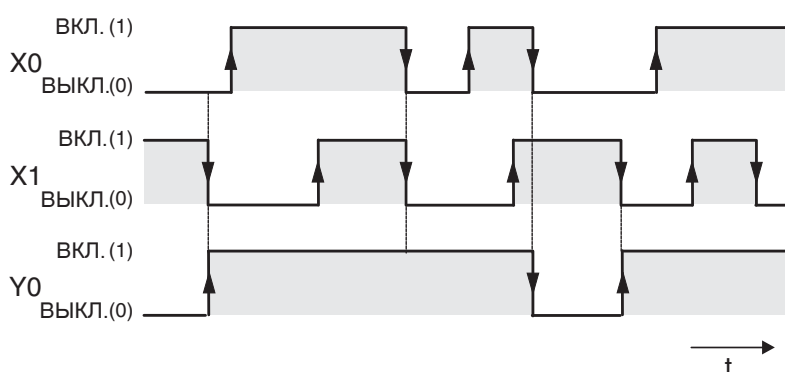


Список инструкций

```

0 LD X000
1 ORI X001
2 OUT Y000
    
```

Выход Y0 включен, если включен X0 или выключен X1:



3.4.6 Команды соединения логических операций

| Команда | Значение | Символ | GX Developer FX |
|------------|---|--------|-----------------|
| ANB | И-блок (последовательное соединение параллельных логических связей) | — | |
| ORB | ИЛИ-блок (параллельное соединение последовательных логических связей) | | |

Хотя для контроллера команды ANB и ORB являются командами, при программировании на языке релейных диаграмм они показываются только в виде соединительных линий. Эти команды появляются лишь при отображении или программировании программы на языке инструкций. В этом случае их требуется вводить с использованием сокращений ANB или ORB.

Обе команды обходятся без операндов и их можно использовать в программе сколь угодно часто. Однако количество команд LD и LDI и, тем самым, количество команд ORB или ANB перед командами вывода ограничено восемью.

Пример команды ANB

Релейная диаграмма



Список инструкций

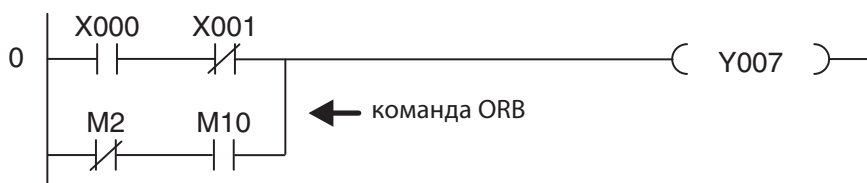
```

0 LD X000
1 ORI M2 ← 1-е параллельное соединение (операция OR)
2 LDI X001
3 OR M10 ← 2-е параллельное соединение (операция OR)
4 ANB ← Команда ANB соединяет между собой оба соединения OR.
5 OUT Y007
    
```

В этом примере выход Y07 включается, если вход X00 имеет состояние "1" или маркер M2 имеет состояние "0", и при этом вход X01 имеет состояние "0" или маркер M10 имеет состояние "1".

Пример команды ORB

Релейная диаграмма



Список инструкций

```

0 LD X000
1 ANI X001 ← 1-е последовательное соединение (операция AND)
2 LDI M2
3 AND M10 ← 2-е последовательное соединение (операция AND)
4 ORB ← Команда ORB соединяет между собой оба соединения AND.
5 OUT Y007
    
```

Выход Y07 включается, если вход X00 имеет состояние "1" и вход X01 имеет состояние "0", или если маркер M2 имеет состояние "0" и маркер M10 имеет состояние "1".

3.4.7 Выполнение логических операций в зависимости от фронта сигнала

| Команда | Значение | Символ | GX Developer FX |
|---------|---|--------|-----------------|
| LDP | Команда загрузки при восходящем фронте операнда | | |
| LDF | Команда загрузки при нисходящем фронте операнда | | |
| ANDP | Операция AND при восходящем фронте операнда | | |
| ANDF | Операция AND при нисходящем фронте операнда | | |
| ORP | Операция OR при восходящем фронте операнда | | |
| ORF | Операция OR при нисходящем фронте операнда | | |

В программе контроллера часто требуется распознавать и оценивать характер изменения сигнала операндов, т. е. нарастающий или нисходящий фронт. При нарастающем фронте состояние сигнала изменяется с "0" на "1", а при ниспадающем - с "1" на "0".

Логические операции, реагирующие на фронт, поставляют сигнал "1" только в том программном цикле, в котором опрошенный операнд изменил состояние своего сигнала.

Если фронт сигнала не оценивается, то (например) выключатель, срабатывающий от проезжающих мимо него по конвейеру упаковок и предназначенный для подсчета упаковок, поставляет неправильный результат, так как в этом случае состояние счетчика повышается на "1" в каждом программном цикле – до тех пор, пока выключатель остается активированным. Если, однако, контролируется только нарастающий фронт входа, то каждая упаковка повышает значение счета только один раз.

Между прочим, большинство прикладных команд также могут учитывать фронт сигнала (см. раздел 5).

Контроль возрастающего фронта

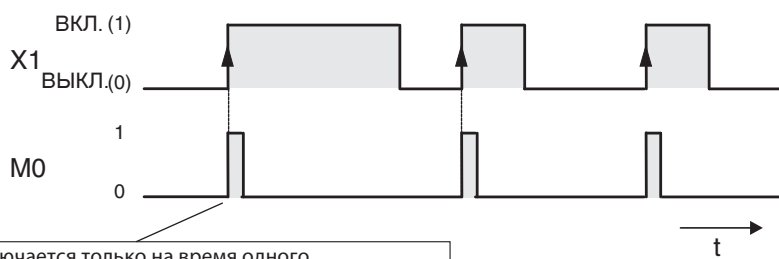
Релейная диаграмма

Список инструкций



```

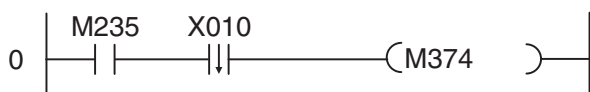
0 LDP X001
1 OUT M0
    
```



Маркер M0 включается только на время одного программного цикла.

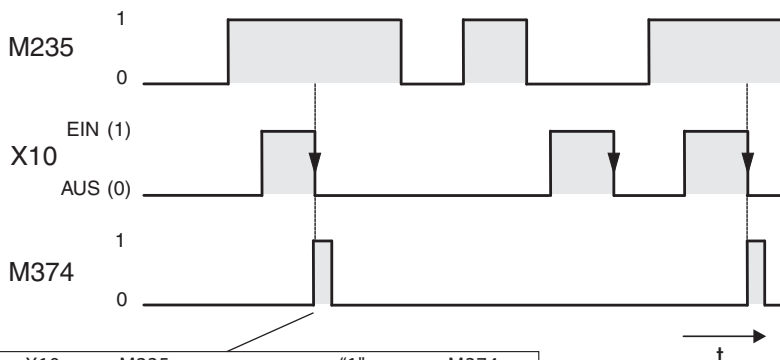
Контроль ниспадающего фронта

Релейная диаграмма



Список инструкций

- 0 LD M235
- 1 ANDF X010
- 2 OUT M374



При выключении X10, если M235 имеет состояние "1", маркер M374 включается на время одного программного цикла.

За исключением оценки фронта сигнала, действие команд LDP и LDF, ANDP и ANDF, а также ORP и ORF идентично действию команд LD, AND и OR, т. е. команды, управляемые фронтом сигнала, можно использовать в программе точно так же, как "обычные" команды.

3.4.8 Установка и сброс

| Команда | Значение | Символ | GX Developer FX |
|---------|--|--------|-----------------|
| SET | Установка операнда(1) (присвоение состояния "1") | | |
| RST | Сброс операнда(2) (присвоение состояния "0") | | |

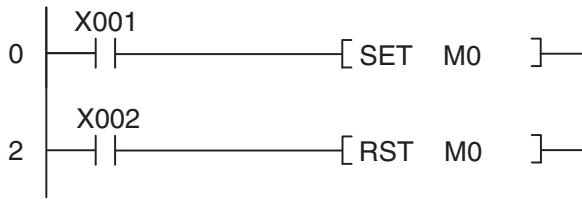
- ① С помощью команды SET можно устанавливать выходы (Y), маркеры (M) и шаговые маркеры (S).
- ② С помощью команды RST можно сбрасывать выходы (Y), маркеры (M), шаговые маркеры (S), таймеры (T), счетчики (C) и регистры (D, V, Z).

Сигнал команды OUT только до тех пор имеет состояние "1", пока результат логической операции перед командой OUT имеет состояние "1". Если, например, ко входу подключена кнопка, а к выходу лампа, то при комбинации команд LD и OUT лампа горит только до тех пор, пока кнопка удерживается нажатой.

С помощью команды SET выход или маркер включается (= устанавливается) после короткого импульса включения. После этого этот операнд остается включенным до тех пор, пока он не будет снова выключен командой RST (= сброшен). Таким способом можно реализовать, например, функцию самоудержания или включение/выключение приводов с помощью кнопок. (Выход выключается также в том случае, если останавливается сам контроллер или выключается напряжение питания. Некоторые маркеры и в этих случаях сохраняют свое последнее сигнальное состояние, т. е., например, остаются установленными.)

Для ввода команды SET или RST в релейную диаграмму щелкните в перечне инструментов GX Developer по показанному выше символу или нажмите клавишу F8. После этого введите команду и операнд, например, SET Y1.

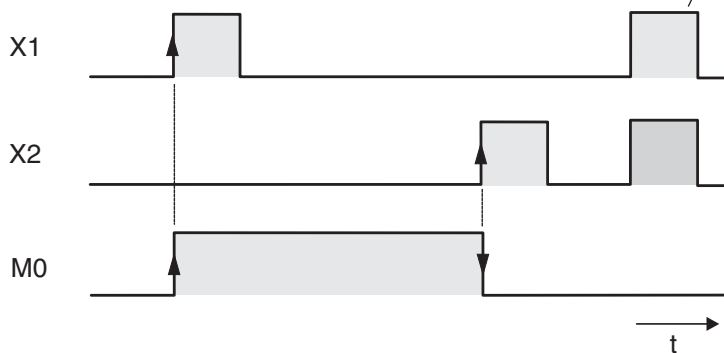
Релейная диаграмма



Список инструкций

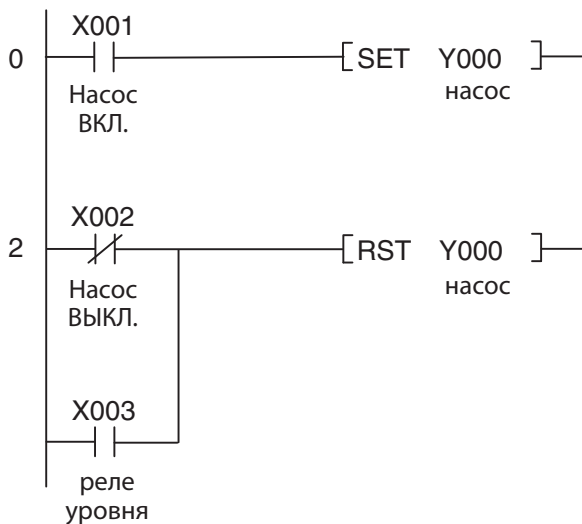
- | | | |
|---|-----|------|
| 0 | LD | X001 |
| 1 | SET | M0 |
| 2 | LD | X002 |
| 3 | RST | M0 |

Если в одном и том же цикле команда установки и сброса операнда имеют состояние "1", то преимущество имеет последняя по очередности команда. В этом примере это команда RST, т. е. M0 не



В качестве примера применения изображено управление насосом для заполнения резервуара. Насосом можно вручную управлять с помощью клавиш "ВКЛ." и "ВЫКЛ.". По соображениям безопасности, для выключения применяется кнопка с размыкающим контактом. Если резервуар заполнен, реле уровня выключает насос.

Релейная диаграмма



Список инструкций

- | | | |
|---|-----|------|
| 0 | LD | X001 |
| 1 | SET | Y000 |
| 2 | LDI | X002 |
| 3 | OR | X003 |
| 4 | RST | Y000 |

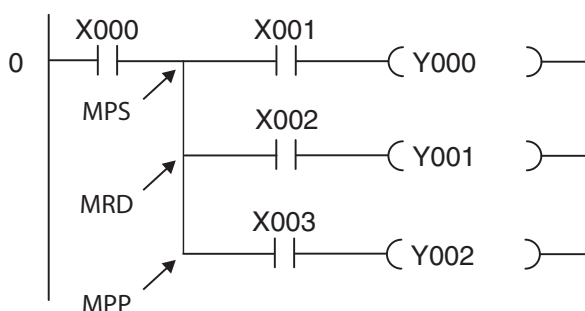
3.4.9 Сохранение, считывание и стирание результата логической операции

| Команда | Значение | Символ | GX Developer FX |
|------------|---|--------|-----------------|
| MPS | Сохранение результата логической операции | — | — |
| MRD | Считывание сохраненного результата логической операции | — | — |
| MPP | Считывание и стирание сохраненного результата логической операции | — | — |

С помощью команд MPS, MRD и MPP можно сохранять, вызывать и стирать (промежуточные) результаты логических операций. С помощью этих команд можно создавать логические уровни и, тем самым, придавать программе хорошую обзорность.

При вводе программы на языке релейных диаграмм эти команды автоматически встраиваются средой программирования. Лишь при отображении или программировании программы на языке инструкций команды MPS, MRD и MPP показываются или требуют ручного ввода.

Релейная диаграмма

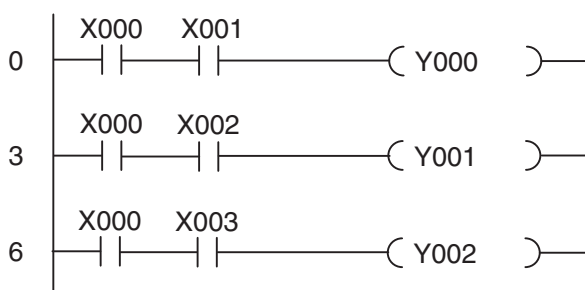


Список инструкций

| | | |
|---|-----|------|
| 0 | LD | X000 |
| 1 | MPS | |
| 2 | AND | X001 |
| 3 | OUT | Y000 |
| 4 | MRD | |
| 5 | AND | X002 |
| 6 | OUT | Y001 |
| 7 | MPP | |
| 8 | AND | X003 |
| 9 | OUT | Y002 |

Для лучшего понимания изображенного вверху фрагмента программы тот же пример показан еще раз с иным программированием.

Релейная диаграмма



Список инструкций

| | | |
|---|-----|------|
| 0 | LD | X000 |
| 1 | AND | X001 |
| 2 | OUT | Y000 |
| 3 | LD | X000 |
| 4 | AND | X002 |
| 5 | OUT | Y001 |
| 6 | LD | X000 |
| 7 | AND | X003 |
| 8 | OUT | Y002 |

Операнды (в примере: X0) пришлось бы программировать повторно. В результате увеличиваются затраты на программирование, в частности, образуются более длинные программы и более обширные цепи токов.

Для последней команды вывода вместо MRD следует применять MPP, чтобы стереть память логических операций. Можно также использовать несколько команд MPS. Таким способом можно образовать до 11 логических уровней. Прочие примеры команд MPS, MPP и MRS вы найдете в руководстве по программированию контроллеров семейства FX, артикул 136748.

3.4.10 Выработка импульса

| Команда | Значение | Символ | GX Developer FX |
|---------|--|--------|-----------------|
| PLS | Установка операнда* на время одного программного цикла при восходящем фронте условия входа | | |
| PLF | Установка операнда* на время одного программного цикла при нисходящем фронте условия входа | | |

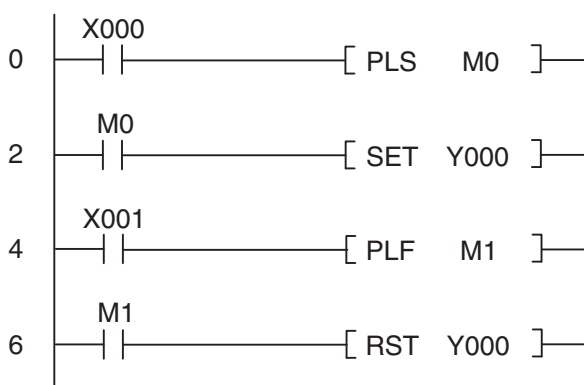
* С помощью команд PLS и PLF можно управлять выходами (Y) и маркерами (M).

Если вместо команды OUT используется команда PLS, указанный операнд имеет сигнальное состояние "1" только в том программном цикле, в котором состояние сигнала логических операций перед командой PLS изменяется с "0" на "1" (нарастающий фронт).

Команда PLF срабатывает при ниспадающем фронте и поставяет состояние сигнала "1" на время одного программного цикла, если состояние сигнала логических операций перед этой командой изменяется с "1" на "0".

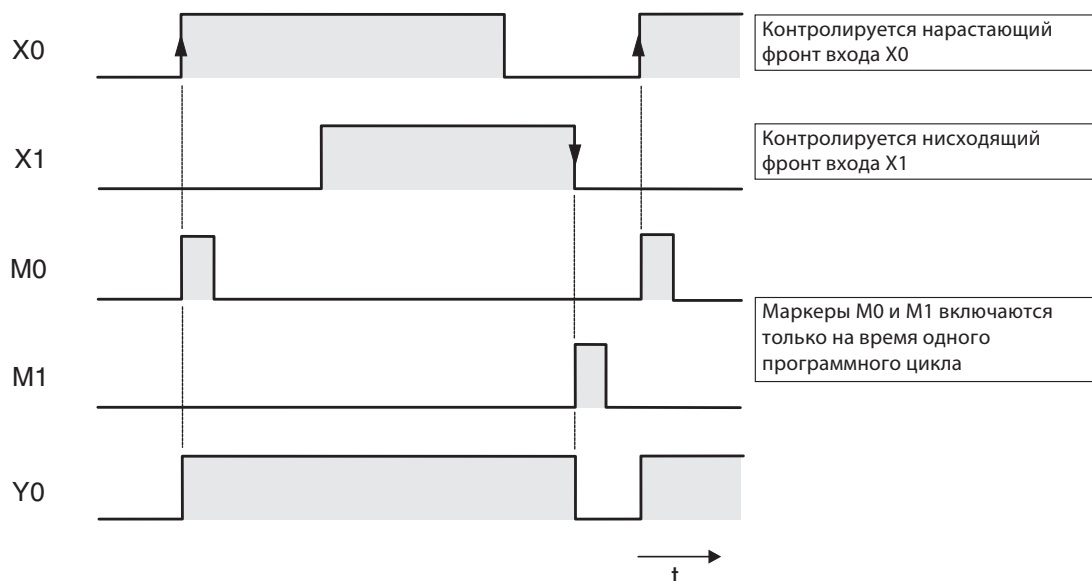
Для ввода команд PLS или PLF на языке релейных диаграмм следует в перечне инструментов GX Developer щелкнуть по изображенному сверху символу или нажать клавишу F8. Затем вводится команда и операнд, например, PLS Y2.

Релейная диаграмма



Список инструкций

| | | |
|---|-----|------|
| 0 | LD | X000 |
| 1 | PLS | M0 |
| 2 | LD | M0 |
| 3 | SET | Y000 |
| 4 | LD | X001 |
| 5 | PLF | M1 |
| 6 | LD | M1 |
| 7 | RST | Y000 |



3.4.11 Функция главного выключателя (команды MC и MCR)

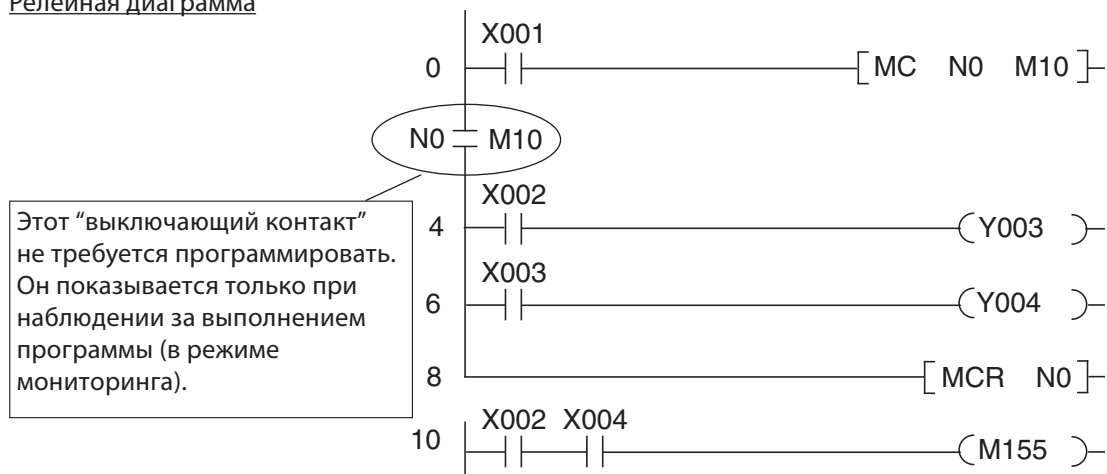
| Команда | Значение | Символ | GX Developer FX |
|---------|---|--------|-----------------|
| MC | Главный выключатель, установка контрольного условия ^① | MC n □ | |
| MCR | Сброс главного выключателя, сброс контрольного условия ^② | MCR n | |

① Операндами команды MC могут быть выходы (Y) и маркеры (M). n: от N0 до N7

② n: от N0 до N7

Путем установки (MC) или сброса (MCR) контрольного условия можно активировать или деактивировать отдельные области программы. При программировании на языке релейных диаграмм команду Master Control можно сравнить с выключателем в левой сборной шине, который должен быть замкнут, чтобы могла быть выполнена следующая часть программы.

Релейная диаграмма



Список инструкций

```

0 LD X001
1 MC N0 M10
4 LD X002
5 OUT Y003
6 LD X003
7 OUT Y004
8 MCR N0
10 LD X002
11 AND X004
12 OUT M155
    
```

В изображенном вверху примере, цепи тока между командами MC и MCR обрабатываются только в том случае, если вход X001 включен.

Какая область программы должна активироваться, устанавливается путем указания адреса разветвления программы от N0 до N7 (так называемого адреса вложения). Указание операнда Y или M определяет включающий контакт. Этот контакт активирует область программы, как только выполняется входное условие для команды MC.

Если входное условие команды MC не выполнено, то между командами MC и MCR состояния операндов изменяются следующим образом:

- Фиксируемые таймеры и счетчики, а также операнды, управляемые командами SET и RST, сохраняют свое состояние.
- Нефиксируемые таймеры и операнды, управляемые командой OUT, сбрасываются.

(Вышеназванные таймеры и счетчики описаны в следующем разделе.)

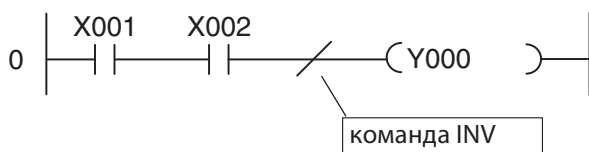
3.4.12 Инвертирование результата логической операции

| Команда | Значение | Символ | GX Developer FX |
|---------|---|--------|-----------------|
| INV | Реверсирование результата логической операции | | |

Команда INV указывается без операндов. Она обращает результат логической операции, который был действителен перед исполнением команды INV:

- Если результат логической операции был "1", после инверсии он становится равным "0".
- Если результат логической операции был "0", после инверсии он становится равным "1".

Релейная диаграмма

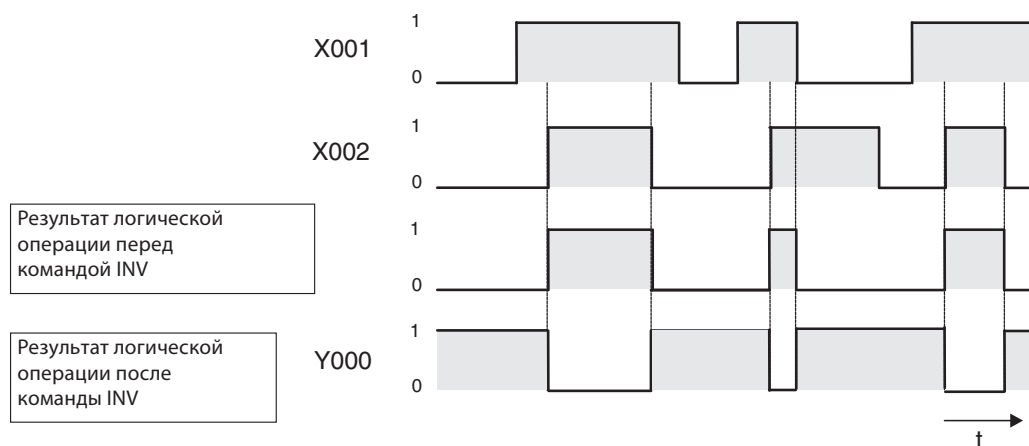


Список инструкций

```

0 LD X001
1 AND X002
2 INV
3 OUT Y000
    
```

Для показанного выше примера образуется следующая диаграмма изменения сигналов:



Команду INV можно использовать для того, чтобы изменить результат сложной логической операции на противоположный. Ее можно вводить в программе в том же месте, что и команды AND или ANI.

Команду INV нельзя вводить в начале логической операции, как команды LD, LDI, LDP или LDF.

3.5 Безопасность превыше всего!

Хотя у контроллера много преимуществ по сравнению с системами управления с неизменяемой, “монтажной” логикой, в вопросах безопасности на него нельзя полагаться полностью.

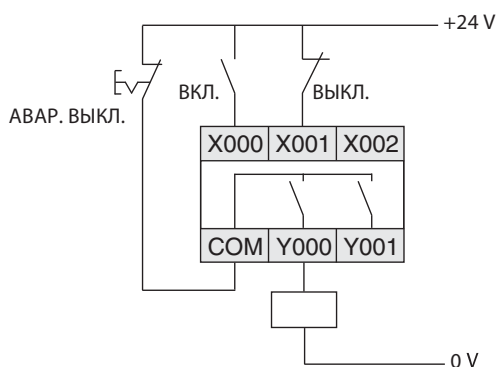
Устройства аварийного выключения

Сбой или ошибка в системе управления установки не должны подвергать опасности людей и оборудование. Поэтому в случае неправильной работы контроллера, должны действовать устройства аварийного выключения, которые, например, отключают электропитание выходов контроллера.

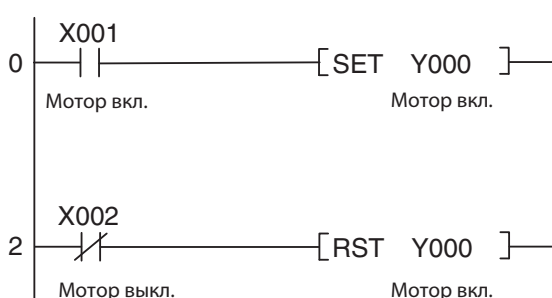
Состояние кнопки аварийного выключения ни в коем случае не должно оцениваться только как вход контроллера, и отключение не должно активироваться программой.

Безопасность должна обеспечиваться и при обрыве проводки

Эксплуатационная безопасность должна быть обеспечена и в том случае, если передача сигналов от выключателей к контроллеру прервана. По этой причине команды включения передаются на контроллер выключателями или кнопками с замыкающими контактами, а команды выключения – размыкающими контактами.



В этом примере контактор привода можно дополнительно выключить с помощью аварийного выключателя.



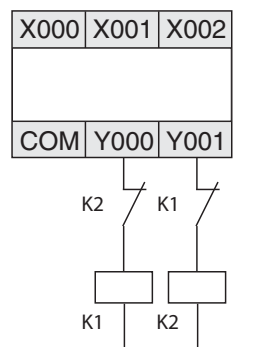
В программе замыкающий контакт кнопки “Вкл.” опрашивается с помощью команды LD, а размыкающий контакт кнопки “Выкл.” – с помощью команды LDI. Выход (и, тем самым, привод) отключается, если вход X002 имеет состояние “0”. Такое состояние он имеет при нажатии кнопки “Выкл.” или обрыве соединения между кнопкой и входом X002.

Таким образом, происходит отключение или предотвращается включение и при обрыве проводки привода. Кроме того, преимущество имеет выключение, так как в программе оно обрабатывается после включения.

Блокировочные контакты

Если в какой-либо схеме два выхода нельзя включать одновременно (например, при переключении направления вращения приводов), то эта блокировка должна осуществляться и контактами самих управляемых контакторов. В программе происходит только внутренняя блокировка, поэтому в случае неисправности контроллера оба выхода могли бы включиться одновременно.

Пример блокировки контактами контактора: контакторы K1 и K2 не могут быть включены одновременно.



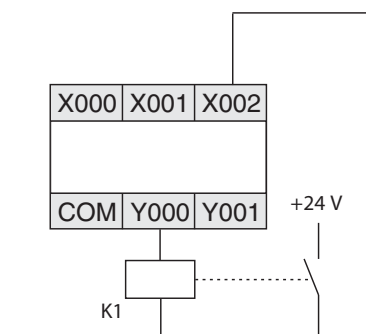
Принудительные отключения

Если контроллер управляет процессами движений и переезд через конечную точку может породить опасности, должны быть предусмотрены дополнительные конечные выключатели, которые в этом случае прерывают движение непосредственно, независимо от контроллера. Пример принудительного отключения вы найдете в разделе 3.6.2.

Обратная связь

Как правило, выходы контроллера не контролируются. Выход включается и программа предполагает, что за пределами контроллера происходит требуемая реакция. В большинстве случаев этого достаточно. Однако при чувствительных применениях, в которых неисправность в выходном контуре (например, обрыв проводки или приварившиеся контакторы) может иметь тяжелые последствия для безопасности или функционирования, следует контролировать и выдаваемые контроллером сигналы.

В этом примере замыкающий контакт контактора K1 включает вход X002, если включился выход Y000. Таким способом в программе можно контролировать, правильно ли работает этот выход и подключенный контактор. При этом не определяется, работает ли включенная нагрузка как это требуется (например, действительно ли вращается привод). Для этого необходим иной вид контроля, например, контроль силового напряжения или датчика вращения.



3.6 Реализация задач управления

Контроллер предоставляет почти неограниченные возможности логической связи входов и выходов. Из множества команд, предлагаемых контроллерами MELSEC семейства FX, следует лишь выбрать команды, подходящие для решения задачи управления, и на их основе составить программу.

Продемонстрируем весь путь от постановки задачи до готовой программы на примере двух простых задач управления.

3.6.1 Установка охранной сигнализации

Уже перед программированием должна быть ясно поставлена задача. То есть, следует начать “с конца” и описать, что контроллер должен делать:

Постановка задачи

Требуется создать установку охранной сигнализации, имеющую несколько контуров сигнализации и функции включения/выключения с задержкой времени.

- С помощью выключателя с ключом установка переводится в охранное состояние по истечении времени задержки длительностью 20 секунд. Благодаря этой задержке остается время на то, чтобы покинуть дом. В это время установка показывает, замкнуты ли сигнальные контуры.
- При обрыве сигнального контура должна срабатывать сигнализация (принцип нормально замкнутой цепи, т. е. сигнализация срабатывает и в случае саботажа). Дополнительно установка должна показывать, какой сигнальный контур вызвал сигнализацию.
- По истечении времени ожидания длительностью 10 секунд должен включаться звуковой сигнал и сигнальный фонарь. (Задержка срабатывания сигнализации нужна для того, чтобы пользователь имел возможность отключить сигнализацию, вернувшись домой. По этой причине имеется также отдельная лампа, которая показывает, находится ли установка в охранном режиме.)
- Звуковой сигнал должен звучать 30 секунд. Однако световая сигнализация должна оставаться включенной до дезактивации установки охранной сигнализации.
- Должна иметься возможность сброса сигнализации с помощью выключателя с ключом.

Определение входных и выходных сигналов

В качестве следующего шага необходимо установить, какие входные и выходные сигналы должны обрабатываться. Из описания принципа работы установки видно, что для управления установкой нужен выключатель с ключом и 4 сигнальные лампы. Кроме того, используются еще как минимум три входа для сигнальных контуров и два выхода для звукового сигнала и проблескового фонаря. В общей сложности используются 4 входа и 6 выходов. После этого сигналы сопоставляются входам и выходам контроллера:

| Функция | | Обозн. | Адрес | Примечание |
|---------|---|----------|-------|--|
| Входы | Установка в охранном режиме | S1 | X1 | Замыкающий контакт (выключатель с ключом) |
| | Сигнальный контур 1 | S11, S12 | X2 | Размыкающие контакты (сигнализация активируется, если вход имеет состояние “0”.) |
| | Сигнальный контур 2 | S21, S22 | X3 | |
| | Сигнальный контур 3 | S31, S32 | X4 | |
| Выходы | Индикация “Охранный режим” | H0 | Y0 | Функция выходов выполняется, если соответствующий выход включается. Например, если включается Y1, раздается звуковой сигнал. |
| | Звуковой сигнал (сирена) | E1 | Y1 | |
| | Световая сигнализация (круговой фонарь) | H1 | Y2 | |
| | Индикация сигнального контура 1 | H2 | Y3 | |
| | Индикация сигнального контура 2 | H3 | Y4 | |
| | Индикация сигнального контура 3 | H4 | Y5 | |

Программирование

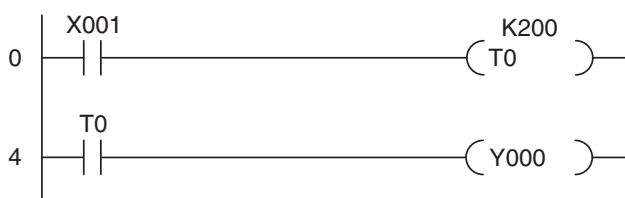
Теперь можно программировать. Нужны ли маркеры и сколько, обычно становится ясным лишь во время программирования. Заранее известно, что в этой установке сигнализации важную роль играют три устройства выдержки времени. В системе управления, основанной на физическом монтаже, применяются реле времени. В программируемом контроллере выдержка времени реализуется электронным способом (см. раздел 4.3). Эти "таймеры" можно установить уже до программирования:

| Функция | Адрес | Примечание |
|----------------|---------------------------------------|------------------------|
| Датчик времени | Задержка постановки на охранный режим | T0 время: 20 секунд |
| | Задержка срабатывания сигнализации | T1 время: 10 секунд |
| | Время работы звукового сигнала | T2 время: 30 секунд |

После этого решаются отдельные частичные задачи управления:

- Перевод установки в охранный режим с задержкой

Релейная диаграмма



Список инструкций

```

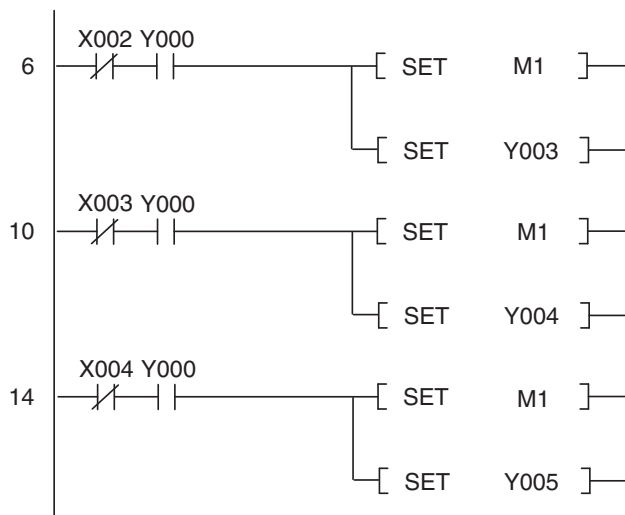
0 LD X001
1 OUT T0 K200
4 LD T0
5 OUT Y000

```

После включения выключателя с ключом датчик времени T0 отсчитывает реализованную задержку включения. По истечении 20 секунд ($K200 = 200 \times 0,1 \text{ с} = 20 \text{ с}$) подключенная к выходу Y000 контрольная лампа показывает, что установка включена.

- Контроль сигнальных контуров и распознавание сигнализации

Релейная диаграмма



Список инструкций

```

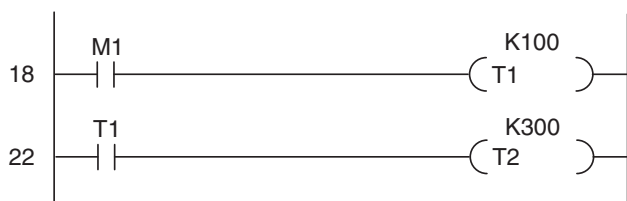
6 LDI X002
7 AND Y000
8 SET M1
9 SET Y003
10 LDI X003
11 AND Y000
12 SET M1
13 SET Y004
14 LDI X004
15 AND Y000
16 SET M1
17 SET Y005

```

В программе опрашивается также выход Y000 – чтобы определить, включена ли установка охранной сигнализации. Для этого можно было бы также использовать маркер, включающийся и выключающийся параллельно выходу Y000. Только в том случае, если установка находится в охранном режиме, при обрыве сигнального контура устанавливается маркер M1, который показывает, что сработала сигнализация. Дополнительно выходы с Y003 по Y005 показывают, какой сигнальный контур был прерван. Маркер M1 и соответствующий выход остаются включенными и после того, как сигнальный контур был снова замкнут.

● Задержка срабатывания сигнализации

Релейная диаграмма



Список инструкций

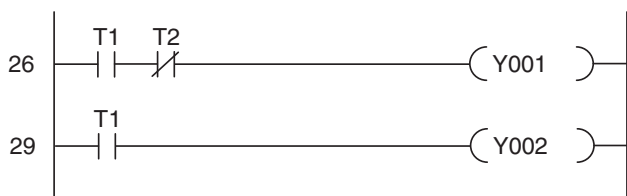
```

18 LD M1
19 OUT T1 K100
22 LD T1
23 OUT T2 K300
    
```

Если сигнализация сработала (M1 в этом случае имеет состояние "1"), начинает отсчитываться время задержки 10 секунд. По истечении этого времени T1 запускается датчик времени T2, который настроен на 30 секунд (время включенного состояния sireны).

● Проявление сигнализации (включение sireны и кругового фонаря)

Релейная диаграмма

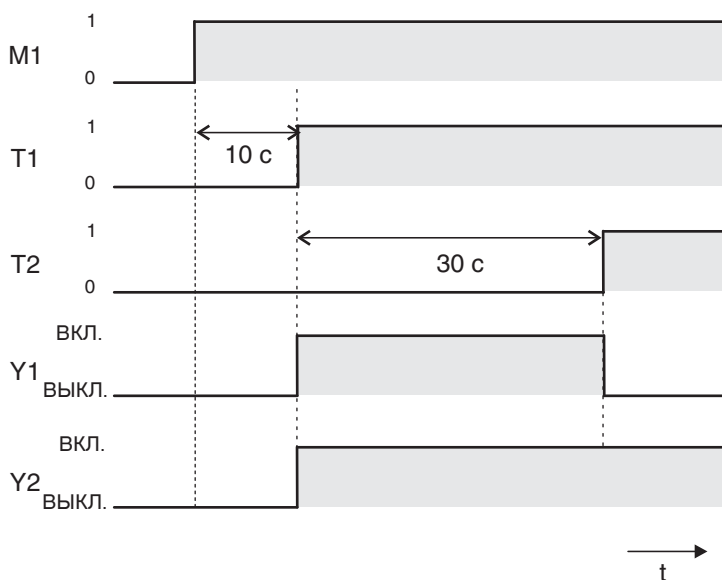


Список инструкций

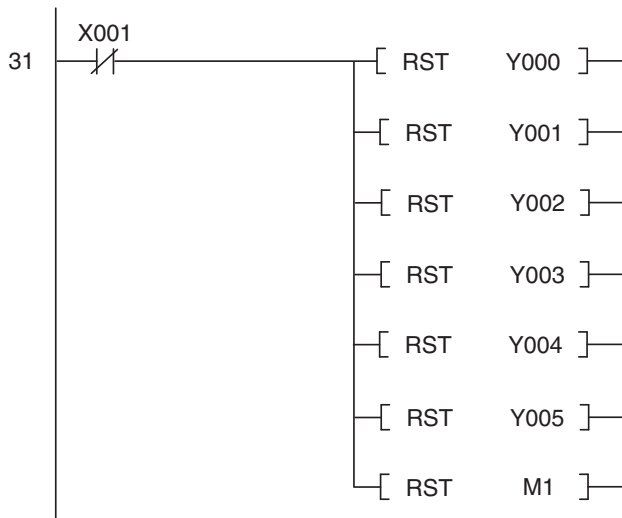
```

26 LD T1
27 ANI T2
28 OUT Y001
29 LD T1
30 OUT Y002
    
```

После истечения задержки включения длительностью 10 секунд (T1) и до тех пор, пока таймер T2 еще работает, включается sireна. По истечении 30 секунд (T2) sireна умолкает. Круговой фонарь также включается через 10 секунд. На рисунке ниже показана диаграмма изменения сигналов для этой части программы:



● Сброс всех выходов и маркеров

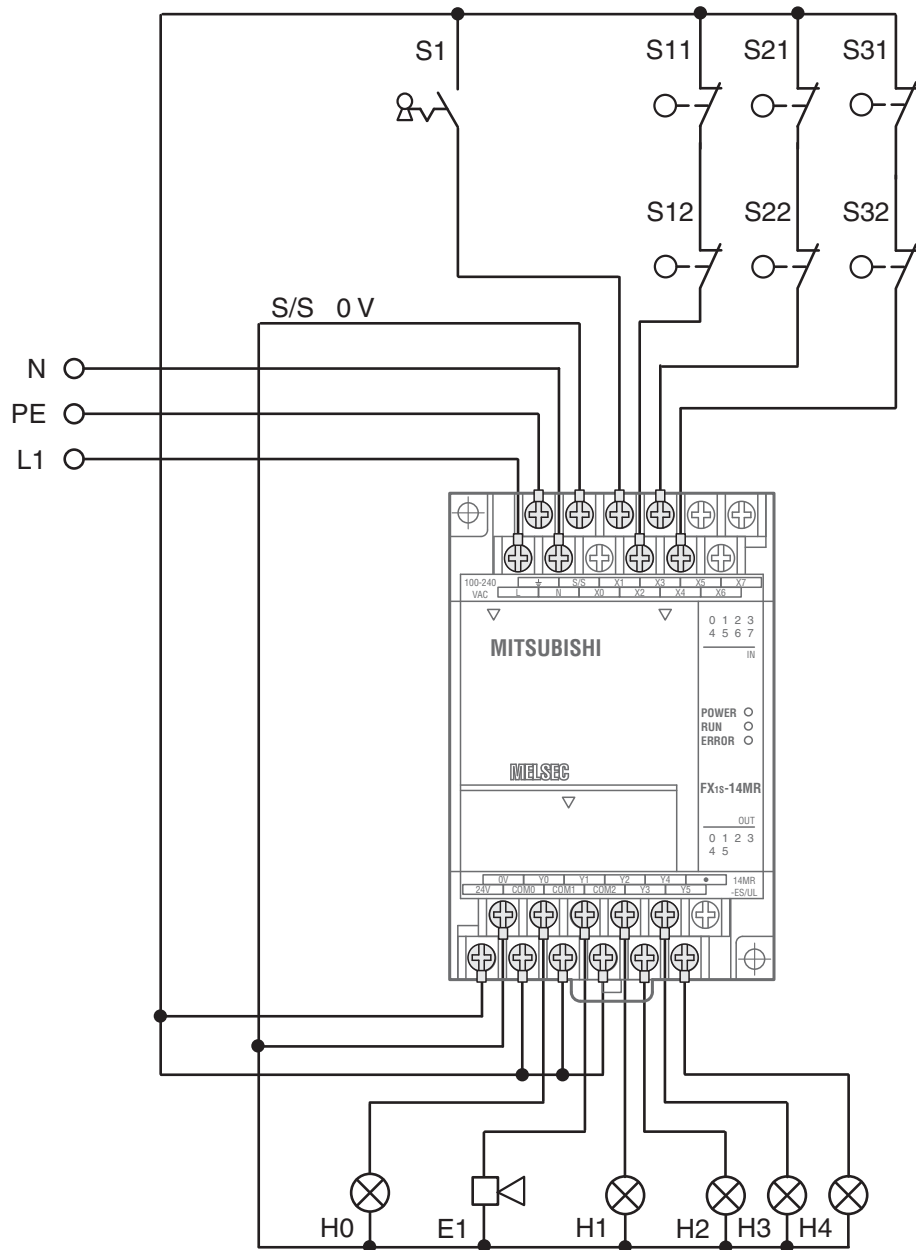
Релейная диаграммаСписок инструкций

```
31 LDI X001
32 RST Y000
33 RST Y001
34 RST Y002
35 RST Y003
36 RST Y004
37 RST Y005
38 RST M1
```

Если пользователь выключил установку охранной сигнализации с помощью выключателя с ключом, все используемые выходы и маркер M1 сбрасываются. Если ранее сработала сигнализация, то до этого момента показывается, какой сигнальный контур был прерван.

Подключение контроллера

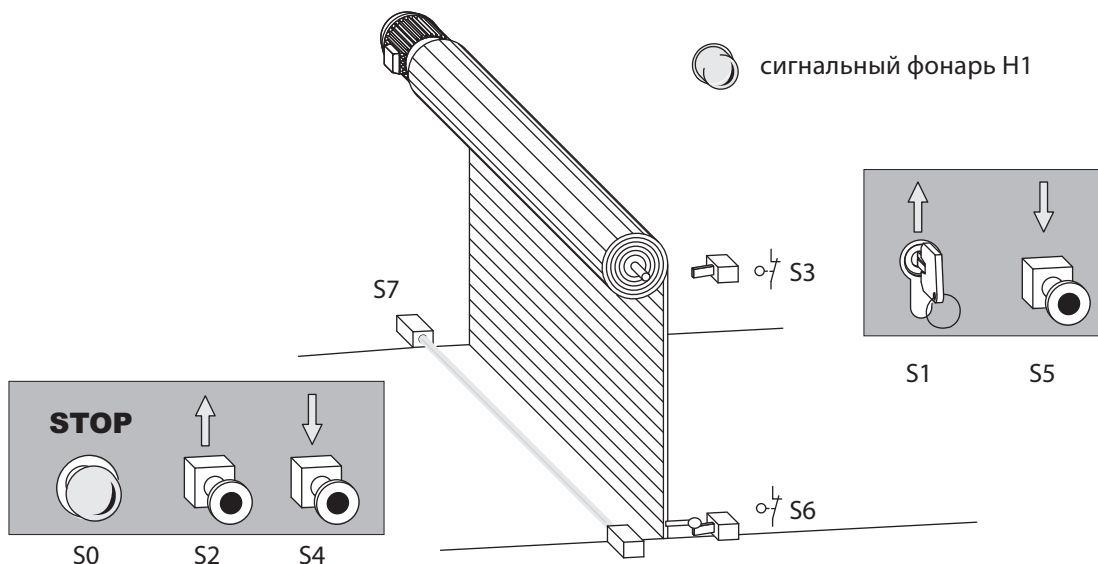
На следующем эскизе показано, насколько малыми затратами можно реализовать установку охранной сигнализации, например, на основе контроллера FX1N-14MR.



3.6.2 Шторные ворота

Описание принципа работы

Шторные ворота склада должны удобно управляться снаружи или изнутри. Но при этом должны учитываться и аспекты безопасности.



● Управление

- Снаружи ворота должны открываться выключателем с ключом S1 и закрываться с помощью кнопки S5. Из помещения ворота должны открываться после нажатия кнопки S2 и закрываться после нажатия кнопки S4.
- Кроме того, дополнительная функция управления по времени должна автоматически закрывать ворота, если они открыты дольше 20 секунд.
- Состояния “Ворота движутся” и “Ворота находятся в неопределенном положении” должны сигнализироваться мигающим сигнальным фонарем.

● Защитные устройства

- Должна иметься возможность в любой момент остановить движение ворот кнопкой “Стоп” (S0). При этом ворота должны оставаться в занимаемом ими в данный момент положении. Эта кнопка “Стоп” не имеет функции аварийного выключения! По этой причине она обрабатывается только в контроллере и не коммутирует внешние напряжения.
- Если при закрывании ворот фоторелейный барьер (S7) распознает препятствие, ворота должны автоматически открываться.
- Для останова электродвигателя в обоих конечных положениях ворот предусмотрены два концевых выключателя S3 (“Ворота открыты”) и S6 (“Ворота закрыты”).

Сопоставление входных и выходных сигналов

Уже из описания принципа работы становится ясным число требуемых входов и выходов. Для управления приводным электродвигателем нужны два выхода. Сигналы сопоставляются входам и выходам контроллера:

| Функция | Обозн. | Адрес | Примечание | |
|---------------------|---|-------|---|--|
| Входы | Кнопочный выключатель "СТОП" | S0 | X0 | Размыкающий контакт (при нажатии кнопки X0 = "0" и ворота останавливаются) |
| | Выключатель с ключом "Открыть ворота" (снаружи) | S1 | X1 | Замыкающие контакты |
| | Кнопка "Открыть ворота" (внутри) | S2 | X2 | |
| | Верхний концевой выключатель ("Ворота открыты") | S3 | X3 | Размыкающий контакт (X2 = "0", если ворота вверх и S3 активирован) |
| | Кнопка "Закрыть ворота" (внутри) | S4 | X4 | Замыкающие контакты |
| | Кнопка "Закрыть ворота" (снаружи) | S5 | X5 | |
| | Нижний концевой выключатель ("Ворота закрыты") | S6 | X6 | Размыкающий контакт (X6 = "0", если ворота вниз и S6 активирован) |
| Фоторелейный барьер | S7 | X7 | Если распознается препятствие, X7 переходит в состояние "1" | |
| Выходы | Сигнальный фонарь | H1 | Y0 | — |
| | Контактор для управления электродвигателем (левое вращение электродвигателя) | K1 | Y1 | Левое вращение = открыть ворота |
| | Контактор для управления электродвигателем (правое вращение электродвигателя) | K2 | Y2 | Правое вращение = закрыть ворота |
| Датчик времени | Задержка автоматического закрывания | — | T0 | Время: 20 секунд |

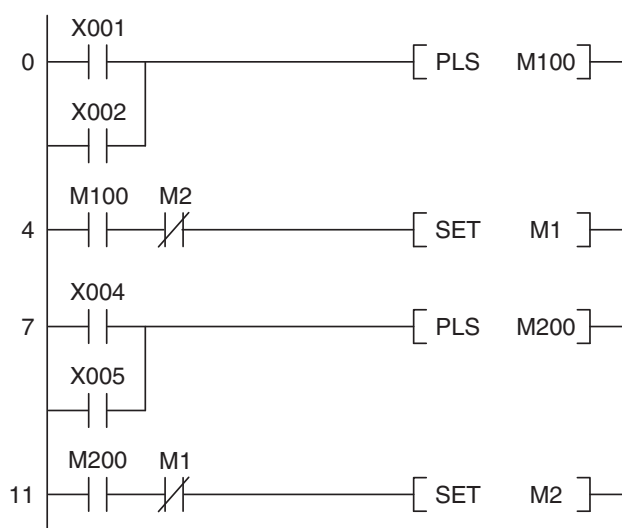
Программа

- Управление шторными воротам с помощью кнопок

Входные сигналы для управления воротами должны преобразовываться программой в две команды для приводного электродвигателя: "Открыть ворота" и "Закрыть ворота". Так как речь идет о сигналах кнопок, т. е. сигналах, приложенных ко входам только короткое время, эти сигналы требуется сохранять в памяти. Для этого устанавливаются и сбрасываются два маркера, которые в программе поначалу заменяют собой выходы:

- M1: Открыть ворота
- M2: Закрыть ворота

Релейная диаграмма



Список инструкций

```

0 LD X001
1 OR X002
2 PLS M100
4 LD M100
5 ANI M2
6 SET M1
7 LD X004
8 OR X005
9 PLS M200
11 LD M200
12 ANI M1
13 SET M2
    
```

Сначала обрабатываются сигналы для открывания ворот: если включить выключатель с ключом S1 или нажать кнопку S2, вырабатывается импульс и маркеру M100 только на один программный цикл присваивается состояние "1". Таким образом, ворота невозможно заблокировать удержанием кнопки или ее заеданием.

Привод разрешается включать только в том случае, если он не вращается в противоположном направлении. По этой причине M1 устанавливается только в том случае, если M2 не установлен.

ПРИМЕЧАНИЕ

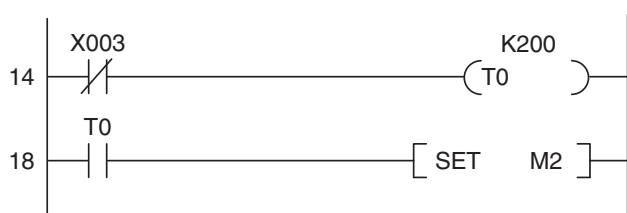
Блокировки направлений вращения должны быть дополнены блокировкой контактов самого контактора вне контроллера (см. электросхему).

Оценка сигналов кнопок S4 и S5 для закрывания ворот реализована аналогичным образом. Здесь опрос M1 на состояние "0" исключает возможность одновременной установки M1 и M2.

- Автоматическое закрывание ворот через 20 секунд

Релейная диаграмма

Список инструкций



```

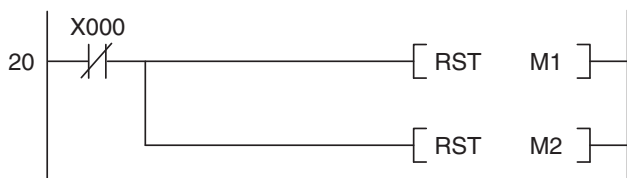
14 LDI X003
15 OUT T0 K200
18 LD T0
19 SET M2
    
```

Если ворота открыты, включается S3 и вход X3 выключается (по соображениям безопасности S3 имеет размыкающий контакт). Теперь начинает отсчитываться время задержки 20 с, реализованное на T0 ($K200 = 200 \times 0,1 \text{ с} = 20 \text{ с}$). По истечении этого времени устанавливается маркер M2, т. е. ворота закрываются.

- Останов ворот с помощью кнопки "СТОП"

Релейная диаграмма

Список инструкций



```

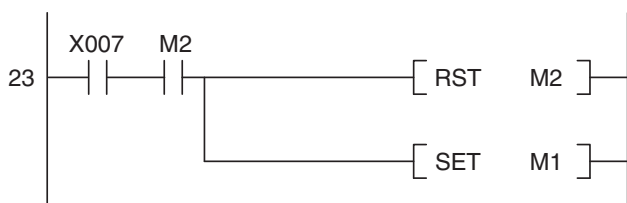
20 LDI X000
21 RST M1
22 RST M2
    
```

При нажатии кнопки "СТОП" S0 оба маркера M1 и M2 сбрасываются и поэтому ворота останавливаются.

- Распознавание препятствия с помощью фоторелейного барьера

Релейная диаграмма

Список инструкций



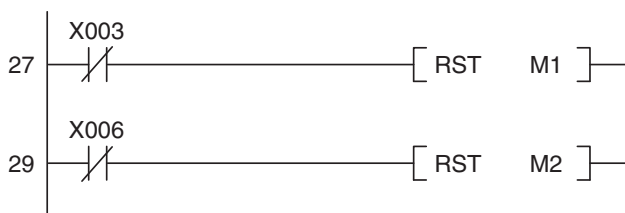
```

23 LD X007
24 AND M2
25 RST M2
26 SET M1
    
```

Если во время закрывания фоторелейный барьер распознал препятствие, M2 сбрасывается и в результате этого процесс закрывания прерывается. Затем устанавливается M1 и ворота снова открываются.

● Отключение электродвигателя концевыми выключателями

Релейная диаграмма



Список инструкций

```

27 LDI X003
28 RST M1
29 LDI X006
30 RST M2
    
```

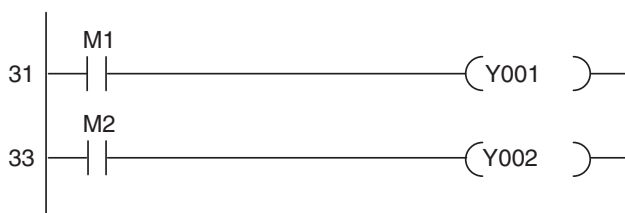
Открытые ворота активируют концевой выключатель S3 и выключают вход X3. В результате этого M1 сбрасывается и привод останавливается. Если ворота достигли нижнего положения, S6 включается, X6 выключается и в результате этого M2 сбрасывается и привод останавливается. По соображениям безопасности концевые выключатели имеют размыкающие контакты. Благодаря этому привод выключается (или предотвращается его включение) и в случае обрыва соединения между выключателем и входом.

ПРИМЕЧАНИЕ

Концевые выключатели должны останавливать привод и независимо от контроллера. Для этого они должны быть встроены в электропроводку (см. электросхему).

● Управление электродвигателем

Релейная диаграмма



Список инструкций

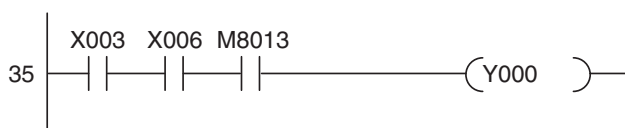
```

31 LD M1
32 OUT Y001
33 LD M2
34 OUT Y002
    
```

В конце программы сигнальные состояния обоих маркеров M1 и M2 передаются выходам Y001 и Y002.

● Сигнальный фонарь: “Ворота движутся” и “Ворота в неопределенном положении”

Релейная диаграмма



Список инструкций

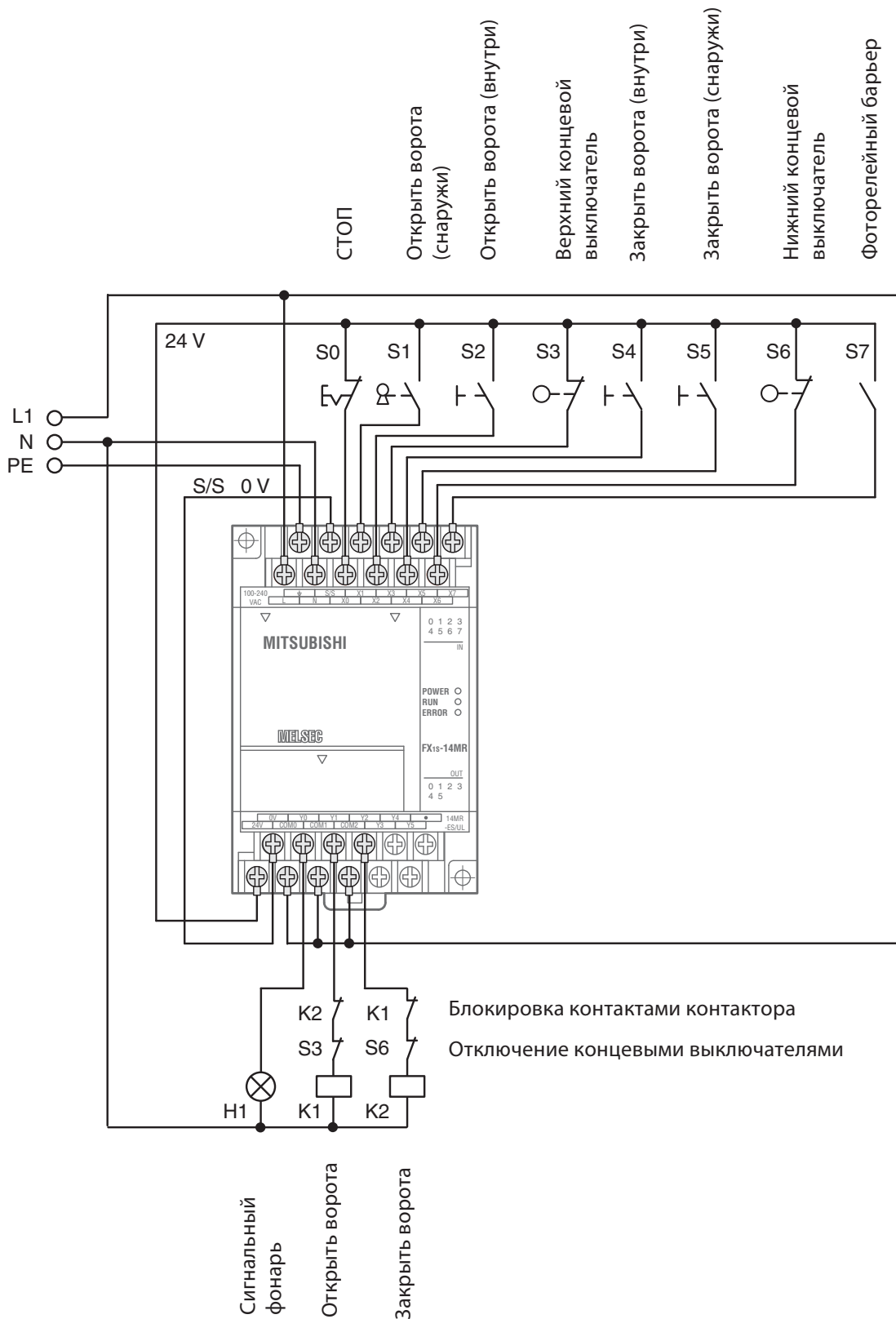
```

35 LD X003
36 AND X006
37 AND M8013
38 OUT Y000
    
```

Если ни один из двух концевых выключателей не активирован, то это означает, что ворота либо открываются, либо закрываются, либо остановлены в промежуточном положении. В этих случаях активируется мигающий сигнальный фонарь. В качестве такта мигания используется специальный маркер M8013, который автоматически устанавливается и сбрасывается с 1-секундным тактом (см. раздел 4.2).

Подключение контроллера

Для вышеописанной системы управления шторных ворот можно применить, например, контроллер FX1N-14MR.

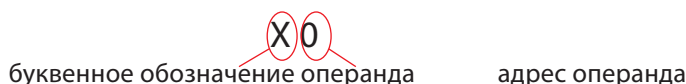


4 Подробное описание операндов

Операнды контроллера используются в управляющих командах, т. е., программа контроллера может опрашивать сигнальные состояния или значения операндов или влиять на них. Операнд состоит из

- буквенного обозначения и адреса операнда.

Пример указания операнда (например, вход 0):



4.1 Входы и выходы

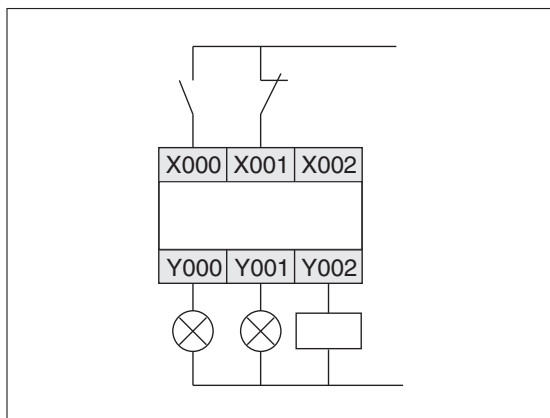
Входы и выходы соединяют контроллер с управляемым процессом. При опросе входа программой контроллера проверяется напряжение на входной клемме контроллера. Так как речь идет о цифровых входах, они могут принимать только два сигнальных состояния: ВКЛ. и ВЫКЛ. Если напряжение на входной клемме достигает 24 В, вход включен (состояние сигнала "1"). При более низком напряжении вход считается выключенным (состояние сигнала "0").

Для буквенного обозначения операндов входов в контроллерах MELSEC применяется буква "X". Один и тот же вход может опрашиваться в программе сколько угодно часто.

ПРИМЕЧАНИЕ

Программа контроллера не может изменять состояние входов. Например, невозможно указать вход в качестве операнда команды OUT.

Если в качестве операнда команды вывода используется выход, результат логической операции (состояние операнда) выводится на выходную клемму контроллера. В случае релейных выходов соответствующее реле притягивает контакты (все реле имеют замыкающие контакты), а в случае контроллеров с транзисторными выходами происходит открытие соответствующего транзистора и, тем самым, включение подключенного потребителя.



Пример подключения выключателей ко входам, а также ламп или контакторов к выходам контроллера MELSEC.

Операнды выходов имеют буквенное обозначение "Y". Выходы можно использовать не только в командах вывода, но и в логических операциях. Однако ни в коем случае нельзя один и тот же выход многократно программировать в качестве операнда команды вывода (см. также раздел 3.4.2).

Следующая таблица содержит обзор входов и выходов контроллеров MELSEC семейства FX.

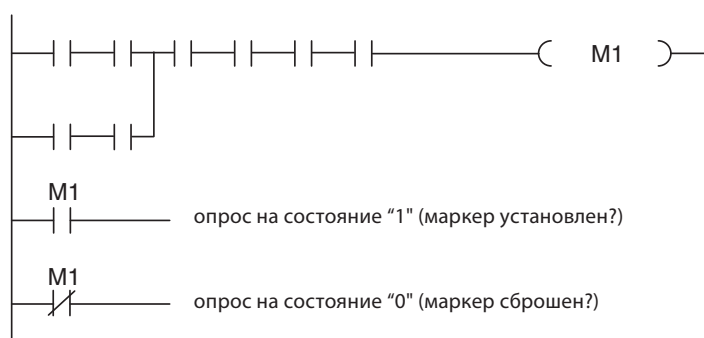
| Операнд | | Входы | Выходы |
|---|--------|---|--|
| Буквенное обозначение операнда | | X | Y |
| Тип операнда | | битовый операнд | |
| Значения, которые может принимать операнд | | 0 или 1 | |
| Указание адреса операнда | | восьмеричное | |
| Количество операндов и адресов (в зависимости от типа базового блока) | FX1S | 6 (X00–X05) 8 (X00–X07) 12 (X00–X07, X10, X11, X12, X13) 16 (X00–X07, X10–X17) | 4 (Y00–Y03) 6 (Y00–Y05) 8 (Y00–Y07) 14 (Y00–Y07, Y10–Y15) |
| | FX1N | 8 (X00–X07) 14 (X00–X07, X10–X15) 24 (X00–X07, X10–X17, X20–X27) 36 (X00–X07, X10–X17, X20–X27, X30–X37, X40, X41, X42, X43) С помощью модулей расширения количество входов можно увеличить максимум до 84 (X123). Однако сумма входов и выходов не должна превышать 128. | 6 (Y00–Y05) 10 (Y00–Y07, Y10, Y11) 16 (Y00–Y07, Y10–Y17) 24 (Y00–Y07, Y10–Y17, Y20–Y27) С помощью модулей расширения количество выходов можно увеличить максимум до 64 (Y77). Однако сумма входов и выходов не должна превышать 128. |
| | FX2N | 8 (X00–X07) 16 (X00–X07, X10–X17) 24 (X00–X07, X10–X17, X20–X27) 32 (X00–X07, X10–X17, X20–X27, X30–X37) 40 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47) 64 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47, X50–X57, X60–X67, X70–X77) | 8 (Y00–Y07) 16 (Y00–Y07, Y10–Y17) 24 (Y00–Y07, Y10–Y17, Y20–Y27) 32 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37) 40 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47) 64 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47, Y50–Y57, Y60–Y67, Y70–Y77) |
| | FX2NC | 8 (X00–X07) 16 (X00–X07, X10–X17) 32 (X00–X07, X10–X17, X20–X27, X30–X37) 48 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47, X50–X57) | 8 (Y00–Y07) 16 (Y00–Y07, Y10–Y17) 32 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37) 48 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47, Y50–Y57) |
| | FX3G | 8 (X00–X07) 14 (X00–X07, X10–X15) 24 (X00–X07, X10–X17, X20–X27) 36 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X43) | 6 (Y00–Y05) 10 (Y00–Y07, Y10–Y11) 16 (Y00–Y07, Y10–Y17) 24 (Y00–Y07, Y10–Y17, Y20–Y27) |
| | FX3U* | 8 (X00–X07) 16 (X00–X07, X10–X17) 24 (X00–X07, X10–X17, X20–X27) 32 (X00–X07, X10–X17, X20–X27, X30–X37) 40 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47) | 8 (Y00–Y07) 16 (Y00–Y07, Y10–Y17) 24 (Y00–Y07, Y10–Y17, Y20–Y27) 32 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37) 40 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47) |
| | FX3UC* | 8 (X00–X07) 16 (X00–X07, X10–X17) 32 (X00–X07, X10–X17, X20–X27, X30–X37) 48 (X00–X07, X10–X17, X20–X27, X30–X37, X40–X47, X50–X57) | 8 (Y00–Y07) 16 (Y00–Y07, Y10–Y17) 32 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37) 48 (Y00–Y07, Y10–Y17, Y20–Y27, Y30–Y37, Y40–Y47, Y50–Y57) |

* С помощью модулей расширения количество входов можно увеличить максимум до 248 (X367) и количество выходов до 248 (Y367). Однако сумма входов и выходов не должна превышать 256.

4.2 Маркеры

В программе контроллера часто требуется сохранять бинарные промежуточные результаты (состояние сигнала "0" или "1"). Для этой цели в контроллере имеются "маркеры" (буквенное обозначение операнда: "M").

В маркерах сохраняется (промежуточный) результат логической операции, например, с помощью команды OUT. После этого его можно опрашивать другими логическими операциями. Маркеры помогают придать программе хорошую обзорность и сэкономить шаги программы. Результаты логических операций, многократно используемые программой, можно сохранить в маркере, а затем использовать сколько угодно часто.



Помимо "обычных" маркеров, в контроллерах семейства FX имеются также так называемые фиксируемые маркеры. Обычные, небуферизованные маркеры при выключении питания контроллера сбрасываются на состояние "0" и имеют это состояние и после включения контроллера. В отличие от них, фиксируемые маркеры сохраняют заложенную в них информацию и при выпадении напряжения.

| Операнд | Маркеры | | |
|---|--------------------------|---|--------------------|
| | Небуферизованные маркеры | Фиксируемые маркеры | |
| Буквенное обозначение операнда | M | | |
| Тип операнда | битовый операнд | | |
| Значения, которые может принимать операнд | 0 или 1 | | |
| Указание адреса операнда | десятичное | | |
| Количество операндов и адресов | FX1S | 384 (M0–M383) | 128 (M384–M511) |
| | FX1N | 384 (M0–M383) | 1152 (M384–M1535) |
| | FX2N | 500 (M0–M499)1 | 524 (M500–M1023)2 |
| | FX2NC | | 2048 (M1024–M3071) |
| | FX3G | 384 (M0–M383) | 1152 (M384–M1535) |
| | | 6144 (M1536–M7679)③ | |
| FX3U FX3UC | 500 (M0–M499)1 | 524 (M500–M1023)2 6656 (M1024–M7679) | |

- ① В параметрах контроллера этим маркерам можно также присвоить функцию фиксируемых маркеров.
- ② В параметрах контроллера этим маркерам можно также присвоить функцию небуферизованных маркеров.
- ③ Если установлена опциональная батарейка, в параметрах контроллера этим маркерам можно присвоить функцию фиксируемых маркеров. В этом случае они имеют буферное батарейное питание.

4.2.1 Специальные маркеры

Помимо маркеров, которые пользователь может в программе произвольно включать и выключать, существуют также специальные маркеры. Эти маркеры занимают область, начиная с адреса M8000, и показывают определенные состояния системы или влияют на обработку программы. В следующей таблице показана лишь небольшая часть специальных маркеров.

| Спец. маркер | Описание | Обработка в программе |
|--------------|--|-----------------------|
| M8000 | В режиме "RUN" контроллера этот маркер всегда имеет состояние "1". | Опрос состояния |
| M8001 | В режиме "RUN" контроллера этот маркер всегда имеет состояние "0". | |
| M8002 | Импульс инициализации (после включения режима "RUN" этот маркер на время одного программного цикла принимает состояние "1".) | |
| M8004 | Неисправность контроллера | |
| M8005 | Низкое напряжение батарейки | |
| M8013 | Датчик тактовых импульсов: 1 секунда | |
| M8031 | Стереть все операнды (кроме регистра данных D), которые не внесены в область с буферным питанием от батарейки. | Опрос состояния |
| M8034 | Заблокировать выходы; выходы не включаются, однако программа продолжает обрабатываться. | Присвоение состояния |

4.3 Таймеры

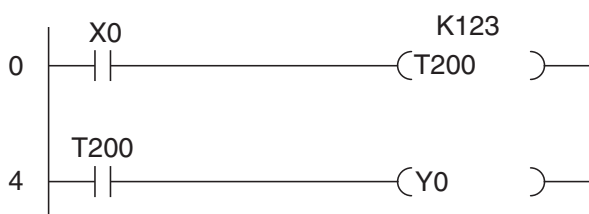
При управлении процессами часто требуется включать или выключать определенные процессы с задержкой. В релейной технике для этого используются реле времени, а в контроллере - внутренние элементы выдержки времени (англ.: timer).

В принципе, таймеры считают внутренний такт контроллера (например, импульсы с тактом в 0,1 с). Если значение счета достигло значения, заданного в программе, выход таймера включается.

Все таймеры действуют в качестве задержки включения и активируются путем подачи на них сигнала "1". Для запуска и сброса таймеры программируются как выходы. В пределах одной программы выход таймера может опрашиваться сколь угодно часто

Релейная диаграмма

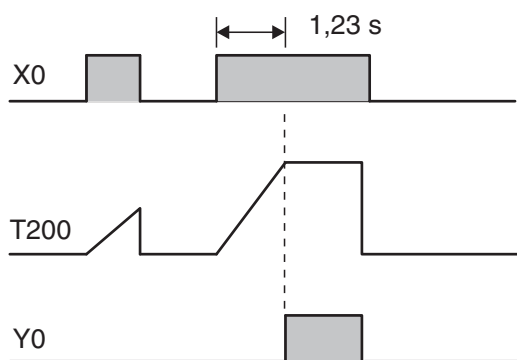
Список инструкций



```

0 LD X0
1 OUT T200 K123
4 LD T200
5 OUT Y0
    
```

Таймер T200 запускается, если вход X0 включен. Заданное значение составляет 123 x 10 мс = 1,23 с. По истечении 1,23 секунды T200 включает выход Y0. Для изображенного сверху примера образуется следующая диаграмма изменения сигналов:



До тех пор, пока X0 включен, таймер считает внутренние 10-миллисекундные импульсы. После достижения заданного значения включается выход T200.

Если вход X0 или напряжение питания контроллера выключается, таймер сбрасывается и его выход тоже выключается.

Указать заданное значение времени можно и косвенным путем, сохранив в регистре данных десятичное числовое значение. Эта возможность описана в разделе 4.6.1.

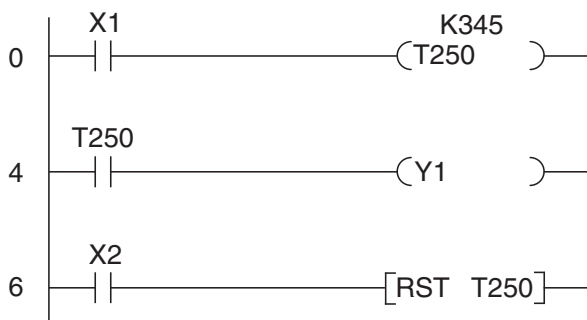
Фиксируемые звенья выдержки времени

Помимо вышеописанных звеньев выдержки времени, контроллеры серий FX1N, FX2N, FX2NC, FX3G, FX3U и FX3UC имеют также фиксируемые звенья выдержки времени, которые сохраняют уже достигнутое фактическое значение времени и после отключения управляющей логической связи.

Фактические значения времени вносятся в память, содержимое которой сохраняется и при выпадении напряжения.

Пример для программирования фиксируемого таймера:

Релейная диаграмма



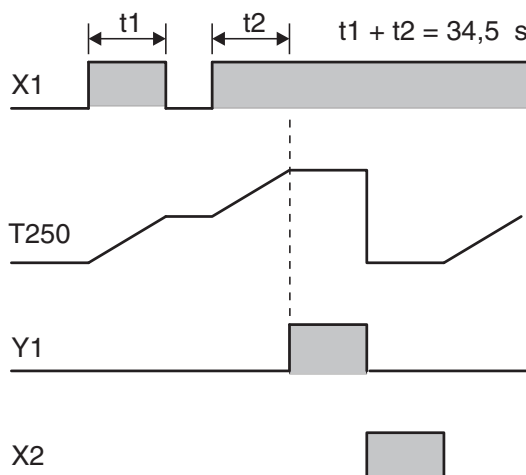
Список инструкций

```

0 LD X0
1 OUT T250 K345
4 LD T250
5 OUT Y1
6 LD X2
7 RST T250

```

Таймер T250 запускается, если вход X0 включен. В качестве заданного значения задано 345 x 0,1 с = 34,5 с. После достижения заданного значения T250 включает выход Y1. С помощью входа X2 таймер сбрасывается и его выход выключается.



Если X1 включен, таймер считает внутренние 100-миллисекундные импульсы. Если X1 выключается, достигнутое на этот момент фактическое значение сохраняется. Если фактическое значение соответствует заданному, включается выход таймера.

Так как при выключении входа X1 или питания контроллера фактическое значение времени не стирается, для его стирания в программе необходима особая команда. Для сброса таймера T250 и выключения его выхода используется вход X2.

Обзор таймеров в базовых блоках MELSEC семейства FX

| Операнд | | Таймеры | | |
|---|---------------|--|---------------------|---------------|
| | | Обычные таймеры | Фиксируемые таймеры | |
| Буквенное обозначение операнда | | T | | |
| Тип операнда (для управления и опроса) | | битовый операнд | | |
| Значения, которые может принимать операнд (выход таймера) | | 0 или 1 | | |
| Указание адреса операнда | | десятичное | | |
| Задание заданного значения времени | | В виде целой десятичной константы. Задание осуществляется либо непосредственно в команде, либо косвенно в регистре данных. | | |
| Количество операндов и адресов | FX1S | 100 мс (диап. от 0,1 до 3276,7 с) | 63 (T0–T62) | — |
| | | 10 мс (диап. от 0,01 до 327,67 с) | 31 (T32–T62)* | — |
| | | 1 мс (диап. от 0,001 до 32,767 с) | 1 (T63) | — |
| | FX1N | 100 мс (диап. от 0,1 до 3276,7 с) | 200 (T0–T199) | 6 (T250–T255) |
| | | 10 мс (диап. от 0,01 до 327,67 с) | 46 (T200–T245) | — |
| | | 1 мс (диап. от 0,001 до 32,767 с) | 4 (T246–T249) | — |
| | FX2N FX2NC | 100 мс (диап. от 0,1 до 3276,7 с) | 200 (T0–T199) | 6 (T250–T255) |
| | | 10 мс (диап. от 0,01 до 327,67 с) | 46 (T200–T245) | — |
| | | 1 мс (диап. от 0,001 до 32,767 с) | — | 4 (T246–T249) |
| | FX3G | 100 мс (диап. от 0,1 до 3276,7 с) | 200 (T0–T199) | 6 (T250–T255) |
| | | 10 мс (диап. от 0,01 до 327,67 с) | 46 (T200–T245) | — |
| | | 1 мс (диап. от 0,001 до 32,767 с) | 64 (T256–T319) | 4 (T246–T249) |
| | FX3U FX3UC | 100 мс (диап. от 0,1 до 3276,7 с) | 200 (T0–T199) | 6 (T250–T255) |
| | | 10 мс (диап. от 0,01 до 327,67 с) | 46 (T200–T245) | — |
| | | 1 мс (диап. от 0,001 до 32,767 с) | 256 (T256–T511) | 4 (T246–T249) |

* Эти таймеры имеются только в том случае, если специальный маркер M8028 включен. В этом случае количество 100-миллисекундных таймеров уменьшается до 32 (T0–T31).

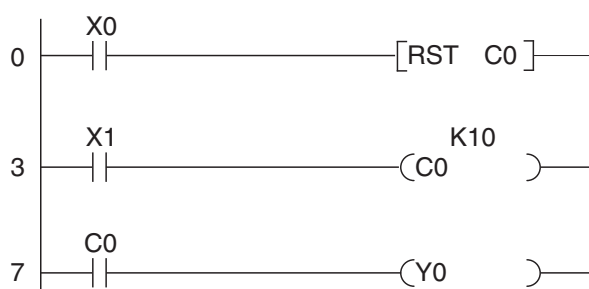
4.4 Счетчики

Для программирования процессов счета в контроллерах семейства FX имеются внутренние счетчики (англ.: counter).

Счетчики считают сигналы, подаваемые на их вход программой. Если значение счета достигло заданного значения, которое также задается с помощью программы, выход счетчика включается. Этот выход в пределах одной программы можно опрашивать сколь угодно часто.

Пример программирования счетчиков:

Релейная диаграмма

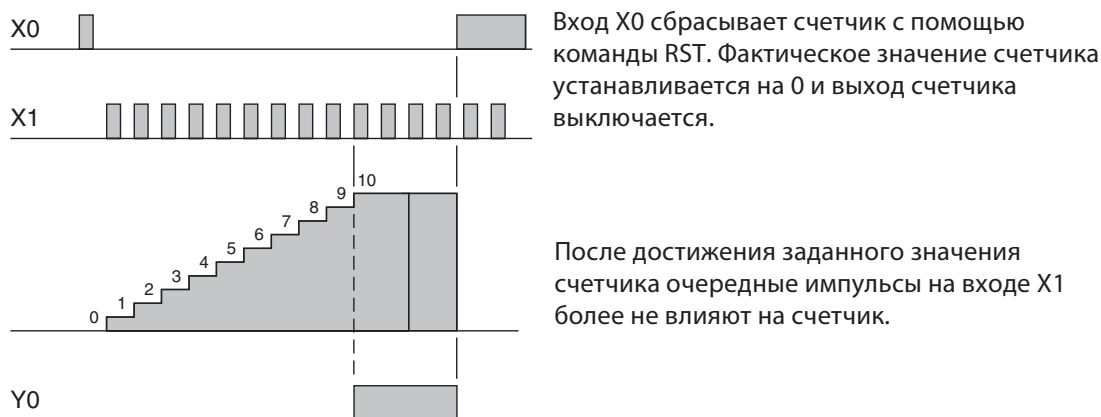


Список инструкций

| | | | |
|---|-----|----|-----|
| 0 | LD | X0 | |
| 1 | RST | C0 | |
| 3 | LD | X1 | |
| 4 | OUT | C0 | K10 |
| 7 | LD | C0 | |
| 8 | OUT | Y0 | |

Каждый раз, когда включается вход X1, счетчик C0 повышает свое числовое значение на 1. Выход Y0 устанавливается после того, как вход X1 был включен и выключен десять раз (в качестве заданного значения счетчика запрограммировано K10).

На рисунке ниже показана диаграмма изменения сигналов для этого примера программы.



Счетчики могут быть 16-битовыми и 32-битовыми. Это обозначение относится к ячейке памяти, используемой для значения счета. В следующей таблице показаны важнейшие свойства этих счетчиков.

| Признак | 16-битовый счетчик | 32-битовый счетчик |
|-------------------------------------|--|---|
| Направление счета | Возрастающий счет | Возрастающий и нисходящий счет (направление счета устанавливается путем включения и выключения специального маркера) |
| Диапазон заданного значения | от 1 до 32767 | от -2 147 483 648 до 2 147 483 647 |
| Задание заданного значения | В виде десятичной константы (K) непосредственно в команде или косвенно в регистре данных | В виде десятичной константы (K) непосредственно в команде или косвенно в паре регистров данных. |
| Поведение при переполнении счетчика | Считает максимум до 32767, после чего фактическое значение более не изменяется | Кольцевой счетчик: после отсчета до 2 147 483 647 следующим значением является -2 147 483 648. (При нисходящем счете происходит скачок от -2 147 483 648 к 2 147 483 647) |
| Выход счетчика | После достижения заданного значения выход остается включенным | При восходящем счете после достижения заданного значения выход остается включенным. При нисходящем счете занижение заданного значения сбрасывает выход. |
| Сброс | С помощью команды RST фактическое значение счетчика стирается и выход выключается | |

Помимо обычных счетчиков в контроллерах MELSEC семейства FX имеются также так называемые высокоскоростные счетчики. Это 32-битовые счетчики, которые быстро перерабатывают внешние сигналы счета, опрашиваемые через входы с X0 по X7. С помощью этих счетчиков (в сочетании со специальными командами) можно, например, легко решать задачи позиционирования.

Высокоскоростные счетчики работают по принципу прерывания. При этом программа контроллера прерывается и сразу реагирует на сигнал счетчика. Подробное описание высокоскоростных счетчиков вы найдете в руководстве по программированию контроллеров MELSEC семейства FX, артикул 136748.

Обзор счетчиков

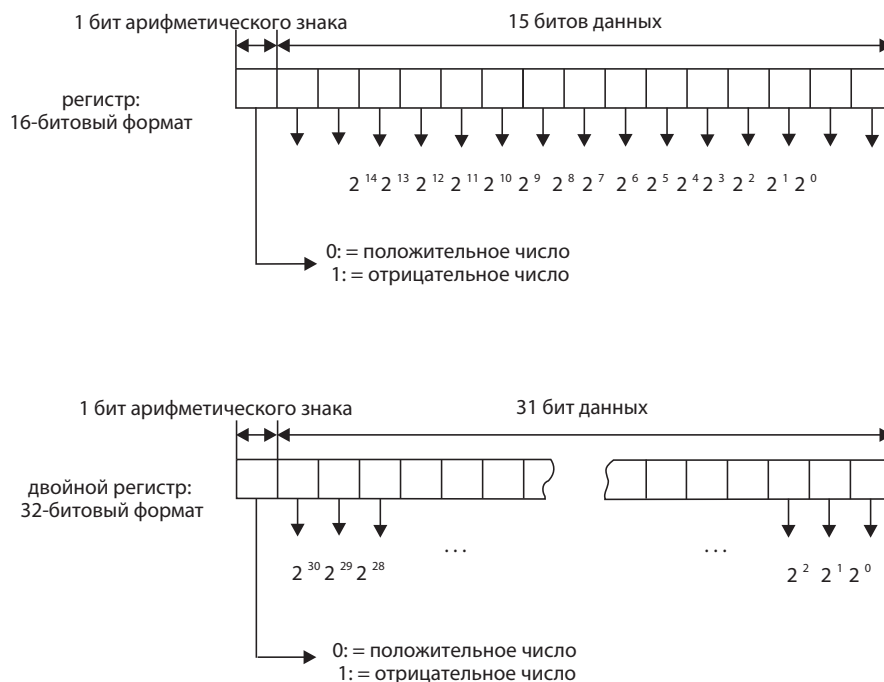
| Операнд | | Счетчики | | |
|--|---------------|---|-----------------|----------------------|
| | | Обычные счетчики | | Фиксируемые счетчики |
| Буквенное обозначение операнда | | C | | |
| Тип операнда (для управления и опроса) | | битовый операнд | | |
| Значения, которые может принимать выход счетчика | | 0 или 1 | | |
| Указание адреса операнда | | десятичное | | |
| Задание заданного значения счетчика | | В виде целой десятичной константы. Задание осуществляется либо непосредственно в команде, либо косвенно в регистре данных (в случае 32-битовых счетчиков - в двух регистрах данных) | | |
| Количество операндов и адресов | FX1S | 16-битовый | 16 (C0–C15) | 16 (C16–C31) |
| | | 32-битовый | — | — |
| | | 32-битовый высокоскор. | — | 21 (C235–C255) |
| | FX1N | 16-битовый | 16 (C0–C15) | 184 (C16–C199) |
| | | 32-битовый | 20 (C200–C219) | 15 (C220–C234) |
| | | 32-битовый высокоскор. | — | 21 (C235–C255) |
| | FX2N FX2NC | 16-битовый | 100 (C0–C99)2 | 100 (C100–C199) ② |
| | | 32-битовый | 20 (C200–C219)2 | 15 (C220–C234) ② |
| | | 32-битовый высокоскор. | 21 (C235–C255)2 | |
| | FX3G | 16-битовый | 16 (C0–C15) | 184 (C16–C199) |
| | | 32-битовый | 20 (C200–C219) | 15 (C220–C234) |
| | | 32-битовый высокоскор. | — | 21 (C235–C255) |
| | FX3U FX3UC | 16-битовый | 100 (C0–C99)2 | 100 (C100–C199) ② |
| | | 32-битовый | 20 (C200–C219)2 | 15 (C220–C234) ② |
| | | 32-битовый высокоскор. | 21 (C235–C255)2 | |

- ① В случае фиксируемых счетчиков фактическое значение счетчика сохраняется и при выключении напряжения питания.
- ② В параметрах контроллера можно выбрать, должны ли сохраняться фактические значения этих счетчиков при выключении напряжения питания.

4.5 Регистры

Маркеры контроллера служат для сохранения в памяти двоичных промежуточных результатов. Однако состояние маркера несет в себе информацию только типа “вкл./выкл.” или 0/1. Поэтому маркер не пригоден для сохранения измеренных значений или результатов расчетов. Для этой цели контроллеры семейства FX имеют регистры.

Регистр состоит из 16 битов или одного слова (см. раздел 3.2). Соединив два 16-битовых регистра, можно образовать “двойной регистр” из 32 битов.



В регистре можно сохранять значения в диапазоне от 0000h до FFFFh (от -32768 до 32767), а в двойном регистре - значения в диапазоне от 00000000h до FFFFFFFFh (от -2 147 483 648 до 2 147 483 647).

Для работы с регистрами контроллеры семейства FX имеют множество команд, с помощью которых, например, осуществляется запись значений в регистр, считывание значений из регистров, копирование содержимого регистров, сравнение или арифметические вычисления (см. раздел 5).

4.5.1 Регистры данных

Регистры данных в программе контроллера можно использовать в качестве запоминающего устройства. Значение, вносимое программой контроллера в регистр данных, сохраняется в нем без изменений до тех пор, пока оно не будет перезаписано в программе другим значением.

В командах для 32-битовых данных указывается только адрес 16-битового регистра – старшей частью 32-битовых данных автоматически занимается следующий регистр. Если, например, для сохранения 32-битового значения указан регистр D0, то D0 содержит биты с 0 по 15, а D1 - биты с 16 по 31.

Что происходит при выключении или останове контроллера

Помимо небуферизованных регистров, содержимое которых при останове контроллера или выключении его питания стирается, контроллеры имеют регистры, содержимое которых в этих случаях сохраняется (фиксируемые регистры).

ПРИМЕЧАНИЕ

Если специальный маркер M8033 включен, то и содержимое небуферизованных регистров данных при останове контроллера не стирается.

Обзор регистров данных

| Операнд | Регистры данных | | | |
|---|--|----------------------------|------------------------------|-------------------|
| | Небуферизованные регистры | Фиксируемые регистры | | |
| Буквенное обозначение операнда | D | | | |
| Тип операнда | словный операнд (два регистра можно объединить в двойной регистр) | | | |
| Значения, которые может принимать операнд | 16-битовый регистр: от 0000h до FFFFh (от -32768 до 32767) 32-битовый регистр: от 00000000h до FFFFFFFFh (от -2 147 483 648 до 2 147 483 647) | | | |
| Указание адреса операнда | десятичное | | | |
| Количество операндов и адресов | FX1S | 128 (D0–D127) | 128 (D128–D255) | |
| | FX1N | 128 (D0–D127) | 7872 (D128–D7999) | |
| | FX2N FX2NC | 200 (D0–D199) ^① | 312 (D200–D511) ^② | 7488 (D512–D7999) |
| | | FX3G | 128 (D0–D127) | 972 (D128–D1099) |
| | 972 (D1100–D7999) ^③ | | | |
| | FX3U FX3UC | 200 (D0–D199) ^① | 312 (D200–D511) ^② | 7488 (D512–D7999) |

① В параметрах контроллера этим регистрам можно также присвоить функцию фиксируемых регистров.

② В параметрах контроллера этим регистрам можно также присвоить функцию небуферизованных регистров.

③ Если установлена опциональная батарейка, в параметрах контроллера этим маркерам можно присвоить функцию фиксируемых маркеров. В этом случае они имеют буферное батарейное питание.

4.5.2

Специальные регистры

По аналогии со специальными маркерами (раздел 4.2.1), начинающимися с адреса M8000, регистры с адреса D8000 являются специальными регистрами. Часто имеется даже непосредственная взаимосвязь между специальным маркером и специальным регистром. Так, например, специальный маркер M8005 показывает, что напряжение батарейки контроллера слишком низкое, а специальный регистр D8005 содержит измеренное значение напряжения. Некоторая небольшая часть специальных регистров приведена в следующей таблице.

| Специальный регистр | Описание | Обработка в программе |
|---------------------|---|--|
| D8004 | Адрес маркера ошибки (показывает, какой маркер ошибки установлен) | Опрос содержимого |
| D8005 | Напряжение батарейки (содержимое "36" соответствует, например, 3,6 вольтам) | |
| D8010 | Текущее время цикла программы | |
| D8013–D8019 | Время суток и дата встроенных часов | Опрос содержимого Изменение содержимого |
| D8030 | Считанное значение потенциометра VR1 (от 0 до 255) | Опрос содержимого (только у FX1S и FX1N) |
| D8031 | Считанное значение потенциометра VR2 (от 0 до 255) | |

Регистры, изменяемые извне

В контроллерах серий FX1S, FX1N и FX3G имеются два потенциометра, с помощью которых можно изменять содержимое специальных регистров D8030 и D8031 в диапазоне от 0 до 255 (см. раздел 4.6.1). С помощью этих потенциометров можно, например, изменять заданные значения для таймеров и счетчиков, не подключая программатор.

4.5.3 Регистры файлов

Содержимое регистров файлов не утрачивается при выключении напряжения питания. По этой причине в регистрах файлов можно сохранять значения, которые после включения контроллера передаются в регистры данных и используются программой, например, для расчетов, сравнения или в качестве заданных значений для таймеров.

Регистры файлов по своему устройству не отличаются от регистров данных. Более того, они даже образуются из регистров данных с D1000 по D7999, объединенных в блоки по 500 адресов.

| Операнд | | Регистры файлов |
|---|---------------|--|
| Буквенное обозначение операнда | | D |
| Тип операнда | | словный операнд (два регистра можно объединить в двойной регистр) |
| Значения, которые может принимать операнд | | 16-битовый регистр: от 0000h до FFFFh (от -32768 до 32767) 32-битовый регистр: от 00000000h до FFFFFFFFh (от -2 147 483 648 до 2 147 483 647) |
| Указание адреса операнда | | десятичное |
| Количество операндов и адресов | FX1S | 1500 (D1000–D2499) В параметрах контроллера можно установить максимум 3 блока, каждый из которых состоит из 500 регистров файлов. |
| | FX1N | 7000 (D1000–D7999) В параметрах контроллера можно установить макс. 14 блоков, каждый из которых состоит из 500 регистров файлов. |
| | FX2N | |
| | FX2NC | |
| | FX3G | |
| | FX3U FX3UC | |

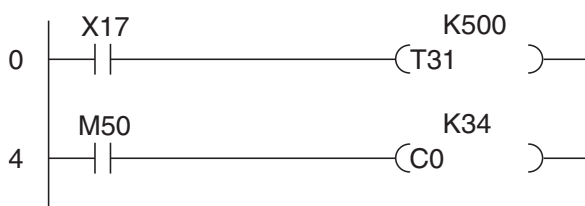
Регистры файлов подробно описаны в руководстве по программированию контроллеров MELSEC семейства FX, артикул 136748.

4.6 Советы по программированию таймеров и счетчиков

4.6.1 Косвенное задание значений для таймеров и счетчиков

Заданные значения времени и счета в программе можно передавать таймерам и счетчикам непосредственно, в рамках команды вывода:

Релейная диаграмма



Список инструкций

| | | | |
|---|-----|-----|------|
| 0 | LD | X17 | |
| 1 | OUT | T31 | K500 |
| 4 | LD | M50 | |
| 5 | OUT | C0 | K34 |

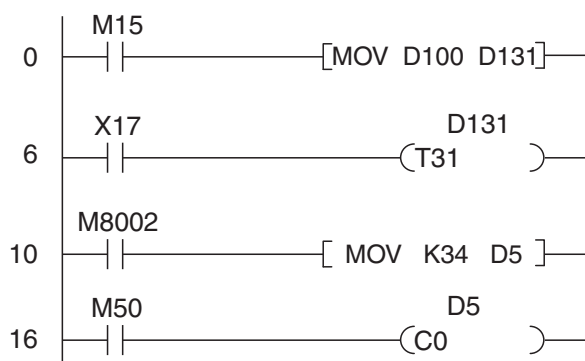
В изображенном вверху примере T31 представляет собой 100-миллисекундный таймер. С помощью константы "500" время задержки устанавливается на $500 \times 0,1 \text{ с} = 50 \text{ с}$. Заданное значение для счетчика C0 устанавливается непосредственно на "34".

Преимущество этого способа задания заключается в том, что более не требуется заботиться о заданном значении. Заданные программой значения действительны и после выпадения напряжения или сразу после включения контроллера. Недостаток, однако, заключается в том, что для изменения заданного значения требуется изменить программу. При этом заданные значения (особенно для таймеров) зачастую окончательно подбираются лишь во время ввода контроллера в эксплуатацию и тестирования программы.

Однако заданные значения для таймеров и счетчиков можно вводить и в регистры данных и считывать программой из этих регистров. Благодаря этому предварительно заданные значения можно быстро изменять с помощью подключенного программатора. В этом случае возможно также задание значений с помощью выключателей в пульте или панели управления.

На рисунке ниже изображены примеры косвенного задания заданных значений:

Релейная диаграмма



Список инструкций

| | | | |
|----|-----|-------|------|
| 0 | LD | M15 | |
| 1 | MOV | D100 | D131 |
| 6 | LD | X17 | |
| 7 | OUT | T31 | D131 |
| 10 | LD | M8002 | |
| 11 | MOV | K34 | D5 |
| 16 | LD | M50 | |
| 17 | OUT | C0 | D5 |

- Если маркер M15 имеет состояние "1", содержимое регистра данных D100 копируется в регистр данных D131. Этот регистр содержит заданное значение для T31. Содержимое D100 можно изменять, например, с помощью панели управления.
- Специальный маркер M8002 включен только после запуска контроллера на время одного программного цикла. Таким образом, после включения контроллера в регистр данных D5, используемый в качестве памяти заданного значения для счетчика C0, записывается константа "34".

Заданные значения в программе контроллера не обязательно должны вноситься в регистры данных. Их можно также устанавливать с помощью программатора перед запуском программы.



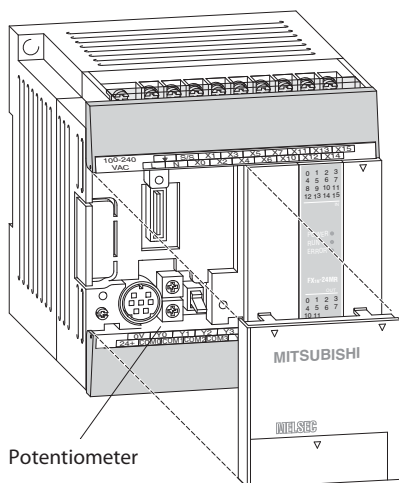
ВНИМАНИЕ:

Если заданные значения не вносятся в регистры программой контроллера, для сохранения заданных значений таймеров и счетчиков используйте фиксируемый регистр данных. Учитывайте, что содержимое этих регистров утрачивается, если батарейка буферного питания села.

Если используются обычные регистры, то заданные значения стираются, как только выключается напряжение питания или выключатель RUN/STOP переключается в положение STOP. После включения напряжения или очередного запуска контроллера эти заданные значения, которые теперь равны "0", могут привести к опасным состояниям.

Задание значений с помощью встроенных потенциометров

В контроллерах серий FX1S, FX1N и FX3G заданные значения (например, время) можно быстро и просто изменять с помощью двух аналоговых потенциометров контроллера.

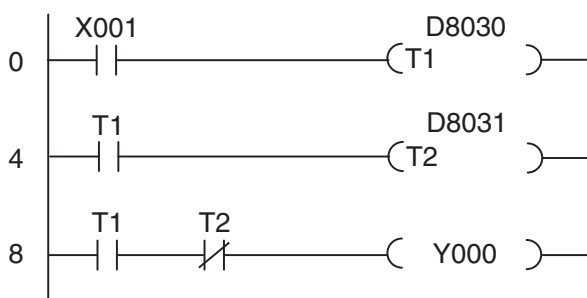


На рисунке слева показан базовый блок серии FX1N. У базовых блоков серий FX1S и FX3G потенциометры расположены так же.

Значение верхнего потенциометра VR1 можно считать из специального регистра D8030. В D8031 хранится значение потенциометра VR2, т. е. нижнего потенциометра. Чтобы в качестве источника заданного значения для датчика времени применялся потенциометр, в программе вместо константы указывается регистр.

В зависимости от положения потенциометра, значение регистра может изменяться от 0 до 255.

Релейная диаграмма

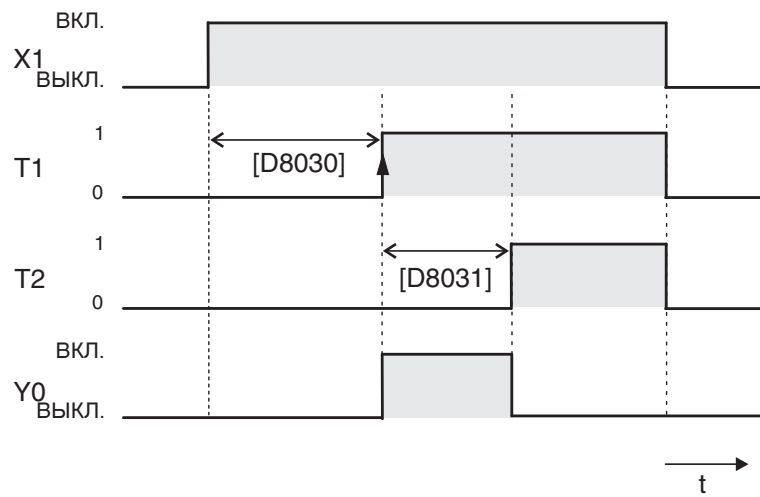


Список инструкций

| | | | |
|----|-----|------|-------|
| 0 | LD | X001 | |
| 1 | OUT | T1 | D8030 |
| 4 | LD | T1 | |
| 5 | OUT | T2 | D8031 |
| 8 | LD | T1 | |
| 8 | ANI | T2 | |
| 10 | OUT | Y000 | |

В вышеприведенном примере программы по истечении T1 выход Y0 включается на время, определяемое T2 (задержка вывода импульсов).

Диаграмма изменения сигналов

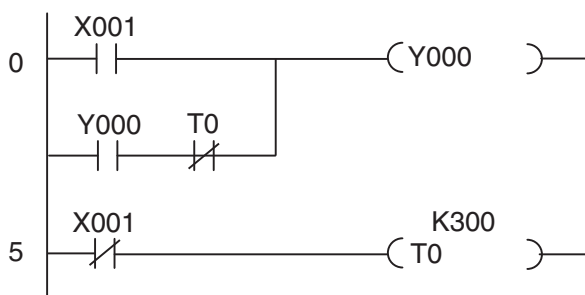


4.6.2 Задержка выключения

Все таймеры контроллеров MELSEC действуют как задержка включения. После истечения заданного времени включается выход таймера. Однако часто бывают нужны задержки выключения. (Пример из практики - управление вентилятором, который должен оставаться включенным еще несколько минут после выключения освещения ванной комнаты).

Вариант 1 программы (самоудержание)

Релейная диаграмма

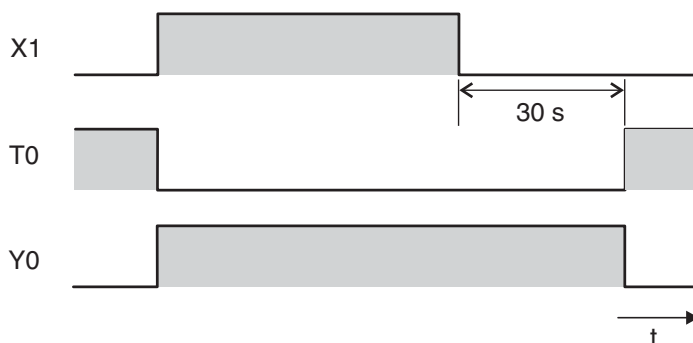


Список инструкций

| | | | |
|---|-----|------|------|
| 0 | LD | | X001 |
| 1 | LD | | Y000 |
| 2 | ANI | T0 | |
| 3 | ORB | | |
| 4 | OUT | Y000 | |
| 5 | LDI | X001 | |
| 6 | OUT | T0 | K300 |

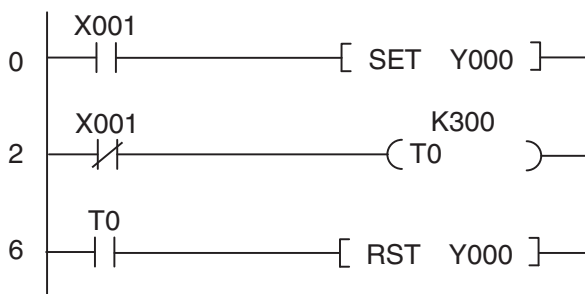
До тех пор, пока вход X1 (например, выключатель освещения) включен, включен также выход Y0 (вентилятор). Но и после выключения X1 выход Y0 остается включенным благодаря самоудержанию, так как таймер T0 еще не истек. Этот таймер запускается в результате выключения X1. По истечении настроенного времени (в примере: $300 \times 0,1 \text{ c} = 30 \text{ c}$) T0 прерывает самоудержание Y0 и этот выход выключается.

Диаграмма изменения сигналов



Вариант 2 программы (установка/сброс)

Релейная диаграмма



Список инструкций

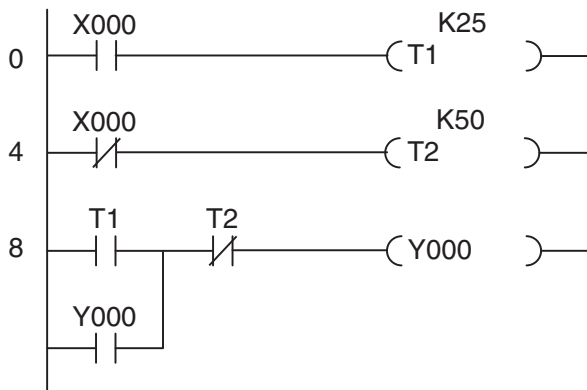
| | | | |
|---|-----|------|------|
| 0 | LD | X001 | |
| 1 | SET | Y000 | |
| 2 | LDI | X001 | |
| 3 | OUT | T0 | K300 |
| 6 | LD | T0 | |
| 7 | RST | Y000 | |

При включении X1 устанавливается (включается) выход Y0. При выключении X1 запускается T0. По истечении настроенного времени T0 сбрасывает выход Y0. Диаграмма изменения сигналов идентична той, которая показана для варианта 1.

4.6.3 Задержка включения и выключения

На практике может понадобиться, чтобы выход с задержкой включался и с задержкой выключался. Эту задачу тоже можно легко решить с помощью базовых логических операций.

Релейная диаграмма



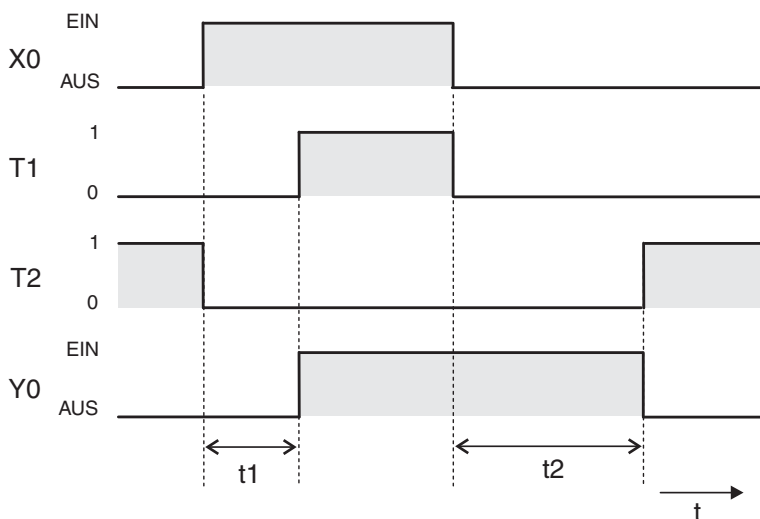
Список инструкций

```

0 LD X000
1 OUT T1 K25
4 LDI X000
5 OUT T2 K50
8 LD T1
9 OR Y000
10 ANI T2
11 OUT Y000

```

Диаграмма изменения сигналов



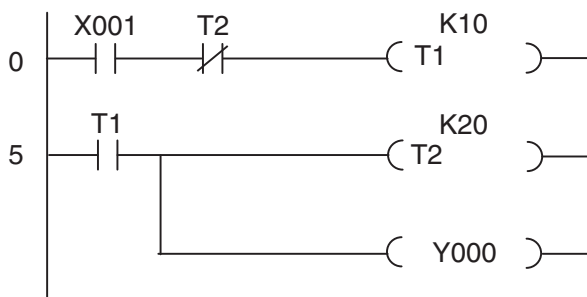
Благодаря самоудержанию Y000 с помощью T1 выход остается включенным на протяжении задержки выключения.

4.6.4 Датчик тактовых импульсов

В контроллере имеются специальные маркеры, с помощью которых можно очень просто решать задачи программирования, требующие неизменного такта (например, для управления сигнальным фонарем). Маркер M8013, например, включается и выключается с 1-секундным тактом. Более подробное описание всех специальных маркеров имеется в руководстве по программированию контроллеров семейства FX, артикул 136748.

Если, однако, нужно иное время такта или различные значения времени включения и выключения, датчик тактовых импульсов можно реализовать на основе двух таймеров.

Релейная диаграмма



Список инструкций

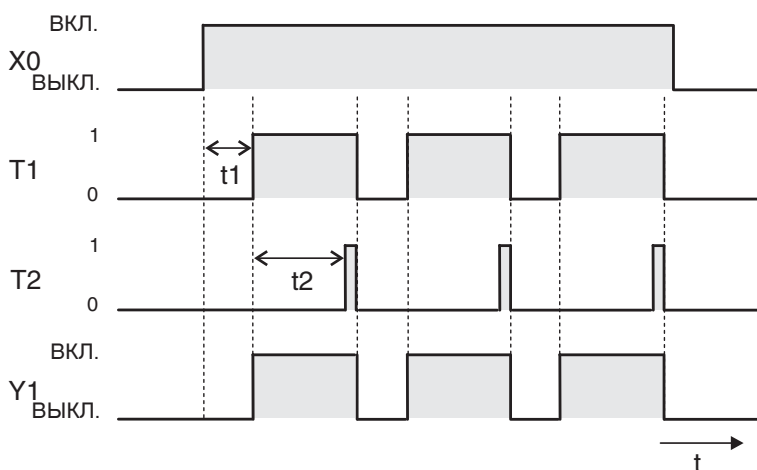
```

0 LD X001
1 ANI T2
2 OUT T1 K10
5 LD T1
6 OUT T2 K20
9 OUT Y000
    
```

X1 запускает датчик тактовых импульсов. Этот вход можно и не использовать. После этого датчик тактовых импульсов постоянно активирован. При дальнейшем выполнении программы обрабатывается выход T1, например, для сигнальных ламп. Время включения зависит от T2, а время выключения - от T1.

Выход таймера T2 включается только на время одного программного цикла. На следующей иллюстрации, показывающей диаграмму изменения сигналов в примере программы, это время изображено преувеличенно долгим. T2 выключает T1, и в результате сразу после этого выключается и T2. Точнее говоря, таким способом время включения удлиняется на время, необходимое для выполнения программы. Но так как время цикла составляет лишь несколько миллисекунд, им, как правило, можно пренебречь.

Диаграмма изменения сигналов



5 Продвинутое программирование

Стандартные логические инструкции, описанные в разделе 3, позволяют контроллеру имитировать функции релейных систем управления. Однако на этом возможности контроллера еще далеко не исчерпаны. Так как центральным элементом любого контроллера является микропроцессор, для него никакой проблемы не представляют и вычисления, сравнения чисел, преобразования систем счисления или обработка аналоговых значений.

Для выполнения этих функций, выходящих за рамки логических операций, нужны особые команды - так называемые прикладные команды.

5.1 Обзор прикладных команд

Прикладные команды однозначно обозначаются сокращением, описывающим их функцию. Например, команда, с помощью которой можно сравнить два 16-битовых или 32-битовых числа, обозначается сокращением "CMP" (от английского слова "to compare", "сравнивать"). Все сокращенные обозначения прикладных команд основываются на английских словах.

При программировании указывается сокращенное обозначение, за которым следует один или несколько операндов. В следующей таблице приведен обзор всех прикладных команд, чтобы наглядно продемонстрировать возможности контроллеров MELSEC семейства FX. Не пугайтесь, вам не требуется запоминать все сокращения. При программировании вы можете пользоваться справочной функцией среды программирования GX Developer или GX IEC Developer. Все команды подробно описаны в руководстве по программированию контроллеров семейства FX, артикул 136748, версия D, и снабжены примерами. Поэтому в этой главе рассматриваются только наиболее часто используемые команды (эти команды в таблице изображены на сером фоне).

| Подгруппа | Команда | Значение | Контроллер | | | | |
|------------------------------|-------------|-------------------------------------|------------|------|---------------|------|---------------|
| | | | FX1S | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC |
| Команды выполнения программы | CJ | Переход внутри программы | | | | | |
| | CALL | Вызов подпрограммы | | | | | |
| | SRET | Конец подпрограммы | | | | | |
| | IRET | Завершить программу прерывания | | | | | |
| | EI | Активировать программу прерывания | ● | ● | ● | ● | ● |
| | DI | Деактивировать программу прерывания | ● | ● | ● | ● | ● |
| | FEND | Конец области программы | | | | | |
| | WDT | Актуализировать сторожевой таймер | | | | | |
| | FOR | Начало повторения программы | | | | | |
| | NEXT | Конец повторения программы | | | | | |
| Команды сравнения и передачи | CMP | Сравнить числовые данные | ● | ● | ● | ● | ● |
| | ZCP | Сравнить числовых области данных | ● | ● | ● | ● | ● |
| | MOV | Передача данных | ● | ● | ● | ● | ● |
| | SMOV | Передача со сдвигом | | | ● | ● | ● |
| | CML | Копировать и инвертировать | | | ● | ● | ● |
| | BMOV | Передача блока | ● | ● | ● | ● | ● |
| | FMOV | Передача одинаковых данных | | | ● | ● | ● |
| | XCH | обмен данными | | | ● | | ● |
| | BCD | Конвертация BCD | ● | ● | ● | ● | ● |
| | BIN | Двоичная конвертация | ● | ● | ● | ● | ● |

| Подгруппа | Команда | Значение | Контроллер | | | | |
|--------------------------------------|--------------|---|------------|------|---------------|------|---------------|
| | | | FX1S | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC |
| Арифметические и логические операции | ADD | Сложение числовых данных | ● | ● | ● | ● | ● |
| | SUB | Вычитание числовых данных | ● | ● | ● | ● | ● |
| | MUL | Умножение числовых данных | ● | ● | ● | ● | ● |
| | DIV | Деление числовых данных | ● | ● | ● | ● | ● |
| | INC | Приращение | ● | ● | ● | ● | ● |
| | DEC | Отрицательное приращение | ● | ● | ● | ● | ● |
| | WAND | Логическая операция И | ● | ● | ● | ● | ● |
| | WOR | Логическая операция ИЛИ | ● | ● | ● | ● | ● |
| | WXOR | Логическая операция "Исключающее ИЛИ" | ● | ● | ● | ● | ● |
| | NEG | Инверсия данных | | | ● | | ● |
| Команды сдвига | ROR | Вращение вправо | | | ● | ● | ● |
| | ROL | Вращение влево | | | ● | ● | ● |
| | RCR | Вращение битов вправо | | | ● | | ● |
| | RCL | Вращение битов влево | | | ● | | ● |
| | SFTR | Побитный сдвиг двоичных данных вправо | ● | ● | ● | ● | ● |
| | SFTL | Побитный сдвиг двоичных данных влево | ● | ● | ● | ● | ● |
| | WSFR | Пословный сдвиг данных вправо | | | ● | ● | ● |
| | WSFL | Пословный сдвиг данных влево | | | ● | ● | ● |
| | SFWR | Запись в память обратного магазинного типа | ● | ● | ● | ● | ● |
| | SFRD | Считывание из памяти обратного магазинного типа | ● | ● | ● | ● | ● |
| Операции с данными | ZRST | Сброс областей операндов | ● | ● | ● | ● | ● |
| | DECO | Декодировать данные | ● | ● | ● | ● | ● |
| | ENCO | Кодировать данные | ● | ● | ● | ● | ● |
| | SUM | Определение установленных битов | | | ● | ● | ● |
| | BON | Проверить биты | | | ● | ● | ● |
| | MEAN | Определение средних значений | | | ● | ● | ● |
| | ANS | Запуск интервала времени | | | ● | ● | ● |
| | ANR | Сброс битов индикации | | | ● | ● | ● |
| | SQR | Определение квадратного корня | | | ● | | ● |
| | FLT | Преобразование формата числа | | | ● | ● | ● |
| Высокоскоростные команды | REF | Обновить входы и выходы | ● | ● | ● | ● | ● |
| | REFF | Настройка входных фильтров | | | ● | | ● |
| | MTR | Считывание матрицы (MTR) | | | ● | ● | ● |
| | DHSCS | Установка с помощью высокоскоростного счетчика | ● | ● | ● | ● | ● |
| | DHSCR | Сброс с помощью высокоскоростного счетчика | ● | ● | ● | ● | ● |
| | DHSZ | Сравнение областей | | | ● | ● | ● |
| | SPD | Распознавание скорости | ● | ● | ● | ● | ● |
| | PLSY | Вывод импульсов (частота) | ● | ● | ● | ● | ● |
| | PWM | Вывод импульсов с широтно-импульсной модуляцией | ● | ● | ● | ● | ● |
| | PLSR | Вывод импульсов (количество) | ● | ● | ● | ● | ● |
| Прикладные команды | IST | Инициализировать состояние шага | ● | ● | ● | ● | ● |
| | SER | Команда поиска | | | ● | ● | ● |
| | ABSD | Абсолютное сравнение счетчика | ● | ● | ● | ● | ● |
| | INCD | Инкрементное сравнение счетчика | ● | ● | ● | ● | ● |
| | TTMR | Таймер заучивания | | | ● | | ● |
| | STMR | Специальный таймер | | | ● | | ● |
| | ALT | Триггерная функция | ● | ● | ● | ● | ● |
| | RAMP | Линейно нарастающая функция | ● | ● | ● | ● | ● |
| | ROTC | Позиционирование круглого стола | | | ● | | ● |
| | SORT | Команда сортировки | | | ● | | ● |


| Подгруппа | Команда | Значение | Контроллер | | | | |
|---|---|--|------------|------|---------------|------|---------------|
| | | | FX1S | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC |
| Команды ввода-вывода | TKY | Десятичная клавиатура | | | ● | | ● |
| | HKY | Шестнадцатиричная клавиатура | | | ● | | ● |
| | DSW | Цифровой выключатель | ● | ● | ● | ● | ● |
| | SEGD | 7-сегментная индикация | | | ● | | ● |
| | SEGL | 7-сегментная индикация с фиксацией | ● | ● | ● | ● | ● |
| | ARWS | 7-сегментная индикация с дополнительными клавишами | | | ● | | ● |
| | ASC | ASCII-конвертация | | | ● | | ● |
| | PR | Вывод данных через выходы | | | ● | | ● |
| | FROM | Считывание данных из специального модуля | | ● | ● | ● | ● |
| | TO | Запись данных в специальный модуль | | ● | ● | ● | ● |
| Команды последовательной коммуникации | RS | Последовательная передача данных | ● | ● | ● | ● | ● |
| | PRUN | Перенос входов или маркеров | ● | ● | ● | ● | ● |
| | ASCI | Преобразование в знак ASCII | ● | ● | ● | ● | ● |
| | HEX | Преобразование в шестнадцатиричное значение | ● | ● | ● | ● | ● |
| | CCD | Контроль по сумме и по четности | ● | ● | ● | ● | ● |
| | VRRD | Считывание заданных значений FX□□-8AV-BD | ● | ● | ● | ● | ● |
| | VRSC | Считывание положения выключателей FX□□-8AV-BD | ● | ● | ● | ● | ● |
| | RS2 | Последовательная передача данных (2) | | | | ● | ● |
| | PID | Программирование контура регулирования | ● | ● | ● | ● | ● |
| Фиксация / восстановление индексных регистров | ZPUSH | Зафиксировать содержимое индексных регистров | | | | | ● |
| | ZPOP | Восстановить содержимое индексных регистров | | | | | ● |
| Операции над числами с плавающей запятой | DECMP | Сравнение чисел с плавающей запятой | | | ● | ● | ● |
| | DEZCP | Сравнение чисел с плавающей запятой (диапазон) | | | ● | | ● |
| | DEMOV | Передача чисел с плавающей запятой | | | | ● | ● |
| | DESTR | Преобразование числа с плавающей запятой в строковую переменную | | | | | ● |
| | DEVAL | Преобразование строковой переменной в число с плавающей запятой | | | | | ● |
| | DEBCD | Преобразование формата с плавающей запятой в научный формат числа | | | ● | | ● |
| | DEBIN | Преобразование научного формата числа в формат с плавающей запятой | | | ● | | ● |
| | DEADD | Сложение чисел с плавающей запятой | | | ● | ● | ● |
| | DESUB | Вычитание чисел с плавающей запятой | | | ● | ● | ● |
| | DEMUL | Умножение чисел с плавающей запятой | | | ● | ● | ● |
| | DEDIV | Деление чисел с плавающей запятой | | | ● | ● | ● |
| | DEXP | Представление числа с плавающей запятой в виде показателя степени по основанию e | | | | | ● |
| | DLOGE | Расчет натурального логарифма | | | | | ● |
| | DLOG10 | Расчет десятичного логарифма | | | | | ● |
| | DESQR | Извлечение квадратного корня из чисел с плавающей запятой | | | ● | ● | ● |
| | DENEG | Изменение арифметического знака чисел с плавающей запятой | | | | | ● |
| INT | Преобразование из формата с плавающей запятой в десятичный формат | | | ● | ● | ● | |
| Арифметические операции над числами с плавающей запятой | SIN | Вычисление синуса | | | ● | | ● |
| | COS | Вычисление косинуса | | | ● | | ● |
| | TAN | Вычисление тангенса | | | ● | | ● |
| | ASIN | Вычисление арксинуса | | | | | ● |
| | ACOS | Вычисление арккосинуса | | | | | ● |
| | ATAN | Вычисление арктангенса | | | | | ● |
| | RAD | Пересчет градусов в радианы | | | | | ● |
| | DEG | Пересчет радианов в градусы | | | | | ● |

| Подгруппа | Команда | Значение | Контроллер | | | | |
|--|----------------------|--|------------|------|---------------|------|---------------|
| | | | FX1S | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC |
| Команды обработки данных | WSUM | Образовать сумму содержимого словных операндов | | | | | ● |
| | WTOB | Разделить данные словных операндов на байты | | | | | ● |
| | BTOW | Образовать словные операнды из отдельных байтов | | | | | ● |
| | UNI | Объединить группы по 4 бита в словные операнды | | | | | ● |
| | DIS | Разделить словные операнды на группы по 4 бита | | | | | ● |
| | SWAP | Поменять местами младший и старший байт | | | ● | | ● |
| | SORT2 | Сортировать данные в таблице | | | | | ● |
| Команды позиционирования | DSZR | Движение в точку отсчета (с бесконтактным выключателем) | | | | ● | ● |
| | DVIT | Позиционирование с помощью прерывания | | | | | ● |
| | TBL | Позиционирование по таблице данных | | | | ● | ● |
| | DABS | Считать абсолютное фактическое положение | ● | ● | ● | ● | ● |
| | ZRN | Движение в точку отсчета | ● | ● | | ● | ● |
| | PLSV | Вывод импульсов с переменной частотой | ● | ● | | ● | ● |
| | DRVI | Позиционирование на значение приращения | ● | ● | | ● | ● |
| Операции с встроенными часами контроллера | DRVA | Позиционирование на абсолютную величину | ● | ● | | ● | ● |
| | TCMP | Сравнение данных часов | ● | ● | ● | ● | ● |
| | TZCP | Сравнение данных часов с диапазоном | ● | ● | ● | ● | ● |
| | TADD | Сложение данных часов | ● | ● | ● | ● | ● |
| | TSUB | Вычитание данных часов | ● | ● | ● | ● | ● |
| | HTOS | Преобразовывать формат "часы, минуты, секунды" в секунды | | | | | ● |
| | STOH | Преобразовывать секунды в формат "часы, минуты, секунды" | | | | | ● |
| | TRD | Считать время суток и дату | ● | ● | ● | ● | ● |
| | TWR | Передать время суток и дату в контроллер | ● | ● | ● | ● | ● |
| Преобразование кода Грея | HOUR | Счетчик часов работы | ● | ● | ● | ● | ● |
| | GRY | Преобразовать код Грея в десятичное число | | | ● | ● | ● |
| Обмен данными с аналоговыми модулями | GBIN | Преобразовать десятичное число в код Грея | | | ● | ● | ● |
| | RD3A | Считать аналоговые значения входов | | ● | ● | ● | ● |
| Команда из внешней памяти | WR3A | Записать аналоговое значение выхода | | ● | ● | ● | ● |
| | EXTR | Выполнить команду, сохраненную во внешней ROM | | | ● | | |
| Различные команды | COMRD | Считать комментарий к операнду | | | | | |
| | RND | Генерировать случайное число | | | | | |
| | DUTY | Выдать импульс определенной длины | | | | | ● |
| | CRC | Проверить данные (проверка CRC) | | | | | |
| | HCMOV | Передать фактическое значение высокоскоростного счетчика | | | | | |
| Команды для данных, сохраненных в следующих друг за другом операндах (блоках данных) | BK+ | Добавить данные в блок данных | | | | | |
| | BK- | Изъять данные из блока данных | | | | | |
| | BKCMPE= | Сравнить данные в блоках данных | | | | | |
| | BKCMP> | | | | | | |
| | BKCMP< | | | | | | |
| | BKCMP<> | | | | | | |
| | BKCMP<= | | | | | | |
| BKCMP>= | | | | | | | |
| | | | | | | ● | |

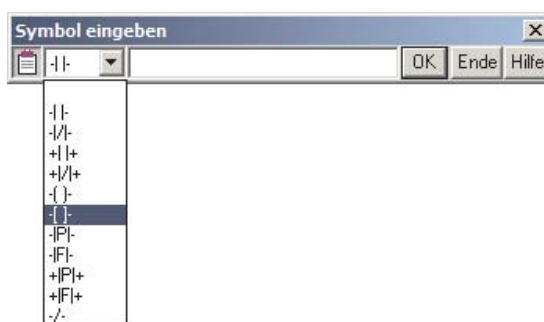
| Подгруппа | Команда | Значение | Контроллер | | | | | |
|--|---------------------------|--|------------|------|---------------|------|---------------|---|
| | | | FX1s | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC | |
| Команды обработки строковых переменных | STR | Преобразовывать двоичные данные в строковую переменную | | | | | | |
| | VAL | Преобразовывать строковую переменную в двоичные данные | | | | | | |
| | \$+ | Соединить строковые переменные | | | | | | |
| | LEN | Определить длину строковой переменной | | | | | | |
| | RIGHT | Фрагмент строковой переменной справа | | | | | ● | |
| | LEFT | Фрагмент строковой переменной слева | | | | | | |
| | MIDR | Выбрать фрагмент строковой переменной | | | | | | |
| | MIDW | Заменить цепочку символов | | | | | | |
| | INSTR | Искать цепочку символов | | | | | | |
| \$MOV | Передать цепочку символов | | | | | | | |
| Команды обработки перечней данных | FDEL | Стереть данные из перечня данных | | | | | | |
| | FINS | Ввести данные в перечень данных | | | | | | |
| | POP | Считать данные, внесенные в перечень данных последними | | | | | ● | |
| | SFR | Сдвинуть 16-битовое слово данных вправо | | | | | | |
| | SFL | Сдвинуть 16-битовое слово данных влево | | | | | | |
| Команды обработки перечней данных | LD= | Сравнение данных в рамках логических операций | | | | | | |
| | LD> | | | | | | | |
| | LD< | | | | | | | |
| | LD<> | | | | | | | |
| | LD<= | | | | | | | |
| | LD>= | | | | | | | |
| | AND= | | | | | | | |
| | AND> | | | ● | ● | ● | ● | ● |
| | AND< | | | | | | | |
| | AND>= | | | | | | | |
| | OR= | | | | | | | |
| | OR> | | | | | | | |
| | OR< | | | | | | | |
| OR<> | | | | | | | | |
| OR<= | | | | | | | | |
| OR>= | | | | | | | | |
| Команды контроля данных | LIMIT | Ограничить диапазон вывода значений | | | | | | |
| | BAND | Установить смещение входа | | | | | | |
| | ZONE | Установить смещение выхода | | | | | | |
| | SCL | Масштабировать значения | | | | | | |
| | DABIN | Преобразовывать число в коде ASCII в двоичное значение | | | | | ● | |
| | BINDA | Преобразовывать двоичное значение в код ASCII | | | | | | |
| | SCL2 | Масштабировать значения (таблица значений имеет иную структуру, чем у команды SCL) | | | | | | |
| Команды для коммуникации с преобразователями частоты | IVCK | Проверить состояние преобразователя частоты | | | | | | |
| | IVDR | Управление преобразователем частоты | | | | | | |
| | IVRD | Считать параметр преобразователя частоты | | | | ● | ● | |
| | IVWR | Записать параметр в преобразователь частоты | | | | | | |
| | IVBWR | Поблочко записать параметры в преобразователь частоты | | | | | ● | |
| Обмен данными со специальными модулями | RBFM | Считать буферную память специальных модулей | | | | | ● | |
| | WBFM | Записать в буферную память специальных модулей | | | | | ● | |
| Команда для высокоскоростного счетчика | HSCT | Сравнить фактическое значение высокоскоростного счетчика с данными в перечнях данных | | | | | ● | |

| Подгруппа | Команда | Значение | Контроллер | | | | |
|--|---------------|---|------------|------|---------------|------|---------------|
| | | | FX1S | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC |
| Команды для расширенных регистров файлов | LOADR | Считать данные из расширенного регистра файлов | | | | ● | ● |
| | SAVER | Записать данные в расширенные регистры файлов | | | | | ● |
| | INITR | Инициализировать расширенный регистр и расширенный регистр файлов | | | | | ● |
| | LOGR | Сохранить значения операндов в расширенном регистре или расширенном регистре файлов | | | | | ● |
| | RWER | Передать данные из расширенного регистра в расширенные регистры файлов | | | | ● | ● |
| | INITER | Инициализировать расширенный регистр файлов | | | | | ● |

5.1.1 Ввод прикладных инструкций

Для ввода прикладных команд в среде программирования GX Developer FX поместите курсор на то место в цепи тока, в котором требуется ввести команду, и введите краткое обозначение команды и операнды. Среда программирования автоматически распознает, что вводится команда, и открывает окно для ввода (см. ниже). Или поместите курсор и щелкните в перечне инструментов по символу .

Команду можно выбрать и в окне для ввода. Чтобы открыть меню, щелкните по символу "▼".



После этого в поле ввода вводится краткое обозначение команды и операнды. Эти вводы отделяются друг от друга знаком пробела.

Перед всеми цифрами должна стоять буква, которая указывает либо тип операнда, либо (в случае констант) формат числа. Буква "K" обозначает десятичные, а буква "H" - шестнадцатичные константы.



В этом примере с помощью команды MOV в регистр данных D12 вводится значение "5".

С помощью клавиши "Помощь" вы можете открыть окно диалога и найти команду с требуемой функцией. Там же вы можете найти и информацию о принципе действия команды, а также о типе и количестве операндов.

После щелчка по "OK" прикладная команда перенимается в программу.



Если вы программируете на языке инструкций, введите в строке сокращенное название команды, а за ним операнды. Отдельные вводы и здесь отделяются друг от друга знаком пробела.

5.2 Команды передачи данных

Регистры данных в контроллере служат в качестве запоминающего устройства для результатов измерений, выводимых величин, промежуточных результатов или табличных значений. Несмотря на то, что арифметические операции, например, считывают свои значения операндов непосредственно из регистров данных и вводят результаты там же (если это требуется), для поддержки этих команд нужны команды передачи, с помощью которых можно копировать данные из одного регистра в другой или заносить константы в регистр данных.

5.2.1 Передача отдельных данных с помощью команды MOV

С помощью команды MOV (от англ. “to move” = двигать) данные “перемещаются” и копируются из источника данных в место назначения.

Релейная диаграмма

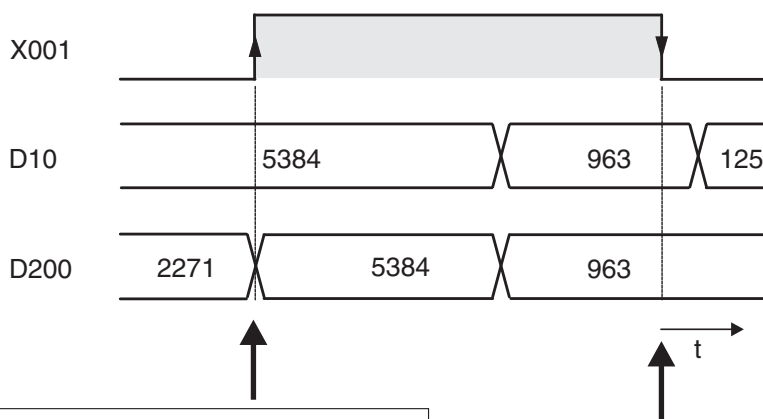
Список инструкций



① источник данных (здесь можно также указать константу)

② место назначения

В этом примере содержимое регистра данных D10 передается в регистр данных D200, если вход X1 включен. На рисунке ниже показана диаграмма изменения сигналов для этого примера.



До тех пор, пока входное условие команды MOV выполнено, содержимое источника данных переносится в место назначения. Содержимое источника данных в результате переноса не изменяется.

Если входное условие более не выполняется, содержимое места назначения в результате этой команды более не изменяется.

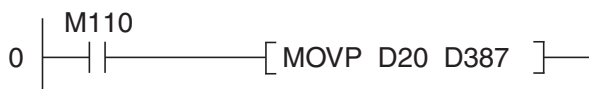
Выполнение команды MOV в зависимости от фронта сигнала

В некоторых приложениях бывает выгоднее, если место назначения описывается только в одном программном цикле. Например, если в другом месте программы происходит передача в то же самое место назначения, или если передача должна происходить только в определенный момент.

Если к обозначению команды MOV добавить букву “P”, то команда MOV выполняется только один раз, при восходящем фронте условия входа. (Буква “P” происходит от английского слова “pulse” и указывает на то, что команда управляется переменной сигнала, т. е. импульсом.)

В следующем примере содержимое регистра D20 вносится в регистр данных D387 только в том случае, если состояние сигнала M11 изменяется с "0" на "1".

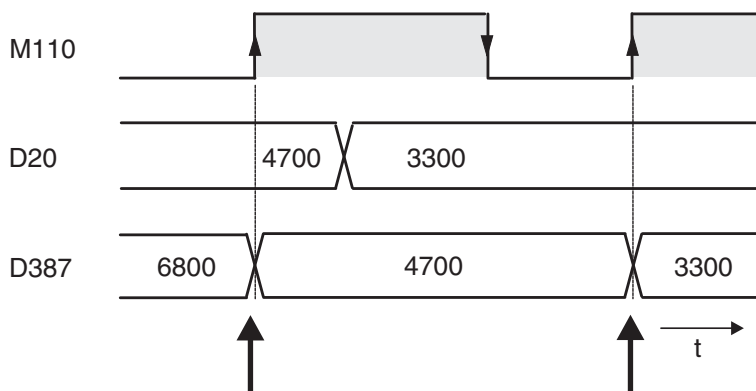
Релейная диаграмма



Список инструкций

```
0 LD      M110
1 MOV    D20  D387
```

Даже если M110 остается установленным, передача в регистр D387 более не происходит. Это можно наглядно пояснить диаграммой изменения сигналов для этого примера:



Содержимое источника данных переносится в место назначения только при возрастающем фронте условия входа.

Перенос 32-битовых данных

Если с помощью команды MOV требуется передать 32-битовые данные, то перед сокращенным названием команды ставится буква "D".

Релейная диаграмма



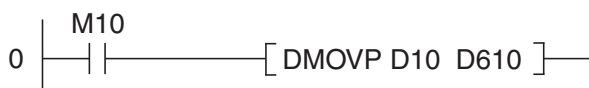
Релейная диаграмма

```
0 LD      X010
1 DMOV   C200  D40
```

Если вход X010 включен, состояние 32-битового счетчика C200 передается в регистры данных D40 и D41. Регистр D40 содержит младшие биты.

Возможна и комбинация обработки двойного слова с управлением по фронту сигнала, как показывает следующий пример.

Релейная диаграмма



Релейная диаграмма

```
0 LD      M10
1 DMOVP  D10  D610
```

При установке маркера M10 содержимое регистров D10 и D11 передается в регистры D619 и D611.

5.2.2 Групповая передача битовых операндов

В предыдущем разделе было показано, как с помощью команды MOV можно передавать константы или содержимое регистров данных в другие регистры данных. Но числовые значения можно сохранять и в следующих друг за другом битовых операндах, например, маркерах. Чтобы адресовать прикладную команду сразу несколькими следующими друг за другом битовым операндам, указывается адрес первого битового операнда и коэффициент "K", означающий количество операндов.

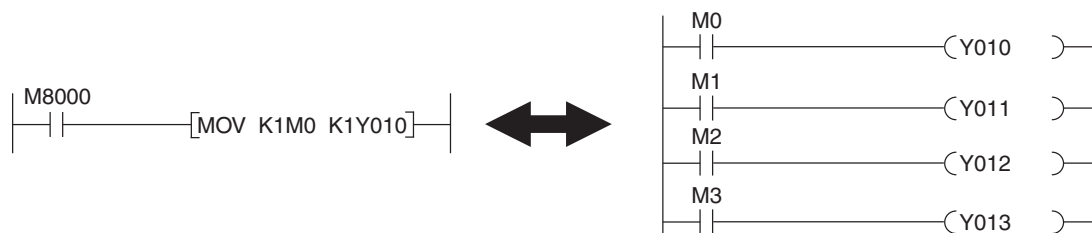
Этот коэффициент "K" указывает количество блоков по 4 операнда: K1 = 4 операнда, K2 = 8 операндов, K3 = 12 операндов и т. д.

Например, указание "K2M0" означает восемь маркеров с M0 по M7. Возможны коэффициенты от K1 (4 операнда) до K8 (32 операнда).

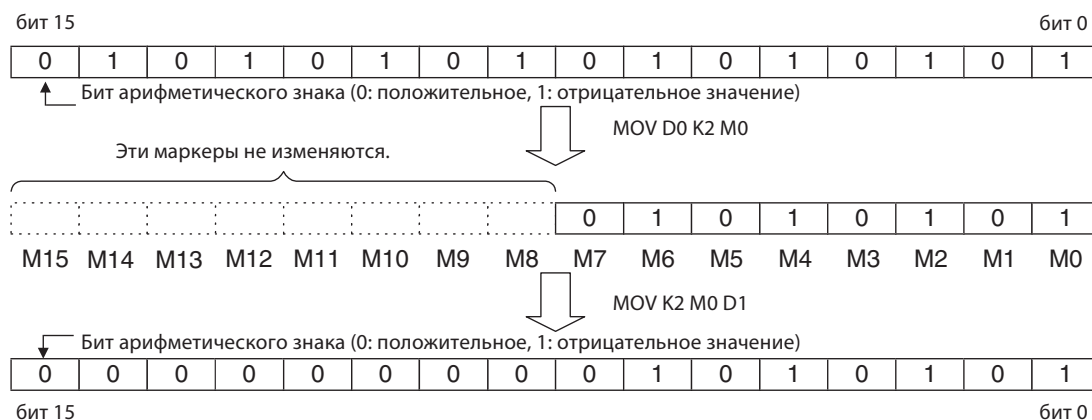
Примеры указания битовых операндов

- K1X0: 4 входа, начиная с X0 (с X0 по X3)
- K2X4: 8 входов, начиная с X4 (с X4 по X13, восьмеричный принцип счета!)
- K4M16: 16 маркеров, начиная с M16 (с M16 по M31)
- K3Y0: 12 выходов, начиная с Y0 (с Y0 по X13, восьмеричный принцип счета!)
- K8M0: 32 маркера, начиная с M0 (с M0 по M31)

Возможность обращаться одной командой сразу к нескольким битовым операндам уменьшает затраты на программирование. Оба следующих фрагмента программы выполняют одну и ту же функцию: передачу состояния маркеров с M0 по M4 выходам с Y10 по Y14.



Если место назначения меньше источника данных, избыточные биты не передаются (см. следующую иллюстрацию, верхний пример). Если место назначения больше источника, отсутствующие места заполняются нулями. В связи с тем, что бит 15 интерпретируется как арифметический знак, образующееся значение всегда положительно. (Как в нижнем примере на следующей иллюстрации.)



5.2.3 Передача взаимосвязанных данных командой BMOV

С помощью команды MOV, представленной в разделе 5.2.1, можно передать в место назначения максимум одно 16- или 32-битовое значение. Для передачи взаимосвязанных данных пришлось бы последовательно программировать несколько команд MOV. Чтобы избавить вас от таких затрат времени на программирование, предусмотрена команда BMOV. Ее сокращение расшифровывается как "Block Move" ("передача блока"): операнды передаются взаимосвязанными, в виде блока.

Релейная диаграмма

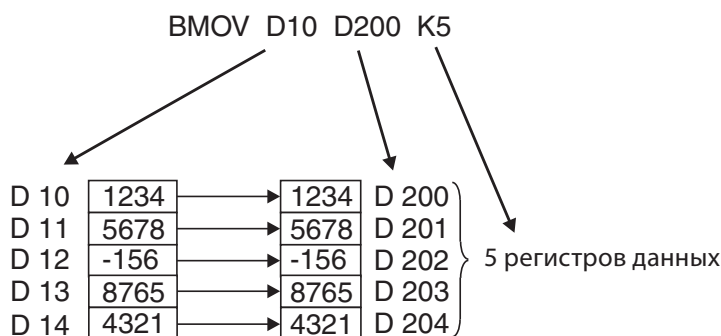


Список инструкций

0 BMOV D10 D200 K5
1 2 3

- ① источник данных (16-битовый операнд, указывается 1-й операнд области-источника)
- ② место назначения (16-битовый операнд, указывается 1-й операнд целевой области)
- ③ количество передаваемых элементов (макс. 512)

С вышеуказанными операндами образуется следующая функция:



Команду BMOV тоже можно выполнять в зависимости от фронта сигнала. В этом случае она записывается как команда BMOV_P (см. раздел 5.2.1).

Если с помощью команды BMOV требуется передать группы битовых операндов, коэффициенты "K" источника данных и места назначения должны быть идентичны.

Пример

BMOV K1M0 K1Y0 K2



5.2.4 Передача одних и тех же данных в несколько целевых операндов (FMOV)

С помощью команды FMOV содержимое словных или двойных словных операндов или констант вносится в несколько следующих друг за другом словных или двойных словных операндов. Таким способом можно, например, стирать таблицы данных или возвращать регистры данных на определенные начальные значения.

Релейная диаграмма

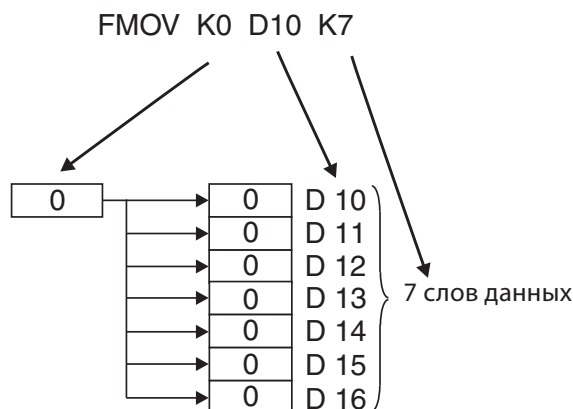


Список инструкций



- ① данные, которые требуется внести в целевые операнды; возможно также указание констант
- ② место назначения (указывается 1-й операнд целевой области)
- ③ количество описываемых элементов целевой области (макс. 512)

В следующем примере в 7 элементов вводится значение "0":



Если команда FMOV используется в виде команды FMOV P, передача данных происходит в зависимости от фронта сигнала (см. описание команды MOV в разделе 5.2.1).

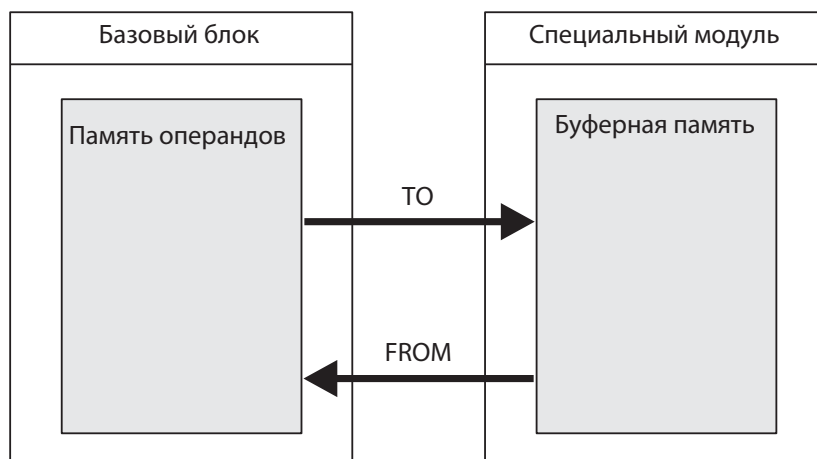
Если требуется передать 32-битовые данные, название команды дополняется спереди буквой "D" (DFMOV или DFMOV P).

5.2.5 Обмен данными со специальными модулями

За исключением серии FX1S, количество дискретных входов и выходов всех базовых блоков MELSEC семейства FX можно увеличить путем подключения модулей расширения. Кроме того, установив так называемые специальные модули, можно еще более повысить объем выполняемых функций контроллера. Например, специальные модули определяют аналоговые значения (токи, напряжения), регулируют температуру или осуществляют коммуникацию с внешним оборудованием.

В то время как для дискретных модулей расширения не нужны никакие особые команды (дополнительные входы и выходы обрабатываются точно так же, как входы и выходы базового блока), для обмена данными между базовым блоком и специальным модулем используются две прикладные команды: FROM и TO ("от" и "к").

В специальном модуле имеется область памяти, в которой, например, промежуточно хранятся (буферизуются) аналоговые измеренные значения или принятые данные. В связи с таким использованием эта область памяти обозначается как "буферная память". К буферной памяти в специальном модуле может получить доступ и базовый блок, например, чтобы считать измеренные значения или принятые данные, или чтобы внести данные, которые специальный модуль затем обрабатывает дальше (настройки для функционирования специального модуля, передаваемые данные и т. п.).



Буферная память может содержать до 32767 отдельных ячеек. Каждый из этих адресов буферной памяти может хранить 16 бит информации. Функционирование адреса буферной памяти зависит от типа специального модуля. Ее можно узнать из руководств по отдельным специальным модулям.

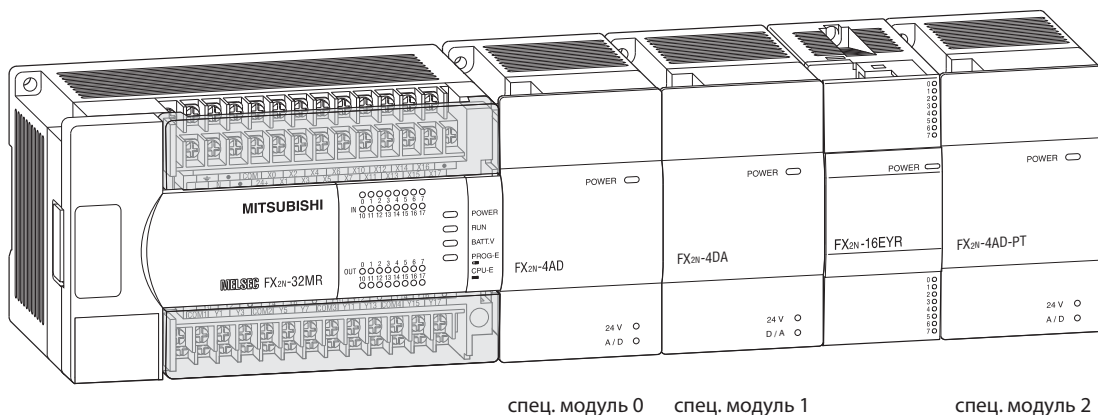
| |
|-----------------------------|
| адрес 0 в буферной памяти |
| адрес 1 в буферной памяти |
| адрес 2 в буферной памяти |
| : |
| : |
| адрес n-1 в буферной памяти |
| адрес n в буферной памяти |

Для правильного функционирования команда FROM или TO нуждается в определенных данных:

- Из какого специального модуля должны быть считаны данные или в какой специальный модуль данные должны быть переданы?
- Каков первый адрес в буферной памяти, из которой требуется считать данные или в которую требуется записать данные?
- Из скольких адресов буферной памяти данные требуется считать или во сколько адресов данные требуется записать?
- Где в базовом блоке должны быть сохранены данные из буферной памяти или где хранятся данные, которые требуется передать в специальный модуль?

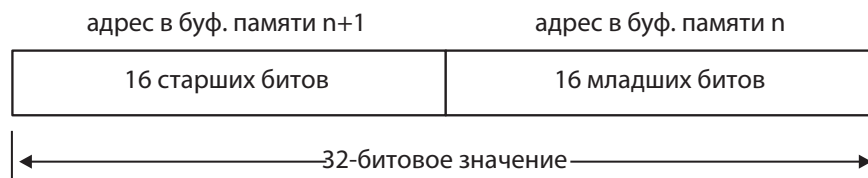
Адрес специального модуля

Если имеются несколько специальных модулей, то чтобы передать или считать данные из правильного модуля, модули необходимо особо обозначить. Для этого каждый специальный модуль автоматически получает номер из диапазона от 0 до 7 (к контроллеру можно подключить максимум 8 специальных модулей). Номера присваиваются непрерывно, и нумерация начинается с модуля, который первым подключается к контроллеру.



Начальный адрес в буферной памяти

Каждый из адресов буферной памяти (до 32767 адресов) можно указать в десятичном виде в диапазоне от 0 до 32766 (FX1N: от 0 до 31). 32-битовые данные вносятся в буферную память таким образом, чтобы ячейка памяти с более низким адресом содержала младшие 16 битов, а следующий адрес в буферной памяти - старшие 16 битов.

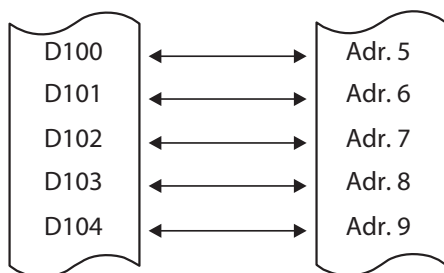


Поэтому в качестве начального адреса для 32-битовых данных необходимо всегда указывать адрес, содержащий 16 младших битов.

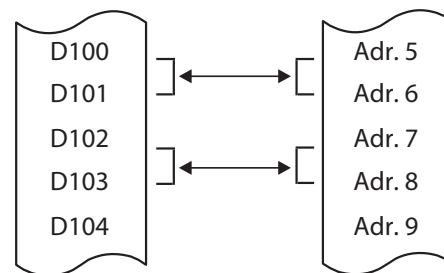
Количество передаваемых данных

Количество данных относится к единицам передаваемых данных. Если команда FROM или TO выполняется в качестве 16-битовой команды, то количество данных соответствует количеству передаваемых слов. В случае 32-битовой команды (DFROM или DTO) указывается количество передаваемых двойных слов.

16-битовая команда
Количество данных: 5



32-битовая команда
Количество данных: 2



Значение, которое можно задать для количества данных, зависит от того, какой контроллер используется и как выполняется команда FROM – в качестве 16-битовой или 32-битовой команды:

| Используемый контроллер | Допустимый диапазон для “количества передаваемых данных” | |
|-------------------------|--|---------------------------------|
| | 16-битовая команда (FROM, TO) | 32-битовая команда (DFROM, DTO) |
| FX2N | от 1 до 32 | от 1 до 16 |
| FX2NC | от 1 до 32 | от 1 до 16 |
| FX3U | от 1 до 32767 | от 1 до 16383 |

Место назначения или источник данных в базовом блоке

В большинстве случаев данные считываются из регистров и передаются в специальный модуль, или передаются из его буферной памяти в область регистров данных базового блока. Но в качестве места назначения или источника передачи данных могут служить и выходы, маркеры или фактические значения таймеров и счетчиков.

Выполнение команд в зависимости от фронта сигнала

Если к обозначению команды добавить букву “P”, то передача данных происходит в зависимости от фронта сигнала (см. описание команды MOV в разделе 5.2.1).

Подробное описание команды FROM

С помощью команды FROM данные передаются из буферной памяти специального модуля в базовый блок. При этом содержимое буферной памяти не изменяется, т. е. данные копируются.

Релейная диаграмма



Список инструкций

0 FROM K0 K9 D0 K1
① ② ③ ④

- ① адрес специального модуля (от 0 до 7)
- ② начальный адрес в буферной памяти (FX1N: от 0 до 31, FX2N, FX2NC и FX3U: от 0 до 32766). Для указания можно использовать константу или регистр данных, содержащий значение адреса.
- ③ место назначения в базовом блоке
- ④ количество передаваемых данных

В изображенном вверху примере из модуля аналого-цифрового преобразователя FX2N-4AD с адресом 0 фактическое значение канала 1 с адреса 9 буферной памяти передается в регистр данных D0.

В следующем примере для 32-битовой команды данные считываются из специального модуля с адресом 2. Начиная с адреса 8 буферной памяти считываются 4 двойных слова и сохраняются в базовом блоке в регистрах данных с D8 по D15.



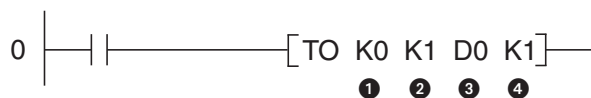
В последнем примере запрограммирована команда FROMP. Это означает, что содержимое четырех адресов буферной памяти с 0 по 3 вносятся в регистры данных с D10 по D13 только в том случае, если состояние условия входа изменяется с “0” на “1”.



Подробное описание команды TO

С помощью команды TO данные из базового блока передаются в буферную память специального модуля. Содержимое источника данных при этом процессе копирования не изменяется.

Релейная диаграмма



Список инструкций

```
0 TO      K0   K1   D0   K1
          ①   ②   ③   ④
```

- ① адрес специального модуля (с 0 по 7)
- ② начальный адрес в буферной памяти (FX1N: от 0 до 31, FX2N, FX2NC и FX3U: от 0 до 32766)
- ③ начальный адрес в буферной памяти (FX1N: от 0 до 31, FX2N, FX2NC и FX3U: от 0 до 32766)
- ④ количество передаваемых данных

В изображенном вверху примере содержимое регистра данных D0 передается по адресу 1 буферной памяти в специальном модуле с адресом 0.

5.3 Операторы сравнения

Чтобы проверить в программе состояние битовых операндов (например, входов или маркеров), достаточно использовать стандартные логические операции, так как эти операнды могут принимать только два состояния - "0" и "1". Однако часто в программе требуется проверить содержимое словных операндов и, в зависимости от результата проверки, выполнять определенное действие (например, включать охлаждающий вентилятор при превышении определенной температуры). Контроллеры MELSEC семейства FX имеют различные возможности сравнения данных.

5.3.1 Команда CMP

Команда CMP сравнивает два числовых значения. Эти значения могут быть константами или содержимым регистров данных. Но возможно также указывать и фактические значения таймеров или счетчиков. В зависимости от результата сравнения (больше, меньше или равно) включается один из трех битовых операндов.

Релейная диаграмма



Список инструкций

```
0 LD      ..①
1 CMP     D0   K100 M0
          ②   ③   ④
```

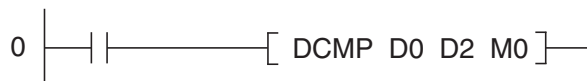
- ① Входное условие
- ② Первое сравниваемое значение
- ③ Второе сравниваемое значение
- ④ Первый из трех следующих друг за другом маркеров или выходов, включающихся в зависимости от результата сравнения (состояние "1").
 - 1-й операнд: ВКЛ., если значение 1 > значения 2
 - 2-й операнд: ВКЛ., если значение 1 = значению 2
 - 3-й операнд: ВКЛ., если значение 1 < значения 2

В этом примере команда CMP управляет маркерами M0, M1 и M2. Маркер M0 имеет состояние "1", если содержимое D0 больше 100, маркер M1 имеет состояние "1", если содержимое D0 точно равно "100", а маркер M2 включается, если в D0 хранится меньшее значение, чем "100".

Состояние трех битовых операндов не изменяется и после выключения условия входа, так как сохраняется их последнее состояние.

Для сравнения 32-битовых данных используется команда CMP с дополнительной буквой "D", то есть команда DCMP:

Релейная диаграмма



Список инструкций

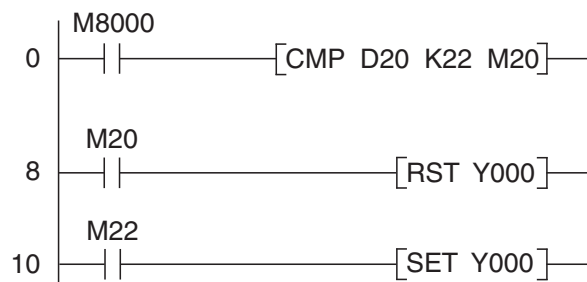
```
0 LD
1 DCMP D0 D2 M0
```

В изображенном вверху примере содержимое D0 и D1 сравнивается с содержимым D2 и D3. Управление тремя битовыми операндами аналогично команде CMP.

Пример применения

С помощью команды CMP можно быстро реализовать простое двухпозиционное регулирование.

Релейная диаграмма



Список инструкций

```
0 LD M8000
1 CMP D20 K22 M20
8 LD M20
9 RST Y000
10 LD M22
11 SET Y0001
```

В этом примере команда CMP обрабатывается циклически. M8000 всегда имеет состояние "1", если контроллер обрабатывает программу. Регистр D20 содержит фактическое значение комнатной температуры. Константа K22 содержит заданное значение 22°C. Маркеры M20 и M22 показывают превышение или занижение заданного значения. Если помещение слишком теплое, выключается выход Y0. При слишком низкой температуре маркер M22 снова включает выход Y0. Через этот выход можно управлять, например, насосом, подающим теплую воду.

5.3.2 Сравнения в рамках логических операций

При использовании вышеописанной команды CMP результат сравнения указывается тремя битовыми операндами. Однако часто бывает удобнее, если от сравнения зависит только одна команда вывода или одна логическая связь, а не три операнда. Для этих целей имеются команды “Загрузки сравнения”, а также сравнения со связью И и ИЛИ.

Сравнение в начале логической связи

Релейная диаграмма



Список инструкций

0 LD>= D40 D50

1 2 3

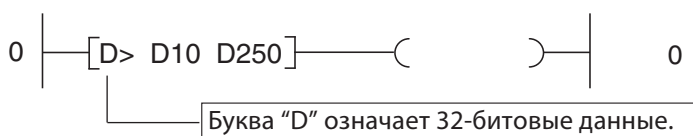
- ❶ Условие сравнения
- ❷ Первое сравниваемое значение
- ❸ Второе сравниваемое значение

Если указанное условие выполнено, то состояние сигнала после команды сравнения равно “1”. Состояние “0” означает, что условие сравнения не выполнено. Возможны следующие сравнения:

- Сравнение на “Равенство”: = (значение 1 = значению 2)
Выход команды имеет состояние “1” только в том случае, если значения обоих операндов были одинаковыми.
- Сравнение на “Больше”: > (значение 1 > значения 2)
Выход команды имеет состояние “1” только в том случае, если 1-е значение больше 2-го значения.
- Сравнение на “Меньше”: < (значение 1 < значения 2)
Выход команды имеет состояние “1” только в том случае, если 1-е значение меньше 2-го значения.
- Сравнение на “Не равно”: <> (значение 1 не равно значению 2)
Выход команды имеет состояние “1” только в том случае, если 1-е и 2-е значения не равны.
- Сравнение на “Меньше или равно”: <= (значение 1 <= значения 2)
Выход команды имеет состояние “1”, если 1-е значение меньше или равно 2-му значению.
- Сравнение на “Больше или равно”: >= (значение 1 >= значение 2)
Выход команды имеет состояние “1”, если 1-е значение больше или равно 2-му значению.

Если требуется сравнить 32-битовые данные, следует добавить букву “D” (что означает “двойные слова”):

Релейная диаграмма



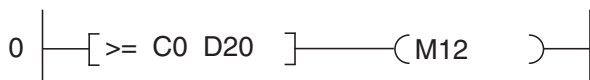
Список инструкций

0 LDD> D10 D250

В этом примере проверяется, превышает ли содержимое регистров данных D10 и D11 содержимое регистров D250 и D251.

Прочие примеры:

Релейная диаграмма

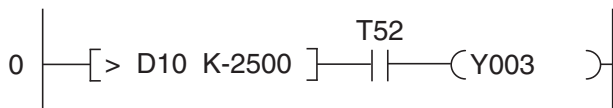


Список инструкций

```
0 LD>= C0 D20
5 OUT M12
```

Маркеры M12 имеют состояние "1", если состояние счетчика C0 равно или превышает содержимое D20.

Релейная диаграмма

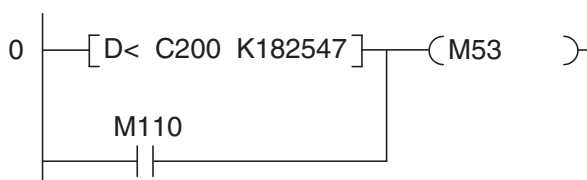


Список инструкций

```
0 LD> D10 K-2500
5 AND T52
6 OUT Y003
```

Если содержимое D10 больше -2500 и таймер T52 истек, включается выход Y003.

Релейная диаграмма



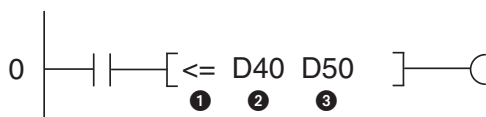
Список инструкций

```
0 LDD< C200 K182547
9 OR M110
10 OUT M53
```

Если состояние 32-битового счетчика C200 меньше 182547 или маркер M110 имеет состояние "1", маркер M53 устанавливается на "1".

Сравнение в качестве операции "И"

Релейная диаграмма



Список инструкций

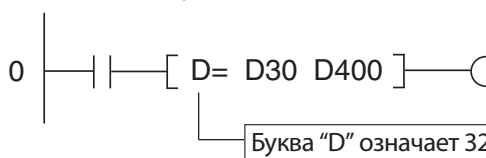
```
0 LD ...
1 AND<= D40 D50
```

- ① Условие сравнения
- ② Первое сравниваемое значение
- ③ Второе сравниваемое значение

Сравнение со связью "И" можно использовать в программе как обычную команду AND (см. раздел 3).

Возможности сравнения аналогичны сравнениям в начале логической связи, описанным выше. Со связью AND тоже можно сравнивать 32-битовые значения:

Релейная диаграмма

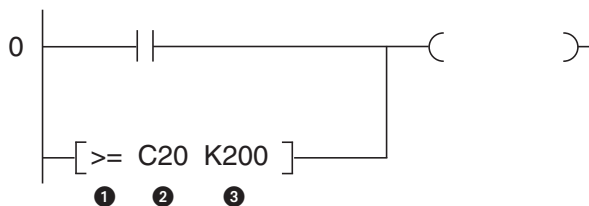


Список инструкций

```
0 ANDD= D30 D400
```


Сравнение в качестве операции "ИЛИ"

Релейная диаграмма



Список инструкций

```

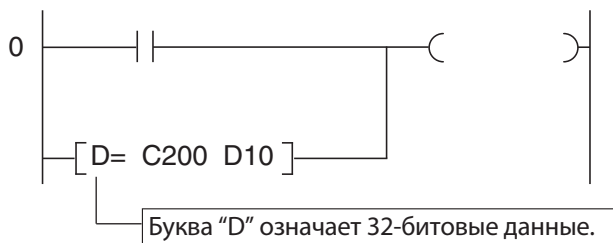
0 LD      ...
1 OR>=   C20   K200
          ①     ②     ③
    
```

- ① Условие сравнения
- ② Первое сравниваемое значение
- ③ Второе сравниваемое значение

Сравнение со связью "ИЛИ" в программе можно использовать как команду OR (см. раздел 3).

Для этого сравнения действительны те же вышеописанные условия. При сравнении 32-битовых данных, как и в случае других операторов сравнения, добавляется буква "D":

Релейная диаграмма



Список инструкций

```

0 LD      ...
1 ORD=   C200   D10
    
```

5.4 Арифметические операции

Все контроллеры MELSEC семейства FX выполняют четыре основные арифметические операции и могут складывать, вычитать, умножать и делить числа без дробной части. Соответствующие команды описываются в этом разделе.

Базовые блоки серий FX2N, FX2NC, FX3G, FX3U и FX3UC могут дополнительно обрабатывать числа с плавающей запятой. Для этого нужны особые команды, подробно описанные в руководстве по программированию семейства FX, артикул 136748.

После сложения или вычитания в программе должны проверяться состояния нижеуказанных специальных маркеров – чтобы определить, не был ли при вычислительной операции превышен допустимый диапазон и не равен ли результат "0".

- M8020

Этот специальный маркер имеет состояние "1", если результат сложения или вычитания равен "0".

- M8021

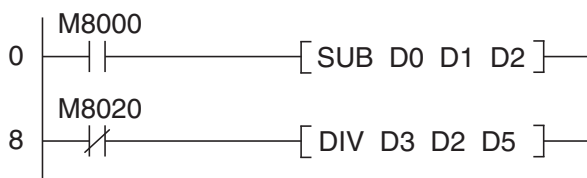
Если результат сложения или вычитания меньше -32 767 (16-битовая операция) или меньше -2 147 483 648 (32-битовая операция), M8021 имеет состояние "1".

- M8022

Если результат превышает значение +32 767 (16-битовые операции) или +2 147 483 647 (32-битовые операции), M8022 имеет состояние "1".

Эти специальные маркеры в программе можно использовать для деблокировки дальнейших вычислительных операций. При нижеописанных вычислениях результат вычитания в D2 используется в качестве делителя. Однако деление на "0" не возможно и вызывает ошибку. Поэтому деление выполняется только в том случае, если делитель не равен "0".

Релейная диаграмма



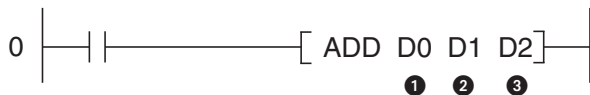
Список инструкций

| | | | | |
|---|-----|-------|----|----|
| 0 | LD | M8000 | | |
| 1 | SUB | D0 | D1 | D2 |
| 8 | LDI | M8020 | | |
| 9 | DIV | D3 | D2 | D5 |

5.4.1 Сложение

С помощью команды ADD складываются два 16- или 32-битовых значения и результат записывается в другой операнд.

Релейная диаграмма



Список инструкций

```
0 ADD   D0   D1   D2
      ①   ②   ③
```

- ① Первый исходный операнд или константа
- ② Второй исходный операнд или константа
- ③ Операнд, в который заносится результат сложения

В изображенном вверху примере складываются содержимые регистров данных D0 и D1 и результат сохраняется в D2.

Примеры

К содержимому регистра данных D100 прибавляется значение "1000":

```
ADD K1000 D100 D102  →  1000 + D 100  
53 → D 102  
1053
```

При сложении учитываются знаки значений:

```
ADD D10 D11 D12  →  D 10  
5 + D 11  
-8 → D 12  
-3
```

Возможно сложение и 32-битовых значений. В этом случае к обозначению команды спереди добавляется буква "D" (ADD -> DADD)

```
DADD D0 D2 D4  →  D 1 D 0  
65238 + D 3 D 2  
27643 → D 5 D 4  
92881
```

Результат можно также снова ввести в один из исходных операндов. Однако учитывайте, что если команда ADD выполняется циклически, то результат изменяется в каждом программном цикле.

```
ADD D0 K25 D0  →  D 0  
18 + 25 → D 0  
43
```

Команду ADD также можно поставить в зависимость от фронта сигнала. В этом случае она выполняется только один раз, если состояние условия входа изменяется с "0" на "1". Для этого просто добавьте к обозначению команды букву "P" (ADD -> ADDP, DADD -> DADDP).

В следующем примере к содержимому D47 прибавляется постоянная "27" только один раз – в том программном цикле, в котором состояние маркера M47 изменяется с "0" на "1".

Релейная диаграмма



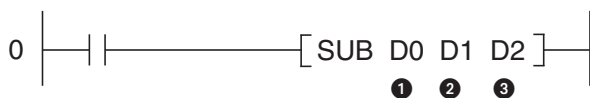
Список инструкций

```
0 LD   M47
1 ADDP D47   K27   D51
```

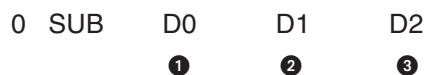
5.4.2 Вычитание

Для вычитания одного числового значения из другого (эти значения могут быть содержимым 16- или 32-битовых операндов или константами) применяется команда SUB. Результат вычитания заносится в третий операнд.

Релейная диаграмма



Список инструкций



- ❶ Уменьшаемое (из этого значения происходит вычитание)
- ❷ Вычитаемое (это значение вычитается)
- ❸ Вычитаемое (это значение вычитается)

Изображенная выше команда вычитает содержимое D1 из содержимого D0 и сохраняет результат в D2.

Примеры

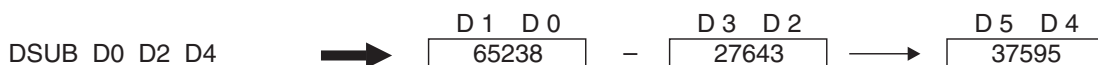
Из содержимого регистра данных D100 вычитается значение "100" и результат сохраняется в D101:



Вычитание происходит с учетом знака значений:



Если операция вычитания применяется для 32-битовых значений, к обозначению команды спереди добавляется буква "D" (SUB -> DSUB).



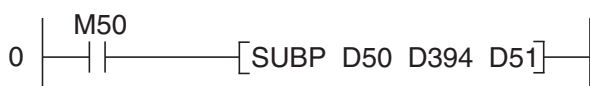
Если операция вычитания применяется для 32-битовых значений, к обозначению команды спереди добавляется буква "D" (SUB -> DSUB).



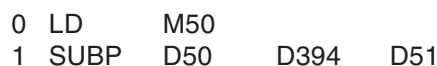
Команда SUB может управляться и фронтом сигнала. В этом случае к обозначению команды добавляется буква "P". (SUB -> SUBP, DSUB -> DSUBP).

Команда SUB может управляться и фронтом сигнала. В этом случае к обозначению команды добавляется буква "P". (SUB -> SUBP, DSUB -> DSUBP).

Релейная диаграмма



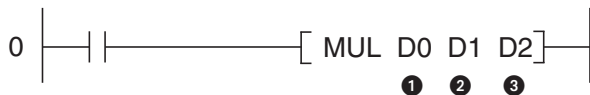
Список инструкций



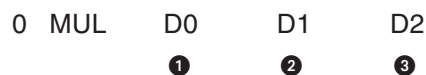
5.4.3 Умножение

Команда MUL в контроллерах FX перемножает два 16- или 32-битовых значения и сохраняет результат в третьей области операндов.

Релейная диаграмма



Список инструкций



- ❶ Умножаемое
- ❷ Множитель
- ❸ Произведение (умножаемое x множитель = произведение)

ПРИМЕЧАНИЕ

При перемножении двух 16-битовых значений результат может выйти за пределы диапазона, представляемого 16 битами. По этой причине произведение всегда сохраняется в двух следующих друг за другом 16-битовых операндах (= 32 бита).

Если перемножаются два 32-битовых значения, результат сохраняется даже в четырех следующих друг за другом 16-битовых операндах (= 64 бита).

Учитывайте размер этих областей операндов при программировании и избегайте двойного присвоения в результате перекрытия диапазонов. В команде указывается операнд, содержащий самые младшие данные.

Примеры

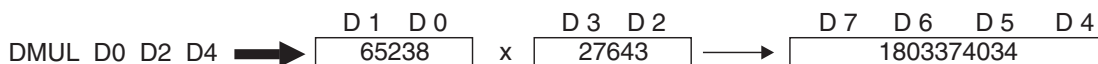
Перемножение содержимого D0 и D1 и сохранение результата в D3 и D2:



Умножение происходит с учетом знака. В этом примере содержимое D10 умножается на константу "-5":



Для перемножения 32-битовых значений обозначение команды дополняется спереди буквой "D" (MUL -> DMUL).

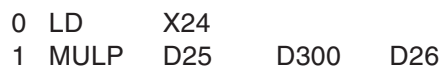


Если к команде MUL добавлена буква "P" (MUL -> MULP, DMUL -> DMULP), то она выполняется в зависимости от фронта сигнала. Следующее умножение происходит только при включении входа X24:

Релейная диаграмма



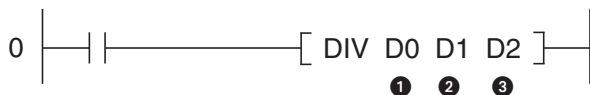
Список инструкций



5.4.4 Деление

Чтобы разделить одно число на другое (это могут быть содержимые 16- или 32-битовых операндов или константы), в контроллерах MELSEC семейства FX имеется команда DIV. Так как эта команда не может обрабатывать числа с дробной частью, результат деления всегда целочисленный. Не разделяемый остаток сохраняется отдельно.

Релейная диаграмма



Список инструкций

0 DIV D0 D1 D2
 ① ② ③

- ① Делимое
- ② Делитель
- ③ Частное (результат деления: делимое / делитель = частное)

ПРИМЕЧАНИЕ

Делитель не должен принимать значение "0". Деление на "0" не возможно и приводит к состоянию ошибки.

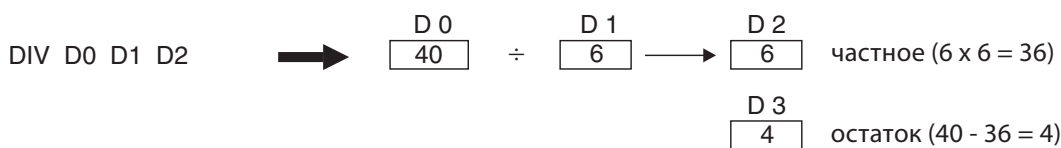
Если в делении участвуют 16-битовые значения, частное сохраняется в 16-битовом операнде, а не разделяемый остаток – в следующем за ним операнде. Таким образом, для результата деления всегда используются два 16-битовых операнда (= 32 бита).

Если в делении участвуют 32-битовые значения, частное сохраняется в двух 16-битовых операндах, а не разделяемый остаток – в двух следующих за ними 16-битовых операндах. При этом типе деления для результата используются четыре следующих друг за другом 16-битовых операнда (=64 бита).

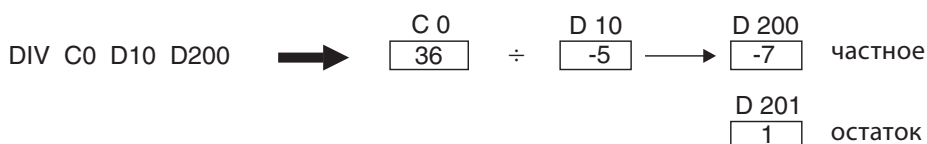
При программировании учитывайте требуемый размер этих областей операндов и избегайте двойного присвоения в результате перекрытия диапазонов. В команде указывается операнд, содержащий самые младшие данные.

Примеры

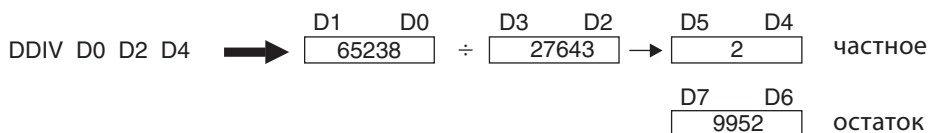
Содержимое D0 делится на содержимое D1 и результат сохраняется в D2 и D3:



При делении учитываются знаки. В этом примере состояние счетчика C0 делится на содержимое D10:



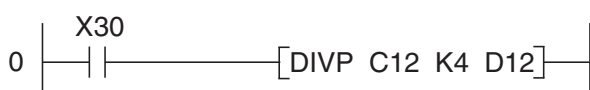
Деление 32-битовых значений:



Если к обозначению команды DIV добавлена буква "P" (DIV -> DIVP, DDIVPL -> DMULP), то она выполняется в зависимости от фронта сигнала. В следующем примере состояние счетчика C12 делится на "4" только в том программном цикле, в котором происходит включение входа X30:

Релейная диаграмма

Список инструкций



```
0 LD X30
1 DIVP C12 K4 D12
```

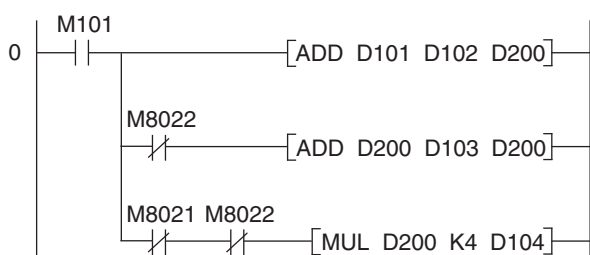
5.4.5 Комбинирование арифметических операций

На практике редко можно обойтись только одним вычислением. Для решения сложных задач арифметические команды можно комбинировать. В зависимости от типа вычисления требуется предусматривать операнды для сохранения промежуточных результатов.

Сложение содержимых регистров данных D101, D102 и D103 с последующим умножением результата на коэффициент "4" можно реализовать, например, следующим образом:

Релейная диаграмма

Список инструкций



```
0 LD M101
1 ADD D101 D102 D200
8 MPS
9 ANI M8022
10 ADD D200 D103 D200
17 MPP
18 ANI M8021
19 ANI M8022
20 MUL D200 K4 D104
```

- Сначала содержимые D101 и D102 складываются и результат сохраняется в промежуточной памяти D200.
- Лишь в том случае, если сумма содержимого D101 и D102 не превышает допустимый диапазон, к нему прибавляется содержимое D103.
- Если сумма содержимого регистров с D101 по D103 находится в допустимом диапазоне, она умножается на коэффициент "4". Результат вычисления заносится в D104 и D105.

6 Возможности расширения

6.1 Общие сведения

В дополнение к базовым блокам имеются модули расширения и специальные модули для дальнейшего расширения контроллера.

Эти модули подразделяются на три категории:

- Модули, занимающие адресное пространство дискретного ввода/вывода. Эти модули монтируются на контроллере справа. К этим модулям относятся дискретные компактные и модульные расширения, а также специальные модули.
- Модули, не занимающие адресное пространство дискретного ввода/вывода. Эти модули монтируются на контроллере слева.
- Интерфейсные и коммуникационные адаптеры, не занимающие адресное пространство дискретного ввода/вывода. Эти модули встраиваются непосредственно в контроллер.

6.2 Обзор

6.2.1 Модули расширения для дополнительных дискретных входов и выходов

Для расширения базовых блоков MELSEC FX1N, FX2N, FX2NC, FX3G, FX3U и FX3UC имеются различные модульные и компактные приборы расширения. Кроме этого, чтобы расширить число дискретных входов/выходов базовые блоки серий FX1S, FX1N, FX3G и FX3U можно дооснастить адаптерами расширения встраиваемыми непосредственно в контроллер. Эти адаптеры удобно использовать, например, в том случае, если нужно лишь небольшое число дополнительных входов-выходов, и не имеется достаточно места для модуля, монтируемого сбоку.

Модульные расширительные приборы имеют только дискретные входы и выходы. Они не имеют собственного сетевого питания. Компактные расширительные приборы имеют большее количество входов/выходов и встроенный сетевой блок для питания системной шины и дискретных входов.

Благодаря обширным возможностям комбинирования базовых и расширительных блоках можно найти оптимальное, экономичное решение для любой задачи.

6.2.2 Аналоговые модули ввода-вывода

Аналоговые модули ввода-вывода способны преобразовывать аналоговые входные сигналы в цифровые значения или цифровые состояния входов в аналоговые сигналы.

Здесь имеется ряд модулей для токовых сигналов и сигналов по напряжению, а также для определения температуры с возможностью непосредственного подключения термометров сопротивления типа Pt100 или термоэлементов.

Основные принципы обработки аналоговых величин разъяснены в разделе 7.

6.2.3 Коммуникационные модули

Мицубиси предлагает ряд интерфейсных модулей и адаптеров с последовательными интерфейсами (RS232, RS422 и RS485) для подключения периферийного оборудования или соединения контроллера с другими системами управления.

Специальные коммуникационные модули позволяют встраивать контроллеры MELSEC FX1N, FX2N, FX2NC, FX3G, FX3U и FX3UC в различные сети.

Имеются модули сетевой коммуникации для Profibus/DP, AS-I, DeviceNet, CANopen, CC-Link, а также для создания собственных сетей Мицубиси.

6.2.4 Модули позиционирования

Помимо внутренних быстрых счетчиков контроллеров MELSEC FX, пользователь может воспользоваться дополнительными высокоскоростными счетными модулями в качестве внешних аппаратных счетчиков, к которым можно подключать инкрементные датчики угла поворота или модули позиционирования для сервоприводов и шаговых приводов.

Для реализации точных задач позиционирования с использованием контроллеров MELSEC семейства FX имеются модули позиционирования для вывода серии импульсов. С помощью этих модулей можно управлять как шаговыми приводами, так и сервоприводами.

6.2.5 Панели интерфейса “человек-машина”

Панели управления Мицубиси Электрик предоставляют пользователю возможность простой и гибкой коммуникации “человек-машина” с контроллерами MELSEC серии FX. Панели интерфейса “человек-машина” привносят прозрачность в функциональные процессы установки.

Все приборы позволяют контролировать и изменять все специфические данные контроллеров, например, заданные и фактические значения времени, счетчиков, регистров данных, а также команды пошагового управления.

Предлагаются на выбор панели с текстовой и/или графической индикацией. Удобство управления повышают свободно программируемые функциональные клавиши или сенсорные экраны. Программирование и конфигурирование осуществляются просто и удобно через компьютер с операционной средой Windows®.

Коммуникация между панелями управления и контроллером FX осуществляется через программирующий интерфейс контроллера с помощью соответствующего кабеля. Для соединения с контроллером вам не понадобятся дополнительные модули.

7 Обработка аналоговых значений

7.1 Аналоговые модули

При автоматизации процессов часто требуется измерять или регулировать аналоговые величины, например, температуру, давление или уровни. Без дополнительных модулей базовый блок MELSEC семейства FX может обрабатывать только дискретные входные или выходные сигналы (т. е. информацию типа "включено-выключено"). Поэтому для оценки и вывода аналоговых сигналов нужны особые аналоговые модули.

В принципе, имеются два вида этих модулей

- входные аналоговые модули и
- выходные аналоговые модули.

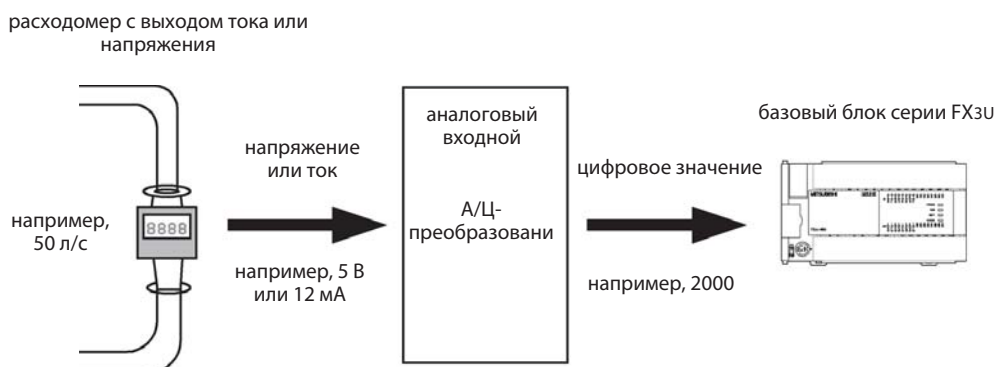
Входные аналоговые модули могут оценивать токи, напряжения или температуру. Выходные аналоговые модули служат для вывода токов или напряжений. Кроме того, имеются комбинированные модули, которые могут и оценивать, и выводить аналоговые сигналы.

Входные аналоговые модули

Входные аналоговые модули преобразуют измеренное аналоговое значение (например, 10 В) в цифровое значение (например, 4000), которым далее может оперировать программируемый контроллер. Этот процесс называется аналого-цифровым преобразованием или, сокращенно, А/Ц-преобразованием.

В отличие от температуры, которую аналоговые модули MELSEC семейства FX могут оценивать непосредственно, прочие физические сигналы (например, давление или расход) для обработки в программируемом контроллере сначала необходимо преобразовать в значения тока или напряжения. Это преобразование выполняют измерительные датчики, с выходов которых снимаются стандартизованные сигналы (например, 0...10 В или 4...20 мА). Измерение тока имеет то преимущество, что измерительный сигнал не зависит от длины проводки или переходных сопротивлений.

На рисунке ниже в качестве примера оценки аналоговой величины показано измерение расхода с помощью программируемого контроллера MELSEC серии FX3U.



Аналоговые входные модули для измерения температуры

Для определения температуры используются либо термометры сопротивления Pt100, либо термоэлементы.

- Термометр сопротивления Pt100

При этом методе измерения температуры измеряется сопротивление платинового элемента, которое по мере возрастания температуры увеличивается. При 0°C платиновый элемент имеет сопротивление 100 Ом (отсюда обозначение "Pt100"). Резистивные датчики подключаются по трехпроводной схеме, благодаря чему сопротивление соединительной проводки не влияет на результат измерения.

Измерительный диапазон термометров сопротивления Pt100 составляет от -200 до 600°C, но он зависит и от используемого модуля измерения температуры.

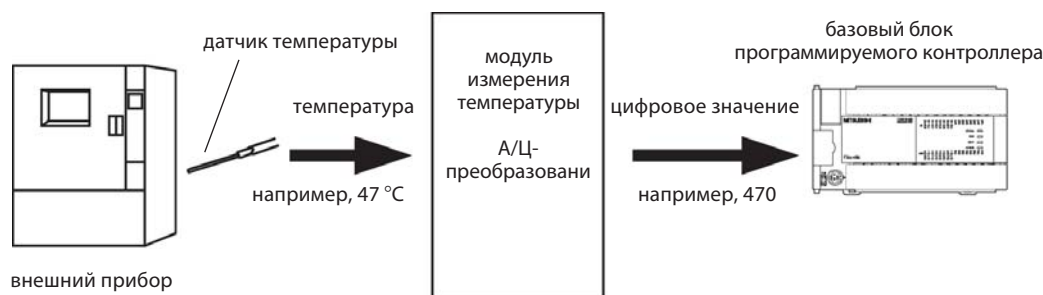
- Термоэлементы

При этом методе измерения температуры используется эффект возникновения напряжения при соединении различных металлов в зависимости от температуры. Таким образом, этот принцип измерения температуры основывается на измерении напряжения.

Имеются термоэлементы различных видов. Они различаются между собой по термоэлектродвижущей силе и диапазону измеряемой температуры. Сочетание материалов стандартизовано и указывается в обозначении типа. Наиболее часто используются термоэлементы типов J и K. В термоэлементах типа K сочетаются материалы NiCr и Ni. Для изготовления термоэлементов типа J используются железо (Fe) и медно-никелевый сплав (CuNi). Помимо конструкции, термоэлементы различаются между собой диапазоном измеряемой температуры.

С помощью термоэлементов можно измерять температуру от -200 до 1200°C.

Пример измерения температуры:

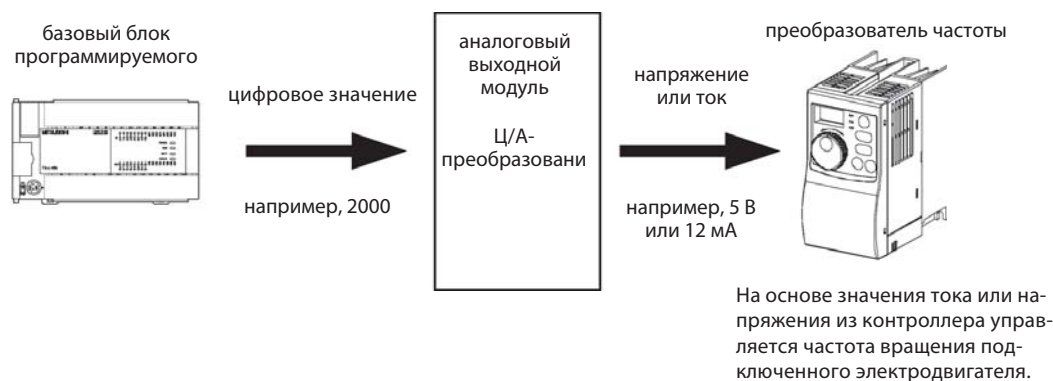


Выходные аналоговые модули

Аналоговые выходные модули преобразуют цифровое значение, поступающее из базового блока программируемого контроллера, в аналоговый сигнал тока или напряжения, с помощью которого можно управлять внешними приборами (цифро-аналоговое преобразование, сокращенно Ц/А-преобразование).

Аналоговые выходные сигналы аналоговых модулей MELSEC семейства FX соответствуют промышленному стандарту 0...10 В или 4...20 мА.

В качестве примера применения иллюстрация на следующей странице показывает заданное значение для преобразователя частоты. Значение тока или напряжения, выводимое контроллером, влияет на частоту вращения электродвигателя, подключенного к преобразователю частоты.



7.1.1 Критерии выбора аналоговых модулей

Семейство MELSEC FX содержит большой ассортимент аналоговых модулей. Поэтому для решения определенных задач автоматизации необходимо сделать определенный выбор. При этом главными критериями выбора являются следующие:

- Совместимость с базовым блоком программируемого контроллера

Аналоговый модуль должен быть совместим с используемым базовым блоком программируемого контроллера. Например, аналоговые модули серии FX3U не могут быть подключены к базовому блоку серии FX1N.

- Разрешающая способность

"Разрешающая способность" означает, какова минимальная физическая величина, которую может оценивать или выдавать аналоговый модуль.

В случае входных аналоговых модулей под разрешающей способностью подразумевается такое изменение напряжения, тока или температуры на входе, которое вызывает увеличение или уменьшение цифрового выходного значения на "1".

В случае выходных аналоговых модулей под разрешающей способностью подразумевается величина изменения напряжения или тока на выходе модуля при увеличении или уменьшении цифрового входного значения на "1".

Разрешающая способность определяется внутренней конструкцией аналоговых модулей, в частности, тем, сколько битов используется для сохранения цифрового значения. Если, например, напряжение 10 В оценивается 12-битным А/Ц-преобразователем, то напряжение разбивается на 4096 ступеней ($2^{12} = 4096$, см. раздел 3.3). Таким образом, получаем разрешающую способность $10\text{В} / 4096 = 2,5\text{ мВ}$.

- Количество аналоговых входов или выходов

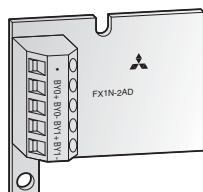
Входы или выходы аналогового модуля называются также "каналами". В зависимости от требуемого количества каналов можно выбрать, например, 2-, 4- или 8-канальные аналоговые входные модули. Учитывайте, что число специальных модулей, подключаемых к базовому блоку программируемого контроллера, ограничено (см. также раздел 7.1.2). Поэтому если требуется установить и другие специальные модули, то целесообразнее вместо двух двухканальных аналоговых модулей установить один четырехканальный.

7.1.2 Адаптеры, адаптерные модули и специальные модули

В семействе MELSEC FX имеются аналоговые модули различных видов.

Аналоговые адаптеры

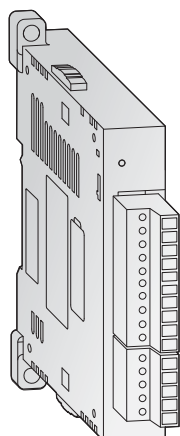
Аналоговые адаптеры представляют собой небольшие платы, встраиваемые непосредственно в базовые блоки серии FX1S, FX1N или FX3G. Благодаря этому для размещения контроллера в распределительном шкафу не требуется дополнительное место.



Аналоговый входной адаптер записывает цифровые значения обоих входных каналов непосредственно в специальный регистр. Это существенно упрощает дальнейшую обработку измеренных значений. Выходное значение для аналогового выходного модуля также записывается программой в специальный регистр, а затем преобразуется адаптером и выводится.

Адаптерные модули

Адаптерные модули можно подключить только с левой стороны базового блока MELSEC серии FX3G, FX3U или FX3UC.

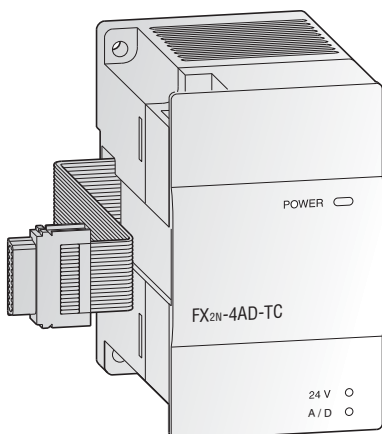


К базовым блокам FX3G с 14 или 24 входами и выходами можно подсоединить один аналоговый адаптерный модуль. К базовым блокам FX3G с 40 или 60 входами и выходами можно подсоединить до двух аналоговых адаптерных модулей, а к базовым блокам FX3U или FX3UC – до четырех.

Адаптерные модули не занимают адресное пространство ввода/вывода в базовом блоке. Для коммуникации между базовым блоком и адаптерным модулем используются специальные маркеры и регистры. Благодаря этому в программе не нужны никакие команды для коммуникации со специальными модулями (см. ниже).

Специальные модули

С правой стороны базового блока MELSEC семейства FX можно подключить до восьми специальных модулей.



Помимо аналоговых модулей, к специальным модулям относятся, например, модули коммуникации и позиционирования. Каждый специальный модуль занимает в базовом блоке восемь входов и восемь выходов. Коммуникация между специальным модулем и базовым блоком программируемого контроллера осуществляется через буферную память специального модуля с помощью команд FROM и TO (см. раздел 5.2.5).

7.2 Обзор аналоговых модулей

| Тип модуля | Обозначение | Аналоговые каналы | Диапазон | Разрешающая способность | FX1S | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC | |
|----------------------------|--------------------|---|--|--|--|------|---------------|------|---------------|---|
| Входные аналоговые модули | Адаптеры | FX1N-2AD-BD | 2 | Напряжение: 0 В...10 В пост. | 2,5 мВ (12 битов) | ● | ● | ○ | ○ | ○ |
| | | | | Ток: 4 мА...20 мА пост. | 8 мкА (11 битов) | | | | | |
| | | FX3G-2AD-BD | 2 | Напряжение: 0 В...10 В пост. | 2,5 мВ (12 битов) | ○ | ○ | ○ | ● | ○ |
| | | | | Ток: 4 мА...20 мА пост. | 8 мкА (11 битов) | | | | | |
| | Адаптерный модуль | FX3U-4AD-ADP | 4 | Напряжение: 0 В...10 В пост. | 2,5 мВ (12 битов) | ○ | ○ | ○ | ● | ● |
| | | | | Ток: 4 мА...20 мА пост. | 10 мкА (11 битов) | | | | | |
| | Специальные модули | FX2N-2AD | 2 | Напряжение: 0 В...5 В пост. 0 В...10 В пост. | 2,5 мВ (12 битов) | ○ | ● | ● | ● | ● |
| | | | | Ток: 4 мА...20 мА пост. | 4 мкА (12 битов) | | | | | |
| | | FX2N-4AD | 4 | Напряжение: -10 В...10 В пост. | 5 мВ (с арифметическим знаком, 12 битов) | ○ | ● | ● | ● | ● |
| | | | | Ток: 4 мА...20 мА пост. -20 мА...20 мА пост. | 10 мкА (с арифметическим знаком, 11 битов) | | | | | |
| | | FX2N-8AD* | 8 | Напряжение: -10 В...10 В пост. | 0,63 мВ (с арифметическим знаком, 15 битов) | ○ | ● | ● | ● | ● |
| | | | | Ток: 4 мА...20 мА пост. -20 мА...20 мА пост. | 2,50 мкА (с арифметическим знаком, 14 битов) | | | | | |
| FX3U-4AD | 4 | Напряжение: -10 В...10 В пост. | 0,32 мВ (с арифметическим знаком, 16 битов) | ○ | ○ | ○ | ● | ● | | |
| | | Ток: 4 мА...20 мА пост. -20 мА...20 мА пост. | 1,25 мкА (с арифметическим знаком, 15 битов) | | | | | | | |
| Выходные аналоговые модули | Адаптеры | FX1N-1DA-BD | 1 | Напряжение: 0 В...10 В пост. | 2,5 мВ (12 битов) | ● | ● | ○ | ○ | ○ |
| | | | | Ток: 4 мА...20 мА пост. | 8 мкА (11 битов) | | | | | |
| | | FX3G-1DA-BD | 1 | Напряжение: 0 В...10 В пост. | 2,5 мВ (12 битов) | ○ | ○ | ○ | ● | ○ |
| | | | | Ток: 4 мА...20 мА пост. | 8 мкА (11 битов) | | | | | |
| | Адаптерный модуль | FX3U-4DA-ADP | 4 | Напряжение: 0 В...10 В пост. | 2,5 мВ (12 битов) | ○ | ○ | ○ | ● | ● |
| | | | | Ток: 4 мА...20 мА пост. | 4 мкА (12 битов) | | | | | |
| | Специальные модули | FX2N-2DA | 2 | Напряжение: 0 В...5 В пост. 0 В...10 В пост. | 2,5 мВ (12 битов) | ○ | ● | ● | ● | ● |
| | | | | Ток: 4 мА...20 мА пост. | 4 мкА (12 битов) | | | | | |
| | | FX2N-4DA | 4 | Напряжение: -10 В...10 В пост. | 5 мВ (с арифметическим знаком, 12 битов) | ○ | ● | ● | ● | ● |
| | | | | Ток: 0 мА...20 мА пост. 4 мА...20 мА пост. | 20 мкА (10 битов) | | | | | |
| | | FX3U-4DA | 4 | Напряжение: -10 В...10 В пост. | 0,32 мВ (с арифметическим знаком, 16 битов) | ○ | ○ | ○ | ● | ● |
| | | | | Ток: 0 мА...20 мА пост. 4 мА...20 мА пост. | 0,63 мкА (15 битов) | | | | | |

* Помимо токов и напряжений, специальный модуль FX2N-8AD может также оценивать температуру.

| Тип модуля | | Обозначение | Аналоговые каналы | Диапазон | Разрешающая способность | FX1S | FX1N | FX2N FX2NC | FX3G | FX3U FX3UC |
|---|-----------------------------------|------------------|--|---|--|------|------|---------------|------|---------------|
| Комбинированные модули аналогового ввода-вывода | Специальные модули | FX0N-3A | 2 входов | Напряжение: 0 В...5 В пост. 0 В...10 В пост. | 40 мВ (8 битов) | ○ | ● | ● | | ● |
| | | | | Ток: 4 мА...20 мА пост. | 64 мкА (8 битов) | | | | | |
| | | | 1 выход | Напряжение: 0 В...5 В пост. 0 В...10 В пост. | 40 мВ (8 битов) | | | | | |
| | | | | Ток: 4 мА...20 мА пост. | 64 мкА (8 битов) | | | | | |
| | | FX2N-5A | 4 входов | Напряжение: -100 мВ...100 мВ пост. -10 В...10 В пост. | 50 мкВ (с арифметическим знаком, 12 битов) 0,312 мВ (с арифметическим знаком, 16 битов) | ○ | ● | ● | ○ | ● |
| | | | | Ток: 4 мА...20 мА пост. -20 мА...20 мА пост. | 10 мкА/1,25 мкА (с арифметическим знаком, 15 битов) | | | | | |
| 1 выход | Напряжение: -10 В...10 В пост. | | 5 мВ (с арифметическим знаком, 12 битов) | | | | | | | |
| | Ток: 0 мА...20 мА пост. | | 20 мкА (10 битов) | | | | | | | |
| Модули измерения температуры | Адаптерный модуль | FX3U-4AD-PT-ADP | 4 | термометр сопротивления Pt100: -50 °C...250 °C | 0,1 °C | ○ | ○ | ○ | ● | ● |
| | | FX3U-4AD-PTW-ADP | 4 | термометр сопротивления Pt100: -100 °C...600 °C | 0,2 °C...0,3 °C | ○ | ○ | ○ | ● | ● |
| | | FX3U-4AD-PNK-ADP | 4 | термометр сопротивления Pt1000: -50 °C...250 °C | 0,1 °C | ○ | ○ | ○ | ● | ● |
| | | | | термометр сопротивления Ni1000: -40 °C...110 °C | 0,1 °C | ○ | ○ | ○ | ● | ● |
| | | FX3U-4AD-TC-ADP | 4 | термоэлемент типа К: -100 °C...1000 °C | 0,4 °C | ○ | ○ | ○ | ● | ● |
| | | | | термоэлемент типа J: -100 °C...600 °C | 0,3 °C | | | | | |
| | Специальные модули | FX2N-8AD* | 8 | термоэлемент типа К: -100 °C...1200 °C | 0,1 °C | ○ | ● | ● | ● | ● |
| | | | | термоэлемент типа J: -100 °C...600 °C | 0,1 °C | | | | | |
| | | | | термоэлемент типа Т: -100 °C...350 °C | 0,1 °C | | | | | |
| | | FX2N-4AD-PT | 4 | термометр сопротивления Pt100: -100 °C...600 °C | 0,2 °C...0,3 °C | ○ | ● | ● | ● | ● |
| | | FX2N-4AD-TC | 4 | термоэлемент типа К: -100 °C...1200 °C | 0,4 °C | ○ | ● | ● | ● | ● |
| | | | | термоэлемент типа J: -100 °C...600 °C | 0,3 °C | | | | | |
| Модуль регулирования температуры (Специальный модуль) | FX2N-2LC | 2 | например термоэлемент типа К: -100 °C...1300 °C термометр сопротивления Pt100: -200 °C...600 °C | 0,1 °C oder 1 °C (в зависимости от используемого датчика температуры) | ○ | ● | ● | ● | ● | |

* Помимо токов и напряжений, специальный модуль FX2N-8AD может также оценивать температуру.

● : Модуль можно сочетать с базовым блоком или расширительным прибором этой серии.

○ : Применение модуля не возможно.

Указатель ключевых слов

А

Аналоговые адаптеры 7-4
Адаптерные модули 7-4

Б

Батарейка 2-9
Блокировочные контакты 3-21
Буферная память 5-12

В

Восьмеричная система счисления 3-4
Выключатель RUN/STOP 2-9
Выходные аналоговые модули 7-2
Входные аналоговые модули 7-1

Д

Двоичная система счисления 3-2
Двоичные числа 3-2

Е

EEPROM 2-9

З

Задержка выключения 4-14
Звенья выдержки времени
см. Таймеры

И

Источник сервисного напряжения 2-9

К

Команды
ADD 5-21
ANB 3-12
AND 3-9
ANDF 3-14
ANDP 3-14
ANI 3-9
BMOV 5-10
CMP 5-15
DIV 5-24
FMOV 5-11
FROM 5-14
INV 3-20
LD 3-6
LDF 3-14

LDI 3-6
LDP 3-14
MC 3-19
MCR 3-19
MOV 5-7
MPP 3-17
MPS 3-17
MRD 3-17
MUL 5-23
OR 3-11
ORB 3-12
ORF 3-14
ORI 3-11
ORP 3-14
OUT 3-6
PLF 3-18
PLS 3-18
RST 3-15
SET 3-15
SUB 5-22
TO 5-15
Нарастающий фронт 3-14
Нисходящий фронт 3-14

М

Модули измерения температуры 7-2

О

Обратная связь 3-22
Операнды
адрес 3-1
счетчики (обзор) 4-8
регистры данных (обзор) 4-10
входы и выходы (обзор) 4-2
регистры файлов (обзор) 4-11
обозначение 3-1
маркеры (обзор) 4-3
таймеры (обзор) 4-6
Оптический соединитель 2-6
Отображение процесса 2-2
Оценка фронта сигнала 3-14

П

Примеры программ

- датчик тактовых импульсов 4-16
- заданное значение таймера и счетчика . . . 4-11
- задержка включения 4-4
- задержка выключения 4-14
- установка охранной сигнализации 3-23
- шторные ворота 3-28

Принудительные отключения 3-22

С

Специальные маркеры 4-3

Специальные модули

- обмен данными с базовым блоком 5-12

Специальные регистры 4-10

Счетчики

- принцип работы 4-7
- косвенное задание значения 4-11

Т

термометр сопротивления Pt100 7-2

У

Управляющая команда 3-1

Устойчивость к обрыву проводки 3-21

Устройства аварийного выключения 3-21

Ф

Фиксируемые таймеры 4-5

Ш

Шестнадцатичная система счисления 3-3

MITSUBISHI ELECTRIC EUROPE B.V. /// РОССИЯ /// Москва /// Космодамианская наб. 52, стр. 3
Тел.: +7 495 721-2070 /// Факс: +7 495 721-2071 /// automation@mer.mee.com /// www.mitsubishi-automation.ru