

АКЦИОНЕРНОЕ ОБЩЕСТВО ЗАКРЫТОГО ТИПА

"КОНСТАР"

Описание языка контроллеров "Констар"

АЛ2.598.020Д2

(Редакция 01)

2004 год

Содержание

Принятые сокращения.....	3
1. Программируемые логические контроллеры	
1.1. Общие сведения.....	4
1.2. Функции.....	4
2. Организация обмена информацией	
2.1. Организация памяти.....	6
2.2. Таблица данных.....	6
2.3. Рабочая программа.....	8
2.4. Организация обмена	10
3. Инструкции рабочей программы	
3.1. Логические инструкции РП.....	11
3.2. Инструкции структурирования.....	14
3.3. Подпрограммы.....	19
3.4. Таймеры и счетчики.....	20
3.5. Работа с данными.....	30
3.6. Арифметические инструкции.....	34
3.7. Инструкции сравнения.....	36
3.8. Файловые инструкции.....	38
3.9. Преобразование битового значения слова в символы.....	41
4. Специальные функции	
4.1. Работа с модулями АЦП.....	43
4.2. Работа с модулями ЦАП.....	44
4.3. Работа с модулями ввода импульсных сигналов CP34.23,CP34.25.....	44
4.4. Работа с панелью оператора К921.....	48
4.5. Работа с двухканальным модулем связи (CP52.05).....	48
4.6. Работа с коммуникационным каналом "RS485"либо "R232"	52
4.7. ПИД регулятор.....	55
4.8. Экстренное обращение к модулям ввода (вывода).....	58
4.9.Операции с вещественными числами.....	59
Приложение 1. Список инструкций ПЛК "Констар".....	61
Приложение 2. Таблица адресов служебных регистров CP59.10.....	63
Приложение 3. Пример типового проекта	66

Принятые сокращения:

ЦП - центральный процессор;

ПЛК - программируемый логический контроллер;

РП - рабочая программа;

ТД - таблица данных;

КС - контрольная сумма;

ОЗУ - оперативное запоминающее устройство;

ППЗУ - перепрограммируемое постоянное запоминающее устройство;

РПО - резидентное программное обеспечение;

БЦС - буквенно-цифровые сообщения;

БВ - Булевый вычислитель;

ПЛК - Программируемый логический контроллер.

1. Программируемые логические контроллеры

Настоящий документ распространяется на программируемые логические контроллеры серии “Констар”.

В документе приведено подробное описание языка релейно-контактных схем программируемых контроллеров, форматы представления данных и правила составления программы. Инструментальным средством для создания рабочих программ служит *Система автоматизированного программирования К748*. Наличие такого мощного средства позволяет и не специалистам в области программирования, создавать программы для контроллера.

1.1. Общие сведения

Программируемый логический контроллер представляет собой специализированную электронную вычислительную машину, предназначенную для управления технологическими процессами и различным оборудованием в реальном масштабе времени.

ПЛК и их модификации являются универсальными техническими средствами, позволяющими в кратчайшие сроки, методом свободной компоновки, создавать технические комплексы для различных объектов управления.

1.2. Функции

Автоматизация объекта управления (любого технологического оборудования) заключается в обеспечении его работы с помощью устройства управления. Созданная таким образом система функционирует на основании состояний параметров, из которых состоит объект управления. Оператор осуществляет лишь общий контроль процесса управления и при необходимости берет на себя управление работой всей системы или ее части.

В работе автоматизированной системы управления объектом используются два типа обмена информацией: постоянный обмен между объектом и комплексом средств управления (входные и выходные данные); обмен по мере необходимости между оператором и системой управления (назначение операций по обработке данных, изменение уставок, частичное или полное переключение управления на себя и др.).

Для осуществления этих обменов информацией необходимы языки, которые были бы понятны для передатчика и приемника сообщений.

Язык – это способ однозначного представления информации.

ПЛК выполняет три основные функции:

- прием сигналов с каналов модулей ввода;
- принятие решений;
- выдача сигналов на каналы модулей вывода.

Информация представлена в виде дискретных, аналоговых, импульсных сигналов, поступающих от различного рода датчиков, являющихся источником информации о состоянии оборудования, положении механизмов и органов ручного управления.

Обработка информации и принятие решения производится ПЛК. Он постоянно опрашивает входные сигналы, поступающие от входных устройств, и производит логические операции с этими сигналами по программе, записанной в ПЛК пользователем. Действия ПЛК состоят во включении и выключении каналов дискретных выходных и аналоговых модулей, управляющих внешними устройствами, в качестве которых могут быть применены пускатели, устройства сигнализации и индикации, логические схемы.

Характерным свойством контроллера является быстрый циклический характер работы, который определяется их способностью вести обработку данных в **“реальном масштабе времени”**.

Автоматизация объекта управления (машины, группы машин или любого технологического оборудования) заключается в обеспечении его работы с помощью устройства управления. Созданная таким образом автоматизированная система функционирует с учетом параметров состояния объекта управления. Оператор осуществляет лишь общий контроль за ходом процесса, наблюдает за работой машин и, при необходимости, берет на себя управление работой всей системы или ее части.

Алгоритм управления задается в виде прикладных программ, которые легко заменить или модифицировать.

2. Организация обмена информацией

2.1. Организация памяти

Память состоит из областей различного функционального назначения. К основным областям относится **Таблица данных** и **Рабочая программа**.

Для запоминания инструкций, выполняющих логические операции, временные задержки, арифметическую обработку и т.д., служит **область РП** и **констант**.

Объемы **ТД** и **РП** определяются пользователем, исходя из алгоритма управления и ограничений по общему объему памяти. Часть памяти, отводимой пользователю, может оказаться неиспользованной и это не влияет на работу **ПЛК**. Каждое слово в памяти имеет полный адрес, указывающий местоположение этого слова в массиве памяти.

Пользователь в процессе программирования может использовать слова в **ТД**

- **Z0 – Z6143;**

- для **РП** - не более 32 Кбайт (**ОЗУ/ППЗУ**).

2.2. Таблица данных

Область памяти **ТД** служит для хранения состояний входных и выходных сигналов, промежуточных результатов вычислений, накопленных значений таймеров и счетчиков, используемых в процессе выполнения **РП**.

Минимальной информационной единицей, обрабатываемой процессором, является **бит**.

Бит принимает два возможных логических значения «0» или «1». Упорядоченная совокупность из восьми **бит** образует **байт**. Упорядоченная совокупность из шестнадцати **бит** образует **слово**. **Биты** в **слове** обозначаются двузначными десятичными числами от 00 до 15. Младшие восемь **бит** (00...07) образуют **младший байт**, старшие восемь бит (08...15)- **старший байт слова**.

ТД разделена на области **Z, C, T** (рис.2.2.1).

Адресуемыми объектами **ТД** являются - 16-ти разрядное **слово** или **бит**. Формат адреса пословного и побитного обращения приведен на рис. 2.2.2, 2.2.3 соответственно.

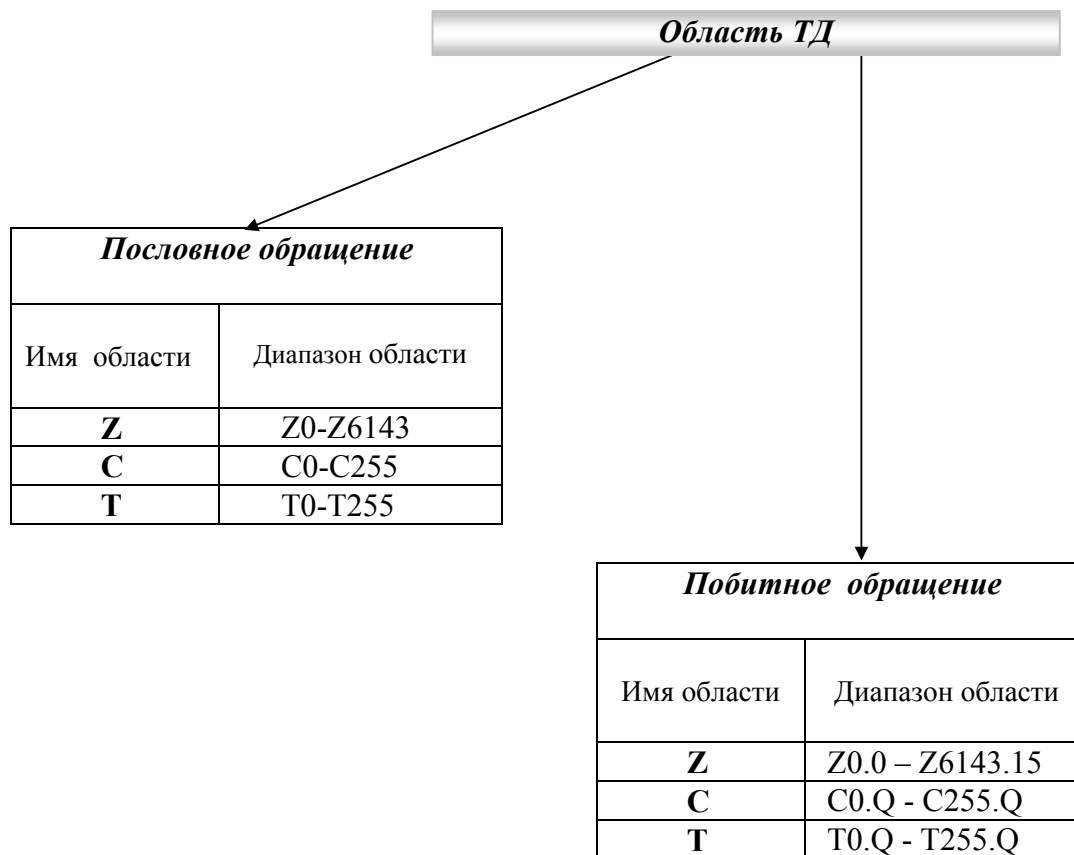


Рис. 2.2.1

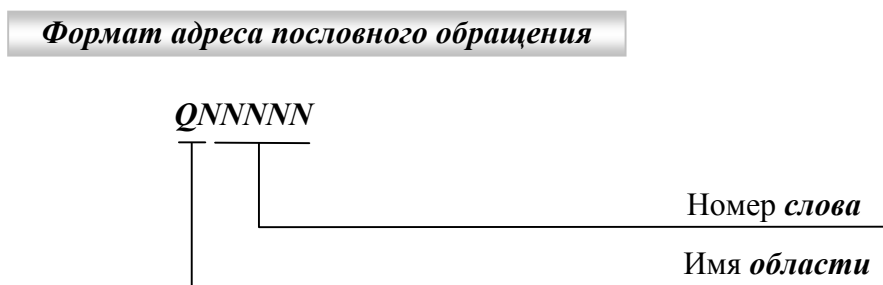


Рис. 2.2.2

Часть таблицы данных при переходе контроллера из режима “Стоп” в режим “Пуск” не обнуляется. Для CP59.10 пользователь сам определяет границы сохраняемой области.

Максимальная сохраняемая область Z400 –Z6143.

Формат адреса побитного обращения

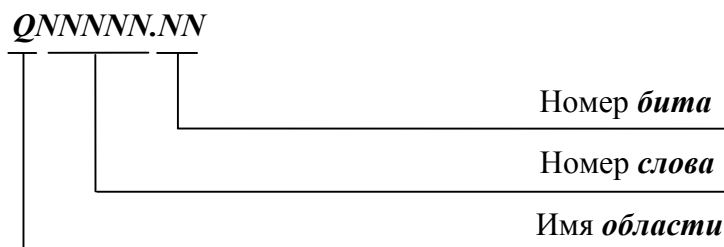


Рис. 2.2.3

В процессе работы **ПЛК** инструкции программы последовательно, начиная с первой, считываются **ЦП** из памяти **РП** и выполняются, при этом изменяется состояние **ТД**.

После выполнения последней инструкции, происходит обмен информацией между определенной областью **ТД** - с одной стороны и модулями ввода-вывода дискретных сигналов - с другой. Эти две фазы работы непрерывно циклически в **контроллере** повторяются.

2.3. Рабочая программа

РП – это упорядоченная последовательность инструкций рабочей программы пользователя.

Инструкция- это распоряжение, выполняемое **ЦП** (об инструкциях **РП** подробно изложено в главе 3).

Область **РП** состоит из **области конфигурации**, области **РП** пользователя, **области констант пользователя** и **области буквенно-цифровых сообщений**.

Количество инструкций в **РП** (длина **РП**) зависит от реализуемого алгоритма, но выходной файл не должен превышать **32 700 байт**.

В зависимости от установленной микросхемы **РП** может записываться сервисными устройствами в ОЗУ или во Flash.

Формат области конфигурации приведен на рис.2.3.1.

Формат области памяти рабочих программ

P0	Контрольная сумма рабочей программы пользователя
P1	Размер сохраняемой области таблицы данных пользователя
P2	Блокировка перехода ПК в режим “ОТЛ. СТОП” при отказе модуля в основном каркасе (места 00 – 15)
P3	Блокировка перехода ПК в режим “ОТЛ. СТОП” при отказе модуля в каркасе расширителя (места 00 – 15)
P4	Адрес ПК в сети обмена по протоколу MODBUS (пассивная станция)
P5	Скорость обмена по протоколу MODBUS
P8	Адрес начала массива буквенно-цифровых сообщений
P9	Количество кадров буквенно-цифровых сообщений
P10	Начальный адрес буфера обмена с панелью в таблице данных ПК
P15	Конфигурация модулей связи CP52.05
P16	Конфигурация модулей АЦП основного каркаса
P17	Конфигурация модулей ЦАП основного каркаса
P18	Конфигурация модулей АЦП каркаса расширителя
P19	Конфигурация модулей ЦАП каркаса расширителя
P20	Конфигурация модулей ввода дискретных сигналов основного каркаса (00-07)
P21	Конфигурация модулей вывода дискретных сигналов основного каркаса (00-07)
P22	Конфигурация модулей ввода дискретных сигналов основного каркаса (08-15)
P23	Конфигурация модулей вывода дискретных сигналов основного каркаса (08-15)
P24	Конфигурация модулей ввода дискретных сигналов каркаса расширителя (00-07)
P25	Конфигурация модулей вывода дискретных сигналов каркаса расширителя (00-07)
P26	Конфигурация модулей ввода дискретных сигналов каркаса расширителя (08-15)
P27	Конфигурация модулей вывода дискретных сигналов каркаса расширителя (08-15)
P28	Конфигурация расширителя
P29	Конфигурация модема CP52.03
P30	Блокировка РИ, время цикла
P31	РЕЗЕРВ
P32	РЕЗЕРВ
P63	
P64	Рабочая программа пользователя, таблицы констант пользователя, таблица буквенно-цифровых сообщений пользователя
P16371	

Рис.2.3.1

2.4. Организация обмена

В ПЛК используется три вида обмена информацией. Обмен между процессором и дискретными модулями, регистровый обмен и обмен по протоколу Modbus с внешними устройствами.

Обмен с модулями ввода (вывода) дискретных или аналоговых сигналов должен осуществляться либо по конфигурации модулей (*системный обмен*) либо, так называемый регистровый обмен, при помощи *экстренного обращения к модулю* ввода (вывода) (инструкции - *ЧТРм* или *ЗПРм*).

Адресация входов (выходов) модулей производится по месту установки модулей в каркасе

Конфигурация модулей – это информация о наличии и месте установки дискретных или аналоговых модулей в каркасе, содержащем до 16 установочных мест (одноблочный вариант) либо 32 установочных места (двухблочный вариант).

Внимание! *Системный обмен и обмен при помощи экстренного обращения к модулю ввода (вывода) не допускается использовать в РП одновременно.*

В обмене участвуют восьмиканальные, шестнадцати канальные модули ввода (вывода) дискретных сигналов и модули ввода (вывода) аналоговых сигналов.

При *системном обмене* адресация входов (выходов) дискретных сигналов производится с привязкой к установочному месту модуля в соответствующем каркасе.

Обмен с дискретными модулями при помощи экстренного обращения к модулю ввода (вывода) осуществляется по инструкциям *ЧТРм* или *ЗПРм*.

3. Инструкции рабочей программы

3.1. Логические инструкции РП

Логические инструкции используются для опроса состояния битовых объектов ТД, а именно – дискретных входов, выходов таймеров, счетчиков, а также для управления состоянием выходных битовых объектов.

Программы, записанные на релейном языке, представляют собой последовательность релейных схем.

Нормально разомкнутый контакт (Прямой опрос битового объекта).



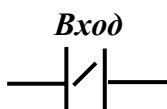
Когда входное устройство находится в замкнутом состоянии, например, включен конечный выключатель, индицируется включенное состояние. Это состояние передается в соответствующий *бит* памяти. Когда в программе записана инструкция опроса на включение с этим адресом, ЦП определяет, что входное устройство включено и считает инструкцию *истинной*. Если входное устройство размыкается, ЦП считает эту инструкцию *ложной*.

Если цепь содержит только входную и выходную инструкцию, то выходной *бит* принимает значение равное «1» при истинном значении входной инструкции.

Если же входная инструкция ложна (вход разомкнут), то выход отключается.

Если цепь содержит несколько входных инструкций, то выходной *бит* принимает значение равное «1», когда может быть найден, по крайней мере, один замкнутый путь, из логически истинных инструкций, ведущий от шины питания к выходной инструкции. Если такого пути не существует, то отключаются.

Нормально замкнутый контакт (Инверсный опрос битового объекта).



Если входное устройство не замкнуто, то входной зажим, к которому это устройство присоединено, индицирует выключенное состояние. Когда в программе встречается инструкция

инверсного опроса, ЦП определяет, что входное устройство выключено и считает инструкцию *истинной*. Если входное устройство замыкается, то ЦП считает инструкцию *ложной*.

Если цепь содержит только инструкцию инверсного опроса и выходную инструкцию, то выходной бит принимает значение равное «1», когда инструкция инверсного опроса находится в истинном состоянии, т.е. вход отключен. Если инструкция ложна, то выход отключается.

Катушка нейтрального реле (Установить битовый объект).

Выход

—()—

Инструкция включения выхода используется для включения отдельных *битов* памяти. Если адрес *бита* соответствует адресу зажима выходного модуля, то выходное устройство, соединенное с этим зажимом, также включается. Состояние включения или выключения выходного *бита* определяется логическим состоянием цепи в РП пользователя.

Если в цепи существует замкнутый путь из логически истинных входных инструкций, то выходная инструкция принимает значение равное «1» (включается). Если такого пути нет, то условия цепи ложны и выходная инструкция отключается.

Инструкция включения выхода подобна катушке реле. Выходная инструкция управляется предшествующими ей инструкциями в программной цепи. Катушка реле управляется контактами в электрической цепи. Замкнутый путь из логически истинных инструкций соответствует электрической цепи из замкнутых электрических контактов.

Любой *бит* ТД может быть опрошен любой инструкцией опроса столько раз, сколько необходимо. Таким образом, *бит*, управляемый выходной инструкцией, также может быть опрошен необходимое количество раз.

Установочная катушка поляризованного реле (Установить битовый объект и зафиксировать).

Выход

—(S)—

Инструкции включения и выключения выхода являются *сохраняемыми* выходными инструкциями. Они обычно используются в паре для управления битом ТД.

Если адрес инструкции *включения* **-(S)-** соответствует адресу зажима выходного модуля, то выходное устройство, соединенное с этим зажимом, включается при включении *бита*, что определяется условиями его цепи.

Если в цепи может быть найден замкнутый путь из логически истинных инструкций, то инструкция включения выхода **включает** управляемый ею **бит**. Когда условия в цепи перестают быть истинными, **бит**, управляемый инструкцией включения остается **включенным**, если он перед этим был включен.

Сбросовая катушка поляризованного реле (Сбросить битовый объект и зафиксировать).

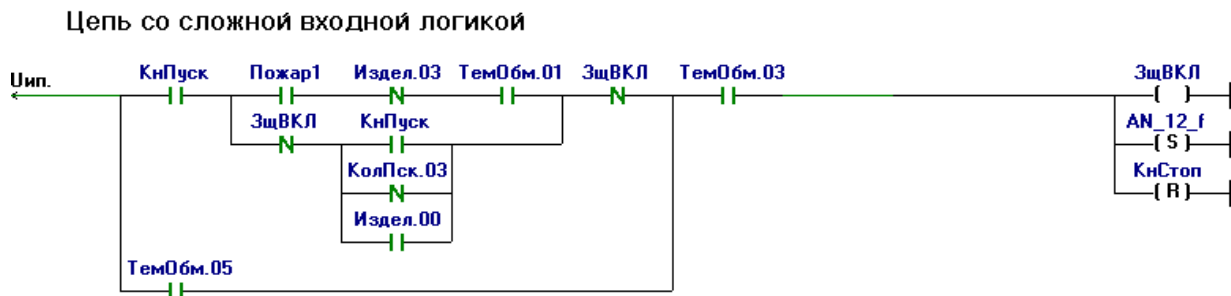
Выход

—(R)—

Инструкция **выключения** **-(R)-**, имеющая тот же адрес, что и инструкция включения, используется для выключения бита памяти. Когда в цепи устанавливается замкнутый путь из логически истинных инструкций, инструкция выключения **выключает** управляемый ею бит.

Цепь - это соединение элементов релейных схем.

РП пользователя может опрашивать **бит**, управляемый инструкциями включения и выключения, произвольное количество раз.



Инструкция **—|P|—** Контакт – Передний фронт импульса, то есть результат опроса контакта является истинным на один скан при переходе соответствующего бита из “0” в “1”.

Инструкция **—|N|—** Контакт – Задний фронт импульса, то есть результат опроса контакта является истинным на один скан при переходе соответствующего бита из “1” в “0”.

3.2. Инструкции структурирования

К инструкциям *структурирования* относятся:

- Начало сегмента (*НСТ*);
- Начало блока (*НБЛ*);
- Перейти на блок (*ПБЛ*);
- Открыть сегмент (*ОСТ*);
- Сбросить сегмент (*ССТ*);
- Закрывать сегмент (*ЗСТ*);
- Условный конец РП (*КНЦ*);
- Программный останов (*СТП*).

Внимание! *Инструкции ПБЛ, ОСТ, ССТ, ЗСТ, КНЦ, СТП являются условными.*

Начало сегмента (НСТ).

Программа подразделяется на сегменты. Начало каждого сегмента отмечается инструкцией «Начало сегмента» (*НСТ*) с соответствующей «Меткой сегмента» (номером сегмента).

Часть программы, начинающаяся инструкцией *НСТ* и заканчивающаяся инструкцией, за которой непосредственно следует инструкция этого же типа или инструкция «КОНЕЦ», называется *программным сегментом* (или *сегментом*).

В свою очередь каждый сегмент программы состоит из блоков, начало каждого из которых отмечается инструкцией «*Начало блока*».

Часть программы, начинающаяся инструкцией *НБЛ* и заканчивающаяся инструкцией, за которой непосредственно следует инструкция этого же типа, инструкция *НСТ* или инструкция «КОНЕЦ», называется программным **блоком**.

Внимание! *В ПЛК в пределах одного сегмента количество программных блоков может быть от 0 ... 64 (рис.3.2.1). Программные блоки не перекрываются. Их чередование в сегменте может быть произвольным, но нумерация должна начинаться с 00.*

Количество сегментов в программе, количество блоков в сегменте, а также порядок следования блоков внутри сегмента могут быть произвольными. В частном случае в сегменте может быть один блок, а вся программа - один сегмент. ***Программа может состоять из одного сегмента и одного блока.***

Каждый сегмент (за исключением нулевого сегмента) может быть в одном из двух состояний: ***«Открыт»*** или ***«Закрыт»***.

«Открыт» – программа активного блока этого сегмента выполняется.

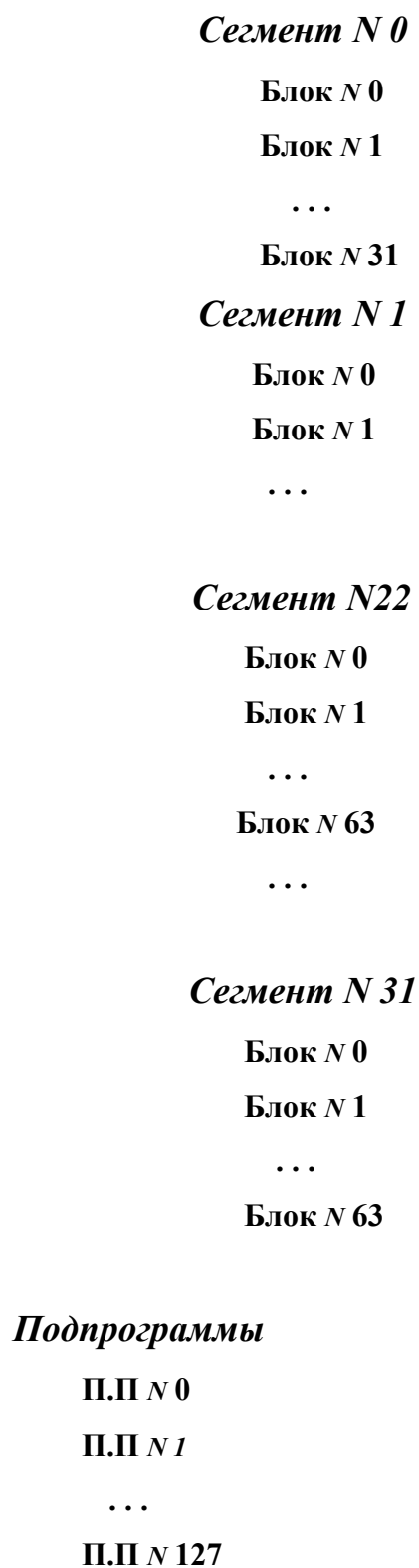
«Закрыт» - программа сегмента не выполняется.

Инструкция ***НСТ*** определяет начало сегмента программы. Все следующие за ней инструкции, до следующей инструкции ***НСТ***, ***выполняются при условии*** - данный сегмент в состоянии вычисления.

Обозначение инструкции - ***НСТ 00***, где «00» –номер сегмента.

Внимание! Нулевой сегмент всегда вычисляется. (При включении питания этот сегмент сбрасывается, то есть активным становится нулевой блок). В исходном состоянии все сегменты закрыты, кроме нулевого и активный только нулевой блок.

В следующих проходах программы, ***ЦП*** при переходе от сегмента к сегменту пользуется информацией ***области задания текущего состояния.***

**Рис. 3.2.1**

Начало блока (НБЛ)

Инструкция определяет начало программного блока в сегменте. Все следующие за ней инструкции, за исключением инструкций **НСТ**, **НБЛ**, **выполняются при условии** - данный блок активный.

Каждый блок сегмента может находиться в одном из двух состояний: **активном** или **пассивном**.

На протяжении одного прохода программы в каждом сегменте может быть только один активный блок. При выключении питания **ПЛК** во всех сегментах программы устанавливаются активными нулевые блоки. Информацию о том, какой блок активный в данном сегменте, **ЦП** получает из области текущего состояния.

Внимание! Программа выполняется только в активном блоке.

Обозначение инструкции – **НБЛ 00**, где «00» – номер блока или

Блок N 0

Перейти на блок (ПБЛ)

Если цепь, содержащая инструкцию **ПБЛ**, истинная, **ЦП** записывает в область текущего состояния в байт, соответствующий обрабатываемому сегменту, номер нового активного блока, указанный в операндной части инструкции **ПБЛ**. Программный блок, в котором была записана инструкция **ПБЛ**, переходит в состояние «Пассивный».

Если цепь, предшествующая **ПБЛ**, ложная, **ЦП** не выполняет инструкцию **ПБЛ**, а выполняет следующие за ней инструкции.

Если инструкция **ПБЛ** не выполняется, то номер активного программного блока может измениться только при выполнении инструкции «Сбросить сегмент».

Обозначение инструкции - **ПБЛ 01**, где «01» –номер блока.

Внимание! В любой момент времени в каждом программном сегменте, независимо от его состояния, только один программный блок может находиться в состоянии «Активный», остальные – в «Пассивном».

Открыть сегмент ОСТ (00)

Когда цепь, содержащая инструкцию **ОСТ**, истинная, то сегмент, указанный в операндной части, устанавливается в состояние вычисления. В байте области текущего состояния данного сегмента бит шесть и семь принимают истинное значение.

Обозначение инструкции – **ОСТ (01)**, где «01» – номер сегмента.

Закреть сегмент ЗСТ (00)

Когда цепь, предшествующая инструкции **ЗСТ**, истинная, то **ЦП** устанавливает в состояние игнорирования тот сегмент, который указан в операндной части инструкции (за исключением нулевого сегмента), не изменяя номера его активного программного блока.

Обозначение инструкции – **ЗСТ (01)**, где «01» – номер сегмента.

Условный конец РП (КНЦ)

Когда цепь, содержащая инструкцию **КНЦ**, истинная, **ЦП** переходит на новый цикл выполнения программы, *игнорируя исполнение инструкций*, находящихся после **КНЦ**.

В следующих проходах программы, повторяется цикл работы **РП**: от начала программы до инструкции **КНЦ**. Так повторяется до тех пор, пока цепь не станет ложной.

Обозначение инструкции – **КНЦ**.

Программный останов (СТП)

В случае необходимости *экстренного останова программы* применяют инструкцию **СТП**.

Когда цепь, содержащая инструкцию **СТП**, истинная, **ЦП** останавливает выполнение **РП**. Дальнейшая работа возможна после принятия решения оператором или изменения условия выполнения логической цепи.

Обозначение инструкции – **СТП**.

3.3. Подпрограммы

Одинаковые фрагменты в программе можно организовать, как одну подпрограмму и вызывать при необходимости. Обращение к подпрограмме (вызов) находится в теле **РП**, а подпрограмма располагается после конца **РП**.

Внимание! В подпрограмме не допускается использование инструкции структурирования.

Вызвать подпрограмму (ВПП)

Когда цепь, содержащая инструкцию **ВПП**, становится истинной, **ЦП** вычисляет подпрограмму с соответствующим номером, находящуюся в конце **РП**, выполняет ее и возвращается на выполнение инструкции, следующей за инструкцией **ВПП**.

Количество подпрограмм – **00...127**

Обозначение инструкции – **ВПП 01**, где «**01**» – номер подпрограммы.

Истина

— ППр.1
{ВПП }

Подпрограммы

Подпрограммы - Начало массива подпрограмм.

П.П.N 0 - П.П.N 127 - подпрограммы.

Пример:



Внимание! инструкции **ВПП** соответствует инструкция **НПП** с тем же номером.

(Инструкции **НПП** и **КПП** являются инструкциями начала и конца подпрограммы).

3.4. Таймеры и счетчики

Группа инструкций таймеров и счетчиков включает следующие инструкции:

- Таймер с задержкой времени на включение - *ТВД*;
- Таймер с задержкой времени на отключение - *ТОД*;
- Таймер накапливающий - *ТНД*;
- Счетчик на сложение - *СЧС*;
- Счетчик на вычитание - *СЧВ*;
- Счетчик накапливающий - *СЧЦ*.

Благодаря введению этих инструкций в ПЛК имеется возможность реализовать функции аналогичные реле времени, временным и счетным устройствам.

Внимание! Таймеры считают интервалы времени, счетчики – события по алгоритмам, определяемым логикой программы пользователя.

Максимальное число таймеров и счетчиков :

- таймеров - 256;
- счетчиков - 256 .

Каждая инструкция таймера и счетчика связана с *уставкой* –заранее заданным числом. Когда накопленное значение достигает *уставки*, включается *бит состояния*, который может быть использован для управления выходным устройством. *Уставка* и накопленное значение для таймеров и счетчиков могут изменяться от $0 \dots 65535_{10}$.

Обозначение бита состояния:

- для таймеров – *выход таймера* - *Имя Таймера.Q*;
- для счетчиков – *выход счетчика* – *Имя Счетчика.Q*.

Накопленное значение - текущее количество базовых интервалов времени, которое сосчитал таймер или количество событий, сосчитанных счетчиком.

Уставка таймера и счетчика задается константой или адресом слова, где находится уставка.

Базовые интервалы времени задаются дискретностью счета. *Дискретность счета* равна 0,1 секунды.

Внимание! Значение таймера сбрасывается в ноль, если левая часть логического выражения равна нулю.

Инструкции запуска таймеров и счетчиков используются только в правой части логического выражения.

Таймер с задержкой времени на включение - ТВД

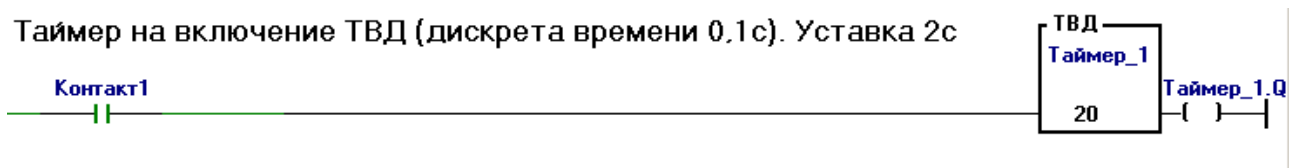
Когда условия в управляющей цепи таймера становятся истинными, таймер начинает считать с указанной дискретностью, наращивая свое накопленное значение на единицу в каждом интервале.

Когда накопленное значение достигает **уставки**, таймер останавливает счет и включает **бит завершения операции** (выход таймера), который остается включенным, пока условия в цепи включения остаются истинными.

Если условия в цепи становятся ложными до окончания отсчета времени, таймер останавливается и его накопленное значение сбрасывается в ноль.

Пример

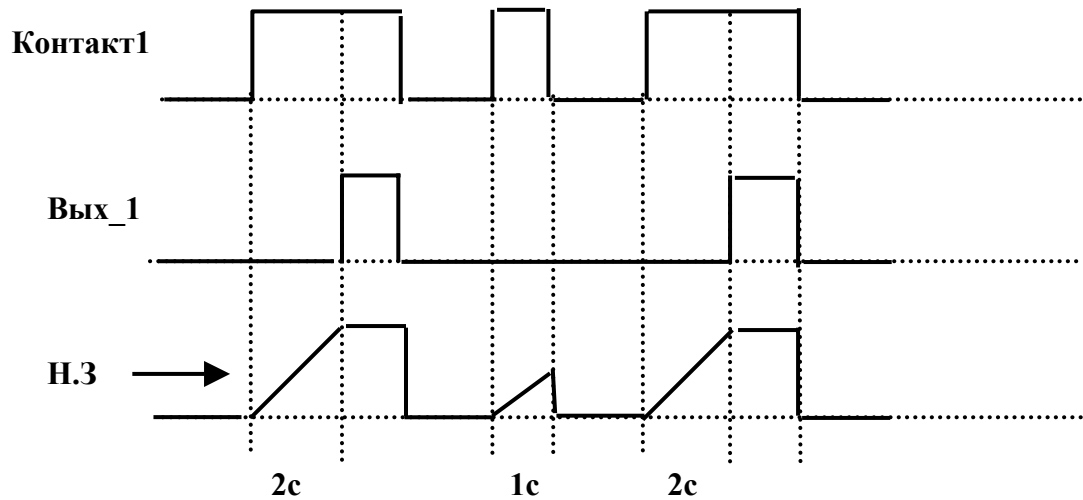
Таймер на включение ТВД (дискрета времени 0,1с). Уставка 2с



Через 2с, после вкл.таймера -Вых_1 станет = 1



Временная диаграмма:



Когда условия в управляющей цепи таймера становятся истинными (Контакт1 -замкнут), таймер начинает считать с дискретностью 0,1с, наращивая свое накопленное значение на единицу в каждом интервале. Если накопленное значение достигает уставки (2с), включается бит завершения операции –Таймер_1.Q(выход таймера).Этот бит остается включенным, пока условие в цепи остается истинным. Как только условие в цепи становится ложным (Контакт1 - разомкнут), таймер останавливается и его накопленное значение и выход таймера сбрасываются в ноль.

Таймер на отключение (ТОД)

Таймер задержки на отключение действует так, что значение его выхода равно значению входа с задержкой заднего фронта на время, заданное уставкой. Передние фронты их совпадают.

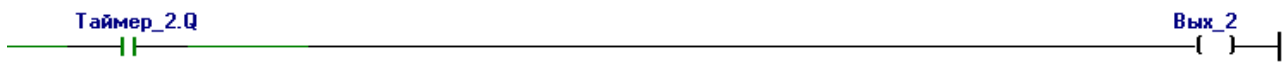
Таймер на отключение считает количество временных дискрет и хранит их счет в накопленном значении.

Пример.

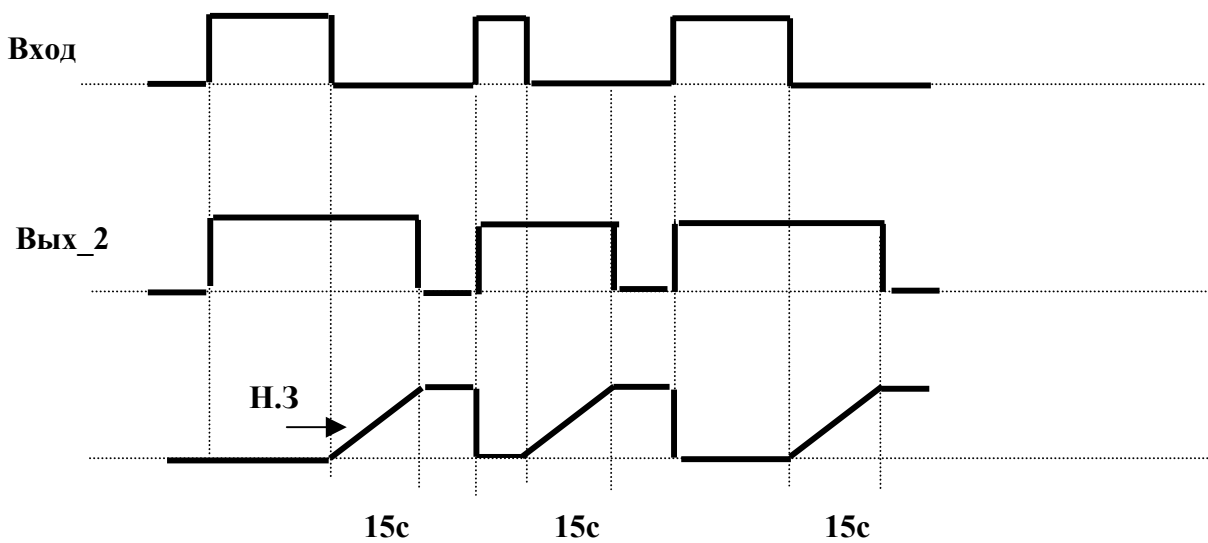
Релейно - контактная схема:



Выход таймера на отключение



Временная диаграмма:



Таймер с задержкой на отключение начинает отсчет, когда условие в его управляющей цепи становится ложным и продолжает счет, пока накопленное значение не достигнет уставки.

Таймер накапливающий

Таймер накапливающий считает дискреты времени, когда его управляющая цепь логически истинна и сохраняет свое накопленное значение, если условия в цепи остаются ложными.

Если в период работы таймера накапливающего условия в цепи становятся ложными, то это не приводит к потере накопленного значения. Когда цепь вновь станет истинной, таймер продолжит счет времени с той величины, которая была накоплена к моменту остановки. Благодаря способности сохранять накопленное значение, этот таймер может измерять суммарное время истинного состояния его цепи.

Сброс выхода таймера накапливающего производится после выполнения инструкции **СБС**, под тем же именем, что и таймер накапливающий.

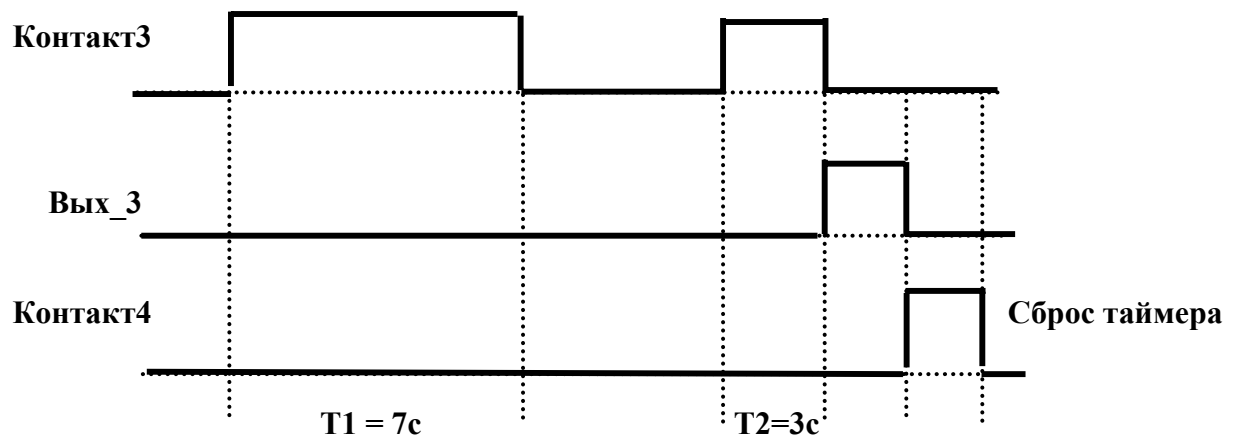
Таймер накапливающий измеряет суммарное время истинного состояния цепи.

Пример.

Релейно - контактная схема:



Временная диаграмма:



где $T = T1 + T2$ – интервал времени, равный 10 с.

Счетчик на сложение (СЧС)

Счетчик на сложение используется для счета переход левой части логического выражения из ложного состояния в истинное, при этом накопленное значение *увеличивается* на единицу.

Когда накопленное значение достигнет *уставки*, выход счетчика становится равным единице. Если на вход счетчика продолжают поступать импульсы, счет продолжается, пока не поступит команда *Сброс* счетчика.

Уставка может быть представлена константой или словом, содержимое которого и есть уставка.

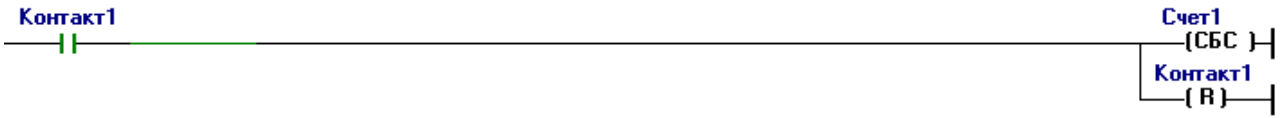
Инструкция счета на сложение предназначена для счета до своей уставки.

Накопленное значение счетчика *сохраняется* при ложном состоянии цепи, а также при потере питания. *Сброс* счетчика происходит после выполнения инструкции *СБС*, имеющей то же имя, что и счетчик.

Пример.

Релейно - контактная схема:

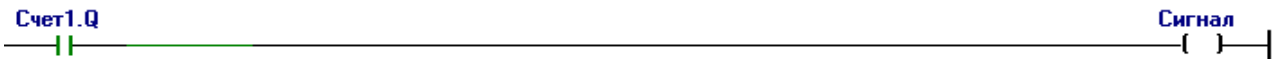
Сброс счетчика



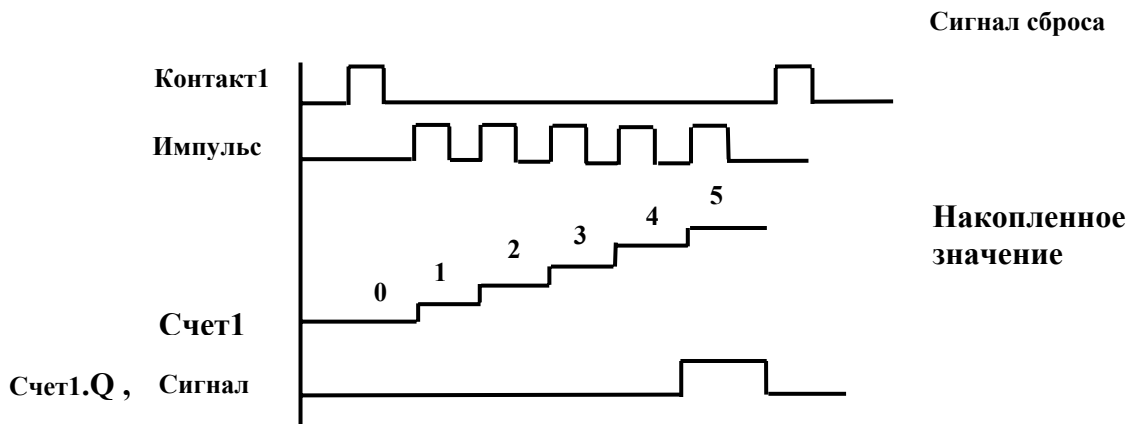
Счетчик на сложение с уставкой =5



Выход счетчика



Временная диаграмма:



Счетчик на вычитание (СЧВ)

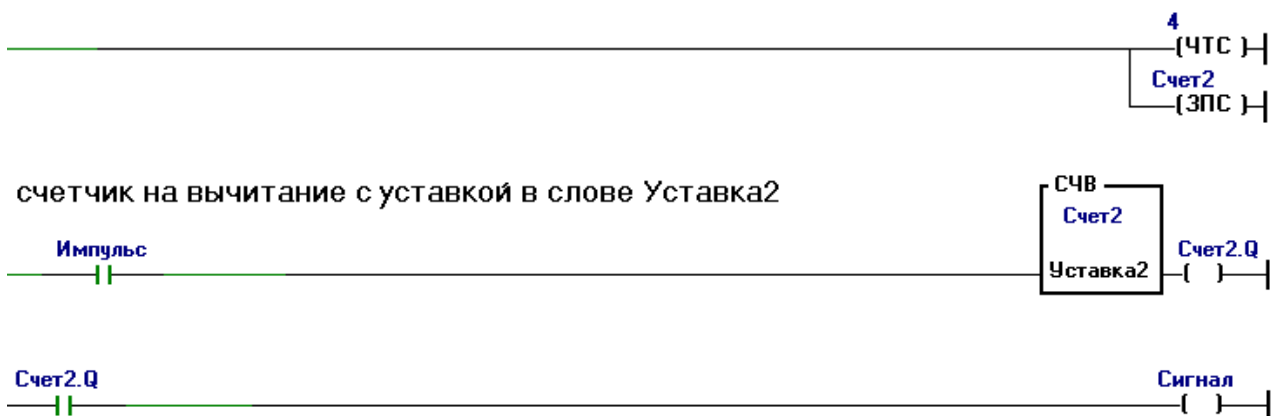
Инструкция имеет два параметра –адрес счетчика и уставку. Уставка может задаваться константой или словом, содержимое которого и есть уставка.

Когда цепь с инструкцией счета переходит из состояния ложного в истинное, накопленное значение *уменьшается* на единицу.

Когда накопленное значение достигнет *уставки*, выход счетчика устанавливается в единицу.

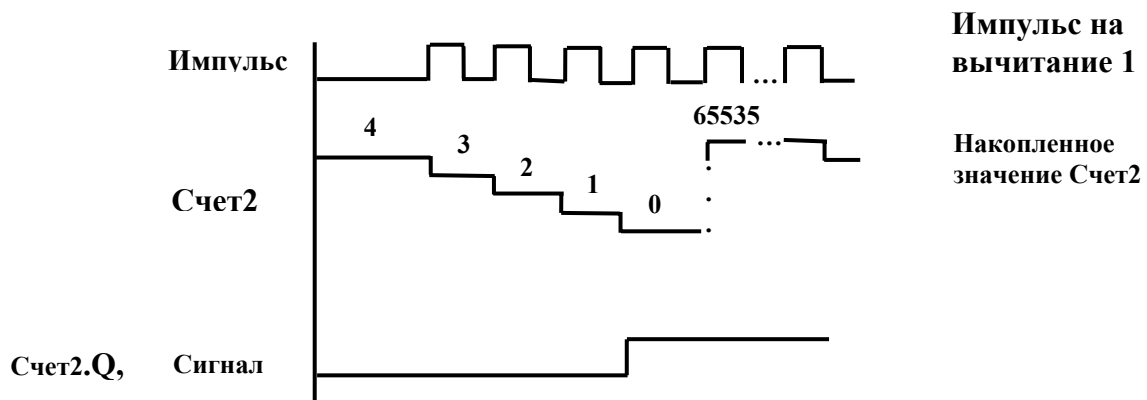
Пример.

Релейно - контактная схема:

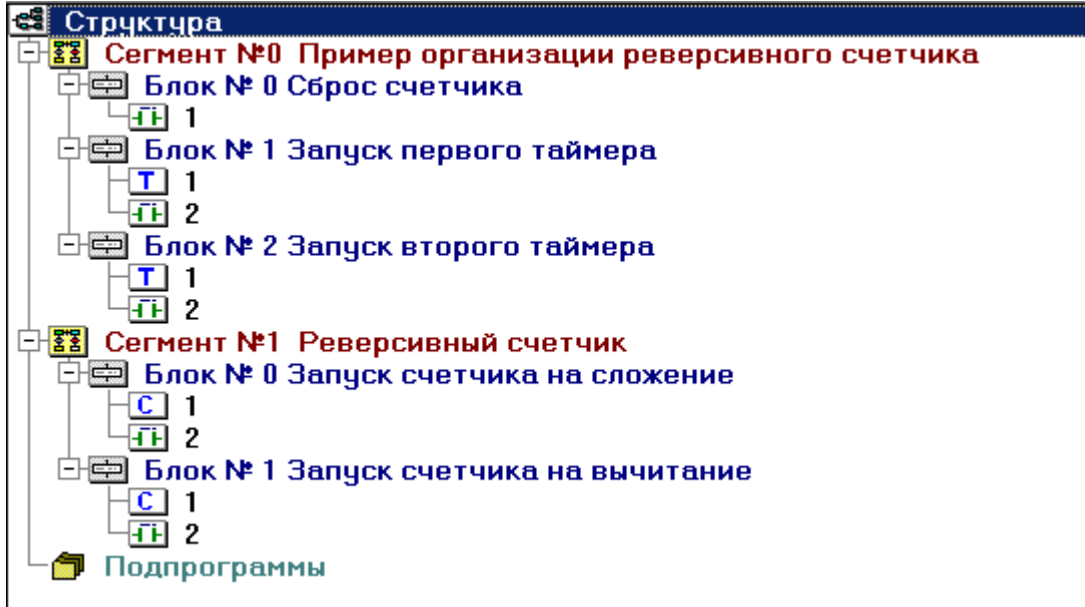


(Значение Счет2 до начала работы программы равно 4,содержимое Уставка2 =0)

Временная диаграмма:



Для организации *реверсивного счетчика* необходимо счетчик на сложение и счетчик на вычитание запускать на одном слове (они должны *работать с одинаковым именем счетчика и значением константы, но с разными условиями запуска счетчиков*).

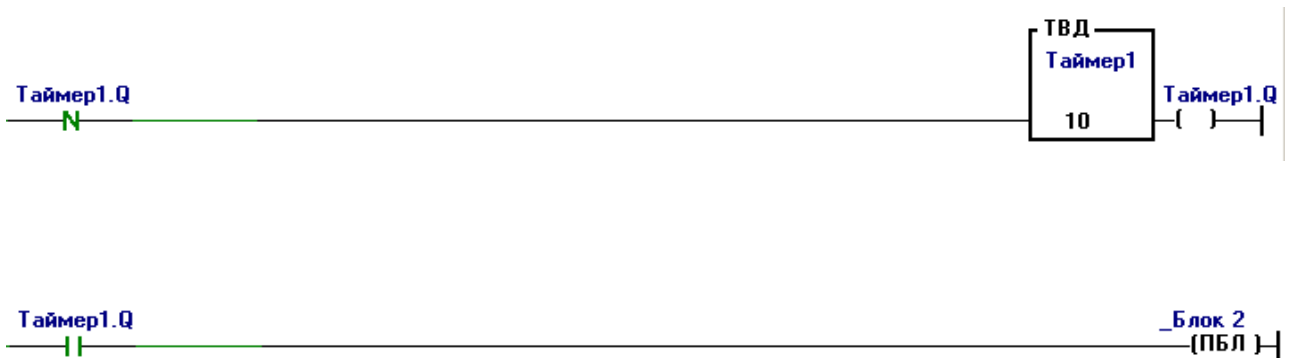


Сегмент 0

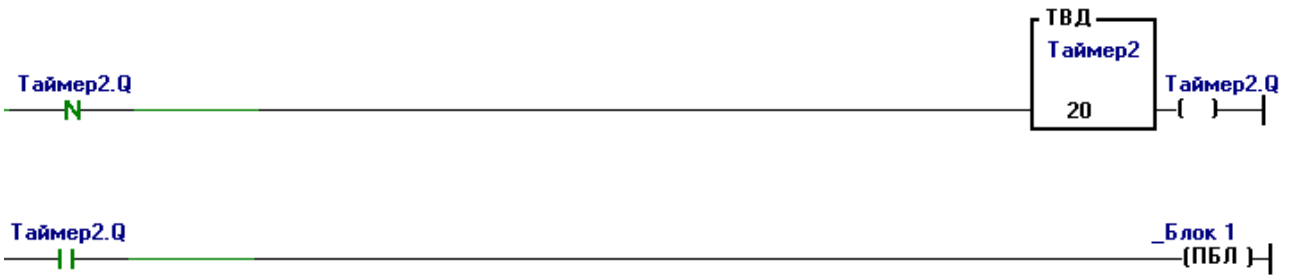
Блок 0



Блок 1

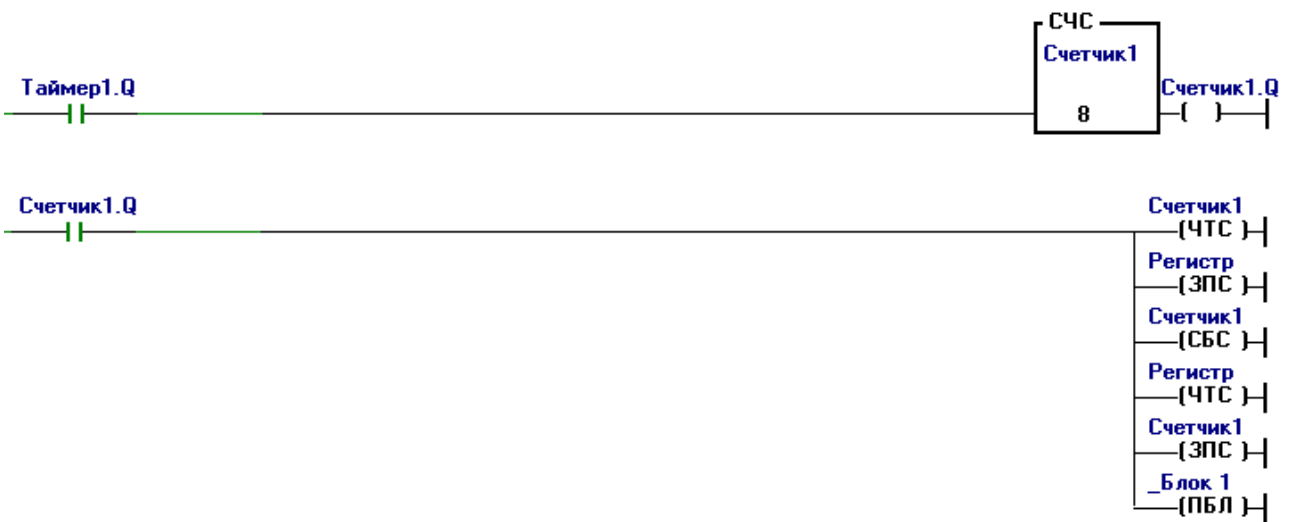


Блок 2

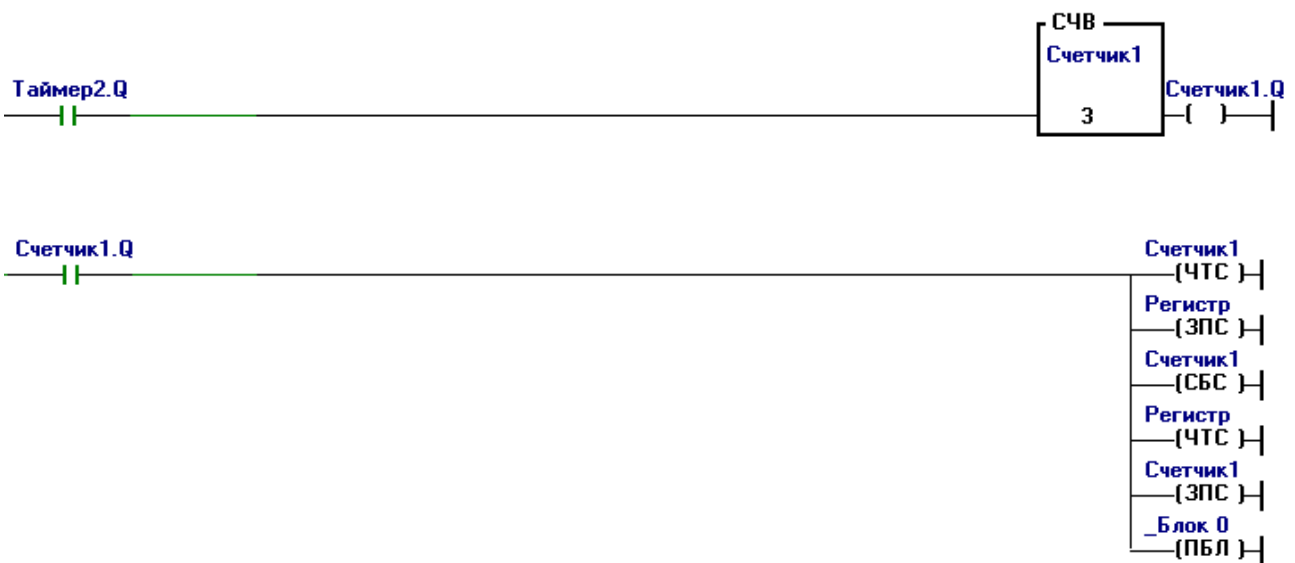


Сегмент 1

Блок 0



Блок 1



Счетчик циклов (СЧЦ)

Инструкция **СЧЦ** *увеличивает* при каждом проходе программы содержимое счетчика в случае истинного состояния логического выражения и оставляет содержимое счетчика *без изменения* при ложном состоянии логического выражения.

Внимание! *Инкремент счетчика циклов происходит в каждом цикле выполнения программы, если левая часть логического выражения равна «1». При этом инструкция СЧЦ должна находиться в открытом сегменте и активном блоке.*

Внимание! *Содержимое счетчика циклов сбрасывает в ноль инструкция СБС*

3.5. Работа с данными

Внимание! *Все инструкции пословной обработки являются условными и выходными. Если условия в цепи истинны, то инструкции исполняются.*

Инструкции для работы с данными используются при:

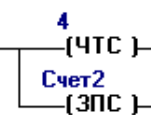
- пересылке данных из одного слова ТД в другое;
- копировании накопленных значений таймеров и счетчиков;
- маскировании отдельных битов.

Читать слово (ЧТС)

При выполнении инструкции **ЧТС** содержимое по адресу слова либо непосредственно адресуемое целое без знака число в диапазоне **0...65535** записывается в **аккумулятор**. Содержимое по адресу при этом не изменяется.

Пример.

Релейно - контактная схема



Записать слово (ЗПС)

При выполнении инструкции ***ЗПС*** содержимое аккумулятора записывается **по адресу** слова. Содержимое аккумулятора при этом не изменяется.

Читать байт старший (ЧБС)

При выполнении инструкции ***ЧБС старший байт содержимого*** по адресу слова записывается в ***старший байт аккумулятора***. Содержимое по адресу при этом не изменяется.

Читать байт младший (ЧБМ)

При выполнении инструкции ***ЧБМ младший байт содержимого*** по адресу слова пересылается в ***младший байт аккумулятора***. Содержимое по адресу при этом не изменяется.

Записать байт старший (ЗБС)

При выполнении инструкции ***ЗБС содержимое старшего байта аккумулятора*** записывается в ***старший байт*** указанного адреса. Содержимое аккумулятора при этом не изменяется.

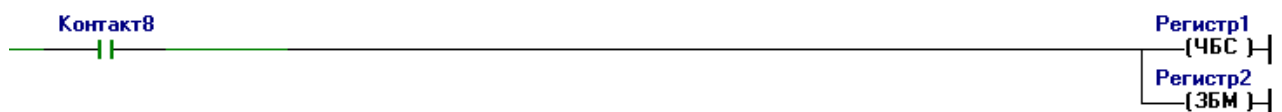
Записать байт младший (ЗБМ)

При выполнении инструкции ***ЗБМ содержимое младшего байта аккумулятора*** пересылается в ***младший байт*** указанного адреса. Содержимое аккумулятора при этом не изменяется.

Пример.

Релейно - контактная схема

Читать байт старший , Записать в байт младший



Переслать байт в байт (ПББ)

При выполнении инструкции ***ПББ*** содержимое старшего байта слова пересылается в байт младший этого слова, а содержимое младшего байта в старший байт, т.е. производится ***обмен информацией старшего и младшего байта адресованного слова.***

Сброс слова (СБС)

При выполнении инструкции ***СБС*** во все биты указанного слова записывается «0». Если в инструкции ***СБС*** указать имя таймера или счетчика, то будет сброшено накопленное значение и выход таймера или счетчика.

Сдвиг циклический к старшему биту (СЦС)

При выполнении инструкции ***СЦС*** содержимое по адресу слова сдвигается на один бит в сторону ***старшего*** разряда.

Сдвиг циклический к младшему биту (СЦМ)

При выполнении инструкции ***СЦМ*** содержимое по адресу слова сдвигается на один бит в сторону ***младшего*** разряда.

Внимание! При однократном выполнении инструкций СЦС, СЦМ происходит циклический сдвиг содержимого операнда на один разряд.

При помощи инструкций ***СЦС, СЦМ*** можно организовать ***битовый синхронный сдвиг*** внутри 16-ти разрядного слова.

Поразрядно умножить слово (ПУС)

Если условия в цепи с инструкцией ***ПУС*** истинны, то над каждой парой одноименных битов из аккумулятора и содержимого по адресу слова либо непосредственного адресуемого целого без знака числа в диапазоне ***0...65535***, производится логическая операция «***И***». Результат записывается в ***аккумулятор***.

Порядно сложить слово (ПСС)

Если условия в цепи с инструкцией ***ПСС*** истинны, то над каждой парой одноименных битов из аккумулятора и содержимого по адресу слова либо непосредственного адресуемого целого без знака числа в диапазоне ***0...65535*** производится логическая операция побитного сложения «ИЛИ».

Инкремент слова (ИНК)

Если условия в цепи с инструкцией ***ИНК*** истинны, то к содержимому слова, указанному за инструкцией ***ИНК***, прибавляется «1». Результат записывается в ***тот же адрес слова***, который указан за инструкцией.

Декремент слова (ДЕК)

Если условия в цепи с инструкцией ***ДЕК*** истинны, то из содержимого слова, указанного за инструкцией ***ИНК***, вычитается «1». Результат записывается в ***тот же адрес слова***, который указан за инструкцией.

Читать слово косвенно (ЧСК)

При выполнении инструкции ***ЧСК*** содержимое по адресу, записанному в слово инструкции, записывается в ***аккумулятор***. Содержимое по адресу при этом не изменяется.

Записать слово косвенно (ЗСК)

При выполнении инструкции ***ЗСК*** содержимое ***аккумулятора*** записывается по адресу, записанному в слово инструкции.

Пример.

Релейно - контактная схема

Блок0



Блок1



В примере каждые 100мс происходит запись константы в регистры с Z100 по Z199.

3.6. Арифметические инструкции

Арифметические инструкции:

- Сложить арифметически слово (**САС**);
- Вычесть арифметически слово (**ВАС**);
- Умножить арифметически слово (**УАС**);
- Делить арифметически слово (**ДАС**).

Эти инструкции дают возможность арифметической обработки данных. С помощью арифметических инструкций можно обрабатывать *целые положительные числа*.

При выполнении арифметических инструкций производится арифметическое действие над *содержимым аккумулятора и операнда*. Результат операции помещается в *аккумулятор*.

Все арифметические инструкции *условные*.

Сложить арифметически слово (САС)

Когда цепь, в которой находится инструкция сложения становится истинной, производится операция сложения чисел, находящихся в *аккумуляторе*, и по *адресу слова*, следующего за *САС*, либо непосредственно адресуемого целого без знака числа в диапазоне *0...65535*.

Вычитать арифметически слово (ВАС)

Когда цепь, в которой находится инструкция вычитания становится истинной, производится вычитание из чисел, находящихся в *аккумуляторе*, числа, записанного *по адресу слова*, следующего за *ВАС*, либо непосредственно адресуемого целого без знака числа в диапазоне *0...65535*.

Умножить арифметически слово (УАС)

Когда цепь, в которой находится инструкция умножения, становится истинной, производится операция умножения чисел, находящихся в *аккумуляторе*, на *содержимое слова по адресу*, следующего за *УАС*, либо на непосредственно адресуемое целое без знака число в диапазоне *0...65535*.

Делить арифметически слово (ДАС)

Когда цепь, в которой находится инструкция деления, становится истинной, производится операция деления чисел, находящихся в *аккумуляторе* на *содержимое слова по адресу*, следующего за *ДАС*, либо на непосредственно адресуемое целое без знака число в диапазоне *0...65535*.

Внимание! Если в результате арифметических операций результат вышел за пределы 16-ти разрядной сетки или получен отрицательный результат при вычитании, или было произведено деление на «0» то в соответствующий разряд Z6162 записывается «1»:

- в Z6162.0 - для операции сложения;
- в Z6162.1 - для операции вычитания.
- в Z6162.2 - для операции умножения;
- в Z6162.3 - для операции деления.

3.7. Инструкции сравнения

Инструкции сравнения:

- Сравнить на «больше» [... > ...];
- Сравнить на «меньше» [... < ...];
- Сравнить на «равно» [... = ...];
- Сравнить на «принадлежность пределам» [... <= ... <= ...].

Эта группа инструкций позволяет производить сравнение в **РП**. Инструкции работают с данными в формате *целых десятичных* чисел.

Внимание! *Инструкции сравнения используются только в левой части логического выражения (входные инструкции).*

Если цепь содержит только *инструкцию сравнения и выходную инструкцию*, то при истинности инструкции сравнения выходной бит включается, при ложном – не сохраняемый выходной бит отключается.

Если цепь содержит другие входные инструкции, кроме сравнения, то выходная инструкция включается, если можно отыскать хотя бы *один замкнутый путь* из логически истинных инструкций. Если же такого пути *нет*, то не сохраняемый выходной бит отключается, а сохраняемый - *остаётся в прежнем состоянии*.

Инструкции сравнения позволяют сравнивать слова, принадлежащие любым областям памяти. Инструкция сравнения содержит обозначение типа операции сравнения и сравниваемые слова.

Выполнение инструкций сравнения эквивалентно **вычислению значения битовой переменной**: если условие сравнения выполнилось, то битовая переменная принимает значение «1», если нет - «0».

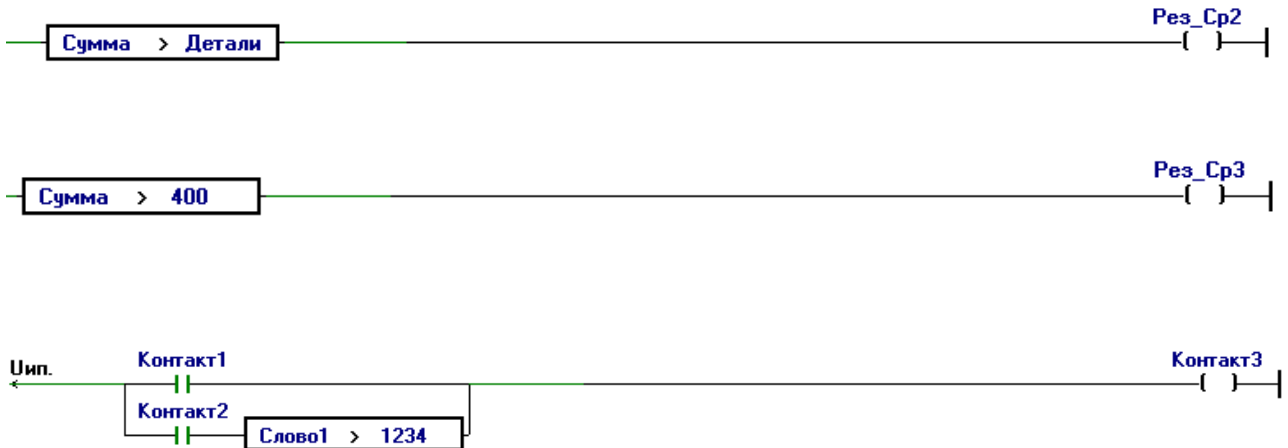
Сравнить на «больше»

Если число, хранящееся в слове, заданном слева, больше чем константа либо число, хранящееся в слове, заданном справа от обозначения кода операции, то результат сравнения равен «1», в противном случае - «0».

Пример.

Релейно - контактная схема.

Сравнить на "больше"



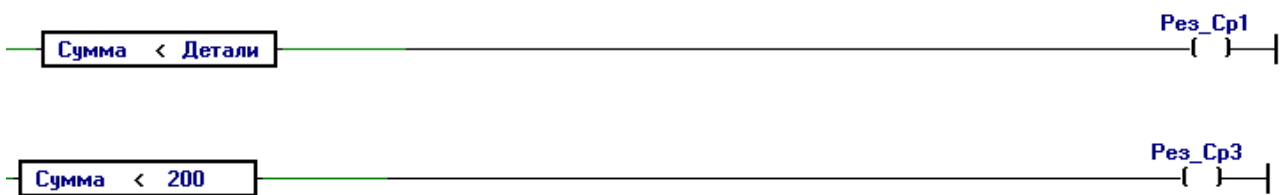
Сравнить на «меньше»

Если число, хранящееся в слове, заданном слева, меньше константа либо число, хранящееся в слове, заданном справа от обозначения кода операции, то результат сравнения равен «1», в противном случае - «0».

Пример.

Релейно - контактная схема.

Сравнить на "меньше"



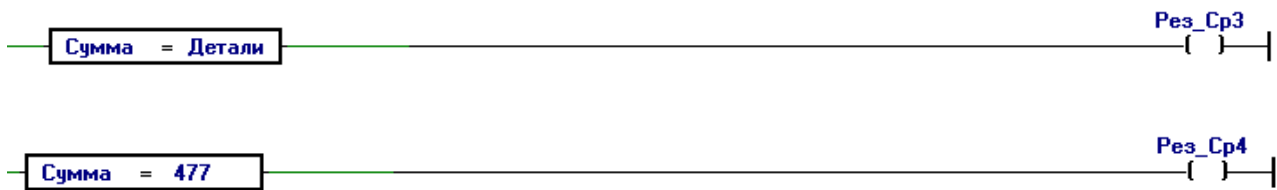
Сравнить на «равно»

Если число, хранящееся в слове, заданном слева, равно константе либо числу, хранящемуся в слове, заданном справа от обозначения кода операции, то результат сравнения равен «1», в противном случае - «0».

Пример.

Релейно - контактная схема.

Сравнить на "равно"



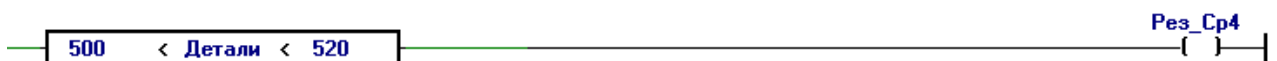
Сравнить на «принадлежность пределам»

Если число, хранящееся в слове (например, Детали, как в примере) одновременно больше или равно числу, заданному непосредственно в инструкции, либо записанному в слово, и меньше или равно числу, заданному непосредственно в инструкции, либо записанному в слово, то результат сравнения равен «1», в противном случае - «0».

Пример.

Релейно - контактная схема.

Сравнить на "принадлежность пределам"



3.8. Файловые

Файл - группа последовательных слов *ТД*, используемых для хранения информации.

Файловые инструкции - это такие инструкции, которые оперируют с файлами из *ТД*.

Файловые инструкции *условные*.

Каждая файловая инструкция использует *счетчик*. *Имя* счетчика выбирает пользователь.

Когда цепь, в которой находятся файловые инструкции, становится истинной, взводится счетчик файловой инструкции, что означает начало работы файловой инструкции.

Перед работой файловой инструкции должен быть сброшен счетчик.

Файловые инструкции:

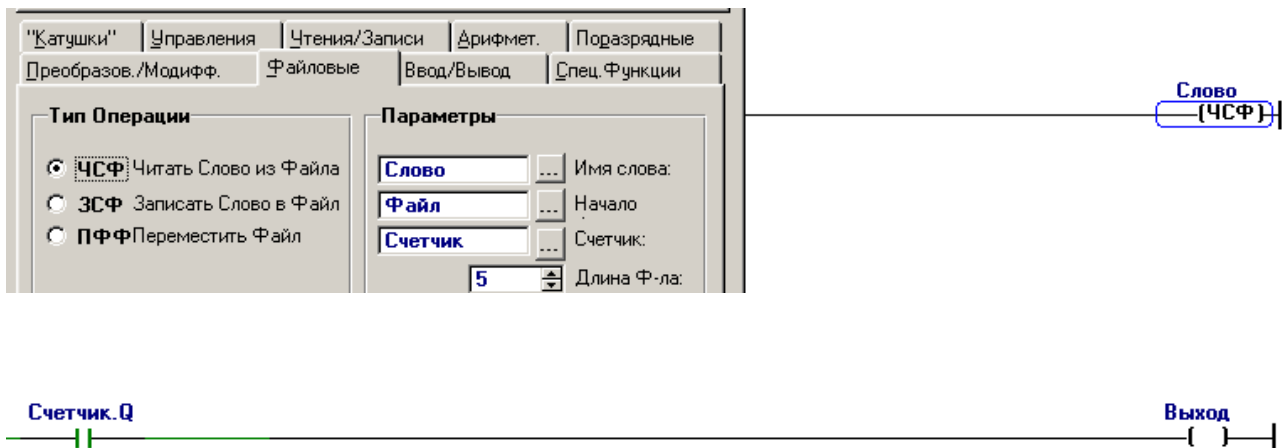
- Читать слово из файла (**ЧСФ**);
- Записать слово в файл (**ЗСФ**);
- Переместить файл (**ПФФ**).

Читать слово из файла (ЧСФ)

При изменении состояния запускающего выражения из «0» в «1» происходит перезапись содержимого слова файла по адресу слова. При следующем изменении логического выражения с «0» на «1» содержимое следующего слова файла перезаписывается *по тому же адресу слова*. Содержимое счетчика увеличивается на единицу после каждой перезаписи. Такая перезапись продолжается до того момента, пока счетчик не отсчитает количество перезаписываемых слов.

Пример.

Релейно - контактная схема



Обозначение инструкции – **ЧСФ**, где

Счётчик – адрес счетчика;

Файл – начальный адрес файла;

Слово - адрес, куда необходимо переписать слово;

5 – количество перезаписываемых слов из файла.

Выход =1 –завершение операции **ЧСФ**.

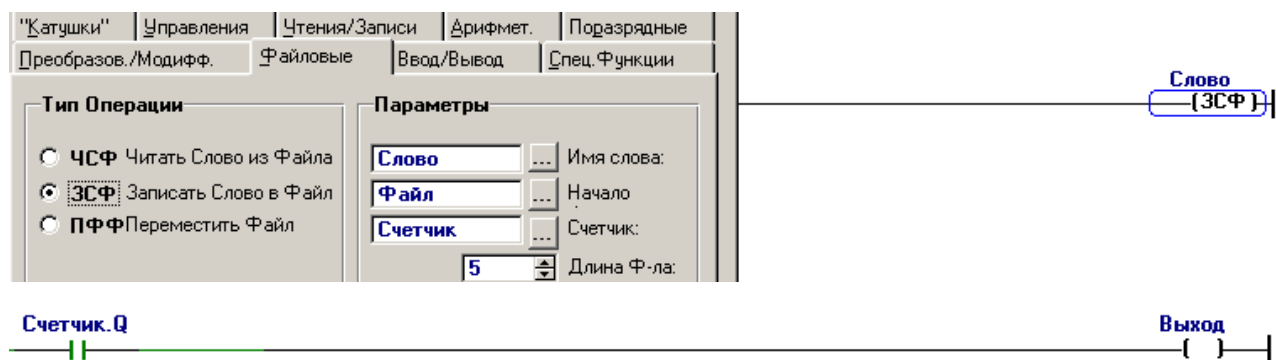
Записать слово в файл (ЗСФ)

При изменении состояния запускающего выражения из «0» в «1» содержимое по адресу слова перезаписывается в начальный адрес файла. При следующем изменении логического выражения с «0» на «1» содержимое по адресу слова записывается **в следующий адрес файла**, т.е. начальный адрес файла увеличивается на «1».

Содержимое счетчика увеличивается на единицу после каждой перезаписи, которая продолжается до того момента, пока счетчик не отсчитает количество перезаписываемых слов.

Пример.

Релейно - контактная схема



Обозначение инструкции – **ЗСФ**, где

Счѐтчик – адрес счетчика;

Файл – начальный адрес файла;

Слово - адрес, откуда необходимо переписать слово;

5 – количество перезаписываемых слов из файла.

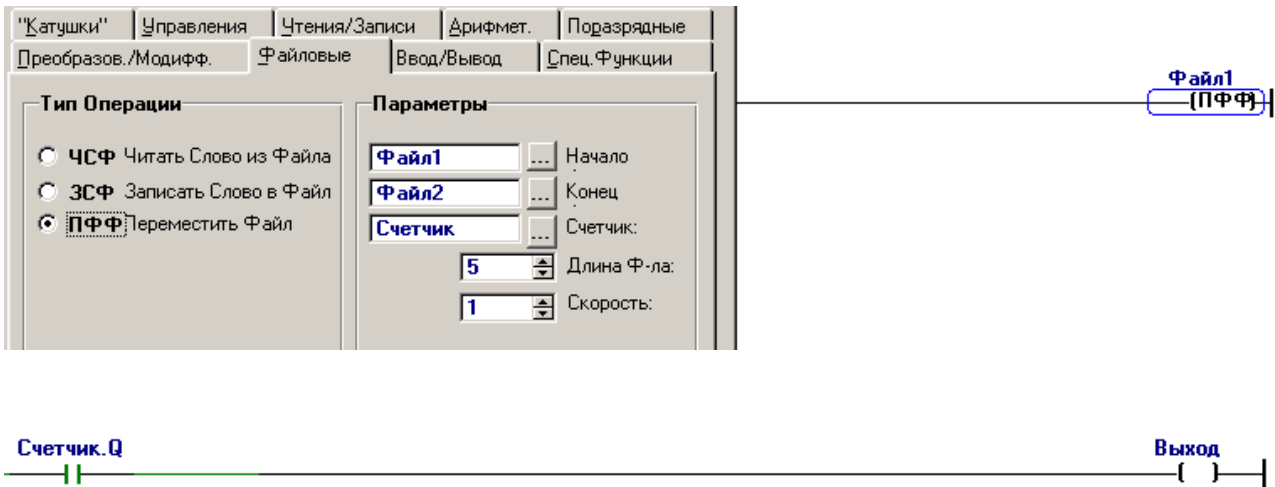
Выход = 1 –завершение операции **ЗСФ**.

Переместить файл (ПФФ)

Выполнение инструкции осуществляется пересылкой заданного количества слов за один проход программы из одного файла в другой. Перемещение производится **слово в слово**. Содержимое счетчика увеличивается после каждого перемещения, которое происходит при изменении запускающего выражения из «0» состояния в «1». Как только содержимое счетчика станет равным длине файла, файловая инструкция завершена.

Пример.

Релейно - контактная схема



Обозначение инструкции – **ПФФ**, где:

Файл1 – начальный адрес файла -источника;

Файл2 – начальный адрес файла -приемника;

Счетчик – адрес счетчика;

5 – количество слов в файле;

1 – количество слов переписываемых за один раз;

Выход =1 –завершение операции **ПФФ**.

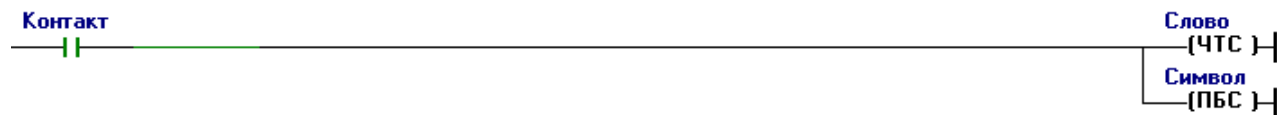
3.9. Преобразование битового значения слова в символы

Преобразование битового значения слова в символы осуществляется при помощи инструкции **ПБС**. Инструкция **ПБС** преобразовывает слова, находящиеся в аккумуляторе, и записывает их по адресам.

Внимание! При работе с инструкцией **ПБС** помнить, что запись после преобразования будет осуществляться в несколько слов, т.е. предусмотреть ячейки для записи.

Пример.

Релейно - контактная схема



Обозначение инструкции – *ПБС (Символ)*, где

Символ - начальный адрес, с которого начинается роспись битового значения слова в символы. Этот адрес необходимо установить фиксированным и зарезервировать ещё два слова для символьной информации

Если в *Слово* хранилось число **65535** в двоичном коде, то после выполнения инструкции *ПБС*, начиная со слова *Символ* хранятся символы в формате *ASCII*:

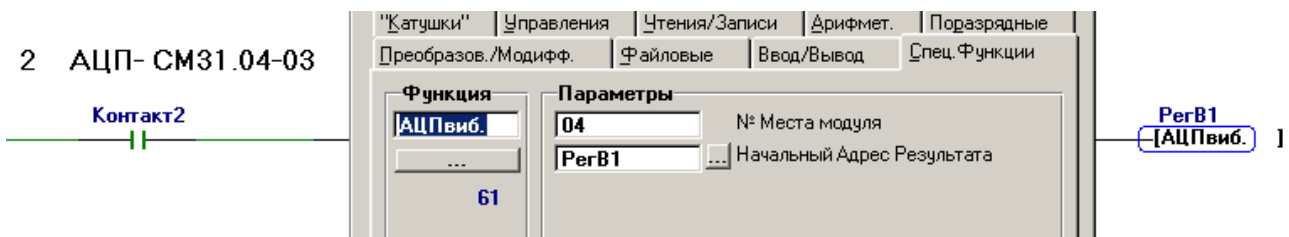
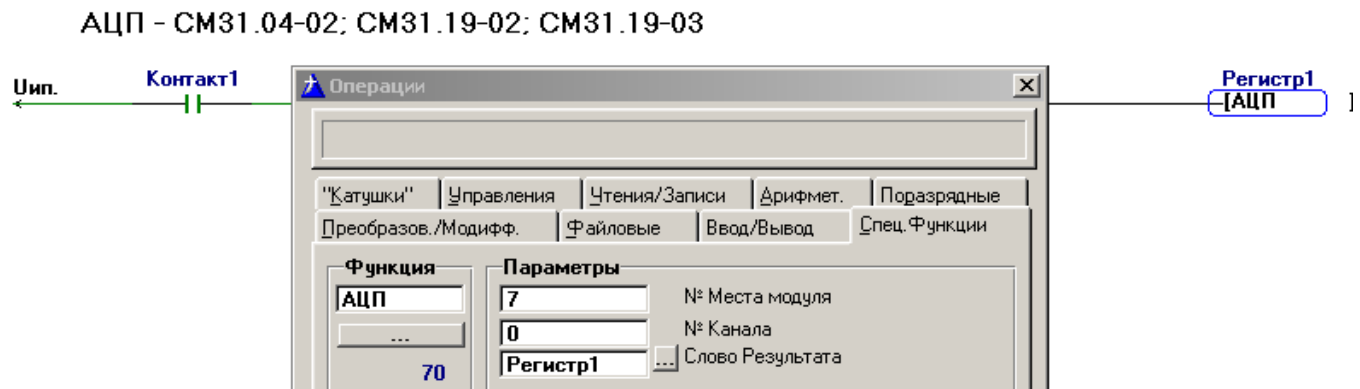
<i>Символ</i>	<i>младший байт</i>	<i>старший байт</i>
	36	35
<i>Символ1</i>	35	33
<i>Символ2</i>	35	00

4. Специальные функции

4.1. Работа с модулями АЦП

Пример.

Релейно-контактная схема:



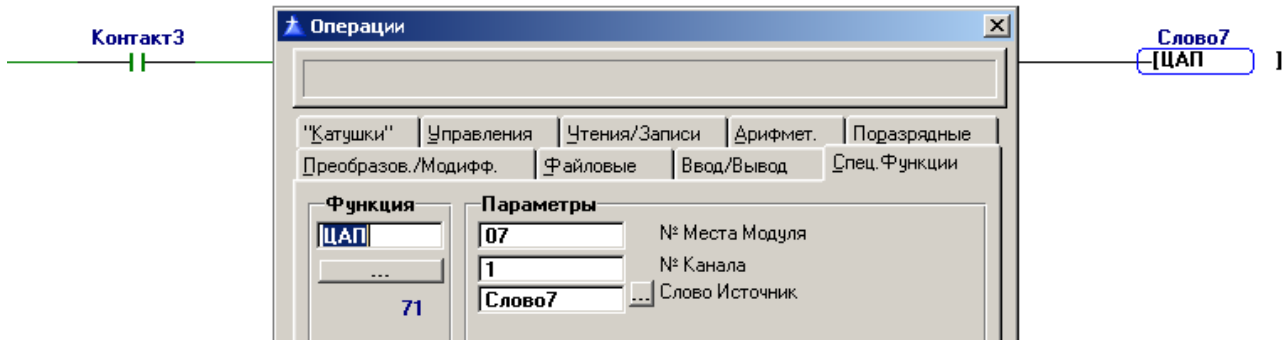
В функции *АЦПвиб.* *Начальный Адрес Результата* необходимо указать, как фиксированный адрес. Результат опроса канала 1 модуля СМ31.04-03 попадает в *Начальный Адрес Результата*, результат опроса канала 2 попадает в следующее за *Начальным Адресом Результата* слово, результат опроса каналов 3,4 –соответственно.

4.2. Работа с модулями ЦАП

Пример.

Релейно-контактная схема:

3 ЦАП - СМ32.03-01, СМ32.03-02, СМ32.03-03



4.3. Работа с модулями ввода импульсных сигналов СР34.23 , СР34.25.

Модуль СР34.23 может работать в следующих режимах:

- накопление без обнуления содержимого счетчика;
- накопление с обнулением содержимого счетчика, после считывания информации ;
- накопление с обнулением содержимого счетчика при поступлении сигнала нуль-метки от датчика.

Режимы работы модуля задаются установкой соответствующих бит в **регистре**

управления. Адрес регистра управления 4. В приведенном ниже примере регистру

управления присвоено имя **РУ**.

РУ.0 =1 -режим работы первого канала – накопление;

РУ.0 =0 -режим работы первого канала – обнуление;

РУ.1 =1 -разрешение обнуления счетчика первого канала при поступлении нуль-метки;

РУ.1 =0 -запрет обнуления счетчика первого канала при поступлении нуль-метки;

РУ.4 =1 -режим работы второго канала – накопление;

РУ.4 =0 -режим работы второго канала – обнуление;

РУ.5 =1 -разрешение обнуления счетчика второго канала при поступлении нуль-метки;

РУ.5 =0 -запрет обнуления счетчика второго канала при поступлении нуль-метки;

РУ.6 =1 -коэффициент умножения счетчика первого канала равен 1;

РУ.6 =0 -коэффициент умножения счетчика первого канала равен 4;

РУ.7=1 -коэффициент умножения счетчика второго канала равен 1;

РУ.7 =0 -коэффициент умножения счетчика второго канала равен 4.

Два счетчика положения подсчитывают количество передних импульсов, поступающих от датчиков (“КАНАЛ1”, ”КАНАЛ2”-обозначение разъемов для подключения сигналов с фото импульсных датчиков) или количество передних и задних фронтов импульсов.

Информация со счетчика 1 поступает на регистр информации с адресом =**0**.

Информация со счетчика 2 поступает на регистр информации с адресом =**2**.

В регистре информации с 0 разряда по 13 разряд находятся данные со счетчика.

В регистре информации 15 разряд –Флаг нуль-метки.

В регистре информации 14 разряд –Флаг счета. Если 14 разряд равен 1 –счет на сложение.

Если 14 разряд равен 0 –счет на вычитание.

В примере регистр информации с адресом **0** имеет имя **РИ1**.

В примере регистр информации с адресом **2** имеет имя **РИ2**.

В модуле имеется **четыре релейных выхода**, которые управляются регистром релейных выходов. Адрес регистра релейных выходов **6** (допускается только запись).

Для быстрого реагирования в модуле имеется **восемь быстрых входов**, которые поступают на регистр дискретных входов. Адрес регистра дискретных входов **6** (допускается только чтение).

В примере регистр релейных выходов имеет имя **РРВ**.

В примере регистр дискретных входов имеет имя **РДВ**.

В модуле есть **регистр прерываний**. Этот регистр доступен только для чтения и имеет адрес 5. При возникновении какой-либо неисправности в линии связи датчика 1 с модулем в разряде 0 будет взведен бит.

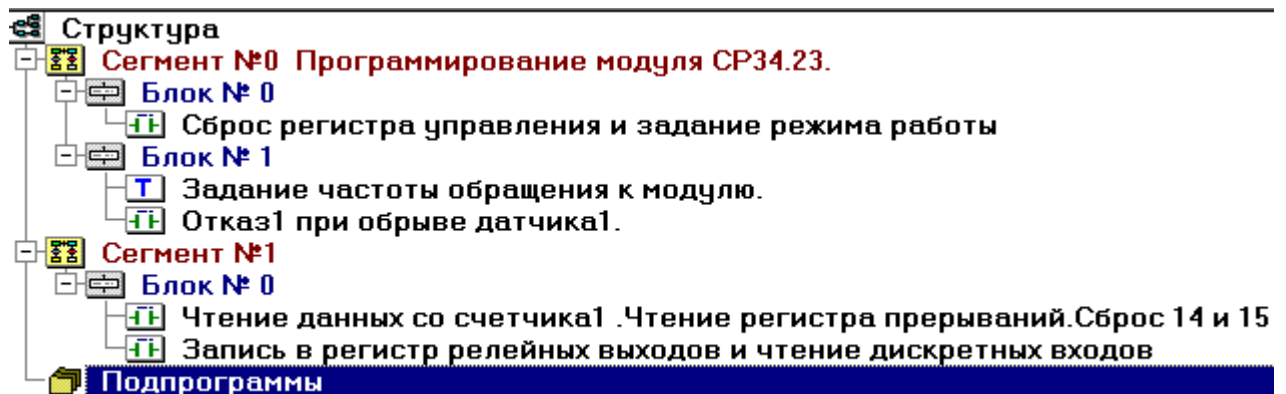
При возникновении какой-либо неисправности в линии связи датчика 2 с модулем в разряде 2 будет взведен бит.

При переполнение накопленного значения первого счетчика, будет взведен бит в разряде 1.

При переполнение накопленного значения второго счетчика, будет взведен бит в разряде 3.

В примере имя **РП** имеет регистр прерываний.

Пример программирования модуля СР34.23, место установки модуля 0.



Сегмент 0

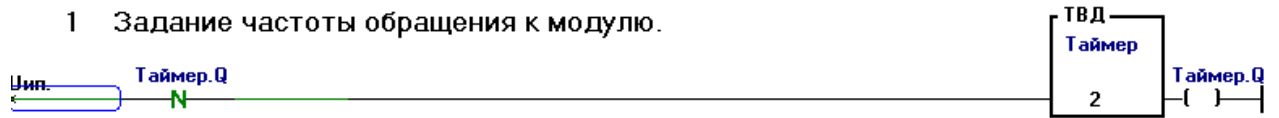
Блок 0

Сброс регистра управления и задание режима работы

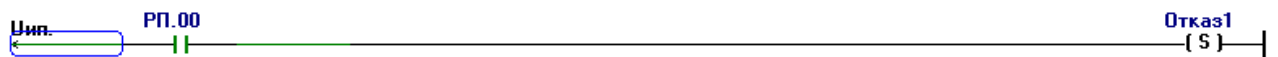


Блок 1

1 Задание частоты обращения к модулю.



2 Отказ1 при обрыве датчика1.

**Сегмент 1****Блок 0**

Чтение данных со счетчика1 .Чтение регистра прерываний.Сброс 14 и 15 разряда



Запись в регистр релейных выходов и чтение дискретных входов



При работе с модулем СР34.25 можно только читать информацию с регистров информации по тому же адресу, что и в модуле СР34.23.

4.4. Работа с панелью оператора К921

Панель оператора **К921** предназначена для работы совместно с **ПЛК**. Связь панели с контроллером осуществляется через коммуникационный канал **RS485** по протоколу **Modbus**. При этом панель является активной станцией по отношению к **ПЛК**.

По включению питания, панель, после самодиагностики и настройки в работу, читает из области конфигурации **ПЛК** начальный адрес массива буквенно-цифровых сообщений (**БЦС**), их количество, а также начальный адрес буфера обмена **ПЛК** с панелью.

Массив **БЦС** создаётся пользователем и располагается в области рабочих программ **ПЛК**.

Массив **БЦС** представляет собой набор текстовых кадров размером по 32 символа в кадре, следующих друг за другом без разрывов. Максимальное количество кадров 128.

Если рабочая программа пользователя вводится в **ПЛК Системой программирования ПЛК “Констар” К748**, то конфигурация и массив **БЦС** создаются автоматически **Системой К748**.

Если рабочая программа пользователя вводится прибором программирования **К901**, то необходимо в область конфигурации (Рис2.3.1) записать начало массива **БЦС**, количество кадров в массиве **БЦС**, задание в **ТД ПЛК** начало адреса буфера обмена с панелью (**Р8,Р9,Р10**). Через этот буфер осуществляется обмен информацией между **ПЛК** и панелью.

Подробное описание по программированию панели можно прочесть в “Инструкции по программированию панели ввода и отображения информации К921”(АЛ2.01148-01-31)

4.5. Работа с двухканальным модулем связи (СР52.05)

Функции обращения к модулю связи (СР52.05) **72, 73, 74, 75** используются для обмена информацией в активном режиме по протоколу **MODBUS** между **ПК** и подключенными по магистрали несколькими **ПК** (до **8**).

Через коммуникационные каналы "**RS232**" / "**RS485**" "**1к**", "**RS485**" "**2к**" модуля связи возможно подключение в пассивном режиме любого другого устройства,

поддерживающего протокол **MODBUS**. В области конфигурации необходимо указать наличие модуля связи на **00** или **01** установочных местах в каркасе.

Рекомендуемые места установки модулей связи СР52.05 – 00 и 01.

Скорость обмена, адрес абонента в магистрали задаются переключками на плате модуля связи согласно АЛ3.094.258РЭ.

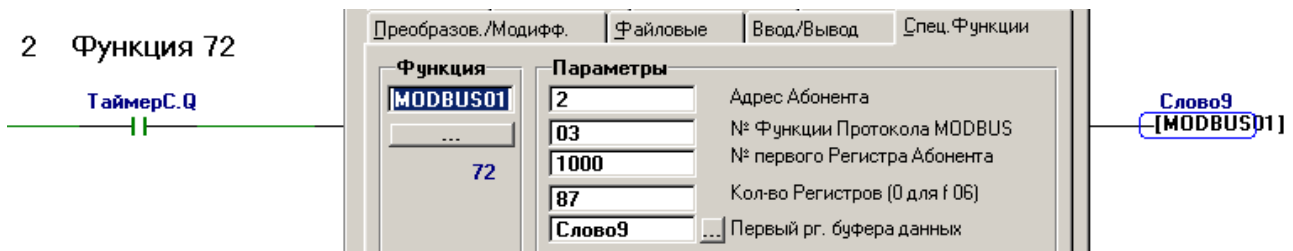
Функция 72 осуществляет обмен информацией по коммуникационному каналу "RS232"/"RS485" "1к" модуля связи на 00 установочном месте в каресе с абонентами 1...8, подключенными по магистрали.

Функция 73 осуществляет обмен информацией по коммуникационному каналу "RS485" "2к" модуля связи на 00 установочном месте в каресе с абонентами 1...8, подключенными по магистрали.

Функция 74 осуществляет обмен информацией по коммуникационному каналу "RS232"/"RS485" "1к" модуля связи на 01 установочном месте в каресе с абонентами 1...8, подключенными по магистрали.

Функция 75 осуществляет обмен информацией по коммуникационному каналу "RS485" "2к" модуля связи на 01 установочном месте в каресе с абонентами 1...8, подключенными по магистрали.

Спец.функции	№ функции	Канал	Место установки CP52.05
MODBUS01	72	1	00
MODBUS02	73	2	00
MODBUS11	74	1	01
MODBUS12	75	2	01



Формат функции 72 (MODBUS –03, 06, 16):

Адрес Абонента - адрес пассивной станции (абонента) - $1 \dots 8_{10}$;

N Функции Протокола MODBUS - номер функции MODBUS-03, 06, 16;

N первого Регистра Абонента - номер первого регистра абонента, с которым производится обмен, (массив регистров в диапазоне $0 \dots 6143$);

Количество Регистров - количество регистров абонента (длина считываемого массива от $1 \dots 123$) для функций 03 или 16, для функции 06- количество регистров =0;

Первый рг. буфера данных - адрес первого регистра абонента для записи массива (функция 16) или для приема массива (функция 03). Адрес любого регистра ТД ПЛК (функция 06), Откуда необходимо взять информацию и передать для записи Абоненту.

Формат функций 73, 74, 75 (MODBUS –03, 06, 16) аналогичен формату **функции 72 (MODBUS –03, 06, 16)**.

При работе с данными функциями по протоколу MODBUS (функции - 03, 06, 16) пользователю предоставляется **ряд служебных регистров**, несущих информацию о состоянии процесса обмена ПЛК с абонентами.

Информация о не завершении (за максимально допустимое время) обмена по **функции 72** содержится в младшем байте **регистра Z6192**, где позиция установленного бита соответствует адресу абонента $1 \dots 8$ - **Z6192.0 ...Z6192.7**.

Информация о нормальном завершении обмена по **функции 72** формируется в **регистре Z6193** в младшем байте, где позиция установленного бита соответствует адресу абонента $1 \dots 8$ - **Z6193.0 ...Z6193.7**. Информация о получении "особого" ответа по **функции 72** содержится в старшем байте **регистра Z6193**. Буфер "особых" ответов от абонентов по **функции 72** расположен с **Z6194** по **Z6208**.

Информация о нормальном завершении обмена по **функции 73** формируется в **регистре Z6211** в младшем байте, где позиция установленного бита соответствует адресу абонента ($1 \dots 8$). Информация о незавершении (за максимально допустимое время) обмена по **функции 73** содержится в младшем байте **регистра Z6210**.

Информация о получении "особого" ответа по **функции 73** содержится в старшем байте **регистра Z6211**.

Буфер "особых" ответов от абонентов по **функции 73** расположен с **Z6212** по **Z6226**.

Информация о нормальном завершении обмена по **функции 74** формируется в **регистре Z6229** в младшем байте, где позиция установленного бита соответствует адресу абонента ($1 \dots 8$). Информация о не завершении (за максимально допустимое время) обмена по **функции 74** содержится в младшем байте **регистра Z6228**.

Информация о получении "особого" ответа по *функции 74* содержится в старшем байте *регистра Z6229*.

Буфер "особых" ответов от абонентов по *функции 74* расположен с *Z6230* по *Z6244*.

Информация о нормальном завершении обмена по *функции 75* формируется в *регистре Z6247* в младшем байте, где позиция установленного бита соответствует адресу абонента (*1...8*).

Информация о не завершении (за максимально допустимое время) обмена по *функции 75* содержится в младшем байте *регистра Z6246*.

Информация о получении "особого" ответа по *функции 75* содержится в старшем байте *регистра Z6247*.

Буфер "особых" ответов от абонентов по *функции 75* расположен с *Z6248* по *Z6262*.

Признак отказа модуля связи (CP52.05) формируется в *Z6190* (по месту установки модуля).

Структура буфера "особых" ответов

Длина буфера "особых" ответов от абонентов составляет 16 x 16- разрядных слов по два на каждый абонент.

Формат "особого" ответа приведен на рис. 4.3.1

Это два шестнадцатиразрядных слова буфера "особых" ответов для абонента с адресом 1.

0...7p	8...15p
<i>Адрес абонента (1)</i>	<i>Номер функции MODBUS</i> 03-83H, 06-86H, 16-90H
<i>Код особого ответа:</i> 01 – запрещенная функция; 02 – запрещенный адрес данных; 03 – запрещенные данные (пре- вышение длины массива); 04 – отказ модуля связи	КС

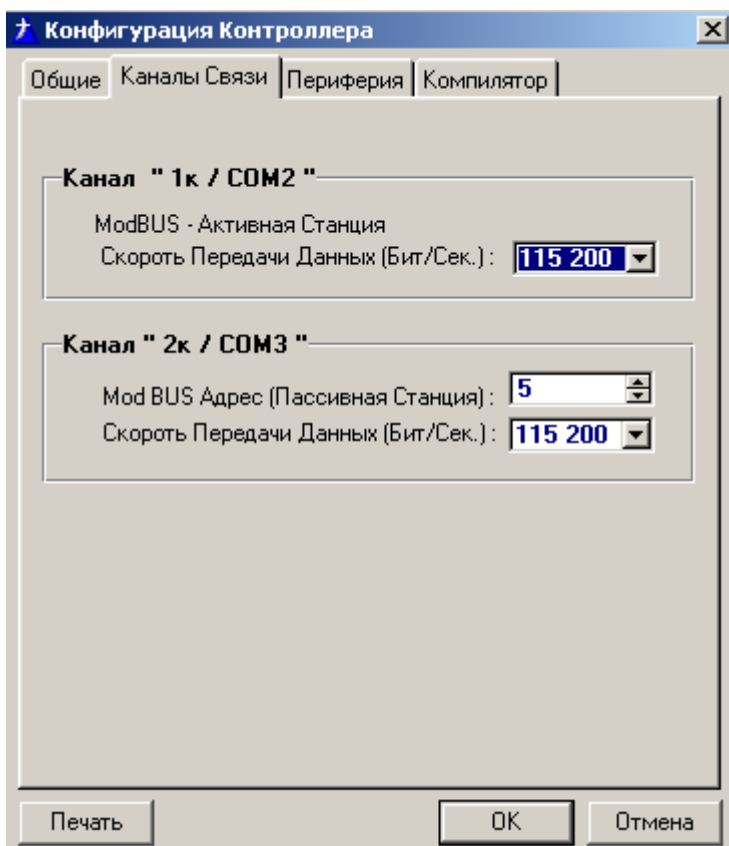
Рис.4.3.1

4.6. Работа с коммуникационным каналом "RS485" либо "RS232"

Связь в CP59.10 обеспечивается через коммуникационный канал "RS485" либо "RS232". Коммуникационный канал может работать как в активном, так и в пассивном режиме. Обмен информацией осуществляется по протоколу MODBUS. Коммуникационный канал может быть активным либо пассивным. Скорость обмена задаётся в области конфигурации.

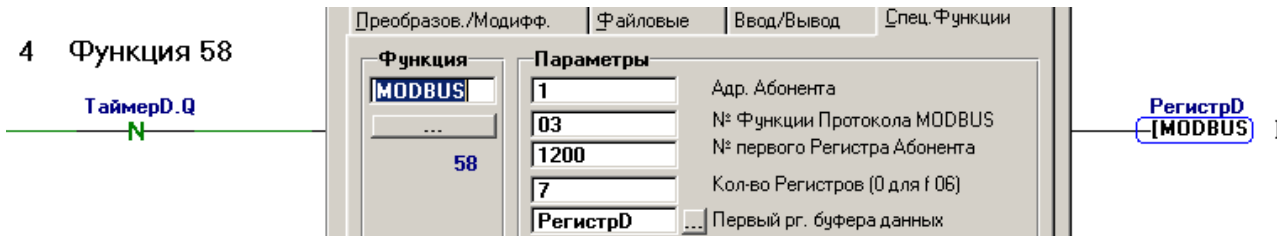
Если канал пассивный, то в меню "Проект\Конфигурация", нажать кнопку "Каналы связи". В поле **Канал "2к/COM3"**:

ModBUS Адрес (Пассивная Станция) записать номер абонента (1 - 8).



Если коммуникационный канал будет работать как активная станция, то нужно записать **0**. Количество абонентов, с которыми может быть реализован обмен, лежит в пределах **1...8**. В качестве абонента может быть использован ПЛК (**K200** или **K300**), или любое другое устройство, поддерживающее протокол MODBUS.

Связь с абонентом реализуется в РП посредством вызова **функции 58**.



Формат функции 58 (MODBUS – 03, 06, 16):

Адрес Абонента - адрес пассивной станции (абонента) - $1 \dots 8_{10}$;

N Функции Протокола MODBUS - номер функции MODBUS-03, 06, 16;

N первого Регистра Абонента - номер первого регистра абонента, с которым производится обмен, (массив в пределах адреса в диапазоне $0 \dots 8191$);

КоличествоРрегистров - количество регистров абонента (длина считываемого массива от $1 \dots 123$) для функций 03 или 16, для функции 06- количество регистров =0;

Первый рг. буфера данных - адрес первого регистра абонента для записи массива (функция 16) или для приема массива (функция 03). Адрес любого регистра ТД ПЛК (функция 06).

При работе с **функцией 58** по протоколу MODBUS (функции - 03, 06, 16) пользователю предоставляется **ряд служебных регистров**, несущих информацию о состоянии процесса обмена ПК с абонентами.

Информация о нормальном завершении обмена по **функции 58** формируется в младшем байте **регистра Z6173**, где позиция установленного бита в младшем байте ($0 \dots 7$) соответствует адресу абонента ($1 \dots 8$), с которым обмен завершен нормально.

Информация о не завершении (за максимально допустимое время) обмена по **функции 58** содержится в младшем байте **регистра Z6172**, где позиция установленного бита в младшем байте ($0 \dots 7$) соответствует адресу ($1 \dots 8$) не ответившего абонента.

Информация о получении "особого" ответа по **функции 58** содержится в старшем байте **регистра Z6173**, где позиция установленного бита в старшем байте ($08 \dots 15$) соответствует адресу ($1 \dots 8$), приславшего "особый" ответ.

Структура буфера "особых" ответов

Длина буфера "особых" ответов от абонентов составляет 16 x 16- разрядных слов по два на каждый абонент, начиная с адреса **Z6174** по **Z6188**.

Формат "особого" ответа приведен на рис.4.6.1.

	0...7p	8...15p
Z6174	<i>Адрес абонента (1)</i>	<i>Номер функции MODBUS</i> 03-83H, 06-86H, 16-90H
Z6175	<i>Код особого ответа:</i> 01 – запрещенная функция 02 – запрещенный адрес данных 03 – превышение длины массива 04 – процессор занят	КС

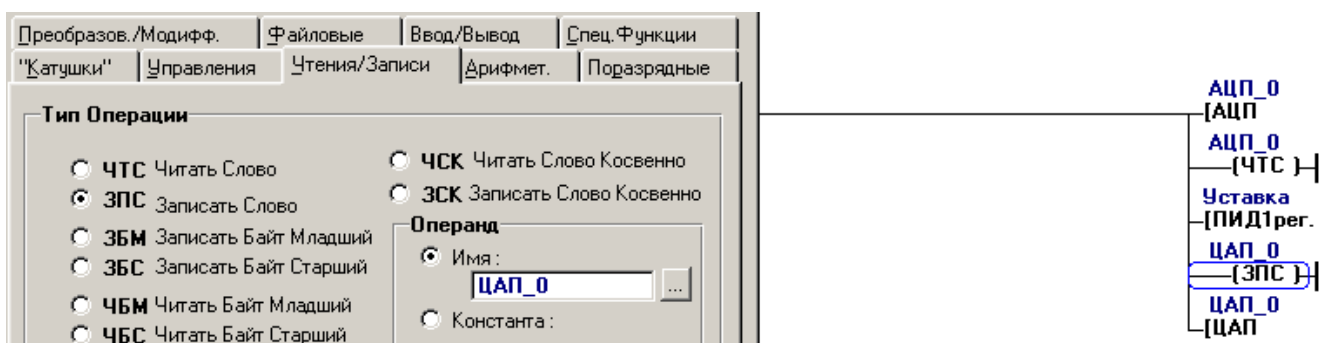
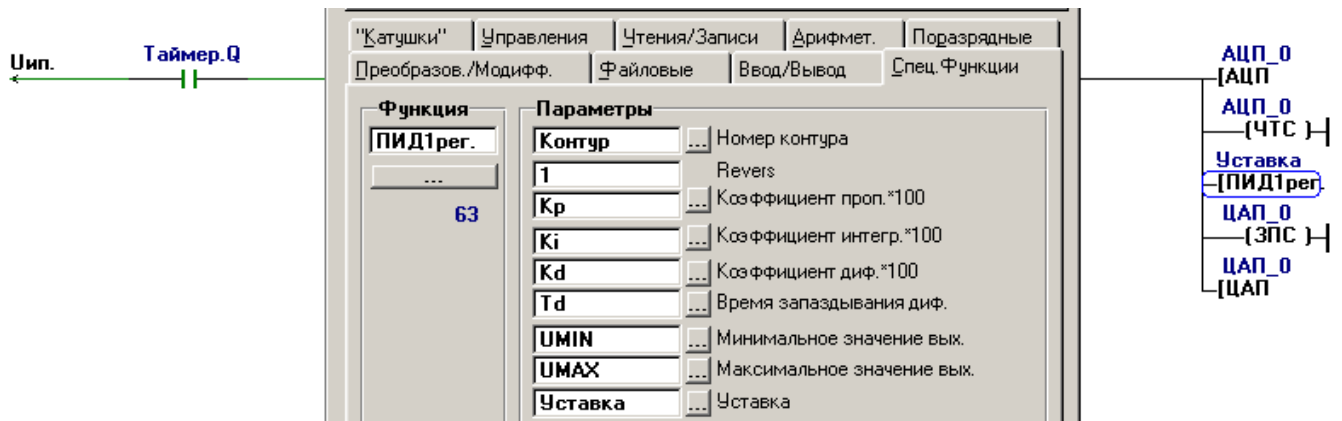
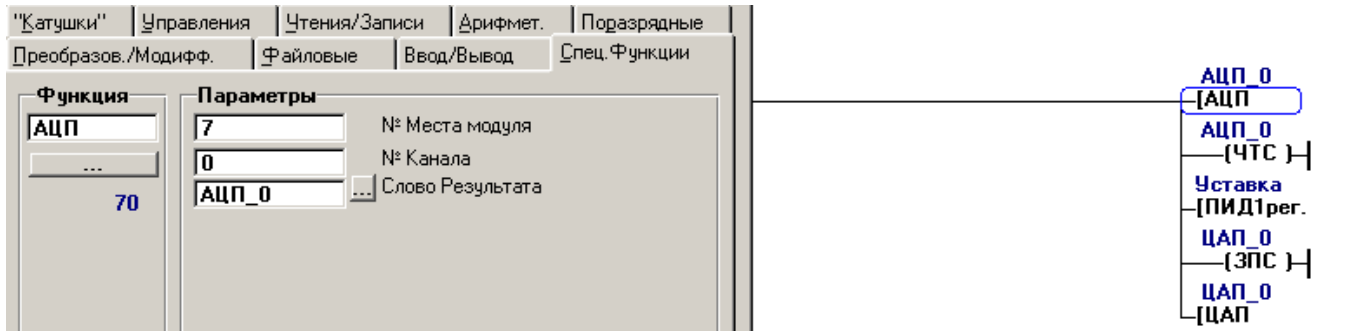
Рис.4.6.1

4.7. ПИД -регулятор

Специальная команда: (пропорционально-интегрально-дифференциальный) регулятор ПИД.

Это команда цифрового управления аналоговыми процессами. Максимальное число независимых контуров управления –12.

Пример.



По функции 70 (АЦП) считывается с модуля АЦП (в данном примере установленном на место 7 с канала 0) информация и записывается в слово АЦП_0. Это же значение поступает на вход функции 63 (ПИД1рег.), просчитывается ПИД- регулятором и выдается на модуль ЦАП (установленный на место 5 в данном примере).

Краткое описание.

Контур регулирования состоит из трех компонент: пропорциональной, интегральной и дифференциальной. Включение компонент в контур, определяется значением коэффициентов. Если значение коэффициента равно нулю – это значит исключена соответствующая компонента.

Отклонение ΔX контролируемого параметра X_t от уставки X_s вызывает выработку ПИД- регулятором управляющего воздействия U :

$$U = \text{PID}(\Delta X)$$

$$U = U_p + U_i + U_d$$

$$\Delta X = X_t - X_s$$

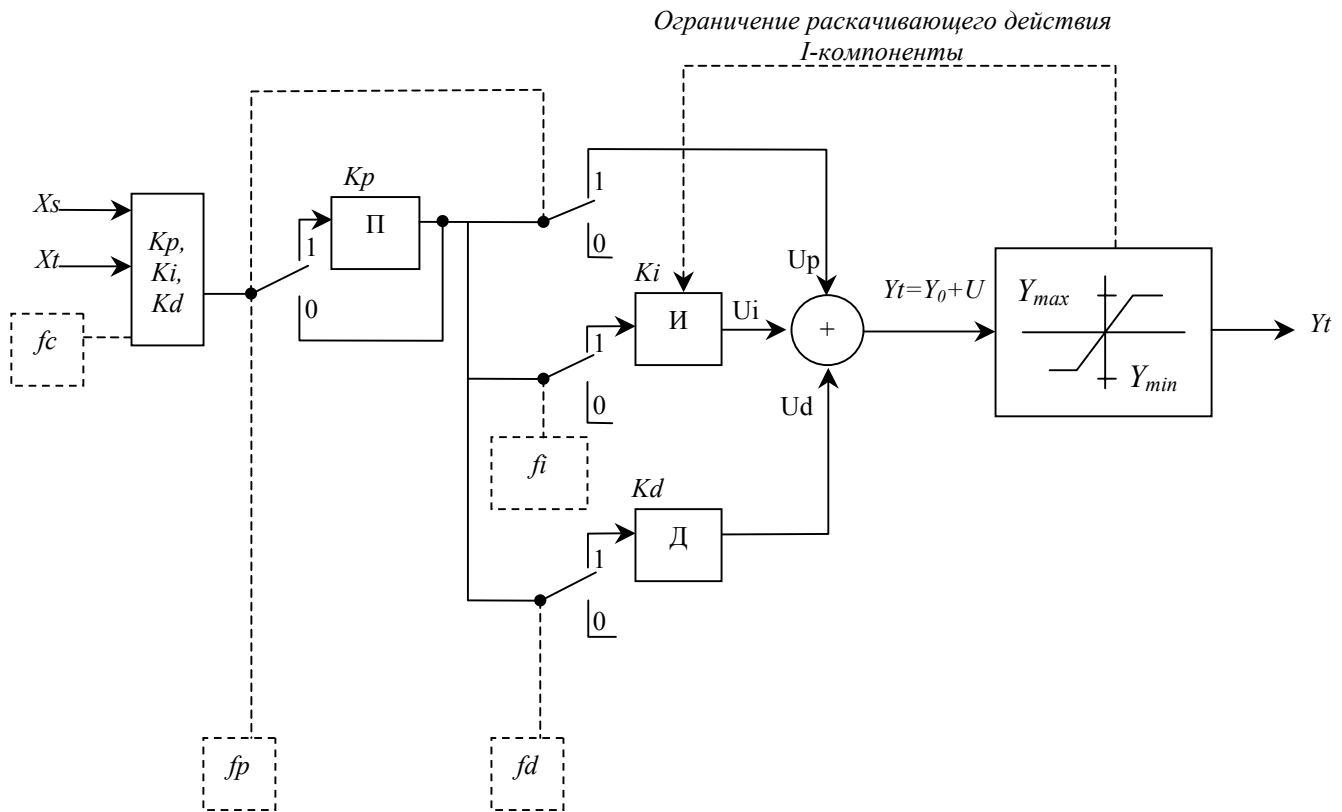
$$U_p = K_p \cdot \Delta X \quad \text{– пропорциональная (P-компонента),}$$

$$U_i = K_i \cdot Dt \cdot (\Delta X + \Delta X_{пред}) + U_{инпред} \quad \text{– интегральная (I-компонента),}$$

$$U_d = (K_d \cdot (\Delta X - \Delta X_{пред}) + U_{днпред}) / Dt \quad \text{– дифференциальная (D-компонента),}$$

Индексом «пред» отмечены параметры на предыдущем обращении к функции.

Структурная схема ПИД-регулятора.



В регуляторе применяется ограничение раскачивающего действия *I*-компоненты, заключающееся в том, что ее вычисленное значение корректируется в соответствии с неравенством:

$$Y_{min} - Y_0 - U_p \leq U_i \leq Y_{max} - Y_0 - U_p,$$

т.е. *Dt*-компонента не участвует в расчете ограничений на *I*-компоненту.

Итоговое управляющее воздействие *Yt* выдается с учетом неравенства:

$$Y_{min} \leq Y_t \leq Y_{max}$$

4.8 Экстренное обращение к модулям ввода (вывода)

Инструкции экстренного обращения к модулям ввода (вывода):

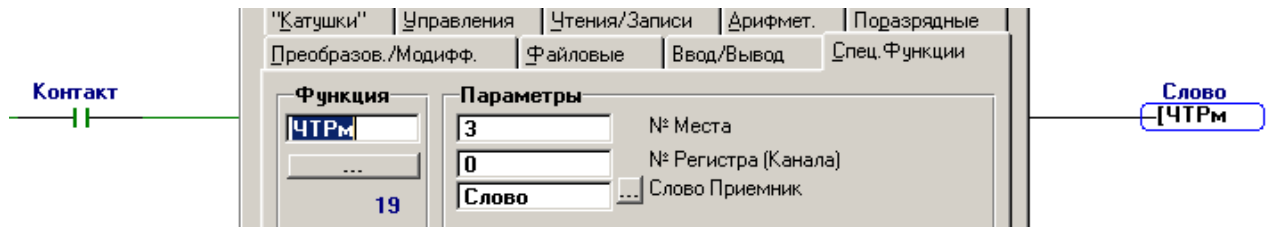
- Читать из регистра (*ЧТРм*);
- Записать в регистр (*ЗПРм*).

Инструкции *ЧТРм* и *ЗПРм* применяются при работе с модулями, имеющими регистры.

Инструкции *ЧТРм* и *ЗПРм* являются условными.

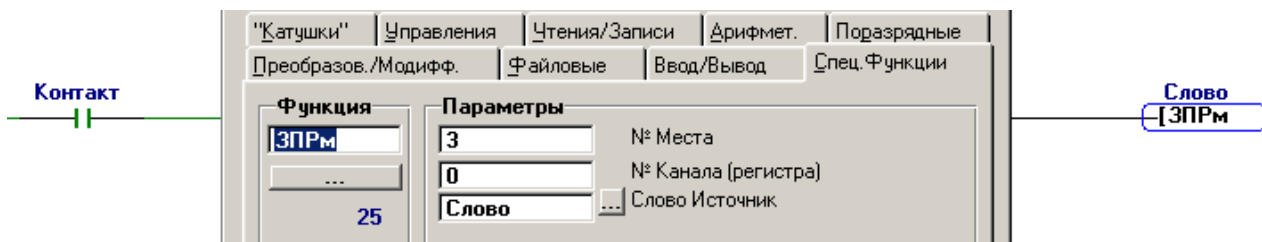
Читать из регистра (ЧТРм)

При работе с *модулем ввода дискретных сигналов* считывается *содержимое всего регистра* и записывается в Слово Приёмник.



Записать в регистр (ЗПРм)

При работе с *модулем вывода дискретных сигналов* в РП формируется *слово-состояние выходного регистра* (состояние каналов на момент выдачи сигнала), которое затем поступит на *выходной регистр*. Данные для *выходного регистра* должны быть предварительно сформированы в Слове Приёмнике.



4.9. Операции с вещественными числами.

Преобразование Целого в Вещественное

Функция	Параметры
FLOAT	Слово1 ... Целое
...	Слово2 ... Результат
52	

Необходимо помнить, что для записи вещественного числа, нужно зарезервировать 2 слова.

Преобразование Вещественного в Целое

Функция	Параметры
IFIX	Слово2 ... Вещественное
...	Слово1 ... Результат
53	

Сложение Вещественных Чисел

Функция	Параметры
FAD	Слово1 ... Первое Слагаемое
...	Слово3 ... Второе Слагаемое
54	

Вычитание Вещественных Чисел

Функция	Параметры
FSB	Слово1 ... Уменьшаемое
...	Слово3 ... Вычитаемое
55	

Умножение Вещественных Чисел

Функция	Параметры
<input type="text" value="FMP"/>	<input type="text" value="Слово1"/> ... Множимое
<input type="text" value="..."/>	<input type="text" value="Слово3"/> ... Множитель
56	

Деление Вещественных Чисел

Функция	Параметры
<input type="text" value="FDV"/>	<input type="text" value="Слово1"/> ... Делимое
<input type="text" value="..."/>	<input type="text" value="Слово3"/> ... Делитель
57	

Результат операции в Функциях *FAD*, *FSB*, *FMP*, *FDV* находится в приведенных примерах в *Слове1*.

Список Инструкций ПК “Констар”

Инструкция	PKC	Наименование
Логическая	- - - /- -()- -(S)- -(R)- ┌ └ - P - - N -	Нормально разомкнутый контакт (Прямой опрос битового объекта) Нормально замкнутый контакт (Инверсный опрос битового объекта) Катушка нейтрального реле (Установить битовый объект) Установочная катушка поляризованного реле (Установить бит-й объект и зафиксировать) Сбросовая катушка поляризованного реле (Сбросить бит-й объект и зафиксировать) Начало ветвления параллельной цепи Конец ветвления параллельной цепи Контакт – Передний фронт импульса Контакт – Задний фронт импульса
Структурирова- ния	НСТ ОСТ ЗСТ ССТ НБЛ ПБЛ	Начало сегмента Открыть сегмент Закрыть сегмент Сбросить сегмент Начало блока Перейти на блок
Для работы с подпрограм- мами	ВПП НПП	Вызвать подпрограмму Начало подпрограммы
Таймеров и счетчиков	ТВД ТВС ТНС СЧС СЧВ СЧЦ	Таймер на включение (0.1 с) Таймер на выключение (0.1 с) Таймер накапливающий (0.1 с) Счетчик на сложение Счетчик на вычитание Счетчик циклов
Сравнения	< > = ><	Сравнить на “меньше” Сравнить на “больше” Сравнить на “равно” Сравнить на “принадлежность пределам”
Для работы с данными	ЧТС ЗПС ПУС ПСС ЗБМ ЗБС ЧБМ ЧБС ПББ СБС СПС СЦМ ИНК ДЕК ЧСК ЗСК ПБС	Читать слово в аккумулятор Записать в слово из аккумулятора Поразрядно умножить слово Поразрядно сложить слово Записать байт младший Записать байт старший Читать байт младший Читать байт старший Обмен содержимым байт млад. и байт стар. Сброс слова Сдвиг циклический к старшим битам Сдвиг циклический к младшим битам Инкремент слова Декремент слова Читать слово косвенно Записать слово косвенно Преобраз. двоич. Представ. числа в символ.
Файловая	ПФФ ЗСФ ЧСФ	Переместить файл Записать слово в файл Читать слово из файла

Инструкция	PKC	Наименование
Экстренного обращения к модулям В/В	ЧТРм ЗПРм	Читать из регистра Записать в регистр
Арифметические	САС ВАС УАС ДАС	Сложить арифметически слово Вычитать арифметически слово Умножить арифметически слово Разделить арифметически слово
Управления выполнением программы	СТП	Программный останов программы пользователя

(Специальные функции)

Функция	Наименование
FLOAT	Преобразование Целого в Вещественное
IFIX	Преобразование Вещественного в Целое
FAD	Сложение Вещественных Чисел
FSB	Вычитание Вещественных Чисел
FMP	Умножение Вещественных Чисел
FDV	Деление Вещественных Чисел
АЦП	Работа с модулями АЦП
АЦПвиб.	Работа с модулем ввода сигналов Виброскорости
ЦАП	Работа с модулями ЦАП
MODBUS	Протокол MODBUS Активная станция Порт COM2
MODBUS01	Работа с модулем CP52.05 посадочное место 00, Порт COM1 Протокол MODBUS
MODBUS02	Работа с модулем CP52.05 посадочное место 00, Порт COM2 Протокол MODBUS
MODBUS11	Работа с модулем CP52.05 посадочное место 01, Порт COM1 Протокол MODBUS
MODBUS12	Работа с модулем CP52.05 посадочное место 01, Порт COM2 Протокол MODBUS
ПИД рег.	Пропорциональный Интегрирующий Дифференциальный Регулятор (ПИД регулятор)
ЧТРм ЗПРм	Читать из регистра Записать в регистр

Таблица адресов служебных регистров CP59.10

Адрес	Назначение
Z6144	Режим работы ПЛК
.0	Z6144.0 = 0 – Отладка; Z6144.0 = 1 – Работа
.1	Z6144.1 = 0 - Блокировка; Z6144.1 = 1 – Обмен
.2...4	Стоп
.2...4	Цикл
.2...4	Пуск
.15	Обобщенный бит отказов
Z6145	Системные отказы ПЛК
.0	Отказ РИ
.1	Ошибка КС РПО
.2	Ошибка ТД
.3	Ошибки в РП
.4	Отказ БВ
.6	Превышение времени цикла
.8	Программный останов
.9	Ошибка ввода/вывода
.10	Нет связи с расширителем
.15	Нет связи с активной станцией
Z6146	Отказы модулей: места 0 – 15 основного каркаса
Z6147	Отказы модулей: места 0 – 15 каркаса расширителя
Z6162.	Регистр результата работы арифметических инструкций
.0	САС. Результат вышел за пределы 16-ти разрядной сетки
.1	ВАС. Отрицательный результат
.2	УАС. Результат вышел за пределы 16-ти разрядной сетки
.3	ДАС. Деление на «0»
Z6172	Регистр сбоев связи с абонентами – CP59.02 “RS485” или “RS232”
.0...7	Нет ответа от абонента с адресом 1 – 8
Z6173	Регистр нормальных и "особых" ответов от абонентов
.0...7	Завершение связи по норме с абонентом по адресу 1 – 8
.8...15	Наличие "особого" ответа от абонента по адресу 1 – 8
Z6174, Z6175	Буфер "особых" ответов от абонента с адресом 1
Z6176, Z6177	Буфер "особых" ответов от абонента с адресом 2
Z6178, Z6179	Буфер "особых" ответов от абонента с адресом 3
Z6180, Z6181	Буфер "особых" ответов от абонента с адресом 4
Z6182, Z6183	Буфер "особых" ответов от абонента с адресом 5
Z6184, Z6185	Буфер "особых" ответов от абонента с адресом 6
Z6186, Z6187	Буфер "особых" ответов от абонента с адресом 7
Z6188, Z6189	Буфер "особых" ответов от абонента с адресом 8
Z6190	Регистр сбоев связи с абонентами - CP52.05 на 0 месте канал 1- MODBUS01
.0...7	Нет ответа от абонента с адресом 1 – 8

Продолжение таблицы адресов служебных регистров CP59.10

Адрес	Назначение
Z6193	Регистр нормальных и "особых" ответов от абонентов
.0...7	Завершение связи по норме с абонентом по адресу 1 – 8
.8...15	Наличие "особого" ответа от абонента по адресу 1 – 8
Z6194, Z6195	Буфер "особых" ответов от абонента с адресом 1
Z6196, Z6197	Буфер "особых" ответов от абонента с адресом 2
Z6198, Z6199	Буфер "особых" ответов от абонента с адресом 3
Z6200, Z6201	Буфер "особых" ответов от абонента с адресом 4
Z6202, Z6203	Буфер "особых" ответов от абонента с адресом 5
Z6204, Z6205	Буфер "особых" ответов от абонента с адресом 6
Z6206, Z6207	Буфер "особых" ответов от абонента с адресом 7
Z6208, Z6209	Буфер "особых" ответов от абонента с адресом 8
Z6210	Регистр сбоев связи с абонентами -CP52.05 на 0 месте канал 2 - MODBUS02
.0...7	Нет ответа от абонента с адресом 1 – 8
Z6211	Регистр нормальных и "особых" ответов от абонентов
.0...7	Завершение связи по норме с абонентом по адресу 1 – 8
.8...15	Наличие "особого" ответа от абонента по адресу 1 – 8
Z6212, Z6213	Буфер "особых" ответов от абонента с адресом 1
Z6214, Z6215	Буфер "особых" ответов от абонента с адресом 2
Z6216, Z6217	Буфер "особых" ответов от абонента с адресом 3
Z6218, Z6219	Буфер "особых" ответов от абонента с адресом 4
Z6220, Z6221	Буфер "особых" ответов от абонента с адресом 5
Z6222, Z6223	Буфер "особых" ответов от абонента с адресом 6
Z6224, Z6225	Буфер "особых" ответов от абонента с адресом 7
Z6226, Z6227	Буфер "особых" ответов от абонента с адресом 8
Z6228	Регистр сбоев связи с абонентами -CP52.05 на 1 месте канал 1 MODBUS011
.0...7	Нет ответа от абонента с адресом 1 – 8
Z6229	Регистр нормальных и "особых" ответов от абонентов
.0...7	Завершение связи по норме с абонентом по адресу 1 – 8
.8...15	Наличие "особого" ответа от абонента по адресу 1 – 8
Z6230, Z6231	Буфер "особых" ответов от абонента с адресом 1
Z6232, Z6233	Буфер "особых" ответов от абонента с адресом 2
Z6234, Z6235	Буфер "особых" ответов от абонента с адресом 3
Z6236, Z6237	Буфер "особых" ответов от абонента с адресом 4
Z6238, Z6239	Буфер "особых" ответов от абонента с адресом 5
Z6240, Z6241	Буфер "особых" ответов от абонента с адресом 6
Z6242, Z6243	Буфер "особых" ответов от абонента с адресом 7
Z6244, Z6245	Буфер "особых" ответов от абонента с адресом 8
Z6246	Регистр сбоев связи с абонентами -CP52.05 на 1 месте канал 2- MODBUS012
.0...7	Нет ответа от абонента с адресом 1 – 8
Z6247	Регистр нормальных и "особых" ответов от абонентов
.0...7	Завершение связи по норме с абонентом по адресу 1 – 8
.8...15	Наличие "особого" ответа от абонента по адресу 1 – 8

Продолжение таблицы адресов служебных регистров СР59.10

Адрес	Назначение
Z6248, Z6249	Буфер "особых" ответов от абонента с адресом 1
Z6250, Z6251	Буфер "особых" ответов от абонента с адресом 2
Z6252, Z6253	Буфер "особых" ответов от абонента с адресом 3
Z6254, Z6255	Буфер "особых" ответов от абонента с адресом 4
Z6256, Z6257	Буфер "особых" ответов от абонента с адресом 5
Z6258, Z6259	Буфер "особых" ответов от абонента с адресом 6
Z6260, Z6261	Буфер "особых" ответов от абонента с адресом 7
Z6262, Z6263	Буфер "особых" ответов от абонента с адресом 8
Z6264, Z6399	Таблицы ПИД регулятора
Z6400	Буфер обмена с модулями дискретного ввода и вывода каркаса расширения
Z6528	Буфер обмена с модулями дискретного ввода и вывода основного каркаса
Z6656	Буфер обмена с модулями АЦП каркаса расширения
Z6656	Буфер обмена с модулями ЦАП каркаса расширения
Z7296	Буфер обмена с модулями АЦП основного каркаса
Z7680	Буфер обмена с модулями ЦАП основного каркаса

Приложение 3

Управление движением вагонетки

Верс. 1.4

Печать : 27/08/02 16:13

Проект:**Наименование:** Управление движением вагонетки**Разработчик:** Скибенко Л.М.**Версия:** 1.4**Создан:** 7/08/2002 16:00**Модифицирован:** 27/08/2002 16:12**Доп. Информация:**

Управление движением вагонетки осуществляется в двух режимах: Автоматическом и Ручном.

Движение ВПЕРЕД - максимальное время в пути - 120с

Движение НАЗАД - максимальное время в пути - 200с

Если вагонетка не достигла заданного упора, то необходимо выдать на панель соответствующее оперативное сообщение.

Пуск и Стоп может быть от кнопок ПУСК и СТОП либо команды с панели.

К оперативным сообщениям следует отнести и сообщения о положении вагонетки, то есть она либо движется либо находится в одном из крайних положений.

Процесс движения вагонетки сопровождается выдачей оперативных сообщений на панель. При нажатии клавиши на панели:

F1 - появится сообщение СТОЙ

F2 -

ЧИСЛО ПРОХОДОВ

F3 - параметр

335 (например)

F5 - сброс счетчика проходов, чтобы возобновить выдачу параметра, нужно нажать клавишу F3.

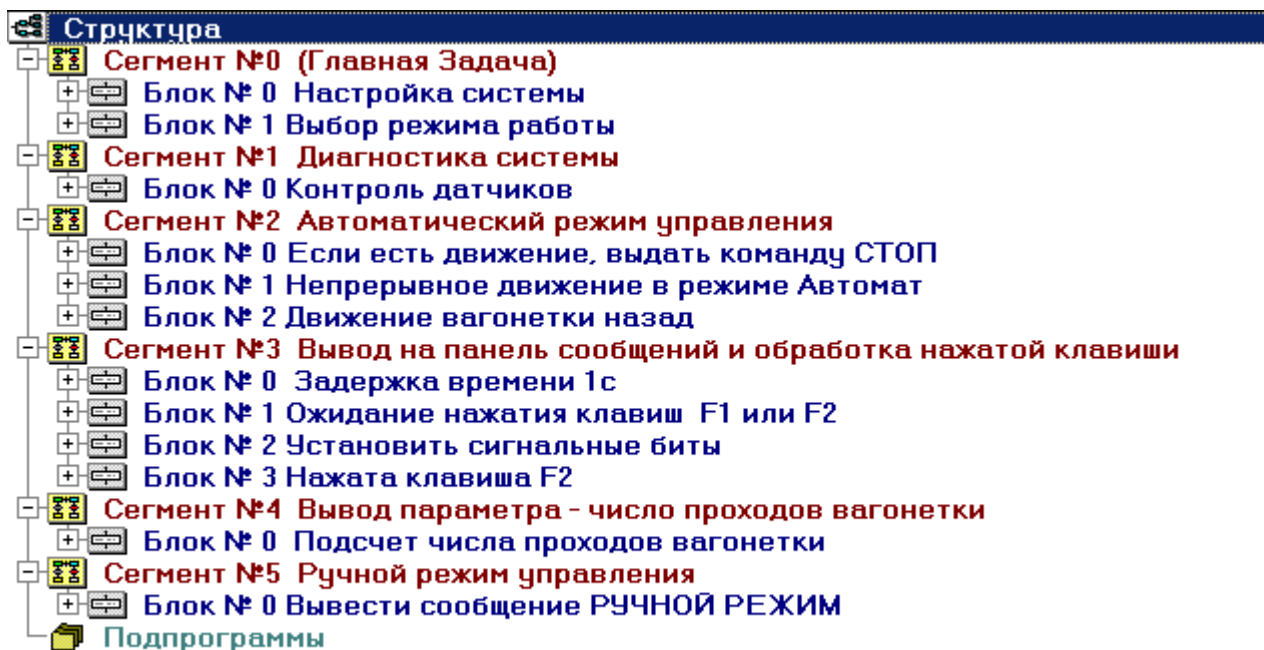
Процессорный Модуль**Тип:** CP59.02**Максимальное время цикла:** 20**Контролировать резервный источник:** **Каркас расширения****Наличие каркаса расширения:** **Тип каркаса:** 1 x 16 мест**Панель K921:****Наличие панели K921:** **Начальный адрес буфера обмена:** 2001**Канал "1к/COM2"****Скорость передачи данных (бит/с):** 115 200**Канал "2к/COM3"****ModBUS Адрес (Пассивная Станция):** 1**Скорость передачи данных (бит/с):** 115 200**Периферия:**

Кр.	Мст.	Обмен	Стоп	Модуль	Назначение
0	00	Да	Да	CP52.05	Модуль Связи RS232, RS485 (смотри доп.)
0	02	Да	Нет	CM34.01	Ввод дискрет. сигналов =24 В
0	03	Да	Да	CM35.01	Вывод дискрет. сигналов =24 В. Вых. ток 1 А

№	Имя	Тип	Д/А	0	Кр.	Мс.	Кн.	Адрес	Бит	Комментарий
1	F1	Внутр.	Д					2001	0	Клавиша на панели F1
2	F2	Внутр.	Д					2001	1	Клавиша на панели F2
3	F3	Внутр.	Д					2001	2	Клавиша на панели F3
4	F5	Внутр.	Д					2001	4	
5	Автомат	Вход			0	2	6			Положение переключателя
6	Вперед	Выход			0	3	0			Движение вагонетки
7	Датчик1	Вход			0	2	0			Датчик положения Старт
8	Датчик2	Вход			0	2	1			Датчик положения Тупик
9	Движение	Вход			0	2	2			Вагонетка движется
10	Кн_Пуск	Вход			0	2	3			Кнопка Пуск
11	Кн_Стоп	Вход			0	2	4			Кнопка Стоп
12	КомВ	Внутр.	Д					2001	11	Команда с панели -Вперед -нажата клавиша стрелка вправо
13	КомН	Внутр.	Д					2002	0	Команда с панели -Назад -нажата клавиша стрелка влево
14	КомРег	Внутр.	А							Командный регистр
15	КомС	Внутр.	Д					2001	15	Команда с панели -Стоп -нажата клавиша стрелка вниз
16	КомУпр	Внутр.	А							Команды управления
17	Назад	Выход			0	3	1			Движение вагонетки
18	ОперСооб	БЦС								Оперативные сообщения
19	ОтказД1	Вход			0	2	7			Отказ Датчика 1
20	ОтказД2	Вход			0	2	8			Отказ Датчика 2
21	Пр_Авт	Внутр.	Д	+						
22	РегСигБ	Внутр.	А					2013		
23	Ручой	Вход			0	2	5			Положение переключателя
24	Стоп	Выход			0	3	2			Команда
25	СчетПр	Счетчик								Счетчик числа проходов
26	T3	Таймер								Введен для имитации числа проходов
27	Таймер1	Таймер								Время движения назад
28	Таймер2	Таймер								Время движения вперед
29	Таймер3	Таймер								Время положения Стоп

№	Имя	Тип	Д/А	0	Кр.	Мс.	Кн.	Адрес	Бит	Коментарий
30	Таймер4	Таймер								Временная задержка для выдачи сообщений
31	ТехПар1	Внутр.	А					2005		Значение первого технологического параметра
32	ТмРежим	Таймер								Таймер Включения Режима
33	ХарТехП1	Внутр.	А					2006		Характеристика первого технологического папарметра

Структурная схема проекта



Буквенно – цифровые сообщения

Управление движением вагонетки

Верс. 1.4

Печать: 27/08/02 14:52

Кадр № 1

Строка 1: **ВАГОН НЕ ДОСТИГ**
.....
Строка 2: **ПОЛОЖЕНИЯ ТУПИК**

Кадр № 2

Строка 1: **СТОП**
.....
Строка 2:

Кадр № 3

Строка 1: **ВАГОН НЕ ДОСТИГ**
.....
Строка 2: **ПОЛОЖЕНИЯ СТАРТ**

Кадр № 4

Строка 1: **ОТКАЗ ДАТЧИКА1**
.....
Строка 2:

Кадр № 5

Строка 1: **ОТКАЗ ДАТЧИКА2**
.....
Строка 2:

Кадр № 6

Строка 1: **РУЧНОЙ РЕЖИМ**
.....
Строка 2:

Кадр № 7

Строка 1: **ДВИЖЕНИЕ**
.....
Строка 2:

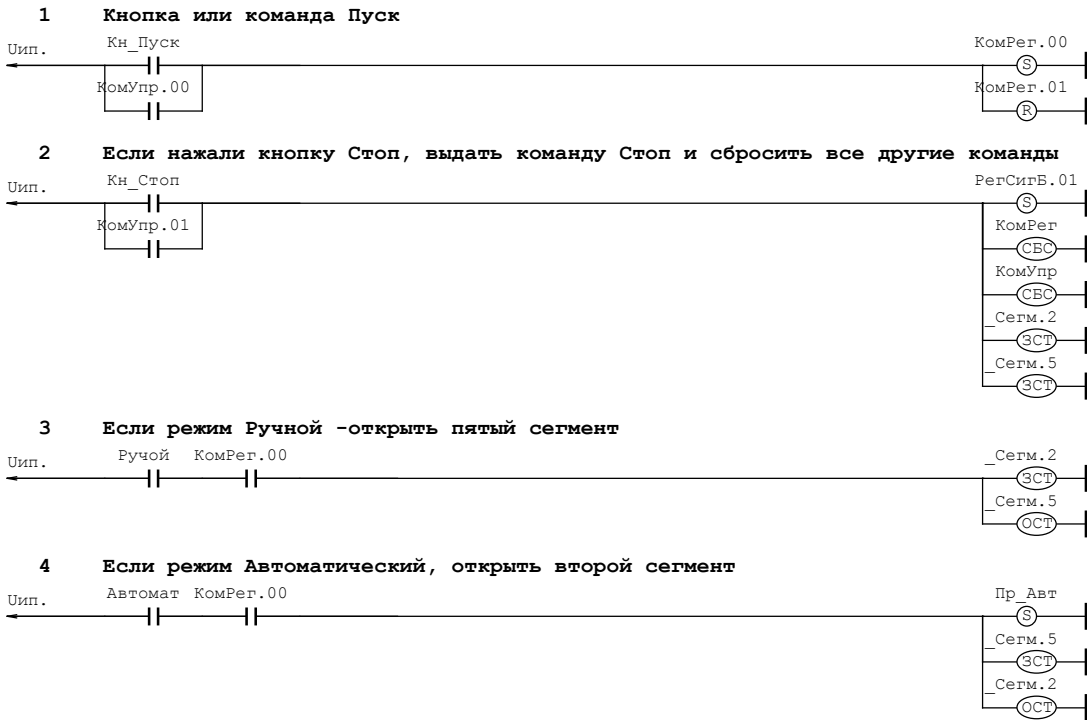
Кадр № 8

Строка 1: **ЧИСЛО ПРОХОДОВ**
.....
Строка 2:

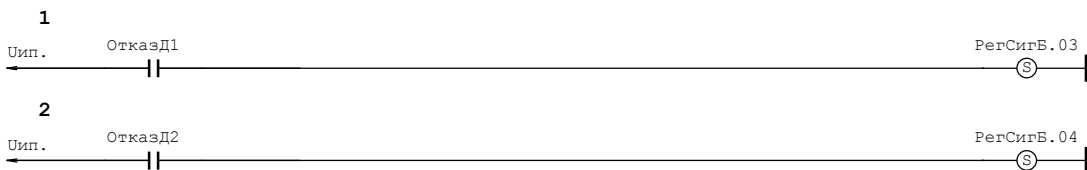
Сегмент № 0 (Главная Задача)
Блок № 0 Настройка системы



Сегмент № 0 (Главная Задача)
Блок № 1 Выбор режима работы

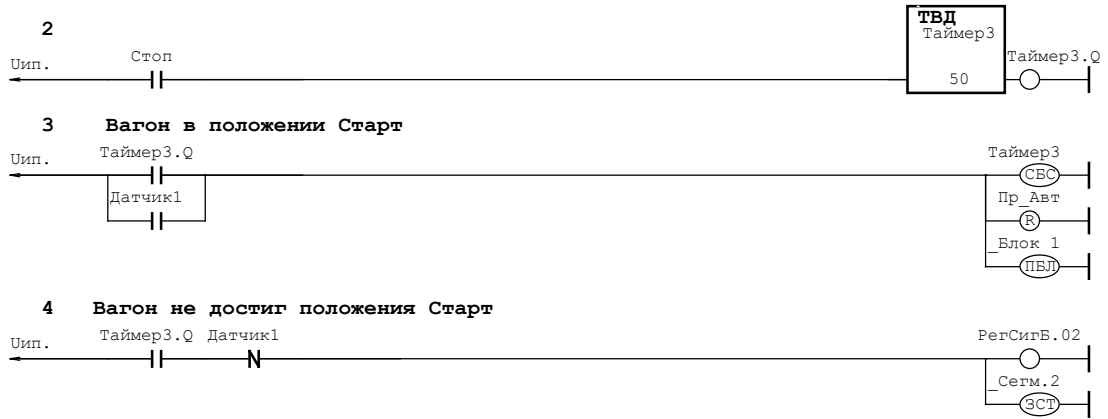


Сегмент № 1 Диагностика системы
Блок № 0 Контроль датчиков

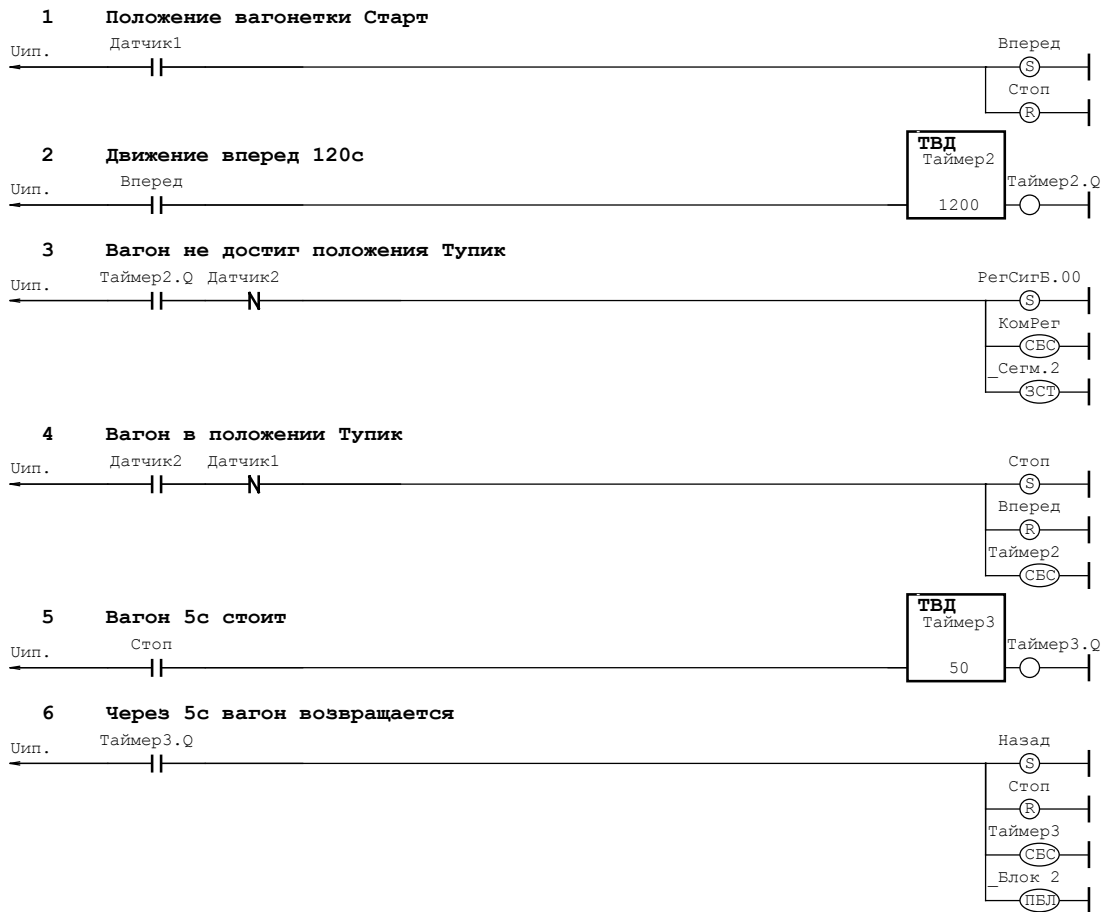


Сегмент № 2 Автоматический режим управления
Блок № 0 Если есть движение, выдать команду СТОП

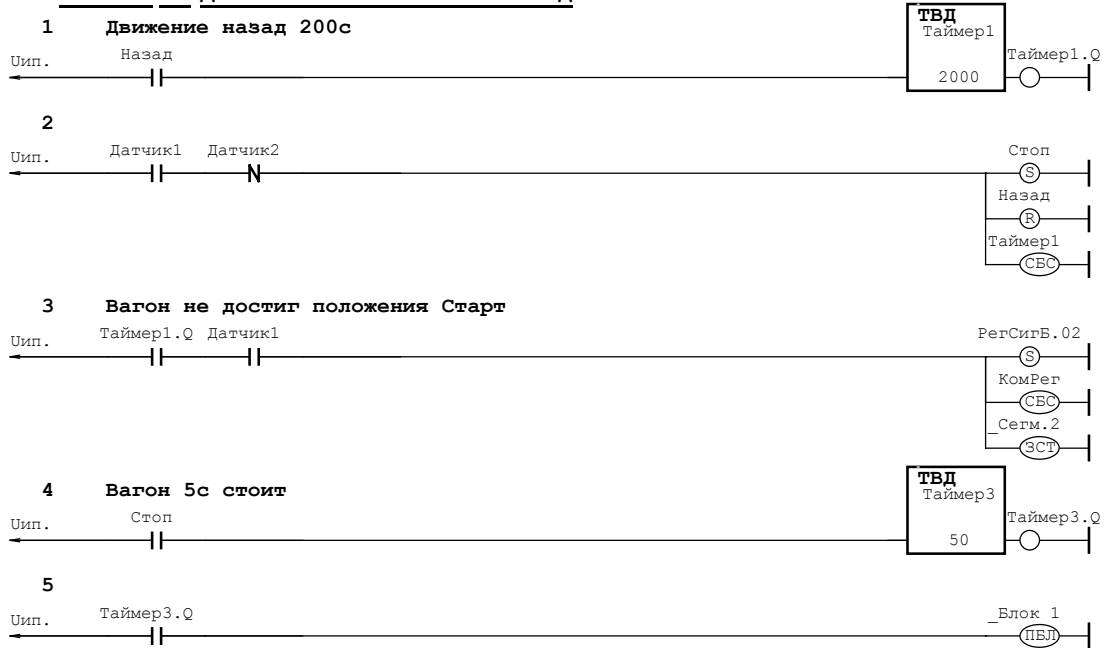




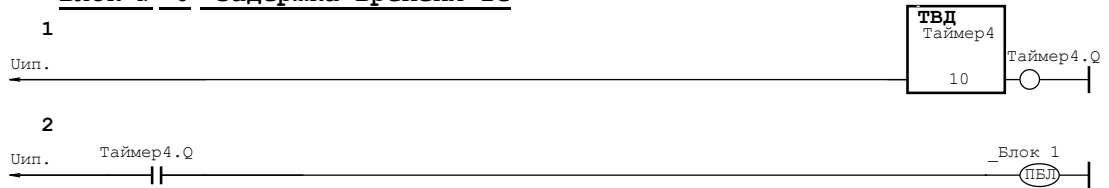
Сегмент № 2 Автоматический режим управления
Блок № 1 Непрерывное движение в режиме Автомат



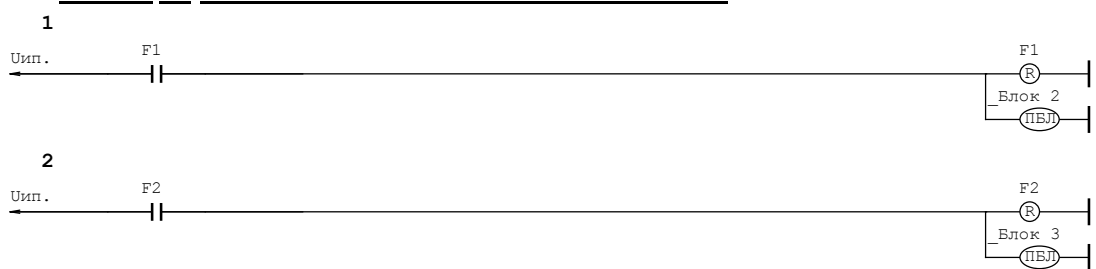
Сегмент № 2 Автоматический режим управления
Блок № 2 Движение вагонетки назад



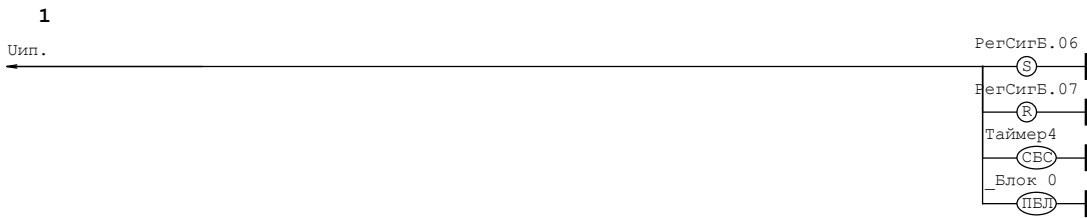
Сегмент № 3 Вывод на панель сообщений и обработка нажатой клавиши
Блок № 0 Задержка времени 1с



Сегмент № 3 Вывод на панель сообщений и обработка нажатой клавиши
Блок № 1 Ожидание нажатия клавиш F1 или F2



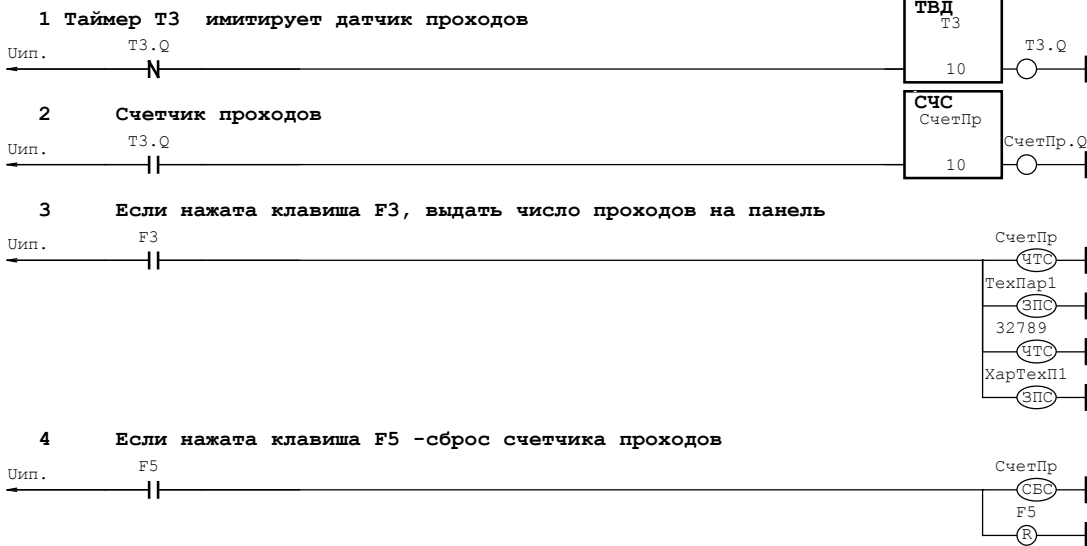
Сегмент № 3 Вывод на панель сообщений и обработка нажатой клавиши
Блок № 2 Установить сигнальные биты



Сегмент № 3 Вывод на панель сообщений и обработка нажатой клавиши
Блок № 3 Нажата клавиша F2



Сегмент № 4 Вывод параметра - число проходов вагонетки
Блок № 0 Подсчет числа проходов вагонетки



Сегмент № 5 Ручной режим управления
Блок № 0 Вывести сообщение РУЧНОЙ РЕЖИМ



