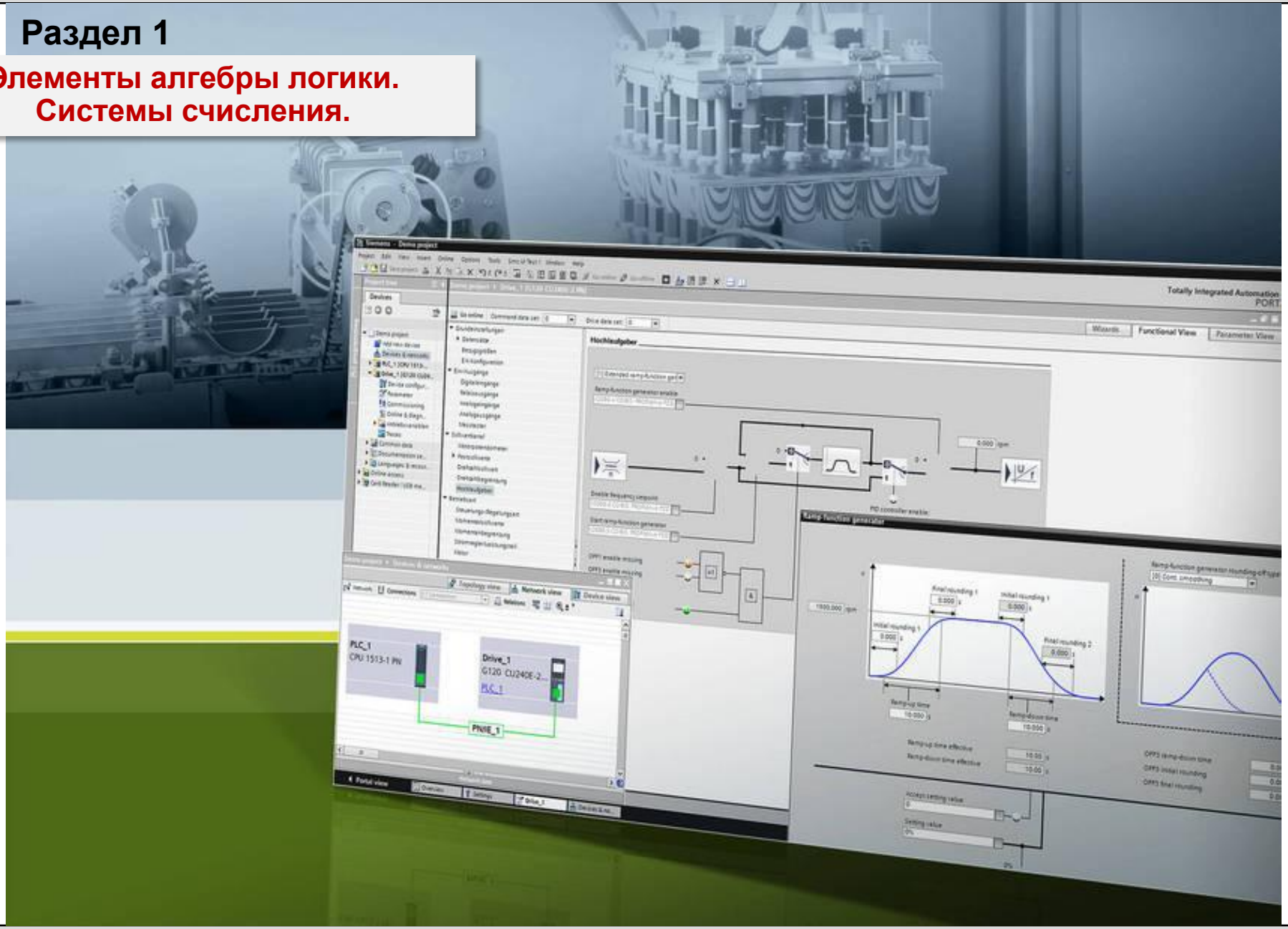
Программируемые логические контроллеры SIMATIC. Инженерная среда SIMATIC TIA-portal. Step-7 V12 Pro.

- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 1

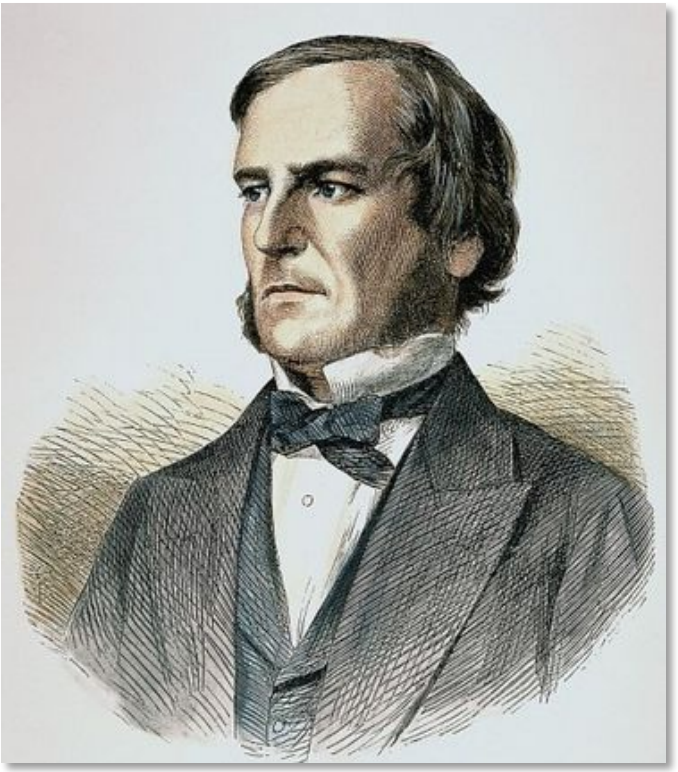
Элементы алгебры логики. Системы счисления.



Коротко о математическом аппарате «Алгебра логики»

В основу инженерного построения дискретных цепей, использующихся в электротехнике, в электронике, в программно реализуемых алгоритмах управления, положен математический аппарат двоичной логики – алгебры логики.

Создатель этого математического аппарата, который еще называют алгеброй событий или высказываний, английский математик Джордж Буль показал, как математическим путем из какого-то исходного количества логических высказываний можно вывести логическое заключение, являющееся функцией этих высказываний. Аппарат алгебры логики предполагает, что область существования переменных, как функций, так и аргументов ограничена двумя значениями:



- «истинно» (true), которому в уравнениях соответствует знак «1» - логическая единица
- «ложно» (false), которому в уравнениях соответствует знак «0» - логический ноль .

Из полного набора предложенных Булем функций (их 16) рассмотрим те, которые применяются в инженерной практике чаще всего.

Функция логического умножения – «И»

«Релейное» представление

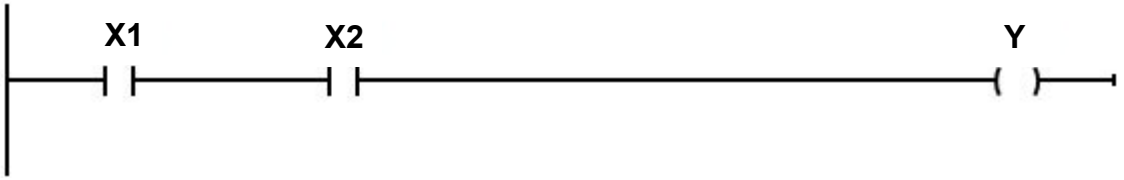


Таблица истинности

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Для того, чтобы переменная Y =1, необходимо, чтобы и X1, и X2 были = 1.

Эта функция носит название «логическое умножение» (коньюнкция), функция «И».

Записывается как Y=X1&X2

Вид построчных инструкций

```
A   X1
A   X2
=   Y
```

Команда **A**: AND logic operation

Функциональная схема

Функция логического сложения – «ИЛИ»

«Релейное» представление

Таблица истинности

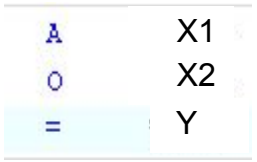
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Для того, чтобы переменная Y =1, необходимо, чтобы или X1=1, или X2 =1, или обе вместе.

Эта функция носит название «логическое сложение» (дизъюнкция), функция «ИЛИ».

Записывается как Y=X1||X2

Вид построчных инструкций



Команда **O**: OR logic operation

Функциональная схема

Функция логического отрицания – «НЕ» (инверсия)

«Релейное» представление

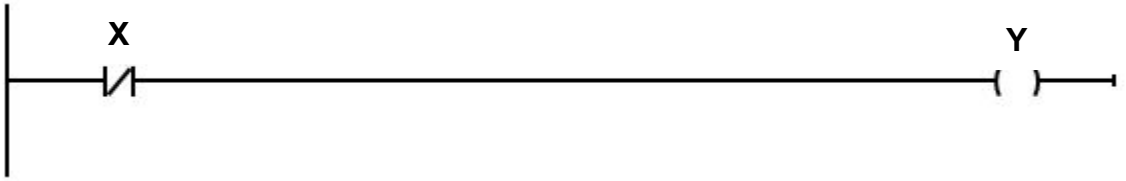


Таблица истинности

X	Y
0	1
1	0

Такая цепочка в терминах алгебры логики описывается как «высказывание Y является истинным, когда исходное высказывание X ложно и наоборот». Y есть не X, или функция есть *инверсия* аргумента.

В отличие от предыдущих функция унитарная, записывается как $Y=\overline{X}$

Вид построчных инструкций

```
AN   X
=    Y
```

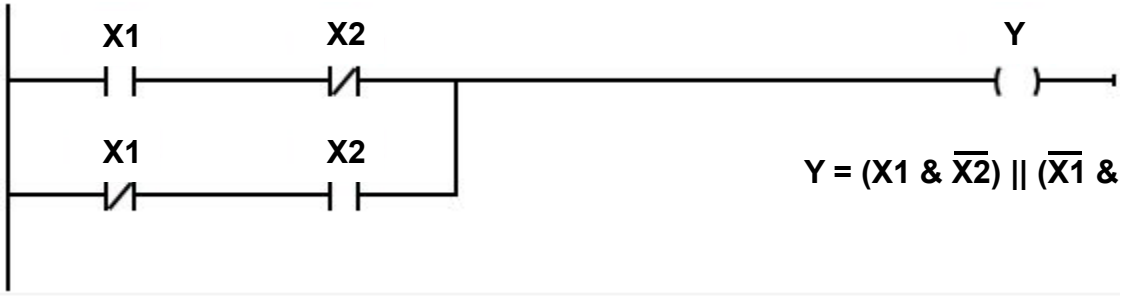
Команда **AN**: AND NOT
logic operation

Функциональная схема

Функция «Исключающее ИЛИ»

Расмотренные функции «И» «ИЛИ» «НЕ» представляют собой полный функциональный набор, то есть их сочетанием можно представить логическую цепь любой степени сложности. Кроме них распространена еще одна логическая функция – «Исключающее ИЛИ» (альтернатива). Для того, чтобы переменная Y =1, необходимо, чтобы или X1=1, или X2 =1, но не обе вместе.

«Релейное» представление



$Y = (X1 \;\&\; \overline{X2}) \;||\; (\overline{X1} \;\&\; X2)$

Таблица истинности

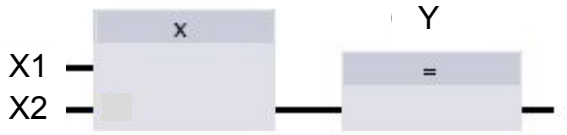
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Вид построчных инструкций

```
X   X1
X   X2
=   Y
```

Команда **X**: EXCLUSIVE OR logic operation

Функциональная схема



Позиционная система счисления

Позиционной называется система счисления, в которой число представлено суммой произведений степени основания на число, представленное цифрой в текущем разряде (позиции).
В общем виде число в такой системе счисления записывается так:

$$[A \times N^{к}]\ldots+\ldots[A \times N^{3}] + [B \times N^{2}] + [C \times N^{1}] + [D \times N^{0}]$$

Где A, B, C, D – цифры, N – целое основание системы счисления, (0,1,2,3…к) – степени основания.
(Здесь следует помнить, что любое число, отличное от нуля, в нулевой степени = 1).

Привычная нам десятичная система счисления является позиционной с основанием 10 и оперирует с десятью знаками (цифрами) – от 0 до 9.
Например:
возьмем любое число, допустим, число 9345 и разложим его на составляющие. Мы увидим, что оно является суммой следующих чисел:

9 тысяч + 3 сотни + 4 десятка + 5 единиц

или: $9 \times 10^{3} + 3 \times 10^{2} + 4 \times 10^{1} + 5 \times 10^{0}$

Мы видим, что от местоположения цифры в числе (позиции или разряда) зависит ее «вес», который определяется степенью основания системы счисления – в данном случае это возрастающие справа налево степени числа 10.

Представление чисел в двоичной и шестнадцатиричной системах

• Цифровые системы работают в множестве только двух значений – «0» (выключено) и «1» (включено). Соответственно, все математические операции в них могут быть произведены только в двоичной системе счисления, в которой построение числа основано на использовании двух цифр – «0» и «1», а «вес» позиции определяется степенью числа 2.

Например, возьмем произвольное двоичное число $(1011)_2$ и разложим на составляющие по правилам позиционной системы:

$$1x\ 2^3 + 0x\ 2^2 + 1x\ 2^1 + 1x\ 2^0$$

При прочтении приведенного двоичного числа в десятичной форме мы имеем:

$$8 + 0 + 2 + 1 = 11,\ \text{т.е}\ (1011)_2 = 11_{10}$$

• Для представлении чисел в цифровой технике часто используется **шестнадцатиричная система** (**hex**). Она удобна тем, что число 16 также является степенью 2 и переход к прочтению чисел в этих системах достаточно прост. Прочтение цифр в шестнадцатиричной системе следующее:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Легко посчитать по ряду, что $A_{16} = 10_{10}$ и, соответственно, $F_{16} = 15_{10}$

Для написания любой шестнадцатиричной цифры в символах «0» и «1» достаточно четырех двоичных разрядов. Например, $A_{16} = (1010)_2$, $F_{16} = (1111)_2$

Представим десятичное число 1250 в шестнадцатиричной и двоичной форме:

$$1250_{10} = 04E2_{16} = (0000\ 0100\ 1110\ 0010)_2$$

Видно, что шестнадцатиричное число легко «развернуть» в двоичную форму поразрядно сответствующими тетрадами двоичного представления, а по исходному написанию оно короче.

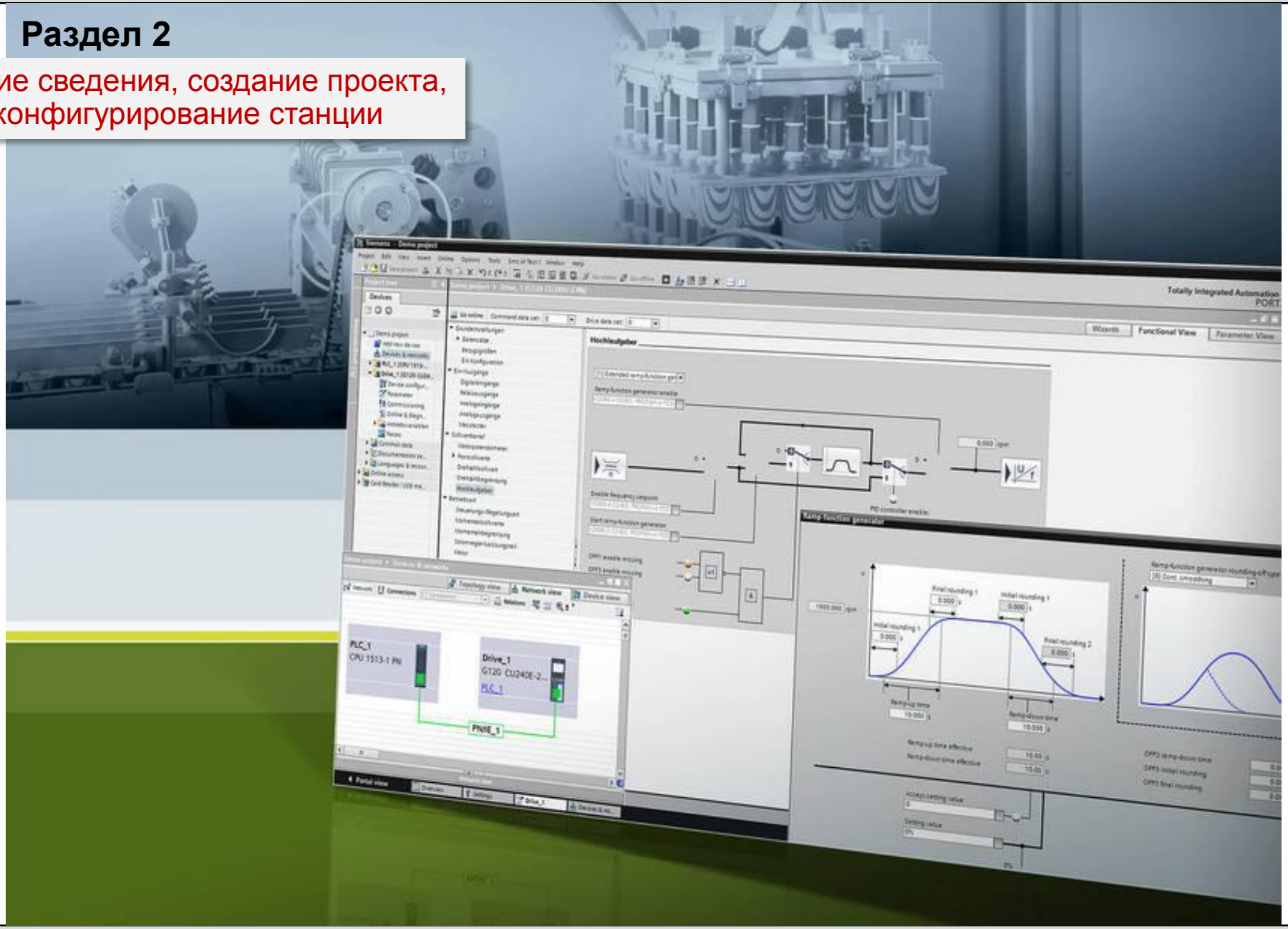
Конец раздела 1. Окно навигации

Основы алгебры логики

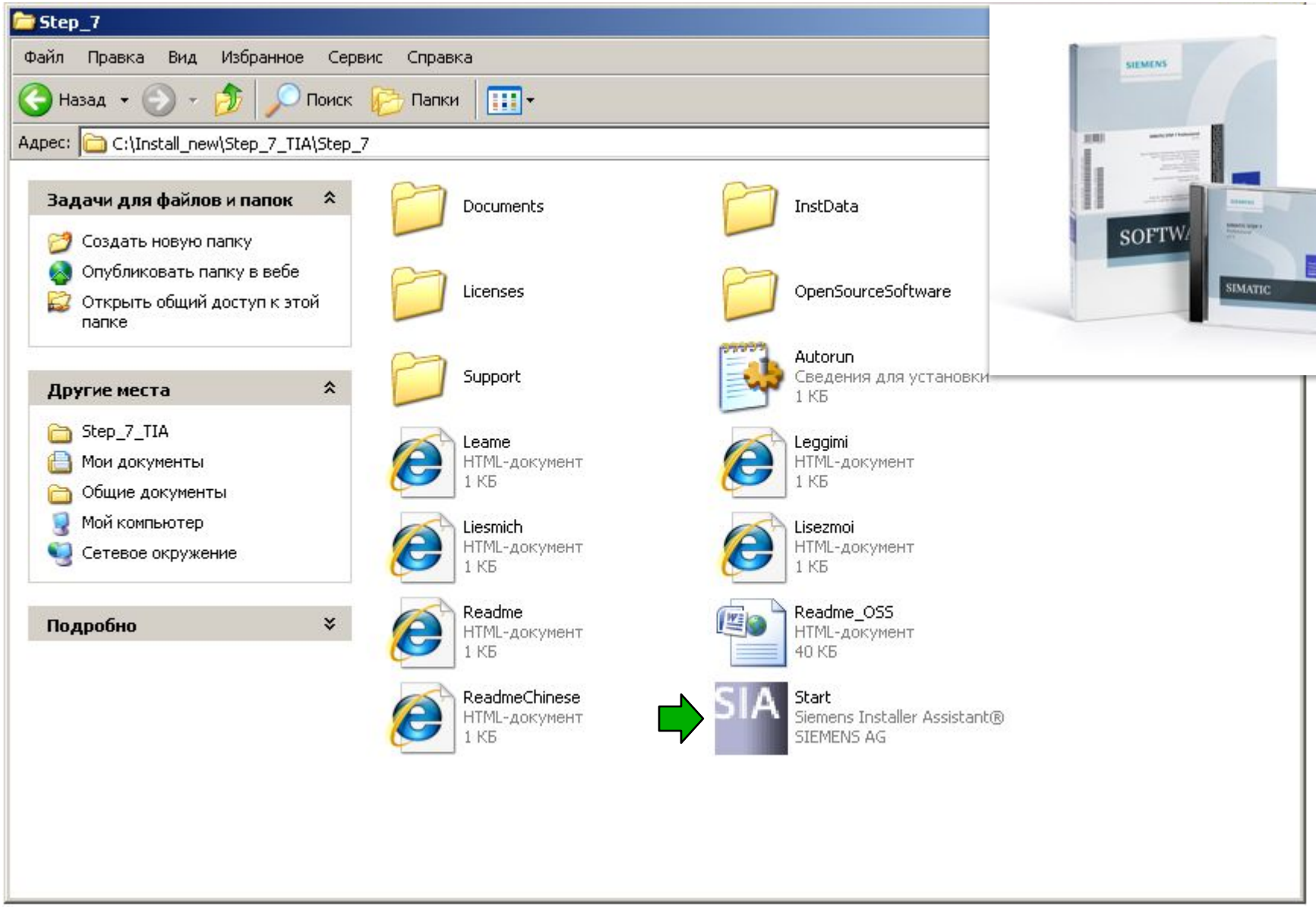
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 2

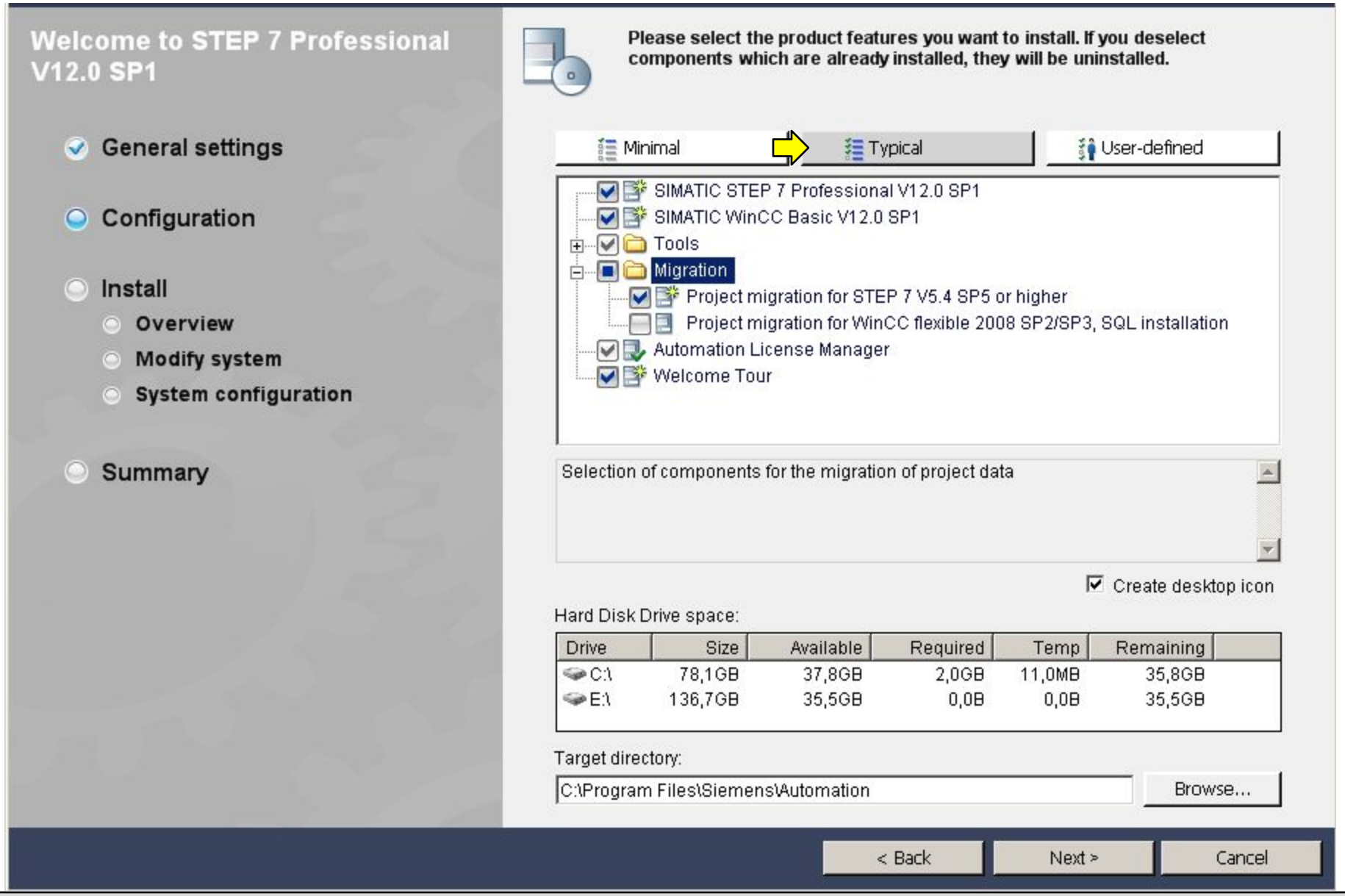
Общие сведения, создание проекта, конфигурирование станции



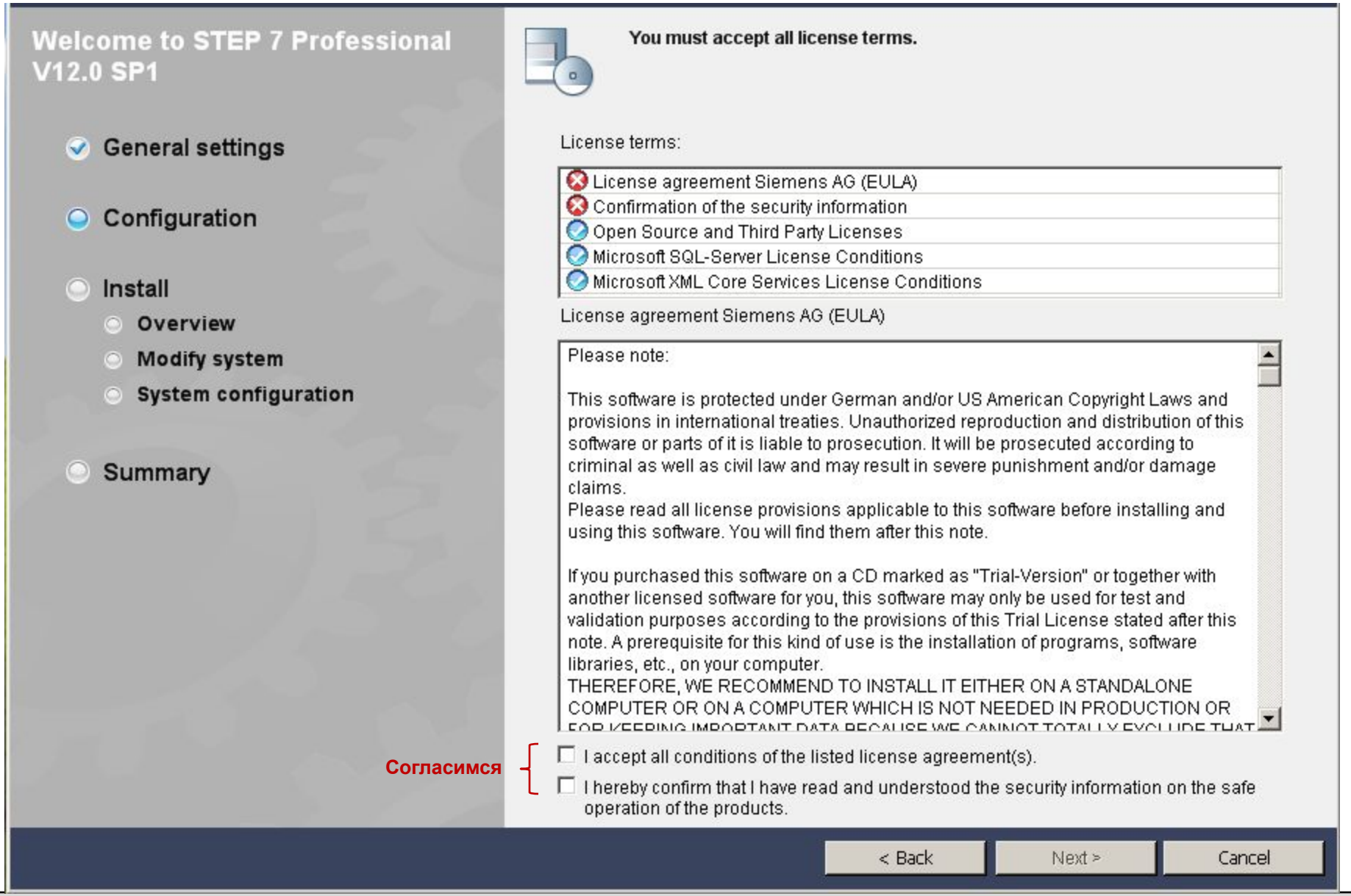
TIA-portal V12.0 SP1. Установка.



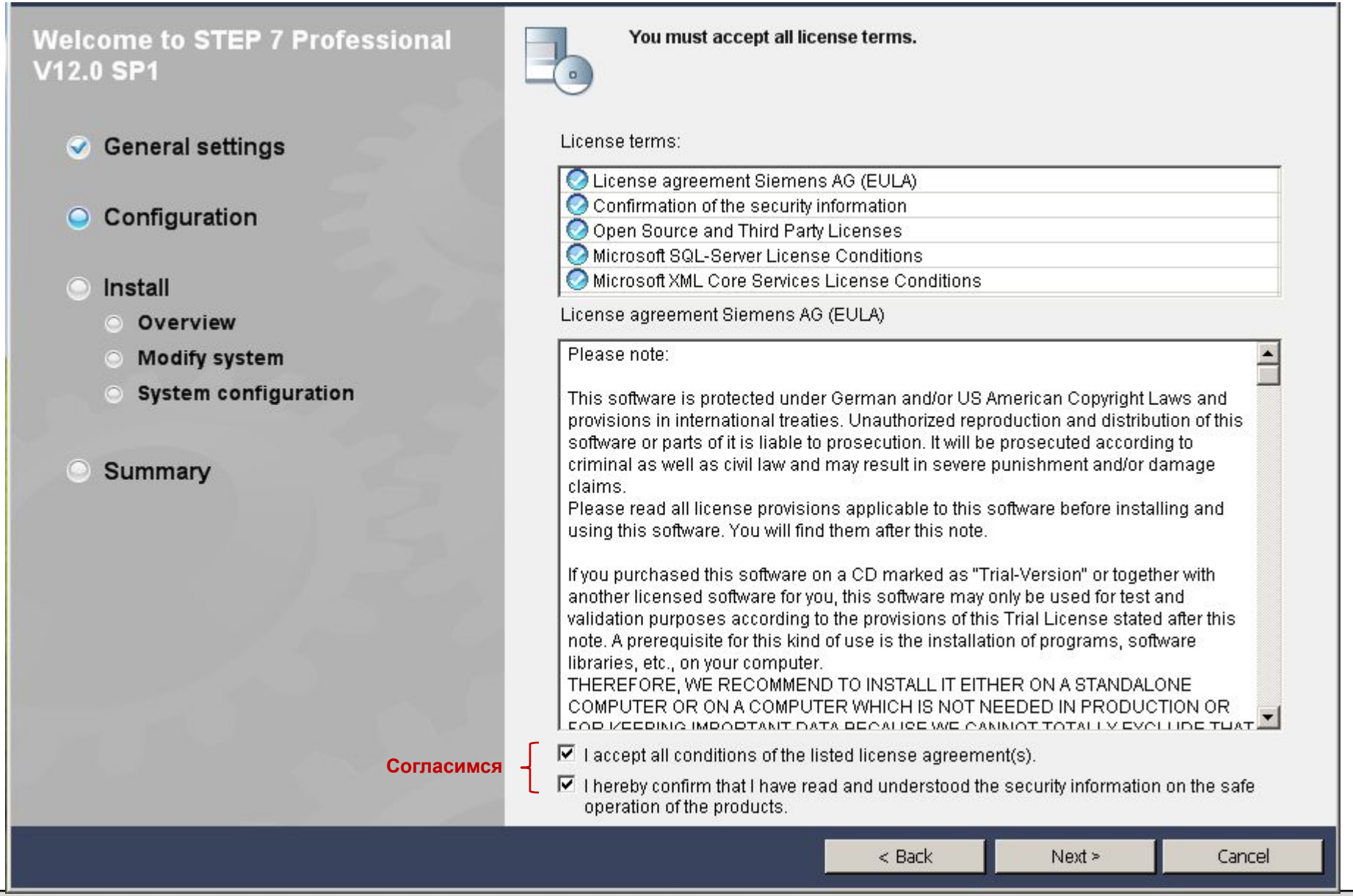
TIA-portal V12.0 SP1. Установка.



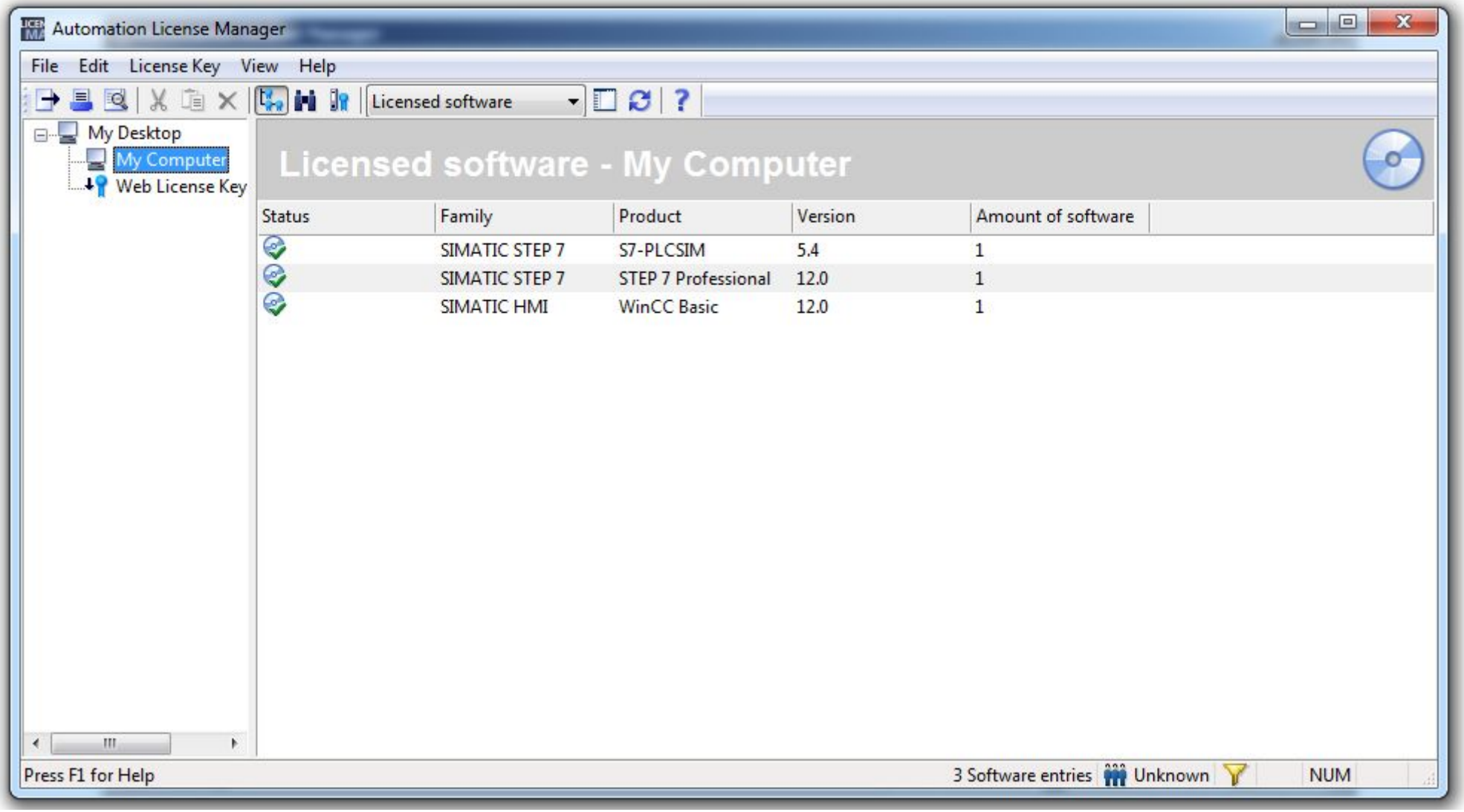
TIA-portal V12.0 SP1. Установка.



TIA-portal V12.0 SP1. Установка.

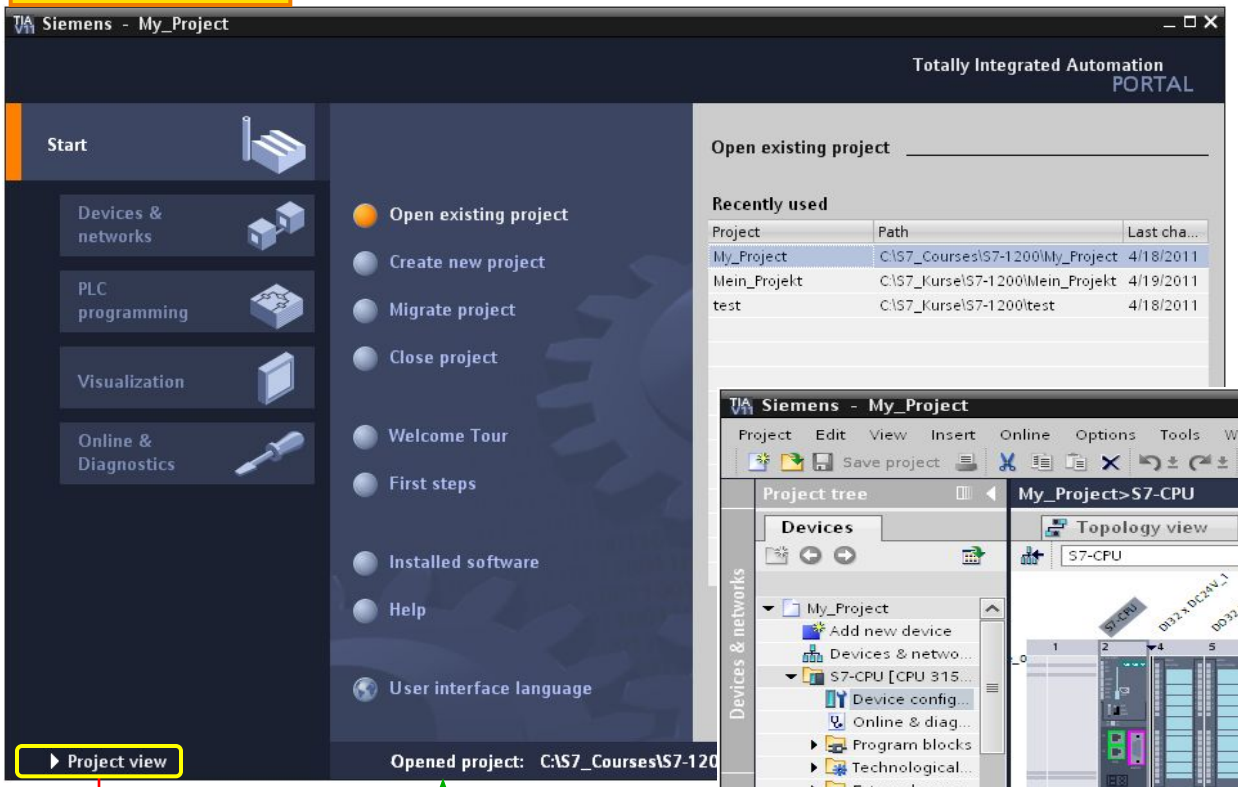


Проверка необходимых лицензий

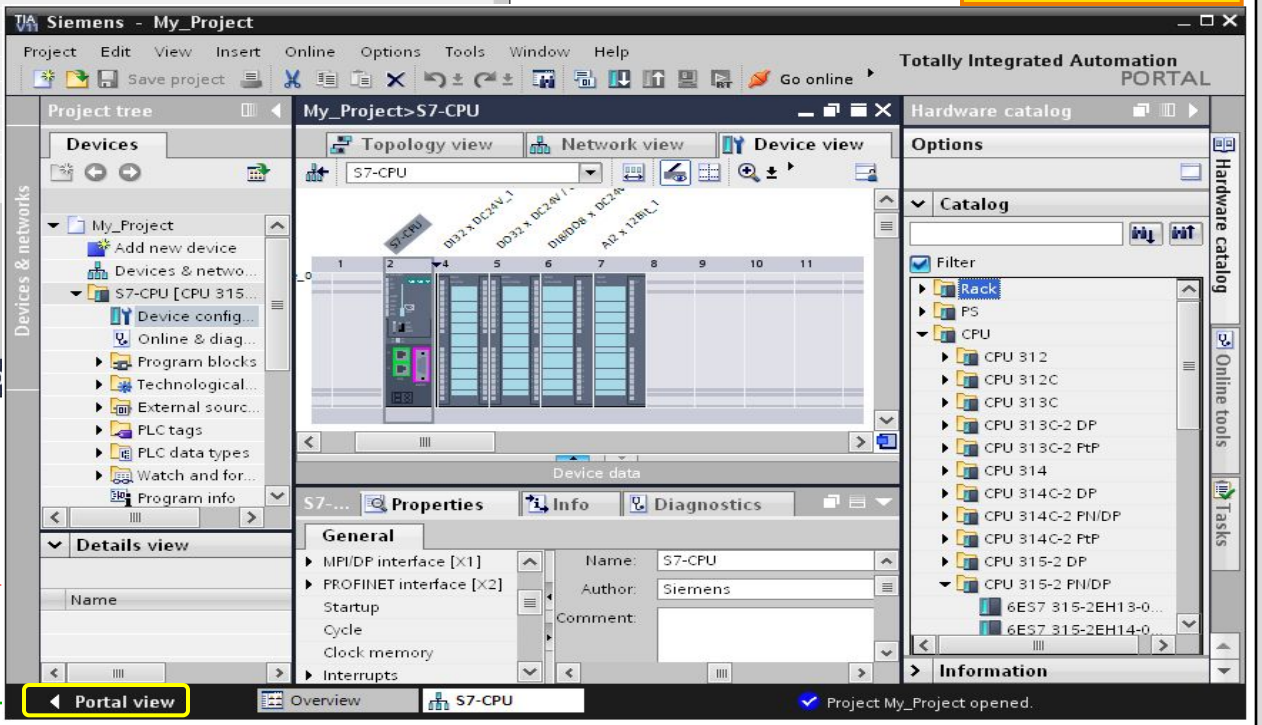


Виды пользовательского интерфейса

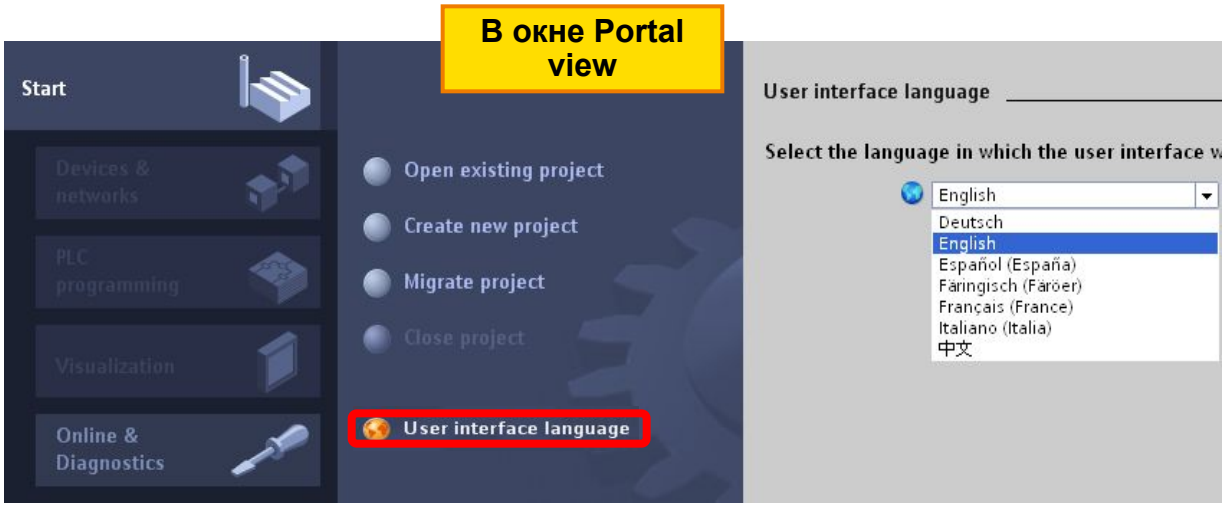
Portal view

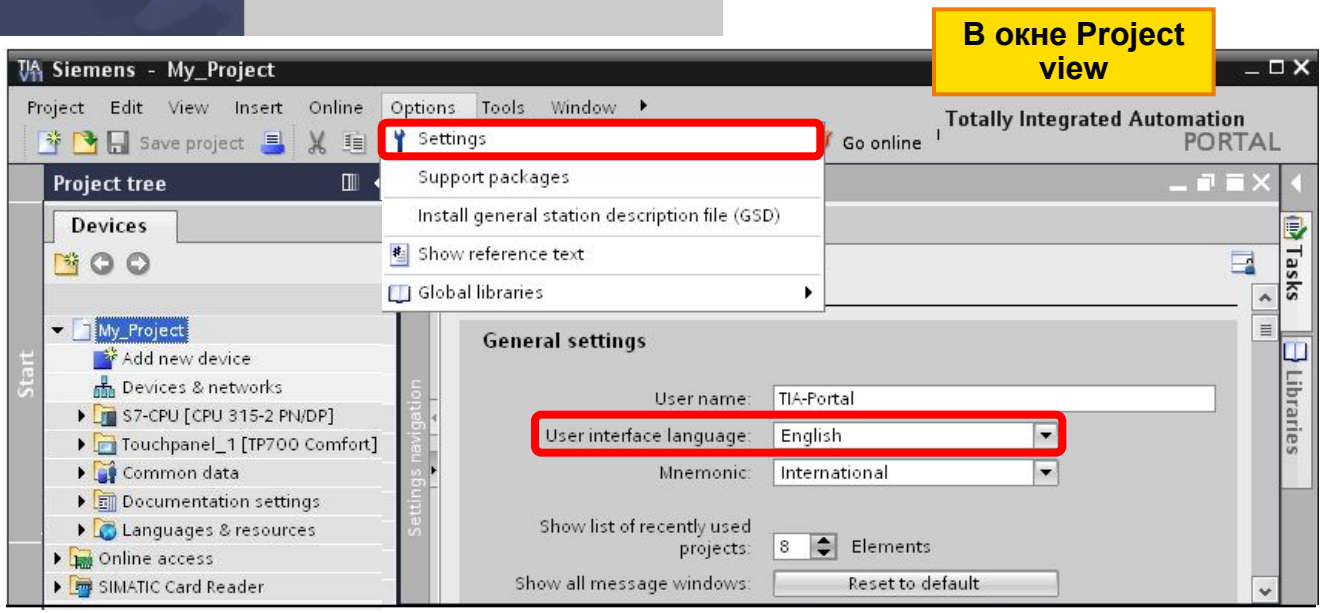


Project view

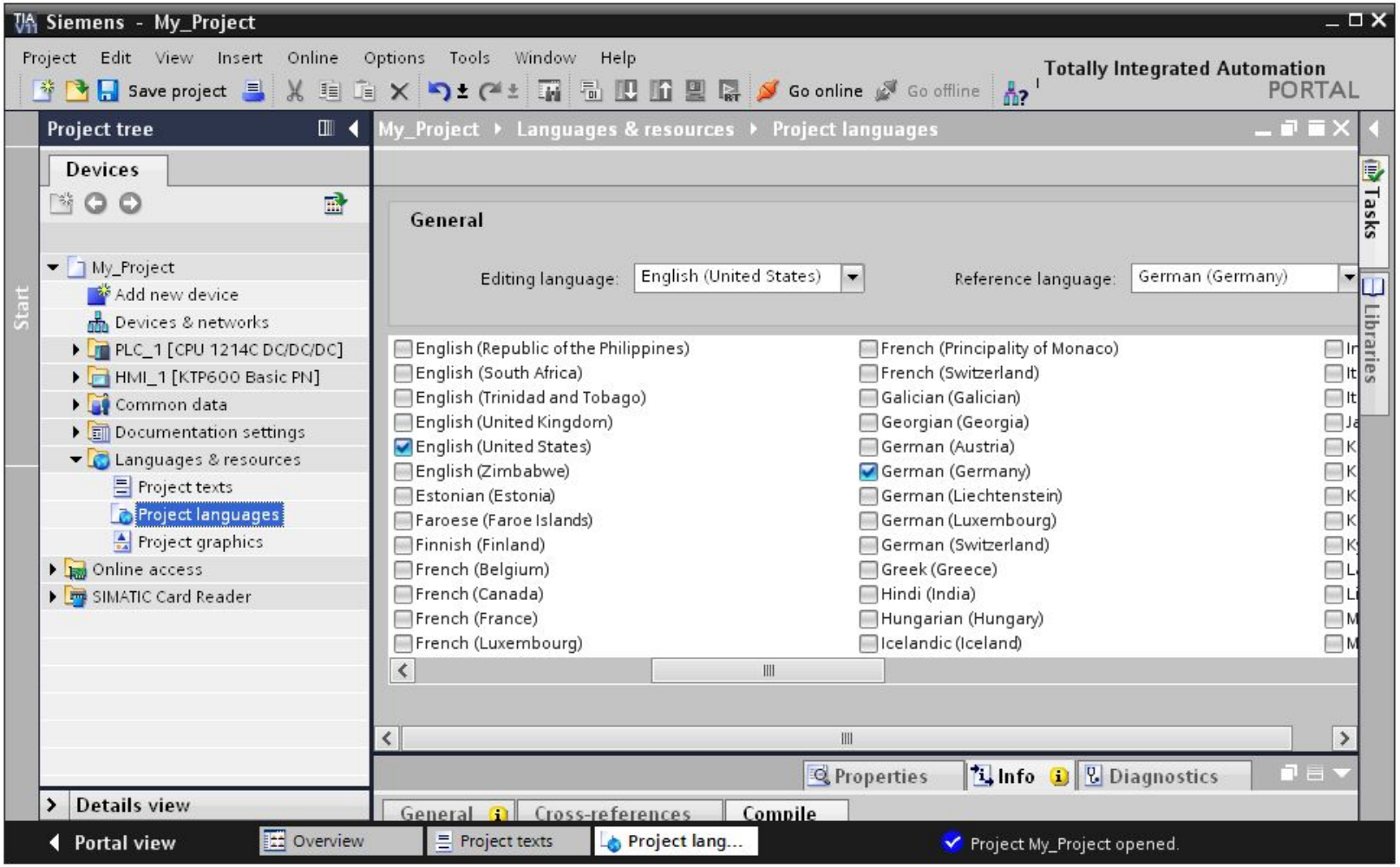


Языки пользовательского интерфейса

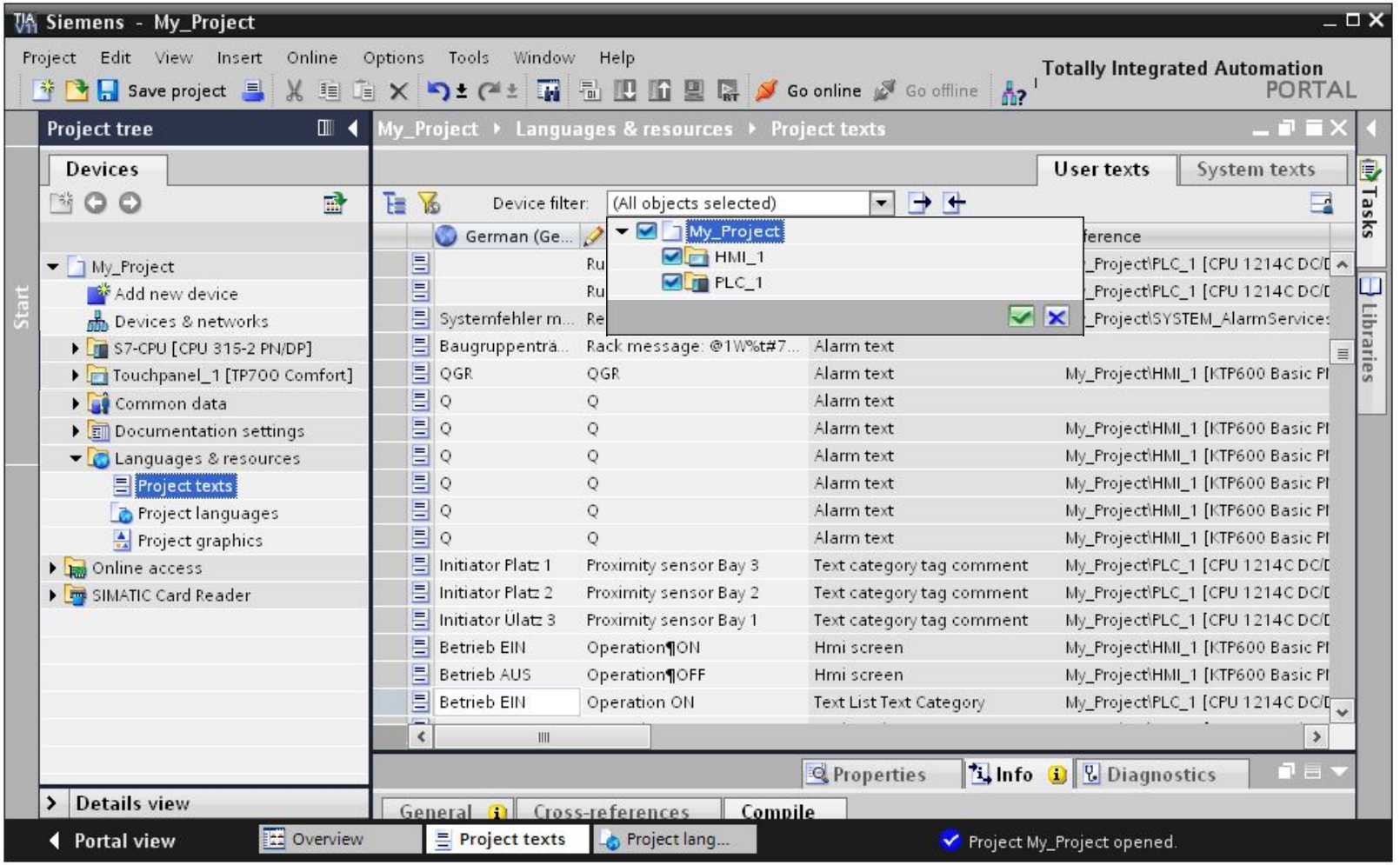




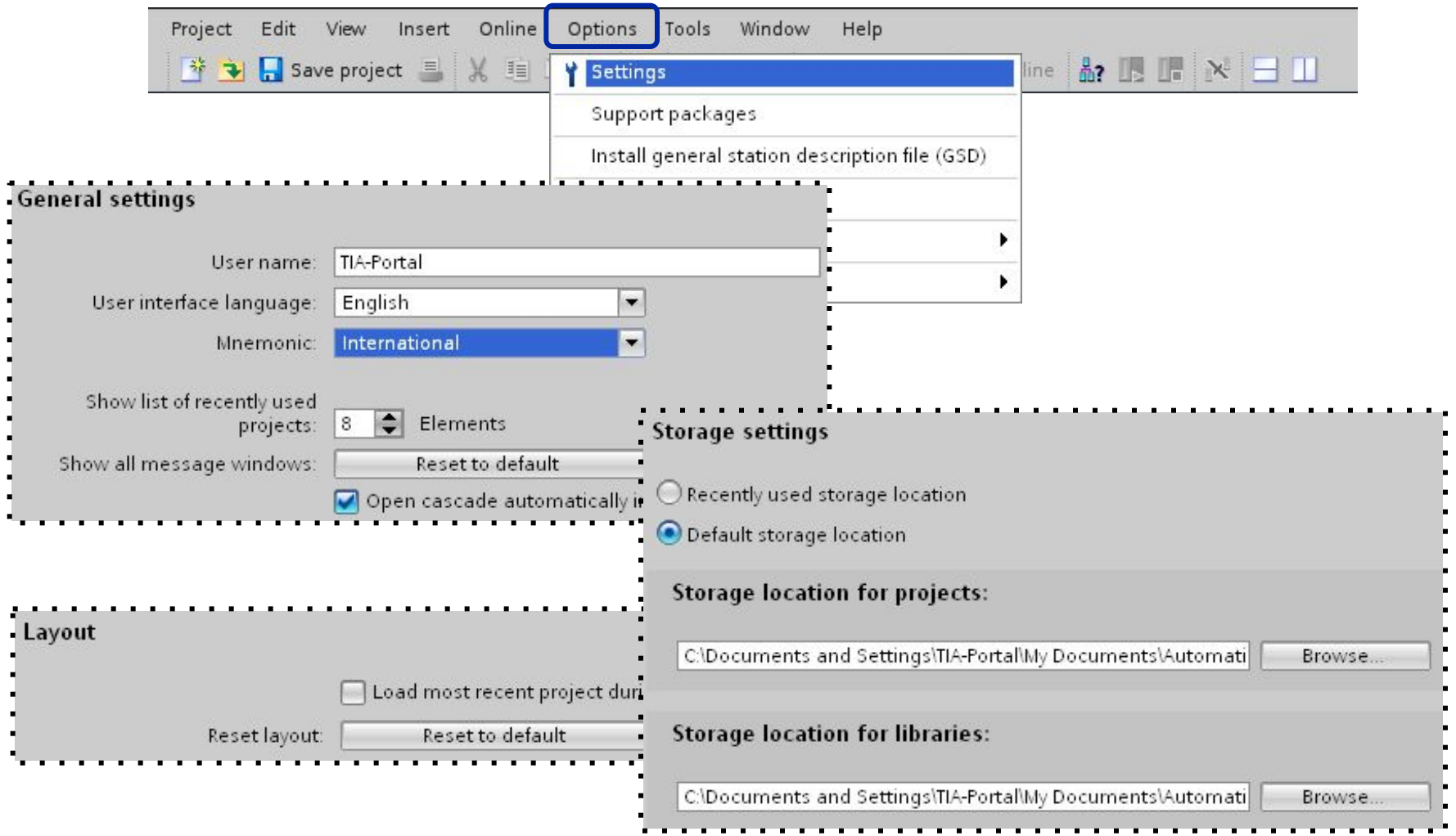
Языки проекта



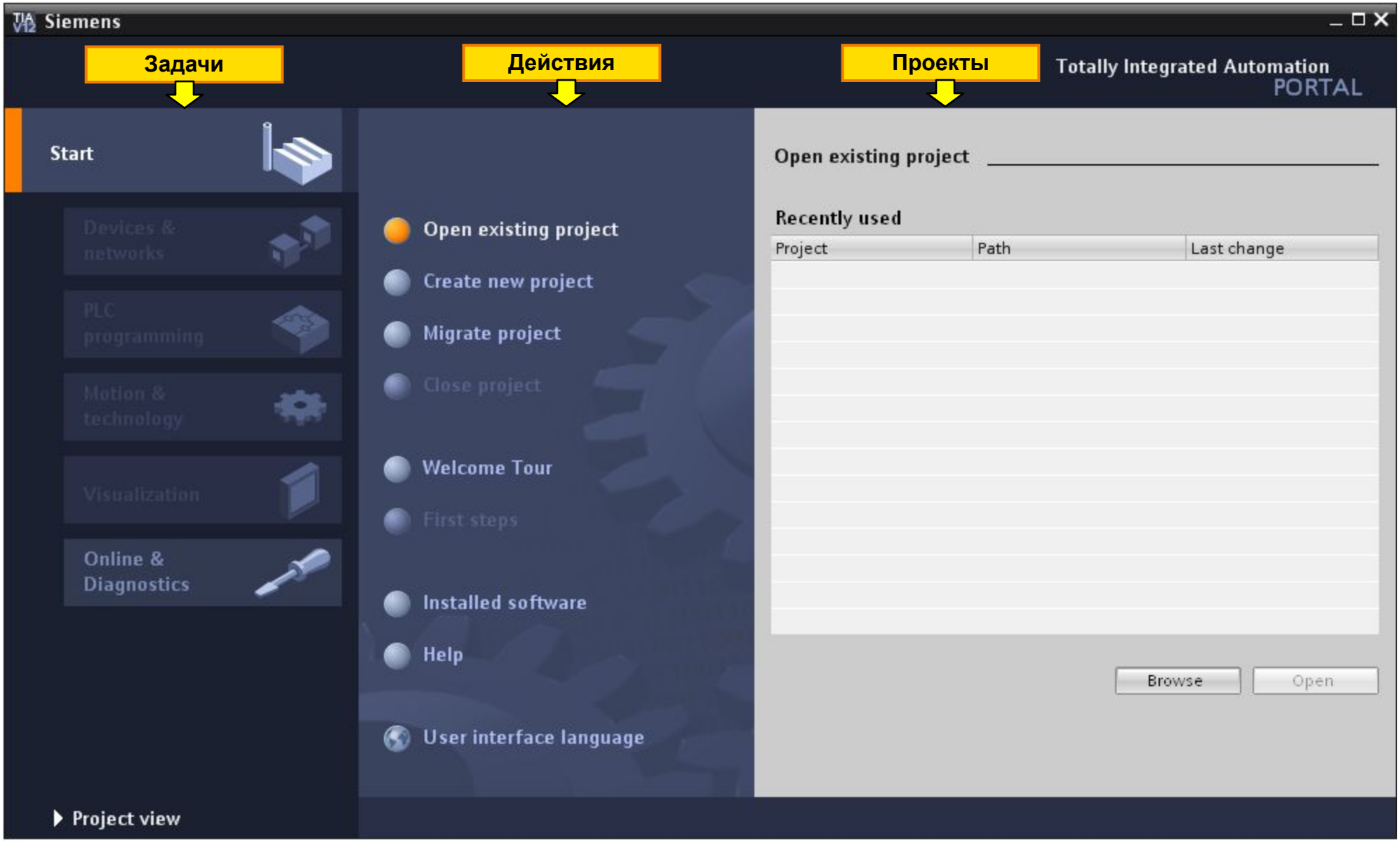
Тексты проекта



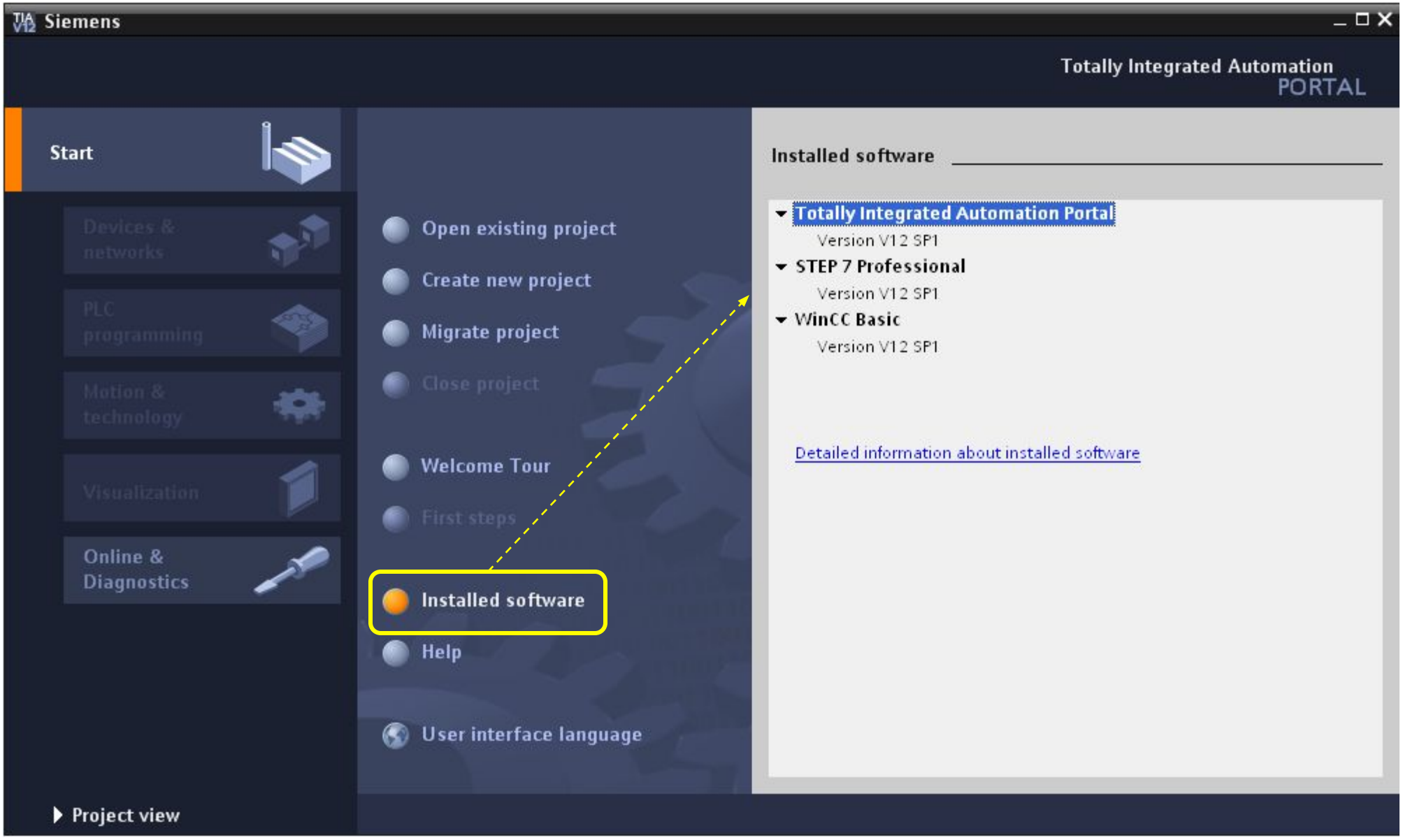
Настройки: язык, место сохранения, отображение



Опции интерфейса Portal view

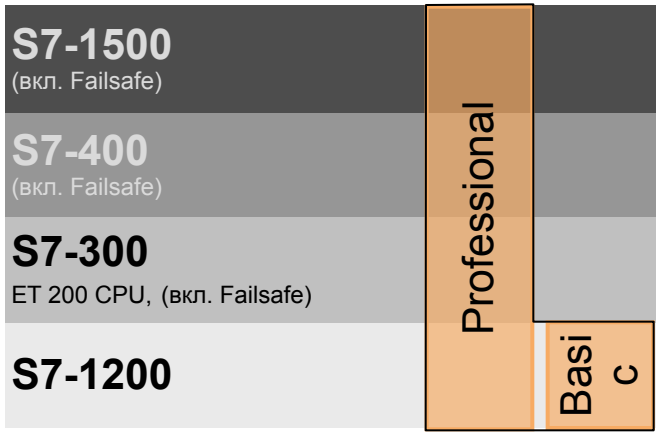


Перечень установленного софта.



Обзор пакета ПО STEP 7 V12 в составе TIA-portal

SIMATIC STEP 7 V12



STEP 7 Basic V12 для S7-1200

- LAD, FBD, SCL

STEP 7 Professional V12 для S7-1200, S7-300, S7-400, S7-1500

- LAD, FBD, SCL для S7-1200
- LAD, FBD, STL, SCL, S7-GRAPH для S7-300/400, S7-1500

Дополнительные пакеты
(STEP 7 Safety, PID controller пакет, STEP 7 Easy Motion)

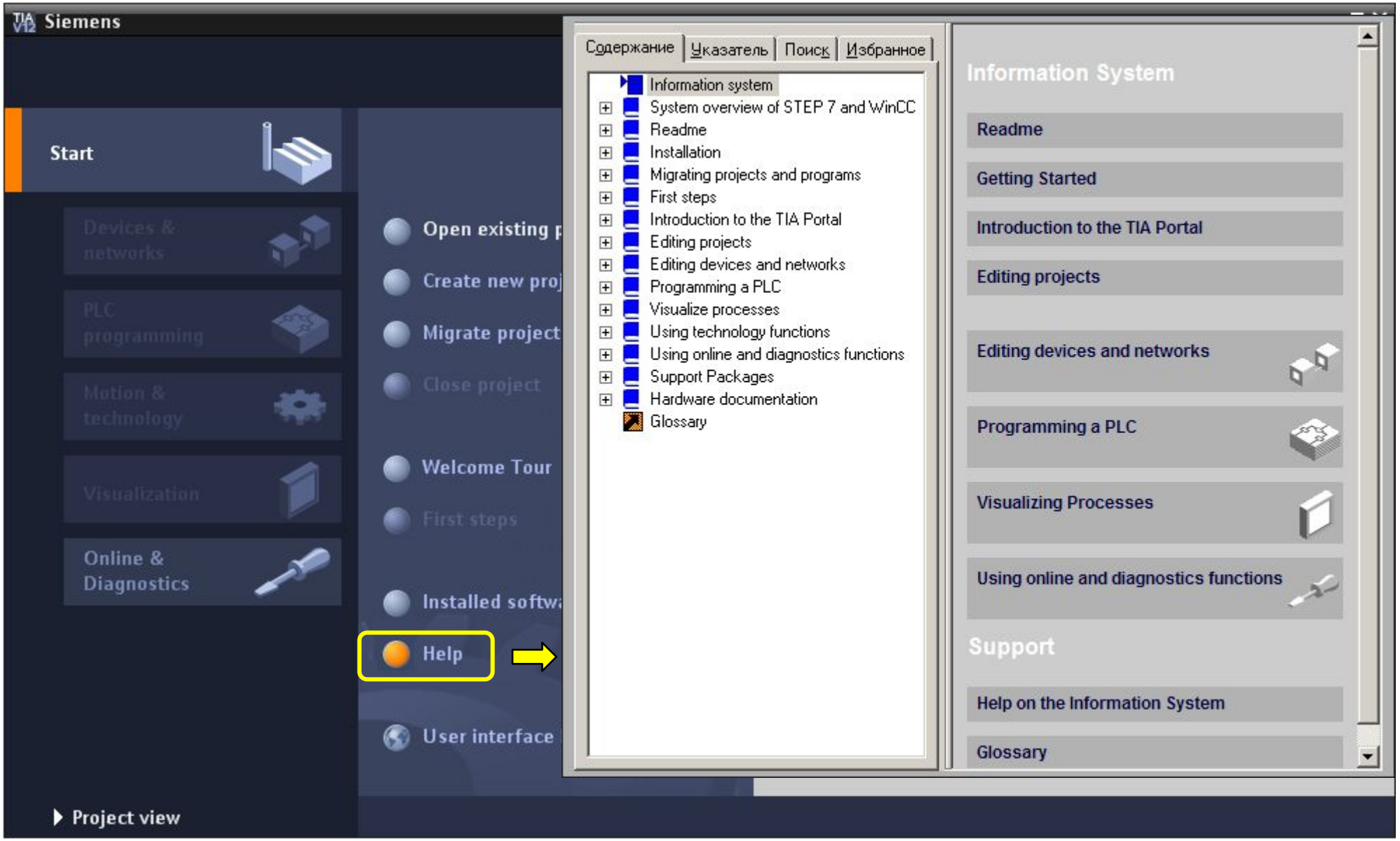
Коммуникации

• PROFIBUS, PROFINET, AS-i, IO-Link, ET 200, топология сети

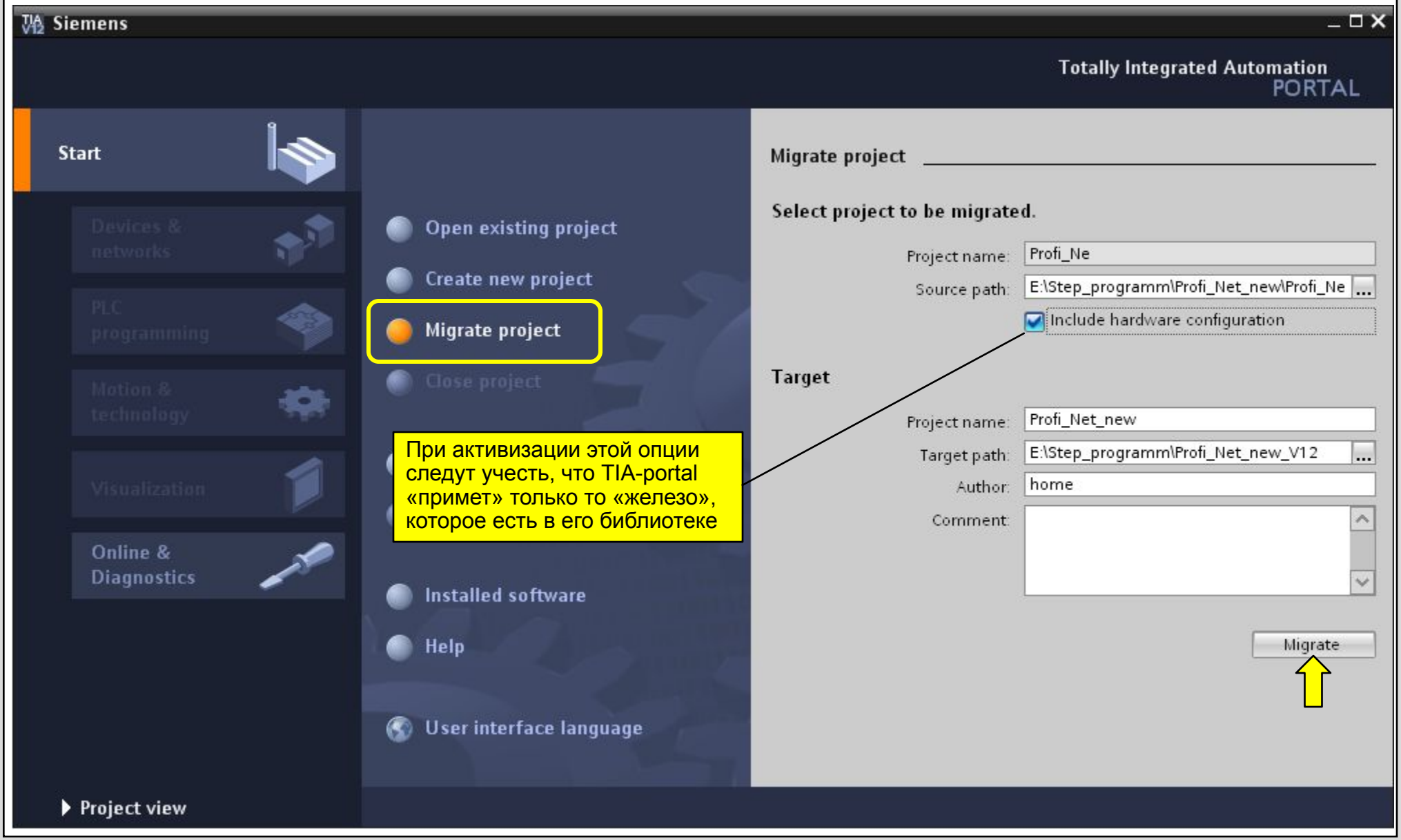
Расширенные функции

- Системная диагностика, импорт/экспорт в Excel.

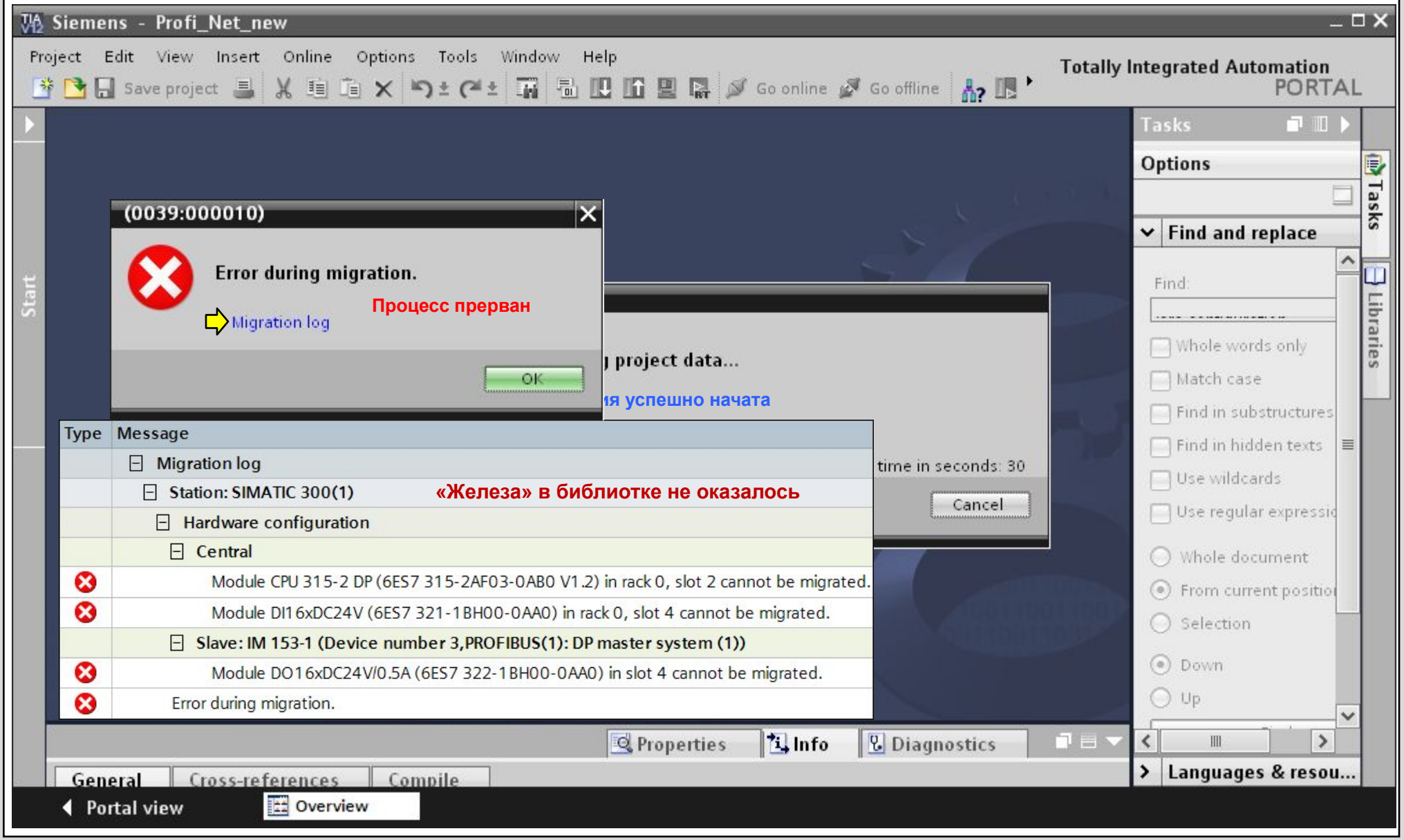
Встроенная информационная поддержка



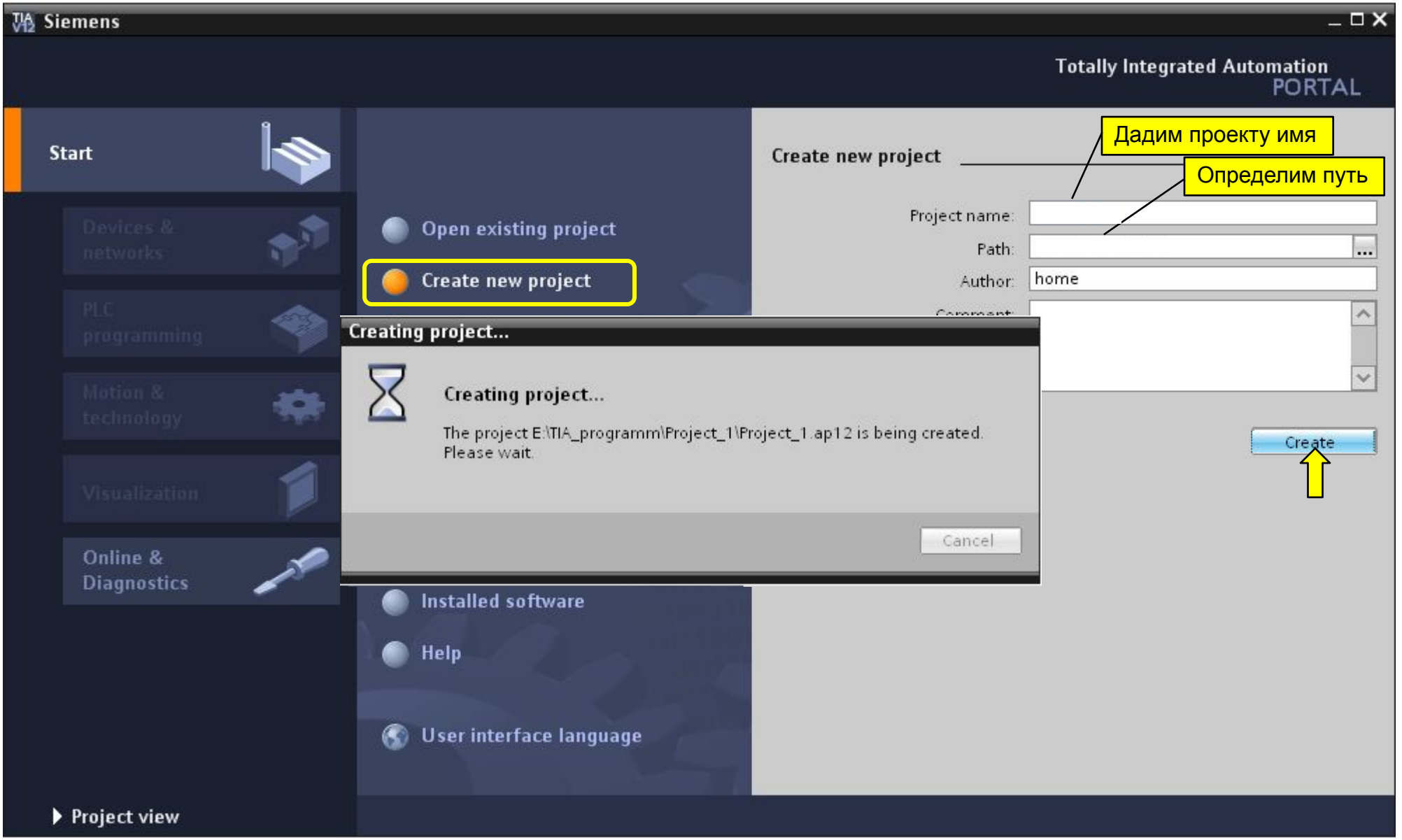
Миграция в проект TIA-portal программы из Step-7



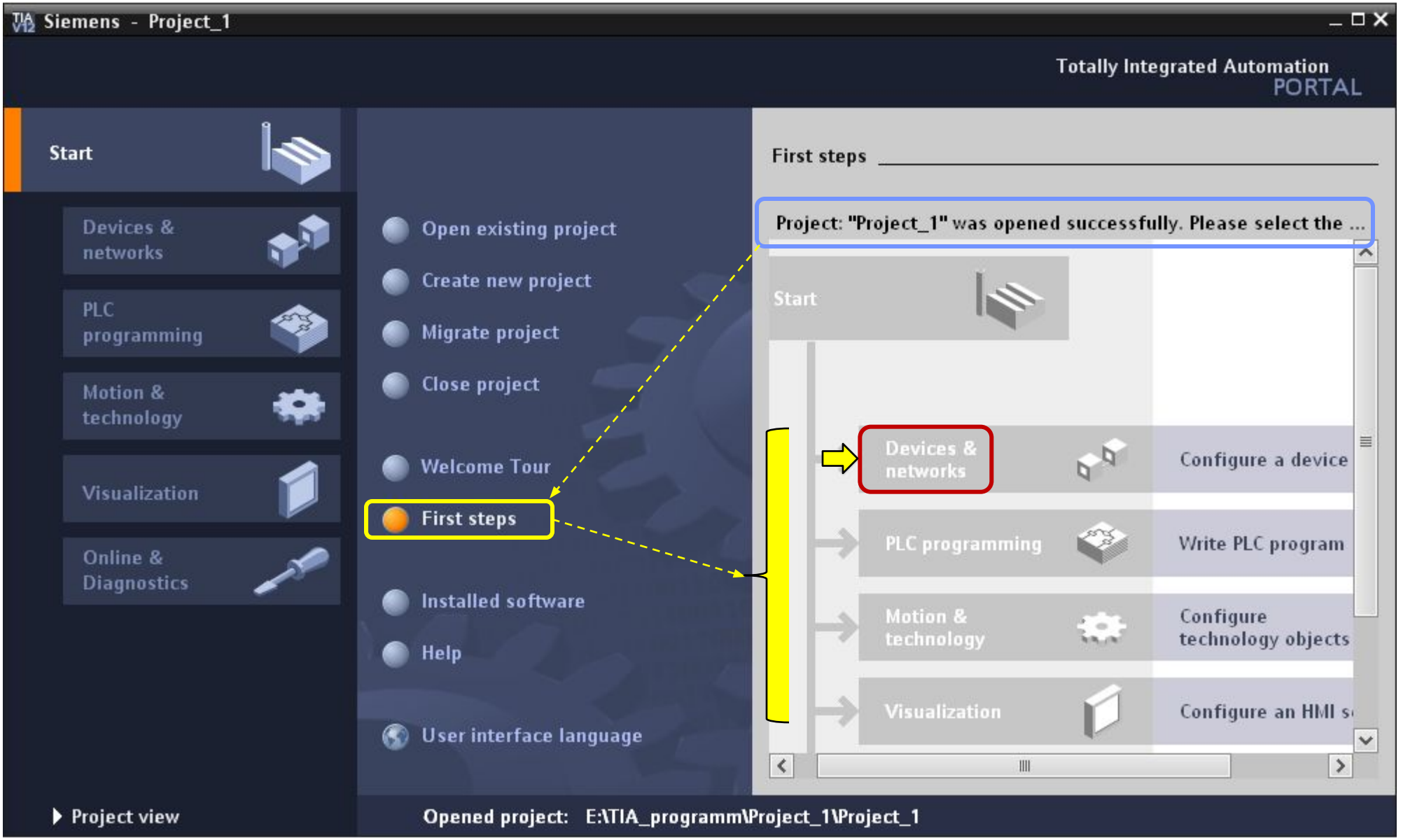
Миграция в проект TIA-portal программы из Step-7



TIA-portal V12.0 SP1. Создание нового проекта

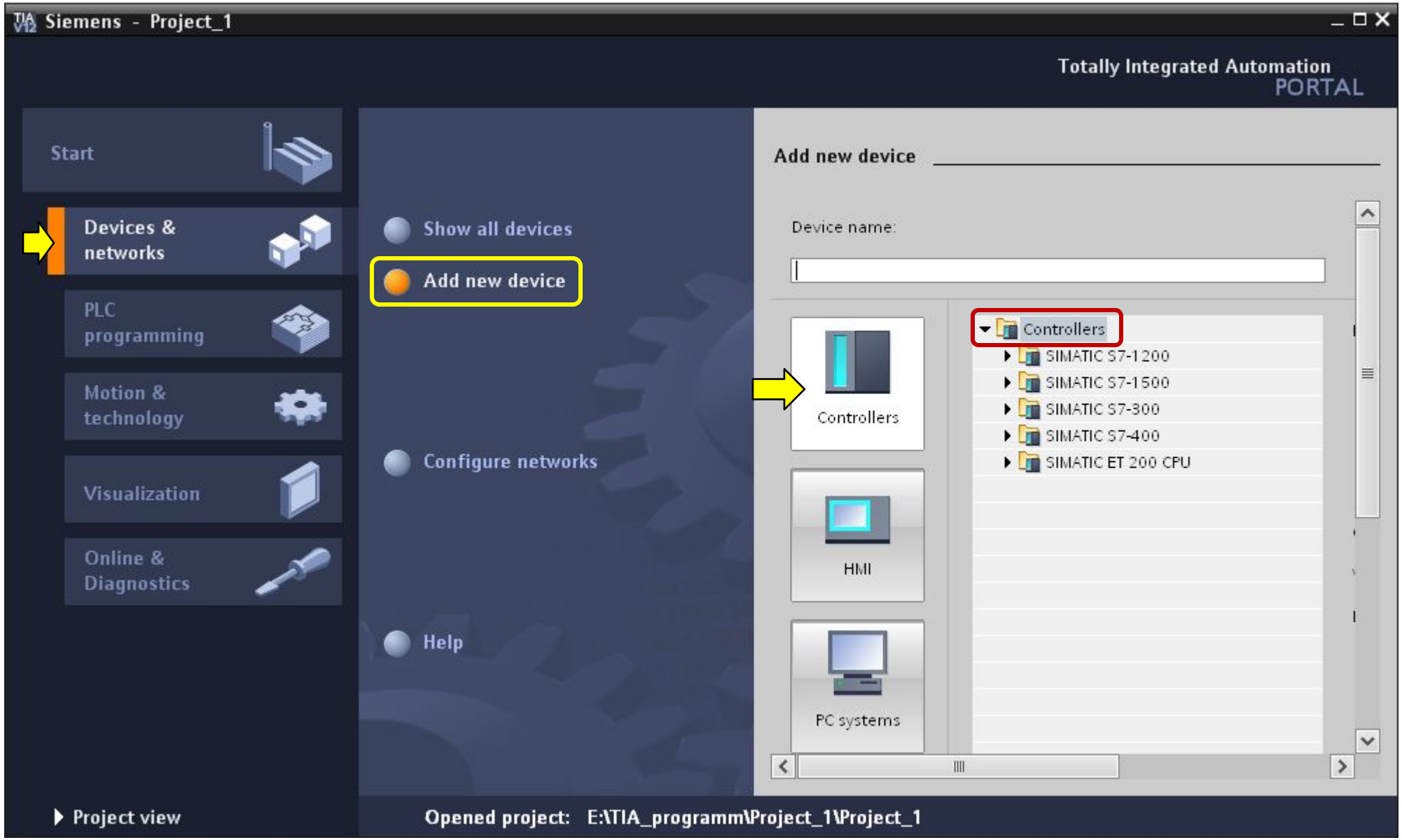


TIA-portal V12.0 SP1. Создание нового проекта

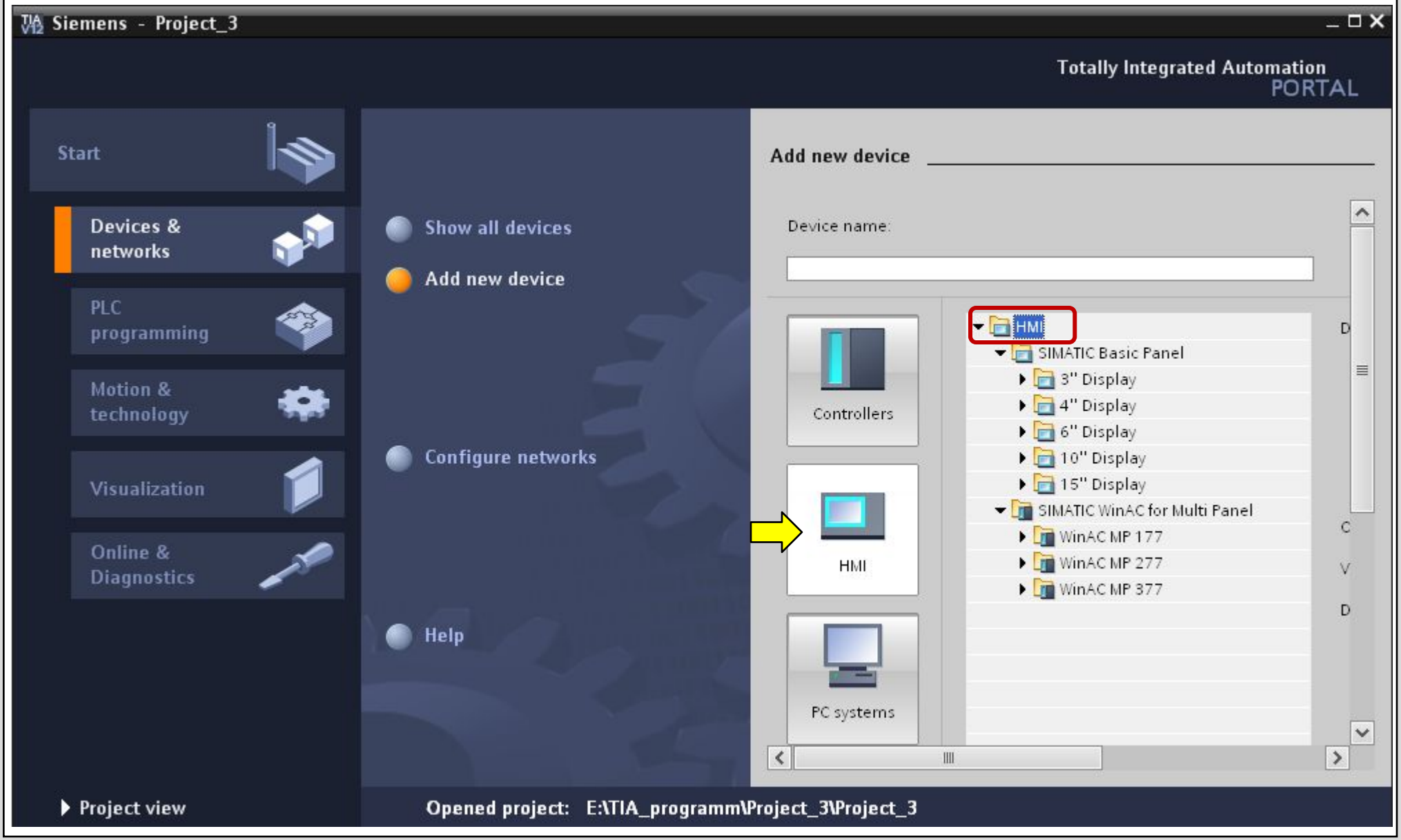


Конфигурация оборудования и установка параметров

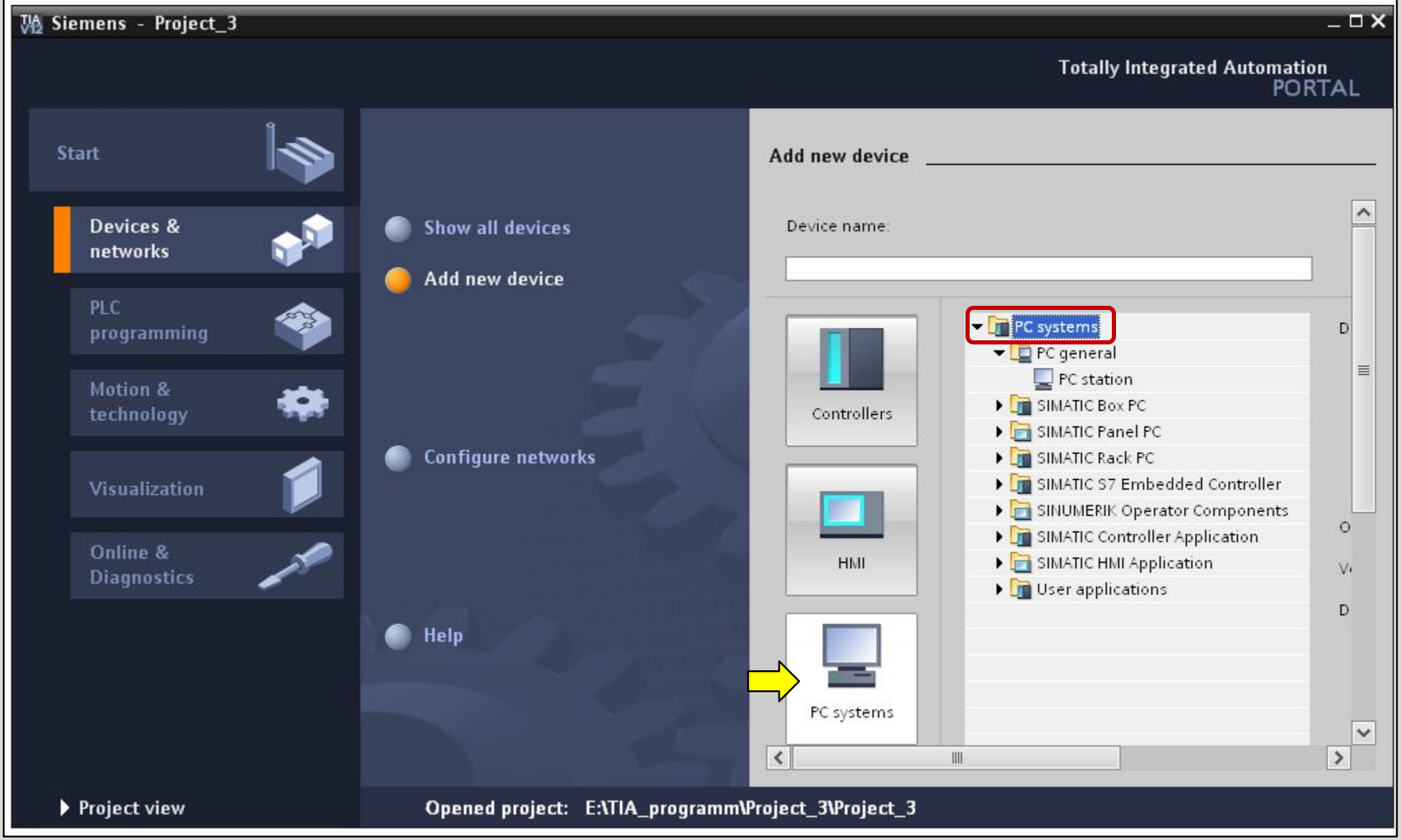
TIA-portal V12.0 SP1. Общий каталог устройств



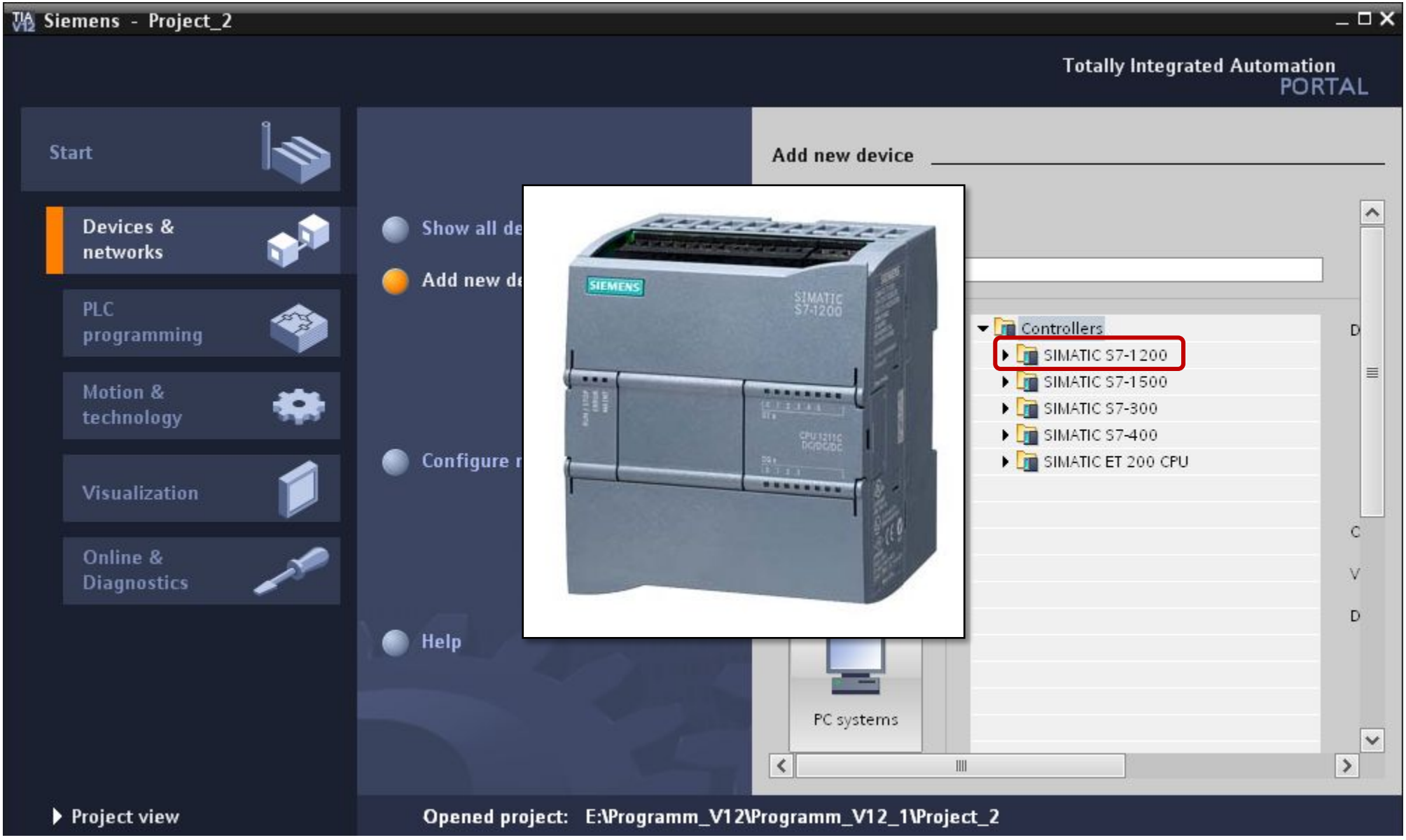
TIA-portal V12.0 SP1. Общий каталог устройств



TIA-portal V12.0 SP1. Общий каталог устройств

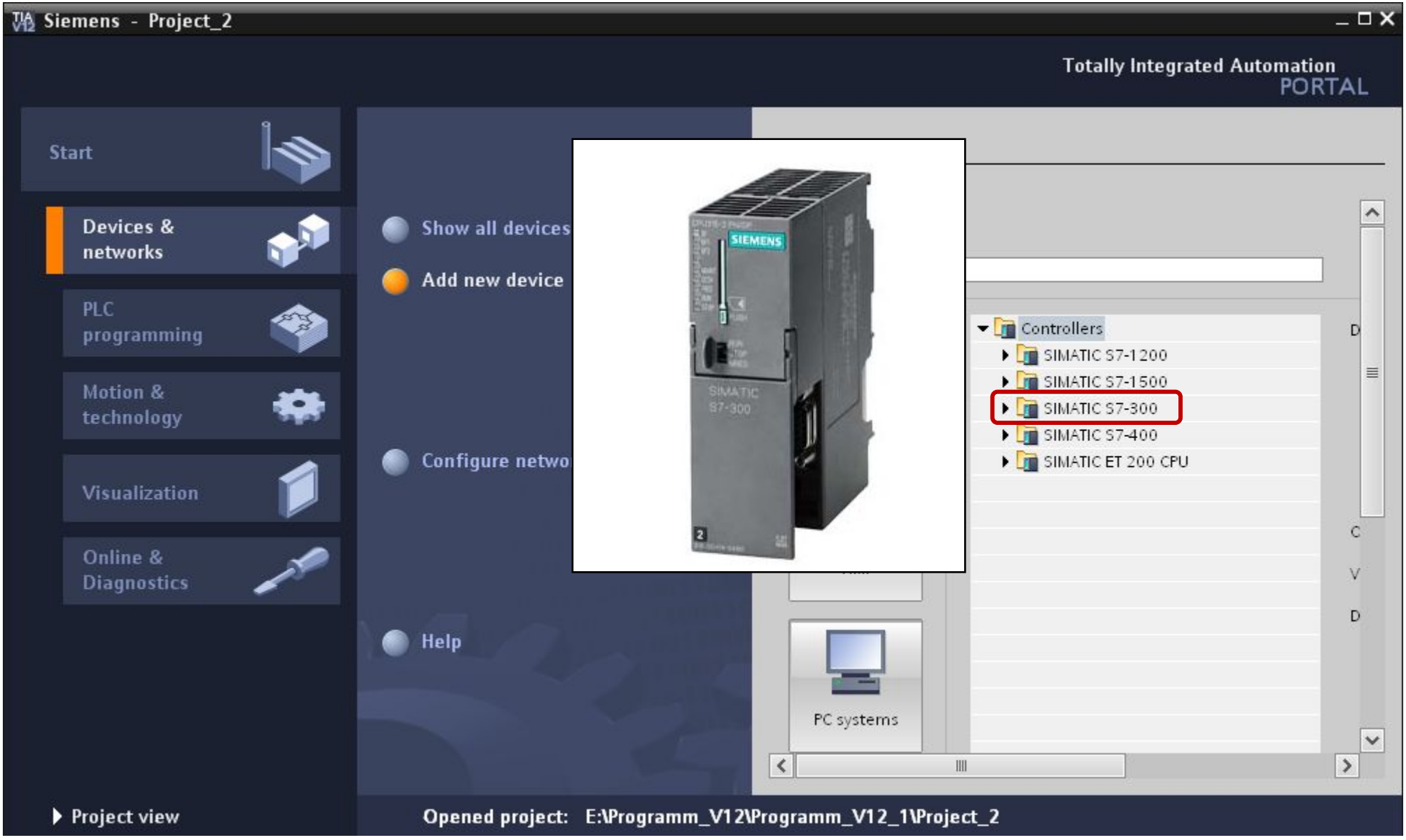


TIA-portal V12.0 SP1. Каталог устройств CPU SIMATIC S-7xx

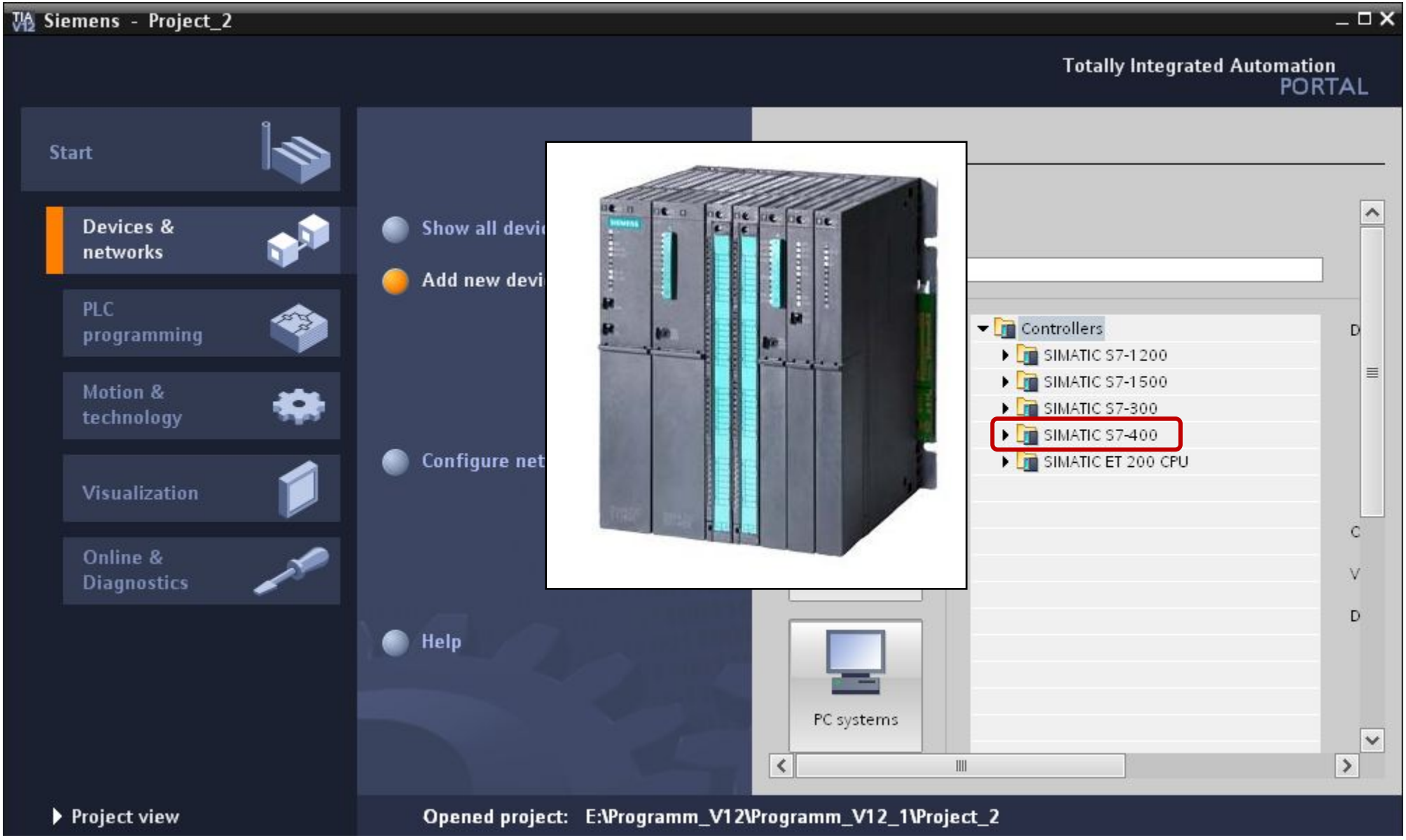


TIA-portal V12.0 SP1. Каталог устройств CPU SIMATIC S-7xx

TIA-portal V12.0 SP1. Каталог устройств CPU SIMATIC S-7xx

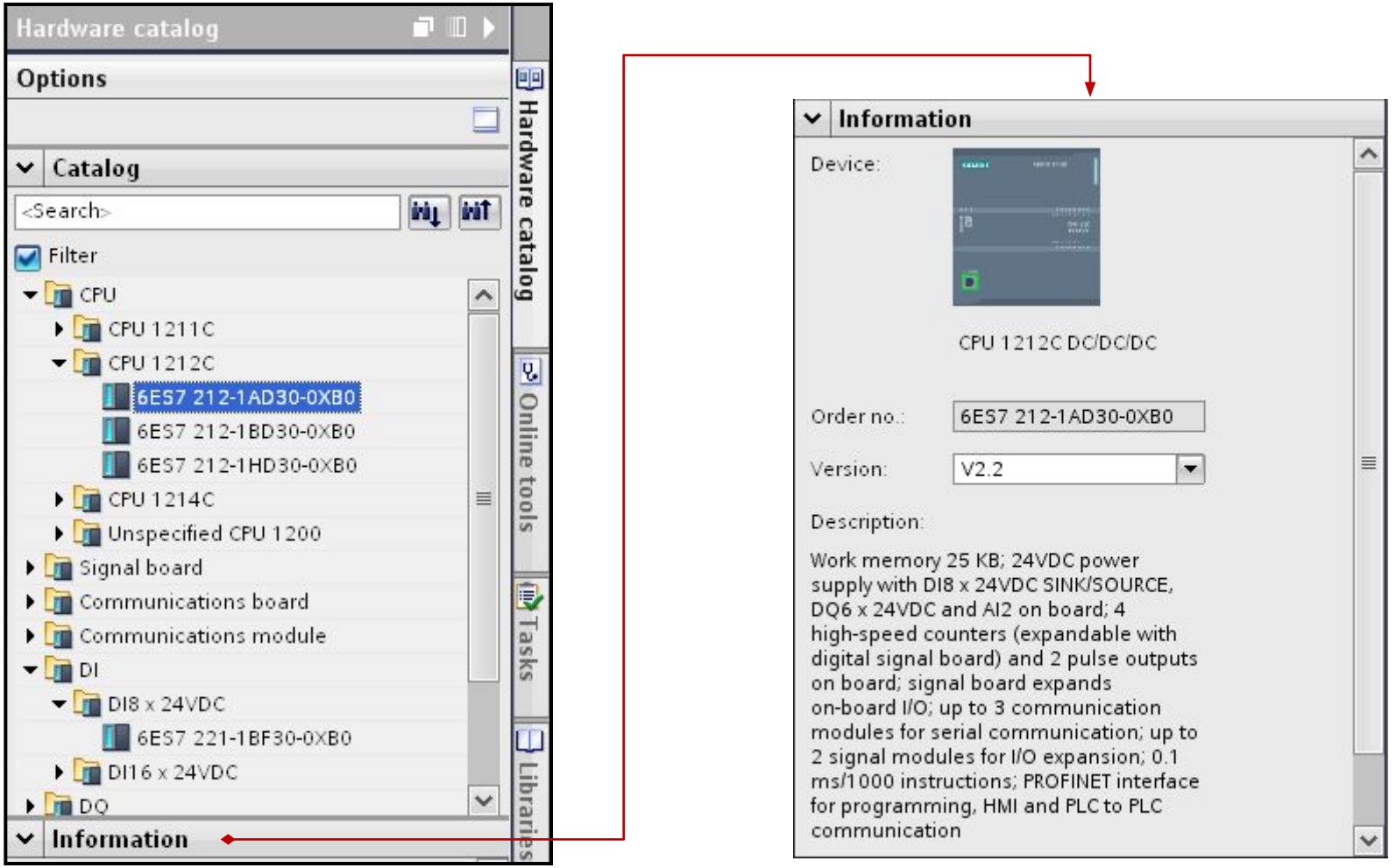


TIA-portal V12.0 SP1. Каталог устройств CPU SIMATIC S-7xx

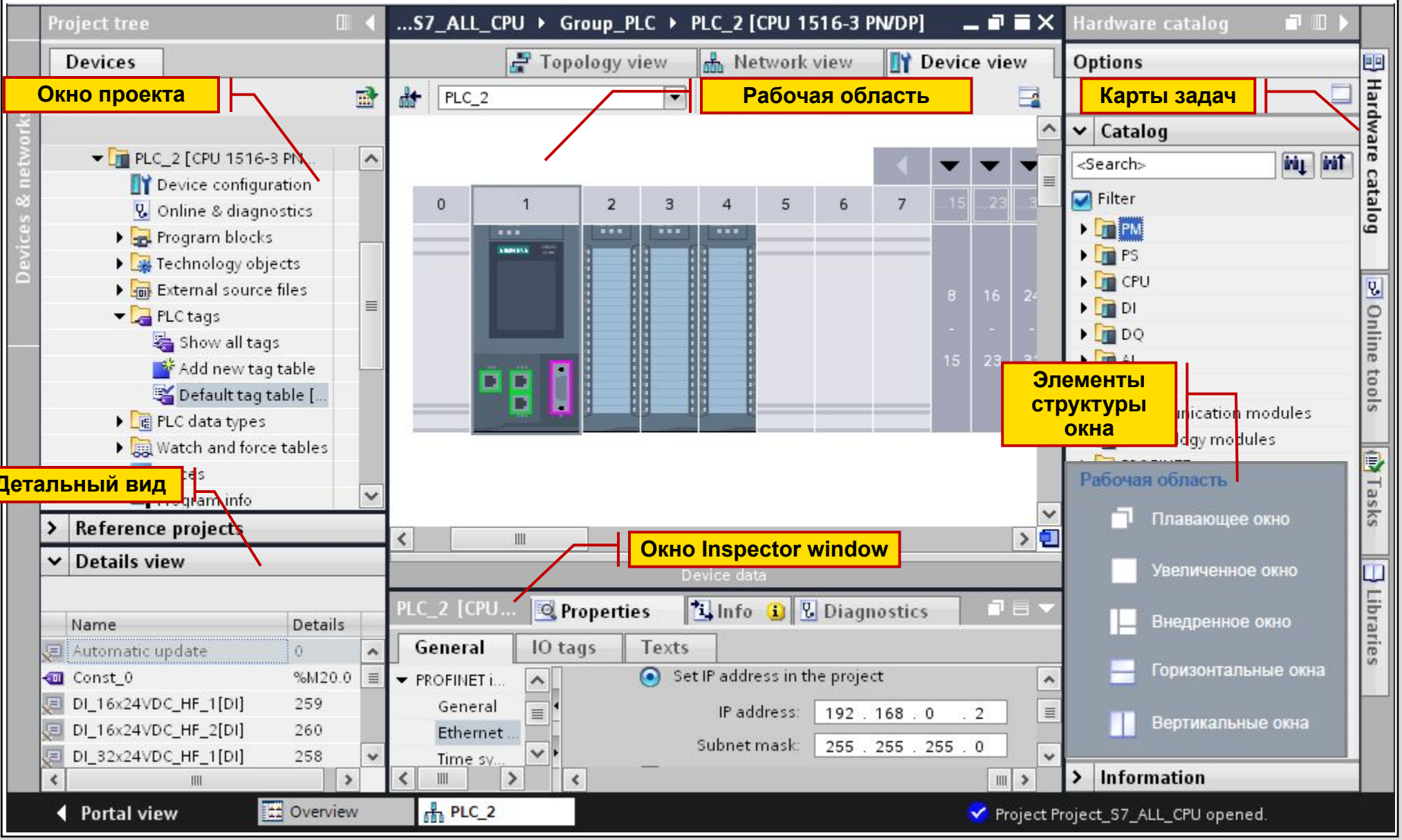


TIA-portal V12.0 SP1. Каталог устройств CPU SIMATIC S-7xx

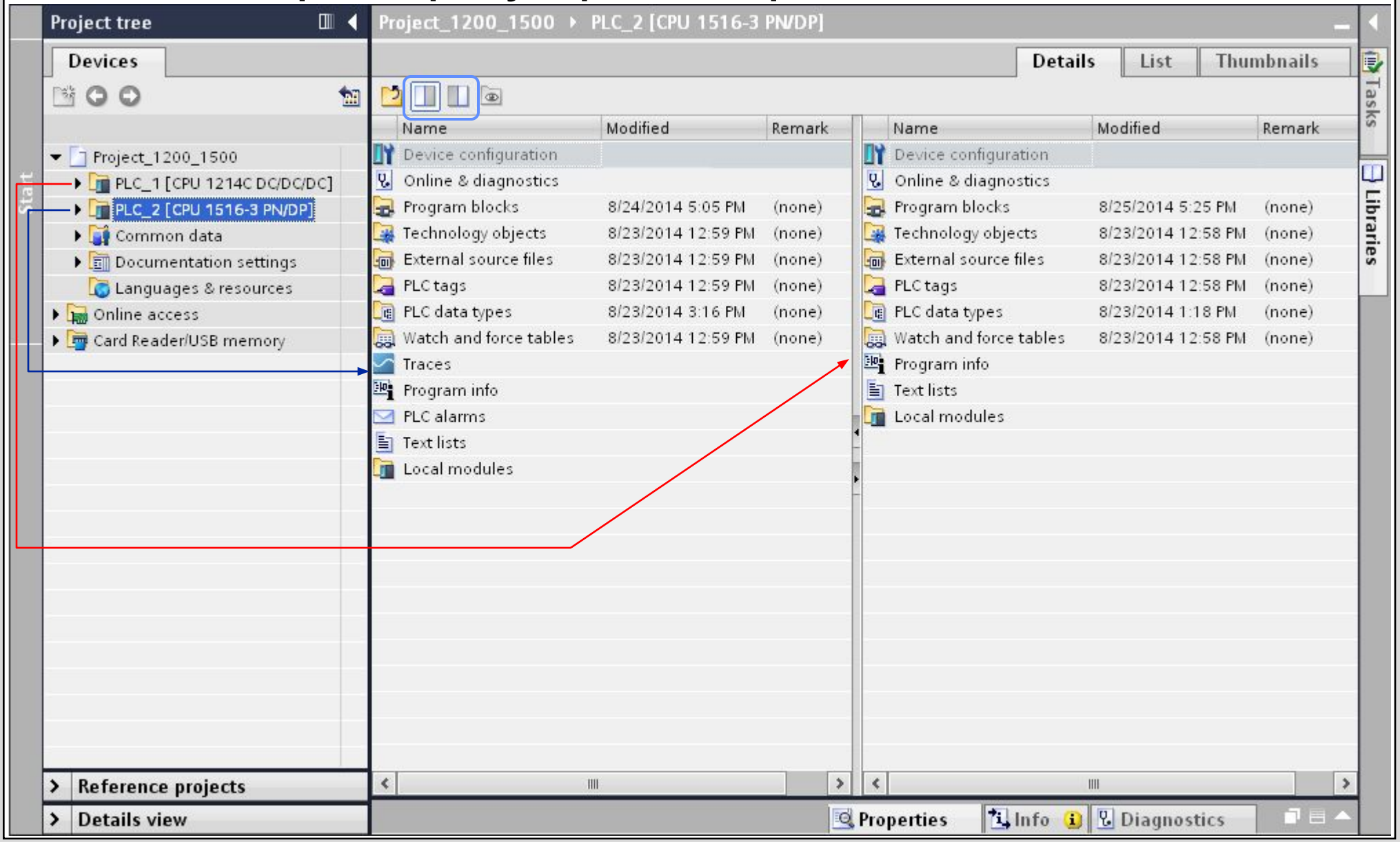
Информация о выбранном устройстве



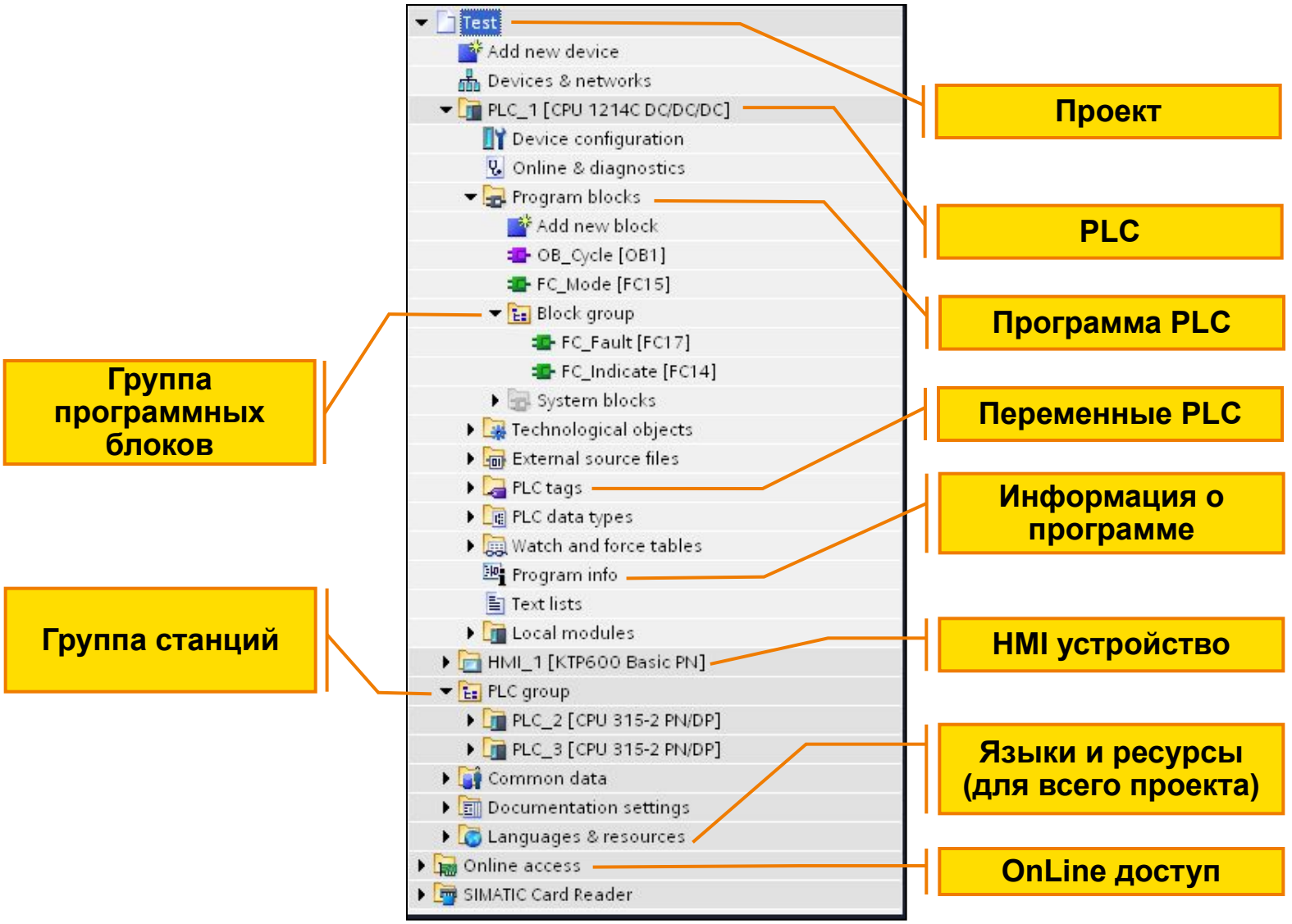
Компоненты отображения в Project view



Просмотр двух проектов в разделенных окнах

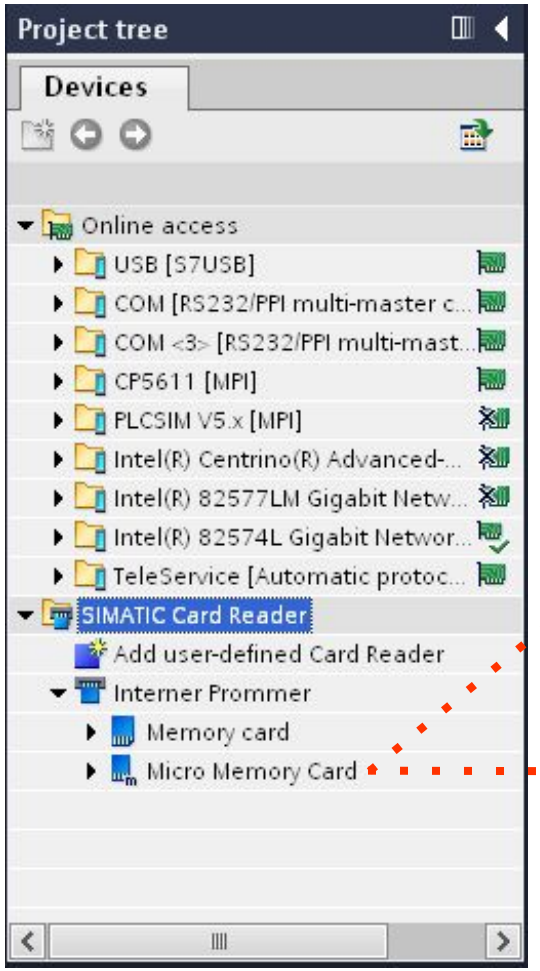


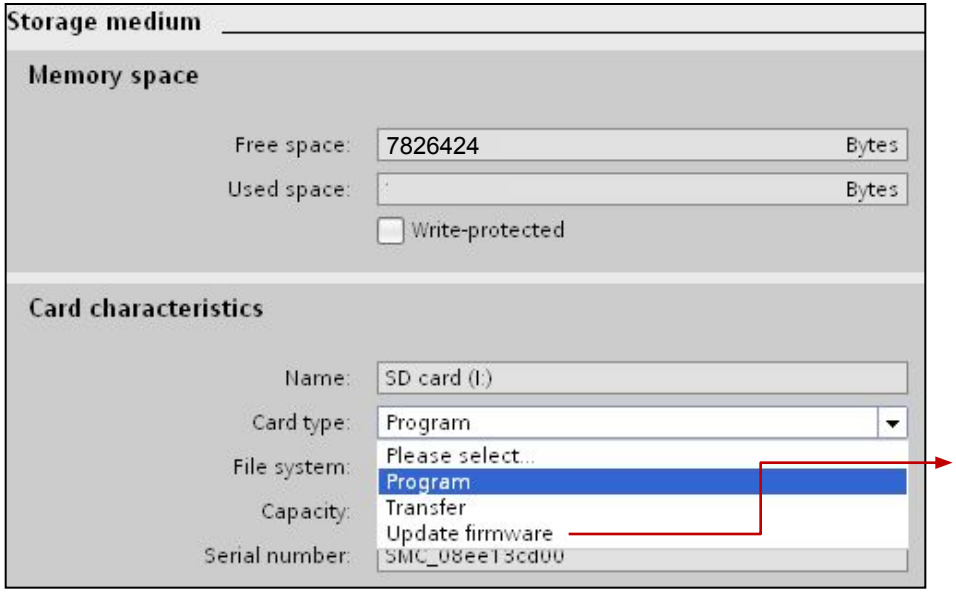
Навигация в окне проекта



Выбор режима использования программной карты

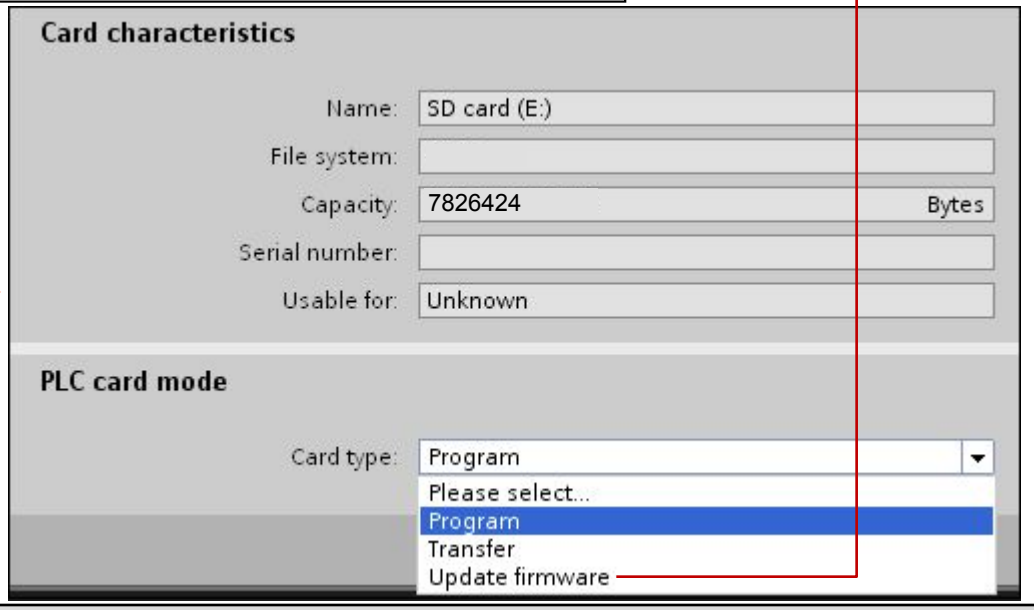
S7-1200



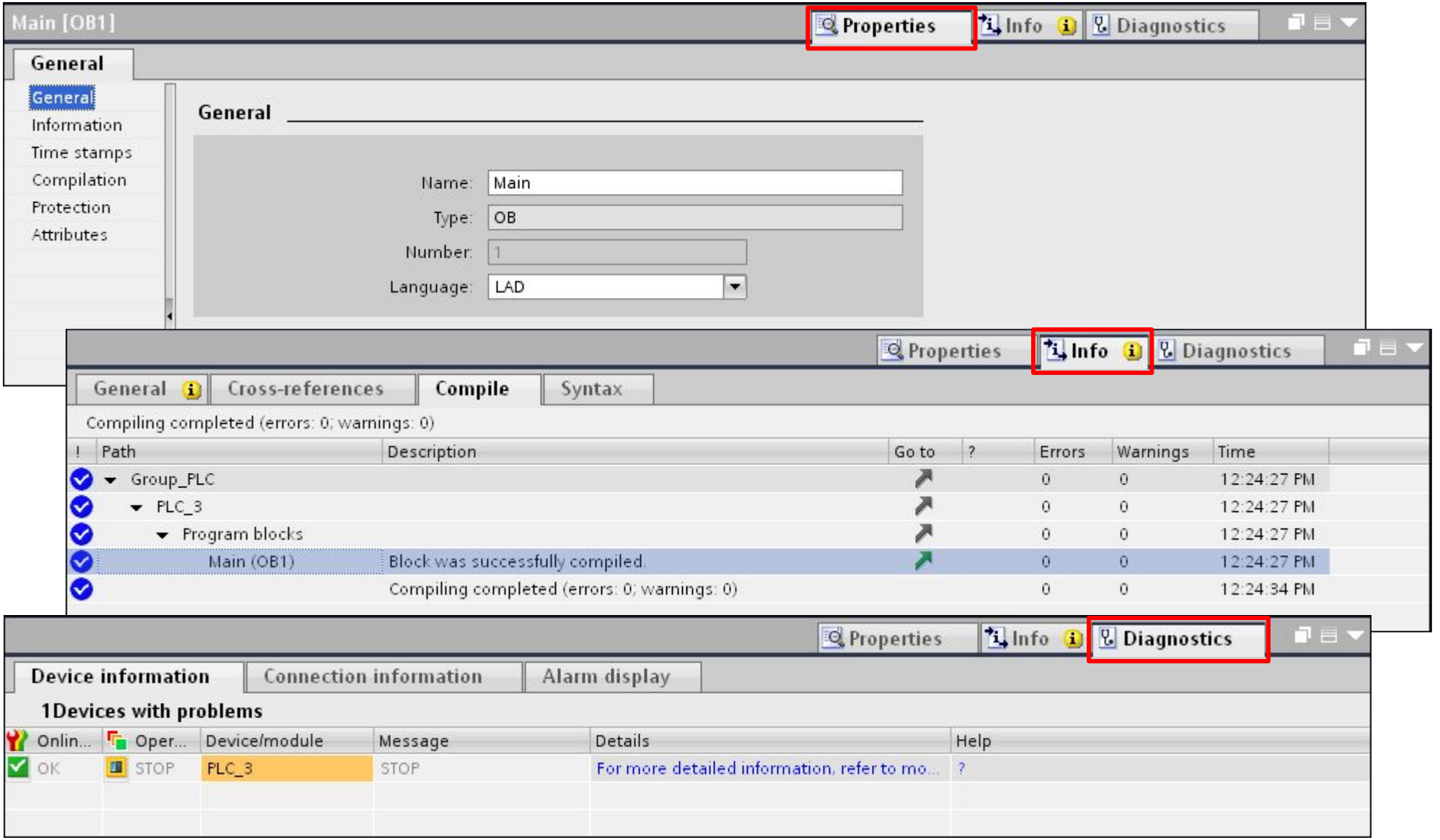


Для обновления версии ОС необходимо иметь ММС емкостью от 2 до 8 мБ

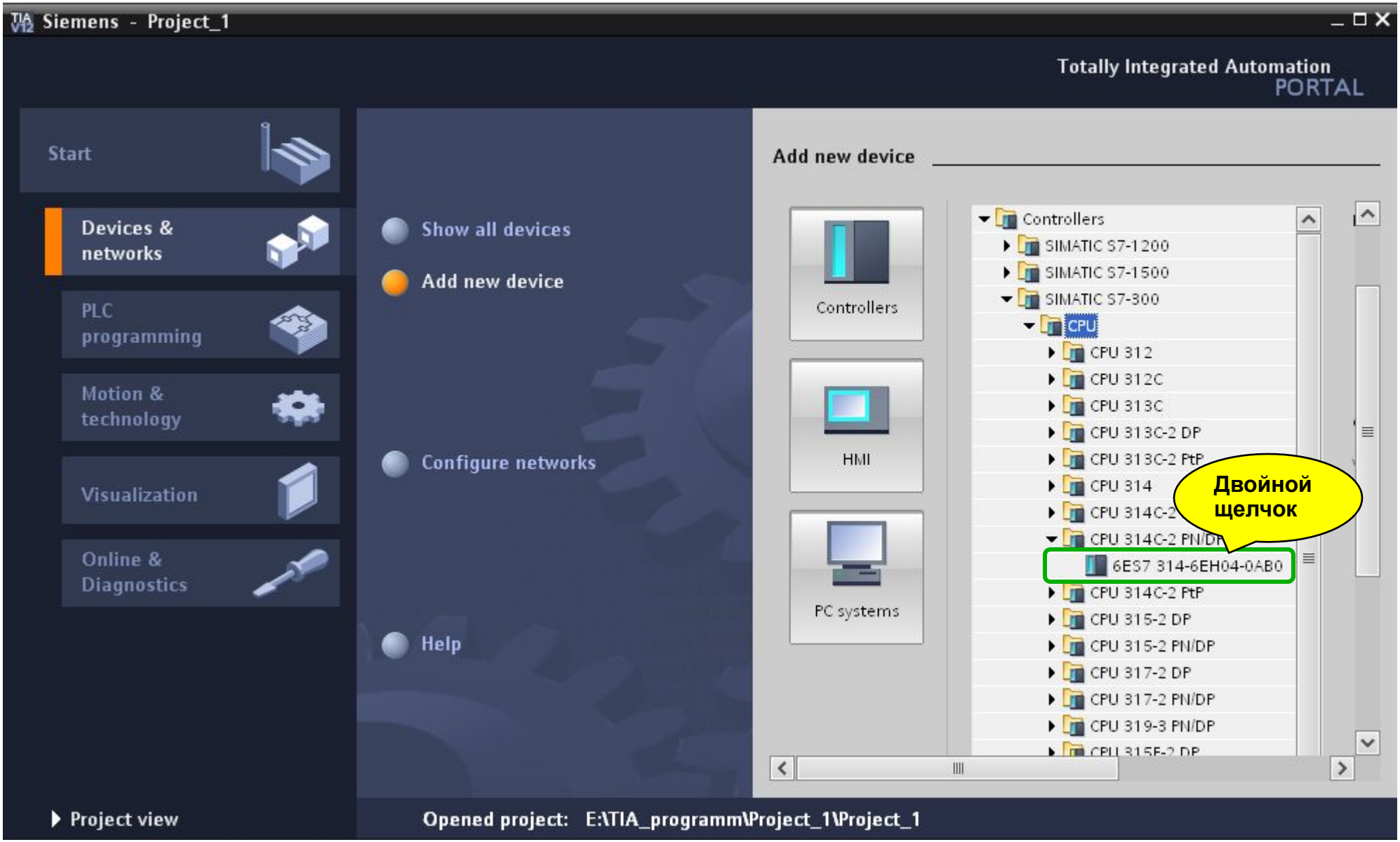
S7-300



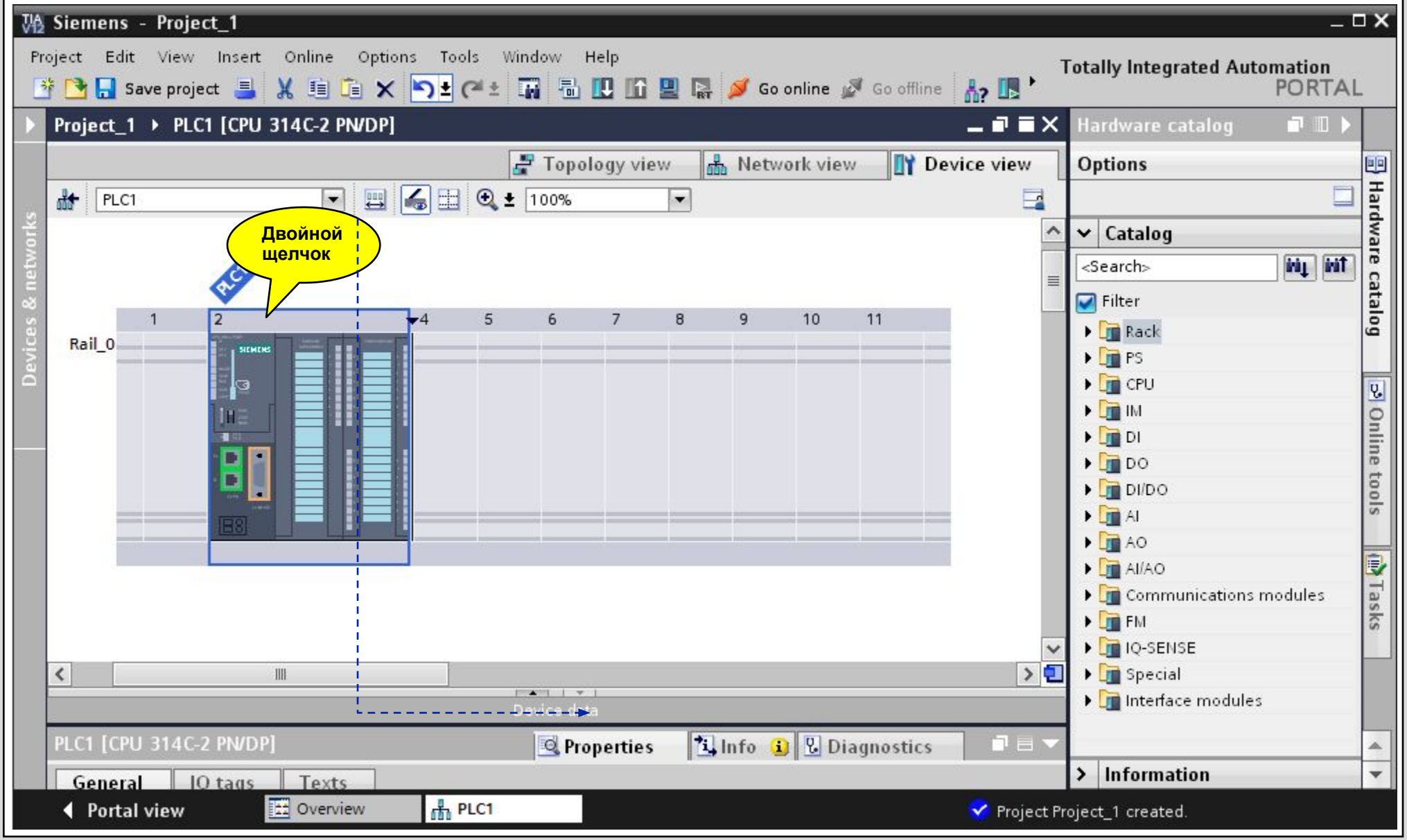
Разделы инспекционного окна (Inspector window)



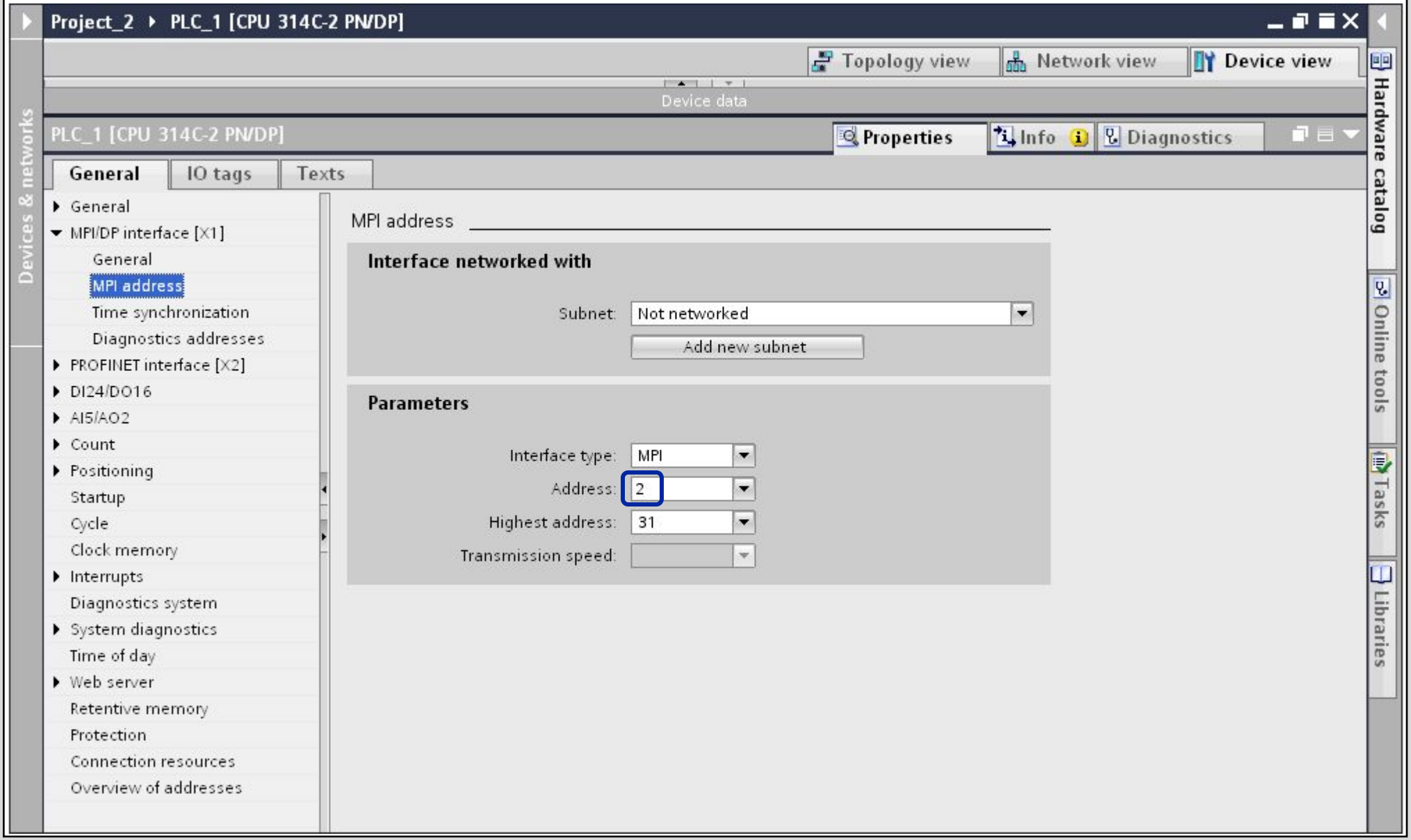
Пример выбора процессора в окне Portal view



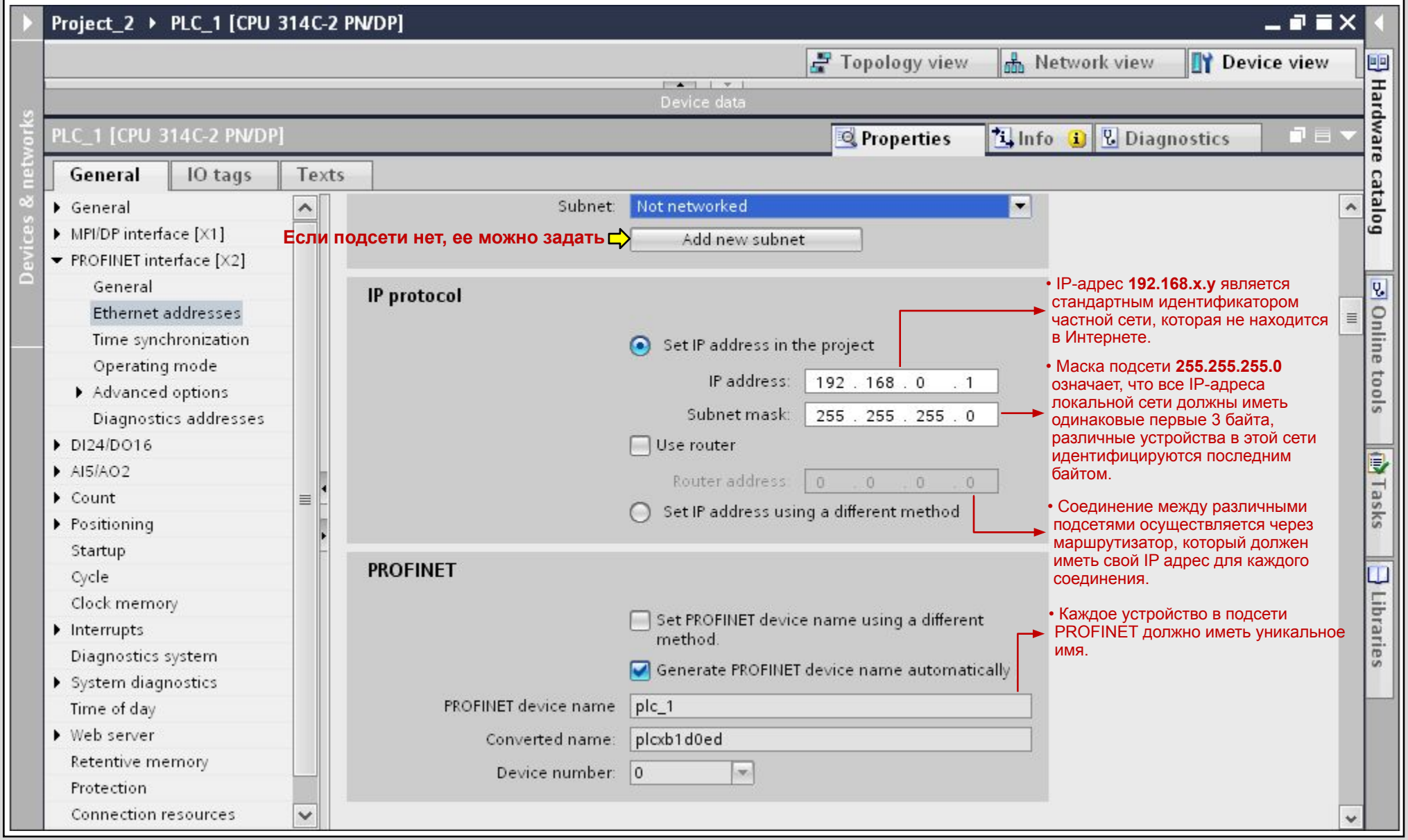
Активизация окна свойств выбранного процессора



Свойства ЦПУ. MPI-адрес

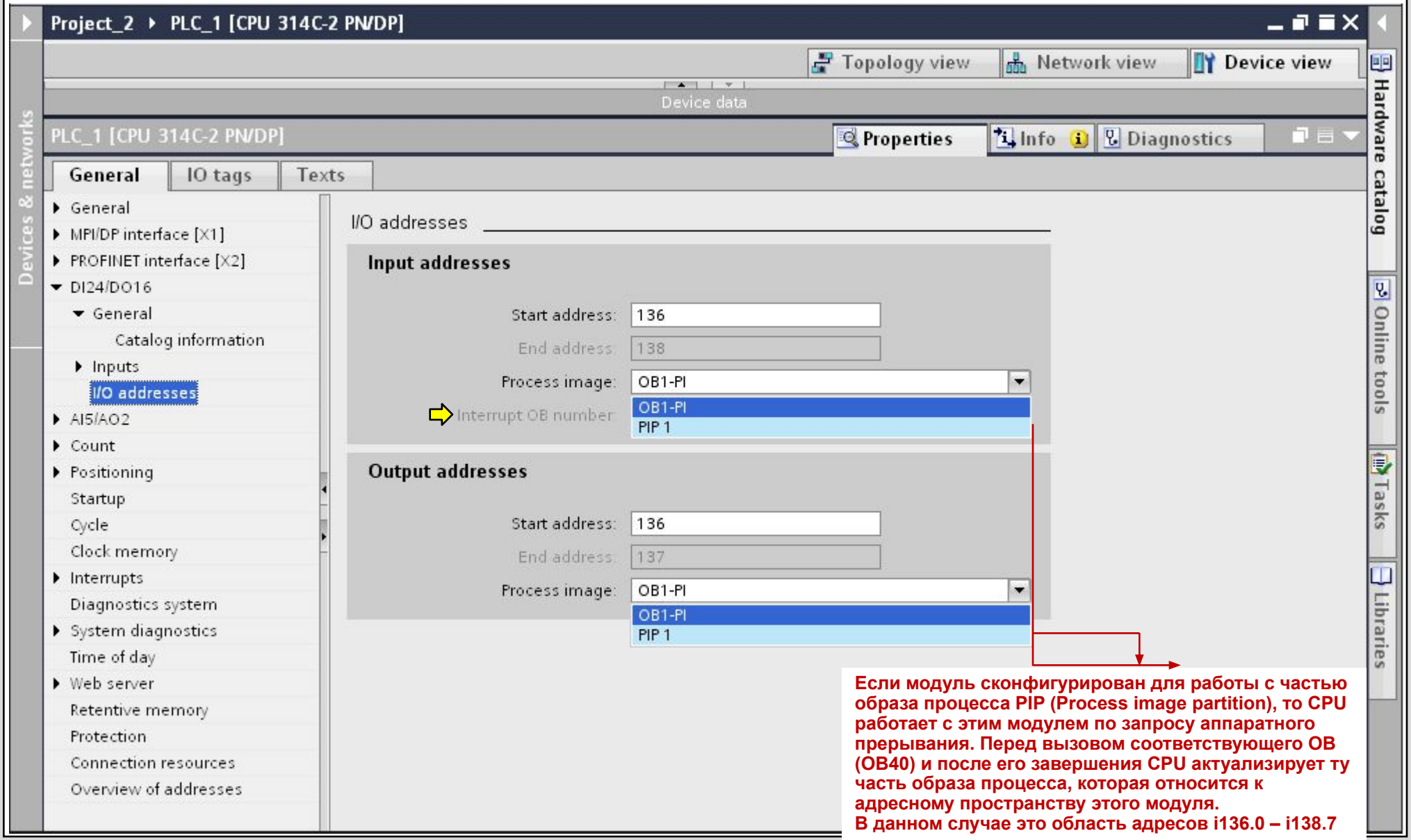


Свойства ЦПУ. Ethernet-адрес



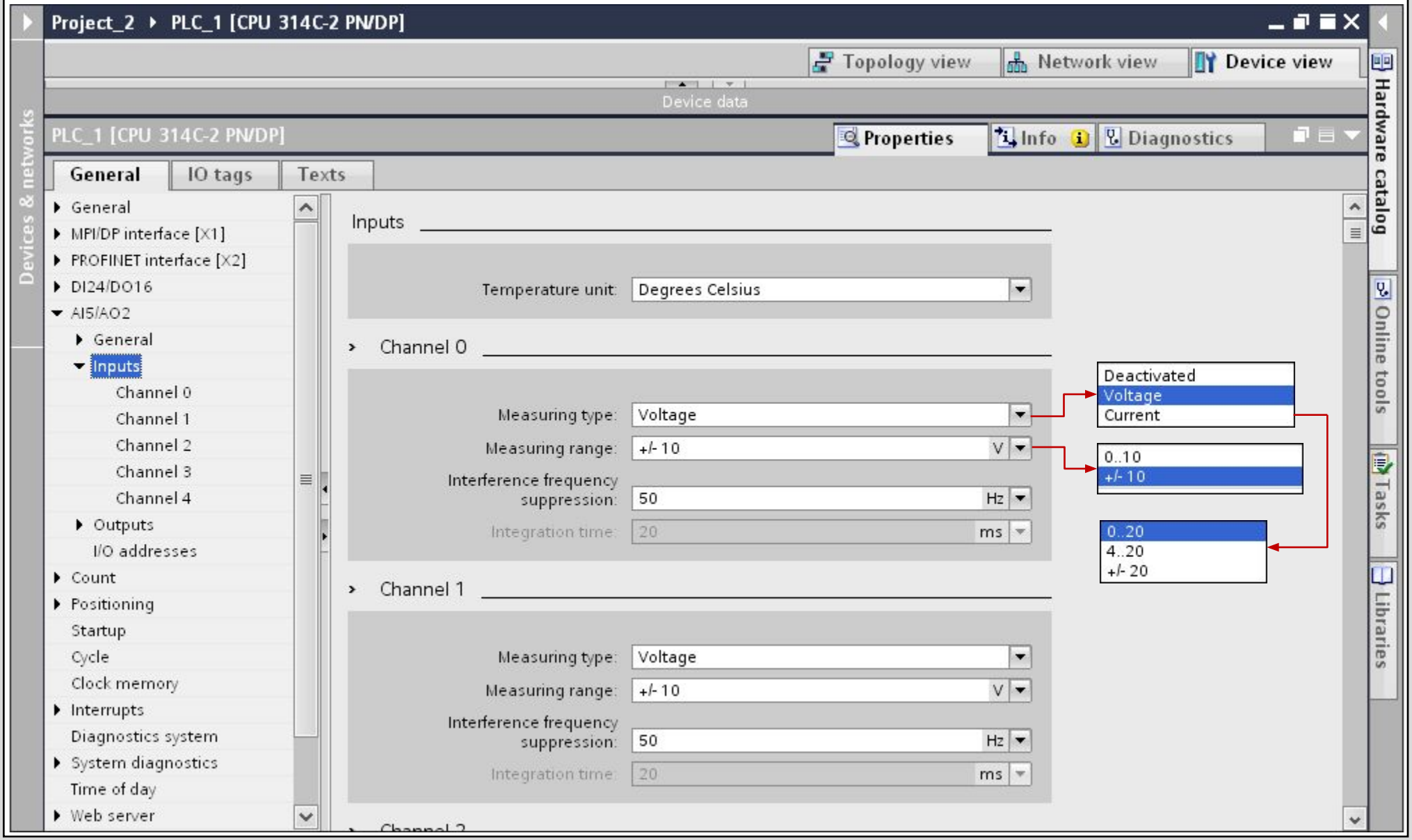
- IP-адрес **192.168.x.y** является стандартным идентификатором частной сети, которая не находится в Интернете.
- Маска подсети **255.255.255.0** означает, что все IP-адреса локальной сети должны иметь одинаковые первые 3 байта, различные устройства в этой сети идентифицируются последним байтом.
- Соединение между различными подсетями осуществляется через маршрутизатор, который должен иметь свой IP адрес для каждого соединения.
- Каждое устройство в подсети PROFINET должно иметь уникальное имя.

Свойства ЦПУ. Адресация встроенных дискретных вх/вых.

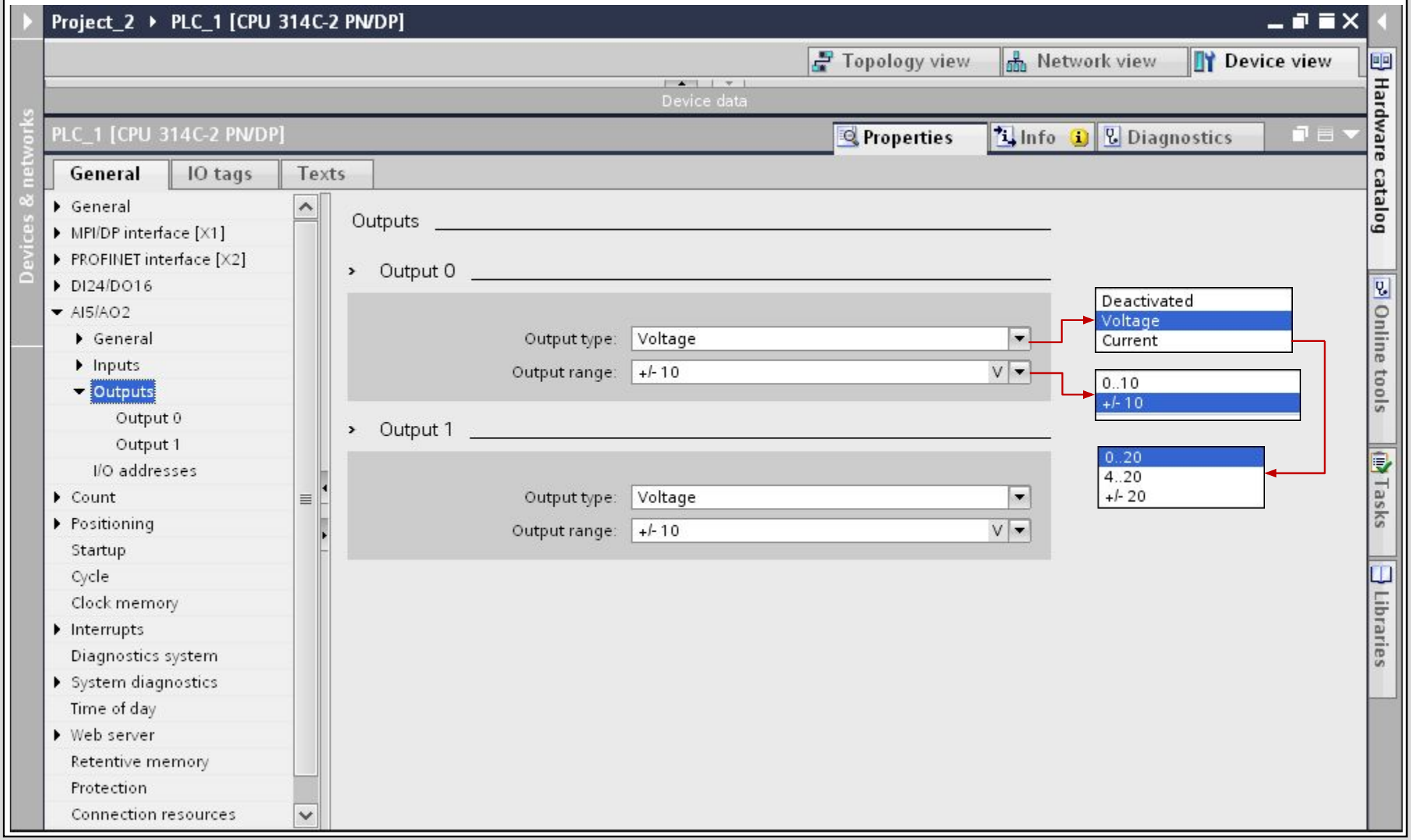


Если модуль сконфигурирован для работы с частью образа процесса PIP (Process image partition), то CPU работает с этим модулем по запросу аппаратного прерывания. Перед вызовом соответствующего OB (OB40) и после его завершения CPU актуализирует ту часть образа процесса, которая относится к адресному пространству этого модуля.
В данном случае это область адресов i136.0 – i138.7

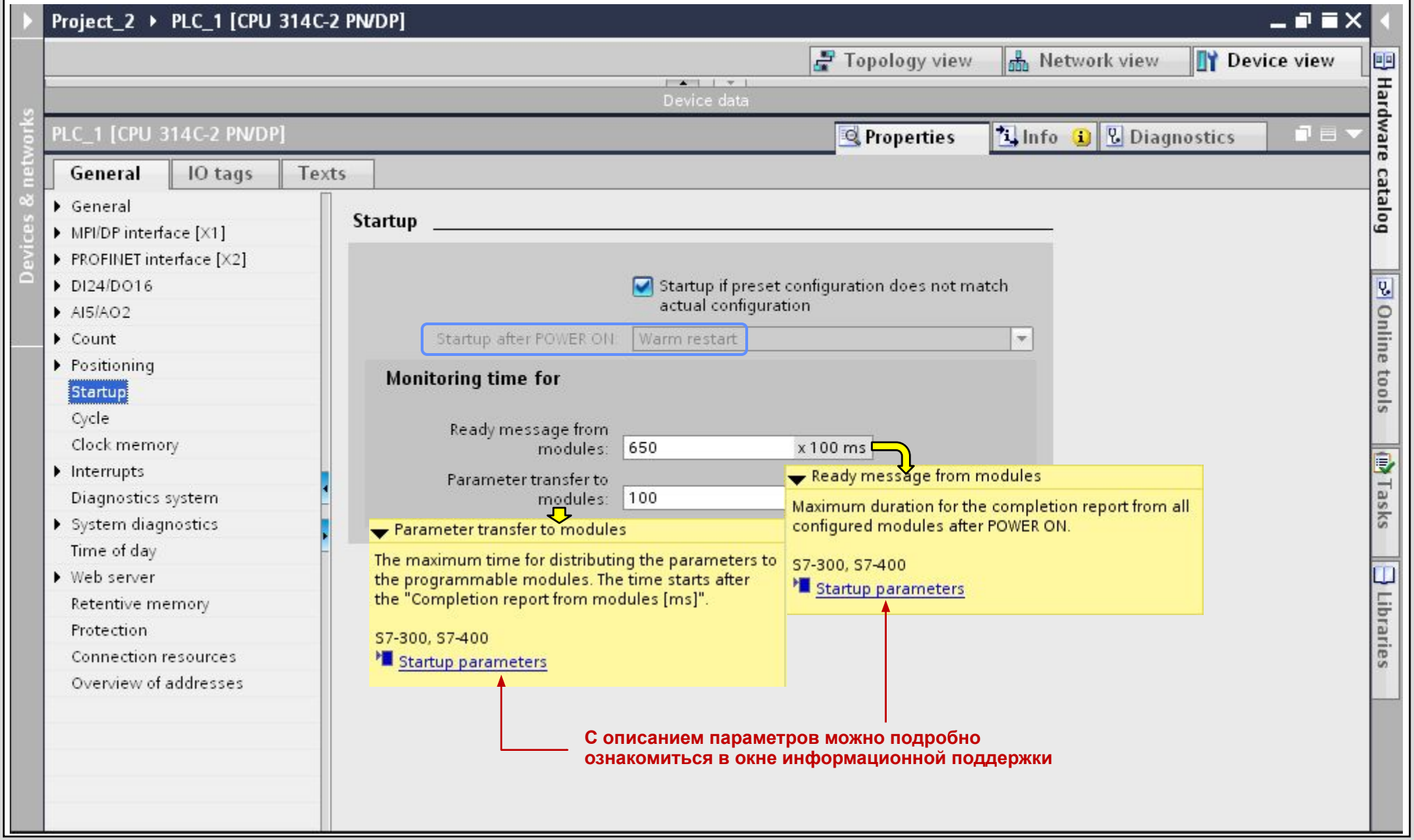
Свойства ЦПУ. Параметры встроенных аналоговых входов.



Свойства ЦПУ. Параметры встроенных аналоговых выходов.

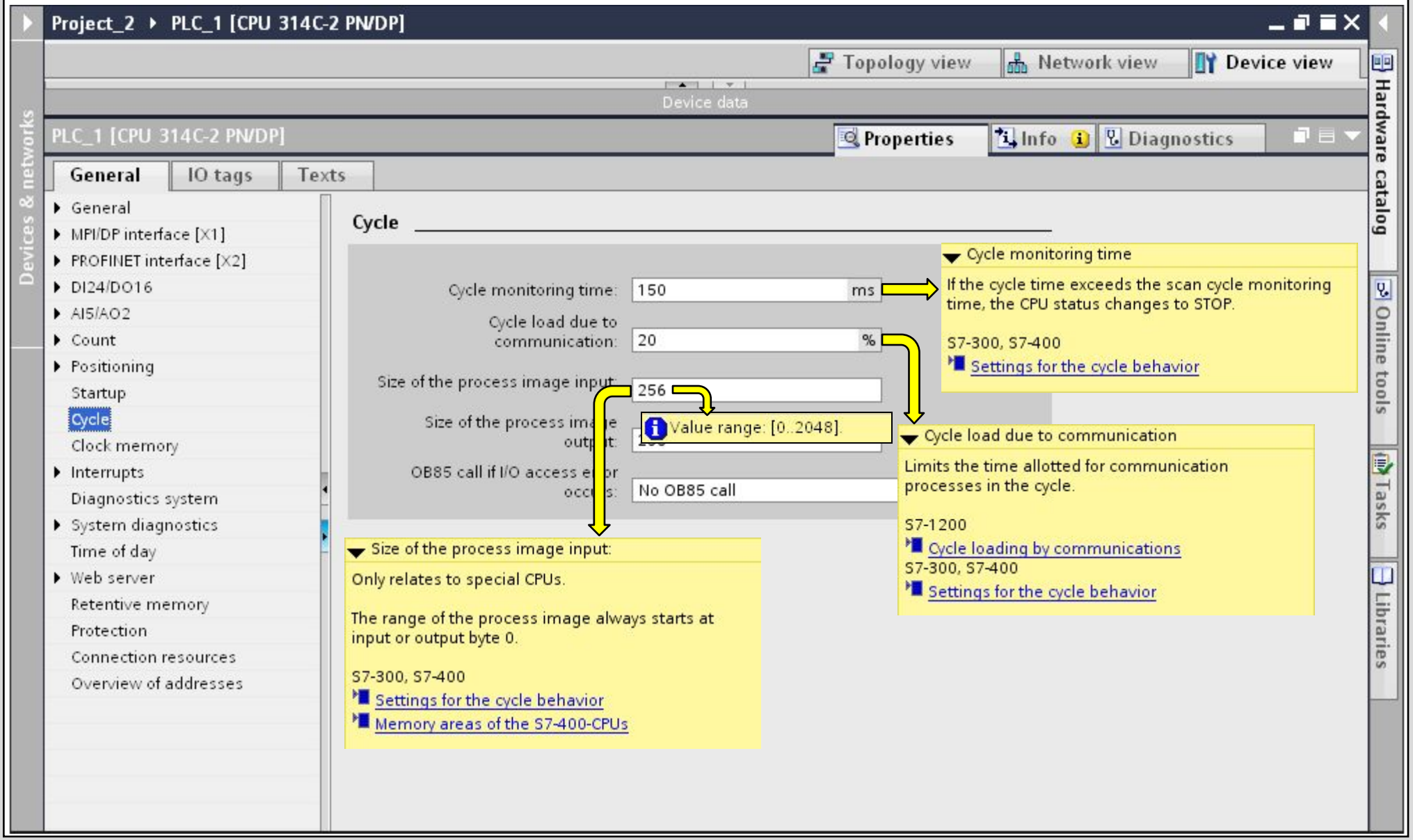


Свойства ЦПУ. Параметры старта.

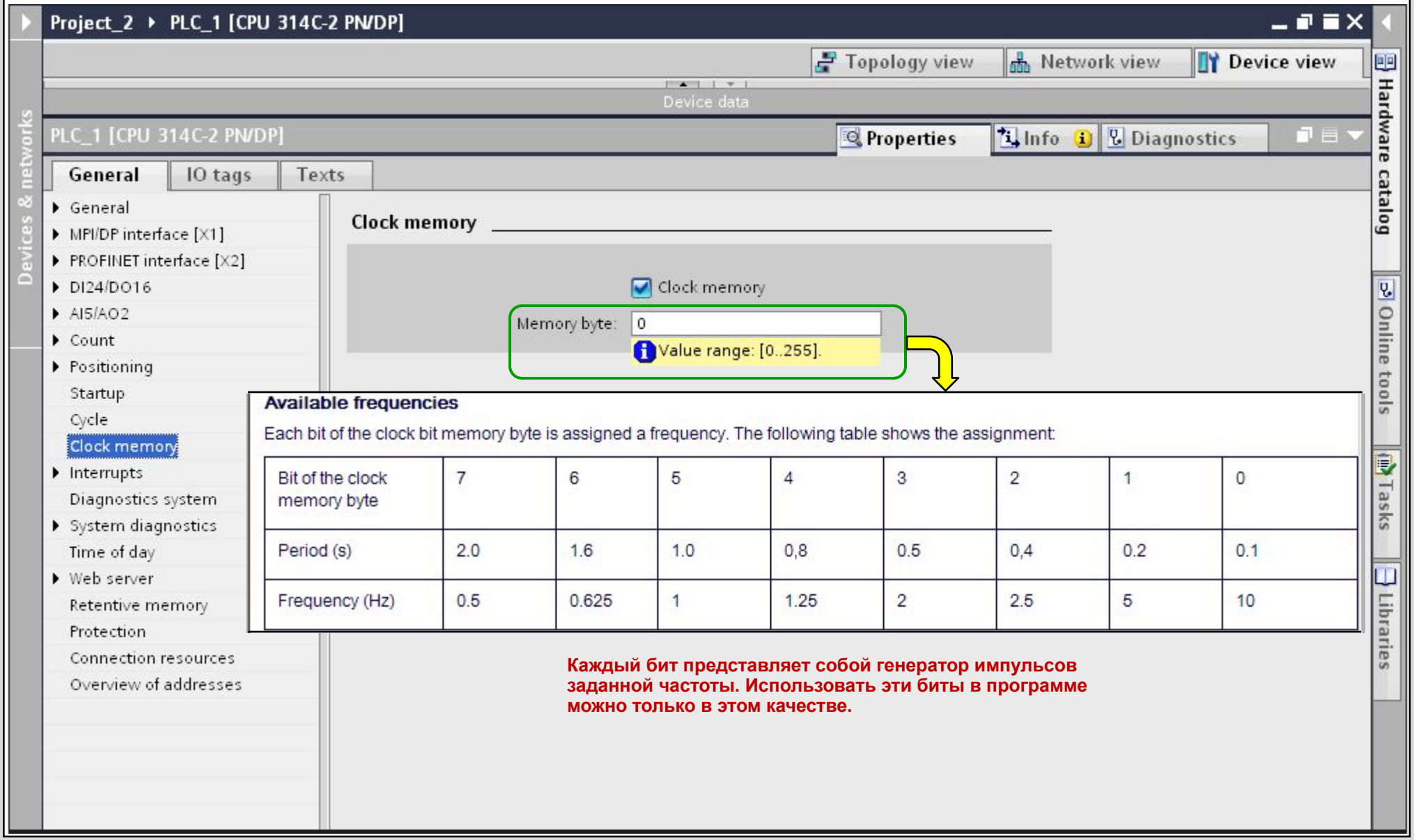


С описанием параметров можно подробно ознакомиться в окне информационной поддержки

Свойства ЦПУ. Время цикла.



Свойства ЦПУ. Байт в качестве побитных генераторов импульсов.



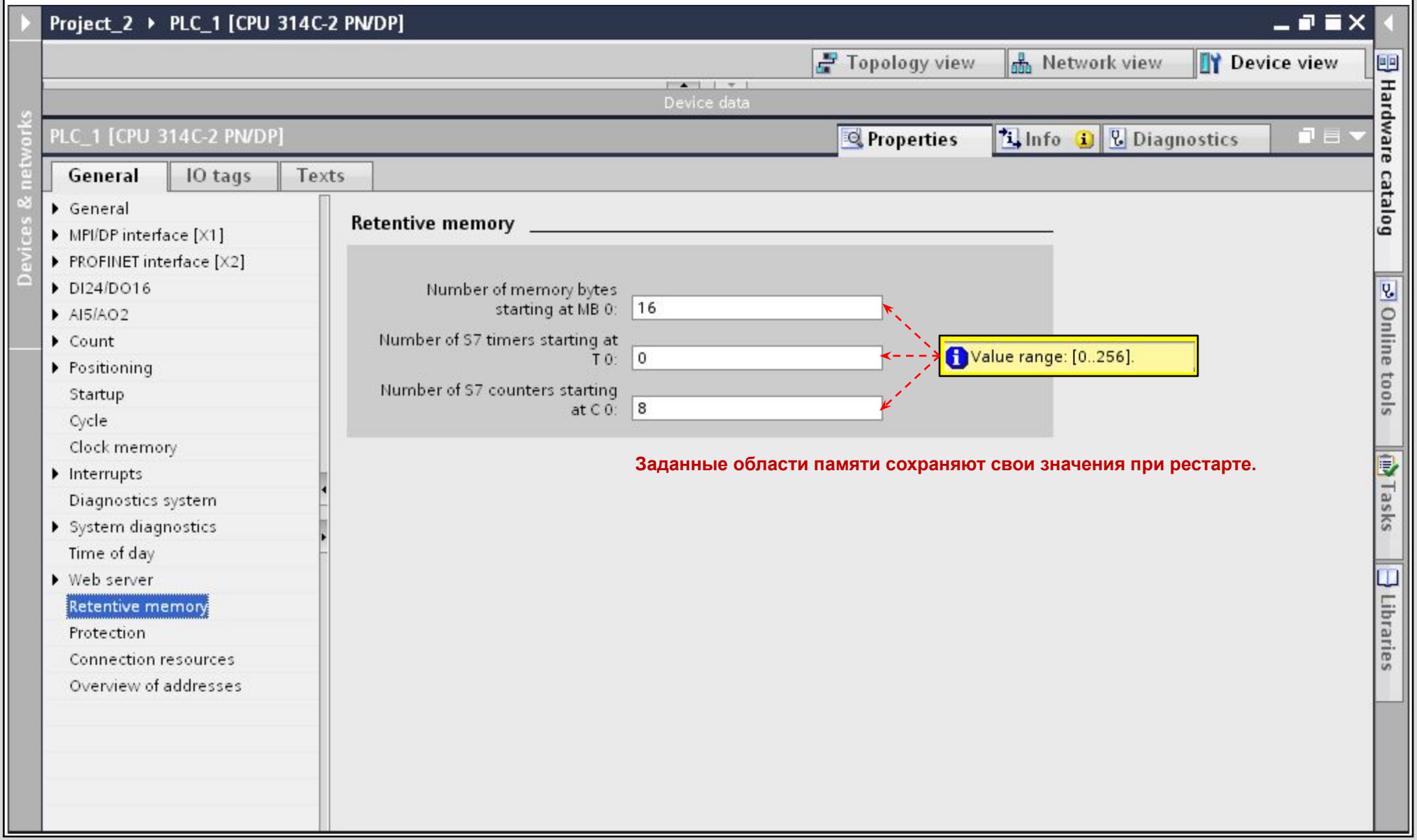
Available frequencies

Each bit of the clock bit memory byte is assigned a frequency. The following table shows the assignment:

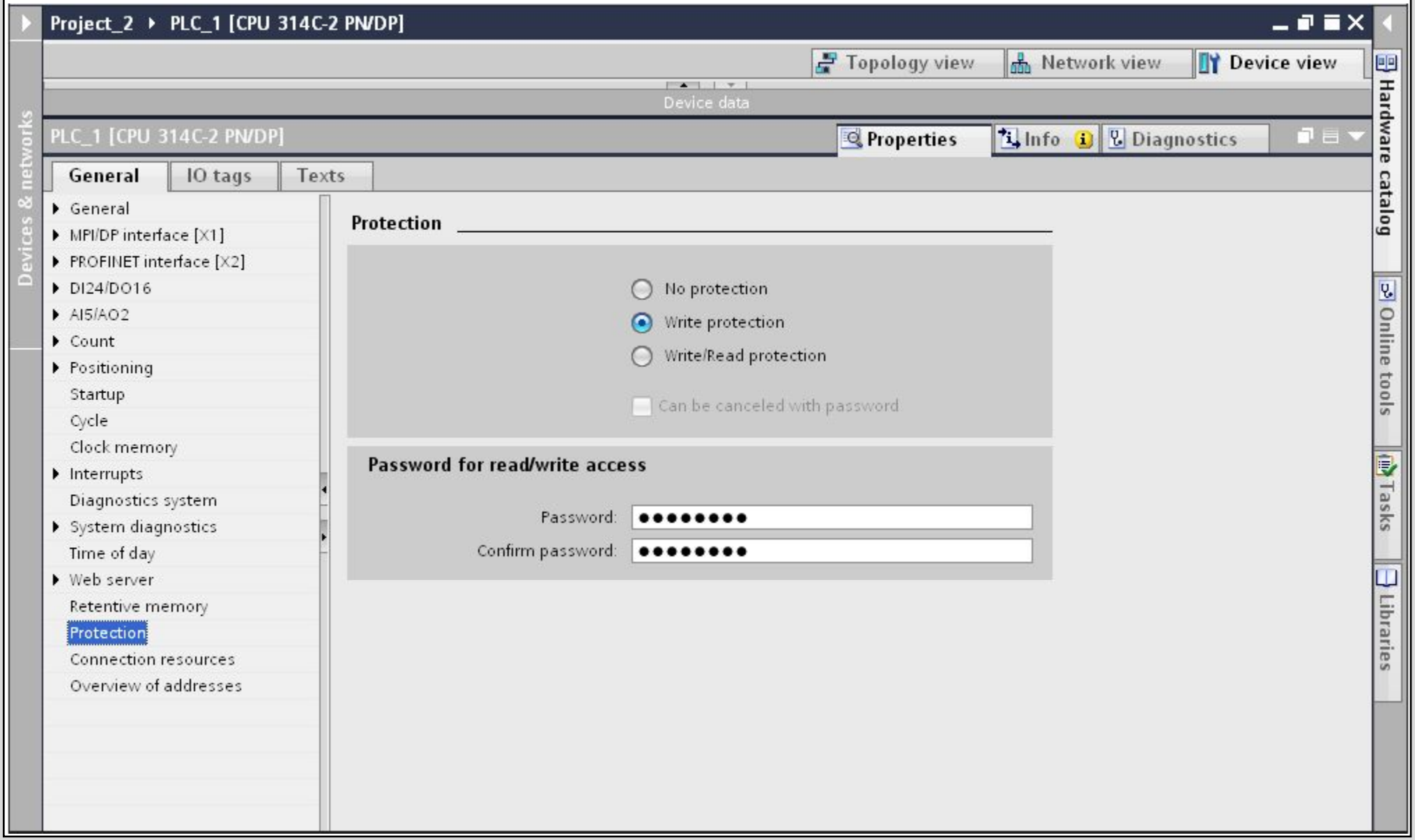
Bit of the clock memory byte	7	6	5	4	3	2	1	0
Period (s)	2.0	1.6	1.0	0,8	0.5	0,4	0.2	0.1
Frequency (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

Каждый бит представляет собой генератор импульсов заданной частоты. Использовать эти биты в программе можно только в этом качестве.

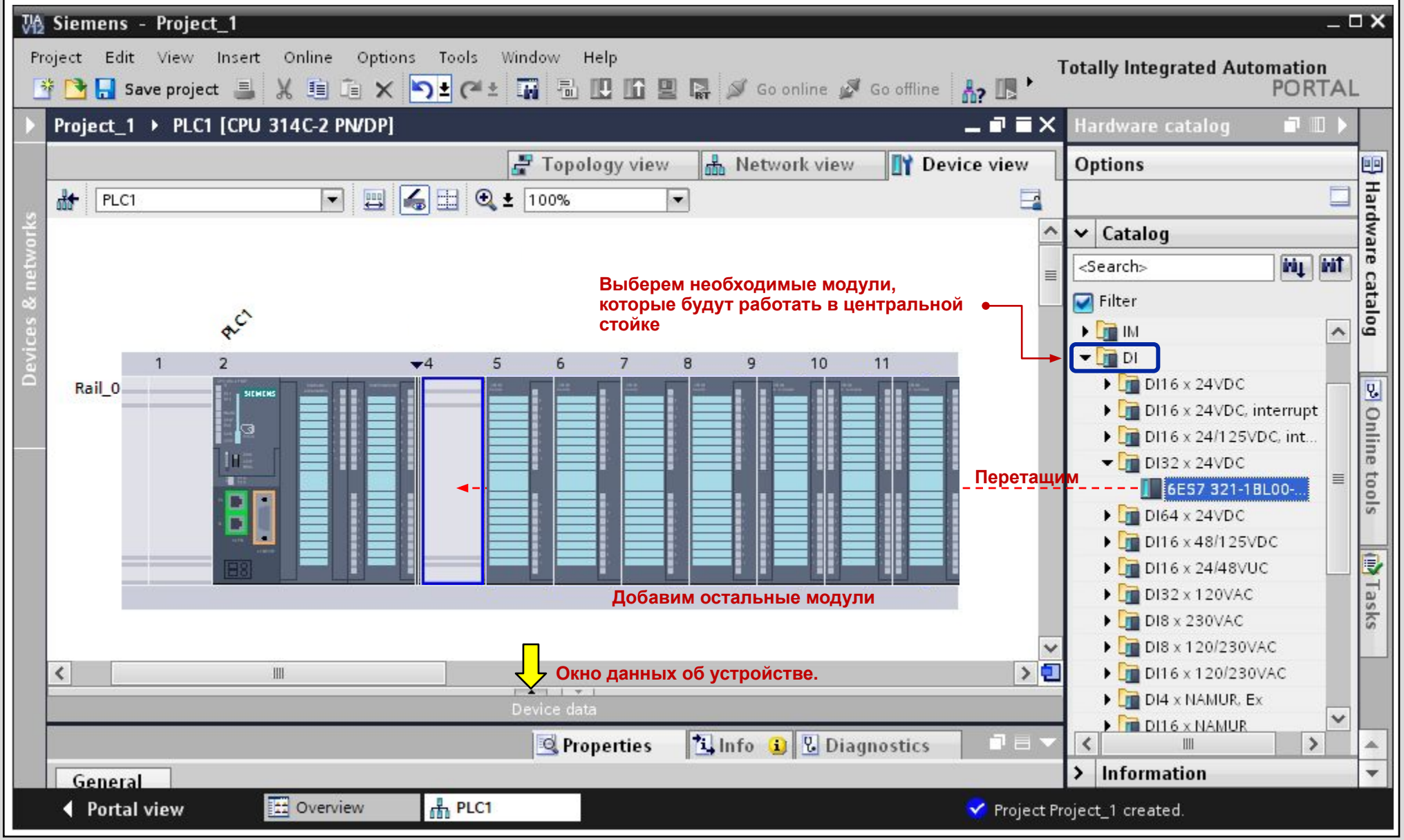
Свойства ЦПУ. Сохраняемые области при рестарте.



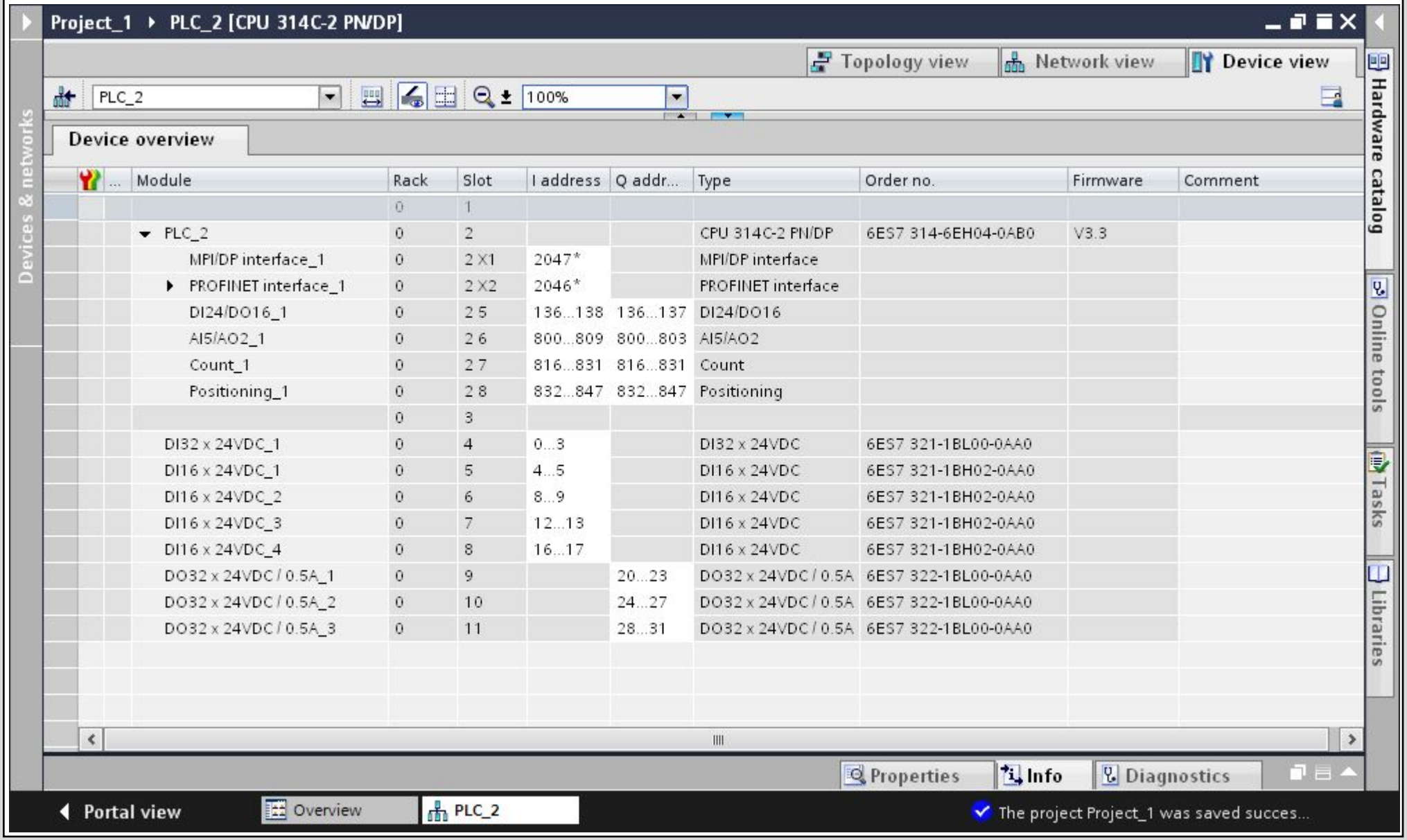
Свойства ЦПУ. Защита паролем.



Выбор модулей центральной стойки

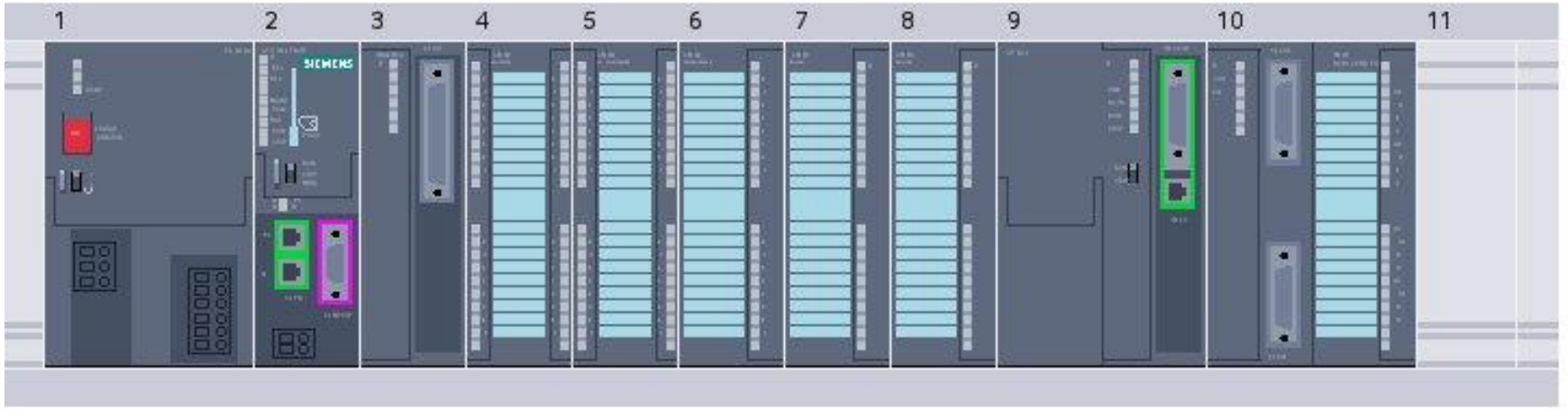


Окно просмотра данных об устройстве



...	Module	Rack	Slot	I address	Q addr...	Type	Order no.	Firmware	Comment
		0	1						
	PLC_2	0	2			CPU 314C-2 PN/DP	6ES7 314-6EH04-0AB0	V3.3	
	MPI/DP interface_1	0	2 X1	2047*		MPI/DP interface			
	PROFINET interface_1	0	2 X2	2046*		PROFINET interface			
	DI24/DO16_1	0	2 5	136...138	136...137	DI24/DO16			
	AI5/AO2_1	0	2 6	800...809	800...803	AI5/AO2			
	Count_1	0	2 7	816...831	816...831	Count			
	Positioning_1	0	2 8	832...847	832...847	Positioning			
		0	3						
	DI32 x 24VDC_1	0	4	0...3		DI32 x 24VDC	6ES7 321-1BL00-0AA0		
	DI16 x 24VDC_1	0	5	4...5		DI16 x 24VDC	6ES7 321-1BH02-0AA0		
	DI16 x 24VDC_2	0	6	8...9		DI16 x 24VDC	6ES7 321-1BH02-0AA0		
	DI16 x 24VDC_3	0	7	12...13		DI16 x 24VDC	6ES7 321-1BH02-0AA0		
	DI16 x 24VDC_4	0	8	16...17		DI16 x 24VDC	6ES7 321-1BH02-0AA0		
	DO32 x 24VDC / 0.5A_1	0	9		20...23	DO32 x 24VDC / 0.5A	6ES7 322-1BL00-0AA0		
	DO32 x 24VDC / 0.5A_2	0	10		24...27	DO32 x 24VDC / 0.5A	6ES7 322-1BL00-0AA0		
	DO32 x 24VDC / 0.5A_3	0	11		28...31	DO32 x 24VDC / 0.5A	6ES7 322-1BL00-0AA0		

Пример ряда модулей центральной стойки S7-300



PS
(опция)

CPU

IM
(опция)

SM:
DI

SM:
DO

SM:
AI

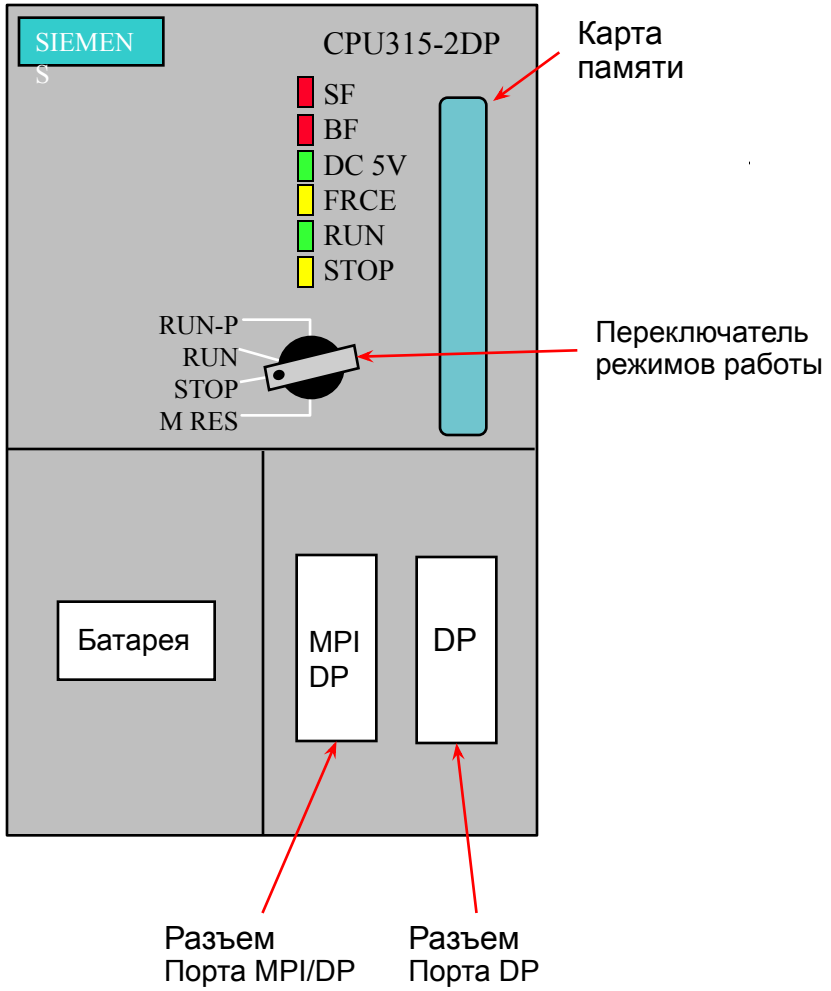
SM:
AO

CP:
- Point-to Point
- PROFIBUS
- PROFINET
- Industrial Ethernet

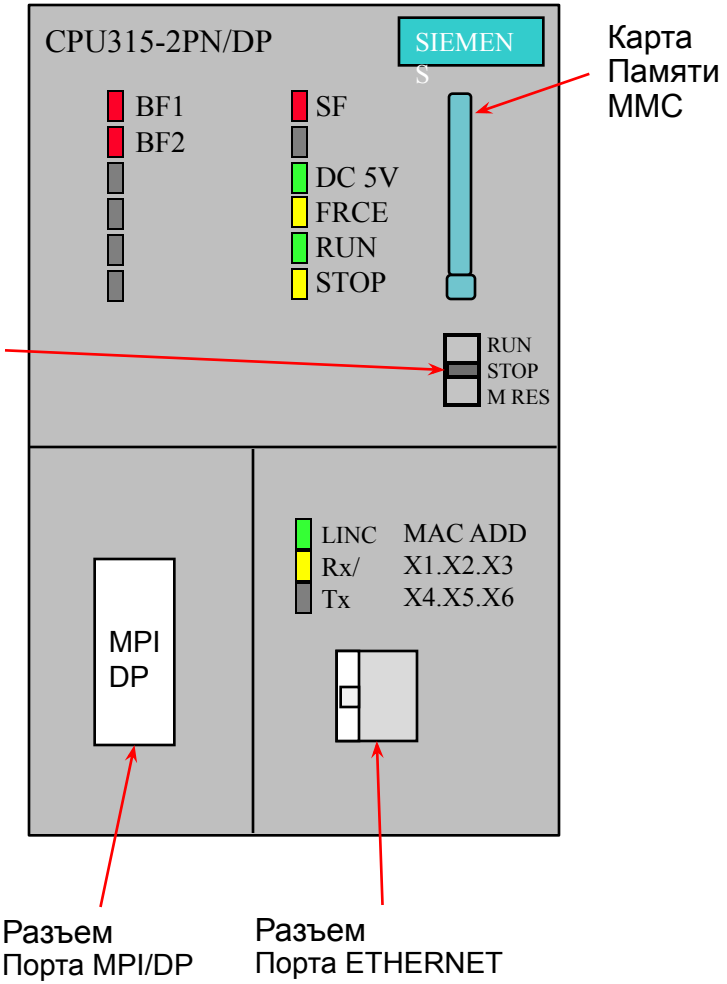
FM:
- счётчики
- позиционеры
- PID-регуляторы

Передняя панель CPU 315-2DP

CPU 315-2DP до октября 2002
(не поддерживается в TIA Portal)

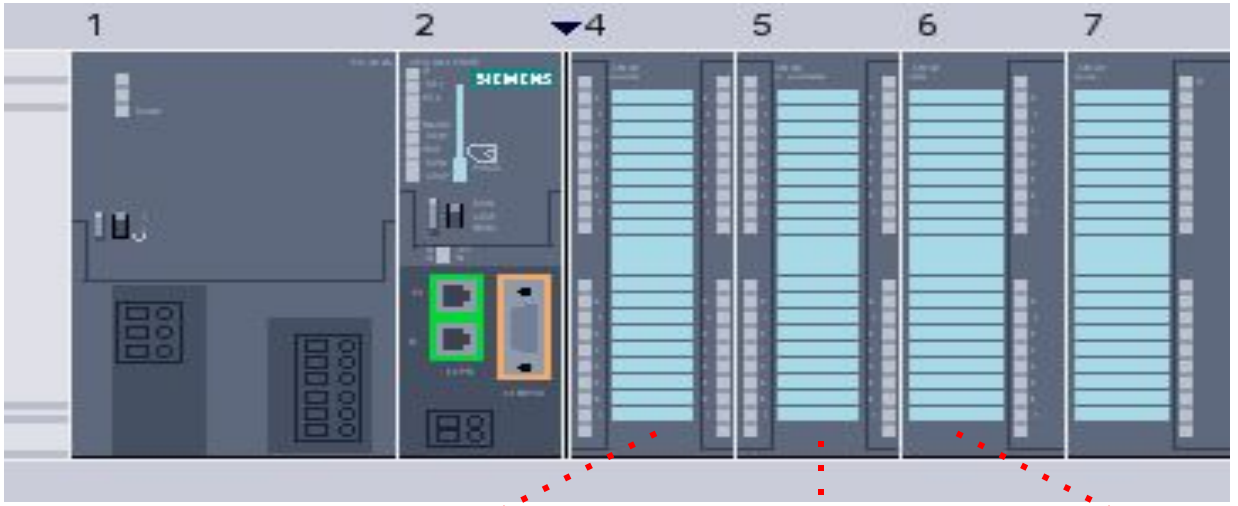


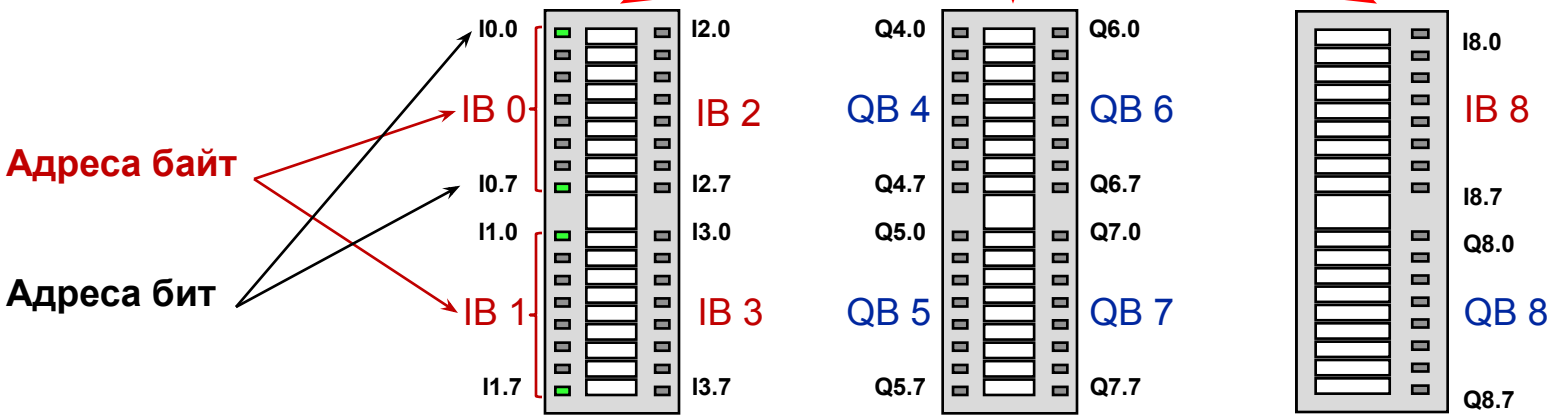
CPU 315-2PN/DP после октября 2002
(с V2.6 поддерживается в TIA Portal)



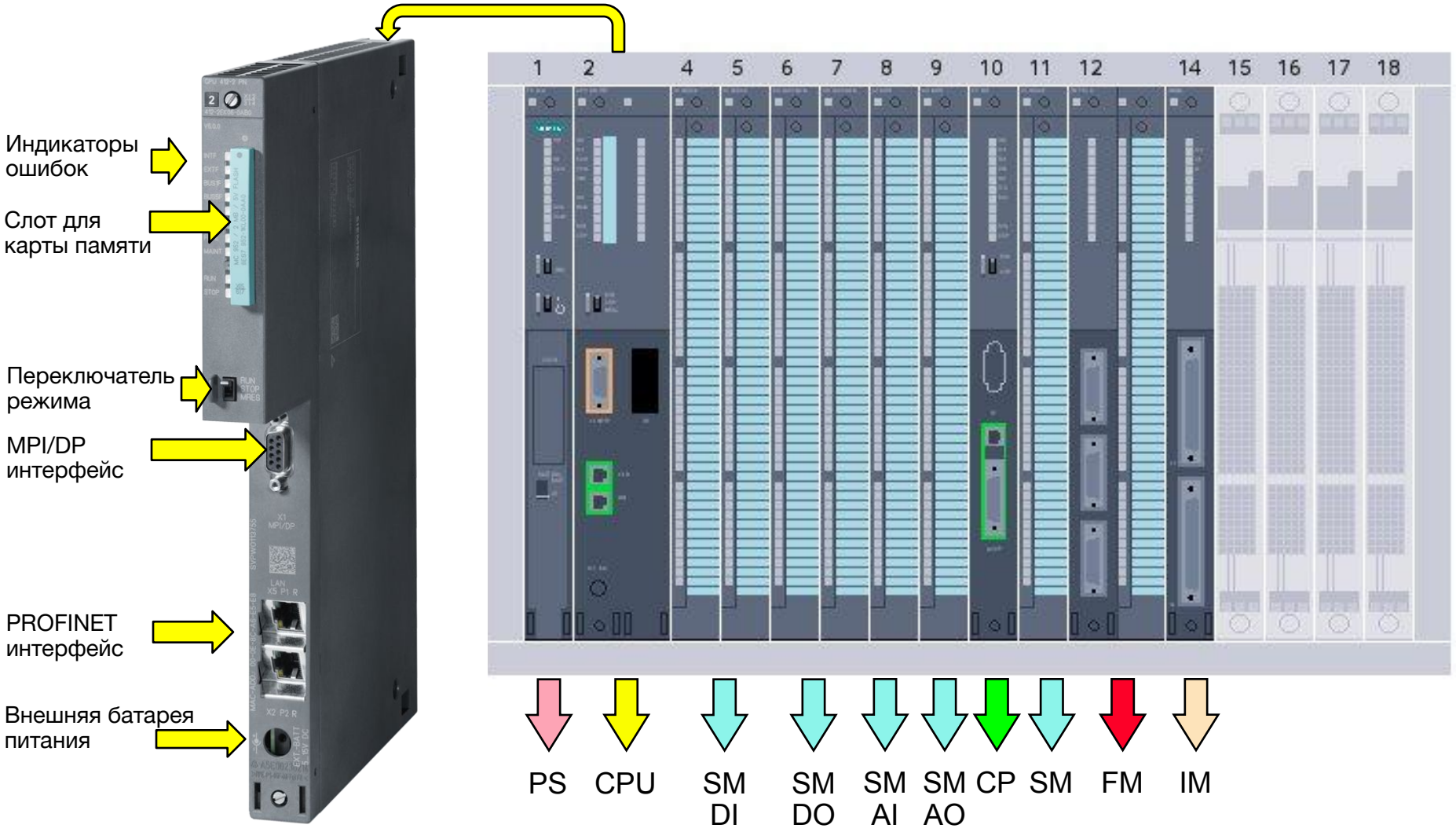
Пример адресации модулей центральной стойки S7-300

Слот №:	1	2	4	5	6	7
Модуль:	PS	CPU	DI 32	DO 32	DI8/DO8	AI 2

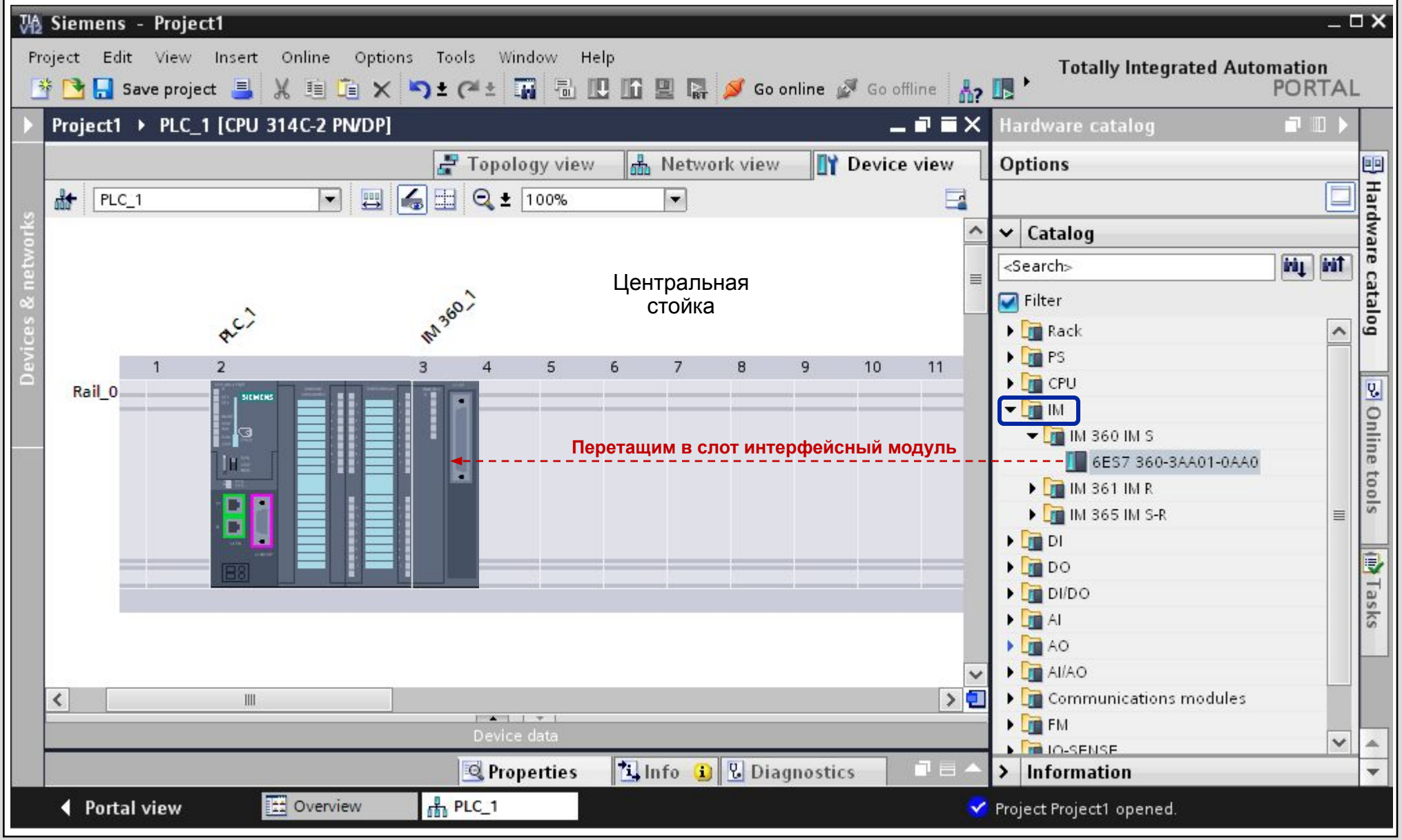




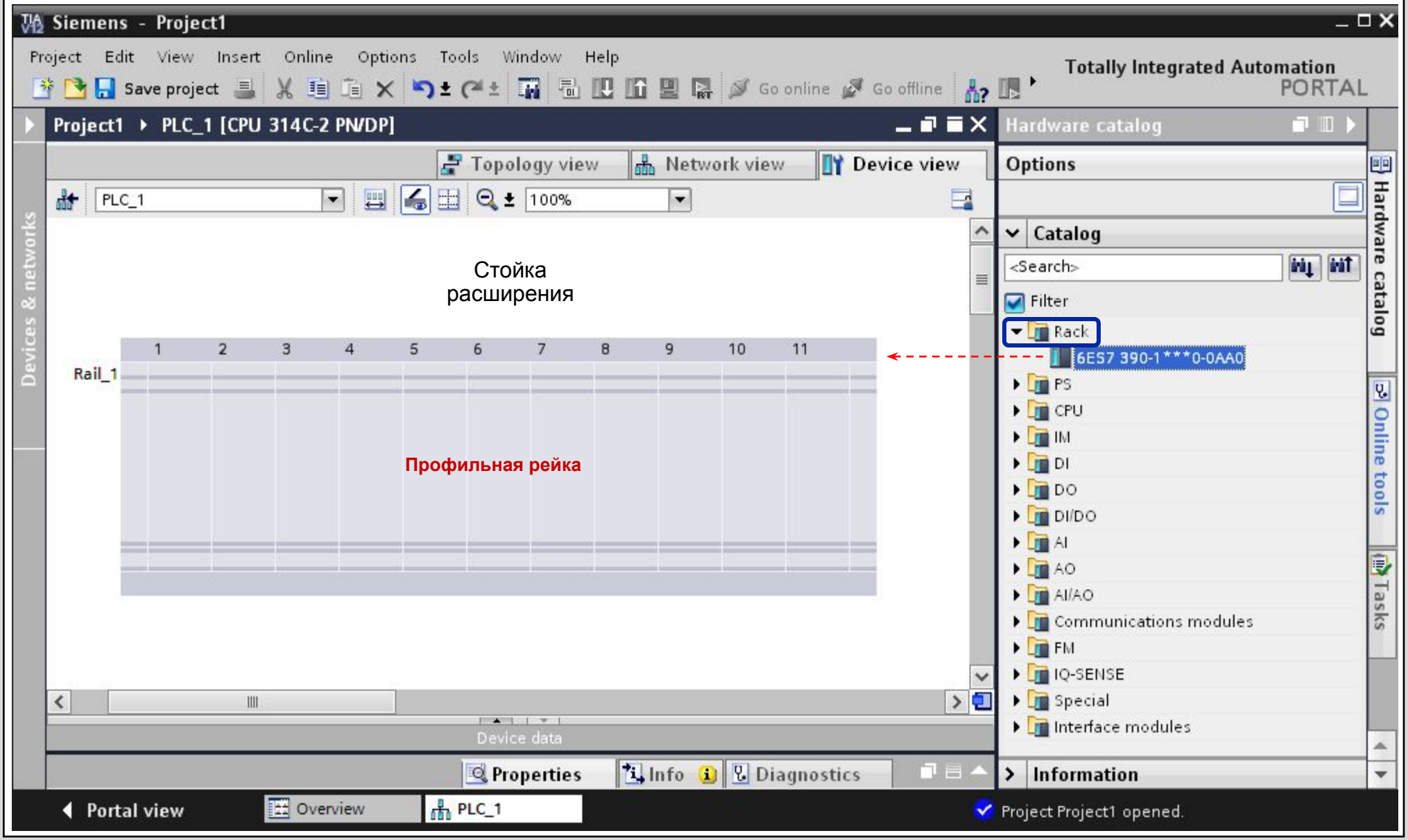
Пример ряда модулей центральной стойки S7-400



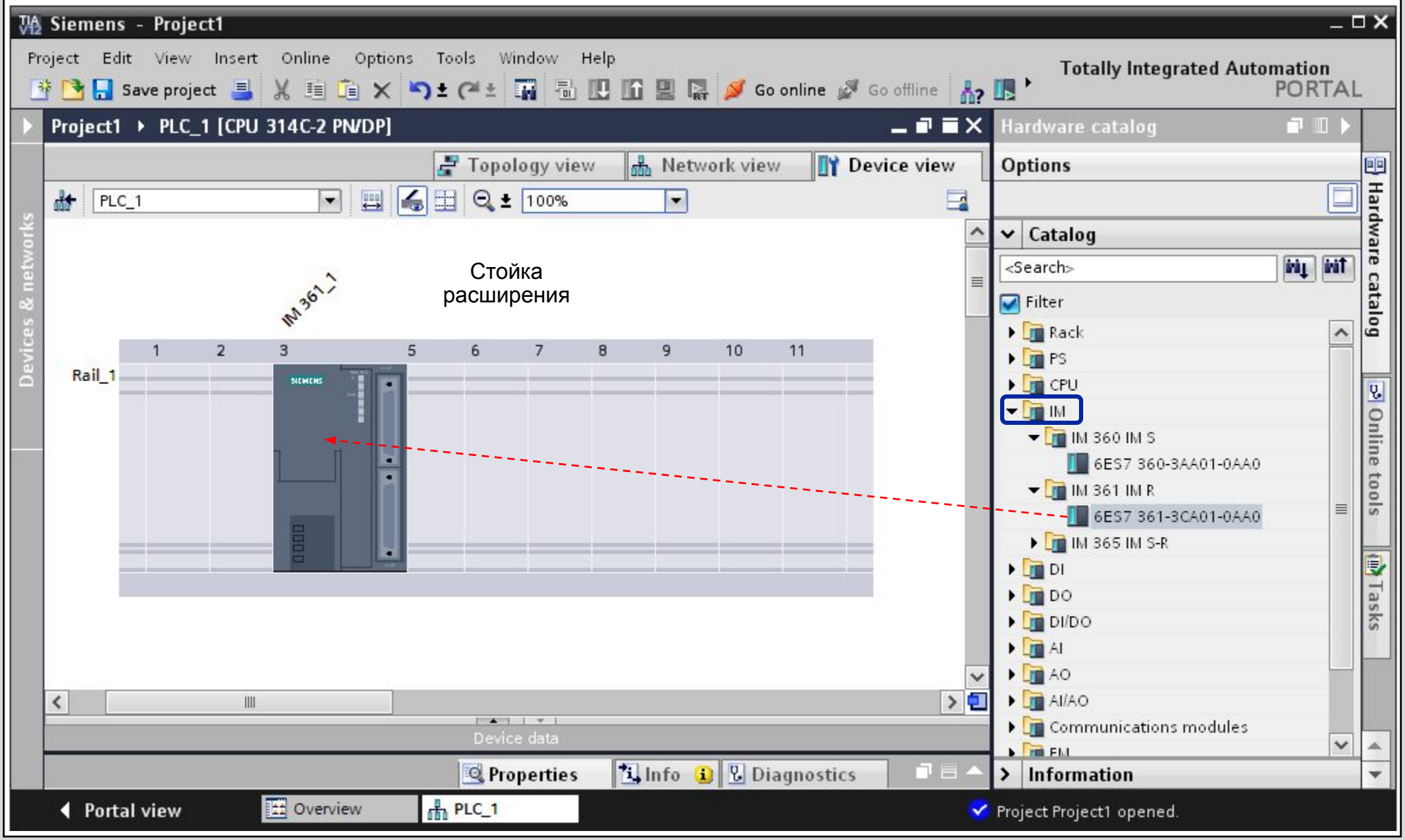
Пример многорядной конфигурации ПЛК с модулями IM360



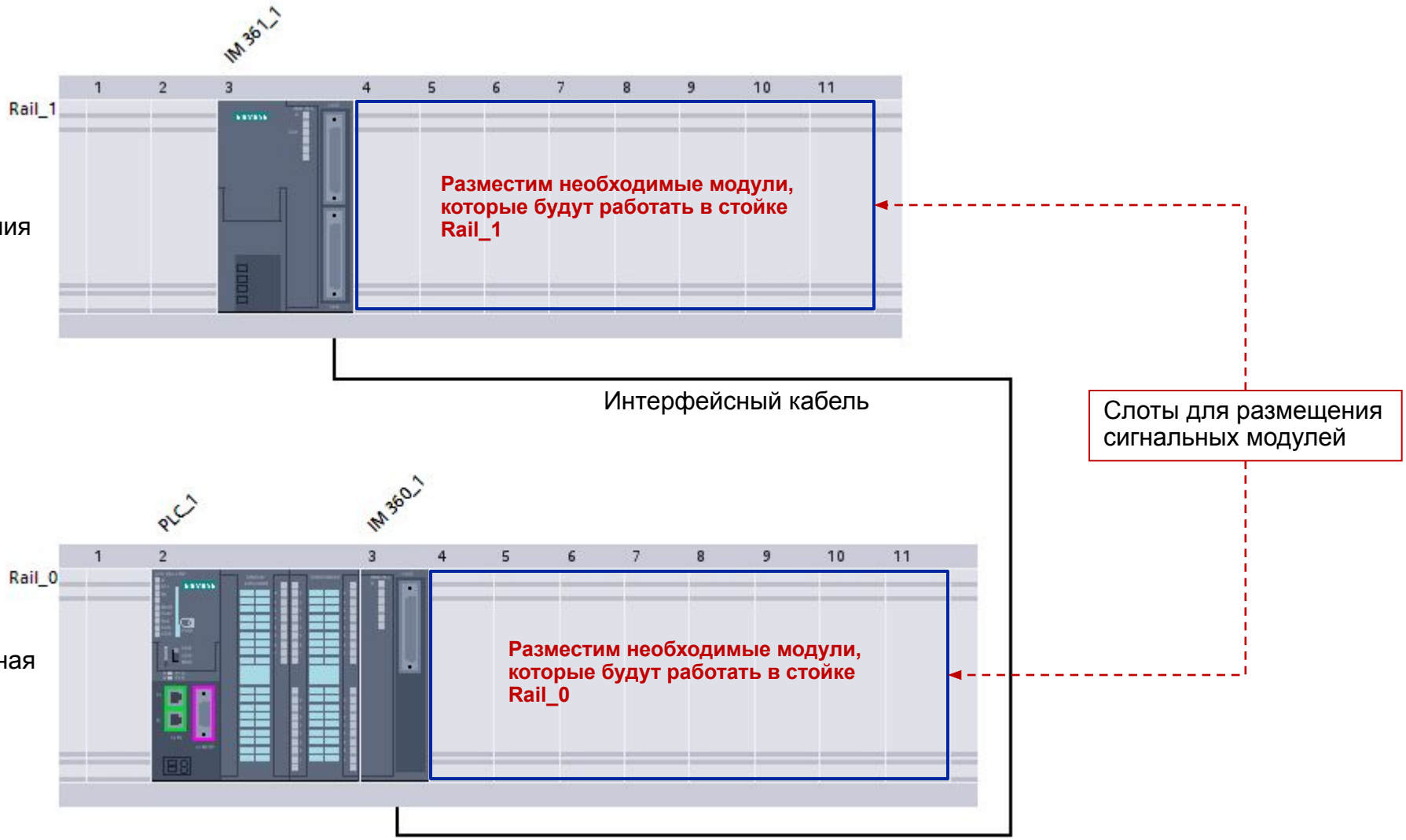
Пример многорядной конфигурации ПЛК с модулями IM360



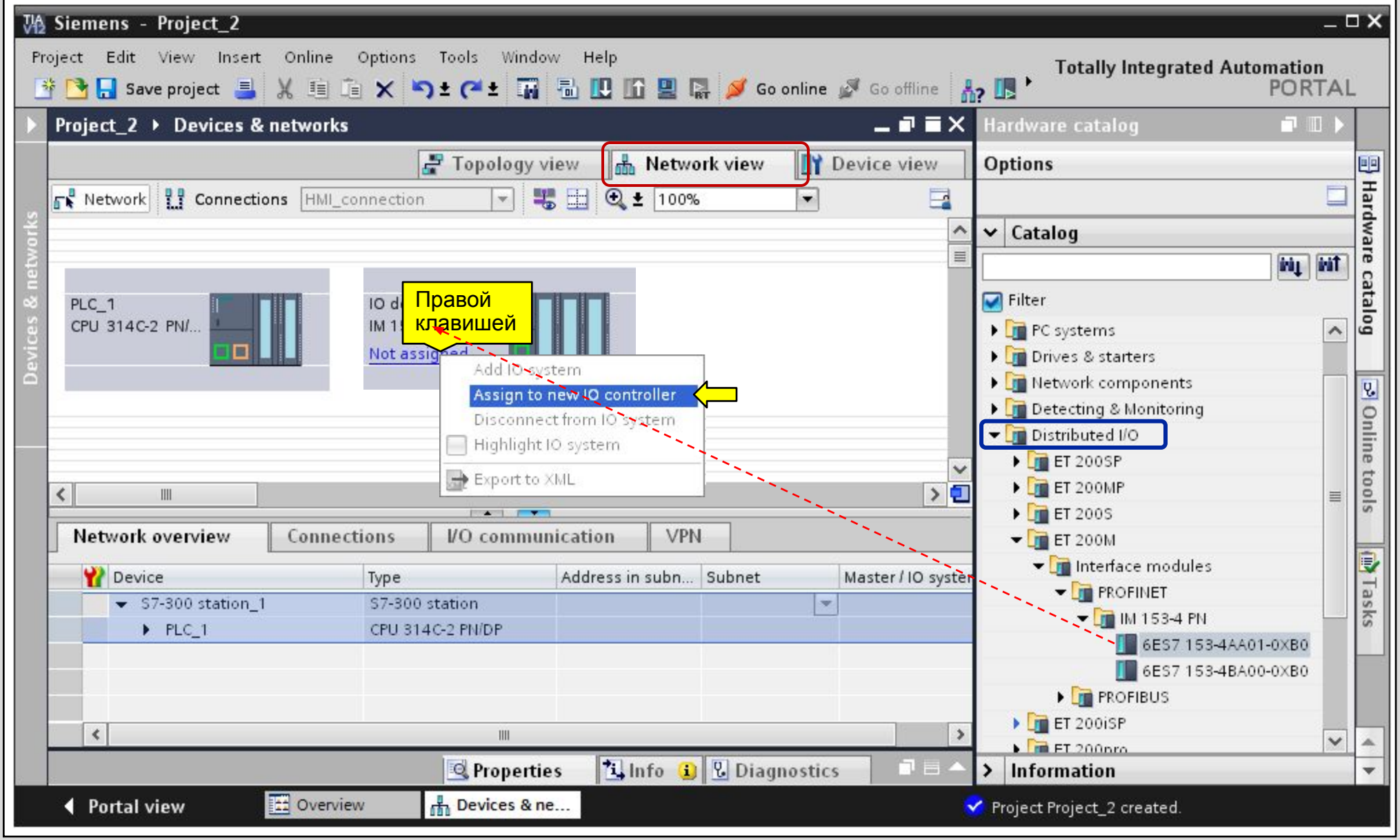
Пример многорядной конфигурации ПЛК с модулями IM360



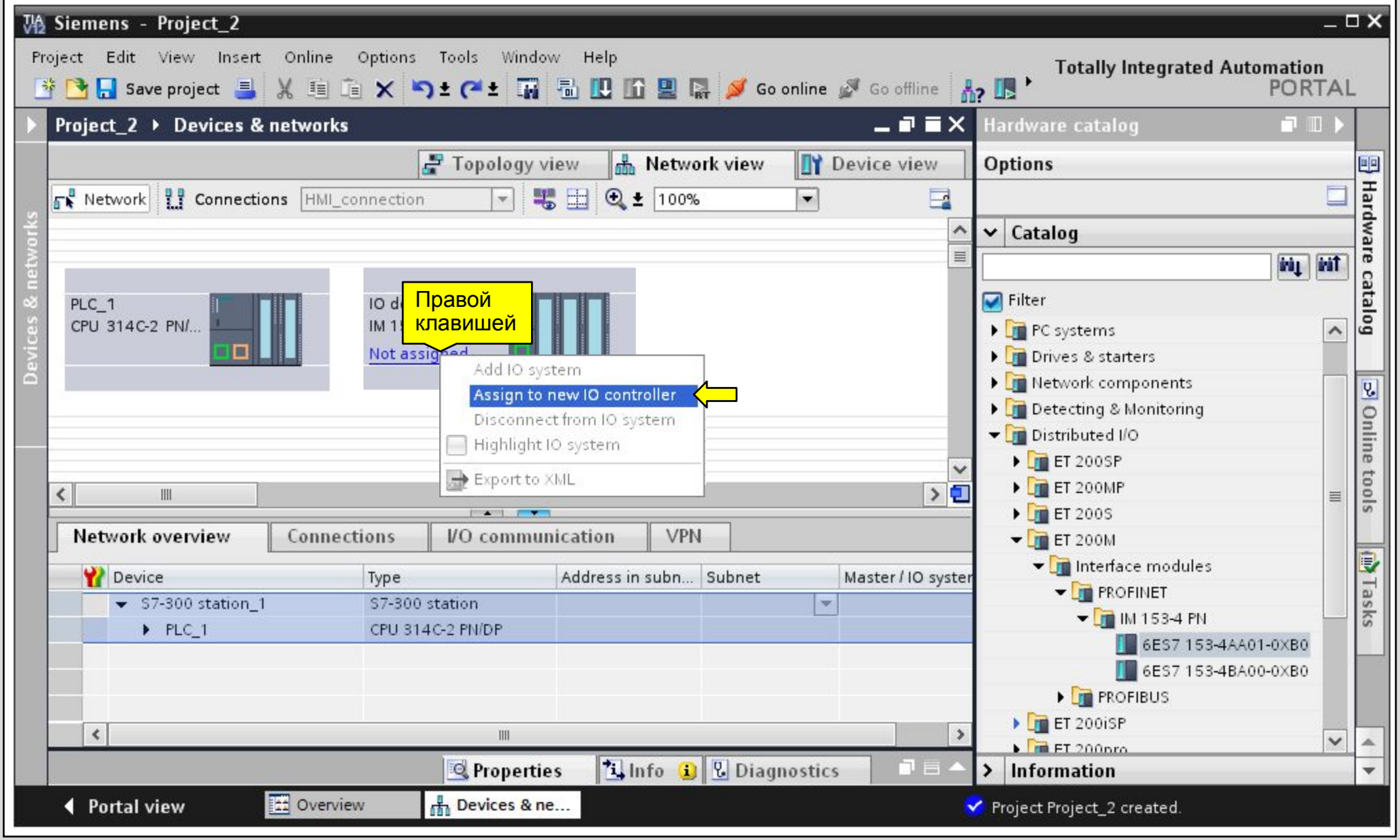
Пример многорядной конфигурации ПЛК с модулями IM360



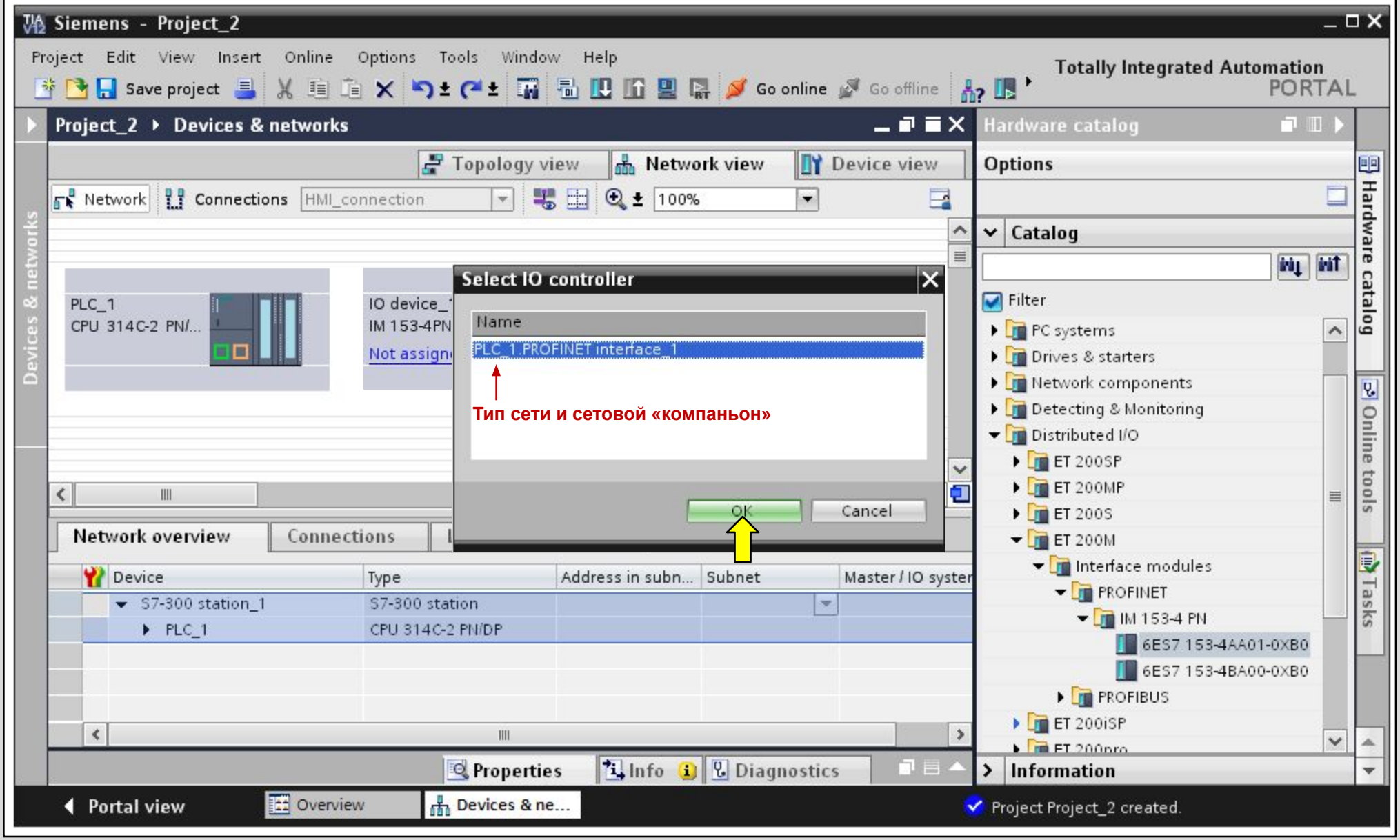
Конфигурирование распределенного ввода-вывода с подсетью PN



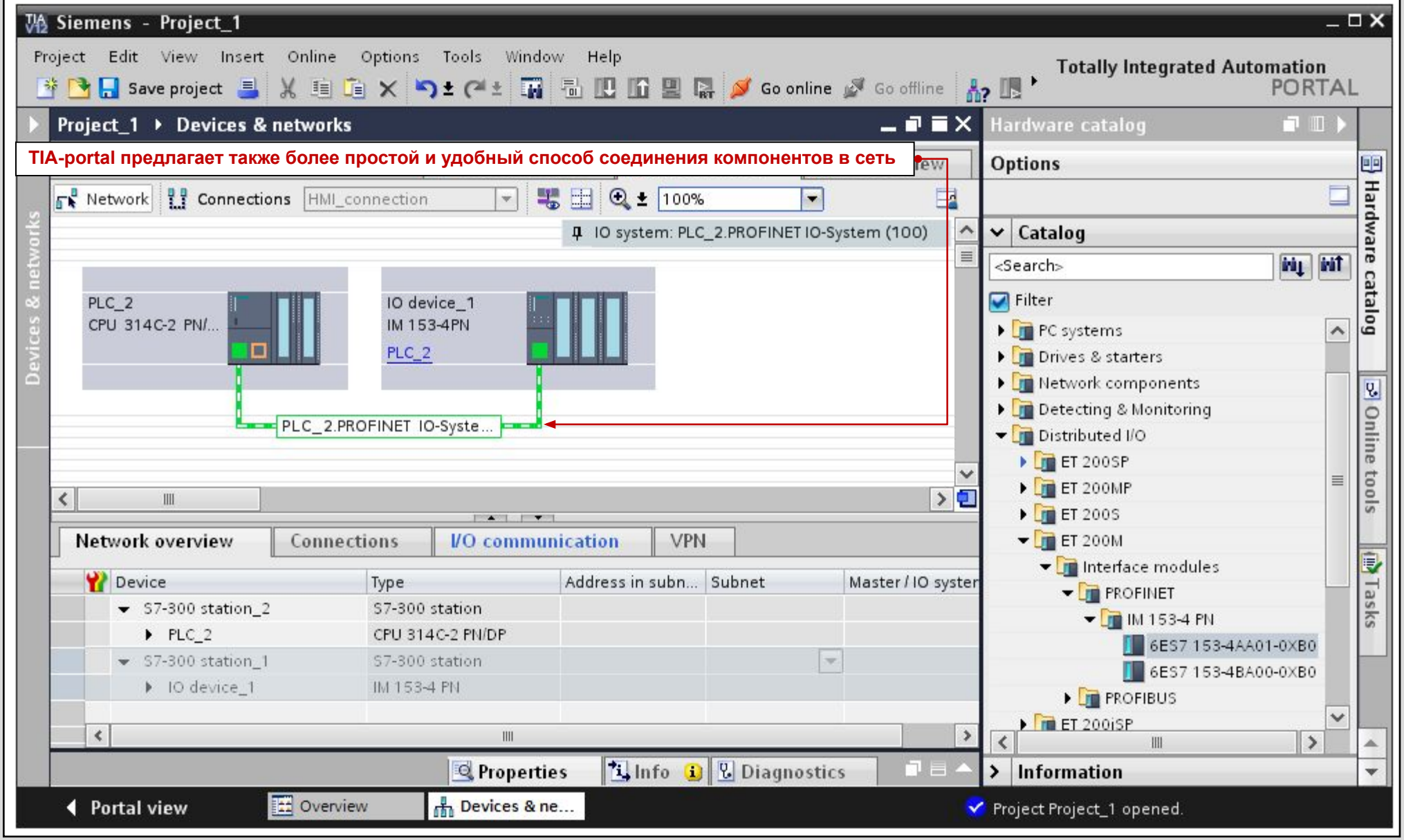
Конфигурирование распределенного ввода-вывода с подсетью PN



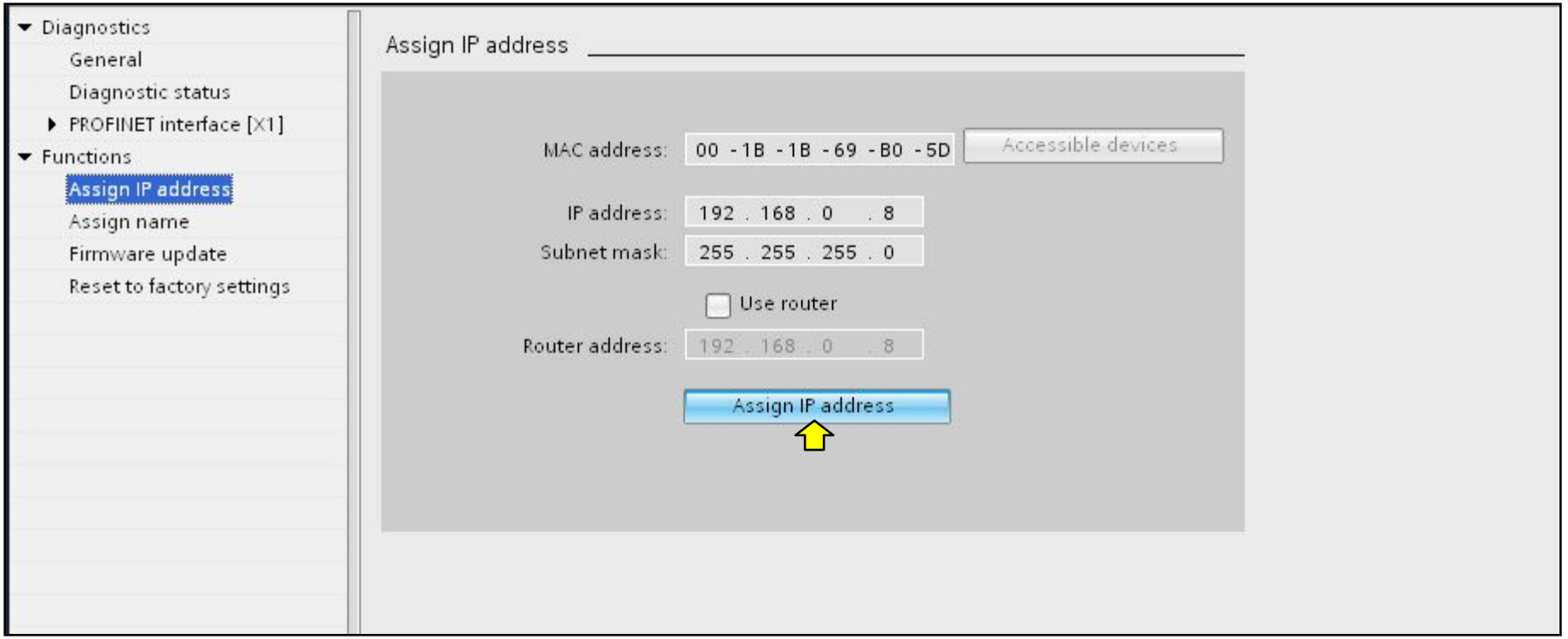
Конфигурирование распределенного ввода-вывода с подсетью PN



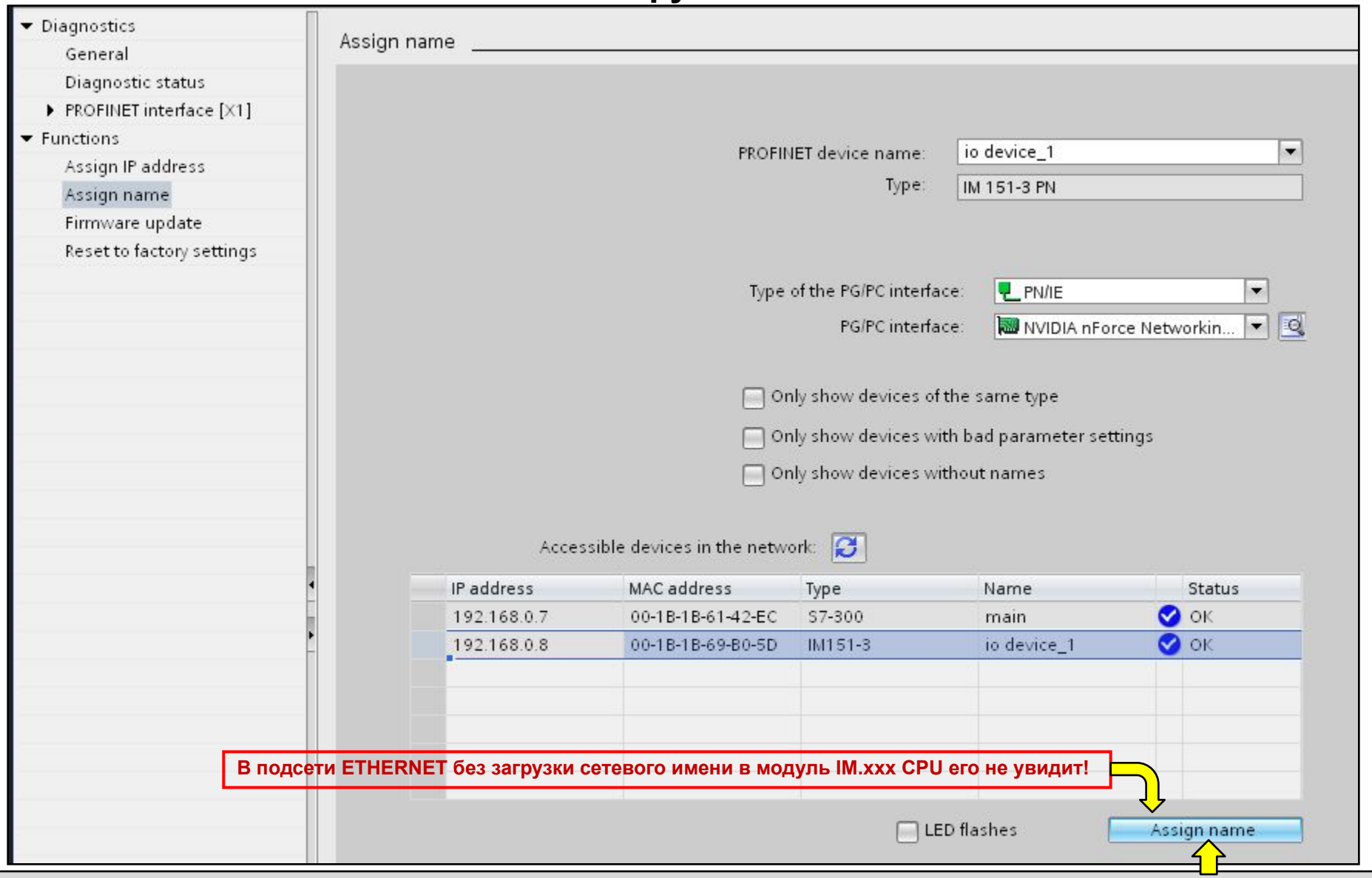
Конфигурирование распределенного ввода-вывода с подсетью PN



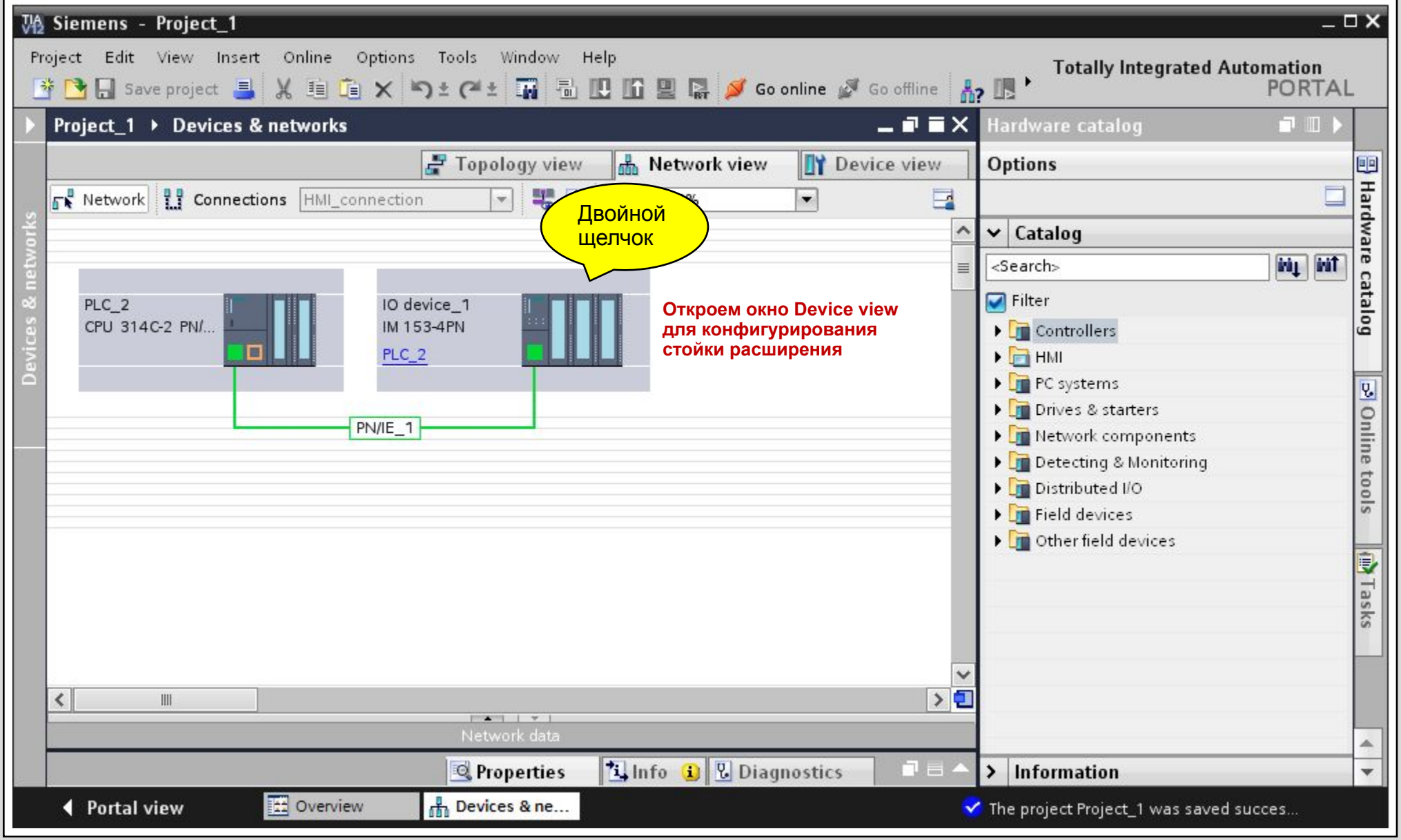
Конфигурирование распределенного ввода-вывода с подсетью PN
Назначение и загрузка IP-адреса



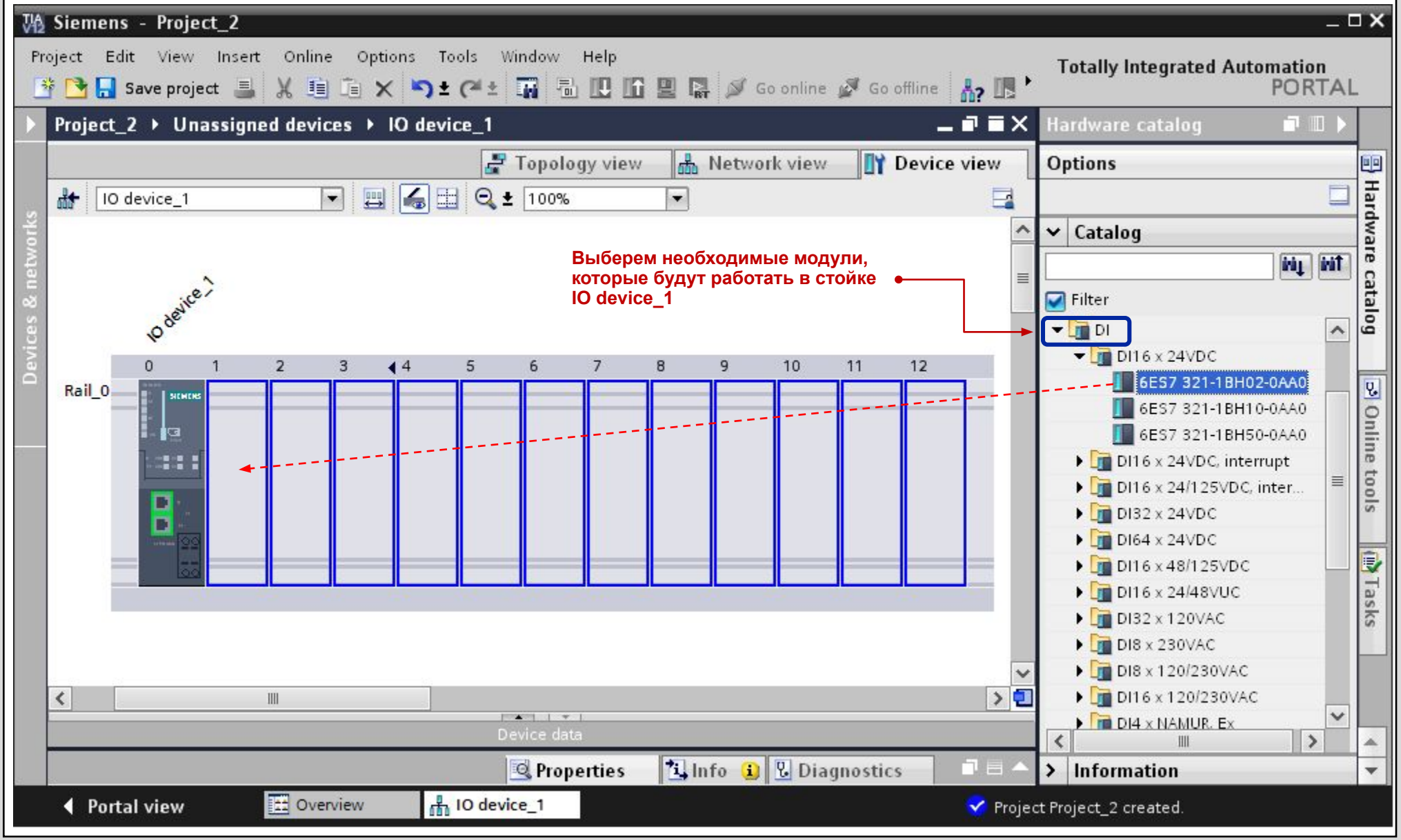
Конфигурирование распределенного ввода-вывода с подсетью PN. Назначение и загрузка сетевого имени



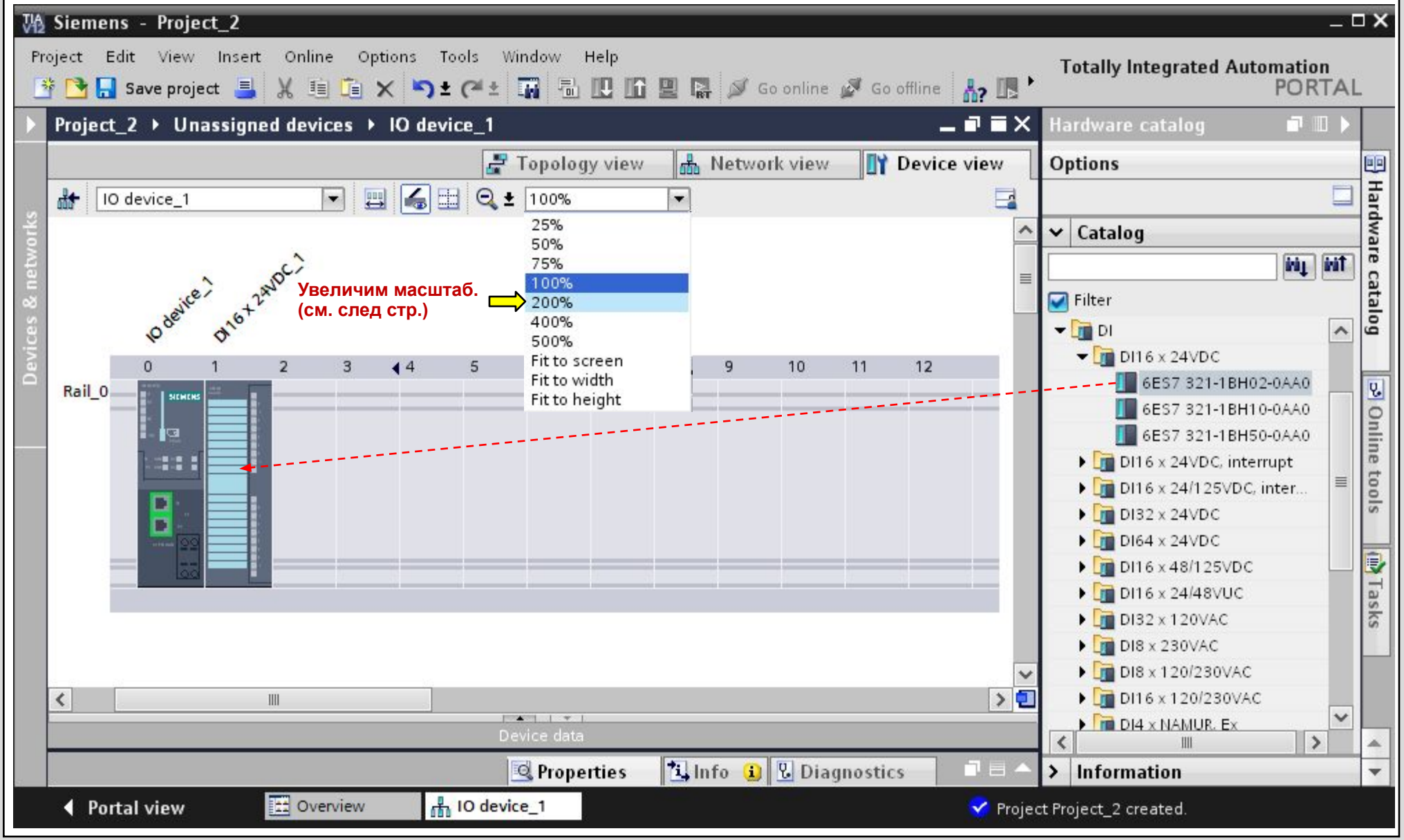
Конфигурирование распределенного ввода-вывода с подсетью PN



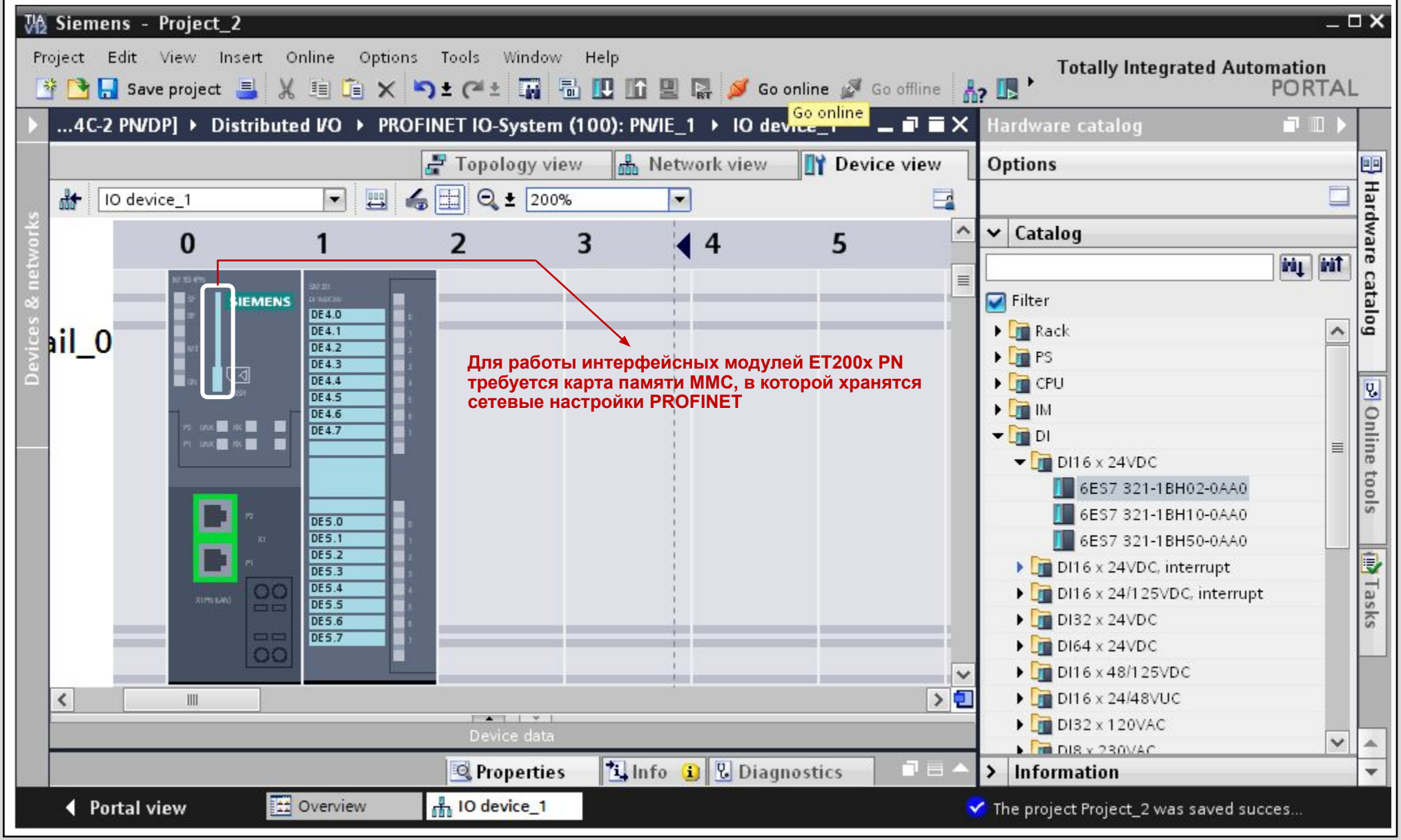
Конфигурирование распределенного ввода-вывода с подсетью PN



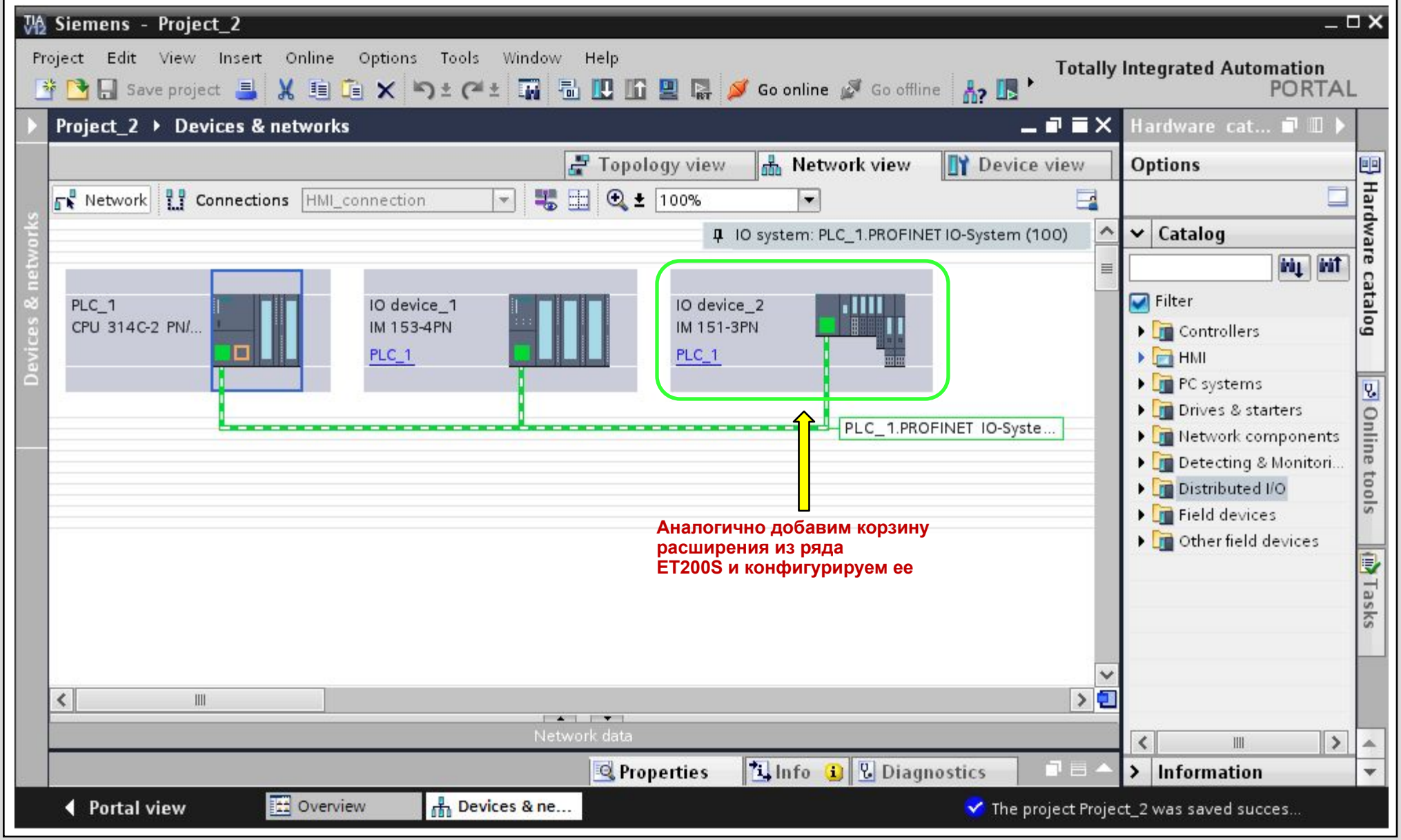
Конфигурирование распределенного ввода-вывода с подсетью PN



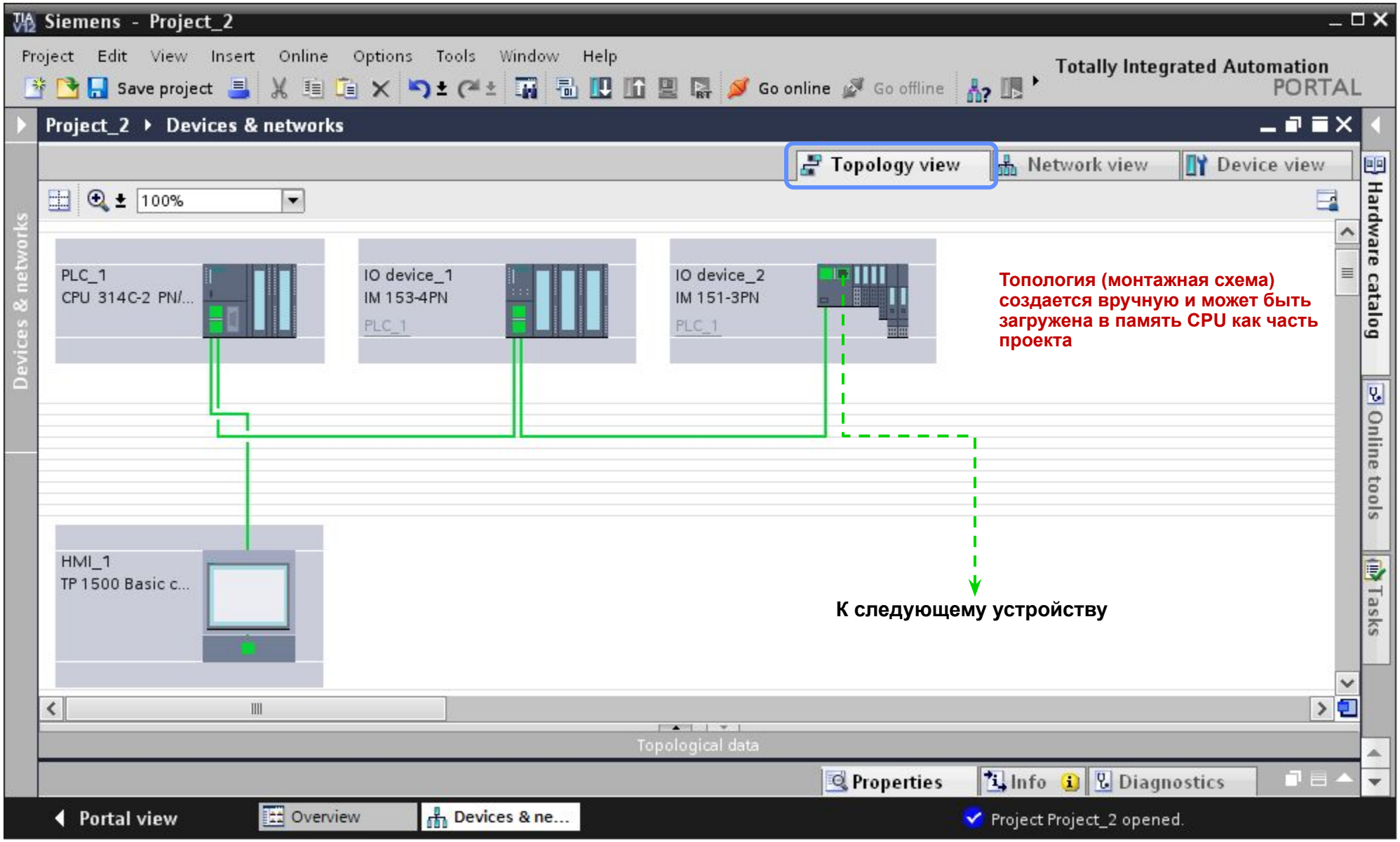
Конфигурирование распределенного ввода-вывода с подсетью PN



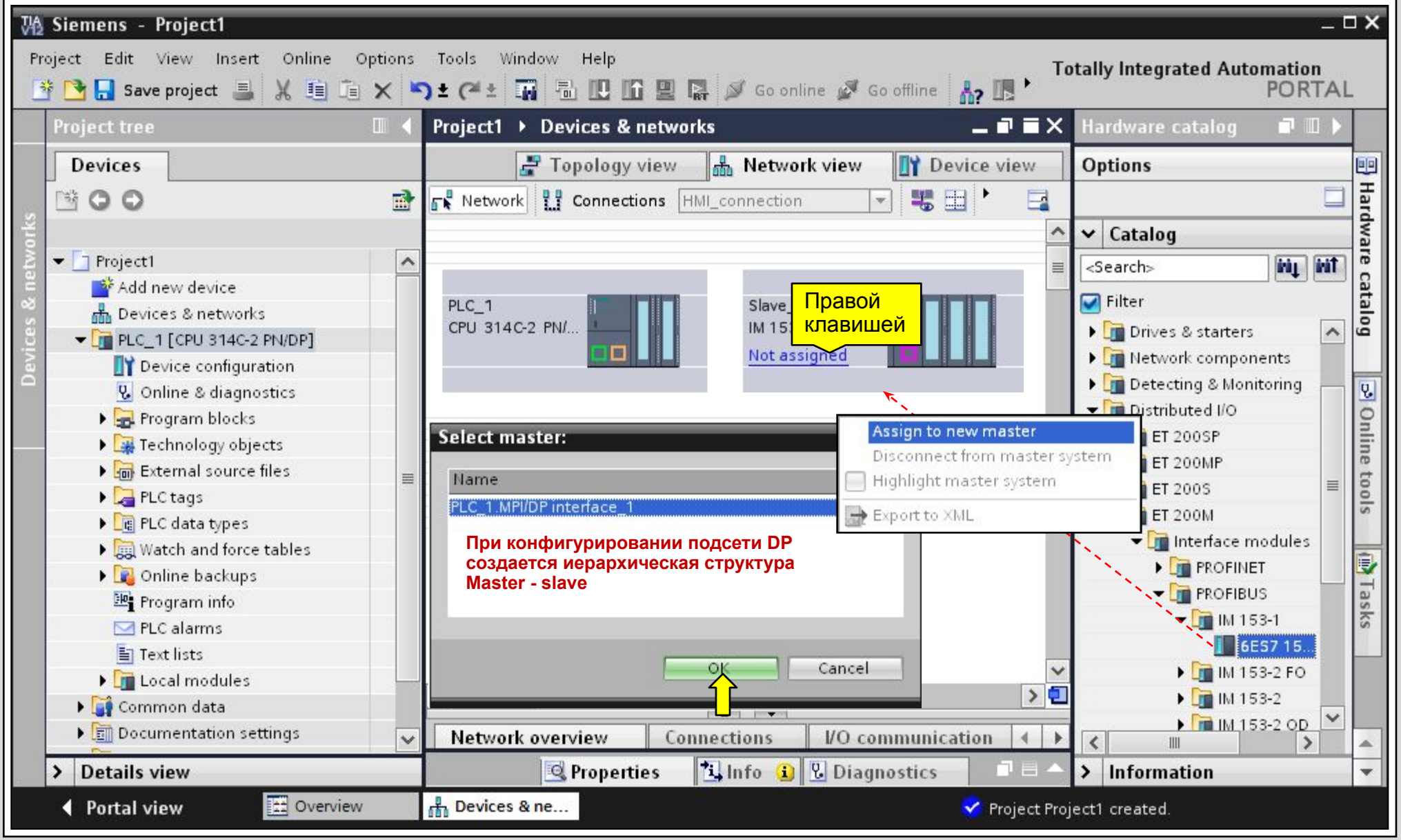
Пример конфигурации распределенного ввода-вывода в подсети PN



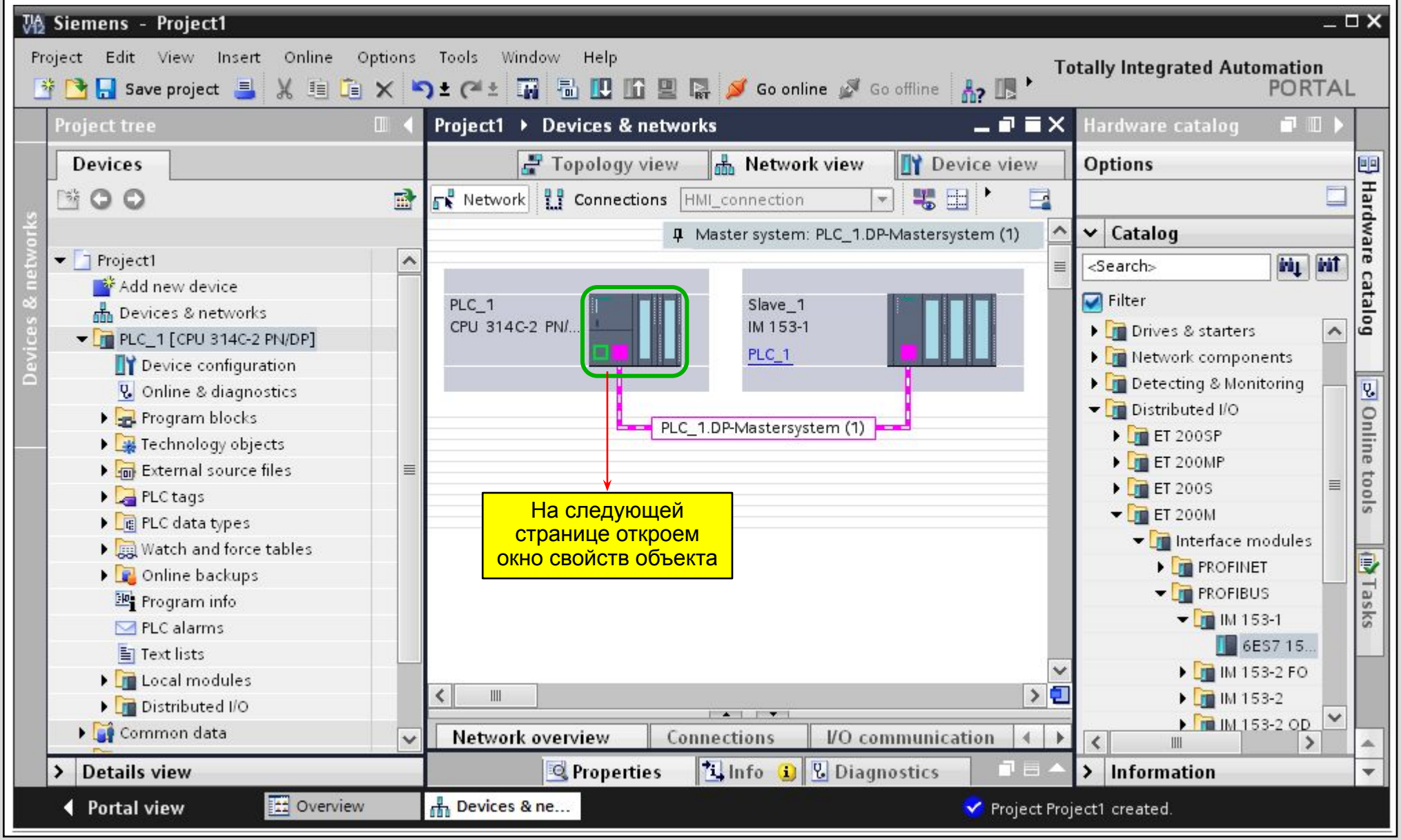
Создание топологии соединений подсети PN



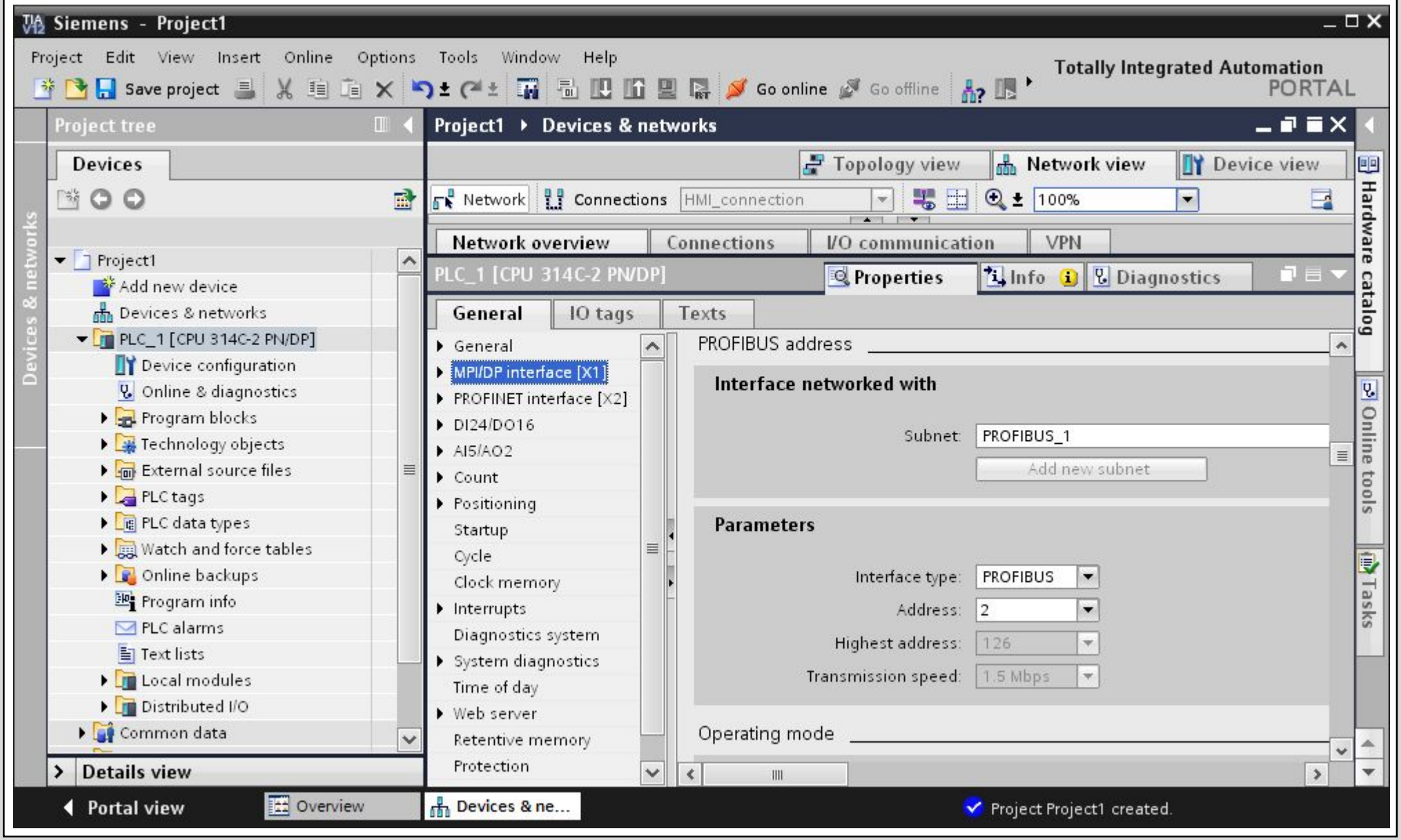
Конфигурации распределенного ввода-вывода в подсети DP



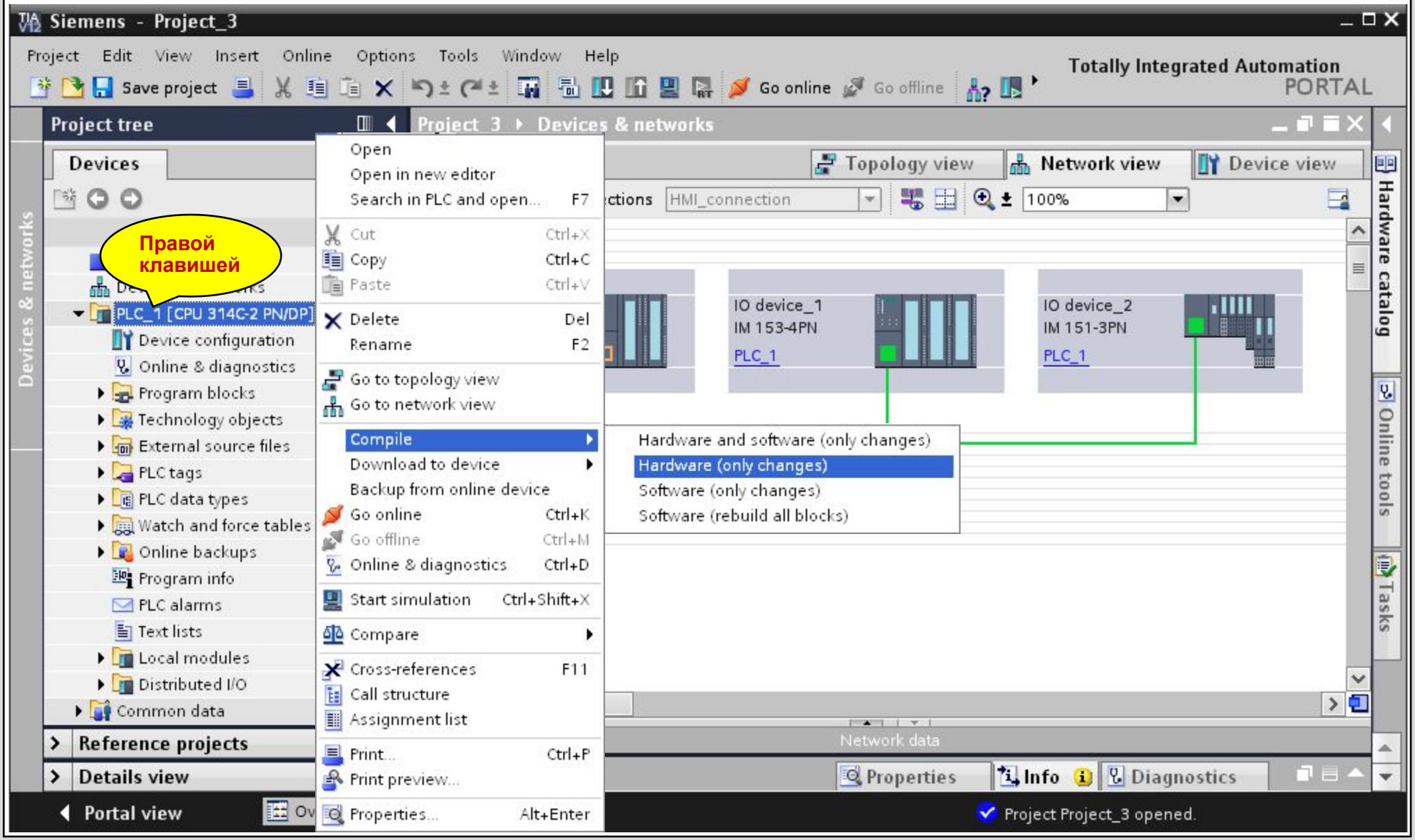
Конфигурации распределенного ввода-вывода в подсети DP



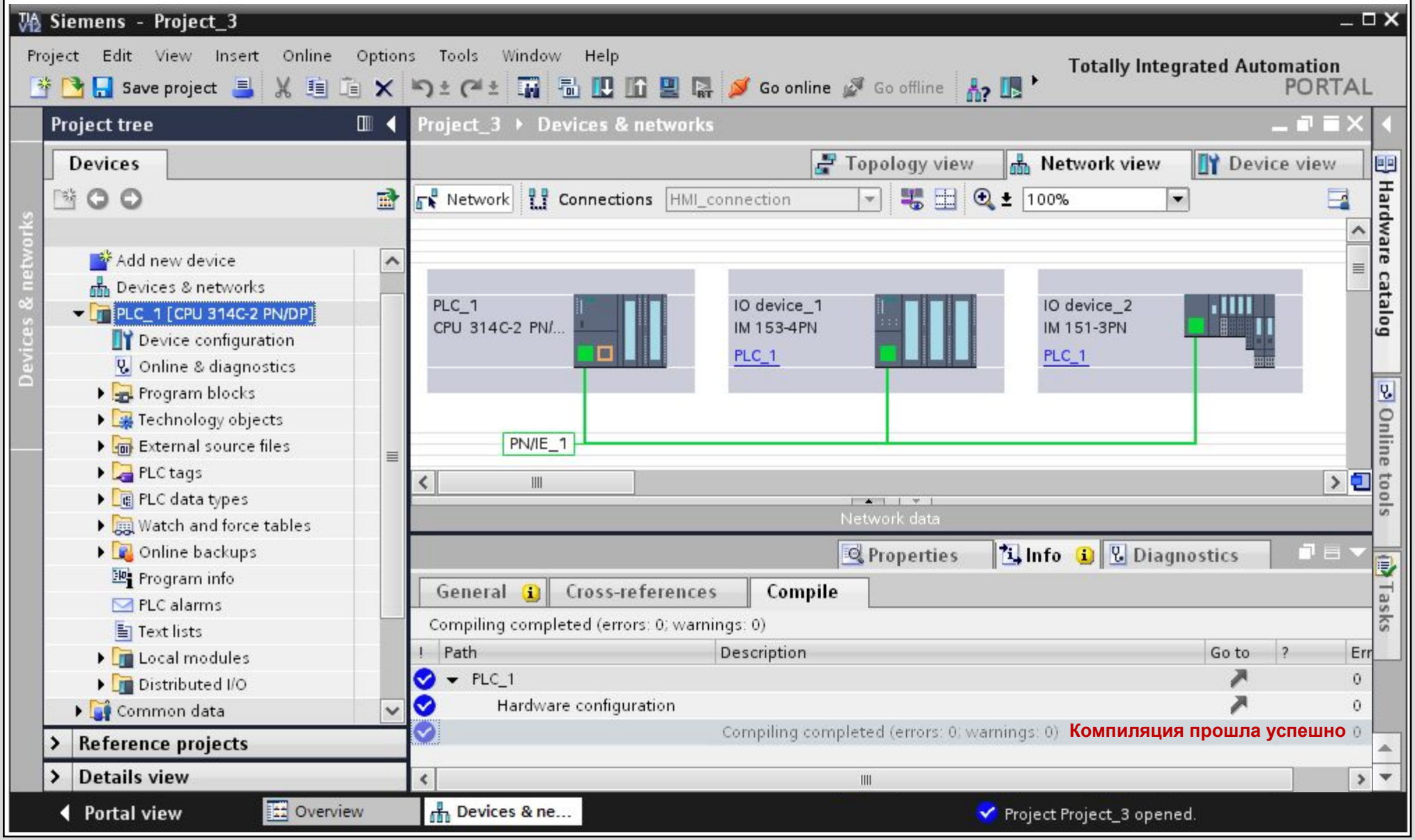
Конфигурации распределенного ввода-вывода в подсети DP



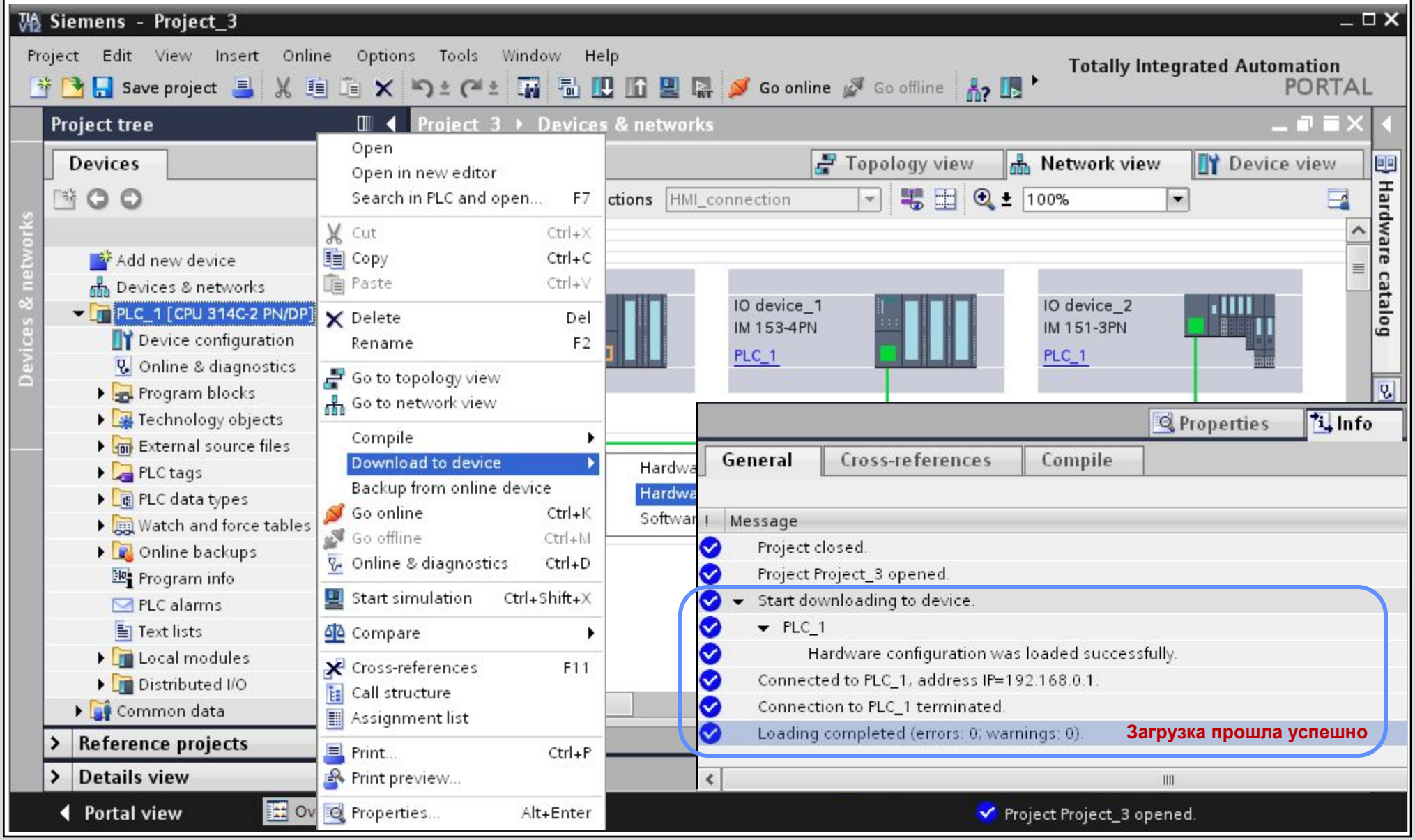
Компилирование аппаратной конфигурации



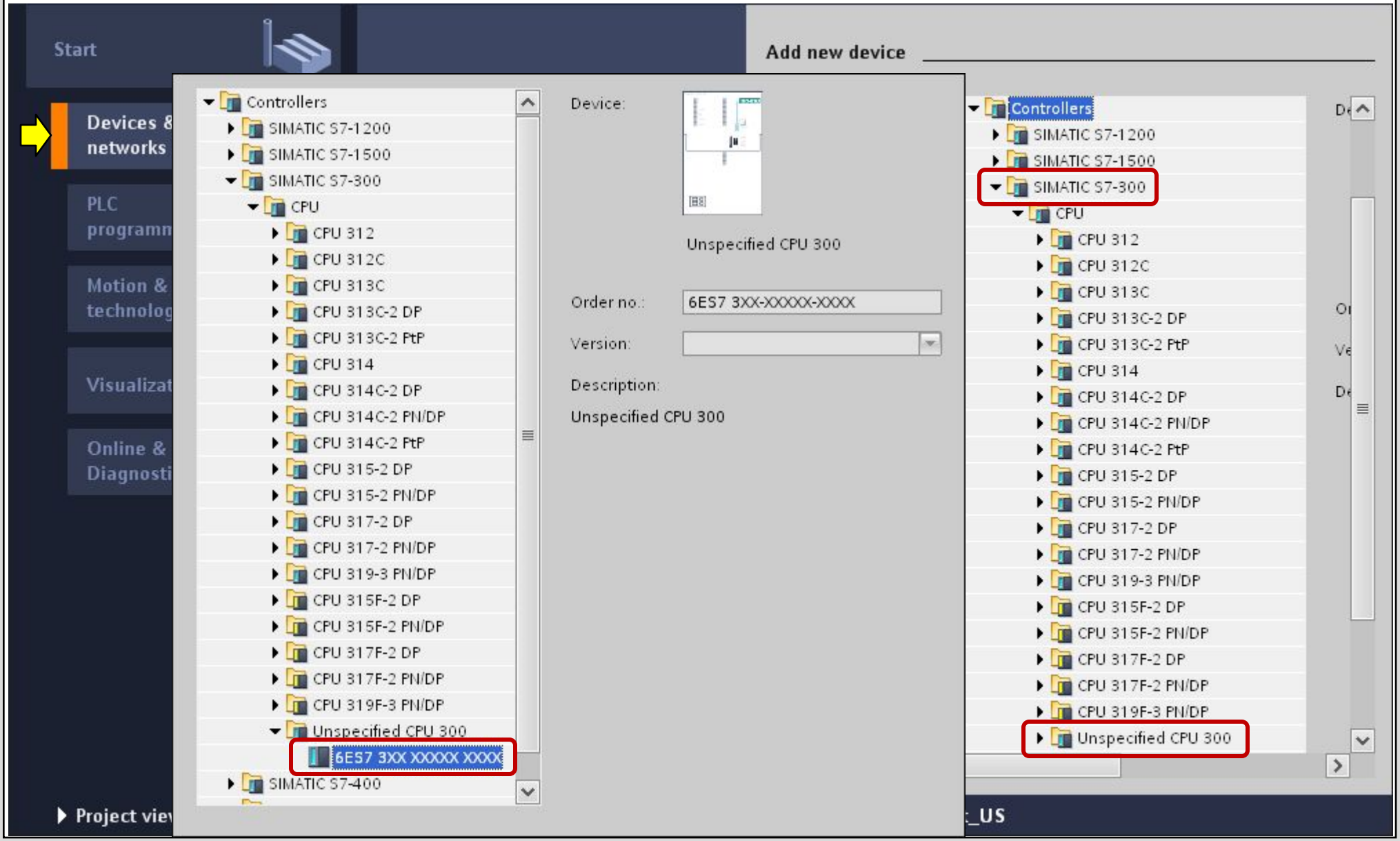
Компилирование аппаратной конфигурации



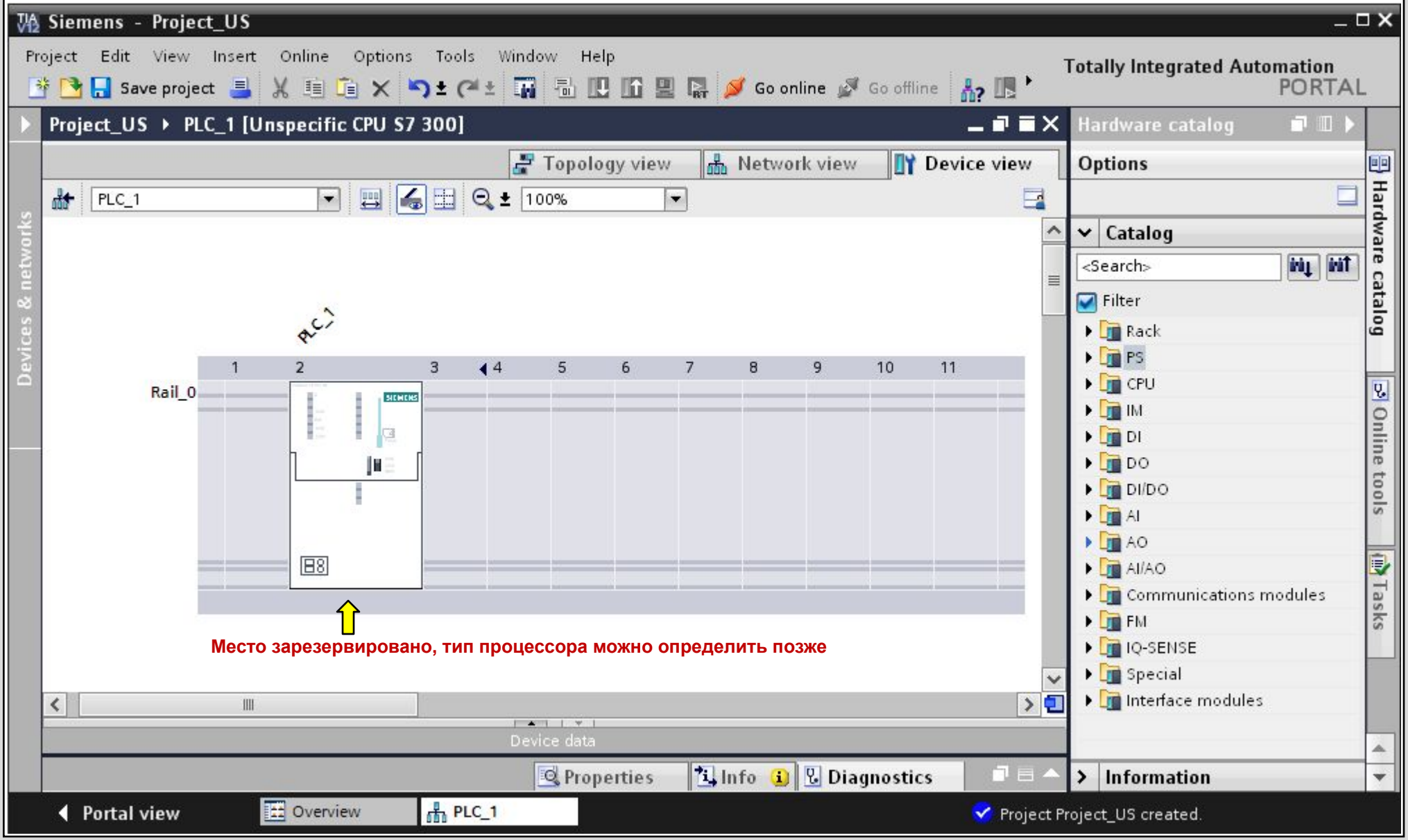
Загрузка аппаратной конфигурации в PLC



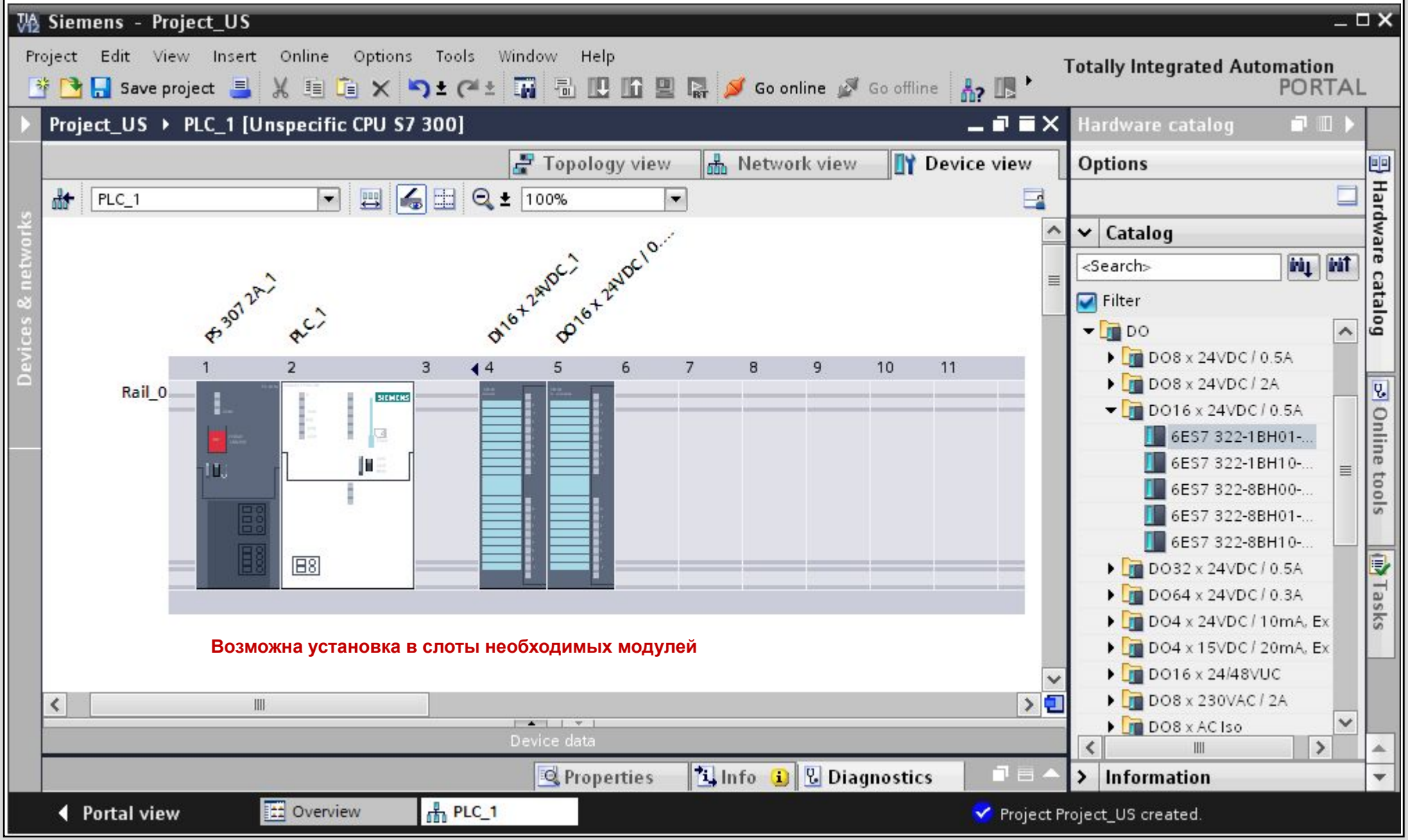
Создание проекта с «пустым» ЦПУ



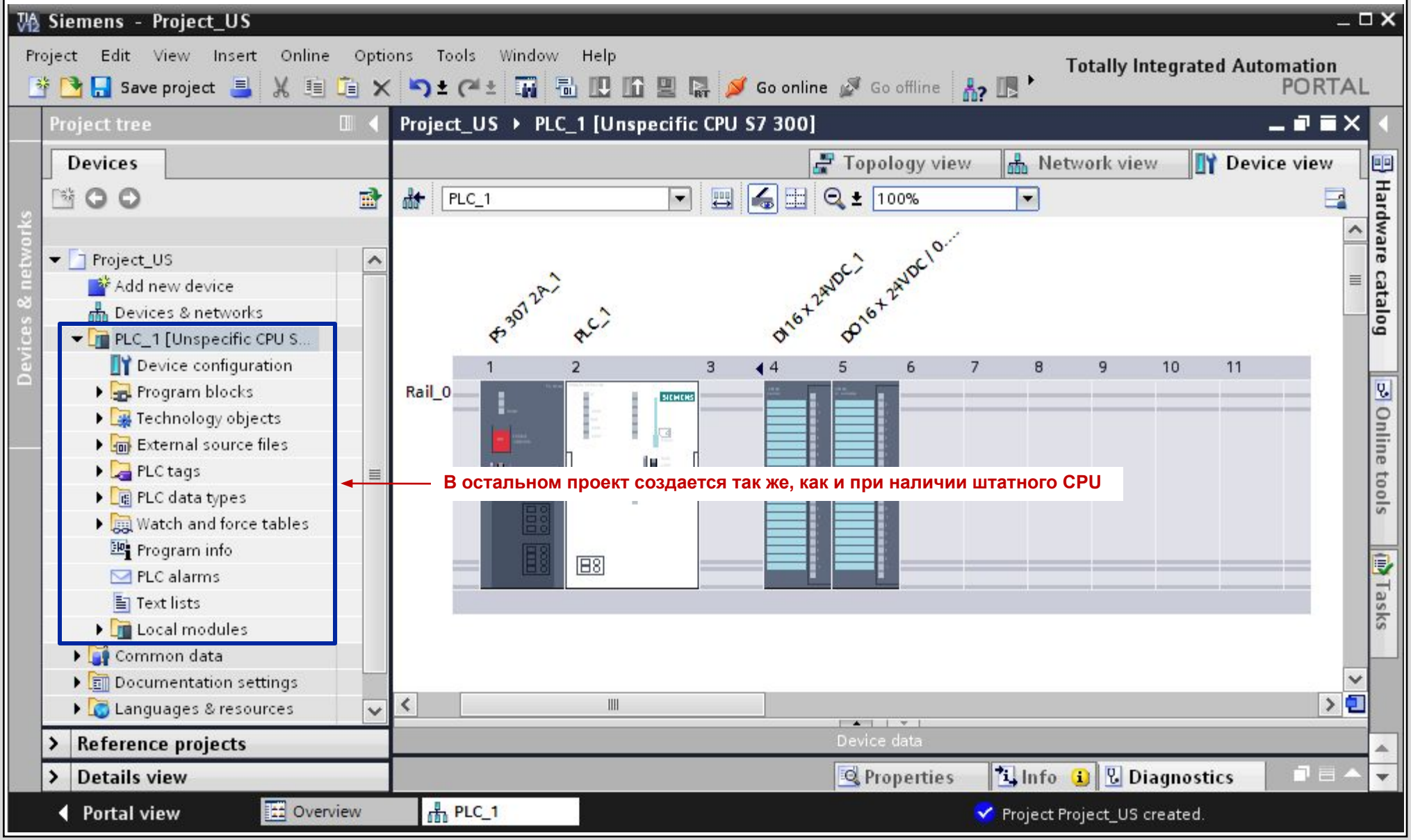
Создание проекта с «пустым» ЦПУ



Создание проекта с «пустым» ЦПУ



Создание проекта с «пустым» ЦПУ



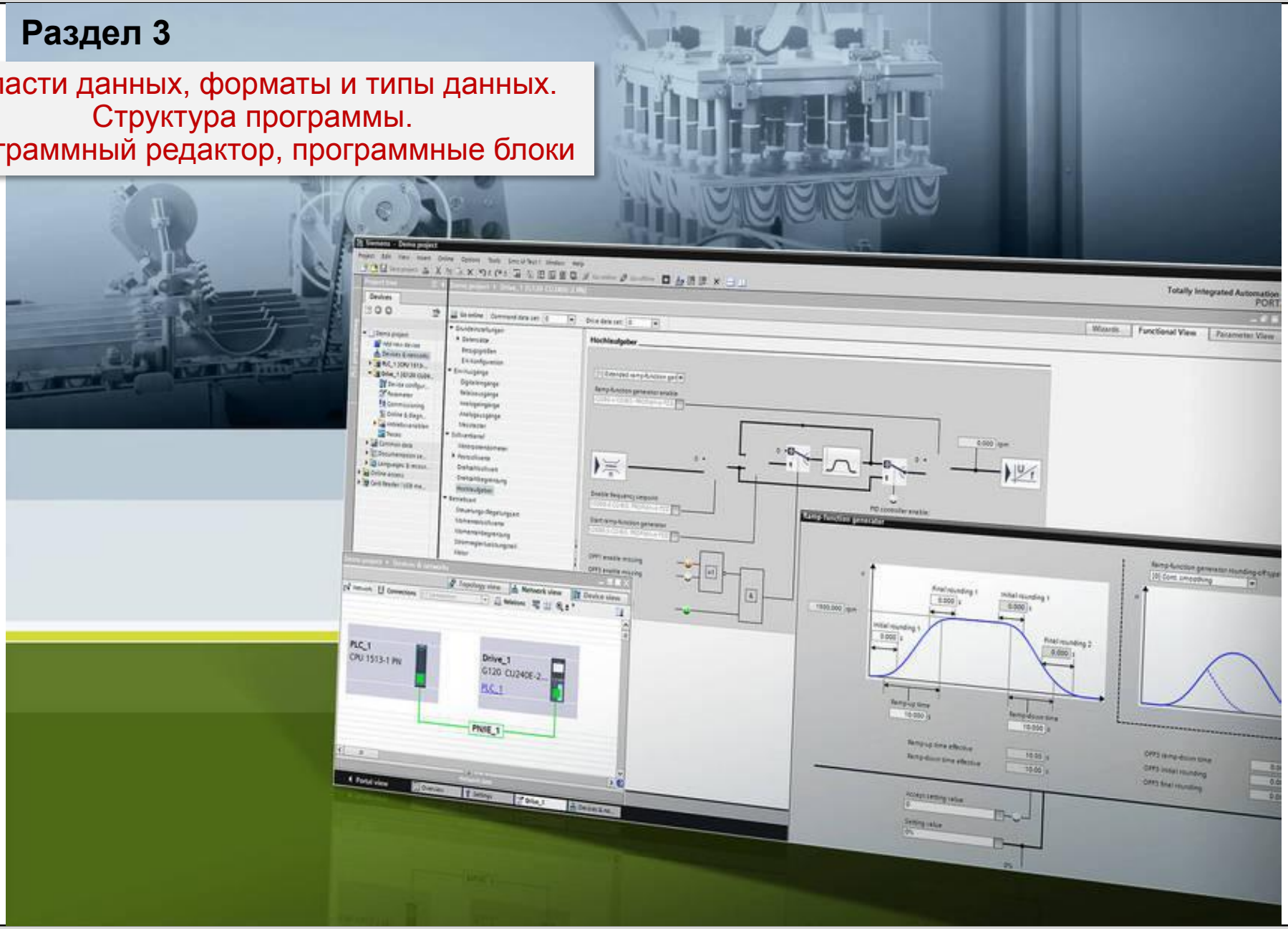
Конец раздела 2. Окно навигации

- Основы алгебры логики

  Общие сведения, создание проекта.
  Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги.
  Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 3

Области данных, форматы и типы данных.
Структура программы.
Программный редактор, программные блоки



Области данных PLC SIMATIC S-7

- Память дискретных входов (образ процесса)
- Память дискретных выходов (образ процесса)
- Память меркеров (меток)
- Память данных общего назначения
- Память временных переменных (стек локальных данных)
- Память внутренних подпрограмм (счетчики, таймеры и т.д.)
- Память данных периферии (в частности, аналоговые входы и выходы)

Форматы и основные типы данных

Формат данных

- Бит – один двоичный разряд
- Байт – 8 двоичных разрядов
- Слово – 16 двоичных разрядов
- Двойное слово – 32 двоичных разряда (в S7-1200/1500 есть 64-х битные данные)

Типы данных

- Bool – битные данные (дань вежливости автору булевой алгебры)
- Byte – число в формате байта
- Word- число без учета знака. Формат слова
- Int- число с учетом знака. Формат слова
- DWord – число без учета знака. Формат двойного слова
- DInt - число с учетом знака. Формат двойного слова
- Real- число с плавающей точкой. Формат двойного слова
- S5Time – задание уставки подпрограммы «таймер». Формат слова
- Типы данных Char, String, составные типы данных

Пример типа данных INT (16-разрядное целое)

Диапазон значений от -32768 до +32767
(Без знака: от 0 до 65535)

Арифметические операции:	Операции сравнения:
+ I, - I, *I, /I	><I, ==I, >=I

DEC: + 662

BIN.: 2# 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0

Знак положительного числа

HEX: W#16 0 2 9 6

Прочтение положительного числа

$6 \times 16^0 = 6$
$9 \times 16^1 = 144$
$2 \times 16^2 = 512$
662

DEC: - 662

BIN.: 2# 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
1 1 1 1 1 1 0 1 0 1 1 0 1 0 1 0

Знак отрицательного числа

HEX: W#16 F D 6 A

Прочтение без учета знака

$10 \times 16^0 = 16$
$6 \times 16^1 = 96$
$13 \times 16^2 = 3328$
$15 \times 16^3 = 61440$
64874

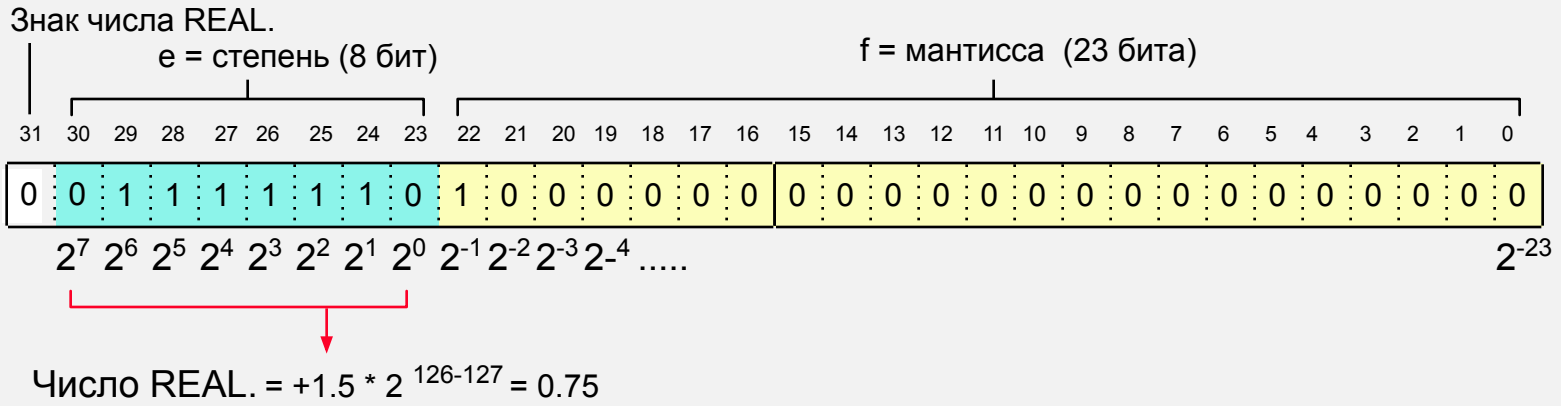
Пример типа данных REAL (32-разрядное число с плавающей точкой)

Диапазон значений от $-1.175495 \cdot 10^{-38}$ до $3.402823 \cdot 10^{+38}$

Общий формат числа REAL = (Sign) • (1.f) • (2^{e-127})

Математические операции:	Операции сравнения:
+ R, - R, *R, /R sin, cos, ln, exp, SQR	><R, ==R, >=R

Пример: 7.50000e-001 ($7.5 * 10^{-1} = 0.75$)



Адресация дискретных входов/выходов через образ процесса

При обращении в программе пользователя в контроллерах S7-300/S7-400 к областям дискретных входов (I x.y) и выходов (Q x.y) опрашиваются не сигналы непосредственно на цифровых сигнальных модулях, а происходит обращение к выделенной области в системной памяти CPU. Эту область памяти называют образом процесса. Образ процесса делится на две части

- образ процесса входов
- образ процесса выходов

Для чего это нужно.
При обработке программы важно, чтобы в текущем цикле значения входов были одинаковы во всех точках обращения к ним. Иначе могут возникнуть логические конфликты. Для выходов это не так критично, но обращение к внутренней памяти по времени гораздо короче, чем обращение к памяти модулей, что необходимо для сокращения общего времени цикла .

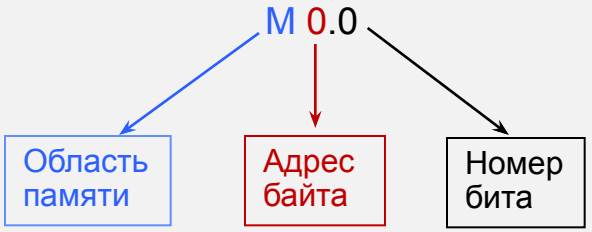
Реализуется этот механизм следующим образом:
в начале цикла программы производится опрос сигнальных модулей ввода, их значения заносятся в память образа процесса входов. При обработке программы производится опрос значений входов из памяти образа процесса.
Далее производится передача полученных в результате обработки программы значений дискретных выходов из памяти образа процесса выходов в сигнальные модули выходов в начале следующего цикла.

Примечание:
в контроллерах серии S7-1200/S7-1500 для сокращения времени реакции на событие возможно непосредственное обращение к модулям дискретных входов-выходов, минуя память образа процессора. Синтаксис обращения: I x.y:P или Q x.y:P. Если префикса (:P) нет, опрашивается образ процесса.

Адресация и структура представления данных

Адресация

Обращение к биту данных

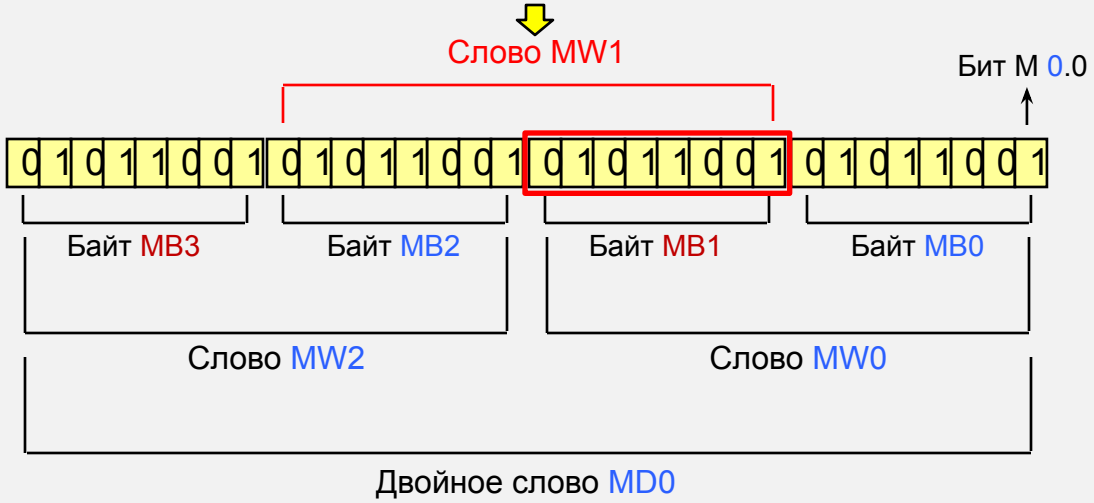


Обращение к данным в других форматах

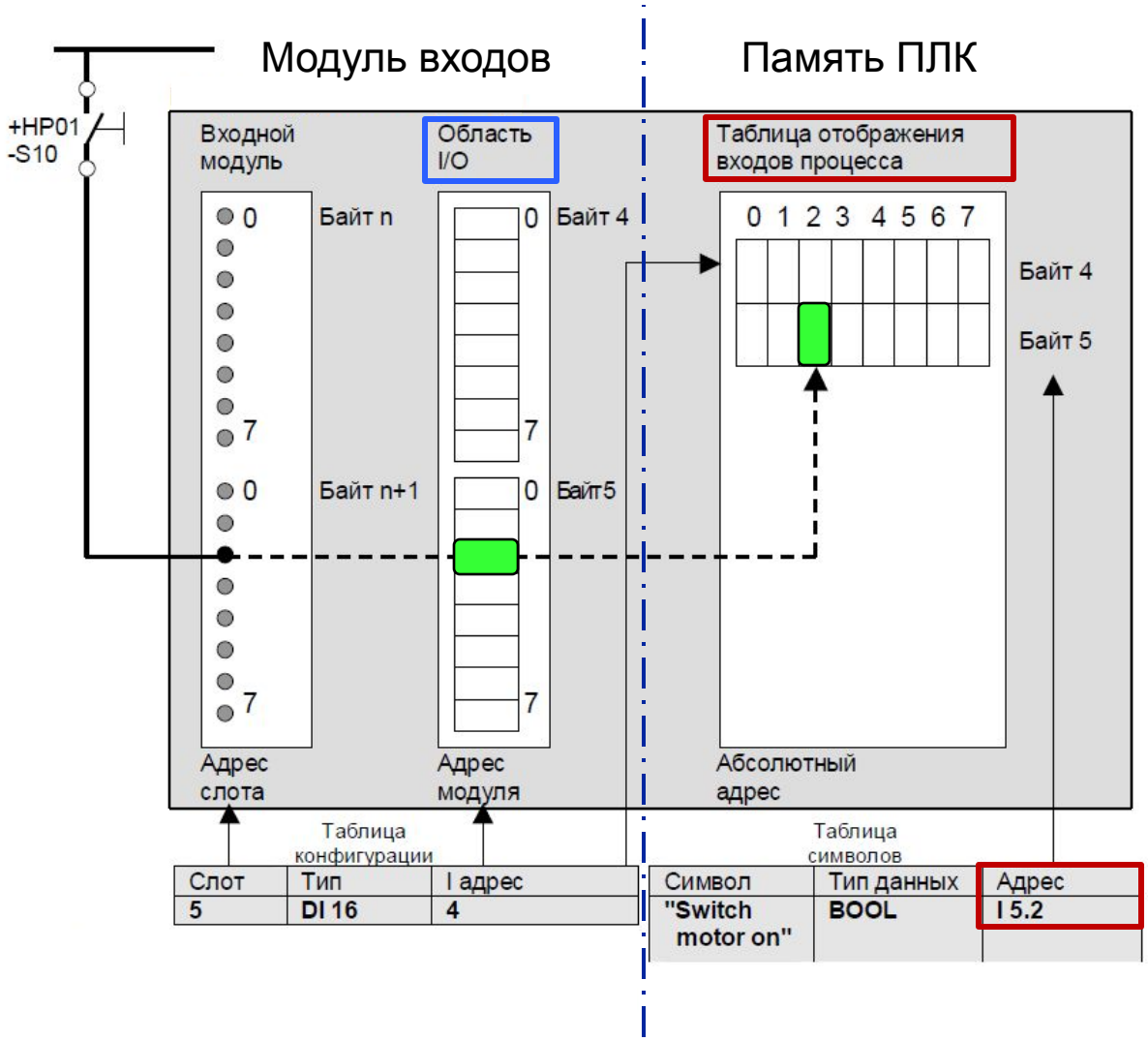
- к байту меркеров — MB0
- к слову меркеров — MW0
- к двойному слову меркеров — MD0

Структура представления

Пример ошибочной адресации n-байтной переменной



Пример адресации дискретного входа I 5.2 через образ процесса



Типы программных блоков

Структурно программа выполняется в виде программных блоков, которые разделяются по их предназначению:

Организационные блоки. Осуществляют связь операционной системы с программой пользователя. Различаются способом запуска. Могут выполняться циклически (основной способ), по времени, по аппаратным прерываниям. В одних OB осуществляется вызов программных блоков, другие производят диагностику и выполняют ряд других функций. Могут выполнять роль программных и содержать программу пользователя.

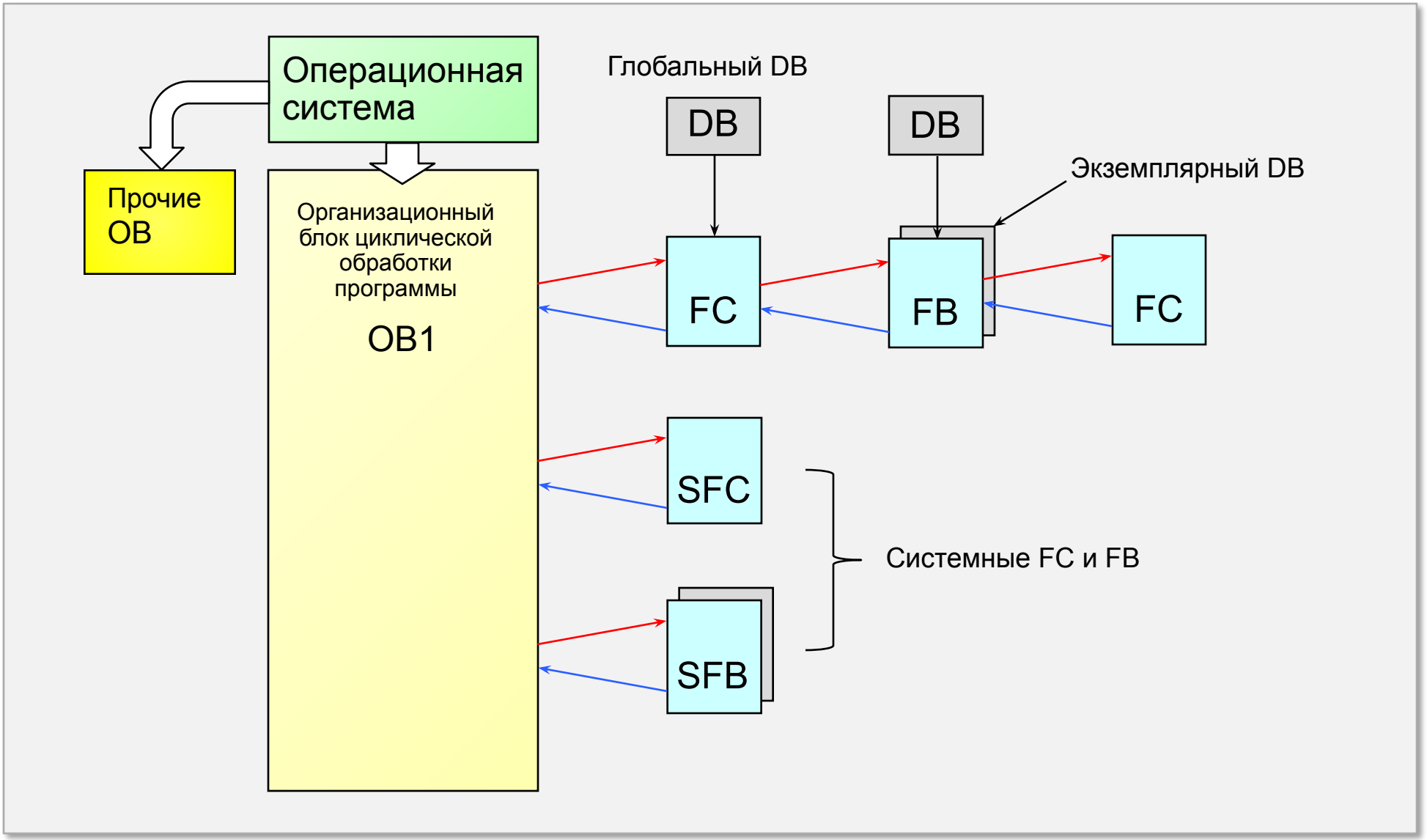
Блоки данных предназначены для хранения данных пользователя. Их два вида. Одни DB могут содержать глобальные данные, доступные в любой точке программы, а другие (IDB) содержат данные для использования их в конкретном программном блоке, который называется функциональным (FB). В блоках данных могут быть организованы данные всех типов.

Функциональные блоки являются программными и содержат программу пользователя. Входные данные, внутренние переменные и результаты обработки, загружаются в выделенный для этой цели блок данных IDB (Instance data Block).

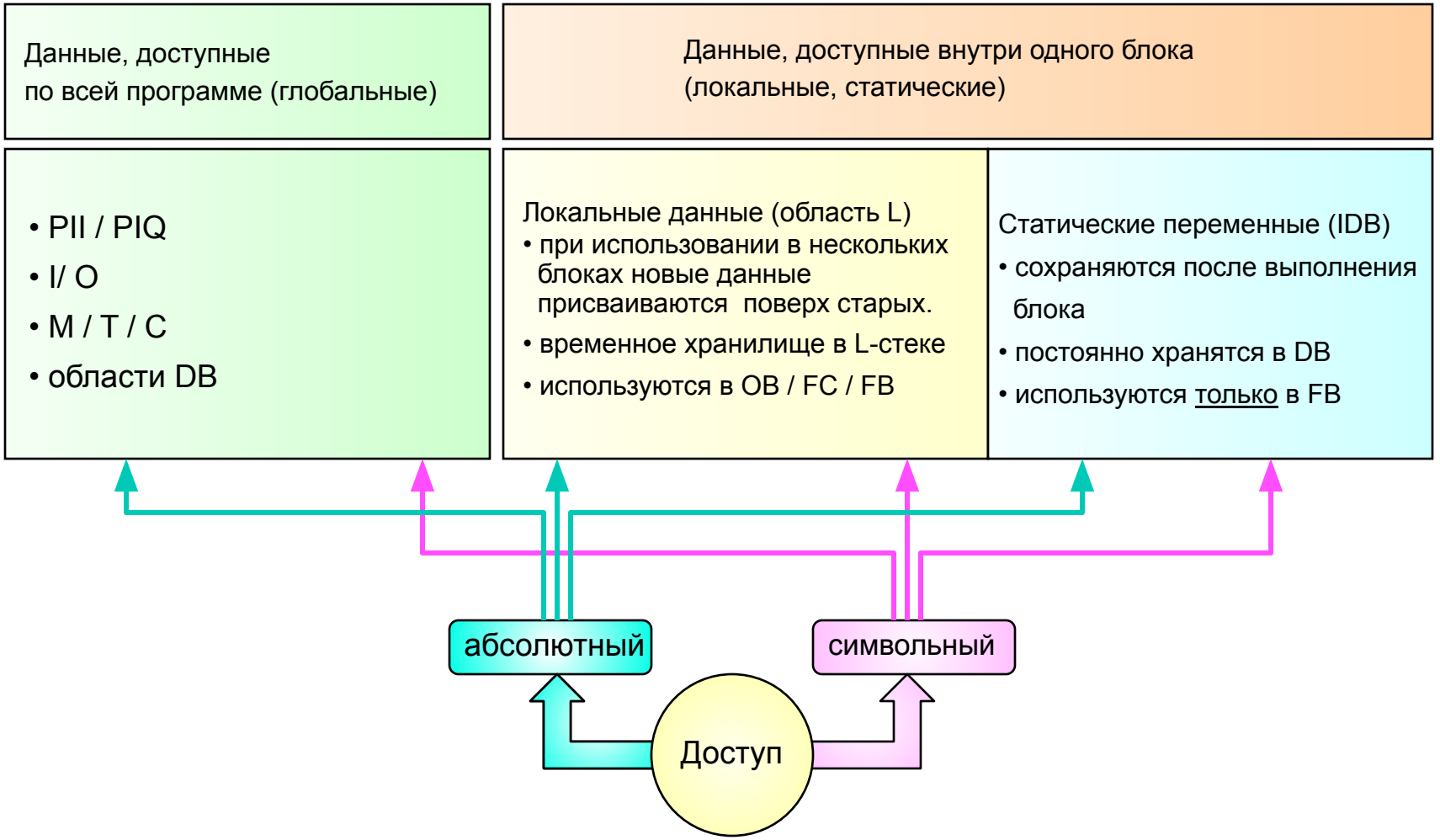
Функции – блоки, которые также являются программными и содержат программу пользователя. К моменту вызова функции все ее входные данные должны быть определены. Такой механизм позволяет использовать функции без выделенных блоков данных.

Системные функциональные блоки (SFB) и системные функции (SFC), которые имеются в составе ПЛК и предназначены для выполнения стандартных функций, например передачи данных, чтения времени и даты из ПЛК и т. д.

Общая структура вызова и обработки программных блоков



Распределение данных в ПЛК



Циклическое выполнение программы

Стартовый блок (OB 100) выполняется, например, после включения питания

Запуск контроля времени цикла

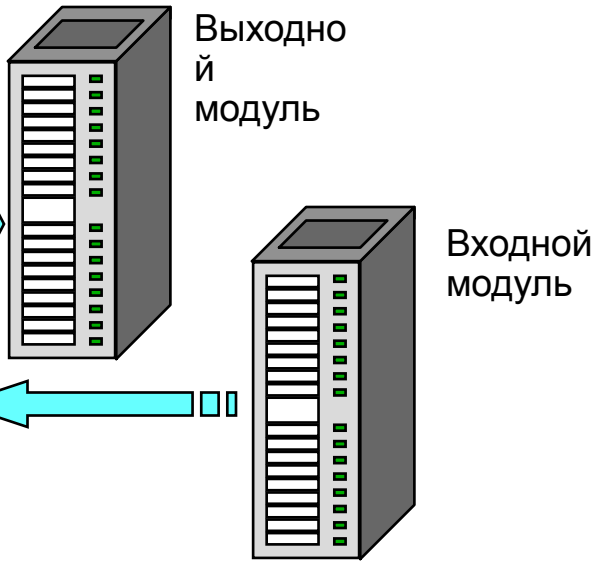
Выгрузка области отображения выходов (PIQ) в выходные модули

Считывание состояний входных сигналов и сохранение их в области отображения входов (PII)

Выполнение программы в OB1, других циклических OB
Вызываются программные блоки FB, FC, блоки данных.

Вызываются OB прерываний.

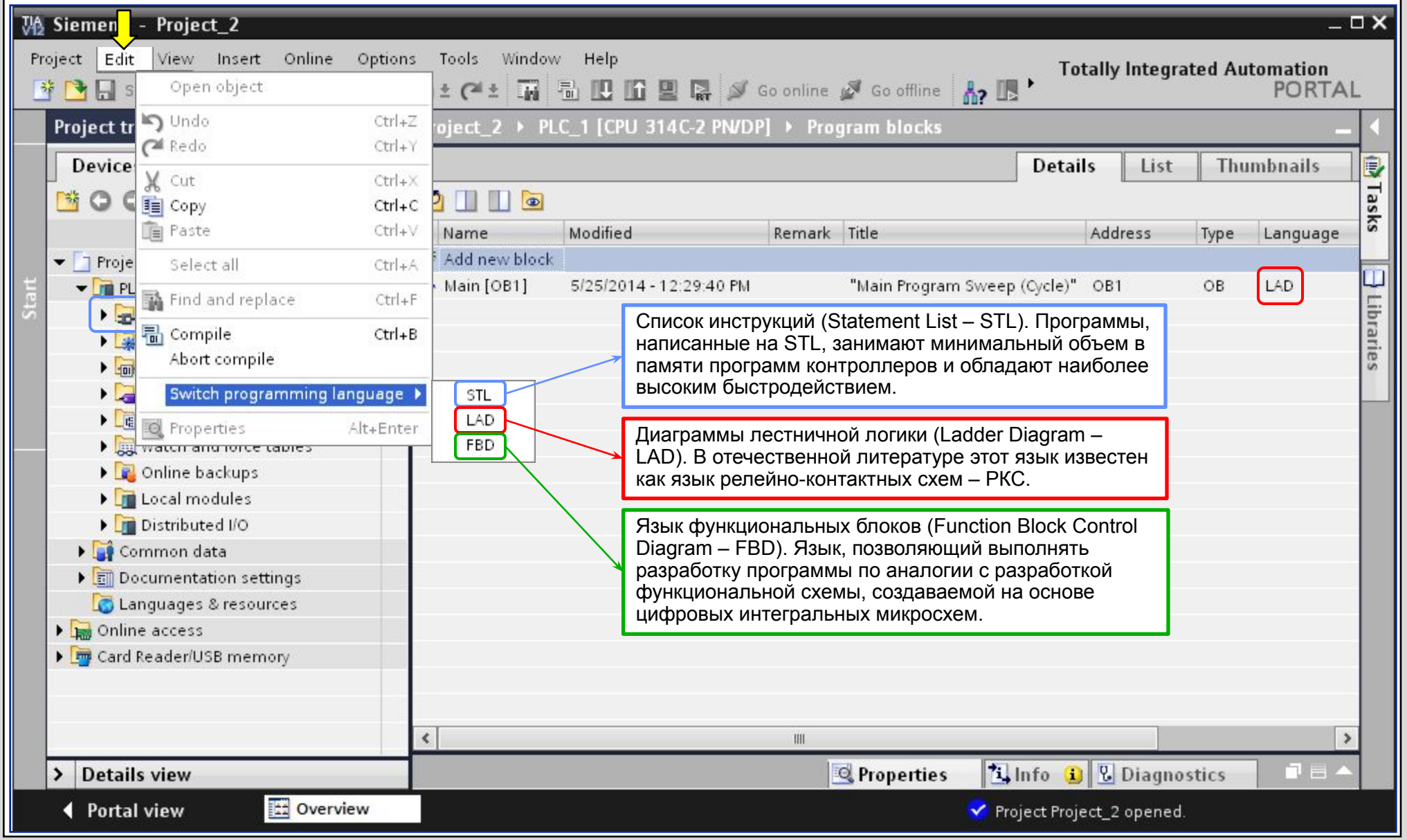
Цикл CPU



Блок OB 1

A I 0.1
A I 0.2
= Q8.0

Структура проекта, языки представления программы

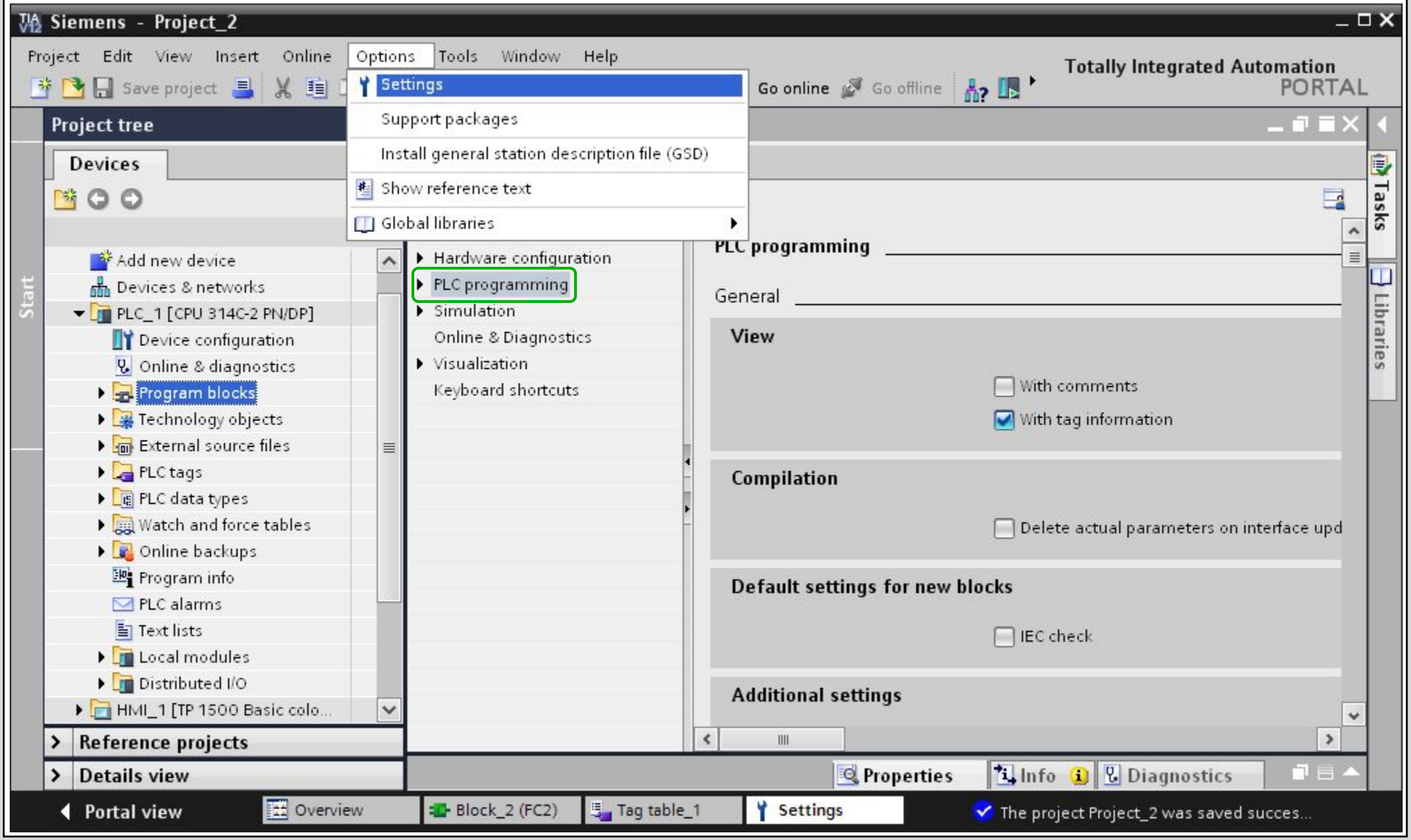


Список инструкций (Statement List – STL). Программы, написанные на STL, занимают минимальный объем в памяти программ контроллеров и обладают наиболее высоким быстродействием.

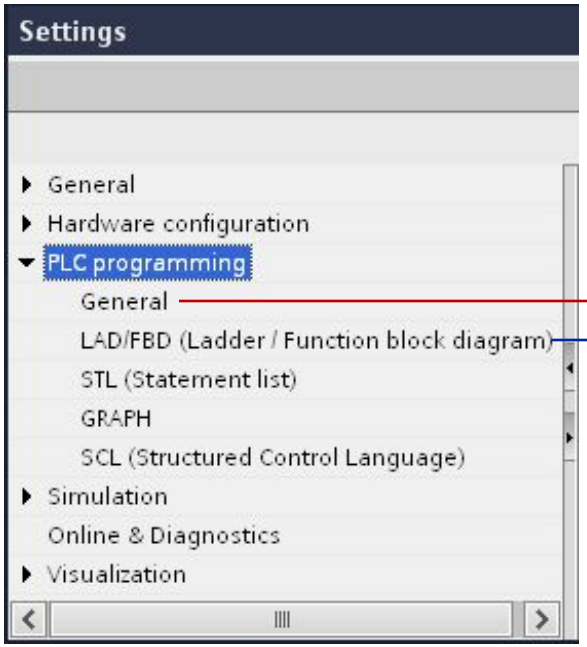
Диаграммы лестничной логики (Ladder Diagram – LAD). В отечественной литературе этот язык известен как язык релейно-контактных схем – РКС.

Язык функциональных блоков (Function Block Control Diagram – FBD). Язык, позволяющий выполнять разработку программы по аналогии с разработкой функциональной схемы, создаваемой на основе цифровых интегральных микросхем.

Настройка редактора



Настройка редактора



View

With comments

With tag information

Compilation

Delete actual parameters on interface update

Default settings for new blocks

IEC check

Additional settings

Show autocomplete list

Mnemonics: International

Font

Font size: 9

View

Layout: Compact

Wide

With absolute information

Operand field

Maximum width: 8 characters

Maximum height: 99 characters

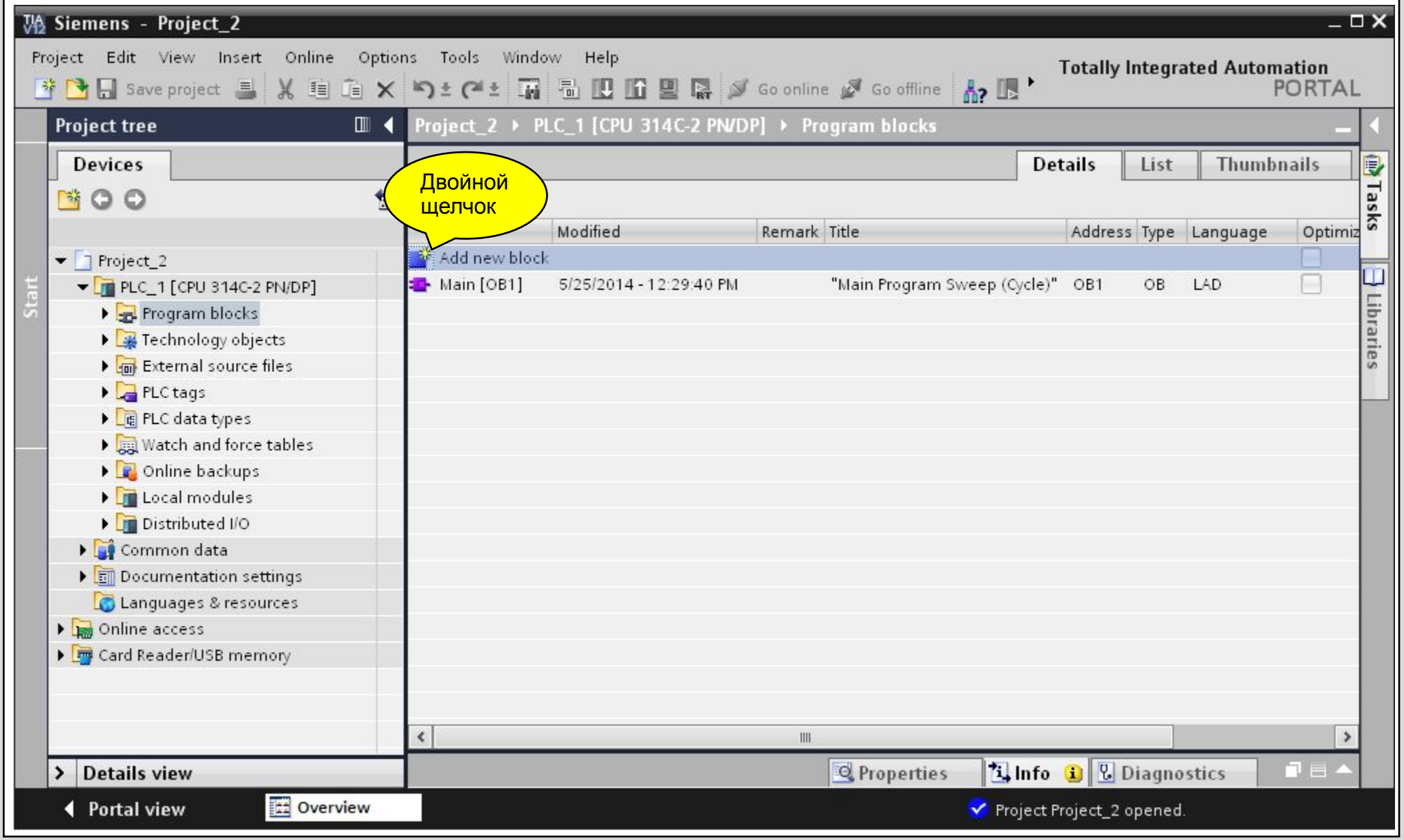
Download without reinitialization

Memory reserve: 100 Bytes

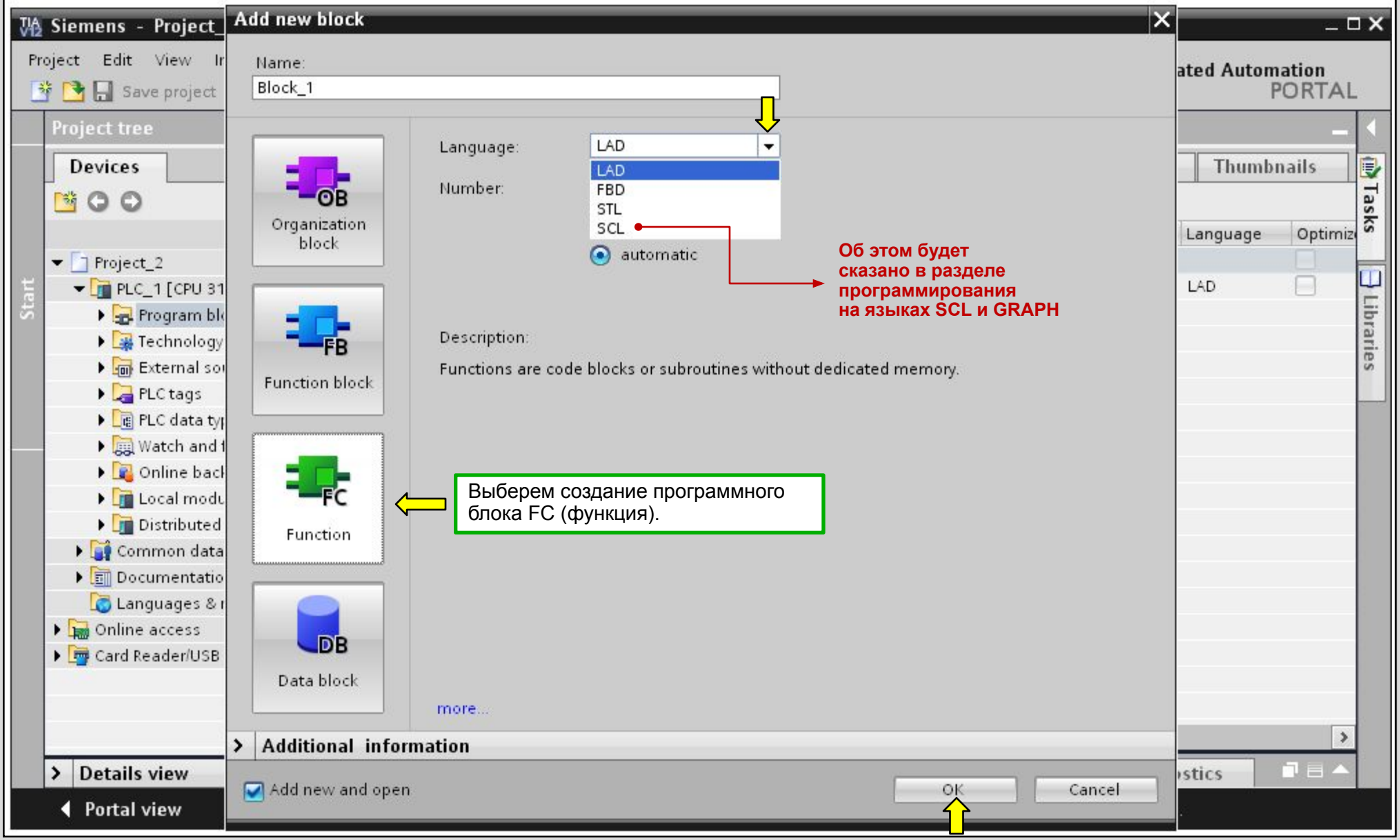
Block interface

Set "Accessible from HMI" for new elements

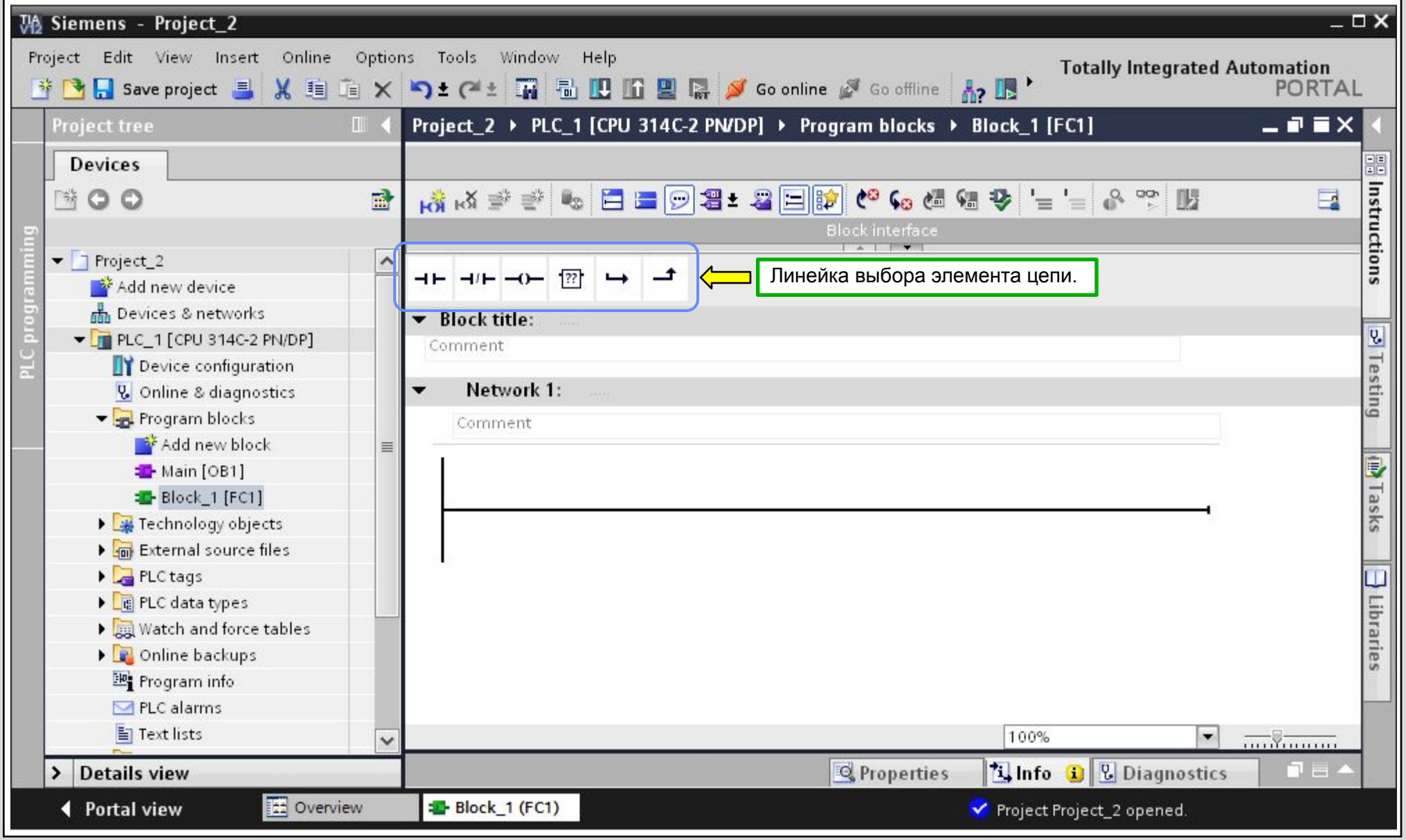
Создание программного блока FC



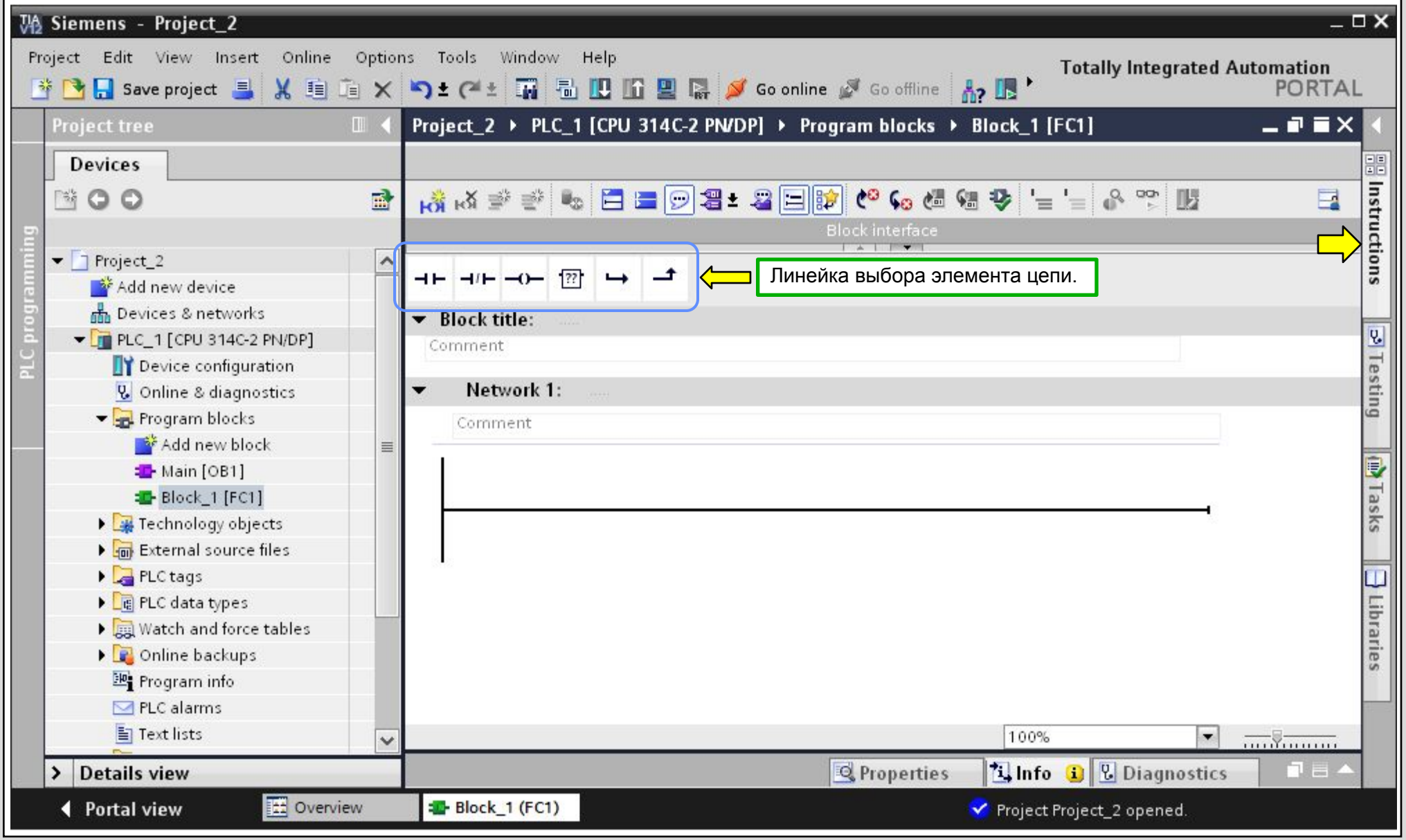
Создание программного блока FC



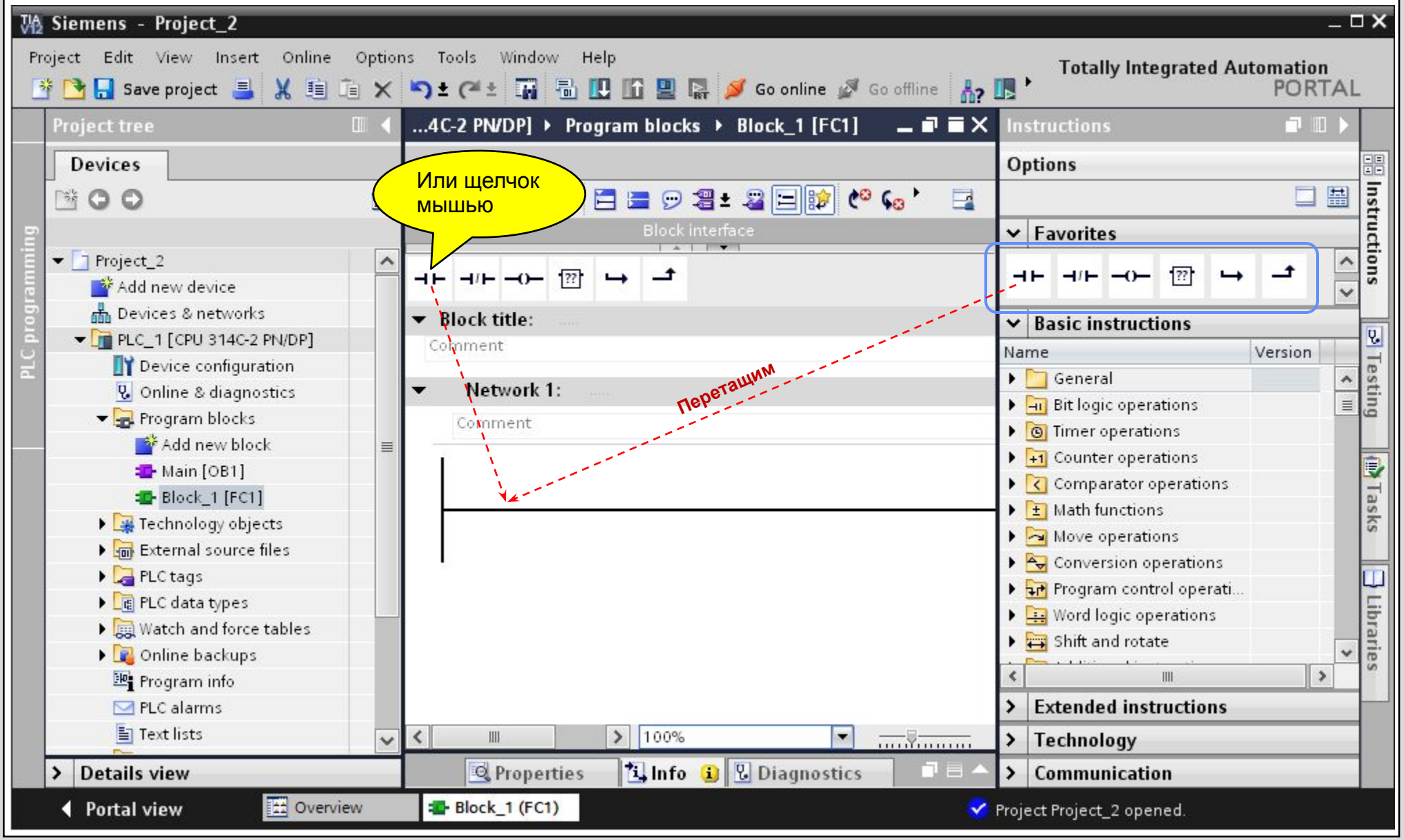
Выбор инструментов построения цепи в языке LAD



Выбор инструментов построения цепи в языке LAD



Выбор инструментов построения цепи в языке LAD



Выбор и адресация элемента цепи

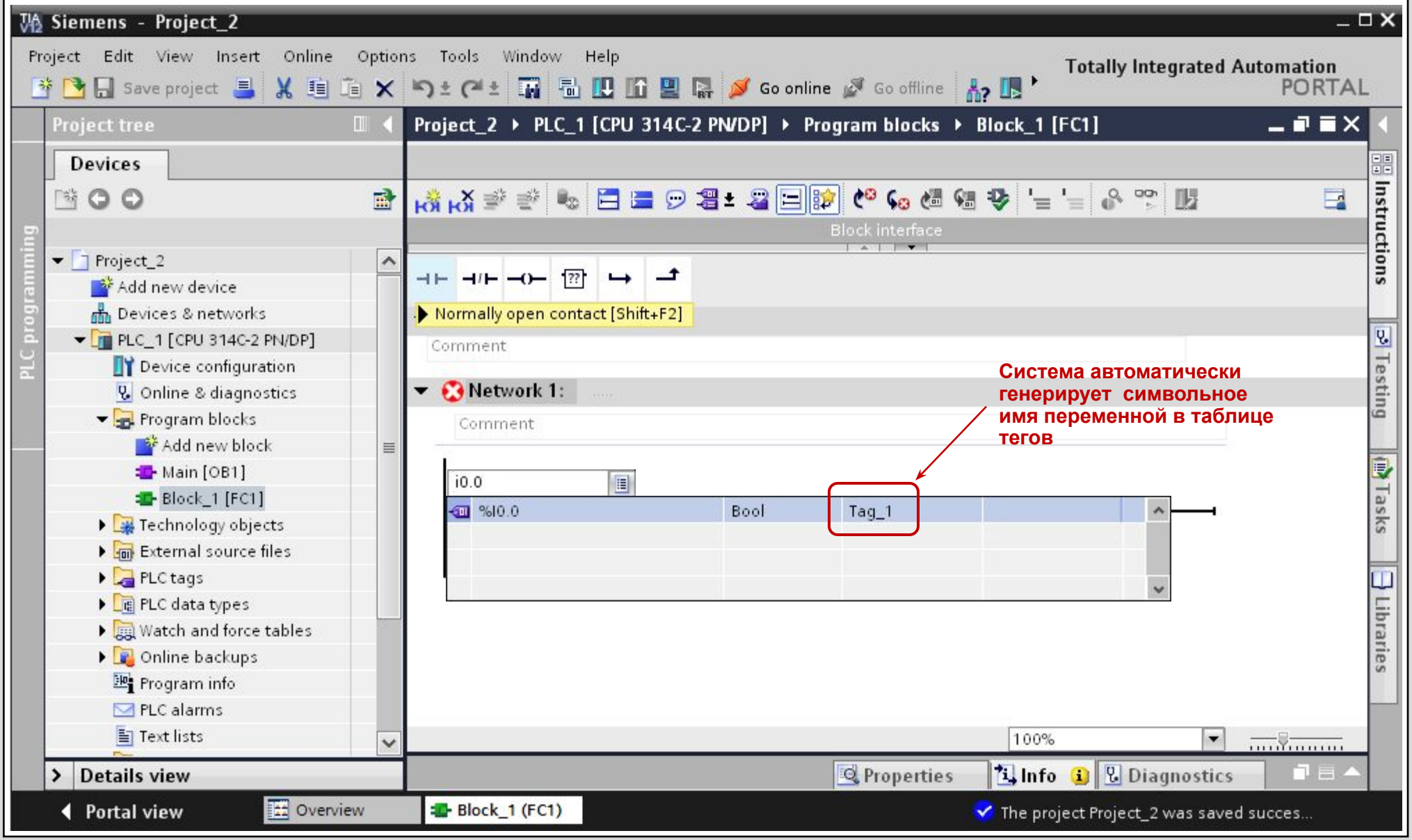
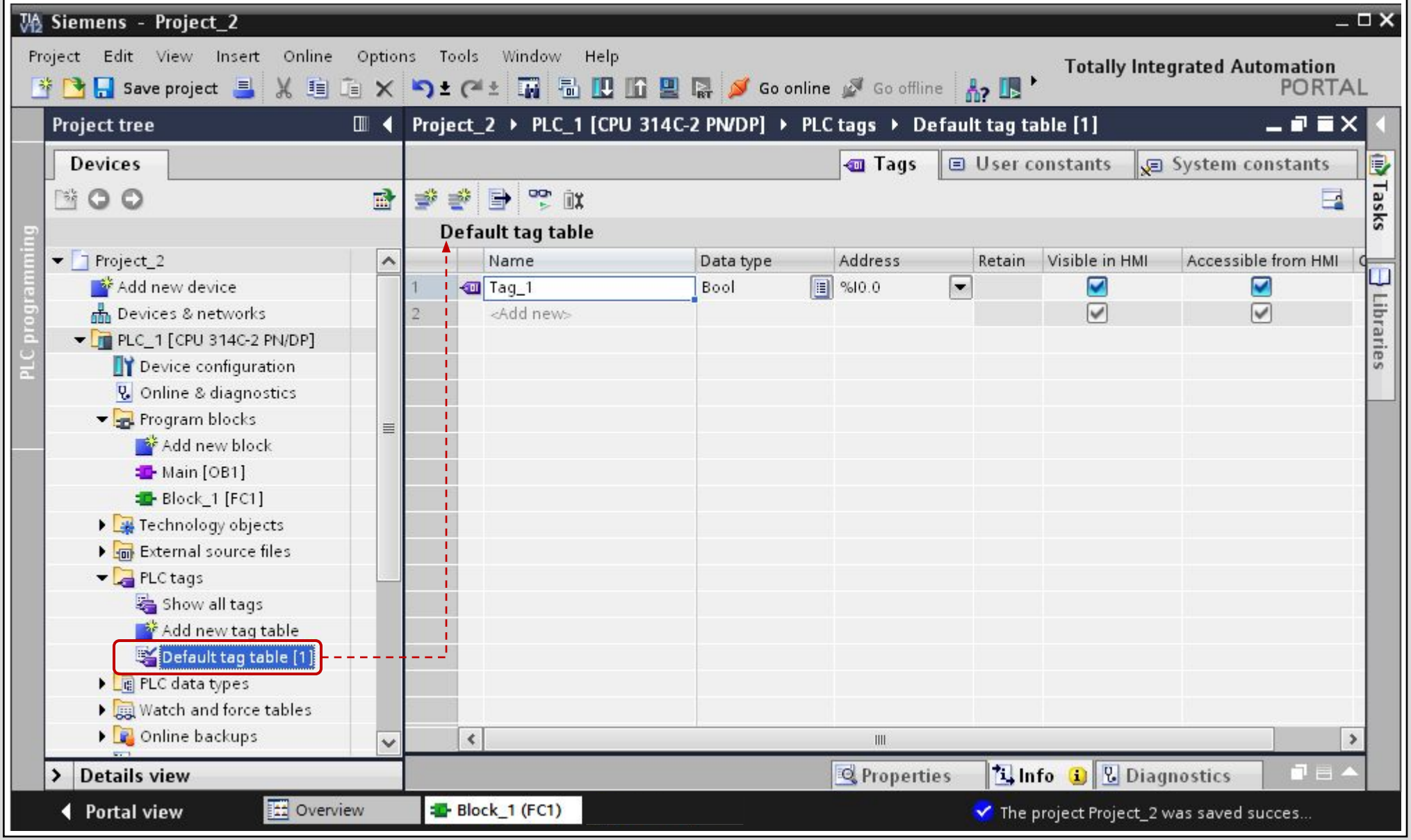
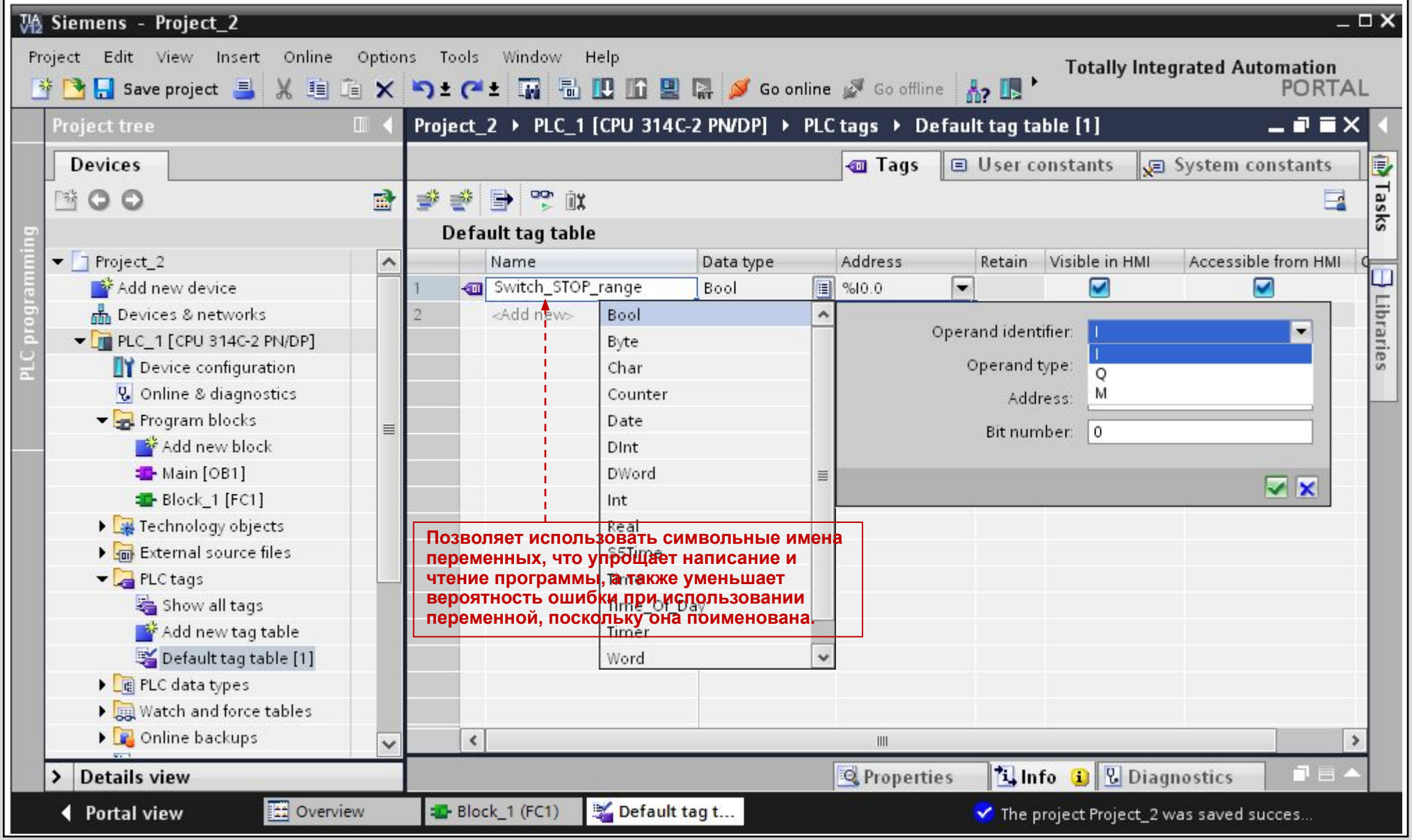


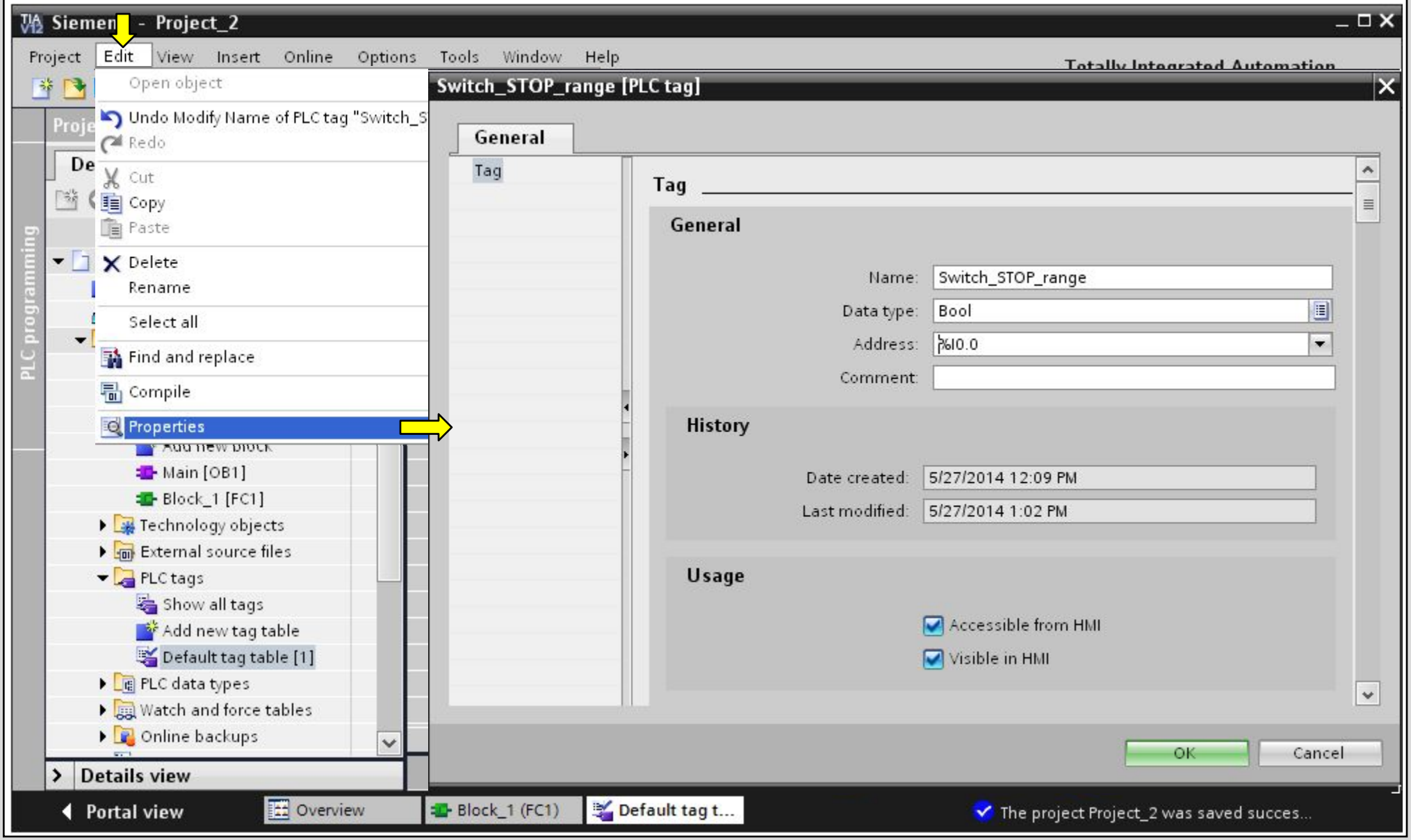
Таблица тегов PLC. Символьное имя переменной



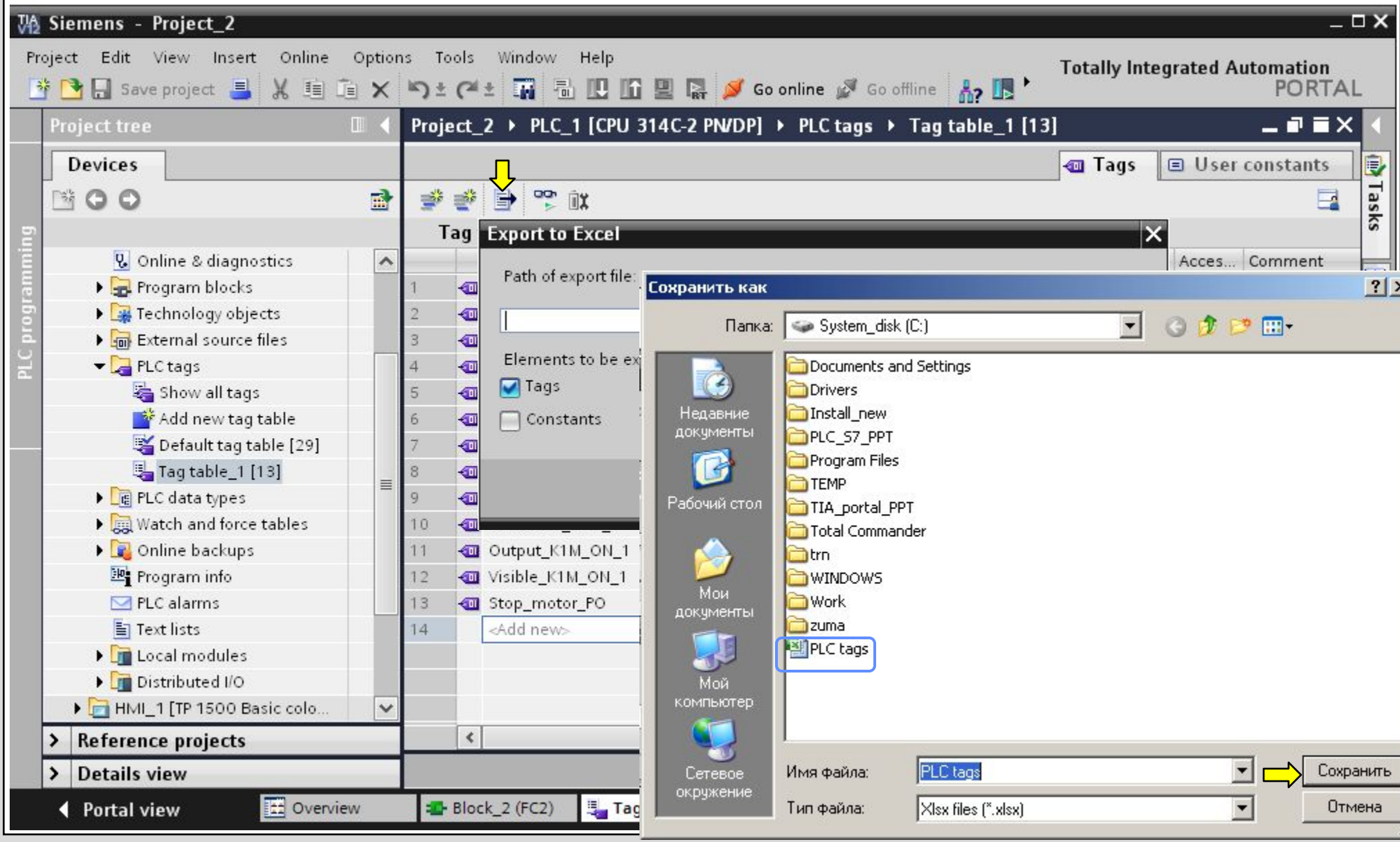
Типы переменных в таблице тегов.



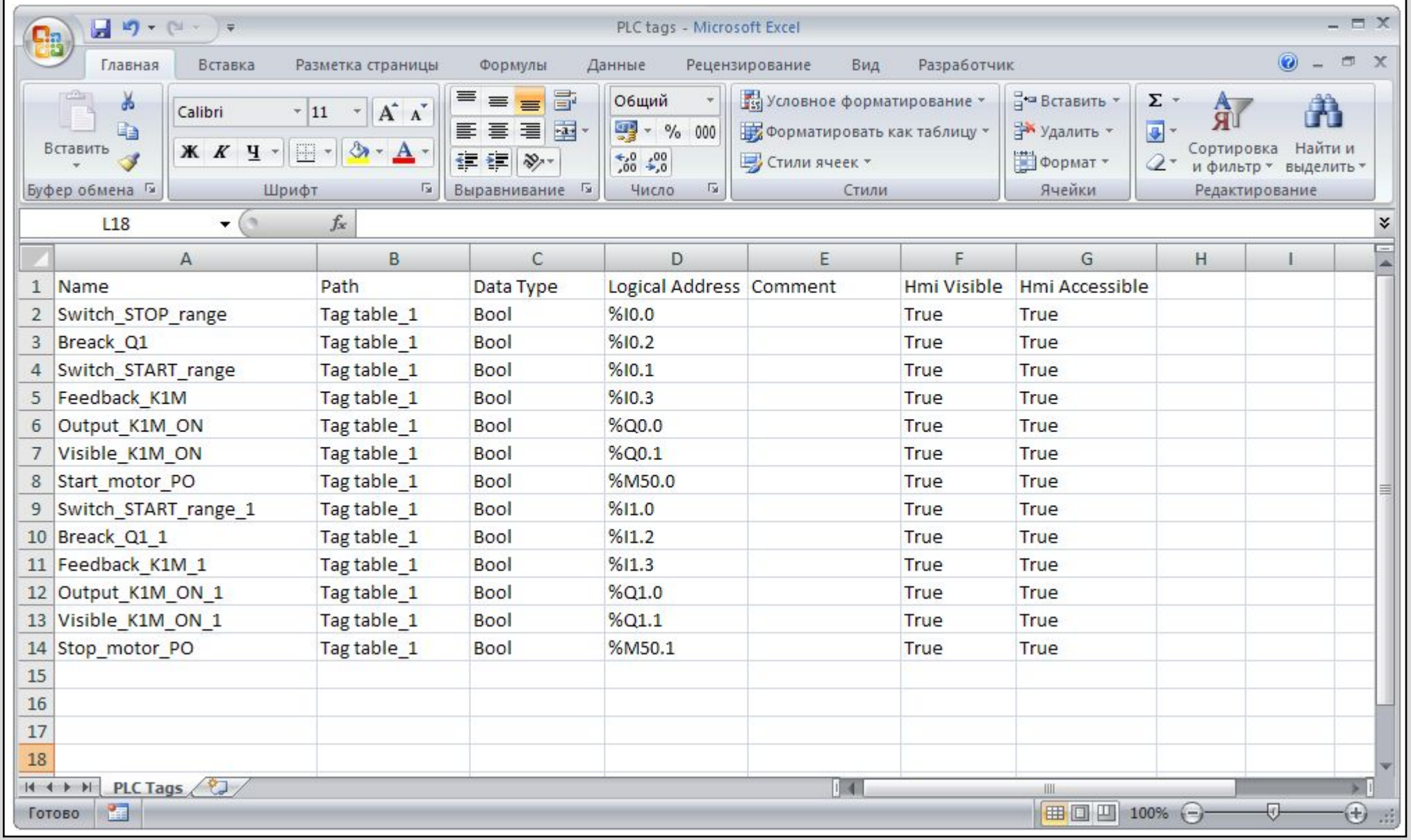
Просмотр свойств выбранного тега



Экспорт таблицы тегов в файл Excel

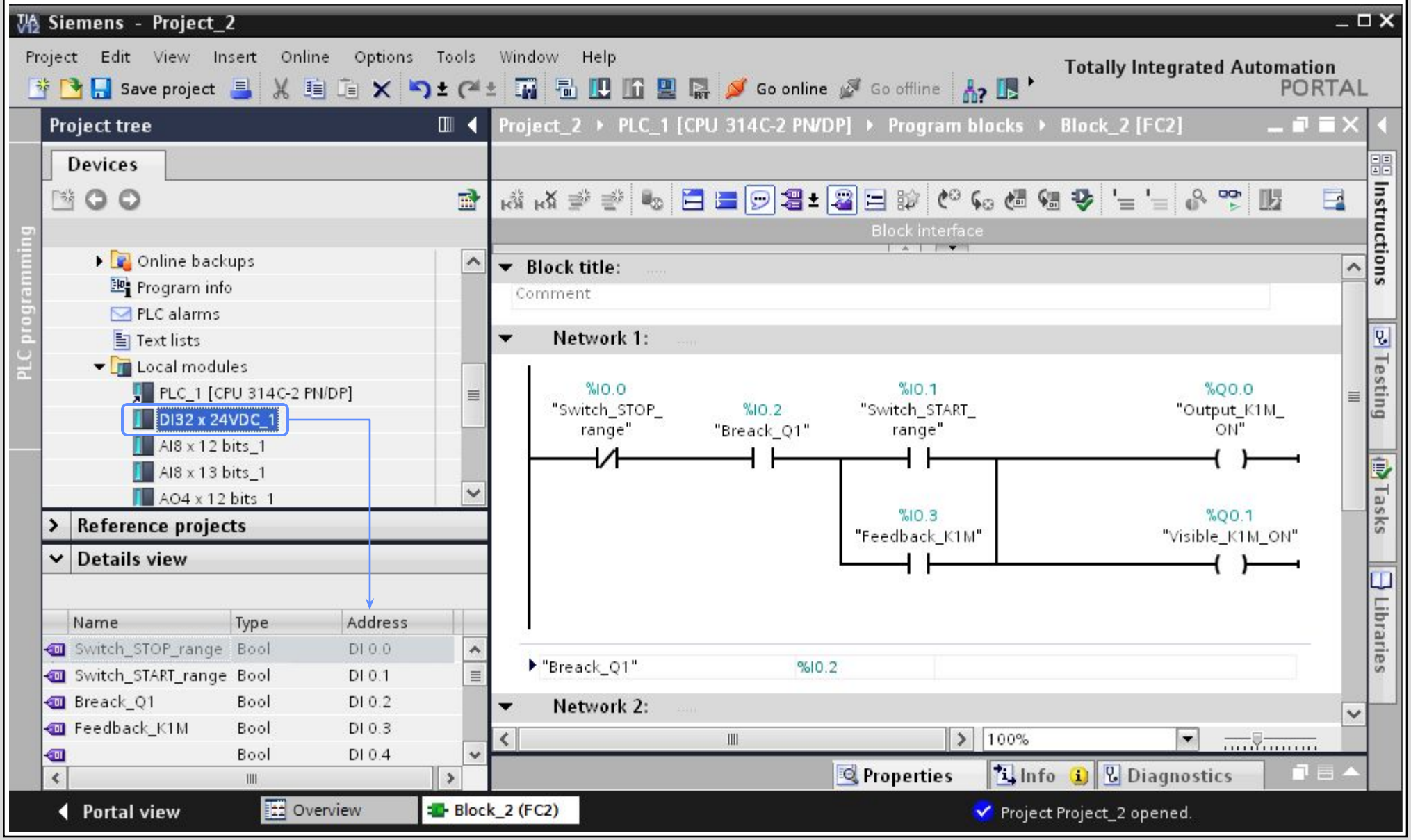


Вид таблицы тегов в формате файла Exel

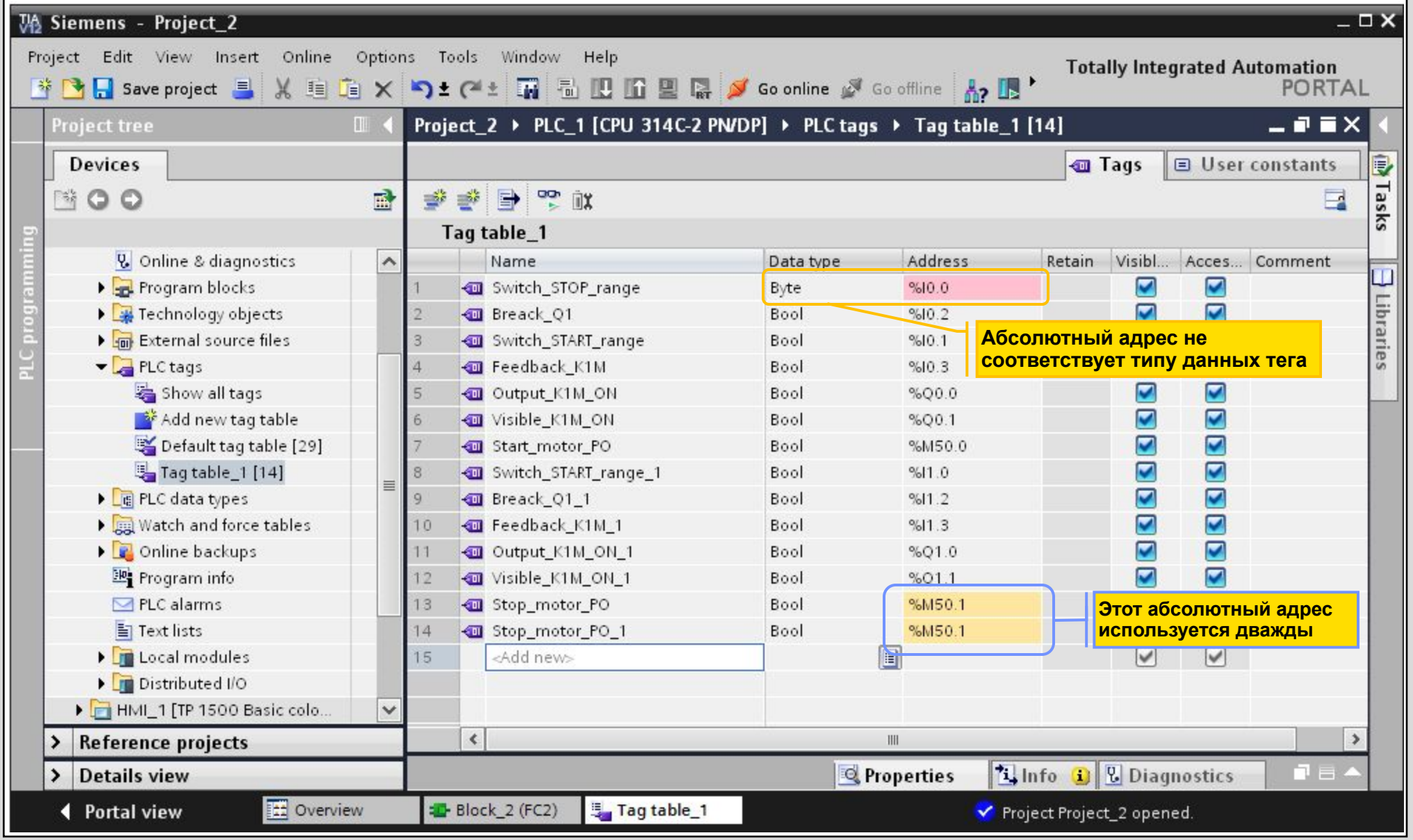


	A	B	C	D	E	F	G	H	I
1	Name	Path	Data Type	Logical Address	Comment	Hmi Visible	Hmi Accessible		
2	Switch_STOP_range	Tag table_1	Bool	%I0.0		True	True		
3	Breack_Q1	Tag table_1	Bool	%I0.2		True	True		
4	Switch_START_range	Tag table_1	Bool	%I0.1		True	True		
5	Feedback_K1M	Tag table_1	Bool	%I0.3		True	True		
6	Output_K1M_ON	Tag table_1	Bool	%Q0.0		True	True		
7	Visible_K1M_ON	Tag table_1	Bool	%Q0.1		True	True		
8	Start_motor_PO	Tag table_1	Bool	%M50.0		True	True		
9	Switch_START_range_1	Tag table_1	Bool	%I1.0		True	True		
10	Breack_Q1_1	Tag table_1	Bool	%I1.2		True	True		
11	Feedback_K1M_1	Tag table_1	Bool	%I1.3		True	True		
12	Output_K1M_ON_1	Tag table_1	Bool	%Q1.0		True	True		
13	Visible_K1M_ON_1	Tag table_1	Bool	%Q1.1		True	True		
14	Stop_motor_PO	Tag table_1	Bool	%M50.1		True	True		
15									
16									
17									
18									

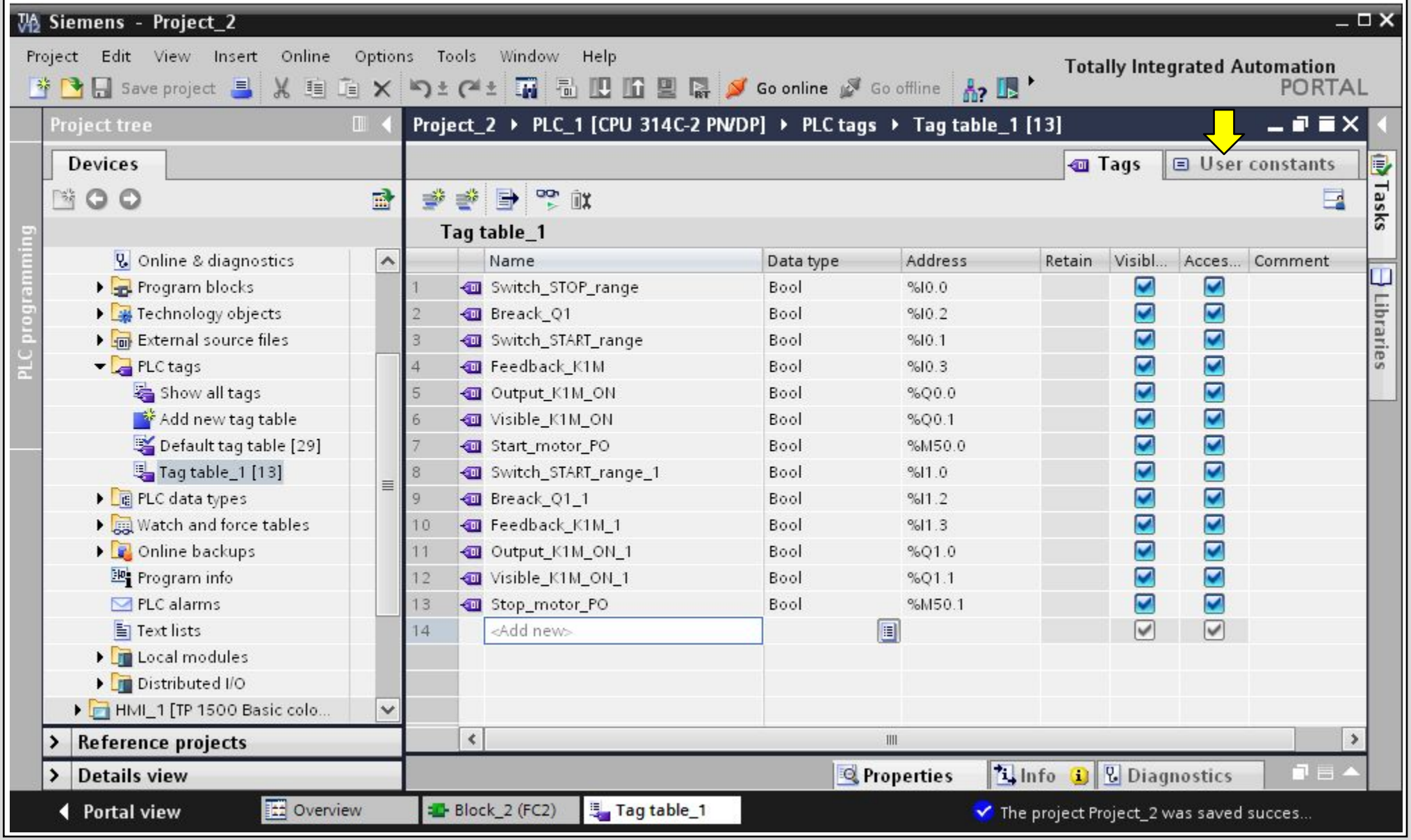
Просмотр символьной таблицы для адресов выбранного модуля



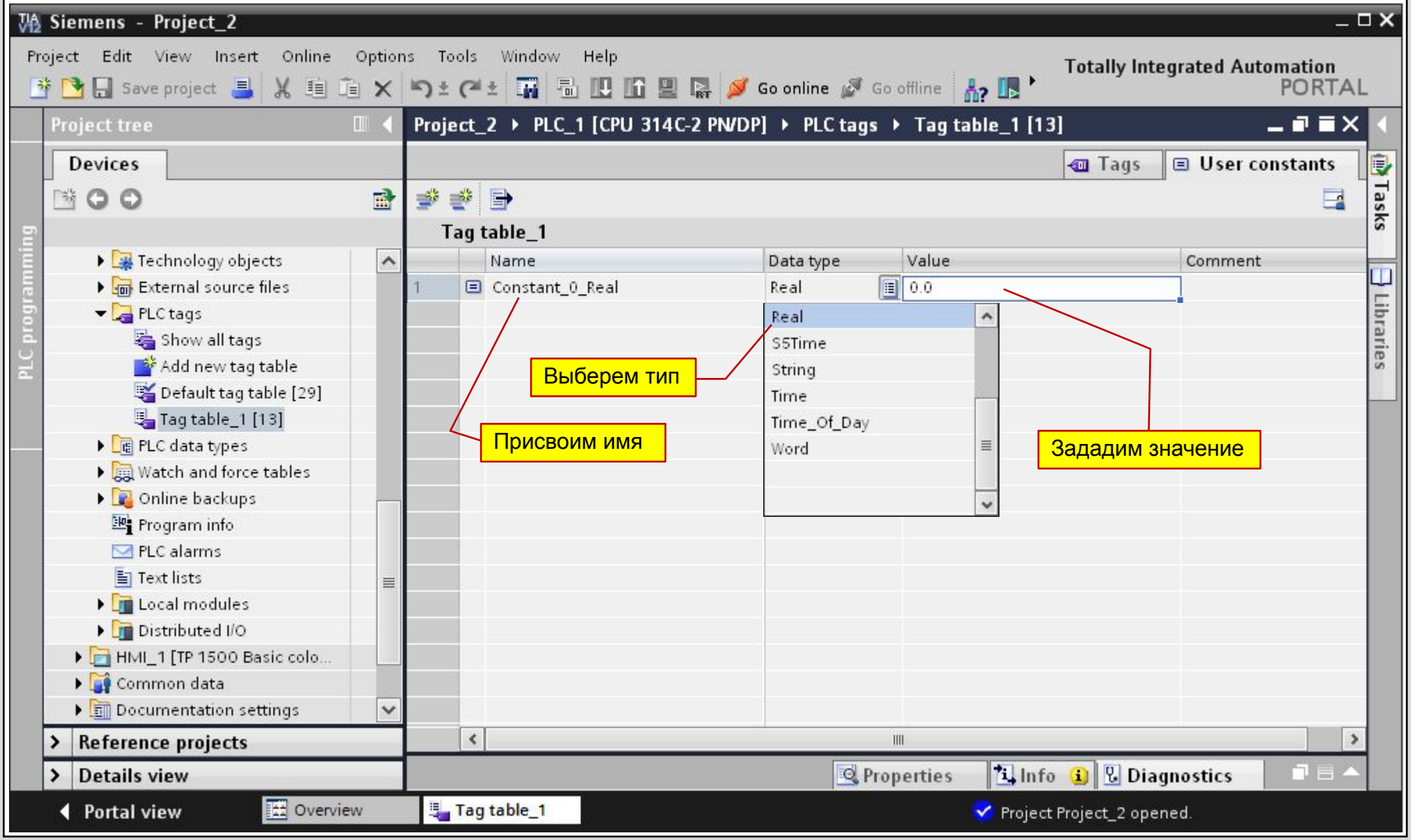
Возможные ошибки определения переменных в таблице тегов



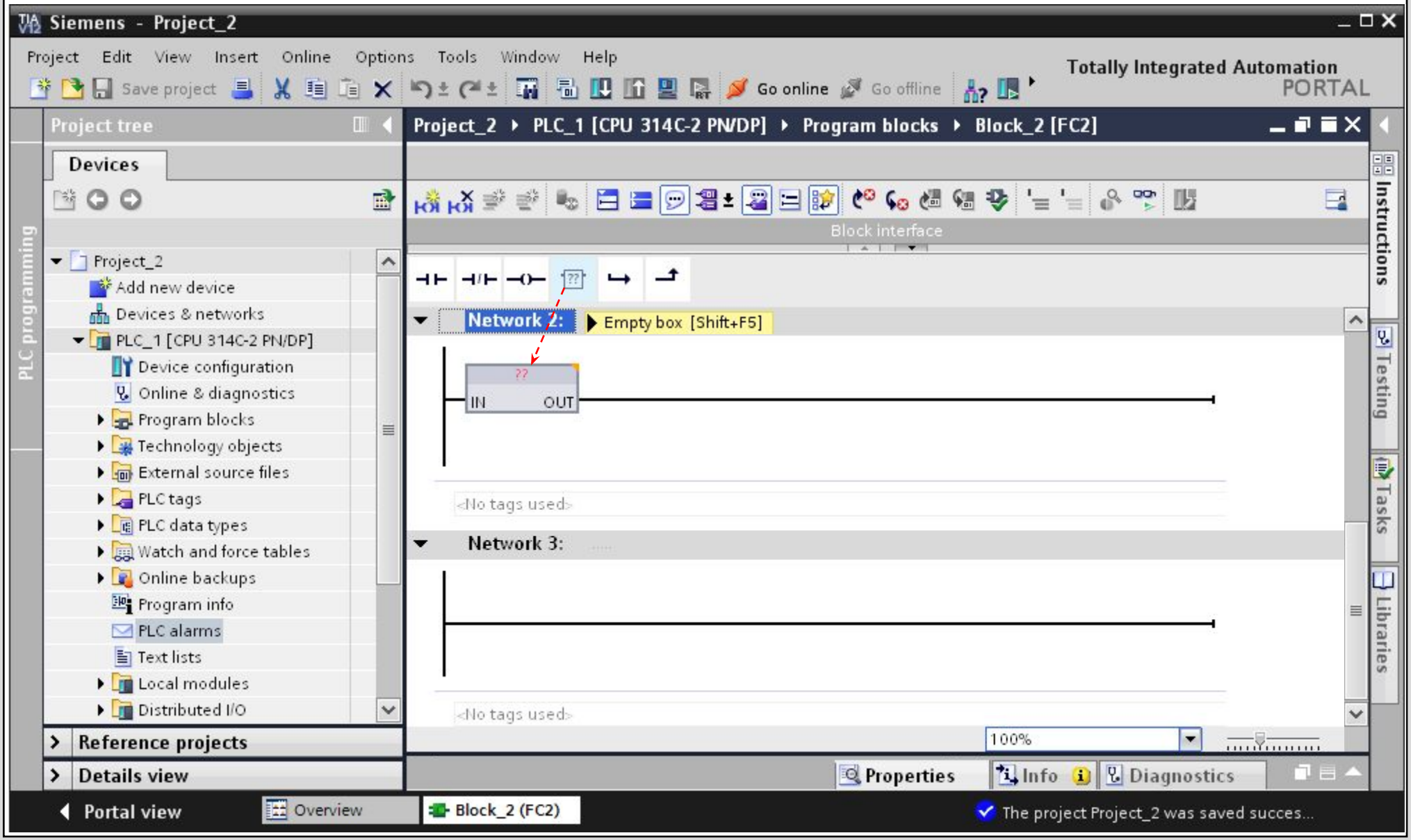
Задание констант пользователя



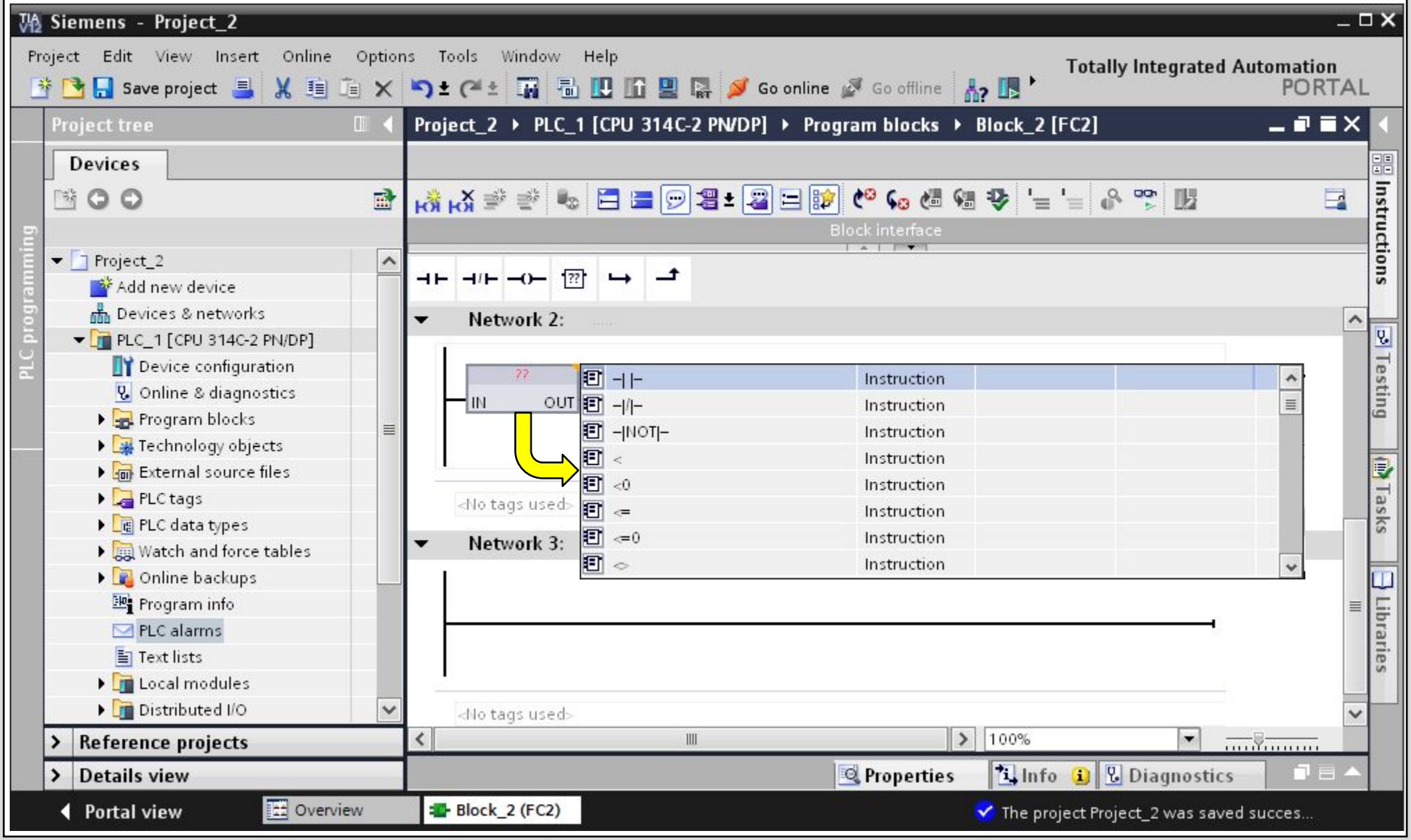
Задание констант пользователя



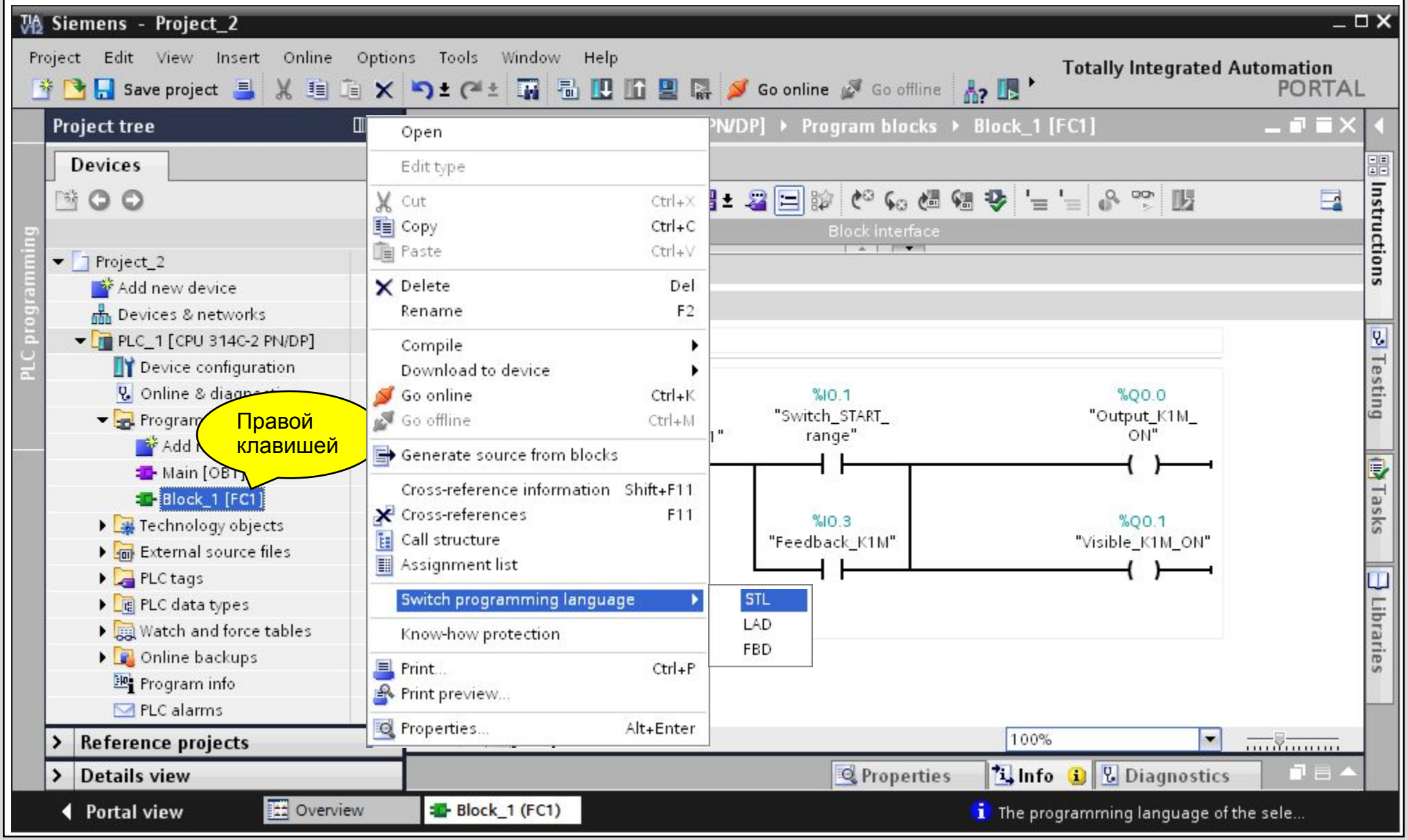
Использование шаблона Empty box



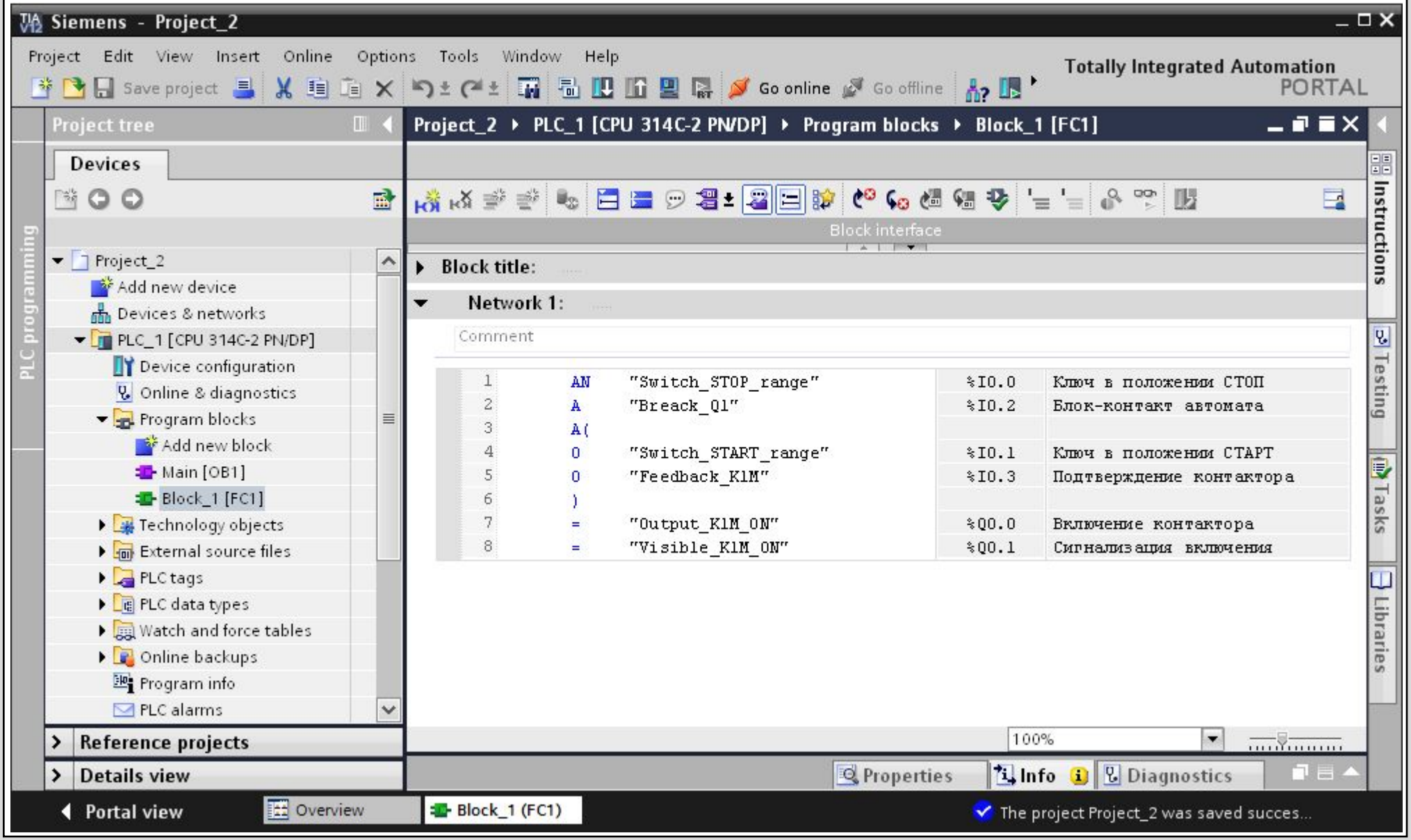
Использование шаблона Empty box



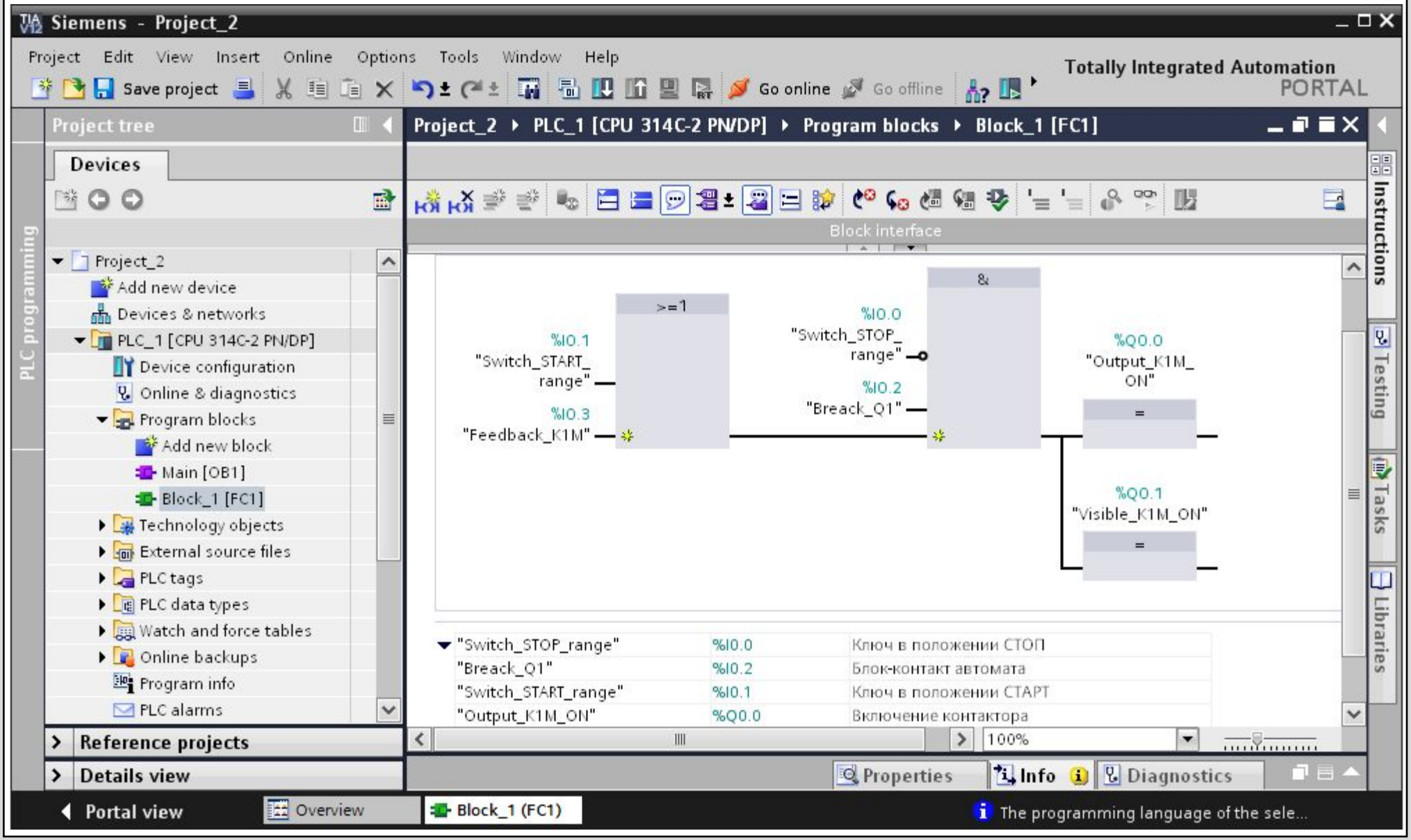
Пример построения цепи в LAD. Переход к другим представлениям.



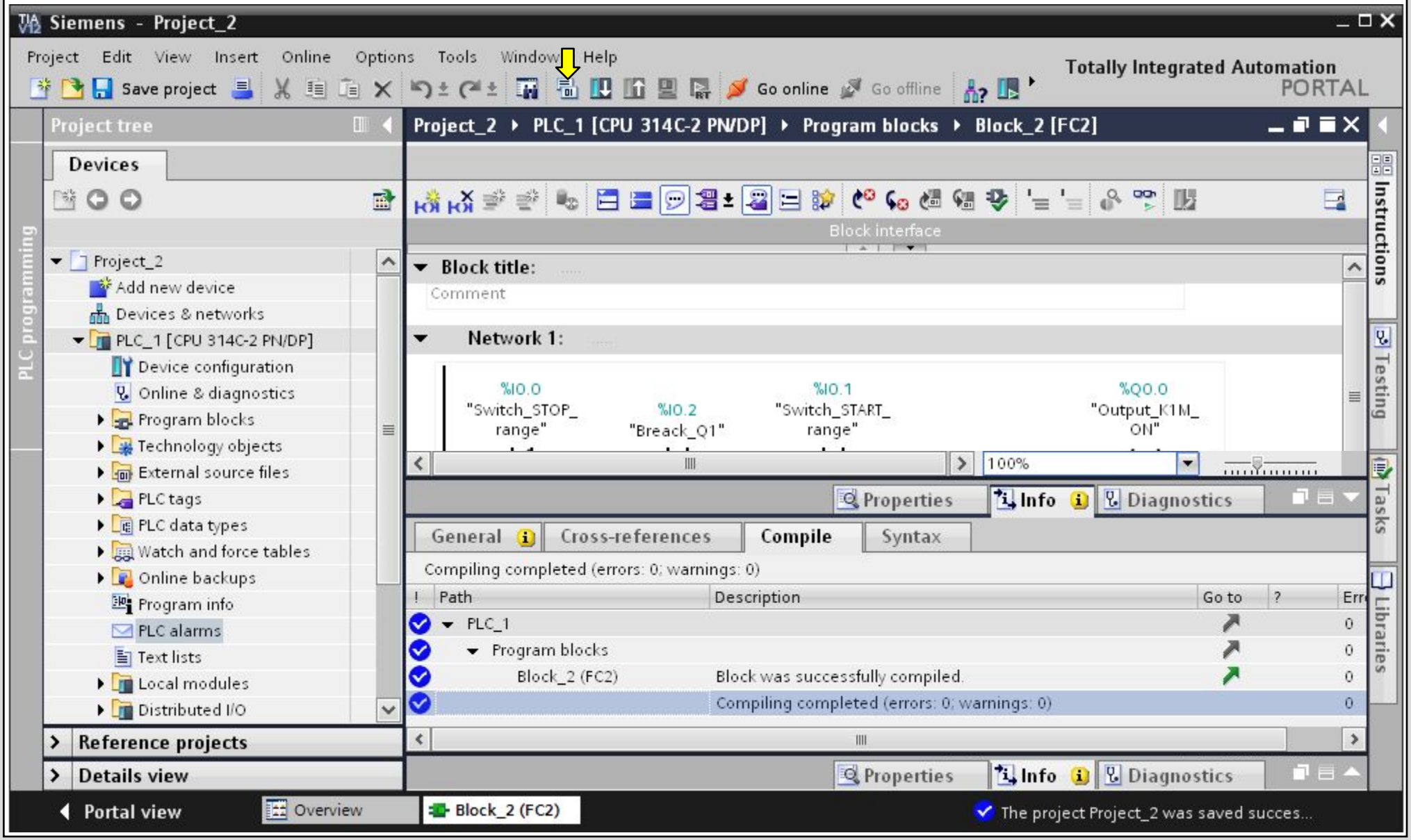
Представление цепи в STL



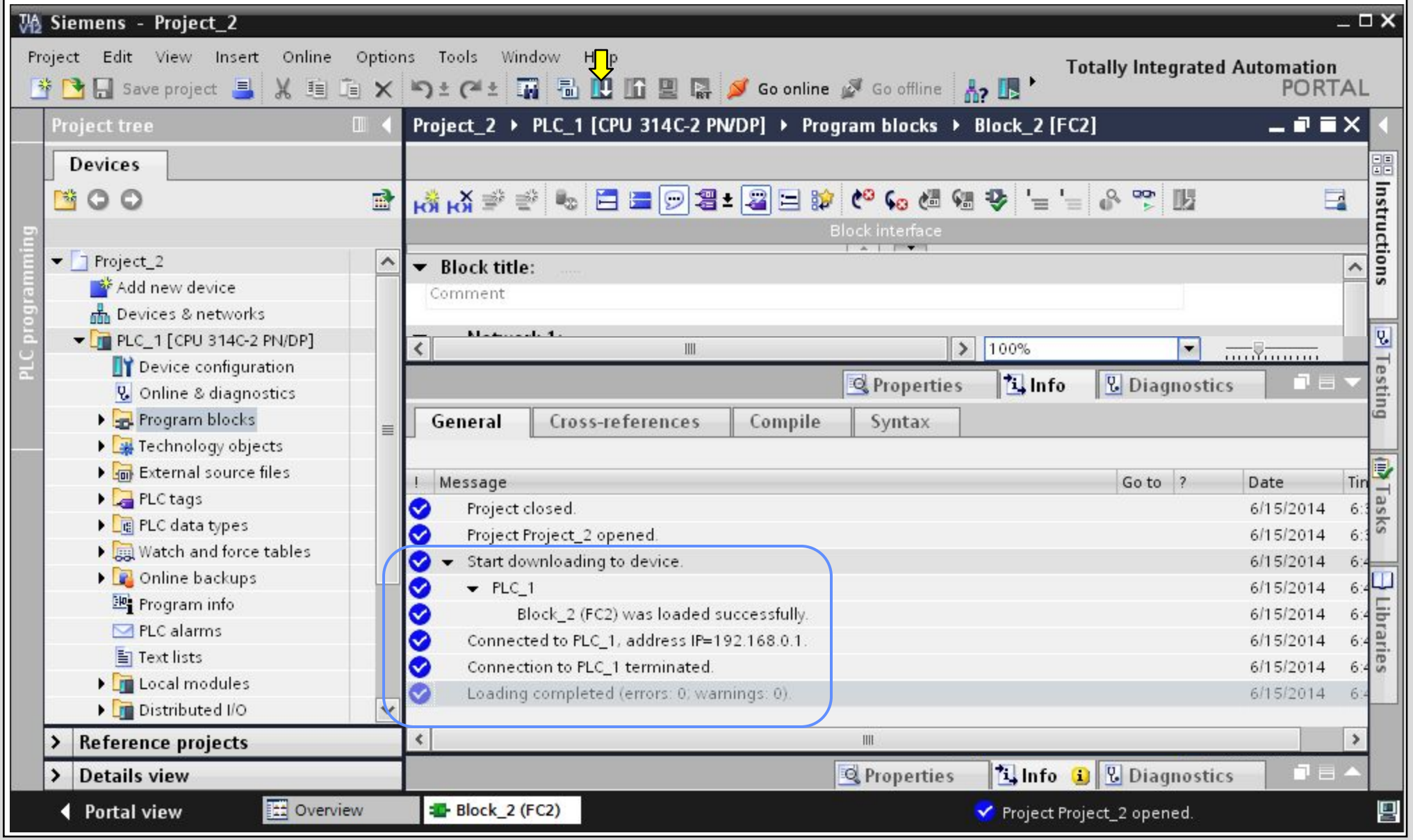
Представление цепи в FBD



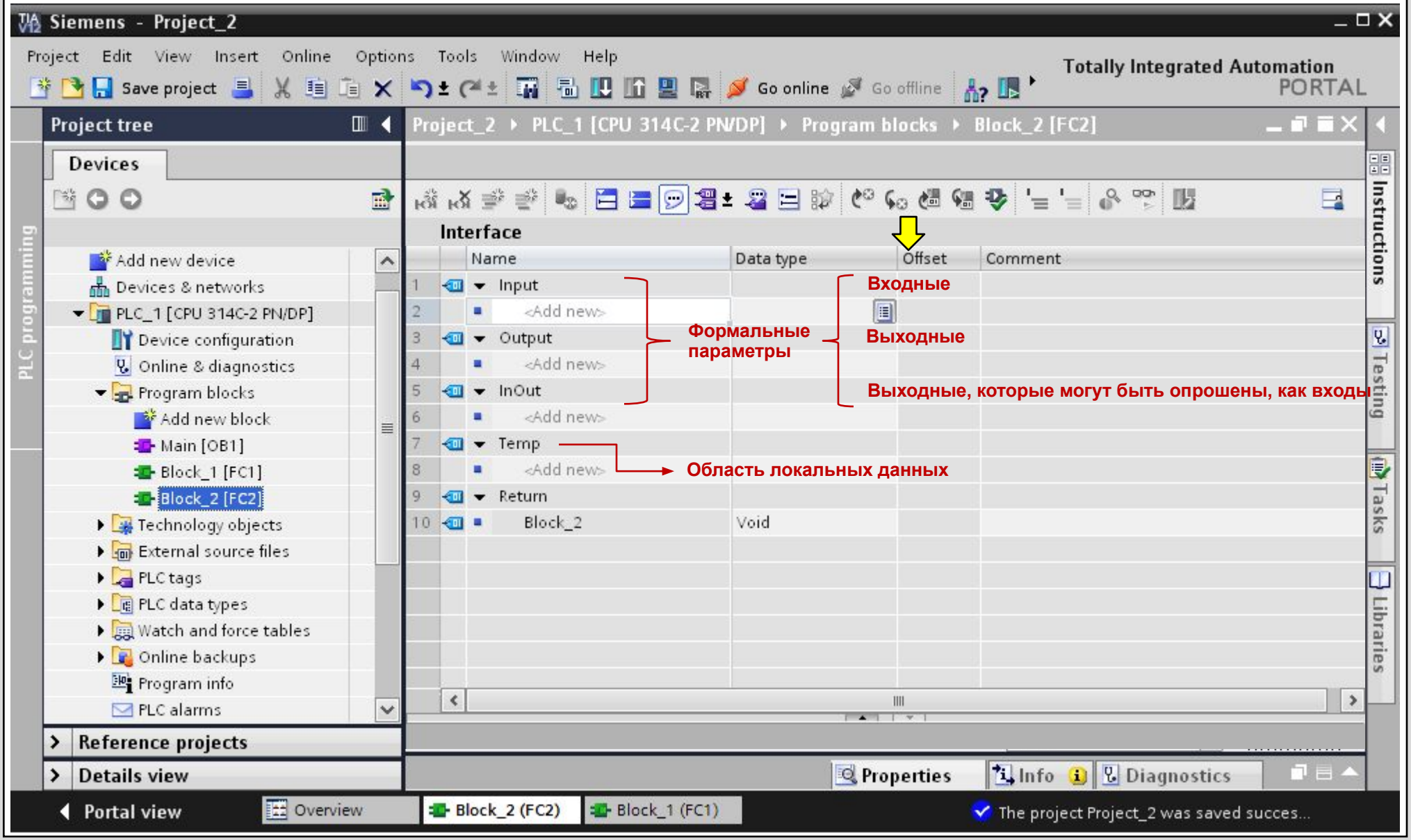
Компиляция программного блока



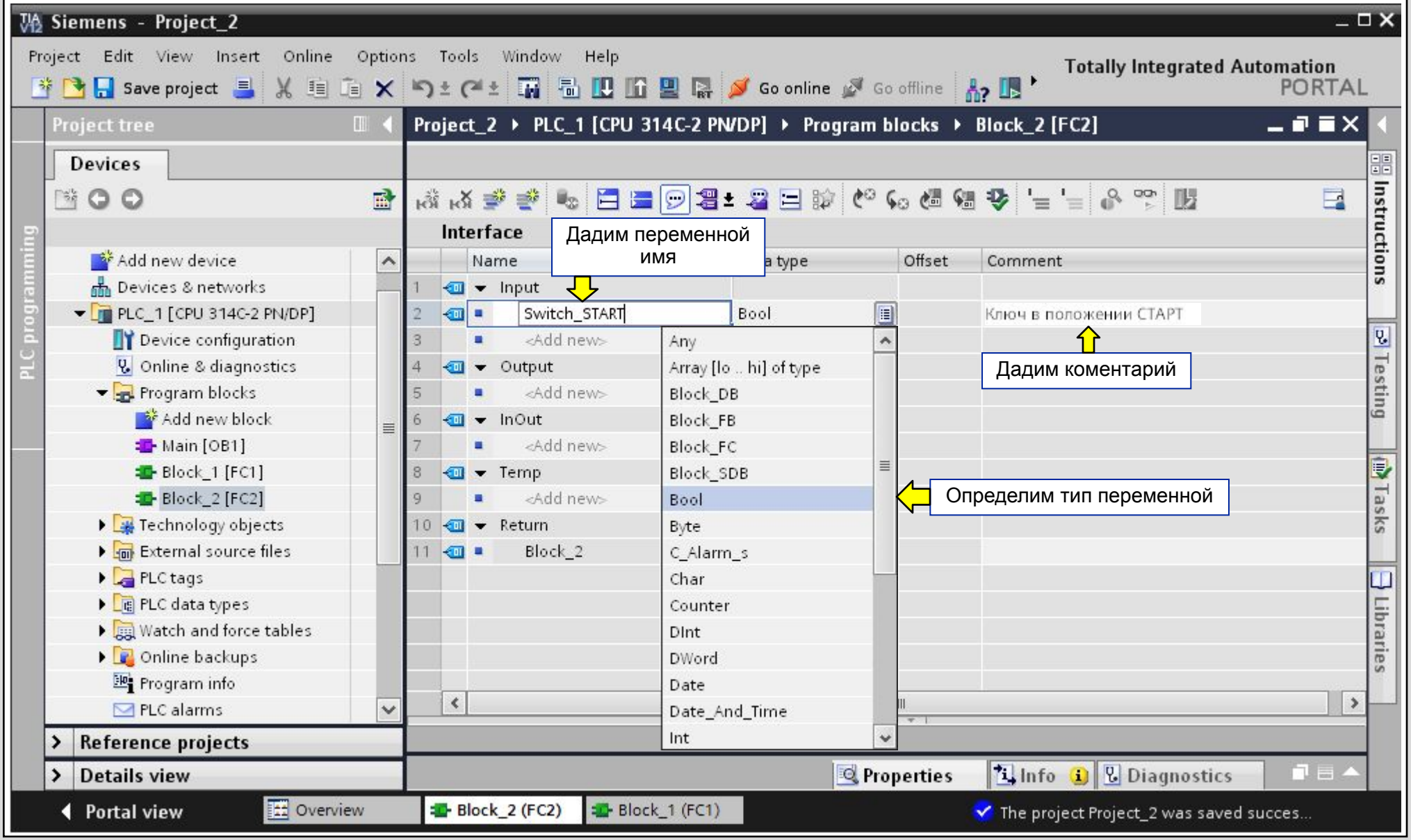
Загрузка программного блока в PLC



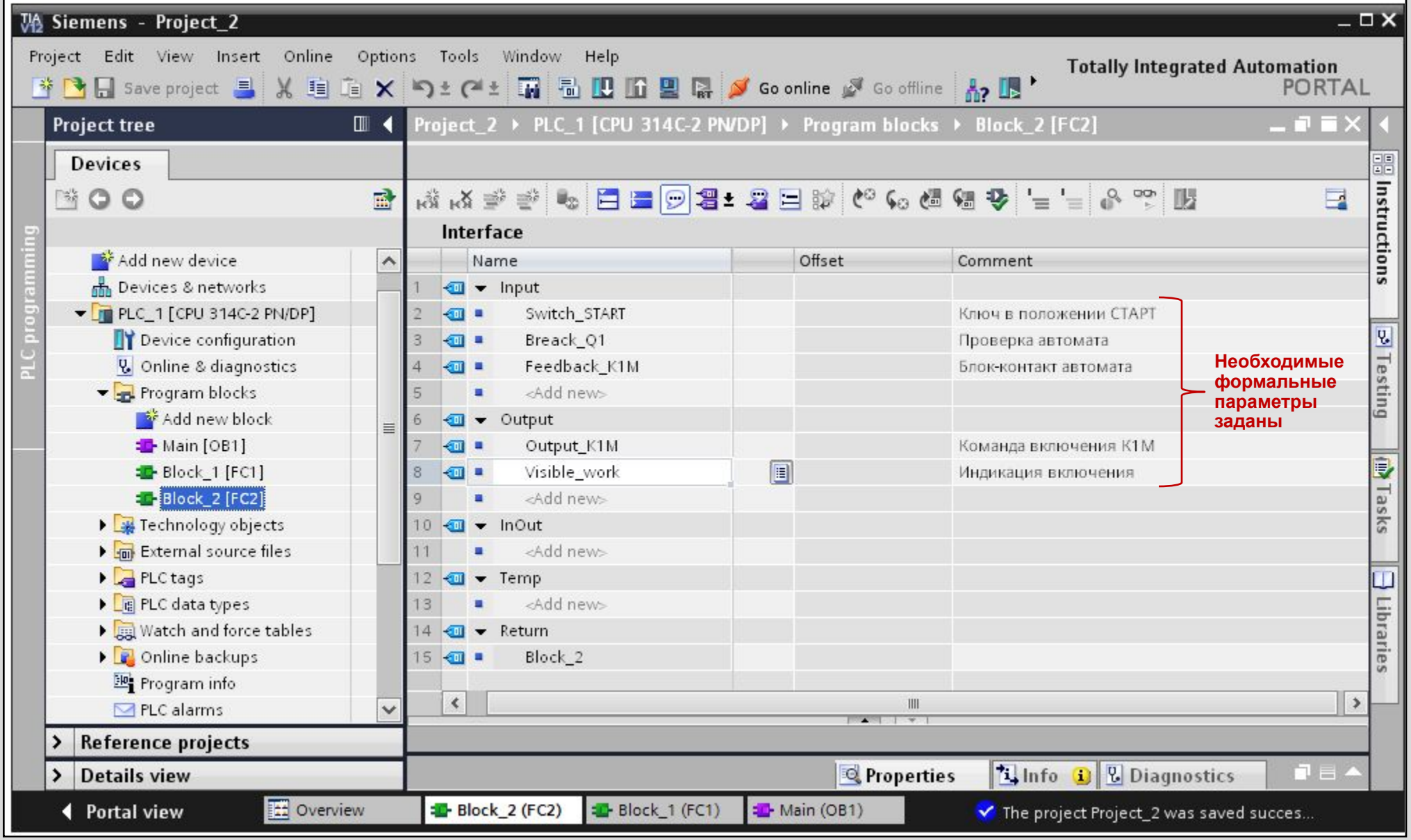
Работа с интерфейсом программного блока. Формальные параметры



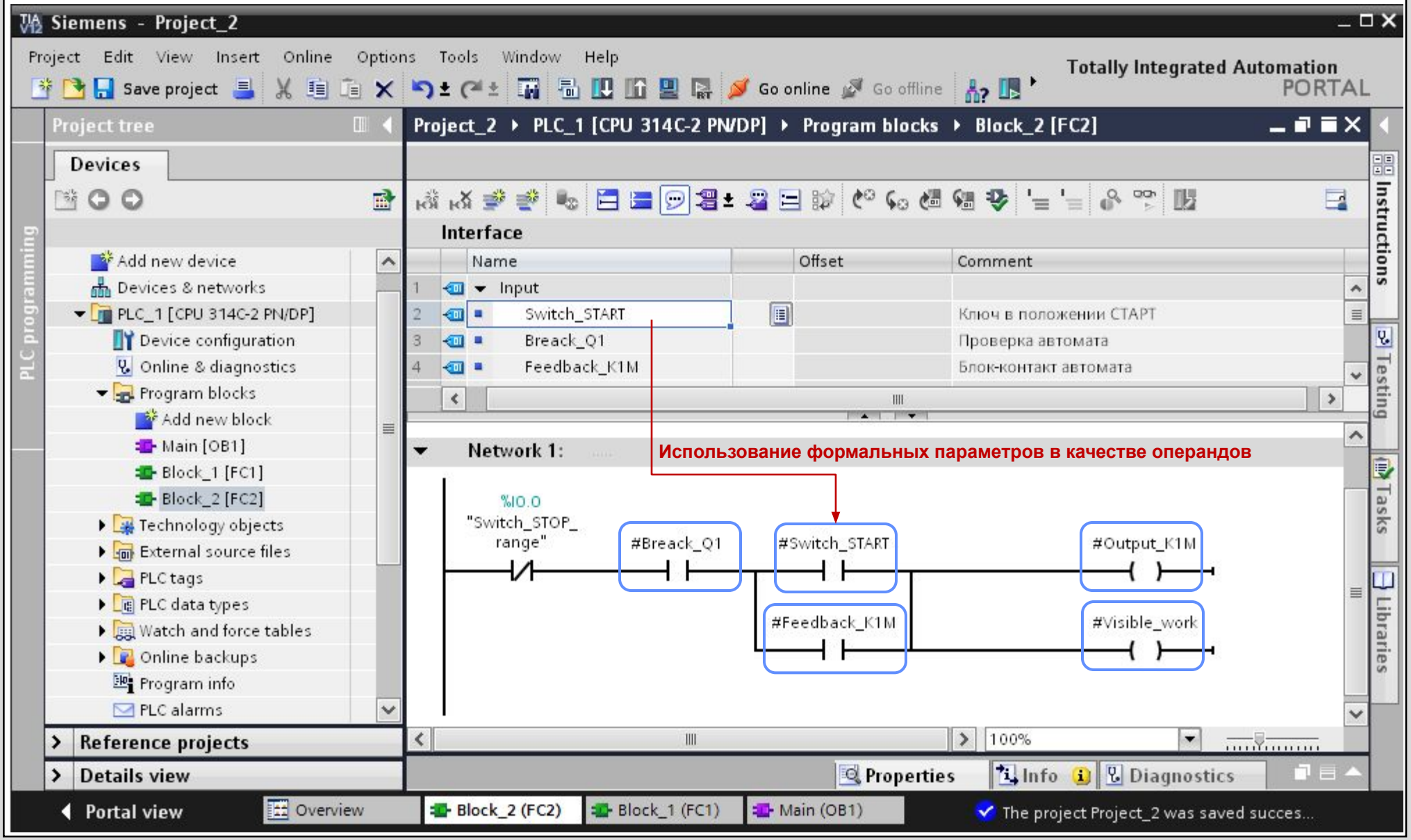
Работа с интерфейсом программного блока



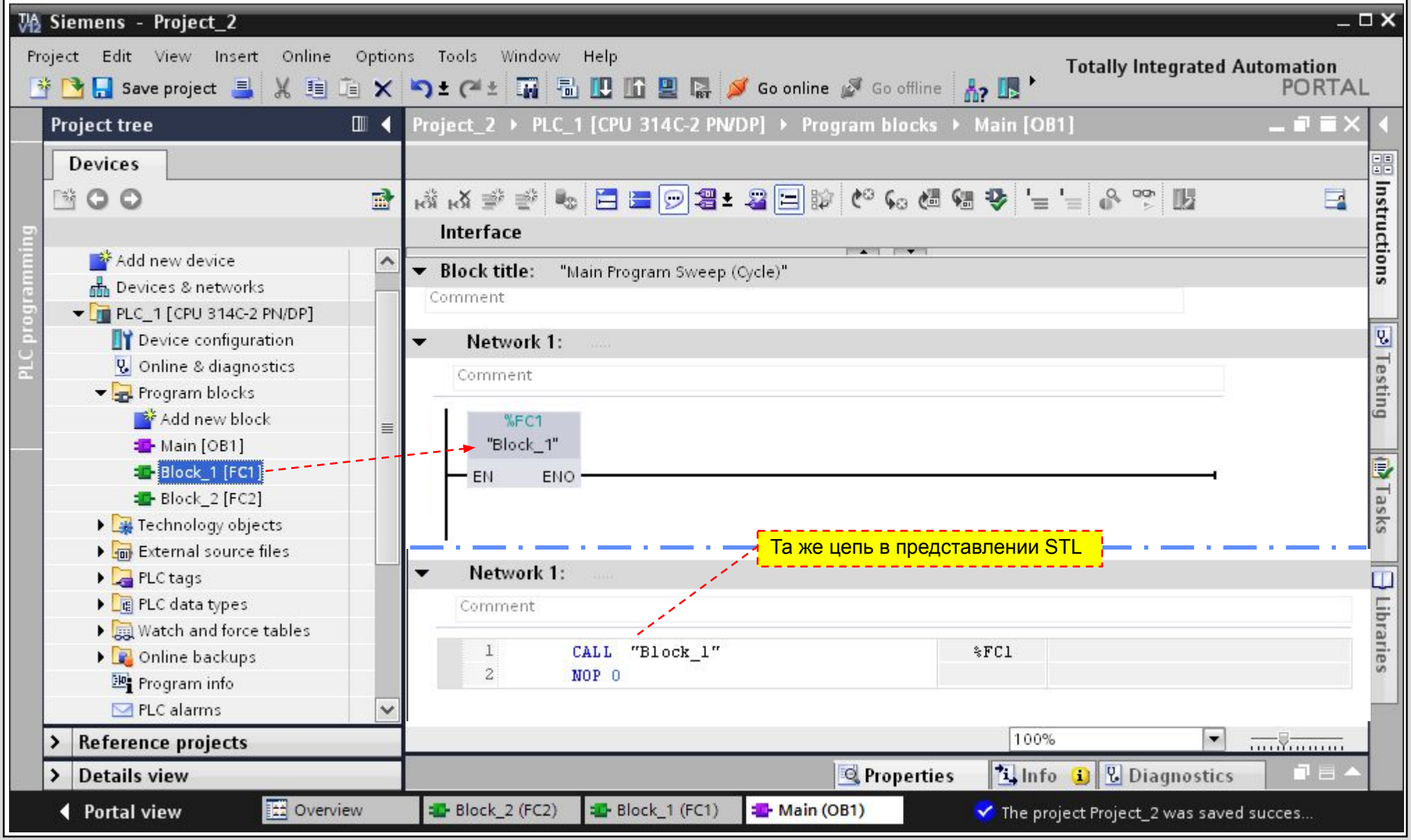
Работа с интерфейсом программного блока



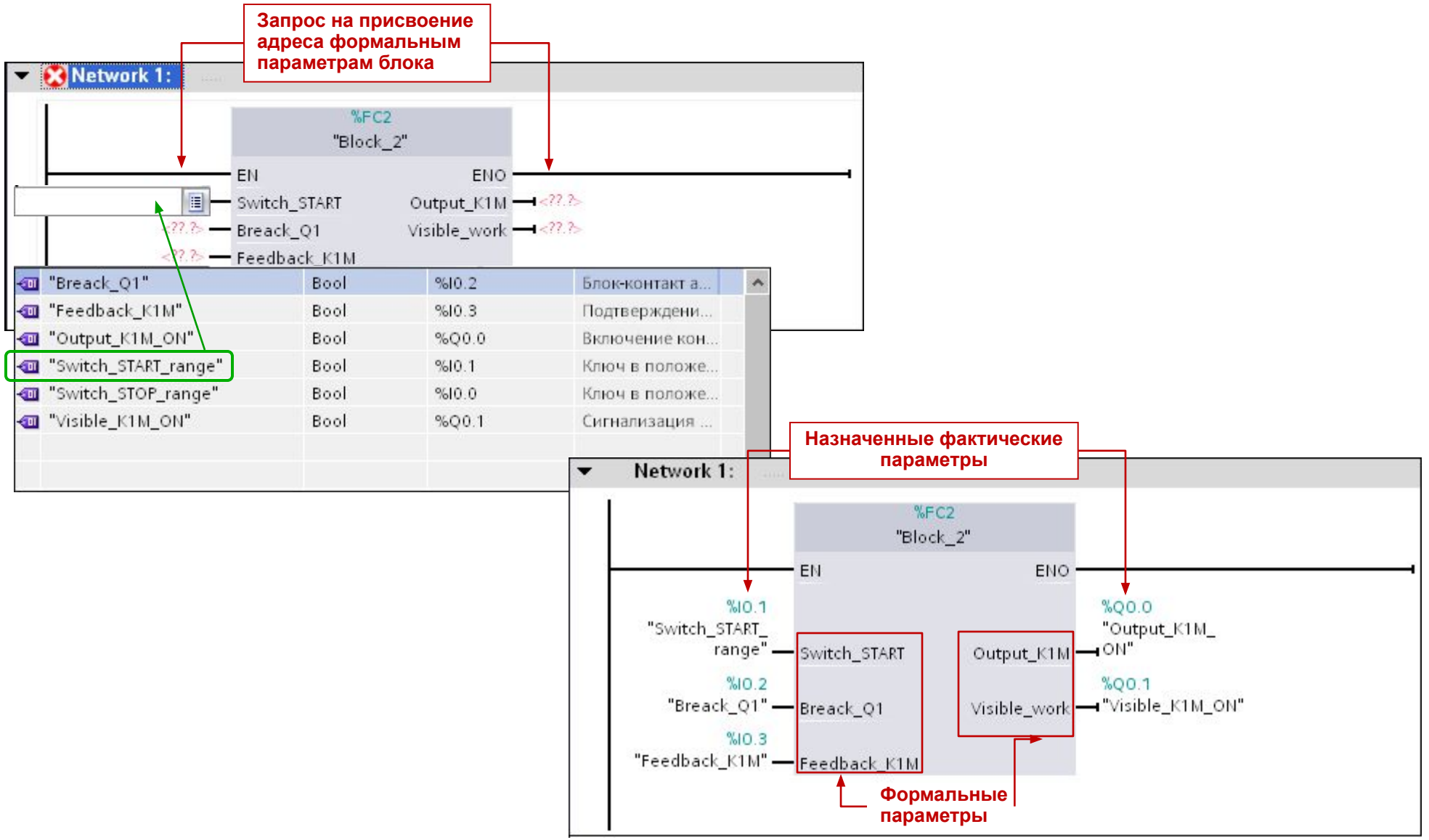
Использование параметров в качестве переменных



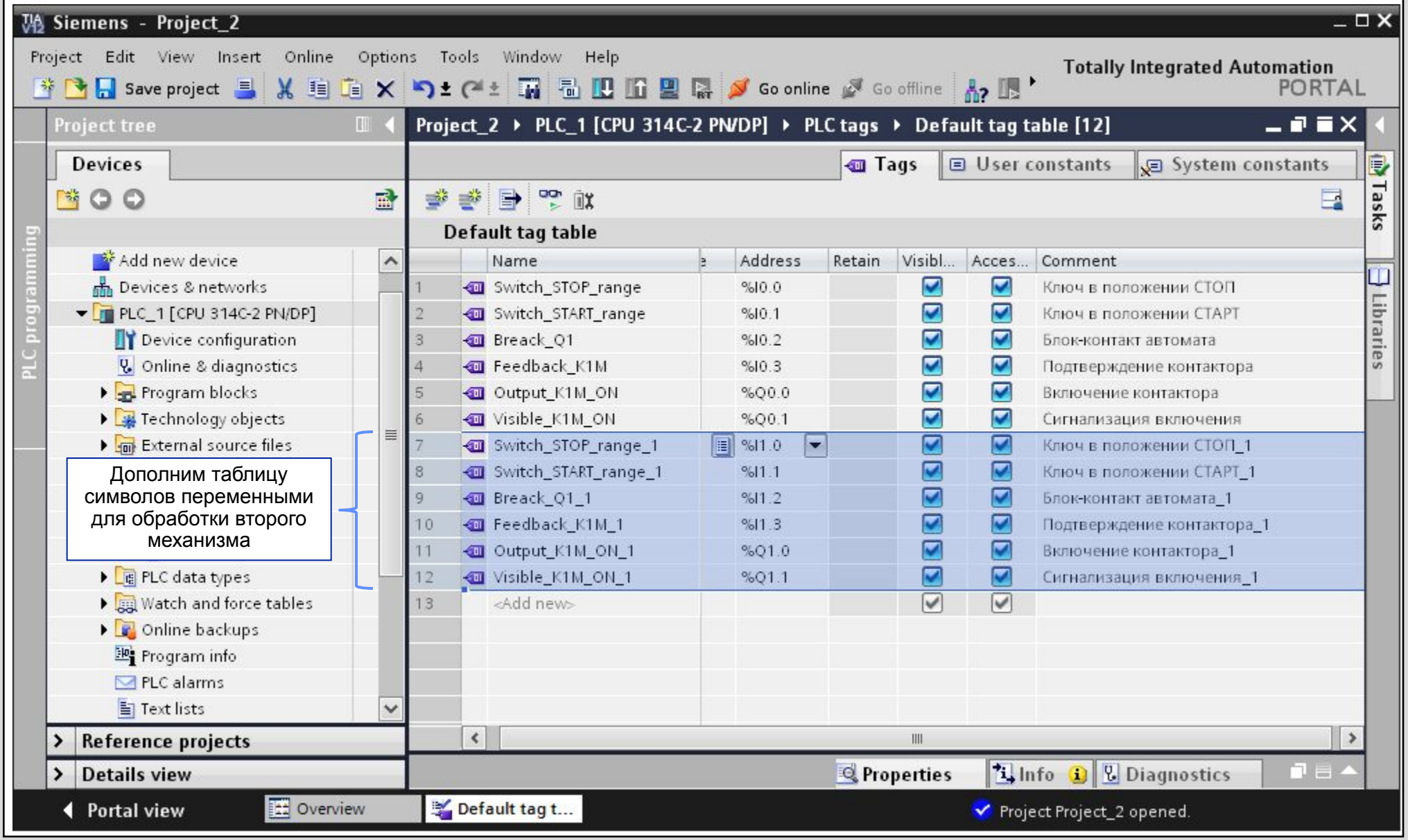
Вызов программного блока без параметров (FC1)



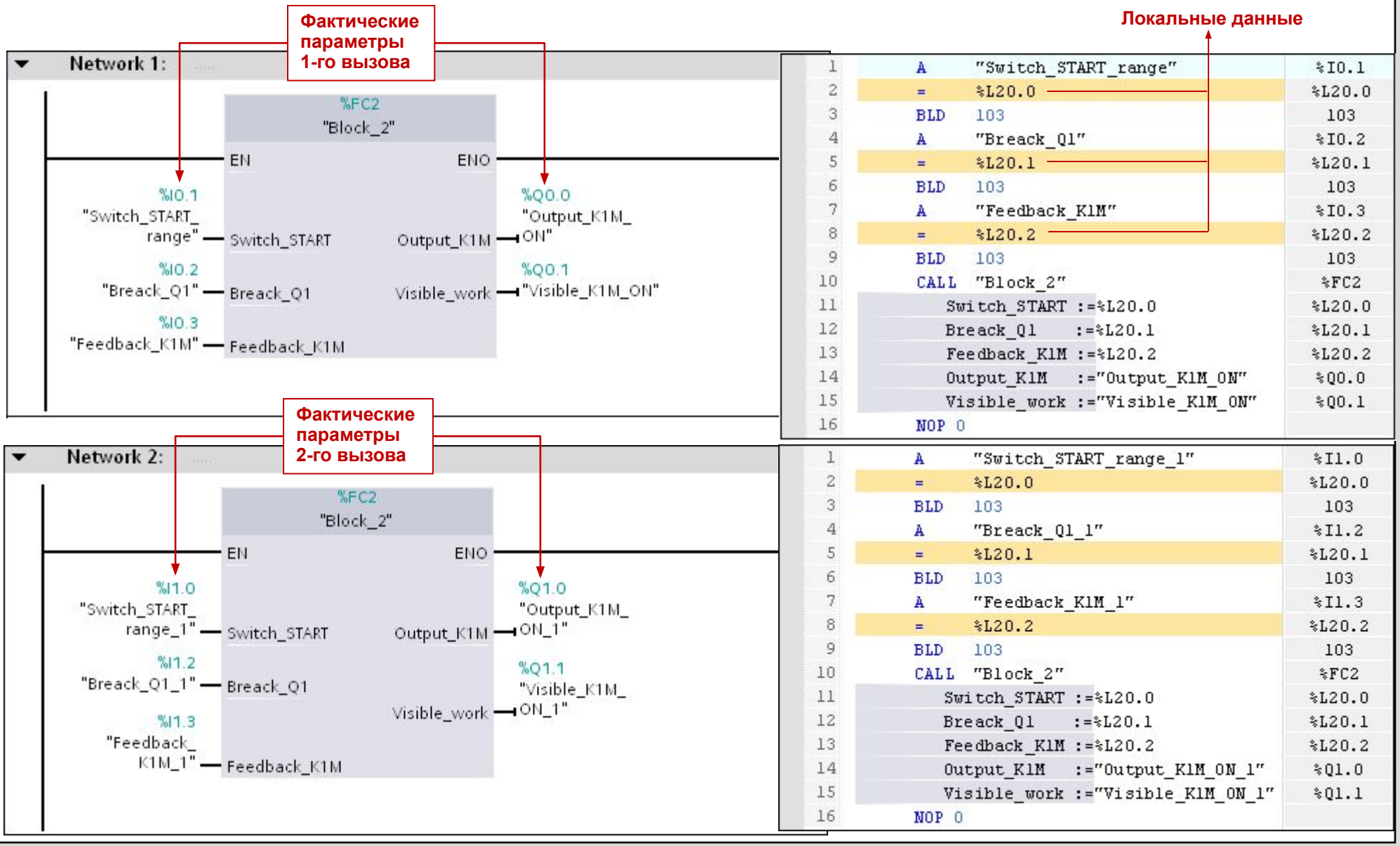
Вызов программного блока с параметрами (FC2)



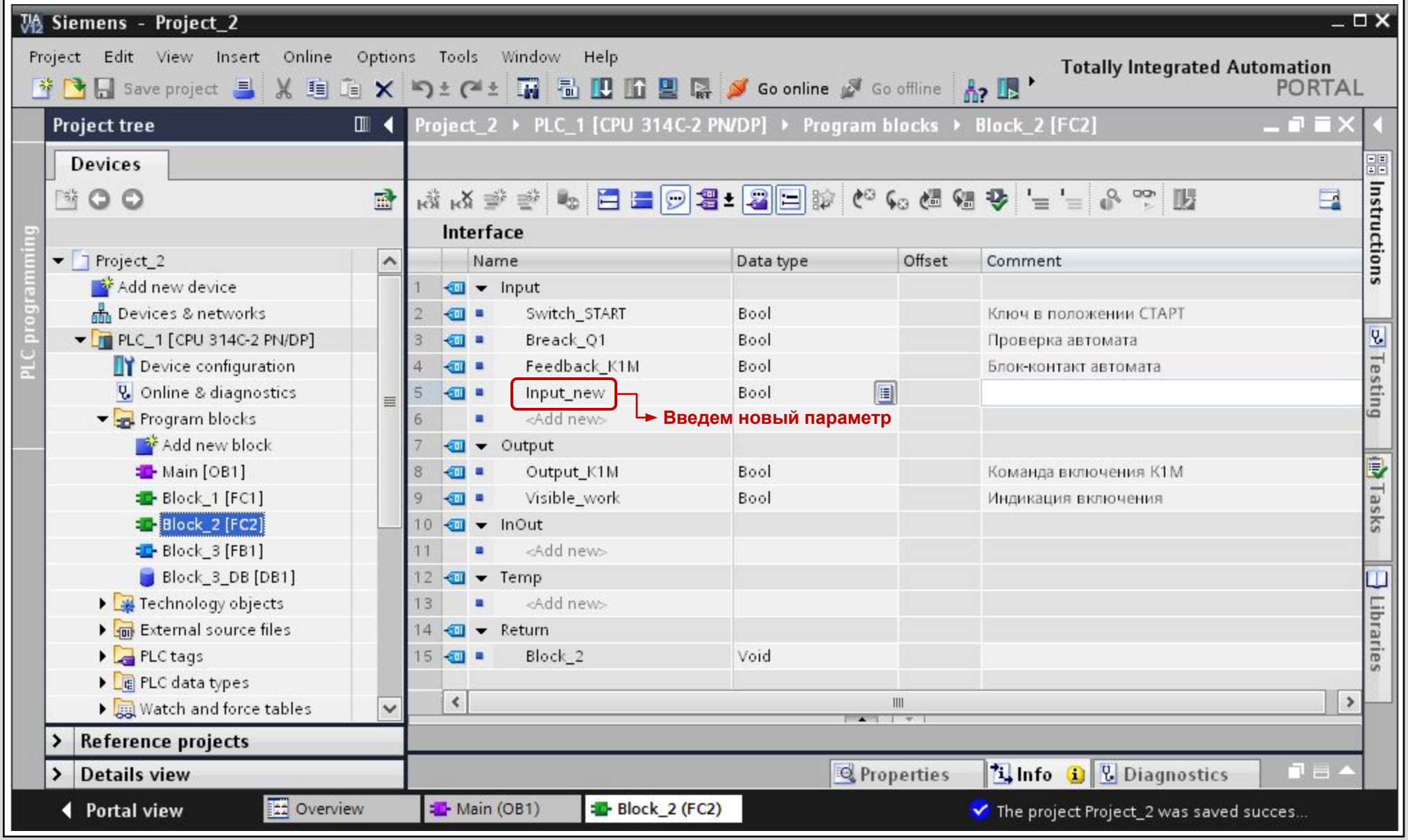
Вызов программного блока с параметрами (FC2)



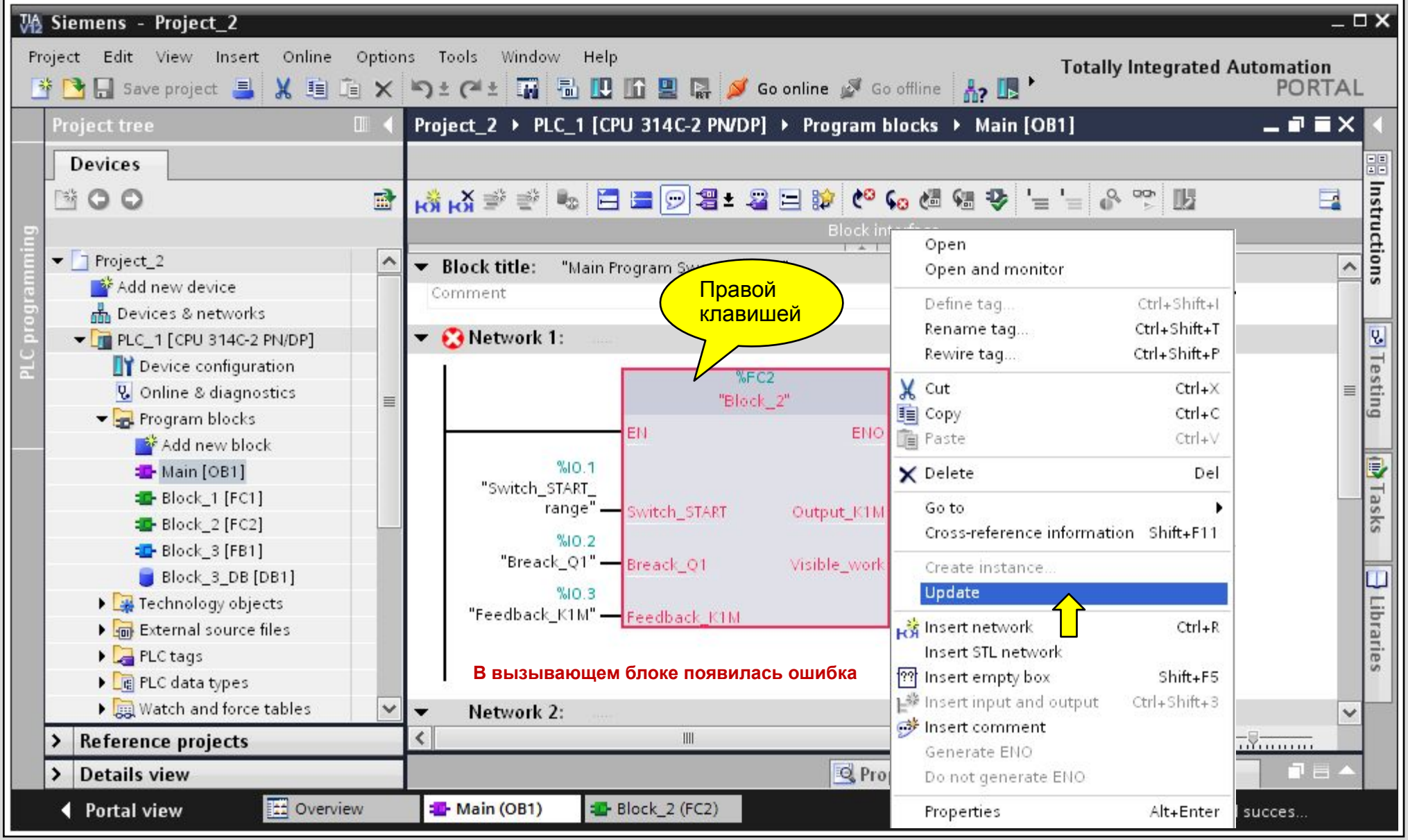
Вызов программного блока с параметрами (FC2)



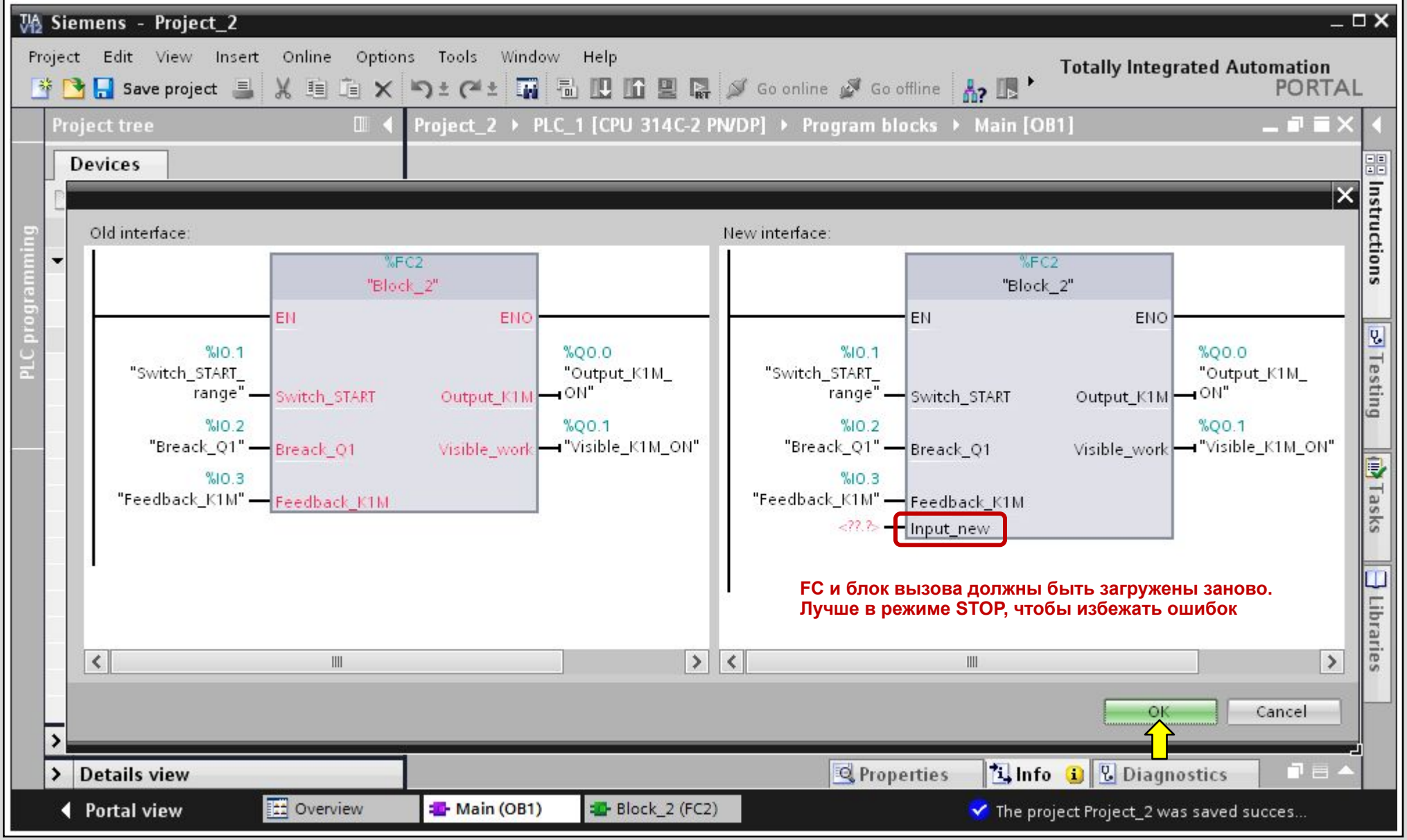
Обновление интерфейса FC



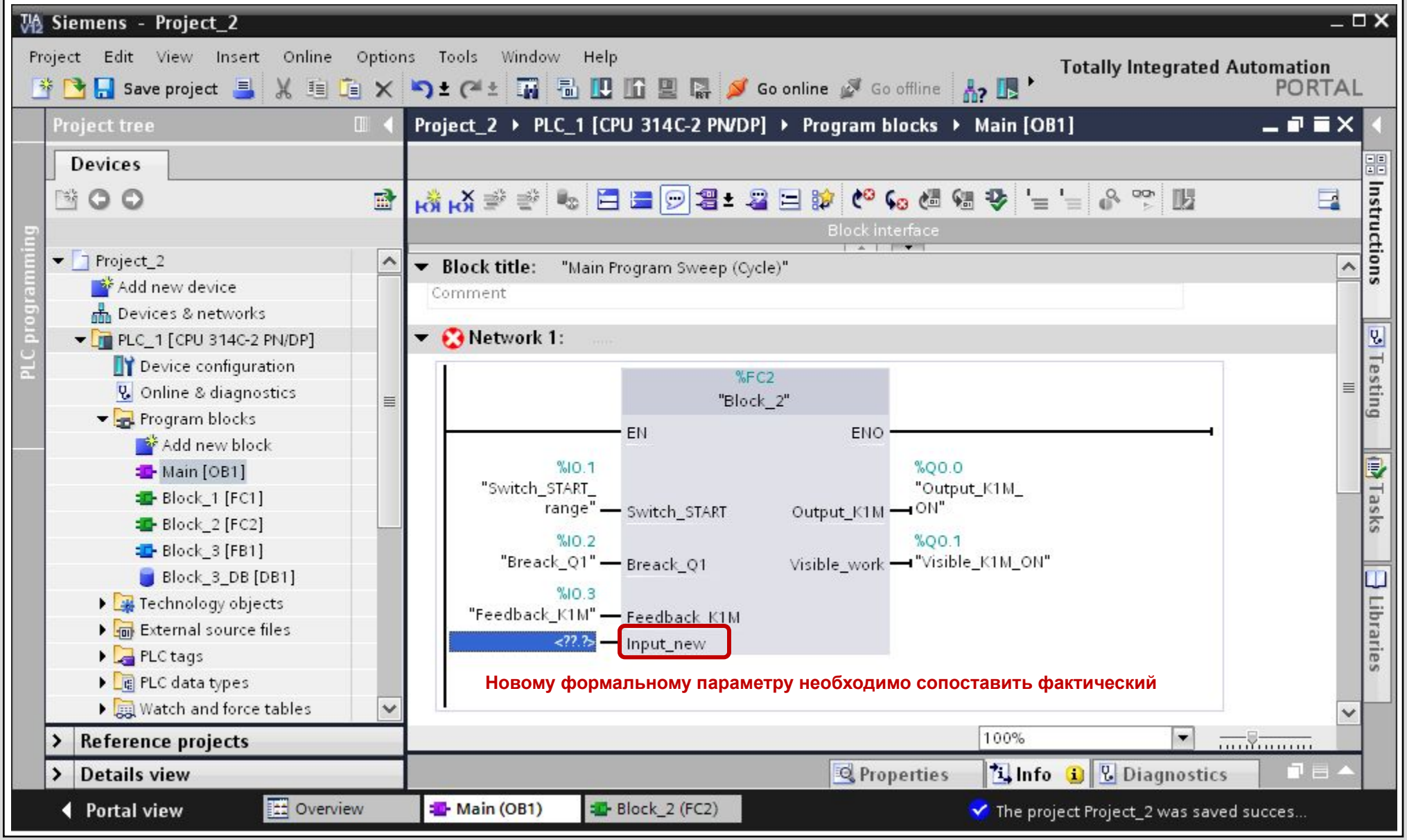
Обновление интерфейса FC в точке вызова



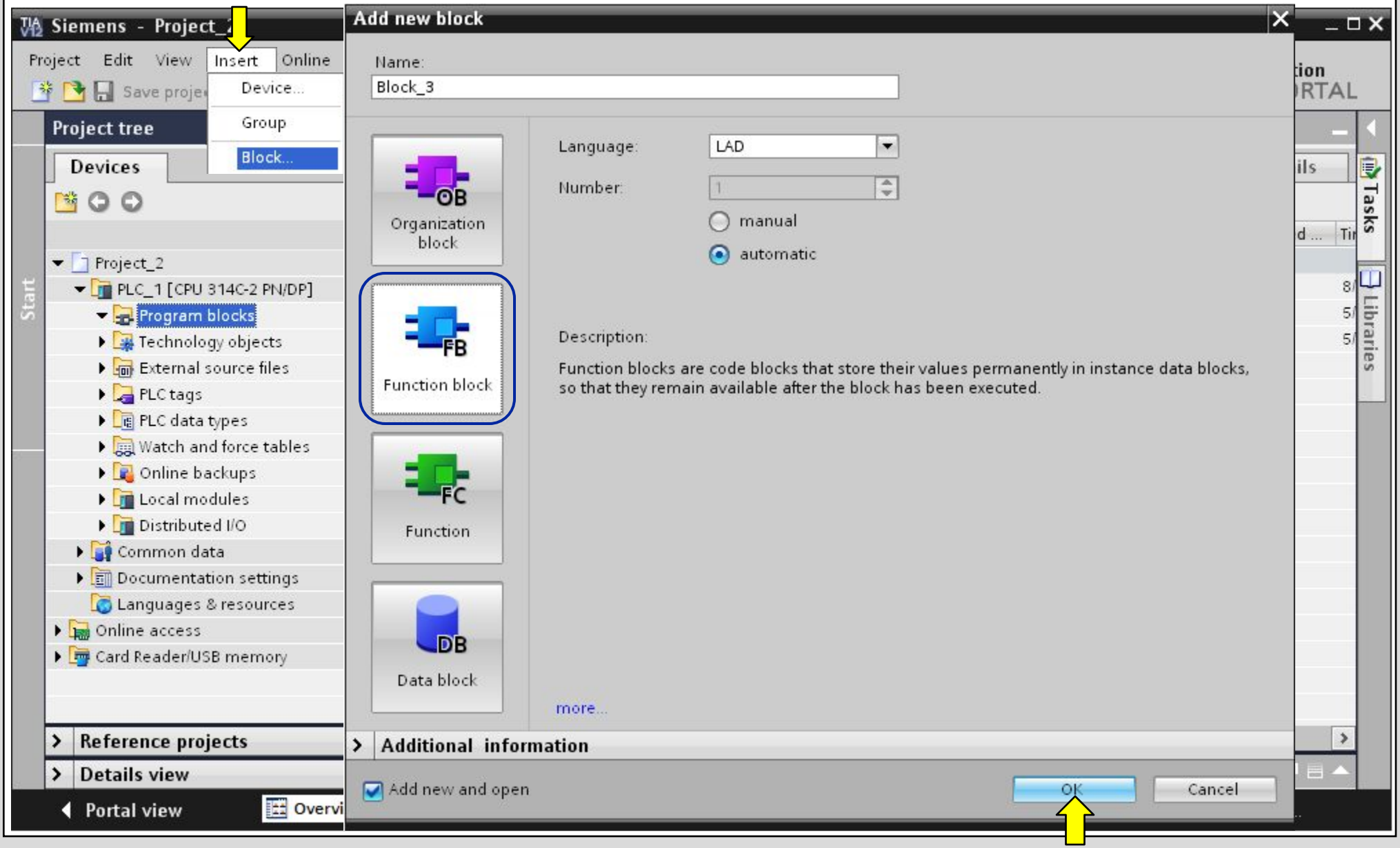
Обновление интерфейса FC в точке вызова



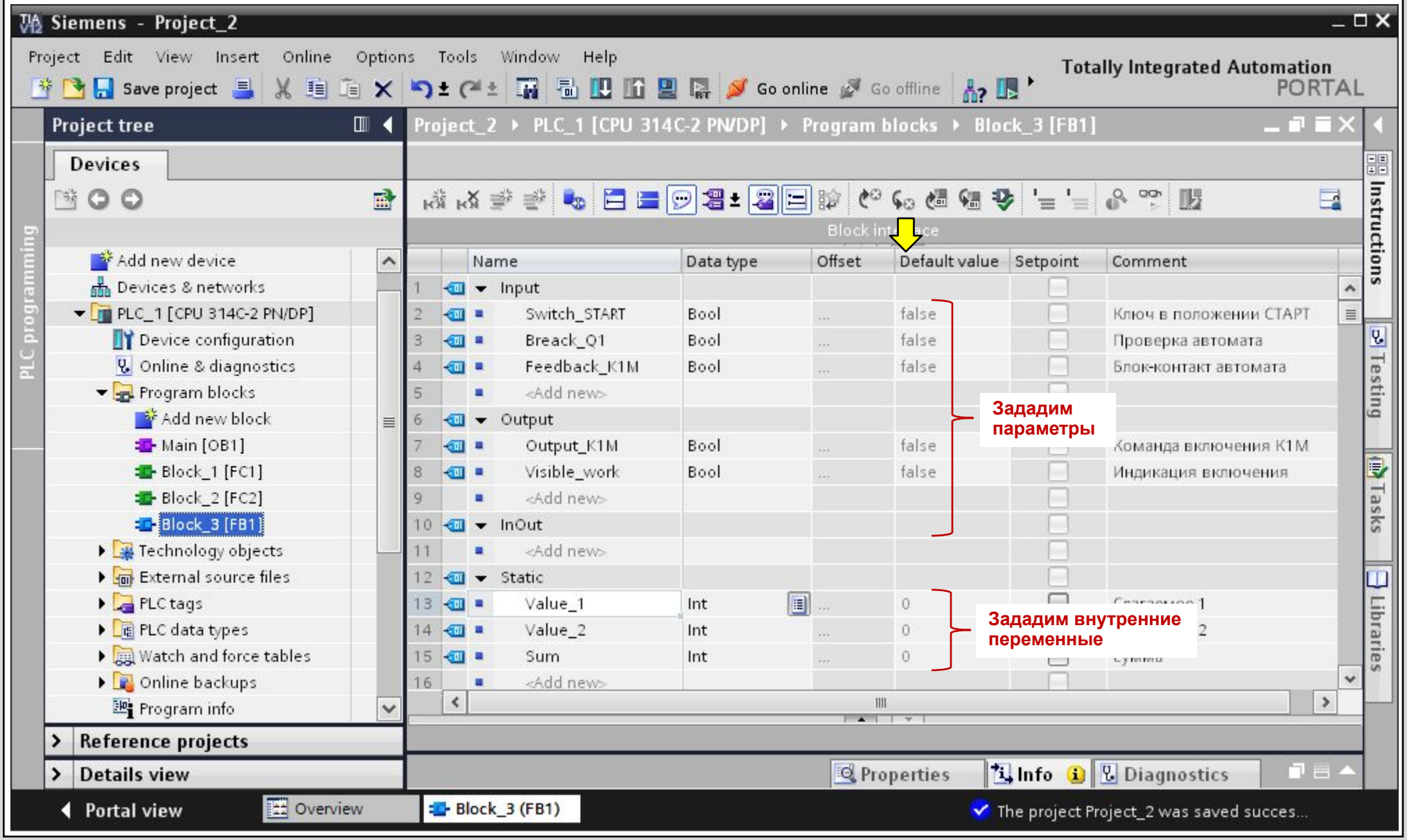
Обновление интерфейса FC в точке вызова



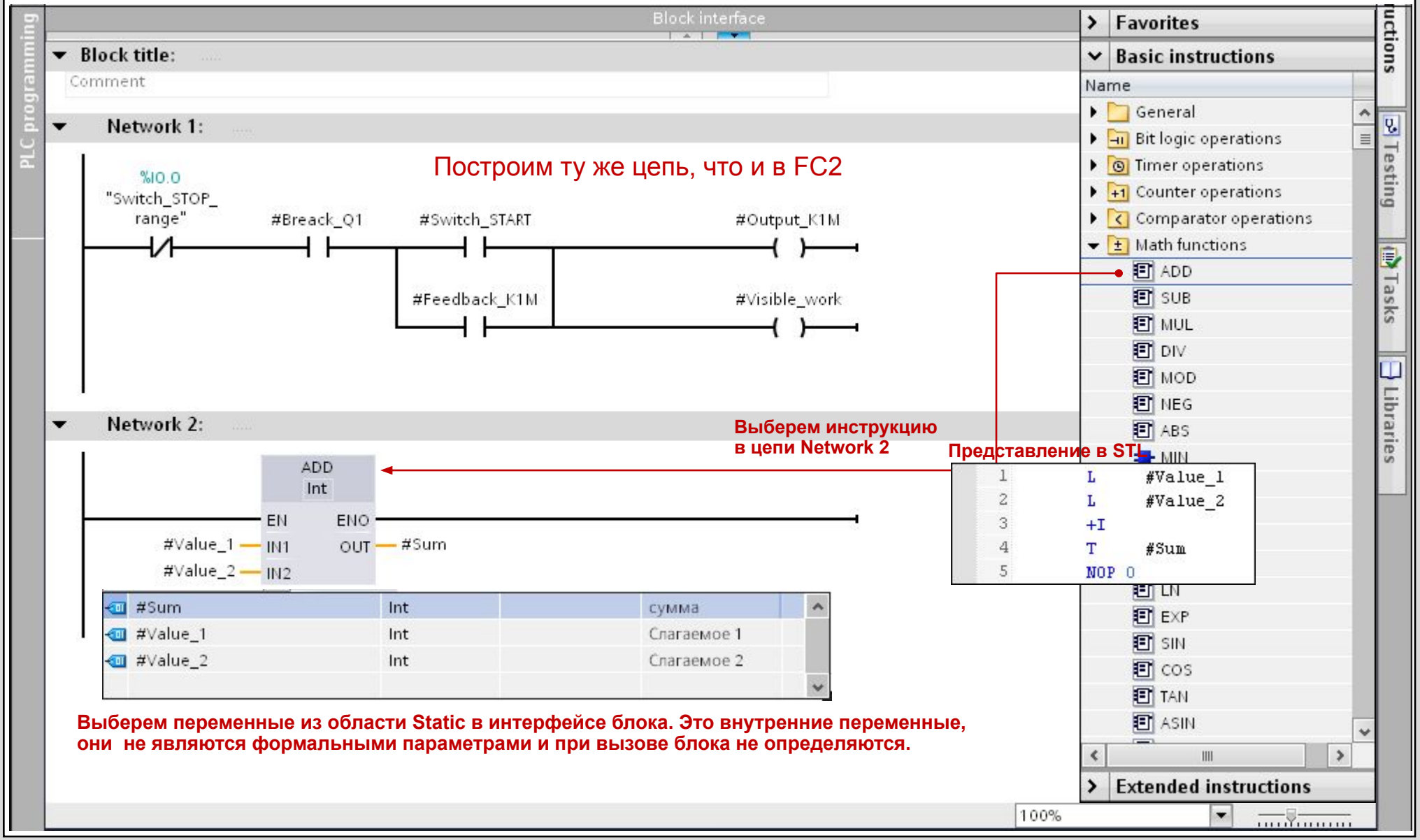
Создание программного блока FB



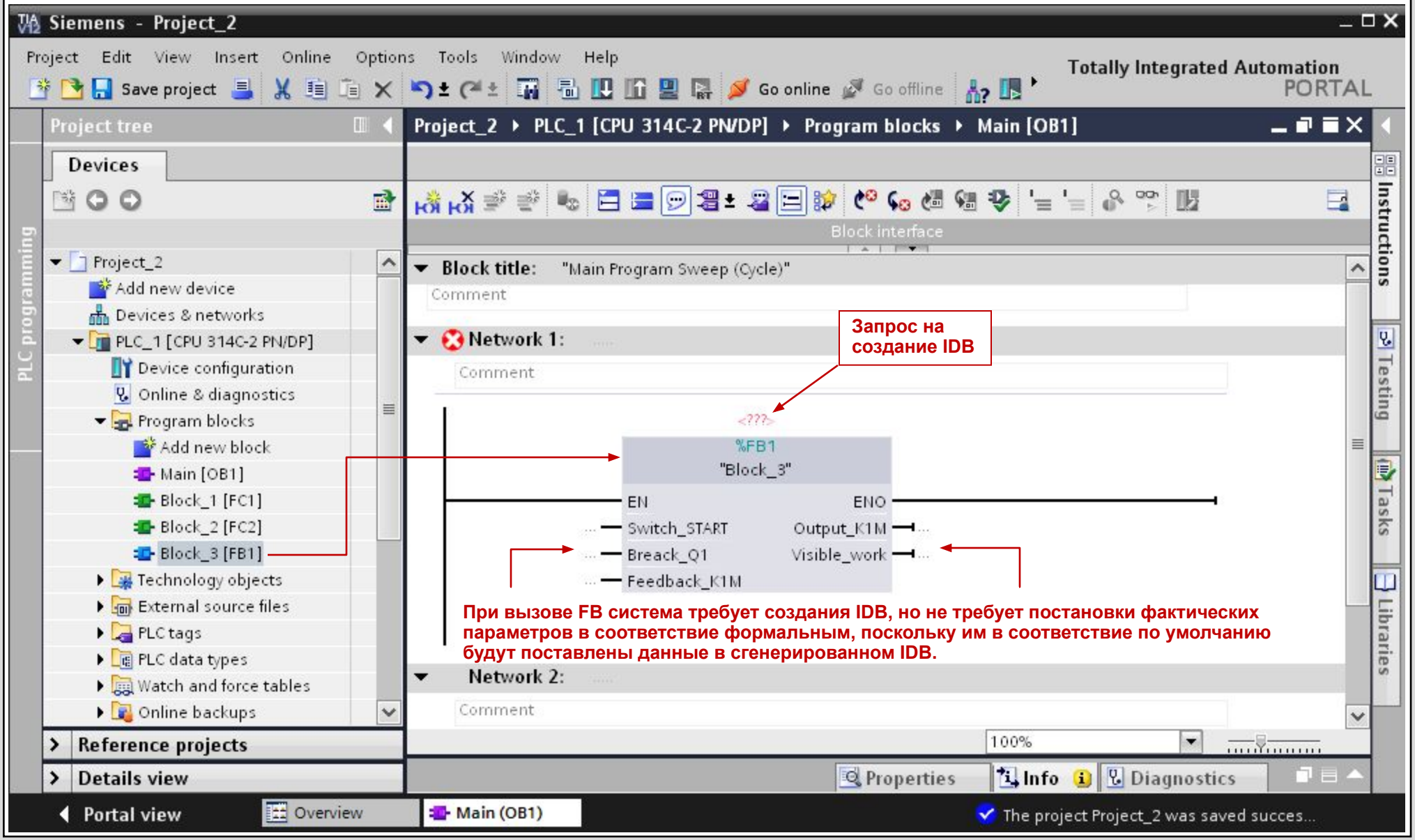
Создание программного блока FB



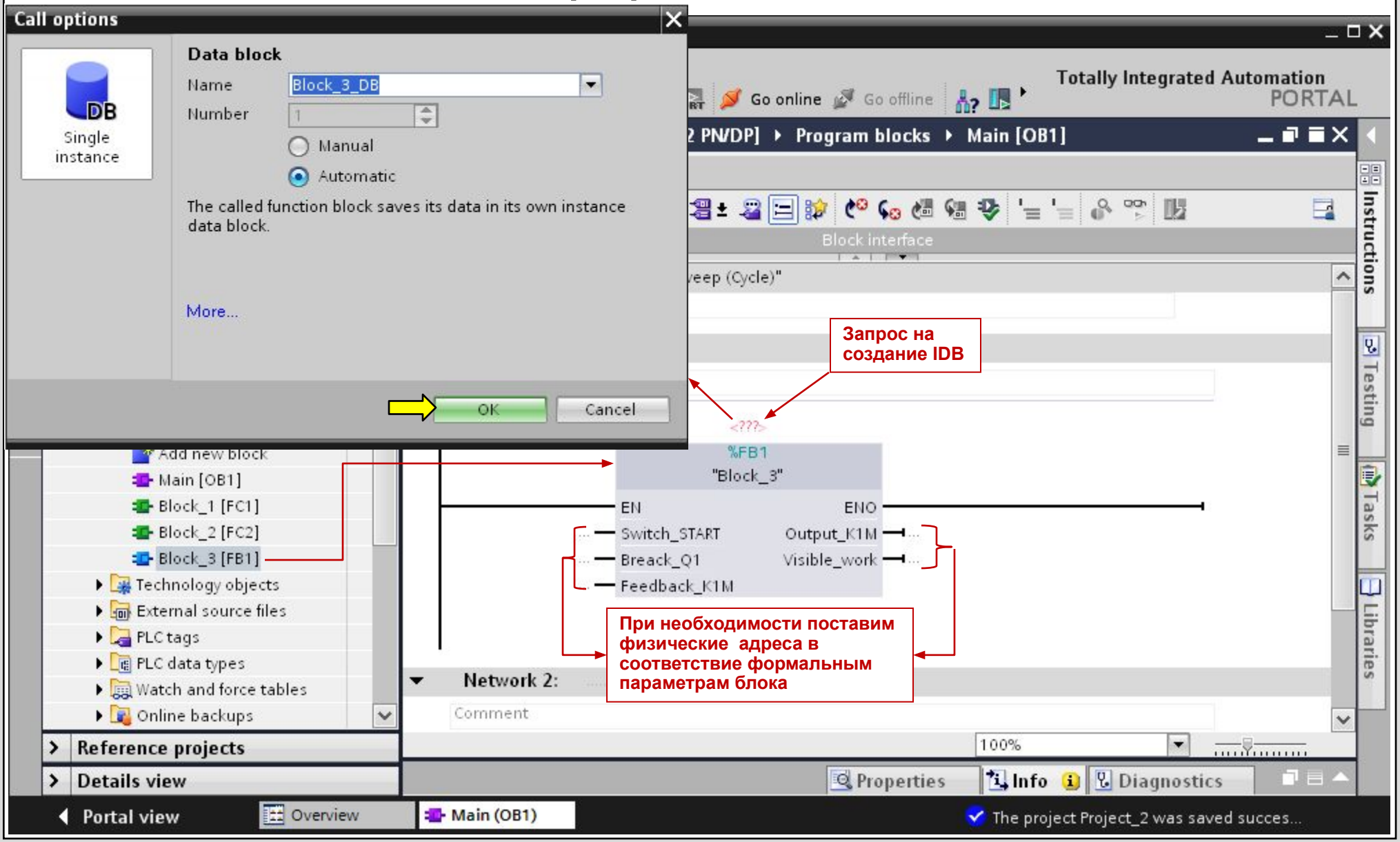
Создание программного блока FB



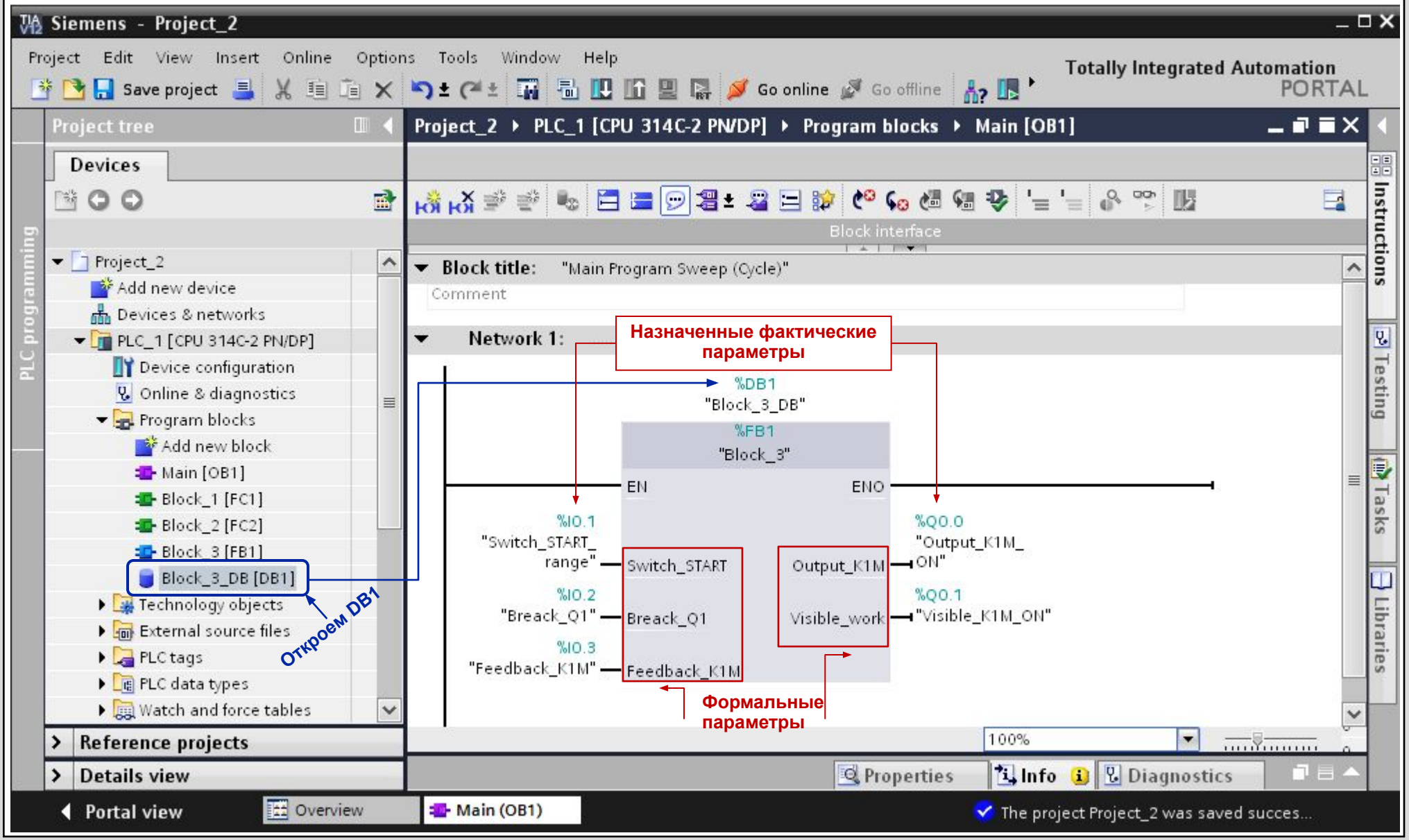
Вызов программного блока FB



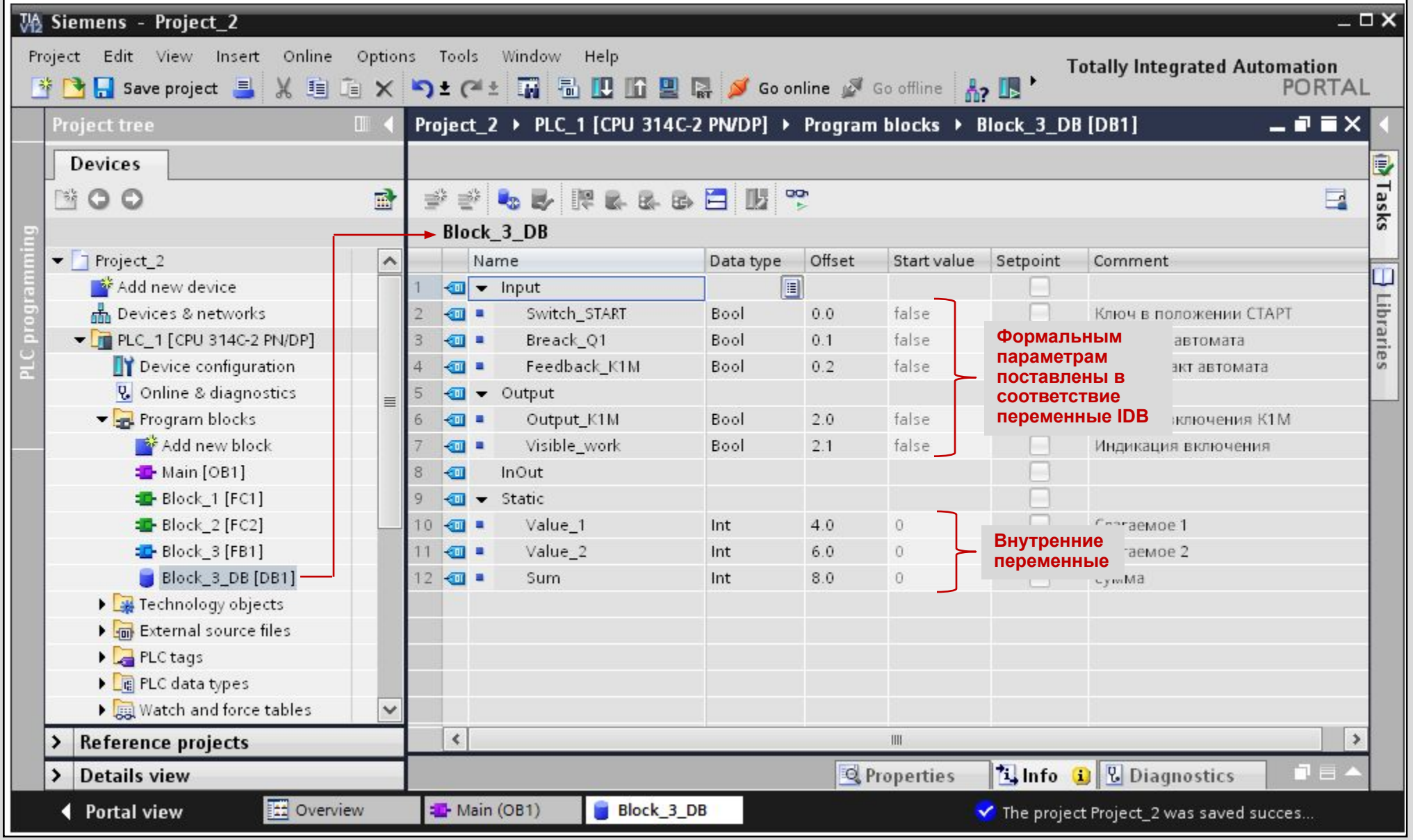
Вызов программного блока FB



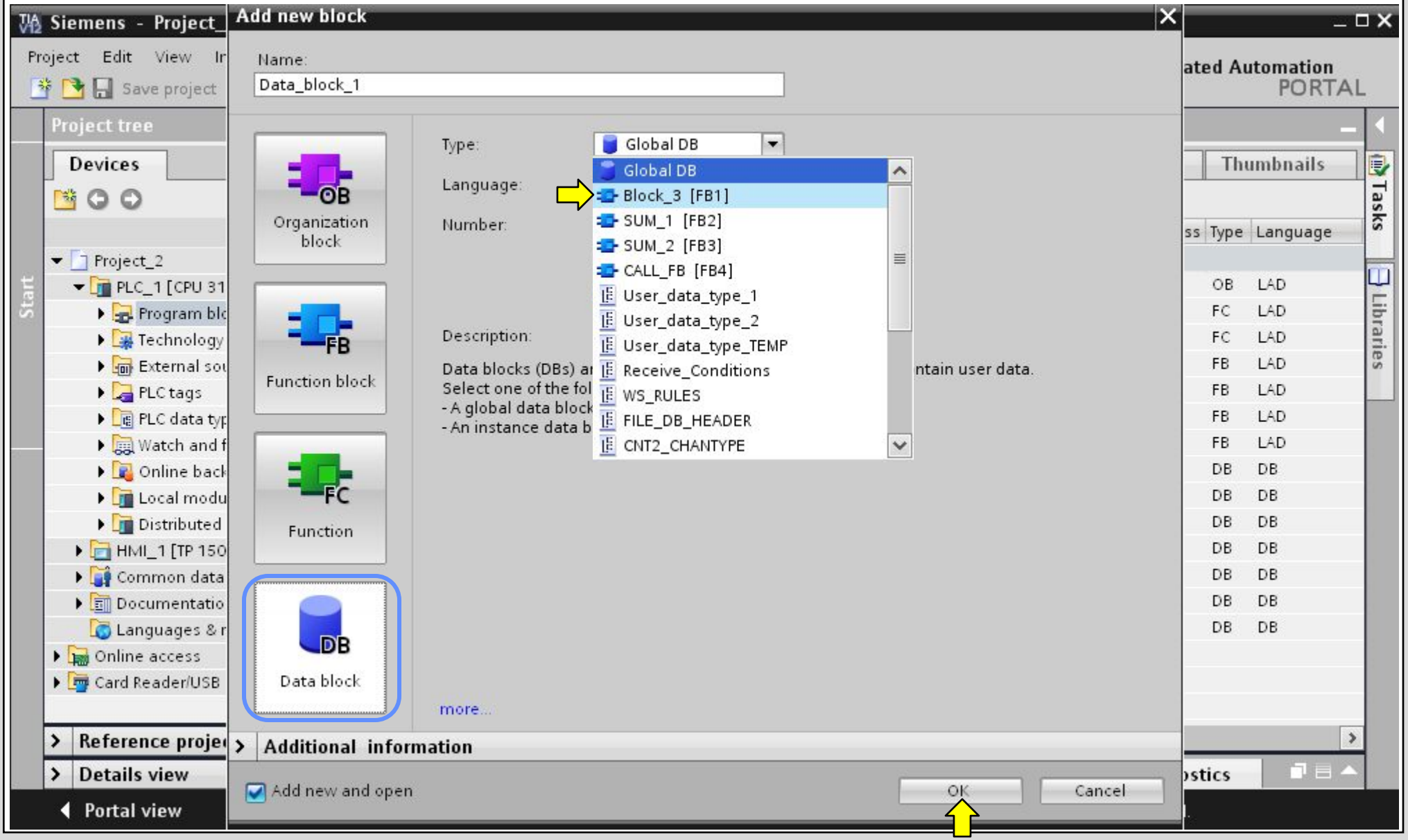
Вызов программного блока FB, создание IDB



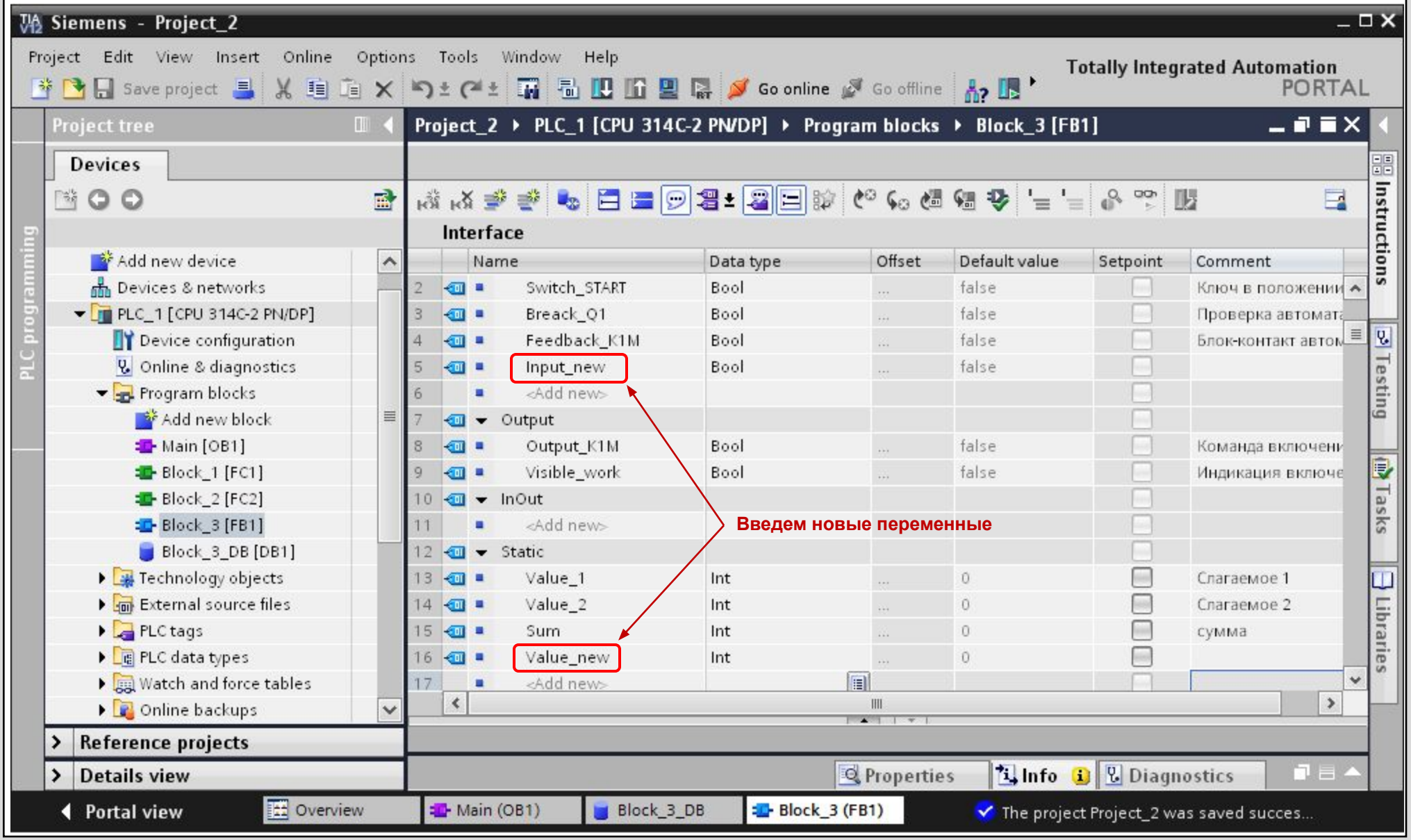
Блок данных IDB, работающий совместно с FB



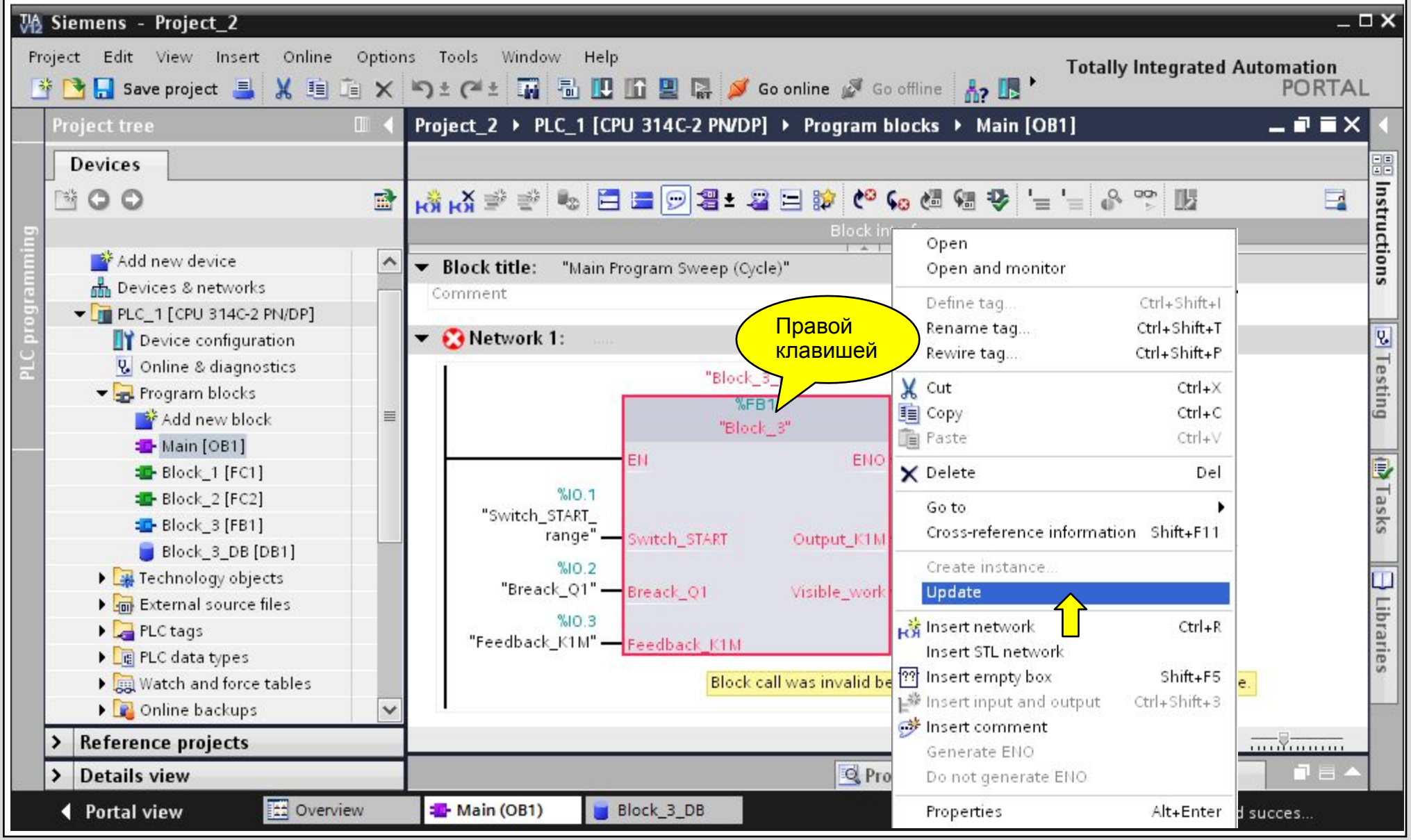
IDB может быть также создан из меню создания блоков



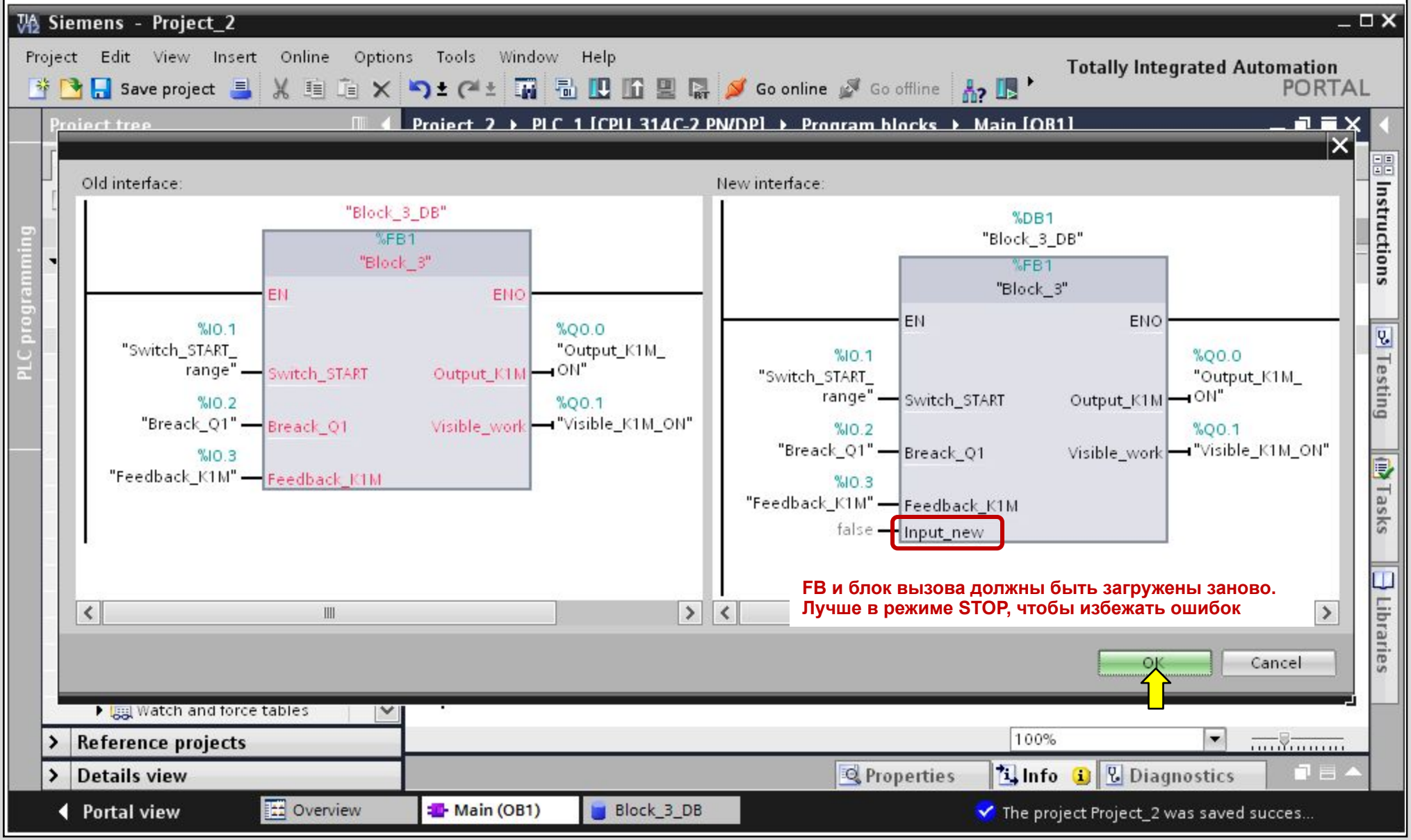
Обновление интерфейса FB



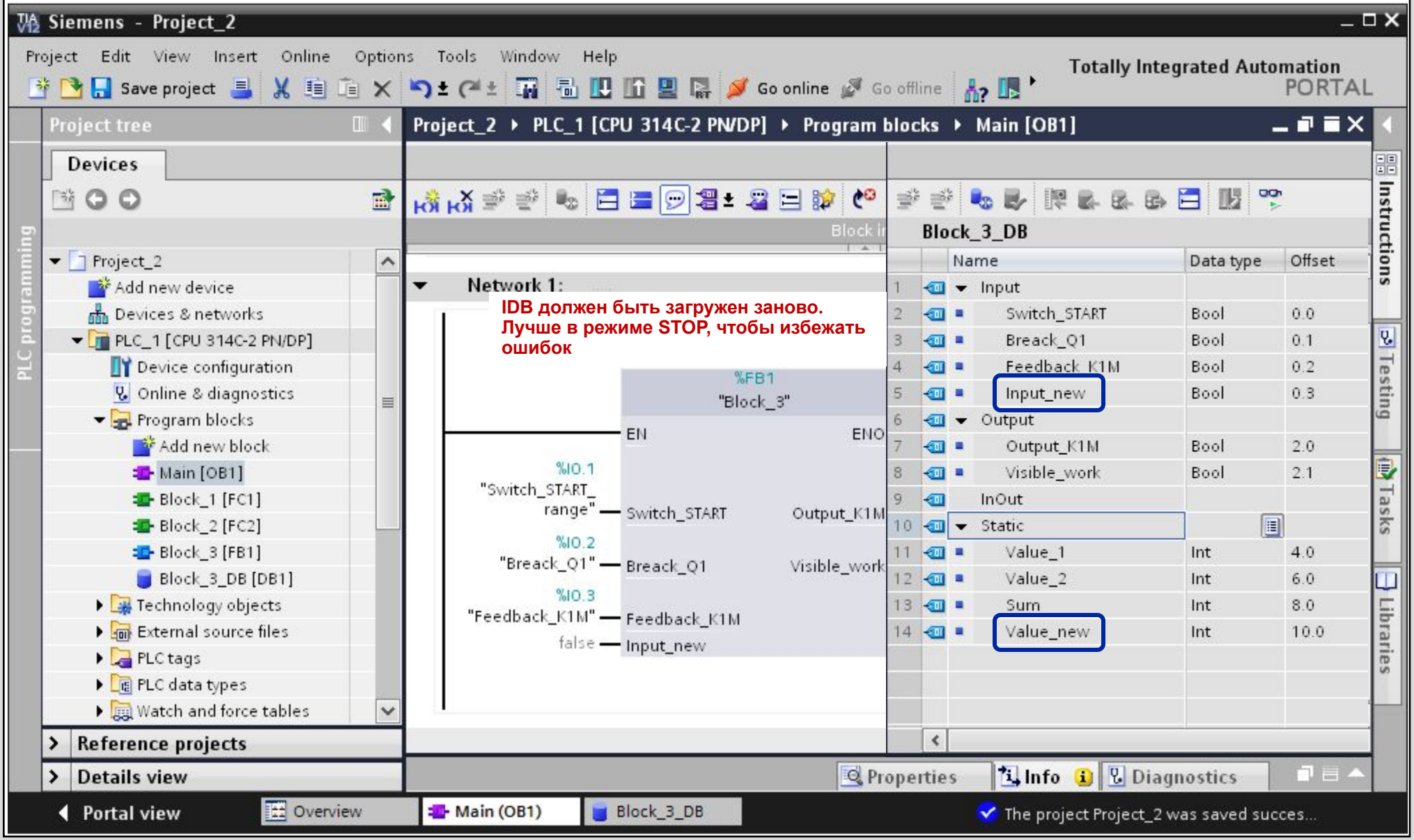
Обновление интерфейса FB в точке вызова



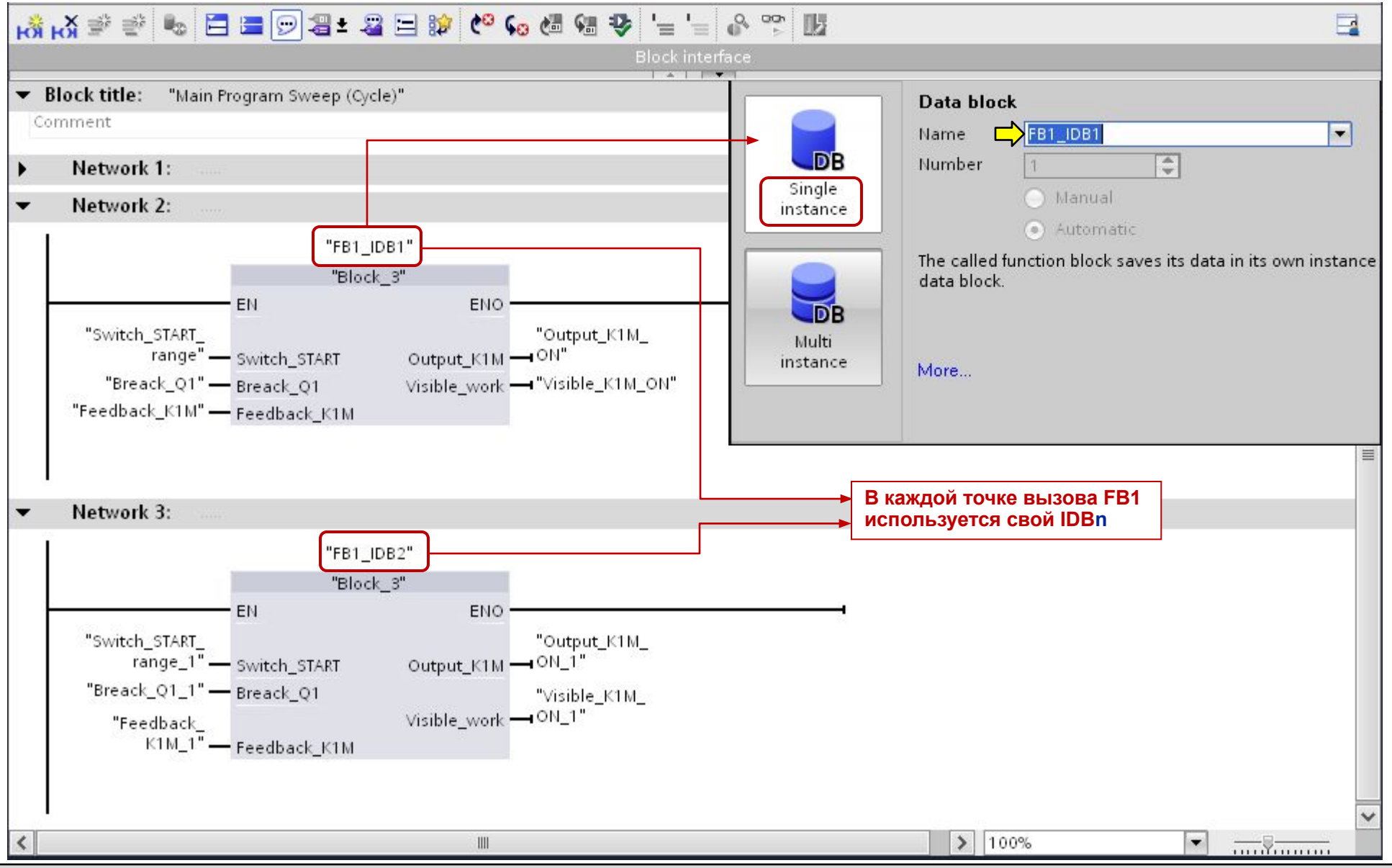
Обновление интерфейса FB в точке вызова



Обновление IDB, работающего с FB



Использование FB с экземплярным IDB в каждом вызове



Мультиэкземплярная модель вызова FB

Создадим блоки FB2 и FB3, определим в них внутренние переменные для операции, например, сложения двух чисел

PLC_1 [CPU 314C-2 PN/DP] ▸ Program blocks ▸ SUM_1 [FB2]

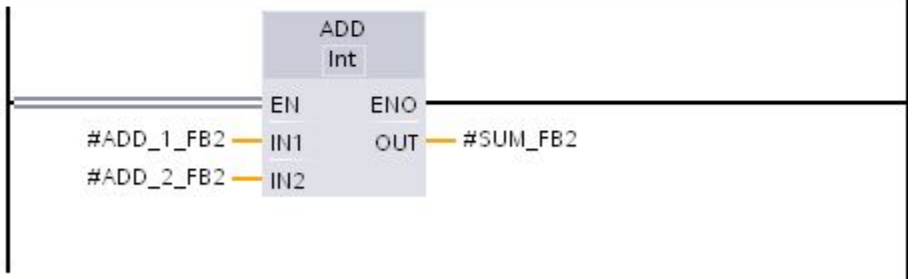
Interface

		Name	Data type	Offset	Default value	Comment
1		Input				
2		Output				
3		InOut				
4		Static				
5		ADD_1_FB2	Int	0.0	0	Слагаемое_1_FB2
6		ADD_2_FB2	Int	2.0	0	Слагаемое_2_FB2
7		SUM_FB2	Int	4.0	0	Сумма_FB2
8		Temp				

Block title:

Comment

Network 1:



#ADD_1_FB2		Слагаемое_1_FB2
#ADD_2_FB2		Слагаемое_2_FB2
#SUM_FB2		Сумма_FB2

PLC_1 [CPU 314C-2 PN/DP] ▸ Program blocks ▸ SUM_2 [FB3]

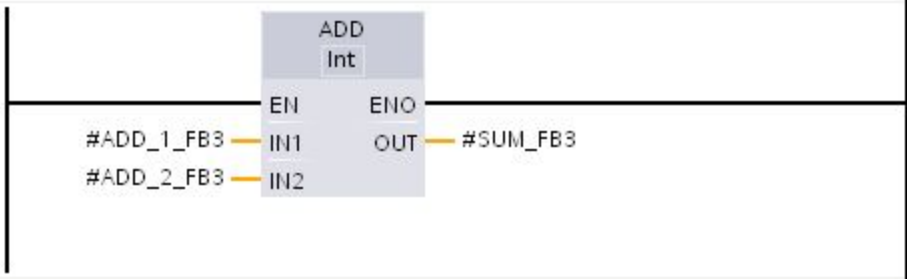
Interface

		Name	Data type	Offset	Default value	Comment
1		Input				
2		Output				
3		InOut				
4		Static				
5		ADD_1_FB3	Int	0.0	0	Слагаемое_1_FB3
6		ADD_2_FB3	Int	2.0	0	Слагаемое_2_FB3
7		SUM_FB3	Int	4.0	0	Сумма_FB3
8		Temp				

Block title:

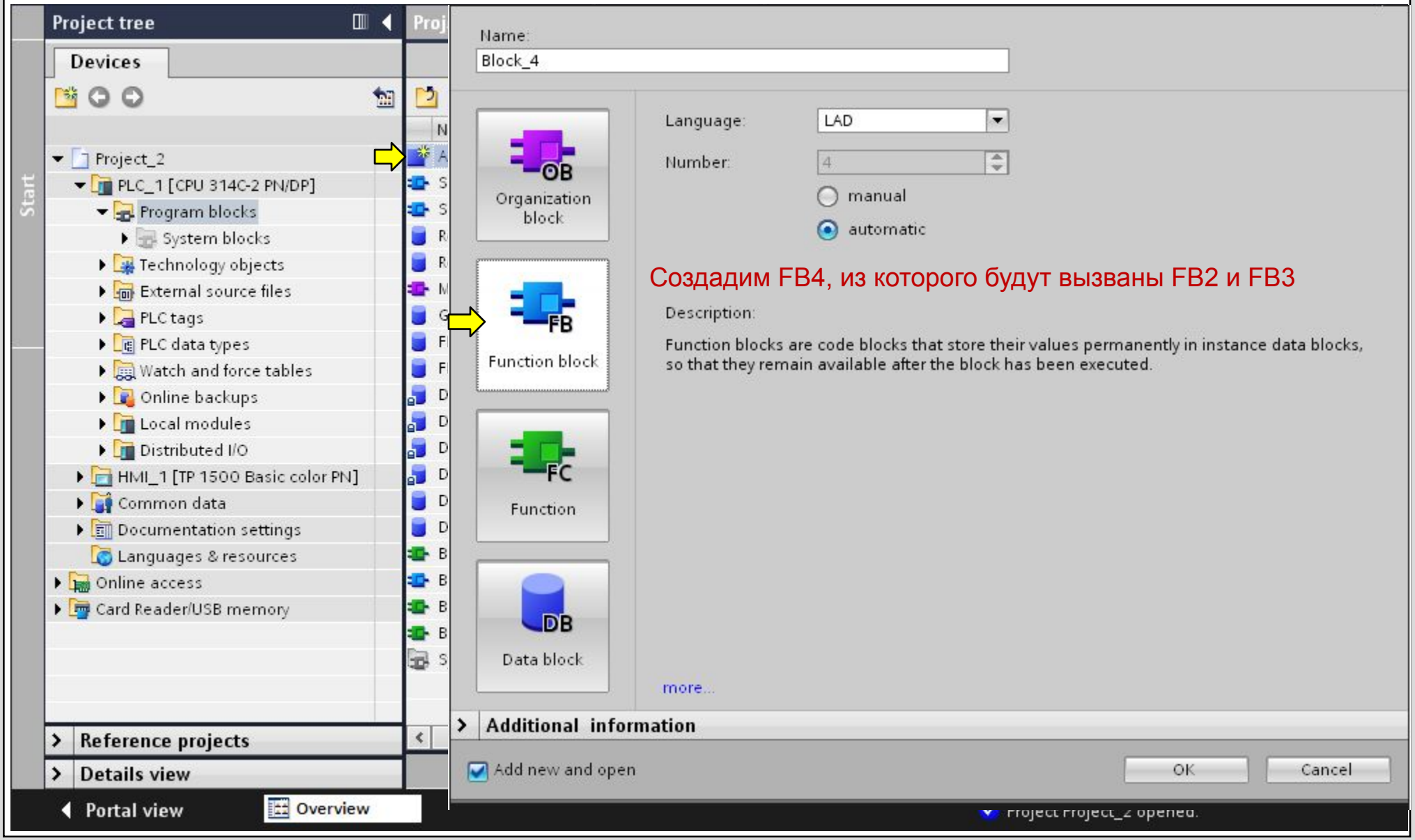
Comment

Network 1:

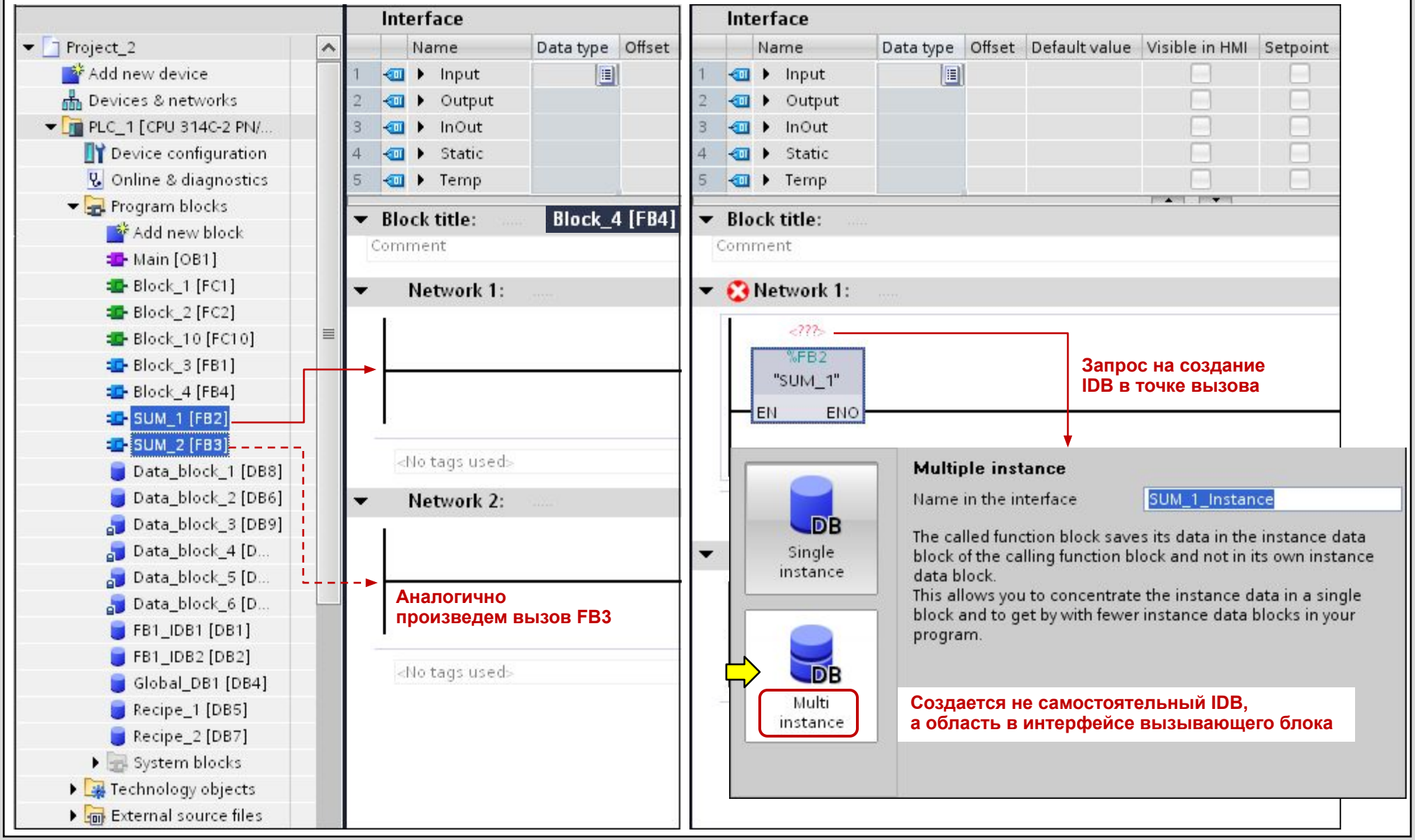


#ADD_1_FB3		Слагаемое_1_FB3
#ADD_2_FB3		Слагаемое_2_FB3
#SUM_FB3		Сумма_FB3

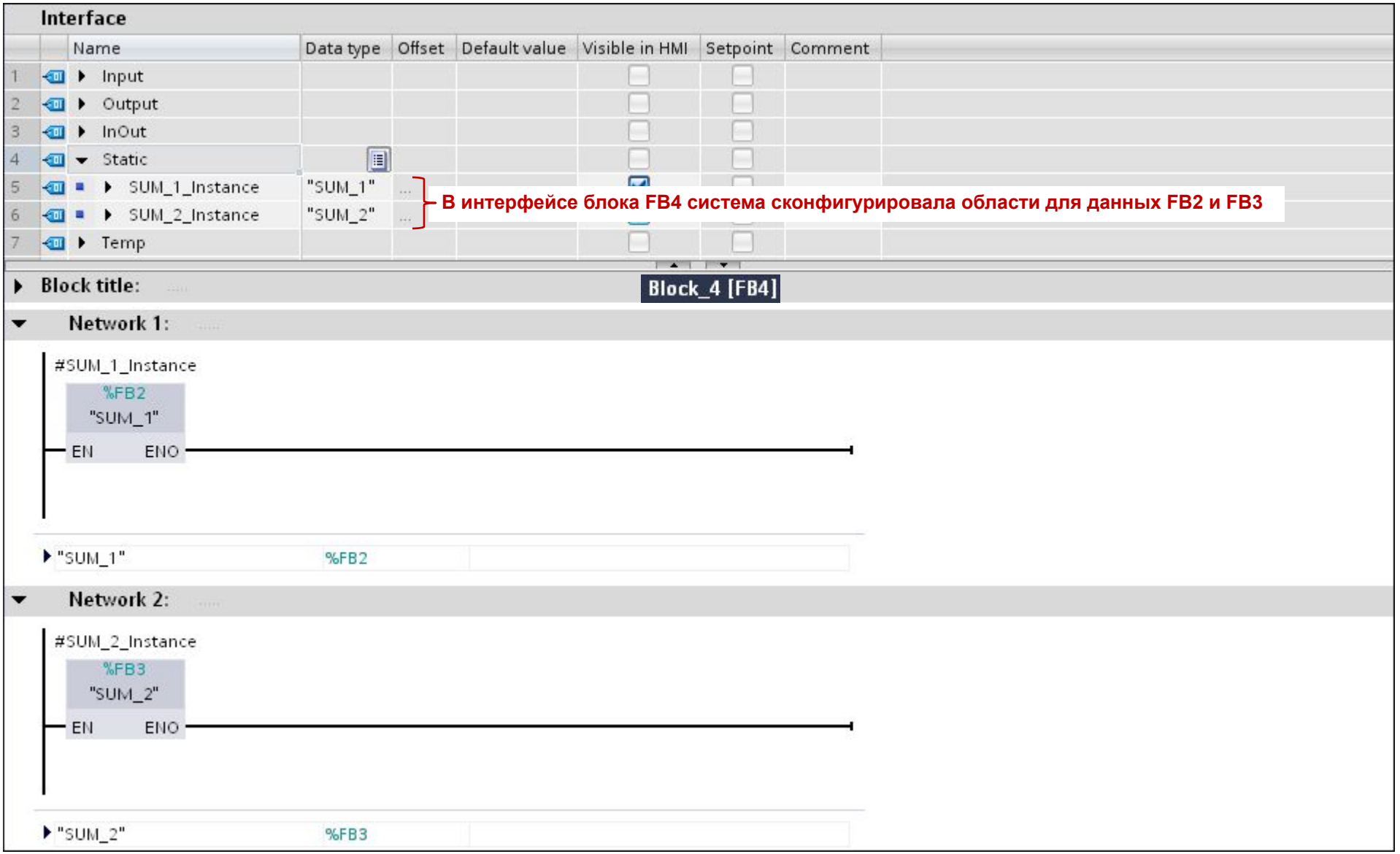
Мультиэкземплярная модель вызова FB



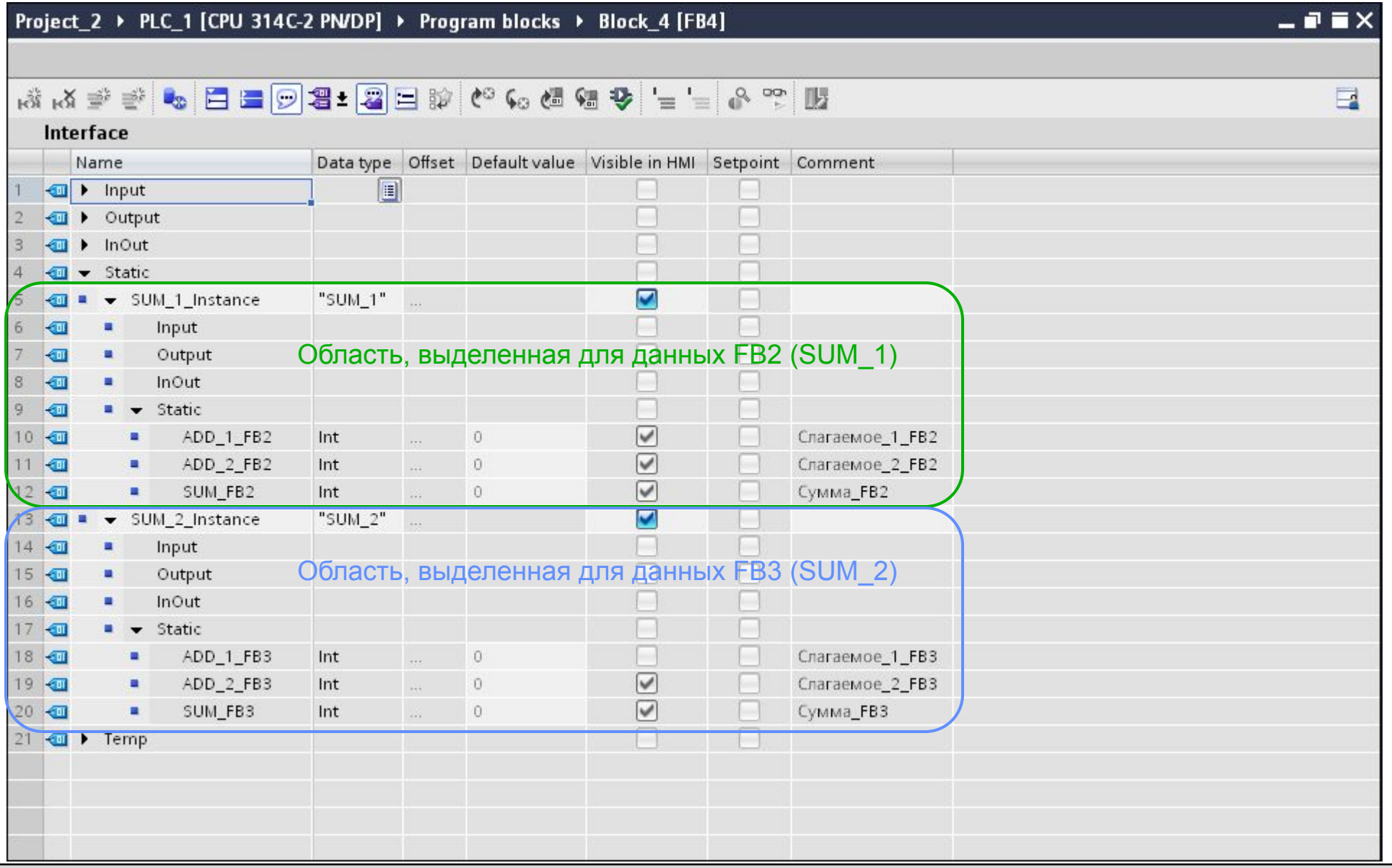
Мультиэкземплярная модель вызова FB



Мультиэкземплярная модель вызова FB

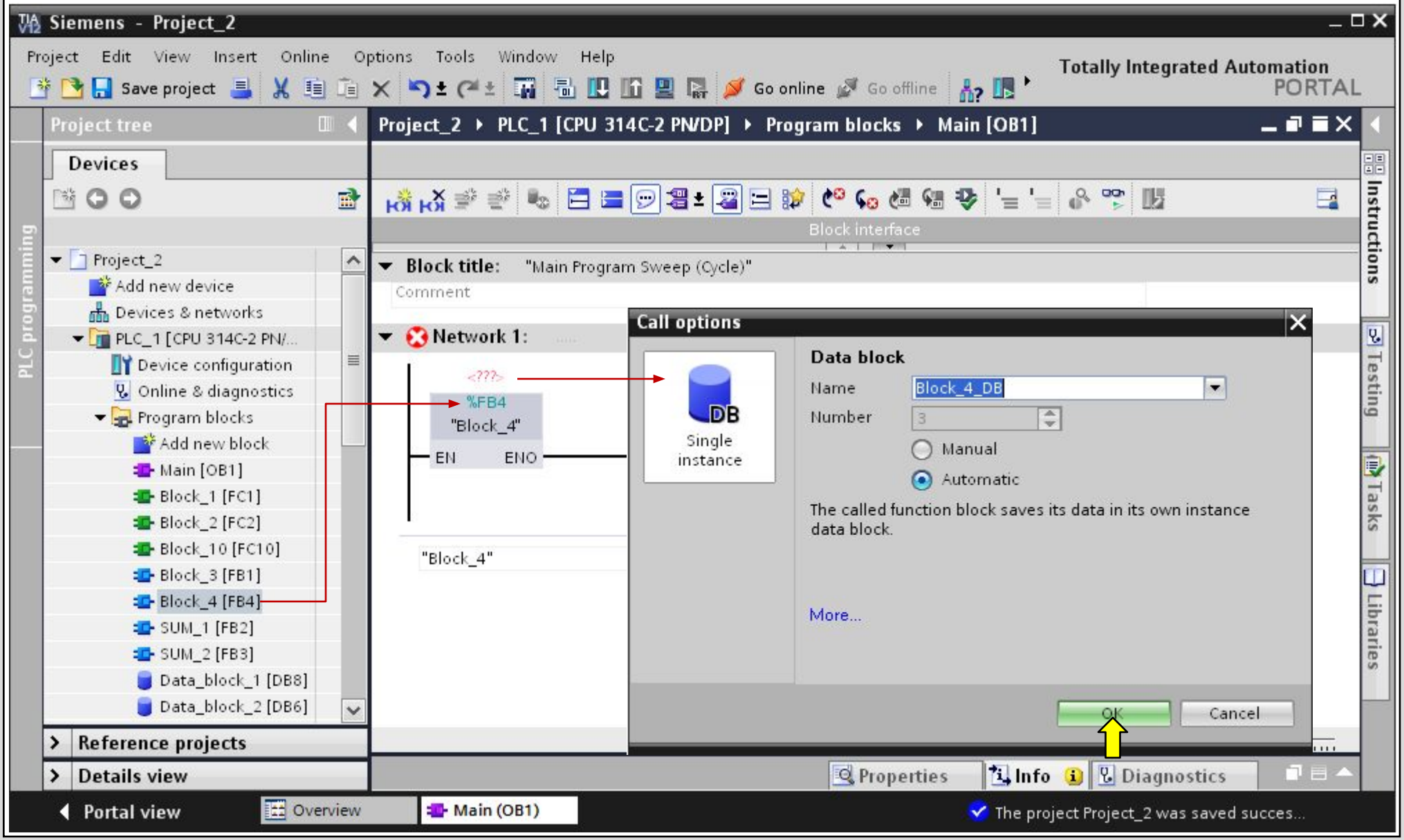


Интерфейс вызывающего блока FB4

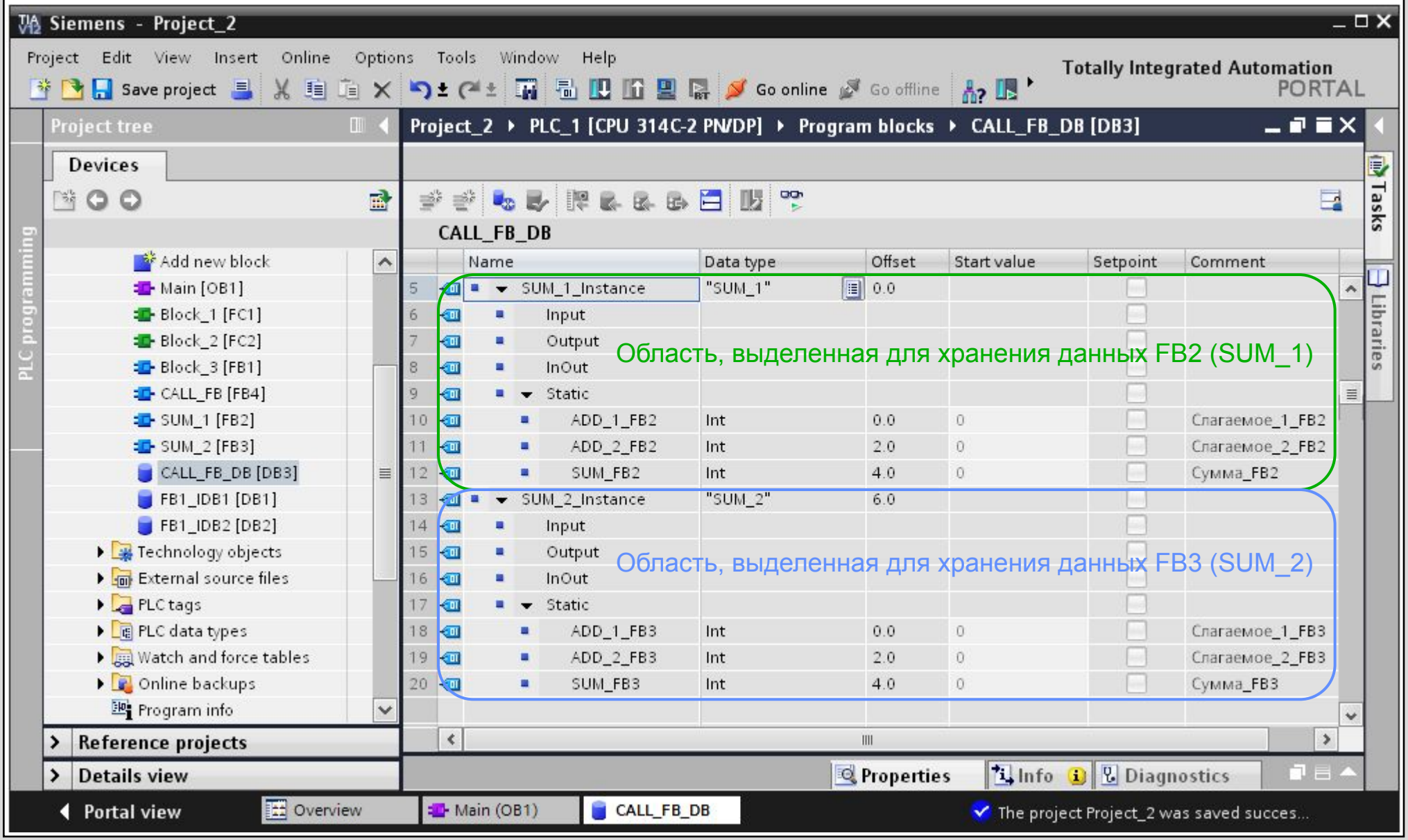


	Name	Data type	Offset	Default value	Visible in HMI	Setpoint	Comment
1	Input				☐	☐	
2	Output				☐	☐	
3	InOut				☐	☐	
4	Static				☐	☐	
5	SUM_1_Instance	"SUM_1"	...		☑	☐	
6	Input				☐	☐	
7	Output				☐	☐	
8	InOut				☐	☐	
9	Static				☐	☐	
10	ADD_1_FB2	Int	...	0	☑	☐	Слагаемое_1_FB2
11	ADD_2_FB2	Int	...	0	☑	☐	Слагаемое_2_FB2
12	SUM_FB2	Int	...	0	☑	☐	Сумма_FB2
13	SUM_2_Instance	"SUM_2"	...		☑	☐	
14	Input				☐	☐	
15	Output				☐	☐	
16	InOut				☐	☐	
17	Static				☐	☐	
18	ADD_1_FB3	Int	...	0	☐	☐	Слагаемое_1_FB3
19	ADD_2_FB3	Int	...	0	☑	☐	Слагаемое_2_FB3
20	SUM_FB3	Int	...	0	☑	☐	Сумма_FB3
21	Temp				☐	☐	

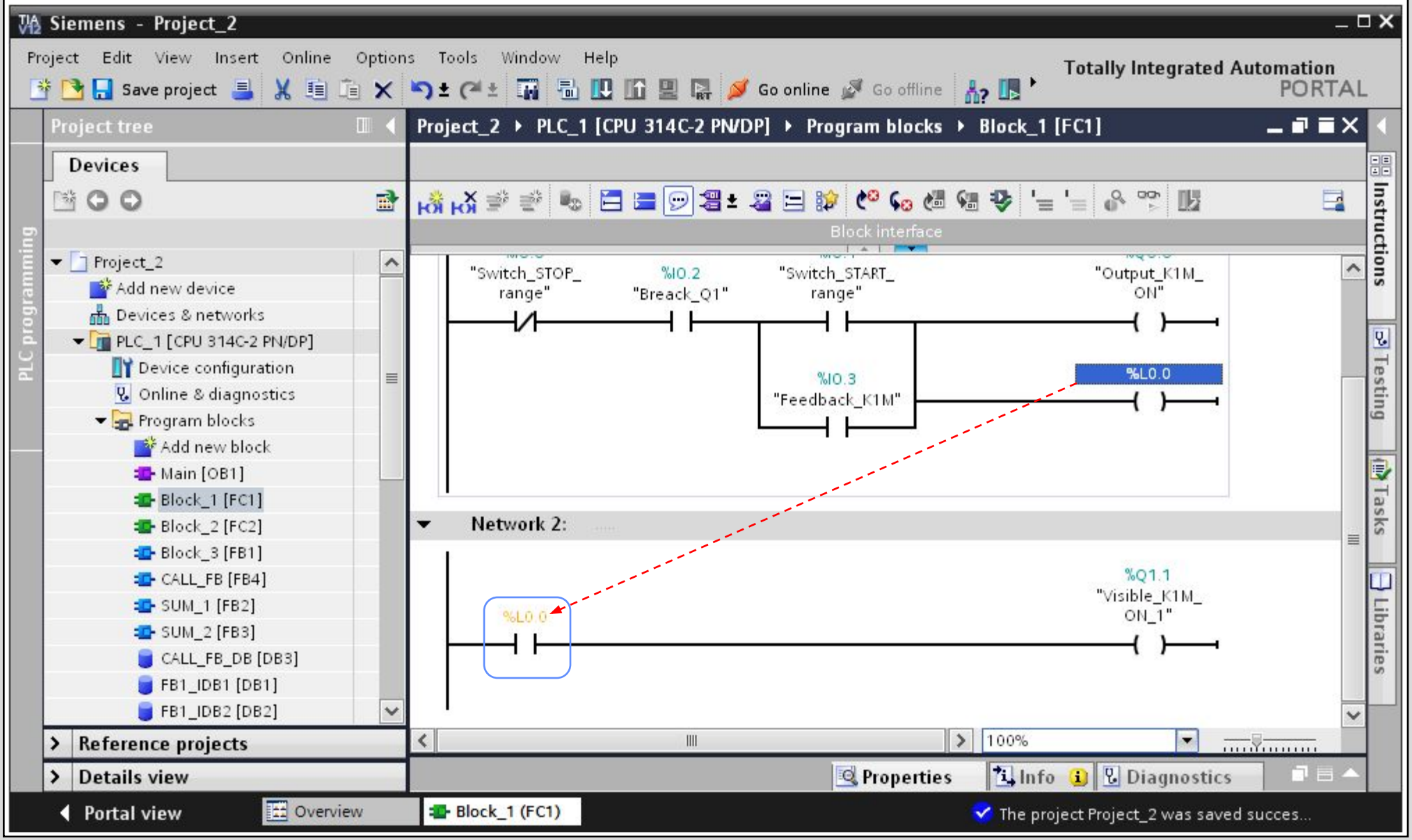
Вызов блока FB4 из OB1 с созданием мультиэкземлярного IDB



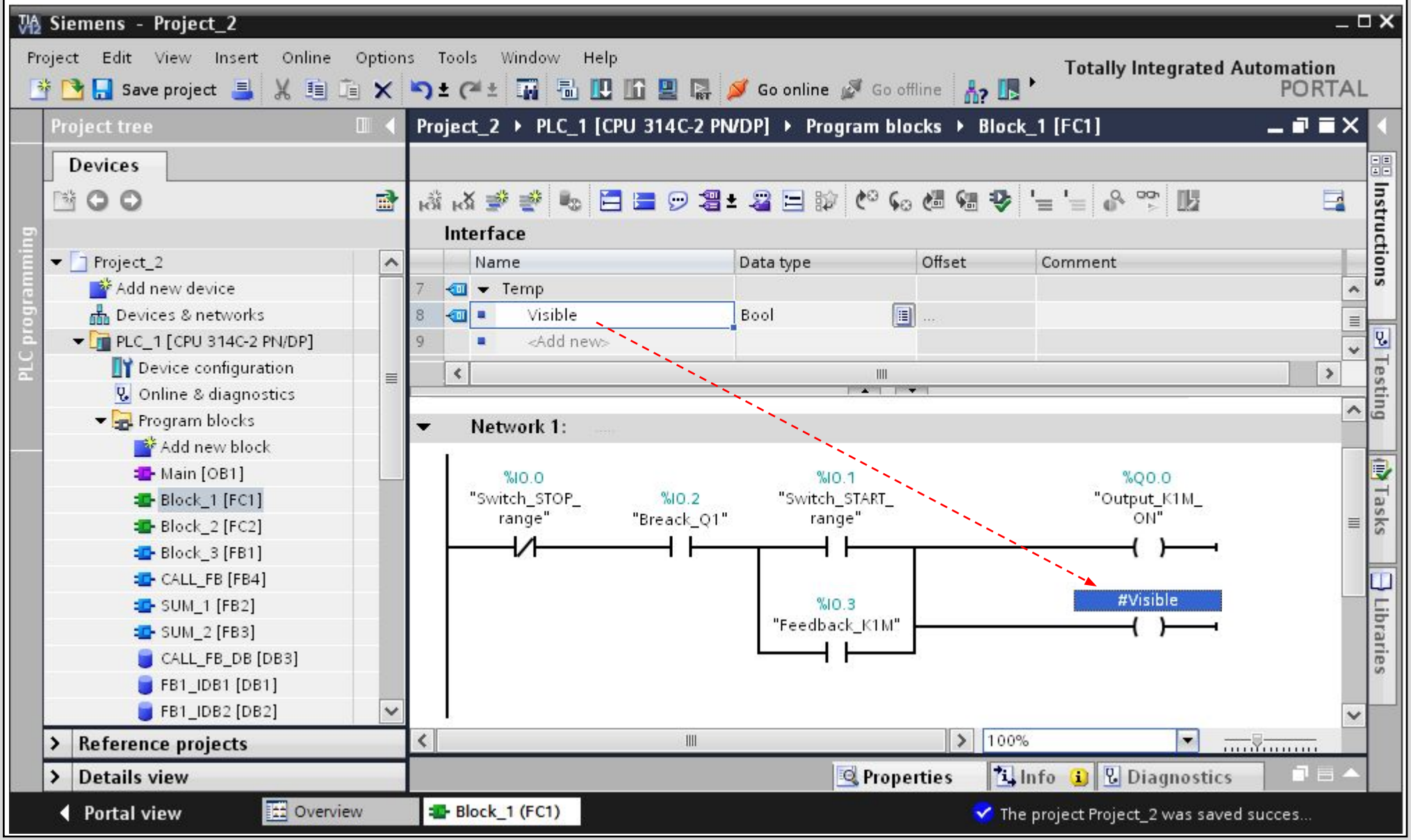
Мультиэкземплярный IDB, содержащий области данных FB2 и FB3



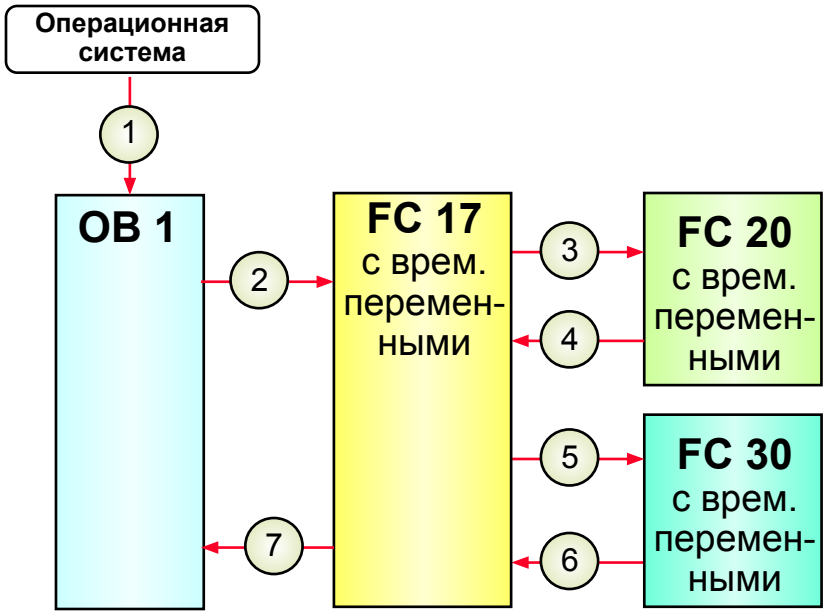
Стек локальных данных. Область TEMP в интерфейсе блока

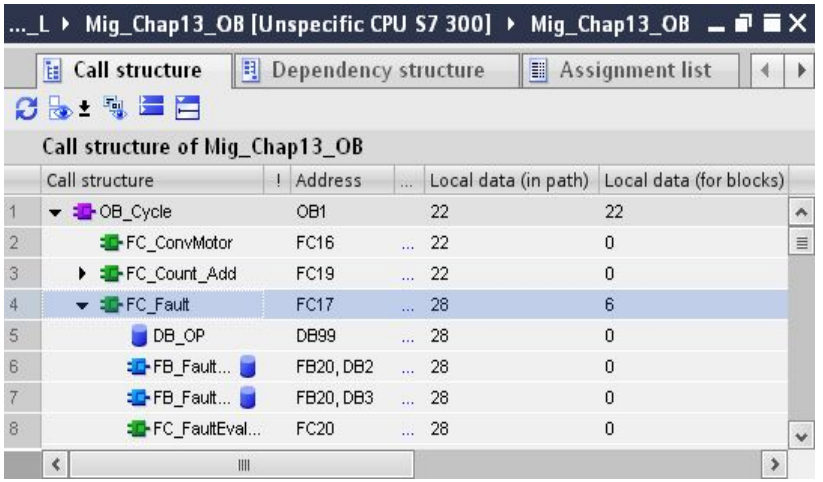


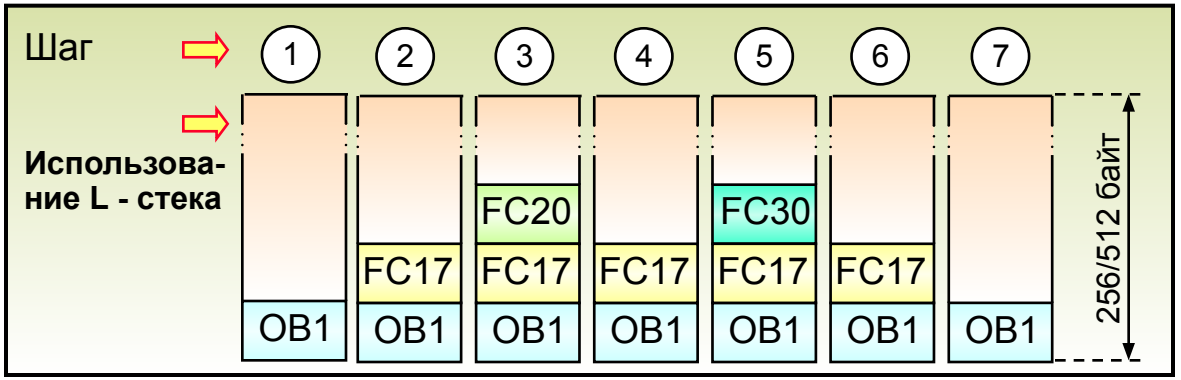
Стек локальных данных. Область TEMP в интерфейсе блока



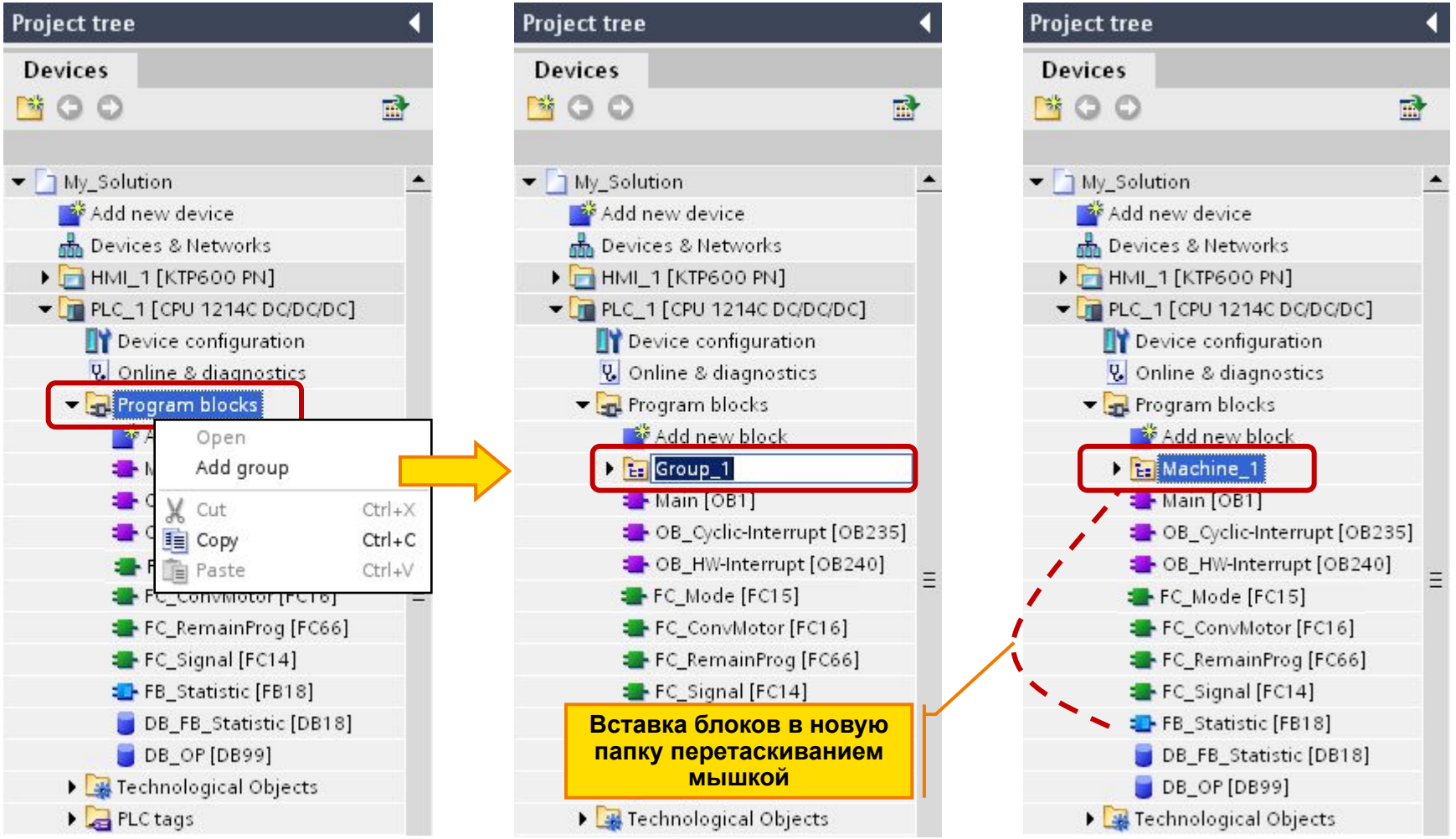
Стек локальных данных. Распределение памяти







Объединение блоков пользователя в группы

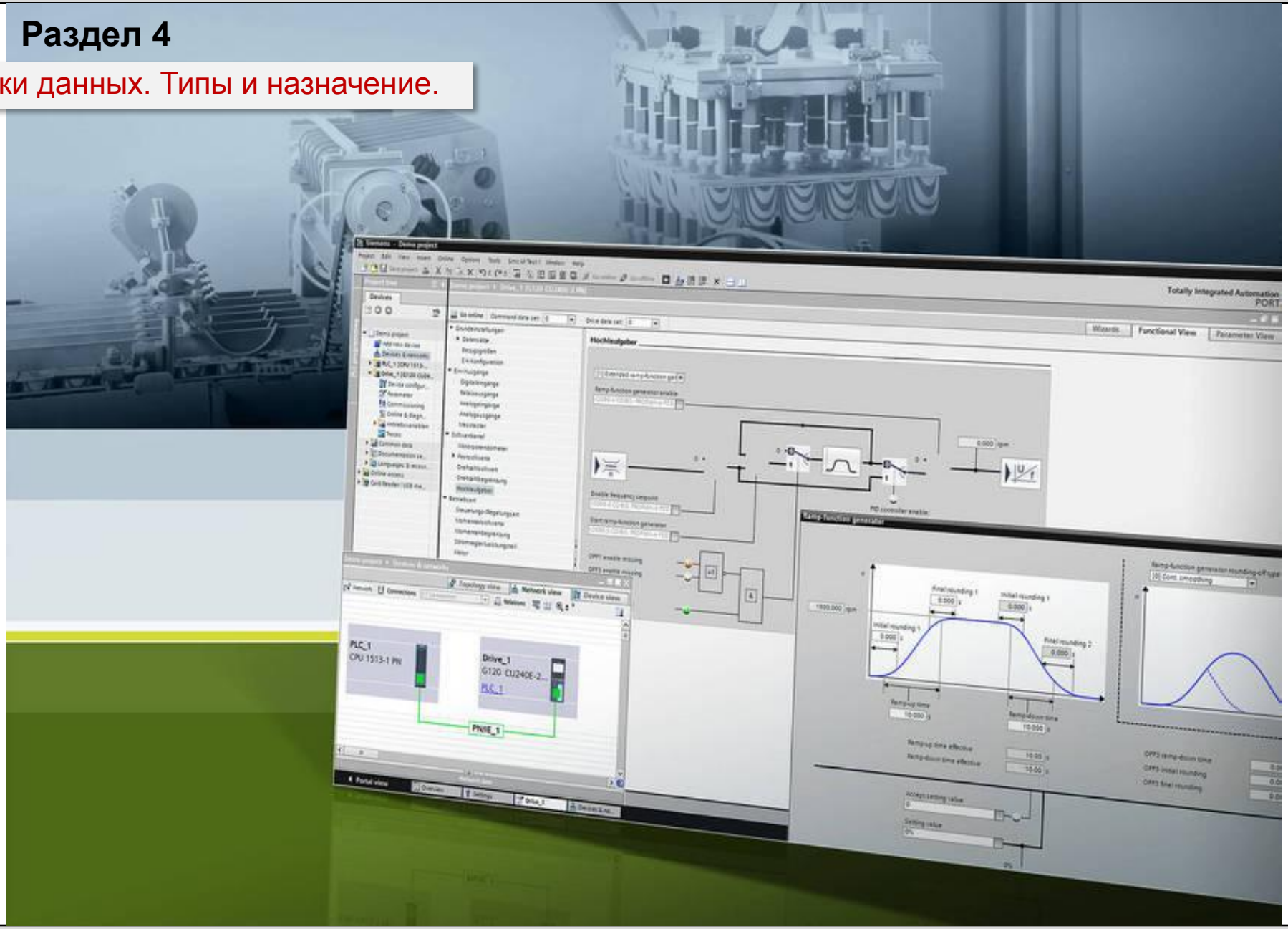


Конец раздела 3. Окно навигации

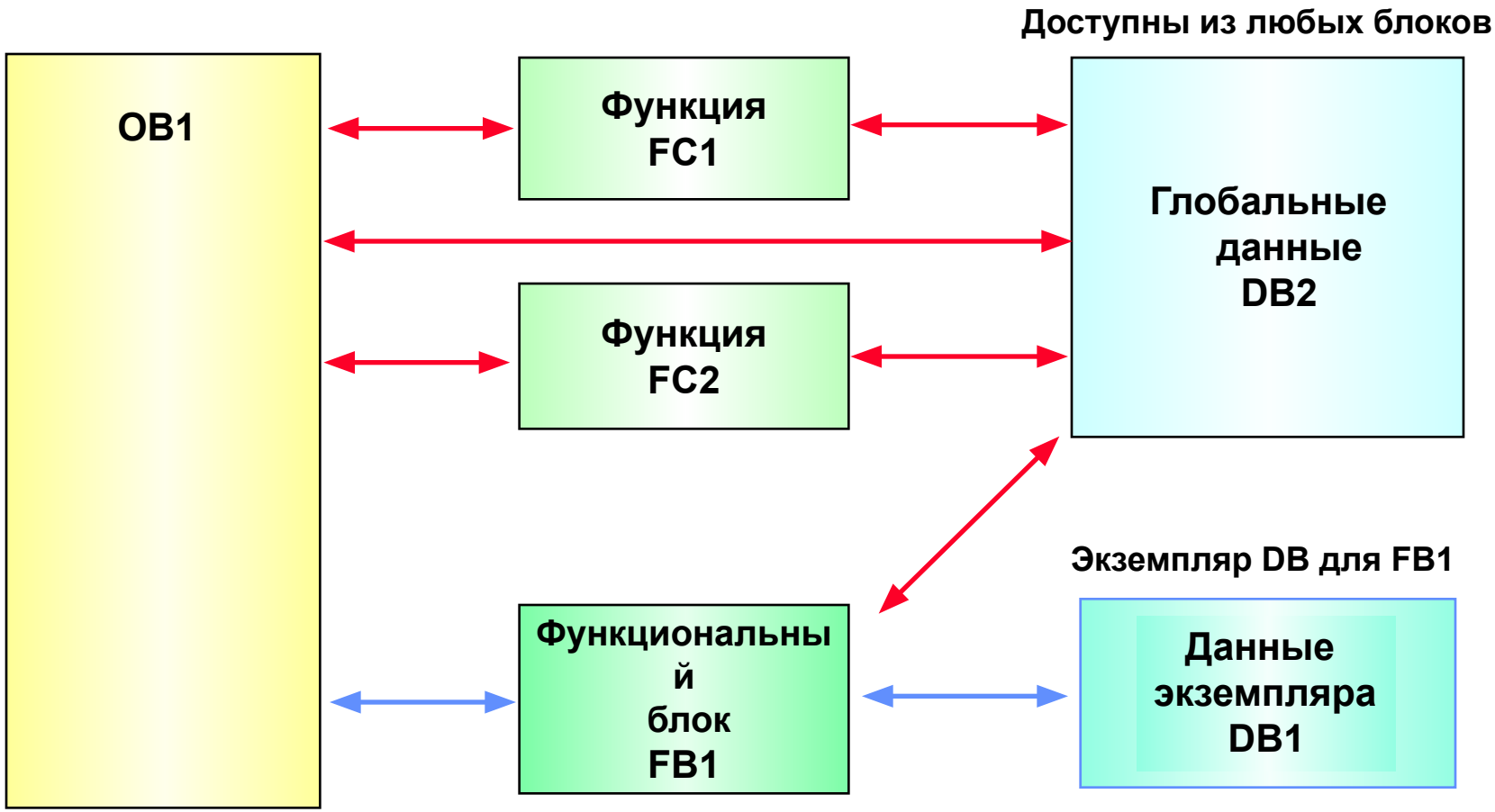
- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 4

Блоки данных. Типы и назначение.



Блоки данных (DB)



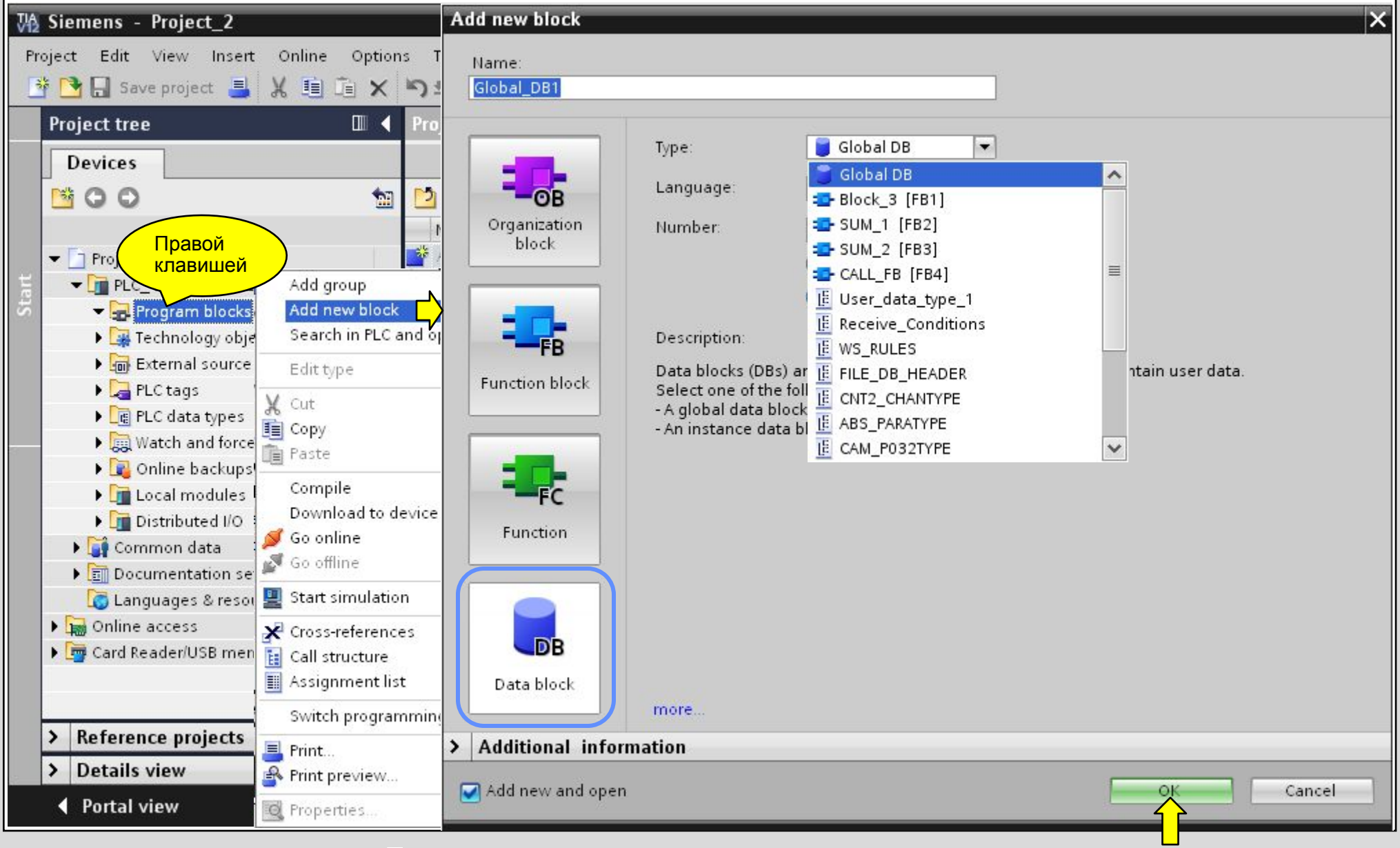
Элементарные типы данных

Обозначение	Длина (биты)	Пример константы данного типа
BOOL	1	1 или 0
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	' w '
S5TIME	16	S5T#5s_200ms
INT	16	34789 (-34789)
DINT	32	109765
REAL	32	1245. 098
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45.12

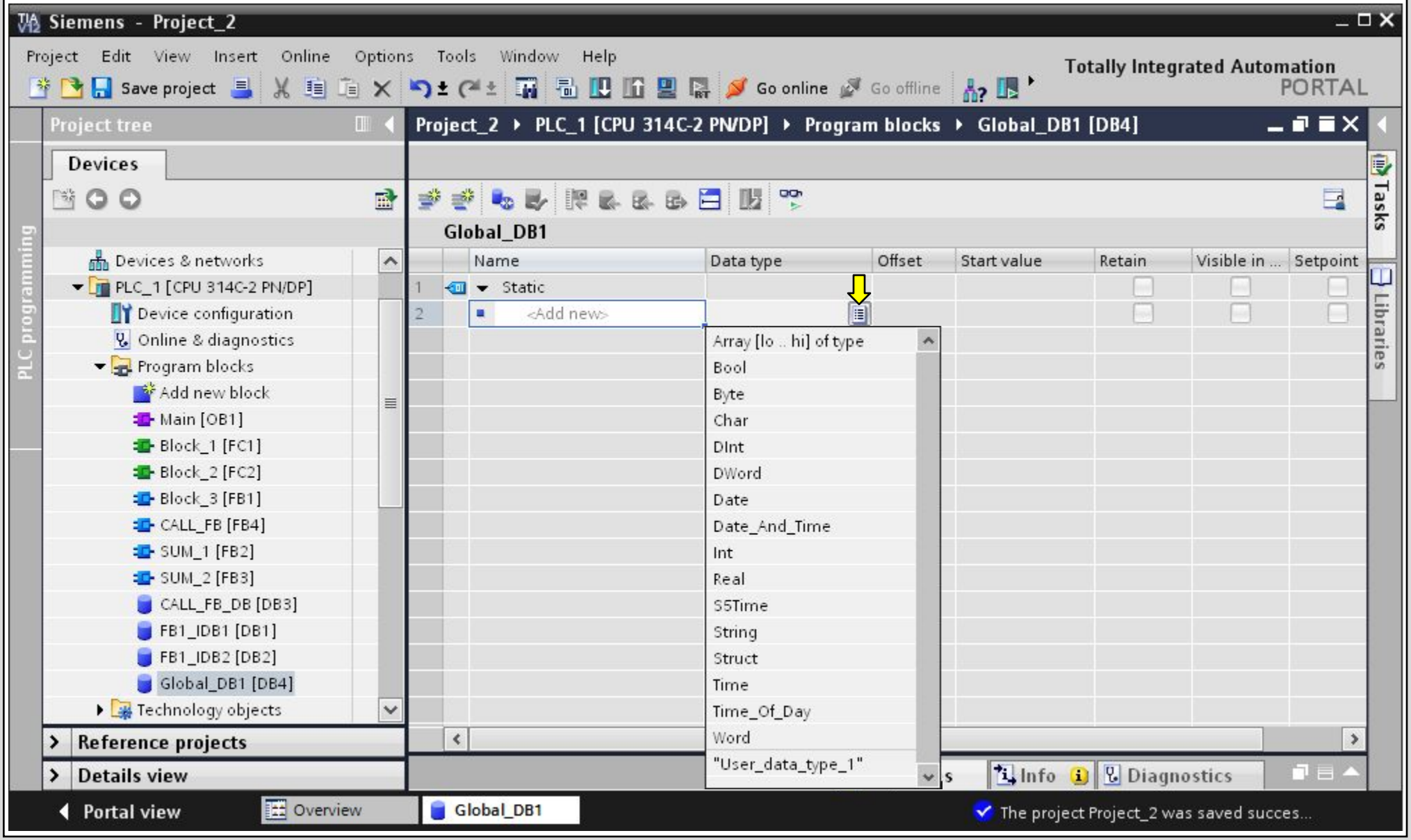
Сложные типы данных

Ключевое слово	Длина (биты)	Пример
DATE_AND_TIME	64	DT#97-09-24-12:14:55.0
STRING (строка символов макс. 254 символа)	8 * (число символов +2)	´This is a string´ ´SIEMENS´
ARRAY (Группа элементов одного типа)	определяется пользователем	Measured values: ARRAY[1..20] INT
STRUCT (Группа элементов различных типов)	определяется пользователем	Motor: STRUCT Speed : INT Current: REAL END_STRUCT
UDT (**U**ser **D**efined Data **T**ype = пользовательский тип данных “Шаблон”)	UDT как блок END_STRUCT	UDT - элемент определяется пользователем Speed : INT
		STRUCT Drive: ARRAY[1..4] UDT1 Current: REAL

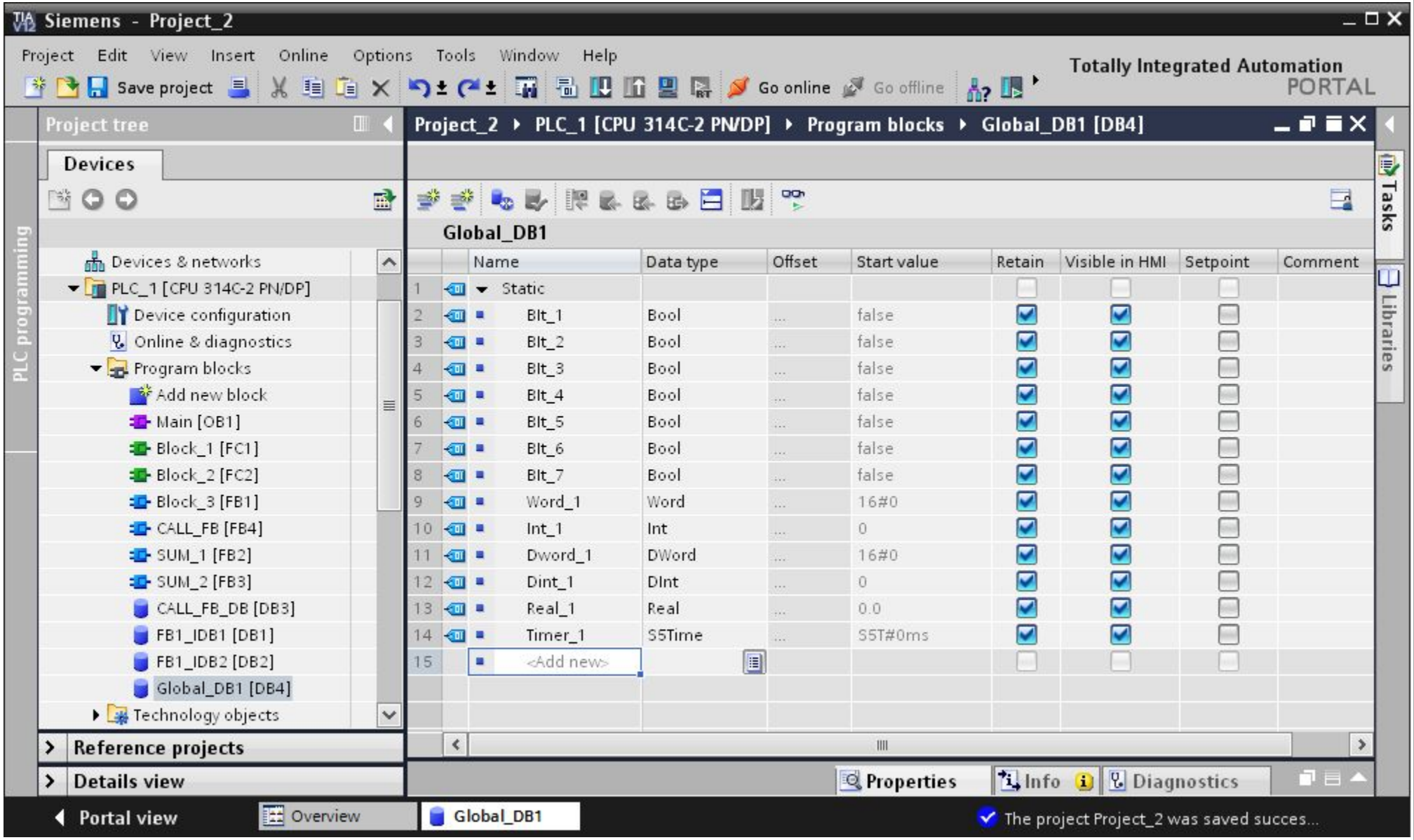
Создание глобального DB



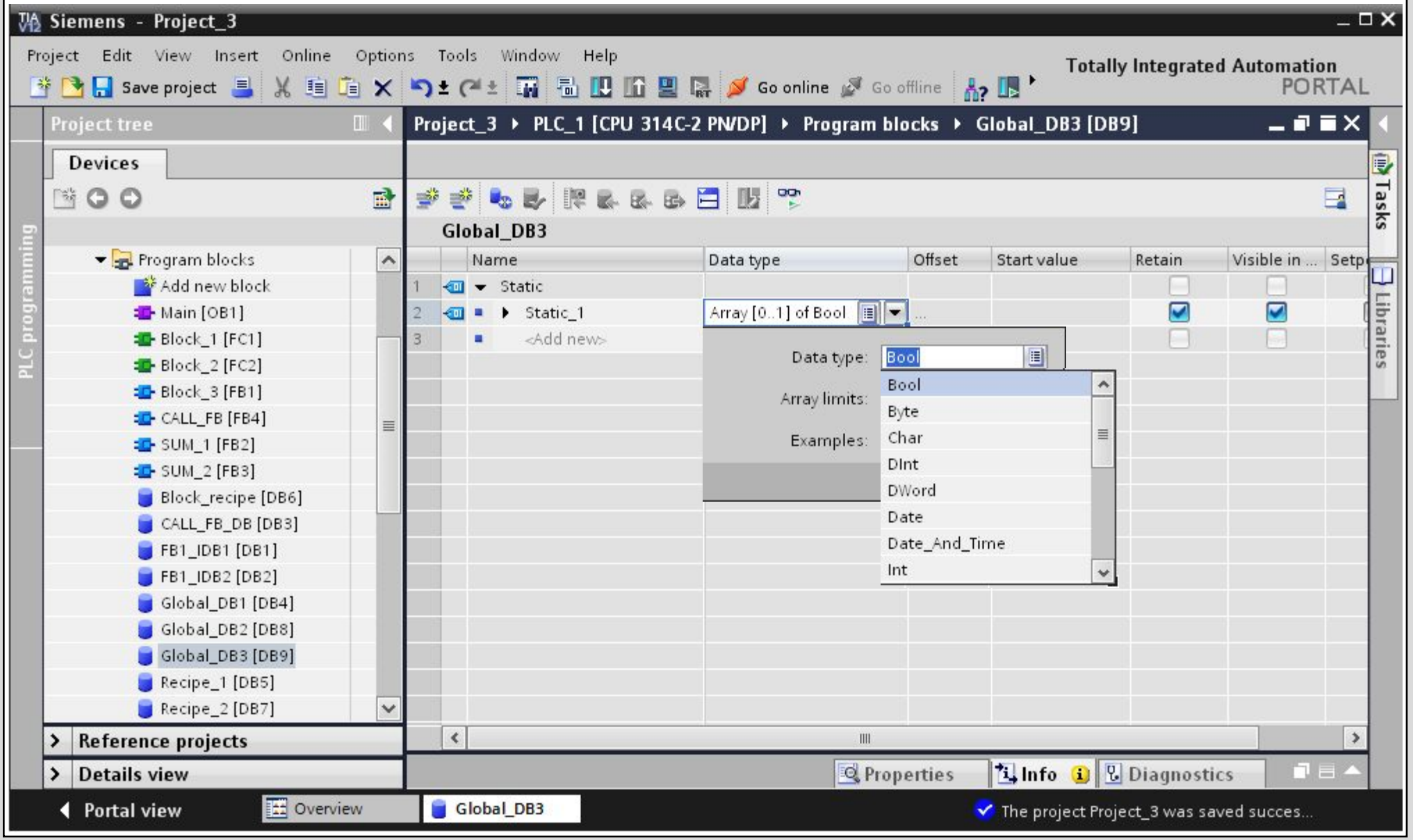
Создание глобального DB



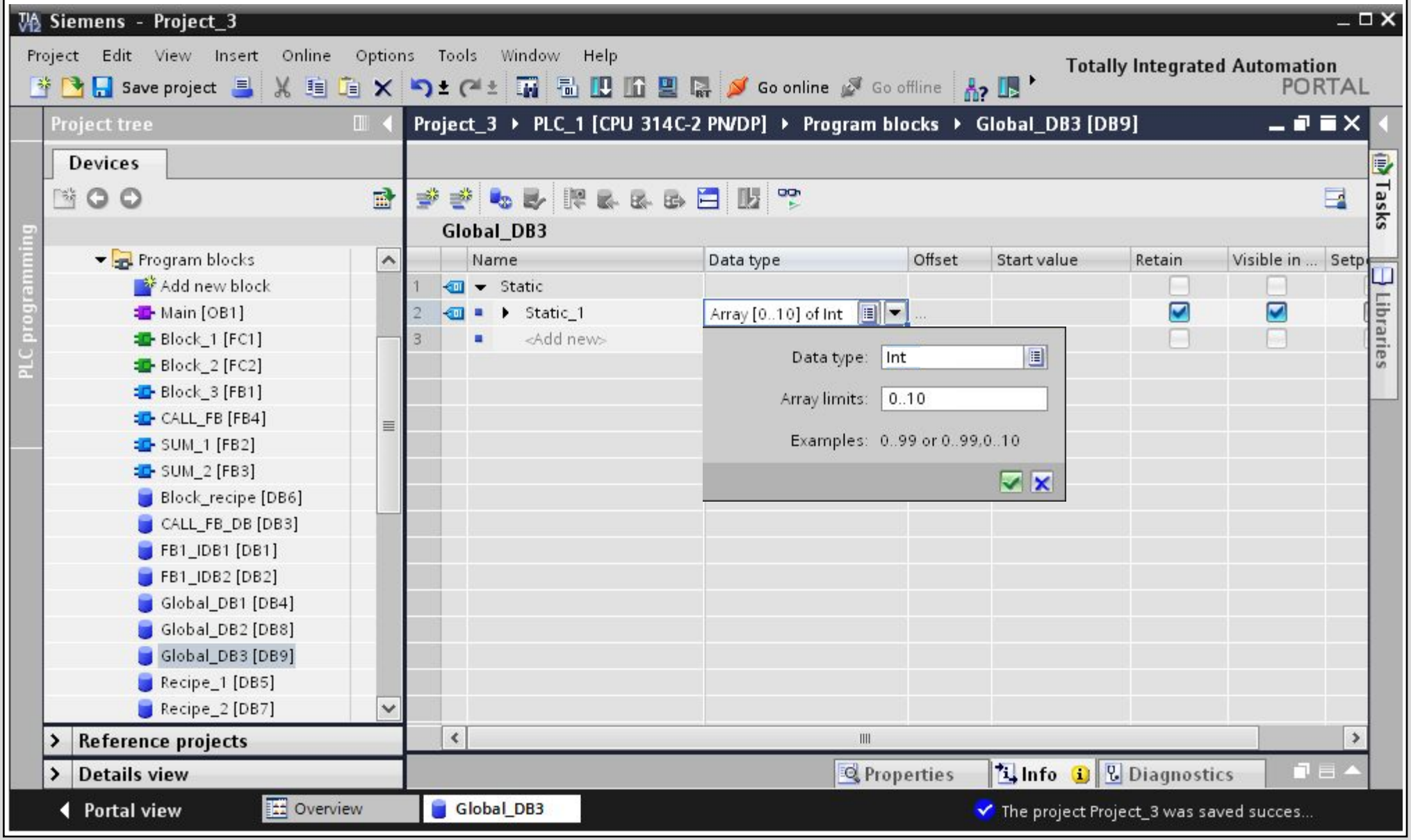
Создание глобального DB



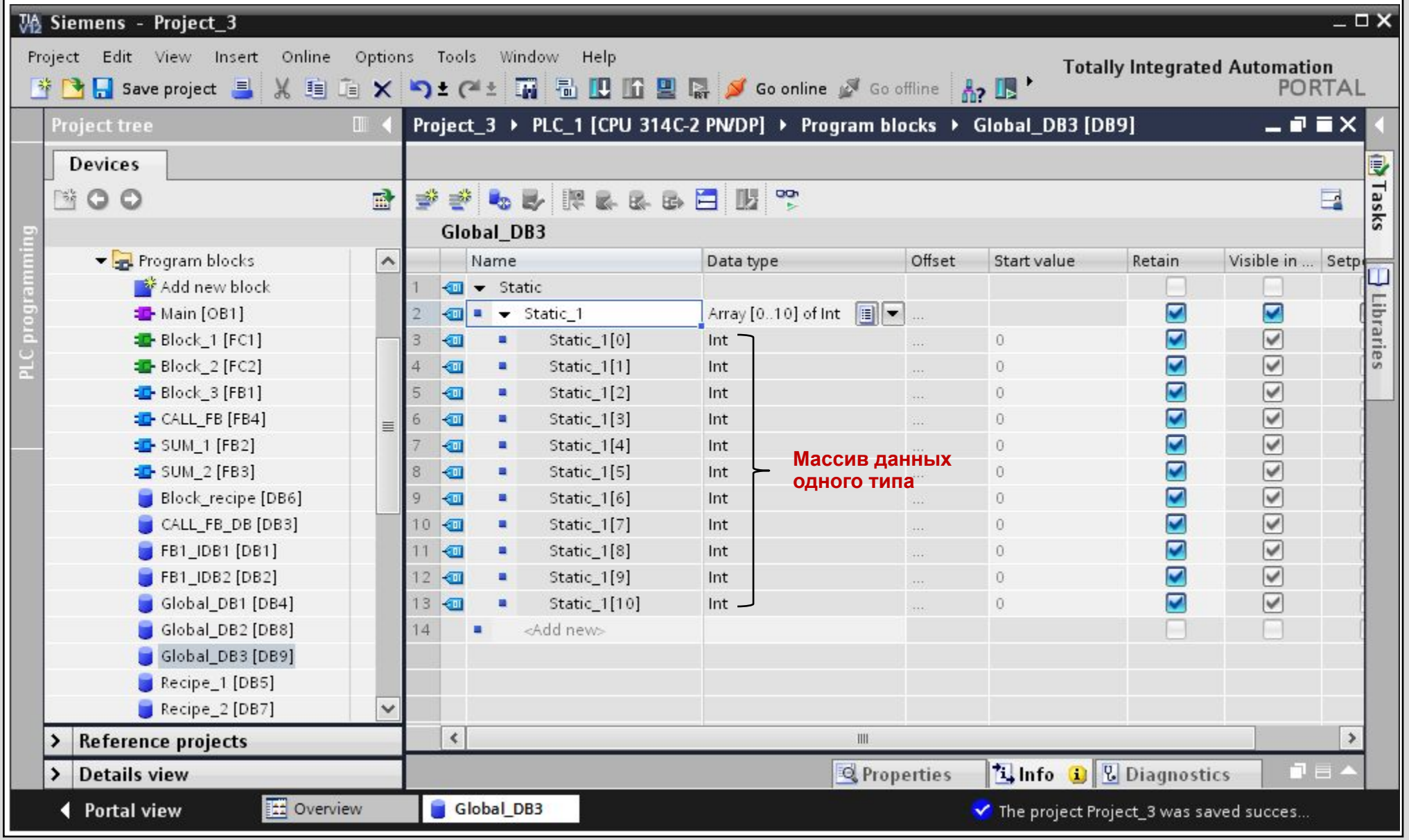
Создание массива в DB



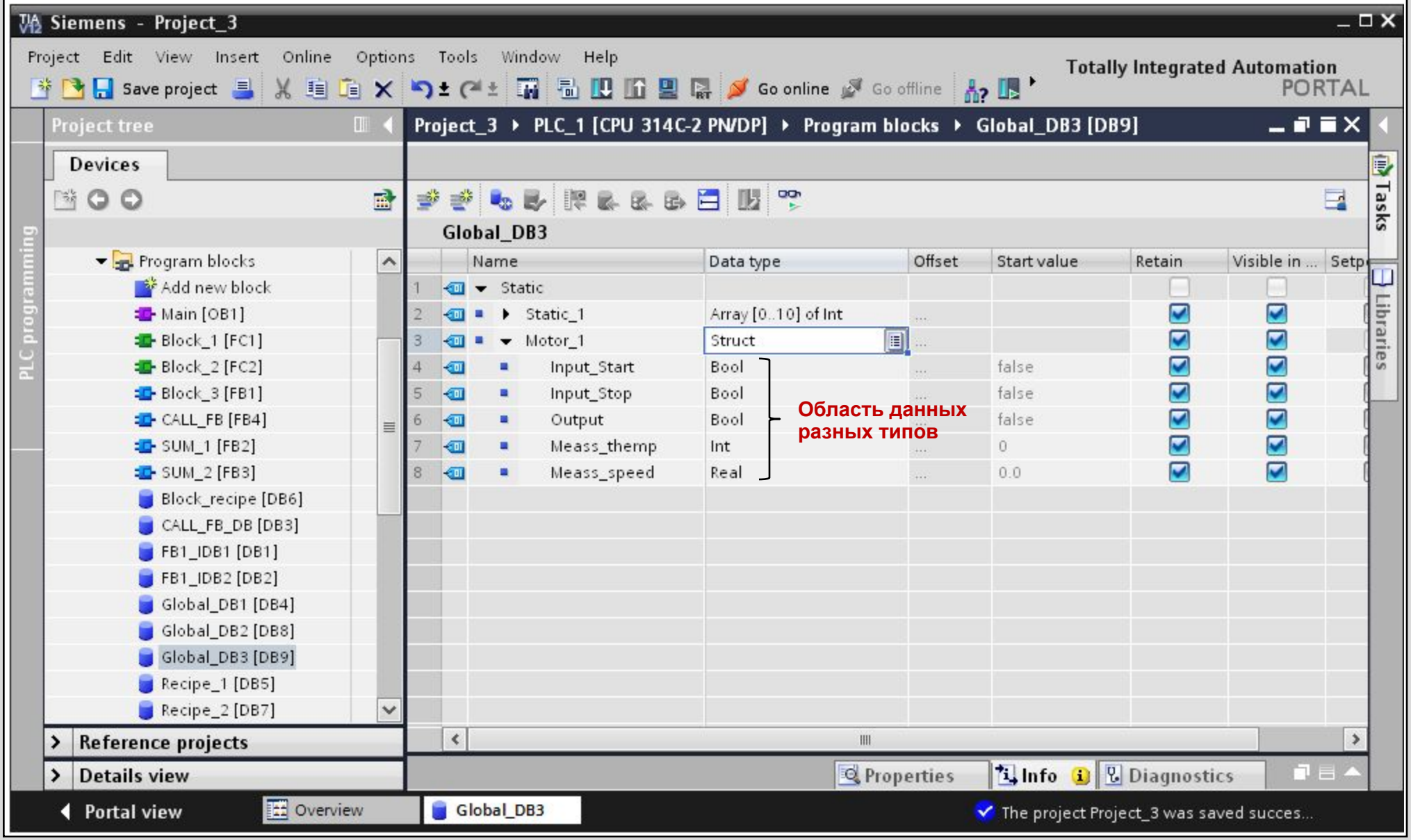
Создание массива в DB



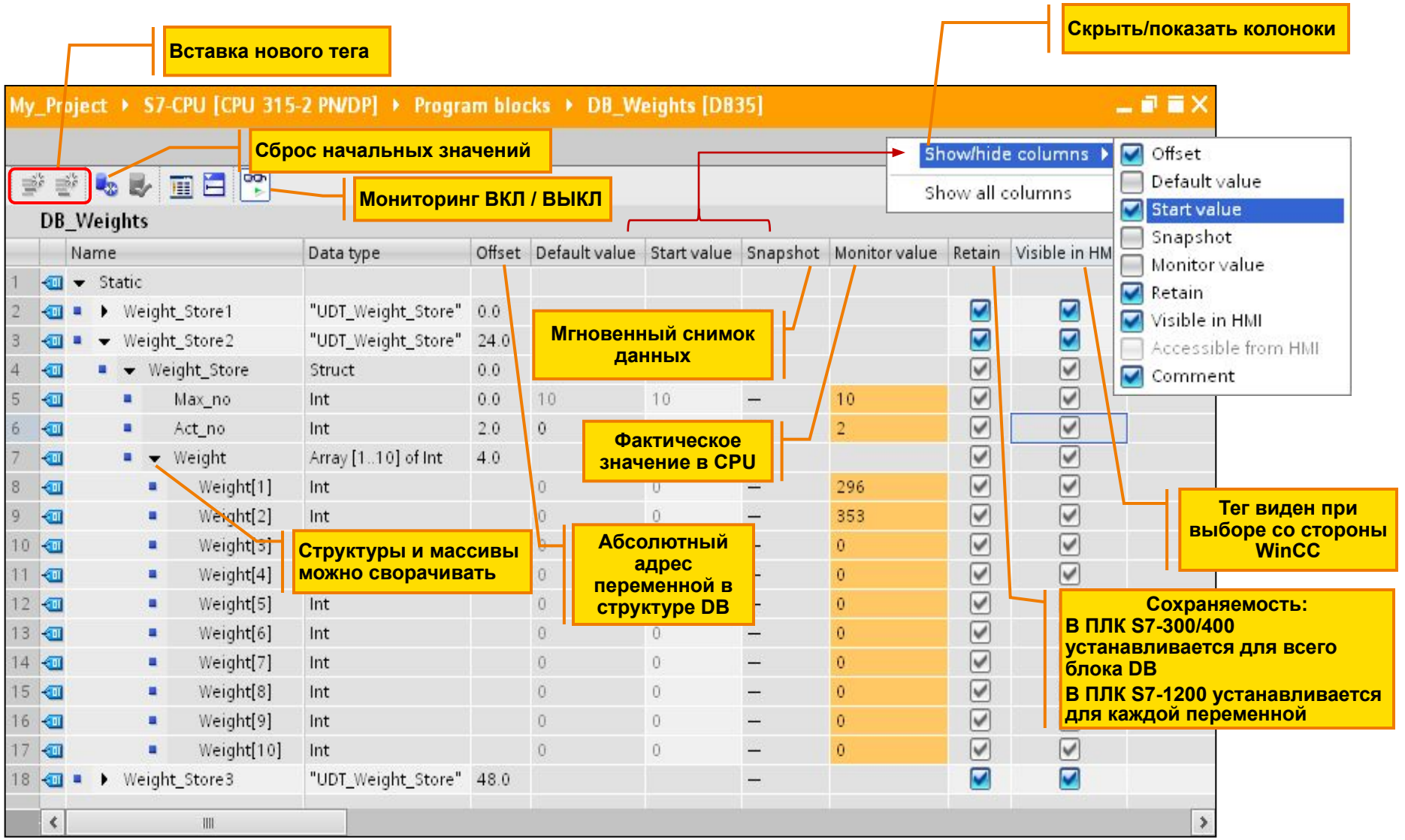
Создание массива в DB



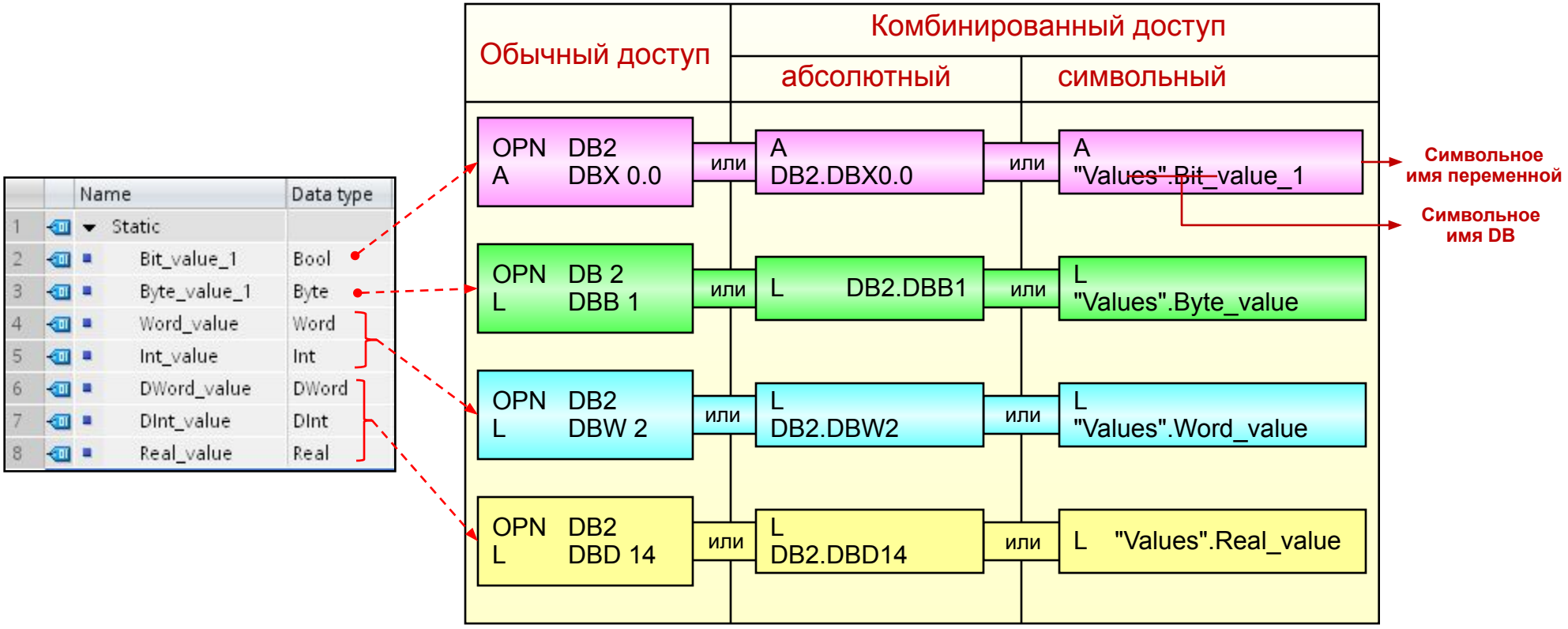
Создание структуры в DB



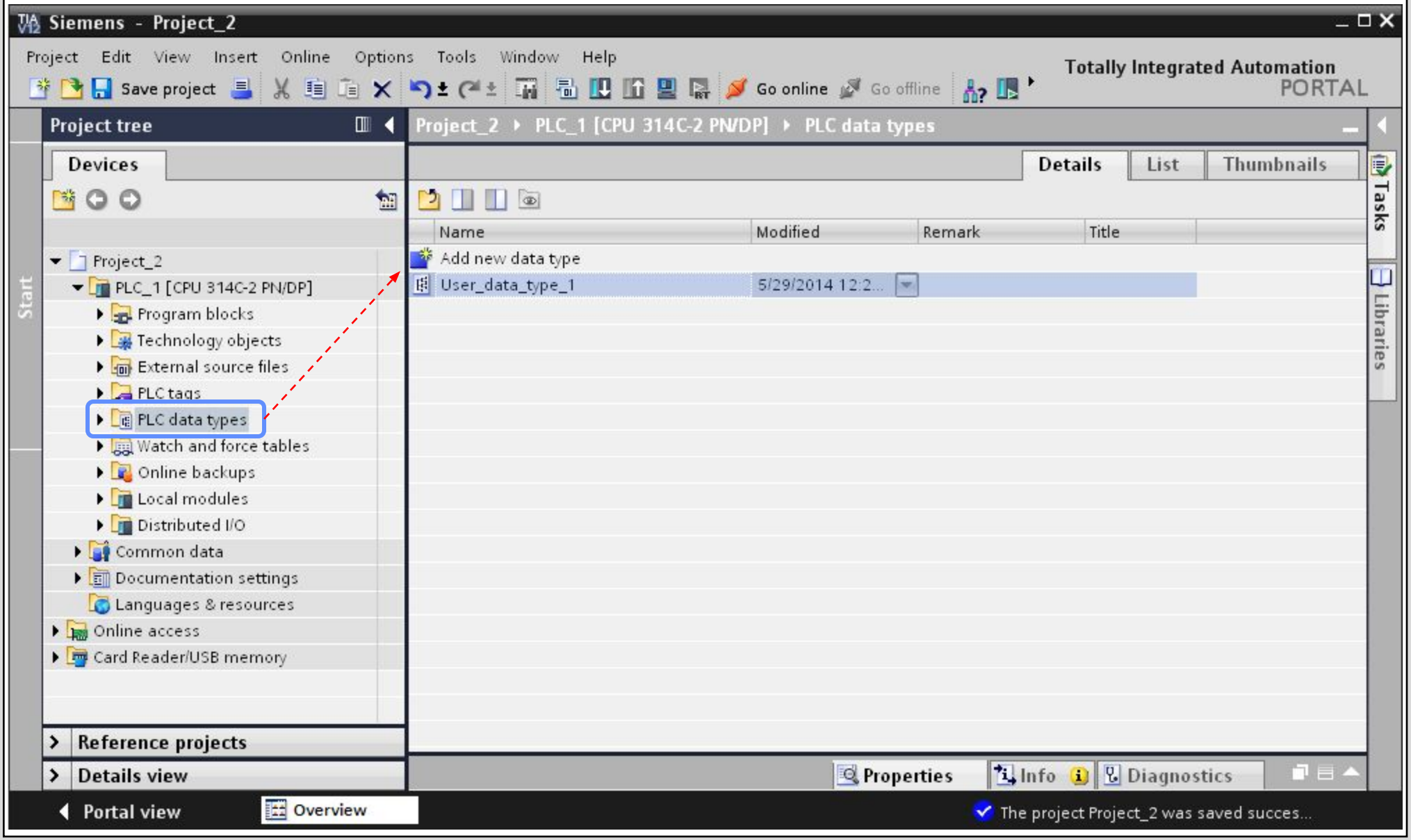
Общий вид и мониторинг данных в DB



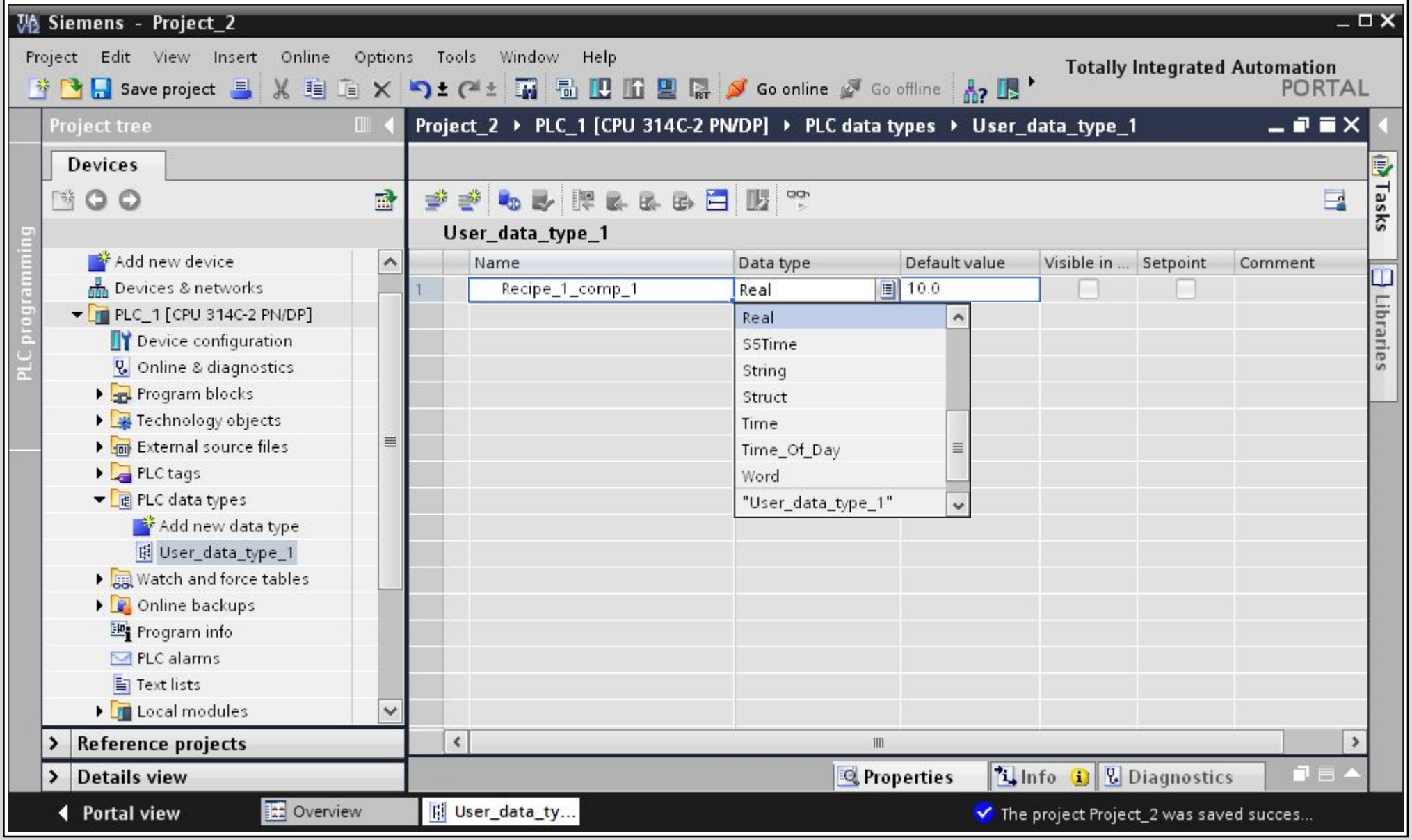
Доступ к элементам данных в DB



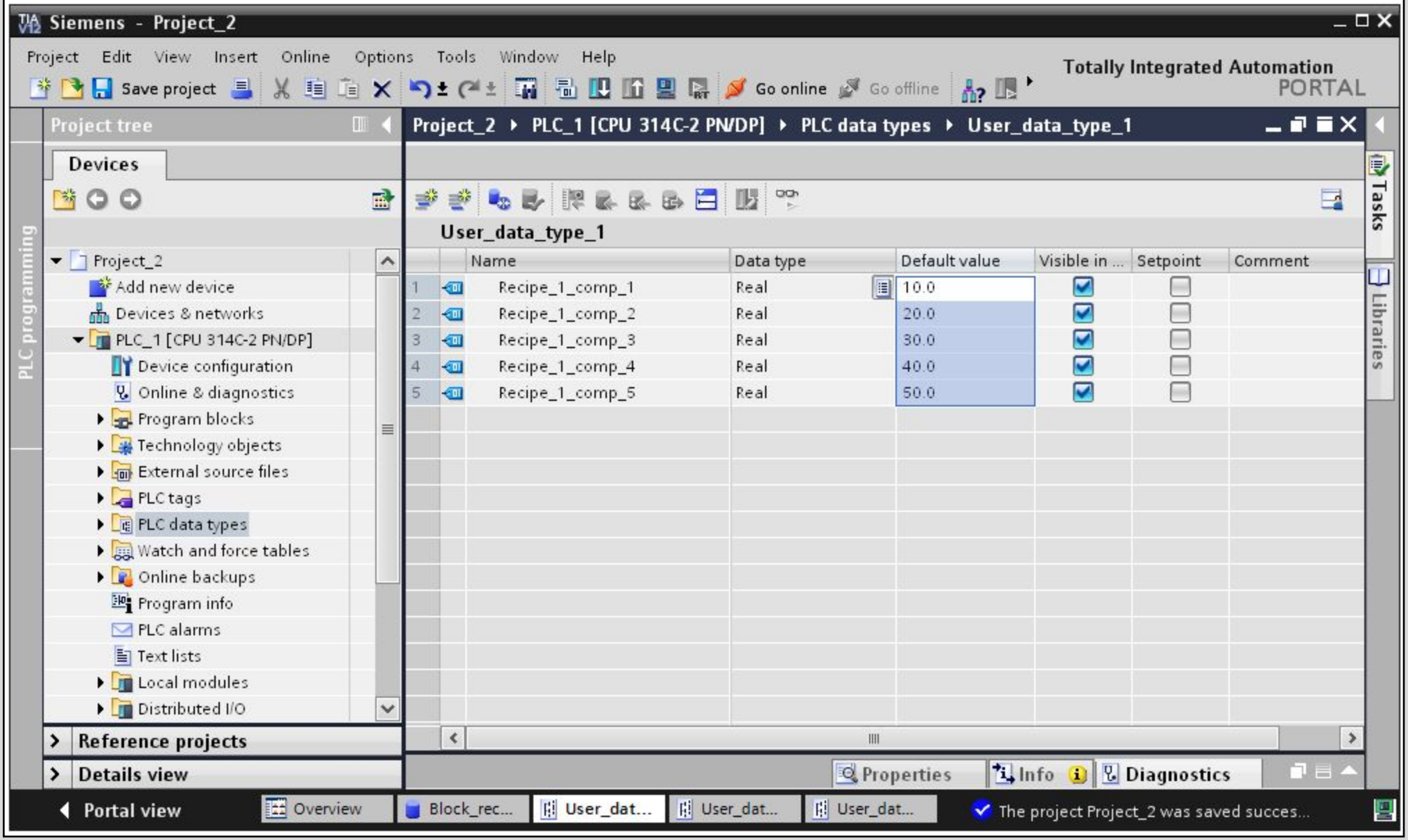
Создание таблицы данных пользователя (UDT-1)



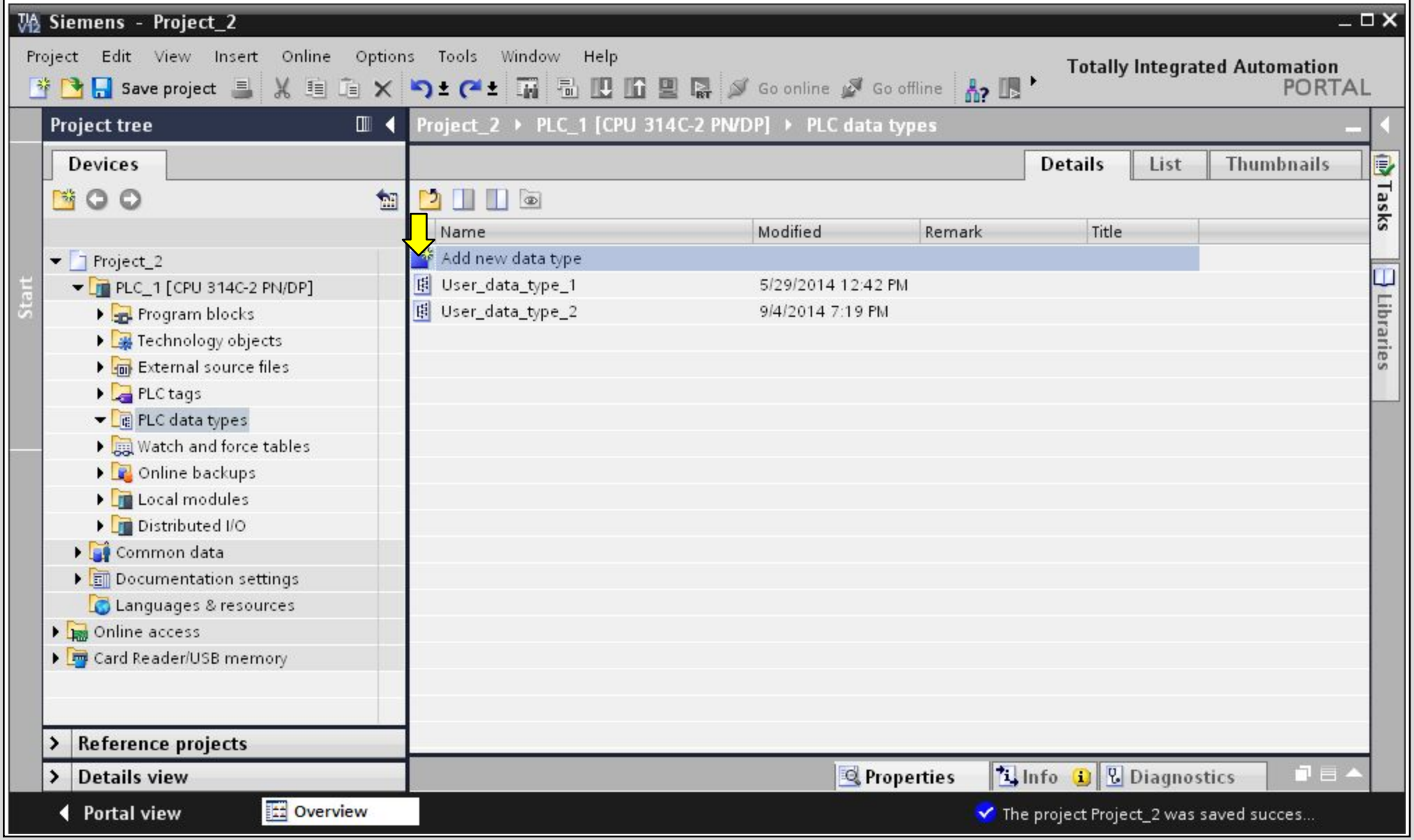
Заполнение таблицы данных пользователя UDT-1



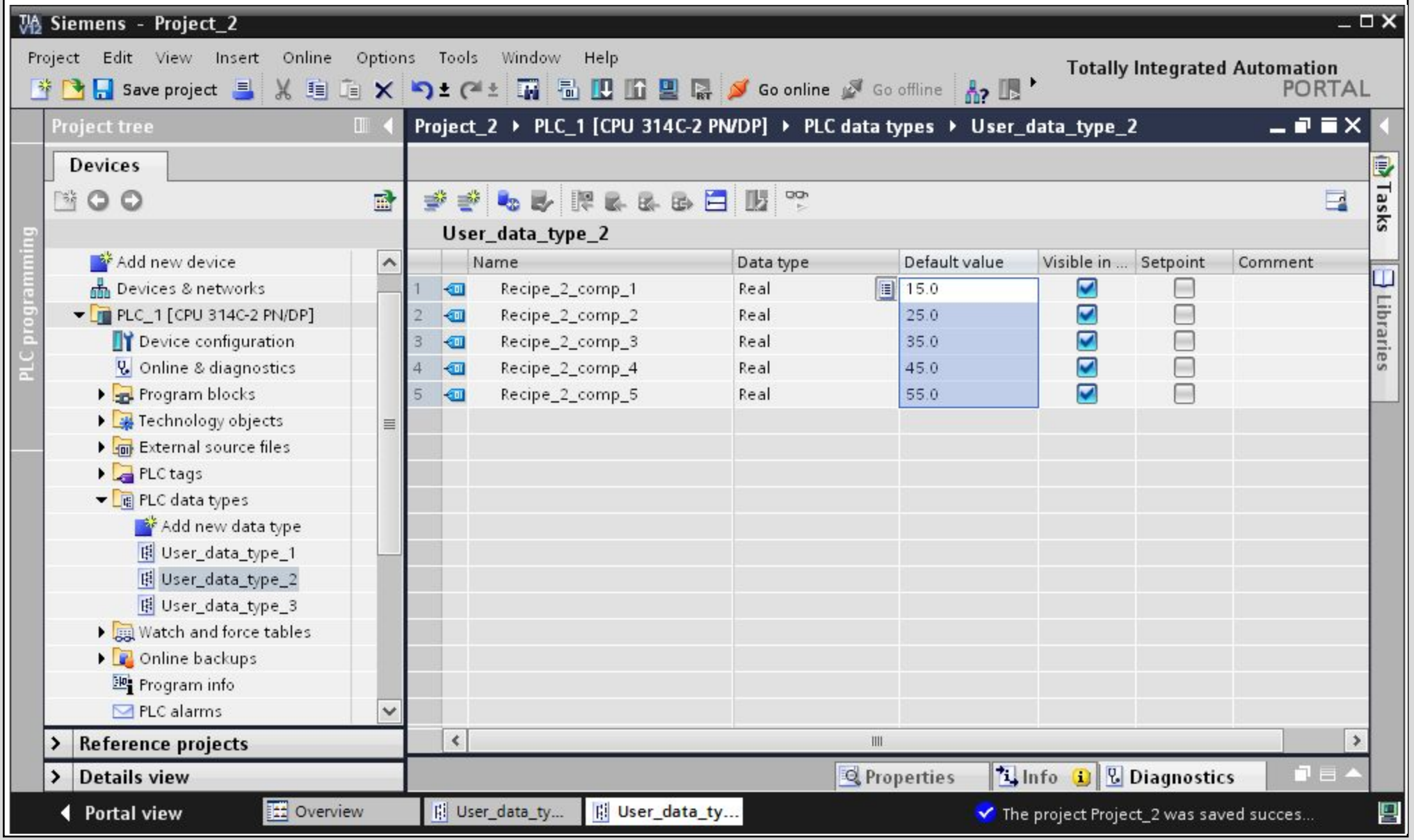
Заполнение таблицы данных пользователя UDT-1



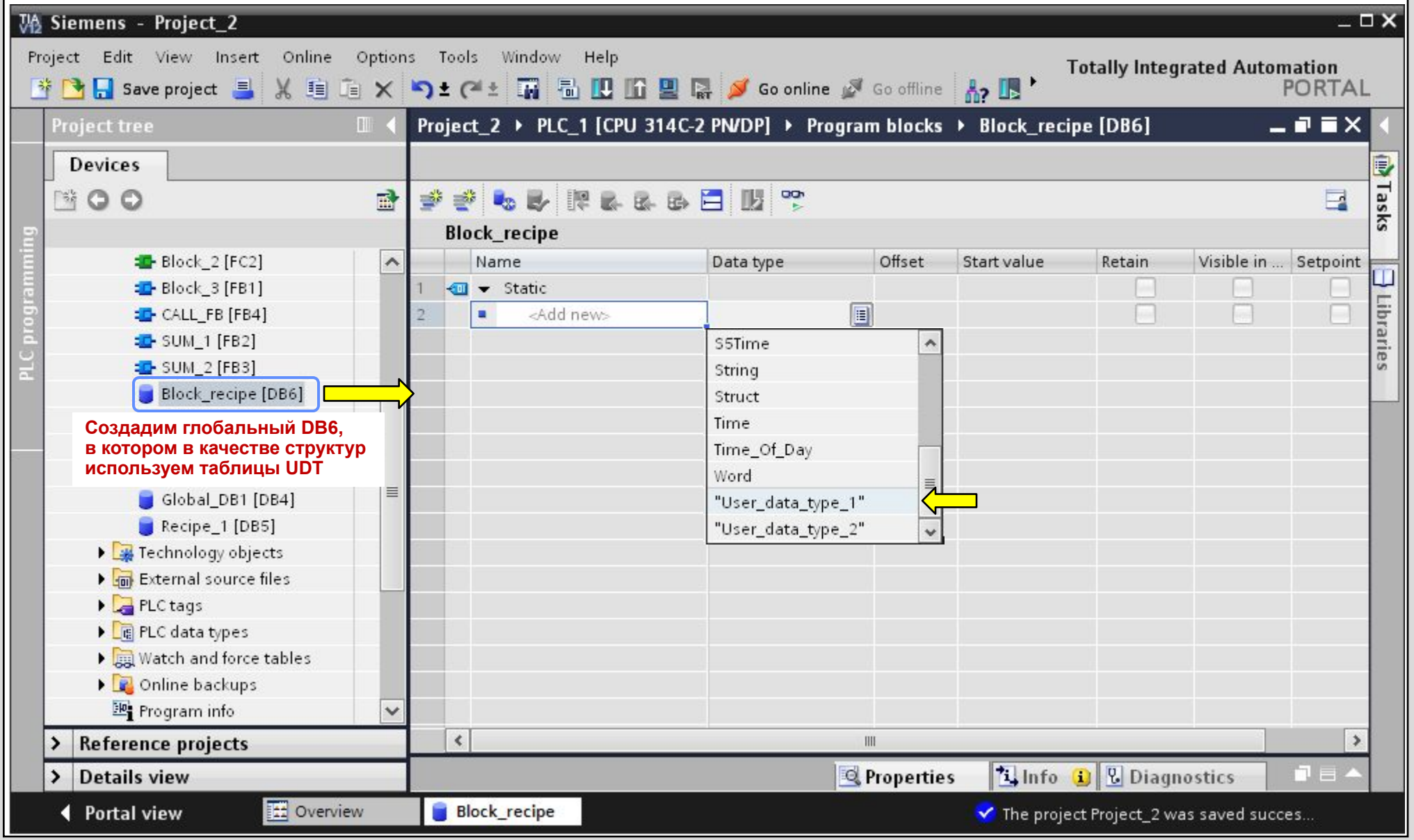
Создание таблицы данных пользователя (UDT-2)



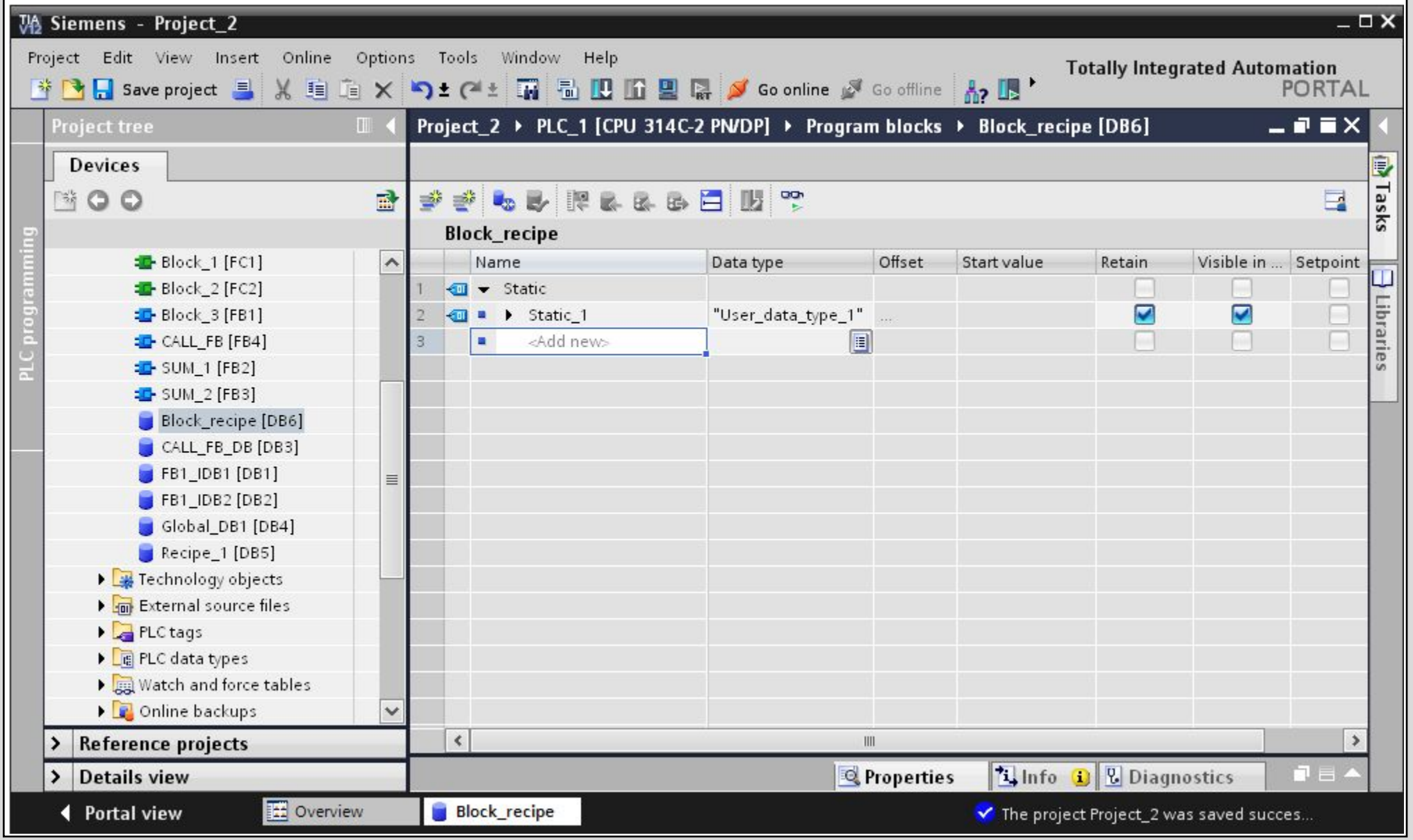
Заполнение таблицы данных пользователя UDT-2



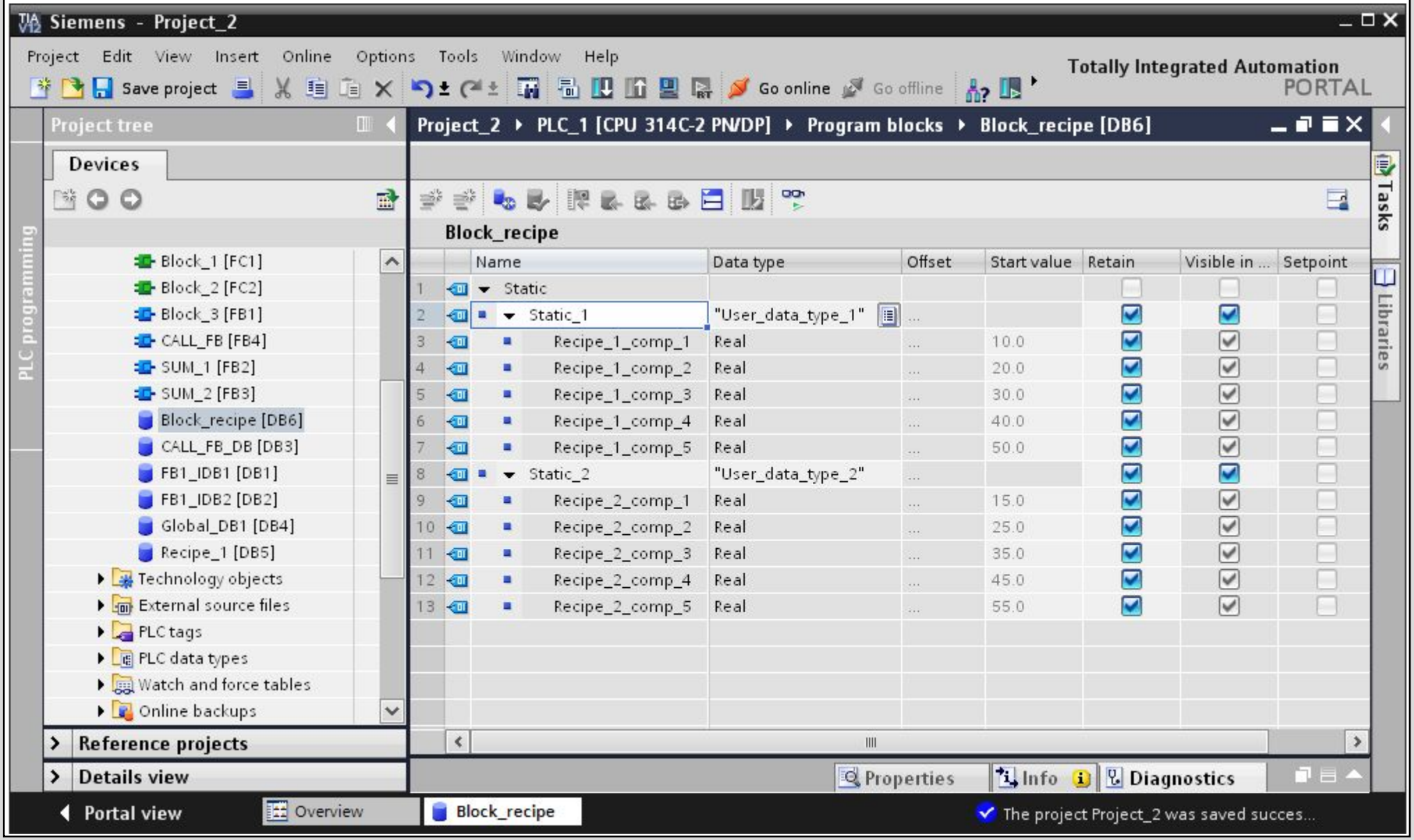
Использование таблицы UDT как шаблона данных в DB



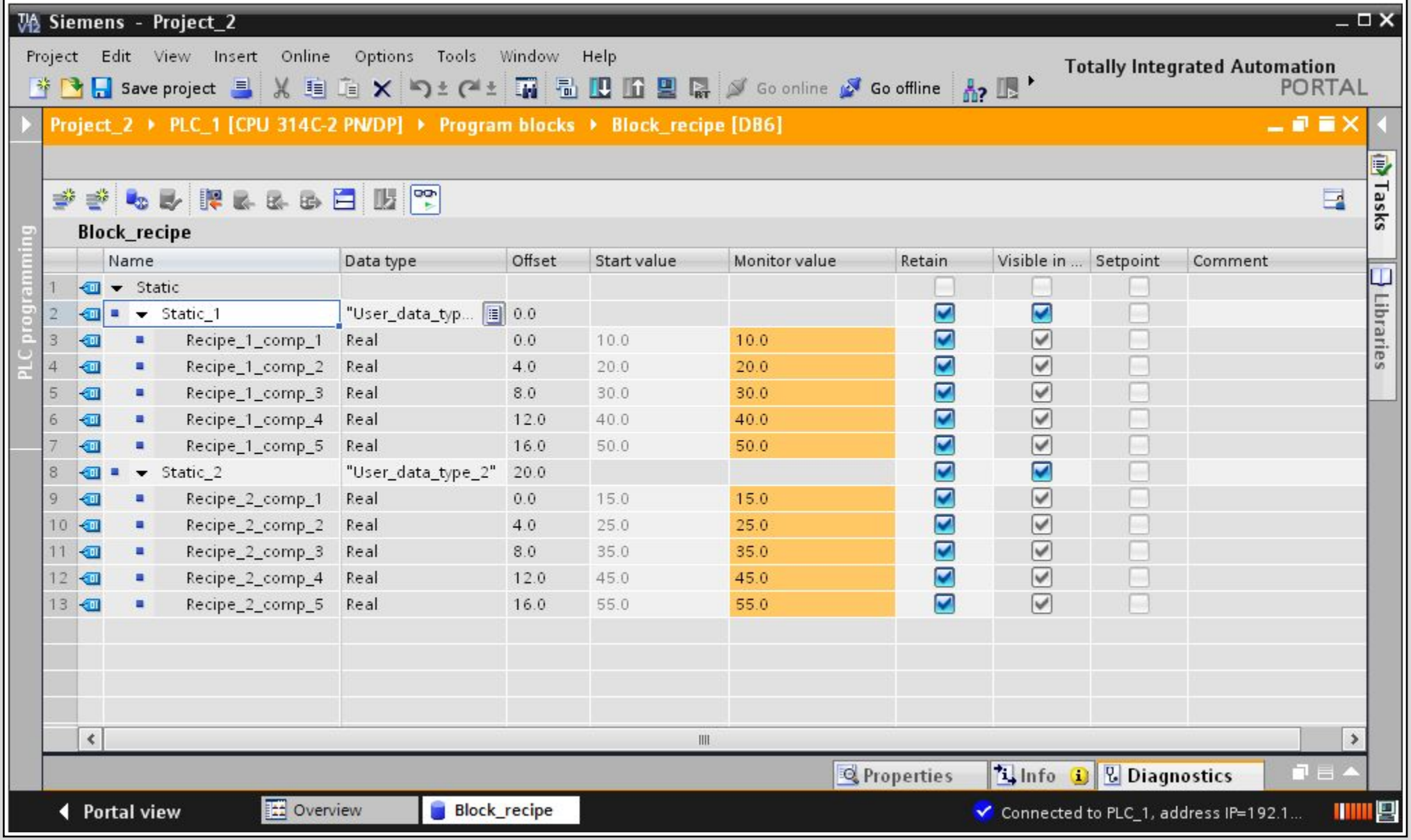
Использование таблицы UDT как шаблона данных в DB



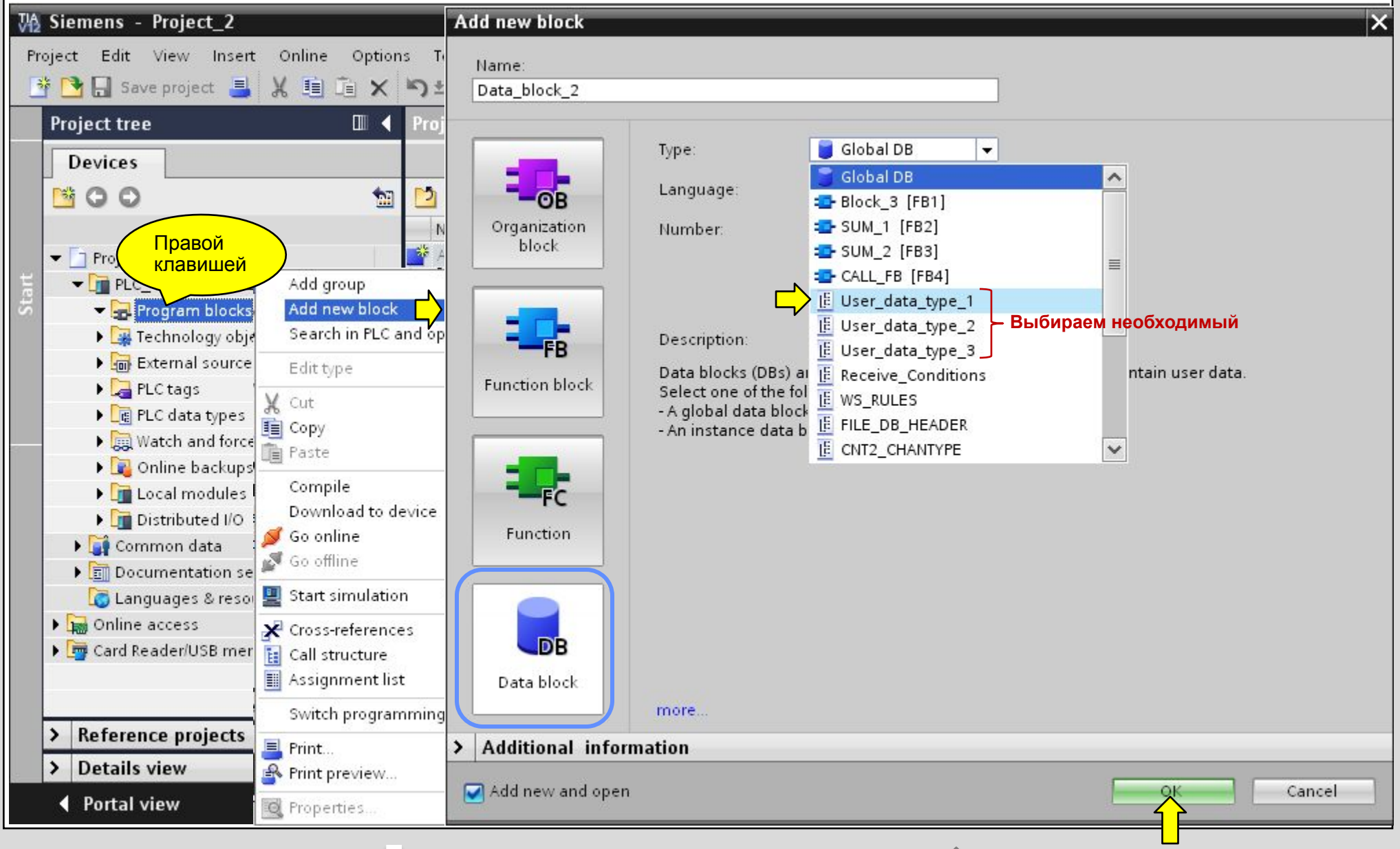
Использование таблицы UDT как шаблона данных в DB



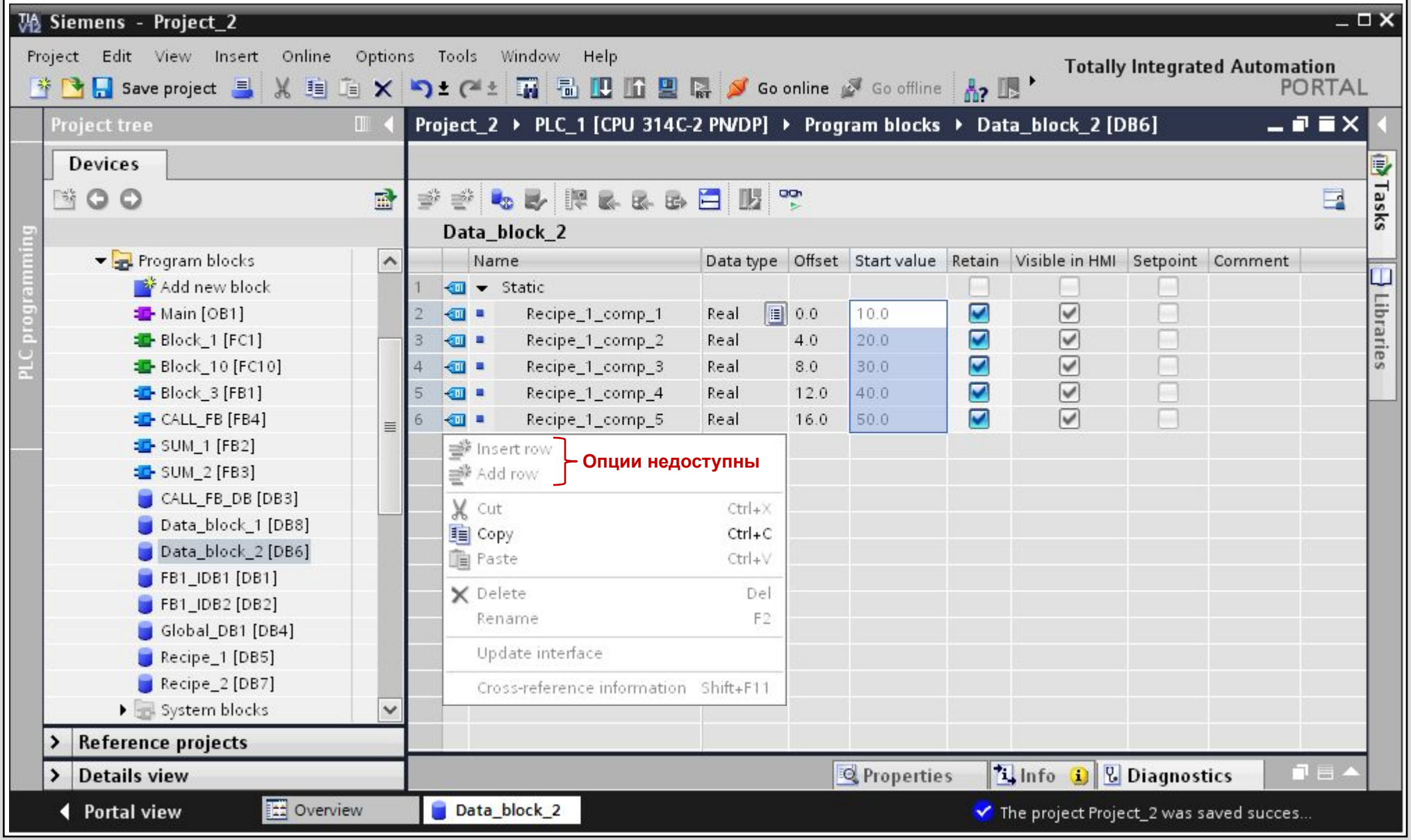
Просмотр глобального DB с двумя массивами рецептов



Создание DB с одним массивом UDT



Создание DB с одним массивом UDT

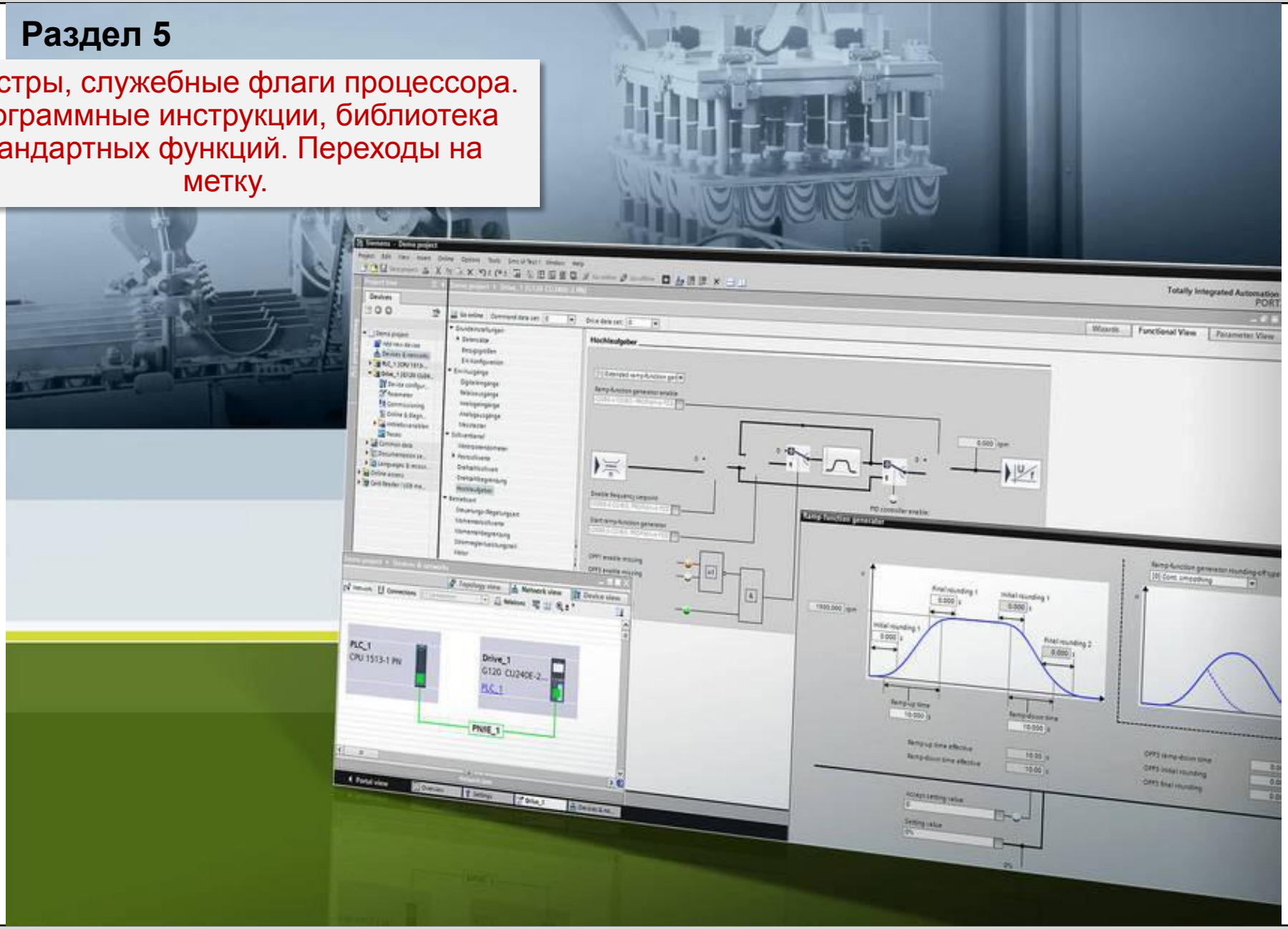


Конец раздела 4. Окно навигации

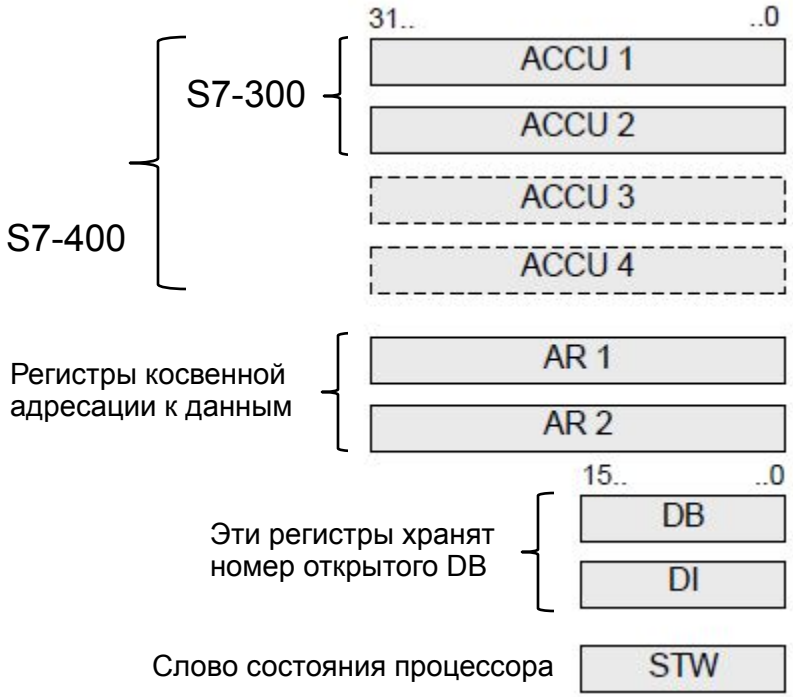
- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

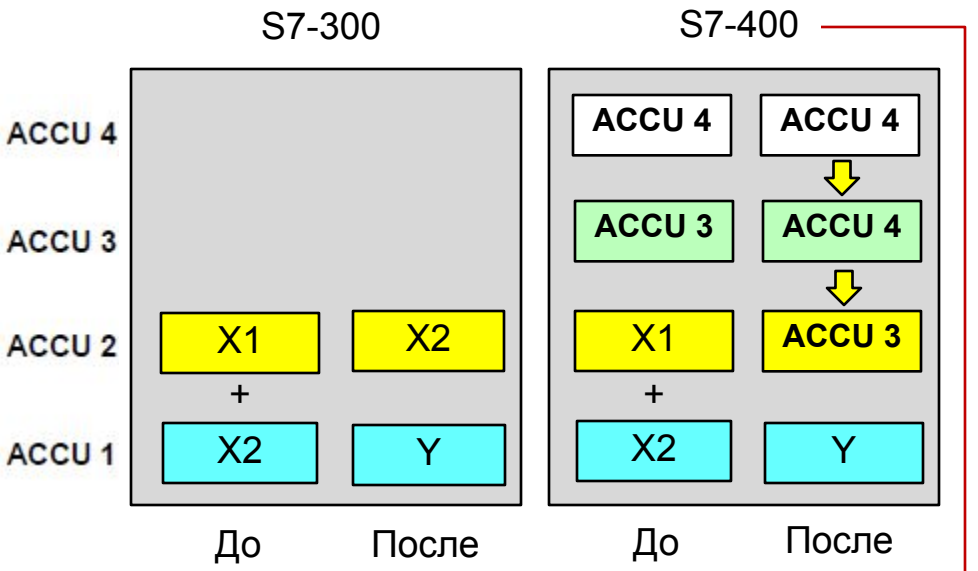
Раздел 5

Регистры, служебные флаги процессора. Программные инструкции, библиотека стандартных функций. Переходы на метку.



Регистры и аккумуляторы S7-300, S7-400



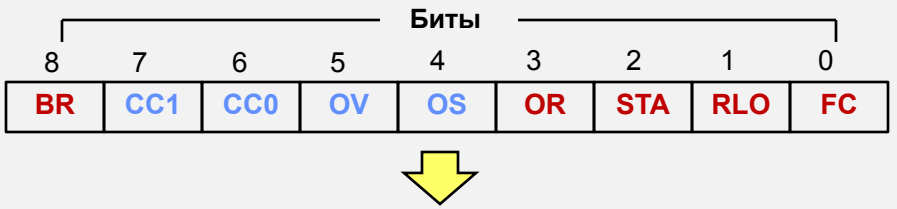


Пример: работа с четырьмя аккумуляторами. Команда **ENT** (ввод в стек аккумуляторов) копирует содержимое ACCU 3 в ACCU 4 и содержимое ACCU 2 в ACCU 3.

```
                Value
L     %MD100    5.0    //Загрузка MD100 в ACCU1
L     %MD104    5.0    //Передача MD100 в ACCU2, загрузка md104 в ACCU1
+R              10.0   //Сложение ACCU1+ACCU2 и пересылка результата в ACCU1
L     %MD108    15.0   //Пересылка из ACCU1 в ACCU2. Загрузка в ACCU1 MD108
ENT             10.0   //Пересылка ACCU3 в ACCU4, пересылка ACCU2 в ACCU3
L     %MD112    10.0   //Загрузка MD112 в ACCU1
-R              5.0    //Вычитание ACCU1-ACCU2, результат в ACCU1.Копирование ACCU3 в ACCU2, ACCU4 в ACCU3
/R              2.0    //Деление ACCU2/ACCU1 (MD100+MD104)/(MD108-MD112). Результат в ACCU1
T     %MD116    2.0    //Пересылка ACCU1 в MD116
```

Служебные флаги состояния процесса (STW)

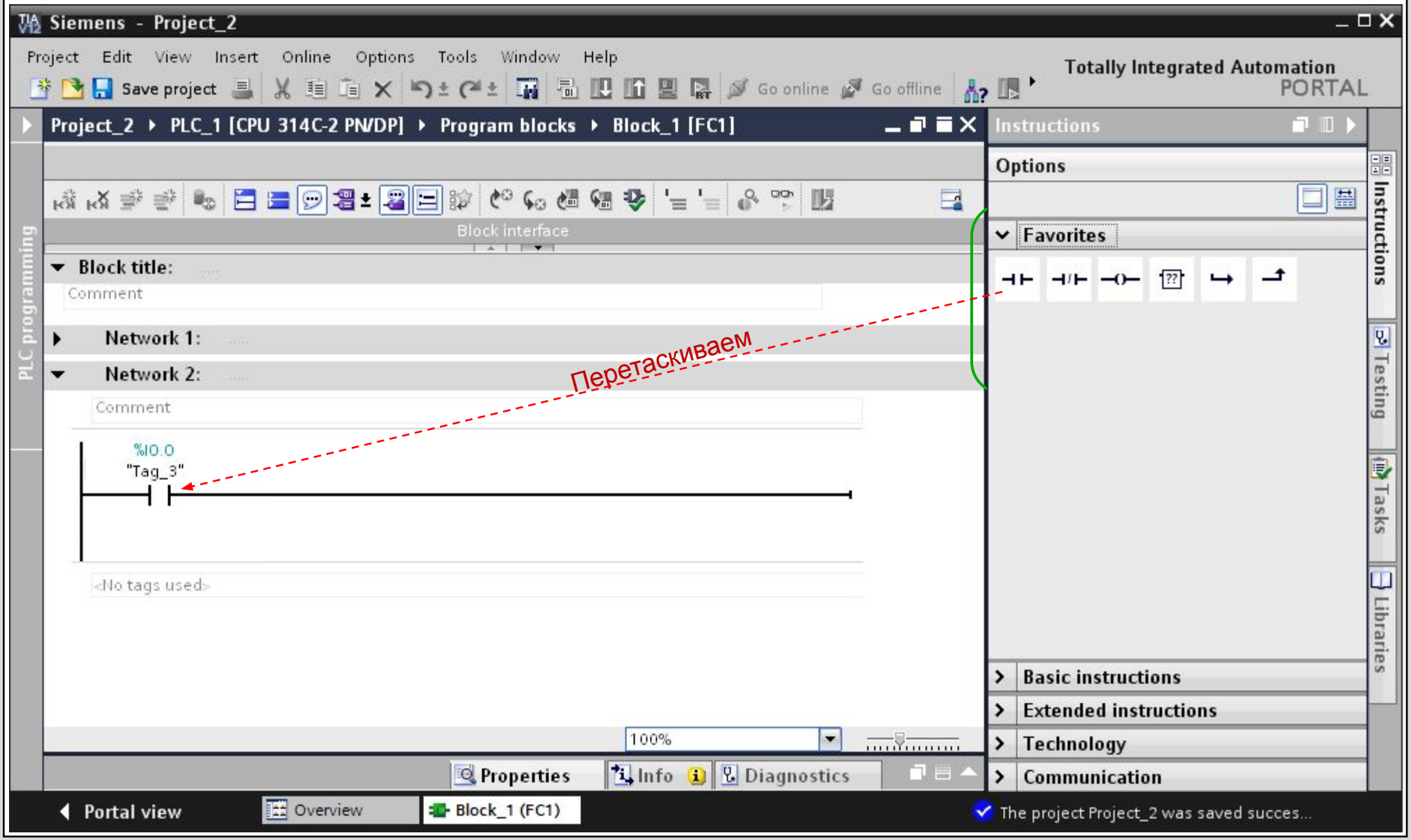
Слово состояния процессора



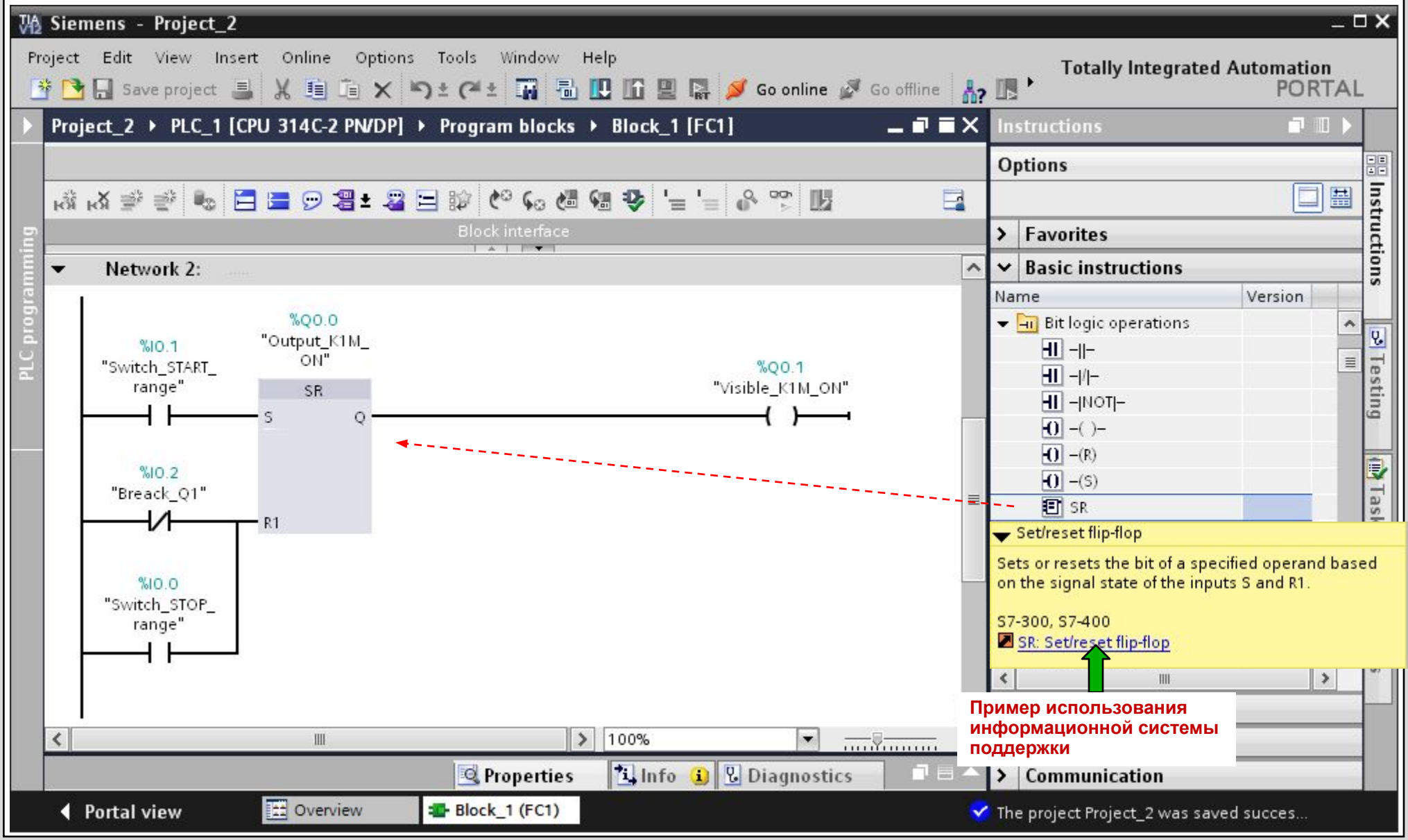
Бит	Двоичные флаги (binary flags)		
0	/FC	Первичный опрос (first check)	В начале цепи (сегмента LAD) производится проверка состояния этого бита
1	RLO	Результат логической операции	Служит для сохранения результата текущего логического сопряжения
2	STA	Состояние (статус - "status")	Бит статуса двоичной переменной
3	OR	Бит состояния OR (OR status bit)	Сохраняет результат логической операции «И» перед операцией «ИЛИ».
8	BR	Двоичный результат (binary result)	Служит для промежуточного хранения значения RLO
Бит	**Числовые флаги (digital flags)**		
4	OS	Для сохранения информации о переполнении (stored overflow)	Указывают на ошибки, возникшие в результате выполнения операций с числами (переполнение)
5	OV	Переполнение (overflow)	
6	CC0	Условный код (condition code)	Несут информацию о результатах выполнения операций с числами
7	CC1	Условный код (condition code)	

CC0	CC 1	Пояснение
0	0	результат операции =0
1	0	результат операции <0
0	1	результат операции >0

Программные инструкции. Способ «перетаскивания»



Ячейка памяти (SR)



Пример использования информационной системы поддержки

Описание ячейки памяти SR в информационной системе

Information System

Содержание | Указатель | Поиск | Избр

- Information system
- System overview of STEP 7 and Wi
- Readme
- Installation
- Migrating projects and programs
- First steps
- Introduction to the TIA Portal
- Editing projects
- Editing devices and networks
- Programming a PLC
  - Creating a user program
  - Displaying program information
  - Displaying cross-references
  - Testing the user program
  - Configuring alarms
  - Using global project functions
  - Programming examples
  - References
    - References (S7-1200, S7-15
    - References (S7-300, S7-400
      - General parameters of th
      - Basic instructions (S7-30
        - LAD (S7-300, S7-40
          - Bit logic operatic
            - ---| |---: Norr
            - ---| / |---: Nor
            - --|NOT|--: Inv
            - ---( )---: Ass

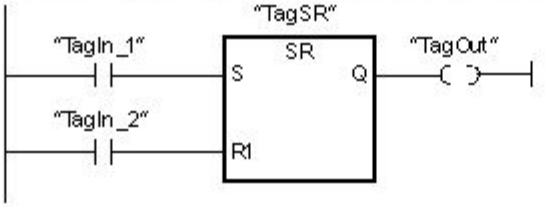
Examples | See also | History | Extra

SR: Set/reset flip-flop

<Operand>	InOut	BOOL	I, Q, M, D, L	Operand that is set or reset.
Q	Output	BOOL	I, Q, M, D, L	Signal state of the operand

Example

The following example shows how the instruction works:



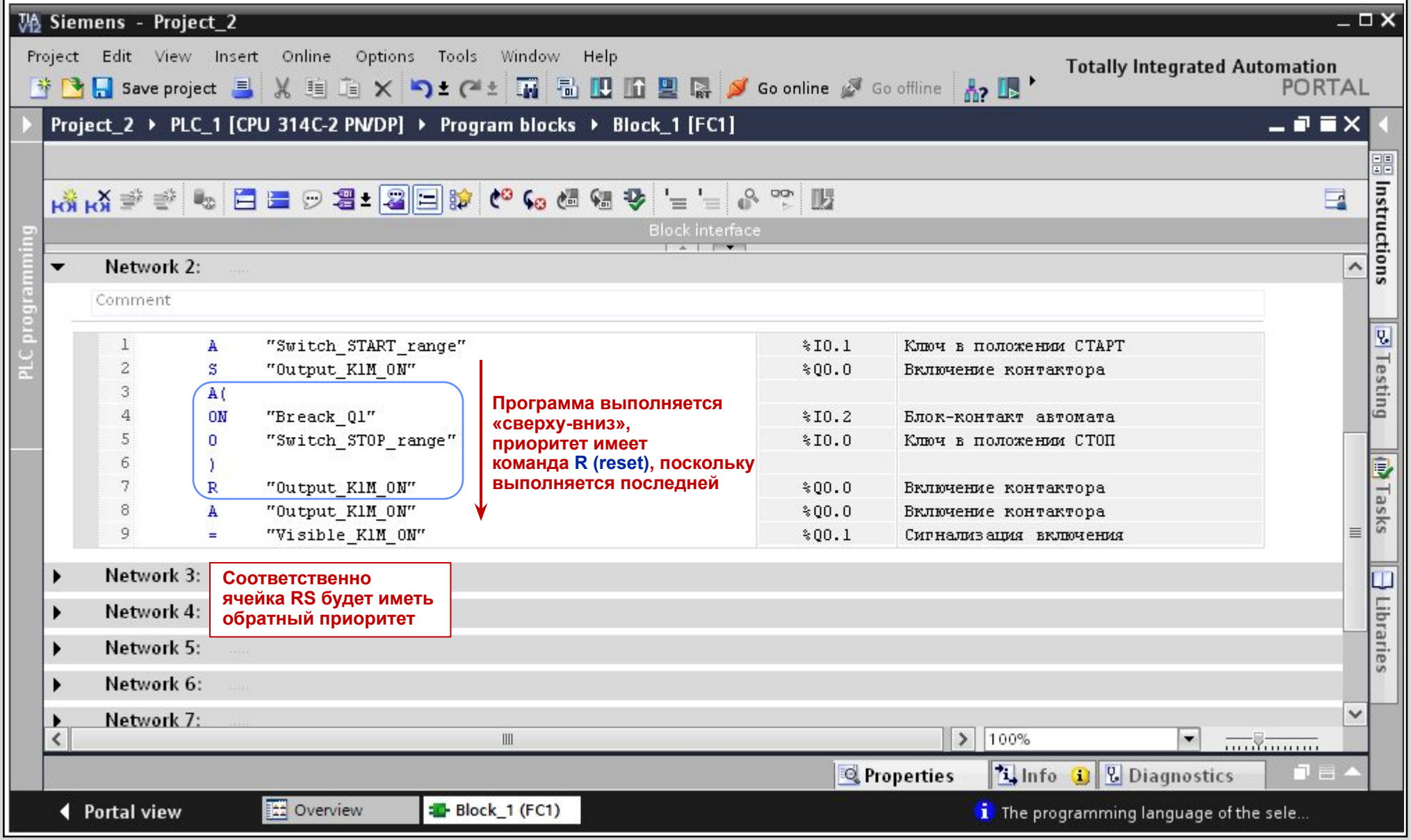
The operands "TagSR" and "TagOut" are set when the following conditions are fulfilled:

- The operand "TagIn_1" has the signal state "1".
- The operand "TagIn_2" has the signal state "0".

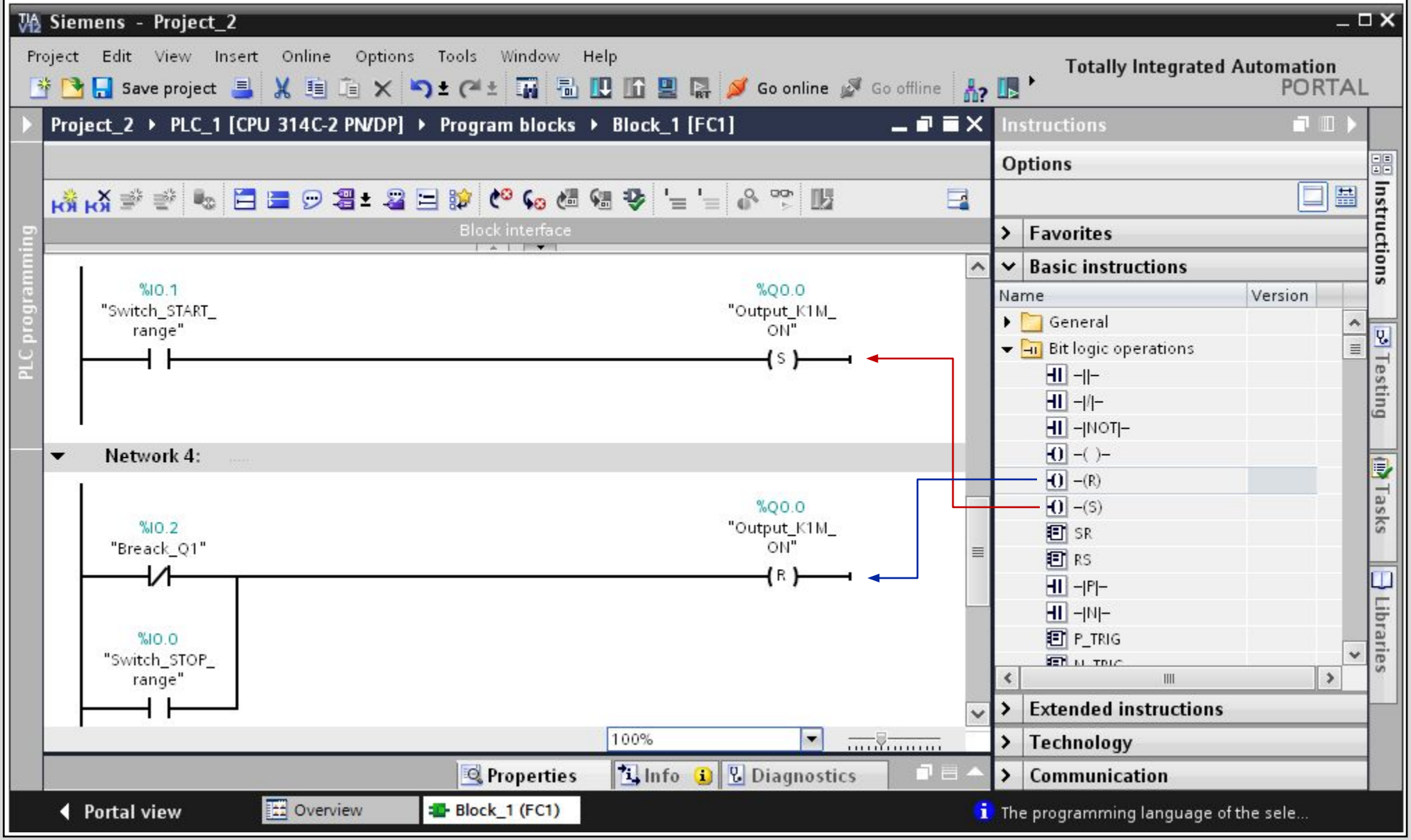
The operands "TagSR" and "TagOut" are reset when one of the following conditions is fulfilled:

- The operand "TagIn_1" has signal state "0" and the operand "TagIn_2" has signal state "1".
- The operands "TagIn_1" and "TagIn_2" have signal state "1".

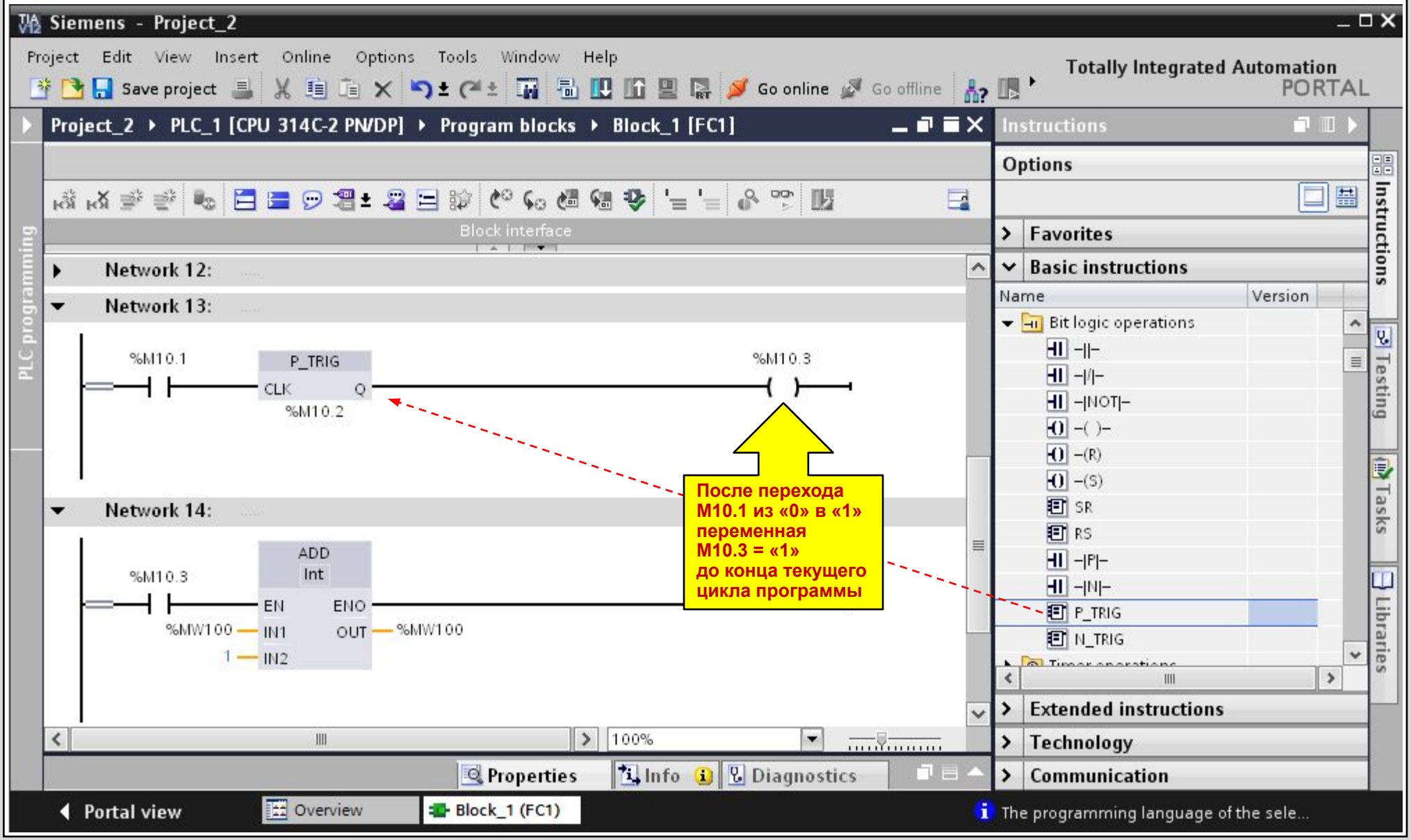
Приоритеты команд в ячейках памяти SR/RS



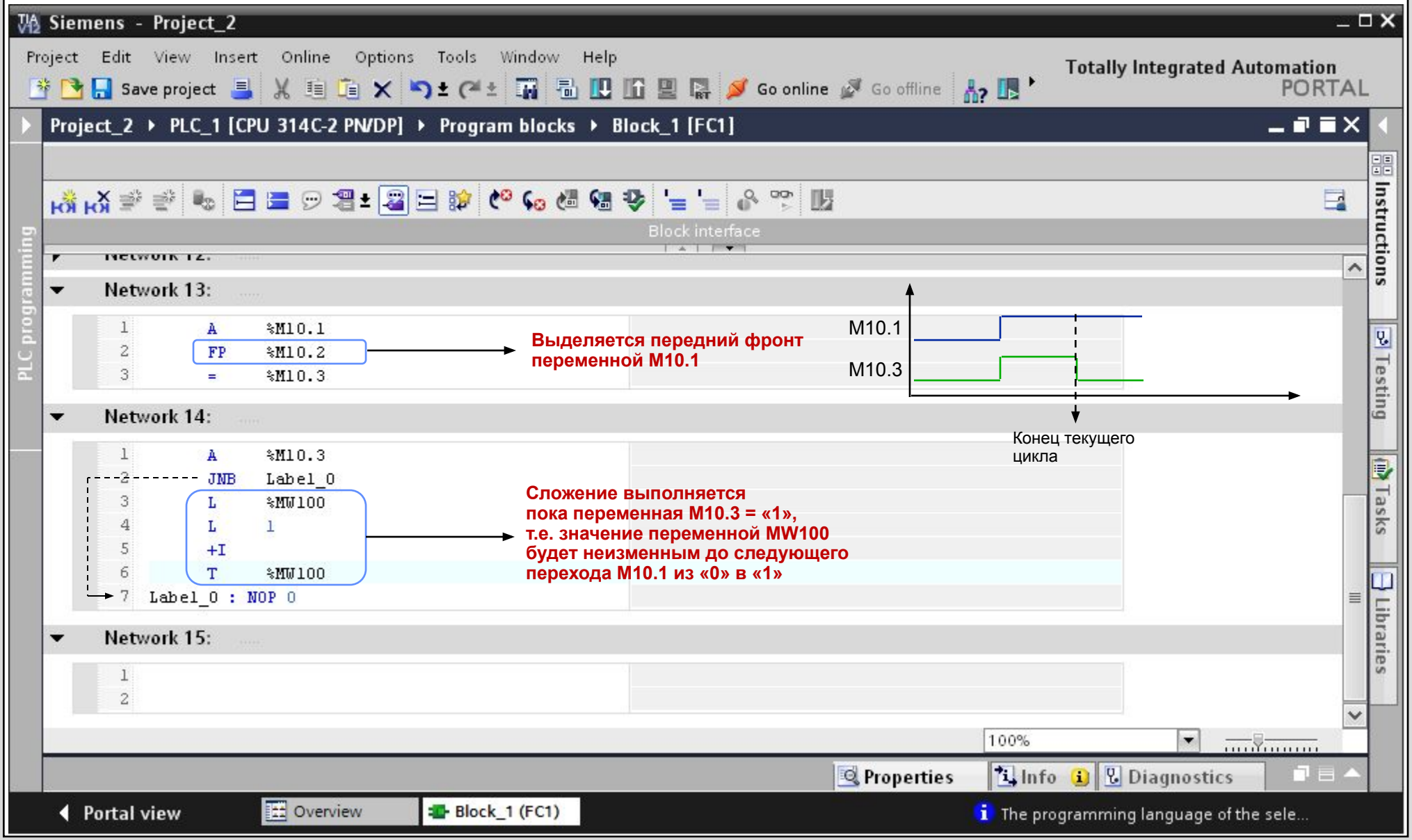
Использование отдельных команд S и R



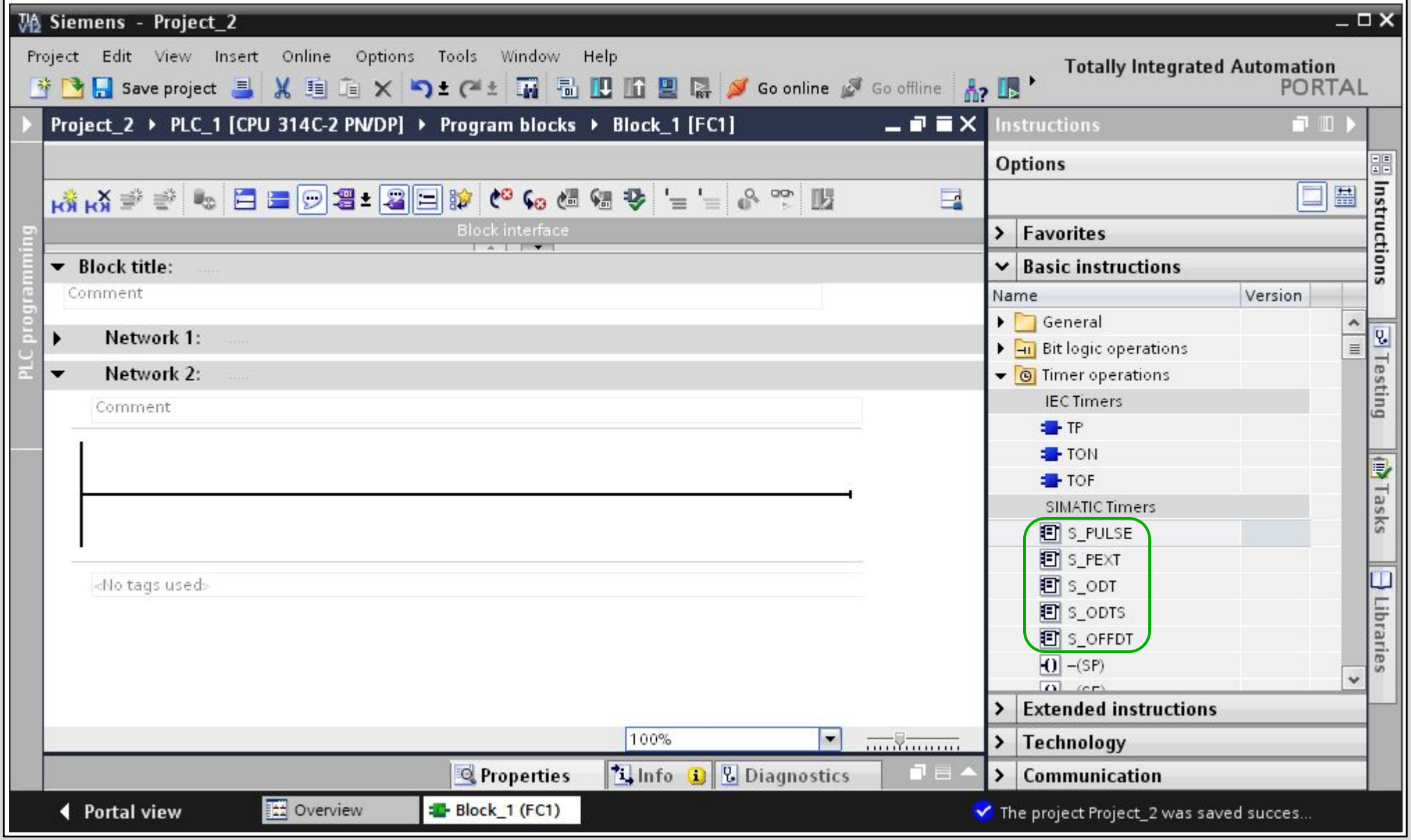
Команда выделения переднего фронта (P_TRIG)



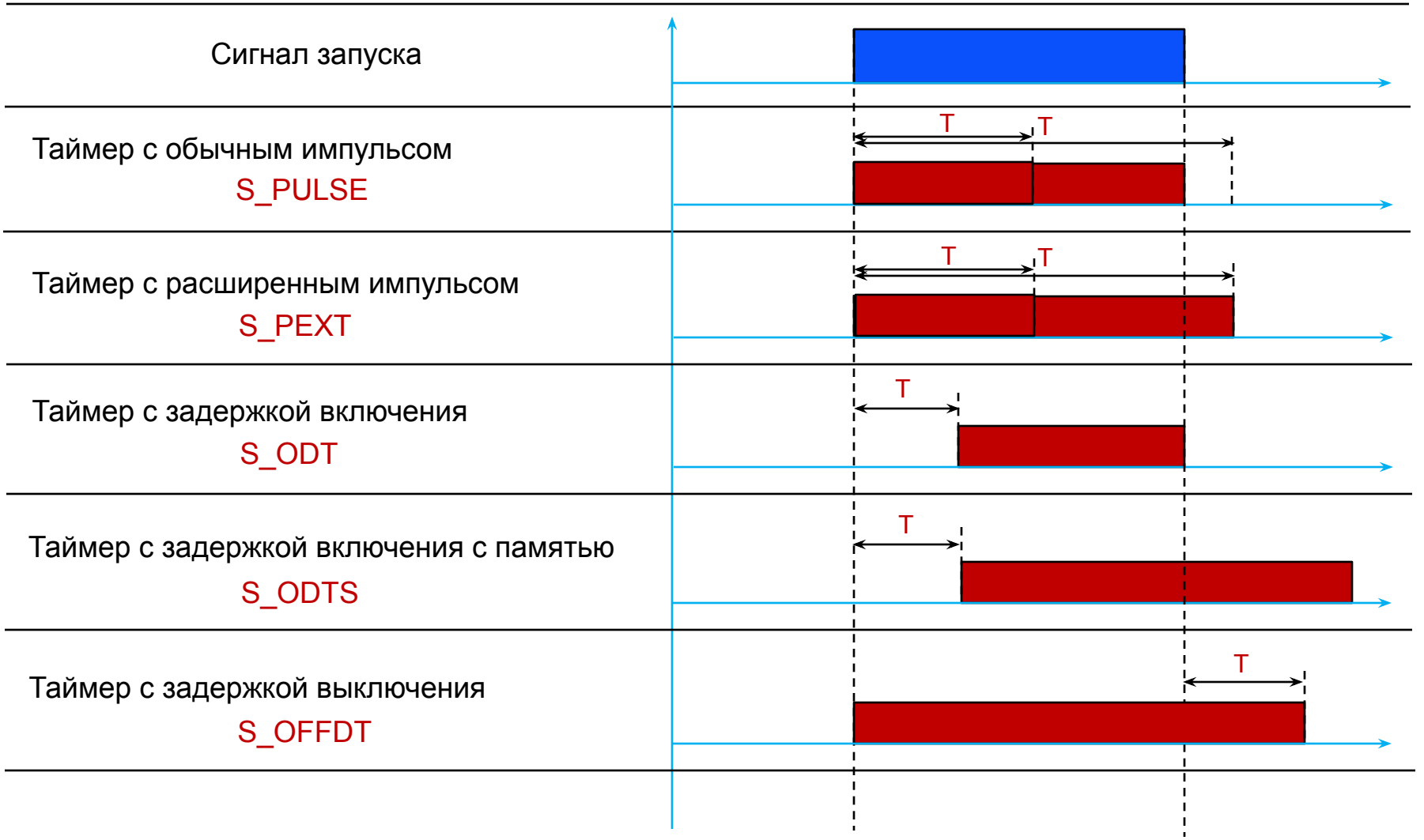
Команда выделения переднего фронта (P_TRIG)



Инструкции SIMATIC Timers



Диаграммы работы таймеров SIMATIC



Двоично-десятичный код (BCD)

Диапазон значений **16 бит:** **от -999 до + 999**
32 бит: **от -9999999 до + 9999999**

Инструкции преобразования: BTI, BTD, ITB, DTB (не арифметические, поскольку BCD системой счисления не является!)

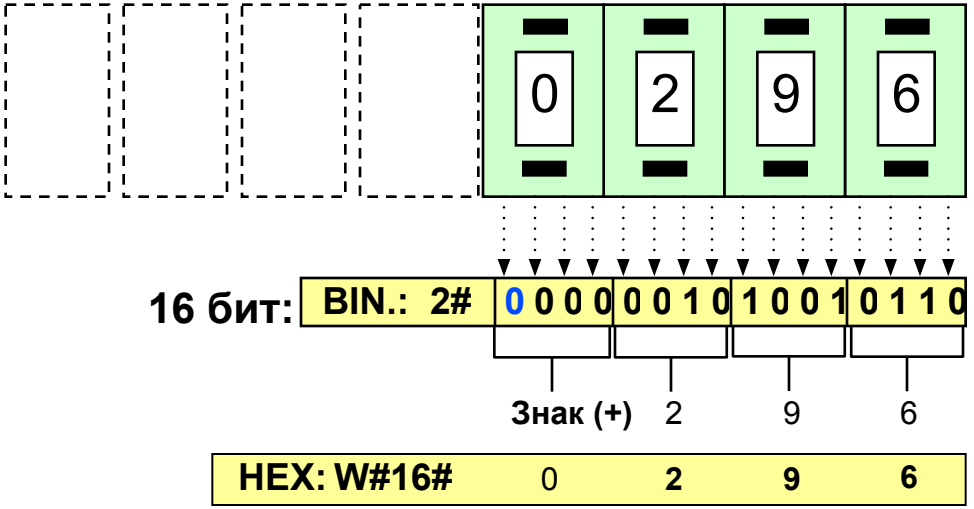
Каждый знак десятичного числа от 0 до 9 представлен в виде четырехбитового двоичного числа

0 ⟶ 0000 $0x\ 2^3 + 0x\ 2^2 + 0x\ 2^1 + 0x\ 2^0$

1 ⟶ 0001 $0x\ 2^3 + 0x\ 2^2 + 0x\ 2^1 + 1x\ 2^0$

---

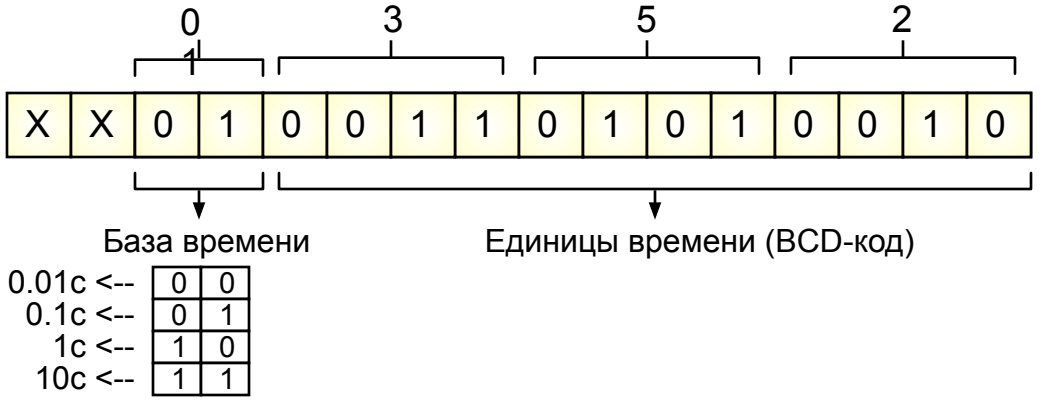
9 ⟶ 1001 $1x\ 2^3 + 0x\ 2^2 + 0x\ 2^1 + 1x\ 2^0$



Форматы записи/чтения времени для таймеров SIMATIC

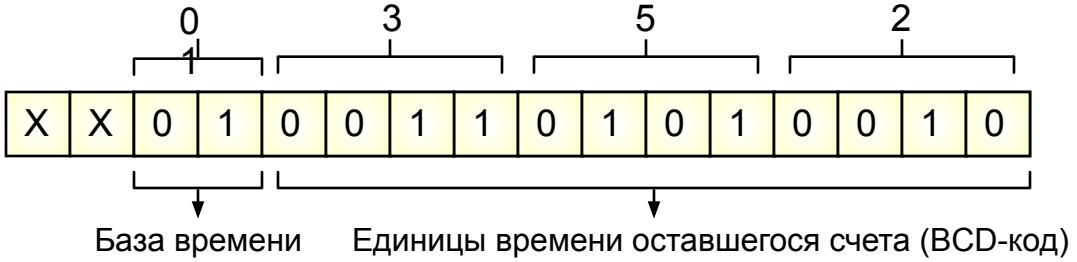
Предположим, уставка таймера составляет 35 секунд 200 миллисекунд (формат записи S5T#35s200ms)

Запись времени в слово таймера



Чтение оставшегося времени в Integer

Чтение оставшегося времени в BCD - коде



Пример программирования таймера «задержка включения»

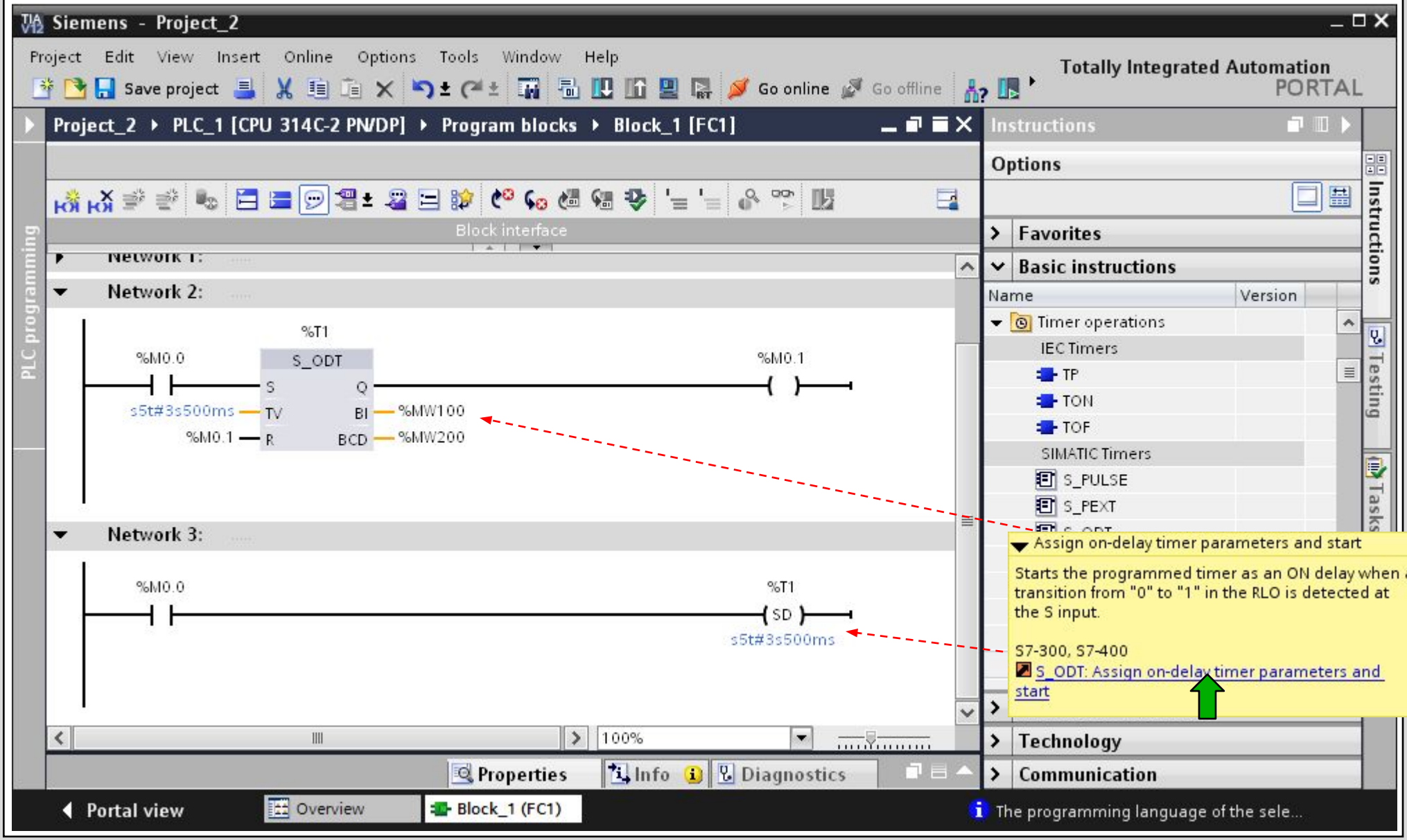
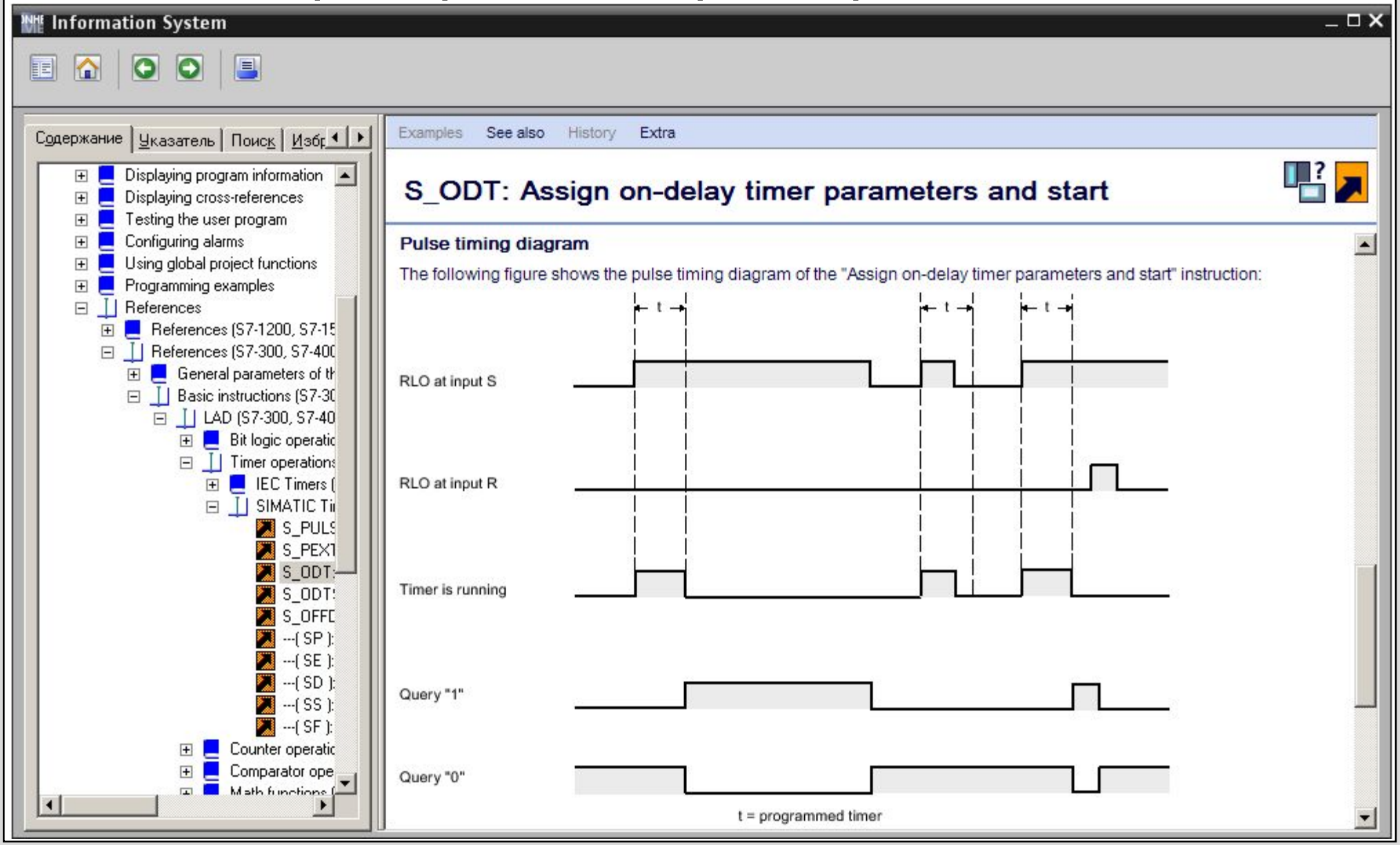
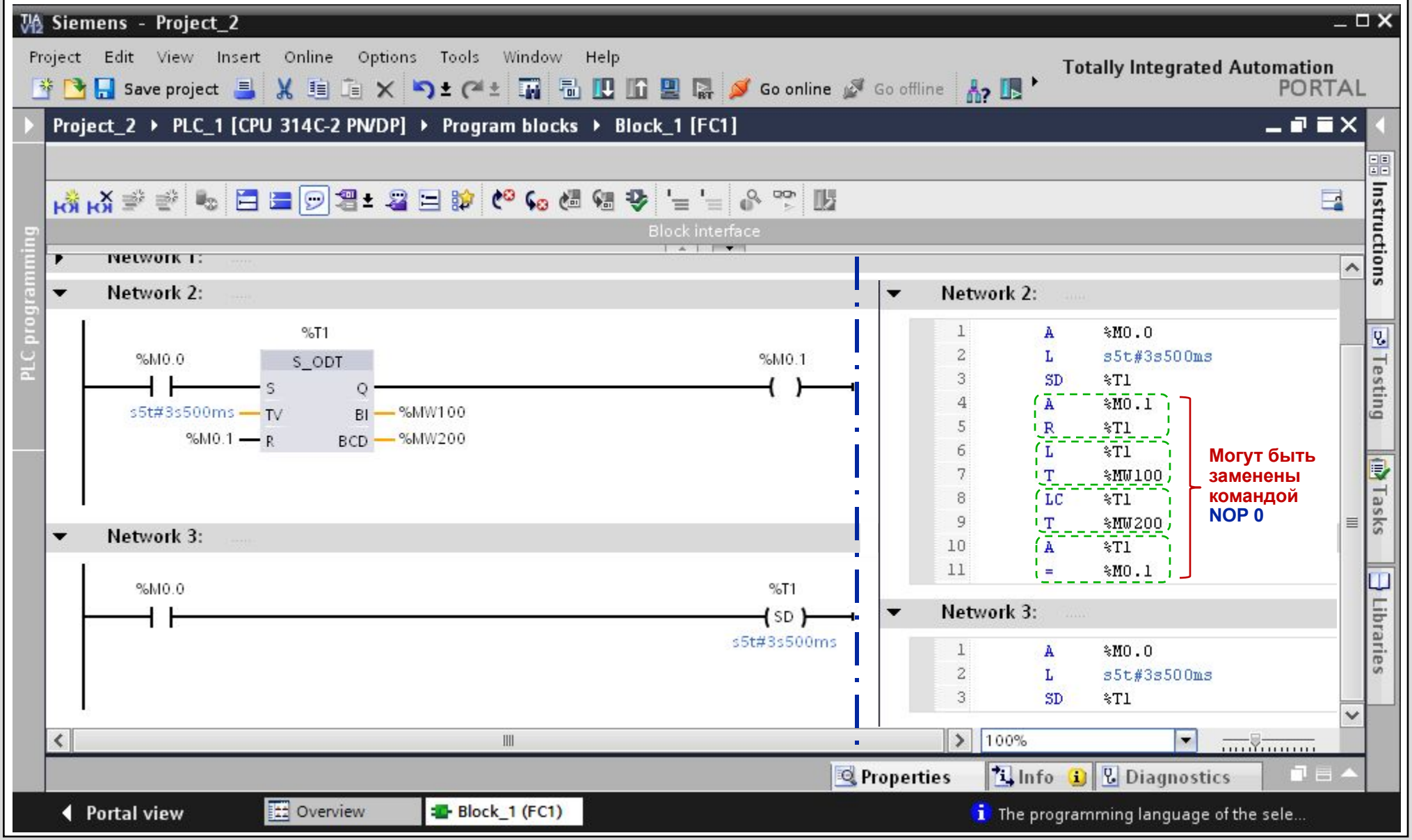


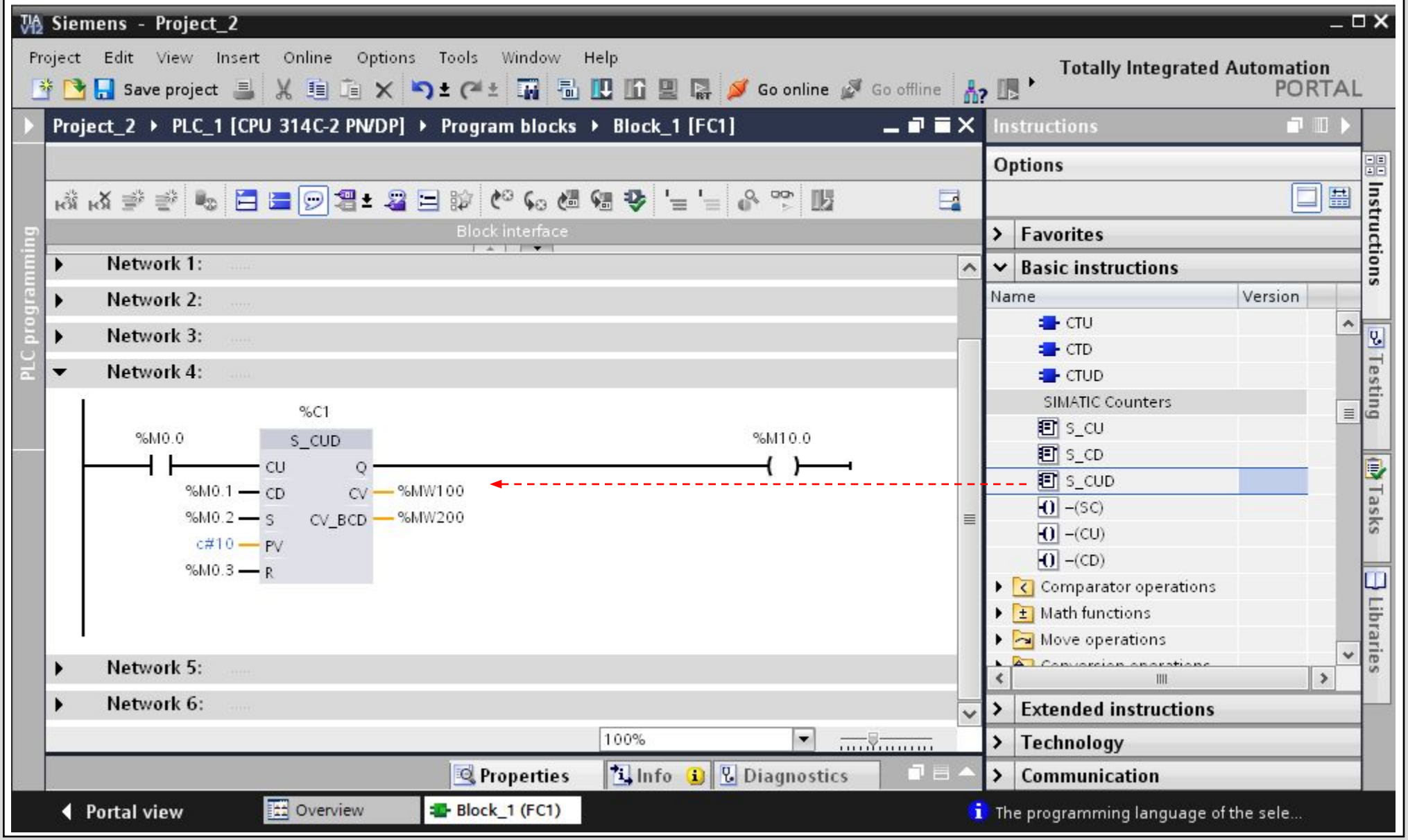
Диаграмма работы таймера «задержка включения»



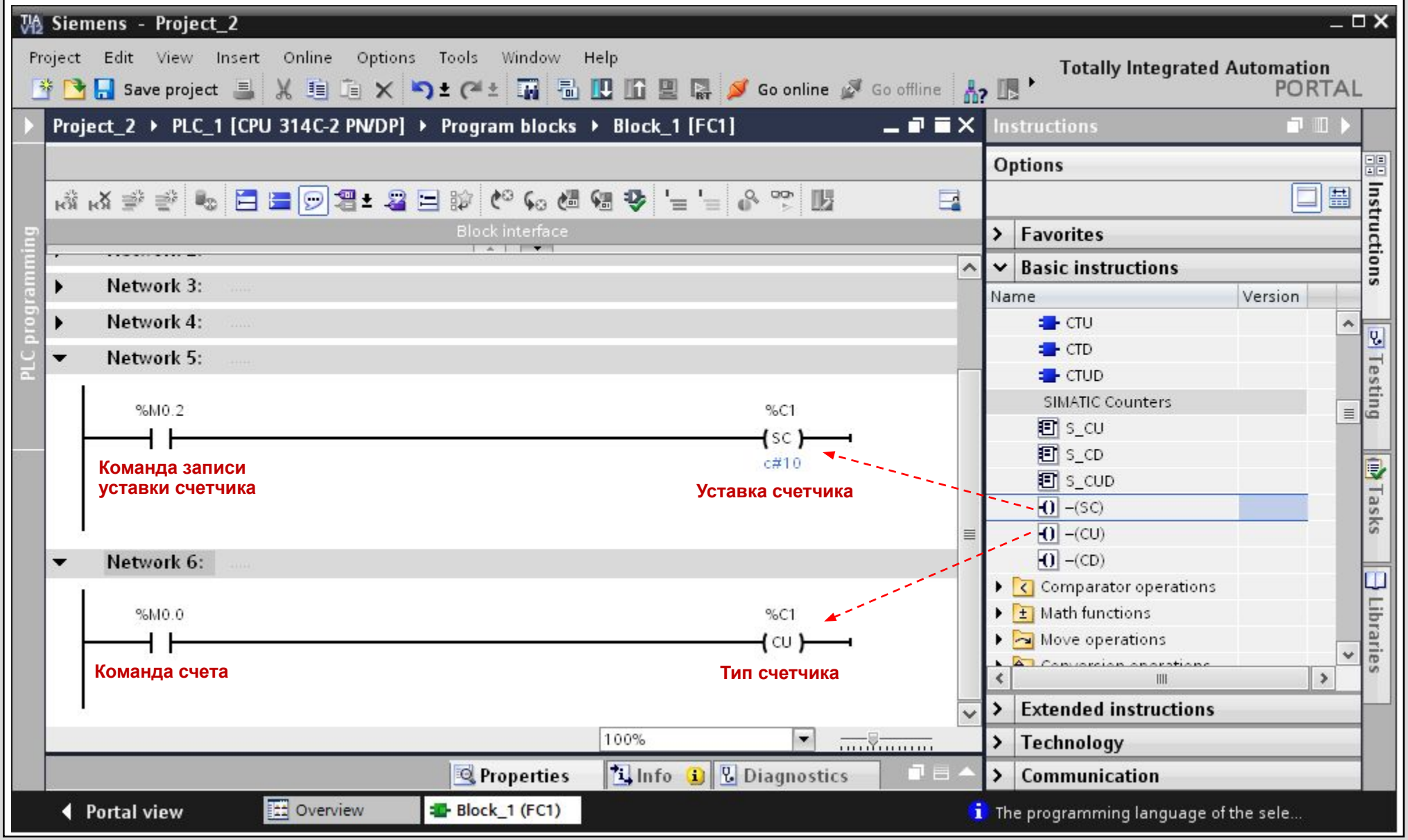
Сравнение форм представления таймера



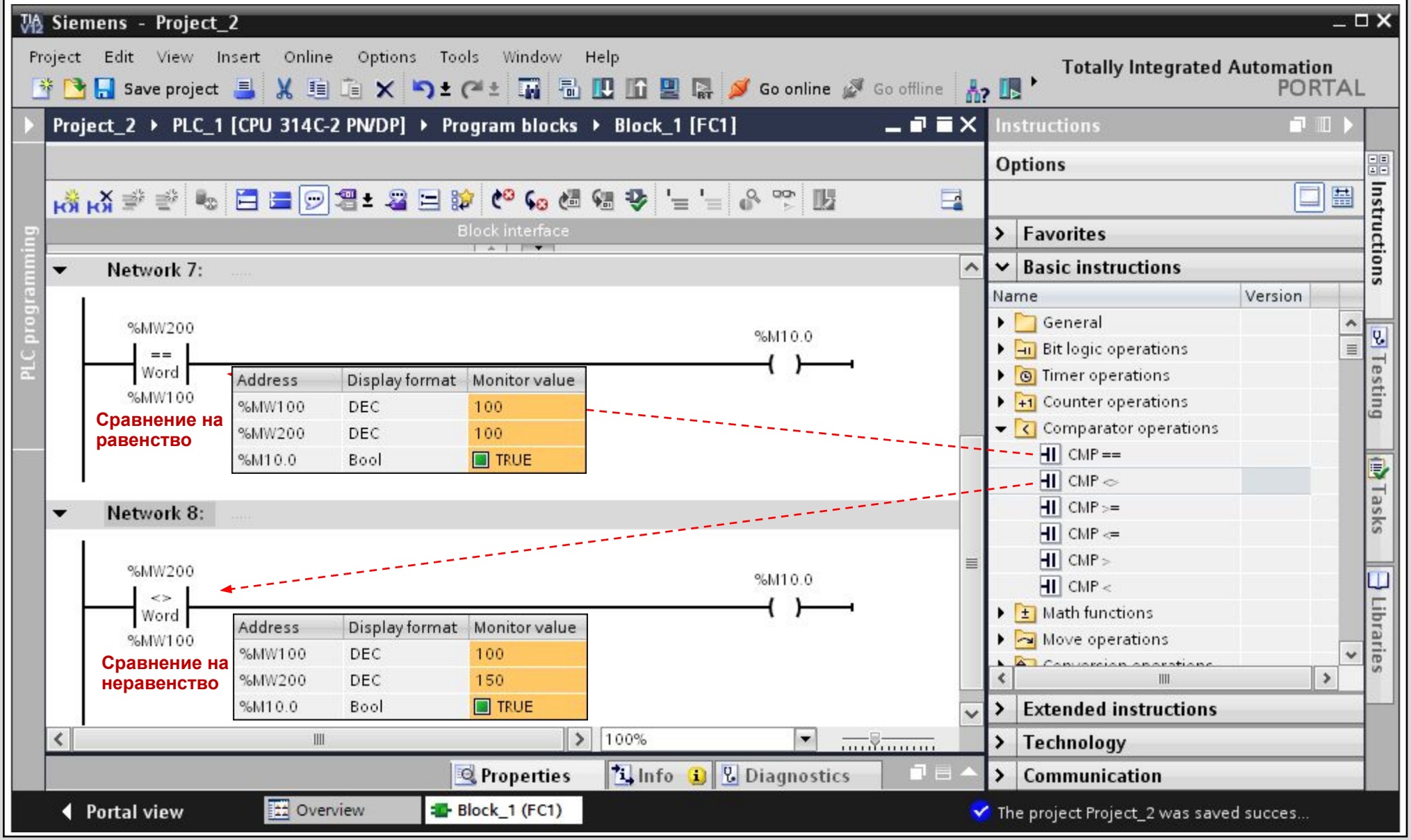
Пример программирования реверсивного счетчика



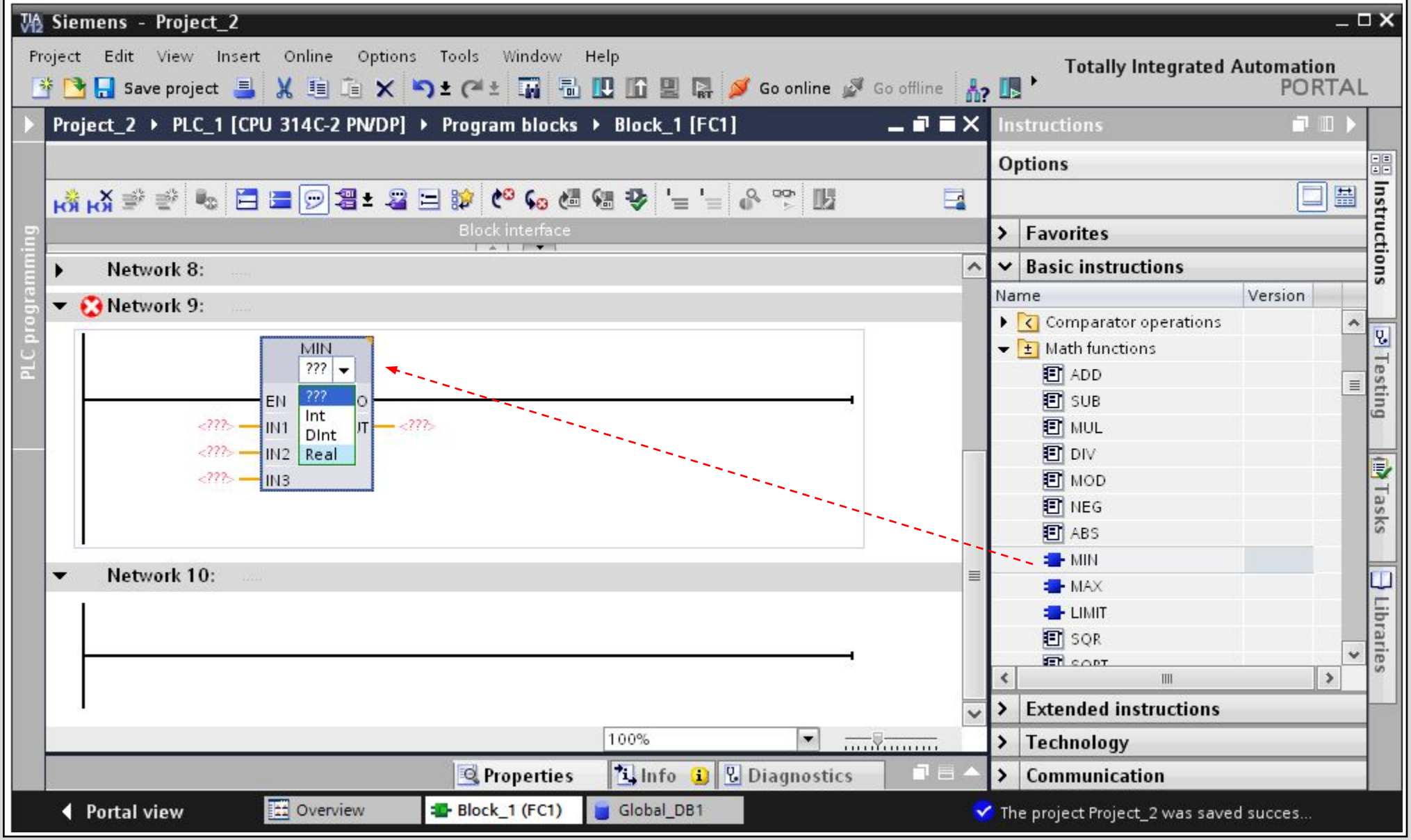
Пример программирования счетчика отдельными командами



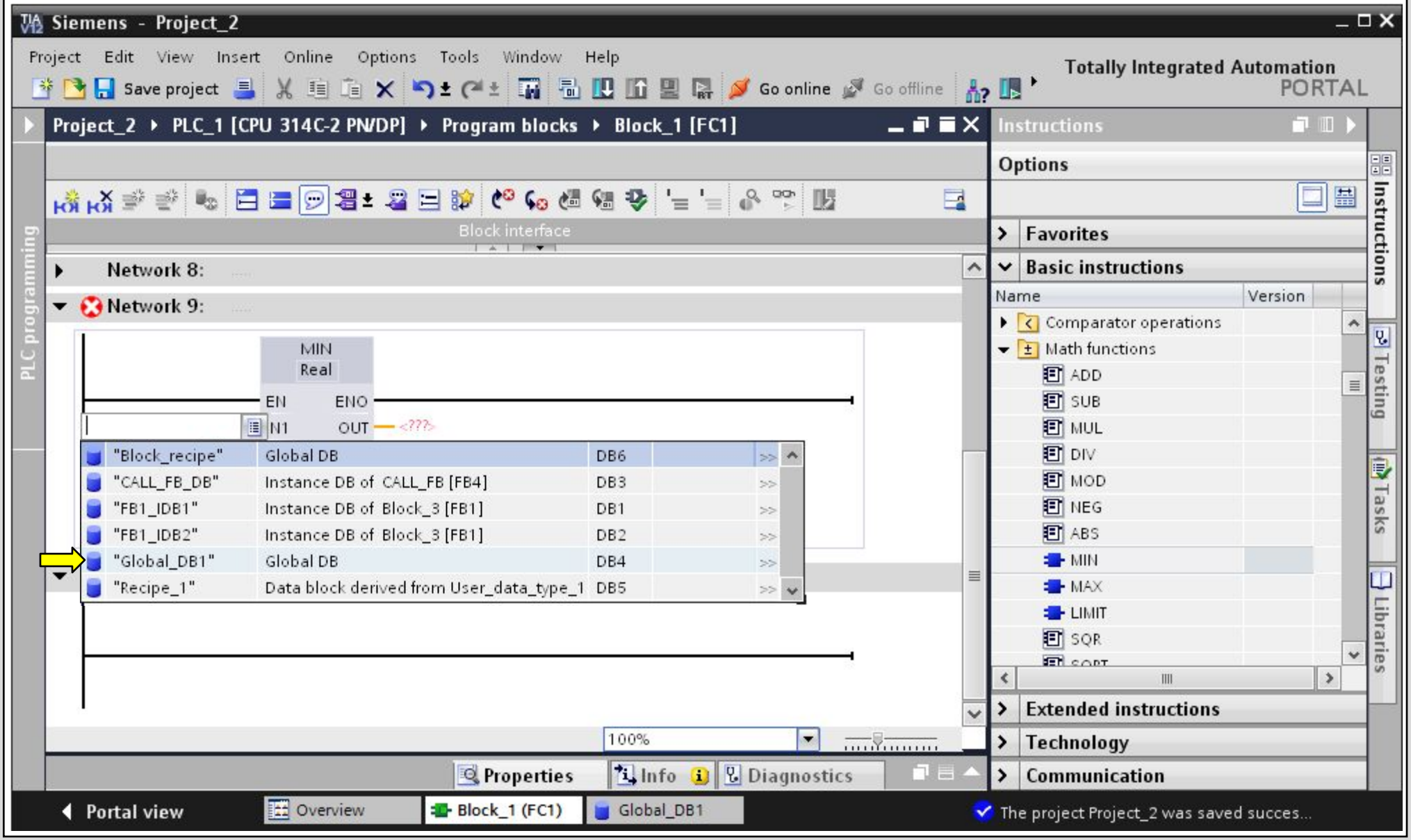
Пример программирования компараторов



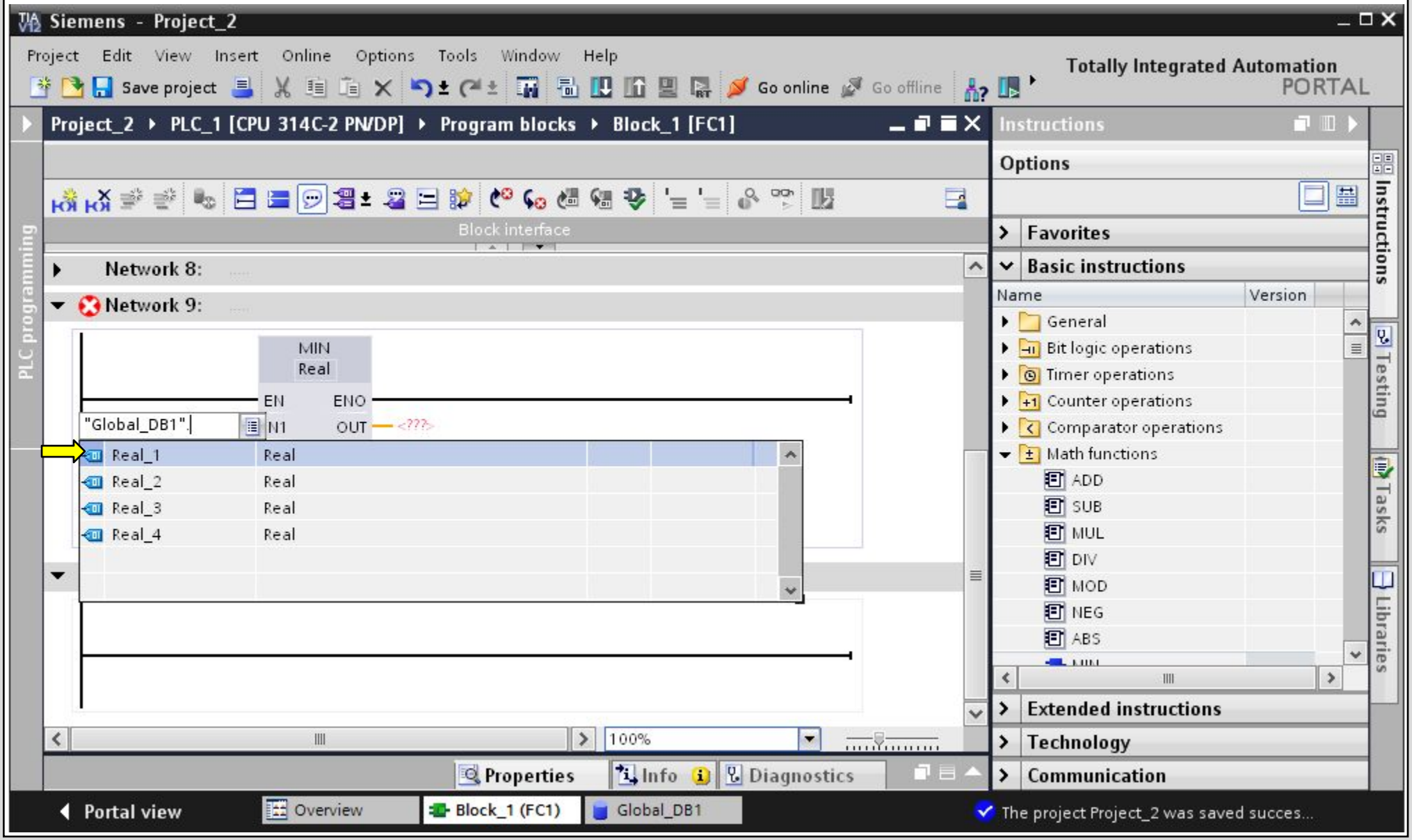
Пример применения функции MIN



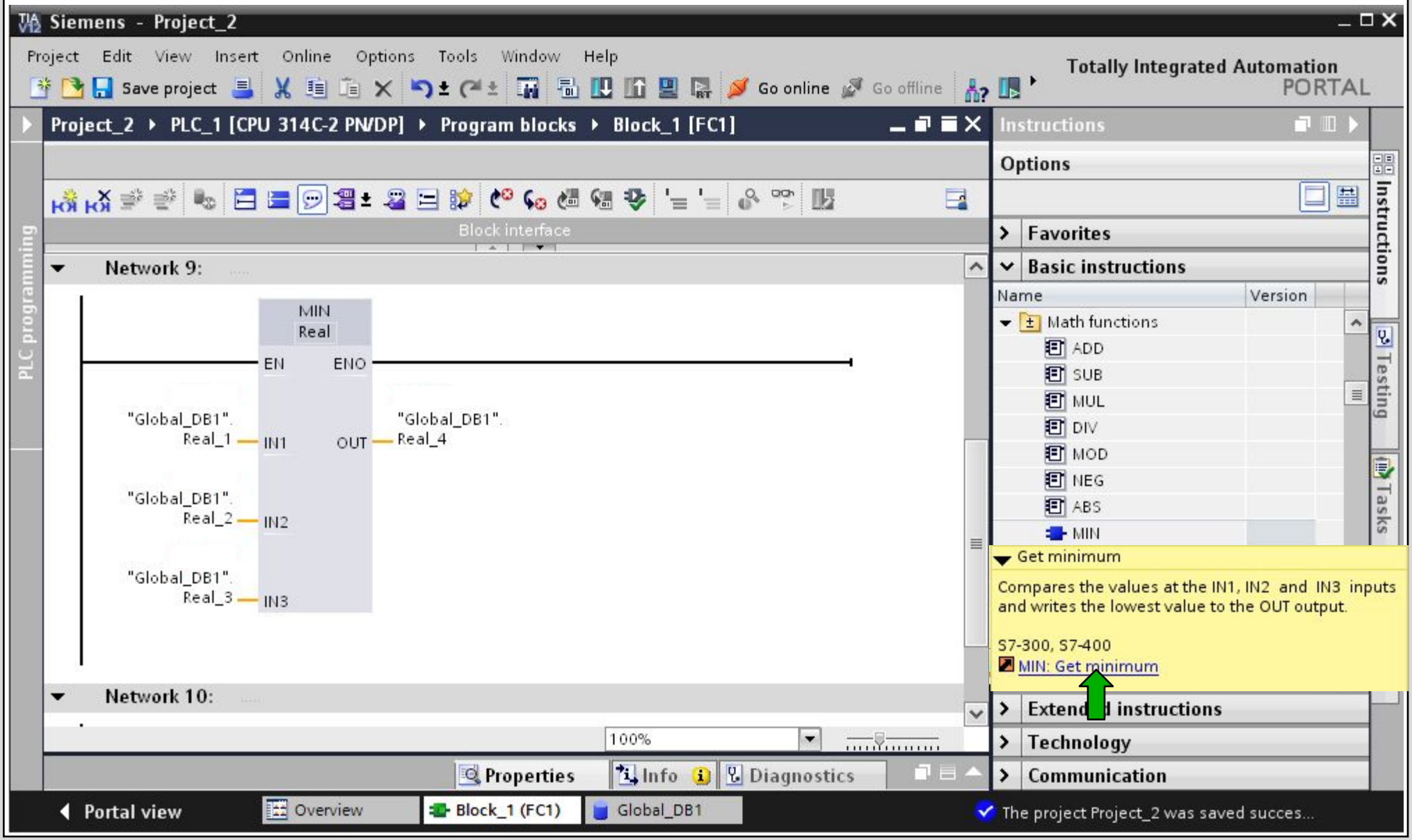
Пример применения функции MIN



Пример применения функции MIN



Пример применения функции MIN



Описание функции MIN в информационной системе

Information System

Содержание | Указатель | Поиск | Избр

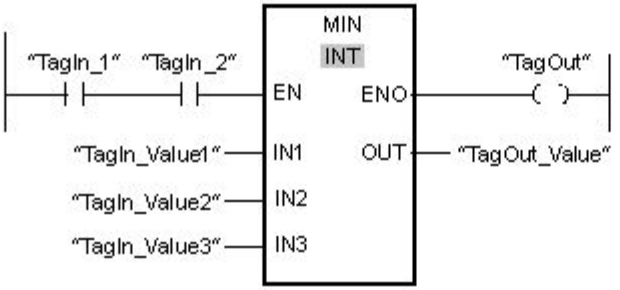
- Introduction to the TIA Portal
- Editing projects
- Editing devices and networks
- Programming a PLC
  - Creating a user program
  - Displaying program information
  - Displaying cross-references
  - Testing the user program
  - Configuring alarms
  - Using global project functions
  - Programming examples
  - References
    - References (S7-1200, S7-15
    - References (S7-300, S7-400
      - General parameters of th
      - Basic instructions (S7-30
        - LAD (S7-300, S7-40
          - Bit logic operatic
          - Timer operations
          - Counter operatic
          - Comparator ope
          - Math functions (
            - ADD: Add (S
            - SUB: Subtra
            - MUL: Multip
            - DIV: Divide
            - MOD: Retur
            - NEG: Create

Examples | See also | History | Extra

MIN: Get minimum

Example

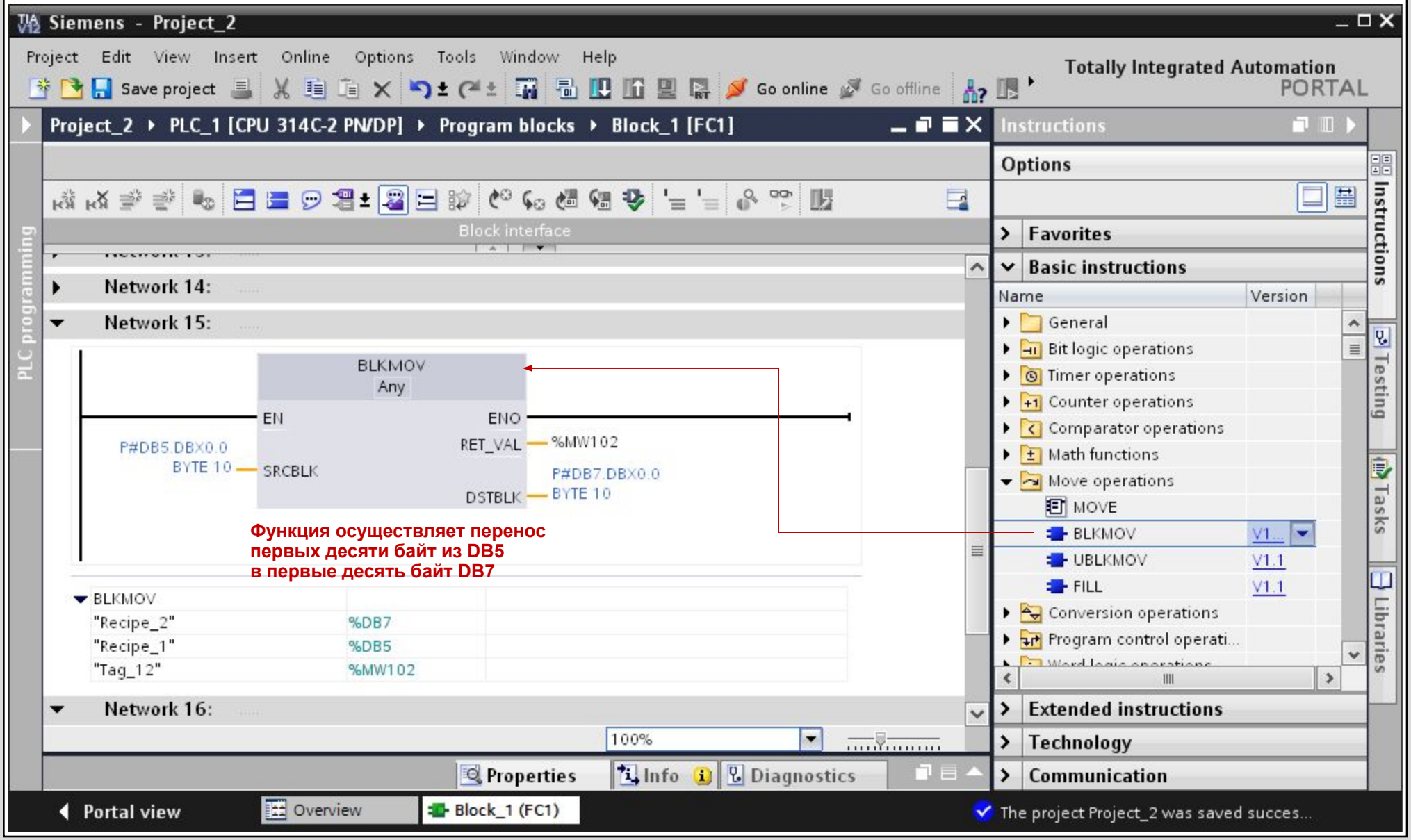
The following example shows how the instruction works:



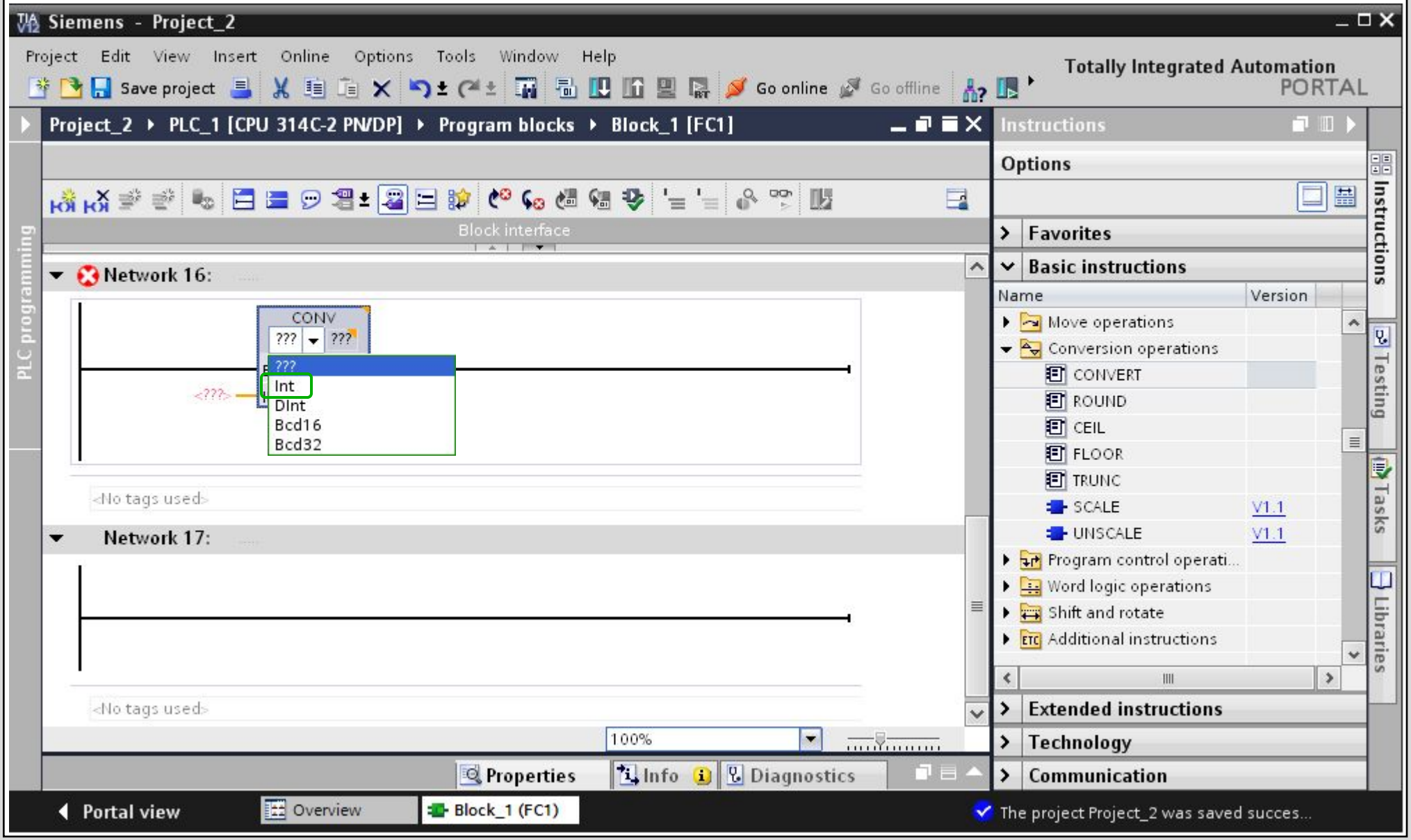
The following table shows how the instruction works using specific operand values:

Parameter	Operand	Value
IN1	TagIn_Value1	12222
IN2	TagIn_Value2	14444
IN3	TagIn_Value3	13333
OUT	TagOut_Value	12222

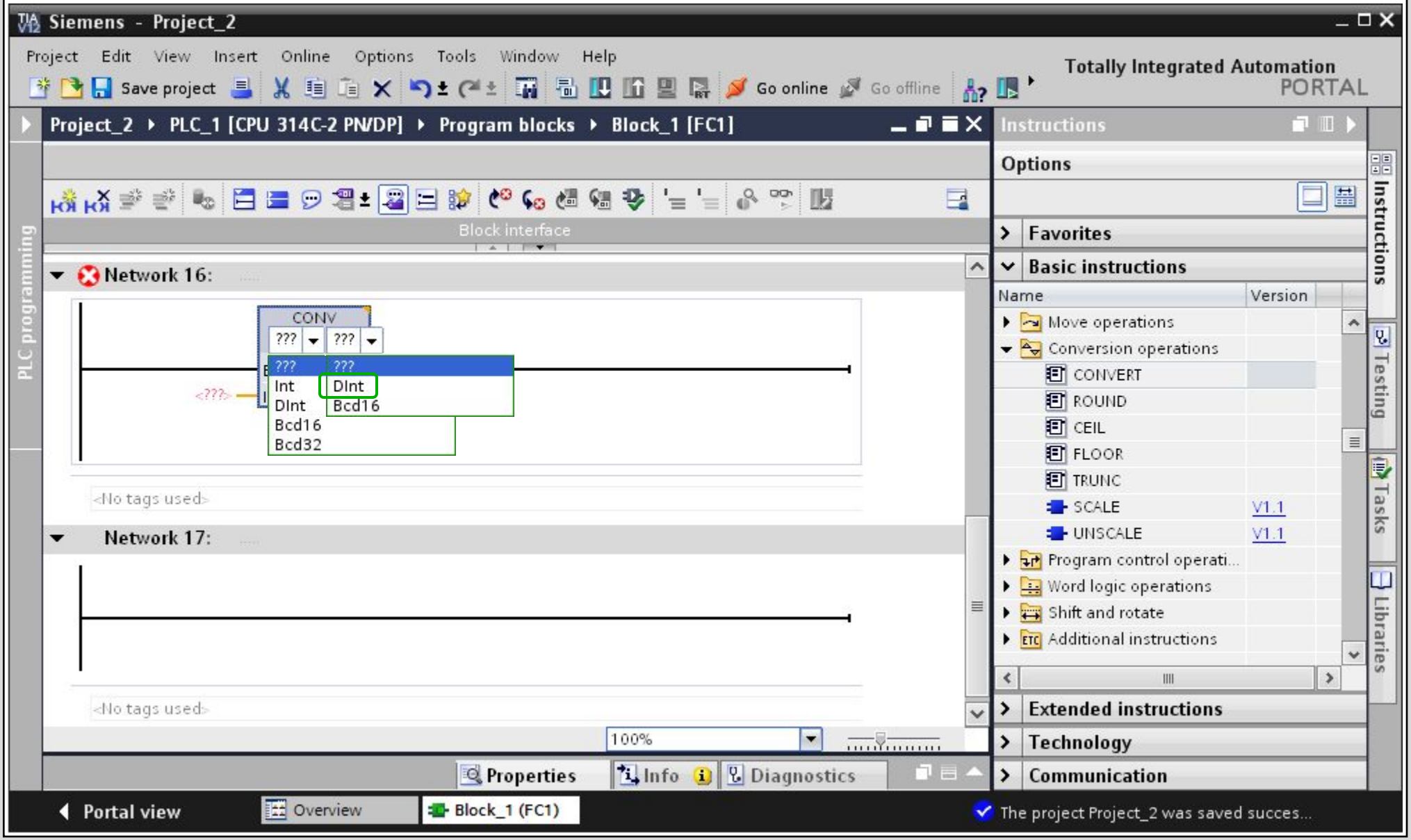
Пример применения функции BLKMOV



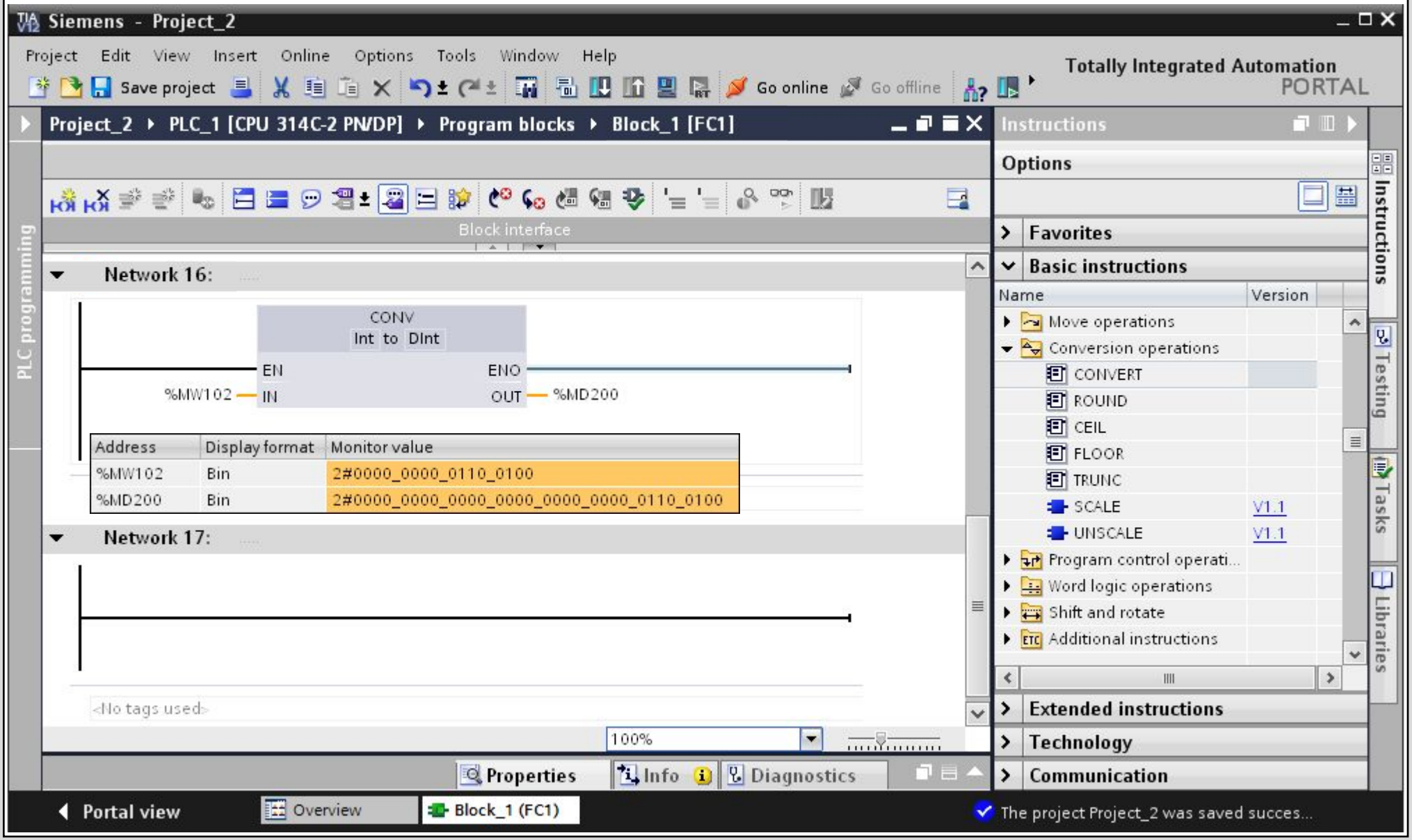
Пример применения функции CONVERT



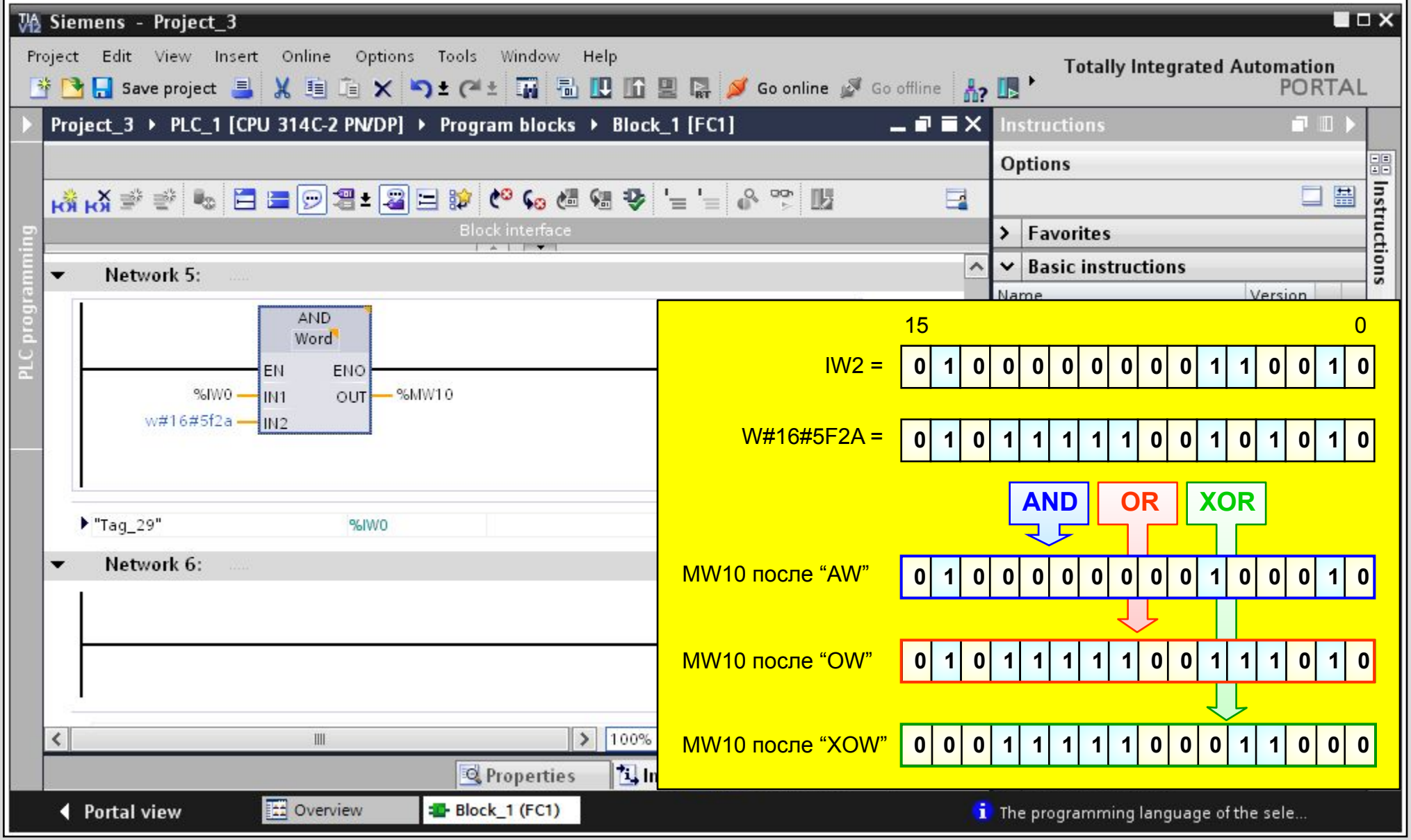
Пример применения функции CONVERT



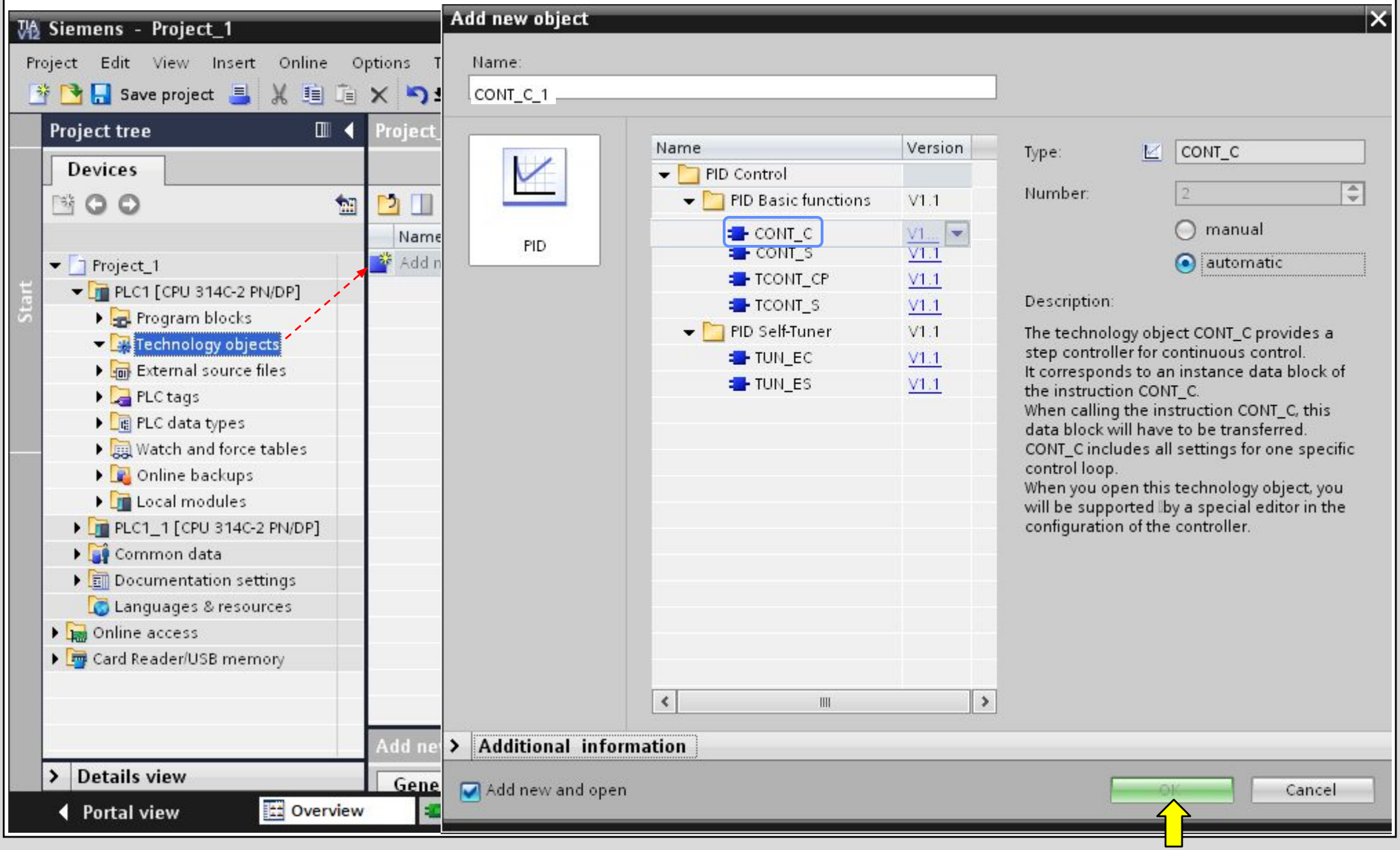
Пример применения функции CONVERT



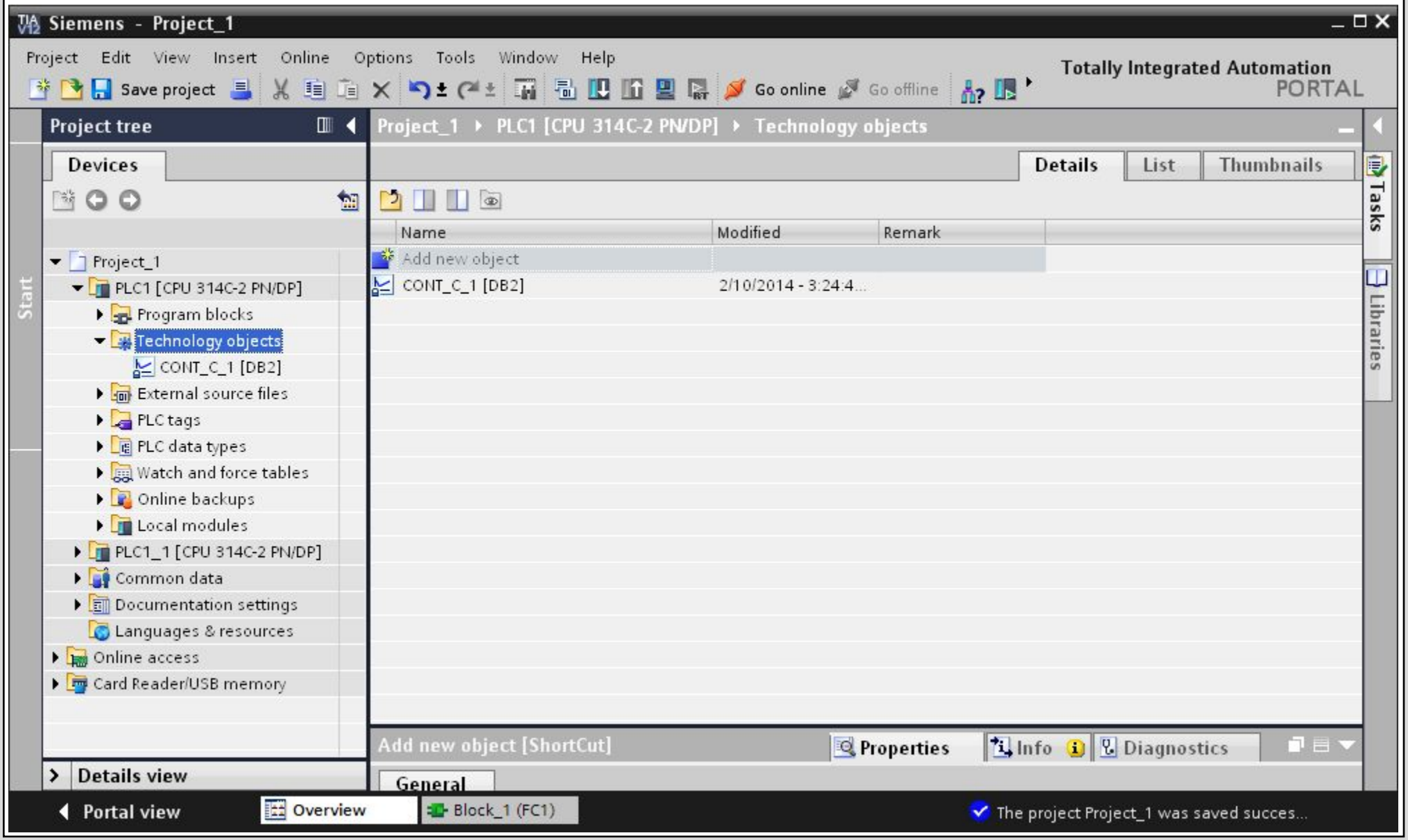
Пример применения функций побитной логики в словах



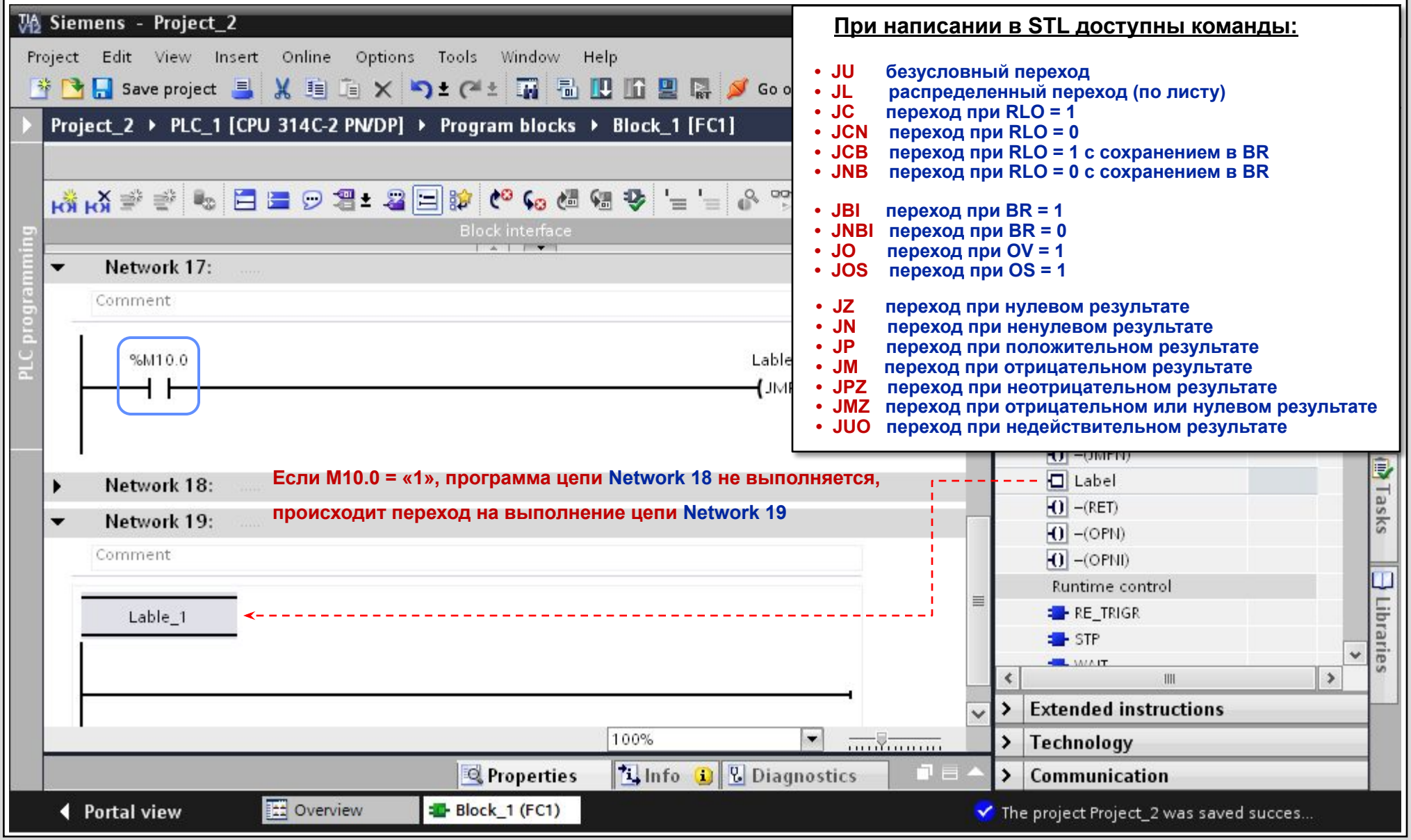
Выбор ПИД-регулятора в папке “Technology objects”.



Отображение выбранного ПИД-регулятора в “Technology objects”.



Переходы. Пример построения перехода на метку Lable_1



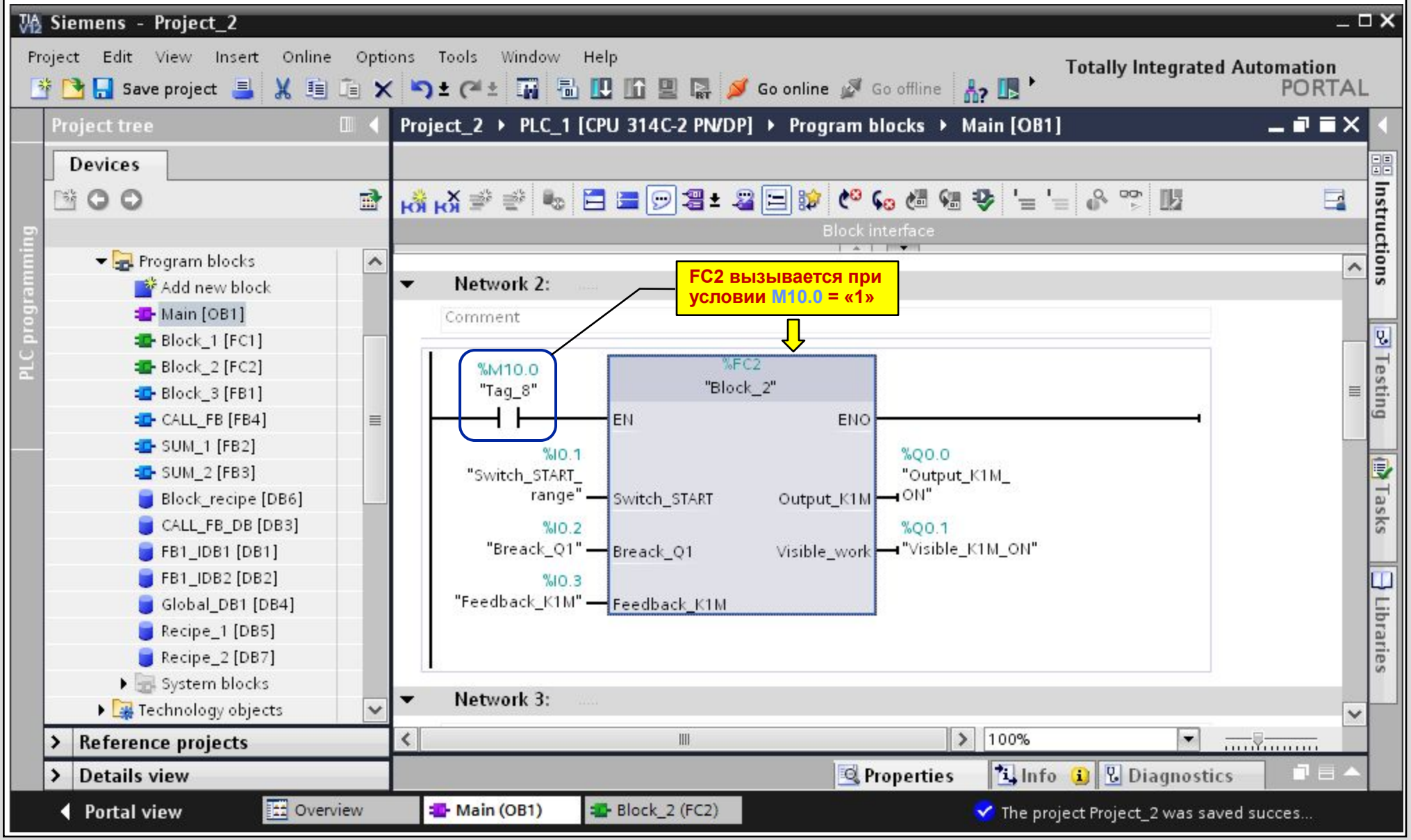
При написании в STL доступны команды:

- **JU** безусловный переход
- **JL** распределенный переход (по листу)
- **JC** переход при RLO = 1
- **JCN** переход при RLO = 0
- **JCB** переход при RLO = 1 с сохранением в BR
- **JNB** переход при RLO = 0 с сохранением в BR

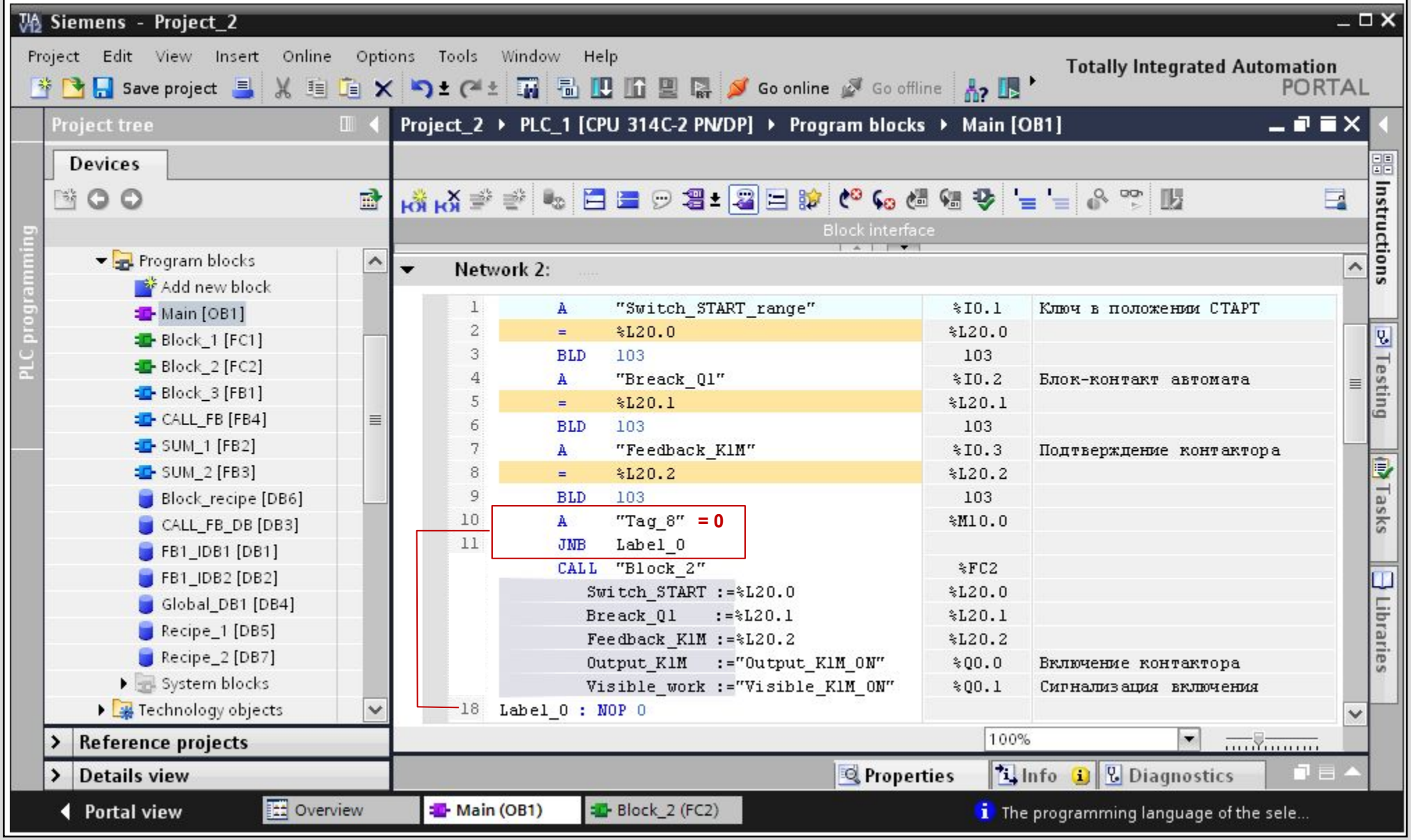
- **JBI** переход при BR = 1
- **JNBI** переход при BR = 0
- **JO** переход при OV = 1
- **JOS** переход при OS = 1

- **JZ** переход при нулевом результате
- **JN** переход при ненулевом результате
- **JP** переход при положительном результате
- **JM** переход при отрицательном результате
- **JPZ** переход при неотрицательном результате
- **JMZ** переход при отрицательном или нулевом результате
- **JUO** переход при недействительном результате

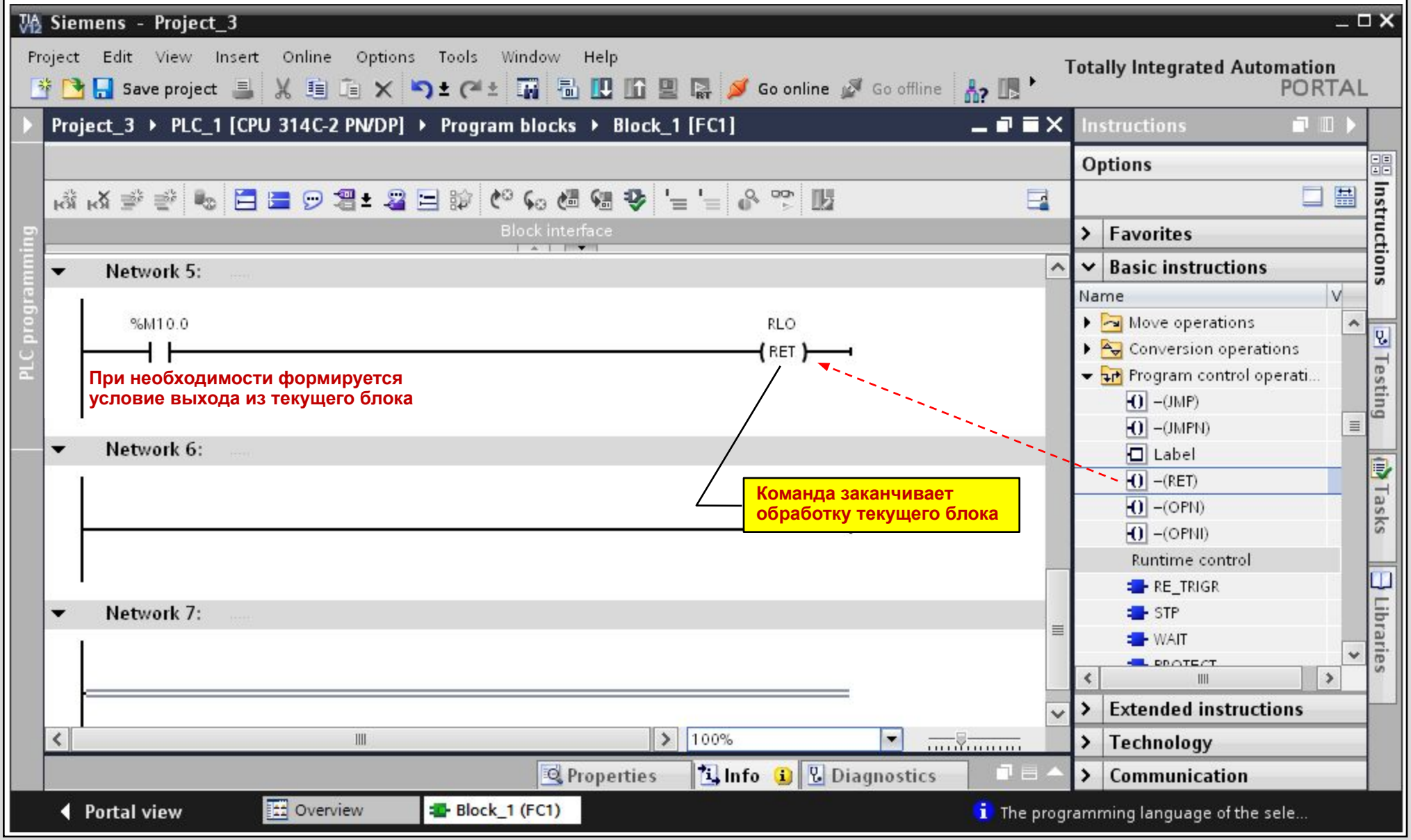
Пример построения перехода при вызове FC по условию



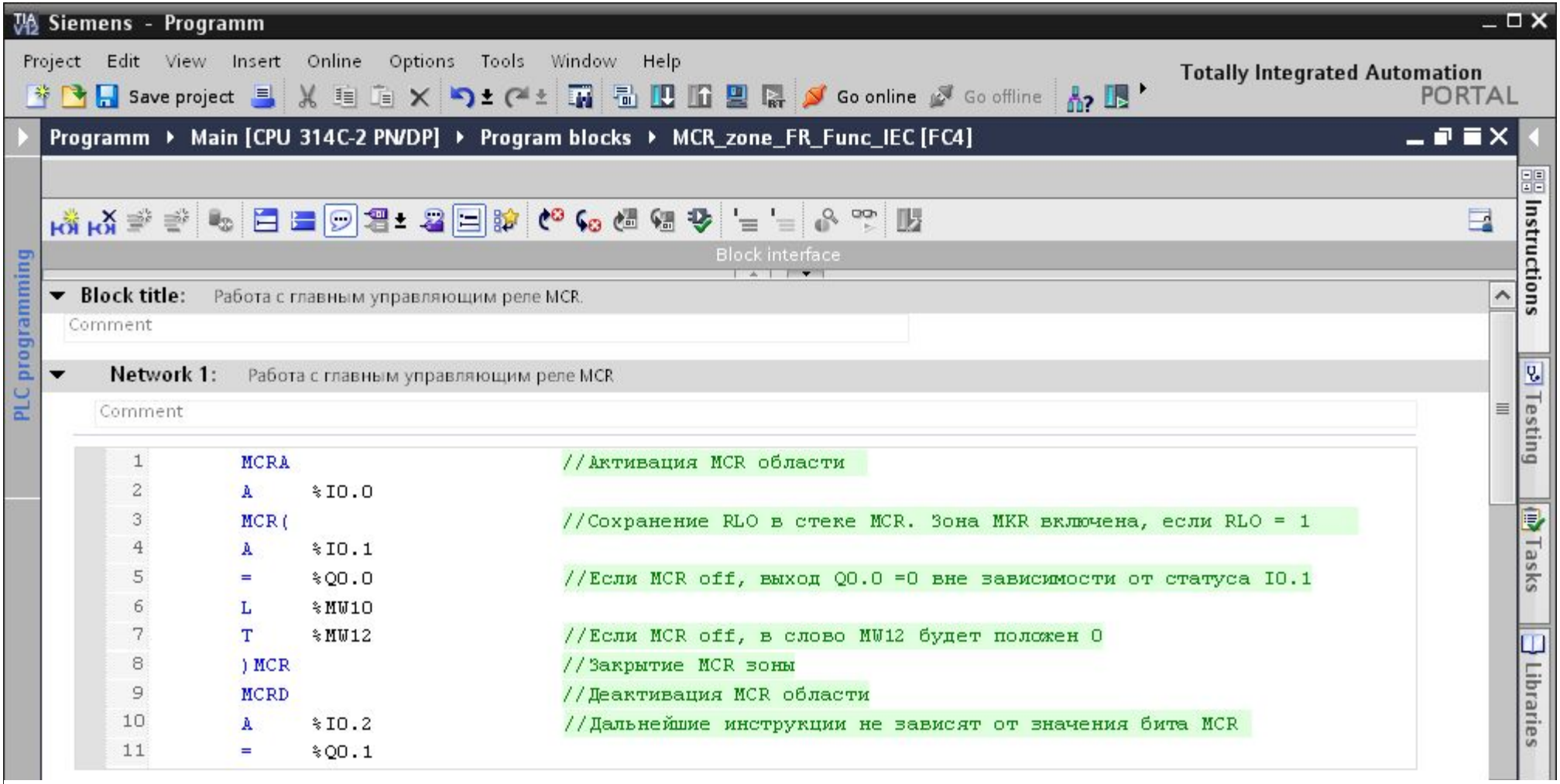
Пример построения перехода при вызове FC по условию



Использование команды RET

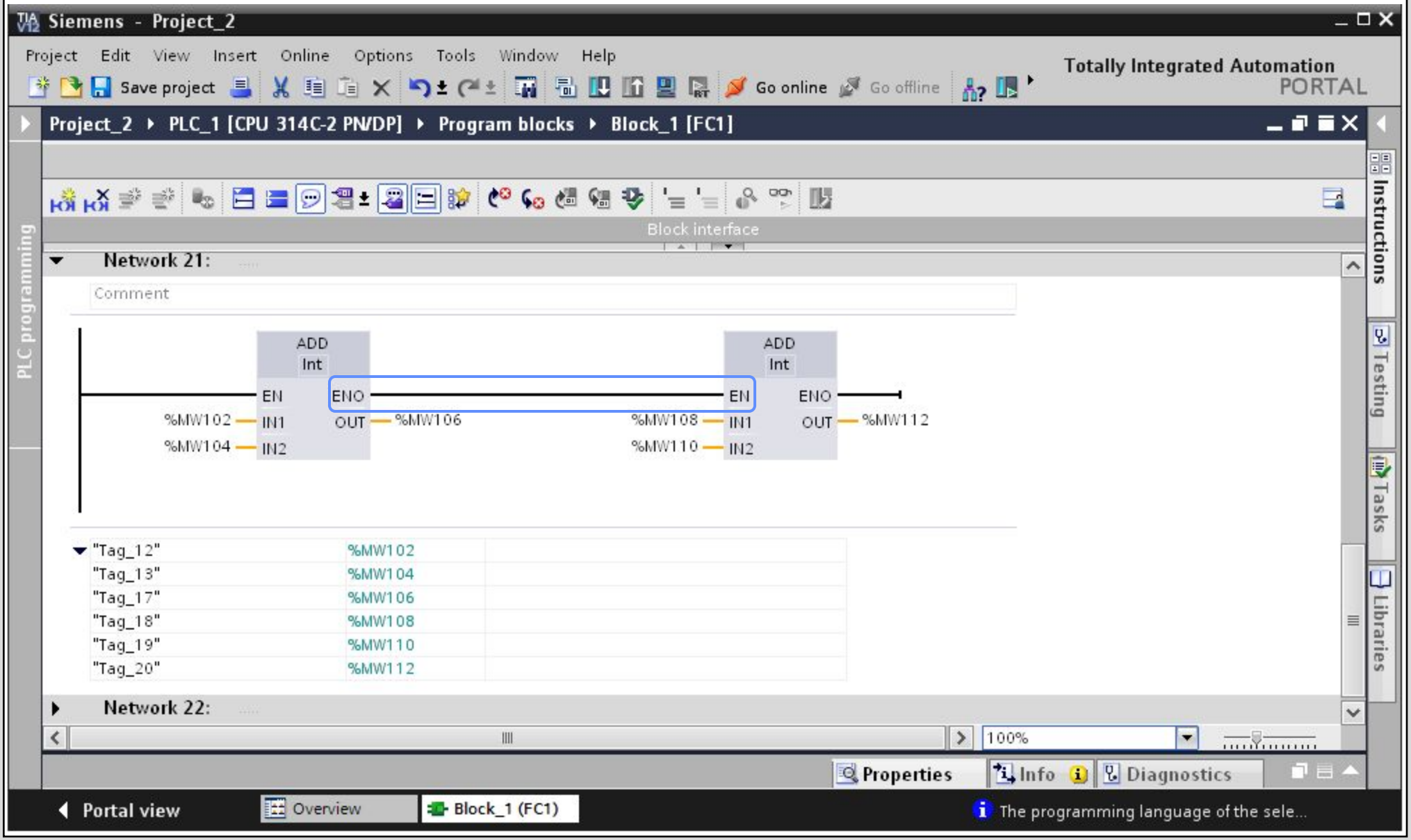


Работа с главным управляющим реле

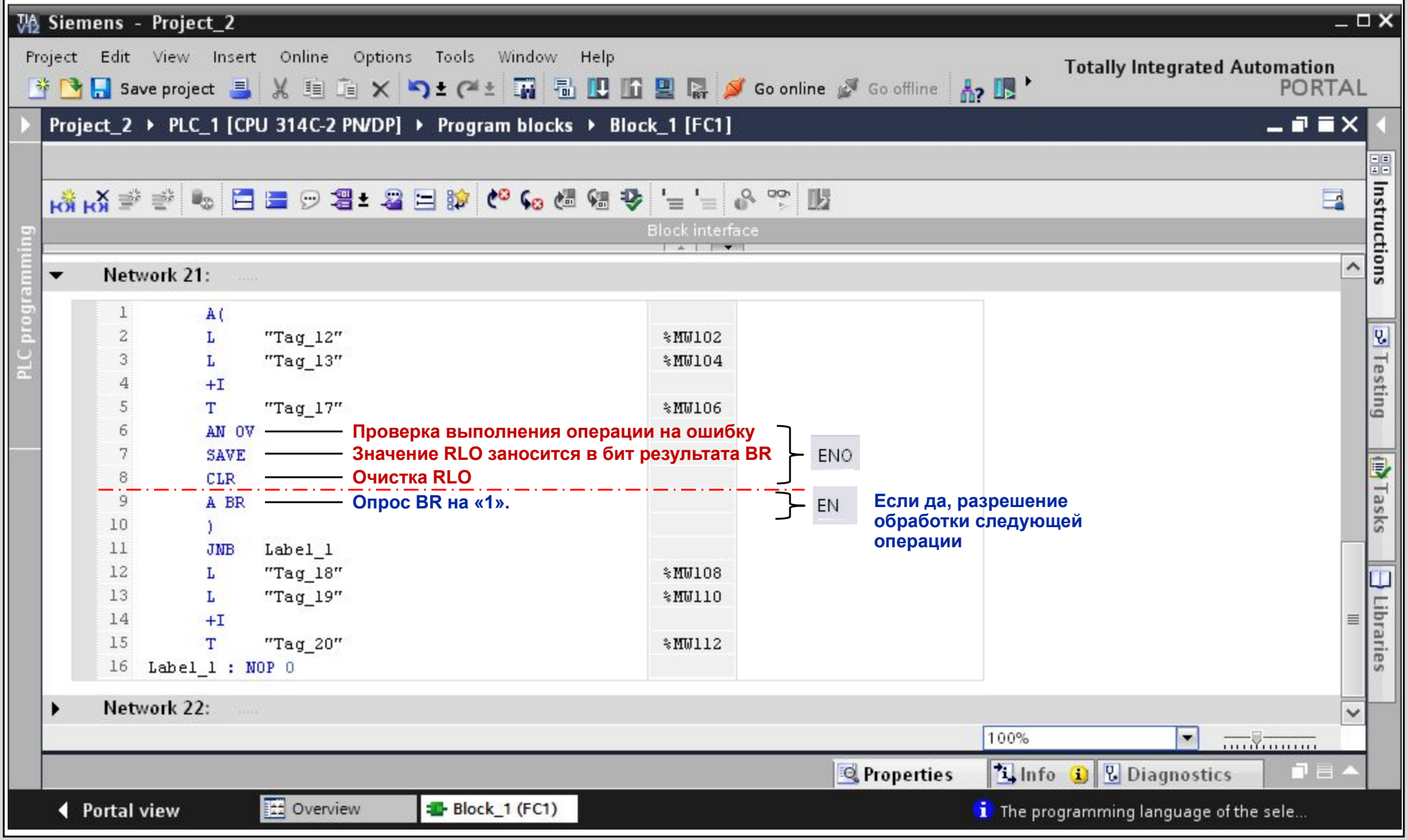


Команда MCRA активирует работу с MCR зоной. В зависимости от состояния бита MCR вложенные в MCR зону инструкции будут выполняться либо на обнуление результата сопряжения или передачи данных, если MCR=0, либо выполняться в нормальном режиме, если MCR=1. Команды SET и RESET при MCR=0 не меняют значения операнда. Деактивация работы с зоной MCR производится командой MCRD.
Команда MCRA должна выполняться обязательно в паре с командой MCRD, а команда открытия зоны MCR(в паре с командой закрытия зоны)MCR.
Глубина вложения MCR зоны – 8 инструкций.

Параметры EN/ENO (для представления в LAD/FBD)



Функциональность параметров EN/ENO



Примеры косвенной адресации

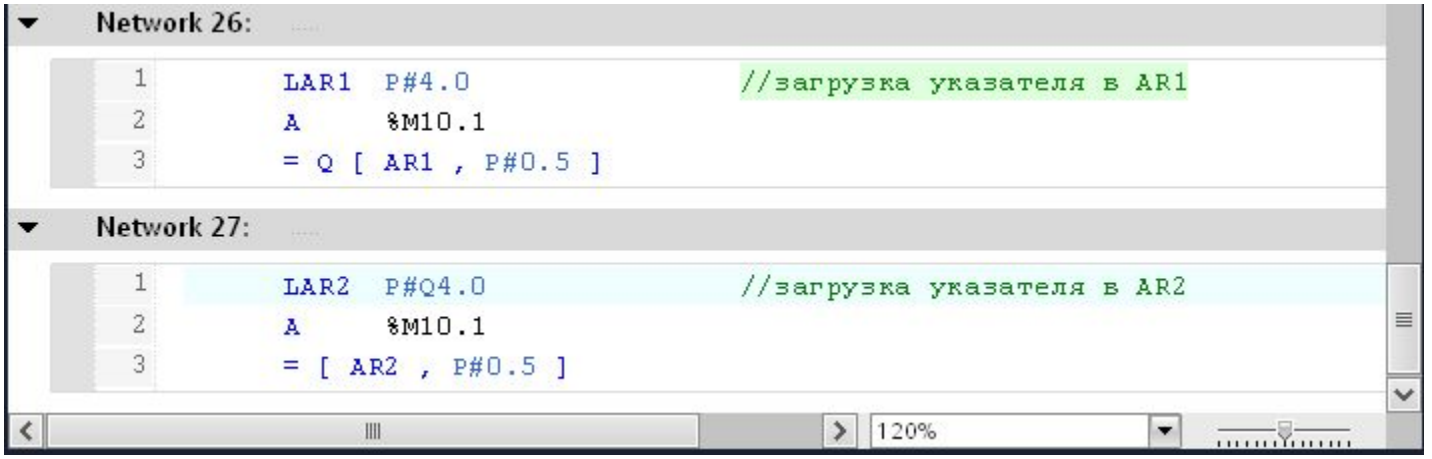
При косвенной адресации вместо указания адреса указывается место, где адрес может быть найден.
Есть два типа косвенной адресации:

1.Посредством памяти, когда указывается адрес в системной памяти, в которой можно найти адрес операнда

При косвенной адресации посредством памяти (memory indirect addressing) адрес указывается посредством адресованной ячейки памяти.
Адрес должен иметь размер:

- размер слова (WORD), если требуется использовать число в качестве указателя.
- двойного слова (DWORD), если требуется использовать указатель на область P# (area pointer)

2.Посредством адресного регистра



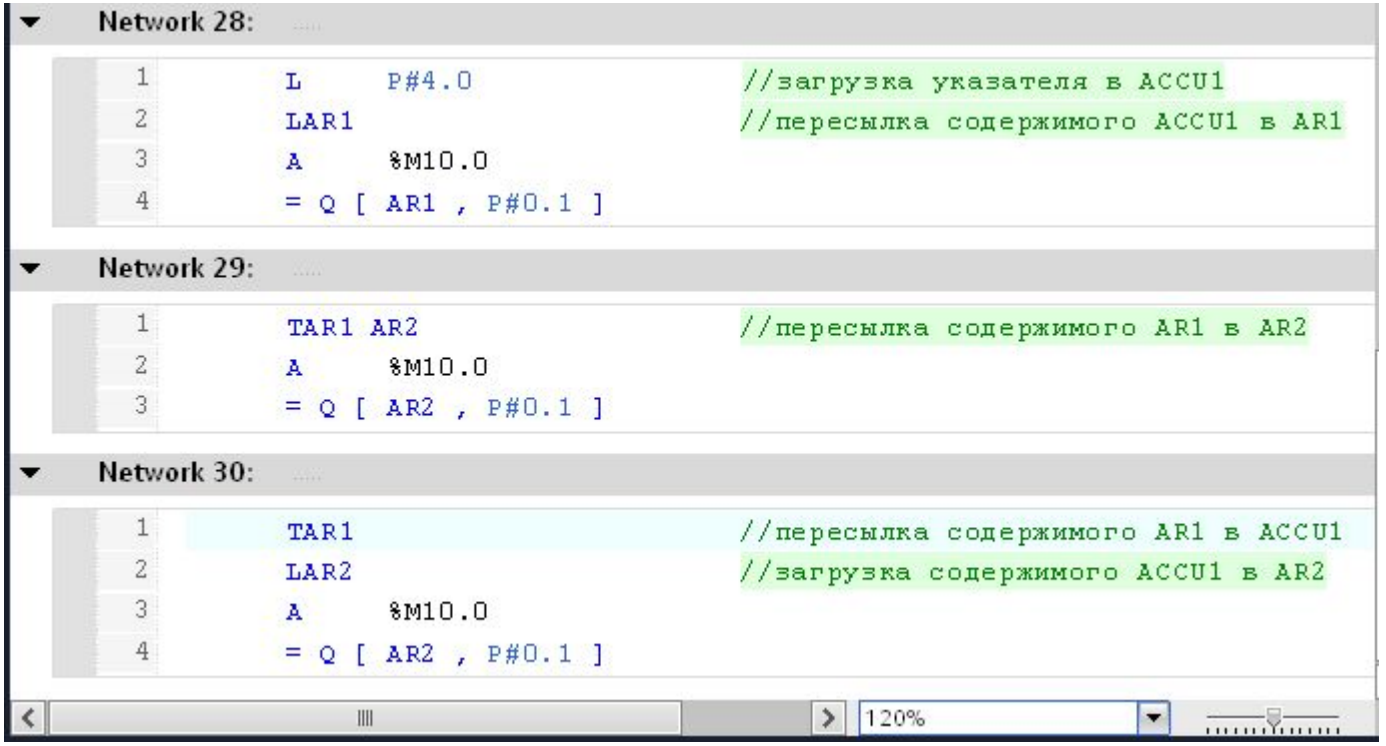
При косвенной адресации посредством регистра (register-indirect addressing) адрес указывается посредством одного из двух адресных регистров AR1 или AR2.

Загрузка и пересылка данных в адресные регистры

- В случае, если не задан адрес операнда, оператор LARn пересылает содержимое аккумулятора ACCU 1 в адресный регистр ARn.

- Если используеся инструкция TAR1 AR2, то при ее выполнении происходит копирование содержимого адресного регистра AR1 в адресный регистр AR2.

- Оператор TARn пересылает содержимое ARn в ACCU 1



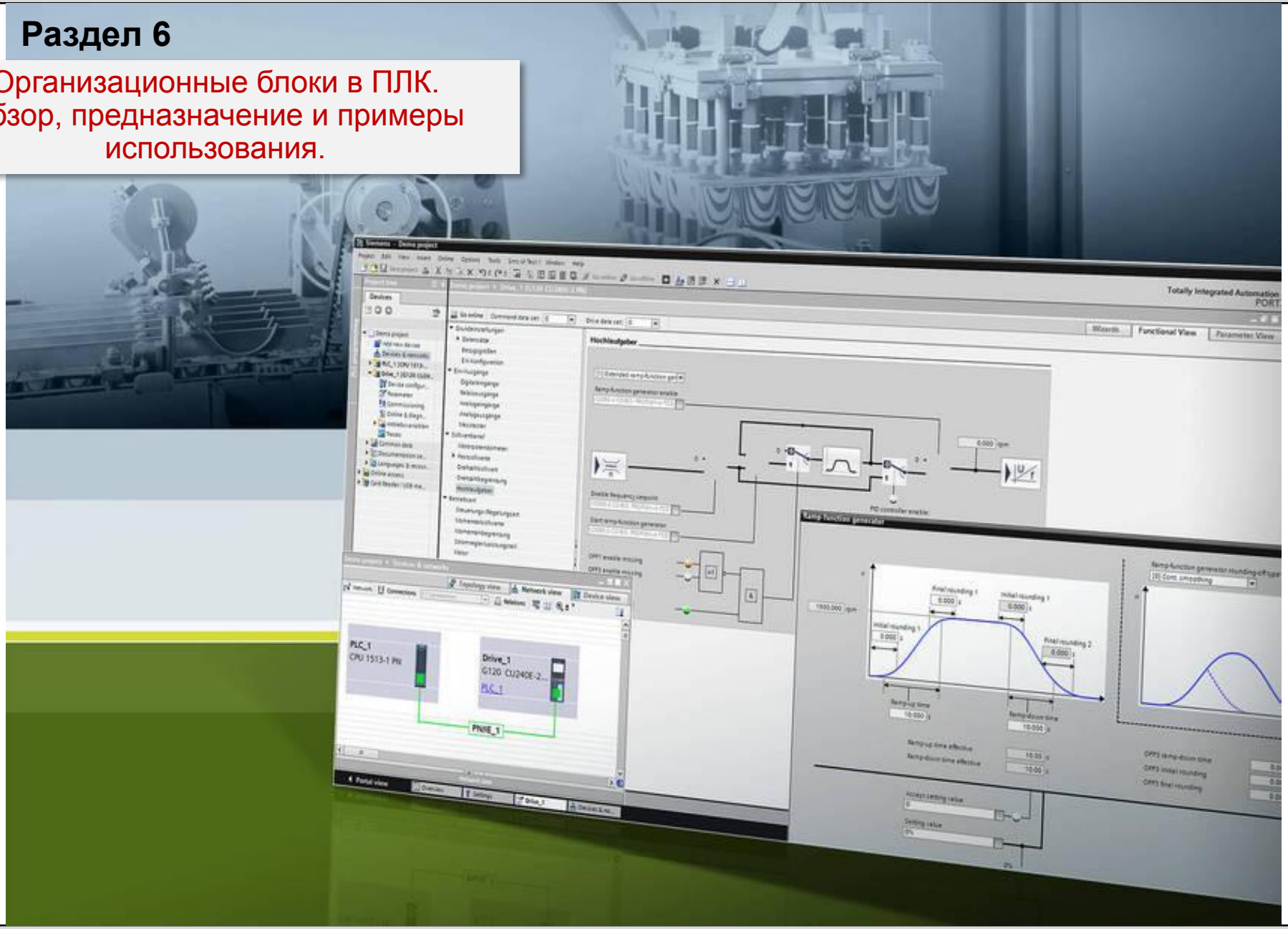
Конец раздела 5. Окно навигации

- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)

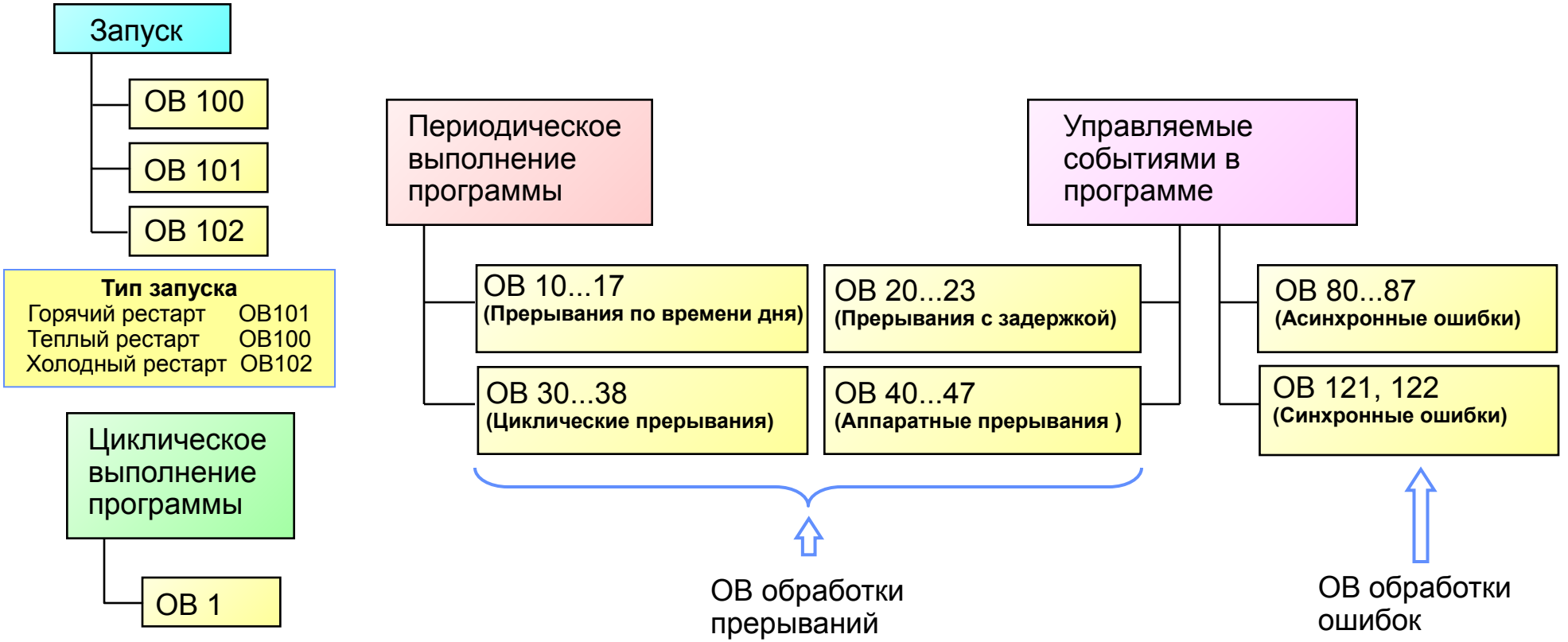
  Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 6

Организационные блоки в ПЛК. Обзор, предназначение и примеры использования.



Обзор организационных блоков



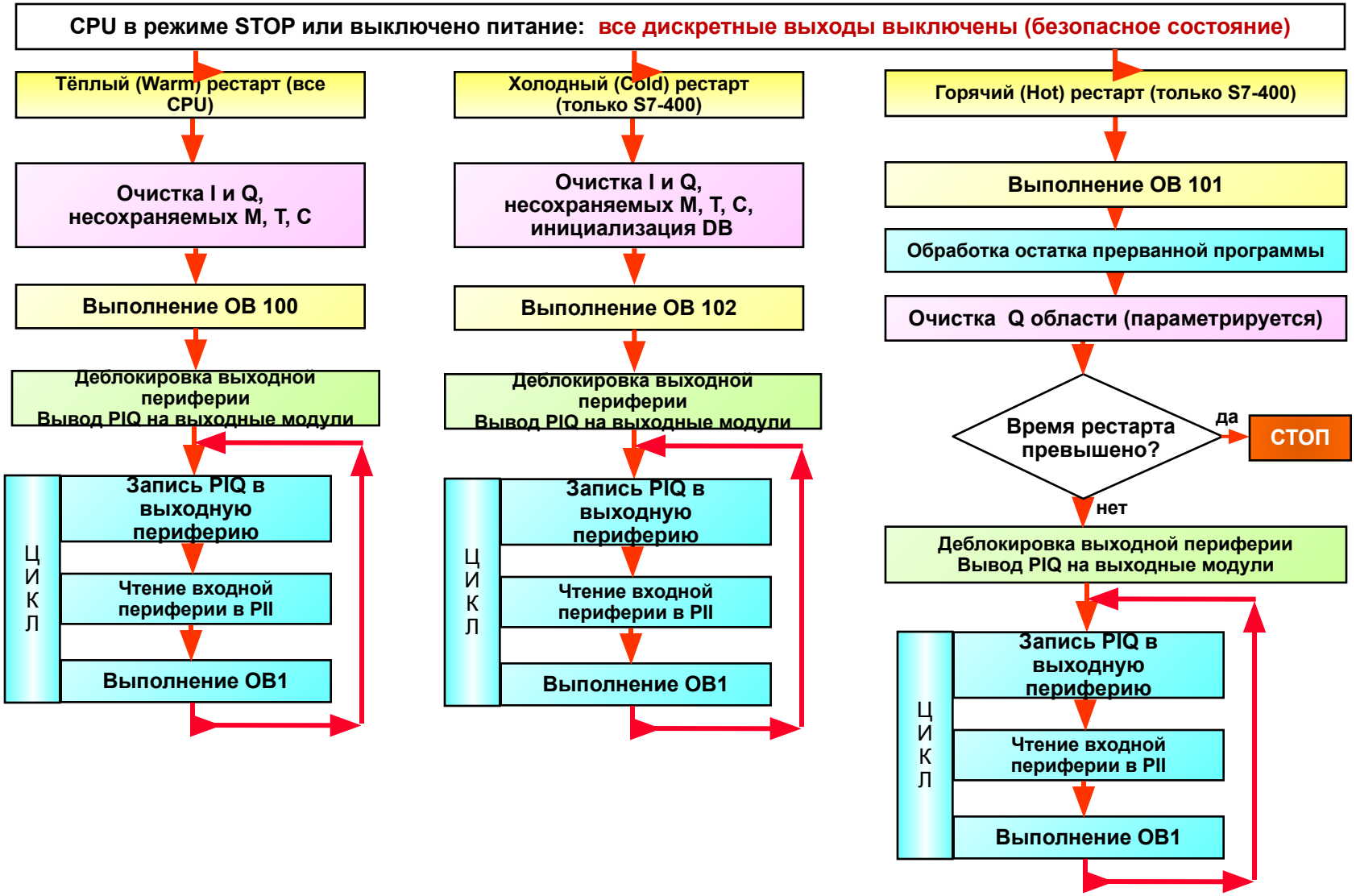
Прерывания циклической программы

Организационный блок		Приоритет в S7-300	Функция для управления OB	Примечание
Функция	Номер (зависит от CPU)			
Прерывание по времени дня	OB 10 ... 17	2	SET_TINT, ACT_TINT	Альтернатива конфигурированию
Циклическое прерывание	OB 30 ... 38	12	нет	
Прерывание с задержкой	OB 20 ... 23	3	SRT_DINT	Обязательное использование
Аппаратное прерывание	OB 40 ... 47	16	нет	
Диагностическое прерывание	OB 81 ... 87	26	нет	

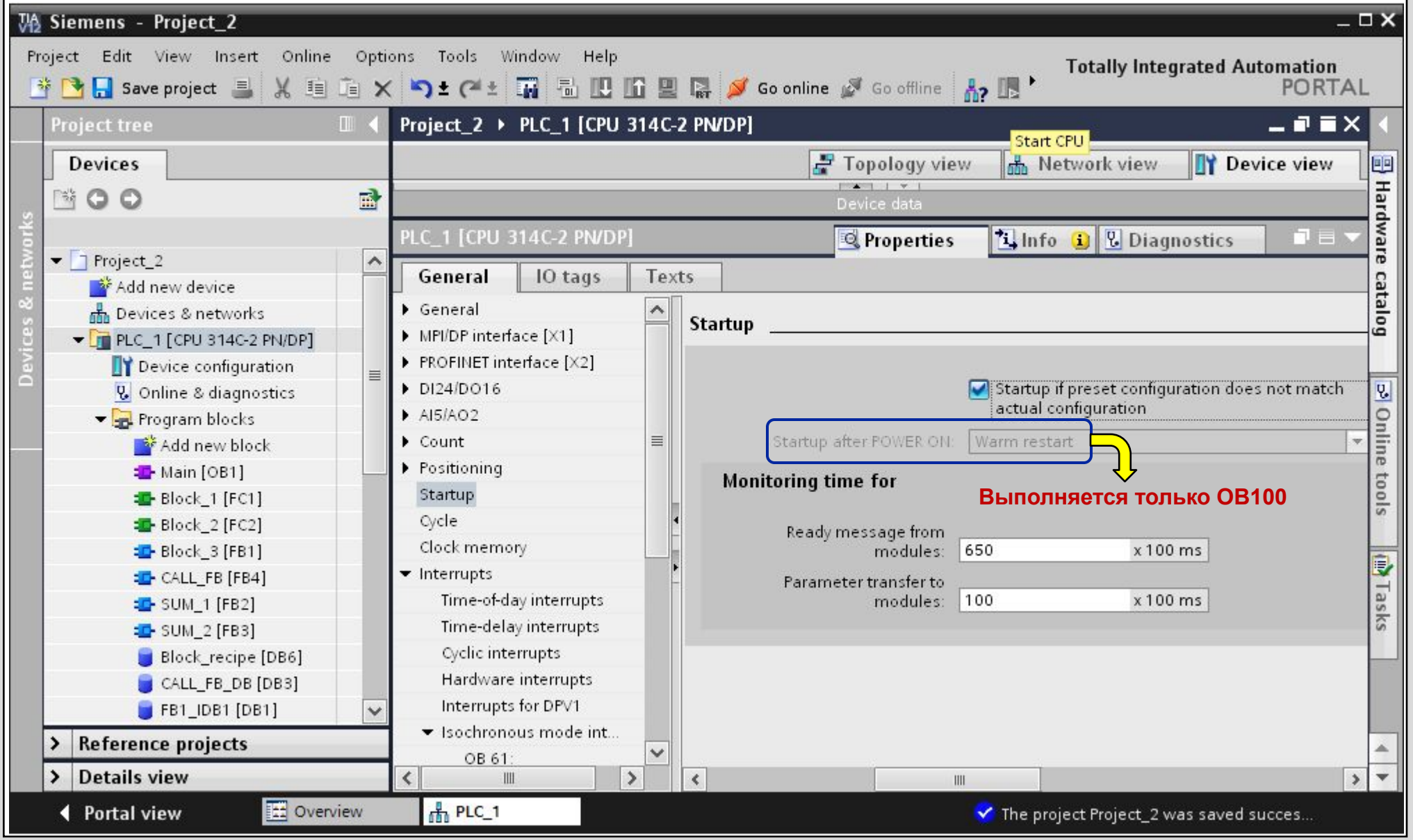
Стартовая информация OB (содержится в области L)

Байты L-стека			
0 / 1	Стартовое событие	Кодовое значение	Управляющая информация
2 / 3	Приоритет	Номер OB	
4 / 5	Форматы данных в байтах локального стека 8, 9, 10, 11		Стартовая информация
6 / 7	Дополнительная информация 1 (например, начальный адрес модуля прерывания)		
8 / 9	Дополнительная информация 2 (например, статус прерывания)		
10 / 11	Дополнительная информация 3 (Например, номер канала)		
12 / 13	Год	Месяц	Стартовое время
14 / 15	День	Часы	
16 / 17	Минуты	Секунды	
18 / 19	1/10 секунды, 1/100 секунды	1 /1000 секунды, день недели	

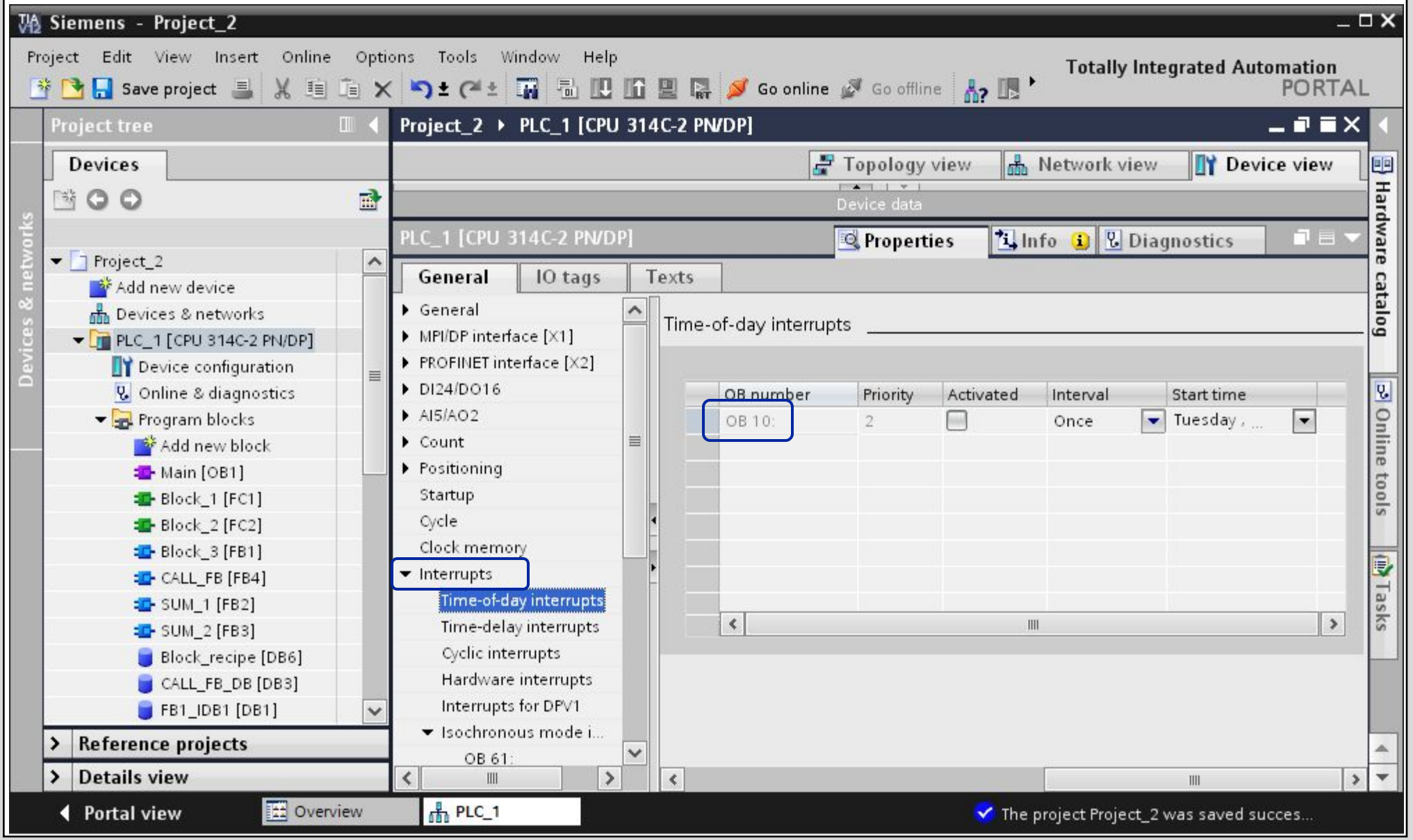
Виды рестарта и вызываемых OB



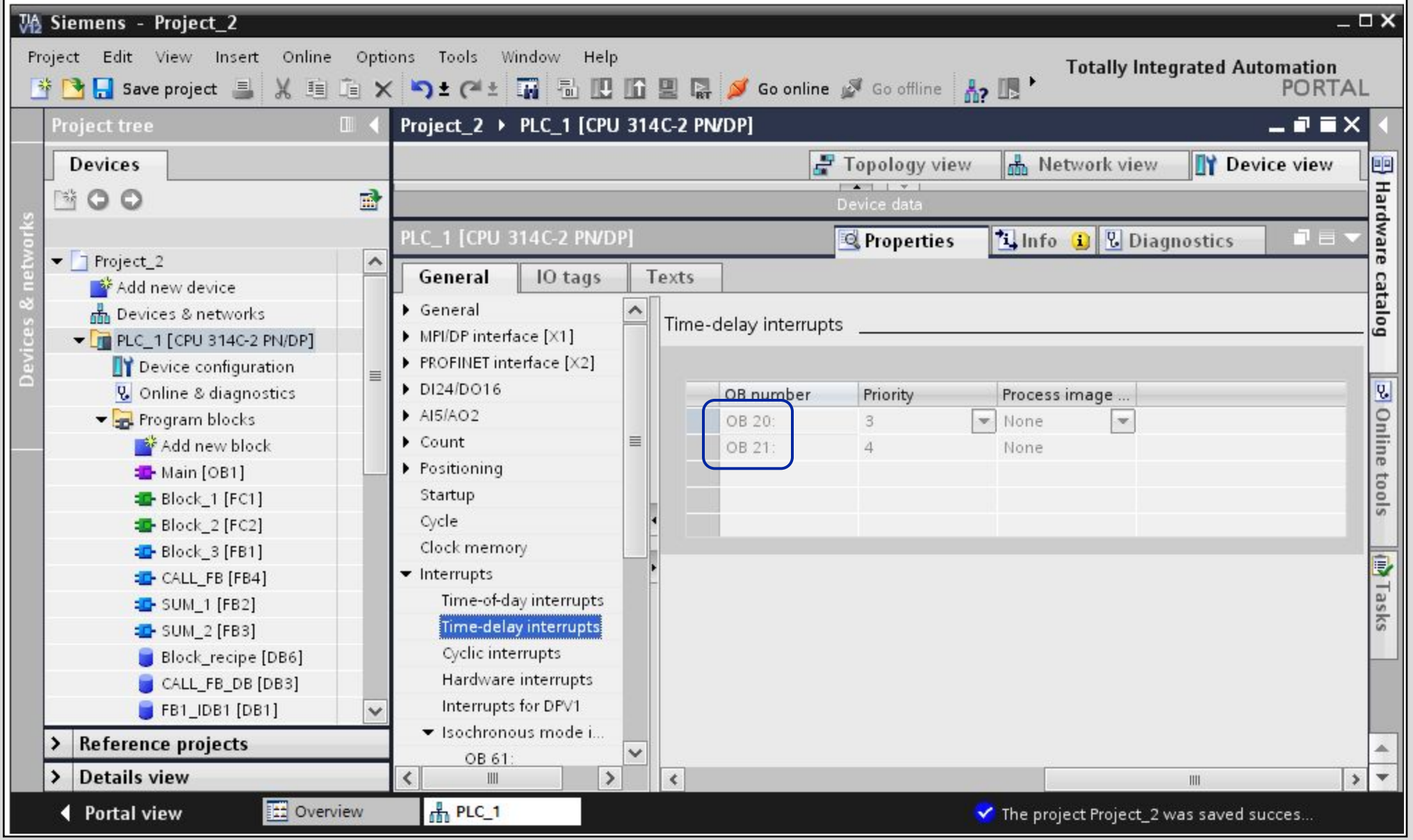
Свойства ЦПУ. OB обработки рестарта



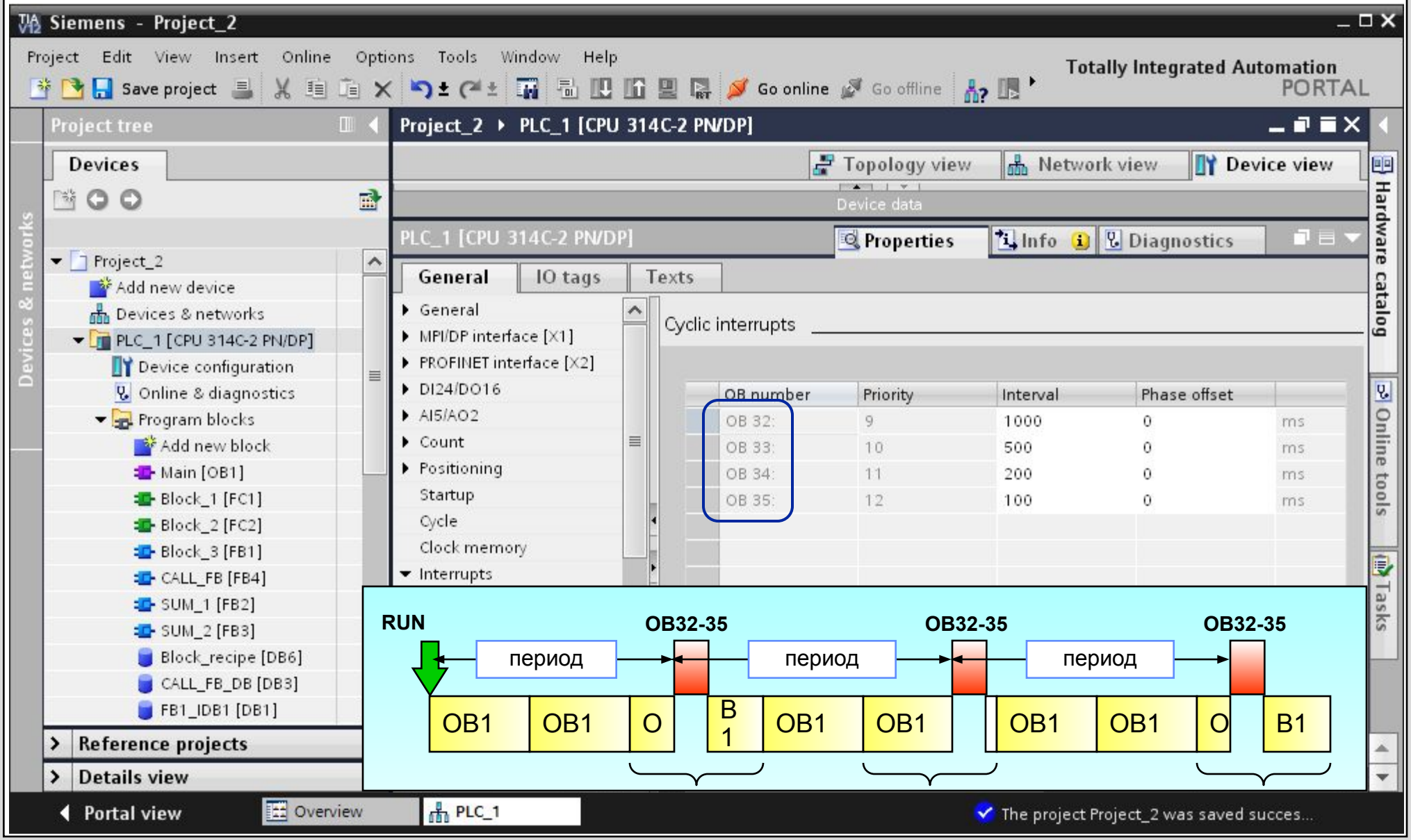
Свойства ЦПУ. OB обработки прерываний



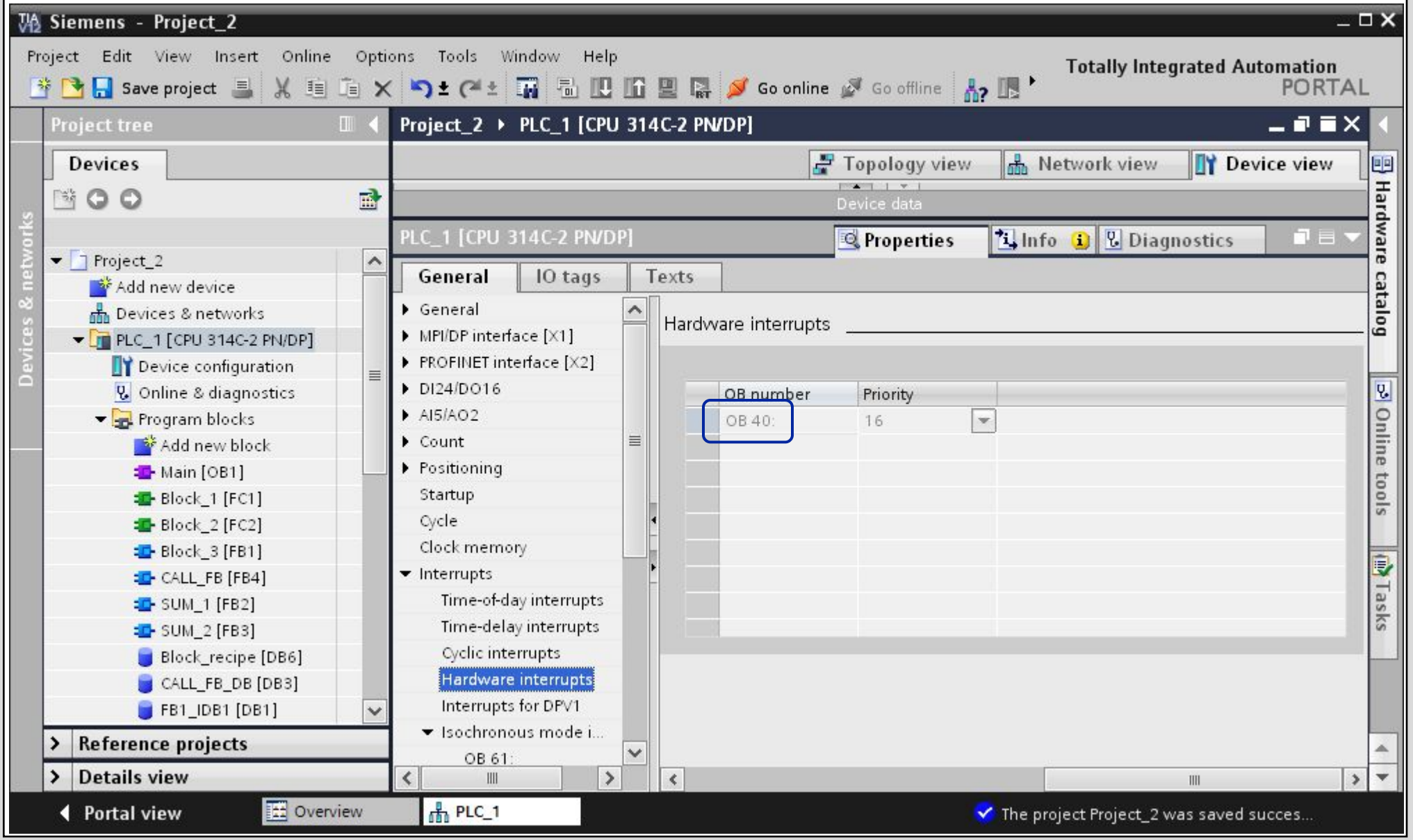
Свойства ЦПУ. OB обработки прерываний



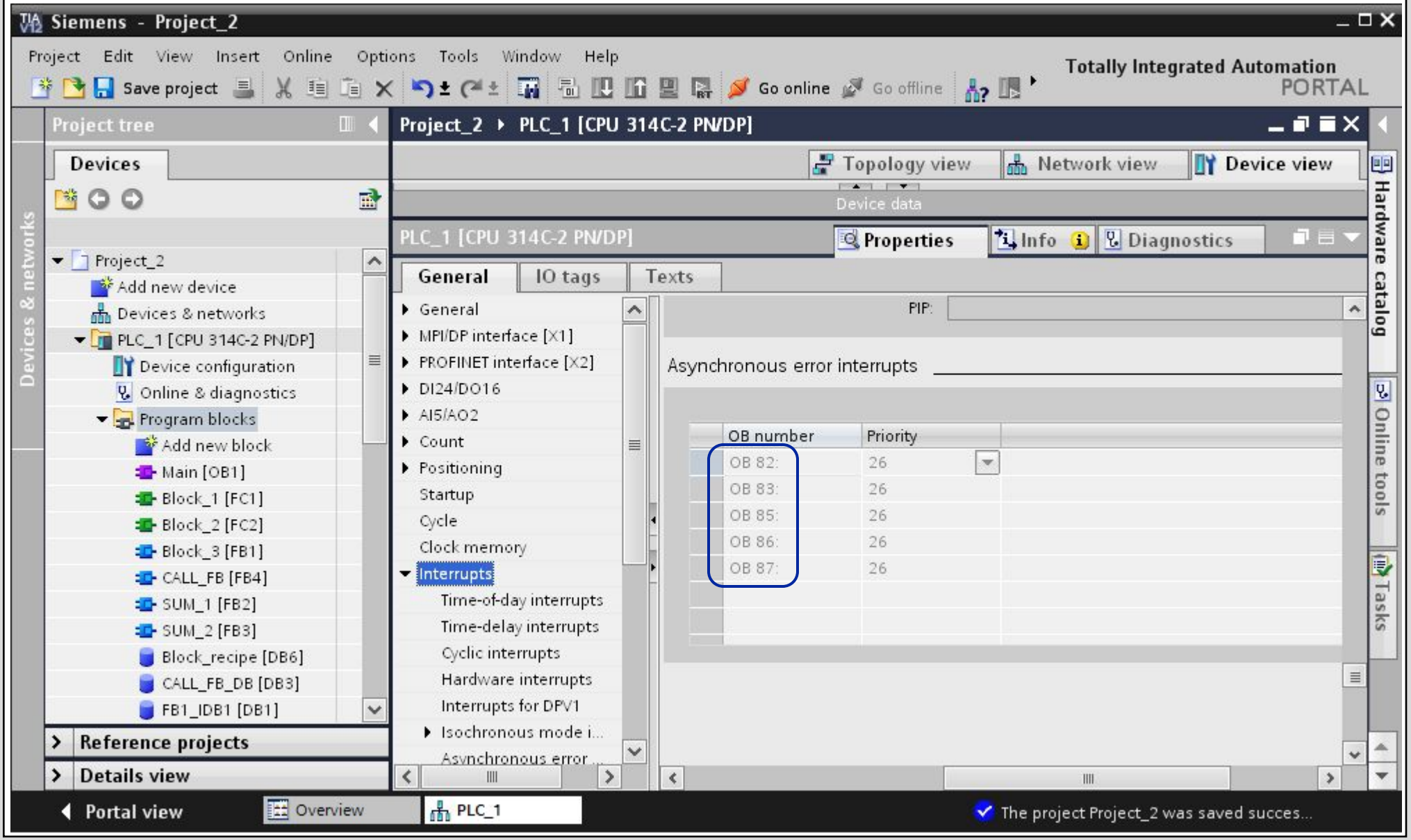
Свойства ЦПУ. OB обработки прерываний



Свойства ЦПУ. OB обработки прерываний



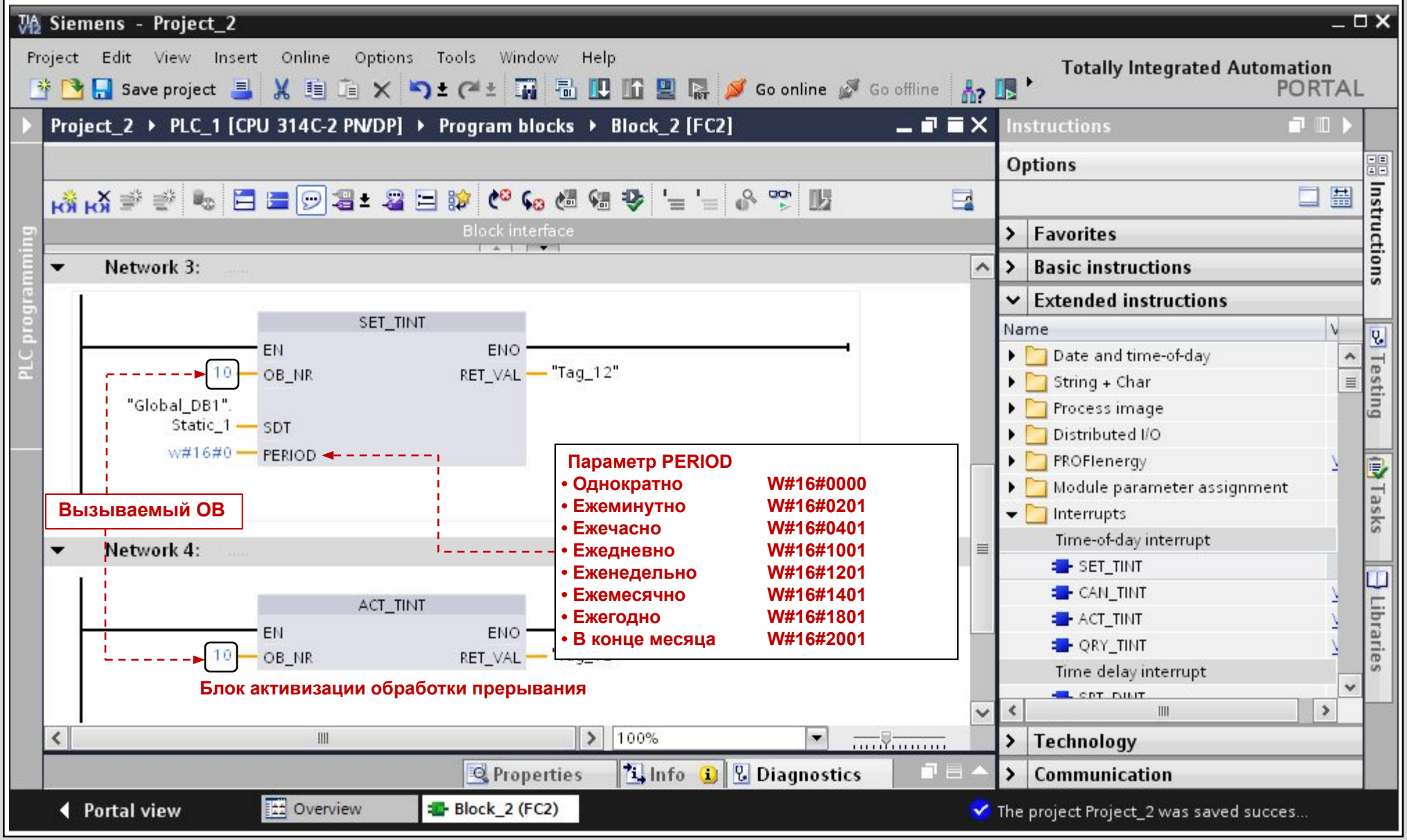
Свойства ЦПУ. OB обработки асинхронных ошибок



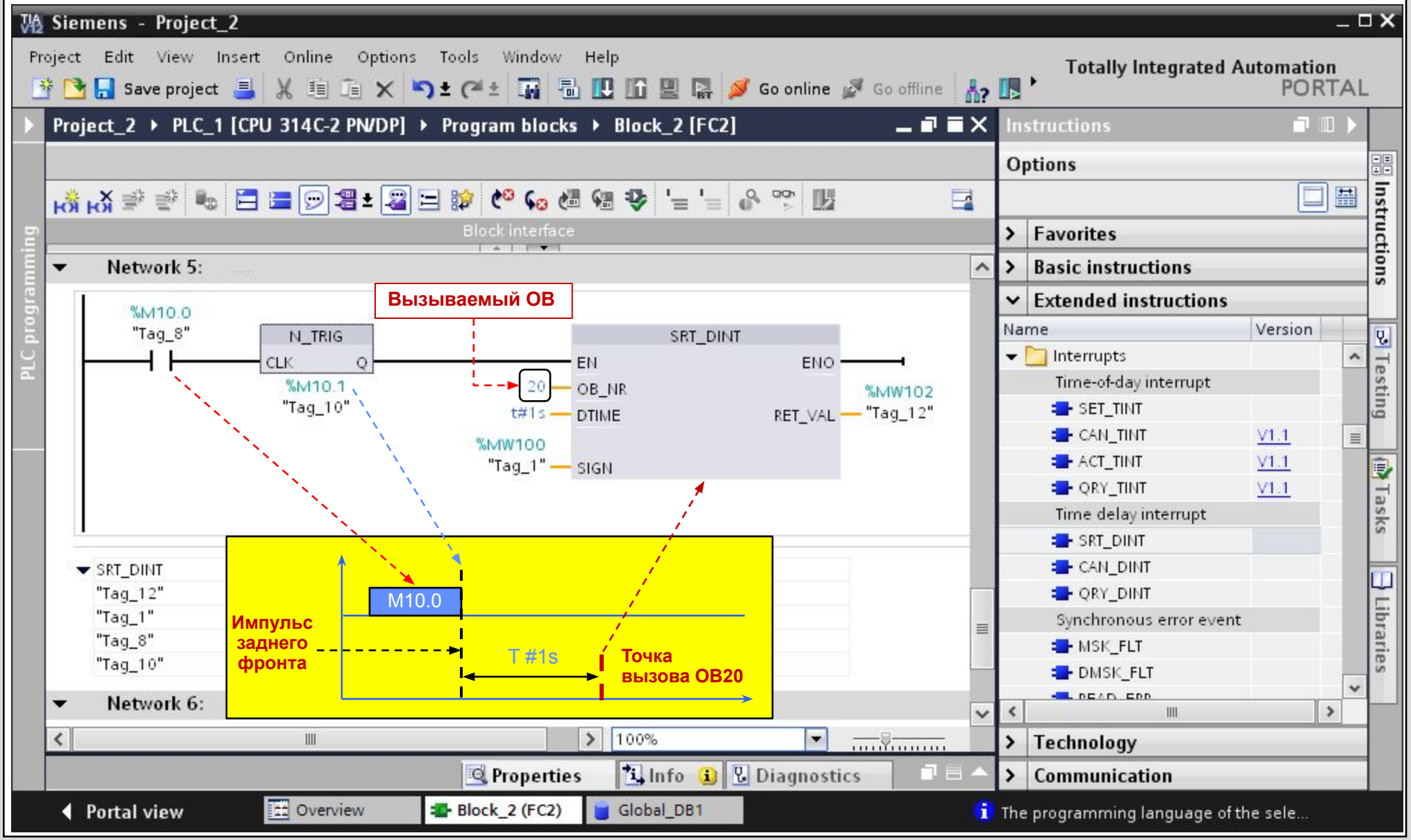
OB обработки асинхронных ошибок

Тип ошибки	Пример	OB	Приоритет
Ошибка времени	Превышено максимальное время цикла	OB80	26
Неисправность питания	Неисправность буферной батареи	OB81	
Диагностическое прерывание	Обрыв провода на входе модуля с внутренней диагностикой	OB82	
Вставка/удаление модуля	Удаление сигнального модуля во время функционирования S7-400™	OB83	
Дефект аппаратуры CPU	Неправильный уровень сигнала в интерфейсе MPI	OB84	
Ошибка выполнения программы	Ошибка в обновлении областей отображения процесса (дефект модуля)	OB85	
Дефект стойки	Неисправность в корзине расширения или В DP- слэйве	OB86	
Ошибка связи	Ошибка при чтении телеграммы сообщения	OB87	26 / 28

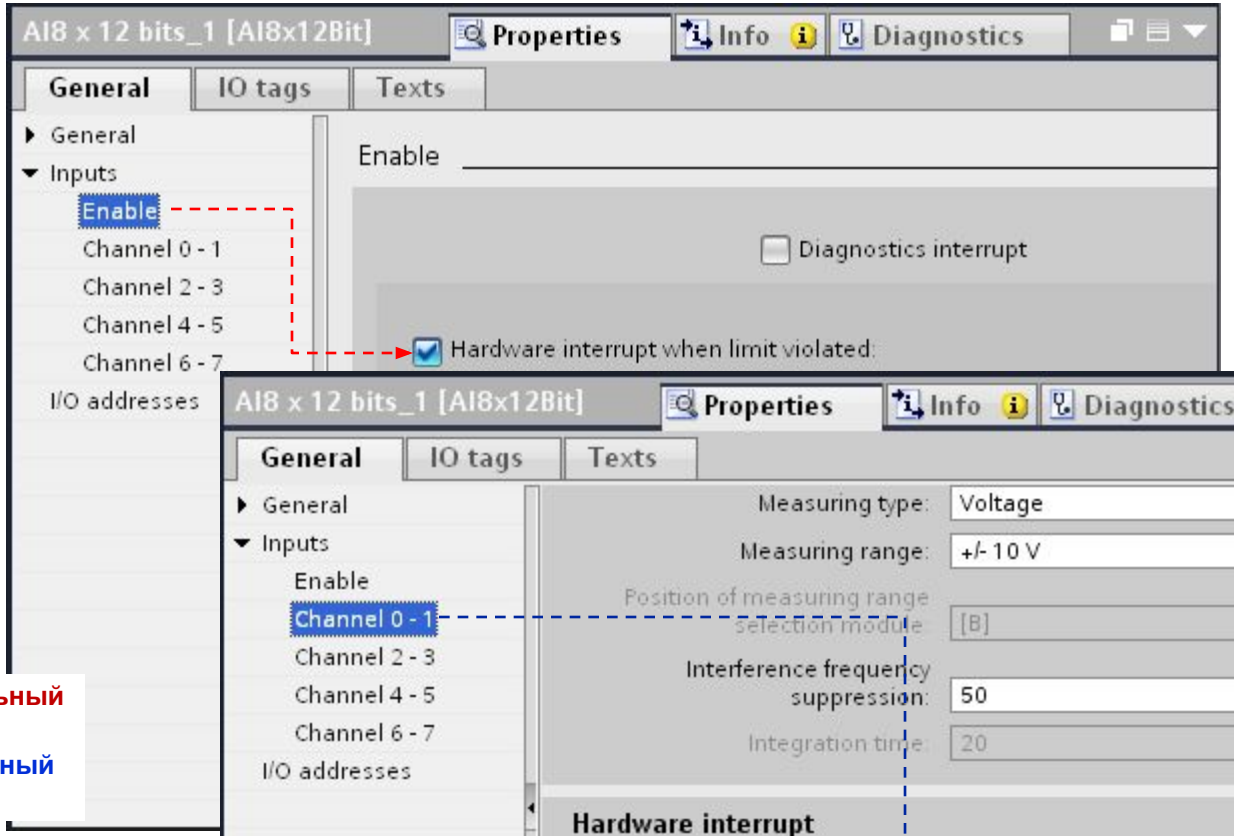
Пример обработки прерывания по времени дня

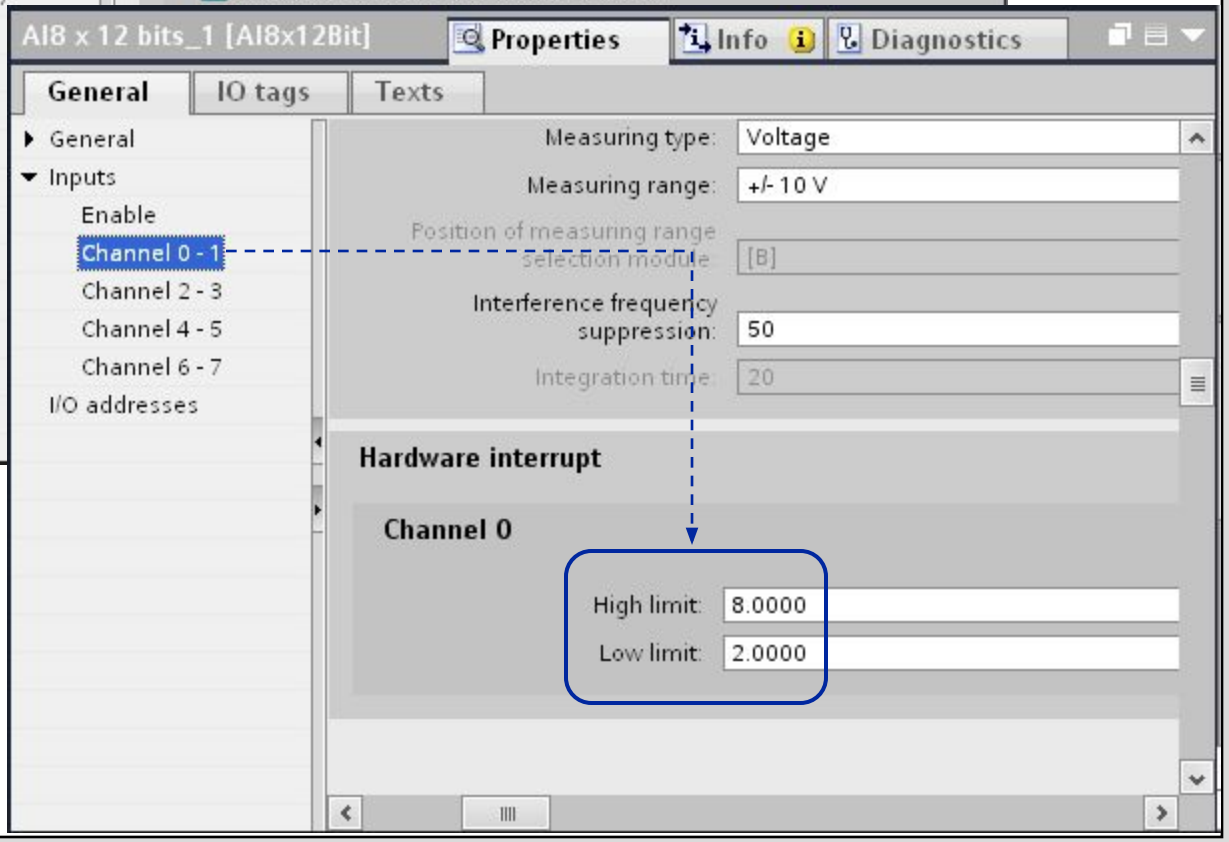


Пример обработки прерывания с задержкой времени



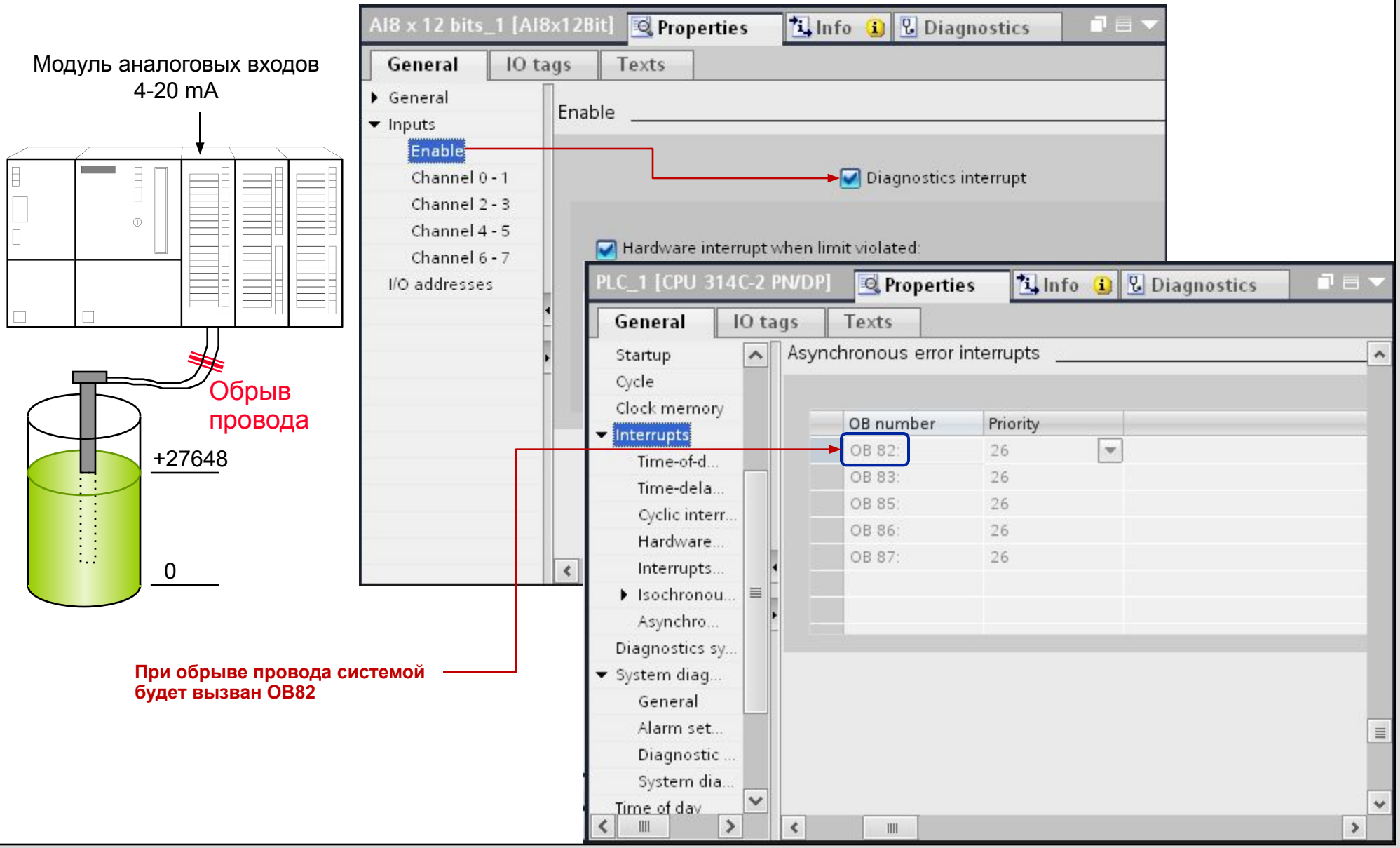
Аппаратное прерывание по ограничениям уровней (OB40)





При выходе сигнала за заданные пределы системой будет вызван OB40

Обработка OB обрыва провода модуля входа AI с диагностикой



OB обработки синхронных ошибок

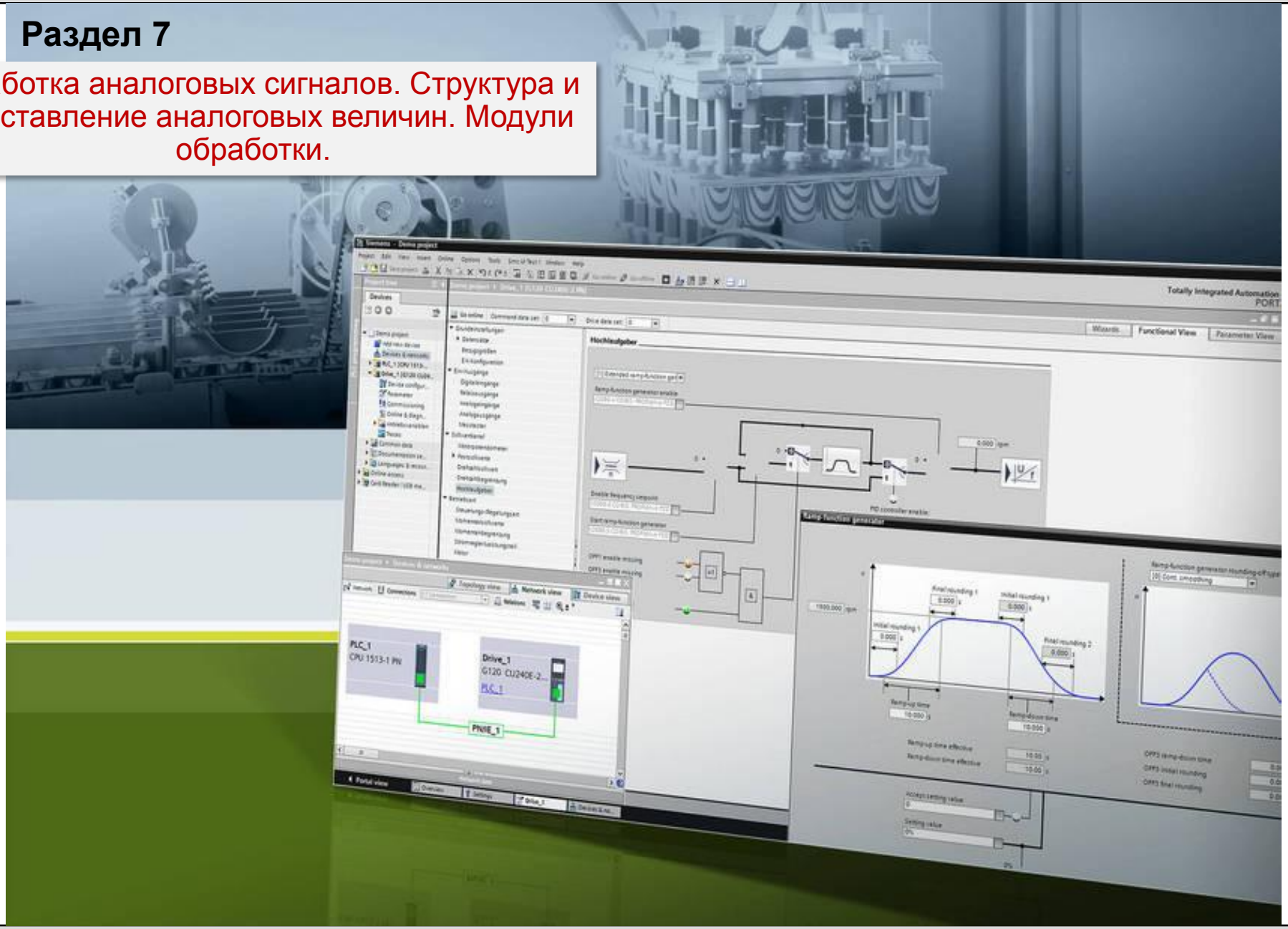
Тип ошибки	Пример	OB	Приоритет
Ошибка программирования	**В программе вызван блок, который отсутствует в CPU**	OB121	**Тот же, что у OB, который прерван в результате ошибки в нем**
Ошибки доступа	**В программе производится обращение к модулю, который или неисправен, или отсутствует (например, при прямом доступе к несуществующему периферийному модулю)**	OB122	

Конец раздела 6. Окно навигации

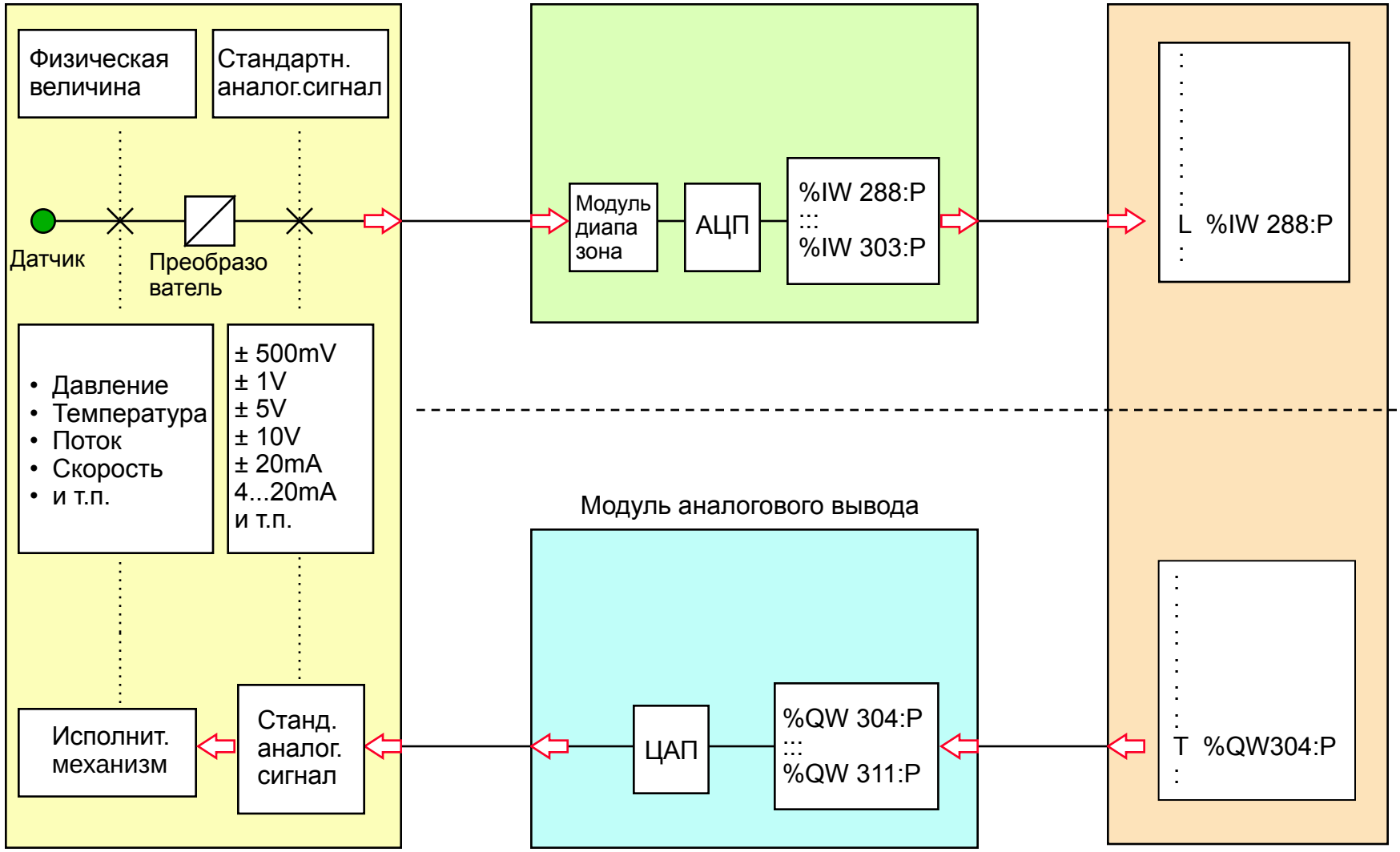
- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 7

Обработка аналоговых сигналов. Структура и представление аналоговых величин. Модули обработки.



Функциональная структура обработки аналоговых сигналов



Представление входной аналоговой величины

Диапазон	Напряжение		Ток		Сопротивление		Температура (PT100)	
	Например, ± 10В	Значение	Например, 4 .. 20 мА	Значение	Например, 0...300 Ом	Значение	Например, -200...+850°C	Значение
Переполнение	>= 11.759	32767	>= 22.815	32767	>=352.778	32767	>= 1000.1	32767
Превышение верхней границы	11.7589 : 10.0004	32511 : 27649	22.810 : 20.0005	32511 : 27649	352.767 : 300.011	32511 : 27649	1000.0 : 850.1	10000 : 8501
Номинальный диапазон	10.00 7.50 : -7.5 -10.00	27648 20736 : -20736 -27648	20.000 16.000 : : 4.000	27648 20736 : : 0	300.000 225.000 : : 0.000	27648 20736 : : 0	850.0 : : : -200.0	8500 : : : -2000
Превышение Нижней границы	- 10.0004 : - 11.759	- 27649 : - 32512	3.9995 : 1.1852	- 1 : - 4864	Отрицат. значения невозможны	- 1 : - 4864	- 200.1 : - 243.0	- 2001 : - 2430
Переполнение	<= - 11.76	- 32768	<= 1.1845	- 32768		- 32768	<= - 243.1	- 32768

Представление выходной аналоговой величины

Диапазон	Значение	Напряжение			Ток		
		Выходные диапазоны:			Выходные диапазоны:		
		0 – 10 В	1 – 5 В	± 10 В	0 – 20 мА	4 – 20 мА	± 20 мА
Переполнение	>=32767	0	0	0	0	0	0
Верхний диапазон	32511 : 27649	11.7589 : 10.0004	5.8794 : 5.0002	11.7589 : 10.0004	23.515 : 20.0007	22.81 : 20.005	23.515 : 20.0007
Номинальный диапазон	27648 : 0 : - 6912 - 6913 : : : - 27648	10.0000 : 0	5.0000 : 1.0000	10.0000 : 0 : : : : : : : -10.0000	20.000 : 0	20.000 : 4.000	20.000 : 0 : : : : : : : -20.000
Нижний диапазон	- 27649 : - 32512	0.9999 0	0.9999 0	- 10.0004 : - 11.7589	3.9995 0	3.9995 0	- 20.007 : - 23.515
Переполнение	<= - 32513	0	0	0	0	0	0

Выбор и параметрирование аналоговых модулей 7KF0xx

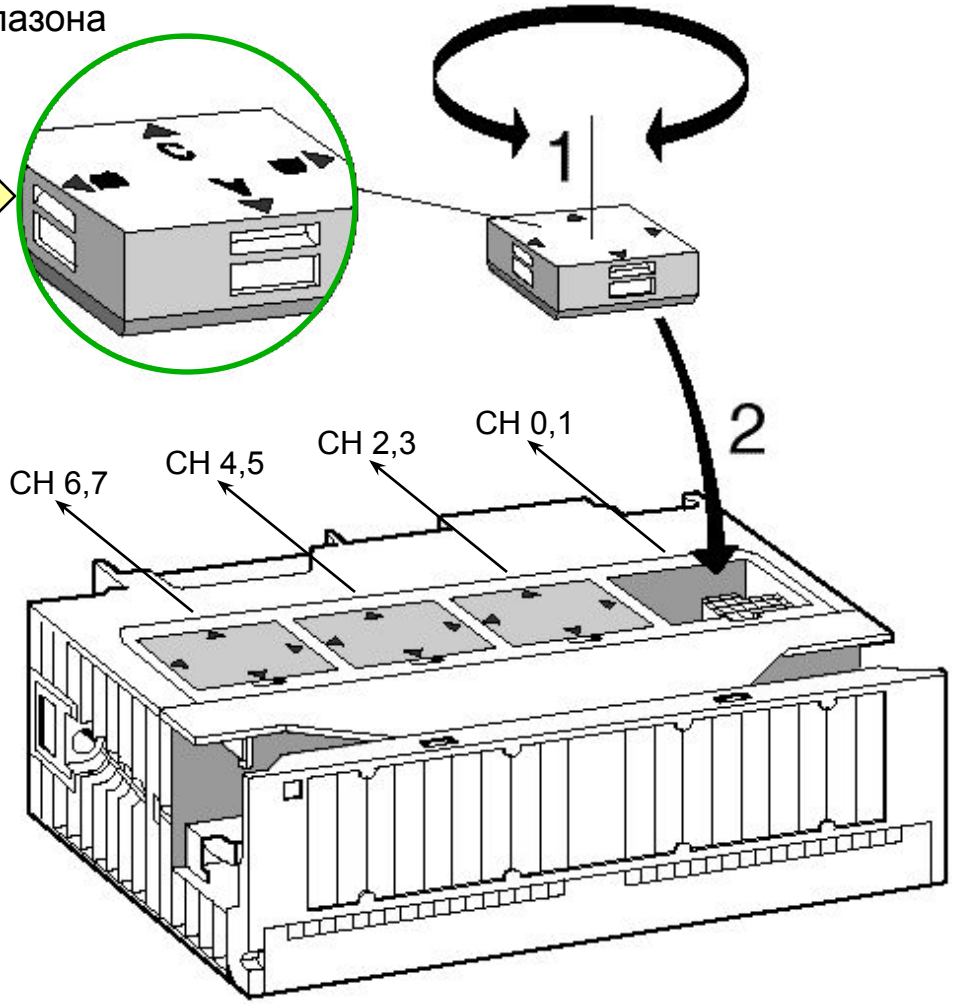
Установка коммутатора величины и диапазона измерений для модулей серии 7KF0xx

A: 80/250/500/1000mv/Pt100

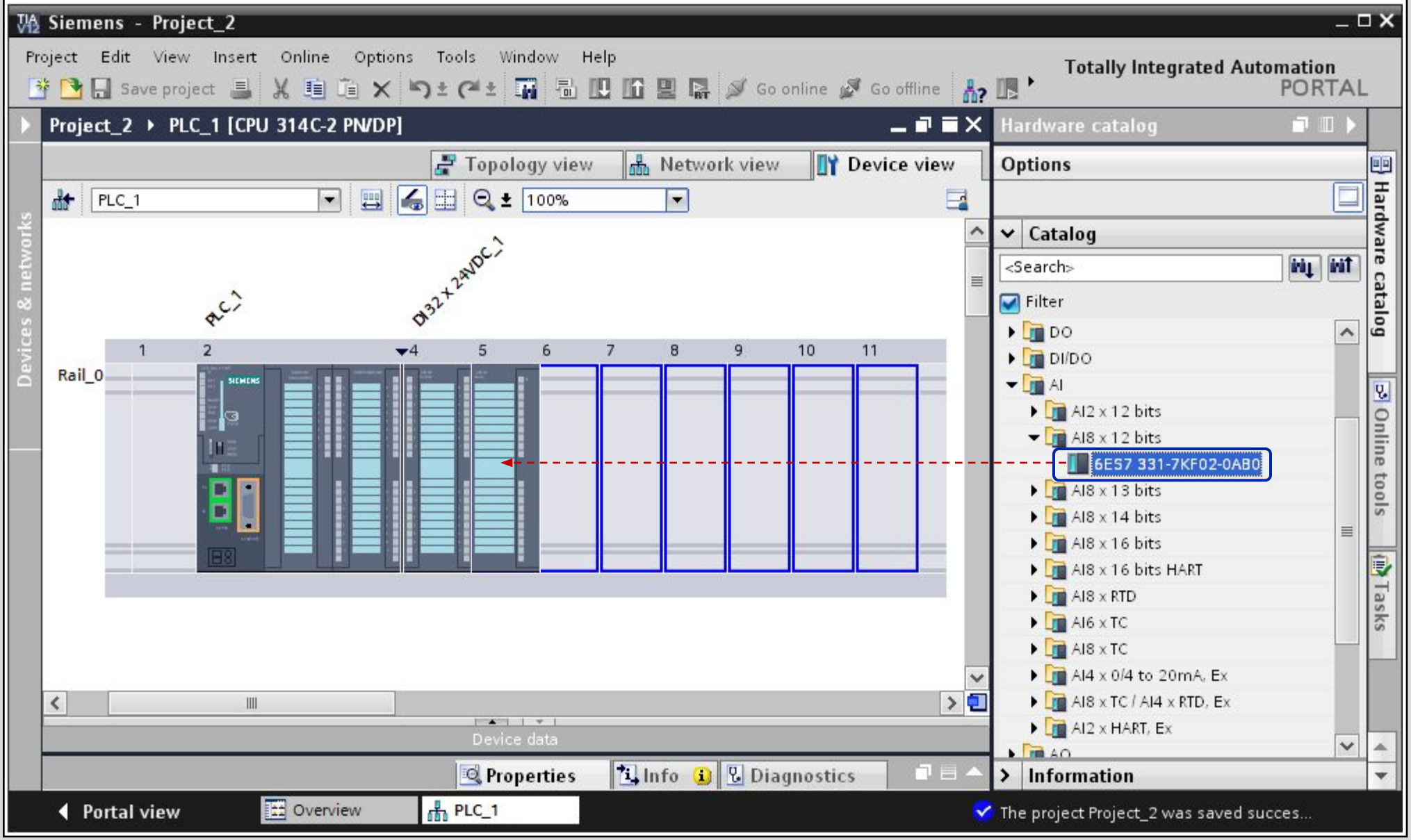
B: 2,5/5/10V

C: 4 WIRE CURRENT

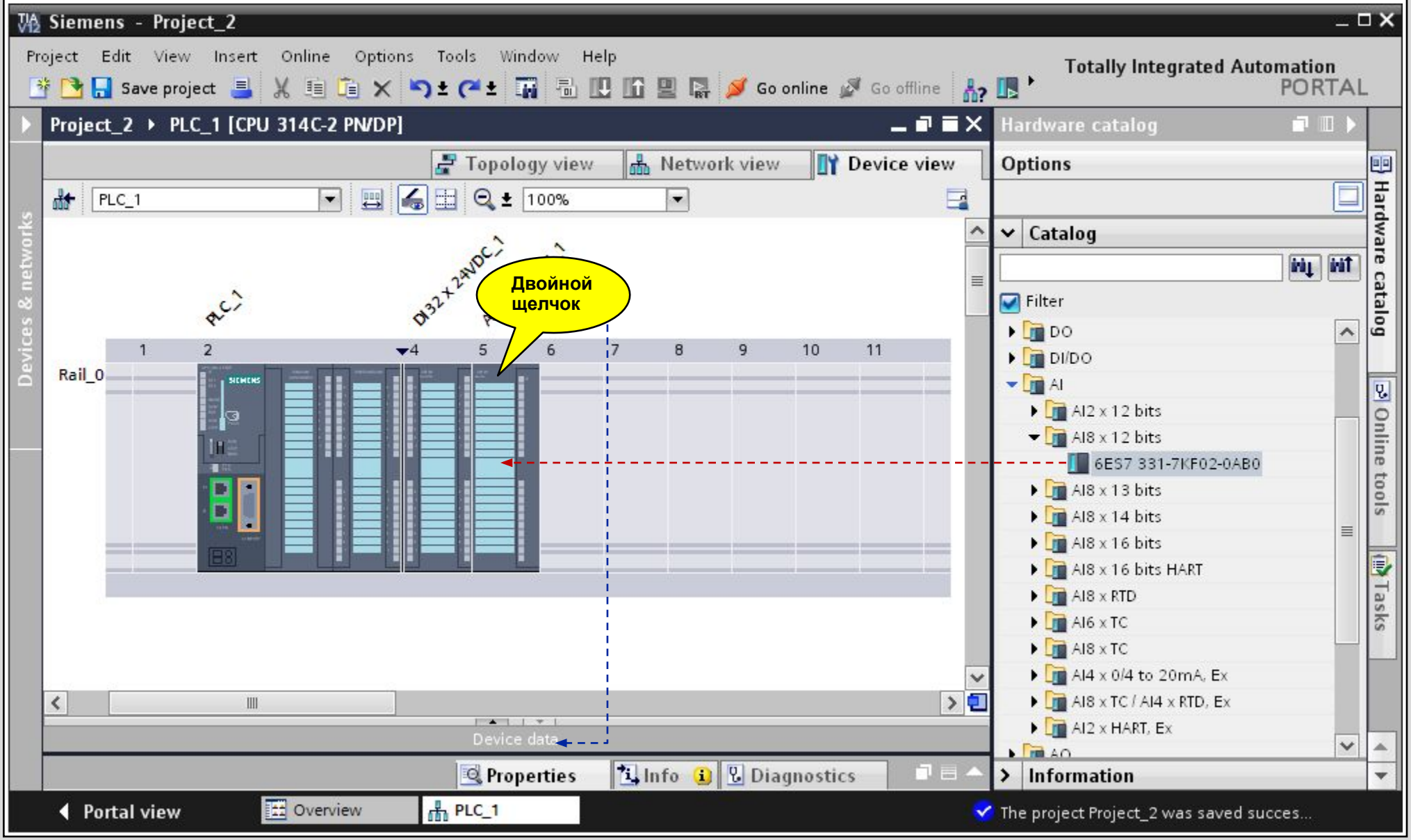
D: 2 WIRE CURRENT



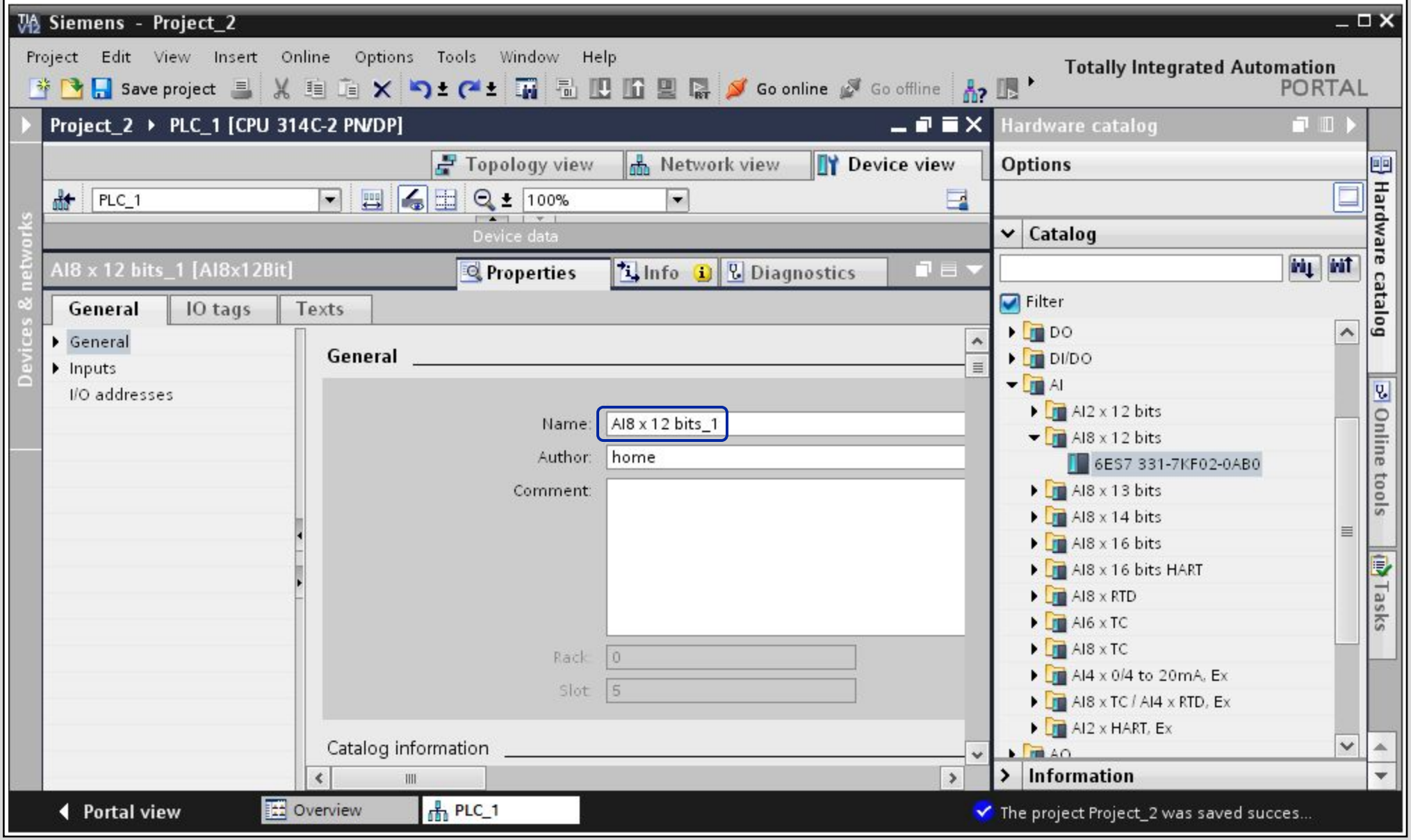
Выбор и параметрирование аналоговых модулей 7KF0xx



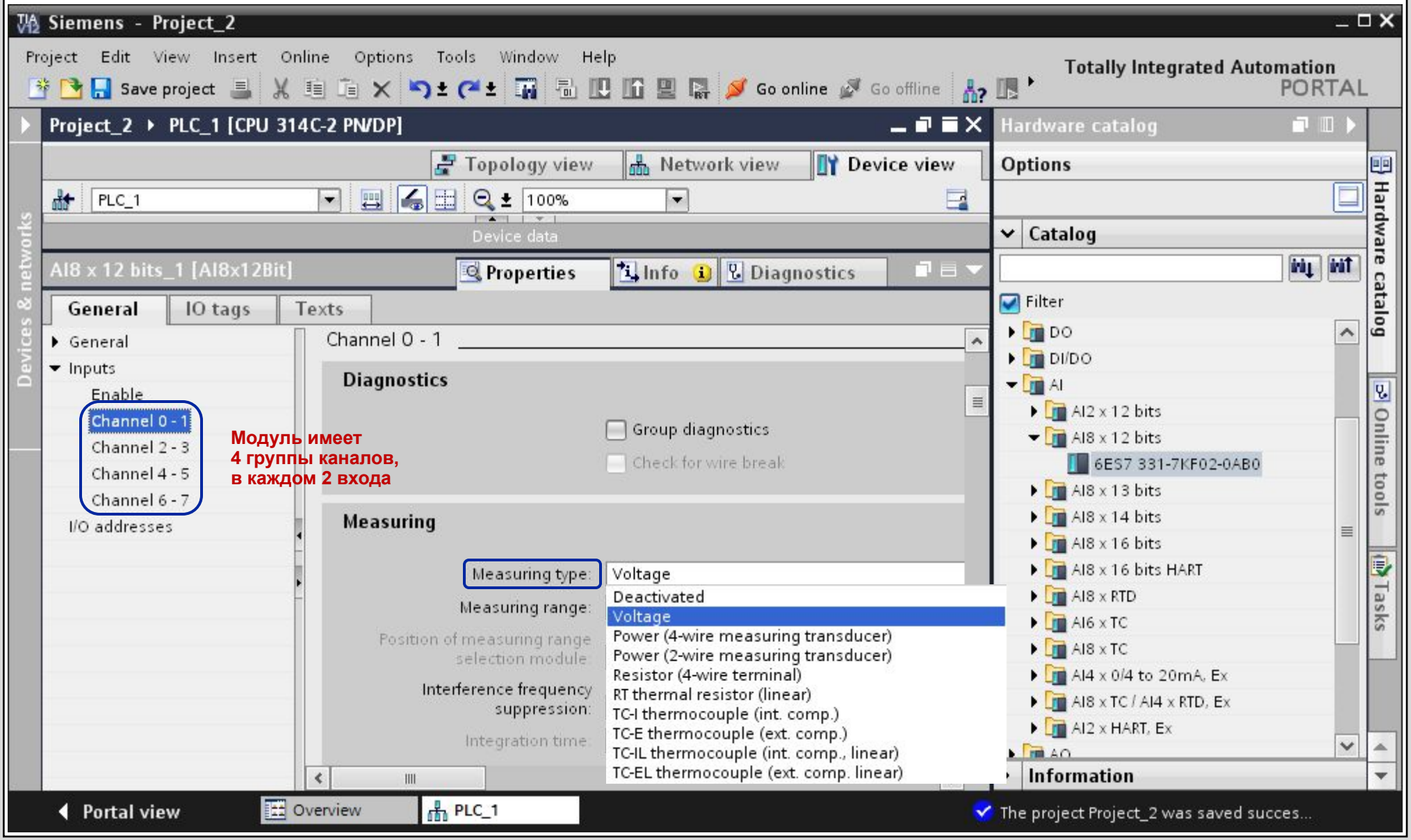
Выбор и параметрирование аналоговых модулей 7KF0xx



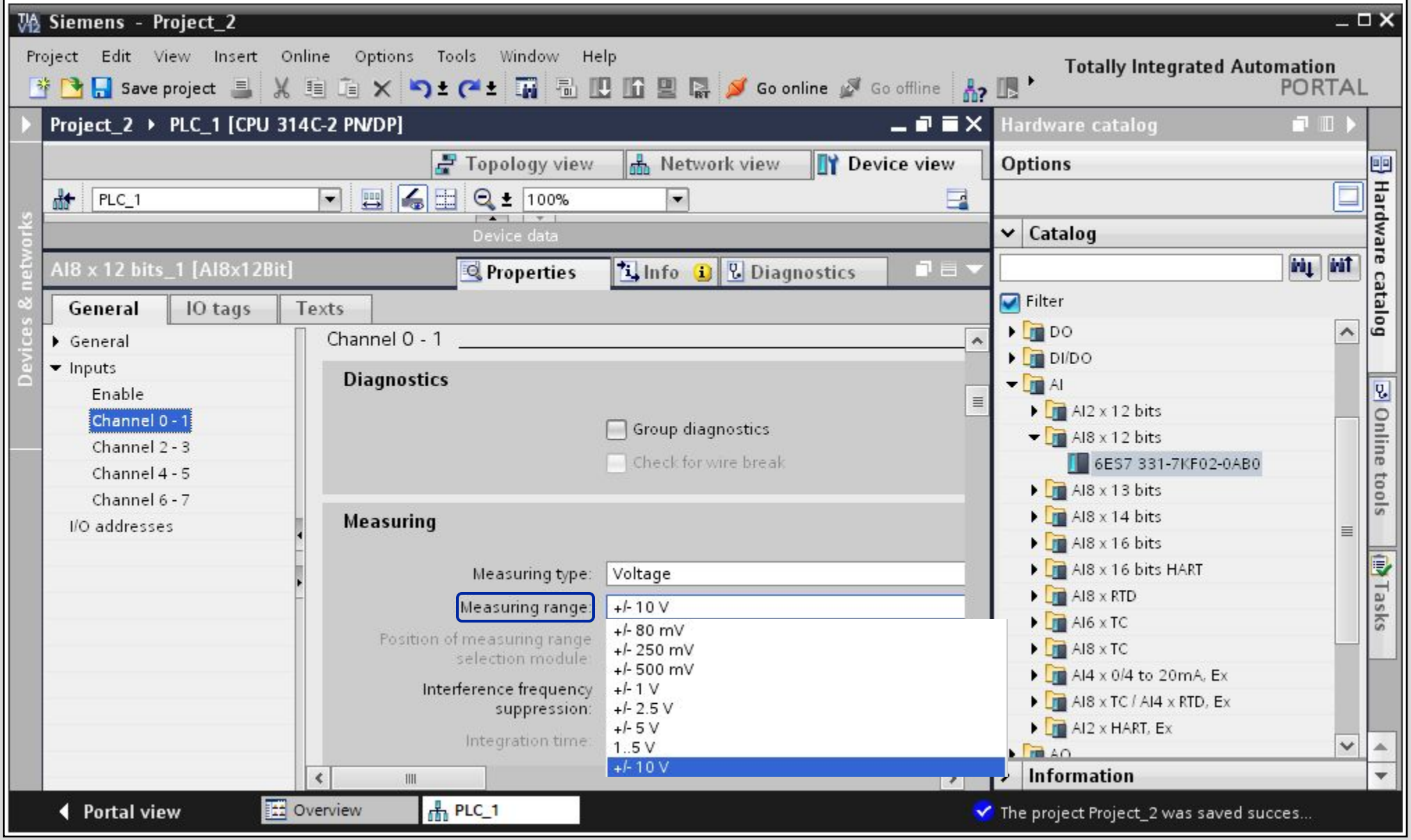
Выбор и параметрирование аналоговых модулей 7KF0xx



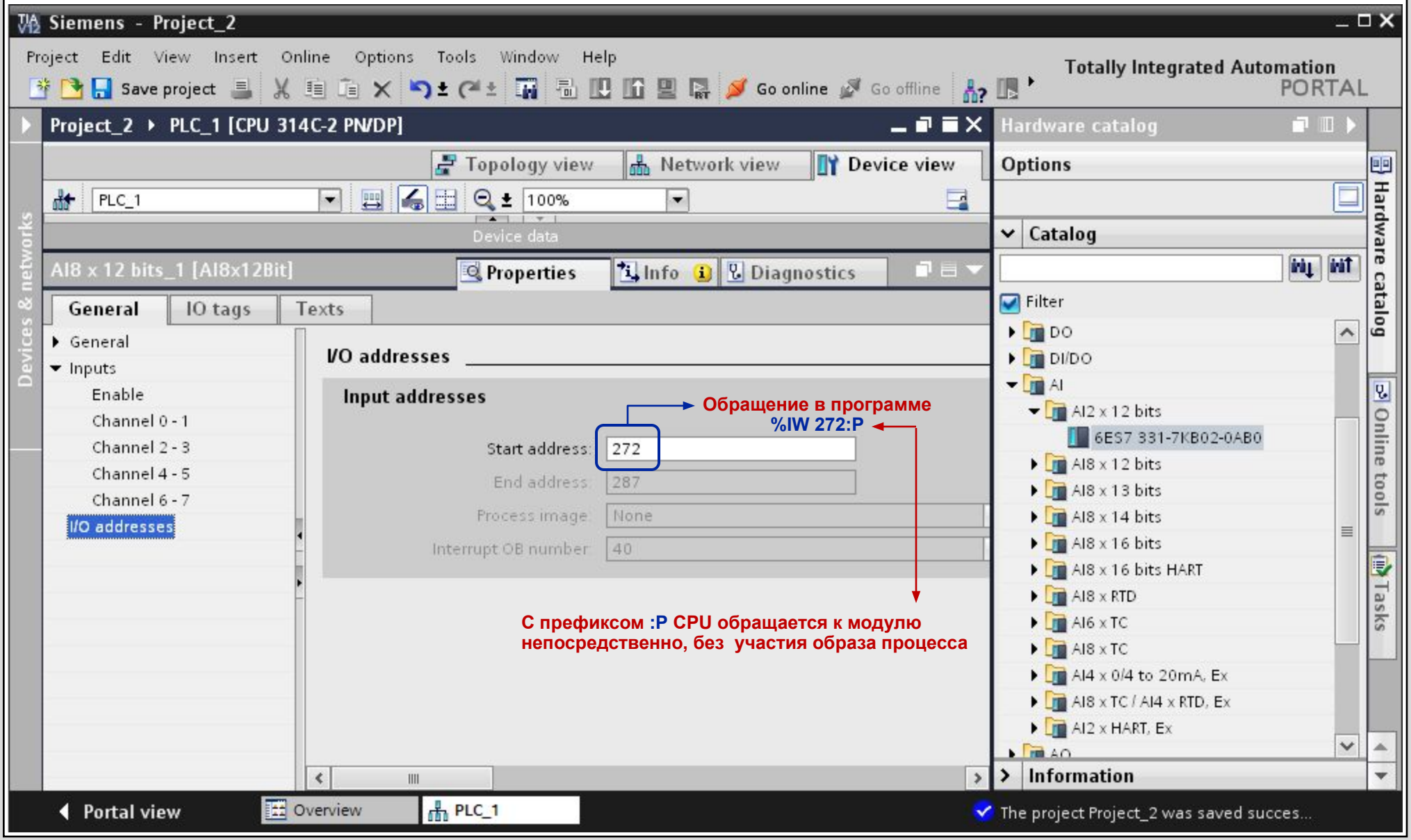
Выбор и параметрирование аналоговых модулей 7KF0xx



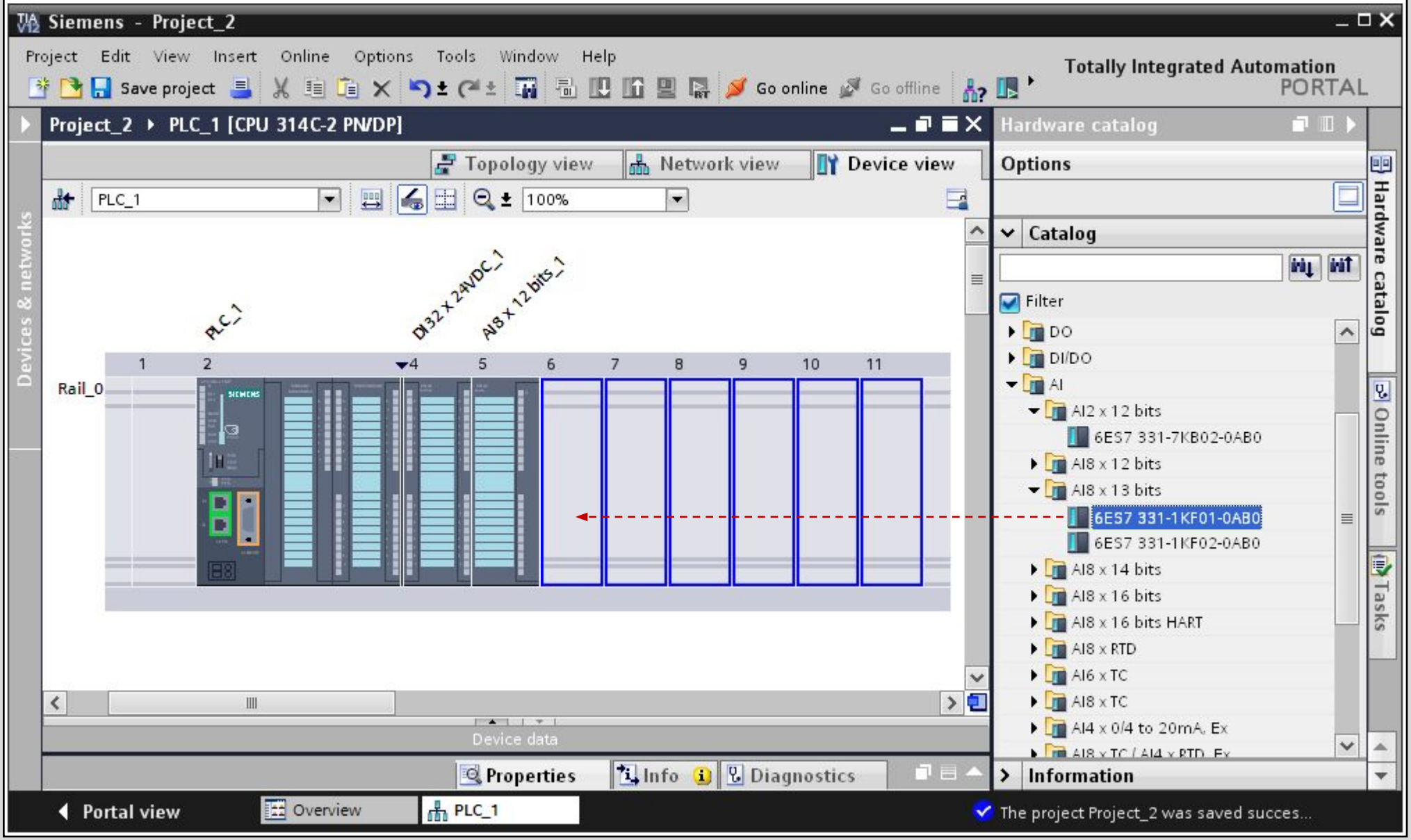
Выбор и параметрирование аналоговых модулей 7KF0xx



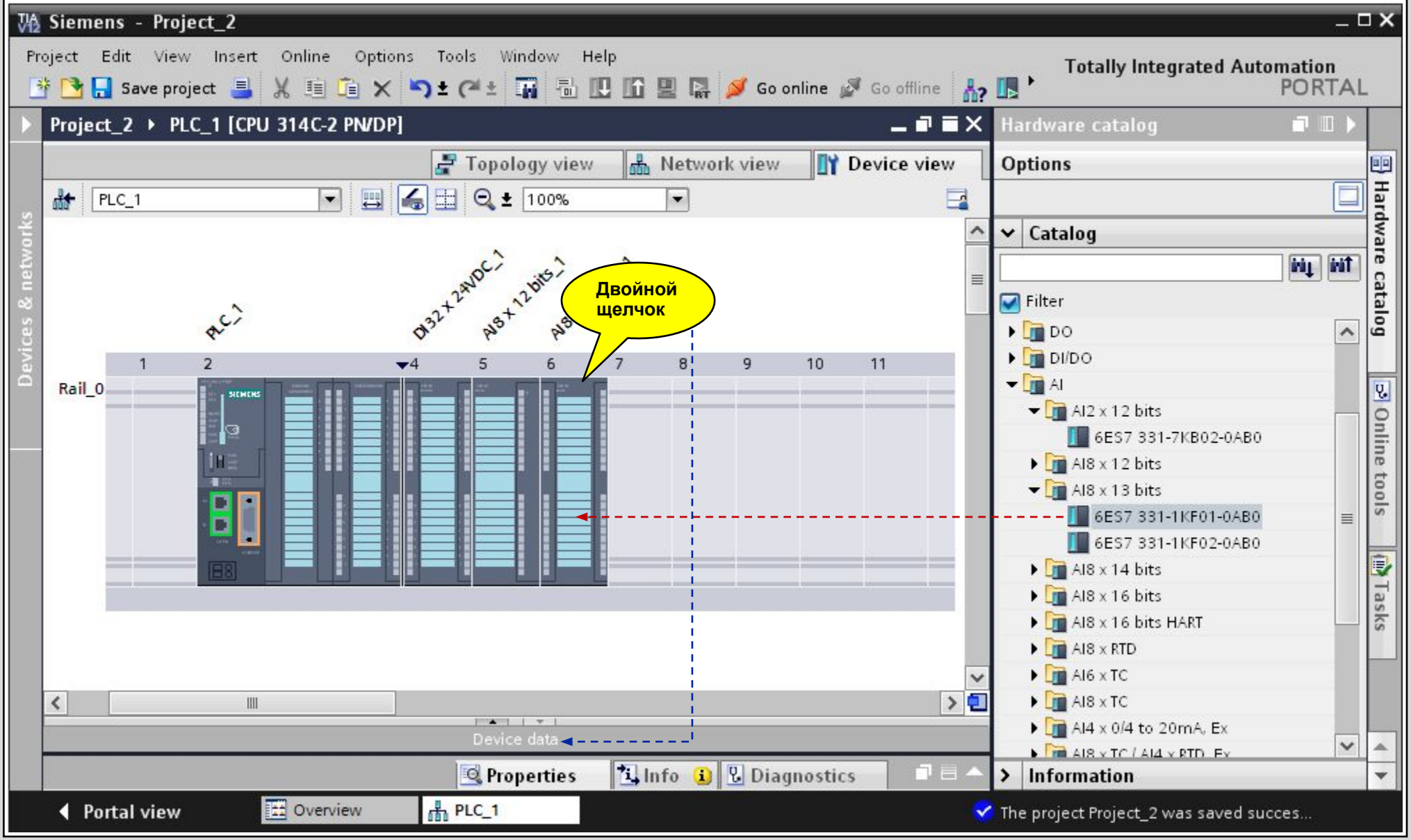
Выбор и параметрирование аналоговых модулей 7KF0xx



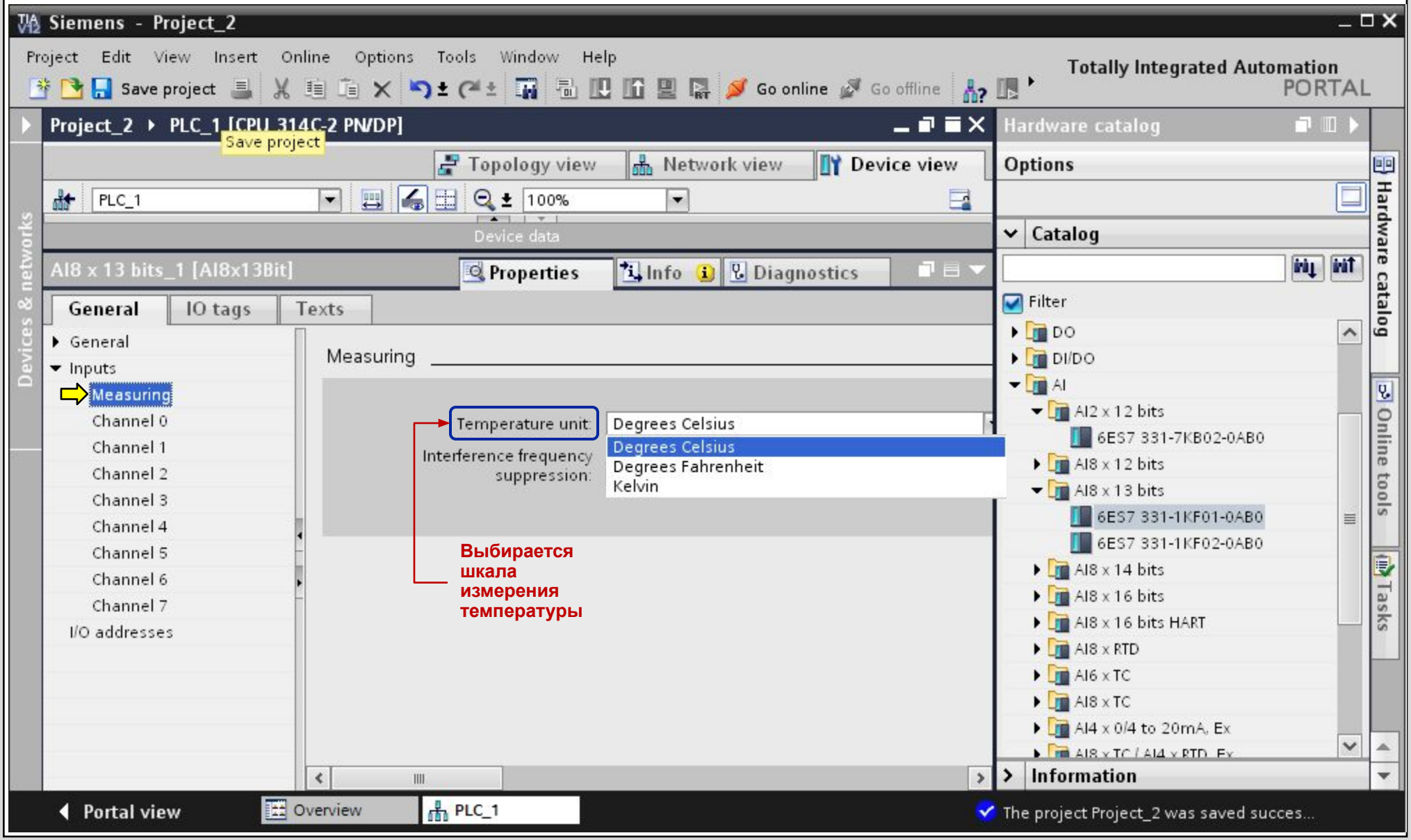
Выбор и параметрирование аналоговых модулей 1KF0xx



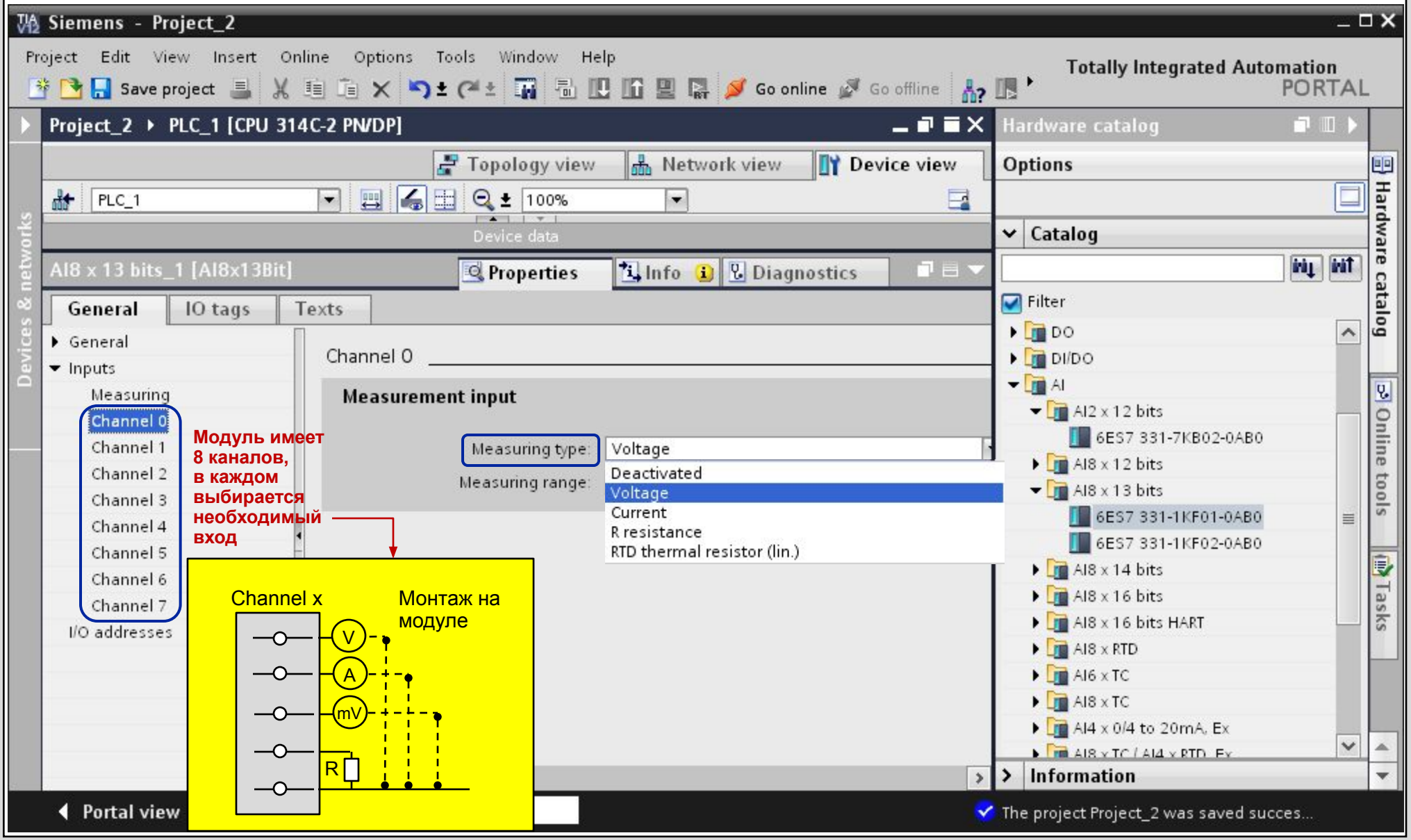
Выбор и параметрирование аналоговых модулей 1KF0xx



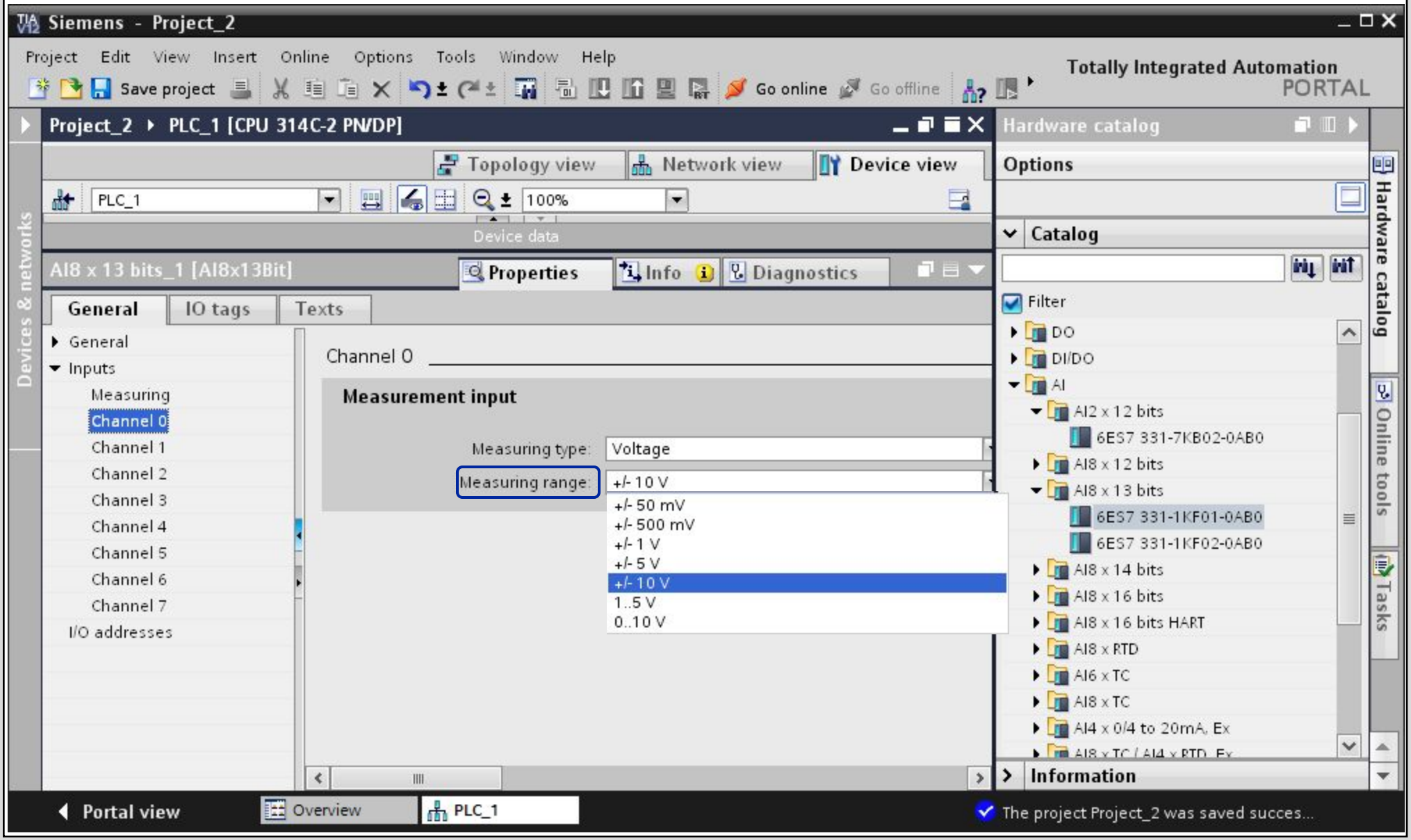
Выбор и параметрирование аналоговых модулей 1KF0xx



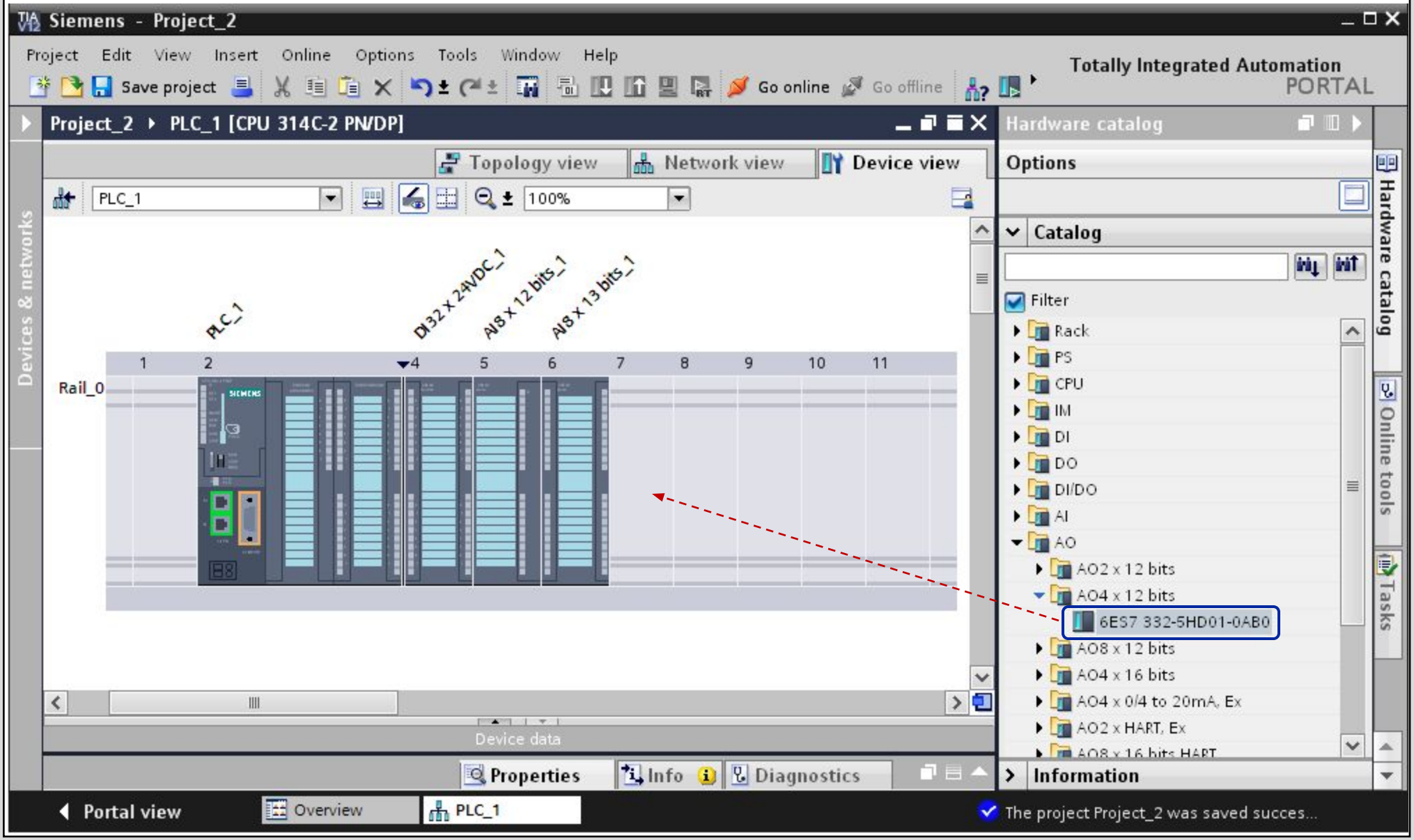
Выбор и параметрирование аналоговых модулей 1KF0xx



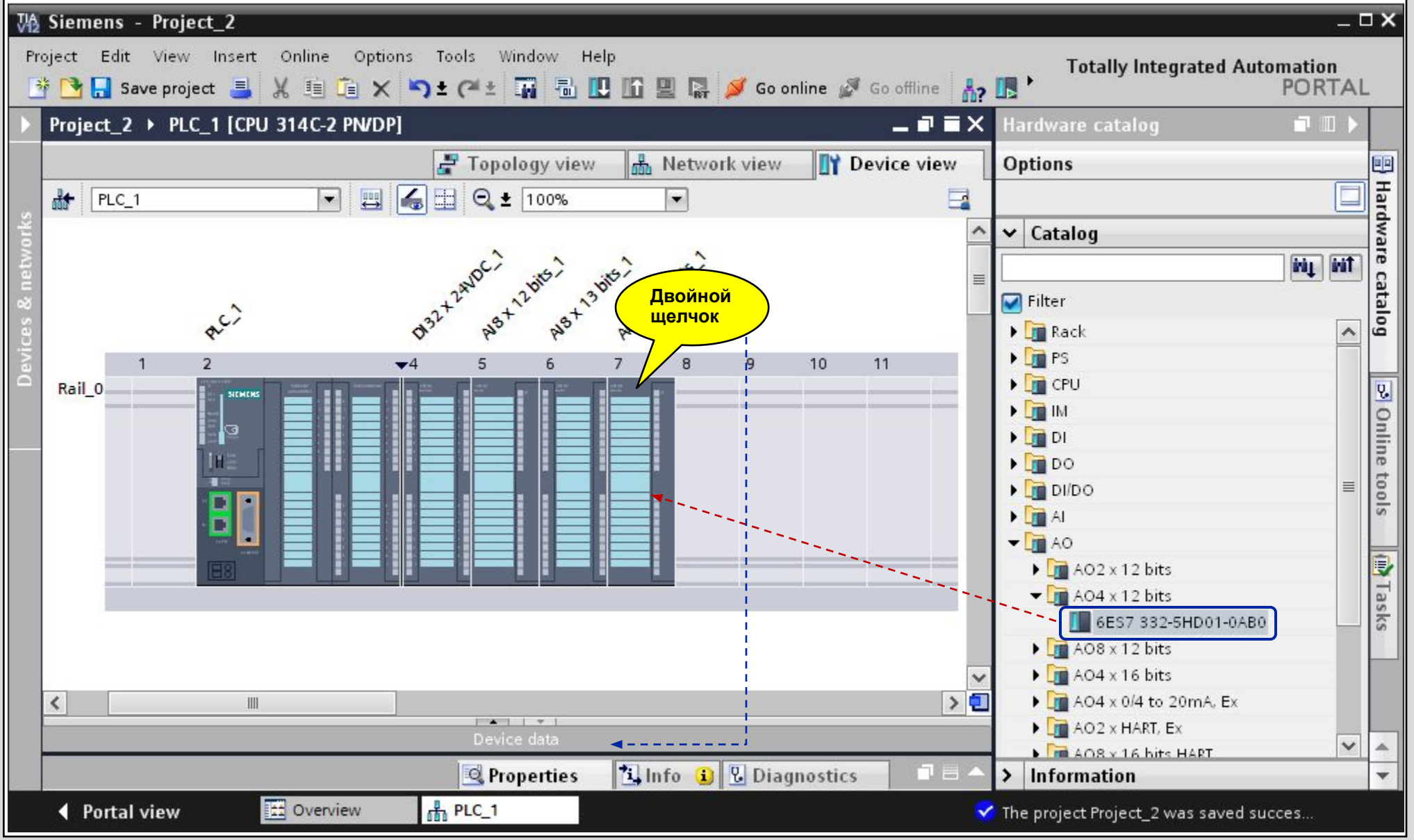
Выбор и параметрирование аналоговых модулей 1KF0xx



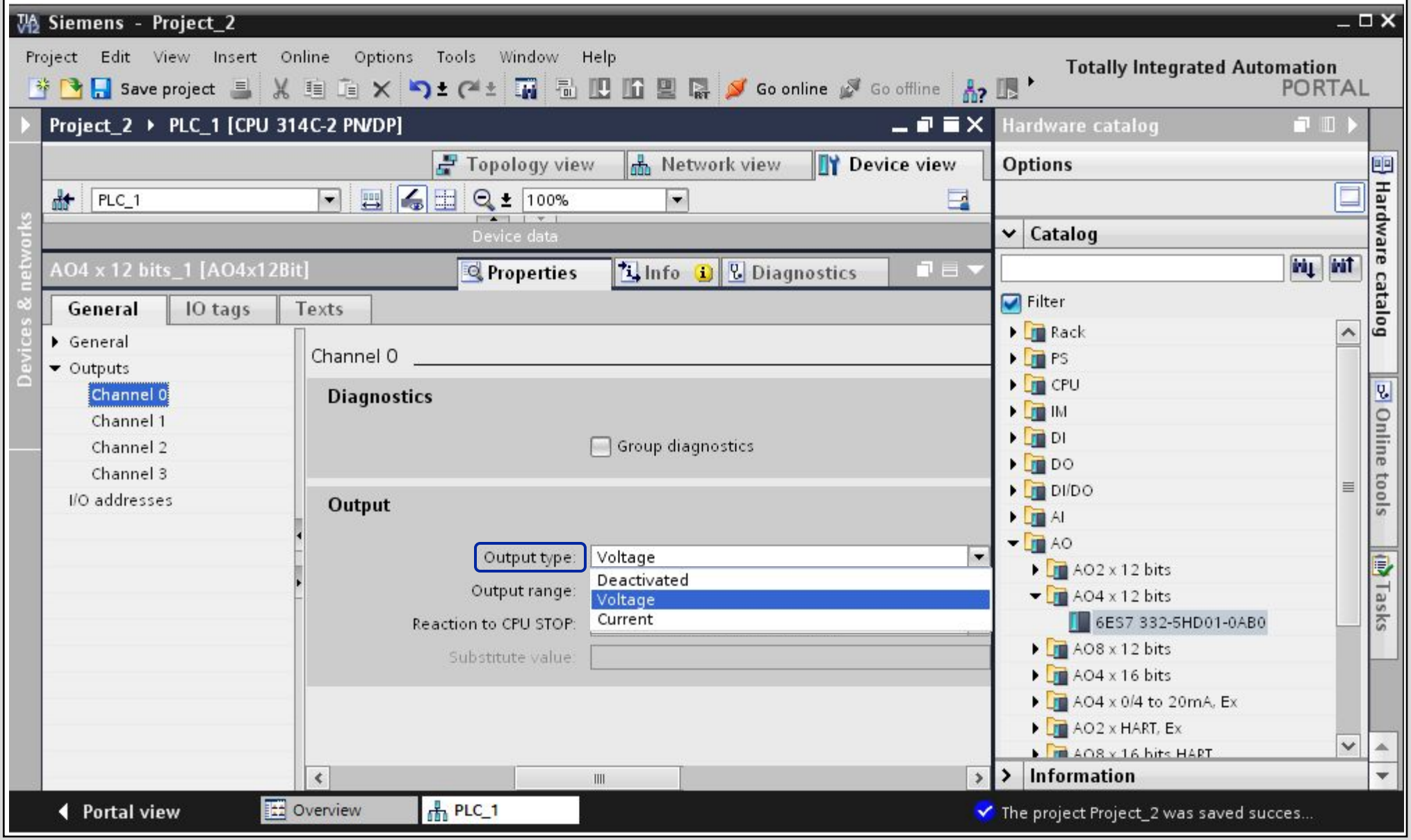
Выбор и параметрирование аналоговых модулей AO



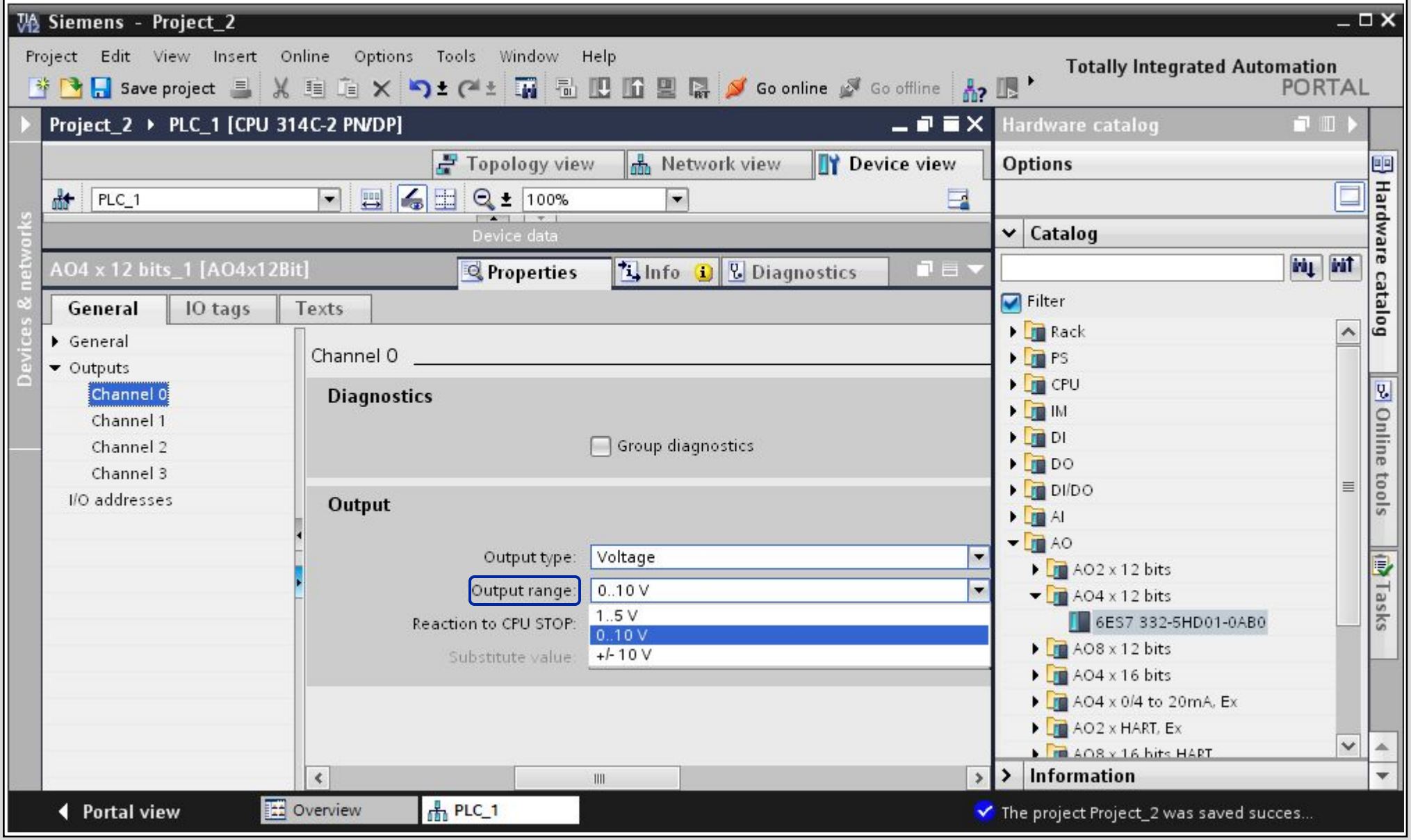
Выбор и параметрирование аналоговых модулей AO



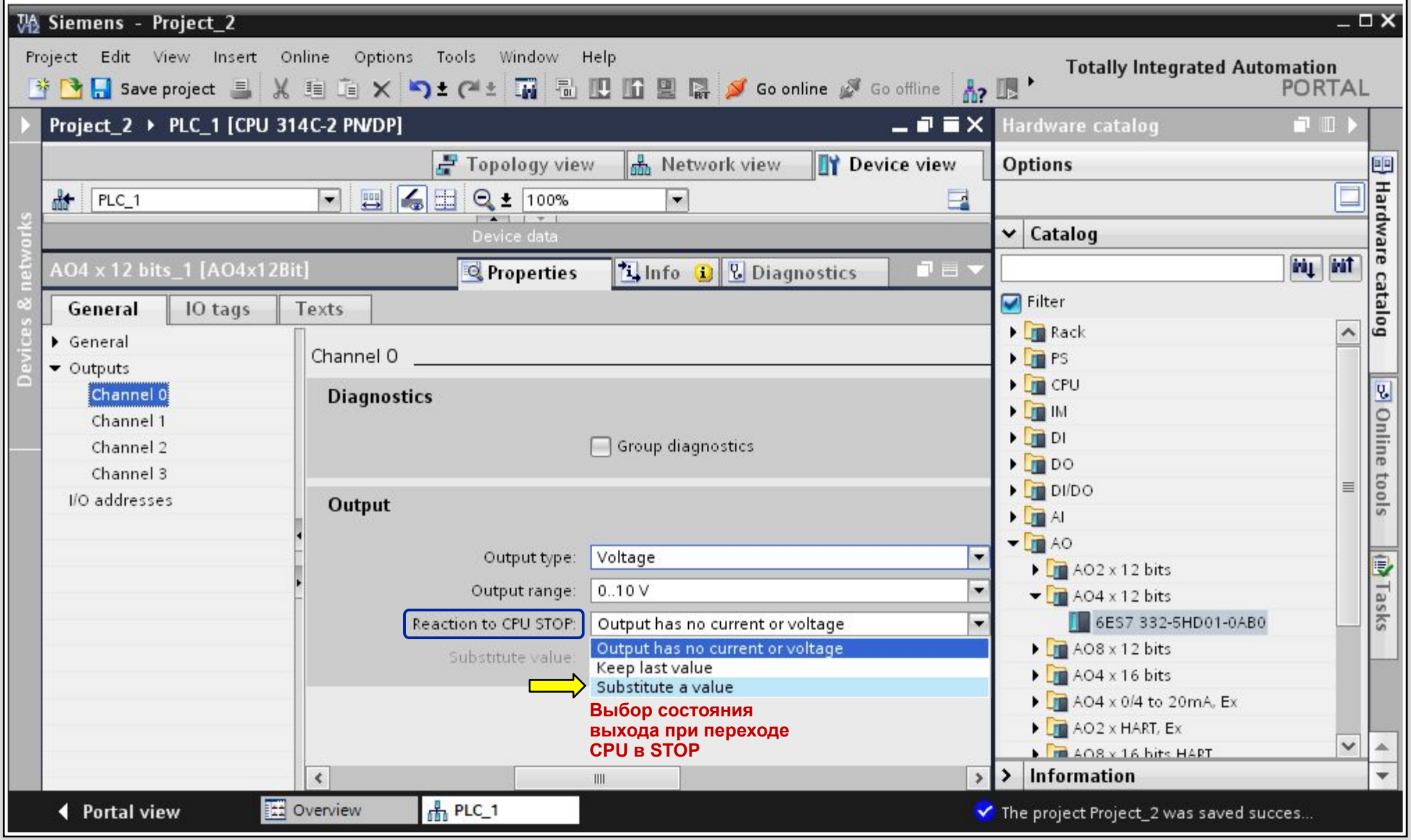
Выбор и параметрирование аналоговых модулей AO



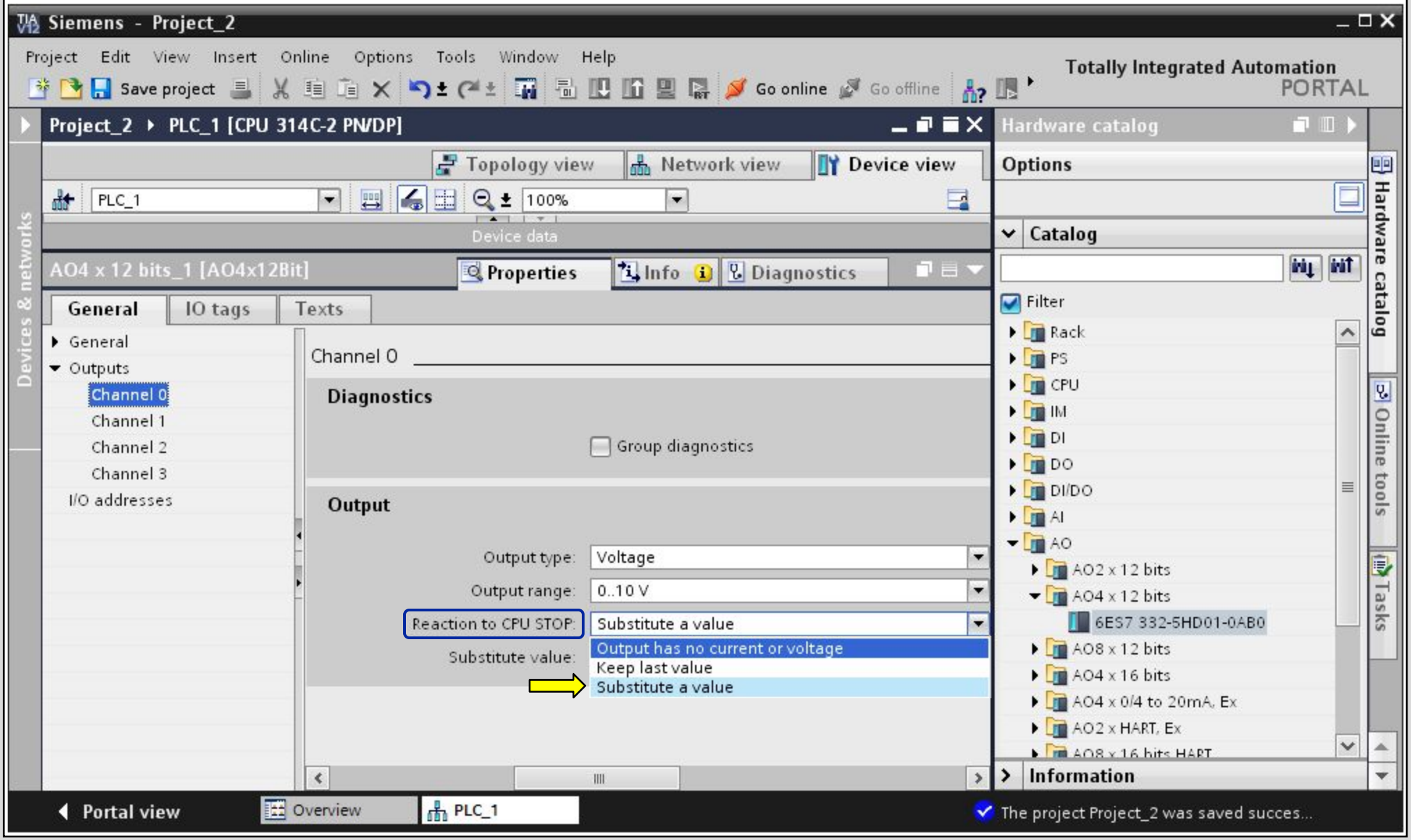
Выбор и параметрирование аналоговых модулей AO



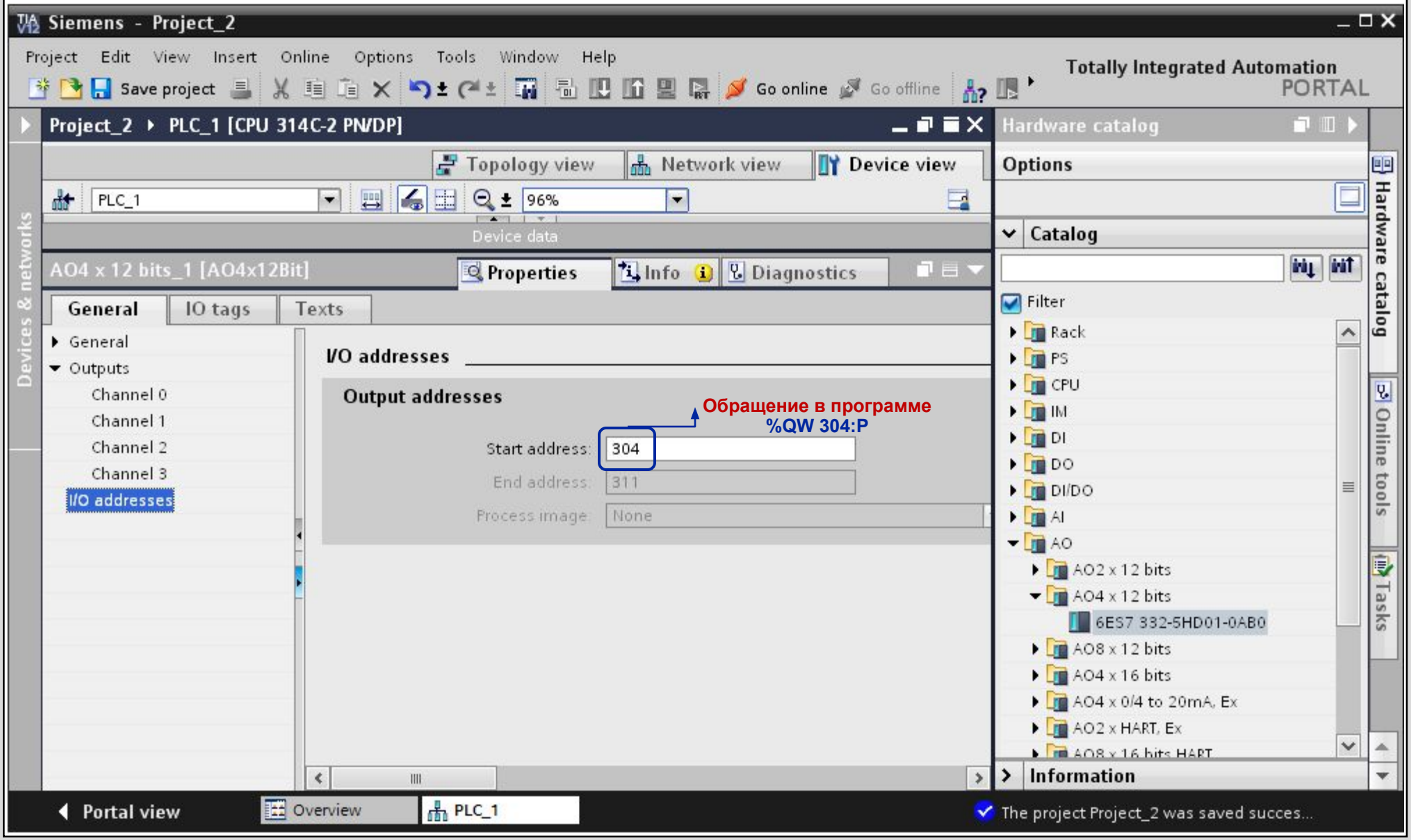
Выбор и параметрирование аналоговых модулей AO



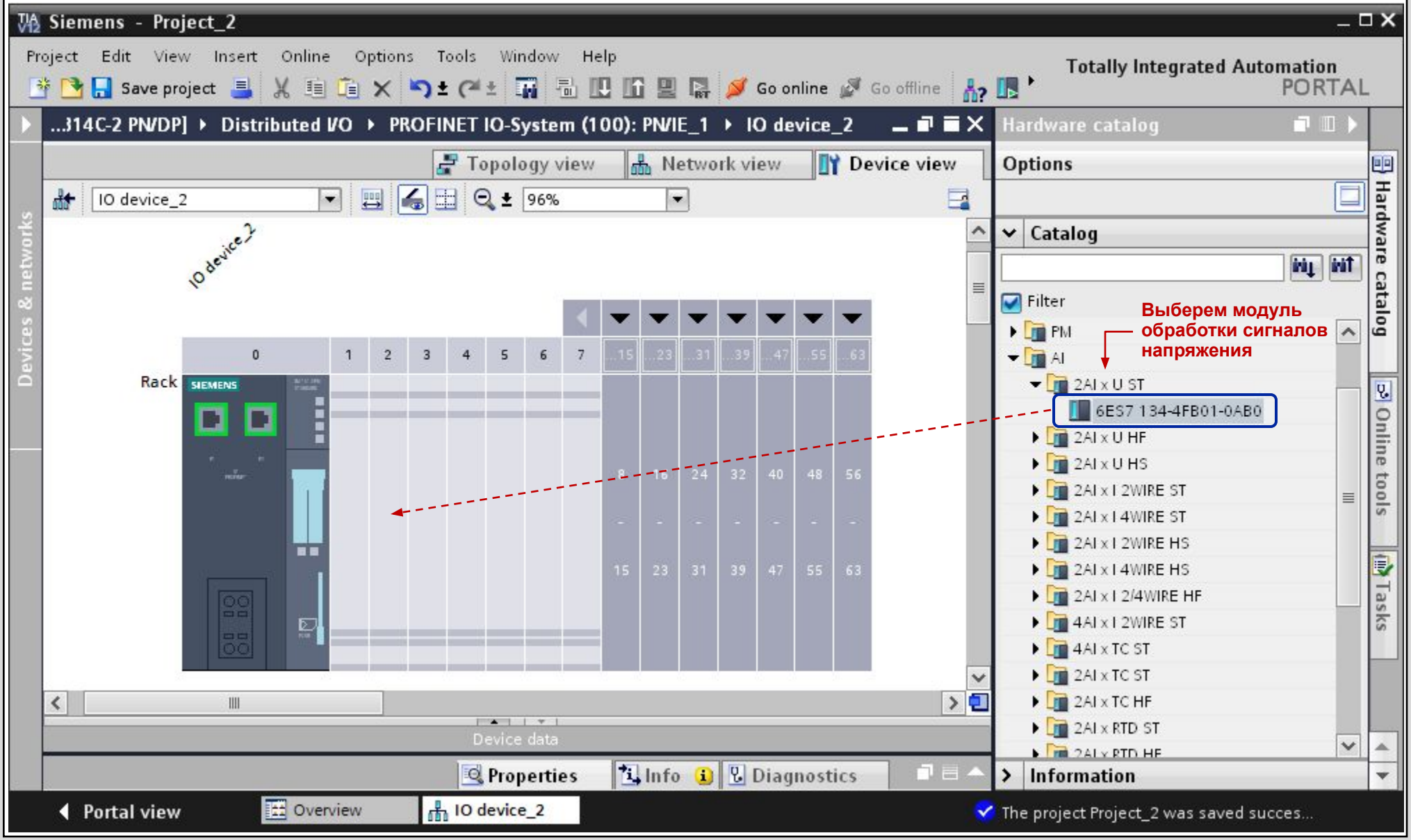
Выбор и параметрирование аналоговых модулей AO



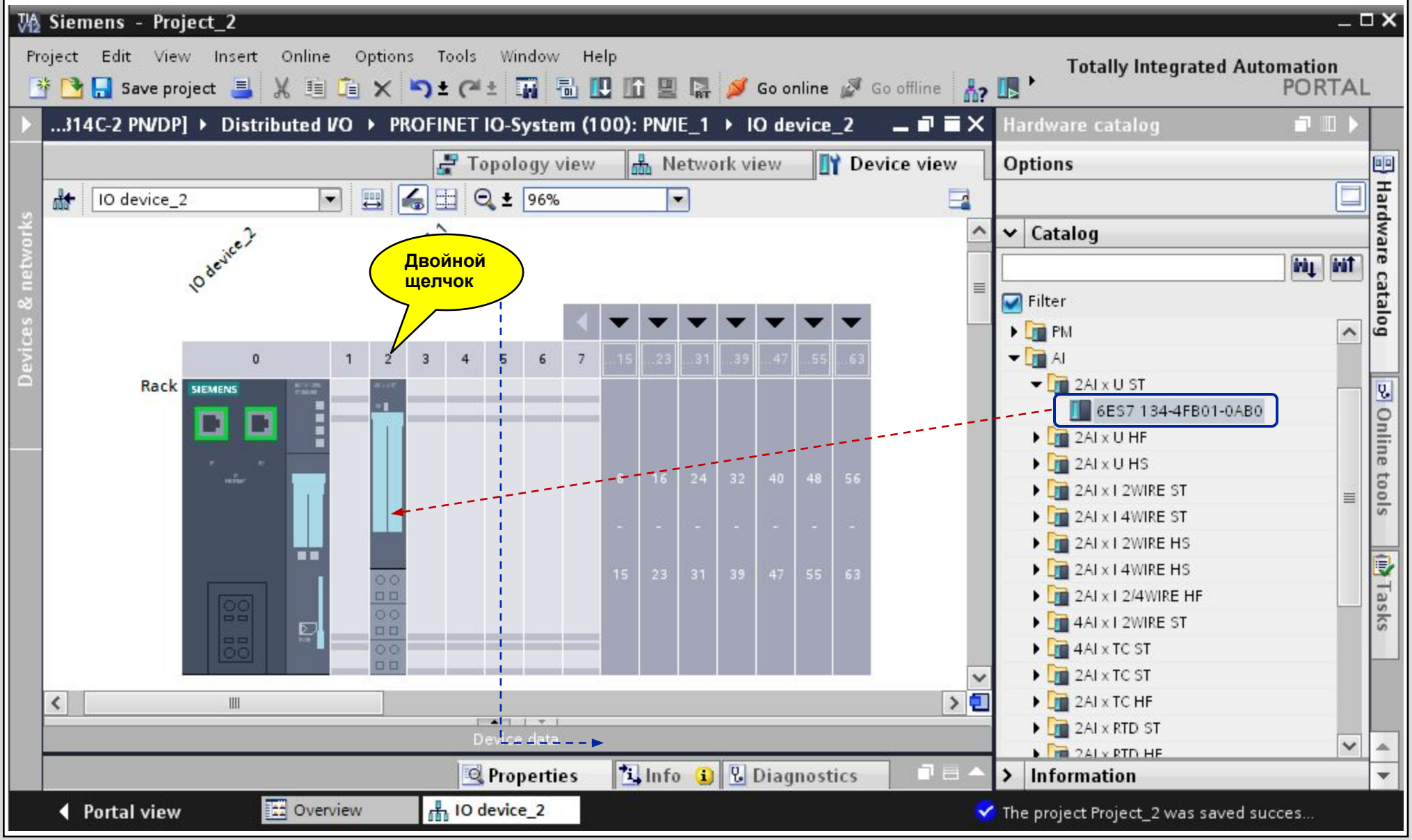
Выбор и параметрирование аналоговых модулей AO



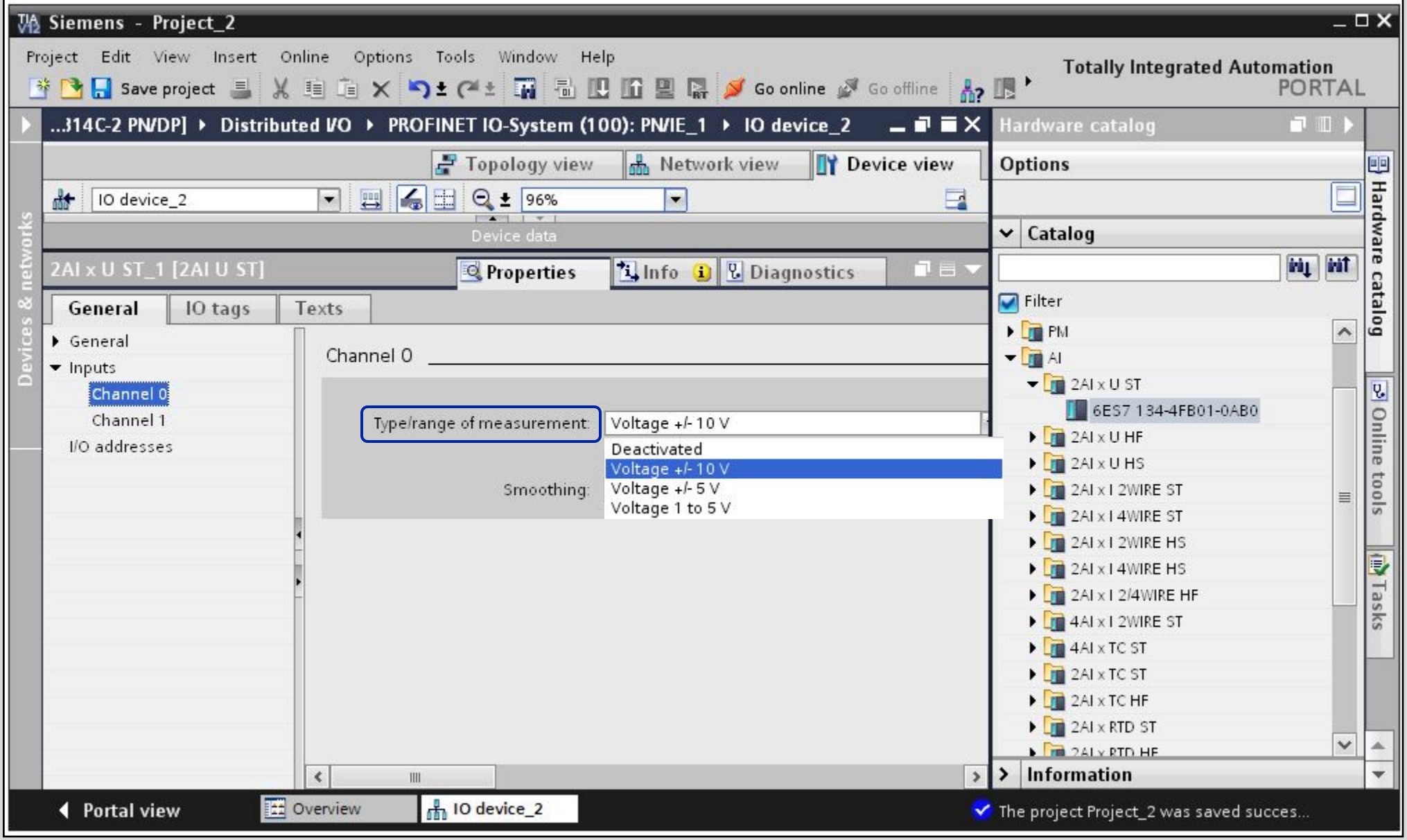
Выбор и параметрирование аналоговых модулей AI в ET200S



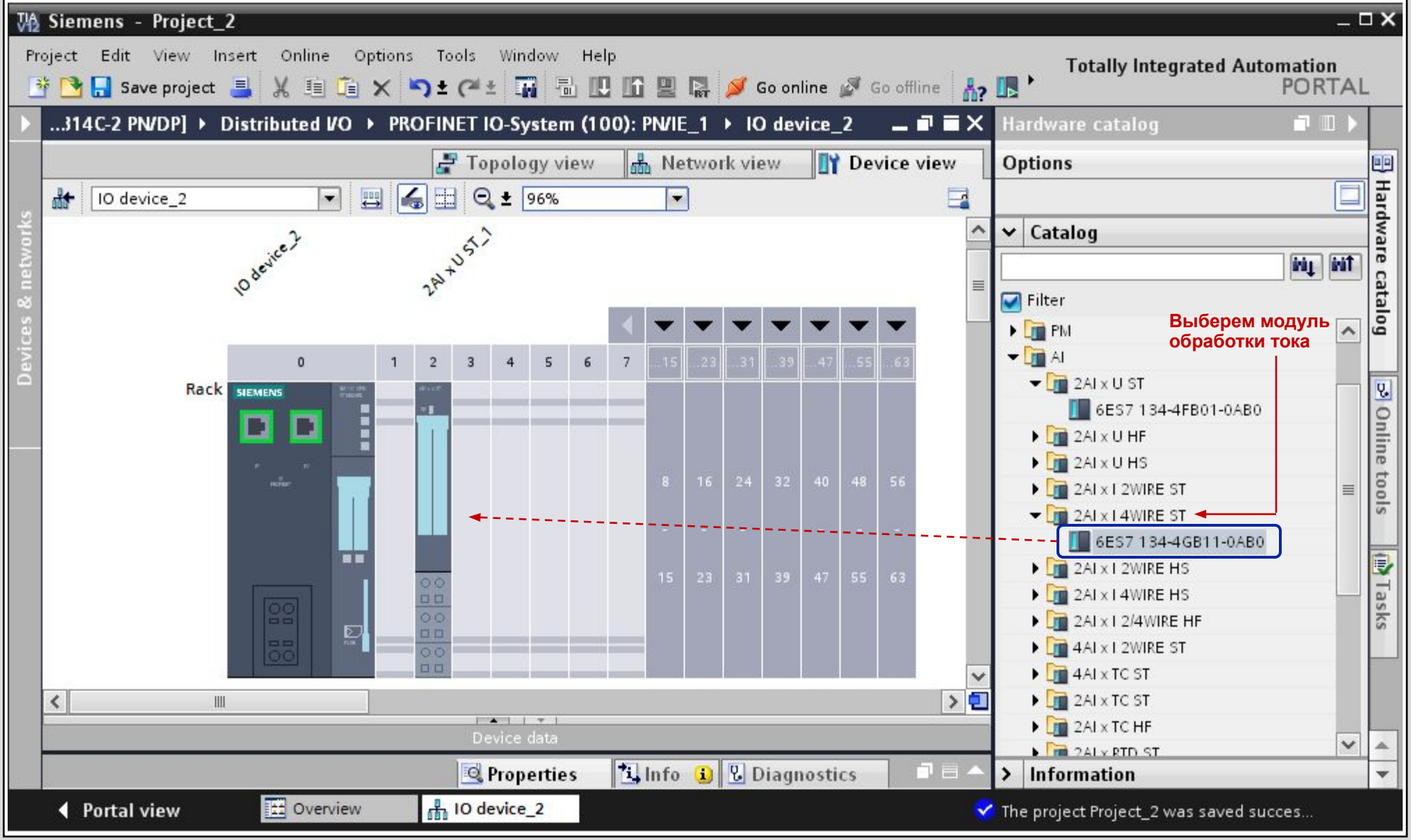
Выбор и параметрирование аналоговых модулей AI в ET200S



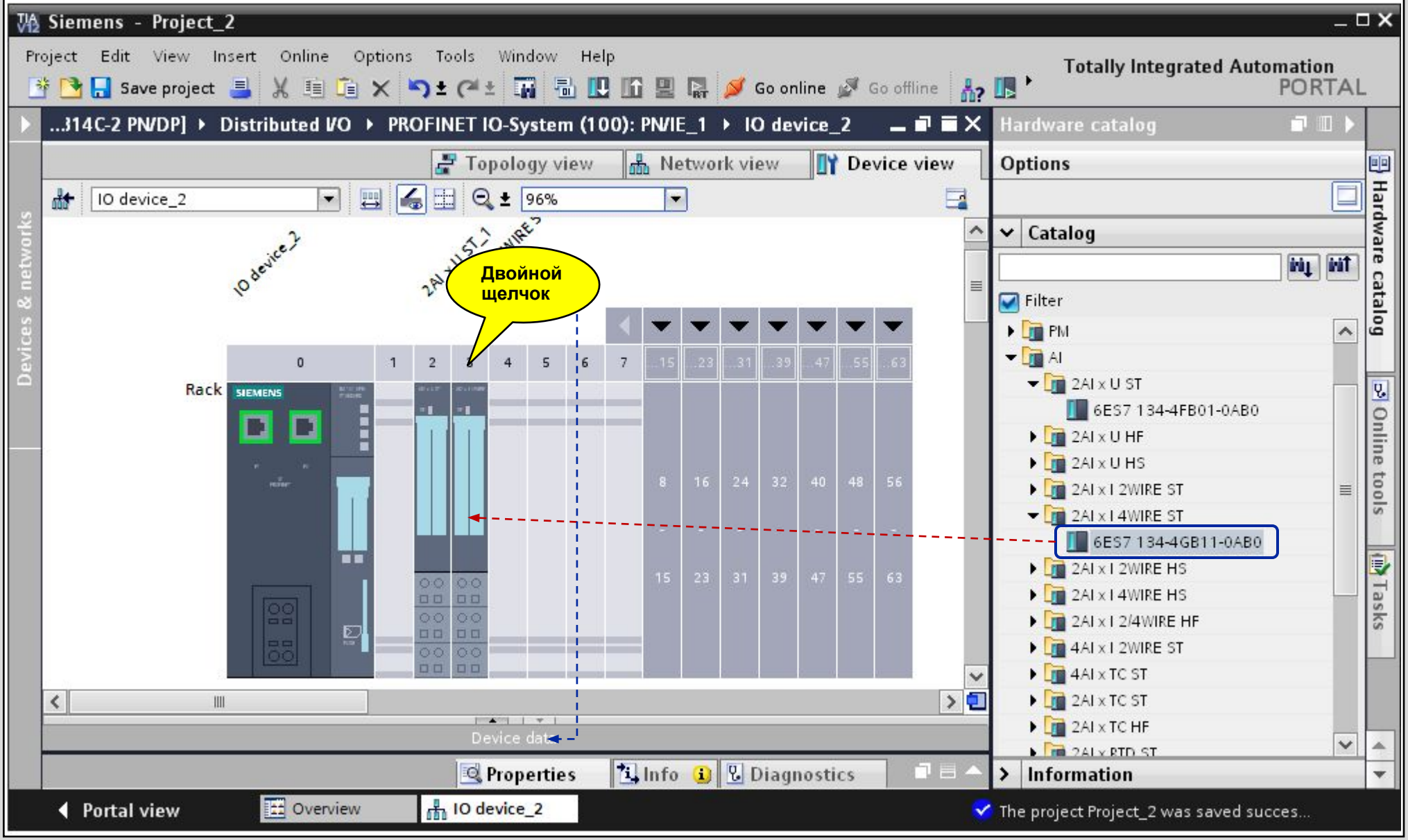
Выбор и параметрирование аналоговых модулей AI в ET200S



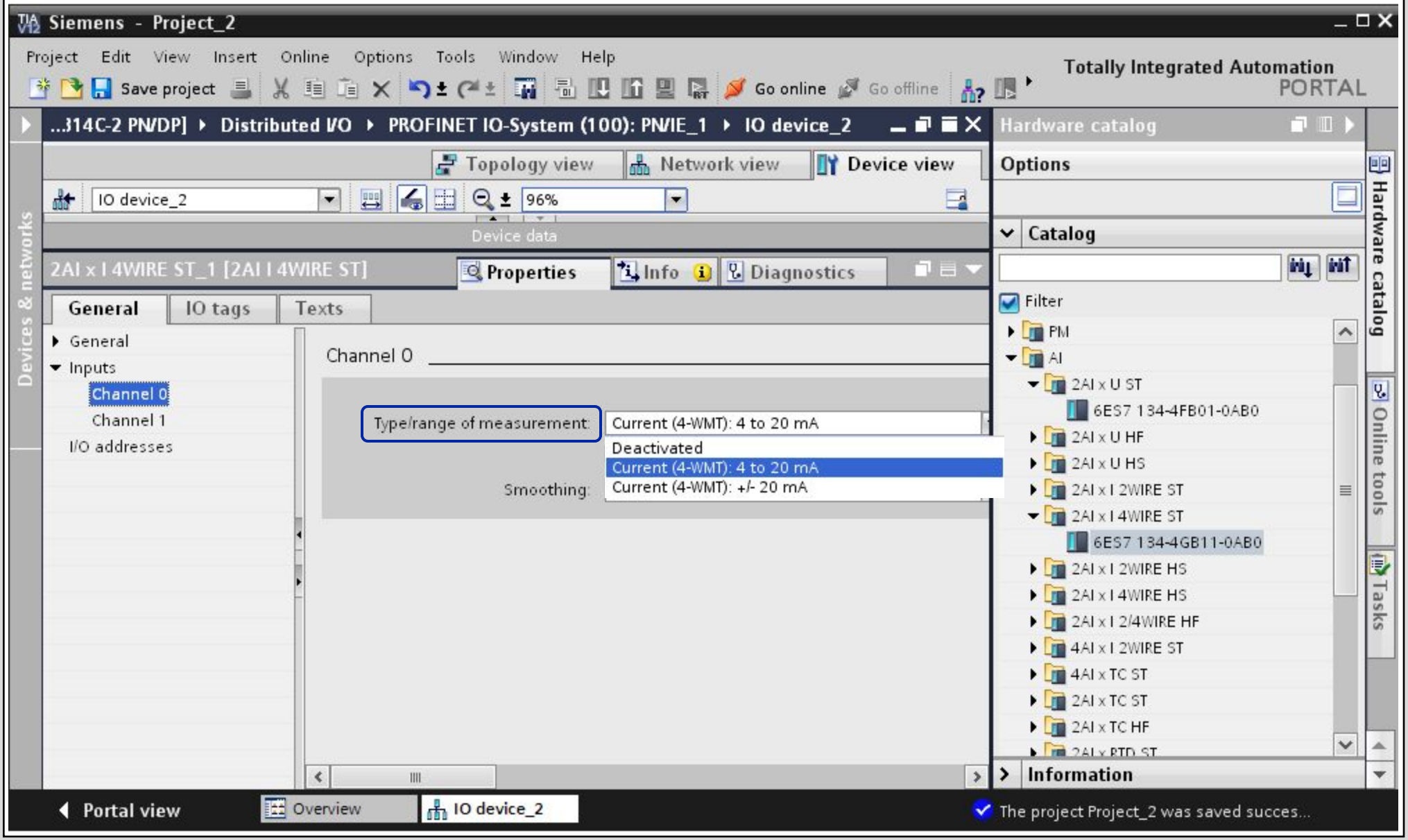
Выбор и параметрирование аналоговых модулей AI в ET200S



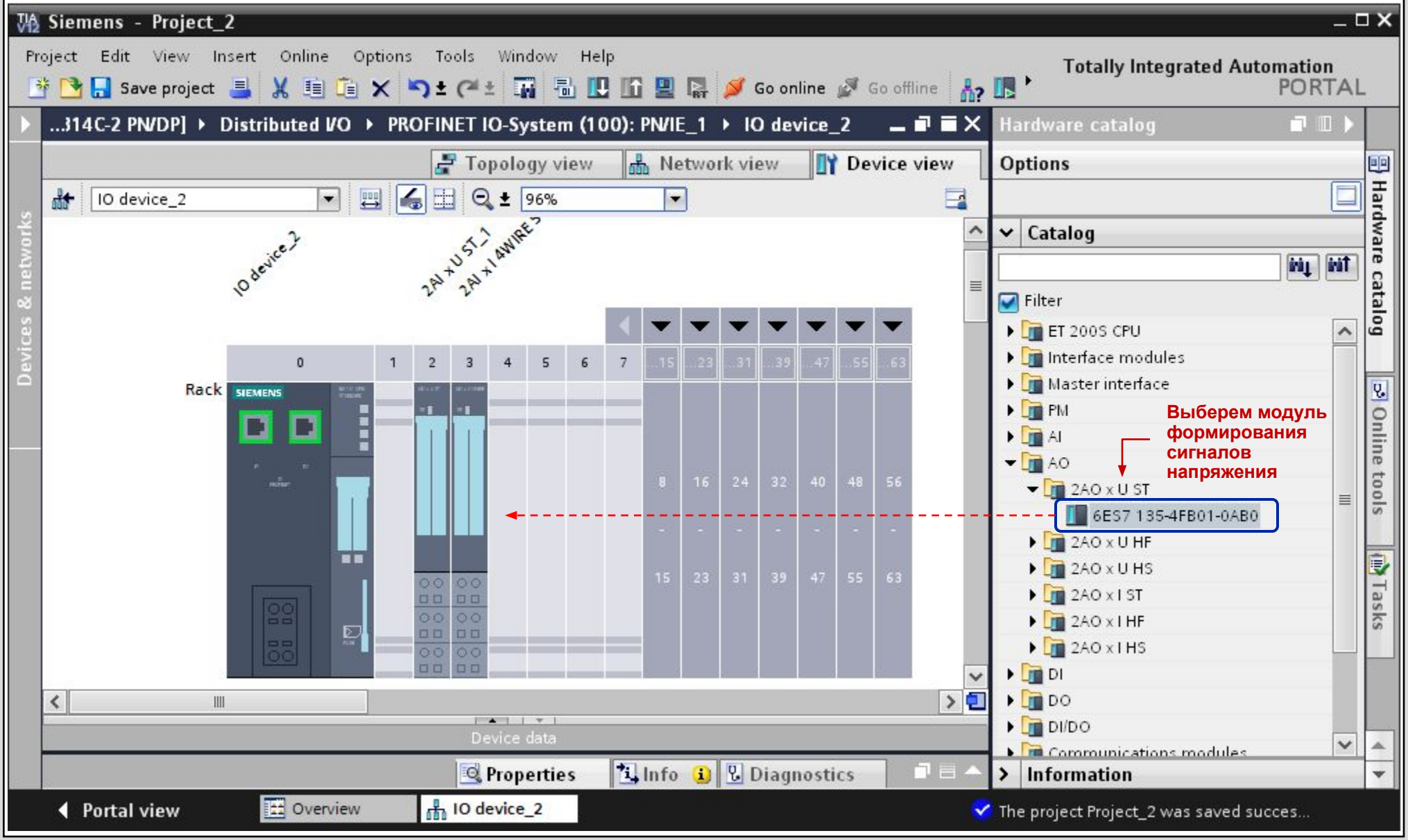
Выбор и параметрирование аналоговых модулей AI в ET200S



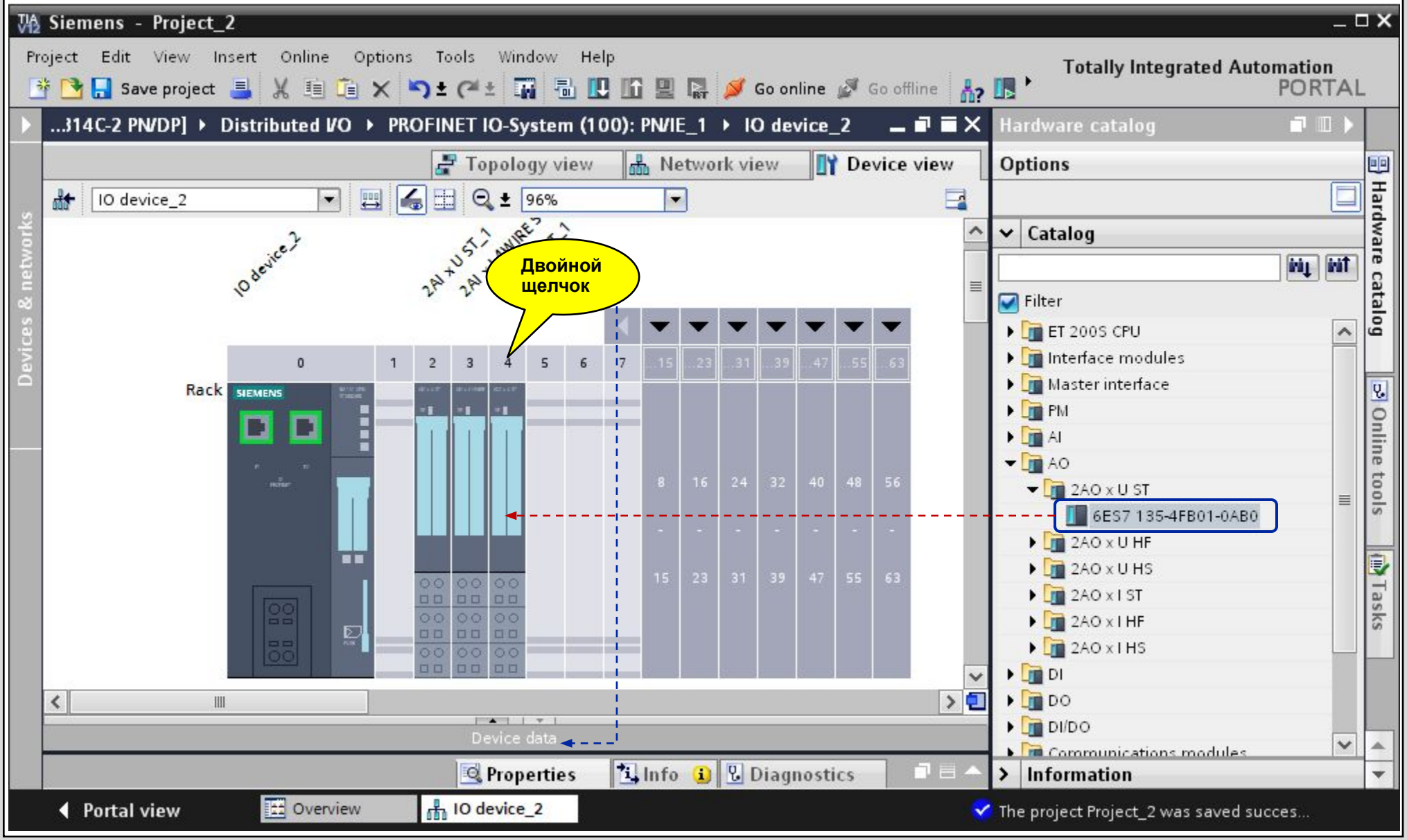
Выбор и параметрирование аналоговых модулей AI в ET200S



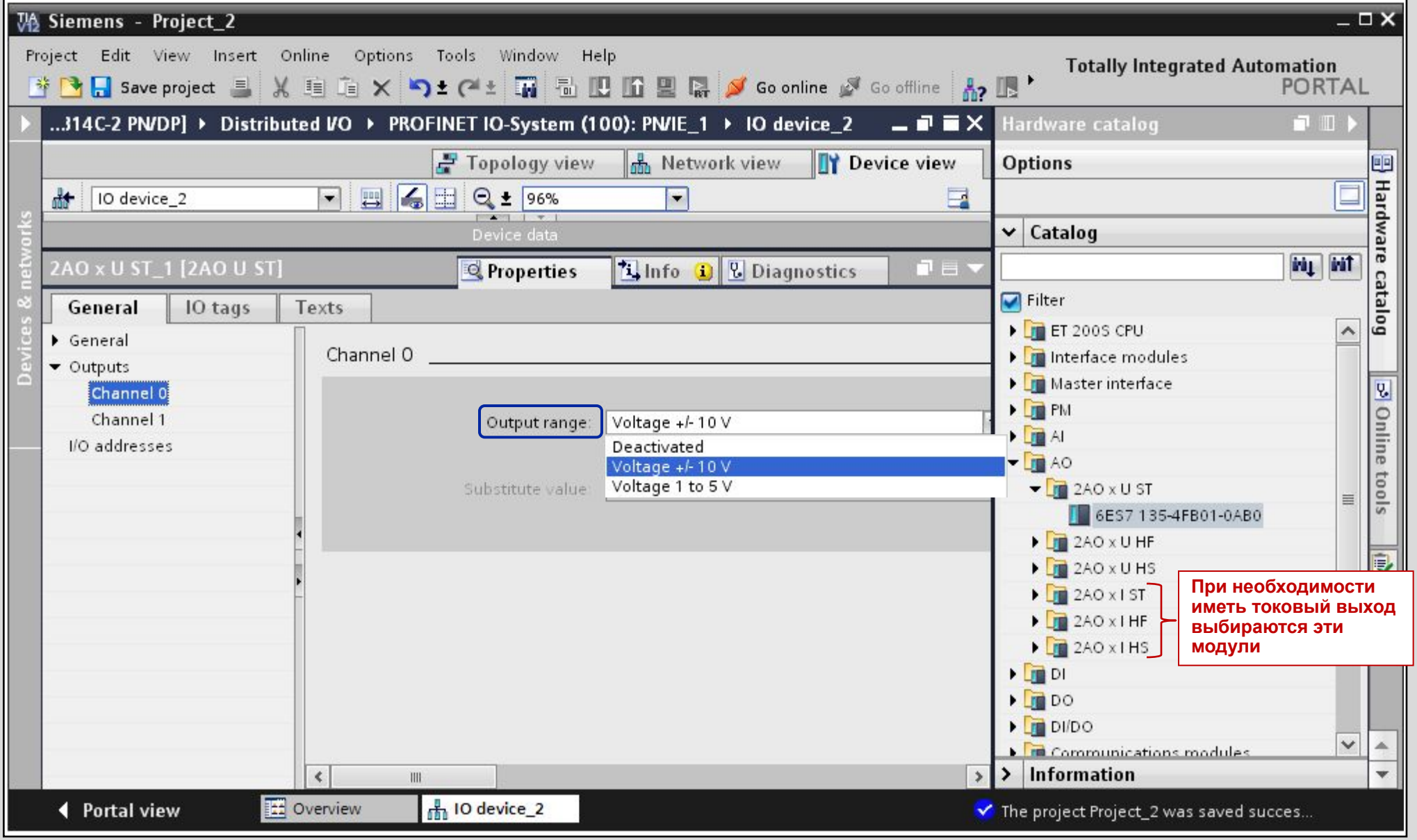
Выбор и параметрирование аналоговых модулей AO в ET200S



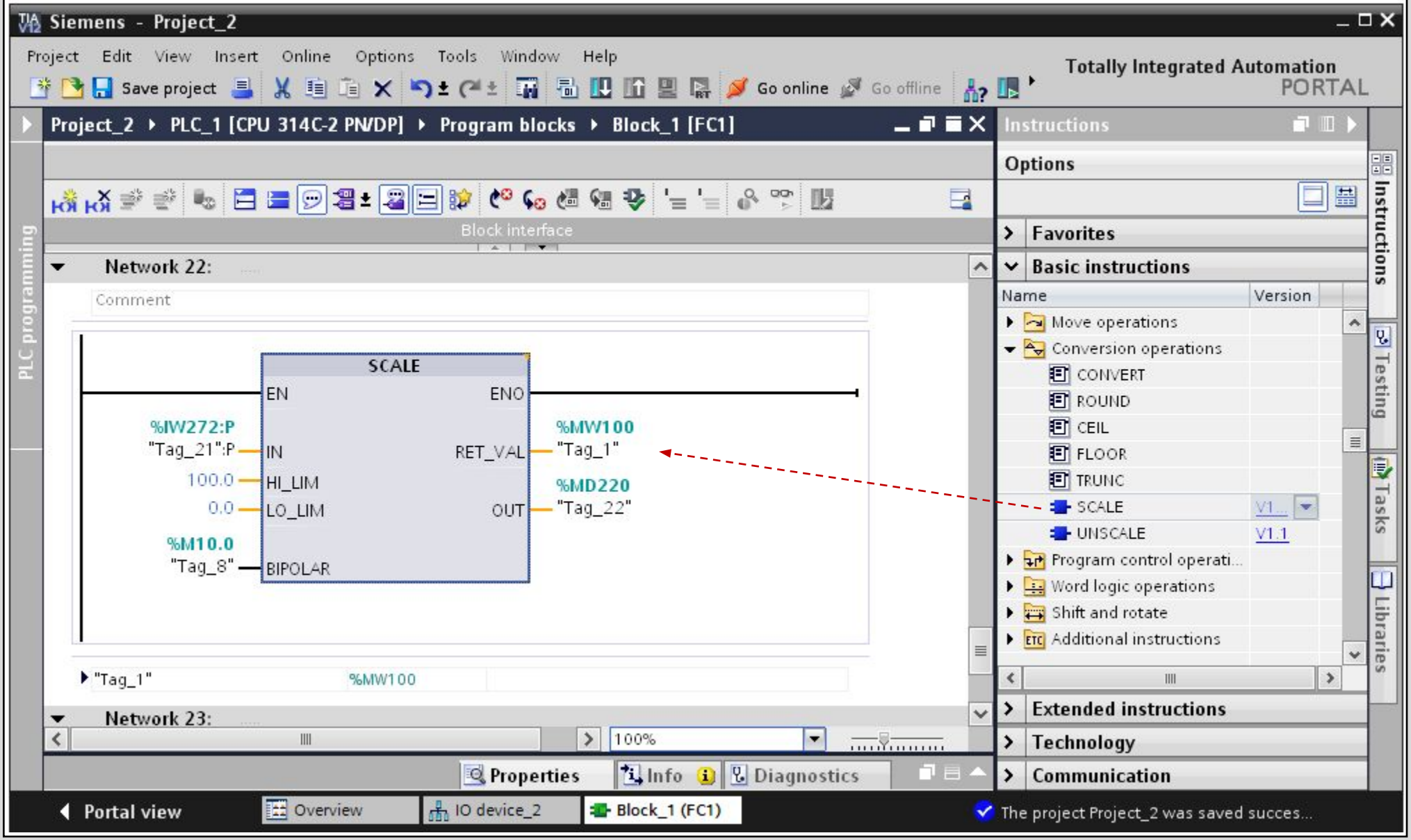
Выбор и параметрирование аналоговых модулей AO в ET200S



Выбор и параметрирование аналоговых модулей AO в ET200S

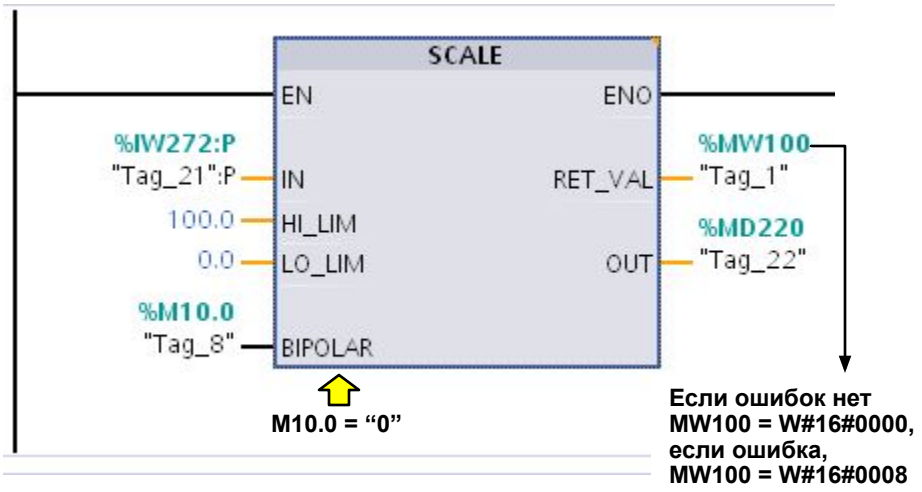


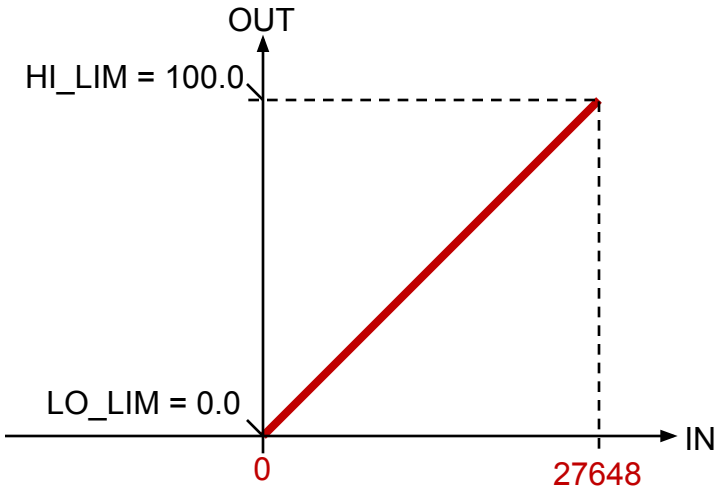
Чтение аналогового входа функцией SCALE



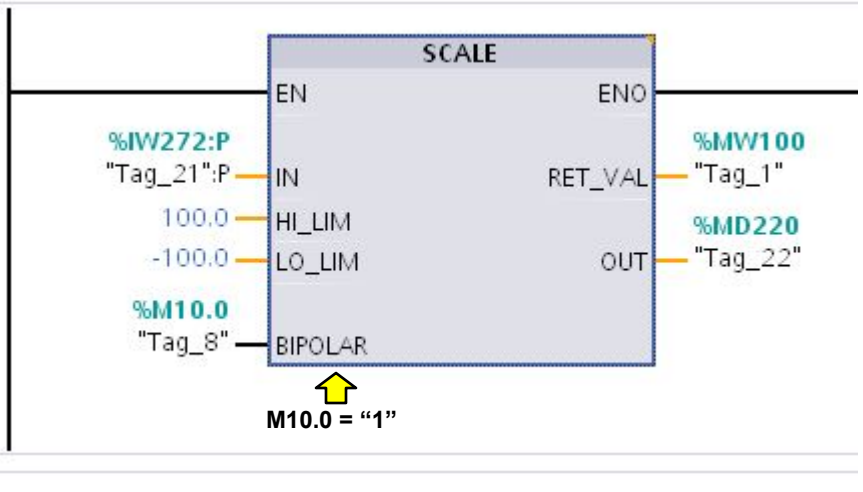
Чтение аналогового входа функцией SCALE

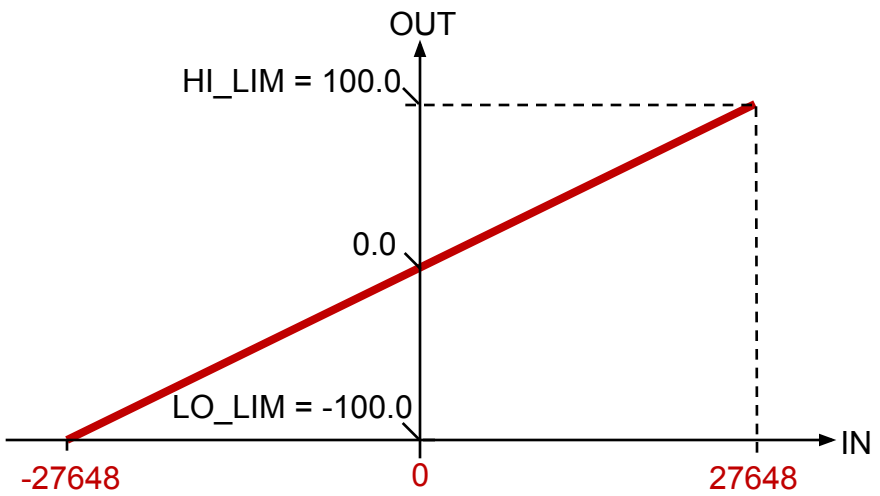
Датчик выдает только положительный сигнал



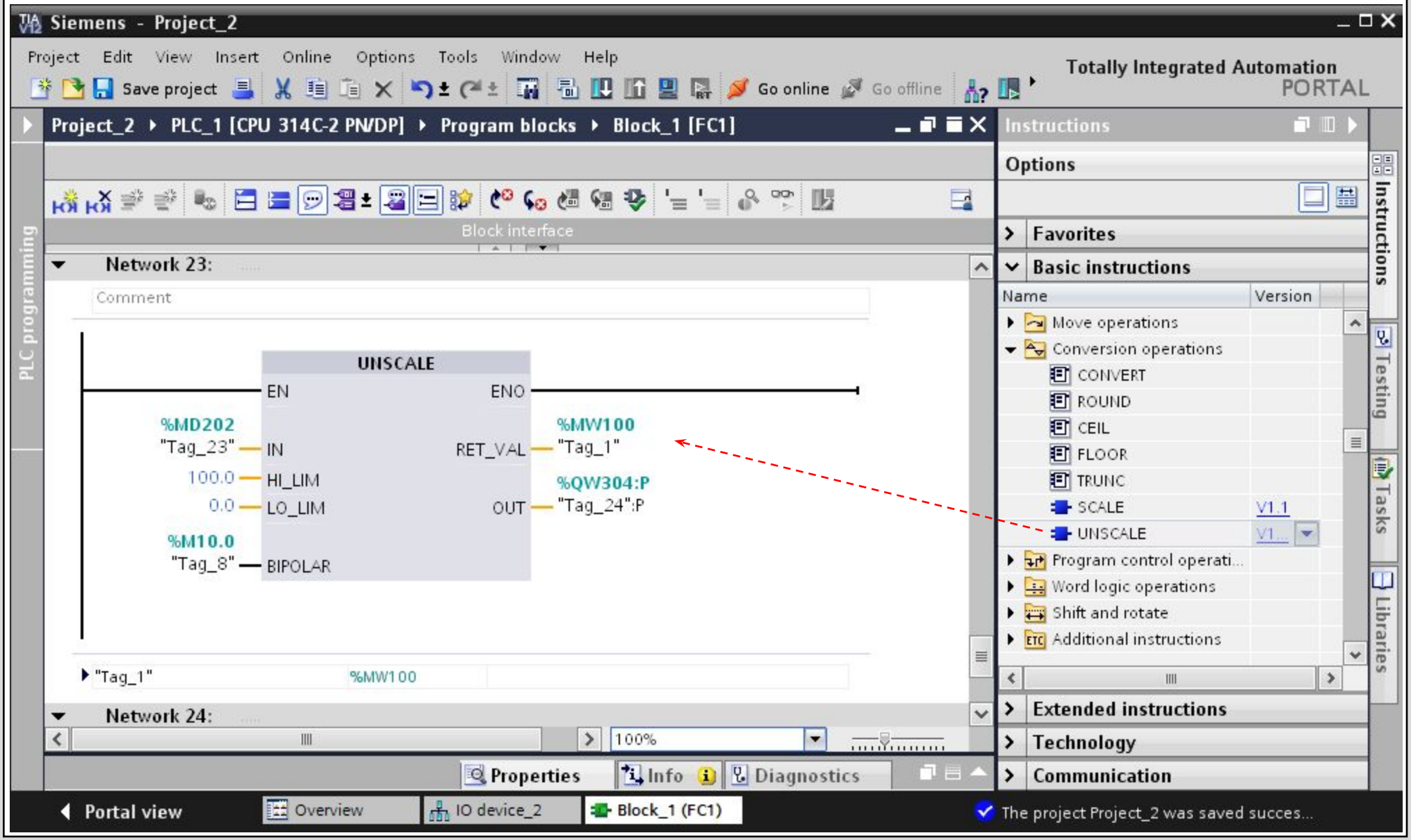


С датчика приходят сигналы обеих полярностей



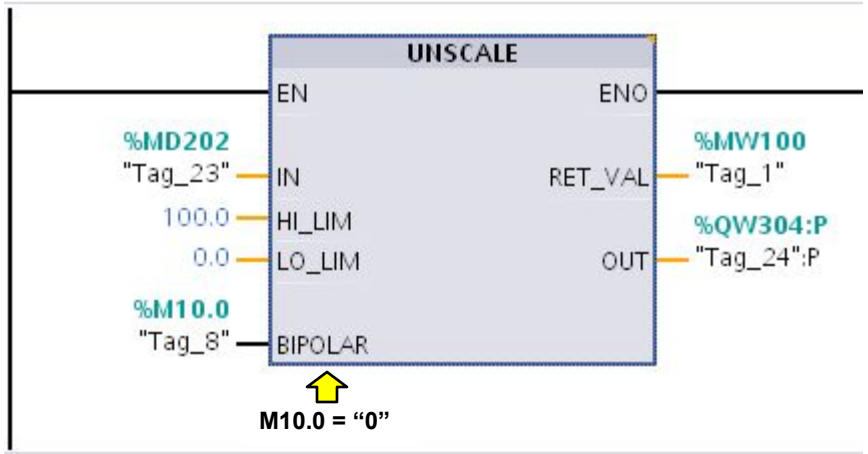


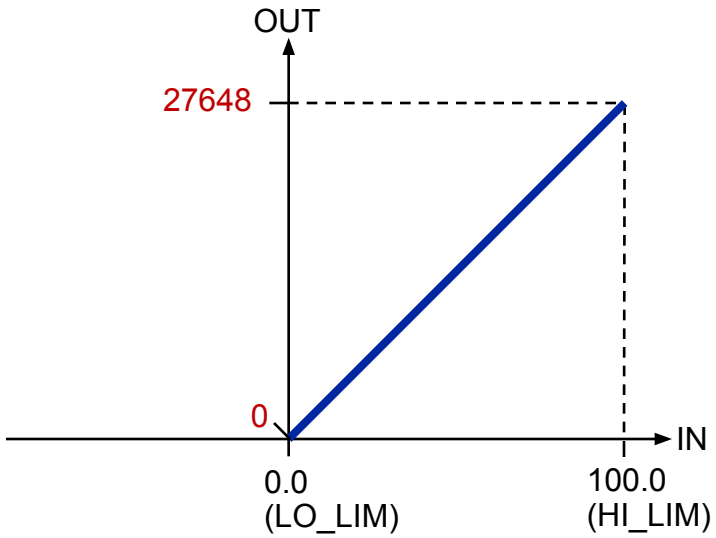
Формирование аналогового выхода функцией UNSCALE



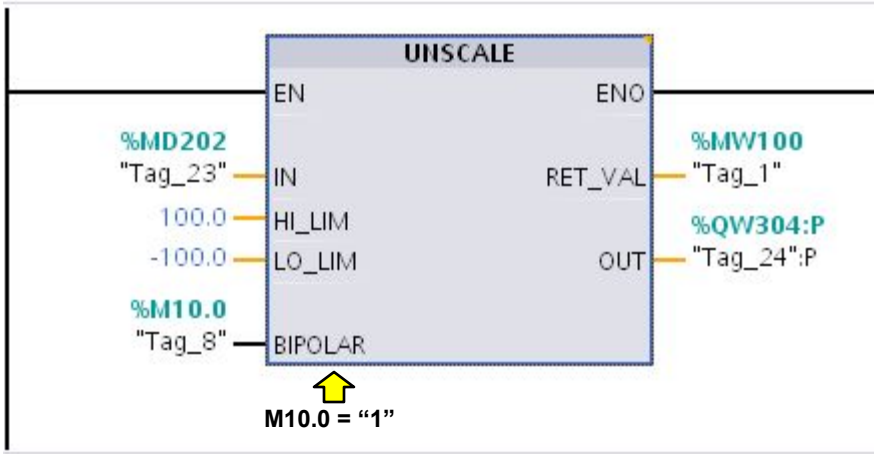
Формирование аналогового выхода функцией UNSCALE

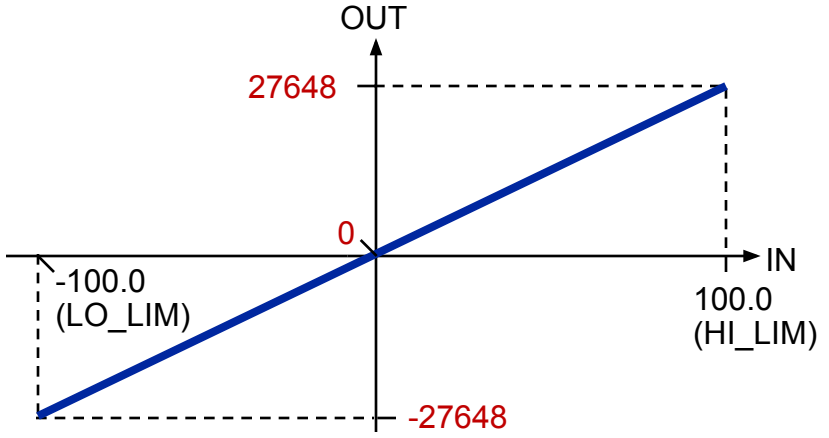
На выходе только положительные значения





На выходе как положительные, так и отрицательные значения



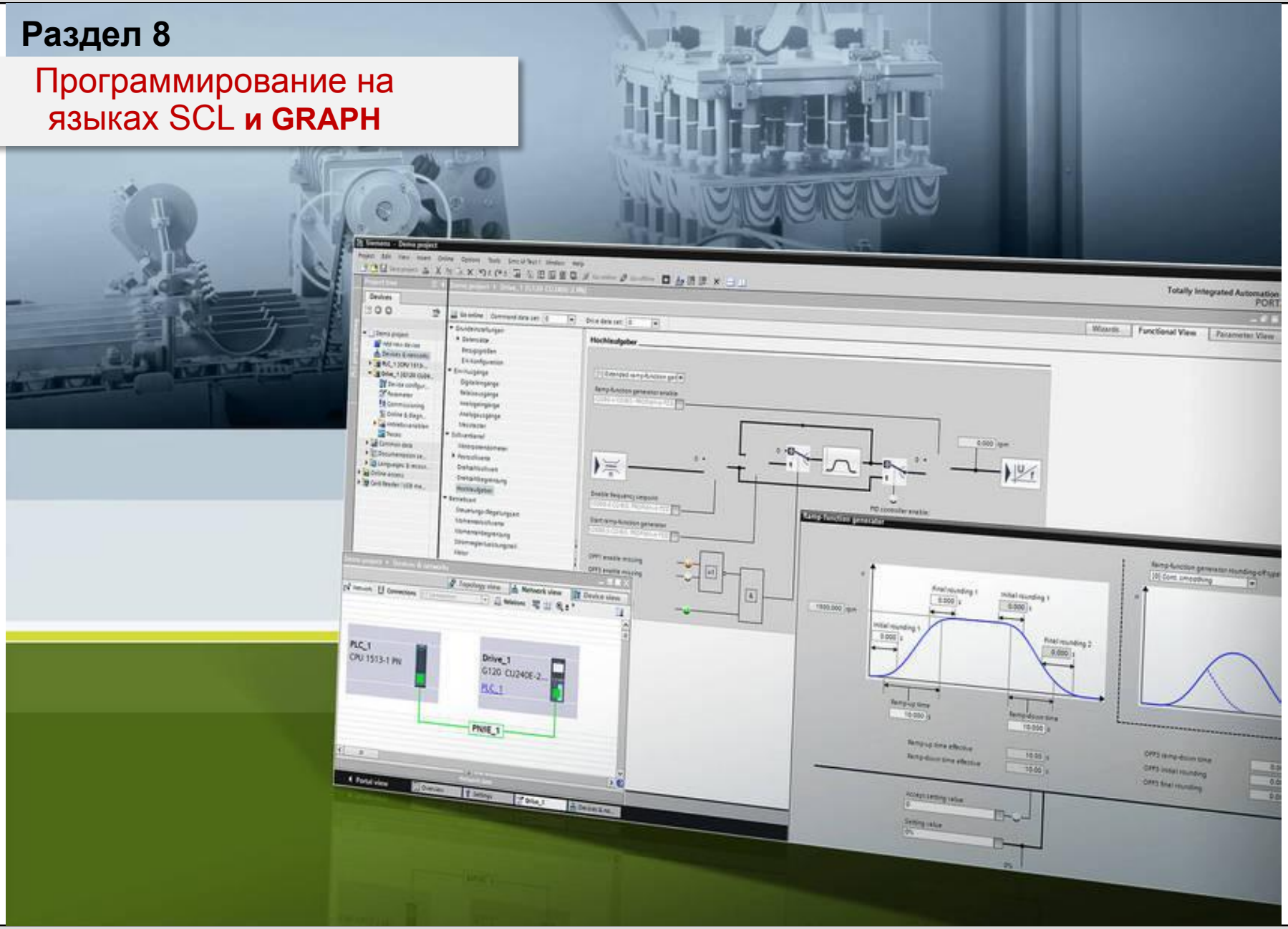


Конец раздела 7. Окно навигации

- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 8

Программирование на языках SCL **и GRAPH**



Операторы языка SCL

	Описание	Оператор
Присваивание	Присваивание	:=
Круглые скобки	(выражение)	(,)
Двоичная логика	Отрицание И ИЛИ Исключающее ИЛИ	NOT AND, & OR XOR
Сравнение	меньше, меньше или равно, больше, больше или равно, равно, не равно	<, <=, >, >= =, <>
Арифметика	плюс, минус (знак) сложение, вычитание умножение, деление возведение в степень	+,- +,- *, /, DIV, MOD **

Управляющие инструкции языка SCL

	Ключевое слово	Функция
Ветвление	**IF**	Ветвление программы со значением типа BOOL
	CASE	Ветвление программы со значением типа INT
Цикл (остановка возможна)	**FOR**	Цикл со счетчиком
	WHILE	Цикл с предварительной оценкой условия
	REPEAT	Цикл с оценкой условия в конце цикла
Остановка цикла	**CONTINUE**	Переход к следующему шагу цикла
	EXIT	Выход из цикла
Выход из блока	**RETURN**	Выход из блока

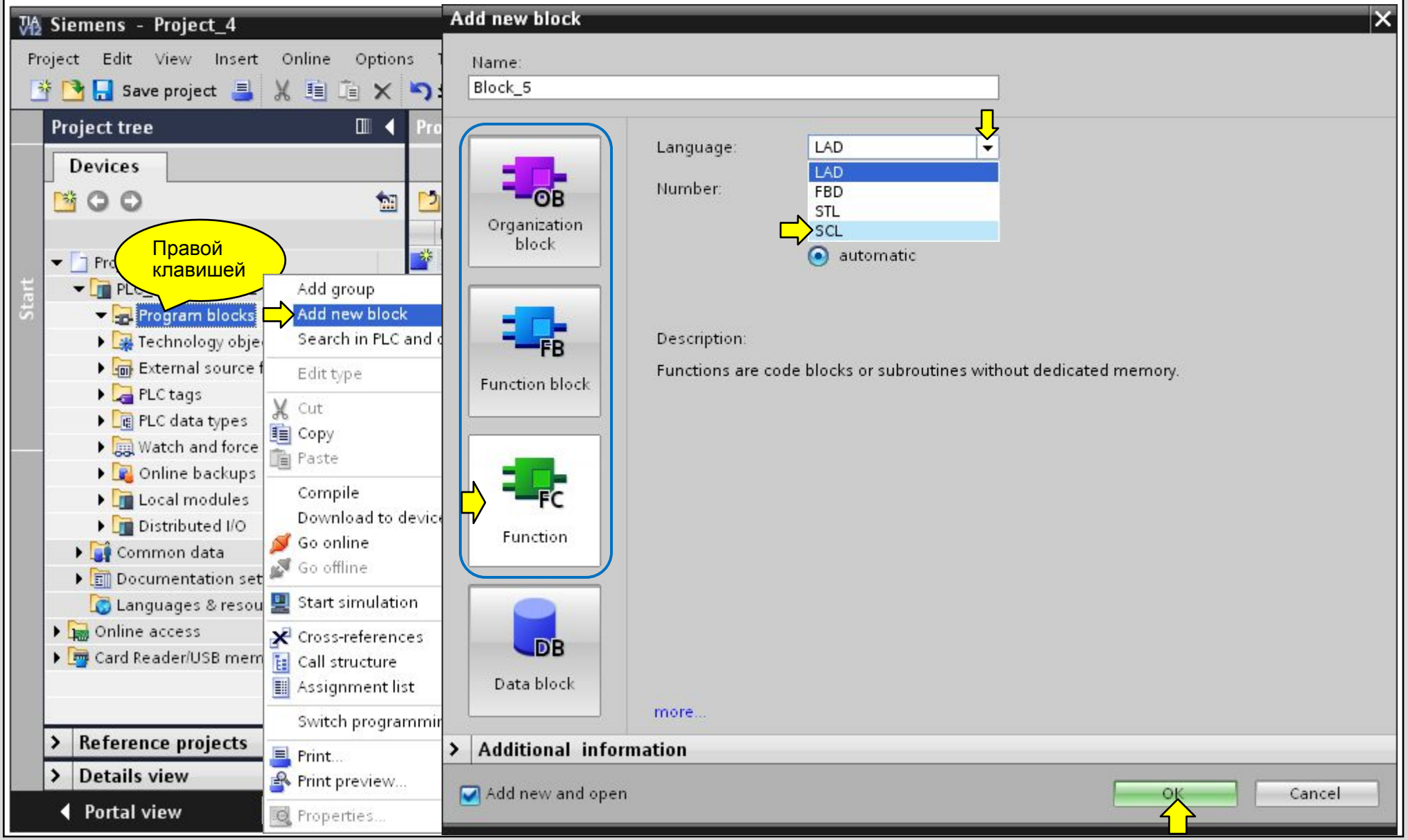
Прямая адресация переменных

	Область	Пример обозначения	Примеры
Абсолютная адресация	Бит	DBz.DBXy.x , Iy.x	DB5.DBX0.7 , I 0.0
	Байт	DBz.DBBy , MWy	DB5.DBB2 , QB2
	Слово	DBz.DBWy , QWy	DB5.DBW4 , MW20
	Двойное слово	DBz.DBDy, QDy	DB5.DBD8 , ID40
Символьная адресация		"<Имя блока DB>".<Имя переменной>	"Motor".Setpoint
	Только для S7-1200	<Имя переменной>.X<номер бита> < Имя переменной >.B<номер байта> < Имя переменной >.W<номер слова>	“Motor“.Alarms.X1 (бит 1 переменной “Alarms“ блока DB “Motor“)
Прямая адресация для входов, выходов, меркеров и переменных DB схожа с LAD/FBD/STL			

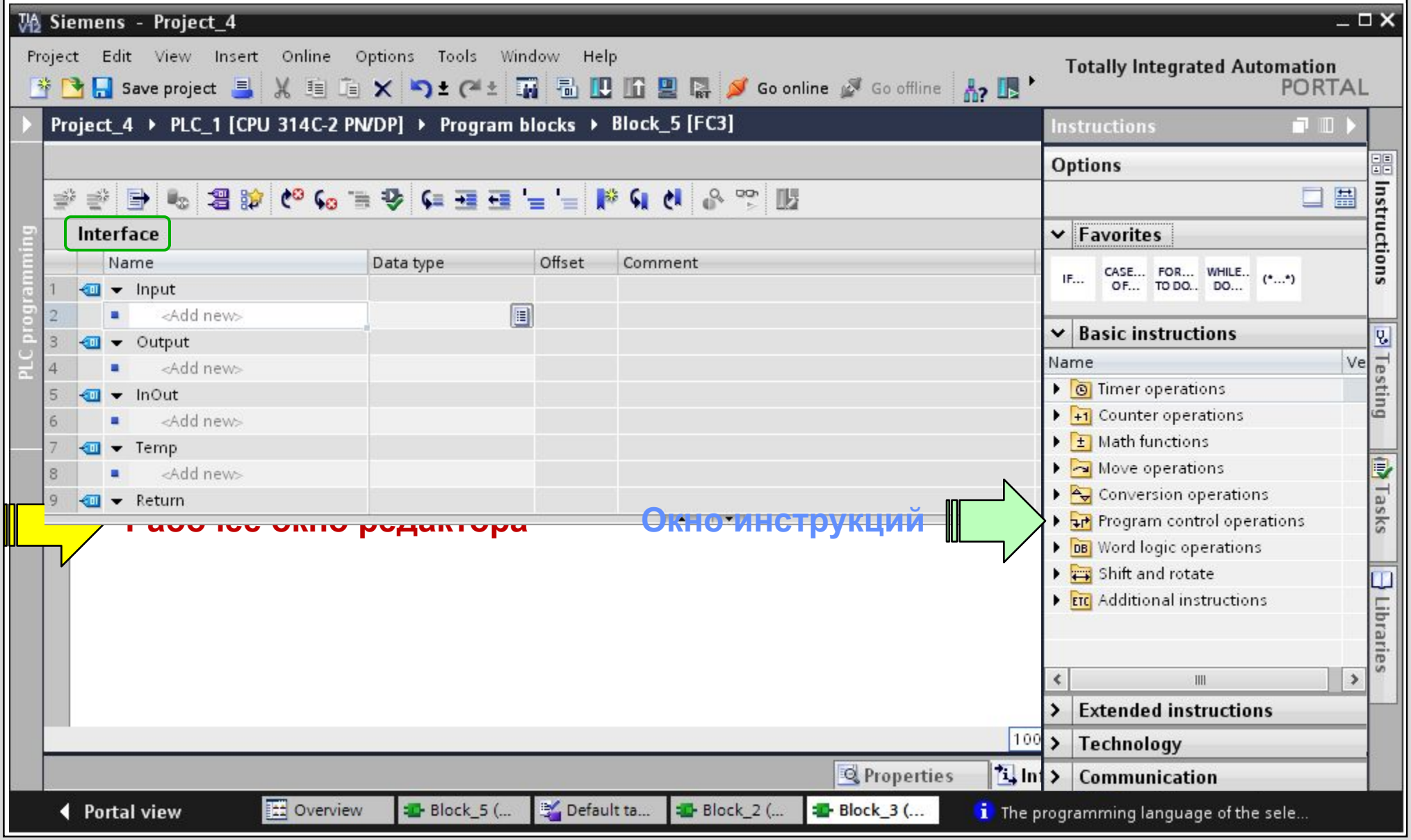
Примеры косвенной адресации

	Область	Пример обозначения	Примеры
Косвенная адресация	Бит	DBz.DBX[номер байта.бит] I[номер байта.бит]	DB10.DBX[a.b] , “Motor“.DBX[a.b] , I[a.b]
	Слово	DBz.DBW[номер слова] MW[номер слова]	“Motor“.DBW[a] , MW[a+b]
	Массив элементов	„<Символьное имя блока DB>“. <Символьное имя массива>[Индекс]	“Motor“.Value[i]
	DB	Word_to_Block_DB[index].DW0	Word_to_Block_DB[i].DW0
Косвенная адресация для входов, выходов, меркеров и переменных DB схожа LAD/FBD/STL			

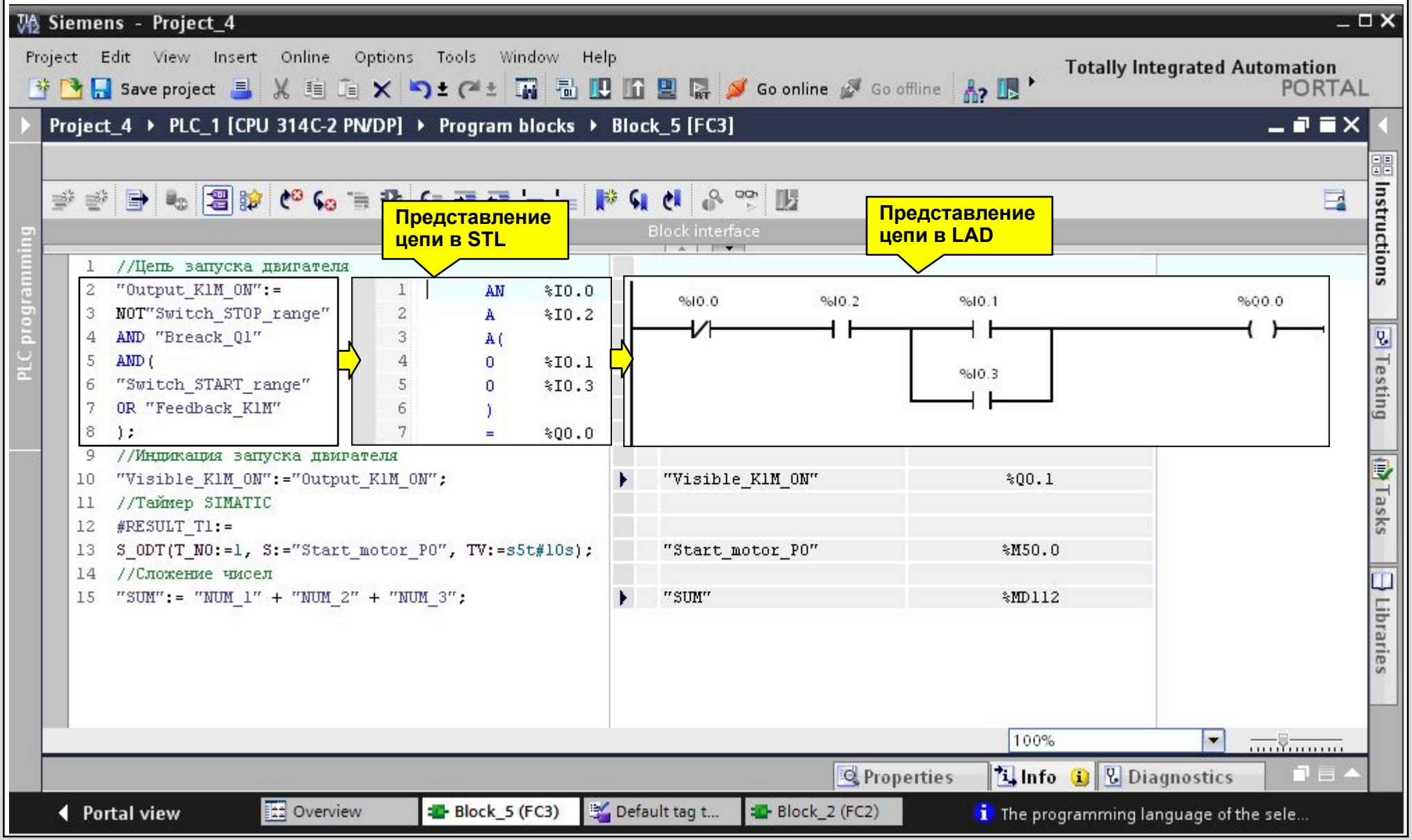
Создание программного блока с выбором языка SCL



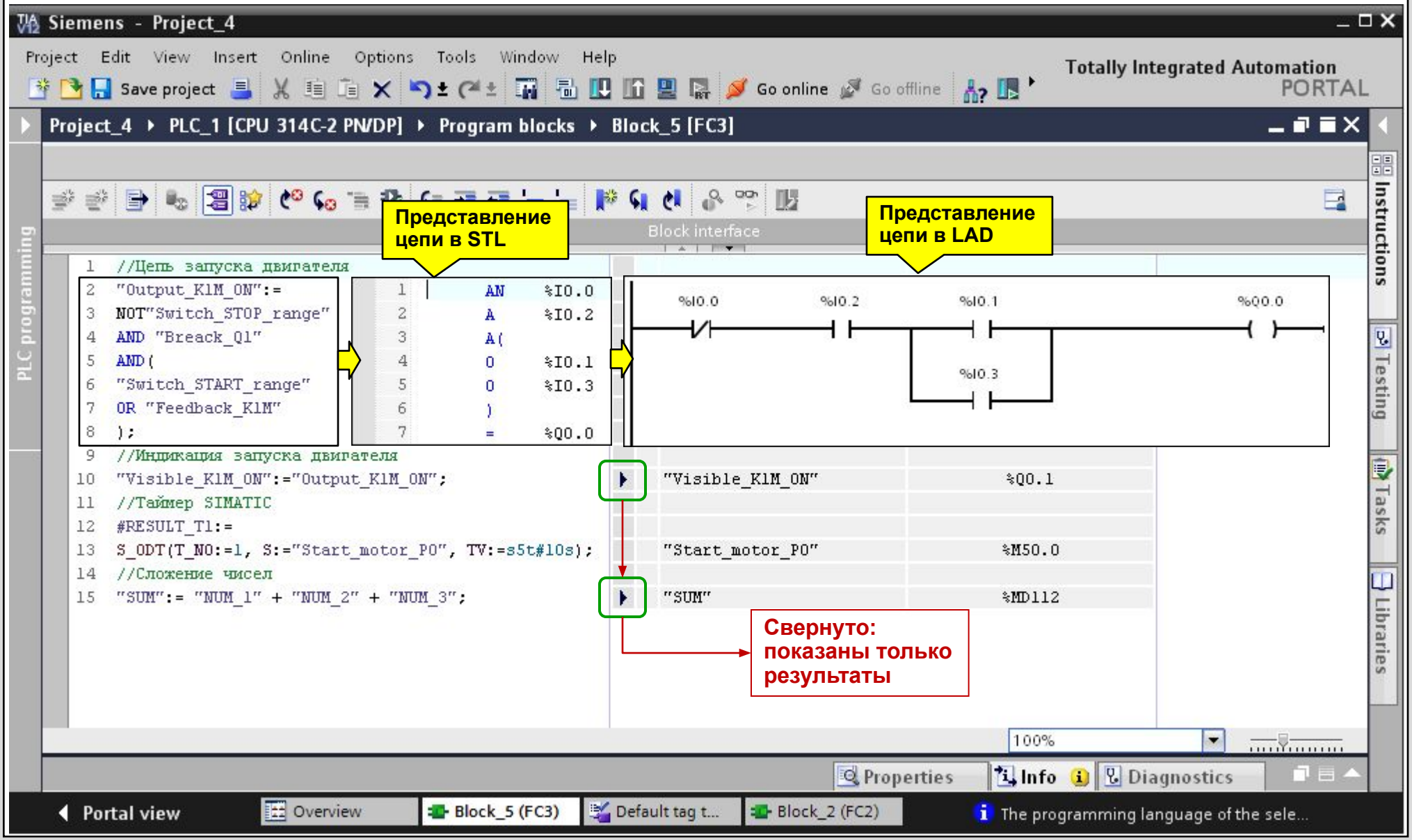
Окна инструментов программирования



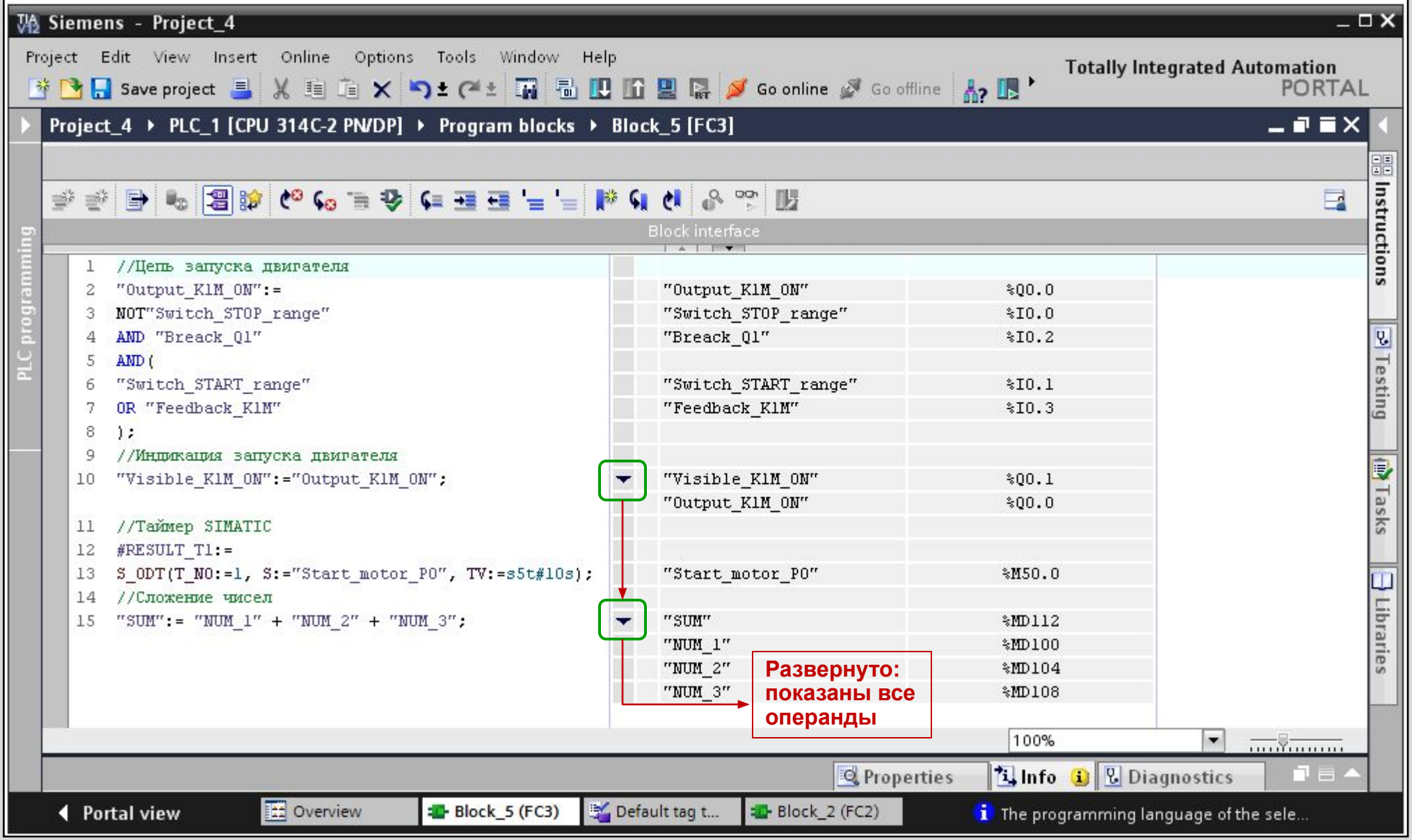
Примеры программирования в рабочем окне редактора



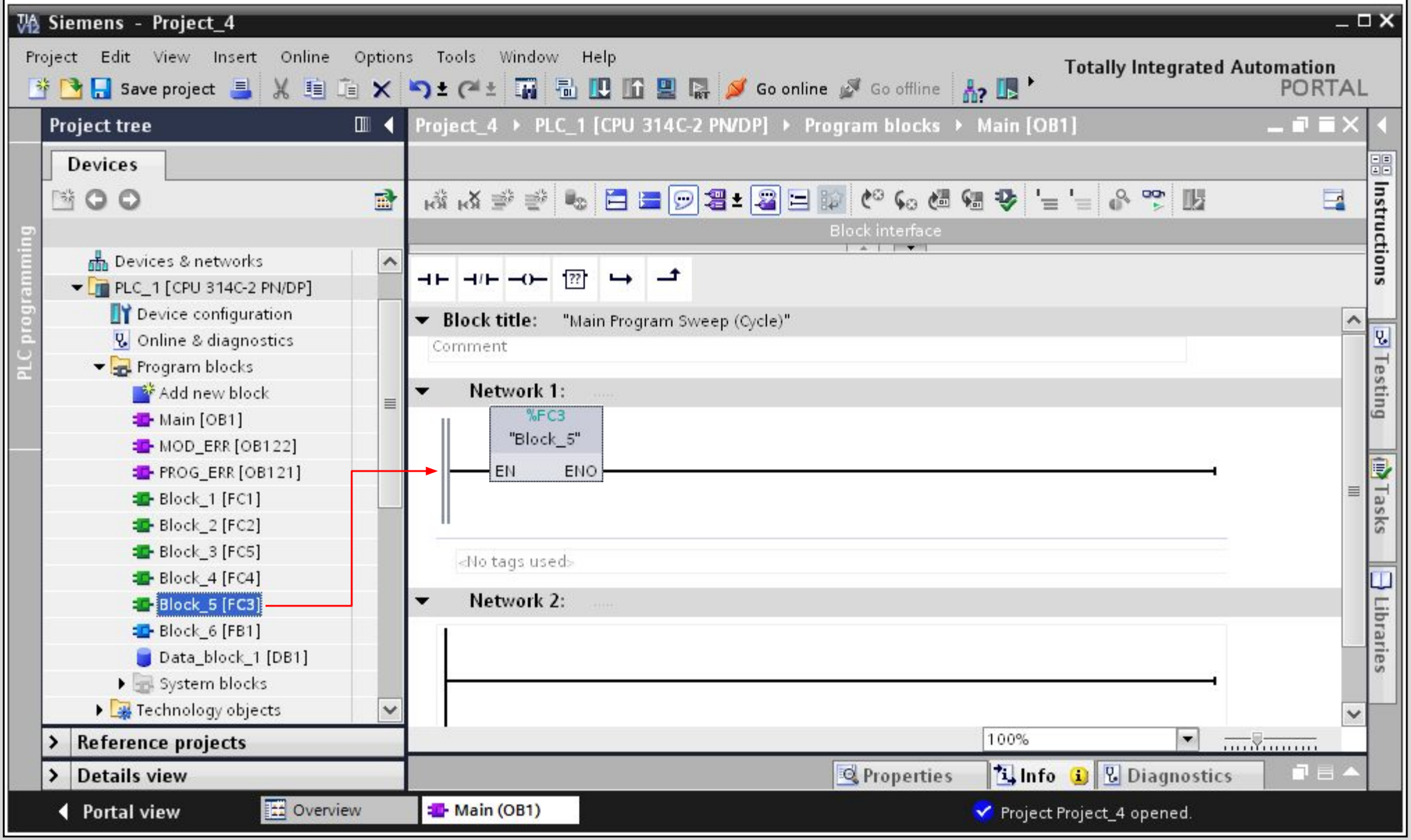
Примеры программирования в рабочем окне редактора



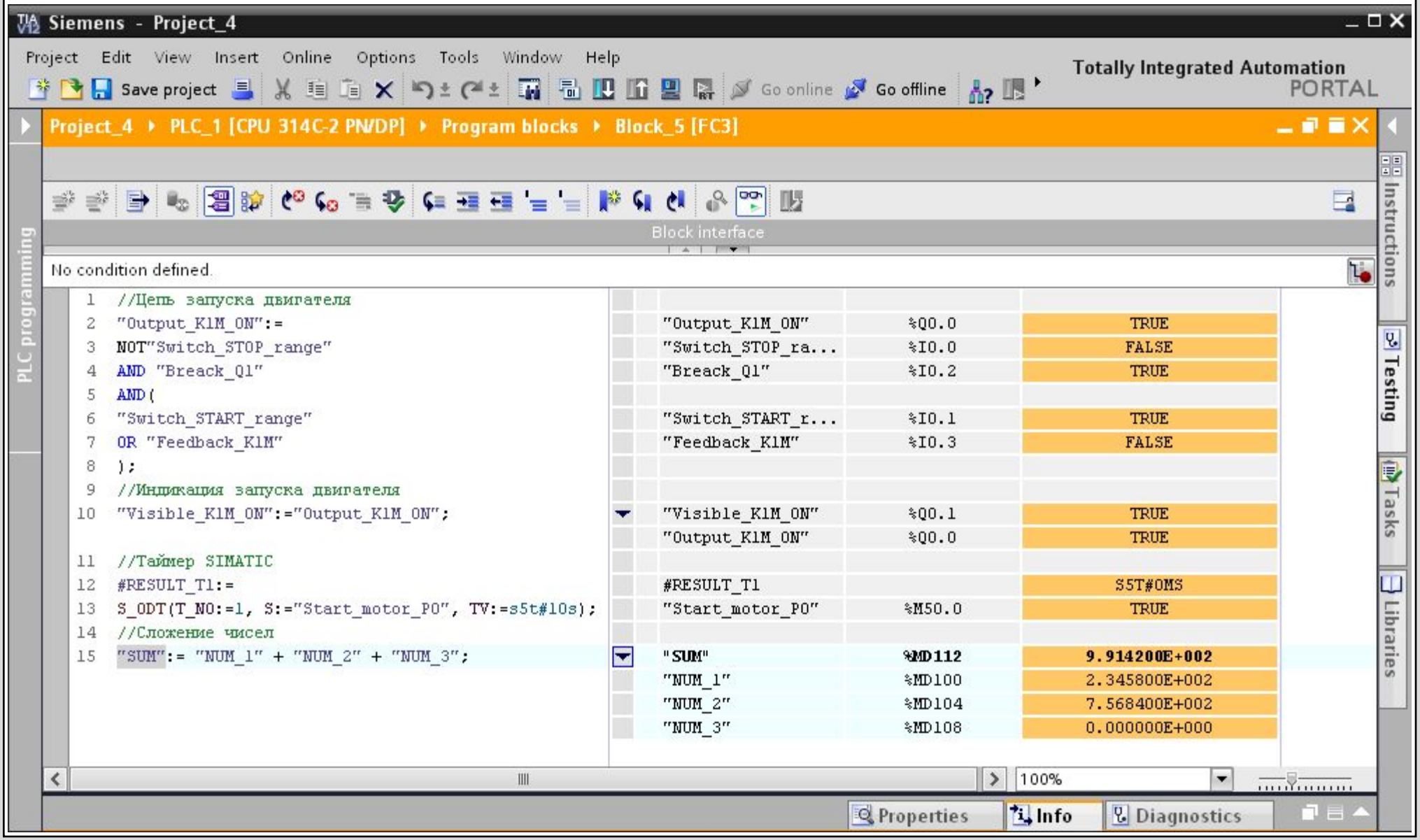
Вариант развернутого показа переменных



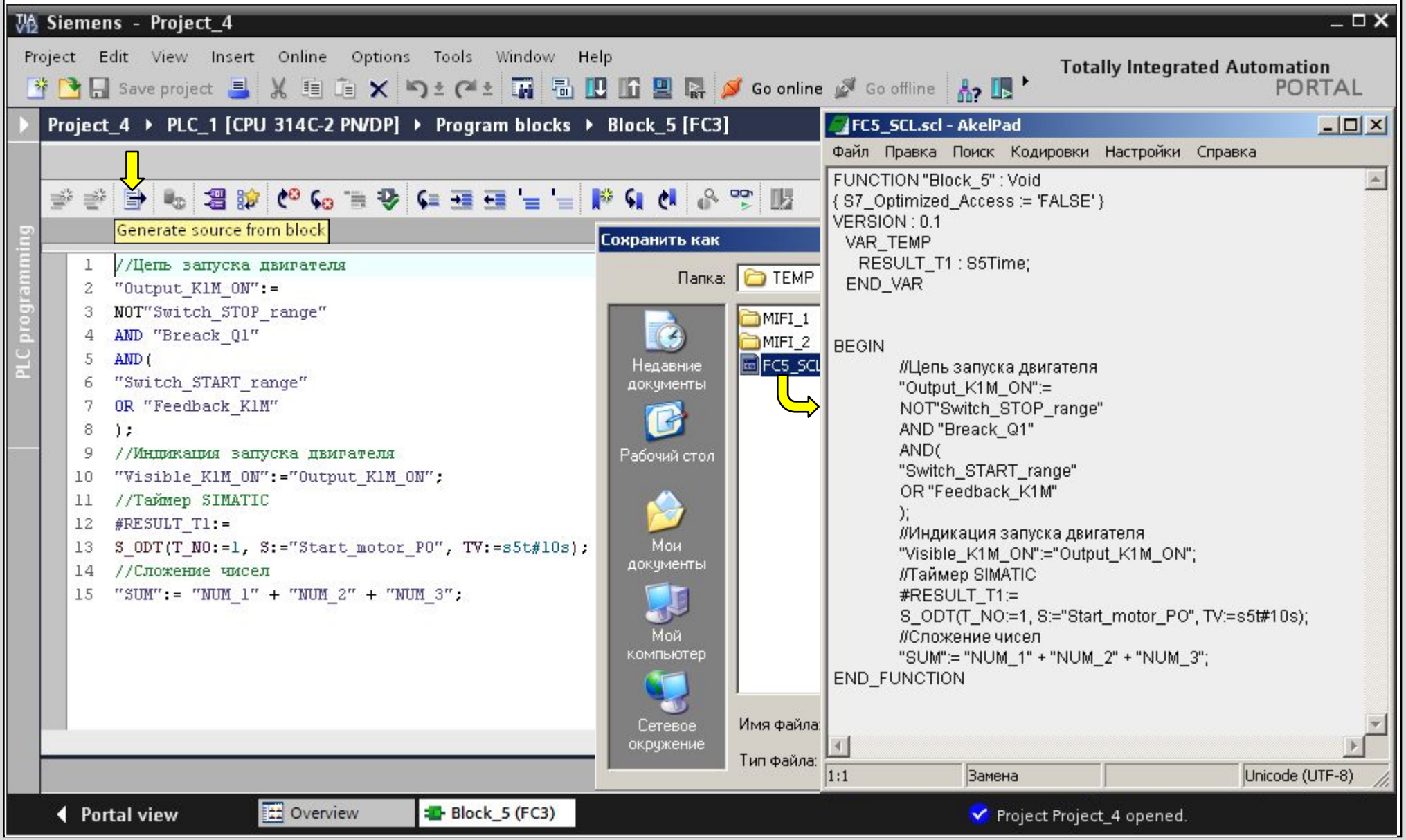
Вызов созданного в CSL блока



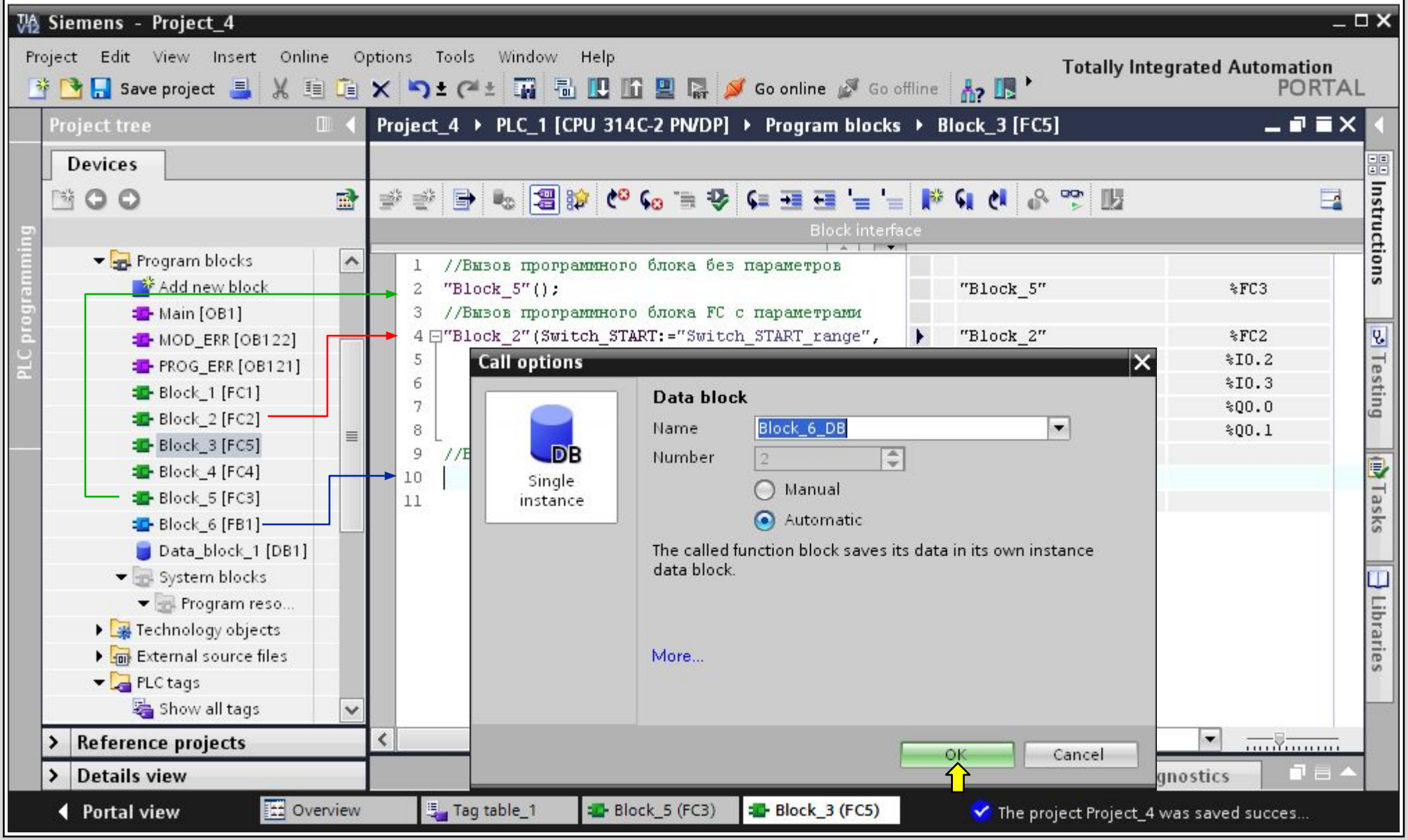
Просмотр состояния переменных в режиме On-Line



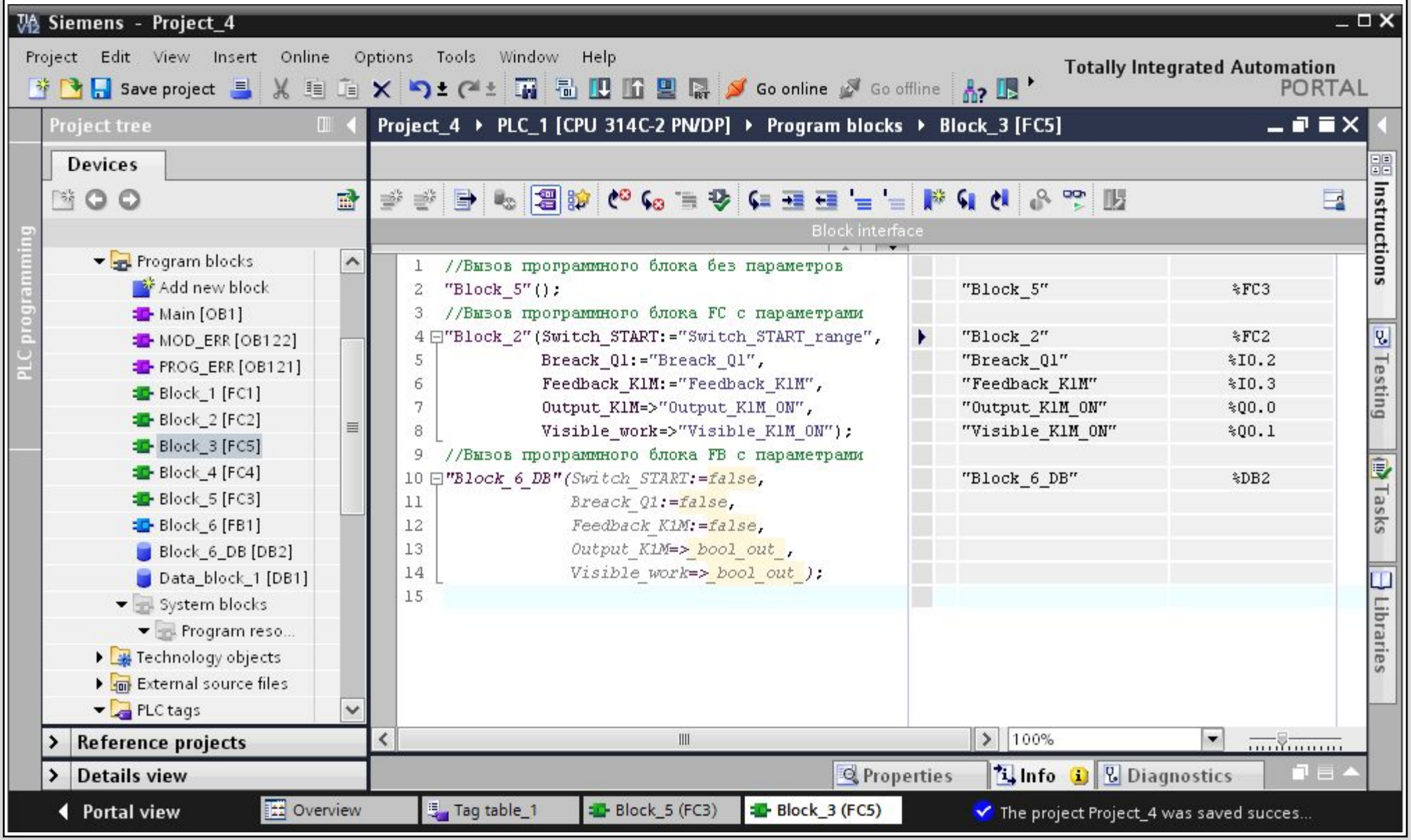
Генерация исходного файла



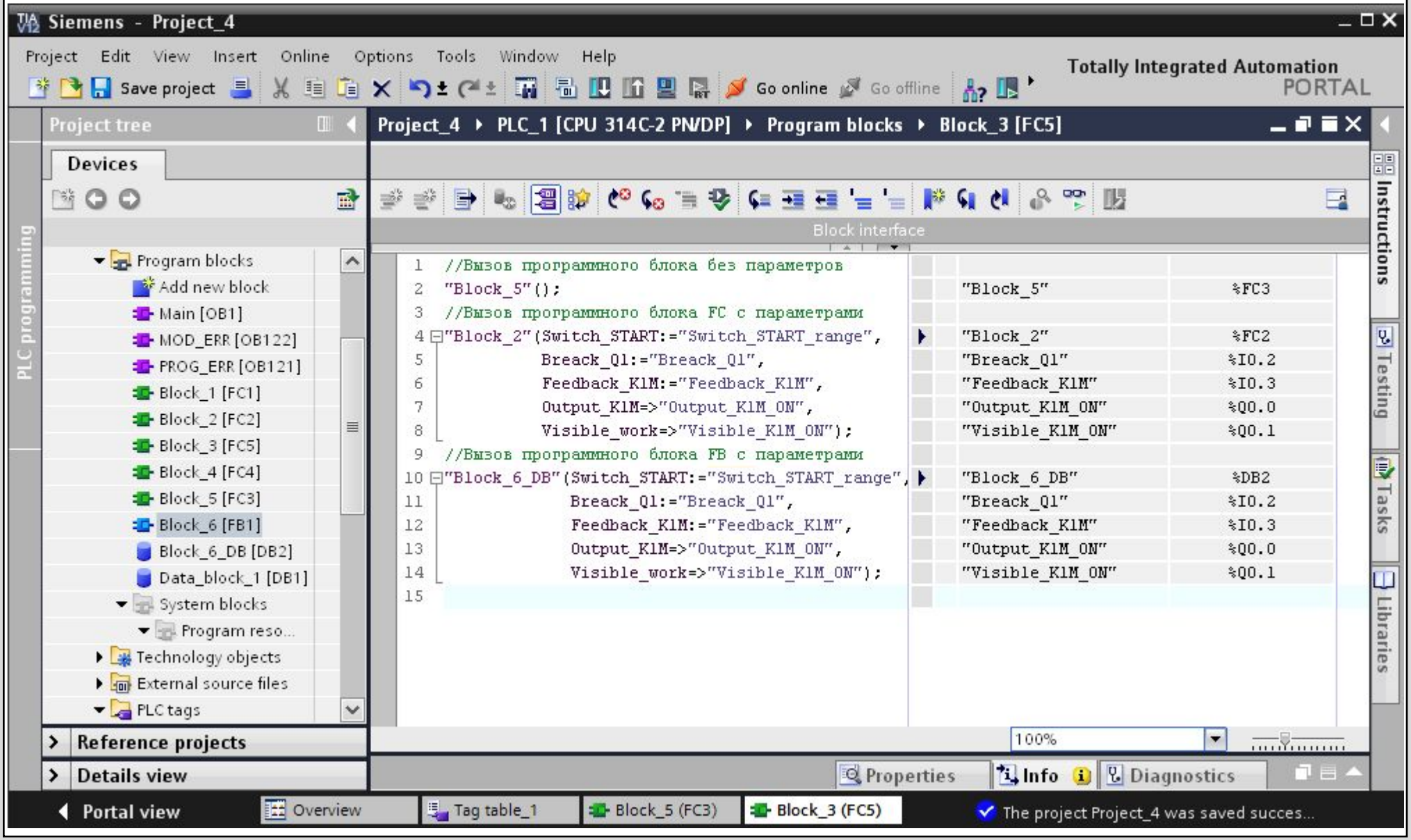
Примеры вызова программных блоков в редакторе SCL



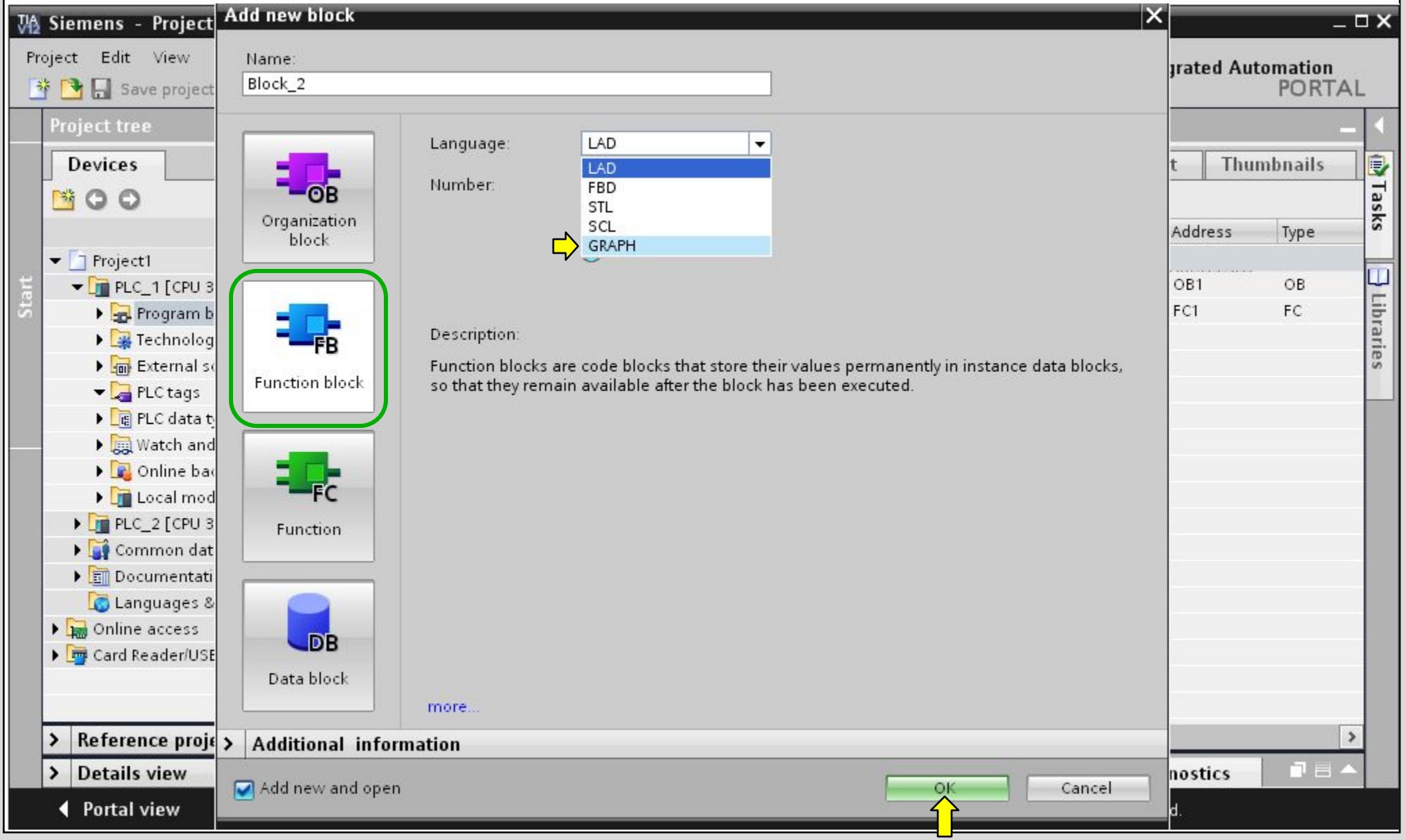
Примеры вызова программных блоков в редакторе SCL



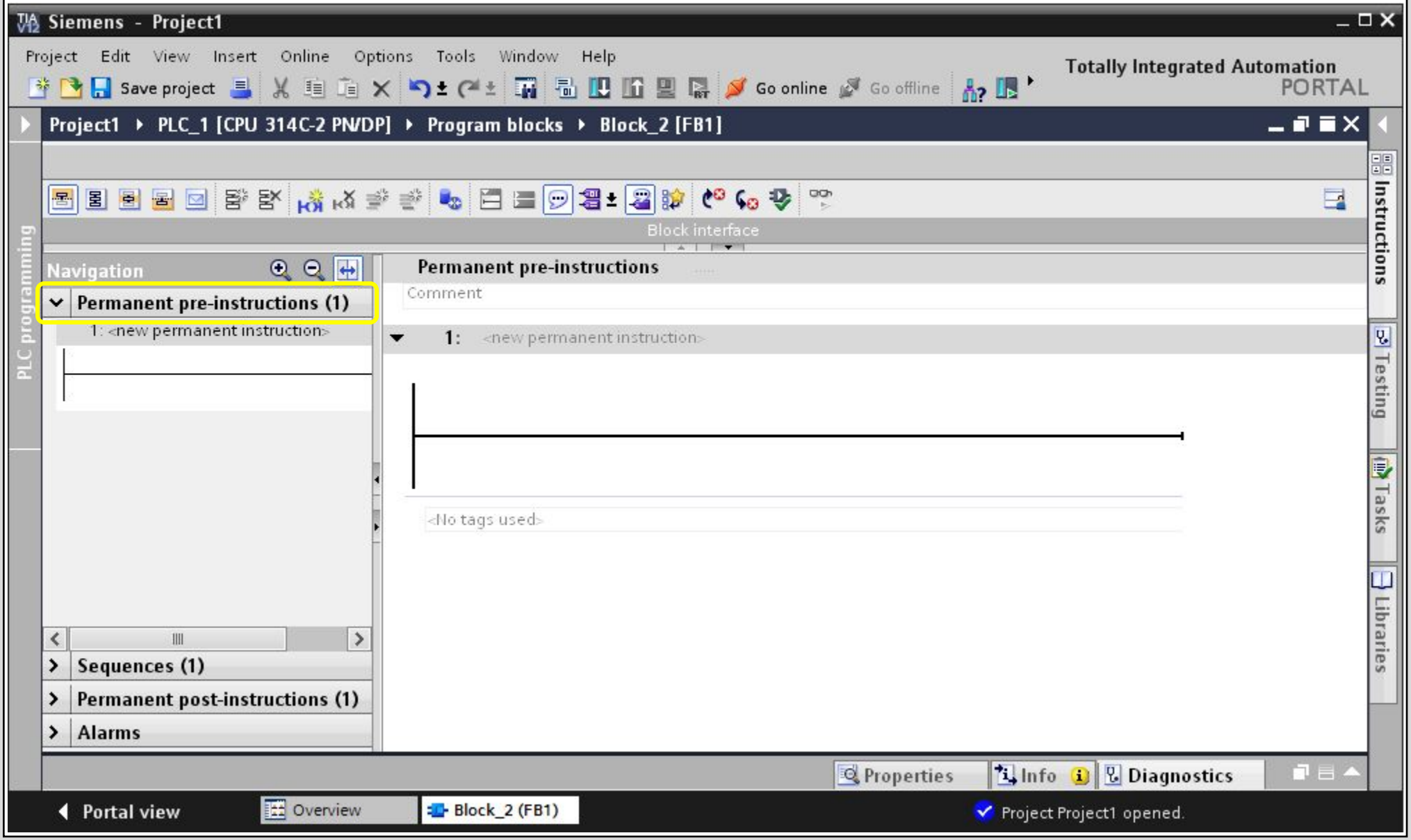
Примеры вызова программных блоков в редакторе SCL



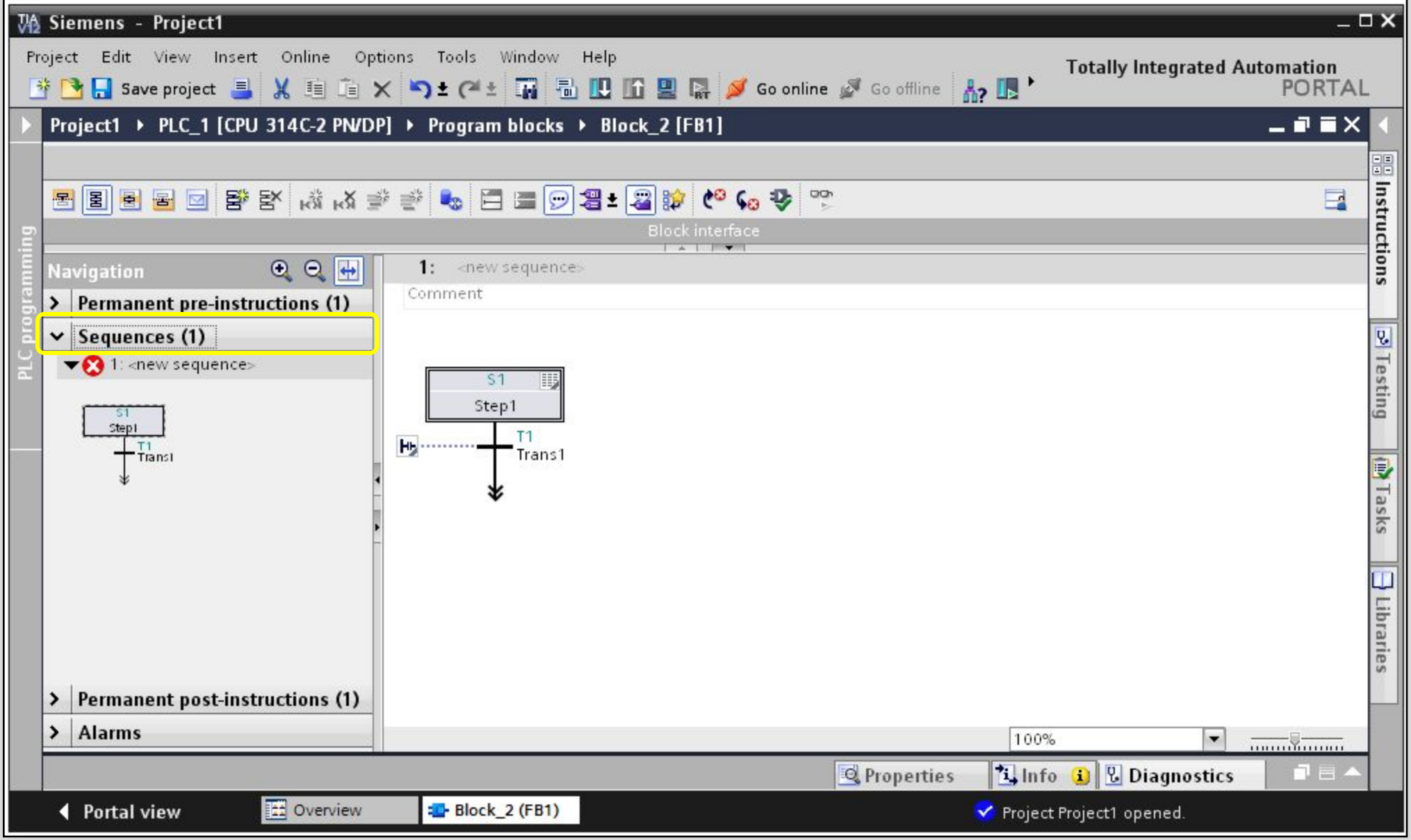
Построение программы в редакторе GRAPH. Создание FB



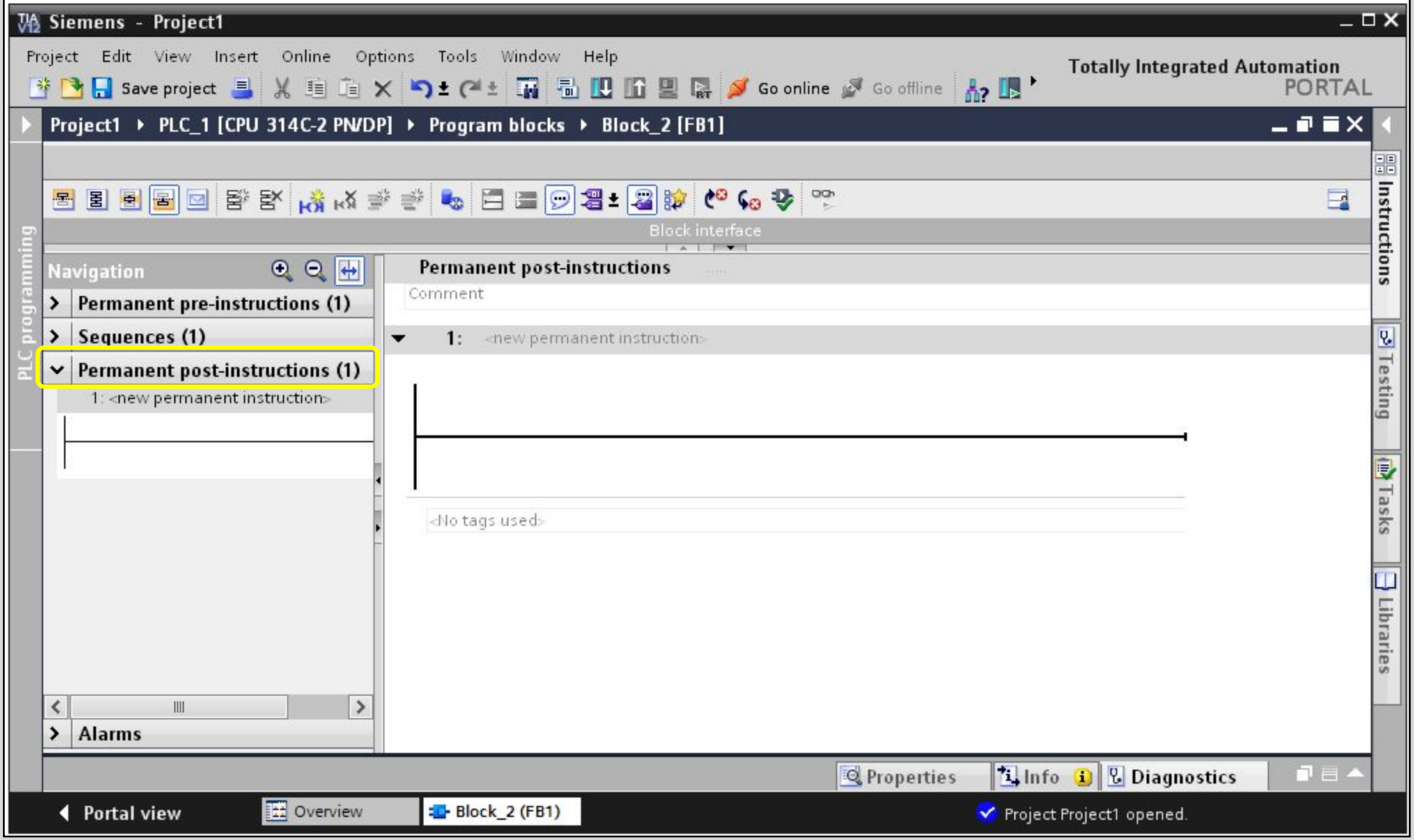
Окно написания программы до активизации пошагового режима



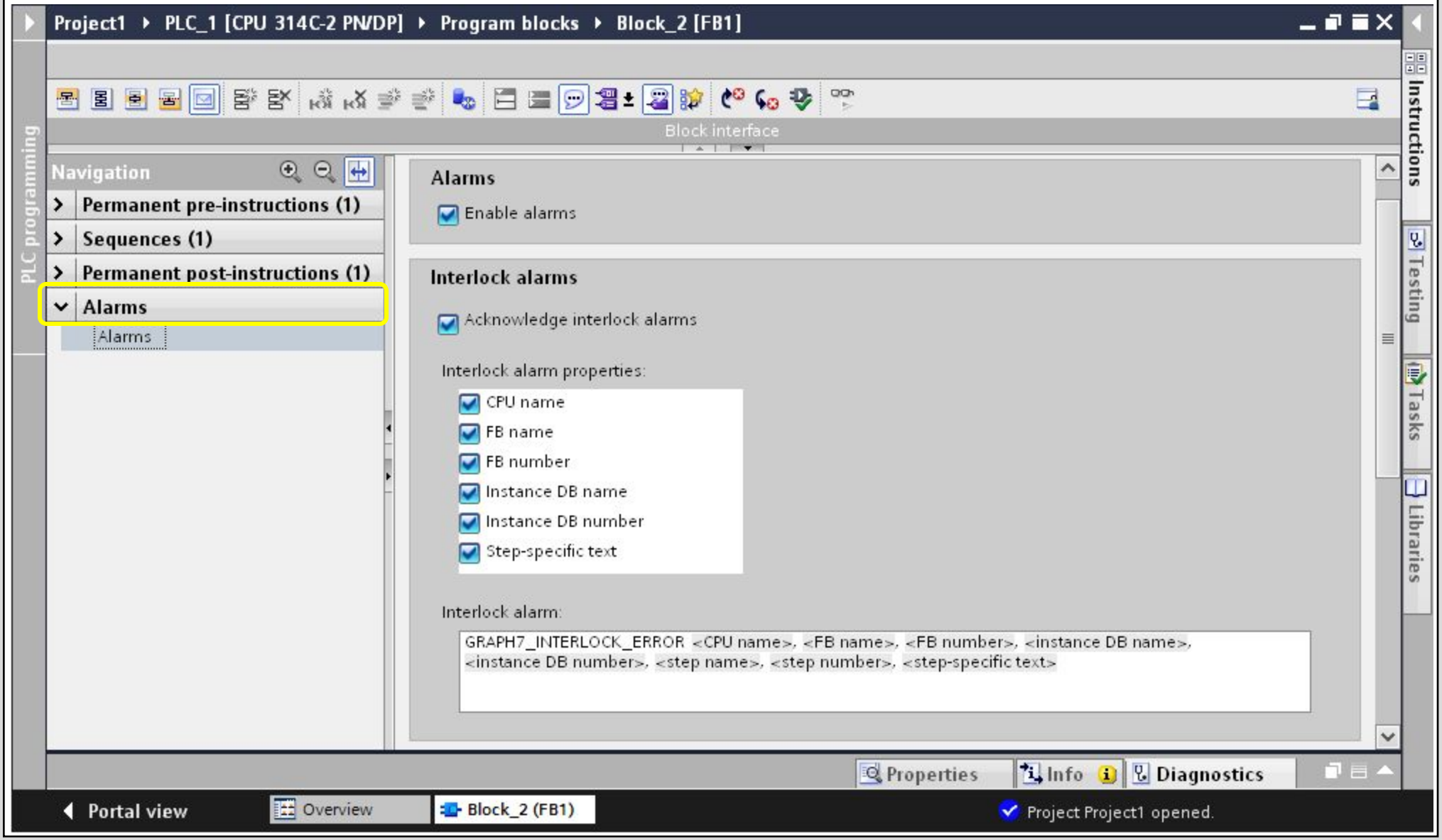
Окно написания программы в пошаговом режиме



Окно написания программы после окончания пошагового режима

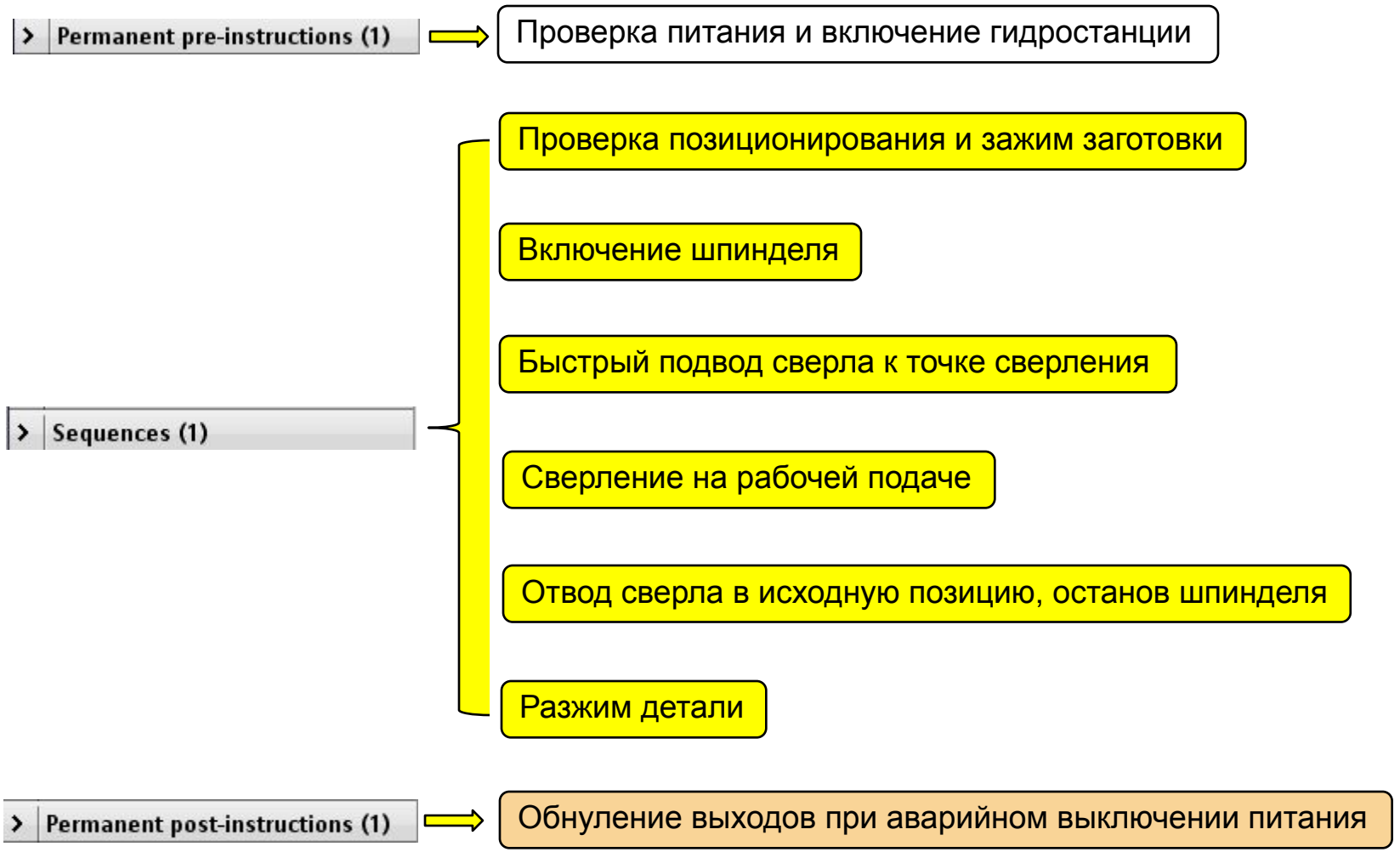


Окно предупреждений об ошибках

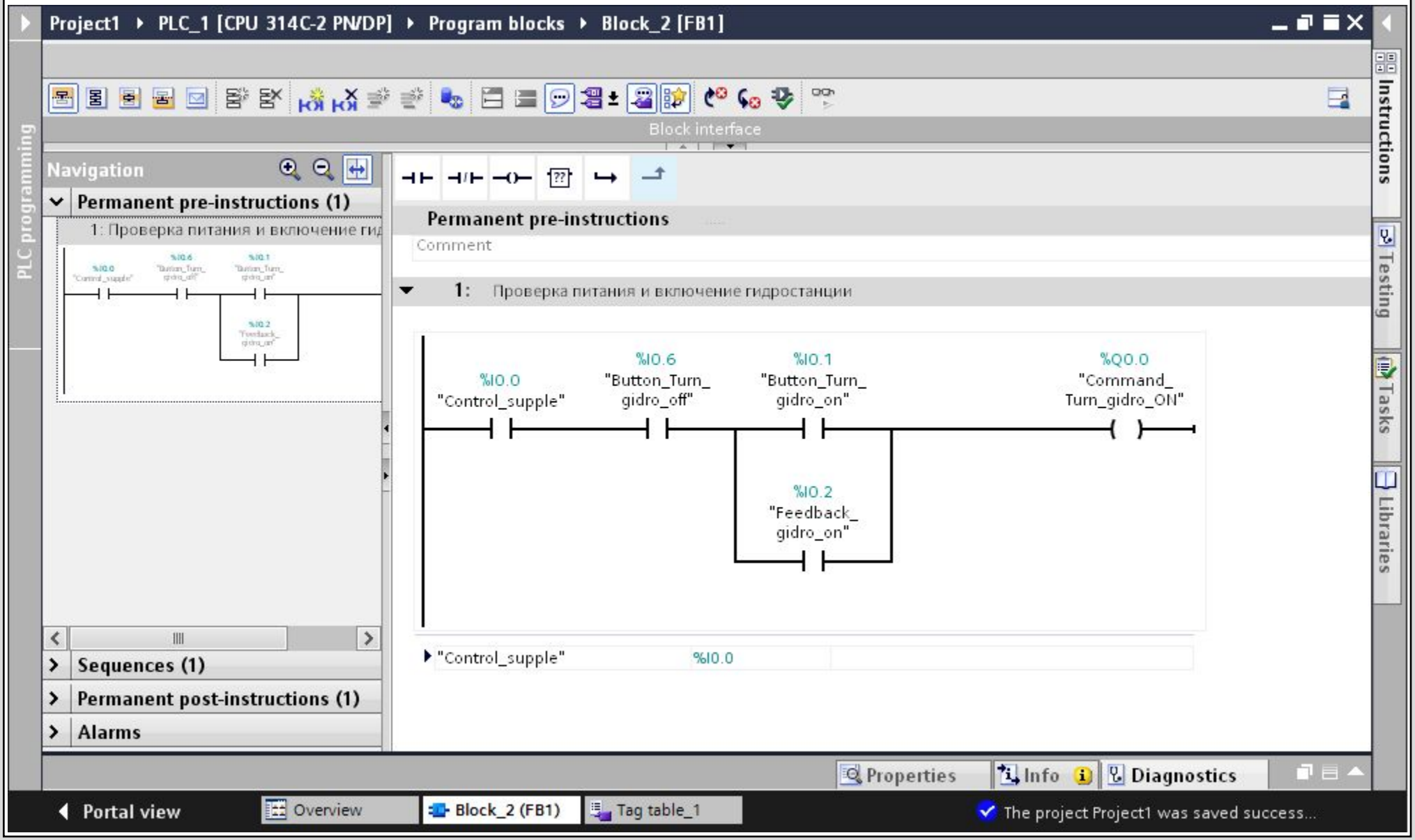


Пример построения программы в редакторе GRAPH

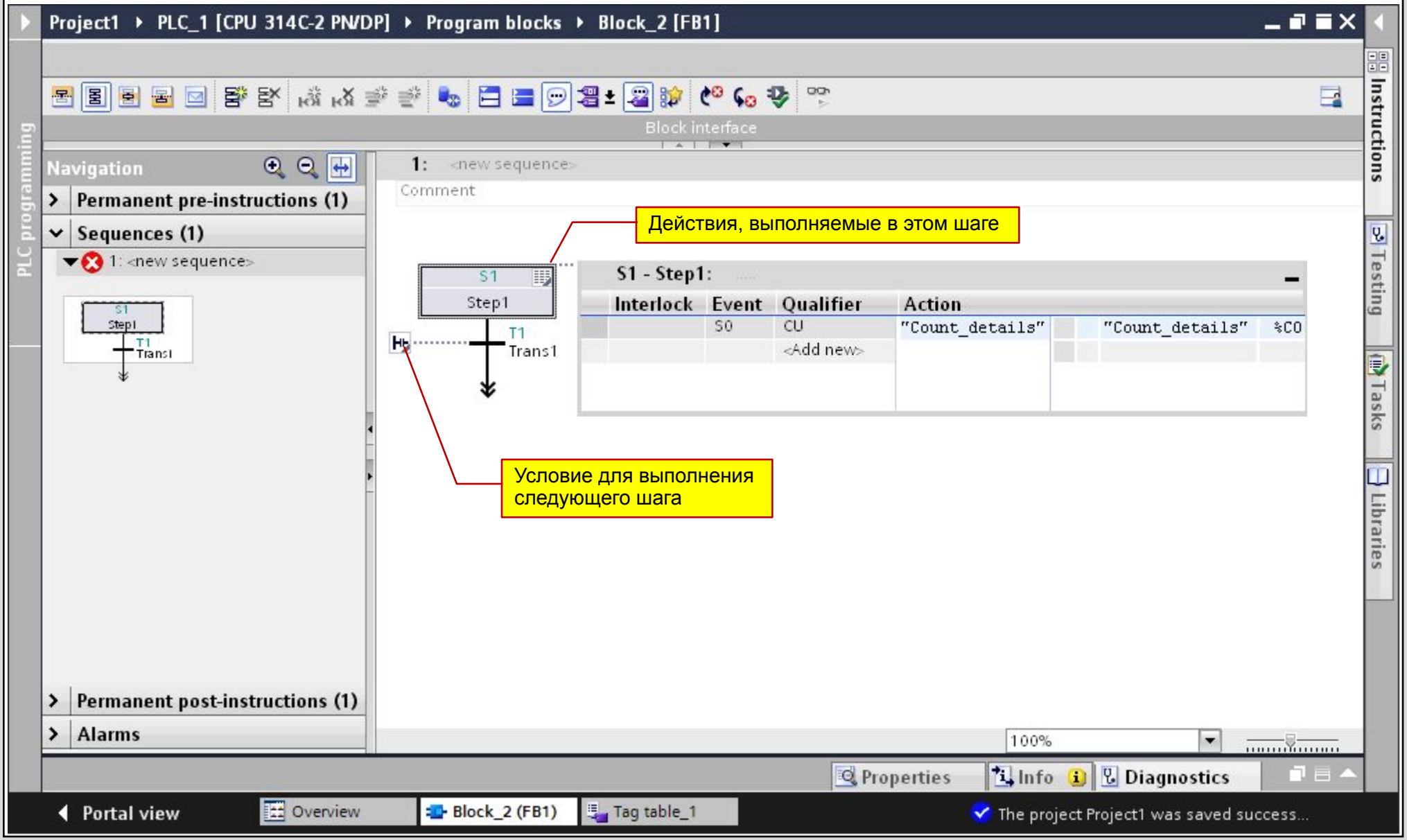
Построим программу управления сверлильным станком



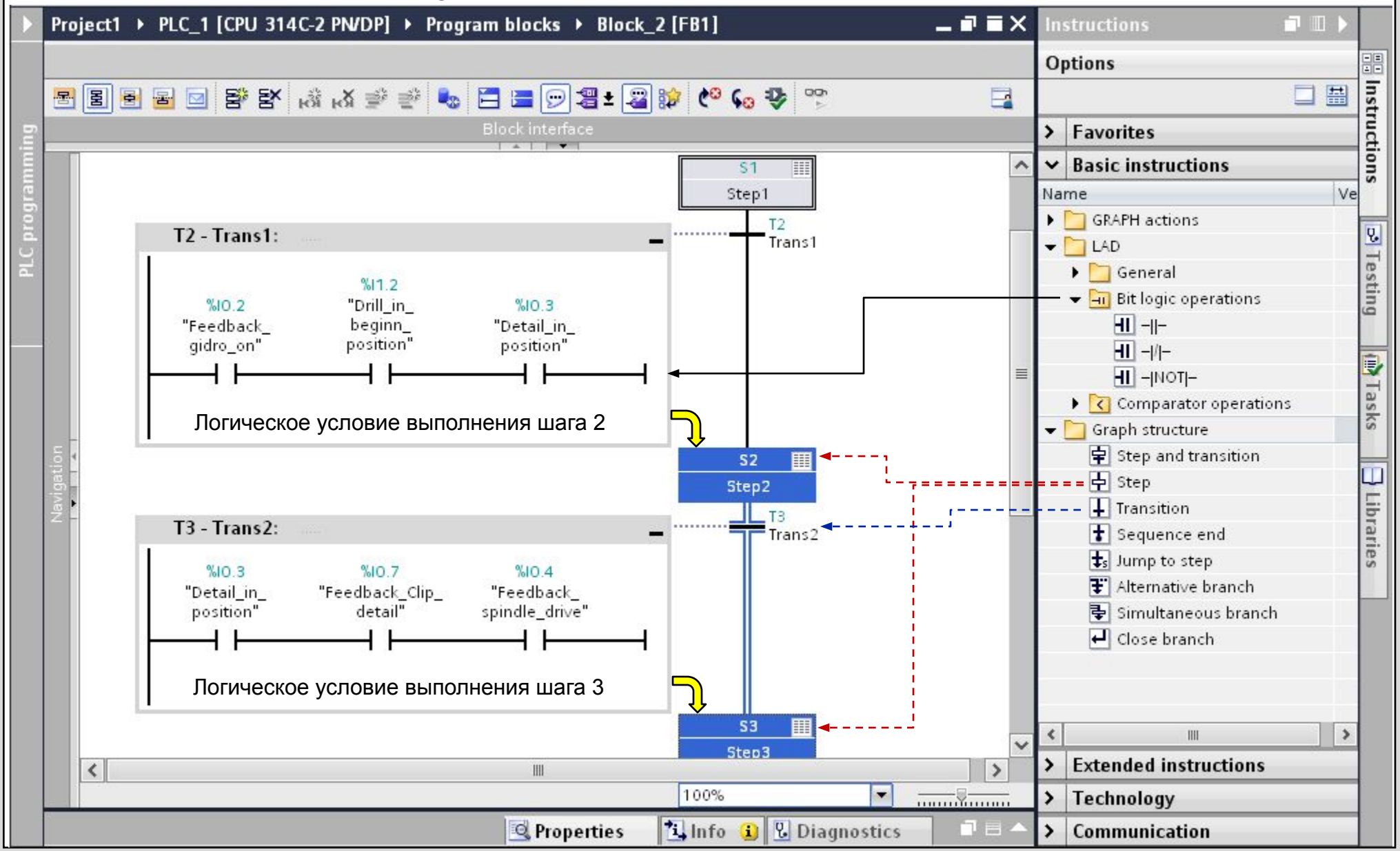
Написание инструкций в разделе Permanent pre-instructions



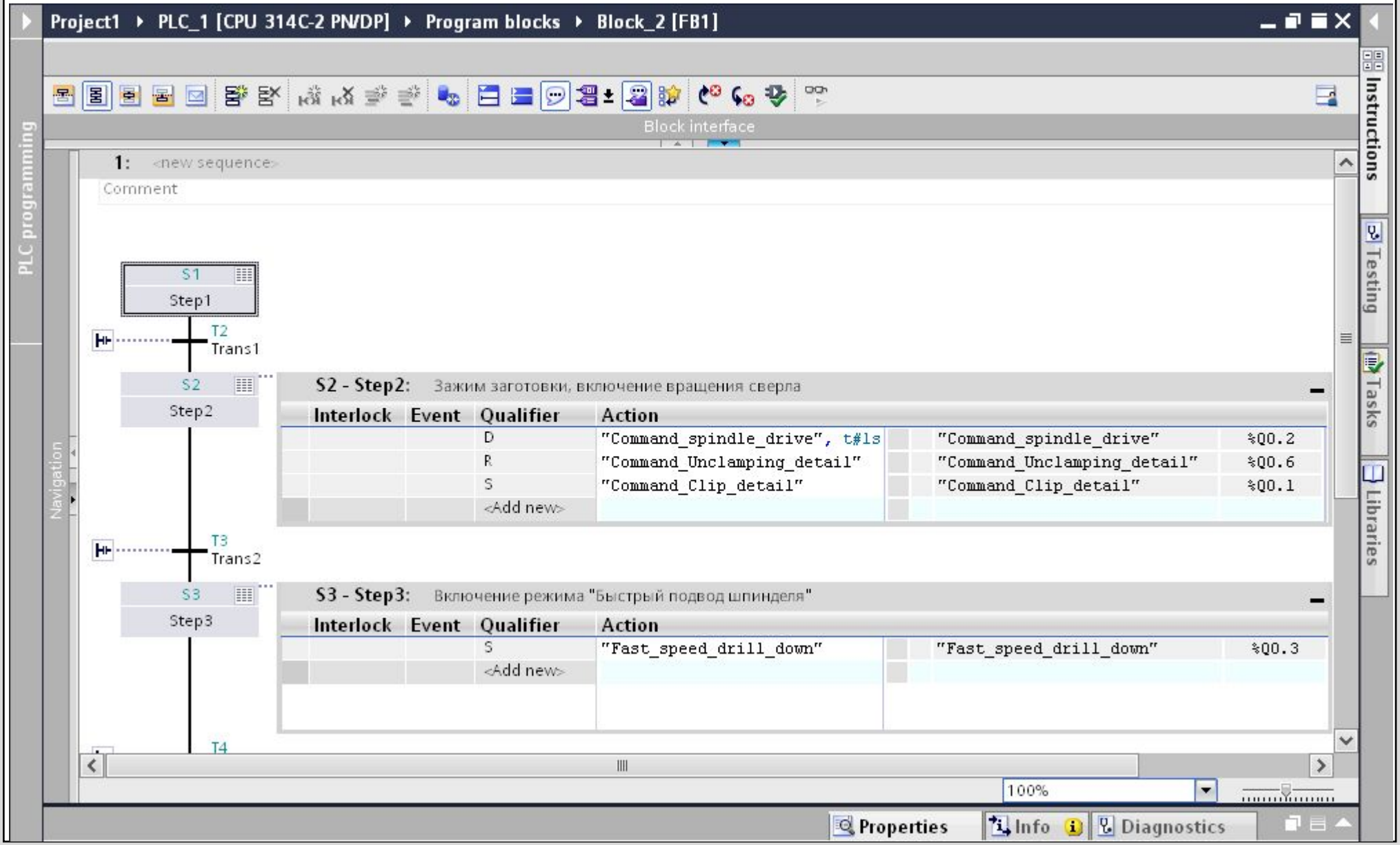
Структура реализации пошагового алгоритма Sequences



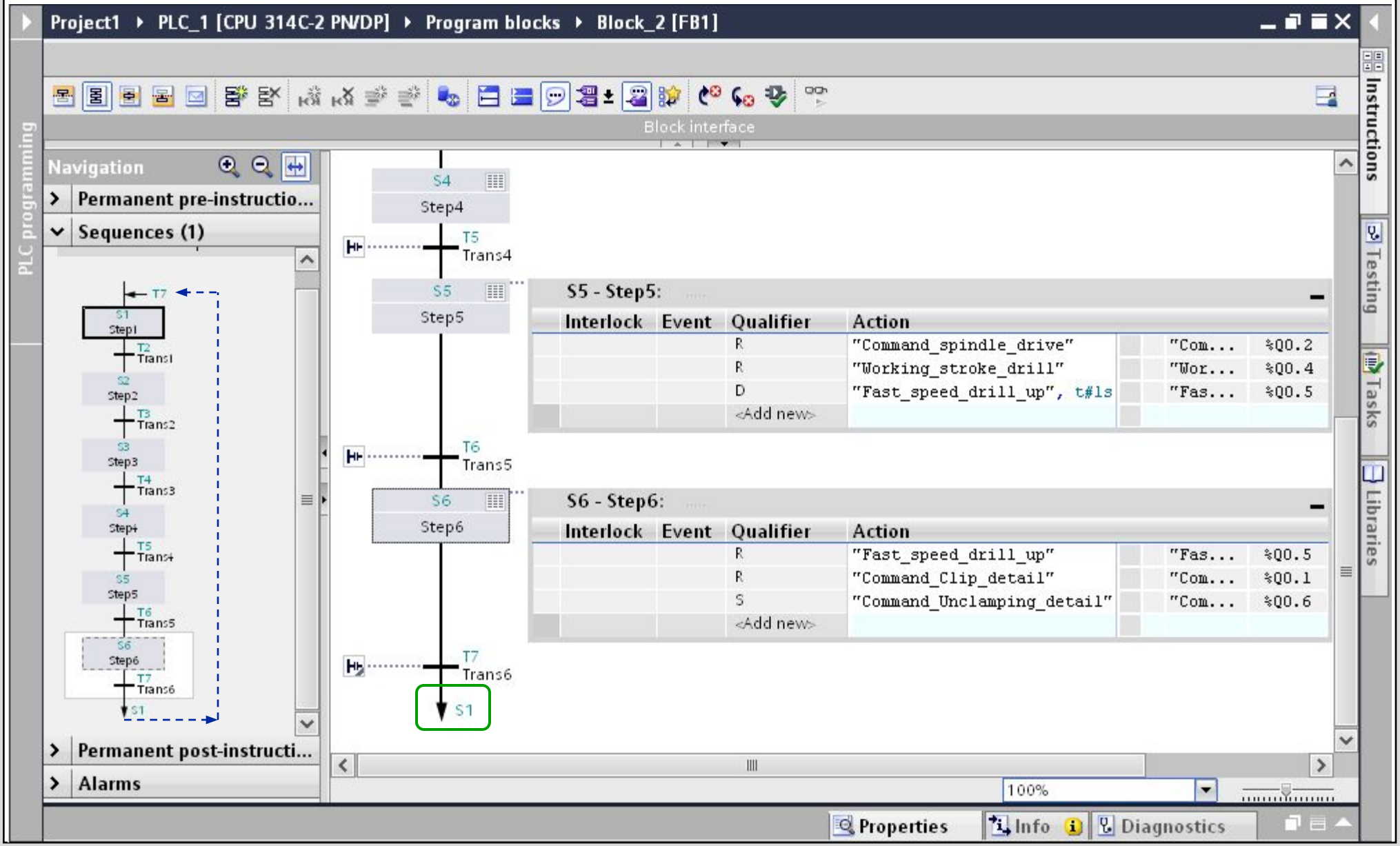
Написание условий для выполнения очередного шага



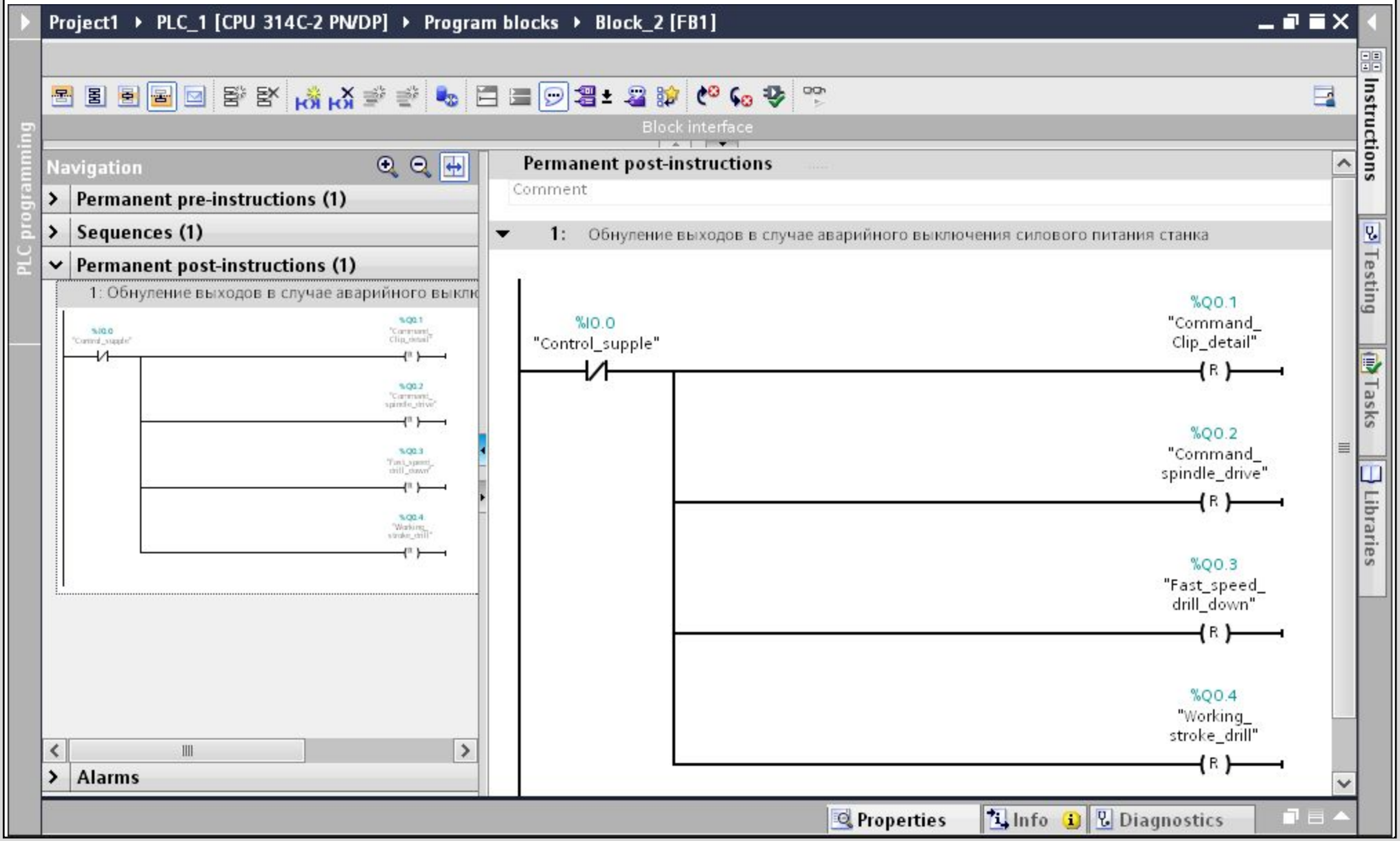
Действия, выполняемые в шагах Step 2 и Step 3



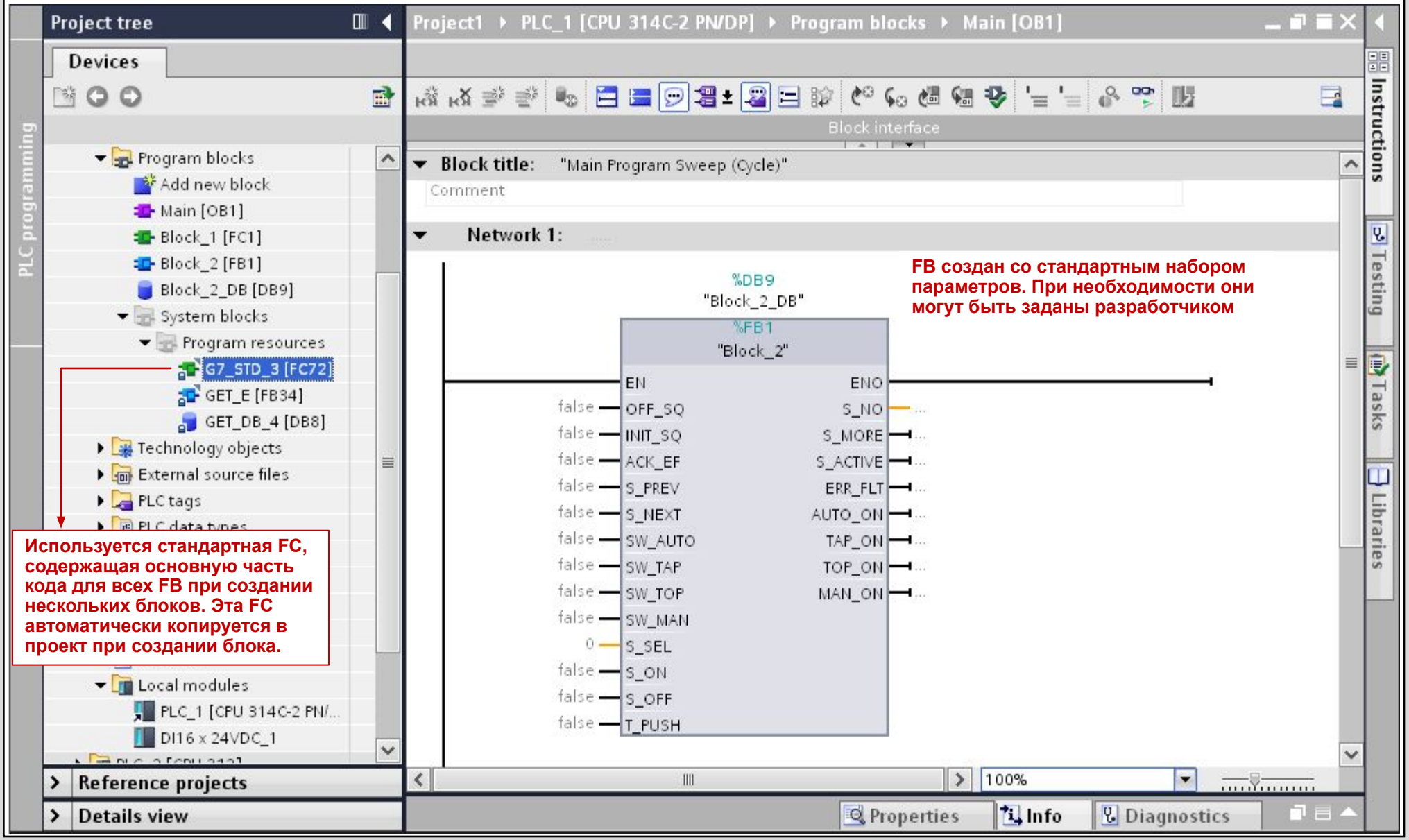
Завершающий шаг, переход к шагу Step 1



Написание инструкций в разделе Permanent post-instructions



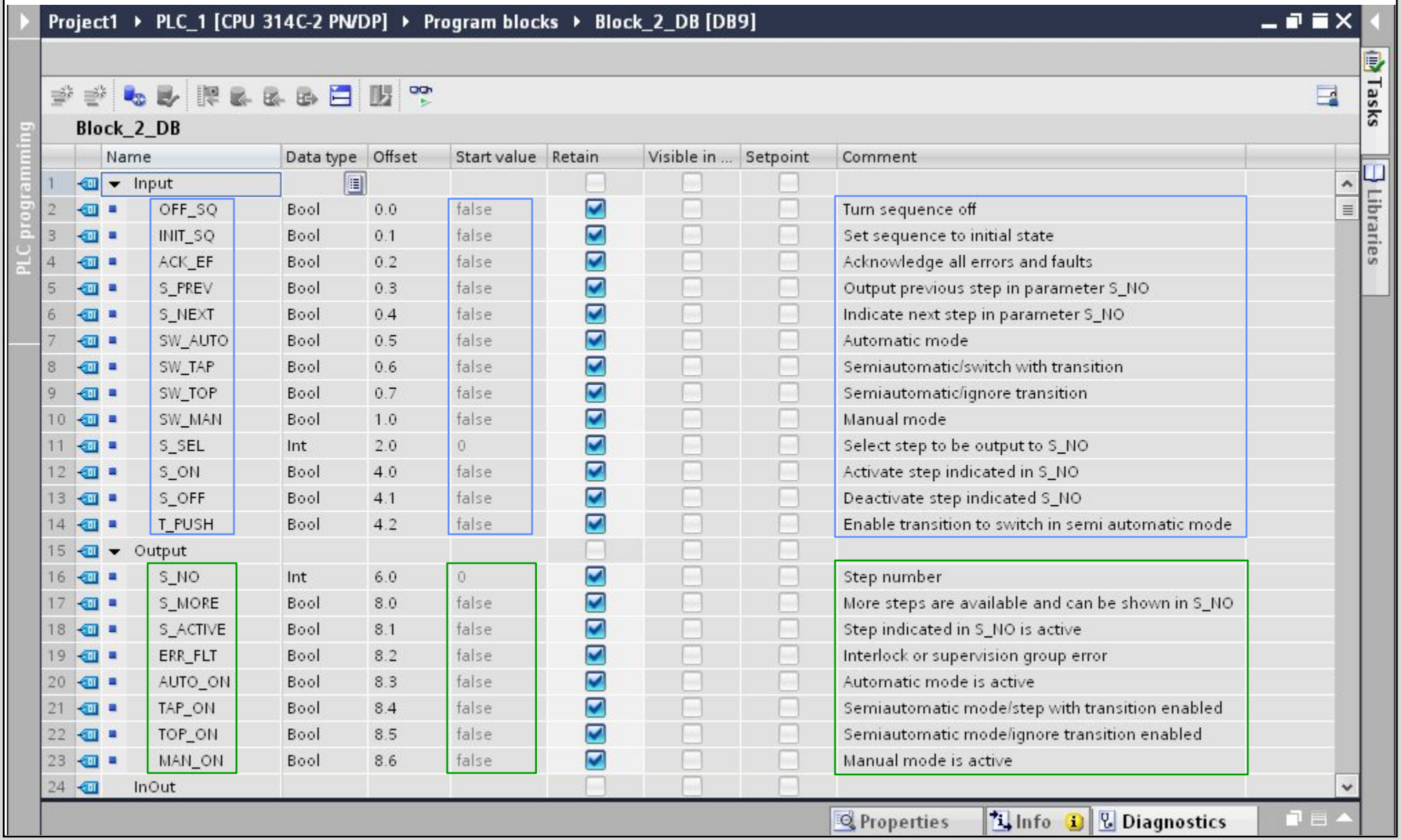
Вызов созданного блока



FB создан со стандартным набором параметров. При необходимости они могут быть заданы разработчиком

Используется стандартная FC, содержащая основную часть кода для всех FB при создании нескольких блоков. Эта FC автоматически копируется в проект при создании блока.

Смысл и значение параметров Input/Output вызываемого блока



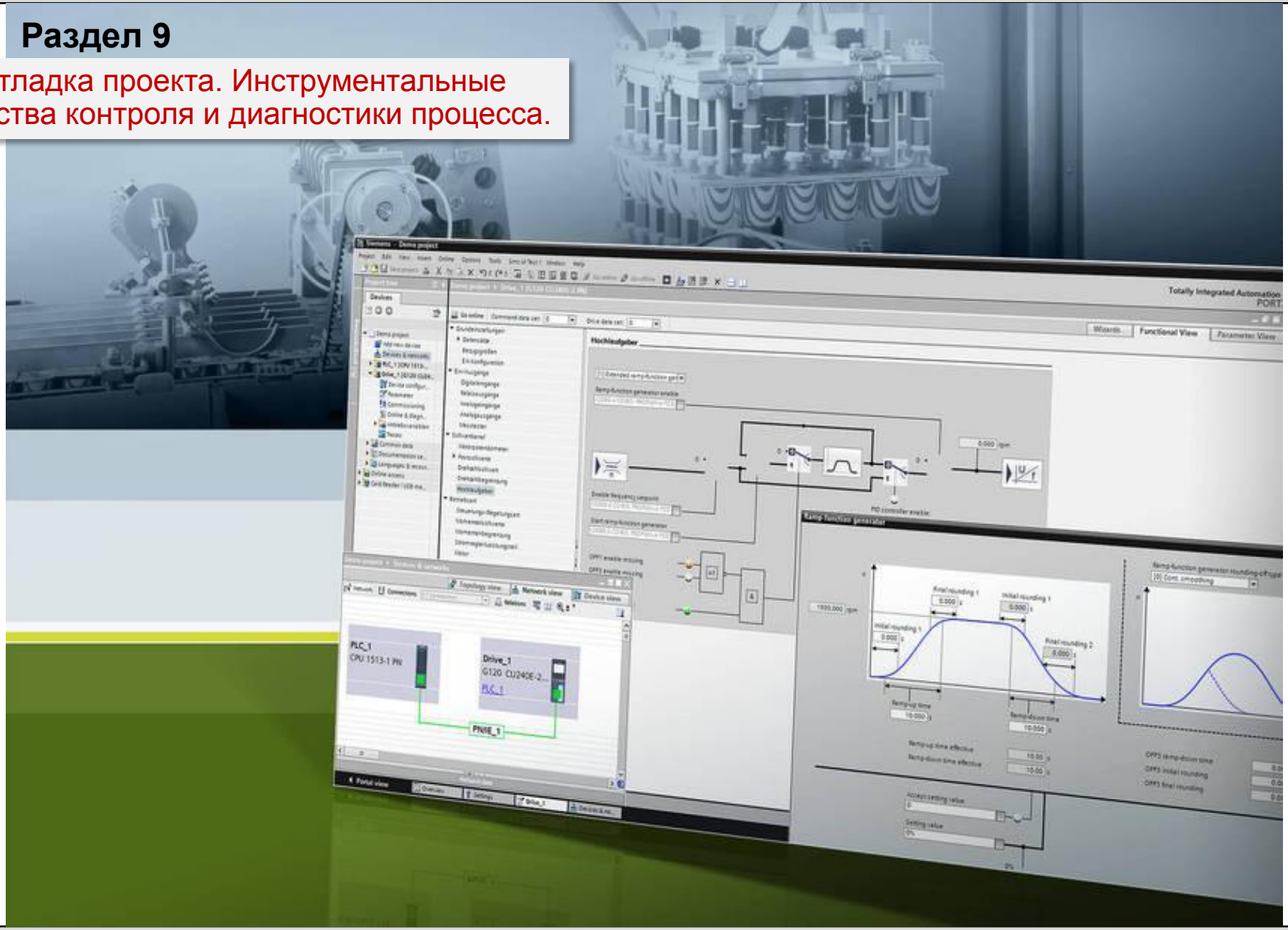
	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint	Comment
1	Input							
2	OFF_SQ	Bool	0.0	false	☑			Turn sequence off
3	INIT_SQ	Bool	0.1	false	☑			Set sequence to initial state
4	ACK_EF	Bool	0.2	false	☑			Acknowledge all errors and faults
5	S_PREV	Bool	0.3	false	☑			Output previous step in parameter S_NO
6	S_NEXT	Bool	0.4	false	☑			Indicate next step in parameter S_NO
7	SW_AUTO	Bool	0.5	false	☑			Automatic mode
8	SW_TAP	Bool	0.6	false	☑			Semiautomatic/switch with transition
9	SW_TOP	Bool	0.7	false	☑			Semiautomatic/ignore transition
10	SW_MAN	Bool	1.0	false	☑			Manual mode
11	S_SEL	Int	2.0	0	☑			Select step to be output to S_NO
12	S_ON	Bool	4.0	false	☑			Activate step indicated in S_NO
13	S_OFF	Bool	4.1	false	☑			Deactivate step indicated S_NO
14	T_PUSH	Bool	4.2	false	☑			Enable transition to switch in semi automatic mode
15	Output							
16	S_NO	Int	6.0	0	☑			Step number
17	S_MORE	Bool	8.0	false	☑			More steps are available and can be shown in S_NO
18	S_ACTIVE	Bool	8.1	false	☑			Step indicated in S_NO is active
19	ERR_FLT	Bool	8.2	false	☑			Interlock or supervision group error
20	AUTO_ON	Bool	8.3	false	☑			Automatic mode is active
21	TAP_ON	Bool	8.4	false	☑			Semiautomatic mode/step with transition enabled
22	TOP_ON	Bool	8.5	false	☑			Semiautomatic mode/ignore transition enabled
23	MAN_ON	Bool	8.6	false	☑			Manual mode is active
24	InOut							

Конец раздела 8. Окно навигации

- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 9

Отладка проекта. Инструментальные средства контроля и диагностики процесса.



Классификация ошибок и инструментов отладки

Ошибки, определяемые системой

Запись, индикация и оценка ошибок в контроллере (как правило: CPU -> ”STOP“)

- **Ошибка в модуле**
- **Короткое замыкание в сигнальном кабеле**
- **Превышение времени цикла**
- **Ошибка программирования, например, вызов несуществующего блока.**

Инструменты отладки:

- **Module Information**
  - **Диагностический буфер**
  - **Стек прерываний**
  - **Стек блоков**
  - **Локальный стек**
- **Hardware Diagnostics**

Ошибки функционирования

Необходимые функции не выполняются или выполняются с ошибкой

- **Ошибки процесса (датчик/исполнительное устройство, дефект проводки)**
- **Логические ошибки программирования, не обнаруженные при отладке**

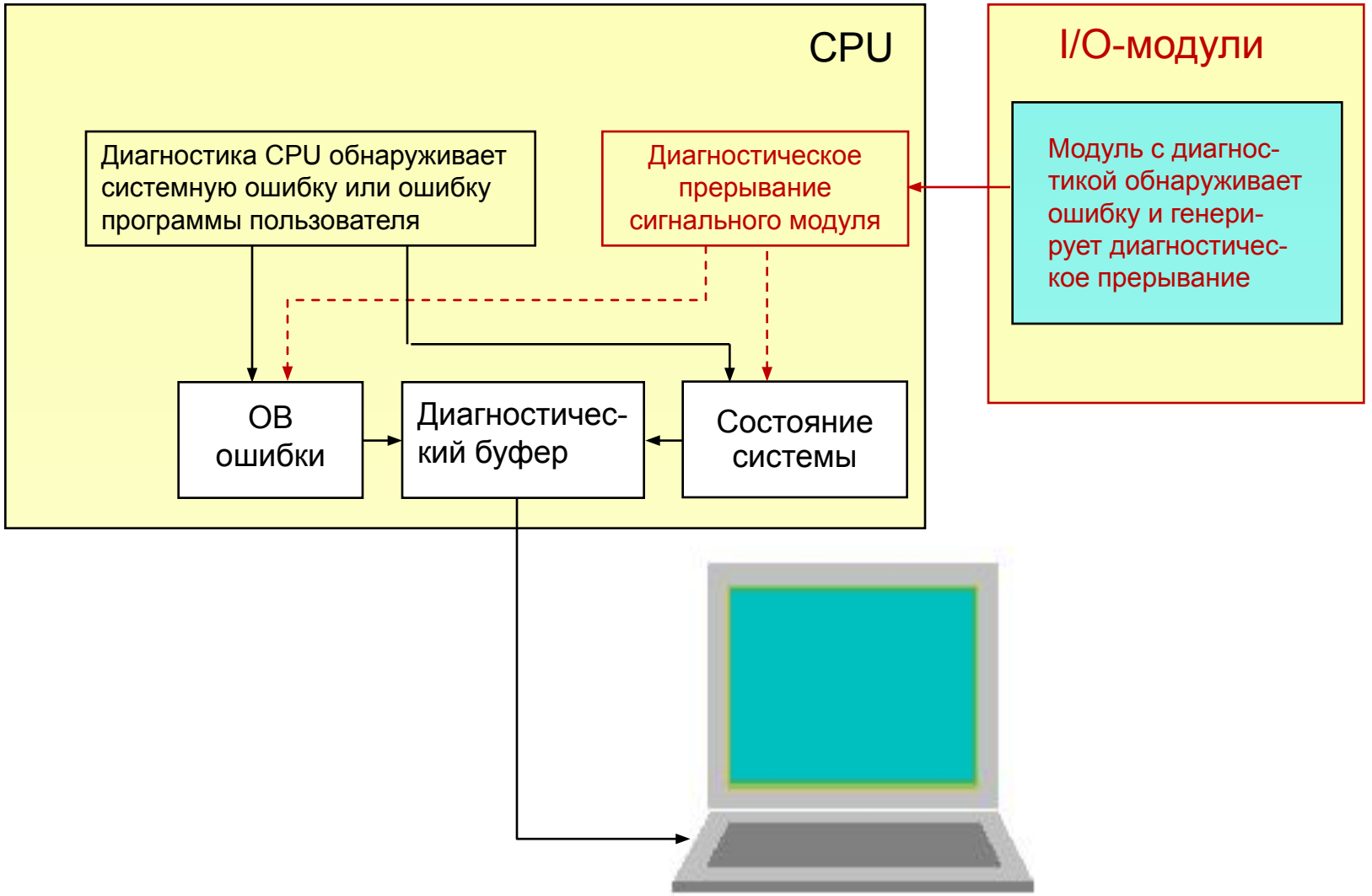
Инструменты отладки :

- **Watch table** — **Управление переменными**
- **Monitoring on/off** — **Просмотр программы OnLine**
- **Reference Data** — **Справочные данные**
  - **Cross References** — **Перекрестные ссылки**
  - **Assignment list** — **План использования переменных**
  - **Call Structure** — **Структура вызовов**

Дополнительные возможности поиска ошибок:

- **Функция “Force“**
- **Сравнение блоков в проекте и в ПЛК**
- **Использование точек останова (Breakpoint)**

Системная диагностика



Настройка соединения ПК - ПЛК

Маска подсети: 255.255.255. 0

IP – адрес: 192.168. 0 . 1

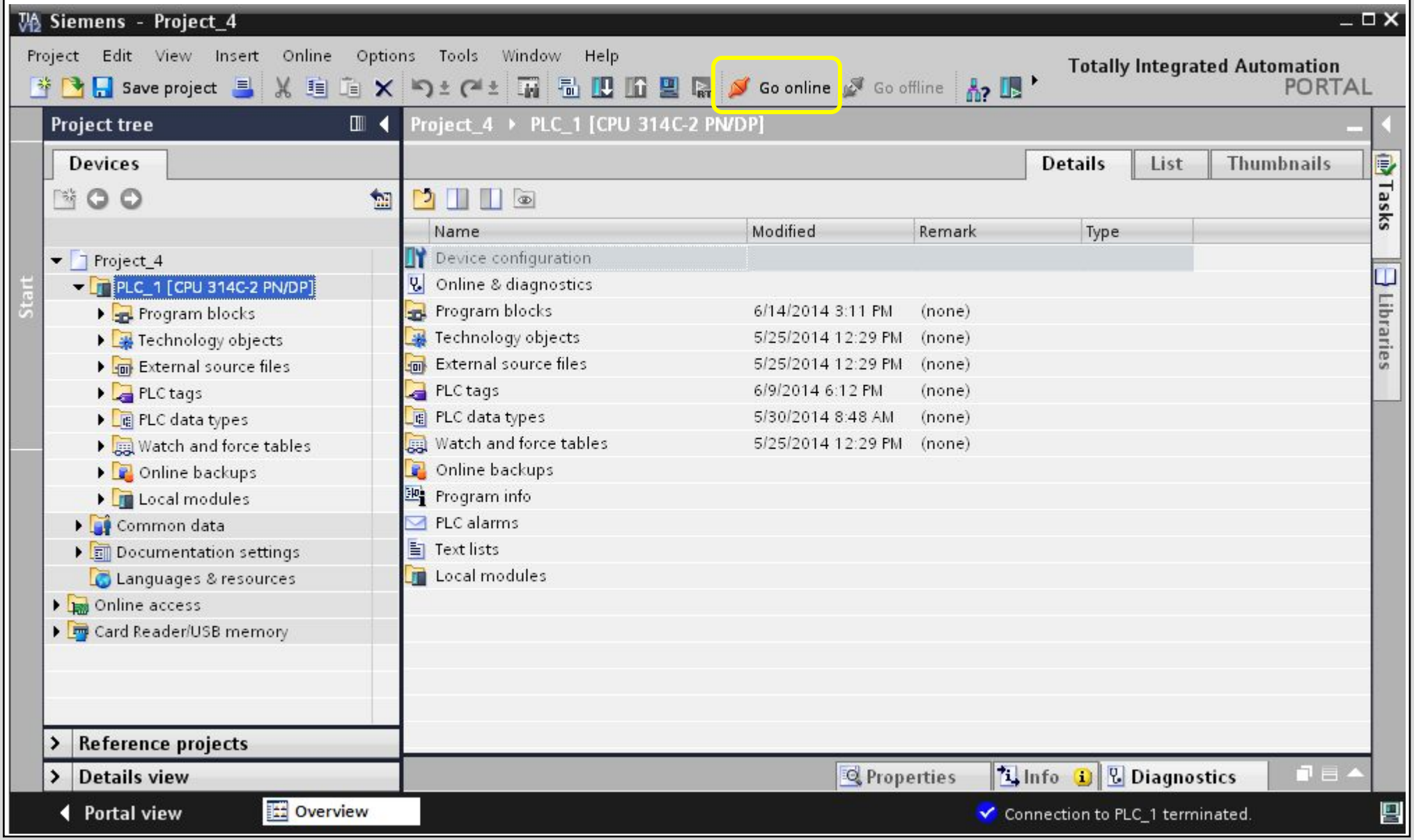
Адрес сети — Подсеть — Адрес хоста

Маска подсети :
255.255.255. 0

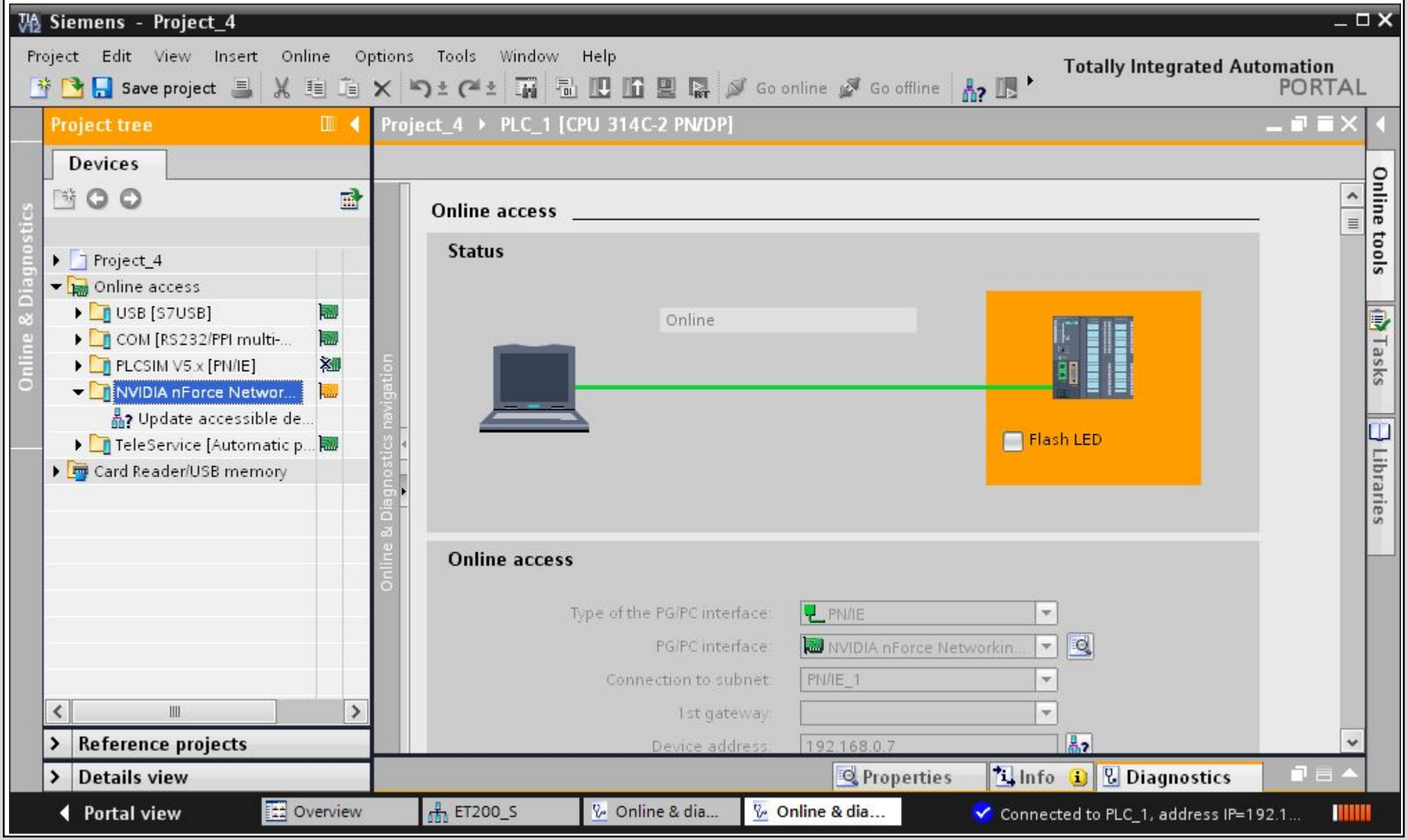
IP – адрес: 192.168. 0 . 2

Чтобы ПК и ПЛК увидели друг друга, они должны быть в одной сети и подсети

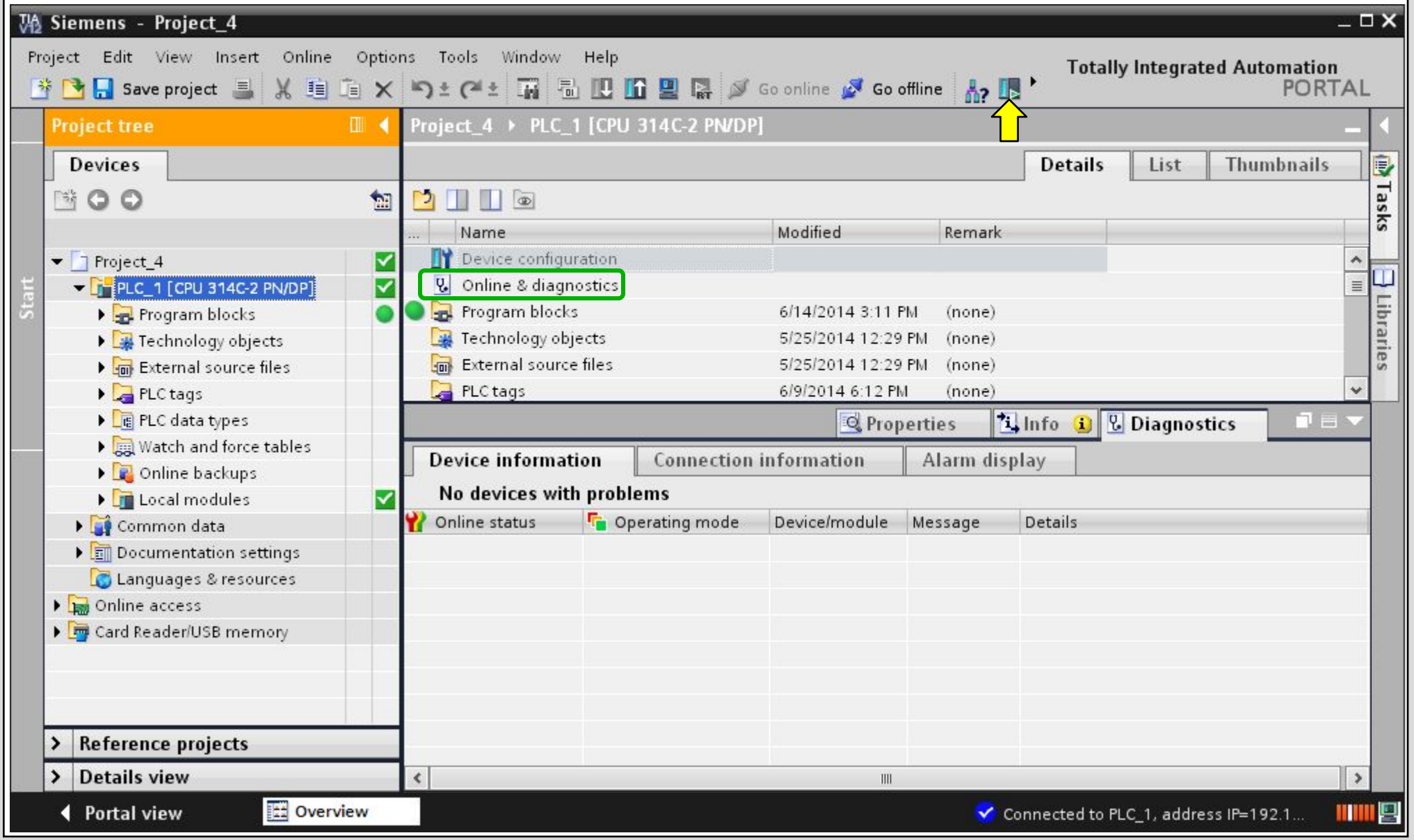
Переход в режим On-line



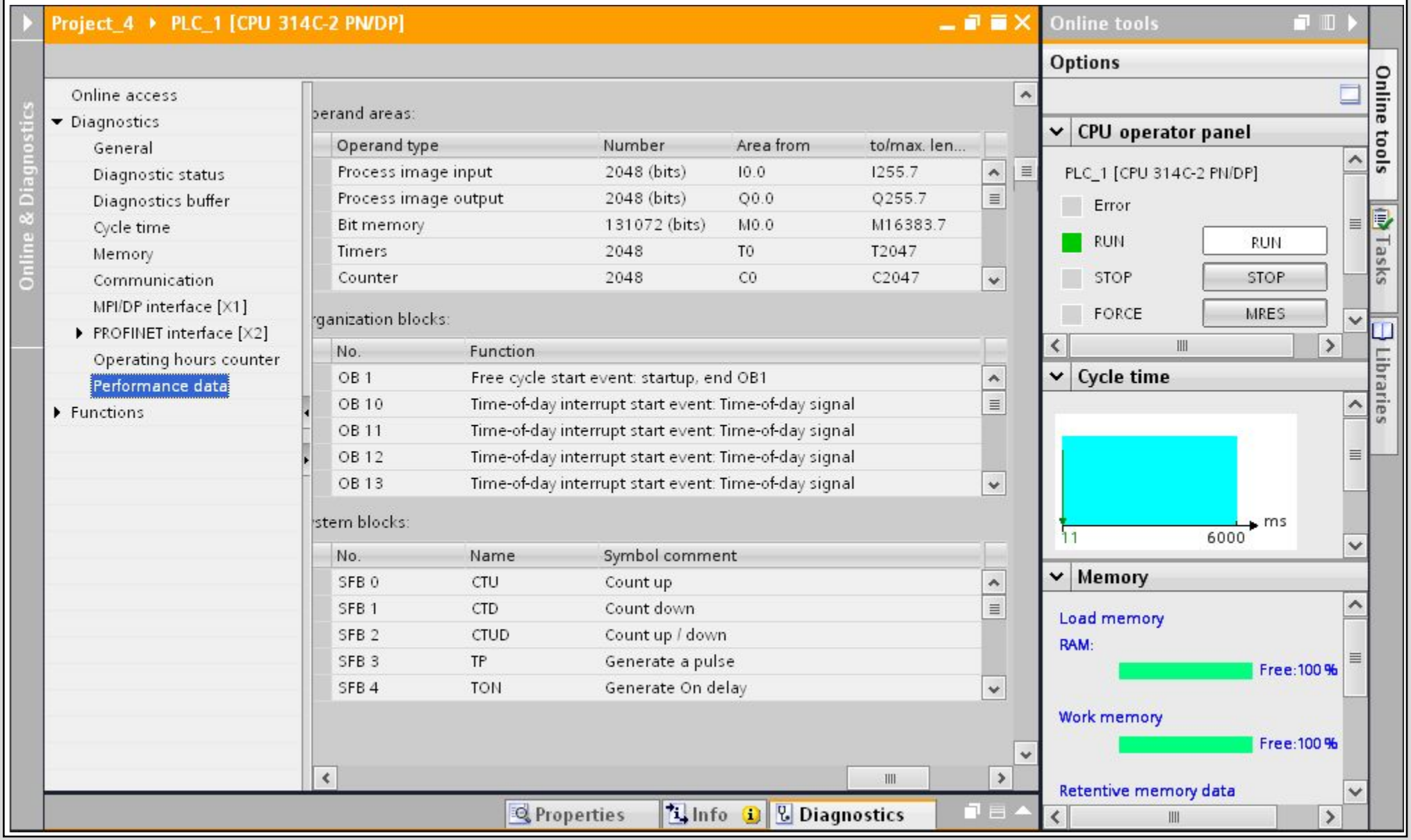
Переход в режим On-line



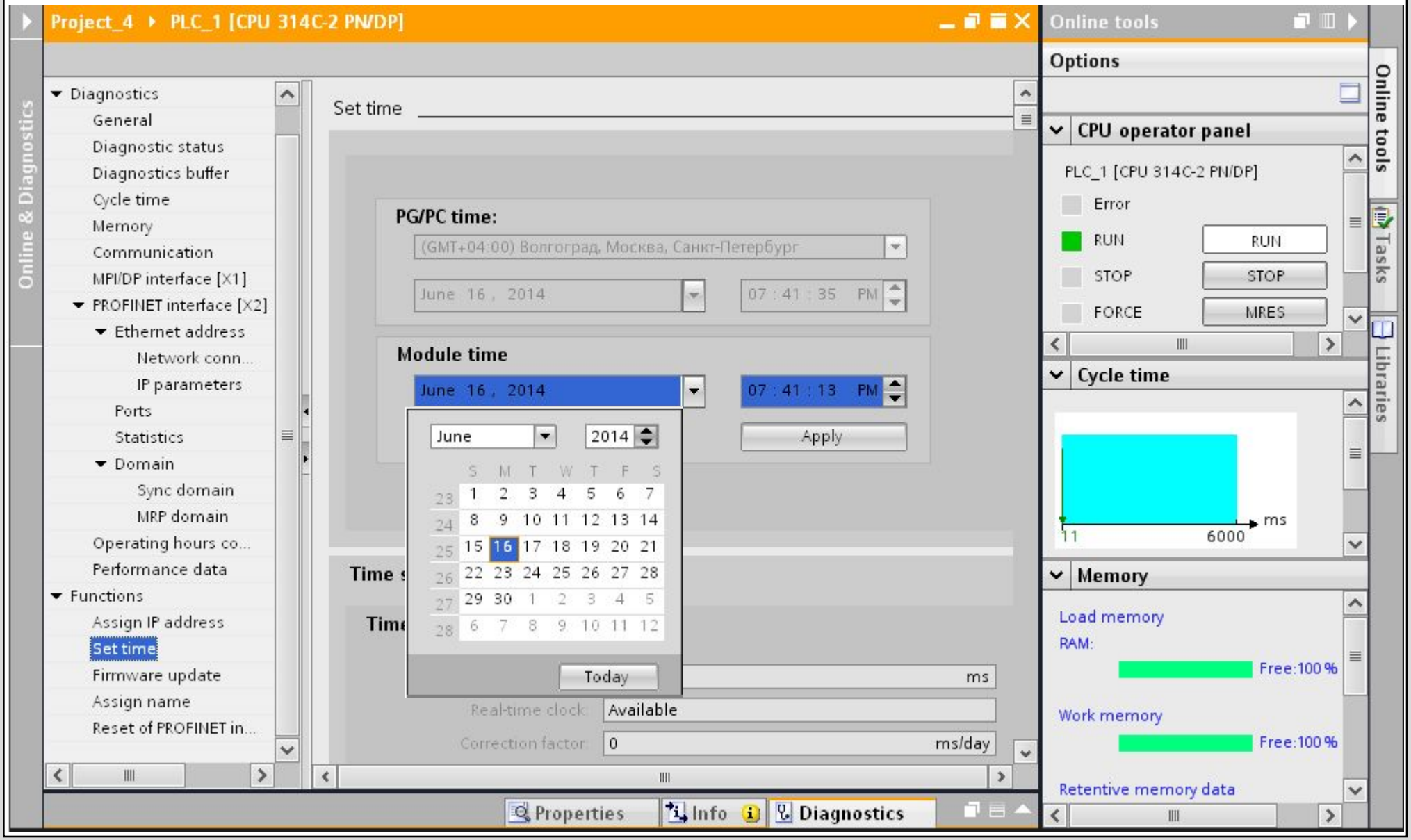
Анализ ЦПУ в режиме On-line



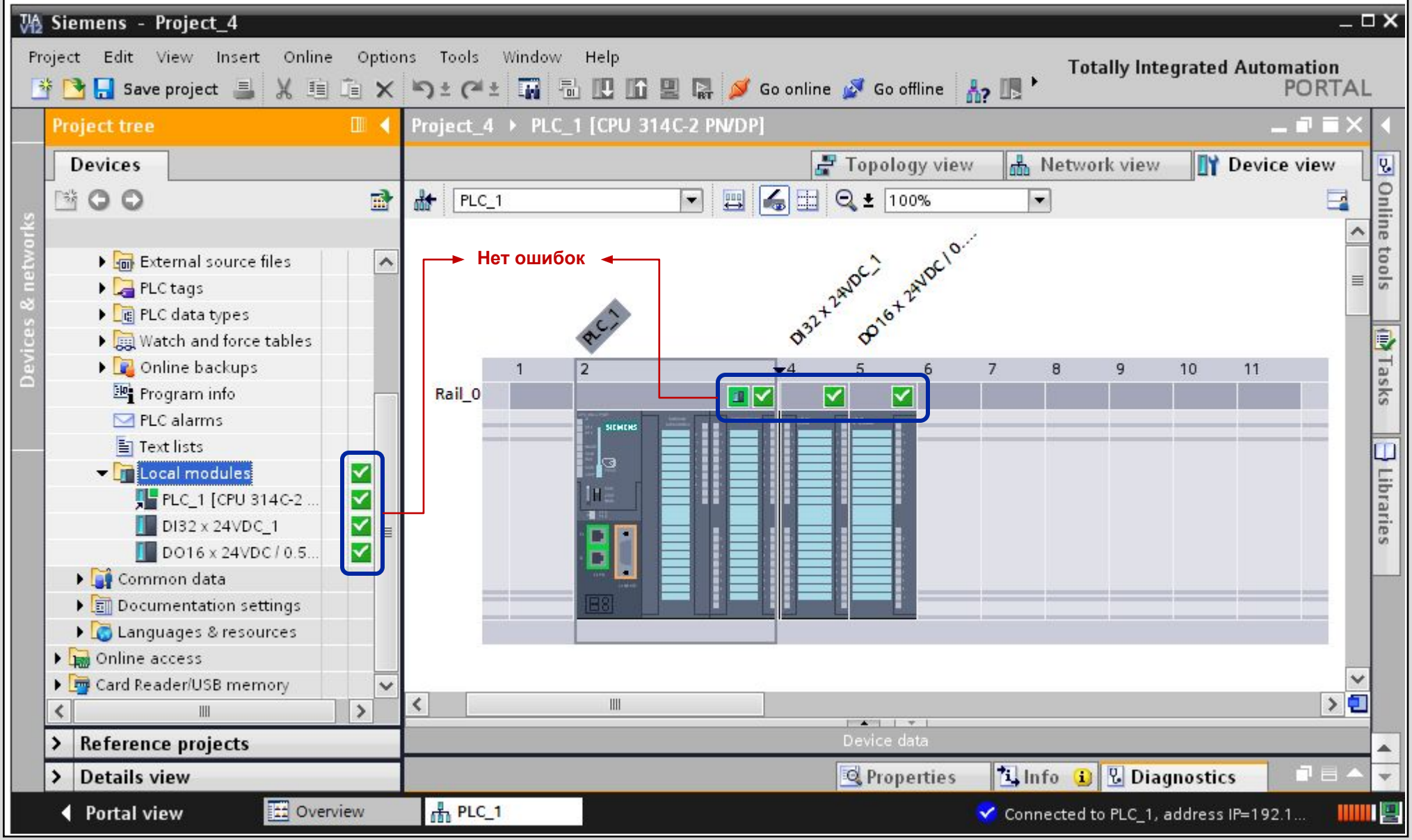
Просмотр ресурсов ЦПУ



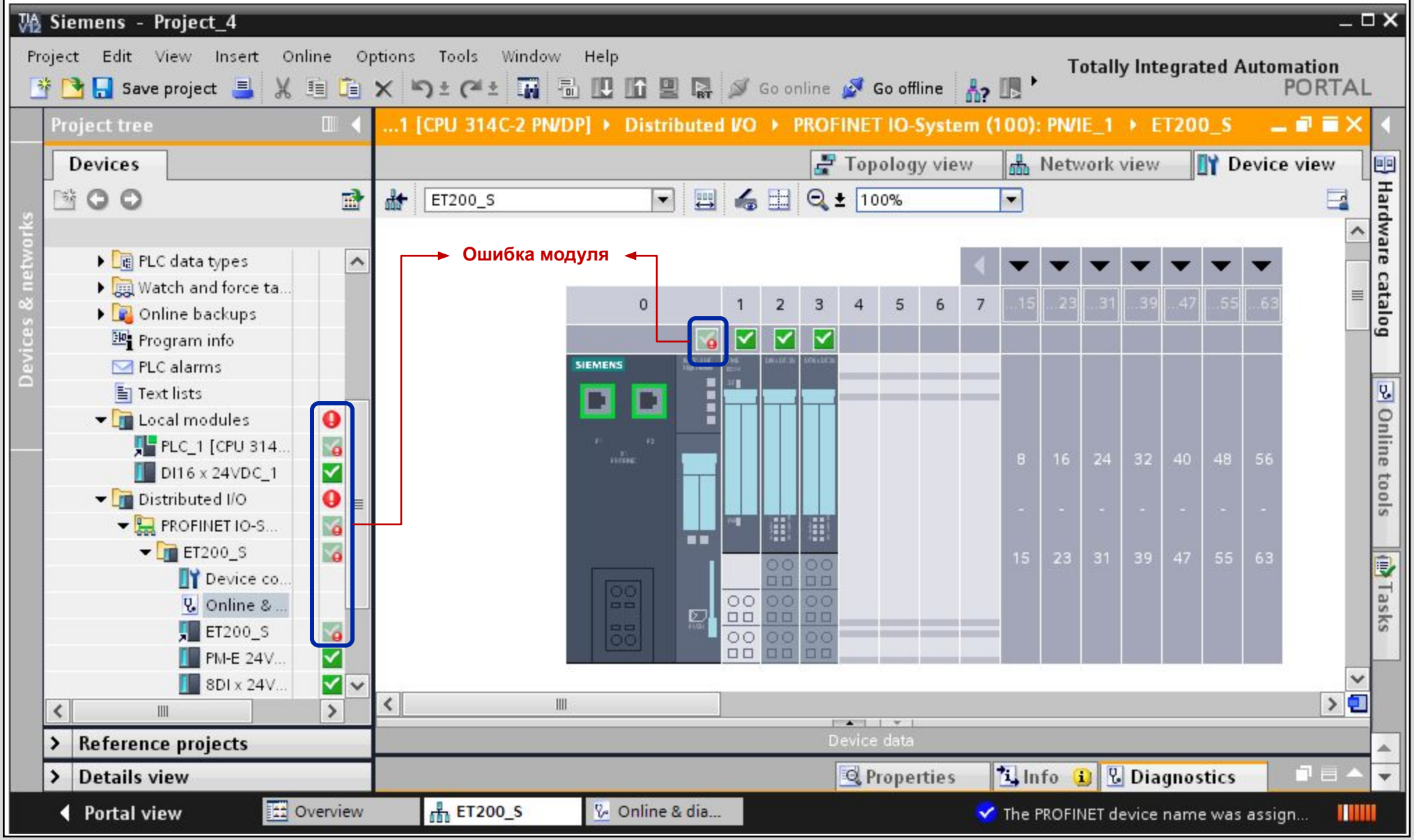
Установка системного времени



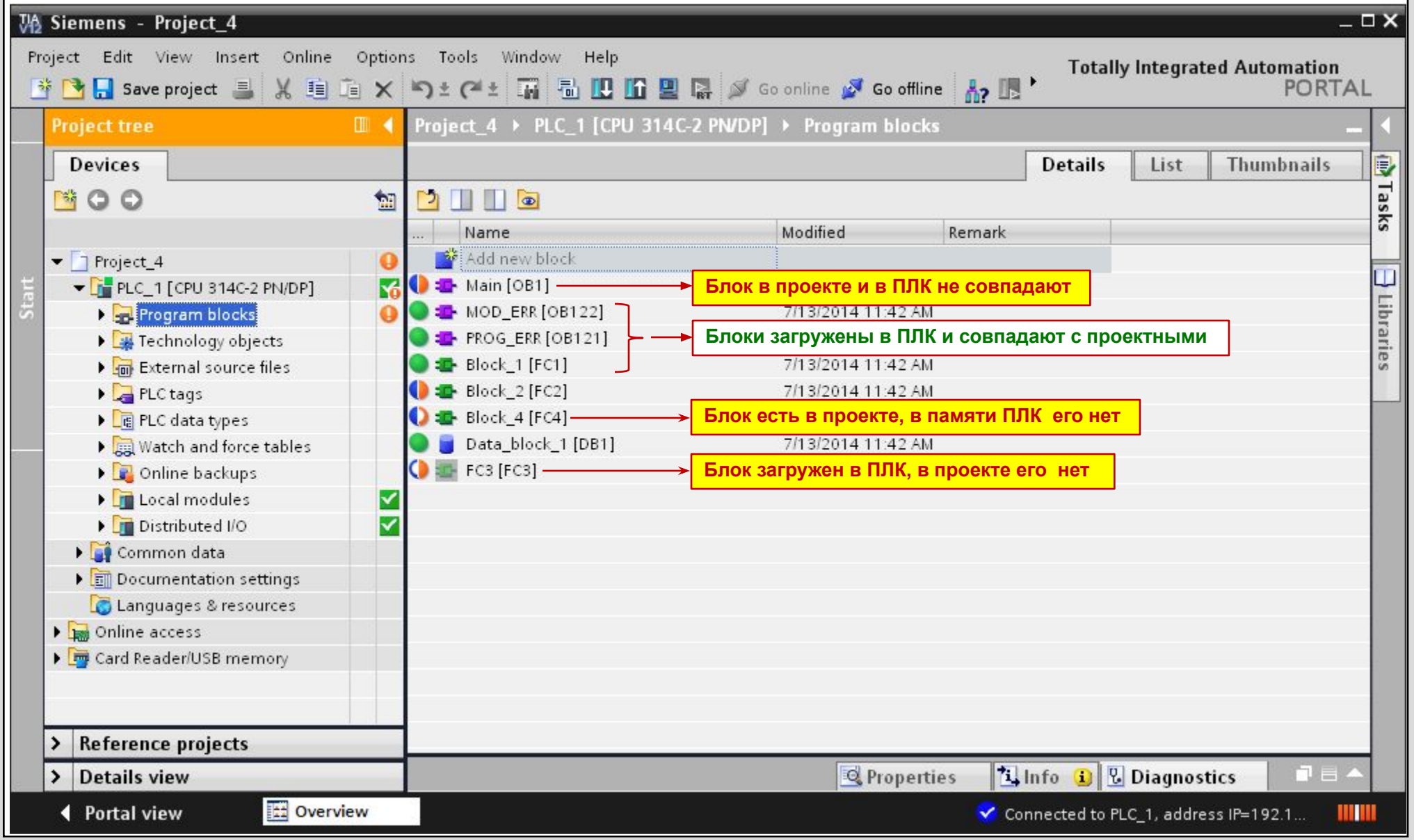
Просмотр состояния модулей в режиме On-line



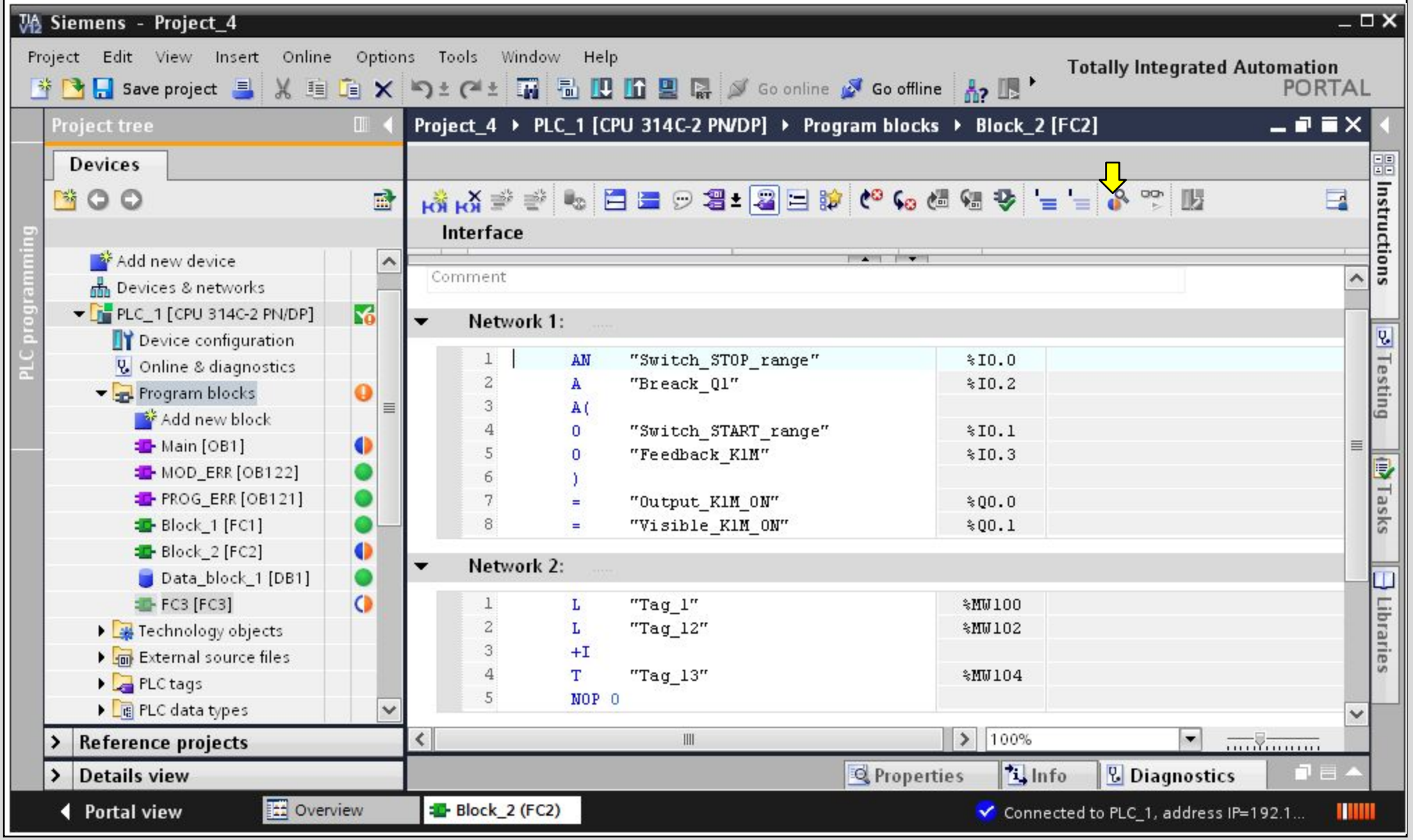
Просмотр состояния модулей в режиме On-line



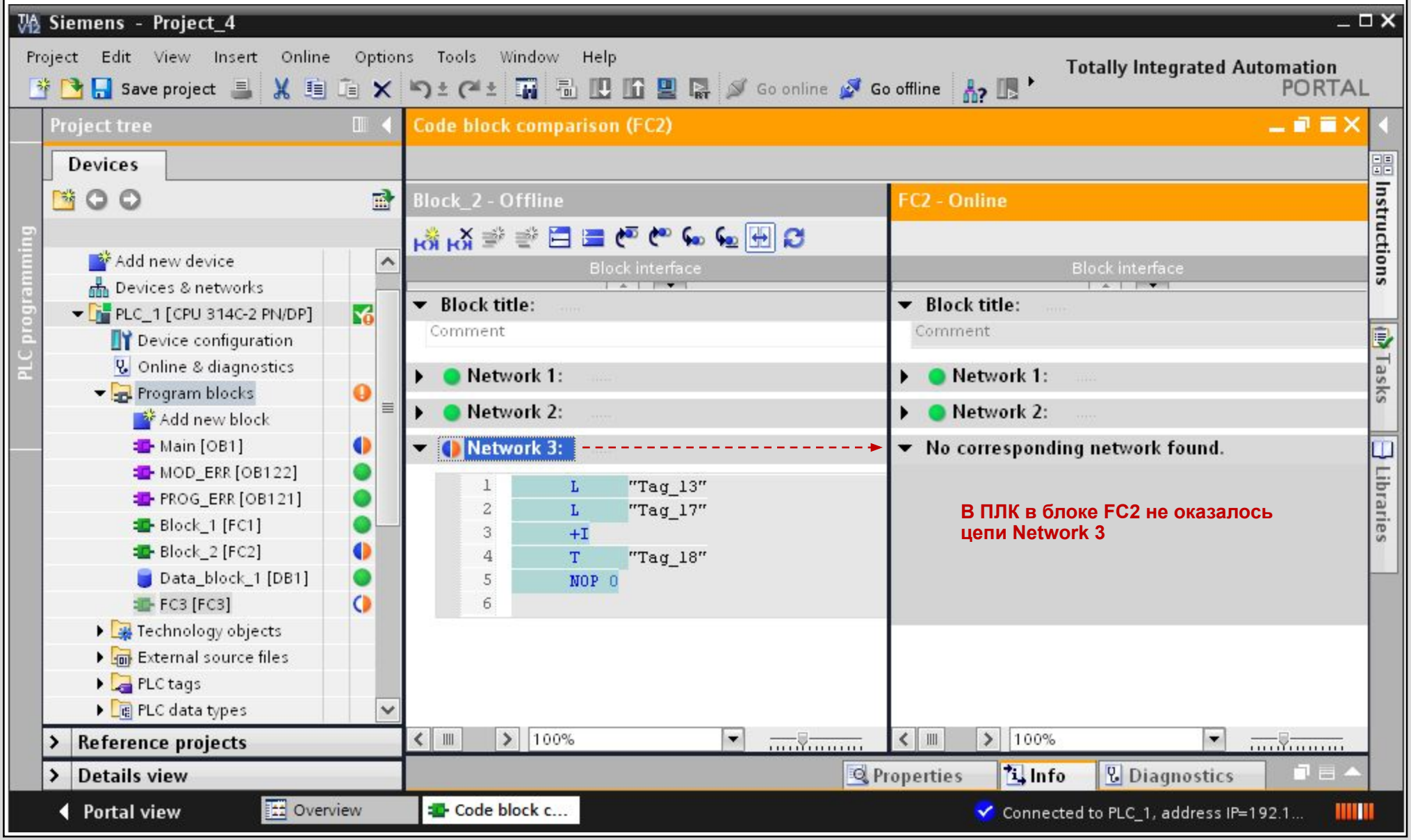
Просмотр и контроль программных блоков в режиме On-line



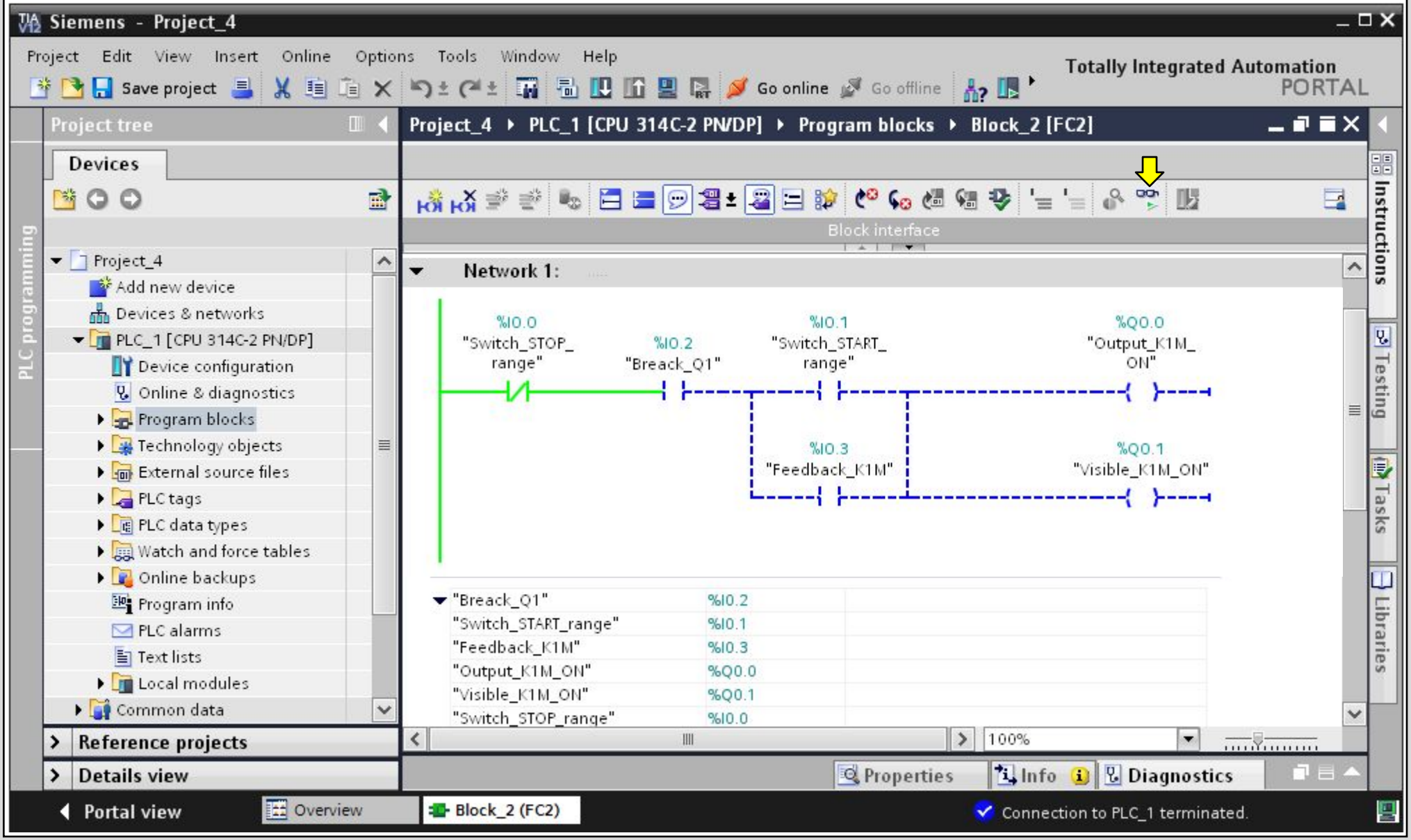
Сравнение открытого блока на идентичность.



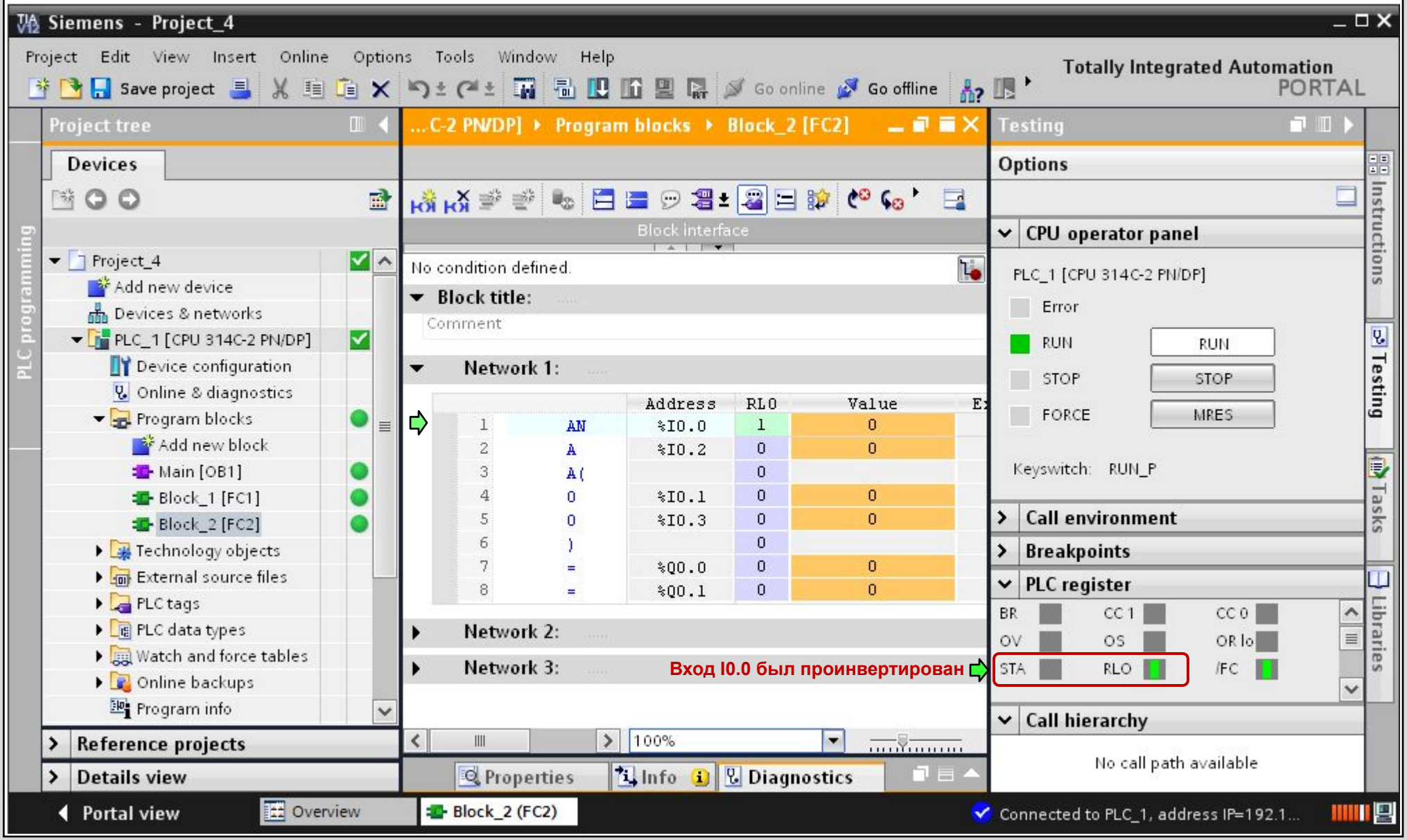
Сравнение открытого блока на идентичность.



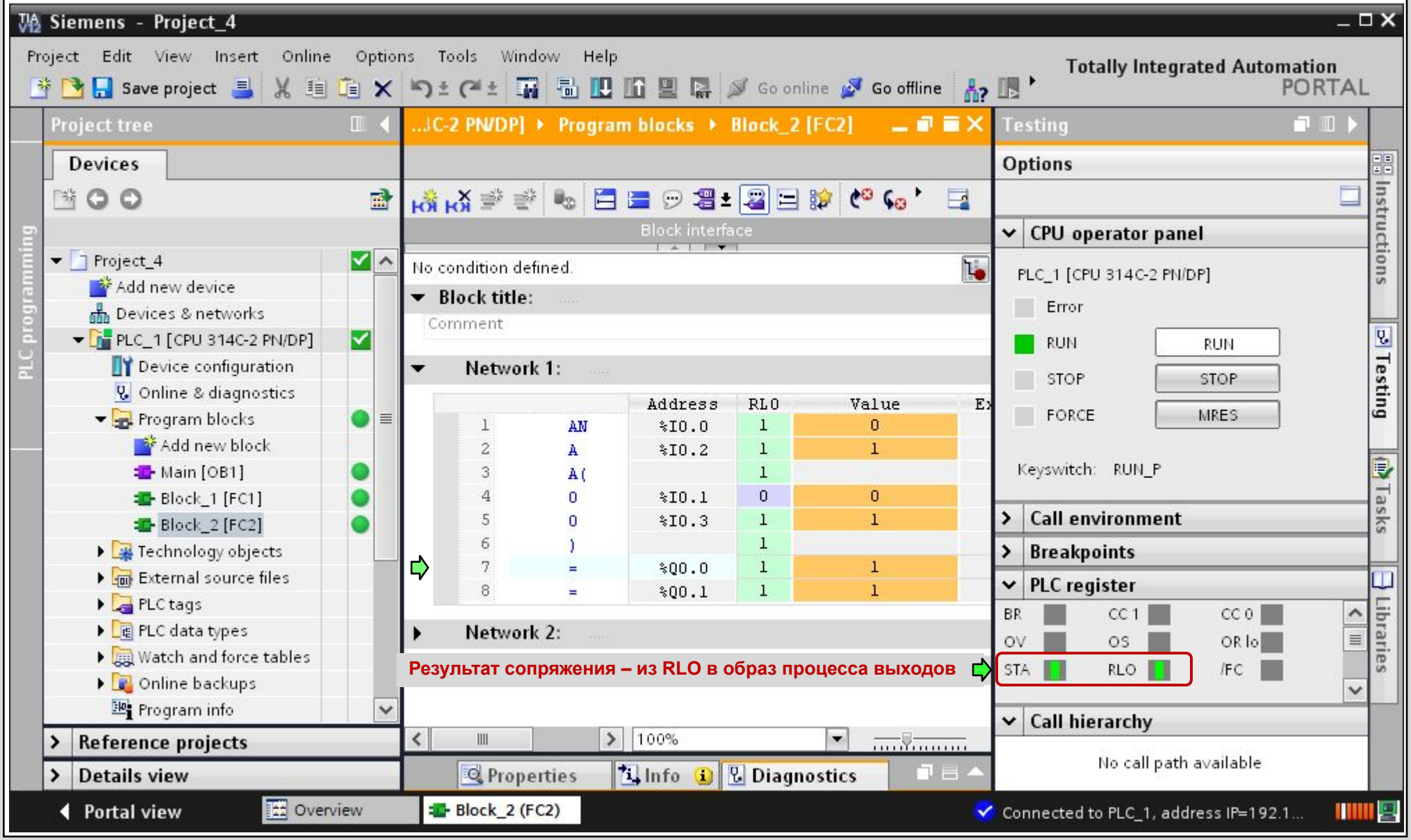
Просмотр состояния переменных в цепи в режиме On-line



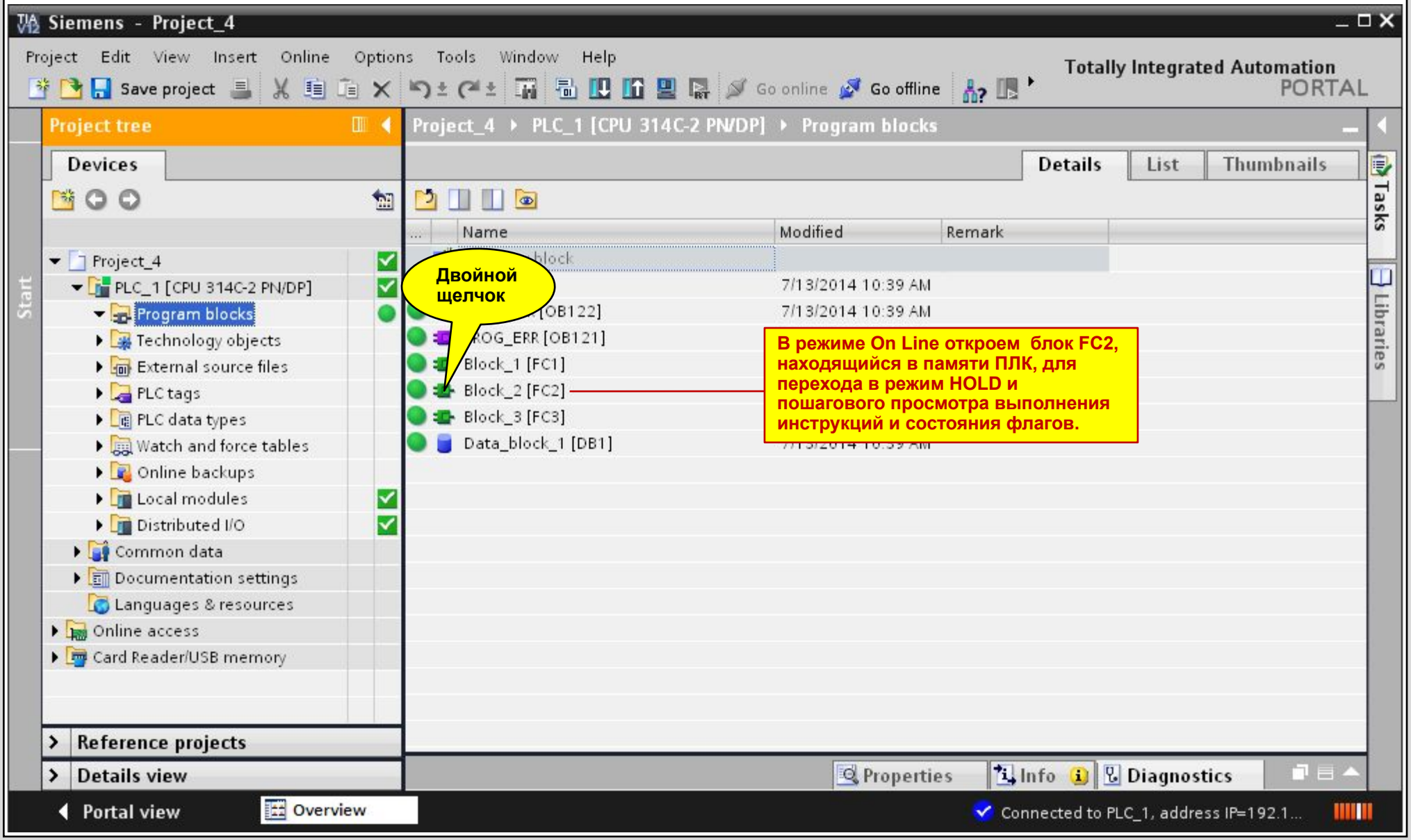
Просмотр состояния переменных в цепи в режиме On-line



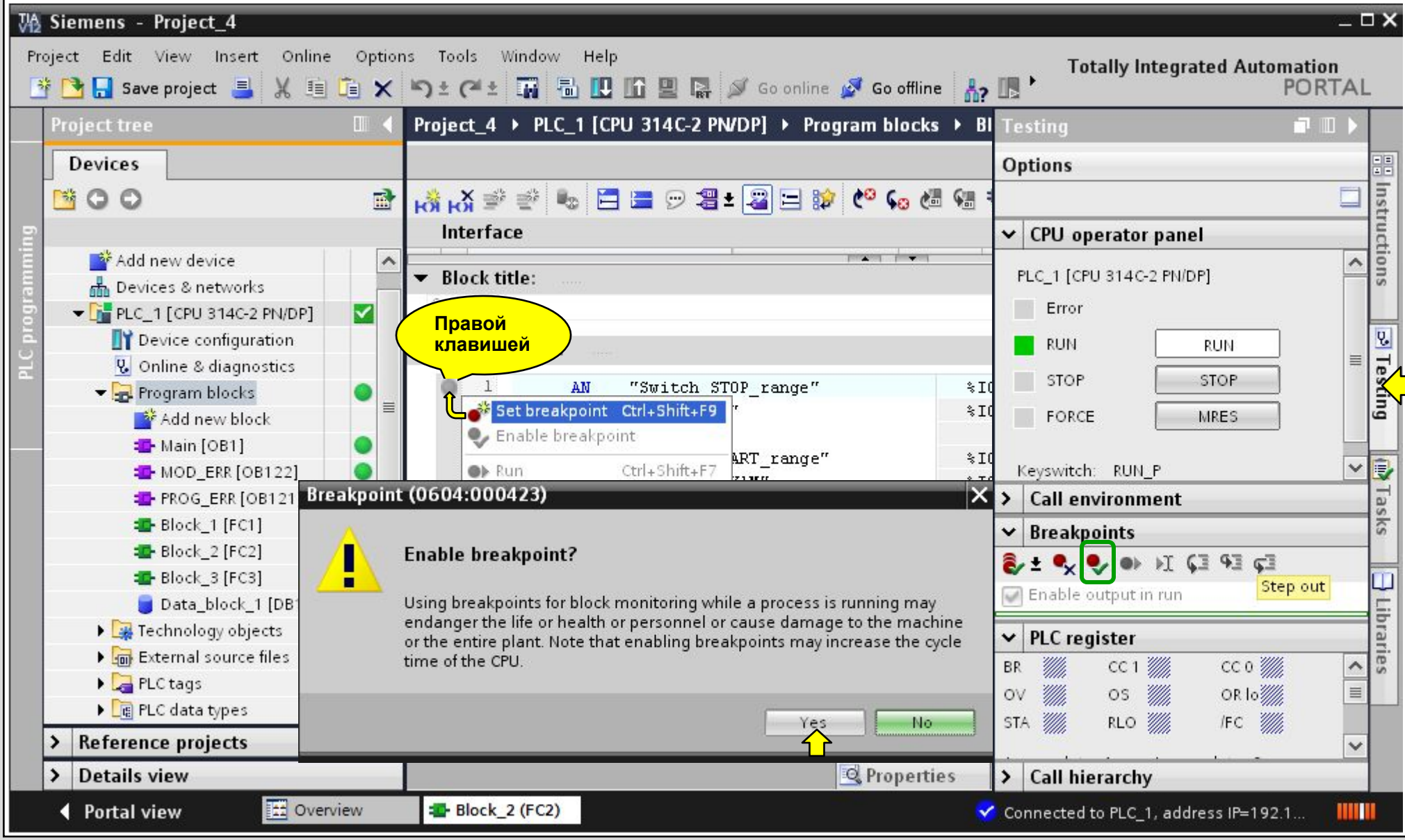
Просмотр состояния переменных в цепи в режиме On-line



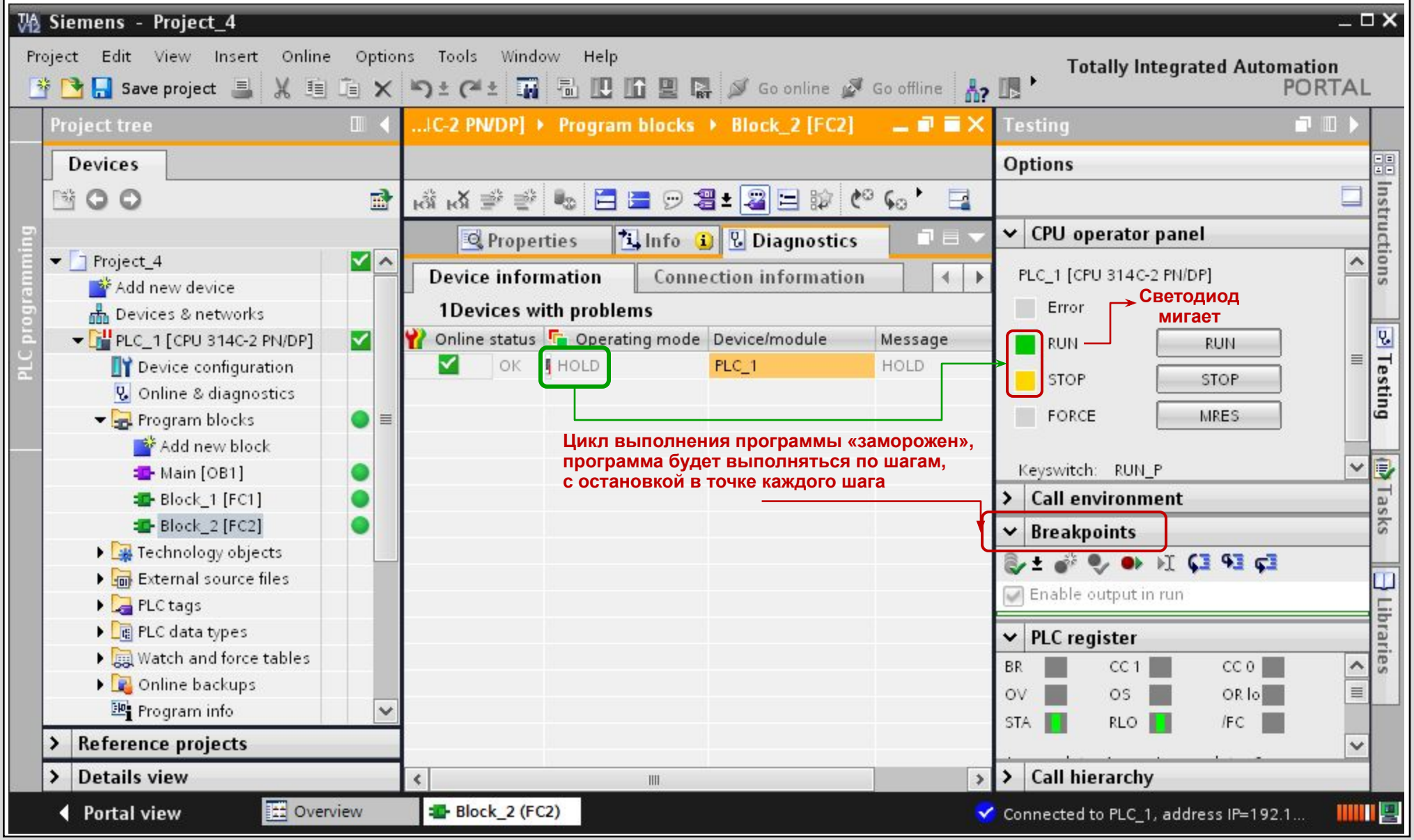
Просмотр программных блоков в памяти ПЛК в режиме On-Line



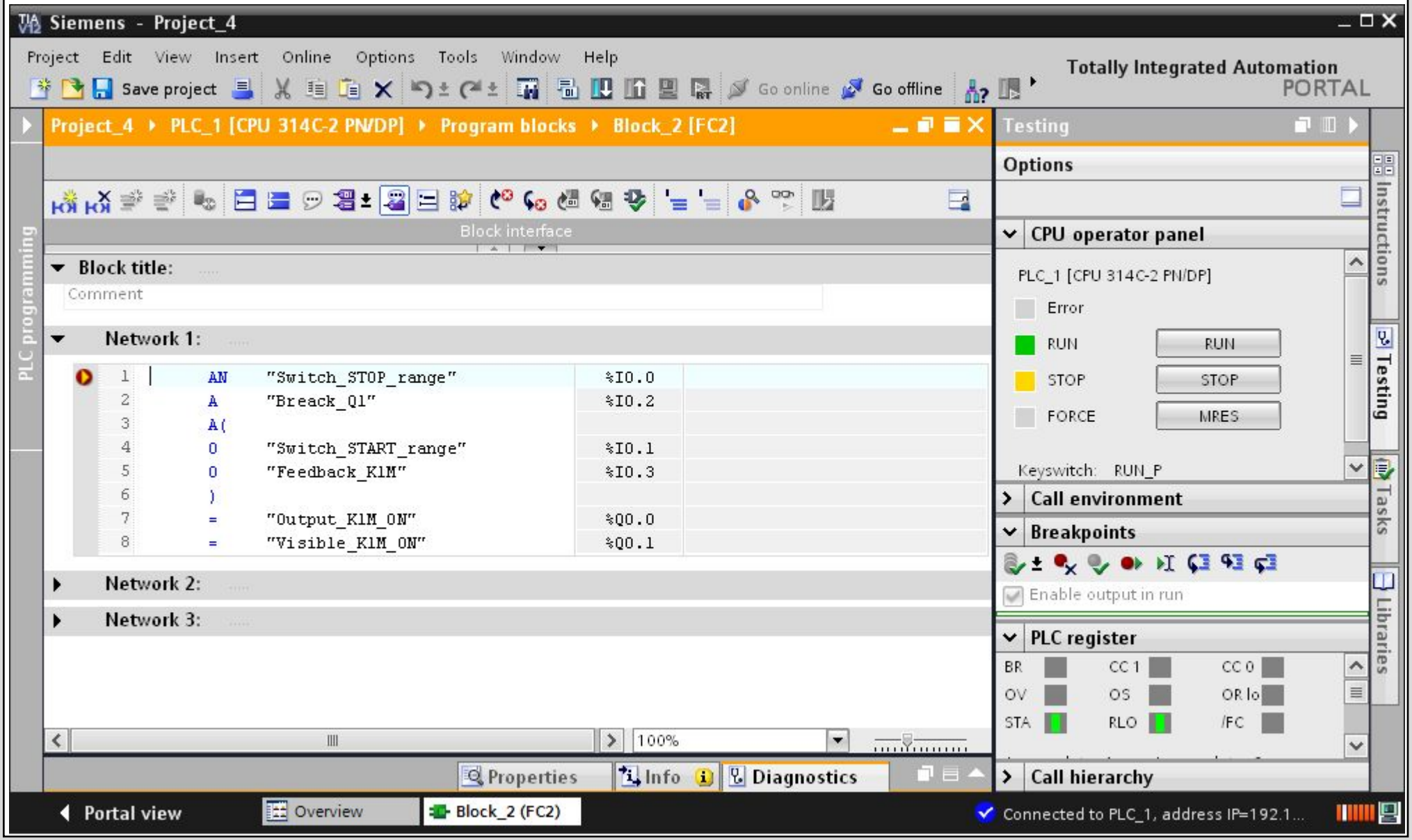
Просмотр инструкций и флагов в цепи в режиме HOLD



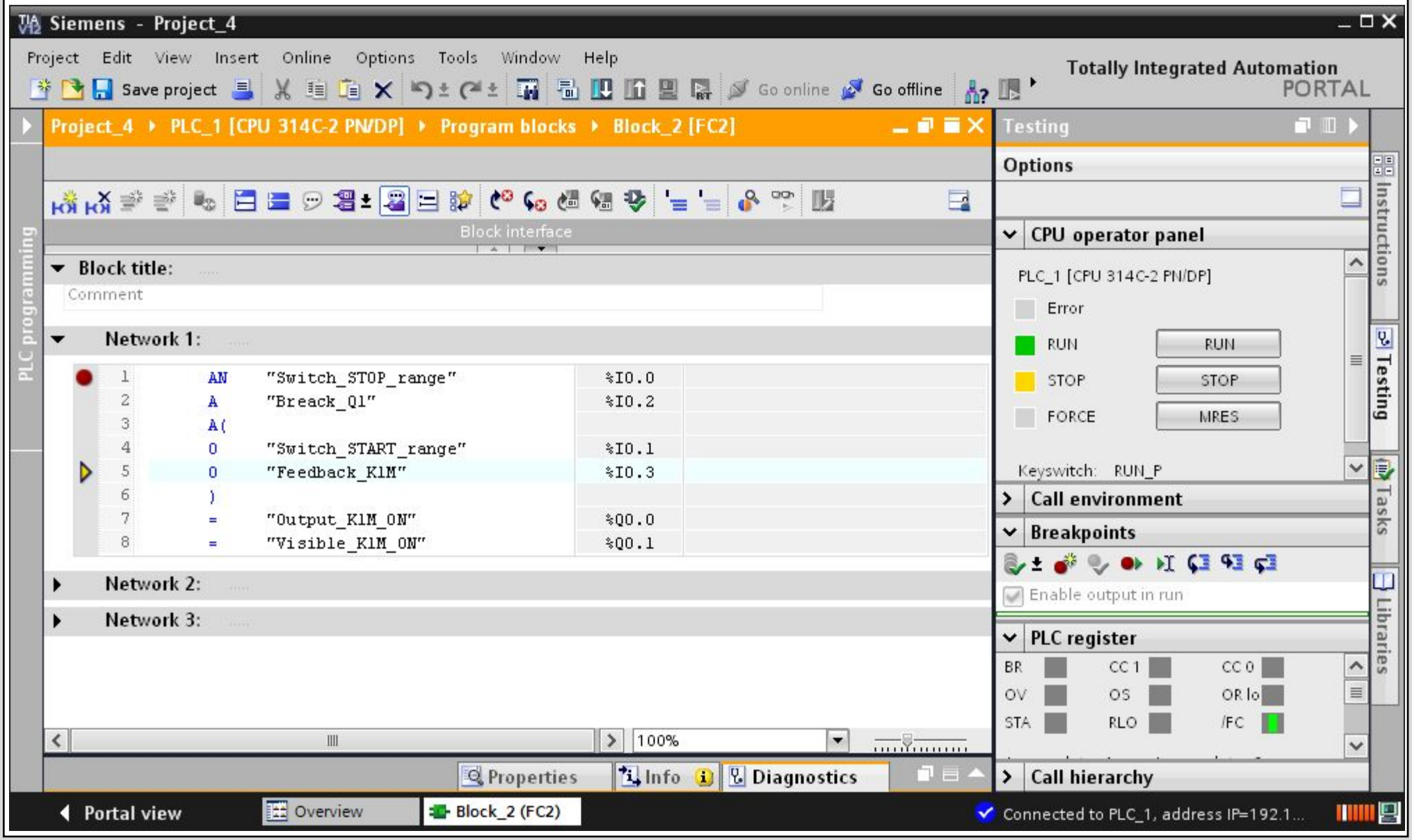
Просмотр инструкций и флагов в цепи в режиме HOLD



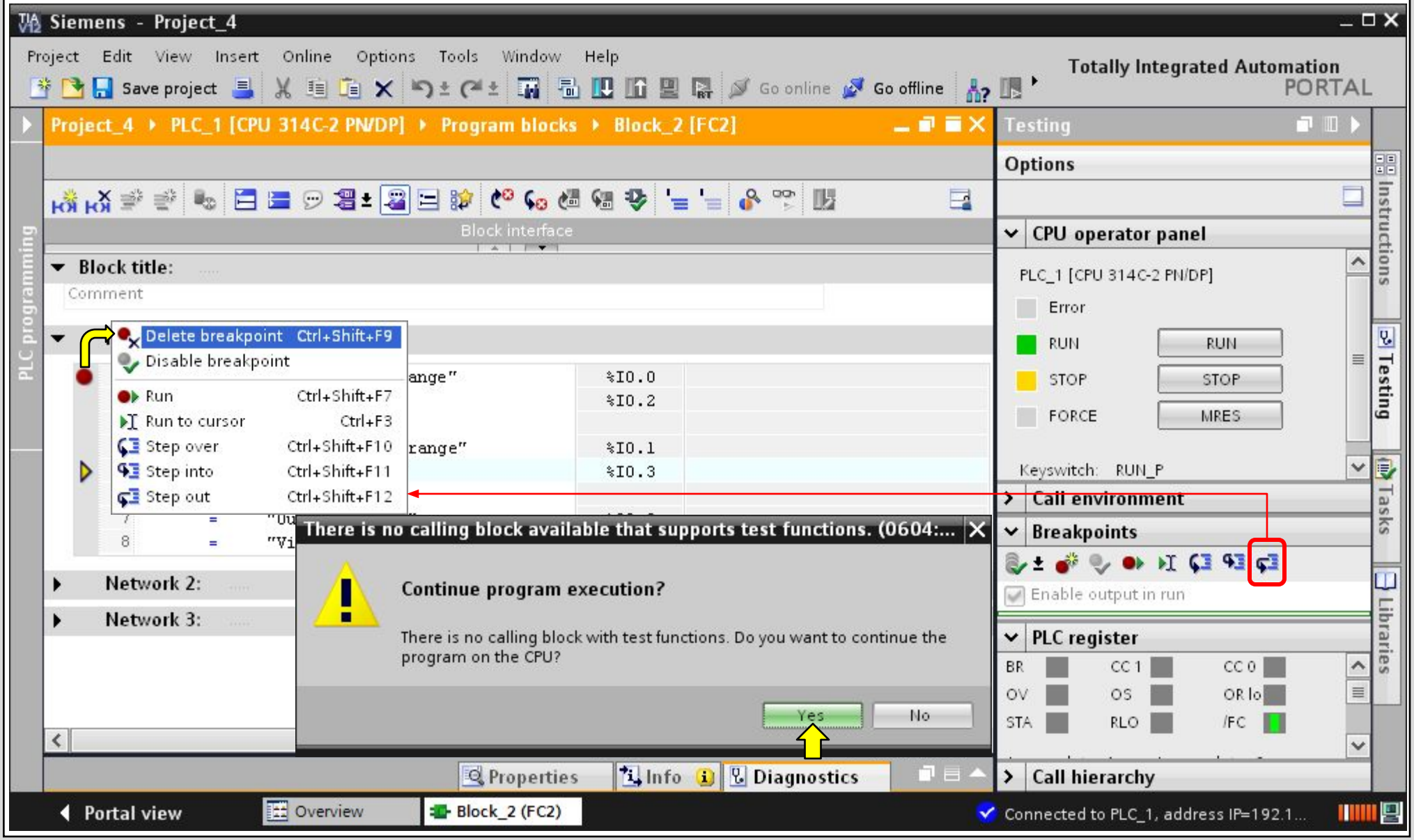
Пошаговый просмотр инструкций и флагов



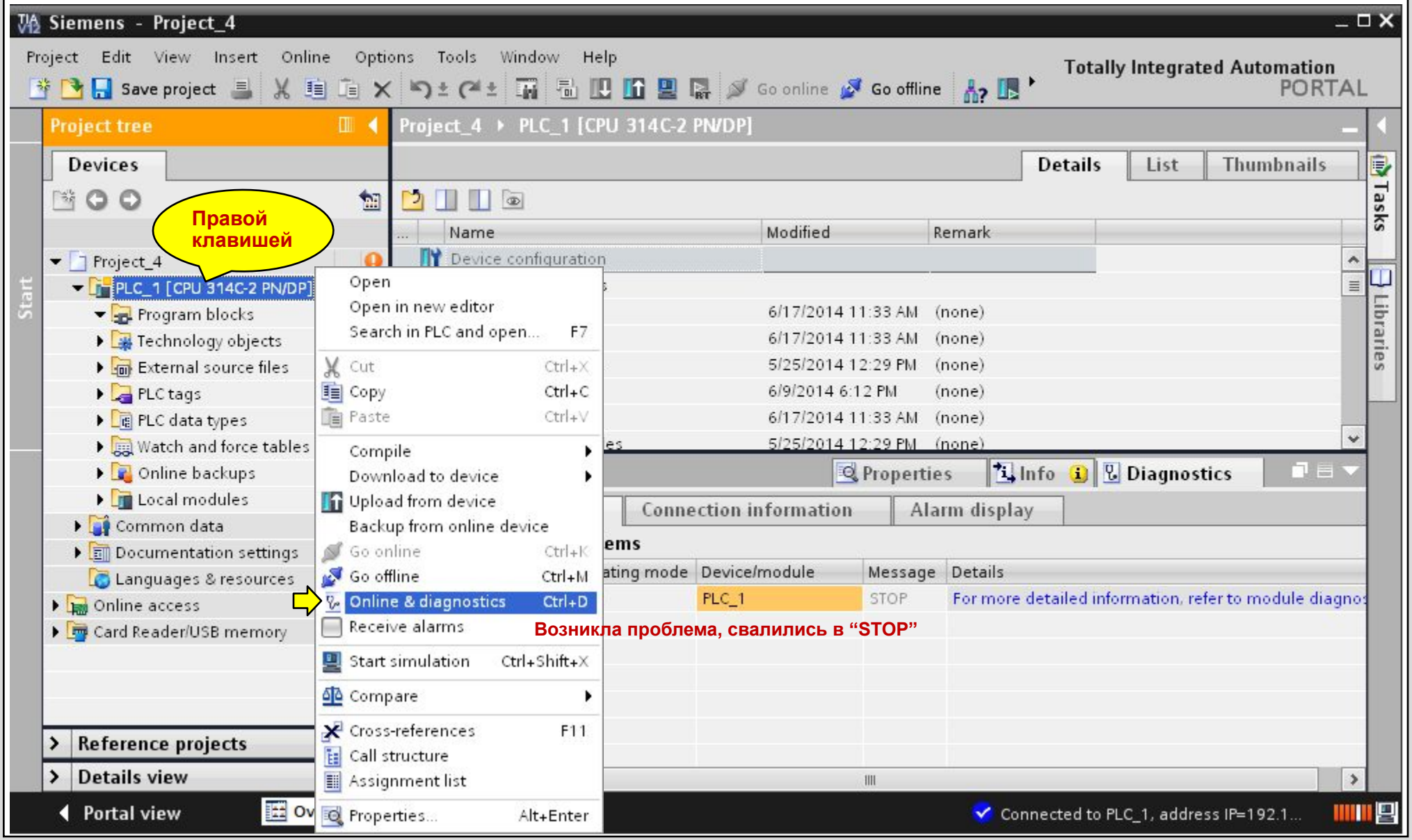
Пошаговый просмотр инструкций и флагов



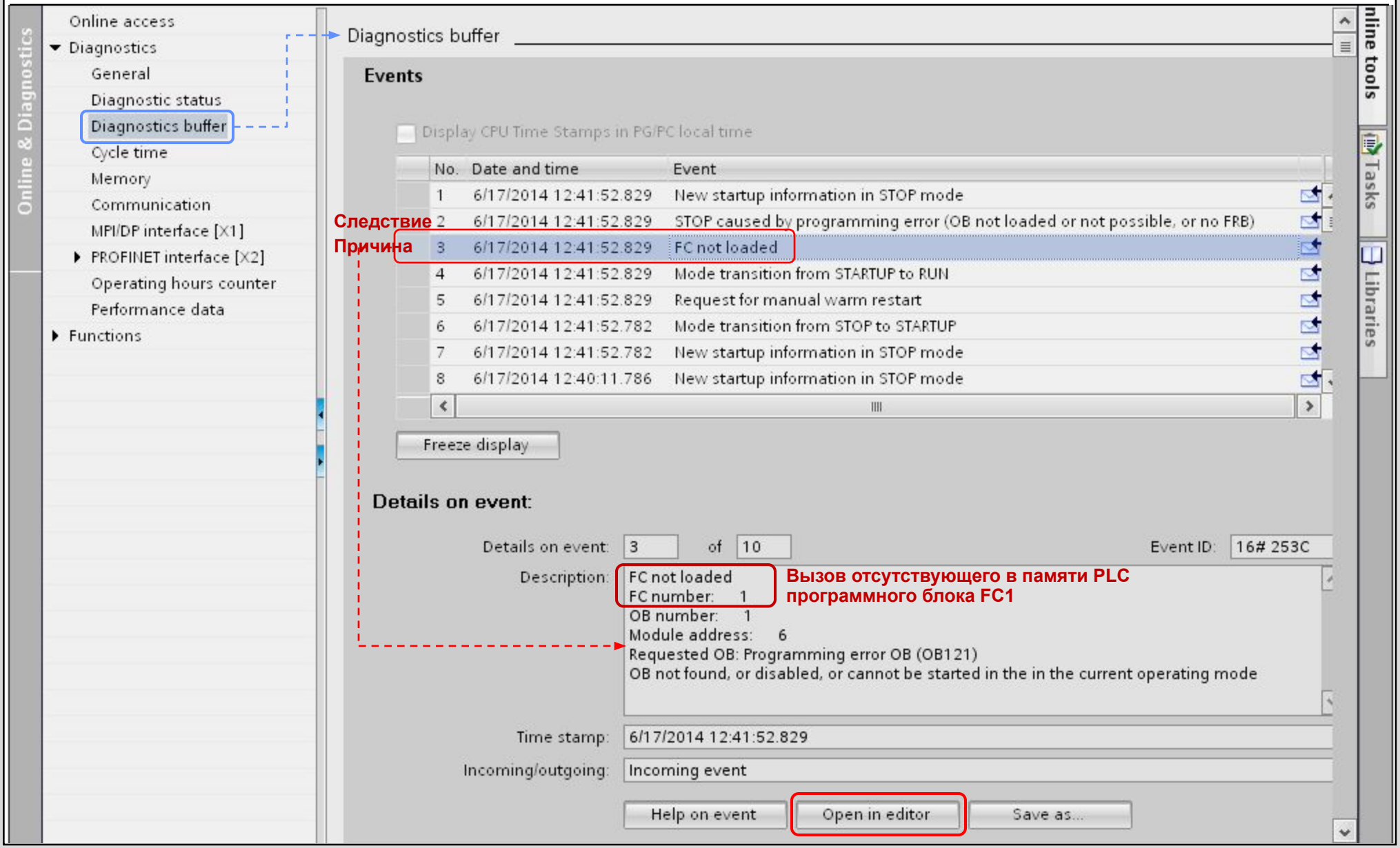
Выход из режима HOLD с возвратом в RUN



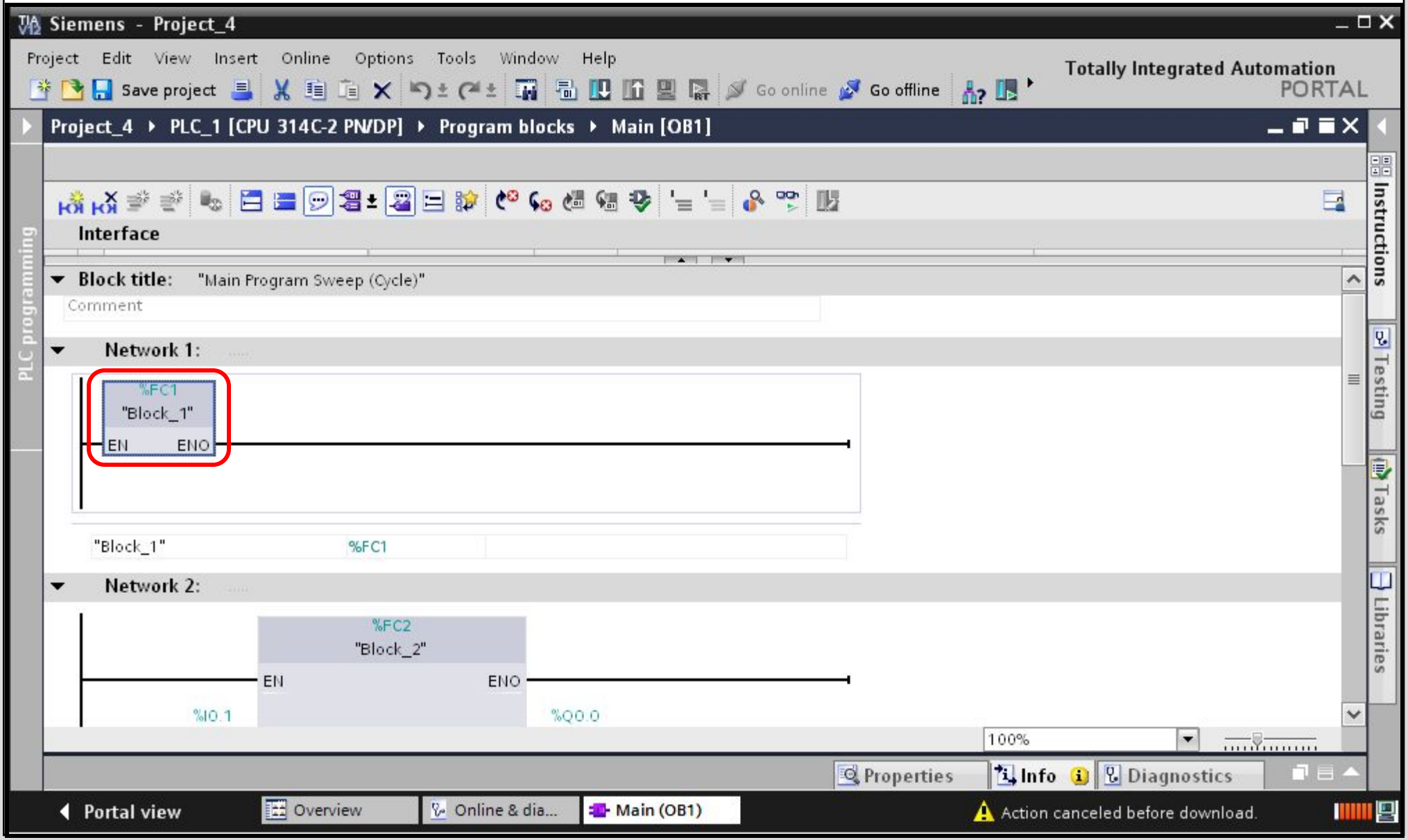
Диагностический буфер. Пример поиска и анализа неисправности



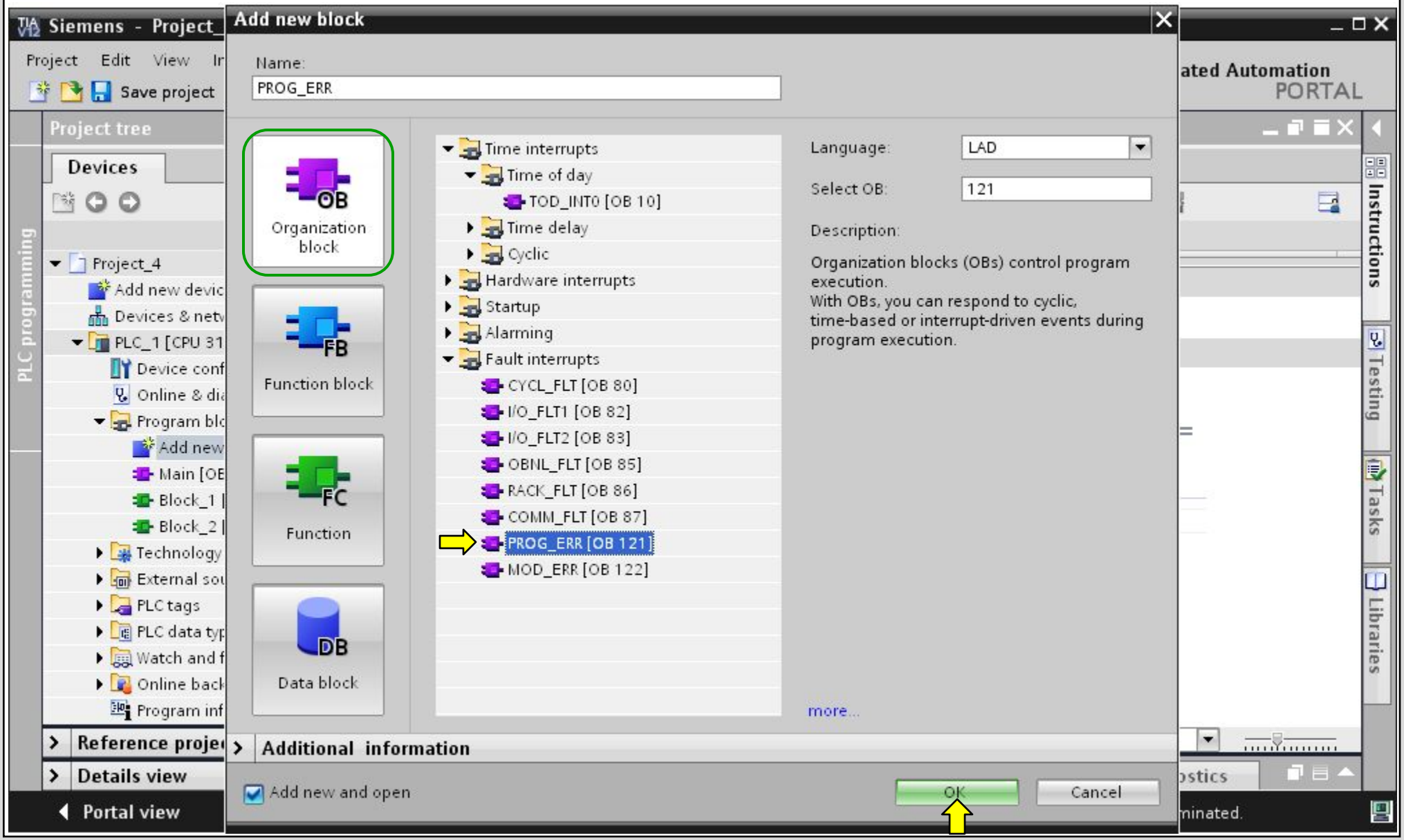
Диагностический буфер. Пример поиска и анализа неисправности



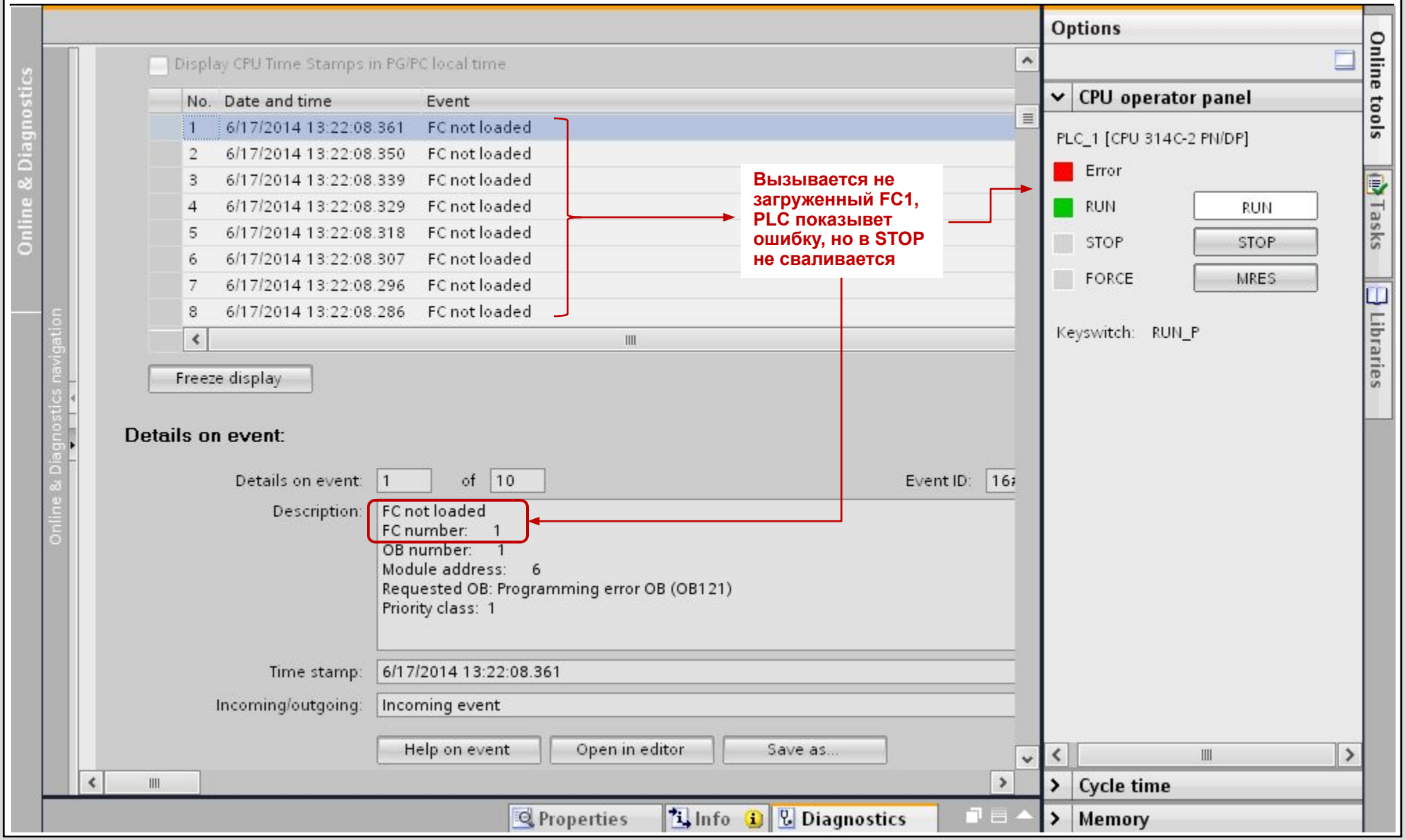
Точка вызова отсутствующего в памяти PLC блока FC1



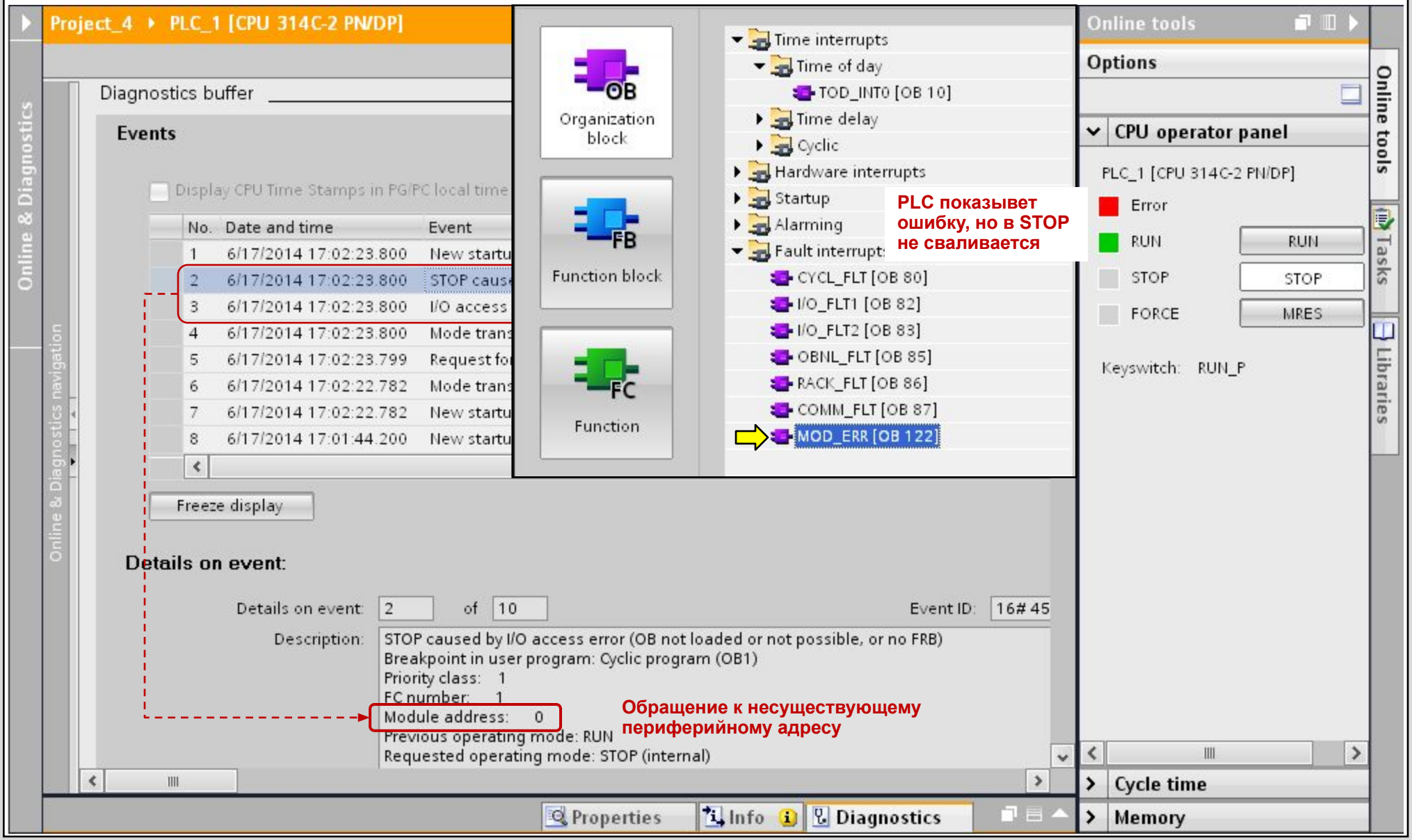
Блокировка перехода в STOP при помощи OB121



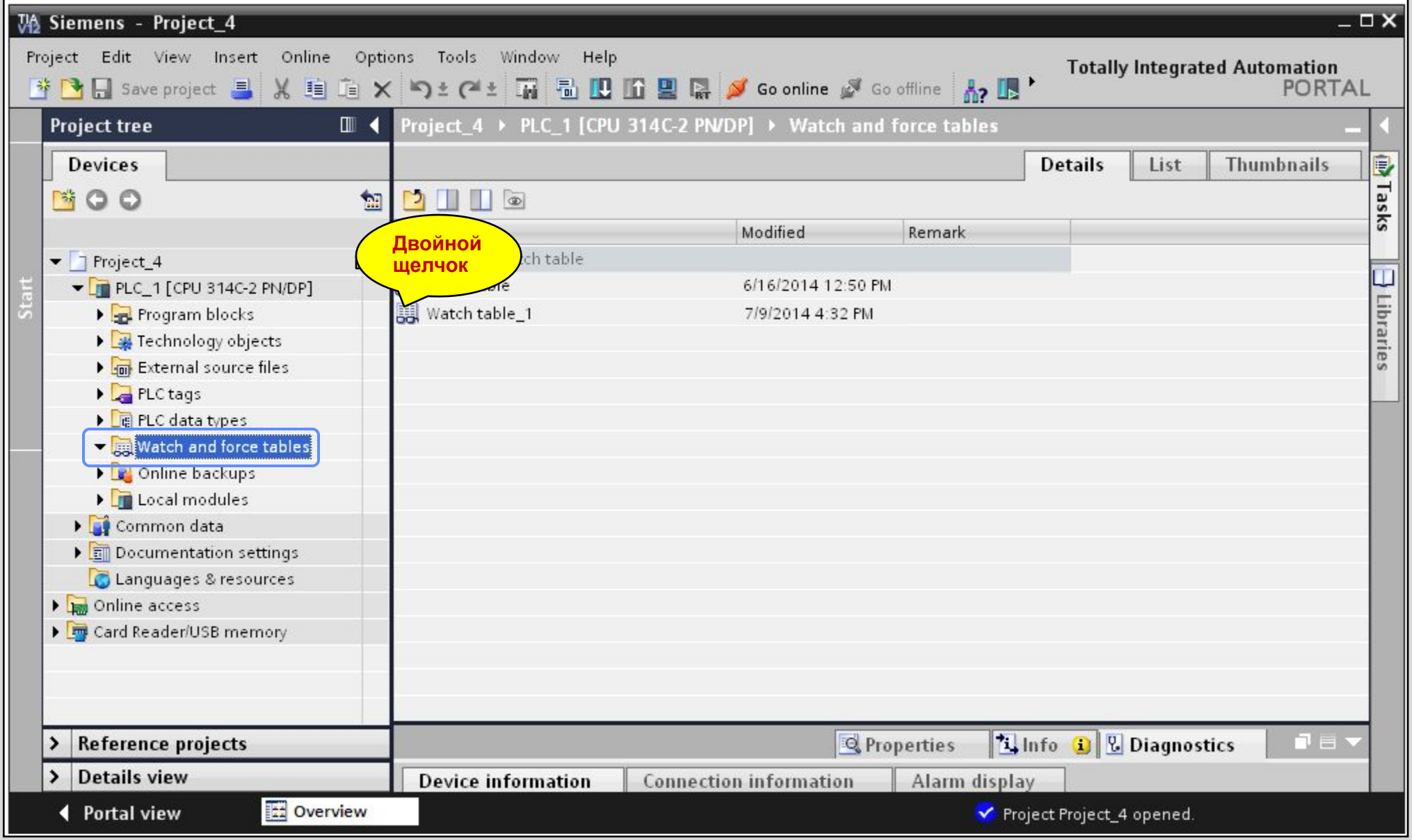
Блокировка перехода в STOP при помощи OB121



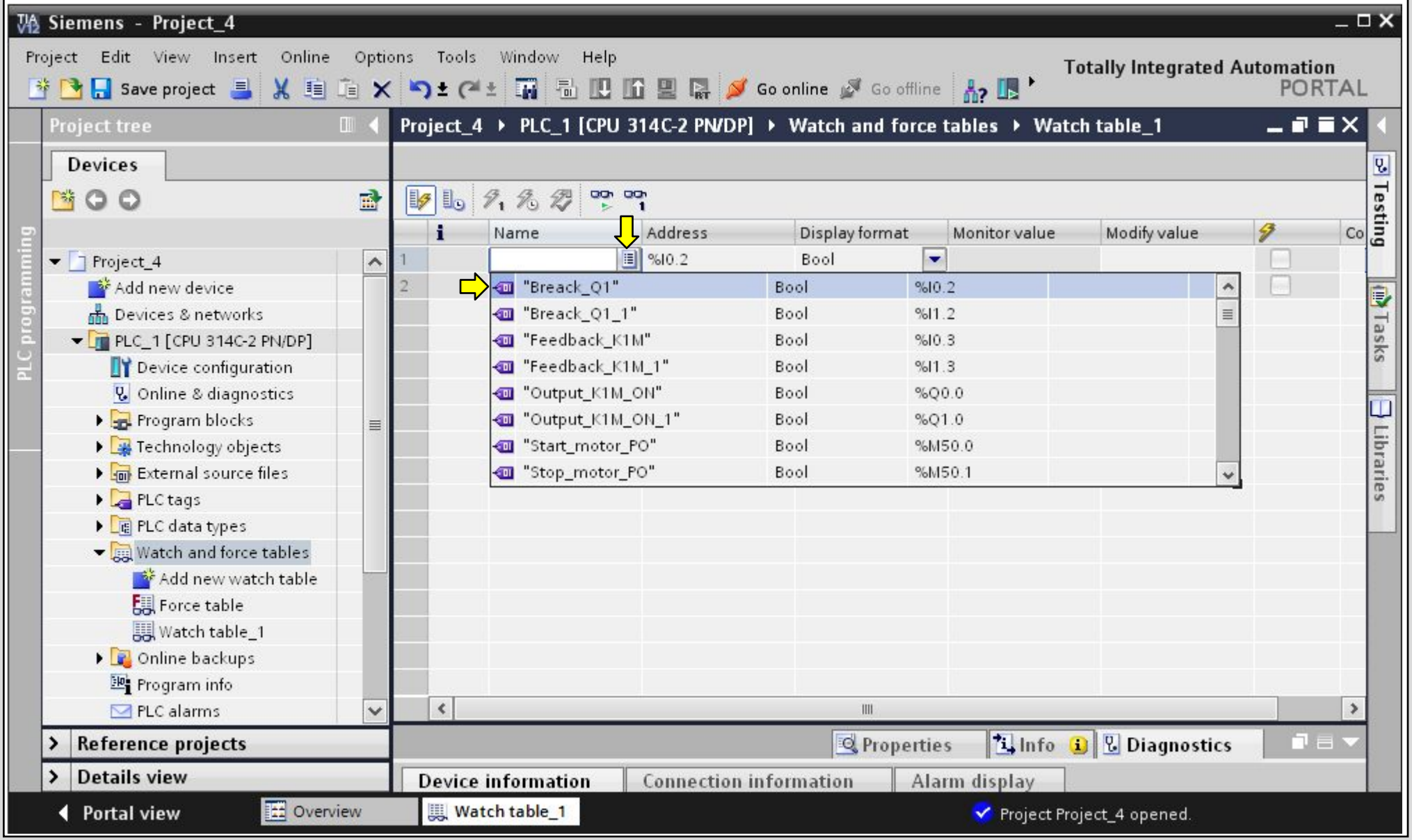
Блокировка перехода в STOP при помощи OB122



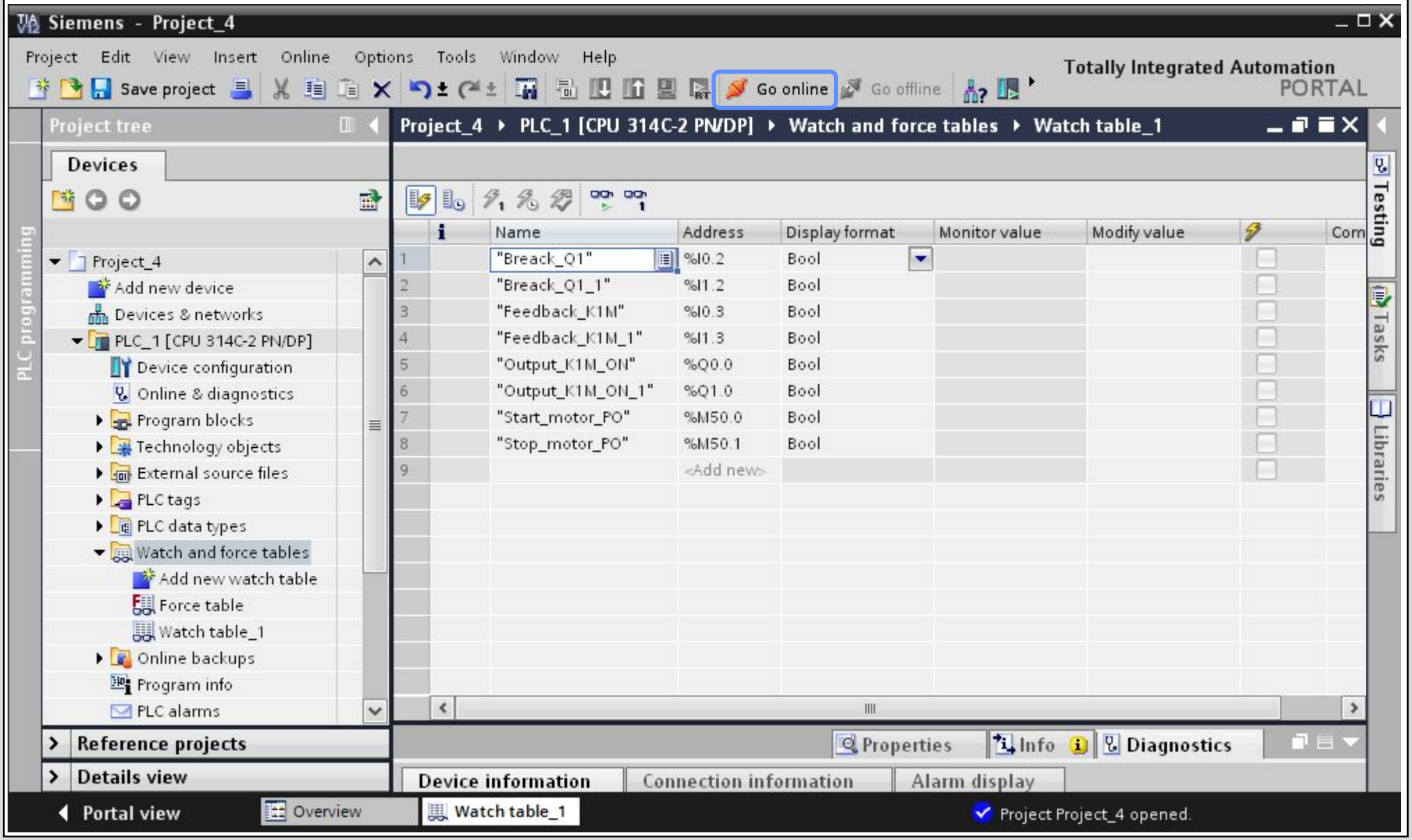
Работа с таблицей переменных Watch table



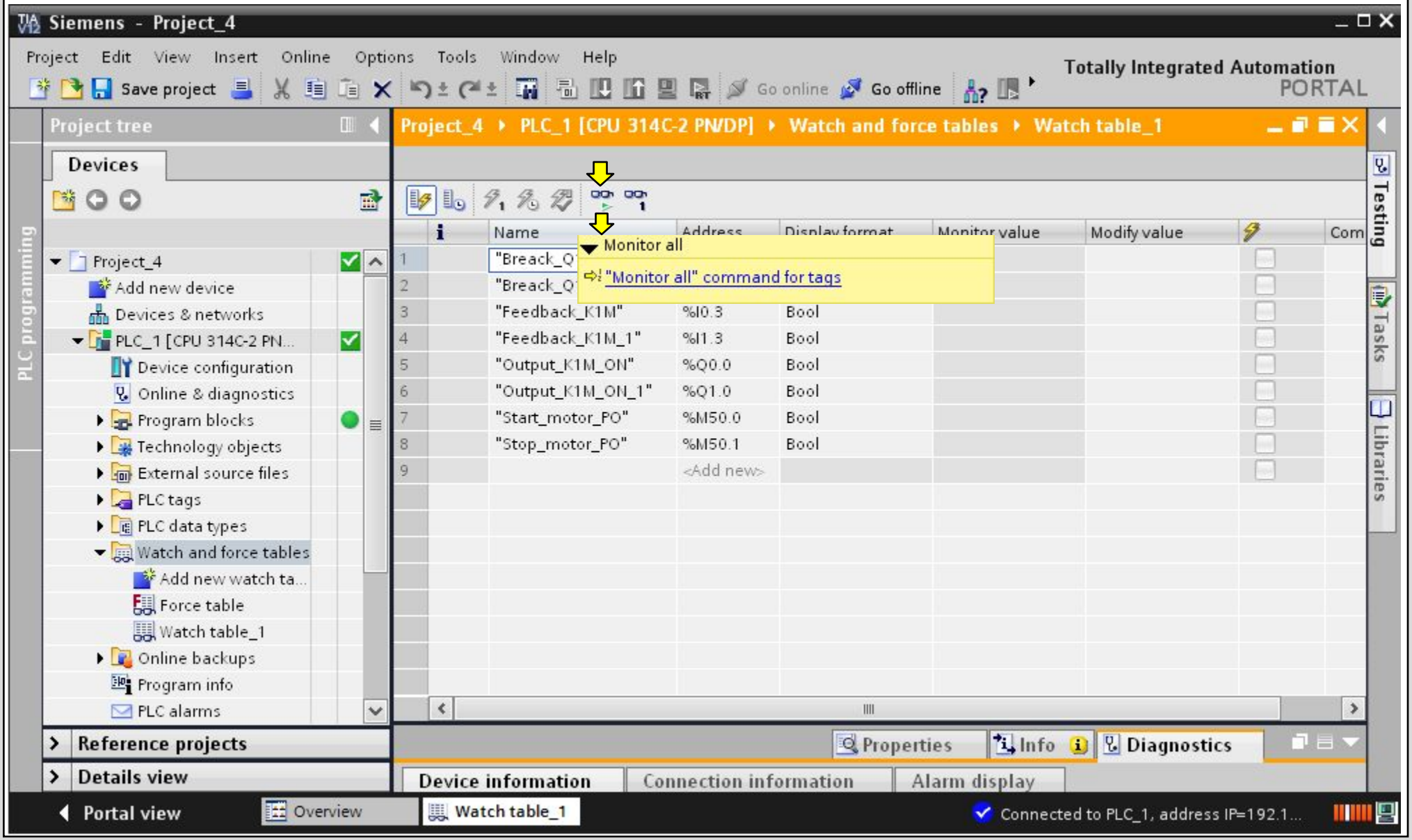
Работа с таблицей переменных Watch table



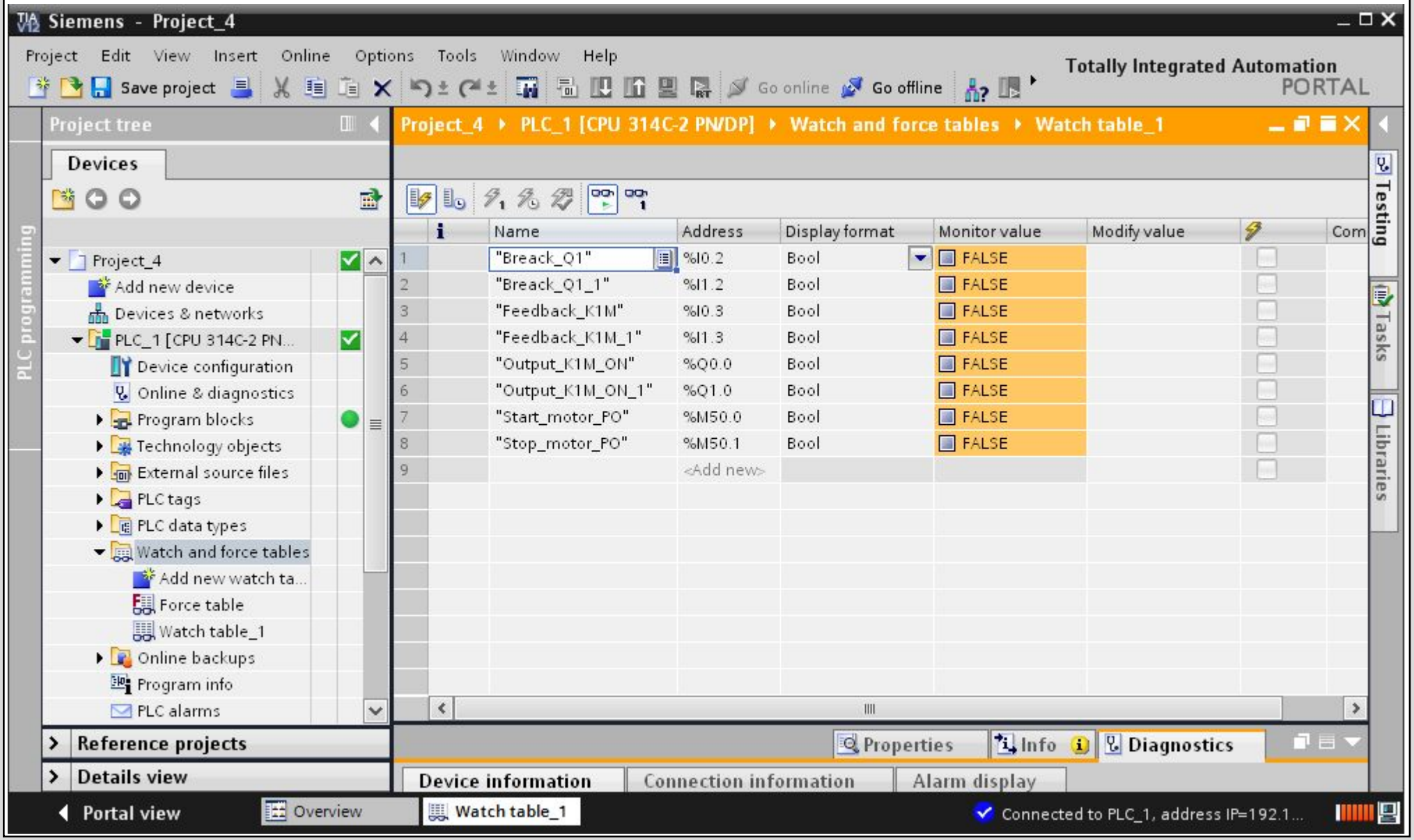
Работа с таблицей переменных Watch table



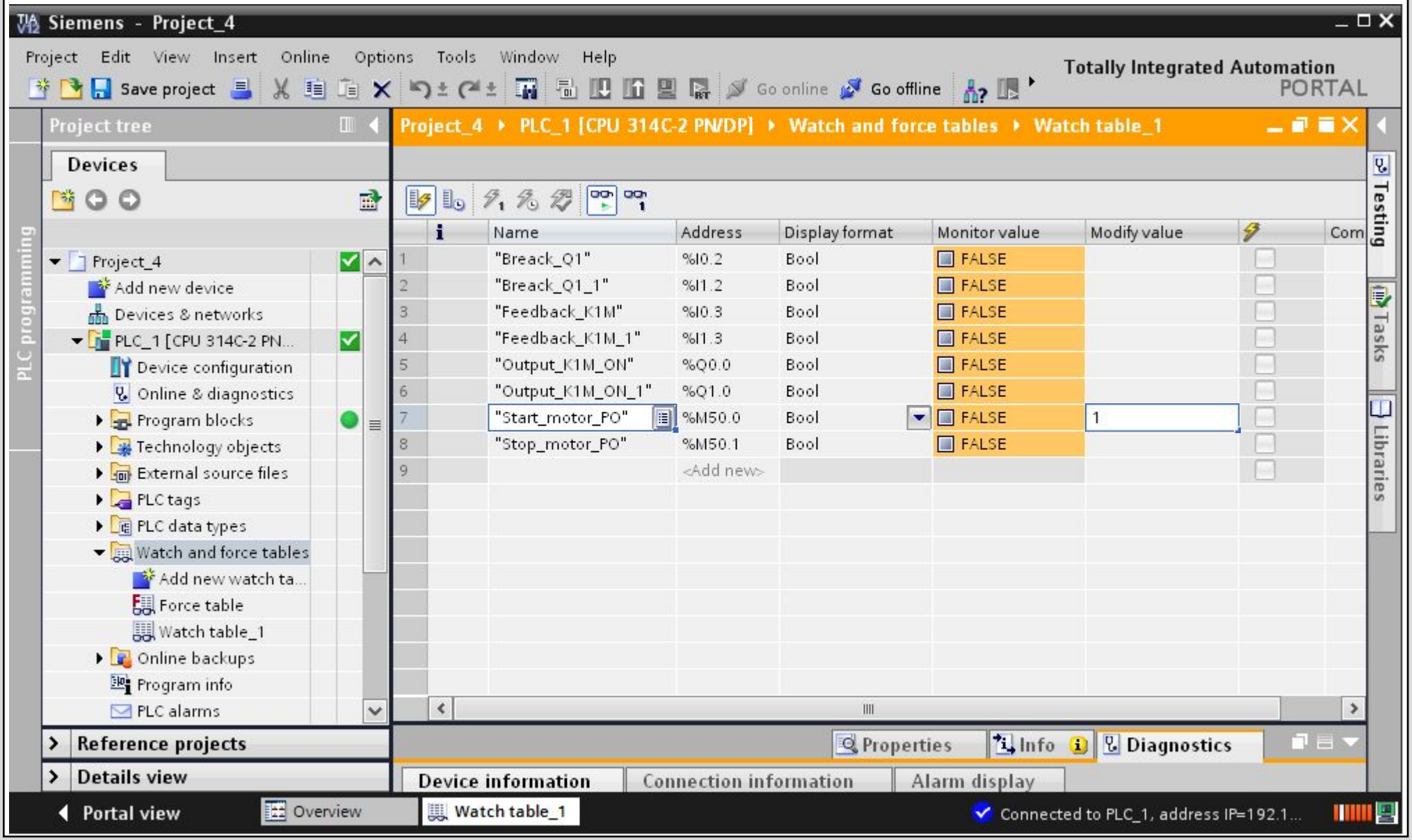
Работа с таблицей переменных Watch table OnLine



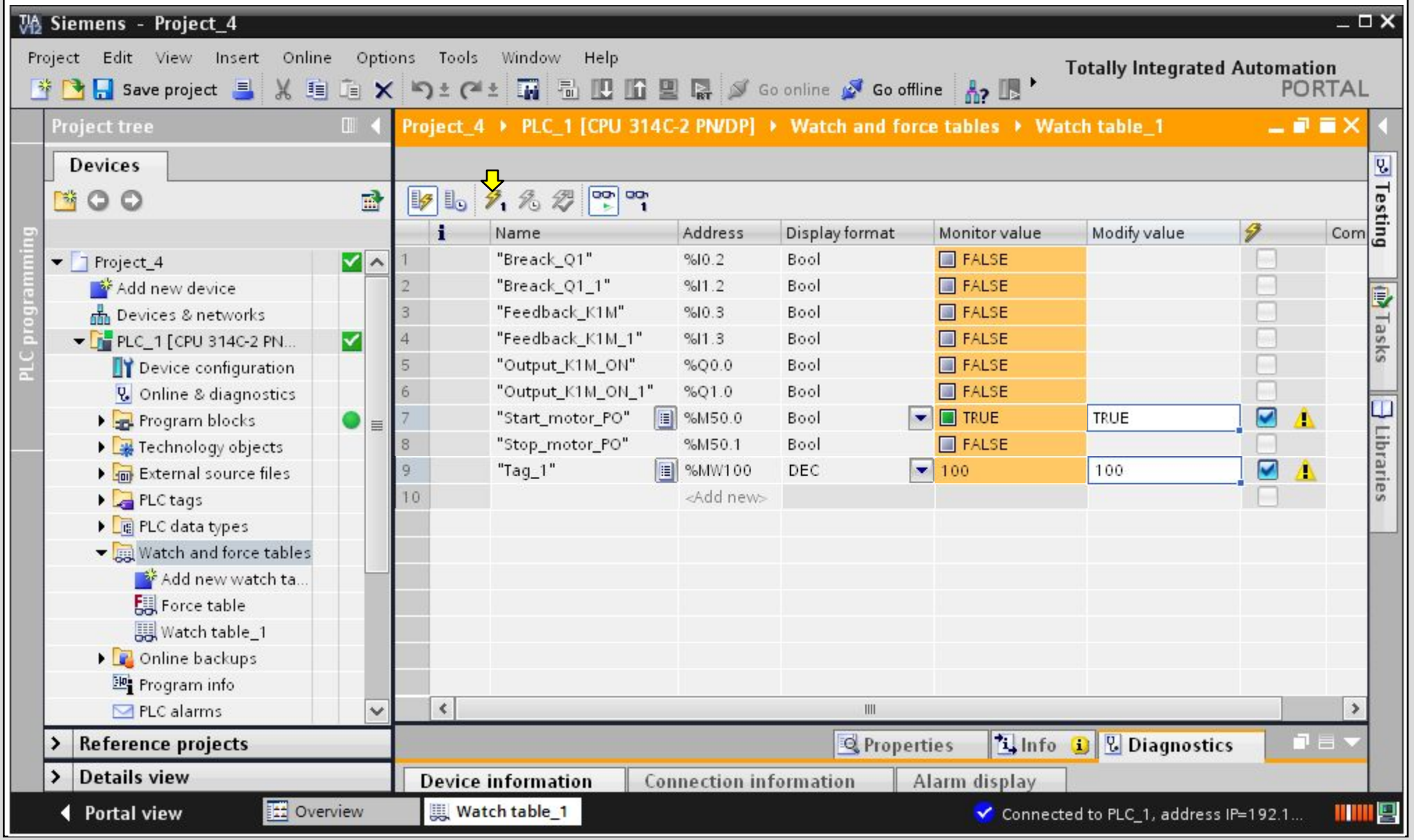
Работа с таблицей переменных Watch table OnLine



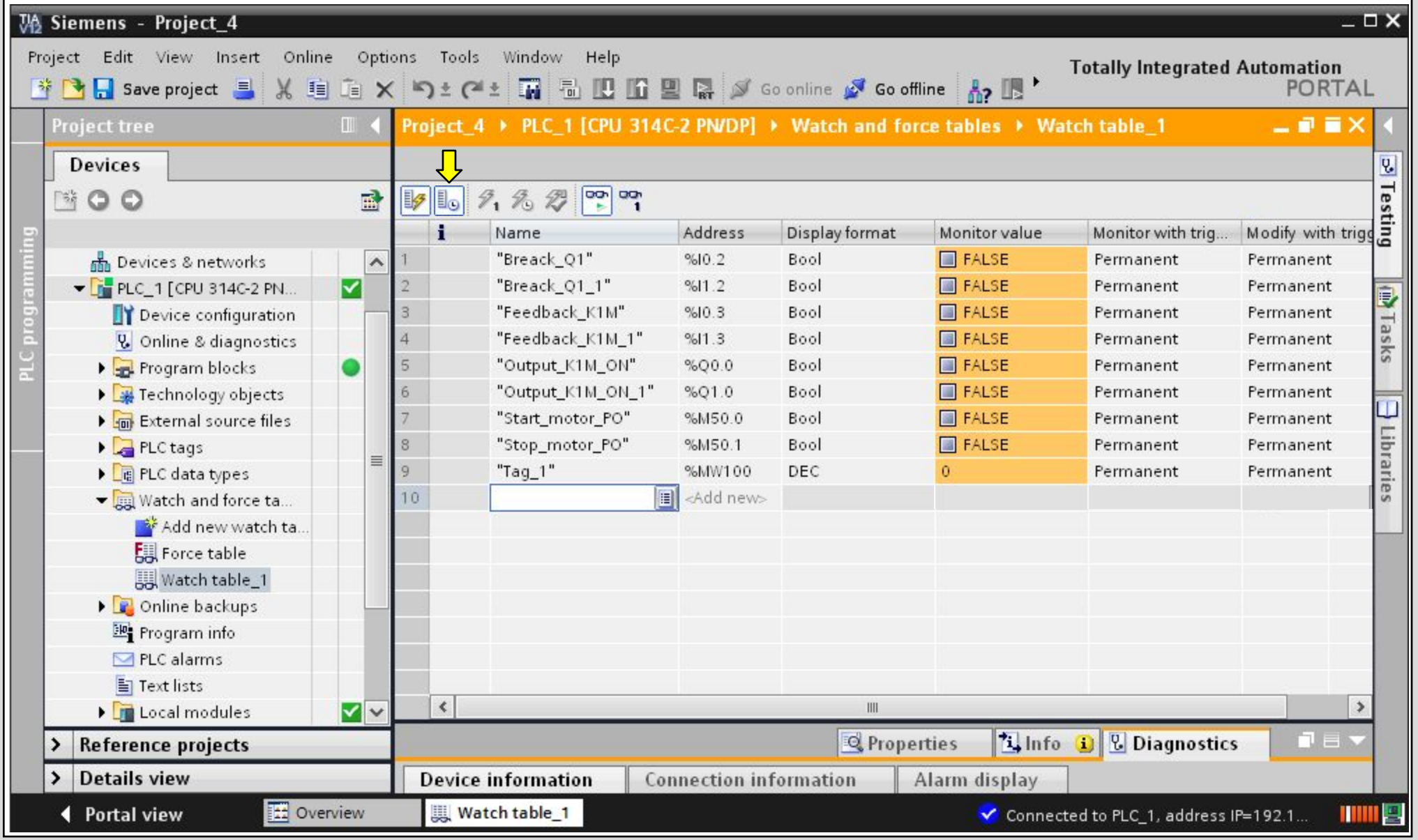
Работа с таблицей переменных Watch table OnLine



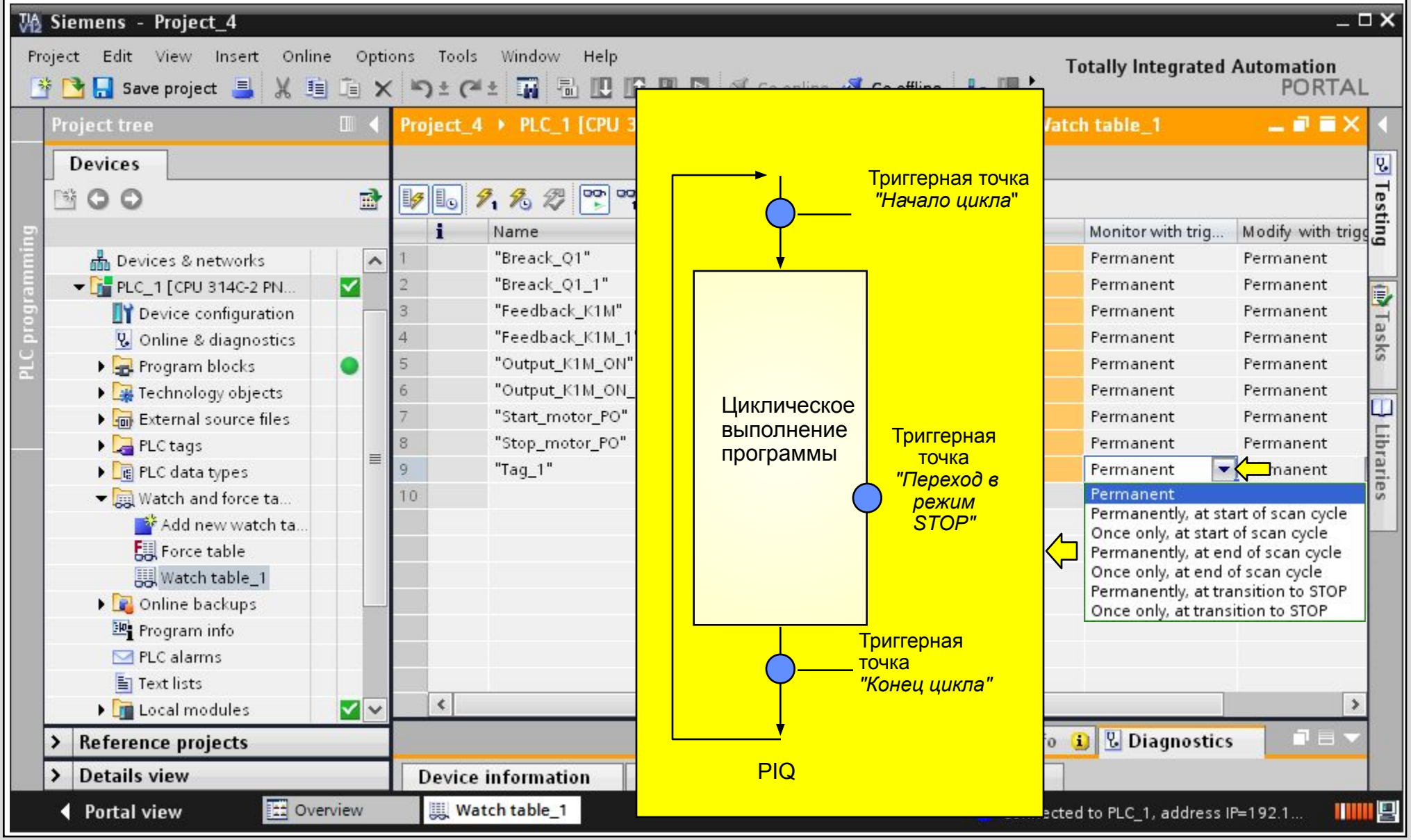
Работа с таблицей переменных Watch table OnLine



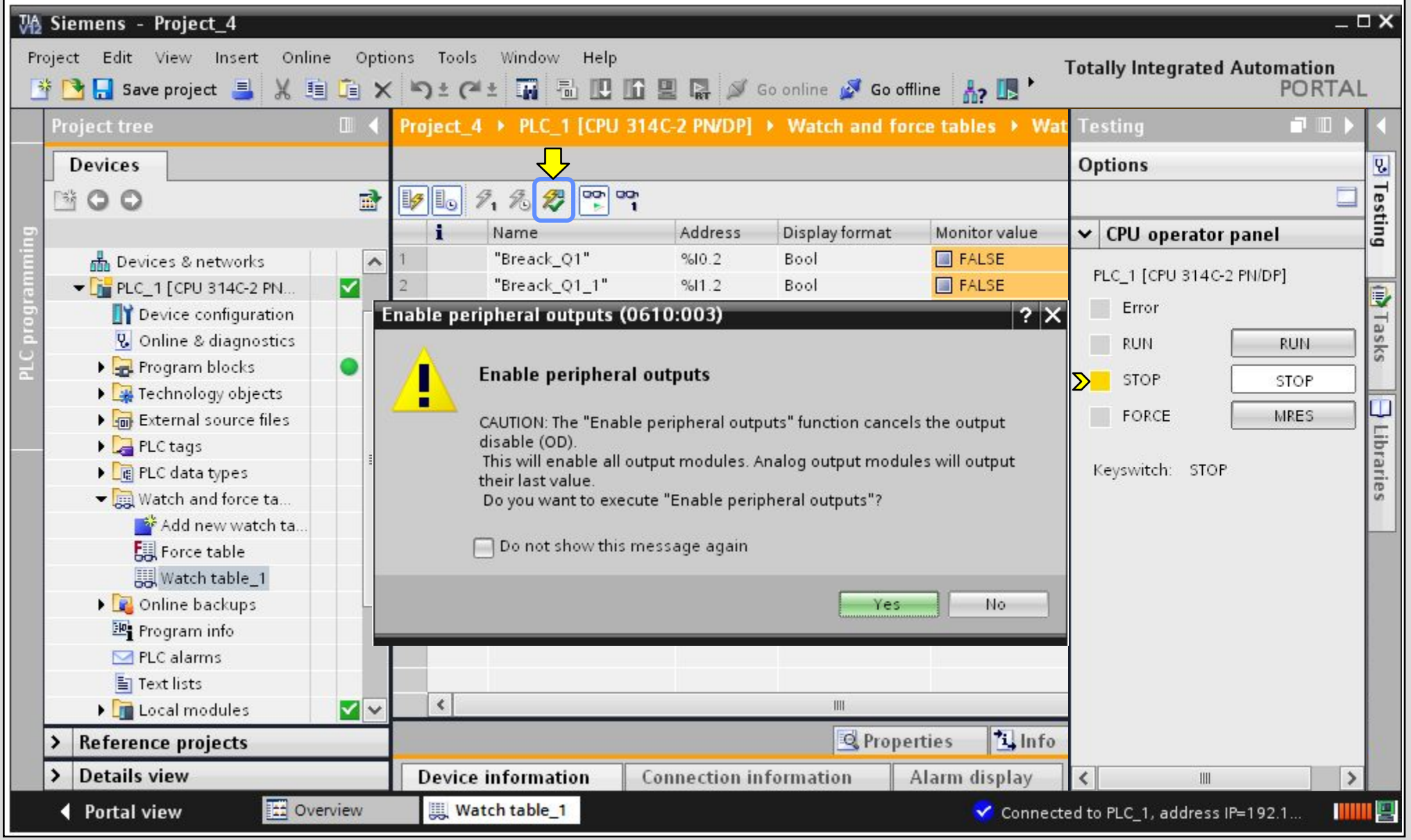
Работа с таблицей переменных Watch table OnLine



Выбор точки просмотра в таблице переменных Watch table



Принудительная установка выходов в режиме STOP



Принудительная установка выходов в режиме STOP

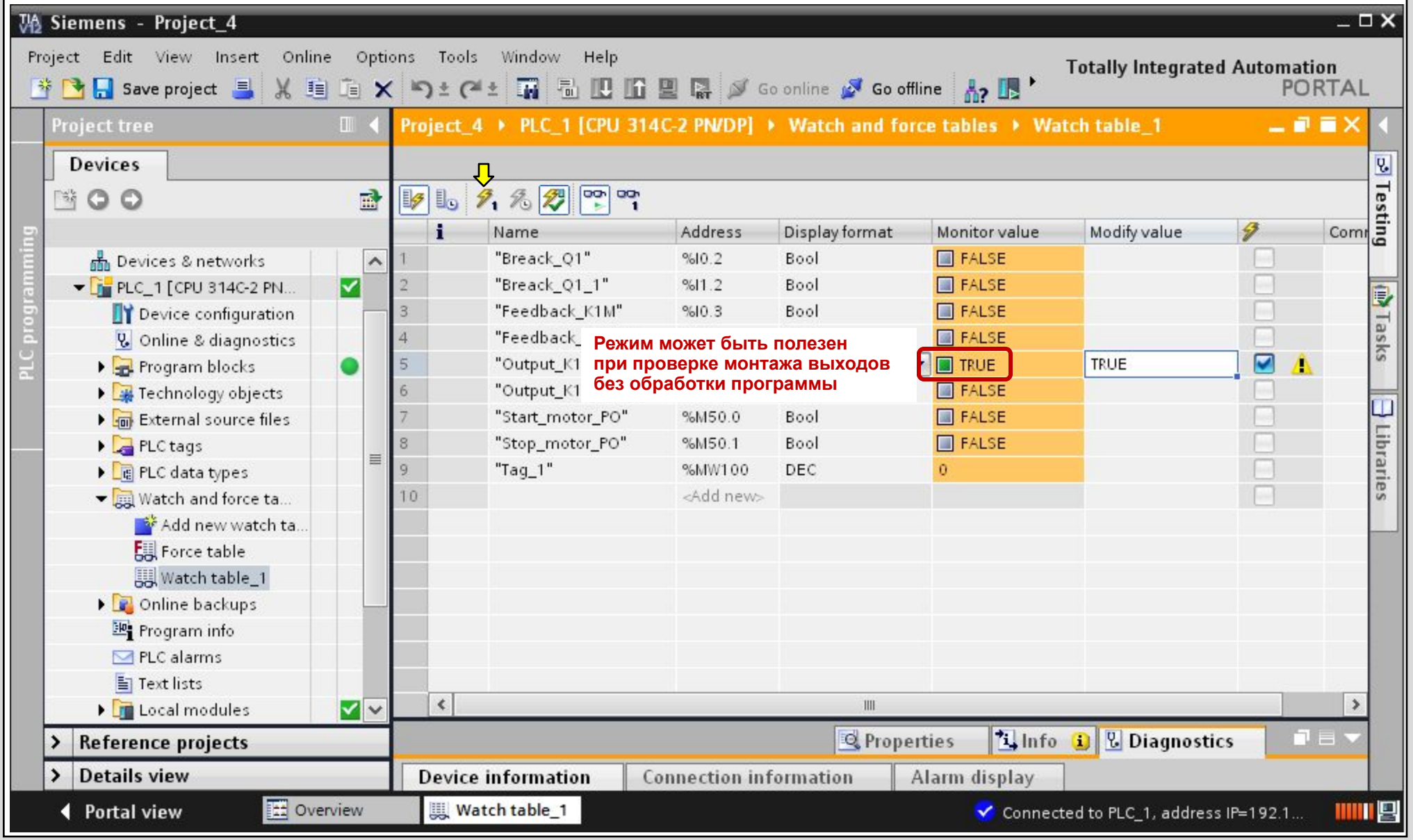


Таблица принудительной установки входов/выходов Force

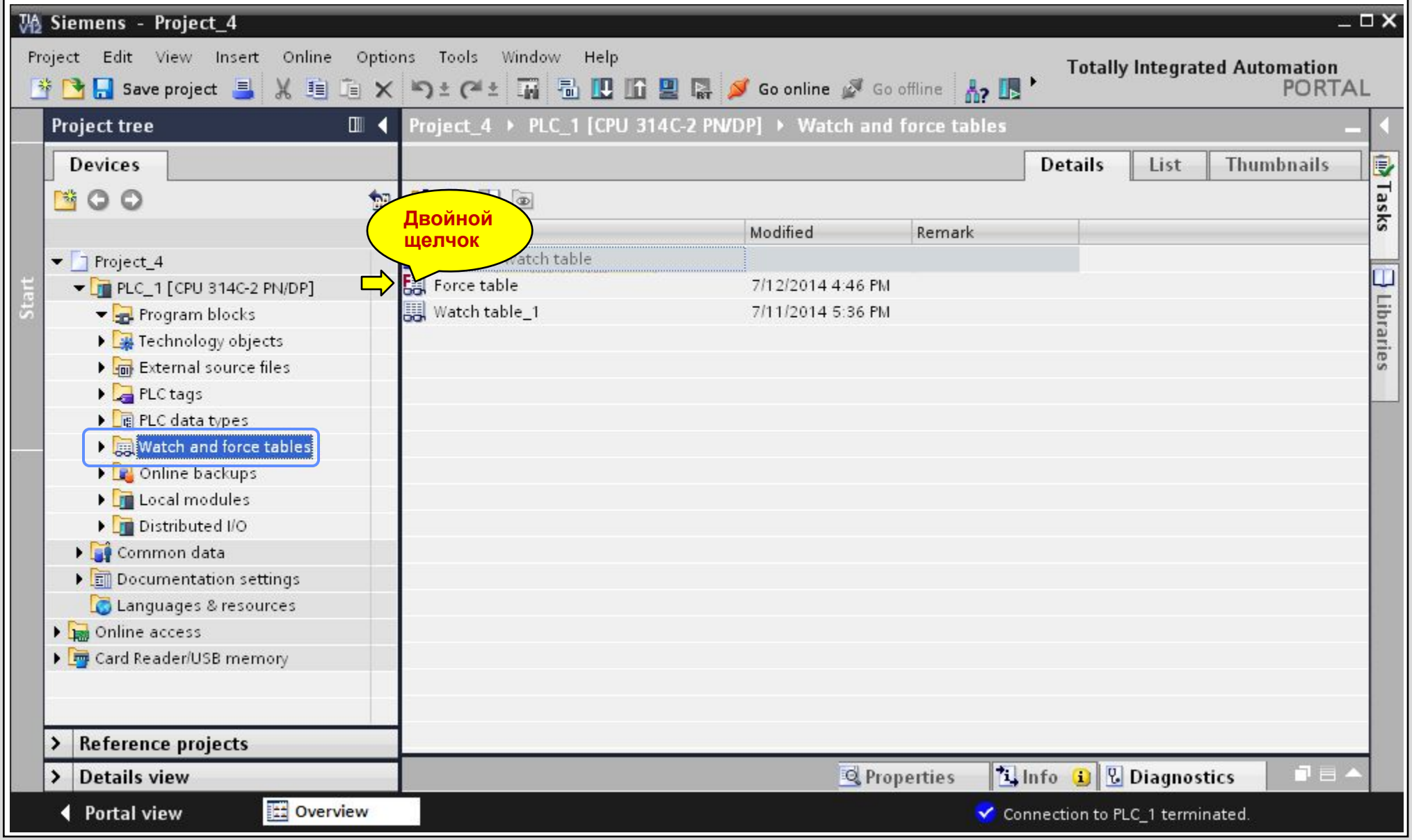


Таблица принудительной установки входов/выходов Force

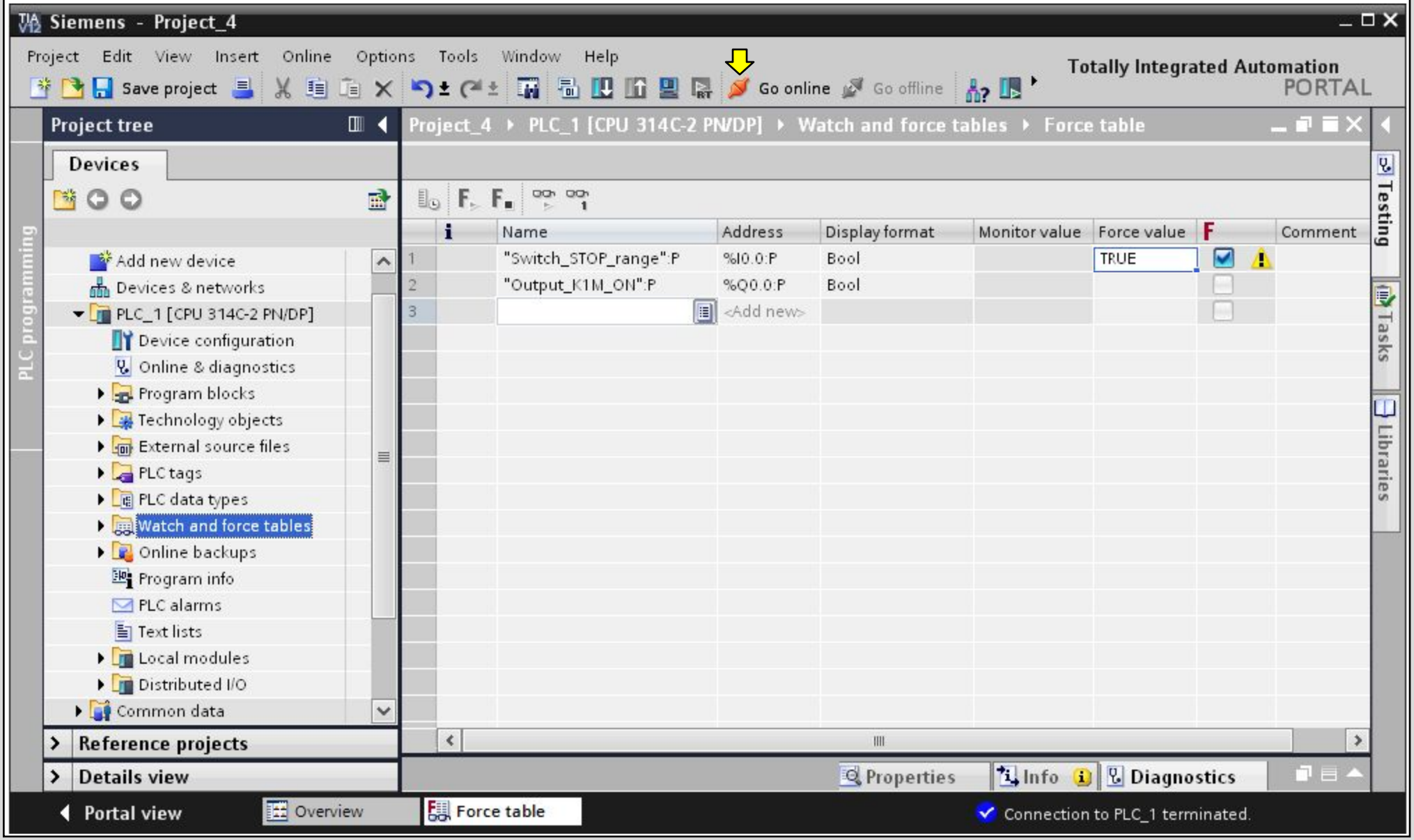


Таблица принудительной установки входов/выходов Force

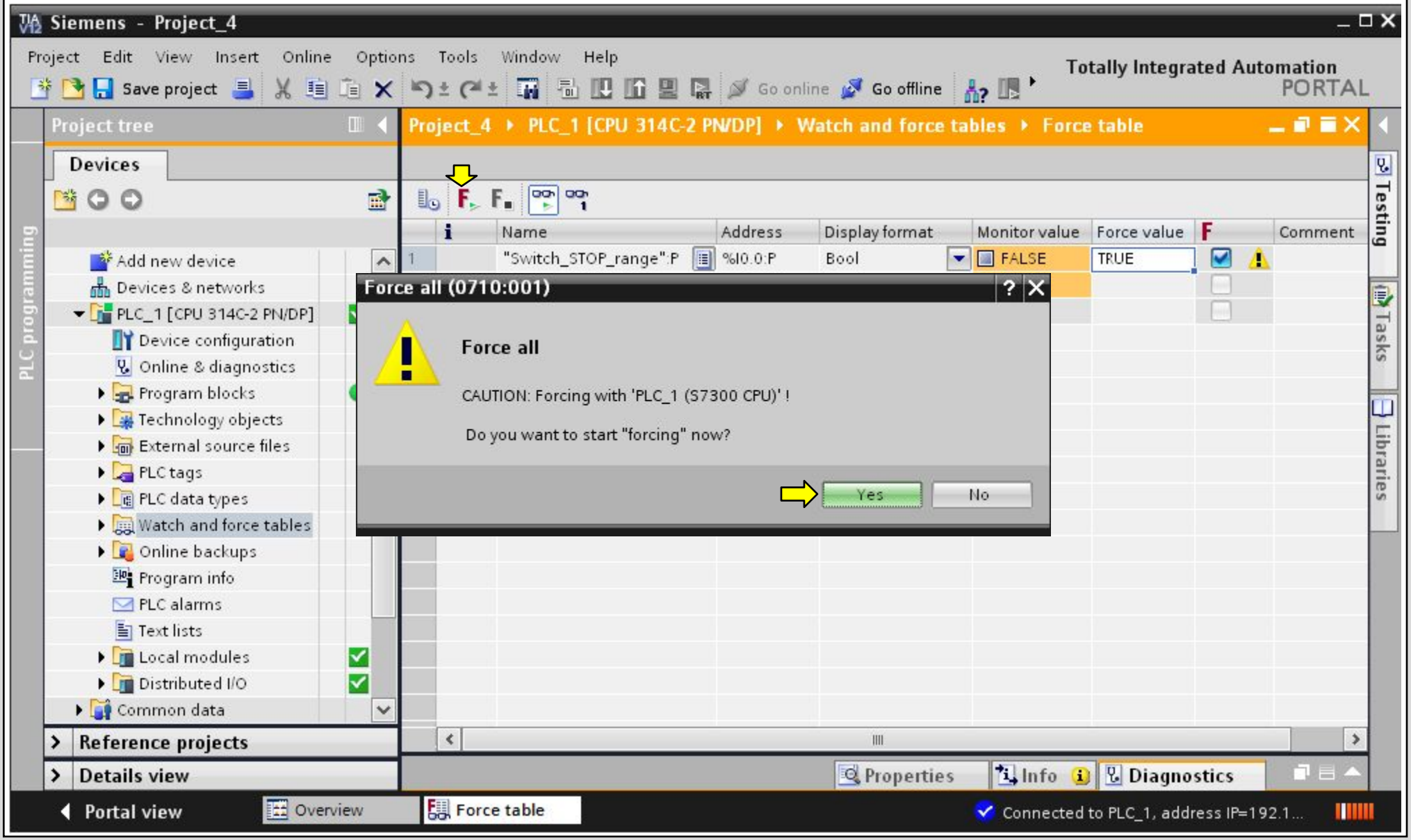
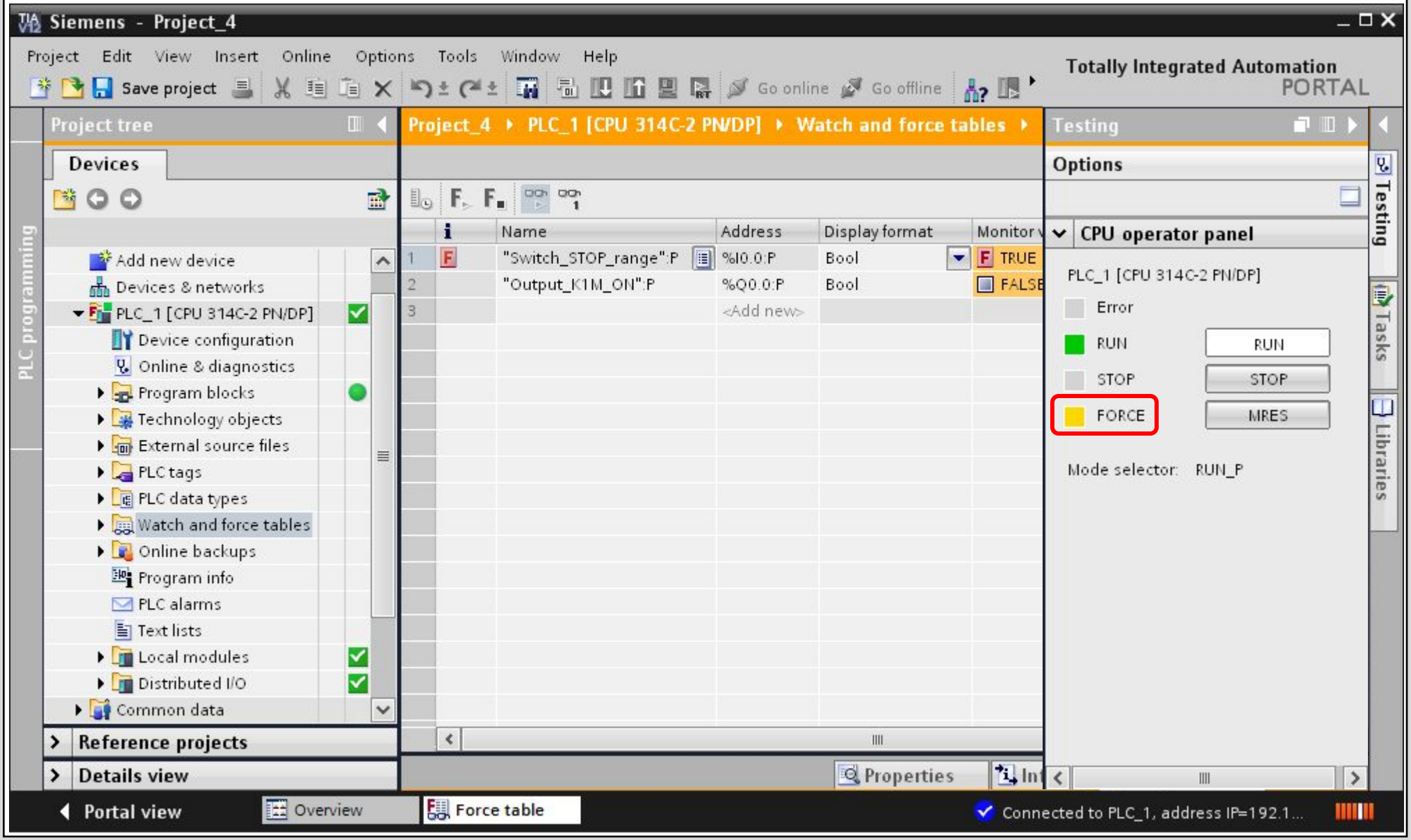
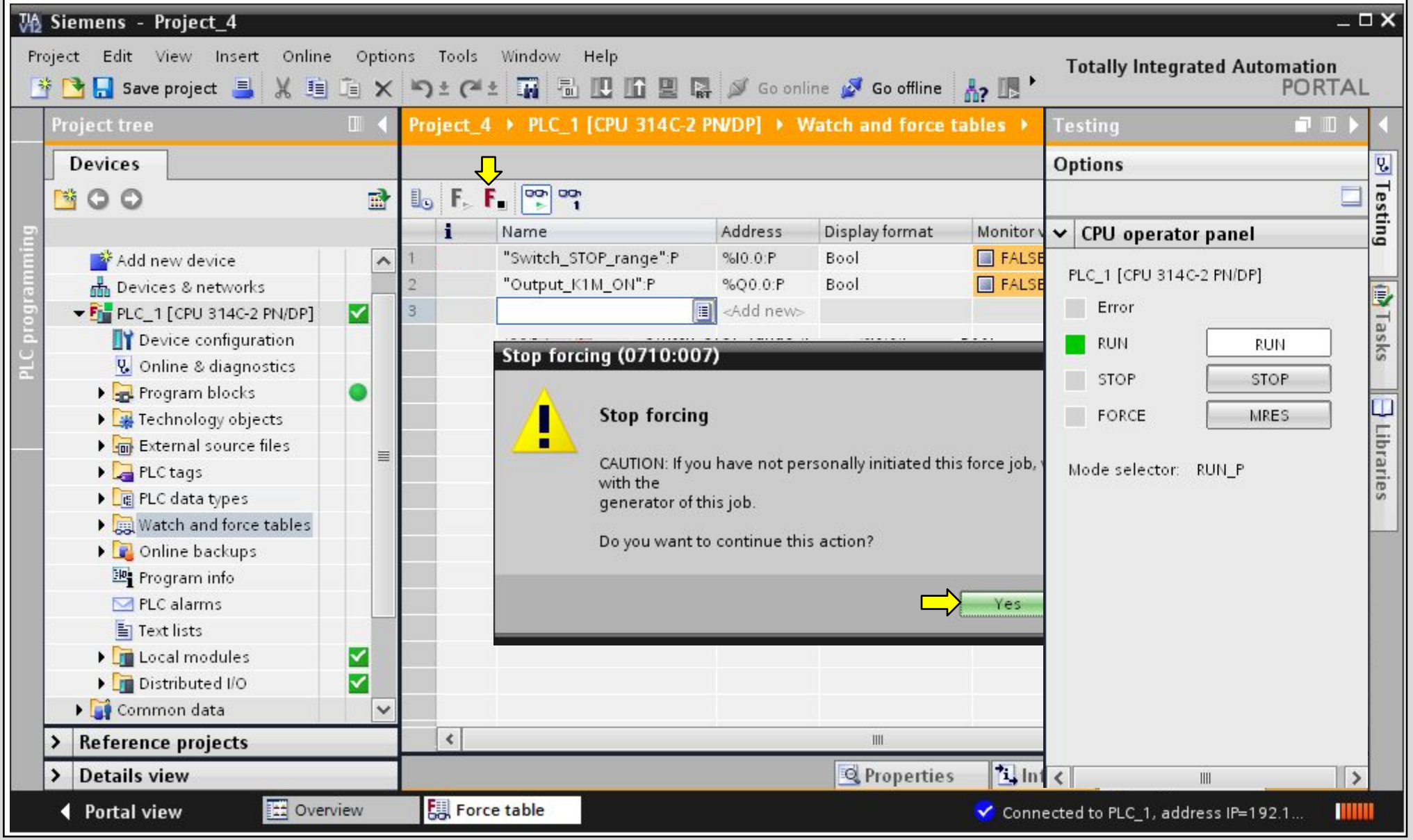


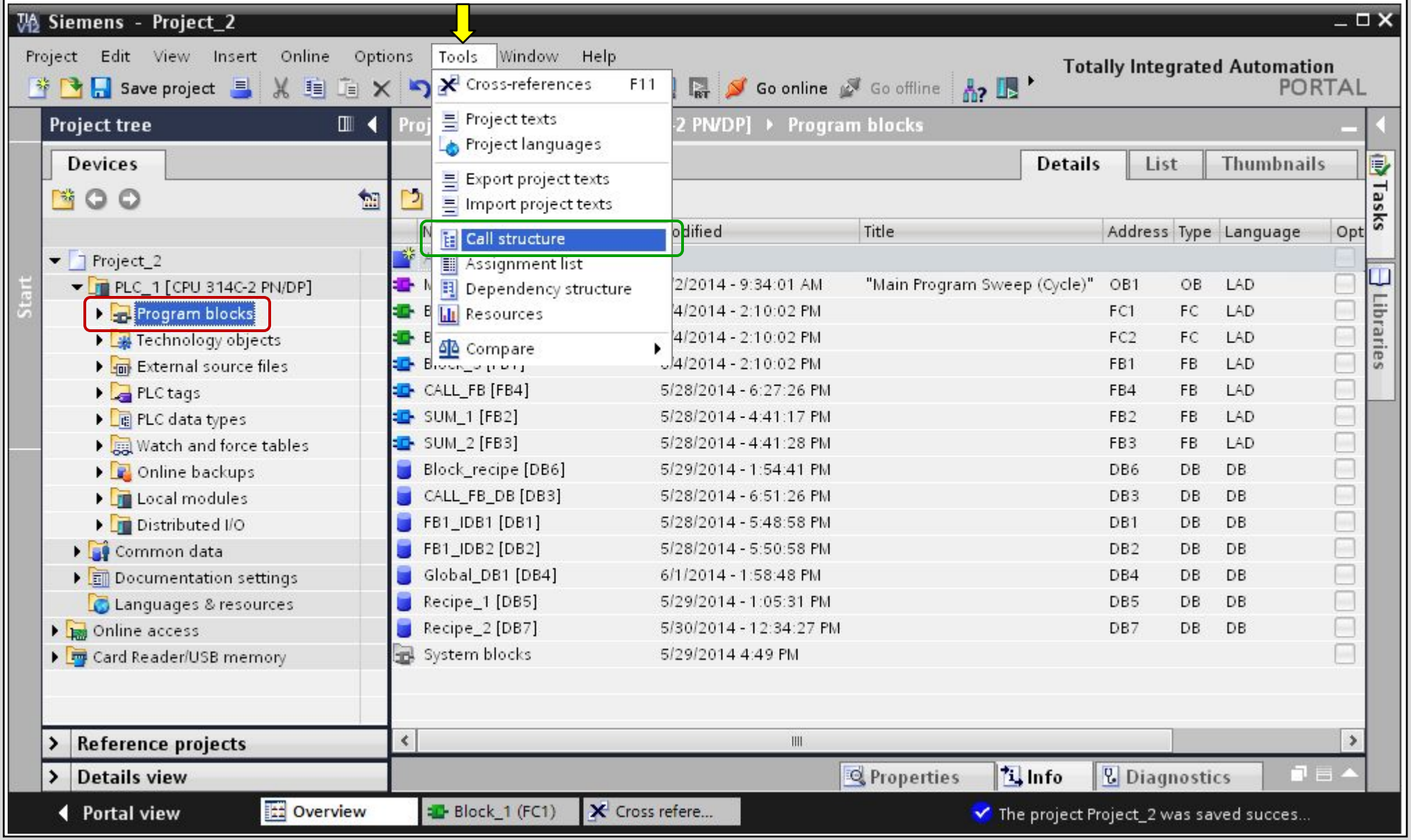
Таблица принудительной установки входов/выходов Force



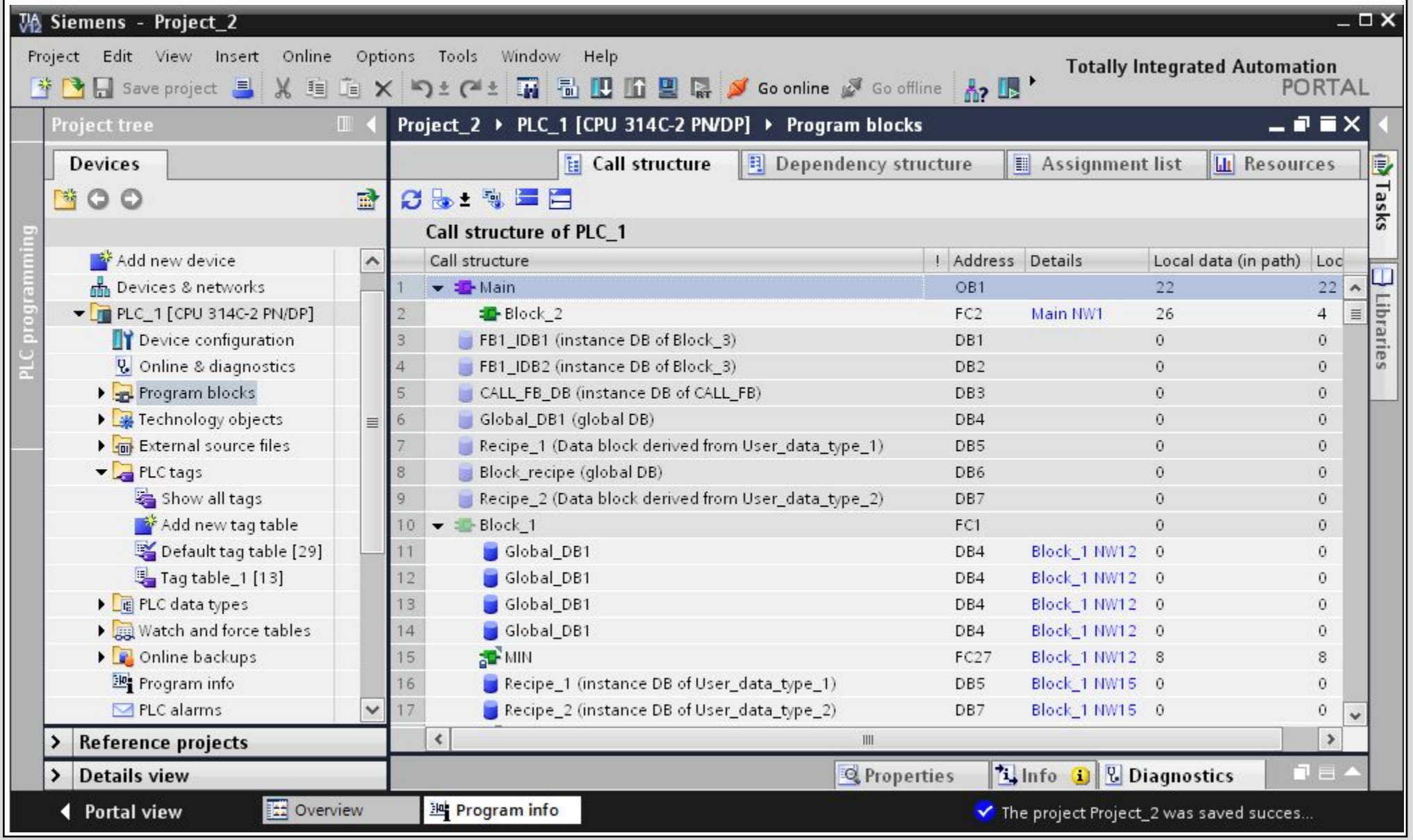
Выход из режима FORCE



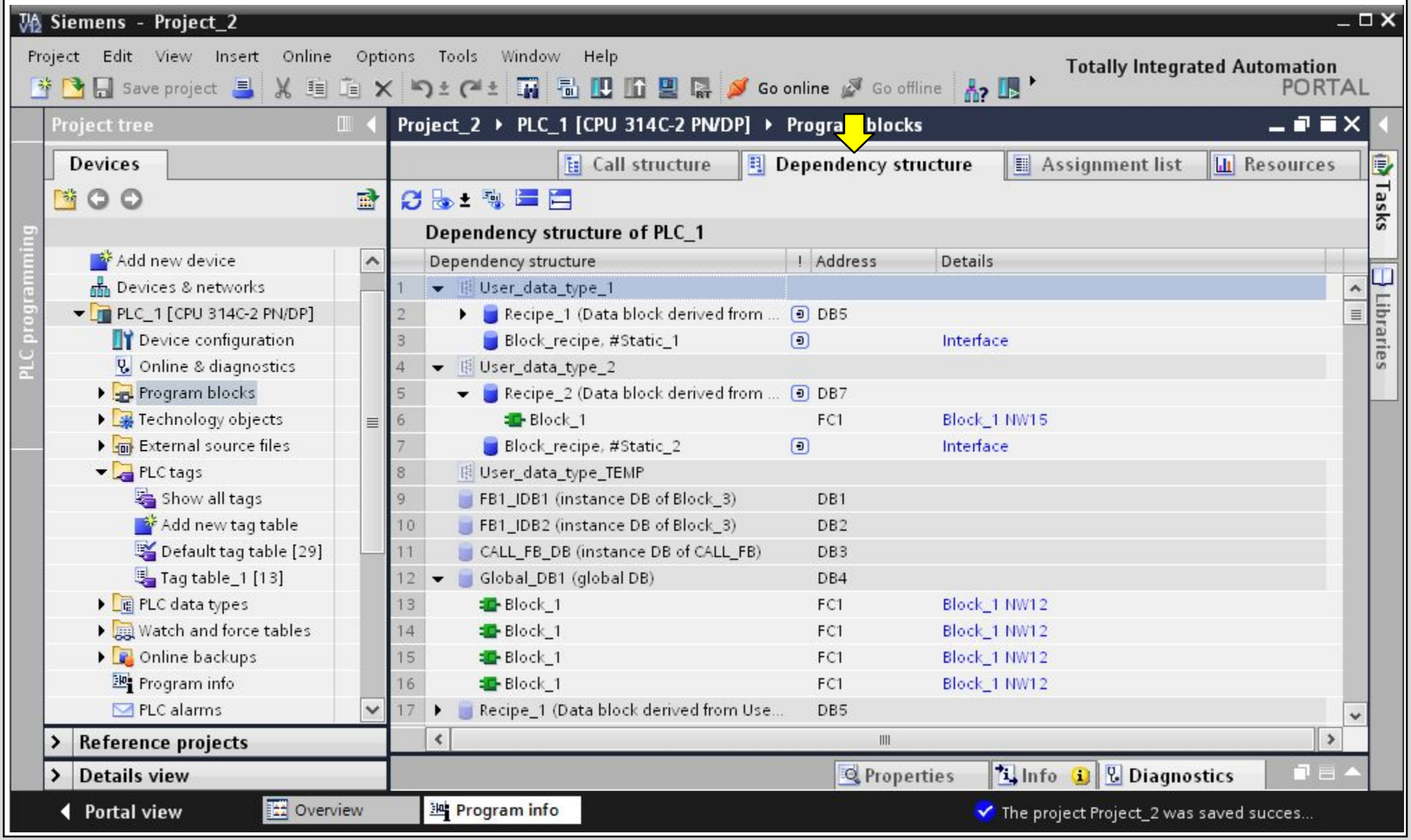
Структура вызовов программных блоков и функций



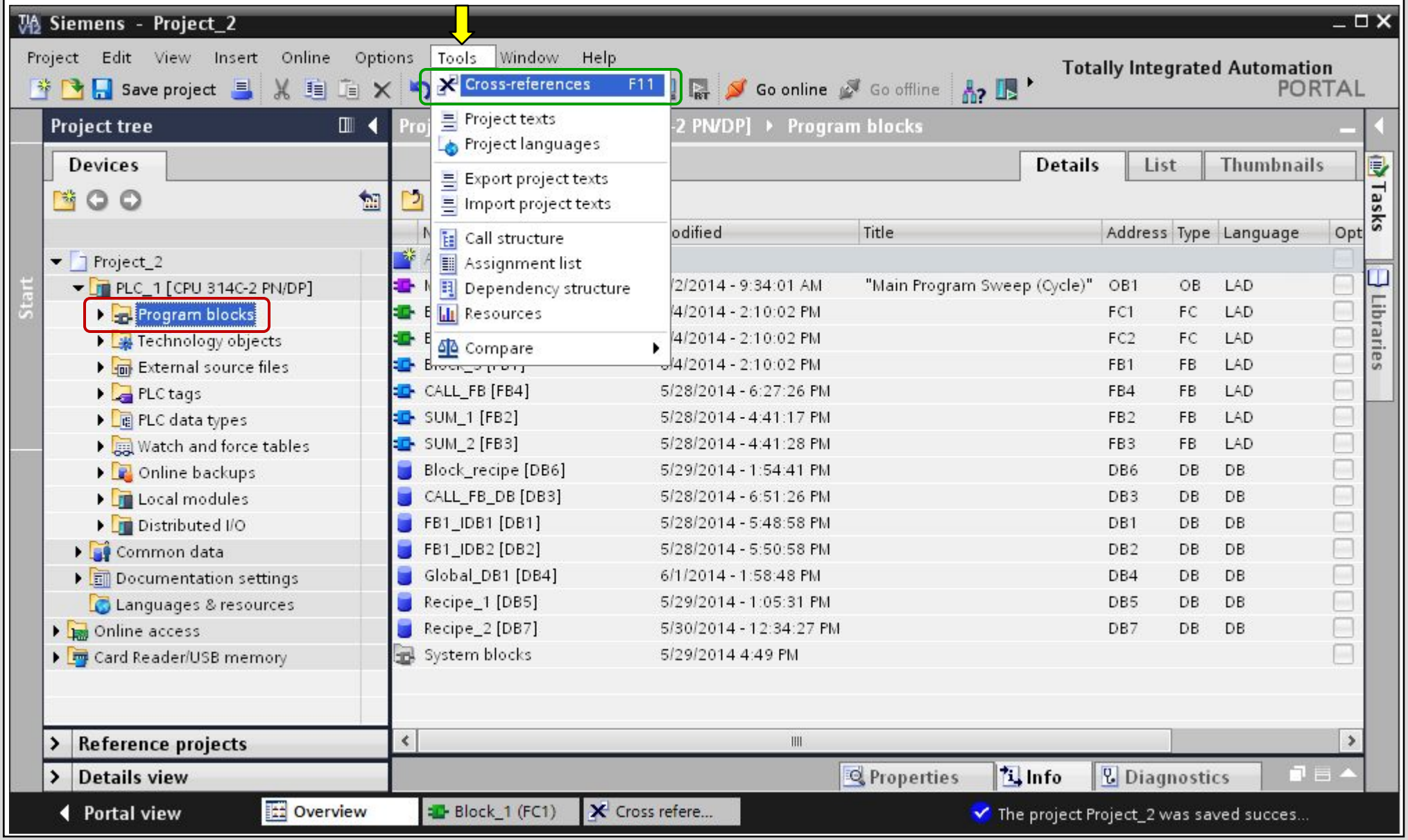
Структура вызовов программных блоков и функций



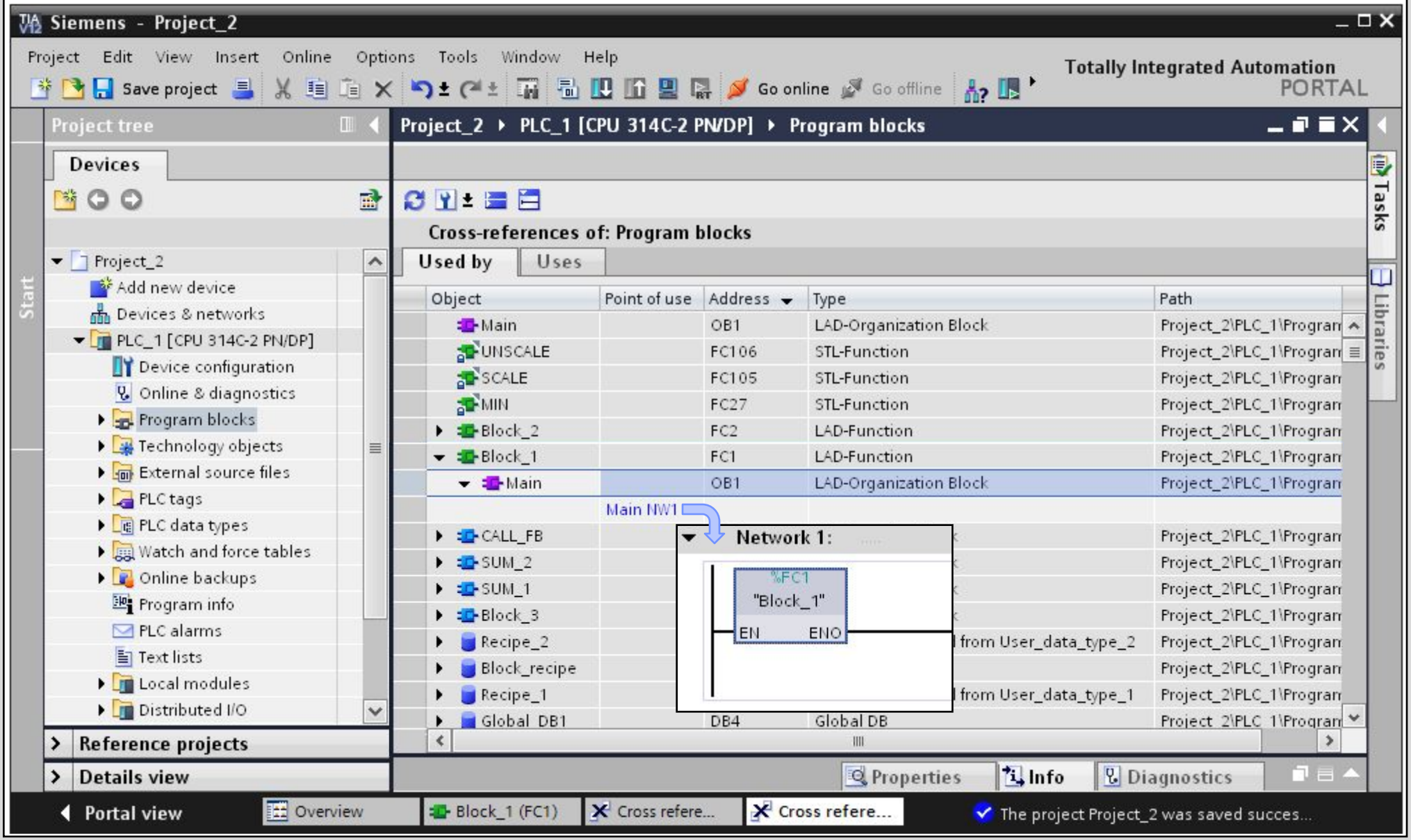
Структура зависимостей вызовов



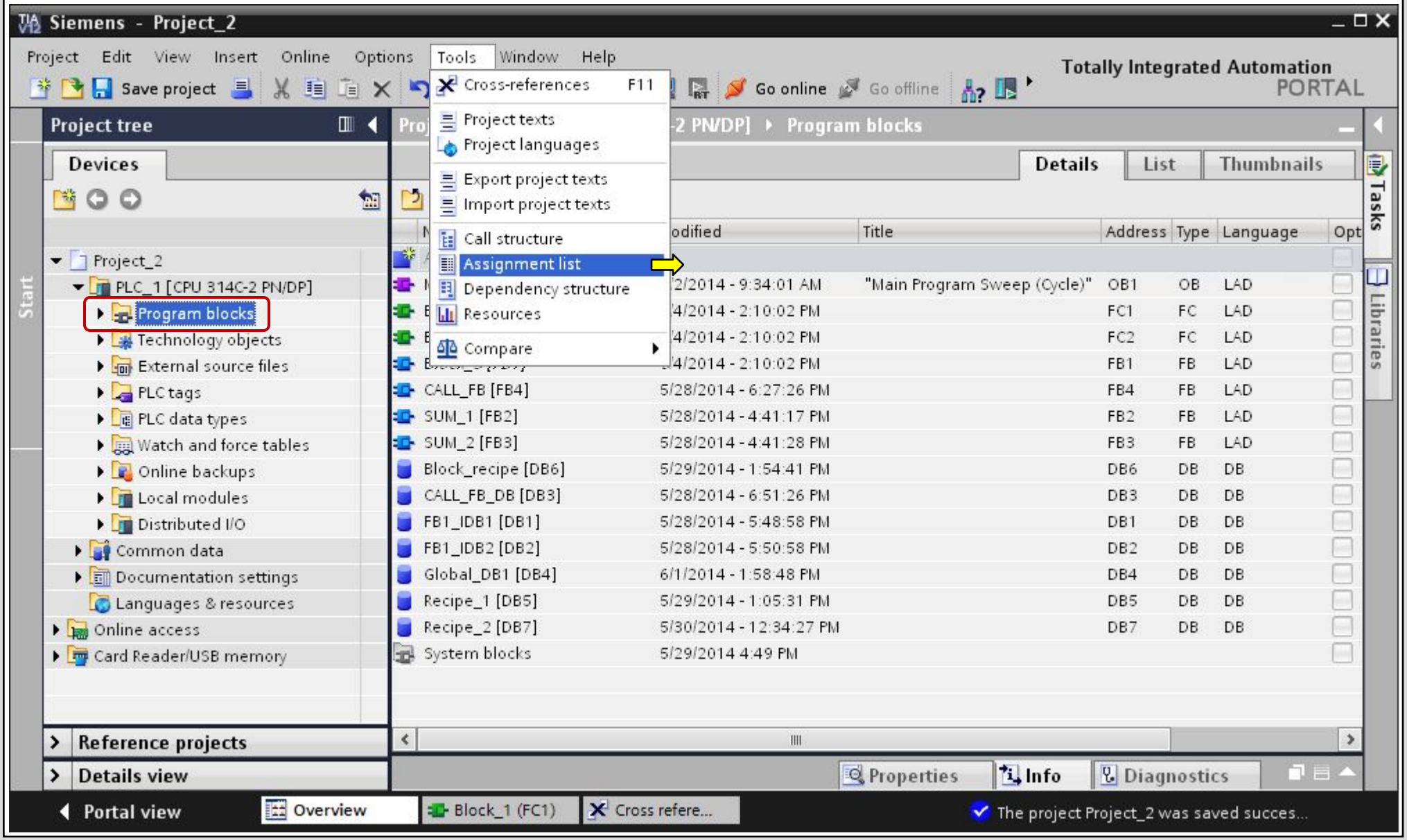
Поиск точки вызова программного блока



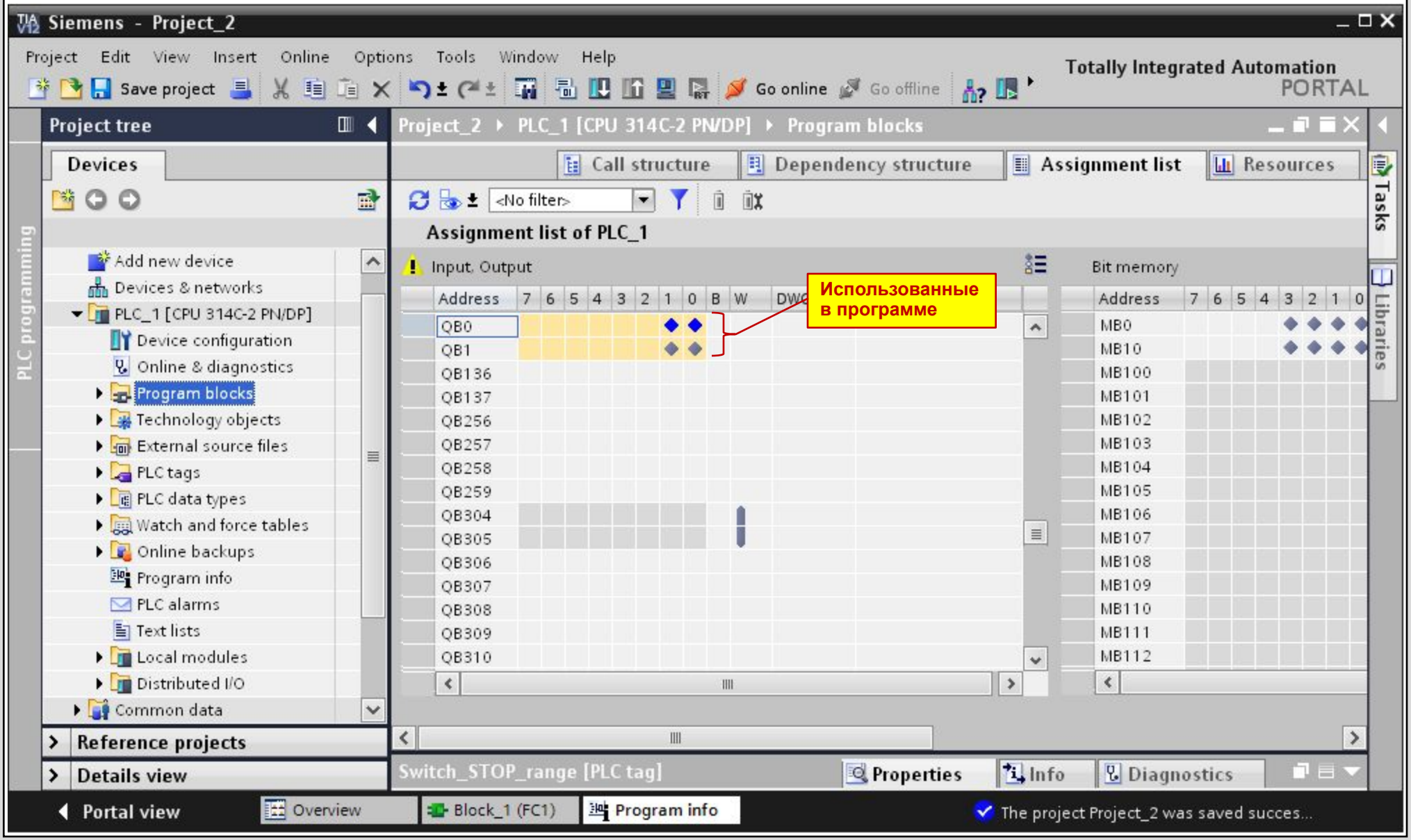
Поиск точки вызова программного блока



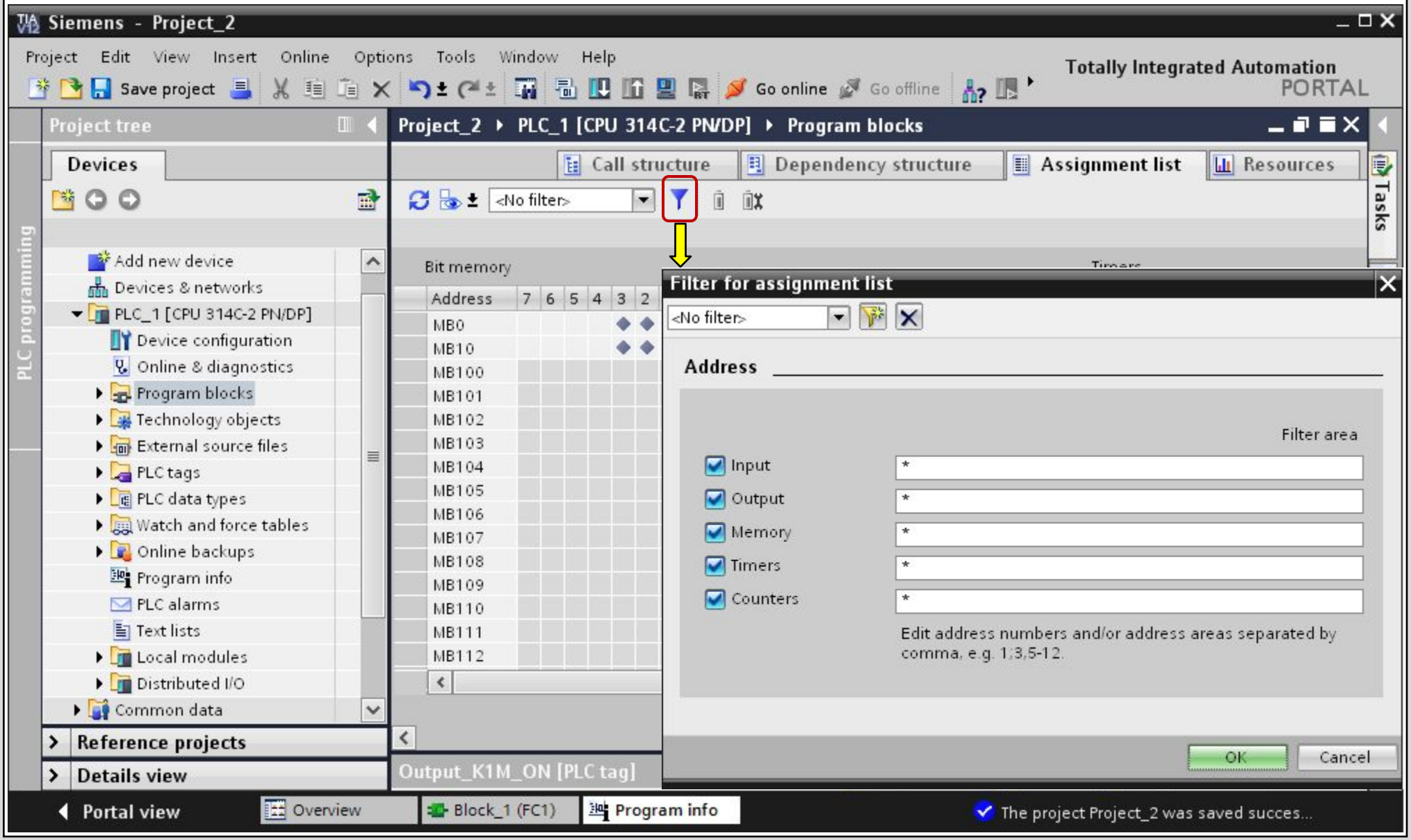
План использования переменных



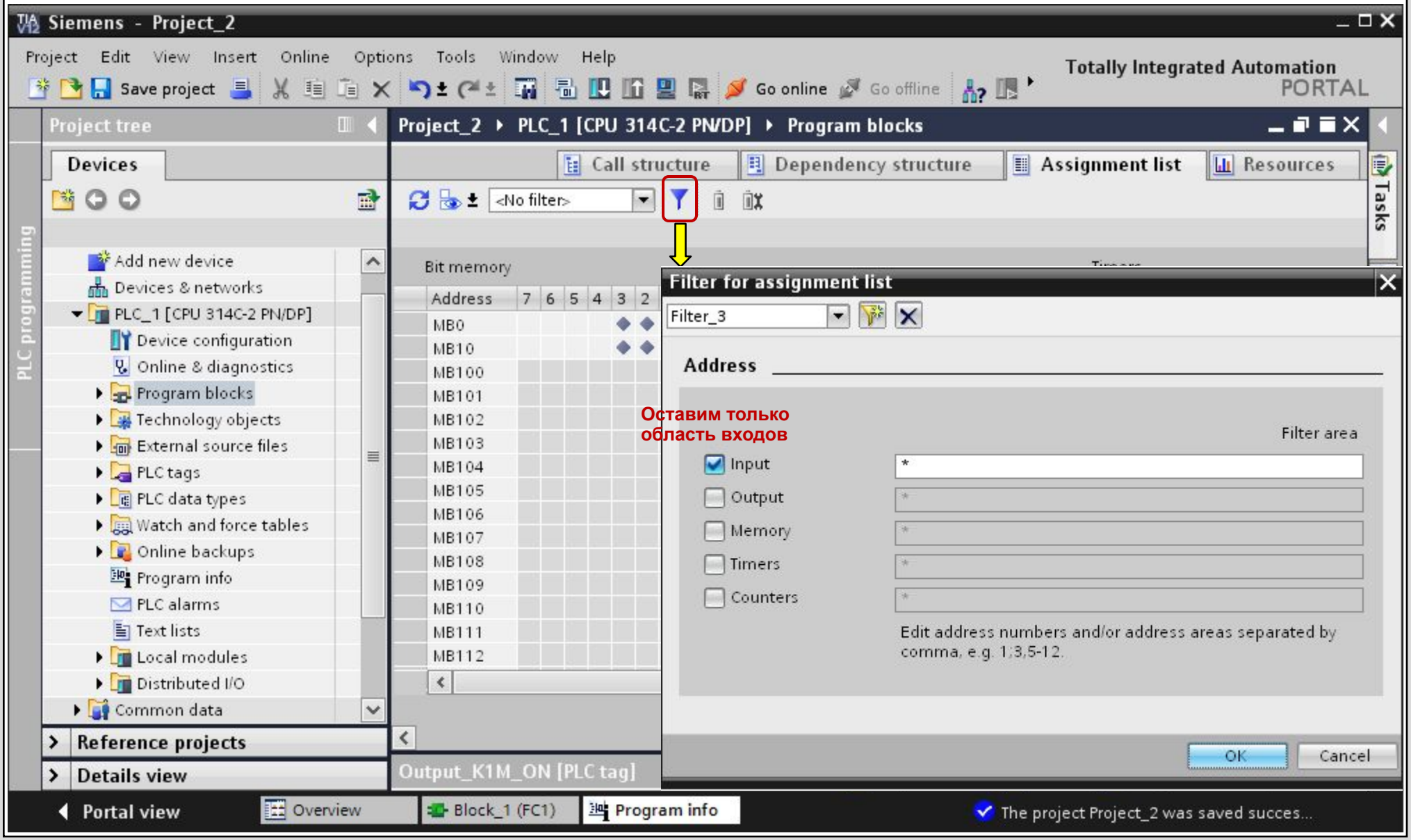
План использования переменных



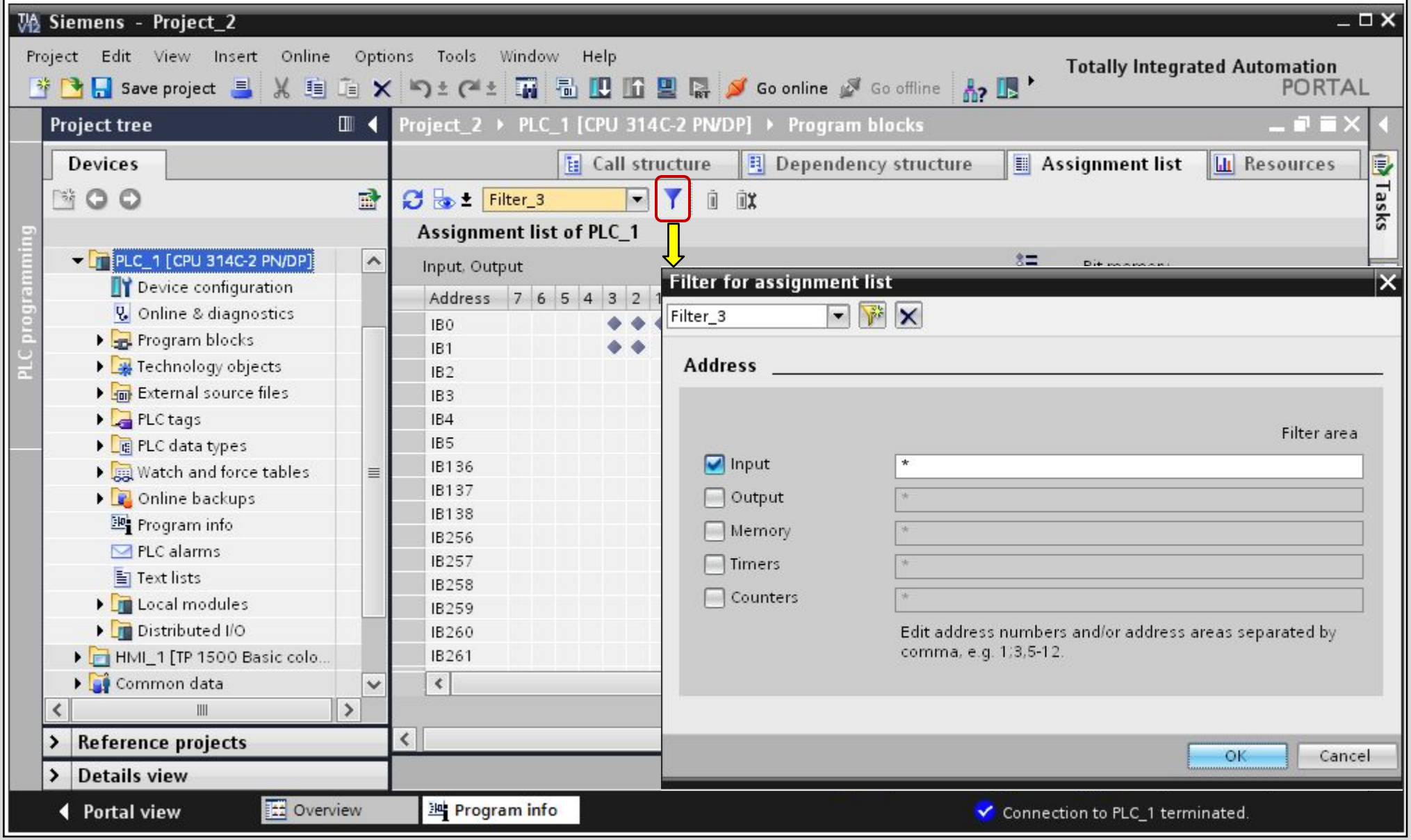
План использования переменных



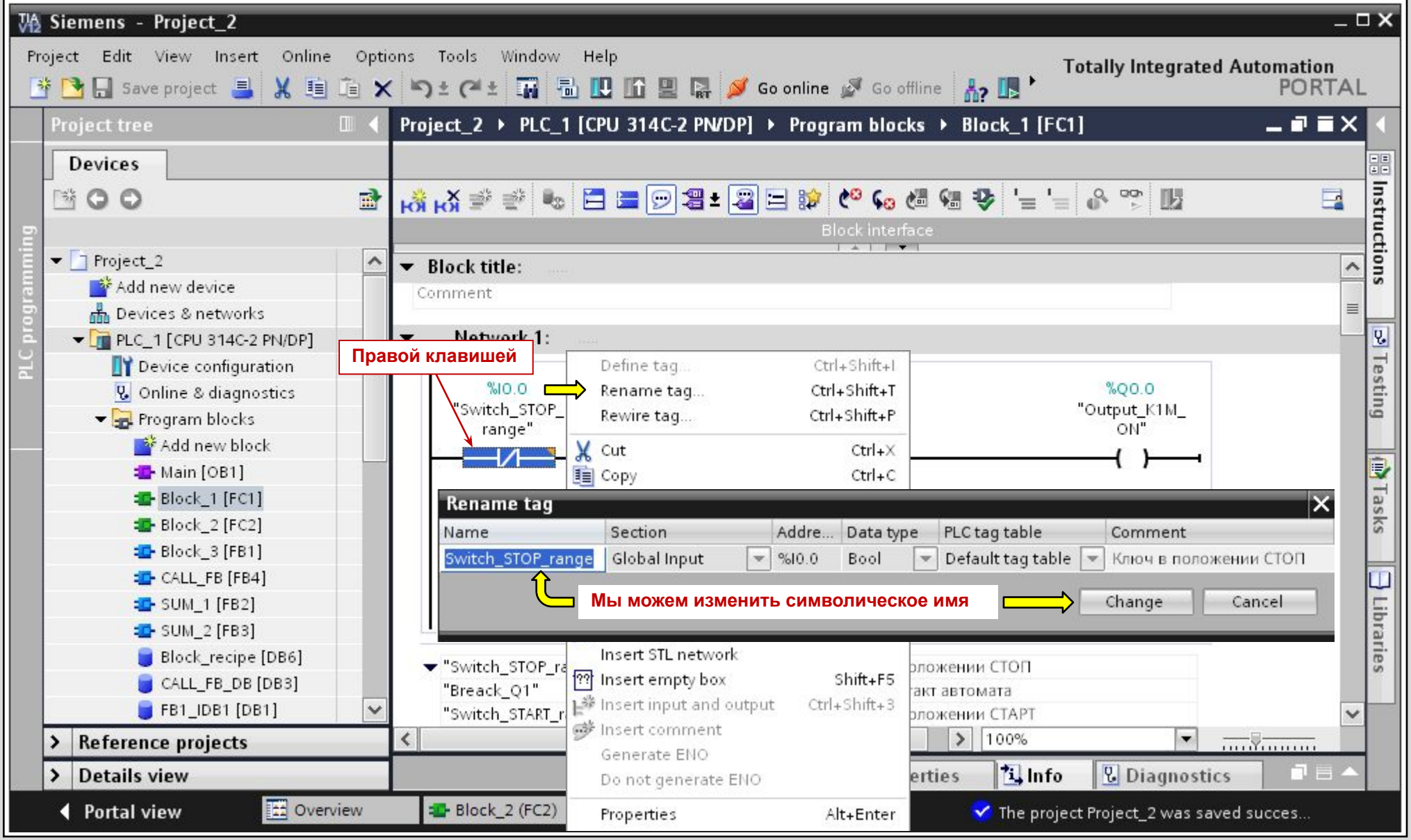
План использования переменных



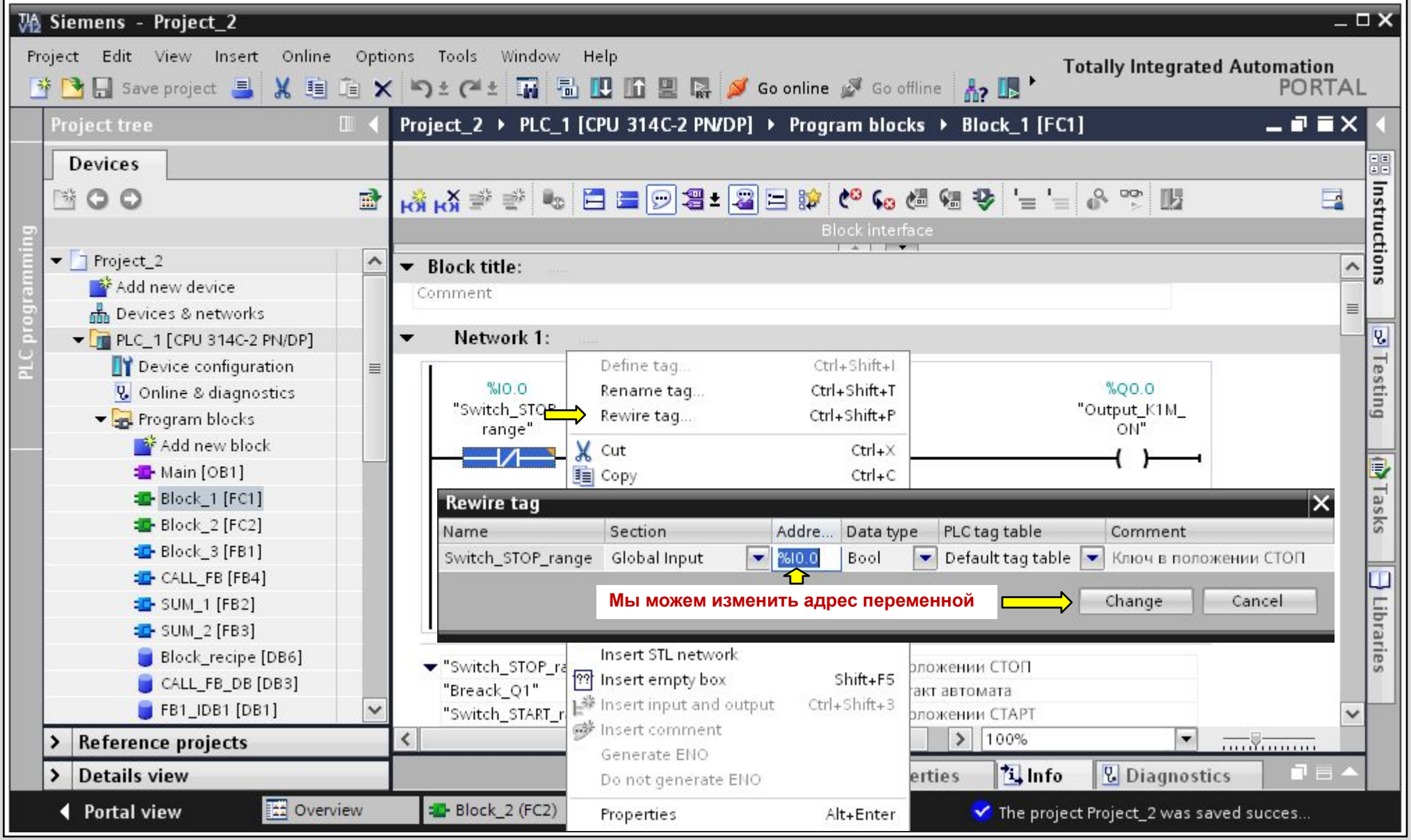
Фильтр при определении плана использования переменных



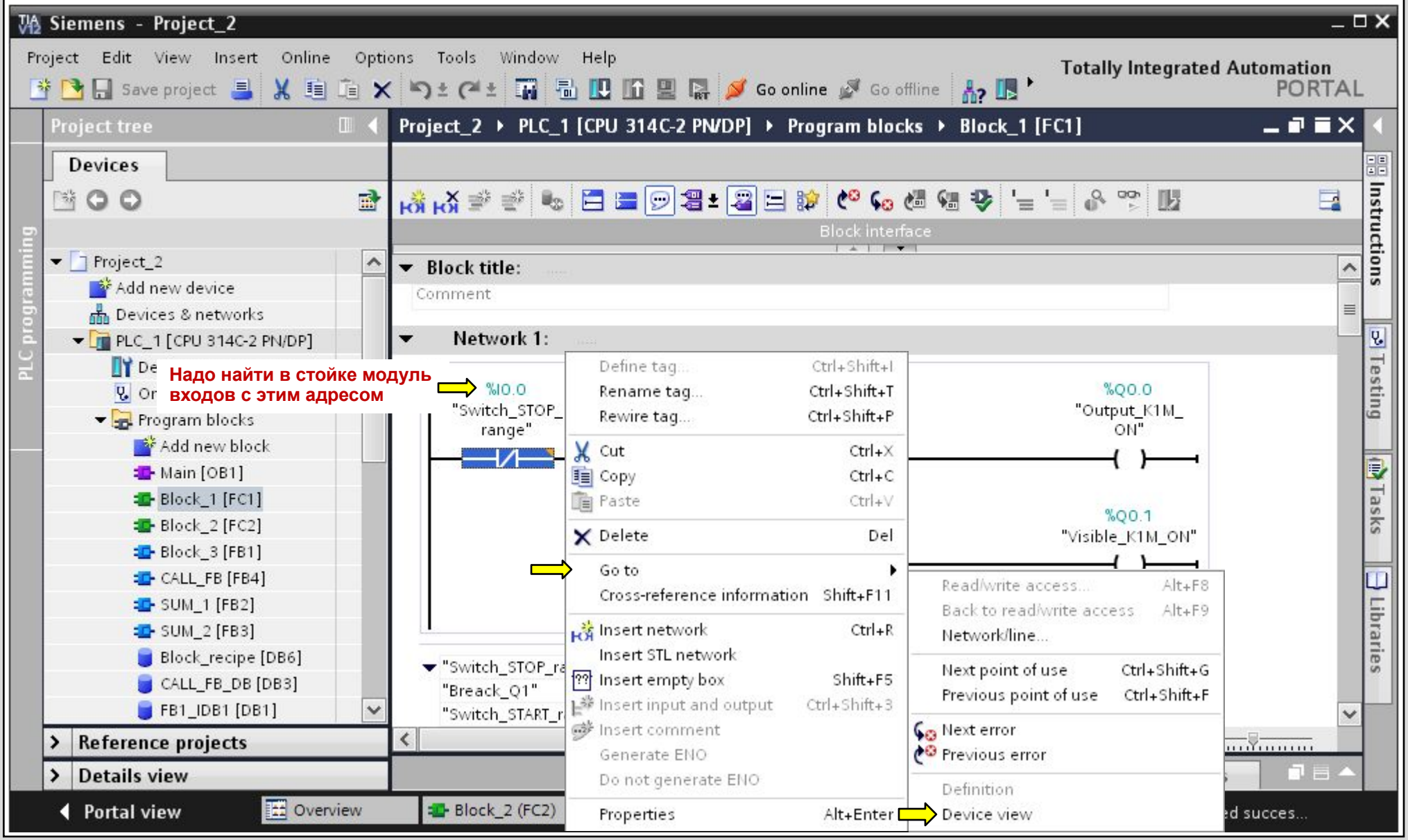
Изменение символического имени



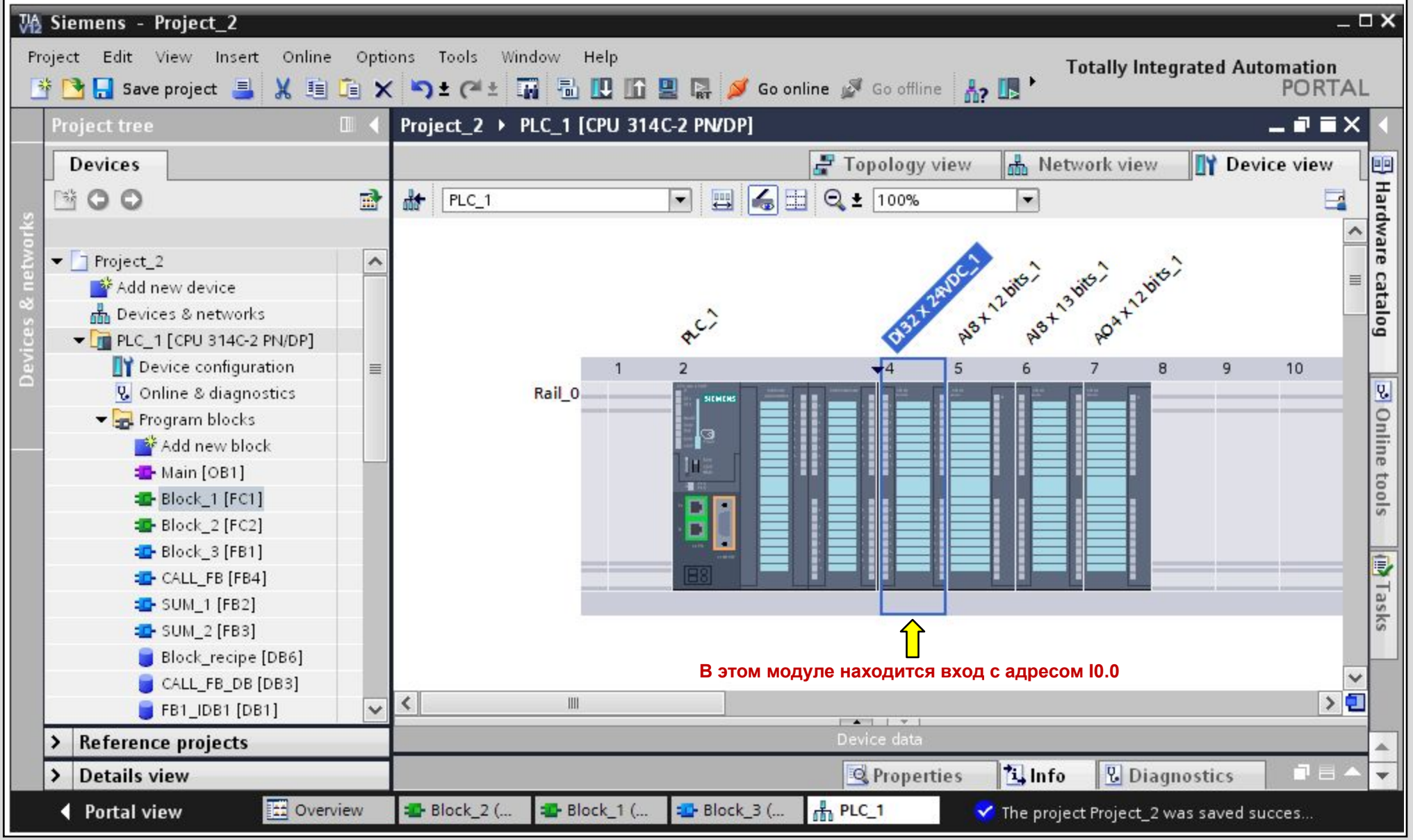
Изменение адресации переменной (перемонтаж)



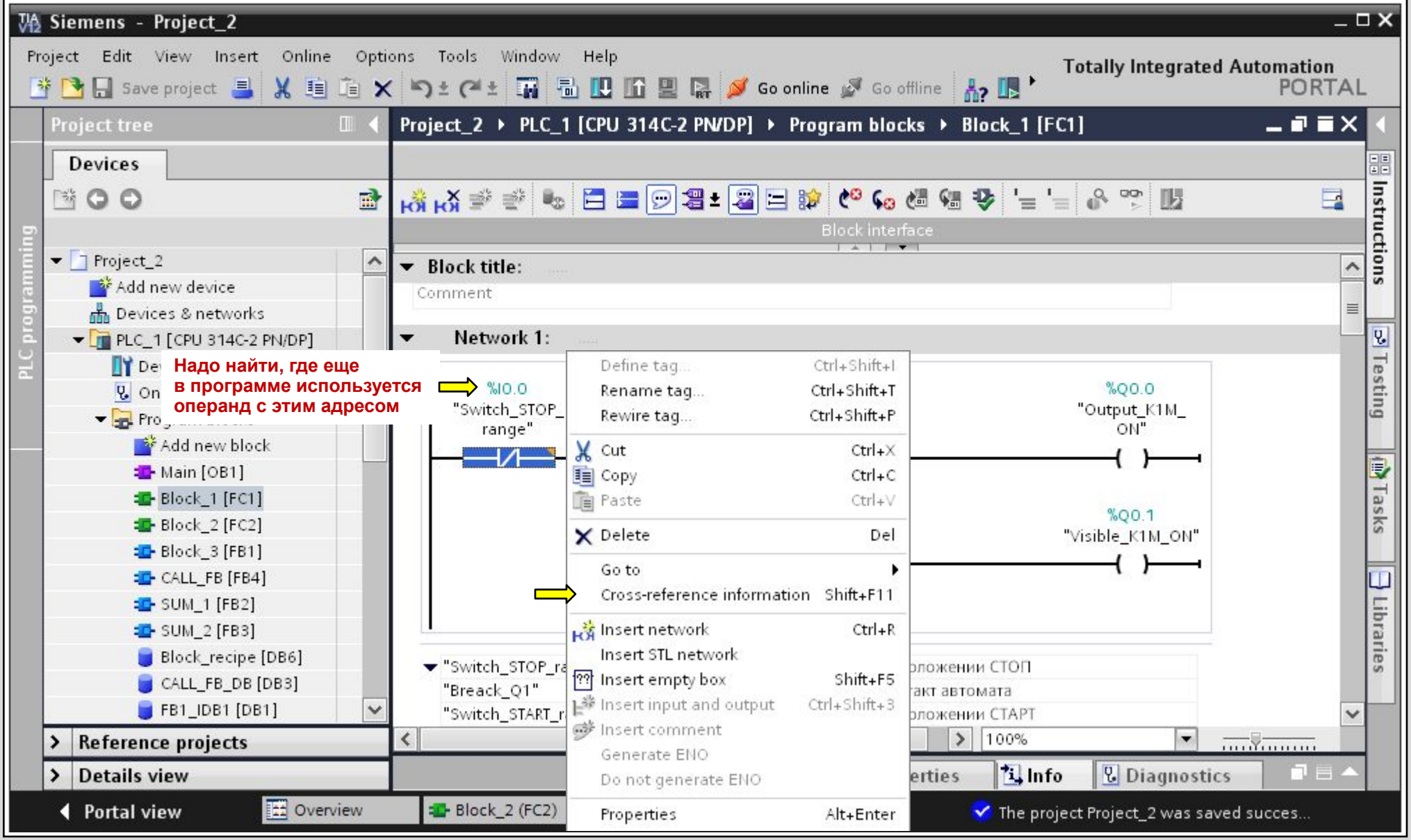
Определение модуля с физическим адресом переменной



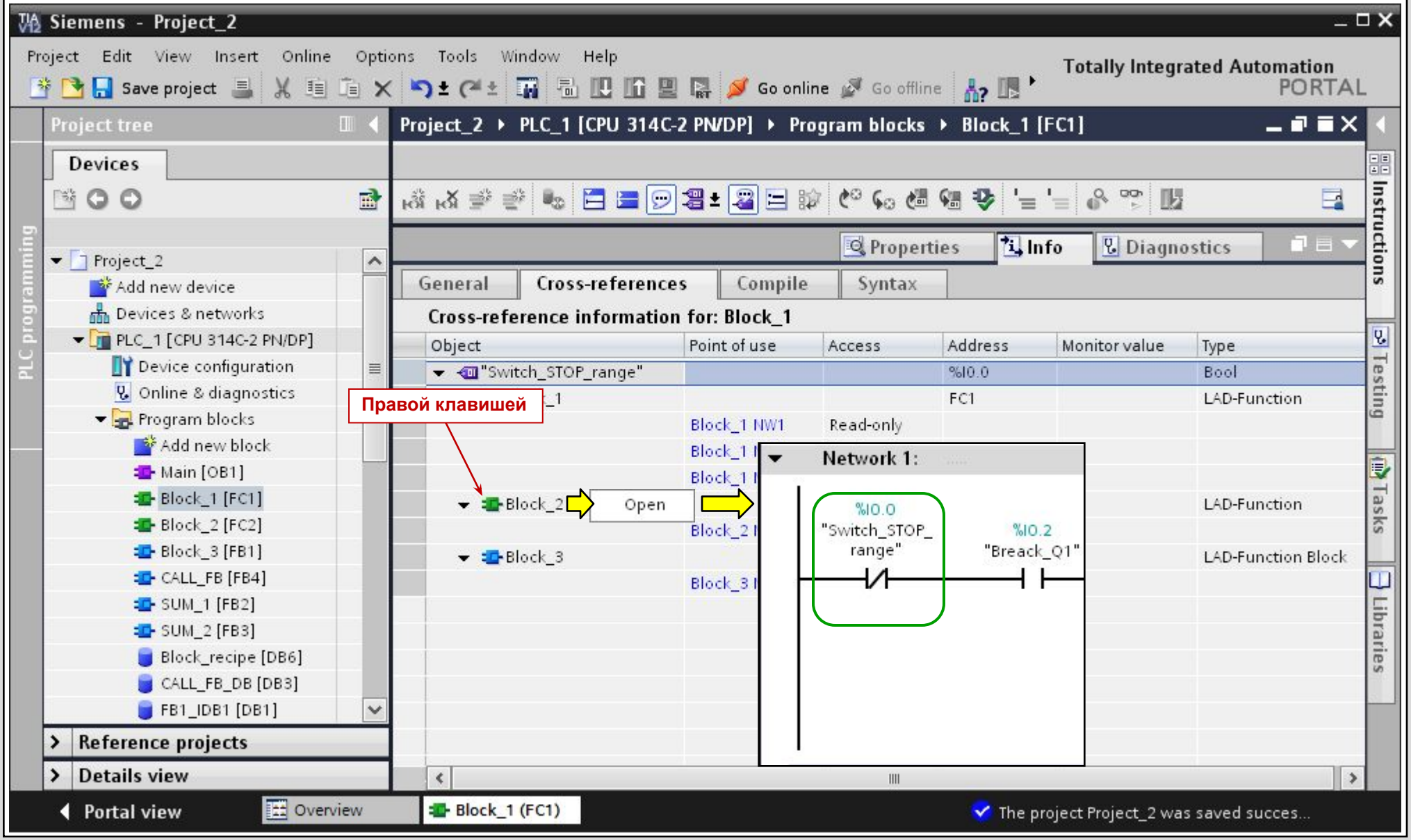
Определение модуля с физическим адресом переменной



Поиск использования в программе выбранной переменной



Поиск использования в программе выбранной переменной

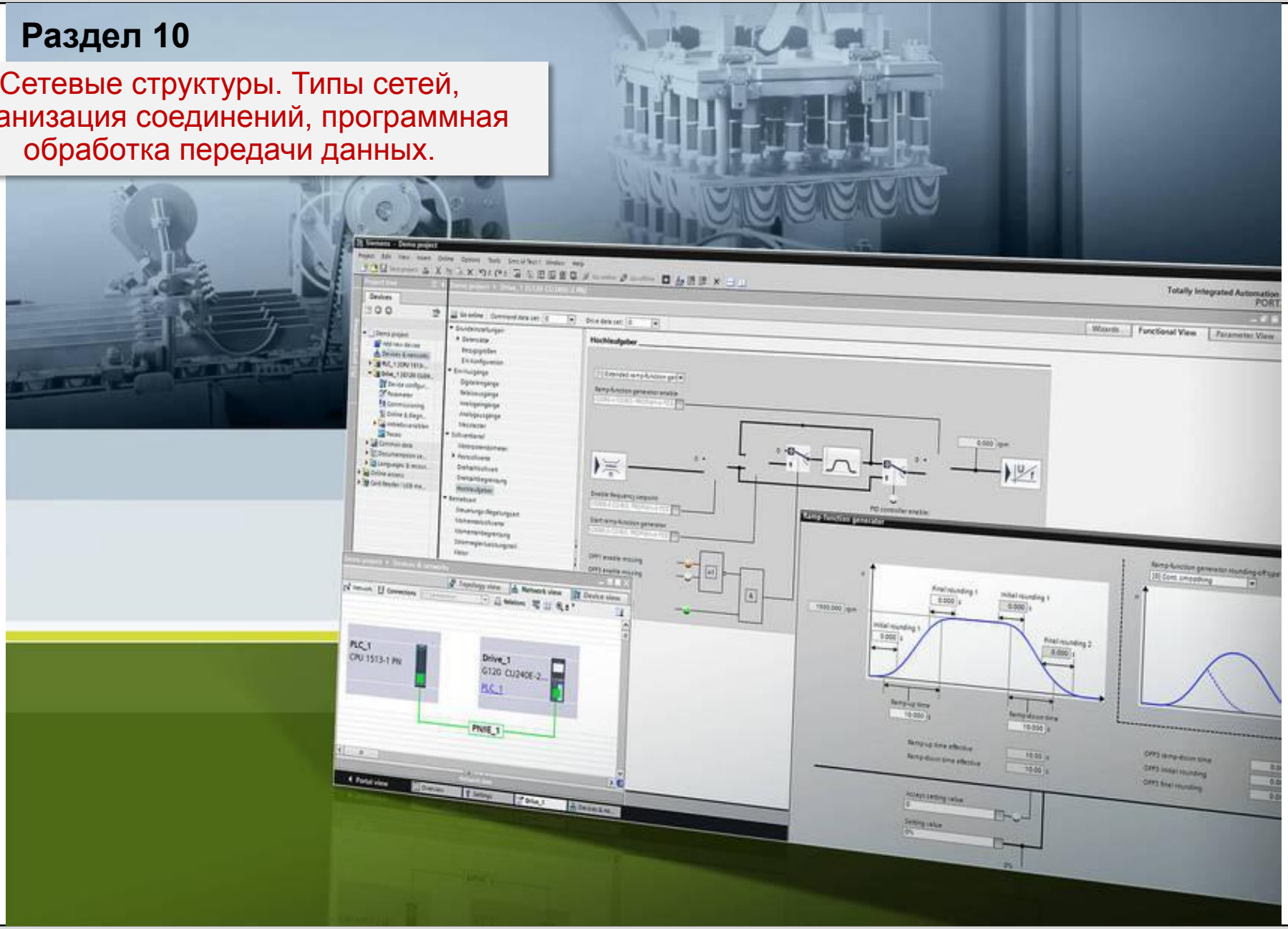


Конец раздела 9. Окно навигации

- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией
- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 10

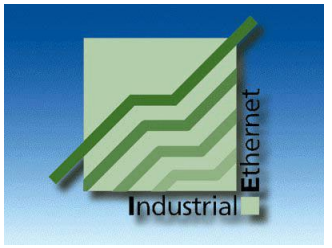
Сетевые структуры. Типы сетей, организация соединений, программная обработка передачи данных.



Шинные системы SIMATIC NET

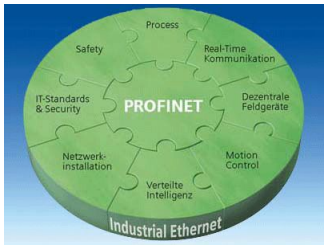
Industrial Ethernet

- Производительная сеть верхнего уровня
- Стандарт IEEE 802.3 (ETHERNET), Стандарт 802.11 (Wireless LAN)



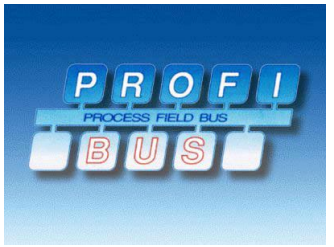
PROFINET

- Для работы в сегменте от полевого уровня до уровня управления
- Открытый Industrial Ethernet Стандарт (IEC 61158)



PROFIBUS

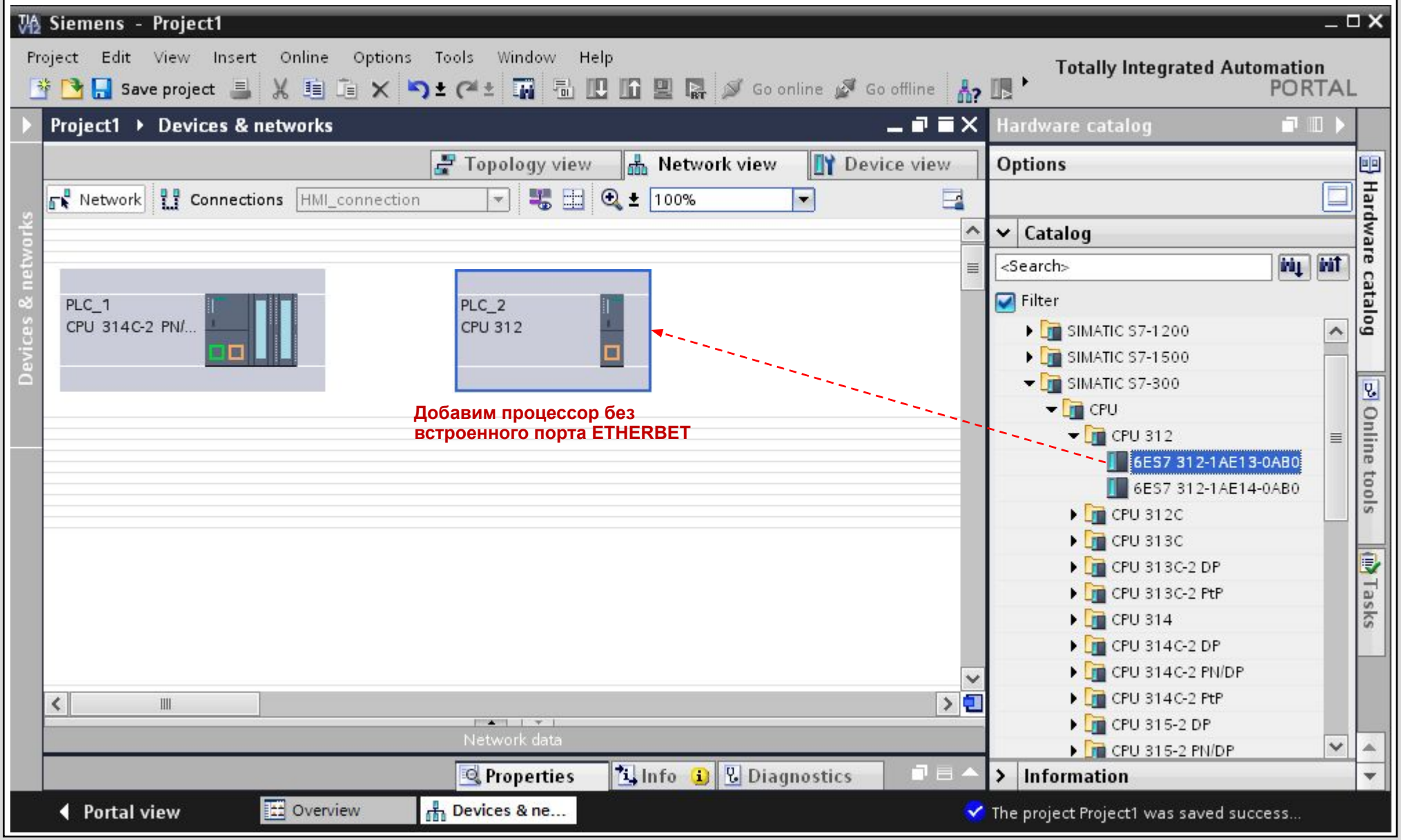
- Для работы в сегменте от полевого уровня до уровня ячеек
- PROFIBUS DP, PROFIBUS PA, PROFIBUS FMS



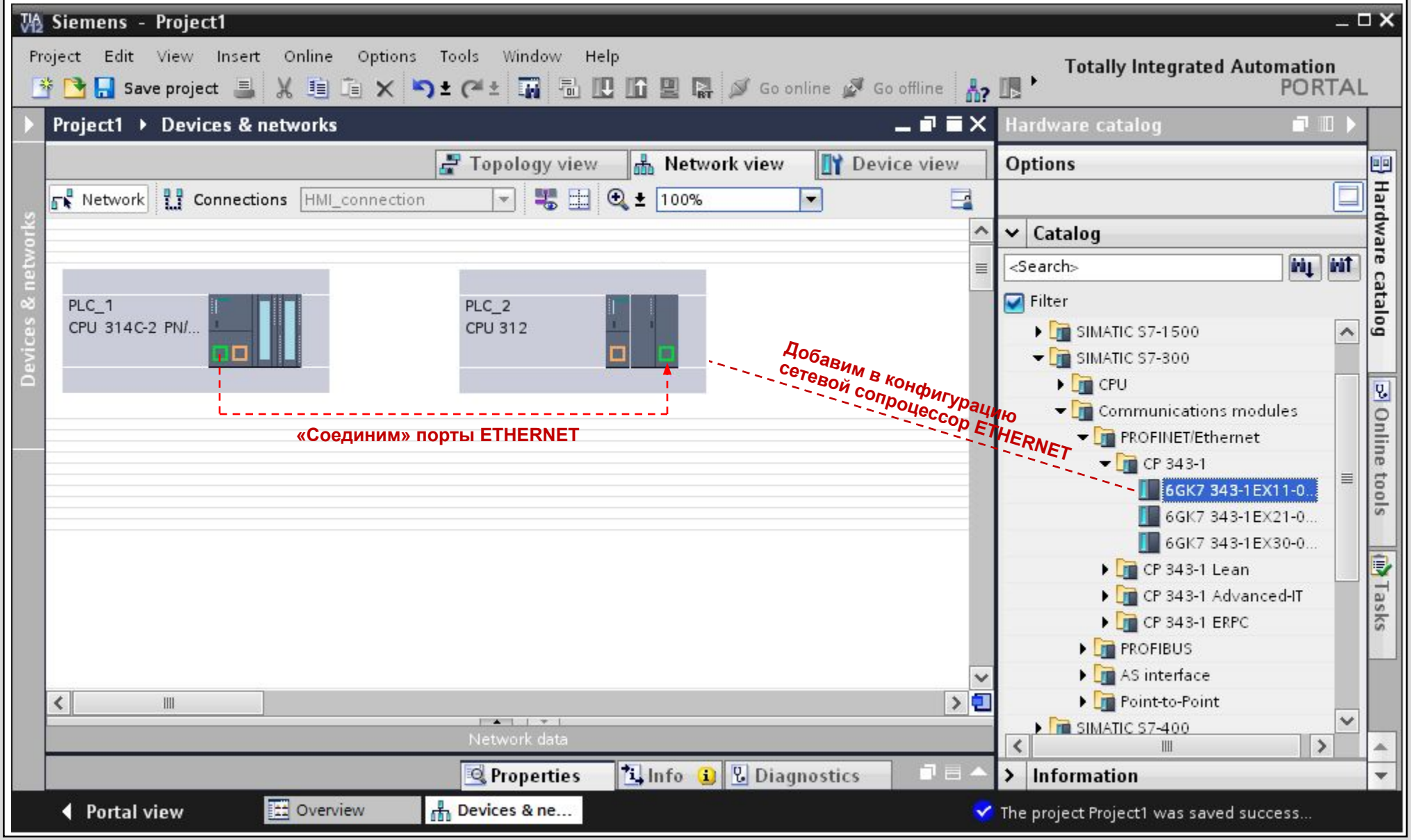
AS-interface

- Непосредственное подключение шины полевого уровня к датчикам и исполнительным устройствам

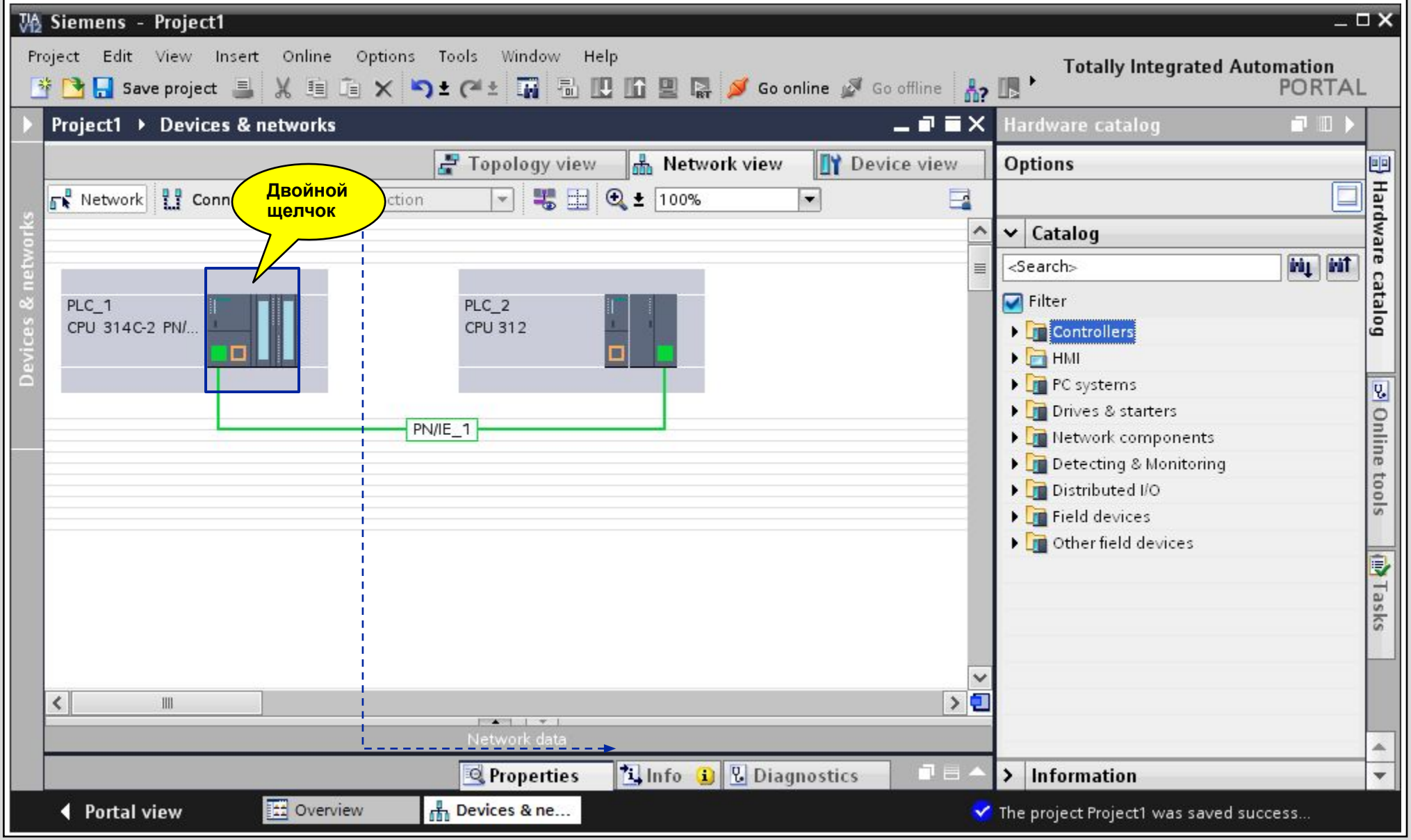
Создание одноранговой сети PLC – PLC на базе ETHERNET



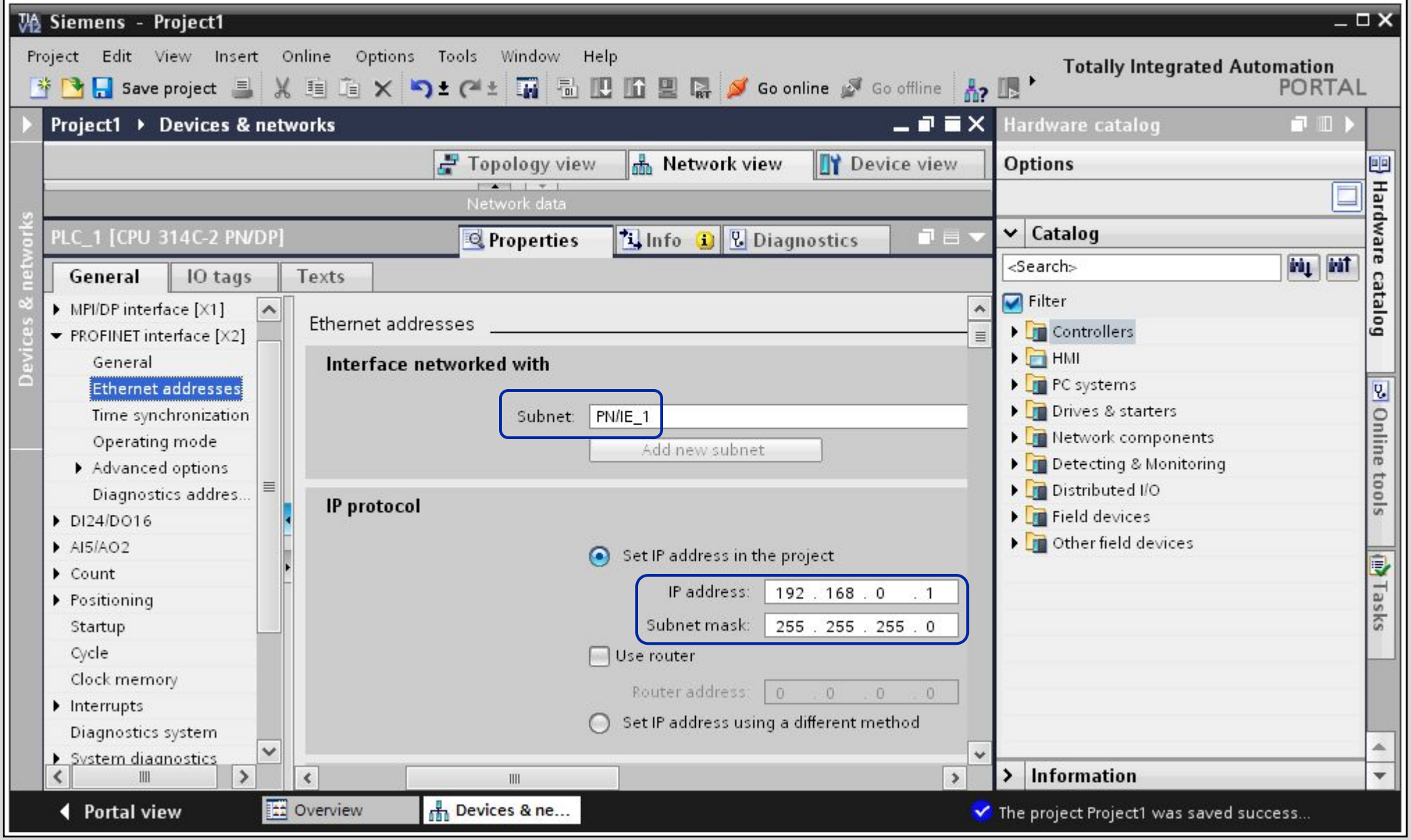
Создание одноранговой сети PLC – PLC на базе ETHERNET



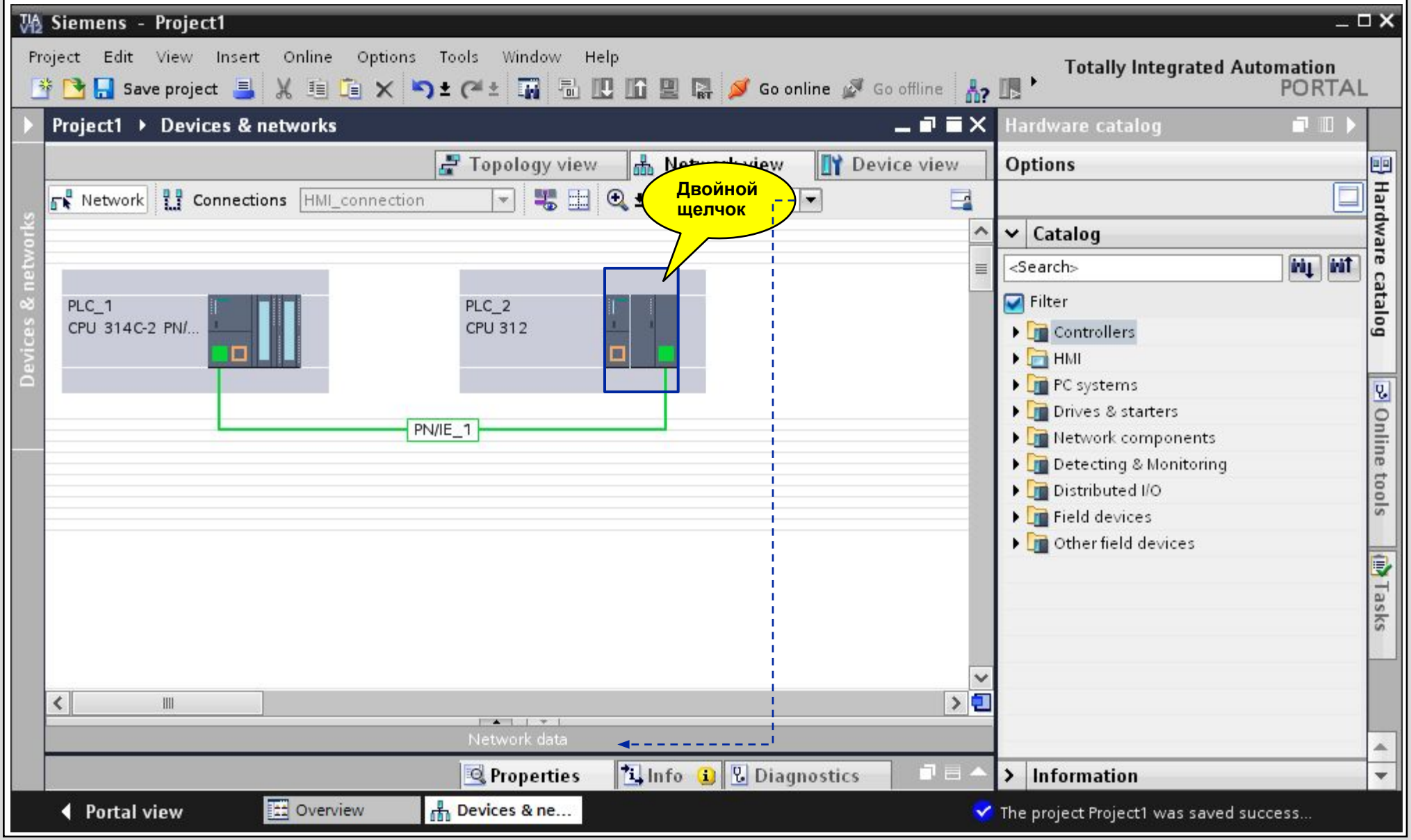
Вызов окна свойств первого процессора



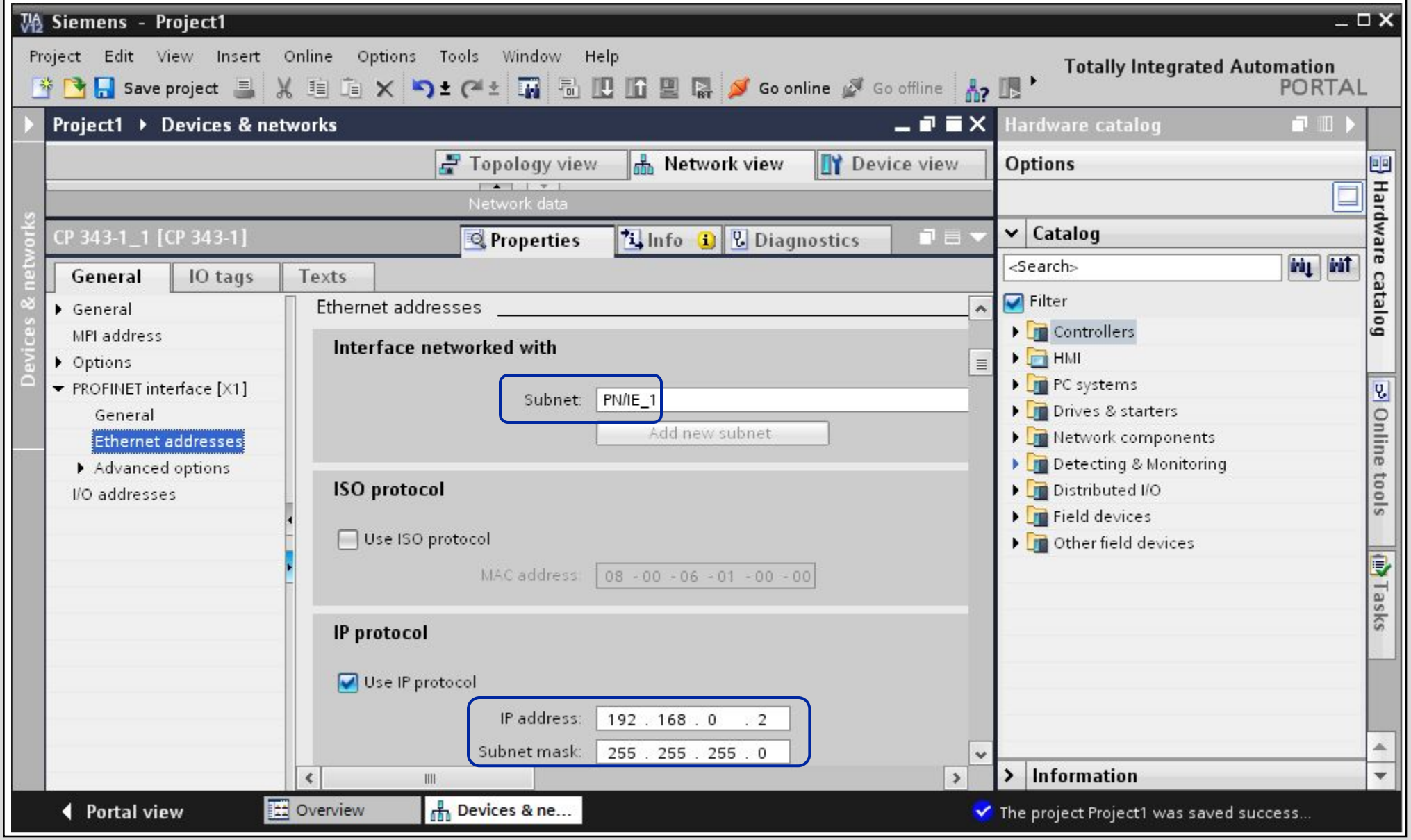
Сетевой адрес первого процессора



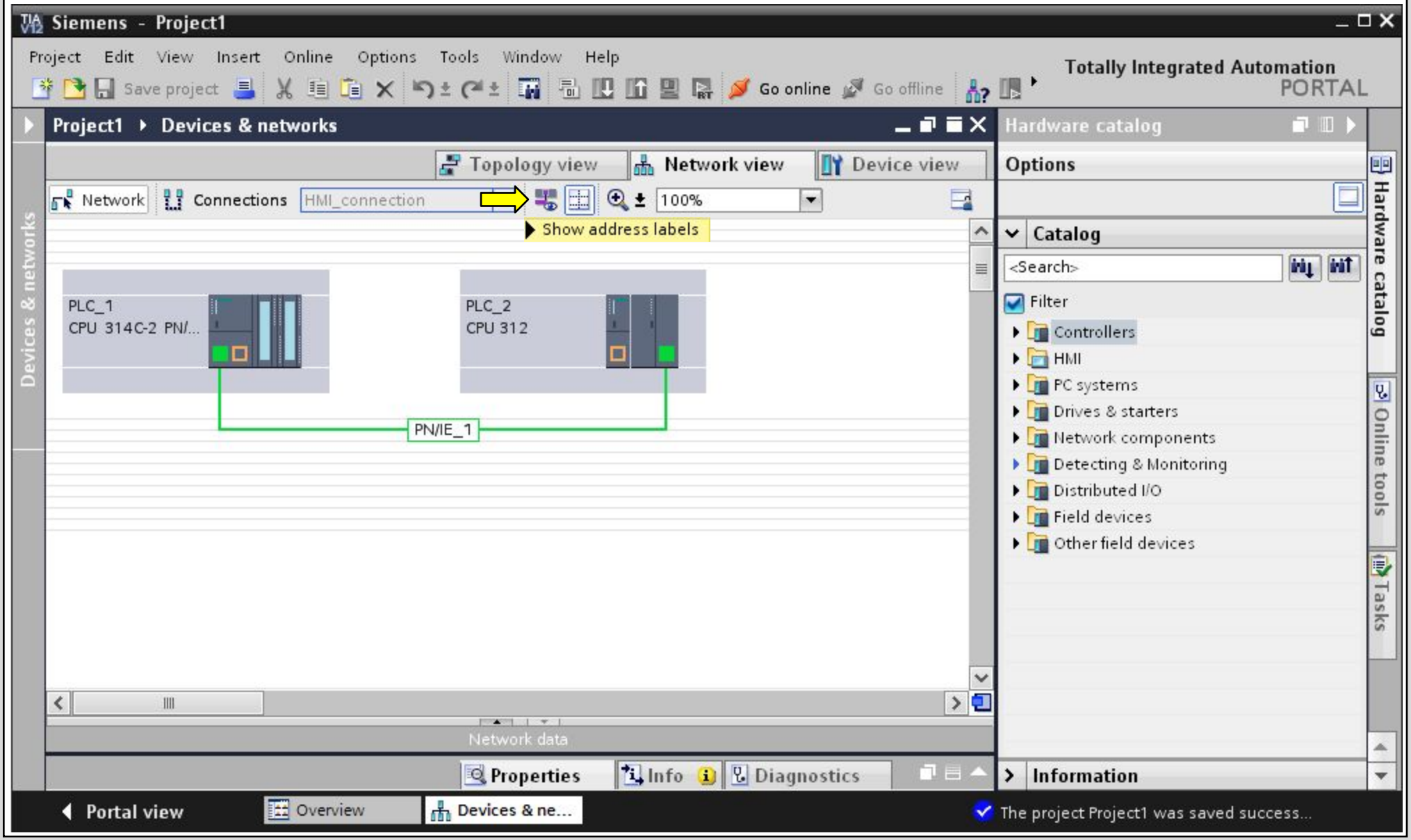
Вызов окна свойств второго процессора



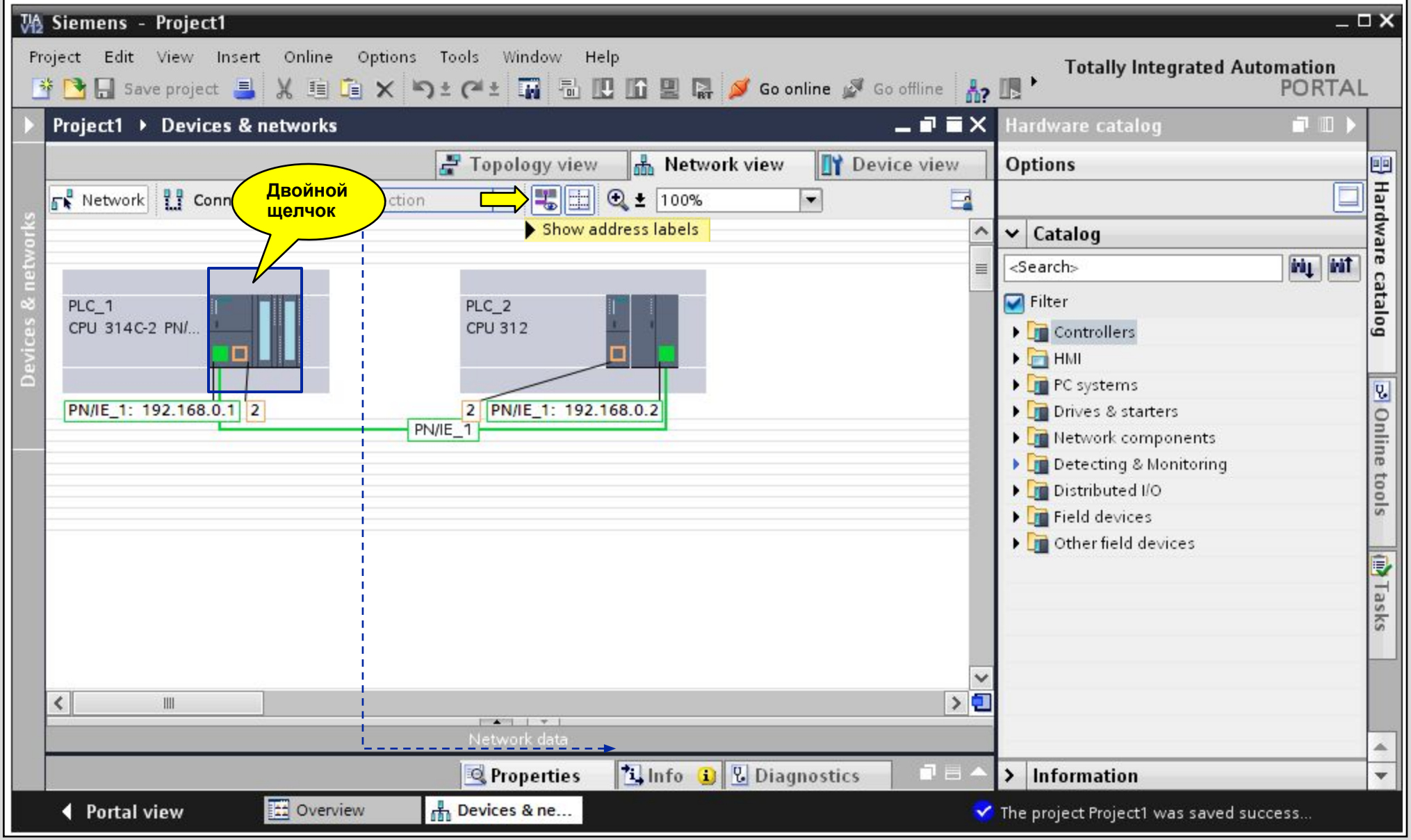
Сетевой адрес второго процессора



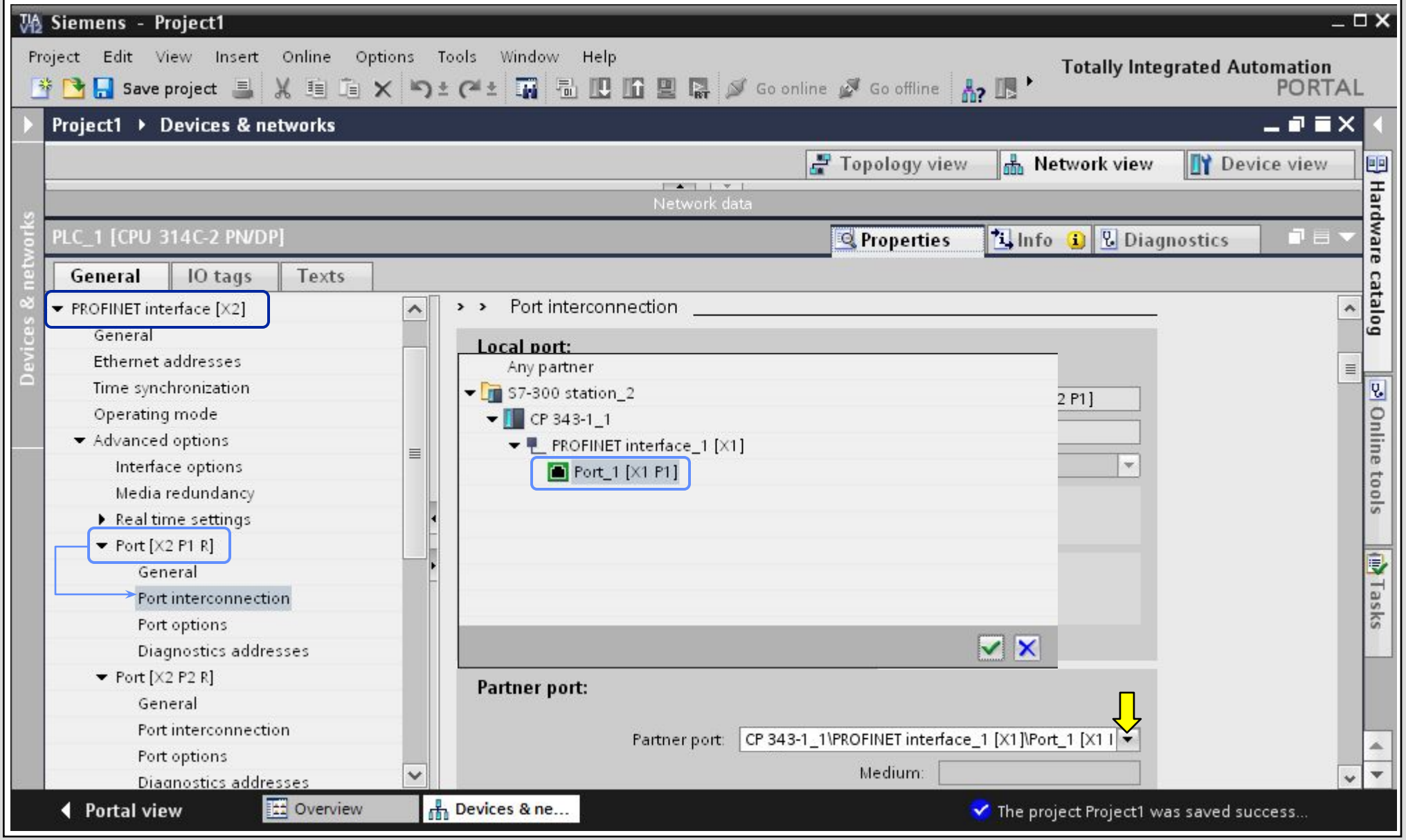
Активация опции «показать адреса абонентов сети»



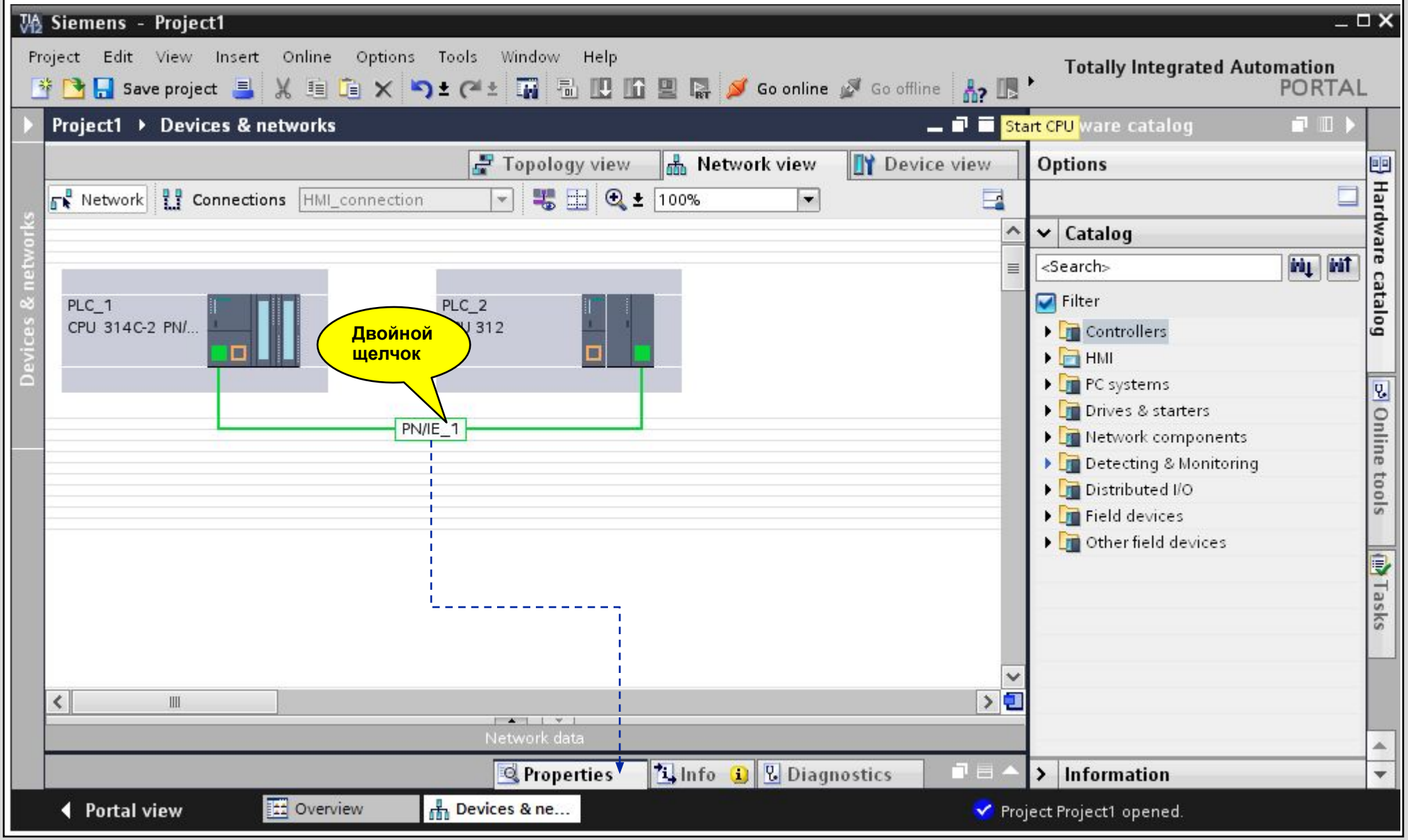
Активация опции «показать адреса абонентов сети»



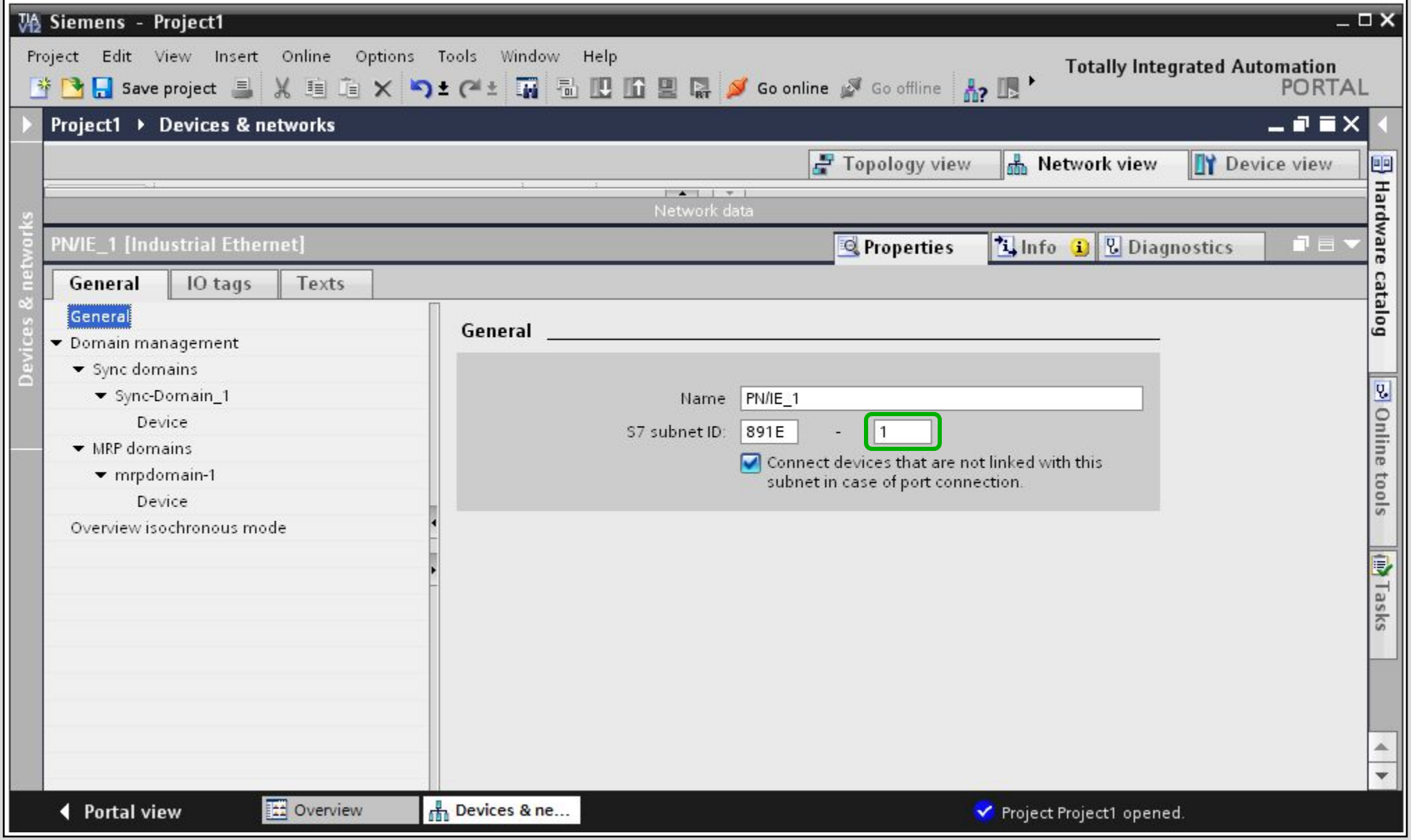
Конфигурирование порта партнера



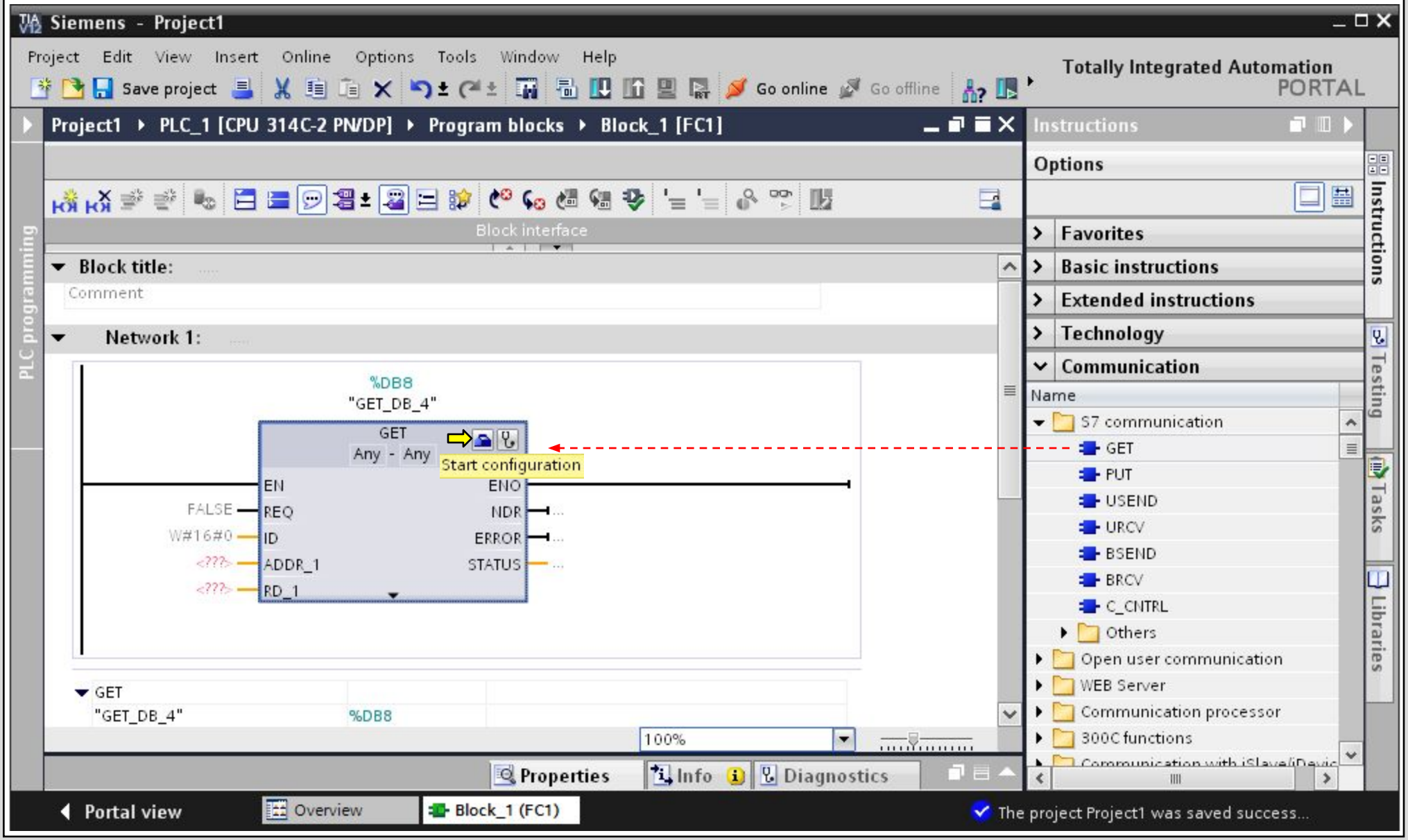
Настройки сети



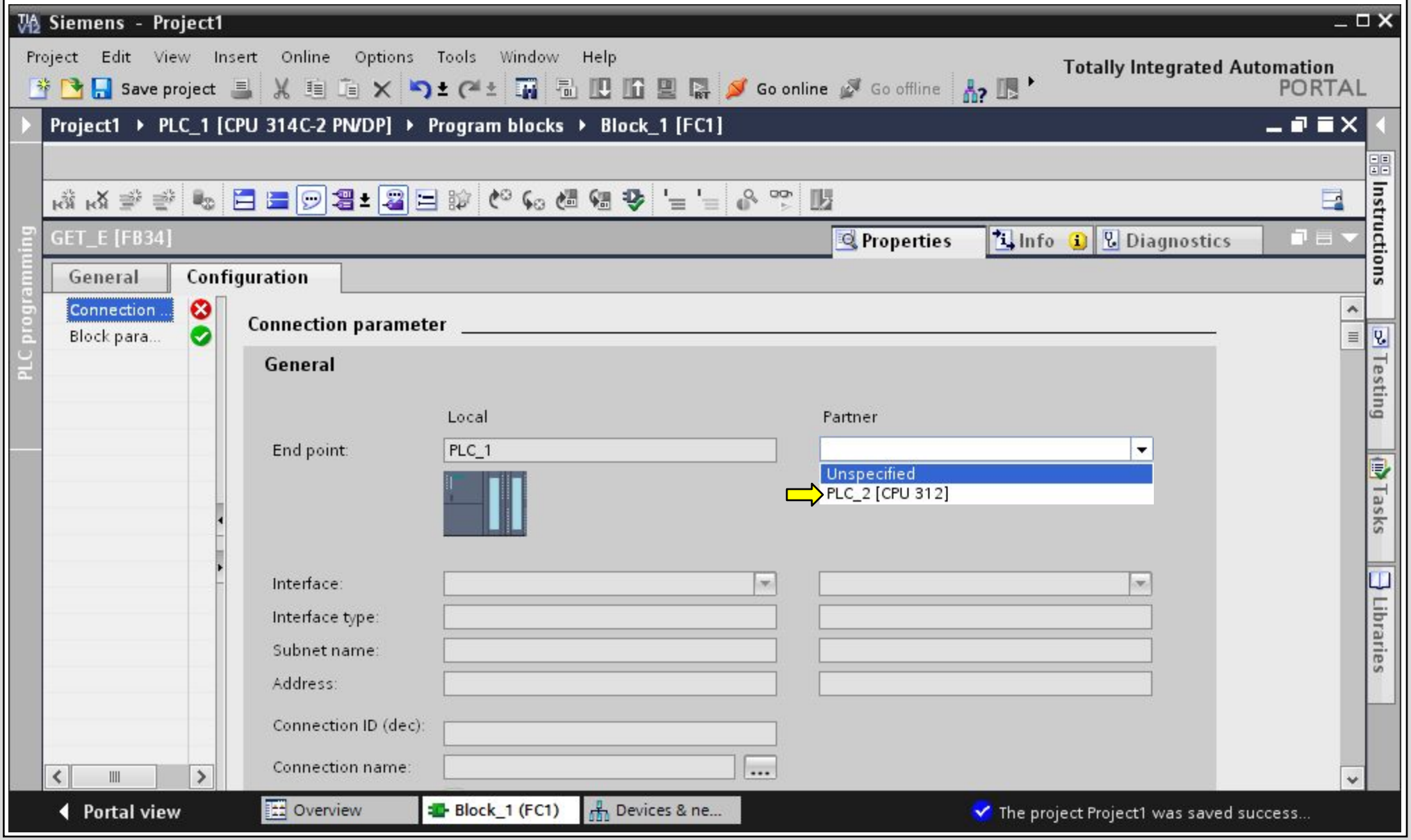
Определение идентификаторов объектов сети



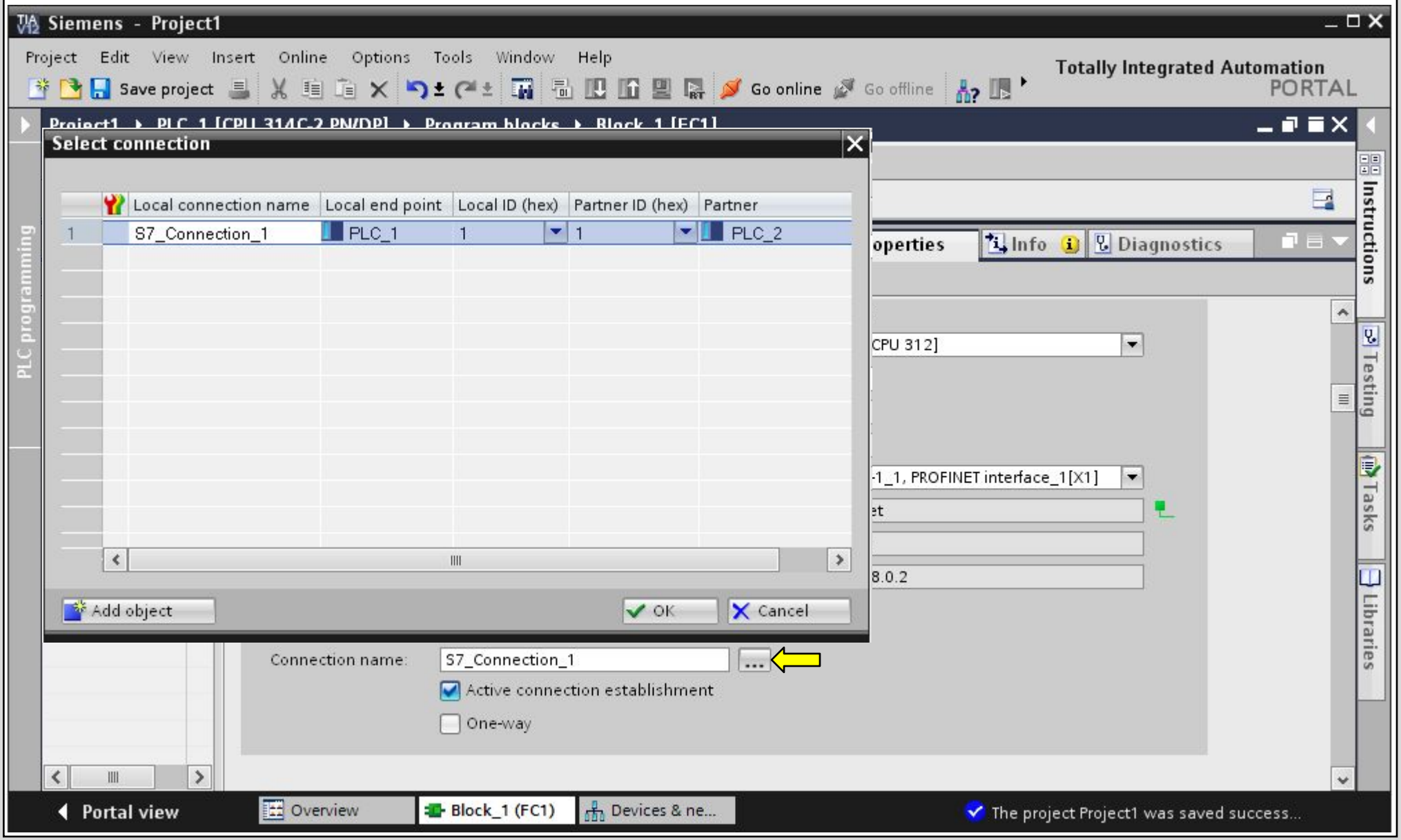
Создание и программирование логического соединения



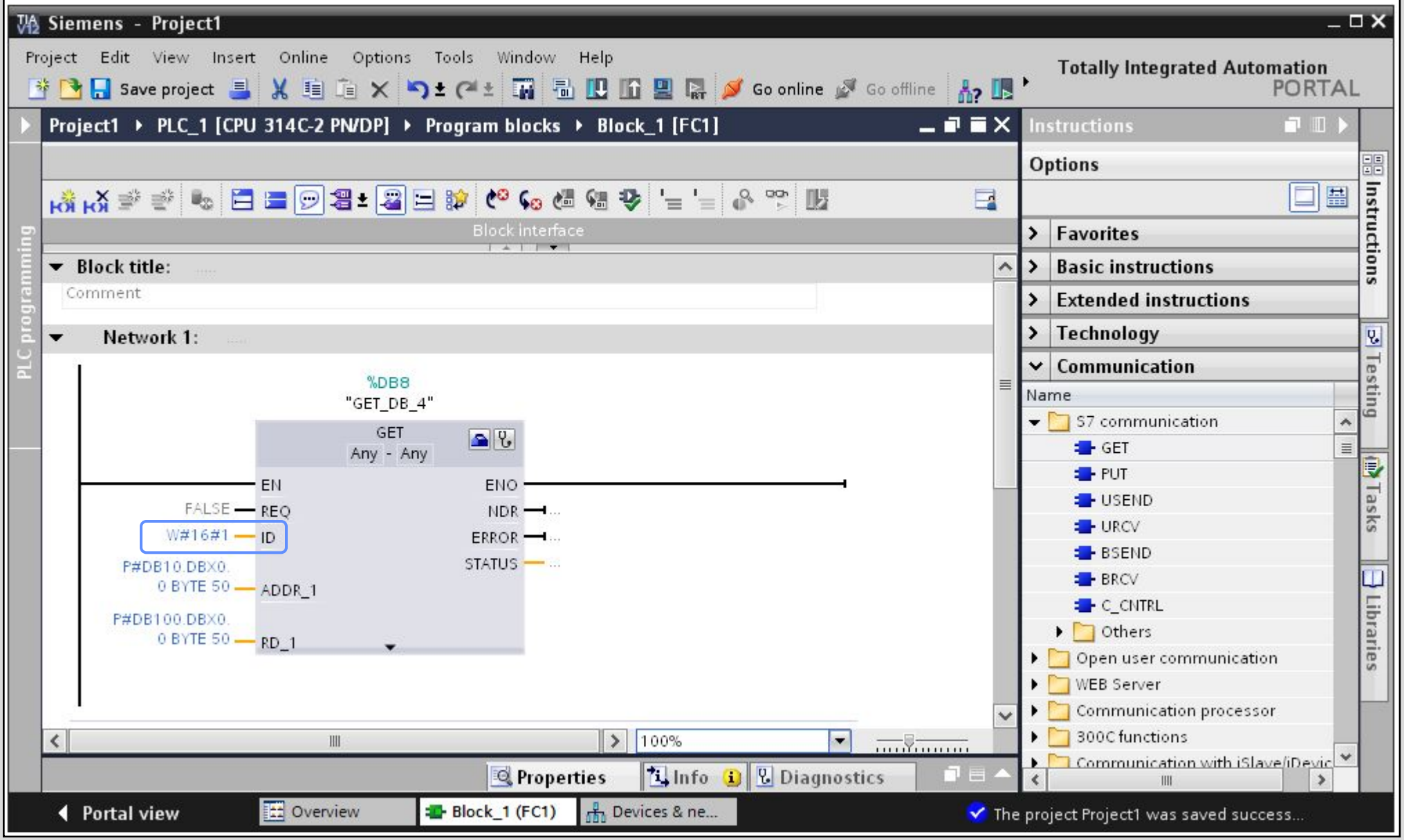
Создание и программирование логического соединения



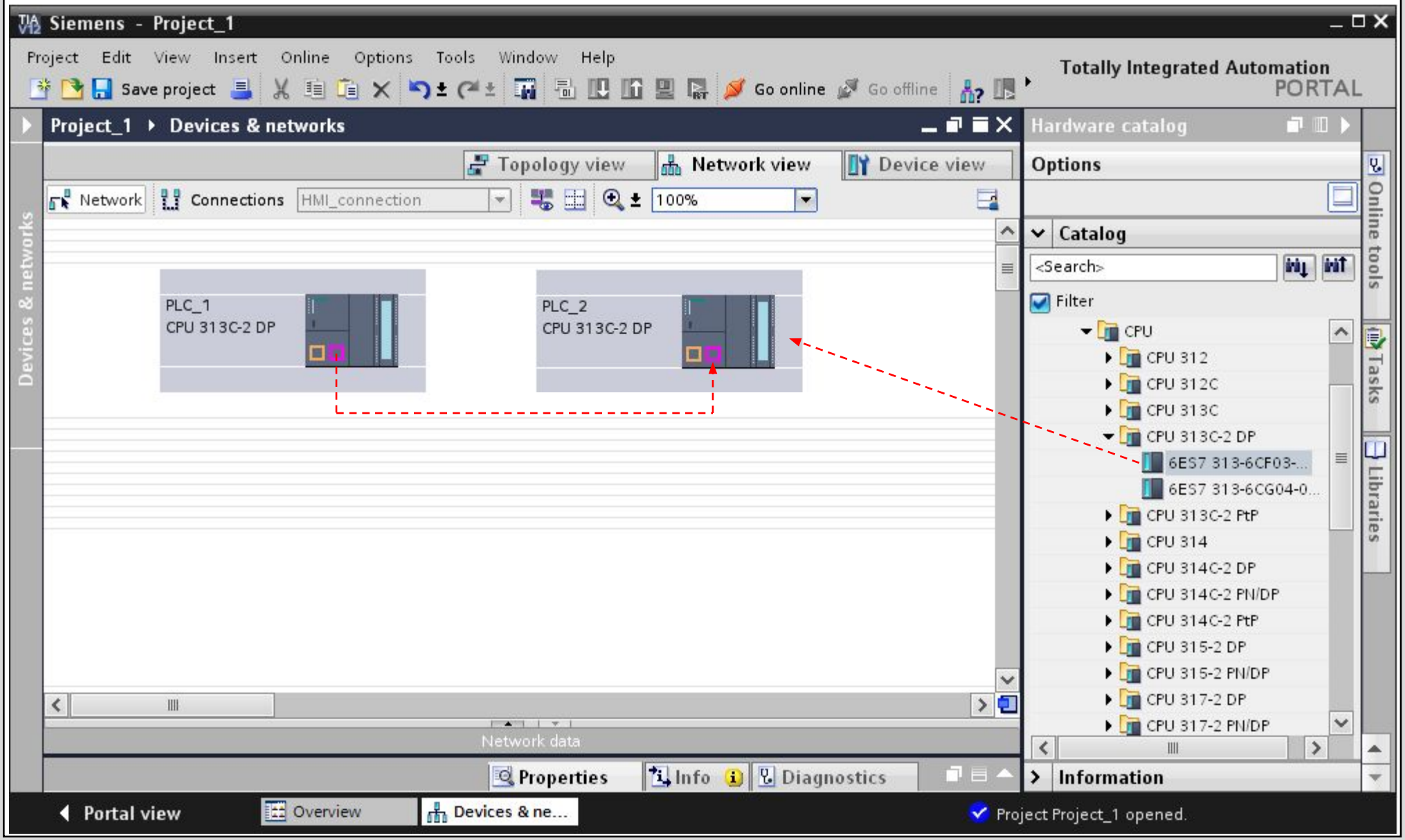
Создание и программирование логического соединения



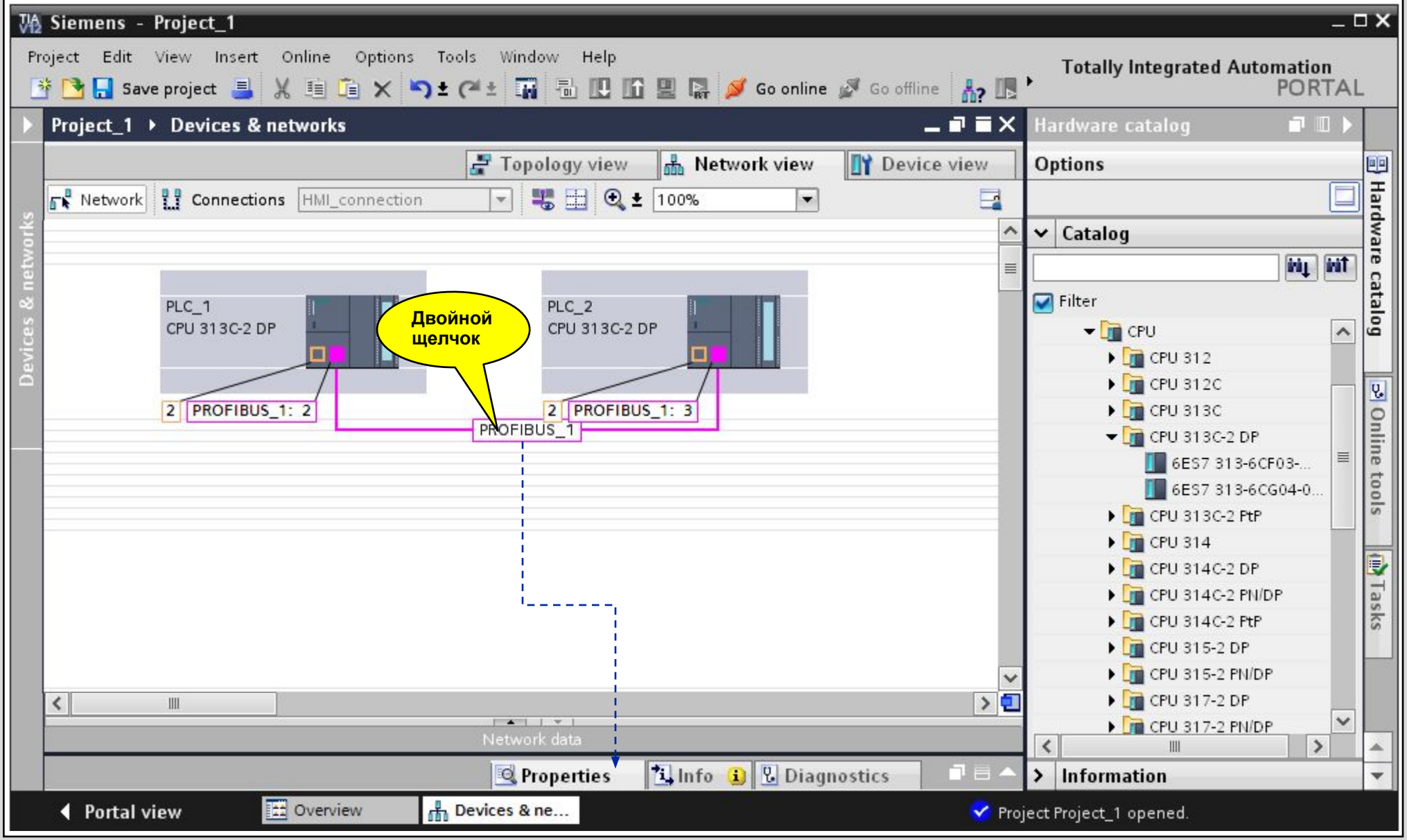
Программирование процесса передачи данных партнеру



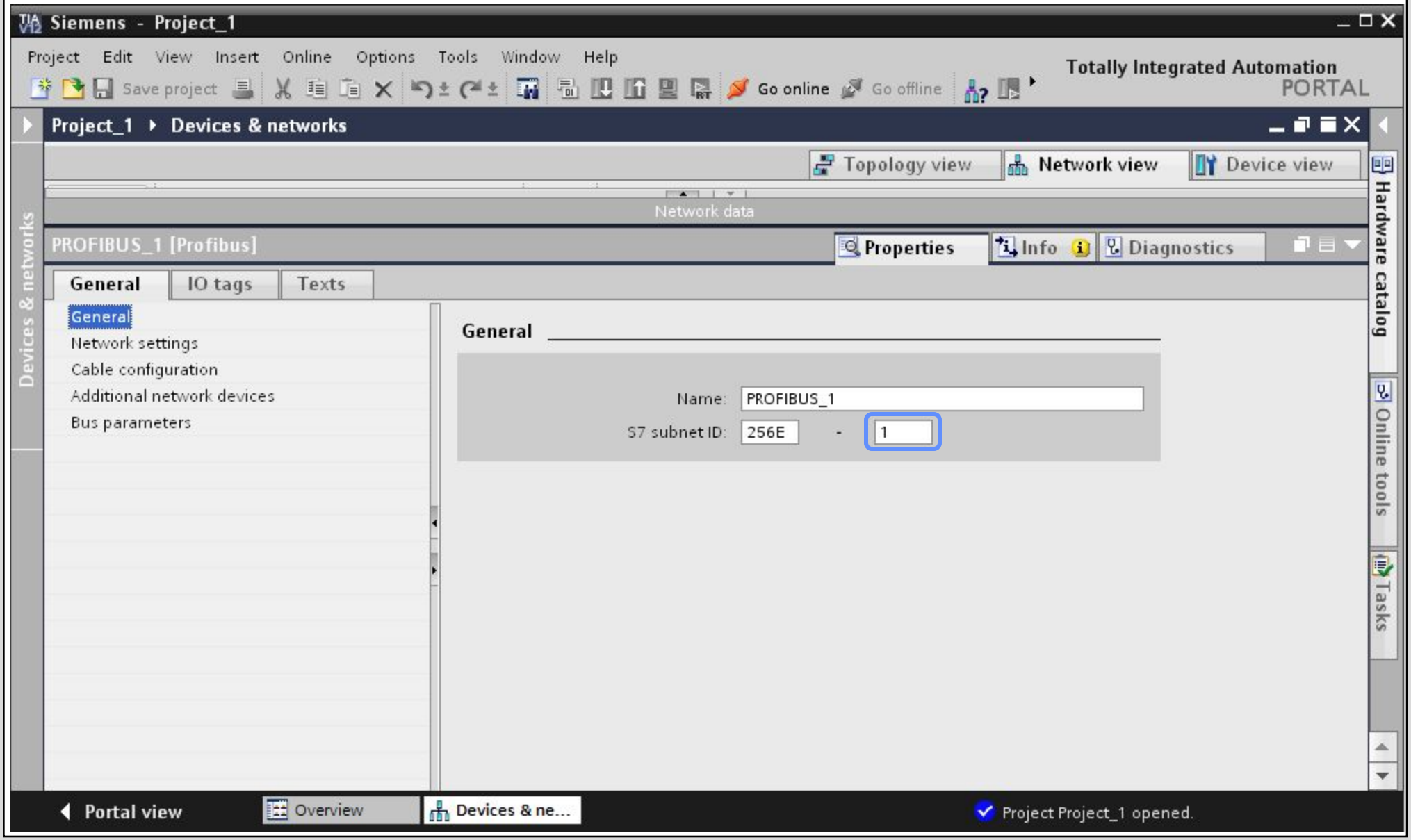
Создание сети PLC – PLC на базе PROFIBUS



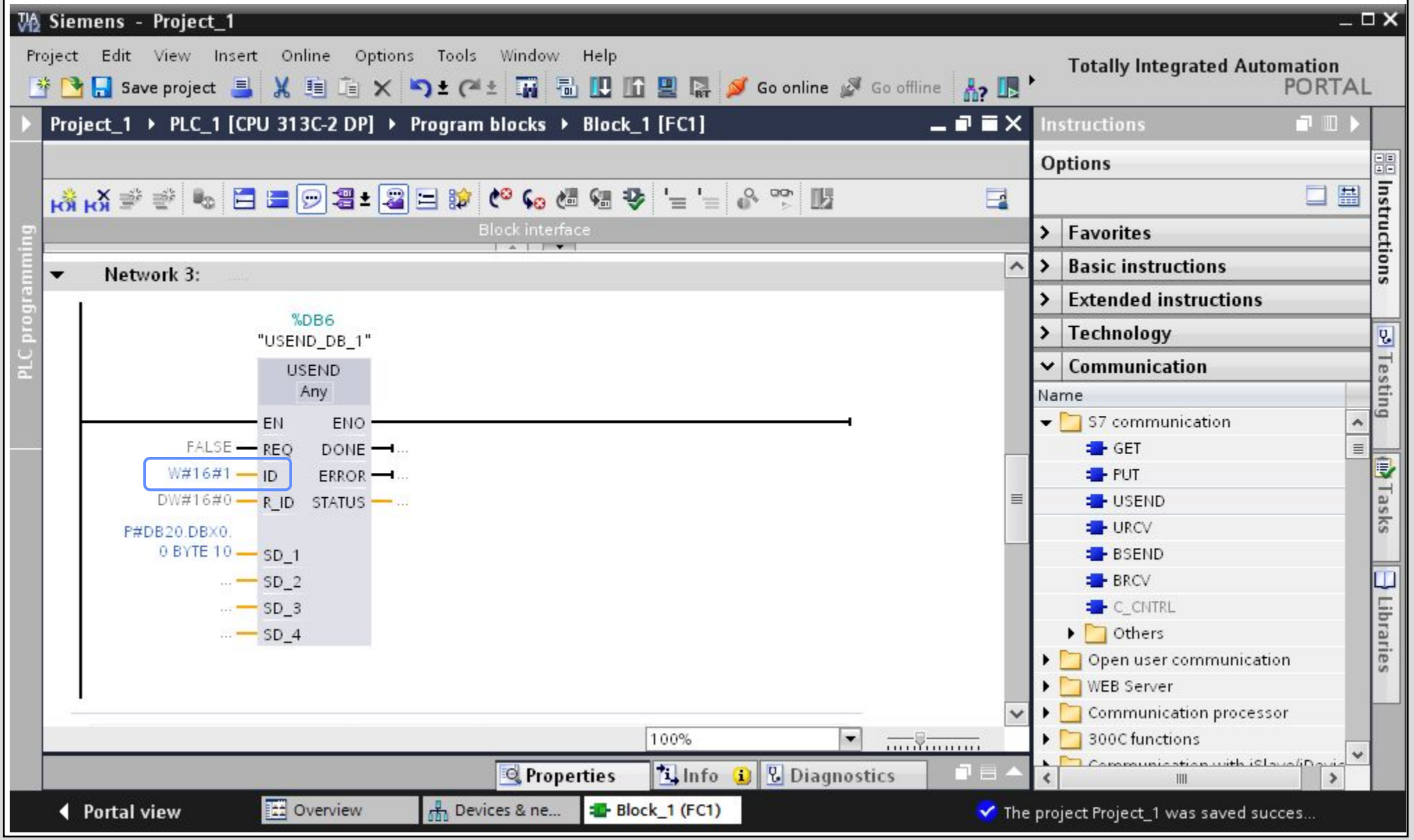
Настройки сети



Определение идентификаторов объектов сети



Программирование процесса передачи данных партнеру



Конец раздела 10. Окно навигации

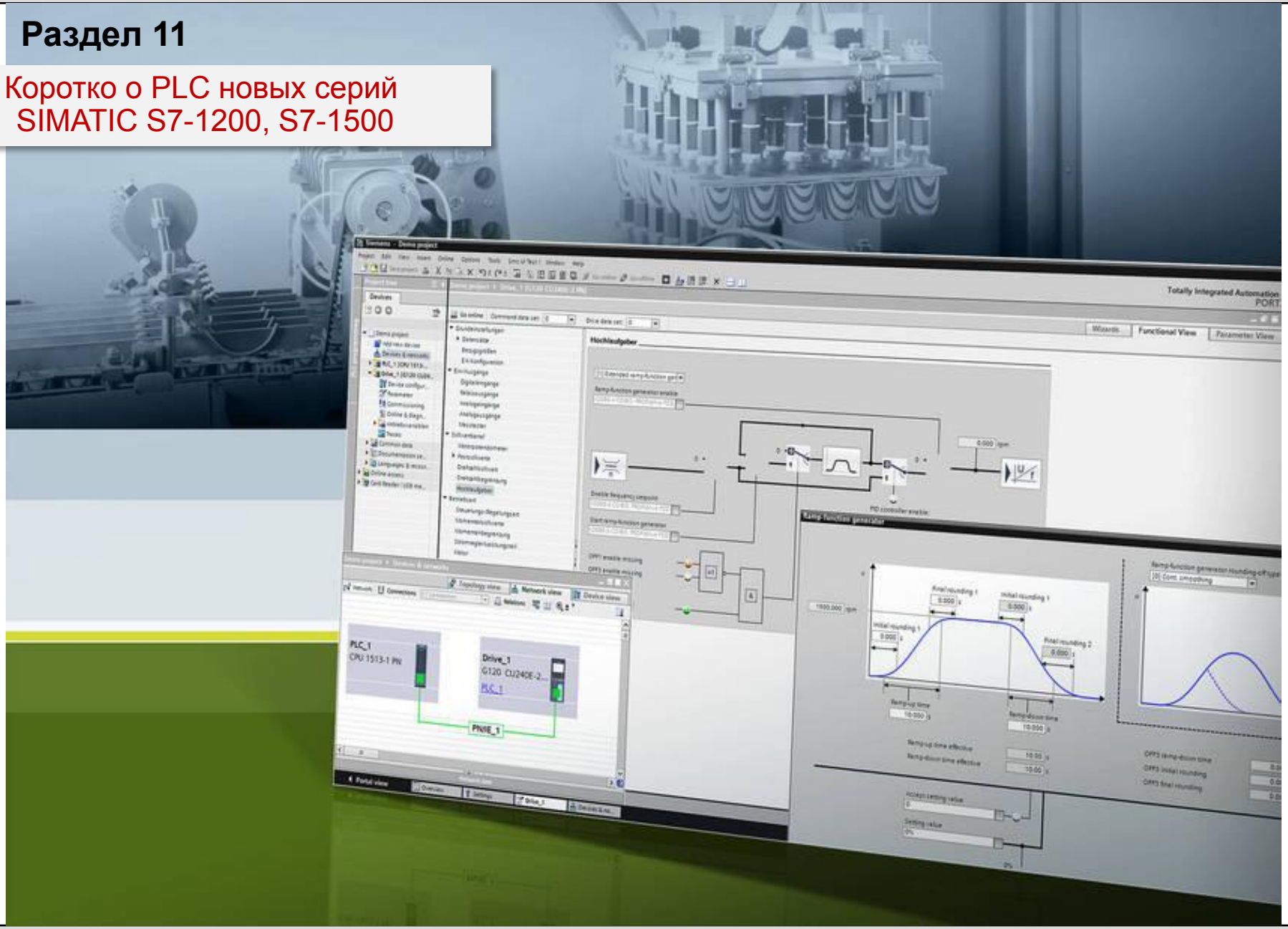
- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка

Системы с сетевой конфигурацией

- Конфигурирование ПЛК S7-1200, S7-1500

Раздел 11

Коротко о PLC новых серий SIMATIC S7-1200, S7-1500



Позиционирование новых моделей PLC SIMATIC S7

SIMATIC S7-400

SIMATIC S7-300

SIMATIC S7-1500

Модульный контроллер для систем автоматизации производств среднего и высокого уровня сложности

SIMATIC S7-1200 (Взамен S7-200)

Компактный модульный контроллер для решения базовых задач автоматизации производств и построения **автономных** систем

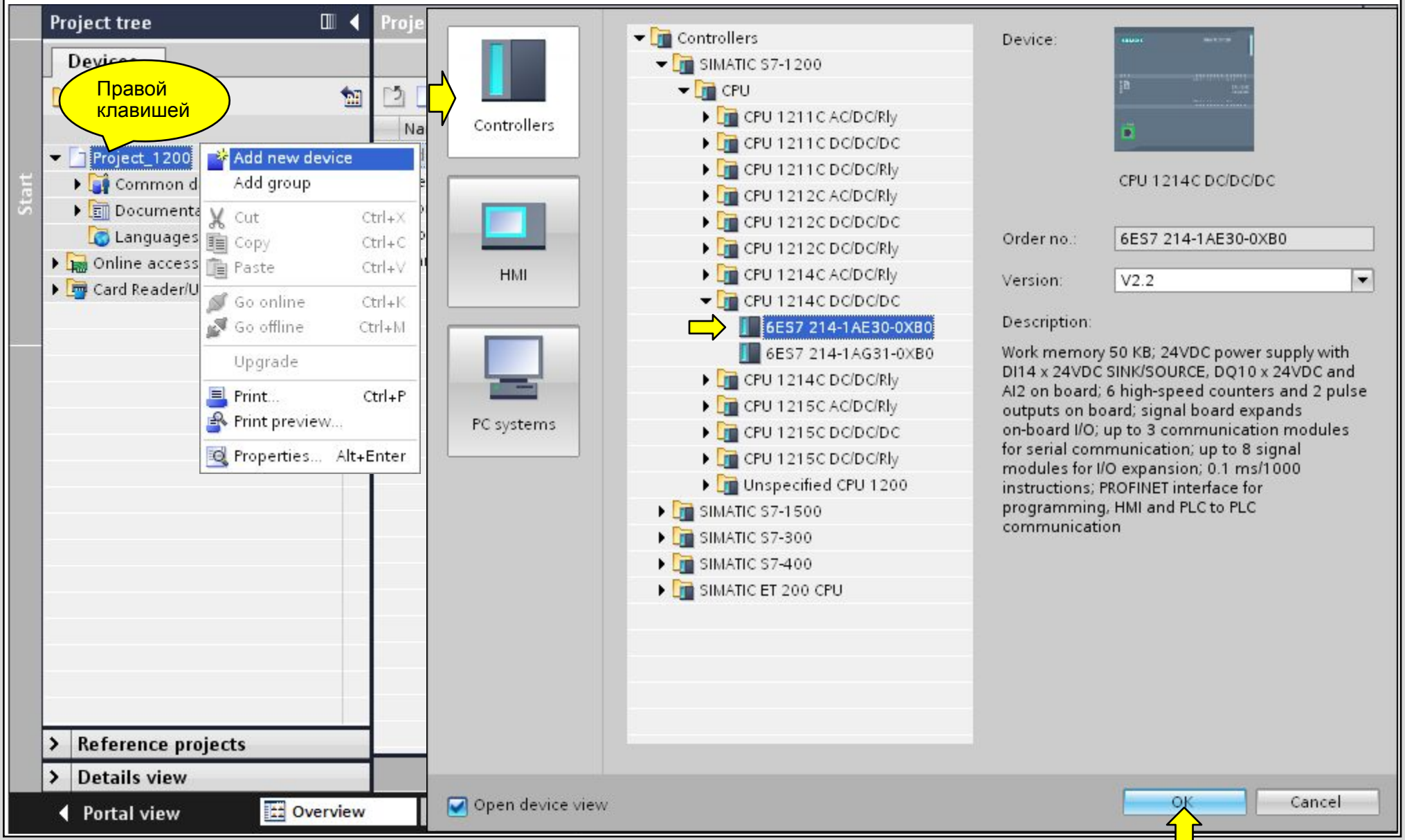
LOGO!

Логические модули для решения простейших задач автоматизации и построения **автономных** систем

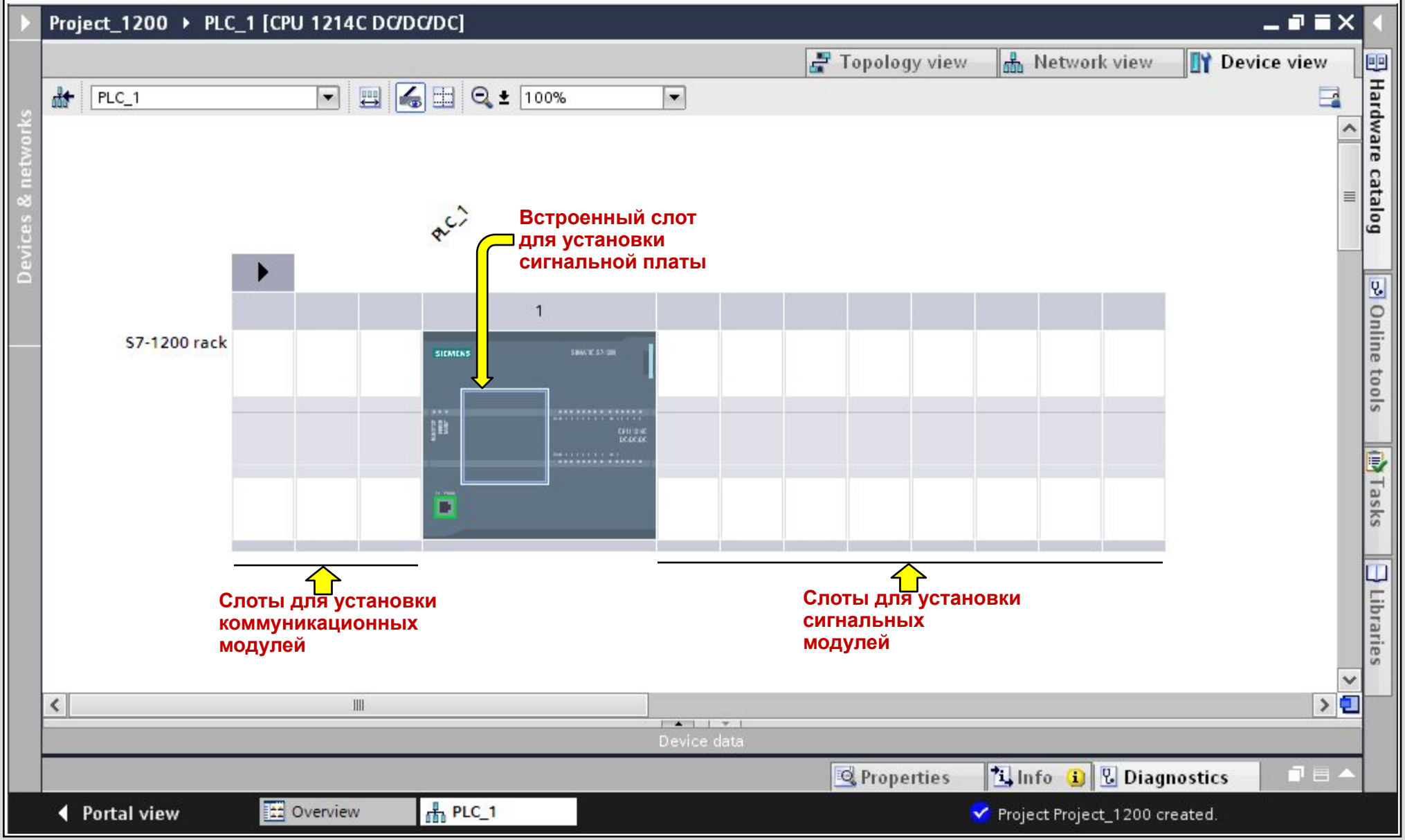
Основные характеристики CPU S7-1200

Характеристика	CPU 1211C	CPU 1212C	CPU 1214C
Физический размер (мм)	90 x 100 x 75		110 x 100 x 75
Пользовательская память • Рабочая память • Загрузочная память • Сохраняемая память	 • 25 Кбайт • 1 Мбайт • 2 Кбайта		 • 50 Кбайт • 2 Мбайта • 2 Кбайта
Локальные встроенные входы/выходы • цифровые • аналоговые	 • 6 входов/4 выхода • 2 входа	 • 8 входов/6 выходов • 2 входа	 • 14 входов/10 выходов • 2 входа
Величина образа процесса	1024 байта входов (I) и 1024 байта выходов (Q)		
Битовая память (M)	4096 байт		8192 байта
Дополнительные сигнальные модули	Нет	2	8
Сигнальная плата	1		
Коммуникационные модули	3 (левостороннее расширение)		
Скоростные счетчики • однофазные • со сдвигом фаз на 90°	3 • 3 на 100 кГц • 3 на 80 кГц	4 • 3 на 100 кГц 1 на 30 кГц • 3 на 80 кГц 1 на 20 кГц	6 • 3 на 100 кГц 3 на 30 кГц • 3 на 80 кГц 3 на 20 кГц
Импульсные выходы	2		
Карта памяти	Карта памяти SIMATIC (факультативно)		
Длительность сохранения времени для часов реального времени	Тип. 10 дней / 6 дней минимум при 40 градусах С.		
PROFINET	1 коммуникационный порт для связи с Ethernet		
Скорость выполнения арифметических операций	18 мкс/команду		
Скорость выполнения булевых операций	0,1 мкс/ команду		

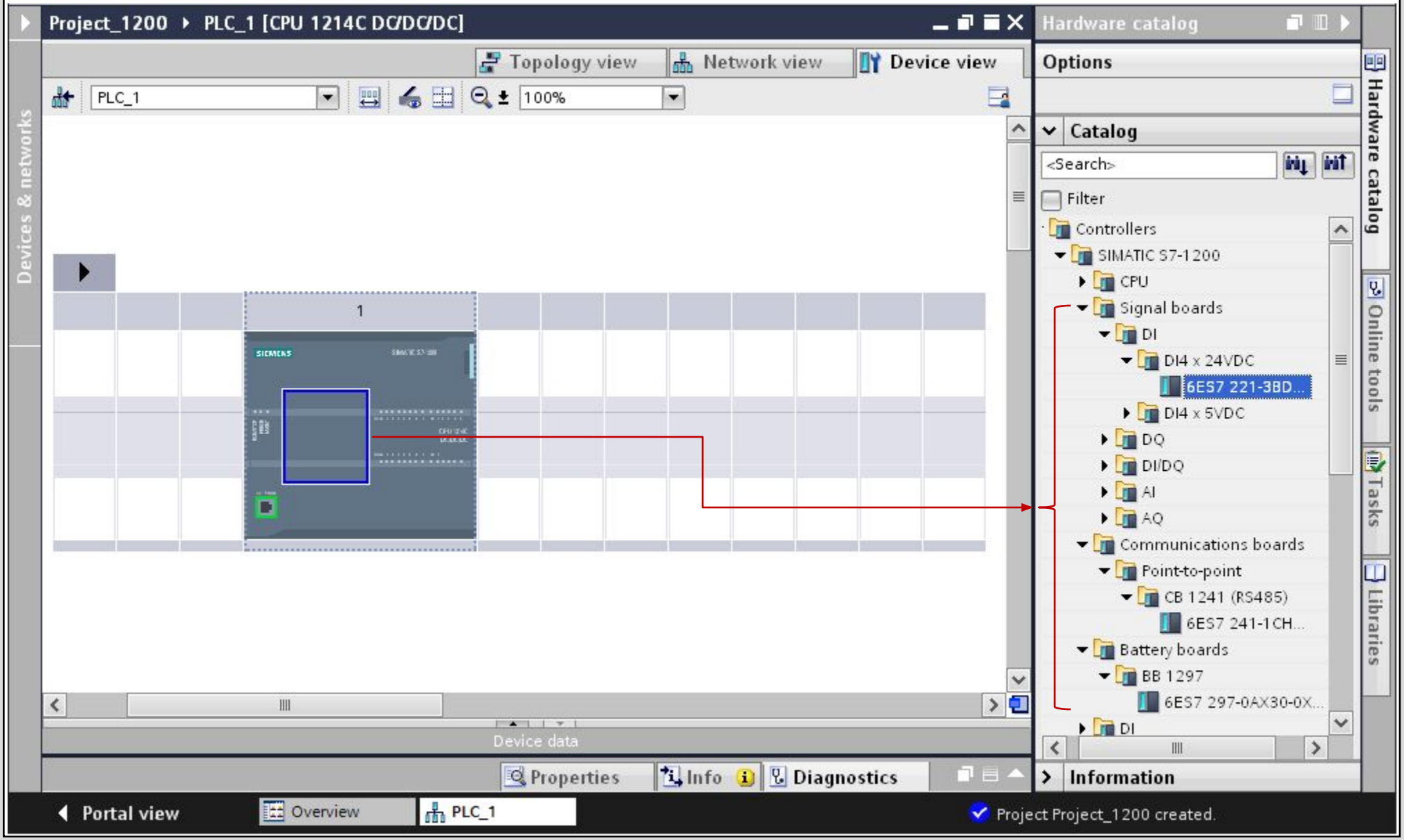
Выбор процессора S7-1200



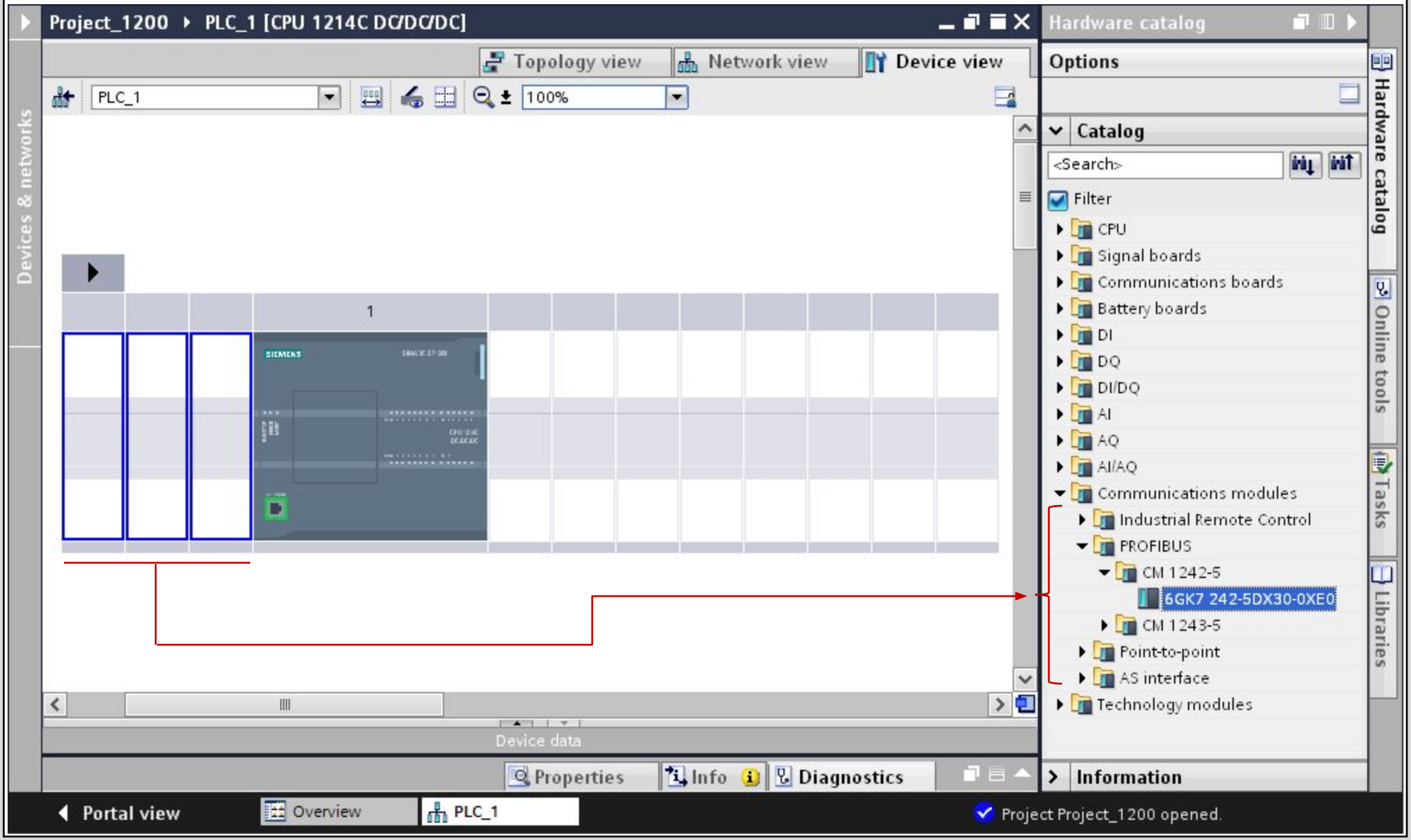
Перечень слотов для установки плат, модулей S7-1200



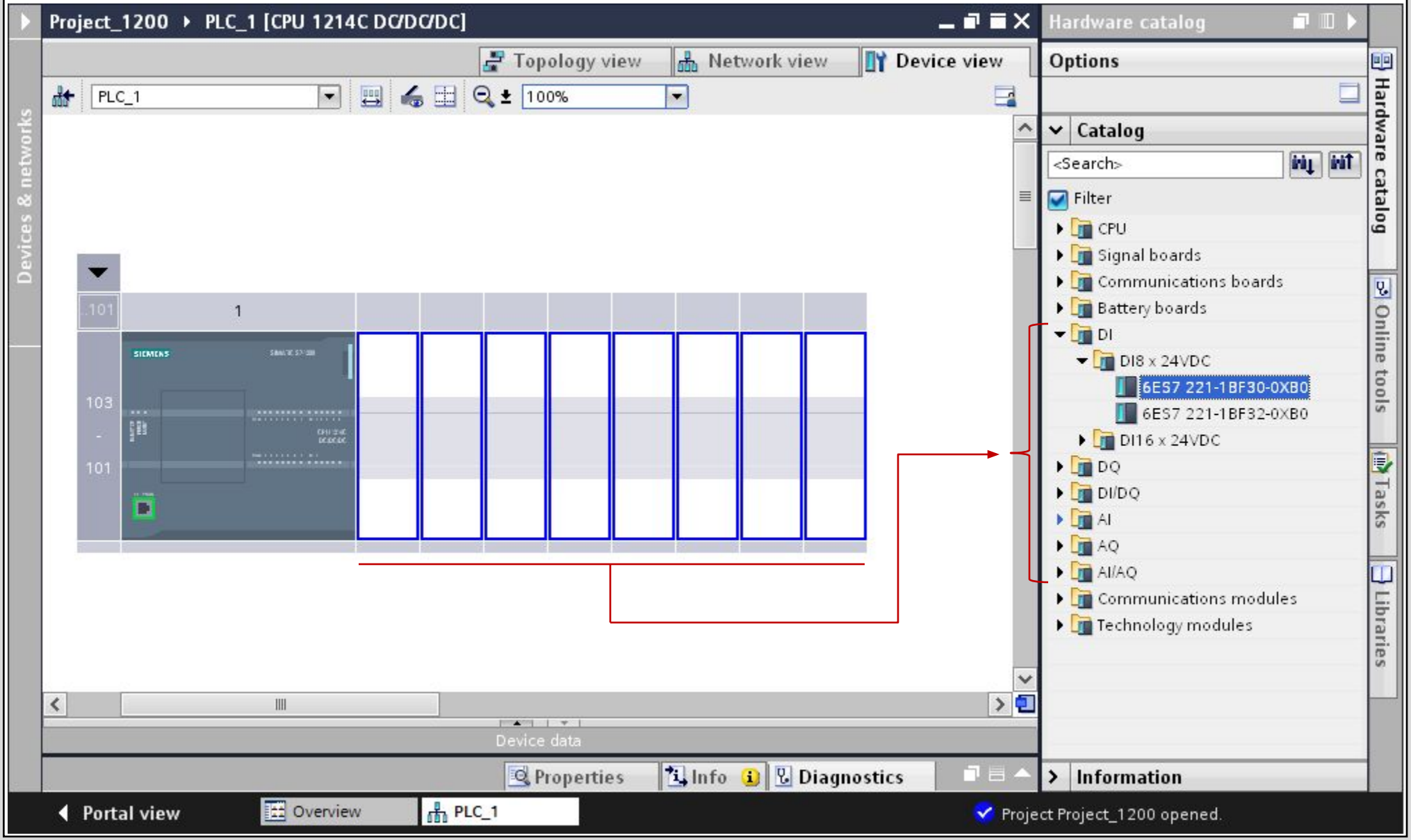
Установка сигнальной платы из каталога S7-1200



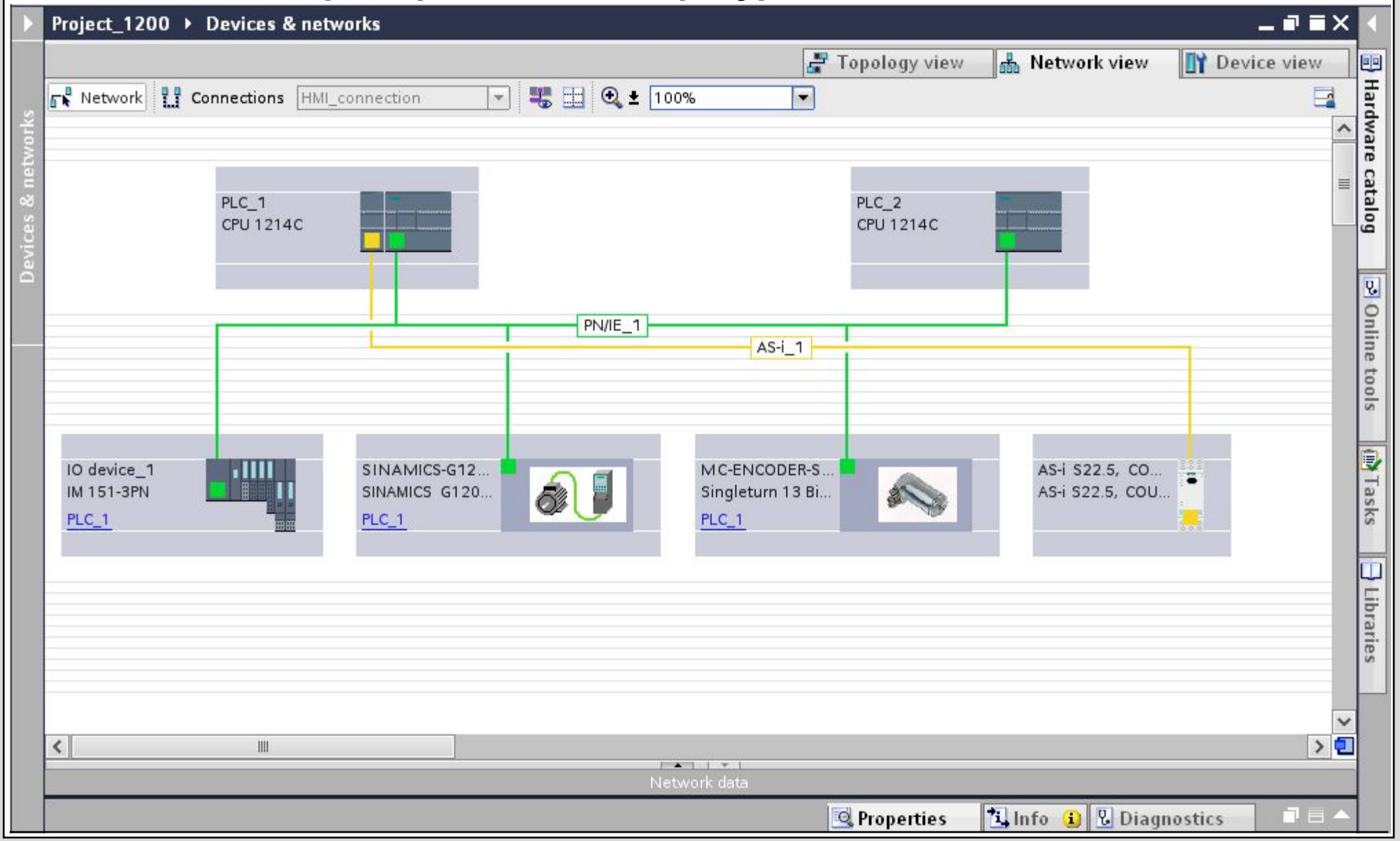
Установка коммуникационных модулей из каталога S7-1200



Установка сигнальных модулей из каталога S7-1200



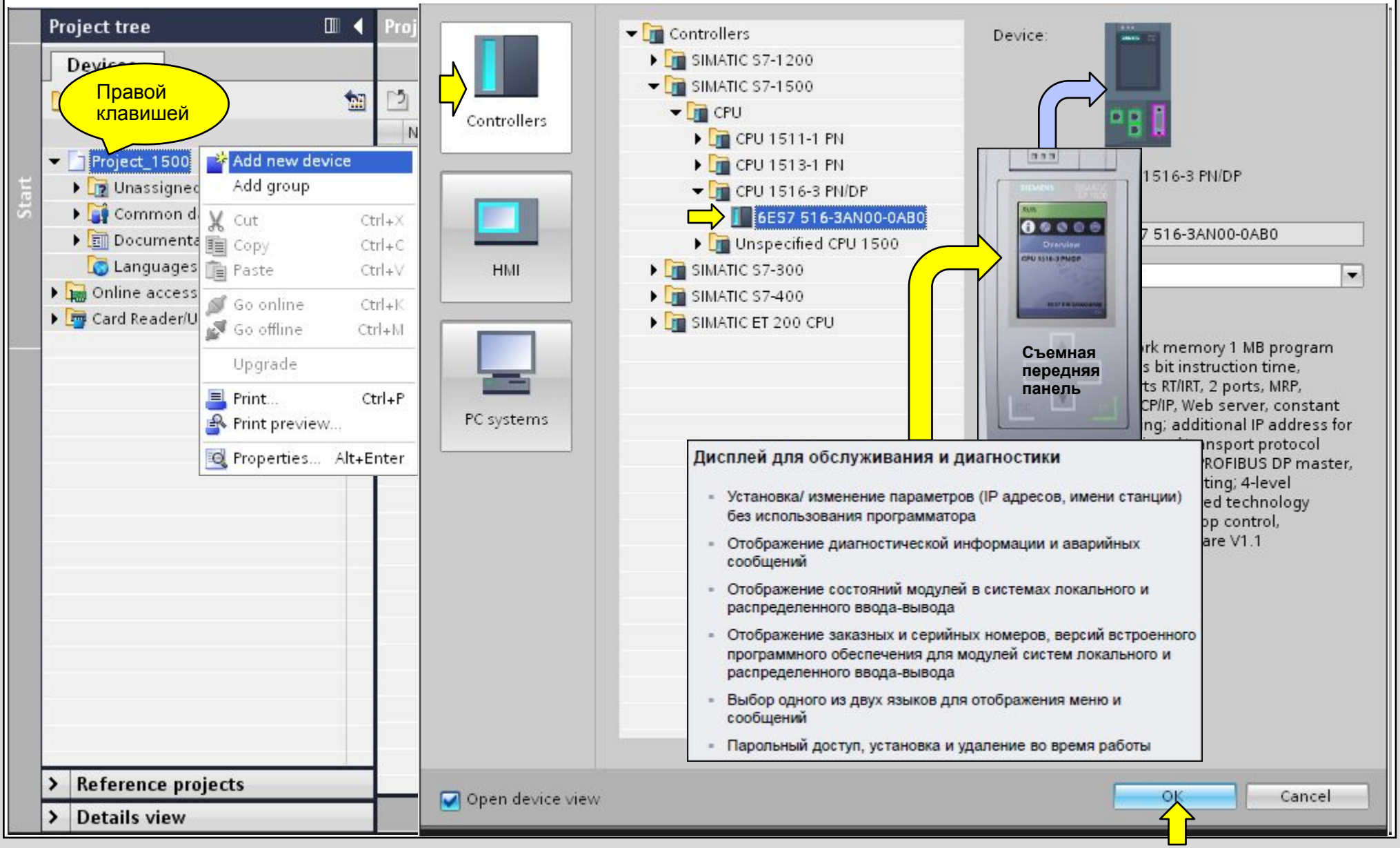
Пример сетевой конфигурации ПЛК S7-1200



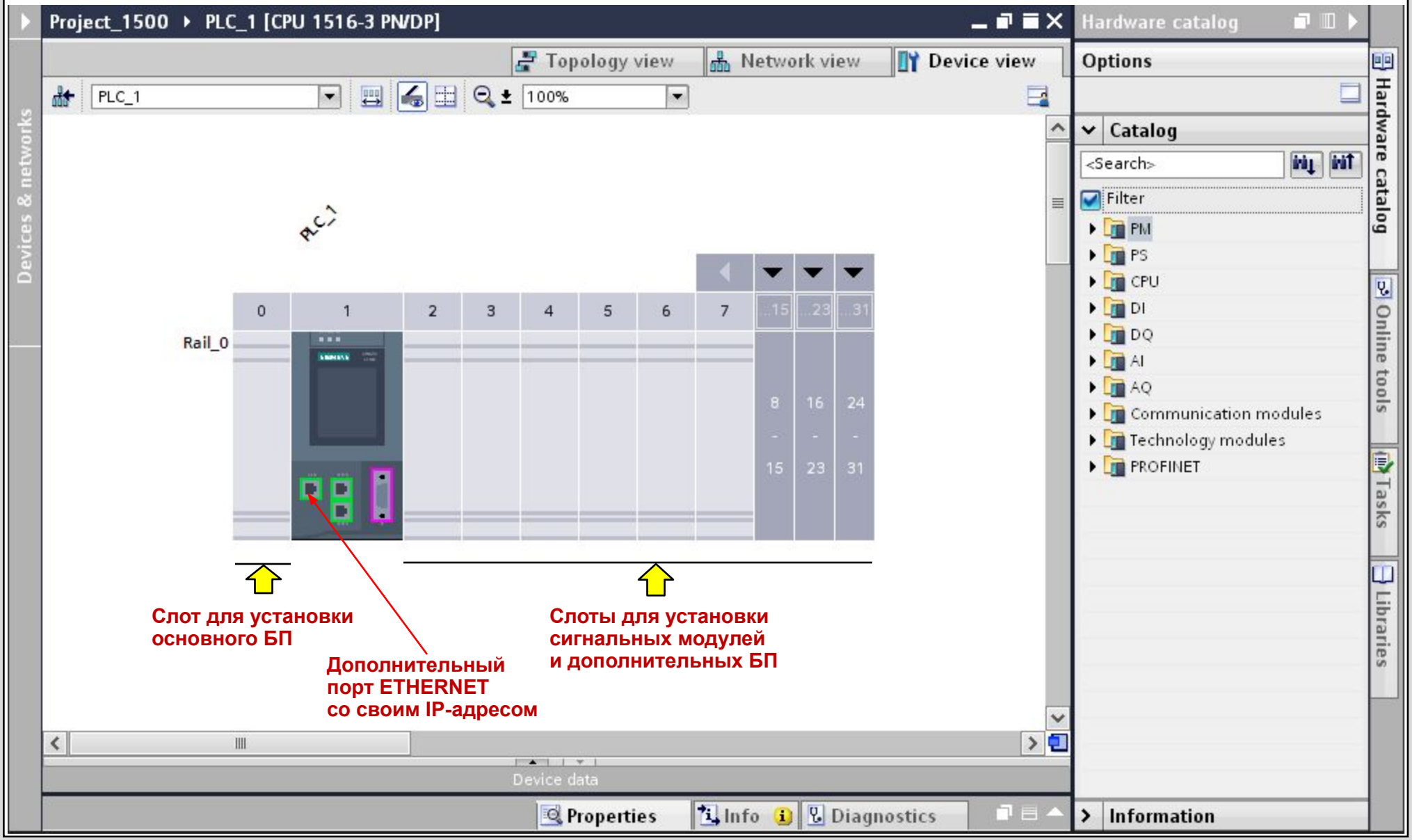
Основные характеристики CPU S7-1500

Центральный процессор 6ES7	511-1AK00-0AB0 CPU 1511-1 PN	513-1AL00-0AB0 CPU 1513-1 PN	515-2AM00-0AB0 CPU 1515-2 PN	516-3AN00-0AB0 CPU 1516-3 PN/DP	518-4AP00-0AB0 CPU 1518-4 PN/DP
Элементы управления					
Дисплей с диагональю экрана	3.45 см	3.45 см	6.1 см	6.1 см	6.1 см
Количество клавиш	6	6	6	6	6
Переключатель режимов работы	1	1	1	1	1
Параметры аппаратной конфигурации					
Количество модулей на стойку, не более	32: центральный процессор + 31 модуль				
Количество базовых стоек, не более	1	1	1	1	1
Количество ведущих DP устройств на систему, не более:					
• встроенных	Нет	Нет	Нет	1	1
• в виде коммуникационных модулей	4: суммарно не более 4 CM/CP	6: суммарно не более 6 CM/CP	8: суммарно не более 8 CM/CP	8: суммарно не более 8 CM/CP	8: суммарно не более 8 CM/CP
	для обмена данными через PROFIBUS, PROFINET, Ethernet				
Счетчики и таймеры					
S7 счетчики:					
• количество	2048	2048	2048	2048	2048
– с сохранением состояний при перебоях в питании контроллера	Настраивается	Настраивается	Настраивается	Настраивается	Настраивается
IEC счетчики:					
• количество	Ограничено только размером рабочей памяти данных				
– с сохранением состояний при перебоях в питании контроллера	Настраивается	Настраивается	Настраивается	Настраивается	Настраивается
S7 таймеры:					
• количество	2048	2048	2048	2048	2048
– с сохранением состояний при перебоях в питании контроллера	Настраивается	Настраивается	Настраивается	Настраивается	Настраивается
IEC таймеры:					
• количество	Ограничено только размером рабочей памяти данных				
– с сохранением состояний при перебоях в питании контроллера	Настраивается	Настраивается	Настраивается	Настраивается	Настраивается

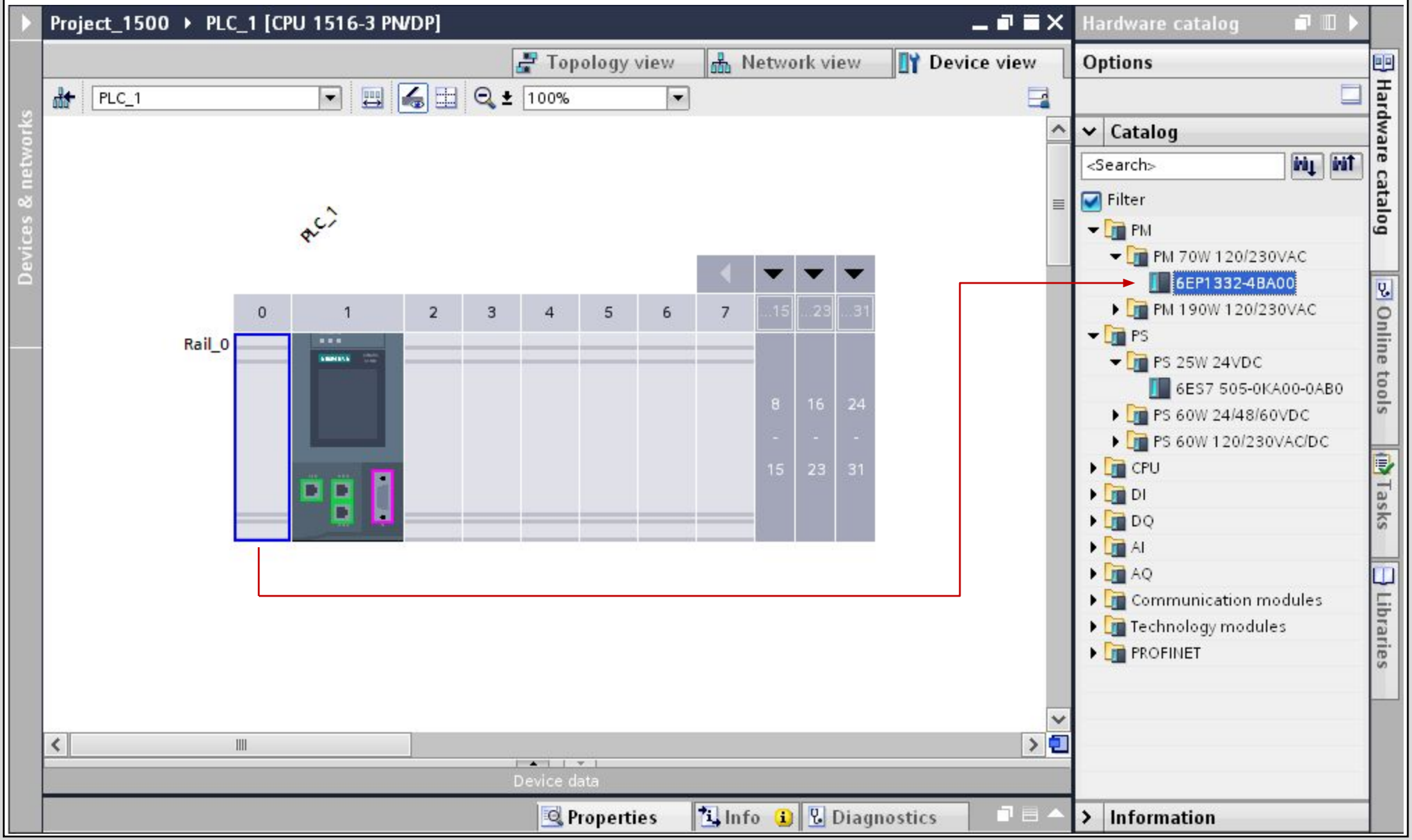
Выбор процессора S7-1500. Функции передней панели.



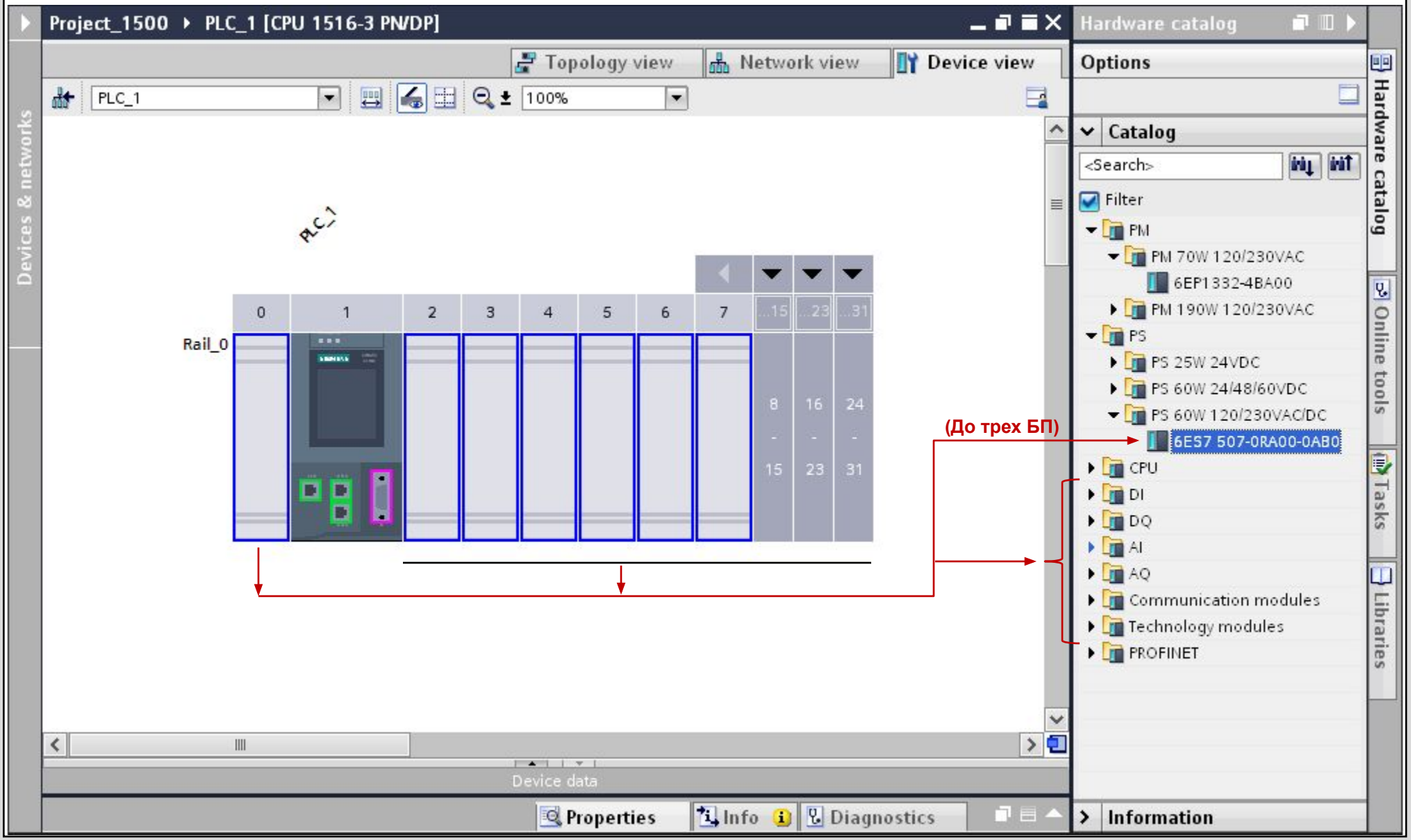
Перечень слотов для установки модулей S7-1500



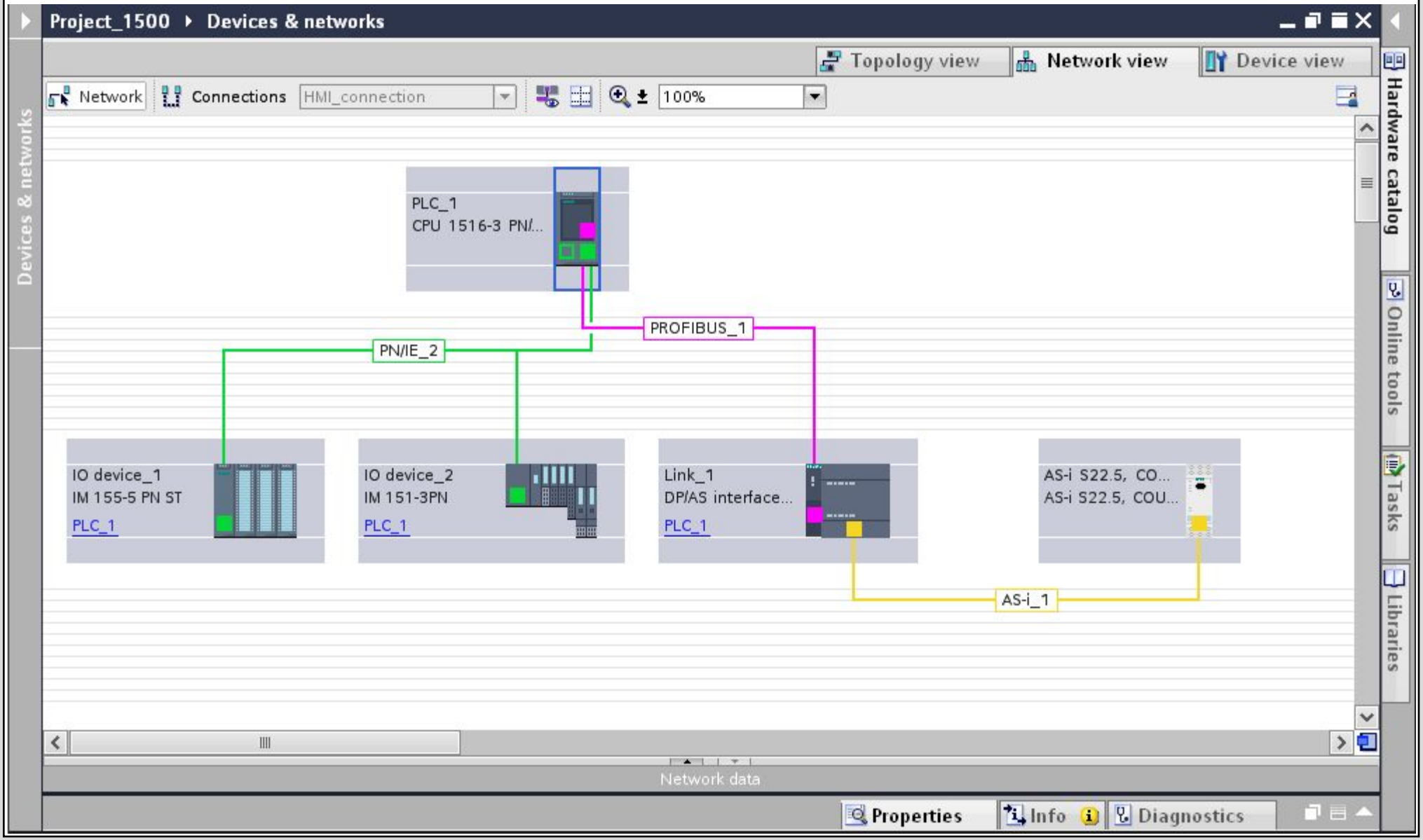
Установка основного блока питания в конфигурации S7-1500



Сигнальные модули и доп.блока питания в конфигурации S7-1500

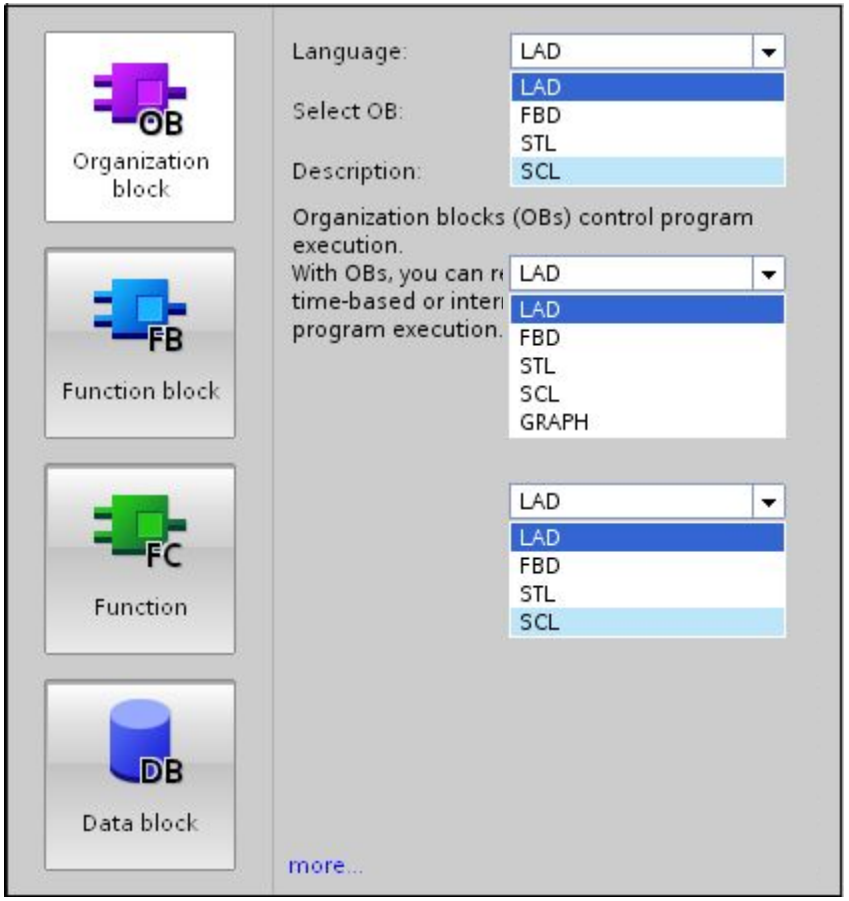


Пример сетевой конфигурации ПЛК S7-1500

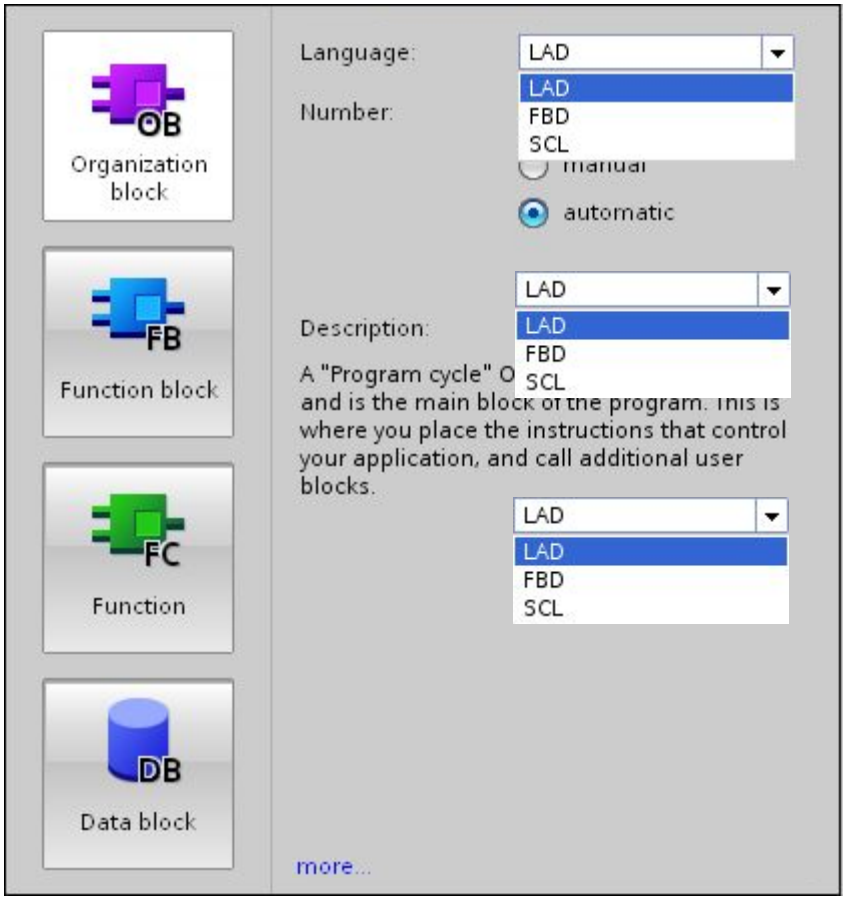


Сравнительный перечень доступных языков программирования

S7-300, S7-400, S7-1500



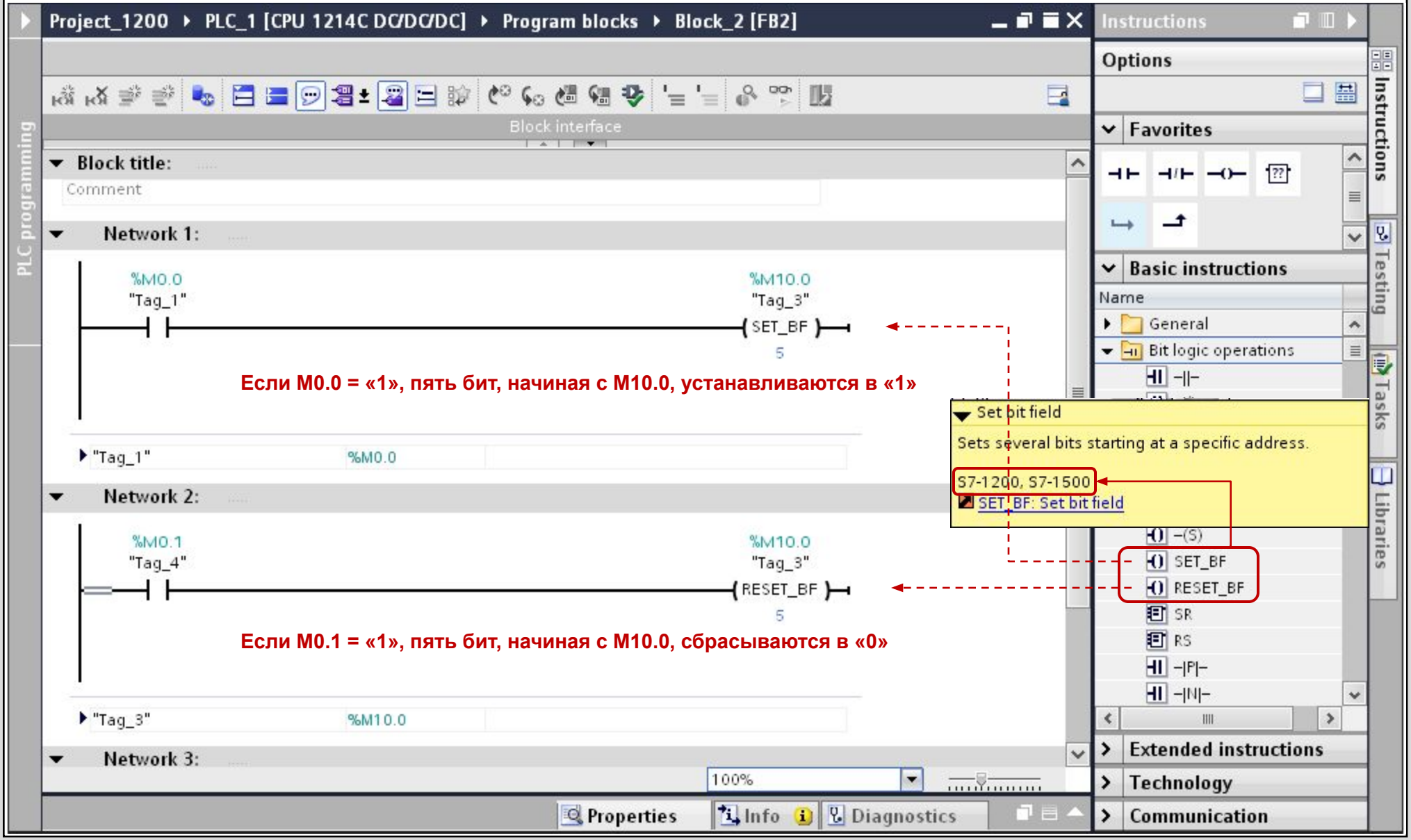
S7-1200



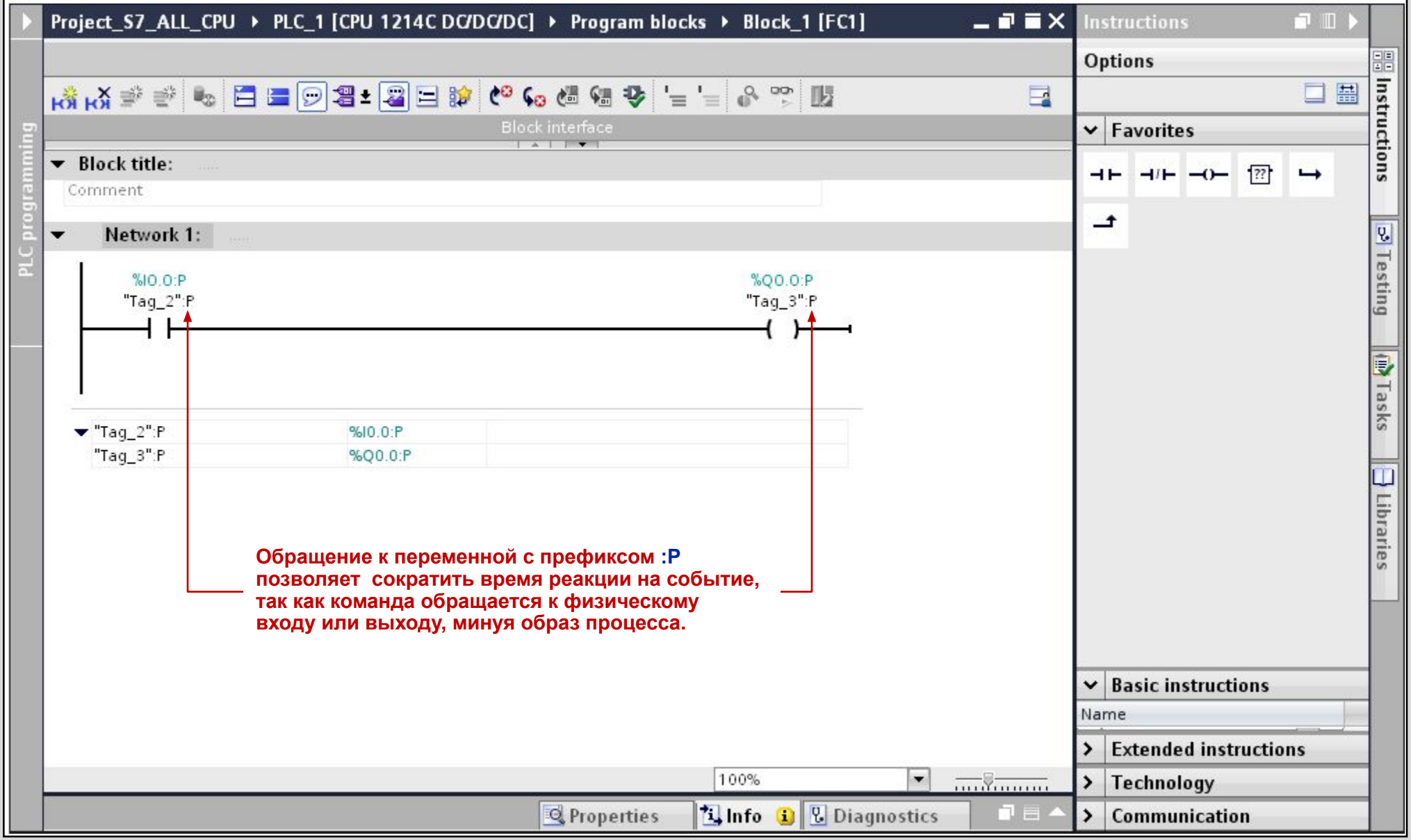
Типы данных, применяемые в S7-1200, S7-1500

Тип данных	Размер (в битах)	Диапазон	Примеры ввода констант
Bool	1	от 0 до 1	TRUE, FALSE, 0, 1
Byte	8	от 16#00 до 16#FF	16#12, 16#AB
Word	16	от 16#0000 до 16#FFFF	16#ABCD, 16#0001
DWord	32	от 16#00000000 до 16#FFFFFFFF	16#02468ACE
Char	8	от 16#00 до 16#FF	'A', 't', '@'
Sint	8	от -128 до 127	123, -123
Int	16	от -32768 до 32767	123, -123
Dint	32	от -2147483648 до 2147483647	123, -123
USInt	8	от 0 до 255	123
UInt	16	от 0 до 65 535	123
UDInt	32	0 до 4294967295	123
Real	32	от +/-1,18 x 10^{-38} до +/-3,40 x 10^{38}	123.456, -3.4, -1.2E+12, 3.4E-3
LReal	64	от +/-2,23 x 10^{-308} до +/-1,79 x 10^{308}	12345.123456789 -1.2E+40
Time	32	от T#-24d_20h_31m_23s_648ms до T#24d_20h_31m_23s_647ms Хранится как: от -2,147,483,648 мс до +2,147,483,647 мс	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	переменный	от 0 до 254 символов в размере байта	'ABC'

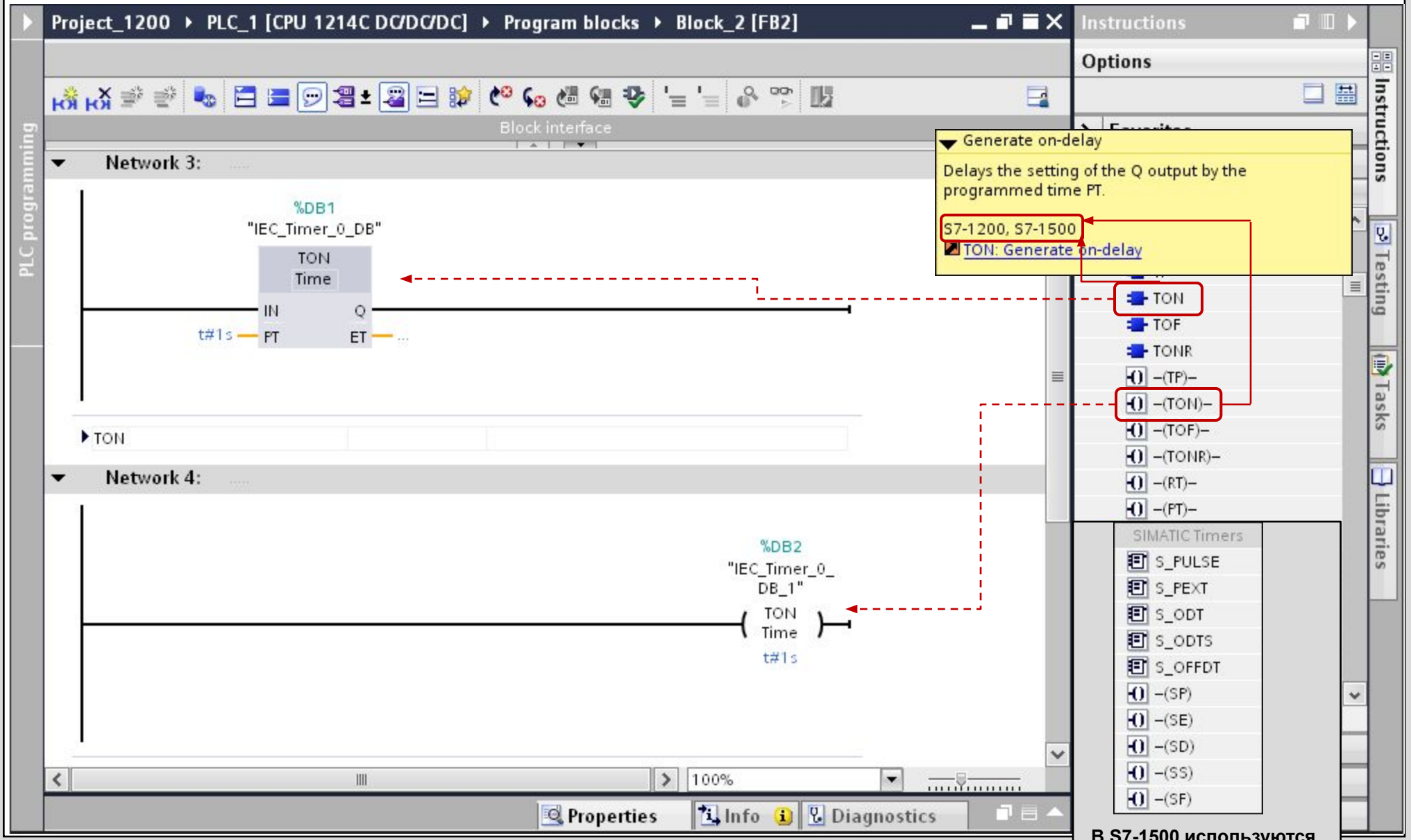
Установка и сброс битового поля в S7-1200, S7-1500



Обращение в программе к физическим входам-выходам

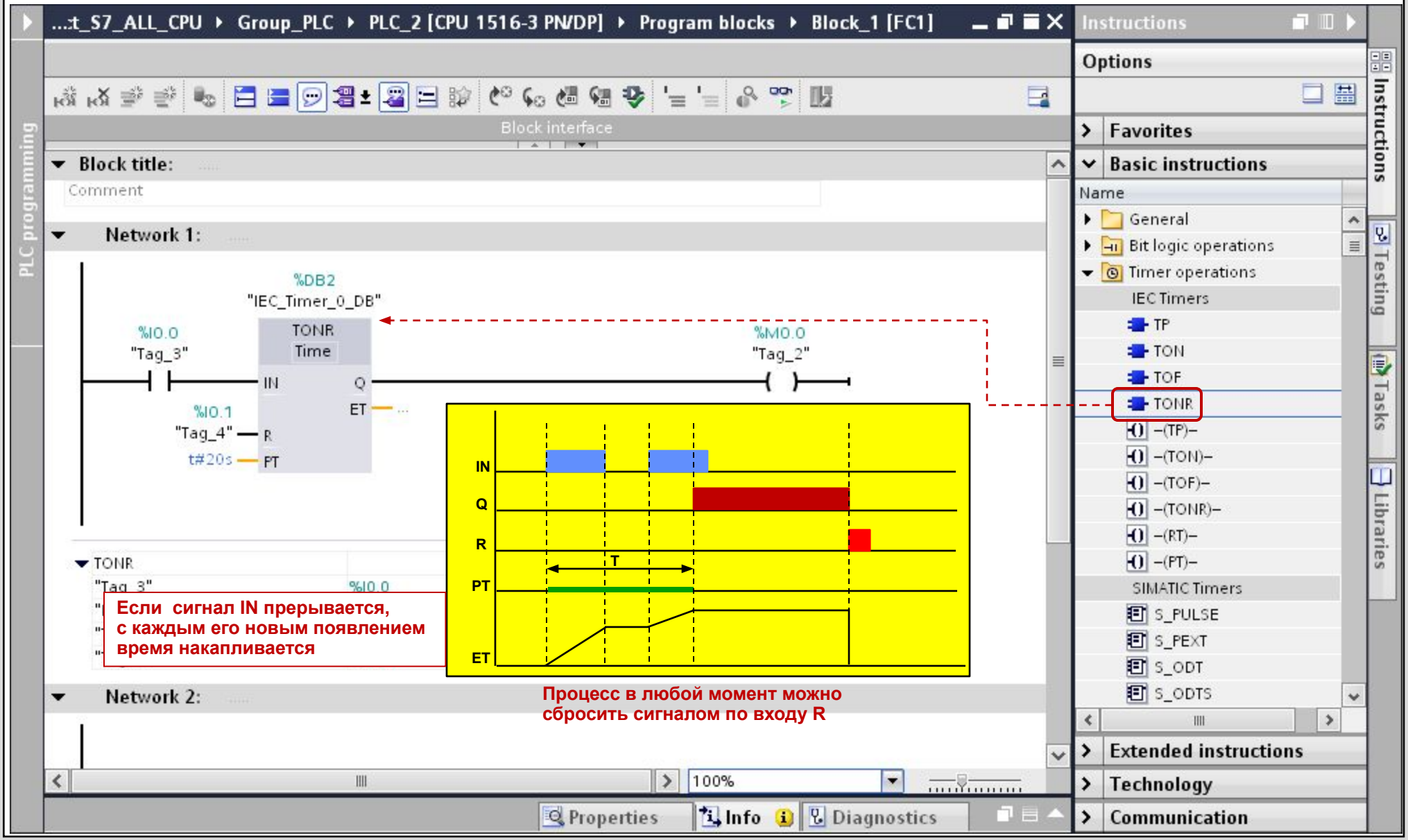


Таймеры в системе команд S7-1200, S7-1500

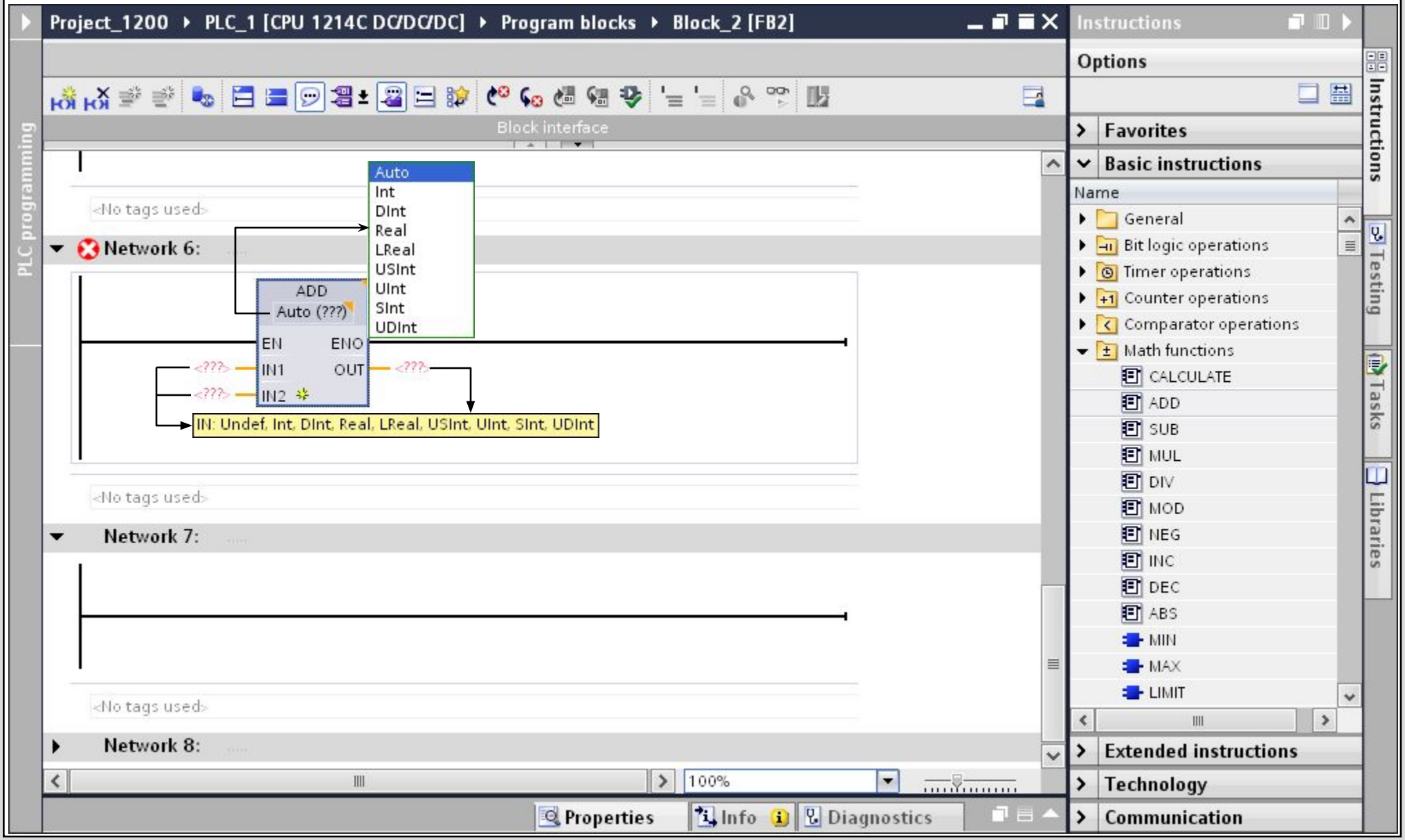


В S7-1500 используются также таймеры SIMATIC

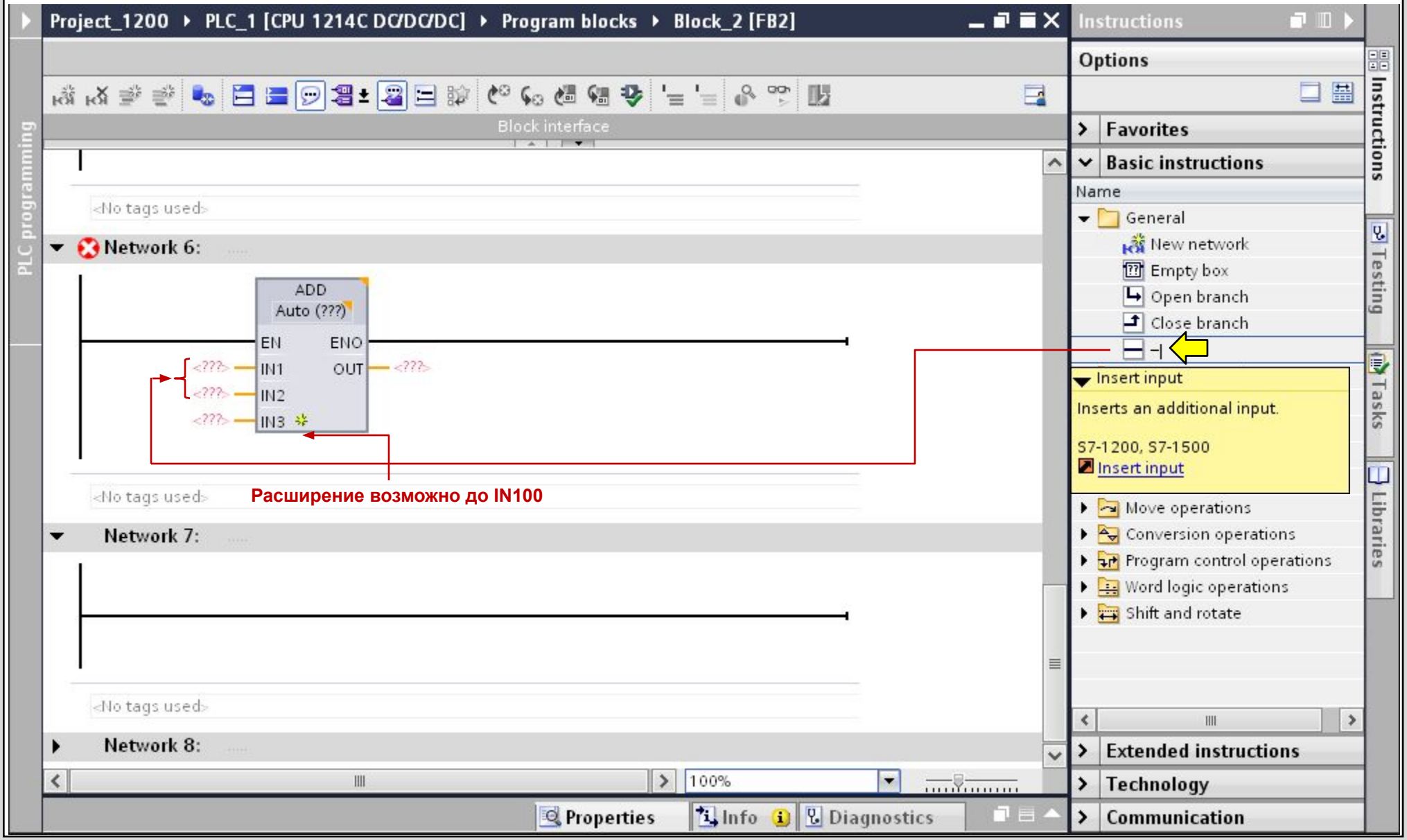
Таймер-аккумулятор



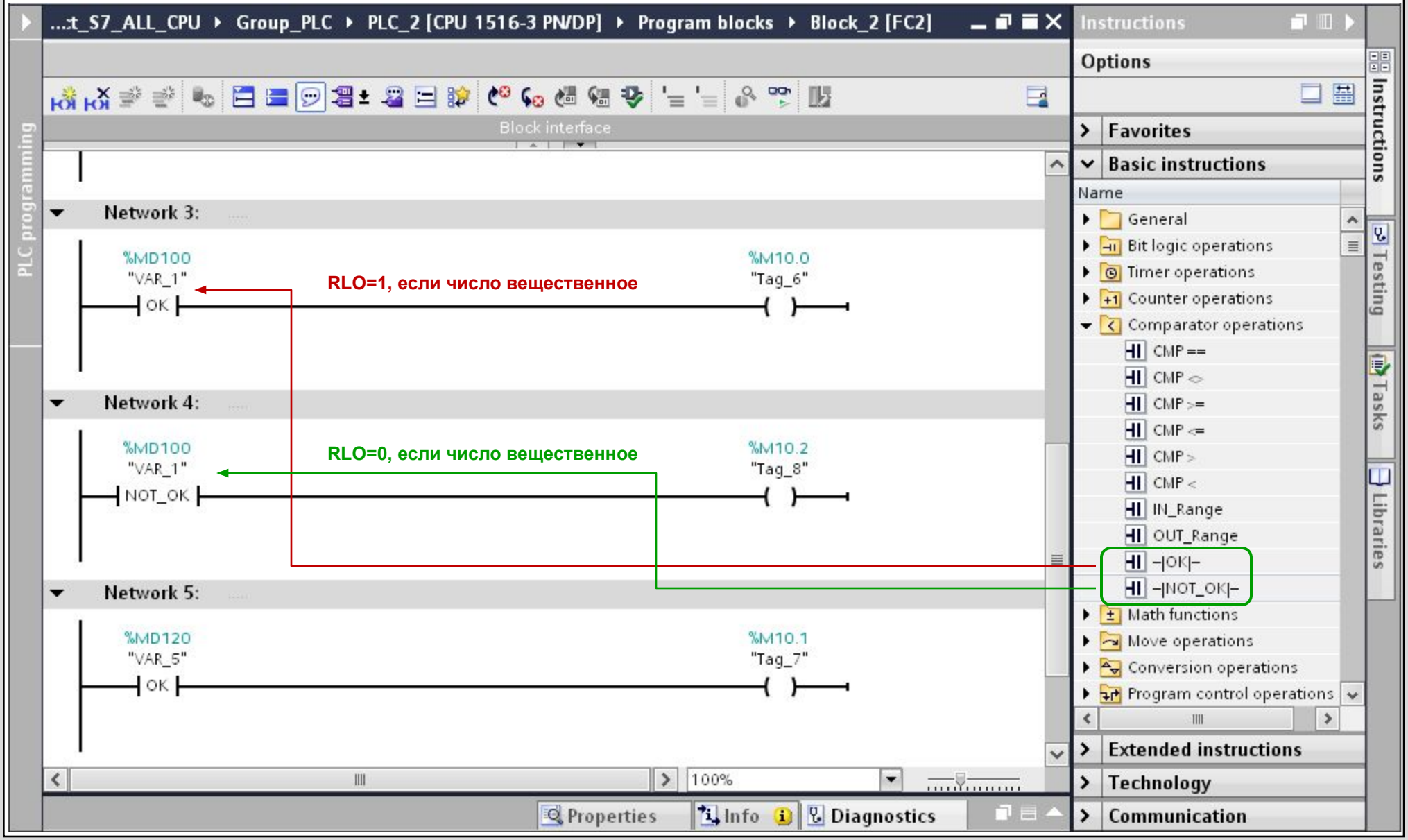
Определение типов переменных в инструкциях



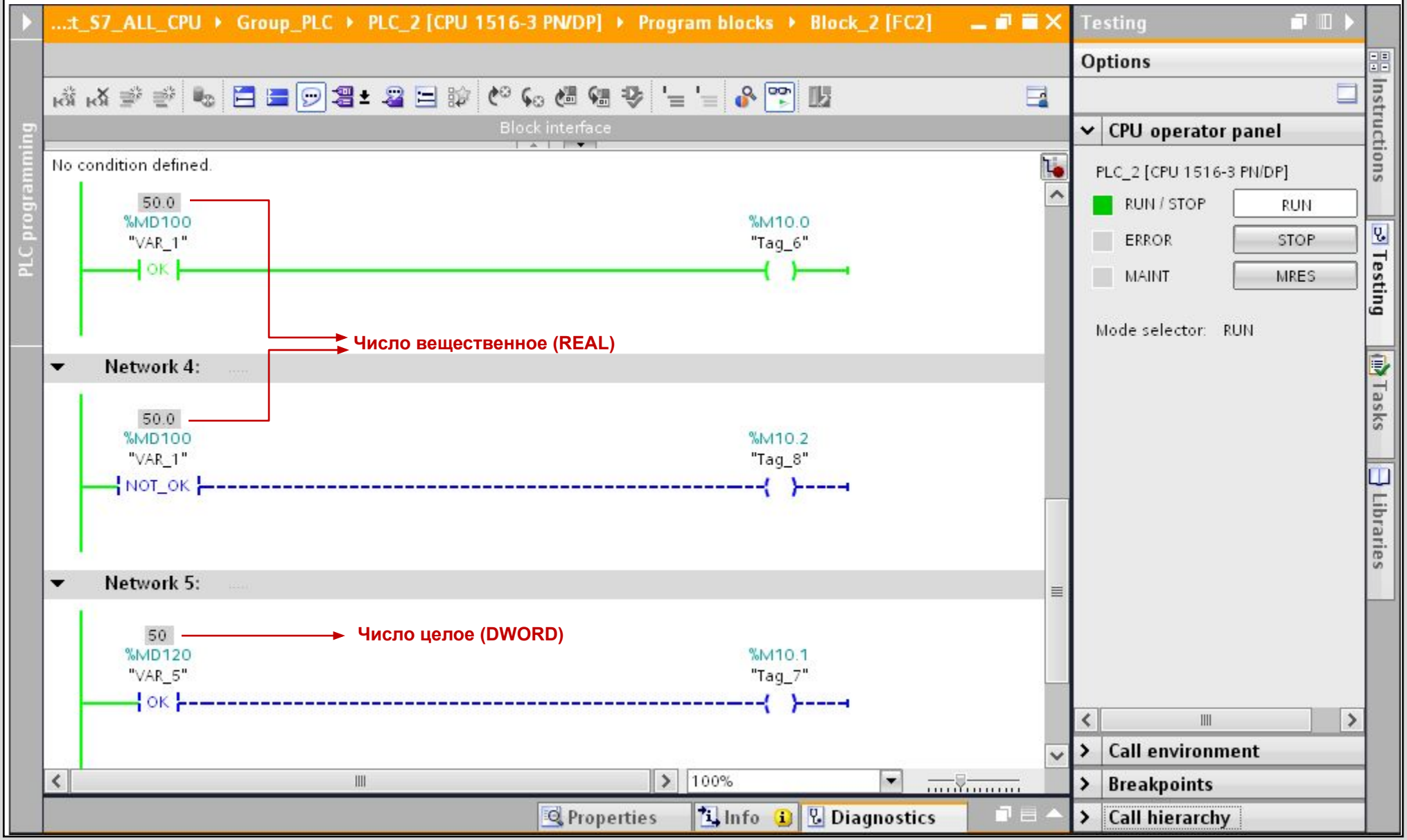
Расширение зоны входов/выходов в ряде инструкций



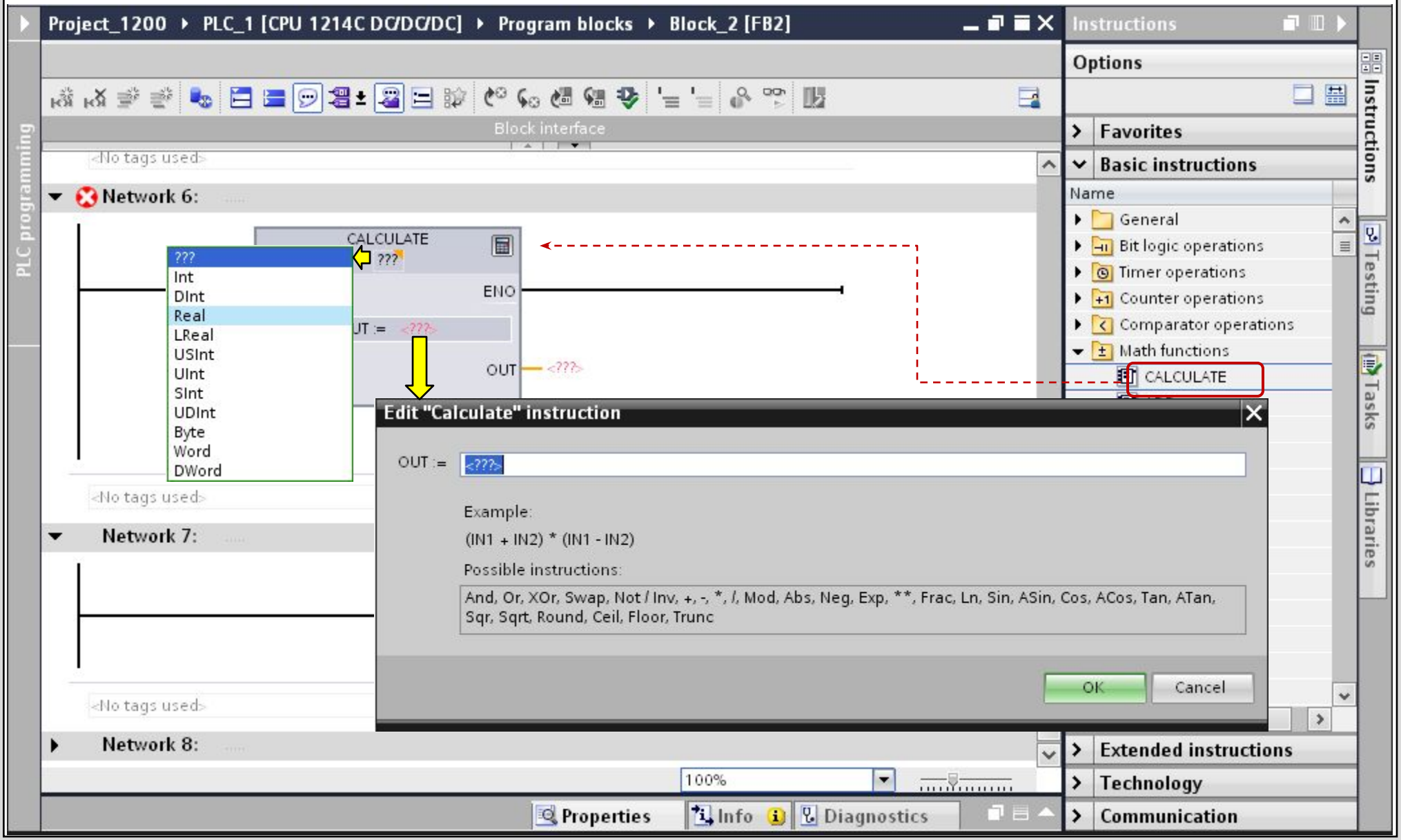
Команды проверки, является ли число вещественным



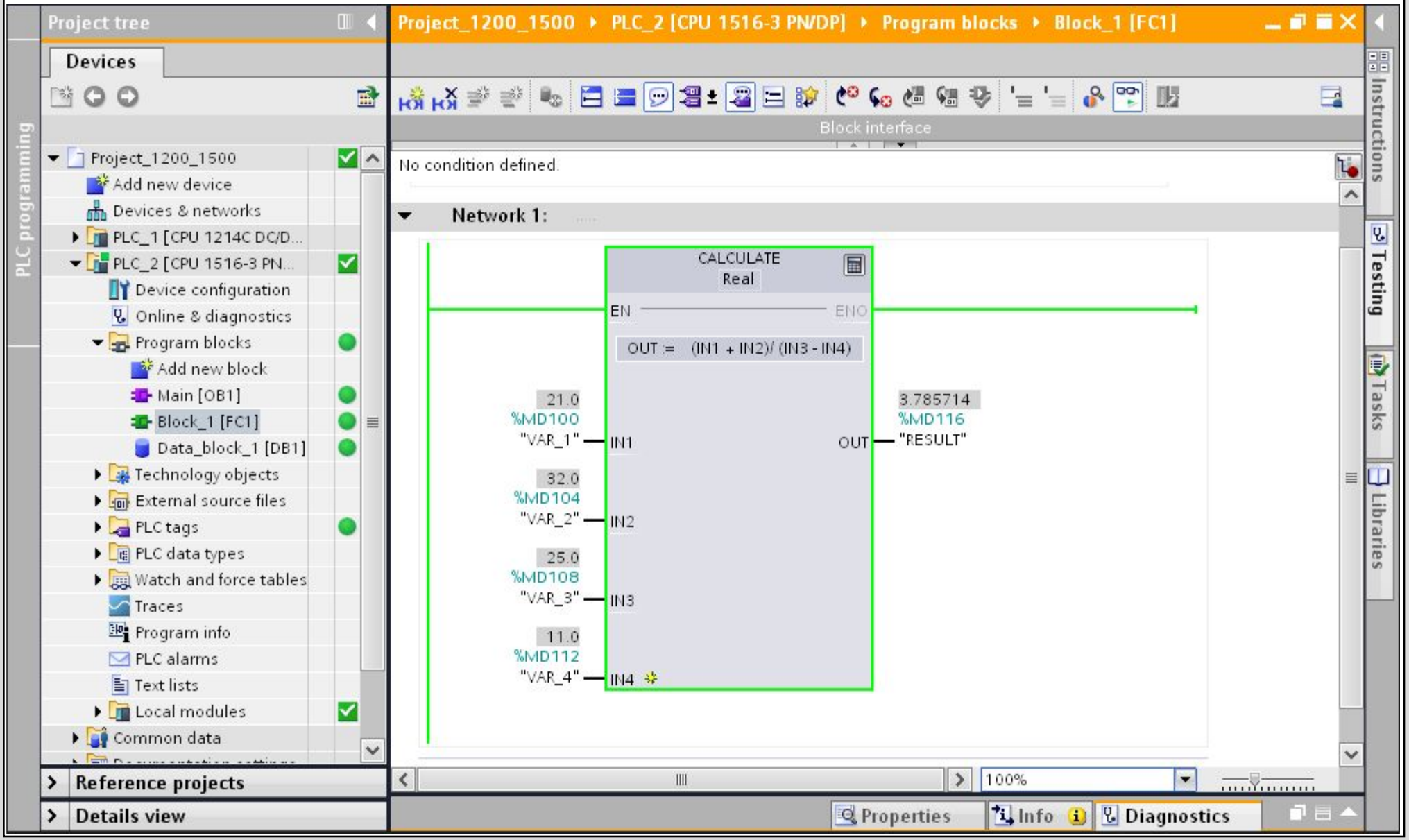
Команды проверки, является ли число вещественным



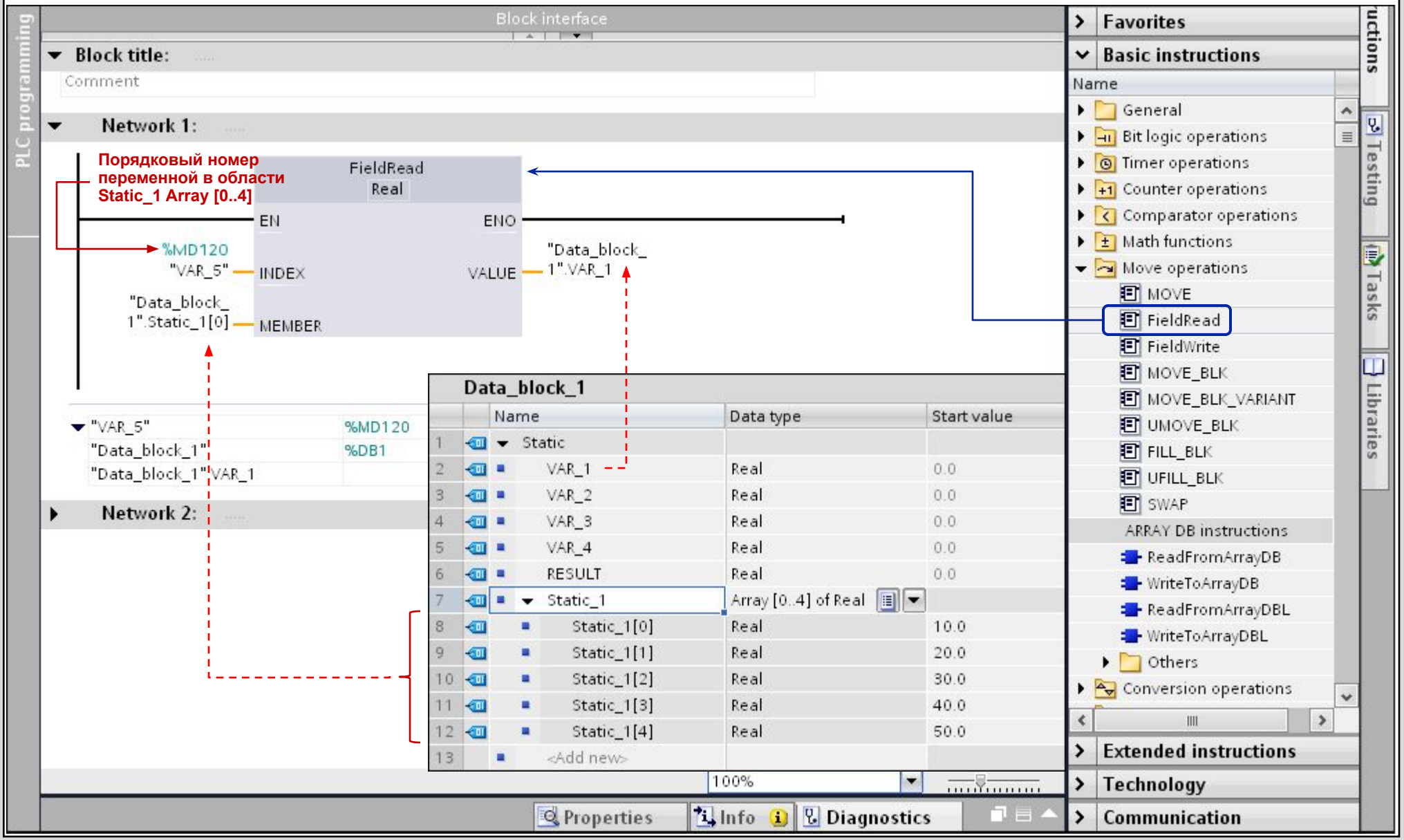
Инструкция CALCULATE



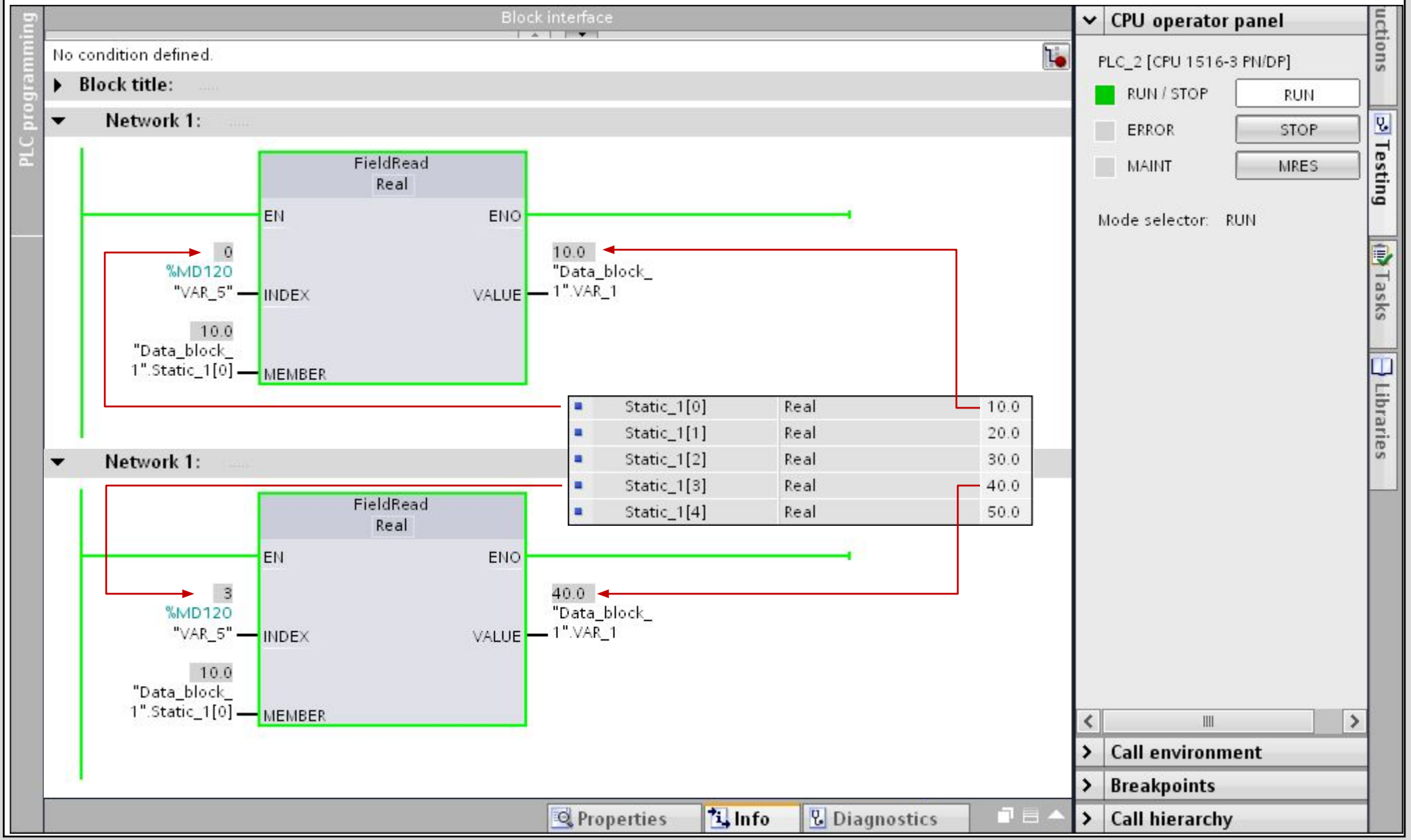
Пример работы инструкции CALCULATE в режиме On-Line



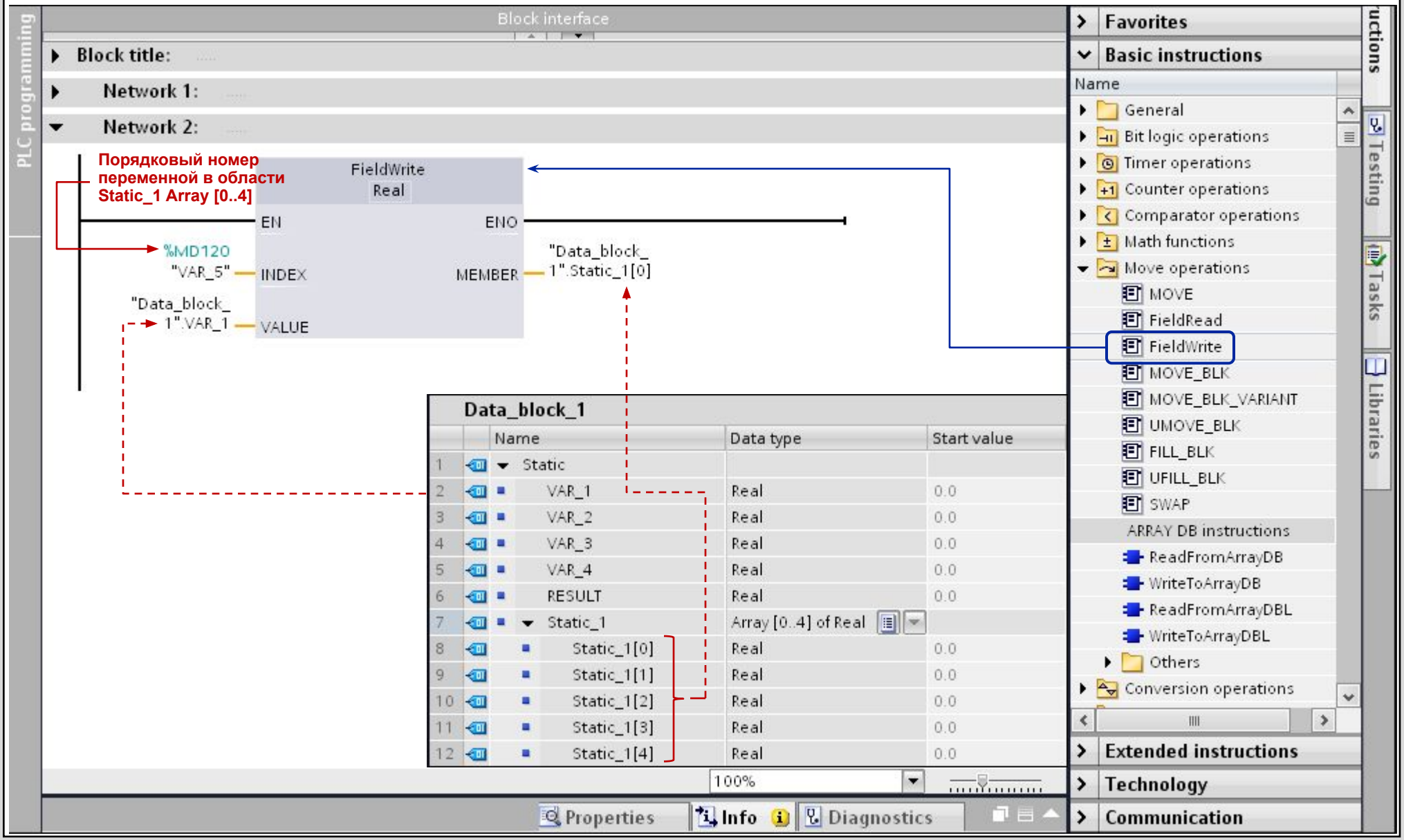
Инструкция «чтение переменной из заданной области»



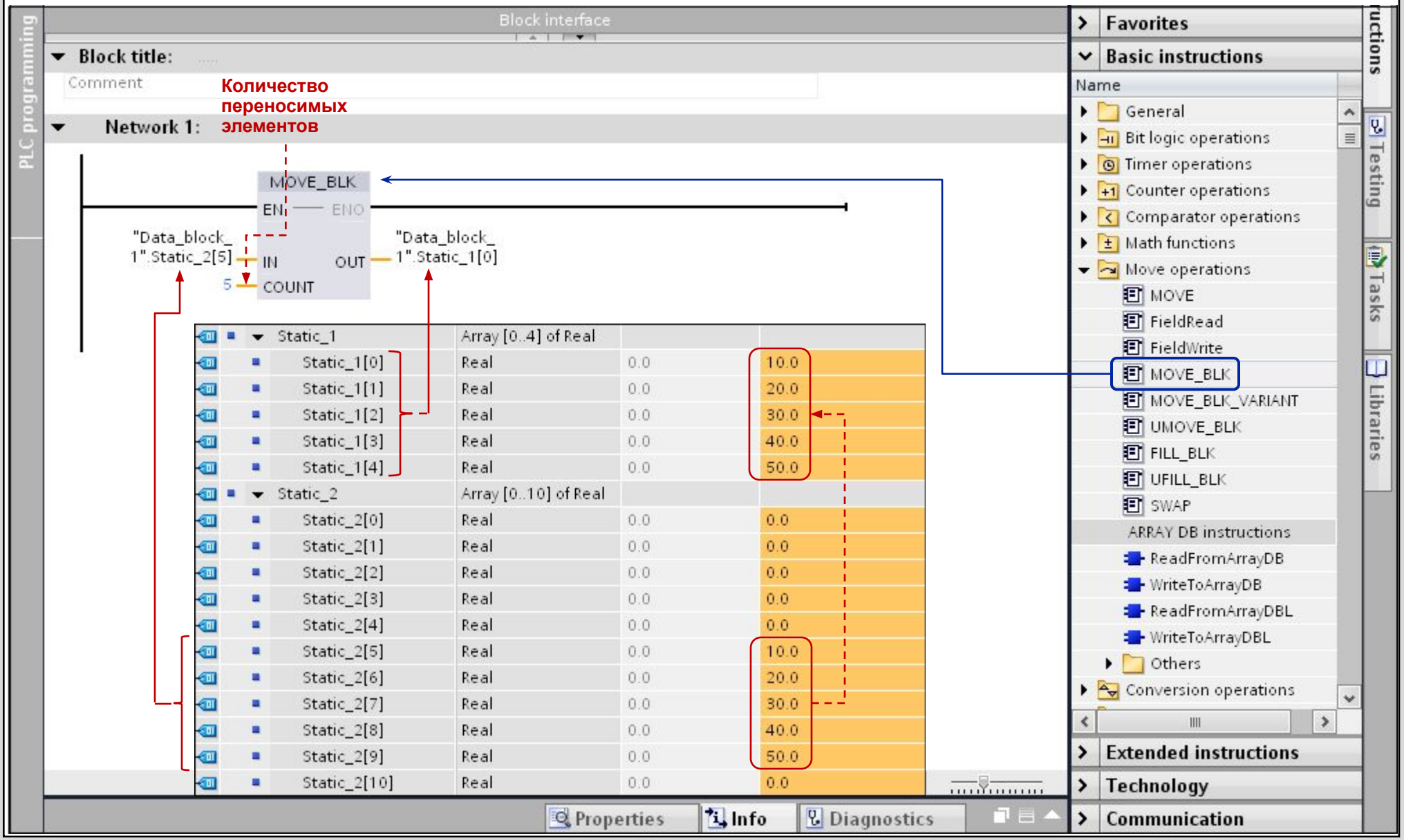
Чтение переменной из заданной области OnLine



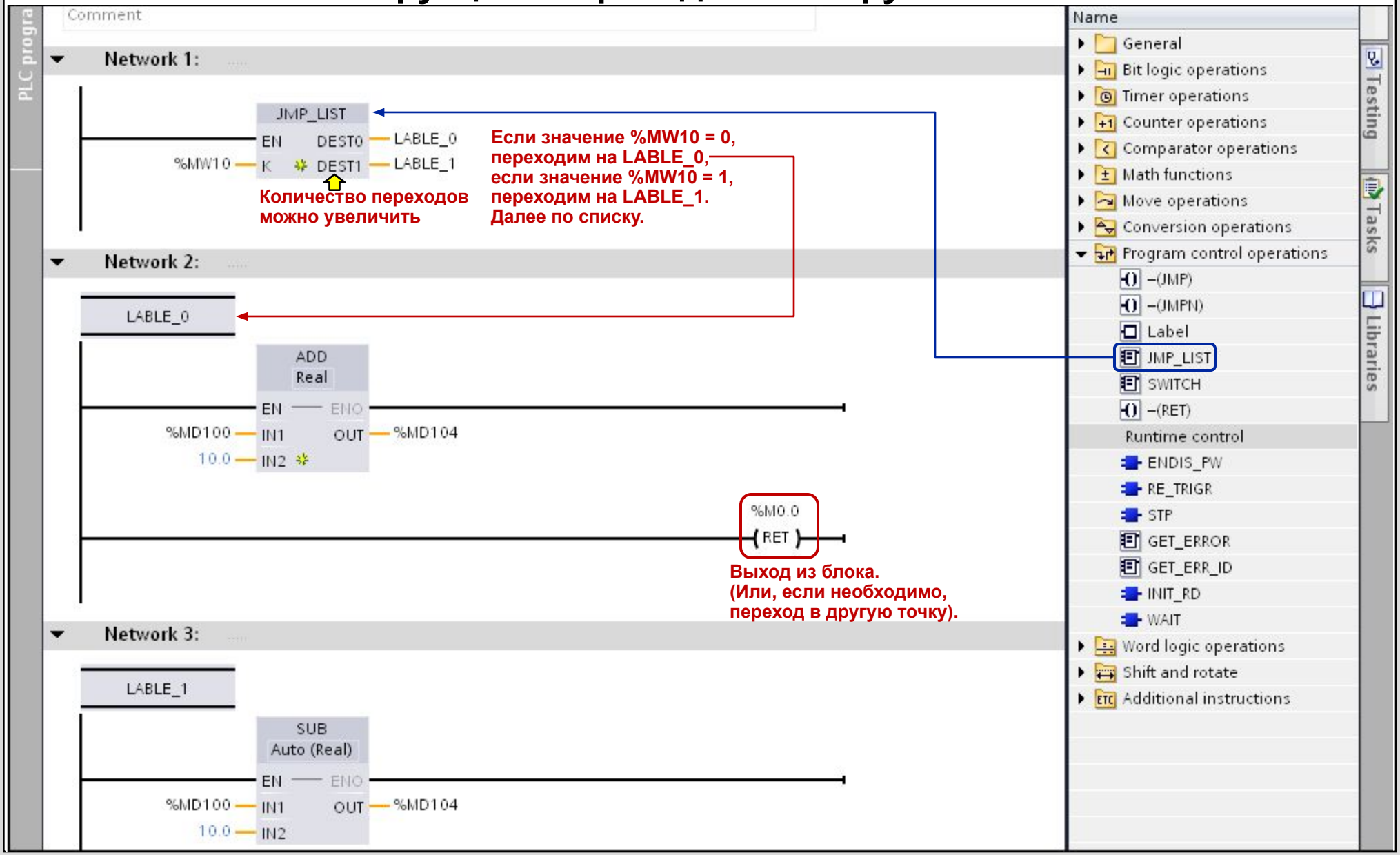
Инструкция «запись переменной в заданную область»

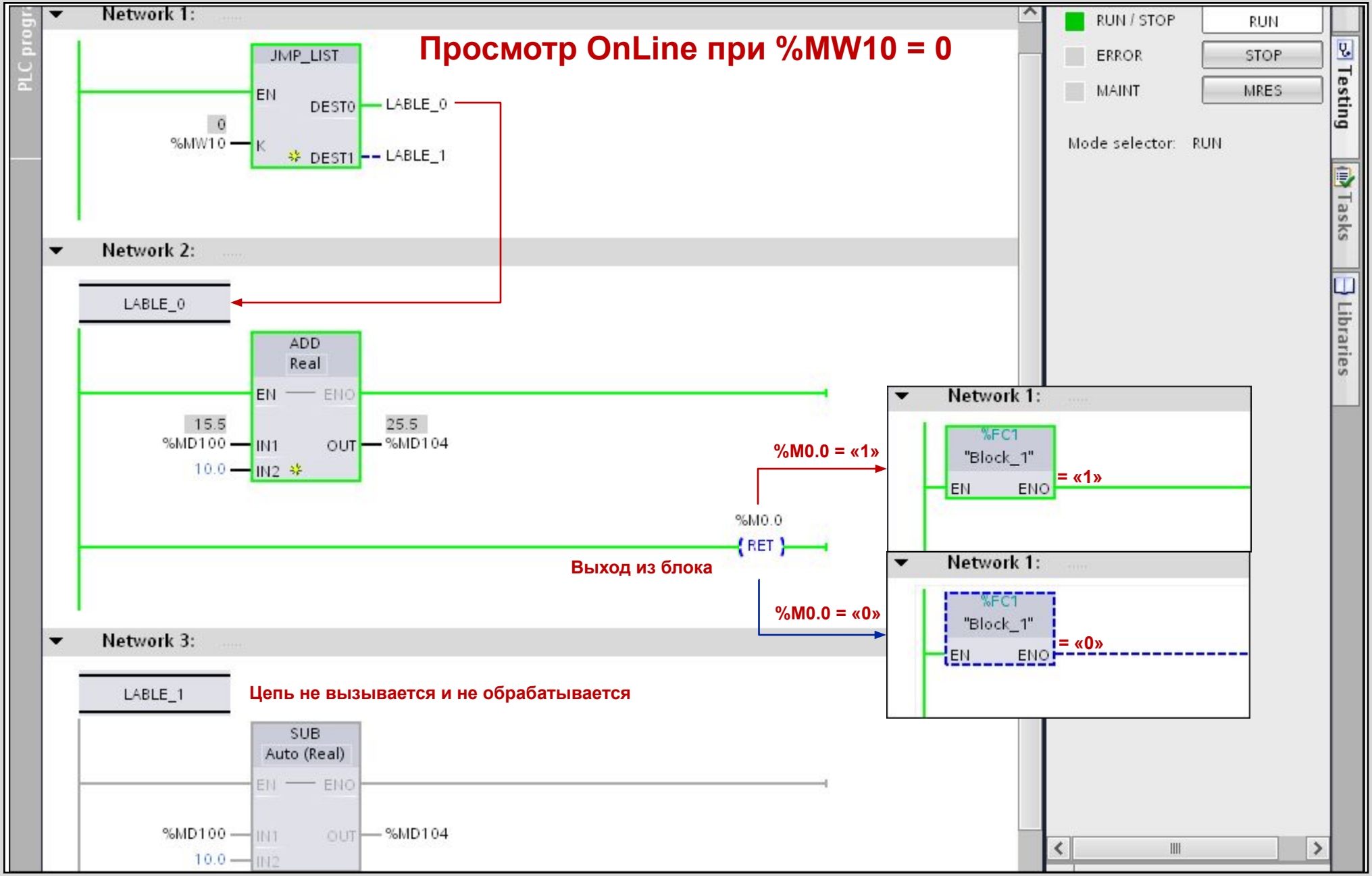


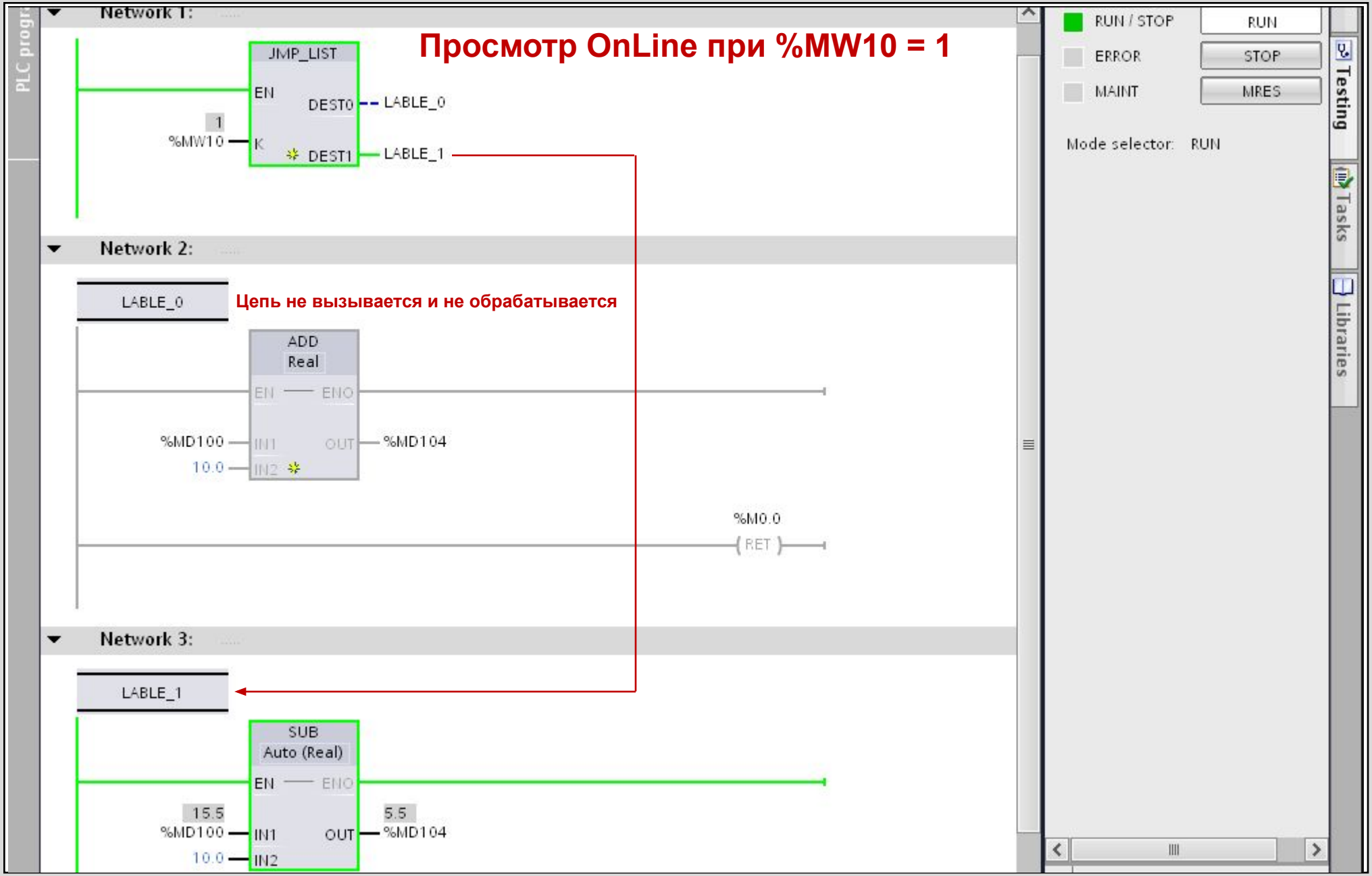
Инструкция копирования из одной области данных в другую

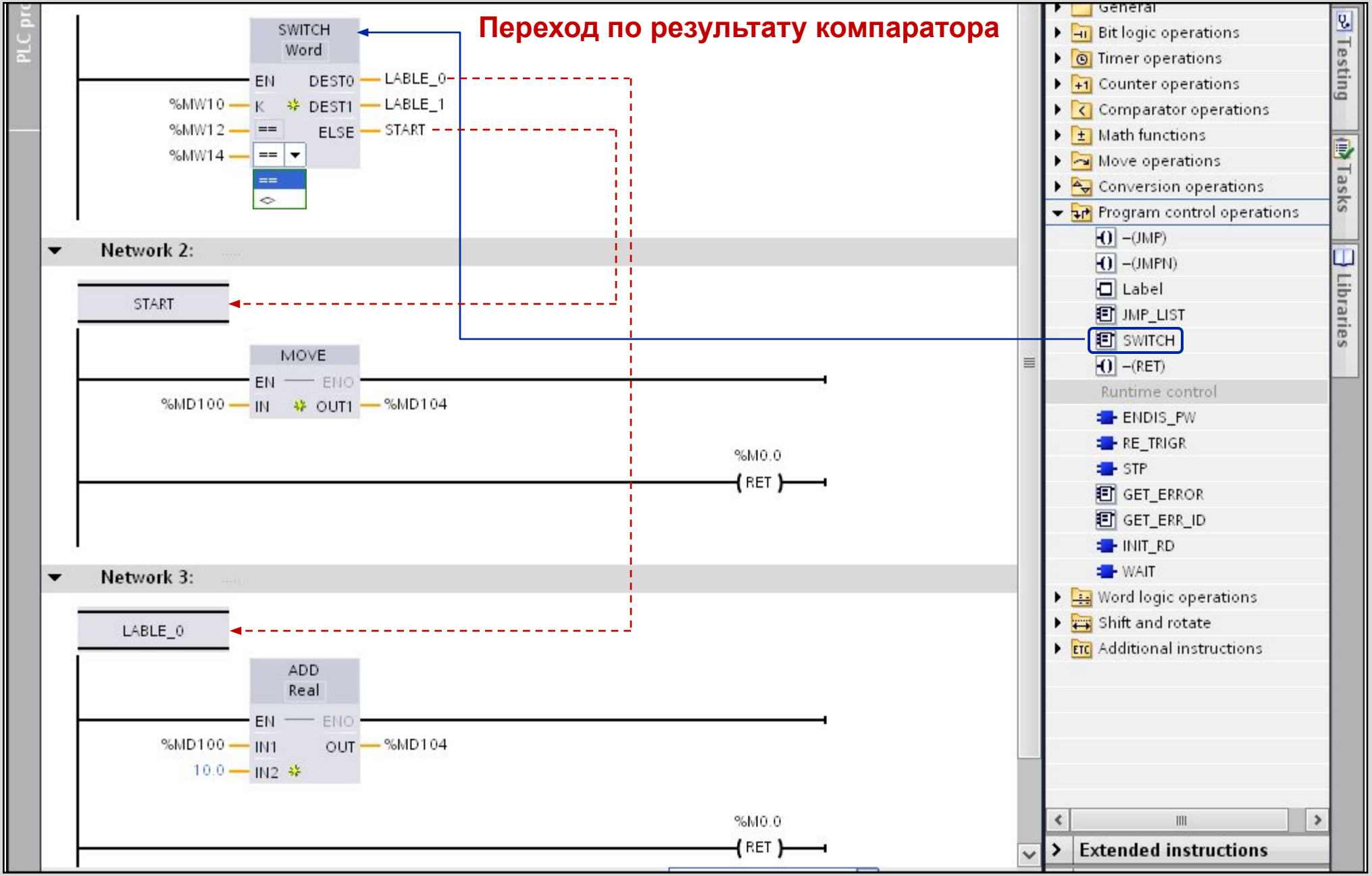


Инструкция «переход по номеру в списке»

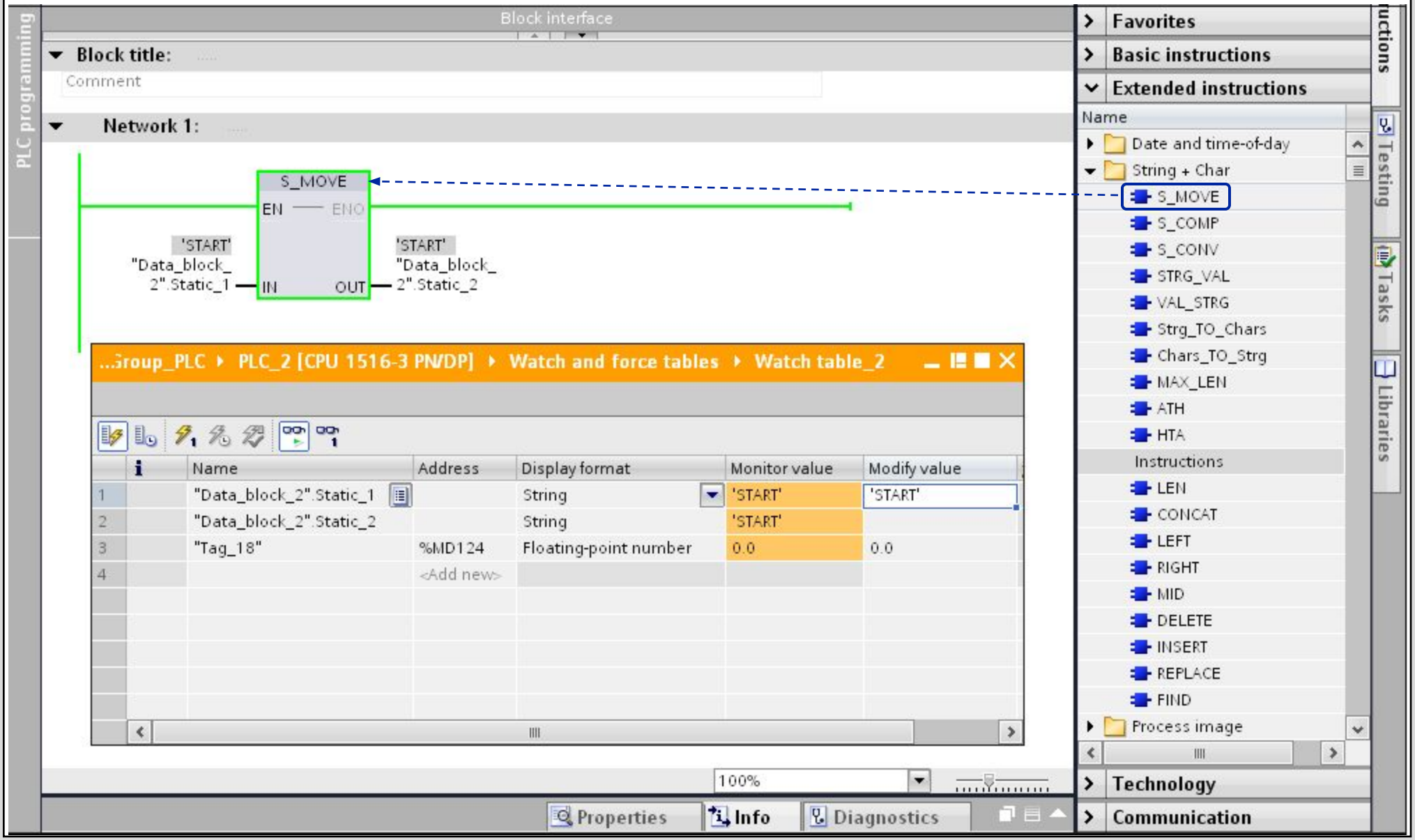



Network 1:
JMP_LIST
EN
DEST0
LABLE_0
0
%MW10
K
DEST1
LABLE_1
Просмотр OnLine при %MW10 = 0
RUN / STOP
RUN
ERROR
STOP
MAINT
MRES
Mode selector: RUN
Testing
Tasks
Libraries
Network 2:
LABLE_0
ADD
Real
EN
ENO
15.5
%MD100
IN1
OUT
25.5
%MD104
10.0
IN2
%M0.0 = «1»
%M0.0
RET
Выход из блока
%M0.0 = «0»
Network 1:
%FC1
"Block_1"
EN
ENO
= «1»
Network 1:
%FC1
"Block_1"
EN
ENO
= «0»
Network 3:
LABLE_1
Цепь не вызывается и не обрабатывается
SUB
Auto (Real)
EN
ENO
%MD100
IN1
OUT
%MD104
10.0
IN2

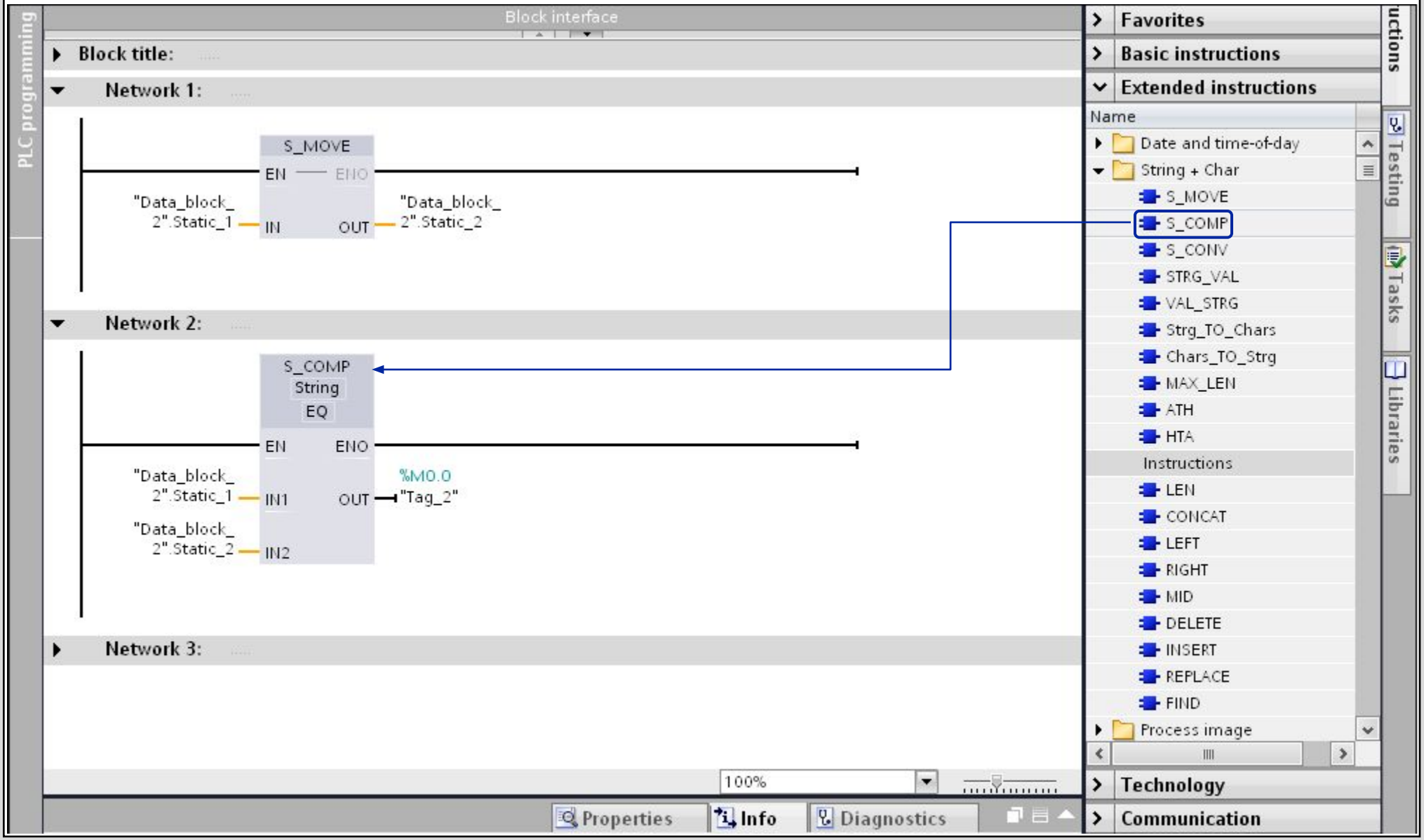

Network 1:
JMP_LIST
EN
DEST0
LABLE_0
1
%MW10
K
DEST1
LABLE_1
Просмотр OnLine при %MW10 = 1
Network 2:
LABLE_0
Цепь не вызывается и не обрабатывается
ADD
Real
EN
ENO
%MD100
IN1
OUT
%MD104
10.0
IN2
%M0.0
RET
Network 3:
LABLE_1
SUB
Auto (Real)
EN
ENO
15.5
%MD100
IN1
OUT
5.5
%MD104
10.0
IN2
RUN / STOP
RUN
ERROR
STOP
MAINT
MRES
Mode selector:
RUN
Testing
Tasks
Libraries


Переход по результату компаратора
SWITCH
Word
EN
DEST0
LABLE_0
%MW10
K
DEST1
LABLE_1
%MW12
==
ELSE
START
%MW14
==
<>
Network 2:
START
MOVE
EN
ENO
%MD100
IN
OUT1
%MD104
%M0.0
RET
Network 3:
LABLE_0
ADD
Real
EN
ENO
%MD100
IN1
OUT
%MD104
10.0
IN2
%M0.0
RET
General
Bit logic operations
Timer operations
Counter operations
Comparator operations
Math functions
Move operations
Conversion operations
Program control operations
–(JMP)
–(JMPN)
Label
JMP_LIST
SWITCH
–(RET)
Runtime control
ENDIS_PW
RE_TRIGR
STP
GET_ERROR
GET_ERR_ID
INIT_RD
WAIT
Word logic operations
Shift and rotate
Additional instructions
Extended instructions
Testing
Tasks
Libraries

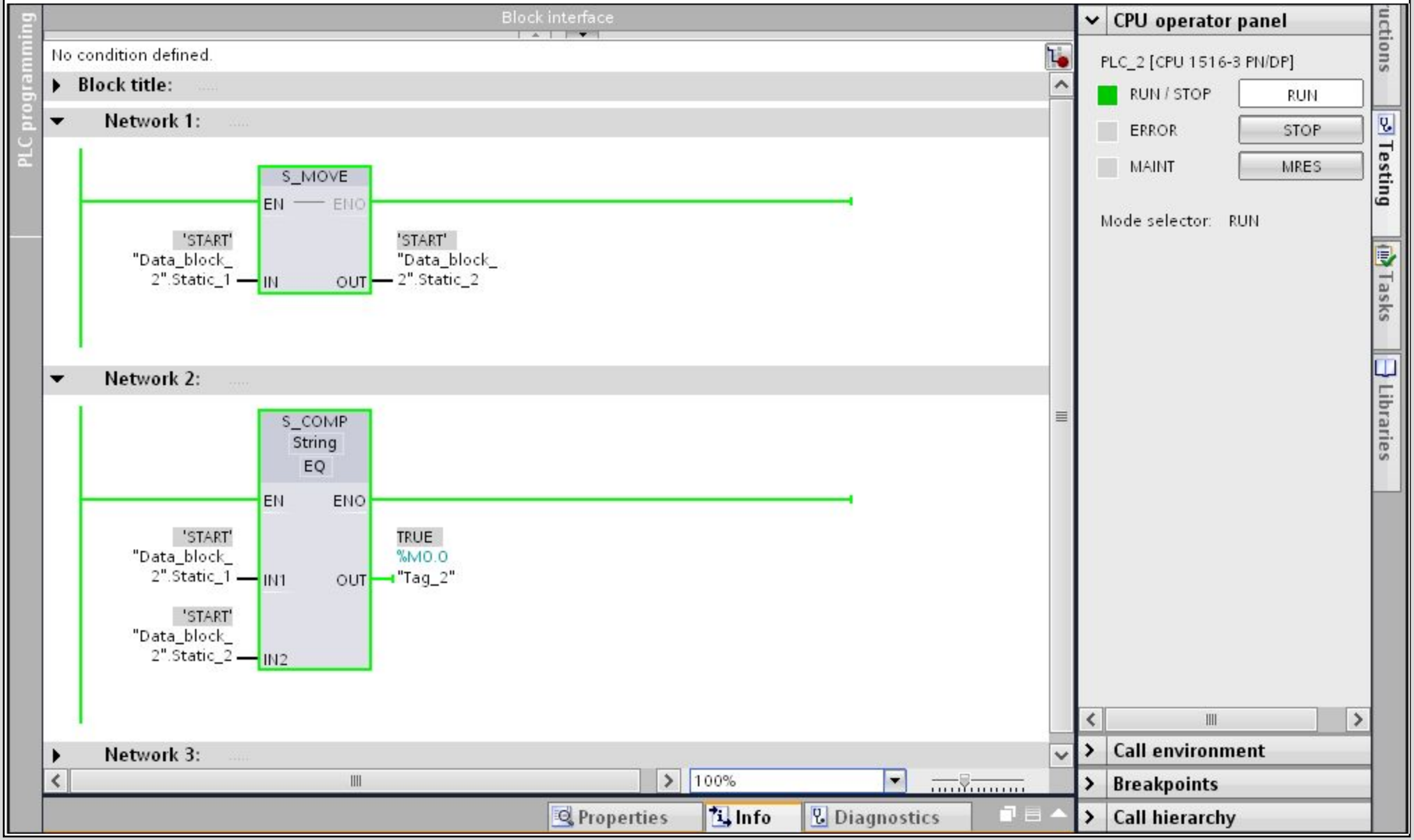
Строковые команды. Перенос строки из одной области в другую



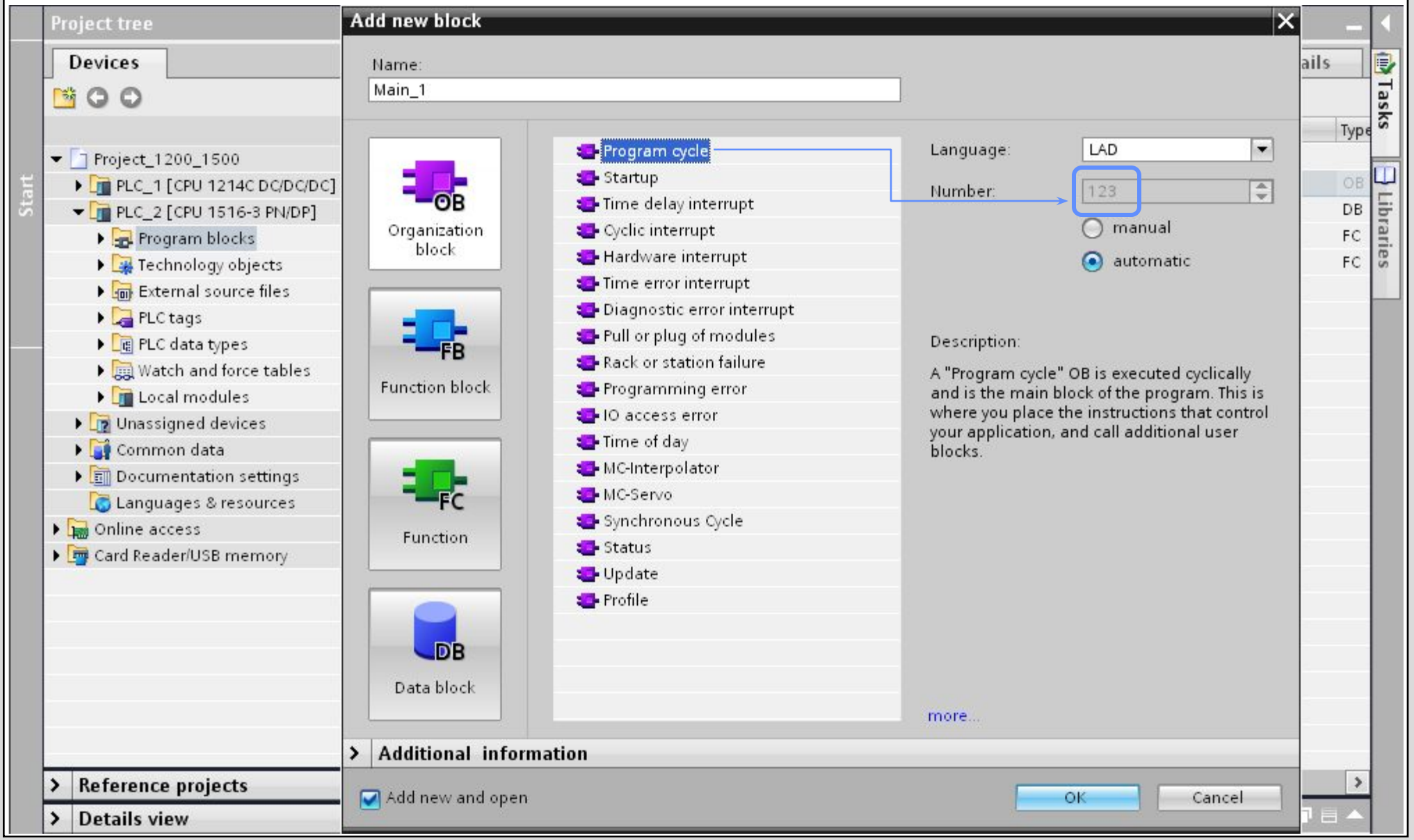
Строковые команды. Сравнение областей, содержащих строки



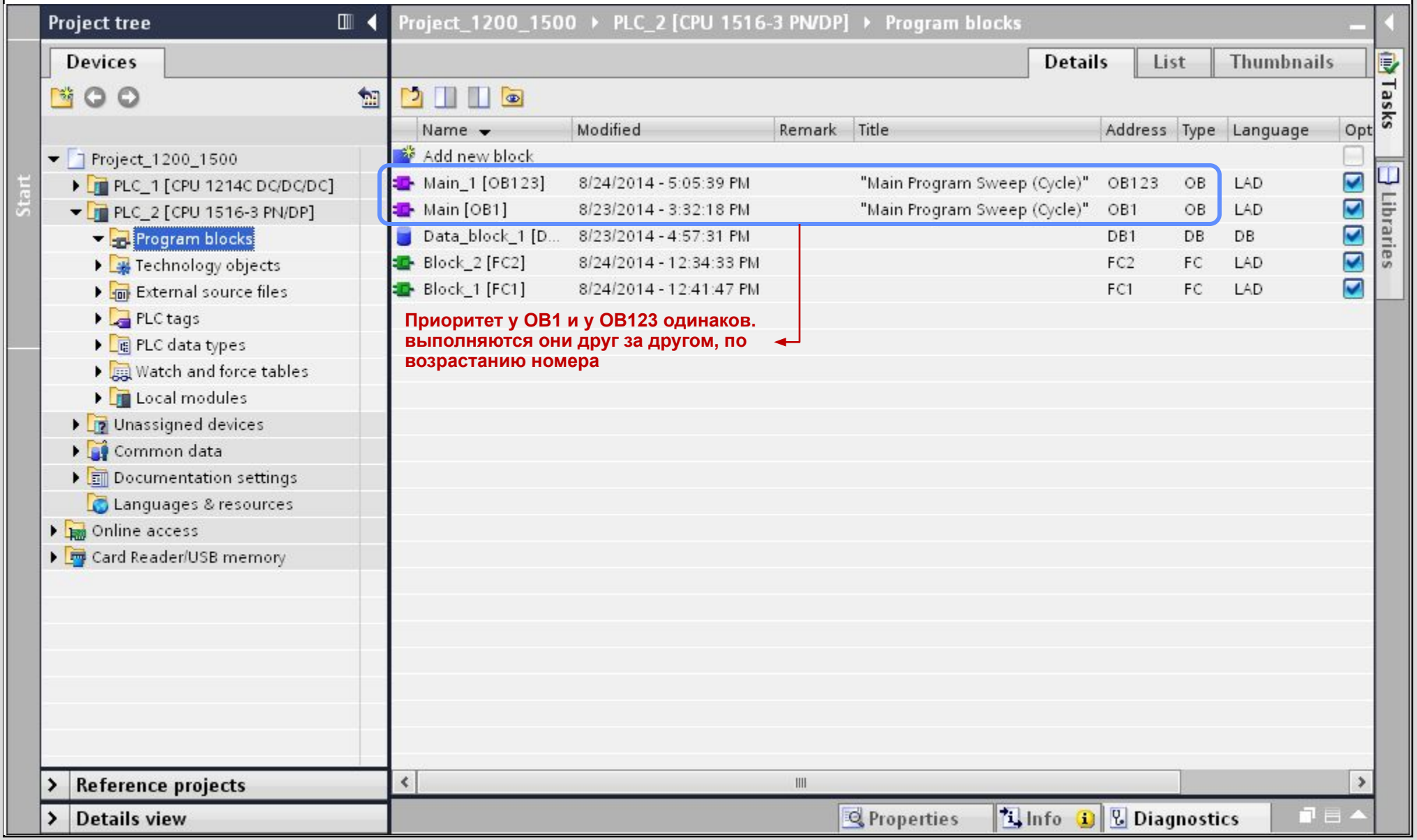
Сравнение областей, содержащих строки, в режиме OnLine



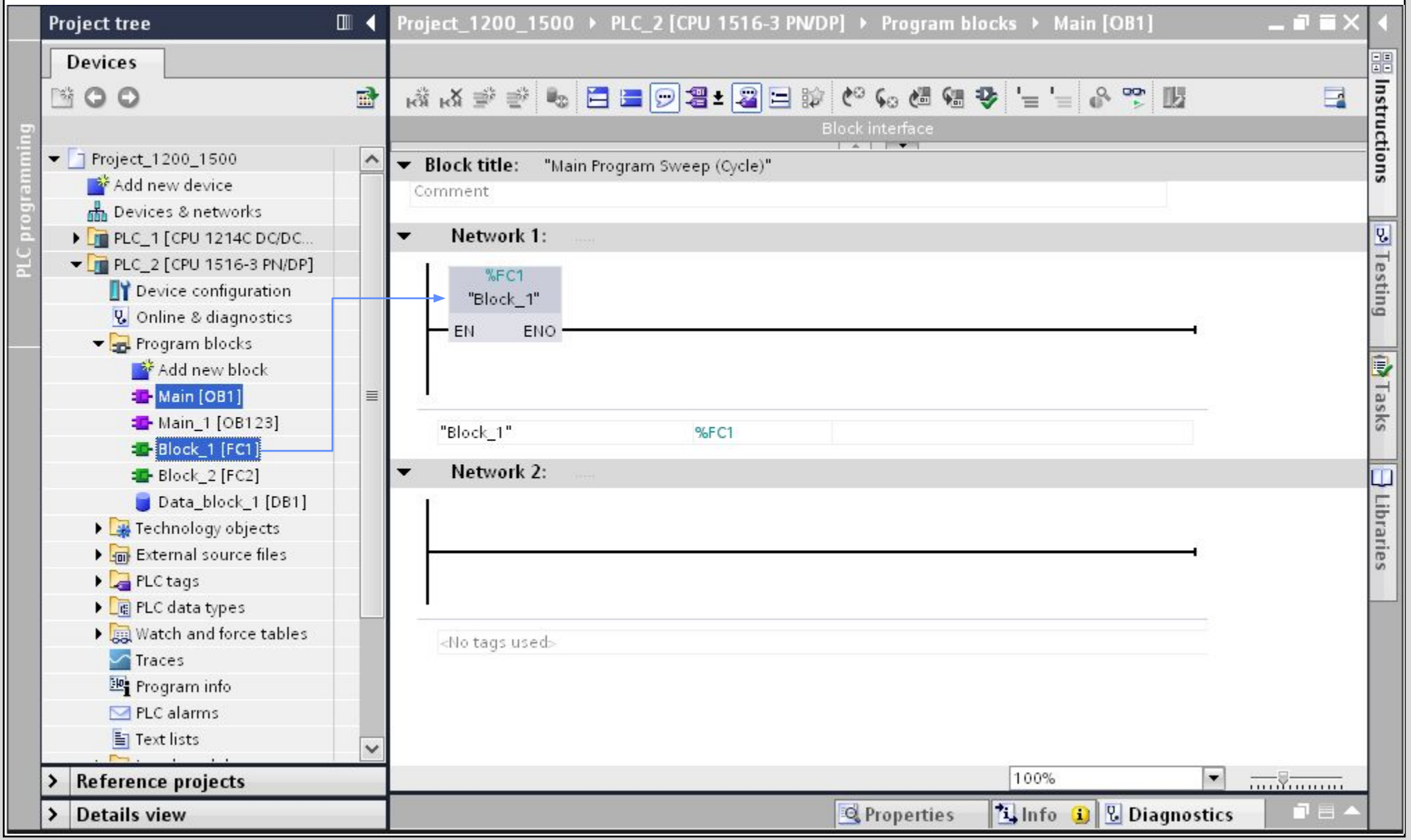
Создание нескольких OB типа "Main Program Sweep (Cycle)"



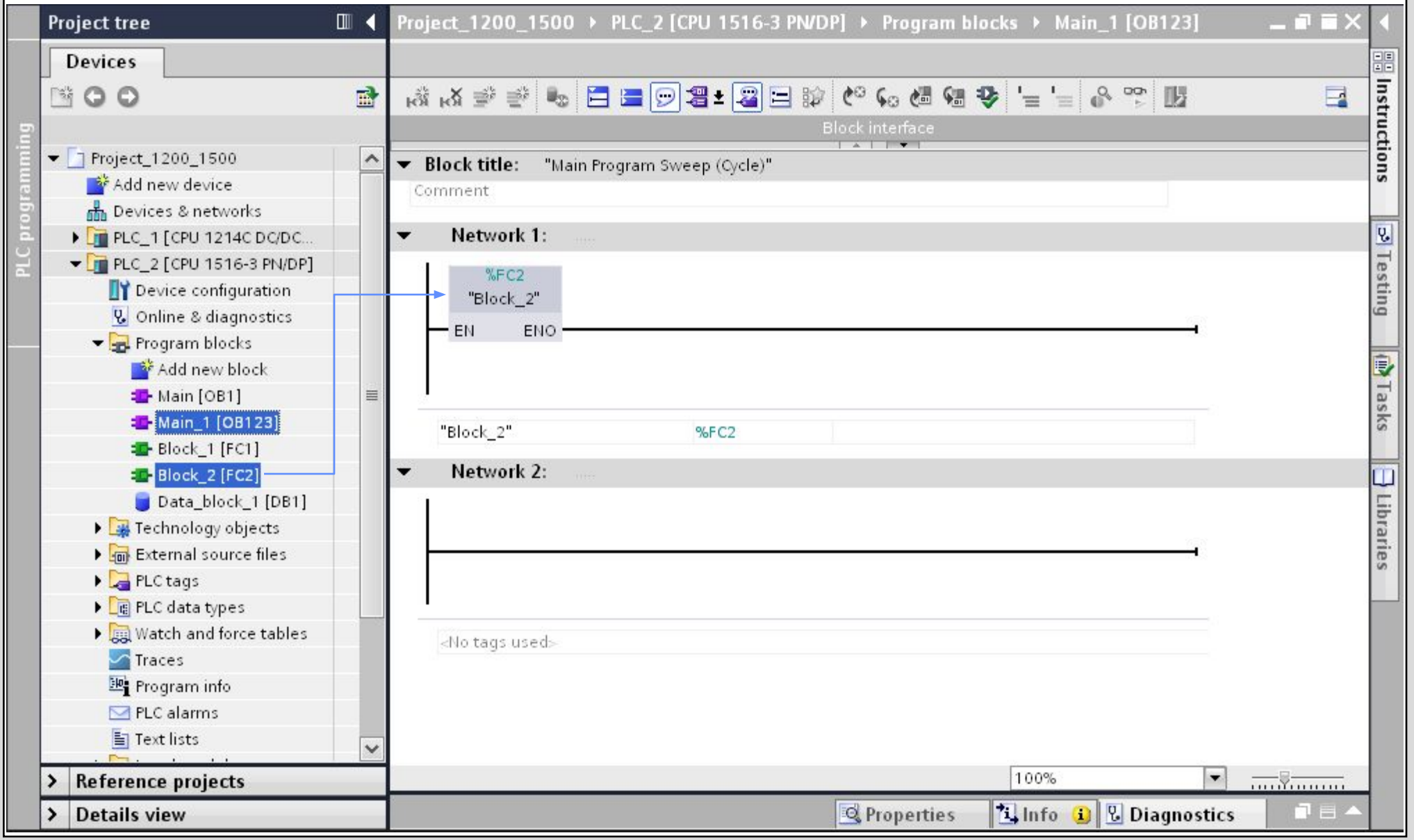
Создание нескольких OB типа "Main Program Sweep (Cycle)"



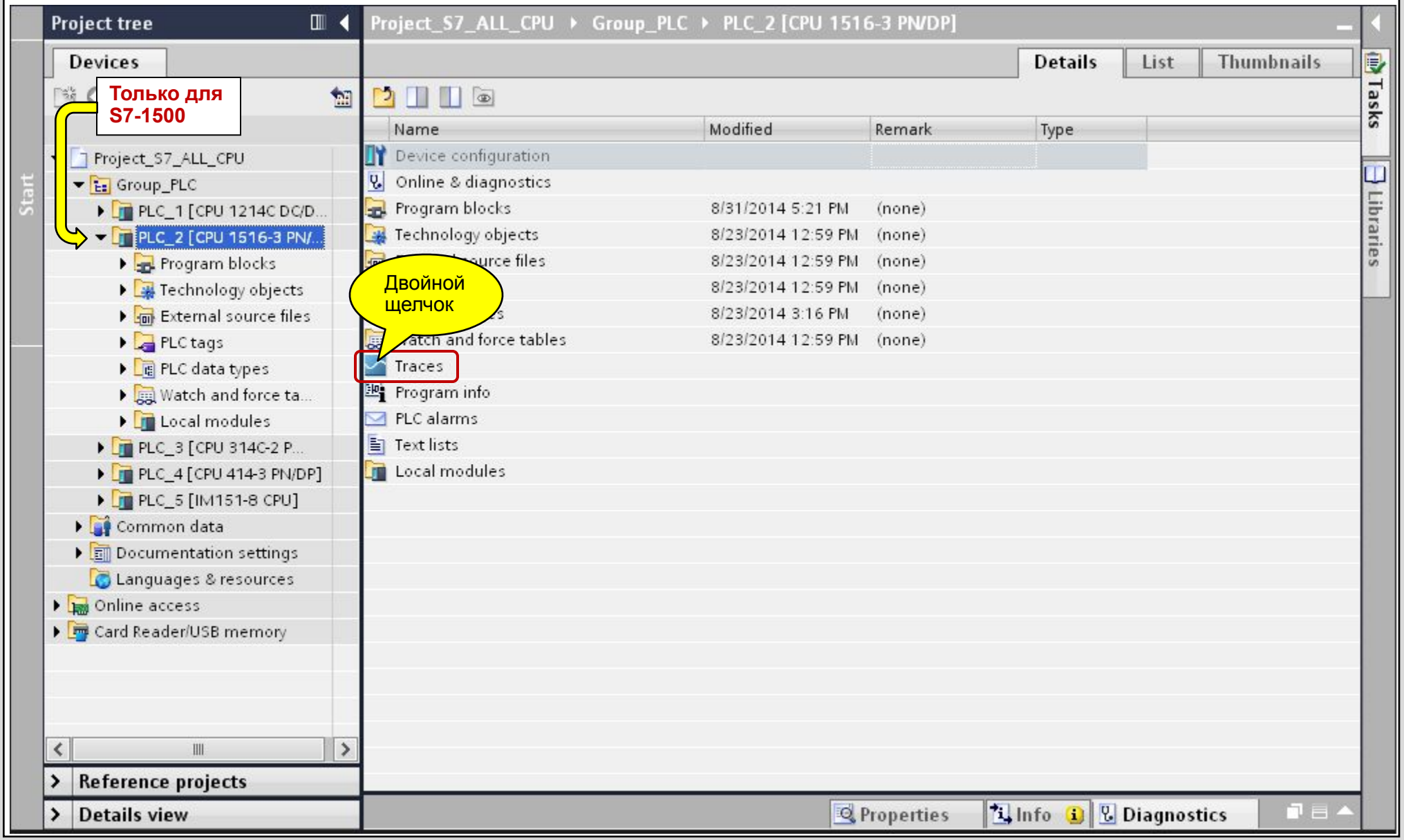
Вызов программных блоков из OB1 "Main Program Sweep (Cycle)"



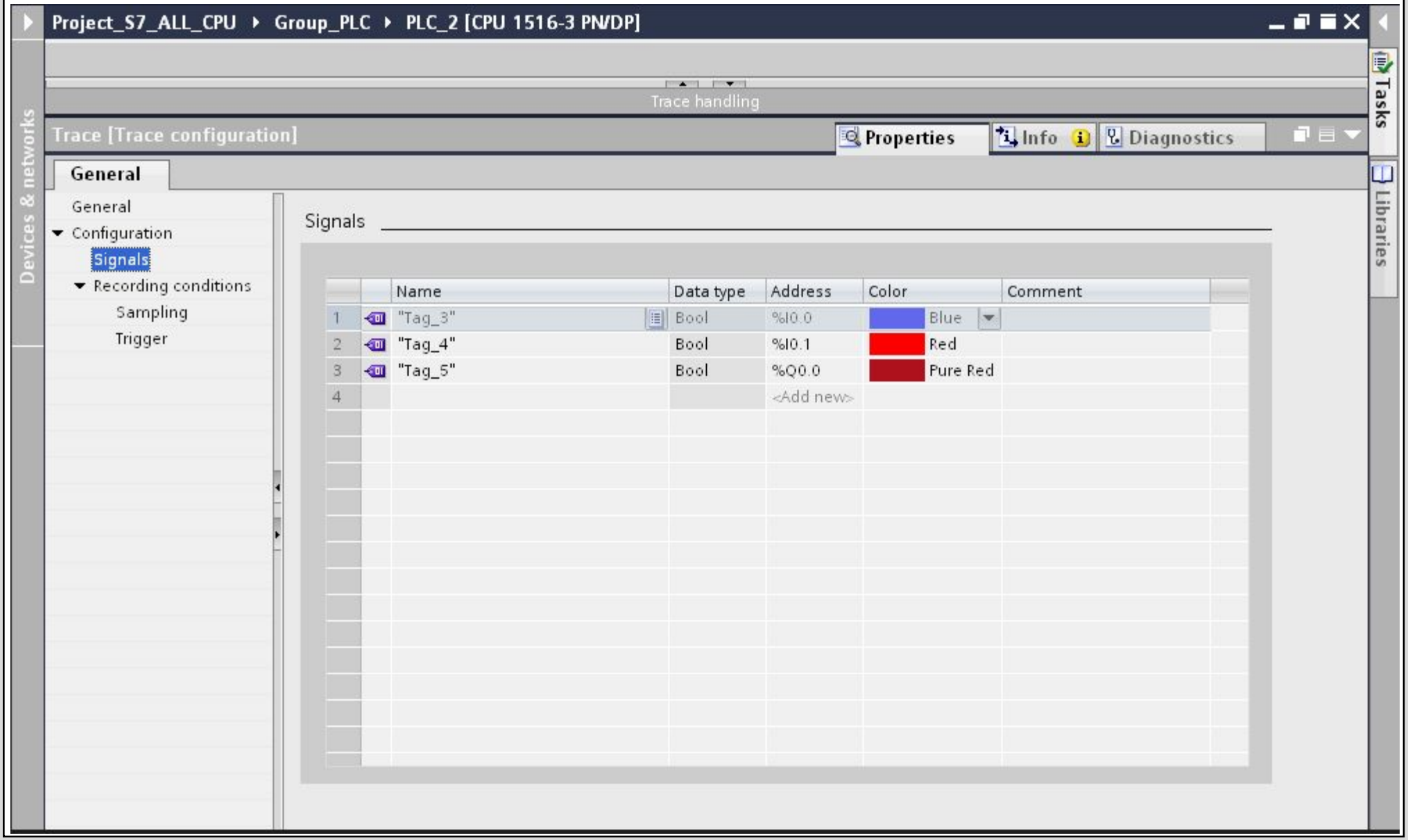
Вызов программных блоков из OB123 "Main Program Sweep (Cycle)"



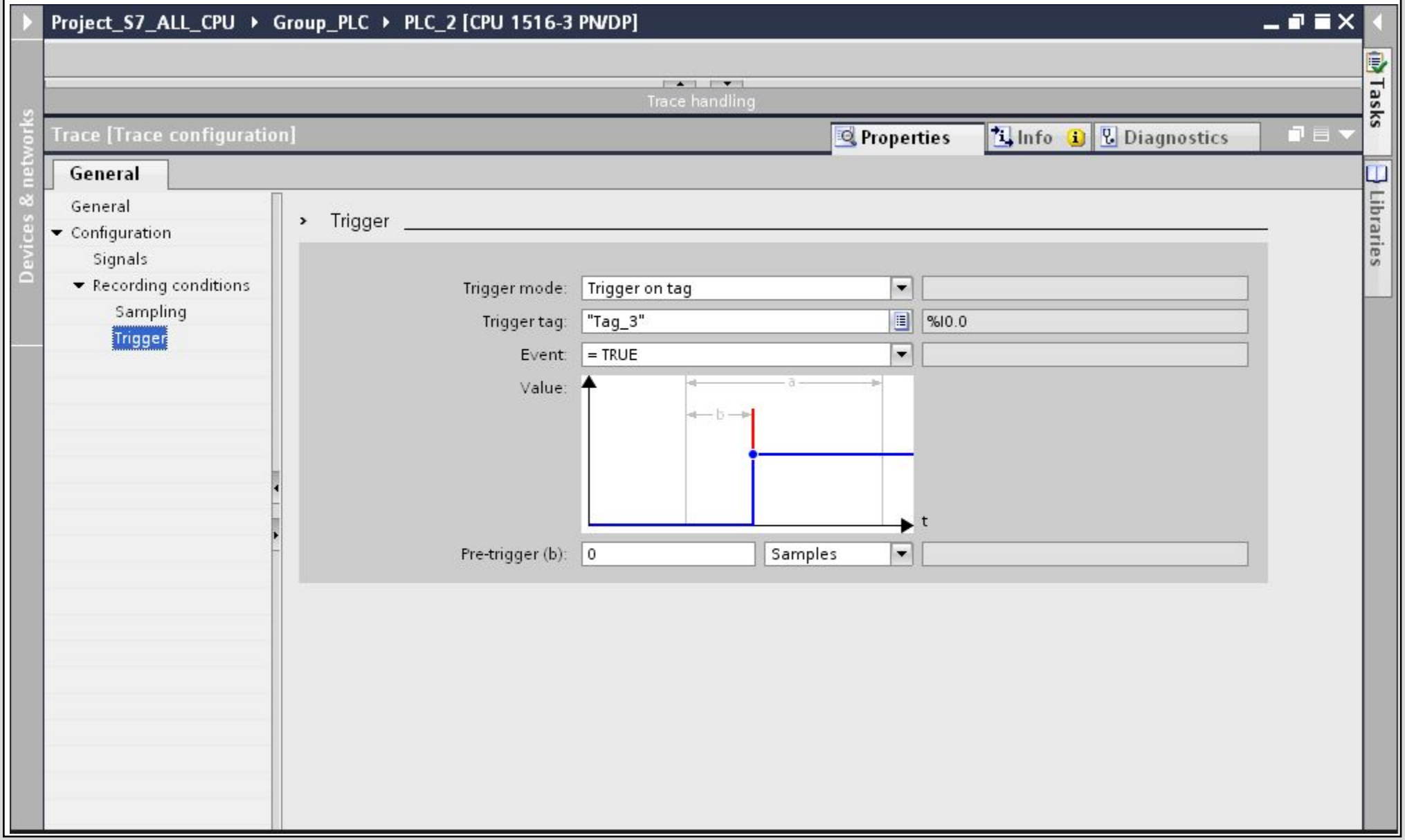
Функция отслеживания значений переменных во времени



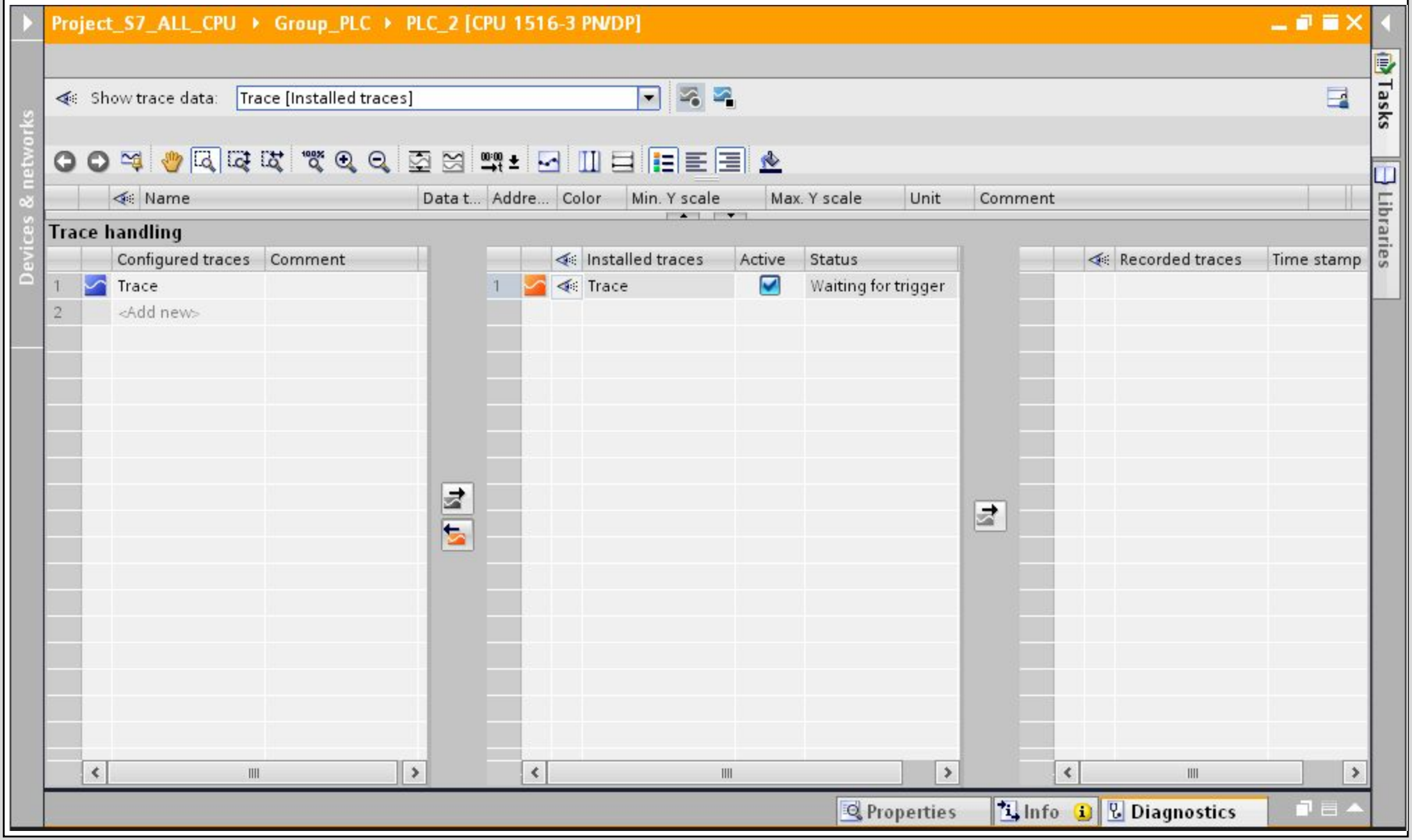
Функция отслеживания значений переменных во времени



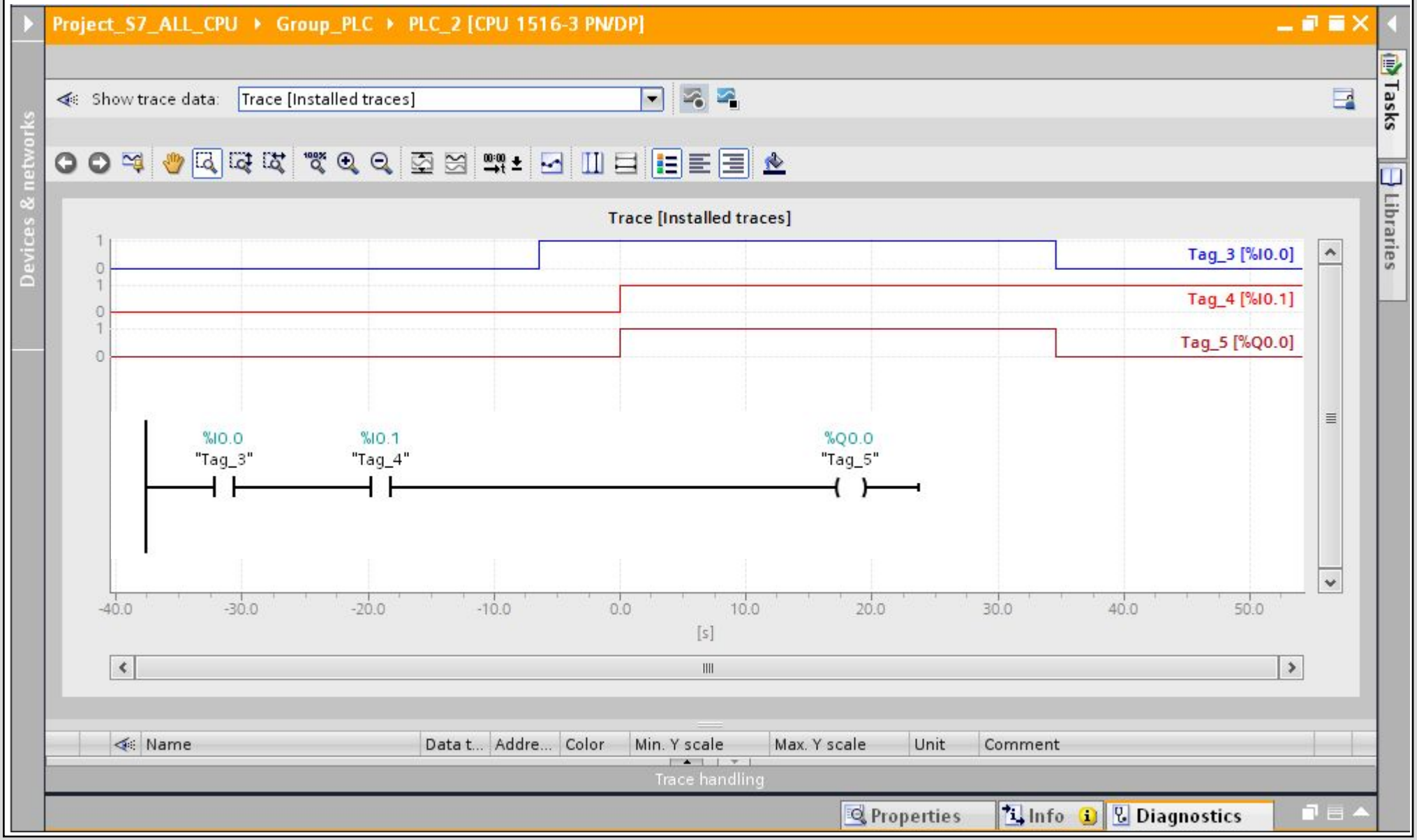
Функция отслеживания значений переменных во времени



Функция отслеживания значений переменных во времени OnLine



Функция отслеживания значений переменных во времени OnLine



Конец раздела 11. Окно навигации

- Основы алгебры логики
- Общие сведения, создание проекта. Конфигурирование станции
- Программные блоки (FC/FB)
- Блоки данных (DB)
- Регистры, служебные флаги. Библиотека программных инструкций.
- Организационные блоки (OB)
- Модули обработки аналоговых сигналов
- Программирование на языках SCL, GRAPH
- Тестирование и отладка
- Системы с сетевой конфигурацией

Конфигурирование ПЛК S7-1200, S7-1500