

# Программное обеспечение



Содержание	Стр.
Запуск Step7/MicroWIN	2
Вид окна начальной загрузки	3
Установка процессора	4
Установка коммуникационных параметров	5
Установка коммуникационного интерфейса	6
Свойства интерфейса PC/PPI	7
Поиск удаленной станции	8
Дерево проекта	9
Системный блок	10
Окна и редакторы	18
Перекрестные ссылки	19
Блок данных	20
Символьная таблица	21
Карта статуса памяти	22
Панель инструментов	23
Создание программы	28
Загрузка программы	29
Пуск и остановка работы	30
Тестирование и наблюдение статуса программы	31

## Запуск Step7/MicroWIN



ИЛИ

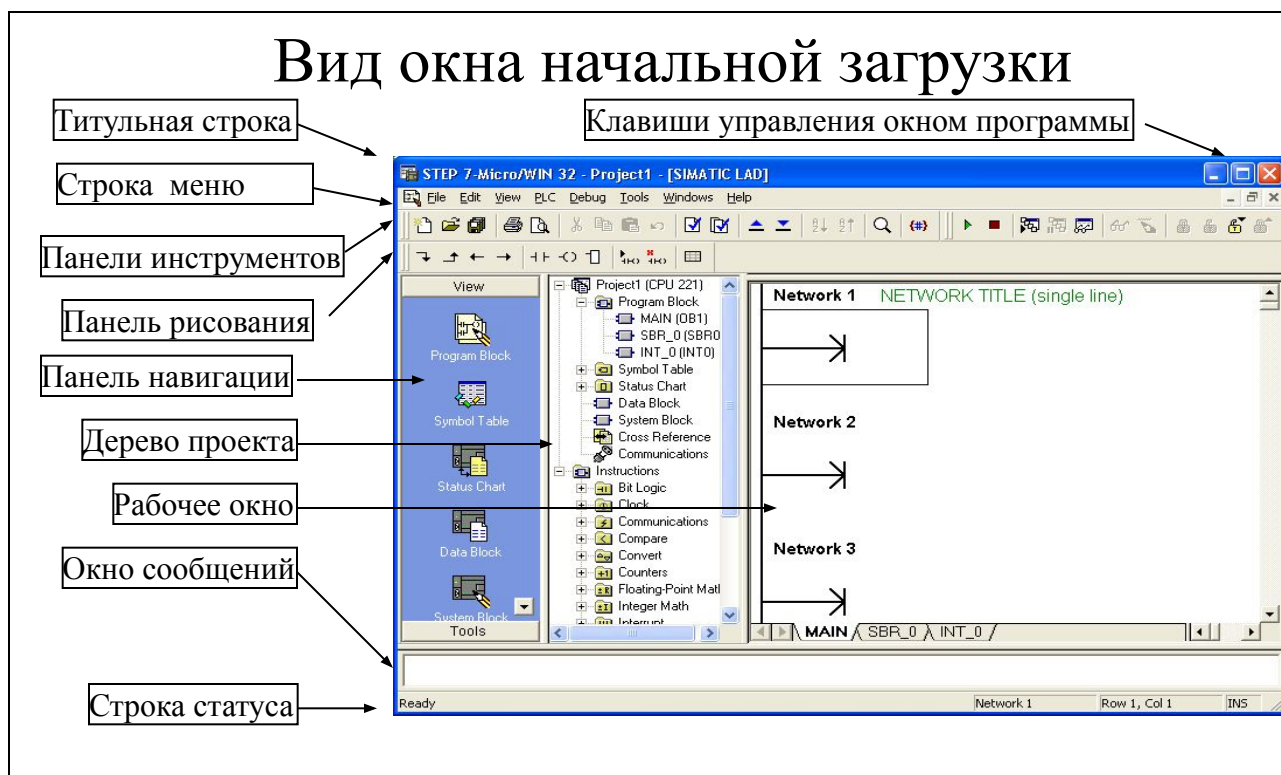


### Начало работы

Имеется два способа запуска программного обеспечения Step7/MicroWIN:

1. На панели задач нажать на кнопку *Start*- выбрать папку *Simatic - Step7 - MicroWIN32V 3.1.1.6 - Step7 - MicroWIN32*;
2. Щелкнуть на иконке *V3.1Step7 MicroWIN SP1* на рабочем столе.

## Вид окна начальной загрузки



### Титульная строка

Титульная строка содержит название окна- программы, окна –редактора , проекта и кнопки управления окном программы.

### Строка меню

Содержит все меню, доступные для текущего окна. При переходе в другое окно число и состав падающих меню изменяется.

### Панель инструментов

Содержит кнопки наиболее часто используемых пунктов меню.

### Панель рисования

Содержит элементы программ и имеет разный состав в зависимости от выбранного языка программирования.

### Панель навигации

Содержит кнопки позволяющие быстро перемещаться между разделами проекта.

### Дерево проекта

Содержит разделы проекта и группы команд.


### Рабочее окно


Предназначено для программирования и отображения редакторов программы.


### Строка статуса

Отображает текущее местоположение курсора, режим работы ПЛК, режим мониторинга и редактирования, сообщение об ошибках, время скана.

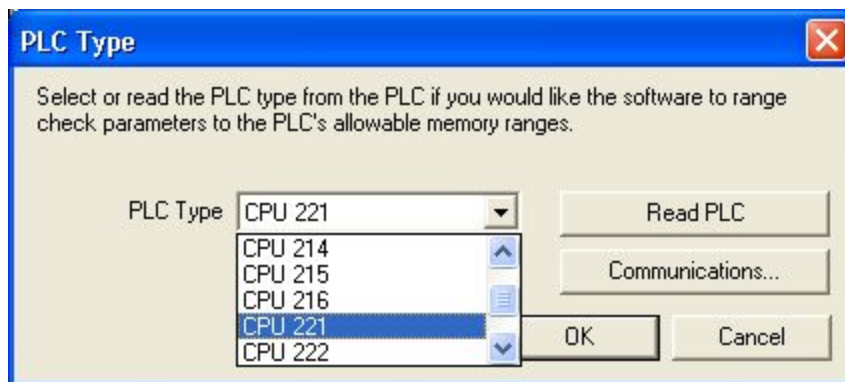
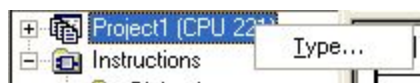
### Клавиши управления окном

Позволяют «сворачивать» окно программы на панель задач 

Управлять размером окна программы 



Закрывать окно программы 

## Установка процессора

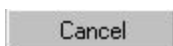


Для установки типа процессорного устройства необходимо вызвать контекстное меню в разделе **Project** дерева проекта, щелкнув правой клавишей. После выбора пункта **Type** появиться диалоговое окно **PLC Type**.

Существует два способа установки CPU:

- выбор типа из списка,  когда проект составляется без наличия CPU;
- автоматическое считывание,  для этого необходимо иметь само CPU, которое включено и соединено кабелем с устройством программирования.

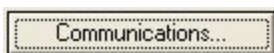
Из этого же окна можно перейти в окно настройки параметров коммуникации.



## Установка коммуникационных параметров

Вызов диалога:

- из PLC Type



- на панели навигации



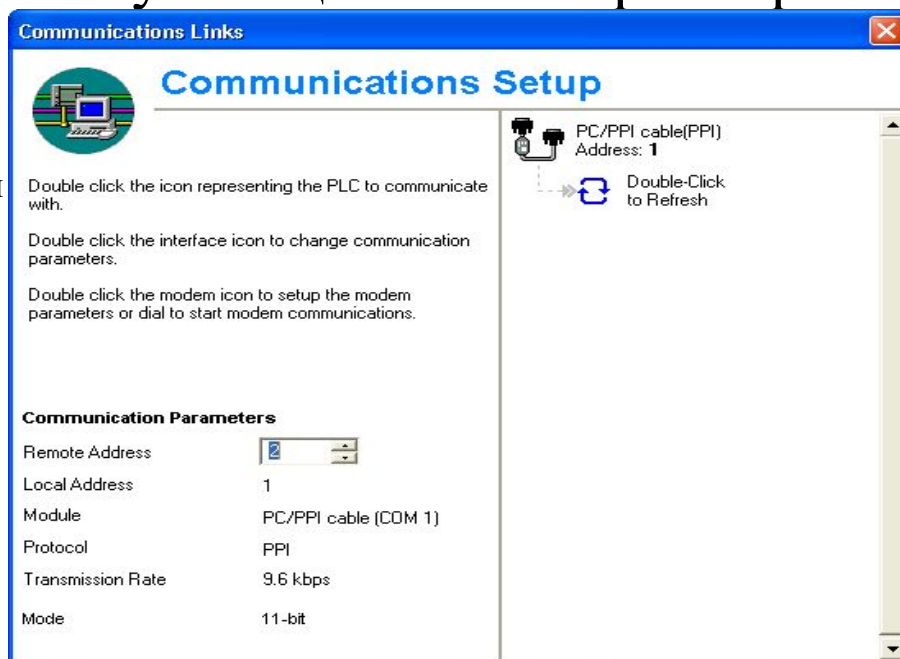
- в дереве проекта



- из МЕНЮ



Communications



**Remote Address** – адрес удаленной станции, в данном случае процессорного устройства, с которым устанавливается режим OnLine. Значение на счетчике должно соответствовать тому, которое задано в параметрах конфигурации порта CPU. По умолчанию это значение 2.

**Local Address** – адрес станции, с которой производится программирование контроллера, в данном случае ПЭВМ, для которой выделяется 0 или 1.

**Module** – тип коммуникационного модуля, через который устанавливается режим OnLine. В данном случае это кабель PC/PPI.

**Protocol** – тип коммуникационного протокола порта контроллера. PPI это стандартный протокол точка к точке.

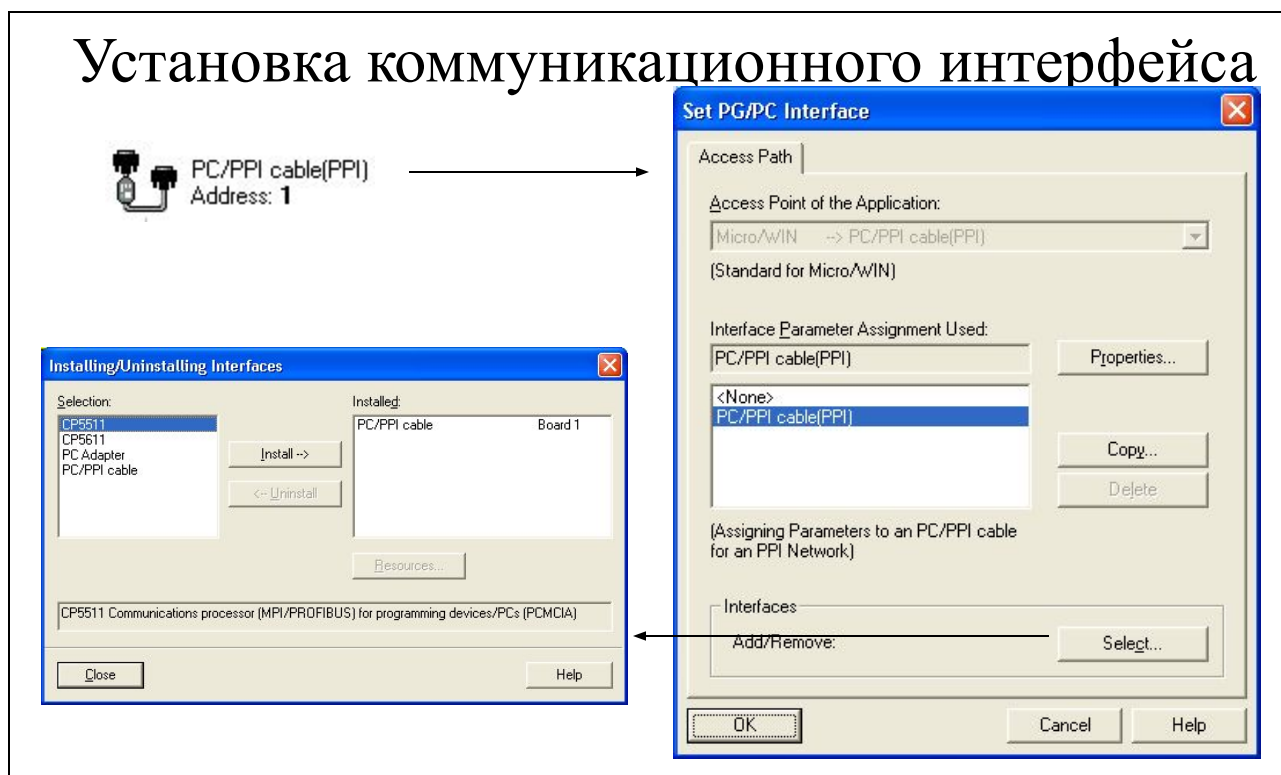
**Transmission Rate** – скорость обмена данными в килобитах в секунду.

**Mode** – размер структуры коммуникационного протокола.

OK

Cancel

## Установка коммуникационного интерфейса



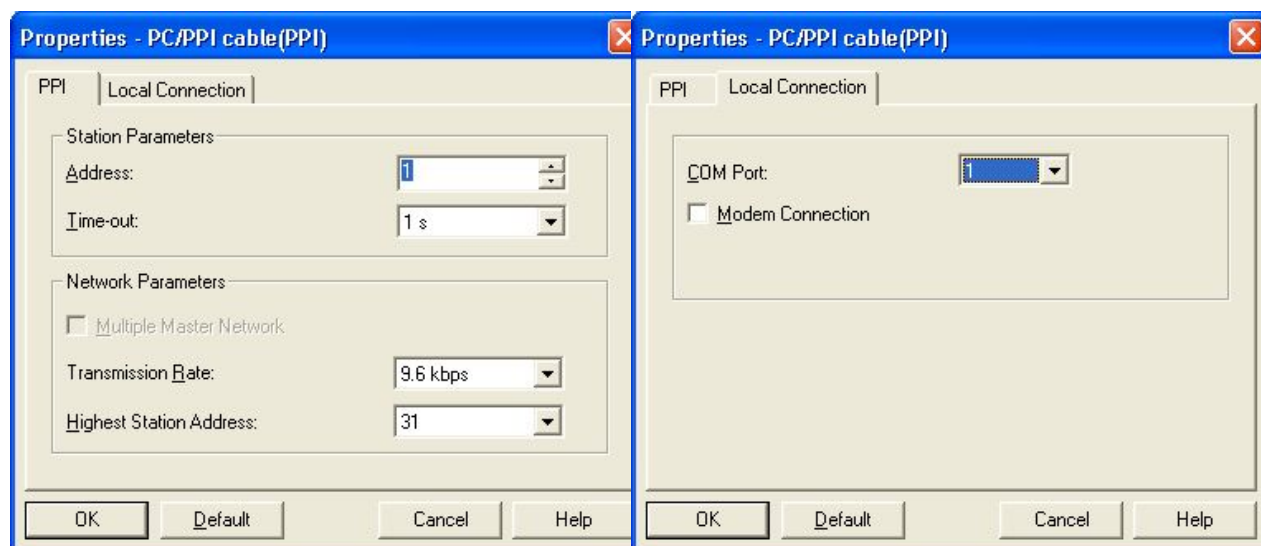
Для установки коммуникационного интерфейса необходимо дважды щелкнуть на клавише в правой части окна Communication Links.

В окне появится список установленных драйверов поддержки коммуникационных модулей. При необходимости можно удалять или добавлять элементы этого списка.

При установке платы коммуникационного процессора в компьютер необходимо выбрать команду Select и тип процессора в списке окна **Installing/Uninstalling Interfaces**. В нижнем окне, при выборе типа модуля появляется его описание:

- CP5511 – коммуникационный процессор MPI/ Profibus через PCMCIA;
- CP5611 – коммуникационный процессор MPI/ Profibus через PCI;
- PC Adapter - MPI/ Profibus интерфейс через COM порт;
- PC/PPI Cable – PPI интерфейс через COM порт.

## Свойства интерфейса PC/PPI



**Закладка PPI** устанавливает параметры станции (контроллера) и сетевые параметры.

- Address – адрес локальной станции (ПЭВМ);
- Time – out – время ожидания ответа от удаленной станции (контроллера);
- Transmission Rate – скорость обмена данными;
- Highest Station Address – максимальный адрес удаленной станции, до которого будет выполняться поиск.

**Закладка Local Connection** устанавливает номер COM – порта к которому подключено процессорное устройство а также выбор опции соединения при помощи модема.

## Поиск удаленной станции

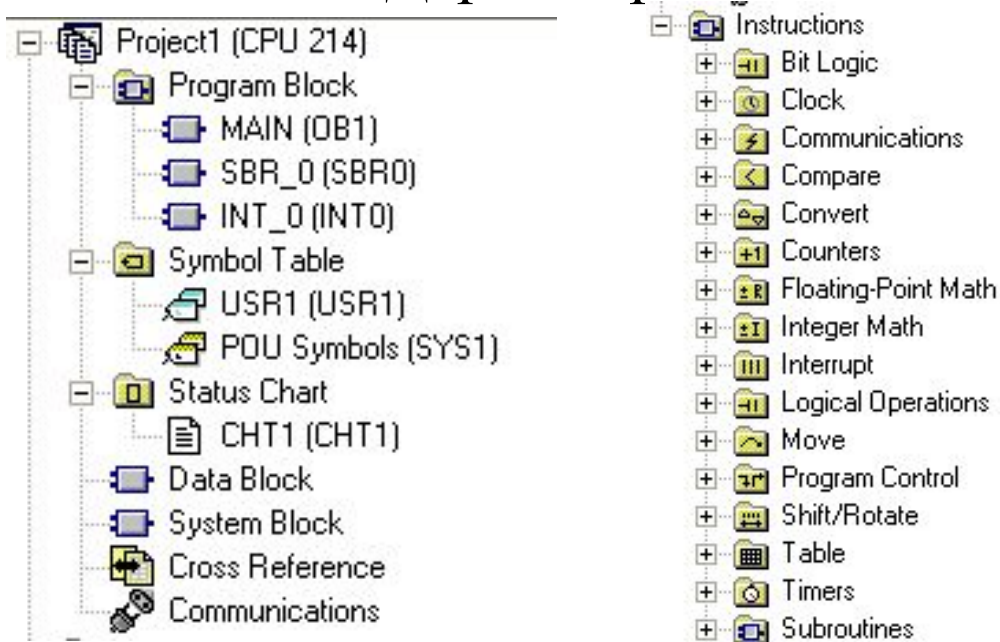
**Закладка PPI** устанавливает параметры станции (контроллера) и сетевые параметры.

- Address – адрес локальной станции (ПЭВМ);
- Time – out – время ожидания ответа от удаленной станции (контроллера);
- Transmission Rate – скорость обмена данными;
- Highest Station Address – максимальный адрес удаленной станции, до которого будет выполняться поиск.

**Закладка Local Connection** устанавливает номер COM – порта к которому подключено процессорное устройство а также выбор опции соединения при помощи модема.



## Дерево проекта



Дерево проекта состоит из двух основных ветвей : Проект и Инструкции.

**Имя проекта**, которое соответствует имени сохраненного файла, и тип процессорного устройства.

### Программные блоки:

- Main (OB1) – блок циклической обработки. Всегда обязателен;
- SBR\_0 (SBR0) – подпрограмма. Необязательный элемент.
- INT\_0 (INT0) – программа обработки прерывания.

*Первая надпись – имя программного блока, надпись в скобках – адрес блока.*

**Символьная таблица** содержит имена и комментарии используемых адресов памяти **USR**. Ее создание способствует более наглядному представлению программы. **POU Symbols** – таблица содержащая имена и комментарии блоков программы.

**Карта статуса** – таблица чтения памяти, мониторинг значений и их изменение.

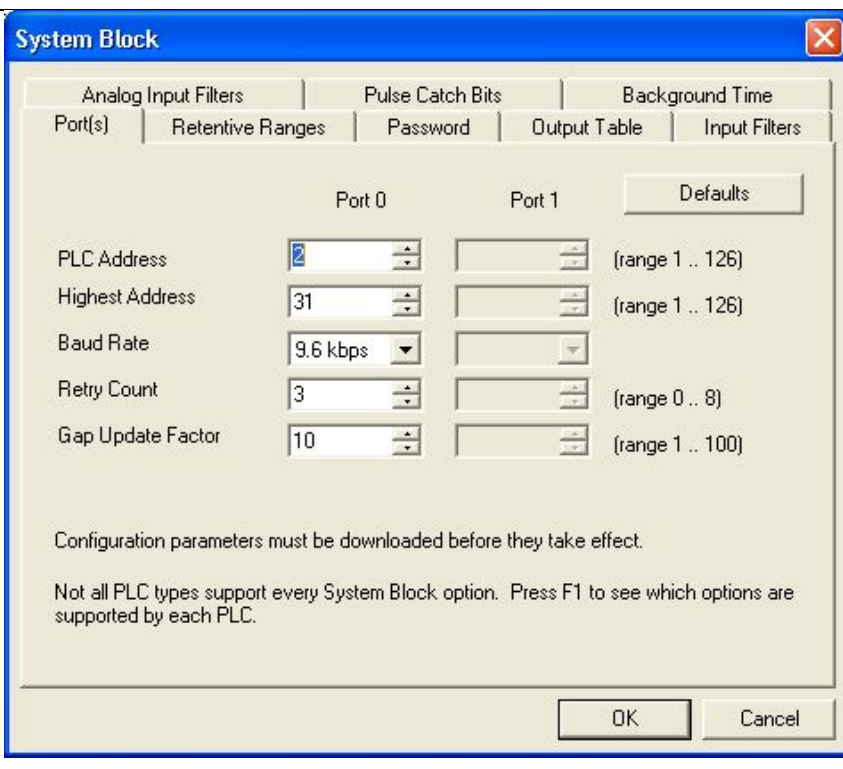
**Блок данных** редактор создания массива значений и констант в области переменных (область V);

**Системный блок** диалог изменения системных параметров процессорного устройства;

**Перекрестные ссылки** таблица использованных в программе адресов памяти;

**Коммуникации** (см. предыдущие слайды)

# Системный блок: порт

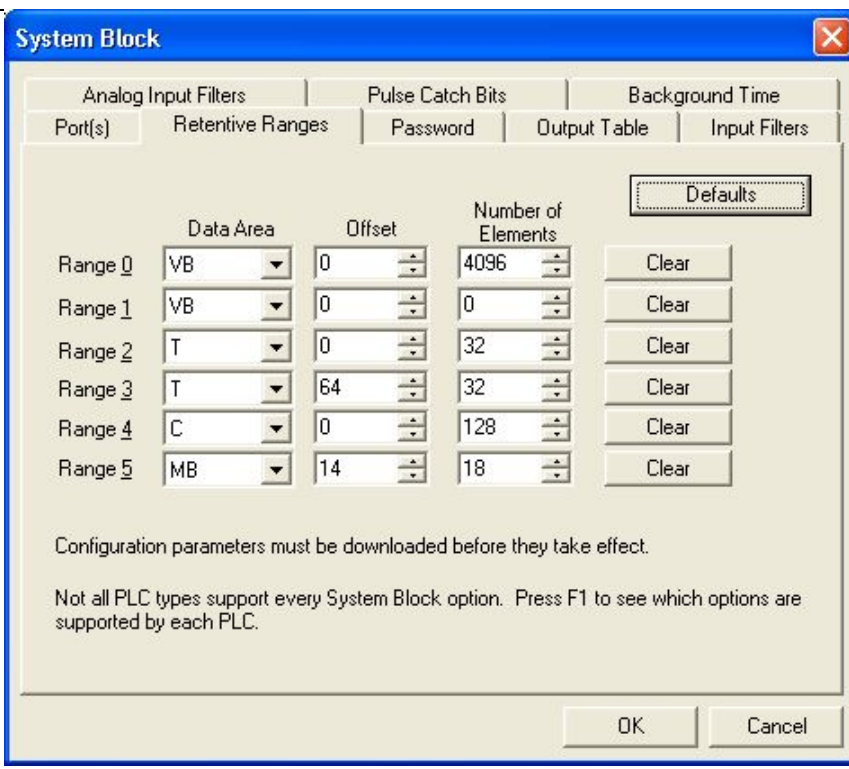


- PLC Address** установка адреса удаленной станции (порта контроллера);
- Highest Address** установка старшего адреса контроллера;
- Baud Rate** установка скорости обмена данными;
- Retry Count** установка количества повторов передачи данных;
- Gap Update Factor** установка коэффициента пропуска адресов, величина которого зависит от размера сети.

Установка параметров коммуникационного порта зависит от типа процессорного устройства и версии программного обеспечения и представлена в таблице.

PLC Ver.	210	212	212	212	212	212	214	214	214	214	214	215	215	216	216	216	221	222	224	226
	0.91	0.92	1.00	1.01	1.10	1.10	.91	1.00	1.01	1.10	1.11	1.00	1.02	1.00	1.01	1.02				
<b>Port</b>	0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1													✓	✓	✓				✓
	DP											✓	✓							
<b>Baud Rate</b>	9.6K	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	19.2K						✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	93.75K											✓	✓							
	187.5K											✓	✓				✓	✓	✓	✓
	500K											✓	✓							
	1.5M											✓	✓							
	3M											✓	✓							
	6M											✓	✓							
	12M											✓	✓							

## Системный блок: сохраняемая память



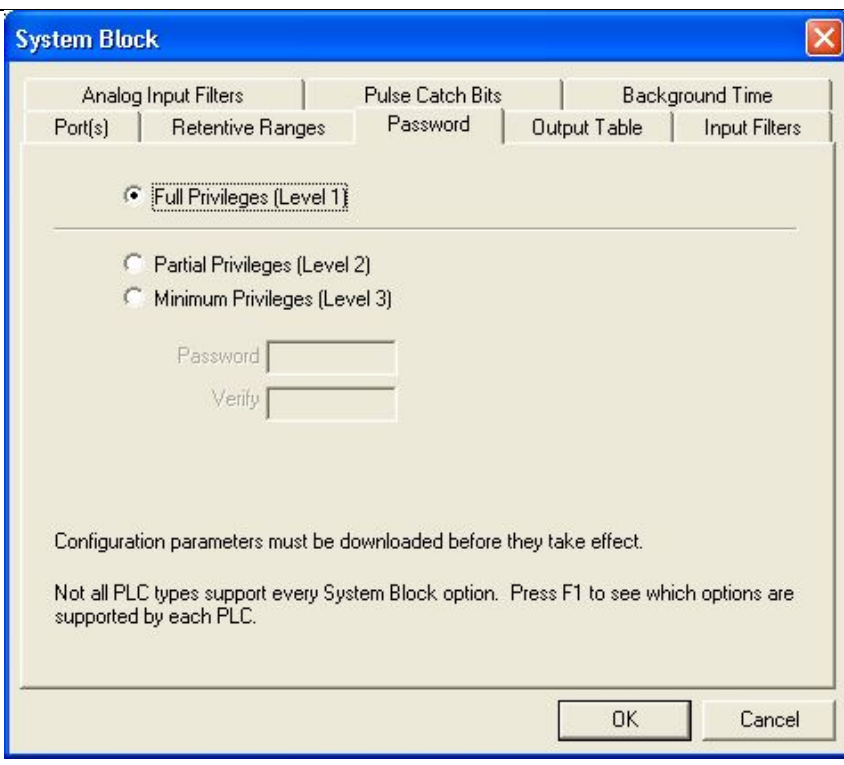
Все модели PLC имеют суперконденсатор, который поддерживает данные ОПЕРАТИВНОЙ ПАМЯТИ после того, как питание PLC выключено. Некоторые модели PLC могут для этих целей использовать батарейный модуль, который увеличивает время хранения данных. Батарейный модуль начинает работать только после разрядки конденсатора.

Вы можете определять до шести удерживающих диапазонов в области памяти V, M, C, и T. Для таймеров, только удерживающие таймеры (TONR) могут быть сохранены, и только текущее значение для таймеров и счетчиков может быть определено как удерживающееся. Биты таймеров и счетчиков обнуляются при каждом включении электропитания.

При включении электропитания: PLC проверяет память, чтобы проверить, что конденсатор или батарея успешно удержали сохраненные данные. Если данные ОПЕРАТИВНОЙ ПАМЯТИ успешно поддерживались, то программа пользователя и Конфигурация PLC также восстанавливаются от СППЗУ. Все другие области памяти сбрасываются в 0.

Если содержимое ОПЕРАТИВНОЙ ПАМЯТИ не поддерживалось (как после сбоя питания), PLC очищает ОПЕРАТИВНУЮ ПАМЯТЬ и устанавливает бит контроля ретанентных данных в области специальных маркеров (SM 0.2).

# Системный блок: пароль

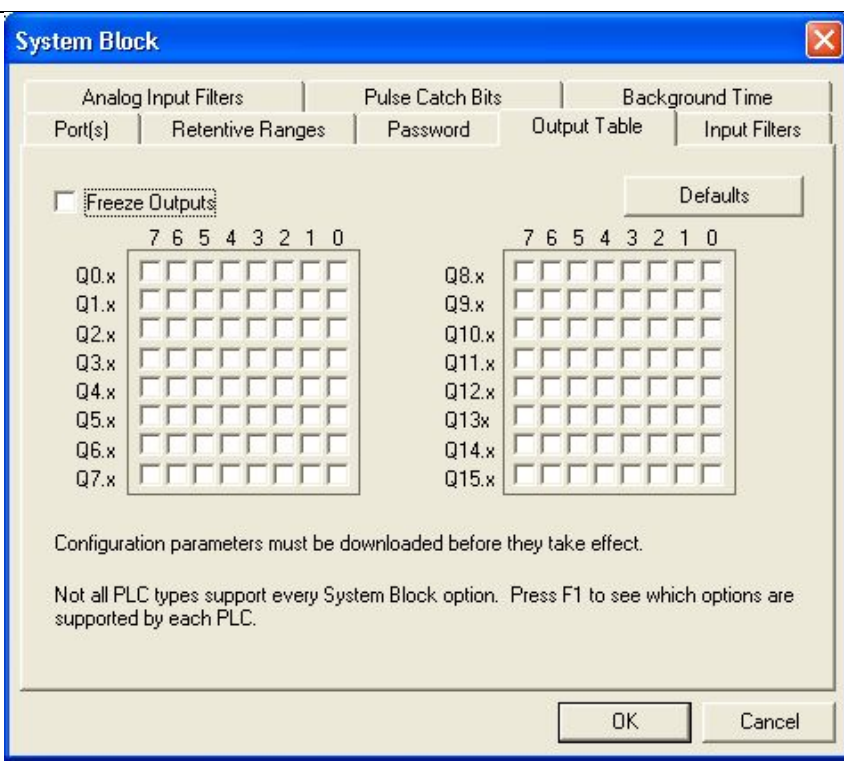


Все варианты CPU S7–200 предоставляют парольную защиту для ограничения доступа к определенным функциям CPU. Благодаря установке пароля доступ к определенным функциям и памяти CPU имеют только уполномоченные лица. При отсутствии пароля CPU неограниченный доступ. Если парольная защита установлена, то CPU запрещает неуполномоченным лицам все операции, ограниченные в конфигурации пароля.

Функция	Уровень защиты 1	Уровень защиты 2	Уровень защиты 3
Чтение и запись данных пользователя	Не ограничены	Не ограничены	Не ограничены
Запуск, останов и новый запуск CPU			
Чтение и установка часов реального времени			
Чтение принудительно установленных данных CPU		Нужен пароль	
Загрузка программы пользователя, данных и конфигурации из CPU			
Загрузка в CPU		Нужен пароль	
Стирание программы пользователя, данных и конфигурации <sup>1</sup>			
Принудительная установка данных и выполнение определенного количества циклов			
Копирование в модуль памяти			
<sup>1</sup> Защита от стирания может быть преодолена паролем "clearplc".			



# Системный блок: таблица выходов



С помощью CPU S7–200 Вы можете устанавливать состояния сигналов цифровых выходов на определенные значения при переходе в режим работы STOP, либо Вы можете оставлять выходы точно в том состоянии, в котором они находились при переходе в STOP.

По умолчанию, все выходы обнуляются при переходе контроллера в режим STOP.

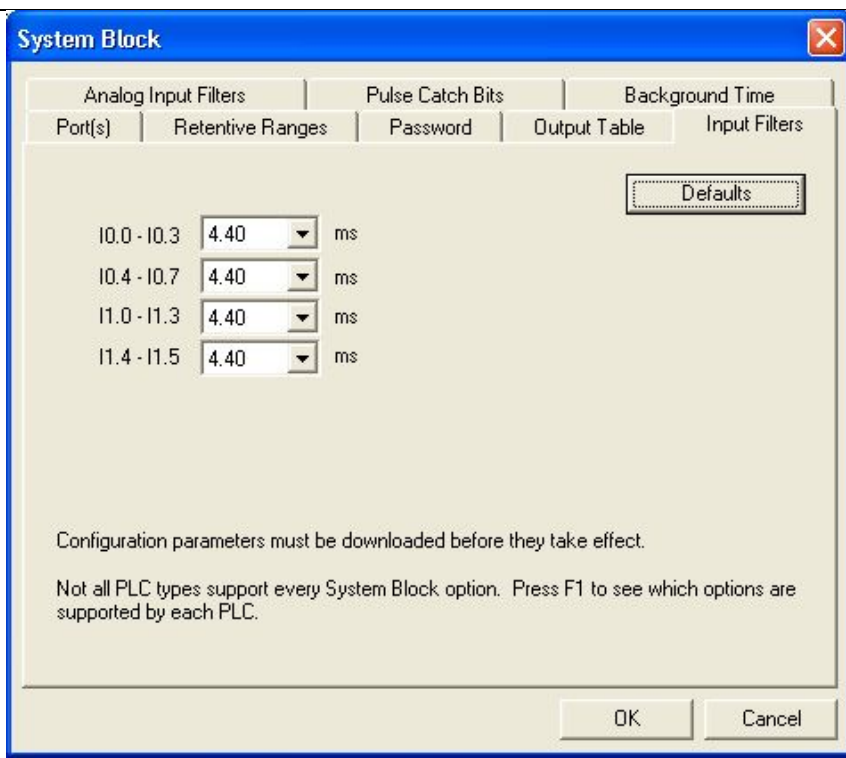
При необходимости установки выходов в единичное состояние, необходимо отметить соответствующие адреса в таблице.

Если необходимо, чтобы выходы сохраняли свое состояние при переходе в режим STOP то необходимо включить функцию Freeze Outputs.

Установки выходов являются частью данных конфигурации CPU для системы, которые загружаются в память CPU и хранятся там.

Конфигурирование выходных значений возможно только для цифровых выходов. Аналоговые выходы при переходе в режим работы STOP замораживаются. Это происходит, так как Ваша программа ответственна за актуализацию аналоговых выходов. Актуализация аналоговых входов и выходов не является системной функцией CPU. Для аналоговых входов и выходов отображение в памяти CPU не хранится.

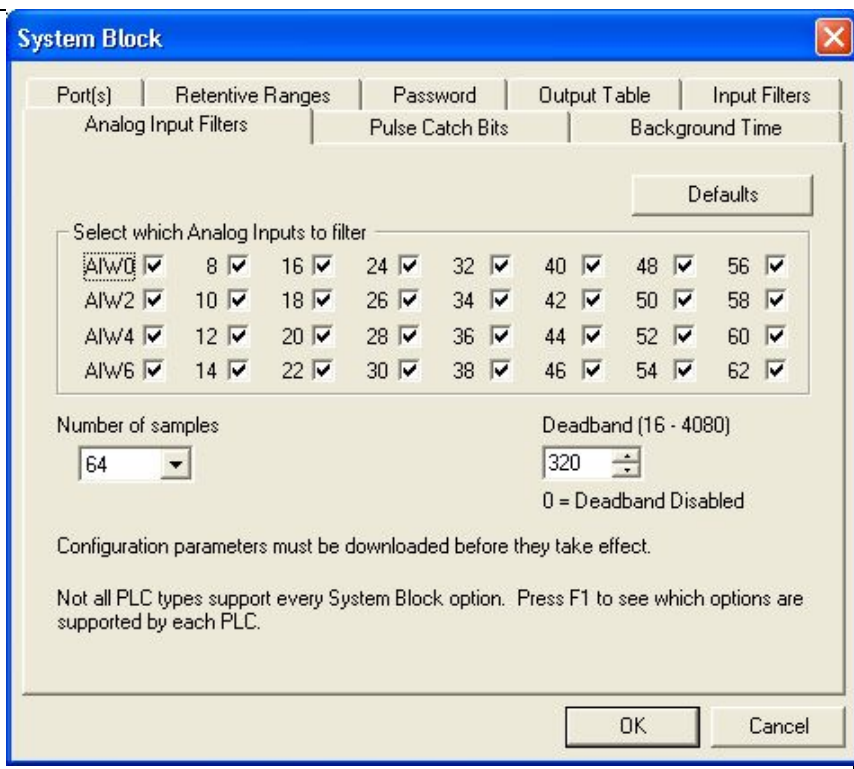
# Системный блок: входной фильтр



Вы можете выбрать для Вашего CPU S7–200 фильтр ввода, который определяет для физических входов время задержки (регулируемое в диапазоне от 0,2 мс до 8,7 мс). Это время задержки прибавляется к обычному времени ответа для групп из четырех входов. Это время задержки служит для того, чтобы отфильтровать в соединительной проводке входа помехи, которые могут вызвать непреднамеренные изменения состояний сигналов на входах.

Входной фильтр является частью данных конфигурации CPU, загружаемых в память CPU и хранимых там.

## Системный блок: фильтр аналоговых входов



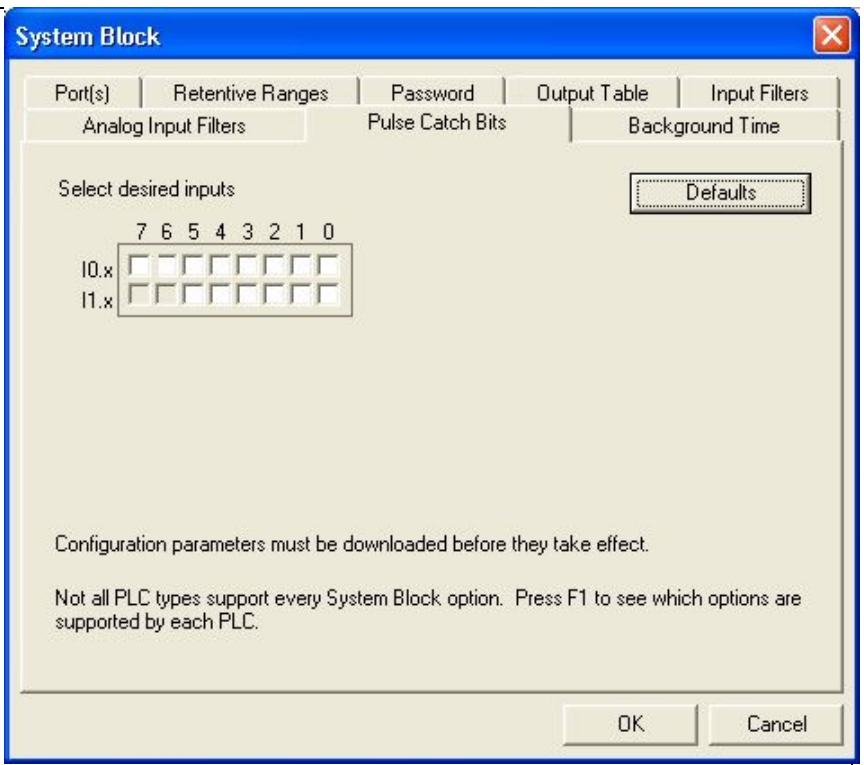
У CPU 222, CPU 224 и CPU 226 вы можете выбрать программную фильтрацию на отдельных аналоговых входах. Отфильтрованное значение – это среднее значение суммы предварительно выбранного количества отсчетов аналогового ввода. Описание фильтра (количество отсчетов и мертвая зона) одинаково для всех аналоговых входов, для которых разрешена фильтрация. Фильтр обладает быстрой реакцией, что позволяет большим изменениям быстро отражаться в отфильтрованном значении. Фильтр производит ступенчатое изменение по отношению к последнему значению аналогового входа, когда на этом входе превышает заданное отклонение от среднего значения. Это отклонение называется мертвой зоной. Она задается в единицах счета цифрового значения аналогового входа.

- **Number of samples** – количество отсчетов;
- **Deadband** – мертвая зона.

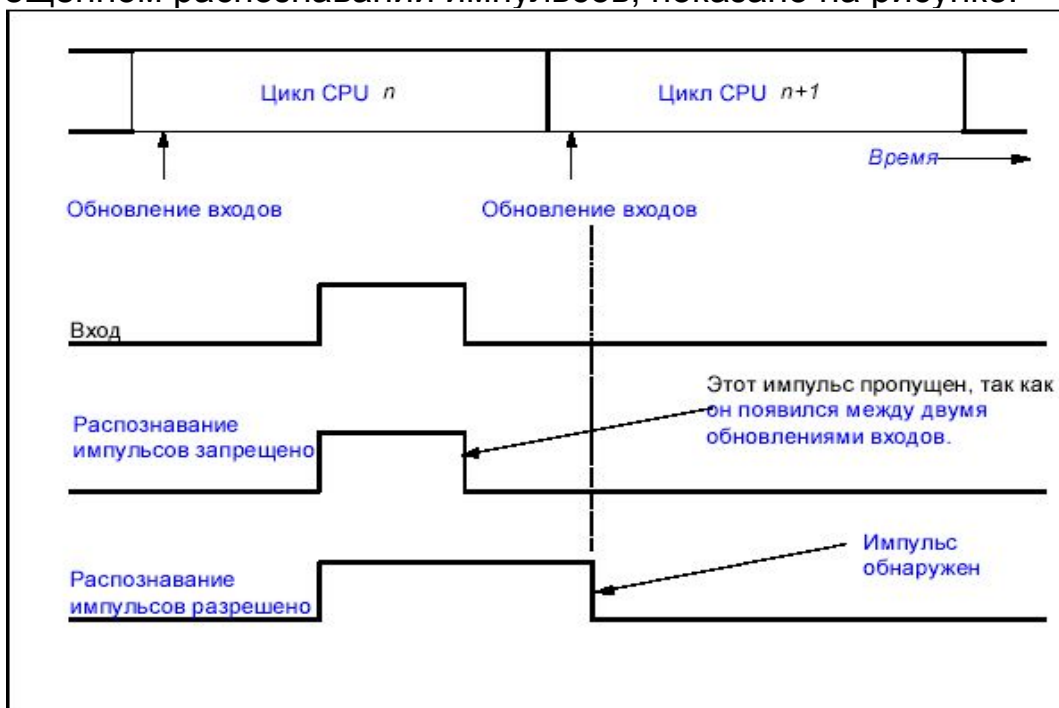
### Примечание

Аналоговую фильтрацию не следует использовать с модулями, которые передают в аналоговых словах цифровую информацию или аварийные сигналы. Используйте экран, показанный на рис. 6.9, чтобы заблокировать аналоговую фильтрацию для термометров сопротивления, термопар и master-модулей интерфейса AS.

# Системный блок: биты распознавания импульсов

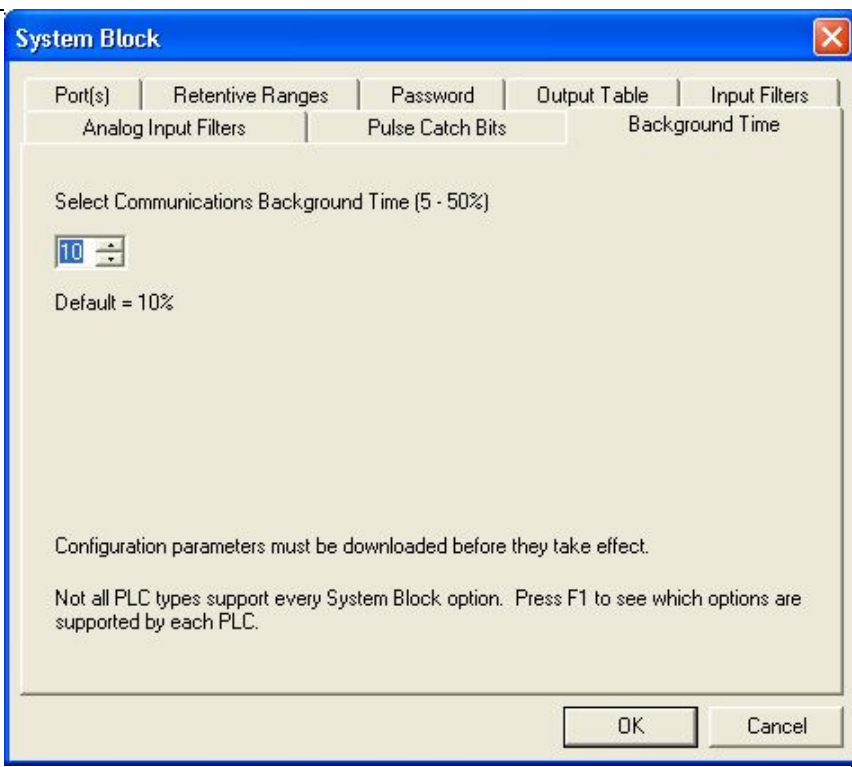


Операция распознавания импульсов может быть разрешена индивидуально для каждого из локальных цифровых входов. Когда распознавание импульсов для входа разрешено, изменение в состоянии этого входа фиксируется и удерживается до следующего циклического обновления входов. Благодаря этому кратковременный импульс распознается и удерживается, пока CPU не прочтает входы, гарантируя, что импульс не будет пропущен. Как действует ПЛК при разрешенном и запрещенном распознавании импульсов, показано на рисунке.





## Системный блок: фоновое время



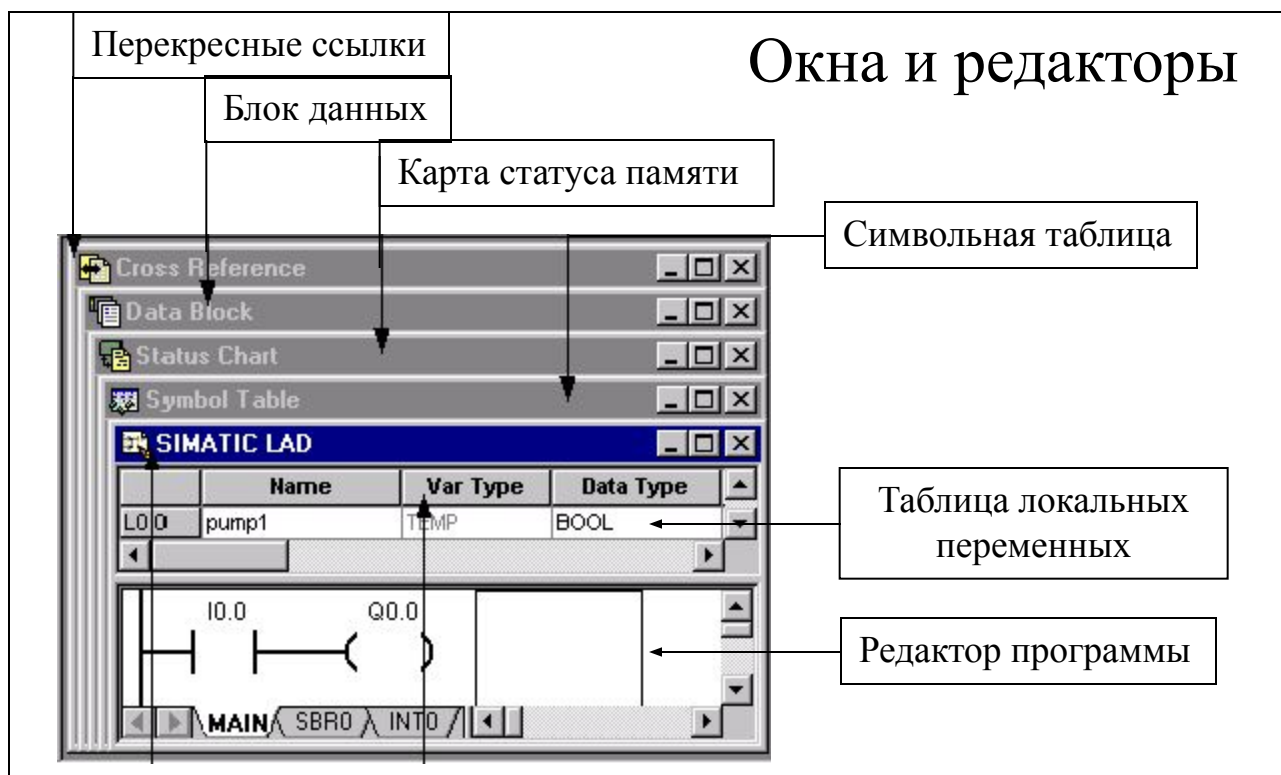
Вы можете задать долю цикла сканирования в процентах, которая будет выделяться обработке коммуникационных запросов, связанных с компиляцией при редактировании в режиме RUN или сбором информации о состоянии STL.

Когда вы увеличиваете процент времени, выделяемого обработке коммуникационных запросов, вы увеличиваете время сканирования, что замедляет выполнение процесс управления.

Отредактируйте характеристики коммуникационного фонового времени и затем загрузите изменения в CPU.

Процент времени сканирования, выделяемый обработке коммуникационных запросов, по умолчанию устанавливается равным 10 %. Такая настройка была выбрана для того, чтобы обеспечить разумный компромисс для обработки операций компиляции/вывода состояния при минимизации влияния на ваш процесс управления. Вы можете корректировать это значение с шагом 5 % до максимального значения 50 %.

## Окна и редакторы



**Перекрестные ссылки** – окно, позволяющее посмотреть какие символьные имена и адреса памяти были использованы в прикладной программе. И дает информацию и местонахождении этих элементов в структуре программы. Окно имеет три закладки: перекрестные ссылки, используемые байты и используемые биты памяти.

**Блок данных** – позволяет задавать начальные значения данных в памяти V (памяти переменных). Можно задавать значения байт, слов, или двойных слов. Типичная строка в редакторе данных, содержит начальный адрес и один или большее количество значений данных. Комментарии отделяются косыми чертами и могут быть произвольными. После ввода текста строки нажимается ENTER - редактор форматирует строку (выравнивает адреса, данные, комментарии; печатает прописными буквами V адреса памяти).

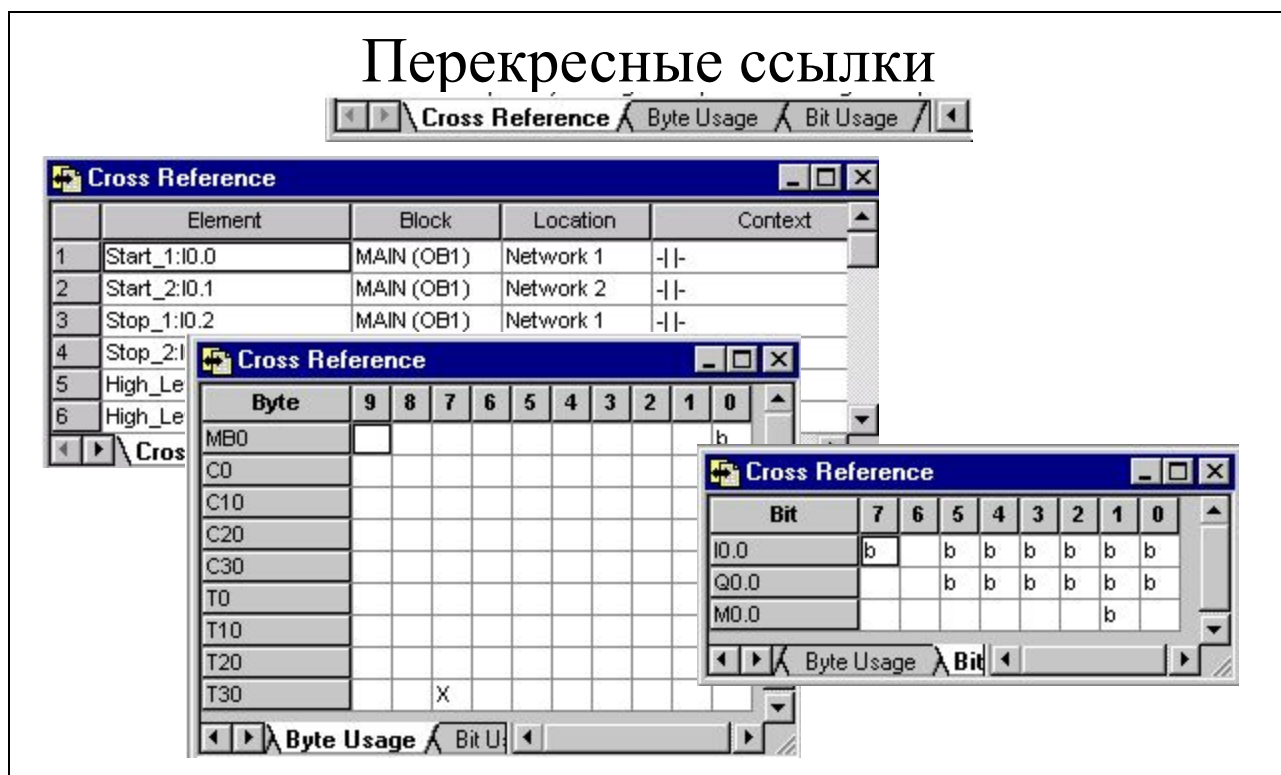
**Карта статуса памяти** – позволяет считывать, записывать, принудительно задавать и контролировать переменные во время выполнения программы,

**Символьная таблица** – позволяет присвоить используемым адресам памяти символьное имя. Ее создание способствует более наглядному представлению программы.

**Редактор программы** – рабочее поле для создания программы на языках LAD, FBD, STL.

**Таблица локальных переменных** – Каждый программный блок в вашей программе имеет L - стек памяти (64 бита), который позволяет определить переменные, которые ограничены пределами только этого программного блока в отличие от переменных символьной таблицы.

## Перекрестные ссылки



**Cross Refrrence** – Используйте таблицу Перекрестной ссылки, когда Вы хотите знать, используется ли символьное имя или адрес памяти в вашей программе, и в каком ее месте (идентифицирует программный блок, цепочку, и контекст команды операнда).

Вы можете переключать между символическим и абсолютным адресом.

- Element – имя или адрес памяти;
- Block – идентификатор программного блока;
- Location – идентификатор вычислительной цепочки;
- Context – идентификатор команды.

**Byte Usage и Bit Usage** – Таблица использования байта и бита позволяет видеть которые байты и биты, в какой области памяти, использовались в вашей программе. Это помогает избежать ошибок с дублированием адресов.

- b Указывает, что бит памяти был назначен.
- B Указывает, что байт памяти был назначен.
- W Указывает, что слово (16 битов) было назначено.
- D Указывает, что двойное слово (32 бита) было назначено.
- X Используется для таймеров и счетчиков.

## Блок данных

```

Data Block
//
//DATA BLOCK COMMENTS
//
VB0    255           //byte value starting at VB0
VM2    256           //word value starting at VM2
VD4    700.59        //double word real number starting at VD4
VB8    -35           //byte value starting at VB8
VM10   16#A          //word value HEX starting at VM10
VD14   146879        //double word value starting at VD14
VM20   2, 4, 8, 16, 32, 64 //table of word values starting at VM20
        -2, -4, -8, -16, -32, -64 //Values that extend to multiple lines cannot be in col 1
//
        The first column is reserved for memory address
VB45   'Up'          //two-byte ASCII string starting at VB45
VM90   65535         //word value starting at VM90
V100   255, 'abc', 65535, 1.0 //Undefined value starting at vb100: Can mix different sizes
VB110  255           //Same as entering B110 255
VM120  256           //Same as entering W120 255
VD130  700.59        //Same as entering D130 700.59
V140   255, 'abc', 65535, 1.0 //Same as entering 140: Must start in column 1
VM150  2#1010101010101010 //Binary word value
VD152  2#11001100110011001100110011001100 //Binary double word value
    
```

При заполнении блока данных необходимо руководствоваться диапазоном значений который соответствует адресам памяти.

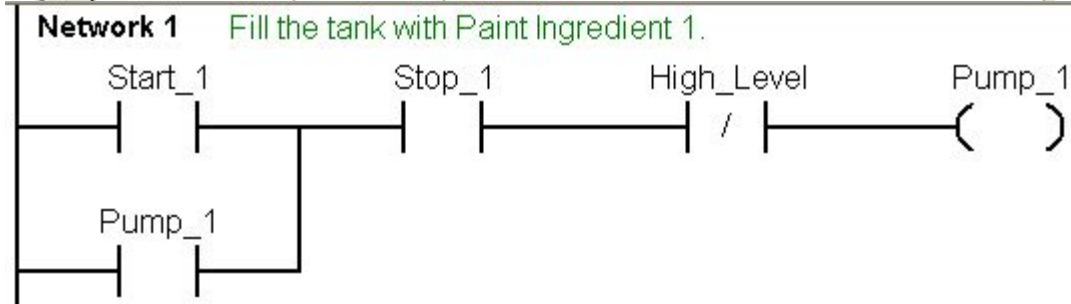
	Unsigned Range		Signed Range	
Data Size:	Decimal:	Hexadecimal:	Decimal:	Hexadecimal:
B (Byte)	0 – 255	0 – FF	-128 - +127	80 - 7F
W (Word)	0 – 65535	0 – FFFF	-32768 - +32767	8000 - 7FFF
D (Double Word)	0 – 4294967295	0 – FFFF FFFF	-2147483648 - +2147483647	8000 0000 - 7FFF FFFF
Data Size:	Decimal Real (Positive)		Decimal Real (Negative)	
D (Double Word)	+1.175495E-38 - +3.402823E+38		-1.175495E-38 - - 3.402823E+38	

Форматы данных различных систем счисления.

<b>2#</b>	for binary numbers	Example: 2#1101
<b>16#</b>	for hexadecimal numbers	Examples: 16#3FB2
'abcd'	for ASCII literal strings (begin and end the ASCII string with single quote characters)	Example: ↑ YES ↑

## Символьная таблица

	Name	Address	Comment
1	Start_1	I0.0	Запуск 1 крана
2	Start_2	I0.1	Запуск 2 крана
3	Stop_1	I0.2	Остановка 1 крана
4	Stop_2	I0.3	Остановка 2 крана
5	High_Level	I0.4	Датчик минимального уровня
6	Low_Level	I0.5	Датчик максимального уровня
7	Reset	I0.7	Сброс
8			
9	Pump_1	Q0.0	Насос№1



## Карта статуса памяти

	Address	Format	Current Value	New Value
1	Start_1	Bit	2#1	
2	Start_2	Bit	2#0	
3	Stop_1	Bit	2#1	
4	Stop_2	Bit	2#0	
5	High_Level	Bit	2#0	
6	Low_Level	Bit	2#0	
7	Reset	Bit	2#0	
8		Signed		
9	Pump_1	Bit	2#1	
10	Pump_2	Bit	2#0	
11	Mixer_Motor	Bit	2#0	
12	Steam_Valve	Bit	2#0	
13	Drain_Valve	Bit	2#1	
14	Drain_Pump	Bit	2#1	
15		Signed		
16	High_Lev_Reached	Bit	2#1	
17	Mix_Timer	Signed	+32767	
18	Cycle_Counter	Signed	+0	

Для мониторинга памяти необходимо ввести адреса в поле **Address** и выполнить команду **Single Read** на панели **Debug** - однократно считанное значение отобразится в поле **Current Value** или включить **Chart Status** и тогда в поле **Current Value** будет периодически отображаться считываемая информация.

При вводе адреса можно выбрать формат отображаемых данных в поле **Format**



Десятичный со знаком  
Десятичный без знака  
Шестнадцатеричный  
Двоичный  
Текст а ASCII коде

Кроме наблюдения за состоянием памяти можно записывать значения. Для этого необходимо ввести требуемое значение в поле **New Value** в соответствующем формате напротив необходимого адреса.







Для однократной записи значения выполнить команду **Write All**.


















Для принудительной перезаписи значения в каждом новом скане **Force**. Но в этом случае в указанную область памяти будет всегда записываться только это значение, а не значение определяемое программой. Поэтому, после отладки программы необходимо выполнить команду **Unforce All**.



## Панели инструментов: стандартная

























	<a href="#">Open New Project</a>		<a href="#">Compile Program Block or Data Block (Whichever Is the Active Window)</a>
	<a href="#">Open Existing Project</a>		<a href="#">Compile All (Program Block, Data Block, and System Block)</a>
	<a href="#">Save Current Project</a>		<a href="#">Upload Project from PLC to STEP 7-Micro/WIN 32</a>
	<a href="#">Print</a>		<a href="#">Download from STEP 7-Micro/WIN 32 to a PLC</a>
	<a href="#">Print Preview</a>		<a href="#">Sort Ascending: Sort Symbol Table Name Column A-Z</a>
	<a href="#">Cut Selection and Copy to Clipboard</a>		<a href="#">Sort Descending: Sort Symbol Table Name Column Z-A</a>
	<a href="#">Copy Selection to Clipboard</a>		<a href="#">Zoom: Set Magnification of LAD and FBD Views</a>
	<a href="#">Paste Clipboard Contents at Cursor Location</a>		<a href="#">Constant Descriptor: Toggle ON/OFF SIMATIC Type Specifiers</a>
	<a href="#">Undo Last Entry</a>		

	Создать новый проект
	Открыть созданный и сохраненный проект
	Сохранить проект
	Распечатать проект
	Посмотреть перед распечатыванием
	Вырезать в буфер обмена
	Скопировать в буфер обмена
	Вставить из буфера обмена
	Отменить операцию редактирования
	Компилировать (проверить синтаксис) программу
	Компилировать программу, блок данных и системный блок
	Выгрузить из контролера в программатор
	Загрузить из программатора в контроллер
	Сортировать имена в символьной таблице по алфавитному возрастанию
	Сортировать имена в символьной таблице по алфавитному убыванию
	Изменить масштаб
	Отобразить размер константы B# - байт, W# - слово, D# - дв. слово

## Панели инструментов: отладка

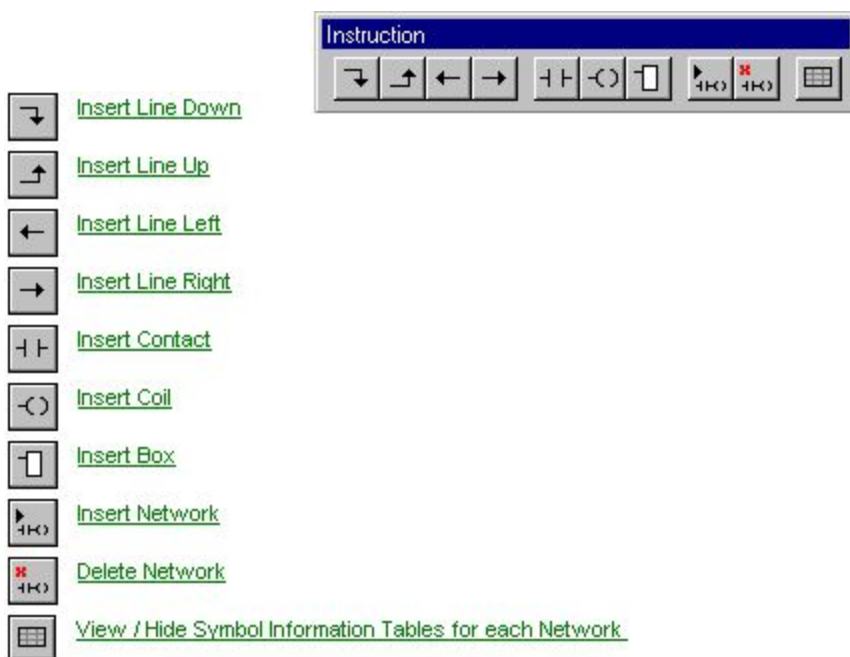












	<a href="#">Set PLC to RUN Mode</a>		<a href="#">Force PLC Data (Status Chart, LAD Edit, or FBD Edit)</a>
	<a href="#">Set PLC to STOP Mode</a>		<a href="#">Unforce PLC Data (Status Chart, LAD Edit, or FBD Edit)</a>
	<a href="#">Toggle Program Status ON/OFF</a>		<a href="#">Unforce All (Status Chart, LAD Edit, or FBD Edit)</a>
	<a href="#">Toggle Triggered Pause ON/OFF (STL only)</a>		<a href="#">Read All Forced Values (Status Chart, LAD Edit, or FBD Edit)</a>
	<a href="#">Toggle Chart Status ON/OFF</a>		
	<a href="#">Status Chart Single Read</a>		
	<a href="#">Status Chart Write All</a>		

-  Установить режим RUN
-  Установить режим STOP
-  Включить/ выключить наблюдение статуса программы LAD или FBD
-  Включить/ выключить триггерную точку программы STL
-  Включить/ выключить циклическое чтение карты статуса памяти
-  Однократное чтение карты статуса памяти
-  Записать значение в карту статуса памяти
-  Принудительно записать значение в карту статуса памяти и программы
-  Отменить принудительную запись значения в карту статуса памяти и программы
-  Отменить все принудительно установленные значения
-  Прочитать все принудительно установленные значения

















## Панели инструментов: команды LAD

















-  Вставить вертикальную линию «справа - вниз»
-  Вставить вертикальную линию «справа - вверх»
-  Вставить горизонтальную линию «влево»
-  Вставить горизонтальную линию «вправо»
-  Вставить контакт , с дальнейшим выбором типа из списка
-  Вставить катушку , с дальнейшим выбором типа из списка
-  Вставить инструкцию, с дальнейшим выбором типа из списка
-  Вставить пустую Network ниже текущей
-  Вставить текущую Network
-  Отобразить часть символьной таблицы с используемыми элементами в цепочке программы

## Панели инструментов: команды FBD



	<a href="#">Insert Line Down</a>		<a href="#">Add Extra Input to AND or OR Box</a>
	<a href="#">Insert Line Up</a>		<a href="#">Remove Extra Input from AND or OR Box</a>
	<a href="#">Insert Line Left</a>		<a href="#">Toggle Negate: (Invert Logic)</a>
	<a href="#">Insert Line Right</a>		<a href="#">Toggle Immediate: (Direct I/O Access)</a>
	<a href="#">Insert And Gate</a>		<a href="#">Insert Network</a>
	<a href="#">Insert Or Gate</a>		<a href="#">Delete Network</a>
	<a href="#">Insert Box</a>		<a href="#">View / Hide Symbol Information Tables for each Network</a>





-  Вставить вертикальную линию «справа - вниз»
-  Вставить вертикальную линию «справа - вверх»
-  Вставить горизонтальную линию «влево»
-  Вставить горизонтальную линию «вправо»
-  Вставить элемент «И», с дальнейшим указанием количества входов
-  Вставить элемент «ИЛИ», с дальнейшим указанием количества входов
-  Вставить инструкцию, с дальнейшим выбором типа из списка
-  Добавить вход логического элемента
-  Удалить вход логического элемента
-  Инвертировать значение входа или выхода логического элемента
-  Установить значение входа/ выхода прямого доступа
-  Вставить пустую Network ниже текущей
-  Вставить текущую Network
-  Отобразить часть символьной таблицы в цепочке программы

## Панели инструментов: команды STL



-  [Toggle Bookmark: Set or Remove Bookmark](#)
-  [Next Bookmark: Scroll Program to Next Bookmark](#)
-  [Previous Bookmark: Scroll Program to Previous Bookmark](#)
-  [Clear All Bookmarks](#)

```
//NETWORK COMMENTS
//
LD      IO.0
=      Q0.0
```

-  Установить закладку в программе
-  Перейти к следующей закладке
-  Вернуться к предыдущей закладке
-  Удалить все закладки из программы

## Создание программы

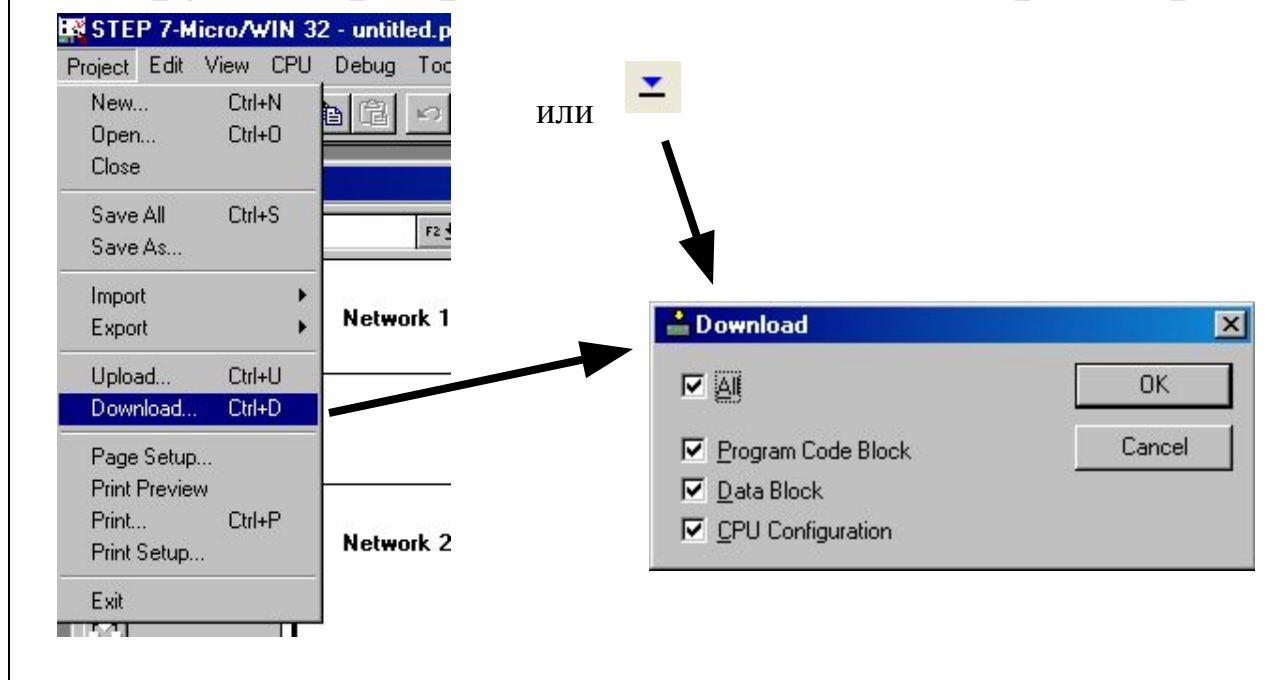


При составлении программы в ladder Editor необходимо:

- переместить курсор внутри вычислительной цепочки, в то место, куда будет добавлен элемент программы;
- выбрать группу функций на панели или в строке рисования из списка;
- выбрать соответствующую команду на панели или из списка в строке рисования;
- задать операнд команды, в открывающемся окне;
- нажать ENTER на клавиатуре, после чего курсор перейдет в следующую позицию.

Любая вычислительная цепочка должна начинаться в отведенном для нее месте, первый элемент должен быть соединен с шиной питания и заканчиваться.

## Загрузка программы в память контроллера



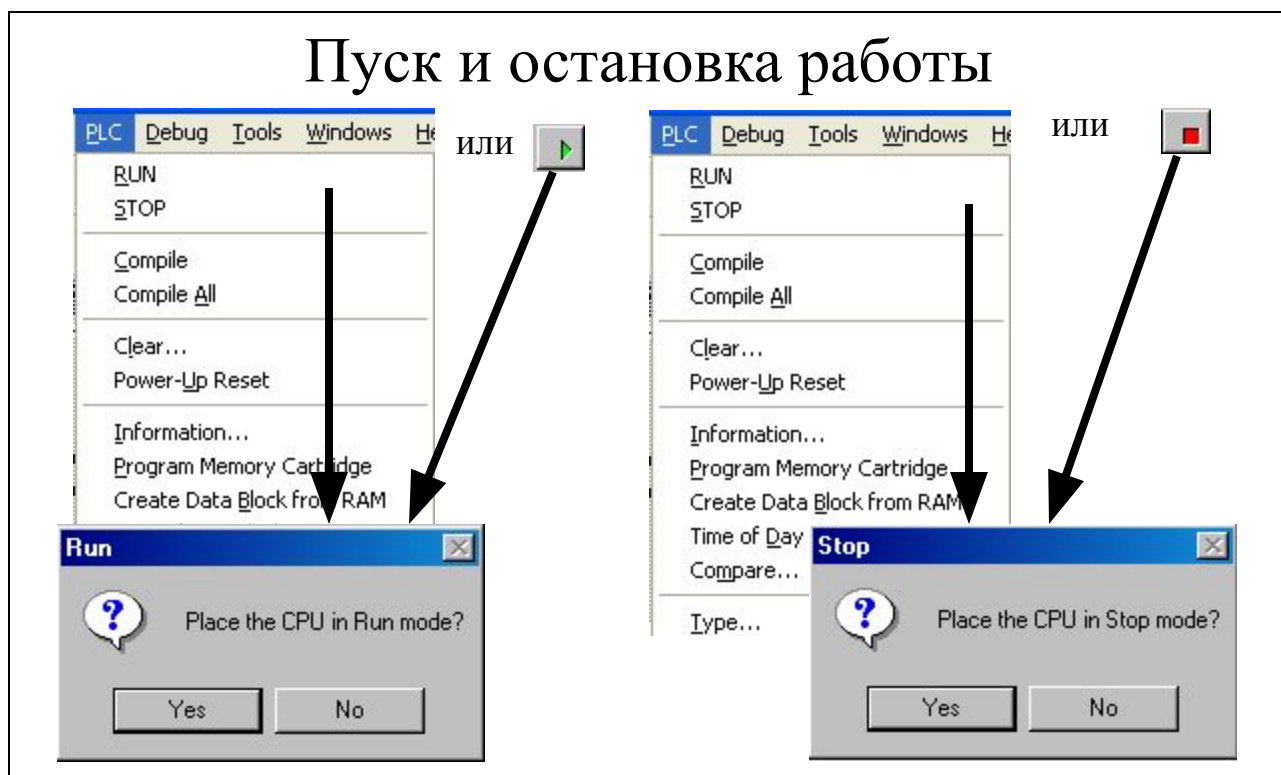
После того, как с контроллером установлена связь, составленную программу можно записывать в память контроллера, выбрав пункт Download в меню Project, либо используя соответствующий значок на панели инструментов.

В диалоге Download осуществляется выбор элементов:

- Program Code Block – программа пользователя;
- Data Block – блок данных, созданный в Data Block Editor;
- CPU Configuration – конфигурацию процессорного устройства, измененную в пункте Configure меню CPU.

Приведенные выше элементы можно выгружать из памяти контроллера в ПЭВМ, для чего предназначена команда Upload в меню Project.

## Пуск и остановка работы



CPU S7-200 имеет в своем распоряжении два режима работы :

- STOP: CPU не обрабатывает программу . В режиме STOP Вы можете загружать в CPU программу и конфигурировать CPU.
- RUN: CPU обрабатывает программу . В режиме RUN Вы не можете загружать в CPU программу , а также не можете конфигурировать CPU.

Индикация состояния на лицевой панели CPU указывает текущий режим работы . Если Вы хотите загрузить программу в программную память , то Вы должны перевести CPU в состояние STOP.

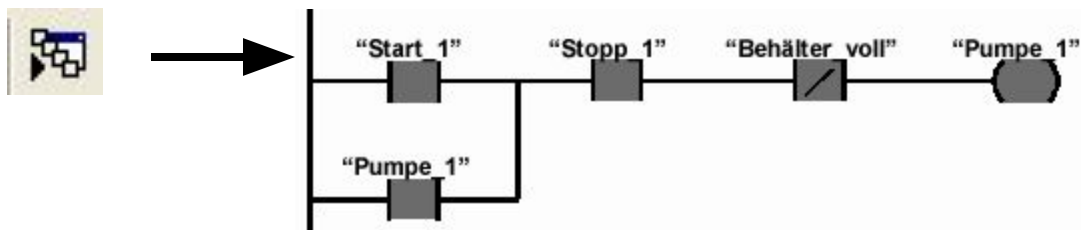
### Установка режима работы с помощью переключателя режимов работы

Под верхней крышкой на панели CPU находится микропереключатель, имеющий три положения : STOP / TERM / RUN. При установке переключателя в положение STOP – CPU будет «остановлено», в положении RUN – CPU будет обрабатывать программу. Если переключатель режимов работы находится в положении STOP или TERM и прерывается подача напряжения питания , то при восстановлении напряжения питания CPU автоматически переходит в режим STOP. Если в тот момент , когда прерывается подача напряжения питания , переключатель режимов работы находится в положении RUN, то при восстановлении напряжения питания CPU автоматически переходит снова в режим работы RUN. Если переключатель режимов работы установлен в положение TERM, то программное обеспечение (STEP 7-Micro/WIN) может управлять режимами работы CPU.

### Установка режима работы с помощью STEP 7-Micro/WIN

В меню CPU расположены пункты RUN и STOP, при выборе которых CPU переходит в соответствующий режим работы после подтверждения .

## Тестирование и наблюдение статуса программы



### Отображение статуса в программе

С помощью редактора программ STEP 7–Micro/WIN Вы можете контролировать статус программы в режиме (online), причем программа должна отображаться в форме контактного плана. Так Вы можете наблюдать состояние операций в программе при ее исполнении в CPU.