

А.С. Гирник, А.Л. Федянин

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ КОНВЕЙЕРОМ



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

А.С. Гирник, А.Л. Федянин

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ КОНВЕЙЕРОМ

Методические указания к выполнению лабораторной работы

Издательство
Томского политехнического университета
2024

Гирник А.С.

Г51 Автоматизированная система управления конвейером : методические указания к лабораторным работам / А.С. Гирник, А.Л. Федянин ; Томский политехнический университет. – Томск : Изд-во Томского политехнического университета, 2024. – 99 с.

В методических указаниях изложена последовательность проектирования автоматизированной системы управления конвейером в составе модульного контроллера, человеко-машинного интерфейса на борту сенсорной панели и SCADA системы.

Предназначено для студентов и других сторонних слушателей, обучающихся по направлению 13.03.02 «Электроэнергетика и электротехника».

© ФГАОУ ВО НИ ТПУ, 2024

© Гирник А.С., 2024

© Оформление. Издательство Томского
политехнического университета, 2024

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	3
ВВЕДЕНИЕ.....	4
1. ОПИСАНИЕ УЧЕБНОГО ОСНАЩЕНИЯ.....	5
2. ЛАБОРАТОРНЫЙ ПРАКТИКУМ	21
2.1. Изучение лабораторной установки «Конвейер».....	21
2.2. Общая концепция системы управления в составе OPC сервера.....	26
2.3. Создание программы на языках программирования CFC и FBD в среде CODESYS для контроллера XC-CPU201.....	29
2.4. Создание программы на языке программирования ST в среде CODESYS для контроллера HMI панели визуализации XV102	49
2.5. Создание графического интерфейса управления в среде GALILEO для HMI панели визуализации XV102	54
2.6. Создание графического SCADA интерфейса управления в среде TRACE MODE	62
2.7. Настройка OPC сервера.....	81
2.9. Аprobация созданной системы управления.....	83
3. УСТАНОВКА ПОГРАММНОГО ОБЕСПЕЧЕНИЯ	89
3.1. Установка и настройка CODESYS 2.3.9 SP8	89
3.2. Установка Trace Mode 6.09.....	91
3.3. Установка и лицензирование Galileo 8.1	91
3.4. Обновление прошивки HMI панели XV102.....	93
СПИСОК ЛИТЕРАТУРЫ.....	99

ВВЕДЕНИЕ

В настоящее время современные технологии позволяют создавать системы управления на базе таких микропроцессорных устройств, с помощью которых можно автоматизировать практически любой технологический процесс [1]. Ранее алгоритм управления реализовывался внутри больших релейно-контактных шкафов. Сегодня она может размещаться внутри компактной электронной памяти программируемой аппаратуры. Такие средства автоматизации не только повышают эффективность производства, но также освобождают человека от выполнения работы по контролю за состоянием технологического процесса и формированию управляющих воздействий на исполнительные органы рабочих механизмов. Применение программируемых логических контроллеров и реле в системах управления производственных объектов способствует автоматизации технологических процессов. В настоящее время данные электронные аппараты относятся к числу обладателей искусственного интеллекта.

Современные автоматизированные системы управления разделяются на три уровня:

1. Верхний уровень – управление технологическим процессом с помощью виртуальной панели оператора, которая представляет собой SCADA-программу, установленную на персональном компьютере.
2. Средний уровень – программно-электрическая часть, реализуемая с помощью программируемых логических контроллеров или реле, имеющих информационные цифровые или аналоговые входы и выходы.
3. Нижний (полевой) уровень – физические модели, исполнительные органы, управляемые элементами среднего уровня.

В совокупности все три уровня автоматизации представляют собой интеллектуальную систему управления технологическими процессами. В данном учебном пособии рассматривается реализация нижнего и среднего уровней на базе программного обеспечения и оборудования производства Eaton (Moeller). Также уделено внимание языкам программирования стандарта МЭК, на которых создаются программные алгоритмы управления такими объектами, как ленточный конвейер, автоматическая дверь, сообщающиеся сосуды и нефтяная задвижка. Эти объекты могут рассматриваться как отдельные элементы автоматики.

В учебном пособии рассматриваются программные продукты, имеющиеся в распоряжении Инженерной школы энергетики НИ ТПУ, EasySoft и CodeSys.

1. ОПИСАНИЕ УЧЕБНОГО ОСНАЩЕНИЯ

Все лабораторные работы, приведённые в данном учебном пособии, проводятся на специализированном оборудовании, которым оснащён учебный центр ТПУ «Технические средства автоматизации» (рис. 1):

- пять универсальных учебных стендов с контроллерами;
- лабораторная установка ленточного конвейера;
- лабораторная установка автоматической двери;
- лабораторная установка тепло-станции (сообщающихся сосудов);
- лабораторная установка нефтяной задвижки.



*Рис. 1. Учебный центр
«Технические средства автоматизации»*

К каждому универсальному стенду (рис. 2) специальными кабелями подключена своя отдельная лабораторная установка (физическая модель).

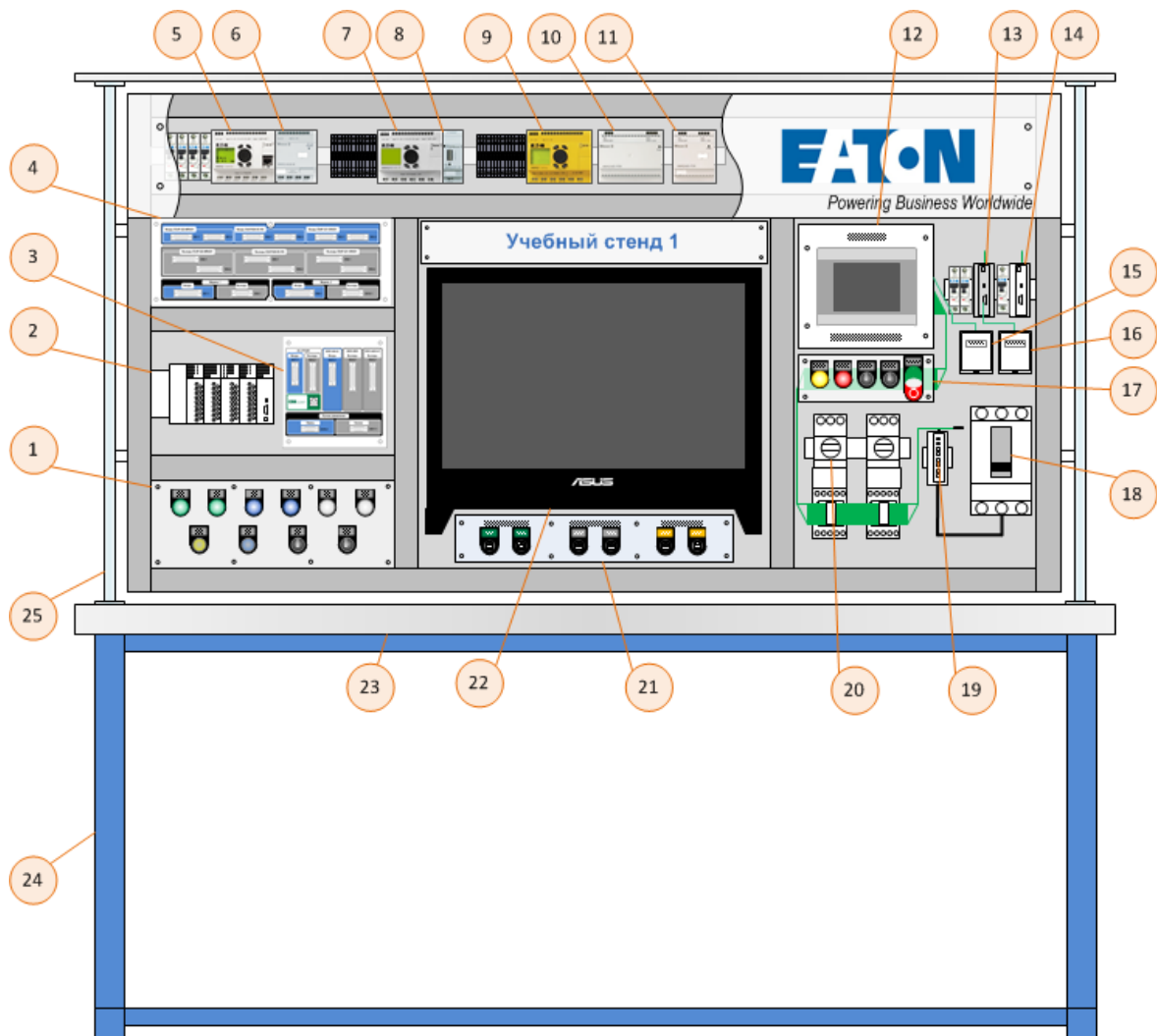


Рис. 2. Структура универсального учебного стенда:

- 1 – кнопочный пост; 2 – модульный ПЛК XC-201 с набором модулей;
 3, 4 – интерфейсные панели разъёмов; 5 – ПЛК EC4P-222-MRAD1; 6 – модуль расширения EASY411-DC-ME; 7 – программируемое реле EASY820-DC-RC;
 8 – модуль Ethernet EASY209-SE; 9 – реле безопасности Easy-Safety ES4P-221-DRXD1;
 10 – блок питания EASY600 POW; 11 – блок питания EASY400 POW; 12 – HMI-панель XV-102-E6-57TVRC-10; 13 – концентратор EU5C-SWD-CAN; 14 – концентратор EASY800 with SmartWire-DT EASY806-DC-SWD; 15 – розетка подключения к CANopen HMI-панели; 16 – розетка подключения к CANopen концентратора EU5C-SWD-CAN; 17 – кнопочный пост; 18 – автомат NZMN2-ME90; 19 – SWD-модуль NZM-XSWD-704 соединения для NZM; 20 – пусковая сборка MSC-DEA-12-M7;
 21 – панель сетевых коммуникаций; 22 – персональный компьютер;
 23 – стол; 24 – стойки; 25 – система подвески оборудования

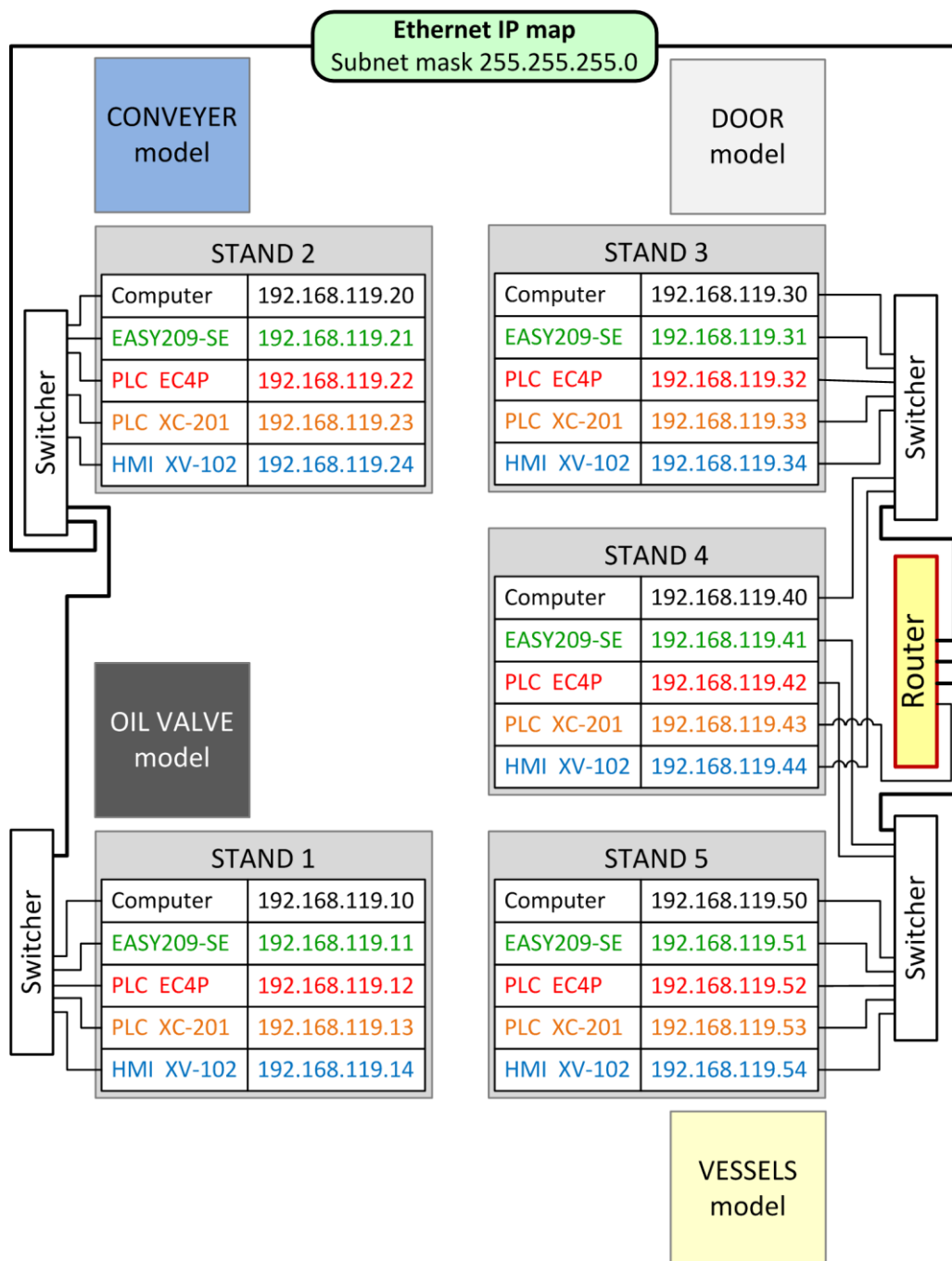


Рис. 3. Подключение всего учебного оборудования в общую сеть Ethernet

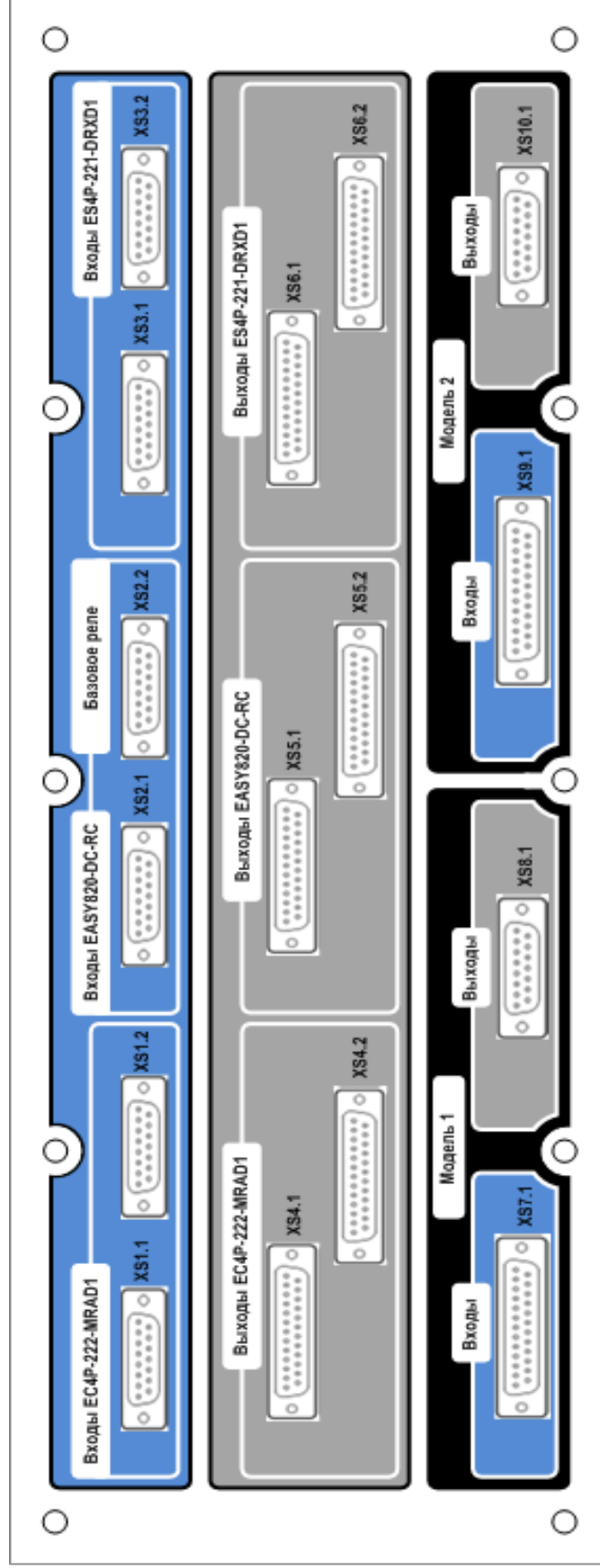


Рис. 4. Первая интерфейсная панель разъемов

На рис. 4, 5 показаны интерфейсные панели разъёмов для электрического соединения:

- 1) кнопок управления с входами аппаратов (реле, контроллеры);
- 2) сигнальных ламп с выходами аппаратов;
- 3) датчиков физических моделей с входами аппаратов;
- 4) исполнительных элементов физических моделей (контакты пуска двигателя, соленоиды и т. д.) с выходами аппаратов.

Первая панель разъёмов (рис. 4) разделена на 3 основных секции:

1. *Синяя секция.* В ней расположены электрические разъёмы, которые соединены со входами программируемых реле и контроллеров. Причём для каждого аппарата предусмотрена пара дублированных разъёмов. То есть на каждом из двух разъёмов контакты с одинаковыми номерами соединены параллельно друг другу. Это необходимо для того, чтобы можно было подключать ко входам контроллера как кабель для передачи сигналов от датчиков лабораторной установки, так и кабель для передачи сигналов управления от кнопок на универсальном стенде.

2. *Серая секция.* В ней расположены электрические разъёмы, которые соединены с выходами программируемых аппаратов. Здесь точно так же предусмотрено парное дублирование разъёмов, для того чтобы можно было подавать управляющие сигналы от контроллеров как на лабораторную установку, так и на сигнальные лампы универсального стенда.

3. *Чёрная секция.* В ней расположены электрические разъёмы, через которые можно подключаться к датчикам лабораторной установки, а также к её исполнительным элементам.

Вторая панель разъёмов (рис. 5) предназначена для подключения к модульному контроллеру XC-201 и его отдельным модулям. Эта панель также условно разделена на секции: синие (для входных сигналов), серые (для выходных сигналов) и черные (для ручного управления от кнопок на универсальном стенде).

На рис. 6–8 показаны электрические схемы соединения первой панели разъёмов со входами и выходами таких программируемых электронных аппаратов, как:

- программируемый логический контроллер EC4P-222-MRAD1;
- программируемое реле EASY820-DC-RC;
- реле безопасности ES4P-221-DRXD1.

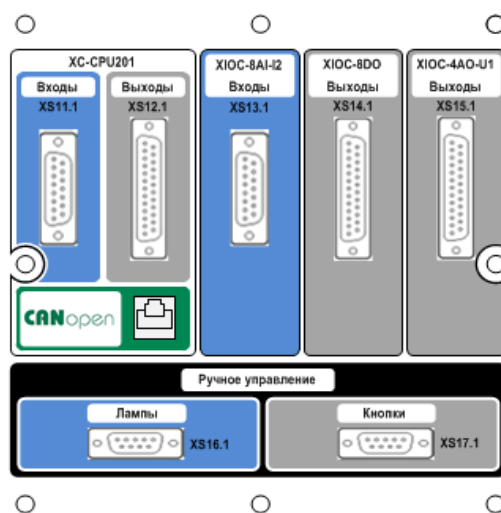


Рис. 5. Вторая интерфейсная панель разъёмов

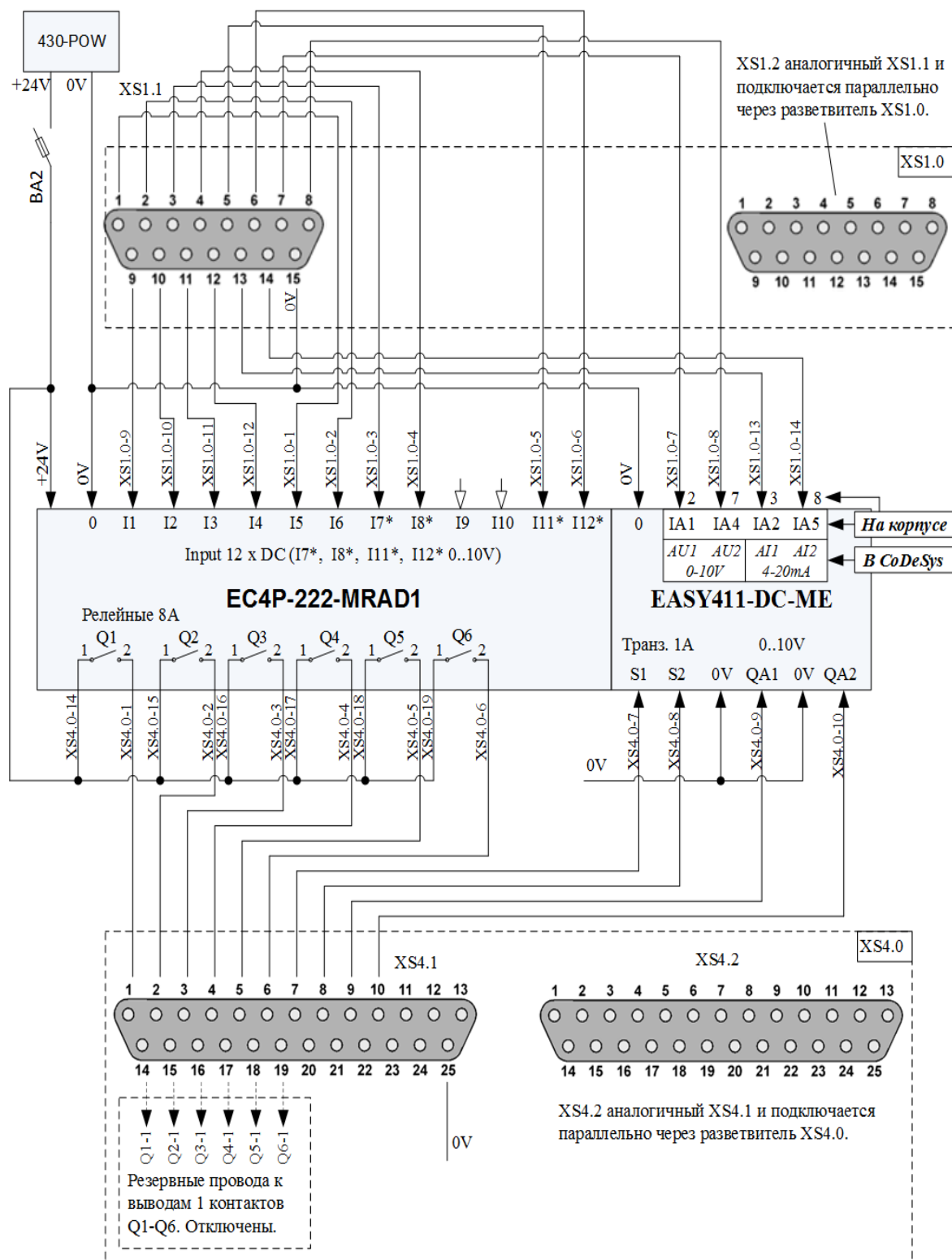


Рис. 6. Подключение контроллера EC4P-222-MRAD1 к первой панели разъемов

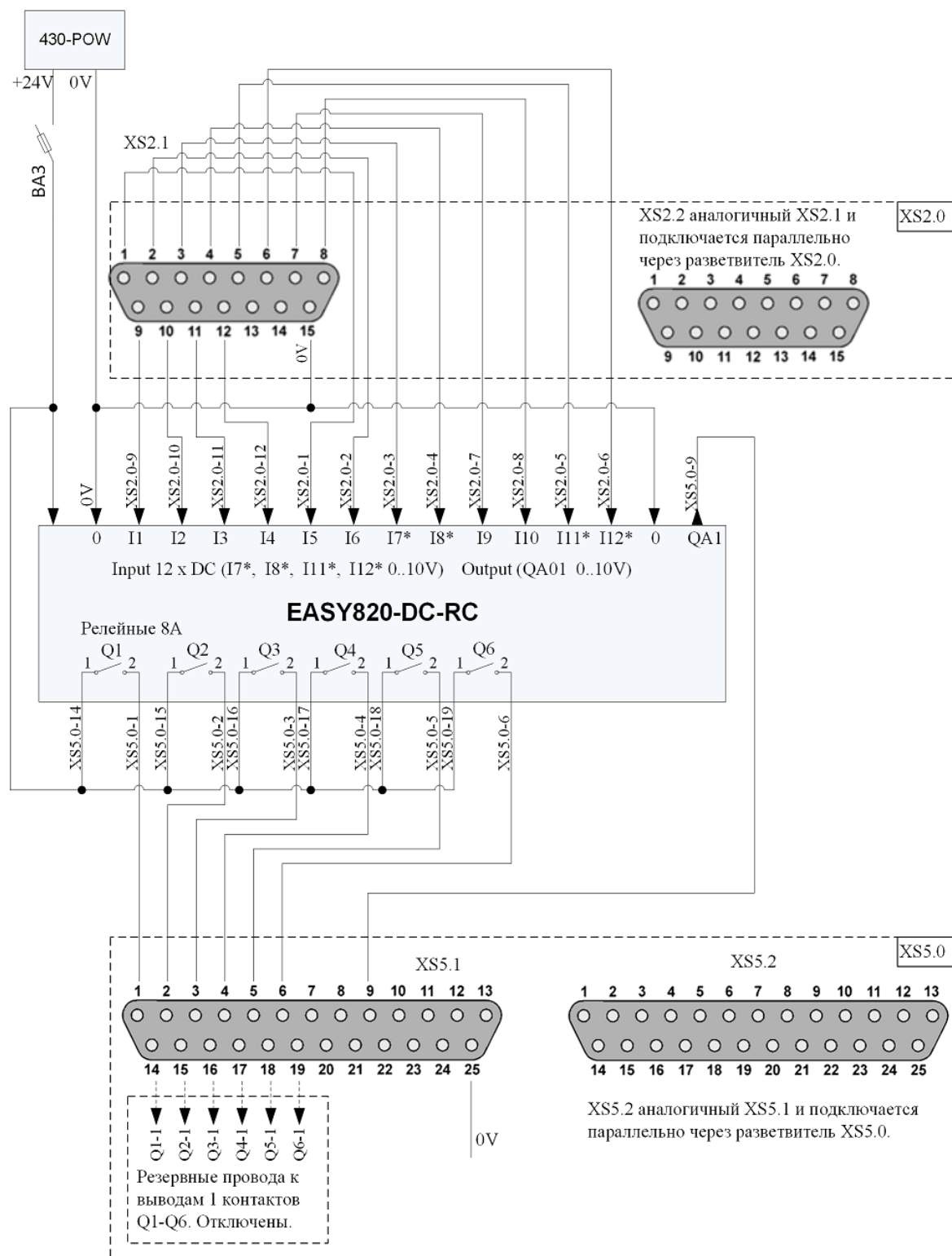


Рис. 7. Подключение реле EASY820-DC-RC к первой панели разъемов

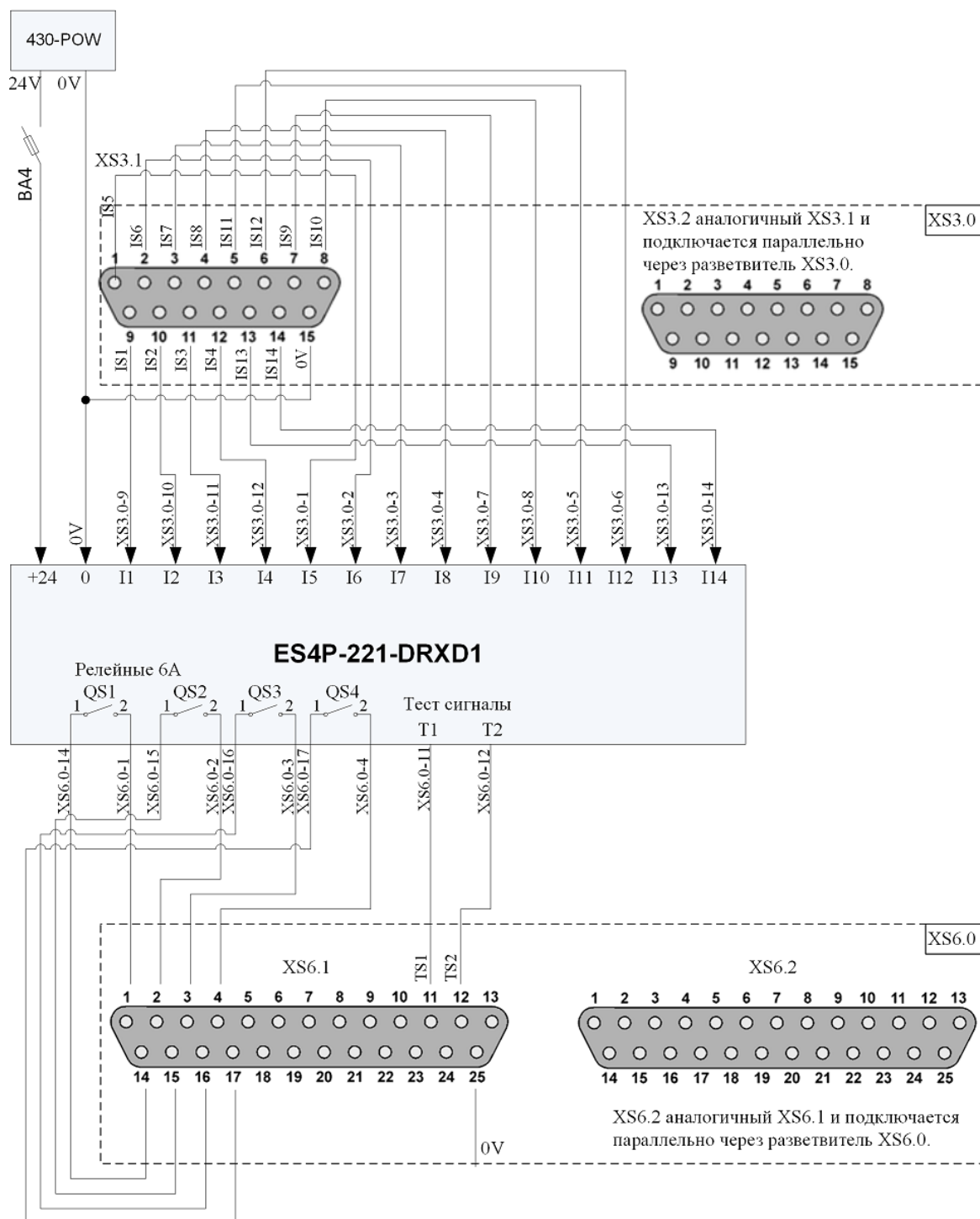


Рис.8. Подключение реле ES4P-221-DRXD1 к первой панели разъёмов

На рис. 9, 10 показаны электрические схемы соединения первой панели разъёмов с согласующими отдельными разъёмами (расположенными на задней стороне универсального стенда) для подключения к лабораторной установке.

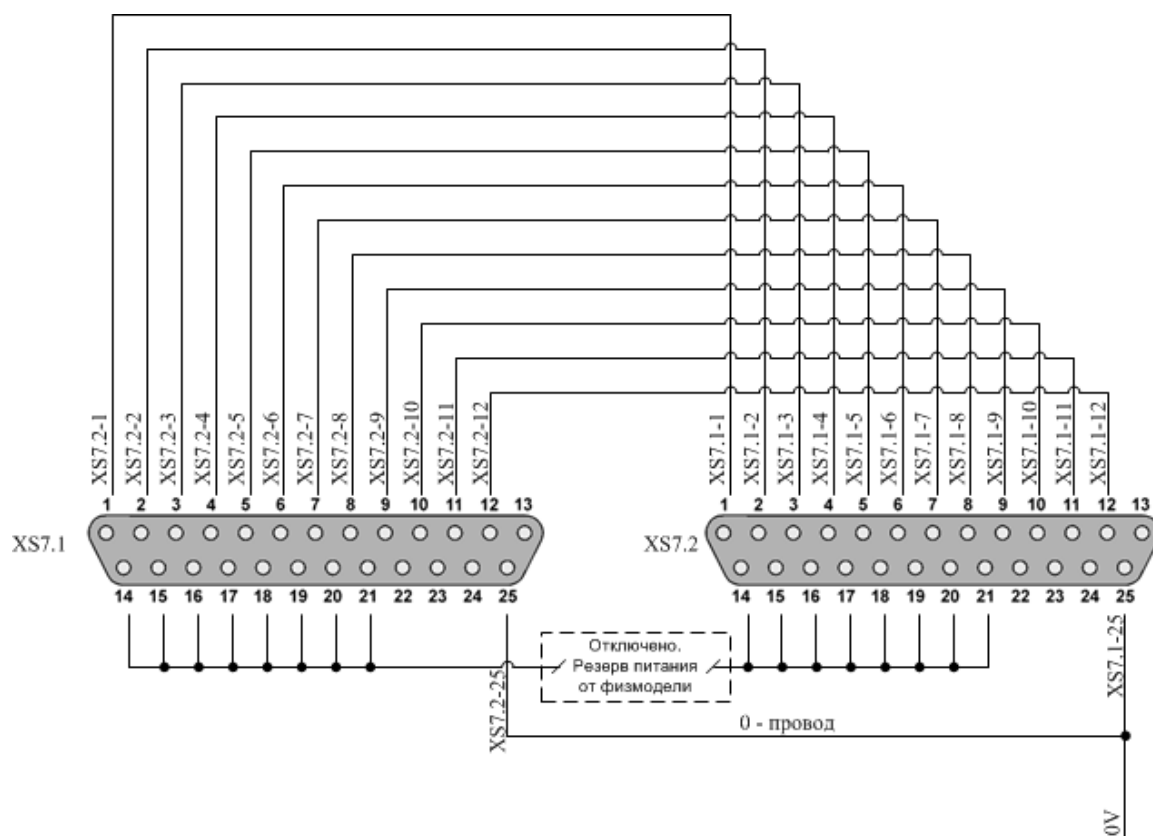


Рис. 9. Входы лабораторной установки

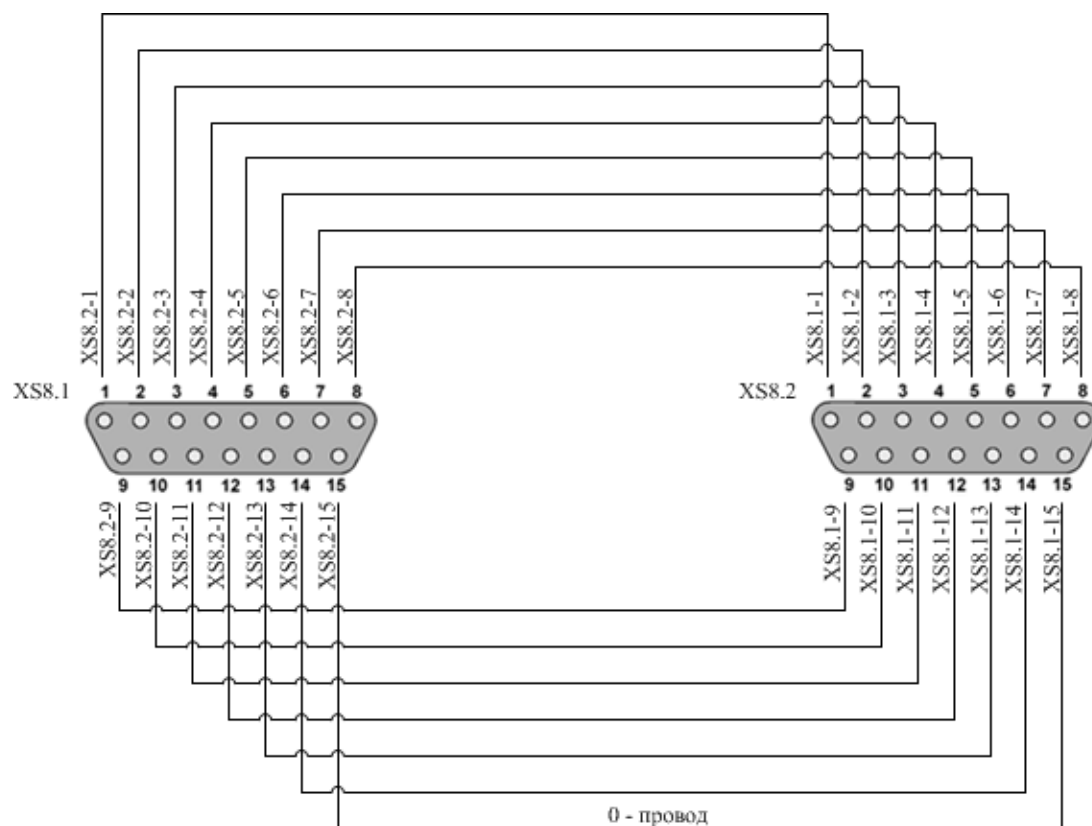


Рис. 10. Выходы лабораторной установки

На рис. 11, 12 показаны электрические схемы соединения второй панели разъемов со входами и выходами модулей контроллера XC-CPU201.

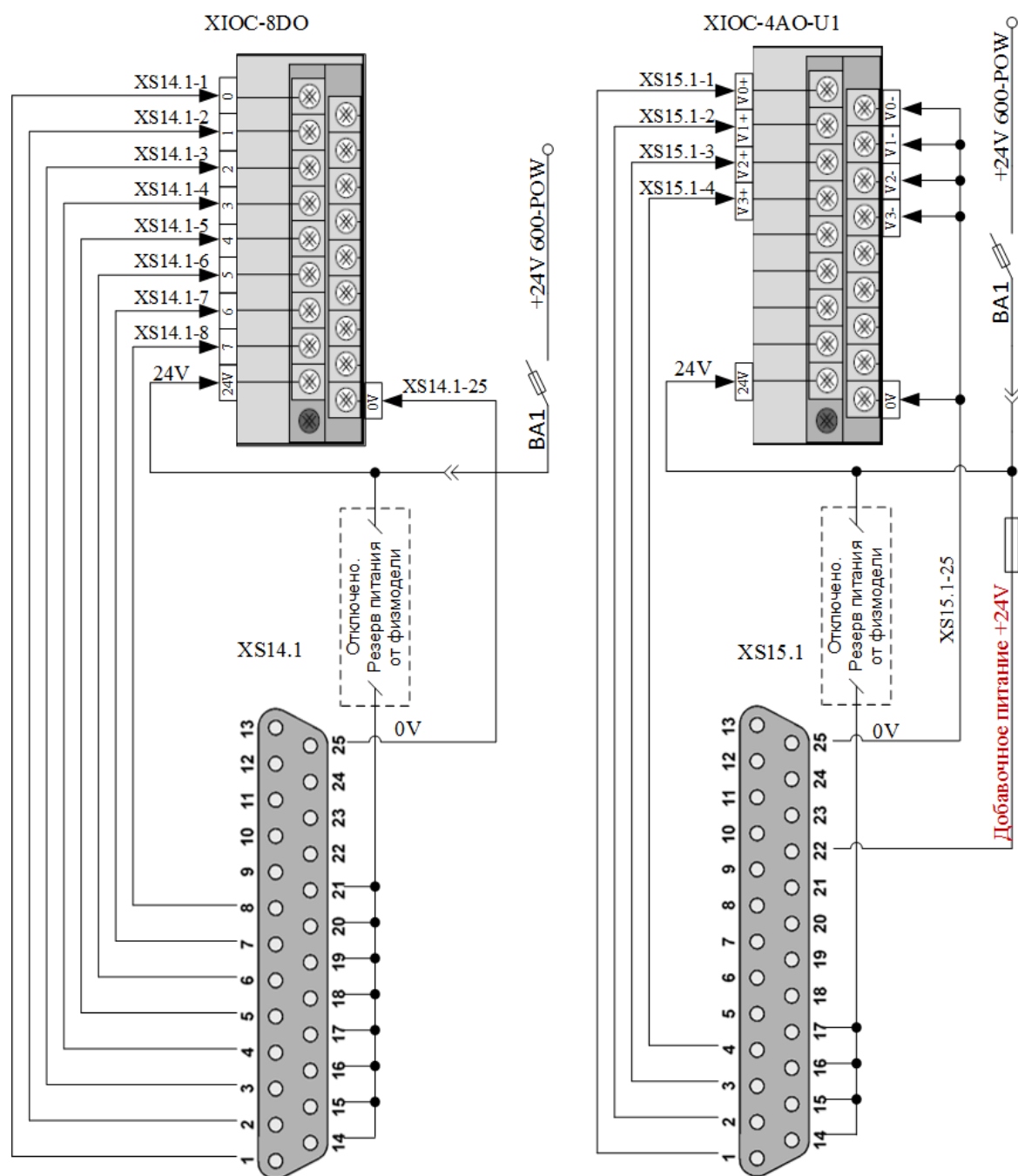


Рис. 11. Подключение выходов модулей контроллера XC-CPU201 ко второй панели разъемов

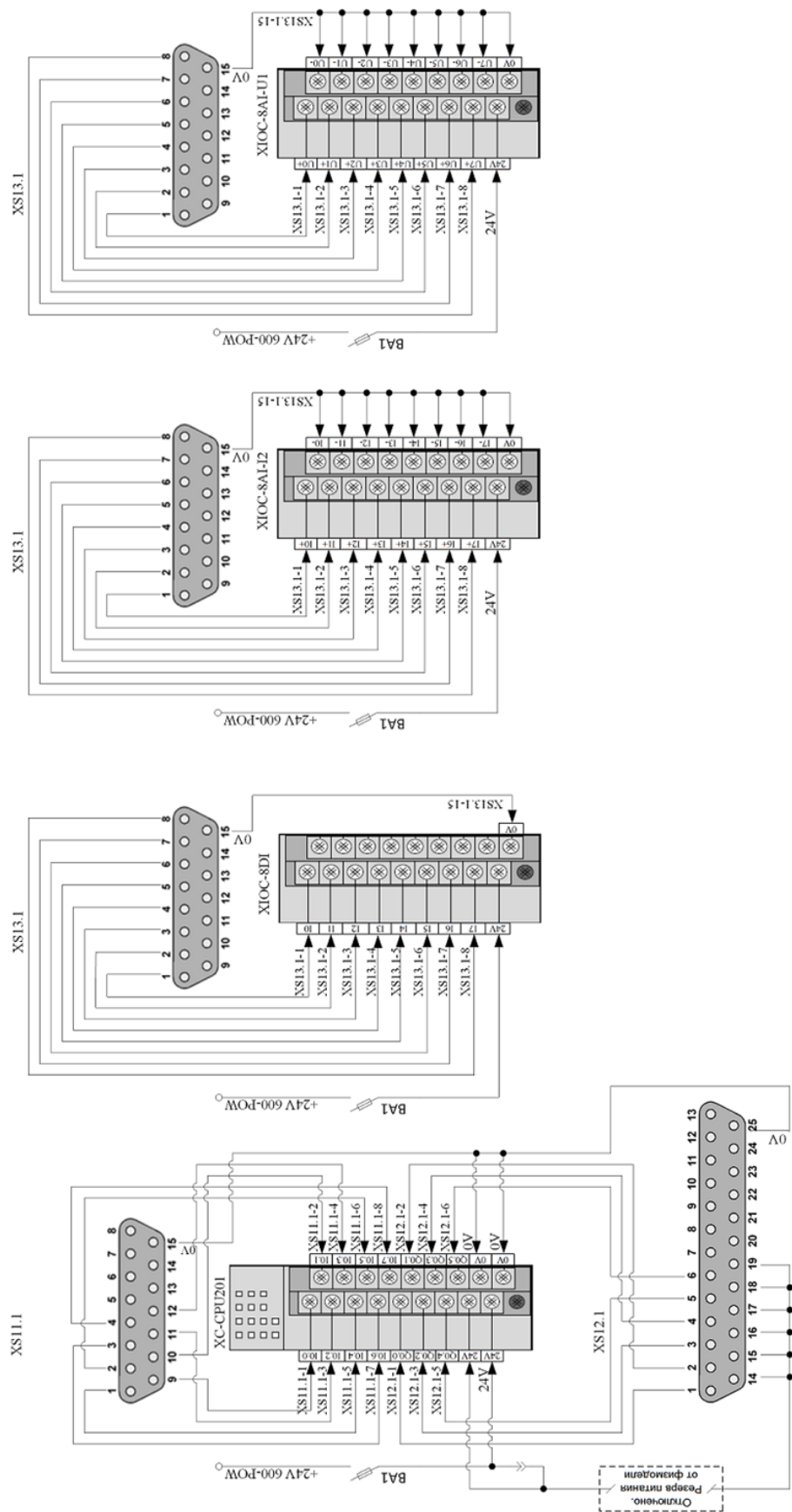


Рис. 12. Подключение модулей контроллера XC-CPU201 ко второй панели разъемов

На рис. 13 показаны электрические схемы подключения кнопок управления и сигнальных ламп ко второй панели разъемов.

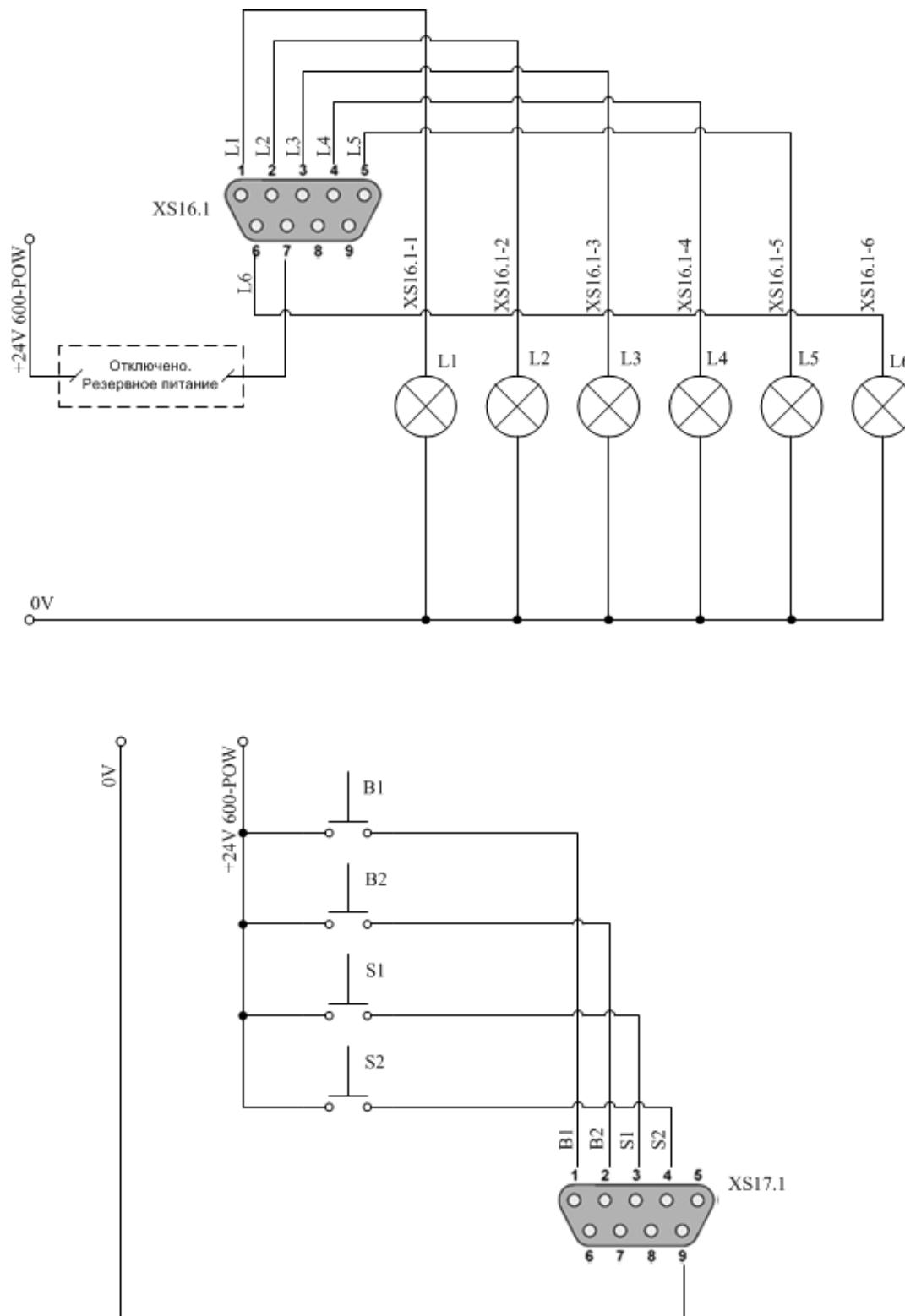
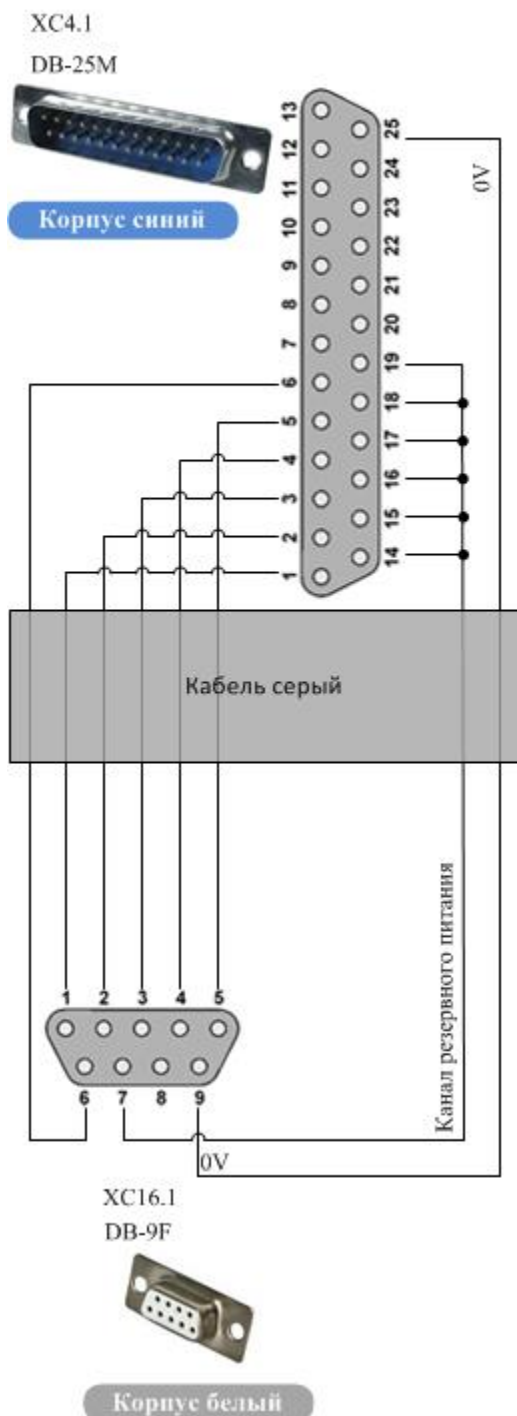


Рис. 13. Подключение элементов ручного управления ко второй панели разъемов

На рис. 14–16 показаны электрические соединения внутри универсальных кабелей, с помощью которых можно подключать входы и выходы контроллеров к лабораторной установке через первую и вторую панели разъёмов.

Кабель «Контроллер / лампы»



Кабель «Кнопки / контроллер»

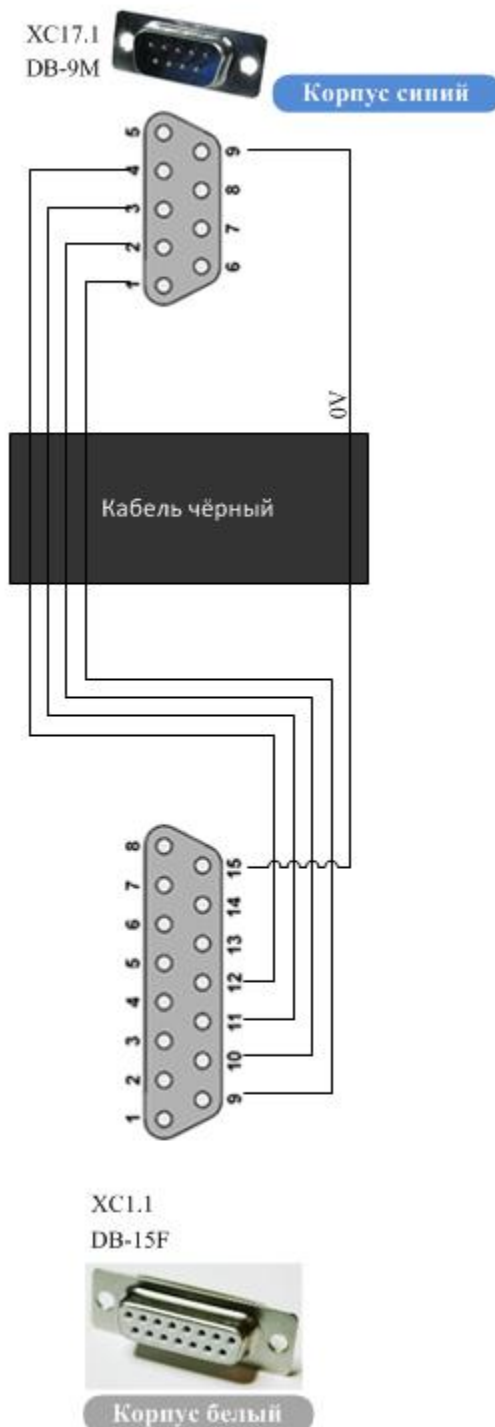


Рис. 14. Кабели ручного управления

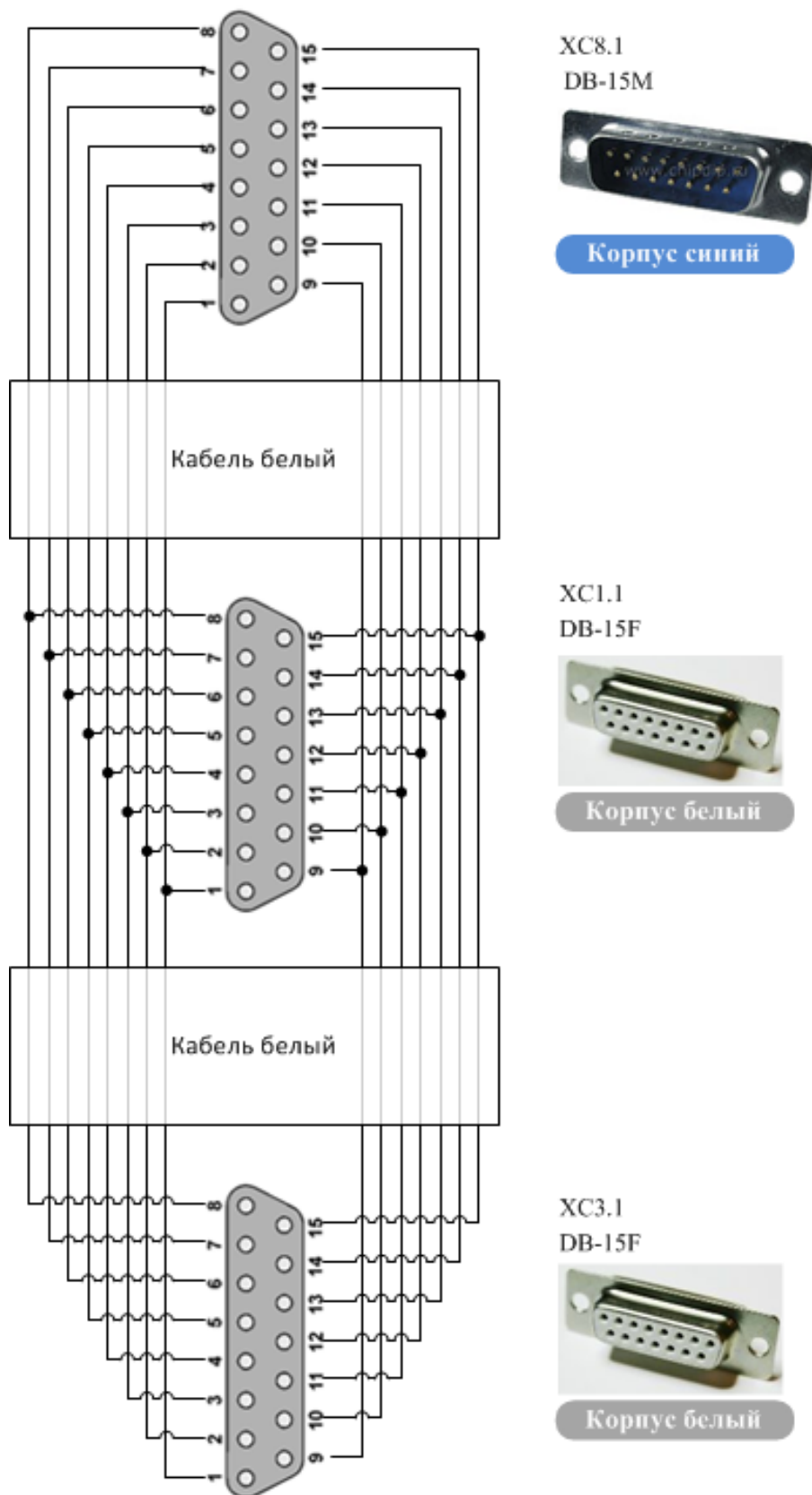


Рис. 15. Кабель подключения лабораторной установки к входам контроллера

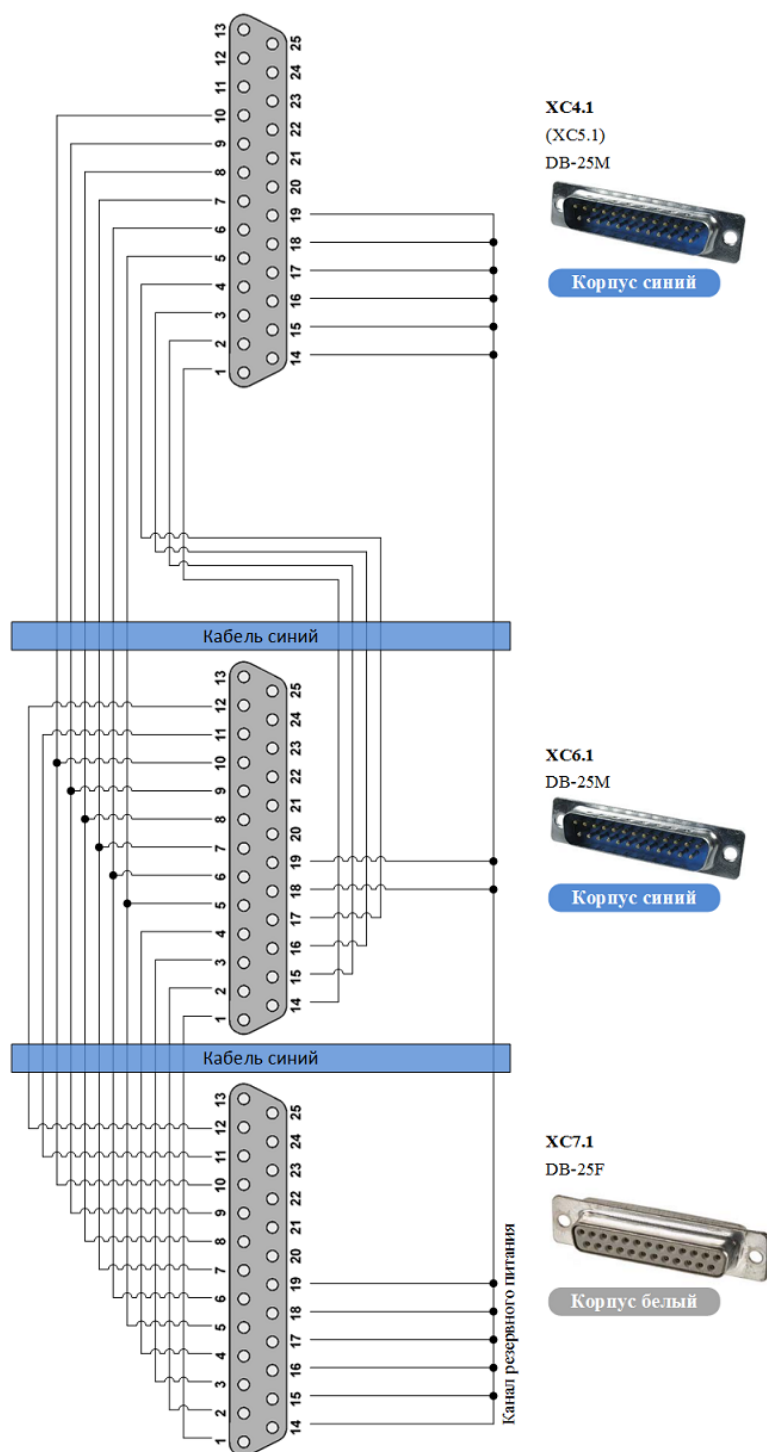


Рис. 16. Кабель подключения лабораторной установки к выходам контроллера

На рис. 17 показаны варианты подключения любого из контроллеров к лабораторной установке: напрямую (А) и через реле безопасности (В). На рис. 18 показана электрическая схема разветвителя, который предназначен

для распределения сигналов датчиков от тепло-станции ко входам модульного контроллера XC-CPU201.

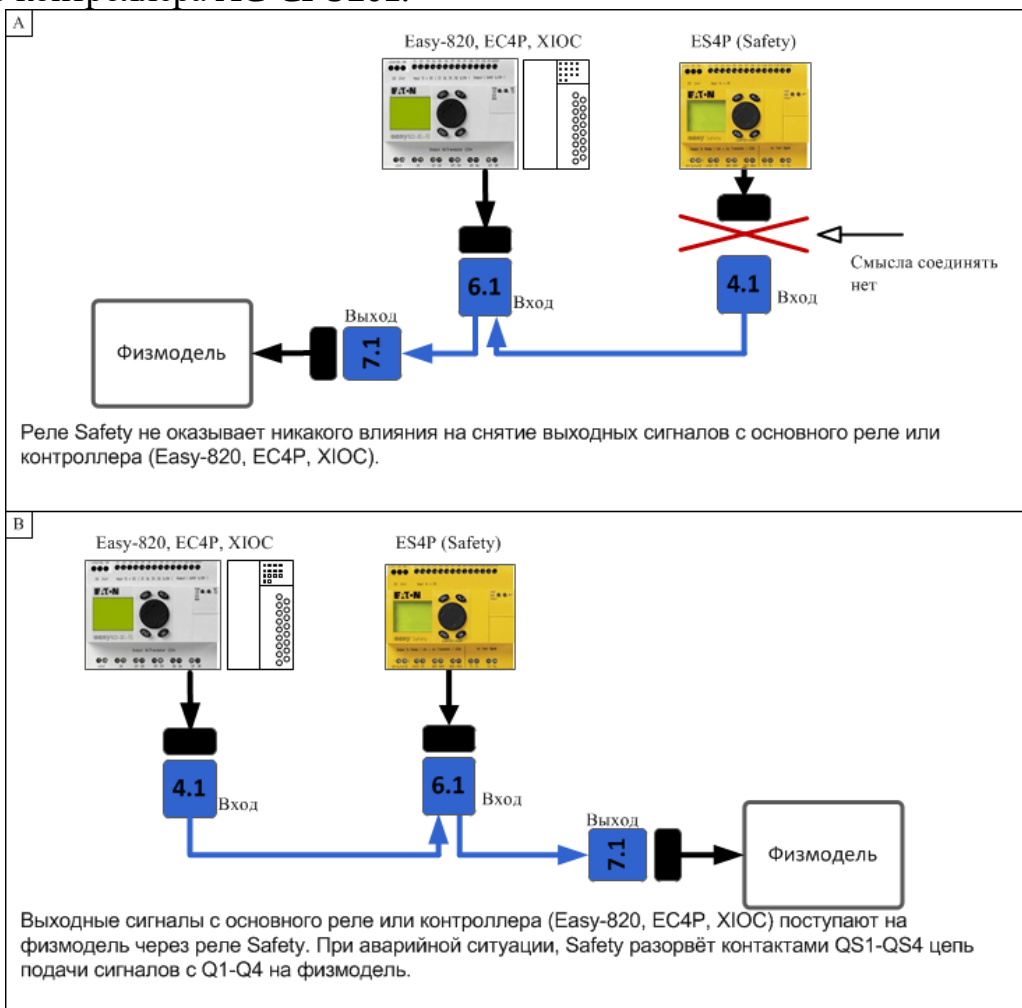


Рис. 17. Варианты подключения лабораторной установки к выходам разных контроллеров

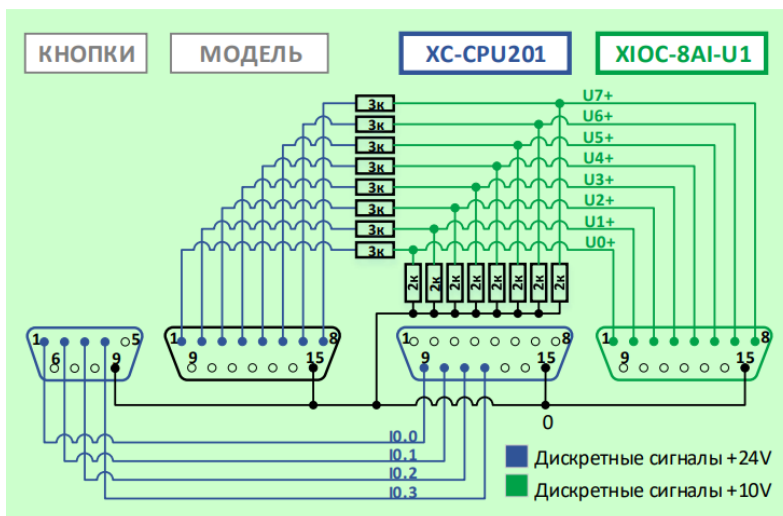


Рис. 18. Разветвитель сигналов от конвейера к модульному контроллеру

2. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Цель: научиться создавать комплекс программ для автоматизированной системы управления конвейером на разных языках программирования в средах разработки CODESYS, TRACEMODE и GALILEO.

Порядок выполнения работы

2.1. Изучение лабораторной установки «Конвейер»

Физическая модель конвейера представляет собой установку с транспортной лентой, приводимой в движение ведущим барабаном, который, в свою очередь, вращается от приводного двигателя. Над лентой расположена оптическая система для контроля габаритов транспортируемых грузов и их высоты. В конце транспортного пути расположены две корзины, предназначенные для приёма как высоких, так и низких грузов, которые сбрасываются поворотной площадкой. Выбор корзины для сброса груза осуществляется системой управления. Под каждой корзиной имеются датчики веса грузов (рис. 19).

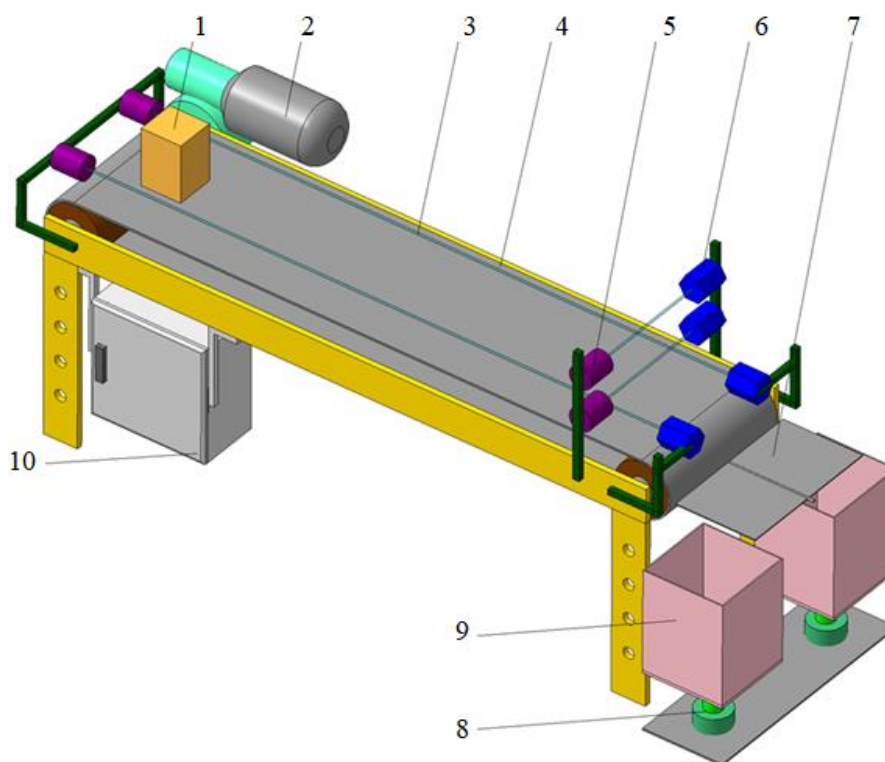


Рис. 19. Абстрактная модель конвейера:

- 1 – транспортируемый груз; 2 – электродвигатель асинхронный;
- 3 – инфракрасный луч; 4 – лента; 5 – источник света; 6 – оптический датчик;
- 7 – направляющая поворотная площадка для спуска грузов в корзины;
- 8 – датчики перемещения для измерения веса грузов; 9 – приёмные корзины для сбора грузов; 10 – силовой шкаф с электрооборудованием

Внешний вид установки конвейера приведён на рис. 20.



Рис. 20. Лабораторная установка конвейера

Система управления конвейером выполняет следующие задачи:

1. Контроль позиционирования грузов по ширине ленты. Реализуется с помощью инфракрасных лучей, направленных вдоль ленты.
2. Контроль заданной высоты транспортируемых грузов за счёт использования двух световых лучей на двух разных высотах поперёк направления движения ленты.
3. Контроль наполнения корзин грузами с помощью двух датчиков перемещения.
4. Контроль скорости ленты с помощью оптического датчика на ведущем барабане. Датчик подсчитывает число свето-импульсов через вращающееся зеркало, измеряя также и длительность между импульсами.

В табл. 1 и 2 приведены способы подключения датчиков и исполнительных элементов к входам и выходам программируемых аппаратов универсального стенда в соответствии с электрической схемой (рис. 21).

Таблица 1

*Варианты подключения датчиков к различным аппаратам
универсального стенда*

Назначение	Контакт на разъёме DB-15	Поступление на входы				
		XC-CPU201	XIOC-8AI-U1	<i>EASY820</i>	<i>ES4P</i>	<i>EC4P (EASY411)</i>
Левый габарит	1	–	U0+	<i>I5</i>	<i>I5</i>	<i>I5</i>
Правый габарит	2	–	U1+	<i>I6</i>	<i>I6</i>	<i>I6</i>
Высокий груз	3	–	U2+	<i>I7*</i>	<i>I7*</i>	<i>I7*</i>
Низкий груз	4	–	U3+	<i>I8*</i>	<i>I8*</i>	<i>I8*</i>
Вес левой корзины	5	–	U4+	<i>I11*</i>	<i>I11*</i>	<i>I11*</i>
Вес правой корзины	6	–	U5+	<i>I12*</i>	<i>I12*</i>	<i>I12*</i>
Ведущий вал	7	–	U6+	<i>I9</i>	<i>I9</i>	<i>(IA1)</i>
Пуск двигателя	9	I0.0	–	<i>I1</i>	<i>I1</i>	<i>I1</i>
Останов двигателя	10	I0.1	–	<i>I2</i>	<i>I2</i>	<i>I2</i>
Скорость ниже	11	I0.2	–	<i>I3</i>	<i>I3</i>	<i>I3</i>
Скорость выше	12	I0.3	–	<i>I4</i>	<i>I4</i>	<i>I4</i>
<i>Общий провод</i>	15	0	0	<i>0</i>	<i>0</i>	<i>0</i>

Таблица 2

Подключение исполнительных элементов к различным аппаратам

Назначение	Контакт на разъёме DB-25	Управление с выходов					
		XC-CPU201	<i>XIOC-8DO</i>	<i>XIOC-4AO-U1</i>	<i>EASY820</i>	<i>ES4P</i>	<i>EC4P</i> (<i>EASY411</i>)
Прямой пуск	1	Q0.0	<i>0</i>	–	<i>Q1</i>	<i>QS1</i>	<i>Q1</i>
Сброс груза влево	2	Q0.1	<i>1</i>	–	<i>Q2</i>	<i>QS2</i>	<i>Q2</i>
Сброс груза вправо	3	Q0.2	<i>2</i>	–	<i>Q3</i>	<i>QS3</i>	<i>Q3</i>
Двоичное задание частоты 001	4	Q0.3	<i>3</i>	–	<i>Q4</i>	<i>QS4</i>	<i>Q4</i>
Двоичное задание частоты 010	5	Q0.4	<i>4</i>	–	<i>Q5</i>	–	<i>Q5</i>
Двоичное задание частоты 100	6	Q0.5	<i>5</i>	–	<i>Q6</i>	–	<i>Q6</i>
Включение ПЧ дистанционно	7	–	<i>6</i>	–	–	–	(<i>S1</i>)
–	8	–	–	–	–	–	–
Аналоговое задание частоты	9	–	–	<i>V0+</i>	<i>QA1</i>	–	(<i>QA1</i>)
Аналоговое ПИД-управление	10	–	–	<i>V1+</i>	–	–	(<i>QA2</i>)
Общий провод	25	0	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>

2.2. Общая концепция системы управления в составе OPC сервера

Общий принцип работы подобной системы управления состоит в следующем:

- Передача сигналов управления (пакетов данных) осуществляются со стороны SCADA системы на персональном компьютере в сторону OPC сервера для записи значения управляющего сигнала в символьный файл. В нашем случае OPC сервер в виде специальной программы находится в составе самого компьютера. В отдельных случаях данные сигналы управления можно передавать дистанционно в OPC сервер по протоколу TCP/IP Ethernet через специальный кабель.
- Значение управляющего сигнала из символьного файла передаётся в программируемый логический контроллер для запуска в нём определённого программного алгоритма. В нашем случае контроллер подсоединён в общую информационную сеть Ethernet с компьютером.
- Электрический сигнал управления передаётся со стороны контроллера в сторону исполнительных органов лабораторной установки конвейера.
- Передача сигналов с датчиков лабораторной установки конвейера осуществляется в обратном порядке в SCADA систему.
- Дополнительным звеном управления в системе является HMI панель, которая соединена с контроллером специальным кабелем для двусторонней передачи данных по сетевому протоколу CAN Open. В данном случае обмен данных производится только между контроллером и самой панелью. OPC сервер напрямую здесь не участвует. Но принятые контроллером сигналы управления от HMI панели могут поступать далее в OPC сервер уже по протоколу TCP/IP Ethernet, попадая в символьный файл и далее в SCADA систему.

Общая концепция работы данной системы управления в виде структурной схемы приведена на рис. 22.

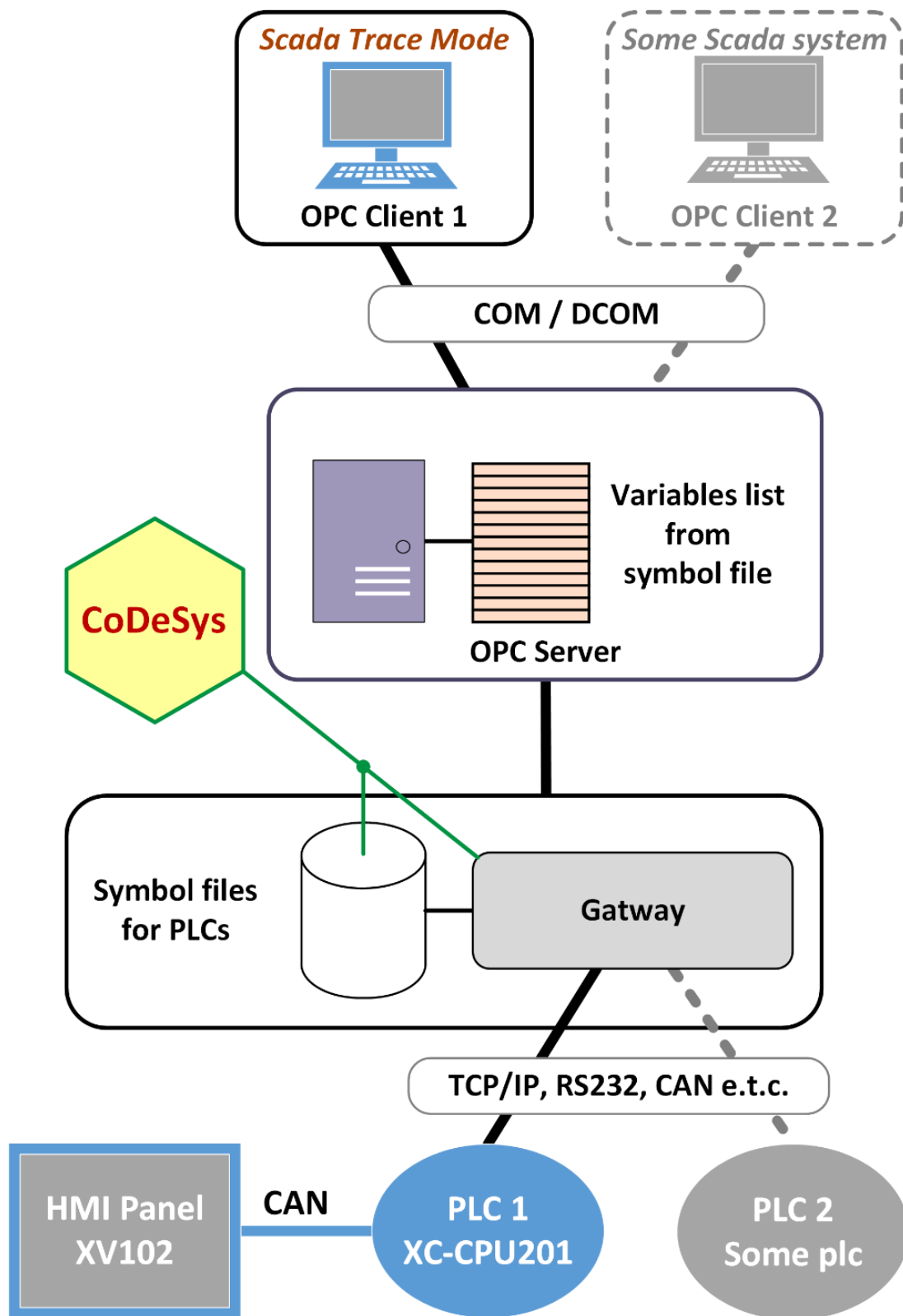


Рис. 22. Общая концепция системы управления

На рис. 23 показан внешний вид универсального учебного стенда, на борту которого находится такое необходимое для системы управления оборудование, как программируемый логический контроллер XC-CPU201, HMI панель серии XV102, кнопки управления, персональный компьютер, панели разъёмов.



Рис. 23. Внешний вид учебного стенда

Всё программируемое оборудование данного стенда подключено в общую информационную сеть Ethernet по рис. 3. Это в первую очередь необходимо для загрузки программных алгоритмов во внутреннюю память оборудования со стороны персонального компьютера.

2.3. Создание программы на языках программирования CFC и FBD в среде CODESYS для контроллера XC-CPU201

1. Запустите программную среду CODESYS 2.3.9.
2. Создайте новый проект: «Файл/Создать» (File/New).
3. Настройте целевую платформу (Target Settings). Откройте вкладку «Целевая платформа» (Target Settings) и на странице диалогового окна «Конфигурация» (Configuration) выберите тип контроллера **XC-CPU201-EC512K-8DI-6DO-XV V2.3.9 SP8**. Во вкладке основных настроек (General) должна стоять галочка напротив «Download symbol file». Во вкладке сетевой функциональности (Network functionality) поставьте галочку на против «Support network variables» и в открывшемся поле ввода пропишите «CAN» (рис. 24). Подтвердите ввод нажатием кнопки ОК.

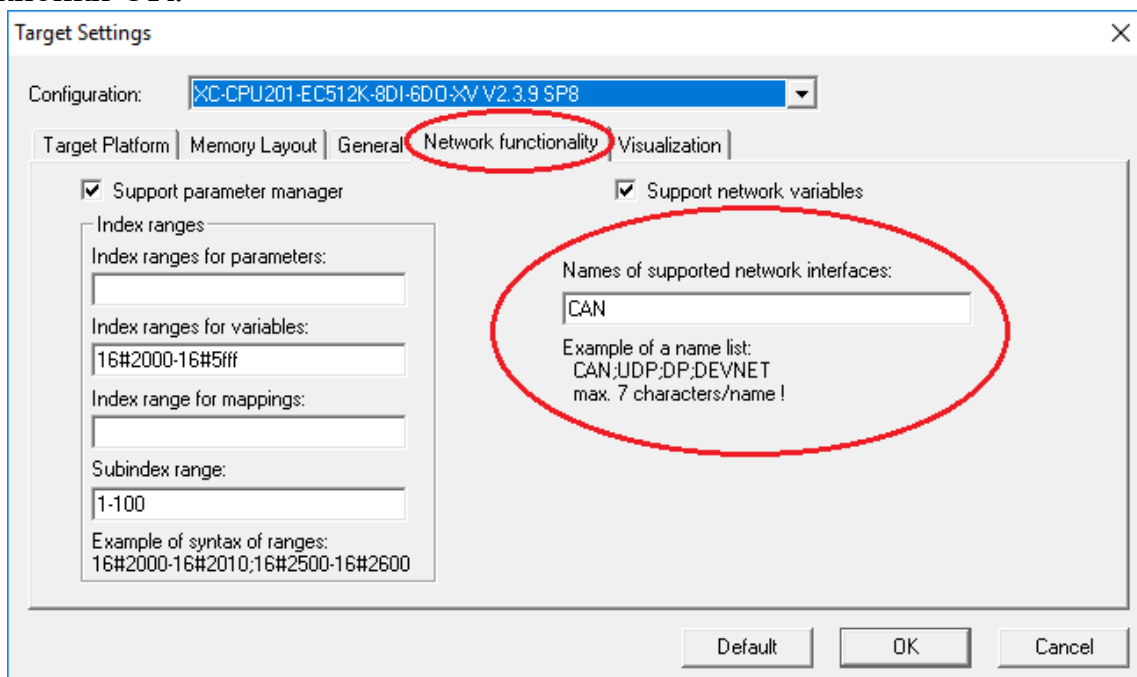


Рис. 24. Установка поддержки сетевой передачи данных через CAN Open протокол

4. Создайте главную программу. После настройки целевой платформы в автоматически появившемся диалоговом окне New POU выберите тип компонента «Программа» (Program), язык CFC, задайте имя программы PLC_PRG.
5. Настройте конфигурацию контроллера. Зайдите во вкладку ресурсов «Resources» проекта и откройте конфигурацию контроллера «PLC Configuration» (рис. 25). Далее нажмите правой клавишей мыши по верхнему уровню меню (Configuration XC-CPU201...) и добавьте «CanMaster».

Нажмите внутри опции «Non Display[SLOT]» по «EMPTY-SLOT» правой клавишей мыши и выберите элемент «XIOC-8AI-U1» - это модуль расширения для приёма сигналов 10В, но уже преобразованных из 24В от датчиков конвейера.

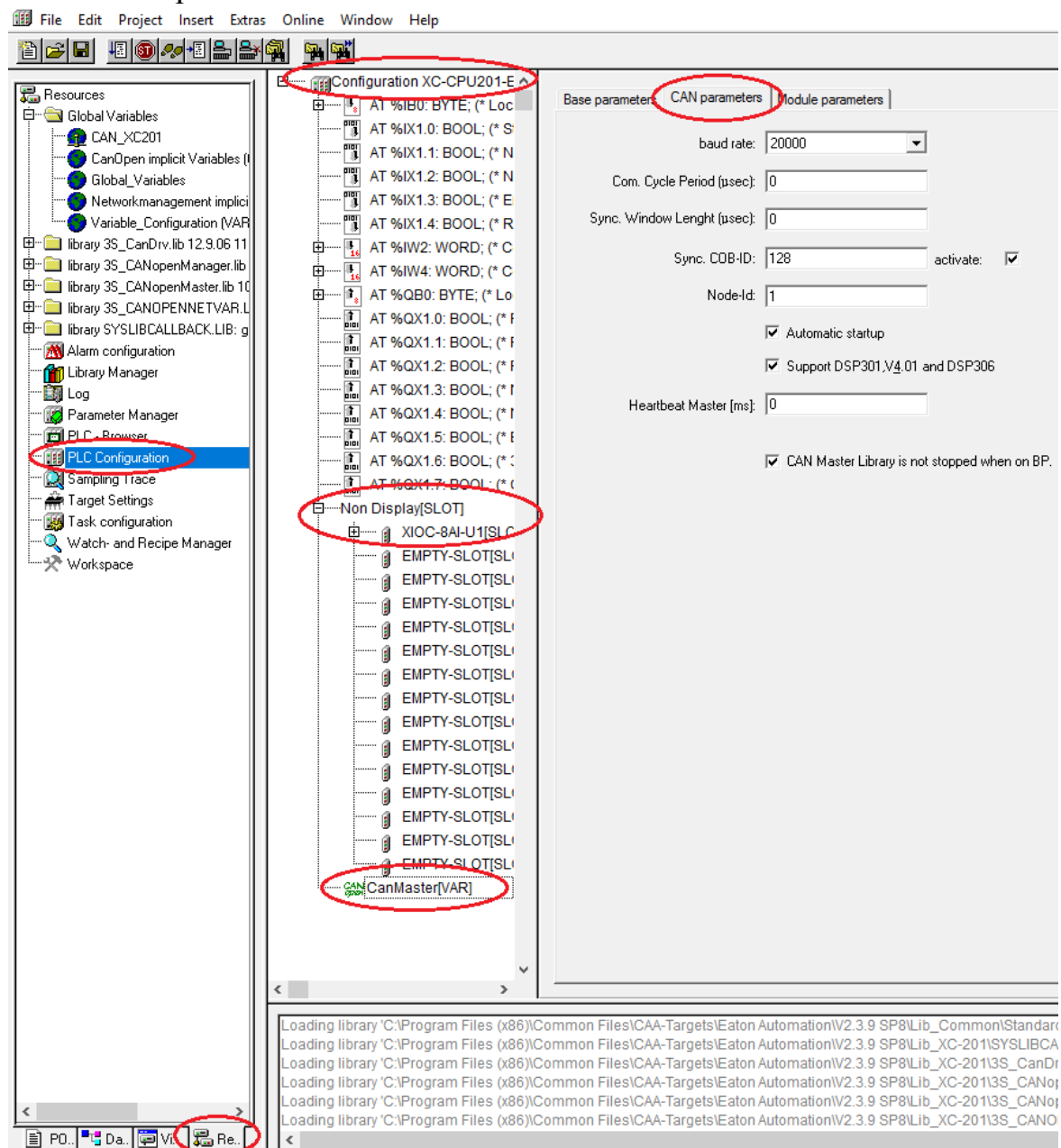


Рис. 25. Конфигурация контроллера

Общая структура главной программы на языке CFC показана на рис. 26, а её переменные – на рис. 27. Чтобы создать данную программу, необходимо сначала создать дополнительные функциональные блоки, а затем соединить их между собой.

Предварительно разберём принцип работы данной программы.

Запуск двигателя производится с помощью блока start_motor путём подачи от кнопки пуска на вход start сигнала TRUE, при условии, что на контрольные входы slowdown, left_dimension, right_dimension и full_weight поступают сигналы FALSE.

Разберём остальные блоки, подающие сигналы на контрольные входы start_motor. Блок slowdown сравнивает скорость вращения ведущего барабана с минимальной уставкой. На вход этого блока поступает сигнал от блока velocity_measure о величине скорости этого барабана. Последний блок вычисляет скорость вращения барабана по интервалу между импульсами от датчика velo_sensor. Таким образом, контролируется снижение скорости из-за возможной неисправности двигателя.

Сигнал о наполнении любой из корзин поступает от переменных weight_high и weight_low через промежуточные блоки инерционной задержки weight_high_inertial и weight_low_inertial на входы аварийного контроля в блок start_motor.

Приведение в движение поворотной площадки осуществляется с помощью блока stage_check совместно с блоками movie_stage и movie_stage2. При этом подаются сигналы в переменные movie_high и movie_low, далее с физических выходов %QX0.1 и %QX0.2, связанных с этими переменными, поступают сигналы на привод поворотной площадки сброса грузов.

Сигнал запуска двигателя поступает от переменной work_motor, связанной с физическим выходом %QX0.0.

Из переменных Q4, Q5, Q6, связанных с физическими выходами %QX0.3, %QX0.4, %QX0.5, подаётся сигнал на специальные входы преобразователя частоты для задания скорости вращения двигателя.

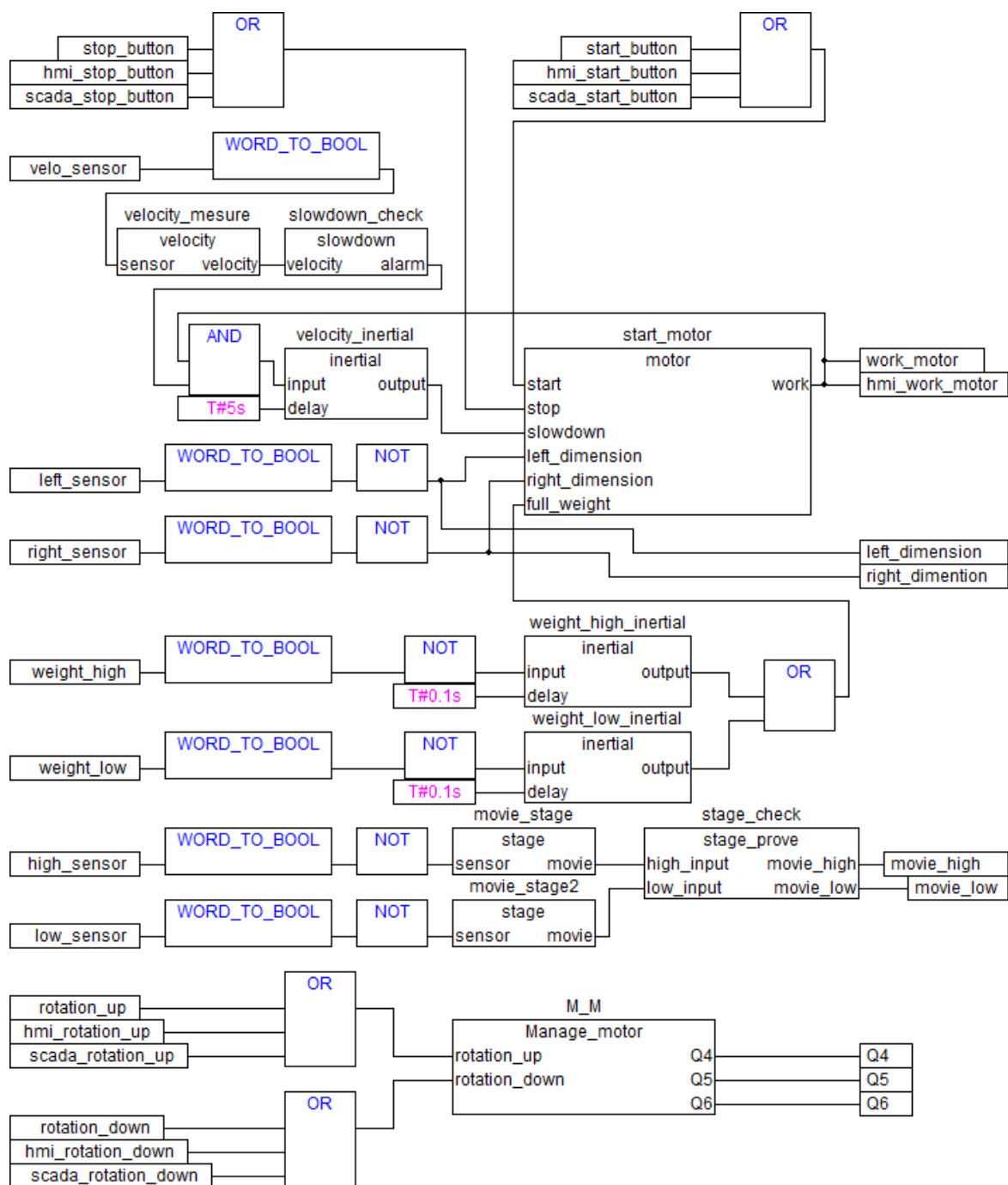


Рис. 26. Корневая программа PLC_PRG

6. Объявите переменные корневой программы PLC_PRG. Пропишите переменные в соответствии с рис. 27. Сделайте привязку переменных к физическим входам и выходам контроллера с помощью оператора «АТ».

Относится к блоку расширения XIOC-8AI-U1, который добавляется в Resources/PLC Configuration (Ресурсы/Конфигурация ПЛК)

```

0001 PROGRAM PLC_PRG
0002 VAR
0003   (* XC-CPU201 *)
0004   start_button AT %IX0.0: BOOL; (* I0.0 *)
0005   stop_button AT %IX0.1: BOOL; (* I0.1 *)
0006   rotation_up AT %IX0.2: BOOL; (* I0.2 *)
0007   rotation_down AT %IX0.3: BOOL; (* I0.3 *)
0008   work_motor AT %QX0.0: BOOL; (* Q0.0 *)
0009   movie_high AT %QX0.1: BOOL; (* Q0.1 *)
0010   movie_low AT %QX0.2: BOOL; (* Q0.2 *)
0011   Q4 AT %QX0.3: BOOL; (* Q0.3 *)
0012   Q5 AT %QX0.4: BOOL; (* Q0.4 *)
0013   Q6 AT %QX0.5: BOOL; (* Q0.5 *)
0014
0015   (* XIOC-8AI-U1 *)
0016   left_sensor AT %IW6: WORD; (* U0+ *)
0017   right_sensor AT %IW8: WORD; (* U1+ *)
0018   high_sensor AT %IW10: WORD; (* U2+ *)
0019   low_sensor AT %IW12: WORD; (* U3+ *)
0020   weight_high AT %IW14: WORD; (* U4+ *)
0021   weight_low AT %IW16: WORD; (* U5+ *)
0022   velo_sensor AT %IW18: WORD; (* U6+ *)
0023
0024   (* Local variables *)
0025   left_dimension: BOOL;
0026   right_dimention: BOOL;
0027
0028   (* Functional blocks *)
0029   velocity_mesure: velocity;
0030   slowdown_check: slowdown;
0031   movie_stage: stage;
0032   movie_stage2: stage;
0033   stage_check: stage_prove;
0034   start_motor: motor;
0035   weight_high_inertial: inertial;
0036   weight_low_inertial: inertial;
0037   velocity_inertial: inertial;
0038   M_M: Manage_motor;
0039
0040   (* Inputs from scada *)
0041   scada_start_button: BOOL;
0042   scada_stop_button: BOOL;
0043   scada_rotation_up: BOOL;
0044   scada_rotation_down: BOOL;
0045 END_VAR

```

Рис. 27. Объявление переменных программы PLC_PRG

Вообще, чтобы определить, как правильно прописывать в такой ситуации физические входы и выходы, необходимо в разделе «Ресурсы» (Resources) открыть вкладку «Конфигурация контроллера» (PLC Configuration) и в диалоговом окне выбрать опцию Configuration XC-CPU201, а в ней – AT %IB0: BYTE (Local Inputs) и AT %QB0: BYTE (Local Outputs).

Объявленные переменные имеют следующие значения:

- start_button AT %IX0.0: BOOL – переменная, принимающая сигнал от кнопки пуска двигателя;
- stop_button AT %IX0.1: BOOL – переменная, принимающая сигнал от кнопки остановки двигателя;
- rotation_up AT %IX0.2: BOOL – переменная, принимающая сигнал от кнопки увеличения частоты вращения двигателя;
- rotation_down AT %IX0.3: BOOL – переменная, принимающая сигнал от кнопки уменьшения частоты вращения двигателя;
- work_motor AT %QX0.0: BOOL – переменная, подающая сигнал для пуска двигателя;
- movie_high AT %QX0.1: BOOL – переменная, подающая сигнал для поворота площадки в сторону корзины для высоких грузов;
- movie_low AT %QX0.2: BOOL – переменная, подающая сигнал для поворота площадки в сторону корзины для низких грузов;
- Q4 AT %QX0.3: BOOL – переменная, подающая сигнал первого регистра для задания частоты вращения двигателя через ПЧ;
- Q5 AT %QX0.4: BOOL – переменная, подающая сигнал второго регистра для задания частоты вращения двигателя через ПЧ;
- Q6 AT %QX0.5: BOOL – переменная, подающая сигнал третьего регистра для задания частоты вращения двигателя через ПЧ;
- left_sensor AT %IW6: WORD – переменная, принимающая сигнал перекрытия инфракрасного луча левого габарита;
- right_sensor AT %IW8: WORD – переменная, принимающая сигнал перекрытия инфракрасного луча правого габарита;
- high_sensor AT %IW10: WORD – переменная, принимающая сигнал от датчика высоких грузов;
- low_sensor AT %IW12: WORD – переменная, принимающая сигнал от датчика низких грузов;
- weight_high AT %IW14: WORD – переменная, принимающая сигнал от датчика веса под корзиной для высоких грузов;
- weight_low AT %IW16: WORD – переменная, принимающая сигнал от датчика веса под корзиной для низких грузов;
- velo_sensor AT %IW18: WORD – переменная, принимающая импульсы от фотодатчика для ведущего барабана;
- left_dimension: BOOL – переменная выхода груза за левый габарит ленты конвейера;
- right_dimention: BOOL – переменная выхода груза за правый габарит ленты конвейера;
- velocity_mesure: velocity – локальное в PLC_PRG имя функционального блока velocity;

- slowdown_check: slowdown – локальное в PLC_PRG имя функционального блока slowdown;
- movie_stage: stage – локальное в PLC_PRG имя функционального блока stage; movie_stage2 – то же;
- stage_check: stage_prove – локальное в PLC_PRG имя функционального блока stage_prove;
- start_motor: motor – локальное в PLC_PRG имя функционального блока motor;
- weight_height_inertial: inertial – локальное в PLC_PRG имя функционального блока inertial;
- weight_low_inertial: inertial – локальное в PLC_PRG имя функционального блока internal;
- velocity_inertial: inertial – локальное в PLC_PRG имя функционального блока internal;
- M_M: Manage_motor – локальное в PLC_PRG имя функционального блока Manage_motor;
- scada_start_button – переменная приёма сигнала от Scada системы для запуска двигателя;
- scada_stop_button – переменная приёма сигнала от Scada системы для останова двигателя;
- scada_rotation_up – переменная приёма сигнала от Scada системы для увеличения скорости вращения двигателя;
- scada_rotation_down – переменная приёма сигнала от Scada системы для уменьшения скорости вращения двигателя.

7. *Объявите глобальные переменные.* Пропишите их в разделе Global Variables, как показано на рис. 28. Эти переменные будут использоваться в остальных функциональных блоках и имеют следующее назначение:

- minimum_velocity – минимальная скорость ленты;
- sensor_way – геометрический путь по длине окружности между зеркалами на барабане конвейера;
- interval_init – время удержания поворотной площадки в наклонённом положении, чтобы после перекрытия инфракрасного луча грузом он успел упасть в корзину.

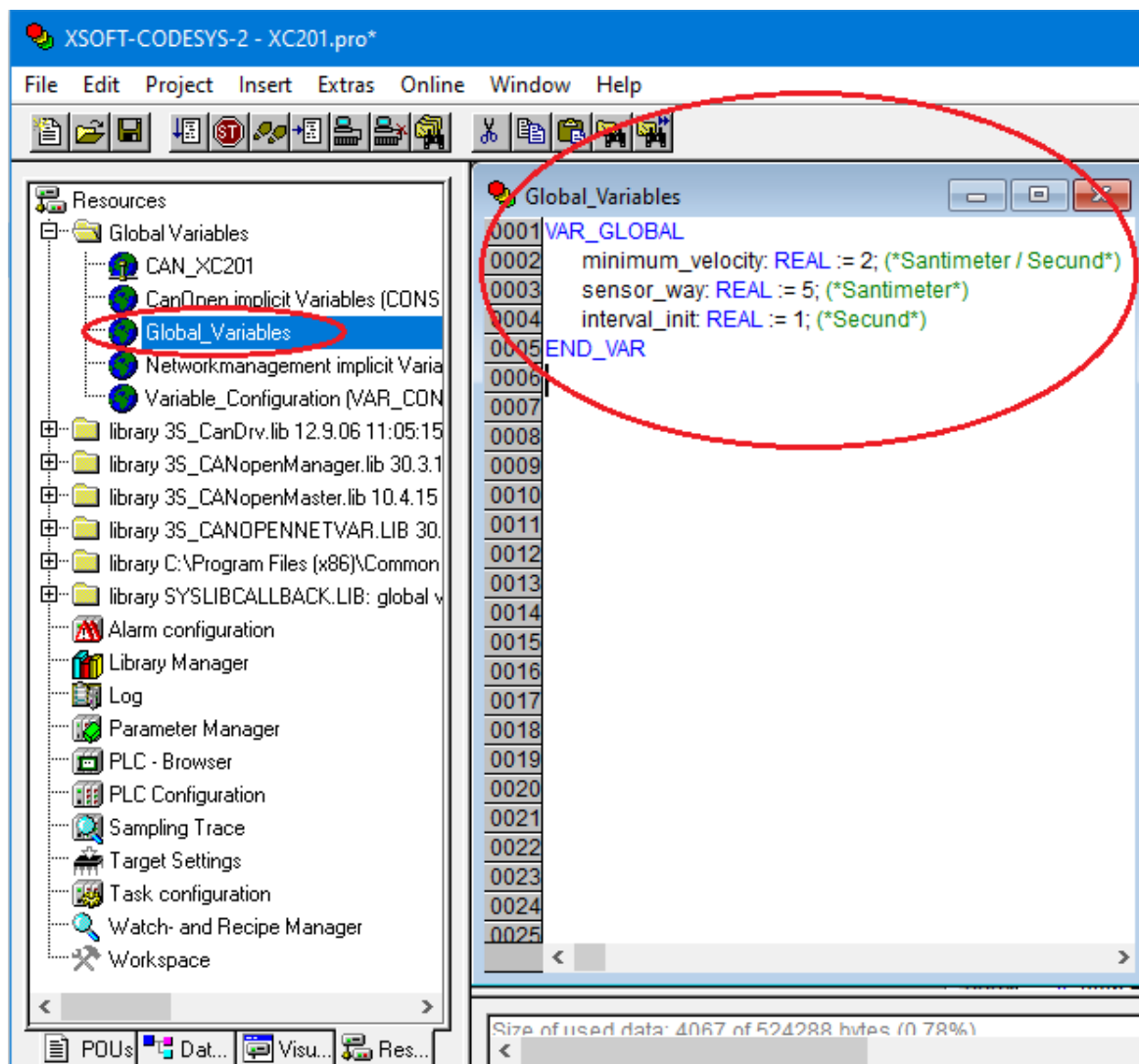


Рис. 28. Глобальные переменные

8. *Создайте дополнительные функциональные блоки.*
 - *Создайте функциональный блок запуска двигателя.* В первой системной вкладке, где список POU, щёлкните правой клавишей мыши и в появившемся контекстном меню выберите команду «Добавить объект» (Add Object). В автоматически открывшемся диалоговом окне выберите тип объекта Function Block, задайте ему имя motor и язык ST (рис. 29).

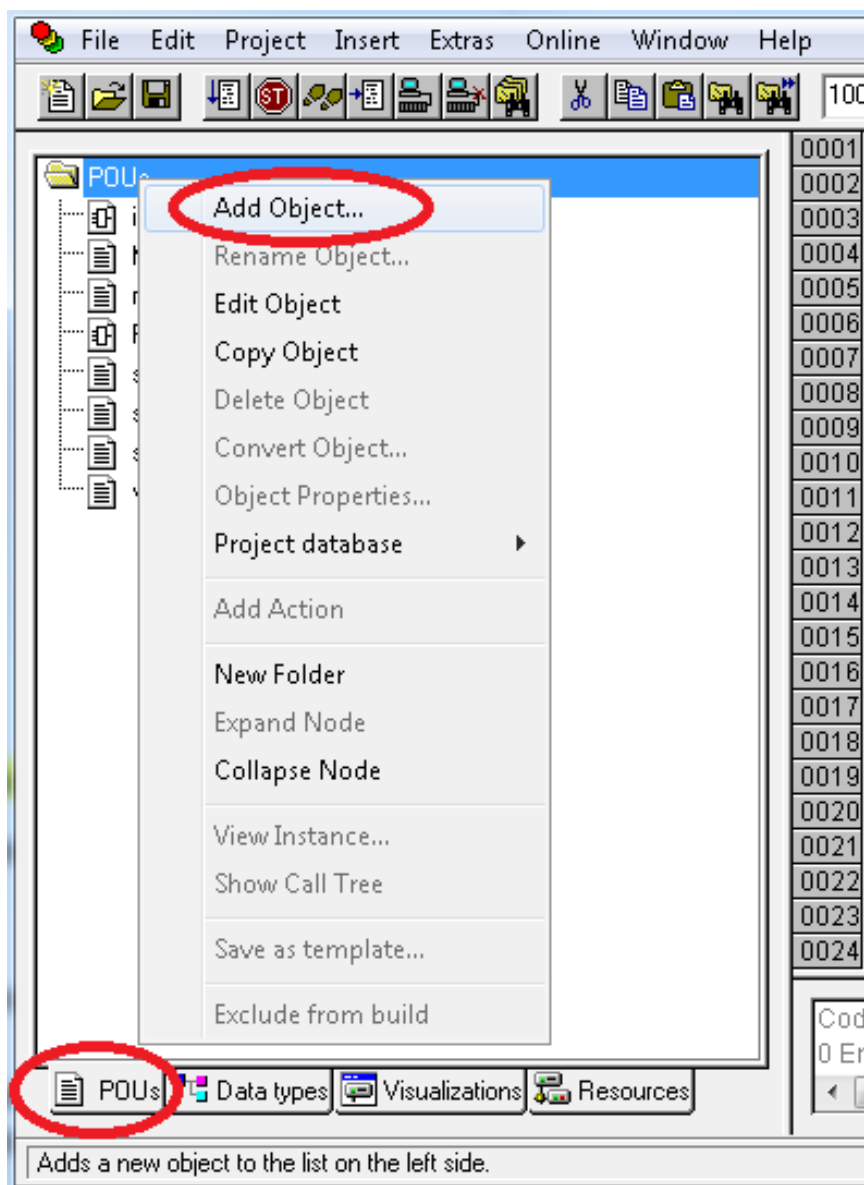


Рис. 29. Добавление нового функционального блока

Объявите в блоке motor следующие входные переменные:

- start типа BOOL – для запуска двигателя;
- stop типа BOOL – для остановки двигателя;
- slowdown типа BOOL – переменная замедления ленты конвейера;
- left_dimension типа BOOL – переменная наличия выхода транспортируемого груза за левую границу (перекрытие светового луча левого габарита);
- right_dimension типа BOOL – переменная наличия выхода транспортируемого груза за правую границу (перекрытие светового луча правого габарита);

– full_weight типа BOOL – переменная полного заполнения грузами любой из приёмных корзин.

Объявите в блоке motor выходную переменную work типа BOOL – переменная, которая в состоянии TRUE подаёт сигнал на запуск двигателя.

В рабочей области редактора блока motor пропишите программный код на языке программирования ST (рис. 30).

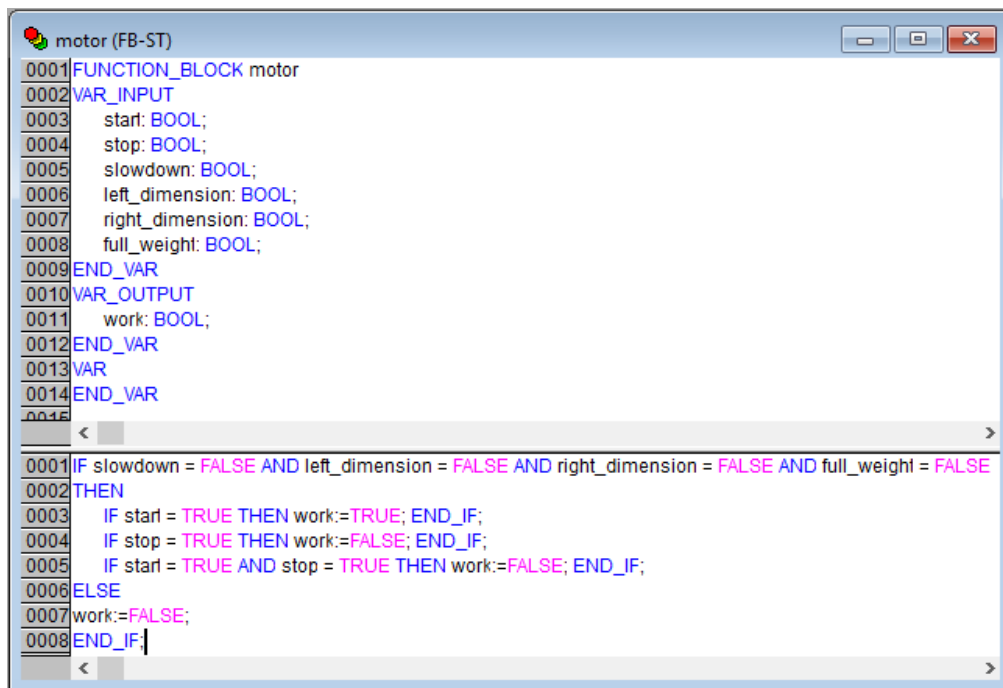


Рис. 30. Окно функционального блока motor

Данный программный код осуществляет проверку условий безаварийной транспортировки: если переменные slowdown, left_dimension, right_dimension, full_weight равны FALSE, то переменной work разрешено принимать значение TRUE. Также данный код проверяет условия запуска и останова двигателя: если переменная start равна TRUE (нажата кнопка пуска конвейера), то переменная work принимает значение TRUE; если переменная stop равна TRUE (нажата кнопка стопа конвейера), то переменная work принимает значение FALSE.

- *Создайте функциональный блок контроля замедления ленты.* В первой системной вкладке, где список POU, щёлкните правой клавишей мыши и в появившемся контекстном меню выберите команду «Добавить объект» (Add Object). В автоматически открывшемся диалоговом окне выберите тип объекта «Функциональный блок» (Function Block), задайте ему имя slowdown и язык ST.

Объявите в блоке slowdown входную переменную velocity типа REAL – это переменная скорости барабана, приводящего в движение ленту.

Объявите в блоке slowdown выходную переменную alarm – переменная типа BOOL, которая сигнализирует о замедлении ленты конвейера.

Программный код функционального блока slowdown показан на рис. 31.

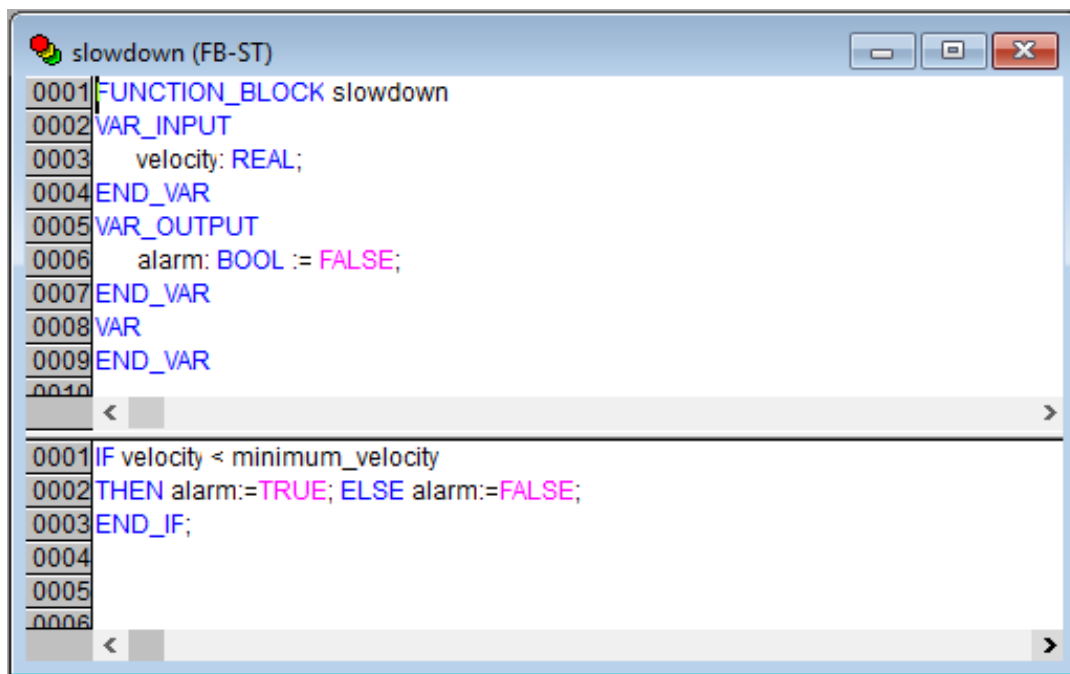


Рис. 31. Окно функционального блока slowdown

- *Создайте блок измерения скорости вращения барабана.* В первой системной вкладке, где список ROU, щёлкните правой клавишей мыши и в появившемся контекстном меню выберите команду «Добавить объект» (Add Object). В автоматически открывшемся диалоговом окне выберите тип объекта «Функциональный блок» (Function Block), задайте ему имя velocity и язык ST.

Объявите в блоке velocity входную переменную sensor типа BOOL – переменная наличия импульса от фотодатчика, принимающего свет от зеркала, которое вращается вместе с барабаном.

Объявите в блоке velocity выходную переменную velocity типа REAL – переменная, выдающая значение окружной скорости вращения барабана в см/с.

Объявите в блоке velocity следующие локальные переменные:

- block – переменная типа BOOL, необходимая для фиксации момента замера времени появления импульса от фотодатчика;
- time1 – переменная типа TIME, принимающая значение момента времени появления «предыдущего» импульса от фотодатчика;
- time2 – переменная типа TIME, принимающая значение момента времени появления «следующего» импульса от фотодатчика;

- interval_time – переменная типа TIME, принимающая значение интервала между «предыдущим» и «следующим» импульсами от фотодатчиков;
 - interval_dword – переменная типа DWORD (двойное слово), т. е. это то же время interval_time, только в формате двойного слова;
 - interval_real – переменная типа REAL, это то же время interval_dword, только измеряется в миллисекундах.
- Вставьте в рабочую область блока velocity код (рис. 32).

```

0001 FUNCTION_BLOCK velocity
0002 VAR_INPUT
0003     sensor: BOOL := FALSE;
0004 END_VAR
0005 VAR_OUTPUT
0006     velocity: REAL;
0007 END_VAR
0008 VAR
0009     block: BOOL := TRUE;
0010     time1: TIME := T#0s;
0011     time2: TIME := T#0s;
0012     interval_time: TIME := T#0s;
0013     interval_dword: DWORD;
0014     interval_real: REAL;
0015 END_VAR
0016 VAR CONSTANT
0017 END_VAR
0018
0001 IF sensor = TRUE AND block = TRUE
0002 THEN
0003     interval_time := time2 - time1;
0004     time1 := TIME();
0005     block := FALSE;
0006 END_IF;
0007
0008 IF sensor = FALSE
0009 THEN
0010     block := TRUE;
0011 END_IF;
0012
0013 time2 := TIME();
0014
0015 interval_dword := TIME_TO_DWORD(interval_time);
0016 interval_real := (DWORD_TO_REAL(interval_dword))/1000; (*Second*)
0017
0018 velocity := sensor_way * interval_real; (*Santimeter / Second*)

```

Рис. 32. Окно функционального блока velocity

Принцип работы кода состоит в следующем. При подаче на вход блока velocity в переменную sensor значения TRUE, так как block=TRUE, переменная time1 принимает значение начального момента времени, затем переменной блокировки block присваивается значение FALSE. Теперь даже при повторном чтении данного кода вычислительной машиной (ПК), уже при block=FALSE, первое IF-условие уже не

будет выполняться, а значит, переменная `time1` далее будет сохранять своё ранее принятое значение. Переменная `interval_time` будет равна 0 на начальном этапе прочтения компьютером кода. При снятии сигнала `sensor`, т. е. при переводе его в состояние `FALSE`, переменная блокировки `block` принимает значение `TRUE`, тем самым производится подготовка к следующему запуску первого IF-условия для последующей стадии расчёта временного интервала. Пока `sensor` находится в `FALSE`, значение переменной `time2` увеличивается в режиме реального времени. Как только `sensor` станет `TRUE`, сработает первый оператор IF и в нём произойдёт расчёт разности времени между конечным значением `time2` и начальной временной точкой `time1`, которая до этого момента сохранялась постоянной. Таким образом, происходит вычисление времени между «предыдущим» импульсом `sensor` и «последующим». После этого `time1` принимает новое «стартовое» время, и когда блокиратор `block` станет `FALSE`, `time1` останется постоянным до следующего такого же импульса `sensor`. После каждого расчёта разности времени между импульсами `sensor` производится вычисление скорости движения ленты конвейера путём деления расстояния, пройденного лентой от одного момента подачи `sensor` до следующего, на время между этими подачами сигнала `sensor`. Причём это время и есть та разница `interval_time:=time2-time1`. Подача импульсов `sensor` напрямую связана с отражением света от зеркал на барабанах конвейера и попаданием его на фотодатчики.

- *Создайте функциональный блок сигнала сброса груза.* В первой системной вкладке, где список `POU`, щёлкните правой клавишей мыши и в появившемся контекстном меню выберите команду «Добавить объект» (Add Object). В автоматически открывшемся диалоговом окне выберите тип объекта «Функциональный блок» (Function Block), задайте ему имя `stage` и язык `ST`.

Объявите в блоке `stage` входную переменную в редакторе объявлений `sensor` типа `BOOL` – переменная наличия импульса, когда перекрывается инфракрасный луч контроля высоты груза.

Объявите в блоке `stage` выходную переменную `movie` типа `BOOL` – переменную, с которой подаётся сигнал на поворот площадки для сброса груза в корзину.

Объявите в блоке `stage` следующие локальные переменные:

- `time1` – переменная типа `TIME`;
- `time2` – переменная типа `TIME`;
- `interval_time` – переменная типа `TIME`;
- `interval_dword` – переменная типа `DWORD` (двойное слово), т. е. это то же время `interval_time`, только в формате двойного слова;

– interval_real – переменная типа REAL, то же время interval_dword, только измеряется в миллисекундах.

Вставьте в рабочую область блока stage код (рис. 33).

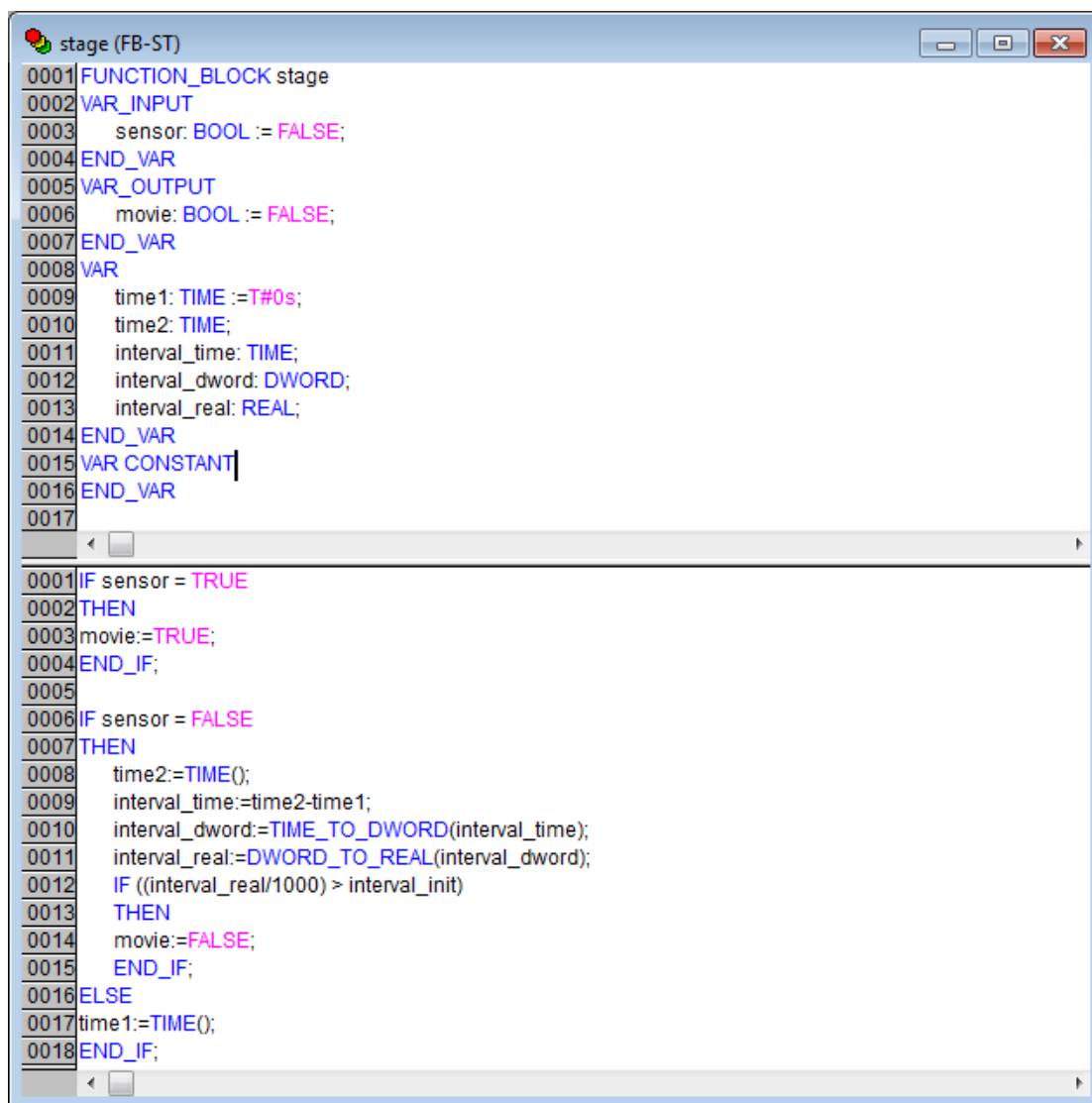


Рис. 33. Окно функционального блока stage

- Создайте функциональный блок координации поворотной площадки. В первой системной вкладке, где список ROU, щёлкните правой клавишей мыши и в появившемся контекстном меню выберите команду «Добавить объект» (Add Object). В автоматически открывшемся диалоговом окне выберите тип объекта «Функциональный блок» (Function Block), задайте ему имя stage_prove и язык ST.

Объявите в блоке stage_prove следующие входные переменные:

- high_input – переменная типа BOOL, принимающая сигнал при перекрытии светового луча высоким грузом;

– low_input – переменная типа BOOL, принимающая сигнал при перекрытии светового луча низким грузом.

Объявите в блоке stage_prove следующие выходные переменные:

– move_high – переменная типа BOOL, подающая сигнал поворота площадки в сторону корзины для высоких грузов;

– move_low – переменная типа BOOL, подающая сигнал поворота площадки в сторону корзины для низких грузов.

Вставьте в рабочую область блока stage_prove код (рис. 34).

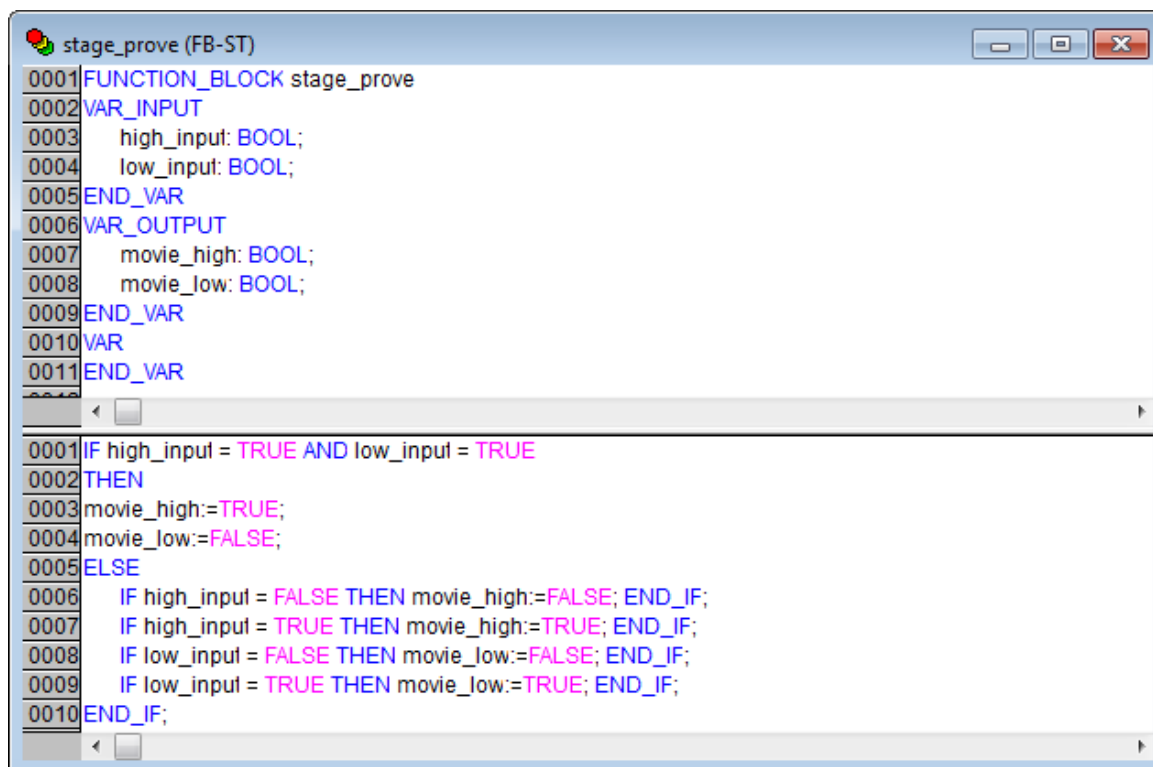


Рис. 34. Окно функционального блока stage_prove

Смысл блока stage_prove – предотвратить ложную подачу сигналов на поворот площадки сброса грузов одновременно в одну и в другую стороны, когда высокий груз перекрывает два световых луча, контролирующих, соответственно, высокие и низкие грузы.

- *Создайте функциональный блок инерционной задержки сигнала.* В первой системной вкладке, где список ROU, щёлкните правой клавишей мыши и в появившемся контекстном меню выберите команду «Добавить объект» (Add Object). В автоматически открывшемся диалоговом окне выберите тип объекта «Функциональный блок» (Function Block), задайте ему имя inertial и язык FBD.

Необходимость в этом блоке обусловлена вероятностью ложных импульсов с датчиков веса, что может привести к остановке конвейера за счёт ложного поступления сигнала на вход full_weight блока motor.

Также этот блок нужен для предотвращения ложной остановки двигателя через блок slowdown, пока лента набирает скорость при пуске.

Программный код функционального блока inertial показан на рис. 35.

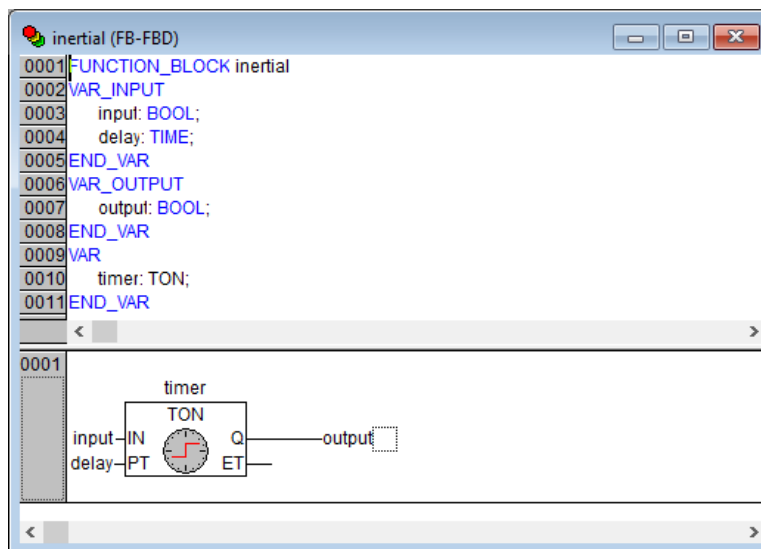


Рис. 35. Окно функционального блока inertial

- Создайте функциональный блок задания скорости вращения двигателя. В первой системной вкладке, где список ROU, щёлкните правой клавишей мыши и в появившемся контекстном меню выберите команду «Добавить объект» (Add Object). В автоматически открывшемся диалоговом окне выберите тип объекта «Функциональный блок» (Function Block), задайте ему имя Manage_motor и язык ST. Этот блок необходим в случае использования частотного преобразователя, когда включены контакторы K2 и K3 (см. рис. 21).

Задайте переменные и создайте программный код, как показано на рис. 35.

У данного блока есть две входные переменные – rotation_up и rotation_down – для увеличения или уменьшения внутренней переменной CV счётчика CTUD_1. Переменная CV передаёт своё значение на внутреннюю переменную rv функционального блока Manage_motor. С помощью условных операторов производится проверка значения переменной rv, на основании которого формируется двоичный 3-разрядный код. Этот код поразрядно подаётся на выходные переменные Q4, Q5, Q6 функционального блока Manage_motor. Последние значения двоичного кода в виде трёх отдельных сигналов подаются через электрические выходы контроллера на входы DI3, DI4, DI5 преобразователя частоты (см. рис. 21).

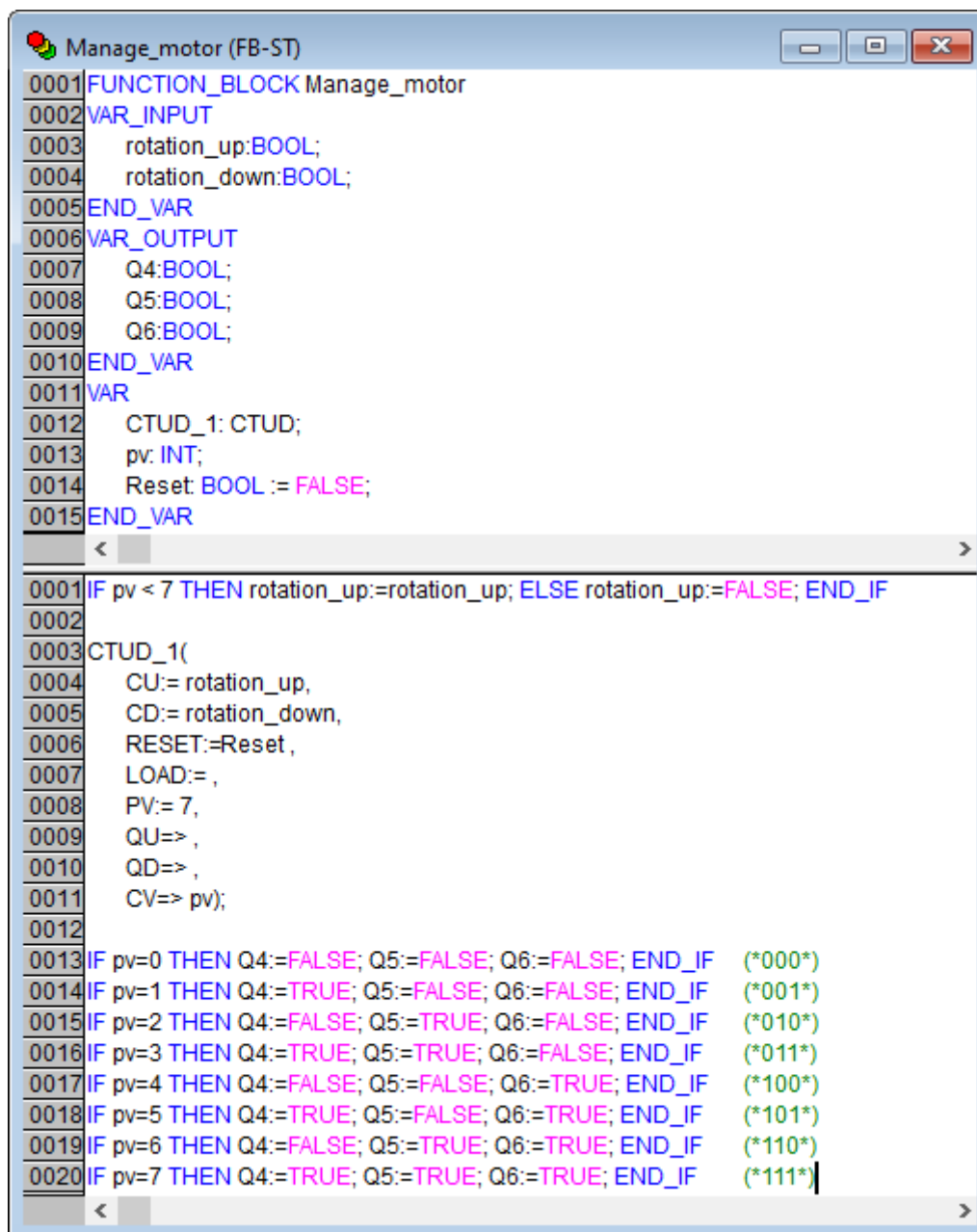


Рис. 36. Окно функционального блока Manage_motor

9. Объявите переменные передачи данных по CAN Open. Перейдите на вкладку «Ресурсы» (Resources), и далее в подраздел «Глобальные переменные» (Global_Variables). Кликните правой клавишей мыши по разделу (папке) глобальных переменных добавьте новый объект с настройками как на рис. 37.

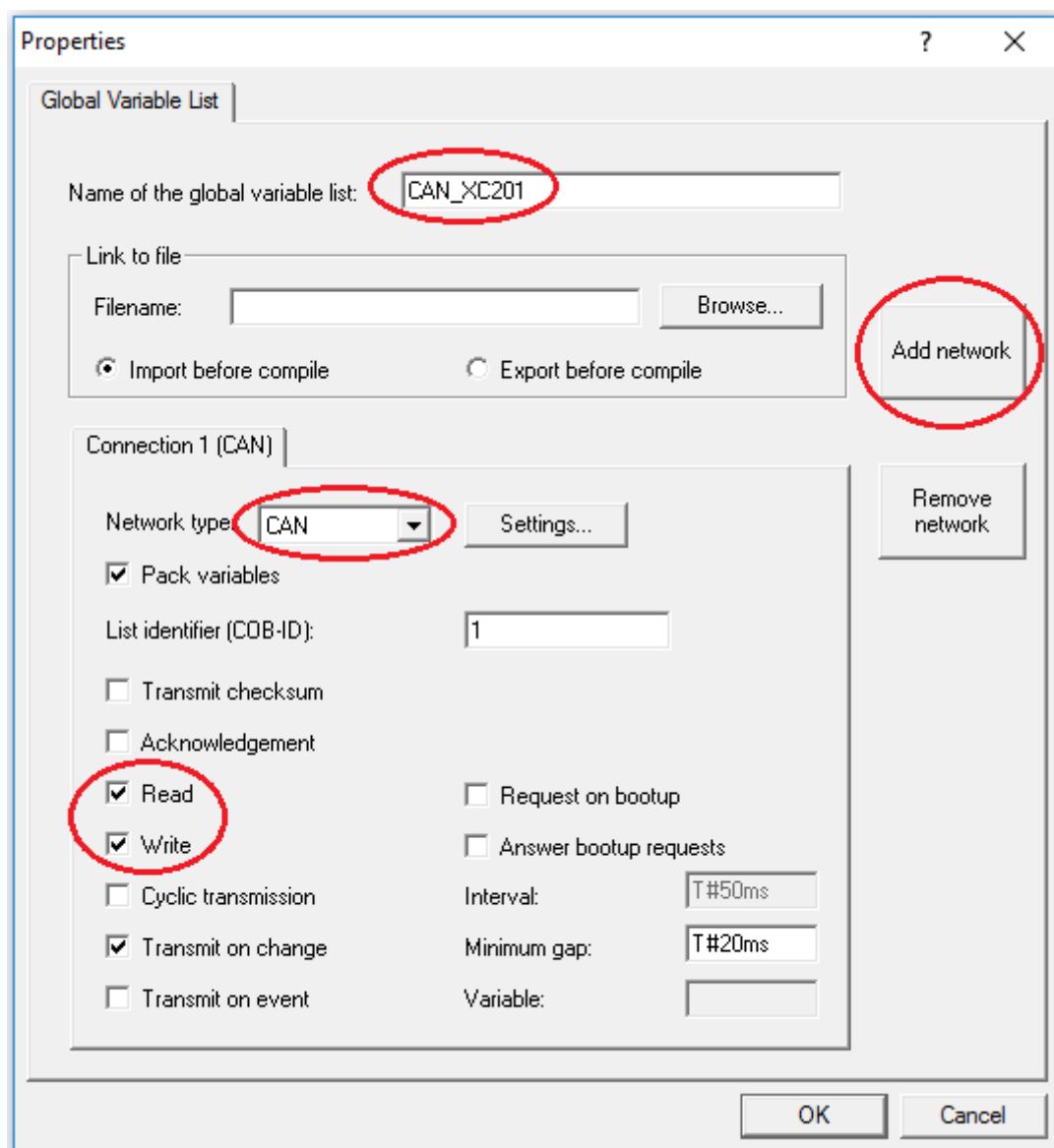


Рис. 37. Добавление нового объекта в глобальные переменные

Задайте переменные передачи данных по протоколу CAN Open от HMI панели в контроллер, как показано на рис. 38.

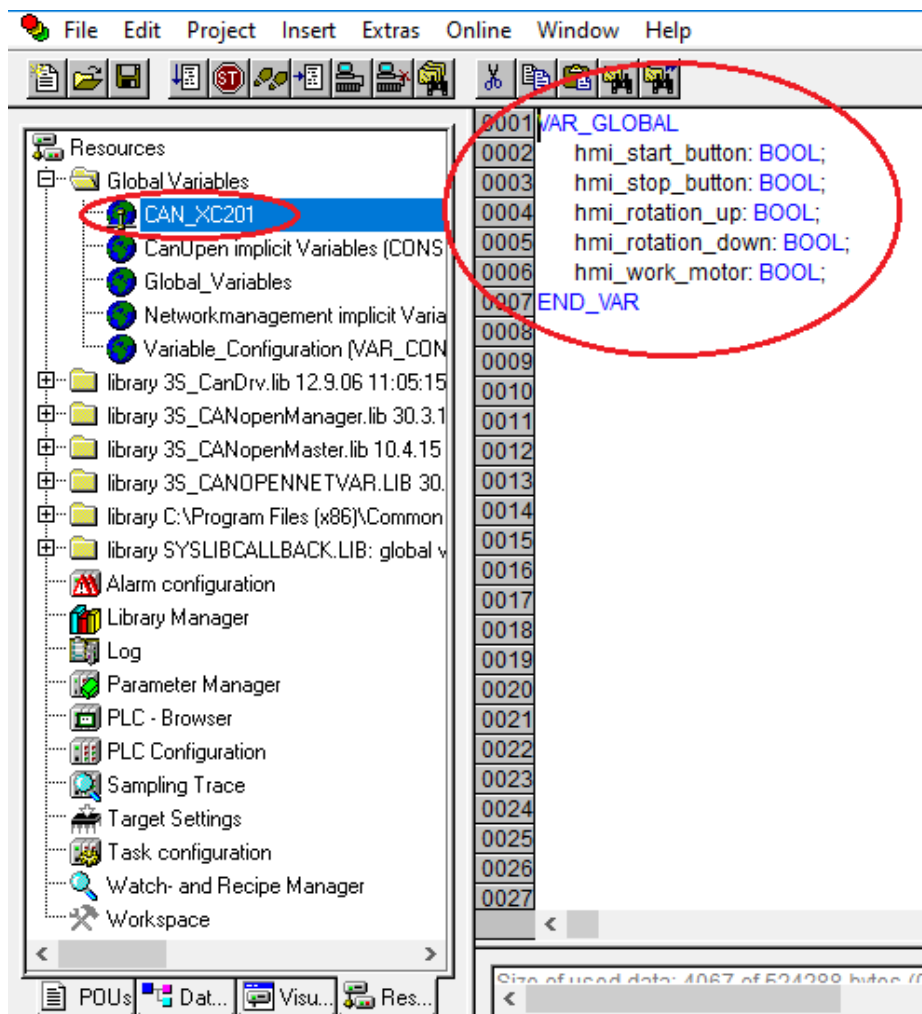


Рис. 38. Переменные передачи CAN Open данных

10. Проверьте работоспособность программы в режиме эмуляции. Для запуска проекта в режиме эмуляции установите во вкладке «Онлайн» (Online) галочку против «Эмуляция» (Simulation). Откомпилируйте проект: «Проект/Компилировать всё» (Project/Rebuild All). Установите соединение с контроллером: «Онлайн/Соединение» (Online/Login). Запустите проект: «Онлайн/Запуск» (Online/Run). Поэкспериментируйте с заданием различных переменных.

11. Сформируйте символьный файл с тегами OPC переменных. Для этого откройте опции проекта в *Project/Options*. Далее зайдите в символьную конфигурацию и в окне установки атрибутов выберите все те переменные программы PLC_PRG. Эти переменные отвечают за следующее: приём сигналов с электрических кнопок, приём сигналов с кнопок и задатчиков от SCADA системы, вывод управляющих сигналов с контактных разъёмов контроллера, посылку сигналов в SCADA систему (рис. 39).

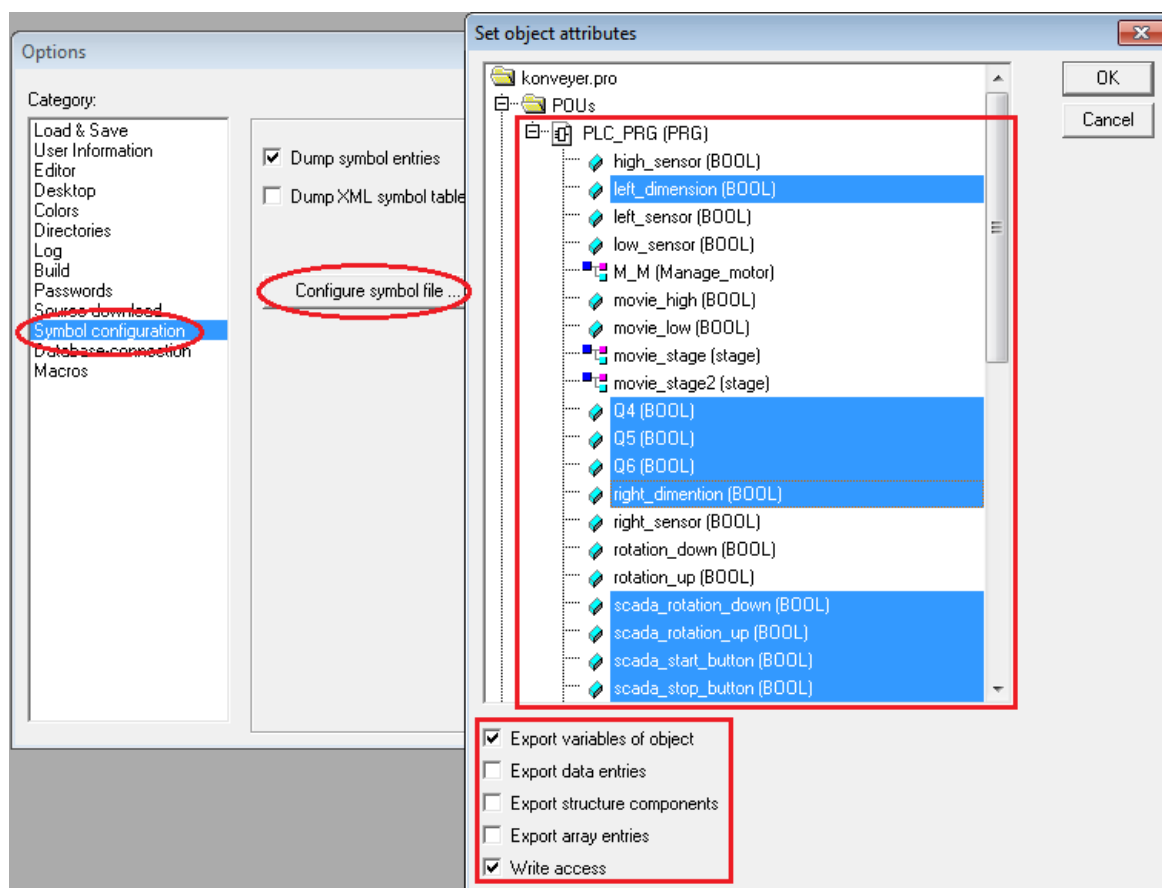


Рис. 39. Конфигурация символьного файла

12. Откомпилируйте проект «Проект/Компилировать всё» (Project/Rebuild All). В результате в папке проекта должен появиться символьный файл с расширением «SYM». В составе тегов символьного файла будут присутствовать следующие записи:

```
PLC_PRG.left_dimension:BOOL:4:541:1:b:16#000000040
PLC_PRG.Q4:BOOL:2:3:0:b:16#000000020
PLC_PRG.Q5:BOOL:2:4:0:b:16#000000020
PLC_PRG.Q6:BOOL:2:5:0:b:16#000000020
PLC_PRG.right_dimention:BOOL:4:542:1:b:16#000000040
PLC_PRG.scada_rotation_down:BOOL:4:546:1:b:16#000000040
PLC_PRG.scada_rotation_up:BOOL:4:545:1:b:16#000000040
PLC_PRG.scada_start_button:BOOL:4:543:1:b:16#000000040
PLC_PRG.scada_stop_button:BOOL:4:544:1:b:16#000000040
PLC_PRG.work_motor:BOOL:2:0:0:b:16#000000020
```

13. Загрузите программу во внутреннюю память контроллера XC-CPU201. Во вкладке «Онлайн» (Online) откройте диалог «Параметры соединения» (Communication parameters) и нажмите кнопку «Создать» (New) для настройки нового соединения с типом TCP/IP. Далее присвойте

ему осмысленное имя. Задайте IP-адрес в соответствии с картой IP адресов (рис. 3), в данном случае 192.168.119.23. Установите соединение с контроллером: «Онлайн/Соединение» (Online/Login). Далее подтвердите загрузку (download) кода проекта. Также во вкладке «Онлайн» (Online) выберите опцию «Создать загрузочный проект» (Create boot project).

2.4. Создание программы на языке программирования ST в среде CODESYS для контроллера HMI панели визуализации XV102

1. Запустите программную среду CODESYS 2.3.9.
2. Создайте новый проект: «Файл/Создать» (File/New). **ВНИМАНИЕ!** путь к папке с файлом проекта и имя самого файла должны быть названы латинскими буквами.

3. Настройте целевую платформу (Target Settings). Откройте вкладку «Целевая платформа» (Target Settings) и на странице диалогового окна «Конфигурация» (Configuration) выберите тип контроллера **XV-1xx-V2.3.9 SP8**. Во вкладке сетевой функциональности (Network functionality) поставьте галочку на против «Support network variables» и в открывшемся поле ввода пропишите «CAN» (рис. 40). Подтвердите ввод нажатием кнопки ОК.

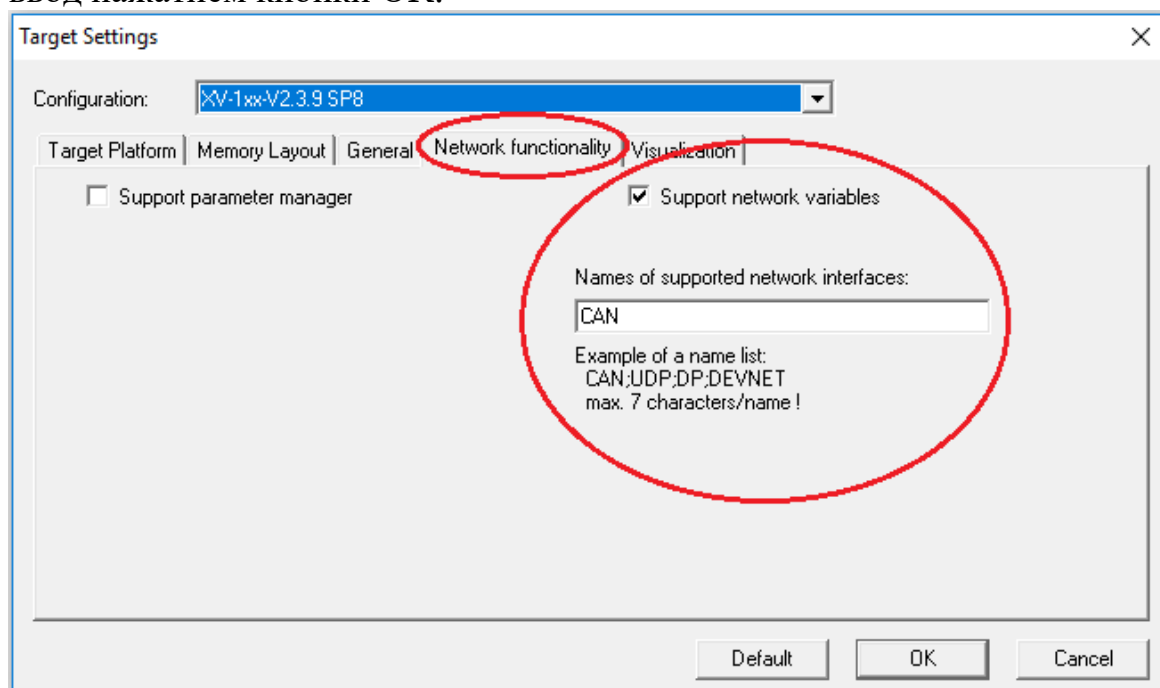


Рис. 40. Установка поддержки сетевой передачи данных через CAN Open протокол

4. *Создайте главную программу.* После настройки целевой платформы в автоматически появившемся диалоговом окне New POU выберите тип компонента «Программа» (Program), язык ST, задайте имя программы PLC_PRG.

5. *Настройте конфигурацию контроллера в HMI.* Зайдите во вкладку ресурсов «Resources» проекта и откройте конфигурацию контроллера «PLC Configuration» (рис. 41). Далее нажмите правой клавишей мыши по верхнему уровню меню (Configuration) и добавьте «CanMaster».

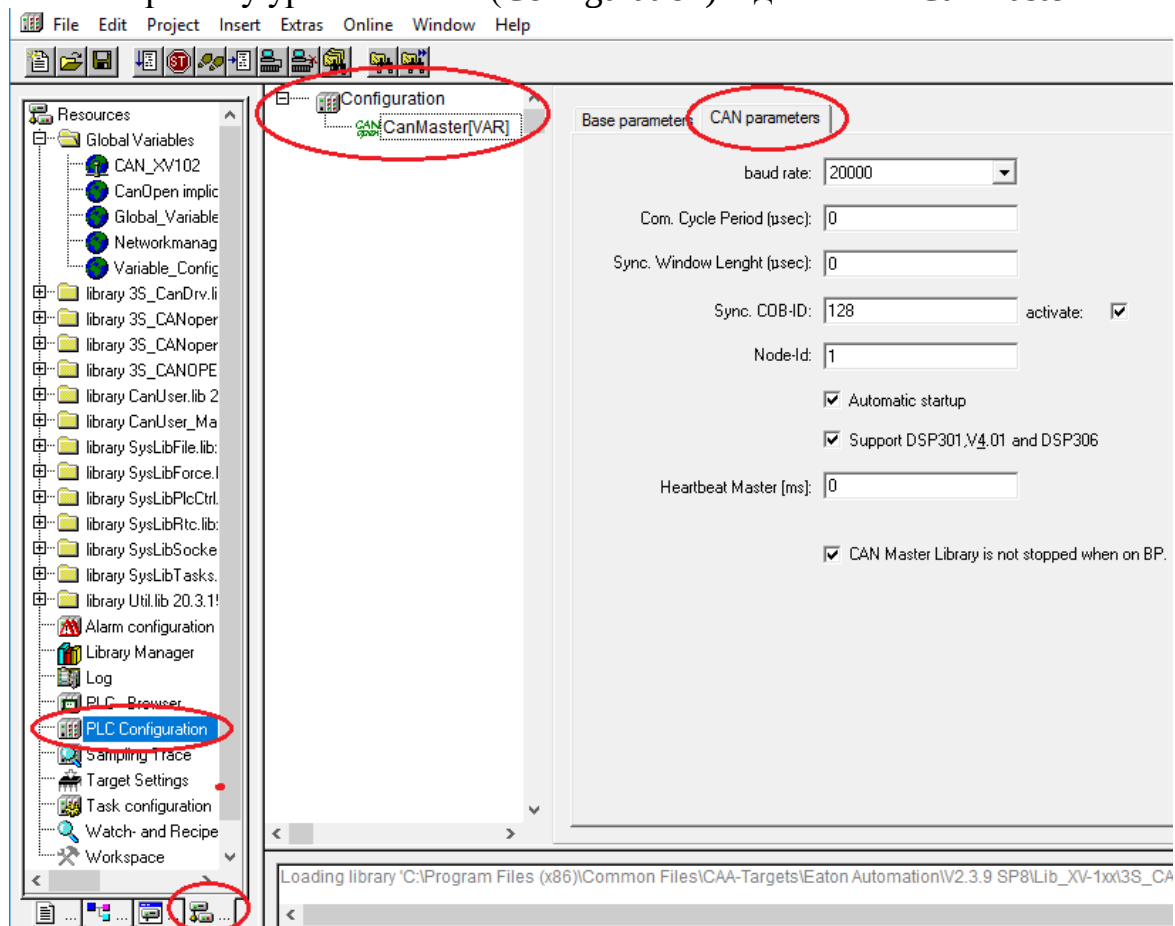


Рис. 41. Конфигурация контроллера в HMI

6. *Объявите переменные передачи данных по CAN Open.* Перейдите на вкладку «Ресурсы» (Resources), и далее в подраздел «Глобальные переменные» (Global_Variables). Кликните правой клавишей мыши по разделу (папке) глобальных переменных добавьте новый объект с настройками как на рис. 42.

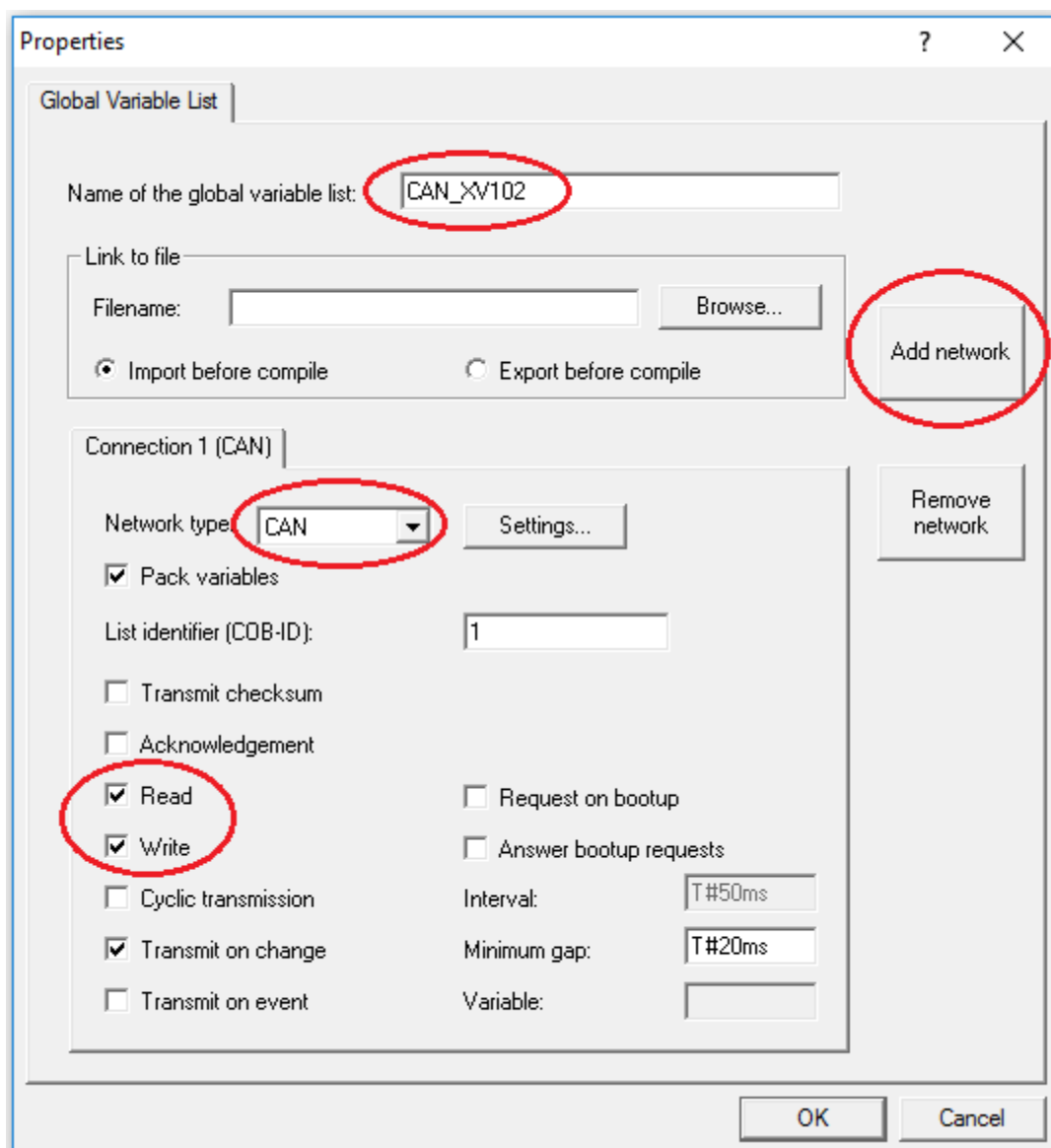


Рис. 42. Добавление нового объекта в глобальные переменные

Задайте переменные передачи данных по протоколу CAN Open от контроллера в HMI панель, так и от HMI панели в контроллер, как показано на рис. 43. **ВНИМАНИЕ! Наименование CAN переменных и тип данных должны совпадать в проекте ПЛК, ПЛК панели и проекте визуализации.**

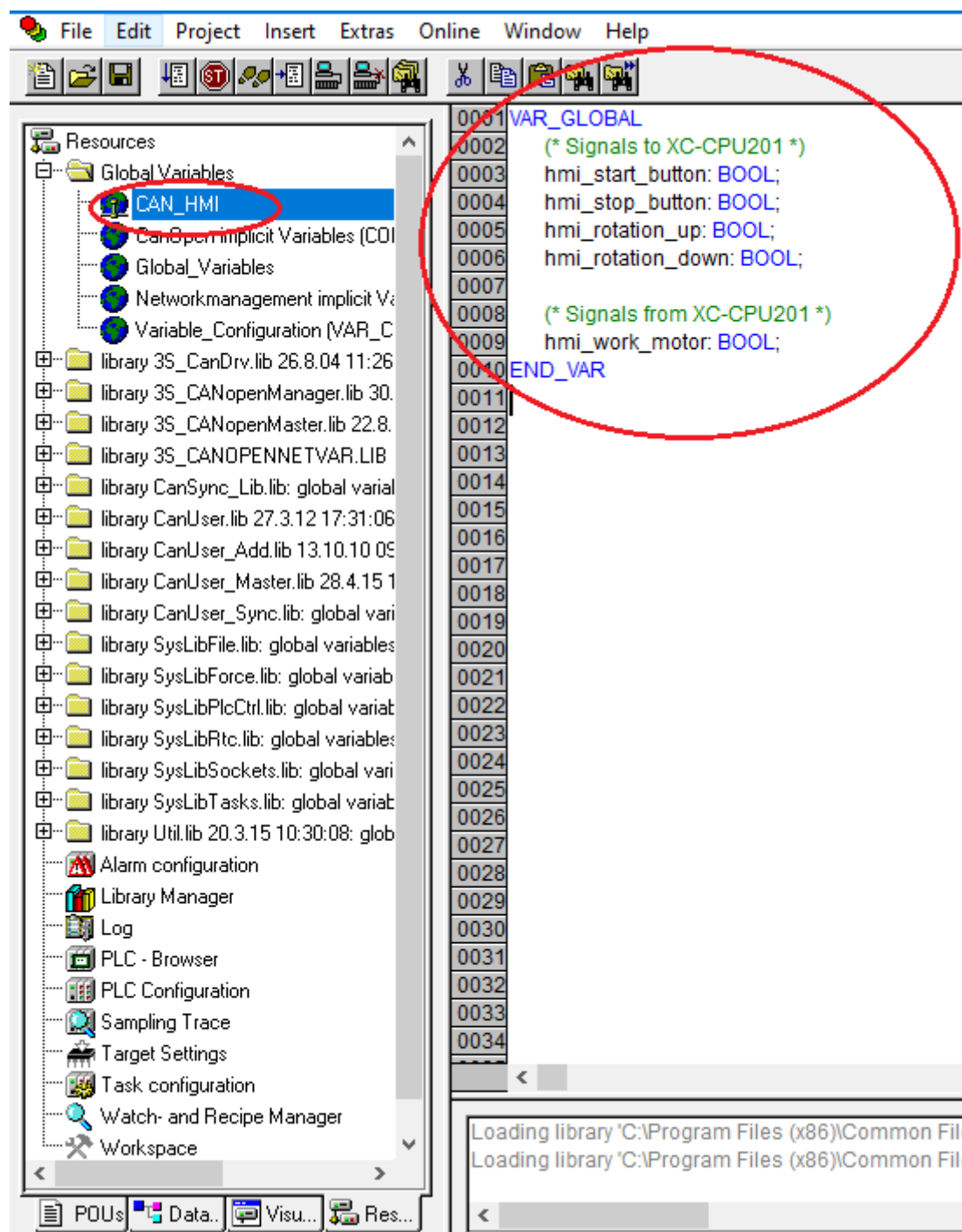


Рис. 43. Переменные передачи CAN Open данных

Главная программа во внутреннем контроллере HMI панели на языке ST показана на рис. 44.

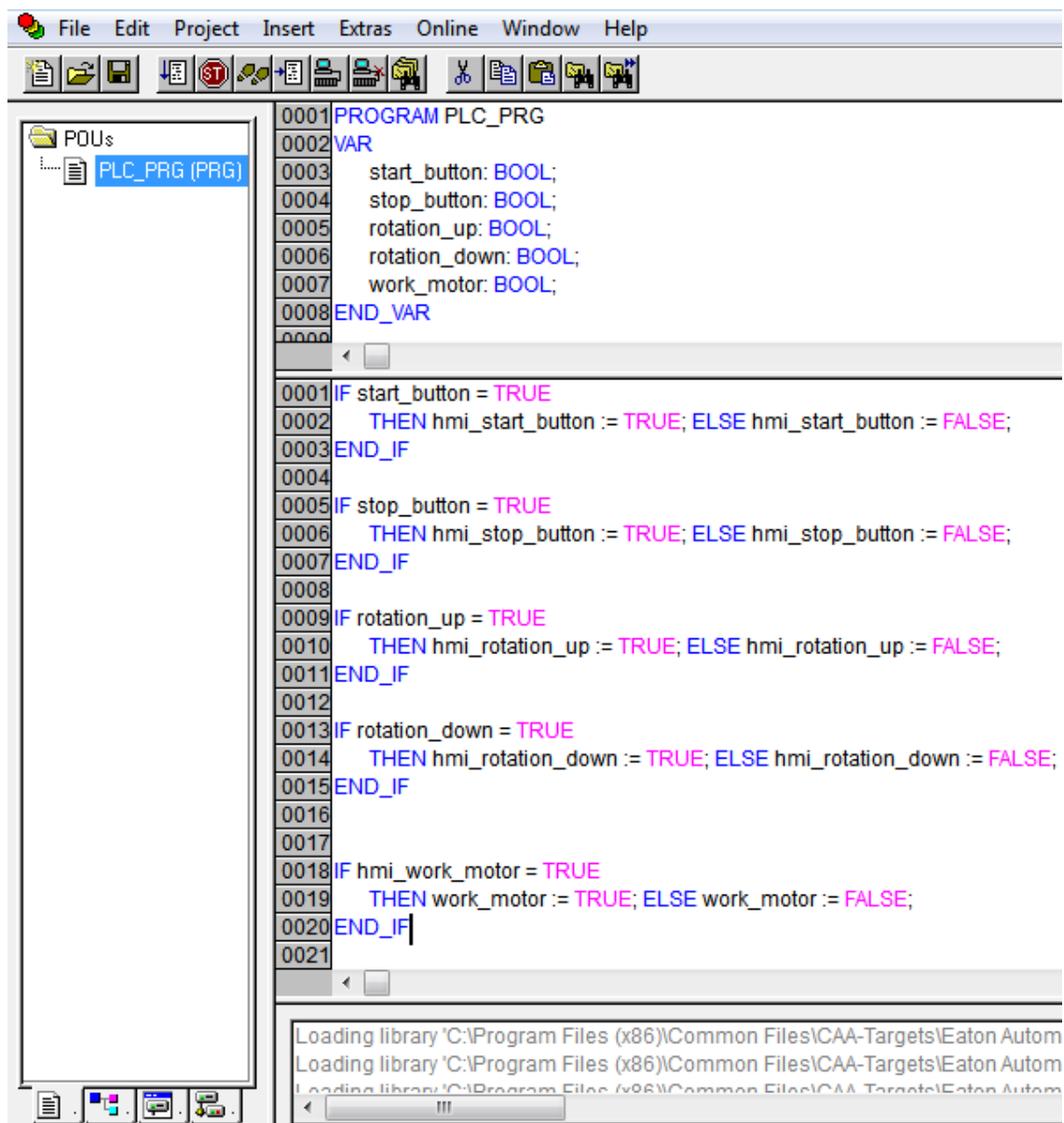


Рис. 44. Главная программа во внутреннем контроллере HMI панели

7. На панели оператора включите RS-сервер Start/Communication/RS-Server.

8. Загрузите программу во внутреннюю память контроллера HMI панели XV102. Во вкладке «Онлайн» (Online) откройте диалог «Параметры соединения» (Communication parameters) и нажмите кнопку «Создать» (New) для настройки нового соединения с типом TCP/IP. Задайте IP-адрес 192.168.119.24. Установите соединение с контроллером: «Онлайн/Соединение» (Online/Login). Подтвердите загрузку (download) кода проекта. Во вкладке «Онлайн» (Online) выберите опцию «Создать загрузочный проект» (Create boot project).

2.5. Создание графического интерфейса управления в среде GALILEO для HMI панели визуализации XV102

1. Откройте Galileo 8 и создайте новый проект Project/New.

2. Выберите тип панели Panel Type. В автоматически открывшемся окне выбрать Panel selection и после этого выбрать наименование панели оператора (рис. 45). Также после создания проекта, тип панели можно выбрать в Config/Panel Type.

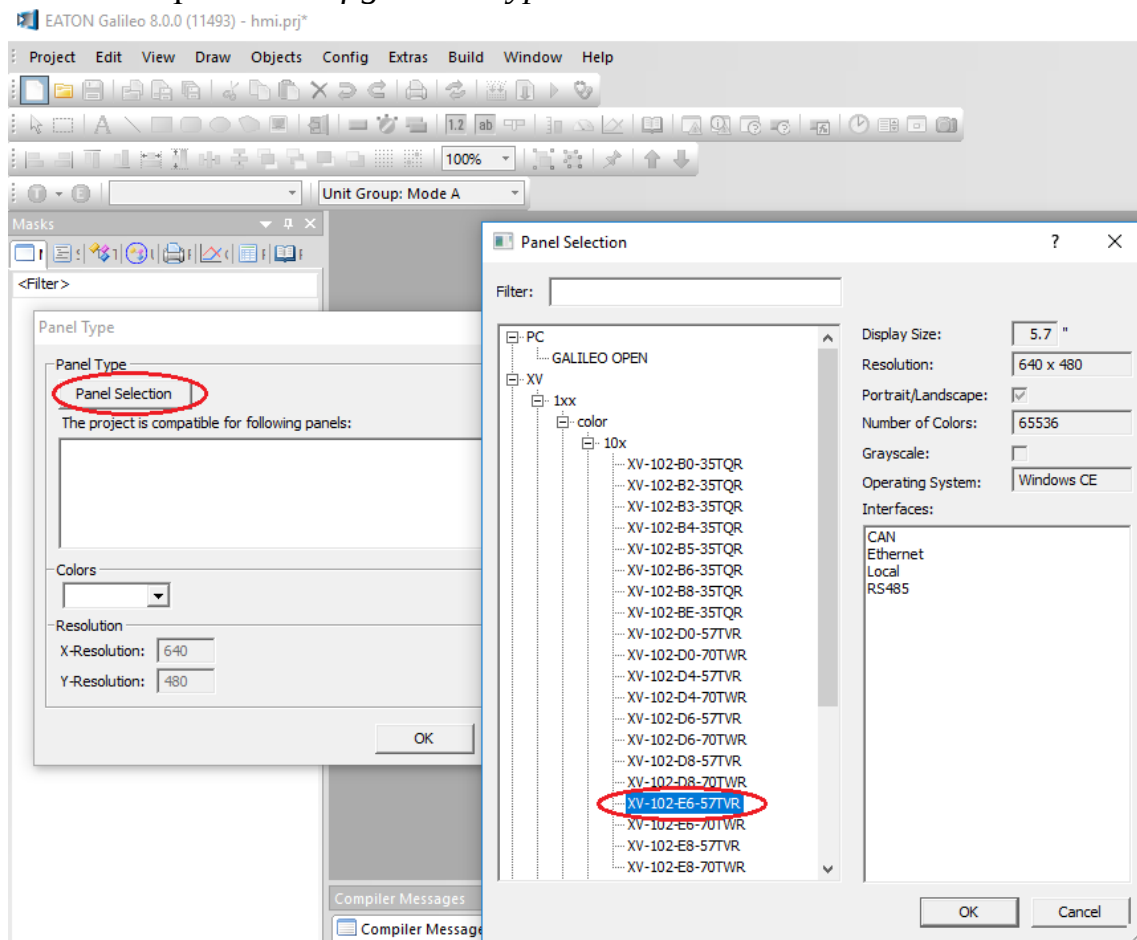


Рис. 45. Выбор панели оператора

3. Выберите коммуникацию Config/Select Communication. В открывшемся окне нажмите кнопку Add, затем выберите Codesys Xsoft-CoDesys-2/MXpro (рис. 46).

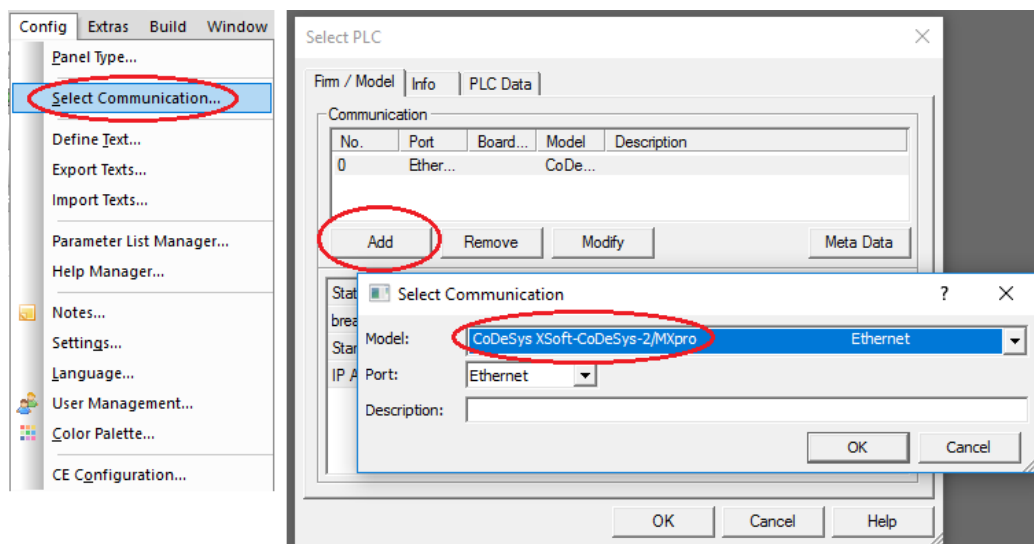


Рис. 46. Выбор коммуникации

4. *Добавьте теги.* В дереве проекта выбрать вкладку Tag, нажать правой клавишей мыши на соответствующем типе данных и задать имя тэгу (рис. 47). **ВНИМАНИЕ! Наименование CAN переменных и тип данных должны совпадать в проекте ПЛК, ПЛК панели и проекте визуализации.**

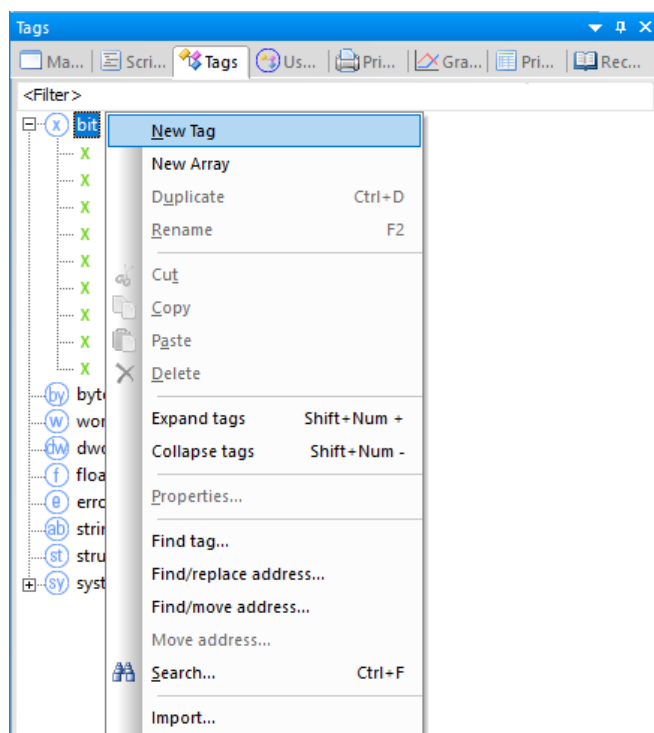


Рис. 47. Добавление тега

5. *Настройте тег.* Два раза кликните на созданном тэге. Перейдите на вкладку Address и в соответствующей строке нажмите клавишу выбора. После этого задайте параметр <prog> и <tag> (рис. 48).

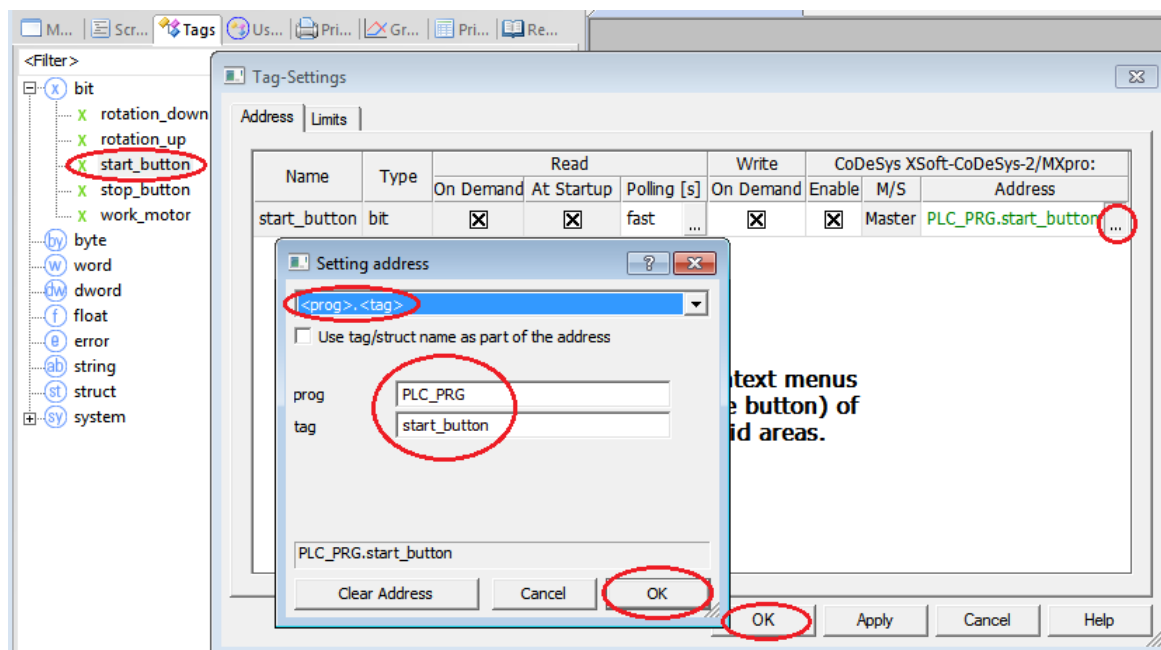


Рис. 48. Задание параметров тега

Аналогичным образом, добавьте и настройте все остальные теги в соответствии со следующим списком необходимых тегов:

- rotation_down
- rotation_up
- start_button
- stop_button
- work_motor

6. *Добавьте маску.* В дереве проекта выберите вкладку Mask, нажмите правой клавишей мыши и в диалоговом окне выберите New. Задайте имя маски (рис. 49).

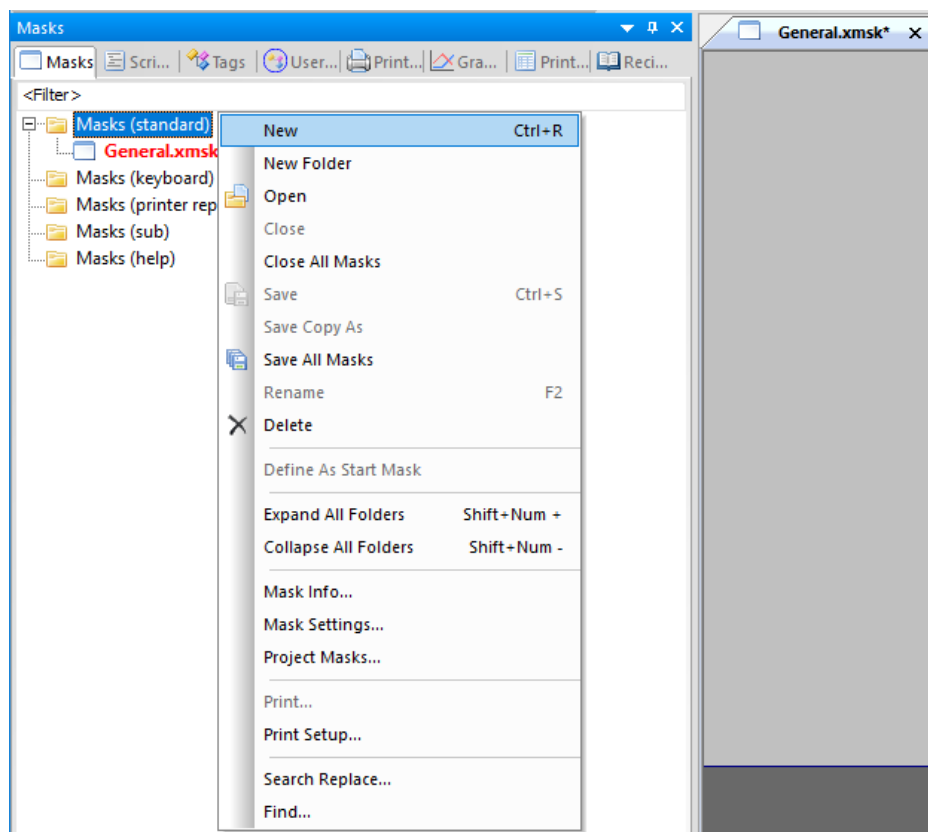


Рис. 49. Добавление маски

7. Нарисуйте необходимые графические элементы (рис. 50). Для этого используйте такие типы элементов, как «Button» и «Flag Display» (рис. 51).

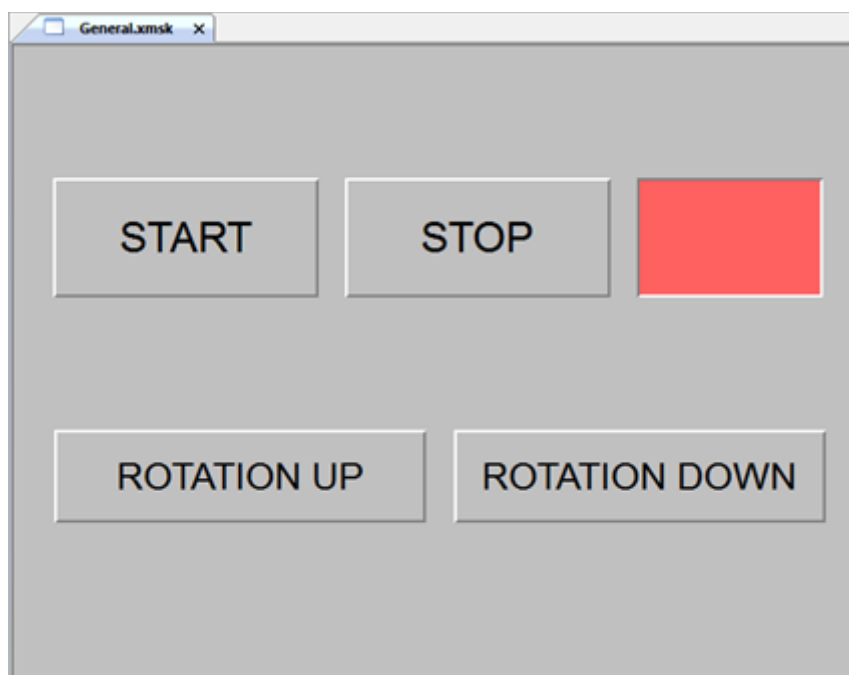


Рис. 50. Графические элементы управления на главной маске

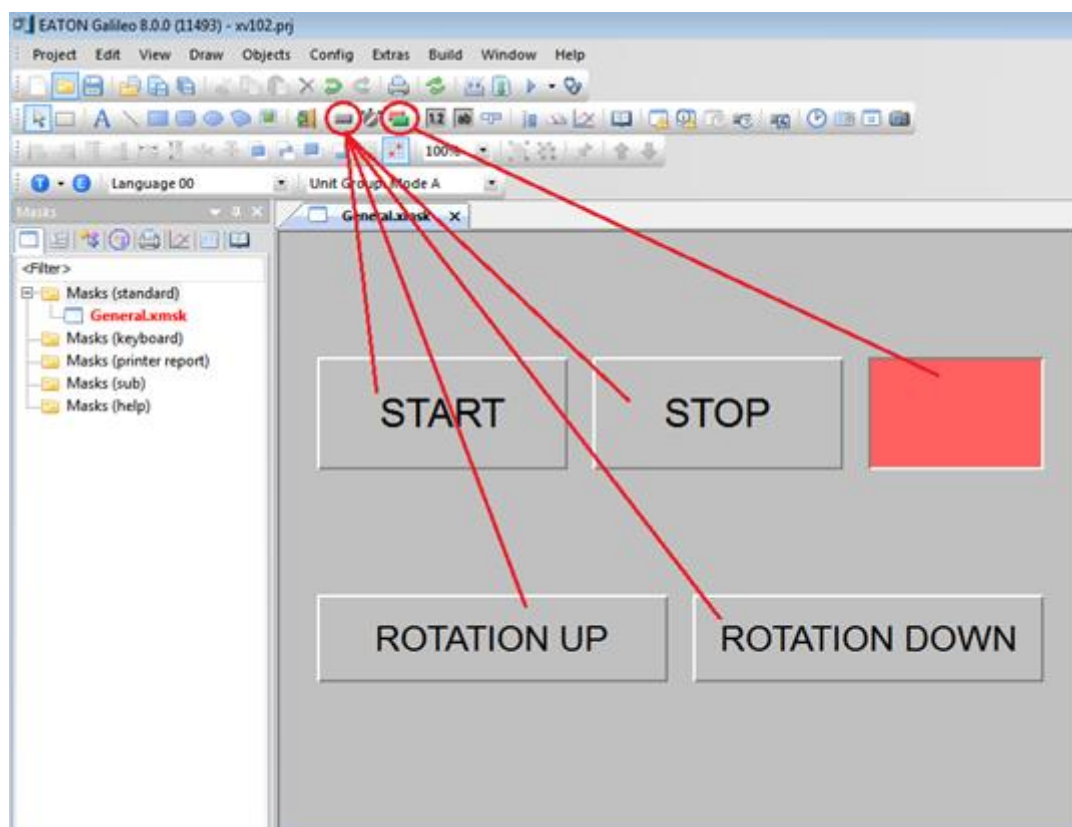


Рис. 51. Инструменты рисования графических элементов

8. *Настройте кнопки управления.* Для этого двойным кликом левой клавишей мыши перейдите в настройки данного графического элемента. Пример настроек для кнопки «drain_hot_button» (DRAIN HOT) приведён на рис. 52, 53.

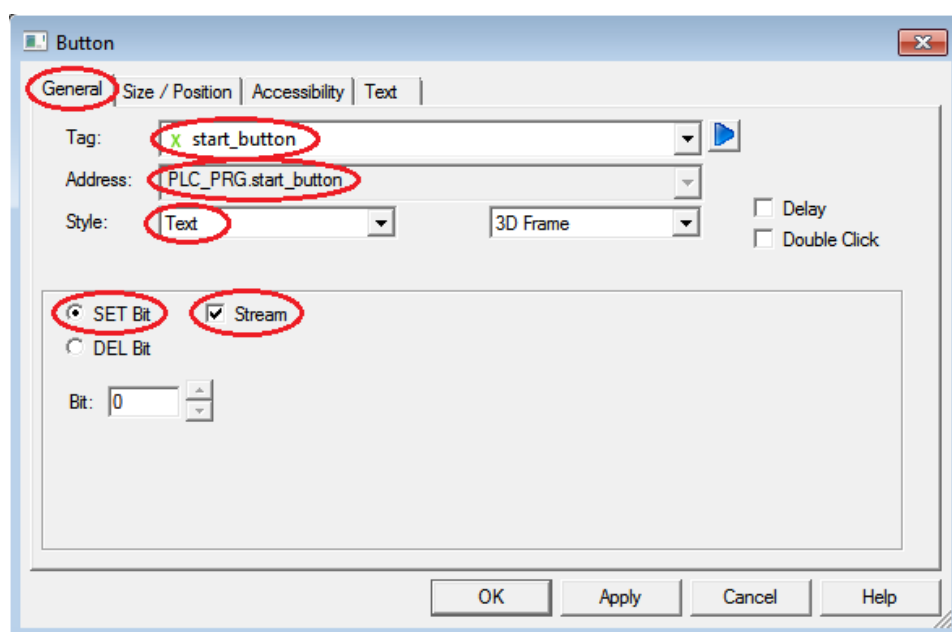


Рис. 52. Основные настройки кнопки «start_button»

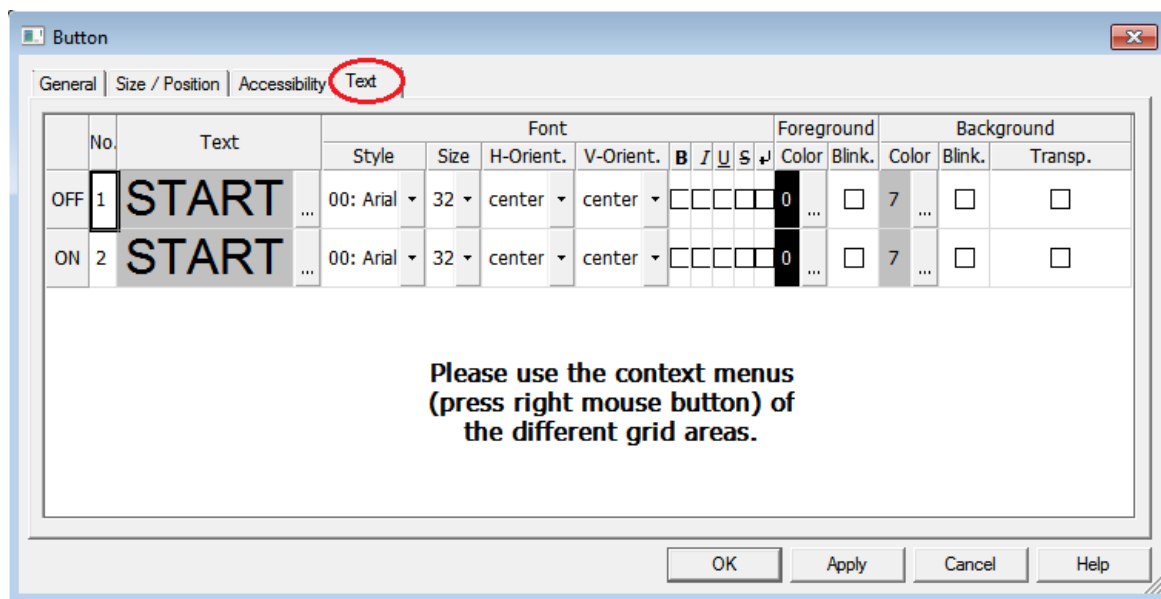


Рис. 53. Настройки текста кнопки «start_button»

Аналогичным образом настройте все остальные кнопки управления, считая, что кнопки должны привязываться к тегам, имена которых заканчиваются на «_button».

9. *Настройте флаг отображения.* Для этого двойным кликом левой клавишей мыши перейдите в настройки данного графического элемента (красный прямоугольник на рис. 51). Пример настроек для флага «work_motor» приведён на рис. 54, 55.

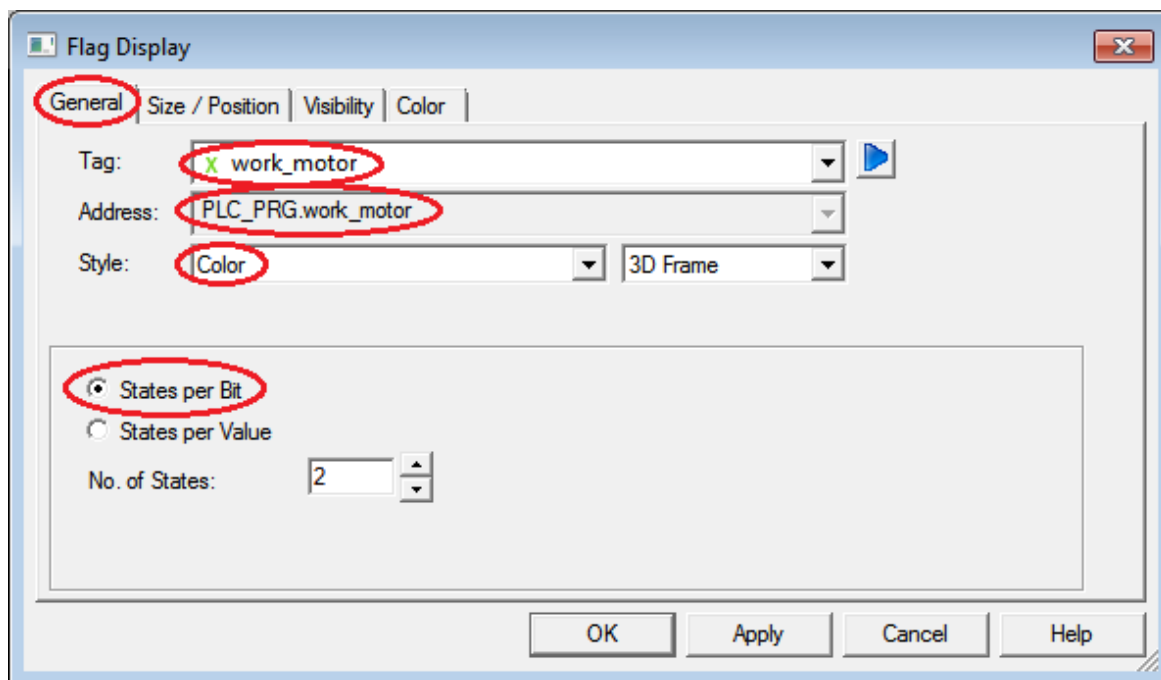


Рис. 54. Основные настройки флага «work_motor»

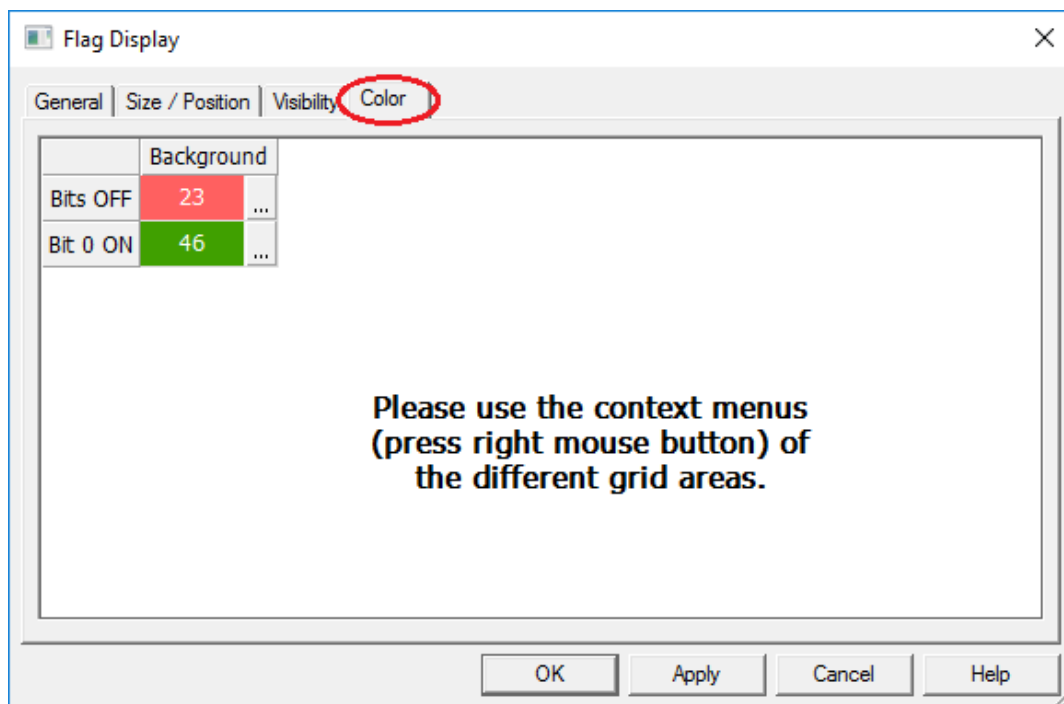


Рис. 55. Цветовые настройки флага «drain_hot_work»

10. Откомпилируйте проект перед загрузкой в HMI панель Build/Compile.

11. На панели оператора включите FTP-сервер Start/Communication/FTP-Server.

12. Загрузите проект визуализации в панель оператора. Для этого выберите Build/Download и в открывшемся окне нажмите кнопку FTP Path. Затем New Connection. В открывшемся окне задайте имя, укажите IP адрес панели (в данном случае 192.168.119.24 в соответствии со схемой по рис. 3) и нажмите ОК (рис. 56). Далее подтвердите загрузку кнопкой «Download» (рис. 57). Для успешного проведения загрузки необходимо, чтобы путь к сохраненным файлам проекта не содержал русских букв.

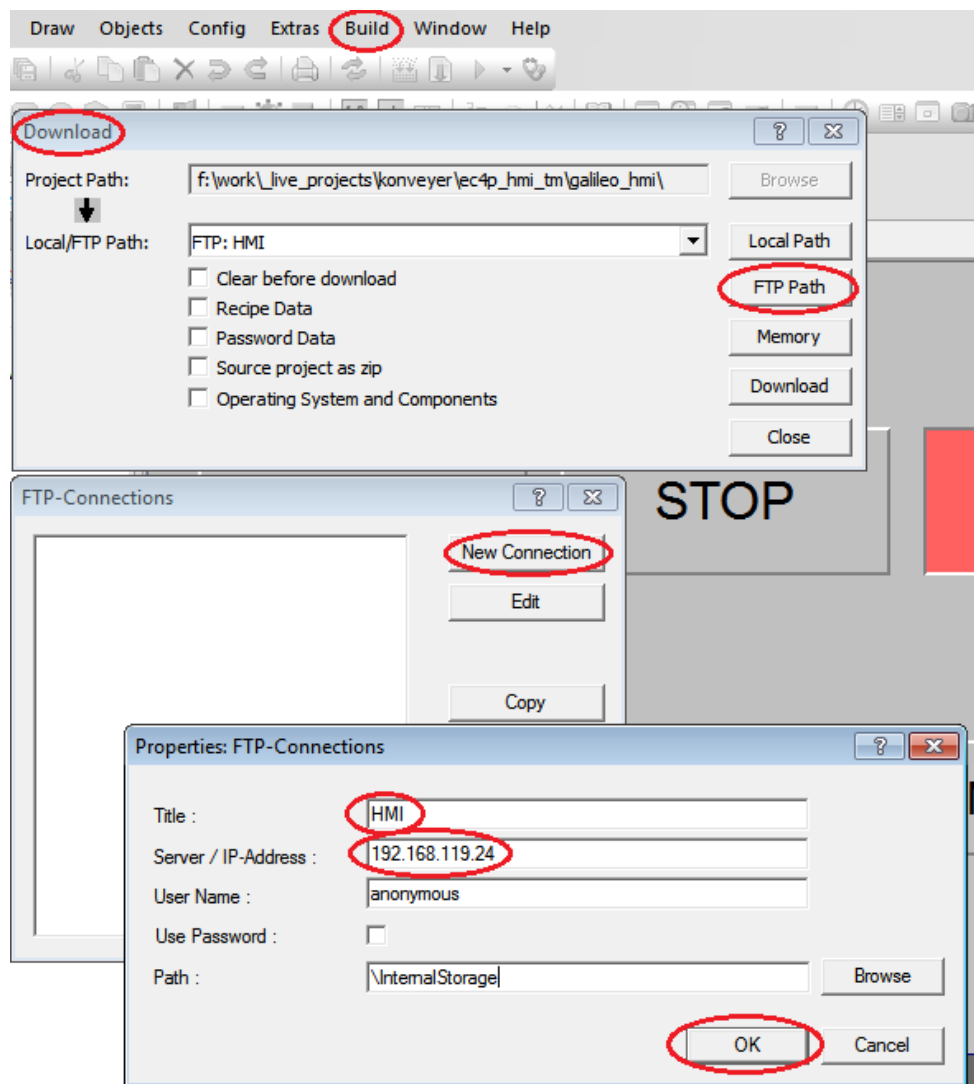


Рис. 56. Настройка FTP соединения с HMI панелью для загрузки проекта

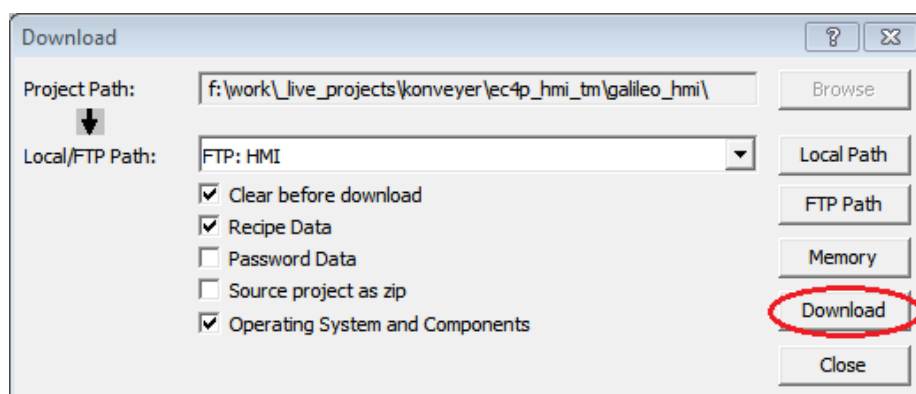


Рис. 57. Подтверждение FTP загрузки проекта в HMI панель

2.6. Создание графического SCADA интерфейса управления в среде TRACE MODE

В режиме исполнения проекта SCADA системы будет установлен двусторонний обмен данными с OPC сервером. При этом сигналы будут поступать по следующим направлениям:

- Нажатие какой-либо SCADA кнопки —► передача сигнала («TRUE» в случае дискретного сигнала) в выходной канал —► передача сигнала в выходной OPC источник —► воздействие сигнала на OPC тег символического файла —► передача значения сигнала через тег символического файла в контроллер —► воздействие сигнала на переменную в контроллере.
- Изменение значения переменной в контроллере (например в «TRUE» в случае дискретного сигнала) —► передача значения переменной через тег символического файла в OPC сервер —► передача сигнала во входной OPC приёмник —► передача сигнала во входной канал —► воздействие сигнала на графический элемент SCADA мнемосхемы (например светящуюся лампу).

1. Откройте Trace Mode 6 и создайте новый проект Файл/Новый (рис. 58).

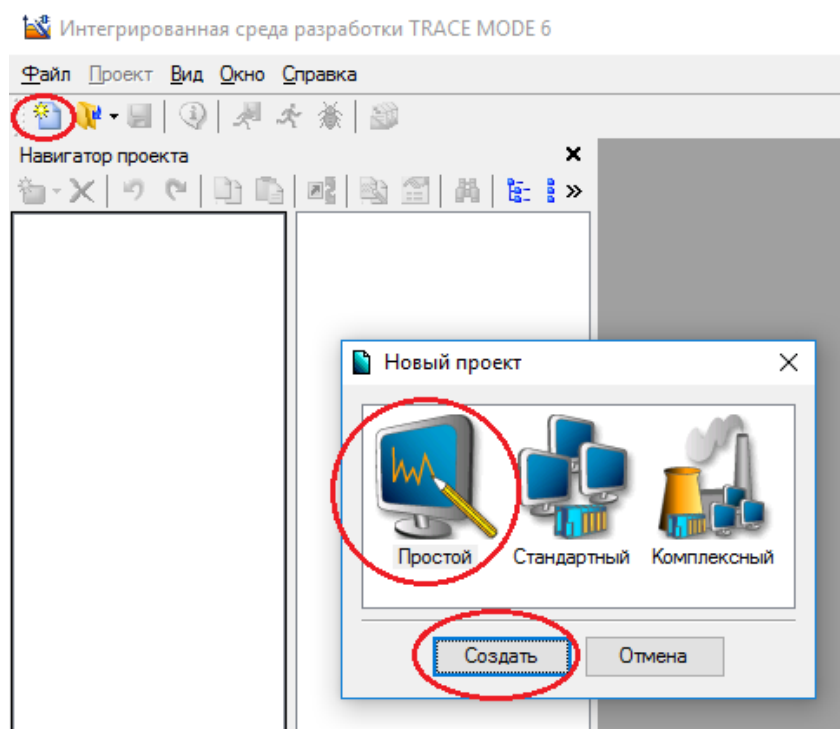


Рис. 58. Создание нового проекта Trace Mode

Для начала обратите внимание на внешний вид мнемосхемы в режиме исполнения проекта, которая в дальнейшем должна получиться путём рисования графических элементов (рис. 59).

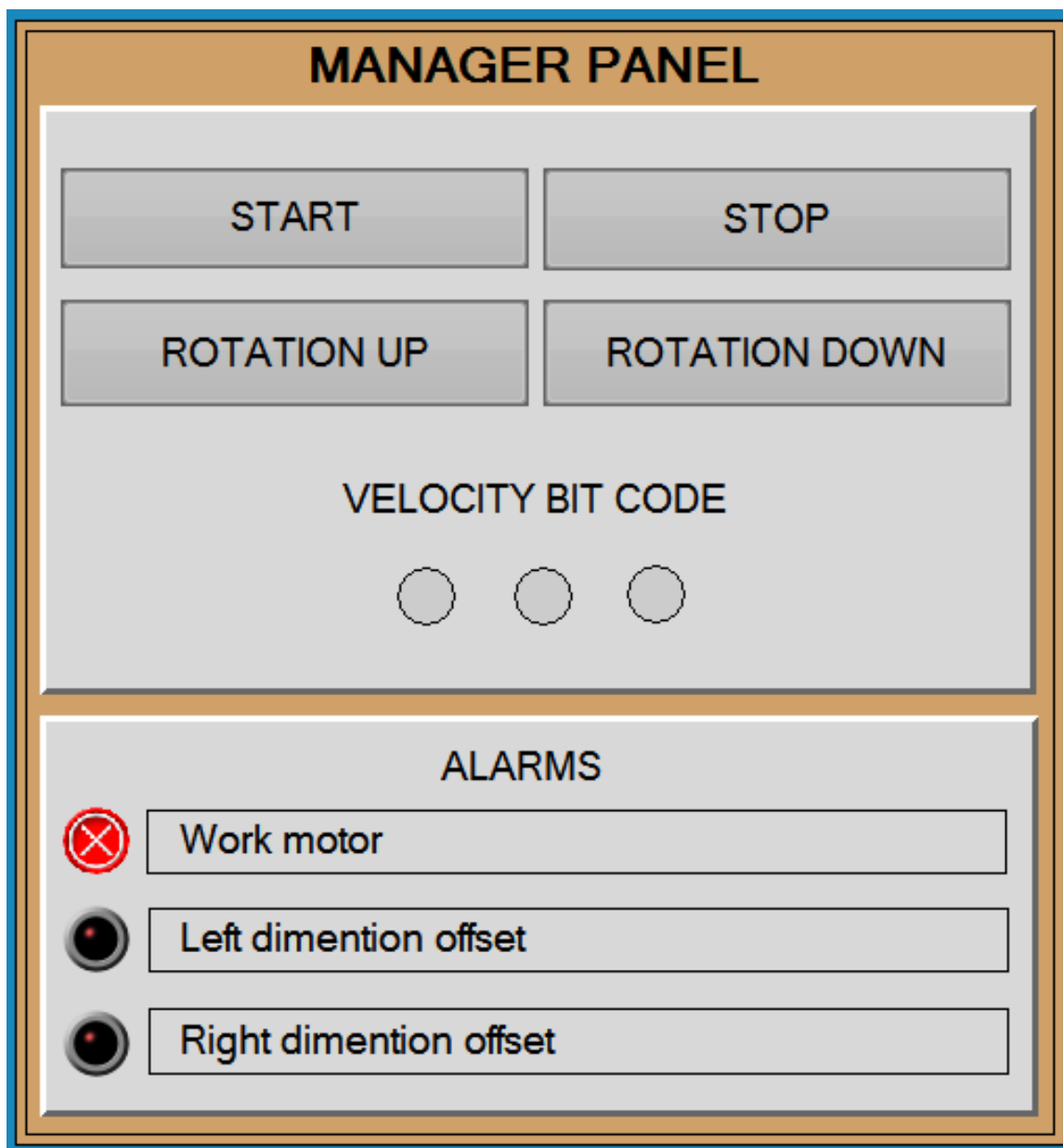


Рис. 59. Мнемосхема с элементами управления

2. *Добавьте каналы.* Для этого нужно добавить как входные, так и выходные каналы с типами HEX16 и FLOAT (рис. 60, 61). Примеры настроек каналов start_button и work_motor приведены на рис. 62, 63.

3. Добавьте OPC группу и OPC сервер в данной группе (рис. 64, 65).

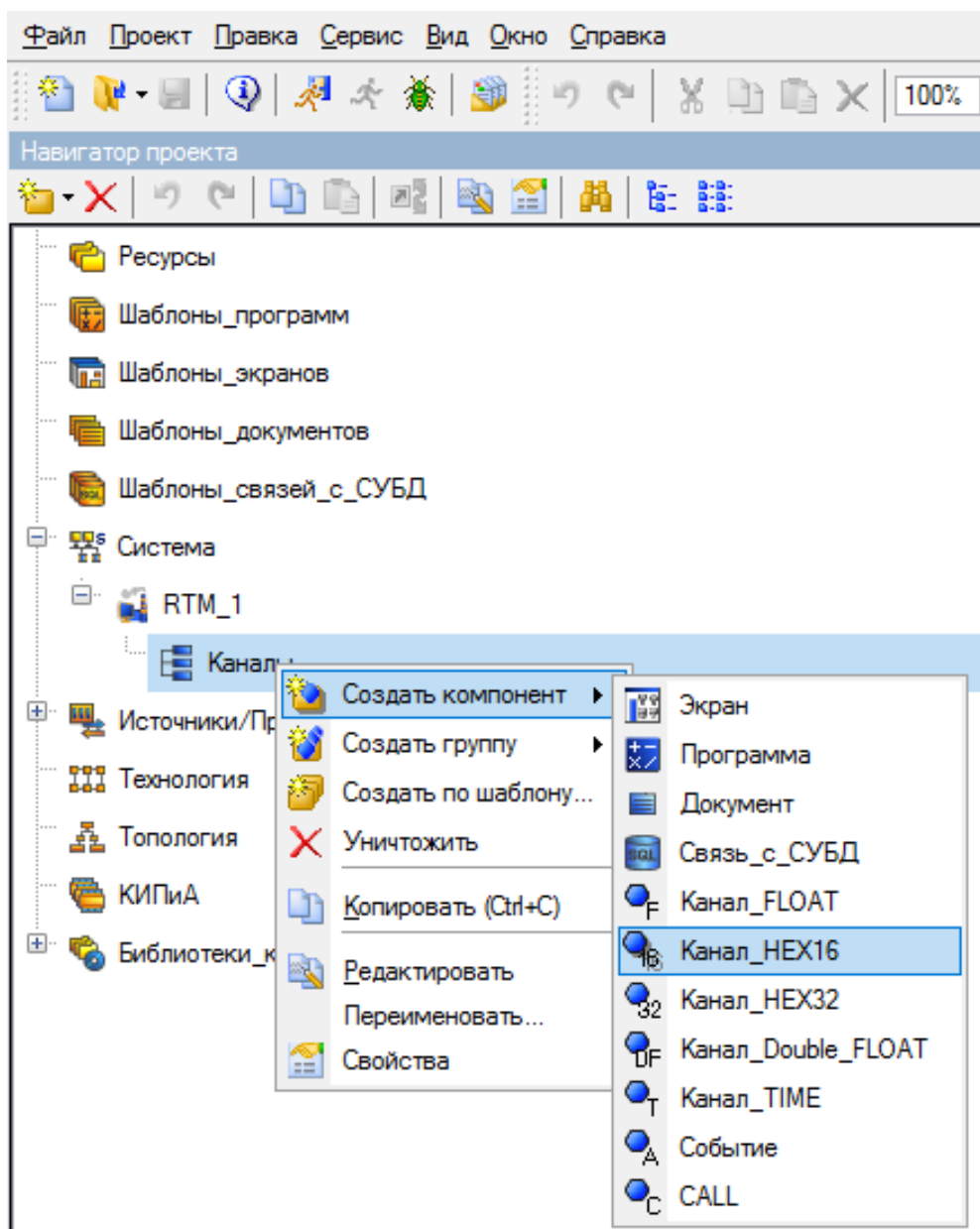


Рис. 60. Добавление нового канала

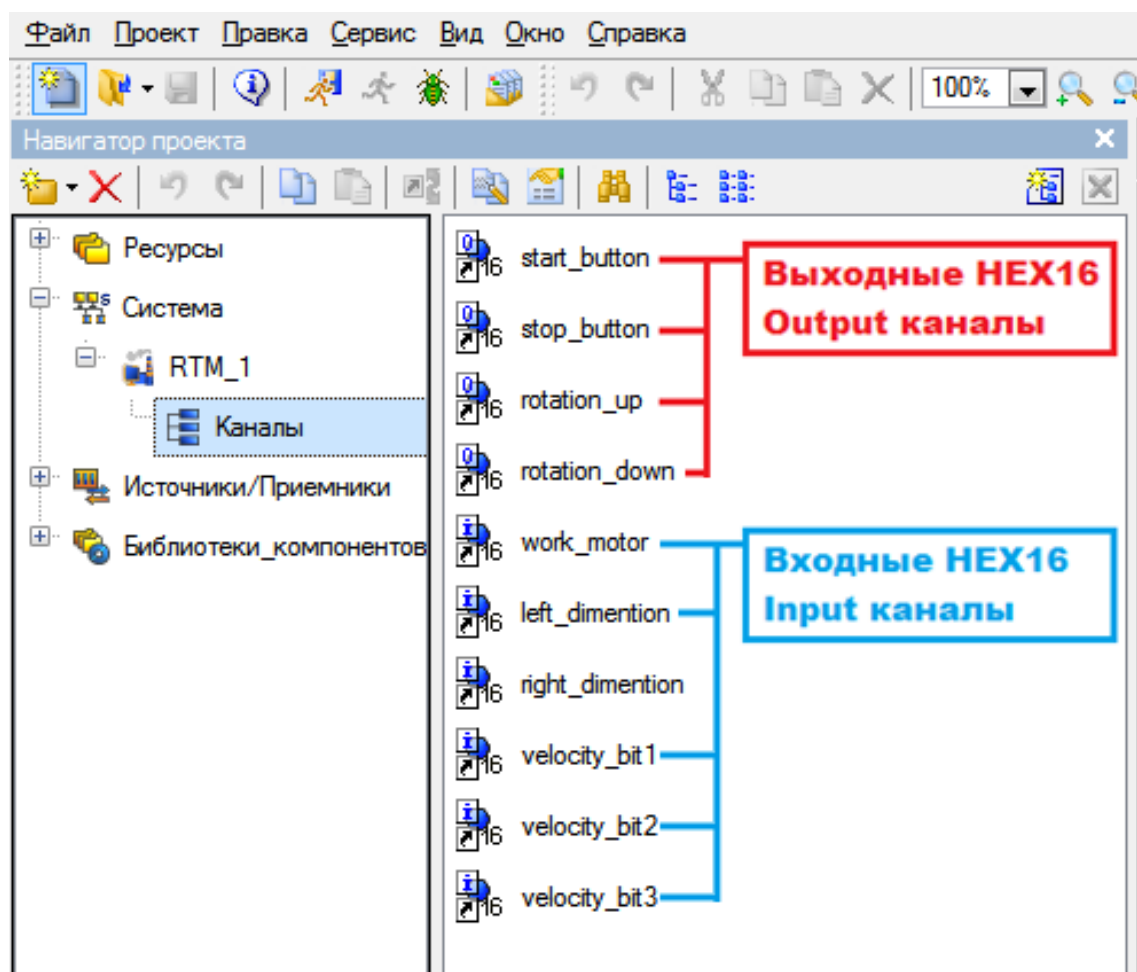


Рис. 61. Список всех необходимых каналов

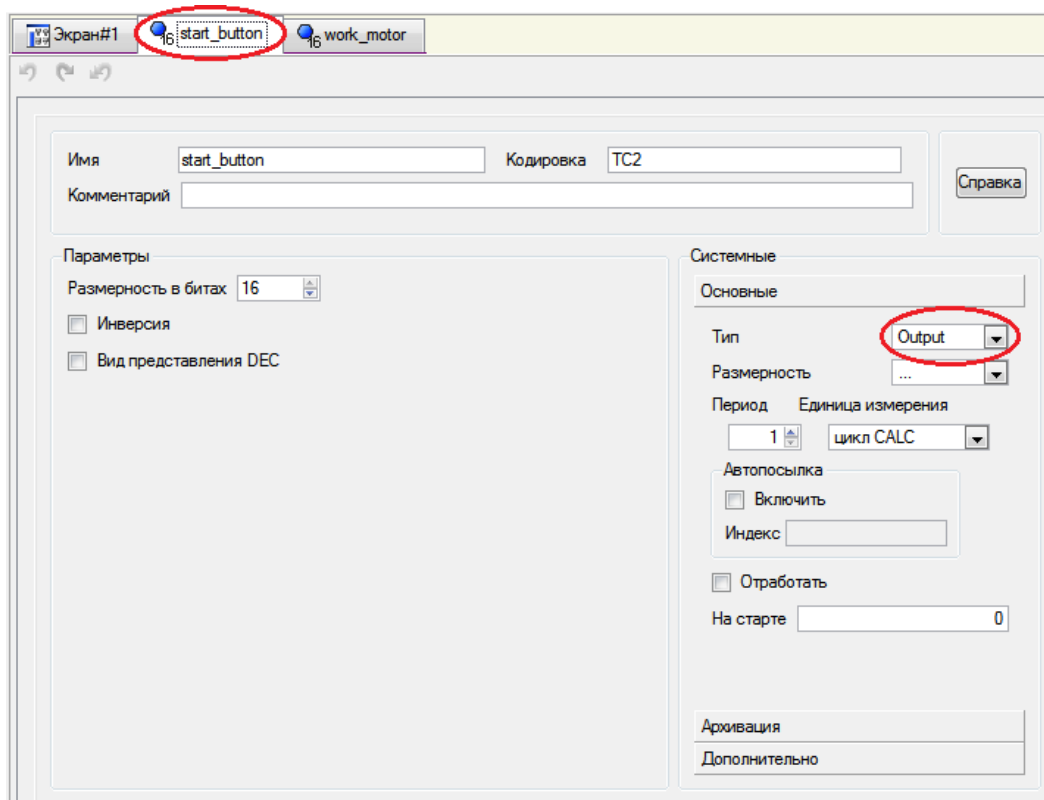


Рис. 62. Пример настроек выходного канала start_button

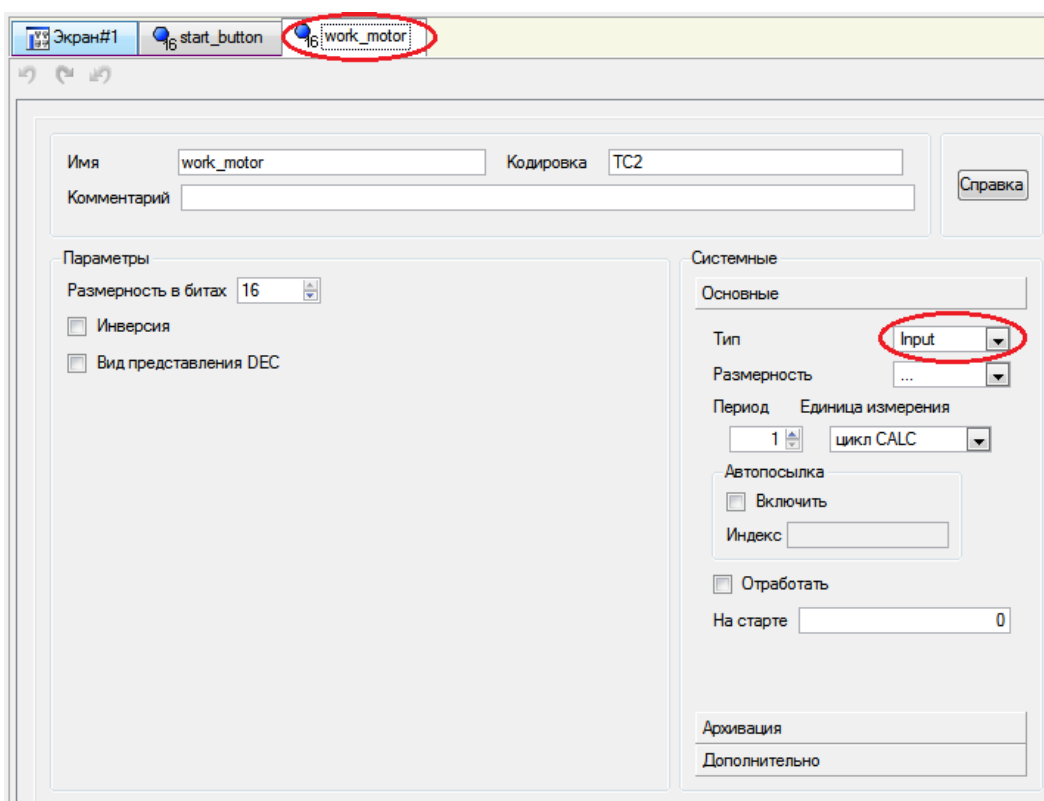


Рис. 63. Пример настроек входного канала work_motor

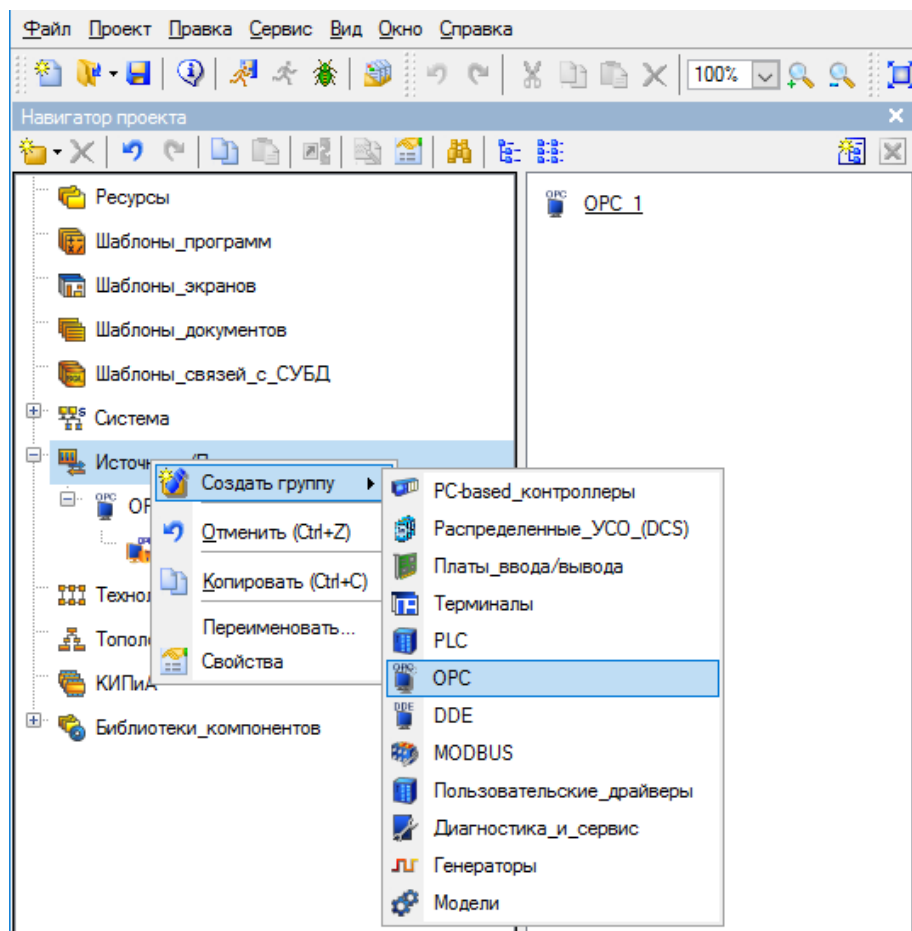


Рис. 64. Добавление OPC группы

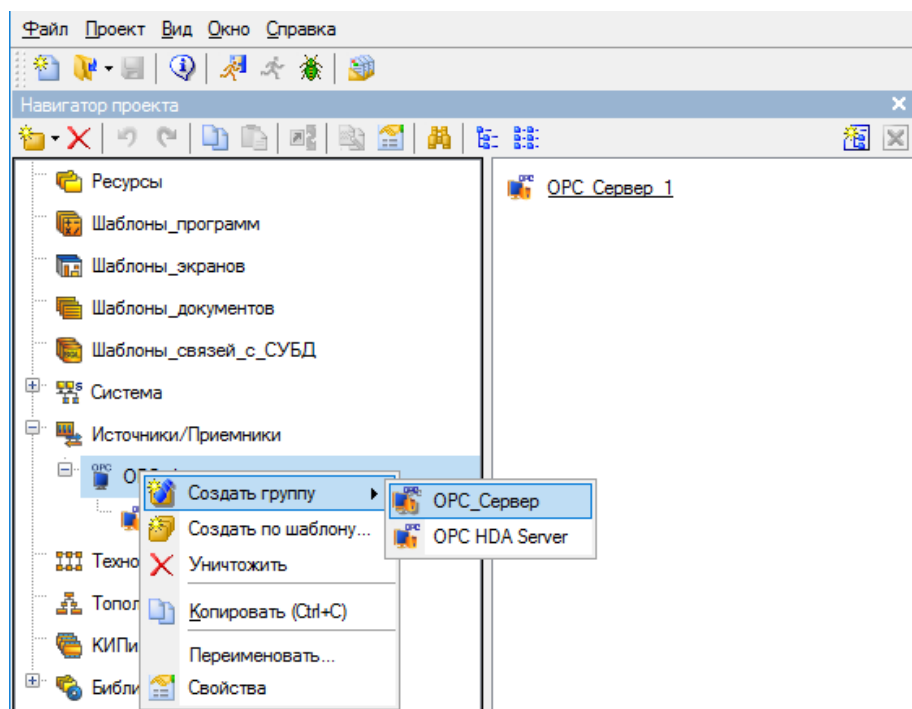


Рис. 65. Добавление OPC сервера

4. *Добавьте OPC источники и приёмники (рис. 66).* Для этого нужно добавить дискретные источники/приёмники сигналов в OPC сервер. На рис. 66 приведён полный список источников/приёмников, для удобного восприятия их имена совпадают с именами ранее добавленных каналов (имя можно задать в окне редактирования после двойного клика клавишей мыши по источнику/приёмнику). К каждому источнику/приёмнику в дальнейшем будет привязан конкретный канал.

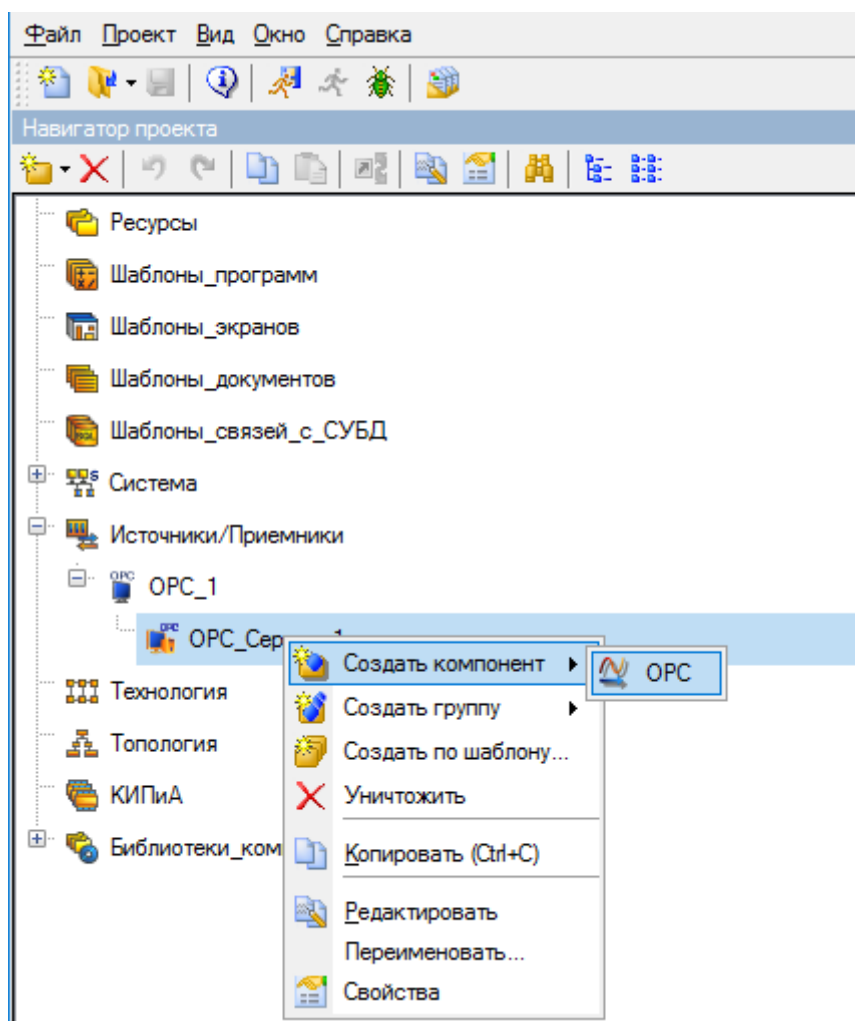


Рис. 66. Добавление OPC источника/приёмника

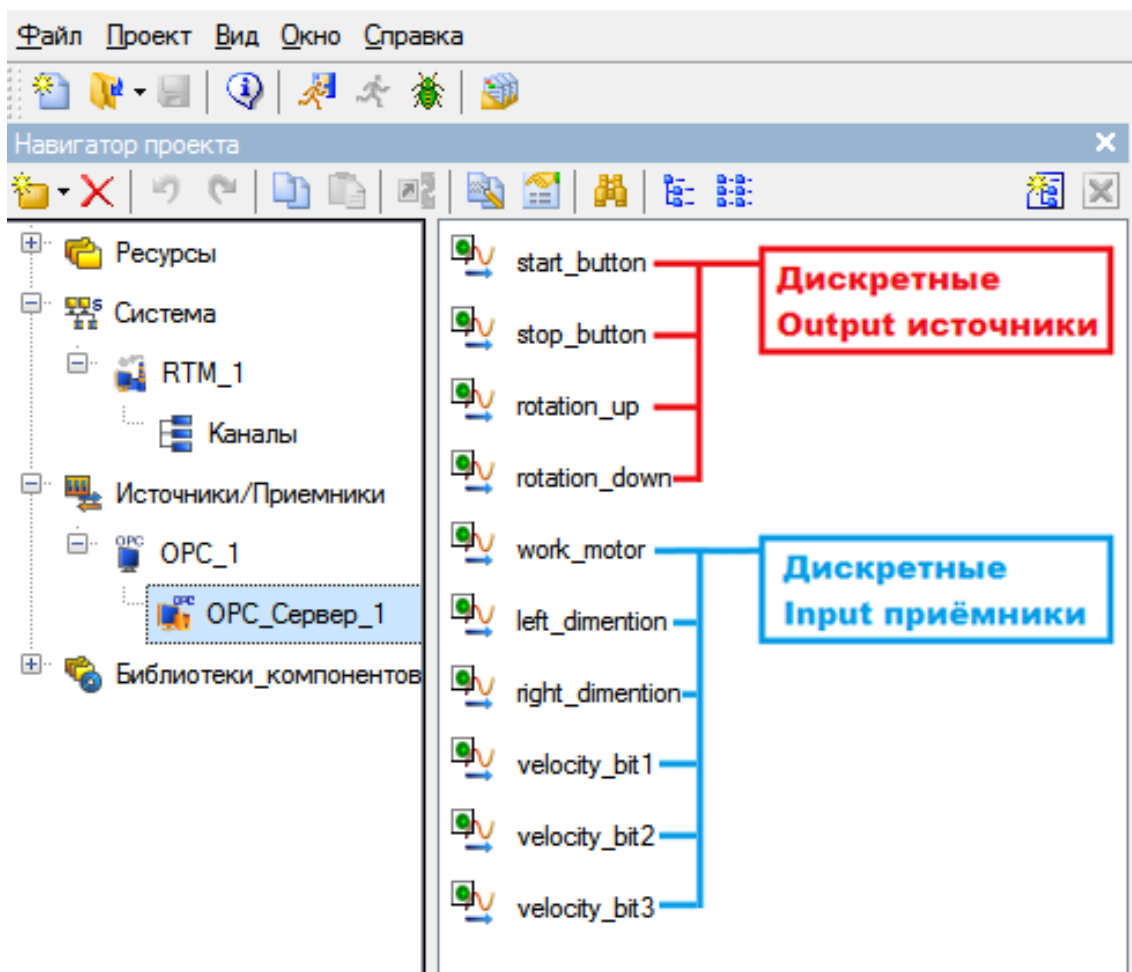


Рис. 67. Полный список источников/приёмников OPC сервера

Создавая нужный источник или приёмник, задайте ему следующие настройки:

- Имя;
- Сервер – CoDeSys.OPC.02 (выбирается кнопкой «Обзор» с последующим выбором тега символьного файла, рис. 68);
- Идентификатор – устанавливается автоматически после выбора тега символьного файла в OPC сервере.
- Направление (Output или Input, ориентируясь на рис. 67).
- Формат – дискрет.

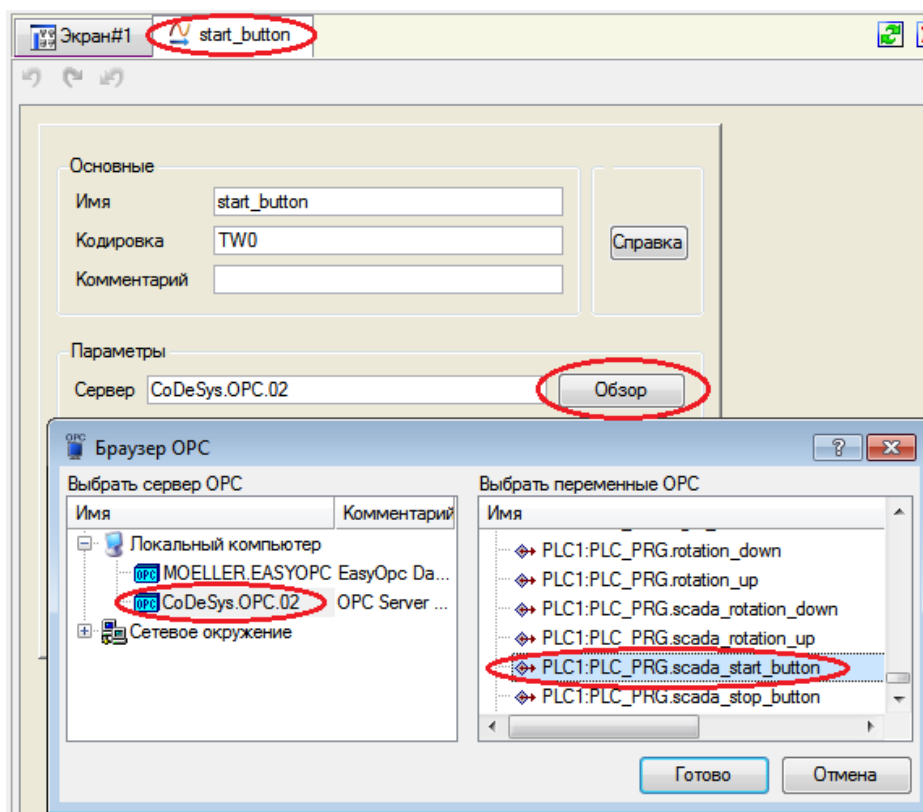


Рис. 68. Выбор тега символьного файла в OPC сервере

ВНИМАНИЕ! Имя OPC тега в символьном файле, к которому привязывается источник или приёмник, может отличаться от имени самого источника или приёмника. Здесь главное понять к какому тегу нужно привязаться на уровне здравого смысла и понимания проекта.

Для удобства можно воспользоваться таблицей 3 для привязки OPC источников/приёмников SCADA системы к тегам символьного файла.

Таблица 3

Привязка OPC источников/приёмников к тегам символьного файла

Имя OPC источника/приёмника в SCADA системе	Направление сигнала	Имя тега в символьном OPC файле
start_button	→	scada_start_button
stop_button	→	scada_stop_button
rotation_up	→	scada_rotation_up
rotation_down	→	scada_rotation_down
work_motor	←	work_motor
left_dimension	←	left_dimension
right_dimension	←	right_dimension
velocity_bit1	←	Q4
velocity_bit2	←	Q5
velocity_bit3	←	Q6

Пример настроек дискретного OPC источника «start_button» приведён на рис. 69, а пример дискретного OPC приёмника «work_motor» на рис. 70.

The screenshot shows the 'start_button' configuration window. The 'Основные' (Basic) section contains fields for 'Имя' (Name) set to 'start_button', 'Кодировка' (Encoding) set to 'TW0', and an empty 'Комментарий' (Comment) field. The 'Параметры' (Parameters) section includes 'Сервер' (Server) set to 'CoDeSys.OPC.02', 'CLSID' set to '{7904C302-AC19-11d4-9E1E-00105A4AB1C6}', 'Идентификатор' (Identifier) set to 'PLC1:PLC_PRG.scada_start_button', 'Режим' (Mode) set to 'SYNC/CACHE', 'Направление' (Direction) set to 'Output', and 'Формат' (Format) set to 'Дискрет' (Discrete). A 'Справка' (Help) button is located on the right.

Рис. 69. Настройки OPC источника «start_button»

The screenshot shows the 'work_motor' configuration window. The 'Основные' (Basic) section contains fields for 'Имя' (Name) set to 'work_motor', 'Кодировка' (Encoding) set to 'TW0', and an empty 'Комментарий' (Comment) field. The 'Параметры' (Parameters) section includes 'Сервер' (Server) set to 'CoDeSys.OPC.02', 'CLSID' set to '{7904C302-AC19-11d4-9E1E-00105A4AB1C6}', 'Идентификатор' (Identifier) set to 'PLC1:PLC_PRG.work_motor', 'Режим' (Mode) set to 'SYNC/CACHE', 'Направление' (Direction) set to 'Input', and 'Формат' (Format) set to 'Дискрет' (Discrete). A 'Справка' (Help) button is located on the right.

Рис. 70. Настройки OPC приёмника «work_motor»

5. Откройте свойства каждого канала и сделайте его привязку к одноимённым источникам/приёмникам (рис. 71). Пример привязки выходного канала «start_button» к OPC источнику «start_button» показан на рис. 72.

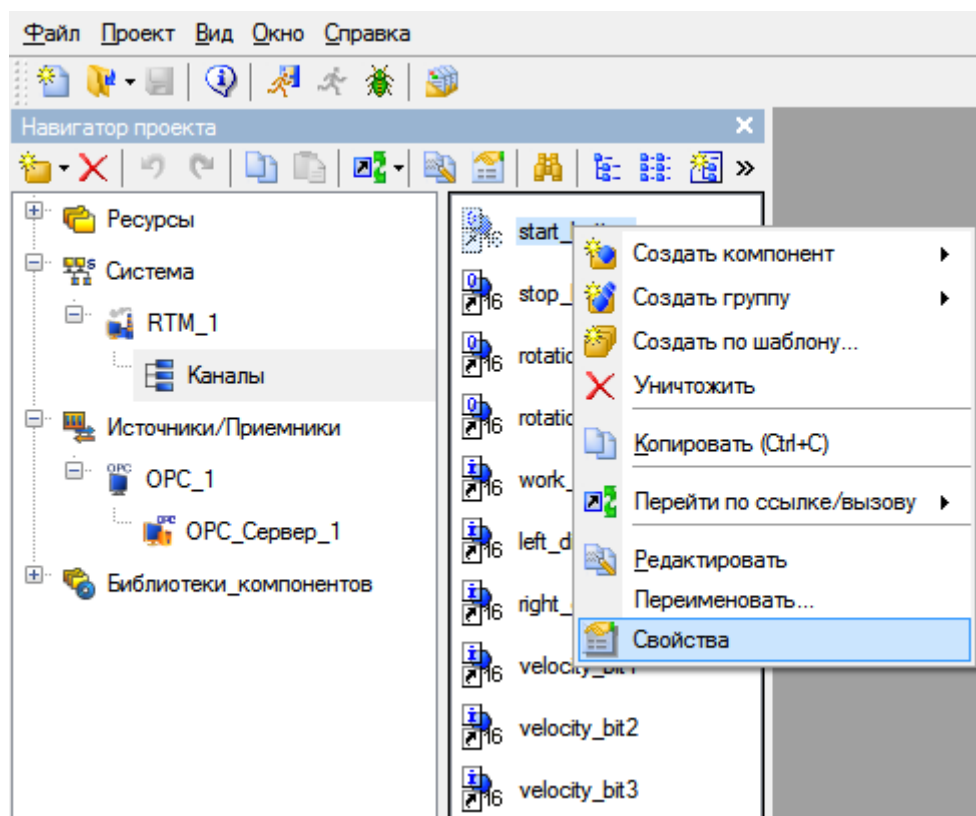


Рис. 71. Открытие свойств канала

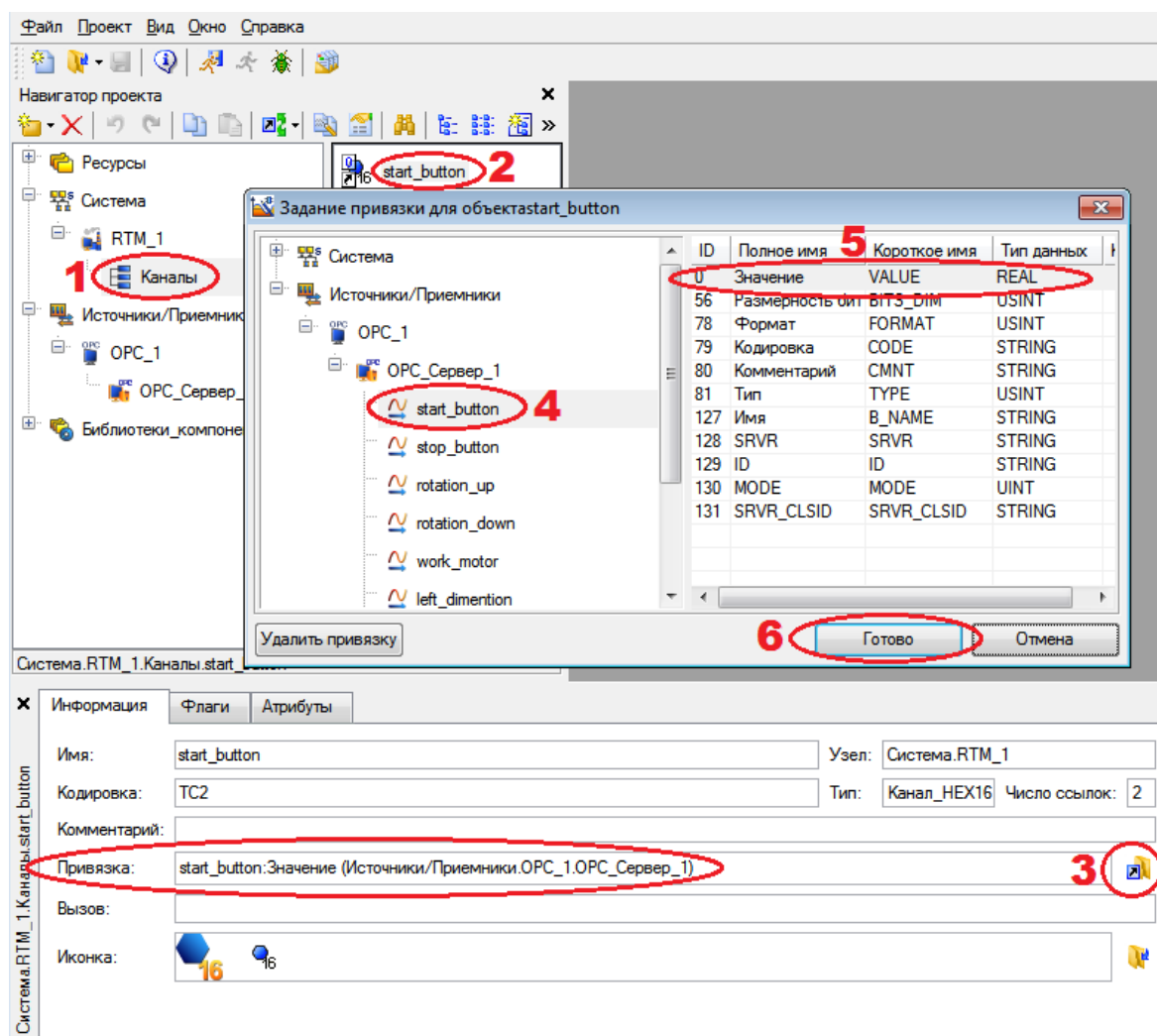


Рис. 72. Привязка канала к OPC источнику

6. Нарисуйте графические элементы (рис. 59), используя инструменты рисования (рис. 73).



Рис. 73. Элементы графического рисования

7. Сделайте настройки и привязки графических элементов к каналам. Пример настроек и привязки кнопки «START» приведён на рис. 74, 75. Чтобы задать привязку, нужно кликнуть левой клавишей мыши по «Результат» и «Источник», далее откроется окно (рис. 75), в котором нужно добавить новую строку в таблице для текущей привязки, задать тип, тип данных и открыть окно привязки (рис. 76). **ВНИМАНИЕ!** В таблице (рис. 77) целый ряд привязок для отдельных графических элементов. Для конкретного графического элемента

строка, после её добавления и **выбора**, будет подсвечиваться синим фоном. Т.е. это некая карта привязок графических элементов. **Обратите внимание** – в таблице в колонке привязки может быть как входное значение, так и реальное значение, в зависимости от направления сигналов.

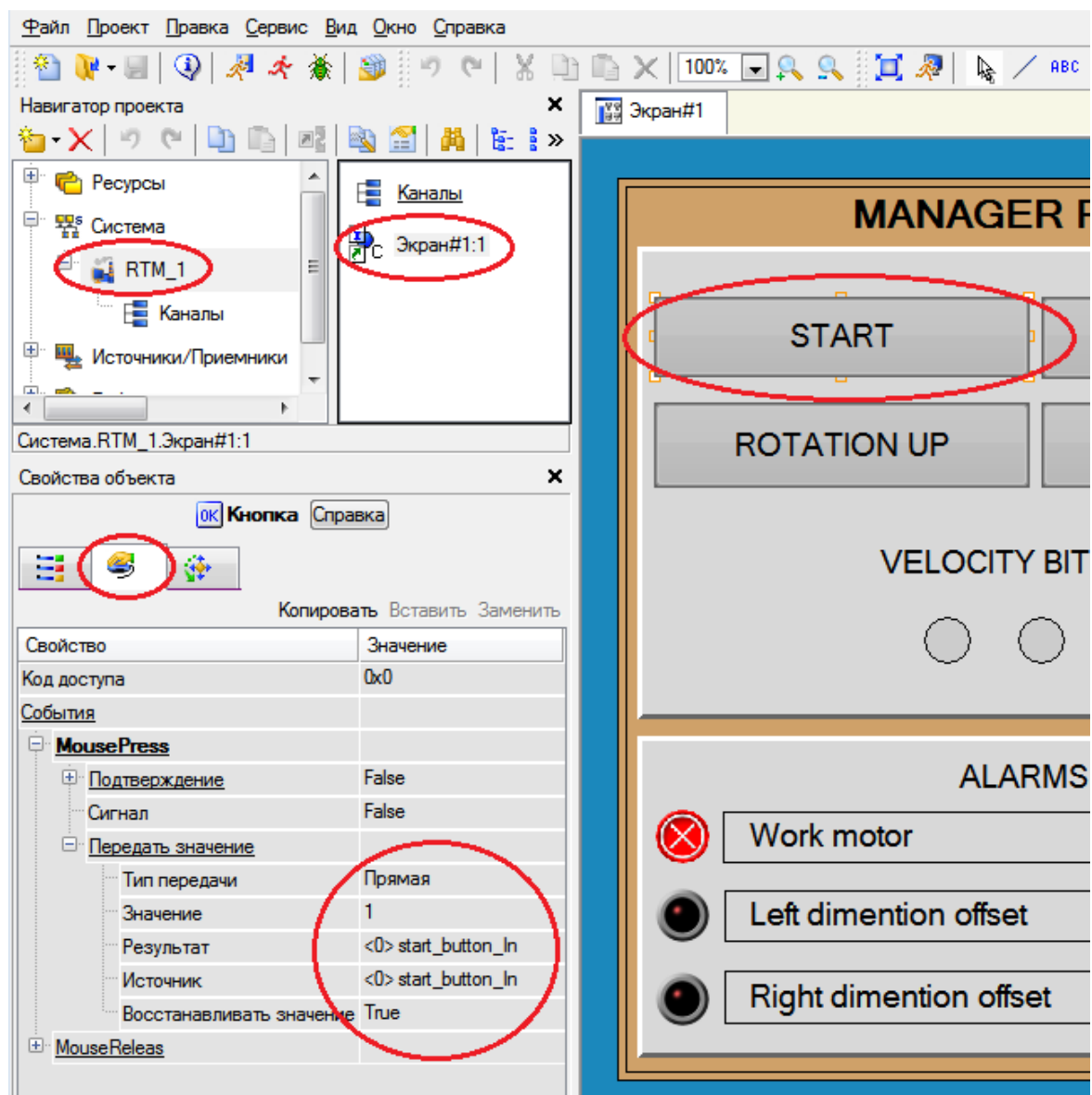


Рис. 74. Настройки кнопки «START»

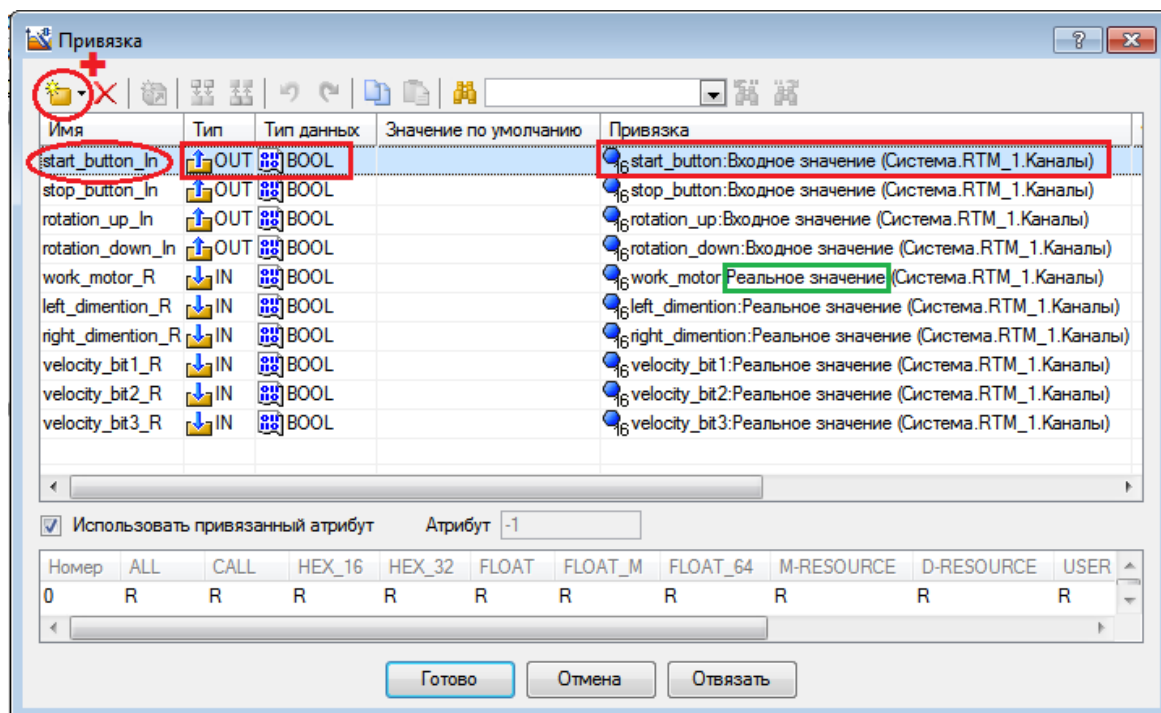


Рис. 75. Создание привязки графического элемента к каналу

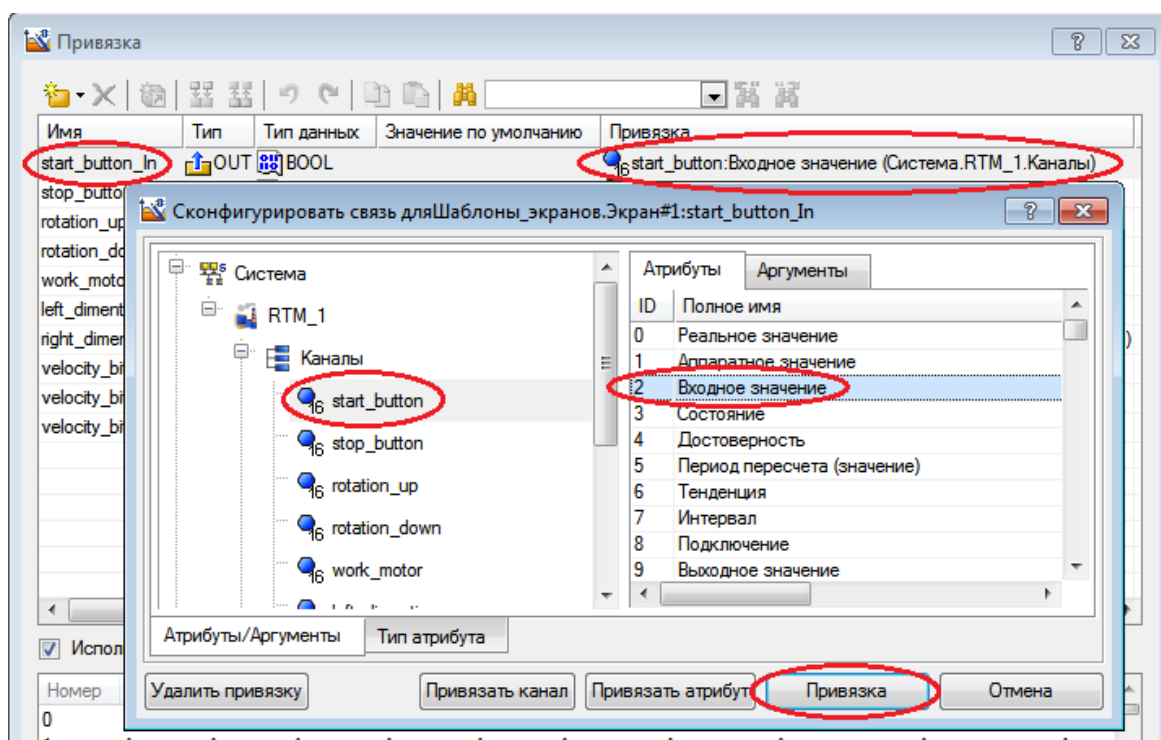


Рис. 76. Настройка привязки к каналу

На рис. 77-82 приведены примеры свойств различных графических элементов мнемосхемы.

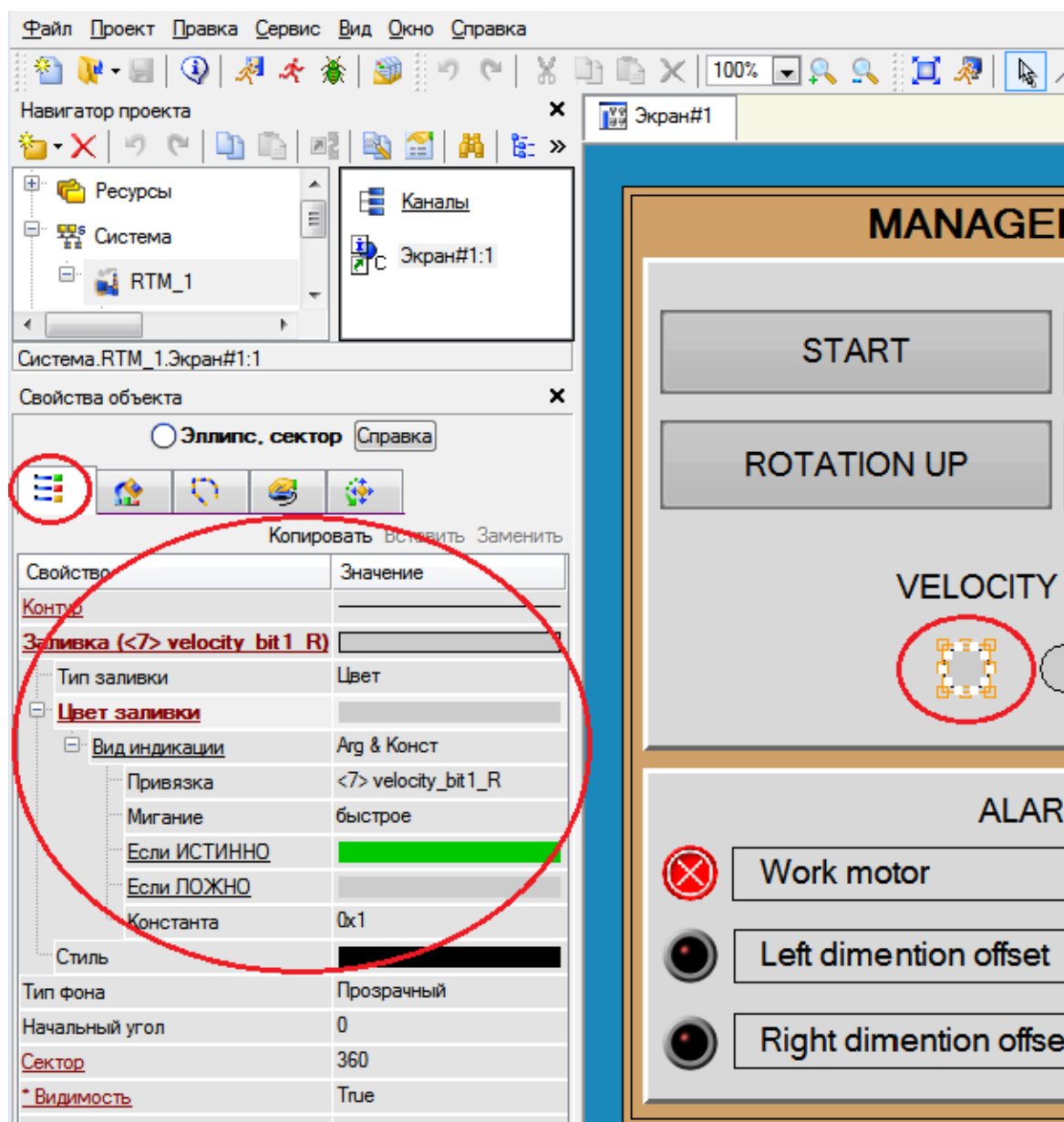


Рис. 77. Свойства фигуры показания кода скорости

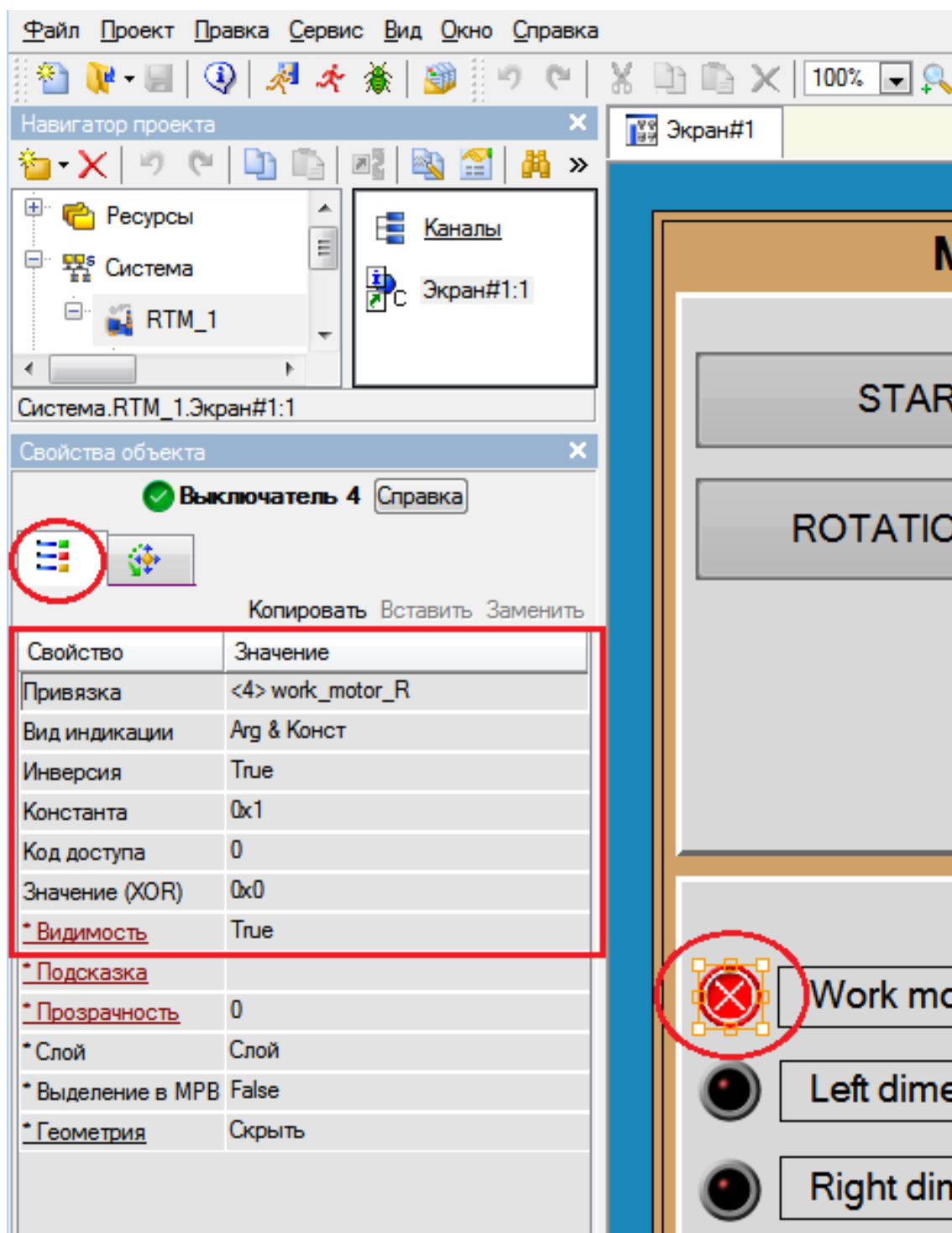


Рис. 78. Свойства флажкового индикатора работы двигателя

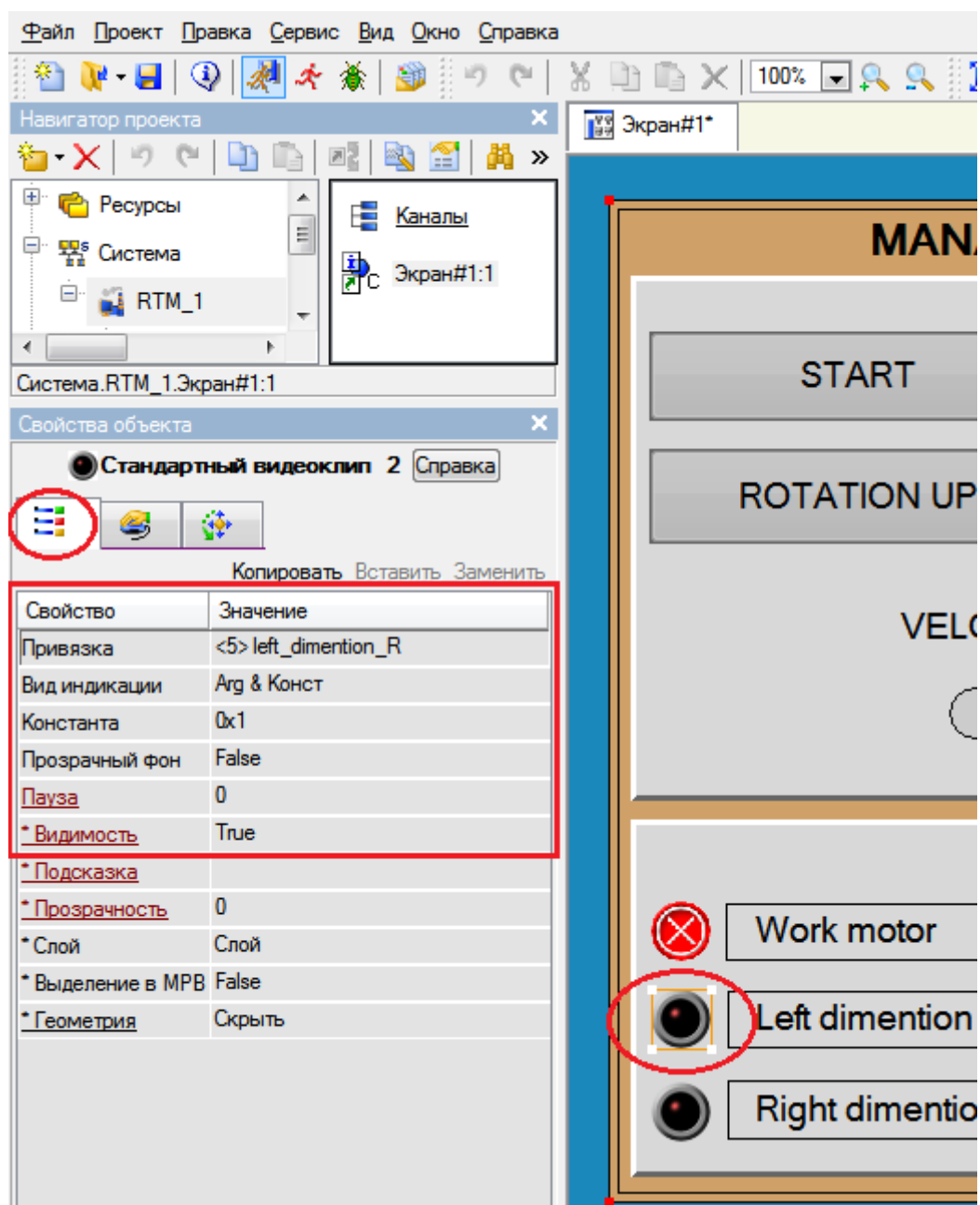


Рис. 79. Свойства сигнализационной лампы левого габарита

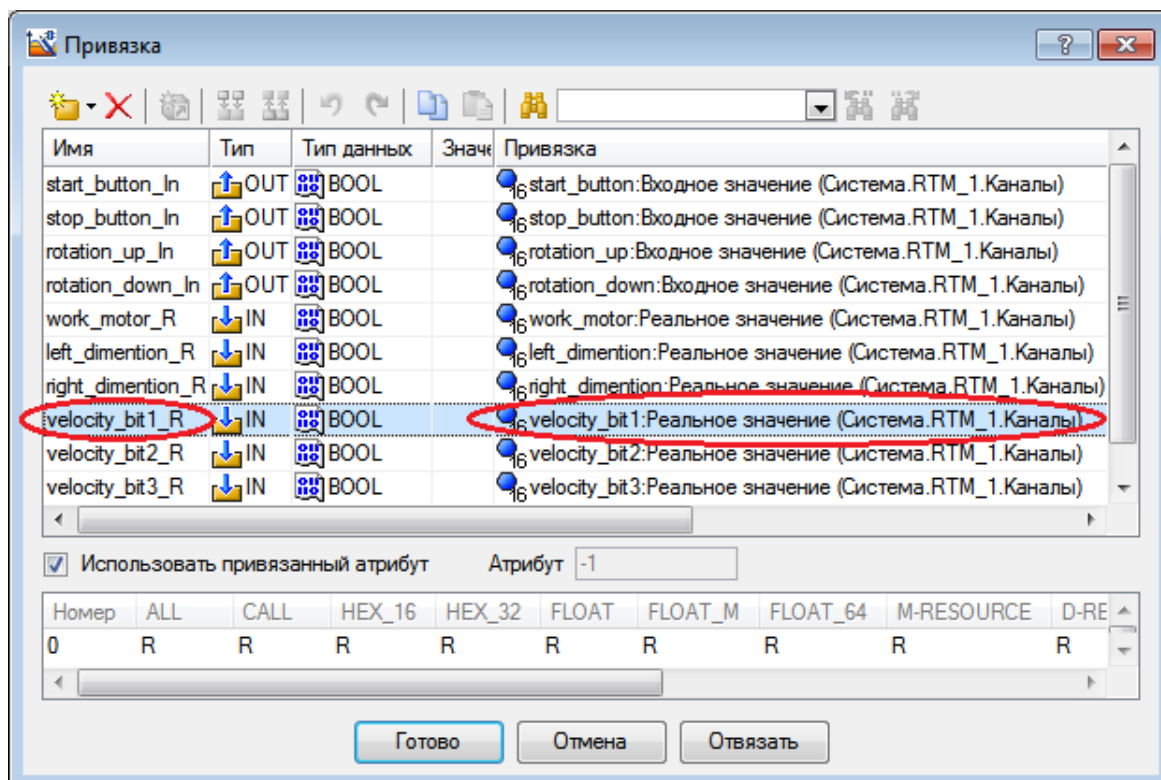


Рис. 80. Привязка фигуры показания кода скорости к каналу

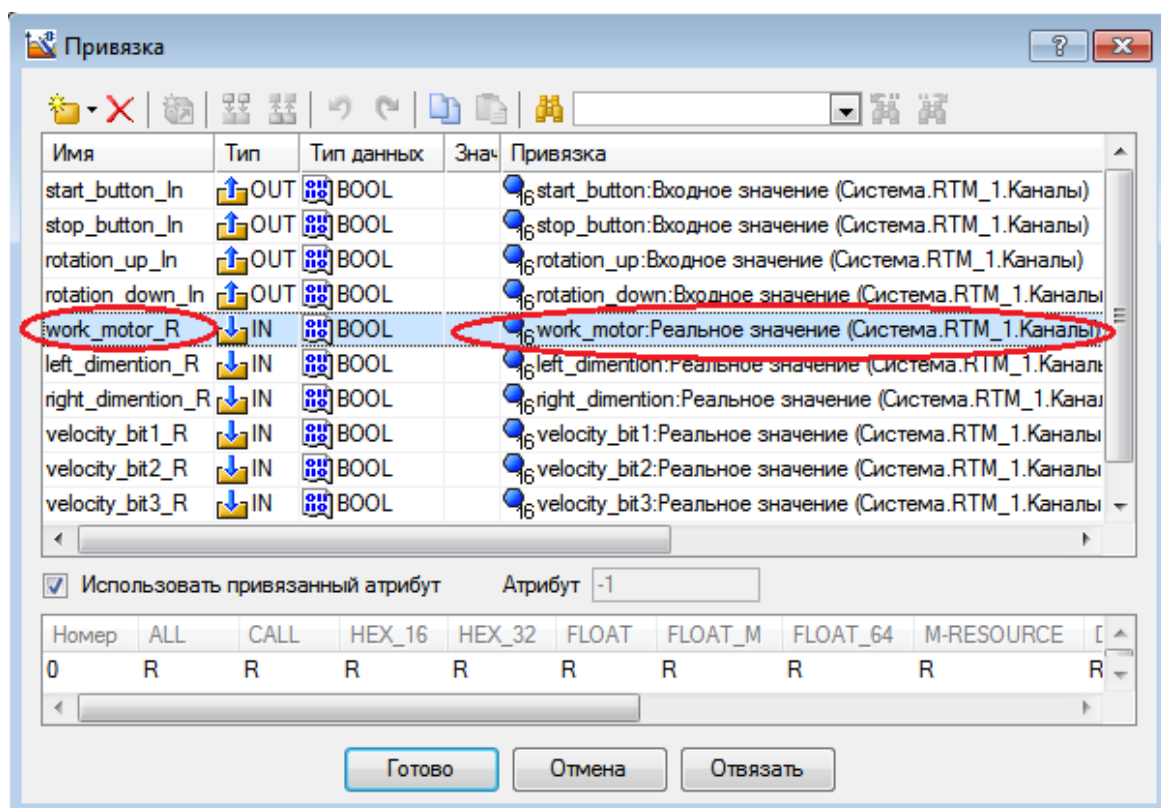


Рис. 81. Привязка флажкового индикатора работы двигателя к каналу

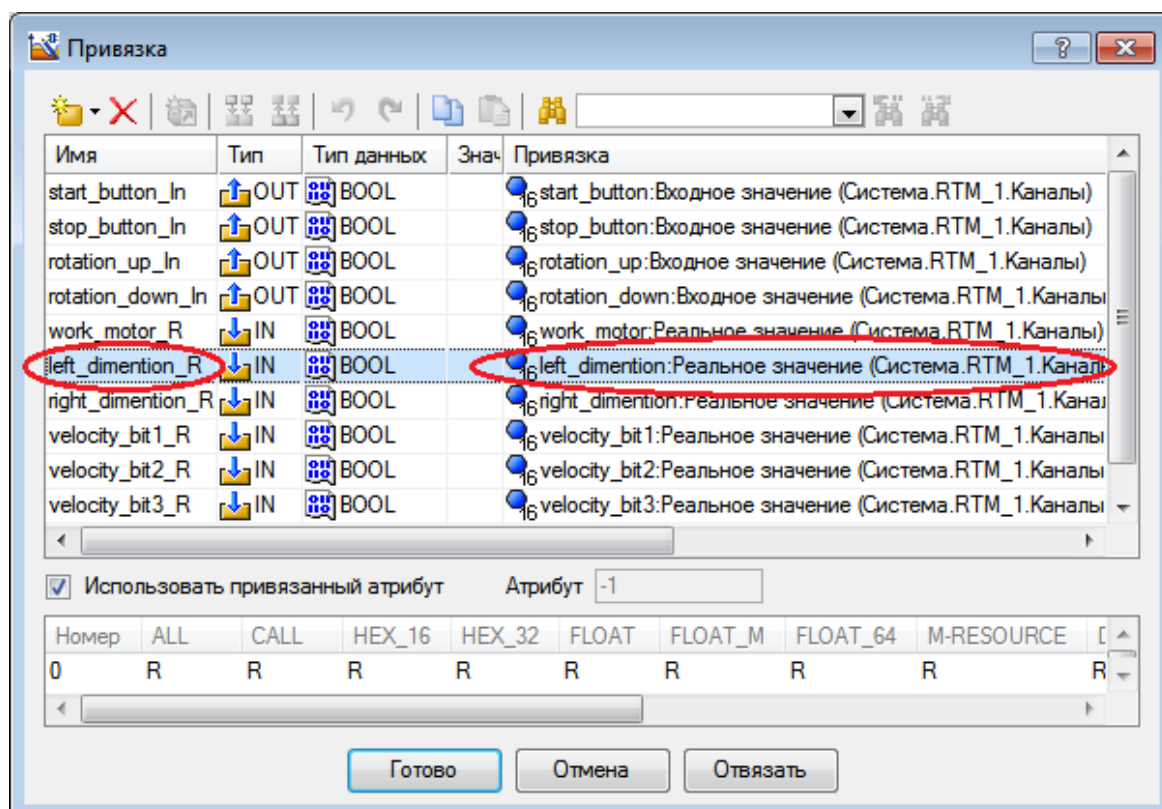


Рис. 82. Привязка лампы левого габарита к каналу

8. Проверьте работоспособность мнемосхемы в режиме эмуляции (рис. 83).

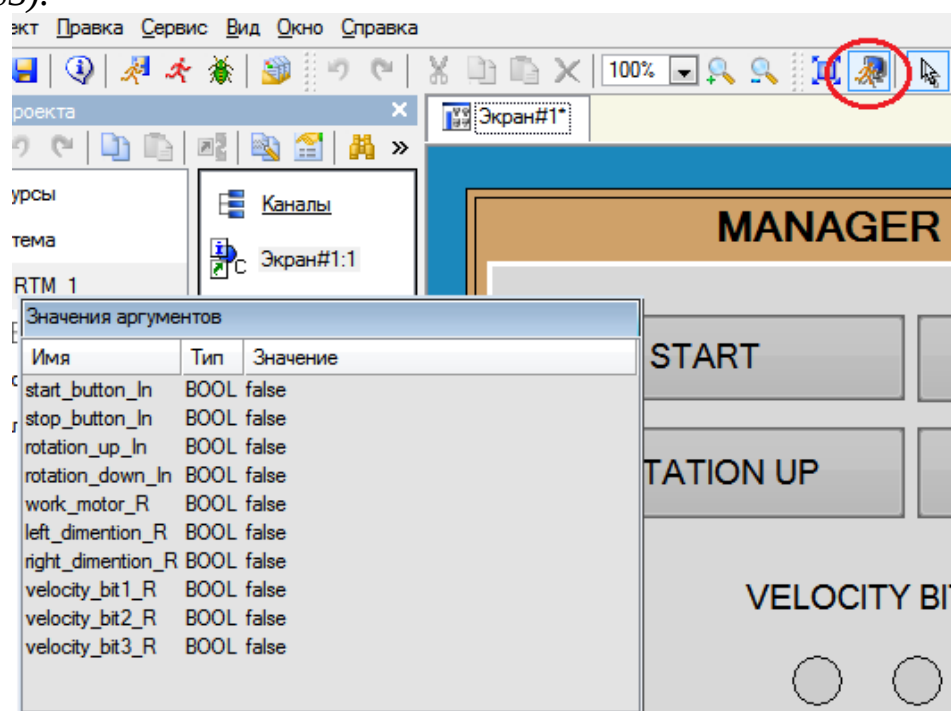


Рис. 83. Проверка мнемосхемы в режиме эмуляции

2.7. Настройка OPC сервера

Чтобы OPC сервер знал с какими контроллерами нужно взаимодействовать, необходимо добавить их в конфигурацию OPC.

1. Запустите OPCConfig и добавьте контроллер (рис. 84).

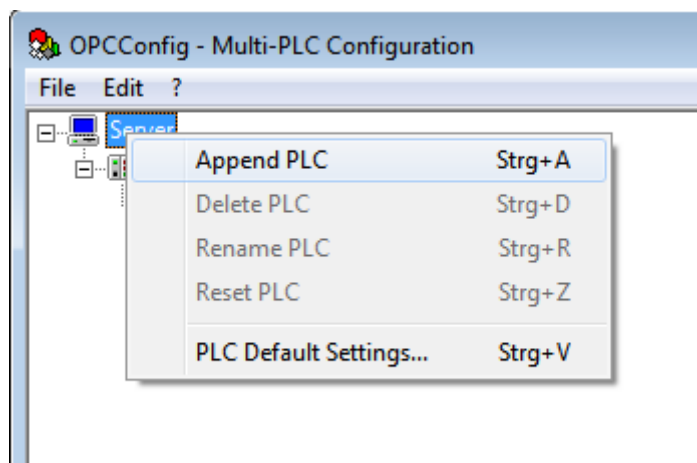


Рис. 84. Добавление контроллера в OPC конфигурацию

2. Задайте настройки контроллера (рис. 85).

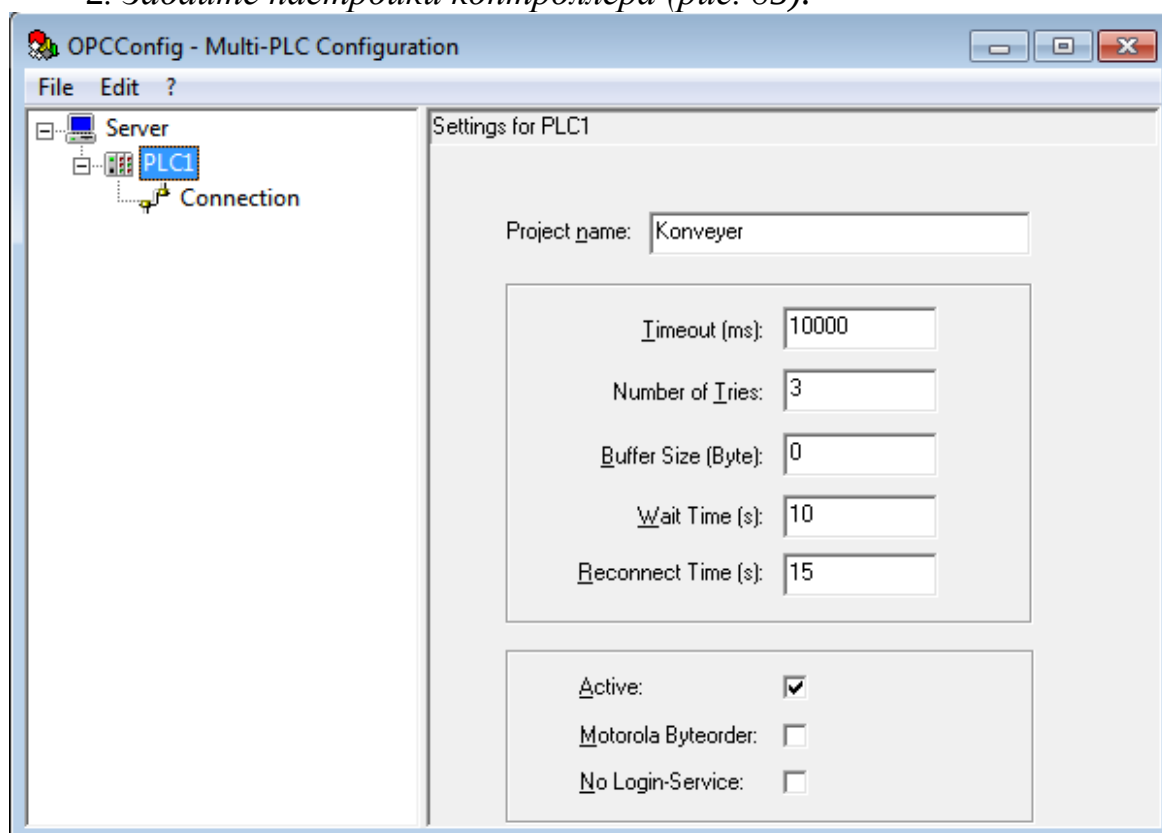


Рис. 85. Настройки контроллера

3. Отредактируйте соединение с контроллером (рис. 86). Задайте его IP адрес, по которому OPC сервер будет обращаться к контроллеру в соответствии с картой IP адресов (рис. 3), в данном случае 192.168.119.23.

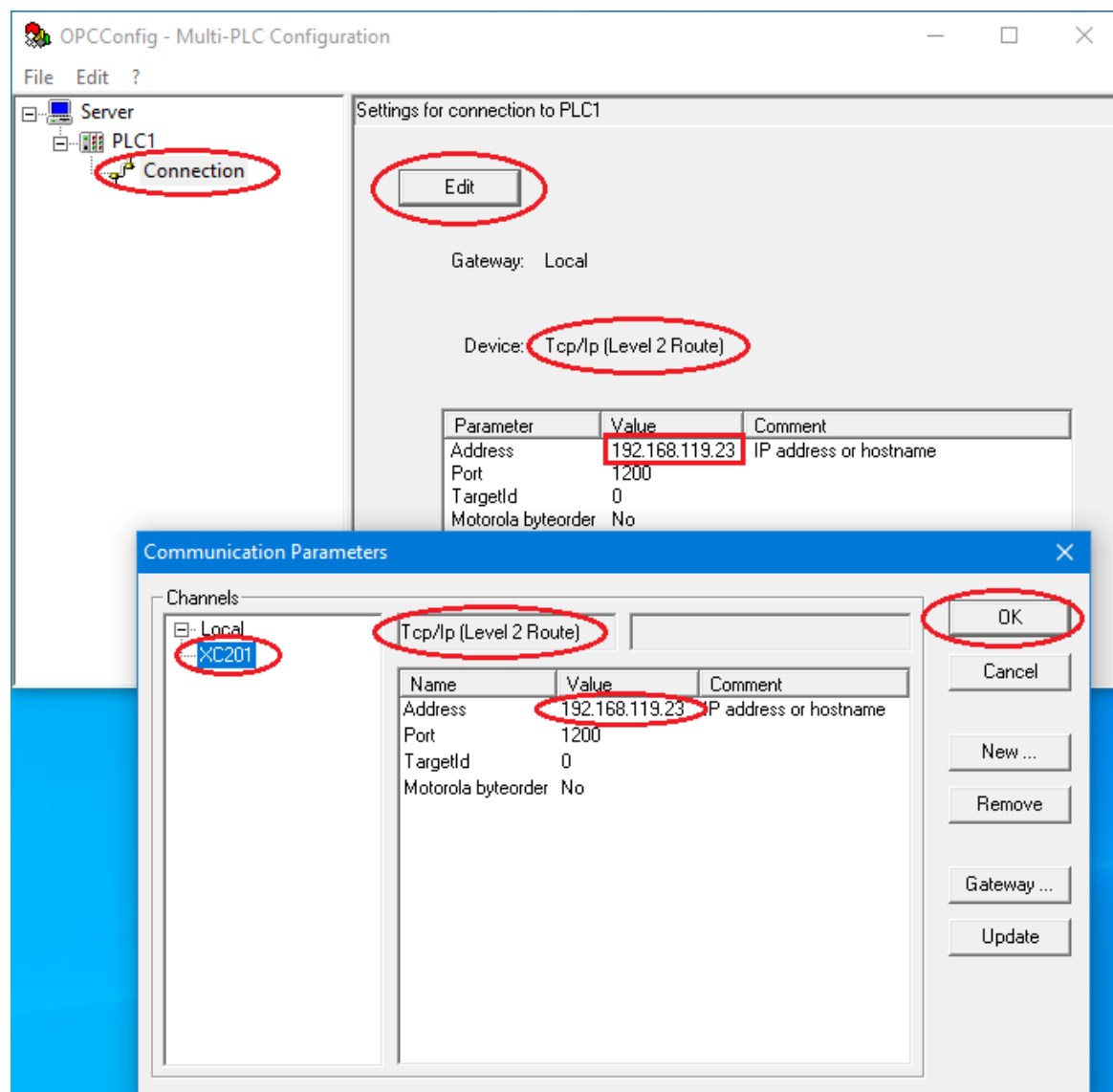


Рис. 86. Редактирование соединения с контроллером

2.9. Аprobация созданной системы управления

1. Создайте электрическое соединение лабораторной установки с контроллером. Соедините необходимые провода между датчиками физической модели и входами «I» контроллера (рис. 87). Также соедините необходимые провода между входами исполнительных элементов физической модели и выходами «Q» контроллера. Данные соединения должны производиться в соответствии с назначением переменных физических входов/выходов в редакторе объявлений PLC_PRG. Объявленные переменные соответствуют определённым датчикам и исполнительным элементам физической модели.

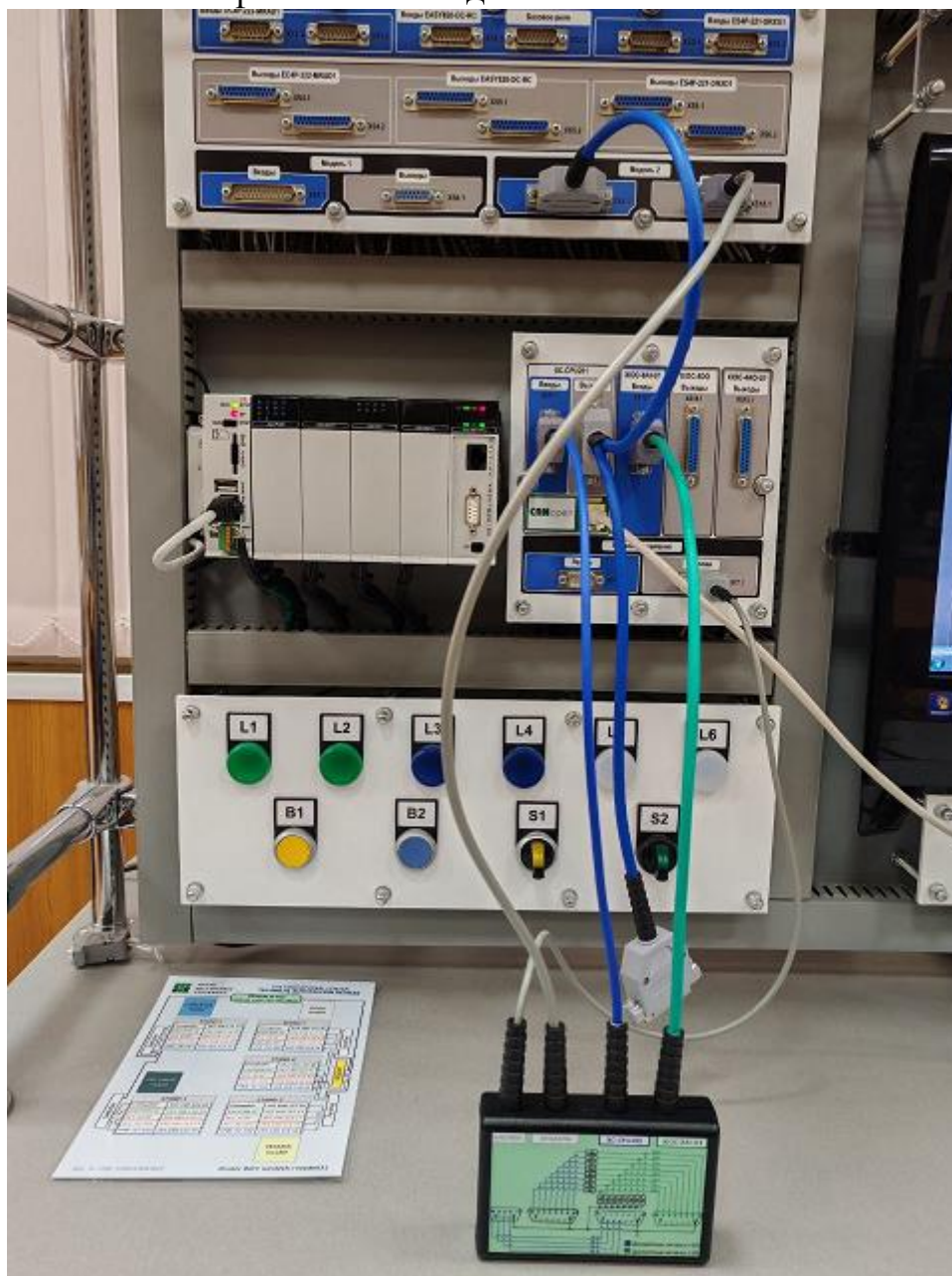


Рис. 87. Подключение контроллера к модели конвейера

Соедините провод передачи данных CAN Open между контроллером XC-CPU201 и HMI панелью XV102 (рис. 88).

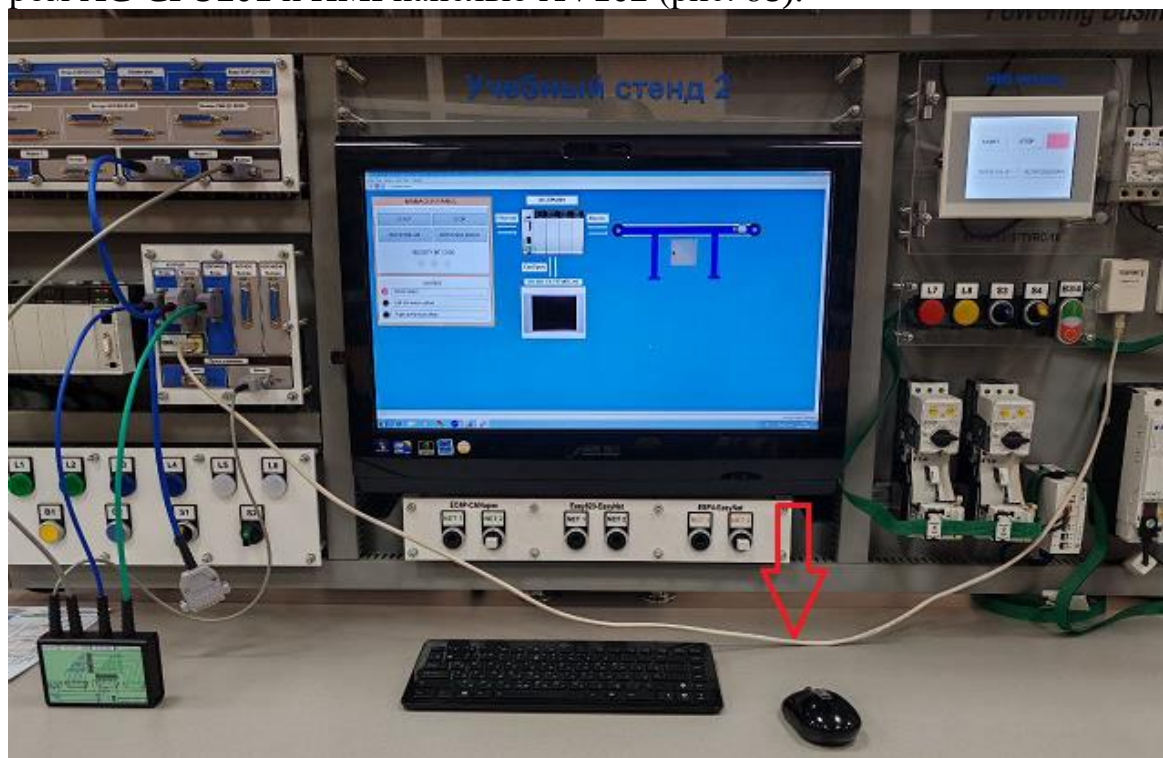


Рис. 88. Подключение провода CAN Open

2. Проверьте соединение с контроллером XC-CPU201. Во вкладке «Онлайн» (Online) откройте диалог «Параметры соединения» (Communication parameters) и удостоверьтесь в наличии настройки TCP/IP соединения с IP-адресом в соответствии с картой IP адресов (рис. 3), в данном случае 192.168.119.23. Установите соединение с контроллером: «Онлайн/Соединение» (Online/Login). CODESYS попросит вас подтвердить загрузку (download) кода проекта. Также во вкладке «Онлайн» (Online) выберите опцию «Создать загрузочный проект» (Create boot project).

4. Запустите контроллер XC-CPU201: «Онлайн/Запуск» (Online/Run).

5. Проверьте соединение с внутренним контроллером HMI панели XV102. Во вкладке «Онлайн» (Online) откройте диалог «Параметры соединения» (Communication parameters) и удостоверьтесь в наличии настройки TCP/IP соединения с IP-адресом в соответствии с картой IP адресов (рис. 3), в данном случае 192.168.119.24. Установите соединение с контроллером: «Онлайн/Соединение» (Online/Login). CODESYS попросит вас подтвердить загрузку (download) кода проекта. Также во вкладке «Онлайн» (Online) выберите опцию «Создать загрузочный проект» (Create boot project).

6. Запустите внутренний контроллер HMI панели XV102: «Он-лайн/Запуск» (Online/Run).

7. Запустите профайлер среды SCADA TRACE MODE в режиме реального времени. Для этого сначала сохраните проект для монитора реального времени (рис. 89), а затем запустите профайлер (рис. 90).

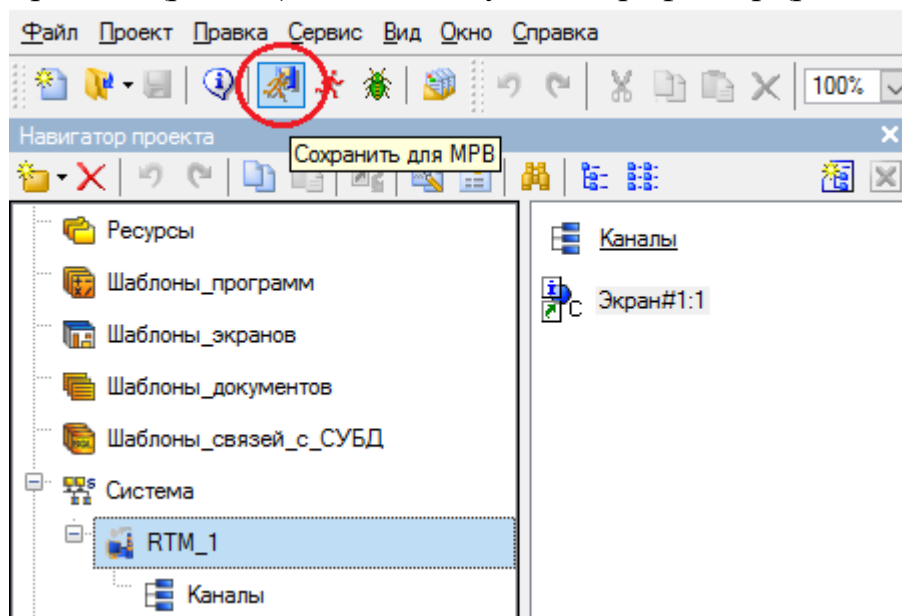


Рис. 89. Сохранение для монитора реального времени

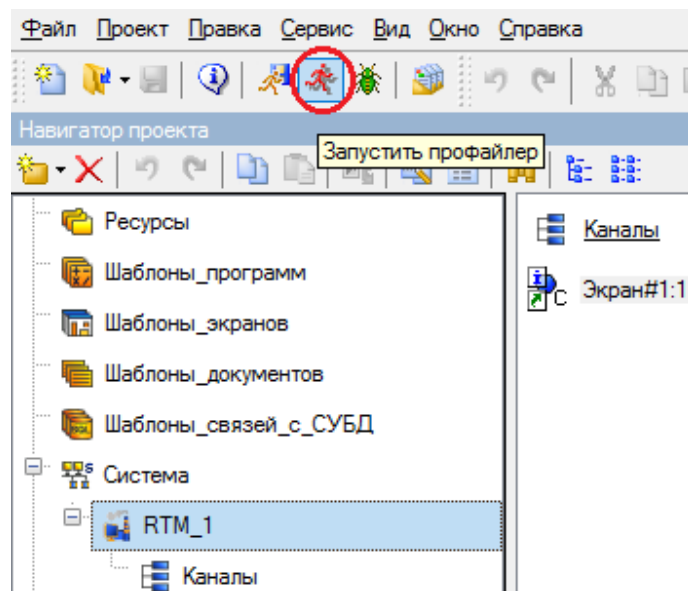


Рис. 90. Запуск профайлера TRACE MODE

В итоге запустится профайлер с работающим монитором реального времени (MPV), где можно просматривать состояние компонентов и полноценно управлять всей системой конвейера, нажимая на кнопки, задавая нужную скорость вращения двигателя (рис. 91).

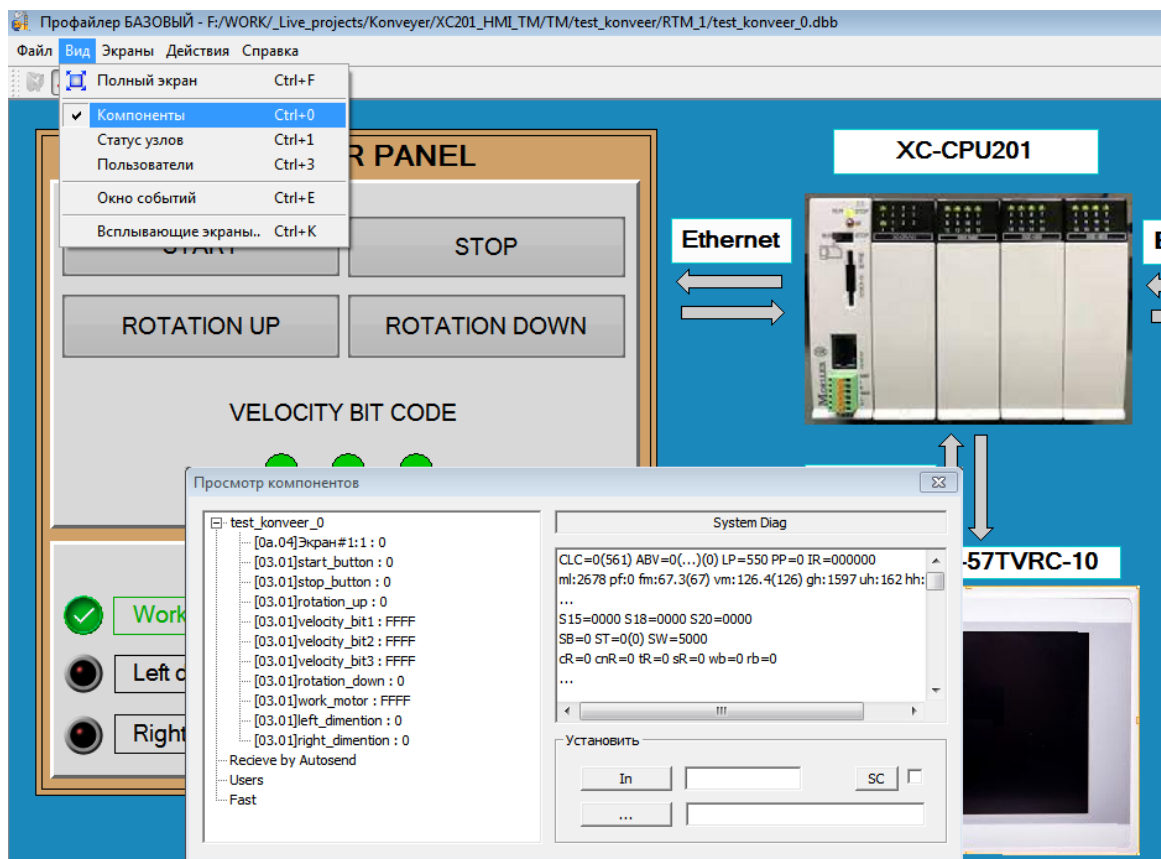


Рис. 91. Работающий монитор реального времени (МРВ)

8. Проверьте работоспособность системы. С кнопок №3-4 станда задайте необходимую частоту вращения двигателя привода ленты конвейера. С помощью кнопки №1 на стенде запустите двигатель. Обратите внимание, что при низкой частоте вращения двигателя, в программе управления должна сработать автоматическая остановка двигателя (это блокировка от пониженной частоты вращения). Поставьте транспортируемые грузы разной высоты на движущуюся ленту конвейера, грузы должны отсортироваться по разным корзинам. Расположение груза за пределами контроля габаритов ленты конвейера должно привести к остановке двигателя.

Результат всех выполняемых манипуляций должен отображаться на мнемосхеме в SCADA системе TRACE MODE (рис. 59).

Дополнительно проверьте передачу значений тегов (переменных) через OPC сервер при включении тех или иных механизмов конвейера. Для этого запустите OPC обозреватель (MatrikonOPC Explorer) и установите соединение с OPC сервером CoDeSys.OPC.02 (рис. 92).

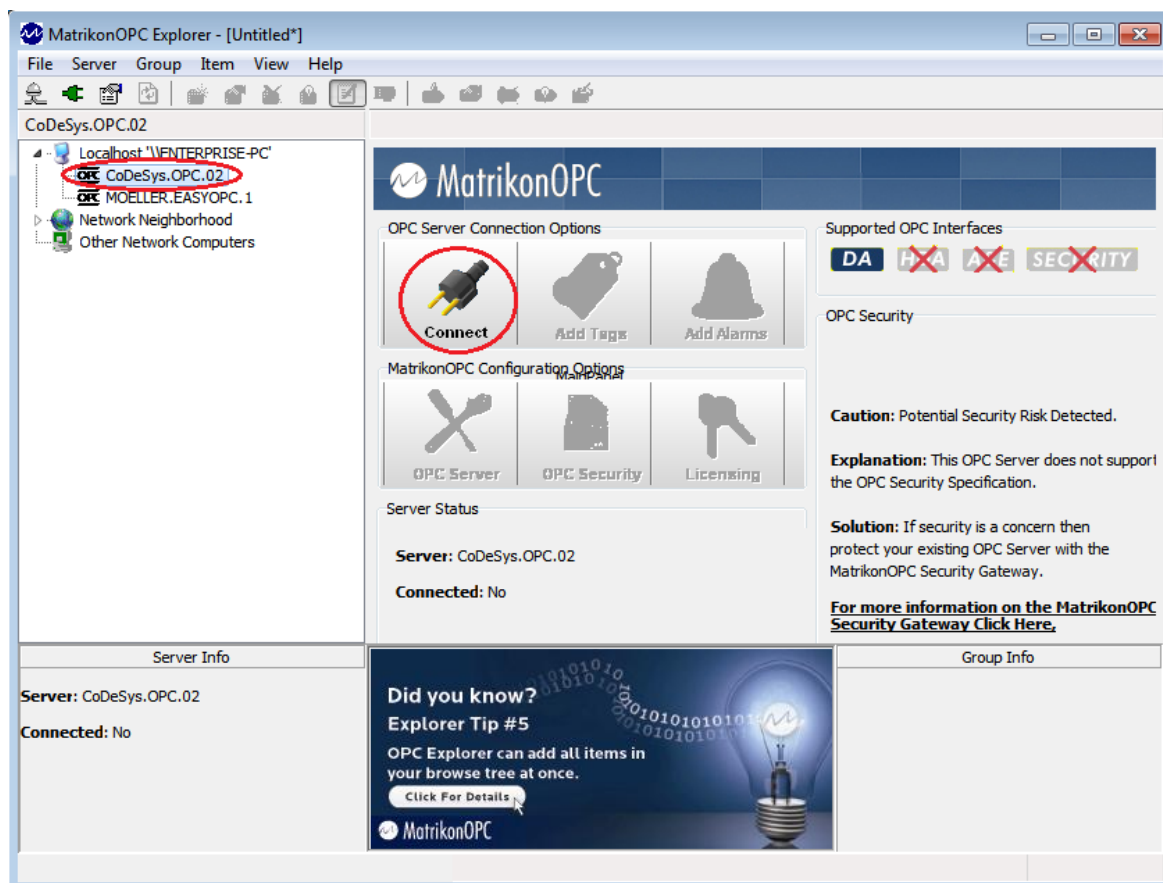


Рис. 92. Соединение с OPC сервером через MatrikonOPC Explorer

Добавьте нужные теги для отображения (рис. 93, 94). Далее можно просматривать состояние добавленных тегов в режиме реального времени (рис. 95).

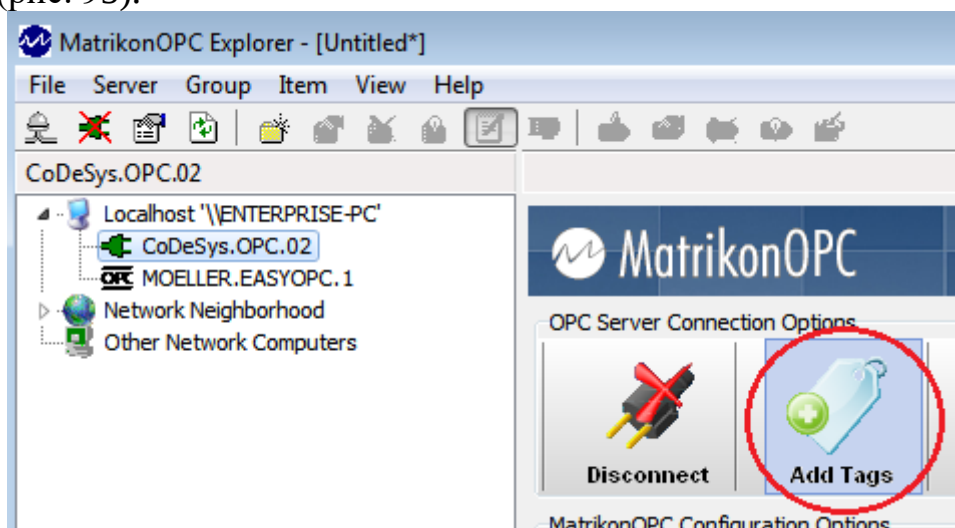


Рис. 93. Вход в раздел добавления тегов

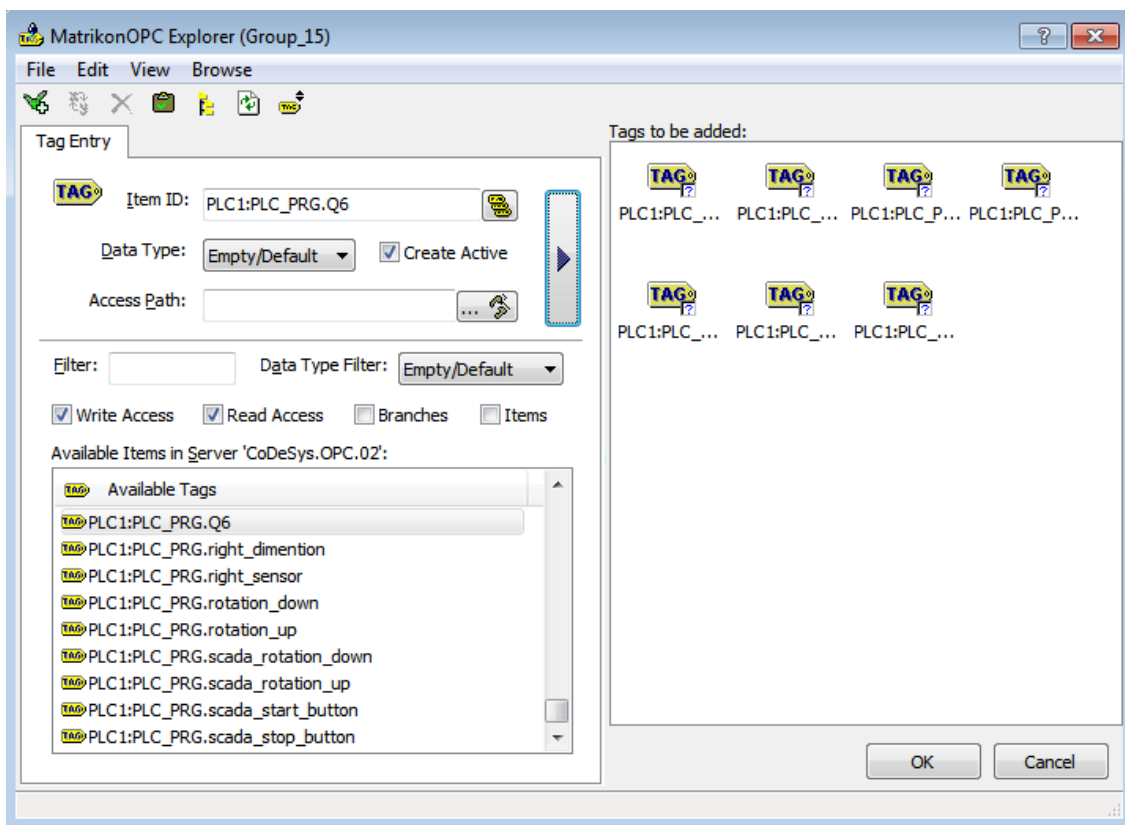


Рис. 94. Добавление тегов в обозреватель

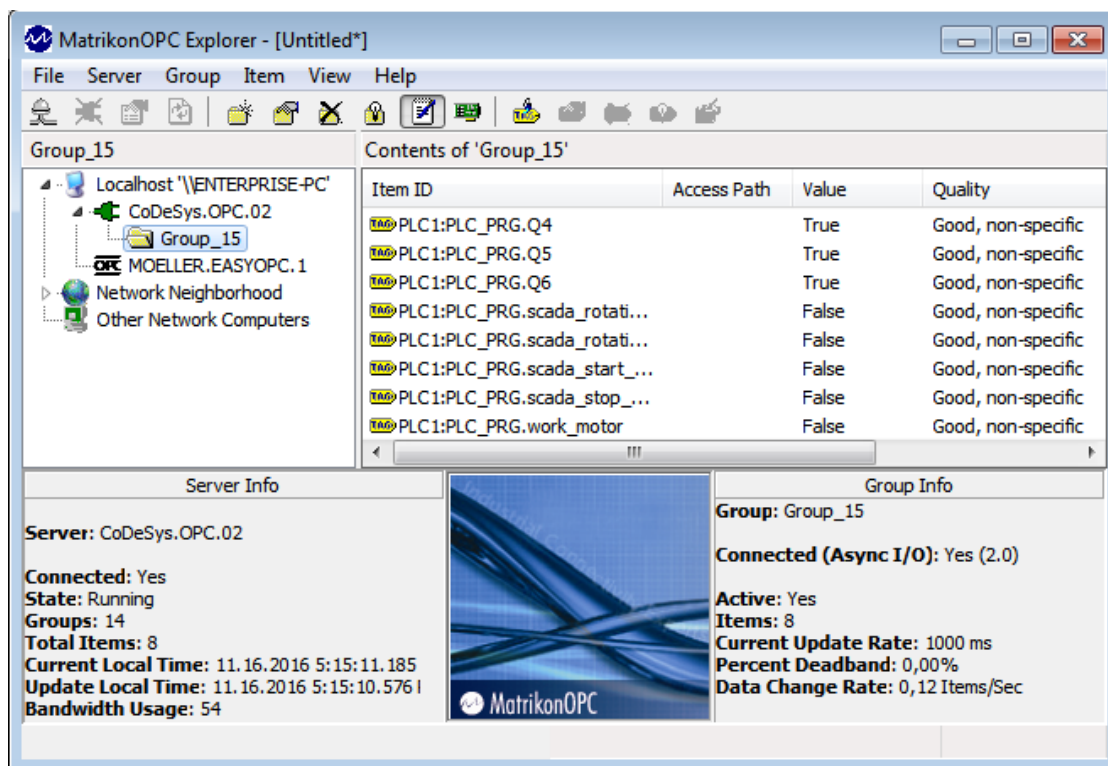


Рис. 95. Отображение состояния тегов в режиме реального времени

3. УСТАНОВКА ПОГРАММНОГО ОБЕСПЕЧЕНИЯ

Внимание! В аудитории учебного центра на стендах данную процедуру выполняет только преподаватель один раз. Обучающийся, при наличии данного программного обеспечения, может его установить только на своём личном компьютере.

3.1. Установка и настройка CODESYS 2.3.9 SP8

1. Запустите файл *Setup_XSOFT_CODESYS_V2.3.9_SP8.exe*.
2. Если автоматически не запустился установщик таргетов *Setup_XC_XV_Targets_V2.3.9_SP8.exe*, запустите его.
3. Активируйте таргеты для контроллера XC-CPU201 и HMI панели оператора XV102. Для этого откройте специальную CODESYS утилиту InstallTarget и следуйте следующим шагам (рис. 96-98).

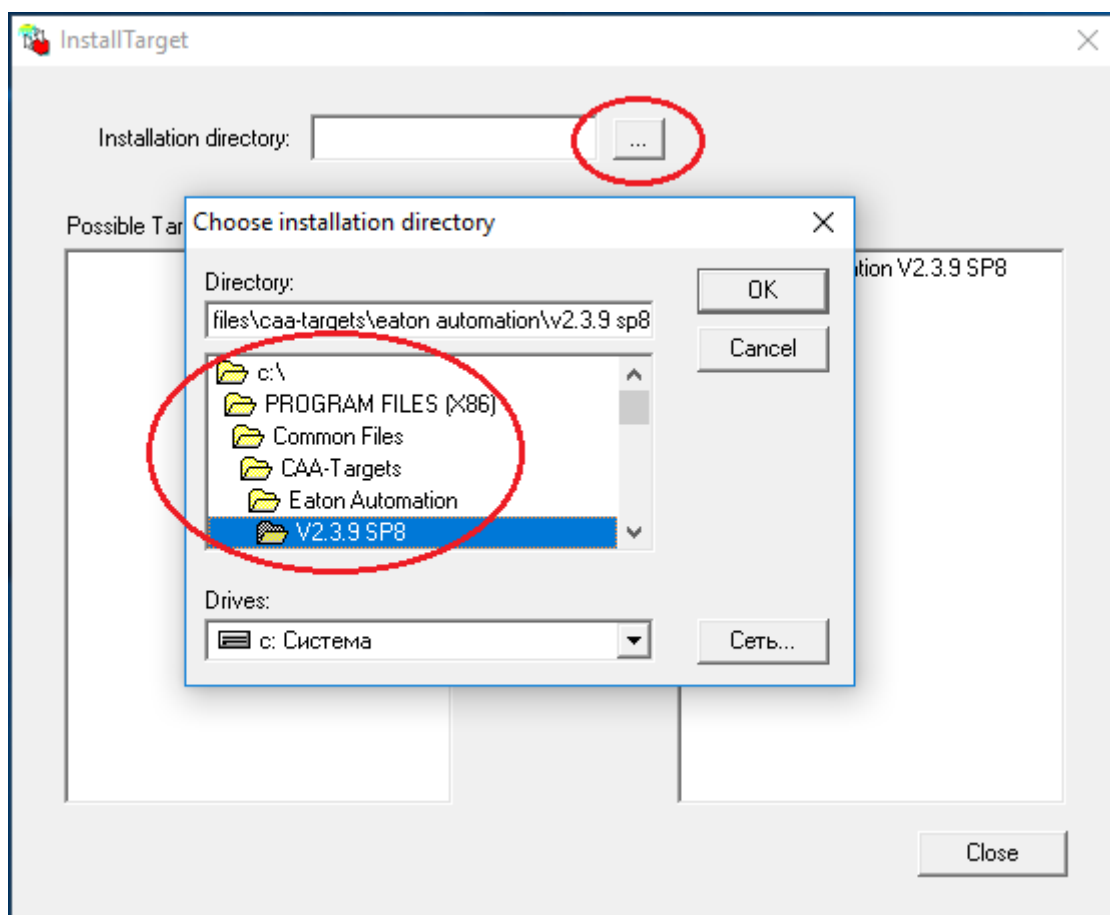


Рис. 96. Определение области видимости выбора таргет-файла

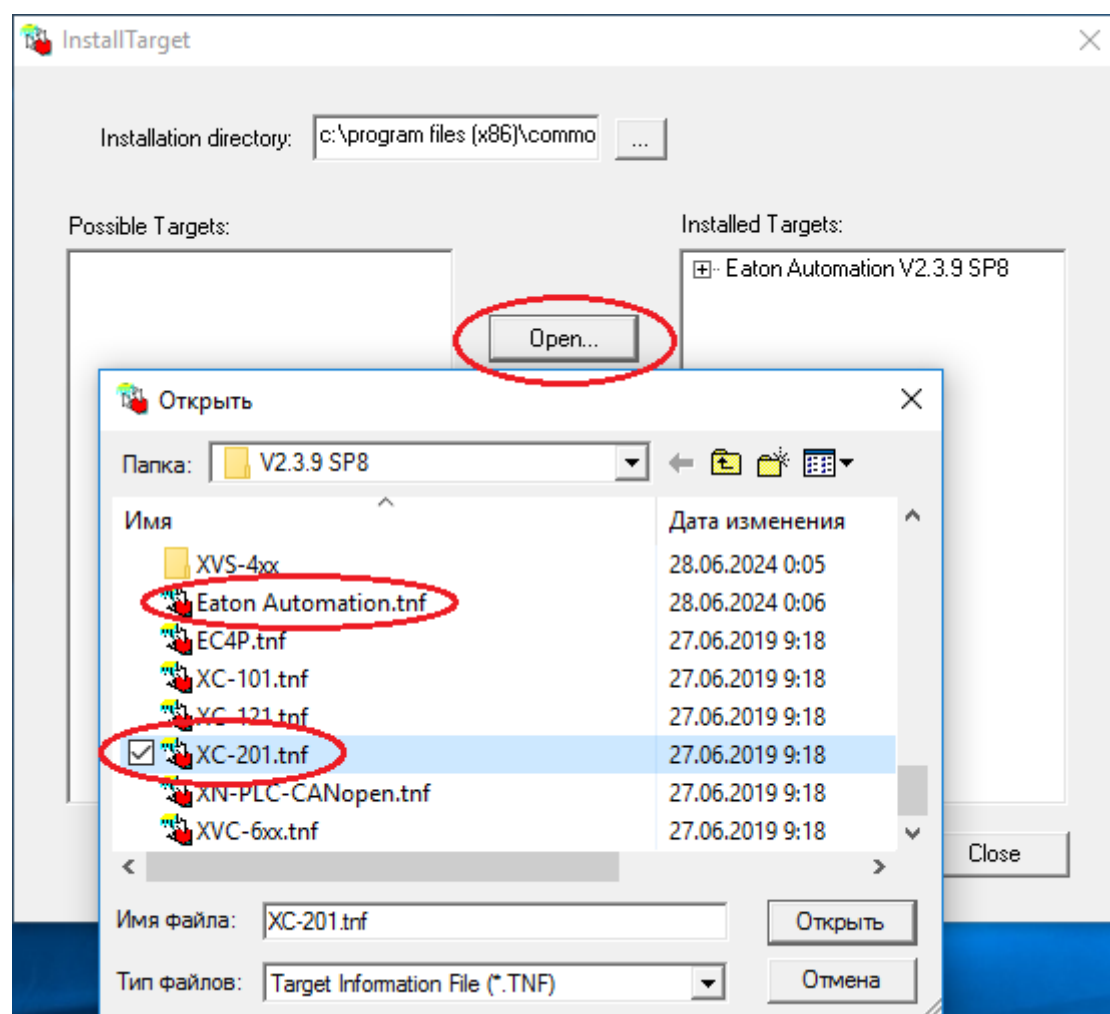


Рис. 97. Выбор TNF таргет-файла для установки таргетов

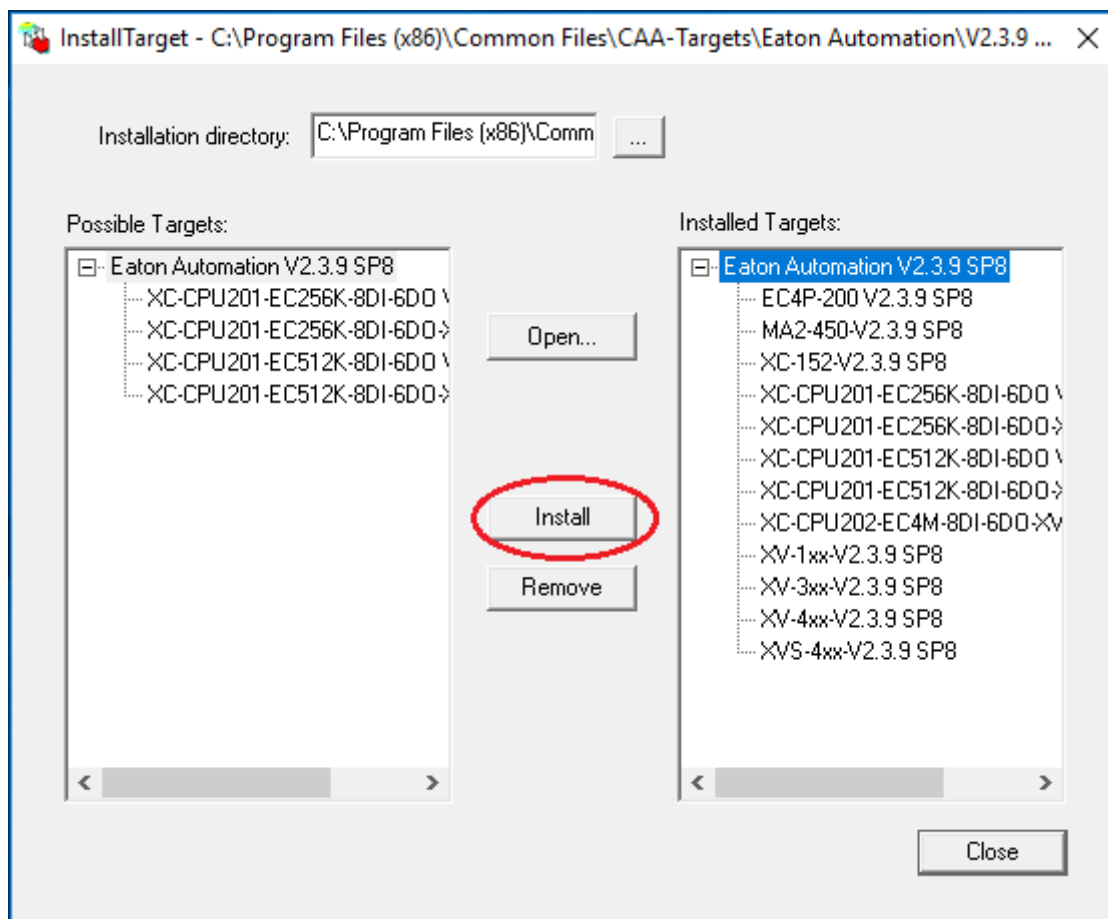


Рис. 98. Установка таргетов для контроллеров

3.2. Установка Trace Mode 6.09

1. Установка программного комплекса Trace Mode 9.09 достаточно проста. Запустите установщик «Setup» и следуйте его дальнейшим инструкциям.

3.3. Установка и лицензирование Galileo 8.1

1. *Получите лицензионный ключ для использования его при установке Galileo.* Для выполнения этой операции необходимо иметь лицензионный сертификат, который необходимо ввести на сайте <http://www.eaton-automation.com/license>, вместе с личными данными (рис. 99). В ответном письме на указанную электронную почту будет выслан лицензионный ключ.

2. *Установите Galileo 8.1.10.* Запустите файл GalileoV8110.exe и в ходе установки введите лицензионный ключ.

The screenshot shows a web browser window with the address bar displaying `www.automation4you.net/licmgr/license_en.htm`. The page features the Eaton logo with the tagline "Powering Business Worldwide" inside a decorative blue border. Below the logo, the section is titled "Licensing". A text prompt asks the user to "Please enter the certificate no. of your license document." Underneath, the label "Certificate" is followed by a form consisting of ten input boxes separated by hyphens. At the bottom right of the form area, there are two buttons labeled "clear" and "next".

← → C www.automation4you.net/licmgr/license_en.htm

EATON
Powering Business Worldwide

Licensing

Please enter the certificate no. of your license document.

Certificate

- - - - - - - - -

Рис. 99. Процесс получения лицензионного ключа

3.4. Обновление прошивки HMI панели XV102

1. Подайте электрическое питание на панель оператора. После загрузки Windows CE, нажмите кнопку *Start* и выберите в меню *Programs/Control Panel* (рис. 100).



Рис. 100. Открывание Control/Panel

2. Два раза кликните по вкладке *Network*.
3. В появившемся окне два раза кликните по вкладке *ONBOARD1* (рис. 101).

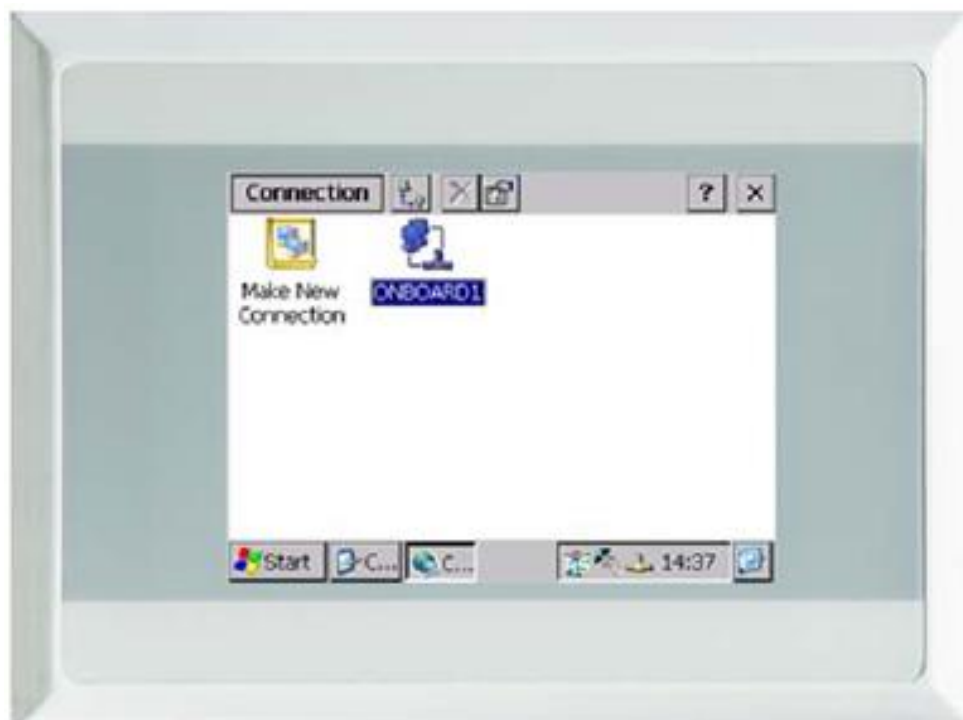


Рис. 101. Открывание Onboard1

4. При этом откроется окно *FEC Internet Driver*, в котором будут отображены сетевые настройки контроллера панели оператора (рис. 102).

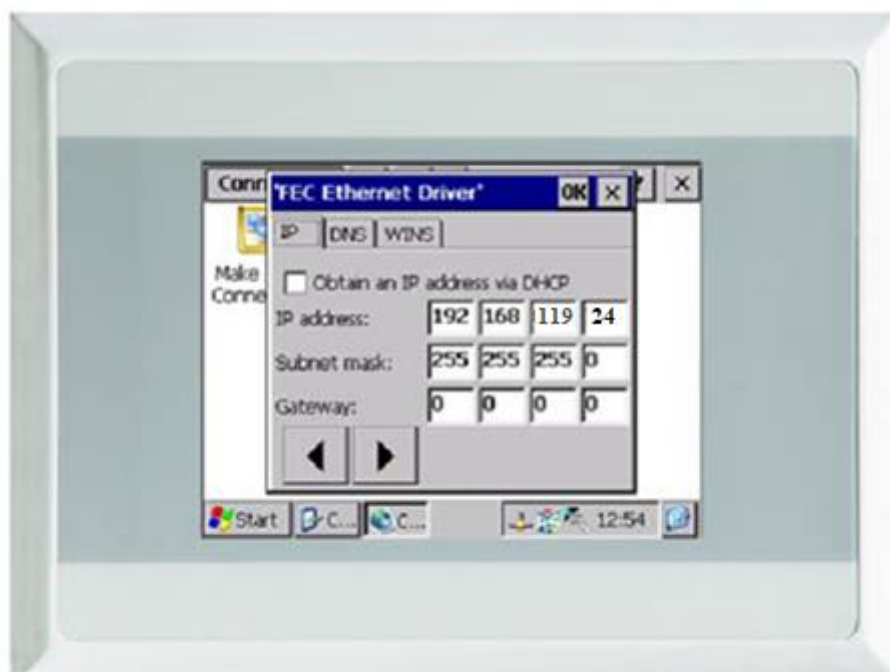


Рис. 102. Сетевые настройки контроллера панели оператора

5. Задайте IP адрес 192.168.119.24 и нажмите «ОК».
6. Откройте Codesys 2.3.9. При выборе целевой платформы, необходимо указать XV-1xx-V2.3.9 SP8.
7. Перейдите во вкладку ресурсов, зайдите в конфигурацию ПЛК, выберите Firmware и нажмите кнопку Start (рис. 103).

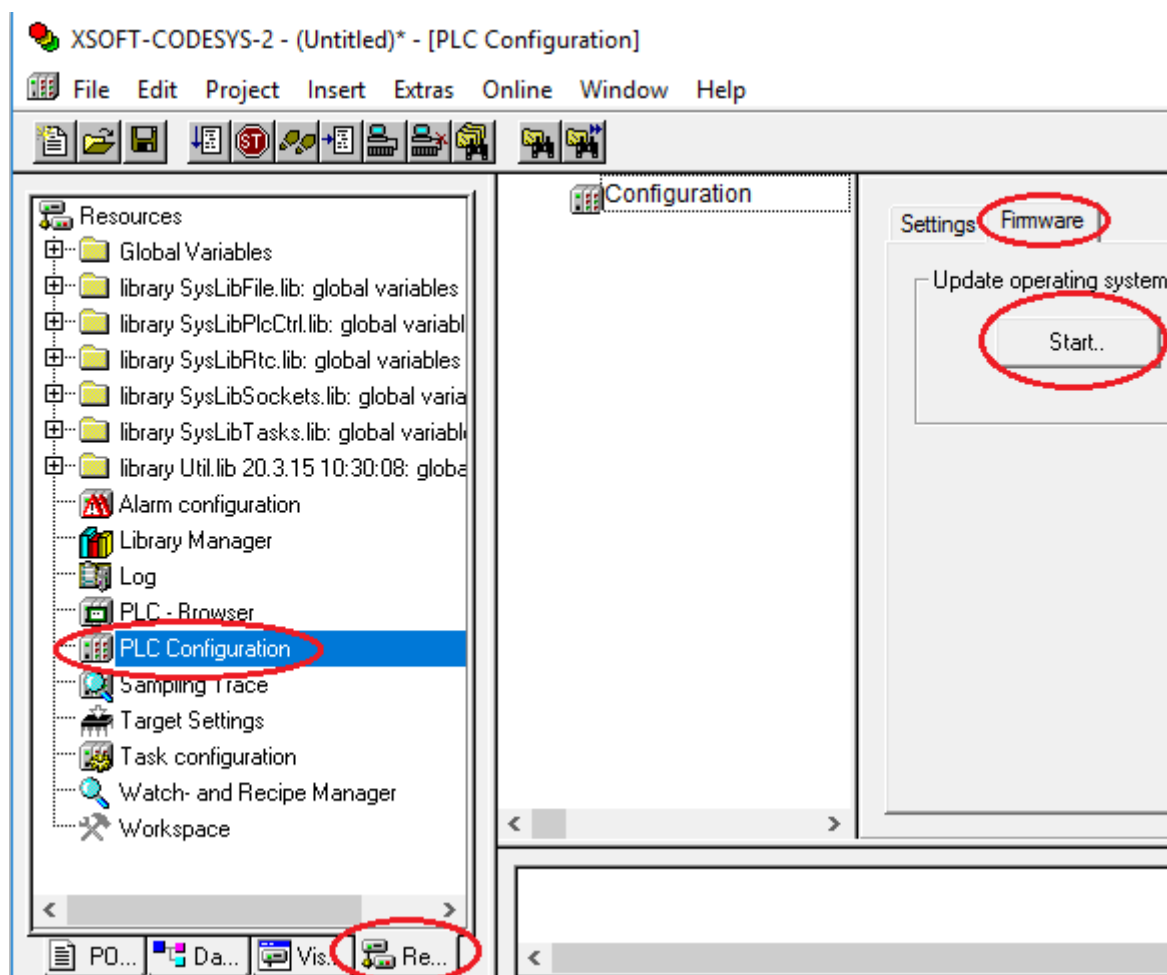


Рис. 103. Запуск прошивки HMI панели

8. В появившемся окне выберите файл TargetFirmwareWinCE_V2.4.21, расположенный по адресу Program Files\Common Files\CAA-Targets\Eaton Automation\V2.3.9 SP8\Firmware\XV-1xx.

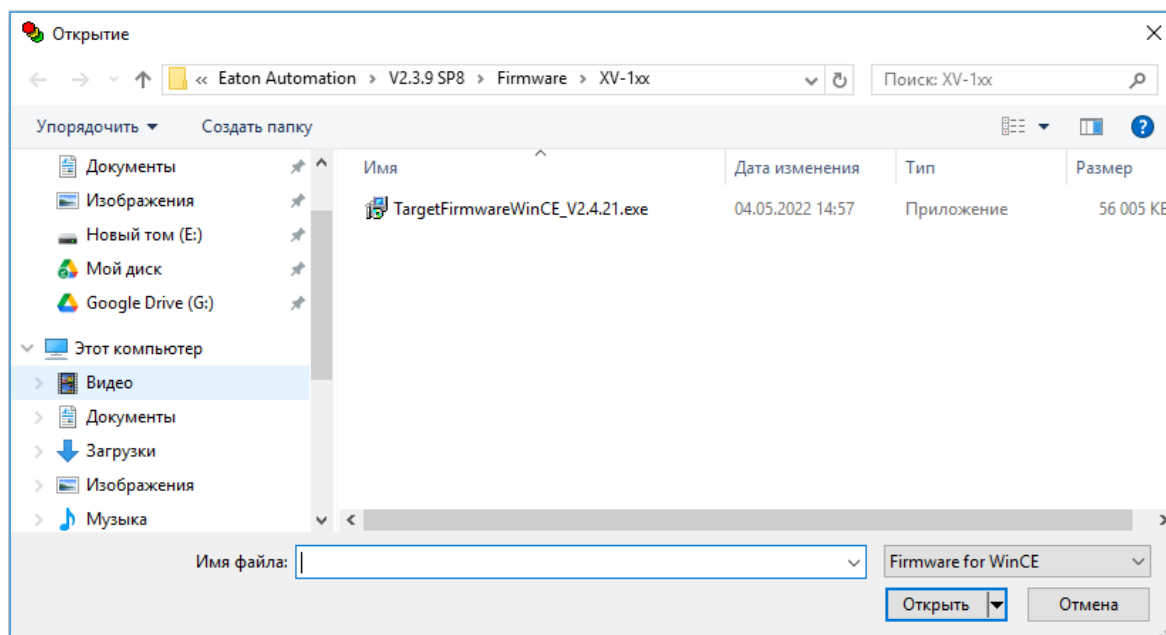


Рис. 104. Выбор TargetFirmwareWinCE_V2.4.21

- После этого запустится окно инсталлятора. В первом приветственном окне, мы нажимаем кнопку Next. В следующем выбираем режим FTP-Installation и нажимаем кнопку Next (рис. 105).

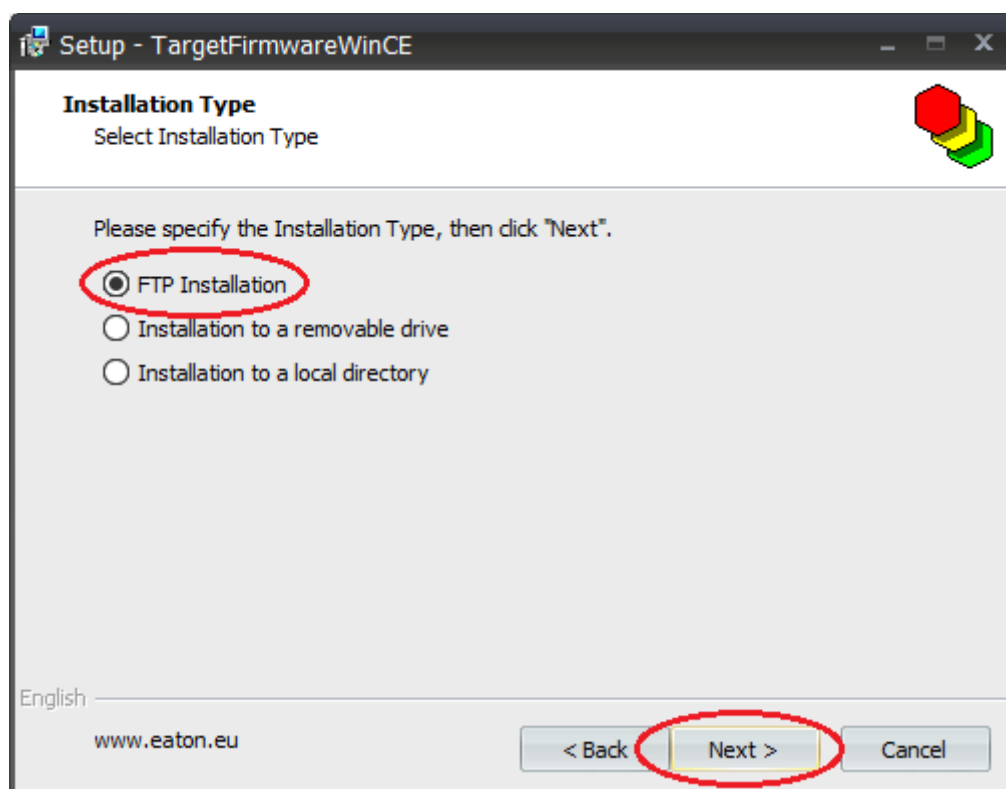
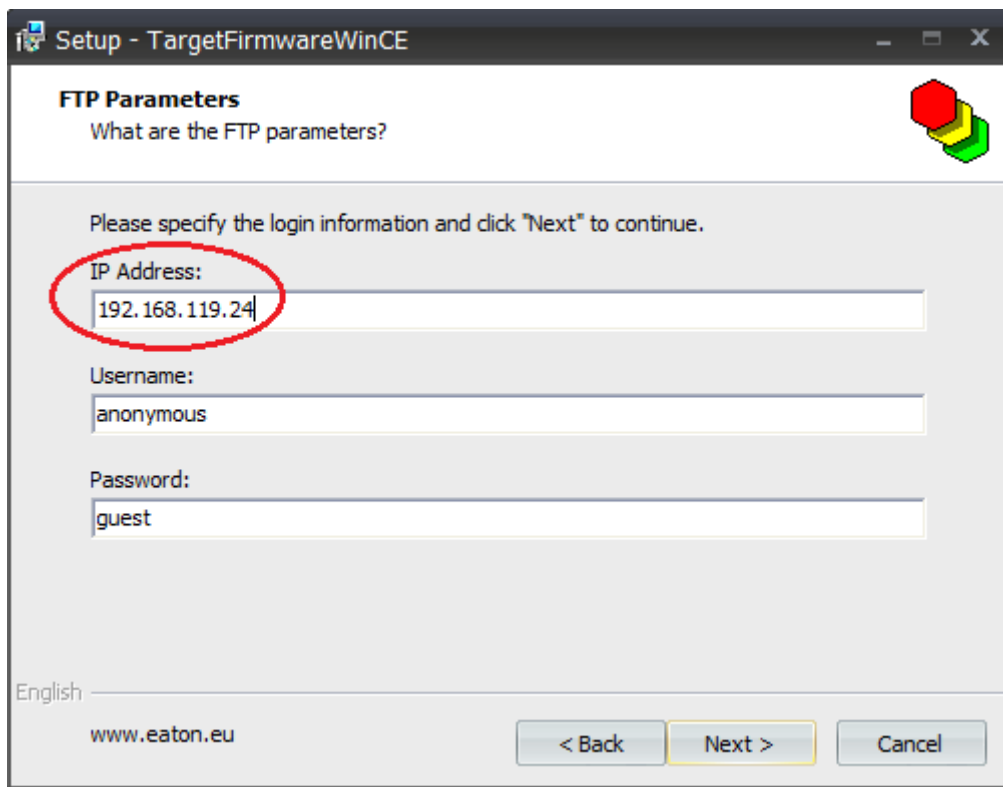


Рис. 105. FTP установка

10. В следующем окне введите IP-адрес панели оператора (192.168.119.24), остальные поля оставьте неизменным (рис. 106).



The screenshot shows a window titled "Setup - TargetFirmwareWinCE". Inside, there is a section titled "FTP Parameters" with the subtitle "What are the FTP parameters?". Below this, a message says "Please specify the login information and click 'Next' to continue." There are three input fields: "IP Address:" with the value "192.168.119.24" (circled in red), "Username:" with the value "anonymous", and "Password:" with the value "guest". At the bottom, there are three buttons: "< Back", "Next >" (highlighted with a yellow border), and "Cancel". The bottom left corner shows "English" and "www.eaton.eu".

Рис. 106. Задание IP адреса для FTP установки

11. Перед выполнением следующего шага необходимо запустить FTP сервер на панели оператора, Start/Programs/Communication/FTP server/OK.
12. Если IP-адрес был указан верно и в панели оператора запущен FTP-сервер, то инсталлятор автоматически определит тип панели (рис. 107). Далее нажмите кнопку Next.

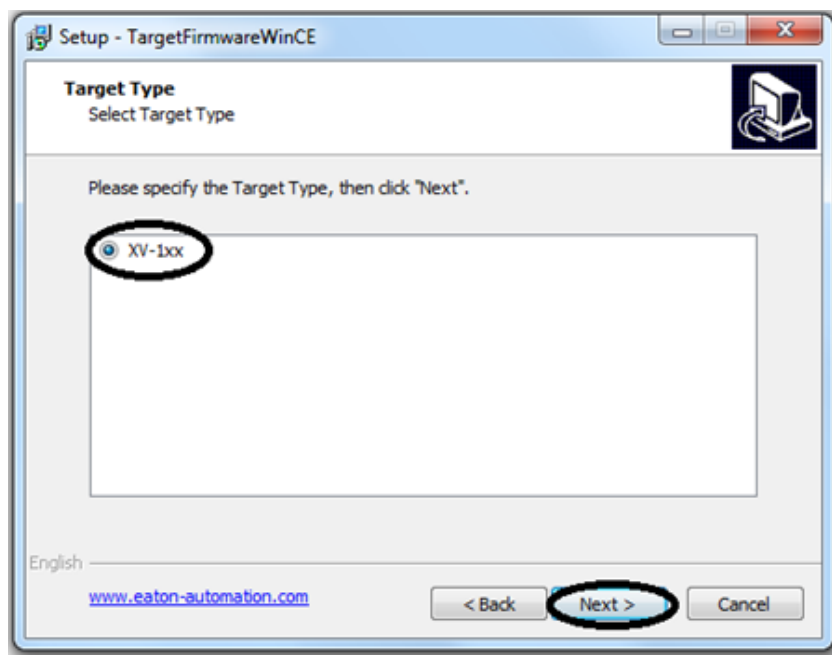


Рис. 107. Автоматическое определение типа НМІ панели

13. Далее происходит выбор устанавливаемых компонентов. Выберите все компоненты и нажмите кнопку *Next* (рис. 108).

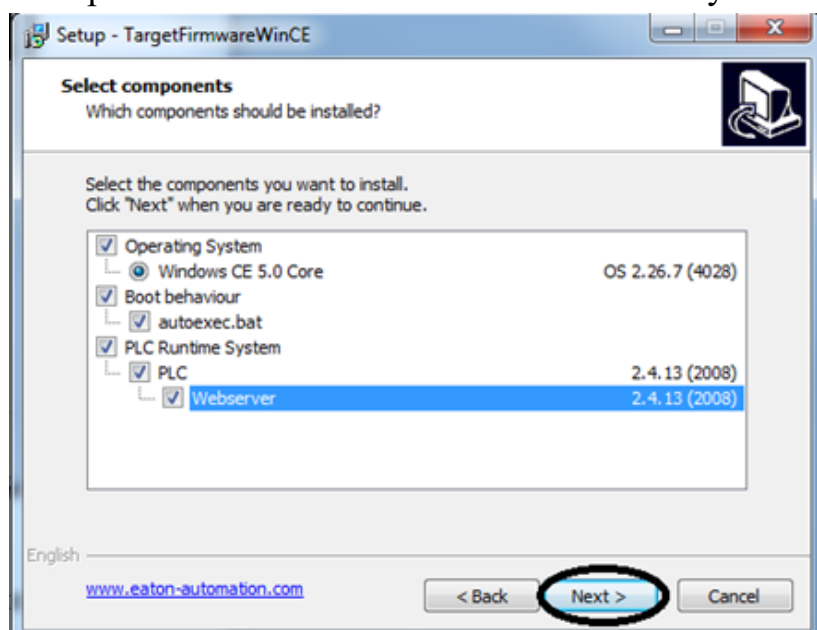


Рис. 108. Выбор устанавливаемых компонентов

14. Завершающим этапом подтвердите установку выбранных компонентов нажатием кнопки *Install*. Дождитесь завершения установки, панель оператора должна автоматически перезагрузиться.

СПИСОК ЛИТЕРАТУРЫ

1. Гирник А.С., Федянин А.Л., Шилин А.А. Системы автоматизации и управления на базе программируемых логических контроллеров: учебное пособие [Электронный ресурс] / А.С. Гирник. – Томск : Изд-во ТПУ, 2024. – Заглавие с титульного экрана. – URL: https://portal.tpu.ru/departments/otdel/publish/catalog/2024/method_2024/Ta b/GirnikFedyaninShilin.pdf.
2. Руководство пользователя по программированию ПЛК в CoDeSys 2.3. 3S – Smart Software Solutions GmbH 2008. – URL: https://owen.ru/uploads/373/cds23_manual_v2.8.pdf.
3. Первые шаги с CoDeSys. 3S – Smart Software Solutions GmbH 2004. – URL: https://owen.ru/uploads/373/cds23_firststeps.pdf.
4. Визуализация CoDeSys. Дополнение к руководству пользователя по программированию ПЛК CoDeSys 2.3. 3S – Smart Software Solutions GmbH 2008. – URL: https://owen.ru/uploads/373/cds23_visu_v1.7.pdf.