

System Q

Программируемые логические контроллеры

Руководство для начинающих
пользователей

Назначение данного руководства

В данном руководстве описания, иллюстрации, схемы и примеры приведены исключительно для справки. Они служат для разъяснения установки, программирования и эксплуатации программируемых логических контроллеров System Q.

Если возникнут вопросы по установке и эксплуатации каких-либо изделий, описанных в данном руководстве, обратитесь в ваше региональное торговое представительство или к региональному дистрибьютору (см. заднюю обложку).

Вы можете ознакомиться с новой информацией и ответами на часто задаваемые вопросы на веб-сайте www.mitsubishi-automation.ru.

MITSUBISHI ELECTRIC EUROPE BV оставляет за собой право вносить изменения в данное руководство и технические данные своих продуктов в любое время без уведомления.

© 07/2008

**Руководство для начинающих пользователей программируемых логических контроллеров System Q
кат. № 209097**

Версия			Изменения / Дополнения / Исправления
A	07/2008	pdp-dk	Первая редакция

Указания по технике безопасности

Только для квалифицированных специалистов

Данное руководство адресовано исключительно квалифицированным специалистам, получившим соответствующее образование и знающим стандарты безопасности в области электротехники и техники автоматизации. Проектировать, устанавливать, конфигурировать, обслуживать и проверять данное оборудование разрешается только квалифицированным специалистам, получившим соответствующее образование. Вмешательства в аппаратную часть и программное обеспечение нашей продукции, не описанные в этом или иных руководствах, разрешены только нашим специалистам. Любые операции и внесение изменений в аппаратную часть и программное обеспечение данной продукции, если они не предусмотрены в настоящем руководстве, допускаются только с разрешения специалистов фирмы Mitsubishi Electric.

Использование изделий по назначению

Программируемые логические контроллеры (ПЛК) серии System Q предназначены только для тех областей применения, которые описаны в данном руководстве. Необходимо соблюдать параметры и настройки, указанные в настоящем руководстве. Представленная продукция разработана, изготовлена, проверена и описана в документации в строгом соответствии с применимыми стандартами безопасности. Неправомочное вмешательство в аппаратную часть или программное обеспечение, либо несоблюдение предупреждений, содержащихся в данном руководстве или указанных на продукции, могут привести к тяжелым травмам и/или материальному ущербу. В сочетании с программируемыми логическими контроллерами System Q разрешается использовать только периферийные устройства и модули расширения, рекомендуемые фирмой Mitsubishi Electric.

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

Нормативные требования по технике безопасности

При проектировании, установке, конфигурировании, техническом обслуживании и проверке оборудования должны соблюдаться нормативные требования по технике безопасности и охране труда, относящиеся к специфическому случаю применения. Особенно важно соблюдать нижеуказанные нормативные требования. Данный перечень не претендует на полноту, однако необходимо ознакомиться с этими требованиями и соблюдать их применительно к конкретным условиям эксплуатации.

- Стандарты VDE (Электротехнический союз Германии)
 - VDE 0100
Требования к монтажу электроустановок с номинальным напряжением ниже 1000 В
 - VDE 0105
Эксплуатация силовых установок
 - VDE 0113
Электроустановки с электронной аппаратурой
 - VDE 0160
Электронная аппаратура для силовых установок
 - VDE 0550/0551
Требования к трансформаторам
 - VDE 0700
Безопасность электрического оборудования бытового и другого аналогичного назначения

- VDE 0860
Нормативные требования по технике безопасности для электронного оборудования с питанием от бытовой электросети и соответствующих принадлежностей бытового и другого аналогичного назначения
- Нормативные требования по противопожарной безопасности
- Нормативные требования по технике безопасности и охране труда
 - VBG № 4
Электрические системы и оборудование

Используемые в руководстве предупреждения по технике безопасности

В настоящем руководстве используются следующие предупреждения по технике безопасности:



ОПАСНО

Несоблюдение мер предосторожности, обозначенных данным символом, может создать угрозу для жизни или здоровья пользователя.



ВНИМАНИЕ

Несоблюдение мер предосторожности, обозначенных данным символом, может привести к значительным повреждениям оборудования или иного имущества.

Общие сведения по технике безопасности и мерам предосторожности

Следующие указания по технике безопасности и меры предосторожности представляют собой общие правила эксплуатации систем ПЛК совместно с другим оборудованием. Данные правила необходимо строго соблюдать при проектировании, установке и эксплуатации любых систем управления.

**ОПАСНО**

- Соблюдайте все правила техники безопасности и охраны труда, относящиеся к данному применению. Обязательно полностью отключайте электропитание перед монтажом и выполнением электропроводки или открыванием узлов, компонентов и устройств.
- Узлы, компоненты и устройства необходимо устанавливать только в ударопрочных корпусах, снабженных подходящими крышками и предохранителями либо автоматическими выключателями.
- Устройства с постоянным подключением к электросети следует подключать к инженерным коммуникациям здания с помощью однополюсного разъединителя и соответствующего предохранителя.
- Регулярно проверяйте подключенные к оборудованию силовые кабели и линии на предмет обрыва и повреждения изоляции. При обнаружении повреждения кабеля немедленно отключите оборудование и кабели от электросети и замените дефектный кабель.
- Перед первым включением оборудования убедитесь, что параметры электросети соответствуют параметрам электропитания оборудования.
- Примите необходимые меры, чтобы исключить возникновение неопределенного состояния оборудования в случае повреждения кабеля или обрыва жилы сигнального провода.
- Вы отвечаете за принятие необходимых мер предосторожности, обеспечивающих корректный и безопасный перезапуск программы в случае её прерывания при падении напряжения или сбое электропитания. В частности, необходимо исключить возникновение опасных условий, в том числе кратковременных, при любых обстоятельствах.
- Средства аварийного отключения, соответствующие требованиям EN 60204/IEC 204 и VDE 0113, должны постоянно находиться в полностью работоспособном состоянии при любых режимах работы ПЛК. Функция сброса средств аварийного отключения должна быть реализована таким образом, чтобы исключался неконтролируемый или произвольный перезапуск.
- Необходимо реализовать аппаратные и программные предохранительные средства, предотвращающие возникновение неопределенного состояния системы управления в случае обрыва кабеля или жилы провода сигнальной линии.
- При использовании модулей обязательно строго соблюдайте все электротехнические и механические параметры и требования.

Содержание

1	Введение	
1.1	Назначение данного руководства	1-1
1.2	Дополнительная информация	1-1
2	Программируемые логические контроллеры	
2.1	Общие сведения о ПЛК	2-1
2.2	Обработка программ ПЛК	2-2
3	Контроллер System Q	
3.1	Конфигурация системы	3-1
3.2	Базовые шасси	3-3
3.2.1	Кабели шасси расширения	3-3
3.2.2	Назначение адресов ввода/вывода	3-4
3.3	Модули питания	3-5
3.4	Модули ЦП	3-7
3.4.1	Компоненты модулей ЦП	3-9
3.4.2	Организация памяти	3-12
3.4.3	Установка батареи для модуля ЦП	3-15
3.5	Модули дискретных входов и выходов	3-16
3.5.1	Модули дискретных входов	3-17
3.5.2	Модули дискретных выходов	3-24
3.6	Специальные функциональные модули	3-31
3.6.1	Аналоговые модули	3-31
3.6.2	Модули регулирования температуры с ПИД-регулятором	3-34
3.6.3	Модули высокоскоростных счётчиков	3-34
3.6.4	Модули позиционирования	3-35
3.6.5	Модули последовательной связи	3-35
3.6.6	Программируемые на BASIC интерфейсные модули	3-36
3.7	Сети и сетевые модули	3-37
3.7.1	Сети разных уровней	3-37
3.7.2	Открытые сети	3-38
3.7.3	Сети MELSEC	3-40
3.7.4	Сетевые модули	3-41

4	Введение в программирование	
4.1	Структура программной команды	4-1
4.2	Биты, байты и слова	4-2
4.3	Системы счисления	4-2
4.4	Коды	4-5
4.4.1	Двоично-десятичный код	4-5
4.4.2	ASCII-код	4-6
4.5	Языки программирования	4-7
4.5.1	Текстовые редакторы	4-7
4.5.2	Графические редакторы	4-8
4.6	Стандарт IEC 61131-3	4-10
4.6.1	Структура программы	4-10
4.6.2	Переменные	4-11
4.7	Набор основных команд	4-13
4.7.1	Запуск логических операций	4-14
4.7.2	Вывод результата логической операции	4-14
4.7.3	Использование выключателей и датчиков	4-16
4.7.4	Операции AND	4-17
4.7.5	Операции OR	4-18
4.7.6	Команды подключения блоков операций	4-20
4.7.7	Импульсный запуск выполнения операций	4-22
4.7.8	Установка и сброс устройств	4-25
4.7.9	Формирование импульсов	4-28
4.7.10	Инверсия результата операции	4-29
4.7.11	Инверсия битового выходного устройства	4-30
4.7.12	Преобразование результата операции в импульс	4-31
4.8	Обеспечение безопасности	4-32
4.9	Программирование приложений ПЛК	4-34
4.9.1	Свёртывающаяся шторная дверь	4-34
4.9.2	Программирование	4-35
4.9.3	Аппаратура	4-46

5	Устройства	
5.1	Входы и выходы	5-1
5.1.1	Внешние сигналы ввода/вывода и номера ввода/вывода.....	5-2
5.1.2	Входы и выходы System Q.....	5-3
5.2	Реле	5-4
5.2.1	Специальные реле.....	5-5
5.3	Таймеры.....	5-6
5.4	Счётчики.....	5-9
5.5	Регистры	5-11
5.5.1	Регистры данных	5-11
5.5.2	Специальные регистры.....	5-12
5.5.3	Файловые регистры	5-13
5.6	Константы	5-14
5.6.1	Десятичные и шестнадцатеричные константы.....	5-14
5.6.2	Константы с плавающей десятичной запятой.....	5-14
5.6.3	Константы типа «символьная строка»	5-14
5.7	Рекомендации по программированию таймеров и счётчиков	5-15
5.7.1	Косвенное задание значений уставок таймеров и счётчиков.....	5-15
5.7.2	Задержка выключения.....	5-17
5.7.3	Включение и выключение с задержкой.....	5-19
5.7.4	Генераторы тактового сигнала	5-20
6	Расширенное программирование	
6.1	Справочная таблица прикладных команд	6-1
6.1.1	Дополнительные команды для ЦП управления непрерывным процессом.....	6-10
6.2	Команды для перемещения данных	6-12
6.2.1	Перемещение отдельных значений с помощью команды MOV	6-12
6.2.2	Перемещение групп битовых устройств.....	6-14
6.2.3	Перемещение блоков данных с помощью команды BMOV.....	6-16
6.2.4	Копирование устройств-источников в несколько адресов назначения (FMOV)	6-17
6.2.5	Обмен данными со специальными функциональными модулями	6-18
6.3	Команды сравнения	6-22

6.4	Команды математических функций.....	6-25
6.4.1	Сложение.....	6-25
6.4.2	Вычитание.....	6-28
6.4.3	Умножение.....	6-29
6.4.4	Деление.....	6-30
6.4.5	Комбинирование команд математических функций.....	6-31

1 Введение

1.1 Назначение данного руководства

В данном руководстве приведено описание использования программируемых логических контроллеров System Q. В первую очередь оно предназначено для пользователей, не имеющих опыта программирования программируемых логических контроллеров (ПЛК).

Специалисты, имеющие опыт работы с ПЛК других производителей, также могут использовать данное руководство в качестве справочного пособия, начиная работать с контроллерами System Q.

1.2 Дополнительная информация

Подробная информация по отдельным модулям System Q содержится в руководствах по эксплуатации и установке отдельных модулей.

Общее описание всех контроллеров System Q содержится в техническом каталоге серии System Q, кат. № 136731.

В данном каталоге также приведена информация по специальным функциональным модулям и предусмотренным аксессуарам. Коммуникационные возможности с использованием сети MELSEC и открытых сетей, таких как Ethernet и Profibus, подробно описаны в техническом каталоге по сетям, кат. № 136730.

В описаниях аппаратной части System Q содержится информация, необходимая для проектирования системы контроллера, а также для установки ПЛК и ввода его в эксплуатацию.

Вводная информация по использованию программного пакета для программирования содержится в руководстве для начинающих пользователей пакета GX IEC Developer, кат. № 043596 и в справочном руководстве, кат. № 043597.

Подробная информация обо всех командах программирования приведена в руководстве по программированию для серий A/Q и System Q, кат. № 87431. Дополнительные примеры программ представлены практически во всех руководствах по специальным функциональным модулям.

ПРИМЕЧАНИЕ

Все руководства и каталоги компании «Мицубиси» можно бесплатно загрузить на веб-сайте компании «Мицубиси» по адресу: www.mitsubishi-automation.ru.

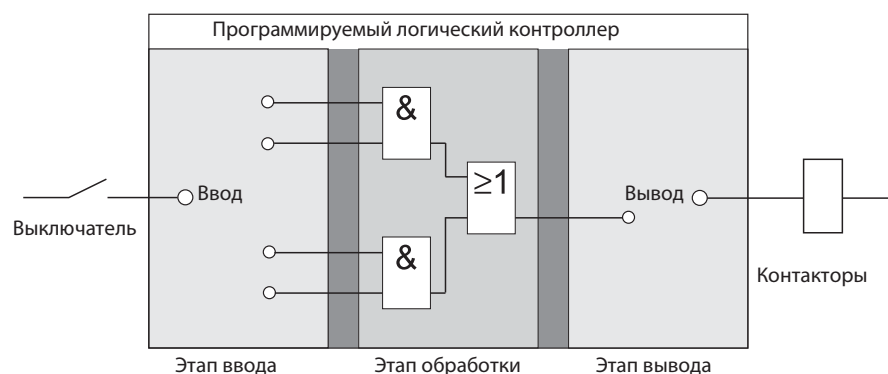
2 Программируемые логические контроллеры

2.1 Общие сведения о ПЛК

В отличие от простейших контроллеров, функционал которых определяется их физическим подключением, функции программируемых логических контроллеров (ПЛК) определяются программой. Данные ПЛК тоже подключаются к внешним системам с помощью кабелей, но содержимое их памяти программы можно изменять в любое время, адаптируя программы к различным задачам управления.

Программируемые логические контроллеры обеспечивают ввод данных, их обработку и вывод результатов. Данный процесс состоит из трех этапов:

- этап ввода,
 - этап обработки
- и
- этап вывода



Этап ввода

На этапе ввода управляющие сигналы с переключателей, кнопок и датчиков передаются на этап обработки.

Сигналы с данных компонентов формируются в процессе управления и передаются на входы в виде логических состояний. С этапа ввода они передаются на этап обработки в предварительно обработанном формате.

Этап обработки

На этапе обработки происходит обработка преобразованных сигналов с этапа ввода и их комбинирование с помощью логических операций и других функций. Память программы этапа обработки является полностью программируемой. Последовательность обработки можно изменять в любое время путем модифицирования или замены программы, хранящейся в памяти.

Этап вывода

Результаты обработки программой входных сигналов передаются на этап вывода, где они осуществляют функции управления подключенными переключаемыми элементами, такими как контакторы, сигнальные лампы, электромагнитные клапаны и т.д.

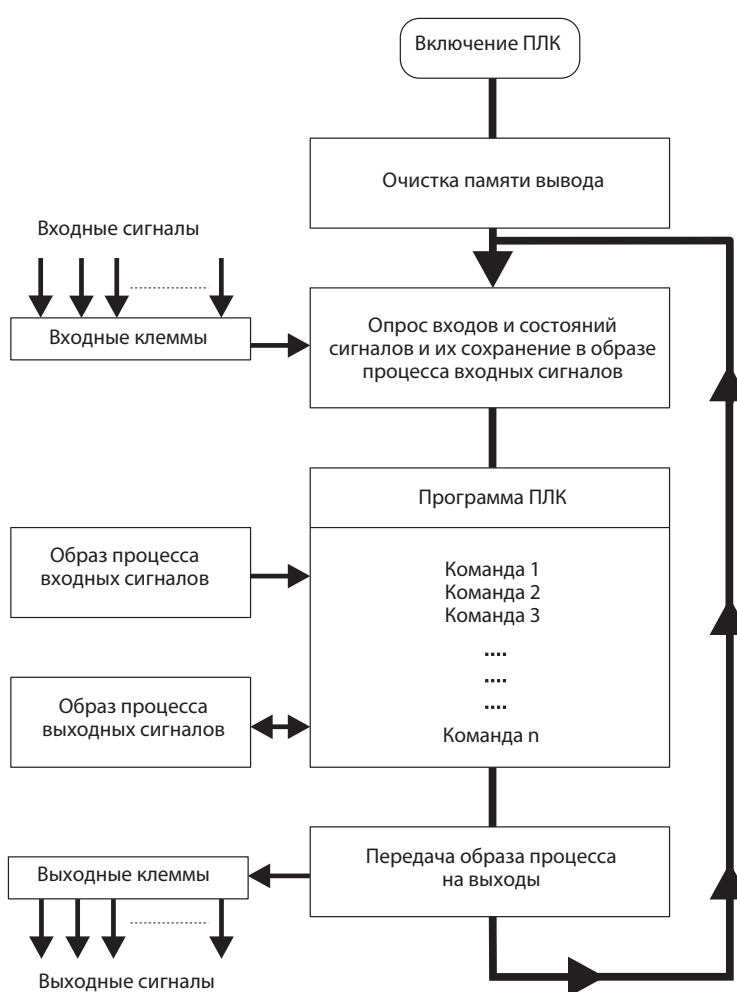
2.2 Обработка программ ПЛК

ПЛК выполняет свои задачи путем исполнения программы, которая обычно разрабатывается вне контроллера, затем передается в его память программы. Прежде чем приступить к программированию, полезно ознакомиться с принципами обработки ПЛК данных программ.

Программа ПЛК состоит из последовательности команд, управляющих функциями контроллера. ПЛК выполняет данные управляющие команды последовательно, одну за другой. Вся последовательность программы циклическая, то есть она повторяется в непрерывном цикле. Продолжительность одного повтора программы называется временем или периодом программного цикла.

Обработка образа процесса

При обработке программы в ПЛК входные и выходные сигналы используются не напрямую, а в виде «образа процесса»:



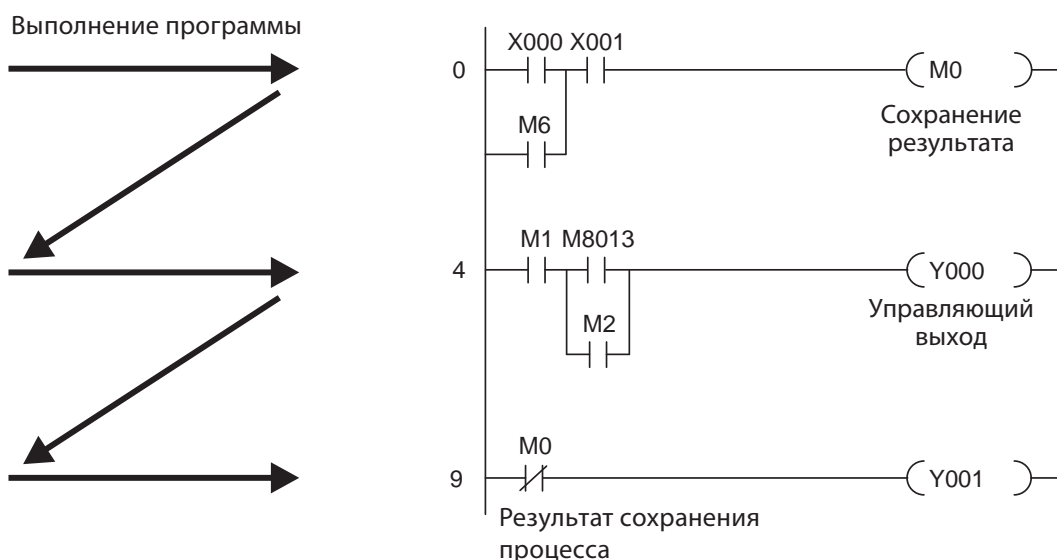
Образ процесса входных сигналов

В начале каждого программного цикла система производит опрос состояний сигналов на входах и сохраняет их в буфере, создавая «образ процесса» входных сигналов.

Выполнение программы

После этого происходит выполнение программы, в процессе которого ПЛК обращается к состояниям входных сигналов, сохраненным в образе процесса. Это означает, что любые последующие изменения состояний входных сигналов не регистрируются до следующего программного цикла.

Программа выполняется последовательно от начала и до конца – в том порядке, в каком были запрограммированы данные команды. Результаты выполнения отдельных шагов программы сохраняются и используются в текущем программном цикле.



Образ процесса выходных сигналов

Результаты логических операций, относящиеся к выходным сигналам, сохраняются в буфере вывода в виде образа процесса выходных сигналов. Образ процесса выходных сигналов хранится в буфере вывода до его перезаписи. После записи значений в виде выходных сигналов данный программный цикл повторяется.

Отличия обработки сигналов в ПЛК и контроллерах на релейно-контактной логике

В контроллерах на релейно-контактной логике программа определяется функциональными элементами и их подключением. Все операции управления выполняются одновременно (параллельное выполнение). При любом изменении в состоянии входного сигнала моментально изменяется состояние соответствующего выходного сигнала.

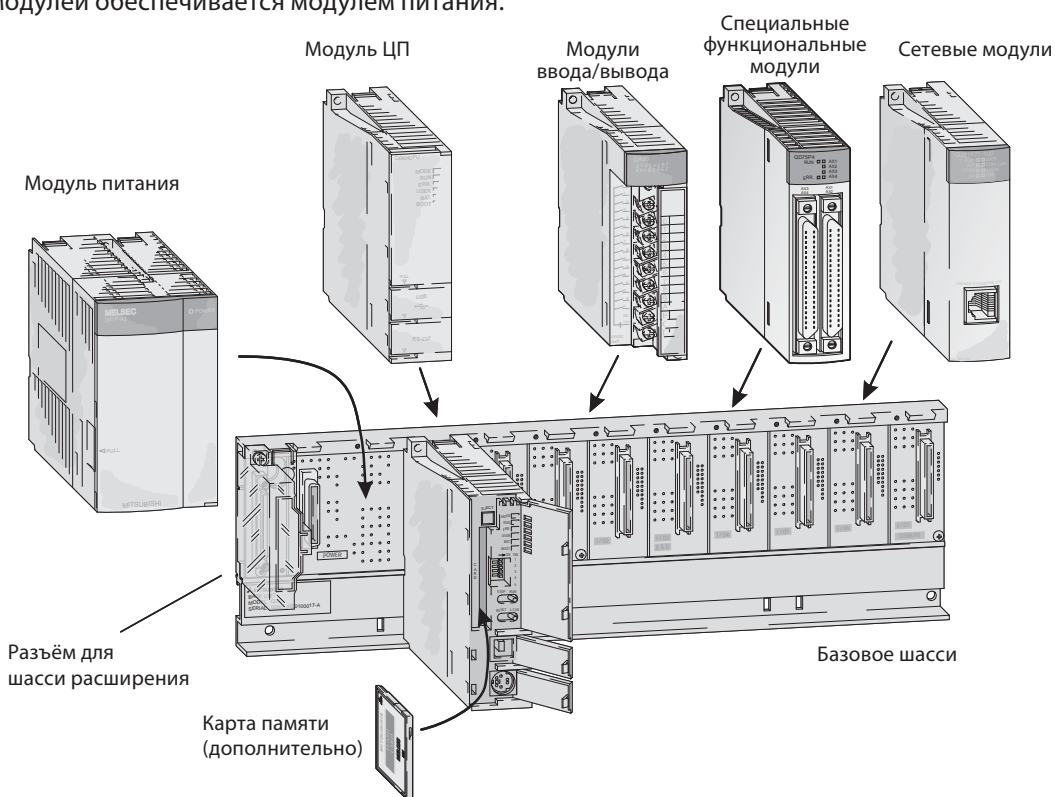
ПЛК не реагирует на изменения в состоянии входных сигналов до следующего после данных изменений программного цикла. В настоящее время этот недостаток в значительной степени компенсируется крайне короткими периодами программных циклов. Длительность периода программного цикла зависит от количества и типа выполняемых команд.

3 Контроллер System Q

3.1 Конфигурация системы

Контроллер System Q представляет собой мощный модульный ПЛК с мультипроцессорной технологией. Модульность означает возможность подбора отдельной, оптимальной конфигурации системы для конкретного применения.

Основными компонентами ПЛК являются базовое шасси, модуль питания и по крайней мере один модуль центрального процессора (ЦП). ЦП выполняет команды программы ПЛК. В зависимости от применения на базовое шасси можно устанавливать другие модули, например модули ввода/вывода и специальные функциональные модули. Питание установленных модулей обеспечивается модулем питания.



Связь между отдельными модулями и ЦП осуществляется по внутренней шине базового шасси.

Шасси, на котором установлен ЦП, называется базовым шасси. Базовые шасси System Q существуют в 5 различных версиях: с количеством слотов для установки модулей до 12.

Расширяемость

Если требуются дополнительные слоты для модулей, каждое базовое шасси можно дополнить шасси расширения. Шасси расширения подключаются к базовому шасси с помощью кабелей расширения. При использовании шасси расширения, не имеющих собственных модулей питания, данные кабели также обеспечивают питание установленных модулей. К базовому шасси можно подключать до 7 шасси расширения. Максимально возможное количество модулей ввода/вывода и специальных функциональных модулей на всех шасси составляет 64.

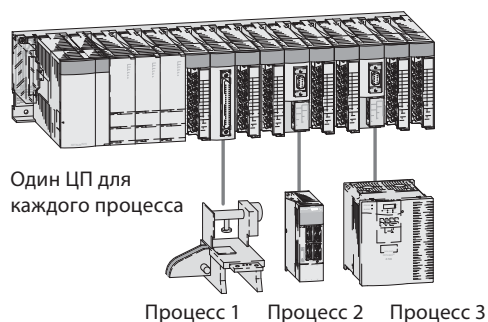
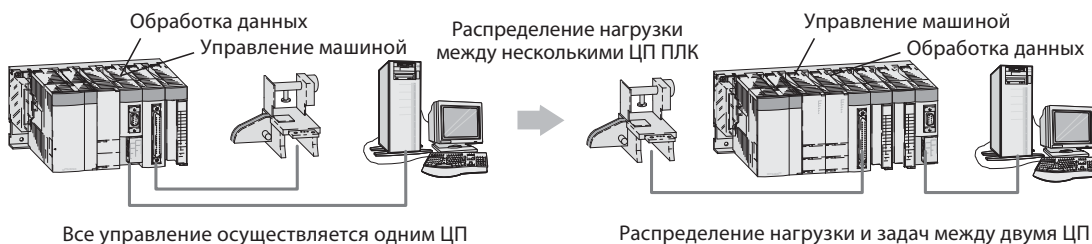


При выборе модуля питания необходимо учитывать общую потребляемую мощность модулей ввода/вывода, специальных функциональных модулей и периферийных устройств. В случае необходимости следует использовать шасси расширения с дополнительным модулем питания.

При монтаже на крупных производствах, а также для машин с модульной конфигурацией использование удаленных вводов и выводов (удаленных станций ввода/вывода), размещенных непосредственно на технических объектах, обеспечивает целый ряд преимуществ. Это позволяет сократить длину линий подключения между входами/выходами и датчиками/приводами. Для подключения станции удаленного ввода/вывода и системы к ЦП ПЛК требуется только сетевой модуль и сетевой кабель. В зависимости от выбранного типа ЦП могут адресоваться до 4096 локальных точек ввода/вывода (на базовом шасси и шасси расширения) и до 8192 удаленных.

Распределение нагрузки между несколькими ЦП ПЛК

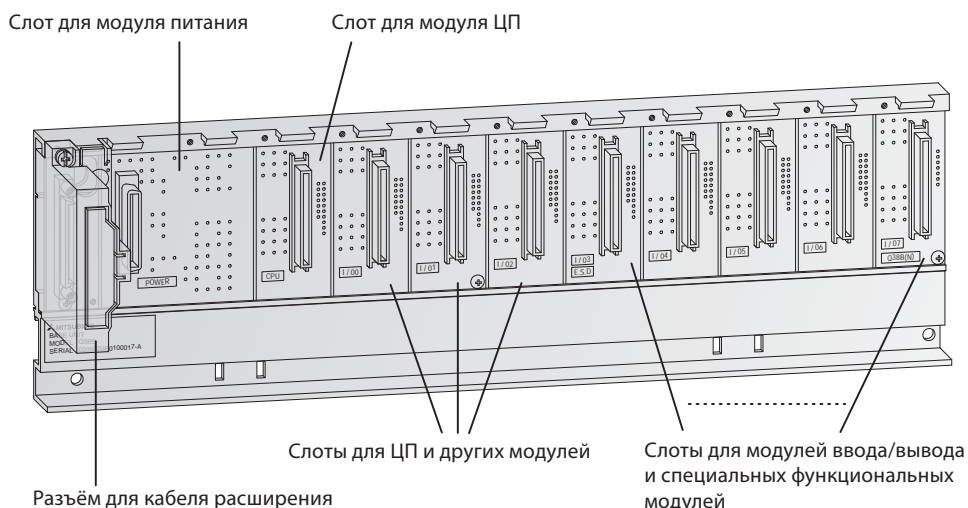
Несколько ЦП ПЛК серии System Q можно использовать вместе, чтобы обеспечить выполнение одной системой действий, которые различаются по тактовому времени, например, управление очередностью выполнения операций и обработку данных. Таким образом, управление очередностью выполнения операций и обработку данных можно распределить между разными ЦП.



Если из-за большого объема программы нагрузка на крупномасштабную систему превышает возможности обработки, использование нескольких ЦП для распределения нагрузки улучшает производительность всей системы.

3.2 Базовые шасси

На базовых шасси имеются слоты для модуля питания, модулей ЦП (до четырех) и для модулей ввода/вывода и специальных функциональных модулей. На шасси расширения можно устанавливать модули ввода/вывода и специальные функциональные модули. Базовые шасси устанавливаются непосредственно на монтажной поверхности с помощью винтов или на DIN-рейке с использованием переходников.



В следующей таблице приведены существующие базовые шасси.

Параметр	Базовое шасси				
	Q33B	Q35B	Q38B	Q38RB	Q312B
Подключаемые модули питания	1	1	1	2*	1
Количество слотов для модулей ввода/вывода и специальных функциональных модулей	3	5	8	8	12

* На базовом шасси Q38RB можно использовать резервные модули питания.

Параметр	Шасси расширения						
	Q52B	Q55B	Q63B	Q65B	Q68B	Q68RB	Q612B
Подключаемые модули питания	—	—	1	1	1	2*	1
Количество слотов для модулей ввода/вывода и специальных функциональных модулей	2	5	3	5	8	8	12

* На шасси расширения Q68RB можно использовать резервные модули питания.

3.2.1 Кабели шасси расширения

Кабели шасси расширения служат для подключения шасси расширения к базовому шасси. Суммарная длина всех кабелей расширения не должна превышать 13,2 м.

Тип	QC05B	QC06B	QC12B	QC30B	QC50B	QC100B
Длина провода	0,45 м	0,50 м	1,2 м	3,0 м	5,0 м	10,0 м

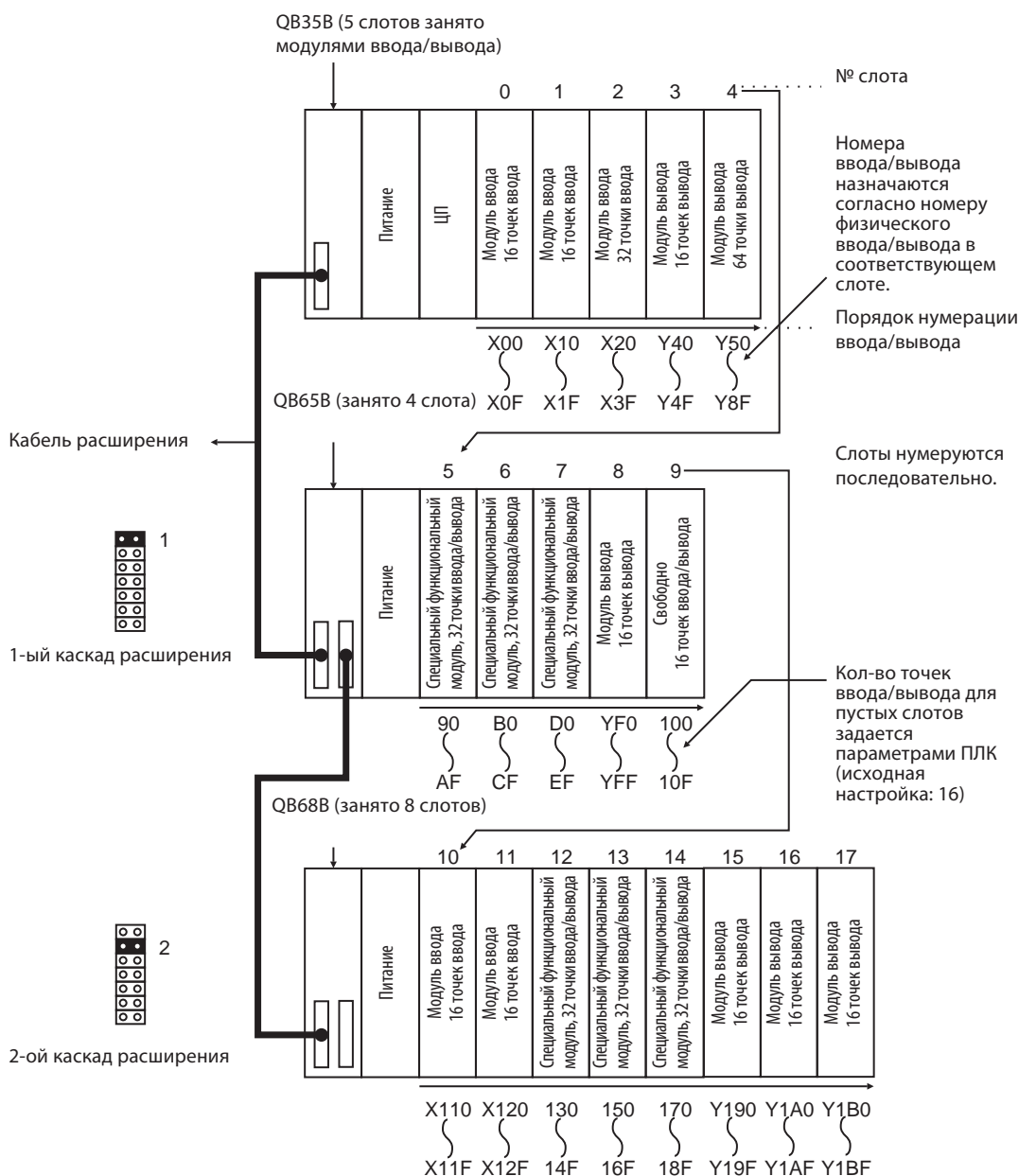
Для подключения шасси расширения, не имеющих собственных модулей питания (Q52B, Q55B), рекомендуется использовать кабель QC05B.

3.2.2 Назначение адресов ввода/вывода

Для адресации вводов и выводов ПЛК в программе они должны иметь уникальную идентификацию. Это осуществляется назначением номера каждому вводу и выводу – адреса ввода/вывода (см. также раздел 4.1). Данные адреса исчисляются шестнадцатеричными числами. (Подробнее о разных системах счисления см. ниже раздел 4.3)

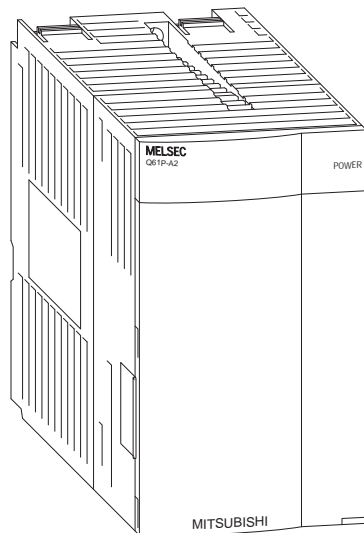
ЦП контроллера System Q автоматически распознаёт слоты, имеющиеся в базовом шасси и шасси расширения, и назначает соответствующие адреса вводам и выводам.

Однако данное назначение можно также выполнять с помощью программного пакета для программирования. Таким образом, слоты можно оставлять пустыми и резервировать адреса для будущих расширений.



Каскад расширения определяется установкой перемычек на шасси расширения.

3.3 Модули питания



Для питания контроллеров System Q предусмотрено постоянное напряжение 5 В. Существуют модули питания с входным напряжением 24 В пост. и 240 В перем. тока.

Выходное напряжение модуля питания (5 В пост.) подается напрямую на базовое шасси и отсутствует на внешних клеммах.

Помимо постоянного напряжения 5 В, модуль питания Q62P обеспечивает постоянное напряжение 24 В, для питания периферийных устройств, таких как датчики. Максимальная нагрузка для данного выхода составляет 0,6 А.

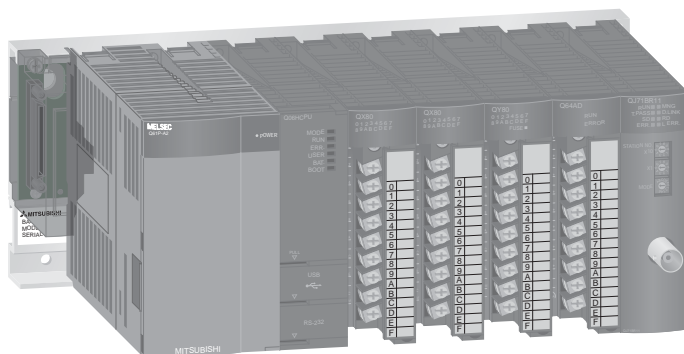
Параметр	Q63P	Q63RP	Q61P-A1	Q61P-A2	Q62P		Q64P	Q64RP
Входное напряжение	24 В пост.		100–120 В пер.	200–220 В пер.	100–240 В пер.		100–120 В пер. 200–240 В пер.	
Потребляемая мощность	45 Вт	65 Вт	105 ВА	105 ВА	105 ВА		105 ВА	160 ВА
Выходное напряжение	5 В пост.		5 В пост.		5 В пост.	24 В пост.	5 В пост.	
Выходной ток	6 А	8,5 А	6 А	6 А	3 А	0,6 А	8,5 А	

Модули питания Q63RP и Q64RP являются резервируемыми и могут использоваться в сочетании с любыми ЦП, кроме типа Q00JCPU. Для систем с резервным питанием на резервированное базовое шасси необходимо устанавливать два резервных модуля питания. Таким образом повышается работоспособность системы, поскольку в случае отказа одного модуля питания вступает в действие другой модуль. Для резервных модулей питания возможна «горячая» замена, то есть замена во время работы системы (в режиме RUN).

Выбор подходящего модуля питания

Общее потребление тока установленных модулей должно быть меньше номинального выходного тока модуля питания. Если потребление тока слишком большое, необходимо сократить количество модулей на базовом шасси.

Примера расчета общего потребления тока



Модуль	Описание	Потребление тока
Q06H-CPU	Модуль ЦП	0,64 А
QX80	Модуль дискретного входа	0,16 А
QX80	Модуль дискретного входа	0,16 А
QY80	Модуль дискретного выхода	0,008 А
Q64AD	Модуль АЦП	0,63 А
QJ71BR11	Модуль MELSECNET/H	0,75 А
Общее потребление тока		2,42 А

Общее потребление тока составляет 2,42 А.

Установленный модуль питания способен обеспечивать ток 6 А. Данная конфигурация будет работать без проблем.

3.4 Модули ЦП

Для контроллеров System Q предусмотрено 19 различных ЦП, благодаря чему обеспечивается высокая производительность. На одно базовое шасси можно устанавливать до четырех модулей ЦП для решения задач управления и коммуникации. Как и для других контроллеров компании «Мицубиси», мощность устройств System Q можно наращивать по мере роста потребностей данного применения – нужно просто заменять или добавлять модули ЦП.

Модули ЦП делятся на три категории:

- Модули ЦП ПЛК

Модули ЦП ПЛК в рамках System Q выполняет традиционные задачи ПЛК. Данный ЦП выполняет программу ПЛК, опрашивает входы, управляет выходами и обменивается данными специальными функциональными модулями.

- Модули ЦП управления процессом

Модули ЦП управления процессом System Q обладают функциональностью модулей ЦП ПЛК и обеспечивают дополнительные расширенные функции ПИД-регулирования и встроенные функции управления процессом с 52 специальными командами. Таким образом данные модули ЦП подходят для применения в сложных условиях, например, в химической промышленности.

- Модули резервированных ЦП управления процессом

Обладая всеми функциями модулей ЦП управления процессом, модули резервированных ЦП управления процессом System Q обеспечивают максимальную работоспособность систем для критически важных задач управления процессами и автоматизации производства.

Резервированная система состоит из двух соединенных кабелем ПЛК с одинаковой конфигурацией (модули питания и ЦП, сетевые модули и т.д.). Один ПЛК управляет процессом, а второй находится в «горячем» резерве. При сбое в активной системе «горячая» резервная система включается автоматически и вступает в действие без прерывания процесса. Это существенно сокращает издержки и расходы, связанные с простоями и перезапуском.

- Модуль ЦП ПК

Модуль ЦПУ ПК — это компактный высококачественный персональный компьютер, который может устанавливаться на базовом шасси. Он может использоваться как для обычных ПК-применений в качестве ведущего Q-ПК, так и для ПЛК-применений. Следовательно, он пригоден для использования в качестве ПК, интегрированного в систему управления — например, для визуализации, баз данных, функций слежения, приложений Microsoft или для программирования System Q на языке высокого уровня. Кроме того, данной системой можно управлять как программным ПЛК в соответствии с IEC1131, используя дополнительное программное обеспечение SX-Controller.

Для подключения к периферийным устройствам можно использовать модули ввода/вывода и специальные функциональные модули System Q.

- С-контроллер

С-контроллер обеспечивает возможность объединения и программирования для платформы автоматизации System Q на языке C++. При использовании принятой во всем мире операционной системы реального времени VxWorks реализация сложных задач, связи и протоколирования становится достаточно легкой.

- Модули ЦП управления движением

Модуль ЦП контроллера движения управляет и синхронизирует подключенные к нему сервоусилители и серводвигатели. Для системы управления движением требуются ЦП контроллера и ЦП ПЛК. Инновационная автономная систем управления движением

создается только путем объединения ПЛК с высокودинамичной системой управления позиционированием.

В такой системе процессорный модуль движения управляет крупномасштабными серво перемещениями, а ЦП ПЛК отвечает за управление механизмом и связь.

В данном руководстве подробно описан только модуль ЦП ПЛК. Подробная информация по другим модулям ЦП содержится в техническом каталоге System Q, кат. № 136731 и в руководствах для отдельных модулей.

Модули ЦП ПЛК

● Q00JCPU

Комбинация модулей ЦП, модуля питания и 5-слотового базового шасси образует неразделимый блок. При использовании модуля Q00JCPU работа в многопроцессорном режиме невозможна.

- Объем памяти для программы: 8 тыс. шагов
- Время выполнения логической инструкции: 0,2 мкс

Все следующие ЦП ПЛК способны работать в многопроцессорном режиме.

● Q00CPU

- Объем памяти для программы: 8 тыс. шагов
- Время выполнения логической инструкции: 0,16 мкс

● Q01CPU

- Объем памяти для программы: 14 тыс. шагов
- Время выполнения логической инструкции: 0,10 мкс

● Q02CPU

- Объем памяти для программы: 28 тыс. шагов
- Время выполнения логической инструкции: 0,079 мкс

● Q02HCPU

- Объем памяти для программы: 28 тыс. шагов (возможность расширения с помощью карты памяти)
- Время выполнения логической инструкции: 0,034 мкс

● Q06HCPU

- Объем памяти для программы: 60 тыс. шагов (возможность расширения с помощью карты памяти)
- Время выполнения логической инструкции: 0,034 мкс

● Q12HCPU

- Объем памяти для программы: 124 тыс. шагов (возможность расширения с помощью карты памяти)
- Время выполнения логической инструкции: 0,034 мкс

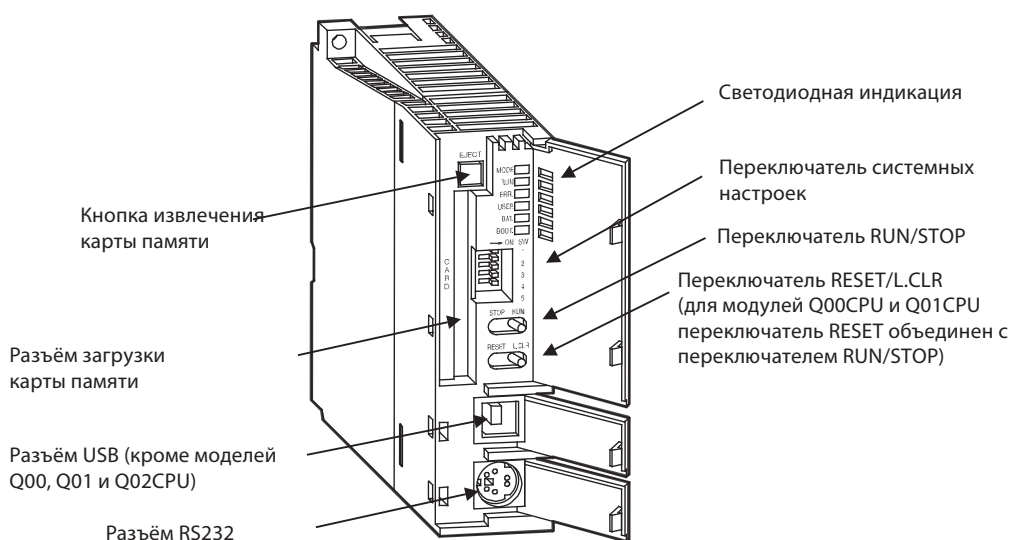
● Q25HCPU

- Объем памяти для программы: 252 тыс. шагов (возможность расширения с помощью карты памяти)
- Время выполнения логической инструкции: 0,034 мкс

В следующей таблице приведены возможности расширения и количество входов и выходов для модулей ЦП ПЛК.

Модуль ЦП	Кол-во подключаемых шасси расширения	Кол-во устанавливаемых модулей	Количество точек ввода/вывода	
			Локальных (на базовом шасси и шасси расширения)	Удаленных
Q00JCPU	2	16	256	2048
Q00CPU	4	24	1024	2048
Q01CPU				
Q02CPU	7	64	4096	8192
Q02HCPU				
Q06HCPU				
Q12HCPU				
Q25HCPU				

3.4.1 Компоненты модулей ЦП



Светодиодная индикация

– Светодиоды MODE и RUN

Состояние светодиода	Описание
Зеленый:	режим Q
ВКЛ.:	При работе в режиме RUN
ВЫКЛ.:	В режиме STOP или после обнаружения ошибки, из-за которой работа остановлена
Мигание:	Переключатель RUN/STOP переведен из положения STOP в RUN после записи программы или параметра в режиме STOP. ЦП не перешел в режим RUN.

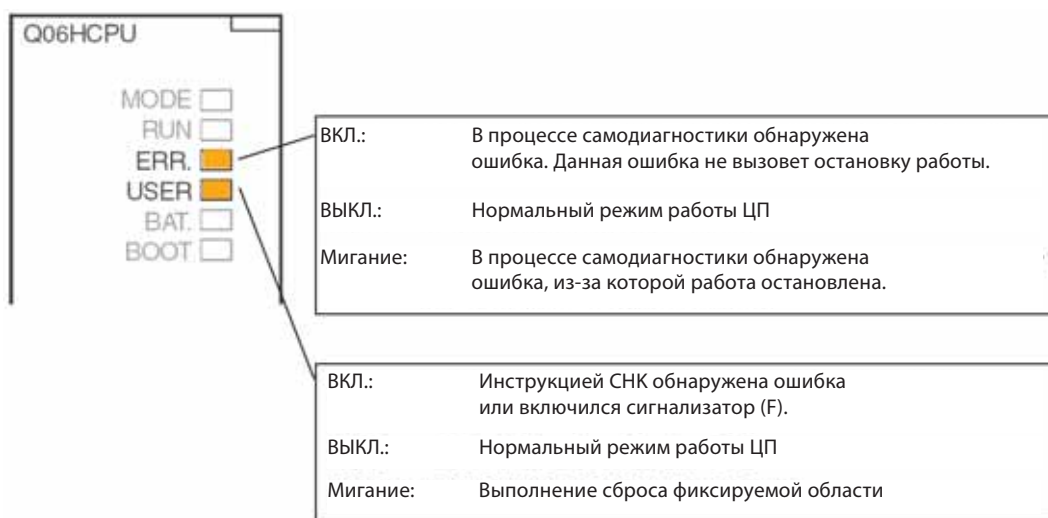
Процедура переключения ЦП ПЛК из режима STOP в режим RUN после изменения программы или параметра в режиме STOP:

- ① Перевести переключатель RESET/L.CLR в положение RESET.
- ② Перевести переключатель RUN/STOP из положения STOP в положение RUN.

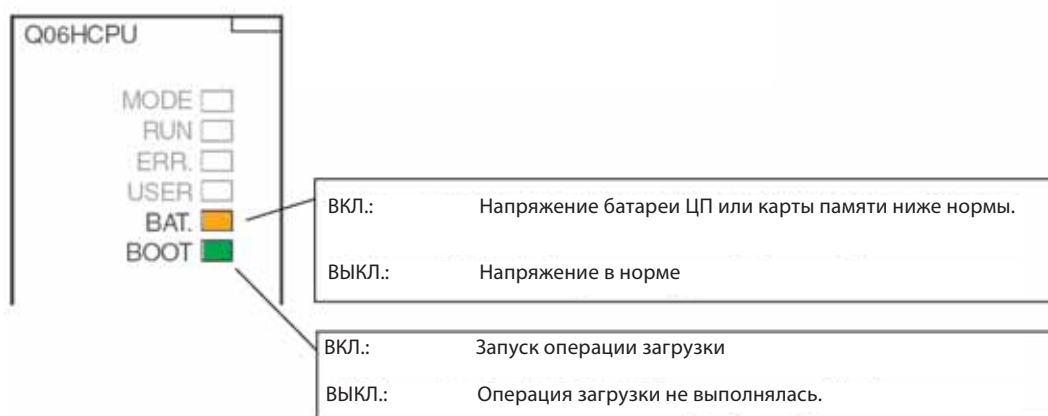
Перевод ЦП в режим RUN без сброса данных для устройства:

- ① Перевести переключатель RUN/STOP из положения STOP в положение RUN.
- ② Перевести переключатель RUN/STOP обратно в положение STOP.
- ③ Перевести переключатель RUN/STOP в положение RUN.

– **Светодиоды ERR. и USER**



– **Светодиоды BAT и BOOT**



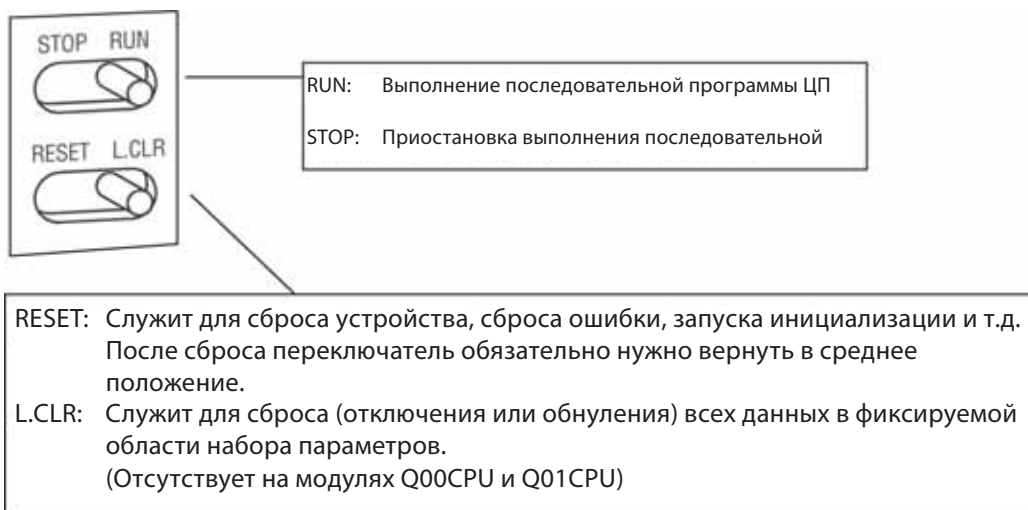
Системные переключатели



Во встроенном ОЗУ (привод 3) параметры не сохраняются (см. также раздел 3.4.2).

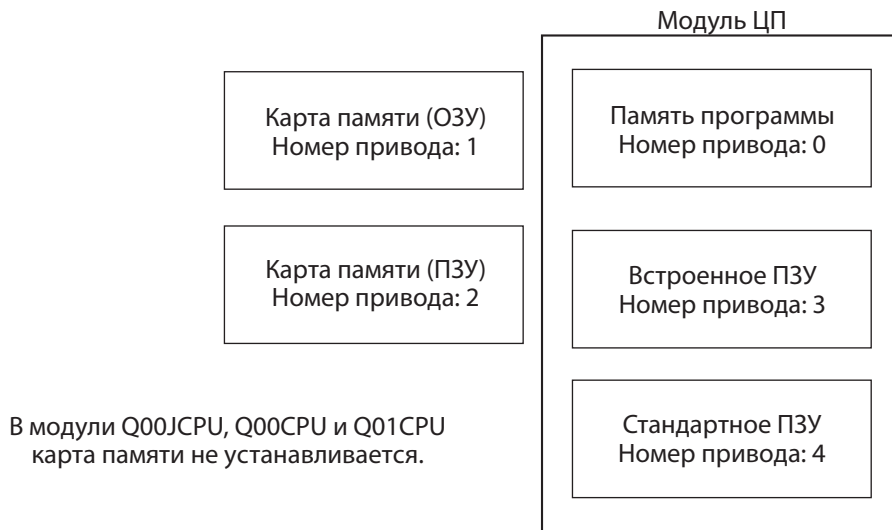
В состоянии поставки все переключатели находятся в положении ВЫКЛ.

Переключатели RUN/STOP Switch и RESET/L.CLR



3.4.2 Организация памяти

В модулях ЦП ПЛК используется несколько типов памяти. Типы памяти идентифицируются по номерам их приводов. Высокопроизводительные модули ЦП в дополнение к встроенной памяти снабжены слотом для карты памяти.



Организация хранения данных

- Q00JCPU, Q00CPU и Q01CPU

Данные	Встроенная память		
	Память программы (привод 0)	ОЗУ (привод 3)	ПЗУ (привод 4)
Программа	●	○	●
Параметры	●	○	●
Параметры модулей интеллектуальных функций	●	○	●
Информация устройства	●	○	●
Файловый регистр	○	●	○

- – Хранение возможно
- – Хранение невозможно

● Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU и Q25HCPU

Данные	Встроенная память			Карты памяти		
	Память программы (устройство 0)	ОЗУ (устройство 3)	ПЗУ (устройство 4)	ОЗУ (устройство 1)	Флэш-ПЗУ (устройство 2)	АТА-ПЗУ (устройство 2)
Программа	●	○	●	●	●	●
Параметры	●	○	●	●	●	●
Параметры модулей интеллектуальных функций	●	○	●	●	●	●
Информация устройства	●	○	●	●	●	●
Исходное значение устройства	●	○	●	●	●	●
Файловый регистр	○	●	○	●	●	○
Локальные устройства	○	●	○	●	○	○
Отладочные данные	○	○	○	●	○	○
Журнал ошибок	○	○	○	●	○	○
Перезапись файла данных командой FWRITE	○	○	○	○	○	●

● – Хранение возможно

○ – Хранение невозможно

Программа хранится в стандартном ПЗУ или на карте памяти (ОЗУ или ПЗУ), при включении питания передается в память программы и там выполняется. Поэтому при хранении программы в стандартном ПЗУ или на карте памяти (ОЗУ или ПЗУ) для размещения программы в памяти программы требуется дополнительное место.

Чтобы использовать отладочные данные для трассировки, журнал ошибок или файл общего назначения, необходимо вставить карту памяти.

Сведения о хранении данных

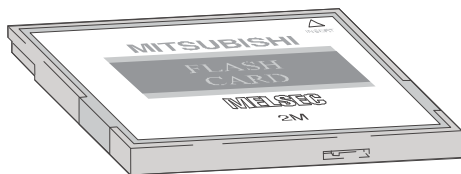
- Программа
Файл программы в формате Ladder, List или SFC-последовательности. При выполнении нескольких программ их файлы хранятся в памяти.
- Параметры
Файл с параметрами ПЛК или сетевыми параметрами, заданными при программировании.
- Параметр модуля специальной функции
Файл параметров, заданных с помощью программы GX Configurator. Данный файл не существует, если не используется настройка, заданная с помощью программы GX Configurator.
- Информация устройства
Файл с информацией по каждому устройству ЦП. Данный файл не существует, если информация устройства не создана.
- Исходное значение устройства
Список значений, присваиваемых устройствам в модуле ЦП при включении питания. Данный файл не существует, если исходные значения устройств не используются.

- **Файловый регистр**
Файл файлового регистра (R, ZR). Присваивая файлам разные имена, можно хранить несколько файлов файлового регистра. Файловые регистры можно хранить на карте памяти ПЗУ (привод 2) и нельзя хранить на карте АТА (Q2MEM-8MBA/16MBA/32MBA). Файловые регистры, хранящиеся на карте флэш-памяти, в программе можно только читать, без изменения данных.
- **Локальные устройства**
Локальные устройства используются только соответствующими программами при наличии нескольких программ. При обработке какой-либо программы данные соответствующего локального устройства передаются из области локального устройства в область существующего устройства, после чего выполняется обработка программы.
- **Отладочные данные**
Файл с результатами трассировки, используемыми для отслеживания при отладке программы.
- **Перезапись файла данных командой FWRITE**
Эти данные можно хранить только на картах памяти АТА (Q2MEM-8MBA/16MBA/32MBA).

Карты памяти

Со всеми модулями ЦП ПЛК серии System Q (кроме модулей Q00JCPU, Q00CPU и Q01CPU) можно использовать карты памяти. Перед первым использованием карту памяти необходимо отформатировать с помощью GX Developer или GX IEC Developer.

Программа хранится на карте памяти, при включении питания передается в память программы и там выполняется. Состояние при включении питания задается параметрами (в загрузочном файле).

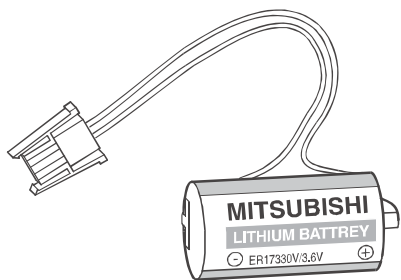


Переключатель защиты записи на карте предотвращает случайную перезапись хранящихся данных. Батарея в карте памяти ОЗУ обеспечивает хранение данных при перебоях питания.

Типы карт памяти

Обозначение	Тип памяти	Объем памяти [байтов]	Объем памяти [кол-во файлов]	Кол-во перезаписей
Q2MEM-1MBS	Стат. ОЗУ	1011 k	256	Неограниченное
Q2MEM-2MBS		2034 k	288	
Q2MEM-2MBF	Флэш-ПЗУ	2035 k	288	100 000
Q2MEM-4MBF		4079 k		
Q2MEM-8MBA	АТА-ПЗУ	7940 k	512	1 000 000
Q2MEM-16MBA		15932 k		
Q2MEM-32MBA		31854 k		

3.4.3 Установка батареи для модуля ЦП

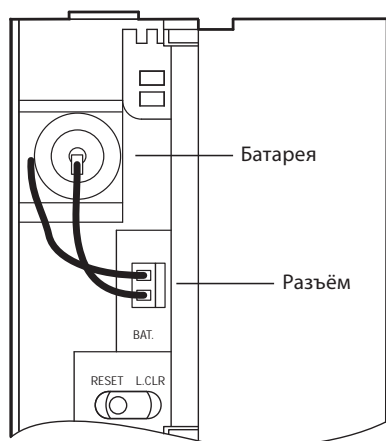


Модуль ЦП ПЛК серии System Q снабжен батареей. При сбое электропитания батарея обеспечивает хранение данных в памяти программы, питание встроенного ОЗУ и часов в течение нескольких тысяч часов. Однако период времени зависит от типа ЦП.

Батарею необходимо менять через каждые 10 лет.

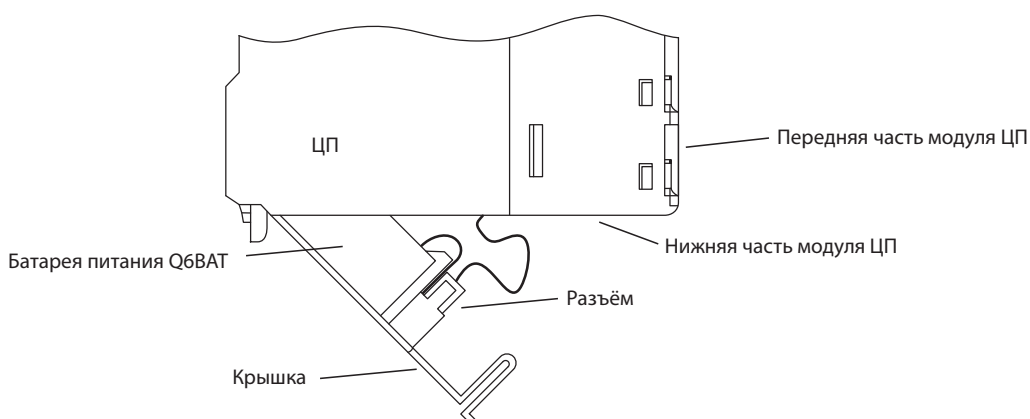
Карты памяти статического ОЗУ снабжены собственной батареей (Q2MEM-BAT), поэтому их питание не зависит от батареи модуля ЦП.

В состоянии поставки батарея ЦП установлена, но её разъём не подключен во избежание разряда и короткого замыкания. Перед первым использованием модуля ЦП необходимо подключить батарею.



На модулях ЦП Q00J, Q00 и Q01 батарея устанавливается за верхней крышкой в передней части модуля ЦП.

На всех остальных модулях ЦП ПЛК батарея устанавливается в нижней части.



Чтобы подключить батарею, откройте крышку батарейного отсека модуля ЦП. Убедитесь, что батарея установлена правильно. Вставьте разъём батареи в гнездо на корпусе. При использовании модулей ЦП Q02(H), Q06H, Q12(P)H и Q25(P)H CPU разъём вставляется в соответствующий держатель, расположенный на крышке батарейного отсека.

3.5 Модули дискретных входов и выходов

Модули входов и выходов обеспечивают подключение ЦП ПЛК к контролируемому процессу. Дискретные входы служат для ввода управляющих сигналов с подключенных выключателей, кнопок или датчиков. На данных входах считываются значения ВКЛ (наличие сигнала питания) и ВЫКЛ (отсутствие сигнала питания). Модули дискретных выходов могут включать и отключать внешние приводы.

Входные сигналы могут поступать со следующих устройств:

- Кнопки
- Поворотные переключатели
- Клавишные переключатели
- Концевые выключатели
- Датчики уровня
- Датчики расхода
- Фотоэлектрические приёмники
- Бесконтактные датчики (индуктивные или емкостные)

Бесконтактные датчики обычно имеют транзисторный выход, который может быть транзистором типа NPN (переключающий на минус) или PNP (переключающий на плюс).

Выходные сигналы могут использоваться для управления следующими устройствами:

- Реле и контакторы
- Сигнальные лампы
- Соленоиды
- Входы других устройств, таких как инверторы

Обзор модулей дискретного ввода/вывода

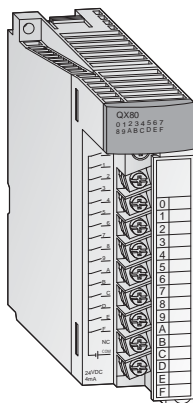
Тип		Кол-во входов/выходов			
		8	16	32	64
Модули входа	120 В пер.	○	●	○	○
	240 В пер.	●	○	○	○
	24 В пост.	○	●	●	●
	24 В пост. (высокоскорост.)	●	○	○	○
	5/12 В пост.	○	●	●	●
Модули выходов	Реле	●	●	○	○
	Отдельные реле	●	○	○	○
	Тиристорный выход	○	●	○	○
	Транзисторный выход (переключающий на минус)	●	●	●	●
	Транзисторный выход (переключающие на плюс)	○	●	●	○
Комбинированные модули входов/выходов		●	○	●	○

● – Модуль существует

○ – Модуль не существует

3.5.1 Модули дискретных входов

Существуют модули входов с различным входным напряжением.



Кол-во входов / Входное напряжение	Модули входов System Q			
	8	16	32	64
5–12 В пост.		QX70	QX71	QX72
24 В пост.		QX40 QX80	QX41 QX81	QX42 QX82
24 В пост. (модуль прерываний)		QI60		
48 В пер./пост.		QX50		
100–120 В пер.		QX10		
100–240 В пер.	QX28			

Модули с 8 и 16 точками подключения снабжены съёмными колодками с винтовыми клеммами. Модули с 32 и 64 точками подключаются через разъём.

Общие сведения о входах ПЛК

Развязка между всеми входами обеспечивается посредством оптронных пар. Это защищает чувствительные элементы электроники ЦП в ПЛК от импульсных электрических помех от внешнего оборудования.

Еще одной общей проблемой является дребезг контактов электромеханических выключателей. Чтобы исключить его негативное влияние на ПЛК, входы фильтруются так, чтобы регистрация состояния ВКЛ осуществлялась только тогда, когда сигнал остается стабильным в течение промежутка времени, превышающего коэффициент фильтра.

ПРИМЕЧАНИЕ

Коэффициент фильтра стандартного модуля входов предварительно устанавливается на 10 мс, но его можно отдельно настроить в диапазоне от 1 до 70 мс при настройке параметров ЦП (см. технические данные конкретных модулей).

При программировании следует учитывать время отклика фильтра, поскольку от него будет напрямую зависеть работа программы. При использовании высокоскоростного входа для дискретных сигналов следует осторожно применять пониженный коэффициент фильтра. Провода должны быть экранированными и прокладываться отдельно от других потенциальных источников электрических помех. Если требуется очень высокоскоростная работа системы, следует использовать специальные модули, такие как модуль прерываний QI60.

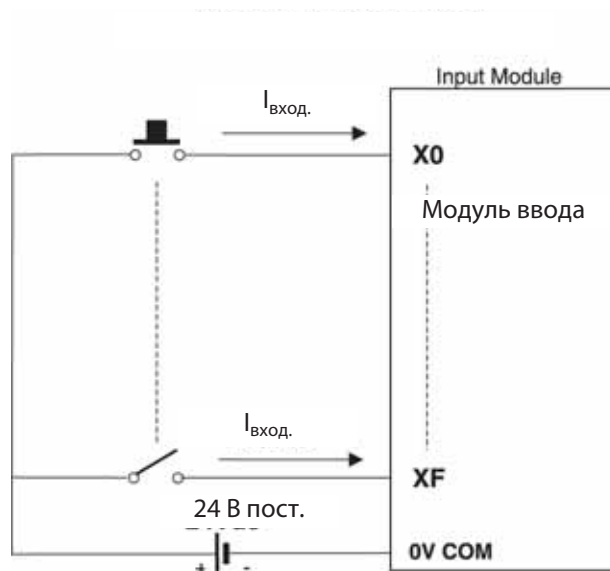
Для регистрации ПЛК изменения логического состояния входа через него должен протекать минимальный ток. Величина минимального тока зависит от типа модуля входов и в большинстве случаев составляет 3 мА. Если ток будет меньше, вход не включится, даже если подключенный к нему датчик будет включен. Входной ток ограничивается входным сопротивлением. Если входное напряжение превышает номинальную величину, входной ток также возрастает. Вход рассчитан на сигнал до 7 мА, ток выше этого уровня может вызывать повреждение входа.

В начале каждого программного цикла ЦП ПЛК производит опрос состояний сигналов на входах и сохраняет их. При выполнении программы ЦП обращается к сохраненным состояниям входных сигналов. Перед выполнением следующего программного цикла сохраненное состояние снова обновляется.

Для аппаратуры System Q предусмотрены модули входов сигналов постоянного тока для подключения как с общим минусом, так и с общим плюсом. Для некоторых модулей, таких как QX71, можно выбрать любой из этих способов подключения.

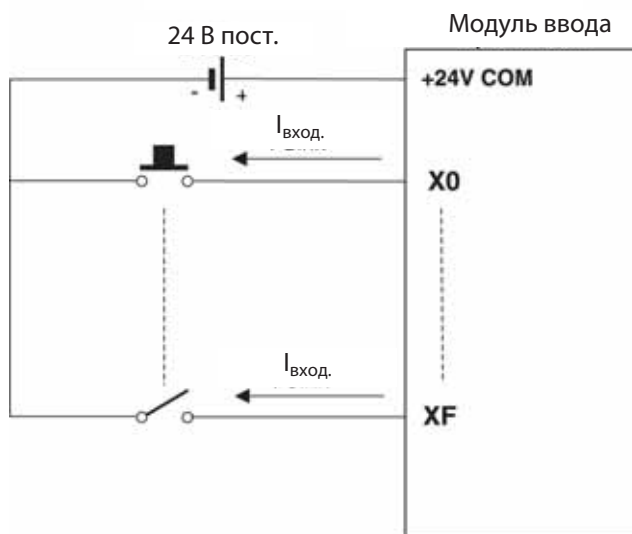
Подключение с общим минусом

Датчик, подключенный к модулю входов с общим минусом, соединяет вход ПЛК с положительным полюсом внешнего источника питания. Отрицательный полюс источника питания подключается к общей клемме для всех входов данной группы. При активации датчика ток входного сигнала протекает в направлении к входу.



Подключение с общим плюсом

Датчик, подключенный к модулю входов с общим плюсом, соединяет вход ПЛК с отрицательным полюсом внешнего источника питания. Положительный полюс источника питания подключается к общей клемме для всех входов данной группы. При активации датчика ток входного сигнала протекает в направлении от входа.



Бесконтактные и оптические датчики

Бесконтактные датчики выдают сигнал на ПЛК, когда объект находится в непосредственной близости от датчика. Контакт объекта с датчиком не требуется. Это преимущество обеспечивает широкое применение данных датчиков. Существует два типа бесконтактных датчиков – индуктивные и емкостные.

В промышленности также широко применяются самые разные оптические датчики.

В большинстве оптических и бесконтактных датчиках используются полупроводниковые выходы двух полярностей:

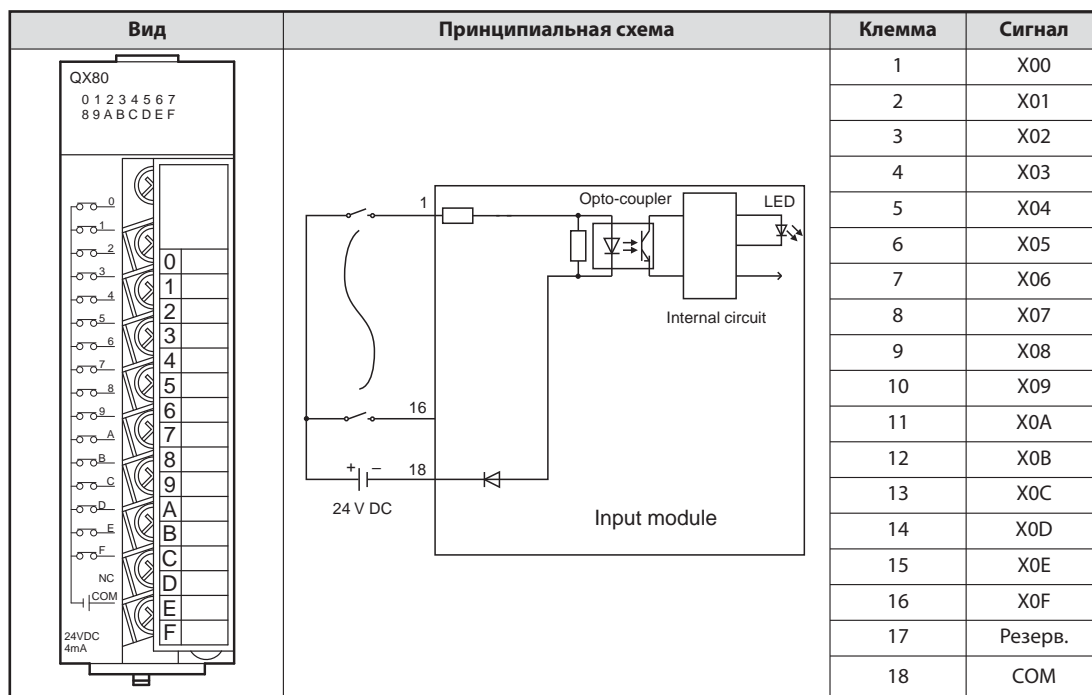
- PNP (переключающие на плюс)
- NPN (переключающие на минус)

Напряжение питания данных датчиков обычно составляет 24 В пост.

Пример модуля входов с общим минусом

Параметр	Технические данные
Тип модуля	QX80
Количество точек входа	16
Способ изоляции	Оптронная пара
Номинальное входное напряжение	24 В пост. (+20/-15%, коэффициент пульсации в пределах 5%)
Номинальный входной ток	ок. 4 мА
Снижение входных сигналов	100 % (можно включать сразу все входы)
Пусковой ток	макс. 200 А в течение 1 мс (при 132 В пер.)
Напряжение/ток включения	19 В пост. или выше/3 мА или выше
Напряжение/ток отключения	11 В пост. или ниже/1,7 мА или ниже
Входное сопротивление	ок. 5,6 кΩ
Время отклика	Выкл. → Вкл.
	Вкл. → Выкл.
	1, 5, 10, 20, 70 мс (настройка параметра ЦП, исходная настройка: 10 мс)*
Электрическая прочность	560 В перем. RMS/3 цикла (высота над уровнем моря: 2000 м)
Сопротивление изоляции	10 МΩ или выше (по тестеру сопротивления изоляции)
Помехоустойчивость	При использовании генератора помех: напряжение помехи - 500 В р-р, ширина импульса помехи - 1 μс, частота помехи – от 25 до 60 Гц
	Первая переходная помеха по IEC61000-4-4: 1 кВ
Группы входов	1 группа с 16 входами (общая клемма: клемма 18)
Индикация работы	по 1 светодиоду на каждый вход
Внешние соединения	18-точечная клеммная колодка (с винтами М3 х 6)
Размеры провода	от 0,3 до 0,75 мм ² , сердечник: макс. диаметр 2,8 мм
Внутреннее потребление тока (5 В пост.)	50 мА (все точки входа Вкл.)
Вес	0,16 кг

* Время отклика для Выкл - Время отклика для Выкл -> Время отклика для Выкл -> Вкл и Вкл - Время отклика для Выкл -> Вкл и Вкл -> Время отклика для Выкл -> Вкл и Вкл -> Выкл нельзя устанавливать отдельно.



Принцип работы модуля входов с общим минусом

Согласно представленной выше принципиальной схеме для модуля QX0 при замыкании контакта кнопки ток будет протекать следующим образом:

- С клеммы +24 В внешнего источника питания, через кнопку и дальше на клемму 1 модуля входов.
- Клемма 1 соединяется с отрицательным полюсом (клеммы 18) внешнего источника питания через резистор и светодиод оптронной пары. Поэтому ток протекает через светодиод.
- Когда ток протекает через светодиод, последний загорается, включая тем самым фоторезистор.
- Назначение оптронной пары – изоляция входной цепи 24 В на стороне производственного оборудования от чувствительной цепи 5 В процессора ПЛК. Она также обеспечивает помехоустойчивость в отношении входа.
- При включении фоторезистора передается сигнал в таблицу входного изображения для сохранения информации о включении входа X0. В этом случае загорается светодиод на лицевой стороне модуля входов, показывая состояние сигнала.

Пример модуля входов с общим плюсом

Параметр		Технические данные
Тип модуля		QX40
Количество точек входа		16
Способ изоляции		Оптронная пара
Номинальное входное напряжение		24 В пост. (+20/-15%, коэффициент пульсации в пределах 5%)
Номинальный входной ток		ок. 4 мА
Снижение входных сигналов		100 % (можно включать сразу все входы)
Пусковой ток		макс. 200 А в течение 1 мс (при 132 В пер.)
Напряжение/ток включения		19 В пост. или выше/3 мА или выше
Напряжение/ток отключения		11 В пост. или ниже/1,7 мА или ниже
Входное сопротивление		Аррrox. 5.6 кΩ
Время отклика	ВЫКЛ. → ВКЛ.	1, 5, 10, 20, 70 мс (настройка параметра ЦП, исходная настройка: 10 мс)*
	ВКЛ. → ВЫКЛ.	
Электрическая прочность		560 В перем. RMS/3 цикла (высота над уровнем моря: 2000 м)
Сопротивление изоляции		10 МΩ или выше (по тестеру сопротивления изоляции)
Помехоустойчивость		При использовании генератора помех: напряжение помехи - 500 В р-р, ширина импульса помехи - 1 мс, частота помехи - от 25 до 60 Гц
		Первая переходная помеха по IEC61000-4-4: 1 кВ
Группы входов		1 группа с 16 входами (общая клемма: клемма 17)
Индикация работы		по 1 светодиоду на каждый вход
Внешние соединения		18-точечная клеммная колодка (с винтами М3 х 6)
Размеры провода		от 0,3 до 0,75 мм ² , сердечник: макс. диаметр 2,8 мм
Внутреннее потребление тока (5 В пост.)		50 мА (все точки входа ВКЛ.)
Вес		0,16 кг

* Время отклика для ВЫКЛ - Время отклика для ВЫКЛ -> Время отклика для ВЫКЛ -> ВКЛ и ВКЛ - Время отклика для ВЫКЛ -> ВКЛ и ВКЛ -> Время отклика для ВЫКЛ -> ВКЛ и ВКЛ -> ВЫКЛ нельзя устанавливать отдельно.

Вид	Принципиальная схема	Клемма	Сигнал
		1	X00
		2	X01
		3	X02
		4	X03
		5	X04
		6	X05
		7	X06
		8	X07
		9	X08
		10	X09
		11	X0A
		12	X0B
		13	X0C
		14	X0D
		15	X0E
		16	X0F
		17	COM
		18	Резерв.

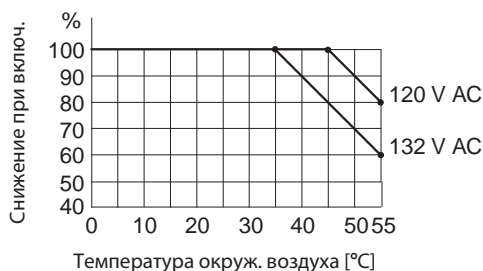
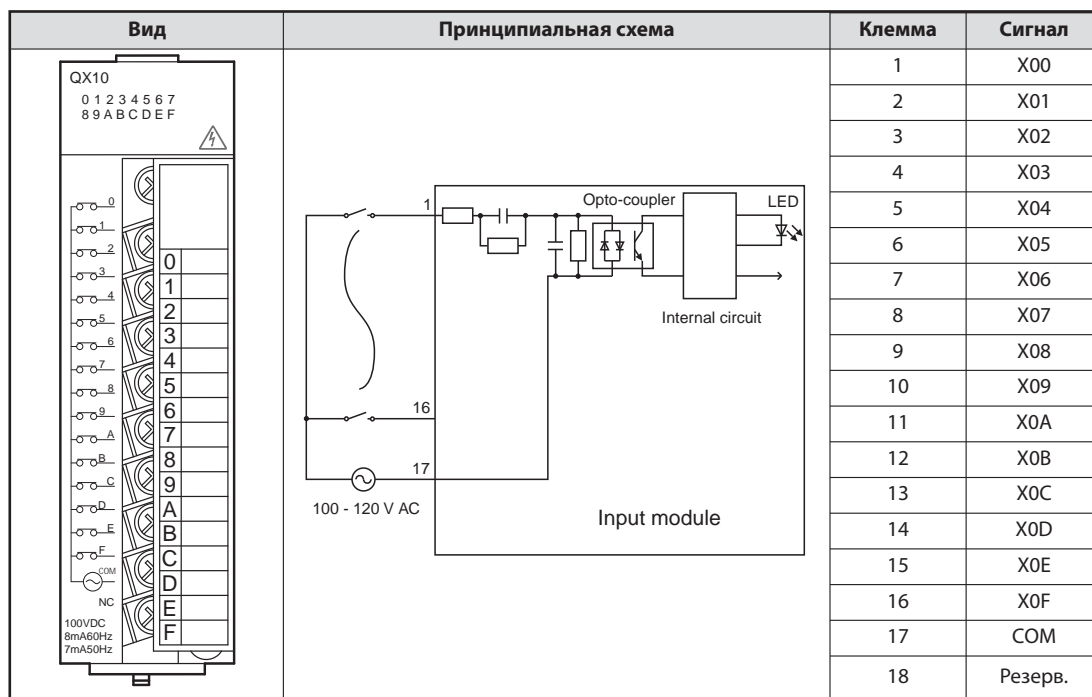
Принцип работы модуля входов с общим плюсом

Согласно представленной выше схеме при замыкании контакта кнопки с клеммой 1 ток будет протекать следующим образом:

- С клеммы +24 В внешнего источника питания на общую клемму (клемма 17).
- Через светодиод оптронной пары и сетевую схему входного резистора на клемму 1 (клемма входа X0) модуля входов.
- Когда ток протекает через светодиод, последний загорается, включая тем самым фоторезистор.
- При включении фоторезистора передается сигнал в таблицу входного изображения для сохранения информации о включении входа X0. В этом случае загорается соответствующий светодиод на лицевой стороне модуля входов, показывая состояние сигнала.
- Затем ток протекает через кнопку назад к отрицательному полюсу внешнего источника питания.

Пример модуля входов сигналов переменного тока

Параметр		Технические данные
Тип модуля		QX10
Количество точек входа		16
Способ изоляции		Оптронная пара
Номинальное входное напряжение		100 - 120 В пер. (+10%/-15%), 50/60 Гц (± 3 Гц) (коэффициент искажений в пределах 5 %)
Номинальный входной ток		ок. 8 мА при 100 В перем., 60 Гц; ок. 7 мА при 100 В перем., 50 Гц
Снижение входных сигналов		см. ниже график снижения
Пусковой ток		макс. 200 А в течение 1 мс (при 132 В пер.)
Напряжение/ток включения		80 В перем. или выше/5 мА или выше (50 Гц, 60 Гц)
Напряжение/ток отключения		30 В пост. или ниже/1 мА или ниже (50 Гц, 60 Гц)
Входное сопротивление		ок. 15 к Ω при 60 Гц, ок. 18 к Ω при 50 Гц
Время отклика	ВЫКЛ. \rightarrow ВКЛ.	15 мс или меньше (100 В перем., 50 Гц, 60 Гц)
	ВКЛ. \rightarrow ВЫКЛ.	20 мс или меньше (100 В перем., 50 Гц, 60 Гц)
Электрическая прочность		1780 В перем. RMS/3 цикла (высота над уровнем моря: 2000 м)
Сопротивление изоляции		10 М Ω или выше (по тестеру сопротивления изоляции)
Помехоустойчивость		При использовании генератора помех: напряжение помехи - 1500 В р-р, ширина импульса помехи - 1 μ с, частота помехи - от 25 до 60 Гц
		Первая переходная помеха по IEC61000-4-4: 1 кВ
Группы входов		1 группа с 16 входами (общая клемма: клемма 17)
Индикация работы		по 1 светодиоду на каждый вход
Внешние соединения		18-точечная клеммная колодка (с винтами М3 х 6)
Размеры провода		от 0,3 до 0,75 мм ² , сердечник: макс. диаметр 2,8 мм
Внутреннее потребление тока (5 В пост.)		50 мА
Вес		0,17 кг

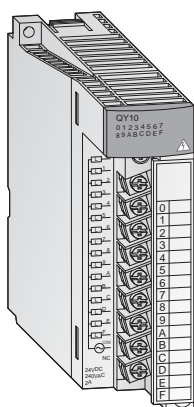


Для модуля QX10 количество входов, которые можно включить одновременно, зависит от температуры окружающего воздуха.

Для модулей с входными сигналами переменного тока для ПЛК рекомендуется использовать такое же напряжение питания, как и для входов (например, 100–120 В перем.). Это сведет к минимуму возможность подачи на входы несоответствующего напряжения.

3.5.2 Модули дискретных выходов

Модули выходов System Q имеет различные переключающие элементы для использования во многих задачах управления.



Тип выхода	Номинальное выходное напряжение	Модуль выхода			
		8	16	32	64
Релейный	24 В пост./ 240 В перем.	QY18A	QY10		
Тиристорный	100–240 В пер.		QY22		
Транзисторный	5/12 В пост.		QY70	QY71	
	12/24 В пост.		QY40P QY50 QY80	QY41P QY81	QY42P
	5–24 В пост.	QY68A			

Модули с 8 и 16 точками подключения снабжены съёмными колодками с винтовыми клеммами. Модули с 32 и 64 точками подключаются через разъём.

Типы выходов

Предусмотрено четыре типа модулей дискретных выходов System Q:

- Релейные
- Тиристорные
- Транзисторные (переключающие на плюс)
- Транзисторные (переключающие на минус)

Тип	Преимущества	Недостатки
Релейный	<ul style="list-style-type: none"> ● Один модуль может переключать разные напряжения ● Возможна работа без напряжения ● Способность переключения при высоком токе 	<ul style="list-style-type: none"> ● Медленный (макс. 1 Гц) ● Ограниченная надёжность (электромеханический) ● Обгорание контактов ● Шум (электрический)
Тиристорный	<ul style="list-style-type: none"> ● Высокая надёжность ● Повышенная скорость переключения ● Подходит для использования в условиях интенсивного переключения 	<ul style="list-style-type: none"> ● Работает только от перем. тока ● Ограничение по току: 0,6 А на точку ● Для ВКЛ./ВЫКЛ. при перем. токе 50 Гц требуется 10 мс
Транзисторный	<ul style="list-style-type: none"> ● Очень высокая надёжность ● Очень высокая скорость переключения ● Оптimalен для использования в условиях интенсивного переключения 	<ul style="list-style-type: none"> ● Работает только от низкого пост. напряжения ● Ограничение по току: 0,1 А на точку

Модули релейных выходов

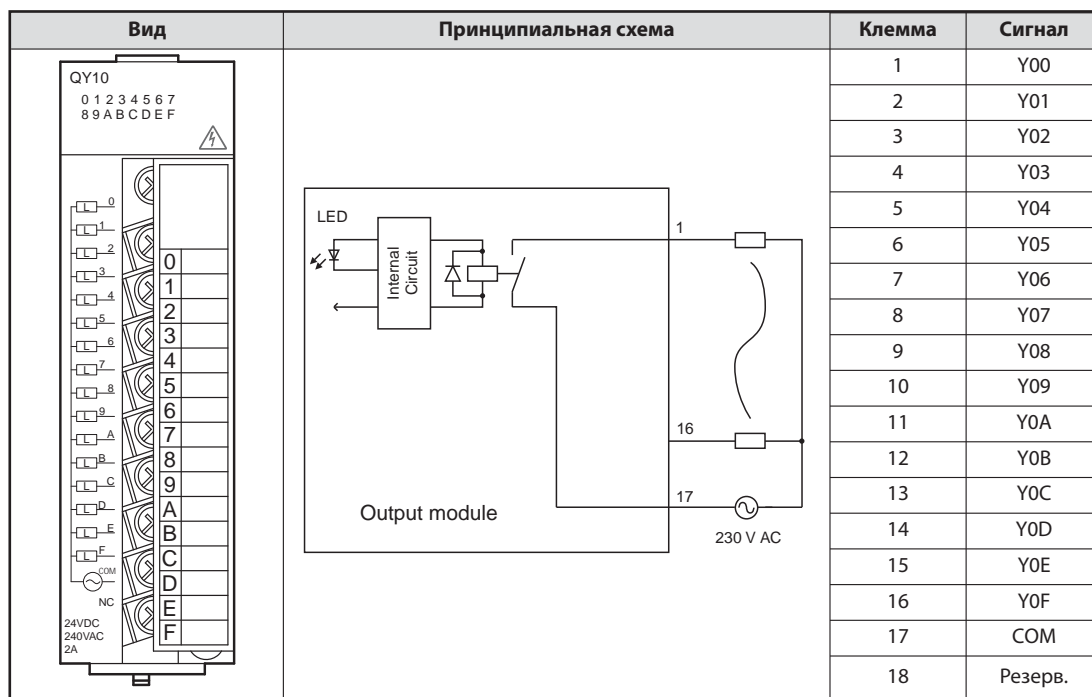
Гальваническая развязка от внутренних и внешних цепей обеспечивается катушками и контактами выходных реле.

Существуют модули как с групповыми выходами с изолированными общими клеммами, так и и одинарные модули с изолированным выходом без напряжения (QY18A).

Как и в модулях выходов других типов, управление выходными контактами осуществляется внутренней программой ЦП. В конце программы ПЛК обновляет состояние выходов в фиксируемой области памяти вывода, загорается светодиод и выходной контакт замыкается. Время срабатывания реле составляет примерно 10 мс.

Пример модуля релейного выхода

Параметр	Технические данные	
Тип модуля	QY10	
Количество точек выхода	16	
Способ изоляции	Релейный	
Номинальное коммутируемое напряжение/ток	24 В пост., 2 А (активная нагрузка) на каждый выход 240 В перем., 2 А ($\cos \varphi = 1$) на каждый выход; макс. 8 А на группу	
Мин. коммутируемая нагрузка	5 В пост., 1 мА	
Мин. коммутируемое напряжение	125 В пост./264 В перем.	
Время отклика	ВЫКЛ. → ВКЛ.	10 мс или меньше
	ВКЛ. → ВЫКЛ.	12 мс или меньше
Долговечность	Механическая	не менее 20 млн. переключений
	Электрическая	не менее 100 тыс. переключений при номин. коммутируемом напряжении/токовой нагрузке
		не менее 100 тыс. переключений при 200 В перем., 1,5 А; 240 В перем. 1 А ($\cos \varphi = 0,7$)
		не менее 300 тыс. переключений при 200 В перем., 0,4 А; 240 В перем. 0,3 А ($\cos \varphi = 0,7$)
		не менее 100 тыс. переключений при 200 В перем., 1 А; 240 В перем. 0,5 А ($\cos \varphi = 0,35$)
не менее 300 тыс. переключений при 200 В перем., 0,3 А; 240 В перем. 0,15 А ($\cos \varphi = 0,35$)		
не менее 100 тыс. переключений при 24 В пост. 1 А; 100 В пост. 0,1 А ($L/R = 0,7$ мс)		
не менее 300 тыс. переключений при 24 В пост. 0,3 А; 100 В пост. 0,03 А ($L/R = 0,7$ мс)		
Макс. частота переключения	3600 переключений в час	
Защита от перенапряжений	—	
Предохранитель	—	
Электрическая прочность	2830 В перем. RMS/3 цикла (высота над уровнем моря: 2000 м)	
Сопротивление изоляции	10 МΩ или выше (по тестеру сопротивления изоляции)	
Помехоустойчивость	При использовании генератора помех: напряжение помехи - 1500 В р-р, ширина импульса помехи - 1 мкс, частота помехи - от 25 до 60 Гц	
	Первая переходная помеха по IEC61000-4-4: 1 кВ	
Группы выходов	1 группа с 16 выходами (общая клемма: клемма 17)	
Индикация работы	по 1 светодиоду на каждый выход	
Внешние соединения	18-точечная клеммная колодка (с винтами М3 х 6)	
Размеры провода	от 0,3 до 0,75 мм ² , сердечник: макс. диаметр 2,8 мм	
Внутреннее потребление тока (5 В пост.)	430 мА	
Вес	0,22 кг	



Тиристорные модули выходов

Модули дискретных выходов могут переключать напряжение от 100 до 240 В перем. Как и для всех остальных типов модулей выходов, развязка между физическим выходом обеспечивается посредством оптронной пары. Время срабатывания тиристорного модуля безусловно меньше, чем релейного, – 1 мс на включение и 10 мс на последующее отключение.

Нагрузка тиристорного выхода ограничивается 0,6 А, поэтому следует учитывать это при конфигурировании системы, чтобы исключить перегрузку выходной цепи.

Поскольку ток утечки в выходной цепи тиристора превышает ток утечки цепи реле, следует соблюдать осторожность, так как этот ток может вызвать включение индикаторов и некоторых миниреле. Фактически это одна из самых распространенных причин поражения током при работе с электрошкафами, управляемыми ПЛК.

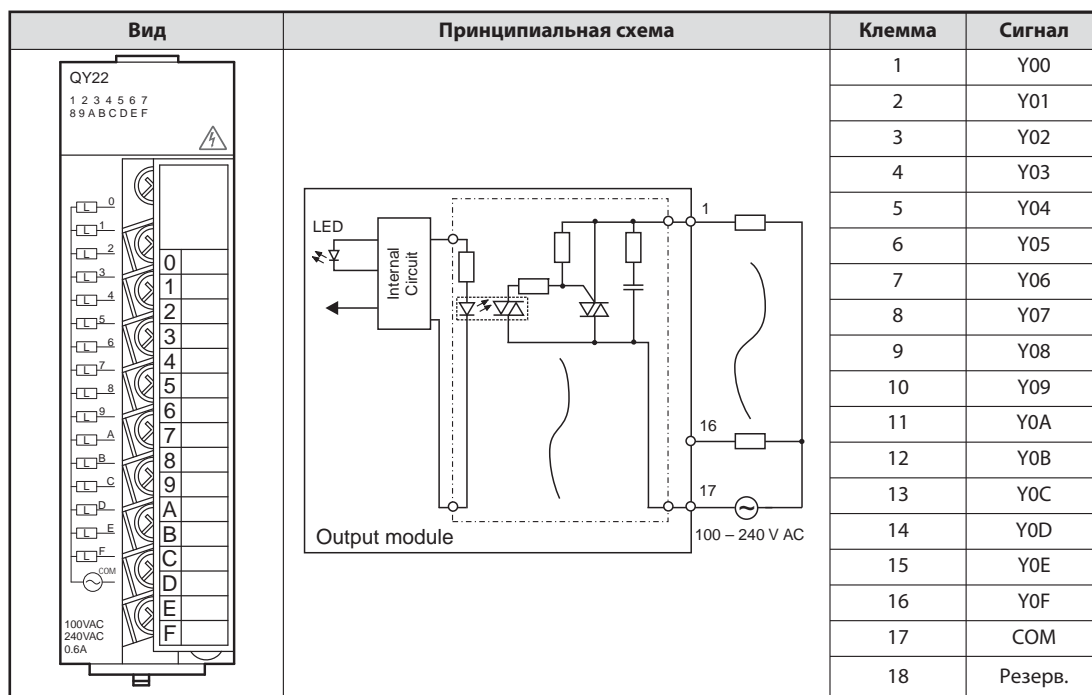


ОПАСНО

При работе с выходными цепями под напряжением, управляемыми тиристорными устройствами, следует соблюдать особую осторожность, даже если выходы однозначно отключены!

Пример для модуля тиристорных выходов

Параметр		Технические данные
Тип модуля		QY22
Количество точек выхода		16
Способ изоляции		Оптронная пара
Номинальное коммутируемое напряжение/ток		100-240 В перем. (+20/-15 %), 0,6 А на каждый выход, 4,8 А на каждый модуль
Мин. коммутируемая нагрузка		24 В перем., 100 мА; 100 В перем., 25 мА, 240 В перем., 25 мА
Макс. пусковой ток		20 А
Ток утечки при отключении		3 мА или меньше при 120 В перем., 60 Гц 1,5 мА или меньше при 240 В перем., 60 Гц
Макс. падение напряжения при включении		1,5 В
Время отклика	ВЫКЛ. → ВКЛ.	0,5 x период + макс. 1 мс
	ВКЛ. → ВЫКЛ.	0,5 x период + макс. 1 мс
Защита от перенапряжений		Устройство защиты от перенапряжений RC-типа
Предохранитель		
Электрическая прочность		2830 В перем. RMS/3 цикла (высота над уровнем моря: 2000 м)
Сопротивление изоляции		10 МΩ или выше (по тестеру сопротивления изоляции)
Помехоустойчивость		При использовании генератора помех: напряжение помехи - 1500 В р-р, ширина импульса помехи - 1μс, частота помехи - от 25 до 60 Гц Первая переходная помеха по IEC61000-4-4: 1 кВ
Группы выходов		1 группа с 16 выходами (общая клемма: клемма 17)
Индикация работы		по 1 светодиоду на каждый выход
Внешние соединения		18-точечная клеммная колодка (с винтами М3 х 6)
Размеры провода		от 0,3 до 0,75 мм ² , сердечник: макс. диаметр 2,8 мм
Внутреннее потребление тока (5 В пост.)		250 мА (когда все выходы включены)
Вес		0,40 кг



Модули транзисторных выходов

Как и для всех остальных типов модулей выходов, развязка между физическими выходами модулей транзисторных выходов обеспечивается посредством оптронной пары.

Время отклика транзистора составляет всего 1 мс при напряжении 24 В пост. и токе 200 мА. Допустимая нагрузка по току для каждого входа указывается в соответствующем описании аппаратуры.

Существует два тип модулей транзисторных выходов System Q – переключающие на минус и на плюс.

Пример для модуля выходов, переключающего на плюс

Параметр		Технические данные
Тип модуля		QY80
Количество точек выхода		16
Способ изоляции		Оптронная пара
Номинальное коммутируемое напряжение		от 12 до 24 В пост. (+20/-15%)
Диапазон коммутируемого напряжения		от 0,2 до 28,8 В пост.
Макс. ток нагрузки		0,5 А на каждый выход, 4 А на каждую группу
Макс. пусковой ток		4 А в течение 10 мс
Ток утечки при отключении		0,1 мА или меньше
Макс. падение напряжения при включении		0,2 В пост. при 0,5 А (тип.), макс. 0,3 В при 0,5 А
Время отклика	ВЫКЛ. → ВКЛ.	1 мс или меньше
	ВКЛ. → ВЫКЛ.	1 мс или меньше (номинальная нагрузка, активная нагрузка)
Защита от перенапряжений		Z-диод
Предохранитель		6,7 А (одноразовый)
Индикация перегорания предохранителя		Светодиод сигнализирует о перегорании предохранителя, на ЦП ПЛК поступает соответствующий сигнал
Внешнее питание	Напряжение	от 12 до 24 В пост. (+20/-15%, коэффициент пульсации в пределах 5%)
	Ток	20 мА (при 24 В пост., когда все выходы включены)
Электрическая прочность		560 В перем. RMS/3 цикла (высота над уровнем моря: 2000 м)
Сопротивление изоляции		10 МΩ или выше (по тестеру сопротивления изоляции)
Помехоустойчивость		При использовании генератора помех: напряжение помехи - 500 В р-р, ширина импульса помехи - 1 мс, частота помехи - от 25 до 60 Гц
		Первая переходная помеха по IEC61000-4-4: 1 кВ
Группы выходов		1 группа с 16 выходами (общая клемма: клемма 17)
Индикация работы		по 1 светодиоду на каждый выход
Внешние соединения		18-точечная клеммная колодка (с винтами М3 х 6)
Размеры провода		от 0,3 до 0,75 мм ² , сердечник: макс. диаметр 2,8 мм
Внутреннее потребление тока (5 В пост.)		80 мА (когда все выходы включены)
Вес		0,17 кг

Вид	Принципиальная схема	Клемма	Сигнал
<p>QY80 0 1 2 3 4 5 6 7 8 9 A B C D E F FUSE □</p> <p>0 1 2 3 4 5 6 7 8 9 A B C D E F COM</p> <p>12VDC 24VDC 0.5A</p>	<p>LED Internal Circuit 1 16 17 18 12 - 24 V DC</p>	1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	COM
		18	0 B

Пример для модуля выходов, переключающего на минус

Параметр		Технические данные
Тип модуля		QY40P
Количество точек выхода		16
Способ изоляции		Оптронная пара
Номинальное коммутируемое напряжение		от 12 до 24 В пост. (+20/-15%)
Диапазон коммутируемого напряжения		от 10,2 до 28,8 В пост.
Макс. ток нагрузки		0,1 А на каждый выход, 1,6 А на каждую группу
Макс. пусковой ток		0,7 А в течение 10 мс
Ток утечки при отключении		0,1 мА или меньше
Макс. падение напряжения при включении		0,1 В пост. при 0,1 А (тип.), макс. 0,2 В при 0,1 А
Время отклика	ВЫКЛ. → ВКЛ.	1 мс или меньше
	ВКЛ. → ВЫКЛ.	1 мс или меньше (номинальная нагрузка, активная нагрузка)
Защита от перенапряжений		Z-диод
Предохранитель		—
Внешнее питание	Напряжение	от 12 до 24 В пост. (+20/-15%, коэффициент пульсации в пределах 5 %)
	Ток	10 мА (при 24 В пост., когда все выходы включены)
Электрическая прочность		560 В перем. RMS/3 цикла (высота над уровнем моря: 2000 м)
Сопротивление изоляции		10 МΩ или выше (по тестеру сопротивления изоляции)
Помехоустойчивость		При использовании генератора помех: напряжение помехи - 500 В р-р, ширина импульса помехи - 1 мс, частота помехи - от 25 до 60 Гц
		Первая переходная помеха по IEC61000-4-4: 1 кВ
Группы выходов		1 группа с 16 выходами (общая клемма: клемма 18)
Индикация работы		по 1 светодиоду на каждый выход
Внешние соединения		18-точечная клеммная колодка (с винтами М3 х 6)
Размеры провода		от 0,3 до 0,75 мм ² , сердечник: макс. диаметр 2,8 мм
Внутреннее потребление тока (5 В пост.)		65 мА (когда все выходы включены)
Вес		0,16 кг

Вид	Принципиальная схема	Клемма	Сигнал
		1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	12/24 В пост.
		18	COM

3.6 Специальные функциональные модули

3.6.1 Аналоговые модули

При автоматизации процессов часто требуется осуществлять сбор или регулирование аналоговых параметров, таких как температура, давление и уровень заполнения. Ввод и вывод аналоговых сигналов производится с помощью дополнительных аналоговых модулей.

Существует два основных типа аналоговых модулей:

- Модули аналоговых входов
- Модули аналоговых выходов

Модули аналоговых входов обеспечивают сбор данных по току, напряжению и температуре. Модули аналоговых выходов передают сигналы тока или напряжения на модули выходов.

Критерии выбора аналоговых модулей

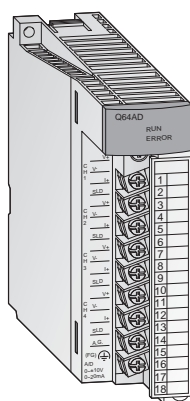
Для аппаратуры System Q предусмотрена широкая номенклатура аналоговых модулей, которые выбираются в зависимости от задачи автоматизации. Основными критериями при выборе являются следующие:

- Разрешение
Разрешением является наименьшее значение физ величины, которое может регистрировать и выводить аналоговый модуль.
Разрешение модулей аналоговых входов определяется как изменение напряжения, тока или температуры на входе, при котором значение дискретного выхода изменяется на 1.
Разрешение модулей аналоговых выходов определяется как изменение напряжения или тока на выходе, при котором значение дискретного входа изменяется на 1.
Разрешение ограничивается внутренней конструкцией аналоговых модулей и зависит от числа разрядов для хранения цифрового значения. Например, когда 12-разрядным модулем АЦП регистрируется напряжение 10 В, диапазон напряжения делится на 4096 диапазонов ($2^{12} = 4096$, см. раздел 4.3). В результате разрешение составляет $10 \text{ В}/4096 = 2,5 \text{ мВ}$.
- Количество аналоговых входов или выходов
Входы и выходы аналоговых модулей называют также каналами. Можно выбрать модули аналоговых входов с 2, 4 или 8 каналами, в зависимости от потребностей.

Модули аналоговых входов

Модули аналоговых входов преобразовывают измеренную аналоговую величину (например, 10 В) в цифровое значение (например, 4000), которое может обрабатываться ПЛК. Данный процесс называется аналого-цифровым преобразованием или АЦП.

Аналоговые модули System Q могут напрямую регистрировать температуру, но другие физические параметры, такие как давление и расход, сначала нужно преобразовать в значение тока или напряжения и только потом преобразовать в цифровые значения для обработки ПЛК. Такое преобразование осуществляется датчиками, которые обеспечивают вывод сигнала в стандартизованных диапазонах (например, 0–10 В или 4–20 мА). Преимущество сигнала тока состоит в том, что значение не искажается влиянием длины проводов или сопротивлением контактов.



Модули аналоговых входов System Q отличаются сочетанием высокого разрешения (0,333 мВ/1,33 мкА) и высокой скорости преобразования (80 мкс на канал).

Все модули снабжены съёмными колодками с винтовыми клеммами.

Аналоговый вход	Диапазон аналогового входа	Выбор входного диапазона	Входных каналов	Модуль
Напряжение	от -10 до +10 В	от 1 до 5 В от 0 до 5 В от 0 до 10 В от -10 до +10 В	8	Q68ADV
Ток	от 0 до 20 мА	от 0 до 20 мА от 4 до 20 мА	8	Q68ADI
Напряжение или ток (выбирается для каждого канала)	от -10 до +10 В от 0 до 20 мА	как для 68ADV и Q68ADI	4	Q64AD

Аналоговые модули входов для измерения температуры

Температуру можно регистрировать двумя типами датчиков: резисторными термометрами Pt100 и термопарами.

- Резисторные термометры Pt100

Данные устройства измеряют сопротивление платинового элемента, изменяющегося при изменении температуры. При температуре 0°C сопротивление элемента составляет 100 Ω (поэтому он называется Pt100). Резисторные термометры подключаются по трехпроводной схеме, исключающей влияние соединительных проводов на результат измерения.

Максимальный диапазон измерений резисторных термометров Pt100 составляет от -200 °C до +600 °C, но на практике он также зависит от возможностей используемого модуля измерения температуры.

В резисторных термометрах также применяется никель (Ni100). В этом случае диапазон измерений меньше (от -60 °C до 180 °C).

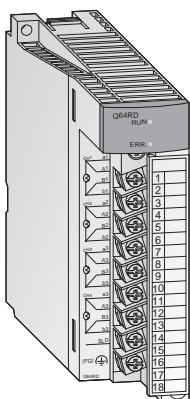
- Термопары

Преимущество данных устройств для измерения температуры заключается в том, что напряжение создается при нагреве элемента, состоящего из двух разных металлов. То есть в данном методе температура измеряется с помощью сигнала напряжения.

Существуют различные типы термопар. Они отличаются по термоэлектродвижущей силе (термоэдс) и диапазону измерения температуры. Используемые сочетания материалов стандартизованы и определяются кодом типа. Наиболее распространенными типами являются J и K. В термопарах типа J используется сочетание железа (Fe) с медно-никелевым сплавом (CuNi), в термопарах типа K – сочетание NiCr и Ni. Помимо конструктивного состава термопары отличаются по диапазону измерения температуры.

Термопары служат для измерения температуры в диапазоне от -200 °C до +1200 °C.

Специальные функции



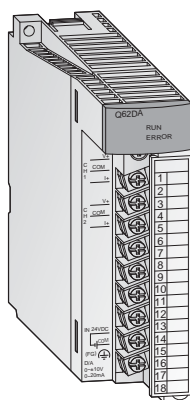
- Один модуль служит для измерения температуры по 4-м каналам;
- Разрыв связи с датчиком температуры может быть обнаружен в каждом канале;
- Возможность выбора: обработки замеров / обработки усреднения по времени / обработки усреднения по отсчёту;
- Коррекция ошибок за счет настройки смещения/усиления;
- Выходной сигнал тревоги при превышении предельного значения;
- Гальваническая развязка между процессом и системой управления посредством оптопары – стандарт для данного оборудования. Дополнительная гальваническая развязка потенциала между каналами для модулей Q64TDV-GH и Q64RD-G.

Датчик температуры	Диапазон измеряемых температур	Макс. разрешение	Модуль
Резисторный термометр (Pt100, JPt100)	Pt100: от -200 до 850 °C, JPt 100: от -180 до 600 °C	0,025 °C	Q64RD
Резисторный термометр (Pt100, JPt100, Ni100)	Pt100: от -200 до 850 °C, JPt 100: от -180 до 600 °C, Ni100: от -60 до 180 °C	0,025 °C	Q64RD-G
Термопары типа K, E, J, T, B, R, S или N	Зависит от используемой термопары	B, R, S, N: 0,3 °C; K, E, J, T: 0,1 °C	Q64TD
		B: 0,7 °C; R, S: 0,8 °C; K, T: 0,3 °C; E, T: 0,2 °C; J: 0,1 °C; N: 0,4 °C; Напряжение: 4 мВ	Q64TDV-GH

Модули аналоговых выходов

Модули аналоговых выходов преобразовывают цифровые значения с ЦП ПЛК в аналоговый сигнал напряжения или тока, используемый для управления внешним устройством (аналого-цифровое преобразование или АЦП).

В аппаратуре System Q используются аналоговые выходные сигналы стандартного промышленного диапазона: 0–10 В и 4–20 мА.



Разрешение 0,333 мВ и, соответственно, 0,83 мкА, а также крайне малое время преобразования 80 мкс на выходной канал – это всего лишь две из множества особенностей данных модулей. Развязка между процессом и системой управления посредством оптоэлектронных пар – стандарт для данного оборудования.

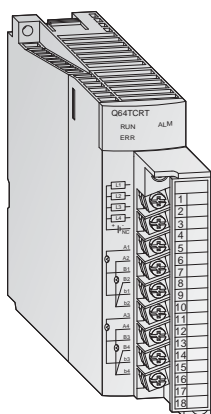
Все модули снабжены съёмными колодками с винтовыми клеммами.

Аналоговый выход	Диапазон аналогового выхода	Выбор выходного диапазона	Выходные каналы		
			2	4	8
Напряжение или ток (выбирается для каждого канала)	от -10 до +10 В от 0 до 20 мА	от 1 до 5 В от -10 до +10 В от 0 до 20 мА от 4 до 20 мА	Q62DA	Q64DA	
Напряжение	от -10 до +10 В	от -10 до +10 В			Q68DAV
Ток	от 0 до 20 мА	от 0 до 20 мА от 4 до 20 мА			Q68DAI

3.6.2 Модули регулирования температуры с ПИД-регулятором

Данные модули позволяют использовать ПИД-регуляторы для регулирования температуры, не загружая ЦП ПЛК этими задачами.

Специальные функции

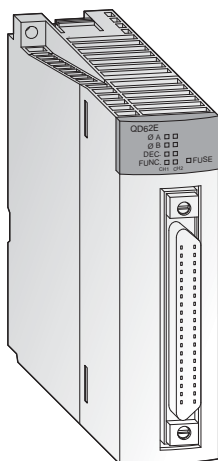


- Четыре канала входа температуры и четыре ПИД-регулятора на модуль;
- Регистрация температуры резисторными термометрами Pt100 (Q64TCRT и Q64TCRTBW) или термопарами (Q64TCTT и Q64TCTTBW);
- Модули 64TCRTBW и Q64TCTTBW обнаруживают разрыв связи с нагревателем;
- Функция самонастройки для четырех ПИД-регуляторов;
- Транзисторный выход с последовательностью импульсов для управления соленоидом в управляющем контуре.

3.6.3 Модули высокоскоростных счётчиков

Модули QD62E, QD62 и QD62D служат для обнаружения сигналов, частота которых превышает возможностей обычных модулей ввода.

Специальные функции

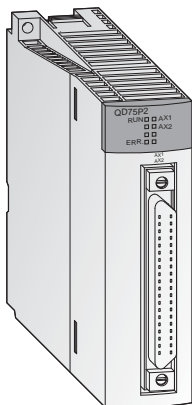


- Максимальная частота счёта составляет 500 кГц;
- Вход для инкрементного энкодера с автоматическим определением прямого и обратного направления вращения;
- Предварительная настройка и выбор функций счёта через внешние дискретные входы;
- 32-разрядный диапазон счёта (от -2 147 483 648 до +2 147 483 647)
- Может использоваться как суммирующий, вычитающий и кольцевой счётчик;
- Все модули с двумя входами счётчика;
- Два дискретных выхода, устанавливаемые в зависимости от значения счётчика на каждом входе.

Все модули подключаются через 40-контактный разъём.

3.6.4 Модули позиционирования

В сочетании с шаговыми двигателями и сервоусилителями модули QD75P1, QD75P2 и QD75P4 используются для регулирования частоты вращения или положения.

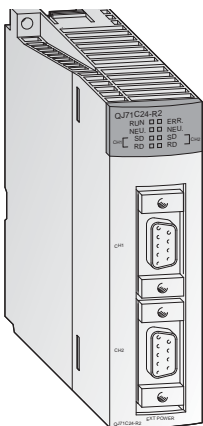


Специальные функции

- Управление четырьмя осями с линейной интерполяцией (QD75P4) или двумя осями с круговой интерполяцией (QD75P2 и QD75P4);
- Хранение до 600 наборов данных позиционирования во флэш-ПЗУ;
- Перемещение может измеряться в импульсах, мкм, дюймах и градусах;
- Конфигурирование и предварительная настройка данных позиционирования выполняется с помощью программы ПЛК или программы GX Configurator QP для Microsoft Windows.

3.6.5 Модули последовательной связи

Модули QJ71C24 и QJ71C24-R2 обеспечивают связь с периферийными устройствами через стандартный последовательный интерфейс.



Специальные функции

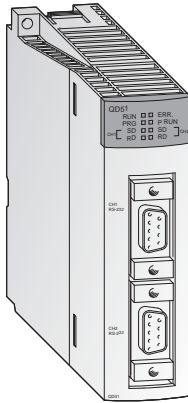
- Два интерфейса RS232C (QJ71C24-R2) или один интерфейс RS422/485 и один RS232C (QJ71C24);
- Скорость передачи до 115200 бит/с;
- Обеспечивают ПК, подключенным к ПЛК, доступ к полному набору данных ЦП System Q;
- Возможности для подключения принтера;
- Встроенная память флэш-ПЗУ для регистрации данных о качестве, производительности или тревожных событиях для последующей передачи по запросу;
- Поддержка обмена данными в формате ASCII, возможность задания пользовательского фрейма;
- Поддержка программирования и контроля ПЛК через линию последовательной связи.

3.6.6 Программируемые на BASIC интерфейсные модули

Модули QD51S-R24 и QD51 работают под управлением собственной программы (на языке BASIC), независимо от ЦП System Q. Это позволяет обрабатывать данные и обеспечивать связь с периферийными устройствами, не загружая ЦП ПЛК этими задачами.

Специальные функции

- Два интерфейса RS232 (QD51) или один интерфейс RS422/485 и один RS232 (QD51S-R24);
- Скорость передачи до 38400 бит/с;
- Поддержка доступа к устройствам в ЦП System Q и к буферной памяти специальных функциональных модулей;
- Поддержка удаленных команд RUN/STOP через линию последовательной связи.



3.7 Сети и сетевые модули

3.7.1 Сети разных уровней

Для сложных или широко разветвленных приложений, внедрения удаленных входов и выходов и визуализации процессов очень важное значение имеет связь между ПЛК, компьютерами управления производством, терминалами операторов и прочими устройствами.

Mitsubishi Electric предлагает оптимальные решения на основе трехуровневой сети:

- Производственный уровень
- Уровень управления
- Командный уровень

Производственный уровень

Представляет собой производственную сеть, связывающую управляющие устройства, такие как ПЛК, с удаленными входами и выходами, инверторами и терминалами операторов на самом низком сетевом уровне на производственных участках.

Раньше управляющие устройства соединялись с датчиками и приводами по проводам по принципу точка-точка. Производственная сеть позволяет соединять множество датчиков и привода одним сетевым кабелем, сокращая количество проводов и объем работ по их подключению. При подключении интеллектуального оборудования, такого как система идентификации, считыватель штрих-кодов, инвертор и дисплей, позволяет управлять производственными данными на концах сети посредством передачи различных данных в дополнение к данным типа ВКЛ/ВЫКЛ и обеспечивает повышение эффективности обслуживания за счет централизованного контроля рабочего состояния оборудования.

Применение ПЛК серии System Q еще больше повышает скорость и производительность, при этом упрощая управление.

Уровень управления

Сеть управления, связывающая такие управляющие устройства, как ПЛК и системы ЧПУ, находится на среднем сетевом уровне на производственных объектах. Сеть управления предназначена для передачи данных, непосредственно относящихся к операциям и движениям машины и оборудования, между управляющими устройствами, поэтому она должна обладать высокими реальновременными характеристиками. Сеть управления MELSECNET(10/H) высоко ценится за превосходные реальновременные характеристики, простые настройки сети, высокую резервированную надежность, обеспечиваемую двойным контуром.

Командный уровень

Представляет собой высший сетевой уровень в информационной сети производства. Информационная сеть предназначена для передачи информации по управлению производством, рабочего состояния объекта и прочих данных между ПЛК или контроллером объекта и компьютером управления производством и предполагает использование наиболее распространенной сети Ethernet. В сети Ethernet могут работать как персональные компьютеры на базе Windows и UNIX, так и различное промышленное оборудование для автоматизации производства. В аппаратуре System Q предусмотрены функции, обеспечивающие оптимальное использование возможностей Ethernet.

В дополнение к вышеуказанным уровням сети можно разделить на следующие типы:

- Открытые сети
- и
- Сети MELSEC.

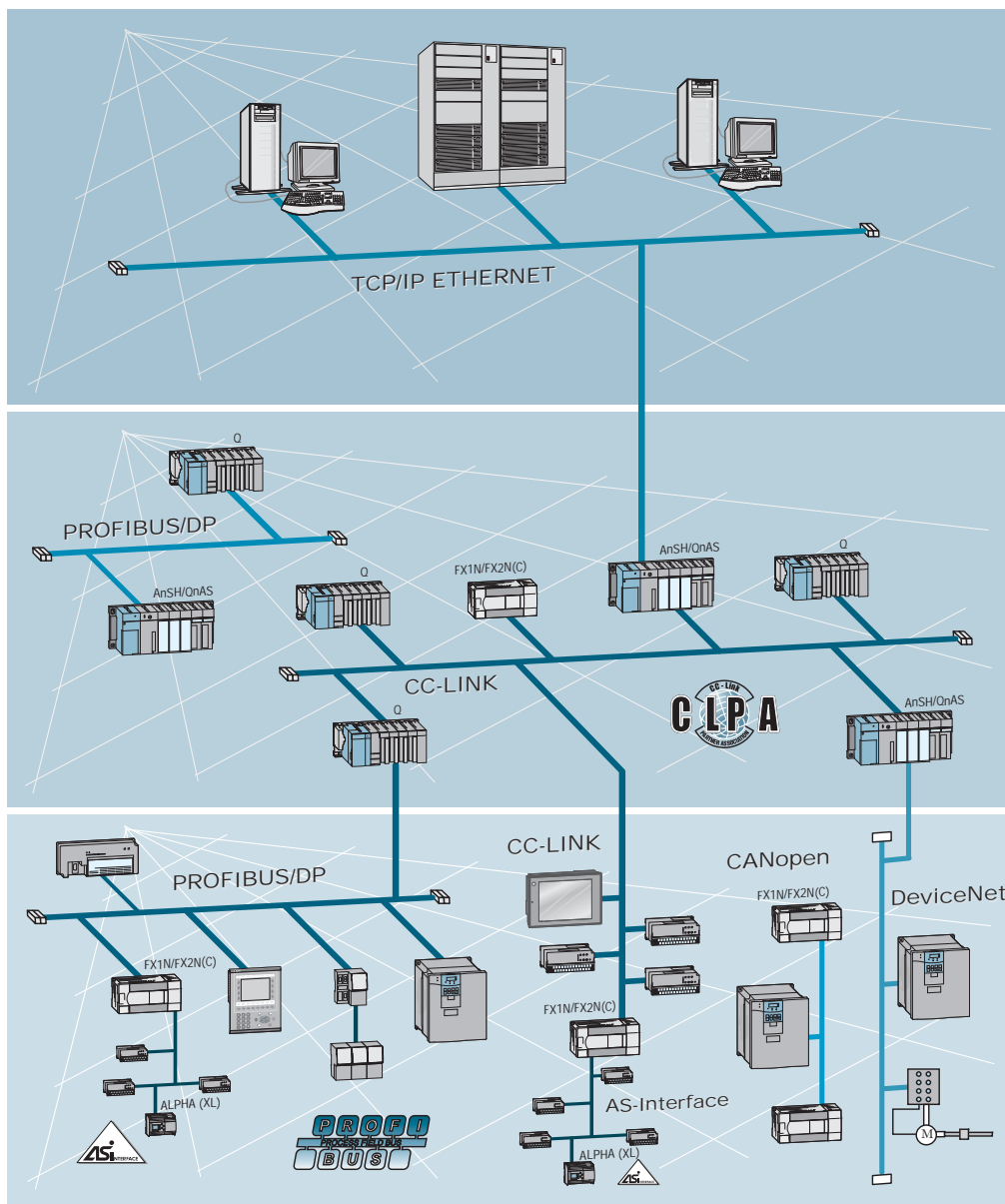
3.7.2 Открытые сети

Открытые сети являются независимыми от производителя, то есть они используются различными производителями. Это позволяет обеспечивать связь между ПЛК серии System Q и устройствами сторонних производителей.

Командный уровень
ETHERNET

Уровень управления
PROFIBUS/DP
CC-Link

Производственный
уровень
PROFIBUS/DP
DeviceNet
AS-Interface
CC-Link
CANopen



ETHERNET

ETHERNET является самой распространенной сетью для подключения устройств информационного уровня, таких как персональные компьютеры и рабочие станции. ETHERNET представляет собой платформу для множества самых разных протоколов обмена данными. Сочетание сети ETHERNET и широко распространенного протокола TCP/IP обеспечивает высокоскоростной обмен данными между системами управления процессами и ПЛК серии System Q.

Протокол TCP/IP обеспечивает прямую связь между двумя станциями в сети ETHERNET. Используя протокол TCP/IP, система управления процессом может выдавать до 960 слов данных на запрос, если применяется модуль серии System Q.

PROFIBUS/DP

Открытая сеть PROFIBUS/DP обеспечивает высокоскоростной обмен данными с множеством самых разных ведомых устройств, в том числе следующих:

- Удаленные цифровые и аналоговые входы/выходы;
- Преобразователи частоты;
- Терминалы операторов;
- Прочие устройства сторонних производителей.

Для снижения затрат на PROFIBUS/DP применяется технология RS 485 с экранированными 2-проводными кабелями.

CC-Link

Открытая шина и сеть управления CC-Link обеспечивают высокоскоростной обмен данными между различными устройствами. В числе прочих предусмотрена интеграция следующих компонентов от MITSUBISHI ELECTRIC:

- ПЛК серии System Q;
- Удаленные цифровые и аналоговые входы/выходы;
- Модули позиционирования;
- Преобразователи частоты;
- Терминалы операторов;
- Роботы;
- Устройства сторонних производителей, такие как считыватели штрих-кодов.

Данная сеть позволяет легко обмениваться цифровыми и аналоговыми данными. Помимо циклической передачи слов данных системы CC-Link обеспечивают также случайную передачу сообщений. Это позволяет осуществлять обмен данными не только с аналоговыми и цифровыми устройствами, но и с интеллектуальными устройствами, такими как дисплеи, считыватели штрих-кодов, измерительные приборы, персональные компьютеры и системы ПЛК (до 24 модулей ЦП).

DeviceNet

DeviceNet представляет собой экономичное решение для сетевой интеграции терминального оборудования низкого уровня. В одну сеть можно объединить до 64 устройств, включая ведущее.

Интерфейс AS

Интерфейс AS является международным стандартом для шин низшего уровня. данная сеть удовлетворяет разнообразным потребностям, очень гибкая и простая в установке. Предназначена для управления исполнительными устройствами (Actuators), такими как соленоиды и индикаторы, и датчиками (sensors) – отсюда и название AS-i.

CANopen

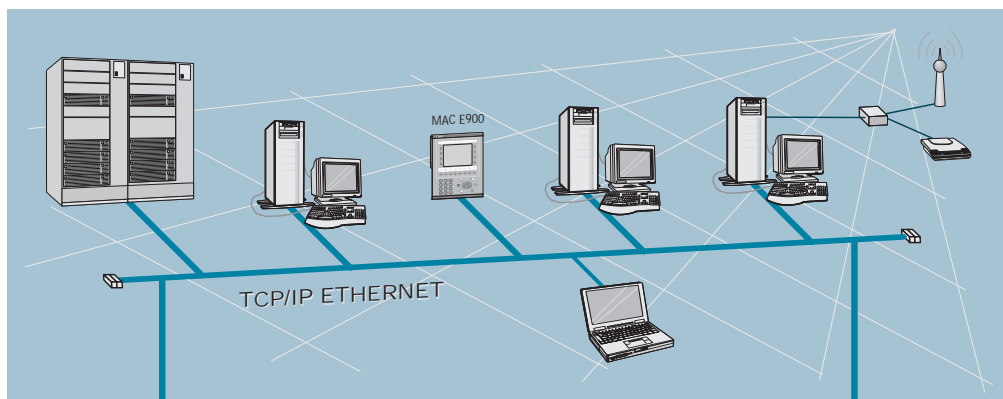
CANopen является «открытой» реализацией сети Controller Area Network (CAN).

Сети CANopen применяются для подключения датчиков, исполнительных устройств и контроллеров в промышленных системах управления. медицинской аппаратуре, морской электронике, на железной дороге, в трамваях и грузовых автомобилях.

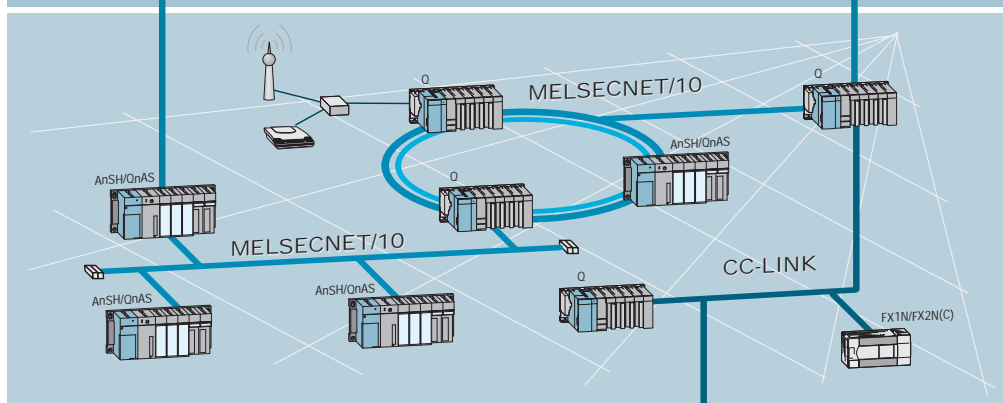
Сетевые модули CANopen доступны для контроллеров серии FX.

3.7.3 Сети MELSEC

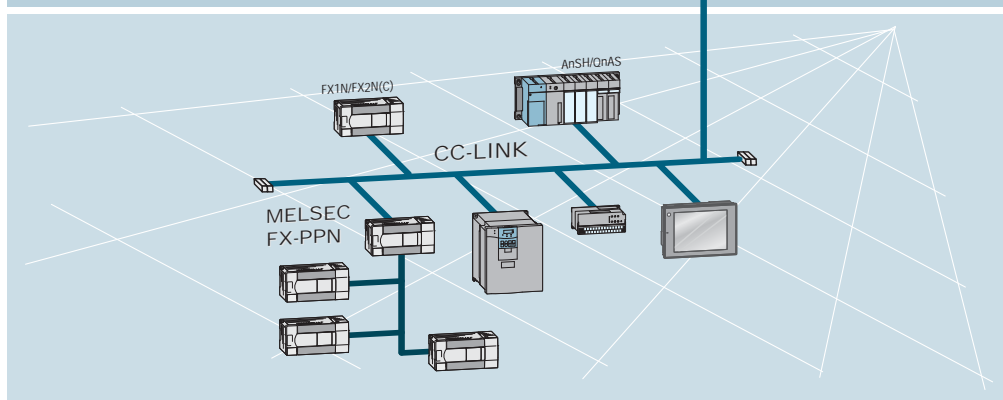
Командный уровень
TCP/IP ETHERNET



Уровень управления
CC-Link
MELSECNET/10
MELSECNET/H



Производственный уровень
CC-Link
MELSEC FX-PPN



MELSECNET/10/H

MELSECNET/10 и MELSECNET/H – это высокоскоростные сети для обмена данными между ПЛК серии System Q. В эти сети можно интегрировать даже удаленные станции ввода/вывода. MELSECNET/10/H позволяет с любой станции программировать и контролировать каждый ПЛК в системе.

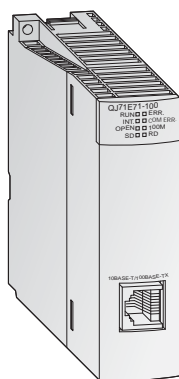
Можно соединять до 255 сетей MELSECNET/10/H. Функция встроенного маршрутизатора упрощает передачу данных из одной сети в другую. Предусмотрена возможность циклической передачи сверхбольшого объема данных (8192 слов и 8192 реле). Параллельно циклическому обмену данным с любой станции можно передавать данные на любую другую станцию и принимать от неё данные даже через несколько сетей.

MELSECNET/10 позволяет использовать самые разные типы кабелей и топологии: от коаксиальной шины (макс. 500 м) по коаксиальному двойному контуру до оптоволоконного двойного контура на расстояние до 30 км.

3.7.4 Сетевые модули

Интерфейсные модули ETHERNET

Модули QJ71E71/E71-100 и QD71E71-B2 используются на стороне ПЛК для подключения хост-системы, например к ПК или рабочей станции, и System Q через ETHERNET. Помимо передачи данных по протоколам TCP/IP и UDP/IP, обеспечивается поддержка чтения и изменения данных ПЛК, а также контроль работы модулей ПЛК и состояния управления.

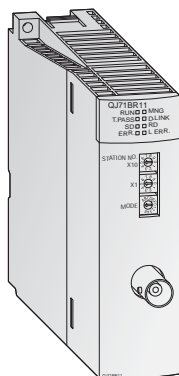


Специальные функции

- Типы сетей: 10BASE5, 10BASE2 и 10BASE-T;
- Скорость передачи 10/100 Мбит/с;
- Функция сервера FTP;
- Функция связи с использованием фиксированных буферов передачи и приема;
- Для одновременно передачи данных могут быть открыты до 16 линий связи;
- Программирование и контроль ПЛК может осуществляться на персональном компьютере с помощью программы GX Developer или GX IEC Developer через ETHERNET.

Модули MELSECNET

Модули QJ71BR11 и QJ71LP21 применяются для подключения аппаратуры System Q к сетевым модулям NET/10 или MELSECNET/H. Это обеспечивает быстрый и эффективный обмен данными между модулями серий Q, QnA и QnAS.

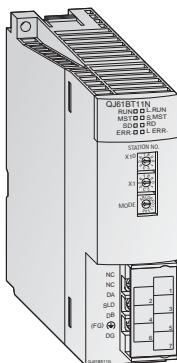


Специальные функции

- Две разные топологии: коаксиальная шина (QJ71BR11) или резервированный оптический контур (QJ71LP21);
- Высокая скорость передачи данных: 10 Мбит/с в системах с коаксиальной шиной и 10 или 20 Мбит/с в системах с оптическим контуром;
- Связь с другими ПЛК, ПК и удаленными станциями ввода/вывода;
- Сетевая система поддерживает обмен данными между двумя станциями, независимо от числа сетей между ними;
- Функция разделения станций в системе с коаксиальной шиной и функция возврата в системах с оптическим двойным контуром при неисправности станции;
- Функция смещения станции управления и функция автоматического возврата.

Ведущий/локальный модуль для CC-Link

Модули QJ61BT11N применяются в системах CC-Link в качестве ведущих и локальных станций для управления удаленными входами и выходами.



Специальные функции

- Параметры всех модулей в сети задаются напрямую через ведущий модуль;
- Обмен данными между удаленными модулями и ведущим модулем осуществляется автоматически. Частота обновления для точек ввод/вывода составляет 3,3 мс.
- Скорость передачи до 10 Мбит/с;
- При одном ведущем модуле систему можно расширять до 2028 точек ввод/вывода;
- При использовании дополнительного резервного ведущего модуля создается дублированная система. При возникновении ошибки на ведущей станции обмен данными не прерывается;
- Автоматический запуск CC-Link без настройки параметров;
- Прерывание запуска программы командой по сети.

Интерфейсные модули PROFIBUS/DP

Ведущие модули PROFIBUS/DP типа QJ71PB92D и QJ71PB92V ведомые модули PROFIBUS/DP типа QJ71PB93D позволяют ПЛК серии System Q обмениваться данными с другими устройствами PROFIBUS.

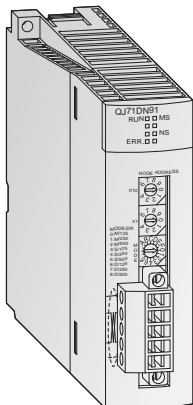
Специальные функции

- Ведущая станция может связываться с 60 ведомыми модулями;
- Одновременно на каждой ведомой станции может обрабатываться 244 входных байта и 244 выходных байта;
- Поддержка таких функций, как SYNC, FREEZE и специальных диагностических сообщений для конкретных типов ведомых модулей;
- Поддержка автоматического обновления при обмене данными. В качестве опции предусмотрена пакетная передача.

Ведущий модуль DeviceNet типа QJ71DN91

Модуль QJ71DN91 подключается к ПЛК серии Q с помощью модуля DeviceNet. DeviceNet представляет собой экономичное решение для сетевой интеграции терминального оборудования низкого уровня.

Специальные функции



- Положение ведущего и ведомых станций выбирается пользователем;
- Скорость передачи 125, 250 и 500 Кбод;
- Дальность передачи до 500 м;
- Методы связи:
 - Опрос;
 - Бит строба;
 - Изменение состояния;
 - Циклический.

Ведущие модули для интерфейса AS

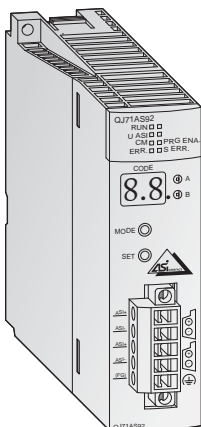
Ведущий модуль QJ71AS92 служит для подключения аппаратуры System Q к системам с AS-интерфейсом.

Модуль QJ71AS92 может управлять 62 ведомыми модулями (группа А: 31 / группа В: 31) с 4 входами и 4 выходами на каждом адресе. Адреса ведомых устройств в системе с AS-интерфейсом назначаются автоматически ведущим модулем.

Максимальная дальность передачи без повторителя составляет 100 м.

При использовании двух повторителей максимальную дальность передачи можно увеличить до 300 м.

Специальные функции



- В двух сетях можно подключать до 62 ведомых модулей;
- Ведущий модуль может управлять 496 дискретными входами/выходами;
- Связь через плоский или круглый кабель AS-i с цветной индикацией жил;
- Высокоэффективная система защиты от ошибок;
- Автоматический обмен данными с ПЛК.

Модуль веб-сервера

Модуль веб-сервера QJ71WS96 предоставляет возможность удаленного слежения и управления ПЛК серии Q.



Специальные функции

- Доступ к ПЛК через Интернет;
- Простые в использовании встроенные функции настроек;
- Для слежения и настройки пользователю нужен только веб-браузер;
- Интерфейс RS232 для подключения модема;
- Для обмена данными возможны различные виды подключения: ADSL, модем, LAN (локальная сеть) и др.;
- Отправка и получение данных по электронной почте или FTP;
- Возможна интеграция веб-сайта и приложений Java собственной разработки;
- Стандартное соединение по сети ETHERNET для обмена данными с другими ПЛК или ПК;
- Функции регистрации событий и данных ЦП.

4 Введение в программирование

Программа состоит из последовательности программных команд. Данные команды определяют функции ПЛК и обрабатываются последовательно в том порядке, в каком они были введены. Чтобы создать программу ПЛК, нужно проанализировать данный управляемый процесс и разбить его на отдельные шаги, которые можно представить командами. Программная команда, представленная строкой или «ступенькой» в формате Ladder Diagram, является простейшей единицей прикладной программы ПЛК.

4.1 Структура программной команды

Программная команда состоит из самой команды (или инструкции) и одного или нескольких (в случае прикладных команд) операндов, которые в ПЛК относятся к устройствам. Некоторые команды вводятся без операндов и служат для управления выполнением программы в ПЛК.

Каждой команде при вводе присваивается номер шага для уникальной идентификации её положения в программе. Это позволяет вводить одну и ту же команду, относящуюся к одному устройству, в нескольких местах программы.

На следующих рисунках показано представление программных команд в форматах языков программирования Ladder Diagram (LD, слева) и Instruction List (IL, справа).



Данная команда описывает что нужно сделать, т.е. функцию, которую должен выполнить контроллер. Операнд или устройство – это объект действия данной функции. Его обозначение состоит из двух частей: имени устройства и адреса устройства:



Примеры устройств

Имя устройства	Тип	Назначение
X	Входное	Входная клемма ПЛК (например, подключенная к выключателю)
Y	Выходное	Выходная клемма ПЛК (например, для контактора или лампы)
M	Релейное	Буферная память ПЛК, имеет два состояния: ВКЛ или ВЫКЛ
T	Таймер	Реле времени для программирования функций, зависящих от времени
C	Счётчик	Счётчик
D	Регистр данных	Область хранения данных в ПЛК, где хранятся такие данные, как измеренные значения и результаты вычислений.

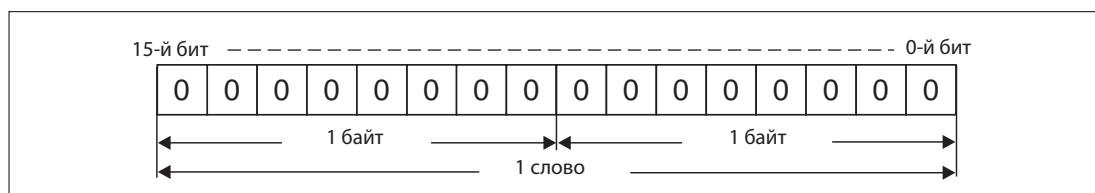
Подробное описание устройств приведено в разделе 5.

Конкретное устройство идентифицируется по своему адресу. Например, поскольку каждый контроллер имеет множество входов, для чтения конкретного входа необходимо указать как имя устройства, так и его адрес.

4.2 Биты, байты и слова

Как и во всей цифровой технологии, наименьшей единицей информации в ПЛК является «бит». Бит может иметь только два состояния: «0» (ВЫКЛ или ЛОЖЬ) и «1» (ВКЛ или ИСТИНА). ПЛК содержат различные так называемые битовые устройства, которые могут находиться только в двух состояниях. К ним относятся входы, выходы и реле.

Следующими по размеру единицами информации являются «байт», состоящий из 8 битов, и «слово», состоящее из двух байтов. В ПЛК серии System Q регистры данных являются устройствами типа «слово», т.е. они могут хранить 16-разрядные значения.



Ширина регистра данных составляет 16 бит, поэтому в нем могут храниться значения со знаком в диапазоне от -32 768 до +32 767 (см. следующий раздел 4.3). Для хранения больших значений используется сочетание двух слов, составляющее 32-разрядное слово, в котором могут храниться значения со знаком в диапазоне от -2 147 483 648 до +2 147 483 647.

4.3 Системы счисления

В ПЛК серии System Q используется несколько систем счисления для ввода и отображения значений, а также для указания адресов устройств.

Десятичные числа

Десятичная система счисления наиболее распространена в повседневной жизни. Это позиционная система с основанием 10, в которой каждый разряд (позиция) числа в десять раз больше значения разряда справа. После того, как счет в позиции доходит до 9, происходит обнуление текущей позиции, а следующая позиция увеличивается на 1, означая следующий десяток (9 → 10, 99 → 100, 199 → 1,000 и т.д.).

- Основание: 10
- Значения: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

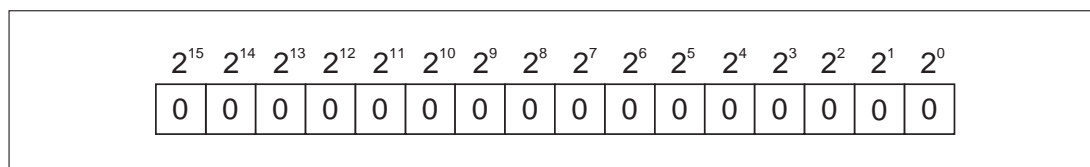
В ПЛК серии System Q десятичные числа используются для ввода констант и значений уставок таймеров и счётчиков. Адреса устройств также вводятся в десятичном формате, за исключением адресов входов и выходов.

Двоичные числа

Как и все вычислительные устройства, ПЛК различает только два состояния: ВКЛ/ВЫКЛ и 0/1. Эти двоичные состояния хранятся в отдельных разрядах. Если требуется ввод или отображение чисел в другом формате, программа автоматически преобразовывает двоичные числа в другую систему счисления.

- Основание: 2
- Значения: 0 и 1

Когда двоичное число хранится в слове (см. выше), значение каждого разряда (позиции) в слове в 2 раза больше значения разряда справа. Принцип представления аналогичен десятичному, только вместо шага 10 используется 2 (см. рис.).



Двоичное представление	Десятичное значение	Двоичное представление	Десятичное значение
2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1024
2^3	8	2^{11}	2048
2^4	16	2^{12}	4096
2^5	32	2^{13}	8192
2^6	64	2^{14}	16384
2^7	128	2^{15}	32768*

* В двоичных числах для представления знака используется 15-й разряд (15-й разряд = 0 – положительное значение, 15-й разряд = 1 – отрицательное значение)

Чтобы преобразовать двоичное число в десятичное, нужно умножить значение 1 каждого разряда на 2 в соответствующей степени и сложить результаты всех разрядов.

Пример

00000010 00011001 (двоичное)

00000010 00011001 (двоичное) = $1 \times 2^9 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0$

00000010 00011001 (двоичное) = $512 + 16 + 8 + 1$

00000010 00011001 (двоичное) = 537 (десятичное)

Шестнадцатеричные числа

Шестнадцатеричными числами легче оперировать, чем двоичными. Преобразовывать двоичные числа в шестнадцатеричные тоже очень просто. Поэтому шестнадцатеричные числа широко используются в цифровых технологиях и программируемых логических контроллерах. В ПЛК серии System Q шестнадцатеричные числа используются для нумерации входов и выходов и представления констант. В данном руководстве по программированию и в других руководствах шестнадцатеричные числа, во избежание путаницы с десятичными числами, всегда обозначаются буквой H на конце (например, 12345H).

- Основание: 16
- Значения: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
(буквы A, B, C, D, E и F представляют десятичные значения 10, 11, 12, 13, 14 и 15)

Шестнадцатеричная система счисления аналогична десятичной, только обнуление и увеличение следующего разряда происходит, когда счёт доходит до FH (15) вместо 9 (FH → 10H, 1FH → 20H, 2FH → 30H, FFH → 100H и т.д.). Значение разряда получается возведением в степень основания 16, а не 10:

1A7FH			
	$16^0 = 1$	(в данном примере: 15×1	= 15)
	$16^1 = 16$	(в данном примере: 7×16	= 112)
	$16^2 = 256$	(в данном примере: 10×256	= 2560)
	$16^3 = 4096$	(в данном примере: 1×4096	= 4096)
			<u>6783</u> (десятичное)

Следующий пример иллюстрирует простоту преобразования двоичных значений в шестнадцатеричные.

1	1	1	1	0	1	1	0	1	0	1	1	1	0	0	1	Двоичное
																Десятичное*
15	5	11	9	F	5	B	9									Шестнадцатеричное

* Преобразование 4-разрядных блоков в десятичные значения не позволяет напрямую получить значение, полностью соответствующее 16-разрядному двоичному значению. И наоборот, двоичное значение можно напрямую преобразовать в шестнадцатеричное с точно таким же значением, что и двоичное.

Восьмеричные числа

Восьмеричные числа указаны здесь только для полноты изложения. Они не используются в ПЛК серии System Q. В восьмеричных системах отсутствуют значения 8 и 9. В этой системе обнуление текущего разряда и увеличение следующего разряда происходит, когда счёт доходит до 7 (0 – 7, 10 – 17, 70 – 77, 100 – 107 и т.д.).

- Основание: 8
- Значения: 0, 1, 2, 3, 4, 5, 6, 7

Заключение

В следующей таблице представлен обзор четырёх систем счисления.

Десятичная	Восьмеричная	Шестнадцатеричная	Двоичная
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 01110 0011
:	:	:	:

4.4 Коды

В целях обеспечения надежного и эффективного обмена данными буквы и десятичные числа необходимо преобразовывать в код, который понятен исполнительным механизмам.

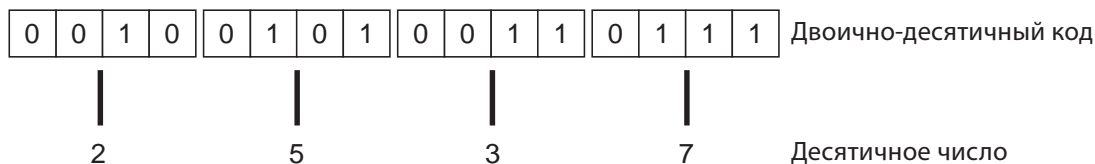
4.4.1 Двоично-десятичный код

Двоично-десятичный код представляет собой систему кодирования десятичных чисел, при которой каждая цифра (от 0 до 9) представлена 4-разрядным двоичным числом (от 0000 до 1001), см. таблицу ниже. Таким образом, в одном байте (8 бит) может храниться два десятичных числа.

Десятичное число	Двоично-десятичный код
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Для преобразования десятичных чисел, состоящих из нескольких цифр, используется сочетание двоично-десятичных кодов отдельных цифр. Двоично-десятичный код четырёхзначного числа занимает одно слово (16 бит) и может находиться в диапазоне от 0000 до 9999.

Пример



В System Q двоично-десятичный код для внутренних операций не используется. Однако в условиях промышленного применения двоично-десятичный код зачастую используется для входных значений и вывода значений на светодиодном индикаторе. Для этих случаев предусмотрено несколько команд по преобразованию в двоично-десятичный код и обратно.

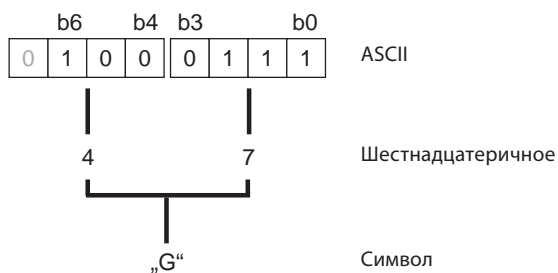
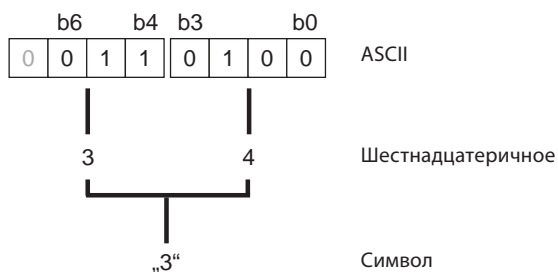
4.4.2 ASCII-код

ASCII – это сокращение от American Standard Code for Information Interchange. В ASCII-коде 7 разрядов представляют буквенно-цифровые символы, знаки препинания, управляющие и прочие символы.

Данные в ASCII-кодах используются для связи с периферийными устройствами.

Биты с 3 по 0		Биты с 6 по 4							
		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	'	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	!!	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB		7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	VS	/	?	O	←	o	DEL

Примеры



4.5 Языки программирования

В программном пакете GX IEC Developer предусмотрены отдельные редакторы для программирования. Можно выбирать ввод и отображение программ в графическом или текстовом режиме. В редакторах для всех языков, кроме Sequential Function Chart, программы ПЛК делятся на разделы, называемыми «сетями».

4.5.1 Текстовые редакторы

Instruction List (IL)

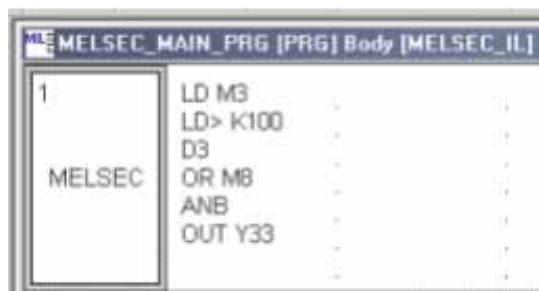
Рабочая область в Instruction List (IL) представляет собой простой текстовый редактор для непосредственного ввода команд. Каждая команда должна содержать оператор (функцию) и один или несколько операндов. Каждая команда начинается с новой строки.

Используются два типа Instruction List:

- IEC Instruction List



№	Команда
1	LD FALSE ST M101
2 Start:	LD Hauptschalter AND M100 ST M102
3	LD M100 ANDN M1 CJ_M Start



№	Команда
1 MELSEC	LD M3 LD> K100 D3 OR M8 ANB OUT Y33

- MELSEC Instruction List

В MELSEC Instruction List можно использовать только набор команд MELSEC; стандартное программирование IEC невозможно.

Structured Text (ST)

Structured Text представляет собой полезный инструмент, особенно для пользователей, привыкших работать с системами PC. При внимательном программировании с учетом работы ПЛК данный редактор очень удобен.

Редактор Structured Text совместим со стандартом IEC 61131-3 и отвечает всем его требованиям.

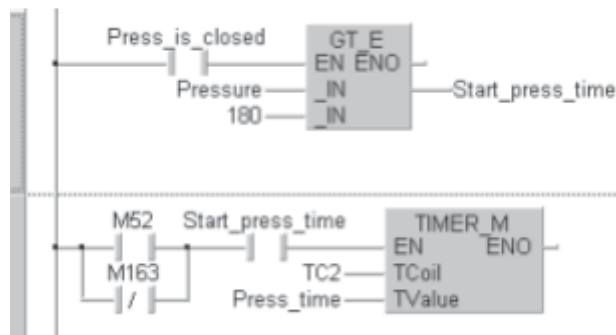
(*Example showing Structured Text*)

```
Y20:=X10;
Y21:=X11 AND X12 OR X13;
M0:=(M1 AND (M2 OR M3)) OR X14;
```

4.5.2 Графические редакторы

Ladder Diagram

Программирование в Ladder Diagram очень похоже на создание принципиальных схем в обычных релейных системах. Схема Ladder Diagram состоит из входных контактов (замыкателей и прерывателей), выходных катушек, функциональных блоков и функций. При соединении данных элементов горизонтальными и вертикальными линиями получается схема. Схемы всегда начинаются со сборной шины (шины электропитания), расположенной слева.

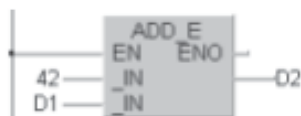


Пример схемы в Ladder Diagram

Для большинства наиболее используемых прикладных команд в формате Ladder Diagram на панели инструментов предусмотрены кнопки.



В Ladder Diagram сложные функции и функциональные блоки отображаются прямоугольниками. Помимо необходимых для функций входов и выходов функции и функциональные блоки имеют входы EN и выходы ENO. Вход EN (EN = ENable, разрешен) контролирует выполнение команды.



Эта команда выполняется в цикле.



Эта команда выполняется только при включенном M12.

Результат данной операции передается на выход ENO (ENO = ENable Out, выход разрешен).

M34 устанавливается, когда команда сравнения обнаруживает идентичность содержимого двух устройств.



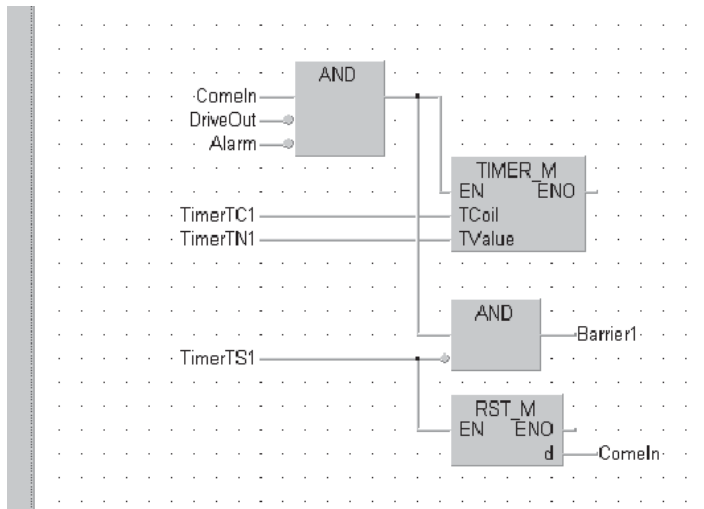
Для контроля за ходом выполнения программы можно подключить выходы ENO и входы EN. В следующем примере выполнение второй команды зависит от результата выполнения первой.



Function Block Diagram

Все команды реализуются с помощью блоков, соединенных между собой горизонтальными и вертикальными соединительными элементами. На этих блок-схемах шины электропитания отсутствуют.

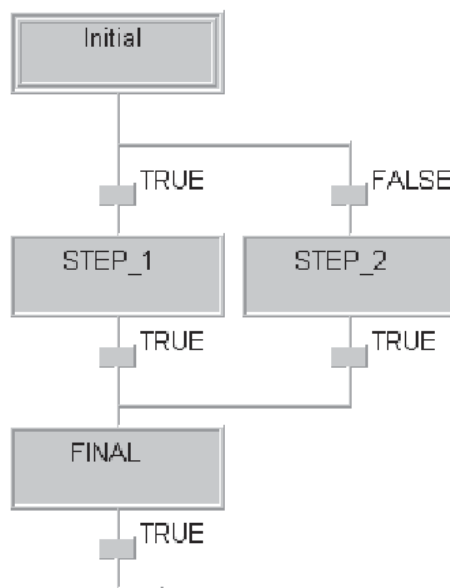
Пример схемы в Function Block Diagram



Sequential Function Chart

Sequential Function Chart (SFC) представляет собой структурированный язык, позволяющий четко представлять сложные процессы.

В Sequential Function Chart используются два основных элемента: шаги и переходы. Последовательность состоит из серии шагов, разделенных между собой переходами. В любое время может быть активен только один из шагов последовательности. Следующий шаг активируется только после завершения предыдущего, при соблюдении условий для перехода.



4.6 Стандарт IEC 61131-3

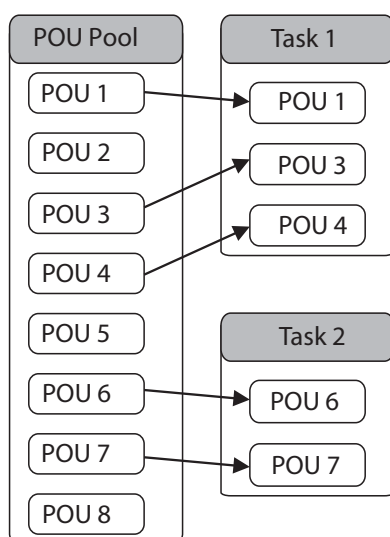
IEC 61131-3 – это международный стандарт для программ ПЛК, установленный Международной электротехнической комиссией (IEC). Стандарт IEC 61131-3 охватывает не только языки программирования ПЛК, но и включает рекомендации по применению ПЛК. ПЛК можно программировать по стандарту IEC 61131-3 с помощью программного пакета GX IEC Developer.

В данном руководстве для начинающих пользователей приведено описание только тех терминов, которые необходимы для понимания примеров программ. Дополнительная информация по программному пакету GX IEC Developer содержится в руководстве для начинающих пользователей (кат. № 043596) и в справочном руководстве (кат. № 043597) для данного программного пакета. В процессе программирования можно также пользоваться функцией справки GX IEC Developer.

4.6.1 Структура программы

Программный модуль (POU)

По стандарту IEC 61131-3 программа ПЛК делится на отдельные программные модули POU (Program Organisation Unit). Программный модуль – это наименьший независимый элемент последовательной программы.



Программные модули хранятся в так называемом пуле программных модулей (POU Pool).

Программные модули сгруппированы в задачи.

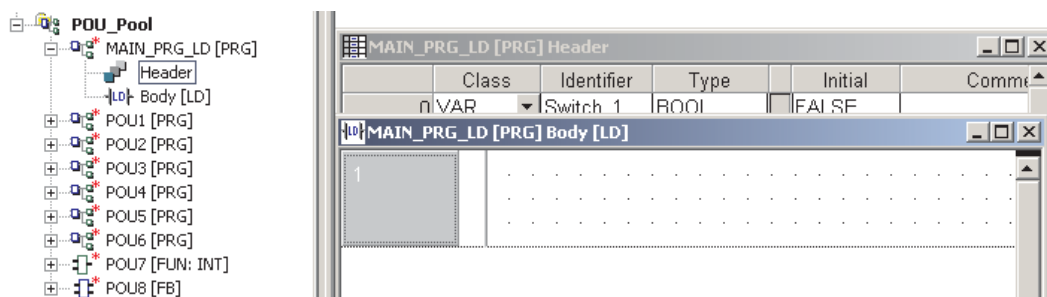
Отдельные задачи группируются, формируя реальную программу ПЛК.

Каждый программный модуль содержит:

- Заголовок
- Тело

Заголовок служит для объявления переменных, используемых в данном программном модуле.

Тело является частью проекта, где редактируется программа. Для редактирования программ предусмотрено несколько языков.

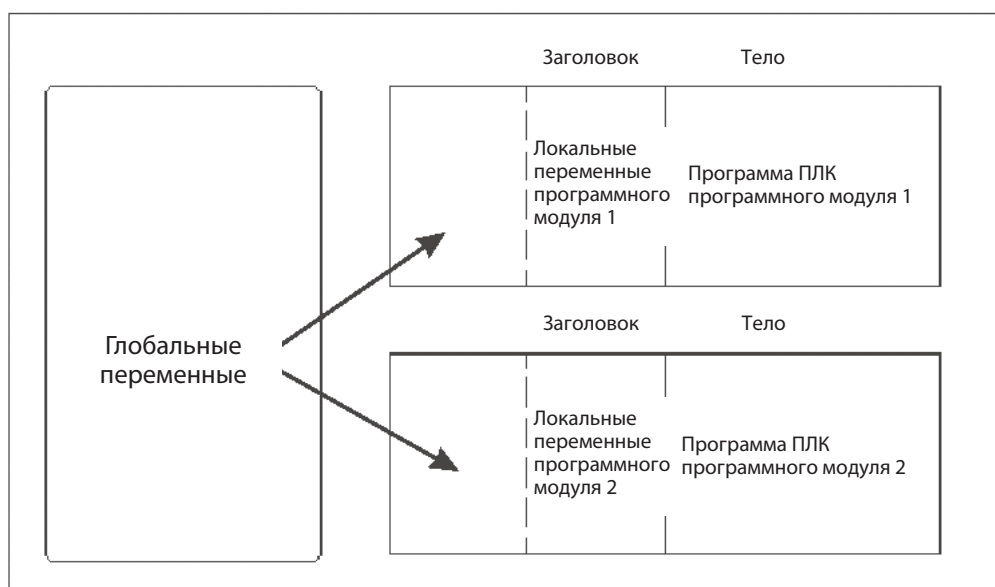


4.6.2 Переменные

Переменные содержат значения входов, выходов и внутренних устройств ПЛК. Существует два типа переменных:

- Глобальные переменные
- Локальные переменные

Глобальные переменные считаются «общими» переменными и используются для обращения к физическим устройствам ПЛК. Они доступны для всех программных модулей и связаны с физическими входами/выходами ПЛК или именованными внутренними устройствами ПЛК. Глобальные переменные позволяют обмениваться данными между отдельными программными модулями.



Для доступа программного модуля к какой-то глобальной переменной она должна быть объявлена в заголовке данного модуля. В заголовок могут входить как глобальные переменные, так и локальные.

Локальные переменные предназначены для промежуточных результатов. Доступ программного модуля к локальным переменным других модулей отсутствует.

Объявление переменных

В начале каждого программного модуля объявляются переменные, т.е. им назначается определенный тип данных, например INT или BOOL.

Каждая переменная содержит следующие элементы:

- Класс
- Идентификатор (имя переменной)
- Абсолютный адрес (необязательный для глобальных переменных)
- Тип данных
- Исходное значение (задается автоматически)
- Комментарий (необязательно)

Global Variable List							
	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial	Comment
0	VAR_GLOBAL	START_MOTOR	X10	%IX16	BOOL	FALSE	
1	VAR_GLOBAL	STOP_MOTOR	X11	%IX17	BOOL	FALSE	
2	VAR_GLOBAL	MOTOR_1_ON	Y20	%QX32	BOOL	FALSE	

Класс

С помощью класса переменной назначается свойство, определяющее её использование в проекте. Примеры:

- VAR: локальная переменная для использования в программном модуле
- VAR_EXTERNAL: внешняя глобальная переменная, объявленная в списке глобальных переменных, которую могут читать и записывать все программные модули.
- VAR_CONSTANT: локальная переменная с постоянным значением для использования в программном модуле

Идентификатор

Каждой переменной назначается символический адрес. Индивидуальное имя (идентификатор) может быть любым, но должно начинаться с буквы или единичного символ подчеркивания. Не допускается использование пробелов и математических операторов (например, +, -, *).

Примеры идентификаторов:

- S02.3
- Drive_2_ready
- _Open_Valve
- Motor_M1_ON

Использование символических объявлений соответствует стандарту IEC 61131.3.

Абсолютные адреса

При объявлении глобальных переменных им также следует назначать абсолютные адреса. Если абсолютные адреса не назначены вручную, то они назначаются автоматически. Абсолютный адрес указывает ячейку памяти переменной в ЦП, входе или выходе.

Абсолютные адреса можно назначать с использованием синтаксиса IEC (IEC-адрес) или MEL-SEC (MIT-адрес). Примеры абсолютных адресов:

Вход X0F = X0F (синтаксис MELSEC) = %IX15 (синтаксис IEC)

Выход Y03 = Y03 (синтаксис MELSEC) = %QX3 (синтаксис IEC)

Типы элементарных данных

Тип данных определяет характеристики переменной, такие как диапазон значений и число разрядов.

Тип данных	Диапазон значений	Длина	
BOOL	Булевы	0 (ЛОЖЬ), 1 (ИСТИНА)	1 бит
INT	Целые	от -32768 до +32767	16 бит
DINT	Целые двойной длины	от -2 147 483 648 до 2 147 483 647	32 бита
WORD	16-битовая строка	от 0 до 65535	16 бит
DWORD	32-битовая строка	от 0 до 4 294 967 295	32 бита
REAL	Значения с плавающей запятой	3,4E +/-38 (7 цифр)	
TIME	Значение времени	от -T#24d0h31m23s64800 до T#24d20h31m23s64700 мс	
STRING	Символьная строка	Длина символьной строки ограничивается 16 символами	

4.7 Набор основных команд

Команды ПЛК серии System Q можно разделить на две основные категории: основные команды и прикладные команды.

Функции основных команд сопоставимы с функциями, реализуемыми в контроллерах на релейно-контактной логике путем физического подключения.

Краткое описание набора основных команд

Команда	Назначение	Описание	Ссылка
LD	Нагрузка	Исходная логическая операция, опрос состояния сигнала «1» (нормально разомкнутый контакт)	Раздел 4.7.1
LDI	Инверсия нагрузки	Исходная логическая операция, опрос состояния сигнала «0» (нормально замкнутый контакт)	
OUT	Команда вывода	Назначение устройству результата логической операции	Раздел 4.7.2
AND	Логическое И	Операция логического И, опрос состояния сигнала «1»	Раздел
ANI	И НЕ	Операция логического И НЕ, опрос состояния сигнала «0»	
OR	Логическое ИЛИ	Операция логического ИЛИ, опрос состояния сигнала «1»	Раздел
ORI	ИЛИ НЕ	Операция логического ИЛИ НЕ, опрос состояния сигнала «0»	
ANB	И блок	Последовательное подключение блока параллельного ответвления цепи к предыдущему параллельному блоку	Раздел
ORB	ИЛИ блок	Параллельное подключение последовательного блока цепей к предыдущему последовательному блоку	
LDP	Команды импульсных сигналов	Нагрузка при импульсе: нагрузка при обнаружении нарастающего фронта импульса сигнала устройства	Раздел 4.7.7
LDF		Нагрузка при падающем импульсе: нагрузка при падающем импульсе сигнала устройства	
ANDP		И импульс: логическое И при нарастающем импульсе сигнала устройства	
ANDF		И падающий импульс: логическое И при падающем импульсе сигнала устройства	
ORP		ИЛИ импульс: логическое ИЛИ при нарастающем импульсе сигнала устройства	
ORF		ИЛИ падающий импульс: логическое ИЛИ при падающем импульсе сигнала устройства	
SET	Установка устройства	Назначение состояния сигнала, которое сохраняется даже после того, как условие на входе уже не является истиной	Раздел
RST	Сброс устройства		
PLS	Команды импульсов	Импульс: установка устройства на один рабочий цикл при нарастающем импульсе условия на входе (вход включается)	Раздел 4.7.9
PLF		Падающий импульс: установка устройства на один рабочий цикл при падающем импульсе условия на входе (вход отключается)	
INV	Инверсия	Инверсия результата операции	Раздел 4.7.10
FF	Инверсия бита	Инверсия битового выходного устройства	Раздел 4.7.11
MEP	Преобразование результата в импульс	Формирование импульса при нарастающем фронте результата операции	Раздел 4.7.12
MEF		Формирование импульса при падающем фронте результата операции	

4.7.1 Запуск логических операций

Команда	Назначение	Ladder Diagram	IEC Instruction List
LD	Команда нагрузки: запуск логической операции и опрос указанного устройства на предмет состояния сигнала «1»		LD
LDI	Команда нагрузки: запуск логической операции и опрос указанного устройства на предмет состояния сигнала «0»		LDN

Схема в программе всегда начинается с команды LD или LDI. Эти команды применяются к входам, реле, таймерам и счётчикам.

Примеры использования данных команд приведены в следующем разделе, см. описание команды OUT.

4.7.2 Вывод результата логической операции

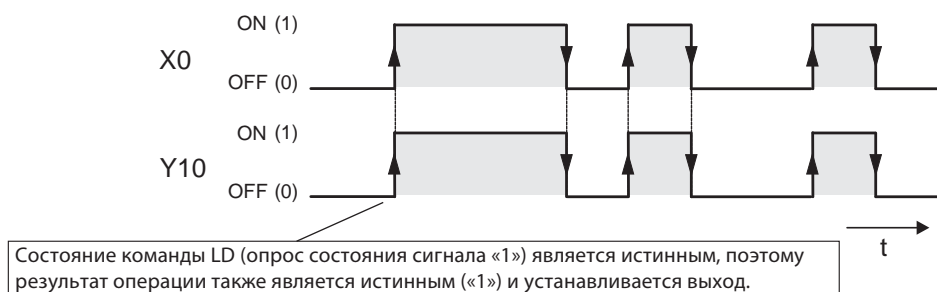
Команда	Назначение	Ladder Diagram	IEC Instruction List
OUT	Команда вывода: назначение устройству результата операции		ST

Для окончания цепи можно использовать команду OUT. Можно также программировать схемы, используя результаты нескольких команд OUT. Это не обязательно должно быть завершение программы. Устройство устанавливается по результату операции с помощью команды OUT, затем может использоваться как состояние входного сигнала в последующих шагах программы.

Пример (команды LD и OUT)

Ladder Diagram	MELSEC Instruction List	IEC Instruction List
	LD X0 OUT Y10	LD X0 ST Y10

Результатом двух данных команд является следующая последовательность сигнала.



Пример (команды LDI и OUT)

Ladder Diagram

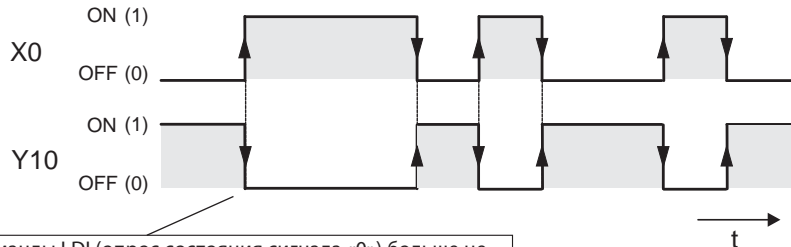


MELSEC Instruction List

LDI X0
OUT Y10

IEC Instruction List

LDI X0
ST Y10

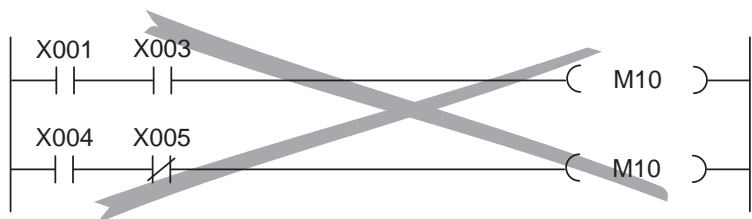


Состояние команды LDI (опрос состояния сигнала «0») больше не является истинным, поэтому выход сбрасывается.

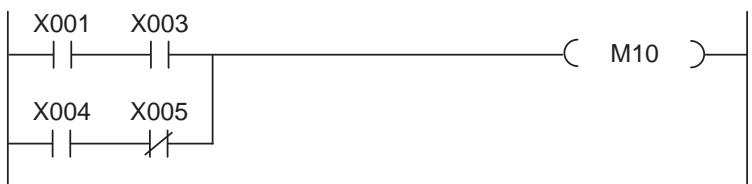
ПРИМЕЧАНИЕ Двойное назначение реле и выходов

Результат операции можно назначать одному и тому же устройству только в одном месте программы.

Программа выполняется последовательно, сверху вниз, поэтому в данном примере второе назначение M10 просто переписывает результат первого



Справа показано решение проблемы путем внесения изменения. При этом учитываются все необходимые условия на входе и получается корректный результат.





Но для каждого правила существуют исключения. Можно воспользоваться преимуществом выполнения программ ПЛК сверху вниз и размещать команды с высоким приоритетом в конце программы, чтобы преднамеренно перезаписывать предыдущие результаты. Соответствующий пример приведен в разделе 4.9.1. В этом случае для сброса внутренних устройств ПЛК и остановки электродвигателя используются средства защиты. Но при этом выходы для электродвигателя в программе назначаются только один раз.

4.7.3 Использование выключателей и датчиков

Прежде чем перейти к описанию остальных команд, рассмотрим как заводятся в программу сигналы с выключателей, датчиков и т.д.

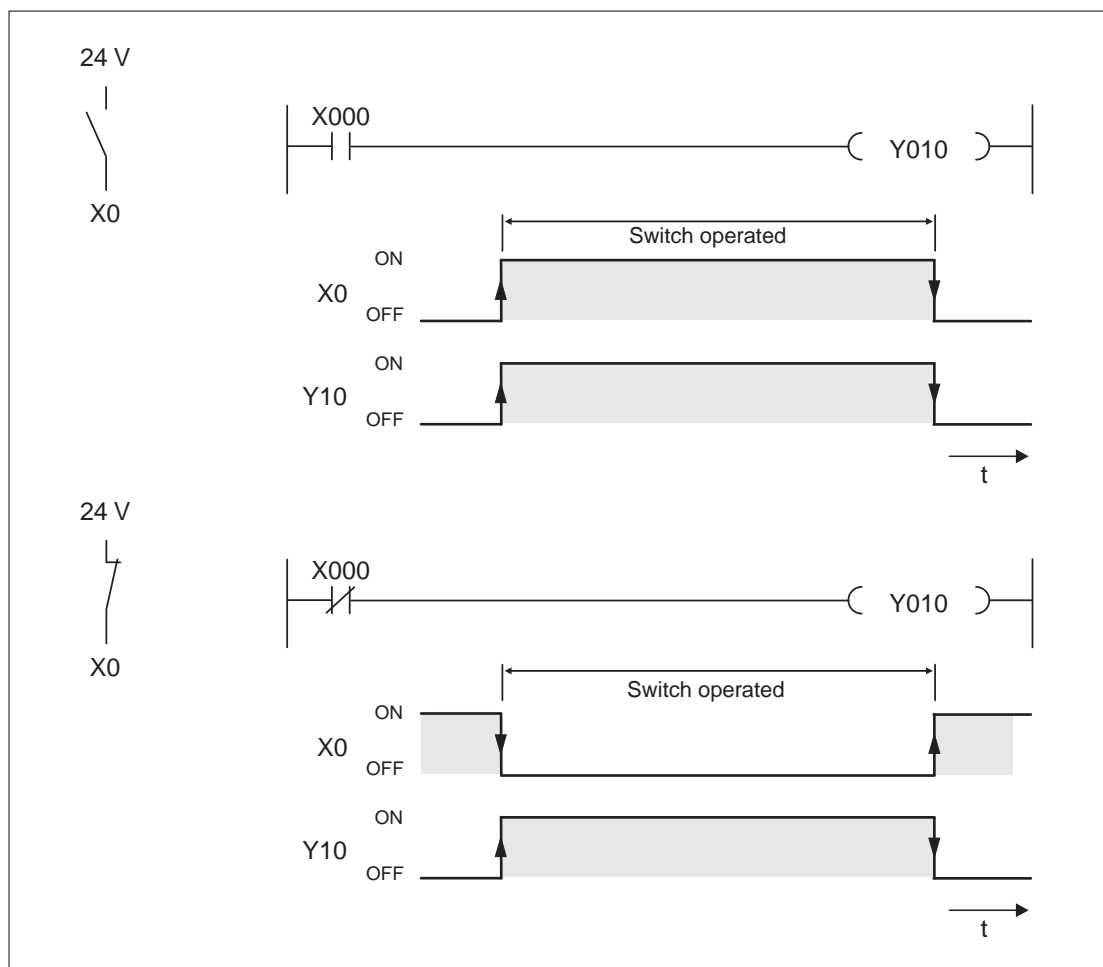
Чтобы обеспечивать правильное выполнение функций, программы ПЛК должны реагировать на сигналы с выключателей, кнопок, датчиков и т.д. Важно учитывать, что программные команды могут опрашивать только двоичное состояние сигнала указанного входа, независимо от типа входа и способа его контроля.

	Нормально разомкнутый контакт (замыкающий контакт)	При замыкании нормально разомкнутого контакта вход устанавливается (ВКЛ, состояние сигнала «1»)
	Нормально замкнутый контакт (размыкающий контакт)	При размыкании нормально замкнутого контакта вход сбрасывается (ВЫКЛ, состояние сигнала «0»)

Это означает, что при создании программы нужно учитывать, устройство какого типа подключается к входу ПЛК – замыкающее или размыкающее. При подключении ко входу замыкающего устройства программа выполняется по-другому, чем при подключении размыкающего. Это видно из данного примера.

Обычно используются выключатели с замыкающими контактами. Но иногда, по соображениям безопасности, используется размыкающие контакты, например, для отключения приводов (см. раздел 4.8).

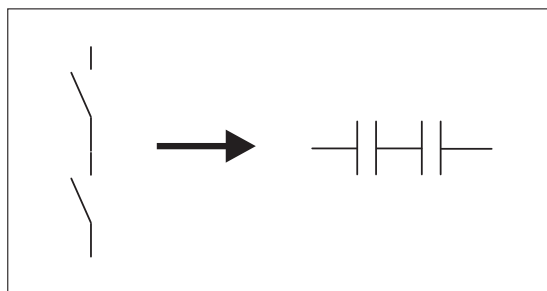
На следующем рисунке показаны две программные последовательности с одинаковым результатом, в которых используются выключатели разного типа. Когда выключатель срабатывает, устанавливается выход (включается).



4.7.4 Операции AND

Команда	Назначение	Ladder Diagram	IEC Instruction List
AND	Логическое И (операция И с опросом состояния сигнала «1» или ВКЛ)		AND
ANI	Логическое И НЕ (операция И с опросом состояния сигнала «0» или ВЫКЛ)		ANDN

Логика операции И аналогична последовательному подключению двух или более выключателей в электрической цепи. Ток течет только тогда, когда замкнуты все выключатели. Если один или несколько выключателей разомкнуты, ток не течет – состояние И является ложным.

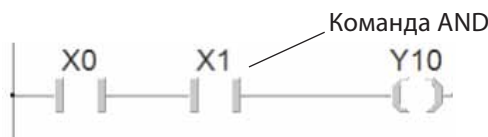


В данном пакете для программирования для команд AND и ANI используются такие же значки и функциональные клавиши, как для команд LD и LDI. При программировании в формате Ladder Diagram правильное назначение команд производится автоматически, на основе позиции вставки.

При программировании в формате Instruction List команды AND и ANI нельзя использовать в начале схемы (строка программы в формате Ladder Diagram). Схемы должны начинаться с команды LD или LDI (см. раздел 4.7.1).

Пример команды AND

Ladder Diagram



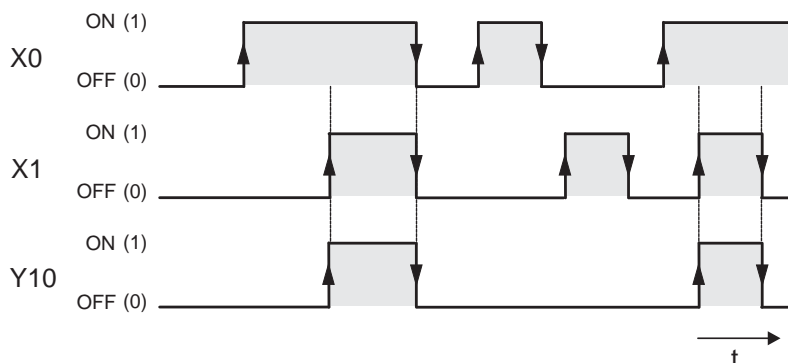
MELSEC Instruction List

```
LD    X0
AND   X1
OUT   Y10
```

IEC Instruction List

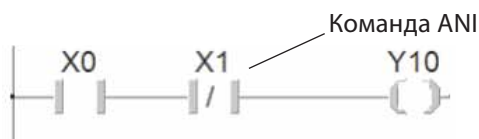
```
LD    X0
AND   X1
ST    Y10
```

В данном примере выход Y10 включается только тогда, когда входы X0 и X1 оба включены.



Пример команды ANI

Ladder Diagram



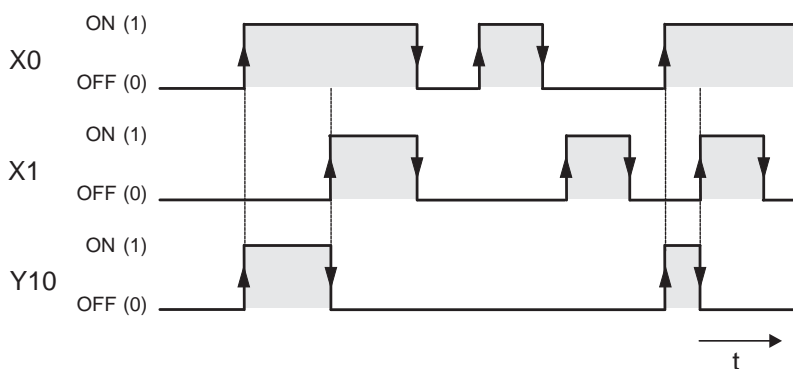
MELSEC Instruction List

```
LD    X0
ANI   X1
OUT   Y10
```

IEC Instruction List

```
LD    X0
ANDN  X1
ST    Y10
```

В данном примере выход Y10 включается только тогда, когда вход X0 включен и вход X1 выключен.

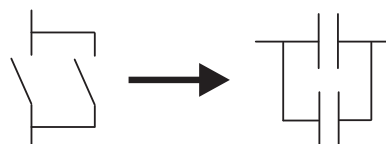


4.7.5

Операции OR

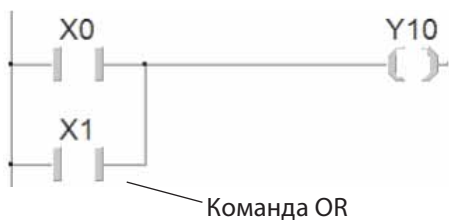
Команда	Назначение	Ladder Diagram	IEC Instruction List
OR	Логическое ИЛИ (операция ИЛИ с опросом состояния сигнала «1» или ВКЛ)		OR
ORI	Логическое ИЛИ НЕ (операция ИЛИ с опросом состояния сигнала «0» или ВЫКЛ)		ORN

Логика операции ИЛИ аналогична параллельному подключению нескольких выключателей в электрической цепи. Как только замыкается один из выключателей, начинает течь ток. Ток прекращает течь только тогда, когда размыкаются все выключатели.



Пример команды OR

Ladder Diagram



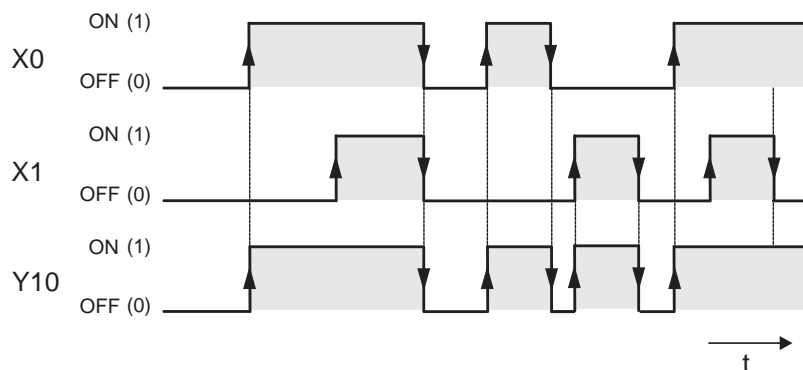
MELSEC Instruction List

```
LD    X0
OR    X1
OUT   Y10
```

IEC Instruction List

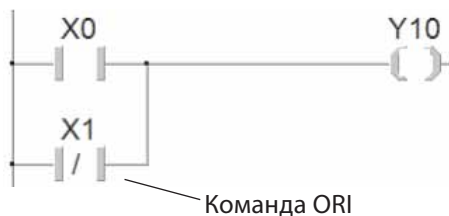
```
LD    X0
OR    X1
ST    Y10
```

В данном примере выход Y10 включается, когда включен либо вход X0 либо вход X1.



Пример команды ORI

Ladder Diagram



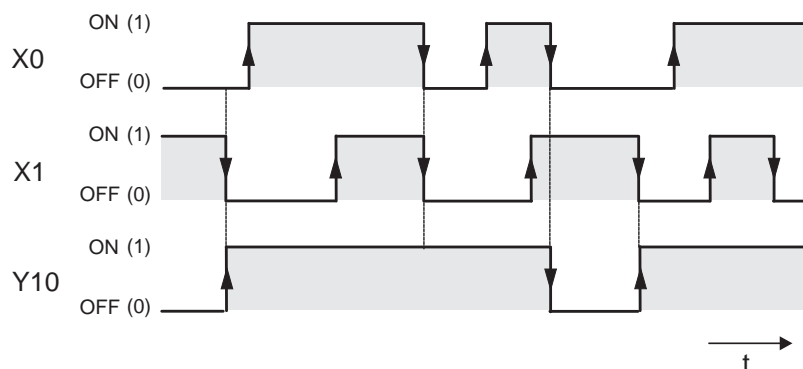
MELSEC Instruction List

```
LD    X0
ORI   X1
OUT   Y10
```

IEC Instruction List

```
LD    X0
ORN   X1
ST    Y10
```

В данном примере выход Y10 включается, когда либо включен вход X0 либо выключен вход X1.



4.7.6 Команды подключения блоков операций

Команда	Назначение	Ladder Diagram	IEC Instruction List
ANB	Блок И (последовательное подключение блоков параллельных операций/схем)	—	AND (...)
ORB	Блок ИЛИ (параллельное подключение блоков последовательных операций/схем)		OR (...)

Несмотря на то, что ANB и ORB являются командами ПЛК, в окне Ladder Diagram они просто отображаются и вводятся как соединительные линии. Как команды они отображаются только в формате Instruction List, где их можно вводить с помощью акронимов ANB и ORB.

Обе команды вводятся без указания устройств и могут использоваться в программе столько, сколько нужно. Но максимальное число команд LD и LDI составляет 15, что также ограничивает до 15 число команд ORB и ANB, которые можно использовать перед командой вывода.

Пример команды ANB

Ladder Diagram



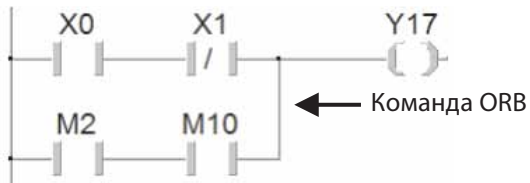
MELSEC Instruction List

```
LD      X0
ORI     M2   ← 1-ое параллельное подключение (операция OR)
LD      X1
OR      M10  ← 2-ое параллельное подключение (операция OR)
ANB
OUT     Y17
```

IEC Instruction List

```
LD      X0
ORN     M2   ← 1-ое параллельное подключение (операция OR)
AND(
      X1   ← Команда ANB подключает обе операции OR
OR      M10 ← 2-ое параллельное подключение (операция OR)
)
ST      Y017
```

В данном примере выход Y17 включается, когда вход X00 – «1» и вход X01 – «0», или когда реле M2 – «0» и реле M10 – «1».

Пример команды ORBLadder DiagramMELSEC Instruction List

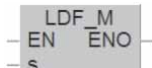
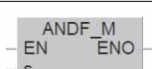
LD	X0	
ANI	X1	← 1-ое последовательное подключение (операция AND)
LD	M2	
AND	M10	← 2-ое последовательное подключение (операция AND)
ORB		← Команда ORB подключает обе операции AND
OUT	Y17	

IEC Instruction List

LD	X0	
ANDN	X1	← 1-ое последовательное подключение (операция AND)
OR(← Команда ORB подключает обе операции AND
	M2	
AND	M10	← 2-ое последовательное подключение (операция AND)
)		
ST	Y17	

В данном примере выход Y17 включается, когда вход X00 – «1» и вход X01 – «0», или когда реле M2 – «0» и реле M10 – «1».

4.7.7 Импульсный запуск выполнения операций

Команда	Назначение	Ladder Diagram	IEC Instruction List
LDP	Нагрузка при импульсе: нагрузки при нарастающем фронте сигнала устройства		—
LDF	Нагрузка при падающем импульсе: нагрузки при падающем фронте сигнала устройства		—
ANDP	И при импульсе: операция логического И при нарастающем фронте сигнала устройства		ANDP_M
ANDF	И при падающем импульсе: операция логического И при падающем фронте сигнала устройства		ANDF_M
ORP	ИЛИ при импульсе: операция логического ИЛИ при нарастающем фронте сигнала устройства		ORP_M
ORF	ИЛИ при падающем импульсе: операция логического ИЛИ при падающем фронте сигнала устройства		ORF_M

В программах ПЛК зачастую необходимо обнаруживать и реагировать на нарастающий или падающий фронт коммутирующего сигнала битового устройства. Нарастающий фронт означает переключение устройства с «0» на «1», а падающий фронт – переключение с «1» на «0».

В процессе выполнения программы операции, реагирующие на нарастающие и падающие импульсы, выдают значение 1 только при изменении состояния сигнала соответствующего устройства.

Когда это требуется использовать? Предположим, к примеру, что имеется ленточный конвейер с сенсорным выключателем, активирующим увеличение показания счётчика, когда по конвейеру проходит упаковка. Если не использовать функцию импульсного запуска выполнения, будет получен неверный результат, поскольку показания счётчика будут увеличиваться на 1 в каждом программном цикле, в котором данный выключатель регистрируется как установленный. Если регистрировать только нарастающий импульс сигнала выключателя, показания счётчика будут увеличиваться правильно – на 1 для каждой упаковки.


ПРИМЕЧАНИЕ

Большинство прикладных команд также может выполняться по импульсным сигналам. Подробные сведения приведены в разделе 6.

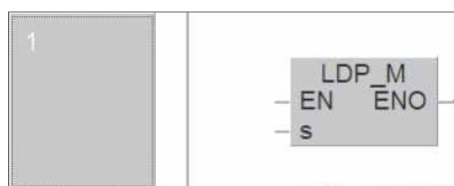
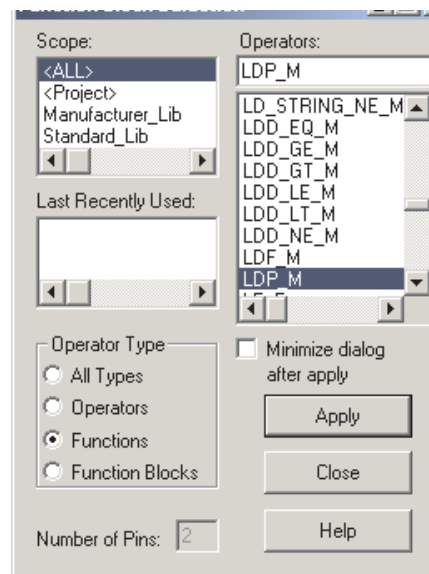
Ввод функций и функциональных блоков в Ladder Diagram

Команды с импульсным запуском и другие сложные команды невозможно вводить с помощью кнопок на панели инструментов GX IEC Developer. Такие команды вводятся путем их выбора в окне выбора функционального блока.

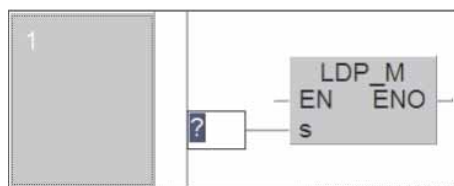



Нажмите кнопку выбора функции/функционального блока  на панели инструментов. Откроется показанное ниже окно выбора функционального блока.

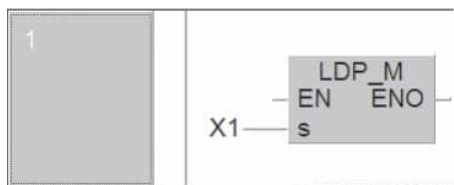
В области Operator type (Тип оператора) щёлкните Functions (Функции) и в списке выберите команду, например LDP_M.



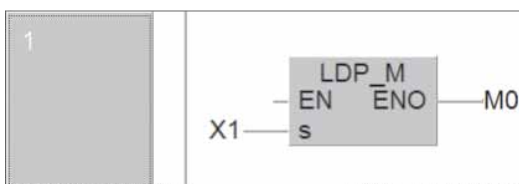
Щёлкните Apply (Применить) или дважды щёлкните выбранный объект, затем щёлкните в теле программного модуля, чтобы вставить данную функцию.




Щёлкните кнопку  (Входная переменная) на панели инструментов, затем щёлкните на вход функции, где нужно ввести устройство.



Введите входное устройство и нажмите клавишу ENTER.



Чтобы ввести переменную в выход функции, щёлкните кнопку  на панели инструментов, затем щёлкните на выход ENO.

Оценка импульса нарастающего сигнала

Ladder Diagram

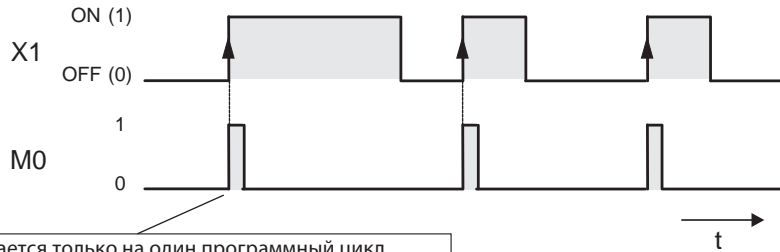


MELSEC Instruction List

LDP X1
OUT M0

IEC Instruction List

LD X1
PLS_M M0



Реле M0 включается только на один программный цикл

Оценка импульса падающего сигнала

Ladder Diagram

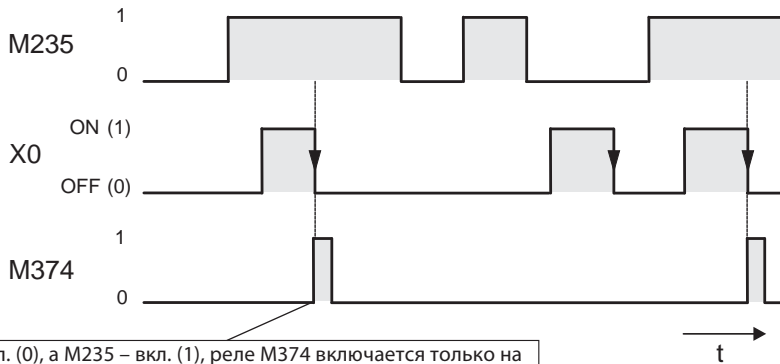


MELSEC Instruction List

LD M235
ANDF X0
OUT M374

IEC Instruction List

LD M235
ANDF_M X0
ST M374



Когда X0 – выкл. (0), а M235 – вкл. (1), реле M374 включается только на один программный цикл

За исключением свойства импульсного запуска, функции команд LDP, LDF, ANDP, ANDF, ORP и ORF аналогичны функциям команд LD, AND и OR. То есть операции с импульсным запуском в программах используются точно так же, как обычные операции.

4.7.8 Установка и сброс устройств

Команда	Назначение	Ladder Diagram	IEC Instruction List
SET	Установка устройства ^① , (назначение состояния сигнала «1»)		S
RST	Сброс устройства ^② , (назначение состояния сигнала «0»)		R

① Команду SET можно использовать для установки выходов (Y), реле (M) и реле состояний (S).

② Команду RST можно использовать для сброса выходов (Y), реле (M), реле состояний (S), таймеров (T), счётчиков (C) и регистров (D, V, Z).

Состояние сигнала команды OUT обычно остается равным «1», только пока результат операции, связанной с данной командой, составляет «1». Например, если подключить кнопку к входу, а лампу к соответствующему выходу и связать их командами LD и OUT, то лампа будет оставаться включенной, пока кнопка остается в нажатом состоянии.

Команду SET можно использовать в качестве кратковременного коммутирующего импульса для включения (установки) выхода или реле, оставляя при этом их во включенном состоянии. Устройство будет оставаться включенным до его выключения (сброса) командой RST. Это дает возможность использовать функции запираения или включать и отключать приводы с помощью кнопок. (При отключении ПЛК или выключении питания выходы обычно также отключаются. Но некоторые реле в данных условиях сохраняют своё последнее состояние сигнала, например, реле установки остается установленным.)

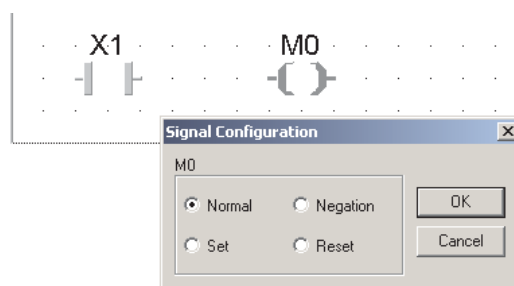
В формате Ladder Diagram команды SET и RST можно программировать в рамках операции вывода или в качестве функции.

Команда OUT с функцией SET или RST

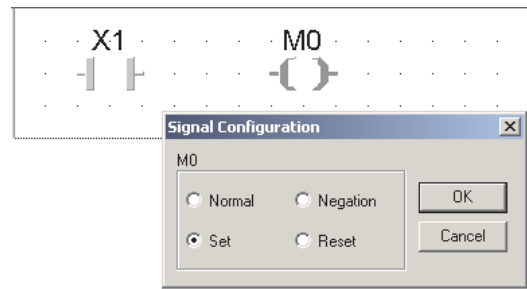
Задайте команду OUT и введите устройство для установки или сброса.



Дважды щёлкните команду OUT. Появится окно Signal configuration (Конфигурация сигнала).



Для использования команды SET щёлкните Set (Установить). Для использования команды RST щёлкните Reset (Сбросить). Затем щёлкните OK, чтобы закрыть окно.



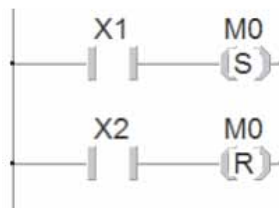
Это приводит к преобразованию команды OUT в команду SET.



Примеры установки и сброса устройств

Ladder Diagram

1-ый вариант



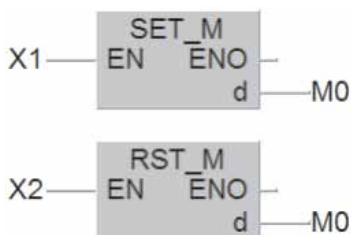
MELSEC Instruction List

```
LD X1
SET M0
LD X2
RST M0
```

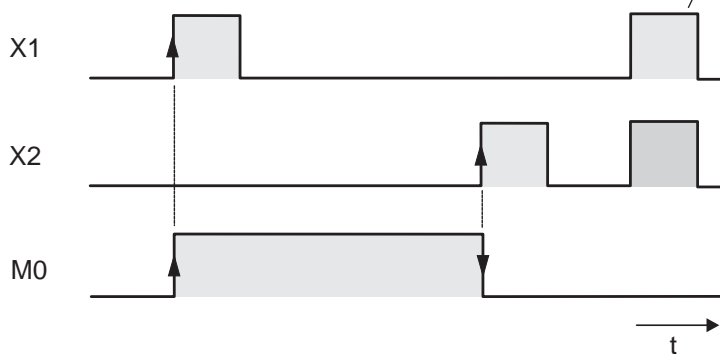
IEC Instruction List

```
LD X1
S M0
LD X2
R M0
```

2-ой вариант

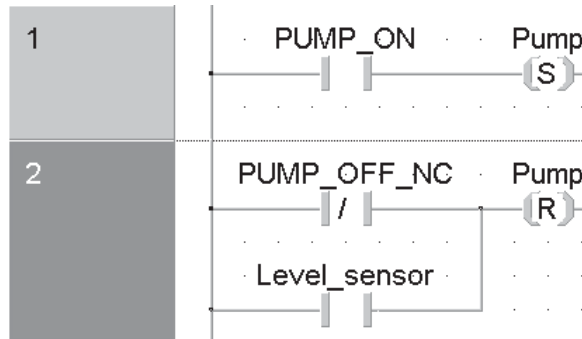


Если при оценке команд установки и сброса для одного устройства в обоих случаях получается «1», приоритет имеет операция, выполнявшаяся последней. В данном примере это команда RST, поэтому M0 остается выключенным.



Данный пример представляет программу управления насосом, заполняющим некоторую ёмкость. Управление насосом осуществляется вручную, с помощью кнопок ВКЛ и ВЫКЛ. По соображениям безопасности для функции ВЫКЛ используется размыкающий контакт. Когда ёмкость заполняется, датчик уровня автоматически отключает насос.

Ladder Diagram



MELSEC Instruction List

```
LD    Pump_ON
SET   Pump
LDI   Pump_OFF_NC
OR    Level_sensor
RST   Pump
```

IEC Instruction List

```
LD    Pump_ON
S     Pump
LDN   Pump_OFF_NC
OR    Level_sensor
R     Pump
```

ПРИМЕЧАНИЕ

Для отображения устройств вместе с их идентификаторами необходимо объявлять их как переменные в списке глобальных переменных. Ниже представлен список глобальных переменных для данного примера программы.

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type
0	VAR_GLOBAL	PUMP_ON	X1	%IX1	BOOL
1	VAR_GLOBAL	PUMP_OFF_NC	X2	%IX2	BOOL
2	VAR_GLOBAL	Level_sensor	X3	%IX3	BOOL
3	VAR_GLOBAL	Pump	Y10	%QX16	BOOL

Дополнительная информация о списке глобальных переменных приведена в разделе 4.6.2.

4.7.9 Формирование импульсов

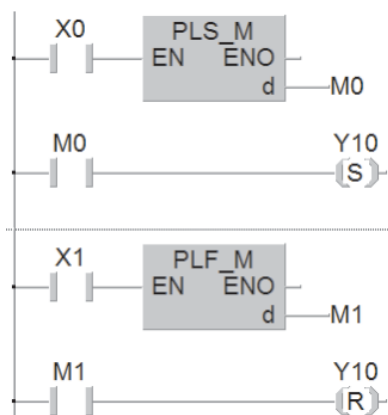
Команда	Назначение	Ladder Diagram	IEC Instruction List
PLS	Импульс: установка устройства* на один программный цикл при нарастающем фронте коммутирующего импульса условия на входе/устройства		PLS_M
PLF	Падающий импульс: установка устройства* на один программный цикл при падающем фронте коммутирующего импульса условия на входе/устройства		PLF_M

* Команды PLS и PLF можно использовать для установки выходов (Y) и реле (M).

Данные команды эффективно преобразовывают статический сигнал в кратковременный импульс, длительность которого зависит от длины программного цикла. При использовании команды PLS вместо OUT состояние сигнала указанного устройства устанавливается на «1» только на один программный цикл, в частности во время цикла, в котором состояние сигнала устройства перед командой PLS в цепи переключается из состояния «0» в «1» (импульс с нарастающим фронтом).

Команды PLF реагирует на импульс с падающим фронтом, устанавливая указанное устройство в «1» на один программный цикл во время цикла, в котором состояние сигнала устройства перед командой PLF в цепи переключается из состояния «1» в «0» (импульс с падающим фронтом).

Ladder Diagram

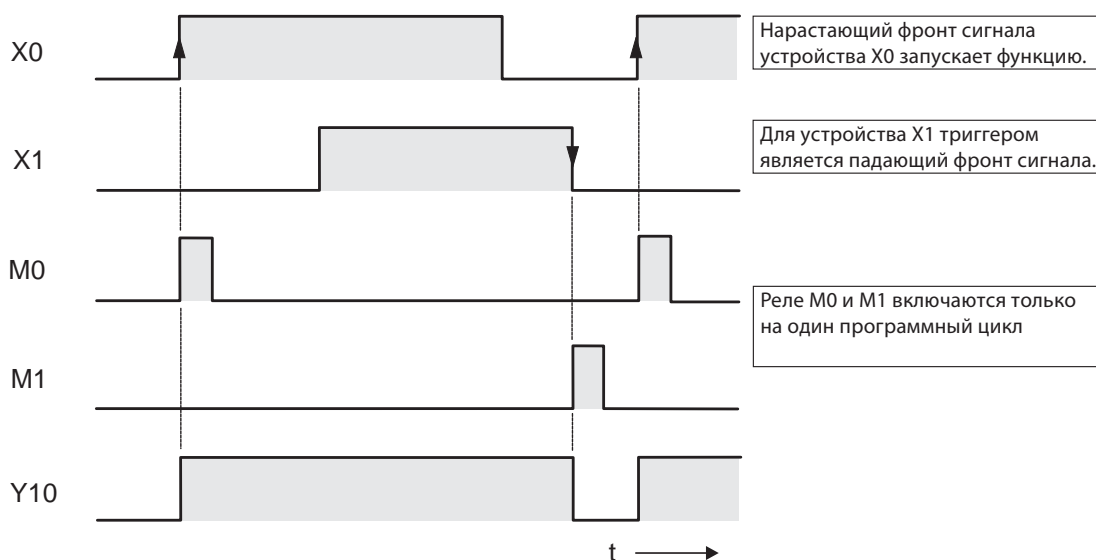


MELSEC Instruction List

```
LD X0
PLS M0
LD M0
SET Y10
LD X1
PLF M1
LD M1
RST Y10
```

IEC Instruction List

```
LD X0
PLS_M M0
LD M0
S Y10
LD X1
PLF_M M1
LD M1
R Y10
```



4.7.10 Инверсия результата операции

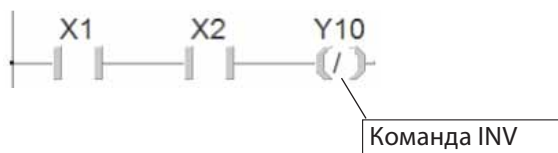
Команда	Назначение	Ladder Diagram	IEC Instruction List
INV	Инверсия: изменение результата операции на противоположный		NOT

Команда INV применяется без использования каких-либо операндов. Она инвертирует результат операции, выполненной непосредственно перед ней.

- Если результат операции «1», он инвертируется в «0».
- Если результат операции «0», он инвертируется в «1».

Ladder Diagram

1-ый вариант



MELSEC Instruction List

```
LD X1
AND X2
INV
OUT Y10
```

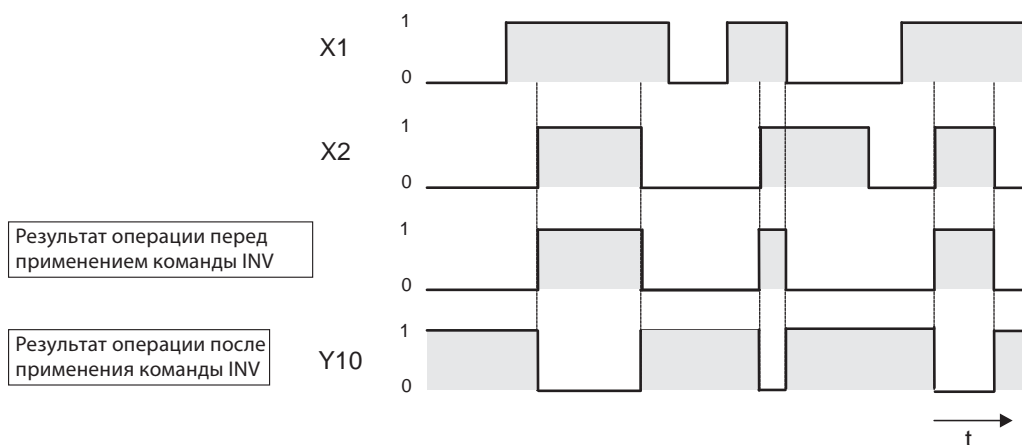
2-ой вариант



IEC Instruction List

```
LD X1
AND X2
NOT
ST Y10
```

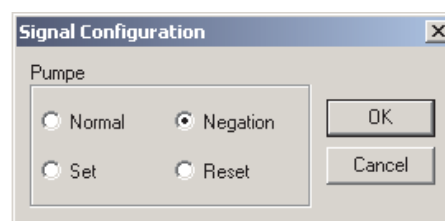
Результатом вышеуказанных примеров является следующая последовательность сигнала.




Команду INV можно использовать для инверсии результата сложной операции. Она используется в той же позиции, что и команды AND и ANI.

ПРИМЕЧАНИЕ

Чтобы запрограммировать команду INV в формате Ladder Diagram в рамках команды OUT, дважды щёлкните команду OUT – появится окно Signal Configuration (Конфигурация сигнала). Щёлкните Negation (Отрицание) и подтвердите выбор нажатием кнопки OK (см. также раздел).



4.7.11 Инверсия битового выходного устройства

Команда	Назначение	Ladder Diagram	IEC Instruction List
Команда FF	Инверсия битового выходного устройства		FF_MD

* Команду FF можно использовать для установки выходов (Y), реле (M) и отдельных разрядов устройств типа «слово».

Команда FF инвертирует состояние операции устройства, обозначенного на выходе с нарастающим фронтом на входе команды FF.

- Если состояние выходного устройства установлено (1), после инверсии оно сбрасывается (0).
- Если состояние выходного устройства сброшено (0), после инверсии оно устанавливается (1).

Ladder Diagram



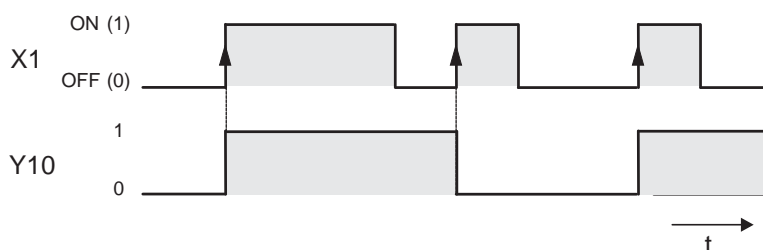
MELSEC Instruction List

```
LD    X1
FF    Y10
```

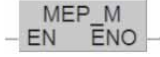
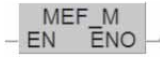
IEC Instruction List

```
LD    X1
FF_MD Y10
```

Приведенная выше программа инвертирует выходное состояние Y10 с нарастающим фронтом с входа X1.



4.7.12 Преобразование результата операции в импульс

Команда	Назначение	Ladder Diagram	IEC Instruction List
MEP	Формирование импульса при нарастающем фронте результата операции		MEP_M
MEF	Формирование импульса при падающем фронте результата операции		MEF_M

Команды MEP и MEF используются без указания устройств. Они формируют один выходной импульс нарастающим (или, соответственно, падающим) фронтом входного сигнала, то есть результата операции, который был действительным перед выполнением данных команд. Следующий импульс формируется со следующим нарастающим (или, соответственно, падающим) фронтом.

Ladder Diagram



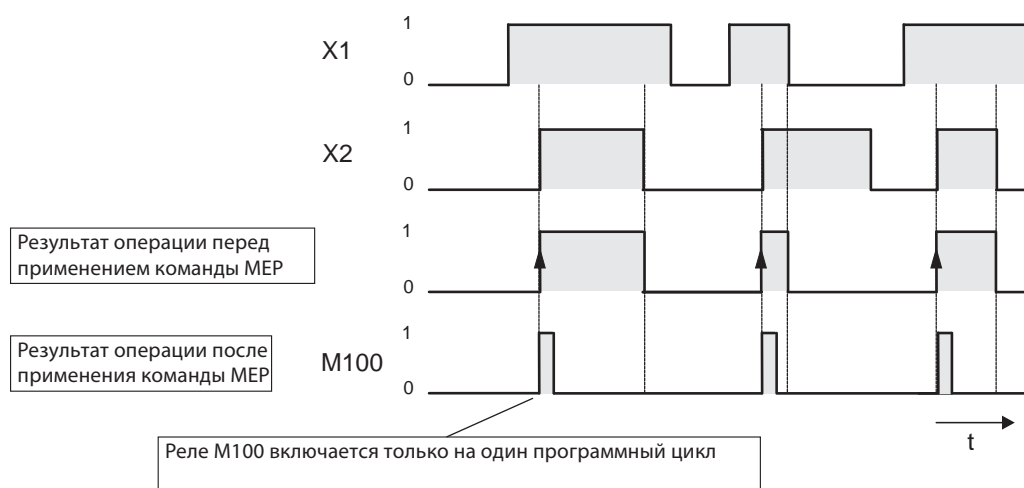
MELSEC Instruction List

```
LD    X1
AND   X2
MEP
OUT   M100
```

IEC Instruction List

```
LD    X1
AND   X2
MEP_M
ST    M100
```

Результатом вышеуказанного примера является следующая последовательность сигнала.



Эти две команды особенно подходят для соединений с несколькими контактами. Например, несколько подключенных последовательно нормально разомкнутых контактов сохраняют результат операции «1», когда все они замкнуты. Если реле устанавливается данным результатом операции, то оно не сбрасывается. При использовании подключенной последовательно команды MEP с данными нормально разомкнутыми контактами реле сбрасывается, поскольку команда приводит к выдаче только одного импульса, если результат последовательного подключения всех контактов меняется с «0» на «1».

4.8 Обеспечение безопасности

По сравнению с контроллерами на релейно-контактной логике ПЛК обладает множеством преимуществ. Тем не менее, в вопросах обеспечения безопасности ПЛК нельзя безоговорочно доверять.

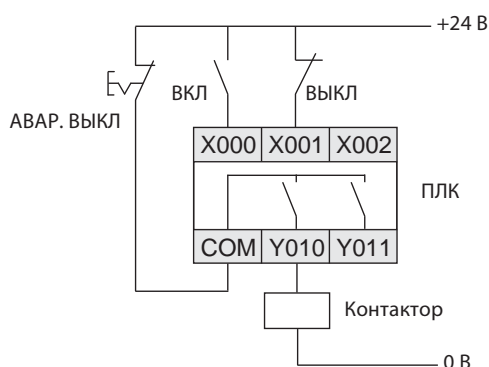
Устройства аварийного останова

Необходимо обеспечить, чтобы в случае ошибок в системе управления или программе исключалась опасность для персонала и оборудования. Устройства аварийного останова должны оставаться полностью работоспособными даже при сбоях в работе ПЛК, например, могли при необходимости отключить питание выходов ПЛК.

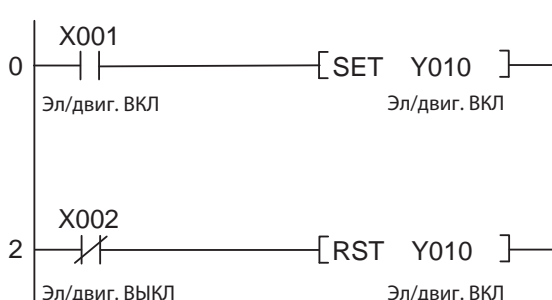
Запрещается реализовывать выключатель аварийного останова исключительно в качестве входа, обрабатываемого ПЛК с использованием программы ПЛК для активации отключения – это крайне опасно.

Меры предосторожности на случай обрыва кабелей

Необходимо также предпринять меры предосторожности на случай прекращения передачи сигнала с выключателей на ПЛК при обрыве кабелей. Для включения и отключения оборудования с помощью ПЛК обязательно используйте выключатели и кнопки с замыкающими контактами на включение и размыкающими на отключение.



В данном примере контактор системы привода можно также выключать вручную с помощью выключателя АВАР. ВЫКЛ.



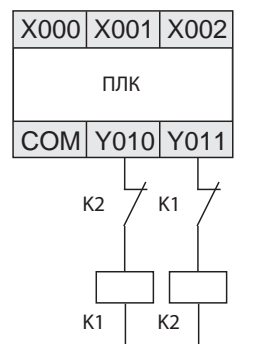
В программе для данной монтажной схемы замыкающий контакт выключателя ВКЛ опрашивается командой LD, а размыкающий контакт выключателя ВЫКЛ – командой LDI. Выход (соответственно, и привод тоже) отключается, когда состояние сигнала входа X002 становится равным «0». Это происходит при задействовании выключателя ВЫКЛ или прерывании соединения между данным выключателем и входом X002.

Это обеспечит автоматическое отключение привода при обрыве кабеля и предотвратит его включение. Кроме того, отключение является приоритетным, поскольку обрабатывается программой после применения команды.

Блокировка контактов

Если имеются два выхода, которые не должны включаться одновременно (например, выходы для выбора прямого и обратного вращения электродвигателя), блокировку для выходов также необходимо реализовать с физическими контактами в контакторе с управлением от ПЛК. Это абсолютно необходимо, так как в программе возможна только внутренняя блокировка и ошибка в ПЛК может привести к одновременной активации обоих выходов.

Пример такой блокировки с контактами контактора показан на рисунке справа. Одновременное включение контакторов K1 и K2 здесь невозможно физически.



Автоматическое отключение

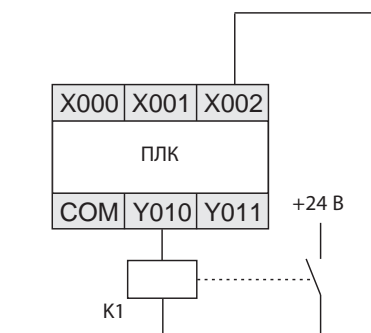
Если ПЛК используется для управления последовательностями движений, когда может возникнуть опасность при перемещении компонентов мимо определенных точек, необходимо установить дополнительные концевые выключатели, автоматически прерывающие движение. Данные выключатели должны подключаться напрямую и не зависеть от ПЛК. Пример такого средства автоматического отключения приведен в разделе 4.9.1.

Обратная связь для выходного сигнала

Как правило, выходы ПЛК не контролируются. При активации выхода в программе предполагается, что вне ПЛК реализуется правильный отклик. В большинстве случаев никаких дополнительных средств не требуется. Однако в критически важных приложениях необходимо контролировать выходные сигналы с помощью ПЛК, например, когда неисправности в выходной цепи (обрыв проводов, залипание контактов и т.д.) могут повлечь за собой значительные последствия для безопасности или работы системы.

В примере справа замыкающий контакт контактора K1 включает вход X002, когда включается выход Y10. Это позволяет программе отслеживать исправность работы выхода и подключенного контактора.

Но в этом простом примере не проверяется исправность работы подключенного оборудования (например, вращение электродвигателя). Для такой проверки потребовались бы дополнительные функции, такие как датчик частоты вращения или устройство контроля коммутируемого напряжения.



4.9 Программирование приложений ПЛК

Программируемые логические контроллеры обеспечивают практически неограниченное количество способов связи входов с выходами. Задача заключается в правильном выборе из множества команд, поддерживаемых контроллерами System Q, чтобы запрограммировать подходящее решение для данного приложения.

В данном разделе приведен простой пример, демонстрирующий создание приложения ПЛК, начиная от постановки задачи и заканчивая готовой программой.

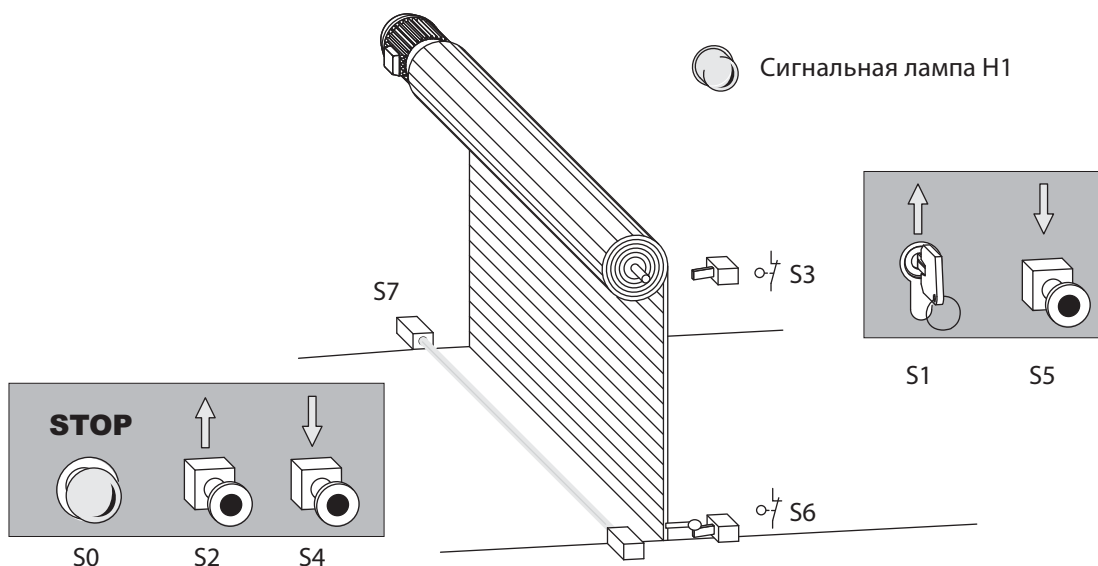
4.9.1 Свёртывающаяся шторная дверь

Вначале необходимо четко определить задачу.

Это значит, что нужно на основе принципа «снизу-вверх» составить четкое описание задачи для ПЛК.

Описание задачи

Требуется реализовать систему управления для свёртывающейся шторной двери склада, позволяющей легко управлять дверью снаружи и изнутри. В данной системе также должны быть предусмотрены средства защиты.



● Принцип работы

- Дверь должна открываться снаружи выключателем S1 с ключом и закрываться кнопкой S5. Изнутри дверь должна открываться кнопкой S4 и закрываться кнопкой S4.
- Дополнительный таймерный выключатель должен автоматически закрывать дверь, когда она остается открытой более 20 с.
- Индикация состояний «дверь движется» и «дверь в неопределенном положении» должна обеспечиваться миганием сигнальной лампы.

- Средства защиты

- Необходимо установить кнопку останова (S0), позволяющей немедленно прекратить движение в любое время, остановив дверь в текущем положении. Данный выключатель типа «Стоп» не является аварийным выключателем. Сигнал с выключателя просто обрабатывается в ПЛК, не переключая никакие подключения к внешнему питанию.
- Необходимо установить фотоэлемент (S7) для определения препятствий на пути движения двери. При обнаружении препятствия при закрывании дверь должна автоматически открываться.
- Необходимо установить два концевых выключателя для останова электродвигателя привода двери в полностью открытом (S3) и полностью закрытом (S6) положениях двери.

Назначение входных и выходных сигналов

Описание задачи четко определяет требуемое количество входов и выходов. Электродвигатель привода двери управляется с помощью двух выходов. Требуемые сигналы назначаются входам и выходам ПЛК следующим образом.

Назначение		Имя	Адрес	Комментарии
Входы	Кнопка СТОП	S0	X0	Размыкающий контакт (когда выключатель срабатывает X0 = «0» и дверь останавливается)
	Выключатель ОТКРЫТЬ с ключом (снаружи)	S1	X1	Замыкающие контакты
	Кнопка ОТКРЫТЬ (внутри)	S2	X2	
	Верхний концевой выключатель (открывание двери)	S3	X3	Размыкающий контакт (X2 = «0», когда дверь вверх и активируется S3)
	Кнопка ЗАКРЫТЬ (внутри)	S4	X4	Замыкающие контакты
	Кнопка ЗАКРЫТЬ (снаружи)	S5	X5	
	Нижний концевой выключатель (закрывание двери)	S6	X6	Размыкающий контакт (X6 = «0», когда дверь вниз и активируется S6)
	Фотоэлемент	S7	X7	При обнаружении препятствия X7 устанавливается на «1»
Выходы	Сигнальная лампа	H1	Y10	—
	Контактор электродвигателя (обратное вращение эл/двиг.)	K1	Y11	Обратное вращение – ОТКРЫВАНИЕ двери
	Контактор электродвигателя (прямое вращение эл/двиг.)	K2	Y12	Прямое вращение – ЗАКРЫВАНИЕ двери
Таймер	Задержка автоматического закрывания	—	T0	Время: 20 секунд

4.9.2 Программирование

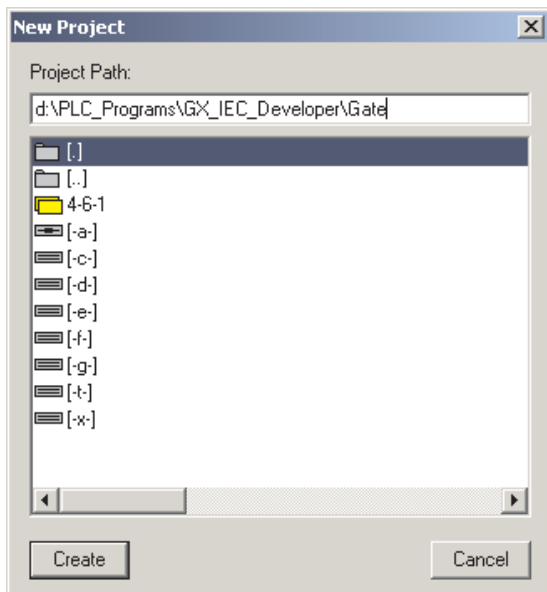
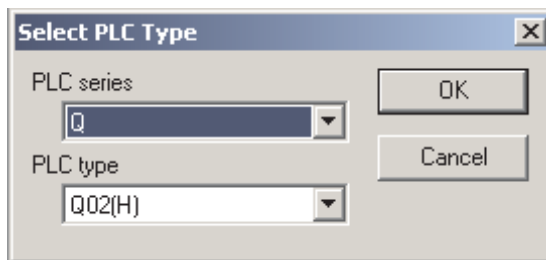
Создание проекта

После запуска GX IEC Developer в меню Project (Проект) выберите New (Создать).



Выберите в списке соответствующий тип ПЛК.

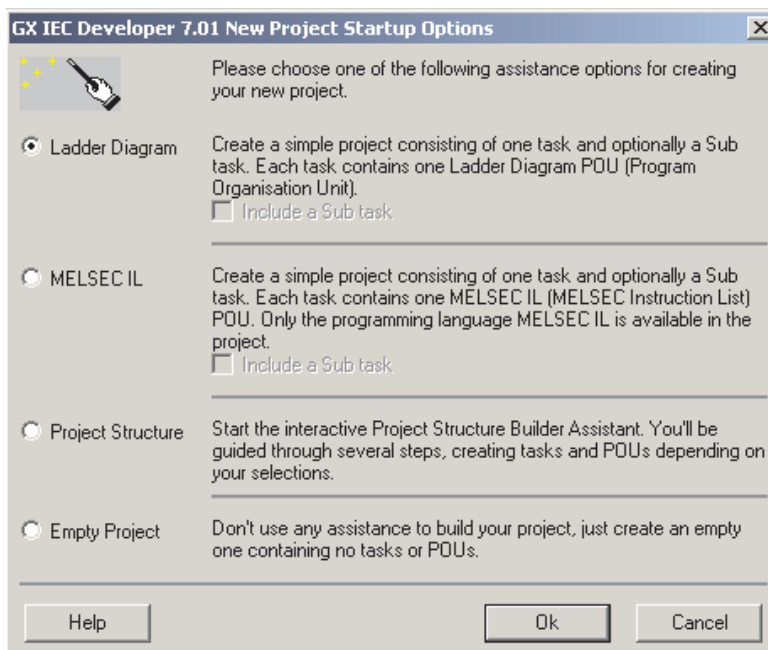
Для подтверждения выбора щёлкните ОК.



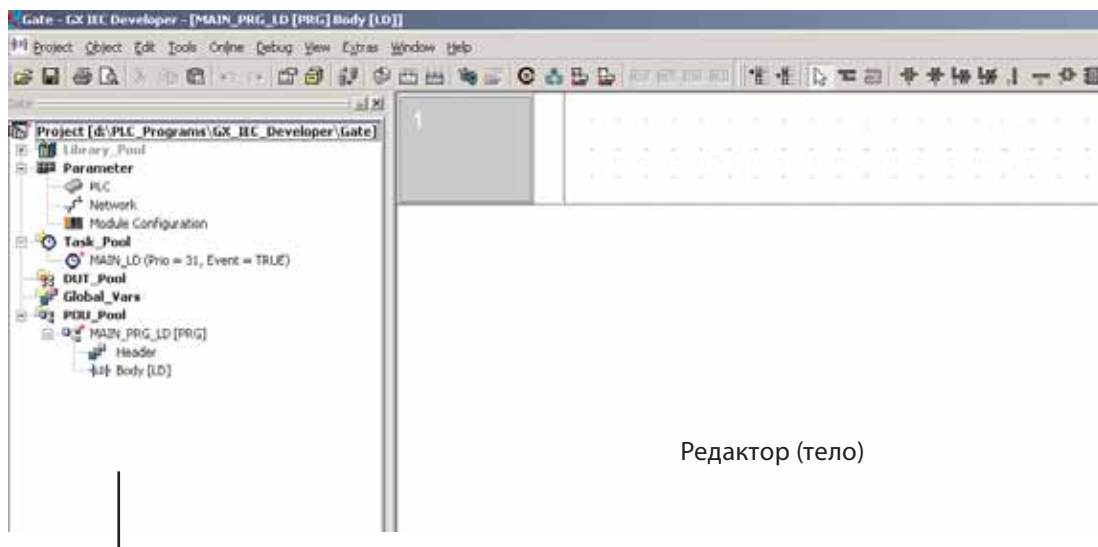
Появится диалоговое окно New Project (Новый проект). Выберите или введите путь для сохранения нового проекта. В конце пути введите также имя для нового проекта.

После нажатия кнопки Create (Создать) в GX IEC Developer будет создан подкаталог с указанным именем нового проекта.

Выберите Startup Options (Опции запуска). В данном примере выбран формат Ladder Diagram.



Подтвердив выбор нажатием кнопки ОК, можно начинать программирование. Экран отображения проекта с пустым телом программного модуля (POU) MAIN показан на следующей странице.



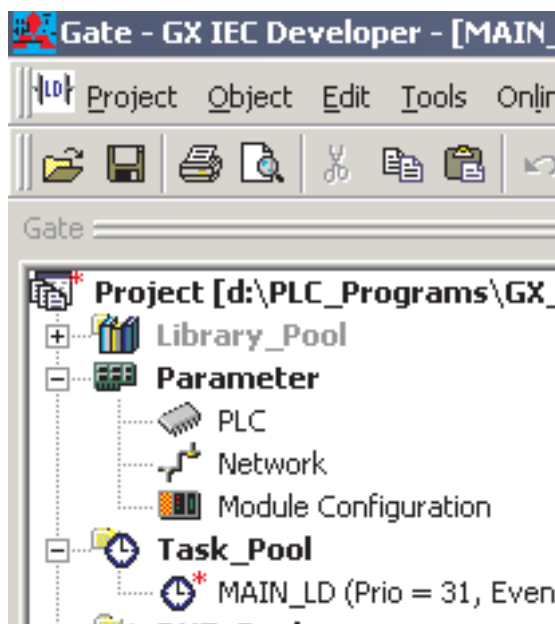
Редактор (тело)

Окно навигатора проекта

Назначение глобальных переменных

ПРИМЕЧАНИЕ

Если в программе не используются символические идентификаторы, а только адреса формата «Мицубиси», то список глобальных переменных заполнять не требуется. Но при этом программа не будет совместимой со стандартом IEC61131-3.



Дважды щёлкните Global_Vars (Глобальные переменные) в окне навигатора проекта.

Откроется список глобальных переменных (GVL).

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type
0	VAR_GLOBAL				

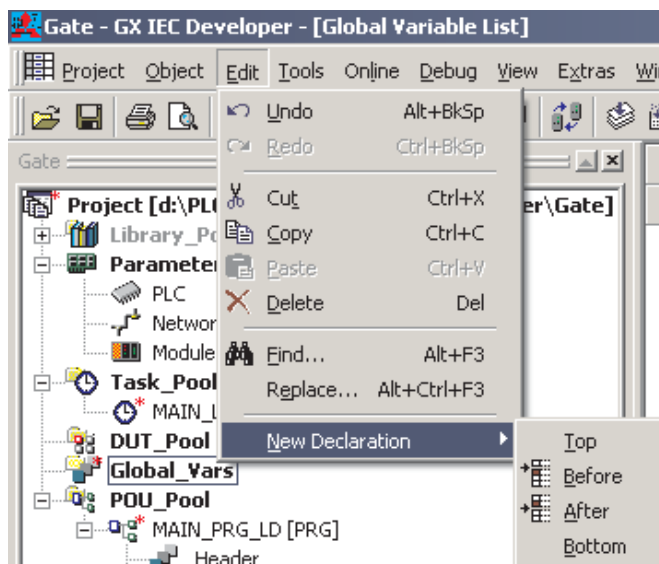
Введите идентификатор и абсолютный адрес первой глобальной переменной. Не требуется вводить оба адреса: в формате «Мицубиси» и IEC. При вводе одного адреса в GX IEC Developer автоматически добавляется другой.

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial	
0	VAR_GLOBAL	S0_STOP_Gate	X0	%IX0	BOOL	FALSE	Norm

При вводе физического входного адреса в качестве типа автоматически выбирается BOOL.

Для объявления остальных переменных список необходимо раскрыть. Это можно сделать одним из следующих способов:

- Если курсор активен в одном из столбцов строки последнего объявления переменной, нужно одновременно нажать клавиши SHIFT и ENTER.
- Можно в меню Edit (Правка) выбрать пункт New Declaration (Новое объявление).



- Можно щёлкнуть кнопку «Вставить перед» или «Вставить после» на панели инструментов.



Для данного проекта в качестве глобальных переменных указаны следующие входы и выходы.

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial	Comment
0	VAR_GLOBAL	S0_STOP_GATE	X0	%IX0	BOOL	FALSE	Normally closed
1	VAR_GLOBAL	S1_OPEN_GATE_Switch	X1	%IX1	BOOL	FALSE	
2	VAR_GLOBAL	S2_OPEN_GATE_PB	X2	%IX2	BOOL	FALSE	
3	VAR_GLOBAL	S3_Upper_limit_switch	X3	%IX3	BOOL	FALSE	Normally closed
4	VAR_GLOBAL	S4_CLOSE_GATE_PB	X4	%IX4	BOOL	FALSE	Inside
5	VAR_GLOBAL	S5_CLOSE_GATE_PB	X5	%IX5	BOOL	FALSE	Outside
6	VAR_GLOBAL	S6_Lower_limit_switch	X6	%IX6	BOOL	FALSE	Normally closed
7	VAR_GLOBAL	S7_Photoelectric_barrier	X7	%IX7	BOOL	FALSE	
8	VAR_GLOBAL	H1_Warning_lamp	Y10	%QX16	BOOL	FALSE	
9	VAR_GLOBAL	K1_Motor_open_gate	Y11	%QX17	BOOL	FALSE	
10	VAR_GLOBAL	K2_Motor_close_gate	Y12	%QX18	BOOL	FALSE	

Ввод программы

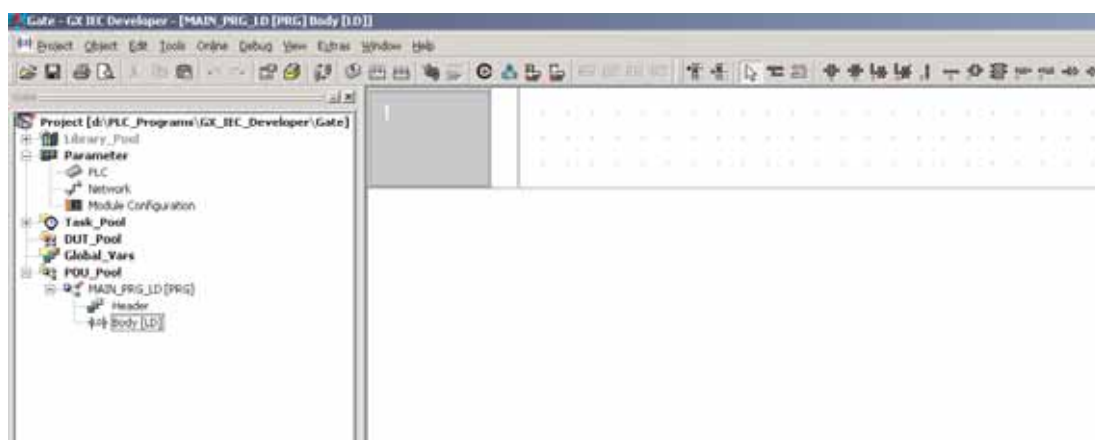
Затем можно приступить к программированию отдельных задач управления.

- Управление свёртывающейся шторной дверью с помощью кнопок

Программа должна преобразовывать входные сигналы для работы двери в две команды управления электродвигателем привода: «Открыть дверь» и «Заккрыть дверь». Поскольку это сигналы с кнопок, появляющиеся на входах только кратковременно, их необходимо сохранять. Для этого используем две переменные, представляющие в программе входы и их установку и сброс по необходимости:

- OPEN_GATE
- CLOSE_GATE

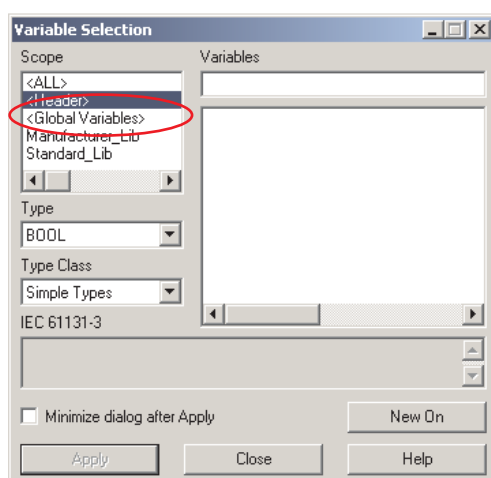
Если тело программного модуля (POU) MAIN не отображается, дважды щёлкните Body [LD] в навигаторе проекта.



Выберите на панели инструментов «нормально разомкнутый» контакт.

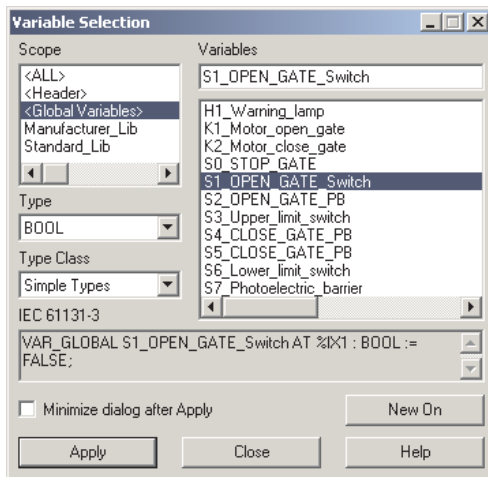


Переместите курсор в рабочую область и щёлкните, чтобы зафиксировать в окне позицию вставки.



Щёлкните правой кнопкой по вопросительному знаку, чтобы вызвать окно Variables Selection (Выбор переменных).

В диалоговой области Scope (Область) щёлкните пункт Global Variables (Глобальные переменные).

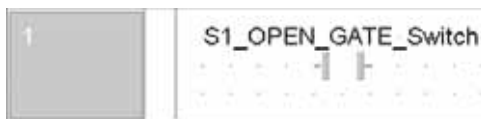


Для выделения переменной щёлкните по ней (в данном случае S1_OPEN_GATE_Switch).

Выбранная таким образом переменная вводится нажатием кнопки Apply (Применить) или двойным щелчком по самой переменной.



Переменная введена.



Щёлкните редактор, чтобы отобразить полный идентификатор переменной.

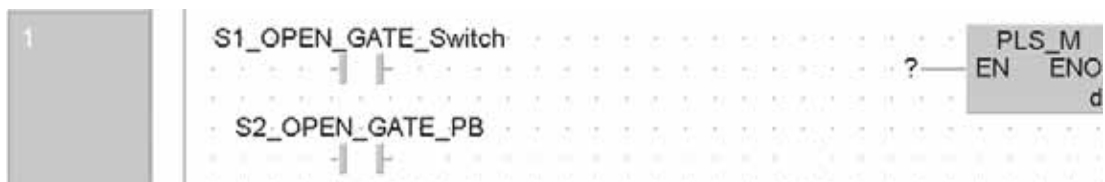


Чтобы расширить сеть, переместите курсор к нижней границе заголовка сети, щёлкните и удерживайте кнопку мыши, одновременно перемещая курсор, чтобы увеличить размер по вертикали.



Введите также кнопку для открывания двери.

Любая активация данных выключателей должна преобразовываться в импульсы. Для этого используется функция PLS_M. Описание ввода программы в формате Ladder Diagram приведено в разделе 4.7.7.



Щёлкните кнопку  (Выходная переменная) на панели инструментов.



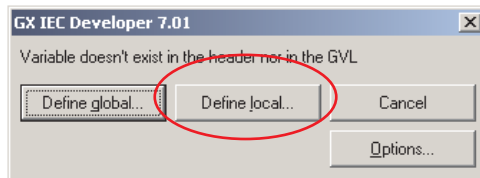
Затем щёлкните выход функции PLS_M для отображения поля сообщения переменной.

- Назначение локальных переменных

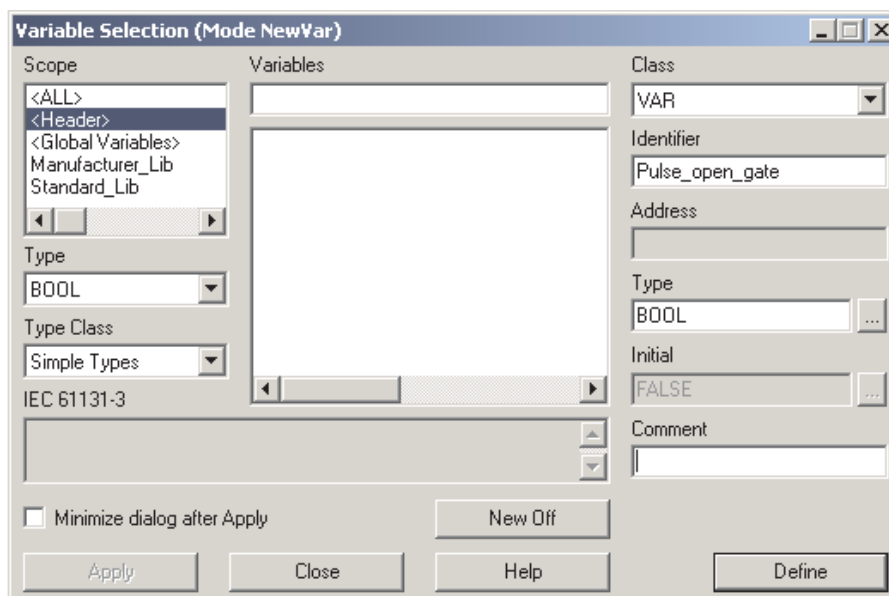
В данном программном модуле используется только выходная переменная функции PLS_M. Поэтому она может быть локальной переменной. В данном проекте локальные переменные пока не назначались, так как это можно делать в процессе программирования.

В пустое поле «?» введите имя переменной «Pulse_open_gate».

Появится следующее сообщение, поскольку данная переменная пока не существует ни в списке локальных переменных, ни в списке глобальных переменных.



Щёлкните кнопку Define local (Определить локальную). Появится окно Variable Selection (Выбор переменной) для определения новой переменной.



Щёлкните кнопку Define (Определить) для ввода новой переменной в список локальных переменных (локальный заголовок программного модуля).

В заключение готовая сеть объединяется путем соединения элементов.

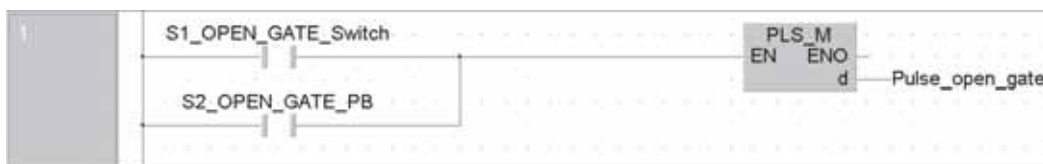


Для этого на панели инструментов предусмотрен значок «Режим линии». При этом указатель становится в виде карандаша.

Щёлкните по сборной шине слева от схемы, затем щёлкните и перетащите курсор по схеме и отпустите на контакте. В данной точке отпустите левую кнопку мыши.



Таким же образом соедините все остальные элементы данной сети.

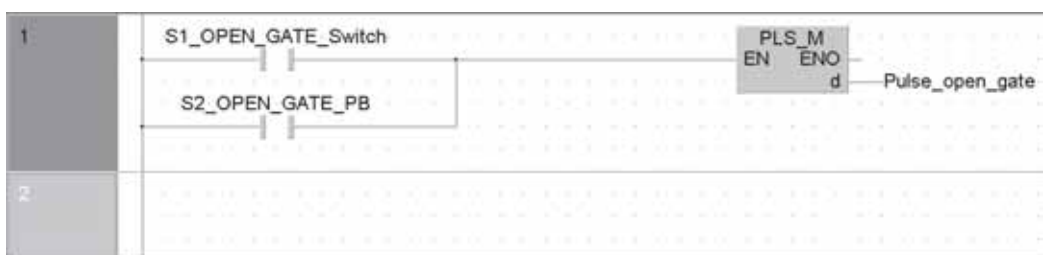


– Создание сети программы

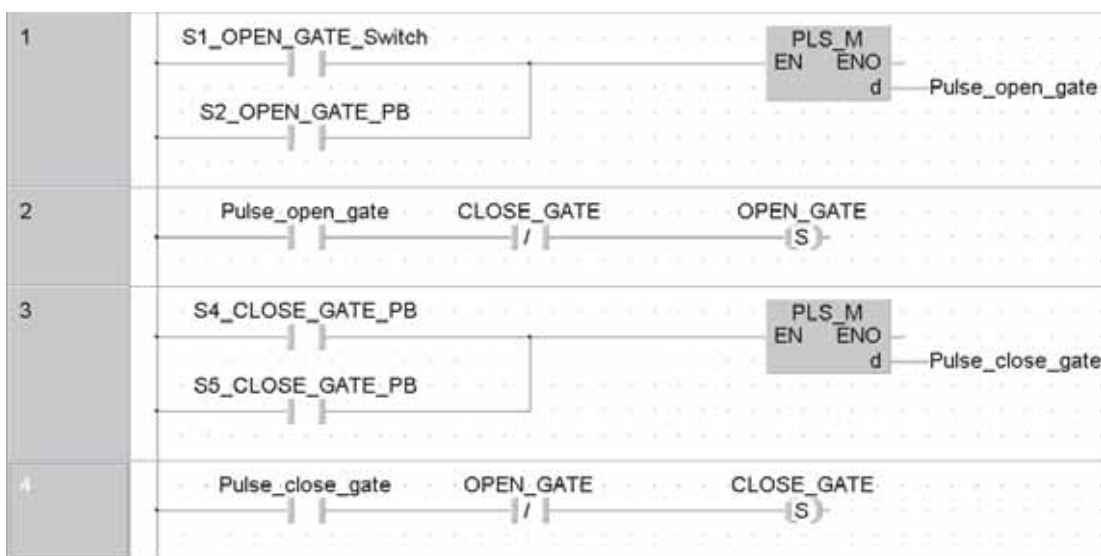
Чтобы создать сеть ниже текущей сети, на панели инструментов щёлкните данную кнопку.



Появится пустая область сети.



Введите в эту и следующие сети нижеуказанные элементы.



Все переменные, кроме выключателей и кнопок, являются локальными. Это еще раз подтверждает преимущество использования переменных с идентификаторами. Даже без ввода информации об устройстве в такой программе легче разобраться, чем в программе с абсолютными адресами (например, X1, X2 и т.д.).

● Описание сетей 1–4

Сначала обрабатываются сигналы для открытия двери. При активации выключателя S1 с ключом или кнопки S2 генерируется сигнал и состояние сигнала для переменной «Pulse_open_gate» устанавливается на «1» только на один программный цикл. Это предотвращает блокировку двери в случае, если кнопка залипнет или оператор её не отпустит. Аналогичный принцип используется для обработки сигналов для закрывания двери с кнопок S4 и S5. Необходимо обеспечить, чтобы привод включался только тогда, когда он не возвращается в противоположном направлении. Это реализуется путем программирования ПЛК таким образом, чтобы дверь закрывалась только тогда, когда она не открывается, и наоборот.

ПРИМЕЧАНИЕ

Блокировку направления вращения электродвигателя следует также дополнить еще одной блокировкой с физическими контакторами вне ПЛК (электрическая схема приведена в разделе 4.9.3).

- Автоматическое закрывание двери через 20 секунд



Если дверь открыта, активируется концевой выключатель S3 и вход X3 отключается (по соображениям безопасности в S3 используется размыкающий контакт). После этого таймер T0 начинает отсчет 20-секундной задержки ($K200 = 200 \times 0,1 \text{ с} = 20 \text{ с}$). Через 20 с по таймеру дверь закрывается.

ПРИМЕЧАНИЕ

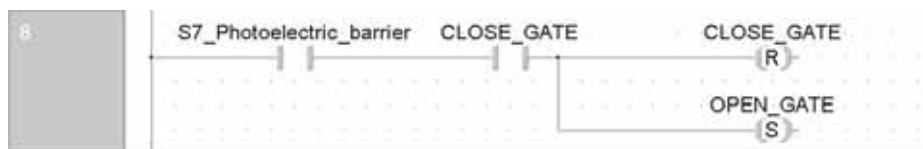
Подробное описание таймера приведено в следующем разделе.

- Останов двери выключателем СТОП



При нажатии кнопки СТОП сбрасываются две локальные переменные и электродвигатель привода останавливается.

- Обнаружение препятствий фотоэлементом



При обнаружении фотоэлементом препятствия при закрывании двери движение останавливается и дверь снова открывается.

- Отключение электродвигателя концевыми выключателями



Если дверь открыта, активируется концевой выключатель S3 и вход X3 отключается. При этом сбрасывается локальная переменная OPEN_GATE и электродвигатель останавливается.

Если дверь полностью закрыта, активируется S6 и электродвигатель также останавливается. По соображениям безопасности в концевых выключателях используются размыкающие контакты. Это также обеспечивает автоматическое отключение электродвигателя (или предотвращение его включения), когда прерывается связь между выключателем и входом.

ПРИМЕЧАНИЕ

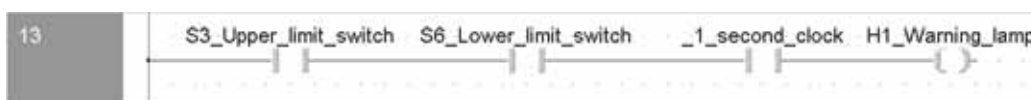
Концевые выключатели должны подключаться так, чтобы дополнительно обеспечивать отключение электродвигателя без поддержки со стороны ПЛК (электрическая схема приведена в разделе 4.9.3).

● Управление электродвигателем

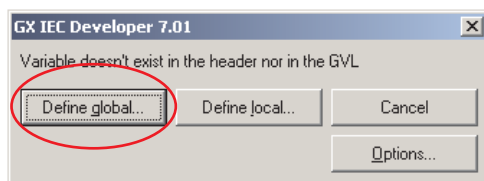


В конце программы состояния сигналов локальных переменных для открывания и закрывания передаются на выходы Y11 и Y12.

● Сигнальная лампа: «дверь движется» и «дверь в неопределенном положении»

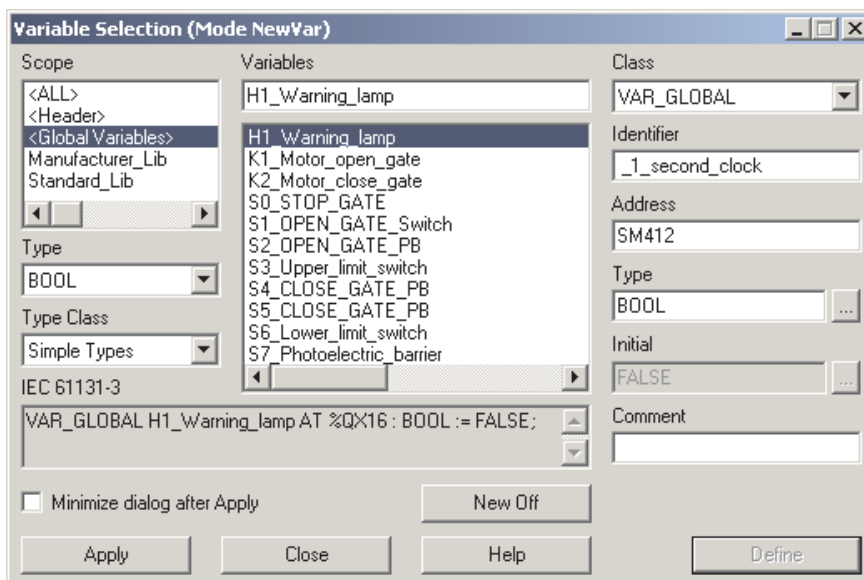


Если ни один из конечных выключателей не активирован, значит дверь открывается или закрывается, либо остановилась в промежуточном положении. В этих случаях сигнальная лампа мигает. Частота мигания регулируется специальным реле SM412, которое автоматически устанавливается и сбрасывается с интервалом 1 с (см. раздел 5.2). При вводе программы SM412 назначается глобальной переменной.

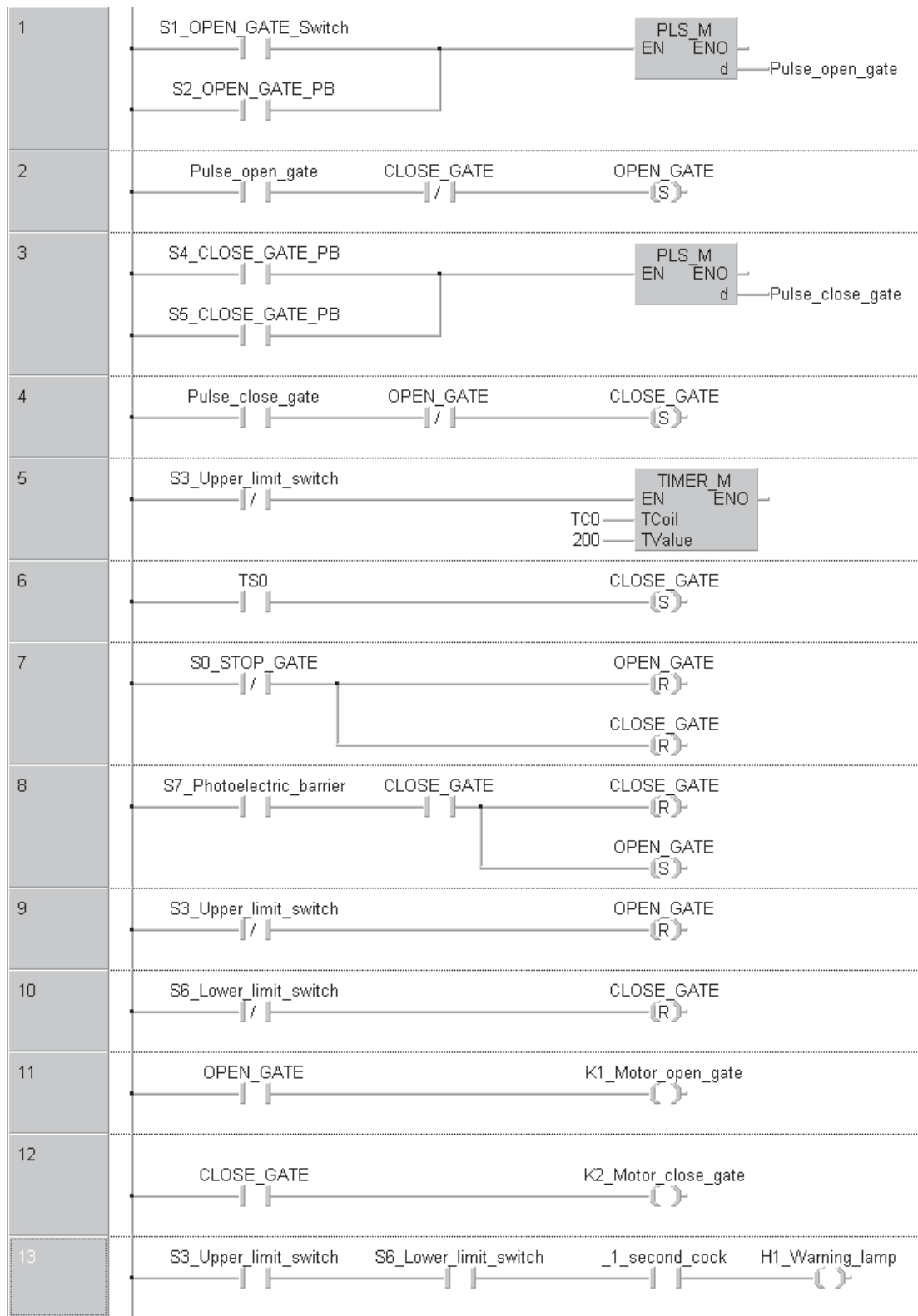


После ввода имени переменной «_1_second_clock» появится показанное слева сообщение, поскольку данная переменная пока не существует. Щёлкните кнопку Define global (Определить глобальную).

В окне Variable Selection (Выбор переменной) введите SM412 в поле Address (Адрес). Затем щёлкните кнопку Define (Определить).



На следующем рисунке показана полная схема в формате Ladder Diagram для управления свёртывающейся шторной дверью.



ПРИМЕЧАНИЕ

Крайне важна последовательность команд, особенно сброс переменных OPEN_GATE и CLOSE_GATE средствами защиты в конце программы после установки данных переменных. В связи с тем, что программа выполняется сверху вниз (см. раздел 2.2), сброс имеет более высокий приоритет над установкой, что обеспечивает безопасность.

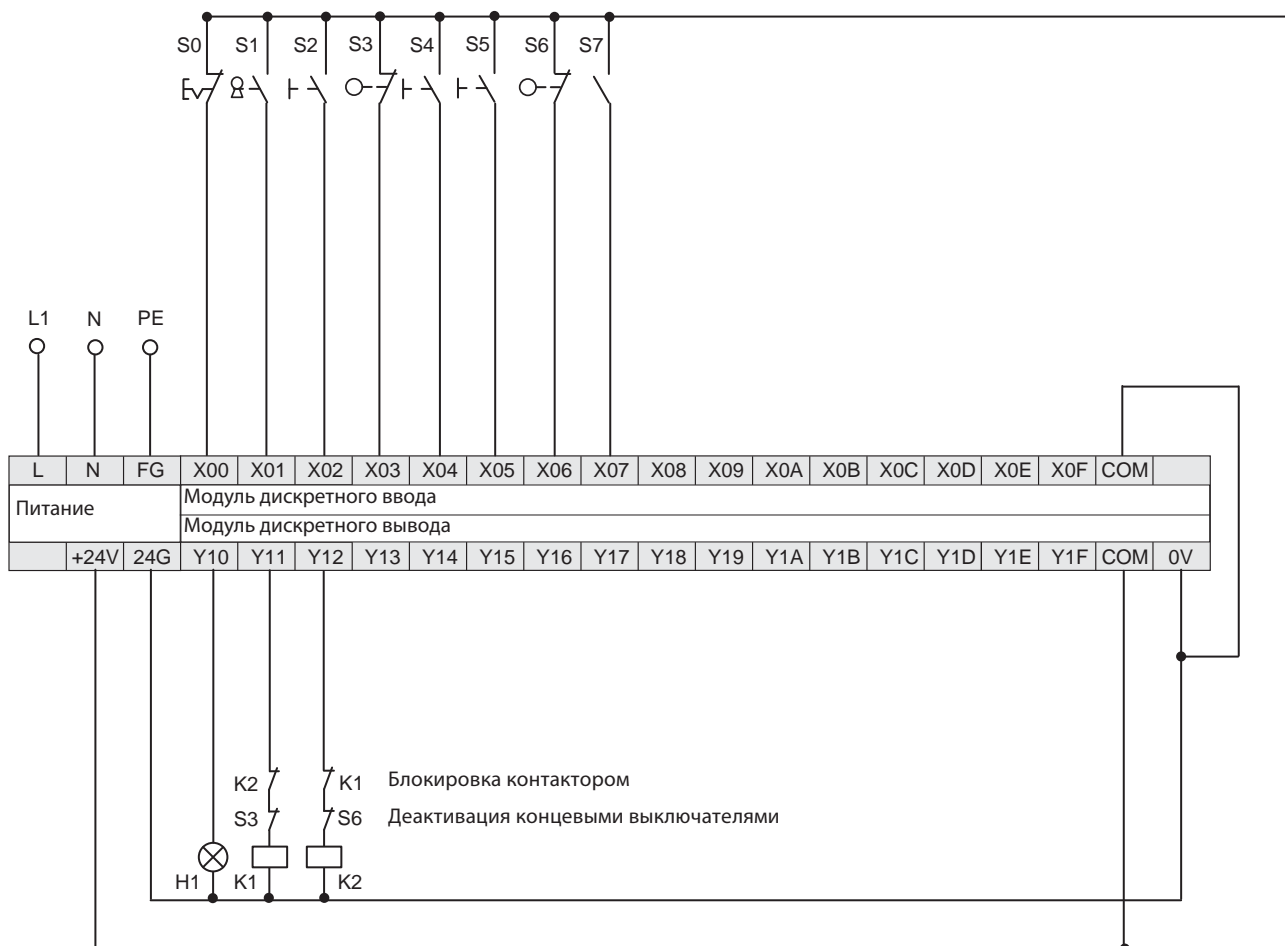
4.9.3 Аппаратура

Систему управления свёртывающейся шторной дверью можно реализовать с помощью следующих компонентов System Q:

- Базовое шасси с не менее чем двумя слотами для модулей ввода/вывода, например, Q33B
- Источник питания Q62P
 Данный модуль питания обеспечивает 24 В пост. для питания датчиков и сигнальных ламп. Следует учитывать, что максимальная нагрузка для данного выхода составляет 0,6 А.
- Модуль ЦП (по необходимости)*
- 1 модуль дискретного ввода QX80 с 16 входами (общий минус)
- 1 модуль дискретного вывода QY80 с 16 транзисторными выходами (переключающими на плюс)

* Несомненно, что использование ПЛК серии System Q только для управления свёртывающейся шторной дверью является необоснованным. При решении только одной задачи ЦП будет незагружен. Но в составе сложного приложения, например, для управления технологической линией, такое применение вполне реально.

Подключение ПЛК



Обозначение электрических компонентов и функций приведено на следующей странице.

Имя	Назначение	Адрес	Комментарии
S0	Кнопка СТОП	X0	Размыкающий контакт (нормально замкнутый, НЗ)
S1	Выключатель ОТКРЫТЬ с ключом (снаружи)	X1	Замыкающий контакт (нормально разомкнутый, НР)
S2	Кнопка ОТКРЫТЬ (внутри)	X2	
S3	Верхний концевой выключатель (открывание двери)	X3	Размыкающий контакт (НЗ)
S4	Кнопка ЗАКРЫТЬ (внутри)	X4	Замыкающие контакты (НР)
S5	Кнопка ЗАКРЫТЬ (снаружи)	X5	
S6	Нижний концевой выключатель (закрывание двери)	X6	Размыкающий контакт (НЗ)
S7	Фотоэлемент	X7	При обнаружении препятствия X7 устанавливается на «1»
H1	Сигнальная лампа	Y10	—
K1	Контактор электродвигателя (обратное вращение эл/двиг.)	Y11	Обратное вращение – ОТКРЫВАНИЕ двери
K2	Контактор электродвигателя (прямое вращение эл/двиг.)	Y12	Прямое вращение – ЗАКРЫВАНИЕ двери

5 Устройства

Устройства в ПЛК используются непосредственно в командах программы управления. Их состояния сигналов можно считывать и изменять программой ПЛК. Ссылка на устройство состоит из двух частей:

- имя устройства
- адрес устройства

Пример ссылки устройства (например, вход 0)



5.1 Входы и выходы

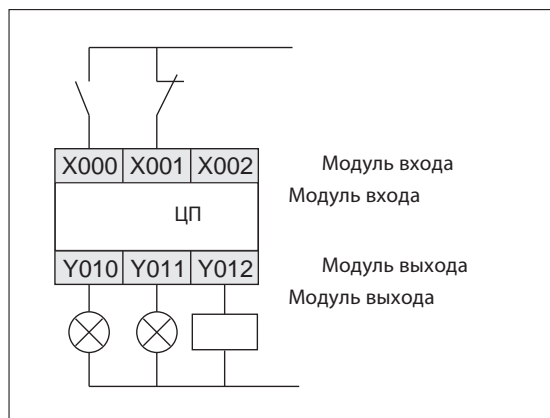
Входы и выходы служат для связи ПЛК с контролируемым процессом. При опросе входа программой ПЛК измеряется напряжение на входной клемме модуля входа. Поскольку данные входы являются дискретными, они могут иметь только два состояния: ВКЛ и ВЫКЛ. Когда напряжение на входной клемме достигает номинального значения (например, 24 В), данный вход включается (состояние «1»). Если напряжение ниже, вход считается отключенным (состояние сигнала «0»).

В ПЛК серии System Q для входов используется идентификатор «X». Один и тот же вход в программе может опрашиваться столько раз, сколько требуется.

ПРИМЕЧАНИЕ

ПЛК не способен изменять состояние входов. Например, невозможно выполнение команды OUT на входном устройстве.

Если команда вывода выполняется на выходе, результат текущей операции (состояние сигнала) применяется к выходной клемме модуля выхода. Если это выход реле, то реле замыкается (все реле имеют замыкающие контакты). Если это транзисторный выход, транзистор устанавливает соединение и активирует подключенную цепь.



На рисунке слева показан пример подключения выключателей к входам, а ламп и контакторов к выходам ПЛК серии System Q.

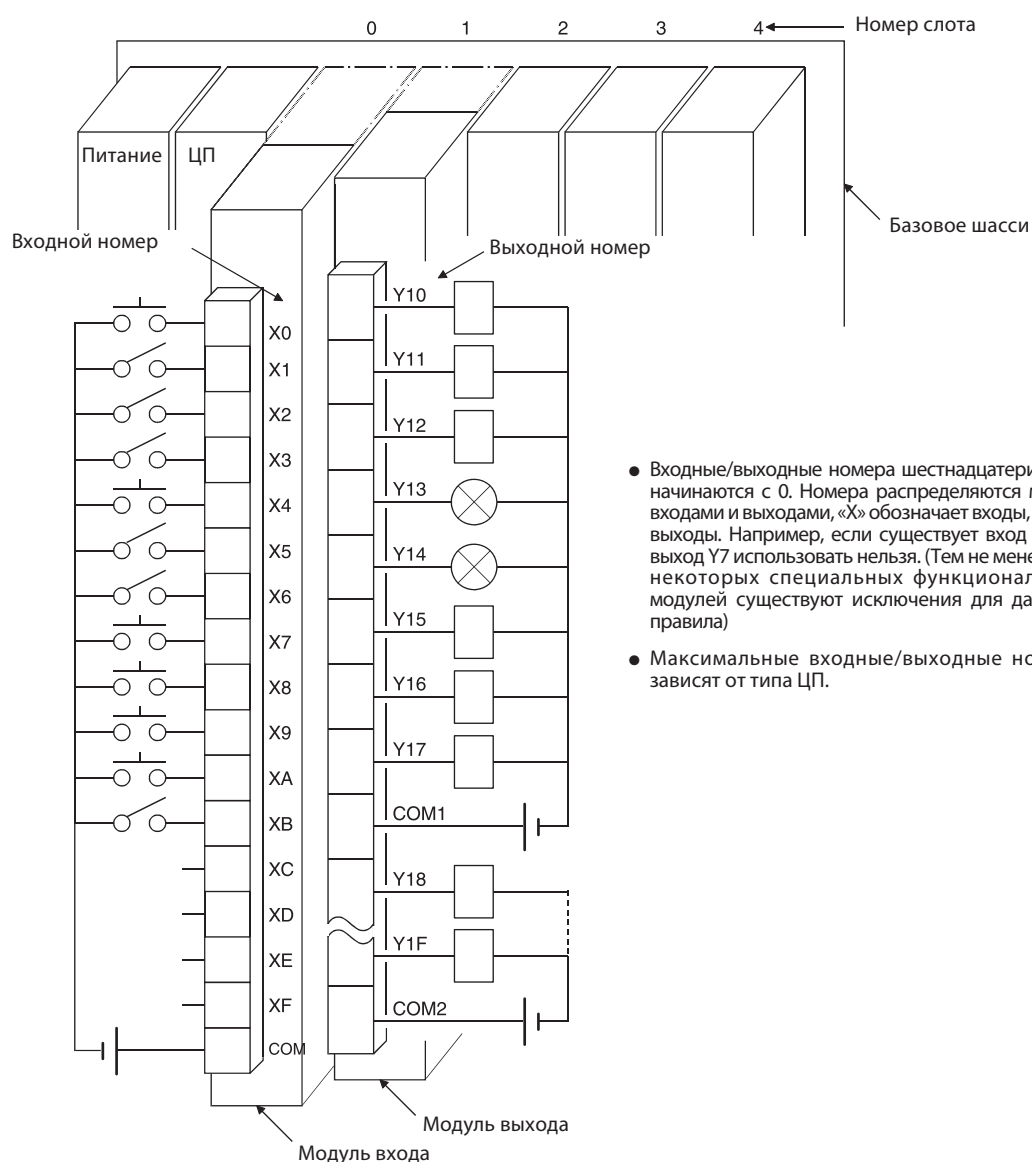
Для выходных устройств используется идентификатор «Y». Выходы можно использовать как в командах логических операций, так и с командами вывода. Однако следует помнить, что команду вывода можно использовать для конкретного выхода только один раз (см. также раздел 4.7.2).

5.1.1 Внешние сигналы ввода/вывода и номера ввода/вывода

Сигналы с внешних входных устройств заменяются входными номерами, которые определяются позицией установки (см. раздел 3.2.2) и номерами клемм подключенного входного модуля и обрабатываются в программе.

Выходы (катушки) результатов операций программы используют выходные номера, которые также определяются позицией установки и номерами клемм выходного модуля, к которому подключены внешние выходные устройства.

Для входов и выходов используется шестнадцатеричная нумерация (0, 1, 2 ...9, A, B, C, D, E, F; 10, 11, 12 ...). Таким образом сигналы ввода/вывода распределяются в группы по 16 входов или выходов.



- Входные/выходные номера шестнадцатеричные, начинаются с 0. Номера распределяются между входами и выходами, «X» обозначает входы, а «Y» – выходы. Например, если существует вход X7, то выход Y7 использовать нельзя. (Тем не менее, для некоторых специальных функциональных модулей существуют исключения для данного правила)
- Максимальные входные/выходные номера зависят от типа ЦП.

5.1.2 Входы и выходы System Q

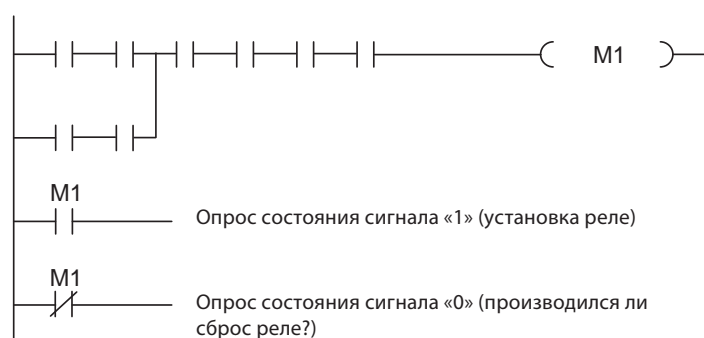
В следующей таблице приведен общий обзор входов и выходов контроллеров серии System Q.

Устройство	Входы и выходы		
	Ввод/вывод на базовом шасси и шасси расширения	Ввод/вывод на базовом шасси и шасси расширения и удаленный ввод/вывод	
Идентификатор устройства	X (входы), Y (выходы)		
Тип устройства	Битовое устройство		
Возможные значения	0 и 1		
Формат адреса устройства	Шестнадцатеричный		
Кол-во устройств и адресов (зависит от типа ЦП)	Q00J	256 (с X/Y000 по X/Y00FF)	2048 (с X/Y000 по X/Y07FF)
	Q00	1024 (с X/Y000 по X/03FF)	2048 (с X/Y000 по X/Y07FF)
	Q01		
	Q02	4096 (с X/Y000 по X/Y0FFF)	8192 (с X/Y000 по X/Y1FFF)
	Q02H		
	Q06H		
	Q12H		
	Q25H		
	Q12PH		
	Q25PH		

5.2 Реле

В программах ПЛК зачастую требуется временное хранение промежуточных двоичных результатов (состояние сигнала «0» или «1») для последующего использования. Для этих целей в ПЛК предусмотрены специальные ячейки памяти, называемые «промежуточные реле» или для краткости просто «реле» (идентификатор устройства: «M»).

Двоичный результат операции можно сохранять в реле, например, с помощью команды OUT, а затем использовать данный результат в последующих операциях. Реле упрощает чтение программ и сокращает число программных шагов. В реле можно сохранять результаты операций, которые требуется использовать не один раз, затем в остальной части программы опрашивать реле столько раз, сколько нужно.



Помимо обычных реле в контроллерах серии System Q используются также запоминающие или «фиксируемые» реле. Все обычные нефиксируемые реле при отключении питания ПЛК сбрасываются в состояние сигнала «0», которое также является стандартным при включении контроллера. В отличие них фиксируемые реле сохраняют свое текущее состояние при отключении и последующем включении питания.

Устройство	Типы реле		
	Нефиксируемые реле	Фиксируемые реле	
Идентификатор устройства	M	L	
Тип устройства	Битовое устройство		
Возможные значения для устройства	0 и 1		
Формат адреса устройства	Десятичный		
Кол-во устройств и адресов	Q00J	8192 (M0–M8191)*	8192 (L0–L8191)*
	Q00		
	Q01		
	Q02		
	Q02H		
	Q06H		
	Q12H		
	Q25H		
	Q12PH		
Q25PH			

* Количество фиксируемых и нефиксируемых реле можно задавать параметрами ПЛК. Приведенные выше значения являются исходными настройками.

5.2.1 Специальные реле

Кроме реле, которые можно включать и отключать с помощью программы ПЛК, существует также еще один класс реле, называемых специальными или диагностическими реле, имеющими идентификатор устройства «SM». Данные реле содержат информацию о состоянии системы или используются для управления выполнением программы. В следующей таблице приведено всего лишь несколько примеров из множества существующих специальных реле.

Специальное реле	Назначение	Опции обработки программы
SM0	Ошибка ПЛК	Опрос состояния сигнала
SM51	Низкое напряжение батареи	
SM400	Когда ПЛК находится в режиме RUN, данное реле всегда устанавливается на «1»	
SM401	Когда ПЛК находится в режиме RUN, данное реле всегда устанавливается на «0»	
SM402	Импульс инициализации (после активации режима RUN данное реле устанавливается на «1» на один программный цикл)	
SM411	Импульс тактового сигнала: 0,2 с (0,1 с – ВКЛ; 0,1 с – ВЫКЛ)	
SM412	Импульс тактового сигнала: 1 с (0,5 с – ВКЛ; 0,5 с – ВЫКЛ)	
SM413	Импульс тактового сигнала: 2 с (1 с – ВКЛ; 1 с – ВЫКЛ)	
SM414	Импульс переменного тактового сигнала	

ПРИМЕЧАНИЕ

Обзор всех специальных реле приведен в руководстве по программированию для серий A/Q и System Q, кат. № 87431.

5.3 Таймеры

При управлении процессами зачастую требуется программировать определенные задержки начала и окончания некоторых операций. В контроллерах на релейно-контактной логике это реализуется с помощью реле времени. В ПЛК это обеспечивается с помощью программируемых внутренних таймеров.

Таймеры фактически представляют собой счётчики внутренних тактовых сигналов ПЛК (например, импульсов 0,1 с). Когда показание счётчика достигает значения уставки, включается выход таймера.

Таймер определяется четырьмя элементами:

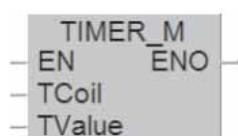
- Значение уставки (**TValue**)
- Текущее значение (**TN**)
- Катушка таймера (**TCoil, TC**)
- Контакт таймера (**TS**)

Все функции таймеров, такие как переключение с задержкой, активируются сигналом «1». Запуск и сброс таймеров программируется так же, как и выходы. Опрос выходов таймеров (TS) в программе можно использовать столько раз, сколько нужно.

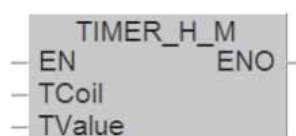
Существует два типа таймеров System Q: низкоскоростные и высокоскоростные. С помощью GX IEC Developer временную базу таймера (частоту подсчитываемых таймером тактовых сигналов) можно задавать в параметрах ПЛК в диапазоне от 1 до 1000 мс. Исходное значение составляет 100 мс. Временную базу высокоскоростных таймеров можно задавать в диапазоне от 0,1 до 100 мс. В этом случае исходное значение составляет 100 мс.

Тип таймера, низко- или высокоскоростной, определяется командой запуска таймера.

Определение низкоскоростного таймера

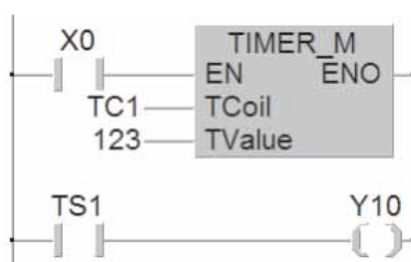


Определение высокоскоростного таймера



Пример программы с использованием низкоскоростного таймера

Ladder Diagram



В качестве переменной для входа TCoil команды TIMER_M указан адрес устройства таймера (в данном примере TC1).

MELSEC Instruction List

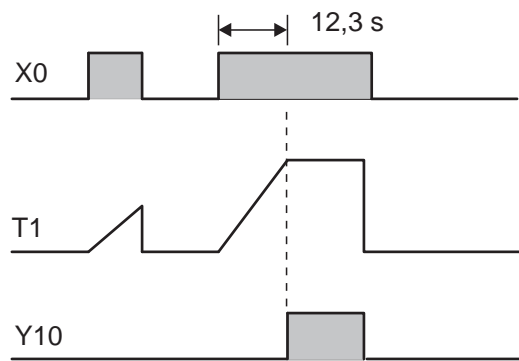
```
LD X0
OUT T1
K123
LD T1
OUT Y10
```

IEC Instruction List

```
LD X0
TIMER_M TC1, 123
LD TS1
ST Y10
```

В приведенном выше примере таймер T1 запускается при включении входа X0. Значение уставки составляет $123 \times 100 \text{ мс} = 12,3 \text{ с}$, то есть T200 включает выход Y10 с задержкой 1,23 с.

Последовательность сигнала, формируемого в данном примере программы, выглядит следующим образом.



Таймер считает внутренние 100-мс импульсы, пока X0 остается включенным. При достижении значения уставки выход T1 отключается.

Если отключается вход X0 или питание ПЛК, таймер сбрасывается и его выход также отключается.

Значение уставки таймера можно также указать десятичной величиной, сохраненной в регистре данных. Подробные сведения приведены в разделе 5.7.1.

Запоминающие таймеры

Помимо вышеуказанных обычных таймеров в контроллерах System Q также используются запоминающие таймеры, сохраняющие текущее показание счётчика времени даже после отключения контролирующего их устройства.

Текущее показание счётчика времени хранится в памяти, не зависящей от сбоев электропитания.

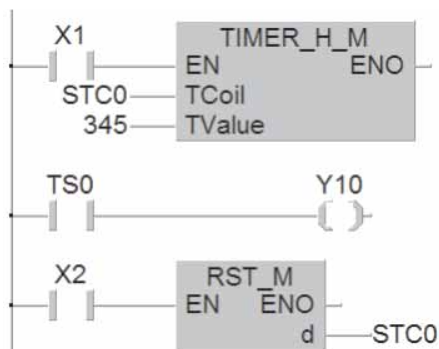
Для запоминающего таймера используется идентификатор устройства «ST». Аналогично обычным таймерам, запоминающие таймеры также можно программировать как низко- и высокоскоростные.

ПРИМЕЧАНИЕ

В состоянии поставки в параметрах ЦП ПЛК заданы обычные таймеры 2048 (2k), без запоминающих таймеров. Чтобы использовать запоминающие таймеры, их нужно задать в параметрах ПЛК.

Пример программы с использованием запоминающего таймера в качестве высокоскоростного

Ladder Diagram



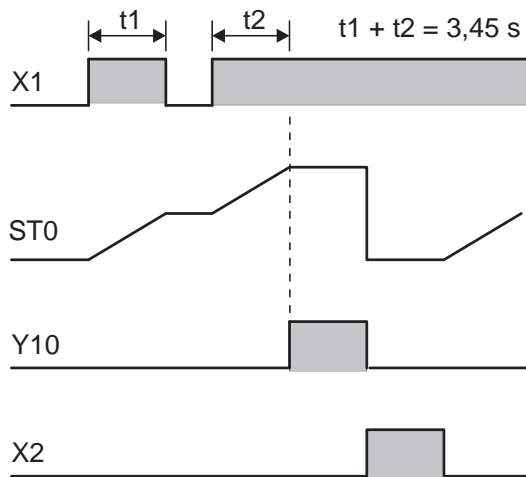
MELSEC Instruction List

```
LD X1
OUTH ST0
      K345
LD ST0
OUT Y10
LD X2
RST ST0
```

IEC Instruction List

```
LD X1
TIMER_H_M STC0, 345
LD STS0
OUT Y10
LD X2
R STC0
```

Таймер T0 запускается при включении входа X1. Значение уставки составляет $345 \times 10 \text{ ms} = 3,45 \text{ s}$. При достижении значения уставки T0 включает выход Y10. Вход X2 сбрасывает таймер и отключает его выход.



Когда X1 включен, таймер считает внутренние 10-мс импульсы. При отключении X1 сохраняется текущее показание счётчика времени. Когда текущее значение достигает значения уставки таймера, включается выход таймера.

Для сброса таймера нужно запрограммировать отдельную команду, так как он не сбрасывается при отключении входа X1 или питания ПЛК. Вход X2 сбрасывает таймер ST0 и отключает его выход.

Таймеры модулей ЦП ПЛК серии System Q

Устройство	Типы таймеров		
	Обычный таймер	Запоминающий таймер	
Идентификатор устройства	T	ST	
Тип устройства (для настройки и опроса)	Битовое устройство		
Возможные значения (выход таймера)	0 и 1		
Формат адреса устройства	Десятичный		
Задание значения уставки таймера	В виде десятичной целой константы. Значение уставки можно задавать напрямую в команде или косвенно в регистре данных.		
Кол-во устройств и адресов	Q00J	512 (с T0 по T511)*	0*
	Q00		
	Q01		
	Q02	2048 (с T0 по T2047)*	0*
	Q02H		
	Q06H		
	Q12H		
	Q25H		
	Q12RH		
	Q25RH		

* Исходные значения, количество таймеров можно задавать в параметрах ПЛК.

5.4 Счётчики

Для программирования операций счёта в ПЛК серии System Q предусмотрены также внутренние счётчики.

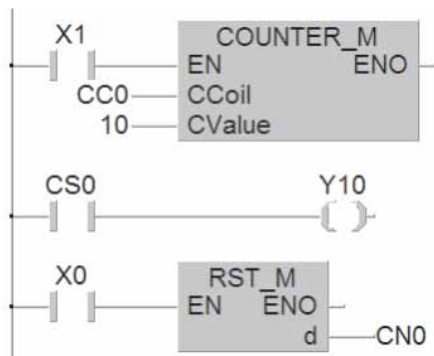
Счётчики используются для счёта импульсов сигналов, подаваемых программой на их входы. Когда текущее значение счётчика достигает заданного программой значения уставки, включается выход счётчика. Как и в случае с таймерами, опрос выходов счётчиков в программе можно использовать столько раз, сколько нужно.

Счётчик определяется четырьмя элементами:

- Значение уставки (CValue)
- Текущее значение (CN)
- Катушка счётчика (CCoil, CC)
- Контакт счётчика (CS)

Пример программы с использованием счётчика

Ladder Diagram



В качестве переменной для входа CCoil команды COUNTER_M указан адрес устройства счётчика (в данном примере **CC0**).

MELSEC Instruction List

```
LD      X1
OUT     C0
        K10
LD      C0
OUT     Y10
LD      X0
RST     C0
```

IEC Instruction List

```
LD      X1
COUNTER_M  CC0, 10
LD      CS0
ST      Y10
LD      X0
R       CN0
```

При каждом включении входа X1 значение счётчика C0 увеличивается на 1. Выход Y10 устанавливается после десяти включений-отключений X1 (значение уставки счётчика: K10).

Последовательность сигнала, формируемого данной программой, выглядит следующим образом.



В следующей таблице приведены основные характеристики данных счётчиков.

Характеристика	Счётчик
Принцип работы	При каждом нарастающем фронте сигнала на входе счётчика текущее значение увеличивается на 1 (при этом не требуется подача импульсного сигнала на вход счётчика)
Направление счёта	Возрастающее
Диапазон значений уставок	от 1 до 32767
Задание значения уставки	Напрямую в команде в виде десятичной константы (K) или косвенно в регистре данных
Действие при переполнении счётчика	Счёт достигает макс. значения 32 767, после чего значение счётчика не меняется
Выход счётчика	При достижении значения уставки выход остается включенным
Сброс	Для обнуления текущего значения счётчика и отключения его выхода используется команда RST

Обзор счётчиков

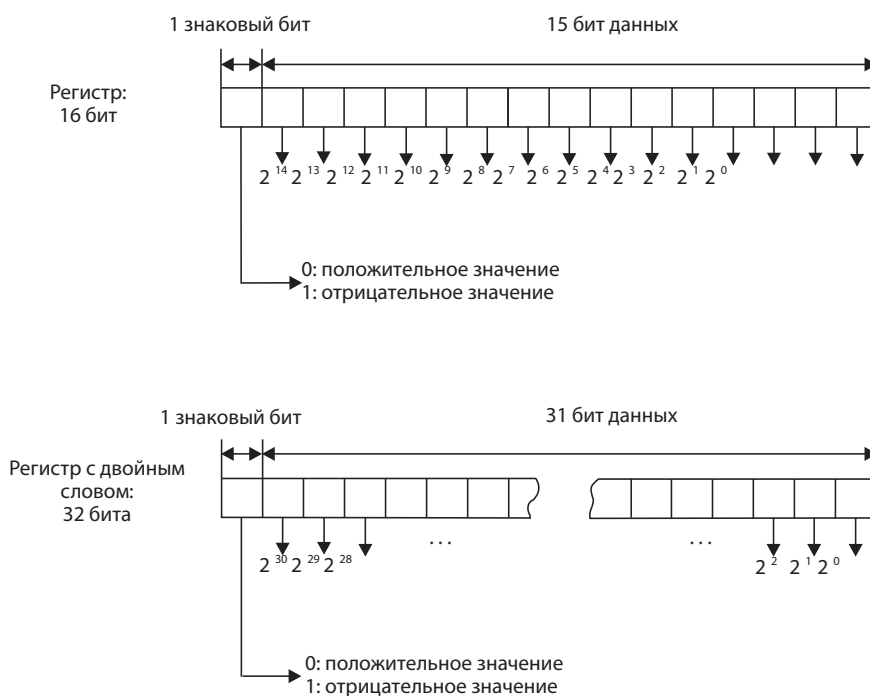
Устройство	Счётчик	
Идентификатор устройства	C	
Тип устройства (для настройки и опроса)	Бит-операнд	
Возможные значения устройства (выход счётчика)	0 и 1	
Формат адреса устройства	Десятичный	
Задание значения уставки счётчика	В виде десятичной целой константы. Значение уставки можно задавать напрямую в команде или косвенно в регистре данных.	
Кол-во устройств и адресов	Q00J	512* (с C0 по C511)
	Q00	
	Q01	
	Q02	1024* (с C0 по C1023)
	Q02H	
	Q06H	
	Q12H	
	Q25H	
	Q12PH	
	Q25PH	

* Исходные значения, количество счётчиков можно задавать в параметрах ПЛК.

5.5 Регистры

Для временного хранения результатов операций в ПЛК используются виртуальные реле. Но реле могут хранить только значения типа Вкл/Выкл и 1/0, то есть они не подходят для хранения результатов измерений и вычислений. Такие значения в контроллерах серии System Q хранятся в «регистрах».

Ширина регистра составляет 16 бит или одно слово (см. раздел). Для хранения 32-разрядных значений можно создавать регистры с двойным словом, объединяя два последовательных регистра данных.



В обычном регистре могут храниться значения в диапазоне от 0000H до FFFFH (от -32768 до 32767). В регистре с двойным словом могут храниться значения в диапазоне от 00000000H до FFFFFFFFH (от -2 147 483 648 до 2 147 483 647).

Для использования и оперирования регистрами в контроллерах System Q предусмотрено множество различных команд. Они позволяют записывать значения в регистры и считывать их, копировать содержимое регистров, сравнивать их и выполнять математические операции с их содержимым (см. раздел 6).

5.5.1 Регистры данных

В программах ПЛК регистры данных используются в качестве ячеек памяти. Значение, записанное программой в регистр данных, хранится до его перезаписи другим значением.

При использовании команд для оперирования 32-разрядными данными нужно лишь указать адрес 16-разрядного регистра. Более значимая часть 32-разрядных данных автоматически запишется в следующий последовательный регистр. Например, если для хранения 32-разрядного значения указать регистр D0, то он будет содержать биты с 0 по 15, а регистр D1 будет содержать биты с 16 по 31.

В случае выключения или остановки ПЛК

Помимо обычных регистров, содержимое которых теряется при выключении или остановке ПЛК, в модулях ЦП серии System Q предусмотрены фиксируемые регистры, содержимое которых в подобных случаях сохраняется.

Обзор регистров данных

Устройство		Регистр данных
Идентификатор устройства		D
Тип устройства (для настройки и опроса)		Устройство типа «слово» (для хранения значений с двойным словом можно объединять два регистра)
Возможные значения устройства		16-разрядный регистр: от 0000H до FFFFH (от -32768 до 32767) 32-разрядный регистр: от 00000000H до FFFFFFFFH (от -2147483648 до 2147483647)
Формат адреса устройства		Десятичный
Кол-во устройств и адресов	Q00J	11136* (с D0 по D11135)
	Q00	
	Q01	
	Q02	12288* (с D0 по D12287)
	Q02H	
	Q06H	
	Q12H	
	Q25H	
	Q12PH	
	Q25PH	

* Исходные значения, количество регистров данных можно задавать в параметрах ПЛК.

5.5.2 Специальные регистры

Аналогично специальным реле (см. раздел 5.2.1), в контроллерах System Q предусмотрены также специальные регистры. Для данных регистров используется идентификатор устройства «SD». Зачастую между специальными реле и специальными регистрами существует прямая связь. Пример: специальное реле SM51 показывает, что напряжение батареи ПЛК ниже нормы, а содержимое специального регистра SM51 предоставляет информацию о том, какая батарея разряжена (ЦП или карты памяти). В следующей таблице приведено всего лишь несколько примеров из множества существующих специальных регистров.

Специальный регистр	Назначение	Опции обработки программы
SD0	Код ошибки	Чтение содержимого регистра
SD392	Версия ПО	
SD520, SD521	Текущее время программного цикла	
SD210–SD213	Время и дата по встроенным часам реального времени (в двоично-десятичном коде)	Чтение содержимого регистра Изменение содержимого регистра

ПРИМЕЧАНИЕ

Обзор всех специальных регистров приведен в руководстве по программированию для серий A/Q и System Q, кат. № 87431.

5.5.3 Файловые регистры

При отключении питания содержимое файловых регистров также не теряется. Поэтому файловые регистры можно использовать для хранения значений, которые нужно передавать в регистры данных в случае выключения ПЛК, чтобы программа могла их использовать для вычислений, сравнений или в качестве значений уставок для таймеров.

Структура файловых регистров аналогичная регистрам данных.

Устройство		Файловые регистры
Идентификатор устройства		R
Тип устройства (для настройки и опроса)		Устройство типа «слово» (для хранения значений с двойным словом можно объединять два регистра)
Возможные значения устройства		16-разрядный регистр: от 0000H до FFFFH (от -32768 до 32767) 32-разрядный регистр: от 00000000H до FFFFFFFFH (от -2147483648 до 2147483647)
Формат адреса устройства		Десятичный
Кол-во устройств и адресов	Q00J	0
	Q00	32767 (с R0 по R32766)
	Q01	
	Q02	
	Q02H	32767 в каждом блоке (с R0 по R32766) При использовании карты памяти можно хранить до 1 миллиона дополнительных файловых регистров.
	Q06H	
	Q12H	
	Q25H	
	Q12PH	
	Q25PH	
Q25PH		

5.6 Константы

5.6.1 Десятичные и шестнадцатеричные константы

Десятичные и шестнадцатеричные константы служат для обозначения в последовательных программах соответственно десятичных и шестнадцатеричных данных (например, значений уставок для таймеров и счётчиков). ЦП ПЛК преобразовывает константы в двоичные числа.

Десятичные константы не имеют специального обозначения в форматах Ladder Diagram и IEC Instruction List. Шестнадцатеричные константы обозначаются префиксом «16#». Например, запись 16#12 интерпретируется ЦП ПЛК как шестнадцатеричное значение 12.

В формате Instruction List для десятичных констант использует префикс «K», а для шестнадцатеричных – префикс «H». Примеры: K100 – десятичное значение 100; H64 – шестнадцатеричное значение 64.

В следующей таблице приведены диапазоны значений для десятичных и шестнадцатеричных констант.

Константы	16 бит	32 бита
Десятичные	от -32768 до +32767	от -2147483648 до +2147483647
Шестнадцатеричные	от 0 до FFFF	от 0 до FFFFFFFF

5.6.2 Константы с плавающей десятичной запятой

Десятичные константы являются целыми значениями. Но значения с плавающей десятичной запятой (или действительные числа) имеют десятичные разряды, поэтому предоставляют преимущества при выполнении арифметических операций.

В последовательной программе константы с плавающей десятичной запятой обозначаются префиксом «E» (например, E1,234 или E1,234 + 3). Как видно из примера, такие константы в программе можно назначать выражением с показателем степени или без него.

- Назначение константы без показателя степени

Требуемое значение назначается обычным способом. Пример: 10,2345 становится «E10,2345».

- Назначение константы с показателем степени

Значение делится на основание и показатель степени. Основание степени: 10 (10). Пример: 1234 можно представить как 1,234 x 1000 или, в степенном выражении, как 1,234 x 10. В последовательной программе данное значение становится равным E1,234 + 3. (+3 обозначает 10).

Для констант с плавающей десятичной запятой предусмотрены следующие диапазоны значений:

с $-1,0 \times 2^{128}$ по $-1,0 \times 2^{-126}$,
0
и с $1,0 \times 2^{-126}$ по $1,0 \times 2^{+128}$

5.6.3 Константы типа «символьная строка»

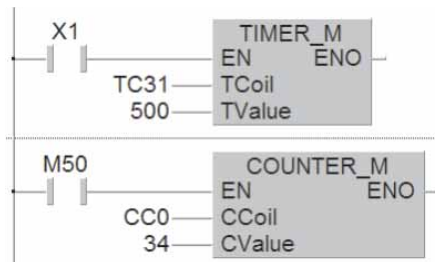
Если в последовательной программе приводятся символы в кавычках, они интерпретируются как ASCII-код (например, «MOTOR12»). Один символ занимает 1 байт. В качестве символьной строки можно использовать до 32 символов.

5.7 Рекомендации по программированию таймеров и счётчиков

5.7.1 Косвенное задание значений уставок таймеров и счётчиков

Обычным способом прямого задания значений уставок таймеров и счётчиков является команда вывода.

Ladder Diagram



MELSEC Instruction List

```
LD X1
OUT T31
K500
LD M50
OUT C0
K34
```

IEC Instruction List

```
LD X1
TIMER_M TC31, 500
LD M50
COUNTER_M CC0, K34
```

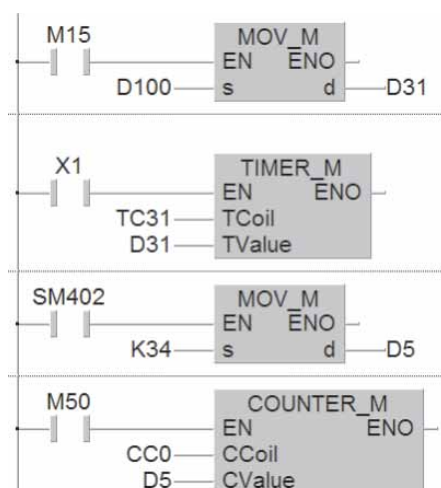
В приведенном выше примере T31 – это 100-мс таймер. Константа K500 задает задержку $500 \times 0,1 \text{ с} = 50 \text{ с}$. Значение уставки 34 для счётчика C0 также задается напрямую константой K34.

Преимуществом такого задания значений уставок является то, что после задания об уставках можно забыть. Эти значения в программы всегда верны, даже после сбоя питания и сразу после включения контроллера. Однако существует и недостаток – если требуется изменить значение уставки, приходится редактировать программу. Это особенно справедливо в отношении значений уставок таймеров, которые зачастую настраиваются при конфигурировании контроллера и тестировании программы.

Значения уставок таймеров и счётчиков можно также хранить в регистрах данных и считывать их программой из данных регистров. В этом случае при необходимости можно быстро изменять значения с помощью программатора или задавать значения уставок с помощью переключателей на пульте управления или на панели управления оператора.

Пример косвенного задания значений уставок приведен на следующей странице.

Ladder Diagram



MELSEC Instruction List

```
LD      M15
MOV     D100
        D31
LD      X1
OUT     T31
        D131
LD      SM402
MOV     K34
        D5
LD      M50
OUT     C0
        D5
```

IEC Instruction List

```
LD      M15
MOV_M   D100, D31
LD      X1
TIMER_M TC31, D31
LD      SM402
MOV_M   K34, D5
LD      M50
COUNTER_M CC0, D5
```

- Если реле M15 включено, содержимое регистра данных D100 копируется в D31. Данный регистр содержит значение уставки для T31. Для настройки содержимого D100 можно использовать программатор или блок управления.
- Специальное реле SM402 устанавливается только на один программный цикл, непосредственно после включения ПЛК. Это используется для копирования значения константы 34 в регистр данных D5, который затем используется как значение уставки для счётчика C0.

Для копирования значений уставок в регистры данных не требуется вводить программные команды. Для их задания, например, перед запуском программы, можно использовать программатор.

**ВНИМАНИЕ**

При отключении питания и переводе переключателя RUN/STOP в положение STOP значения уставок в обычных регистрах теряются. В этом случае при следующем включении питания и/или запуске ПЛК могут возникнуть опасные состояния, поскольку все значения уставок обнуляются.

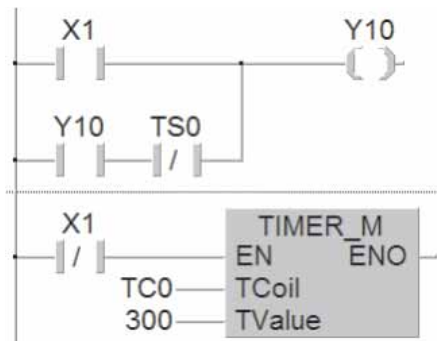
Если в программе не реализовано автоматическое копирование данных, необходимо обязательно использовать фиксируемые регистры данных для хранения значений уставок таймеров и счётчиков. Кроме того, в случае разрядки батареи резервного питания при выключении ПЛК содержимое данных регистров также теряется.

5.7.2 Задержка выключения

По умолчанию все таймеры в ПЛК серии System Q являются включающими с задержкой, то есть выход включается после заданного периода задержки. Однако зачастую также требуется запрограммировать операцию выключения с задержкой. Типичным примером является вытяжной вентилятор в ванной комнате, который должен продолжать работать в течение нескольких минут после выключения света.

Программа, версия 1 (фиксация)

Ladder Diagram



MELSEC Instruction List

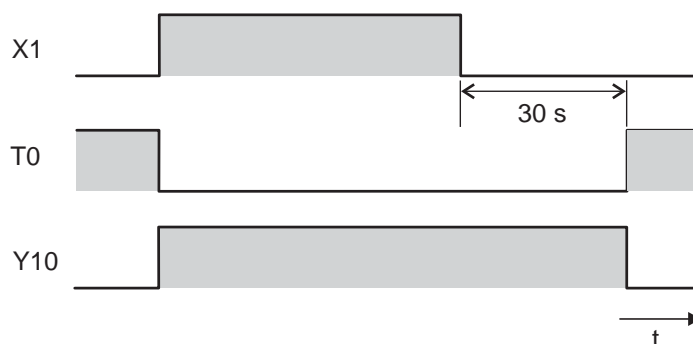
```
LD      X1
LD      Y10
ANI     T0
ORB
OUT     Y10
LDI     X1
OUT     T0
          K300
```

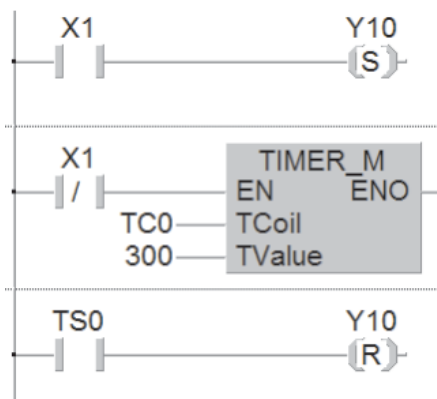
IEC Instruction List

```
LD      X1
OR(
          Y10
ANDN
          TS0
)
ST      Y10
LDN     X1
TIMER_M TC0, 300
```

Пока вход X1 (например, выключатель освещения) остается включенным, выход Y10 (вентилятор) также остается включенным. Благодаря функции фиксации Y10 остается включенным даже после выключения X1, потому что таймер T0 продолжает работать. Таймер T0 запускается при выключении X1. По завершении периода задержки (в данном примере $300 \times 0,1 \text{ с} = 30 \text{ с}$) T0 прерывает фиксацию Y10 и выключает выход.

Последовательность сигнала



Программа, версия 2 (установка/сброс)Ladder DiagramMELSEC Instruction List

```
LD      X1
SET     Y10
LDI     X1
OUT     T0
        K300
LD      T0
RST     Y000
```

IEC Instruction List

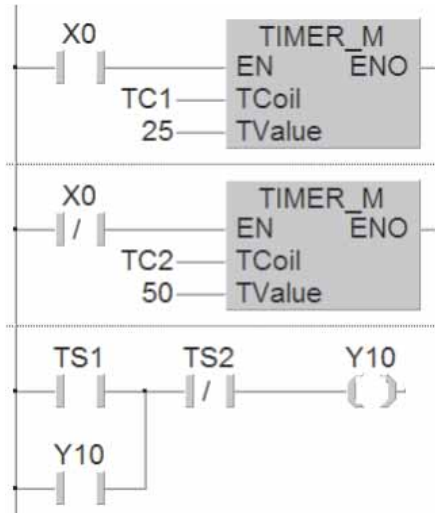
```
LD      X1
S       Y10
LDN     X1
TIMER_M TC0, 300
LD      TS0
R       Y10
```

Когда включается X1, устанавливается выход Y10 (включается). Таймер T0 запускается при выключении X1. По истечении периода задержки T0 сбрасывает выход Y10. Итоговая последовательность сигнала идентична последовательности, формируемой версией 1 данной программы.

5.7.3 Включение и выключение с задержкой

Иногда требуется включать выход с задержкой, а затем выключать его по истечении другой задержки. Это очень просто реализуется с помощью основных логических команд контроллера.

Ladder Diagram



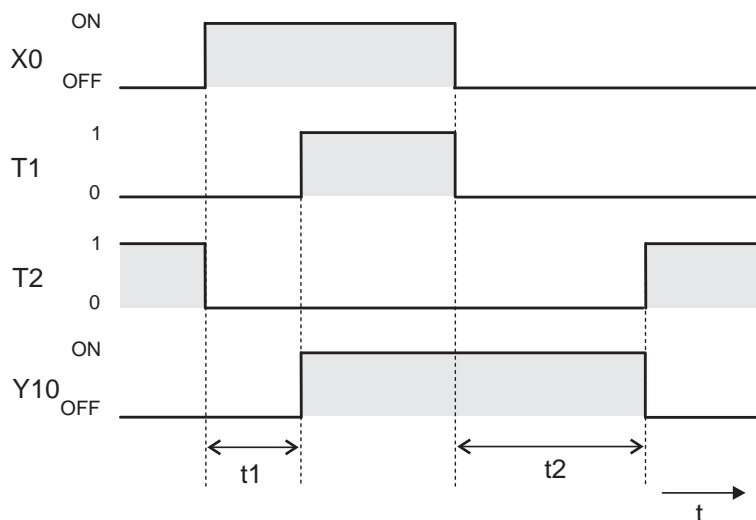
MELSEC Instruction List

```
LD      X0
OUT     T1
        K25
LDI     X0
OUT     T2
        K50
LD      T1
OR      Y10
ANI     T2
OUT     Y10
```

IEC Instruction List

```
LD      X0
TIMER_M TC1, 25
LDN     X0
TIMER_M TC2, 50
LD      TS1
OR      Y10
ANDN    TS2
ST      Y10
```

Последовательность сигнала



Когда включается X0, запускается T1 и сбрасывается T2. По истечении периода задержки t1 включается выход Y10. Он остается включенным, пока включен X0.

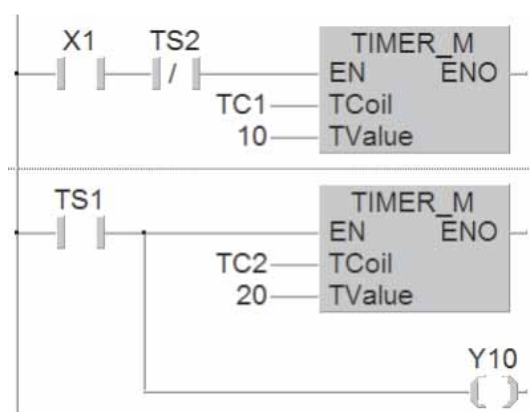
Но даже при выключении X0 и сбросе T1 выход Y10 остается включенным за счет функции фиксации. Таймер T2 запускается при выключении X1. По истечении периода задержки t2 выход Y10 сбрасывается.

5.7.4 Генераторы тактового сигнала

В данных контроллерах предусмотрены специальные реле, существенно упрощающие задачи программирования с использованием регулярного тактового сигнала (например, для управления мигающим сигнализатором неисправности). Например, реле SM413 включается и отключается с 1-секундным интервалом. Полные сведения обо всех специальных реле приведены в руководстве по программированию для серий A/Q и System Q, кат. № 87431.

Если требуется разная тактовая частота или различное время включения и отключения, генератор тактового сигнала можно запрограммировать с помощью двух таймеров, как показано в следующем примере.

Ladder Diagram



MELSEC Instruction List

```
LD      X1
ANI     T2
OUT     T1
        K10
LD      T1
OUT     T2
        K20
OUT     Y10
```

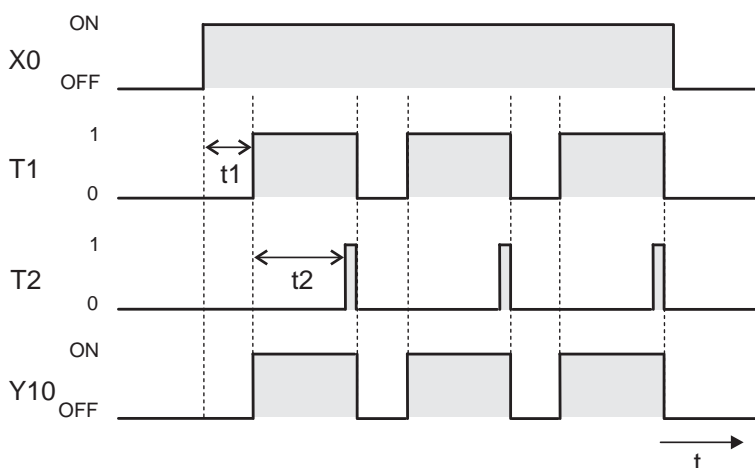
IEC Instruction List

```
LD      X1
ANDN    TS2
TIMER_M TC1, 10
LD      TS1
TIMER_M TC2, 20
ST      Y10
```

Вход X1 запускает тактовый генератор. Данный вход можно пропустить, если это нужно – тогда тактовый генератор будет постоянно включен. Для управления мигающим сигнализатором в программе можно использовать выход T1. Период включения определяется T2, а период отключения – T1.

Выход таймера T2 включается только на один программный цикл. На представленной ниже последовательности сигнала это время показано намного большим, чем оно есть на самом деле. T2 выключает T1 и сразу после этого T2 тоже отключается. В действительности это означает, что длительность периода увеличивается на время, требуемое для выполнения программного цикла. Однако длительность цикла составляет всего несколько миллисекунд, поэтому её обычно можно игнорировать.

Последовательность сигнала



6 Расширенное программирование

Перечисленные в разделе 3 основные логические команды можно использовать для имитации функций контроллера на релейно-контактной логике с помощью программируемого логического контроллера. Но это всего лишь небольшая часть возможностей современных ПЛК. Поскольку ПЛК созданы на основе микропроцессора, они способны легко выполнять такие операции, математические вычисления, сравнение чисел, перевод из одной системы счисления в другую и обработка аналогового сигнала.

Такие функции, выходящие за рамки возможностей логических операций, выполняются с помощью специальных команд, которые называются прикладными командами.

6.1 Справочная таблица прикладных команд

Для прикладных команд используются краткие имена на основе названия их функций на английском языке. Например, команда перемещения 16-разрядных данных называется MOV.

При программировании прикладной команды вводится имя команды, после которого указывается имя устройства. В следующей таблице приведены все прикладные команды, которые в настоящее время поддерживаются контроллерами серии System Q. Перечень команд довольно обширный, но их не нужно запоминать. Для поиска команд при программировании можно использовать мощные функции помощи программных пакетов GX Developer и GX IEC Developer. В данном разделе рассматриваются только наиболее часто используемые команды, выделенные в справочной таблице серым фоном. Полное описание всех команд с примерами приведено в руководстве по программированию для серий A/Q и System Q, кат. № 87431.

ПРИМЕЧАНИЕ

Многие прикладные команды выполняются циклически или запускаются импульсом (с нарастающим фронтом входного состояния). В этом случае к имени команды добавляется буква «P». Пример: MOV -> каждый программный цикл выполняется до тех пор, пока входное условие является истиной; MOV P -> разовое выполнение при нарастающем фронте импульса сигнала, формируемого входным условием.

Категория	Команда	Назначение
Операции сравнения	LD=	Сравнение на «равно»
	LD>	Сравнение на «больше чем»
	LD<	Сравнение на «меньше чем»
	LD	Сравнение на «не равно»
	LD<=	Сравнение на «меньше чем или равно»
	LD>=	Сравнение на «больше чем или равно»
	AND=	Сравнение на «равно»
	AND>	Сравнение на «больше чем»
	AND<	Сравнение на «меньше чем»
	AND<>	Сравнение на «не равно»
	AND<=	Сравнение на «меньше чем или равно»
	AND>=	Сравнение на «больше чем или равно»
	OR=	Сравнение на «равно»
	OR>	Сравнение на «больше чем»
	OR<	Сравнение на «меньше чем»
	OR<>	Сравнение на «не равно»
OR<=	Сравнение на «меньше чем или равно»	
OR>=	Сравнение на «больше чем или равно»	

Категория		Команда	Назначение	
Операции сравнения	Сравнение 32-разрядных данных	LDD=	Сравнение 16-разрядных данных в рамках операций	
		LDD>		
		LDD<		
		LDD<>		
		LDD<=		
		LDD>=		
		ANDD=		
		ANDD>		
		ANDD<		
		ANDD<>		
		ANDD>=		
		ANDD<=		
		ORD=		
		ORD>		
		ORD<		
	ORD<>			
	ORD<=			
	ORD>=			
	Сравнение вещественных данных	Сравнение вещественных данных	LDE=	Сравнение данных в рамках операций
			LDE>	
			LDE<	
			LDE<>	
			LDE<=	
			LDE>=	
			ANDE=	
			ANDE>	
			ANDE<	
			ANDE<>	
			ANDE>=	
			ANDE<=	
			ORE=	
			ORE>	
			ORE<	
	ORE			
	ORE<=			
	ORE>=			
Сравнение данных символьных строк	Сравнение данных символьных строк	LD\$=	Сравнение двух символьных строк (один символ за раз) в рамках операций	
		LD\$>		
		LD\$<		
		LD\$<>		
		LD\$<=		
		LD\$>=		
		AND\$=		
		AND\$>		
		AND\$<		
		AND\$<>		
		AND\$>=		
		AND\$<=		
		OR\$=		
		OR\$>		
		OR\$<		
Операции сравнения	Сравнение данных символьных строк	OR\$<>	Сравнение двух символьных строк (один символ за раз) в рамках операций	
		OR\$<=		
		OR\$>=		
	Сравнение данных блоков	Сравнение данных блоков	BKCMPI=	Сравнение 16-разрядных двоичных данных в последовательных устройствах (блоках данных). Количество боков данных указывается в команде. Результат сохраняется в отдельной области.
			BKCMPI>	
			BKCMPI<	
			BKCMPI<>	
			BKCMPI<=	
			BKCMPI>=	

Категория		Команда	Назначение	
Команды математических операций	Сложение и вычитание	+	Сложение 16-разрядных двоичных данных	
		-	Вычитание 16-разрядных двоичных данных	
		D+	Сложение 32-разрядных двоичных данных	
		D-	Вычитание 32-разрядных двоичных данных	
		V+	Сложение 4-значных двоично-десятичных кодов	
		V-	Вычитание 4-значных двоично-десятичных кодов	
		DB+	Сложение 8-значных двоично-десятичных кодов	
		DB-	Вычитание 8-значных двоично-десятичных кодов	
		E+	Сложение значений с плавающей десятичной запятой	
		E-	Вычитание значений с плавающей десятичной запятой	
		BK+	Сложение 16-разрядных двоичных данных, хранящихся в блоках данных	
		BK-	Вычитание 16-разрядных двоичных данных, хранящихся в блоках данных	
	Умножение и деление	x	Умножение 16-разрядных двоичных данных	
		/	Деление 16-разрядных двоичных данных	
		Dx	Умножение 32-разрядных двоичных данных	
		D/	Деление 32-разрядных двоичных данных	
		Bx	Умножение 4-значных двоично-десятичных кодов	
		V/	Деление 4-значных двоично-десятичных кодов	
		DBx	Умножение 8-значных двоично-десятичных кодов	
		DB/	Деление 8-значных двоично-десятичных кодов	
	Комбинирование данных символьных строк	S+	Связывание одной символьной строки с другой	
	Инкремент и декремент	INC	Инкремент 16-разрядных двоичных данных (прибавление «1» к текущему значению)	
		DINC	Инкремент 32-разрядных двоичных данных	
		DEC	Декремент 16-разрядных двоичных данных (вычитание «1» из текущего значения)	
		DDEC	Декремент 32-разрядных двоичных данных	
	Команды преобразования данных	Двоичные данные -> двоично-десятичные коды	BCD	Преобразование 16-разрядных двоичных данных в двоично-десятичные коды
			DBCD	Преобразование 32-разрядных двоичных данных в двоично-десятичные коды
BKBCD			Преобразование блоков 16-разрядных двоичных данных в двоично-десятичные коды	
Двоично-десятичные коды -> двоичные данные		BIN	Преобразование 4-значных двоично-десятичных кодов в двоичные данные	
		DBIN	Преобразование 8-значных двоично-десятичных кодов в двоичные данные	
		BKBIN	Преобразование блоков 4-значных двоично-десятичных кодов в блоки двоичных данных	
Двоичные данные -> значения с плавающей десятичной запятой		FLT	Преобразование 16-разрядных двоичных данных в значения с плавающей десятичной запятой	
		DFLT	Преобразование 32-разрядных двоичных данных в значения с плавающей десятичной запятой	
Значения с плавающей десятичной запятой -> двоичные данные		INT	Преобразование значений с плавающей десятичной запятой в 16-разрядные двоичные данные	
		DINT	Преобразование значений с плавающей десятичной запятой в 32-разрядные двоичные данные	
Двоичные данные -> двоичные данные		DBL	Преобразование 16-разрядных двоичных данных в 32-разрядные двоичные данные	
		WORD	Преобразование 32-разрядных двоичных данных в 16-разрядные двоичные данные	

Категория		Команда	Назначение
Команды преобразования данных	Двоичные данные -> двоичный циклический код	GRY	Преобразование 16-разрядных двоичных данных в двоичный циклический код
		DGRY	Преобразование 16-разрядных двоичных данных в двоичный циклический код
	Двоичные циклический код -> двоичные данные	GBIN	Преобразование двоичного циклического кода в 16-разрядные двоичные данные
		DGBIN	Преобразование двоичного циклического кода в 32-разрядные двоичные данные
	Перемена знака	NEG	Дополнение 2 разрядами (перемена знака) 16-разрядных двоичных данных
		DNEG	Дополнение 2 разрядами (перемена знака) 32-разрядных двоичных данных
ENEG		Перемена знака значений с плавающей десятичной запятой	
Функции перемещения	для 16-разрядных двоичных данных	MOV	Перемещение одиночных 16-разрядных двоичных данных их одной области памяти в другую
		BMOV	Передача данных блоков
		FMOV	Групповое перемещение, копирование в несколько устройств
		XCH	Перестановка данных в указанных устройствах
		BXCH	Перестановка блока данных
		SWAP	Перестановка старшего и младшего байтов слова
		EROMWR	Запись данных в регистр EEPROM
	для 32-разрядных двоичных данных	DMOV	Перемещение одиночных 32-разрядных двоичных данных их одной области памяти в другую
		DXCH	Перестановка данных в указанных устройствах
	для значений с плавающей десятичной запятой	EMOV	Перемещение значений с плавающей десятичной запятой
	для символьных строк	\$MOV	Перемещение символьных строк
	Отрицание, логическая инверсия содержимого устройства	CML	Побитовая инверсия 16-разрядных двоичных данных
		DCML	Побитовая инверсия 32-разрядных двоичных данных
	для файлов	SP.FWRITE	Запись в файл
		SP.FREAD	Чтение из файла
для блока данных	RBMOV	Высокоскоростная передача блоков файловых регистров	
Команды ветвления программы		CJ	Условный переход в программе в указанную позицию
		SCJ	Условный переход при следующем цикле выполнения программы
		JMP	Команда перехода
		GOEND	Переход в конец программы
Команды управления выполнением программы	Разрешение прерываний	EI	Разрешение вызова прерванной программы
	Запрещение прерываний	DI	Запрещение выполнения прерванной программы
	Разрешение/запрещение отдельных прерываний	IMASK	Управление условиями выполнения прерванных программ
	Завершение прерванной программы	IRET	Возврат из прерванной программы в основную
Команды обновления данных	Входы и выходы	RFS	Обновление входов и выходов указанного диапазона устройств ввода/вывода во время одного цикла программы
	Данные по связи и интерфейсу	COM	Обновление данных по связи и интерфейсу
	Условие выполнения обновления связи	DI	Запрещение выполнения обновления связи
		EI	Разрешение выполнения обновления связи

Программирование команд FCALL, ECALL и EFCALL с помощью GX IEC Developer не предусмотрено.

Категория		Команда	Назначение
Команды логических операций	Логическое И	WAND	Комбинирование двух 16-разрядных устройств
		DAND	Комбинирование двух 32-разрядных устройств
		BKAND	Комбинирование 16-разрядных устройств в блоках данных
	Логическое ИЛИ	WOR	Комбинирование двух 16-разрядных устройств
		DOR	Комбинирование двух 32-разрядных устройств
		BKOR	Комбинирование 16-разрядных устройств в блоках данных
	Логическое исключение ИЛИ (исключающее ИЛИ)	WXOR	Комбинирование двух 16-разрядных устройств
		DXOR	Комбинирование двух 32-разрядных устройств
		BKXOR	Комбинирование 16-разрядных устройств в блоках данных
	Логическое исключение НЕ-ИЛИ (исключающее НЕ-ИЛИ)	WNXR	Комбинирование двух 16-разрядных устройств
DNXR		Комбинирование двух 32-разрядных устройств	
BKXNR		Комбинирование 16-разрядных устройств в блоках данных	
Команды циклического сдвига данных	16-разрядные данные	ROR	Циклический сдвиг битов вправо
		RCR	Циклический сдвиг битов вправо флагом переноса
		ROL	Циклический сдвиг битов влево
		RCL	Циклический сдвиг битов влево флагом переноса
	32-разрядные данные	DROR	Циклический сдвиг битов вправо
		DRCR	Циклический сдвиг битов вправо флагом переноса
		DROL	Циклический сдвиг битов влево
		DRCL	Циклический сдвиг битов влево флагом переноса
Команды сдвига данных	16-разрядные данные	SFR	Сдвиг n битов вправо в слове 16-разрядных данных (n: от 0 до 15)
		SFL	Сдвиг n битов влево в слове 16-разрядных данных (n: от 0 до 15)
	Битовые устройства	BSFR	Сдвиг нескольких битовых устройств на 1 бит вправо
		BSFL	Сдвиг нескольких битовых устройств на 1 бит влево
	Устройства типа «слово»	DSFR	Сдвиг нескольких устройств типа «слово» на 1 бит вправо или влево
		DSFL	
Команды обработки битов	Установка/сброс	BSET	Установка отдельных битов
		BRST	Сброс отдельных битов
		BKRST	Сброс группы битов
	Проверка битов	TEST	Проверка отдельных битов в словах 16-/32-разрядных данных
		DTEST	
Команды обработки данных	Поиск данных	SER	Поиск 16-разрядных данных
		DSER	Поиск 32-разрядных данных
	Проверка битов данных	SUM	Определение количества битов в словах 16-/32-разрядных данных
		DSUM	
	Декодирование данных	DECO	Декодирование 8-разрядных данных в 256-разрядные (двоичных в десятичные)
	Кодирование данных	ENCO	Кодирование 256-разрядных данных в 8-разрядные (десятичных в двоичные)
	7-сегментное декодирование	SEG	Преобразование 4-значных двоичных значений в 7-сегментные коды для отображения значений с 0 по F
	Разделение/объединение слов 16-разрядных данных	DIS	Разделение слова 16-разрядных данных в группы по 4 бита
		UNI	Сохранение каждого 4-го младшего бита 16-разрядных значений (до четырех) в 16-разрядном значении
		NDIS	Разделение данных в случайных битовых блоках
		NUNI	Объединение данных в случайных битовых блоках
		WTOB	Разделение данных в байтовых блоках
		BTOW	Объединение данных в байтовых блоках
	Поиск макс. значений	MAX	Поиск макс. значения в блоках 16-разрядных данных
		DMAX	Поиск макс. значения в блоках 32-разрядных данных
	Поиск мин. значений	MIN	Поиск мин. значения в блоках 16-разрядных данных
		DMIN	Поиск мин. значения в блоках 32-разрядных данных
	Sort	SORT	Упорядочение 16-разрядных данных
		DSORT	Упорядочение 32-разрядных данных
	Вычисление суммы	WSUM	Вычисление суммы блоков 16-разрядных двоичных данных
DWSUM		Вычисление суммы блоков 32-разрядных двоичных данных	
Команды структурирования программы	Команды повтора	FOR	Начало повтора программы
		NEXT	Конец повтора программы
		BREAK	Завершение цикла FOR/NEXT

Категория		Команда	Назначение
Команды структурирования программы	Подпрограммы	CALL	Вызов подпрограммы
		RET	Конец подпрограммы
		FCALL*	Сброс выходов в подпрограммах
		ECALL*	Вызов подпрограммы в файле программы
		EFCALL*	Сброс выходов в подпрограммах из файлов программ
	Индексация	IX	Индексация всех частей программы
		IXEND	
IXDEV			
	IXSET	Сохранение номеров проиндексированных устройств в списке индексации	
Команда оперирования таблицами данных	Запись данных	FIFW	Запись данных в таблицу данных
	Чтение данных	FIFR	Чтение данных, введенных первыми в таблицу данных
		FPOP	Чтение данных, введенных последними в таблицу данных
	Удаление данных	FDEL	Удаление из таблицы данных указанных блоков данных
Вставка данных	FINS	Вставка в таблицу данных указанных блоков данных	
Команды обращения к буферной памяти	Чтение	FROM	Чтение 16-разрядных данных из специального функционального модуля
		DFRO	Чтение 32-разрядных данных из специального функционального модуля
	Запись	TO	Запись 16-разрядных данных в специальный функциональный модуль
		DTO	Запись 32-разрядных данных в специальный функциональный модуль
Команды отображения	Вывод символов ASCII	PR	Вывод строки символов ASCII на периферийное устройство
		PRC	Вывод комментария (в кодах ASCII) на периферийное устройство
	Очистка дисплея	LEDR	Сброс сигнализаторов и светодиодных индикаторов
Диагностика ошибок и отладка	Контроль ошибок	CHKST	Команда запуска для команды CHK
		CHK	Контроль ошибок
		CHKCIR	Формирование цепей контроля для команды CHK
		CHKEND	Команда завершения части программы сформированными цепями контроля
	Сохранение состояния устройства	SLT	Установка фиксации состояния (сохранение состояния устройства)
		SLTR	Сброс фиксации состояния (удаление состояния устройства)
	Трассировка выборки	STRA	Установка трассировки выборки
		STRAR	Сброс трассировки выборки
	Трассировка программы	PTRA	Установка трассировки программы
		PTRAR	Сброс трассировки программы
		PTRAEXE	Трассировка выполнения программы
Трассировка	TRACE	Установка трассировки	
	TRACER	Удаление данных, сохраненных командой трассировки	
Комбинированное обращение к символическим строкам	Двоичные -> десятичные (ASCII)	BINDA	Преобразование 16-/32-разрядных двоичных данных в десятичные значения в кодах ASCII
		DBINDA	
	Двоичные -> шестнадцатеричные (ASCII)	BINHA	Преобразование 16-/32-разрядных двоичных данных в шестнадцатеричные значения в кодах ASCII
		DBINHA	
	Двоично-десятичные коды -> ASCII-коды	BCDDA	Преобразование 4-значных двоично-десятичных кодов в ASCII-коды
		DBCDDA	Преобразование 8-значных двоично-десятичных кодов в ASCII-коды
	Десятичные (ASCII) -> двоичные	DABIN	Преобразование десятичных кодов ASCII в 16-/32-разрядные двоичные данные
		DDABIN	
Шестнадцатеричные (ASCII) -> двоичные	HABIN	Преобразование десятичных кодов ASCII в 16-/32-разрядные двоичные данные	
	DHABIN		

Категория		Команда	Назначение
Комбинируван ие обработки символьных строк	Десятичные (ASCII) -> двоично-десятичные коды	DABCD	Преобразование десятичных кодов ASCII в 4-значные двоично-десятичные коды
		DDABCD	Преобразование десятичных кодов ASCII в 8-значные двоично-десятичные коды
	Чтение данных информации устрой- ства	COMRD	Чтение данных информации устройства и сохранение в ASCII-кодах
	Определение длины	LEN	Определение длины символьных строк
	Двоичные данные -> символьная строка	STR	Вставка десятичной запятой и преобразование 16-/32-разрядных двоичных данных в символьную строку
		DSTR	
	Символьная строка -> двоичные данные	VAL	Преобразование символьных строк в 16-/32-разрядные двоичные данные
		DVAL	
	Значения с плавающей запятой -> символьная строка	ESTR	Преобразование значений с плавающей запятой в символьные строки
	Символьная строка -> значения с плавающей запятой	EVAL	Преобразование символьных строк в значения с плавающей запятой
	Значения с плавающей запятой -> двоично-десятичные коды	EMOD	Преобразование значений с плавающей запятой в двоично-десятичные коды
	Значения с плавающей запятой -> десятичные	EREXP	Преобразование значений с плавающей запятой в двоично-десятичных кодах в десятичный формат
	16-разрядные двоичные данные -> ASCII-коды	ASC	Преобразование 16-разрядных двоичных данных в ASCII-коды
	ASCII-коды -> двоичные	HEX	Преобразование шестнадцатеричных символов ASCII в двоичные значения
	Извлечение данных символьной строки	RIGHT	Извлечение подстроки справа
LEFT		Извлечение подстроки слева	
Сохранение	MIDR	Сохранение указанных частей символьных строк	
Перемещение	MIDW	Перемещение символьных строк в заданную область	
Поиск	INSTR	Поиск символьной строки	
Instructions for floating point numbers	Команды тригонометрических функций	SIN	Функция синуса
		COS	Функция косинуса
		TAN	Функция тангенса
		ASIN	Функция арксинуса
		ACOS	Функция арккосинуса
		ATAN	Функция арктангенса
		RAD	Перевод градусов в радианы
	DEG	Перевод радиан в градусы	
	Команды математических функций	SQR	Извлечение квадратного корня
		EXP	Показательная функция с основанием Е и значением с плавающей запятой в качестве показателя степени
LOG		Вычисление логарифма	
Специальные функции	Рандомизация значений	RND	Генерирование случайных чисел
		SRND	Обновление последовательности случайных чисел
Команды для данных в двоично-десят ичных кодах	Команды тригонометрических функций	BSIN	Функция синуса
		BCOS	Функция косинуса
		BTAN	Функция тангенса
		BASIN	Функция арксинуса
		BACOS	Функция арккосинуса
		BATAN	Функция арктангенса
	Команды математических функций	BSQR	Извлечение квадратного корня для данных в 4-значных двоично-десятичных кодах
		BDSQR	Извлечение квадратного корня для данных в 8-значных двоично-десятичных кодах

Категория		Команда	Назначение
Команды управления данными	Регулирование ограничений	LIMIT	Ограничение выходных значений для 16-/32-разрядных двоичных данных
		DLIMIT	
	Регулирование зоны нечувствительности	BAND	Регулирование зоны нечувствительности для 16-/32-разрядных двоичных данных
		DBAND	
Регулирование зоны	ZONE	Регулирование зоны для 16-/32-разрядных двоичных данных	
	DZONE		
Команды для файлового регистра	Команды переключения	RSET	Переключение между блоками файлового регистра
		QDRET	Переключение между файлами в файловых регистрах
		QCDSET	Переключение между файлами комментариев в файловых регистрах
	Чтение	ZRRDB	Чтение байта напрямую в файловом регистре
Запись	ZRWRB	Запись напрямую в байт файлового регистра	
Операции со встроенными часами ПЛК	Чтение	DATERD	Чтение времени и даты в часах
	Установка	DATEWR	Запись времени и даты в часы ПЛК
	Сложение	DATE+	Прибавление даты в часах
	Вычитание	DATE-	Вычитание даты в часах
	Перевод единиц времени	SECOND	Перевод времени в часах/минутах/секундах в секунды
		HOURL	Перевод времени в секундах в часы/минуты/секунды
Команды для периферийных устройств	Вывод	MSG	Вывод сообщений на периферийные устройства
	Ввод	PKEY	Клавиатурный ввод данных с периферийных устройств
		KEY	Клавиатурный ввод числовых значений
Команды управления программой	Режим ожидания	PSTOP	Перевод программы в режим ожидания
		POFF	Перевод программы в режим ожидания и сброс выходов
	Режим контроля выполнения	PSCAN	Перевод программы в режим контроля выполнения
Режим медленного выполнения	PLOW	Перевод программы в режим медленного выполнения	
Команды для программ	Загрузка программы	PLOADP	Загрузка программы с карты памяти
	Удаление программы	PUNLOADP	Удаление ожидающей программы из программной памяти
	Удаление и загрузка	PSWAPP	Удаление ожидающей программы из программной памяти и загрузка программы с карты памяти
Команды передачи данных	Обновление	ZCOM	Обновление сетевых данных
	Маршрутизация	RTREAD	Чтение маршрутной информации
		RTWRITE	Запись маршрутной информации
Команды для многопроцессорных систем	Запись данных	S.TO	Запись данных в совместную память процессоров
	Чтение данных	FROM	Чтение данных из совместной памяти другого ЦП
	Обновление данных	COM	Обновление совместной памяти многопроцессорной системы
Команды управления системой	Сторожевой таймер	WDT	Сброс сторожевого таймера
	Информация модуля	UNIRD	Чтение информации модуля
		ZPUSH	Групповое сохранение содержимого индексных регистров
	Индексный регистр	ZPOP	Групповое восстановление содержимого индексных регистров
		Сохранение адреса устройства	ADRSET
	Контроль выполнения	DUTY	Предустановка контроля выполнения для устройств
Прикладные команды	Счётчик	UDCNT1	1-фазный входной суммирующий/вычитающий счётчик
		UDCNT2	2-фазный входной суммирующий/вычитающий счётчик
	Таймер	TTMR	Программируемый таймер
		STMR	Таймер специальной функции (низкоскоростной таймер)
		STMRH	Таймер специальной функции (высокоскоростной таймер)
	Команда для поворотного стола	ROTC	Позиционирование поворотного стола
	Линейное изменение сигнала	RAMP	Линейное изменение содержимого устройства
	Измерение плотности импульсов	SPD	Счёт импульсов на входе в течение указанного времени и сохранение результата
	Формирование импульсов	PLSY	Формирование импульсов с регулируемой частотой
Широтно-импульсная модуляция	PWM	Формирование импульсов с регулированием длительности цикла и времени запуска	
Входная матрица	MTR	Формирование входной матрицы для чтения информации	

Категория		Команда	Назначение	
Команды для модулей с последовательной связью	Чтение данных	BUFRCV CVS	Чтение принятых данных с интерфейсного модуля	
	Запись данных	PRR	Передача данных через интерфейсный модуль посредством пользовательских фреймов	
	Фреймы, зарегистрированные пользователем	GETE	Чтением фреймов, зарегистрированных пользователем	
PUTE		Регистрация/удаление пользовательских фреймов		
Команды для интерфейсных модулей PROFIBUS/DP	Чтение данных	BBLKRD	Чтение данных из буферной памяти интерфейсного модуля PROFIBUS/DP и сохранение данных в ЦП ПЛК	
	Запись данных	BBLKWR	Перемещение данных из ЦП ПЛК в буферную память интерфейсного модуля PROFIBUS/DP	
Команды для интерфейсных модулей ETHERNET	Чтение данных	BUFRCV	Чтение принятых данных из фиксированных буферов	
		CVS		
	Запись данных	BUFSND	Перемещение данных из ЦП ПЛК в интерфейсный модуль PROFIBUS/DP	
	Разомкнутое соединение	OPEN	Разомкнутое соединение	
	Замкнутое соединение	CLOSE	Замкнутое соединение	
	Сброс ошибки	ERRCLR	Сброс кода ошибки и выключение светодиода ERR LED	
	Чтение кода ошибки	ERRRD	Чтение кода ошибки из буферной памяти	
Команды для CC-Link	Повторная инициализация	UINI	Повторная инициализация интерфейсного модуля	
	Настройка параметров	RLPASET	Настройка параметров сети CC-Link и запуск канала передачи данных	
		Чтение данных	RIRD	Чтение из буферной памяти станции интеллектуальных устройств или из памяти устройства ЦП ПЛК
			RICV	Чтение данных из буферной памяти станции интеллектуальных устройств (с подтверждением связи)
	RIFR		Чтение данных с другой станции, введенной в буферную память с автоматическим обновлением головной станции CC-Link	
	Запись данных	RIWT	Запись в буферную память станции интеллектуальных устройств или в память устройства ЦП ПЛК	
		RISEND	Запись (с подтверждением) в буферную память станции интеллектуальных устройств	
		RITO	Чтение данных с ЦП ПЛК в буферную память с автоматическим обновлением головной станции CC-Link. После этого данные передаются на указанную станцию	

6.1.1 Дополнительные команды для ЦП ЦП управления непрерывным процессом

В целях обеспечения эффективного программирования ПИД-регуляторов для ЦП управления процессом типа Q12PHCPU и Q25PHCPU предусмотрены следующие команды.

Категория		Команда	Назначение
Команды управления вводом/выводом	Input	IN	Обработка аналогового ввода (текущего значения)
	Вывод	OUT1	Обработка вывода
		OUT2	
	Ручной вывод	MOUT	Обработка вывода в ручном режиме ПИД-регулирования
	PWM	DUTY	Вывод широтно-импульсно модулированного сигнала (от 0 до 100 %)
	Сравнение	BC	Сравнение входного значения с одним или двумя наборами значений уставок и вывод результата в виде битовых данных
Запоминание импульсов	PSUM	Суммирование входного значения с определением предела и вывод результата	
Команды функций управления	ПИД-регулирование	PID	Основное ПИД-регулирование
		2PID	ПИД-регулирование с 2 степенями свободы
		PIDP	ПИД-регулирование позиционного типа
	ПИ-регулирование	SPI	Простое ПИ-регулирование
	И-ПД-регулирование	IPD	И-ПД-регулирование
	ПИ-регулирование	BPI	ПИ-регулирование
	2-позиционный выключатель	ONF2	2-позиционный выключатель
3-позиционный выключатель	ONF3	3-позиционный выключатель	
Обработка сигнала	Регулирование скорости изменения	R	Ограничение скорости изменения выходного сигнала
	Тревожные сигналы при выходе за ограничения	PHPL	Проверка входного значения и выдача тревожных сигналов при выходе за верхний или нижний пределы
	Опережение/запаздывание	LLAG	Определение вывода команды LLAG: после входа с задержкой или перед входом
	Интегрирование	I	Интегрирование входного сигнала
	Дифференцирование	D	Дифференцирование входного сигнала
	Время запаздывания	DED	Вывод входного значения с запаздыванием
	Вывод макс./среднего/мин. значения	HS	Вывод максимума для группы входных значений (до 16)
		LS	Вывод минимума для группы входных значений (до 16)
		MID	Вывод среднего для группы входных значений (до 16)
	Среднее значение	AVE	Вычисление среднего для группы входных значений (до 16)
	Верхний/нижний ограничитель	LIMIT	Ограничение входного значения областью, заданной верхним и нижним пределами
	Скорость изменения	VLMT1	Ограничение скорости изменения выходного значения
		VLMT2	
	Зона нечувствительности	DBND	Если входное значение находится в зоне нечувствительности, выход отсутствует
	Устройство настройки программы	PGS	Регулирование выхода по схеме
	Переключатель контуров управления	SEL	В автоматическом режиме на выход подаются один или два входных сигнала В ручном режиме на выход подается регулируемое значение
Плавная передача управления	BUMP	Плавная передача управления при переключении с ручного режима на автоматический	
Аналоговая память	AMR	Увеличение или уменьшение выходного значения в фиксированном соотношении	

Категория		Команда	Назначение
Команды операций компенсации и преобразования	Полигон	FG	Выходное значение зависит от входного и полигонального шаблона, заданного пользователем.
	Инвертированный полигон	IFG	
	Filter	FLT	Выборка входного значения с заданными интервалами и вычисление среднего.
	Запоминание	SUM	Суммирование входного значения и вывод результата
	Компенсация температуры/давления	TPC	Компенсирование температуры или давления для входного значения. Результат является выходом.
	Преобразование технических единиц	ENG	Преобразование входного значения в единицах % в выходное значение в физических единицах.
	Обратное преобразование технических единиц	IENG	Преобразование входного значения в физических единицах в выходное значение в единицах %.
Команды арифметических операций	Addition	ADD	Арифметические операции с дополнительными коэффициентами
	Вычитание	SUB	
	Умножение	MUL	
	Деление	DIV	
	Извлечение корня	SQR	Извлечение квадратного корня из выходного значения.
	Абсолютное значение	ABS	Выходом является абсолютная величина входного значения.
Команды операций сравнения	Сравнение на «больше чем»	> (GT)	Сравнение входных значений с гистерезисом
	Сравнение на «меньше чем»	< (LT)	
	Сравнение на «равно»	= (EQ)	
	Сравнение на «больше чем или равно»	>= (GE)	
	Сравнение на «меньше чем или равно»	<= (LE)	

ПРИМЕЧАНИЕ

Подробная информация о командах ПИД-регулирования приведена в руководстве по программированию ЦП QnPD, кат. № 149256.

6.2 Команды для перемещения данных

Регистры данных в ПЛК служат для хранения измеренных и выходных значений, промежуточных результатов операций и табличных значений. Команды математических функций контроллеров могут считывать свои операнды непосредственно из регистра данных и записывать результаты обратно в регистры, если это необходимо. Для поддержки этих команд предусмотрены также дополнительные команды перемещения для копирования данных между регистрами и записи постоянных значений в регистры данных.

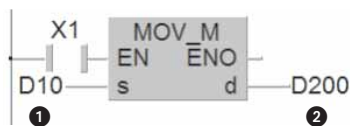
6.2.1 Перемещение отдельных значений с помощью команды MOV

Команда MOV служит для перемещения данных из указанного источника в заданное место назначения.

ПРИМЕЧАНИЕ

Несмотря на название команды, фактически она служит для копирования данных, без их удаления в источнике.

Ladder Diagram



MELSEC Instruction List

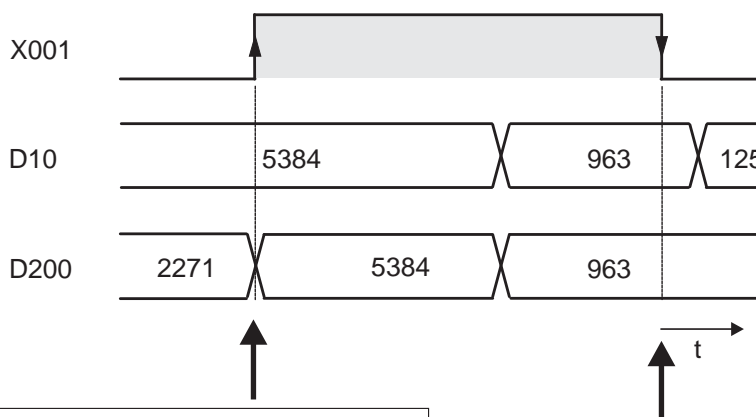
```
LD X1
MOV D10 ①
      D200 ②
```

IEC Instruction List

```
LD X1
MOV_M D10, D200
      ① ②
```

- ① Источник данных (может быть константой). В командах Ladder Diagram «s» означает источник.
- ② Адрес назначения данных (в командах Ladder Diagram «d» означает адрес назначения).

В данном примере значение из регистра данных D10 копируется в регистр D200, когда включен вход X1. Результатом является следующая последовательность сигнала.



Содержимое источника данных копируется по адресу назначения, пока условие на входе является истиной. При копировании содержимое источника данных не изменяется.

Когда условие на входе уже не является истиной, данная команда перестает изменять содержимое по адресу назначения данных.

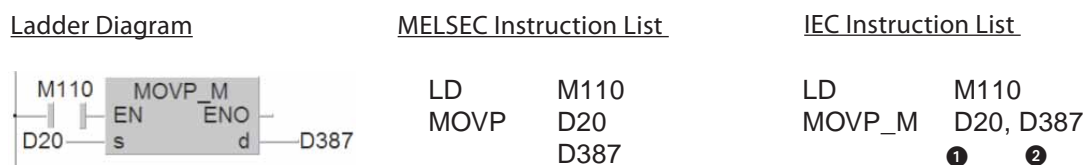
Импульсный запуск выполнения команды MOV

В некоторых приложениях требуется, чтобы значение записывалось по адресу назначения только на один программный цикл. Это требуется, например, когда другие команды в

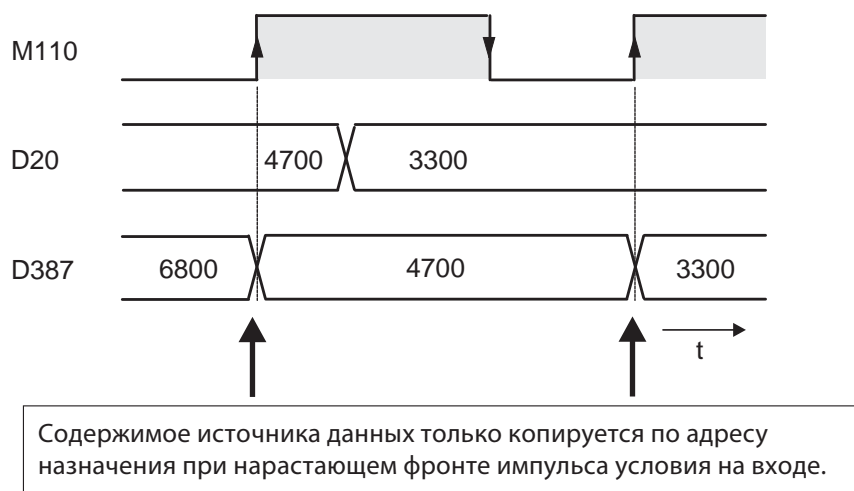
программе также используют для записи данный адрес назначения или операцию перемещения необходимо выполнять в заданное время.

Если к команде MOV добавить букву P (MOV_P), она выполнится один раз, при нарастающем фронте импульса сигнала, формируемого состоянием входа.

В следующем примере содержимое регистра данных D20 записывается в регистр D387, когда состояние M110 меняется с «0» на «1».

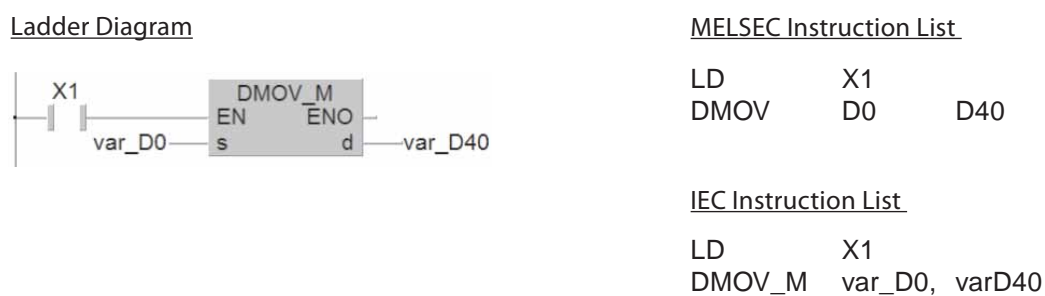


После выполнения данного одноразового копирования прекращается копирование в регистр D387, даже если M110 остается установленным. Последовательность сигнала выглядит следующим образом.



Перемещение 32-разрядных данных

Для перемещения 32-разрядных данных к команде MOV добавляется префикс «D» (DMOV).



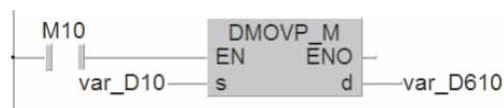
Когда вход X1 включен, содержимое регистров данных D0 и D1 записывается в регистры D40 и D41 (содержимое D0 копируется в D40, содержимое D1 в – D41).

ПРИМЕЧАНИЕ

В пакете GX IEC Developer в форматах Ladder Diagram и IEC Instruction List невозможно напрямую определять 32-разрядные устройства в качестве входных и выходных переменных. Данные устройства следует объявлять глобальными переменными (см. раздел). В данном примере на это указывают идентификаторы var_D0 и var_D40.

Для 32-разрядной команды DMOV существует также вариант с импульсным запуском.

Ladder Diagram



MELSEC Instruction List

```
LD      M10
DMOVP  D10  D610
```

IEC Instruction List

```
LD      X1
DMOVP_M var_D10, var_D610
```

Когда реле M10 установлено, содержимое регистров D10 и D11 записывается в регистры D610 и D611.

ПРИМЕЧАНИЕ

В форматах Ladder Diagram и IEC Instruction List 32-разрядные устройства следует объявлять глобальными переменными (см. раздел). Данные устройства невозможно вводить напрямую.

6.2.2 Перемещение групп битовых устройств

В предыдущем разделе было представлено использование команды MOV для записи в регистры данных констант или содержимого других регистров. Последовательные группы реле и других битовых устройств также можно использовать для хранения числовых значений и их копирования группами с помощью прикладных команд. Для этого нужно добавить к адресу первого битового устройства в качестве префикса коэффициент K и указать количество устройств, копируемых данной операцией.

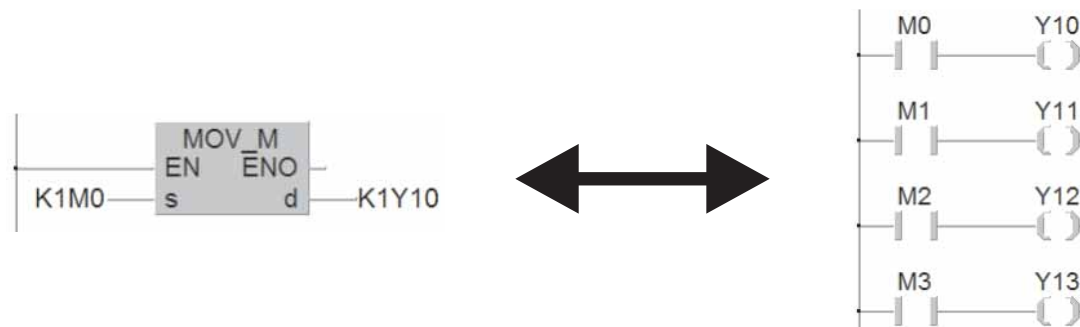
Счёт ведется группами по 4 битовых устройства, а коэффициент K указывает число таких групп: K1 – 4 устройства, K2 – 8 устройств, K3 – 12 устройств и т.д.

Например, K2M0 означает 8 реле: с M0 по M7. Предусмотренный диапазон: с K1 (4 устройства) по K8 (32 устройства).

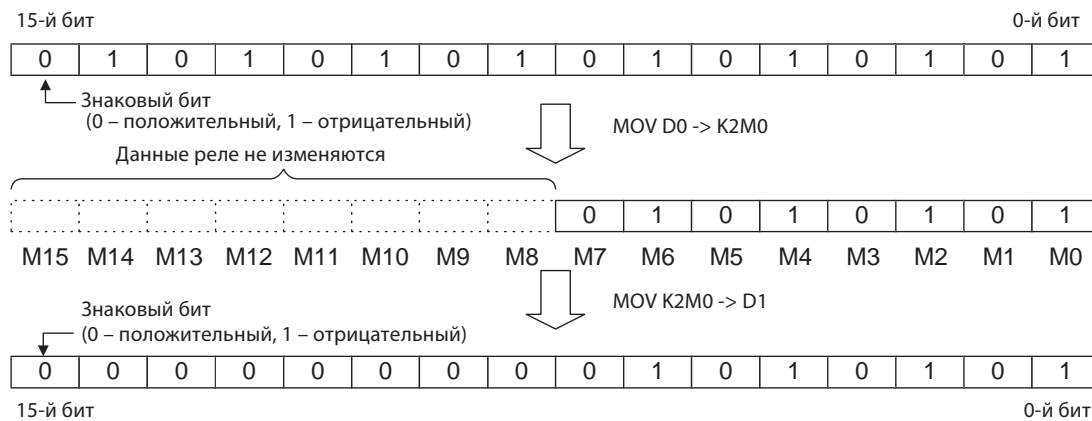
Примеры адресации групп битовых устройств:

- K1X0: 4 входа, начиная с X0 (с X0 по X3)
- K2X4: 8 входов, начиная с X4 (с X4 по X1B, шестнадцатеричные)
- K4M16: 16 реле, начиная с M16 (с M16 по M31)
- K3Y0: 12 выходов, начиная с Y0 (с Y0 по Y1B, шестнадцатеричные)
- K8M0: 32 реле, начиная с M0 (с M0 по M31)

Адресация групп битовых устройств с помощью одной команды ускоряет программирование и позволяет создавать более компактные программы. В следующих двух примерах выполняется передача состояния сигнала реле M0 и M3 на выходы Y10 и Y13.



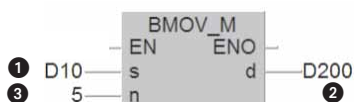
Если длина диапазона места назначения меньше диапазона источника, лишние биты просто игнорируются (см. следующий рисунок, верхний пример). Если длина места назначения больше источника, в лишние устройства записывается 0. В этом случае результат всегда положителен, поскольку 15-й бит интерпретируется как знаковый (см. следующий рисунок, нижний пример).



6.2.3 Перемещение блоков данных с помощью команды BMOV

Команда MOV, описание которой приведено в разделе , позволяет записывать по адресу назначения только единичные 16- и 32-разрядные значения. Для перемещения последовательных блоков данных можно запрограммировать несколько последовательностей команд MOV. Однако намного эффективнее использовать команду BMOV (Block MOVE), предназначенную специально для этих целей.

Ladder Diagram



MELSEC Instruction List

BMOV D10 ①
D200 ②
K5 ③

IEC Instruction List

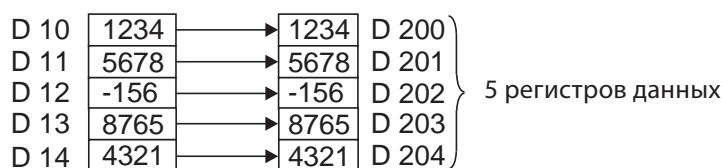
BMOV_M D10, 5, D200
① ③ ②

- ① Источник данных (16-разрядное устройство, первое в последовательности источников)
- ② Адрес назначения данных (16-разрядное устройство, первое в последовательности адресов назначения)
- ③ Количество перемещаемых элементов

Вышеуказанный пример реализуется следующим образом.

Источник данных (D10)

Адрес назначения данных (D200)

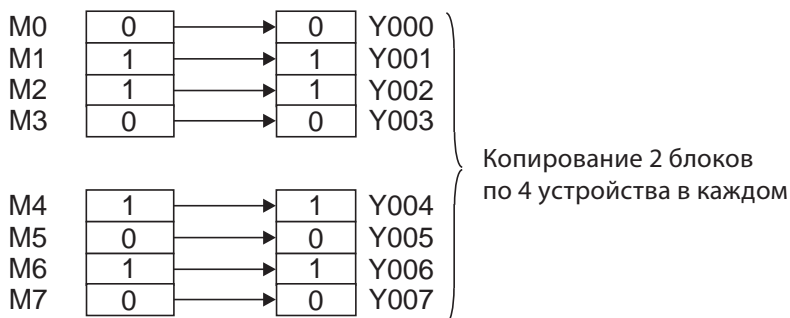


Для команды BMOV существует также вариант с импульсным запуском BMOVP (описание импульсного запуска выполнения приведено в разделе).

При перемещении блоков битовых устройств с помощью команды BMOV коэффициенты K источников данных и адресов назначения должны быть идентичными

Пример

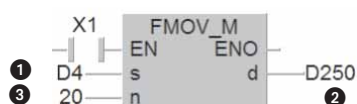
- Источник данных: K1M0
- Адрес назначения данных: K1Y0
- Количество перемещаемых элементов; 2



6.2.4 Копирование устройств-источников в несколько адресов назначения (FMOV)

Команда FMOV (Fill MOVE) служит для копирования содержимого устройства типа «слово» или константы в несколько последовательных устройств типа «слово». Она обычно используется для удаления таблиц данных и установки данных, зарегистрированных на предварительно заданное начальное значение.

Ladder Diagram



MELSEC Instruction List

```
FMOV  D4   ①
      D250  ②
      K20   ③
```

IEC Instruction List

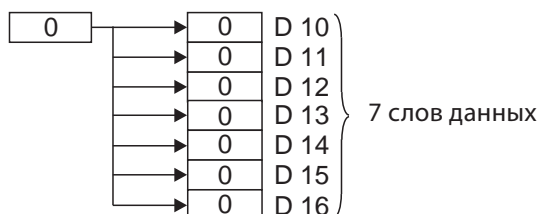
```
FMOV_M  D4, 20, D250
        ①  ③  ②
```

- ① Данные, записываемые на устройства-назначения (здесь также можно использовать константы)
- ② Адрес назначения данных (первое устройство в последовательности адресов назначения)
- ③ Количество записываемых элементов в данный диапазон адресов назначения

В следующем примере значение 0 записывается в 7 элементов.

- Источник данных: K0 (константа)
- Адрес назначения данных: D10
- Количество записываемых элементов в данный диапазон адресов назначения: 7

Источник данных («0») Адрес назначения данных (D10)



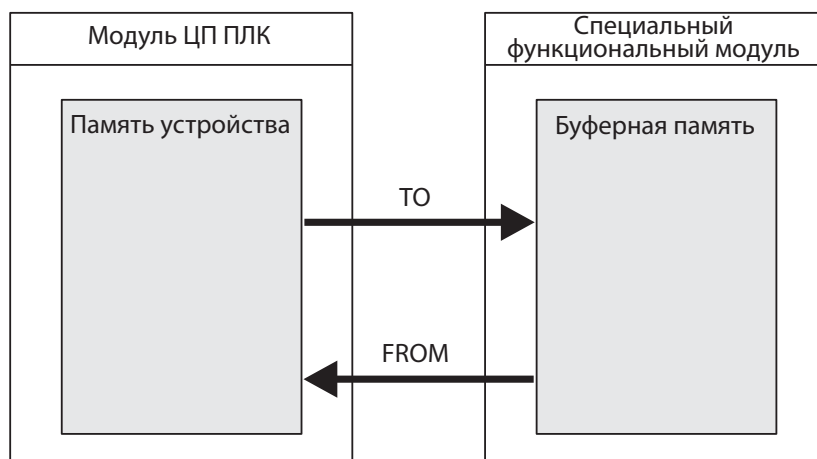
Для команды FMOV также существует также вариант с импульсным запуском FMOV_P (описание импульсного запуска выполнения приведено в разделе).

6.2.5 Обмен данными со специальными функциональными модулями

Функции контроллеров можно дополнять так называемым „специальными функциональными модулями“ например, для чтения аналоговых сигналов тока и напряжения, регулирования температуры и связи с внешним оборудованием.

Каждый специальный функциональный модуль имеет область памяти, назначенную в качестве буфера для временного хранения данных, таких как результаты измерения аналоговых значений или принятые данные. ЦП ПЛК обращается к данному буферу для считывания сохраненных в нем значений и записи новых значений, которые затем обрабатываются модулем (настройки функций модуля, данные для передачи и т.д.).

Поимо буферной памяти специальные функциональные модули снабжены дискретными входами и выходами. Данные сигналы ввода/вывода используются, например, для обмена сигналами состояния между ЦП ПЛК и специальными функциональными модулями. Для дискретных входов/выходов специальных функциональных модулей не требуются специальные команды. Оперирование этими входами и выходами осуществляется аналогично входам/выходам модулям дискретных входов/выходов. Но обмен данными между ЦП ПЛК и буферной памятью специальных функциональных модулей осуществляется с помощью двух специальных прикладных команд FROM и TO.



Буферная память рассчитана на 32767 отдельных адресуемых ячеек памяти, в каждой из которых может храниться 16 бит данных. Функции ячеек буферной памяти зависят от конкретного специального функционального модуля, подробные сведения приведены в документации для модуля.

Буферная память, адрес 0
Буферная память, адрес 1
Буферная память, адрес 2
:
:
Буферная память, адрес n-1
Буферная память, адрес n

Для использования команд FROM и TO требуется следующая информация:

- Специальный функциональный модуль для чтения или записи
- Адрес первой ячейки буферной памяти для чтения или записи
- Количество ячеек буферной памяти для чтения или записи
- Местоположение в ЦП ПЛК для сохранения данных из модуля или для их чтения и записи в модуль.

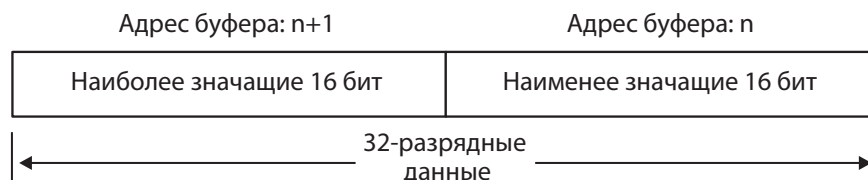
Адрес специального функционального модуля

На базовое шасси можно устанавливать несколько специальных функциональных модулей, поэтому для каждого модуля требуется уникальный идентификатор для адресации при обмене данными. Данный идентификатор определяется слотом, в котором установлен модуль, соответственно по номерам ввода/вывода, занимаемым дискретными входами и выходами специального функционального модуля (см. раздел).

Ключевым является головной адрес специального функционального модуля. Например, если специальный функциональный модуль занимает диапазон с X/Y010 по Y/X01F, головной адрес будет X/Y010. Но при использовании команды FROM или TO наименее значащая цифра опускается, поэтому головной адрес в этом случае будет «1». Если специальный функциональный модуль занимает диапазон с X/Y040 по Y/X04F, головной адрес будет «4».

Начальный адрес буферной памяти

К каждому из 32 767 буферных адресов можно обращаться напрямую, используя десятичные числа в диапазоне от 0 до 32 767. При обращении к 32-разрядным данным необходимо знать ячейку памяти с младшим адресом, где хранятся наименее значащие 16 бит, и ячейку памяти со старшим адресом, где хранятся наиболее значащие биты.

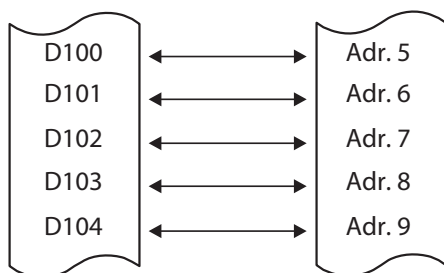


Это означает, что начальным адресом для 32-разрядных данных всегда является адрес, содержащий наименее значащие 16 бит двойного слова.

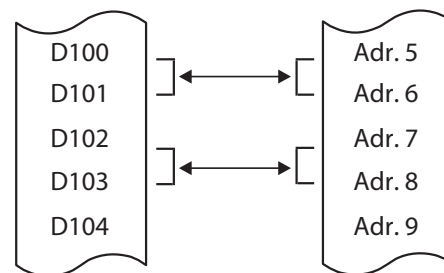
Количество передаваемых единиц данных

Объём данных определяется количеством передаваемых единиц данных. При выполнении команды FROM или TO в качестве 16-разрядной команды данным параметром является количество передаваемых слов. При использовании 32-разрядного варианта DFROM или DTO данным параметром является количество передаваемых двойных слов.

16-разрядная команда
Единиц данных: 5



32-разрядная команда
Единиц данных: 2



Источник данных и адрес назначения в ЦП ПЛК

В большинстве случаев используется чтение данных из регистров и их запись в специальные функциональные модули или копирование данных из буферов модулей в регистры данных в ЦП ПЛК. Однако в качестве источников данных и адресов назначения можно также использовать выходы, реле и текущие значения таймеров и счётчиков.

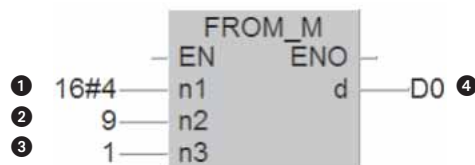
Импульсный запуск выполнения команд

При добавлении к командам префикса «P» передача данных инициируется импульсным запуском (подробное описание команды MOV приведено в разделе).

Использование команды FROM

Команда FROM служит для передачи данных из буфера специального функционального модуля в ЦП ПЛК. Данная операция представляет собой копирование, поэтому содержимое данных в буфере модуля не изменяется.

Ladder Diagram



MELSEC Instruction List

```
FROM   H4  ①
        K9  ②
        D0  ④
        K1  ③
```

IEC Instruction List

```
FROM_M  16#4, 9, 1, D0
        ①  ②  ③  ④
```

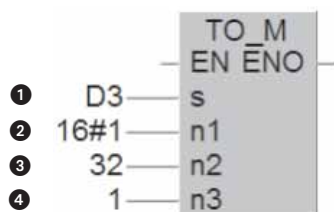
- ① Головной адрес специального функционального модуля на базовом шасси. Адрес может вводиться как десятичная или шестнадцатеричная константа (16#).
- ② Начальный адрес в буфере. Можно использовать константу или значение из регистра данных.
- ③ Количество передаваемых единиц данных
- ④ Адрес назначения данных в ЦП ПЛК

В приведенном выше примере команда FROM используется для передачи данных из специального функционального модуля с головным адресом X/Y040. Данная команда считывает содержимое буферного адреса 9 и записывает его в регистр данных D0.

Использование команды TO

Команда TO служит для передачи данных из ЦП ПЛК в буфер специального функционального модуля. Данная операция представляет собой копирование, поэтому содержимое данных в источнике не изменяется.

Ladder Diagram



MELSEC Instruction List

```
TO     H1  ②
        K32 ③
        D3  ①
        K1  ④
```

IEC Instruction List

```
FROM_M  D3, 16#1, 32, 1
        ①  ②  ③  ④
```

- ① Источник данных в ЦП ПЛК
- ② Головной адрес специального функционального модуля на базовом шасси. Адрес может вводиться как десятичная или шестнадцатеричная константа.
- ③ Starting address in buffer
- ④ Количество передаваемых единиц данных

В приведенном выше примере содержимое регистра данных D3 копируется в буферный адрес 32 специального функционального модуля с головным адресом 1 (X/Y010).

Прямое обращение к буферной памяти

К буферной памяти специального функционального модуля также можно обращаться напрямую, например, с помощью команды MOV.

При таком обращении специальный функциональный модуль может быть установлен на базовом шасси или шасси расширения, но только не на удаленной станции ввода/вывода.

Формат адреса устройства:

Uxxx\Gxxx

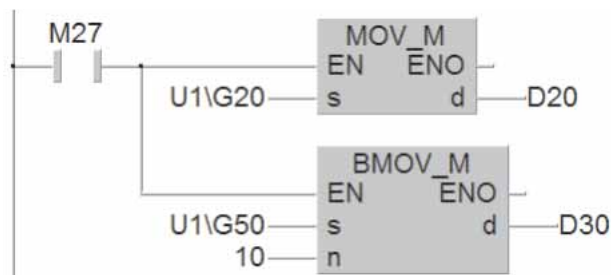
Головной адрес специального функционального модуля

Адрес буферной памяти

Например, адрес устройства U3\G11 обозначает адрес 11 буферной памяти в специальном функциональном модуле с головным адресом 3 (с X/Y30 по X/Y3F).

Когда в следующем примере реле M27 установлено, содержимое адреса 20 буферной памяти в специальном функциональном модуле с головным адресом 1 копируется в регистр данных D20. Затем содержимое адресов 50 и 59 буферной памяти копируется в регистры данных с D30 по D39.

Ladder Diagram



MELSEC Instruction List

```

LD      M27
MOV     U1\G20
        D20
MOV     U1\G50
        D30
        K10
  
```

IEC Instruction List

```

LD      M27
MOV_M   U1\G20, D20
BMOV_M  U1\G50, 10, D30
  
```

Автоматическая передача данных между ЦП ПЛК и специальными функциональными модулями

Для установки исходных данных и данных об условиях специальных функциональных модулей для пакета GX IEC Developer предусмотрено несколько дополнительных инструментальных средств. Данные инструменты упрощают конфигурирование специальных функциональных модулей и автоматическую передачу данных между ЦП ПЛК и этими модулями. Данные дополнительные программы обычно называются GX Configurator. Добавления к этому имени означают предназначение для конкретных специальных функциональных модулей.

Например, программа GX Configurator-AD предназначена для выполнения всех настроек модулей аналоговых выходов. При этом пользователю не требуется знать структуру буферной памяти. Параметры специальных функциональных модулей загружаются в ПЛК один раз вместе с программой, тем самым сокращая объем кода и обеспечивая более удобное параметрирование. Тем самым сокращается объем программы и уменьшается число источников.

В программе GX Configurator-AD можно также указать, например, место хранения измеренных значений в ЦП ПЛК. Затем передача этих данных будет осуществляться автоматически. При этом не требуются команды FROM/TO и вышеописанное прямое обращение к буферной памяти.

6.3 Команды сравнения

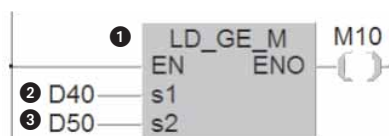
Проверка состояния битовых устройств, таких как входы и реле, обеспечивается с помощью основных логических команд, поскольку данные устройства могут иметь только два состояния: 0 и 1. Тем не менее, зачастую необходимо проверять содержимое устройств типа «слово» перед выполнением какой-либо операции, например, перед включением вентилятора охлаждения, когда температура превышает указанную уставку.

Для этого команду вывода или логическую операцию можно обуславливать зависимостью от результата операции сравнения. Помимо представленных здесь команд сравнения модули ЦП ПЛК серии System Q могут также сравнивать значения с плавающей десятичной запятой, блоки двоичных данных и символьные строки.

Для сравнения ниже приведены команды MELSEC и IEC.

Сравнение начала логической операции

Ladder Diagram



Данная команда идентична «записи» входа EN в формате Ladder Diagram. «TRUE» означает, что условие на входе выполняется всегда.

MELSEC Instruction List

```

1 LD>=      D40 2
              D50 3
OUT         M10

```

IEC Instruction List

```

LD         TRUE
1 LD_GE_M  D40, D50 2 3
ST         M10

```

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

Если состояние является истинным, состояние сигнала после сравнения устанавливается на 1. Состояние сигнала 0 означает ложный результат сравнения. В приведенном выше примере реле M10 устанавливается, когда содержимое регистра данных D40 больше или равно содержимому регистра D50.

Возможны следующие сравнения:

- Сравнение на «равно»: = (значение 1 = значение 2)
Команда IEC: EQ (Равно)

Выход данной команды устанавливается на «1» только тогда, когда идентичны значения обоих устройств.

- Сравнение на «больше чем»: > (значение 1 > значение 2)
Команда IEC: GT (Больше чем)

Выход данной команды устанавливается на «1» только тогда, когда первое значение больше второго.

- Сравнение на «меньше чем»: <W0> (значение 1 < значение 2)
Команда IEC: LE (Меньше чем)

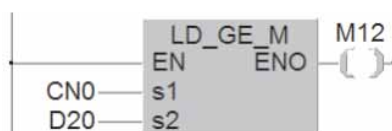
Выход данной команды устанавливается на «1» только тогда, когда первое значение меньше второго.

- Сравнение на «не равно»: $<>$ (значение 1 значение 2)
Команда IEC: NE (Не равно)
Выход данной команды устанавливается на «1» только тогда, когда два значения не равны.
- Сравнение на «меньше чем или равно»: $<=$ (значение 1 \leq значение 2)
Команда IEC: LE (Меньше или равно)
Выход данной команды устанавливается на «1» только тогда, когда первое значение меньше или равно второму.
- Сравнение на «больше чем или равно»: $>=$ (значение 1 \geq значение 2)
Команда IEC: GE (Больше или равно)
Выход данной команды устанавливается на «1» только тогда, когда первое значение больше или равно второму.

Для сравнения 32-разрядных данных к условию сравнения добавляется префикс «D» (для двойного слова) (например, LDD_EQ-M или LDD_GE_M).

Примеры:

Ladder Diagram



MELSEC Instruction List

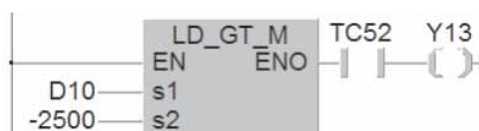
```
LD>=      C0
           D50
OUT       M12
```

IEC Instruction List

```
LD        TRUE
LD_GE_M   CN0, D20
ST        M12
```

Реле M12 устанавливается на «1», когда значение счётчика C0 больше или равно содержимому регистра D20.

Ladder Diagram



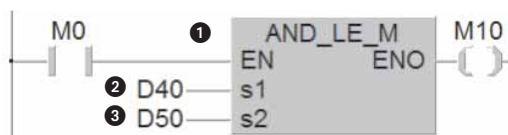
MELSEC Instruction List

```
LD>       D10
           K-2500
AND       T52
OUT       Y13
```

IEC Instruction List

```
LD        TRUE
LD_GT_M   D10, -2500
AND       TC52
ST        Y13
```

Выход Y13 включается, когда содержимое регистра D10 больше чем -2500, а время по таймеру T52 истекло.

Сравнение в качестве операции логического ИLadder DiagramMELSEC Instruction List

```
LD      M0
① AND<= D40 ②
        D50 ③
OUT     M10
```

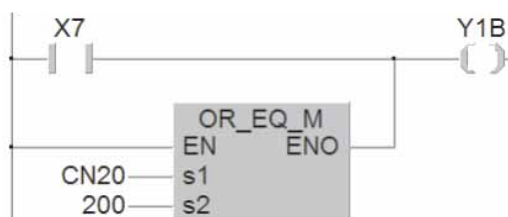
IEC Instruction List

```
LD      M0
① AND_GE_M D40, D50
        ② ③
ST      M10
```

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

Сравнение И можно использовать как обычную команду AND (см. раздел 4).

Варианты сравнения те же, что описаны выше для сравнения в начале операции. В приведенном выше примере реле M10 устанавливается, когда реле M0 установлено в «1» и содержимое регистра данных D40 меньше или равно содержимому регистра D50.

Сравнение в качестве операции логического ИЛИLadder DiagramMELSEC Instruction List

```
LD      X7
① OR=   C20 ②
        K200 ③
OUT     Y1B
```

IEC Instruction List

```
LD      X7
① OR_EQ_M CN20, 200
        ② ③
ST      Y1B
```

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

Сравнение ИЛИ можно использовать как обычную команду OR (см. раздел 4). В данном примере выход Y1B устанавливается, когда включен вход X7 или текущее значение счётчика C20 достигло «200».

6.4 Команды математических функций

Все контроллеры System Q способны выполнять четыре основных арифметических операции: сложение, вычитание, умножение и деление. Команды математических функций MELSEC предусмотрены для операций с двоичными значениями, блоками двоичных данных, двоично-десятичными кодами и символьными строками.

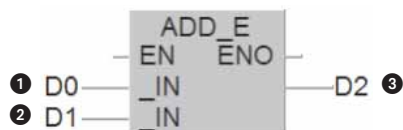
При редактировании программы с помощью пакета GX IEC Developer в формате Ladder Diagram или IEC Instruction List можно использовать дополнительные команды IEC. В данном разделе приведены только команды IEC. Подробное описание команд MELSEC приведено в руководстве по программированию для серий A/Q и System Q, кат. № 87431.

Команды IEC для сложения, вычитания, умножения и деления могут применяться к таким типам данных, как INT (16-разрядные целые), DINT (32-разрядные целые) и REAL (значения с плавающей десятичной запятой). Данные команды нельзя напрямую применять к устройствам DINT и REAL – устройства следует объявить вначале глобальными переменными (см. раздел 4.6.2).

6.4.1 Сложение

Команда ADD служит для вычисления суммы двух значений и записи результата в другое устройство.

Ladder Diagram



IEC Instruction List

LD	D0	①
ADD	D1	②
ST	D2	③

- ① Первое устройство-источник или константа
- ② Второе устройство-источник или константа
- ③ Устройство, в котором сохраняется результат сложения

В приведенном выше примере складывается содержимое D0 и D1 и результат записывается в D2.

Примеры

Прибавление 1000 к содержимому регистра данных D100.



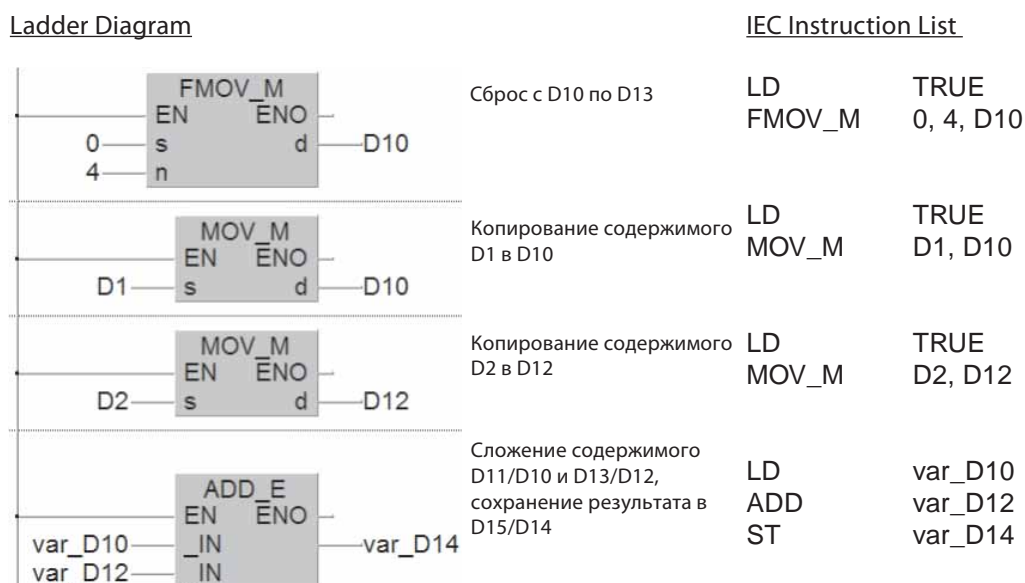
При необходимости результат можно также записать в одно из устройств-источников. Однако необходимо помнить, что в этом случае результат будет изменяться в каждом программном цикле, если команда ADD выполняется циклически.



Команда ADD учитывает знаки значений (например, $10 + (-5) = 5$).

Типы данных входных и выходных переменных команды ADD должны быть идентичными. Это может вызывать проблемы, когда результат сложения выходит за пределы диапазона значений данных переменных. Например, при сложении двух целых чисел (16-разрядных) «32700» и «100» сохраняется не ожидаемый результат «32800», а значение «-32736», поскольку максимальное значение 16-разрядной переменной составляет «32767». Переполнение интерпретируется как отрицательное значение, что приводит к неверному результату.

Единственным возможным решением является копирование складываемых значений перед сложением в 32-разрядные переменные. После этого производится сложение 32-разрядных переменных.

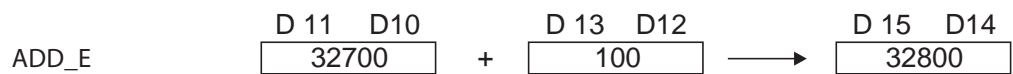
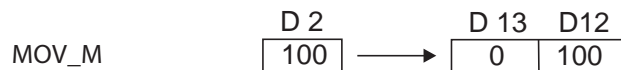
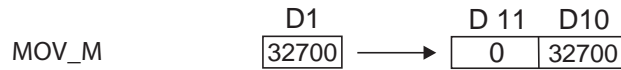
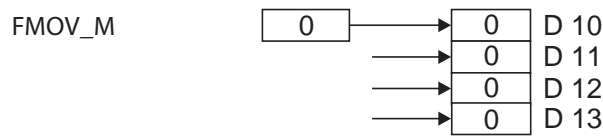


Поскольку невозможно напрямую определить 32-разрядные устройства как входные и выходные переменные команды ADD, требуется объявление глобальных переменных.

Global Variable List					
	Class	Identifier	MIT-Addr.	IEC-Addr.	Type
0	VAR_GLOBAL	var_D10	D10	%M00.10	DINT
1	VAR_GLOBAL	var_D12	D12	%M00.12	DINT
2	VAR_GLOBAL	var_D14	D14	%M00.14	DINT

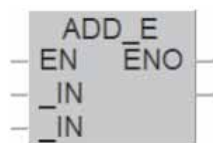
Имена для переменных (идентификаторы) выбираются произвольно. Для простоты в данном примере используются адреса устройств.

При вышеуказанных значениях содержимое регистров данных во время выполнения данной команды изменяется следующим образом.

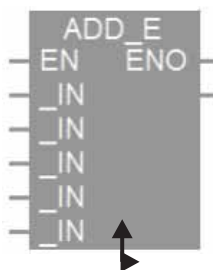


Регистр D14 с двойным словом содержит правильный результат данной операции сложения.

Применение команды ADD не ограничивается двумя входными переменными. Можно задавать до 28 входных переменных. В формате Ladder Diagram это осуществляется следующим образом.



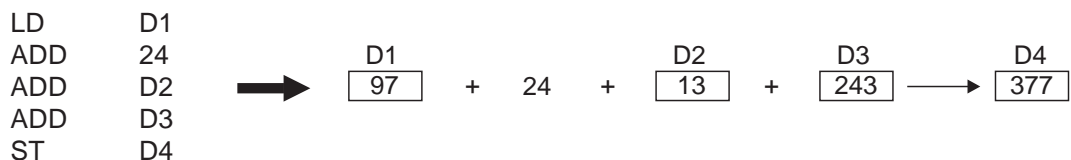
В окне Function Block Selection (Выбор функционального блока) выберите команду ADD-E (см. раздел 4.7.7) и вставьте данную команду в тело программного модуля.



Щёлкните команду. Цвет поля изменится. Переместите курсор вниз, чтобы он превратился в двойную стрелку.

Затем щёлкните левой кнопкой мыши и перетащите курсор вниз так, чтобы отобразилось нужное количество входных переменных. В данной точке отпустите левую кнопку мыши.

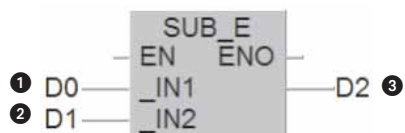
При программировании в формате IEC Instruction List нужно просто несколько раз ввести команду ADD. Пример:



6.4.2 Вычитание

Команда SUB служит для вычисления разницы между двумя числовыми значениями (содержимым 16- или 32-разрядных устройств или константами). Результат вычитания записывается в третье устройство.

Ladder Diagram



IEC Instruction List

LD	D0	①
SUB	D1	②
ST	D2	③

- ① Уменьшаемое (от данного значения отнимается вычитаемое)
- ② Вычитаемое (данное значение отнимается от вычитаемого)
- ③ Разность (результат вычитания)

Типы данных входных и выходных переменных команды SUB должны быть идентичными.

В приведенном выше примере содержимое D1 вычитается из содержимого D0 и разность записывается в D2.

Примеры

Вычитание значения «100» из содержимого регистра данных D100 и запись результата в D101, когда установлено реле M37.



Команда SUB учитывает знаки значений.

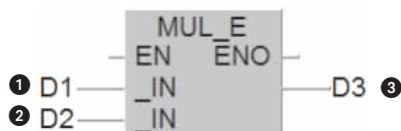


При необходимости результат можно также записать в одно из устройств-источников. Однако необходимо помнить, что в этом случае результат будет изменяться в каждом программном цикле, если команда SUB выполняется циклически.

6.4.3 Умножение

Команда MUL служит для вычисления произведения двух 16- или 32-разрядных значений и записи результата в третье устройство.

Ladder Diagram



IEC Instruction List

LD	D1	①
MUL	D2	②
ST	D3	③

- ① Множимое
- ② Множитель
- ③ Произведение (результат умножения, множимое x множитель = произведение)

В приведенном выше примере умножается содержимое D1 и D2 и результат записывается в D3.

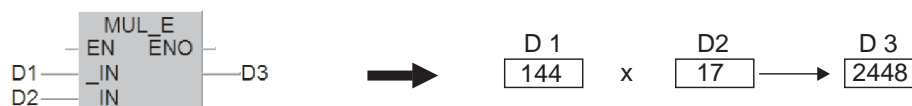
ПРИМЕЧАНИЕ

Типы данных входных и выходных переменных команды MUL должны быть идентичными. Когда результат умножения выходит за пределы диапазона значений для 16- или 32-разрядных переменных, наиболее значащие биты теряются и произведение становится неверным. Если требуется умножить 16-разрядные значения, их можно скопировать в 32-разрядные переменные, как указано в описании команды ADD (см. раздел 6.4.1). После этого производится умножение 32-разрядных переменных с помощью команды MUL и получается правильный результат.

Для команды MUL можно задавать до 28 входных переменных. Задание аналогично команде ADD (см. раздел 6.4.1).

Примеры

Умножение содержимого D1 и D2 и сохранение произведения в D3.



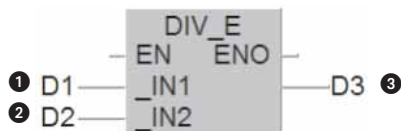
Команда MUL учитывает знаки значений. В данном примере значение D10 умножается на константу -5.



6.4.4 Деление

Команда DIV служит для деления одного числа на другое.

Ladder Diagram



IEC Instruction List

```
LD    D1  ①
DIV   D2  ②
ST    D3  ③
```

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

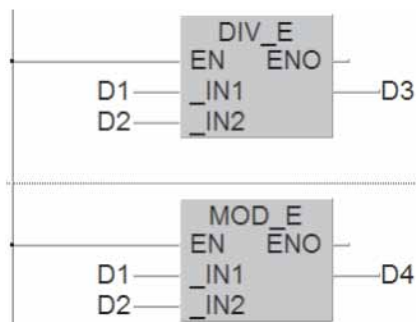
В приведенном выше примере содержимое регистра данных D1 делится на содержимое D2 и результат сохраняется в D3.

ПРИМЕЧАНИЕ

Делитель не должен быть равен 0. Деление на 0 невозможно и приводит к ошибке, при которой останавливается ЦП ПЛК. (Это может произойти когда, как показано в приведенном выше примере, деление выполняется с содержимым регистров данных, обнулённых операцией RESET. Во избежание остановки ПЛК регистр данных с делителем должен быть установлен в программе ПЛК на определенное значение перед выполнением команды DIV.)

Типы данных входных и выходных переменных команды DIV должны быть идентичными. При делении целых значений (INT или DINT) частное также будет целой величиной. Для определения остатка можно использовать команду MOD.

Ladder Diagram



IEC Instruction List

```
LD    D1
DIV   D2
ST    D3

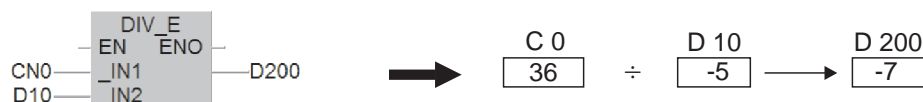
LD    D1
MOD   D2
ST    D4
```

Входные переменные команды MOD те же, что и для команды DIV. В приведенном выше примере содержимое регистра данных D1 делится на содержимое D2. Частное сохраняется в D3, а остаток в D4.

$$\begin{matrix} D1 \\ \boxed{40} \end{matrix} \div \begin{matrix} D2 \\ \boxed{6} \end{matrix} \longrightarrow \begin{matrix} D3 \\ \boxed{6} \end{matrix} \quad \text{Частное (6 x 6 = 36) (Выход команды DIV)}$$

$$\begin{matrix} D4 \\ \boxed{4} \end{matrix} \quad \text{Остаток (40 - 36 = 4) (Выход команды MOD)}$$

Команда DIV учитывает знаки значений. В данном примере значение счётчика C0 делится на значение в D10.

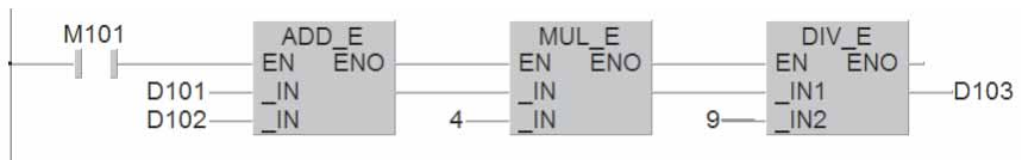


6.4.5 Комбинирование команд математических функций

На практике редко можно обойтись каким-то одним вычислением. Для решения сложных вычислительных задач команды математических функций можно легко комбинировать.

В следующем примере показано вычисление суммы значений в регистрах данных D101 и D102, затем умножение результата на 4 и деление полученного произведения на 9. Результат данного вычисления сохраняется в регистре данных D103.

Ladder Diagram



Указатель

A-Z

ASCII-код	
Символьная строка	5-14
Обзор	4-6
CANopen	3-39
CC-Link	3-39
DeviceNet	3-39
ETHERNET	3-38
Function Block Diagram	4-9
IEC61131-3	4-10
GX Configurator	6-21
GX IEC Developer	
Объявление переменных	4-11
IEC61131-3	4-10
Создание проекта	4-35
Языки программирования	4-7
Instruction List	4-7
Ladder Diagram	
Ввод функций	4-23
MELSECNET	3-40
Обзор	4-8
Q64TCRT	3-34
Q64TCRTBW	3-34
Q64TCTT	3-34
Q64TCTTBW	3-34
QD51	3-36
QD62	3-34
QD75	3-35
QJ61BT11	3-42
QJ71AS92	3-43
QJ71BR11	3-41
QJ71C24	3-35
QJ71DN91	3-43
QJ71E71	3-41
QJ71LP21	3-41
QJ71PB92D	3-42
QJ71PB93D	3-42
QJ71WS96	3-44
Sequential Function Chart	4-9
Structured Text	4-7
SFC	
Обзор	4-9

A

Автоматическое отключение	4-33
---------------------------	------

Б

Базовое шасси	
Определение	3-1
Обзор	3-3
Базовое шасси	3-3
Бесконтактные датчики	3-19
Блокировка	
контактов	4-32
Буферная память	6-18

В

Выход ENO	4-8
Восьмеричные числа	4-4
Вход EN	4-8

Г

Головной адрес	6-19
----------------	------

Д

Двоично-десятичный код	4-5
Двоичные числа	4-2

З

Задержка выключения	5-17
Задержка выключения	5-17
Запоминающие таймеры	5-7
Заголовок (программного модуля)	4-10
Значения с плавающей десятичной запятой	5-14

И

Интерфейс AS	3-39
Определение	4-11
Пример назначения	4-37
Использование в программе	4-39
Импульсный запуск выполнения	4-22

К

Кабели расширения	
Определение	3-1
Обзор	3-3
Карты памяти	3-14
Команда	
ADD (команда IEC)	6-25

ANB	4-20	Команда INV	4-29
AND	4-17	Команда MEF	4-31
ANDF	4-22	Команда MEP	4-31
ANDN	4-17	Команда MOD	6-30
ANDP	4-22	Команда MOV	6-12
ANI	4-17	Команда MUL	6-29
BMOV	6-16	Установка/Сброс	4-25
DIV (команда IEC)	6-30	Команда OR	4-18
FF	4-30	Команда ORB	4-20
FMOV	6-17	Команда ORI	4-18
FROM	6-20	Команда ORN	4-18
INV	4-29	Команда ORP/ORF	4-22
LD	4-14	Команда OUT	4-14
LDF	4-22	Команда R	4-25
LDI	4-14	Команда SUB	6-28
LDP	4-22	Константы	
MEF	4-31	Команда FMOV	6-17
MEP	4-31	Команда FROM	6-20
MOD (команда IEC)	6-30	Символьная строка	5-14
MOV	6-12	Десятичные	5-14
MUL (команда IEC)	6-29	Шестнадцатеричные	5-14
OR	4-18	Действительные числа	5-14
ORB	4-20	Команда LD	4-14
ORF	4-22	Команда LDI	4-14
ORN	4-18	Команда LDP/LDF	4-22
ORP	4-22	Команда PLF	4-28
OUT	4-14	Команда PLS	4-28
PLF	4-28	Команда S	4-25
PLS	4-28	Команда RST	4-25
R	4-25	Команда SET	4-25
RST	4-25	Конфигурация сигнала	
S	4-25	Отрицание	4-29
SET	4-25		
TO	6-20		
Команда ANB	4-20		
Команда ANDN	4-17		
Команда ANDP/ANDF	4-22		
Команда ANI	4-17		
Команда BMOV	6-16		
Команда DIV	6-30		
Команда FF	4-30		
Команды IEC			
ADD	6-25		
DIV	6-30		
MOD	6-30		
MUL	6-29		
SUB	6-28		
Команда INV	4-29		
Команда MEF	4-31		
Команда MEP	4-31		
Команда MOD	6-30		
Команда MOV	6-12		
Команда MUL	6-29		
Установка/Сброс	4-25		
Команда OR	4-18		
Команда ORB	4-20		
Команда ORI	4-18		
Команда ORN	4-18		
Команда ORP/ORF	4-22		
Команда OUT	4-14		
Команда R	4-25		
Команда SUB	6-28		
Константы			
Команда FMOV	6-17		
Команда FROM	6-20		
Символьная строка	5-14		
Десятичные	5-14		
Шестнадцатеричные	5-14		
Действительные числа	5-14		
Команда LD	4-14		
Команда LDI	4-14		
Команда LDP/LDF	4-22		
Команда PLF	4-28		
Команда PLS	4-28		
Команда S	4-25		
Команда RST	4-25		
Команда SET	4-25		
Конфигурация сигнала			
Отрицание	4-29		
Л			
Локальные переменные			
назначение в процессе программирования	4-41		
Определение	4-11		
Локальные переменные			
назначение в процессе программирования	4-41		
М			
Меры предосторожности на случай обрыва			
кабелей	4-32		
Многопроцессорный режим работы ЦП	3-2		
Модуль веб-сервера	3-44		
Запись данных			
Модули выходов			
Обзор	3-24		

Релейные	3-25
Транзисторные	3-28
Транзисторные (переключающие на минус)	3-28
Транзисторные (переключающий на плюс)	3-28
Тиристорные	3-26
Модули транзисторных выходов	3-28
Модули тиристорных выходов	3-27
Модули питания	
Обзор	3-5
критерии выбора	3-6
Модули ЦП управления процессом	3-7
Модули высокоскоростных счётчиков	3-34
Модуль входа	
для входа перем. тока	3-22
общий минус	3-19
общий плюс	3-21
Модули измерения температуры	3-32
Модули регулирования температуры	3-34
Модули ETHERNET	3-41
Модуль DeviceNet	3-43
Модули ЦП	
Батарея	3-15
Карты памяти	3-14
Обзор	3-7
Модули ЦП ПЛК	3-8
Переключатель RUN/STOP	3-11
Системные переключатели	3-11
Модули аналоговых входов	
Назначение	3-31
Обзор	3-17
Модули аналоговых выходов	
Назначение	3-33
Обзор	3-24
Модули PROFIBUS/DP	3-42
Модули MELSECNET	3-41
Модули ЦП управления движением	3-7
Модули ЦП ПЛК	3-7
Модули позиционирования	3-35
Модули релейных выходов	3-25
Модуль выхода, переключающий на минус	3-30
Модуль выхода, переключающий на плюс	3-28

Н

на станцию интеллектуальных устройств (CC-Link)	6-9
---	-----

О

Обзор	3-3
Обработка образа процесса	2-2
Обратная связь для выходного сигнала	4-33
Оптические датчики	3-19
Определение	

П

Падающий фронт	4-22
Переменные	4-11
Программный модуль	
Тело	4-10
Заголовок	4-10
Пример программирования	
Свёртывающаяся шторная дверь	4-34
Генератор тактового сигнала	5-20
Выключение с задержкой	5-6
Задание значений уставок таймеров и счётчиков	5-15
Программная команда	4-1

Р

Разрешение (аналоговых модулей)	3-31
Резисторный термометр	3-32
Режим линии (GX IEC Developer)	4-41
Регистры данных	5-11
Резисторные термометры Pt100	3-32
PROFIBUS/DP	3-39

С

Сетевой модуль CC-Link	3-42
Служебные программы	6-21
Специальные функциональные модули	
Обмен данными с ЦП ПЛК	6-18
прямое обращение	6-21
Головной адрес	6-19
Специальные регистры	5-12
Специальные реле	5-5
Сетевые модули	
AS-интерфейс	3-43
CC-Link	3-42
DeviceNet	3-43
ETHERNET	3-41
MELSECNET	3-41
PROFIBUS/DP	3-42
Счётчик	
Функции	5-9

Модули	3-34
Косвенное задание значений уставок	5-15

Т

Таймеры	5-6
Термопары	3-32
Тело (программного модуля)	4-10

У

Установка/Сброс	4-25
Устройство	
Адрес	4-1
Обзор счётчиков	5-10
Обзор регистров данных	5-12
Обзор файловых регистров	5-13
Обзор входов/выходов	5-3
Имя	4-1
Обзор реле	5-4
Обзор таймеров	5-8

Устройства аварийного останова	4-32
--	------

Ф

Фиксируемые реле	5-4
Функции	4-23

Ч

Чтение данных	
с другого ПЛК (CC-Link)	6-9
из станции интеллектуальных устройств (CC-Link)	6-9

Ш

Шасси расширения	
Определение	3-1
Обзор	3-3
Шасси расширения	
Определение	3-1
Шестнадцатеричные числа	4-3

MITSUBISHI ELECTRIC EUROPE B.V. /// РОССИЯ /// Москва /// Космодамианская наб., 52, стр. 5
Тел.: +7 495 721 20 70 /// Факс: +7 495 721 20 71 /// automation@mitsubishielectric.ru /// www.mitsubishi-automation.ru