

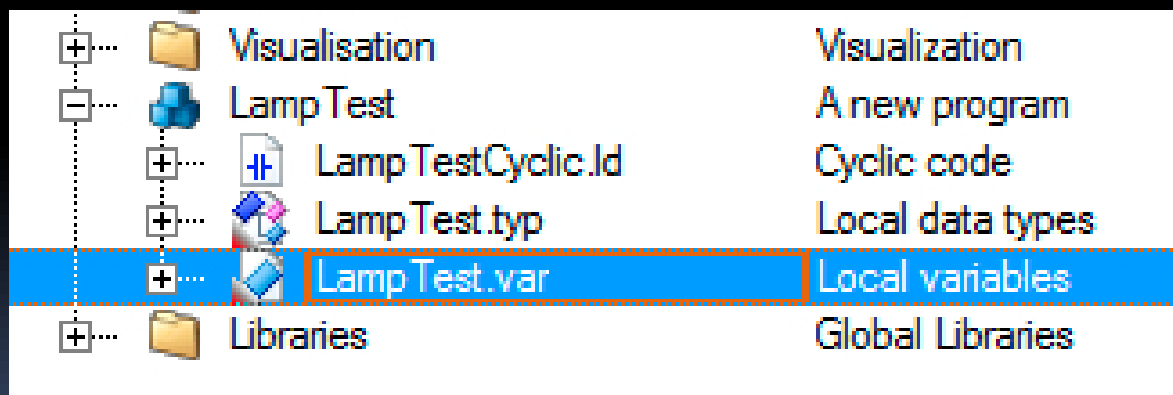


ОСНОВЫ ПРОГРАММИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ

Лекция №3

Переменные.


Переменные представляют собой участки памяти, предназначенные для хранения данных определенного приложения. В среде Automation Studio переменные объявляются в файле .var.





Типы данных.

Типы данных описывают такие свойства переменной, как например, диапазон или точность числа, содержащегося в переменной, или какие операции могут выполняться с ней.



Типы данных.

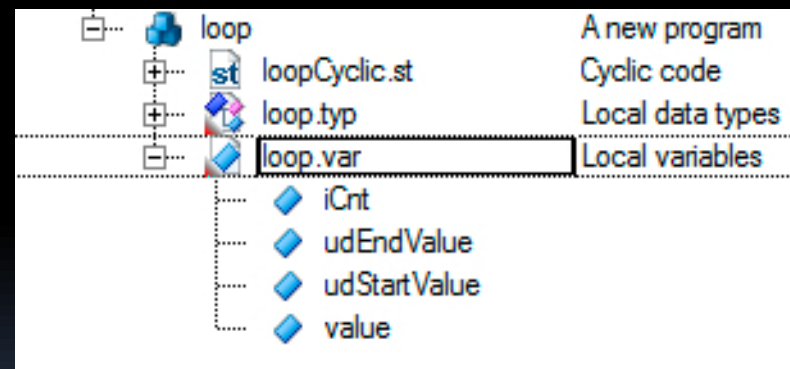
Type	Channel width (bytes)	Value range
BOOL	1	TRUE (1) and FALSE (0)
SINT	1	-128 ... +127
INT	2	-32768 ... +32767
DINT	4	-2147483648 ... +2147483647
USINT	1	0 ... 255
UINT	2	0 ... 65535
UDINT	4	0 ... 4294967295
REAL	4	-3.4E38 ... +3.4E38
LREAL	8	-1.79769313486231E308 ... +1.79769313486231E308
TIME	4	T#-24d_20h_31m_23s_648ms ... T#24d_20h_31m_23s_647ms
DATE	4	D#1970-01-01 ... D#2106-02-07
TIME_OF_DAY or TOD	4	TOD#00:00:00.000 ... TOD#23:59:59.999
DATE_AND_TIME or DT	4	DT#1970-01-01-00:00:00 ... DT#2106-02-07-06:28:15
STRING	1 per character	
BYTE	1	Bit sequence with a length of 8
WORD	2	Bit sequence with a length of 16
DWORD	4	Bit sequence with a length of 32
WSTRING	2 per character	

Глобальные и локальные переменные.

Область видимости и свойства переменной определяют ее поведение во время запуска и исполнения.

Локальные переменные:

Локальные переменные имеют локальную область видимости в пределах определенной программы и не могут использоваться в других программах.






Локальная переменная описывается файлом .var на том же уровне, что и программа.

Глобальные переменные.

Глобальные переменные отображаются на верхнем уровне панели Logical View и могут использоваться в любом месте проекта Automation Studio.

Глобальная переменная описывается на самом высоком уровне в файле Global.var. Для улучшения структурированности проекта могут быть созданы дополнительные файлы .var.

Object Name	Description
 CoffeeMachine	Default project
 Global.var	Global Variables
 gMain	

Глобальные переменные уровня пакета, которые объявлены в пакете, видны только в пределах этого пакета и всех подчиненных пакетов.

Инициализация области памяти для хранения переменной

По умолчанию во время инициализации переменным присваиваются значения «0».

При необходимости в файле объявления переменной могут быть указаны иные значения, которые должны быть присвоены при инициализации.

◆ iCnt	UDINT	<input type="checkbox"/>	0
◆ udStartValue	UDINT	<input type="checkbox"/>	344
◆ udEndValue	UDINT	<input type="checkbox"/>	120
◆ value	UDINT	<input type="checkbox"/>	0

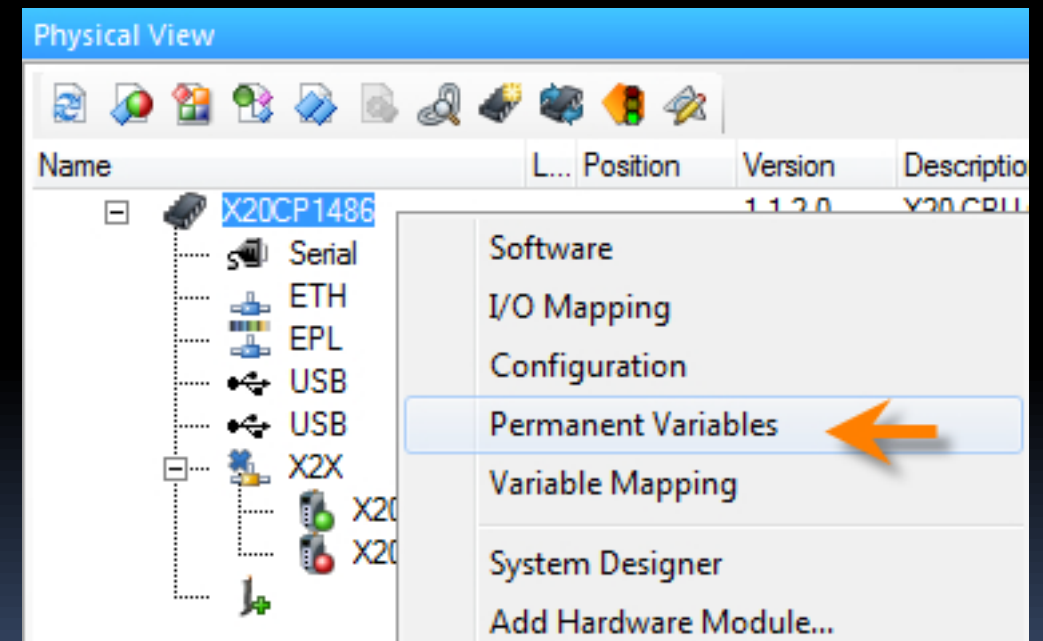
Реманентные и перманентные переменные.

Реманентные и перманентные переменные (Retain) сохраняются в энергонезависимой памяти (SRAM).

Перманентные переменные защищены от холодного перезапуска (Cold Restart).

Холодная перезагрузка происходит в следующих случаях:

- Перезагрузка после замены карты памяти CompactFlash
- Перезагрузка после очистки памяти UserROM
- Выполнение холодной перезагрузки из среды Automation Studio
- Перезагрузка в случае неисправности батареи автономного питания памяти для хранения переменных типа Retain.



Реманентные переменные.

Условия для хранения переменных в энергонезависимой (remanent) памяти:

- Наличие в целевой системе статического ОЗУ (SRAM) с автономным питанием от батареи
- Для переменных должен быть указан тип Retain

Name	Type	& Reference	Constant	Retain	Value
 OperatingHours	UINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
 ProductCounter	UDINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0

Реманентные переменные.

Примеры данных, которые могут храниться в статическом ОЗУ (SRAM):

- Счетчики рабочего времени
- Количество производственных сбоев
- Идентификационные коды изделий
- Типы счетчиков
- Прочая информация

Перезапуск происходит при возникновении следующих событий:

- Включение питания
- Загрузка измененной конфигурации в целевую систему
- Выполнение горячей перезагрузки из среды Automation Studio (аналогично включению питания)

Перманентные переменные.

Примеры:

- Счетчики рабочего времени

Особенности:

- В качестве перманентных могут использоваться только **глобальные** переменные типа Retain.
- Память, предназначенная для хранения перманентных переменных, не может быть отформатирована или перезаписана системой. За управление значениями переменных полностью отвечает пользователь.

Инициализация программы.

В каждой программе может содержаться блок инициализации.

В нем решаются следующие задачи:

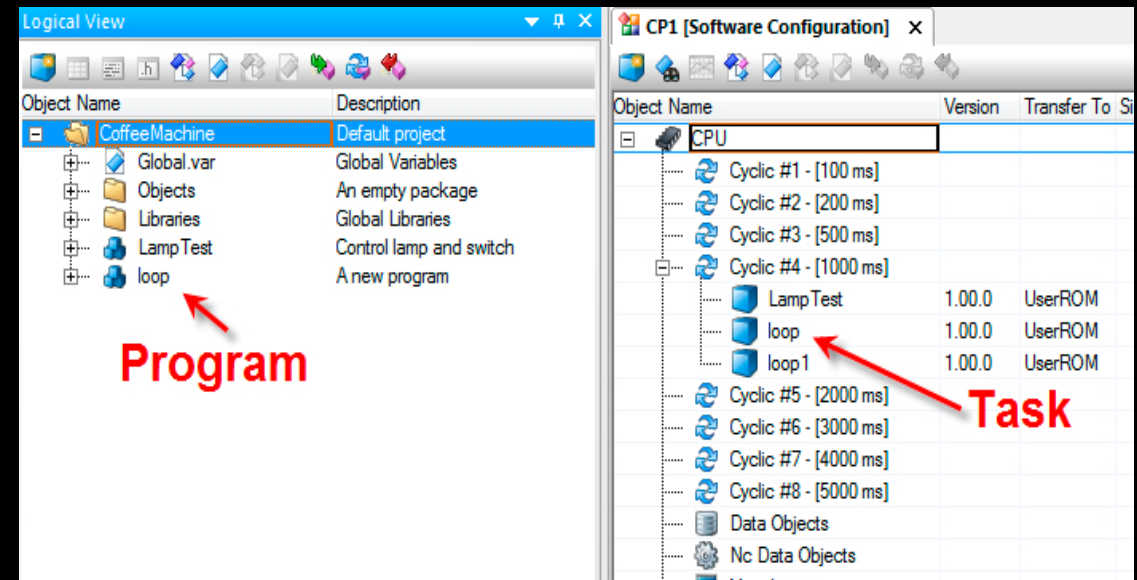
- инициализация переменных,
- считывание данных с ПЛК,
- считывание данных о текущей системной конфигурации.

Перед запуском первой циклической программы **однократно** выполняются подпрограммы инициализации всех задач в порядке, указанном в конфигурации программного обеспечения.

Циклические программы.

В конфигурации программного обеспечения программе назначается определенное время выполнения или класс задач.

Программы в конфигурации программного обеспечения называются задачами.



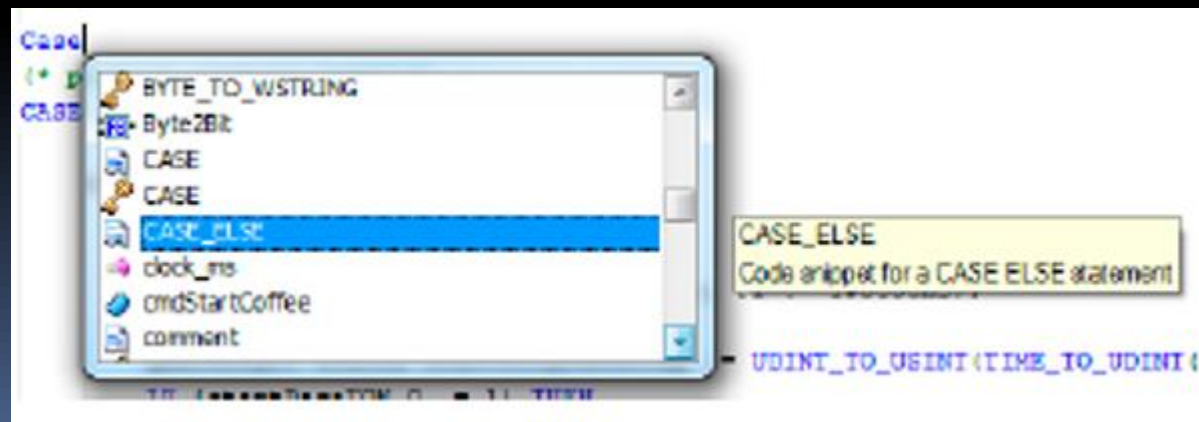
Smart Edit.

Автозаполнение

Для автоматического завершения кода используется клавиша <TAB> .

Данная функция поддерживается следующими элементами:

- Имена переменных и элементы структур
- Имя функции
- Языковые конструкции (IF THEN, CASE, FOR)



1100



The screenshot shows the 'Insert Snippet' menu with 'CASE_ELSE' selected. A tooltip indicates: 'Code snippet for a CASE ELSE statement. Shortcut: CASE_ELSE'. Below is the 'Code Snippet Manager' dialog. It shows 'ANSI C Snippets' as the language. Under 'Existing Snippets', a list includes #if, #if#elif#else, #if#else, #ifdef, #ifndef#elif#else, #ifndef#else, and _CYCLIC. The #ifdef snippet is selected, showing its details: Title: #ifdef, Shortcut: #ifdef, Description: Code snippet for #if, and Snippet Type: Expansion and Surrounds/With. The 'Code' field contains: #ifdef \$expression\$.

Insert Snippet:

- ACTION
- CASE
- CASE_ELSE**
- comment
- FOR
- FUNCTION
- FUNCTION_BLOCK
- IF

Code snippet for a CASE ELSE statement
Shortcut: CASE_ELSE

Code Snippet Manager

Language: ANSI C Snippets

Existing Snippets:

#if	#ifdef#else	_EX
#if#elif#else	#ifndef	_INI
#if#else	#ifndef#elif#else	com
#ifdef	#ifndef#else	do
#ifdef#elif#else	_CYCLIC	else

Header

Title: #ifdef

Shortcut: #ifdef

Description: Code snippet for #if

Snippet Type: ☒ Expansion ☒ Surrounds/With

Literals:

ID	Default
expression	expression

Code:

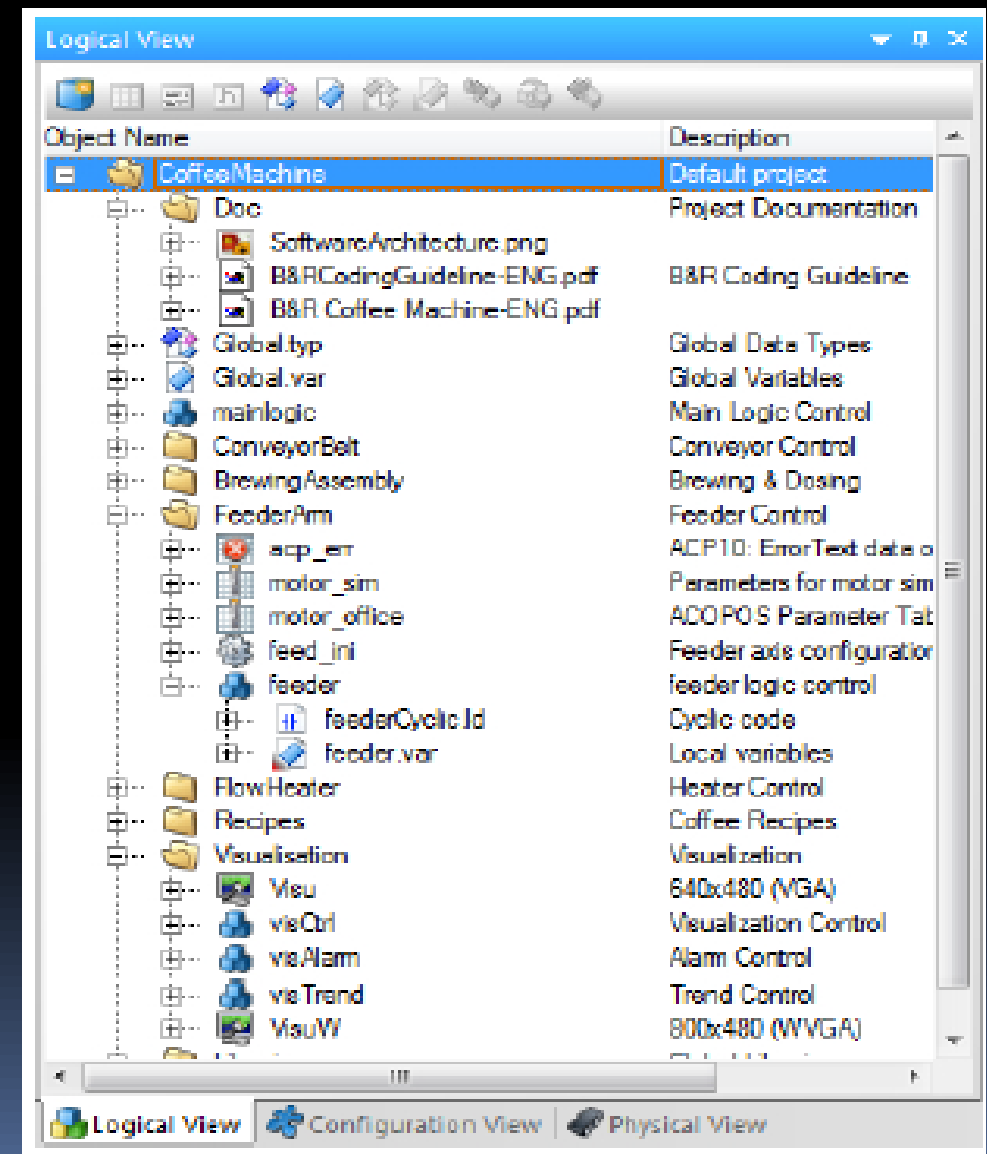
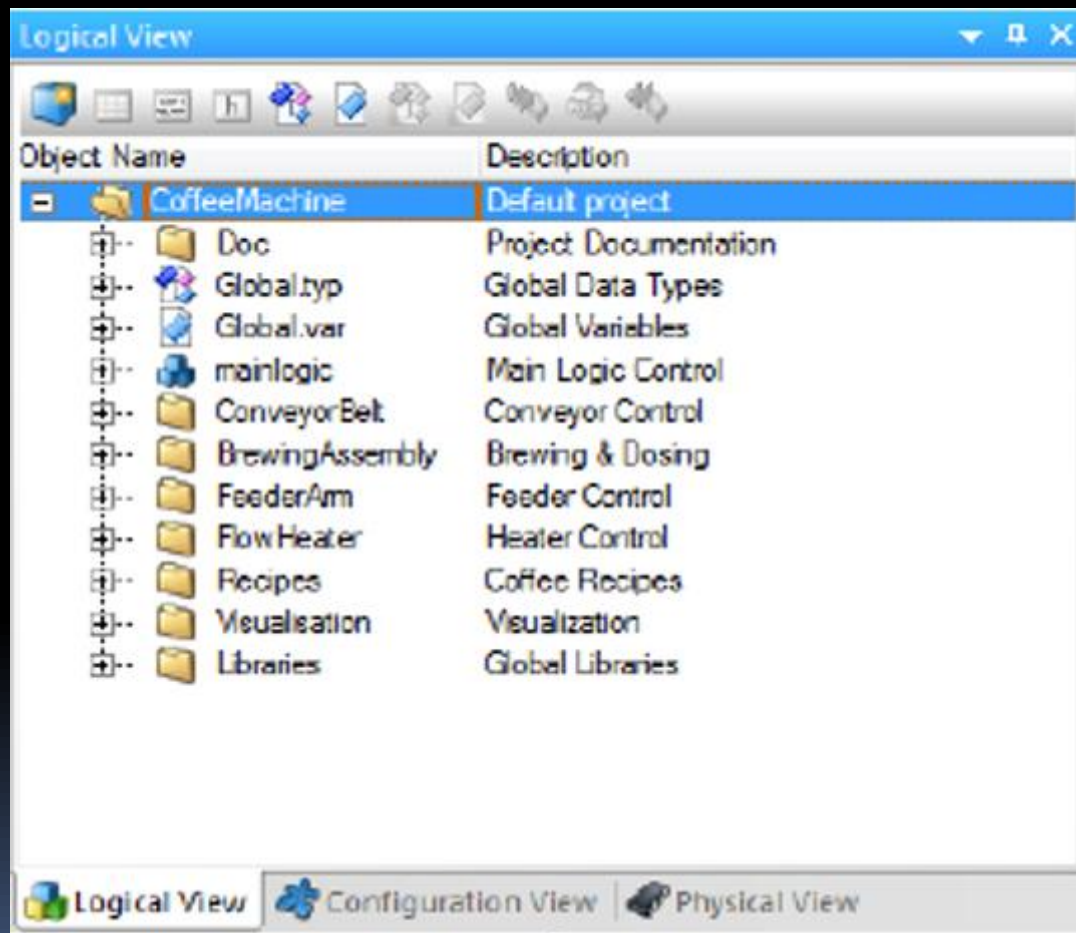
```
#ifdef $expression$
```

Добавление фрагментов кода

Горячие клавиши.

Функция	Сочетание клавиш
Заккрытие фрагментов кода	<TAB>
Автозаполнение имен переменных и функций	<CTRL> + <SPACE>
Активация окна выбора фрагмента кода	<CTRL> + <q>, <k>
Включение и отключение выделения всех парных скобок	<CTRL> + <q>, <l>
Включение и отключение выделения текущих парных скобок	<CTRL> + <q>, <m>
Переход к объявлению переменной	<CTRL> + <d>
Переход к указанию типа данных для переменной	<CTRL> + <t>
Переход к реализации функции или функционального блока	<CTRL> + <i>

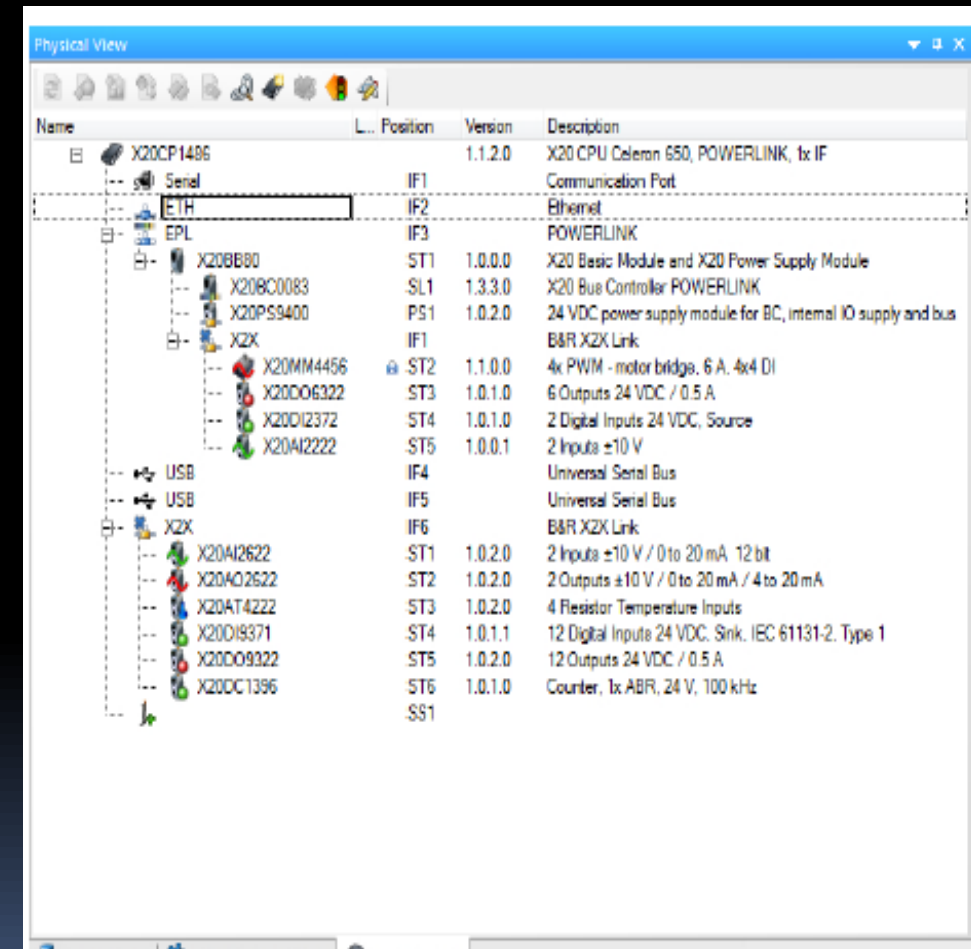
Automation studio. Logical view.



Automation studio. Physical view.

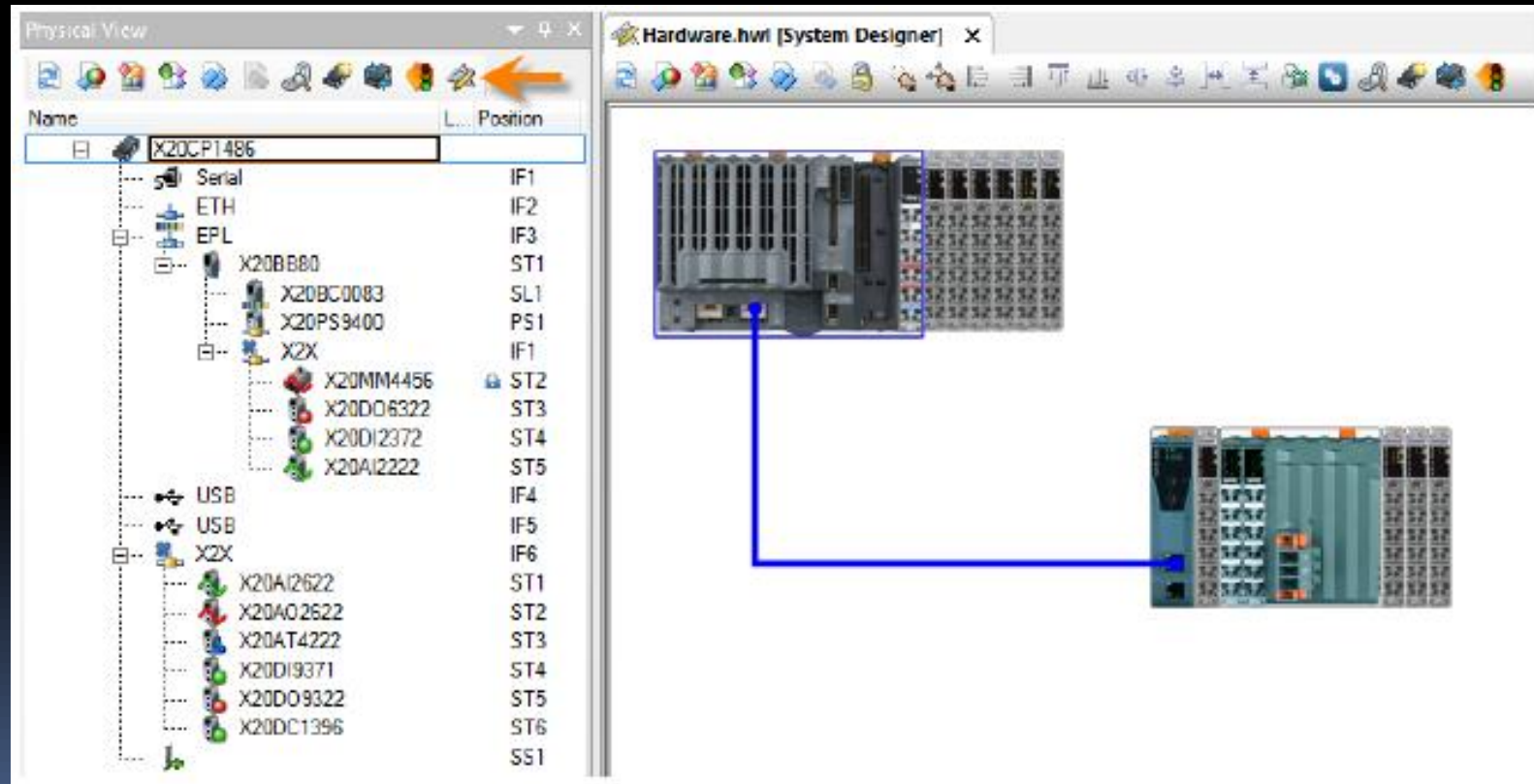
Решаемые задачи:

- Добавление и конфигурирование модулей ввода/вывода
- Назначение переменным точек ввода/вывода данных
- Конфигурирование модули полевой шины и интерфейсов в сети
- Добавление и конфигурирование аппаратных модулей (например, терминалов, устройств полевой шины)
- Просмотр конфигурации программного обеспечения



Automation studio. System designer.

Компонент System Designer обеспечивает графическое изображение конфигурации в Physical view.

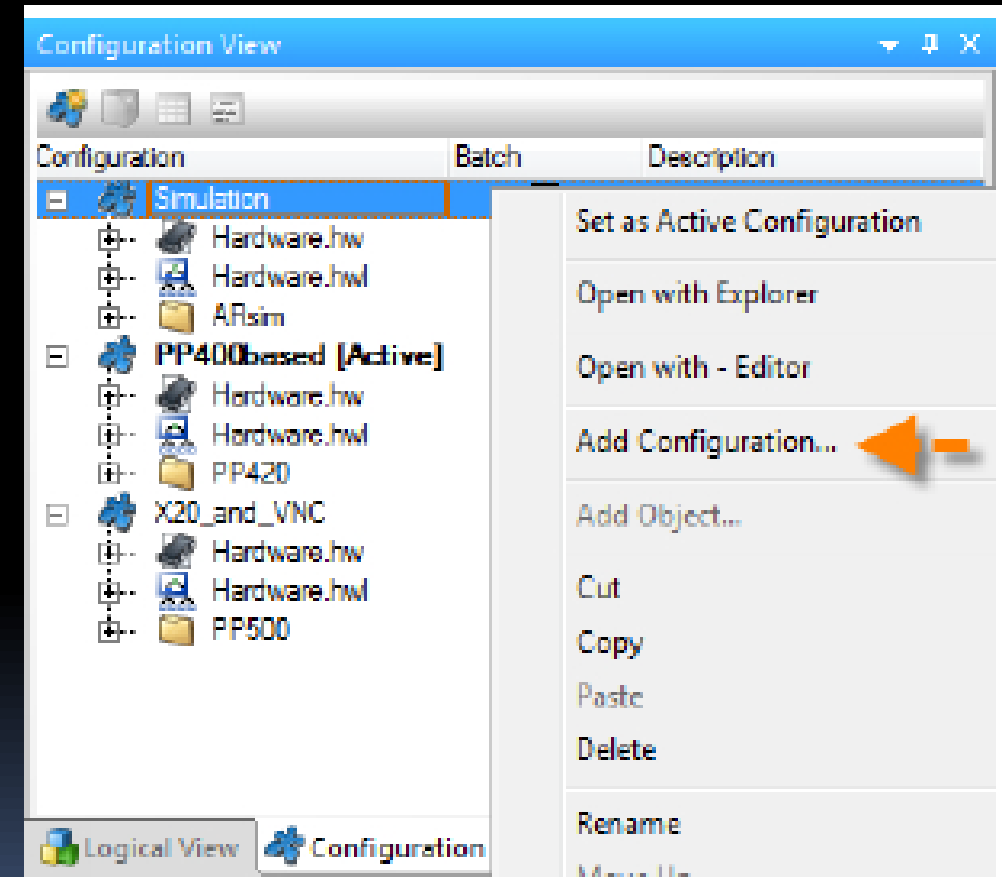


Automation studio. Configuration view.

В окне **Configuration View**, осуществляется переключение и управление различными типами систем.

Конфигурация включает в себя аппаратные и программные компоненты.

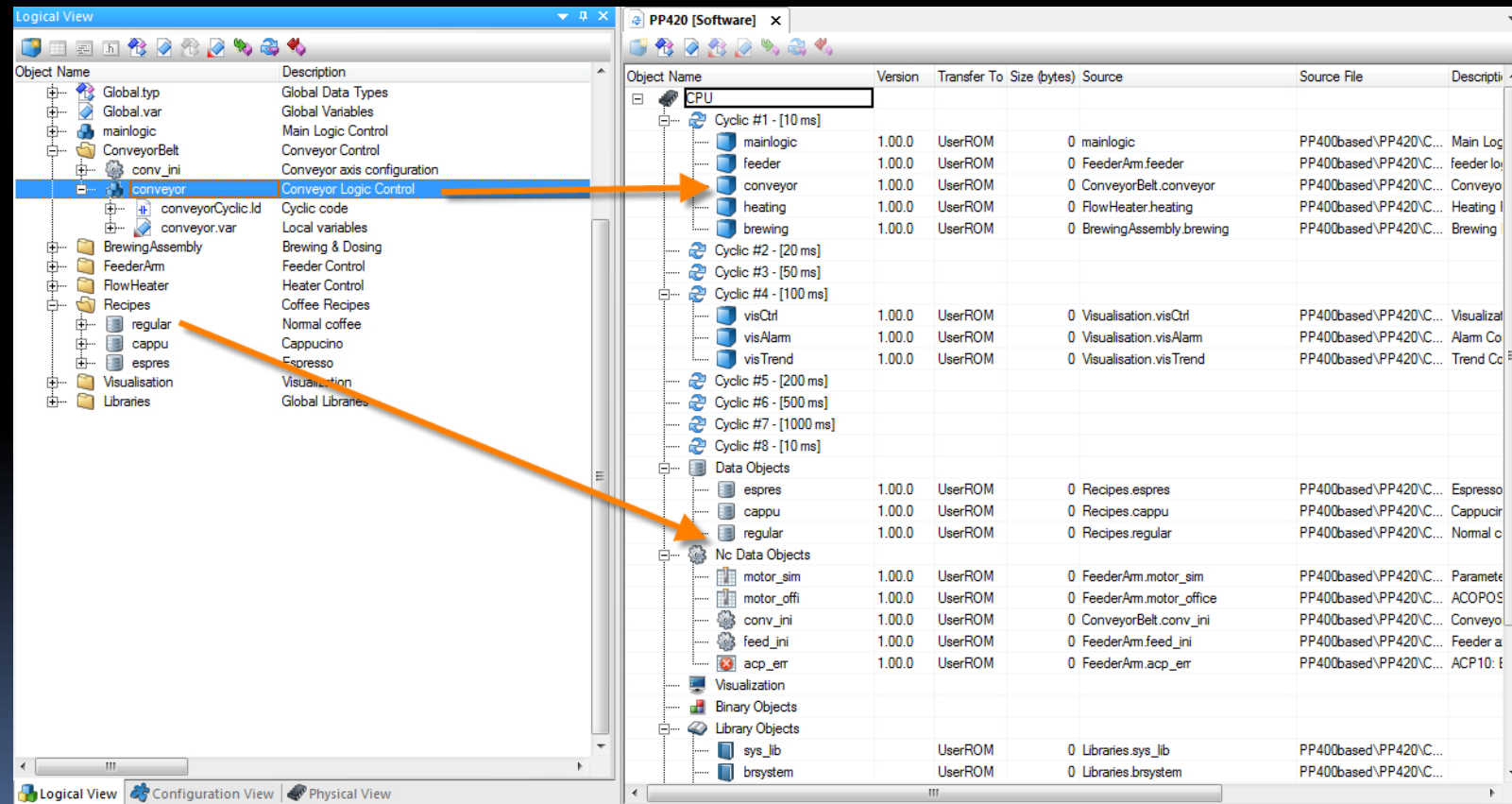
Только одна конфигурация может быть активной (**Active**) в каждый момент времени.



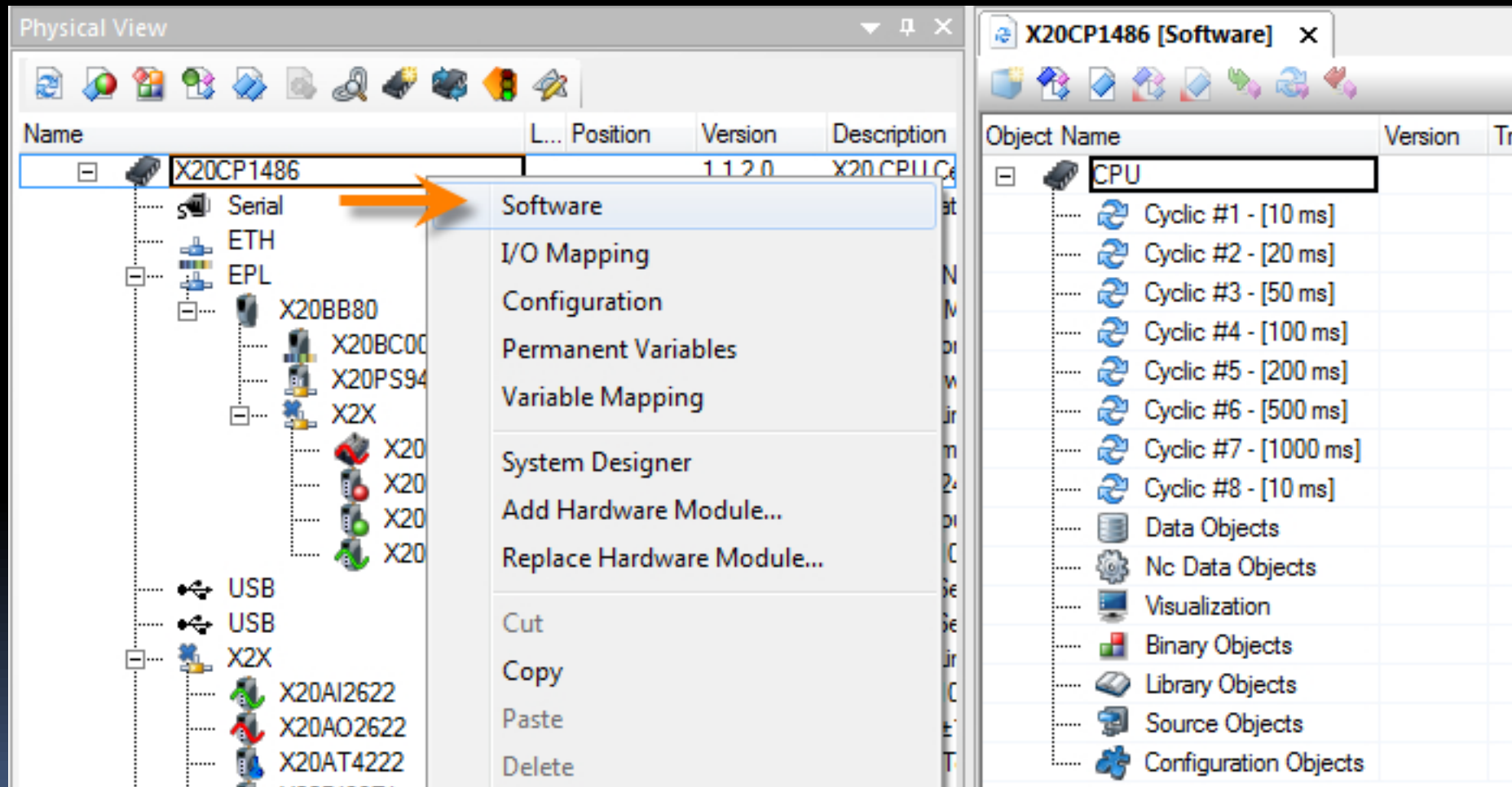
Automation studio. Software Configuration.

Два способа назначить элемент ПО соответствующей конфигурации:

- "Автоматическое назначение при создании программы"
- "Добавление имеющихся программ вручную"



Доступ к Software configuration.



Добавление имеющихся программ вручную.

The screenshot displays two windows from a software development environment. The left window, titled "Logical View", shows a hierarchical tree of objects. The right window, titled "PP420 [Software]", shows a table of objects with their properties.

Logical View Table:

Object Name	Description
Global.type	Global Data Types
Global.var	Global Variables
mainlogic	Main Logic Control
ConveyorBelt	Conveyor Control
conv_ini	Conveyor axis configuration
conveyor	Conveyor Logic Control
conveyorCyclic.ld	Cyclic code
conveyor.var	Local variables
BrewingAssembly	Brewing & Dosing
FeederArm	Feeder Control
FlowHeater	Heater Control
Recipes	Coffee Recipes
regular	Normal coffee
cappu	Cappuccino
espres	Espresso
Visualisation	Visualization
Libraries	Global Libraries

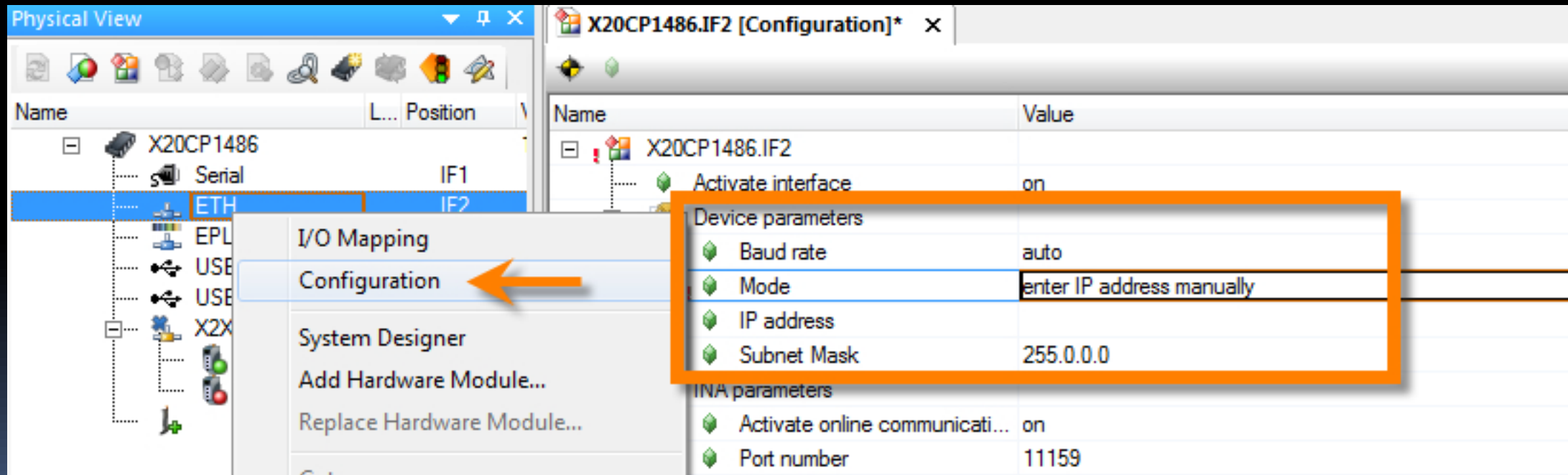
PP420 [Software] Table:

Object Name	Version	Transfer To	Size (bytes)	Source	Source File	Description
CPU						
Cyclic #1 - [10 ms]						
mainlogic	1.00.0	UserROM	0	mainlogic	PP400based\PP420\C...	Main Log
feeder	1.00.0	UserROM	0	FeederArm.feeder	PP400based\PP420\C...	feeder lo
conveyor	1.00.0	UserROM	0	ConveyorBelt.conveyor	PP400based\PP420\C...	Conveyo
heating	1.00.0	UserROM	0	FlowHeater.heating	PP400based\PP420\C...	Heating I
brewing	1.00.0	UserROM	0	BrewingAssembly.brewing	PP400based\PP420\C...	Brewing
Cyclic #2 - [20 ms]						
Cyclic #3 - [50 ms]						
Cyclic #4 - [100 ms]						
visCtrl	1.00.0	UserROM	0	Visualisation.visCtrl	PP400based\PP420\C...	Visualizat
visAlarm	1.00.0	UserROM	0	Visualisation.visAlarm	PP400based\PP420\C...	Alarm Co
visTrend	1.00.0	UserROM	0	Visualisation.visTrend	PP400based\PP420\C...	Trend Co
Cyclic #5 - [200 ms]						
Cyclic #6 - [500 ms]						
Cyclic #7 - [1000 ms]						
Cyclic #8 - [10 ms]						
Data Objects						
espres	1.00.0	UserROM	0	Recipes.espres	PP400based\PP420\C...	Espresso
cappu	1.00.0	UserROM	0	Recipes.cappu	PP400based\PP420\C...	Cappuccir
regular	1.00.0	UserROM	0	Recipes.regular	PP400based\PP420\C...	Normal c
Nc Data Objects						
motor_sim	1.00.0	UserROM	0	FeederArm.motor_sim	PP400based\PP420\C...	Paramete
motor_offi	1.00.0	UserROM	0	FeederArm.motor_office	PP400based\PP420\C...	ACOPOS
conv_ini	1.00.0	UserROM	0	ConveyorBelt.conv_ini	PP400based\PP420\C...	Conveyo
feed_ini	1.00.0	UserROM	0	FeederArm.feed_ini	PP400based\PP420\C...	Feeder a
acp_err	1.00.0	UserROM	0	FeederArm.acp_err	PP400based\PP420\C...	ACP10: E
Visualization						
Binary Objects						
Library Objects						
sys_lib		UserROM	0	Libraries.sys_lib	PP400based\PP420\C...	
brsystem		UserROM	0	Libraries.brssystem	PP400based\PP420\C...	

Настройка сетевых интерфейсов.

Условия для подключения к контроллеру по сети:

- ПК и контроллер находятся в одной и той же сети
- Известны разрешенные и назначенные IP адреса






Установка соединения.

Два способа установки соединения:

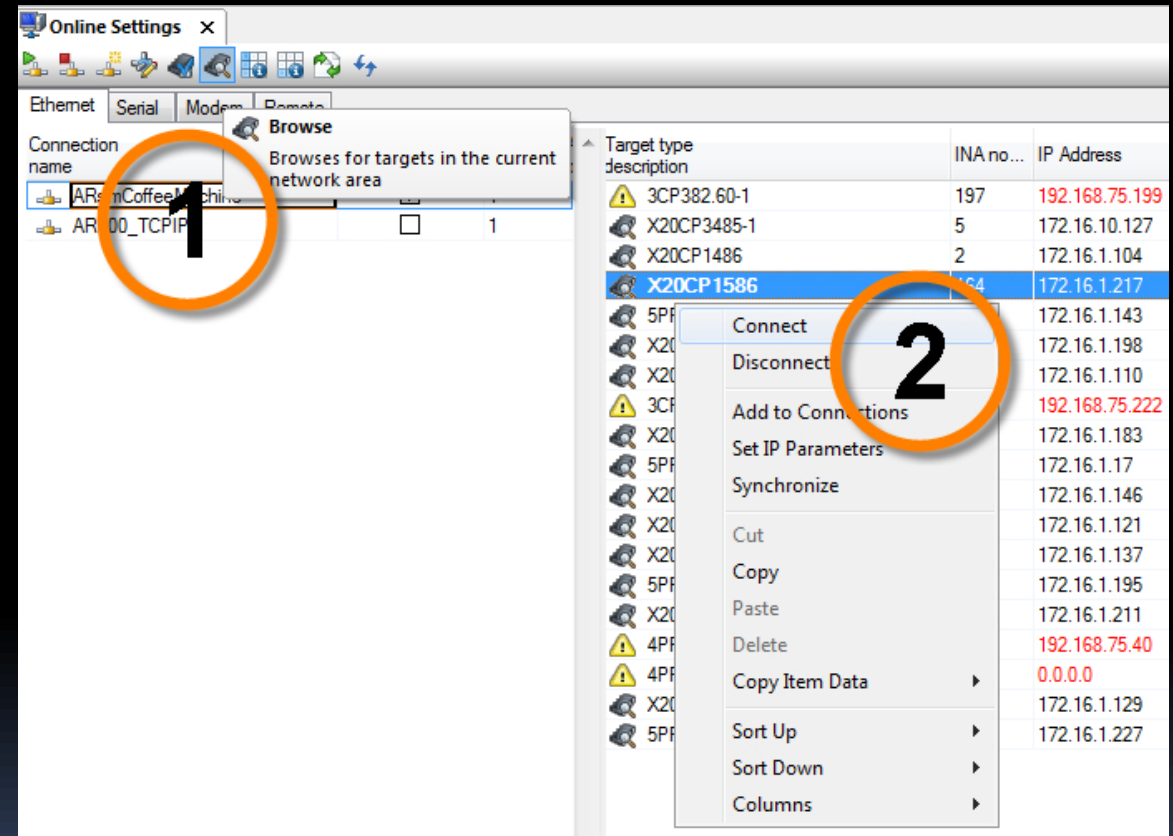
- Поиск устройств в сети
- Ручная настройка



Для настройки параметров подключения контроллера необходимо выбрать из главного меню **<Online - Settings>**.

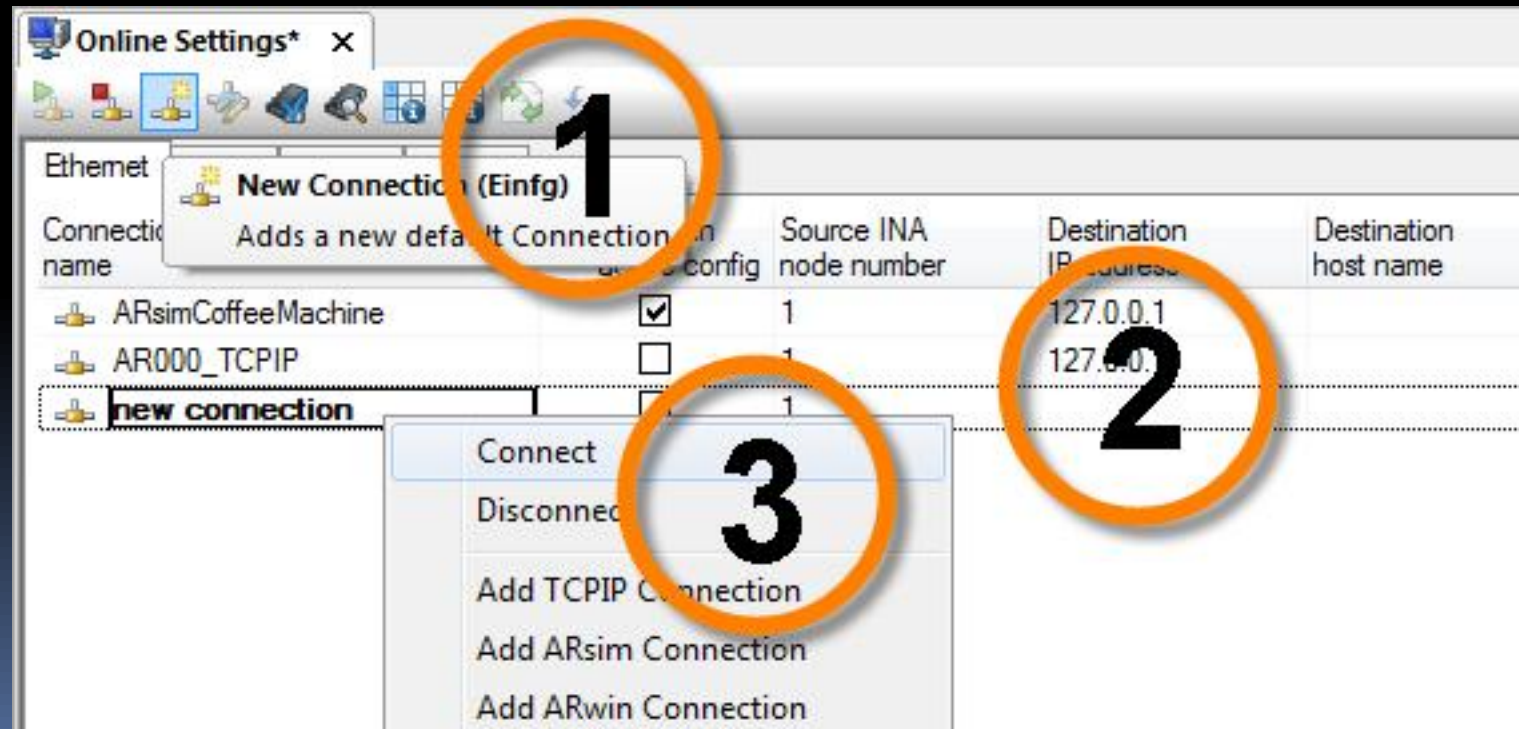
Поиск сетевых устройств.

1. Поиск сетевых устройств, выполняется нажатием на пиктограмму **Browse** на панели инструментов, результаты выводятся в правой части рабочей области.
2. Из контекстного меню выбирается **Connect**



Ручная настройка соединения.

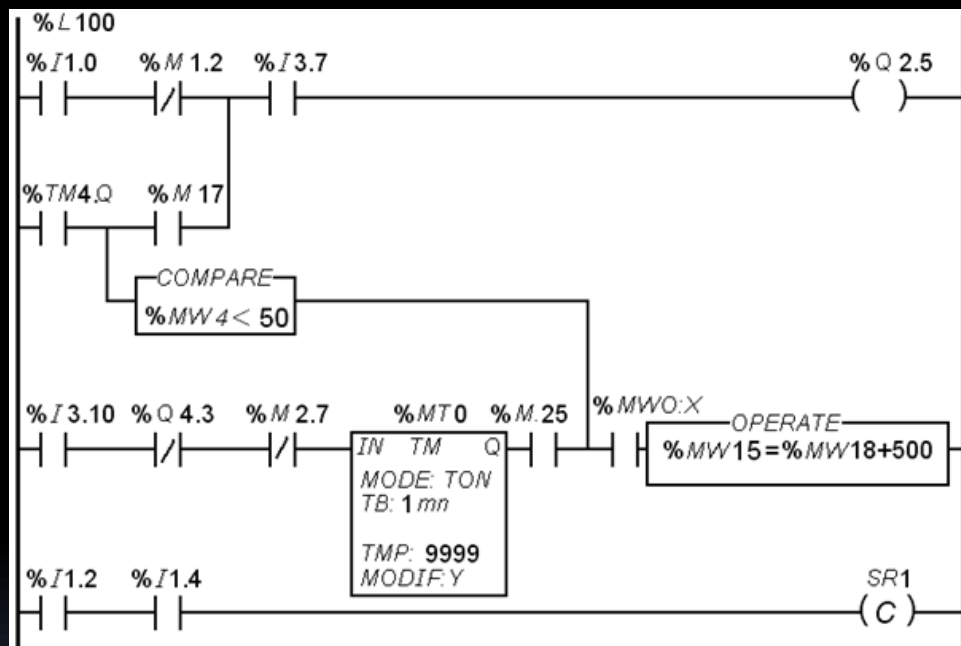
1. Добавьте новое соединение, нажав на пиктограмму на панели инструментов
2. Введите параметры соединения (IP-адрес, номер узла)
3. Активируйте онлайн соединение в контекстном меню нового соединения



Языки программирования.

Язык программирования	Примечания
Язык лестничных диаграмм (LD)	Графический
Язык диаграмм функциональных блоков (FBD)	Графический
Язык непрерывных функциональных диаграмм (CFC)	Графический
Язык последовательных функциональных диаграмм (SFC)	Графический и текстовый
Язык списка инструкций (IL)	Текстовый
Язык структурированного текста (ST)	Текстовый
ANSI C и C++	Текстовый

Языки программирования.

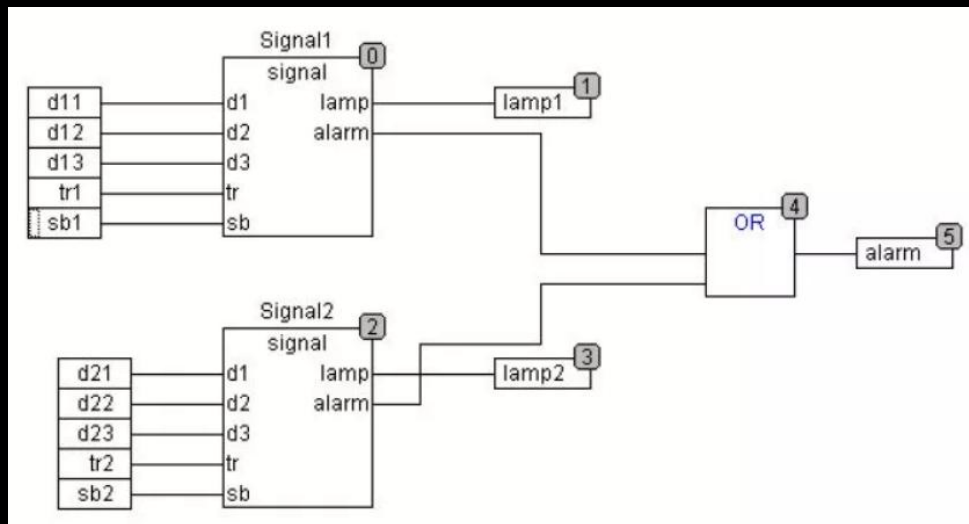


Ladder diagram

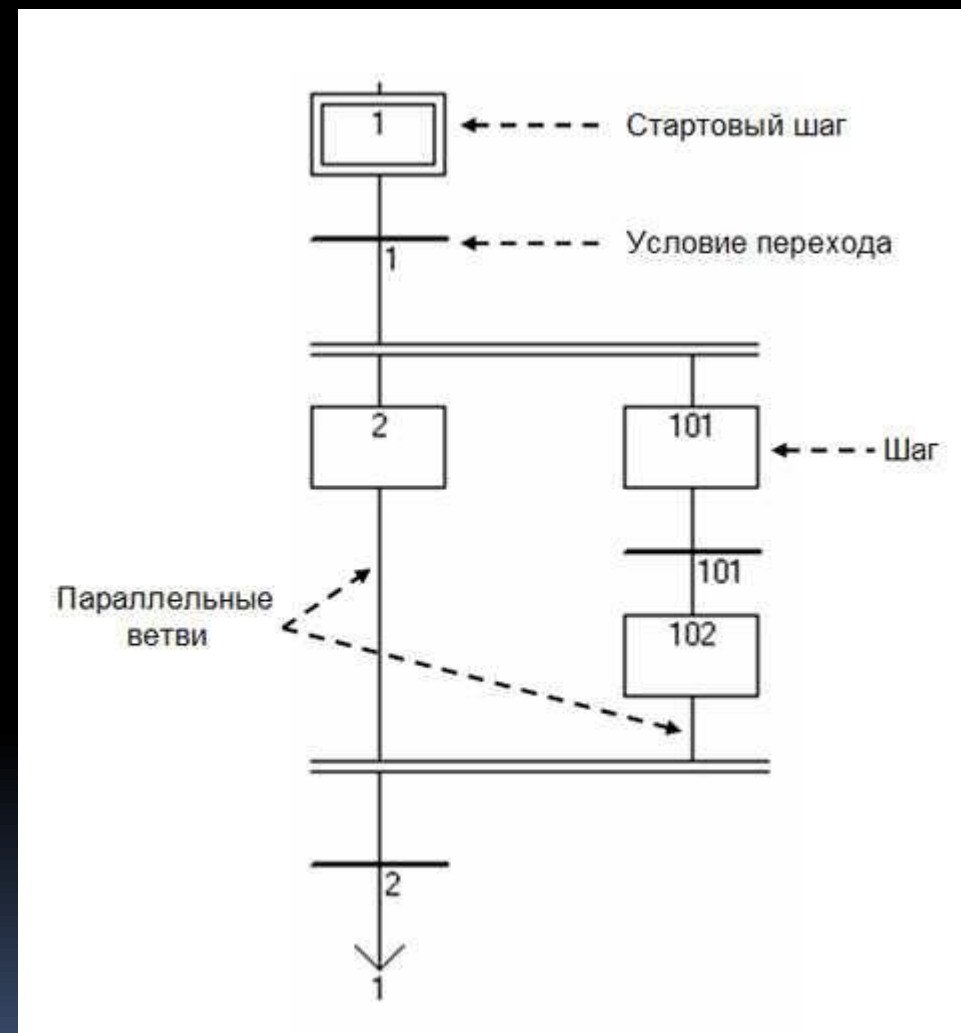


Function Block Diagram.

Языки программирования.



(Continuous Flow Chart)



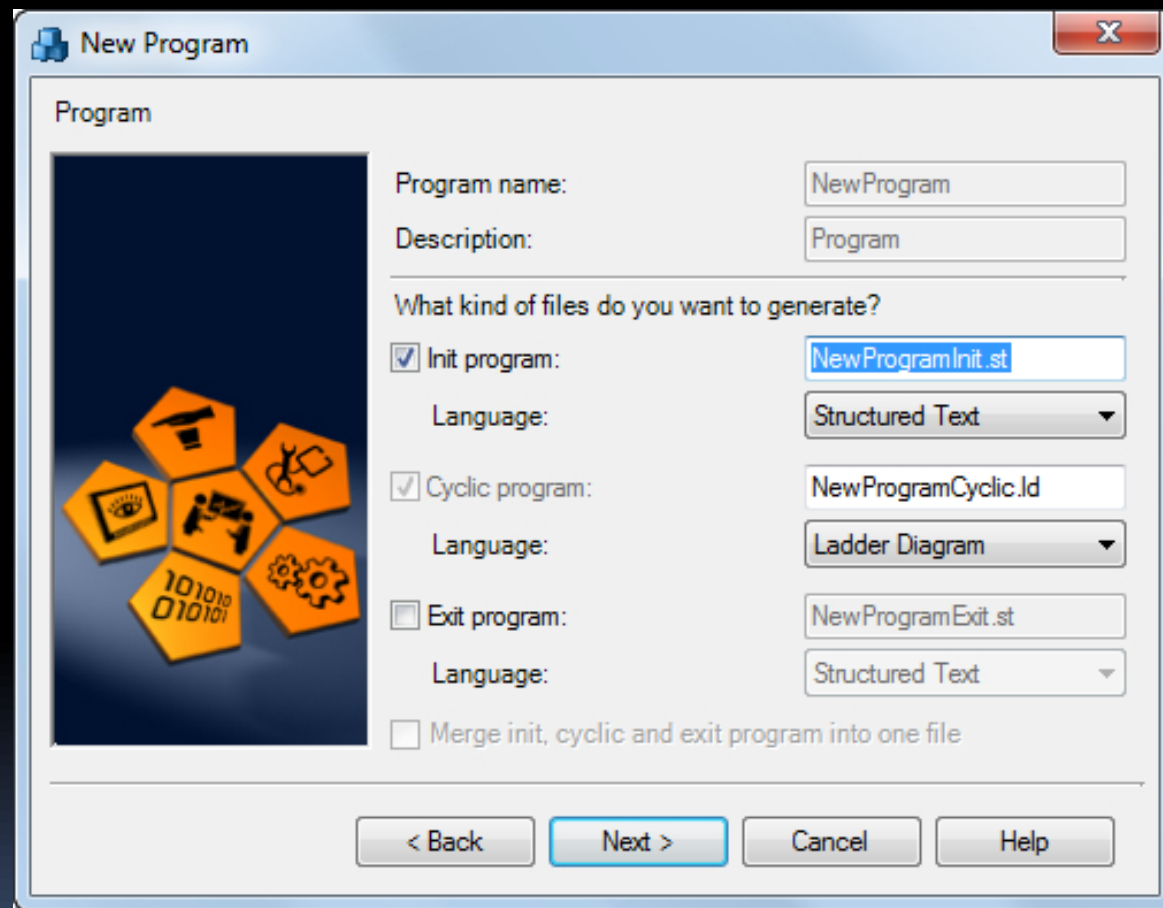
Sequential Function Chart

Части программы.

Инициализация
задачи

Циклическая часть

Завершение задачи



New Program

Program

Program name: NewProgram

Description: Program

What kind of files do you want to generate?

☒ Init program: NewProgramInit.st
Language: Structured Text

☒ Cyclic program: NewProgramCyclic.ld
Language: Ladder Diagram

☐ Exit program: NewProgramExit.st
Language: Structured Text

☐ Merge init, cyclic and exit program into one file

< Back Next > Cancel Help