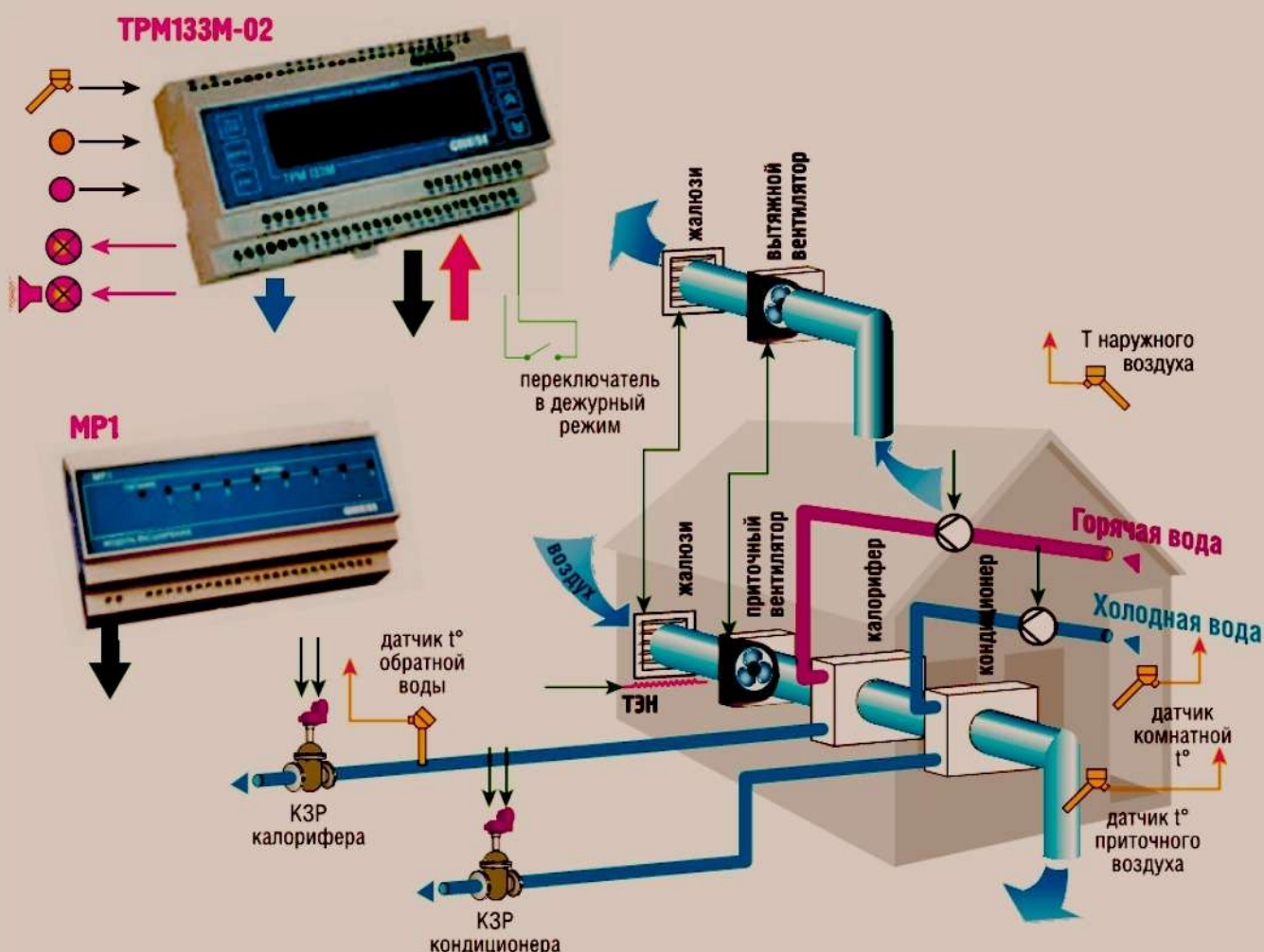


О.И. Максимычев  
А.В. Либенко  
В.А. Виноградов



# ПРОГРАММИРОВАНИЕ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ (PLC)

Учебное пособие



ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ

О.И. МАКСИМЫЧЕВ, А.В. ЛИБЕНКО  
В.А. ВИНОГРАДОВ

# **ПРОГРАММИРОВАНИЕ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ (PLC)**



*Учебное пособие*



МОСКВА  
МАДИ  
2016

УДК 32.96:32.97 С78  
ББК 681.58:681.32  
М171

*Рецензенты:*

*Остроух А.В.* – д-р техн. наук, проф. каф. АСУ,  
декан факультета «Управление» МАДИ;

*Попов Д.И.* – д-р техн. наук, проф.,  
заведующий каф. «Информатика и информационные технологии»  
ФГБОУ ВПО «Московский государственный  
университет печати имени Ивана Фёдорова».

**Максимычев, О.И.**

М171 Программирование логических контроллеров (PLC): учеб.  
пособие / О.И. Максимычев, А.В. Либенко, В.А. Виноградов. – М.:  
МАДИ, 2016. – 188 с.

В учебном пособии рассмотрены вопросы, связанные с методами программирования логических контроллеров (ПЛК/PLC), которые применяются для автоматизации технологических процессов на транспорте, в строительном комплексе и в промышленном производстве. Рассмотрен инструментальный синтез программного обеспечения с применением интегрируемой среды разработки LOGO Soft Comfort на языке функциональных блоков и диаграмм стандарта МЭК 61131-3.

Пособие предназначено для студентов, изучающих дисциплины «Микропроцессорные системы», «Проектирование микропроцессорных систем», «Аппаратно-программные комплексы», «Вычислительные машины системы и сети», «Вычислительная техника и сети в отрасли», «Эксплуатация информационных систем на транспорте», «Инструментальные средства информационных систем на транспорте» по направлениям подготовки 090301 – Информатика и вычислительная техника, 090302 – «Информационные системы и технологии», 150304 – «Автоматизация технологических процессов и производств», 190600.2 – «Эксплуатация транспортно-технологических машин и комплексов», 130302 и 1304021 – «Электрооборудование автомобилей и тракторов». Материалы пособия могут быть использованы инженерами и аспирантами, занимающимися задачами автоматизации технологических процессов, автоматизированного управления и мониторинга.

УДК 32.96:32.97 С78  
ББК 681.58:681.32

© МАДИ, 2016

## ВВЕДЕНИЕ

Программируемые логические контроллеры (ПЛК/PLC) являются основой автоматизации технологических процессов. Применение технологий автоматизации, основанных на ПЛК, значительно ускоряют осуществление проектов и дают возможность значительно увеличивать жизненный цикл базовых производств и технологий, основываясь на международном стандарте IEC 61131. Использование ПЛК в системах управления позволяет создавать проекты автоматизации открытого типа, способных интегрировать различные компоненты комплекса технических и программных средств на основе единого стандарта. Компонента программного обеспечения является самым гибким инструментом автоматизации, позволяющим изменять алгоритм управления без конструктивных изменений всей системы. Изучение основ программирования ПЛК (PLC) в учебном пособии производится на примере программной инструментальной среды LogoSoftComfort 7.x (LSC), разработанной немецким концерном Siemens, признанным мировым лидером в области исследований, разработки и производства комплексных систем средств автоматизации.

Среда разработки LSC реализует возможность программирования на двух языках: функциональных блоков и диаграмм (FBD, Function Block Diagram) и релейно-контактных схем (LD, Ladder Diagram). Изучение основ технологии программирования систем управления и регулирования производится комплексно согласно техническим и методическим рекомендациям разработчиков LSC.

Комплекс LSC имеет аналоги и у других разработчиков производителей систем автоматизации, что делает изучаемые принципы и технологии программирования универсальными, служащими основой для освоения более универсальных и высокопроизводительных систем, во всех сферах современных производств и мониторинга, в том числе в строительстве и на транспорте, поддерживаемых стандартом МЭК 61131-3.

## Глава 1. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ

### 1.1. Основные характеристики ПЛК

Под понятием – программируемые логические контроллеры ПЛК (Programmable logic controllers, PLC) подразумеваются блочно-модульные микропроцессорные системы универсального и общего назначения, предназначенные для автоматизации в различных областях промышленности, техники и сферах инженерной деятельности.

Обобщённая структурная схема ПЛК показана на рис. 1.1. Аппаратный комплекс ПЛК конструктивно выполняется в виде единого модуля, содержащего плату микропроцессорной системы с возможностью расширения через разъёмы локальной шины и/или сетевого интерфейса. Расширительные модули (платы) входят в технически регламентированный состав устройств связи с объектом (УСО), осуществляя нормализацию и предобработку аналоговых (AI/AO) и дискретных (DI/DO) сигналов ввода/вывода (I/O) для ПЛК.

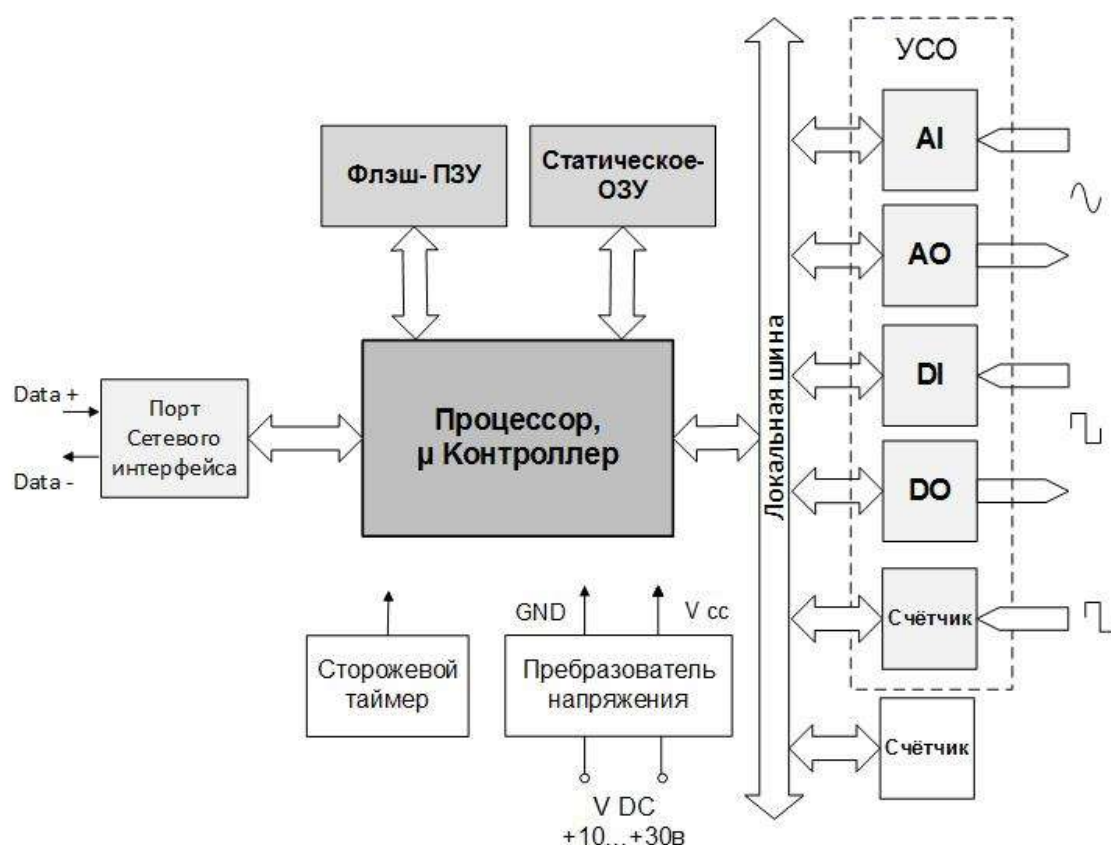


Рис. 1.1. Обобщённая структурная схема ПЛК (PLC)

Микропроцессорная система программируемого контроллера содержит:

1) центральный процессор ЦП (CPU) в основном выполняется на 32-разрядных процессорах. В некоторых семействах могут быть вы-

полнены на базе однокристальных микроЭВМ (ОМЭВМ или микроконтроллеры);

2) память программ на основе флэш памяти постоянно-запоминающего устройства (ПЗУ), служащего для хранения исполняемых программ пользователя и долговременного хранения данных в защищённой области;

3) блок оперативно-запоминающего устройства памяти (ОЗУ);

4) сетевой интерфейс, он может быть встроенным или заменяемым в виде отдельного модуля;

5) источник питания модуля ПЛК, он может быть встроенным или выполнен в виде отдельно устанавливаемого блока.

Основное общепромышленное напряжение питания систем ПЛК контрольно-измерительных приборов и автоматики (КИПиА) составляет 24 в постоянные тока (VDC).

**ПЛК (PLC)** – это программно-управляемый дискретный автомат (рис. 1.2), имеющий некоторое множество входов, подключенных посредством датчиков к объекту управления, и множество выходов, подключенных к исполнительным устройствам. ПЛК контролирует состояния входов и вырабатывает определённые последовательности программно заданных действий, отражающихся в изменении выходов.



Рис. 1.2. Принцип функционирования логического контроллера

Детальная регламентация всех составляющих ПЛК содержится в нормативном документе – ГОСТ Р 51840-2001. «Программируемые контроллеры. Общие положения и функциональные характеристики».

### 1.1.1. Рабочий цикл ПЛК

Рабочий цикл ПЛК состоит из фаз, показанных на рис. 1.3:

1. Начало цикла.
2. Чтение состояния входов.
3. Выполнение кода программы пользователя.
4. Запись состояния выходов.
5. Обслуживание аппаратных ресурсов ПЛК.



6. Монитор системы исполнения.
7. Контроль времени цикла.
8. Переход на начало цикла.



Рис. 1.3. Рабочий цикл ПЛК (PLC)

### 1.1.2. Время реакции

**Время реакции** – это время, прошедшее с момента изменения состояния системы до момента выработки соответствующей реакции.

Очевидно, для ПЛК время реакции зависит от распределения момента возникновения события и начала фазы чтения входов.

Если изменение значений входов произошло непосредственно перед фазой чтения входов, то время реакции будет наименьшим и равным времени сканирования.

Когда изменение значений входов происходит сразу после фазы чтения входов, тогда время реакции будет **наибольшим**, равным удвоенному времени сканирования минус время одного чтения входов.

Следовательно, время реакции ПЛК не превышает удвоенного времени сканирования. Таким образом, на время обработки данных контроллером влияют:

- 1) время цикла исполнения;
- 2) время цикла сервисных операций;
- 3) время реакции системной шины на операциях:
  - Чтение/запись;
- 4) Задержки интерфейсов:
  - дискретные сигналы (DI) → замедление фильтрацией на входе;
  - аналоговые сигналы (AI) → время преобразования;
  - дискретные выходные сигналы (DO) → время срабатывания;
  - аналоговые выходные сигналы (AO) → время преобразования.

### 1.1.3. Время цикла сканирования

Время цикла сканирования является базовым показателем быстродействия ПЛК. Принято, что при измерении времени рабочего цикла пользовательская программа должна содержать 1К логических ко-

манд. Для ПЛК, поддерживающих стандарт МЭК 61131-3, используют команды на языке IL. Иногда изготовители приводят несколько значений времени цикла, полученных при работе с переменными различной разрядности. Ориентировочно о скорости обработки различных типов, данных можно судить по тактовой частоте и разрядности центрального процессора. Стоит заметить, что восьмиразрядные ПЛК не редко оказываются быстрее 32-разрядных при выполнении битовых операций. Объясняется это тем, что 8-разрядных микропроцессорах более распространена аппаратная поддержка работы с битами. Так, в РС-совместимых процессорах для выделения бита приходится использовать логические команды и циклический сдвиг.

#### ***1.1.4. Контроль времени сканирования***

Правильно составленная пользовательская программа не должна содержать бесконечных циклов (и/или циклов, увеличивающих время реакции системы). В противном случае управление системе исполнения не будет передано и соответственно нормальное функционирование контроллера будет нарушено. Для преодоления данной проблемы служит контроль времени цикла. Контроль осуществляется при поддержке аппаратно реализованного «сторожевого таймера». Если фаза пользовательского кода выполняется дольше установленного порога, то ее работа будет прервана. Таким образом, достигается предсказуемое поведение ПЛК при ошибках в программе и при «зависании» по причине аппаратных сбоев.

Обслуживание сторожевого таймера выполняется в рабочем цикле ПЛК. Выполнять эту операцию по прерыванию нельзя, поскольку при «зависании» процессора система прерываний достаточно часто продолжает исправно работать.

#### **Определение времени цикла программы**

Программный цикл представляет собой выполнение всей коммутационной программы, т.е. в первую очередь считывание входных сигналов, обработку коммутационной программы и последующий вывод выходных значений.

**Время цикла** – время, необходимое для однократного полного выполнения коммутационной программы.

Время одного программного цикла можно определить с помощью короткой тестовой программы. Тестовая программа создаётся в модуле ПЛК и возвращает значение в процессе выполнения в режиме ввода параметров, из которого определяется текущее время цикла.

#### ***1.1.5. Режим реального времени***

В системах реального времени помимо правильности решения определяющую роль играет время реакции. Логически верное ре-



шение, полученное с задержкой более допустимой, не является приемлемым.

В системах жесткого реального времени существует определённый временной порог. При его превышении наступают необратимые катастрофические последствия.

В системах мягкого реального времени характеристики системы ухудшаются с увеличением времени управляющей реакции. Система может работать плохо или еще хуже, но ничего катастрофического при этом не происходит.

### Вопросы для самоконтроля

1. Дайте определение программируемого контроллера?
2. С помощью, каких средств может быть записана программа пользователя в ПЛК?
3. Назовите основные функциональные блоки аппаратного комплекса ПЛК. Перечислите их функциональное назначение.
4. Каков состав цикла работы ПЛК?
5. Как определяется время цикла ПЛК?
6. Для чего используется сторожевой таймер в структуре ПЛК?
7. Что такое режим реального времени?

### 1.2. Введение в стандарт МЭК 61131-3

В 1979 году в рамках **Международной электротехнической комиссии (МЭК)** была создана специальная группа технических экспертов по проблемам программируемых логических контроллеров (ПЛК), включая аппаратные средства, монтаж, тестирование, документацию и связь.

Первый вариант стандарта был опубликован в 1982 году. Ввиду сложности получившегося документа было решено разбить его на несколько частей. В настоящее время стандарт МЭК 61131 включает в себя следующие части:

- Часть 1. Общая информация;
- Часть 2. Требования к оборудованию и тестам;
- Часть 3. Языки программирования;**
- Часть 4. Руководства пользователя;
- Часть 5. Спецификация сообщений;
- Часть 6. Промышленные сети;
- Часть 7. Программирование с нечеткой логикой;
- Часть 8. Руководящие принципы применения и реализации языков ПЛК.

Стандарт МЭК 61131-3 описывает 5 языков программирования, явившихся результатом изучения наиболее удачных фирменных раз-

работок мировых лидеров рынка ПЛК. Языки ПЛК весьма оригинальны и существенно отличаются от известных языков программирования для компьютеров. Рассмотрим два языка, получившие признание инженеров в области автоматизации технологических процессов.

Развитие идеи программной реализации электрических схем привело к появлению языка **функциональных блоков и диаграмм (ФБД), FBD (Function Block Diagram)**. Диаграмма FBD очень напоминает принципиальную схему электронного устройства на микросхемах. Выходы экземпляров функциональных блоков могут быть поданы на входы других блоков либо непосредственно на выходы ПЛК. Сами блоки, представленные на схеме как функциональные модули, могут выполнять стандартные и специальные функции.

FBD схемы ясно отражают взаимосвязь входов и выходов диаграммы. Если алгоритм хорошо описывается с позиции сигналов, то его FBD – представление всегда получается значительно нагляднее, чем в текстовых языках.

Популярностью среди прикладных ПЛК программистов пользуется **язык релейно-контактных схем (РКС) или релейных диаграмм LD (Ladder Diagram)** – графический язык, реализующий структуры электрических цепей.

Графически LD диаграмма представлена в виде двух вертикальных шин питания. Между ними расположены цепи, образованные соединением контактов. Контакт может выступать вход ПЛК или некоторая промежуточная переменная. Нагрузкой каждой цепи служит реле. Каждое реле имеет контакты, которые можно использовать в других цепях. Логически последовательное (И), параллельное (ИЛИ) соединение контактов и инверсия (НЕ) образуют базис Буля.

LD идеально подходит для реализации комбинационных логических схем и конечных автоматов. Благодаря возможности включения в LD функций и функциональных блоков, выполненных на других языках, сфера применения практически не ограничена.

Популярность LD объясняется его наглядностью. Специалист, владеющий основами схемотехники, без особого труда разберется в работе LD диаграммы.

Включенный в стандарт язык **IL (Instruction list) – типичный ассемблер** с аккумулятором и переходами по меткам. Набор инструкций стандартизован и не зависит от конкретной целевой платформы. Язык IL позволяет работать с любыми типами данных, вызывать функции и функциональные блоки, реализованные на любом языке. На IL можно описать алгоритм любой сложности, хотя текст иногда бывает достаточно громоздким.

**Язык ST (Structured Text)** является текстовым языком высокого уровня. ST представляет собой несколько адаптированный язык Пас-

каль. Для специалистов знакомых с программированием на языках С и Паскаль, освоение ST не вызовет сложностей.

Среди всех IEC языков, пожалуй, самым интересным являются **SFC (Sequential Function Chart) диаграммы**. Своим происхождением SFC обязан **сетям Петри**. Графическая диаграмма SFC состоит из шагов и переходов между ними. Разрешение перехода определяется условием. С шагом связаны определенные действия. Описания действий выполняются на любом языке. Сам SFC не содержит каких-либо управляющих команд.

Действия могут быть описаны и в виде вложенной SFC – схемы.

Диаграммы SFC являются великолепным инструментом быстрого проектирования систем управления, позволяют создавать наглядные и эффективные модели.

Даже столь краткое представление МЭК языков показывает очевидную сложность полноценной реализации инструмента программирования, поддерживающего все пять языков ПЛК. Однако стандарт не обязывает разработчиков ПЛК реализовывать все языки программирования. Обычно небольшие фирмы-изготовители ПЛК реализуют несколько или один единственный язык. Во многом приоритет использования того или иного языка программирования зависит от исторически сложившихся традиций в отрасли или сферы применения.

Рассмотрим два языка FBD (ФБД) и LD (РКС) получивших признание инженеров в области автоматизации технологических процессов.

### Вопросы для самоконтроля

1. Какой документ является регламентирующим развитие PLC?
2. Из каких частей состоит регламентирующий документ?
3. Какие языки программирования применяются для программирования PLC?
4. Какое программное обеспечение пишется на языке С для PLC?

### 1.3. Инструментальная среда разработки программ

Стандартизация языков программирования и модели ПЛК привела к появлению специализированных фирм, занятых проектированием систем программирования для ПЛК. Известность среди систем программирования ПЛК получил пакет инструментов программирования – Logo Soft Comfort (LSC), разработчик – электротехнический концерн Siemens.

Структура таких инструментов разделяется на две части: систему исполнения и рабочее место программиста. Система исполнения размещается в ПЛК, обеспечивает загрузку и исполнение кода, отладочные функции, содержит драйверы ввода-вывода. Как правило, систе-

ма исполнения написана на языке С и достаточно легко может быть адаптирована для работы в любом ПЛК.

Рабочее место программиста включает специализированные текстовые и графические редакторы (FBD, LD), средства быстрого ввода, отладки, трассировки и визуализации модели объекта управления, документирования, управления версиями. В силу такого разделения программное обеспечение рабочего места программиста оказывается практически независимым от аппаратной платформы. В простейшем случае, компьютер рабочего места программиста связан с ПЛК через интерфейс USB или через сетевой интерфейс (рис. 1.4).

Станция программатора LOGO! PROM позволяет тиражировать запрограммированные картриджи памяти. За один цикл он позволяет записывать программу в восемь картриджей. Программирование может производиться двумя способами:

1. С применением пилотного картриджа производится копирование программы.
2. Программа переносится из инструментальной среды разработки программ в картриджи.



Рис. 1.4. Автоматизированное рабочее место программиста ПЛК LOGO! (Siemens)

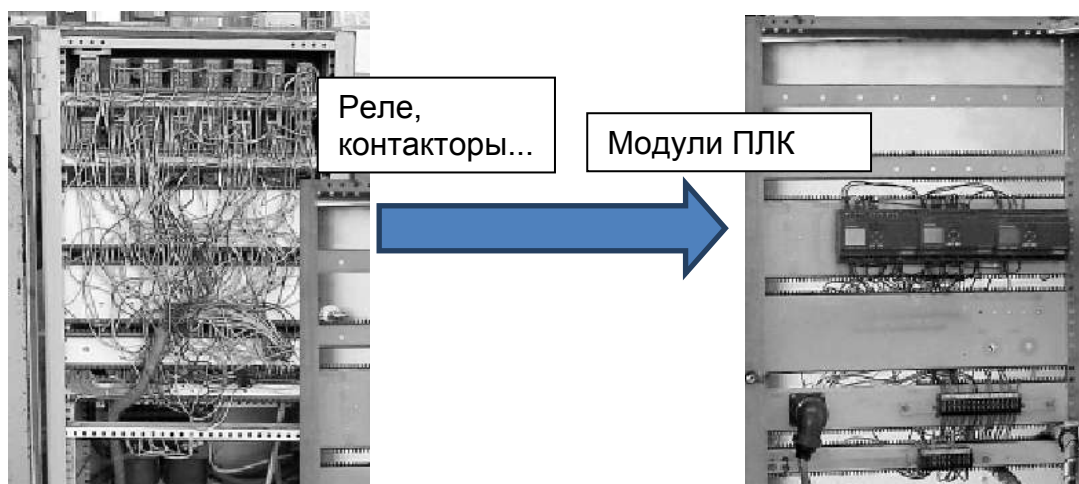
## Вопросы для самоконтроля

1. Какое программное обеспечение создаётся на языке С для PLC?
2. Из каких частей состоит рабочий комплекс программиста PLC?
3. В чём заключается универсальность технологии программирования PLC?
4. Из каких компонент состоит рабочее место программиста PLC?

### 1.4. Контроллеры семейства микро ПЛК LOGO!

Специфика технологии синтеза программного обеспечения для ПЛК заключается в учёте и подборе технических характеристик ПЛК. Поэтому при реализации проектов прикладных программ следует учитывать в первую очередь аппаратную составляющую проекта автоматизации.

В настоящее время широкое распространение получили модули – микро ПЛК LOGO! (SIMENS), занявшие нишу систем автоматизации в различных сферах применения: в серийно выпускаемых изделиях малого машиностроения, в транспортных системах, шкафах управления инженерными системами зданий и в управлении технологическими процессами. Важной функцией стало применение модулей микро ПЛК LOGO! для замены и модернизации электромеханических систем автоматизации, реализуемых на отдельных элементах: контакторах, электромагнитных реле времени и т.п., пример показан на рис. 1.5.



*Рис. 1.5. Замена системы реле автоматики шкафа управления на модули микро ПЛК LOGO!*

Доступность по цене и применению типы логических контроллеров, соответствующие характеристикам и конструктивному дизайну ПЛК LOGO! дают возможность модернизировать и реализовывать низкобюджетные проекты микроавтоматизации. Позиционирование ПЛК LOGO! в системе аппаратных средств автоматизации показано на рис. 1.6.

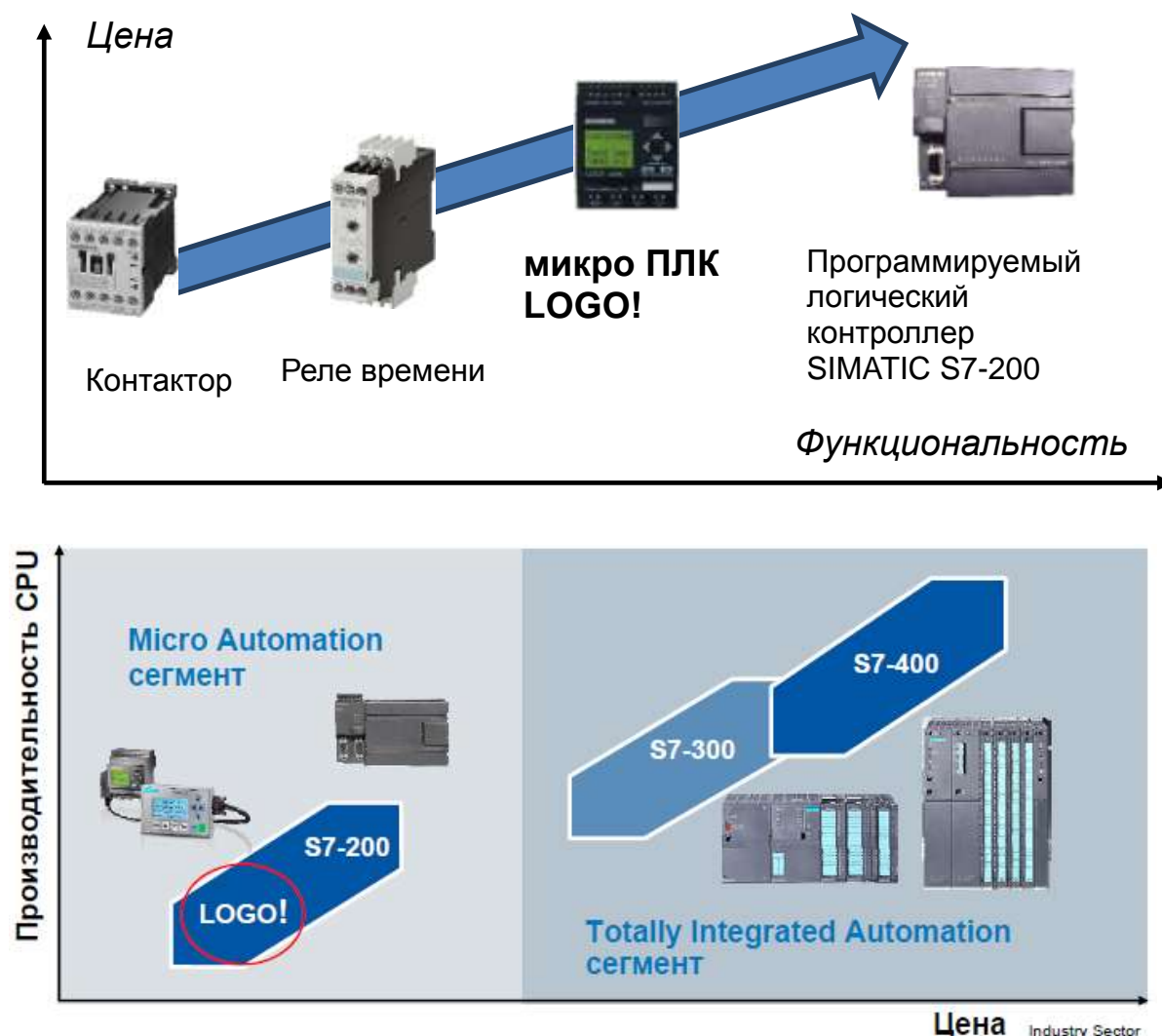


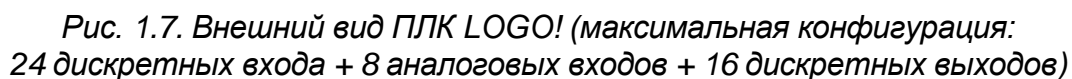
Рис. 1.6. Позиция модулей LOGO! в иерархии аппаратных средств комплексной автоматизации на базе контроллеров SIEMENS

Преимущество LOGO! Состоит в экономии: средств, времени, места установки оборудования. Гибкость при модернизации и масштабировании.

Схема легко обозрима. Снижение трудоемкости обслуживания сокращает вероятность возникновения дефектов, вызванных обрывом кабеля или нарушения электрических контактов в местах клемных соединений. Сокращается трудоемкость при монтаже проводов. Уменьшается потребляемая мощность и, как следствие, тепловыделение. Можно быстро изменить параметры структуры программы (алгоритма управления). Наличие средств сигнализации и самодиагностики.

Преимуществом является функциональное интегрирование с другими подсистемами на уровне сетевых интерфейсов, что позволяет поднять эффективность автоматизации на новый качественный уровень.





Например, логические модули микро ПЛК LOGO! Basic характеризуются следующими показателями:

- 8 дискретных входов, 4 дискретных выхода;
- 30 встроенных функций, сгруппированных в библиотеки логических (GF) и специальных (SF) функций;
- 8 внутренних флагов;



- встроенный календарь и часы реального времени (кроме LOGO! 24);
- встроенный жидкокристаллический дисплей и клавиатура;
- интерфейс для установки модуля памяти или подключения кабеля ПК для программирования с компьютера;
- дополняемые до 24 дискретных входов + 8 аналоговых входов + + 16 дискретных выходов.

Логические модули LOGO! Pure являются функциональными аналогами модулей LOGO! Basic, в которых отсутствуют дисплей и клавиатура. Программирование таких модулей производится либо с компьютера, оснащенного инструментальной средой разработки программ, либо установкой заранее запрограммированного модуля памяти.

Все входы модулей могут использоваться для ввода дискретных сигналов. Напряжение питания входных цепей соответствует напряжению питания модуля. В некоторых моделях 2 из 8 входов имеют универсальное назначение. Они могут использоваться для ввода дискретных сигналов или аналоговых сигналов 0...10В.

Различные модели модулей оснащены транзисторными или релейными выходами. Транзисторные выходы способны коммутировать токи до 0.3 А в цепях напряжением равным 24В и оснащены электронной защитой от короткого замыкания. Релейные выходы способны коммутировать токи до 10А (активная нагрузка) или до 3А (индуктивная нагрузка) в цепях напряжением равным 12/24В, ~24В или ~/= 115/240В.

Маркировка семейства модулей ПЛК содержит в своем составе логотип **LOGO!** (см. прил. 1), за которым следуют буквенно-цифровые обозначения, характеризующие конструктивные особенности данной модели:

- 12/24, 24, 230: напряжение питания модуля;
- R: релейные выходы;
- C: часы реального времени и календарь;
- o: модели LOGO! Pure без дисплея и клавиатуры.

Модули LOGO! Basic имеют 4, модули LOGO! Pure – 3 модификации:

1) **LOGO! Basic:**

LOGO! 12/24RC; LOGO! 24; LOGO! 24RC и LOGO! 230RC;

2) **LOGO! Pure:**

LOGO! 12/24RCo; LOGO! 24RCo и LOGO! 230RCo.

#### *1.4.1.1. Устройство и характеристики микро ПЛК LOGO!*

### **Модули LOGO! Basic и LOGO! Pure**

Логические модули **LOGO!** – это универсальные программируемые модули, предназначенные для построения первичных устройств автоматического управления (рис. 1.8).



*Рис. 1.8. Логические модули LOGO! Basic и LOGO! Pure с модулями расширения ввода/вывода и дисплеем*

Они могут использоваться автономно или дополняться необходимым набором модулей расширения. Компактные размеры, относительно низкая стоимость, простота программирования, монтажа и эксплуатации позволяют получать на их базе множество рентабельных решений для различных областей мониторинга, промышленного производства и автоматизации зданий. Данные модули обладают следующими особенностями:

- простота монтажа и эксплуатации, удобное и простое программирование;
- «все в одном»: встроенный дисплей и клавиатура, программируемая логика библиотеки встроенных функций, входы и выходы;
- возможность программирования с местной клавиатуры;
- использование дисплея и клавиатуры для решения простейших задач оперативного управления.

Конструктивно (рис. 1.9) логические модули LOGO! выпускаются в пластиковых корпусах размерами 72x90x55 мм и имеют степень защиты IP 20. Следует заметить важность для PLC параметра IP (Ingress Protection Rating) – системы классификации степеней защиты оболочки электрооборудования от проникновения твёрдых предметов и воды в соответствии с международным стандартом IEC 60529, DIN 40050, ГОСТ 14254-96).

На модуле располагаются:

- клеммы для подключения питания электроники модуля;
- клеммы для подключения входных цепей;
- клеммы для подключения цепей нагрузки;
- интерфейс для установки модуля памяти или подключения соединительного кабеля PC-LOGO!;
- клавиатура (только в модулях LOGO! Basic);
- дисплей (только в модулях LOGO! Basic);
- интерфейс внутренней шины для подключения модулей расширения;

- кодировочные пазы.

Все модули LOGO! монтируются на 35 мм профильную шину DIN, показанную на рис. 1.9, (стандарт – применяется для монтажа в шкафах управления (ШУ) АСУ ТП или на плоскую поверхность. Объединение всех модулей в единое устройство осуществляется через внутреннюю шину.

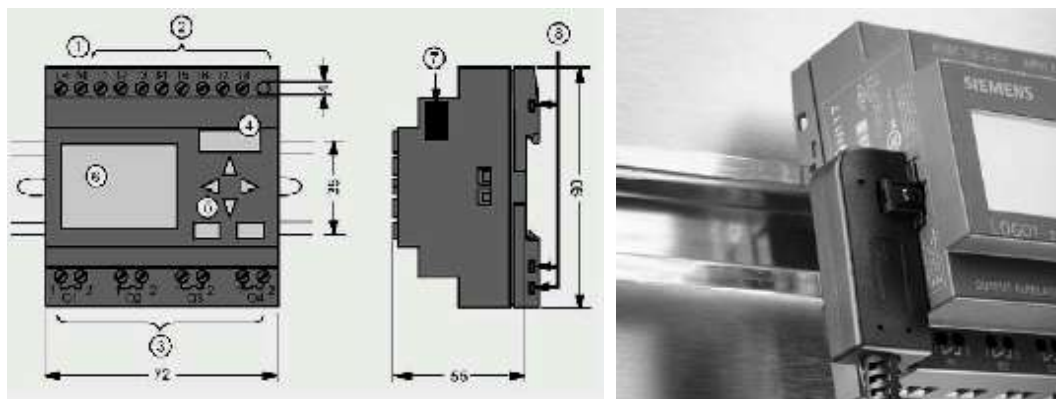


Рис. 1.9. Установочные параметры для установки на DIN рейку для модуля LOGO! Basic

Существуют ограничения на состав используемых модулей расширения. Для исключения ошибок при монтаже все модули семейства оснащены кодировочными пазами, а модули расширения и кодировочными штифтами.

Логические модули LOGO! версии 6ED1...-0BA4 имеют следующие конструктивные и функциональные особенности, показанные на рис. 1.1, 1.7:

- встроенный 32-разрядный микропроцессор;
- объем памяти программ, позволяющий использовать до 130 функций на программу;
- дисплей с внутренней светодиодной подсветкой, 4 строки по 12 символов (без поддержки кириллицы);
- увеличенный набор встроенных функций.

Все модули оснащены 8 входами и 4 дискретными выходами. В зависимости от модификации напряжение питания модуля может составлять =12 В, =24 В, =115 В, =230 В, ~24 В, ~115 В или ~230 В.

Напряжением питания модуля определяется и напряжение питания его входных цепей. Выходные каскады модулей выполняются на основе транзисторных ключей или герконовых реле. В моделях с транзисторными выходами два выхода могут использоваться в импульсном режиме.

Дисплей и клавиатура логических модулей LOGO! Basic используются как на этапе программирования, так и на этапе эксплуатации готового устройства. В процессе эксплуатации на экран дисплея вы-

водятся простейшие оперативные сообщения, (кириллица не поддерживается), которые можно использовать для модификации параметров управления (рис. 1.10).



Рис. 1.10. Дизайн аппаратного интерфейса модуля ПЛК LOGO!

#### 1.4.2. Особенности выбора и разработки конфигурации

Программа LSC позволяет разрабатывать пользовательские программы для устройств LOGO! различных серий. Различия заключаются в производительности, объеме памяти, числе программных блоков (например, блоков флагов), а также в структуре устройств.

Ввиду различия функций, выполняемых отдельными устройствами, необходимо не только различать версии устройств LOGO!, но и учитывать модификации устройств. Можно определить номер модификации по окончанию в заводском номере, указываемом в спецификации аппаратных средств КИПиА проекта автоматизации (прил. 2).

Текущая версия устройств LOGO! относится к 7-му поколению. Они могут быть идентифицированы по последним цифрам в номере: 0BA6.

Можно идентифицировать первые три поколения устройств LOGO! по окончаниям на 0, 1 и 2. Основные различия первых трех поколений устройств LOGO! заключаются в их модульной структуре.

#### Модули расширения

Модули ПЛК оборудуются интерфейсом расширения, к которому можно подключить модули расширения (цифровые/аналоговые модули и модули сетевых интерфейсов) для адаптации системы LOGO! под уникальные требования проекта. Выполнить подключение можно по внутренней шине лишь в том случае, если кодировочные штифты модуля расширения вошли в кодировочные пазы предшествующего модуля.

Модули расширения:

- DM: дискретных сигналов;
- AM: аналоговых сигналов;
- CM: сетевых интерфейсов (ASi, EIB/KNX);
- TD: выносной дисплей человеко-машинного интерфейса ЧМИ (англ. HMI).

Тем не менее, эта модульная структура оказывает либо незначительное влияние на программирование, либо не оказывает на него.

#### 1.4.2.1. Цифровые модули

Модули LOGO! DM8/ DM16 (рис. 1.11, 1.12) позволяют увеличивать количество дискретных входов и выходов. При этом общее количество дискретных каналов логического модуля и модулей расширения не должно превышать 24 дискретных входов и 16 дискретных выходов.

Модули ввода-вывода дискретных сигналов имеют два исполнения:

- LOGO! DM8 с 4 дискретными входами и 4 дискретными выходами.
- LOGO! DM16 с 8 дискретными входами и 8 дискретными выходами.



Технические характеристики модулей DM8

- |            |   |
|------------|---|
| DM8 230R   | <ul style="list-style-type: none"> <li>• Питание 120/230В AC/DC</li> <li>• 4DI 120/230В AC/DC</li> <li>• 4DO, 5А на реле</li> </ul> |
| DM8 12/24R | <ul style="list-style-type: none"> <li>• Питание 12/24В DC</li> <li>• 4DI 12/24В DC</li> <li>• 4DO, 5А на реле</li> </ul>           |
| DM8 24     | <ul style="list-style-type: none"> <li>• Питание 24В DC</li> <li>• 4DI 24В DC</li> <li>• 4DO, 0.3А на реле</li> </ul>               |
| DM8 24R    | <ul style="list-style-type: none"> <li>• Питание 24В AC/DC</li> <li>• 4DI 24В AC/DC PNP NPN</li> <li>• 4DO, 5А на реле</li> </ul>   |

Рис. 1.11. Цифровые модули расширения DM 8



Технические характеристики модулей DM16

- |                 |   |
|-----------------|---|
| LOGO! DM16 24   | <ul style="list-style-type: none"> <li>• Питание 24В DC</li> <li>• 8DI 24В DC</li> <li>• 8DO транзистор 0.3А</li> </ul>             |
| LOGO! DM16 230R | <ul style="list-style-type: none"> <li>• Питание 120/230В AC/DC</li> <li>• 8DI 120/230В AC/DC</li> <li>• 8DO, 5А на реле</li> </ul> |
| LOGO! DM16 24R  | <ul style="list-style-type: none"> <li>• Питание 24В DC</li> <li>• 8DI 24В DC</li> <li>• 8DO, 5А на реле</li> </ul>                 |

Рис. 1.12. Цифровые модули DM 16

Внутренняя шина модулей LOGO! DM не имеет устройств гальванического разделения цепей. Поэтому напряжение питания и род тока модуля расширения должны совпадать с аналогичными параметрами модуля, к которому он подключается.

#### 1.4.2.2. Аналоговые модули ввода/вывода

Модули LOGO! AM2/AM2, AM2 PT100/ AM2 AQ количество аналоговых входов и выходов, обслуживаемых одним логическим модулем микро ПЛК LOGO, показаны на рис. 1.13. На рис. 1.14 показаны типовые схемы включения аналоговых модулей вывода AM2 и AM2 Pt100.



##### Технические характеристики модулей аналоговых входов

- |           |  |
|-----------|--|
| AM2       | <ul style="list-style-type: none"> <li>• Питание 12/24В DC</li> <li>• 2AI 0...10В или 0...20mA</li> <li>• Разрешение: 10 бит</li> <li>• Поддержка датчиков: нет</li> </ul> |
| AM2 PT100 | <ul style="list-style-type: none"> <li>• Питание 12/24В DC</li> <li>• 2AI PT100, -50°C ... 200°C</li> <li>• 2- и 3- проводная схема подключения</li> </ul>                 |



##### Технические характеристики модулей аналоговых выходов

- |        |  |
|--------|--|
| AM2 AQ | <ul style="list-style-type: none"> <li>• Питание 24В DC</li> <li>• АО 0-10В или 0...20mA (значения 0...1000)</li> <li>• Разрешение: 10 бит</li> <li>• Точность аналоговых выходов: 2.5%</li> </ul> |
|--------|--|

Рис. 1.13. Аналоговые модули AM

#### 1.4.2.3. Аналоговые модули вывода

Функциональность выходов модуля LOGO! ...0BA6 – AM2 AQ LOGO! AM2 AQ – представлена на рис. 1.14. Показана типовая схема включения модуля аналогового вывода AM2 AQ.

Возможна подача сигнала в диапазонах:

- 0–10В / 0–20mA;
- 4–20mA.

Тип сигнала можно задать для каждого канала в LOGO! Soft Comfort или во встроенном меню центрального модуля.



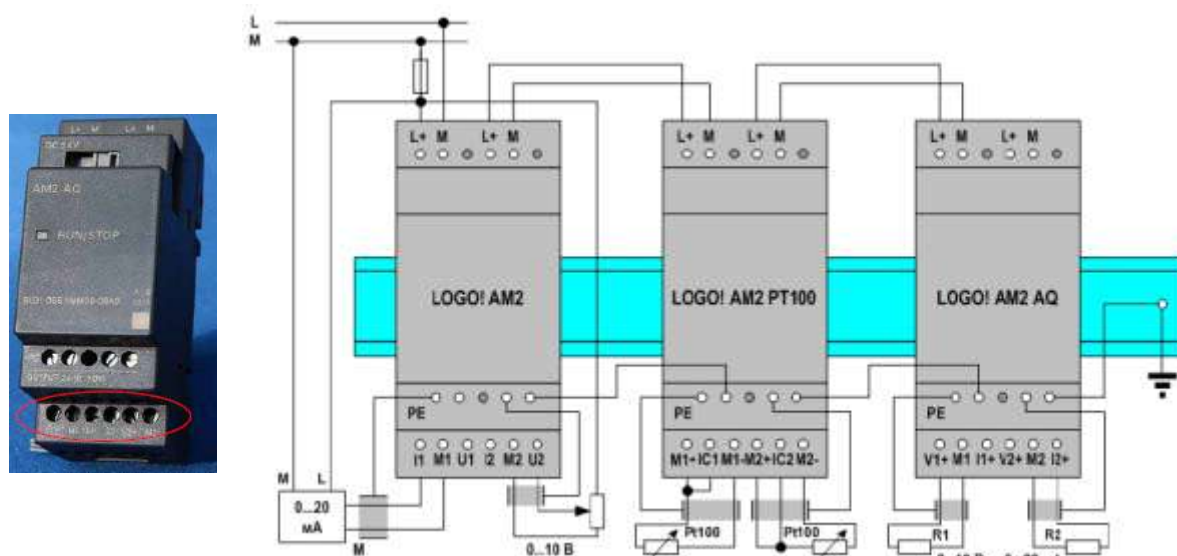


Рис. 1.14. Аналоговые модули AM LOGO!

#### 1.4.2.4. Коммутационные модули подключения омических нагрузок LOGO! Contact 24/230

Не всегда в электрических схемах проектов автоматизации используется только питающее напряжение модулей. Нередко необходимо управлять силовыми исполнительными механизмами (ИМ) такими, как электродвигатели, системы обогрева, освещения и др. Для этого применяются модули коммутации – LOGO! Contact (рис. 1.15).



Рис. 1.15. Модули LOGO! Contact 24 и LOGO! Contact 24/230

Модули LOGO! Contact предназначены для бесшумной коммутации цепей трехфазного переменного тока напряжением до 400 В. В цепях с активной нагрузкой силовые контакты модулей способны коммутировать токи до 20 А или производить включение/отключение трехфазных электродвигателей мощностью до 4 кВт. Важной является характеристика отсутствия фоновых помех. Модули выпускаются в двух модификациях с питающим напряжением электромагнита 24в DC (постоянного напряжения) и 230в AC (переменного напряжения).

Модули LOGO! Contact не имеют интерфейса подключения к внутренней шине логического модуля LOGO! Управление их работой осу-



ществляется через дискретные выходы логического модуля или модуля расширения DM8/ DM16. В линейке расширения они должны монтироваться в крайней левой или крайней правой позиции по отношению к модулям, объединенным внутренней шиной логического модуля.

#### 1.4.2.5. Текстовый дисплей LOGO! TD связи с оператором

Интерфейсные модули (рис.1.16) связи с оператором позволяют размещать выносные контрольные панели в удобных для доступа местах. Установка параметров и программирование панели осуществляются в единой среде программирования LSC.



Рис. 1.16. Выносная текстовая панель LOGO! TD

#### Технические характеристики панели TD:

- строки по 12 или 24 символа, до 50 текстовых сообщений на программу;

- включение в тексты сообщений состояний дискретных и аналоговых входов, а также дискретных выходов;
- поддержка 9 языков, в том числе и русского;
- 6 стандартных клавиш модуля LOGO! Basic и 4 дополнительных функциональных клавиш (F1–F4), используемых как дополнительные входы логического модуля;
- управление подсветкой дисплея из программы пользователя;
- построение бар-графиков;
- питание 12В DC, 24В AC/DC;
- степень защиты фронтальной панели IP65.

#### 1.4.2.6. Модули сетевых интерфейсов

##### Объединение в сеть по интерфейсу KNX

Коммуникационный модуль LOGO! CM EIB/KNX позволяет использовать логические модули LOGO! 0BA6/0BA7 в режиме ведущих устройств сети EIB/KNX (рис. 1.17). Эта сеть находит применение в системах автоматизации зданий. В сети KNX модуль LOGO! CM EIB/KNX позволяет опрашивать до 16 дискретных и до 8 аналоговых датчиков, а также управлять работой до 12 дискретных и до 2 аналоговых исполнительных устройств со скоростью передачи данных через 9600 бит/с (см. табл. 1.1).

Сигналы датчиков сети KNX обрабатываются программой логического модуля. Формируемые управляющие воздействия выдаются на исполнительные устройства сети KNX.



Рис. 1.17. Коммуникационные модули LOGO! CM EIB/KNX: 1 – источник питания; 2 – шинный ползок сопряжения с LOGO! 3 – RUN/STOP индикаторы подключения к LOGO! 4 – сетевой LED, EIB связь; 5 – метка для физического адреса; 6 – входы – EIB соединение; 7 – интерфейс расширения LOGO! 8 – механическое кодирование – штифт; 9 – кнопка программирования

Один логический модуль LOGO! способен обслуживать до 24 дискретных и до 8 аналоговых входов, до 16 дискретных выходов и до

2 аналоговых. Поэтому при использовании модуля LOGO! CM EIB/KNX в сочетании с другими модулями расширения количество датчиков и исполнительных устройств, подключаемых через сеть KNX, снижается. Например, при использовании модуля LOGO! CM EIB/KNX в сочетании с модулем LOGO! DM8 через сеть KNX может быть подключено не более 12 дискретных датчиков и не более 8 дискретных исполнительных устройств.

Интерфейс подключения к внутренней шине модуля LOGO! CM EIB/KNX оснащен устройствами гальванического разделения цепей. Поэтому в линейке расширения он может подключаться к модулям с любым напряжением питания и родом тока. В линейке модулей расширения модуль LOGO! CM EIB/KNX должен устанавливаться в крайней правой позиции.

Перевод модуля в режим программирования осуществляется с помощью кнопки Prog на фронтальной части корпуса. Переход в это состояние индицируется оранжевым свечением светодиодов.

Программирование модуля LOGO! CM EIB/KNX выполняется с компьютера, оснащенного программным обеспечением ETS2 от V1.2. С помощью этого программного обеспечения задается адрес модуля LOGO! CM

### **Адресация подчиненных устройств в сети KNX**

В программе каналам ввода-вывода, подключаемым через сеть KNX, присваиваются адреса, следующие за адресами каналов модуля ПЛК или последнего модуля расширения. Например, в конфигурации, включающей в свой состав логический модуль LOGO! 12/24RC с двумя аналоговыми входами и коммуникационный модуль LOGO! CM EIB/KNX, адресация сетевых входов и выходов будет выглядеть следующим образом (табл. 1.1):

- дискретные входы модуля LOGO!: I1 ... I6;
- дискретные KNX входы: I7 ...;
- дискретные выходы модуля LOGO!: Q1 ... Q4;
- дискретные KNX выходы: Q5 ...;
- аналоговые входы модуля LOGO!: AI1, AI2;
- аналоговые KNX входы: AI3 ...;
- аналоговые KNX выходы: AQ1 ....

Таблица 1.1

#### **Характеристики модуля LOGO! CM EIB/KNX**

Виртуальные каналы ввода/вывода	Кол-во
Дискретных входов, не более	16
Дискретных выходов, не более	12
Аналоговых входов, не более	8
Аналоговых выходов, не более	2
Количество групп адресов, не более	56
Количество объединений, не более	56

### Подключение в сеть по шине AS-Interface

Коммуникационный модуль LOGO! CM AS-i выполняет функции ведомого устройства AS-Interface и поддерживает 4 виртуальных дискретных входа и 4 виртуальных дискретных выхода, показаны на рис. 1.18 (табл. 1.2). Он и способен передавать информацию о состоянии 4 дискретных входов LOGO! ведущему устройству AS-Interface, а также принимать от него 4 управляющих дискретных сигнала.



Рис. 1.18. Коммуникационные модули CM LOGO! AS-Interface

Модуль LOGO! CM AS-i может использоваться в сочетании с другими модулями расширения.

Таблица 1.2

#### Характеристики модуля ASi

Допустимый диапазон питания	19,2... 28,8 В постоянного тока
Общее потребление тока	$I_{tot}$ макс. 70 мА
Цифровые входы (I)	следующие четыре входа после физических входов модуля LOGO!: ( $I_n \dots I_{n+3}$ )
Цифровые выходы (Q)	следующие четыре выхода после физических выходов модуля LOGO!: ( $Q_n \dots Q_{n+3}$ )
Конфигурация входов и выходов (шестнадцатеричные значения)	7
Код ID (шестнадцатеричный)	F
Код ID1 (шестнадцатеричный)	F (по умолчанию, настраиваемый от 0 до F)
Код ID2 (шестнадцатеричный)	F
Подключение шины	AS-Interface
Аналоговые входы (AI)	нет
Аналоговые выходы (AQ)	нет

При использовании модуля LOGO! CM AS-i количество реально существующих дискретных входов и выходов на систему из логического модуля и его модулей расширения не должно превышать 20 и 12 соответственно.

## Коммутаторы Industrial Ethernet LOGO! CSM

### Назначение

Модули LOGO! CSM выполняют функции неуправляемых коммутаторов Ethernet и предназначены для совместного использования с логическими модулями LOGO! 0BA7. С их помощью можно получить три дополнительных интерфейса Ethernet для организации обмена данными между модулем LOGO! 0BA7 и программатором, приборами и системами человеко-машинного интерфейса, другими логическими модулями и системами автоматизации.

Неуправляемые 4-канальные коммутаторы Industrial Ethernet для построения линейных, древовидных и звездообразных сетевых структур:

- подключение к LOGO! 0BA7 до трех дополнительных сетевых узлов;
- обмен данными со скоростью 10/ 100 Мбит/с;
- рентабельное решение для построения небольших локальных сетей;
- четыре гнезда RJ45 промышленного исполнения;
- одно гнездо RJ45 на фронтальной панели модуля для выполнения диагностики непосредственно в шкафу управления.
- встроенная светодиодная индикация;
- наличие модификаций с напряжением питания =12/24 В и  $\cong$ 230 В.

### Особенности

- Снижение затрат на монтаж и экономия монтажных объемов по сравнению с использованием внешних сетевых компонентов других серий.
- Рентабельное решение для построения небольших локальных сетей Ethernet.
- Замена модуля без повторного конфигурирования сети.
- Получение дополнительных интерфейсов Ethernet логического модуля LOGO! 0BA7 для подключения различных сетевых узлов.
- Работа с естественным охлаждением, снижение затрат на обслуживание.

### Функции

- Увеличение количества портов Industrial Ethernet логического модуля LOGO! 0BA7.
- Формирование небольших локальных сетей Industrial Ethernet с подключением к контроллеру до трех дополнительных сетевых узлов.
- Автоматическая настройка на скорость обмена данными в сети, автоматическая кроссировка подключаемых кабелей.
- Индикация состояний каждого из 4-х коммуникационных портов.

- Подключение программатора для диагностики аппаратуры непосредственно в шкафу управления.

#### 1.4.2.7. Модульная конфигурация LOGO!

##### Максимальная конфигурация

Максимальная конфигурация модуля LOGO! с аналоговыми входами – используется четыре входа (LOGO! 12/24 RC/RCo и LOGO! 24/24o).

##### Оптимально быстрый обмен данными

Для получения оптимального по скорости обмена данными между LOGO! Basic и различными модулями рекомендуется структура «Сначала цифровые модули, а затем аналоговые модули» (рис. 1.19, 1.20).

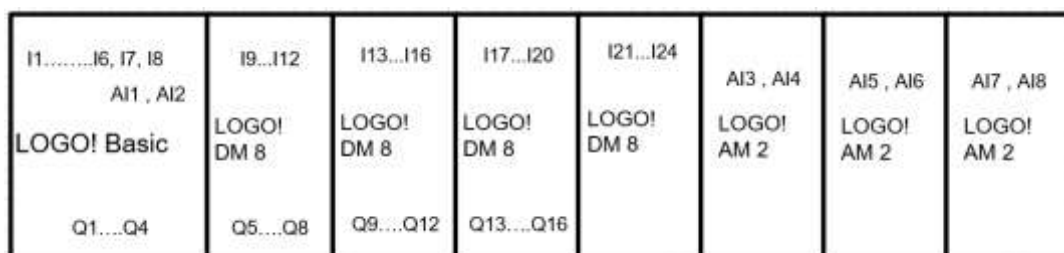


Рис. 1.19. LOGO! Basic, 4 цифровых модуля и 3 аналоговых модуля

Исключение составляют аналоговый вход AI ПИ регулятора для ввода параметра PV, он должен находится на модуле ПЛК либо на соседнем модуле аналоговых входов.

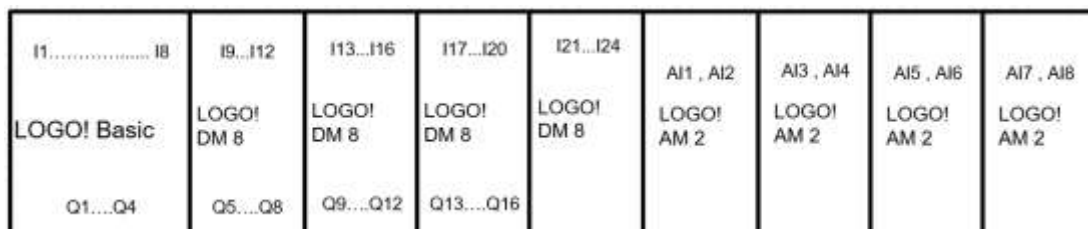


Рис. 1.20. LOGO! Basic, 4 цифровых модуля и 4 аналоговых модуля

##### Примеры модульных структур

Модули TD матричных дисплеев подключается через отдельный интерфейсный кабель (рис. 1.21–1.23).

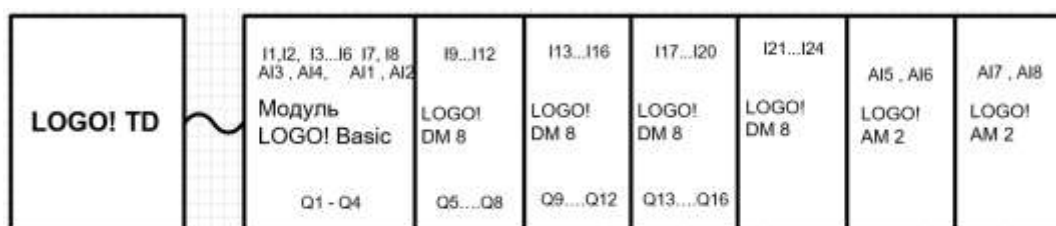


Рис. 1.21. LOGO! Basic, 4 цифровых модуля и 2 аналоговых модуля



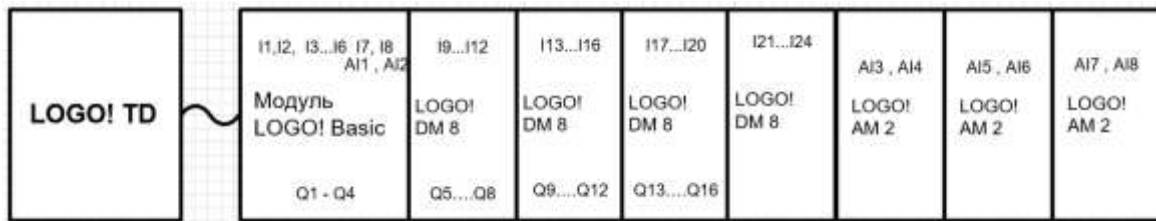


Рис. 1.22. LOGO! Basic, 4 цифровых модуля и 3 аналоговых модуля

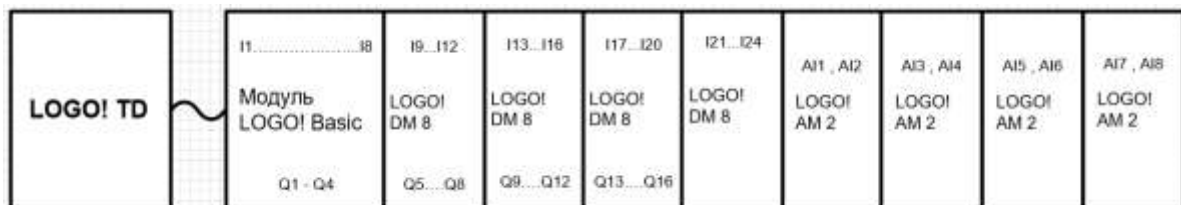


Рис. 1.23. LOGO! Basic, 4 цифровых модуля и 4 аналоговых модуля

### 1.4.3. Сетевые структуры ПЛК

#### 1.4.3.1. ASi-интерфейс

AS-i – открытый стандарт для построения сетей на уровне управляемого технологического процесса (рис. 1.24). Электрические и механические характеристики AS-интерфейса были разработаны с участием одиннадцати компаний, специализирующихся в области бинарных датчиков и исполнительных механизмов. Спецификации доступны для всех компаний, имеющих отношение к этой области. AS-интерфейс является открытым гетерогенным стандартом.

Под именем AS-интерфейса, ведущий производитель электронного оборудования управления – Siemens предлагает систему, которая удовлетворяет требованиям стандарта AS-i.

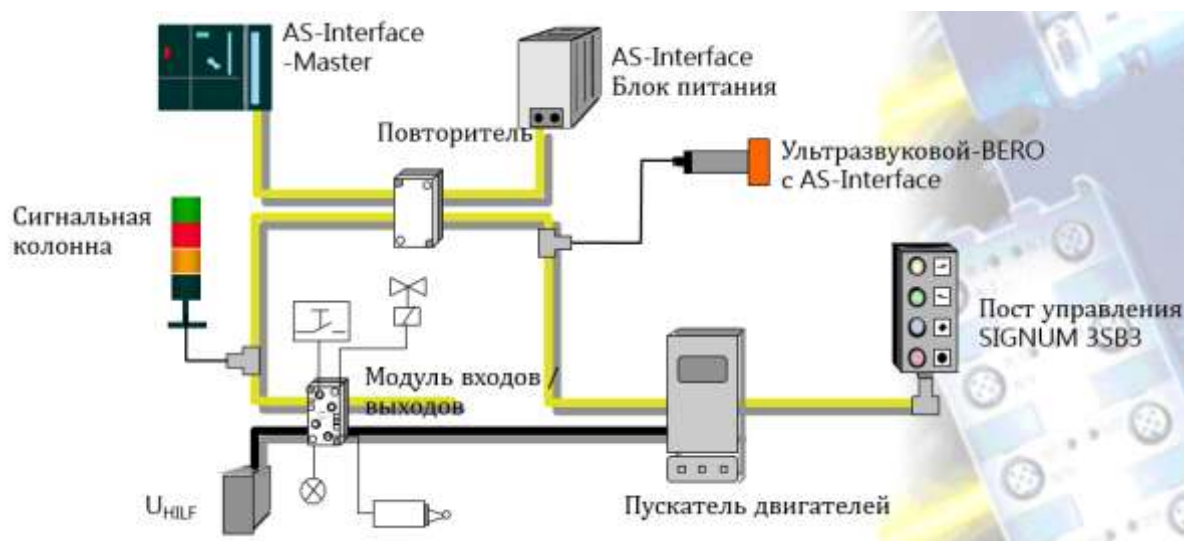


Рис. 1.24. Пример системы управления с ASi интерфейсом



За продвижение и распространение AS-i систем отвечает "Ассоциация поддержки интерфейсов для подключения бинарных исполнительных механизмов и датчиков в шину" ("Association for Promoting Interfaces with Bus Capability for Binary Actuators and Sensors" – Ассоциация AS-i). В частности, Ассоциация отвечает за спецификации, характеристики, стандартизацию, сертификацию и пользовательскую информацию общего назначения.

Основные характеристики AS-интерфейса обладают отличительными чертами:

- AS-интерфейс оптимален для подключения бинарных датчиков и исполнительных механизмов. Кабель AS-i используется как для обмена данными между датчиками/исполнительными механизмами (ведомыми устройствами AS-i) и ведущим устройством AS-i, так и для подачи напряжения питания на датчики/исполнительные механизмы;
- более простой и экономичный монтаж соединений. Благодаря использованию метода прокалывания изоляции упрощается монтаж кабеля и достигается высокая гибкость, необходимая для построения древовидной топологии;
- малое время реакции: ведущему устройству AS-i требуется не более 5мс для циклического обмена данными с 31 узлом сети;
- в качестве узлов (AS-i ведомых) кабеля AS-интерфейса могут выступать либо датчики/исполнительные механизмы со встроенным AS-i интерфейсом, либо модули AS-i, к которым можно подключить до 4 обычных бинарных датчиков/исполнительных механизмов;
- при использовании стандартных AS-i модулей на кабеле AS-i может находиться до 124 исполнительных механизмов/датчиков.
- если используются AS-i модули с расширенным режимом адресации, с одним ведущим устройством с расширенным режимом адресации, то могут работать до 186 исполнительных механизмов и 248 датчиков;
- расширенные ведущие устройства AS-интерфейса семейства SIMATIC NET (рис. 1.25) обеспечивают чрезвычайно простой доступ к аналоговым датчикам/исполнительным механизмам или модулям, функционирование которых соответствует профилю ведомых устройств AS-интерфейса.

Ведущее устройство AS-Interface опрашивает виртуальные входы модуля LOGO! CM AS-i и передает управляющие воздействия на виртуальные выходы. Опрос реально существующих входов и управление реально существующими выходами осуществляет логический модуль LOGO! При этом в программе модуля LOGO! используются как реально существующие, так и виртуальные входы и выходы.

Применение модулей LOGO! CM AS-i позволяет выполнять согласованное функционирование нескольких модулей LOGO!, использовать

входные сигналы одних модулей LOGO! для управления выходами другого модуля, использовать в работе модуля LOGO! сигналы, формируемые датчиками AS-Interface и т.д. При этом логические модули LOGO! могут быть разнесены на расстояние до 600 м, а функции согласования их работы выполняет ведущее устройство AS Interface (рис. 1.25).



Рис. 1.25. Сеть на основе интерфейса AS-i с ведущим Master PLC S7-1200

В зависимости от типа к одному ведущему устройству AS Interface допускается подключать до 32 или до 64 ведомых устройств. Каждому ведомому устройству присваивается свой адрес в сети AS-Interface. Для этой цели необходим прибор адресации AS-Interface.

Интерфейс подключения к внутренней шине модуля LOGO! CM AS-i оснащен устройствами гальванического разделения цепей. Поэтому в линейке расширения он может подключаться к модулям с любым напряжением питания и родом тока.

Сетевой модуль ASi подключается последним во избежание пропадаания питания в связи с отключением питания в сети ASi.

#### 1.4.3.2. KNX/EIB – интерфейс

KNX – коммуникационная шина, широко используемая для автоматизации зданий, показана на рис. 1.26.

Стандарт шины KNX стал развитием более ранней разработки EIB (от англ. European Installation Bus, рус. Европейская инсталляционная шина). EIB – устаревшее обозначение, но оно продолжает использоваться, особенно в Европе. Иногда используется обозначение EIB/KNX.

Продукция KNX распространялась под несколькими торговыми марками. Наиболее известны Instabus, ABB i-Bus, Tebis, Theben.

В феврале 1990 года была организована ассоциация EIBA со штаб-квартирой в г. Брюссел (Бельгия). Основателями ассоциации были компании Siemens, Gira, ABB, Berker, Jung и другие – всего 15 компаний. Задачи ассоциации: продвигать на рынке саму технологию, следить за качеством и совместимостью оборудования, производимого её членами, и подготавливать программы обучения специалистов.



Рис. 1.26. Пример структуры сети KNX/EIB

Подключенные к шине (сети) абоненты (устройства) могут обмениваться информацией через общий канал передачи, шину. При этом подлежащая передаче информация упаковывается в телеграмму и передается по кабелю от датчика (сенсора – отправителя команд) к одному или нескольким исполнительным механизмам. При успешной передаче и приеме каждое устройство-приёмник квитирует (подтверждает) получение телеграммы. При отсутствии подтверждения передача повторяется еще два раза. Если и после этого квитирование отсутствует, то процесс передачи заканчивается. Именно поэтому протокол EIB не является «промышленным», то есть его нельзя применять в приложениях, связанных с опасностью для людей.

#### 1.4.3.3. Сетевой интерфейс Ethernet

Логические модули LOGO! 0BA7 оснащены встроенным интерфейсом Ethernet (рис. 1.27) и позволяют производить дальнейшее расширение своей системы ввода-вывода за счет сетевого обмена данными с другими модулями LOGO! 0BA7.

В такой системе один логический модуль выполняет функции ведущего, остальные модули функции ведомых устройств. К одному ведущему модулю может быть подключено до 8 ведомых модулей (рис. 1.28). Каждый логический модуль ПЛК может иметь собственный

набор модулей расширения. Программу выполняет только ведущий логический модуль.



Рис. 1.27. Модуль LOGO! в версии 0BA7 с сетевым интерфейсом Ethernet и модулями расширения I/O

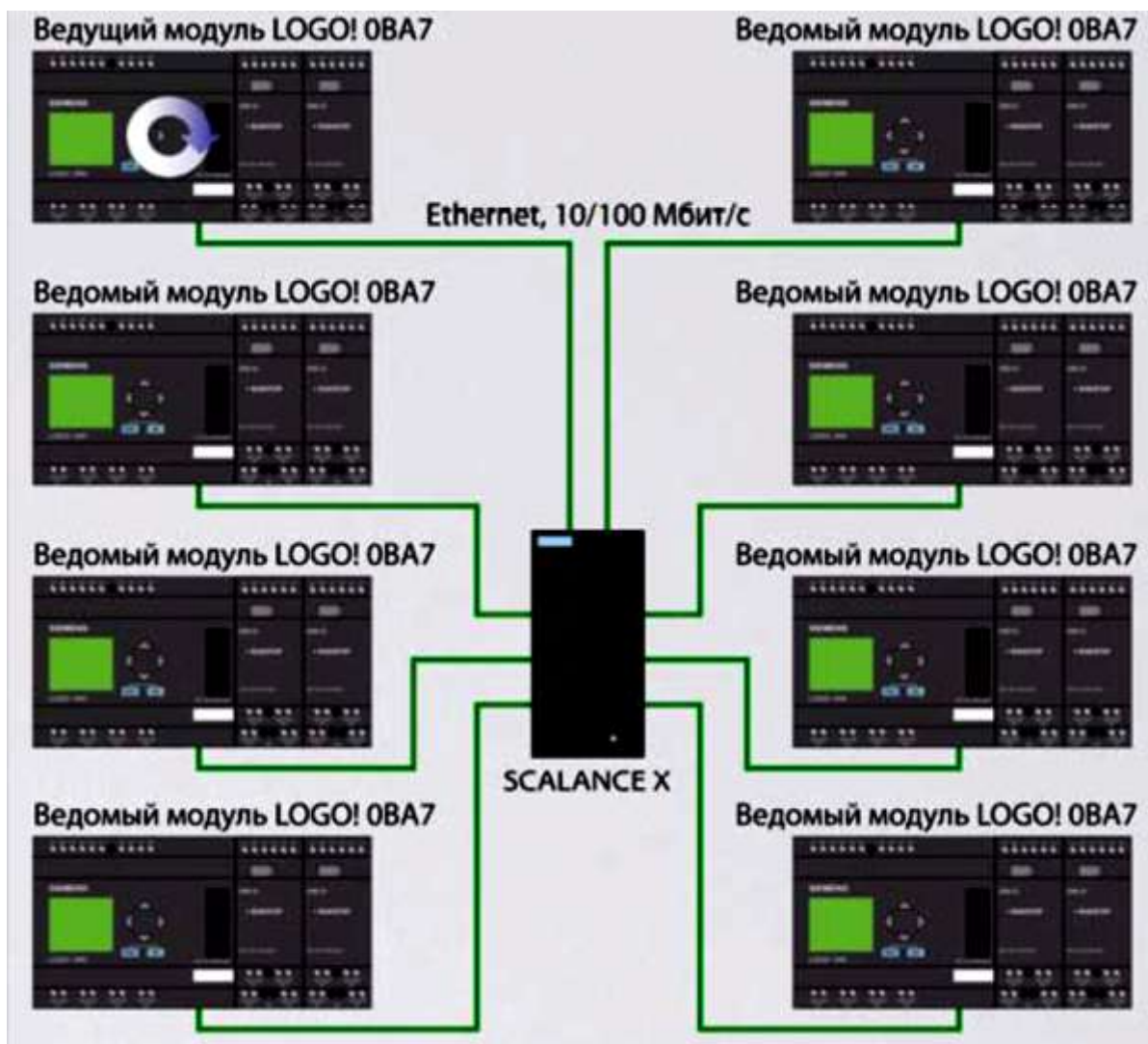


Рис. 1.28. Пример структуры сети Ethernet. Режим работы «Mastre-Slave»

Ведомые модули собственной программы не имеют и выполняют функции блоков расширения ведущего логического модуля. За счет этого ведущий логический модуль способен обслуживать до 64 дискретных входов, до 64 дискретных выходов, до 32 аналоговых входов и до 16 аналоговых выходов.

#### 1.4.3.4. LON – интерфейс

Коммуникационный модуль (CM) Local Operating Network (CM LON) – это устройство сопряжения для подключения LOGO! к сети LON Works (Локальная рабочая сеть), используемая в системах автоматизации зданий, показан на рис. 1.29.

Коммуникационный модуль LOGO! применяется как модуль для управляющего модуля LOGO (12/24 или 110/240 Вольт).

Модуль поддерживает связь между LOGO! ведущим устройством и внешними LON устройствами через LON сеть.

Коммуникационный модуль – это шинная станция в сети LON и он позволяет LOGO! связываться с другими LON устройствами.

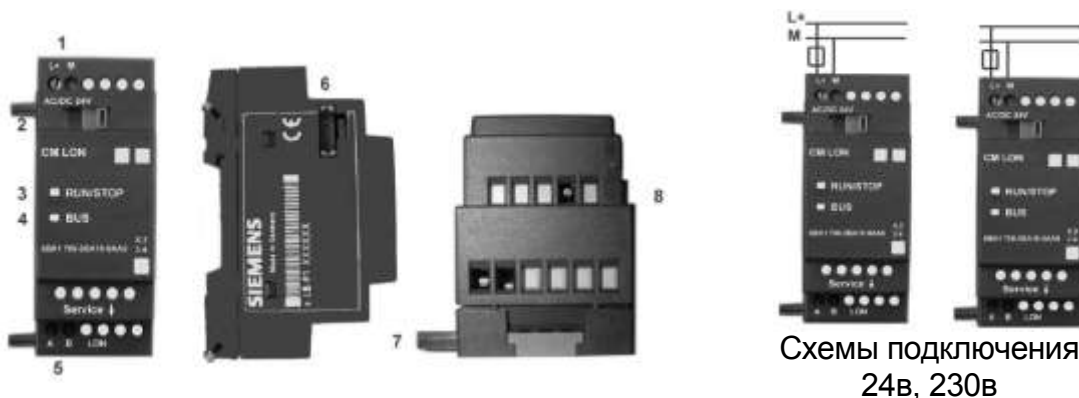


Рис. 1.29. Коммуникационные модули LOGO! CM LON: 1 – источник питания; 2 – шинный ползок сопряжения с LOGO!; 3 – RUN/STOP индикаторы подключения к LOGO! 4 – сетевой индикатор LED, EIB связь; 5 – метка для физического адреса; 6 – входы – LON соединение; 7 – интерфейс расширения LOGO!; 8 – механическое кодирование – штифт; 9 – кнопка программирования

CM передает текущее состояние станции LON в LOGO!, который таким образом может задействовать свои логические функции и таймеры для объединения их вместе. В процессе работы сигналы LON могут также быть скомбинированы с сигналами локальных входов и входов LOGO! CM передает любые изменения выходных сигналов через сеть LON.

Комбинация LOGO! и LON-CM дает пользователю функциональность децентрализованного контроллера для сети LON с возможностью устанавливать и изменять параметры или действия быстро, просто и без устройства программирования.

Система LON образует систему распределённого ввода вывода логического модуля ПЛК. Встроенные входы и выходы ПЛК и модулей расширения образуют систему локального ввода-вывода. Один модуль ПЛК способен обслуживать как распределённую, так и локальную систему ввода-вывода.

### **Вопросы для самоконтроля**

1. Для каких функций предназначен АПК микроPLC LOGO?
2. Какие преимущества для разработчиков даёт комплекс PLC?
3. Назовите типы модулей LOGO!. Приведите принципы кодировки их модификации в заводских номерах.
4. Назовите состав конфигурации модуля PLC LOGO!
5. Для чего предназначены встроенные интерфейсы?
6. На основе какого процессора выполнена микро ЭВМ ПЛК?
7. Назовите типы модулей расширения.
8. Как подключаются модули расширения к ПЛК?
9. Как подключаются исполнительные механизмы к ПЛК?
10. Какие сетевые интерфейсы поддерживаются семейством модулей LOGO!
11. Назовите основные характеристики сетевого интерфейса ASi.
12. Приведите характеристики сетевого интерфейса EIB/KNX.
13. В каких видах сетевых интерфейсов модули ПЛК LOGO! могут иметь функции мастера?



## Глава 2. ИНСТРУМЕНТАЛЬНАЯ СРЕДА РАЗРАБОТКИ ПРОГРАММ

### 2.1. Пользовательский интерфейс

#### 2.1.1. Интерфейс программирования

Интерфейс программы инструментальной среды разработки LOGO! SoftComfort (LSC) имеет вид, показанный на рис. 2.1.

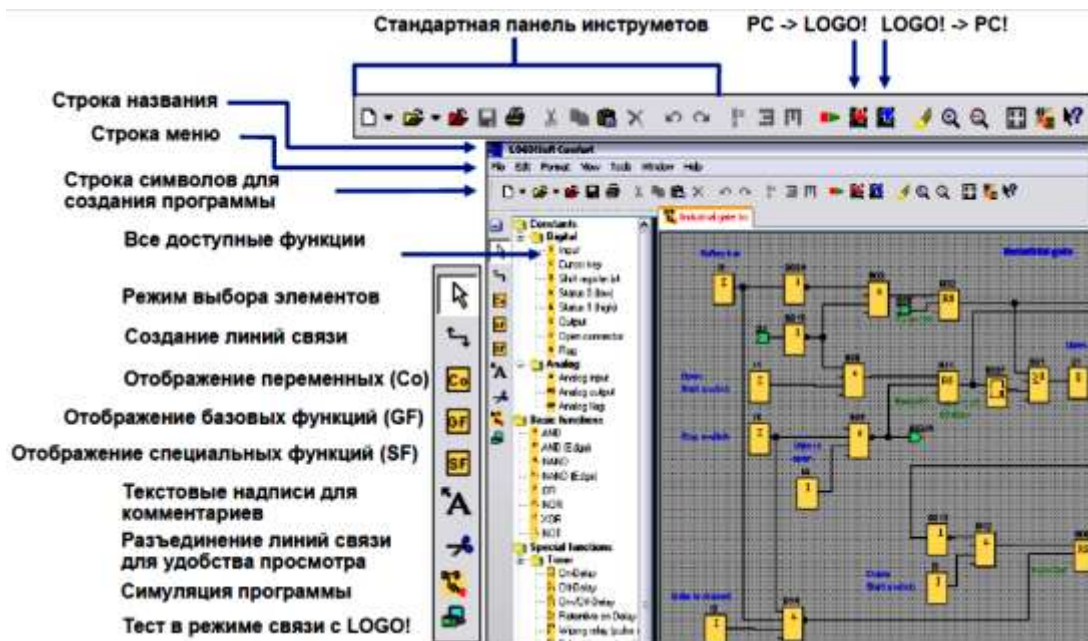


Рис. 2.1. Пользовательский интерфейс LOGO! SoftComfort

#### Панель меню

Панель меню расположена в верхней части окна. На ней располагаются различные команды для редактирования и управления создаваемых программ, а также функции для задания параметров по умолчанию.

#### Панели инструментов

Среда программирования содержит три панели инструментов:

- стандартную панель инструментов;
- панель инструментов программирования;
- панель инструментов эмуляции.

**Строка состояния** расположена в нижней части окна программы в ней отображаются: текущий активный инструмент, состояние программы, коэффициент масштабирования, номер страницы электрической схемы и выбранное устройство ПЛК.

**Стандартная панель инструментов** находится над интерфейсом программирования.



**Панель инструментов программирования** располагается в левой части экрана. Находящиеся в ней значки могут использоваться для перехода в другие режимы редактирования или для быстрого и легкого создания и редактирования коммутационной программы.

### **Панель инструментов эмуляции**

Эта панель инструментов необходима только для эмуляции коммутационных программ.

**Окно информации**, расположенное в нижней части интерфейса программирования, отображает информацию и примечания, а также модули ПЛК, рекомендуемые к использованию в вашей коммутационной программе, функцией Сервис -> Определить модули ПЛК.

### **2.1.2. Функции окна информации**

Окно информации отображает:

- сообщения об ошибках, сформированные в начале эмуляции;
- модули LOGO! определенные командой меню «Сервис -> Определить LOGO!» или функциональной клавишей F2;
- дату и время сообщения;
- название коммутационной программы, для которой было создано сообщение.

При запуске режима эмуляции производится функция анализа программы относительно имеющихся ресурсов и подлежащей использованию системы. Используемые ресурсы и произошедшие ошибки отображаются в окне информации. Окно информации отображает все данные последовательно.

### **Описание строки состояния**

Строка состояния на рис. 2.2. имеет четыре поля и содержит полезную информацию о коммутационной программе.



*Рис. 2.2. Строка состояния*

1. Поле информации. Отображает, например, текущий выбранный инструмент.

2. Отображает выбранное устройство LOGO! при помощи всплывающей подсказки. Если надо изменить выбор, то двойное нажатие значка LOGO! активирует меню Сервис -> Выбор оборудования.

3. Отображает текущий выбранный масштабный коэффициент.

4. Это последнее поле отображает текущую страницу коммутационной программы.

## Вопросы для самоконтроля

1. Для каких основных функций предназначен аппаратный комплекс микро PLC?
2. Какие панели инструментов содержит среда программирования?
3. Какие функции строки состояния, Вы знаете?
4. Какие функции окна информации, Вам известны?

## 2.2. Панели инструментов

### 2.2.1. Стандартная панель инструментов

Значки стандартной панели инструментов обеспечивают быстрый доступ к командам, доступным также при помощи меню (рис. 2.3).



Рис. 2.3. Стандартная панель инструментов

На стандартной панели инструментов находятся следующие дополнительные команды:

1. **Сервис:** Переключение LOGO! Режим, ПК -> LOGO! ->LOGO! -> ПК (Загрузка).
2. **Вид:** Выбрать линии, масштаб.
3. **Файл:** Свойства, вкладка «Разметка страницы».
4. **Преобразовать** функциональную блок-схему в релейно-контактную схему и наоборот.

### 2.2.2. Панель инструментов моделирования и окно состояния

При входе в режим эмуляции раскрывается панель инструментов моделирования (рис. 2.4). Она содержит следующие значки:

- например, переключатели для управления оператором входами;
- для эмуляции отказа питания, для испытания коммутационного отклика с учетом характеристик сохранения после аварии питания;
- например, лампочки для контроля выходов;
- для управления эмуляцией;
- для управления временем.

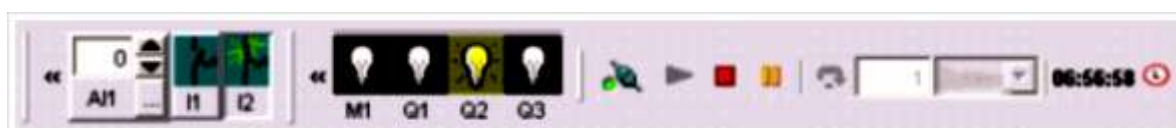


Рис. 2.4. Панель инструментов эмуляции

## Дисплей состояния

Предварительное условие: отображение состояний сигнала и технологических переменных разрешается функциями «Сервис – Параметры: Эмуляция».

Цветовая индикация позволяет определить состояние «1» или «0» на соединительной линии (рис. 2.5). По умолчанию цвет соединительной линии с сигналом «1» – красный. По умолчанию цвет соединительной линии с сигналом «0» – синий.

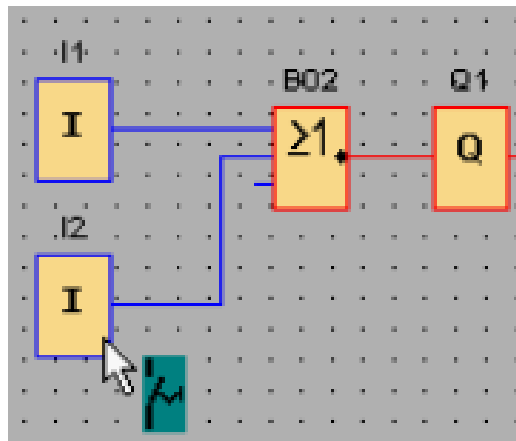


Рис. 2.5. Пример для редактора функциональных блок-схем в режиме эмуляции

### 2.2.3. Панель инструментов программирования

#### 2.2.3.1. Панель инструментов программирования – обзор

Панель инструментов программирования содержит значки для создания, редактирования и испытания программ. Каждый из этих инструментов представляет собой режим программирования, в котором манипуляции мышью оказывают различное воздействие. Сервис редактирования не доступен в виде команд меню.

#### 2.2.3.2. Каталог элементов коммутационной программы

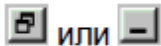







Каталог содержит иерархический перечень всех элементов, которые можно использовать для создания коммутационной программы.

#### 2.2.3.3. Разрезать/связать

Этот инструмент служит для разрезания и связывания соединений между блоками. Для разрезания соединения выполните нажатие для выбора соответствующей линии при активном инструменте. Разрезать/связать. Соединение заменяется указанием у блоков ссылки на блок-партнер. Ссылка содержит информацию на номер страницы, номер блока и входы и выходы блока-партнера.

Таблица 2.1

## Значки меню панели инструментов программирования

Значки	Описание
	Каталог элементов коммутационной программы открыть / закрыть
	Разрезать-связать
	Инструмент соединителей
	Постоянные и клеммы
	Базовые функции (только для редактора функциональных блок-схем)
	Специальные функции
	Эмуляция
	Оперативное тестирование

**Вопросы для самоконтроля**

1. Что отображается цветовая индикация в режиме эмуляции программы?
2. Из каких компонент состоит панель инструментов программирования?
3. Для чего нужен инструмент «разрезать-связать»?
4. Какую информацию содержит ссылка в режиме «разрезать»?

**2.3. Строка меню****2.3.1. Файл. Сохранить как**

Открывается диалоговое окно, в котором вы задаете путь и имя файла, под которым должна быть сохранена текущая коммутационная программа. Это позволяет вам сохранять измененные программы под другими именами и в других папках, сохраняя предыдущие версии для последующего извлечения.

Ниже перечислены типы файлов, которые можно сохранять:

- Файл LOGO!Soft Comfort FBD (\*.lsc);
- Файл LOGO!Soft Comfort LAD (\*.lld);
- Файл LOGO!SoftStandard (\*.lgo);

- Файл в формате для обмена документами (\*.pdf);
- Файл JPG (\*.jpg);
- Файл точечного рисунка (\*.bmp).

По расширению в программе выбираются следующие расширения имени файла: для программ функциональных блок-схем – \*.lsc, и \*.lld – для программ релейно-контактных схем. Можно экспортировать программу в более старую версию программы LSC.

Сохранить коммутационную программу можно в графическом формате, например – \*.jpg, \*.bmp или \*.pdf, т.е. в формате программы Acrobat Reader, с целью документирования программы и ее презентации.

### **2.3.2. Файл -> Сравнение коммутационных программ**

Эта функция может использоваться для сравнения двух коммутационных программ (не распознает различия в графическом размещении блоков и в комментариях).

#### **Предварительное условие**

- Для этого коммутационные программы должны быть открыты программой LSC.
- Вы можете сравнивать только коммутационные программы одного типа, например, \*.lsc с \*.lsc.

### **2.3.3. Файл. Преобразование РКС в ФБД**

Преобразование из релейно-контактной схемы (РКС) в функциональную блок-схему (ФБД) осуществляется с использованием следующих правил:

- схемы с последовательными контактами преобразуются в блок И;
- схемы с параллельными контактами преобразуются в блок ИЛИ;
- задаваемые пользователем комментарии не включают, поскольку их положение в коммутационной программе не может быть задано на блоках;
- перекрестные ссылки (соединения, при которых выход блока соединяется со входами нескольких блоков, и по крайней мере один из входов соединен с выходами нескольких блоков) преобразуются в блок ИЛИ;
- входами блока ИЛИ являются все выходы блоков перекрестной ссылки;
- выход блока ИЛИ подсоединяется ко всем входам блока перекрестной ссылки;
- выполняется разрешение всех внутренних флагов, для текущих путей устанавливаются ссылки.

### **2.3.4. Файл. Преобразование ФБД в РКС**

Функция преобразования предназначена для того, чтобы преобразовать коммутационную программу из функциональной блок-схемы в релейно-контактную схему.

Преобразование из функциональной блок-схемы в релейно-контактную схему осуществляется с использованием следующих правил:

- блок И преобразуется в схему с последовательными контактами;
- блок ИЛИ преобразуется в схему с параллельными контактами;
- комментарии к базовым функциям не применяются в релейно-контактной схеме, поскольку базовая функция преобразуется в множество контактов. Следовательно, комментарий не может быть назначен однозначно;
- в релейно-контактной схеме комментарии назначаются всем контактам данного входа;
- задаваемые пользователем комментарии не включаются, поскольку их положение в коммутационной программе не может быть задано на блоках;
- блоки, исключаящее ИЛИ, должны преобразовываться в соответствующие схемы релейно-контактной логики, включающей положительные и отрицательные контакты.

#### *Примечание*

При выполнении преобразования суммарное число блоков в коммутационной программе может иногда увеличиваться. Это может привести к превышению максимально допустимого числа блоков в вашем устройстве. Следовательно, преобразование функциональной блок-схемы в релейно-контактную схему не всегда возможно.

Способ устранения: В меню «Сервис > Выбор» оборудования выберите серию оборудования OBA6. Начните преобразование в релейно-контактную схему. Затем при помощи «Сервис > Определить LOGO!», отобразите серии аппаратуры, совместимые с коммутационной программой.

## **2.4. Меню «Правка»**

### **2.4.1. «Правка -> Имена входов и выходов»**

Эта команда позволяет вводить имена клемм входов и выходов (имена соединителей). Вызовите меню «Сервис -> Параметры -> Экран» для задания того, следует ли отображать имена соединителей на интерфейсе программирования.

Вызовите команду «Сервис -> Параметры -> Печать» для открытия диалога и задания того, следует ли включать имена соединителей



и перечень соединений в распечатываемую копию коммутационной программы.

### **2.4.2. «Правка -> Свойства блока»**

Эта команда отображает свойства блока, выбранного в интерфейсе программирования. Диалоговый блок свойств блока содержит несколько вкладок. У каждого блока имеется вкладка «Комментарий», в которую можно ввести соответствующие комментарии о блоке. У некоторых блоков имеется вкладка «Параметр», в которой вы можете описать конкретные параметры блока. Вкладка «Эмуляция» позволяет задать параметры для эмуляции.

### **2.4.3. «Правка -> Свойства блока» (все блоки)**

Эта команда открывает окно, содержащее два раздела. В левом разделе отображаются все блоки, используемые в программе. Нажмите блок для просмотра его соответствующих параметров в правом столбце. Можно отредактировать эти параметры и задействовать изменения нажатием кнопки «Применить» (рис. 2.6).

Измененный блок отображается, синим в перечне выбора. Все изменения отменяются при нажатии кнопки «Отмена». В случае отметки блока «Выбрать блок из чертежа» при последующем выборе блока из перечня они будут подсвечиваться на электрической схеме.

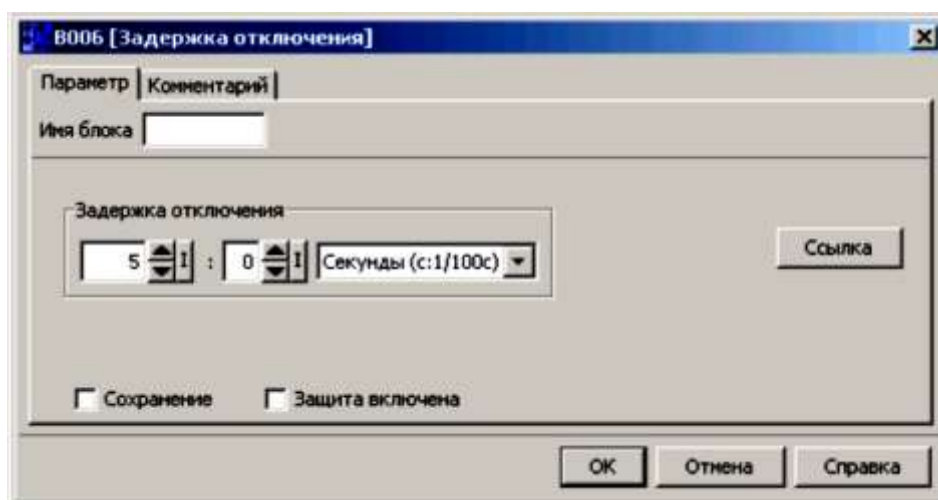


Рис. 2.6. Меню свойства блока задержки отключения

### **2.4.4. «Правка -> Разрезать соединения»**

Можете выбрать ручное или автоматическое разрезание соединений при помощи инструмента Разрезать/Связать.

В этом диалоге можно задать один из следующих типов разрезаемых соединений, или оба типа:

- соединения, проведенные через блок;
- соединения, выходящие за конфигурируемую длину.

Подтвердите выбор нажатием на кнопку ОК, и программа LSC выполнит разрезание соединений соответствующим образом.

## **2.5. Меню «Сервис»**

### **2.5.1. Меню «Инструменты – Обзор»**

Меню параметров содержит следующие команды меню:

- передача (в контроллер);
- определить LOGO!;
- выбор оборудования;
- эмуляция;
- параметры эмуляции;
- оперативное тестирование;
- подключить/отключить модем;
- параметры.

### **2.5.2. «Сервис. Определить модуль LOGO!»**

При нажатии этой команды меню программа LSC рассчитывает минимальные требования к модулям LOGO! для коммутационной программы LOGO!. Результат демонстрируется в строке состояния. Информационное окно отображает все версии системы, в которые может быть загружена программа. Вы также можете воспользоваться функциональной клавишей [F2] для выполнения этой команды меню.

### **2.5.3. «Сервис. Выбор оборудования»**

Есть два варианта выбора модулей ПЛК при создании программы:

1. Создать программу и затем определить требуемые устройства LOGO! при помощи диалога «Сервис -> Определить LOGO!»
2. Сначала определить версию LOGO, для которой создаётся программа, вызвав диалог «Сервис -> Выбор оборудования».

Диалог выбора устройства показывает, какие блоки и ресурсы памяти доступны. Если уже создали коммутационную программу или использовали некоторые блоки, то диалог выбора оборудования предложит только те устройства LOGO, которые могут работать с текущими выбранными блоками.

### **2.5.4. «Сервис. Эмуляция»**

Режим эмуляции программы позволяет провести тестирование программы и изменить ее параметры. Это обеспечивает загрузку пол-

ностью функционирующей и оптимизированной программы в модуль LOGO!.

### **Режим эмуляции**

Нажмите значок «Эмуляция» (см. табл. 2.1) на панели инструментов программирования для запуска эмуляции. При этом программа переключается в режим эмуляции (см. рис. 2.5).

### **Эмуляция**

Активный значок подсвечивается в режиме эмуляции. При повторном нажатии на значок «Эмуляция» или на любой другой значок на панели инструментов программирования вы выходите из режима эмуляции и открываете режим другого инструмента (например, инструмента для выбора или вставки блоков). При запуске режима эмуляции выполняется проверка программы, и результаты выводятся в окно информации.

#### *2.5.4.1. Параметры эмуляции: Входные функции*

Выберите команду «Сервис – Параметры эмуляции» для конфигурирования реакции на вход. В диалоге отображаются только те входы, которые фактически используются вашей коммутационной программой. У вас есть четыре варианта выбора:

- выключатель;
- кнопка без фиксации;
- частота (не для аналоговых входов);
- аналоговый (только для аналоговых входов).

### **Выключатель**

Переключатель фиксируется при воздействии на него и освобождается при повторном воздействии.

### **Кнопка без фиксации**

Кнопка без фиксации действует только при удержании ее нажатой. Контакт размыкается сразу после отпускания кнопки. Можно задать для кнопок функцию замыкания или размыкания цепи.

### **Частота (не для аналоговых входов)**

Частота частотного входа может быть предустановлена или изменена в процессе выполнения эмуляции. Частотный вход является специальным случаем в рассматриваемом контексте, так как он является полезным только совместно со специальным функциональным блоком порогового выключателя.

### **Аналоговый (только для аналоговых входов)**

Можно выполнить предварительную установку значения аналогового входа или изменить его во время работы в режиме эмуляции. Единицы аналогового значения должны устанавливаться в соответствии с используемой по умолчанию переменной технологического процесса.

Диапазон соответствует заданному диапазону измерений, если выбран вариант «Автоматический диапазон».

Диапазон значений соответствует в этом случае диапазону измерений функции, связанной со входом.

Аналоговый вход представляет собой специальный случай, поскольку он должен использоваться только для аналоговых специальных функциональных блоков.

### **Столбец названия**

Цифровые входы обозначаются **I**.

Аналоговые входы обозначаются **AI**.

### **Настройки**

При сохранении коммутационной программы вместе с ней сохраняются настройки эмуляции схемы. В режиме эмуляции можно нажать на цифровой вход в перечне входных переключателей для изменения их настроек. Из раскрывающегося перечня выберите «Переключатель», «Кнопка без фиксации (замыкающая)», «Кнопка без фиксации (размыкающая)» или «Частота».

### **Вопросы для самоконтроля**

1. Какие типы файлов проектов программ относятся к языкам FBD и LAD?
2. Какой тип файла предназначен для сохранения документации проекта?
3. Для чего предназначено преобразование файлов проектов, написанных на языках FBD и LAD?
4. Для чего нужна функция «Разрезать -Соединить»?
5. Какую информацию предоставляет меню «Сервис. Выбор оборудования»?
6. Какие тесты можно проводить в режиме эмуляции программы?

### Глава 3. ОСНОВЫ ЗАПИСИ ПРОГРАММ

Рассмотрим особенности записи программ на двух языках стандарта МЭК-61131-3: функциональных блок-схем (ФБД\FBD) и релейно-контактных схем (РКС\LAD).

Функциональную блок-схему (FBD) представления кода программы удобна пользователям, знакомым с логическими блоками, реализующими функции булевой алгебры.

Релейно-контактную схему (LAD) реализации программы оптимально применять пользователям, которые привыкли работать с принципиальными электрическими схемами.

При помощи среды разработки можно быстро начать создание программы. Сначала составляется программа, а затем в среде разработки рассчитывается минимальная версия модулей LOGO ПЛК, необходимых для аппаратной реализации проекта. Тип модулей зависит от ряда факторов:

- требуемого числа входов и выходов;
- требований программы к памяти;
- применения конкретных специальных функциональных блоков.

#### 3.1. Запись программы в среде разработки

##### Этапы записи коммутационной программы:

1. Создание нового проекта программы.
2. Выбор блоков.
3. Размещение блоков.
4. Выполните конфигурирование блоков и размещение комментариев к блокам и частям программы.
5. Соединение блоков программы.
6. Выполнение оптимизации программы.
7. Сохранение программы в проекте. Запись программы в файле документации проекта (\*.pdf).

##### 3.1.1. Разработка программы

После этапа формализации алгоритма управления переходим к записи программы в среде LSC. Сначала вводим данные своего проекта при помощи диалога меню «Свойства» (рис. 3.1).

Для начала создания новой (коммутационной) программы выбираем значок [Создать файл] на стандартной панели инструментов. При этом по умолчанию открывается редактор функциональных блок-схем, и можно приступить к созданию новой коммутационной программы в новом окне.

Выбираем маленькую стрелку на правой стороне значка [Создать файл] для открытия редактора релейно-контактной схемы или функциональной блок-схемы.

Свойства из Принципиальная схема1

Общие | Комментарий | Статистика | Разметка страницы | Параметр

Разработчик:

Название проекта:

Наименование уста...

Заказчик:

Схема №:

Проверено:

Компания:

Версия:

☐ Показать для нового файла

OK Отмена Справка

Рис. 3.1. Ввод данных о проекте

### Размещение блоков

Следующий шаг заключается в размещении требуемых блоков в интерфейс программирования. Помимо стандартных и специальных функций также размещаются входные и выходные блоки. На этой стадии достаточно разместить блоки приблизительно в те положения, которые представляются соответствующими для выполнения последующих соединений. Точное размещение выполняется после всех соединений (рис. 3.2–3.3).

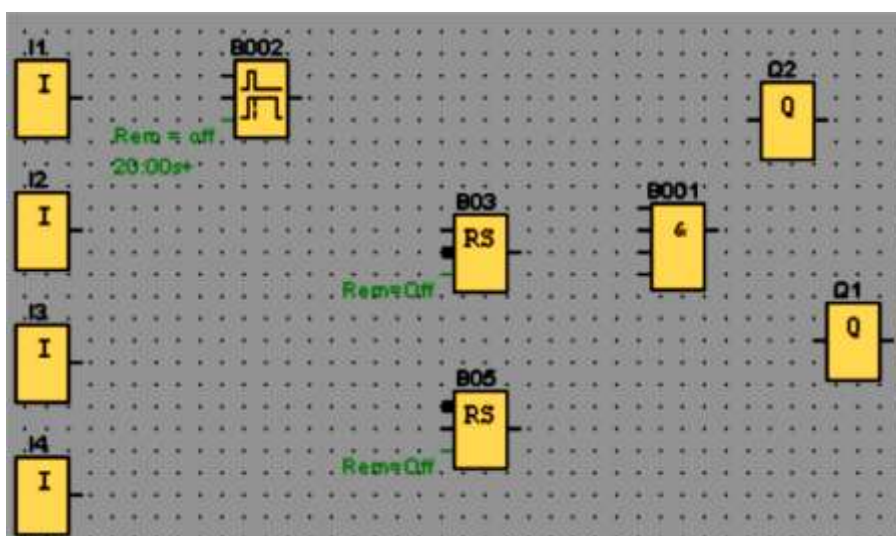


Рис. 3.2. Редактор функциональных блок-схем



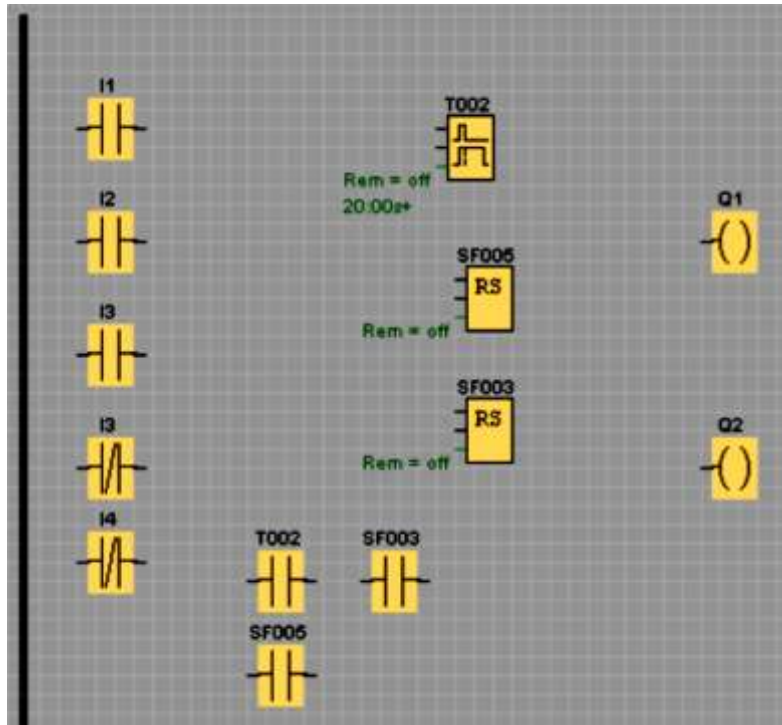


Рис. 3.3. Редактор релейно-контактных схем (PKC)

### Соединение блоков

Выполнить соединение блоков путем построения соединительной линии, начинающейся от выхода исходного блока и завершающейся у входа целевого блока (рис. 3.4, 3.5). При этом имеется преимущество, заключающееся в том, что демонстрируется имя соединителя при подключении входа, что особенно удобно при работе с различными соединителями специальных функциональных блоков.

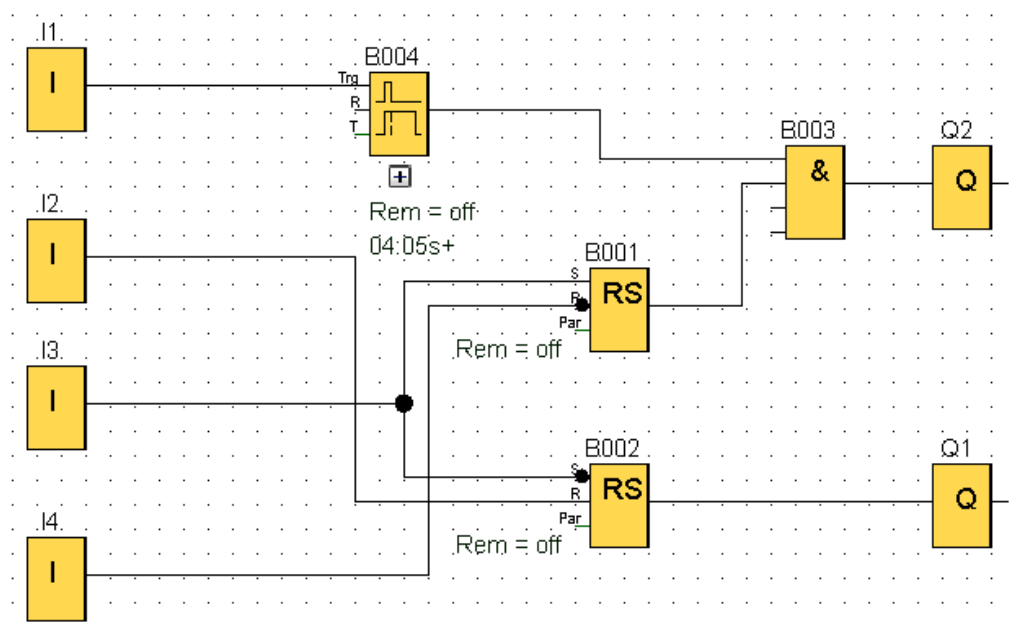


Рис. 3.4. Редактор функциональных блок-схем (FBD)

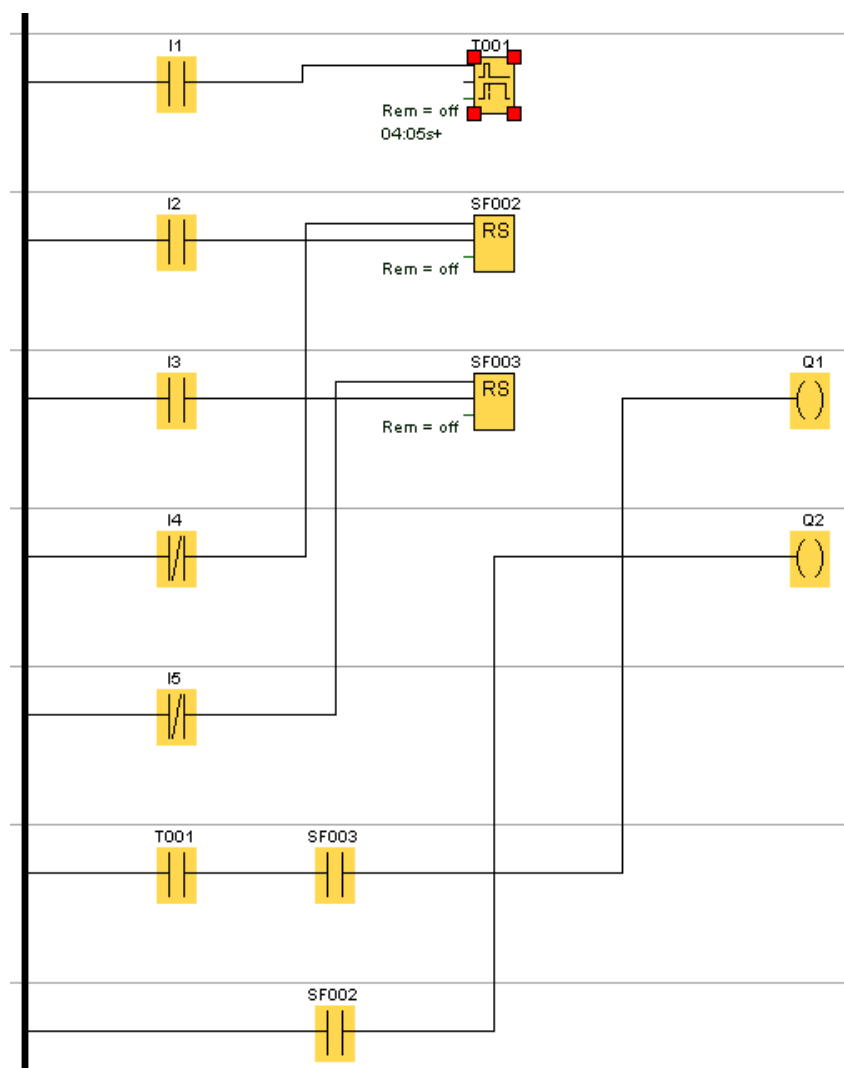


Рис. 3.5. Редактор релейно-контактных схем (LAD)




### 3.1.2. Выбор блоков

Первый шаг при программировании схемы алгоритма заключается в выборе блоков для схемы (обратите внимание на то, что не все блоки доступны при всех обстоятельствах).

Необходимо определить порядок, в котором необходимо вставлять блоки ввода/вывода и стандартные блоки/специальные функциональные блоки.

Следует установить и разместить блоки входных и выходных сигналов согласно таблице сигналов.

В разделе Со на панели инструментов программирования находятся константы и клеммы (только в редакторе релейно-контактной схемы), т.е. выбор входов/выходов и постоянных сигналов. В разделе GF находятся базовые логические функции булевой алгебры: стандартные цифровые логические блоки. В разделе SF находятся специальные функции. Можно вызвать соответствующие группы функций при помощи функциональных клавиш:

	константы / клеммы или [F6];
	[F8] Специальные функциональные блоки;
	только в редакторе функциональных блок-схем или [F7] Базовые функции.

Это быстрый способ выбора блоков и размещения их в коммутационной программе.

### 3.1.3. Размещение блоков

Выбрать значок группы, содержащей требуемый блок, или, как вариант, нажмите функциональную клавишу (рис. 3.6). Все блоки, принадлежащие выбранной группе функций, отображаются под интерфейсом программирования.

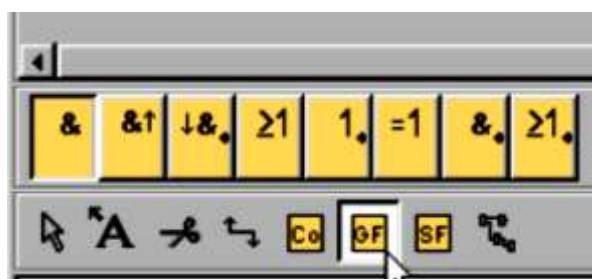


Рис. 3.6. Пример для редактора функциональных блок-схем

Можно вставить выбранную функцию на интерфейсе программирования простым нажатием кнопки мыши. Первая функция группы устанавливается по умолчанию, и можно выбрать другие функции мышью перед их размещением (рис. 3.7).

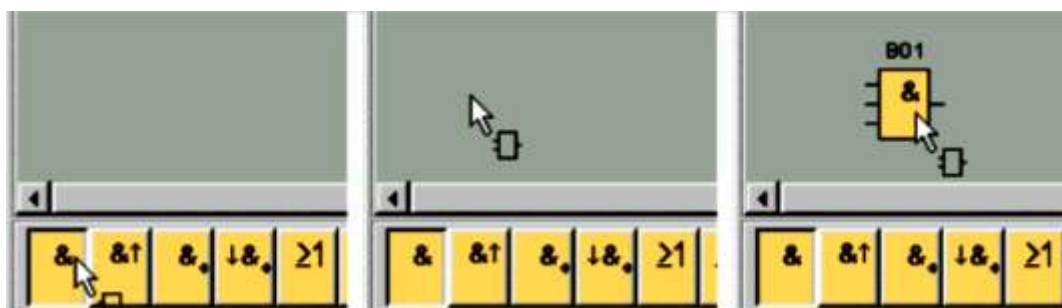


Рис. 3.7. Пример для редактора функциональных блок-схем

Не следует сразу выполнять точное выравнивание блоков, это следует делать после выполнения соединений между ними и ввода комментариев в коммутационную программу.

### 3.1.4. Конфигурирование свойств блоков

Нажатие правой кнопкой мыши над объектом открывает меню быстрого выбора, предлагающее различные параметры редактирования объекта. Параметры выбора зависят от выбранного объекта. Объекты состоят не только из блоков и соединительных линий, но и интерфейса программирования и панелей инструментов.

#### Конфигурирование блоков

Двойное нажатие левой кнопкой мыши вызывает конфигурирование свойств блока (рис. 3.8). *Меню настройки блока функции задержки на включение*. Диалог свойств содержит вкладку «Комментарий», а также вкладки различных параметров для специальных функциональных блоков и для некоторых из базовых функций и констант и соединительных элементов. Здесь задаются значения и настройки для блоков. Специальные функции могут быть определены по зеленому шрифту слева от блока интерфейса программирования.

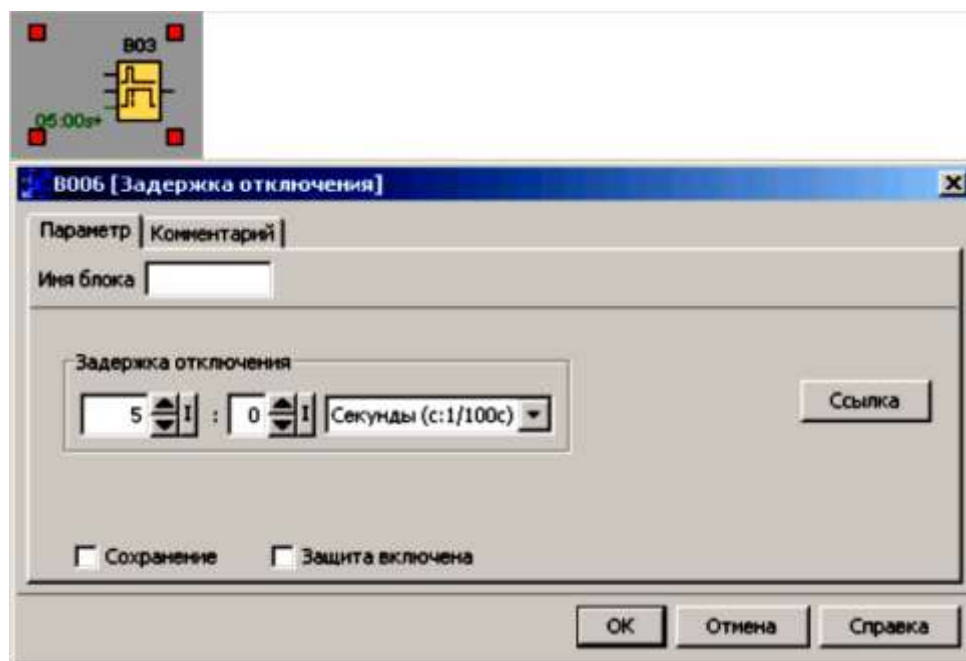


Рис. 3.8. Меню настройки блока функции задержки на включение

### 3.1.5. Соединение блоков

Для выполнения схемы необходимо выполнить соединения блоков. На панели инструментов программирования выберите значок соединения блоков. Пример для функциональных блок-схем (рис. 3.9).

#### Соединительный элемент

Наведите указатель мыши на соединительный элемент блока. Нажмите левую кнопку мыши и удерживайте ее нажатой. Переместите

указатель мыши от исходного соединительного элемента к целевому соединительному элементу.

Отпустите кнопку мыши. Программа LSC выполняет соединение двух клемм. Программа LSC предлагает вам дополнительную функцию соединения блоков нажатием правой кнопкой мыши на вход или выход блока. В меню быстрого выбора нажмите команду меню соединить с блоком. При этом вызывается перечень выбора, содержащий все блоки, с которыми может быть осуществлено соединение. Нажмите на соответствующий целевой блок.

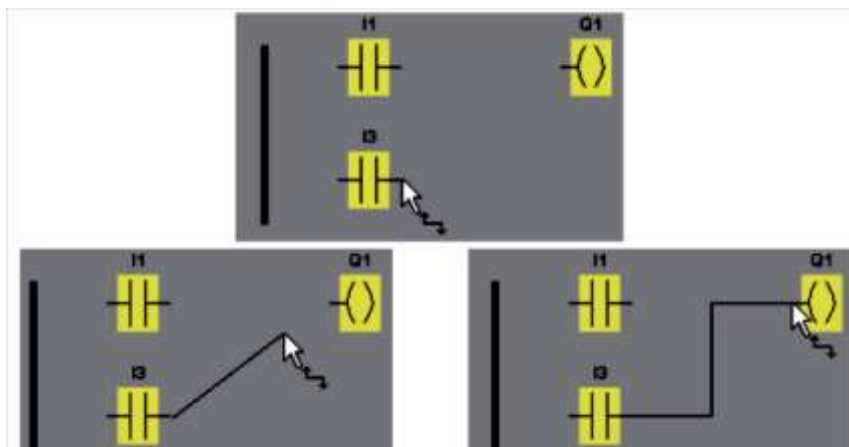


Рис. 3.9. Пример для языка релейно-контактных схем (РКС)

Программа LSC выполнит построение соединительной линии (рис. 3.10). Этот метод является особенно полезным для соединения источника с целевым блоком, расположенным на большом расстоянии друг от друга, на поле редактора программирования.

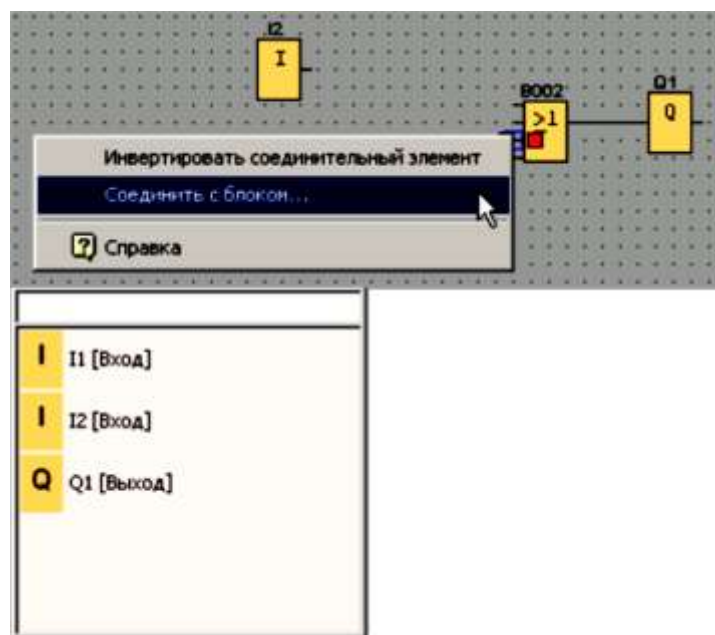


Рис. 3.10. Метод соединения блоков

Примечание для редактора релейно-контактных схем:

Не забудьте соединить входы и выходы к сборной шине на левой кромке окна редактора.

### **Соединение блоков**

После выполнения соединительной линии от выхода ко входу, или наоборот, раскрывается подсказка у указателя мыши с информацией о соединении. Отпустите кнопку мыши для соединения линии с указанным входом.

### **Правила соединения блоков**

Соединение блоков осуществляется с использованием следующих правил:

- можно соединить один вход с несколькими выходами;
- нельзя соединять несколько входов с одним выходом;
- можно соединять между собой входы и выходы в одном проекте коммутационной программы. *Рекурсия не допускается. Выполняется соединение флага или выхода, если это необходимо;*
- специальные функциональные блоки также имеют зеленые «соединители». Они не представляют соединительные выводы, но используются вместо этого для назначения значений параметров;
- не допускается подключение аналоговых входов и выходов к цифровым входам и выходам.

### **3.1.6. Определение требуемого ресурса оборудования. Доступность блоков**

#### **Параметры оборудования по умолчанию**

Пространство памяти и серии устройств вашей системы LOGO! определяют следующее:

- сколько блоков вы можете использовать в коммутационной программе;
- какие блоки имеются в вашем распоряжении для создания коммутационной программы.

По умолчанию выбирается самое последнее поколение системы LOGO! После создания коммутационной программы можно вызвать команду меню «Сервис -> Определить LOGO!» или нажать функциональную клавишу [F2] для отображения окна информации, указывающего на то, какие устройства LOGO! доступны для составления вашей коммутационной программы. Блоки, не доступные для выбранной вами системы LOGO!, отображаются серым.

### **Оптимизация программы**

Если выяснится в ходе создания программы, что устройство LOGO! не может выполнить программу, то необходимо сначала вер-



наться в начало и учесть функциональные ресурсы, предлагаемые вам семейством LOGO! Необходимо, в частности, заменить блоки, интенсивно использующие память, на структуры, состоящие из нескольких блоков, которые совместно требуют меньшего объема памяти.

Если различные попытки оптимизации не приведут к получению результатов, то можно воспользоваться дополнительным устройством LOGO! или попытаться оптимизировать / упростить функции системы.

### **Вопросы для самоконтроля**

1. Какой стандарт регламентирует применение языков программирования для ПЛК (англ. PLC)?
2. Как называются языки программирования, используемые в инструментальной среде LSC?
3. От каких основных технических характеристик зависит выбираемый тип модулей ПЛК LOGO?
4. Какое меню заполняется для синтеза файла документации при создании нового проекта программы?
5. Что находится в разделах: Co, GF, SF?
6. Как производится настройка свойств блока?
7. Какие Вы знаете правила соединений блоков?
8. От чего зависит доступность блоков для программирования?

## Глава 4. РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ АВТОМАТИЗАЦИИ НА ОСНОВЕ ПЛК

Рассмотрим принцип решения задач автоматизации на этапе разработки прикладных программ для ПЛК.

Технические характеристики комплекса аппаратного и программного обеспечения микропроцессорной системы (МП) ПЛК требуют своих методов проектирования и разработки программного обеспечения, для этого рассмотрим ряд примеров, иллюстрирующих специфику анализа задач программирования и разработки программ на языке ФБД (FBD).

### 4.1. Принципы разработки программ

Пример. Нагрузка (лампа индикации) E1 включается и выключается с помощью выключателей (S1 **OR** S2) **AND** S3. Реле K1 срабатывает, когда замкнуты S1 или S2, а также S3. На рис. 4.1 представлена электрическая схема и программа (рис. 4.2) предложенной логической функции.

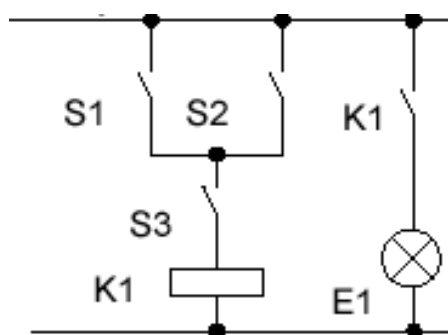


Рис. 4.1. Электрическая схема реализации логической функции

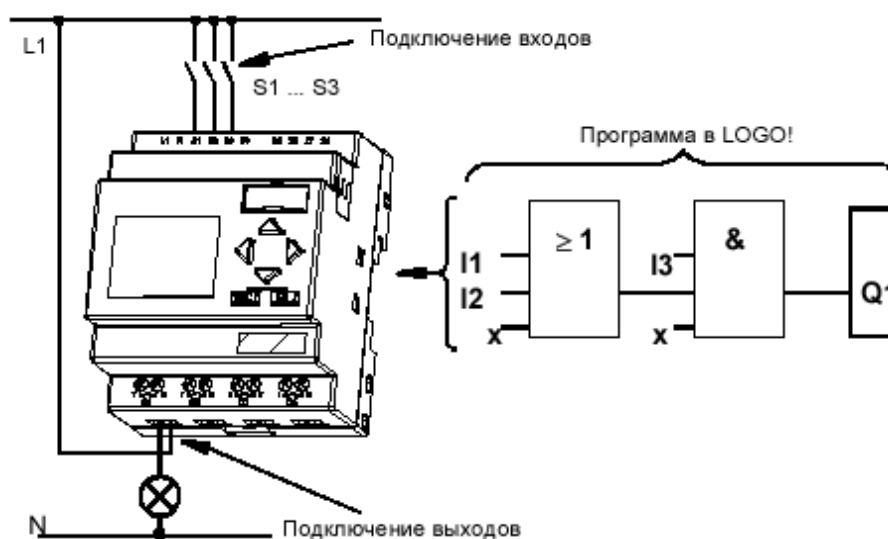


Рис. 4.2. Представление коммутационной схемы и программы

В LOGO! данная схема реализуется следующим образом: схема создается соединением друг с другом блоков и соединительных элементов. На рис. 4.2 показано это соединение.

Реализация схемы всегда начинается с выхода схемы. Выходом является нагрузка или реле, которым необходимо управлять. Схема преобразуется пользователем в блоки. Для этого нужно пройти по схеме от выхода к входу.

**Шаг 1:** На выходе Q1 имеется последовательное соединение нормально открытого контакта S3 с другим компонентом схемы. Последовательное соединение соответствует блоку AND [И]. На рис. 4.3 представлено это соединение.

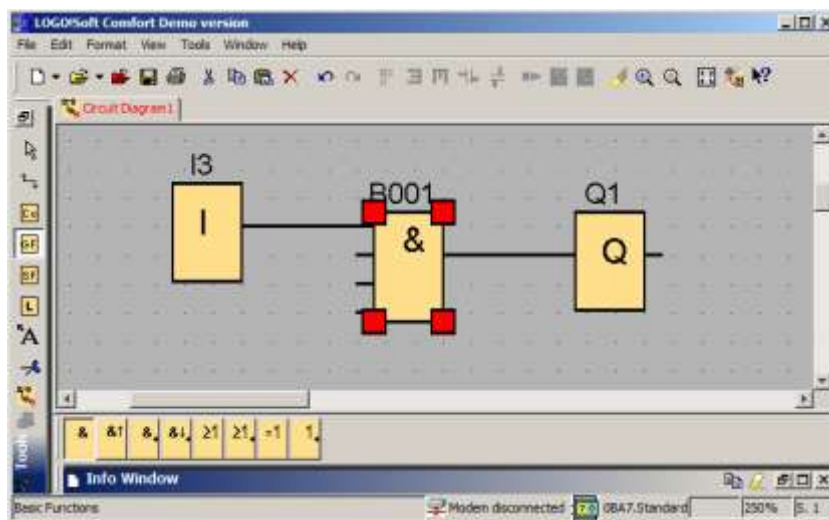


Рис. 4.3. Последовательное соединение

**Шаг 2:** S1 и S2 соединены параллельно. Параллельное соединение соответствует блоку OR [ИЛИ], на рис. 4.4 представлено параллельное соединение.

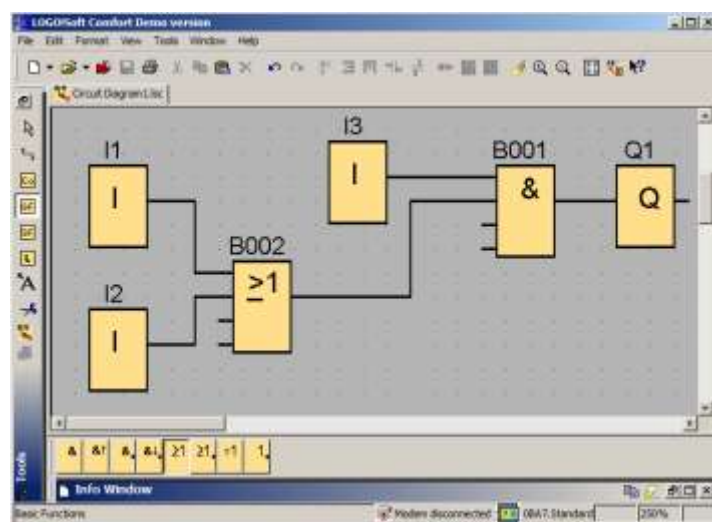


Рис. 4.4. Параллельное соединение

Посмотрим теперь на следующую схему параллельного соединения двух ключей.

#### 4.1.1. Пример разработки программы системы управления

В проекте «Умный дом» наряду со снабжением питьевой водой применяется дождевая вода. Вода накапливается в подземном хранилище-ёмкости, используется для технических нужд вместо питьевой, помогая защищать окружающую среду.

##### Описание системы

Дождевая вода собирается в ёмкость. Из ёмкости она подается насосом в соответствующую систему водоснабжения. Из нее она может отбираться точно так же, как и питьевая вода. Для поддержания работоспособности системы в нее может подаваться питьевая вода, если в баке закончилась техническая.

Можно разработать схему регулирования, отвечающую требованиям такой установки. На приведенном ниже рис. 4.5 показана функциональная схема автоматизации системы снабжения технической водой.

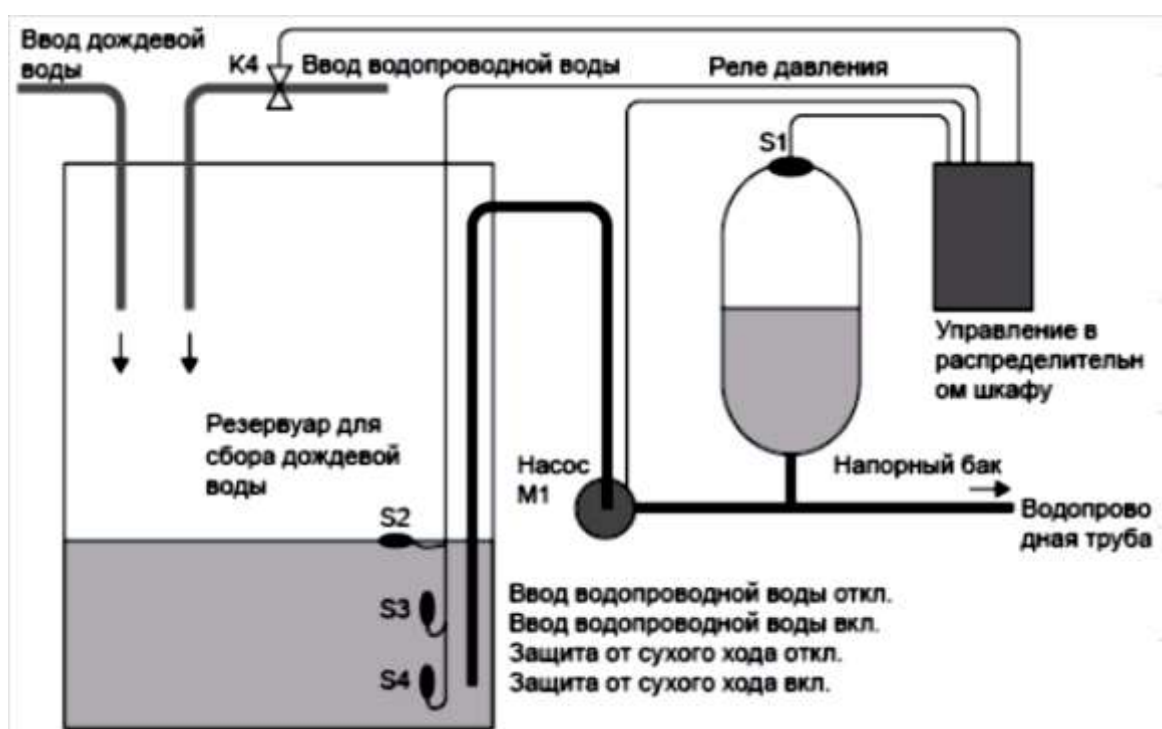


Рис. 4.5. Функциональная схема автоматизации системы регулирования подачи воды

##### Требования к системе регулирования

1. Техническая вода должна быть в постоянном наличии. Система аварийного управления должна, например, выполнять переключение на использование питьевой воды при низком уровне технической.

2. При переключении на использование питьевой воды должно предотвращаться проникновение технической воды в систему снабжения питьевой водой.

3. Насос должен отключаться в случае низкого уровня технической воды в баке (защита от холостого хода).

#### 4.1.2. Алгоритм управления

Насос и электромагнитный клапан управляются на основе сигналов от реле давления и трех поплавковых выключателей, установленных в баке для технической воды (рис. 4.6). Насос должен включаться при падении давления в баке ниже минимального. По достижении рабочего давления насос выключается по истечении времени задержки в несколько секунд. Время задержки выключения предотвращает частое включение и выключение насоса при продолжительном отборе воды.

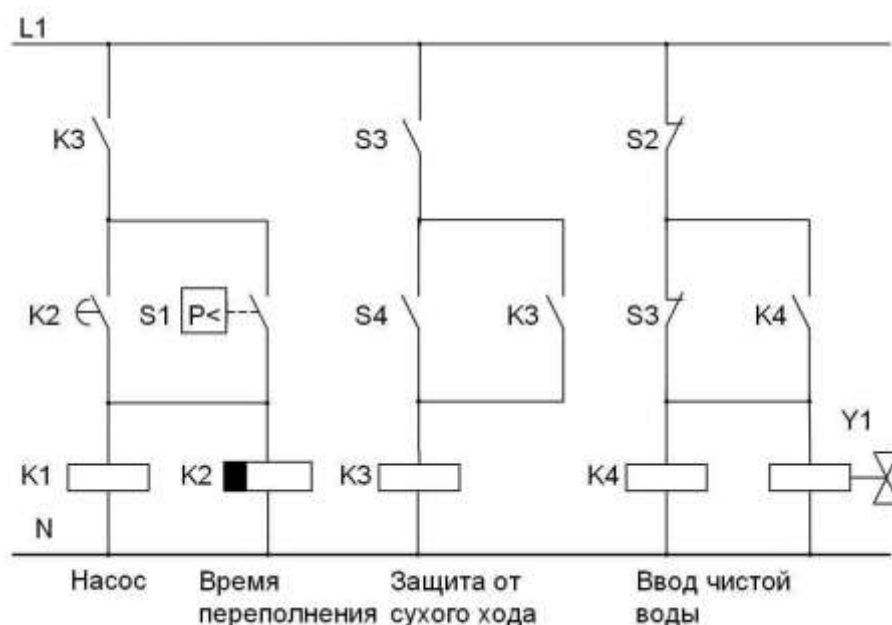


Рис. 4.6. Релейная электрическая принципиальная схема управления

#### 4.1.3. Аппаратное решение автоматической системы регулирования

##### Подключение внешних устройств

Помимо устройства ПЛК LOGO!, требуется реле давления и поплавковые датчики для управления насосом. (Если используется 3-фазный двигатель переменного тока, то для управления насосом требуется пускатель двигателя. Если в системе будут применены однофазные насосы переменного тока, то устанавливается пускатель двигателя в случае, если ток двигателя переменного тока превышает коммутационный ток реле K1(Q1). Управление электромагнитным

клапаном, как правило, осуществляется непосредственно из-за его низкой потребляемой мощности.

Далее вопросы электрической коммутации датчиков и исполнительных устройств должны учитываться при выполнении проектов электрических – принципиальных схем систем управления, как на рис. 4.7.

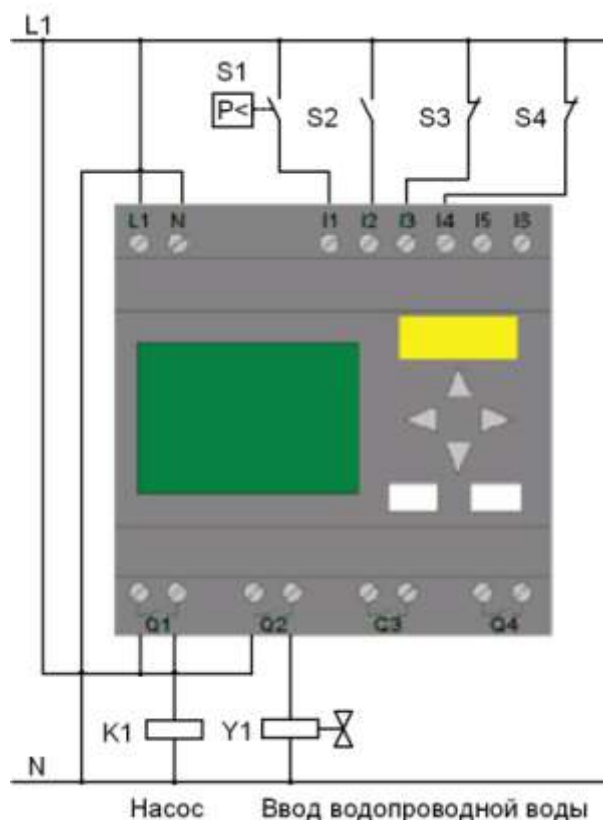


Рис. 4.7. Электрическая схема подключения модуля ПЛК LOGO!

#### 4.1.4. Блок-схема структуры управления

Блок-схема отображает порядок соединения функциональных блоков управления насосом и электромагнитного клапана (см. рис. 4.8). Её структура соответствует структуре принципиальной схемы, показанной на рис. 4.7.

#### Параметры

В случае применения в других областях можно включить дополнительные функции, которые в противном случае потребовали бы применения дополнительных аппаратных средств:

- разрешение на работу насоса в определенное время, например – в летние месяцы или в определенное время суток;
- индикация приближающегося или фактического недостатка воды;
- контроль температуры;
- сообщения о системных ошибках.





Рис. 4.8. Блок-схема программы на языке ФБД

### Вопросы для самоконтроля

1. С какой части начинается реализация схемы программы?
2. Какими схемами и техническим данными пользуется программист для разработки программ для ПЛК?
3. Как называется язык программирования, графически наиболее полно соответствующий электрическим принципиальным схемам на основе реле?
4. В каких случаях применяются устройства-пускатели?
5. Какая документация служит для разработки электрических схем соединения ПЛК и периферийных устройств?

### 4.2. Методы формализованного подхода к разработке программ

Системы автоматического управления объектами и процессами, реализованные на ПЛК (PLC), имеют как правило высокие технико-экономические показатели.

Перед инженерами-разработчиками программного обеспечения ПЛК стоит задача синтеза оптимальных алгоритмов, в первую очередь не содержащих алгоритмических ошибок, поэтому для получения кода исходной программы необходимо выполнить ряд этапов:

- 1) подробное описание задачи (технологического процесса управления или мониторинга);
- 2) анализ задачи;
- 3) инженерную интерпретацию задачи, желательно с привлечением того или иного аппарата формализации (граф автомата, сети Петри, матрицы состояний и связности и т.п.);
- 4) разработку общей блок-схемы алгоритма (БСА) работы контроллера;

- 5) разработку детализированных БСА отдельных процедур, выделенных на основе модульного принципа составления программ;
- 6) детальную проработку интерфейса контроллера и внесение исправлений в общую и детализированные БСА;
- 7) контроль требуемого объёма и распределения памяти программ и данных;
- 8) анализ быстродействия программы и системы в целом в реальном времени;
- 9) запись кода исходной программы.

В результате работы по трем первым пунктам данного перечня получают так называемую функциональную спецификацию прикладной программы ПЛК, в которой основное внимание уделяется детализации способов формирования входной и выходной информации.

На языке схем алгоритмов разработчик описывает метод, выбранный для решения поставленной задачи. Довольно часто бывает, что одна и та же задача может быть решена различными методами. Способ решения задачи, выбранный на этапе ее инженерной интерпретации, на основе которого формируется БСА, определяет не только качество разрабатываемой прикладной программы, но и качественные показатели конечного изделия.

Разработка БСА очень похожа на разработку аппаратных средств систем автоматики и обработки данных. *В основу разработки БСА положена та же самая процедура модульного проектирования, которая традиционно используется разработчиками аппаратных средств.* Отличие состоит в том, что при разработке аппаратных средств в качестве «строительного» материала используются логические схемы триггеры, регистры и другие интегральные элементы, а при создании программного обеспечения на языке ФБД разработчик оперирует функциональными блоками, реализующими логические и специальные функции.

Так как алгоритм есть точно определенная процедура, предписывающая ПЛК однозначно определенные действия по преобразованию «сырых» исходных данных в обработанные выходные данные, то разработка БСА требует предельной точности и однозначности используемой атрибутики: символических имен переменных, констант (уставок), подпрограмм (модулей), символических адресов таблиц, портов ввода/ вывода и т.п. Основное внимание при разработке БСА следует уделить тому разделу функциональной спецификации прикладной программы, в котором приводится описание функций сопряжения ПЛК с объектом управления (ОУ). (Это описание для успешной разработки программного обеспечения должно быть детализировано вплоть до электрических и временных характеристик каждого входного и выходного сигнала или устройства.)

Основа разработки программы ПЛК заключается в использовании метода декомпозиции, при котором вся задача последовательно разделяется на меньшие функциональные модули, каждый из которых можно анализировать, разрабатывать и отлаживать отдельно от других. При выполнении прикладной программы в ПЛК управление без всяких двусмысленностей передается от одного функционального модуля к другому. Схема связности этих функциональных модулей, каждый из которых реализует некоторую процедуру, образует общую (или системную) БСА прикладной программы. Это разделение задачи на модули и submodule выполняется последовательно до такого уровня, когда разработка БСА модуля становится простым и понятным делом. Метод последовательной декомпозиции обладает достаточной гибкостью, что позволяет привести степень детализации БСА в соответствие со сложностью процедуры. Не следует стесняться при выполнении декомпозиции дойти до модулей, которые почти тривиальны.

*Структурное программирование* есть процесс построения прикладной программы из строгого набора программных модулей, каждый из которых реализует определенную процедуру обработки данных. *Программные модули должны иметь только одну точку входа и одну точку выхода.* Только в этом случае отдельные модули можно разрабатывать и отлаживать независимо, а затем объединять в законченную прикладную программу с минимальными проблемами их взаимосвязей. В международном стандарте на программный продукт HIPO (Hierarchy-Input-Process-Output) декларируется аналогичный подход к разработке прикладных программ.

Разработка БСА функционального модуля программы имеет ярко выраженный итеративный характер, т.е. требует многократных проб, прежде чем возникает уверенность, что алгоритм реализации процедуры правильный и заверченный. Вне зависимости от функционального назначения процедуры при разработке ее БСА необходимо придерживаться следующей очередности работы:

- 1) *Определить, что должен делать модуль* (это уже было сделано при разработке системной БСА, но теперь разработчик имеет дело с фрагментом прикладной программы, а не с целой программой, и, следовательно, он может потребоваться для определения и уточнения целевого назначения процедуры).

- 2) *Определить способы получения модулем исходных данных* (от датчиков через систему ввода или из памяти).

- 3) *Определить необходимость какой-либо предварительной обработки введенных исходных данных* (маскирование, сдвиг, масштабирование, перекодировка). Если до использования «сырых» данных требуется их предобработка, то в состав БСА включаются соответствующие функции.

4) *Определить способ преобразования входных данных* в требуемые выходные. Используя операторы процедур и условные операторы принятия решения, отобразить на языке БСА выбранный метод содержательной обработки исходных данных.

5) *Определить способы выдачи из модуля обработанных данных* (передать в память, или в вызывавшую программу, или в порты вывода информации). Необходимые действия отобразить в БСА.

6) *Определить необходимость какой-либо постобработки выводимых данных* (изменение формата, перекодирование, масштабирование, маскирование). Ввести в БСА операторы подготовки данных для вывода из Модуля.

7) Вернуться к п. 1 настоящего перечня работ и *проанализировать полученный результат*. Выполнить итеративную корректировку БСА с целью сделать ее простой, логичной, стройной и обладающей четким графическим образом.

8) Проверить работоспособность алгоритма на бумаге путем подстановки в него действительных данных. Убедиться в его сходимости и результативности.

9) Рассмотреть предельные случаи и попытаться определить граничные значения информационных объектов, с которыми оперирует алгоритм, за пределами которых он теряет свойства конечности, сходимости или результативности. (Особое внимание при этом следует уделить анализу возможных ситуаций переполнения разрядности максимальных значений, изменения знака результата операции, деления на переменную, которая может принять нулевое значение.)

10) *Провести мысленный эксперимент по определению работоспособности алгоритма в масштабе реального времени*, когда стохастические события, происходящие в объекте управления, могут оказать влияние на работу алгоритма. При этом самому тщательному анализу следует подвергнуть реакцию алгоритма на возможные прерывания с целью определения критических функциональных блоков и ветвей (модулей) алгоритма. Кроме того, в ходе этого мысленного эксперимента следует проанализировать логику алгоритма с целью определения таких последовательных действий операторов, при выполнении которых МК может «не заметить» кратковременных событий в объекте управления. При обнаружении таких ситуаций в логику БСД следует внести коррективы.

Практика разработки программного обеспечения для программируемых контроллеров показала, что последовательное использование описанной поэтапной процедуры проектирования алгоритмов, составляющей основу метода структурного программирования, позволяет уверенно получать работоспособные прикладные программы.

### 4.2.1. Методы формализации алгоритмов

Процесс творческого освоения технических возможностей ПЛК может быть облегчен тем, что на начальном этапе проектирования ПЛК-систем можно с успехом использовать некоторые формализованные приемы синтеза комбинационных схем и цифровых автоматов на интегральных элементах.

К настоящему времени формализованные методы проектирования комбинационных и последовательностных (автоматных) схем на интегральных элементах низкой степени интеграции получили широкое теоретическое развитие и практическое применение. Некоторые классические приемы проектирования «схемной логики» могут быть применены для разработки ПЛК-систем. К таким приемам относятся прежде всего методы минимизации булевых функций и автоматных таблиц переходов/выходов методы описания работы последовательностных устройств на языке графов автоматов Мили и/или Мура.

**Булевы функции.** Законы функционирования отдельных устройств системы в процессе их разработки часто описывают на языке булевых функций или на языке релейно-контактных схем. Такие устройства реализуются на основе схем комбинационной логики или на основе релейной логики. Все эти устройства могут быть с успехом реализованы программными средствами в ПЛК. При этом следует иметь в виду, что программная реализация булевых функций в отличие от их реализации на комбинационных электронных схемах, как правило, не требует дополнительной аппаратуры, но сопряжена с потерей быстродействия.

В качестве примера рассмотрим комбинационную схему на языке ФБД, показанную на рис. 4.9а и реализующую булеву функцию от шести переменных

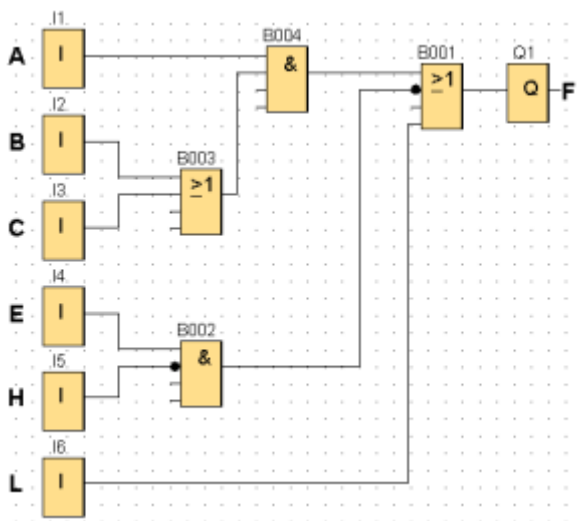
$$F = A (B \vee C) \vee E \wedge H \vee L.$$

Предположим, что группа датчиков объекта управления генерирует для ПЛК пять независимых двоичных сигналов: А и В от двух концевых переключателей; С, Е и Н – три бита от 3 – позиционного переключателя, установленного на пульте управления оператора. Пусть шестая переменная L – это флаг некоторого программного признака, например, признака не прохождения теста. Выходной сигнал F через порт подается на реле исполнительного механизма.

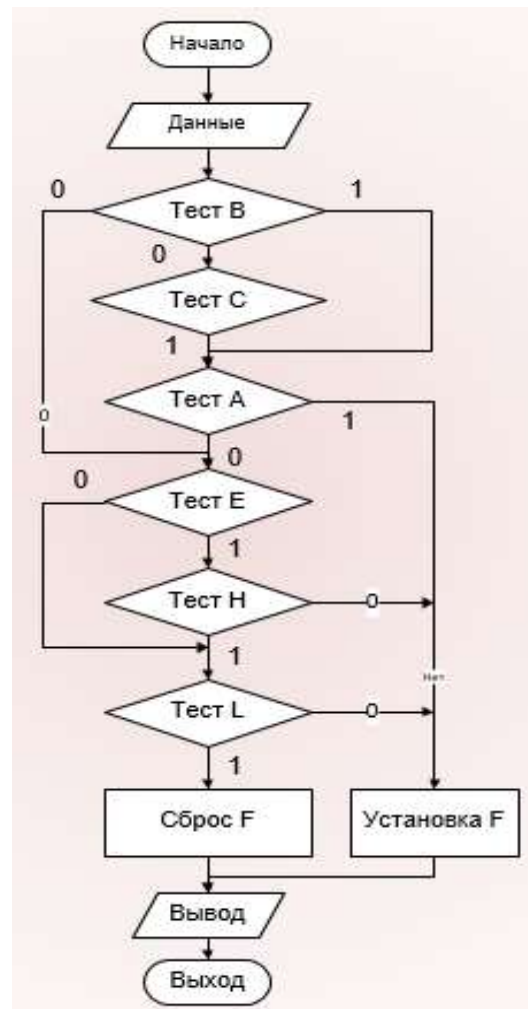
Сигналы от указанных датчиков подаются в строго определенные биты некоторого порта ввода информации. Следовательно, в процессе программной реализации заданной булевой функции необходимо после операции ввода обеспечить серию проверок состояния отдельных бит введенного байта и формирование значения выходного сигнала F.

На рис. 4.9б показана схема алгоритма программной реализации заданной булевой функции, учитывающая последовательность проверок, которая определяется скобочной формой записи функции  $F$ . Заметим, что при программной реализации булевых функций практически нет никаких ограничений на число переменных. Однако с ростом числа переменных возрастает время задержки формирования выходного сигнала  $F$ .

**Граф автомата.** Законы функционирования последовательностных схем задаются или таблицами переходов/выходов или в виде графа (диаграммы состояний) автомата. *Переход от графа проектируемой системы управления к БСА является ключевым этапом методики*, позволяющим отобразить таблицы состояний и входов/выходов автомата на языке программирования ПЛК.



а)



б)

Рис. 4.9. Реализация алгоритма:  
 а) комбинационная схема программы на языке ФБД;  
 б) блок-схема алгоритма программы

Сравнив фрагменты графа автомата Мили (автомат Мура здесь не рассматривается, так как эти две модели автомата однозначно связаны) и фрагмент БСА, показанные на рис. 4.10, можно заметить, что они эквивалентны. Здесь  $X$  и  $Z$  – двоичные векторы, принадлежащие множествам наборов входных  $x$  и выходных  $z$  сигналов, а  $Q$  – состояния автомата из множества возможных состояний  $q$ . Состояние автомата  $Q_i$  и выходной вектор  $Z_i$  являются функциями предыдущего автоматного состояния  $Q_{i-1}$  и текущего входного вектора  $X_i$ . Операторные прямоугольники БСА могут быть интерпретированы как вершины (устойчивые состояния) графа автомата (рис. 4.10б), а условные операторы – как дуги графа  $X_i/Z_i$ , т.е. переходы к иному автоматному состоянию. При этом операторный прямоугольник представляет собой фрагмент программы. В течение всего периода устойчивого состояния  $Q_i$  выходной вектор  $Z_i$  остается неизменным до тех пор, пока не изменит своего значения входной вектор  $X$ . Для этого в течение устойчивого состояния  $Q_i$  операторы алгоритма программной реализации автомата селектируют с заданным периодом дискретизации изменение под воздействием внешних (по отношению к ПЛК) входного вектора  $X$ . В результате установления факта изменения вектора  $X$  контроллер переходит к новому «устойчивому» автоматному состоянию  $Q_{i+1}$  и генерирует новый выходной вектор  $Z_{i+1}$ , значения которого полностью определены графом автомата (рис. 4.10а).

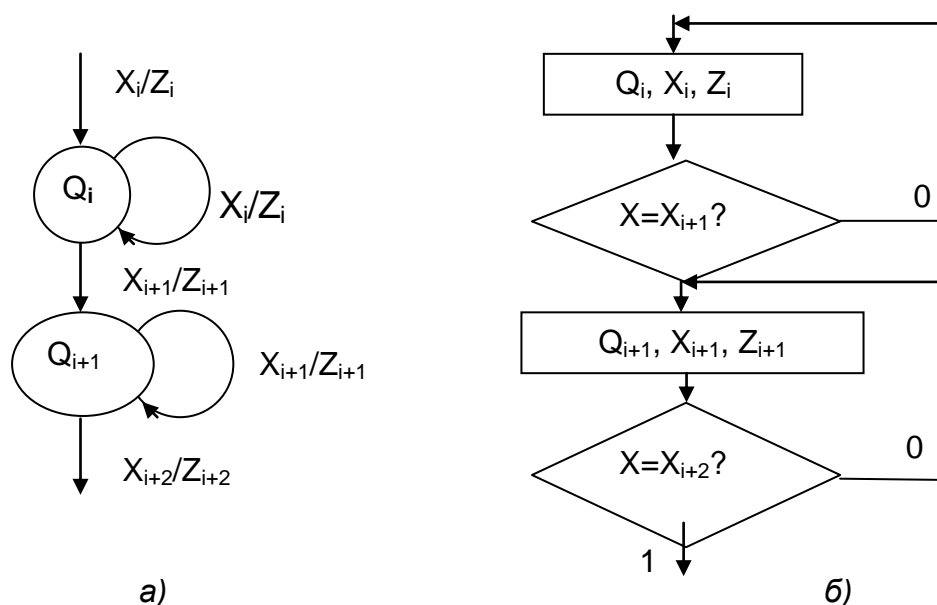


Рис. 4.10. Граф автомата (а) и схема алгоритма (б)



Так как любой оператор БСА может быть представлен последовательностью команд ПЛК, то дальнейшие действия по разработке прикладных программ для ПЛК-системы представляют собой трансляцию (отображение алгоритма на систему команд ПЛК), в результате которой формируется исходный текст программы (рис. 4.11).

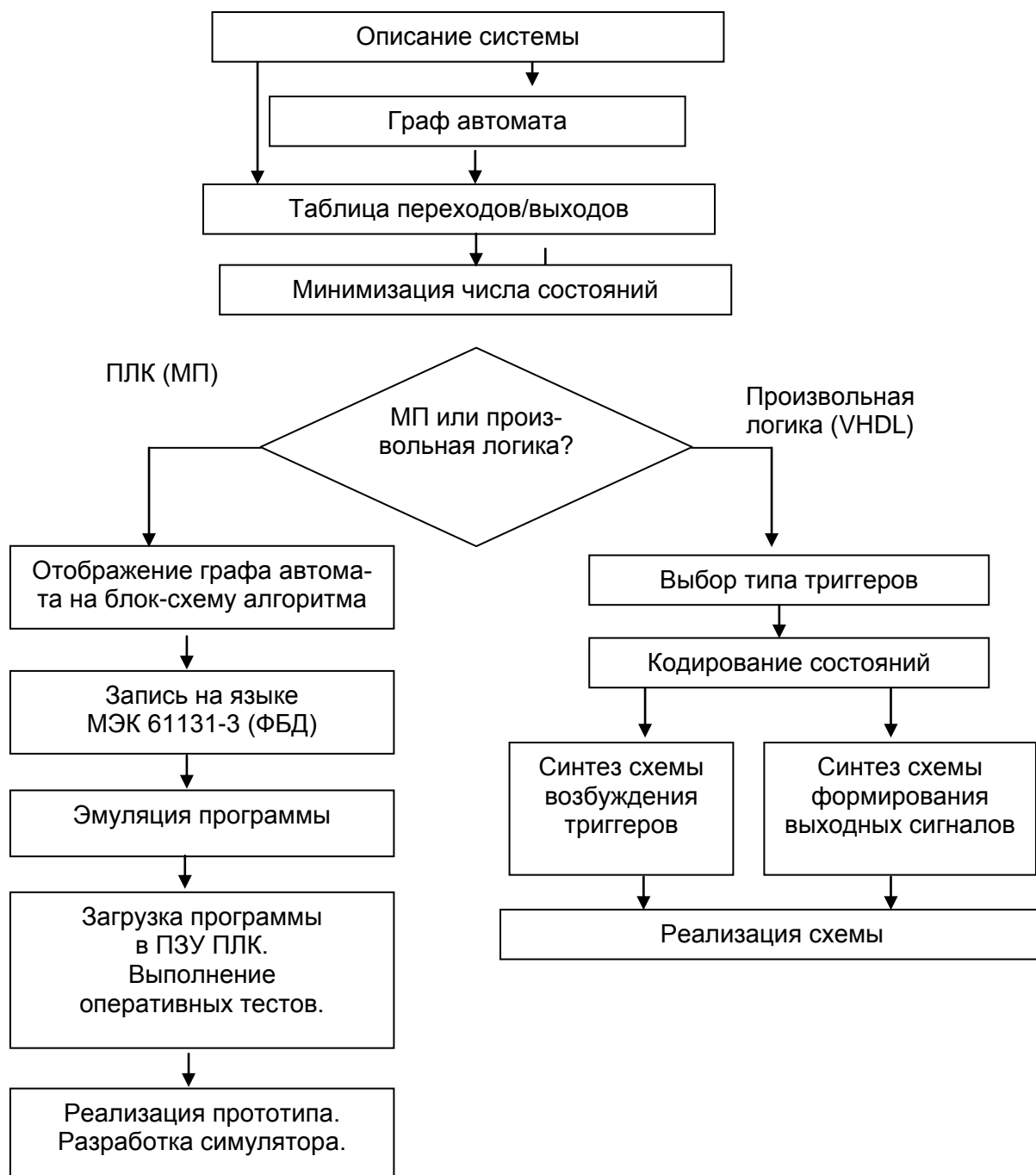
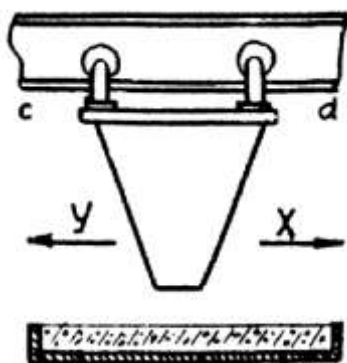


Рис. 4.11. Последовательность разработки МП-системы на базе ПЛК и программируемых логических матриц (ПЛМ)

### 4.2.2. Синтез многотактных схем на основе таблиц переходов и карт Карно

Рассмотрим один из формализованных методов синтеза многотактных схем. В качестве примера, на котором он может быть показан, рассмотрим следующую задачу. Требуется составить схему автоматического управления приводом раздаточного бункера бетонной смеси при его работе в режиме наполнения формы при формовании железобетонных изделий (рис. 4.12).



а)



б)

Рис. 4.12. Технологический процесс наполнения формы раздаточным бункером:  
а) функциональная схема системы раздаточного бункера;  
б) цех заливки бетоном длинномерных форм

Заполнение формы происходит в результате возвратно-поступательного движения бункера 1 между точками, где установлены конечные выключатели **c** и **d**, и ее перемещения в перпендикулярной плоскости схемы направлении. Введем и обозначим входные и выходные сигналы см. табл. 4.1.

Для упрощения процесса синтеза входные переменные **b** и **e** можно исключить и ввести их в полученное выражение в виде конъюнкции.

Таблица 4.1

Таблица сигналов

№	Название сигнала (Тип)	Обозначение	Адрес в ПЛК
<b>Входные сигналы (I):</b>			
1	Пуск	<b>a</b>	<b>I1</b>
2	Защита электропривода	<b>b</b>	<b>I2</b>
3	Бункер справа	<b>d</b>	<b>I3</b>
4	Бункер слева	<b>c</b>	<b>I4</b>
5	Стоп	<b>e</b>	<b>I5</b>
<b>Выходные сигналы (Q):</b>			
6	Движение бункера вправо	<b>X</b>	<b>Q1</b>
7	Движение бункера влево	<b>Y</b>	<b>Q2</b>

Синтезируемые функции  $X$  и  $Y$  будут последовательностными, а реализующие их схемы – многотактными, поскольку значение переменных  $c$  и  $d$  во время движения бункера не определяет его направления: при  $c=d=0$  бункер может двигаться и влево, и вправо (в зависимости от того, какие значения эти переменные имели в предыдущие моменты времени). Другими словами, значение выходных сигналов зависит не только от значения входных в данный момент времени, но и от того, какие значения они имели в предыдущие моменты времени.

На первом этапе решения задачи рассматриваемым методом составляется таблица переходов (табл. 4.2).

В левой части табл. 4.2 указываются все возможные состояния системы в процессе ее функционирования. В правой ее части – номер состояния, в которое перейдет система при том или другом наборе входных сигналов.

Если в клетке соответствующего состояния данный набор значений переменных не может иметь места, то в ней проставляется прочерк. В последнем столбце табл. 4.2 переходов проставляются значения выходных переменных, соответствующих данному состоянию.

Таблица 4.2

Таблица переходов

№ сост.	Состояние	acd								XY
		000	001	011	010	110	111	101	100	
1.	Бункер в крайнем правом положении	–	2	–	–	–	–	2	–	00
2.	Движение влево	2	–	–	3	3	–	–	2	01
3.	Бункер в крайнем левом положении	–	–	–	4	4	–	–	–	00
4.	Движение вправо	4	1	–	–	–	–	1	4	10
5.	Стоянка бункера в среднем положении	5	–	–	–	–	–	–	4	00

Соседние состояния должны отличаться значением хотя бы одной выходной переменной. Если это требование нарушается, то необходимо ввести промежуточную переменную, при помощи которой можно различить эти состояния.

Если в табл. 4.2 получаются совершенно одинаковые строки, это значит, что в нее введены лишние состояния. Покажем на данном примере, как заполняется таблица переходов.

В первую очередь следует проставить прочерки в столбцах табл. 4.2, соответствующих невозможным (запрещенным) наборам переменных. Таковыми являются наборы, где одновременно  $c$  и  $d$  равны 1 (так как бункер не может одновременно находиться в крайних левом и правом положениях). Дальнейшее заполнение табл. 4.2 производится исходя из следующих соображений.

Состояние 1. В столбцах, где  $d=0$ , следует проставить прочерк, так как в этом состоянии  $d=1$ . В столбцах, где  $d=1$ , независимо от значения  $a$  система перейдет в состояние 2.

Состояние 2. В столбцах, где  $d=1$ , следует проставить прочерк, так как в этом состоянии  $d=0$  и переход в состояние, при котором  $d=1$ , невозможен. При  $c=1$  система переходит в состояние 3 независимо от значения  $a$ . При  $c=0$  и  $d=0$  независимо от значения  $a$  сохраняется состояние 2.

Состояние 3. В столбцах, где  $c=0$ , проставляются прочерки; так как в этом состоянии  $c=1$ ; в столбцах, где  $d=1$ , также проставляются прочерки, так как здесь  $d=0$ . Только в столбцах, где  $c=1$ , проставляется состояние 4 (бункер движется вправо).

Состояние 4. Там, где  $c=1$ , проставляются прочерки, так как сигнал  $c$  такого значения в данном состоянии поступить не может. В столбцах, где  $c=0$  и  $d=0$ , проставляется состояние 4, а где  $d=1$  – состояние 1.

Состояние 5. В столбцах, где  $c$  или  $d$  равны 1, проставляются прочерки, так как в этом состоянии такие сигналы поступить не могут. При  $a=1$  система переходит в состояние 4 или 2. Для определенности проставим 4.

После составления таблицы переходов (см. табл. 4.2) выполняется операция ее сжатия (сокращения числа строк путем их объединения), что ведет к упрощению процесса синтеза, уменьшению количества промежуточных переменных и упрощению синтезируемой схемы.

Объединение строк таблицы переходов можно производить в следующих случаях:

а) когда строки содержат в каждом столбце одни и те же номера состояний, имеют на одних и тех же местах прочерки и одно и то же значение выходных переменных;

б) когда строки имеют различные номера состояний, но в одноименных столбцах содержатся или одинаковые номера состояний, или в одной строке номер состояния, а в другой прочерк. В объединенной строке в последнем случае проставляется номер состояния.

В данном случае таблица переходов может быть сжата за счет объединения строк 1, 3 и 5 и преобразовываться в табл. 4.3.

Следующий этап синтеза заключается в определении количества промежуточных переменных.

В общем случае минимальное количество промежуточных переменных определяется в соответствии с формулой

$$2^{P_{\min}} \geq S \geq 2^{P_{\min}-1},$$

где  $P_{\min}$  – минимальное количество промежуточных переменных;  $S$  – число состояний (количество строк в таблице переходов).

Таблица 4.3

Таблица переходов

№ сост.	acd								XY
	000	001	011	010	110	111	101	100	
1, 3, 5	5	2	–	4	4	–	2	4	00
2	2	–	–	3	3	–	–	2	01
4	4	1	–	–	–	–	1	4	10

В нашем случае  $S=3$ , следовательно,  $P_{\min} \geq 2$ . В качестве промежуточных сигналов могут быть использованы выходные сигналы. Это возможно в случаях, когда их значения во всех состояниях различны, как это, например, в табл. 4.3. Следовательно, в качестве промежуточных можно принять переменные  $X$  и  $Y$ .

На следующем этапе синтеза составляется карта соответствий, определяющая соответствие каждому состоянию определенного набора значений промежуточных переменных (рис. 4.13).

X \ Y	0	1
0	1,3,5	4
1	2	

Рис. 4.13. Карта соответствий

Количество клеток в карте –  $2^P$ , где  $P$  – общее количество промежуточных переменных (вместе с выходными, если они приняты в качестве промежуточных). В клетках карты соответствий проставляются номера состояний, соответствующие данному набору значений промежуточных переменных.

Наличие таблицы переходов и карты соответствий позволяет приступить к составлению карты Карно для выходных и промежуточных переменных. Количество клеток в такой карте будет равно  $2^{P+n}$ , где  $n$  – количество входных переменных. В нашем случае оно будет равно  $2^{2+3}=32$  (рис. 4.15).

При составлении карты Карно, переменные следует располагать так, как это показано на рис. 4.14. Тогда строки карты Карно будут соответствовать состояниям по карте соответствий (см. рис. 4.13) (строка  $S_1$  соответствует состояниям 1, 3, 5; строка  $S_2$  – состоянию 2, строка  $S_3$  – состоянию 4).

acd \ XY	000	001	011	010	110	111	101	100
00	00	01	–	10	10	–	01	10
01	01	–	–	00	00	0	0	01
11	–	–	–	–	–	–	–	–
10	10	00	–	–	–	–	00	10

Рис. 4.14. Карта Карно

Клетки карты Карно заполняются по данным таблицы переходов (см. табл. 4.3) и карты соответствий (см. рис. 4.13) следующим образом. По таблице переходов определяются номера состояний, соответствующие данным наборам входных переменных и заполняемой строке карты Карно, а по карте соответствий – значения выходных и промежуточных переменных.

Например, в таблице переходов (табл. 4.3) в первой строке в столбце 000 имеем состояние 5, чему соответствует набор выходных (в данном случае и промежуточных) переменных **XY** – **00** по карте соответствий. Этот набор и проставляется в соответствующей клетке карты Карно. Аналогично:

строка **S<sub>1</sub>**, столбец **001**, состояние 2, набор **XY** – 01;

"**010**", "4" – 10;

"**110**", "4" – 10;

"**101**", "2" – 01 и т.д.

Если в таблице переходов в данной строке при данном наборе входных переменных стоит прочерк, то и в соответствующей клетке карты Карно проставляется прочерк.

По карте Карно для каждой выходной и промежуточной переменной могут быть записаны структурные формулы и построены соответствующие схемы. Для этого ее следует расчленить на отдельные карты для каждой из выходных и промежуточных переменных. В рассматриваемом случае карту Карно (рис. 4.14) можно расчленить на две карты: для **X** (рис. 4.15а) и для **Y** (рис. 4.15б).

Структурные формулы в ДНФ для **X** и **Y**, записанные по этим картам, будут иметь вид:

$$X = \bar{Y}c \vee X\bar{d} \vee \bar{Y}a\bar{d},$$

$$Y = \bar{X}d \vee Y\bar{c}.$$

Эти формулы следует дополнить переменными **b** и **e** (защита и «Стоп»), опущенными на время синтеза. В результате такого дополнения получим структурные формулы для выходных переменных в окончательном виде:

$$X = (\bar{Y}c \vee X\bar{d} \vee \bar{Y}a\bar{d})\bar{b}\bar{e},$$

$$Y = (\bar{X}d \vee Y\bar{c})\bar{b}\bar{e}.$$

XY \ acd	000	001	011	010	110	111	101	100
00	0	0	–	1	1	–	0	1
01	0	–	–	0	0		0	0
11	–	–	–	–	–	–	–	–
10	1	0	–	–	–	–	0	1

а)

XY \ acd	000	001	011	010	110	111	101	100
00	0	1	–	0	0	–	1	0
01	1	–	–	0	0	–	–	1
11	–	–	–	–	–	–	–	–
10	0	0	–	–	–	–	0	0

б)

Рис. 4.15. Карта Карно: а – для X; б – для Y

### Вопросы для самоконтроля

1. Перечислите этапы, которые необходимо выполнить для синтеза программы?
2. Какими схемами и техническим данными пользуется программист для разработки программ для ПЛК?
3. В чём заключается назначение БСА?
4. Что такое метод декомпозиции при создании программ?
5. В чём заключается структурное программирование?
6. Какова очерёдность шагов при разработке БСА?

### 4.3. Эмуляция функционирования программы

После создания программы необходимо убедиться в том, что коммутационная программа будет работать в соответствии с алгоритмом. Переведите коммутационную программу в режим эмуляции (off-Line) командой F3 меню «Сервис -> Эмуляция» или значком эмуляции на панели инструментов программирования.

Убедитесь в том, что в программе нет ошибок. В начале эмуляции программа LSC проверяет коммутационную программу и отображает все имеющиеся в ней ошибки в информационном окне, которое вызывается при помощи команды меню «Вид -> Информационное» окно или нажатием функциональной клавиши F4. Вы можете также воспользоваться функциональной клавишей F2 в информационном окне для отображения модулей LOGO! способных выполнить вашу программу.



В режиме эмуляции имеется панель инструментов (рис. 4.16) эмуляции и окно состояния, позволяющая выполнить эмуляцию, а также наблюдение и управление ходом выполнения коммутационной программы.

В процессе отладки можно изменить некоторые параметры. Можно попробовать различные входные значения, испытать реакцию системы на аварию питания, а также сравнить свои расчеты и ожидания с эмулируемым откликом выходов. Панель инструментов эмуляции и окно состояния предоставляют в распоряжение необходимые средства для выполнения этих задач. Поплавковые датчики уровня и реле давления действуют без фиксации. Тем не менее, для эмуляции схемы в целях ее испытания можно изменить входную функцию блока с действия без фиксации на переключение.

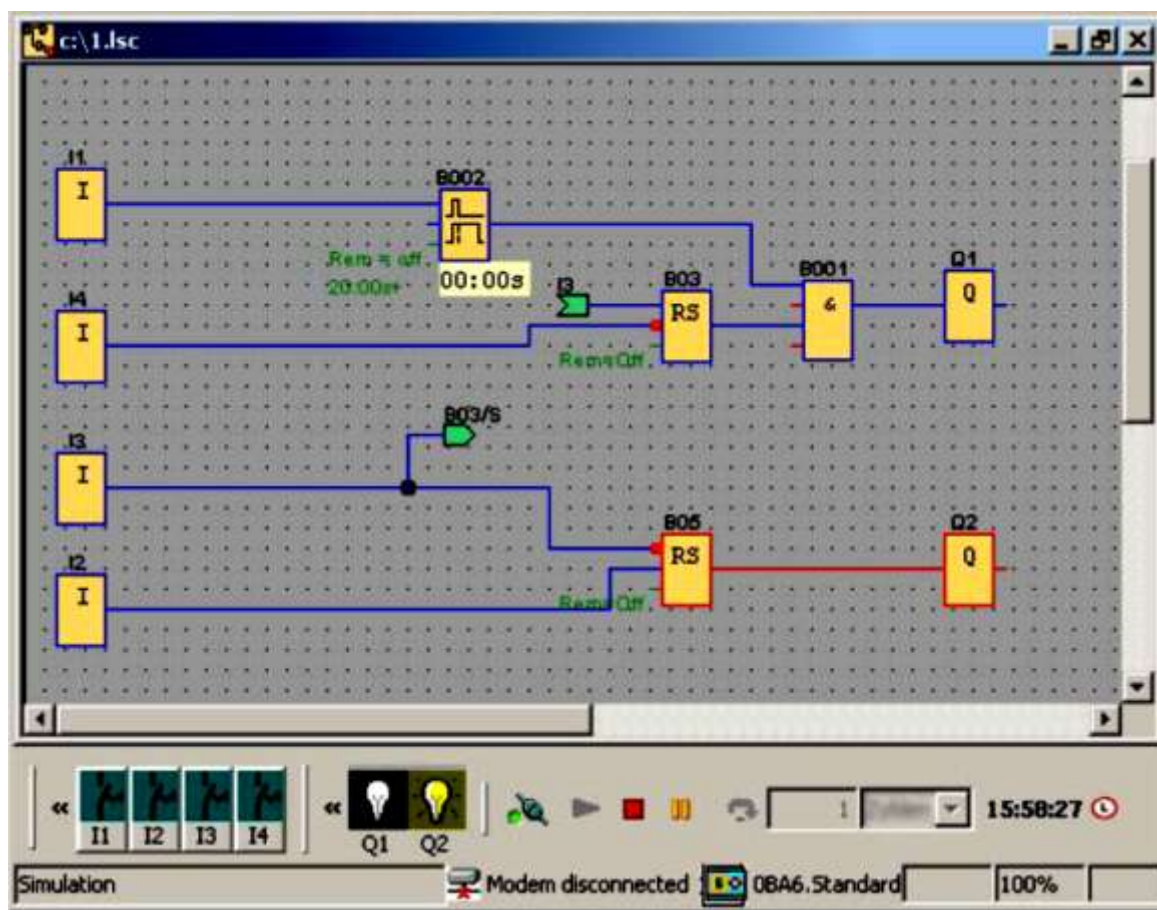


Рис. 4.16. Пример теста для редактора языка ФБД

#### 4.3.1. Авария питания

Пользователь может эмулировать аварию питания нажатием на значок «Питание» для прерывания подачи питания ко всем входам.

Эта функция может использоваться для проверки реакции схемы на аварию питания и перезапуск, а также ее свойства сохранения.

### 4.3.2. Выполнение оперативных тестов программ

После того, как эмуляция выполнения программы покажет её соответствие реализуемому алгоритму, её загружают в ПЛК и выполняют оперативное тестирование коммутационной программы.

Оперативное тестирование сходно с эмуляцией в том, что можно просматривать значения входов и выходов и параметры блоков. Оно отличается тем, что тестируется программа, работающая в устройстве LOGO! с «реальными» (on-Line) входными значениями, а не на ПК с моделируемыми входными воздействиями. Пример на рис. 4.17 показывает on-Line тестирование практического примера коммутационной программы.

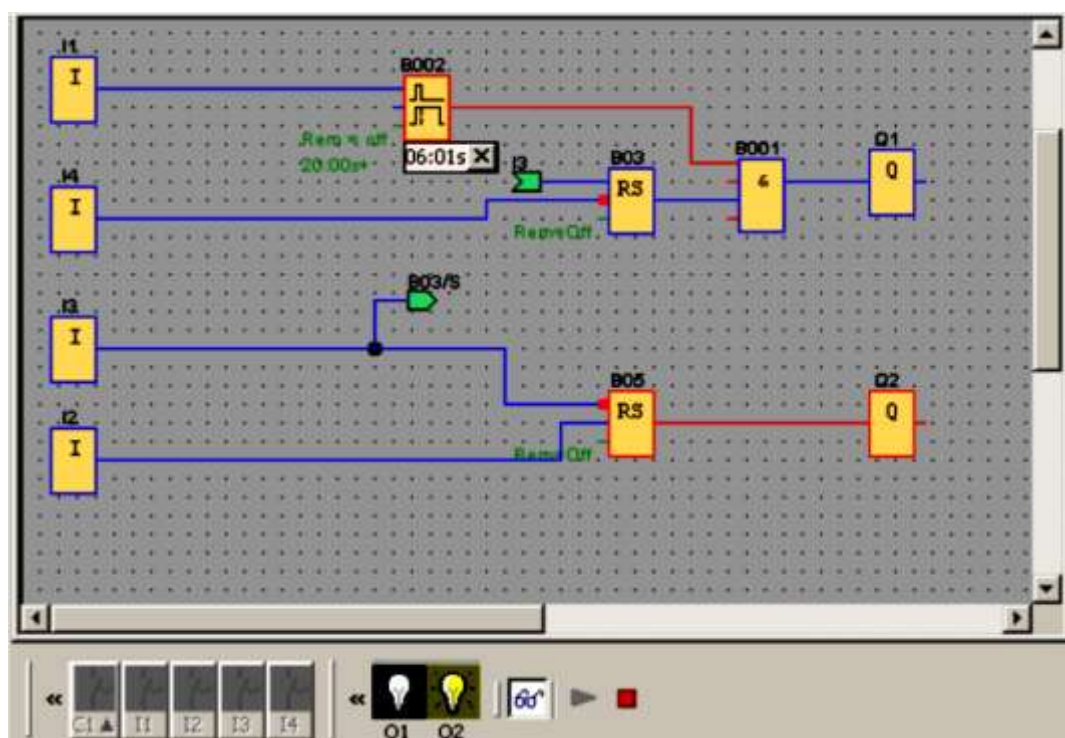


Рис. 4.17. Тест программы FBD в режиме on-Line

В этом примере вход I1 был сначала включен, а затем выключен. Таймер задержки времени начал отсчет при выключенном I1.

### 4.3.3. Документирование программы

Хороший стиль программирования требует размещать комментарии в программе, также входов и выходов для того, чтобы обеспечить легкое понимание программы.

В принципе не обязательно отображать названия соединителей на экране. Тем не менее, необходимо задать названия соединителей (входов), поскольку впоследствии может потребоваться распечатка перечня соединений. Из вкладки комментариев или диалога «Файл ->

Свойства» можно добавить комментарий к коммутационной программе (см. рис. 4.18).



Рис. 4.18. Комментарии в программе (Rainwaterpump.lsc)

### Вопросы для самоконтроля

1. Какие возможности предоставляет инструмент эмуляции?
2. Для чего нужен тест эмуляции аварии питания? Какие требования выдвигаются при этом к характеристикам программы?
3. Для чего необходимо задавать названия соединителей?

## Глава 5. ЯЗЫК ФУНКЦИОНАЛЬНЫХ БЛОКОВЫХ ДИАГРАММ

### 5.1. Постоянные и соединители. ФБД

#### 5.1.1. Входы

Входные блоки представляются в виде входных клемм LOGO в распоряжении до 24 цифровых входов (I), показаны в табл. 5.1.

Таблица 5.1

Ячейки памяти и клемм ввода-вывода

Отображение в редакторе функциональных блок-схем		Отображение в редакторе релейно-контактных схем	
	Входы		Замыкающий контакт
	Клавиши управления курсором		
	Функциональные клавиши модуля LOGO! TD		Размыкающий контакт
	Биты регистра сдвига		
	Постоянные логические уровни		Инвертированный выход
1 	Выходы		Катушка реле
1 	Открытые соединительные элементы		
1 	Флаги		Внутренний флаг
	Аналоговые входы		Аналоговый контакт
1 	Аналоговые выходы		
	Аналоговые флаги		

Число доступных значков зависит от выбранной версии LOGO! (см. прил. 2).

#### 5.1.2. Клавиши управления курсором

Имеется до четырех клавиш управления курсором представлены в табл. 5.1. Клавиши управления курсором программируются для коммутационной программы, как и остальные входы. Клавиши управления курсором позволяют экономить выключатели и входы и дают опе-

ратору возможность ручного управления работой коммутационной программы.

### **5.1.3. Выходы**

Выходные блоки представляются в виде выходных клемм LOGO!. Вы можете использовать до 16 выходов. В своей конфигурации блоков вы можете назначить выходному блоку новую клемму, при условии, что эта клемма еще не используется коммутационной программой. Выход всегда сохраняет сигнал предыдущего цикла программы. Это значение не изменяется в течение текущего цикла программы.

### **5.1.4. Постоянные логические уровни**

Устанавливает вход блока в логическое состояние **hi** ( $hi$  = высокий уровень) для постоянной установки его в состояние «1» или «Н».

Устанавливает вход блока в логическое состояние **lo** ( $lo$  = низкий уровень) для постоянной установки его в состояние «0» или «1».

### **5.1.5. Биты регистра сдвига**

Модуль LOGO! предоставляет биты регистра сдвига S1 – S8, которым в коммутационной программе присваивается атрибут «только для чтения» (см. табл. 5.1). Содержимое битов регистра сдвига может быть изменено при помощи специальной функции «Регистр сдвига».

### **5.1.6. Флаги**

Флаги являются промежуточными функциональными элементами памяти. Например, флаги соединяют выход неиспользуемого блока (например, тексты сообщения) с блоком «открытого соединителя» (см. табл. 5.1).

Блоки флагов выдают на выход свои входные сигналы. LOGO! использует 27 цифровых флагов M1...M27 и 6 аналоговых флагов AM1...AM6.

**0BA4, 0BA5:** 24 цифровых флагов M1...M24; 6 аналоговых флагов AM1...AM6

**0BA3, 0BA2:** 8 цифровых флагов M1...M8

**0BA1:** 4 цифровых флагов M1...M4

**0BA0:** 0 флагов

В своей конфигурации блоков можно назначить флагу новый номер, при условии, что этот номер флага еще не существует в коммутационной программе.

Выход всегда сохраняет сигнал предыдущего цикла программы. Это значение не изменяется в течение текущего цикла программы.

### **Флаг запуска: M8**

Флаг M8 устанавливается в первом цикле пользовательской программы и может использоваться коммутационной программой как флаг запуска. Он сбрасывается после первого цикла выполнения программы. В последующих циклах флаг M8 действует точно также, как флаги M1 – M7.

### **Флаги задней подсветки: M25 и M26**

Флаг M25 управляет подсветкой дисплея модуля LOGO! Флаг M26 управляет подсветкой дисплея модуля LOGO! TD (текстовый дисплей).

### **Флаг установки символа текстового сообщения: M27**

Флаг M27, в случае его применения, определяет отображение текстов сообщений с помощью основного или вспомогательного набора символов.

### **Аналоговые флаги: AM1 – AM6**

Аналоговые флаги используются в качестве маркеров для аналоговых входов или блоков аналоговых команд. Аналоговый флаг просто принимает аналоговое входное значение и осуществляет вывод этого значения.

#### ***5.1.7. Аналоговые входы***

Версии LOGO! 12/24 RC, 12/24 RCo, 24 и 24o, а также модули расширения AM212/24 обрабатывают аналоговые сигналы, и имеют до восьми аналоговых входов (табл. 5.1).

Некоторые из входных клемм модулей LOGO! 0BA6 версий 12/24 RC, 12/24 RCo, 24и 24o имеют двойное назначение: им могут назначаться как цифровые входы, так и аналоговые. См. информацию о конкретных модулях в руководстве LOGO!. Коммутационные программы, написанные для этих модулей выпуск 0BA5 и более ранних, могут использоваться для модулей 0BA6 без изменений. Новые коммутационные программы могут использовать новые входные функции.

#### ***5.1.8. Аналоговые выходы***

Имеются два аналоговых выхода: AQ1 и AQ2. Для аналогового выхода может задать только аналоговое значение, то есть функцию с аналоговым выходным значением или аналоговый флаг AM.

В случае подключения к специальной функции (имеющей аналоговый выход) к реальному аналоговому выходу обратите внимание, на то, что аналоговый выход может обрабатывать только значения от 0



до 1000. Начиная с серии устройств 0BA5 можно настраивать работу аналоговых выходов в режиме STOP. При переходе модуля LOGO! в режим STOP на аналоговых выходах могут сохраняться последние значения. Или же вы можете задать установку конкретных значений для AQ1 и AQ2 при переходе LOGO! в режим STOP.

Имеется два варианта диапазона значений для аналогового выхода:

- Нормальный (0 – 10 В или 0 – 20 мА);
- 4-20 мА.

### Вопросы для самоконтроля

1. Чем представляются входные клеммы программируемого контроллера в программе?
2. Для чего предназначены клавиши ручного управления?
3. Каковы свойства выходных сигналов контроллера во время выполнения цикла программы?
4. Для чего предназначены постоянные логические уровни?
5. Что такое флаги? Какие функции они выполняют? Перечислите
6. От чего зависит число доступных блоков для программирования?
7. Какие диапазоны значений для аналоговых сигналов используются в LOGO?

### 5.2. Базовые функции языка ФБД (FBD)

Конкретный тип блока базовой функций (см. рис. 5.1) выбирается из группы на дополнительной панели инструментов, раскрывающейся при выборе инструмента базовые функции.



Рис. 5.1. Базовые логические функции

### Инверсия входов

- Логическая «1» на конкретном входе инвертируется в логический «0» в коммутационной программе.



- Логический «0» инвертируется в логическую «1» в коммутационной программе.

Для этого правой кнопкой мыши на входе и выбирается команда инвертировать из меню быстрого выбора. Нельзя инвертировать входы выходных блоков.

### Вопросы для самоконтроля

1. Какие базовые функции, языка ФБД Вы знаете?
2. Приведите пример реализации логических функций на языке ФБД?

### 5.3. Специальные функции языка ФБД (FBD)

Конкретный тип блока специальной функции выбирается на дополнительной панели инструментов, раскрывающейся при выборе инструмента специальные функциональные блоки (**SF**) (см. рис. 5.2).

Описание задач или характеристик специальных функций:

- инвертирующие входы;
- конфигурация блока;
- опорные параметры (Reference parameters);
- защита (Protection);
- сохранение.

#### Инвертирующие входы

Инвертирование отдельных входов:

- Логическая «1» на конкретном входе инвертируется в логический «0» в коммутационной программе.
- Логический «0» инвертируется в логическую «1» в коммутационной программе.

#### Опорные функции

Можно задавать параметры для блоков при помощи других блоков. Такие параметры называются опорными параметрами. При нажатии кнопки «Опорный» около параметра в окне свойств блока можно выбрать другой блок, задающий фактическое значение для данного параметра. Этот способ позволяет, например, назначить время для таймера задержки выключения равным аналоговому выходному значению другого блока.

#### Защита

При наличии блока выбора «Защита активна» для защиты параметра блока можно разрешить или заблокировать отображение и редактирование этого параметра в режиме конфигурирования устройства LOGO!.

Таймеры		Аналоговые	
	Задержка включения		Аналоговый пороговый выключатель
	Задержка отключения		Аналоговый дифференциальный выключатель
	Задержка включения и отключения		Аналоговый компаратор
	Задержка включения с сохранением		Контроль аналоговых значений
	Интервальное реле (импульсный выход)		Аналоговый усилитель
	Интервальное реле с запуском по фронту		Аналоговый мультиплексор
	Генератор симметричных импульсов		Блок широтно-импульсного модулятора (ШИМ)
	Асинхронный генератор импульсов		Блок аналоговых вычислений
	Генератор случайных импульсов	<b>Управление и регулирование</b>	
	Выключатель лестничного освещения		
	Многофункциональный выключатель		ПИ-регулятор
	Семидневный таймер		Линейно нарастающее регулирование
	Годовой таймер	<b>Прочие</b>	
<b>Счетчики</b>			Реле с блокировкой
			Импульсное реле
			Текст сообщения
	Реверсивный счетчик		Программный выключатель
	Счетчик рабочего времени		Регистр сдвига
	Пороговый выключатель		Обнаружение ошибок аналоговых вычислений

Рис. 5.2. Специальные функции

### Функция сохранения

Состояние коммутации и значения в счетчиках специальных функциональных блоков могут обладать функцией сохранения. Это означает, что текущие значения данных сохраняются, например, после аварии электропитания, и выполнение этой функции возобновля-

ется с прерванного места после включения питания. Следовательно, таймер не сбрасывается, а вместо этого истекает записанное в него время. Для того чтобы разрешить применение этого свойства для конкретной функции должна быть задана функция сохранения. Имеется две возможные установки:

- вкл.: «Сохраняются текущие значения данных»;
- откл.: «Текущие значения данных не сохраняются» (по умолчанию).

Счетчик рабочего времени является исключением, поскольку он в целом обладает свойством сохранения.

#### 5.3.1.1. Задержка включения

Выход не включается до истечения заданного интервала  $T$  (см. табл. 5.2) задержки (см. рис. 5.3).

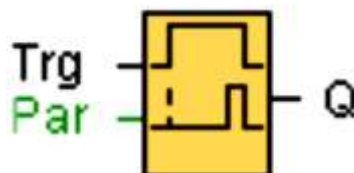


Рис. 5.3. Блок функции задержки включения

Таблица 5.2

Параметры функции задержки включения

Соединение	Описание
Вход Trg (выключатель)	Время задержки включения управляется при помощи входа Trg (выключатель).
Параметр $T$	$T$ представляет собой время задержки включения, по истечении которого выход включается (происходит переход сигнала из 0 в 1). <b>Сохранение</b> вкл. = состояние сохраняется в памяти.
Выход $Q$	$Q$ включается по истечении заданного интервала времени $T$ при условии, что выключатель Trg по-прежнему взведен.

#### Параметр $T$

Время для параметра  $T$  может быть получено как значение другой ранее запрограммированной функции:

- аналоговый компаратор:  $Ax-Ay$ ;
- аналоговый выключатель:  $Ax$ ;
- аналоговый усилитель:  $Ax$ ;
- аналоговый мультиплексор:  $AQ$ ;
- линейно нарастающий аналоговый сигнал:  $AQ$ ;
- блок аналоговых вычислений:  $AQ$ ;

- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

Информация о достоверности и точности временного интервала рассматривается в руководстве пользователя [3].

### Описание функции

Время  $T_a$  (текущее время в системе LOGO!) запускается переходом из 0 в 1 входа Trg (см. рис. 5.4).

Если состояние входа Trg остается равным 1 по крайней мере в течение заданного в конфигурации времени  $T$ , выход устанавливается равным 1 по истечении этого времени (сигнал выхода следует за сигналом входа с задержкой).

Время сбрасывается, если состояние входа Trg снова изменяется на 0 до того, как истечет время  $T$ .

Выходной сигнал сбрасывается в 0, когда вход Trg равен 0.

#### 5.3.1.2. Задержка отключения

Выход с задержкой выключения не сбрасывает до истечения заданного времени (см. рис. 5.4, табл. 5.3).

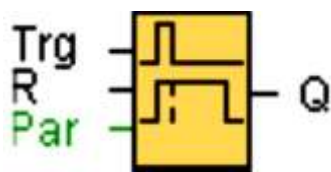


Рис. 5.4. Блок функции задержки отключения

### Параметр T

Время, заданное параметром  $T$ , может быть получено как значение другой ранее запрограммированной функции:

- аналоговый компаратор: Ax-Ay;
- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

### Описание функции

Выход  $Q$  кратковременно устанавливается в 1 при переходе из 0 в 1 на входе Trg (рис. 5.5). При переходе из 1 в 0 вход Trg, LOGO! повторно взводит текущее время  $T$  и выход остается в установленном

состоянии. Выход Q сбрасывается в 0, когда значение  $T_a$  достигает величины, заданной в Т ( $T_a=T$ ) (задержка выключения).

Однократное срабатывание на входе Trg повторно взводит время  $T_a$ . Можно сбросить время  $T_a$  и выход через вход R (сброс) до истечения времени  $T_a$ .

Таблица 5.3

Параметры функции задержки отключения

Соединение	Описание
<b>Вход Trg</b>	Начинает отсчет времени задержки по отрицательному фронту (переход из 1 в 0) входа Trg (выключатель).
<b>Вход R</b>	Сбрасывает время задержки выключения и устанавливает на выходе 0 при помощи входа R (сброс). Сброс обладает приоритетом над входом Trg.
<b>Параметр</b>	Т: Выход выключается по истечении времени ожидания Т (выходной сигнал переходит из 1 в 0). <b>Сохранение</b> вкл. = состояние сохраняется в памяти.
<b>Выход Q</b>	Q включается на интервал времени Т после срабатывания входа Trg.

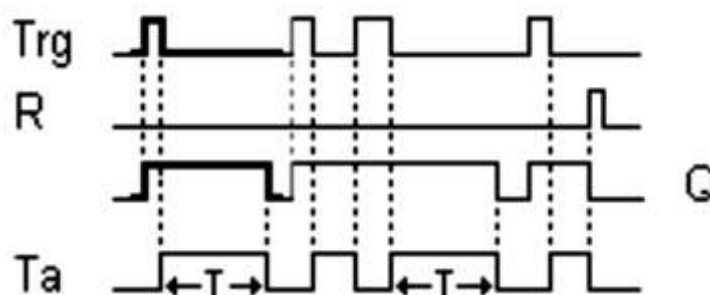


Рис. 5.5. Временная диаграмма задержки отключения

### Пример программы задержки выключения

Электрическая схема управления приведена на рис. 5.6. Нагрузка включается ключом S1 или ключом S2. Это параллельное соединение соответствует **логическому 'ИЛИ'**, так как ключ S1 или ключ S2 включает выход K1. При размыкании S1 и S2 нагрузка выключается через заданное время задержки.

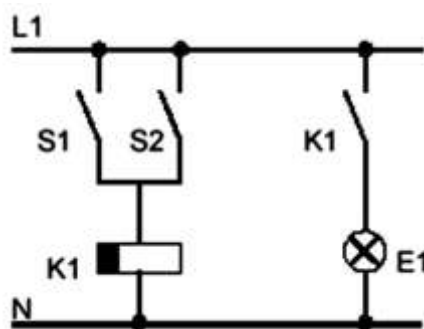


Рис. 5.6. Электрическая схема

## Электрическое подключение

Электрическое включение модуля LOGO! показано на рис. 5.7.

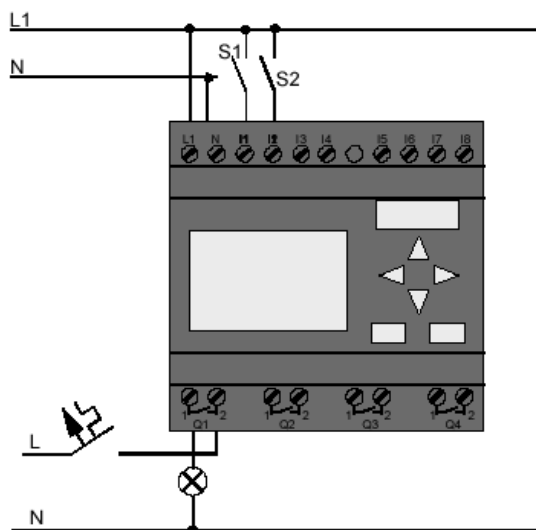


Рис. 5.7. Электрическое подключение модуля

## Программа на языке ФБД

Электрическая схема (см. рис. 5.6), будучи преобразованной в программу ФБД, имеет вид (рис. 5.8): реле K1 (LOGO! через выход Q1) управляется блоком «ИЛИ». Ключи S1 и S2 управляют реле. Это реле K1 (с задержкой по времени  $t$ ) включает нагрузку E1. Оно должно выключить потребителя по истечении  $t=10$  секунд (рис. 5.9).

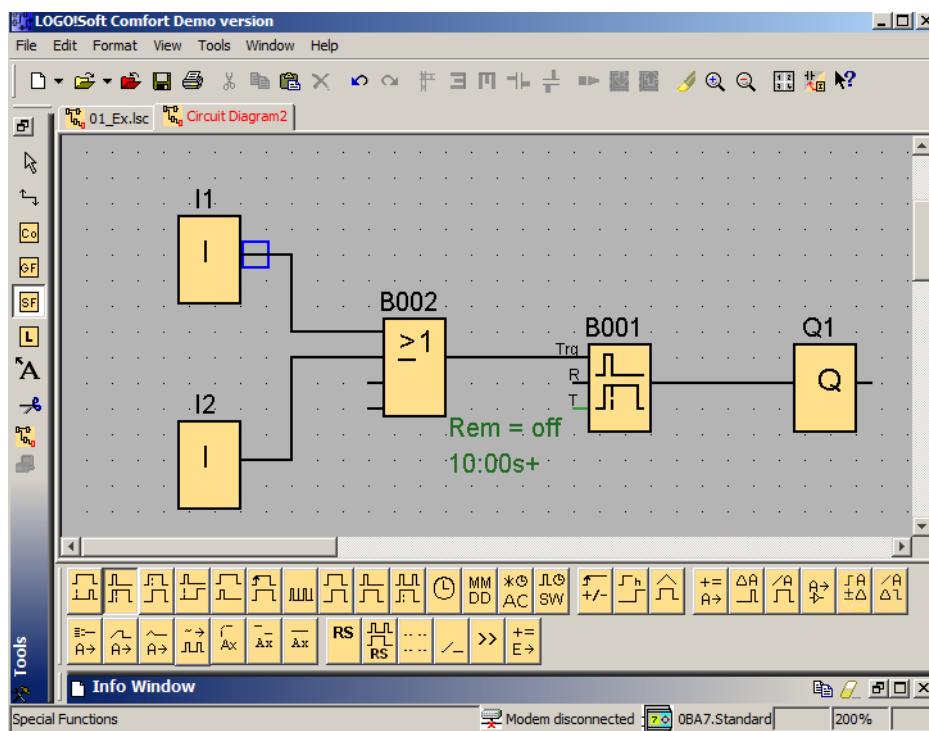


Рис. 5.8. Коммутационная программа на языке ФБД

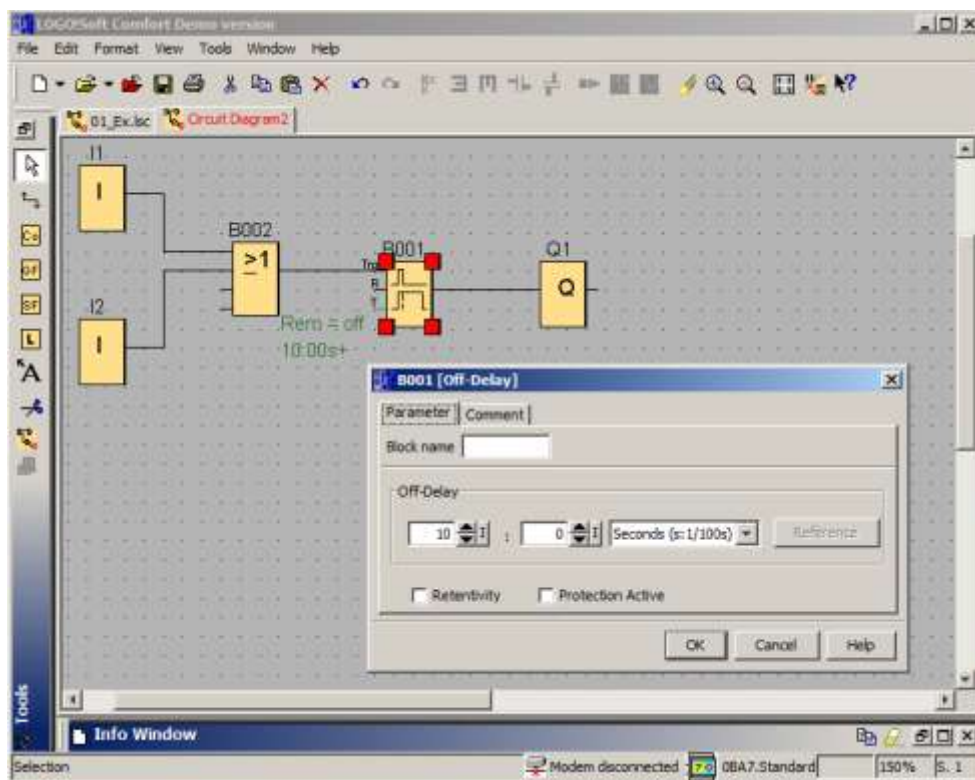


Рис. 5.9. Настройка функции задержки отключения на 10 сек

### 5.3.1.3. Задержка включения и отключения

Блок функции задержки включения и выключения (см. рис. 5.10, табл. 5.4) используется для установки выхода после истечения заданного времени задержки включения и его сброса после завершения второго запрограммированного значения времени.

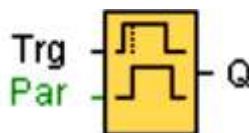


Рис. 5.10. Блок функции задержки включения и отключения

### Параметры $T_H$ и $T_L$

Время задержки включения и время задержки выключения, установленные параметрами  $T_H$  и  $T_L$ , могут быть получены как значение другой ранее запрограммированной функции:

- аналоговый компаратор:  $A_x - A_y$ ;
- аналоговый выключатель:  $A_x$ ;
- аналоговый усилитель:  $A_x$ ;
- аналоговый мультимплексор:  $AQ$ ;
- линейно нарастающий аналоговый сигнал:  $AQ$ ;
- блок аналоговых вычислений:  $AQ$ ;



- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

Таблица 5.4

### Параметры функции задержки включения и отключения

Соединение	Описание
Вход Trg	Запускает отсчет времени задержки включения по положительному фронту (переход из 0 в 1) входа Trg (выключатель). Вы запускаете отсчет времени задержки выключения по отрицательному фронту (переход из 1 в 0).
Параметр	$T_H$ представляет собой время задержки включения выхода (переход выходного сигнала из 0 в 1). $T_L$ представляет собой время задержки выключения выхода (переход выходного сигнала из 1 в 0). Сохранение вкл. = состояние сохраняется в памяти.
Выход Q	Q включается по истечении заданного времени $T_H$ , если Trg по-прежнему установлен. Он снова выключается по истечении времени $T_L$ , если Trg не был снова установлен.

### Описание функции

Отсчет времени  $T_H$  запускается при изменении сигнала на входе (см. рис. 5.11) Trg с 0 на 1. Если состояние входа Trg остается равным 1 по крайней мере в течение заданного в конфигурации времени  $T_H$ , выход устанавливается равным логической 1 по истечении этого времени (сигнал выхода устанавливается с задержкой по отношению ко входному сигналу). Время  $T_H$  сбрасывается, если состояние входа Trg сбрасывается 0 до того, как истечет это время. Отсчет времени  $T_L$  запускается при изменении сигнала на выходе с 1 на 0. Если состояние входа Trg остается равным 0 по крайней мере в течение заданного в конфигурации времени  $T_L$ , выход сбрасывается в 0 по истечении этого времени (сигнал выхода сбрасывается с задержкой по отношению ко входному сигналу). Время  $T_L$  сбрасывается, если состояние входа Trg возвращается в 1 до того, как истечет это время.

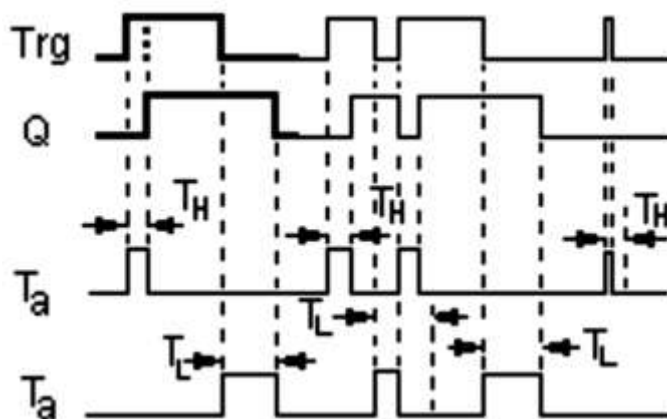


Рис. 5.11. Временная диаграмма задержки включения-отключения

#### 5.3.1.4. Задержка включения с сохранением

Блок функции задержки с сохранением (рис. 5.12). Однократное срабатывание на входе запускает отсчет конфигурируемого интервала времени. Установка выхода осуществляется по истечении этого времени (табл. 5.5).

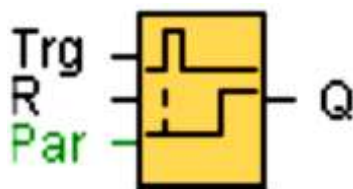


Рис. 5.12. Блок функции задержки включения с сохранением

Таблица 5.5

Параметры функции задержки включения с сохранением

Соединение	Описание
Вход <b>Trg</b>	Запуск времени задержки включения Trg (выключатель).
Вход <b>R</b>	Сброс времени задержки включения и сброс выхода в 0 при помощи входа R (сброс).
Параметр	<b>T</b> время задержки включения выхода (переход выходного сигнала из 0 в 1) <b>Сохранение</b> вкл. = состояние сохраняется в памяти.
Выход <b>Q</b>	Q переключается по истечении времени T

#### Параметр T

Время для параметра T может быть получено как значение другой ранее запрограммированной функции:

- аналоговый компаратор: Ax – Ay;
- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

#### Описание функции

Отсчет времени Ta (см. рис. 5.13) запускается при изменении сигнала на входе Trg с 0 на 1. Выход Q устанавливается в 1, когда Ta достигает значения времени T. Следующий импульс на входе Trg не влияет на значение Ta. Выход и время Ta сбрасываются в 0 только при наличии сигнала 1 на входе R. Если не включено сохранение, то выход Q и истекшее время сбрасываются после аварии питания.

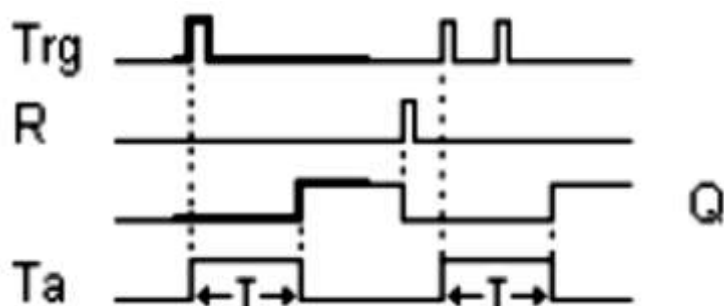


Рис. 5.13. Временная диаграмма задержки включения с сохранением

#### 5.3.1.5. Интервальное реле (импульсный выход)

Входной сигнал (рис. 5.14, см. табл. 5.6) генерирует выходной сигнал конфигурируемой длины (рис. 5.15).

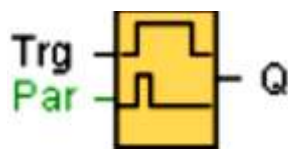


Рис. 5.14. Блок функции интервального реле (импульсный выход)

#### Параметр Т

Время выключения параметра Т может быть получено как значение другой ранее запрограммированной функции:

- аналоговый компаратор: Ax – Ay;
- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

Таблица 5.6

Параметры функции интервального реле

Соединение	Описание
Вход <b>Trg</b>	Переключение интервального реле сигналом на входе Trg (выключатель).
Параметр	<b>Т</b> представляет собой время, по истечении которого выход сбрасывается (переход выходного сигнала из 1 в 0). <b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.
Выход <b>Q</b>	Импульс на входе Trg устанавливает Q. Выход продолжает быть установленным до истечения времени Т, и если Trg = 1 в течение этого времени. Переход из 1 в 0 на входе Trg до истечения Т также сбрасывает выход в 0.

### Описание функции

При входном сигнале  $Trg=1$  выход  $Q$  устанавливается в 1 (рис. 5.15). Этот сигнал также запускает отсчет времени  $Ta$ , в течение которого выход остается установленным. Когда  $Ta$  достигает значения, определенного в  $T$  ( $Ta=T$ ), то выход  $Q$  сбрасывается в состояние 0 (импульсный выход). Если сигнал на входе  $Trg$  изменяется с 1 на 0 до истечения времени, выход немедленно сбрасывается с 1 в 0.

### Временная диаграмма

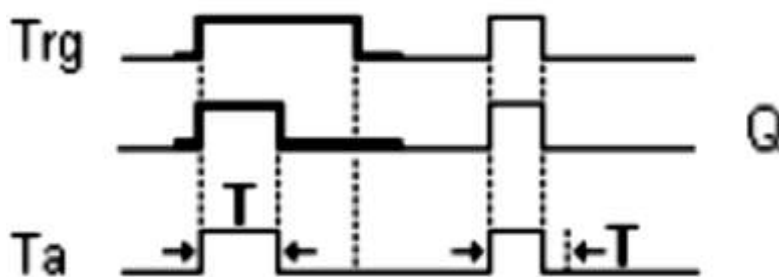


Рис. 5.15. Временная диаграмма интервального реле

#### 5.3.1.6. Интервальное реле с запуском по фронту

При подаче импульса на вход по истечении заданного времени задержки генерируется заданное число импульсов на выходе с определенным соотношением импульса и паузы (повторный запуск) (рис. 5.16).

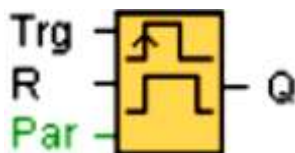


Рис. 5.16. Блок функции интервального реле с запуском по фронту

**0BA2, 0BA3:** Существует только параметр  $TN$  (табл. 5.17),  $TN$  который представляет собой время задержки отключения выхода. Вход  $R$  недоступен. Параметры  $TN$  и  $TL$ . Ширина импульса  $TN$  и промежуток между импульсами  $TL$  могут быть получены как значение другой ранее запрограммированной, функции:

- аналоговый компаратор:  $Ax - Ay$ ;
- аналоговый выключатель:  $Ax$ ;
- аналоговый усилитель:  $Ax$ ;
- аналоговый мультиплексор:  $AQ$ ;
- линейно нарастающий аналоговый сигнал:  $AQ$ ;
- блок аналоговых вычислений:  $AQ$ ;

- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

Таблица 5.7

Параметры функции интервального реле с запуском по фронту

Соединение	Описание
Вход <b>Trg</b>	Переключение значения времени интервального реле с запуском по фронту сигналом на входе Trg (выключатель).
Вход <b>R</b>	Выход и текущее время $T_a$ сбрасываются в 0 сигналом на входе R.
Параметр	<b><math>T_H</math>, <math>T_L</math></b> : Ширина импульса $T_H$ и промежуток между импульсами $T_L$ являются регулируемыми. <b><math>N</math></b> определяет число циклов импульс/пауза $T_L / T_H$ : Диапазон значений: 1...9. <b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.
Выход <b>Q</b>	Выход Q устанавливается по истечении времени $T_L$ и сбрасывается по истечении времени $T_H$ .

### Описание функции

При изменении состояния входа Trg на 1 запускается время  $T_L$  (время низкого уровня) (рис. 5.17). По истечении времени  $T_L$  выход Q устанавливается в 1 на время, равное значению  $T_H$  (время высокого уровня). В случае перезапуска входа Trg до истечения предустановленного времени ( $T_L + T_H$ ) время  $T_a$  сбрасывается и отсчет периода импульса/паузы запускается снова

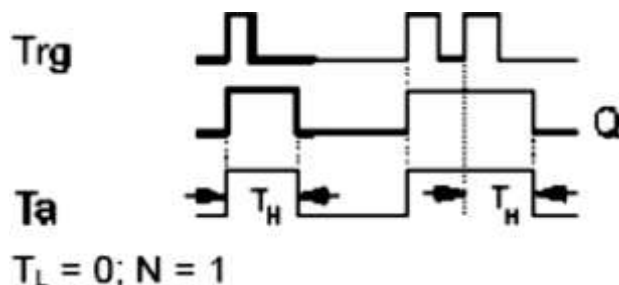


Рис. 5.17. Временная диаграмма интервального реле с запуском по фронту

#### 5.3.1.7. Генератор симметричных импульсов

Генератор симметричных импульсов (рис. 5.18, табл. 5.8) доступен только для устройств серии 0BA3. Устройства LOGO! текущих серий используют асинхронный генератор импульсов вместо генератора симметричных импульсов.

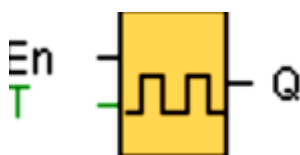


Рис. 5.18. Блок функции генератора симметричных импульсов

Таблица 5.8

### Параметры функции генератора симметричных импульсов

Соединение	Описание
Вход <b>En</b>	Разрешающий вход. Разрешить ( $En=1$ ) или запретить ( $En=0$ ) работу генератора импульсов.
Параметр	<b>T</b> время включения или выключения выхода.
Выход <b>Q</b>	Q периодически включается и выключается, период импульса равняется T.

### Описание функции

Продолжительность времени включения и выключения задаётся параметром T (рис. 5.19). Вход En включает генератор импульсов. Генератор импульсов устанавливает на выходе значение 1 в течение времени T, затем 0 в течение времени T, и так далее, пока на входе не будет установлено  $En = 0$ . Всегда следует указывать время T, равное 0,1 с. Время T не определено для  $T = 0,05$  с и  $T = 0,00$  с.

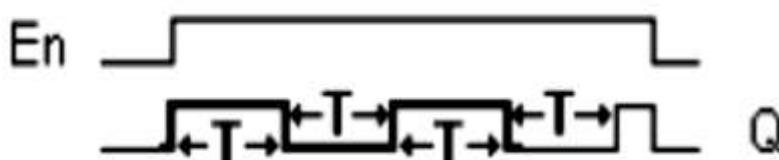


Рис. 5.19. Временная диаграмма генератора симметричных импульсов

### 5.3.1.8. Асинхронный генератор импульсов

Асинхронный генератор показан на рис. 5.20 (табл. 5.9).

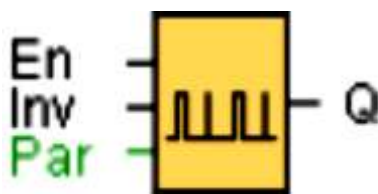


Рис. 5.20. Блок функции асинхронного генератора импульсов

### Описание функции

Форма выходного импульса может быть изменена благодаря применению конфигурируемого отношения импульс/пауза (рис. 5.21). Можно задать отношение продолжительности импульса/паузы при помощи параметров TH (время высокого уровня) и TL (время низкого уровня). Вход INV может использоваться для инвертирования выходного сигнала. Входной блок INV инвертирует выходной сигнал только в том случае, если работа блока разрешена сигналом EN.

Таблица 5.9

### Параметры функции асинхронного генератора импульсов

Соединение	Описание
Вход <b>En</b>	Разрешающий вход асинхронного генератора импульсов.
Вход <b>Inv</b>	Вход Inv может использоваться для инверсии выходного сигнала активного асинхронного генератора импульсов.
Параметр	<b>T<sub>H</sub>, T<sub>L</sub></b> : Вы можете регулировать ширину импульса (T <sub>H</sub> ) и промежуток между импульсами (T <sub>L</sub> ). <b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.
Выход <b>Q</b>	Q периодически включается и выключается, периоды импульса/паузы равны T <sub>H</sub> и T <sub>L</sub> .

### Параметры T<sub>H</sub> и T<sub>L</sub>

Ширина импульса T<sub>H</sub> и промежуток между импульсами T<sub>L</sub> могут быть получены как значение другой ранее запрограммированной функции:

- Аналоговый компаратор: Ax – Ay;
- Аналоговый выключатель: Ax;
- Аналоговый усилитель: Ax;
- Аналоговый мультиплексор: AQ;
- Линейно нарастающий аналоговый сигнал: AQ;
- Блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- Реверсивный счетчик: Cnt.



Рис. 5.21. Временная диаграмма

### 5.3.1.9. Генератор случайных импульсов

Выход генератора (рис. 5.22) случайных импульсов меняется по истечении устанавливаемого параметра интервала времени (табл. 5.10).

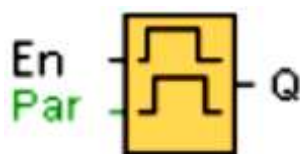


Рис. 5.22. Блок функции генератора случайных импульсов

### Описание функции

При переходе из 0 в 1 (рис. 5.23) на входе  $E_n$  устанавливается произвольное время (время задержки включения) от 0 с до  $T_H$  и производится запуск. Если вход  $E_n$  находится в состоянии 1 хотя бы в течение времени задержки включения, то выход устанавливается равным 1 по истечении этого времени задержки включения. Время сбрасывается, если состояние входа  $E_n$  было сброшено в 0 до того, как истечет время задержки включения. При сбросе в 0 входа  $E_n$  устанавливается произвольное время (время задержки выключения) от 0 с до  $T_L$  и производится запуск.

Таблица 5.10

### Параметры функции

Соединение	Описание
Вход $E_n$	Положительный фронт (переход из 0 в 1) на входе разрешения $E_n$ (разрешение) запускает задержку включения генератора случайных импульсов. Отрицательный фронт (переход из 1 в 0) запускает задержку выключения генератора случайных импульсов.
Параметр	<b><math>T_H</math></b> : Задержка включения определяется случайным образом и находится в пределах от 0 с до $T_H$ . <b><math>T_L</math></b> : Задержка выключения определяется случайным образом и находится в пределах от 0 с до $T_L$ .
Выход $Q$	$Q$ устанавливается по истечении времени включения, если $E_n$ по-прежнему находится в установленном состоянии. Он сбрасывается, если время задержки выключения истекло и если $E_n$ не был установлен повторно.

Если вход  $E_n$  находится в состоянии 0 хотя бы в течение времени задержки выключения, то выход  $Q$  устанавливается в 0 по истечении этого времени задержки выключения. Время сбрасывается, если состояние входа  $E_n$  возвращается в 1 до того, как истечет время задержки включения.

### Параметры $T_H$ и $T_L$

Время задержки включения  $T_H$  и время задержки выключения  $T_L$  могут быть получены как значение другой ранее запрограммированной функции:

- аналоговый компаратор:  $A_x - A_y$ ;



- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

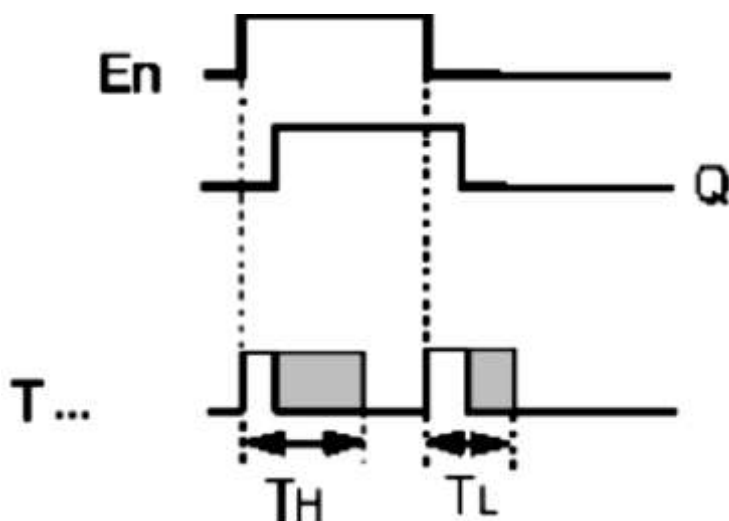


Рис. 5.23. Временная диаграмма задержки включения

#### 5.3.1.10. Выключатель освещения

Фронт на входе блока выключателя освещения (рис. 5.24, табл. 5.11) запускает отсчет конфигурируемого интервала времени. Выход сбрасывается после истечения этого времени. Перед истечением этого времени может быть выведено предупреждение о выключении.

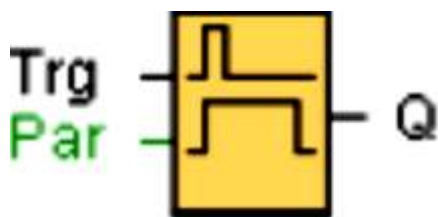


Рис. 5.24. Блок функции выключателя освещения

#### Параметры $T$ , $T_I$ и $T_{IL}$

Время задержки выключения  $T$ , время предварительного предупреждения  $T_I$  и период предварительного предупреждения  $T_{IL}$  могут быть получены как значение другой ранее запрограммированной функции (рис. 5.25).

Параметры функции

Соединение	Описание
Вход <b>Trg</b>	Вход выключателя на лестнице сигналом на входе Trg (выключатель).
Параметр	<p><b>T</b>: Выход сбрасывается (переход из 1 в 0) по истечении времени задержки выключения T.</p> <p><b>T<sub>I</sub></b> определяет время срабатывания предварительного предупреждения.</p> <p><b>T<sub>IL</sub></b> определяет продолжительность предварительного предупреждения.</p> <p><b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.</p>
Выход <b>Q</b>	Q сбрасывается по истечении времени T. До истечения этого времени может быть выдан предупреждающий сигнал.



Рис. 5.25. Временная диаграмма задержки включения

**0BA1, 0BA3:**

Время предупреждения установлено равным 15 с. Параметры T<sub>I</sub> и T<sub>IL</sub> таким образом становятся ненужными (рис. 5.26).

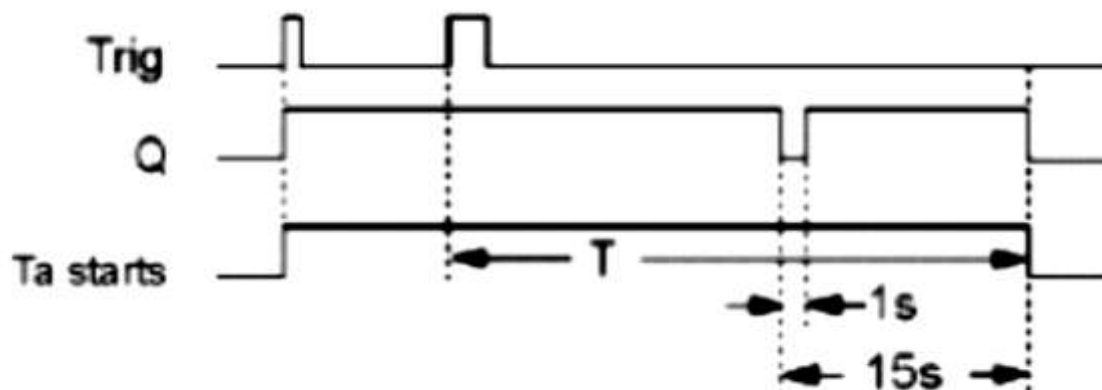


Рис. 5.26. Временная диаграмма задержки включения (0BA1, 0BA3)

### Изменение временного интервала

Изменяется временной интервал предварительного предупреждения и период (табл. 5.12).

Таблица 5.12

Параметры функции

Временной интервал Т	Время предварительного предупреждения	Период предварительного предупреждения
Секунды*	750 мс	50 мс
Минуты	15с	1 с
Часы	15 мин.	1 мин.

\* Имеет смысл только в программах с временем цикла < 25 мс

### Описание функции

Выход Q кратковременно устанавливается в 1 при переходе сигнала из 0 в 1 на входе Trg. Переход из 1 в 0 на входе Trg взводит текущее время и выход остается в установленном состоянии.

Выход Q сбрасывается в 0, когда Та достигает времени Т. Перед тем, как время задержки выключения ( $T - T_I$ ) истечет, вы можете выдать предварительное предупреждение, сбрасывающее Q в течение интервала времени предварительного предупреждения  $T_{IL}$ . Та перезапускается (по выбору) при следующем переходе из высокого в низкое состояние на входе Trg, при условии истечения Та.

#### 5.3.1.11. Многофункциональный выключатель

Импульсный переключатель с задержкой выключения «Выключатель (непрерывный свет)» (рис. 5.27).



Рис. 5.27. Блок функции многофункционального выключателя)

### Параметры Т, $T_L$ , $T_I$ и $T_{IL}$

Время задержки выключения Т (табл. 5.13, рис. 5.28, 5.29), время постоянного свечения  $T_L$ , время задержки включения времени предварительного предупреждения  $T_I$  и период времени предварительного предупреждения  $T_{IL}$  могут быть получены как значение другой, ранее запрограммированной функции.

Таблица 5.13

Параметры функции многофункционального выключателя

Соединение	Описание
Вход <b>Trg</b>	Сигнал на входе Trg (Trigger = запуск) устанавливает выход Q (постоянное освещение) или сбрасывает выход Q с задержкой отключения. Если на выходе Q присутствует сигнал 1, его можно сбросить, подав сигнал на вход Trg.
Вход <b>R</b>	Сигнал на входе R сбрасывает текущее время $T_a$ и сбрасывает выход.
Параметр	<p><math>T</math>: определяет время задержки выключения. Выход сбрасывается (переход из 1 в 0) по истечении времени <math>T</math>.</p> <p><math>T_L</math> задает период, в течение которого вход должен быть установлен для обеспечения постоянного освещения.</p> <p><math>T_I</math> определяет задержку включения времени предварительного предупреждения.</p> <p><math>T_{IL}</math> определяет продолжительность периода времени предварительного предупреждения.</p> <p><b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.</p>
Выход <b>Q</b>	Выход Q устанавливается сигналом на входе Trg, и снова сбрасывается по истечении заданного времени. В зависимости от ширины импульса на входе Trg он сбрасывается при подаче другого сигнала на вход Trg.

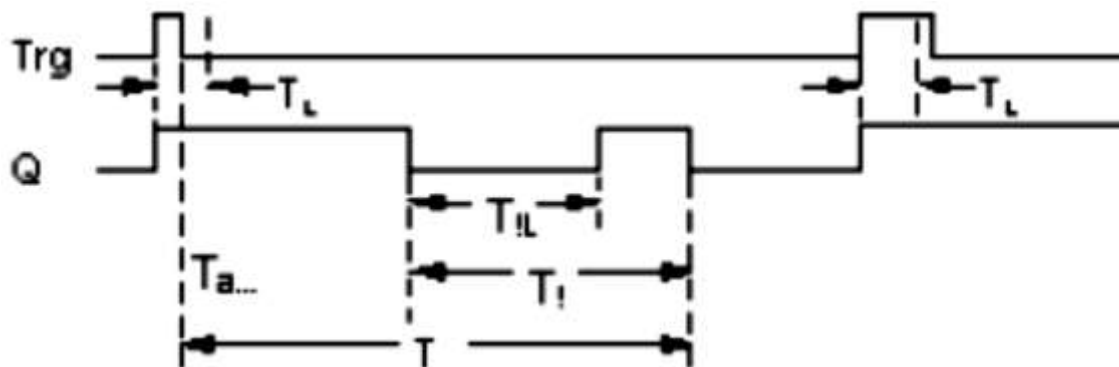


Рис. 5.28. Временная диаграмма многофункционального выключателя

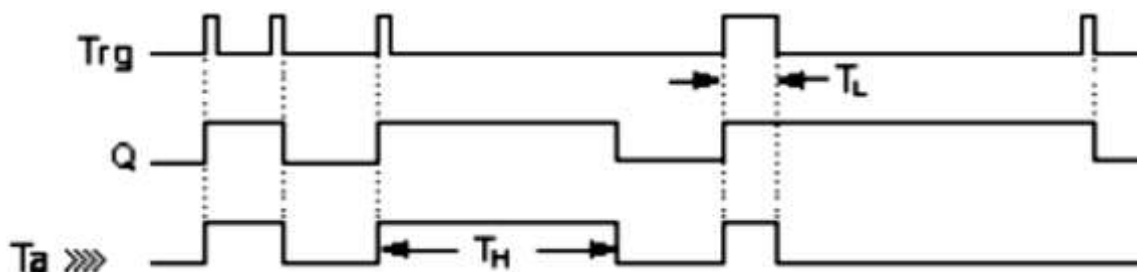


Рис. 5.29. Временная диаграмма

### 0BA2, 0BA3:

Имеются только параметры  $T_L$  и  $T$ . Выход сбрасывается после истечения времени  $T$ .  $T_L$  задает период, в течение которого вход дол-

жен быть установлен для обеспечения постоянного свечения. Вход R недоступен для пользователя.

### Описание функции

Выход Q устанавливается в 1 при переходе сигнала из 0 в 1 на входе Trg. Если выход  $Q = 0$  и вход Trg были установлены в высокое положение в течение (как минимум) длительности  $T_L$ , то функция постоянного свечения задействуется и выход Q будет установлен соответствующим образом.

Время задержки выключения T сбрасывается, если состояние входа Trg переходит в 0 до того, как истечет время  $T_L$ . Выход Q сбрасывается после наступления момента  $T_a = T$ . Можно вывести сигнал предупреждения о выключении до истечения времени задержки выключения  $(T - T_I)$ , сбрасывающего Q в течение интервала времени предварительного предупреждения  $T_{IL}$ . Следующий сигнал на входе Trg обязательно сбрасывает T и выход Q.

Внимание! Временные интервалы T,  $T_I$  и  $T_{IL}$  должны быть идентичными.

### Вопросы для самоконтроля

1. Перечислите типы специальных функций для анализа интервалов времени?
2. Что такое параметр Trg?
3. Как задаётся параметр T?
4. Каковы характеристики генератора случайных импульсов?

## 5.3.2. Счетчики

### 5.3.2.1. Реверсивный счетчик

Входной импульс увеличивает или уменьшает внутреннее значение счётчика Cnt (рис. 5.30) в зависимости от настройки параметра (табл. 5.14). Выход устанавливается или сбрасывается по достижении заданного порогового значения. Направление счета может быть изменено сигналом на входе Dir.

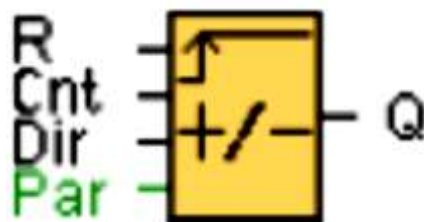


Рис. 5.30. Блок функции реверсивного счётчика

## Параметры On и Off

Порог включения On и порог выключения Off могут быть получены как значение другой ранее запрограммированной функции:

- аналоговый компаратор: Ax – Ay;
- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

Таблица 5.14

Параметры функции реверсивного счётчика

Соединение	Описание
Вход <b>R</b>	Выход и значение во внутреннем счетчике сбрасываются и принимают исходное значение (StartVal) сигналом на входе R (сброс).
Вход <b>Cnt</b>	Эта функция подсчитывает переходы из 0 в 1 на входе Cnt. Она не подсчитывает переходы из 1 в 0. – входы I3, I4, I5 и I6 используются в качестве высокочастотных счетчиков (LOGO! 12/24 RC/RCo и LOGO! 24/24o): макс. 2 кГц; – используйте любые остальные счетчики или элементы схем в качестве низкочастотных счетчиков (обычно – 4 Гц).
Вход <b>Dir</b>	Вход Dir (направление) определяет направление счета: Dir = 0: Вверх Dir = 1: Вниз:
Параметр	<b>On:</b> Порог включения Диапазон значений: 0...999999 <b>Off:</b> Порог выключения Диапазон значений: 0...999999 <b>StartVal:</b> Начальное значение, с которого начинается счет вверх или вниз. <b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.
Выход <b>Q</b>	Q устанавливается и сбрасывается в соответствии с фактическим значением Cnt и установленными порогами.

## Описание функции

Функция (рис. 5.31) увеличивает (Dir = 0) или уменьшает (Dir = 1) на единицу состояние внутреннего счетчика при каждом положительном фронте на входе Cnt.

Значение внутреннего счетчика можно сбросить до начального значения сигналом на входе сброса R. Пока R = 1, на выходе Q установлено значение 0, и подсчет импульсов на входе Cnt не выполняется.

Выход Q устанавливается и сбрасывается в соответствии с фактическим значением Cnt и установленными порогами. См. приведенные ниже правила расчета.

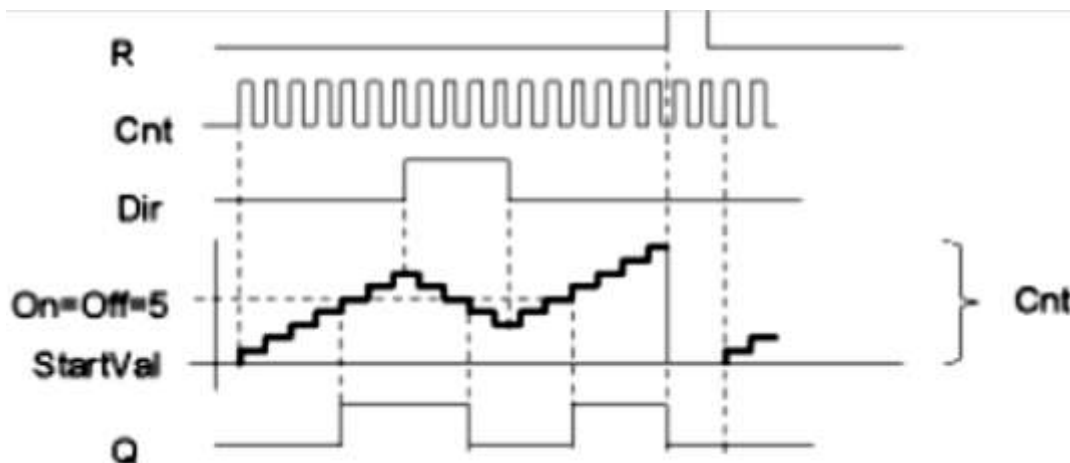


Рис. 5.31. Временная диаграмма реверсивного счётчика

### Правило расчета

- Если порог включения  $\geq$  порога выключения, то:  
 $Q = 1$ , если  $Cnt \geq On$   
 $Q = 0$ , если  $Cnt < Off$ .
- Если порог включения  $<$  порога выключения, то:  
 $Q = 1$ , если  $On \leq Cnt < Off$ .

**0BA0-0BA5:** Параметр StartValue не существует. Счетчик всегда считает вверх или вниз с 0.

**0BA0-0BA3:** Параметр off не существует. Следовательно, правило вычисления недействительно.

Внимание! Функция опрашивает предельное значение в счетчике один раз за цикл.

Следовательно, если импульсы на быстродействующих входах I3, I4, I5 или I6 следуют быстрее времени цикла сканирования, специальный функциональный блок может не переключиться до превышения заданного предела.

Пример. Может быть подсчитано до 100 импульсов за цикл; до настоящего момента было подсчитано 900 импульсов. Включение = 950, отключение = 10000. Выход устанавливается в следующем цикле после достижения значения 1000.

Выход не будет устанавливаться никогда, если значение Off = 980.

Время цикла сканирования Информация о порядке определения времени цикла сканирования в системе LOGO! см. соответствующее руководство LOGO!.

#### 5.3.2.2. Счетчик рабочего времени

Отсчет заданного времени запускается по сигналу на входе контроля En (рис. 5.32). Выход устанавливается после истечения этого времени (табл. 5.15, рис. 5.33).

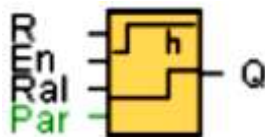


Рис. 5.32. Блок функции счётчика рабочего времени

### Параметр MI

Значение интервала технического обслуживания MI может быть получено как значение другой ранее запрограммированной функции:

- аналоговый компаратор: Ax – Ay;
- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

Таблица 5.15

Параметры функции счётчика рабочего времени

Соединение	Описание
Вход <b>R</b>	Положительный фронт (изменение состояния с 0 на 1) на входе R сбрасывает выход Q и устанавливает заданное значение MI для счетчика оставшегося времени (MN).
Вход <b>En</b>	En – вход контроля. LOGO! сканирует время включения на этом входе.
Вход <b>Ral</b>	Положительный фронт на входе Ral (Общий сброс) сбрасывает счетчик рабочего времени (OT) и выход и устанавливает заданное значение для счетчика оставшегося времени (MN) равным заданному интервалу технического обслуживания (MI): <ul style="list-style-type: none"> <li>• Выход Q = 0.</li> <li>• Измеренная наработка в часах OT = 0.</li> <li>• Оставшееся время интервала технического обслуживания MN = MI.</li> </ul>
Параметр	<p><b>MI:</b> Интервал технического обслуживания, задаваемый в часах и минутах.  Диапазон значений: 0000...9999 ч, 0...59 мин.</p> <p><b>OT:</b> Суммарное общее рабочее время. Смещение начального времени может быть задано в часах и минутах.  Диапазон значений: 00000...99999 ч, 0...59 мин.</p> <p><b>Q</b> → 0:</p> <ul style="list-style-type: none"> <li>• Если «R» выбрано:  Q = 1, если MN = 0;  Q = 0, если R = 1 или Ral = 1.</li> <li>• Если «R+En» выбрано:  Q = 1, если MN = 0;  Q = 0, если R = 1 или Ral = 1 или En = 0.</li> </ul>



Соединение	Описание
Выход Q	<p>Выход устанавливается после истечения оставшегося времени <math>MN = 0</math>. Выход сбрасывается:</p> <ul style="list-style-type: none"> <li>• Если «<math>Q \rightarrow 0:R+En</math>», если <math>R = 1</math> или <math>Ral = 1</math> или <math>En = 0</math>.</li> <li>• Если «<math>Q \rightarrow 0:R</math>», если <math>R = 1</math> или <math>Ral = 1</math>.</li> </ul>

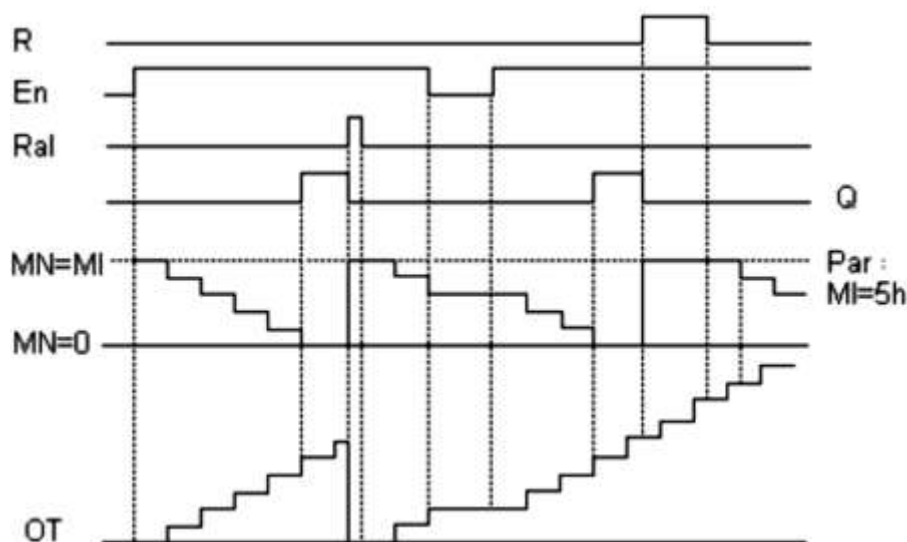


Рис. 5.33. Временная диаграмма счётчика рабочего времени:  
*Ml* – заданный интервал времени; *MN* – оставшееся время;  
*OT* – суммарное время, истекшее после последнего сигнала 1 на входе *Ral*

Все эти значения всегда являются значениями с сохранением.

### Описание функции

Счетчик рабочего времени контролирует вход *En*. Пока этот вход находится в состоянии 1, LOGO! рассчитывает истекшее время и оставшееся время *MN*. LOGO! отображает эти значения времени при работе в режиме конфигурирования. Выход устанавливается в 1 после того, как оставшееся время станет равным нулю.

Сброс выхода *Q* и счетчика оставшегося времени и установка исходного значения *Ml* производятся подачей сигнала на вход *R*. Это не влияет на работу счетчика рабочего времени *OT*.

Сброс выхода *Q* и счетчика оставшегося времени и установка исходного значения *Ml* производятся подачей сигнала на вход *Ral*. Счетчик рабочего времени *OT* при этом сбрасывается в 0.

В зависимости от выбранной конфигурации параметра *Q* выход либо сбрасывается сигналом сброса, подаваемым на вход *R* или *Ral* (« $Q \rightarrow R$ »), или когда сигнал сброса равен 1, или если сигнал *En* равен 0 (« $Q \rightarrow R+En$ »).

### Просмотр значений MI, MN и OT

В программе LSC можно выполнить выборку из счетчика рабочего времени при помощи команды меню «Сервис -> Передача»: Счетчик рабочего времени.

### Предельное значение OT

Значение рабочего времени в параметре OT сохраняется при сбросе счетчика рабочего времени сигналом на входе R. Счетчик рабочего времени OT продолжает отсчет до тех пор, пока  $E_n = 1$ , независимо от состояния входа сброса R.

Предельное значение счетчика OT равно 99999 часам. По достижении этого значения счетчик рабочего времени останавливается. В режиме программирования вы можете установить начальное значение OT. Счетчик начинает работу с любого значения, отличного от нуля. MN автоматически вычисляется при ЗАПУСКЕ на основании значений MI и OT.

**Пример:** MI = 100, OT = 130, результат MN = 70.

### Предустановка параметров

В LSC можно задать MI и начальное значение OT. Устанавливается, что Q не зависит от  $E_n$ , путем выбора соответствующего блока выбора.

### Сохранение и счетчик рабочего времени

Счетчик рабочего времени в LOGO! в целом обладает свойством сохранения. Тем не менее, если значения счетчика рабочего времени будут потеряны из-за аварии (пропадания) питания, то выберите соответствующий блок в коммутационной программе. Нажмите правой кнопкой мыши на счетчик рабочего времени и выберите «Свойства блока > Параметры2».

Функция «Сохранение» должна быть задействована и не допускать изменения (отображаться серым). Если функция Сохранение недоступна, удалите блок и вставьте новую специальную функцию «счетчик рабочего времени» в этом же положении.

### 0BA0-0BA5

Интервал техобслуживания (MI) и начальное время для рабочего времени (OT) выражались в часах. Эти значения не обеспечивались другими функциями в сериях устройств, выпускавшихся до серии 0BA6.

#### 5.3.2.3. Пороговый выключатель

Выход включается и выключается в зависимости от двух конфигурируемых частот (рис. 5.34, табл. 5.16).



Рис. 5.34. Блок функции порогового выключателя

### Параметр G\_T

Интервал времени G\_T может быть получен как значение другой ранее запрограммированной функции. fa – входная частота (рис. 5.35).

### Описание функции

Выключатель измеряет сигналы на входе Fre показанные на рис. 5.34. Импульсы учитываются во время программируемого периода G\_T. Q устанавливается или сбрасывается в соответствии с заданными пороговыми значениями. Ниже приведены правила расчета порогового выключателя.

Таблица 5.16

Параметры функции порогового выключателя

Соединение	Описание
Вход <b>Fre</b>	Эта функция подсчитывает переходы из 0 в 1 на входе Fre. Переходы из 1 в 0 не подсчитываются. <ul style="list-style-type: none"> <li>Используйте входы I3, I4, I5 и I6 в качестве высокочастотных счетчиков (LOGO! 12/24 RC/RCo и LOGO! 24/24o): макс. 2 кГц;</li> <li>Используйте любые остальные счетчики или элементы схем в качестве низкочастотных счетчиков (обычно – 4 Гц).</li> </ul>
Параметр	<b>On:</b> Порог включения Диапазон значений: 0000...9999 <b>Off:</b> Порог выключения Диапазон значений: 0000...9999 <b>G_T:</b> Интервал времени, в течение которого измеряются входные импульсы. Диапазон значений: 00:05 с...99:99 с
Выход <b>Q</b>	Q устанавливается или сбрасывается в соответствии с пороговыми значениями.

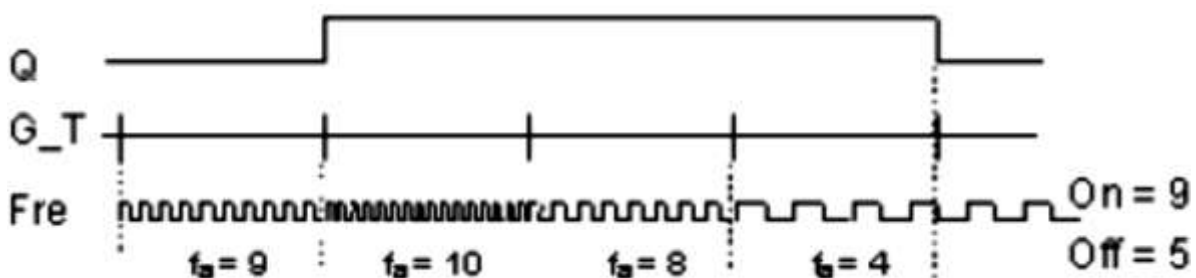


Рис. 5.35. Временная диаграмма порогового выключателя

### Правило расчета параметра порогового выключателя

- Если порог включения (On) > порога выключения (Off), то:  
 $Q = 1$ , если  $f_a \geq \text{On}$   
 $Q = 0$ , если  $f_a < \text{Off}$ .
- Если порог включения (On) < порога выключения (Off), то  $Q = 1$ , если  $\text{On} \leq f_a < \text{Off}$ .

#### **0BA0-0BA3:**

Применяются следующие расчеты:

- Если  $F_{re} > \text{порога (On)}$ , то:  $Q = 1$ .
- Если  $F_{re} \leq \text{порога (Off)}$ , то:  $Q = 0$ .

### 5.3.3. Аналоговые функции

#### 5.3.3.1. Аналоговый пороговый выключатель

Выход порогового выключателя (рис. 5.36) устанавливается или сбрасывается с учетом двух конфигурируемых пороговых значений параметра **On** и **Off** (гистерезис) (табл. 5.17). Значение параметров On и Off может быть получено как значение другой, ранее запрограммированной, функции.



Рис. 5.36. Блок функции аналогового порогового выключателя

Параметры принимают значения в зависимости от используемых модулей. Значения параметров устанавливаются через меню настроек, показанных на рис. 5.37.

#### **0BA4:**

A: Усиление

Диапазон значений: 0,00 – 10,00

#### **0BA2, 0BA3:**

Эти параметры применяются:

G: Усиление %

Диапазон значений: 0 – 1000%

O: Смещение. Диапазон значений: 999

On: Порог включения On. Диапазон значений: 0 – 9999

Off: Порог выключения Off. Диапазон значений: 0 – 9999

Параметр  $r$  (число разрядов после десятичной точки). Параметр  $r$  относится только к отображению значений On, Off и Ax в тексте сообщения. Параметр  $r$  не относится к сравнению значений On и Off. (Функция сравнения игнорирует десятичную точку.)

Таблица 5.17

### Параметры аналогового порогового выключателя

Соединение	Описание
Вход <b>Ax</b>	Вход аналогового сигнала, оцениваемого на входе Ax. Используйте аналоговые входы AI1...AI8, аналоговые флаги AM1...AM6, номер блока функции с аналоговым выходом, или аналоговые выходы AQ1 и AQ2. 0 – 10 В пропорционально 0 – 1000 (внутреннее значение).
Параметр	<b>A:</b> Усиление Диапазон значений: +- 10,00 <b>B:</b> Смещение нуля Диапазон значений: +- 10,000 <b>On:</b> Порог включения Диапазон значений: +- 20,000 <b>Off:</b> Порог выключения Диапазон значений: +- 20,000 <b>p:</b> Число разрядов после десятичной точки Диапазон значений: 0, 1, 2, 3
Выход <b>Q</b>	Q устанавливается или сбрасывается в соответствии с заданными пороговыми значениями.

### Описание функции

Функция считывает значение сигнала на аналоговом входе Ax (рис. 5.37). Это значение умножается на значение параметра A (усиление). Параметр B (смещение) добавляется к произведению, таким образом **(Ax \* усиление) + смещение = фактическое значение Ax**.

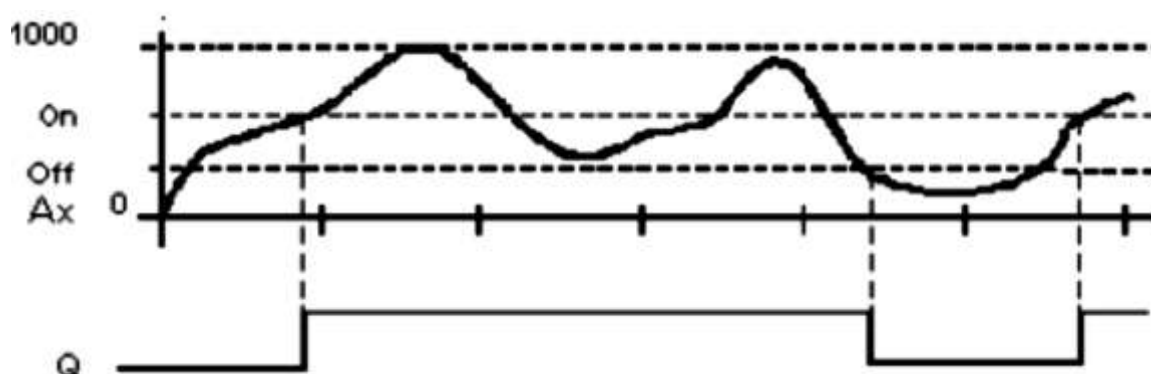


Рис. 5.37. Временная диаграмма работы аналогового порогового выключателя

Выход Q устанавливается или сбрасывается в соответствии с заданными пороговыми значениями. Ниже приведено правило расчета.

#### **ОВА2, ОВА3:**

Функция выглядит следующим образом:

Параметр смещения добавляется к считанному аналоговому значению. Сумма умножается на значение параметра усиления.

$$\text{Значение} = (AI + \text{смещение}) * \text{усиление}$$

Выход Q устанавливается равным 1, если вычисленное значение превышает порог включения (ТН высокое).

Q сбрасывается в 0, если значение достигает или снижается ниже порога выключения (ТН низкое). Меню порогового выключателя показано на рис. 5.38.

### Правило расчета

- Если порог включения (On)  $\geq$  порогу выключения (Off), то:  
Q = 1, если фактическое значение  $Ax > On$ ;  
Q = 0, если фактическое значение  $Ax \leq Off$ .
- Если порог включения (On) < порога выключения (Off), то:  
Q = 1, если  $On \leq \text{фактическое значение } Ax < Off$ .

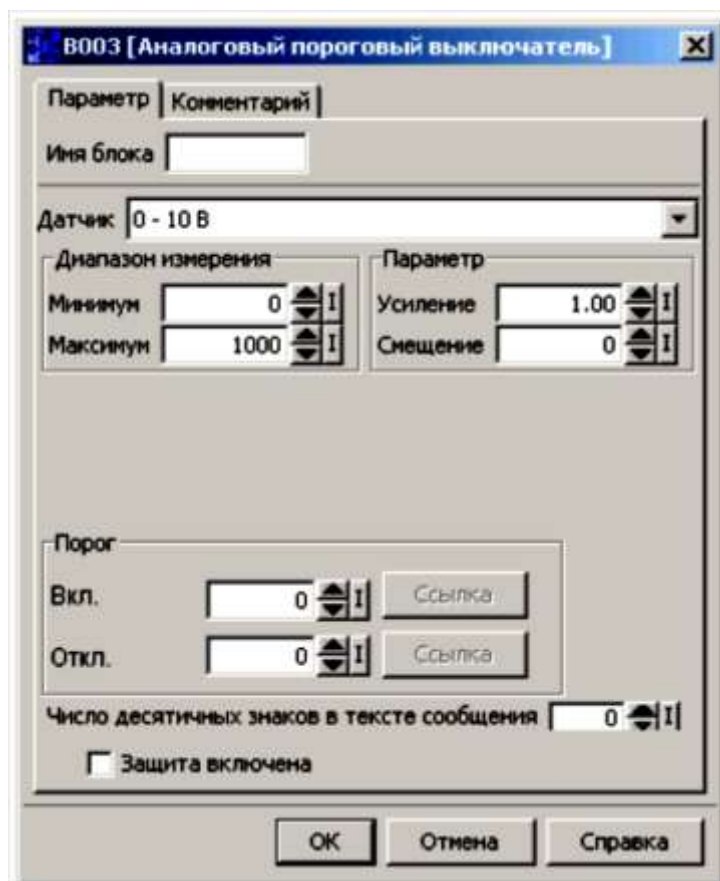


Рис. 5.38. Меню настроек порогового выключателя

Следует заметить, что настройка десятичной точки должна быть идентичной для минимального и максимального диапазонов.

#### 5.3.3.2. Аналоговый дифференциальный выключатель

Выход устанавливается и сбрасывается в зависимости от настраиваемого порога и значения разности (рис. 5.39, табл. 5.18).

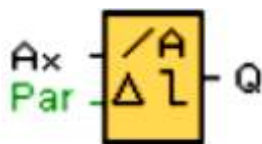


Рис. 5.39. Блок функции дифференциального выключателя

Таблица 5.18

Параметры функции дифференциального выключателя

Соединение	Описание
Вход <b>Ax</b>	Анализируемый аналоговый сигнал подается на вход Ax. Используйте аналоговые входы AI1...AI8, аналоговые флаги AM1...AM6, номер блока функции с аналоговым выходом, или аналоговые выходы AQ1 и AQ2. 0 – 10 В пропорционально 0 – 1000 (внутреннее значение).
Параметр	<b>A:</b> Усиление Диапазон значений: +- 10,00 <b>B:</b> Смещение нуля Диапазон значений: +- 10,000 <b>On:</b> Порог включения On Диапазон значений: +- 20,000 <b>Delta:</b> Значение разницы для расчета параметра выключения Диапазон значений: +- 20,000 <b>p:</b> Число разрядов после десятичной точки Диапазон значений: 0, 1, 2, 3
Выход <b>Q</b>	Выход Q устанавливается или сбрасывается в зависимости от порогового значения и значения разности.

**ОВА4:**

A: Усиление

Диапазон значений: 0,00...10,00

**Параметр p (число разрядов после десятичной точки)**

Параметр p относится только к отображению значений On, Off и Ax в тексте сообщения.

Параметр p не относится к сравнению значений On и Off. (Функция сравнения игнорирует десятичную точку.)

**Описание функции**

Функция считывает аналоговый сигнал на входе Ax.

Ax умножается на значение параметра A (усиление), а значение параметра B(смещение) прибавляется к произведению, т.е.

**(Ax \* усиление) + смещение = фактическое значение Ax.**

Выход Q устанавливается или сбрасывается в зависимости от заданного порогового значения (On) и значений разности (Delta).



Функция автоматически вычисляет параметр Off.  $Off = On + Delta$ , где Delta может быть положительной или отрицательной. Ниже приведено правило расчёта.

### Правило расчета

- При установке отрицательного значения разности Delta, порог включения  $On \geq$  порога выключения Off, и:

$Q = 1$ , если фактическое значение  $Ax > On$

$Q = 0$ , если фактическое значение  $Ax \leq Off$ .

Временная диаграмма показана на рис. 5.40.

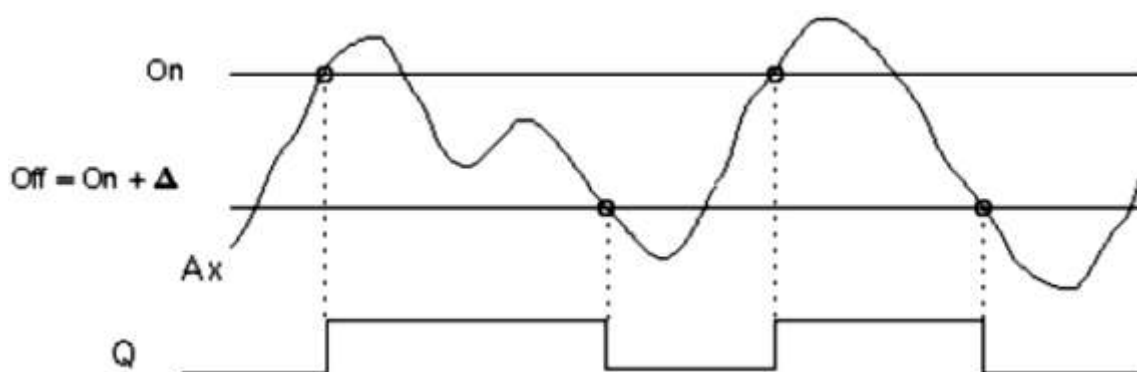


Рис. 5.40. Временная диаграмма функции с отрицательной разностью Delta

- При установке положительного значения разности Delta порог включения  $On <$  порога выключения Off и:

$Q \leq 1$ , если фактическое значение  $Ax < Off$ .

Временная диаграмма показана на рис. 5.41.

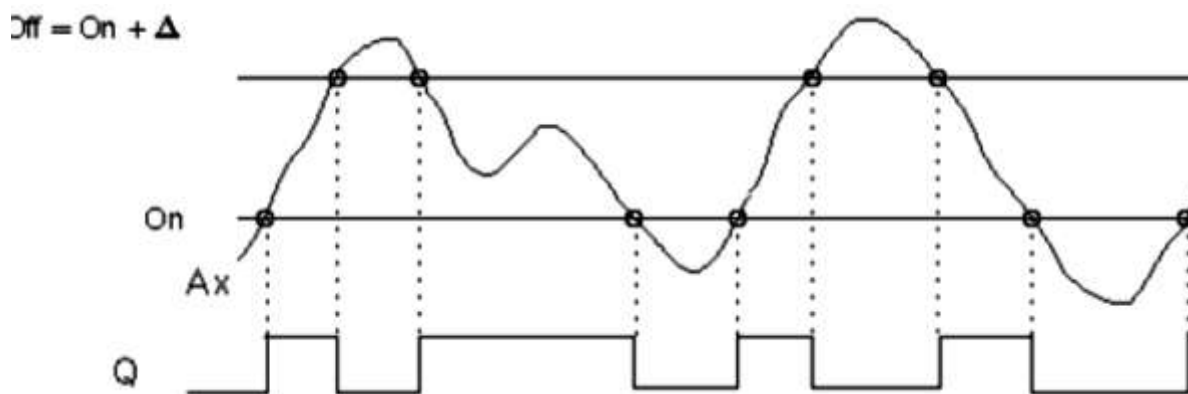


Рис. 5.41. Временная диаграмма функции с положительной разностью Delta

### 5.3.3.3. Аналоговый компаратор

Выход компаратора (рис. 5.42, 5.43) устанавливается и сбрасывается в зависимости от разности  $Ax - Ay$  и двух настраиваемых порогов (табл. 5.19).



Рис. 5.42. Блок функции интервального реле (импульсный выход)

Таблица 5.19

Параметры функции аналогового компаратора

Соединение	Описание
Входы <b>Ax, Ay</b>	Аналоговые сигналы, разность которых необходимо определить, на входы Ax и Ay. Используются аналоговые входы AI1...AI8, аналоговые флаги AM1...AM6, номер блока функции с аналоговым выходом, или аналоговые выходы AQ1 и AQ2. AI1...AI8: 0 – 10 В соответствует 0 – 1000 (внутреннее значение).
Параметр	<b>A:</b> Усиление Диапазон значений: +- 10,00 <b>B:</b> Смещение нуля Диапазон значений: +- 10,000 <b>On:</b> Порог включения On Диапазон значений: +- 20,000 <b>Off:</b> Порог выключения Диапазон значений: +- 20,000 <b>p:</b> Число разрядов после десятичной точки Диапазон значений: 0, 1, 2, 3
Выход Q	<b>Q</b> устанавливается или сбрасывается в соответствии с заданными пороговыми значениями.

#### **0BA4:**

A: Усиление

Диапазон значений: 0,00 ... 10,00

#### **0BA0-0BA3:**

Применяются следующие параметры:

G: Усиление, %

Диапазон значений: 0...1000 %

O: Смещение

Диапазон значений: +-999

Разность: Порог

Q устанавливается в 1, когда разность Ax-Ay превышает порог.

#### **Параметры On и Off**

Порог включения On и порог выключения Off могут быть получены как фактическое значение другой ранее запрограммированной функции.

#### **Параметр p (число разрядов после десятичной точки)**

Параметр p относится только к отображению значений Ax, Ay, Delta, On и Off в тексте сообщения.

Параметр  $p$  не относится к сравнению значений On и Off. (Функция сравнения игнорирует десятичную точку.)

$Q$  для  $Ax - Ay > 200$ , On = Off = 200

### Описание функции

Функция считывает значение сигнала на аналоговом входе  $Ax$ .

Это значение умножается на значение параметра  $A$  (усиление). Параметр  $B$  (смещение) добавляется к произведению, таким образом,

**$(Ax * \text{усиление}) + \text{смещение} = \text{фактическое значение } Ax$ .**

**$(Ay * \text{усиление}) + \text{смещение} = \text{фактическое значение } Ay$ .**

Выход  $Q$  устанавливается или сбрасывается в зависимости от разности фактических значений  $Ax - Ay$  и заданных пороговых значений. Ниже приведено правило расчета.

### Правило расчета

- Если порог включения (On)  $\geq$  порогу выключения (Off), то:  
 $Q = 1$ , если (фактическое значение  $Ax$  – фактическое значение  $Ay$ )  $>$  On;  
 $Q = 0$ , если (фактическое значение  $Ax$  – фактическое значение  $Ay$ )  $\leq$  Off.
- Если порог включения (On)  $<$  порога выключения (Off), то:  
 $Q = 1$ , если On  $\leq$  (фактическое значение  $Ax$  – фактическое значение  $Ay$ )  $<$  Off.

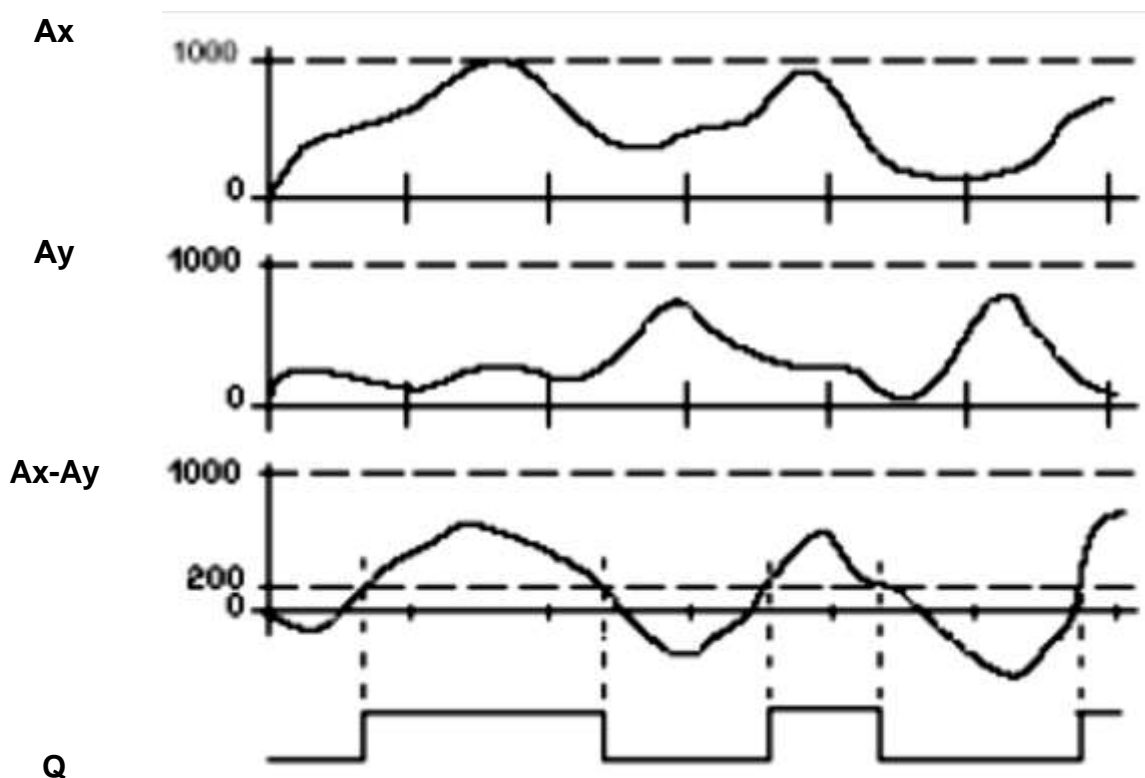


Рис. 5.43. Временная диаграмма задержки включения

## ОВА2, ОВА3

Применяются следующие функции/правила расчета

Функция добавляет соответствующее заданное смещение к аналоговым значениям  $A_x$  и  $A_y$ . Сумма умножается на значение параметра усиления. Разность формируется с использованием обоих расчетных значений.

Выход  $Q$  устанавливается, если эта разность между этими значениями превышает порог, заданный при помощи параметра разности  $\Delta$ .

Правило расчета:  $Q = 1$ , если:

$((A_x + \text{смещение}) * \text{усиление}) - ((A_y + \text{смещение}) * \text{усиление}) > \text{пороговое значение } \Delta$   $Q$  сбрасывается в 0, если пороговое значение достигает или снижается ниже значения  $\Delta$ .

### Снижение входной чувствительности аналогового компаратора

Можно задержать выходные значения аналогового компаратора при помощи специальных функциональных блоков «задержки включения» и «задержки выключения». Таким образом задаётся порядок, в соответствии с которым выход  $Q$  устанавливается только в том случае, когда продолжительность входного переключателя  $\text{Trg}$  (= выход аналогового компаратора) превышает заданное время задержки включения.

Этот способ позволяет задать виртуальный гистерезис, делающий вход менее чувствительным к кратковременным изменениям.

#### 5.3.3.4. Контроль аналоговых значений

Специальная функция контроля аналоговых значений (рис. 5.44) сохраняет текущее значение аналогового сигнала на входе в памяти и устанавливает выход, если отклонение выходной переменной от сохраненного значения превышает заданную величину.

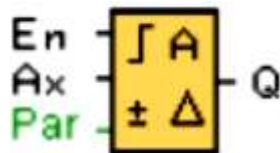


Рис. 5.44. Блок функции контроля аналоговых значений

**Параметры** «Пороговое значение 1» и «Пороговое значение 2». Значение двух пороговых параметров 1 и 2 (табл. 5.20) может быть получено как фактическое значение другой ранее запрограммированной функции. Выбирается требуемая функция по номеру блока. Пороговые значения показаны символом  $\Delta$  на модуле LOGO! Basic и на приведенной временной диаграмме на рис. 5.44.

**0ВА4:**

А: Усиление

Диапазон значений: 0,00...10,00

Параметр р (число разрядов после десятичной точки)

Параметр р относится только к отображению значений  $A_{en}$ ,  $A_x$ , Порог 1 и Порог 2 в тексте сообщения.

**Описание функции**

При переходе из 0 в 1 на входе  $E_n$  сохраняется значение сигнала на аналоговом входе  $A_x$ . Это сохраненное текущее значение обозначается « $A_{en}$ » (рис. 5.45).

Оба фактических аналоговых значения  $A_x$  и  $A_{en}$  умножается на значение параметра А (усиление), а значение параметра В (смещение) затем прибавляется к произведению следующим образом:

$(A_x \cdot \text{усиление}) + \text{смещение} = \text{Фактическое значение } A_{en}$ , когда вход  $E_n$  изменяет состояние с 0 на 1, или  $(A_x \cdot \text{усиление}) + \text{смещение} = \text{фактическое значение } A_x$ .

Выход Q устанавливается, когда сигнал на входе  $E_n = 1$ , и если фактическое значение на входе  $A_x$  выходит за пределы  $A_{en} + \text{Порог 1} / A_{en} - \text{Порог 2}$ .

Выход Q сбрасывается, когда фактическое значение на входе  $A_x$  выходит за пределы  $A_{en} + \text{Порог 1} / A_{en} - \text{Порог 2}$ , или когда сигнал на входе  $E_n$  изменяется на 0.

Таблица 5.20

Параметры функции контроля аналоговых значений

Соединение	Описание
Вход <b><math>E_n</math></b>	Положительный фронт (переход из 0 в 1) на входе $E_n$ сохраняет аналоговое значение на входе $A_x$ (« $A_{en}$ ») в память и начинает контролировать диапазон аналоговых значений $A_{en} \pm \Delta$ .
Вход <b><math>A_x</math></b>	Контролируемый аналоговый сигнал подается на вход $A_x$ . Используйте аналоговые входы $AI1...AI8$ , аналоговые флаги $AM1...AM6$ , номер блока функции с аналоговым выходом, или аналоговые выходы $AQ1$ и $AQ2$ . 0 – 10В пропорционально 0 – 1000 (внутреннее значение).
Параметр	<b>А:</b> Усиление Диапазон значений: $\pm 10,00$ <b>В:</b> Смещение нуля Диапазон значений: $\pm 10,000$ <b>Пороговое значение 1.</b> Значение разности выше $A_{en}$ : Порог включения и отключения Диапазон значений: 0 – 20,000 <b>Пороговое значение 2.</b> Значение разности ниже $A_{en}$ : порог включения и отключения Диапазон значений: 0 – 20,000 <b>р:</b> Число десятичных знаков Диапазон значений: 0, 1, 2, 3
Выход <b>Q</b>	Выход Q устанавливается и сбрасывается в зависимости от сохраненного аналогового значения и отклонения.

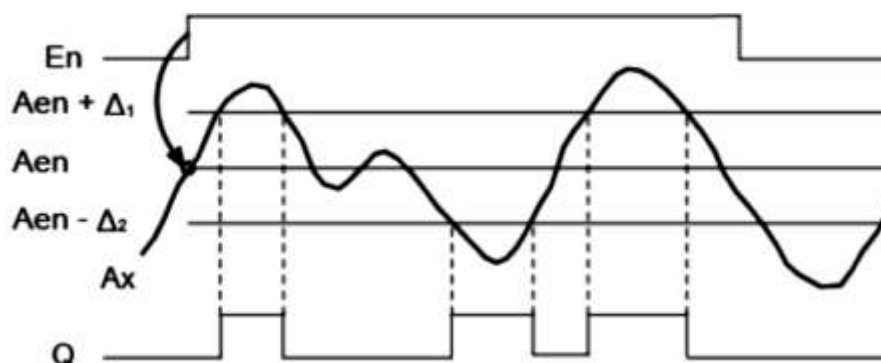


Рис. 5.45. Временная диаграмма функции блока контроля аналоговых значений

### 5.3.3.5. Аналоговый усилитель

Этот специальный функциональный блок (рис. 5.46) усиливает аналоговое входное значение и подает его на аналоговый выход.

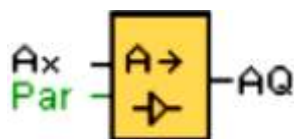


Рис. 5.46. Блок функции аналогового усилителя

Характеристики функции показаны в табл. 5.21.

Таблица 5.21

Параметры функции аналогового усилителя

Соединение	Описание
Вход <b>Ax</b>	Вход аналогового сигнала, усиливаемого на входе Ax. Используются аналоговые входы AI1...AI8, аналоговые флаги AM1...AM6, номер блока функции с аналоговым выходом или аналоговые выходы AQ1 и AQ2. AI1...AI8: 0 – 10 В соответствует 0 – 1000 (внутреннее значение).
Параметр	<b>A</b> : Усиление. Диапазон значений: +- 10.00 <b>B</b> : Смещение нуля. Диапазон значений: +- 10000 <b>p</b> : Число разрядов после десятичной точки Диапазон значений: 0, 1, 2, 3
Выход <b>AQ</b>	Аналоговый выход Диапазон значений AQ: -32768...+32767

Характеристики модуля **0BA4**:

A: Усиление

Диапазон значений: 0,00...10,00

**Параметр p (число разрядов после десятичной точки)**

Параметр p относится только к отображению значений Ax и Ay в тексте сообщения.

Параметр  $p$  не относится к сравнению значений On и Off. (Функция сравнения игнорирует десятичную точку.)

### Описание функции

Функция считывает значение аналогового сигнала на аналоговом входе  $Ax$ . Это значение умножается на усиление (параметр  $A$ ), а затем к результату прибавляется значение параметра  $B$  (смещение):

$$(Ax * \text{усиление}) + \text{смещение} = \text{фактическое значение } Ax.$$

Текущее значение  $Ax$  подается на выход  $AQ$ .

### Специальные характеристики, подлежащие учету при конфигурировании

#### Аналоговый выход

При подключении этой специальной функции к физическому аналоговому выходу (рис. 5.47) следует учитывать, что аналоговый выход может обрабатывать только значения от 0 до 1000. Поэтому следует подключить дополнительный усилитель между аналоговым выходом специальной функции и физическим аналоговым выходом. При помощи усилителя можно произвести масштабирование выходного диапазона специальной функции в пределах значений от 0 до 1000.

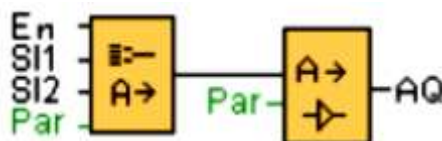


Рис. 5.47. Пример: дополнительный усилитель за аналоговым мультиплексором

#### 5.3.3.6. Аналоговый мультиплексор

Специальная функция аналогового мультиплексора отображает 0 или одно из 4-х сохраненных аналоговых значений аналогового выхода (рис. 5.48).

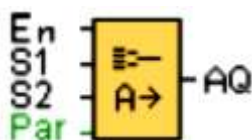


Рис. 5.48. Блок функции аналогового мультиплексора

Характеристики функции аналогового мультиплексора приведены в табл. 5.22.

### Параметры V1 – V4

Значения для параметров V1 – V4 могут быть получены как значение другой ранее запрограммированной, функции.



### Параметр *p* (число разрядов после десятичной точки)

Параметр *p* относится только к отображению значений AQ, V1, V2, V3 и V4 в тексте сообщения.

Таблица 5.22

#### Параметры функции аналогового мультиплексора

Соединение	Описание
Вход <i>En</i>	1 на входе <i>En</i> (разрешение) переключает, в зависимости от <i>S1</i> и <i>S2</i> , параметрическое аналоговое значение на выходе AQ. 0 на входе <i>En</i> переключает выход AQ в состояние 0.
Входы <b>S1</b> и <b>S2</b>	<i>S1</i> и <i>S2</i> (селекторы) для выбора выходного AQ. <i>S1</i> = 0 и <i>S2</i> = 0: Выдается значение 1 <i>S1</i> = 0 и <i>S2</i> = 1: Выдается значение 2 <i>S1</i> = 1 и <i>S2</i> = 0: Выдается значение 3 <i>S1</i> = 1 и <i>S2</i> = 1: Выдается значение 4
Параметр	<b>V1...V4</b> : Выдаваемые аналоговые значения Диапазон значений: -32768...+32767 <b>p</b> : Число десятичных мест Диапазон значений: 0, 1, 2, 3
Выход <b>AQ</b>	Аналоговый выход. (32768...+32767)

### Описание функции

В случае установки входа *En* (рис. 5.49), функция выдает одно из 4-х возможных аналоговых значений V1 – V4 на выход AQ в зависимости от значений параметров *S1* и *S2*.

Если вход *En* не установлен, то функция выводит аналоговое значение 0 на выходе AQ.

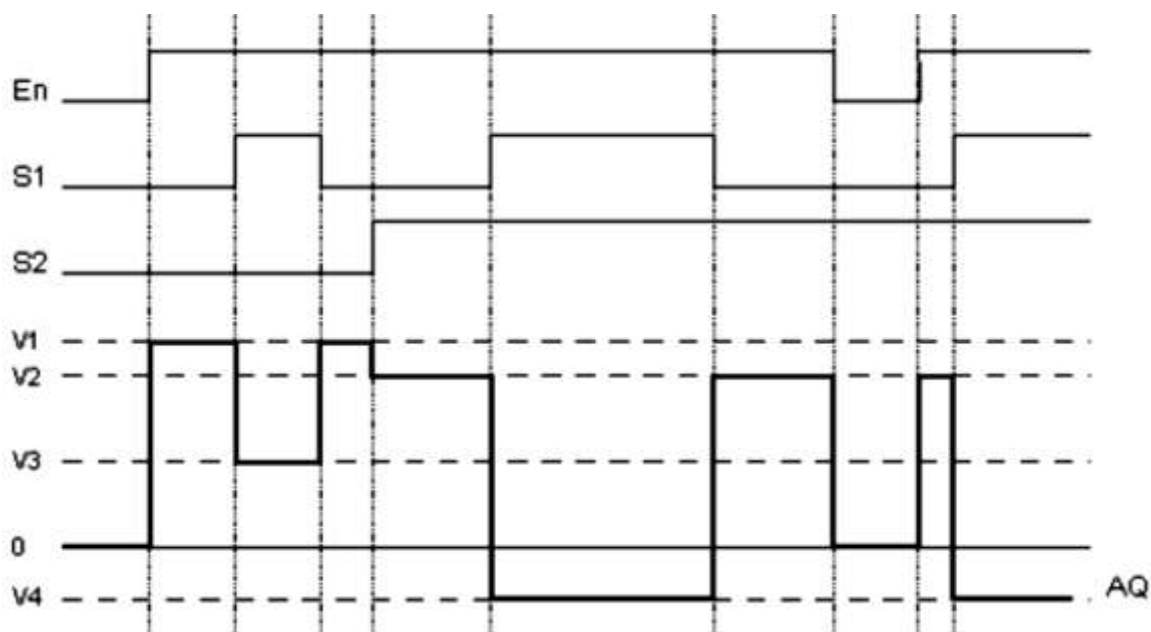


Рис. 5.49. Временная диаграмма задержки включения

### 5.3.3.7. Блок широтно-импульсного модулятора (ШИМ)

Функция широтно-импульсного модулятора (рис. 5.50) выполняет преобразование аналогового входного значения  $A_x$  в импульсный цифровой выходной сигнал. Длительность импульса пропорциональна аналоговому значению  $A_x$ . Параметры функционального блока приведены в табл. 5.23.

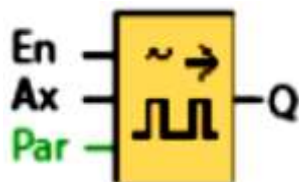


Рис. 5.50. Блок широтно-импульсного модулятора (ШИМ)

Таблица 5.23

Параметры функции широтно-импульсного модулятора (ШИМ)

Соединение	Описание
Вход <b>Ax</b>	Аналоговый сигнал для преобразования в импульсный цифровой выходной сигнал.
Параметр	<b>A</b> : Усиление. Диапазон значений: +-10.00 <b>B</b> : Смещение нуля. Диапазон значений: +-10,000 <b>PT</b> : Период времени, в течение которого модулируется выходной цифровой сигнал <b>p</b> : число разрядов после десятичной точки Диапазон значений: 0, 1, 2, 3
Выход <b>Q</b>	Выход Q устанавливается и сбрасывается в течение определенной части каждого интервала времени в соответствии с отношением стандартизованного значения $A_x$ к диапазону аналогового значения.

**0BA1-0BA5:** Блок ШИМ отсутствует в системах версии до 0BA6.

#### Параметр PT

Интервал времени PT может быть получен как значение другой, ранее запрограммированной функции.

#### Параметр p (число разрядов после десятичной точки)

Параметр p относится только к отображению значения  $A_x$  в тексте сообщения.

#### Описание функции

Функция считывает значение сигнала на аналоговом входе  $A_x$ .

Это значение умножается на значение параметра А (усиление). Параметр В (смещение) прибавляется к произведению, таким образом:  $(A_x \cdot \text{усиление}) + \text{смещение} = \text{текущее значение } A_x$ .

Функциональный блок определяет отношение значения  $A_x$  к диапазону. Блок устанавливает высокое значение цифрового выхода Q в течение доли времени, определяемой параметром РТ (интервал времени), и сбрасывает Q на оставшийся период времени.

### Примеры с временными диаграммами

Приведенные примеры (рис. 5.51, 5.52) поясняют порядок модулирования цифрового выходного сигнала входным аналоговым значением при помощи команды ШИМ:

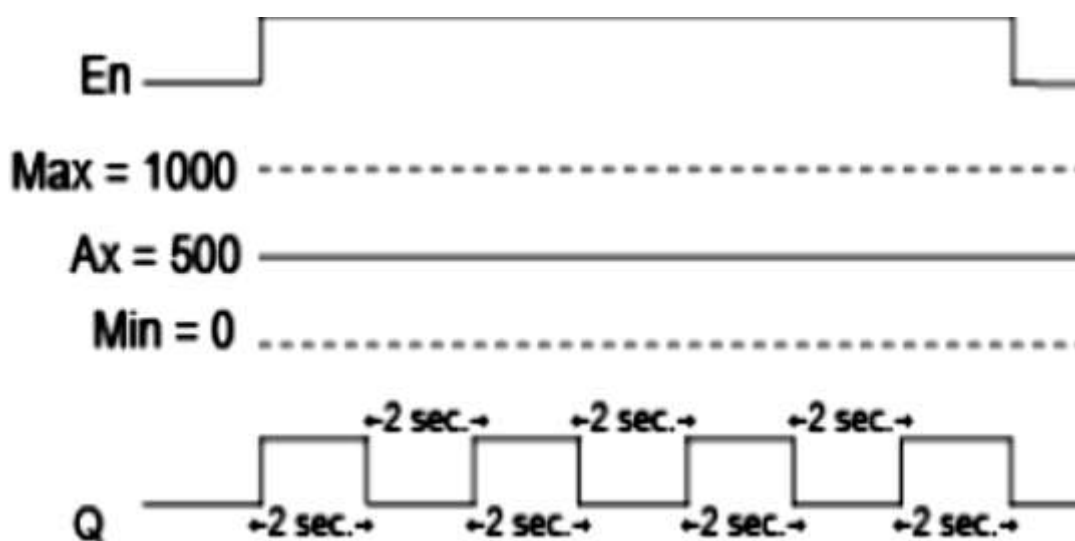


Рис. 5.51. Временная диаграмма

1. Аналоговое значение 500 (диапазон 0 – 1000) в качестве значения  $A_x$  должно быть преобразовано в последовательность цифровых сигналов. Задаваемый пользователем параметра РТ (интервал времени) равен 4 секундам. На цифровом выходе функции широтно-импульсного модулятора (PWM) цифровой сигнал имеет 2 секунды высокий уровень, 2 секунд низкий уровень, 2 секунды высокий уровень, 2 секунд низкий уровень, и это изменение продолжается до тех пор, пока параметр «En» имеет высокий уровень.

2. Аналоговое значение 300 (диапазон 0 – 1000) в качестве значения  $A_x$  должно быть преобразовано в последовательность цифровых сигналов. Задаваемый пользователем параметр РТ (интервал времени) равен 10 секундам. На цифровом выходе функции широтно-импульсного модулятора (ШИМ / PWM) цифровой сигнал имеет 3 секунды высокий уровень, 7 секунд низкий уровень, 3 секунды высокий уровень, 7 секунд низкий уровень, и это изменение продолжается до тех пор, пока параметр «En» имеет высокий уровень.

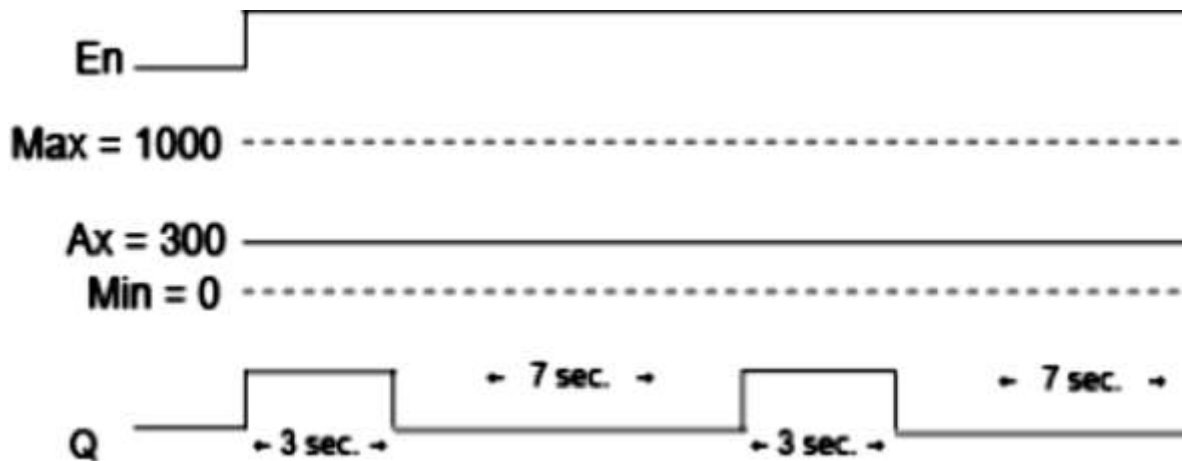


Рис. 5.52. Временная диаграмма задержки включения

### Правило расчета

$Q = 1$  для  $(Ax - Min) / (Max - Min)$  периода времени  $PT$

$Q = 0$  для  $PT - [(Ax - Min) / (Max - Min)]$  периода времени  $PT$ .

**Примечание.** В данной формуле  $Ax$  обозначает фактическое значение  $Ax$ , вычисленное с учетом усиления и смещения.  $Min$  и  $Max$  – минимальное и максимальные значения, установленные для диапазона.

#### 5.3.3.8. Блок аналоговых вычислений

Блок аналоговых вычислений (рис. 5.53) рассчитывает значение  $AQ$  по уравнению, сформированному из определенных пользователем операндов и операторов. Параметры блока приведены в табл. 5.24.

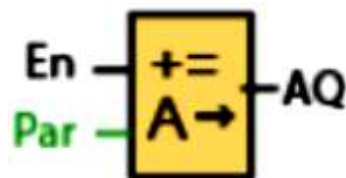


Рис. 5.53. Блок аналоговых вычислений

Таблица 5.24

Параметры функции аналоговых вычислений

Соединение	Описание
Вход <b>En</b>	Выберите блок аналоговых вычислений.
Параметр	<b>V1:</b> значение 1-первый операнд <b>V2:</b> значение 2-второй операнд <b>V3:</b> значение 3-третий операнд <b>V4:</b> значение 4-четвертый операнд <b>Оператор 1:</b> Первый оператор <b>Оператор 2:</b> Второй оператор <b>Оператор 3:</b> Третий оператор <b>Приоритет 1:</b> Приоритет первой операции

Соединение	Описание
Параметр	<b>Приоритет 2:</b> Приоритет второй операции <b>Приоритет 3:</b> Приоритет третьей операции <b>р:</b> Число разрядов после десятичной точки (0,1,2,3)
Выход <b>AQ</b>	Выход AQ является результатом вычисления по формуле, образованной значениями операндов и операторами. При делении на 0 или переполнении на выходе AQ будет установлено значение 32767, а при отрицательном переполнении будет установлено значение – 32768.

**0BA1-0BA5:** Блок аналоговых вычислений отсутствует в системах модулей версии до 0BA6.

### Параметры V1, V2, V3 и V4

Значения для параметров V1, V2, V3 и V4 могут быть получены как значения другой ранее запрограммированной, функции:

- аналоговый компаратор:  $Ax - Ay$ ;
- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

### Параметр р (число разрядов после десятичной точки)

Параметр р относится к отображению значений V1, V2, V3, V4 и AQ в текстах сообщений.

### Описание функции

Функция аналоговых вычислений объединяет в уравнение четыре операнда и три оператора. Оператором может быть любое из четырех стандартных действий: +, -, \* или /. Для каждого оператора необходимо установить неповторяющийся уровень приоритета: высокий («Н»), средний («М») или низкий («L»). Сначала будет выполнена операция с высоким приоритетом, затем – со средним, и затем – с низким. Необходимо наличие ровно одной операции каждого приоритета. В качестве значений операндов могут использоваться другие, ранее запрограммированные функции.

Число значений операндов установлено равным четырем, а число операторов – трем. Если требуется использовать меньшее число операндов, то используйте конструкции вида «+0» или «\*1», чтобы заполнить оставшиеся параметры.

Можно также настроить работу этой функции при значении параметра Enable «En» = 0. Функциональный блок может сохранять последнее значение или может быть сброшен в 0.

### Возможные ошибки: деление на ноль и переполнение

Если выполнение функционального блока приводит к делению на ноль или переполнению, то устанавливаются внутренние биты, указывающие тип произошедшей ошибки. Можно использовать в коммутационной программе функциональный блок обнаружения ошибок аналоговых вычислений, чтобы обнаруживать эти ошибки и соответствующим образом управлять выполнением программы. Один блок обнаружения ошибок аналоговых вычислений может использоваться с одним определенным блоком аналоговых вычислений.

### Примеры

В приведенных ниже табл. 5.25–5.27 показаны некоторые простые примеры параметров блока аналоговых вычислений, соответствующие им уравнения и выходные значения:

Таблица 5.25

$$\text{Уравнение: } (12 + (6 / 3)) - 1$$

V1	Оператор 1 (Приоритет 1)	V2	Оператор 2 (Приоритет 2)	V3	Оператор 3 (Приоритет 3)	V4
12	+(M)	6	/(H)	3	-(L)	1

**Результат: 13**

Таблица 5.26

$$\text{Уравнение: } 2 + (3 * (1 + 4))$$

V1	Оператор 1 (Приоритет 1)	V2	Оператор 2 (Приоритет 2)	V3	Оператор 3 (Приоритет 3)	V4
2	+(L)	3	*(M)	1	+(H)	4

**Результат: 17**

Таблица 5.27

$$\text{Уравнение: } (100 - 25) / (2 + 1)$$

V1	Оператор 1 (Приоритет 1)	V2	Оператор 2 (Приоритет 2)	V3	Оператор 3 (Приоритет 3)	V4
100	+(H)	25	/(L)	2	+(M)	1

**Результат: 25**

### **5.3.4. Обработка аналоговых сигналов**

#### **5.3.4.1. Алгоритм функционирования ПЛК с аналоговыми сигналами**

Аналоговый сигнал представляет собой физическую величину, которая может принимать любое значение в заданном диапазоне от электрического сигнала к аналоговому значению.

#### **Алгоритм обработки**

Для обработки системой LOGO! физических величин необходимо выполнить несколько шагов:

1. Модули LOGO! могут считывать электрические напряжения от 0 до 10 В или ток и от 0 до 20 мА на одном аналоговом входе. Поэтому физическая величина (например, температура, давление, частота вращения и т.п.) должна быть преобразована в необходимый вид электрического сигнала в выбранном диапазоне и иметь соответствующую величину. Это преобразование выполняется внешним датчиком.

2. Устройство LOGO! считывает электрическое значение и, в ходе последующего преобразования оно преобразуется в стандартизованное значение в диапазоне от 0 до 1000. Потом это значение используется в коммутационной программе на входе аналоговой специальной функции.

3. Для того чтобы приспособить стандартизованное значение к конкретной области применения, устройства LOGO! используют специальную аналоговую функцию, учитывающую усиление и смещение, для вычисления аналогового значения.

Аналоговое значение после этого оценивается специальной функцией (например, аналоговым усилителем). Если у специальной аналоговой функции имеется аналоговый выход, то аналоговое значение также подается на выход специальной функции.

4. В ПЛК можно преобразовывать аналоговые значения обратно в электрический сигнал. В этом случае напряжение может принимать значения в пределах от 0 до 10 В.

5. С помощью этого напряжения устройство LOGO! может управлять внешним исполнительным механизмом, преобразующим напряжение, а также аналоговое значение обратно в физическую величину.

Схема, показанная на рис. 5.54 иллюстрирует последовательность операций преобразования «физическая величина» (1), «электрическая величина» (2), «стандартизованное значение», «вычисление» (3), «электрическая величина» (4), преобразование сигнала исполнительным механизмом (5) в физическую величину.

#### **Усиление**

Стандартизованное значение умножается на параметр. Применение этого параметра позволяет в большей или меньшей степени

изменить электрическую величину. Соответственно этот параметр называется «усилением».

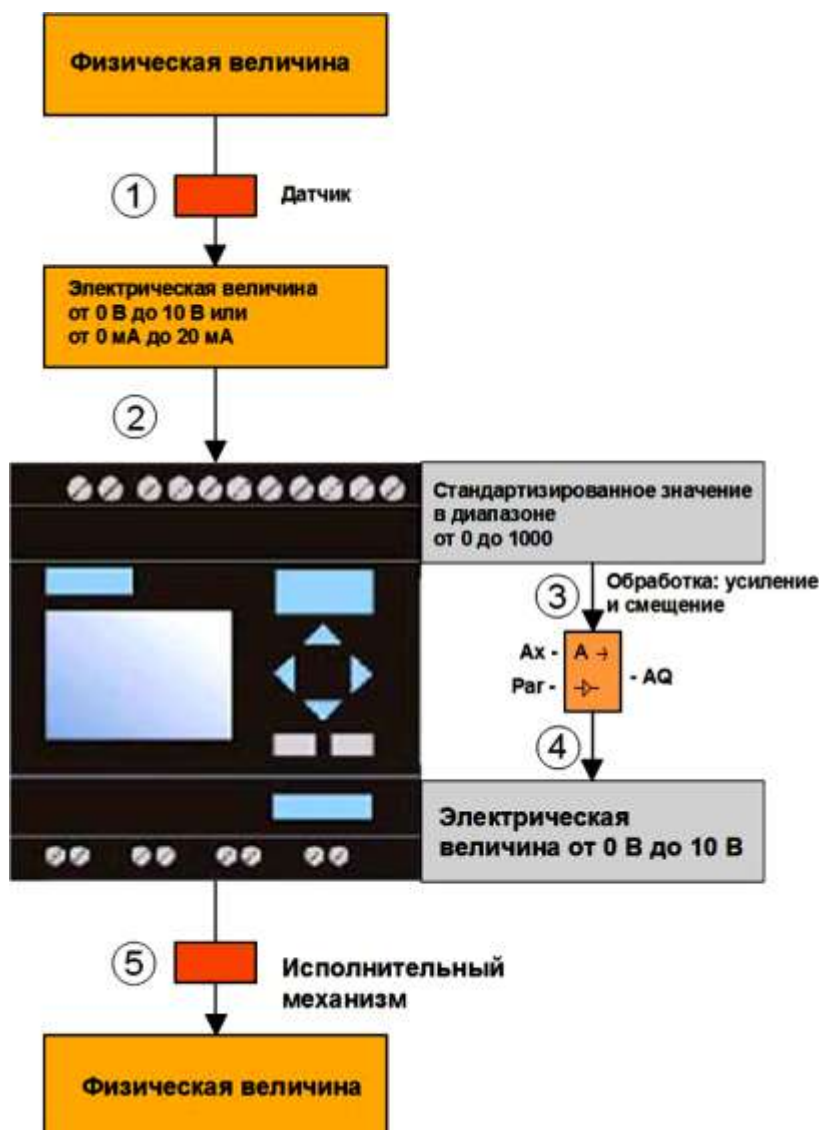


Рис. 5.54. Структура обработки аналоговых сигналов

### Смещение нулевой точки

Можно прибавить параметр к измененной стандартизированной величине или вычесть параметр из этой величины.

Применение этого параметра позволяет в большей или меньшей степени переместить нулевую точку электрической величины. Соответственно, этот параметр называется «смещением нулевой точки».

### Усиление и смещение

Следовательно, аналоговое значение вычисляется таким образом:

**Аналоговое значение** = (стандартизированное значение × усиление) + смещение. На схеме, приведенной на рис. 5.55 показана эта формула и значение усиления и смещения.





Рис. 5.55. Графическое отображение процесса усиления

Прямая линия на графике описывает то, какому стандартизованному значению соответствует при преобразовании аналоговое значение. Усиление соответствует наклону прямой линии, а смещение соответствует месту пересечения прямой линией оси Y при нулевом значении.

### Аналоговый выход

В случае подключения к специальной функции (имеющей аналоговый выход) к **реальному** аналоговому выходу, обратите внимание, что аналоговый выход может обрабатывать только значения от 0 до 1000.

#### 5.3.4.2. Возможные настройки конфигурации каналов ввода аналоговых сигналов

Задаётся тип датчика (0–10В; 0–20мА; 4–20мА; РТ100; без датчика). В случае датчика типа 4–20мА диапазон стандартизованных значений составляет 200 – 1000.

### Диапазон измерения

Меню задает диапазон измерения (рис. 5.56), который представляет собой диапазон значений, отображаемый для аналоговой величины.

Рис. 5.56. Задание диапазона

Затем LSC автоматически вычисляет по этим данным усиление и смещение.

## Усиление и смещение

Если необходимо задать усиление самостоятельно, то вводится значение от -10,00 до 10,00. Значение 0 не имеет смысла, поскольку вне зависимости от подаваемого аналогового значения вы всегда будете получать в качестве результата 0.

Если необходимо задать смещение самостоятельно, то вводится значение от -10 000 до +10 000.

## Ошибка округления

Программа разработки LSC вычисляет усиление и смещение нулевой точки с максимально возможной точностью. Тем не менее, вычисления в устройстве LOGO! осуществляются внутренними средствами с использованием целых чисел. Поэтому не все сочетания параметров допустимы для использования в устройствах LOGO!.

В этом случае перед передачей в LOGO! можно сообщить программе LSC об ошибке округления и предложить альтернативное значение диапазона.

## Эмуляция обработки аналоговых сигналов

Эмуляция позволяет считывать следующие значения, показанные на рис. 5.57:

- 1) физическая величина; диапазоны определяются диапазоном измерений (3);
- 2) стандартизованное значение;
- 3) диапазон измерения;
- 4) аналоговое значение (после обработки усиления и смещения);
- 5) аналоговое значение аналогового выхода.

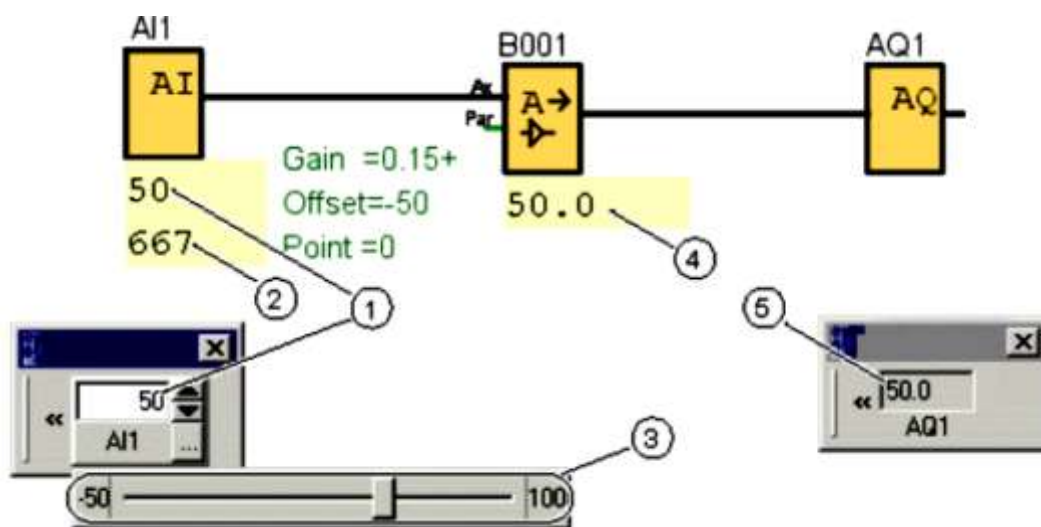


Рис. 5.57. Эмуляция обработки аналогового сигнала: 1 – физическая величина; 2 – масштабированное значение; 3 – диапазон измерений; 4 – аналоговое значение; 5 – аналоговое значение аналогового выхода

### 5.3.4.3. Возможные настройки при использовании LOGO!

В случае непосредственного программирования коммутационной программы в устройстве LOGO! можно только вводить значения усиления и смещения. Можно рассчитать усиление и смещение следующим образом:

#### Диапазон внешних значений $min_{\text{датчик}} - max_{\text{датчик}}$

Диапазон физических значений, которые могут быть измерены датчиком.

#### Стандартизированный диапазон значений $min_{\text{норм.}} - max_{\text{норм.}}$ : Диапазон значений для стандартизированных значений.

В случае датчиков, формирующих сигнал 0 – 10 В или 0 – 20 мА, диапазон стандартизированных значений равен 0 – 1000.

В случае датчиков, формирующих сигнал 4 – 20 мА, диапазон стандартизированных значений равен 200 – 1000.

Соответственно, усиление и смещение определяются так:

$$\text{Усиление} = (max_{\text{датчик}} - min_{\text{датчик}}) / (max_{\text{норм.}} - min_{\text{норм.}})$$

$$\text{Смещение} = [(min_{\text{датчик}} \times max_{\text{норм.}}) - (max_{\text{датчик}} \times min_{\text{норм.}})] / (max_{\text{норм.}} - min_{\text{норм.}})$$

В случае вычисления усиления или смещения в соответствии с приведенными выше формулами, вы можете затем вычислить другое из значений по следующей формуле:

$$\text{Усиление} = (min_{\text{датчик}} - \text{Смещение}) / min_{\text{норм.}}$$

$$\text{Смещение} = (min_{\text{датчик}} - (\text{Усиление} \times min_{\text{норм.}}))$$

### 5.3.4.4. Пример

#### Предварительные условия

Датчик: датчик температуры, диапазон измерения от –50 до 100°C. Измеряемая температура 25°C.

#### Порядок действий

1. Датчик преобразует температуру из 25°C в напряжение 5,0 В.
2. LOGO! преобразует 5,0 В в стандартизированное значение 500.
3. Используя данные датчика и диапазона измерения, LOGO! устанавливает значение 0,15 для усиления и значение –50 для смещения.

В соответствии с формулой:

**Аналоговое значение** = (стандартизированное значение x усиление) + смещение.

LOGO! вычисляет аналоговое значение:

$$\text{Аналоговое значение} = (500 \times 0,15) - 50 = 25.$$

Варианты расчёта показаны в табл. 5.28.

Таблица 5.28

## Дополнительные примеры расчёта

Физ. значение	Электр. количественное значение, выдаваемое датчиком	Стандартизованное значение	Кт. Усил.	Смещение	Аналог. значение
	0 В 5 В 10 В	0 500 1000	0.01	0	0 5 10
	4 мА 12 мА 20 мА	0 500 1000	10	0	0 5000 10000
	0 мА 10 мА 20 мА	0 500 1000	1	50	50 550 1050
1000 мбар 3700 мбар 5000 мбар	0 В 6.75 В 10 В	0 675 1000	4	1000	1000 3700 5000
-30°C 0°C 70°C	0 мА 6 мА 20 мА	0 300 1000	0.1	-30	-30 0 70

**Вопросы для самоконтроля**

1. Как осуществляется гистерезис с помощью аналогового порогового выключателя?
2. Что такое дифференциальный аналоговый выключатель?
3. Что такое аналоговый компаратор?
4. Что такое аналоговый мультиплексор?
5. Что такое ШИМ?
6. Из каких этапов состоит преобразование электрического сигнала с датчика в сигнал, подаваемый на исполнительный механизм?

**5.3.5. Управление и регулирование****5.3.5.1. Основы управления и регулирования**

В инженерной практике количественные значения допускают управление и регулирование. При управлении осуществляется изменение количества без возможности компенсации внешних воздействий. При выполнении регулирования количество поддерживается равным заданному значению с целью компенсации внешних воздействий.

В приведенном на рис. 5.58 примере «управление» означает, что человек может установить фиксированное значение тепловой мощно-

сти. Нагреватель не может компенсировать падение температуры в помещении в случае открытия окна.

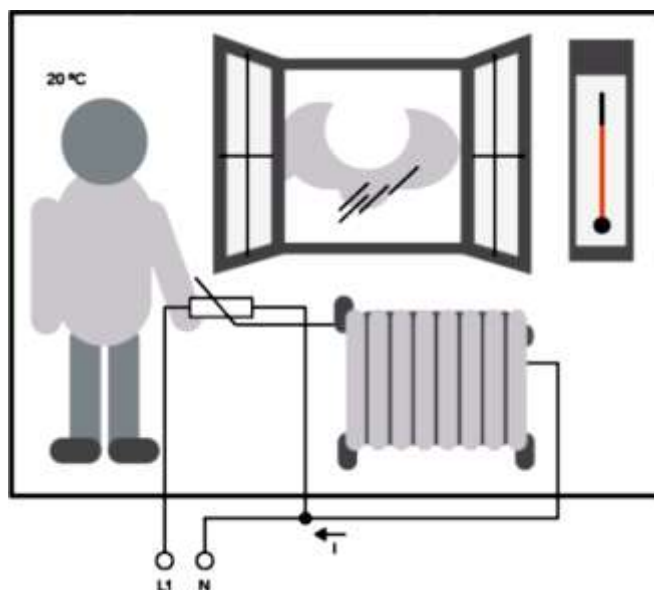


Рис. 5.58. Пример системы управления

В приведенном примере на рис. 5.59 «регулирование» означает, что оператор может увеличить значение тепловой мощности, если температура в помещении опускается ниже 20°C. Если температура в помещении становится выше 20°C, то тепловая мощность снижается.

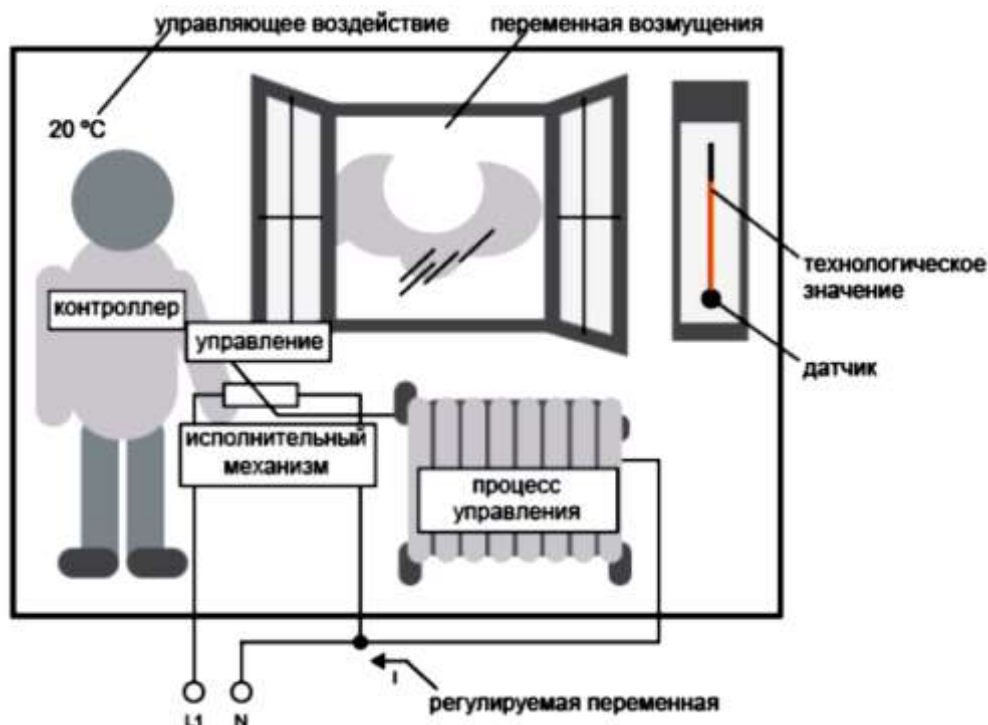


Рис. 5.59. Функциональная схема регулирования температуры электронагревателем в помещении

## Основные понятия регулирования

В приведенном примере ток электронагревателя является **регулируемой** переменной. Переменный резистор является **исполнительным механизмом**.

Рука, управляющая исполнительным механизмом (ИМ), называется **органом управления**. Фактическая температура в помещении является управляемой переменной или **значением технологического процесса**. Требуемая температура в помещении является **заданным значением переменной** или **уставкой**. Электрическая система отопления называется **управляющим процессом**, термометр называется **датчиком**, падение температуры при открытии окна называется **переменной возмущения**.

Таким образом, человек измеряя значение технологического процесса (температуру в помещении) датчиком (термометром), сравнивает значение технологического процесса (температуру в помещении) с заданным значением переменной (требуемой температурой в помещении) и использует исполнительный механизм (переменный резистор) для ручного регулирования регулируемой переменной (тока отопления) с целью компенсации переменной возмущения (падения температуры при открытии окна). Человек в этой системе играет роль **регулятора**.

**Устройство управления** состоит из исполнительного механизма и органа управления. Орган управления совместно с регулятором образуют **регулирующее устройство**. На приведенном ниже рис. 5.60 содержится абстрактное отображение описанной выше ситуации.

Элемент сравнения использует датчик для сравнения заданного значения переменной со значением технологического процесса. При выявлении отклонения заданного значения переменной от значения технологического процесса в контуре формируется положительное или отрицательное значение ошибки, которое в свою очередь изменяет значение технологического процесса.

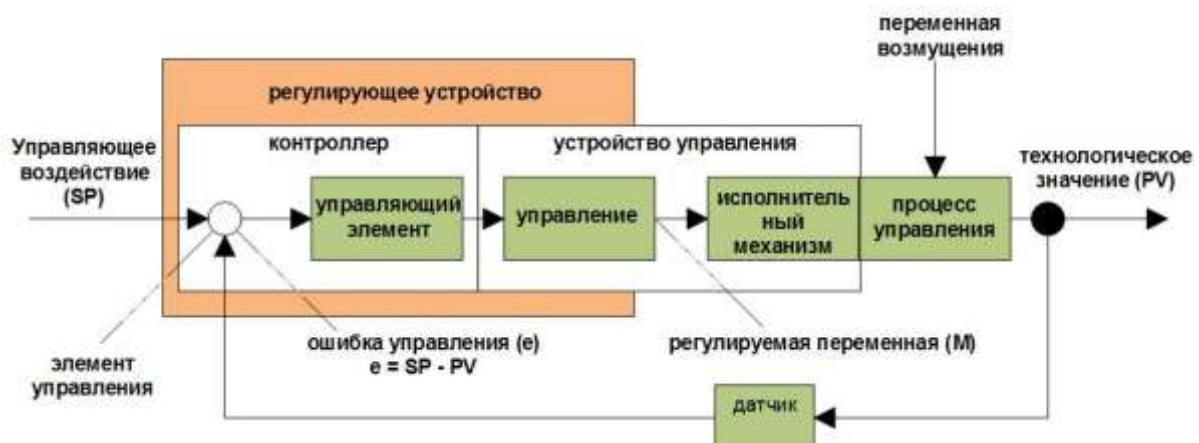


Рис. 5.60. Система автоматического регулирования температуры

### Контур регулирования

Значение технологического процесса  $x$  оказывает влияние на регулируемую переменную  $M$  при помощи устройства регулирования. При этом образуется замкнутый контур, известный под названием **контур регулирования**.

Если в рассмотренном выше примере открывается окно, то температура в помещении уменьшается. Человек должен увеличить тепловую мощность отопителя. В случае чрезмерного увеличения тепловой мощности температура становится слишком высокой. В этом случае человеку необходимо уменьшить тепловую мощность.

В случае слишком быстрого увеличения или уменьшения тепловой мощности в контуре регулирования начинаются колебания. Температура в помещении начинает колебаться: становится то слишком тепло, то слишком холодно. Для предотвращения этого оператор должен осторожно и медленно уменьшать или увеличивать тепловую мощность.

### Ошибка контура

Ошибка контура ( $e$ ) представляет собой разность заданного значения переменной ( $SP$ ) и значения технологического процесса ( $PV$ ). Иными словами, отклонение значения технологического процесса от установленного значения:

$$e = SP - PV.$$

Ошибка контура  $e$  вызывает изменение регулируемой переменной  $M$ .

Это можно прекрасно проиллюстрировать на приведенном выше примере: если при требуемом значении температуры  $20^{\circ}\text{C}$  (= заданное значение  $w$ ) температура в помещении равна  $22^{\circ}\text{C}$  (= технологическое значение  $PV$ ), то этому случаю соответствует ошибка контура:

$$e = SP - PV = 20^{\circ}\text{C} - 22^{\circ}\text{C} = -2^{\circ}\text{C}.$$

В этом случае отрицательный знак указывает на обратное действие: тепловая мощность уменьшается.

Если контур регулирования находится в состоянии равновесия, то ошибка контура является нулевой или очень низкой. При изменении управляемой переменной или при возмущении ошибка контура увеличивается. Ошибка контура корректируется изменением регулируемой переменной  $M$ .

#### 5.3.5.2. Основы работы регулятора

Регулятор может быть упрощенно изображен следующим образом, показан на рис. 5.61.

Элемент сравнения и функция регулятора описывают поведение регулятора.

Далее описываются наиболее важные типы регуляторов. Переходная характеристика регулятора в значительной степени определяет его поведение.

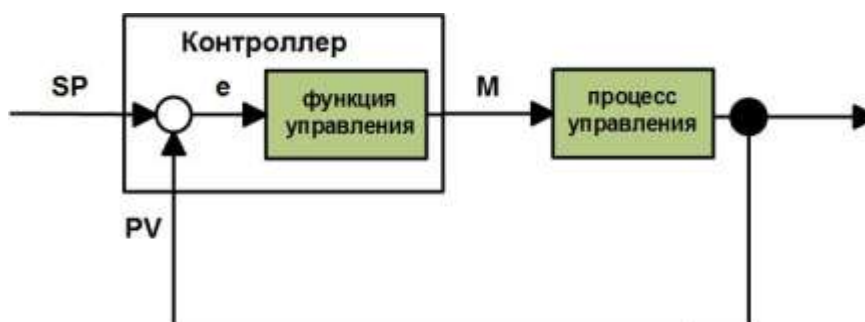


Рис. 5.61. Регулятор процесса управления

Переходная характеристика описывает то, как регулятор реагирует на хаотическое изменение значения технологической переменной.

Существует 3 основных типа регуляторов:

- регулятор пропорционального действия (П-регулятор);
- регулятор интегрального действия (И-регулятор);
- регулятор дифференциального действия (Д-регулятор – он здесь не рассматривается).

Они комбинируются и образуют реальный регулятор. В частности, ПИ-регулятор.

### П-регулятор

Регулятор пропорционального действия (П-регулятор) изменяет регулируемую переменную  $M$  пропорционально ошибке контура. Регулятор П-типа срабатывает немедленно. Сам по себе он не может сделать ошибку контура равной нулю.

$$M_{Pn} = k_P e_n,$$

$M_{Pn}$ : регулируемая переменная П-регулятора в момент времени  $n$ ;  $k_P$ : усиление П-регулятора;  $e_n$ : ошибка контура в момент времени  $n$ .

На рис. 5.62 показаны резкое изменение значения технологического процесса и переходная характеристика регулятора.

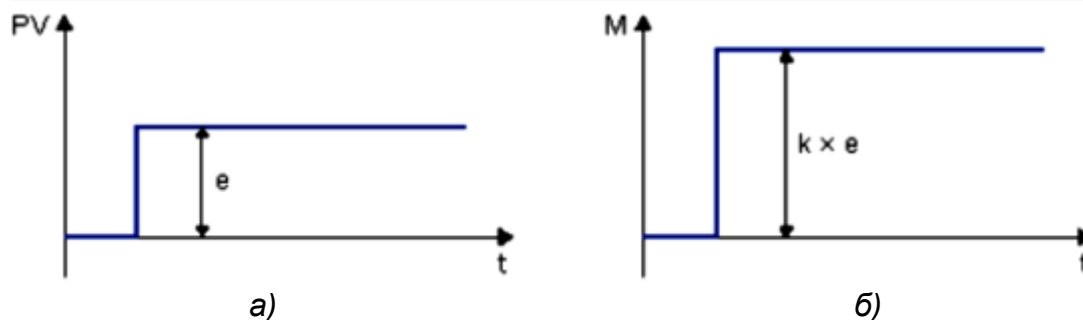


Рис. 5.62. Параметры регулирования: а – изменение величины сигнала процесса, б – изменение величины переходного процесса регулируемой переменной



## Резюме

П-регулятор обладает следующими характеристиками:

- не может корректировать ошибки в процессе управления> остаточная ошибка контура;
- немедленно реагирует на изменение значения технологического процесса;
- является стабильным.

## И-регулятор

Регулятор интегрального действия (И-регулятор) изменяет регулируемую переменную  $M$  пропорционально ошибке контура и времени. И-регулятор работает с задержкой. Он полностью устраняет ошибку контура.

Для того чтобы рассчитать регулируемую переменную для значения времени  $n$ , время до этого периода должно быть разделено на небольшие отрезки. Ошибки контура на конец каждого отрезка времени должны быть просуммированы (проинтегрированы) и включены в расчет:

$M_{In} = k_I \times (T_S / T_I) \times (e_n + e_{n-1} + e_{n-2} + e_{n-3} + \dots + e_0) = k_I \times (T_S / T_I) \times e_n + M_{In-1}$ ,  
 где  $M_{In}$ : регулируемая переменная И-регулятора в момент времени  $n$ ;  
 $M_{In-1}$ : регулируемая переменная И-регулятора в момент времени  $n-1$ ;  
 так же называемая интегральной суммой;  $k_I$ : усиление И-регулятора;  
 $T_S$ : время выборки, продолжительность отрезка времени;  $T_I$ : интеграл по времени: при помощи этого времени, известного также как время интегрального воздействия, производится воздействие интегральной части на регулируемую переменную;  $e_n$ : Ошибка контура в момент времени  $n$ ;  $e_{n-1}$ : Ошибка контура в момент времени  $n-1$ ;  $e_0$ : Ошибка контура в момент начала расчетов.

На рис. 5.63 показано резкое изменение значения технологического процесса и переходная характеристика регулятора.

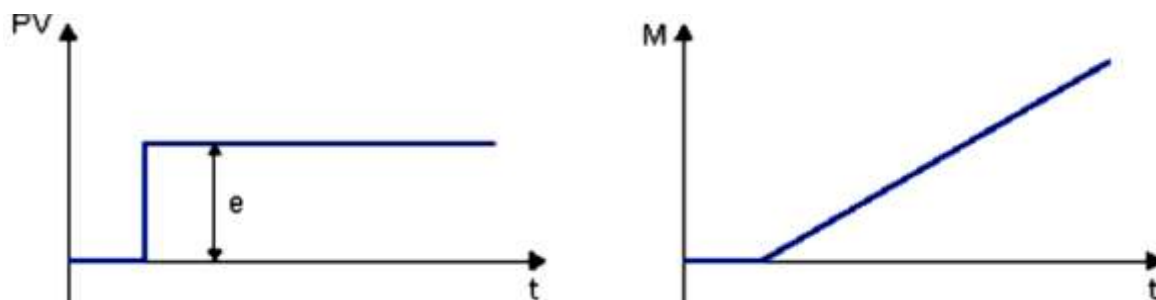


Рис. 5.63. Временная диаграмма задержки включения

## Резюме

И-регулятор обладает следующими характеристиками:

- устанавливает значение технологического процесса в точности равным заданному значению переменной;

- при этом он имеет тенденцию к колебаниям и является нестабильным;
- требует больше времени для выполнения функции управления по сравнению с П-регулятором.

### ПИ-регулятор

ПИ-регулятор немедленно сокращает ошибку контура и способен свести ошибку контура до нуля.

$$M_n = M_{Pn} + M_{In} = k_P \times e_n + k_I \times (T^S / T_I) \times e_n + M_{In-1},$$

$M_n$ : регулируемая переменная в момент времени  $n$ ;  $M_{Pn}$ : пропорциональная часть регулируемой переменной;  $M_{In}$ : интегральная часть регулируемой переменной;  $M_{In-1}$ : регулируемая переменная И-регулятора в момент времени  $n-1$  так же называемая интегральной суммой;  $k_P$ : усиление П-регулятора;  $k_I$ : усиление И-регулятора;  $T^S$ : время выборки, продолжительность отрезка времени;  $T_I$ : интегральное время; при помощи этого времени, известного также как время интегрального воздействия, производится воздействие интегральной части на регулируемую переменную;  $e_n$ : ошибка контура в момент времени  $n$ .

На рис. 5.64 показано скачкообразное изменение значения технологического процесса и переходная характеристика регулятора.

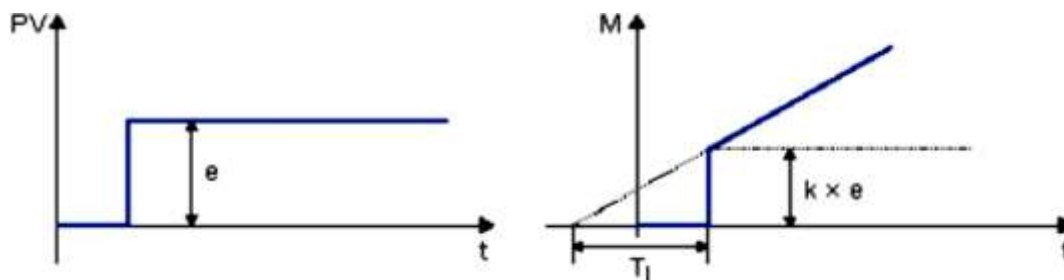


Рис. 5.64. Параметры регулирования: а – изменение величины сигнала процесса, б – изменение величины переходного процесса регулируемой переменной

### Резюме

ПИ-регулятор обладает следующими характеристиками (табл. 5.29):

- элементы П-регулятора быстро обнаруживают возникающую ошибку контура;
- элементы И-регулятора могут затем устранить остаток ошибки контура;
- элементы регулятора дополняют друг друга и поэтому ПИ-регулятор действует быстро и точно.

#### 5.3.5.3. Описание отдельных параметров

Параметр Dir задает направление действия регулятора.

Положительное значение означает:

- если уставка > значения технологической переменной, то значение технологической переменной увеличивается;
- если уставка < значения технологической переменной, то значение технологической переменной уменьшается.

Отрицательное значение означает:

- если уставка > значения технологической переменной, то значение технологической переменной уменьшается;
- если уставка < значения технологической переменной, то значение технологической переменной увеличивается.

Таблица 5.29

### Параметры ПИ-регулятора

Параметры регулятора	Реализовано с помощью системы LOGO!	Возможный диапазон значений
1	2	3
<b>Mn</b> Регулируемая переменная в момент времени n	Выход блока ПИ-регулятора	от 0 до 1000
<b>KC</b> <b>kP</b> Усиление П-части <b>kI</b> Усиление И-части	Параметр KC применяется для равного увеличения составляющих I и P регулятора. В случае ввода значения KC=0 составляющая P регулятора выключается. В этом специальном случае k для части И автоматически устанавливается в 1. Если KC = 0: kP = 0 и kI = 1 Если KC <> 0: kP = kI = KC	от 0,00 до 99,99
<b>TS</b> Время выборки	Фиксированное	500мс
<b>TI</b> Интегральное время	Параметр TI, в случае установки этого параметра равным 99:59 мин вы выключаете часть И – регулятора.	от 00:01 мин до 99:59 мин
<b>en</b> Ошибка контура в момент времени n;	См. SP – PV Обычно применяется: $e = SP - PV$	
<b>SP</b>	Параметр SP является заданным значением w.	от -10,000 до +20,000
<b>PV</b>	PV представляет собой переменную технологического процесса x: $PV = (\text{аналоговое значение входа} * \text{усиление}) + \text{смещение}$	
	Параметр усиления оказывает влияние на PV	от 0,0 до 10,0
	Параметр смещения оказывает влияние на PV	от -10,000 до +20,000
	PV ограничивается при помощи параметров Мин. и Макс.	-10,000 – +20,000

### Пример, регулирование системы отопления

Если уставка превышает значение технологической переменной (в помещении слишком холодно), регулируемая переменная увеличивает значение технологической переменной.

#### 5.3.5.4. ПИ-регулятор

ПИ-регулятор представляет собой регулятор пропорционального и интегрального действия (рис. 5.65). Можно использовать как пропорциональное, так и интегральное действие как по отдельности, так и вместе. Параметры ПИ-регулятора приведены в табл. 5.30.

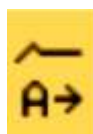


Рис. 5.65. Функциональный блок ПИ-регулятора

Таблица 5.30

Параметры функции ПИ-регулятора

Соединение	Описание
Вход <b>A/M</b>	Установка режима регулятора: 1: автоматический режим 0: ручной режим
Вход <b>R</b>	Вход R используется для сброса выхода AQ. Пока этот вход установлен, вход A/M отключен. Выход AQ установлен равным 0.
Вход <b>PV</b>	Аналоговое значение: параметр технологического процесса, определяет выходное значение.
Параметр	Датчик: Тип используемого датчика <b>min.:</b> Минимальное значение для диапазона значений PV: от -10,000 до +20,000 <b>max.:</b> Максимальное значение для диапазона значений PV: от -10,000 до +20,000 <b>A:</b> Усиление. Диапазон значений: +-10.00 <b>B:</b> Смещение. Диапазон значений: +-10,000 <b>SP:</b> Диапазон значений для назначения уставки: от -10,000 до +20,000 <b>Mq:</b> Значение для AQ в ручном режиме. Диапазон значений: от 0 до 1000 <b>Наборы параметров:</b> связанные с конкретным приложением предустановки параметров KC, TI и Dir (см. ниже) <b>KC:</b> Диапазон значений усиления: от 00.00 до 99.99 <b>TI:</b> Диапазон значений интегрального времени от 00:01 мин до 99:59 мин <b>Dir:</b> Диапазон значений направления действия регулятора: + или – <b>p:</b> Диапазон значений числа десятичных знаков: 0, 1, 2, 3
Выход <b>AQ</b>	Аналоговый выход (регулируемая переменная) Диапазон значений AQ: от 0 до 1000

## Параметры SP и Mq

Заданные значения SP и Mq могут быть получены как значения другой, ранее запрограммированной функции:

- аналоговый компаратор: Ax – Ay;
- аналоговый выключатель: Ax;
- аналоговый усилитель: Ax;
- аналоговый мультиплексор: AQ;
- линейно нарастающий аналоговый сигнал: AQ;
- блок аналоговых вычислений: AQ;
- ПИ-регулятор: AQ;
- реверсивный счетчик: Cnt.

### Параметр p (число разрядов после десятичной точки)

Параметр p относится только к отображению значений PV, SP, Min. и Max. в тексте сообщения.

## Временная диаграмма

Характер, свойства и скорость изменения AQ зависят от параметров KC и TI. Представленный на рис. 5.66 ход изменения AQ является только примером. Регулирующее действие является непрерывным, поэтому на диаграмме представлена лишь часть процесса:

1. Возмущение вызывает снижение PV, и поскольку Dir направлено вверх, то AQ увеличивается до тех пор, пока PV снова не будет соответствовать SP.

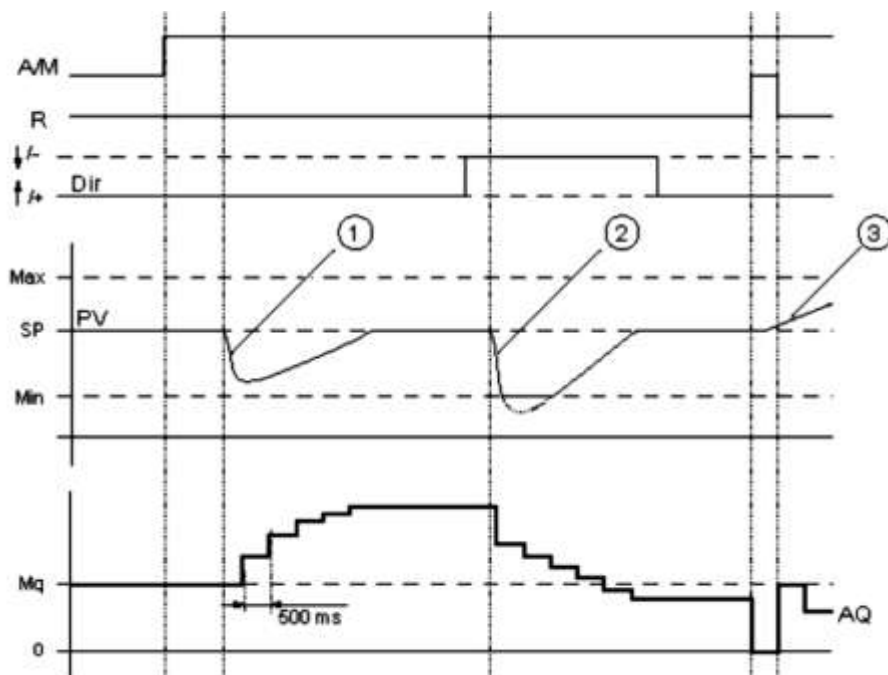


Рис. 5.66. Временная диаграмма изменения выходного значения AQ ПИ-регулятора

2. Возмущение вызывает снижение PV, и поскольку Dir направлено вверх, то AQ уменьшается до тех пор, пока PV снова не будет соответствовать SP. Значение Dir выбирается в соответствии с общим поведением контура регулирования. Направление (Dir) не может быть изменено во время работы функции. Изменение Dir показано здесь для наглядности.

3. Когда AQ сбрасывается в 0 при помощи входа R, то PV изменяется. Это связано с тем, что PV увеличивается, что, в свою очередь, вызывает уменьшение AQ, когда Dir направлено вверх.

При просмотре программы в режиме оперативного тестирования или эмуляции, LSC отображает тренд аналогового выходного значения ПИ-контроллера. Тренд демонстрирует изменение аналогового выходного значения AQ (рис. 5.66) с течением времени.

### Описание функции

Если на входе A/M устанавливается 0, то специальная функция выдает на выходе AQ значение, заданное параметром Mq.

Если на входе A/M устанавливается 1, то включается автоматический режим. В качестве интегральной суммы используется значение Mq, функция регулятора начинает расчет по формулам, заданным в п.п. 5.3.5.1 «Основы управления и регулирования».

В формулах используется обновленное значение PV:

**Обновленное значение PV = (PV \* усиление) + смещение**

Если обновленное значение PV = SP, то специальная функция не изменяет значения AQ.

Dir = вверх/+ (номера 1 и 3 на временной диаграмме):

- Если обновленное значение PV > SP, то специальная функция уменьшает значение AQ.
- Если обновленное значение PV < SP, то специальная функция увеличивает значение AQ.

Dir = вниз/+ (номер 2 на временной диаграмме):

- Если обновленное значение PV > SP, то специальная функция увеличивает значение AQ.
- Если обновленное значение PV < SP, то специальная функция уменьшает значение AQ.

При наличии возмущения AQ увеличивается или уменьшается до тех пор, пока обновленное значение PV не станет снова соответствовать SP. Скорость изменения AQ определяется параметрами KC и TI.

Если входное значение PV превышает значение параметра Max, то обновленное значение PV устанавливается равным значению Max.

Если PV становится меньше значения параметра Min, то обновленное значение PV устанавливается равным значению Min.

Если на входе R устанавливается 1, то выход AQ сбрасывается. Пока вход R установлен, вход A/M отключен.

### Интервал дискретизации

Установлен фиксированный интервал дискретизации, равный 500мс.

### Наборы параметров

Для упрощения использования ПИ-регулятора в нем запрограммированы наборы коэффициентов параметров КС, ТI и Dir для некоторых областей применения (табл. 5.31).

Таблица 5.31

Коэффициенты параметров по областям применения ПИ-регулятора

Набор параметров	Пример применения	КС	TI (с)	Dir
Температура – быстро	Температура, управление охлаждением небольших пространств; небольших объемов	0.5	30	+
Температура – медленно	Отопление, вентиляция, температура, управление охлаждением больших пространств; больших объемов	1.0	120	+
Давление 1	Быстрое изменение давления управление компрессором	3.0	5	+
Давление 2	Медленное изменение давления дифференциальное управление давлением (регулятор расхода)	1.2	12	+
Заполнение до уровня 1	Заполнение бочки и/или бака без слива	1.0	99:59	+
Заполнение до уровня 2	Заполнение бочки и/или бака со сливом	0.7	20	+

#### 5.3.5.5. Регулирование линейно нарастающего сигнала

Функция линейно нарастающего аналогового сигнала (рис.5.67) позволяет изменять выходное значение от текущего уровня **Level 1** до выбранного уровня **Level 2** с заданной скоростью.

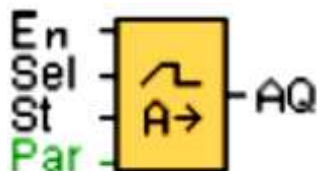


Рис. 5.67. Функциональный блок регулирования линейно нарастающего аналогового сигнала

**Параметры Level 1 и Level 2.** Значение параметров уровня Level 1 и Level 2 может быть получено как значение другой ранее запрограммированной функции. Параметры функции приведены в табл. 5.32, рис. 5.68.

Таблица 5.32

**Параметры регулирования линейно нарастающего сигнала**

Соединение	Описание
Вход <b>En</b>	При изменении состояния входа En (Enable =включение) с 0 на 1 на выход в течение 100 мс подается уровень пуска / останова (смещение «В» +StSp) и запускается линейное изменение сигнала до выбранного уровня. Изменение состояния с 1 до 0 устанавливает текущий уровень равным Смещению «В», в результате чего на выходе AQ устанавливается 0.
Вход <b>Sel</b>	Sel = 0: Выбирается шаг 1 (уровень 1). Sel = 1: Выбирается шаг 2 (уровень 2). Изменение состояния входа Sel запускает изменение текущего уровня к выбранному уровню с заданной скоростью.
Вход <b>St</b>	Изменение состояния с 0 до 1 на входе St (замедленный останов) вызывает уменьшение текущего уровня с постоянной скоростью до достижения уровня пуска / останова (смещение «В» + StSp). Уровень пуска / останова поддерживается 100мс, после чего текущий уровень устанавливается равным смещению «В», в результате чего на выходе AQ устанавливается 0.
Параметр	<b>Level 1 и Level 2:</b> Уровни, которые должны быть достигнуты; диапазоны значений: от -10,000 до +20,000 <b>Max L:</b> Максимальное значение, которое не должно превышать. Диапазон значений: от -10,000 до +20,000 <b>StSp:</b> смещение пуска/останова: значение, добавляемое к смещению «В» для создания уровня пуска/останова. Если смещение пуска/останова равно 0, то уровень пуска/останов равен смещению «В». Диапазон значений: от 0 до +20,000 <b>Rate:</b> скорость достижения значений параметров level 1, level 2 или Смещение. Задается число шагов в секунду. Диапазон значений: от 1 до 10,000 <b>A:</b> Диапазон значений усиления: от 0 до 10,00 <b>B:</b> Диапазон значений смещения: +- 10.000 <b>p:</b> Диапазон значений числа десятичных знаков: 0, 1, 2, 3
Выход <b>AQ</b>	Масштабирование выхода AQ: <b>(Текущий уровень – Смещение «В») / Усиление «А»</b>

**Параметр p  
(число разрядов после десятичной точки)**

Параметр p относится только к отображению значений AQ, Level 1, Level 2, Max L, StSp, и Rate в тексте сообщения.



### Временная диаграмма для AQ

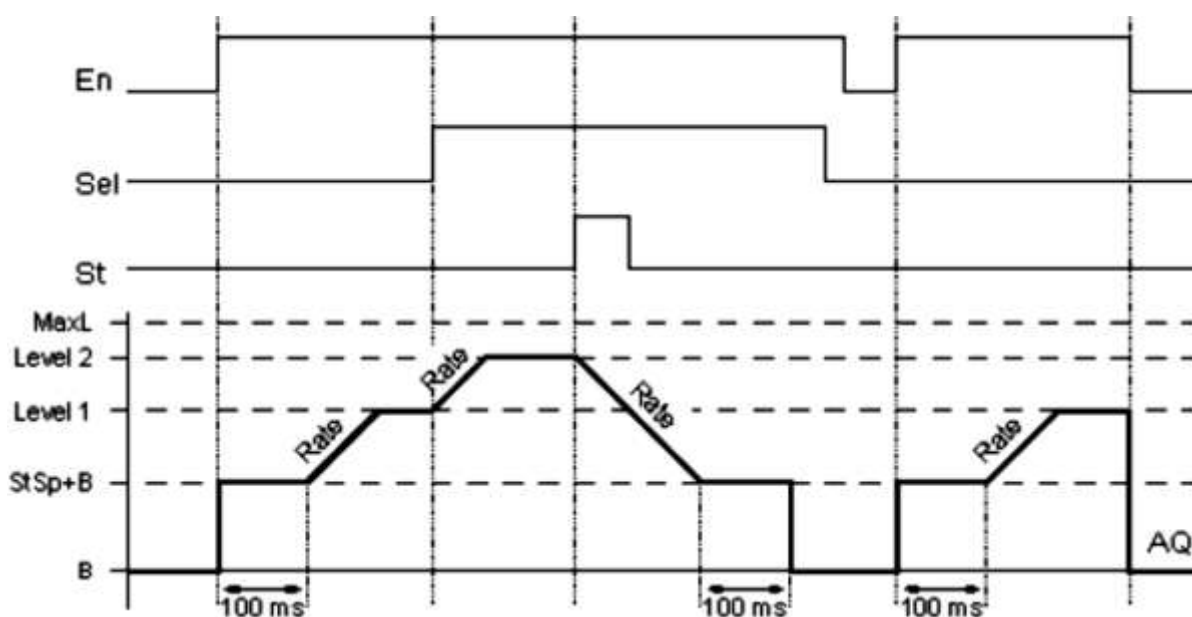


Рис. 5.68. Временная диаграмма регулирования линейно нарастающего сигнала

### Описание функции

Если вход En установлен, функция устанавливает значение StSp + Смещение «В» равным 100 мс (см. рис. 5.68).

После этого, в зависимости от подключения Sel, функция изменяется от уровня StSp + смещение «В» до уровня 1 или до уровня 2 со скоростью, установленной параметром Rate.

В случае установки входа St функция изменяется до уровня StSp + В с ускорением, заданным параметром Rate. После этого значение функции сохраняется равным StSp + смещение «В» в течение 100 мс. Через 100 мс уровень устанавливается равным смещению «В». При этом масштабированное значение (выход AQ) равно 0.

В случае установки входа St перезапуск функции возможен только после сброса входов St и En.

При изменении состояния входа Sel, в зависимости от подключения Sel, функция изменяется от текущего заданного уровня до нового заданного уровня с указанной скоростью.

При сбросе входа En функция немедленно устанавливает текущий уровень равным смещению «В».

Текущий уровень обновляется через каждые 100 мс. Обратите внимание на соотношение между состоянием выхода AQ и текущим уровнем:

$$\text{Выход AQ} = (\text{текущий уровень} - \text{смещение «В»}) / \text{усиление «А»}$$

### 5.3.6. Специальные функции

#### 5.3.6.1. Реле с блокировкой (RS триггер)

Сигнал на входе S устанавливает выход Q. Сигнал на входе R сбрасывает выход Q (рис. 5.69, 5.70, табл. 5.34).



Рис. 5.69. Функциональный блок RS триггера

Таблица 5.34

Параметры блока функции RS триггера

Соединение	Описание
Вход <b>S</b>	Устанавливает выход Q по сигналу на входе S (установка).
Вход <b>R</b>	Сбрасывает выход Q по сигналу на входе R (сброс). Выход Q сбрасывается, если установлены оба входа S и R (сброс обладает приоритетом над установкой).
Параметр	<b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.
Выход <b>Q</b>	Q устанавливается сигналом на входе S и продолжает быть установленным до сброса сигналом на входе R.

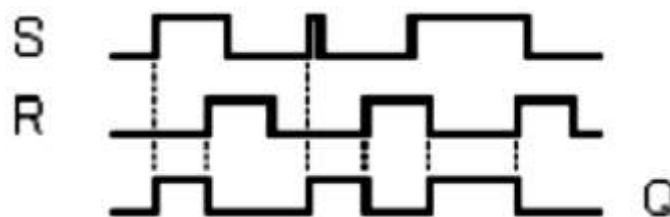


Рис. 5.70. Временная диаграмма функции RS-триггера

#### Описание функции

Реле с блокировкой представляет собой запоминающее устройство- RS-триггер таблица истинности приведена в табл. 5.35. Выходное значение зависит от входных состояний и от предыдущего статуса выхода.

Таблица 5.35

Таблица истинности RS-триггера

S	R	Q	Примечание
0	0	x	Состояние без изменения
0	1	0	Сброс
1	0	1	Установить
1	1	0	Сброс

При задействованном режиме сохранения выходной сигнал соответствует состоянию сигнала перед аварией питания.

### 5.3.6.2. Импульсное реле

Выход устанавливается и сбрасывается коротким однократным срабатыванием на входе Trg (рис. 5.71).

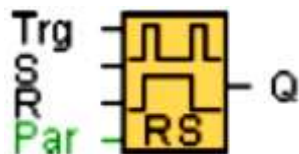


Рис. 5.71. Блок функции импульсного реле (RS триггер)

Следует отметить, если Trg=0 и Par=RS, то специальный функциональный блок «Импульсное реле» соответствует специальному функциональному блоку «Реле с блокировкой». Реализует функцию RS триггера приведенные в табл. 5.35 и 5.36.

Таблица 5.36

Импульсное реле с функциями RS триггера

Соединение	Описание
Вход <b>Trg</b>	Вы переключаете выход Q в состояние вкл. и выкл. сигналом на входе Trg (выключатель).
Вход <b>S</b>	Короткий импульс на входе S (установка) устанавливает на выходе логическую 1.
Вход <b>R</b>	Короткий импульс на входе R (сброс) устанавливает на выходе логический 0.
Параметр	<b>Выбор:</b> RS (приоритет входа R) или SR (приоритет входа S) <b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.
Выход <b>Q</b>	Q включается сигналом на входе Trg и снова сбрасывается следующим импульсом Trg, если оба сигнала S и R = 0.

#### 0ВА0-0ВА3:

Временная диаграмма функции импульсного реле показана на рис. 5.72. Специальная функция не использует вход S для предварительного выбора приоритета.

Выходной сигнал Q включается сигналом на Trg и снова выключается следующим сигналом на Trg или R.

#### Описание функции

Статус выхода Q изменяется при каждом переходе из 0 в 1 на входе Trg, и, если оба значения S и R = 0, то происходит включение и выключение выхода.

Вход Trg не оказывает влияния на специальные функциональные блоки, если  $S=1$  или  $R=1$ .

Короткий импульс на входе S устанавливает импульсное реле таким образом, что на выходе устанавливается логическая 1.

Короткий импульс на входе R сбрасывает импульсное реле в его исходное состояние, таким образом, на выходе он сбрасывается в состояние логического 0.

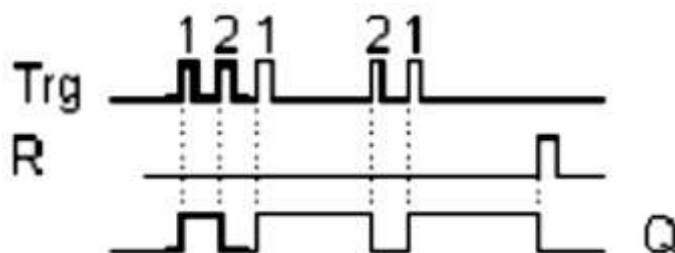


Рис. 5.72. Временная диаграмма включения-выключения выхода Q функции импульсного реле

В зависимости от выбранной вами конфигурации либо вход R обладает приоритетом по сравнению со входом S (сигнал на входе S не оказывает никакого действия, пока  $R=1$ ), либо вход S обладает приоритетом по сравнению со входом R (сигнал на входе R не оказывает никакого действия, пока  $S=1$ ).

#### **0BA0-0BA3:**

Действительная функция имеет вид:

Статус на выходе Q изменяется на противоположный при каждом переходе из 0 в 1 на входе Trg, то есть происходит включение и выключение выхода.

Импульсное реле переводится в его исходное состояние коротким импульсом на входе. Таким образом, выход сбрасывается в состояние логического 0.

Импульсное реле сбрасывается и выход  $Q=0$  после включения питания или сигнала сброса.

#### **5.3.6.3. Вывод текстовых сообщений (LOGO! 0BA6)**

Эта функция отображает тексты сообщений и параметры других блоков в LOGO! (рис. 5.73, 5.74, табл. 5.37). Служит для программирования текстовых дисплеев и модулей выносных дисплеев LOGO! TD.

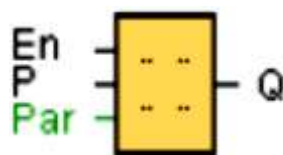


Рис. 5.73. Блок отображения текстовых сообщений на дисплее

Таблица 5.37

## Параметры бока вывода текстовых сообщений

Соединение	Описание
Вход <b>En</b>	Переход из 0 в 1 на входе En (разрешение) вызывает вывод текста сообщения.
Вход <b>P</b>	P определяет приоритет текста сообщения. 0 – минимальный, 127 – самый высокий приоритет. Ask: Подтверждение текста сообщения.
Параметр	<b>Text:</b> Ввод текста сообщения <b>Par:</b> Параметр или фактическое значение другой, ранее сконфигурированной функции, может отображаться в виде чисел или гистограммы (см. «Видимые параметры или фактические значения») <b>Time:</b> Отображает постоянно обновляемое текущее время <b>Date:</b> Отображает постоянно обновляемую дату <b>EnTime:</b> Отображает время перехода из 0 в 1 сигнала на входе En <b>EnDate:</b> Отображает дату перехода из 0 в 1 сигнала на входе En <b>Имена состояния входа/выхода:</b> Отображение имени состояния цифрового входа или выхода, например, «On» или «Off». <b>Analog Input:</b> Отображение входного аналогового значения в отображаемом текстовом сообщении и обновляемого в соответствии со значением аналогового времени.
Выход <b>Q</b>	Q остается установленным все время, пока тексты сообщений находятся в очереди.

## Настройки

В дополнение ко входам функционального блока и параметрам текста сообщения следующие настройки обеспечивают дополнительное управление отображением текстов сообщений:

- **Выбор набора символов:** можно выбрать составление текста сообщения из символов основного набора символов или вспомогательного набора символов. Можно задать двумя способами либо из меню «Настр.сбщ.» модуля LOGO! Basic, либо при помощи команды программы LSC «Файл -> Настройка текста сообщений».

- **Назначение сообщения:** Можно выбрать отображение текста сообщения на дисплее модуля LOGO!, на дисплее модуля LOGO! TD или на обоих дисплеях.

- **Настройка прокрутки:** текст сообщения может как прокручиваться, так и нет; прокручивание означает его прохождение по дисплею. Возможности прокрутки и предлагаемые настройки более подробно описываются ниже.

## Описание функции

При переходе из 0 в 1 сигнала на входе En и работе LOGO! в режиме RUN дисплей LOGO! и/или модуль LOGO! TD отображают заданный текст сообщения.

Если блок выбора «Подтверждение сообщения» не был выбран, то текст сообщения делается скрытым при изменении сигнала на входе  $E_n$  с 1 на 0.

Если блок выбора «Подтверждение сообщения» выбран, то после сброса в 0 входа  $E_n$  текст сообщения отображается до тех пор, пока он не будет подтвержден нажатием кнопки ОК. Текст сообщения не может подтверждаться, пока вход  $E_n$  находится в высоком состоянии.

В случае срабатывания нескольких функций текстовых сообщений при  $E_n=1$ , отображается сообщение с самым высоким приоритетом (0 = минимальный, 127 = максимальный). Это также подразумевает, что текст нового сообщения отображается только в том случае, если его приоритет выше ранее разрешенных текстов сообщений.

Если коммутационная программа использует флаг M27, то в случае, если M27=0 (низкий уровень) LOGO! отображает только те тексты сообщений, которые составлены из символов основного набора (Набор символов 1). Когда M27=1 (высокое значение), LOGO! отображает тексты сообщений только вспомогательного набора символов (Набор символов 2).

После отключения или подтверждения текста сообщения дисплей автоматически отображает ранее отображавшийся активный текст сообщения, имеющий наивысший приоритет.

### **Специальные характеристики, подлежащие учету при конфигурировании**

#### **1. Область имени блока**

Здесь задается имя блока, показано на рис. 5.74.

#### **2. Область настроек**

Здесь можно задать следующие настройки:

- приоритет текста сообщения;
- блок выбора подтверждения сообщения: в случае выбора требует подтверждения сообщения для его закрытия;
- выбор набора символов для текста сообщения.

#### **3. Область прокрутки**

Здесь задаются параметры прокрутки для текста сообщения:

- посимвольный формат прокрутки;
- построчный формат прокрутки;
- блок выбора разрешения прокрутки для каждой строки дисплея.

#### **4. Область назначения сообщения**

Задаётся область назначения для отображения сообщений на дисплее LOGO!, LOGO! TD или на обоих устройствах.

#### **5. Зона сообщений**

Задаётся выравнивание текста сообщения. Введенная в этой зоне информация соответствует тому, что будет отображаться на дисплее модуля LOGO! или модуля LOGO! TD.

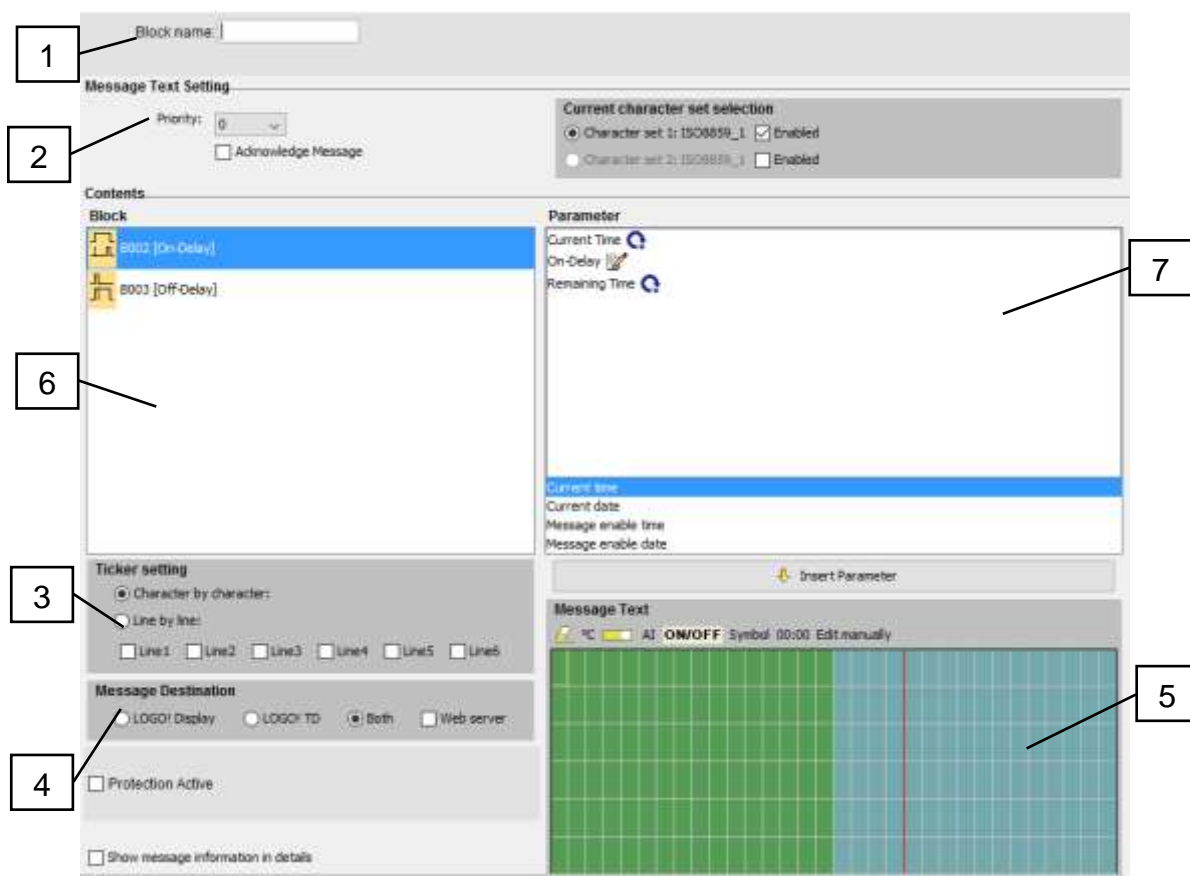


Рис. 5.74. Меню конфигурации блока текстовых сообщений

Над этой зоной располагаются дополнительные клавиши:



**Клавиша «Удалить»:** для удаления записей из области сообщений



**Клавиша специальных символов:** для вставки специальных символов в область сообщений



**Клавиша гистограммы:** для вставки горизонтальной или вертикальной гистограммы в область сообщения

**Клавиша AI:** для вставки аналогового входного значения в область «Сообщения».

**Клавиш включения и отключения:** для задания двух строк, представляющих цифровые значения, соответствующих состоянию 0 и состоянию 1, например – «ВЫКЛ.» и «ВКЛ.».

**Клавиша ручного редактирования:** для использования статического редактора с целью перемещения, добавления и изменения элементов текста сообщения без изменения положения других элементов.

## 6. Зона блоков

В этой зоне выполняется выбор блоков из всей совокупности блоков коммутационной программы. После этого обеспечивается воз-

возможность выбора параметров из этих блоков для отображения в тексте сообщения.

### **7. Зона параметров блока**


В этой зоне выполняется выбор параметров, подлежащих отображению в тексте сообщения из блока, выбранного в зоне блоков.

### **8. Клавиши вставки параметра**

Эта клавиша служит для вставки выбранного параметра блока в текст сообщения.

## **Гистограммы**

Можно задать представление текущего значения любого другого функционального блока программы в виде гистограммы. Гистограмма может отображаться на дисплее модуля LOGO! или LOGO! TD горизонтально или вертикально. Можно задать до четырех гистограмм на одно текстовое сообщение.

Для вставки гистограммы в область текста сообщения служит кнопка гистограммы . В диалоговом окне Настройка гистограммы необходимо ввести следующую информацию:

- Блок коммутационной программы, представляемый гистограммой.
- Минимальное и максимальное значение для гистограммы: Модуль LOGO! вычисляет длину или высоту гистограммы, масштабируя текущее значение между минимальным и максимальным значениями.
- Ориентация гистограммы: по горизонтали или по вертикали.
- Ширина или высота гистограммы в знаках.

## **Пример**

Рассмотрим применение в тексте сообщения гистограммы со следующими характеристиками:

- длина гистограммы после конфигурирования: 4 знака;
- ориентация: по горизонтали;
- заданное минимальное значение: 1000;
- заданное максимальное значение: 2000;
- фактическое значение: 1750.

В результате гистограмма будет занимать 3 знака.

## **Текстовое представление состояний цифровых входов и выходов**

Можно назначить такие названия, как «Вкл.» или «Выкл.» для двух состояний цифрового входа или выхода. В случае устройств LOGO! серии 0BA6 это название состояния цифрового входа или выхода может отображаться в тексте сообщения.

Максимальное число символов в названии состояния равно восьми для наборов символов западных языков; для наборов симво-



лов азиатских языков число символов равно четырем. Воспользуйтесь кнопкой ВКЛ./ВЫКЛ. в области текста сообщения для задания названий двух состояний цифрового входа или выхода, подлежащих использованию в тексте сообщения.

В одном тексте сообщения вы можете отображать до четырех названий состояний цифровых входов или выходов.

В одной программе в функциональных блоках текстов сообщений можно использовать до 20 имен состояний входов и выходов.

### **Отображение оставшегося времени таймера**

В случае устройств LOGO! серии 0BA6 в тексте сообщения может отображаться оставшееся время таймера. Для использования этой функции необходимо отобразить текущее истекшее время таймера и параметры таймера.

При размещении в тексте сообщения параметр оставшегося времени отображает время, оставшееся до истечения времени таймера. Для таймеров с многочисленными значениями таймеров (например, с временем задержки включения, временем задержки выключения) вы можете отображать в тексте сообщения оставшееся время каждого параметра.

### **Отображение аналоговых входов**

Значения аналоговых входов, могут быть выведены на текстовый дисплей. Для вставки конкретного AI в область текста сообщения служит кнопка AI.

Можно выбрать периодичность обновления 100 мс, 200 мс, 400мс, 800 мс и 1000 мс. Если в тексте сообщения используется больше одного аналогового входа, то периодичность обновления используется для всех входов.

### **Статический редактор («Редактирование вручную»)**

Программа LSC содержит статический редактор для текстов сообщений, позволяющий при необходимости изменить положение элементов текста. Например, у него имеется зона корзины, в которую вы можете временно перемещать элементы текста сообщения для изменения расположения элементов на дисплее. Статический редактор позволяет перемещать любые элементы вверх, вниз, влево или вправо, не изменяя положения других элементов.

Для того чтобы воспользоваться статическим редактором, нажмите кнопку «Редактировать вручную» над областью текста сообщения. В меню будет предложено выполнить ручное редактирование в случае попытки разместить или переместить в области сообщения элементы, положение которых вступает в конфликт с существующими элементами.

## **Режим эмуляции**

Компоновка текстов сообщений.

### **Обзор применения текстовых блоков сериях модулей**

#### **0BA0-0BA3:**

Максимальное число текстов сообщений: 5.

Не поддерживаются: прокрутка сообщения, гистограммы, аналоговые входы, названия состояний входов и выходов и оставшееся время таймера.

#### **0BA4-0BA5:**

Максимальное число текстов сообщений: 10.

Не поддерживаются: прокрутка сообщения, гистограммы, аналоговые входы, названия состояний входов и выходов и оставшееся время таймера.

## **Ограничения**

На функциональные блоки текстов сообщений накладываются следующие ограничения:

- может использоваться до 50-ти текстовых функций сообщений;
- может всего использоваться до 32-х гистограмм в текстах сообщений;
- программа LSC поддерживает все заданные параметры текстов сообщений. При выполнении программирования непосредственно в ПЛК вы можете запрограммировать только ограниченный набор параметров текстов сообщений;
- каждое сообщение может содержать 24 символа (наборы символов западных языков) или 16 символов (наборы символов азиатских языков).

На тексты сообщений накладываются следующие ограничения:

- Максимальное число параметров 4.
- Максимальное число гистограмм 4.
- Максимальное число имен состояний входа/выхода 4.
- Максимальное число значений времени/даты 4.
- Максимальное число аналоговых входов 2.

### **Специальные характеристики, подлежащие учету при конфигурировании**

Текст сообщения может быть сконфигурирован в диалоге свойств блока. Можно задать до четырех строк для каждого текста сообщения (текстовый дисплей модуля LOGO! и модуля LOGO! TD имеет четыре строки) и установить приоритет.

Можно задать отображение фактических значений других блоков в текстовых строках. Для этого выберите соответствующий блок из

диалога **Блок**. Открывается диалог **Параметр** для отображения перечня всех параметров, доступных для выбранного блока. Параметр блока, выбранного в этом диалоге, записывается в выбранную строку текста. Фактическое значение параметра теперь будет включено в текст при вызове текста сообщения.

#### 5.3.6.4. Программный выключатель

Этот специальный функциональный блок выполняется функции механической кнопки или выключателя показан на рис. 5.75.



Рис. 5.75

**Вход En.** Выход Q устанавливается по переходу сигнала из 0 в 1 на входе En (разрешение), если, помимо этого, в режиме конфигурирования был подтвержден параметр «Status=On».

**Переключатель:** Выбирает для функции либо режим кнопки без фиксации в одном цикле (Вкл.), либо режим переключателя (Откл.).

**Состояние:** Состояние Вкл. или Выкл., устанавливаемое в начальном цикле после запуска программы, если не были установлены функции сохранения.

**Сохранение** установлено (вкл.) = состояние сохраняется в памяти.

Выход Q остается установленным в положение 1 все время, пока En=1 и состояние параметров Тип = Переключатель и Состояние = Вкл.

Выход Q остается установленным в течение одного цикла, если EN=1 и Тип = без фиксации (кнопка) и Состояние = Вкл.

#### Описание функции

Выход устанавливается (рис. 5.76), когда вход En установлен, а для параметра «Состояние» установлено значение «Вкл.», подтвержденное кнопкой ОК. Это действие выполняется независимо от заданной функции выключателя или механической кнопки.

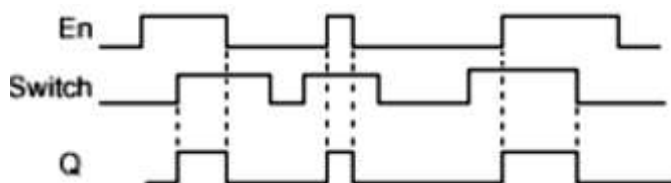


Рис. 5.76. Временная диаграмма

Выход сбрасывается в 0 в следующих трех случаях:

- при переходе сигнала на входе En из 1 в 0;

- когда задана функция кнопки и после ее задействования завершился один цикл.

когда параметр «Состояние» задает состояние «Выкл.» в режиме конфигурирования, и это было подтверждено кнопкой ОК.

### Специальные характеристики, подлежащие учету при конфигурировании

Программная клавиша может использоваться как в режиме кнопки без фиксации, так и переключателя. При помощи параметра состояния вы можете задать включенное (задействованное) или выключенное состояние переключателя/кнопки.

Если для программной кнопки задан режим кнопки без фиксации, то выход всегда устанавливается на время одного цикла при переходе из 0 в 1 на входе En, когда кнопка без фиксации находится во включенном состоянии, или если состояние кнопки без фиксации изменяется с Выкл. на Вкл., когда En=1.

#### 5.3.6.5. Регистр сдвига

Функция регистра сдвига может использоваться для считывания входного значения и для сдвига битов. Выходное значение соответствует настроенному биту регистра сдвига. Направление сдвига может быть изменено при помощи специального входа Dir (табл. 5.38).



Рис. 5.77. Временная функция регистра сдвига

Таблица 5.38

Параметры функции регистра сдвига

Соединение	Описание
Вход <b>In</b>	Функция при запуске считывает это входное значение.
Вход <b>Trg</b>	Специальный функциональный блок запускается по положительному фронту (переход из 0 в 1) входа Trg (выключатель). Переход из 1 в 0 не имеет значения.
Вход <b>Dir</b>	Направление сдвига битов сдвигового регистра S1...S8 задается при помощи входа Dir: Dir = 0: сдвиг вверх (S1 >> S8) Dir = 1: сдвиг вниз (S8 >> S1)
Параметр	<b>Бит регистра сдвига</b> , определяющий значение выхода Q. Возможные настройки: S1 – S8 <b>Сохранение</b> установлено (вкл.) = состояние сохраняется в памяти.
Выход <b>Q</b>	Выходное значение соответствует настроенному биту регистра сдвига.

## Описание функции

Функция считывает значение входа In по положительному фронту (переход из 0 в 1) входа Trg (выключатель) (рис. 5.78).

Значение записывается в биты регистра сдвига S1 или S8, в зависимости от заданного направления сдвига:

- Сдвиг в сторону старших разрядов: S1 принимает значение входа In; предыдущее значение S1 сдвигается в S2, S2 сдвигается в S3 и так далее.
- Сдвиг в сторону младших разрядов: S8 принимает значение входа In; предыдущее значение S8 сдвигается в S7, S7 сдвигается в S6 и так далее.

На выход Q поступает значение настроенных битов регистра сдвига.

Если не установлен режим сохранения, после аварии питания функция сдвига возобновляется с S1 или S8.

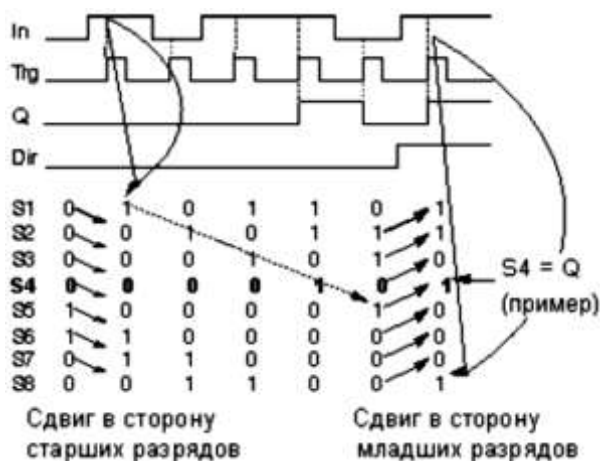


Рис. 5.78. Временная диаграмма

## Примечание

Специальная функция регистра сдвига может использоваться в коммутационной программе только один раз (рис. 5.79).

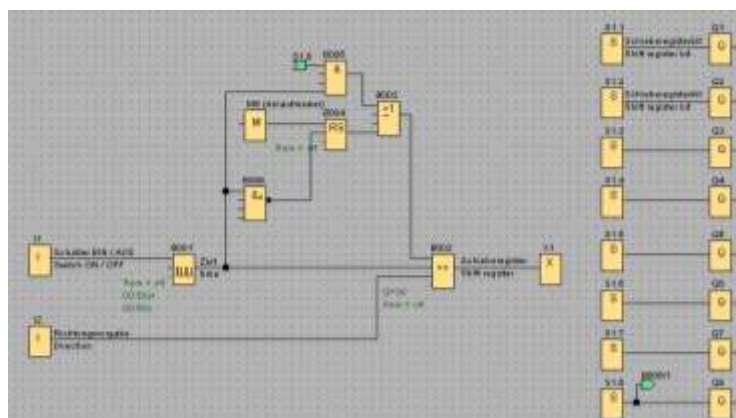


Рис. 5.79. Пример применения сдвигового регистра

### 5.3.6.6. Обнаружение ошибок аналоговых вычислений

Выход блока обнаружения ошибок аналоговых вычислений (рис. 5.80, табл. 5.39) устанавливается, если в соответствующем функциональном блоке аналоговых вычислений произошла ошибка.

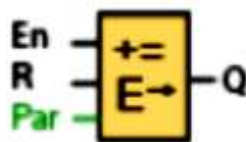


Рис. 5.80

Таблица 5.39

Параметры блока обнаружения аналоговых вычислений

Соединение	Описание
Вход <b>En</b>	Включение функционального блока обнаружения ошибок аналоговых вычислений.
Вход <b>R</b>	Сброс выхода.
Параметр	<b>Функциональный блок:</b> номер блока аналоговых вычислений. <b>Обнаруживаемая ошибка:</b> деление на ноль, переполнение или деление на ноль ИЛИ переполнение. <b>Автоматический сброс:</b> сброс выходного сигнала после устранения состояния ошибки.
Выход <b>Q</b>	Выход <b>Q</b> устанавливается, если во время выполнения соответствующего функционального блока аналоговых вычислений произошла обнаруживаемая ошибка.

**0BA1-0BA5:** Блок обнаружения ошибок аналоговых вычислений отсутствует в системах версии до 0BA6.

#### Параметр функционального блока

В значении параметра функционального блока указывается номер блока уже имеющегося в программе функционального блока аналоговых вычислений.

#### Описание функции

Выход блока обнаружения ошибок аналоговых вычислений устанавливается, если в соответствующем функциональном блоке аналоговых вычислений произошла ошибка. Функция может быть запрограммирована для установки выхода при ошибке деления на ноль, при ошибке переполнения или при любой из ошибок.

При установке флажка «Автоматический сброс» выход будет сбрасываться перед следующим выполнением функционального блока. Если флажок очищен, выходной сигнал сохраняется до сброса блока обнаружения ошибок аналоговых вычислений при помощи параметра **R**.

В любом цикле сканирования при выполнении соответствующего блока аналоговых вычислений до блока обнаружения ошибок аналоговых вычислений, ошибка будет обнаружена в том же цикле сканирования. Если соответствующий блок аналоговых вычислений выполняется после блока обнаружения ошибок аналоговых вычислений, то ошибка будет обнаружена в следующем цикле сканирования.

### **Логическая таблица обнаружения ошибок аналоговых вычислений**

В приведенной табл. 5.40 соответствует параметру инструкции обнаружения ошибок аналоговых вычислений, используемому для выбора обнаруживаемых ошибок. Ноль обозначает бит деления на ноль, установленный инструкцией аналоговых вычислений в конце выполнения: 1, если произошла ошибка, 0 – если нет. OF обозначает бит переполнения, установленный инструкцией аналоговых вычислений: 1, если произошла ошибка, 0 – если нет. Деление на ноль ИЛИ переполнение представляет собой результат применения функции «логическое ИЛИ» к биту деления на ноль и биту переполнения соответствующей инструкции аналоговых вычислений. Выход (Q) представляет собой выход функции обнаружения ошибок аналоговых вычислений.

Знак «х» означает, что бит может иметь значение 0 или 1, не оказывая влияния на выходное значение.

Таблица 5.40

#### Обнаруживаемые ошибки

Обнаруживаемая ошибка	Ноль	OF	Выход (Q)
Деление на ноль	1	х	1
Деление на ноль	0	х	0
Переполнение	х	1	1
Переполнение	х	0	0
Деление на ноль ИЛИ переполнение	1	0	1
Деление на ноль ИЛИ переполнение	0	1	1
Деление на ноль ИЛИ переполнение	1	1	1
Деление на ноль ИЛИ переполнение	0	0	0

### **Вопросы для самоконтроля**

1. Из каких компонентов состоит контур регулирования?
2. Как называется цепь, в которой находится датчик регулятора?
3. Что такое ошибка контура?
4. Что такое переходная характеристика регулятора?
5. Назовите основные типы регуляторов.
6. Приведите характеристики основных типов регуляторов.
7. Что такое ПИ-регулятор?
8. Что такое линейно-нарастающее регулирование?

### 5.3.7. Контроль памяти программ и данных

Блоки коммутационной программы требуют определенных объемов памяти. В табл. 5.41 указываются объемы памяти, занимаемыми каждым из блоков.

Объем памяти, необходимый для резервирования данных на случай аварии питания указан в столбце «Rem» (функция сохранения задействована).

Таблица 5.41

Объем памяти для резервирования данных

Блок	ОЗУ (байты)	Rem (байты)
И (без/с анализом фронта)	12	—
И-НЕ	12	—
ИЛИ (без/с анализом фронта)	12	—
ИЛИ-НЕ	12	—
Исключающее ИЛИ	8	—
НЕ	4	—
Задержка включения	8	3
Задержка отключения	12	3
Задержка включения и отключения	12	3
Задержка включения с сохранением	12	3
Интервальное реле	8	3
Интервальное реле с запуском по фронту	16	4
Асинхронный генератор импульсов	12	3
Генератор случайных импульсов	12	—
Выключатель лестничного освещения	12	3
Многофункциональный выключатель	16	3
Семидневный таймер	20	—
Годовой таймер	12	—
Реверсивный счетчик	28	5
Счетчик рабочего времени	28	9
Аналоговый пороговый выключатель	16	—
Аналоговый выключатель	16	—
Аналоговый дифференциальный выкл.	16	—
Аналоговый компаратор	24	—
Контроль аналоговых значений	20	—
Аналоговый усилитель	12	—
Блок ШИМ	24	—
Блок аналоговых вычислений	20	—
Обнаружение ошибок аналоговых вычисл.	12	1
Реле с блокировкой	8	1
Импульсное реле	12	1
Текст сообщения	8	—
Программный выключатель	8	2
Аналоговый мультиплексор	20	—
ПИ-регулятор	40	2
Линейно нарастающее регулирование	36	—
Регистр сдвига	12	1



Блоки коммутационной программы требуют определенных объемов памяти (табл. 5.42). В приведенной ниже таблице указаны объемы памяти, используемые блоками в различных областях памяти.

Таблица 5.42

**Объемы памяти, используемые блоками  
в различных областях памяти**

Блок	Par	ОЗУ (байты)	Таймер	Rem (байты)
Базовые функции	0	0	0	0
Задержка включения	1	1	1	0
Задержка отключения	2	1	1	0
Задержка включения и отключения	2	1	1	0
Задержка включения с сохранением	2	1	1	0
Интервальное реле	1	1	1	0
Интервальное реле с запуском по фронту	1	1	1	0
Генератор симметричных импульсов	1	1	1	0
Асинхронный генератор импульсов	3	1	1	0
Генератор случайных импульсов	2	1	1	0
Выключатель лестничного освещения	1	1	1	0
Многофункциональный выключатель	2	1	1	0
Семидневный таймер	6	2	0	0
Годовой таймер	2	0	0	0
Реверсивный счетчик*	2	(2)	0	(2)
Счетчик рабочего времени	2	0	0	4
Пороговый выключатель	3	3	1	0
Аналоговый выключатель	4	2	0	0
Аналоговый компаратор	3	4	0	0
Реле с блокировкой*	0	(1)	0	(1)
Импульсное реле*	0	(1)	0	(1)
Тексты сообщений	1	0	0	0
Программный выключатель*	1	(1)	0	(1)

\* В зависимости от конфигурирования функции (с сохранением или без сохранения) она занимает следующий объем памяти:

- Сохранение откл.: Функция занимает объем в ОЗУ;
- Сохранение вкл.: Функция занимает объем в Rem.

### **Расчёт объема памяти программы**

В программе возможно использование до 200 блоков. Максимальный объем памяти, используемый коммутационной программой LOGO! составляет:

- ОЗУ: 3800 байт;
- данные с сохранением: 250 байт.

Информационное окно отображает объем используемой памяти при вызове функции Сервис -> Определить LOGO! или при нажатии функциональной клавиши F2.

Применяются следующие спецификации приведены в табл. 5.43.

Таблица 5.43

## Спецификация модулей, использующих виды памяти

Серии LOGO!	Блоки	Par	ОЗУ	Таймер	REM	Флаги
LOGO! 0BA4... 0BA5	130	Без ограничения	Без ограничения	Без ограничения	60	24
LOGO! 0BA2... 0BA3	56	48	27	16	15	8
LOGO! 0BA1	56	48	27	16	15	4
LOGO! 0BA0	30	27	24	10	0/7	0

**Путь программы**

Путь программы состоит из ряда блоков, начинающихся со входа и завершающихся выходом.

**Глубина вложений (LOGO! серий 0BA0, 0BA1)**

Число блоков на пути программы называется глубиной вложений. Входами и выходами в смысле глубины вложений считаются:

- входы (I, AI);
- высокий, Низкий (Hi, Lo);
- маркеры памяти (M);
- выходы (Qblock.Output).

Блоками в смысле глубины вложений считаются:

- базовые функции;
- специальные функции.

Далее входами / выходами, зависящими от глубины вложения, являются:

- входы интерфейса AS (Ia);
- выходы интерфейса AS (Qa).

Система LOGO! ограничивает глубину вложений коммутационной программы. Максимальная глубина вложений программы составляет 58 объектов. Она определяет следующее:

**1 вход + 56 блоков + 1 выход = 58 (max. глубина пути программы)**

Во время эмуляции или загрузки система выдает сообщение об ошибке в случае ввода пути программы, глубина вложений которого превышает максимально допустимое значение.

**Реализация программ с большей длиной пути**

Более длинные пути программы могут быть реализованы с помощью рекурсии сигнала. Это обеспечивается добавлением флагового блока в конце пути (см. п.п. 5.1.1). Оставшиеся блоки соединяются с выходом флагового блока. LOGO! интерпретирует флаг как выход первого пути программы и также как клемный блок для последующих блоков. LOGO! интерпретирует длинный путь с флагом как два отдельных пути. При отсутствии свободных флагов можно использовать вместо них выходной блок.

### Вопросы для самоконтроля

1. На основании чего вычисляется объём используемой памяти?
2. Что называю глубиной вложений?
3. Какова максимальная глубина пути программы?
4. Как увеличивается длина пути программы?
5. Что такое выходной блок?







#### 5.3.8. Язык релейно-контактных схем (РКС)

Разрабатывая схемы алгоритмов автоматизации, мы всегда встречаем модель инициирования сигнала нуля или единицы с помощью простейшей модели – реализованной в виде релейных контактов.

Среда программирования контроллера LOGO! предоставляет возможность создавать программы на языке релейно-контактных схем РКС (LD). Поэтому следует привести описание и примеры основных функций языка релейно-контактных схем, показанных в табл. 5.44.

Таблица 5.44

Ячейки памяти и ввода-вывода

Отображение в редакторе релейно-контактных схем	
	Замыкающий контакт
	Размыкающий контакт
	Инвертированный выход
	Катушка реле
	Внутренний флаг
	Аналоговый контакт

Число доступных значков зависит от выбранной версии LOGO! (см. прил. 2).

##### 5.3.8.1. Размыкающие и замыкающие контакты

Размыкающие контакты, а также замыкающие контакты и аналоговые контакты представляют собой входные клеммы модуля ПЛК. Раскрывающееся окно открывается при вставке контакта в электрическую схему. В этом диалоге можно задать тип входа в соответствии с

выбранным типом модуля LOGO! (см. прил. 2). Могут использоваться клавиши управления курсором, а также как входы функциональные клавиши модуля выносного дисплея- LOGO! TD.

Для изменения входа в релейно-контактной схеме выполняется двойное нажатие соответствующего блока для открытия соответствующего диалогового окна, в котором можно выполнить необходимые изменения.

#### *5.3.8.2. Аналоговые контакты*

Аналоговые контакты, а также размыкающие контакты и замыкающие контакты являются вариантами входных клемм модуля LOGO!

Раскрывающееся окно открывается при вставке контакта в вашу электрическую схему. В этом диалоге вы можете задать тип входа в соответствии с вашим устройством LOGO!

#### *5.3.8.3. Катушка реле*

Обмотки реле представляют собой выходные клеммы наподобие инвертированных выходов и аналоговых выходов в системе LOGO!

#### *5.3.8.4. Инвертированный выход*

Инвертированные выходы, а также обмотки реле и аналоговые выходы представляют собой выходные клеммы модуля LOGO!

#### *5.3.8.5. Аналоговый выход*

Аналоговые выхода представляют собой выходные клеммы наподобие обмоток реле и инвертированного выхода в системе LOGO!

В случае подключения к специальной функции, имеющей аналоговый выход, к реальному аналоговому выходу, следует учесть, на то что аналоговый выход может обрабатывать только значения от 0 до 1000.

#### *5.3.8.6. Внутренний флаг*

Внутренний флаг применяется для завершения текущего пути и продолжения его в качестве нового пути. В отличие от блока флага, он не использует ресурс флага устройства LOGO!.

### **Вопросы для самоконтроля**

1. Как обозначаются контакты и катушки реле в языке LAD?
2. Для чего применяется внутренний флаг? Приведите пример схемы.
3. Что такое аналоговый выход в LOGO!?

## Глава 6. ПРИМЕРЫ ПРОГРАММ

Для того, чтобы составить представление о функциональных возможностях устройств микро ПЛК LOGO!, в программу LOGO! Soft Comfort включен набор практических примеров. Дополнительные примеры программ доступны в Интернете по адресу <http://www.siemens.com/logo>. При изучении, и расшифровки приводимых примеров программ, необходимо использовать справочные источники [5, 6, 7].

### 6.1. Пример №1. «Управление доступом»

#### Задание

Во многих случаях въезд в гараж или на стоянку перекрывается воротами с приводом и ручным управлением. Эти ворота открываются только при въезде и выезде транспортных средств. Ворота управляются оператором.

- Сдвижные ворота открываются и закрываются кнопкой в помещении контрольного пункта. Сдвижные ворота обычно бывают либо полностью открытыми, либо полностью закрытыми. Тем не менее, необходимо обеспечить возможность прерывания любого перемещения ворот.
- Защитный нажимной выключатель предохраняет людей от травм и имущество от повреждения при закрытии ворот.
- Сигнальный маяк задействуется за пять секунд до начала перемещения ворот и во время их движения.

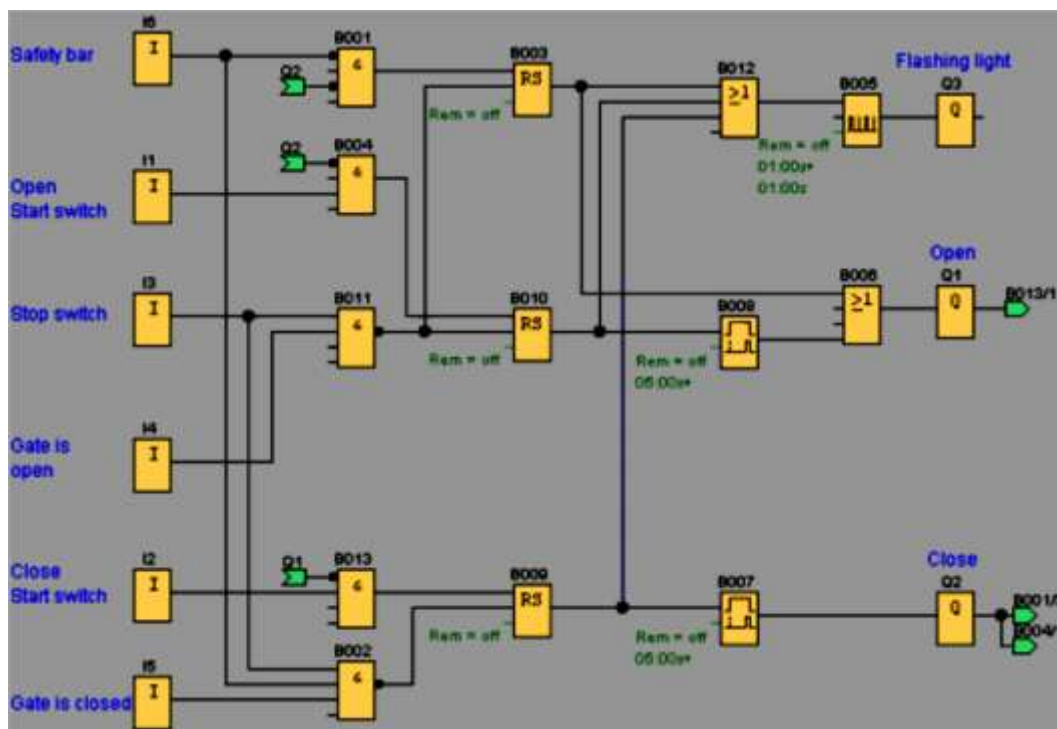


Рис. 6.1. Программа управления доступом

### **Стандартное решение**

Существует множество различных систем управления для автоматических ворот. Кнопки ОТКРЫТЬ и ЗАКРЫТЬ инициируют движение ворот в соответствующем направлении, если они не перемещаются в противоположном направлении. Перемещение ворот завершается либо при помощи кнопки СТОП, либо соответствующим конечным выключателем.

#### **Решение**

ПЛК позволяет реализовать дополнительные функции в сравнении со стандартными средствами управления (рис. 6.1): применение защитного нажимного выключателя прерывает закрывание ворот. За пять секунд до открытия или закрытия ворот задействуется сигнал маяка, свидетельствующий о начале движения. Он продолжает мигать до остановки ворот.

### **6.2. Пример №2. «Контроль мест для стоянки автомобилей»**

#### **Задание**

На автостоянке имеется в распоряжении определенное количество мест для стоянки автомобилей. Входной светофор должен автоматически переключаться с зеленого на красный, когда все места заняты. Как только места снова освобождаются, въезд снова обеспечивается включением зеленого сигнала.

#### **Решение**

Въезжающие и выезжающие автомобили подсчитываются с помощью с помощью фотодатчиков (на I1 и I2) со встроенным в LOGO! счетчиком (рис. 6.2). При въезде автомобиля (I1) счетчик увеличивается на 1, а при выезде автомобиля (I2) он снова уменьшается на 1. Через I2 и импульсную функцию задается направление счета (вперед/назад) на счетчике. При достижении установленного параметра переключается светофорное устройство на Q1. С помощью кнопки на I3 значение счетчика и выход Q1 могут быть сброшены.

#### **Используемые компоненты**

- напр., LOGO! 230R;
- I1 – фотодатчик «Въезд» (замыкающий контакт);
- I2 – фотодатчик «Выезд» (замыкающий контакт);
- I3 – кнопка сброса (замыкающий контакт);
- Q1 – реле светофора (переключающий контакт).

#### **Преимущества и особенности**

- Текущее значение счетчика может быть просто отображено через дисплей.

- Максимальное значение счетчика может изменяться произвольно.

Применение может быть легко расширено, напр., запрещать въезд, когда стояночные места заняты, или переключать между двумя значениями счетчика (стояночные места зарезервированы для служащих предприятия).

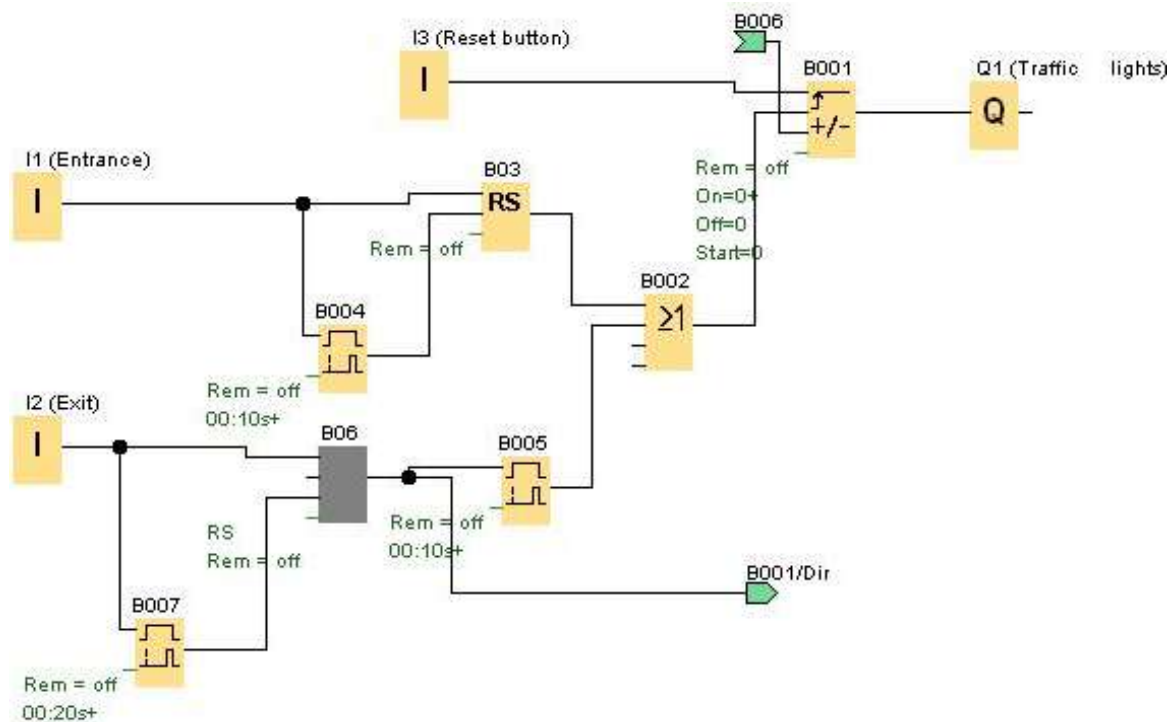


Рис. 6.2. Программа управления светофором стоянки

### 6.3. Пример №3. «Управление загрузкой бункера»

## Задание

Производится автоматическое управление и контроль загрузочными устройствами бункеров известью или цементом. Бункеры заполняются через загрузочный шланг грузового автомобиля.

## Решение

Процесс загрузки может быть начат только тогда, когда включен деблокирующий выключатель на I1 и загрузочный шланг надлежащим образом подключен (рис. 6.3). Язычковый контакт на загрузочном штуцере сигнализирует, правильно ли загрузочный шланг соединен с бункером. Этот сигнал считывается в LOGO! через вход I2. Затем открывается запорный клапан на Q2. Одновременно вводится в действие выпускной фильтр на Q1. Он должен быть включен в течение всего процесса заполнения. Теперь известь или цемент может закачиваться в бункер. Если бункер полон, то это сигнализируется через

предельный выключатель уровня заполнения на I3. Звуковой сигнал сообщает оператору, что до автоматического завершения процесса еще осталось 99 секунд. В течение этого времени должен быть закрыт клапан на грузовом автомобиле, чтобы еще освобождался от содержимого загрузочный шланг. Звуковой сигнал может быть досрочно отключен с помощью квитирующей кнопки на I6. Или он будет автоматически выключен через 25 секунд.

Если шланг не удалось своевременно освободить, то через кнопку на I5 можно выполнить аварийное заполнение в течение 30 секунд. Контроль избыточного давления в бункере также автоматически отключает процесс заполнения. Это отображается с помощью сигнальной лампы на Q4.

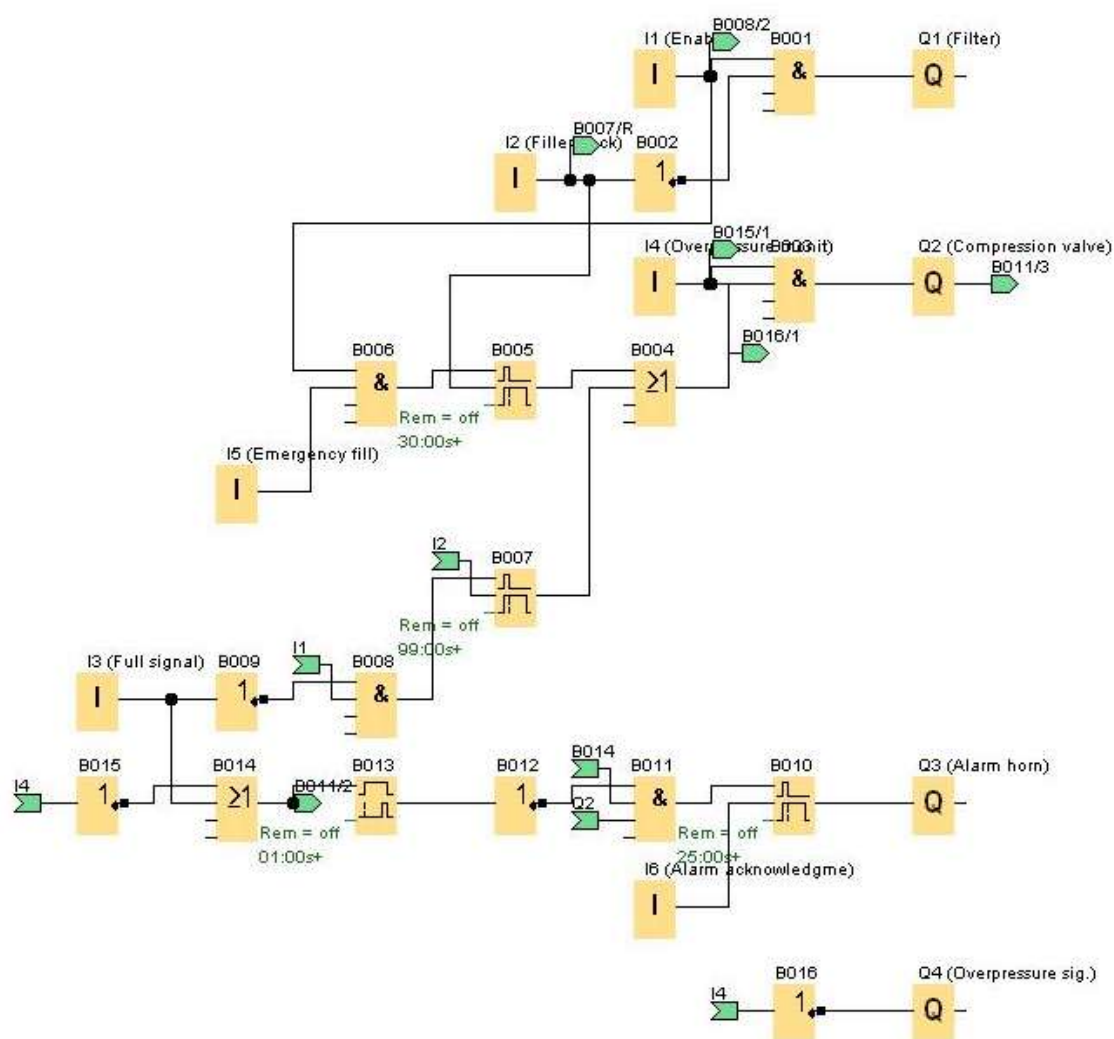


Рис. 6.3. Управление загрузочным устройством бункера

### Используемые компоненты

- - LOGO! 230R;
- I1 – деблокирующий выключатель (закрывающий контакт);



- I2 – язычковый контакт загрузочного штуцера (замыкающий контакт);
- I3 – предельный выключатель уровня заполнения (замыкающий контакт);
- I4 – выключатель избыточного давления (размыкающий контакт);
- I5 – кнопка аварийного заполнения (замыкающий контакт);
- I6 – кнопка квитирования звукового сигнала (замыкающий контакт);
- Q1 – фильтр;
- Q2 – запорный клапан;
- Q3 – звуковой сигнал;
- Q4 – световой сигнал избыточного давления.

#### 6.4. Пример №4. «Управление бетономешалкой»

##### Задание

Необходимо управлять мешалкой строительных смесей. С помощью переключателя режимов работы может быть выбран автоматический режим или режим ручного управления. Неисправности сигнализируются с помощью лампы и аварийного звукового сигнала.

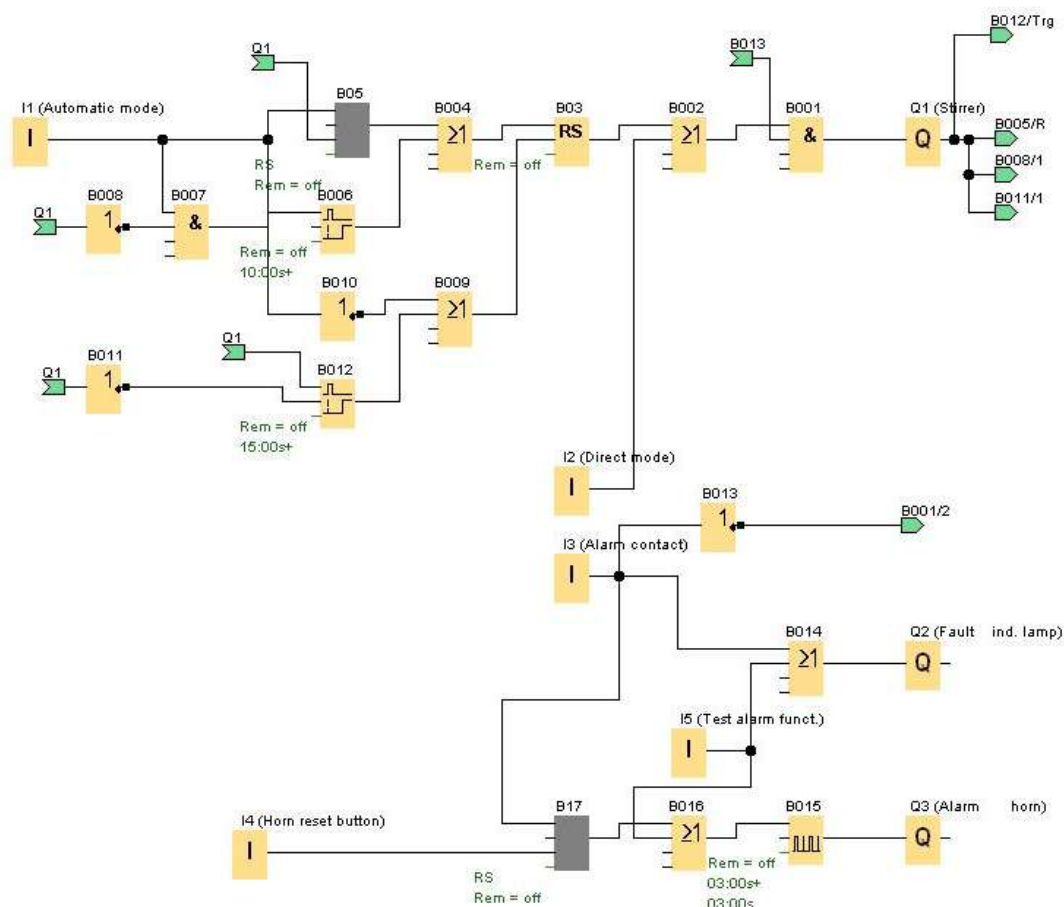


Рис. 6.4. Программа управления бетономешалкой

### Решение

Если переключатель режимов работы находится в положении «Автоматика» (I1), то мешалка (на Q1) запускается немедленно (рис. 6.4). Автоматический режим означает, что мешалка включается и выключается через заданные интервалы времени (15 секунд – включена, 10 секунд – пауза). Мешалка работает с этими интервалами, пока переключатель режимов работы не будет переведен в положение 0. В режиме прямого управления (I2 – положение «Прямое управление») мешалка работает без учета интервалов времени. При срабатывании автомата защиты двигателя (на I3) включается лампа сигнализации о неисправности (Q2) и аварийный звуковой сигнал (Q3). Интервалы, с которыми подается звуковой сигнал, устанавливаются с помощью датчика тактовых импульсов на 3 секунды. Звуковой сигнал может быть прерван с помощью кнопки сброса на I4. Если неисправность устранена, то сигнальная лампа и звуковой сигнал снова сбрасываются.

С помощью кнопки «Контроль аварийной сигнализации» на I5 можно проверить как сигнальную лампу, так и звуковой сигнал.

### Используемые компоненты

- напр., LOGO! 230R;
- I1 – переключатель режимов «Автоматика» (замыкающий контакт);
- I2 – переключатель режимов работы – положение «Прямое управление» (замыкающий контакт);
- I3 – аварийный контакт автомата защиты двигателя (замыкающий контакт);
- I4 – кнопка сброса звукового сигнала (замыкающий контакт);
- I5 – кнопка проверки функционирования аварийной сигнализации замыкающий контакт);
- Q1 – мешалка;
- Q2 – сигнальная лампа;
- Q3 – аварийный звуковой сигнал.

## 6.5. Пример №5. «Управление подъемной платформой»

### Задание

Реализовано управление подъемной платформой. Для контроля зоны перемещения подъемной платформы установлено несколько параллельно включенных датчиков (напр., ультразвуковых).

### Решение

Подъемная платформа с помощью кнопок может перемещаться вверх или вниз. Для этого к I1 подключена кнопка «Вверх», а к I3 –

кнопка «Вниз» (рис. 6.5). Соответствующее конечное положение распознается конечным выключателем. Конечный выключатель на I2 – для верхнего положения платформы, конечный выключатель на I4 – для нижнего положения. Если конечное положение достигнуто, то двигаться можно только в противоположном направлении. Направление перемещения задается через кнопки на I1 и I3. С помощью кнопки «Стоп» на I7 платформу можно остановить.

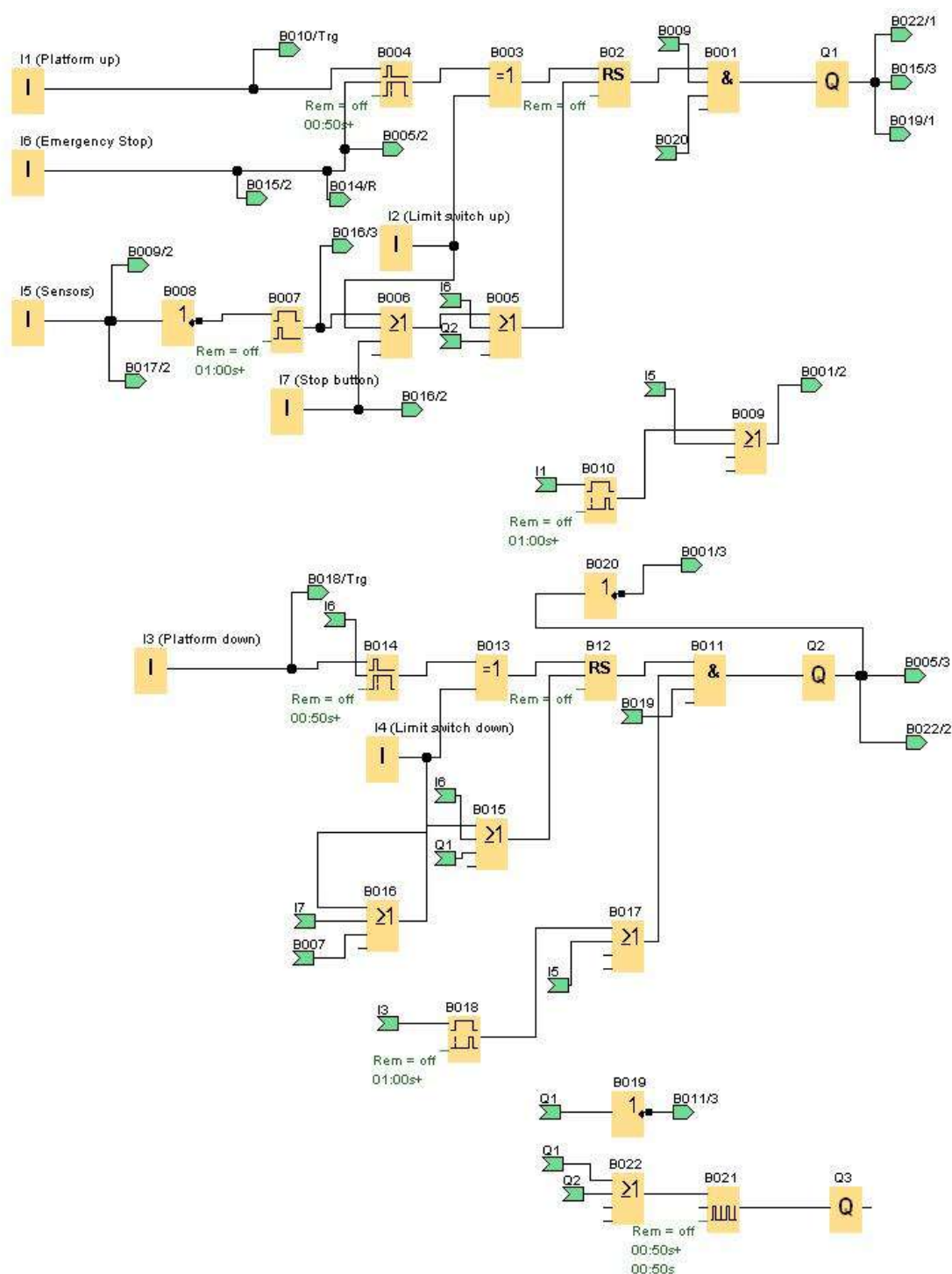


Рис. 6.5. Управление подъемной платформой

Ультразвуковые датчики для контроля зоны перемещения платформы подключены к I5. Если датчиками распознается препятствие, то платформа останавливается. Но ее можно перемещать в ручном режиме, если клавиша направления нажата дольше 2 секунд. Однако, если нажата кнопка аварийного останова на I7, то платформа останавливается немедленно и не может больше перемещаться с помощью кнопок направления, пока не отпущена кнопка аварийного останова. Для лучшего распознавания того, что платформа движется, активизируется предупредительное сигнальное устройство на Q3. Если платформа движется вверх или вниз, то мигает предупредительный световой сигнал на Q3.

### **Используемые компоненты**

- напр., LOGO! 230RL;
- I1 – кнопка «Вверх» (замыкающий контакт);
- I2 – верхний конечный выключатель (замыкающий контакт);
- I3 – кнопка «Вниз» (замыкающий контакт);
- I4 – нижний конечный выключатель (замыкающий контакт);
- I5 – датчики (размыкающие контакты);
- I6 – кнопка «Стоп» (замыкающий контакт);
- I7 – кнопка аварийного останова (замыкающий контакт);
- Q1 – платформа вверх;
- Q2 – платформа вниз;
- Q3 – предупреждающий световой сигнал.

### **Преимущества и особенности**

Времена задержки могут быть просто приспособлены к соответствующей ситуации. Установка может быть легко расширена/изменена без дополнительных компонентов.

## **6.6. Пример №6. «Наполнительная станция»**

Контейнер заполняется двумя видами предметов, до достижения заданного количества по каждому из них. После того, как все предметы будут положены в коробку, она перевозится на станцию упаковки. Транспортеры подают предметы обоих типов к контейнеру. (в примере наполнительный транспортер не показан).

### **Описание технологического процесса**

- Для заполнения контейнера каждая позиция доставляется транспортером к контейнеру.
- Последовательность падения предметов в контейнер является произвольной.
- Все предметы, падающие в контейнер, подсчитываются датчиком.

- Подключенный модуль LOGO! TD, а также дисплей модуля LOGO! должны отображать, сколько предметов каждого типа было подсчитано и сколько всего предметов находится в контейнере.

### **Коммутационная программа**

Датчики на фотоэлементах подключаются к двум цифровым входам I1 и I2 и обнаруживают падение предметов в контейнер (рис. 6.6). Два счетчика (B001 и B002) подсчитывают отдельно предметы двух отдельных типов при их падении в коробку.

Порог включения каждого счетчика задает максимально возможное количество по каждому типу предмета. После заполнения коробки транспортер включается на десять секунд для перемещения заполненной коробки на упаковочную станцию и для перемещения пустого контейнера на наполнительную станцию.

В коммутационной программе используется функциональный блок текста сообщения, который отображает на дисплее модулей LOGO! TD общее количество предметов и число предметов каждого типа. Для текста сообщения используется функция прокрутки, позволяющая попеременно отображать гистограммы подсчитанных предметов и текстовое сообщение о результатах счета.

### **Параметры функциональных блоков**

Пороги включения счетчиков B001 и B002 определяют максимальное число предметов каждого типа, подлежащих размещению в контейнере.

Ширина импульса (ТН) интервального реле с запуском по фронту определяет продолжительность перемещения транспортера.

Блок аналоговой математической функции B007, вычисляющий общее количество, запрограммирован следующим образом:

- Значение-1 является опорным параметром для текущего значения счетчика B001.
- Значение-2 является опорным параметром для текущего значения счетчика B002.
- Оператор 1 имеет значение «+» для того, чтобы суммировать два счетчика.
- Приоритет 1 имеет высокое значение. (Это – единственная операция функционального блока, поэтому приоритет в данном случае не имеет значения.)

Остальные операнды и операторы установлены равными «+0» и не влияют на выходные значения уравнения.

### **Тексты сообщений**

Блок текста сообщения B010 показывает текст сообщения, четыре строки которого прокручиваются по одной строке, при этом попе-

ременно отображаются две экранные формы. На первой форме показаны гистограммы подсчитанных предметов (текущие значения B001 и B002) и гистограмма суммарного количества, подсчитанного блоком B007. На второй экранной форме показаны те же значения в виде чисел с текстовыми описаниями.

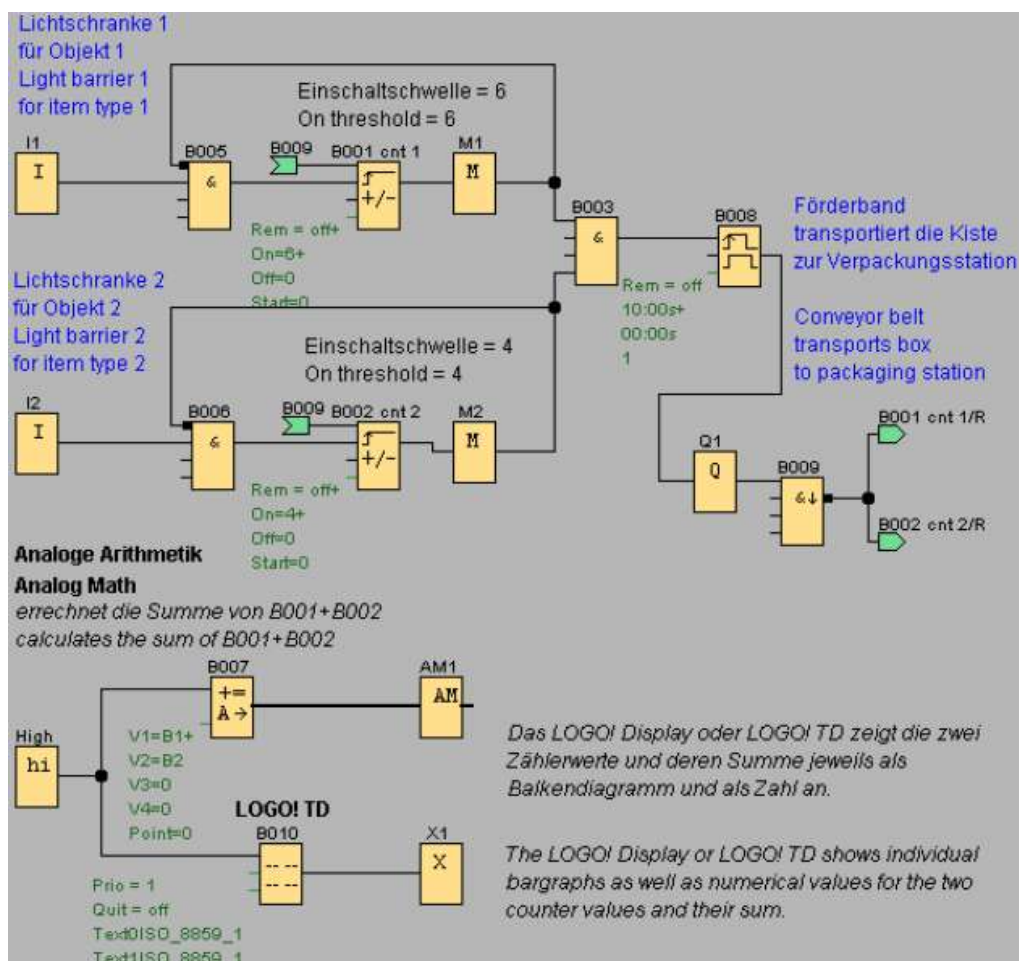


Рис. 6.6. Конвейерная наполнительная станция

Функциональный блок текста сообщения настроен с двумя (нем./англ.) включенными наборами символов. Для выбора двух наборов символов используется команда меню «Файл -> Настройка» текста сообщений. В рассматриваемом случае выбирается один и тот же набор символов для отображения символов английского и немецкого языка. Когда ПЛК находится в режиме RUN, отображаемое сообщение определяется текущим выбранным набором символов.

## 6.7. Пример №7. «Освещение и сигнализация жилого дома»

### Задание

Производить управление внешним и внутренним освещением жилого дома. В случае отсутствия хозяев или в темное время суток

оно должно сигнализировать о приближении людей. Через сигнализатор перемещения и контакт тревоги установки тревожной сигнализации включается внешнее и внутреннее освещение.

### Решение

Внешнее освещение разделено на три области (Q1, Q2, Q3) (рис. 6.7). Для каждой области используется собственный сигнализатор перемещения (I2, I3, I4). Если на определенном интервале времени срабатывает один из этих сигнализаторов, то соответствующее внешнее освещение включается на 90 секунд.

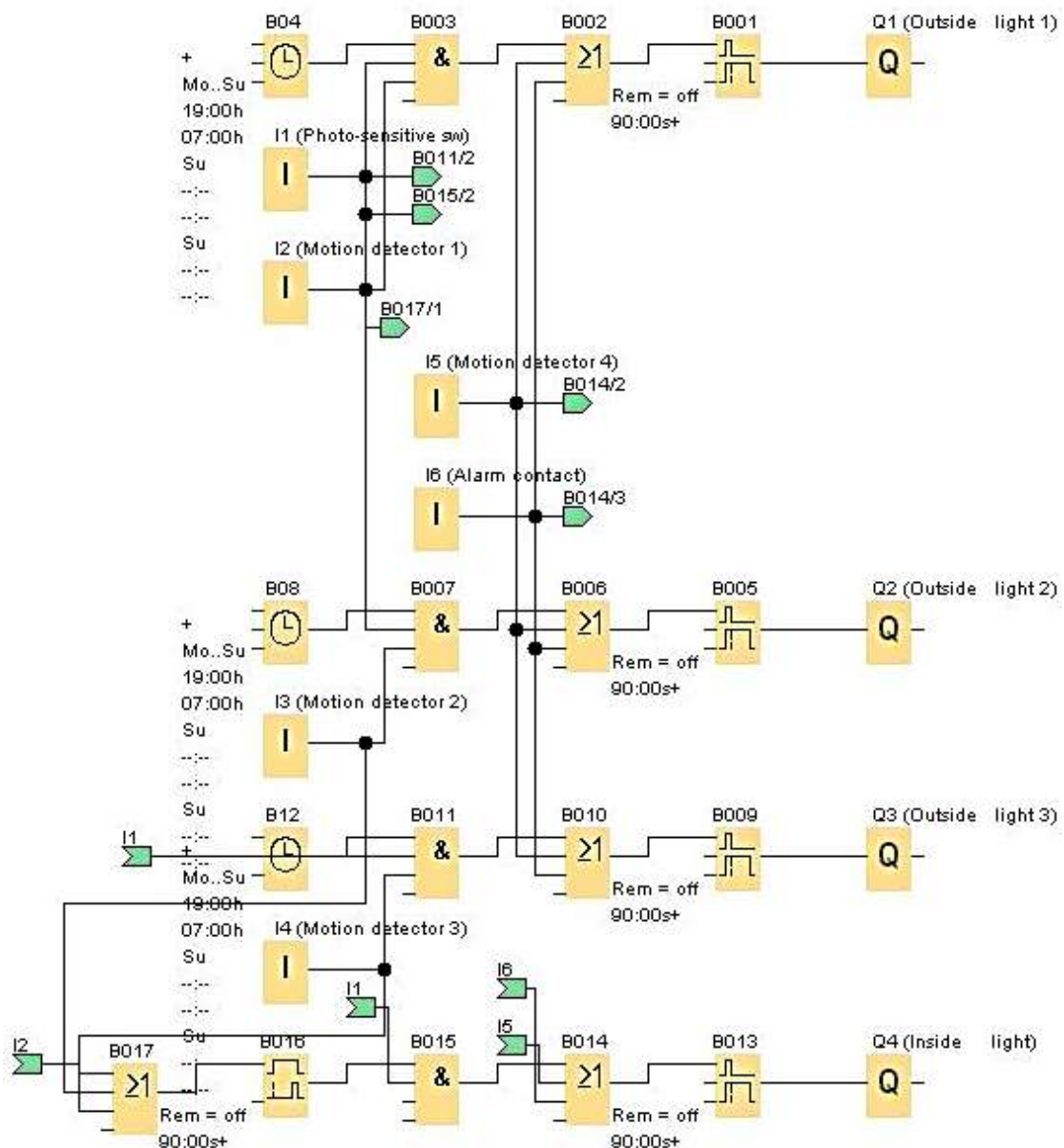


Рис. 6.7. Программа управления освещением и сигнализацией

Диапазон времени задается через встроенное реле времени (с 17:00 до 7:00). Благодаря сумеречному реле на I1 гарантируется, что

включение происходит только в темное время суток. На I5 подключен четвертый сигнализатор перемещения, который независимо от времени и темноты включает все три внешних освещения на 90 секунд.

Внешние освещения включаются также на 90 секунд через контакт тревоги установки тревожной сигнализации на I6. Кроме того, после отключения внешнего освещения на 90 секунд включается внутреннее освещение. Через сигнализатор перемещения на I5 и контакт тревоги внутреннее освещение включается на 90 секунд немедленно.

### **Используемые компоненты**

- LOGO! 230RC;
- I1 – сумеречное реле (замыкающий контакт);
- I2 – сигнализатор перемещения 1 (замыкающий контакт);
- I3 – сигнализатор перемещения 2 (замыкающий контакт);
- I4 – сигнализатор перемещения 3 (замыкающий контакт);
- I5 – сигнализатор перемещения 4 (замыкающий контакт);
- I6 – контакт тревоги установки тревожной сигнализации (замыкающий контакт);
- Q1 – внешнее освещение 1;
- Q2 – внешнее освещение 2;
- Q3 – внешнее освещение 3;
- Q4 – внутреннее освещение.

### **Преимущества и особенности**

Экономия энергии благодаря осуществляется сопряжению реле времени, сумеречного реле и сигнализаторов перемещения.

Простота изменения установленных временных пределов, например, другого диапазона реле времени или другой длительности освещения.

## **6.8. Пример №8. «Контроль длительности подключения потребителей (нагрузки)»**

### **Задание**

Должно быть обеспечено, что бы потребители могли быть включены только на определенный интервал времени. Если заданное время превышено, то ПЛК автоматически отключает потребителей (нагрузку). Это, например, очень полезно для солнечных энергетических установок, так как при этом удастся избежать глубокого разряда батарей.

### **Решение**

ПЛК контролирует длительность включения присоединенных потребителей (рис. 6.8). Для отдельных потребителей может быть задано различное время. Каждому выходу ставится в соответствие вход,



т.е. если нажат выключатель на входе I1, то немедленно включается потребитель на Q1. В течение заданного интервала времени потребитель может произвольно часто включаться и выключаться. Но если интервал включения превышен, то LOGO! автоматически отключает этого потребителя. Остальные входы и выходы (I2, I3 и Q2, Q3) связаны друг с другом таким же образом.

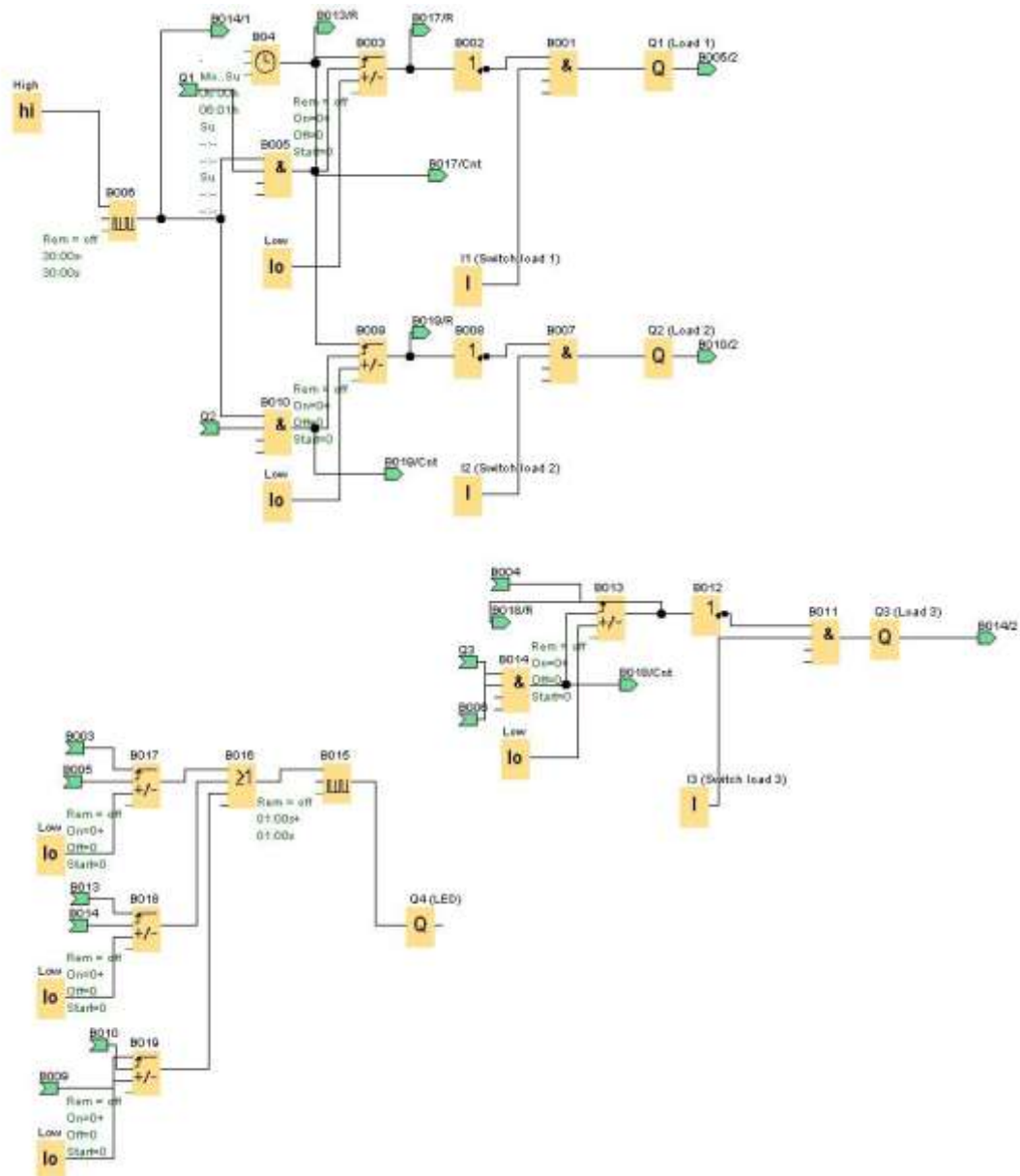


Рис. 6.8. Контроль длительности подключения потребителей

Завершение деблокировки реализовано следующим образом: Через встроенный в LOGO! счетчик устанавливается текущая длительность включения путем подачи каждую минуту на счетчик импульса датчиком тактовых импульсов. Тем самым может быть подсчитано число прошедших минут. Заданное граничное значение соответ-

ствуется максимальной длительности включения (напр., 120 = 120 минут для Q1). Когда это значение счетчика достигается, потребитель отключается. Потребитель остается заблокированным до тех пор, пока через реле времени не будет дана команда на деблокировку (напр., каждый день в 6:00). Чтобы дать сигнал о том, что максимальное время включения скоро истечет, к выходу Q4 подключена дополнительная сигнальная лампа, которая начинает мигать за 15 минут до истечения времени.

### **Используемые компоненты**

- LOGO! 12RC;
- I1 – выключатель для потребителя 1 (замыкающий контакт);
- I2 – выключатель для потребителя 2 (замыкающий контакт);
- I3 – выключатель для потребителя 3 (замыкающий контакт);
- Q1 – потребитель 1;
- Q2 – потребитель 2;
- Q3 – потребитель 3;
- Q4 – сигнальная лампа.

### **Преимущества и особенности**

Благодаря автоматическому отключению потребителей гарантируется, что батареи защищены от глубокого разряда. Времена включения могут быть просто изменены и адаптированы к той или иной ситуации. Момент деблокировки может произвольно изменяться для каждого потребителя, напр., только раз в неделю. Кроме того, через реле времени работа потребителей может быть ограничена определенными интервалами времени.

## **6.9. Пример №9. «Система кондиционирования воздуха»**

### **Задание**

Система кондиционирования воздуха подает в помещение свежий воздух и удаляет загрязненный. Рассмотрим следующий пример системы:

- в помещении установлены вытяжной насос и насос для подачи свежего воздуха;
- работа каждого из вентиляторов контролируется датчиком расхода;
- давление в помещении может повышаться сверх атмосферного;
- вентилятор подачи свежего воздуха может быть включен только в том случае, когда датчики сигнализируют о безопасном рабочем состоянии вытяжного вентилятора;

- отказ одного из вентиляторов индицируется при помощи сигнальной лампы.

Работа вентиляторов контролируется датчиками расхода. При отсутствии потока воздуха после истечения короткой задержки система выключается и отображается сообщение об ошибке, которое может быть подтверждено нажатием кнопки выключения.

### Разработка схемы

Входы I1 и I2 служат для включения и выключения системы (рис. 6.9). Вентиляторы подключены к выходам Q1 и Q2, датчики расхода подключены ко входам I3 и I4.

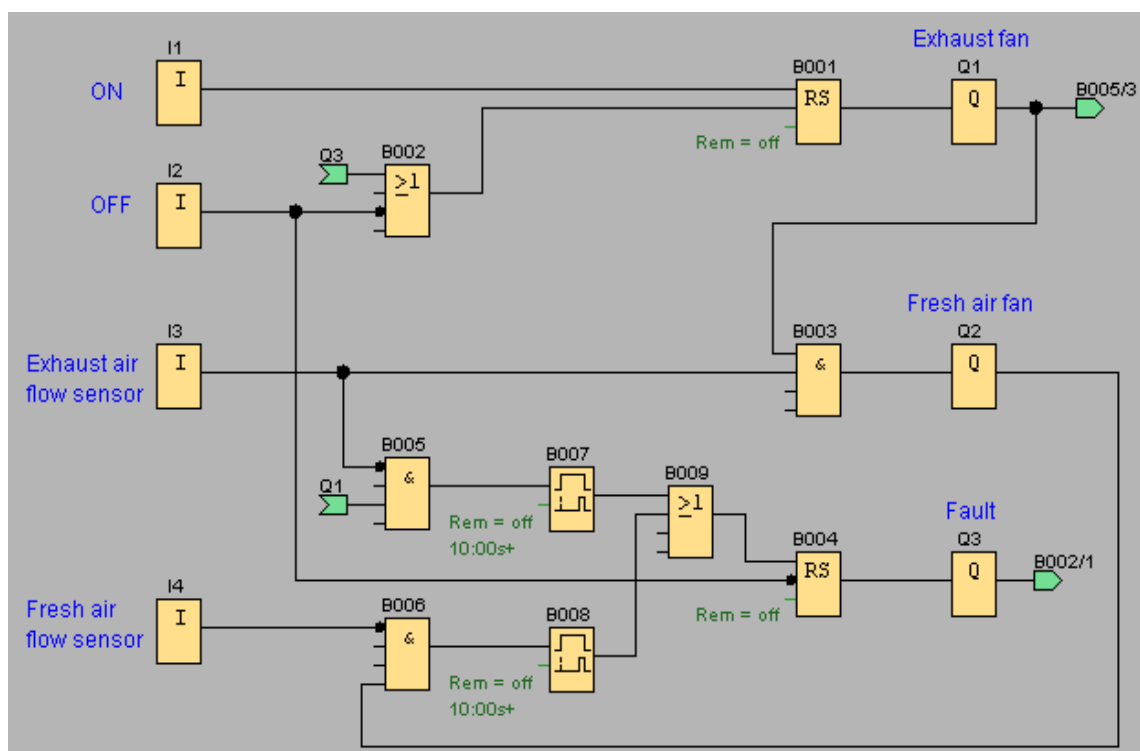


Рис. 6.9. Система кондиционирования воздуха

Блоки B007 и B008 служат для установки контрольных значений времени, по истечении которых датчики расхода должны направить сигнал на выход отказа Q3.

Можно инвертировать выход Q3 для использования выходных сообщений на выходе Q4. Реле Q4 отключается только в случае аварии сетевого питания или сбоя в системе. Поэтому выход может использоваться для передачи дистанционного сообщения.

### 6.10. Пример №10. «Базовые арифметические операции»

На рис. 6.10. приведён пример базовых арифметических операций(+, -, x, /) с значениями аналоговых сигналов.

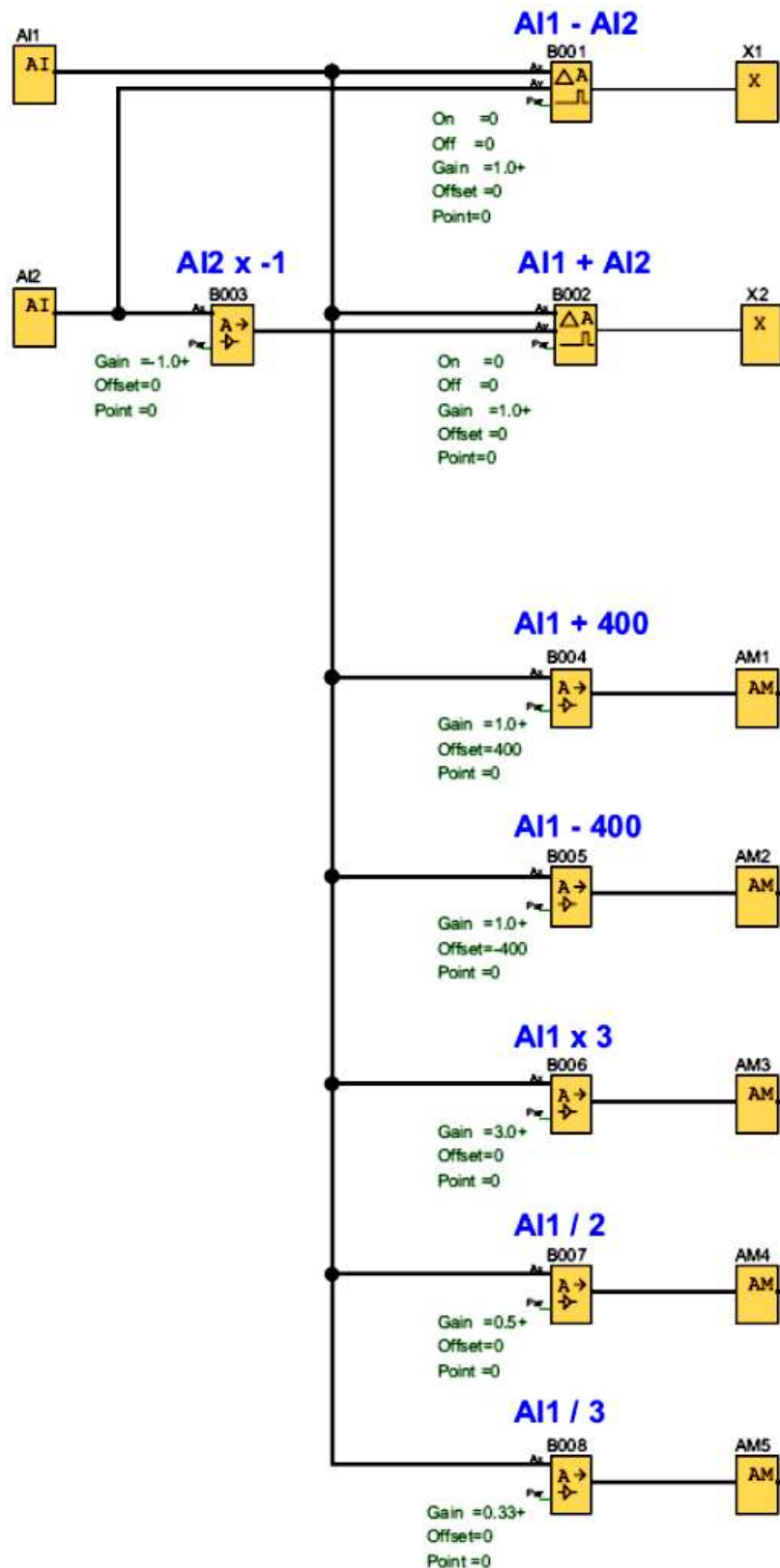


Рис. 6.10. Программа базовых арифметических операций

На рис. 6.11. приведён пример подсчёта количества входных импульсов дискретных сигналов, поступающих на два счётчика со входами I1, I2 с последующим суммированием.

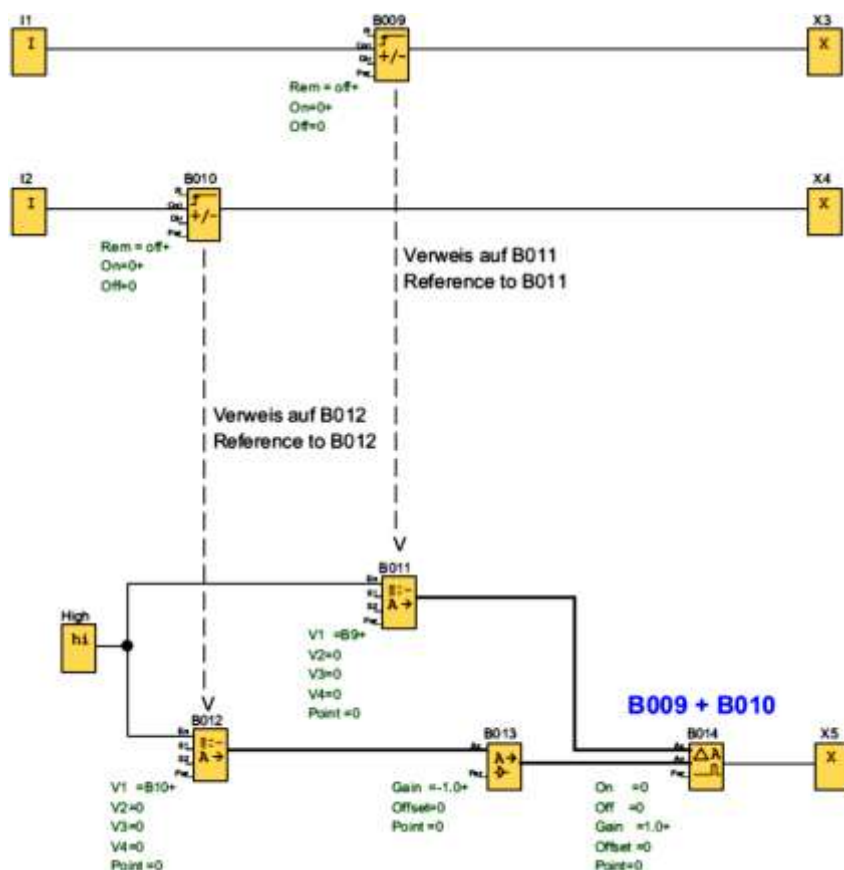


Рис. 6.11. Подсчёт импульсов с последующим суммированием

На рис. 6.12. приведён пример преобразование результата суммирования, увеличенного в 10 раз, в значение уровня аналогового сигнала.

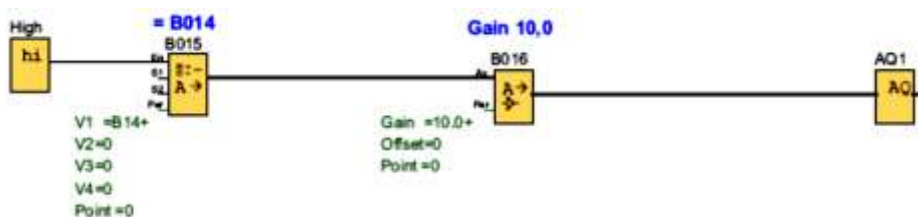


Рис. 6.12. Масштабирование значения в уровень аналогового сигнала

### 6.11. Пример №11. «Управление отоплением. Регулирование»

Этот пример иллюстрирует на рис. 6.13 реверсивное изменение температуры подводимого тепла и наружной температуры при управлении системой отопления. Температура подводимого тепла системы отопления должна управляться обратно пропорционально наружной температуре. Это означает: Чем ниже наружная температура, тем выше температура подводимого тепла. Наружная температура и температура подводимого тепла измеряются датчиками РТ100. Когда наружная температура равна 0град.С, температура подводимого тепла

должна составлять 50град.С (х). Отопление должно включаться, если наружная температура опускается ниже 4град.С.

1. Датчик PT100 подсоединен к AI1 и служит для измерения подводящего тепла.

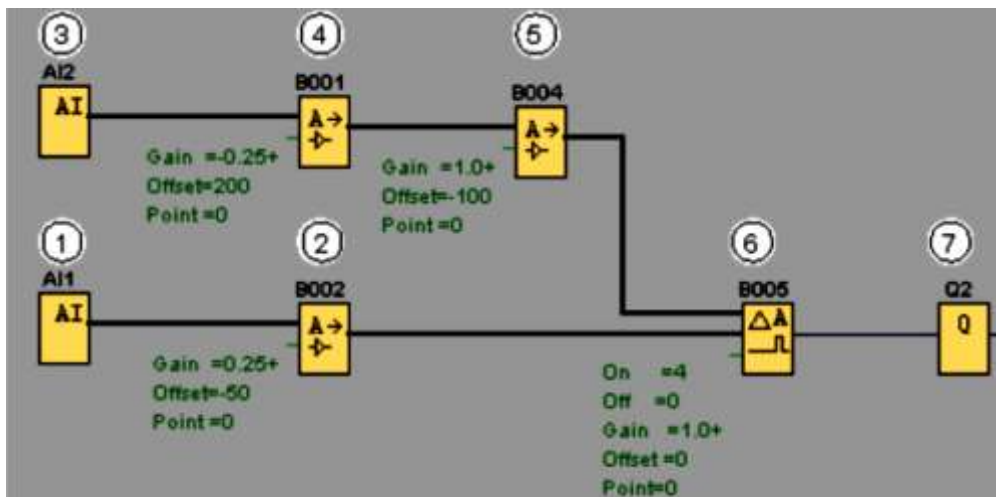


Рис. 6.13. Программа управления отоплением (Heating.lsc)

2. Параметры аналогового усилителя заданы следующим образом:
- датчик: PT100 (пропорциональный);
  - диапазон измерения и параметры определяются датчиками PT100;
  - единица измерения: градусы Цельсия;
  - разрешение: x 1.

Усилитель вызывает передачу на выход фактической температуры, измеряемой датчиком.

3. Датчик PT100 подсоединен к AI2 и служит для измерения наружной температуры.

4. Параметры аналогового усилителя заданы следующим образом:
- Датчик: PT100 (обратно пропорциональный);
  - Диапазон измерения и параметры определяются датчиками PT100;
  - Единица измерения: градусы Цельсию;
  - Разрешение: x 1.

Усилитель вызывает передачу на выход значения, обратно пропорциональное измеренной датчиком температуры.

Следовательно, чем выше наружная температура, тем ниже передаваемое значение.

5. Параметры аналогового усилителя заданы следующим образом:
- Датчик: без датчика;
  - Усиление: 1;
  - Смещение: -100 (y).

Значение, выдаваемое (4), корректируется (нормируется) этим аналоговым усилителем таким образом, что ее можно сравнивать с температурой подводимого тепла.

6. Параметры аналогового компаратора заданы следующим образом:

- Датчик: без датчика;
- Усиление: 1;
- Смещение: 0;
- Входное пороговое значение: 4
- Выходное пороговое значение: 0.

Аналоговый компаратор включает выход Q2, (7) если разность между температурой подводимого тепла и нормированной наружной температурой превышает 4°C.

Если разность приближается к 0град.С, аналоговый компаратор снова выключает выход Q2.

7. Выход Q2 включает и выключает отопление.

### **Режим работы**

Наружная температура уменьшается; это вызывает пропорциональное увеличение значения, (6) подаваемого на аналоговый усилитель. На аналоговом компараторе увеличивается разность между температурой подводимого тепла и наружной температурой. При достижении разности в 4°C включается отопление.

Включение отопления повышает температуру подводимого тепла. По этой причине разность на аналоговом компараторе между температурой подводимого тепла и наружной температурой уменьшается (при условии, что наружная температура уменьшается медленнее повышения температуры подводимого тепла). При достижении разности в 0град.С отопление выключается.

### **Изменение параметров**

Параметр Смещение (y) аналогового усилителя (5) зависит от требуемой температуры подводимого тепла (x) при наружной температуре 0град.С. Параметр рассчитывается следующим образом:

$$y = x - 150.$$

Более того, порог включения и порог выключения отопления может быть изменен при помощи порога аналогового компаратора (6).

### **Задание**

1) Сократите блок (5) за счет надлежащей установки порога аналогового компаратора.

2) Выполните различные варианты.

3) Загрузите программу и попытайтесь выполнить приведенные выше указания в режиме эмуляции.

**ПРИЛОЖЕНИЕ 1**

<b>Версия</b>	<b>Имя</b>	<b>Номер для заказа</b>
Стандартн.	LOGO! 12/24RC (пост, ток)	6ED1 052-1MD00-0BA6
Стандартн.	LOGO! 24 (пост, ток)	6ED1 052-1CC00-0BA6
Стандартн.	LOGO! 24RC (пер./пост. ток)	6ED1 052-1HB00-0BA6
Стандартн.	LOGO! 230RC (пер. ток)	6ED1 052-1FB00-0BA6
Стандартн.	LOGO! 12/24RCo (пост, ток)	6ED1 052-2MD00-0BA6
Стандартн.	LOGO! 24RCo (пер./пост. ток)	6ED1 052-2HB00-0BA6
Стандартн.	LOGO! 230RCo (пер. ток)	6ED1 052-2FB00-0BA6
Стандартн.	LOGO! 24o (пост, ток)	6ED1 052-2CC00-0BA6

<b>Версия</b>	<b>Имя</b>	<b>Номер для заказа</b>
Цифровые	LOGO! DM8 230R	6ED1 055-1FB00-0BA1
Цифровые	LOGO! DM16 230R	6ED1 055-1FBIO-0BA0
Цифровые	LOGO! DM8 24	6ED1 055-1CB00-0BA0
Цифровые	LOGO! DM16 24	6ED1 055-1CBЮ-0BA0
Цифровые	LOGO! DM8 12/24R	6ED1 055-1MB00-0BA1
Цифровые	LOGO! DM8 24R (пер./пост. ток)	6ED1 055-1HB00-0BA0
Цифровые	LOGO! DM16 24R	6ED1 055-1NB10-0BA0
Аналоговые	LOGO! AM2 (пост, ток)	6ED1 055-1MA00-0BA0
Аналоговые	LOGO! AM2 PT100 12/24	6ED1 055-1MD00-0BA0
Аналоговые	LOGO! AM2 AQ (пост, ток)	6ED1 055-1MM00-0BA1
Текстовый дисплей	LOGO! TD	6ED1 055-4MH00-0BA0



## ПРИЛОЖЕНИЕ 2

## Специальные функции в зависимости от версий модулей

[illegible]

## СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 51841-2001. Программируемые контроллеры. Общие положения и функциональные характеристики.
2. Теория, логическое проектирование, измерение, контроль и диагностика в системах автоматического управления: монография / В.А. Воробьев [и др.]. – М.: Российская инженерная академия, 2009. – 790 с.
3. Остроух, А.В. Автоматизация управления производством. Повышение эффективности автоматизированных аналитических систем предприятий автомобильной промышленности / А.В. Остроух, Э.А. Чернов, Д.Т. Нгуен. – Saarbrücken: LAP LAMBERT Academic Publishing, 2013. – 284 с.
4. Парр, Э. Программируемые контроллеры. Руководство для инженера / Э. Парр; пер. с англ. – 3-е изд. – М.: БИНОМ Библиотека знаний, 2009. – 516 с.
5. Петров, И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного программирования / И.В. Петров; под ред. проф. В.П. Дьяконова. – М.: СОЛОН-Пресс, 2004. – 256 с.
6. Руководство по программированию в среде LOGO!SoftComfort 6.0 (Siemens).
7. Система LOGO и программа LOGO!SoftComfort в сети Интернет <http://www.siemens.com/logo/>
8. [http://sitrain.automation.siemens.com/DE/sitrain/mcms/sitrain2/content/DE\(en\)/Training/Trainingstopics/Pages/Microautomation.aspx#footer-position-wrapper](http://sitrain.automation.siemens.com/DE/sitrain/mcms/sitrain2/content/DE(en)/Training/Trainingstopics/Pages/Microautomation.aspx#footer-position-wrapper).

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
-----------------------	----------

**Глава 1. ПРОГРАММИРУЕМЫЕ**

<b>ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ .....</b>	<b>4</b>
1.1. Основные характеристики ПЛК .....	4
<i>Вопросы для самоконтроля .....</i>	<i>8</i>
1.2. Введение в стандарт МЭК 61131-3 .....	8
<i>Вопросы для самоконтроля.....</i>	<i>10</i>
1.3. Инструментальная среда разработки программ.....	10
<i>Вопросы для самоконтроля .....</i>	<i>12</i>
1.4. Контроллеры семейства микро ПЛК LOGO! .....	12
1.4.1. Аппаратный комплекс модулей ПЛК.....	14
1.4.2. Особенности выбора и разработки конфигурации.....	18
1.4.3. Сетевые структуры ПЛК .....	28
<i>Вопросы для самоконтроля .....</i>	<i>34</i>

**Глава 2. ИНСТРУМЕНТАЛЬНАЯ СРЕДА**

<b>РАЗРАБОТКИ ПРОГРАММ.....</b>	<b>35</b>
2.1. Пользовательский интерфейс.....	35
2.1.1. Интерфейс программирования .....	35
2.1.2. Функции окна информации .....	36
<i>Вопросы для самоконтроля.....</i>	<i>37</i>
2.2. Панели инструментов .....	37
2.2.1. Стандартная панель инструментов .....	37
2.2.2. Панель инструментов моделирования и окно состояния .....	37
2.2.3. Панель инструментов программирования.....	38
<i>Вопросы для самоконтроля.....</i>	<i>39</i>
2.3. Строка меню.....	39
2.3.1. Файл. Сохранить как .....	39
2.3.2. Файл -> Сравнение коммутационных программ.....	40

2.3.3. Файл. Преобразование РКС в ФБД.....	40
2.3.4. Файл. Преобразование ФБД в РКС.....	41
2.4. Меню «Правка» .....	41
2.4.1. «Правка -> Имена входов и выходов» .....	41
2.4.2. «Правка -> Свойства блока» .....	42
2.4.3. «Правка -> Свойства блока» (все блоки).....	42
2.4.4. «Правка -> Разрезать соединения» .....	42
2.5. Меню «Сервис» .....	43
2.5.1. Меню «Инструменты – Обзор».....	43
2.5.2. «Сервис. Определить модуль LOGO!» .....	43
2.5.3. «Сервис. Выбор оборудования» .....	43
2.5.4. «Сервис. Эмуляция» .....	43
<i>Вопросы для самоконтроля.....</i>	<i>45</i>
<b>Глава 3. ОСНОВЫ ЗАПИСИ ПРОГРАММ .....</b>	<b>46</b>
3.1. Запись программы в среде разработки .....	46
3.1.1. Разработка программы .....	46
3.1.2. Выбор блоков .....	49
3.1.3. Размещение блоков.....	50
3.1.4. Конфигурирование свойств блоков.....	51
3.1.5. Соединение блоков.....	51
3.1.6. Определение требуемого ресурса оборудования. Доступность блоков .....	53
<i>Вопросы для самоконтроля .....</i>	<i>54</i>
<b>Глава 4. РЕШЕНИЕ ПРИКЛАДНЫХ ЗАДАЧ АВТОМАТИЗАЦИИ НА ОСНОВЕ ПЛК .....</b>	<b>55</b>
4.1. Принципы разработки программ .....	55
4.1.1. Пример разработки программы системы управления .....	57
4.1.2. Алгоритм управления .....	58
4.1.3. Аппаратное решение автоматической системы регулирования .....	58
4.1.4. Блок-схема структуры управления.....	59
<i>Вопросы для самоконтроля.....</i>	<i>60</i>
4.2. Методы формализованного подхода к разработке программ .....	60

4.2.1. Методы формализации алгоритмов .....	64
4.2.2. Синтез многотактных схем на основе таблиц переходов и карт Карно .....	68
<i>Вопросы для самоконтроля</i> .....	73
4.3. Эмуляция функционирования программы.....	73
4.3.1. Авария питания .....	74
4.3.2. Выполнение оперативных тестов программ .....	75
4.3.3. Документирование программы.....	75
<i>Вопросы для самоконтроля</i> .....	76

## **Глава 5. ЯЗЫК ФУНКЦИОНАЛЬНЫХ**

<b>БЛОКОВЫХ ДИАГРАММ</b> .....	<b>77</b>
5.1. Постоянные и соединители. ФБД.....	77
5.1.1. Входы.....	77
5.1.2. Клавиши управления курсором .....	77
5.1.3. Выходы .....	78
5.1.4. Постоянные логические уровни .....	78
5.1.5. Биты регистра сдвига .....	78
5.1.6. Флаги .....	78
5.1.7. Аналоговые входы .....	79
5.1.8. Аналоговые выходы.....	79
<i>Вопросы для самоконтроля</i> .....	80
5.2. Базовые функции языка ФБД (FBD).....	80
<i>Вопросы для самоконтроля</i> .....	81
5.3. Специальные функции языка ФБД (FBD) .....	81
<i>Вопросы для самоконтроля</i> .....	100
5.3.2. Счетчики .....	100
5.3.3. Аналоговые функции .....	107
5.3.4. Обработка аналоговых сигналов .....	124
<i>Вопросы для самоконтроля</i> .....	129
5.3.5. Управление и регулирование.....	129
5.3.6. Специальные функции .....	143
<i>Вопросы для самоконтроля</i> .....	156
5.3.7. Контроль памяти программ и данных .....	157
<i>Вопросы для самоконтроля</i> .....	160
5.3.8. Язык релейно-контактных схем (РКС) .....	160
<i>Вопросы для самоконтроля</i> .....	161

<b>Глава 6. ПРИМЕРЫ ПРОГРАММ .....</b>	<b>162</b>
6.1. Пример №1. «Управление доступом» .....	162
6.2. Пример №2. «Контроль мест для стоянки автомобилей» .....	163
6.3. Пример №3. «Управление загрузкой бункера» .....	164
6.4. Пример №4. «Управление бетономешалкой» .....	166
6.5. Пример №5. «Управление подъемной платформой» .....	167
6.6. Пример №6. «Наполнительная станция» .....	169
6.7. Пример №7. «Освещение и сигнализация жилого дома» .....	171
6.8. Пример №8. «Контроль длительности подключения потребителей (нагрузки)» .....	173
6.9. Пример №9. «Система кондиционирования воздуха» .....	175
6.10. Пример №10. «Базовые арифметические операции» .....	176
6.11. Пример №11. «Управление отоплением. Регулирование» .....	178
 <b>ПРИЛОЖЕНИЕ 1 .....</b>	 <b>181</b>
<b>ПРИЛОЖЕНИЕ 2 .....</b>	<b>182</b>
 <b>СПИСОК ЛИТЕРАТУРЫ .....</b>	 <b>183</b>

Учебное издание

**МАКСИМЫЧЕВ** Олег Игоревич  
**ЛИБЕНКО** Александр Владимирович  
**ВИНОГРАДОВ** Вадим Алексеевич

ПРОГРАММИРОВАНИЕ  
ЛОГИЧЕСКИХ  
КОНТРОЛЛЕРОВ (PLC)

УЧЕБНОЕ ПОСОБИЕ

*Редактор И.А. Короткова*

Подписано в печать 21.10.2016 г. Формат 60×84/16.  
Усл. печ. л. 11,75. Тираж 300 экз. Заказ . Цена 385 руб.  
МАДИ, 125319, Москва, Ленинградский пр-т, 64.