

Учебное пособие предназначено для студентов очной и очно-заочной формы обучения специальности 180400 «Электропривод и автоматика промышленных установок и технологических комплексов» и содержат задания и примеры разработки систем программного управления на базе микроконтроллеров семейства S7-200 фирмы SIEMENS. Задания систематизированы по темам по принципу от простого к сложному. Их выполнение позволит научиться азам программирования микроконтроллеров S7-200, что даст возможность будущему инженеру ориентироваться в огромном объеме информации, посвященной микроконтроллерам различных фирм.

Рецензенты:

кафедра вычислительных систем Международного института компьютерных технологий;  
С.П. Слаута, к.т.н., доцент

ISBN 5 - 88247 - 198 - 2 © Липецкий государственный технический университет, 2005 г.

## Оглавление

ВВЕДЕНИЕ.....	2
Обзор продукции SIMATIC.....	2
Программное обеспечение: .....	2
CPU семейства S7-200.....	3
Машинный цикл CPU состоит из 5 частей: .....	3
Редакторы в STEP7 MicroWin.....	4
Форматы представления данных.....	5
Константы.....	5
Области памяти CPU.....	5
Битовые операции.....	7
Команды сравнения.....	8
Блоки арифметических операций.....	9
Таймеры.....	9
Организация импульсного вывода с помощью быстрых выходов.....	11
ТЕМА 1. Битовые операции, операции сравнения и арифметические операции .....	16
ТЕМА 2. Операции с таймерами и счетчиками, сегментация программ.....	21
ТЕМА 3. Быстрые счетчики, табличные операции, часы.....	25
ТЕМА 4. Импульсные выходы.....	30
ТЕМА 5. PID-операция .....	34
Основные характеристики CPU S7-200 и S7-220.....	38
Идентификаторы областей памяти .....	39
Клавиши панели инструментов STEP 7 - Micro/WIN .....	40
Библиографический список.....	40

## ВВЕДЕНИЕ

Различают 3 уровня автоматизации:

*Полевой* – включает в себя совокупность датчиков, исполнительных механизмов и промышленные контроллеры – это автоматизация отдельных технологических участков.

*Цеховой уровень* – система управления несколькими связанными технологическими процессами.

*Уровень предприятия* – сюда включают еще и систему документирования и визуализации.

### **Обзор продукции SIMATIC.**

- ASI – совокупность аппаратных и программных средств для работы с объектом управления, то есть со сбором информации с датчиков и с выдачей сигналов на исполнительный механизм.
- SIMATIC DP – средство обеспечивающее работу децентрализованной периферии (DP).
- SIMATIC Controller – семейство программируемых логических контроллеров (ПЛК):
  - S7-200 – для малых систем автоматизации.
  - S7-300 – средняя система.
  - S7-400 – крупная система.
- SIMATIC PG, SIMATIC PC – средства для программирования всех основных компонентов автоматизации:
  - программатор PG
  - компьютер PC

### Программное обеспечение:

Для S7 – 200 разработано по STEP7 Micro/Win.

Для S7 – 300 и S7 – 400 по STEP7 в оболочке SIMATIC Manager.

Для систем визуализации (человеко-машинный интерфейс HMI) разработана программа WinCC и PRO TOOL.

Средства HMI – могут быть реализованы или на базе PC, или на базе специальных средств – панелей оператора и обозначаются OP.

SIMATIC NET – средство для работы сетей.

- a) MPI – многоточечный интерфейс.
- b) PROFIBUS – промышленная полевая сеть.
- c) Industrial Ethernet – для организации уровня автоматизации предприятия.

Для семейства **S7-300** и **S7-400** характерна модульная конструкция. ПЛК этих семейств представляет собой «корзину» с набором модулей следующих типов.

- PS – блок питания, бывает встроенный в CPU.
- CPU – центральное устройство, содержит блок МП, память и выполняет программу управления.
- IM – интерфейсный модуль (модуль связи) – для организации работы ПЛК, состоящего из нескольких «корзин» (DP).
- SM – сигнальные модули, есть 4 вида:
  - 1) DI – цифровой вход.
  - 2) DO – цифровой выход.
  - 3) AI – аналоговый вход.
  - 4) AO – аналоговый выход.
- FM – функциональные модули – для решения задач, требующих большего количества времени, но имеющих четко заданный алгоритм (счетчики и т. д.)
- CP – коммуникационный процессор – для организации связи между несколькими ПЛК по сетям следующих видов: Point-to-Point (PTP) – интерфейс для организации связи между двумя абонентами: Industrial, Ethernet, PROFIBUS.

## ***CPU семейства S7-200.***

Проектирование электроустановок на современной элементной базе осуществляется на базе микроконтроллеров, в частности, программируемых логических контроллеров (ПЛК) S7-200 фирмы SIEMENS [1,2] с программным пакетом STEP7 Micro/Win промышленного программного обеспечения SIMATIC. Данное инструментальное средство применяется в самых простых системах автоматизации. Помимо него в SIMATIC входят более сложные программы:

- ✓ STEP7 - пакет программирования для всех систем автоматизации SIMATIC за исключением S7-200;
- ✓ STEP7 Lite - функционально усеченная версия STEP7 для программирования систем автоматизации SIMATIC S7-300 и SIMATIC C7.

Основная структура систем автоматического управления с использованием ПЛК S7-200 (рис. 1) включает в себя:

- центральное устройство S7-200;
- персональный компьютер в качестве программирующего устройства для ПЛК;
- программное обеспечение STEP7-Micro/WIN;
- соединительный кабель, осуществляющий коммутацию персонального компьютера с ПЛК (или несколькими ПЛК).

В свою очередь центральное устройство содержит:

- CPU - центральное программирующее устройство. В таблице П1 приложения приведены основные характеристики CPU для семейств ПЛК S7-200;
- входы - контролируют сигналы приборов (переключателей, датчиков);
- выходы - управляют электроприводами и другими устройствами;
- коммуникационный порт - через него можно подключить к CPU устройство программирования или другие CPU;
- индикаторы состояния - предоставляют визуальную информацию о режиме работы CPU (RUN - рабочий режим; STOP - режим останова), текущем состоянии сигналов встроенных входов и выходов.

Помимо индикации состояния встроенных цифровых входов и выходов на передней панели реализованы переключатели и индикация режимов работы:

STOP – режим, в котором программа не выполняется, но возможна загрузка программы из программатора, загрузка программы из CPU в программатор. При этом сам CPU ничего не считает.

RUN – режим выполнения программы, записанной в CPU.

В S7-200 реализованы следующие устройства расширения:

EM 221 DI8×DC24V – 8 цифровых входов × 24в постоянного тока.

EM 221 DO8×DC24V – 8 цифровых выходов × 24в постоянного тока.

Машинный цикл CPU состоит из 5 частей:

- 1) Опрос состояния физических входов и его запись в соответствующей ячейке области отображения информации на входах.
  - 2) Выполнение программы с начала до конца по принципу «слева направо, сверху вниз».
  - 3) Обработка коммуникационных прерываний (если они есть).
  - 4) Самодиагностика CPU.
  - 5) Установка физических выходов CPU в состояние, соответствующие информации в области отображения информации на выходах.
- 3 и 4 части не зависят от пользователя.

Прямая адресация областей памяти в CPU осуществляется в формате бита (рис. 2,а), байта (рис. 2,б), слова из 16 бит (рис. 2,в) и двойного слова из 32 бит (рис. 2,г). Соотношение форматов иллюстрирует рис. 3. Идентификаторы областей памяти с примерами адресации

приведены в таблице 1.

В CPU S7-200 доступны две основные системы команд:

- SIMATIC - язык программирования микроконтроллеров фирмы SIEMENS;
- IEC 1131-3 - общий для всех микроконтроллеров язык программирования, стандартизованный Международной электротехнической комиссией.

### ***Редакторы в STEP7 MicroWin.***

Программный пакет STEP7 Micro/Win работает с тремя редакторами программ:

**STL - Statement List - Список команд** – визуально напоминает программу на ассемблере. Позволяет создавать управляющие программы в текстовом виде на «родном» языке CPU, состоящем из мнемоник команд (рис. 4,а), а не в графическом редакторе, где должны применяться некоторые ограничения для правильного начертания схемы. Из редакторов LAD и FBD можно перейти в редактор STL автоматически, а программа, написанная в STL, не всегда может быть переведена на язык контактных или функциональных планов. Кроме того, при выборе системы команд IEC 1131-3 редактор списка команд STL недоступен.

**LAD - Ladder Logic - редактор контактного плана** - позволяет формировать программы, имеющие визуальное сходство с релейно-контакторными схемами (рис. 4,б); эти программы эмулируют протекание тока от источника питания через ряд логических состояний входов, которые, в свою очередь, разблокируют логические состояния выходов. Логика подразделяется на малые, легко понимаемые сегменты, или цепи Networks. Программа выполняется по «цепям» слева направо сверху вниз по программе. По достижении конца программы CPU снова начинает ее выполнение с начала в следующем машинном цикле (МЦ). Данный редактор работает с тремя видами графических символов:

1. контакты - представляют логические состояния входов ПЛК, соответствующих выключателям, кнопкам, внутренним маркерам программы;
2. катушки - представляют логические результаты на выходах ПЛК, к которым подключены исполнительные объекты (лампы, пускатели электродвигателей), либо состояние внутренних маркеров;
3. блоки - дополнительные команды такие, как таймеры, счетчики, математические команды.

Редактор LAD можно использовать и в системе команд SIMATIC, и в системе команд IEC 1131-3. А для отображения программы, созданной при помощи редактора SIMATIC LAD, всегда можно использовать редактор STL.

**FBD - Function Block Diagram - редактор функционального плана** - позволяет формировать программы, имеющие визуальное сходство с логическими схемами (рис. 4, в). В нем не используются ни контакты, ни катушки, как в редакторе LAD, но имеются эквивалентные им команды в форме блоков. Так, последовательному соединению контактов в LAD здесь соответствует подача сигналов на вход блока логического умножения AND. Нормально замкнутому контакту в LAD соответствует инверсный вход блока в FBD, параллельному соединению - логическое сложение сигналов на блоке OR. Как и LAD, FBD можно использовать и с системой команд SIMATIC, и с системой команд IEC 1131-3. А для отображения программы можно использовать редактор STL. Редактор списка команд STL является предпочтительным для более опытных пользователей, а FBD и LAD - для начинающих.

Переход из одного редактора в другой осуществляется с помощью меню «Вид».

Примеры записи программ в этих редакторах показаны на рис. 4.

### Форматы представления данных.

Шина данных CPU 32 разрядная, поэтому возможны 4 формата обращения:

1. Формат 2-го слова: обращение сразу к 32 разрядам числа.  
Обозначение этого формата : **#D#**
2. Формат слова: обращение к 16 разрядному числу.  
Обозначение этого формата : **#W#**
3. Формат байта: обращение к 8 разрядному числу.  
Обозначение этого формата : **#B#**
4. Формат бита: обращение к 1 разрядной информации.  
Обозначение этого формата : **##.#**

Рассмотрите более подробно на рис. 2 обозначение форматов представления данных в STEP7 MicroWin.

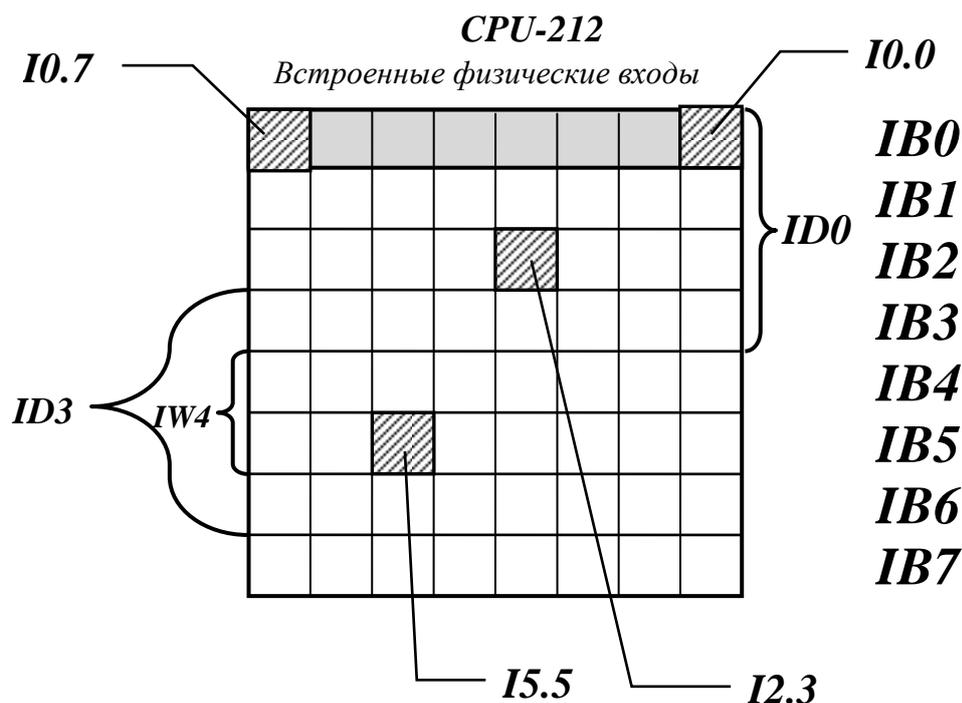
Соотношения этих 4 форматов подробно показаны на рис. 3.

### Константы.

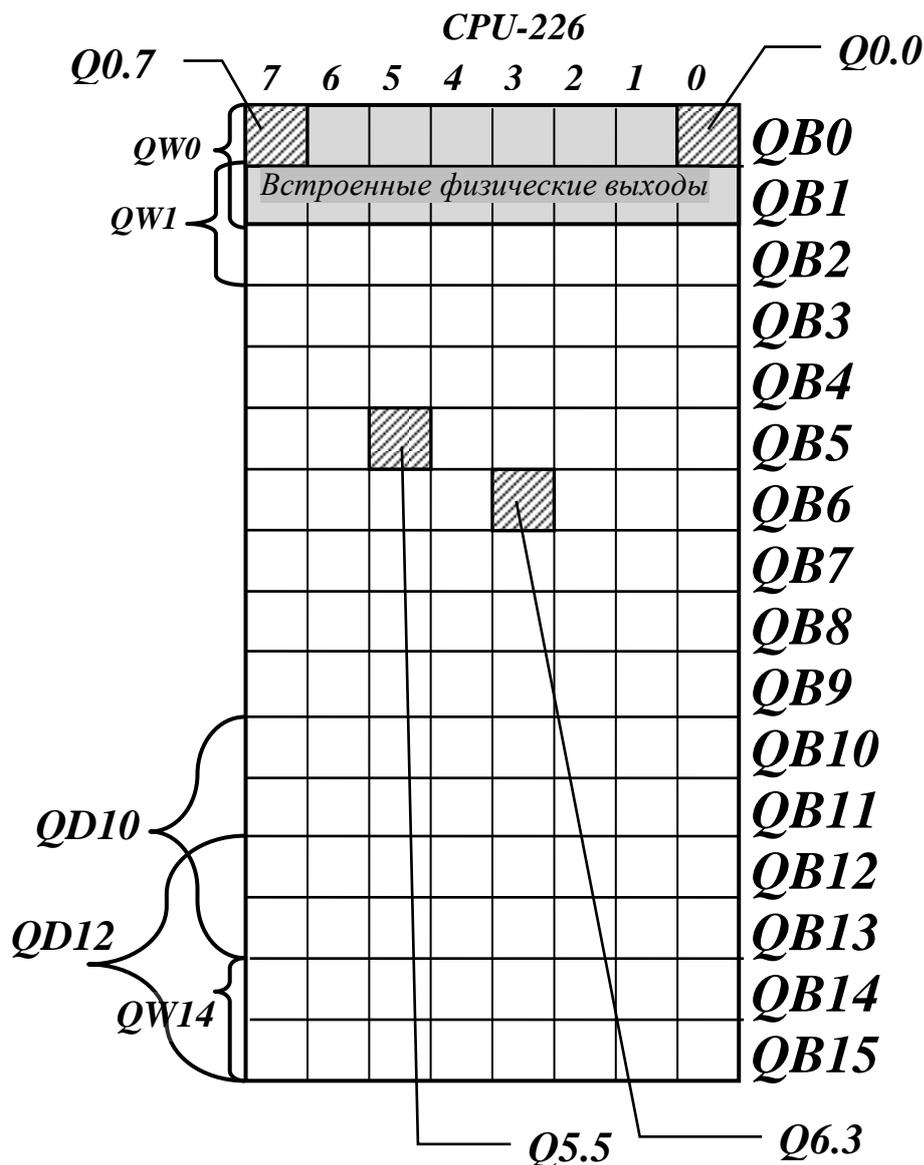
1. Десятичные – пишутся без всяких дополнительных обозначений.  
Пример: **127; 255.**
2. Шестнадцатеричные константы обозначаются символами **16#**.  
пример: **16#58 16#9A 16#BF.**
3. Двоичные константы обозначаются символами **2#**, при этом пишутся только значащие цифры.  
пример: **2#10111**

### Области памяти CPU.

1. Область отображения информации на входах. Обозначение **I** (*Input*)  
Предназначена для записи информации о состоянии цифровых входов. Для CPU-212 эта область представляет собой матрицу  $8 \times 8 = 64$  бита. (см. в таблице П1 **64DI**)  
При этом встроенные физические цифровые входы 8 бит (см. в таблице П1 **8DI**)



2. Область отображения информации на выходах. Обозначение **Q** (Output)  
 Предназначена для записи информации о состоянии цифровых выходов. Для CPU-226 эта область представляет собой матрицу 16×8=128 бита. (см. в таблице П1 **128DO**)  
 При этом встроенные физические цифровые выходы 16 бит (см. в таблице П1 **16DO**)



3. Область памяти переменных. Обозначение **V**.

Помимо хранения переменных данная область используется для организации работы с таблицами, для хранения профиля РТО и т.д.

4. Внутренние маркеры. Обозначение **M**

Используется для хранения промежуточных результатов вычислений.

5. Область для организации структурирования программ. Обозначение **S**.

При большом объеме управляющей программы удобно поделить ее на отдельные технологически обоснованные шаги. Начало каждого шага обозначается S битом. Если этот бит равен 1, то CPU выполняет шаг. Если 0, то полностью его игнорирует.

6. Область специальных маркеров. Обозначение **SM**

Предназначена для организации специальных функций CPU (быстрые входы, быстрые выходы и т.д.) Кроме того есть отдельные специфические биты: SM0.0 - бит всегда равный 1, используется для безусловного выполнения блоков. SM0.1 – равен 1 только в первом машинном цикле CPU.

7. Область таймеров. Обозначение **T**.  
Служит для организации отсчета временных интервалов.
8. Область синхронных счетчиков. Обозначение **C**.  
Служит для подсчета указанных в программе событий.
9. Область аналоговых входов. Обозначение **AI**.  
Область предназначена для сохранения цифрового кода, полученного с устройства расширения, реализующего АЦП.
10. Область аналоговых выходов **AO**.  
Предназначена для хранения цифрового кода, который будет передан на устройство расширения, осуществляющее ЦАП.
11. Область аккумуляторов **AC**.  
У CPU имеется в наличии 4 32- разрядных аккумулятора AC0, AC1, AC2, AC3. Адресация в них возможна в форматах:  
Двойного слова – изменяются все 32 разряда.  
Слова – изменяются младшие 16 разрядов аккумулятора.  
Байта - изменяются младшие 8 разрядов аккумулятора.  
Такое обращение называется принципом выравнивания информации по правому краю.
12. Область быстрых счетчиков **HC**.  
Эта область содержит текущие значения быстрых счетчиков.

### **Битовые операции.**

Располагаются в папке **Bit Logic** дерева инструкций.

1. Простые контакты.

##.# - любой адрес в формате бита.



Контакт считается замкнутым, когда указанный бит равен 1.

2. Нормально замкнутый контакт

##.# - любой адрес в формате бита.



Контакт считается замкнутым, когда указанный бит равен 0.

3. Инверсия или отрицание



4. Выделение положительного (переднего) фронта сигнала.

Контакт замкнут в течении 1 машинного цикла после поступления переднего фронта сигнала на данный контакт.

5. Выделение отрицательного фронта сигнала.

Контакт замкнут в течении 1 машинного цикла после поступления отрицательного фронта сигнала на данный контакт.

6. Простая катушка или присваивание.

##.# Результат выполнения операции присваивается биту с указанным адресом.

7. Установка нескольких бит в 1 "SET".

—  $\begin{matrix} \text{##.}\# \\ \text{( S )} \end{matrix}$  - начальный адрес.  
 $n$  – количество устанавливаемых в единицу бит.

$\begin{matrix} \text{M0.6} \\ \text{( S )} \\ 4 \end{matrix}$  - установка в 1 бит: M0.6; M0.7; M1.0; M1.1

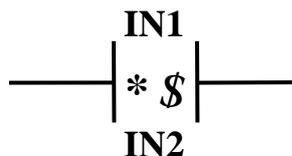
8. Сброс нескольких бит в 0 "RESET".

—  $\begin{matrix} \text{##.}\# \\ \text{( R )} \end{matrix}$  - начальный адрес.  
 $n$  – количество сбрасываемых в 0 бит.

—  $\begin{matrix} \text{S5.5} \\ \text{( R )} \\ 5 \end{matrix}$  - сброс в 0 бит: S5.5; S5.6; S5.7; S6.0; S6.1

**Команды сравнения.**

Все команды сравнения собраны в папке *Compare* дерева инструкций. Они представляют собой контакт, который считается замкнутым при выполнении указанного условия.



\* - знак операции сравнения ( >=; <=; >; <; <>; == ).

\$ - формат сравниваемых данных:

**B** – формат байта без учета знака числа – сравниваются положительные числа от 0 до 255.

**I** – сравнение 16 разрядных чисел с учетом их знака.

Максимальное положительное число:  $\underline{0}111.1111.1111.1111_2$   
 знак

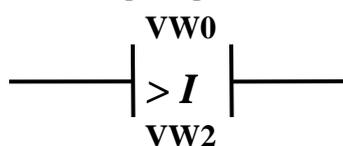
Следующее число по числовому ряду:  $\underline{1}000.0000.0000.0000_2$  или  $8000_{16}$  считается меньше, чем предыдущее так как оно является *отрицательным*.

**D** – сравнение 32 разрядных чисел с учетом их знака.

**R** – сравнение действительных чисел.

**IN1** и **IN2** - данные для сравнения в указанном формате.

Пример:



Контакт замкнут, когда содержимое VW0 больше, чем VW2.

## Блоки арифметических операций.

Все блоки арифметических операций собраны в папке *Integer Math* дерева инструкций. Обозначение операций.

ADD – сложение.

SUB – вычитание.

MUL – умножение.

DIV – деление.

Блоки арифметических операций, как и все остальные блоки (за исключением таймеров и синхронных счетчиков) содержит пару разрешающих сигналов.

**EN** – входной разрешающий сигнал

EN=1 - выполнение блока.

EN=0 – данный блок и все последующие в сети не выполняются

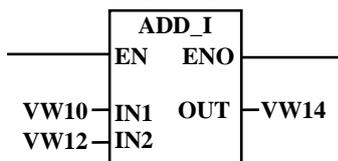
**ENO** – выходной разрешающий сигнал, который устанавливается в 1 при корректном завершении выполнения блока.

Если работа блока завершена некорректно, то CPU переходит в режим “STOP”, а в режиме “Program status” этот блок выделяется красным цветом.

**IN1, IN2** – входы, на которые подаются исходные значения операндов, над которыми производится арифметическая операция. Формат представления их должен совпадать с форматом блока арифметических операций, в противном случае они подчеркиваются красным цветом в редакторе LAD, и в дальнейшем при компиляции будет сообщение об ошибке.

**OUT** – выходной результат вычислений, который помещается в нужную ячейку памяти.

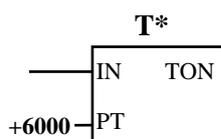
Пример:



## Таймеры.

Все таймеры располагаются в папке *Timers* дерева инструкций, которая содержит следующие команды.

### 1. Таймер задержки включения без запоминания. Функция **TON**.



\* - номер таймера из допустимого диапазона для каждого типа CPU. Например, для CPU 226: от T0 до T255.

IN – разрешение на счет.

PT – предварительно установленное значение задержки включения.

Каждый таймер работает с двумя областями памяти:

- 1) Бит таймера
- 2) Текущее значение таймера в формате слова.

Диапазон текущего и предварительно установленного значения составляет от нуля до 32767, то есть из 16 разрядов используется 15.

Если на вход IN подается 1, то начинается изменение текущего значения таймера через промежутки времени  $\Delta t$ , которые называются разрешающей способностью таймера и определяются номером таймера и его функцией.

Например, для таймера **TONR T30** разрешающая способность составляет 100 мс.

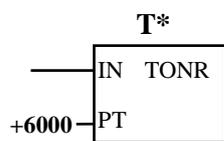
1 ms – T0, T64

10ms – T1-T4, T65-T68

100ms – T5-T31, T69-T95.



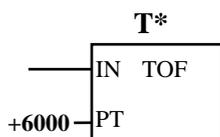
### 2. Таймер задержки включения с запоминанием. Функция **TONR**.



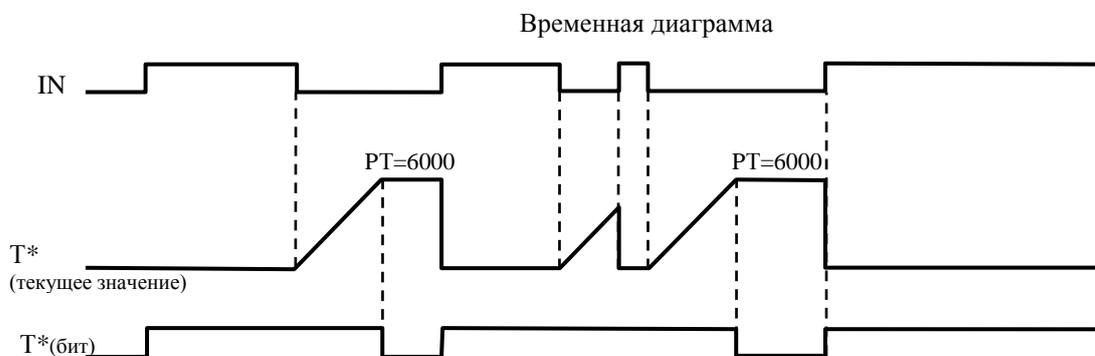
Когда выключается разрешающий вход IN, значение Таймера сохраняется. Вы можете использовать Таймер для накопления времени за несколько периодов, когда включен разрешающий вход. Для стирания текущего значения Таймера используется команда Сброс(R).



### 3. Таймер задержки выключения **TOF** реализован в CPU, начиная с CPU 221.



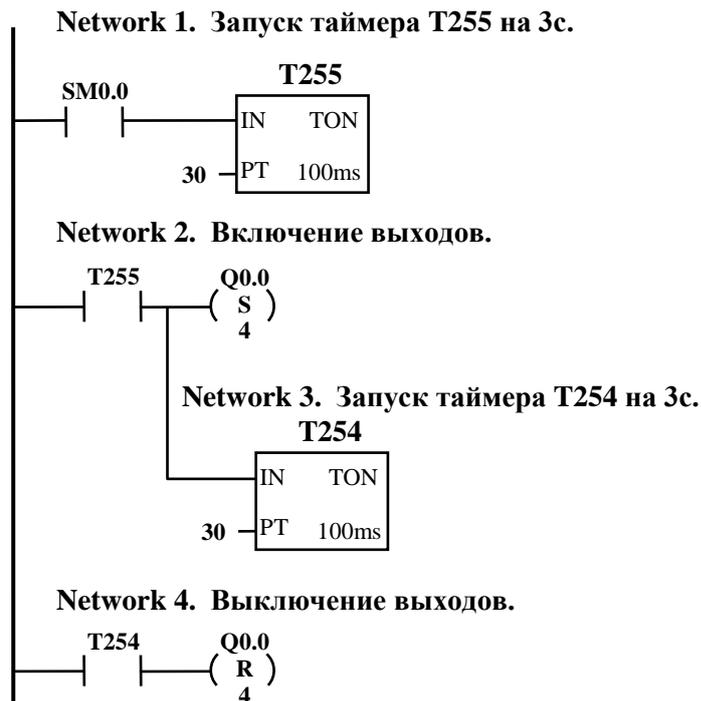
Используется для увеличения времени после сбойных ситуаций, например для охлаждения двигателей после их отключения.



PT – предварительно установленное значение таймера.

Пример задачи: Через 3с после запуска программы включить все физические входы, а через 6 секунд выключить.

$$PT_{255} = \frac{t_1}{\Delta t} = \frac{3}{100 \cdot 10^{-3}} = 30$$

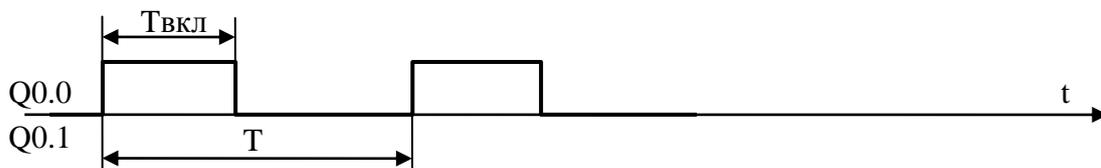


### Организация импульсного вывода с помощью быстрых выходов.

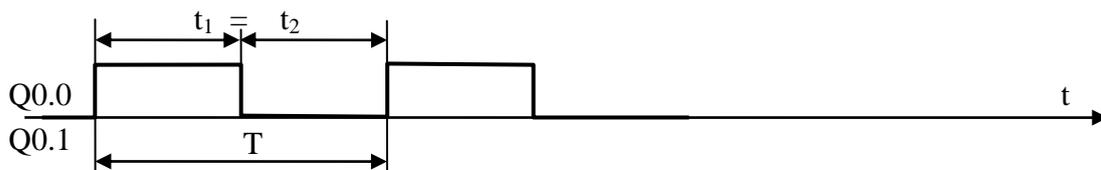
В качестве быстрых выходов используются физические выходы CPU Q0.0 и Q0.1, которые работают независимо друг от друга и программируются одинаково.

Каждый CPU имеет два генератора: РТО - для вывода конечной последовательности импульсов и PWM для управления широтно-импульсной модуляцией. Когда функция РТО или PWM активна на Q0.0 или Q0.1, то выходом управляет генератор РТО или PWM, а нормальное использование выхода заблокировано. На форму выходного сигнала не влияет ни присваивание значения выходам, ни выполнения команд непосредственного вывода.

1. Функция PWM (Pulse width modulate). Выдача бесконечной последовательности импульсов с возможностью задания времени включения (ширина импульса).



2. Функция РТО (Pulse train output). Выдача конечной последовательности импульсов с относительной длительностью включения 50%.



Для этих функций задаются 3 параметра:

1. Период следования импульсов T (для PTO и PWM) в области специальных маркеров SMW68(78) для Q0.0(Q0.1).
2. Время включения (ширина импульса) SMW70(80) для Q0.0(Q0.1). Задается только для PWM.
3. Количество импульсов (только для PTO). В SMD72(82) для Q0.0(Q0.1).

Программирование импульсного режима осуществляется через задание управляющего байта в специальном формате, который указан в таблице. Управляющие байты записываются по адресам SMB67(77) для Q0.0(Q0.1).

Передача параметров импульсов и инициализация работы быстрого выхода осуществляется с помощью блока PLS. (пример см. на рис. 13)



### Управляющие регистры PTO/PWM

Q0.0	Q0.1	Биты состояния
SM66.4	SM76.4	Профиль PTO завершен из-за ошибки расчета приращения 0- нет ошибки 1- завершен
SM66.5	SM76.5	Профиль PTO завершен по команде пользователя. 0- нет завершения 1- завершен
SM66.6	SM76.6	Переполнение (потеря значимости) в конвейере PTO. 0-нет переполнения 1- переполнение
SM66.7	SM76.7	PTO не действует. 0- действует 1- не действует
Q0.0	Q0.1	Управляющие биты
SM67.0	SM77.0	Обновление значения периода следования импульсов. 0- не обновлять 1- обновлять
SM67.1	SM77.1	Обновит значение ширины импульсов. 0- не обновлять 1- обновлять
SM67.2	SM77.2	Обновить количество импульсов PTO. 0- не обновлять 1- обновлять
SM67.3	SM77.3	Выбрать базу времени PTO/PWM. 0=1мкс/такт 1=1мс/такт
SM67.4	SM77.4	Метод обновления PWM. 0- асинхронный 1- синхронный
SM67.5	SM77.5	Режим PTO: 0- односегментный 1- многосегментный
SM67.6	SM77.6	Выбрать PTO/PWM: 0- выбирает PTO 1- выбирает PWM
SM67.7	SM77.7	Разблокировка PTO/PWM 0- блокирует 1- разблокирует
Q0.0	Q0.1	Другие значащие биты
SMW68	SMW78	Значение периода следования импульсов (2-65535)
SMW70	SMW80	Ширина импульсов PWM (2-65535)
SMD72	SMD82	Количество импульсов PTO (4 294 967 295)
SMB166	SM B176	Номер действующего сегмента (в многосегментном режиме PTO)
SMW168	SMW178	Начальный адрес таблицы профиля

Для того чтобы во время выполнения программы определить момент завершения конечного режима PTO необходимо считать бит **SM66.7(76.7)**.

Для того чтобы во время выполнения программы остановить бесконечный режим PWM необходимо сбросить бит разблокировки PTO/PWM **SM67.7(77.7)** и затем применить блок PLS.

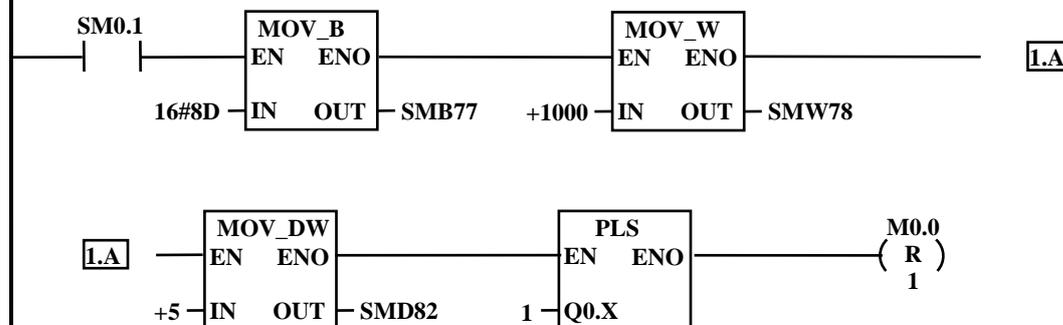
Задача: На выходе Q0.1 организовать выдачу 10 импульсов с параметрами:

1...5 – T=1с, время включения 0,5с.

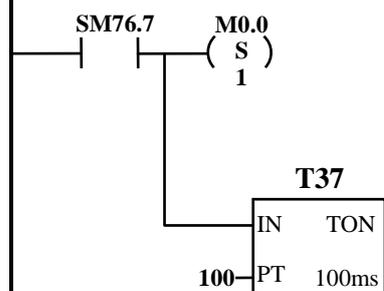
6...10 – T=2с, время включения 0,75с.

Решение: Для организации 1...5 импульсов используем режим PTO, затем для 6...10 импульсов включим бесконечный режим PWM. Чтобы определить момент выключения режима PWM после окончания 6...10 импульсов включим таймер на расчетное время длительности этих импульсов:  $T_{\text{таймера}} = 5 * 2 = 10\text{с}$ .

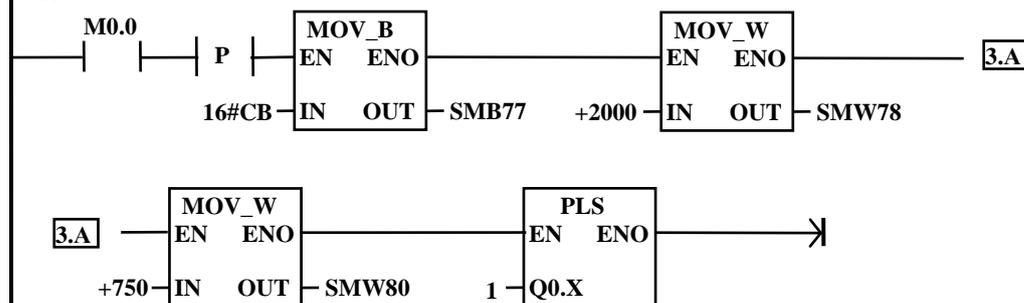
**Network 1** Задание в МЦП первоначальных параметров последовательности импульсов в режиме PTO - 5 импульсов T=1с и включение ее.



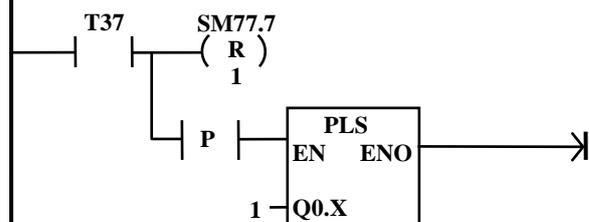
**Network 2.** После окончания режима PTO установка метки и включение таймера на 10с.



**Network 3.** Загрузка и включение режима PWM T=2с Tвкл=0,75с по переднему фронту появления сигнала на метке M0.0.



**Network 4.** После срабатывания таймера сброс бита разблокировки PWM и остановка PLS.



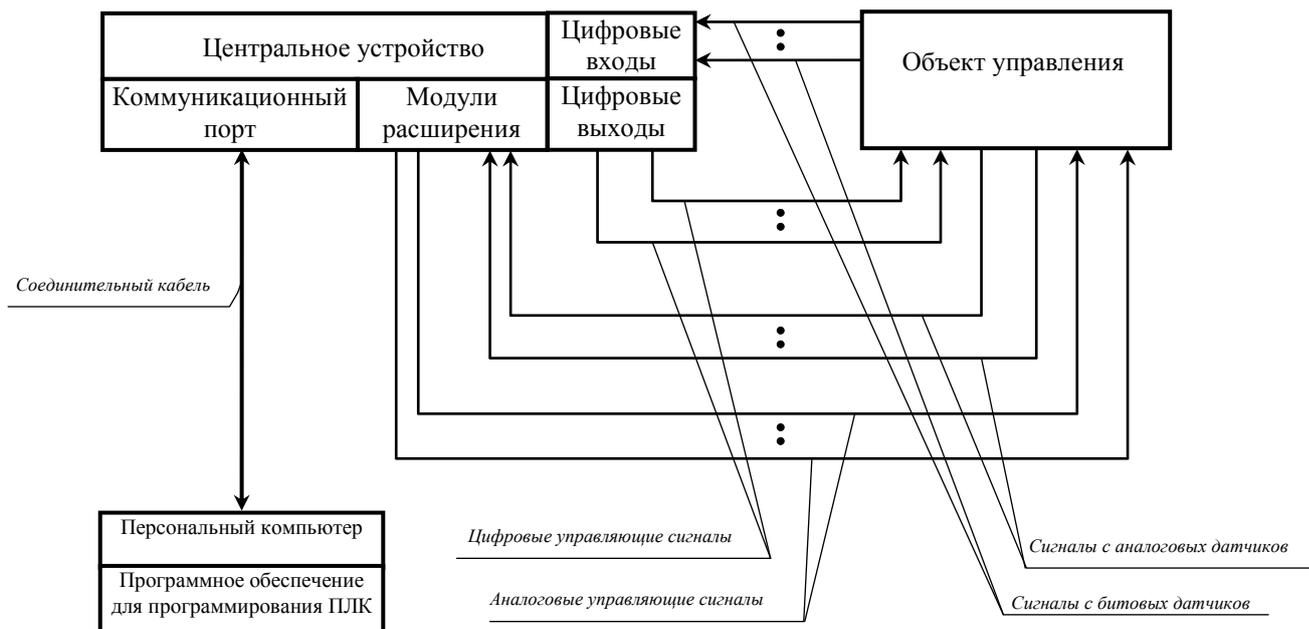


Рис. 1. Структурная схема системы управления электроустановкой полевого уровня на базе одного ПЛК.

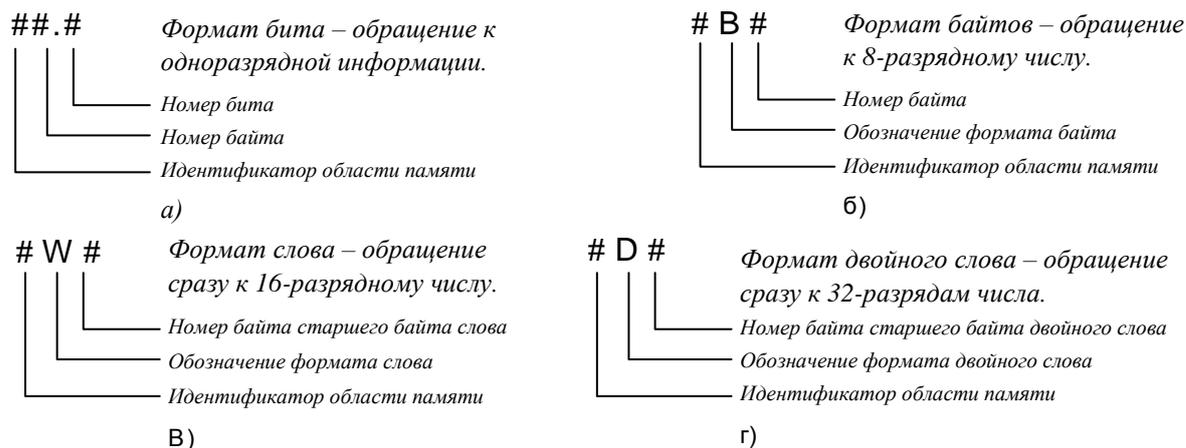


Рис. 2. Форматы прямой адресации STEP7-Micro/WIN:  
 а - формат бита; б - формат байта; в - формат слова; г - формат двойного слова

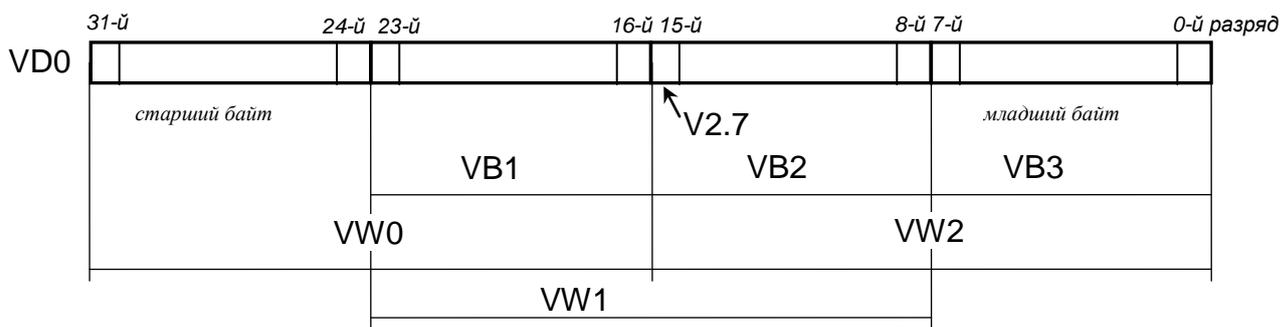


Рис. 3. Соотношение форматов прямой адресации.

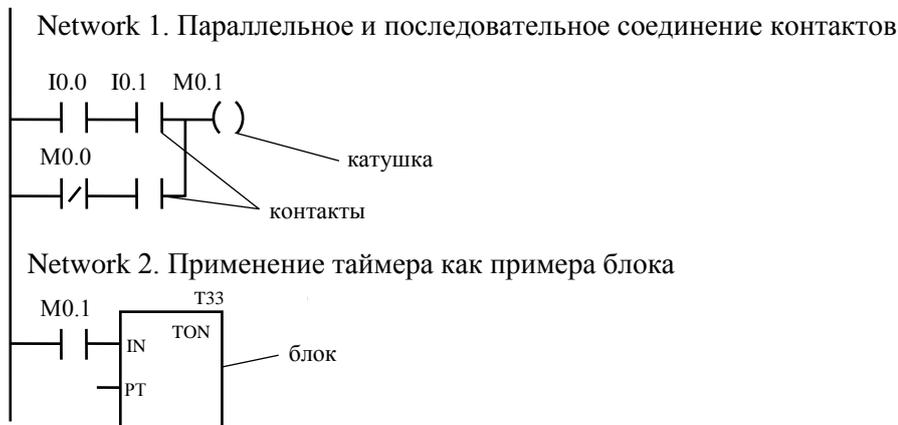
Пример программы в редакторе списка команд

**Network 1.** Параллельное и последовательное соединение контактов  
LD I0.0 / Загрузка вершины стека из I0.0  
A I0.1 / Логическое умножение I0.0 & I0.1  
LDN M0.0 / Загрузка вершины стека величиной M0.0  
A T1 / Логическое умножение вершины стека на T1  
OLD / Логическая ИЛИ  
= M0.1 / Вершина стека копируется в M0. 1

**Network 2.** Применение таймера как примера блока  
LD M0. 1 / Загрузка вершины стека в M0. 1  
TON T33,+135 / Таймер T33 с PT= 135

а)

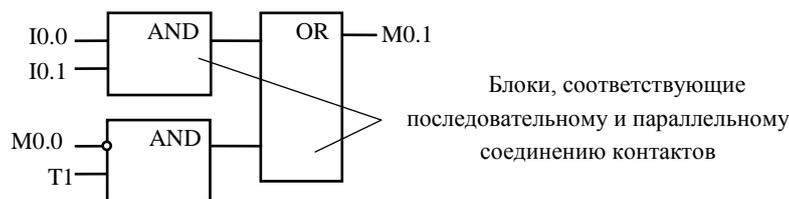
Пример программы в редакторе контактных планов



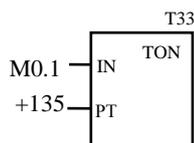
б)

Пример программы в редакторе функциональных планов

**Network 1.** Параллельное и последовательное соединение контактов



**Network 2.** Применение таймера как примера блока



в)

Рис.4. Пример программы на SIMATIC STEP 7 - Micro/WIN:  
а - в редакторе STL; б - в редакторе LAD; в - в редакторе FBD

## ТЕМА 1. Битовые операции, операции сравнения и арифметические операции

**Задание:** написать, отладить и скомпилировать программу в редакторе контактных (LAD) планов в соответствии с алгоритмом на рис. 5 и таблицей 1.

### Пример решения для варианта 1

Заданный для варианта 1 алгоритм приведен на рис. 6, а программа в редакторе LAD - на рис.7.

#### Варианты задания по теме 1.

Таблица 1.

№	Условие	Формула 1	Адреса переменных			Формула 2	Формула 3	Адреса переменных			
			x	y	k			a1	b1	a2	a3
1	$x > y$	$(x+y) \cdot 2 = k$	VB0	VB1	VB3	$k+1 \rightarrow k$	$(\overline{a1} \& b1) \vee a2 = a3$	M0.0	M1.0	I0.0	Q0.0
2	$x < y$	$(x-y) \cdot 7 = k$	VW0	VW10	VW6	$k-1 \rightarrow k$	$(\overline{a1} \& \overline{b1}) \vee a2 = a3$	CO	I0.1	CI	Q0.2
3	$x = y$	$(x-y) : 5 = k$	VD0	VD10	VD18	$k+1 \rightarrow k$	$(\overline{a1} \& \overline{b1}) \vee a2 = a3$	I0.0	T0	T1	M0.0
4	$x \geq y$	$(x+y) \cdot 6 = k$	VB0	VB1	VB2	$k+10 \rightarrow k$	$\overline{a1} \& (\overline{b1} \vee a2) = a3$	I2.0	I0.1	C5	M0.1
5	$x \leq y$	$(x+y) : 8 = k$	VW0	VW10	VW4	$k-10 \rightarrow k$	$\overline{a1} \& (b1 \vee a2) = a3$	I0.3	I0.4	CI	Q0.2
6	$x < > y$	$(x-y) \cdot 9 = k$	VD0	VD10	VD6	$k \cdot 11 \rightarrow k$	$(a1 \& b1) \vee a2 = a3$	C2	I0.5	CI	M0.3
7	$x > y$	$(x+y) : 10 = k$	VB0	VB1	VB7	$k+10 \rightarrow k$	$(\overline{a1} \& b1) \vee a2 = a3$	TI	I0.7	C24	Q0.4
8	$x < y$	$(x+y) \cdot 21 = k$	VW0	VW10	VW8	$k-12 \rightarrow k$	$(\overline{a1} \vee \overline{b1}) \& \overline{a2} = a3$	I0.0	I0.1	C14	M0.4
9	$x = y$	$(x-y) : 22 = k$	VD0	VD10	VD4	$k+17 \rightarrow k$	$\overline{a1} \vee (\overline{b1} \& \overline{a2}) = a3$	I0.0	I0.2	T33	Q0.5
10	$x \geq y$	$(x+y) \cdot 23 = k$	VB0	VB1	VB4	$k+19 \rightarrow k$	$(\overline{a1} \vee \overline{b1}) \& a2 = a3$	CI	I0.4	Q0.0	M1.0
11	$x \leq y$	$(x-y) : 24 = k$	VW0	VW10	VW6	$k-13 \rightarrow k$	$(\overline{a1} \vee b1) \& \overline{a2} = a3$	CI	I0.5	Q0.2	Q1.1
12	$x < > y$	$(x+y) \cdot 25 = k$	VD0	VD10	VD38	$k \cdot 18 \rightarrow k$	$(\overline{a1} \vee b1) \& \overline{a2} = a3$	C24	I0.1	M0.0	M1.2
13	$x > y$	$(x-y) : 26 = k$	VB0	VB1	VB2	$k+6 \rightarrow k$	$\overline{a1} \vee (b1 \& a2) = a3$	C14	I0.1	M0.1	M1.3
14	$x < y$	$(x+y) \cdot 27 = k$	VW0	VW10	VW4	$k-16 \rightarrow k$	$(a1 \vee b1) \& a2 = a3$	T33	I0.1	Q0.2	M0.4
15	$x = y$	$(x-y) : 28 = k$	VD0	VD10	VD6	$k+17 \rightarrow k$	$(\overline{a1} \vee \overline{b1}) \& \overline{a2} = \overline{a3}$	Q1.0	I0.1	M0.3	Q0.0
16	$x \geq y$	$(x+y) \cdot 29 = k$	VB0	VB1	VB7	$k+12 \rightarrow k$	$\overline{a1} \vee (\overline{b1} \& \overline{a2}) = \overline{a3}$	I0.5	I0.1	CI	Q0.1
17	$x \leq y$	$(x-y) : 30 = k$	VW0	VW10	VW8	$k-18 \rightarrow k$	$(\overline{a1} \vee \overline{b1}) \& a2 = \overline{a3}$	I0.6	I0.1	CI	Q0.2
18	$x < > y$	$(x+y) \cdot 32 = k$	VB0	VB1	VB4	$k \cdot 13 \rightarrow k$	$(\overline{a1} \vee b1) \& \overline{a2} = \overline{a3}$	I0.7	I0.1	CI	Q0.3
19	$x > y$	$(x+y) : 6 = k$	VW0	VW10	VW6	$k+19 \rightarrow k$	$\overline{a1} \vee (b1 \& \overline{a2}) = \overline{a3}$	CI	Q0.2	C48	Q0.5
20	$x < y$	$(x-y) \cdot 71 = k$	VD0	VD10	VD48	$k-18 \rightarrow k$	$(\overline{a1} \vee b1) \& \overline{a2} = \overline{a3}$	CI	M0.0	Q0.2	M0.1

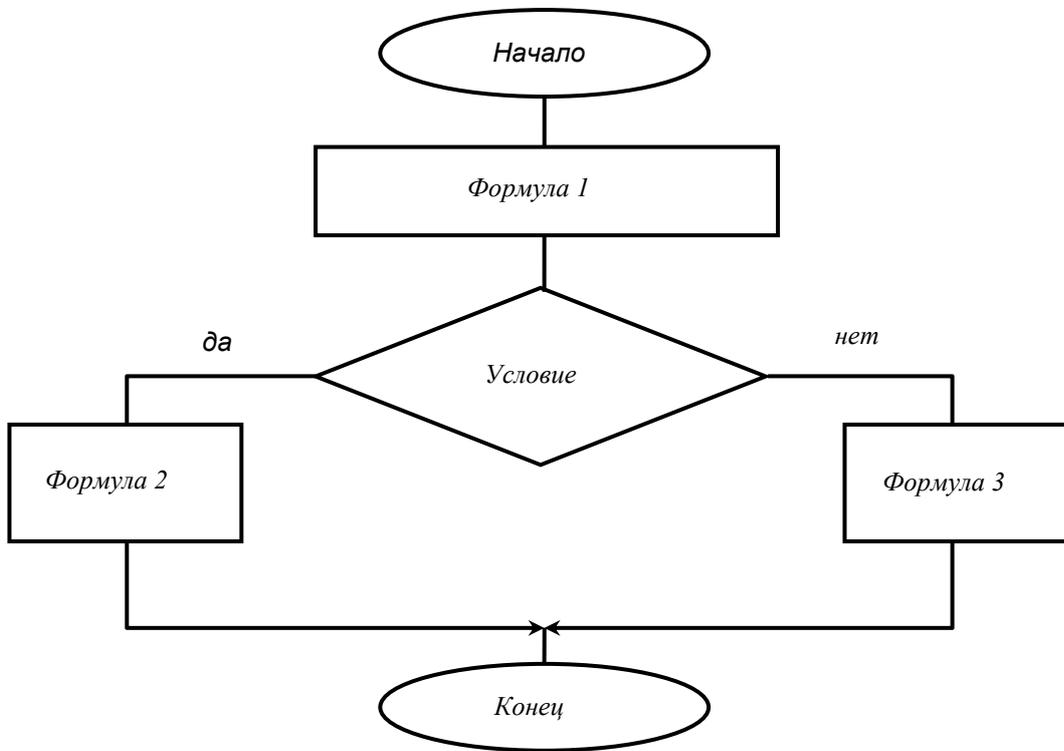


Рис. 5. Алгоритм для программы в задании 1

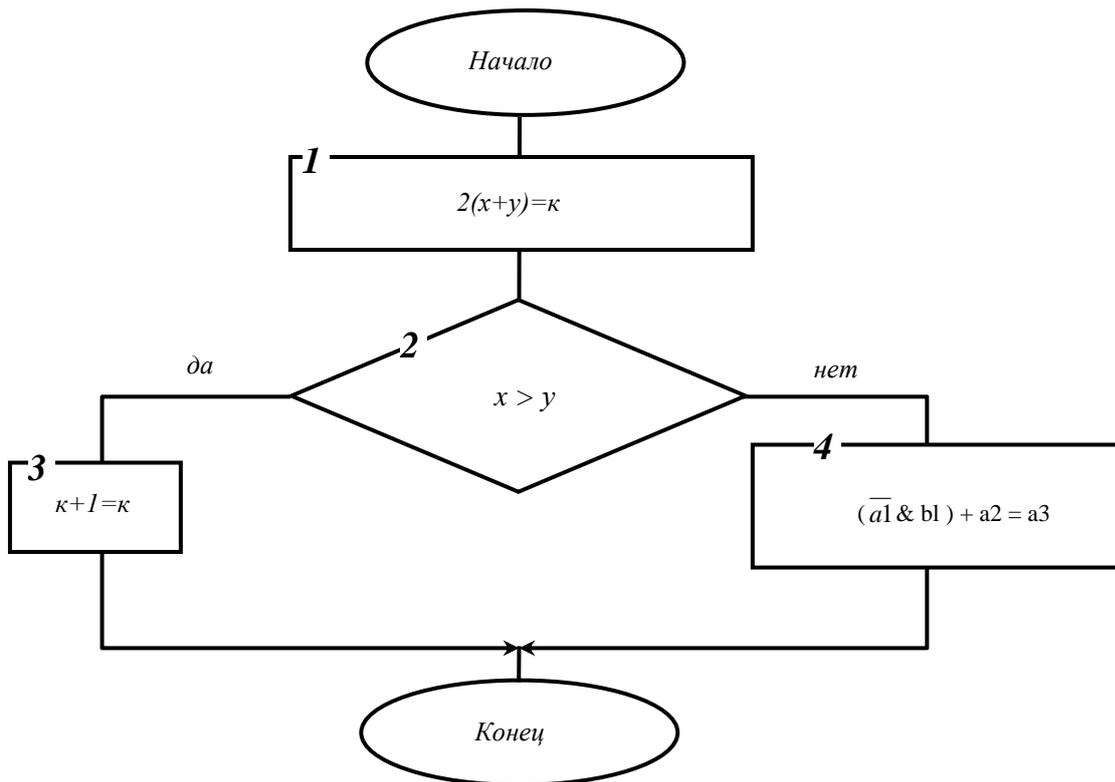
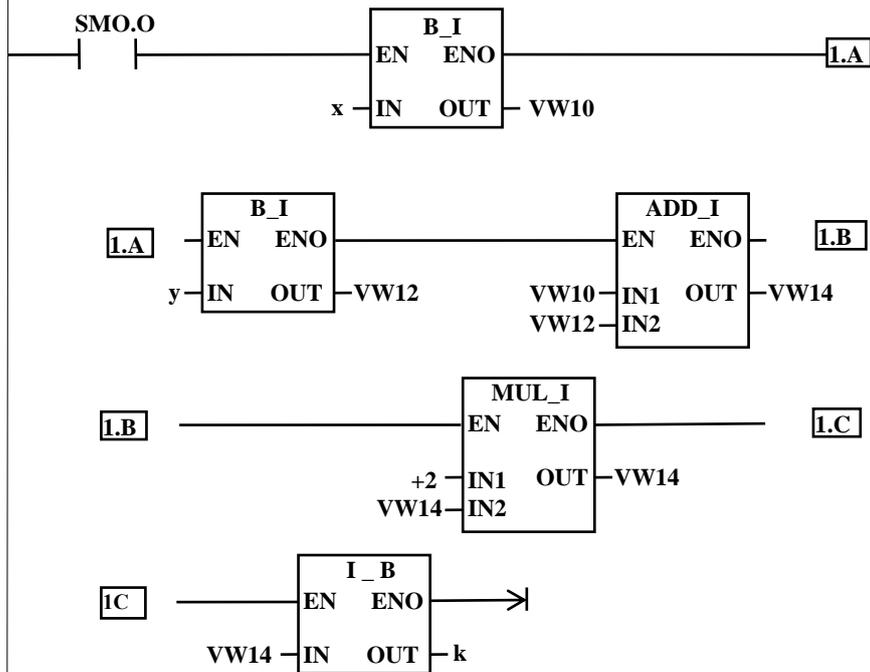
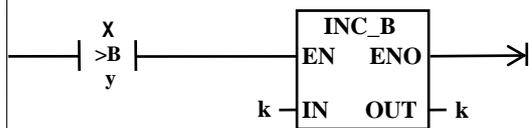


Рис. 6. Алгоритм для варианта 1в задании 1.

**Задание 1 вариант 1**  
**Network 1 Блок 1 алгоритма**  
**Преобразование данных в формат integer для вычислений по формуле 1**



**Network 2 Если X>Y, выполнить блок 3**



**Network 3 Блоки 2 и 4 алгоритма**

Если не выполняется  $x > y$ , выполнить битовые операции формулы 3

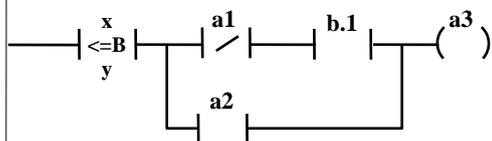


Рис. 7. Программа для примера решения варианта 1 по теме 1.

При разработке проекта, прежде всего, необходимо заполнить таблицу символьных имен в соответствии с таблицей 2.

Таблица 2 Таблица символьных имен

Символ	Адрес	Комментарий
<i>x</i>	VB0	К формуле 1
<i>y</i>	VB1	К формуле 1
<i>к</i>	VB3	К формуле 1
<i>a1</i>	M0.1	К формуле 3
<i>b1</i>	M1.0	К формуле 3
<i>a2</i>	I0.0	К формуле 3
<i>a3</i>	Q0.0	К формуле 3

Далее нужно написать программу, соответствующую алгоритму на рис. 6. В блоке 1 должно осуществляться сложение и умножение данных в формате байта. Такими операциями STEP7 Micro/Win не располагает, поэтому вычисления осуществляются в формате Integer. В этой связи требуется преобразование переменных из формата байта в формат Integer с помощью соответствующего блока, выбираемого из папки Converts (Преобразования). Еще одна особенность редактора контактных планов - блок не может быть, безусловно соединен с левой силовой шиной. Эта задача решается посредством специального маркера SM0.0, величина которого постоянно установлена в 1.

Последовательность действий такая:

1. Через ярлык Program Block левого поля экрана вывести на экран поле редактора контактных планов SIMATIC LAD.
2. Записать первую сеть Network 1. Для этого нужно:
  - получить полный список папки битовых операций Bit Logic;
  - выбрать операцию обычного нормально разомкнутого контакта  $\uparrow \downarrow$  однократным нажатием левой кнопки мыши (ЛКМ) на ее обозначение в папке (выбор подтверждается инверсным выделением);
  - дважды нажав ЛКМ, получить изображение операции  $\uparrow \downarrow$  на рабочем (крайнем правом) поле программы на месте пустого прямоугольника;
  - заполнить поле параметров данной операции, выделенное знаками  $[\text{????}]: \text{SM0.0}$ . При ошибочном определении переменной или параметра редактор производит подчеркивание красной или зеленой ломаной линией (аналогично редактору WORD);
  - аналогично выбрать блок B\_1 (преобразование из формата байта в формат Integer) в папке Convert, блоки ADD\_I и MUL\_I (сложение и умножение в формате слова) в папке Integer Math и заполнить поля соответствующих параметров (рис. 7);
  - заполнить поля комментариев данной сети.
3. Записать вторую сеть Network2. Здесь (рис. 7), кроме блока сравнения  $\uparrow < \text{V} \downarrow$ , присутствуют только битовые операции. Поэтому после записи этого блока необходимо:
  - Описанным выше способом развернуть список битовых операций (папка Bit Logic).
  - Выбрать необходимые контакты из папки, причем контакт второй параллельной ветви остается «висящим» без соединений. Связи для соединений, отличающихся от простого последовательного, осуществляются с помощью клавиш панели управления  $\downarrow$ ,  $\uparrow$ ,  $\leftarrow$ ,  $\rightarrow$
  - Ввести параметры контактов в соответствии с рис. 7.
  - Заполнить поля комментариев данной сети.

4. Записать третью сеть Network3 (рис. 7).
5. Заполнить поля комментариев всей программы.
6. Сохранить проект под выбранным именем.
7. Выполнить компиляцию программы, нажав клавишу  панели управления (см. таблицу ПЗ в приложении). При необходимости исправить ошибки и скомпилировать заново.
8. Просмотреть отлаженный проект в редакторах LAD, FBD и STL. Для этого нужно выбрать соответствующий редактор (FBD или STL) в меню VIEW (Вид). При ошибочной компиляции хотя бы в одном из редакторов вместо ошибочной сети появляется надпись Invalid с номером сети красного цвета. 9. Распечатать проект в редакторе контактных планов. Рекомендации к печати:
  - убрать флаг Print in Color (Печать в цвете) для получения черно-белого текста на печати. В противном случае при черно-белой печати комментарии распечатаны не будут;
  - в окне Print Contents/Order поставить флаг только в строке Program Editor (Редактор программ);
  - в окне Range - Program Editor убрать флаг из строки АН и поставить флаг только в строке MAIN (ОВ 1);
  - войти в окно Options...(Опции), убрать все флаги и в закладке Program Editor, и в закладке Data Block. В этом случае производится печать текста программы без служебных сообщений;
  - нажав Preview... (Предварительный просмотр...), посмотреть расположение программы на листах;
  - после внесения необходимых исправлений распечатать программу в выбранном редакторе.

#### Вопросы для самопроверки

1. Каковы функции специальных маркеров?
2. Назовите области памяти микроконтроллера, к которым можно произвести адресацию в формате бита, байта, слова, двойного слова.
3. Каковы особенности адресации к аккумуляторам? к таймерам? к счетчикам? к быстрым счетчикам?
4. Что такое область отображения информации на входах? на выходах?
5. Что такое физические входы и выходы?
6. Каково назначение модулей расширения?
7. Перечислите форматы, с которыми работают блоки арифметических операций.
8. Перечислите форматы, с которыми работают операции сравнения.
9. Назовите допустимый диапазон параметров для каждого формата.
10. Назовите особенности графических редакторов FBD и LAD.
11. Назовите особенности текстового редактора STL.
12. Какие графические элементы применяются в редакторах FBD и LAD?
13. Каким образом в редакторе FBD осуществляется инвертирование сигналов?

## ТЕМА 2. Операции с таймерами и счетчиками, сегментация программ.

Задание: разработать систему управления заданным объектом на базе микроконтроллера S7-200 в двух вариантах:

- > без применения реле шагового управления;
- > с использованием сегментации программ.

### Варианты объектов управления

1. Управление дорожным движением на простом пешеходном переходе.
2. Поддержание постоянства уровня и температуры жидкости в резервуаре.
3. Грузовой лифт с возможностью останова при вызове в движении.
4. Пассажирский лифт в 9-этажном доме.
5. Климат-контроль 3-комнатной квартиры.
6. Пропускная система на предприятии.
7. Управление движением грузов в складском помещении.
8. Система жизнеобеспечения и безопасности в кинотеатре.
9. Управление фигурным фонтаном и его освещением.
10. Экономичное освещение подъезда 5-этажного жилого дома без лифта.
11. Экономичное освещение подъезда 9-этажного жилого дома с лифтом.
12. Контроль в медицинском учреждении.
13. Управление лифтами в 20-этажной гостинице.
14. Домофон.
15. Управление тепличным хозяйством.
16. Управление торговым комплексом.

### *Пример решения для варианта 1*

#### Управление дорожным движением на простом пешеходном переходе

Объектом управления являются два светофора - трехсигнальный автомобильный и двухсигнальный пешеходный. Они работают по следующему алгоритму. После включения системы с помощью переключателя «ПУСК» на 1 минуту ( $t_1 = 60$  с) включаются красный сигнал на автомобильном светофоре (КА) вместе с зеленым сигналом пешеходного светофора (ЗП). По истечении 1 минуты происходит переключение во второе состояние системы: загораются желтый сигнал на автомобильном светофоре (ЖА) и красный - на пешеходном (КП) на 30 секунд ( $t_2 = 30$  с). По окончании этого промежутка времени система переходит в следующее, третье, состояние, когда на 2 минуты ( $t_3 = 120$  с) включены зеленый сигнал на автомобильном (ЗА) и красный сигнал на пешеходном (КП) светофоре. Далее начинается новый цикл работы.

Очевидно, что для реализации данного алгоритма достаточно возможностей ПЛК S7-212, имеющего 8 входов и 6 выходов. Для включения системы в работу используется один цифровой вход. Подключение выводов микроконтроллера к системе иллюстрируется таблицей 3.

Подключение ПЛК S7-212 в системе управления дорожным движением

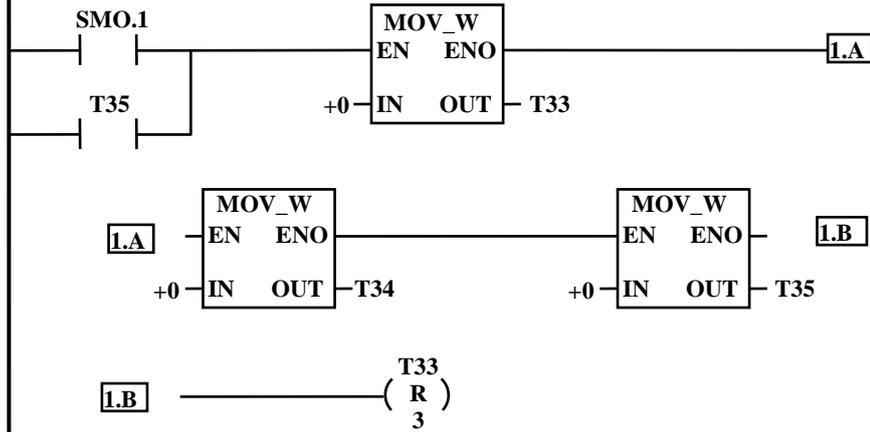
Вывод микроконтроллера	Функциональное назначение сигнала	Обозначение сигнала в программе
Q0.0	Красный сигнал автомобильного светофора	КА
Q0.1	Желтый сигнал автомобильного светофора	ЖА
Q0.2	Зеленый сигнал автомобильного светофора	ЗА
Q0.3	Красный сигнал пешеходного светофора	КП
Q0.4	Зеленый сигнал пешеходного светофора	ЗП
I0.0	Пуск в работу	Начало

Для реализации заданных выдержек времени выбраны таймеры без запоминания TON с дискретностью  $\Delta t = 10 \cdot 10^{-3}$  с: T33, T34 и T35. Их предустановленные значения (PT) рассчитываются

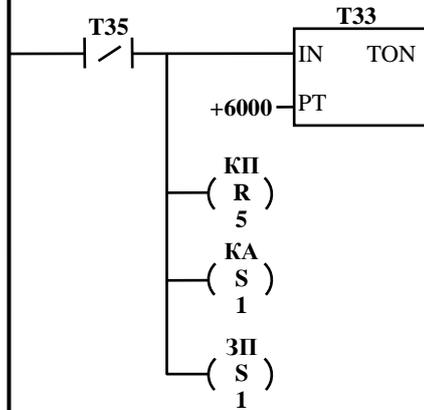
$$\text{как: } PT_{33} = \frac{t_1}{\Delta t} = \frac{60}{10 \cdot 10^{-3}} = 6000; \quad PT_{34} = \frac{t_2}{\Delta t} = \frac{30}{10 \cdot 10^{-3}} = 3000; \quad PT_{35} = \frac{t_3}{\Delta t} = \frac{120}{10 \cdot 10^{-3}} = 12000;$$

На рис. 8 и 9 представлена программа управления дорожным движением без применения реле шагового управления и с использованием сегментации программ соответственно.

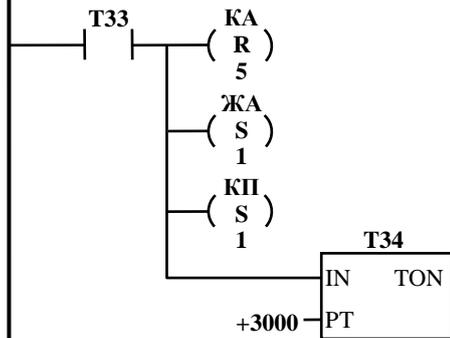
**Network 1. Сброс таймеров.**



**Network 2. Первое состояние системы (включены красный автомобильный и зеленый пешеходный сигнал) и запуск таймера на 1 минуту.**



**Network 3 Второе состояние схемы (включены желтый автомобильный и красный пешеходный сигналы) и запуск второго таймера на 30 секунд**



**Network 4 Третье состояние схемы (включены зеленый автомобильный и красный пешеходный сигналы) и запуск третьего таймера на 2 минуты**

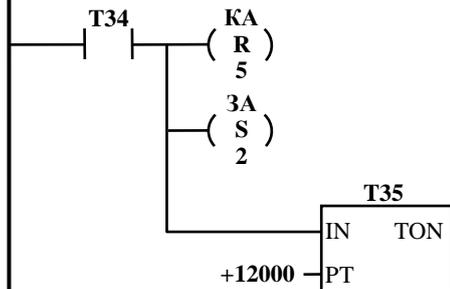
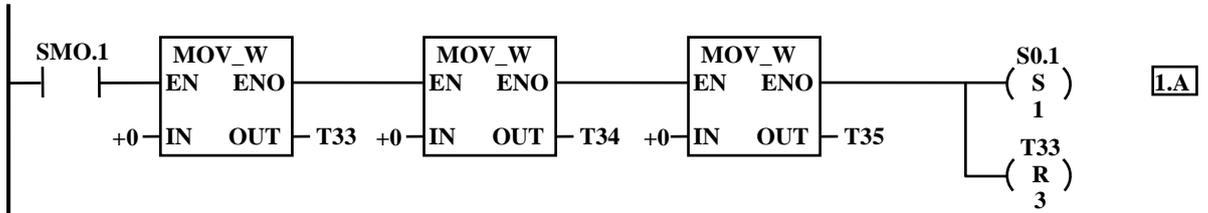
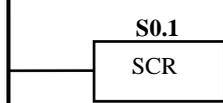


Рис. 8. Программа управления дорожным движением без применения реле шагового управления.

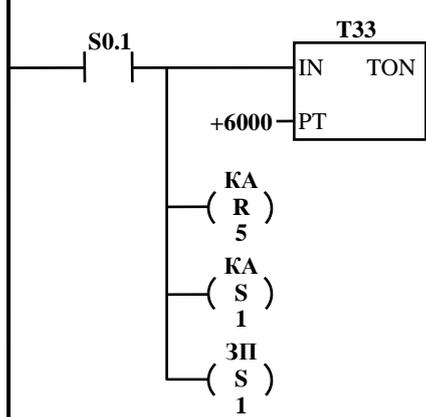
Network 1. Первоначальный сброс таймеров и установка первого шага



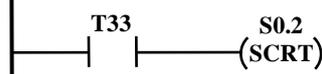
Network 2. Начало ШАГа-1



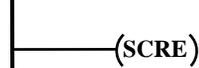
Network 3 ШАГ-1 - Первое состояние системы: включены красный автомобильный и зеленый пешеходный сигнал на 1 минуту



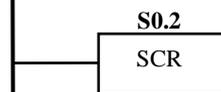
Network 4 Переход к ШАГу-2



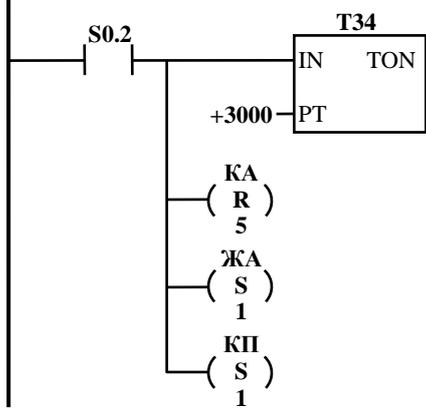
Network 5 Конец ШАГа-1



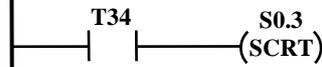
Network 6 Начало ШАГа-2



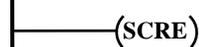
Network 7. ШАГ2 - второе состояние системы: включены желтый автомобильный и красный пешеходный сигналы на 30 секунд



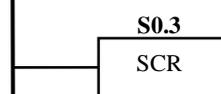
Network 8 Переход к ШАГу-3



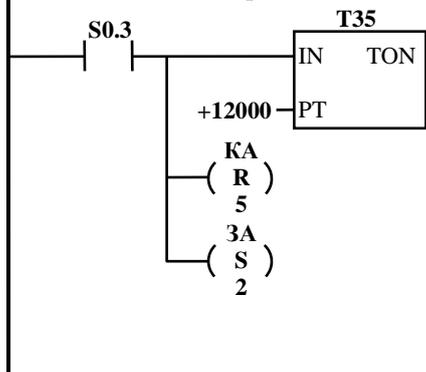
Network 9 Конец ШАГа-2



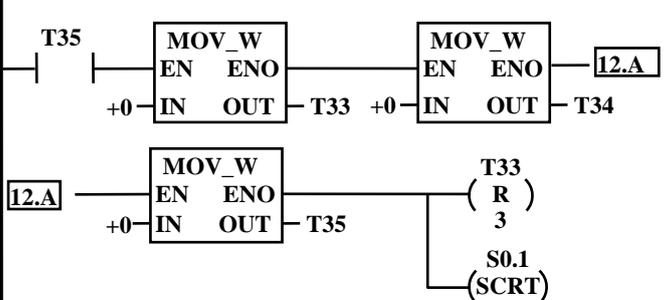
Network 10 Начало ШАГа-3



Network 11 ШАГ3 - третье состояние системы: включены зеленый автомобильный и красный пешеходный сигналы на 2 минуты



Network 12 Сброс всех таймеров



Network 13 Конец ШАГа-3

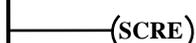


Рис. 9. Программа управления дорожным движением с сегментацией.

## Вопросы для самопроверки

1. Сравните временные диаграммы различных видов таймеров.
2. Сравните временные диаграммы различных видов счетчиков.
3. Перечислите операции, обеспечивающие работу реле шагового управления.
4. Как обеспечить разветвление основной программы на 4 ветви? Приведите алгоритм и типовое программное решение.
5. Как обеспечить разделение подпрограммы на 4 шага? Приведите алгоритм и типовое программное решение.
6. Как с помощью операции перехода и меток «обойти» блок программы? Как аналогичную задачу решить с использованием реле шагового управления? Приведите алгоритм и типовое программное решение.
7. Как с помощью операции перехода и меток обеспечить разделение программы на 3 шага, выполняемых по одному и тому же условию? Как аналогичную задачу решить с использованием реле шагового управления? Приведите алгоритм и типовое программное решение для каждого способа решения. Сделайте их сравнительный анализ.

### ТЕМА 3. Быстрые счетчики, табличные операции, часы.

Задание по темам 3, 4 и 5: написать программу в соответствии с заданным вариантом; скомпилировать программу в редакторах LAD и FBD; документировать скомпилированную программу в редакторе контактных планов.

#### Варианты задания

1. Снятие показаний скорости посредством быстрого счетчика и фиксация скорости, измеренной в об/мин и в рад/с, в таблицах с начальными адресами VW200 и VW400 соответственно. Фиксация скорости производится с интервалом 2 секунды. Цифровой датчик скорости (ЦДС) выдает 500 импульсов на один оборот вала двигателя, максимальная скорость двигателя 800 об/мин.

2. Измерение скорости через каждые 10 секунд. ЦДС выдает 400 импульсов на один оборот вала двигателя.

3. Постоянный контроль скорости двигателя: если скорость ниже величины  $W_0 = 500$  об/мин, то подается сигнал на выход Q0.0 ПЛК, если скорость выше этой величины, то - на выход Q0.1, если равна, то - на выход Q0.2. Фиксация скорости производится с интервалом 2 секунды. ЦДС выдает 400 импульсов на один оборот вала двигателя.

4. Постоянный контроль перемещения рабочего органа (РО) станка с программным управлением вдоль контрольной линейки длиной 1 метр. Через каждые 5 см линейки установлены датчики, переключающиеся в 1 при прохождении РО в одну или другую сторону. В памяти микроконтроллера в области VDO должна сохраняться информация о расстоянии, на котором находится РО от левого края линейки.

5. Сохранение статистической информации о скорости двигателя в течение 1 часа в таблице максимальной длины с начальным адресом VW10. Подсчет скорости организован на быстром счетчике. ЦДС выдает 400 импульсов на один оборот вала двигателя. Фиксация скорости производится с интервалом 2,5 секунды. Минимальная скорость должна быть зафиксирована в области VW8.

6. Фиксация значений скорости, измеренной с помощью ЦДС, выдающего 500 импульсов на один оборот вала двигателя, в области VW0. При нажатии кнопки «Таблица» данные сохраняются в таблице на 50 строк с начальным адресом VW20. После заполнения таблицы ввод в неё новых данных блокируется.

7. Фиксация значений скорости, измеренной с помощью ЦДС, выдающего 600 импульсов на один оборот вала двигателя, в таблице на 60 строк с начальным адресом VW4. После

заполнения таблицы производится поиск всех значений скорости, превышающих 500 об/мин, и регистрация их в таблице максимальной длины с начальным адресом VW300.

8. Пропускная система на предприятии на базе быстрого счетчика. Количество присутствующих должно каждый час заноситься в таблицу с начальным адресом VW100. В начале рабочего дня вчерашняя таблица очищается. В обеденный перерыв данные в таблицу не вносятся.

9. Пропускная система на предприятии на базе быстрого счетчика. Количество прибывших и вышедших с территории работников должно каждый час фиксироваться в таблицах с начальными адресами VW10 и VW100. Перед началом рабочего дня вчерашние таблицы очищаются.

10. При нажатии кнопки «ВПЕРЕД» быстрый счетчик начинает счет вперед со значения 1000. Если его текущее значение превышает 2000, устанавливаются выходы ПЛК Q0.0=1 и Q0.1=1. При нажатии кнопки «НАЗАД» быстрый счетчик начинает счет назад с полученного текущего значения. Если кнопка «НАЗАД» нажата первой, то отсчет начинается со значения 1000.

11. В 8<sup>00</sup> быстрый счетчик начинает подсчет импульсов с нулевого значения. После каждого нажатия на кнопку, соединенную с входом I0.0, текущее значение счетчика фиксируется в таблице максимальной длины с начальным адресом VW100. При нажатии на кнопку, соединенную с входом I0.1, происходит обнуление и блокировка счетчика. В 17<sup>00</sup> счетчик блокируется без обнуления.

12. После нажатия на кнопку, соединенную с входом I0.0, быстрый счетчик начинает подсчет импульсов. Каждые 50 секунд текущее значение счетчика фиксируется в таблице максимальной длины с начальным адресом VW100. При нажатии на кнопку, соединенную со входом I0.1, происходит обнуление и блокировка счетчика.

13. Фиксация в рабочее время (с 8<sup>00</sup> до 17<sup>00</sup>) значений скорости, измеренной с помощью ЦДС, выдающего 600 импульсов на один оборот вала двигателя, в таблице на 60 строк с начальным адресом VW4. После заполнения таблицы производится поиск всех значений скорости, меньших 750 об/мин, и регистрация их в таблице на 75 записей с начальным адресом VW300. В начале рабочего дня вчерашняя таблица очищается.

14. Контроль скорости двигателя постоянного тока: если она выше 500 об/минуту, выход ПЛК Q0.0=1 сдвигает ползунок реостата в цепи якоря до тех пор, пока скорость не вошла в допустимые значения. ЦДС выдает 750 импульсов на один оборот вала двигателя. Фиксация скорости осуществляется каждую минуту,

15. После нажатия на кнопку, соединенную с входом I0.0, быстрый счетчик начинает подсчет импульсов. Каждые 30 секунд текущее значение счетчика фиксируется в таблице максимальной длины с начальным адресом VW180. После заполнения таблицы при нажатии на кнопку, соединенную с входом I0.3, минимальное значение из таблицы переписывается в область VW0.

### ***Пример решения для варианта 1***

Простейший цифровой датчик скорости (ЦДС) с разрешением 500 импульсов на 1 оборот вала представляет собой установленный на валу двигателя диск с 500 прорезями, а также источник света и фотодатчик по разные стороны диска. Разрешающая способность датчика  $\Delta_{\text{ЦДС}}=500$  имп/об. Максимальная скорость двигателя  $w_{\text{max}}=800$  об/мин =13,3 Гц. Тогда максимальная скорость подачи импульсов с ЦДС равна:

$$f_{\text{max}} = w_{\text{max}} \cdot \Delta_{\text{ЦДС}} = 13,3 \text{ об/с} \cdot 500 \text{ имп/оборот} = 6650 \text{ имп/с} = 6,65 \text{ кГц.}$$

С такой максимальной частотой работают быстрые счетчики всех CPU, кроме CPU 212. Программа составлена для CPU 224. Для решения поставленной задачи необходимо подсчитать количество импульсов с ЦДС за 2 секунды. Отсчет этого промежутка времени можно организовать с помощью управляемого временем прерывания на таймере T32 (событие прерывания 21). Программа обработки прерывания (ПОП) с именем INTO будет вызываться через каждые 2 секунды.

Она обеспечивает:

- блокировку счетчика HSC1 с фиксацией его текущего значения в памяти микроконтроллера и сохранение полученного количества импульсов  $N$  в области переменных VD0;
- подсчет скорости по формуле  $n \left[ \frac{\text{об}}{\text{мин}} \right] = \frac{N \cdot 60}{\Delta_{\text{цс}} \cdot 2} = \frac{N \cdot 30}{\Delta_{\text{цс}}}$  с сохранением результата в области переменных VD4;
- расчет скорости  $\omega \left[ \frac{\text{рад}}{\text{с}} \right]$  по формуле  $\omega = \frac{\pi n}{30}$  с сохранением результата в области переменных VD8.

Алгоритм решения задачи приведен на рис. 10, а соответствующая этому алгоритму программа на языке контактных планов - на рис. 11.

В первой сети основной программы в первом машинном цикле (МЦ1) работы микроконтроллера осуществляется загрузка первоначального значения управляющего байта 98H. В ПОП управляющий байт в сети 1 принимает значение 00H для блокировки счетчика, в сети 2 - 80H, поскольку служит для перезапуска счетчика с теми же параметрами (активен только старший бит управления). Использование специального маркера SM0.1, установленного в 1 только в МЦ1, связано с особенностью блока HDEF: он может быть применен к каждому быстрому счетчику только один раз. Кроме того, здесь производится организация перехода на ПОП каждые 2 секунды после запуска быстрого счетчика HSC1: с помощью блока ATCH событию прерывания 21 (отсчет интервала времени таймером T32) назначается программа обработки прерывания INT\_0.

#### Вопросы для самопроверки

1. Сравните принцип работы обычных и быстрых счетчиков.
2. Напишите алгоритм изменения направления счета для режимов 0-2.
3. Напишите алгоритм изменения предварительно установленного значения для режимов 0-11.
4. Напишите алгоритм загрузки нового текущего значения.
5. Скольким событиям может соответствовать одна ПОП?
6. Скольким ПОП может соответствовать одно событие прерывания?
7. Напишите алгоритм обнуления переполненной таблицы.
8. Напишите алгоритм последовательного заполнения двух таблиц максимальной длины, следующих в области переменных без свободных промежутков, начиная с адреса VW500.
9. Рассчитайте адрес последнего байта таблицы на 70 записей с начальным адресом VW220.
10. Напишите программу поиска минимального числа в таблице на 50 записей с начальным адресом VW10 (таблица заполнена полностью).
11. Напишите программу поиска максимального числа в таблице на 50 записей с начальным адресом VW10 (таблица заполнена полностью).

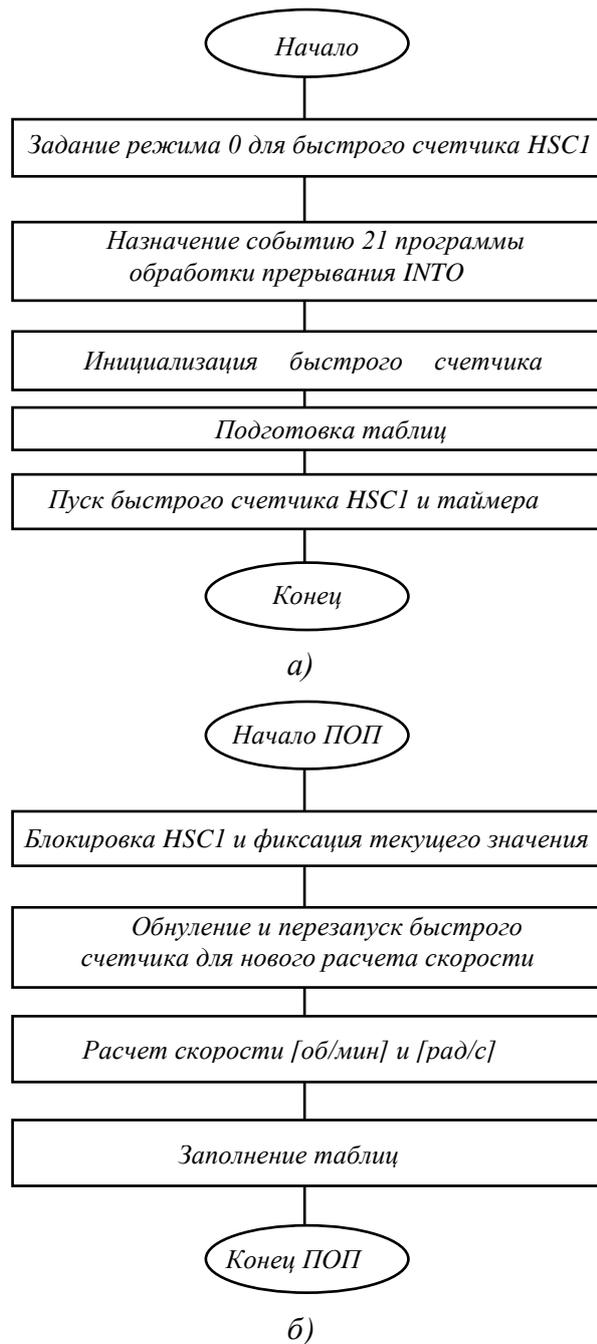
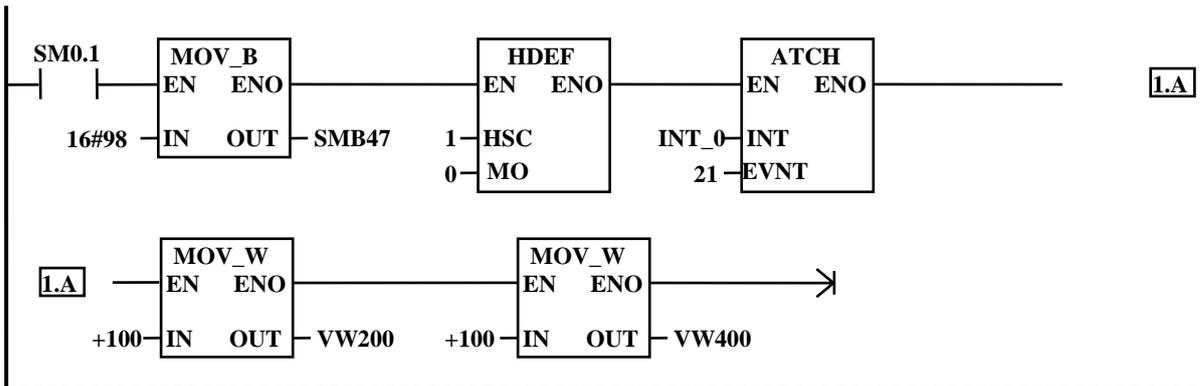
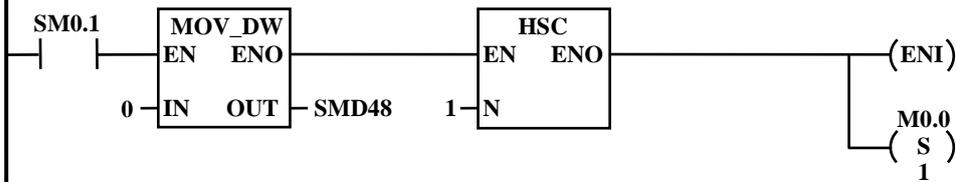


Рис. 10 . Алгоритм для варианта 1 задания по теме 3:  
 а - основная программа; б - программа обработки прерывания

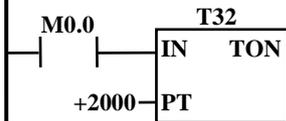
Network 1. Однократное задание в МЦ1 режима 0 для HSC1.



Network 2 Первоначальный запуск счетчика и разрешение обработки прерывания

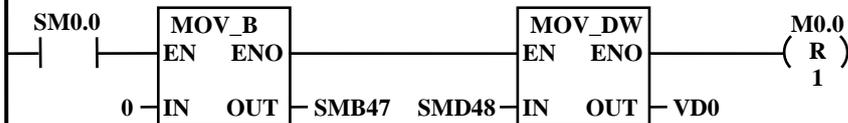


Network 3. Запуск таймера для организации прерывания, управляемого временем через таймер (событие 21).

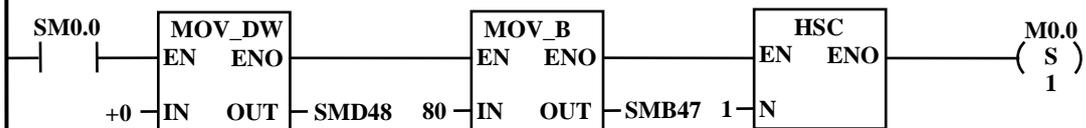


a)

Network 1. Фиксация текущего значения счетчика, обнуление разрешающего маркера для таймера для его сброса.



Network 2. Обнуление и перезапуск быстрого счетчика и таймера для следующего расчета скорости.



Network 3 Текущий расчет скорости.

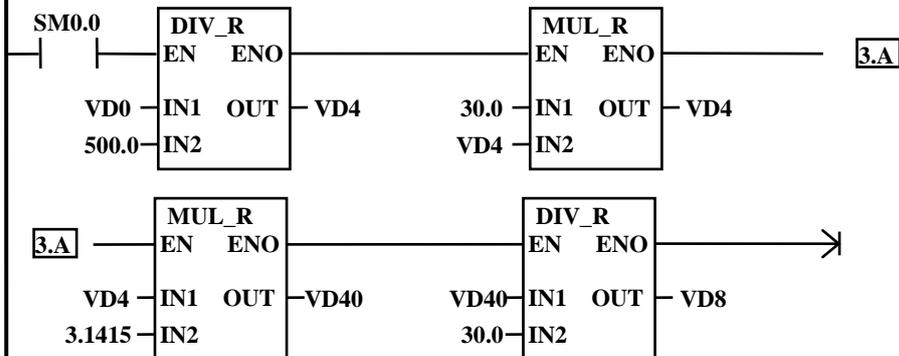


Рис. 11. Программа снятия показаний скорости посредством быстрого счетчика (вар. 1).

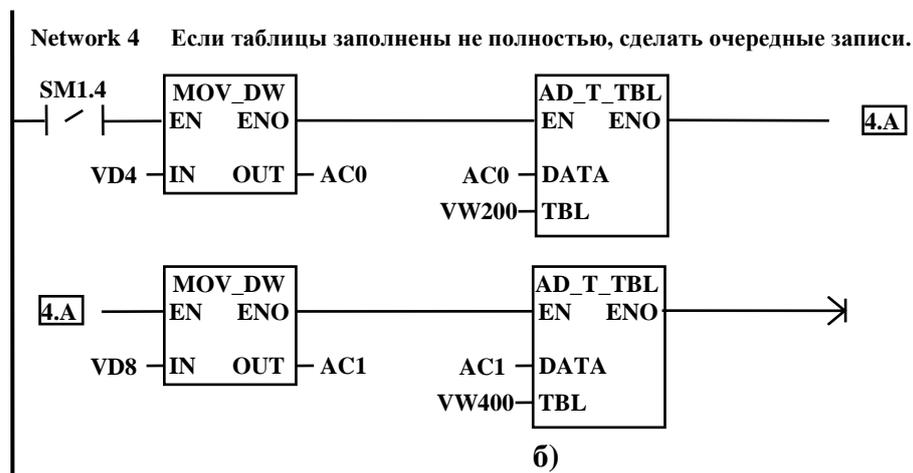


Рис. 11. Окончание.

## ТЕМА 4. Импульсные выходы

### Варианты задания

1. Организовать непрерывную выдачу импульсов на выходе Q0.0 с относительной длительностью включения 50%. Первые 10 импульсов период следования составляет 1 с. Затем он возрастает в 2 раза в каждой новой последовательности из 10 импульсов.
2. При нажатии на кнопку «Пуск» начинают работать три двигателя: шаговые ШД1 и ШД2, совершающие 1 оборот при подаче 500 импульсов на их ОВ и аналоговый двигатель ДЗ, вращающийся с постоянной скоростью при замыкании цепи питания. ШД1 должен сделать 4 оборота за 1 минуту, ШД2 - 7 оборотов за 3 минуты. По окончании движения Т11Д1 и ШД2 двигатель ДЗ нужно отключить от сети. Окончание цикла работы наступает после самовыбега ДЗ через 1 минуту после его отключения.
3. Организовать непрерывную выдачу импульсов на выходе Q0.1 с относительной длительностью включения 30%. Первые 10 минут после включения период следования составляет 0,5 с. Затем каждые 7 минут он удваивается.
4. В течение 1 часа после начала работы ПЛК с помощью многосегментной конвейерной обработки РТО организовать непрерывную выдачу импульсов на выходе Q0.0. Первые 20 минут после включения период следования составляет 0,5 с. Затем он возрастает в 2 раза через каждые 5 минут.
5. В рабочее время ( $8^{00} - 17^{00}$ ) в начале каждого астрономического часа реализовать поворот ШД: в первый раз - на 3 оборота за 1 минуту, во второй - на 4 оборота за 1 минуту и т.д. Один оборот ШД совершает при подаче на его ОВ 1000 импульсов.
6. Если среднее значение полностью заполненной таблицы из 10 записей с начальным адресом VW0 больше 45, то ШД должен повернуться на 4 оборота за 3 минуты. Если это среднее значение меньше 45, то - на 5 оборотов за 2 минуты. Если же оно равно 45, то - на 7 оборотов за 8 минут. Один оборот ШД совершает при подаче на его ОВ 500 импульсов.
7. Если нажата кнопка «1», происходит пуск первого ШД, а затем через 1 минуту - пуск второго ШД. Если нажата кнопка «2», то наоборот, сначала происходит пуск второго ШД, а затем через 3 минуты - пуск первого ШД. Оба ШД делают 3 оборота за 1 минуту. Один оборот ШД совершает при подаче на его ОВ 700 импульсов.
8. При нажатии на кнопку «ПУСК» ШД1 совершает 3 оборота за 4 минуты, а ШД2 - 3 оборота за 5 минут. Через 30 секунд после окончания вращения обоих двигателей найти максимальное значение в заполненной таблице из 15 записей с начальным адресом VW0. Один оборот ШД совершает при подаче на его ОВ 500 импульсов.

9. С 10<sup>00</sup> до 15<sup>00</sup> в рабочие дни каждые 3 минуты происходит поворот сначала ШД1 (350 импульсов на ОВ для реализации 1 оборота), затем после его останова - поворот Т1ТД2 (350 импульсов на ОВ для реализации 1 оборота). В первый раз оба двигателя делают 1 оборот, во второй раз - 2 и т.д. Количество оборотов за пятидневную рабочую неделю ежедневно фиксируется в таблице с начальным адресом VW0. В начале недели таблица очищается и начинается ее повторное заполнение.

10. При нажатии на кнопку «ПУСК» каждые 5 минут происходит поочередный запуск ШД1 и ШД2. В первый раз ШД1 совершает 1 оборот за 1 минуту, а ШД2 всегда вращается в два раза быстрее; второй раз - ШД1 делает 1 оборот за 70 секунд, третий раз - 1 оборот за 80 секунд и т.д. до нажатия на кнопку «Стоп». Один оборот ШД1 совершает при подаче на его ОВ 100 импульсов, ШД2 - при подаче 400 импульсов.

11. При нажатии на кнопку «ПУСК» каждые 5 минут происходит запуск ШД. В первый раз он совершает 1 оборот за 1 минуту, второй раз - 1 оборот за 70 секунд, третий раз - 1 оборот за 80 секунд и т.д. до нажатия на кнопку «Стоп». Один оборот ШД совершает при подаче на его ОВ 100 импульсов.

12. Через 10 секунд после нажатия на кнопку «ПУСК» ШД совершает сначала 3 оборота за 50 секунд. Затем через 1 минуту после останова - еще 3 оборота за 50 секунд и т.д. до нажатия на кнопку «+» или «-». Если нажата кнопка «+», время поворота на 3 оборота увеличивается на 5 секунд, если нажата кнопка «-», то оно уменьшается на 5 секунд. Один оборот ШД совершает при подаче на его ОВ 200 импульсов.

13. При нажатии на кнопку «ПУСК» ШД совершает N циклов работы. N - среднее арифметическое содержимого заполненной таблицы на 8 записей с начальным адресом VW500, округленное в сторону увеличения. Первый цикл: сначала 1 оборот за 30 секунд, затем пауза 30 секунд. 2 цикл: 2 оборота за 30 секунд, затем пауза 30 секунд и т.д. Один оборот ШД совершает при подаче на его ОВ 300 импульсов.

14. На выходе Q0.0 организовать ШИМ с относительной длительностью включения, пропорциональной коду в области VD0. Коррекция длительности включения производится по переднему фронту внешнего управляющего сигнала, подаваемого на вход 10.0. Статистика изменения длительности включения сохраняется в течение суток в таблице с начальным адресом VW600.

15. ШИМ на выходе Q0.1 с периодом следования импульсов 0,5 секунд с относительной длительностью включения, определяемой внешними сигналами.

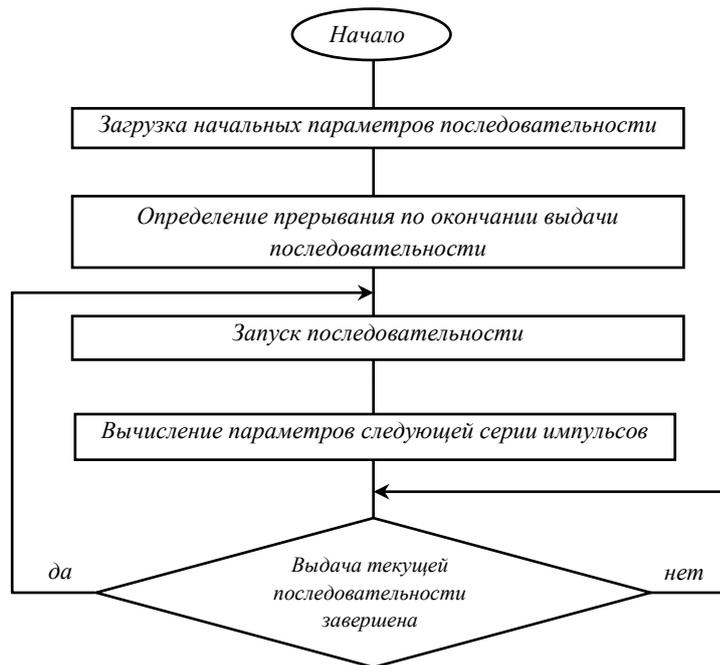
При нажатии на кнопку «1» она задается равной 60%, при нажатии на «2» - 70%, при нажатии на «3» - 80%. Статистика изменения длительности включения сохраняется в течение суток в таблице с начальным адресом VW600. 16. ШИМ на выходе Q0.1 с относительной длительностью включения 75% и начальным периодом в 1 секунду. Период следования импульсов увеличивается вдвое при каждом срабатывании датчика (отрицательный фронт на входе 10.1). Статистика изменения периода следования импульсов сохраняется в течение суток в таблице с начальным адресом VW600.

### ***Пример решения для варианта 1***

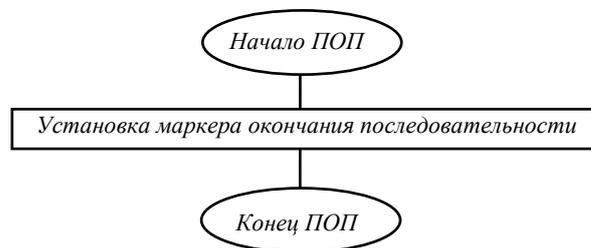
Поскольку время выдачи импульсов по условию задачи не ограничено, количество сегментов неизвестно, поэтому задача решается посредством односегментной обработки конвейера в соответствии с алгоритмом, представленным на рис. 12. Очевидно, для второй последовательности параметры можно рассчитать следующим образом:  $T_2=2T_1$ . В соответствии с правилом односегментной обработки конвейера параметры следующей последовательности должны быть загружены в область специальных маркеров после запуска текущей последовательности. На рис. 13 приведена соответствующая этому алгоритму программа на языке контактных планов.

## Вопросы для самопроверки

1. Что такое конвейер РТО?
2. Что такое переполнение конвейера РТО?
3. Что такое относительная длительность включения?
4. Каково состояние быстрого выхода при 100%-ной относительной длительности включения?
5. Напишите алгоритм изменения параметров ШИМ при работающем быстром выходе.
6. Напишите алгоритм изменения периода следования импульсов для работающего ШИМ.
7. Напишите алгоритм изменения периода следования импульсов для функции РТО, для работающего быстрого выхода.
8. Напишите алгоритм изменения периода следования импульсов для функции РТО, для неработающего быстрого выхода.
9. Напишите алгоритм блокировки быстрого выхода.



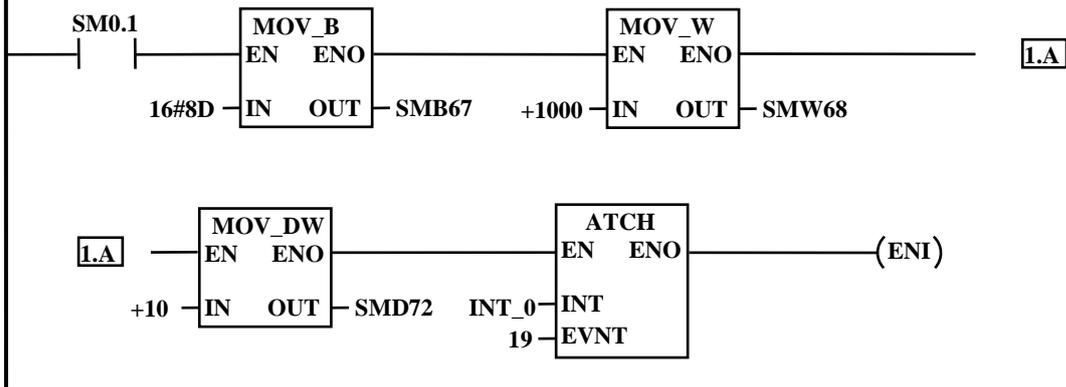
а)



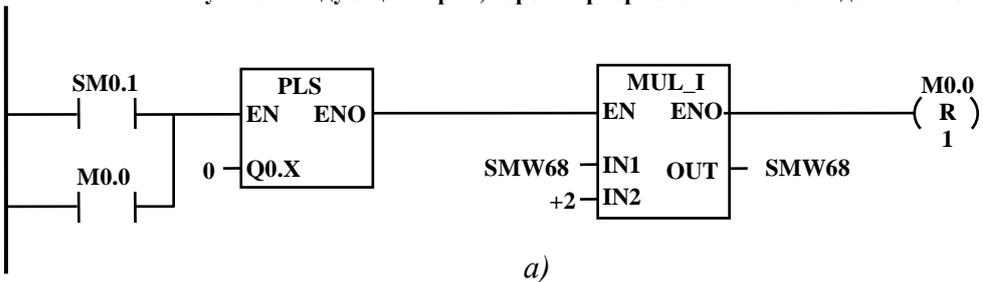
б)

Рис. 12 . Алгоритм односегментной обработки РТО:  
а - основная программа;  
б - программа обработки прерывания по окончании последовательности

**Network 1** Задание в МЦП первоначальных параметров последовательности импульсов и задание фиксации конца последовательности маркером M0.0 в INT\_0



**Network 2** Запуск подготовленной последовательности и расчет параметров импульсов следующей серии, сброс маркера окончания последовательности



**Network 1** Установка маркера конца текущей последовательности импульсов

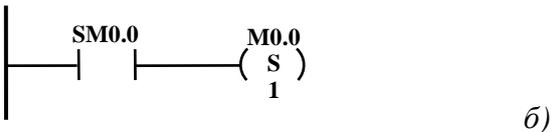


Рис. 13. Программа односегментной обработки ПТО:  
а - основная программа; б - программа обработки прерывания

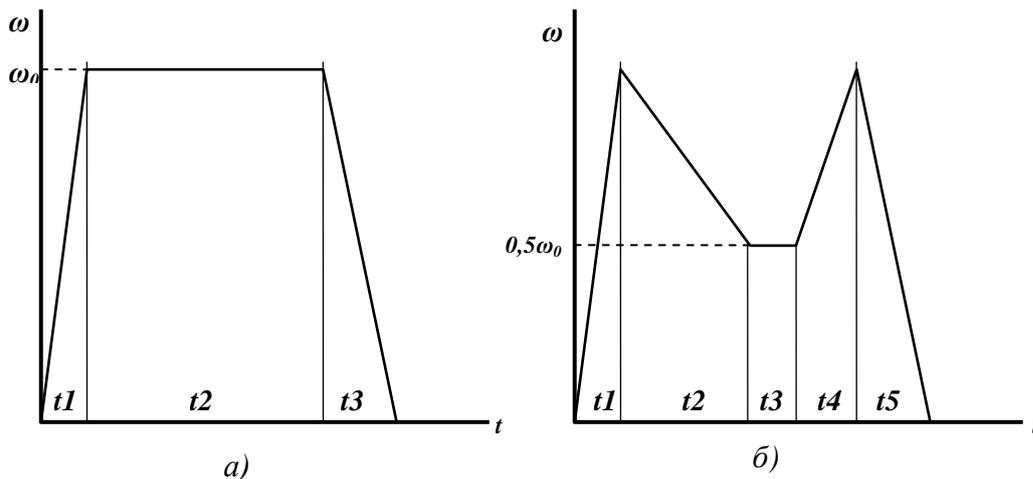


Рис. 14. Тахограммы к заданию по теме 4:  
а - с монотонным изменением скорости; б - с пилообразным изменением скорости

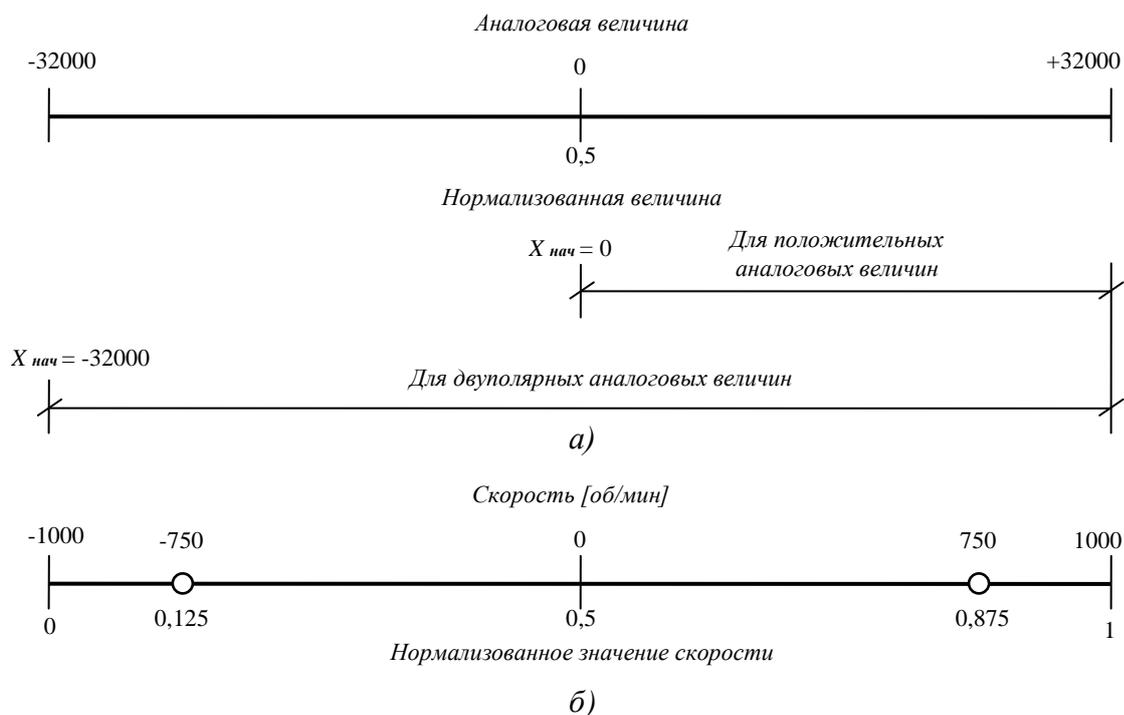


Рис. 15. Нормализация и масштабирование:  
а - в общем виде; б - для примера решения варианта 1 задания по теме 5

## ТЕМА 5. PID-операция

### Варианты задания

1. PID-регулятор должен осуществлять поддержание постоянства скорости движения тележки на уровне 750 об/мин. По ходу движения на тележку кладется или убирается груз. При этом скорость двигателя находится в диапазоне от 0 до  $\pm 1000$  об/мин. При срабатывании конечных выключателей на I0.0 и I0.1 задание на скорость меняет знак и тележка движется в противоположную сторону.

2. ПИ-регулирование скорости двигателя по линейному закону.

3. PID-регулирование скорости двигателя в соответствии с тахограммой, представленной на рис. 14,а, где  $\omega_0 = 700$  об/мин;  $t_1 = 5$  мин;  $t_2 = 1$  час;  $t_3 = 7$  минут.

4. ПИ-регулирование скорости двигателя в соответствии с тахограммой, представленной на рис. 14,а, где  $\omega_0 = 800$  об/мин;  $t_1 = 7$  мин;  $t_2$  определяется датчиком, соединенным с входом I0.0;  $t_3 = 9$  минут.

5. PID-регулирование скорости двигателя в соответствии с тахограммой на рис. 14,б, где  $\omega_0 = 1500$  об/мин;  $t_1 = t_5 = 40$  сек;  $t_2 = t_4 = 5$  минут;  $t_3 = 1,5$  часа.

6. ПД-регулирование скорости двигателя в соответствии с тахограммой, представленной на рис. 14,б, где  $\omega_0 = 1200$  об/мин;  $t_1 = t_5 = 30$  сек;  $t_2 = t_4 = 6$  минут;  $t_3$  определяется датчиком, соединенным с входом I0.2;  $t_3 = 9$  минут.

7. ПД-регулирование скорости двигателя в соответствии с тахограммой, представленной на рис. 14,а, где  $\omega_0 = 750$  об/мин;  $t_1 = 1,5$  мин;  $t_2 = 1$  час;  $t_3 = 4$  минуты. Обработка тахограммы начинается ежедневно в 10<sup>00</sup>.

8. PID-регулирование скорости двигателя в соответствии с тахограммой на рис. 14,а, где  $\omega_0 = 880$  об/мин;  $t_1 = 7,5$  мин;  $t_2$  определяется датчиком, соединенным с входом I0.1;  $t_3 = 5,5$  минут.

9. PID-регулирование скорости двигателя в соответствии с тахограммой, представленной на рис. 14,б, где  $\omega_0 = 1100$  об/мин;  $t_1 = t_5 = 20$  сек;  $t_2 = t_4 = 3$  минуты;  $t_3 = 2,5$  часа. Обработка тахограммы начинается в рабочие дни в 11<sup>45</sup>.

10. ПИ-регулирование скорости двигателя в соответствии с тахограммой, представленной на рис. 14,б, где  $\omega_0=1350$  об/мин;  $t_1=t_5=20$  сек;  $t_2=t_4=3$  минуты;  $t_2$  определяется датчиком, соединенным с входом Ю.3;  $t_3=7$  минут. Обработка тахограммы начинается через день в 8<sup>50</sup>.

11. ПИ-регулирование скорости двигателя в соответствии с тахограммой на рис. 14,а, где  $\omega_0=700$  об/мин;  $t_1=5$  мин;  $t_2=1$  час;  $t_3=7$  мин. Отклонения скорости от задания фиксируются в таблице с начальным адресом VW0.

12. ПИ-регулирование скорости двигателя в соответствии с тахограммой на рис. 14,а, где  $\omega_0=820$  об/мин;  $t_1=1$  мин;  $t_2$  определяется датчиком на Ю.0;  $t_3=3$  мин.

13. ПИД-регулирование скорости двигателя в соответствии с тахограммой на рис. 14,б, где  $\omega_0=1150$  об/мин;  $t_1=t_5=35$  сек;  $t_2=t_4=25$  минут;  $t_3=1,5$  часа. Отклонения скорости от задания фиксируются в таблице с начальным адресом VW200.

14. ПИ-регулирование скорости двигателя в соответствии с тахограммой, представленной на рис. 14,б. Здесь  $\omega_0=300$  об/мин;  $t_1=t_5=20$ с;  $t_2=t_4=2$  минуты;  $t_2$  определяется датчиком, соединенным с входом Ю.2;  $t_3=6$  минут. Отклонения скорости фиксируются в таблице с начальным адресом VW500.

15. ПИД-регулятор должен обеспечить регулирование в соответствии с вращением эталонного двигателя. Отклонения регулируемой величины от задания фиксируются в таблице с начальным адресом VW0.

### *Пример решения для варианта 1*

В первом приближении выбраны параметры ПИД-регулятора:  $k_c=0,5$ ;  $T_s=0,1$  с;  $T_i=30$  мин;  $T_D=0,01$  с. По условию предусмотрен реверс, поэтому скорость изменяется от -1000 до +1000 об/мин. Для контроля скорости и формирования аналогового сигнала задания на скорость используется модуль расширения ЕМ 235 АИ4/АО 1x12 бит. Для данной задачи используются области АИW0 и АQW0. Конфигурация модуля [2, стр. А43] должна обеспечивать диапазон входного напряжения  $\pm 10$  В. Нормализация входных значений контура осуществляется в соответствии со шкалой нормализации, показанной на рис. 15,а. В соответствии с этой шкалой нормализованное значение любой величины рассчитывается как

$$X_{НОРМ} = \frac{X_{АНАЛОГ}}{\Delta X} + X_{НАЧ}; \quad (1)$$

где  $X_{аналог}$  - диапазон изменения аналоговой величины;  $X_{нач}$  - начальное значение нормализованной величины;  $\Delta X$  - разность между максимальным и минимальным значением аналоговой величины. Для величин, меняющих знак,  $X_{аналог}=-32000 \dots 32000$ ;  $X_{нач}=0$ ;  $\Delta X=64000$ . Для реализации целей управления код аналоговой величины рассчитывается из нормализованного значения по формуле, следующей из (1):

$$X_{АНАЛОГ} = (X_{НОРМ} - X_{НАЧ}) \cdot \Delta X; \quad (2)$$

В соответствии с рис. 15,б для нормализации и масштабирования нужно ввести коэффициенты:  $X_{АНАЛОГ}=-32000 \dots 32000$ ;  $X_{нач}=0$ ;  $\Delta X=64000$ . Алгоритм решения задачи приведен на рис. 16, а соответствующая ему программа в редакторе контактных планов - на рис. 17.

В первой сети основной программы (рис. 17,а) устанавливается задание на скорость, соответствующее +750 об/мин, и задаются все постоянные времени ПИД-регулятора. Во второй сети определяется прерывание 0, управляемое временем (событие 10): ПОП INT\_0 будет выполняться каждые 0,1 секунды в соответствии с постоянной времени  $T_s$ . Данная ПОП (рис. 17,б) осуществляет фиксацию скорости в аккумуляторе АСО, нормализацию его значения в соответствии с формулой (1), расчет параметров PID-контура, масштабирование рассчитанного в PID-операции задания на скорость и выдачу его на аналоговые выходы АQW0. При срабатывании конечных выключателей на входе ПЛК 10.0 и 10.1 (события прерывания 0 и 1), будут вызываться ПОП INT\_1 и INT\_2 (рис. 17,в, г). При их выполнении определяются задания на скорость, соответствующие +750 об/мин. и (-750 об/мин).

## Вопросы для самопроверки

1. Объясните принципы нормализации параметров PID-операции для униполярных величин.
2. Объясните принципы масштабирования параметров PID-операции для униполярных величин.
3. Объясните принципы нормализации параметров PID-операции для биполярных величин.
4. Объясните принципы масштабирования параметров PID-операции для биполярных величин.
5. Назовите порядок постоянных времени составляющих регулятора.
6. Как обеспечивается плавность регулирования в начале работы.
7. Сколько контуров регулирования можно реализовать в одной программе?
8. Назовите диапазон параметра TBL для CPU 221.
9. Напишите алгоритм программирования ПИ-регулятора и соответствующую программу на языке контактных планов.
10. Напишите алгоритм программирования ПД-регулятора и соответствующую программу на языке контактных планов.

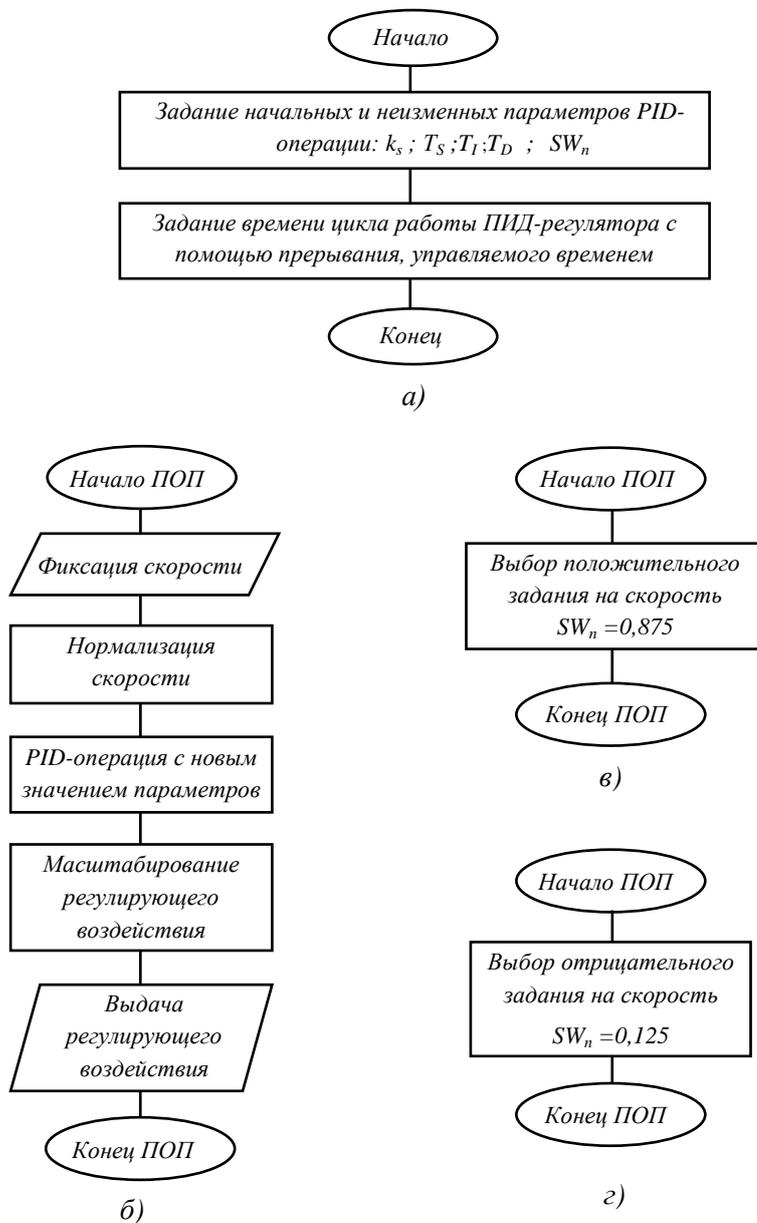
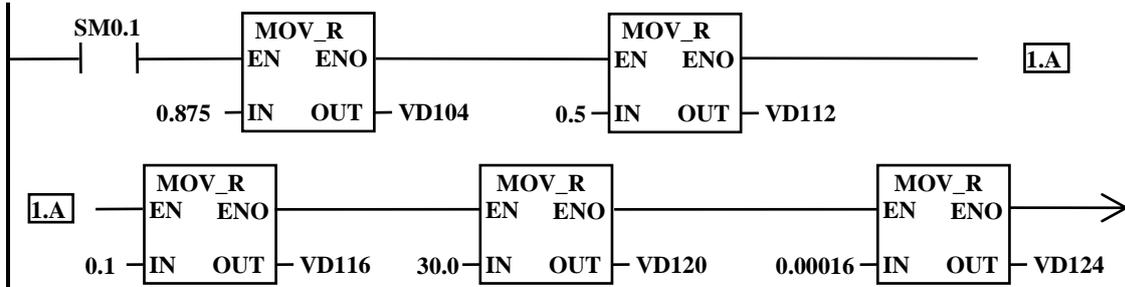
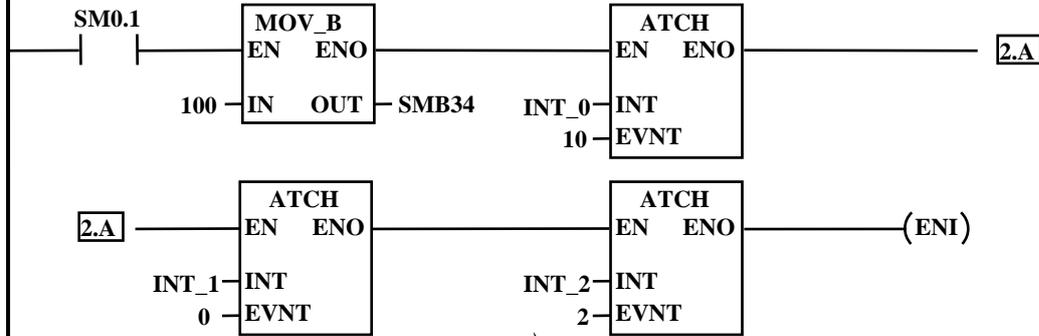


Рис. 16 . Алгоритм настройки ПИД-регулятора:  
а - основная программа; б - ПОП INT\_0; в - ПОП INT\_1; г - ПОП INT\_2

**Network 1 Установка начальных и неизменных параметров регулятора**

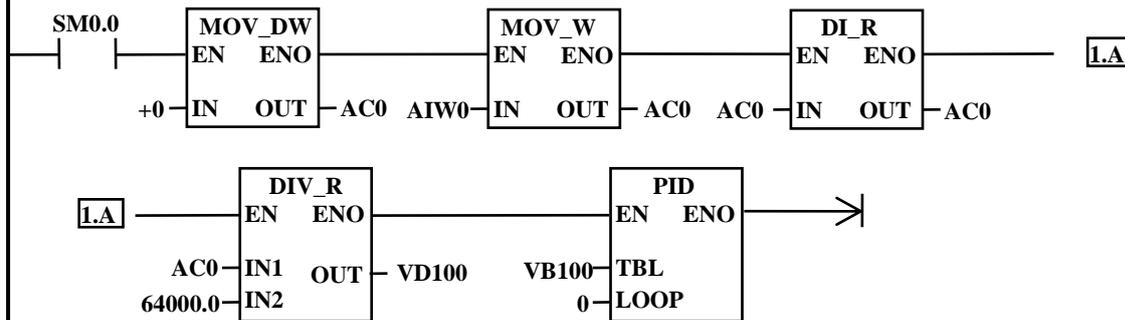


**Network 2 Установка прерываний**



a)

**Network 1 Фиксация текущего значения скорости, нормализация его значения и запуск ПИД-вычислений**

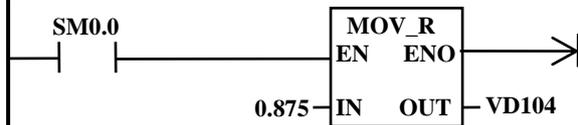


**Network 2 Масштабирование расчетного значения Мп и передача его на аналоговые выходы**



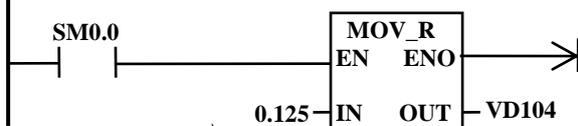
b)

**Network 1 Смена заданного значения скорости**



c)

**Network 1 Смена заданного значения скорости**



d)

Рис. 17. Программа настройки ПИД-регулятора:  
а - основная программа; б - ПОП INT\_0; в - ПОП INT\_1; г - ПОП INT\_2

### Основные характеристики CPU S7-200 и S7-220

CPU 212	CPU 214	CPU 215	CPU 216	CPU 221	CPU 222	CPU 224	CPU 226
1. Физический размер, мм							
160x80x62	197x80x62	218x80x62	218x80x62	90x80x62	90x80x62	120,5x80x62	190x80x62
2. Память программ							
512 слов	2Кслов	4К слов	4К слов	2К слов	2К слов	4К слов	4К слов
3. Данные пользователя							
512 слов	2К слов	2,5К слов	2,5К слов	ПС слов	1К слов	2,5К слов	2,5К слов
4. Внутренние маркеры							
128	256	256	256	256	256	256	256
5. Встроенные цифровые входы/выходы							
8DI/6DO	14DI/10DO	14DI/10DO	14DI/10DO	6DI/4DO	8DI/6DO	14DI/10DO	24DI/16DO
6. Максимальное количество модулей расширения							
2	7	7	7	0	2	7	7
7. Отображение процесса ввода/вывода							
64DI/64DO	64DI/64DO	64DI/64DO	64DI/64DO	128DI/128DO	128DI/128DO	128DI/128DO	128DI/128DO
8. Аналоговые входы/выходы (расширение)							
16AI/16AQ	16AI/16AQ	16AI/16AQ	16AI/16AQ	нет	16AI/16AQ	32AI/32AQ	32AI/32AQ
9. Время исполнения булевых операций							
1,2 мкс/оп	0,8 мкс/оп	0,8 мкс/оп	0,8 мкс/оп	0,37 мкс/оп	0,37 мкс/оп	0,37 мкс/оп	0,37 мкс/оп
10. Таймеры/счетчики							
64/64	128/128	256/256	256/256	256/256	256/256	256/256	256/256
11. Быстрые счетчики							
1	3	3	3	4	4	6	6
12. Быстрые выходы							
-	2	2	2	2	2	2	2
13. Коммуникационные прерывания (передача/прием)							
1/1	1/1	1/2	2/4	1/2	1/2	1/2	2/4
14. Прерывания, управляемые временем							
1	2	2	2	2	2	2	2
15. Аппаратные прерывания							
1	4	4	4	4	4	4	4

### Идентификаторы областей памяти

Идентификатор области памяти	Назначение области памяти	Форматы адресации	Примеры адресации	Примечание
<b>I</b>	Отображение информации на входах	Все	I0.0, IB4, IWO, ID0	
<b>Q</b>	Отображение информации на выходах	Все	Q0.0, QB4, QWO, QD0	
<b>V</b>	Память переменных	Все	V1.0, VB40, VWO, VD100	
<b>M</b>	Внутренние маркеры	Все	MO.0, MB4, MW0, MEЮ	
<b>S</b>	Шаговое управление	Все	S0.0, SB4, SWO, SD0	
<b>SM</b>	Специальные маркеры	Все	SMO.0, SMB4, SMWO, SMD0	
<b>T</b>	Таймеры	Бит, слово	T32	Формат - по контексту
<b>C</b>	Счетчики	Бит, слово	CO	Формат - по контексту
<b>AI</b>	Аналоговые входы	Слово	AIW0	Только четные адреса
<b>AQ</b>	Аналоговые выходы	Слово	AQW4	Только четные адреса
<b>AC</b>	Аккумуляторы	Все, кроме бита	ACO	Формат - по контексту
<b>HC</b>	Быстрые счетчики	Двойное слово	HC1	Обращение к текущему значению

## Клавиши панели инструментов STEP 7 - Micro/WIN

Внешний вид клавиши	Название		Назначение
	Английское	Русское	
	Download	Загрузить	Передать программу из компьютера в ПЖ
	Upload	Выгрузить	Передать программу из ПЖ в компьютер
	Compile All	Компиляция всех блоков	Компиляция всех программных блоков
	Compile	Компиляция	Компиляция текущего программного блока
	Run	Пуск	Перевод контроллера в режим пуска программы
	Stop	Стоп	Прервать выполнение программы контроллером
	Toggles Symbol Inf-ion Table	Таблица символьных имен сети	Вывод части таблицы символьных имен, используемых в данной сети, непосредственно в программе
	Insert Network	Вставка сети	Вставка сети перед активной сетью
	Delete Network	Удаление сети	Удаление активной сети
	Toggles Network Comments	Комментарий к сети	Вывод на экран комментариев к сети
	Toggles POU Comments	Комментарий к программе	Вывод на экран комментариев к программе
	Box	Блок	Вывод на экран списка всех блоков
	Coil	Катушка	Вывод на экран списка всех операций над выходами
	Contact	Контакт	Вывод на экран списка всех операций над контактами
CTRL+Z	–	–	Отмена последней операции
CTRL+C	–	–	Сохранение в буфере обмена
CTRL+V	–	–	Извлечение из буфера обмена

### Библиографический список

1. Система автоматизации S7-200. Системное руководство C79000-G7000-C230-01. SIMATIC S7. Программируемый контроллер SIMATIC S7-200 [Текст] - руководство пользователя. ООО Сименс. Отдел систем автоматизации A&D AS.
2. SIEMENS. SIMATIC. Системное руководство по программируемым контроллерам S7-200 [Текст] - руководство пользователя. C79000 - G7076 - C233 -02. - Выпуск 2.

<http://kp580bm.narod.ru/>