

Министерство образования и науки

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Серия «Учебно-методический комплекс дисциплины»



А.Н. Рыбалев

# Программируемые логические контроллеры и аппаратура управления: лабораторный практикум

## Часть 5.

### Панели оператора

Учебное пособие



Благовещенск  
2015

ББК 32.965 я 73  
Р 93

*Печатается по решению  
редакционно-издательского совета  
энергетического факультета  
Амурского государственного  
университета*

Рыбалев А.Н. Программируемые логические контроллеры и аппаратура управления: лабораторный практикум. Часть 5. Панели оператора. Учебное пособие. – Благовещенск: Амурский гос. ун-т, 2015.

Пособие предназначено для студентов направления 220700.62 «Автоматизация технологических процессов и производств», изучающих дисциплины «Средства автоматизации и управления», «Автоматическое управление на электрических станциях», «Автоматизация технологических процессов» и выполняющих лабораторные работы по данным дисциплинам. Может быть также использовано при выполнении курсовых и дипломных проектов.

*Рецензенты:*

Ю.В. Мясоедов, декан Энергетического факультета АмГУ, профессор, канд. техн. наук;

Е.П. Ялама, главный инженер управления пуско-наладочных работ ОАО «Гидроэлектромонтаж»

***В авторской редакции***

© Амурский государственный университет, 2015

© Рыбалев А.Н., 2015

## Содержание

<b>Введение и благодарности .....</b>	<b>5</b>
<b>Лабораторная работа №1. Система дистанционного управления лабораторным объектом на основе модулей ввода-вывода Овен МВА8, МВУ8 и панели оператора ИП320 (ИП320 в режиме Master).....</b>	<b>7</b>
Теоретические сведения .....	7
1. Модули ввода-вывода.....	7
2. Лабораторный стенд .....	14
3. Конфигурирование модулей ввода-вывода.....	21
4. Панель оператора ИП320 .....	27
Программа работы .....	40
Содержание отчета.....	42
Контрольные вопросы.....	43
<b>Лабораторная работа №2. Система дистанционного управления лабораторным объектом на основе модулей ввода-вывода Овен МВА8, МВУ8, панели оператора ИП320 и ПЛК150 (ИП320 в режиме Slave) .....</b>	<b>45</b>
Теоретические сведения .....	45
1. ПЛК150.....	45
2. Конфигурирование ПЛК150 для работы в качестве Master-устройства сети Modbus.....	49
Программа работы .....	55
Содержание отчета.....	58
Контрольные вопросы.....	58
<b>Лабораторная работа №3. Знакомство с панелью оператора WeintekMT8070iE. Off-line симуляция.....</b>	<b>59</b>
Теоретические сведения .....	59
1. Панель оператора Weintek MT8070iE.....	59
2. Конфигурирование панели в EasyBuilder Pro.....	59
Программа работы .....	62
Содержание отчета.....	65
Контрольные вопросы.....	65
<b>Лабораторная работа №4. Имитационное моделирование системы автоматического регулирования. On-line симуляция Weintek MT8070iE ..</b>	<b>66</b>
Программа работы .....	66
Содержание отчета.....	66
Контрольные вопросы.....	66
<b>Лабораторная работа №5. Система дистанционного управления лабораторным объектом на основе модулей ввода-вывода Овен МВА8, МВУ8, панели оператора Weintek MT8070iE и ПЛК150 .....</b>	<b>67</b>
Теоретические сведения .....	67

Программа работы .....	68
Содержание отчета.....	69
Контрольные вопросы.....	69
<b>ПРИЛОЖЕНИЕ 1. Краткое описание протокола Modbus .....</b>	<b>71</b>
Общие сведения .....	71
Структура протокола Modbus RTU .....	71
Функции Modbus .....	72
<b>ПРИЛОЖЕНИЕ 2. Имитационное моделирование системы управления с панелью оператора Weintek MT8070iE .....</b>	<b>80</b>
Назначение и состав системы.....	80
Межпрограммный обмен.....	80
Запуск ОРС сервера .....	81
Имитационная модель объекта управления.....	85
Разработка управляющей программы.....	88
Интерфейс пользователя.....	112
<b>Библиографический список.....</b>	<b>118</b>

## ВВЕДЕНИЕ

В пособии рассматриваются вопросы разработки человеко-машинного интерфейса систем автоматизации на основе операторских панелей.

Первые две лабораторные работы посвящены изучению и применению текстовой панели ИП 320 производства фирмы «Овен», остальные три – сенсорной панели МТ8070iE производства фирмы Weintek. Кроме операторских панелей, в работах задействованы лабораторная установка для нагрева и охлаждения воды и система управления на базе модулей ввода-вывода МВА8, МВУ8 и программируемого логического контроллера ПЛК150 производства фирмы «Овен», размещенных в лаборатории технических и программных средств автоматизации кафедры автоматизации производственных процессов и электротехники Амурского государственного университета.

В приложениях помещены материалы, имеющие самостоятельную ценность:

краткое описание протокола Modbus, используемого при передаче данных;

описание системы имитационного моделирования с визуализацией на программном симуляторе панели Weintek. Система может быть использована как прототип для разработки программных симуляторов АСУ ТП.

Автор выражает признательность:

выпускникам Амурского государственного университета специальности 220301 «Автоматизация технологических процессов и производств»:

*Денису Алексеевичу Теличенко* (ныне доцент кафедры АПП и Э), *Максиму Петровичу Баркалову* – разработчикам концепции лабораторного стенда (2003 г.);

*Владимиру Ивановичу Василевичу, Виктору Павловичу Отскочному* – проектировщикам и непосредственным создателям лабораторной установки (2006 г.);

*Денису Анатольевичу Пигур, Кириллу Андреевичу Пилипенко, Евгению Петровичу Чебанову* – разработчикам системы управления на основе приборов «Овен» (2008 г.);

*Алексею Игоревичу Овсянникову* (2009 г.), *Роману Александровичу Ахметшину, Алене Владимировне Белоусовой* (2012 г.), *Сергею Александровичу Ерескину* (2013 г.), проделавшим большой объем работы по освоению технических средств и разработке программного обеспечения;

*всем выпускникам группы 941*, подарившим лаборатории кафедры операторскую панель Weintek МТ8070iE (2014 г.);

*компании «Овен»*, одному из ведущих российских производителей оборудования для автоматизации, предоставившей комплекс приборов для лаборатории в 2007 году;

*Благовещенской ТЭЦ* и лично инженерам *Николаю Сергеевичу Безрукову, Денису Алексеевичу Теличенко*, выпускниками преподавателям кафедры, за многолетнюю помощь в оснащении лаборатории и предоставлении панели оператора ИП 320 (2013 г.);

*ООО «Интеграл Плюс» и лично Павлу Сергеевичу Русину, старшему группы КИП и А, выпускнику и аспиранту кафедры, за предоставленные программируемый логический контроллер и электротехнический шкаф (2014 г.);*

*Дмитрию Владимировичу Капитонову, студенту специальности 220301, выполнившему монтажные работы по установке панели Weintek MT8070iE и контроллера ПЛК 150;*

*Роману Дмитриевичу Редозубову, бывшему старшему преподавателю кафедры, соавтору идеи лабораторной установки, многолетнему руководителю и консультанту курсовых и дипломных проектов по ее разработке и модернизации;*

*Владимиру Пантелеевичу Харькову, высококвалифицированному рабочему кафедры, главному исполнителю всех монтажных работ.*

*Лабораторная работа №1. Система дистанционного управления лабораторным объектом на основе модулей ввода-вывода Овен МВА8, МВУ8 и панели оператора ИП320 (ИП320 в режиме Master)*

Цели работы:

знакомство с функциональными возможностями модулей ввода-вывода Овен МВА8, МВУ8 и получение навыков их конфигурирования;

знакомство с функциональными возможностями панели оператора ИП320 и получение навыков ее конфигурирования;

разработка и реализация системы дистанционного управления лабораторным объектом на базе модулей ввода-вывода Овен МВА8, МВУ8 и панели оператора ИП320, работающей в режиме Master.

***Теоретические сведения***

***1. Модули ввода-вывода***

***1.1 Описание и характеристики модуля МВА8***

Модуль ввода аналоговый измерительный МВА8 предназначен для построения автоматических систем контроля и регулирования производственных технологических процессов в различных областях промышленности, сельском и коммунальном и других отраслях народного хозяйства [2].

Прибор выполняет следующие основные функции:

измерение физических параметров объекта, контролируемых входными первичными преобразователями;

цифровую фильтрацию измеренных параметров от промышленных импульсных помех;

коррекцию измеренных параметров для устранения погрешностей первичных преобразователей;

формирование аварийного сигнала при обнаружении неисправности первичных преобразователей;

передачу компьютеру информации о значениях измеренных датчиками величин или значениях, полученных после преобразования этих величин;

изменение значений его программируемых параметров с помощью программы конфигурирования;

сохранение заданных программируемых параметров в энергонезависимой памяти при отключении напряжения питания МВА8;

снятие показаний датчиков положения (резистивного и токового типа) и контактных дискретных датчиков.

Основные характеристики модуля представлены в табл. 1,2.

*Таблица 1*

**Общие характеристики модуля МВА8**

Наименование	Значение
Диапазон переменного напряжения питания частотой 47...63 Гц	90...245 В
Потребляемая мощность	не более 6 ВА
Количество измеряемых каналов	8

Наименование	Значение
Время опроса одного канала	не более 0,4 с
Напряжение источника питания активных датчиков	24±3 В постоянного тока (180 мА макс.)
Интерфейс связи с компьютером	RS-485
Протокол связи, используемый для передачи информации о результатах измерения	Овен; Modbus-RTU; Modbus-ASCII; DCON
Степень защиты корпуса	IP20
Габаритные размеры прибора	157x86x57 мм
Масса прибора	не более 0,5 кг
Средний срок службы	8 лет

Таблица 2

### Датчики и входные сигналы

Наименование и НСХ	Диапазон измерения	Разрешающая способность	Предел основной приведенной погрешности
Термопреобразователи сопротивления по ГОСТ Р 6651-94			
ТСМ (Cu50)W <sub>100</sub> =1,4260	-50 <sup>0</sup> С...+200 <sup>0</sup> С	0,01 <sup>0</sup> С	0,25 %
ТСМ (50М)W <sub>100</sub> =1,4280	-190 <sup>0</sup> С...+200 <sup>0</sup> С		
ТСП (Pt50)W <sub>100</sub> =1,3850	-200 <sup>0</sup> С...+750 <sup>0</sup> С		
ТСП (50П)W <sub>100</sub> =1,3910	-200 <sup>0</sup> С...+750 <sup>0</sup> С		
ТСМ (Cu100)W <sub>100</sub> =1,4260	-50 <sup>0</sup> С...+200 <sup>0</sup> С		
ТСМ (100М)W <sub>100</sub> =1,4280	-190 <sup>0</sup> С...+200 <sup>0</sup> С		
ТСП (Pt100)W <sub>100</sub> =1,3850	-200 <sup>0</sup> С...+750 <sup>0</sup> С		
ТСП (100П)W <sub>100</sub> =1,3910	-200 <sup>0</sup> С...+750 <sup>0</sup> С		
ТСН (Ni100)W <sub>100</sub> =1,6170	-60 <sup>0</sup> С...+180 <sup>0</sup> С		
ТСМ (Cu500)W <sub>100</sub> =1,4260	-50 <sup>0</sup> С...+200 <sup>0</sup> С		
ТСМ (500М)W <sub>100</sub> =1,4280	-190 <sup>0</sup> С...+200 <sup>0</sup> С		
ТСП(Pt500)W <sub>100</sub> =1,3850	-200 <sup>0</sup> С...+650 <sup>0</sup> С		
ТСП (500П)W <sub>100</sub> =1,3910	200 <sup>0</sup> С...+650 <sup>0</sup> С		
ТСН (Ni500)W <sub>100</sub> =1,6170	-60 <sup>0</sup> С...+180 <sup>0</sup> С		
ТСМ (Cu100)W <sub>100</sub> =1,4260	-50 <sup>0</sup> С...+200 <sup>0</sup> С		
ТСМ(1000М)W <sub>100</sub> =1,4280	-190 <sup>0</sup> С...+200 <sup>0</sup> С		
ТСМ (Pt1000)W <sub>100</sub> =1,3850	-200 <sup>0</sup> С...+650 <sup>0</sup> С		
ТСМ(1000П)W <sub>100</sub> =1,3910	-200 <sup>0</sup> С...+650 <sup>0</sup> С		
ТСМ(Ni1000)W <sub>100</sub> =1,3170	-60 <sup>0</sup> С...+180 <sup>0</sup> С		
по ГОСТ 6651-78			
ТСМ (53М)W <sub>100</sub> =1,4260 (гр.23)	-50 <sup>0</sup> С...+200 <sup>0</sup> С	0,01 <sup>0</sup> С	0,25 %
Термопары по ГОСТ Р 8.585-2001			
ТХК(L)	-200 <sup>0</sup> С...+800 <sup>0</sup> С	0,01 <sup>0</sup> С	0,5 %
ТХА(J)	-200 <sup>0</sup> С...+1200 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТНН(N)	-200 <sup>0</sup> С...+1300 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТХА(K)	-200 <sup>0</sup> С...+1300 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТПП(S)	0 <sup>0</sup> С...+1600 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТПП(R)	0 <sup>0</sup> С...+1600 <sup>0</sup> С	0,01 <sup>0</sup> С	



Наименование и НСХ	Диапазон измерения	Разрешающая способность	Предел основной приведенной погрешности
ТПР(В)	+200 <sup>0</sup> С.+1800 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТВР(А-1)	0 <sup>0</sup> С..+2500 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТВР(А-2)	0 <sup>0</sup> С..+1800 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТВР(А-3)	0 <sup>0</sup> С..+1600 <sup>0</sup> С	0,01 <sup>0</sup> С	
ТМК(Т)	-200 <sup>0</sup> С..+400 <sup>0</sup> С	0,01 <sup>0</sup> С	
Сигналы постоянного тока и напряжения по ГОСТ 26.011			
0...5 мА	0...100%	0,01%	0,25 %
0...20 мА	0...100%	0,01%	
4...20 мА	0...100%	0,01%	0,25 %
-50...+50 мВ	0...100%	0,01%	
0...1 В	0...100%	0,01%	
Датчик положения задвижек			
резистивный(до 900Ом)	0...100%	1%	не устанавливается
резистивный (0...2КОм)	0...100%	1%	
токовый 0(4)...20 мА	0...100%	1%	
токовый 0...5 мА	0...100%	1%	

Первичные преобразователи и датчики предназначены для контроля физических параметров объекта (температуры, положение заслонки) и преобразования их в электрические сигналы, оптимальные с точки зрения дальнейшей их обработки.

В качестве входных измерительных преобразователей прибора были использованы:

термопара (преобразователь термоэлектрический);

активные преобразователи с выходным аналоговым сигналом в виде постоянного напряжения и тока;

измерительные преобразователи положения исполнительных механизмов;

сухие контакты магнитных пускателей и выключателей.

Схема подключения термопары показана на рис. 1.

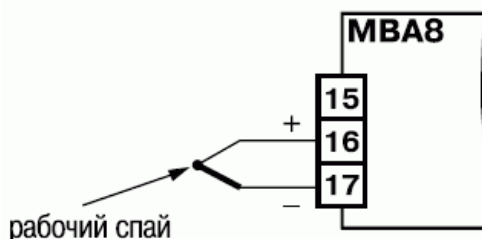


Рис. 1. Схема подключения ТП.

МВА8 способен обрабатывать сигналы датчиков резистивного типа с сопротивлением до 900 Ом или 2,0 кОм.

На лабораторной установке датчик положения выдает сигнал по положению воздушной заслонки. Кроме того, к датчикам резистивного типа относятся

переменные сопротивления на лицевой панели стенда, задающие сигналы ручного управления.

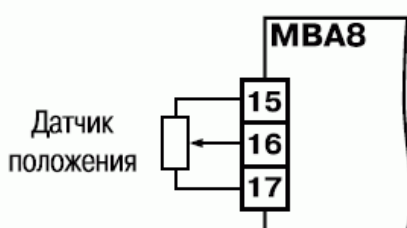


Рис. 2. Схема подключения ИП положения.

К МВА8 можно подключать до 16 дискретных датчиков, называемых «Сухие контакты». В качестве датчиков могут выступать различные выключатели, кнопки, контактные группы реле и т.д. Каждый аналоговый вход может быть использован для подключения двух дискретных датчиков.

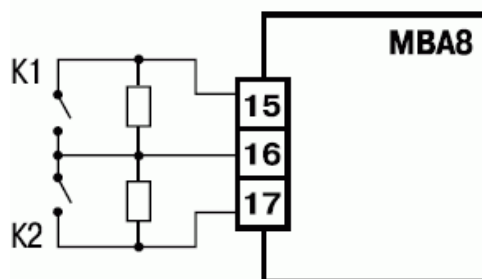


Рис. 3. Схема подключения сухих контактов.

В качестве шунтирующих сопротивлений можно использовать любые резисторы с одинаковым номиналом в 60-90 Ом. При опросе датчика «Сухие контакты» его состояние описывается целым числом от 1 до 4.

Таблица 3

### Расшифровка состояний датчика

Значение датчика	Состояние контакта 1	Состояние контакта 2
1	Разомкнут	Разомкнут
2	Замкнут	Разомкнут
3	Разомкнут	Замкнут
4	Замкнут	Замкнут

Прибор может быть использован одновременно для работы с различными типами датчиков: термопреобразователями сопротивления, термопарами и т.п. При этом несущественно, к какому из входов МВА8 будет подключен датчик того или иного типа, так как все восемь входов прибора абсолютно идентичны. После подключения датчикам присваиваются порядковые номера тех входов прибора, с которыми они соединены (входу 1 соответствует датчик № 1, входу 2 – датчик № 2 и т.д.).

Тип каждого датчика устанавливается пользователем в виде цифрового кода в программируемом параметре in-t при подготовке прибора к работе.

## 1.2. Описание и характеристики модуля МВУ8

Прибор МВУ8 предназначен для преобразования цифровых сигналов, передаваемых по сети RS 485, в аналоговые или дискретные сигналы управления исполнительными механизмами [3].

МВУ8 может осуществлять управление ВЭ в двух режимах:

непосредственное управление по сети RS-485;

интеллектуальное управление ИМ.

МВУ8 может управлять ИМ (от 1 до 8) следующих типов:

2-х позиционными (ТЭНами, холодильниками, вентиляторами и т.п.);

3-х позиционными (задвижками постоянной скорости с датчиком или без датчика положения, шиберами, поворотными заслонками, регулирующими клапанами и т.п.);

ИМ с аналоговым управлением;

устройствами сигнализации или защиты оборудования.

МВУ8 может использоваться совместно с модулем расширения MP1, содержащим восемь дискретных ВЭ, что позволяет увеличить количество ВЭ и ИМ в системе. МВУ8 работает в сети RS-485 по стандартным протоколам ОВЕН, Modbus-RTU, Modbus-ASCII, DCON. МВУ8 не является мастером сети, поэтому сеть RS-485 должна содержать мастер сети (например, SCADA-систему, OPC-драйвер, контроллер или регулятор). В качестве мастера сети могут использоваться приборы ОВЕН ТРМ151, ТРМ133, контроллеры ОВЕН ПЛК и т.п.

Конфигурирование МВУ8 осуществляется с помощью программы «Конфигуратор МВУ8», входящей в комплект поставки. Персональный компьютер подключается к сети через адаптер интерфейса RS-485/RS-232 (например, ОВЕН АС3).

Основные характеристики модуля представлены в табл. 4,5.

Таблица 4

### Общие характеристики модуля МВУ

Наименование	Значение
Напряжение питания переменного тока	90...264 В Частотой 47...63 Гц
Потребляемая мощность	не более 12 ВА
Количество выходных элементов	8
Интерфейс связи с компьютером	RS-485
Протокол связи, используемый для передачи информации	ОВЕН; Modbus-RTU; Modbus-ASCII; DCON
Степень защиты корпуса	IP20
Габаритные размеры прибора	157x86x57 мм
Масса прибора	не более 0,5 кг
Средний срок службы	8 лет

## Рабочие электрические параметры выходных элементов

Обозначение / Наименование ВЭ		Значение рабочих электрических параметров
Р	Реле электромагнитное	4А при напряжении не более 220 В частотой 50 Гц ( $\cos\varphi > 0,4$ )
К	Оптопара транзисторная n-p-n - типа	400мА при напряжении не более 60 В постоянного тока
С	Оптопарасимисторная	50мА при напряжении 250 В (в импульсном режиме частотой 50 Гц с длительностью импульса до 1А не более 5мс)
И	ЦАП «параметр-ток» 4...20 мА	Нагрузка 0...900Ом Напряжение питания 15...32 В
У	ЦАП «параметр-напряжение» 0...10 В»	Нагрузка не менее 2кОМ, Напряжение питания 15...32 В
Т	Выходные элементы для управления твердотельным реле	Выходное напряжение 4...6 В, Максимальный ток 50 мА

Схемы подключения к выходным элементам ВЭ прибора могут быть двух типов: аналогового (маркировка И, У) и дискретного (маркировка Р, К, С, Т), модуль расширения МР1 может быть оснащен ВЭ только дискретного типа.

*ВЭ дискретного типа.*

К таким ВЭ относятся реле (маркировка Р), симисторные оптопары (С), транзисторные ключи (К), выходы для управления твердотельным реле (Т).

*Электромагнитное реле* позволяет подключать нагрузку с максимально допустимым током 4 А при напряжении 220 В. На клеммы прибора выведены сухие контакты реле.

*Транзисторный ключ* применяется, как правило, для управления низковольтным реле (до 60 В). Схема включения приведена на рис. 4. Во избежание выхода из строя транзистора из-за большого тока самоиндукции параллельно обмотке реле Р1 необходимо устанавливать диод VD1, рассчитанный на напряжение 100 В и ток 1 А.

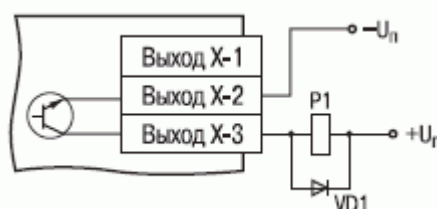


Рис. 4. Схема включения транзисторного выхода.

*Оптосимистор* включается в цепь управления мощного симистора через ограничивающий резистор R1 по схеме, показанной на рис. 5. Значение сопротивления резистора определяет величину тока управления симистора.

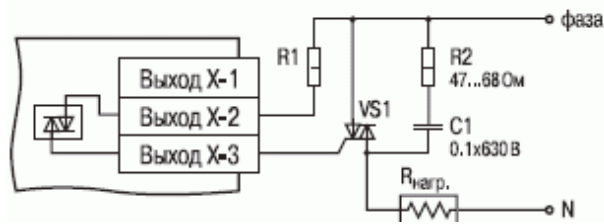


Рис. 5. Схема включения оптосимисторного выхода.

Оптосимистор может также управлять парой встречно-параллельно включенных тиристоров VS1 и VS2 (рис. 6). Для предотвращения пробоя тиристоров из-за высоковольтных скачков напряжения в сети к их выводам рекомендуется подключать фильтрующую RC-цепочку (R2-C1).

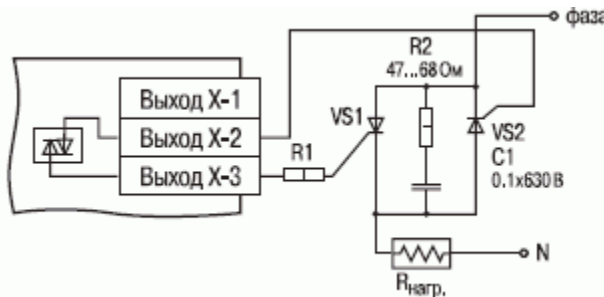


Рис. 6. Схема включения оптосимисторного выхода.

Оптосимистор имеет встроенное устройство перехода через ноль и поэтому обеспечивает полное открытие подключаемых тиристоров без применения дополнительных устройств.

Транзисторный ключ и оптосимистор имеют гальваническую развязку от схемы прибора.

*ВЭ аналогового типа*

*ЦАП 4...20 мА (маркировка И)*



Рис. 7. Схема подключения к токовому выходу.

Для работы ЦАП 4...20 мА используется внешний источник питания постоянного тока, номинальное значение напряжения  $U_n$  которого рассчитывается следующим образом:

$$U_{n \min} < U_n < U_{n \max};$$

$$U_{n \min} = 10 \text{ В} + 0,02 \text{ А} \cdot R_n \text{ Ом};$$

$$U_{n \max} = 17,5 \text{ В},$$

где  $U_{n \min}$ ,  $U_{n \max}$  – соответственно минимально и максимально допустимое напряжение источника питания, В;

$R_n$  – сопротивление нагрузки ЦАП, Ом.

Если напряжение источника питания ЦАП, находящегося в распоряжении пользователя, превышает расчетное значение  $U_{n,max}$ , то последовательно с нагрузкой необходимо включить ограничительный резистор.

*ЦАП 0...10 В (маркировка У)*



*Рис.8. Схема подключения к выходу по напряжению.*

Для работы ЦАП 0...10 В используется внешний источник питания постоянного тока, номинальное значение напряжения которого лежит в диапазоне от 15 до 32 В.

Выходной элемент рассчитан на резистивную нагрузку не менее 2 кОм. Любой аналоговый ВЭ необходимо юстировать.

## **2. Лабораторный стенд**

### **2.1. Структура стенда**

Лабораторный стенд представляет собой шкаф управления с размещенным в нем оборудованием (рис. 9).



*Рис. 9. Шкаф управления.*

На рис. 9 обозначены:

1 – модуль ввода МВА8;

2 – модуль вывода МВУ8;

3 – блок переключателей и сопротивлений. Переключатели позволяют отключать выходы модулей от стандартной конфигурации и использовать каналы ввода-вывода произвольным образом. В стандартной конфигурации за каналами закреплены определенные функции, часть из которых требует подключения к входам модуля МВА8 сопротивлений по специальной схеме;

4 – клеммный соединитель. К нему подключаются каналы модулей МВА8 и МВУ8, выведенные из стандартной конфигурации с помощью переключателей 3;

5 – блок управления симисторами и тиристорами БУСТ[1];

6 – пара встречновключенных силовых тиристоров;

7 – пускатель бесконтактный реверсивный ПБР 2М;

8 – преобразователь частоты HitachiSJ100[8];

9 – автоматические выключатели питания;

10 – магнитные пускатели;

11 – органы управления лицевой панели;

12 – миллиамперметр, показывающий сигнал управления БУСТ.

В стандартной конфигурации каналы модулей ввода-вывода используются для управления процессом нагрева и охлаждения воды в лабораторной установке. Управление осуществляется посредством двух магнитных пускателей, коммутирующих нерегулируемые потребители энергии на установке, тиристорного регулятора напряжения для регулирования мощности нагревательного элемента, пускателя бесконтактного реверсивного для управления приводом воздушной заслонки на линии охлаждения, преобразователем частоты для управления приводом вентилятора охлаждения.

Все каналы воздействия на объект допускают как ручное, так и автоматическое (посредством модулей МВА8 и МВУ8) управление. На дверце шкафа располагаются органы переключения режимов и управления в ручном режиме, а также мнемосхема, наглядно демонстрирующая возможности и текущее состояние цепей управления (рис. 10).

Стандартное назначение каналов модуля МВА8 следующее:

канал 1: разделен на две «половинки» (дискретные сигналы):

1.1 – сигнал о срабатывании магнитного пускателя 1;

1.2 – сигнал о срабатывании магнитного пускателя 2;

канал 2: сигнал обратной связи по положению исполнительного механизма МЭП (резистивный датчик);

канал 3: разделен на две «половинки» (дискретные сигналы):

3.1 – сигнал о срабатывании концевого выключателя МЭП в направлении «меньше»;

3.2 – сигнал о срабатывании концевого выключателя МЭП в направлении «больше»;

канал 4: разделен на две «половинки» (дискретные сигналы):

4.1 – сигнал о переводе ПБР в режим ручного управления;  
4.2 – сигнал о переводе ПЧ в режим ручного управления;  
канал 5: сигнал управления ПЧ в ручном режиме (резистивный датчик);  
канал 6: сигнал управления БУСТ в ручном режиме (резистивный датчик);  
канал 7: разделен на две «половинки» (дискретные сигналы):  
7.1 – сигнал о переводе БУСТ в режим ручного управления;  
7.2 – сигнал о переводе тиристоров режим прямого управления модулем МВУ8;  
канал 8: сигнал по температуре (формируемый термопарой ХК).  
Стандартное назначение каналов модуля МВУ8 следующее:  
канал 1 (реле) – управление магнитным пускателем 1;  
канал 2 (реле) – управление магнитным пускателем 2;  
канал 3 (транзисторный ключ) – управление ПБР, включение привода в направлении «меньше»;  
канал 4 (транзисторный ключ) – управление ПБР, включение привода в направлении «больше»;  
канал 5 (симистор) – не используется;  
канал 6 (симистор) – прямое управление тиристорами;  
канал 7 (токовый выход 4...20 мА) – управление ПЧ (задание частоты);  
канал 8 (выход по напряжению 0..10В) – управление БУСТ.



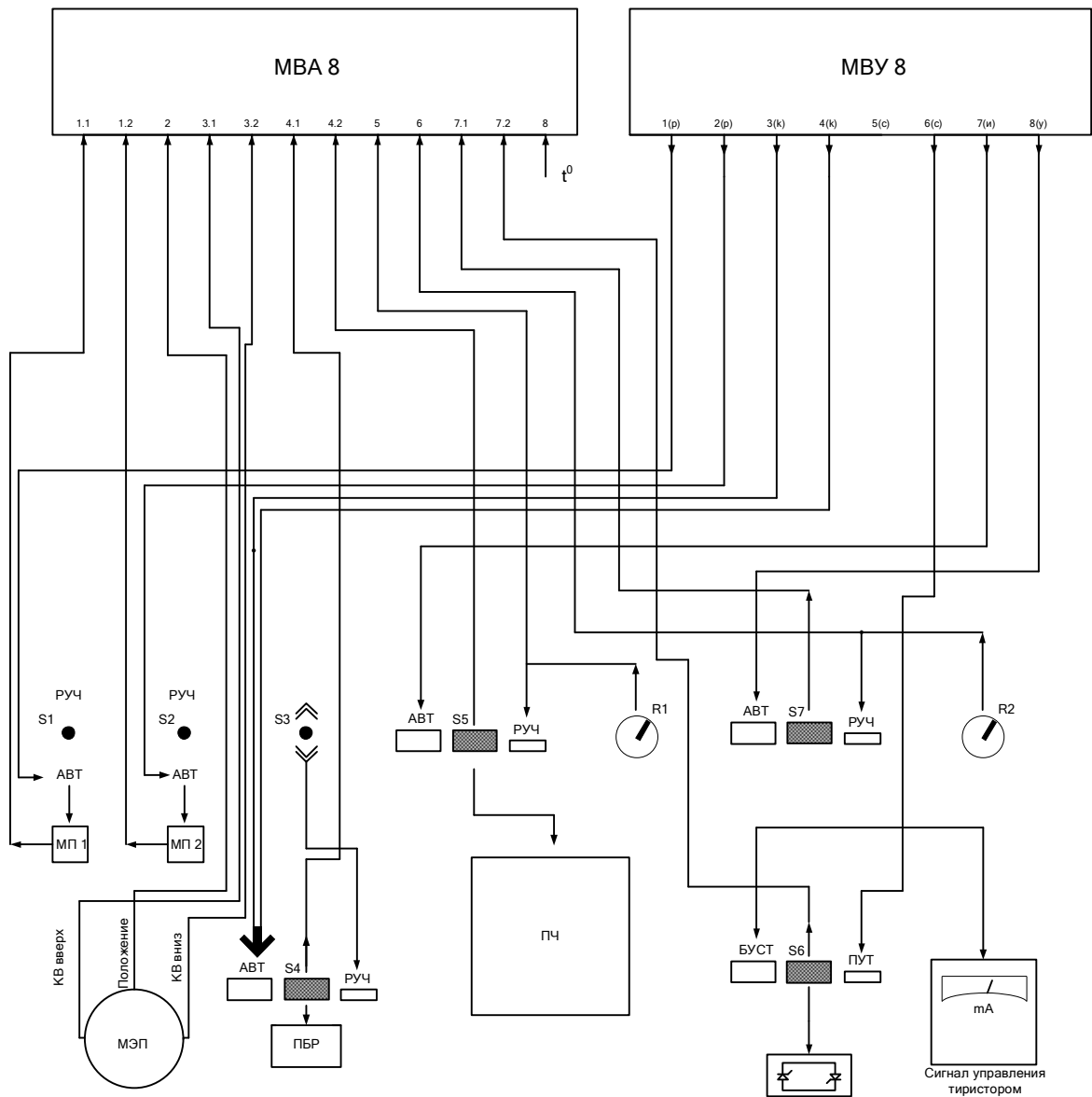


Рис.10. Внешний вид шкафа управления (мнемосхема).

## 2.2. Схемы соединений цепей управления

### Подсистема управления пускателями

Принципиальная схема соединений подсистемы показана на рис. 11.

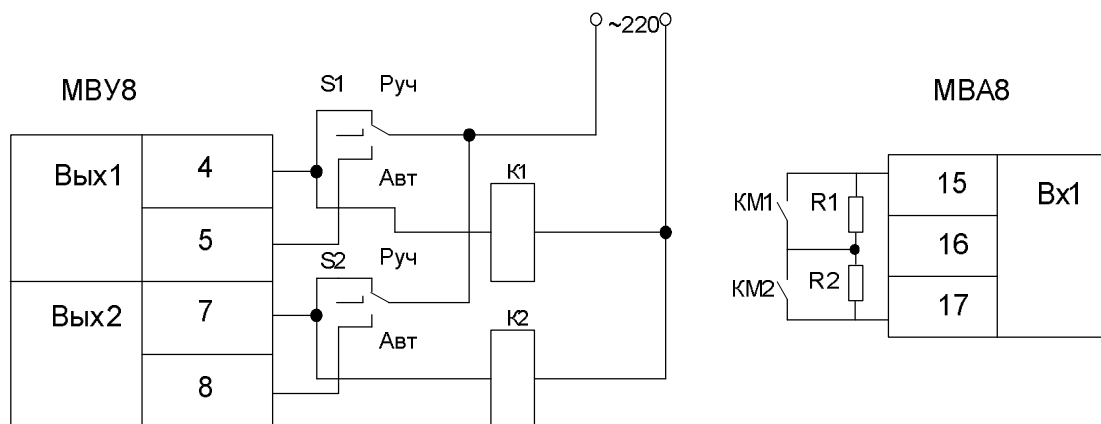


Рис. 11. Подсистема управления пускателями.

Пускатели, которые коммутируют силовые розетки, подключены к первому и второму выходам модуля МВУ8 типа «Р». Индикация включения пускателей производится при помощи первого входа МВА8 типа «сухие контакты», который может находиться в четырех состояниях, в зависимости от включения или выключения ПМ1 и ПМ2 соответственно. Трехпозиционные переключатели S1, S2 позволяют включать/выключать пускатели непосредственно вручную на стенде или переводят управление пускателями в автоматический режим.

*Подсистема управления исполнительным механизмом*

Информация о состоянии исполнительного механизма с его клеммной колодки (положение, срабатывание концевых выключателей) поступает на второй и третий входы модуля МВА8 «Резистивный датчик положения задвижки» (до 2.0кОм в нашем случае) и «сухие контакты» соответственно. Схема подключения приведена ниже на рис. 12.

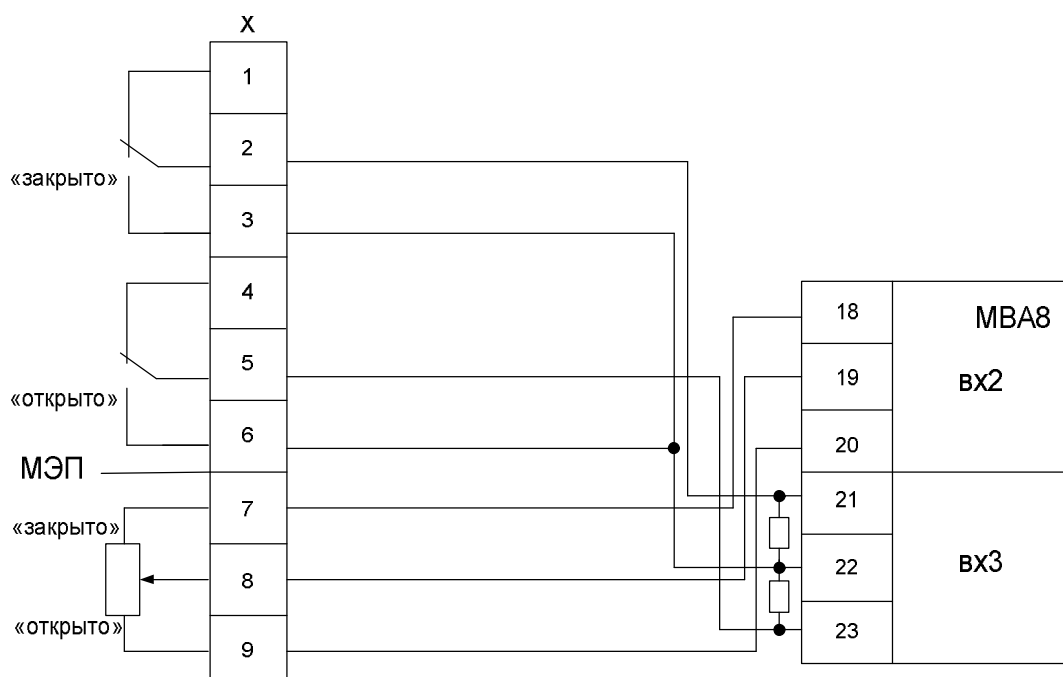


Рис. 12. Подсистема получения информации о состоянии ИМ.

Включение исполнительного механизма организовано с помощью пускателя бесконтактного реверсивного двухфазного ПБР-2М, позволяющего пускать ИМ как в прямую, так и в обратную сторону. Схема соединений приведена на рис. 13. Управляющие входы 7,9 и 10 ПБР подключены к третьему и четвертому выходам модуля МВУ8 типа «К» (транзисторный ключ) через переключатель перевода управления ПБР из автоматического в ручной режим S4. Управление ПБР в ручном режиме непосредственно на стенде осуществляется переключателем S3. Информация о состоянии режима ПБР передается системе управления через четвертый вход модуля МВА8 типа «сухие контакты».

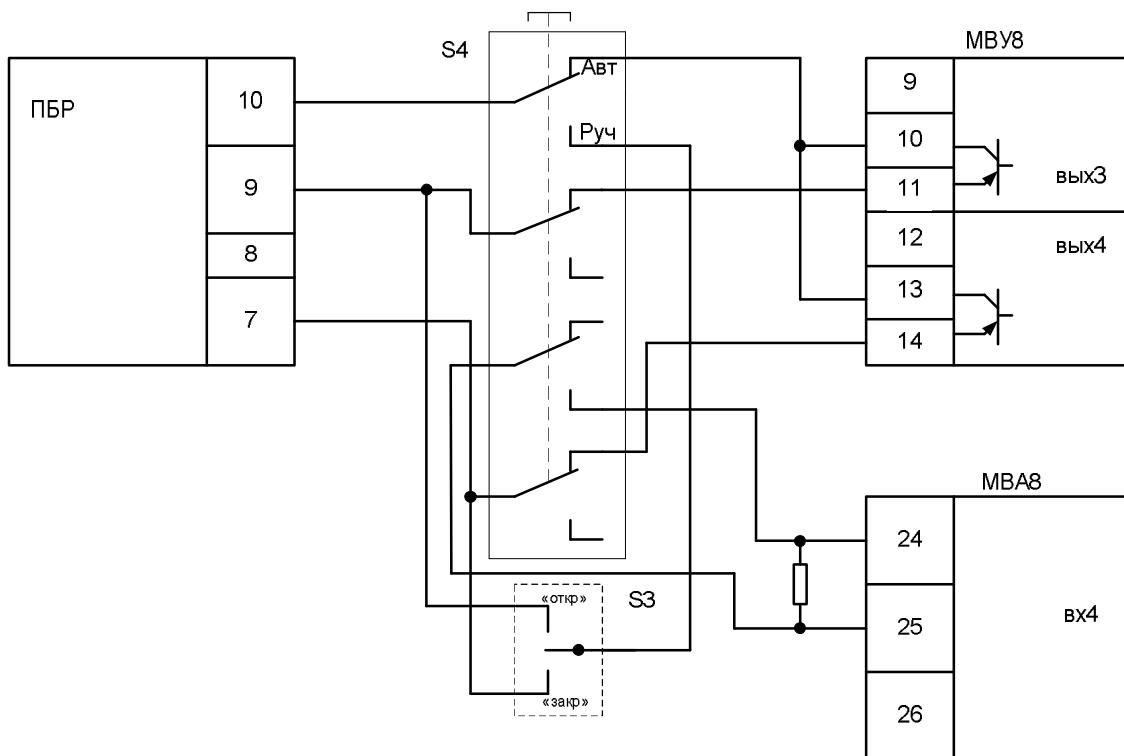


Рис. 13. Схема управления ПБР.

#### *Подсистема частотного управления электродвигателем*

Для управления частотой вращения двигателя вентилятора используется преобразователь частоты HitachiSJ 100. Управление системой может быть реализовано как в ручном, так и в автоматическом режимах. Переключение между режимами осуществляется посредством переключателя S5. Автоматическое управление осуществляет МВУ8. Управляющий вход «ОI» ПЧ подключен через переключатель S5 и ограничительное сопротивление «R<sub>огр</sub>» к седьмому выходу модуля. Ручное управление осуществляется с помощью переменного сопротивления R1 соединенного с входами ПЧ «Н», «О» и «L(-)». Оно механически связано с идентичным сопротивлением R3, подключенным к пятому входу МВА8. «Половинка» четвертого входа МВА8 служит для индикации режима ПЧ. Дискретный вход 1 ПЧ информирует ПЧ о том, с каких входов нужно считывать сигнал задания по частоте: с «О» (сигнал по напряжению – ручной режим) или «ОI» (токовый сигнал – автоматический режим). Схема подключения представлена на рис. 14.

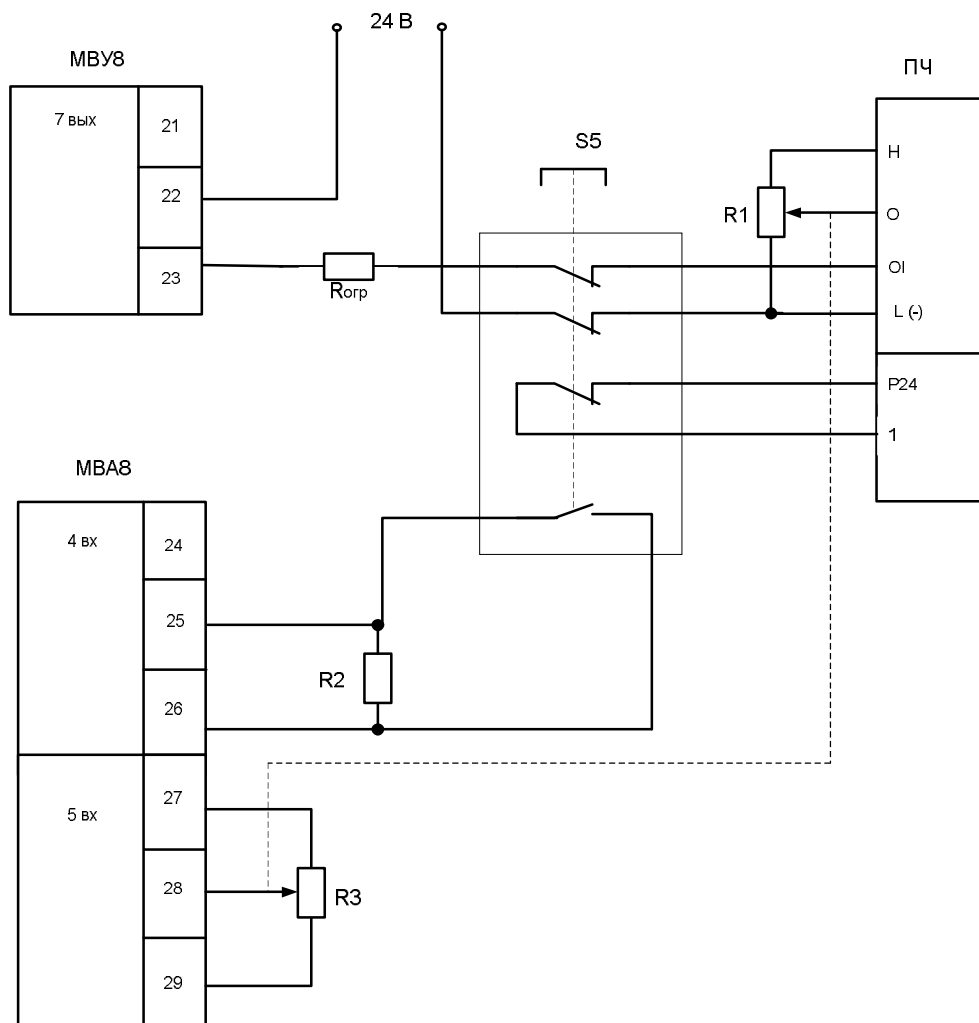


Рис. 14. Схема подключения ПЧ.

#### Подсистема управления тиристорами

Управление тиристорами осуществляется выходами 6 и 8 модуля МВУ8 типа «С» и «У» соответственно. Тиристоры могут управляться как напрямую модулем (прямое управление тиристорами), так и посредством БУСТ. Переключение режимов осуществляется переключателем S6. При его нажатии шестой выход МВУ8 через ограничительное сопротивление подключается напрямую к тиристорам, при этом происходит отсоединение БУСТ. Переключатель S7 осуществляет переход из автоматического в ручной режим управления БУСТ. В автоматическом режиме он связывает восьмой выход МВУ8 с управляющими входами БУСТ (X1), который через свои выходы (X2) управляет тиристорами. В ручном режиме управление БУСТ осуществляется посредством переменного сопротивления R1, которое жестко связано с идентичным сопротивлением R2, подключенным к шестому входу модуля МВА8. Схема подключения показана на рис. 15.

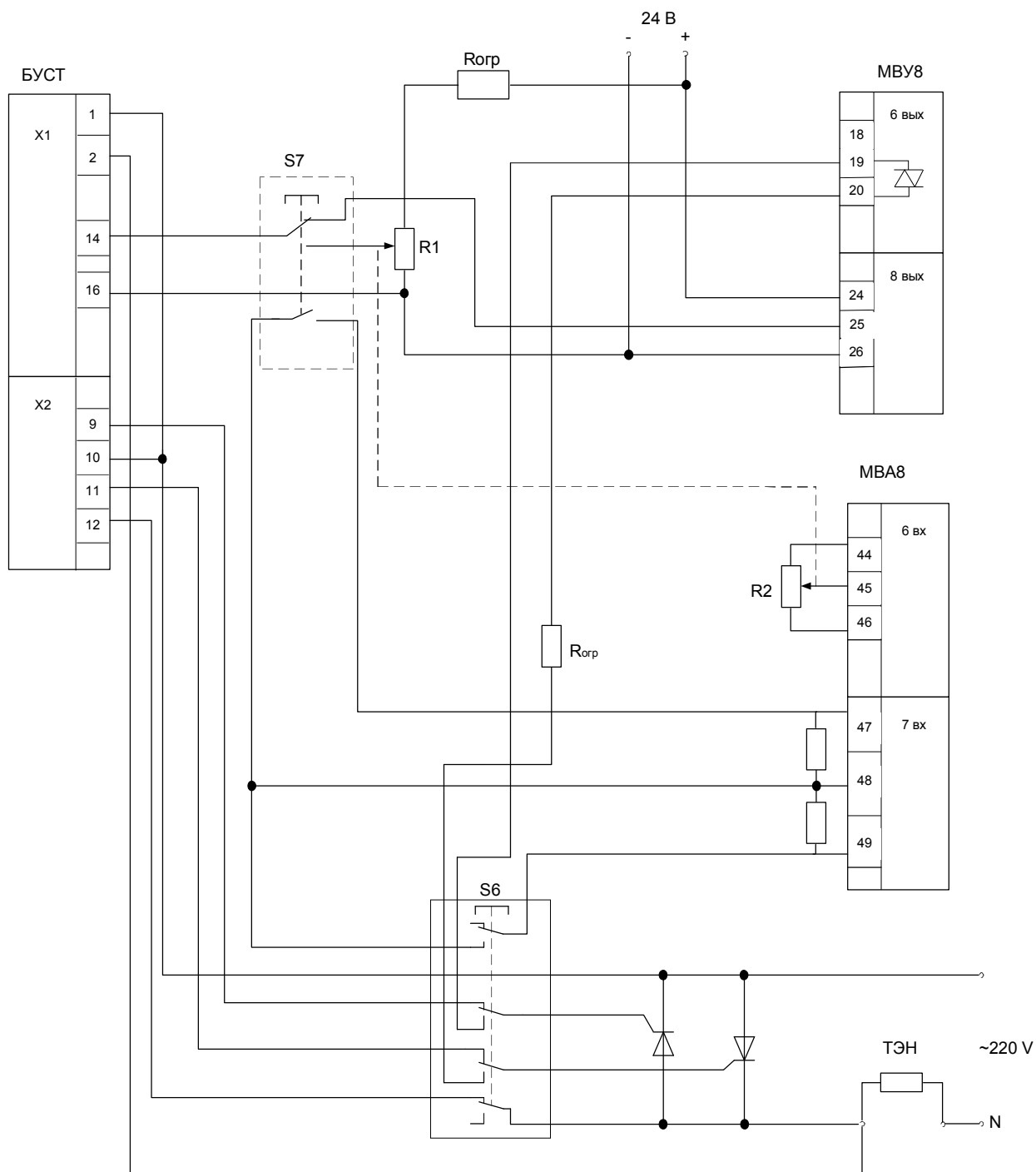


Рис. 15. Схема управления тиристорами.

### 3. Конфигурирование модулей ввода-вывода

Для конфигурирования прибора МВА8 необходимо подключить его через адаптер интерфейса RS-485 ОВЕН АС3 (или аналогичный) к персональному компьютеру и питать питание на прибор.

Конфигурирование производится с помощью программы «Конфигуратор МВА8» и включает настройку сетевых параметров и задание конфигурации МВА8. Конфигурация прибора задает полный набор значений параметров, определяющий работу прибора.

Порядок программирования прибора:

1. Запускается программа «Конфигуратор МВА8».

2. Устанавливается связь с прибором. Наличие связи определяется в процессе отправки команд по переходу на протокол ОВЕН. Установка связи происходит на сетевых параметрах, которые были установлены при прошлом запуске программы. Если связь установлена, открывается главное окно программы. Если программа не смогла установить связь с прибором и перевести МВА8 на работу по протоколу ОВЕН, открывается окно установки сетевых настроек рисунок 16.

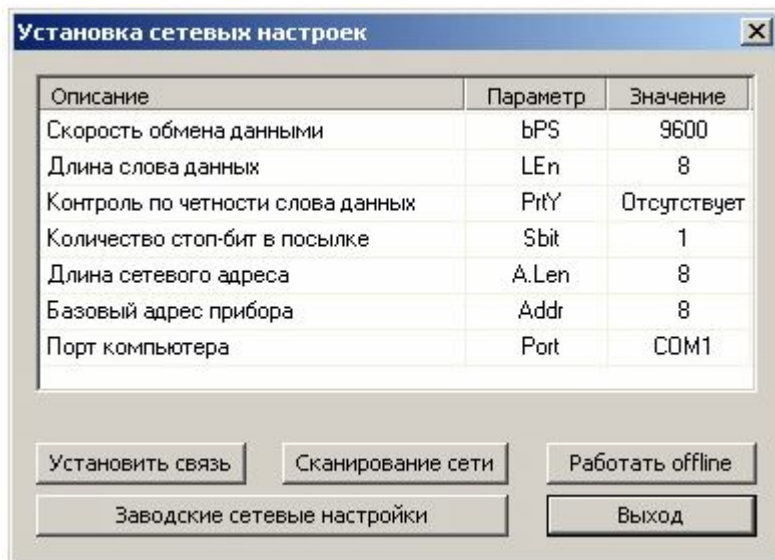


Рис. 16. Окно сетевых настроек.

В окне отображается информация о текущих сетевых настройках прибора. Значение параметров могут быть заданы непосредственно в таблице.

3. Открывается конфигурация из файла или создается новая.

4. Задается для каждого используемого входа тип датчика, период опроса и другие характеристики.

Данные параметры задаются в главном окне программы, которое показано на рис. 17.

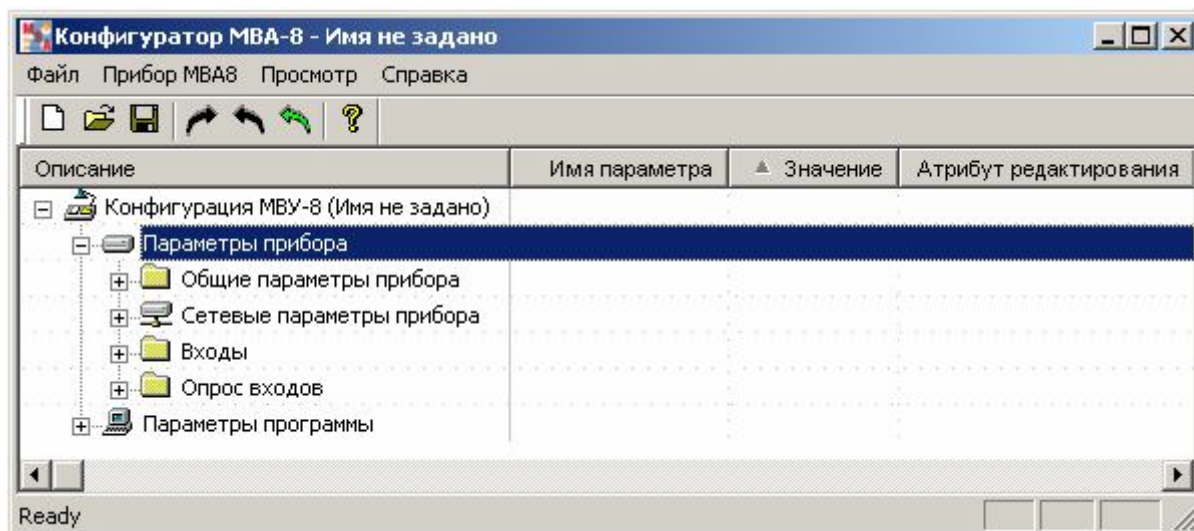


Рис. 17. Главное окно программы.

5. Для активных датчиков задается верхняя и нижняя границы измеряемого диапазона.

6. Записывается полученная конфигурация в прибор, если нужно, сохраняется в файл.

Главное окно программы включает титульную строку, в которой отображается запись «Конфигуратор МВА8» и наименование текущей конфигурации прибора, меню и панель инструментов, область отображения перечня разделов и параметров программы и прибора (в левой части окна) и область отображения значений параметров (в правой части).

При работе с перечнем параметров, чтобы развернуть пункт перечня, следует щелкнуть левой кнопкой мыши на значке «плюс» около названия свернутого пункта; чтобы свернуть – по значку «минус» около названия развернутого пункта перечня.

Управление программой производится посредством меню, кнопок панели управления или горячих клавиш.

МВА8 необходимо сконфигурировать следующим образом (табл. 6, 7,8).

Таблица 6

### Сетевые параметры прибора МВА8

Описание	Имя параметра	Значение
Скорость обмена данными	bPS	по указанию преподавателя (по умолч. 9600)
Длина слова данных	Len	8
Контроль по четности слова данных	PrtY	Отсутствует
Количество стоп бит в посылке	Sbit	1
Длина сетевого адреса	A.Len	8
Базовый адрес прибора	Addr	8
Задержка ответа по RS-485,мс	RS.dl	2
Протокол обмена	Prot	Modbus RTU

Таблица 7

### Входы МВА8

Номер входа	Тип датчика
1	Датчики контактные (сухие)
2	Резистивный датчик задвижки до 2.0кОм
3	Датчики контактные (сухие)
4	Датчики контактные (сухие)
5	Резистивный датчик задвижки до 2.0кОм
6	Резистивный датчик задвижки до 2.0кОм
7	Датчики контактные (сухие)
8	ТХК(L)

### Параметры входов МВА8

Описание	Имя параметра	Значение
Постоянная времени цифрования	in.Fd	0
Интервал между измерениями	Itrl	0.500
Сдвиг характеристики датчика	in.SH	0
Наклон характеристики датчика	in.SL	1.000
Полоса цифрового фильтра	in.FG	0
Нижняя граница диапазона измерения	Ain.L	0
Верхняя граница диапазона измерения	Ain.H	100
Смещение десятичной точки	dP	0 для входов 1,3,4,7 1 для входов 2,5,6,8

*Юстировка МВА8* заключается в проведении технологических операций, обеспечивающих восстановление его метрологических характеристик в случае изменения их после длительной эксплуатации прибора. В данной работе юстировка применяется для настройки каналов ввода, связанных с резистивными датчиками.

#### *Юстировка датчика положения*

Порядок юстировки следующий:

1. Подключить к контактам выбранного входа датчик положения. Схема подключения выбирается в зависимости от типа датчика.
2. При выключенном питании перевести прибор в режим юстировки. Соответствующий переключатель вынесен на короб проводки.
3. Включить питание. Запустить программу конфигурирования и перейти в режим юстировки.
4. Выбрать тип **4** юстировки.
5. Выбрать тип датчика и входной канал прибора
6. Ввести код доступа в режим юстировки **118**.
7. Выполнить действия, предписываемые программой.
8. По окончании юстировки одного датчика аналогично провести юстировку остальных датчиков положения.
9. После проведения всех юстировок – выключить питание прибора. Установить переключатель в положение «Р».

#### *Работа МВА8 в сети Modbus*

Работа по протоколу Modbus может идти в режимах ASCII или RTU, в зависимости от значения параметра **Prot**. По протоколу Modbus можно считать результаты измерений каждого входа, время измерения и его статус. Считывание идет стандартными для протокола командами чтения группы регистров (команда номер 03 или 04).

Результаты измерения представляются в двух форматах: 4-х байтовых значениях с плавающей точкой (без времени) и 2-х байтовое целое. Целое число – это результат измерения, умноженный на 10 в степени, заданной парамет-



ром **dP**. Значение **dP** может быть равно 0, 1, 2, 3 и задается отдельно для каждого канала.

Оба формата можно считать независимо, каждое по своему адресу. Время измерения – это циклическое время с шагом 0.01 сек, передаваемое в двух байтах. Время точно соответствует времени проведения измерения в данном канале и при работе с ним (например, при вычислении дифференциальной составляющей при ПИД-регулировании) можно не учитывать задержку передачи по сети RS-485. Отсчет циклического времени начинается при включении прибора и каждые 65536 тактов (что соответствует 656.36 с) время обнуляется.

Статус измерения – это регистр протокола Modbus; значение в регистре содержит код исключительной ситуации, возникшей в результате измерения.

Таблица 9

### Регистры ModbusMBA8

Параметр	Тип	Адрес	
		(hex)	(dec)
Положение десятичной точки в целом значении для входа 1 (знач. DP)	int16	0000	0
Целое значение измерения входа 1 со смещением точки	int16	0001	1
Статус измерения входа 1 (код исключительной ситуации)	int16	0002	2
Циклическое время измерения входа 1	int16	0003	3
Измерение входа 1 в представлении с плавающей точкой	Float32	0004,0005	4,5
Положение десятичной точки в целом значении для входа 2 (знач. DP)	int16	0006	6
Целое значение измерения входа 2 со смещением точки	int16	0007	7
Статус измерения входа 2 (код исключительной ситуации)	int16	0008	8
Циклическое время измерения входа 2	int16	0009	9
Измерение входа 2 в представлении с плавающей точкой	Float32	000A,000B	10,11
...			
Положение десятичной точки в целом значении для входа 8 (знач. DP)	int16	002A	42
Целое значение измерения входа 8 со смещением точки	int16	002B	43
Статус измерения входа 8 (код исключительной ситуации)	int16	002C	44
Циклическое время измерения входа 8	int16	002D	45
Измерение входа 8 в представлении с плавающей точкой	Float32	002E,002F	46,47

Конфигурирование модуля MBY8 происходит по схожей с MBA8 схеме с помощью программы «Конфигуратор MBY8».

МВУ8 необходимо сконфигурировать следующим образом (таблица 10, 11).

Таблица 10

### Сетевые параметры МВУ8

Описание	Имя параметра	Значение
Скорость обмена данными	bPS	по указанию преподавателя (по умолч. 9600)
Длина слова данных	Len	8
Контроль по четности слова данных	PrtY	Отсутствует
Количество стоп бит в посылке	Sbit	1
Длина сетевого адреса	A.Len	8
Базовый адрес прибора	Addr	16
Задержка ответа по RS-485,мс	RS.dl	0
Протокол обмена	Prot	Modbus RTU

Таблица 11

### Выходы МВУ8

Описание	Параметр	Значение
Тип выходного элемента	POUt	Дискретный
Период ШИМ при управлении ВЭ из сети	THPD	00:05
Аварийное значение на ВЭ	o.ALr	0.000

#### Работа МВУ8 в сети Modbus

Обмен по протоколу Modbus может происходить в режимах ASCII или RTU в зависимости от заданного значения параметра **Prot**. По протоколу Modbus в прибор передается скважность генерации ШИМ или значение, подаваемое на аналоговый ВЭ (в диапазоне от 0 до 1000). Регистры записываются стандартной для этого протокола командой 16 (10h) и читаются командами 4 (04h) или 3 (03h).

Таблица 12

### Регистры Modbus МВУ8 (MP1 – модуль расширения)

Параметр	Ед. измерения	Значение	Тип	Адрес	
				(hex)	(dec)
Значение на выходе 1	0,1 %	0..1000	int16	0000	0000
Значение на выходе 2	0,1 %	0..1000	int16	0001	0001
....					
Значение на выходе 8	0,1 %	0..1000	int16	0008	0008
Значение на выходе 9 (MP1)	0,1 %	0..1000	int16	0009	0009
Значение на выходе 10 (MP1)	0,1 %	0..1000	int16	000A	0010
....					
Значение на выходе 16 (MP1)	0,1 %	0..1000	int16	000F	0015

Параметр	Ед. измерения	Значение	Тип	Адрес	
				(hex)	(dec)
Аварийное значение на выходе 1	0,1 %	0..1000	int16	0010	0016
Аварийное значение на выходе 2	0,1 %	0..1000	int16	0011	0017
....					
Аварийное значение на выходе 8	0,1 %	0..1000	int16	0018	0024
Аварийное значение на выходе 9 (MP1)	0,1 %	0..1000	int16	0019	0025
....					
Аварийное значение на выходе 16 (MP1)	0,1 %	0..1000	int16	001F	0031
Период ШИМ на выходе 1	1 сек	0..900	int16	0020	0032
Период ШИМ на выходе 2	1 сек	0..900	int16	0021	0033
....					
Период ШИМ на выходе 8	1 сек	0..900	int16	0028	0040
Период ШИМ на выходе 9 (MP1)	1 сек	0..900	int16	0029	0041
....					
Период ШИМ на выходе 16 (MP1)	1 сек	0..900	int16	002F	0047
Макс. сетевой тайм-аут	1 сек	0..600	int16	0030	0048

#### 4. Панель оператора ИП320

Панель ИП320 представляет собой человеко-машинный интерфейс, предназначенный для отображения и редактирования значений параметров ПЛК и др. приборов [4]. Логика работы панели ИП320 определяется потребителем в процессе конфигурирования панели.



Рис. 18. Панель оператора ИП320 – внешний вид.

Панель ИП320 предназначена для выполнения следующих функций:  
 отображение русских и латинских символов;  
 запись и чтение значений регистров ПЛК и/или других приборов;  
 отображение графических пиктограмм (индикаторы, графики, линейки и т. д.);

защита с помощью пароля от несанкционированного изменения значений параметров и перехода на другой экран;

отображение «списка тревог» (нештатных ситуаций) в режиме реального времени;

работа в режиме «Мастера сети» (Master) или «Подчиненного» (Slave-устройства).

Основные технические характеристики панели ИП320:

напряжение питания постоянного тока, В 20...28;

потребляемая мощность, Вт, не более 4;

интерфейсы связи RS-232, RS-485;

скорости работы интерфейсов, бит/с: 2400, 4800, 9600, 19200, 38400, 115200;

универсальный протокол обмена: Modbus RTU;

конструктивное исполнение: корпус щитового крепления;

степень защиты корпуса со стороны лицевой панели: IP65;

тип дисплея, диагональ: графический монохромный ЖК с подсветкой, 3.7";

разрешение дисплея, пиксел: 192×64;

размеры дисплея Д×Ш, мм: 100×35;








количество кнопок: 20;





размеры панели Д×Ш×Г, мм: 172×94×30;

масса, кг, не более 0,5.

Таблица 13

**Назначение кнопок панели**

Кнопка	Функциональное назначение
	Возвращает дисплей к главному (часто используемому) экрану проекта. Как правило, главным экраном назначается либо главное меню проекта, либо наиболее часто используемый экран проекта
	Используется для перемещения курсора при вводе числа
	Используется для перемещения курсора при вводе числа
	Используется для перехода между экранами, а также в режиме редактирования параметра для изменения его численного значения
	Используется для перехода между экранами, а также в режиме редактирования параметра для изменения его численного значения
	Запускает процедуру редактирования значения регистра: строка отображения регистра перейдет в режим редактирования (изменит цвет). Если текущий экран не содержит области редактирования значения регистра, процедура не будет запущена. Осуществляется переход между элементами редактирования в области текущего экрана
	Записывает измененное значение текущего регистра и включает режим редактирования следующего регистра. После редактирования последнего регистра текущего экрана – завершает процедуру редактирования регистров

	При нажатии этой кнопки вызывается «Список тревог» (перечень нестандартных ситуаций)
	В режиме редактирования параметра происходит обнуление значения параметра
	Задаёт положительный или отрицательный знак редактируемого значения
	Цифровые кнопки предназначены для ввода и редактирования численного значения

Для конфигурирования панели используется программа «Конфигуратор ИП320»[7]. Последовательность операций в программе следующая:

1. Запуск программы «Конфигуратор ИП320».
2. Создание нового или запуск существующего проекта.
3. Создание нового экрана или редактирование существующего экрана.
4. Настройка панели и сохранение изменений в проекте.
5. Загрузка проекта в панель оператора ИП320.

1. *Запуск программы* осуществляется посредством выполнения файла IP320.exe. Интерфейс программы прост и интуитивно понятен. Все кнопки панели инструментов снабжены «подсказками» и поэтому не требуют дополнительного описания.

2. *Создание нового проекта* предполагает определение режима работы устройства в сети (Master или Slave), а также параметров сетевого обмена.

В режиме Master ИП320 (рис. 19, а) становится *ведущим* устройством в сети Modbus и поэтому не имеет адреса. В режиме Slave ИП320 – *ведомое* устройство, поэтому требуется указать его адрес в сети (рис. 19, б). В обоих случаях необходимо настроить сетевые параметры (рис. 19, в).

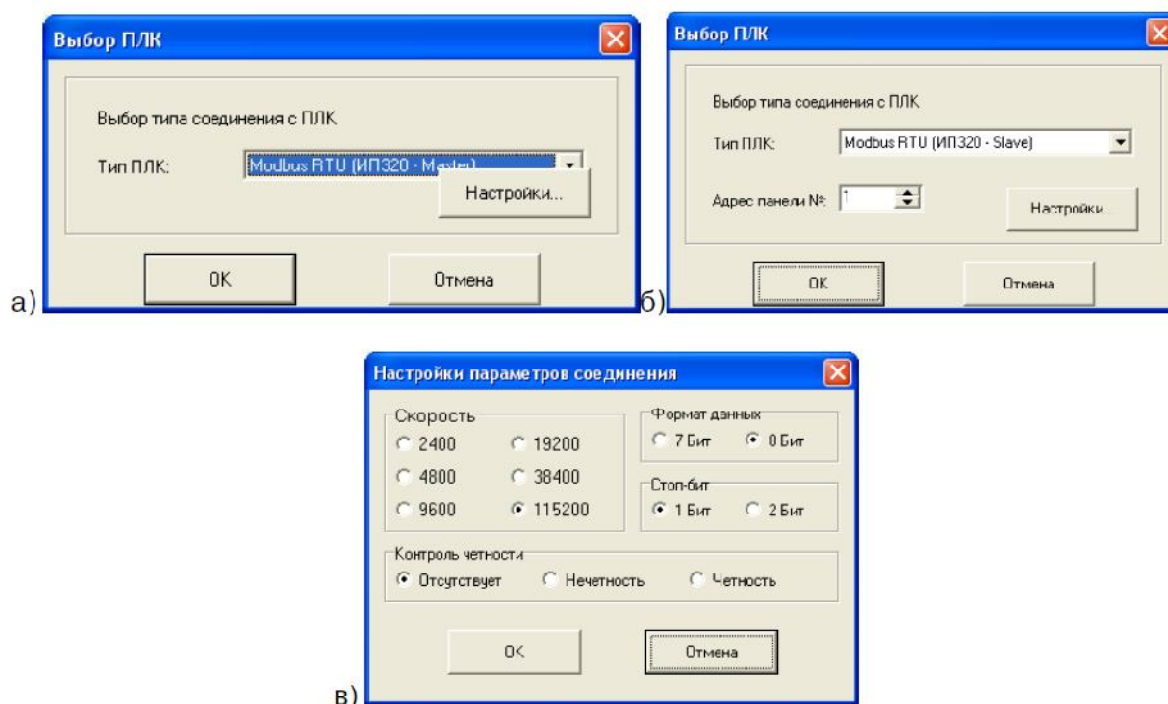


Рис. 19. Выбор ПЛК и настройка параметров соединения.

В окне «Настройки параметров соединения» задаются:  
скорость – скорость информационного обмена ПЛК (бит/сек);  
формат данных – длина слова данных информационного обмена. Следует выбрать только 8 бит, так как протокол Modbus RTU предполагает использование только такой длины слова данных;

контроль четности – задает тип контроля четности данных. Значение выбирается из вариантов: Отсутствует / Нечетность / Четность;

стоп-бит – задает количество стоповых бит. Выбирается «1» или «2».

Указанные параметры должны иметь одинаковые значения для всех устройств, работающих в сети.

В данной работе ИП320 будет выполнять роль ведущего устройства, опрашивая модуль МВА8 и управляя модулем МВУ8. Поэтому параметры сетевого обмена должны быть аналогичны таковым для модулей (см. табл. 6,10).

3. *Создание нового экрана* подразумевает задание его атрибутов и наполнение его содержимым.

В поле «Название» (рис. 20) вводится краткое название экрана. Оно должно позволить пользователю легко идентифицировать экран. Введенное наименование отобразится в колонке «Название» области «Экраны».

В полях «Предыдущий экран» и «Следующий экран» задаются идентификационные номера экранов, которые будут вызываться в режиме эксплуатации при нажатии кнопок со стрелками, направленными вверх и вниз.

Содержание экрана создается с помощью инструментов, расположенных в правой части Главного окна и приведенных ниже в табл. 14. Инструменты и элементы экрана, которые будут использоваться в данной работе, будут рассмотрены позднее.

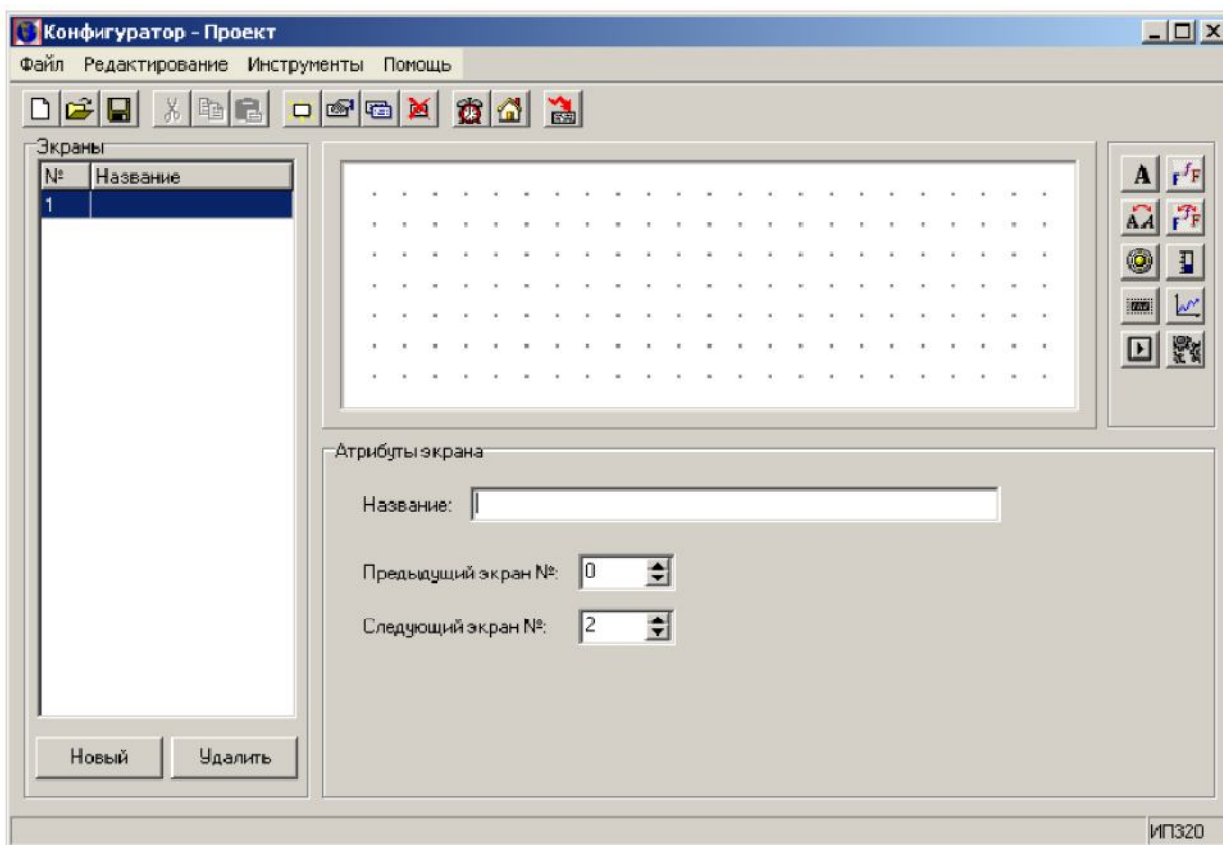


Рис. 20. Задание атрибутов экрана.

4. Для настройки панели оператора следует выбрать команду «Сервис | Настройка панели оператора» главного меню программы. Откроется окно «Настройки панели оператора», предназначенное для задания параметров проекта (рис. 21).

В этом окне задаются следующие параметры.

В группе полей «Настройки проекта»:

в поле «Главный экран» – указывается номер экрана, который будет переходить панель по нажатию кнопки [ESC] (обычно это либо экран главного меню, либо наиболее часто используемый экран);

в поле «Пароль» – задается числовой пароль, с помощью которого можно запретить доступ к функциям перехода на экран, перехода на экран тревог и экран времени и другим.

В поле «Выполнить» задается время, после которого в процессе работы панель выключит подсветку либо переключится на определенный номер экрана, который устанавливается рядом со словом «Экран». Если задать значение «Никогда», то панель не будет выполнять перехода на какой-либо экран либо выключать подсветку. По умолчанию стоит функция «Выключение подсветки через 3 минуты».

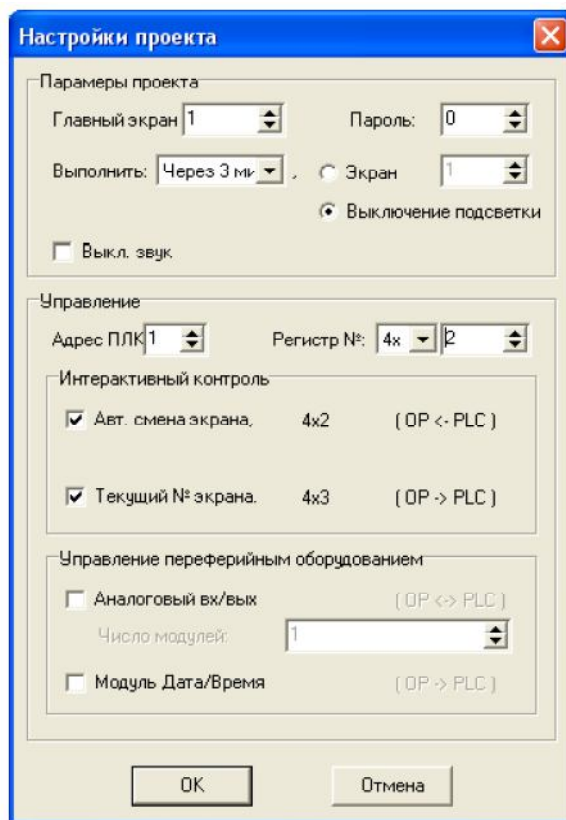


Рис. 21. Окно «Настройки проекта».

Переключатель «Экран/Выключение подсветки» задает режим выключение подсветки либо переход на экран с номером, заданным в поле справа, в который переходит панель по истечению времени, которое было установлено в поле «Выполнить»

Если установлен флажок «Выкл.звук», то нажатие на кнопки не будет сопровождаться звуковым сигналом.

В группе полей «Управление»:

поле «Адрес ПЛК» активно, когда режим работы панели ИП320 Master. В данном поле задается адрес ПЛК, с которого будет производиться управление экранами, либо этому ПЛК будет сообщаться текущий номер экрана;

поле «Регистр №» активно, когда в поле «Интерактивный контроль» устанавливается флажок «Автоматическая смена экрана» или флажок «Текущий № экрана». В поле «Регистр №» записывается стартовый номер регистра (в режиме Master – для ПЛК, в режиме Slave – для панели ИП320) для группы функций «Интерактивный контроль». Значения регистров отображается справа от слов «Автоматическая смена экрана» и «Текущий номер экрана», если данные функции активны.

В группе полей «Интерактивный контроль»:

если флажок «Автоматическая смена экрана» установлен, то в поле «Регистр №» следует установить номер регистра (для режима Master – это номер регистра в ПЛК, для режима Slave – это номер регистра в панели ИП320), в который в процессе работы ПЛК устанавливает значение номера экрана, на который нужно перейти. После того как панель перейдет на заданный номер экрана, она записывает в этот данный регистр ПЛК значение «0».



Если флажок «Текущий № экрана» установлен, то в поле «Регистр №»: в режиме Master следует установить номер регистра в ПЛК, в который в процессе работы панель будет отсылать номер активного экрана;

в режиме Slave следует установить номер регистра в панели ИП320, в котором будет храниться номер текущего экрана и который в процессе работы будет опрашивать ПЛК. Причем, если установлена функция «Автоматическая смена экрана», то номер регистра будет следующим за номером регистра этой функции.

Группа полей «Управление периферийным оборудованием» панелью не используется.

Созданный проект может быть сохранен на жестком диске в файле с расширением dr2.

5. Для загрузки проекта в панель следует подсоединить переходник для программирования к панели ИП320 и нуль-модемный кабель к СОМ-порту компьютера и к переходнику для программирования (рис. 22), включить питание панели ИП320 и вызвать команду меню Файл | Загрузить (или нажать соответствующую кнопку панели инструментов). Начнется загрузка, отображаемая окном монитора процедуры. По завершению загрузки появится информационное сообщение «Загрузка прошла успешно».



Рис. 22. Подсоединение панели ИП320 к компьютеру.

В используемом лабораторном стенде все соединения уже выполнены. При отсутствии связи следует убедиться в правильности установки номера СОМ-порта в меню Файл | СОМ - порт...











Элементы экранов панели ИП320 и соответствующие кнопки панели инструментов области редактирования экранов приведены в таблице 14.

Ниже подробно описаны те элементы, которые будут использоваться в данной работе.

Таблица 14

### Кнопки панели инструментов области редактирования экранов

Кнопка	Описание
	Текст.
	Графический текст. Символы отображаются в графическом режиме, который позволяет устанавливать размер и тип шрифта.

Кнопка	Описание
	Динамический текст, связанный с текущим состоянием регистра ПЛК.
	Графический динамический текст, связанный с текущим состоянием ПЛК. Отображаются в графическом режиме.
	Индикатор. Отображает состояние бита ПЛК.
	Линейка. Используется для нестрогого отображения аналоговых численных параметров (таких, как изменение давления, уровня жидкости и т.д.). Высота, ширина и направление элемента могут быть заданы произвольно.
	Регистр. Область отображения и ввода данных, связанная с регистром ПЛК.
	График. Используется для отображения изменений измеряемой величины.
	<p>Функциональная кнопка. Любая из 20 кнопок панели ИП320 могут быть определены как функциональные кнопки (т.е. их нажатие связывается с выполнением определенных функций). Использование функциональных кнопок позволяет, осуществлять переключение между экранами проекта, устанавливать значения битов и регистров.</p> <p>Если кнопкам панели ИП320 со стрелками «вверх» и «вниз» ( и ) , по умолчанию используемыми для переключения экранов, будут назначены в редактируемом экране другие функции, их нельзя будет использовать для переключения экранов.</p> <p>Если этим кнопкам не будут назначены в редактируемом экране другие функции, то их нажатие будет приводить к переключению экранов. При достижении начального экрана (экрана №1) произойдет циклическое переключение на последний экран проекта.</p> <p>Если этим кнопкам будут назначены в редактируемом экране функции редактирования данных, то, после выполнения этих функций, кнопки вновь могут быть использованы для переключения экранов.</p>
	Картинка. Данная функция позволяет вставить битовое (растровое) изображение (заранее созданное на компьютере размером 64x132) и позволяет отобразить на дисплее панели, например, чертежи механизмов, облегчающие понимание оператором протекающих процессов. Кроме того, позволяет отобразить товарные марки и т.п.

### Элемент экрана «Текст»

При нажатии кнопки «Текст» в окне редактирования режима отобразится прямоугольная область текста. Одновременно этот текст отображается в поле ввода «Текст», расположенном под окном редактирования (рис. 23). В этом окне текст можно отредактировать, используя символы (как кириллицы, так и латинского алфавита) и цифры. Возможности настройки данного элемента можно изучить экспериментально.

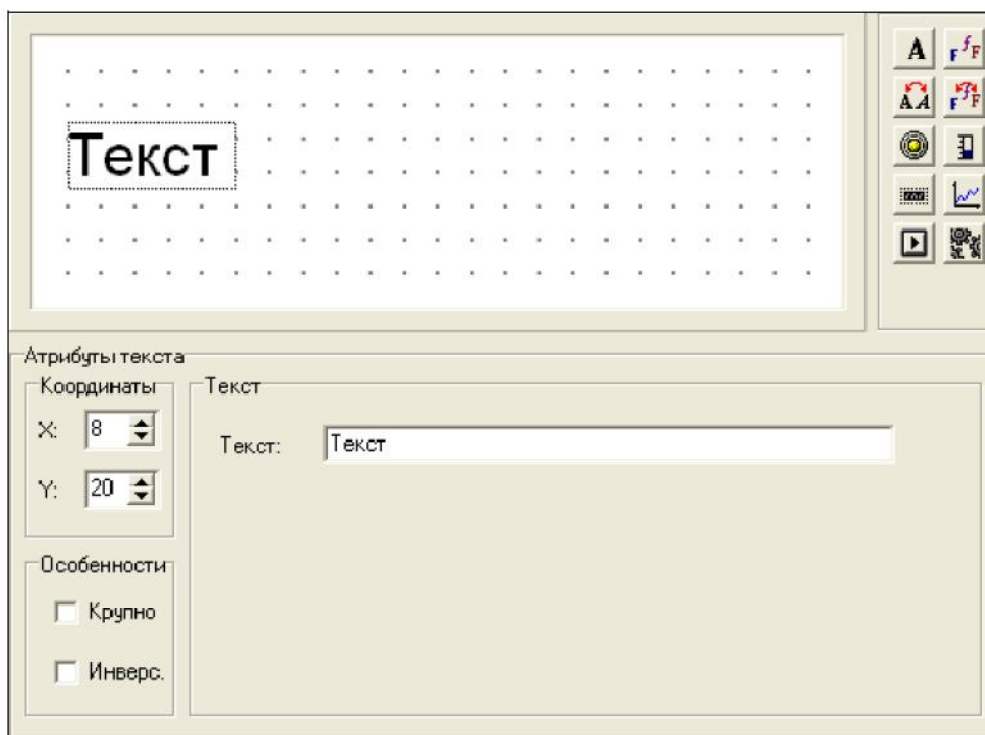


Рис. 23. Область отображения текста.

*Элемент экрана «Динамический текст»*

Элемент предназначен для словесного описания текущего состояния управляемого процесса. Текст динамически изменяется в процессе работы в зависимости от значения определенного регистра некоторого устройства (в терминологии конфигуратора – «ПЛК») сети Modbus (рис. 24).

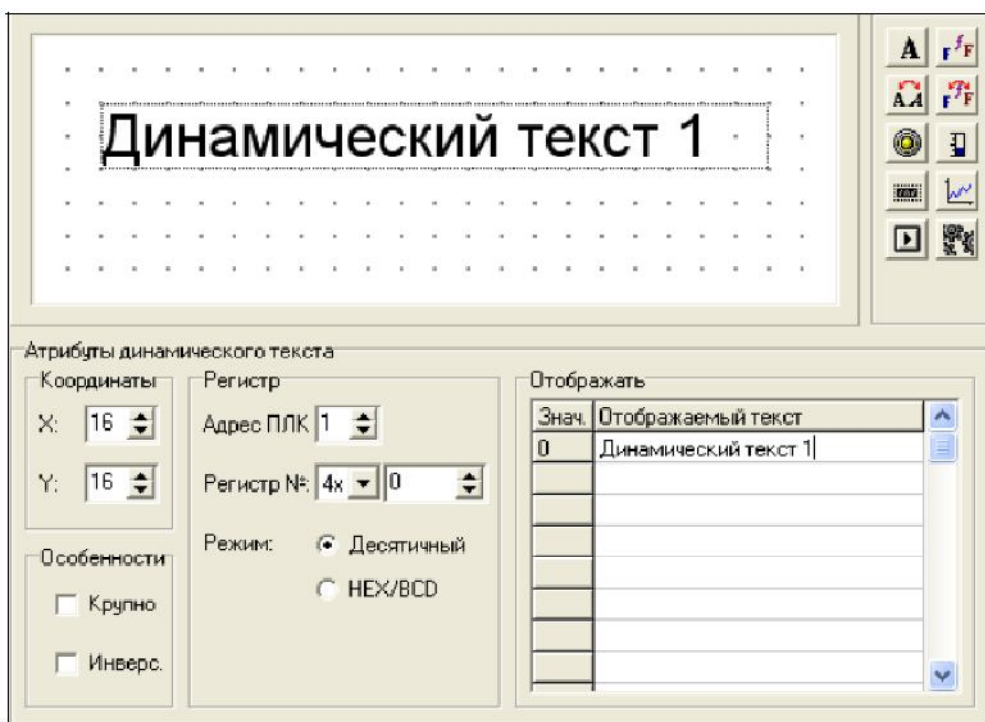


Рис. 24. Динамический текст.

В области «Регистр» следует задать:

в поле «Адрес ПЛК» в режиме Master – требуемый адрес устройства, с которого получают данные для переключения текста;

в полях «Регистр №»: в режиме Master – номер регистра устройства, связываемый с динамически меняющимся текстом; в режиме Slave – номер регистра в панели ИП320.

Переключатель «Режим» (возможные варианты «Десятичный/HEX/VCD») – задает отображение данных либо в десятичной системе счисления, либо отображение данных в режиме HEX (в шестнадцатеричной системе счисления/двоично-десятичной системе).

В области «Отобразить» задаются строки динамического текста. Они вводятся непосредственно в строки таблицы. Для ввода текста следует выбрать требуемую строку, щелчком левой кнопки мыши перевести ее в режим редактирования и ввести требуемый текст.

Содержание строк может редактироваться с применением копирования текста в буфер. В область ввода текста могут вводиться символы латиницы и цифры.

Связанная строка динамического текста отображается на дисплее панели тогда, когда заданный регистр принимает значение, соответствующее номеру строки (от 0 до 7).

В данной работе динамический текст будет использоваться для отображения значений каналов МВА8 типа «Датчики контактные сухие».

#### Элемент экрана «Функциональные кнопки»

Элемент предназначен для управления исполнительными механизмами и позволяет задействовать для этого кнопки панели. Окно настройки элемента показано на рис. 25.

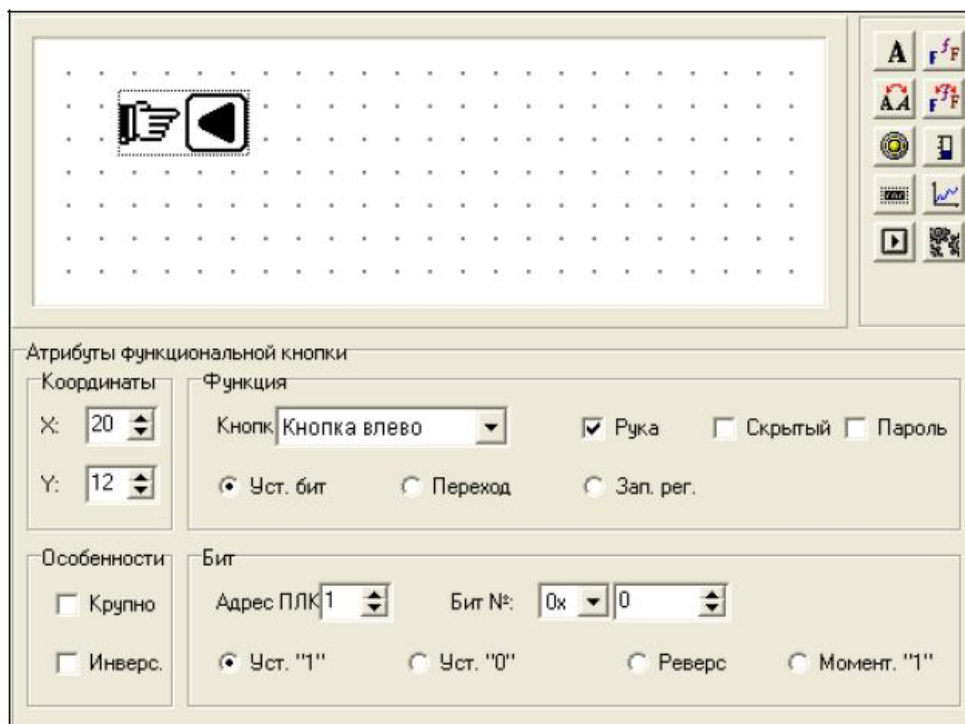


Рис. 25. Функциональная кнопка.

В области задания параметров в поле «Кнопка» следует выбрать из раскрывающегося списка двадцати доступных кнопок клавиатуры требуемую. Варианты: «Влево» ([ ]), «Вправо» ([ ]), «Вверх» ([ ]), «Вниз» ([ ]), «ESC», «SET», «ENT», «ALM», «CLR», «+/\_», «0», «1», «2», «3», «4», «5», «6», «7», «8», «9», «0». Выбранная кнопка отобразится на экране области редактирования.

При установке флажка переключателя «Рука» на экране отображается символ руки, нажимающей кнопку. Символ руки – необязательный элемент, который служит только напоминанием оператору о необходимости нажатия кнопки.

При установке флажка в поле переключателя «Крупно» размеры редактируемого элемента (кнопки) увеличиваются вдвое.

При установке флажка в поле переключателя «Инверс.» происходит инверсия цветов кнопки и фоновой подсветки.

При установке флажка в поле переключателя «Скрытый» кнопка на дисплее панели не показывается.

При установке флажка переключателя «Пароль» редактируемый элемент будет доступен только после ввода системного пароля.

Для выбора действия, которое будет выполняться при нажатии кнопки, используется переключатель «Уст.бит / Переход / Зап.рег.»:

«Уст. бит» –кнопка будет предназначена для задания установки значения бита в устройстве;

«Переход» – для переключения экранов;

«Зап. рег.» – установки значения регистра устройства по нажатию кнопки.

При выборе позиции «Уст. бит» активируется группа полей «Бит» (рис. 25), в которой следует задать функцию, которая будет выполняться при нажатии кнопки.

Поле «Адрес ПЛК» в режиме Master задает адрес управляемого устройства. Поле «Бит №» в режиме Master задает требуемый номер бита устройства (того, значение которой будет устанавливаться кнопкой), в режиме Slave задает номер бита в панели ИП320.

Переключатель «Уст. «1» / Уст. «0» / Реверс / Момент. «1» позволяет выбрать одну из следующих функций:

«Уст. «1» – задает состояние бита «1»;

«Уст. «0» – задает состояние бита «0»;

«Реверс» – изменяет состояние бита на противоположное;

«Момент. «1» – при нажатой кнопке устанавливает бит в состояние «1», соответственно при опущенной кнопке устанавливает бит в состояние «0».

При выборе позиции «Переход» кнопка будет предназначена для переключения экранов; активируется группа полей «Экран», в которой следует задать параметры того экрана, в который будет произведено переключение по нажатию редактируемой кнопки (рис. 26).

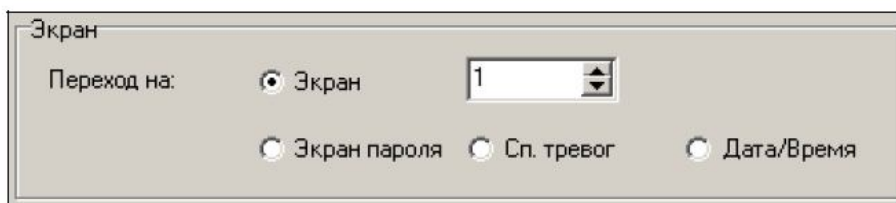


Рис. 26. Параметры переключения экрана.

В этой области следует выбрать требуемую позицию переключателя «Переход на:»:

«Экран» задает переключение на экран, номер которого следует задать в поле справа, в который будет производиться переключение по нажатию кнопки;

«Экран пароля» задает переключение на экран ввода пароля. При нажатии функциональной кнопки панель перейдет на экране открытия/закрытия пароля;

«Сп. тревог» задает переключение на экран списка тревог.

При выборе позиции «Зап. рег.» активируется группа полей «Регистр», в которой следует задать параметры регистра управляемого устройства, значение которого будет задаваться при нажатии функциональной кнопки (рис. 27).

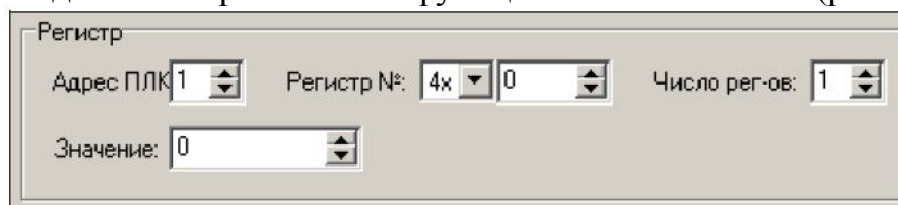


Рис. 27. Параметры ввода данных.

В полях этой области задаются:

«Адрес ПЛК» – в режиме Master задает адрес управляемого устройства;

«Регистр №» – в режиме Master задает номер регистра в управляемом устройстве, в который записывается значение («4x» и номер); в режиме Slave – номер регистра в панели ИП320, в который записывается значение;

«Число рег-ов» задает число регистров (1 или 2), в которые производится запись (два регистра может потребоваться, если записываемое значение не помещается в один);

«Значение» – значение, которое будет записано при нажатии функциональной кнопки в регистр.

В данной работе функциональные кнопки будут использоваться для управления каналами МВУ8, которые связаны с механизмами, работающими по принципу «включено/выключено»: магнитными пускателями и электроприводом задвижки. Для включения механизма необходимо в соответствующий регистр МВУ8 записать число 1000, для отключения – 0 (см. табл. 12). В случае записи в регистр промежуточного значения, механизм будет работать в импульсном режиме с соответствующей скважностью импульсов включения, что может привести к выходу его из строя, если период ШИМ мал.

### Элемент экрана «Регистр»

Элемент предназначен для отображения и изменения численных величин. С его помощью выводятся на экран значения регулируемых и других величин, формируются сигналы управления и задания. Окно настройки элемента показано на рис. 28.

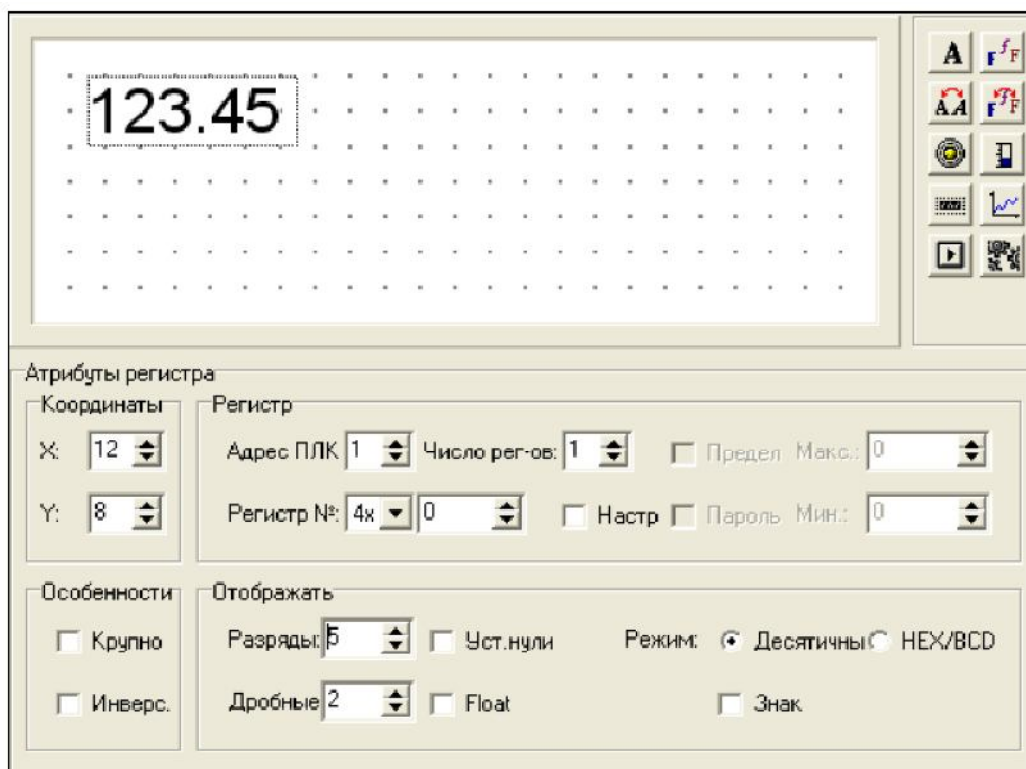


Рис. 28. Настройка элемента «Регистр».

В области «Регистр» указываются:

«Адрес ПЛК» (в режиме Master) – адрес устройства. В поле «Число регистров» задается число регистров (1 или 2), связываемое с читаемым параметром;

«Регистр №»: в режиме Master – номер регистра в устройстве, в котором располагается читаемый параметр; в режиме Slave – номер регистра в панели ИП 320;

Переключатель «Настр.» задается возможность редактирования значения регистра (или двух регистров).

Когда флажок переключателя установлен, то активируются переключатель «Предел», поля задания минимального и максимального значения контролируемого регистра, а также переключатель «Пароль» для задания необходимости ввода пароля для редактирования изменяемой величины. При эксплуатации вход в режим редактирования параметра осуществляется нажатием кнопки SET. Если на экране две или более редактируемые величины, то последовательное нажатие кнопки SET приводит к переключению между ними.

В области «Отображать» задаются следующие настройки:

в поле «Разряды» указывается общее количество символов отображаемой величины;

в поле «Дробные» указывается количество знаков после запятой в отображаемой величине. При задании ненулевого числа в этом поле десятичная точка при отображении смещается, т.е. число делится на десять в степени, заданной в этом поле.

Переключатель «Уст.нули» задает режим, когда отображаются все разряды числа, количество которых задано в поле «Разряды». Если значащих разрядов в отображаемом числе меньше, то число дополняется спереди нулями.

Переключатель «float» задает тип параметра float.

Переключатель «Режим» (возможные варианты «Десятичный/HEX/BSCD») задает режим получения и отображения данных либо десятичной системе счисления, либо двоично-десятичной системе. Для ОВЕН ПЛК применяется десятичная система счисления. Переключатель «Знак» активен только если выбрано отображение в десятичной системе счисления. Установка переключателя определяет тип числа, получаемого из регистра ПЛК: знаковый или беззнаковый.

В данной работе элемент экрана «Регистр» будет использоваться:

для отображения непрерывных сигналов, поступающих от МВА8: температуры, положения потенциометров ручного управления. Эти величины должны отображаться с точностью до одного знака после запятой. Поскольку ИП320 способна обрабатывать только целые числа, при конфигурировании МВА8 следует предусмотреть умножение значений сигналов на 10 перед посылкой их в сеть (установкой параметра каналов  $dP = 1$ ), а при конфигурировании ИП320, соответственно, деление значений сигналов на 10 при отображении (установкой значения «1» в поля «Дробные»);

для формирования сигналов управления преобразователем частоты и тиристорным регулятором, выдаваемых через модуль МВУ8. На панели ИП320 эти сигналы должны принимать значения в диапазоне от 0 до 1000. Нулевому значению редактируемой величины соответствует нулевая частота задания ПЧ и нулевая мощность тиристорного регулятора; значению 1000 – частота 50 Гц и мощность 100%.

### ***Программа работы***

*1. Изучение лабораторного стенда. Аprobация органов ручного управления*

1.1. Ознакомьтесь с теоретическими сведениями, приведенными выше.

1.2. Отключите шкаф от питания.

1.3. Подключите нагревательный элемент к розетке, связанной с тиристорами, вентилятор – к розетке, связанной с ПЧ, водяной насос – к розетке, связанной с ПМ1, силовой и информационный кабели МЭП – к соответствующим розетке и разъему. Все розетки установлены на боковой стенке щита справа.

1.4. Убедитесь в наличии достаточного уровня воды в баке (вода с запасом должна закрывать нагревательный элемент).



1.5. Откройте дверцу шкафа. Включите автоматические выключатели питания. Закройте дверцу шкафа. Подключите шкаф к питанию.

1.6. Переведя выключатели S1 и S2 в положение «Руч», наблюдайте срабатывание магнитных пускателей (при срабатывании пускателя 1 включится водяной насос). Переведите выключатели S1 и S2 в положение «Авт».

1.7. Переведите переключатель S4 в положение «Руч». С помощью переключателя S3 включите привод МЭП сначала в направление «меньше», потом – «больше». Переведите переключатель S4 в положение «Авт».

1.8. Переведите переключатель S5 в положение «Руч». Включите преобразователь частоты в режим «Run» (с помощью кнопки его лицевой панели). Изменяя положение потенциометра R1, наблюдайте изменение частоты вращения привода вентилятора. Переведите переключатель S5 в положение «Авт».

1.9. Переведите переключатель S6 в положение «БУСТ», переключатель S7 в положение «РУЧ». Изменяя положение потенциометра R2, наблюдайте изменение сигнала управления БУСТ по миллиамперметру. Переведите переключатель S7 в положение «Авт».

## *2. Конфигурирование прибора МВА8. Аprobация работы прибора*

2.1. Подключите персональный компьютер к сети RS-485 через преобразователь интерфейса. Включите персональный компьютер. **Внимание: «горячее» подсоединение преобразователя к компьютеру через интерфейс RS-232 не допускается!**

2.2. Запустите программу «Конфигуратор МВА8» и установите связь с прибором.

2.3. Проведите конфигурирование входов МВА8 для работы в сети Modbus. Проведите юстировку всех «резистивных датчиков» (входы 2, 5, 6).

2.4. Проведите опрос всех входов. Удостоверьтесь в правильности работы модуля.

## *3. Конфигурирование прибора МВУ8. Аprobация работы прибора*

3.1. Подключите персональный компьютер к сети RS-485 через преобразователь интерфейса. Включите персональный компьютер.

3.2. Запустите программу «Конфигуратор МВУ8» и установите связь с прибором.

3.3. Проведите конфигурирование входов МВУ8 для работы в сети Modbus. Режим работы – непосредственное управление.

3.4. Проведите опрос всех выходов (в данном случае под опросом понимается изменение состояния). Удостоверьтесь в правильности работы модуля.

## *4. Конфигурирование панели оператора ИП320*

4.1. Запустите программу «Конфигуратор ИП320» и создайте новый проект. Режим работы панели – Master.

4.2. Создайте «главный» экран №1, на котором будет отображаться температура воды в баке. Элементы экрана: «Текст» (для пояснения), «Регистр». Включите панель ИП320 и загрузите проект в ИП320. Убедитесь в правильности работы экрана.

4.3. Создайте экран №2 «Управление магнитными пускателями». Экран должен содержать элементы:

отображения состояния магнитных пускателей (элемент «Динамический текст»);

управления магнитными пускателями (элементы «Функциональная кнопка», по две кнопки на каждый пускатель).

Загрузите проект в ИПЗ20. Убедитесь в правильности работы экрана.

4.4. Создайте экран №3 «Управление воздушной заслонкой». Экран должен содержать элементы:

отображения состояния и положения воздушной заслонки (элемент «Динамический текст» для отображения состояния концевых выключателей привода, элемент «Регистр» для вывода текущего положения заслонки по сигналу резистивного датчика положения, элементы «Текст» для пояснений);

управления электроприводом заслонки (элементы «Функциональные кнопки», 3 кнопки: «Увеличить подачу воздуха», «Уменьшить подачу воздуха», «Остановить привод», элементы «Текст» для пояснений).

Загрузите проект в ИПЗ20. Убедитесь в правильности работы экрана.

4.5. Создайте экран №4 «Управление преобразователем частоты». Экран должен содержать элементы:

отображения режима управления (авт./руч.) преобразователем и сигнала управления преобразователем в ручном режиме (элемент «Динамический текст» для отображения режима, элемент «Регистр» для вывода текущего значения сигнала управления в ручном режиме, элементы «Текст» для пояснений);

управления выходной частотой ПЧ (элемент «Регистр» с возможностью редактирования, элемент «Текст» для пояснения).

Загрузите проект в ИПЗ20. Убедитесь в правильности работы экрана.

4.5. Создайте экран №5 «Управление тиристорным регулятором». Экран должен содержать элементы:

отображения режима управления (авт./руч.) тиристорным регулятором и сигнала управления в ручном режиме (элемент «Динамический текст» для отображения режима, элемент «Регистр» для вывода текущего значения сигнала управления в ручном режиме, элементы «Текст» для пояснений);

управления мощностью тиристорного регулятора (элемент «Регистр» с возможностью редактирования, элемент «Текст» для пояснения).

При разработке рассматривайте только вариант управления тиристорами посредством БУСТ (прямое управление тиристорами не реализуйте).

Загрузите проект в ИПЗ20. Убедитесь в правильности работы экрана.

### ***Содержание отчета***

1. Структурная схема лабораторного стенда.
2. Пошаговое описание проведенных процедур конфигурирования модулей МВА8, МВУ8.
3. Экраны проекта ИПЗ20 с указанием настроек всех элементов.

### ***Контрольные вопросы***

1. Назначение и функциональные возможности модуля ввода МВА8.
2. Какие входные сигналы способен измерять и обрабатывать модуль МВА8?
3. Приведите схему подключения к модулю МВА8 датчиков типа «сухие контакты».
4. Каким образом МВА8 описывает состояние датчиков типа «сухие контакты»?
5. Назначение и функциональные возможности модуля вывода МВУ8.
6. Какие типы выходных элементов может иметь модуль вывода МВУ8?
7. Какие типы выходных элементов установлены в используемом модуле вывода МВУ8?
8. Укажите и охарактеризуйте приборы и устройства, расположенные в лабораторном шкафу управления (непосредственно перед открытым шкафом).
9. Состав и назначение органов управления лицевой панели стенда (непосредственно перед дверцей шкафа).
10. Опишите каналы ввода и вывода модулей МВА8 и МВУ8, задействованные в лабораторном стенде по мнемосхеме на его лицевой панели (непосредственно перед дверцей шкафа).
11. Какие сетевые настройки МВА8 необходимо установить для организации связи с прибором (имя – значение)?
12. Назовите основные параметры входов МВА8, устанавливаемые при конфигурировании прибора.
13. Что такое юстировка? Опишите последовательность действий при юстировке резистивного датчика положения.
14. Назовите основные параметры выходов МВУ8, устанавливаемые при конфигурировании прибора.
14. Каков диапазон изменения значений переменных, связанных с выходными каналами модуля вывода МВУ8 при работе по протоколу Modbus?
15. Как поведет себя выходной элемент дискретного типа, если в соответствующий канал записать значение от 0 до 1000 (протокол Modbus)?
16. Назначение и функциональные возможности панели оператора ИП320.
17. Режимы работы ИП320 в сети.
18. Адресация регистров устройств в ИП320 в режимах Master и Slave.
19. Опишите функциональные возможности элемента экрана «Динамический текст». Как они используются в лабораторной работе?
20. Опишите функциональные возможности элемента экрана «Функциональные кнопки». Как они используются в лабораторной работе?
21. Каковы режимы работы элемента экрана «Функциональные кнопки» (три режима)?
22. Опишите функциональные возможности элемента экрана «Регистр». Как они используются в лабораторной работе?

23. Каким образом передаются данные вещественного типа из модуля МВА8 в панель ИП320 и как обеспечить требуемую точность их отображения на экране?

24. Каков диапазон передаваемых из ИП320 в МВУ8 числовых значений?

*Лабораторная работа №2. Система дистанционного управления лабораторным объектом на основе модулей ввода-вывода Овен МВА8, МВУ8, панели оператора ИП320 и ПЛК150 (ИП320 в режиме Slave)*

Цели работы:

знакомство с функциональными возможностями ПЛК150 как Master - устройства сети Modbus и получение навыков его конфигурирования;

разработка и реализация системы дистанционного управления лабораторным объектом на базе модулей ввода-вывода Овен МВА8, МВУ8 и панели оператора ИП320, работающей в режиме Master.

**Теоретические сведения**

**1. ПЛК150**

Программируемый логический контроллер ОВЕН ПЛК150 предназначен для создания систем автоматизированного управления технологическим оборудованием в различных областях промышленности, жилищно-коммунального и сельского хозяйства [6]. Логика работы ПЛК150 определяется потребителем в процессе программирования контроллера. Программирование осуществляется с помощью системы программирования CoDeSys 2.3 [10].

Контроллер ОВЕН ПЛК150 выпускается в корпусе, предназначенном для крепления на DIN-рейке 35 мм. Подключение всех внешних связей осуществляется через разъемные соединения, расположенные по двум боковым и передней (лицевой) сторонам контроллера.

Открытие корпуса для подключения внешних связей не требуется.

Схематический внешний вид контроллера показан на рис.29.

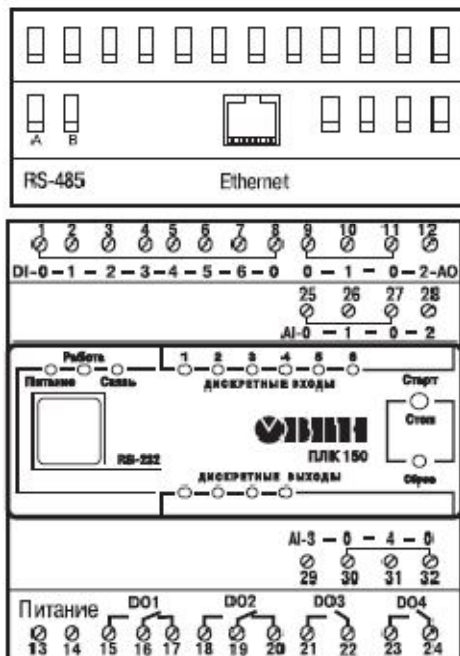


Рис. 29. Внешний вид ПЛК150.

На боковой стороне расположены разъемы интерфейсов Ethernet и RS-485.

На лицевой панели расположен порт Debug RS-232, предназначенный для связи со средой программирования, загрузки программы и отладки. Подключение к этому порту осуществляется кабелем, входящим в комплект поставки. Также порт Debug RS-232 может быть использован для подключения устройств, работающих по протоколам Modbus, OVEN и DCON.

По обеим боковым сторонам контроллера расположены клеммы для подключения дискретных датчиков и исполнительных механизмов.

Любой дискретный вход ПЛК150 может работать в режиме аппаратного счетчика или триггера (частота до 10 кГц при скважности 50%), к двум дискретным входам можно подключить энкодер (частота импульсов до 10 кГц). Частота обработки аппаратных счетчиков и обработчиков энкодера не зависит от времени выполнения цикла ПЛК.

На переднюю панель контроллера выведена светодиодная индикация о состоянии дискретных входов и выходов, о наличии питания и о наличии связи со средой программирования CoDeSys.

Также на передней панели имеются две кнопки: кнопка, предназначенная для запуска и остановки программы в контроллере, и скрытая кнопка, предназначенная для перезагрузки контроллера. Нажать кнопку возможно только тонким заостренным предметом.

В корпусе контроллера расположен маломощный звуковой излучатель, управляемый из пользовательской программы как дополнительный дискретный выход.

Звуковой излучатель может быть использован для функций аварийной или иной сигнализации или для отладочных нужд. Частота звукового сигнала излучателя фиксирована и не поддается настройке.

Контроллер ПЛК150 оснащен встроенными часами реального времени, имеющими собственный аккумуляторный источник питания. Энергии полностью заряженного аккумулятора хватает на непрерывную работу часов реального времени в течение 6 месяцев (при температуре 15–35°C). В случае износа аккумулятора, неполной его зарядки, а также при работе при более низких температурах время работы часов реального времени может сократиться.

Аккумулятор, используемый для питания часов реального времени, дополнительно используется как источник аварийного питания микропроцессора контроллера. При случайном отключении основного питания контроллер переходит на аварийное питание и сохраняет промежуточные результаты вычислений и работоспособность интерфейсов Ethernet в течение 10 минут. Светодиодная индикация и выходные элементы контроллера при этом не запрашиваются и не функционируют. При включении основного питания во время работы на аварийном питании контроллер сразу приступает к выполнению пользовательской программы, не тратя время на загрузку и сохраняя все промежуточные результаты вычислений. После 10 мин. работы на аварийном питании контроллер записывает Retain-переменные в энергонезависимую память и отключается. Часы реального времени остаются в рабочем состоянии. После включения основного питания контроллер загружается и запускает программу пользователя (ес-

ли установлена опция автозапуска). Время работы от аварийного источника питания может быть автоматически скорректировано самим контроллером в зависимости от степени зарядки аккумулятора и температуры окружающей среды.

Для полной зарядки аккумулятора требуется не менее пяти часов бесперебойной подачи основного питания.

Во время загрузки контроллера его выходы переводятся в заранее заданное «безопасное состояние», в которых находятся до полной загрузки контроллера и запуска пользовательской программы. «Безопасное состояние» – это состояние выходов контроллера, при котором подключенные к ним исполнительные механизмы находятся в состоянии, наиболее безопасном для объекта управления, не приводящим к его поломке. Значение «безопасного состояния» выходов задается при конфигурировании контроллера.

Основные технические характеристики, характеристики входных сигналов и характеристики встроенных выходных элементов контроллера ПЛК150 приведены в табл. 15, 16, 17.

Таблица 15

### Основные технические характеристики контроллера ПЛК150

Параметр	Значение
Общие сведения	
Конструктивное исполнение	Унифицированный корпус для крепления на DIN-рейку, длина 105 мм (6U), шаг клемм 7,5 мм
Степень защиты корпуса	IP20
Напряжение питания: ПЛК150-24 ПЛК150-220	18...29 В постоянного тока (номинальное 24 В) 90...264 В переменного тока (номинальное 220 В) частотой 47...63 Гц
Потребляемая мощность	6 Вт
Индикация передней панели	1 индикатор питания 6 индикаторов состояний дискретных входов 4 индикатора состояний выходов 1 индикатор наличия связи с CoDeSys 1 индикатор работы программы пользователя
Ресурсы	
Центральный процессор	32-х разрядный RISC-процессор 200 МГц на базе ядра ARM9
Объем оперативной памяти	8 МВ
Объем энергонезависимой памяти хранения ядра CoDeSys программ и архивов*	4 МВ
Размер Retain-памяти	4 кВ
Время выполнения цикла ПЛК	Минимальное 250 мкс, типовое от 1 мс
Дискретные входы	
Количество дискретных входов	6

Параметр	Значение
Гальваническая изоляция дискретных входов	есть, групповая
Электрическая прочность изоляции дискретных входов	1,5 кВ
Максимальная частота сигнала, подаваемого на дискретный вход	1 кГц при программной обработке 10 кГц при применении аппаратного счетчика и обработчика энкодера
Дискретные выходы	
Количество дискретных выходов	4 э/м реле
Характеристики дискретных выходов	Ток коммутации до 2 А при напряжении не более 220 В 50 Гц и $\cos\varphi > 0,4$
Гальваническая изоляция дискретных выходов	есть, индивидуальная
Электрическая прочность изоляции дискретных выходов	1,5 кВ
Аналоговые входы	
Количество аналоговых входов	4
Типы поддерживаемых унифицированных входных сигналов	Напряжение 0...1 В, 0...10 В, -50...+50 мВ. Ток 0...5 мА, 0(4)...20 мА Сопротивление 0...5 кОм
Типы поддерживаемых датчиков	Термосопротивления: ТСМ50М, ТСП50П, ТСМ100М, ТСП100П, ТСН100Н, ТСМ500М, ТСП500П, ТСН500Н, ТСП1000П, ТСН1000Н Термопары: ТХК (L), ТЖК (J), ТНН (N), ТХА (K), ТПП (S), ТПП (R), ТПР (B), ТВР (A-1), ТВР (A-2)
Время опроса одного аналогового входа	0,5 с
Предел основной приведенной погрешности измерения аналоговыми входами	0,5 %
Гальваническая изоляция аналоговых входов	отсутствует
Аналоговые выходы	
Количество аналоговых выходов	2
Разрядность ЦАП	10 бит
Тип выходного сигнала: ПЛК150-И ПЛК150-У ПЛК150-А	Ток 4...20 мА Напряжение 0...10 В Ток 4...20 мА или напряжение 0...10 В
Питание аналоговых выходов	встроенное, общее на все выходы
Гальваническая изоляция аналоговых выходов	есть, групповая
Электрическая прочность изоляции аналоговых выходов	1,5 кВ
Интерфейсы связи	
Интерфейсы	Ethernet 100 Base RS-232



Параметр	Значение
	RS-485
Скорость обмена по интерфейсам RS	от 4800 до 115200 bps
Протоколы	ОВЕН Modbus-RTU, Modbus-ASCII DCON Modbus-TCP GateWay (протокол CoDeSys)
Программирование	
Среда программирования	CoDeSys 2.3
Интерфейс для программирования и отладки	RS&232 или Ethernet

Таблица 16

### Характеристики дискретных входных сигналов

Модификация контроллера	Сигнал, подаваемый на дискретный вход	Комментарий
ПЛК150-24	29...17 В* – логическое значение 1 5...0 В* – логическое значение 0	Вход срабатывает при протекающем через него токе не менее 3 мА
ПЛК150-220	С помощью сухого контакта или ключа, коммутирующего общую клемму контакта и линии подключения дискретных входов и клемму конкретного входа	Суммарное сопротивление должно быть не более 100 Ом

Таблица 17

### Характеристики встроенных аналоговых выходных элементов

Обозначение при заказе	Наименование	Характеристики
И	Цифроаналоговый преобразователь "параметр – ток 4...20 мА"	Сопротивление нагрузки от 0 до 900 Ом
У	Цифроаналоговый преобразователь "параметр – напряжение 0...10 В"	Сопротивление нагрузки от 2 кОм
А	Цифроаналоговый преобразователь параметр – ток 4...20 мА или напряжение 0...10 В"	Сопротивление нагрузки от 150 до 900 Ом для токового сигнала и свыше 10 кОм для сигнала напряжения

### **2. Конфигурирование ПЛК150 для работы в качестве Master-устройства сети Modbus**

Утилита PLC Configuration (Конфигуратор ПЛК) доступна пользователю на вкладке ресурсов (Resources) Организатора объектов среды разработки CoDeSys (далее – CoDeSys) (см. рис. 30). Конфигуратор является редактором ресурсов ПЛК, относящихся к области ввода-вывода, через которую программа

ПЛК осуществляет информационный обмен с внешней средой [5]. В конфигурации присутствуют модули, отвечающие за структурирование областей ввода и/или вывода, каждый из которых может содержать вложенные подэлементы (субмодули и каналы). Для каналов могут быть назначены символические имена (прямые МЭК адреса отображаются в конфигурации для каждого символического имени).

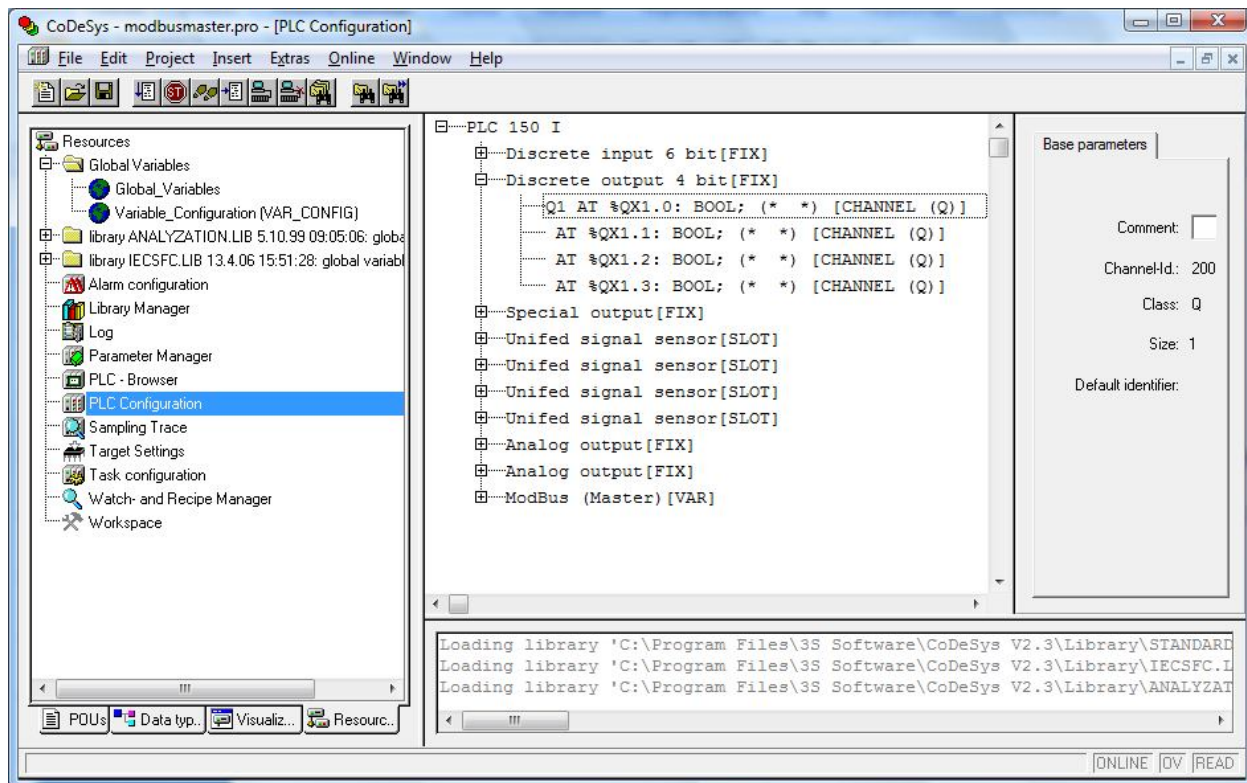


Рис. 30. Окно PLC Configuration.

Существует два вида модулей:

*фиксированные*– жестко задаются и не могут быть удалены или заменены. Допускается только редактирование их параметров;

*добавляемые*– вставляются по желанию пользователя во время конфигурирования, подразделяются на два типа:

*тип SLOT*– означает, что зарезервировано место для модуля, которое может быть занято или оставлено пустым. На одно зарезервированное место может быть установлен один модуль;

*свободный тип (VAR)*– означает возможность установить любое количество модулей (с учетом физических возможностей области ввода/вывода).

Добавление модулей осуществляется иницированием опции Append Subelement («Добавление подэлемента») в контекстном меню, вызываемом нажатием правой кнопки манипулятора «мышь» при установке курсора в дереве Конфигурации ПЛК, и последующим выбором конкретного модуля из выпадающего списка. После появления экранной формы модуля его параметры соответствующим образом настраиваются, как это необходимо для функционирования этого подэлемента.

*Модуль Modbus (Master)*

Для того, чтобы ПЛК работал по протоколу Modbus в режиме «ведущего» (Master), необходимо установить модуль Modbus (Master).

При опросе модулем Modbus (Master) подчиненных устройств информация о ходе обмена записывается в его переменных.

Каналы модуля:

Last Address – адрес последнего опрошенного Modbus (Master) устройства. Модуль запрашивает устройство, и, соответственно, тут же меняется значение: показывается значение адреса последнего запроса.

Last Error – код ошибки. В переменной отображается код ошибки, если информационный обмен прошёл неудачно. Это необходимо для корректности работы опрашиваемого устройства.

Кроме того, в состав модуля Modbus (Master) фиксировано вставлен один из подмодулей RS-485 или Debug RS-232, задающий параметры обмена по соответствующему коммуникационному интерфейсу. На рис. 31 представлена вкладка параметров последовательного порта.

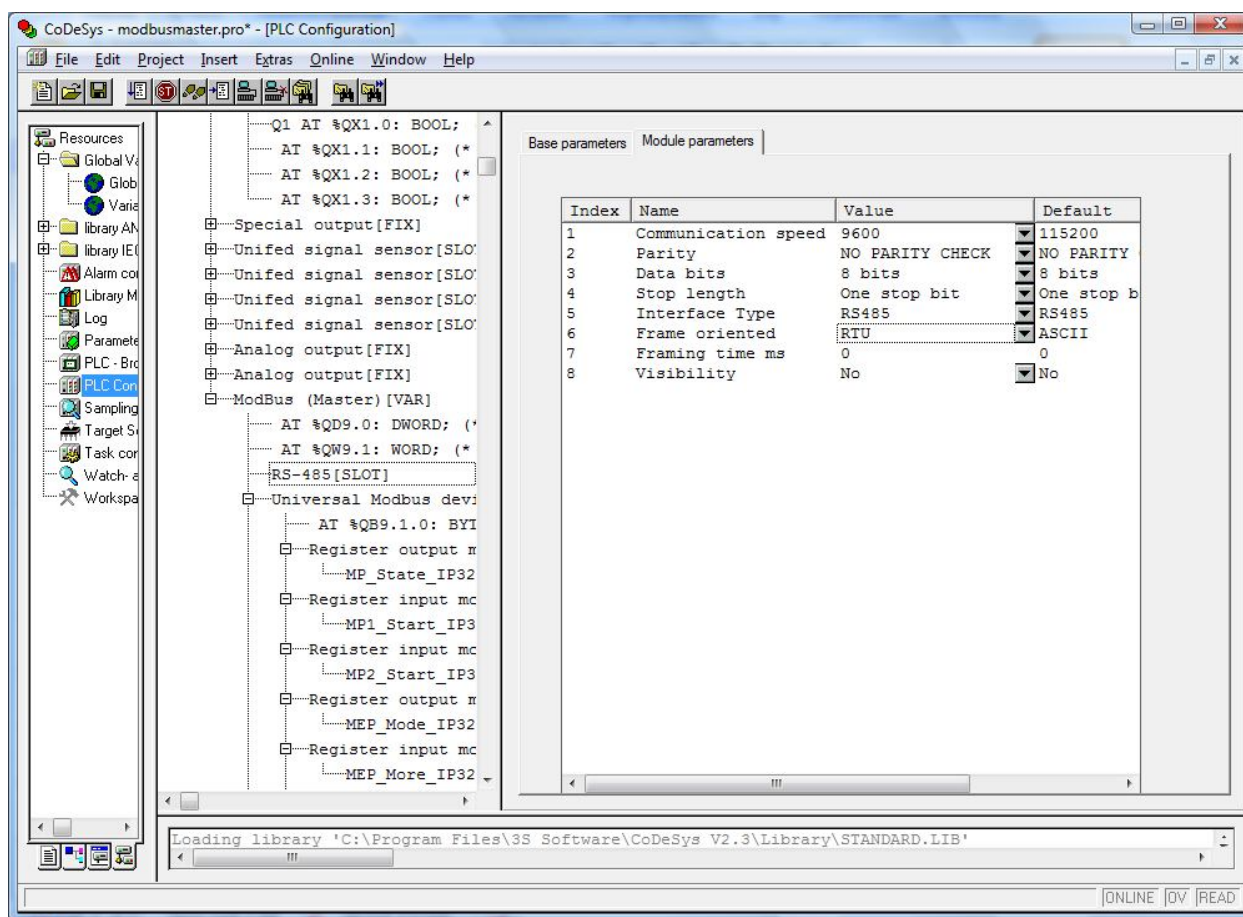


Рис. 31. Окно задания параметров порта.

Параметры последовательного порта:

«Скорость передачи информации» (*Communication speed, bod*) – значения выбираются из списка (2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200), значение по умолчанию – 115200.

«Проверка четности» (*Parity*) – значения выбираются из списка, значение по умолчанию – NO PARITY CHECK (отсутствие проверки четности).

«Количество бит данных» (*Data Bits*) – значения в диапазоне от 5 до 8 б, значение по умолчанию – 8.

«Количество стоп-битов» (*Stop Length*) – значения выбираются из списка (один, полтора или два стоп-бита), значение по умолчанию – один стоп-бит (Onestopbit).

«Тип интерфейса» (*Interface Type*) – «RS-232» или «RS-485» (параметр неизменен).

«Тип протокола обмена» (*Frame Oriented*) – значения выбираются из списка «ASCII» и «RTU-режим», значение по умолчанию – ASCII.

В ПЛК используются следующие типы протоколов обмена: ориентированный на передачу текстовых символов режим ASCII и ориентированный на передачу потока байтов (RTU-режим).

В ASCII-режиме информация передается последовательностью символов, и начало и окончание посылки имеют четко обозначенные специальные символы, обычно это – символы решетки, перевода строки и др.

В RTU-режиме иная структура передачи информации: передаются байты, несущие полезную информацию, без какого-либо указания начальных и/или конечных границ (заголовочных и конечных байтов). Сама посылка и ее границы определяются по наличию разрыва. Если время разрыва превышает определенное время (например, для Modbus – время передачи 3,5 символов), – устройство определяет, что посылка закончилась, началась другая посылка. Таким образом, посылки отделяются друг от друга и их можно идентифицировать.

«Время, на которое необходимо задерживать ответ на запрос в мс» (*Framingtime*) – рекомендуемый диапазон значений от 0 до 50 мс, значение по умолчанию – 0. Задержка бывает необходима для работы с устройствами с низкими скоростями информационного обмена.

«Видимость» (*Visibility*) – задает видимость параметров модуля в программе EasyWorkPLC. Значения выбираются из списка «yes» и «no», значение по умолчанию – «no».

Кроме RS-232, RS-485 предусмотрена также возможность обмена данными по интерфейсу Modbus-TCP (через Ethernet). В этом случае необходимая настройка производится в подмодуле устройства (подмодуль Universal Modbus Device).

#### *Универсальное устройство Modbus (Universal Modbus Device)*

Установка и настройка устройств, опрашиваемых Мастером, в общем случае производятся через установку подмодуля и задание необходимых параметров этому подмодулю. Его подключение выполняется командой «Append Universal Modbus Device», вызываемой из контекстного меню.

Окно настройки модуля «Универсальное устройство Modbus (Master)» представлено на рис. 32.

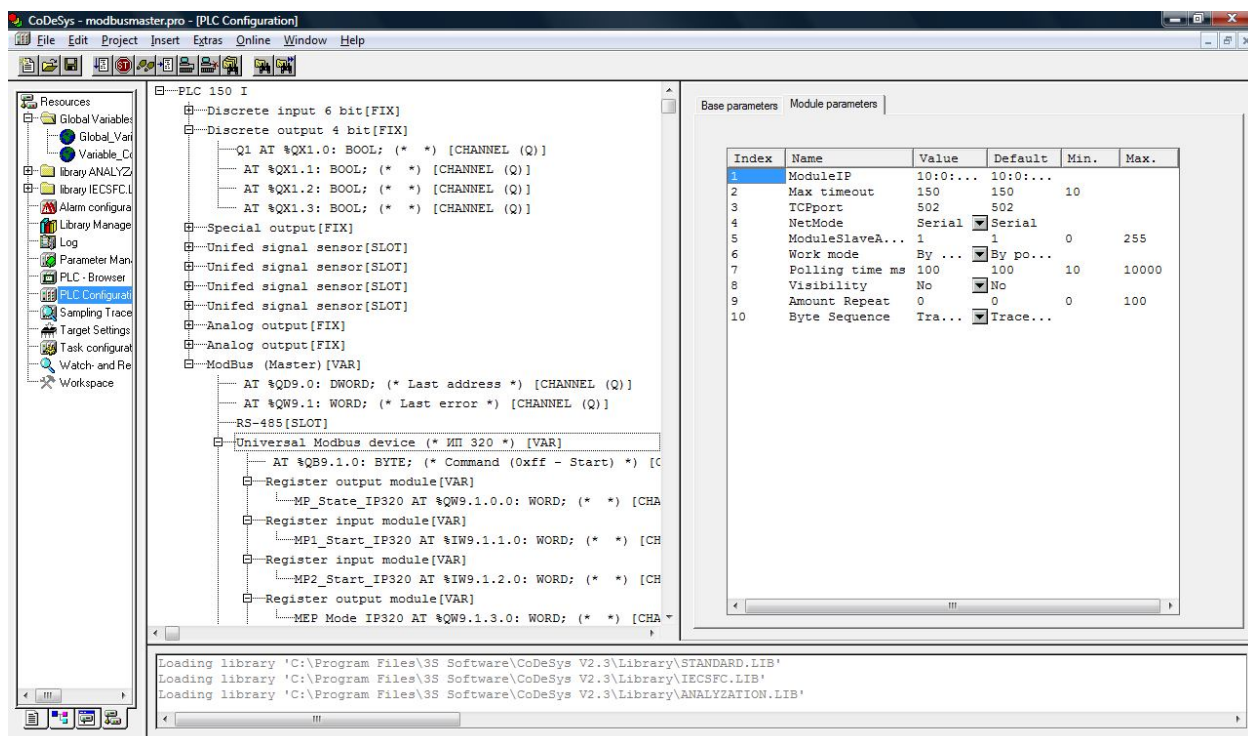


Рис. 32. Окно конфигурирования модуля «Универсальное устройство Modbus (Master)».

При вызове модуля универсального устройства появляется незаполненный шаблон, параметры и идентификаторы которого не привязаны к конкретному внешнему устройству (модулю ввода-вывода, операторской панели). При задании значений параметров модуля конфигурируется конкретный вариант внешнего устройства.

Универсальное устройство Modbus имеет канал Start/Stop. Если в него записывается значение 0x00FF, то происходит старт работы данного устройства Modbus, если же в канал записано значение 0x00FE, то происходит его остановка и прекращение всех посылок в сеть. Если модуль уже запущен, то повторная запись в канал значения 0x00FF приводит к внеочередному запросу одной очередной переменной устройства Modbus. **По умолчанию модуль запущен.**

Параметры универсального устройства Modbus:

«IP-адрес» (*Module IP*)– IP-адрес подчиненного устройства, которым управляет Мастер, если оно будет работать в режиме TCP. Если обмен осуществляется по последовательному интерфейсу (RS-232 или RS-485) устанавливать этот параметр не имеет смысла;

«Максимальный тайм-аут, в мс» (*Max timeout*)– максимальное время, в течение которого устройство должно ответить на запрос. Если по истечении этого времени Мастер не получил ответ на запрос, то это значит, что произошел сбой или авария. Информация о сбое фиксируется в переменной модуля Lasterrog. Мастер продолжает опрос других устройств. Значение сверху не ограничено, может быть любым, в т.ч. дробным, но не меньше 10 мс, значение по умолчанию – 150;

«Порт TCP» (*TCP port*)– значение по умолчанию – 502. Значение параметра может быть установлено и другое, при необходимости. Если обмен осу-

ществляется по последовательному интерфейсу (RS-232 или RS-485) устанавливать этот параметр не имеет смысла;

«Режим работы в сети» (*NetMode*) – значения выбираются из списка «TCP» и «Serial». При задании режима Serial обмен ведется через тот последовательный интерфейс, который подключен к модулю Мастера. В случае задания режима TCP используется интерфейс Ethernet. В этом случае опрашиваемое внешнее устройство идентифицируется по IP-адресу;

«Адрес подчиненного устройства» (*Module Slave Address*)– диапазон значений от 1 до 247, значение по умолчанию – 1. Значение 0 специфично и используется для широковещательных сообщений. Например, при работе через шлюз;

«Режим работы» (*Work mode*) – значения выбираются из списка:

Polling time – «по времени» – контролируемые устройства опрашиваются с периодичностью, заданной в параметре «Период опроса устройства» (*Pollingtime*);

Value change – «по изменению значения переменных» – модуль Owen (Master) генерирует запрос устройству при изменении значений выходных переменных модуля;

Both – «оба варианта» – опрос производится с временным интервалом, заданным в параметре Polling time и тогда, когда изменяются значения выходных переменных;

By Command – «по команде» – производится однократная посылка запроса, когда в командный канал Command переменной записывается значение 0x00FF.

Для переменных с командным каналом при работе в режиме «По команде» (*By Command*) управление осуществляется следующим образом: первая посылка значения 0x00FF в командный канал включает функционирование этой переменной, повторная посылка значения 0x00FF инициирует проведение опроса. Аналогично опрос инициируется для переменных с командным каналом при работе в других режимах. При посылке в командный канал значения 0x00FE переменная выключается из цикла опроса мастера;

«Число повторов» (*Amount Repeat*) – определяют число повторов чтения/записи переменных при неудачном сеансе связи. В режиме «По времени» (*Polling time*) значение этого параметра не используется. Рекомендуемый диапазон значений от 0 до 5, значение по умолчанию – 0;

«Порядок передачи байтов посылки» (*Byte Sequence*) определяет, в каком порядке будут передаваться байты посылки протокола Modbus для переменных длиной 32 бита. У устройств разных производителей этот порядок разный, он не стандартизирован в самом протоколе, поэтому необходимо обговаривать его для конкретных устройств. Значения выбираются из списка «Порядок байтов, используемый в ПЛК» (*Native*) и «Порядок байтов, используемый в программе Trace Mode» (*Trace mode*), значение по умолчанию – «Native».

«Период опроса устройства, в мс» (*Polling time*)– диапазон значений от 10 до 10000, значение по умолчанию – 100. В Мастере, когда он работает в ре-

жиме «По изменению значения переменных» или «По команде», нельзя ставить значение параметра Polling time слишком маленьким. По умолчанию его значение 100 мс и в этих режимах оно не влияет на периодичность посылки запросов мастера. Однако, если на реальном проекте будет замечено, что Мастер при загрузке программы или при Login формирует лишние пакеты и/или запросы, которых не должно быть, значение параметра увеличивают (до 200, 300 и т.д.) до предотвращения появления ложных пакетов;

«Видимость» (*Visibility*)– задает видимость параметров модуля в программе EasyWorkPLC. Значения выбираются из списка «yes» и «no», значение по умолчанию – «no».

После задания значений параметров модуля Universal Modbus Device к нему требуется подключить каналы, задающие переменные ввода/вывода модуля. Могут быть подключены следующие типы каналов (переменных): REAL, STRING, 4 байта, 2 байта или 8 бит. Кроме того, реализована возможность работы по команде: вставляются переменные с обозначением «Comm», содержащие дополнительный управляющий канал.

Параметры каналов ввода/вывода модуля Modbus (Master):

«Номер команды протокола» (*Command*)– значения выбираются из списка возможных вариантов, значение по умолчанию – «presetsingleregister (0x06)» для выходной строковой переменной («write multiple registers (0x10)» для 2-байтной переменной и др.);

«Адрес (номер) регистра опрашиваемого устройства» (Start Address) – устанавливают адрес регистра опрашиваемого устройства;

«Размер переменной» (*Amount byte*) – для переменных типа String, диапазон значений от 1 до 80, значение по умолчанию – 80.

### *Шаблоны устройств*

В Modbus (Master) реализована возможность вставлять шаблоны с наиболее часто применяемых устройств – МВА, МБУ и МДВВ. При вставлении шаблона пользователь получает настроенные каналы выходов и выходов с настроенными номерами регистров, командами и пр., и ему остается в универсальном устройстве настроить тип сервера, адрес устройства, выбрать режим работы и задать период опроса.

Особенности CoDeSys не допускают удаление настроенного в шаблоне канала по команде «Delete». По этому, если пользователю не нужны какие-то каналы и для понижения расхода памяти области ввода-вывода, канал заменяется на специальный модуль Empty. При этом определенный параметр фактически исключается из опроса.

Шаблон может быть дополнен дополнительными каналами, добавление каналов производится аналогично добавлению каналов в Универсальное устройство Modbus.

### *Программа работы*

#### *1. Конфигурирование панели оператора ИП320*

1.1. Сделайте копию проекта, созданного при выполнении лабораторной работы №1. Запустите программу «Конфигуратор ИП320» и откройте проект (копию). Установите режим работы панели – Slave (меню Файл/Выбор ПЛК).

1.2. Замените на всех экранах проекта номера регистров Modbus внешних устройств (МВА8 и МВУ8), связанных с элементами экранов («Динамический текст», «Кнопка», «Регистр» и т.д.), номерами внутренних регистров панели ИП320. Номера можно назначать по порядку. Зафиксируйте (выпишите) назначения всех регистров ИП320.

1.3. На экранах «Управление преобразователем частоты» и «Управление тиристорным регулятором» измените настройки элементов экрана, отвечающих за ввод заданий по частоте ПЧ и мощности тиристорного регулятора таким образом, чтобы частота вводилась в Гц (0-50 Гц с дискретностью 0,1 Гц), а мощность – в % (0-100% с дискретностью 0,1%). Возможному изменению подлежат значения полей ввода «Предел Мах», «Разряды» и «Дробные».

## 2. Конфигурирование и программирование ПЛК150

2.1. Запустите программу CoDeSys и создайте новый проект для контроллера PLC150.I-M. Выберите язык программирования ST для написания главной программы PLC\_PRG.

2.2. Откройте вкладку Resources в левой части окна программы и окно конфигурирования контроллера PLCConfiguration.

2.3. Добавьте в конфигурацию контроллера модуль Modbus Master. Настройте подмодуль RS-485[SLOT], введя параметры сетевого обмена (скорость обмена, контроль четности и т.д.), равные таковым в панели ИП320.

Если в модуле Modbus Master первоначально был установлен подмодуль DebugRS-232[SLOT], замените его на подмодуль RS-485[SLOT], выбрав «Replacement» в контекстном меню.

2.4. Добавьте в модуль Modbus Master подмодуль UniversalModbusDevice для обмена данными с панелью ИП320. В базовых параметрах подмодуля желательно ввести комментарий, например «ИП320».

Создайте в подмодуле каналы обмена с регистрами Modbus панели ИП320 путем добавления элементов («подэлементов»):

*RegisterInputmodule* – для входных каналов контроллера (т.е. для тех переменных, которые будут считываться контроллером из панели);

*RegisterOutputmodule* – для выходных каналов контроллера (т.е. для тех переменных, которые будут записываться контроллером в панель).

Настройте все каналы (модули) введя номера регистров, функции Modbus для чтения и записи регистров и имена переменных.

Номера регистров были зафиксированы при выполнении пункта 1.2.

Функции Modbus:

*Read holding Registers (0x03)* – для Register Input module;

*Preset single register (0x06)* – для Register Output module.

Имена переменных следует задавать информативными. Например, выходная переменная, в которую будет записываться информация о состоянии магнитных пускателей с целью отображения ее на панели ИП320, может быть на-



звана `MP_State_IP320`, а входная переменная, в которую попадает сигнал управления магнитным пускателем, сформированный панелью, – `MP1_Start_IP320`.

2.5. Добавьте в модуль Modbus Master подмодуль Universal Modbus Device для чтения данных из модуля ввода МВА8. В базовых параметрах подмодуля желательно ввести комментарий, например «МВА8».

Создайте в подмодуле каналы обмена с регистрами Modbus модуля ввода МВА8 путем добавления элементов («подэлементов») Register Input module.

Настройте все каналы, введя номера регистров, функции Modbus для чтения регистров и имена переменных.

Номера регистров МВА8 см. в табл. 9. Информацию о значениях переменных следует извлекать из регистров «Целое значение измерения входа  $i$  со смещением точки», где  $I$  – номер канала. Для первого канала  $i = 1$ , для второго  $i = 7$  и т.д.

Для чтения всех перечисленных выше регистров применяется функция Modbus Read holding Registers (0x03).

Имена переменных следует задавать информативными. Например, входная переменная, из которой будет считываться информация о состоянии магнитных пускателей с целью пересылки ее в панель ИП320, может быть названа `MP_State_MVA8`.

2.6. Добавьте в модуль Modbus Master подмодуль Universal Modbus Device для передачи данных в модуль вывода МВУ8. В базовых параметрах подмодуля желательно ввести комментарий, например «МВУ8».

Создайте в подмодуле каналы обмена с регистрами Modbus модуля вывода МВУ8 путем добавления элементов («подэлементов») Register Output module.

Настройте все каналы, введя номера регистров, функции Modbus для записи регистров и имена переменных.

Номера регистров МВУ8 см. в табл. 12.

Для записи во все регистры модуля используется функция Modbus *Write-multiple registers* (0x10).

Имена переменных следует задавать информативными. Например, выходная переменная, в которую будет записываться сигнал управления магнитным пускателем 1, может быть названа `MP1_Start_MVU8`.

2.7. Введите пользовательскую программу в ПЛК.

Программа контроллера будет выполнять функцию связи между модулями ввода-вывода МВА8 и МВУ8 и панелью оператора ИП320: считывать сигналы о состоянии объекта (стенда) из МВА8 и заносить ее в ИП320 для отображения, считывать сигналы управления, сформированные в ИП 320 и подавать их в модуль вывода МВУ8 для реализации.

Пример (с использованием принятых выше обозначений):

```
(*Отображение состояния магнитных пускателей на ИП320*)
MP_State_IP320:=MP_State_MVA8;
(*Управление магнитным пускателем 1*)
MP1_Start_MVU8:=MP1_Start_IP320;
```

Сформируйте остальные строки кода самостоятельно.

При формировании некоторых каналов связи требуется осуществлять простейшие преобразования сигналов, в частности масштабирование. В нашем случае необходимо масштабировать сигналы управления преобразователем частоты, формируемые с панели и по месту (на стенде) для того, чтобы они вводились и отображались на экране панели в Гц.

### *3. Запуск системы*

3.1. Подайте питание на ИП320 и загрузите разработанный в п.1 проект в панель.

3.2. Подайте питание на все цепи лабораторного стенда. Конфигурирование модулей ввода-вывода МВА8 и МВУ8 было проведено ранее при выполнении лабораторной работы №1. Если модули были переконфигурированы, необходимо вернуться к требуемой конфигурации.

3.3. Подайте питание на контроллер, загрузите программу и запустите ее на выполнение.

3.4. Убедитесь в правильности работы всех экранов ИП320 и продемонстрируйте работу системы преподавателю.

### *4. Апробирование шаблонов МВА8, МВУ8 (дополнительное задание)*

В модуле Modbus Master конфигурации ПЛК вместо подмодулей Universal Modbus Device для обмена данными с модулями ввода-вывода используйте предоставляемые шаблоны устройств.

Внесите необходимые изменения в программу контроллера, связанные с изменением формата выходных данных МВА8 (шаблон использует Float32 вместо int16 для передачи значений входных переменных).

Запустите проект, убедитесь в правильности работы экранов и продемонстрируйте работу системы преподавателю.

### *Содержание отчета*

1. Экраны проекта ИП320 с указанием настроек всех элементов.
2. Конфигурация и программа ПЛК.

### *Контрольные вопросы*

1. Номенклатура входов и выходов ПЛК150.
2. Интерфейсы связи ПЛК150.
3. Опишите процедуру конфигурирования ПЛК150.
4. Какие бывают виды модулей конфигурации ПЛК150?
5. Назначение, каналы и параметры модуля Modbus (Master).
6. Назначение, каналы и параметры подмодуля «Универсальное устройство Modbus (Master)».
7. Какие физические каналы связи может использовать подмодуль «Универсальное устройство Modbus (Master)»?
8. Шаблоны каких устройств могут использоваться как подмодули Modbus (Master)?
9. Опишите взаимодействие сетевых устройств в разработанной системе.

*Лабораторная работа №3. Знакомство с панелью оператора Weintek MT8070iE. Off-line симуляция*

Цели работы:

знакомство с функциональными возможностями панели оператора Weintek MT8070iE;

апробация основных графических объектов панели.

***Теоретические сведения***

***1. Панель оператора Weintek MT8070iE***

MT8070iE – сенсорная графическая операторская панель (панель оператора) для систем автоматизации повышенной производительности в ультратонком корпусе. Панель имеет высокую скорость передачи данных и отрисовки объектов интерактивного графического дисплея и двойную изоляцию по RS485. Поддерживаются протоколы MODBUS TCP/IP, BacNET/IP и MPI/PPi соединения с множеством устройств. Печатная плата панели обработана специальным образом для защиты от воздействия влаги и пыли. Конфигурирование операторской панели (панели оператора) MT8070iE осуществляется с помощью бесплатного программного обеспечения EasyBuilderPro [9].

Основные характеристики панели MT8070iE:

дисплей: 7" TFT LCD, широкоэкранный;

цветность: 16М цветов максимально;

разрешение: 800 x 480;

поверхностная яркость: (cd/m<sup>2</sup>): 350;

контрастность: 500:1;

светодиодная подсветка матрицы;

последовательный интерфейс: COM1: RS-232/ RS-485 2w/4w, COM3: RS-485 2w;

1 USB 2.0 (Host);

1 Ethernet порт (10/ 100 Base-T);

процессор: 32 бит RISC CPU / 600 МГц;

память: 128 Mb RAM;

встроенные часы реального времени;

питание : +24В, 350 мА;

защита лицевой панели: NEMA4 / IP65;

рабочая температура: 0 ... +50°C.

***2. Конфигурирование панели в EasyBuilderPro***

Основные интерфейсы программы EasyBuilderPro просты и интуитивно понятны. Упрощенно последовательность действий при работе над проектом показана на рис. 33.



Рис.33. Этапы работы с программой

Создание проекта в общем случае включает:

1) выбор модели панели и ориентации экрана:

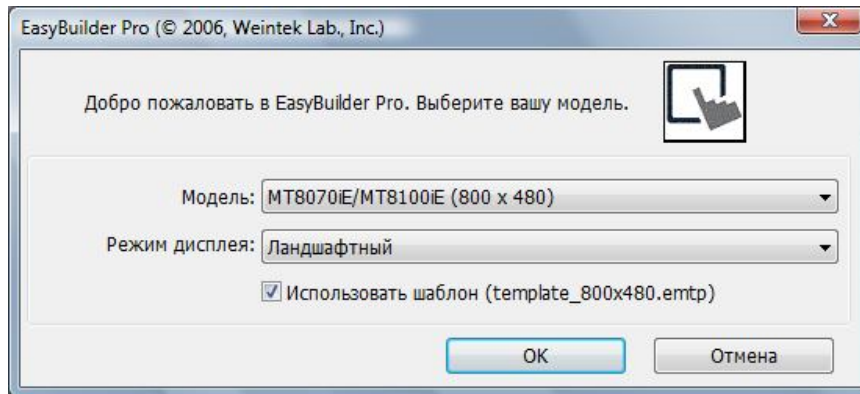


Рис.34. Окно выбора модели

2) настройка системных параметров:

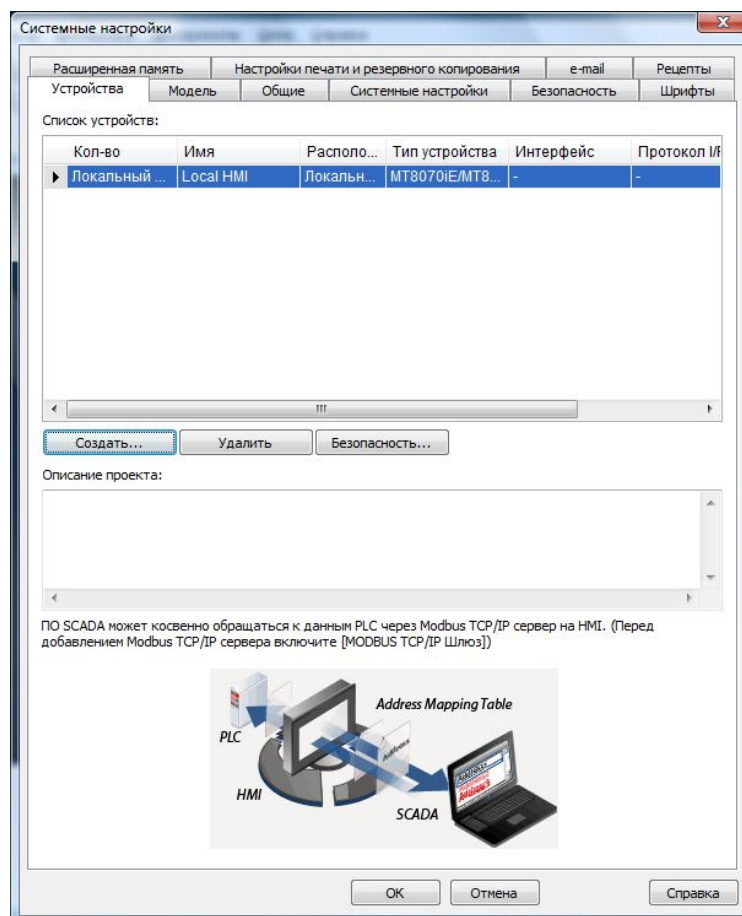


Рис.35. Настройка системных параметров.

Здесь регистрируются все устройства, которые будут участвовать в обмене данными (в том числе и сама панель), интерфейсы связи и устанавливаются многочисленные настройки панели. Повторно вызвать это окно можно из меню «Правка»;

3) формирование экранов панели (заполнение их графическими элементами и их настройка). В EasyBuilderPro могут быть использованы до 1997 окон, пронумерованных от 3 до 1999. Существуют 4 типа окон, отличающихся по функциональности и способу использования:

Base Window (Основное окно);

Common Window (Общее окно);

Fast Selection Window (Окно быстрого выбора);

System Message Window (Окно системных сообщений).

*Основное окно* применяется:

в качестве фона (background) для других окон;

как окно с клавиатурой;

всплывающее окно для объекта типа «функциональная кнопка» (function key);

всплывающее окно для объектов [Directwindow] и [Indirectwindow];

заставка (screen saver).

*Окно быстрого вызова.* Окно 3 определено, как окно быстрого выбора. Это окно может использоваться совместно с основным окном. Как правило, оно содержит часто вызываемые команды. Кнопка быстрого вызова (Fast Selection Button), используется для активации/деактивации функции быстрого вызова. Все настройки кнопки быстрого вызова прописаны в Системных настройках (вкладка «Общие»). Кроме настройки атрибутов быстрого вызова с помощью кнопки быстрого вызова, в системных регистрах выделено несколько адресов:

[LB9013] Управление окном быстрого вызова (FS) - [Показать (ON)/Спрятать (OFF)];

[LB9014] Управление кнопкой быстрого вызова [Спрятать (ON)/Показать (OFF)];

[LB9015] Управление окном / кнопкой быстрого вызова [Спрятать (ON)/Показать (OFF)].

*Общее окно.* Окно 4 по умолчанию является общим окном. Объекты этого окна отображаются во всех других окнах, кроме всплывающих окон (popup windows). Поэтому, объекты совместного использования всегда размещают в общем окне. Например: логотип изделия или общие кнопки. Имеется возможность изменить общее окно [9].

*Окно системных сообщений.* По умолчанию окнами системных сообщений назначены окна 5, 6, 7 и 8. Например, WindowNo. 5: PLCResponse– «Ответ контроллера». При прерывании соединения между ПЛК и панелью оператора это окно сообщений автоматически появляется справа на ранее открытом основном окне.

Графические объекты рассмотрим при выполнении программы работы.

*Сохранение и компиляция проекта* не должны вызывать затруднений.

*Имитация работы* панели на компьютере может производиться как в Off-line так и On-line в режиме. В режиме Off-line программа-симулятор работает автономно, без соединения с внешними устройствами, в On-line соединение с ПЛК устанавливается доступными способами, например, через локальную сеть. Таким образом, если все внешние устройства каким-либо образом связаны с компьютером, на котором запущен On-line симулятор, последний становится полнофункциональной заменой панели. Поэтому разработчики ограничили время работы On-line симулятора 10 минутами.

*При загрузке конфигурации в панель* требуется указать параметры соединения с панелью. Если панель связана с компьютером по сети, достаточно указать лишь IP-адрес панели.

### ***Программа работы***

1. Запустите программу EasyBuilderPro, создайте новый проект для панели MT8070iE. Системные настройки не изменяйте.

#### ***2. Общее окно.***

Разработайте общее окно (№4), поместив в него все, что Вы хотели бы видеть на всех экранах, например, какой-либо текст с информацией о системе в целом или о разработчике. Наблюдайте результат в Off-line симуляторе.

#### ***3. Окно быстрого вызова и битовые операции.***

Разработайте окно быстрого вызова (№3), поместив в него битовую кнопку, тумблер и две лампы. Кнопка должна управлять первой лампой: по первому нажатию лампа должна менять свет с зеленого на красный и мигать, по второму - вновь становиться зеленой и прекращать мигание. Тумблер должен изменять цвет второй лампы с черного на зеленый. Для управления лампами задействуйте биты локальной памяти LW0 и LW1. Наблюдайте результат в Off-line симуляторе.

#### ***4. Ввод и отображение чисел.***

На основном окне разместите ползунок и объект «Числа». Оба объекта «привяжите» к локальному слову LW0. При изменении положения ползунка выводимое значение должно изменяться от 0 до 100. Такие же пределы ввода установите для объекта «Числа». Наблюдайте результат в Off-line симуляторе.

#### ***5. Тренд.***

Создайте объект «Выборки данных» (с помощью одной из кнопок нижнего ряда панели инструментов). Объект должен считывать одно слово по адресу LW0. На основном окне разместите «Графические выборки», настройте его на созданный объект «Выборки данных», задайте пределы отображения от 0 до 100. Наблюдайте результат в Off-line симуляторе.

#### ***6. Отображение состояний.***

На основном окне разместите объект «Байтовая кнопка». Настройте кнопку на циклическое изменение («Циклический JOG+») слова LW1 от 0 до двух с единичным шагом.

На основное окно поместите прямоугольник, сделайте его зеленым с белым контуром. Выделите прямоугольник (с помощью «стрелки») и нажмите на пиктограмму «Добавить фигуры в библиотеку» (верхний ряд панели инструментов). В появившемся окне создайте новую библиотеку. Появится отображение новой пустой библиотеки. Раскройте окно «Менеджер фигур» полностью с помощью кнопки «Дополнительно» и сохраните прямоугольник два раза: сначала «Кадр» и «Вставить», потом «Внутренний» и «Заменить». Прямоугольник должен появиться в среднем фрагменте окна с атрибутами «Заливка», «Контур» и состоянием 0.

Измените цвет прямоугольника на основном окне на зеленый. Добавьте этот прямоугольник в ту же библиотеку и в тот же объект с помощью описанной выше процедуры. Вновь измените цвет прямоугольника, теперь уже на красный, и вновь добавьте его в библиотеку.

В результате создана новая библиотечная фигура с тремя состояниями. Удалите прямоугольник с основного окна.

Поместите на основное окно «Байтовый индикатор». Установите для него адрес чтения LW1 и число состояний 3. Во вкладке «Фигура» из библиотеки фигур выберите созданную Вами фигуру. Во вкладке «Метка» установите для состояния 0 метку «Иди», для состояния 1 – «Внимание» и для состояния 3 – «Стой».

В результате выполненных в данном пункте действий по нажатию на «Байтовую кнопку» «Байтовый индикатор» должен циклически менять цвет и надпись. Наблюдайте результат в Off-linesимуляторе.

#### *7. Косвенный вызов окна.*

Создайте новое окно (№11) небольших размеров, например, 250×100. Убедитесь, что режим использования окна не монопольный. Разместите на нем произвольный текст.

Поместите на основное окно №10 объект «Косвенный вызов окна». В настройках объекта установите адрес чтения LW2.

Пометите на основное окно №10 два объекта «Байтовая кнопка». В настройках обоих объектов установите адрес записи LW2, стиль «Записать постоянное значение». Первая кнопка должна записывать в LW2 число 11, вторая – 0.

Наблюдайте результат в Off-linesимуляторе.

#### *8. Прямой вызов окна.*

Создайте новое окно (№12) небольших размеров. Убедитесь, что режим использования окна не монопольный. Разместите на нем произвольный текст.

Поместите на основное окно №10 объект «Прямой вызов окна». В настройках объекта установите адрес чтения LB2и атрибут «№ окна» «12. Окно\_12».

Пометите на основное окно №10 объект «Битовая кнопка», установите адрес записи LB2 и стиль «Переключить».

Наблюдайте результат в Off-linesимуляторе.

#### *9. Функциональная кнопка. Управление полноэкранными окнами.*

Создайте новое полноэкранное окно (№13). Поместите на него изображение из файла и функциональную кнопку. В настройках кнопки установите «Возврат в предыдущее окно».

Пометите на основное окно №10 функциональную кнопку с вызовом окна №13.

Наблюдайте результат в Off-line симуляторе.

10. Оформите визуализацию, сгруппировав элементы и введя поясняющие надписи, см. рис.36-38.

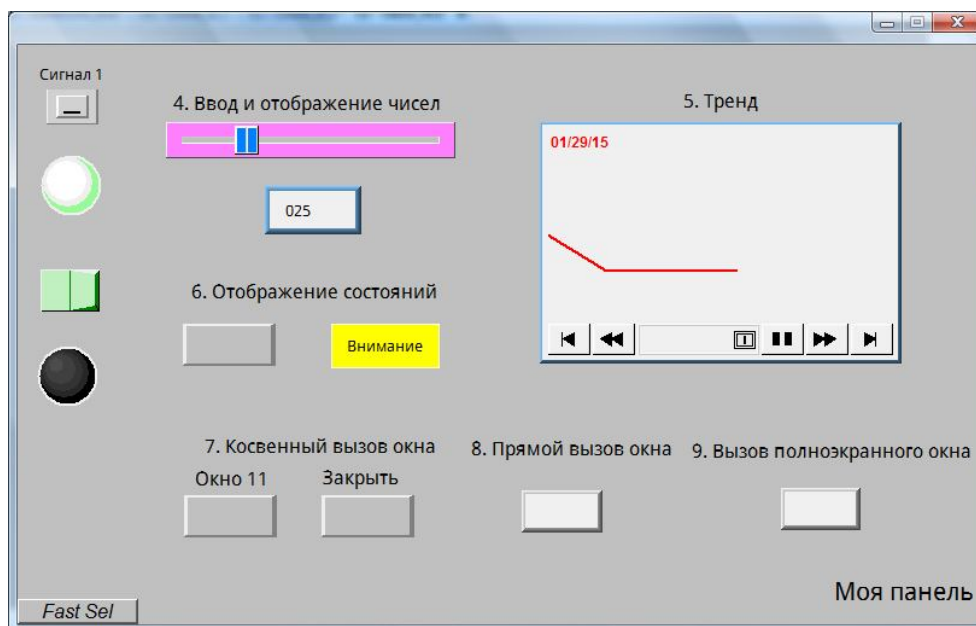


Рис. 36. Основное окно (с окном FastSelection).

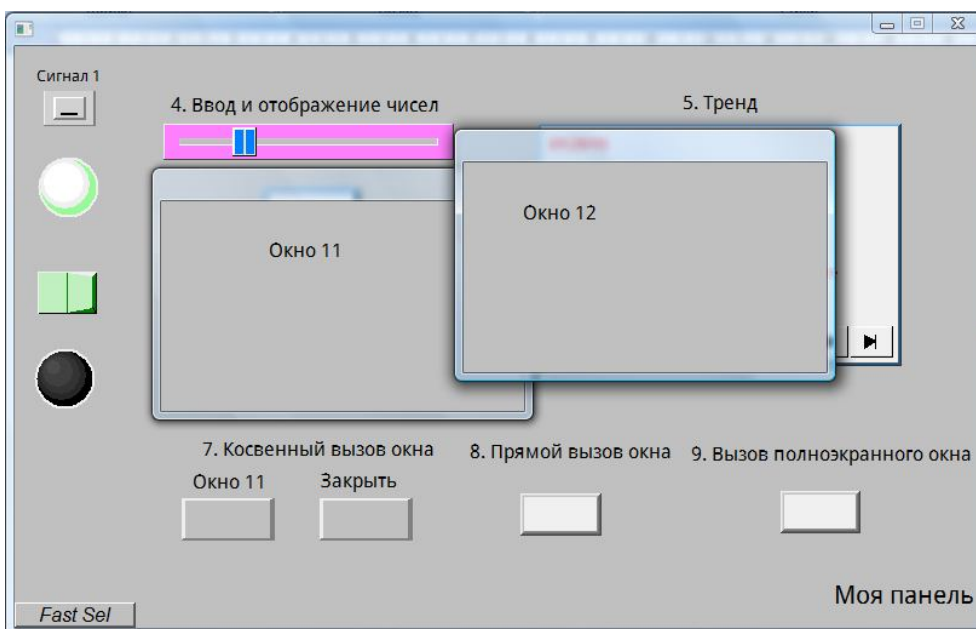


Рис. 37. Вызов окон 11, 12.





Рис. 38. Окно 13.

11. Загрузите проект в панель и опробуйте все элементы визуализации.

### ***Содержание отчета***

Экраны проекта EasyBuilderPro с указанием настроек всех элементов.

### ***Контрольные вопросы***

1. Какие последовательные и сетевые интерфейсы имеет панель оператора МТ8070iЕ?
2. Опишите общий порядок работы (последовательность действий) в программе EasyBuilderPro.
3. Какие типы окон можно создать в EasyBuilderPro?
4. Какие инструменты имитации работы панели имеются в EasyBuilderPro?
4. Опишите объекты управления и отображения битов, используемые в работе.
5. Опишите объекты ввода и отображения чисел, используемые в работе.
6. Опишите объекты отображения трендов, используемые в работе.
7. Опишите объекты отображения и изменения состояний, используемые в работе.
8. Опишите способы вызова окон, примененные в работе.

*Лабораторная работа №4. Имитационное моделирование системы автоматического регулирования. On-line симуляция Weintek MT8070iE*

Цели работы:

получение навыков разработки имитационных моделей систем автоматического регулирования с человеко-машинным интерфейсом;  
апробация системы на реальной панели оператора.

***Программа работы***

В Приложении 2 рассмотрена имитационная система, включающая модель объекта управления, симулятор ПЛК для персонального компьютера под управлением соответствующего программного обеспечения и программа On-Line имитации панели оператора.

Требуется:

1. Изучить материал, приведенный в Приложении 2, и воссоздать имитационную систему;  
запустить OPC-сервер;  
построить Simulink-модель объекта управления;  
запрограммировать контроллер;  
сконфигурировать панель;  
настроить межпрограммный обмен и запустить процесс симуляции.
2. Загрузить проект EasyBuilderPro в панель и опробовать работу системы с реальной панелью.

***Содержание отчета***

1. Simulink-модель объекта управления.
2. Управляющая программа для PLC WinNT.
3. Экран визуализации панели MT8070iE.

***Контрольные вопросы***

1. Какие задачи могут быть решены с помощью имитационной системы?
2. Состав и назначение программных средств имитационной системы.
3. Опишите схему взаимодействия программных компонентов.
4. Какова последовательность действий по запуску OPC-сервера?
5. Опишите Simulink-модель объекта управления.
6. Опишите программу PLC\_PRG контроллера.
7. Какие системные настройки были сделаны в проекте EasyBuilderPro?
8. Опишите объекты экрана панели.

*Лабораторная работа №5. Система дистанционного управления лабораторным объектом на основе модулей ввода-вывода Овен МВА8, МВУ8, панели оператора Weintek MT8070iE и ПЛК150*

Цели работы:

разработка и реализация системы дистанционного управления лабораторным объектом на базе модулей ввода-вывода Овен МВА8, МВУ8, панели оператора Weintek MT8070iE и ПЛК 150.

***Теоретические сведения***

При выполнении программы работы потребуется сконфигурировать панель как клиента Modbus. В EasyBuilderPro клиент Modbus называется сервером Modbus. Для конфигурирования требуется добавить новое устройство «MODBUS Server» в список устройств окна «Системные настройки». Для параметра PLC I/F – Интерфейс ПЛК может выбрано любое значение: RS232, RS485 2W (2-х проводный), RS485 4W (4-х проводный), Ethernet, USB, CAN.

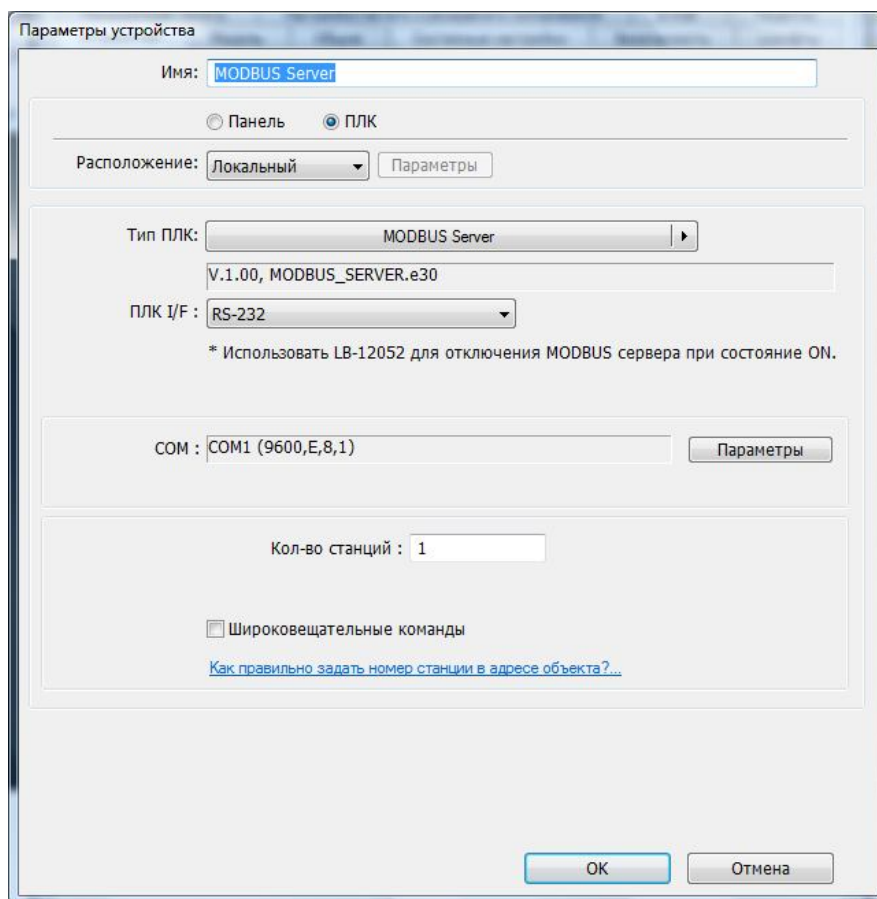


Рис.39. Настройка параметров Modbus сервера.

Если выбраны интерфейсы RS232 или RS485, необходимо задать настройки COM-порта. Если выбран интерфейс Ethernet, то IP – то же, что у панели оператора.

Для MODBUS-сервера и панели оператора используется один и тот же номер порта. Если по каким-либо причинам требуется использовать стандартный номер ModbusTCP/IP 502, нужно изменить номер порта MODBUS-сервера на вкладке «Модель». Если оставить использующийся по умолчанию номер 8000, то все устройства,

обменивающиеся данными с панелью по ModbusTCP/IP, должны быть соответственно сконфигурированы.

После настройки внутренняя память панели переносится на адреса протокола MODBUS стандартным образом.

### **Программа работы**

#### *1. Панель MT8070iE в роли ведущего узла Modbus*

Структура системы в данной конфигурации представлена на рис. 40.

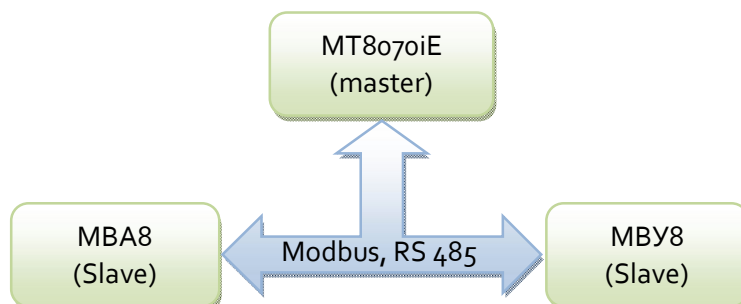


Рис. 40. Структура системы (вариант 1).

Разработайте и реализуйте систему, подобную той, что была создана при выполнении лабораторной работы №1. Визуализация должна включать:

главный экран, на который поместите фотографию объекта управления (лабораторного стенда), элементы отображения основной информации о его состоянии, основные органы управления;

дополнительные экраны с детальной информацией о работе всех подсистем и соответствующими средствами воздействия;

экран «Графики», на котором разместите тренды основных сигналов системы.

#### *2. Панель MT8070iE в роли подчиненного узла Modbus, обмен по RS485*

Структура системы в данной конфигурации представлена на рис. 41.

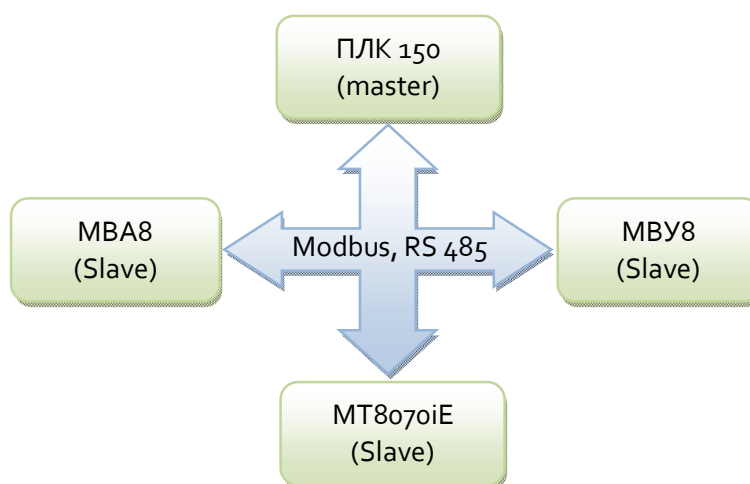


Рис. 41. Структура системы (вариант 2).

Разработайте и реализуйте конфигурацию панели, подобную той, что была создана при выполнении лабораторной работы №2 для ИП 320. Дополните

ее средствами переключения режимов работы подсистем (ручной/автоматический). В ручном режиме воздействие на объект формируется с помощью органов управления панели, в автоматическом – программой ПЛК.

Разработайте программу для ПЛК 150. Для организации сетевого обмена используйте следующие модули и подмодули: Modbus Master, Universal ModBus Device, Register input module (с командой Read holding register 0x03) и Register output module (с командой Preset single register 0x06). Все регистры нумеруются с нуля.

Помимо сетевого обмена, программа должна включать одну или более (по указанию преподавателя) подпрограмм автоматического регулирования температуры воды в баке, вызов которых должен осуществляться при переводе соответствующей подсистемы в автоматический режим.

Возможные варианты:

релейное регулирование с воздействием на нагрев или/и охлаждение (магнитные пускатели);

непрерывное регулирование с воздействием на нагрев (тиристорный регулятор);

квазинепрерывное (импульсное) регулирование с воздействием на нагрев (магнитный пускатель);

непрерывное регулирование с воздействием на охлаждение (преобразователь частоты);

квазинепрерывное (импульсное) регулирование с воздействием на охлаждение (МЭП, управляющий воздушной заслонкой).

При необходимости экспериментально определите математическое описание объекта и настройте регуляторы.

### *3. Обмен по Ethernet*

Структура системы в данной конфигурации представлена на рис. 42.

В системе используются две сети: ModbusRS 485, в которой мастером является ПЛК150, и ModbusTCP/IP (Ethernet), в котором мастером может быть как контроллер, так и панель.

Скорректируйте программу ПЛК и конфигурацию панели, разработанные для предыдущего варианта, и реализуйте один или оба (по указанию преподавателя) подварианта системы.

### ***Содержание отчета***

1. Все варианты программ для ПЛК.
2. Все варианты экранов визуализации панели MT8070iE.

### ***Контрольные вопросы***

1. Опишите экраны визуализации, разработанные по варианту 1.
2. Какие преимущества предоставляет работа панели в режиме Slave (вариант 2)?

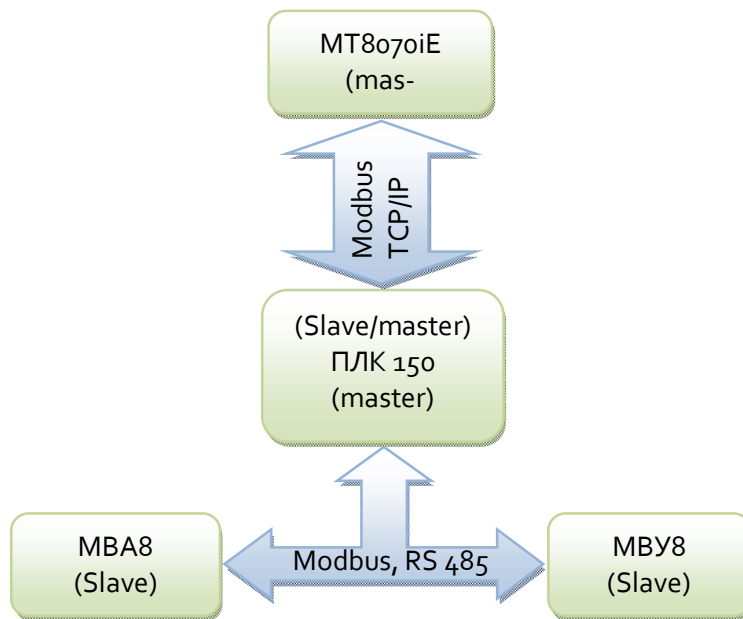


Рис. 42. Структура системы (вариант 3).

3. Опишите программу ПЛК.
4. Каким образом ПЛК конфигурируется как ModbusMaster?
5. Опишите изменения, сделанные в конфигурации операторской панели при переходе ко второму и третьему вариантам системы.

## ПРИЛОЖЕНИЕ 1. Краткое описание протокола Modbus

### **Общие сведения**

Modbus был разработан компанией Modicon (в настоящее время принадлежит Schneider Electric) для использования в её контроллерах с программируемой логикой. Впервые спецификация протокола была опубликована в 1979 году. Это был открытый стандарт, описывающий формат сообщений и способы их передачи в сети, состоящей из различных электронных устройств.

Первоначально контроллеры MODICON использовали последовательный интерфейс RS-232. Позднее стал применяться интерфейс RS-485, так как он обеспечивает более высокую надёжность, позволяет использовать более длинные линии связи и подключать к одной линии несколько устройств.

Многие производители электронного оборудования поддержали стандарт, на рынке появились сотни использующих его изделий.

Основной особенностью протокола является наличие в сети одного ведущего устройства – Master. Только ведущее устройство может опрашивать остальные устройства сети, которые являются ведомыми (Slave). Подчиненное устройство не может самостоятельно инициировать передачу данных или запрашивать какие-либо данные у других устройств, работа сети строится только по принципу «запрос-ответ». Мастер может также выдать широковещательный запрос, адресованный всем устройствам в сети, в таком случае ответное сообщение не посылается.

Существует три типа протокола Modbus: Modbus ASCII, Modbus RTU и Modbus TCP.

Далее рассмотрим элементы Modbus RTU.

### **Структура протокола Modbus RTU**

При использовании режима Modbus RTU сообщение начинается с так называемого интервала тишины, равного времени передачи 3,5 символов, при заданной скорости обмена. Первым полем передается адрес устройства. Вслед за последним передаваемым символом также следует интервал тишины продолжительностью не менее 3,5 символов. Новое сообщение может начинаться после этого интервала. Фрейм сообщения передаётся непрерывно. Если интервал тишины продолжительностью 1,5 символов возник во время передачи фрейма, принимающее устройство должно игнорировать этот фрейм как неполный. Если новое сообщение начнется раньше интервала 3,5 символа, принимающее устройство воспримет его как продолжение предыдущего сообщения. В этом случае устанавливается ошибка CRC (несовпадение контрольной суммы).

Пакет данных в ModBus выглядит следующим образом:

Адрес | Код функции | Данные | Контрольная сумма.

Адрес – это поле, содержащее номер устройства, которому адресован запрос. Каждое устройство в сети должно иметь уникальный адрес. Устройство отвечает только на те запросы, которые поступают по его адресу, во избежание конфликтов. При этом ведомое устройство в своем ответе также посылает поле

Адрес, кроме широковещательного запроса (когда ответа от ведомого быть вообще не должно).

Код функции содержит номер функции Modbus. Функция может запрашивать данные или давать команду на определенные действия. Коды функций являются числами в диапазоне от 1 до 127. Функции с номерами от 128 являются зарезервированными для пересылки в ответном сообщении информации об ошибках.

В поле *Данные* содержится информация, которую передает узел Master узлу Slave, либо наоборот в случае ответного сообщения. Длина этого поля зависит от типа передаваемых данных.

Поле *Контрольная сумма* является важным элементом протокола: в нем содержится информация, необходимая для проверки целостности сообщения и отсутствия ошибок передачи.

Максимальный размер пакета для сетей RS232/RS485 - 256 байт, для сетей TCP - 260 байт.

#### *Методы контроля ошибок*

Стандартная MODBUS сеть использует два метода контроля ошибок: контроль паритета (even/odd) и контрольная сумма. Обе эти проверки генерируются в головном устройстве. Подчиненное устройство проверяет каждый байт и все сообщение в процессе приема.

Пользователь может устанавливать продолжительность интервала таймаута, в течение которого головное устройство будет ожидать ответа от подчиненного. Если подчиненный обнаружил ошибку передачи, то он не формирует ответ главному.

#### *Контроль паритета*

Пользователь может конфигурировать контроллеры на проверку четного или нечетного паритета (Even/Odd).

Например, 8 бит RTU-режима содержат следующую информацию.

1100 0101 Общее количество единиц – 4. Если используется четный паритет, то бит паритета будет равен 0, и общее количество единиц будет по-прежнему четным числом. Если используется нечетный паритет, то бит паритета будет равен 1, тогда общее количество единиц вместе с битом паритета будет равно 5, т.е. нечетному числу.

#### *Контрольная сумма*

В RTU-режиме поле контрольной суммы содержит 16-ти битовую величину. Контрольная сумма является результатом вычисления Cyclic Redundancy Check сделанного над содержанием сообщения. CRC добавляется к сообщению последним полем младшим байтом вперед.

#### **Функции Modbus**

Существует три типа функций.

*Стандартные.* Описание этих функций опубликовано и утверждено Modbus-IDA. Эта категория включает в себя как опубликованные, так и свободные в настоящее время коды.



*Пользовательские.* Два диапазона кодов (от 65 до 72 и от 100 до 110), для которых пользователь может создать произвольную функцию.

*Зарезервированные.* В эту категорию входят коды функций, не являющиеся стандартными, но уже используемые в устройствах, производимых различными компаниями. К этим кодам относятся 9, 10, 13, 14, 41, 42, 90, 91, 125, 126 и 127.

Таблица 18

### Функции Modbus

Код	Название	Действие
01	READ COIL STATUS	Получение текущего состояния (ON/OFF) группы логических ячеек.
02	READ INPUT STATUS	Получение текущего состояния (ON/OFF) группы дискретных входов.
03	READ HOLDING REGISTERS	Получение текущего значения одного или нескольких регистров хранения.
04	READ INPUT REGISTERS	Получение текущего значения одного или нескольких входных регистров.
05	FORCE SINGLE COIL	Изменение логической ячейки в состояние ON или OFF.
06	FORCE SINGLE REGISTER	Запись нового значения в регистр хранения.
07	READ EXCEPTION STATUS	Получение состояния (ON/OFF) восьми внутренних логических ячеек, чье назначение зависит от типа контроллера. Пользователь может использовать эти ячейки по своему выбору.
08	LOOPBACK DIAGNOSTIC TEST	Тестовое сообщение, посылаемое SL для получения данных о связи.
11	FETCH EVENT COUNTER COMMUNICATIONS	Позволяет MS путем последовательной отправки одного сообщения определить выполнение операции.
12	FETCH COMMUNICATIONS EVENT LOG	Позволяет MS получить журнал связи, который содержит информацию о каждой Modbus транзакции данного SL. Если транзакция не выполнена, в журнал заносится информация об ошибке.
13	PROGRAM	Позволяет MS запрограммировать SL.
14	POLL PROGRAM COMPLETE	Позволяет MS связываться с другими SI, если один SL выполняет долговременную операцию программирования. SL периодически опрашивается на момент завершения программирования. Данный запрос посылается только в том случае, если предварительно

Код	Название	Действие
		был послан опрос PROGRAM.
15	FORCE MULTIPLE COILS	Изменить состояние (ON/OFF) нескольких последовательных логических ячеек.
16	FORCE MULTIPLE REGISTERS	Установить новые значения нескольких последовательных регистров.
17	REPORT SLAVE I.D.	Позволяет MS определить тип адресуемого SL и его рабочее состояние.
19	RESET COMMUNICATIONS LINK	Сбрасывает SL в известное состояние после неустранимой ошибки. Сбрасывает счетчик принятых байт.
20-64	Зарезервировано под расширение Modbus	
65-72	Зарезервировано под пользовательские функции	В дальнейшем не будет использоваться в продуктах Modicon.
73-119	ILLEGAL FUNCTION	
120-127	Зарезервировано	Зарезервировано Modicon для внутреннего использования.
128-255	Зарезервировано	Зарезервировано для исключительных ситуаций.

Ниже рассмотрены наиболее часто используемые функции.

*Функция 1: чтение логических ячеек*

Функция позволяет пользователю получить статус (1/0) логических ячеек. Широковещательный режим не поддерживается. Помимо полей адреса SL и функции, сообщение требует, чтобы информационное поле содержало логический адрес первой ячейки и число ячеек, статус которых необходимо получить.

Адресация позволяет получить за один запрос до 2000 логических ячеек. Однако некоторые приборы имеют ограничение на максимальное число ячеек, статус которых можно получить за один запрос. Ячейки нумеруются с нуля (ячейка 1 = 0, ячейка 2 = 1 и т.д.).

Ниже представлен запрос на чтение логических ячеек 0020-0056 из прибора с адресом 17.

Адрес	Функция	Старший байт адреса первой ячейки	Младший байт адреса первой ячейки	Старший байт число ячеек	Младший байт числа ячеек	Контрольная сумма	
11	01	00	13	00	25	B6	LRC

Пример ответного сообщения на запрос:

Адрес	Функция	Количество байт в поле данных	Статус ячеек 20-27	Статус ячеек 28-35	Статус ячеек 36-43	Статус ячеек 44-51	Статус ячеек 52-56	Контрольная сумма
11	01	05	CD	6B	B2	0E	1B	D6

Данные в поле данных упакованы один бит на каждую ячейку. Ответное сообщение включает адрес SL, код функции, число байт в поле данных, данные и контрольную сумму. Младший значащий бит первого байта поля данных содержит первую адресуемую ячейку, за которой следуют остальные. Если число

ячеек не кратно 8, то остальные биты заполняются нулями в порядке от старших битов к младшим.

Статус ячеек 20-27 равен CDH = 1101 1101. Читая слева направо, видим, что ячейки 27, 26, 23, 22 и 20 установлены. Остальные данные разбираются также. Так как было запрошено число ячеек некротное 8, старшие три бита в последнем байте данных (1BH) заполнены нулями.

Так как запрос обслуживается в конце рабочего цикла прибора, то данные в ответном сообщении отражают состояние ячеек на тот момент.

*Функция 2: чтение дискретных входов*

Данная функция позволяет пользователю получить состояние (ВКЛ/ВЫКЛ) входных дискретных линий адресуемого SL. Широковещательный запрос не поддерживается. В дополнение к адресу SL и номеру функции запрос требует, чтобы информационное поле содержало начальный адрес и количество требуемых линий.

Адресация позволяет получить за один запрос до 2000 линий. Однако некоторые устройства имеют ограничение на максимальное количество линий, получаемых за один запрос. Входные линии нумеруются с нуля (10001 = 0, 10002 = 1 и т.д.).

Пример запроса на чтение дискретных входов 10197-10218 из SL с номером 17:

Адрес	Функция	Старший байт номера первой требуемой ячейки	Младший байт номера первой требуемой ячейки	Старший байт количество требуемых ячеек	Младший байт количество требуемых ячеек	Контрольная сумма	
11	02	00	C4	00	16	13	LRC

Ответное сообщение включает адрес SL, код функции, количество байт данных, данные и поле контрольной суммы. Данные упакованы по биту на каждый вход (1 = ON, 0 = OFF). Младший бит первого байта содержит значение первого адресуемого входа, за которым следуют остальные. Если количество запрошенных входов некротное 8, то остальные биты заполняются нулями. Количество байт данных всегда определяется как количество RTU данных.

Так как SL обслуживает запрос в конце рабочего цикла, данные в ответе отражают состояние входов на данный момент. Некоторые устройства имеют ограничение на максимальное количество входов, запрашиваемых за один запрос.

Пример ответного сообщения:

Адрес	Функция	Количество байт данных	Дискретные входы 10197-10204	Дискретные входы 10205-10212	Дискретные входы 10213-10218	Контрольная сумма	
11	02	03	AC	DB	35	2E	LRC

Статус входов 10197-10204 = ACh = 1010 1100. Читая слева направо, видим, что входы 10204, 10202, 10200 и 10199 в состоянии ON. Все остальные байты данных распаковываются аналогично.

Так как было запрошено 22 линии, последний байт данных (35h = 0011 0101) содержит только 6 входов (10213-10218) вместо 8. Два последних бита заполняются нулями.

*Функция 3: чтение регистров*

Данная функция позволяет получить двоичное содержимое 16 разрядных регистров адресуемого SL. Адресация позволяет получить за каждый запрос до 125 регистров. Однако некоторые устройства имеют ограничение на максимальное количество регистров, получаемых за один запрос. Регистры нумеруются с нуля (40001 = 0, 40002 = 1 и т.д.). Широковещательный режим не допускается. Ниже представлен пример запроса на чтение регистров 40108-40110 из SL с адресом 17.

Адрес	Функция	Номер первого регистра		Число регистров для чтения (N)		Контрольная сумма	
		Старший байт	Младший байт	Старший байт	Младший байт		
11	03	00	6B	00	03	7E	LRC

Адресуемый SL посылает в ответе свой адрес, код выполненной функции и информационное поле. Информационное поле содержит 2 байта, описывающих количество возвращаемых байт данных. Длина каждого регистра данных – 2 байта. Первый байт данных в посылке является старшим байтом регистра, второй – младшим.

Так как SL обычно обслуживает запрос в конце своего рабочего цикла, данные в пакете отражают содержимое регистров в данный момент. Некоторые SL ограничивают количество регистров, передаваемых за один запрос. В этом случае для получения большего числа регистров необходимо выполнить несколько последовательных запросов.

Ниже представлен пример ответного сообщения на чтение регистров 40108-40110, имеющих содержимое, соответственно, 555, 0, 100, из SL с адресом 17.

Адрес	Функция	Количество байт данных	Старший байт регистра 40108	Младший байт регистра 40108	Старший байт регистра 40109	Младший байт регистра 40109	Старший байт регистра 40110	Младший байт регистра 40110	Контрольная сумма	
11	03	06	02	2B	00	00	00	64	55	LRC

*Функция 5: запись одной ячейки*

Это сообщение модифицирует одну логическую ячейку. Ячейки нумеруются с нуля (ячейка 1=0, ячейка 2 = 1 и т.д.). Число 65280 (FF00H) устанавливает ячейку в 1, а число 0 – в 0. Другие числа не влияют на содержимое ячейки. Данная функция может использоваться в широковещательном режиме. Ниже приведен пример установки в 1 ячейки 0173 в SL 17.

Адрес	Функция	Старший байт адреса ячейки	Младший байт адреса ячейки	Индикатор установки или сброс ячейки	Всегда 0	Контрольная сумма	
11	05	00	AC	FF	00	3F	LRC

Нормальное ответное сообщение полностью совпадает с запросом.

*Функция 6: запись одного регистра*

Данная функция позволяет модифицировать содержимое одного регистра. Хотя запрос и является асинхронным, SL изменяет содержимое регистра только в конце рабочего цикла.

Когда в запросе указан адрес, равный 0 (широковещательный запрос), все SL, подключенные к шине, загрузят соответствующий регистр указанным значением.

Ниже приведен пример записи регистра 40136 значением 926 в SL с номером 17.

Адрес	Функция	Старший байт адреса регистра 40136	Младший байт адреса регистра 40136	Старший байт значения 926	Младший байт значения 926	Контрольная сумма	
11	06	00	87	03	9E	C1	LRC

В случае успешного выполнения функции ответное сообщение идентично запросу.

*Функция 7: чтение статуса*

Во многих случаях для быстрого получения статуса некоторых событий контроллера, желательно иметь в протоколе сообщение, имеющее небольшой размер. Данная функция разработана именно для этой цели.

Функция с номером 7 позволяет пользователю опрашивать состояние восьми ячеек контроллера. Эти ячейки могут программироваться для хранения информации о состоянии контроллера. Широковещательный режим не поддерживается. Назначение этих ячеек зависит от типа контроллера.

Ниже представлен пример запроса статуса SL с номером 17.

Адрес	Функция	Контрольная сумма	
11	07	E8	LRC

В этой функции не требуется поле данных.

Нормальный ответ содержит статус восьми ячеек, упакованных в один байт данных:

Адрес	Функция	Данные ячеек	Контрольная сумма	
11	07	6D	7B	LRC

*Функция 8: тестовая функция*

Данная функция предназначена для проверки коммуникационной системы и не влияет на данные прибора.

Поле информации содержит 2 байта диагностического кода, указывающего SL выполнить определенное действие, и 2 байта необходимой для данной диагностики информации.

Таблица 19

### Диагностические коды

Код	Действие
00	Вернуть запрос
01	Сбросить установки связи (без ответа)
02	Вернуть регистр диагностики
03	Изменить символ начала пакета
04	Перевести SL в режим прослушивания линии без посылки ответных сообщений (Listen Only Mode)
05	Сбросить счетчики и регистр диагностики
06	Вернуть счетчик сообщений, полученных с шины MODBUS.
07	Вернуть счетчик сообщений с неправильными контрольными суммами.
08	Вернуть счетчик сообщений, вызвавших исключительную ситуацию.
09	Вернуть счетчик сообщений, адресованных только данному SL.
10	Вернуть счетчик сообщений, адресованных данному SL и оставленных без ответа.
11	Вернуть счетчик сообщений, адресованных данному SL и вызвавшим исключительную ситуацию NACK.
12	Вернуть счетчик сообщений, адресованных данному SL и вызвавшим исключительную ситуацию BUSY.

Ниже дан пример запроса вернуть эхо (диагностический код 0) SL с номером 17.

Адрес	Функция	Старший байт диагностического кода	Младший байт диагностического кода	Старший байт данных <sup>9</sup>	Младший байт данных	Контрольная сумма	
11	08	00	00	00	00	0B	LRC

Ответ:

Адрес	Функция	Старший байт диагностического кода	Младший байт диагностического кода	Старший байт данных <sup>10</sup>	Младший байт данных	Контрольная сумма	
11	08	00	00	00	00	0B	LRC

#### Функция 16: запись нескольких регистров

Данное сообщение меняет содержимое любого регистра опрашиваемого контроллера. Сообщение позволяет записывать регистры с максимальным логическим адресом до FFFFH. Неиспользуемые старшие биты адреса регистра должны заполняться нулями. Если используется адрес SL, равный 0, то содержимое поля данных записывается во все устройства, подключенные к шине (широковещательный режим).

Ниже дан пример записи в SL с номером 17 двух регистров 40136, 40137 значениями 0x00a0, 0x0102.

Адрес	Функция	Старший байт адреса первого регистра	Младший байт адреса первого регистра	Количество регистров	Количество байт в поле данных	Старший байт регистра 40136	Младший байт регистра 40136	Старший байт регистра 40137	Младший байт регистра 40137	Контрольная сумма		
11	10	00	87	00	02	04	00	0A	01	02	45	LRC

Нормальное ответное сообщение возвращает адрес SL, функцию, адрес первого регистра и количество записанных регистров.

Адрес	Функция	Старший байт адреса первого регистра	Младший байт адреса первого регистра	Количество регистров	Контрольная сумма		
11	10	00	87	00	02	56	LRC

*Функция 17: чтение информации об адресуемом устройстве*

Пример запроса прибору с адресом 17:

Адрес	Функция	Контрольная сумма	
11	11	DE	LRC

Общая форма ответного сообщения приведена ниже.

Адрес	Функция	Число байт в поле данных	Поле данных	Контрольная сумма
-------	---------	--------------------------	-------------	-------------------

## *ПРИЛОЖЕНИЕ 2. Имитационное моделирование системы управления с панелью оператора Weintek MT8070iE*

### ***Назначение и состав системы***

Рассматриваемая система является имитационной моделью простейшей АСУ ТП и включает модель объекта управления, симулятор ПЛК для персонального компьютера под управлением соответствующего программного обеспечения и программа On-Line имитации панели оператора. Система предназначена для обучения студентов навыкам моделирования, программирования и настройки всех компонентов.

В рамках единого комплекса задействуются программные средства разных производителей и классов:

система имитационного моделирования (для построения модели объекта регулирования) MathWorks<sup>®</sup> MATLAB<sup>®</sup>, Simulink<sup>®</sup>;

система класса PC-based controller (для программной реализации алгоритмов управления на языках программирования промышленных контроллеров) 3S-Smart Software Solutions<sup>®</sup> CODESYS<sup>®</sup>, включая PC-эмулятор ПЛК SP PLC WinNT и OPC-сервер;

программное обеспечение для операторских панелей Weintek – EasyBuilderPro<sup>®</sup>, Weintek Lab Ink<sup>®</sup>.

### ***Межпрограммный обмен***

Для связи виртуального контроллера с Simulink-моделью используется технология OPC (OLEforProcessControl).

OPC– набор повсеместно принятых спецификаций, предоставляющих универсальный механизм обмена данными в системах контроля и управления. OPC обеспечивает независимость потребителей от наличия или отсутствия драйверов или протоколов, что позволяет выбирать оборудование и программное обеспечение, наиболее полно отвечающее реальным потребностям приложения.

OPC-сервер – программа, получающая данные во внутреннем формате устройства или системы и преобразующая эти данные в формат OPC. OPC-сервер является источником данных для OPC-клиентов. По своей сути OPC-сервер – это некий универсальный драйвер физического оборудования, обеспечивающий взаимодействие с любым OPC-клиентом.

В системе используется OPC-сервер CoDeSys, связанный с контроллером CoDeSys SP PLC WinNT через «общий» шлюз типа TCP/IP, порт 1200. Список переменных для обмена формируется в контроллере. Matlab является OPC-клиентом (рис. 43).

Симулятор панели оператора Weintek и виртуальный ПЛК ведут обмен информацией друг с другом по протоколу ModbusTCP/IP, порт 502 (рис. 43).

ModbusTCP/IP – версия протокола Modbus, использующая сети TCP/IP для передачи данных. Адресация узлов производится по их IP-адресам. Доступ к данным «внутри» устройства осуществляется по стандартным правилам Modbus.



Панель оператора является «мастером сети», а ПЛК – «ведомым устройством», или *сервером Modbus*. Протокол позволяет обеим программам работать на одном и том же компьютере.

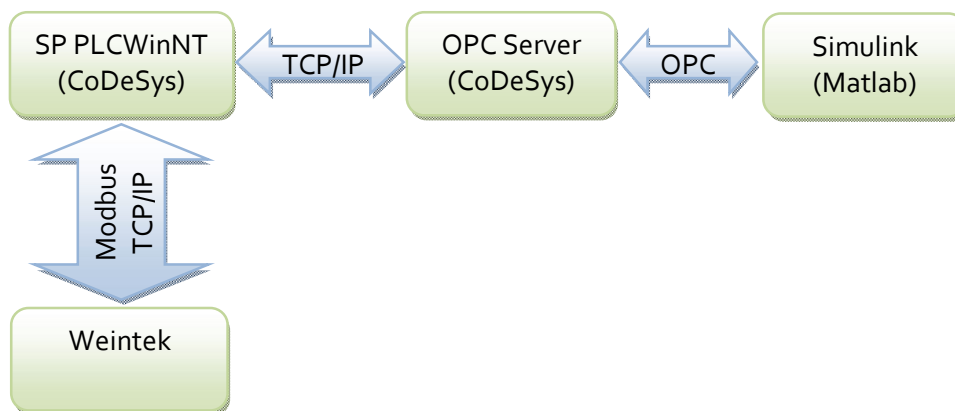


Рис. 43. Взаимодействие программ.

### Запуск OPC-сервера

Для запуска OPC-сервера требуется выполнить следующие шаги.

1. Создать программу (проект) в среде CoDeSys целевой платформой 3S CoDeSys SP PLC WinNT V2.4. При настройке целевой платформы следует установить параметр *Download symbol file* вкладки *General* (рис. 44). Главная программа контроллера PLC\_PRG в нашем случае будет написана на языке FBD (Function Block Diagram).

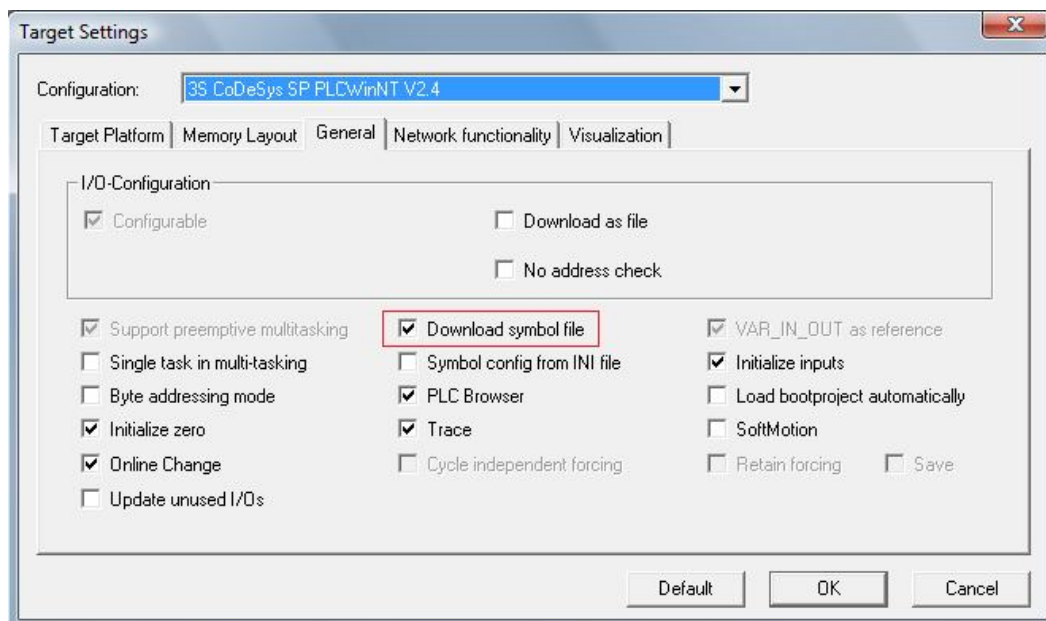


Рис. 44. Настройка целевой платформы.

2. Объявить переменные для обмена по OPC и написать программу PLC\_PRG. В программе должен быть, по меньшей мере, один оператор, по-

скольку без этого она не компилируется. В нашем случае переменными OPC будут локальные переменные PLC\_PRG, а сама программа пока лишь будет вызывать функциональный блок PID (ПИД-регулятор) из библиотеки Util.lib (рис. 45). Библиотеку Util.lib нужно подключить в менеджере библиотек Library manager, вызванном из меню Window, она находится в папке

C:\Program Files\3S Software\CoDeSys V2.3\Library.

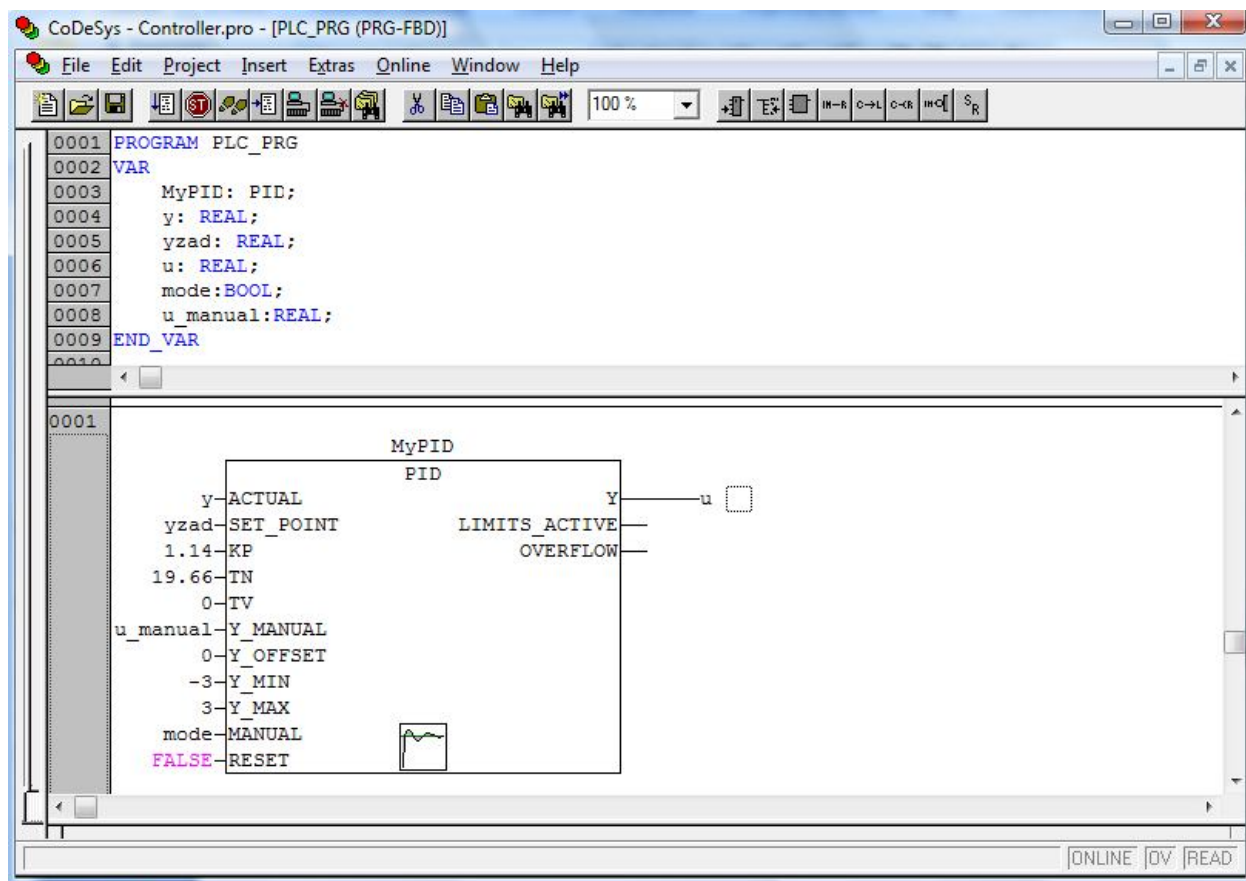


Рис. 45. Программа PLC\_PRG (не закончена).

Функциональный блок MyPID реализует алгоритм ПИ-регулирования с ограничением выходного сигнала и возможностью перехода на ручное управление. Его настройки KP, TN определены для объекта, описанного ниже.

Переменные y, u, yzad, u\_manual (ручное управление), mode (режим) будут задействованы в обмене с панелью оператора по протоколу Modbus TCP/IP; y, u, кроме того, будут участниками обмена с Simulink-моделью по OPC.

3. Сохранить проект под осмысленным именем в отдельную папку.

4. Запустить PLCWinNT (Пуск → Все программы → 3S Software → CoDeSys SP PLC WinNT → CoDeSys SP PLC WinNT V2.4) (рис. 46), установить связь и загрузить программу в «контроллер».

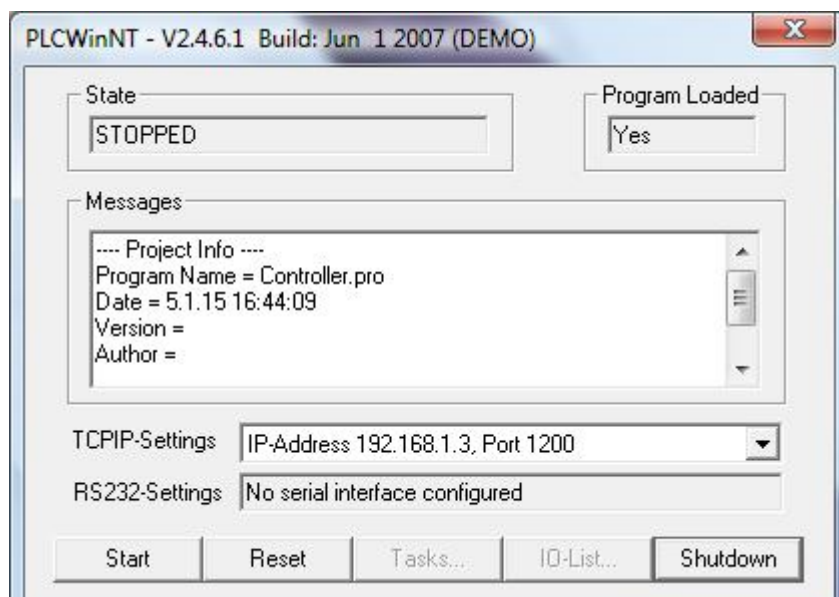


Рис. 46. Окно программы PLC WinNT.

5. Отключить CoDeSys от PLC WinNT и перейти в Опции (Options) в меню Project. Выбрать Symbol Configuration и установить «галочку» *Dump symbol entries* (рис. 47).

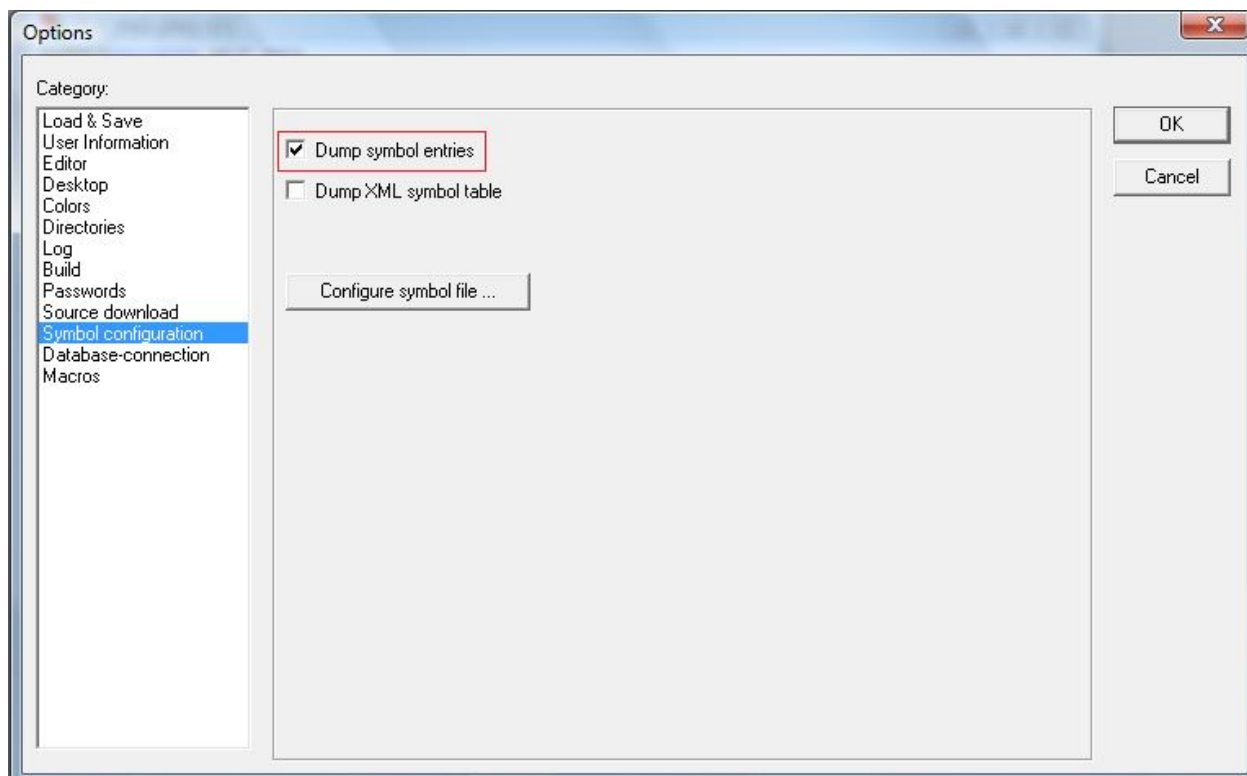


Рис. 47. Установка опций.

6. Сконфигурировать «символьный файл», выбрав переменные для обмена по OPC (рис. 48). В нашем случае это будут переменные PLC\_PRG.u (управляющий сигнал) и PLC\_PRG.y (регулируемая величина объекта).

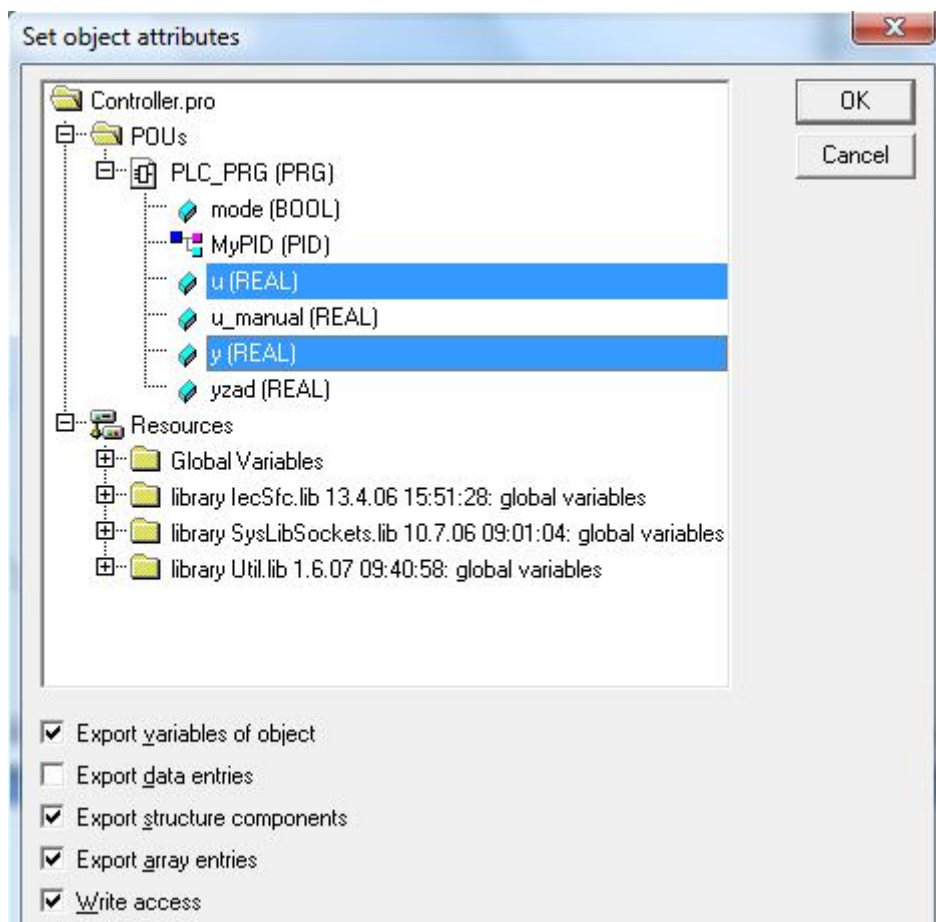


Рис. 48. Выбор переменных для обмена по OPC.

7. Настроить параметры OPC-сервера. Для этого необходимо запустить конфигуратор OPC-сервера (Пуск → Все программы → 3S Software → Communication → CoDeSys OPC Configurator). В окне конфигуратора требуется добавить PLC (Append PLC) и настроить соединение (Connection). По существу достаточно выбрать из списка соединение, настроенное в проекте CoDeSys (рис. 49).

8. «Перестроить» программу (проект) CoDeSys, включив в нее все изменения, сделанные после загрузки в контроллер. Для этого требуется вызвать команду Clean all из меню Project, затем команду Rebuild all из того же меню. CoDeSys перекомпилирует программу и перезагрузит проект при следующем подключении к ПЛК.

9. Запустить программу на выполнение.

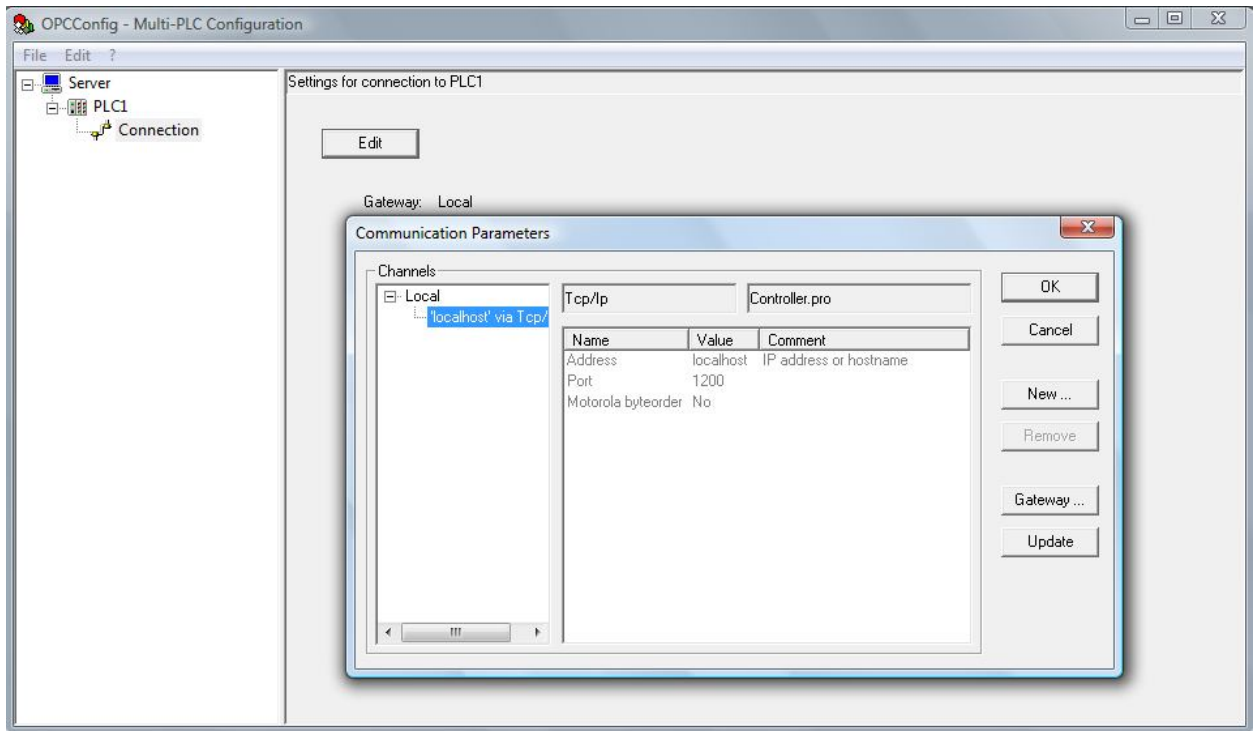


Рис. 49. Настройка OPC-сервера.

**Имитационная модель объекта управления**

Simulink-модель объекта управления показана на рис.50.

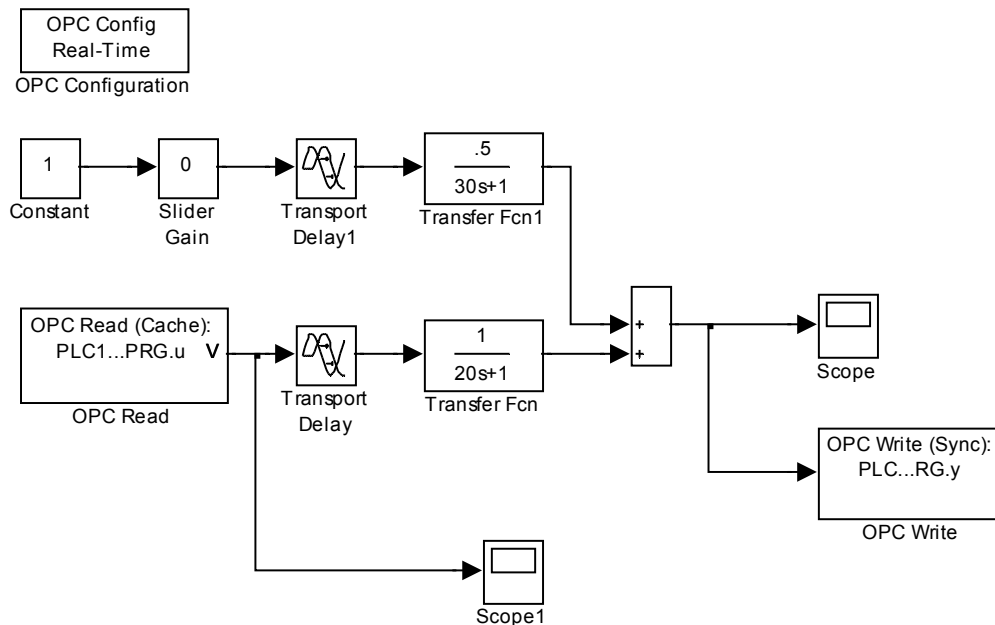


Рис.50. Simulink-модель объекта управления.

Объект имеет два канала воздействия:

управление, формируемое контроллером, считывается блоком OPC Read; возмущение, формируемое с помощью блока Slider Gain.

По обоим каналам объект описывается передаточными функциями первого порядка с запаздыванием. Запаздывание по управлению равно 10 с, по возмущению – 15 с.

Выходная величина объекта с помощью блока OPC Write передается контроллеру.

Для создания блоков OPC Read и OPC Write рекомендуется использовать утилиту орstool, вызвав ее из окна команд Matlab. Эта утилита позволяет достаточно просто подключиться к OPC-серверу и сконфигурировать OPC-клиента, рис. 51, 52.

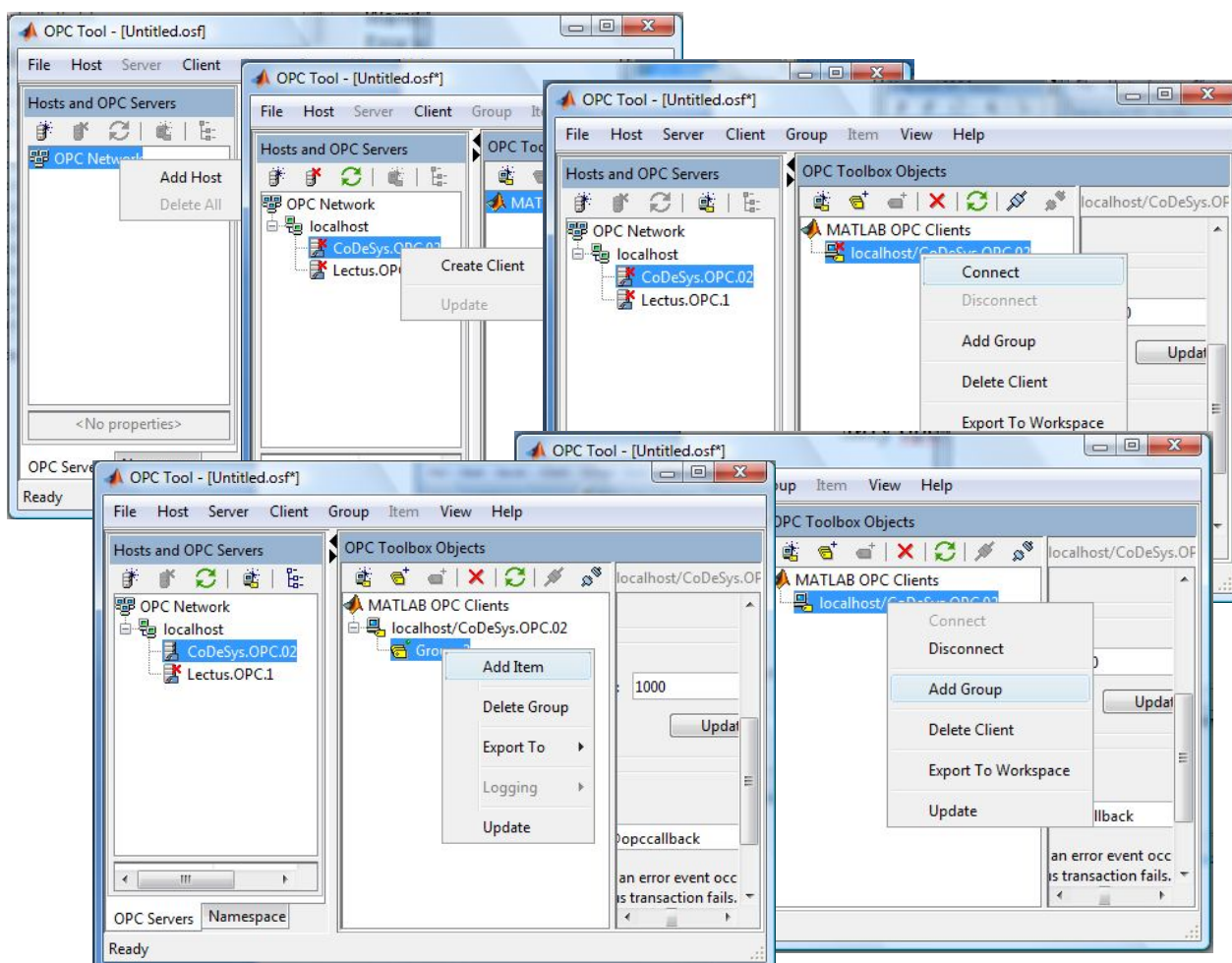


Рис. 51. Подключение к OPC-серверу в орstool.

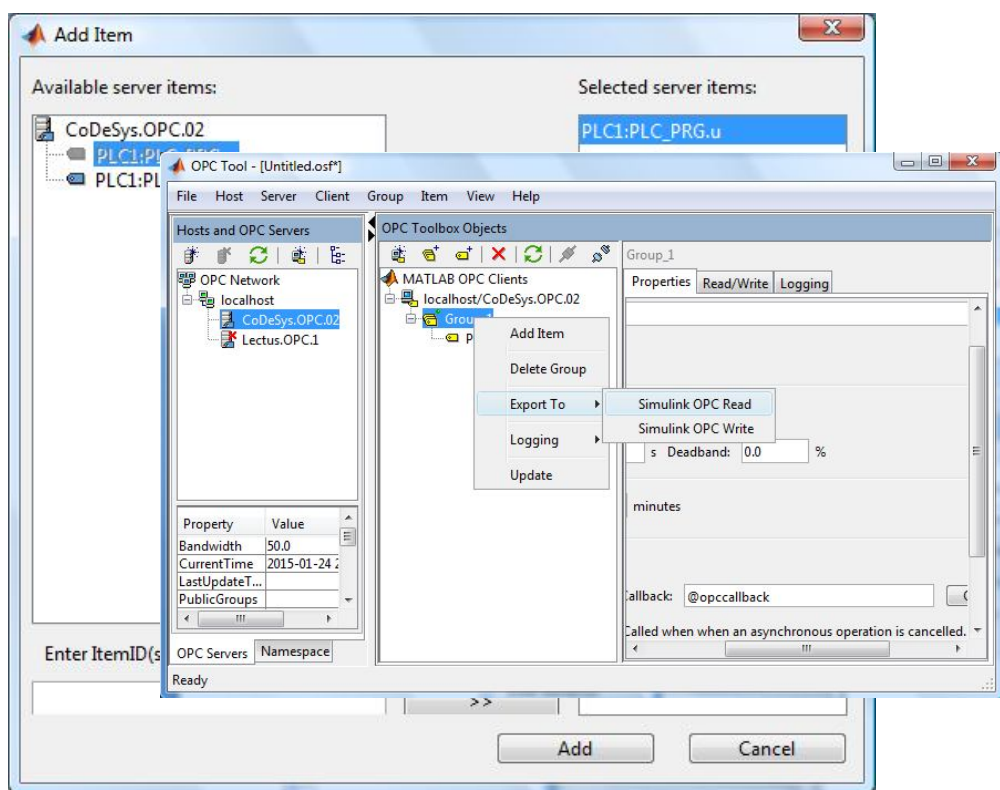


Рис. 52. Выбор тегов и создание блока OPC Write (Read).

Последовательность действий по созданию блоков OPC Read и OPC Write следующая, см. рис. 51:

- 1) Add Host – добавить узел сети, на котором запущен OPC-сервер. В нашем случае сервер запускается на локальном компьютере, поэтому выбирается localhost. После выбора узла появляется список доступных OPC-серверов;
- 2) Create Client – создать клиента, здесь выбирается конкретный OPC-сервер, в нашем случае CoDeSys;
- 3) Connect – подключиться к серверу (в частности, для получения информации о тегах-переменных);
- 4) Add Group – добавить группу тегов (в OPC обмен всегда ведется группами, группы создает клиент);
- 5) Add Item – добавить теги. Из списка доступных тегов выбираются те, что должны образовать группу, после этого они становятся доступны в окне OPC Toolbox Object;
- 6) Export To – экспортировать в Simulink OPC Read или OPC Write. Созданный блок попадает в новую Simulink модель, откуда его можно скопировать.

После создания всех необходимых блоков OPC Read и OPC Write утилиту орstool можно закрыть (с сохранением или без сохранения конфигурации). Связав интерфейсные блоки с вычислительной частью, запустите модель на исполнение, задав время расчета достаточно большим (500 – 1000 с). Блок OPC Config будет создана автоматически.

После запуска модели возможно появление сообщения об ошибках. В большинстве случаев ошибки связаны либо с отсутствием связи (OPC-сервер не запущен, запустите программу контроллера на выполнение), либо с несовпадением типов данных (настройте блок OPC Read, указав тип данных double, кроме того, полезно отключить «ненужные» выходы блока).

При успешном запуске модели убедитесь в наличии обмена данными с PLCWinNT. Для этого можно, в частности, изменить значение переменной `yzad` и наблюдать изменение других переменных.

### ***Разработка управляющей программы***

Основная проблема при разработке программы состоит в организации обмена данными между контроллером и панелью оператора. Симулятор панели не поддерживает обмен данными по OPC-протоколу, однако вполне может использовать встроенную поддержку ModbusTCP/IP в том числе для межпрограммного соединения внутри одного и того же компьютера. С другой стороны, PLCWinNT в отличие от многих «реальных» контроллеров, например ПЛК 150, не имеет встроенной поддержки протокола Modbus.

Проблема решается с помощью специальной программы, использующей «внешние» библиотеки для PLCWinNT. Программа выложена в общий доступ на форуме сайта компании «Овен» пользователем `capzar` (<http://www.owen.ru/forum/attachment.php?attachmentid=12187&d=1394041761>) и с незначительными изменениями приведена ниже.

Для организации связи нам потребуются следующие библиотеки:

```
SYSLIBCALLBACK.LIB 16.4.07
```

```
SysLibSockets.lib 10.7.06
```

Следует отметить, что и библиотеки разработаны специально для PLCWinNT и поэтому они размещаются не в стандартной папке

```
C:\Program Files\3S Software\CoDeSys V2.3\Library,
```

где хранятся lib-файлы для реальных контроллеров, а в специальной:

```
C:\Program Files\Common Files\CAA-Targets\3S\Lib_PLCWinNT.
```

После подключения библиотек требуется:

объявить глобальные переменные и константы;

объявить пользовательские типы данных;

создать ряд программных единиц (POU), которые, используя средства библиотек, и будут осуществлять обмен по ModbusTCP/IP.

*Глобальные переменные и константы* объявляются в разделе Global Variables вкладки Resources:

```
VAR_GLOBAL
    TCP_SERVER_RESET : BYTE;
    SYSLIBSOCKETS_OPTION : BYTE := BYTE#2#0000_0000;
    DATA: ARRAY [0..255] OF WORD;
END_VAR
VAR_GLOBAL CONSTANT
    NETWORK_BUFFER_LONG_SIZE : UINT := 4095;
    NETWORK_BUFFER_SHORT_SIZE : UINT := 1407;
END_VAR
```



*Пользовательские типы данных* объявляются во вкладке Datatypes. Необходимо объявить следующие типы: IP\_C, IP\_FIFO\_DATA, NETWORK\_BUFFER\_SHORT, NW\_BUF\_SHORT и VMAP\_DATA:

```
TYPE IP_C :
STRUCT
    C_MODE : BYTE;
    C_PORT : WORD;
    C_IP : DWORD;
    C_STATE : BYTE;
    C_ENABLE : BOOL;
    R_OBSERVE : BOOL;
    TIME_RESET : BOOL;
    ERROR : DWORD;
    FIFO : IP_FIFO_DATA;
    MAILBOX : ARRAY [1..16] OF BYTE;
END_STRUCT
END_TYPE

TYPE IP_FIFO_DATA :
STRUCT
    X : ARRAY [1..128] OF BYTE;
    Y : ARRAY [1..128] OF BYTE;
    ID : BYTE;
    MAX_ID : BYTE;
    INIT : BOOL;
    EMPTY : BOOL;
    FULL : BOOL;
    TOP : INT;
    NW : INT;
    NR : INT;
END_STRUCT
END_TYPE

TYPE NETWORK_BUFFER_SHORT :
STRUCT
    SIZE : UINT;
    BUFFER : NW_BUF_SHORT;
END_STRUCT
END_TYPE

TYPE
    NW_BUF_SHORT : ARRAY[0..NETWORK_BUFFER_SHORT_SIZE] OF BYTE;
END_TYPE

TYPE VMAP_DATA :
STRUCT
    FC : DWORD;
    V_ADR : INT;
    V_SIZE : INT;
    P_ADR : INT;
    TIME_OUT : TIME;
END_STRUCT
END_TYPE
```

Состав программных единиц (рис. 43) включает:

- 1) функцию BIT\_LOAD\_W, тип возвращаемого параметра WORD, язык программирования ST;
- 2) функцию DWORD\_OF\_BYTE, тип возвращаемого параметра DWORD, язык программирования ST;

- 3) функциональный блок IP\_CONTROL2, язык программирования ST;
- 4) функциональный блок IP\_FIFO, язык программирования ST;
- 5) функциональный блок MB\_SERVER, язык программирования ST;
- 6) функциональный блок MB\_VMAP, язык программирования ST;
- 7) функцию T\_PLC\_MS, тип возвращаемого параметра DWORD, язык программирования ST;
- 8) функцию WORD\_OF\_BYTE, тип возвращаемого параметра WORD, язык программирования ST;
- 9) программу MODBUS, язык программирования CFC;
- 10) программу PLC\_PRG, язык программирования FBD (начата ранее, см. рис. 45).

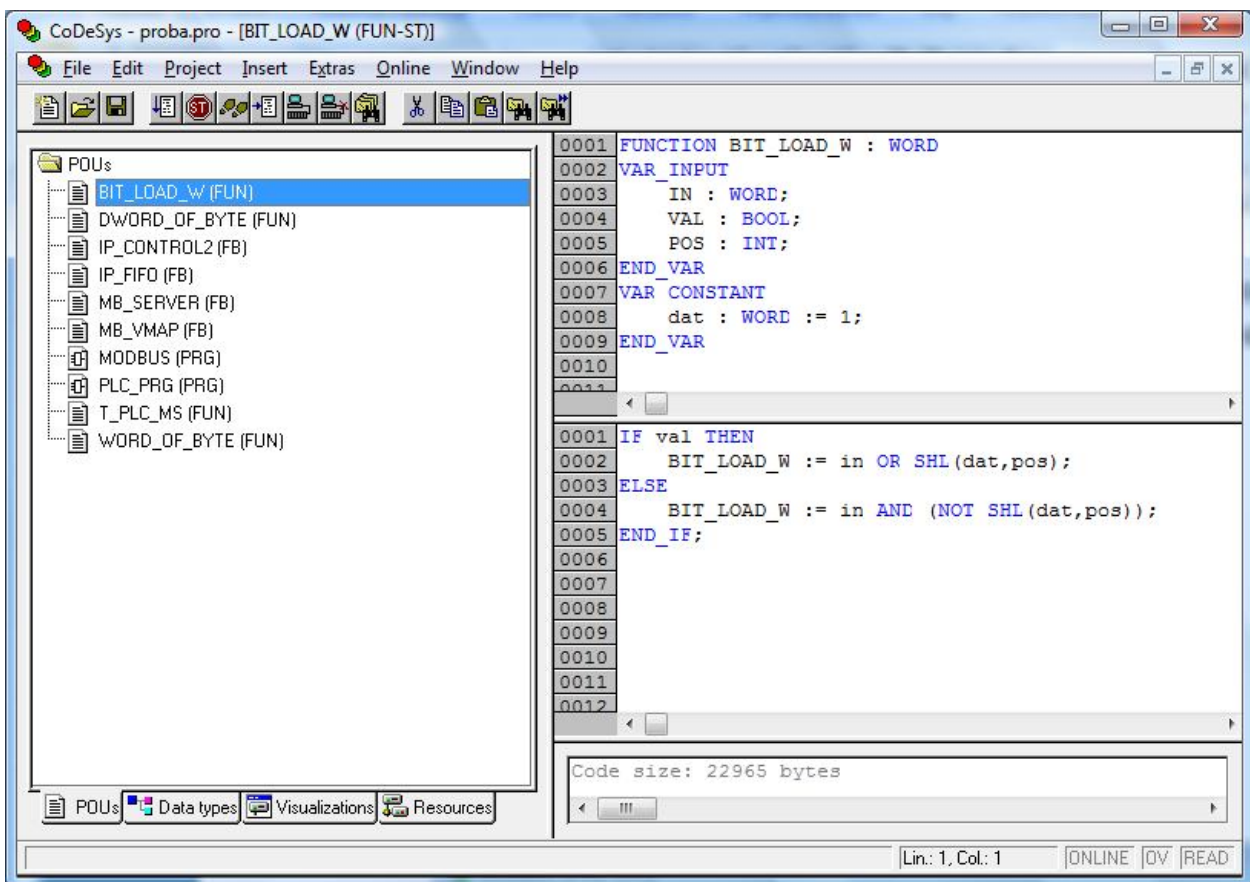


Рис. 53. Состав программы.

Разделы объявлений переменных и коды функций и функциональных блоков на языке ST приведены ниже. Они проверены на работоспособность путем обратного переноса из документа в программу.

```

FUNCTION BIT_LOAD_W : WORD
VAR_INPUT
  IN : WORD;
  VAL : BOOL;
  POS : INT;
END_VAR
VAR_CONSTANT
  dat : WORD := 1;
END_VAR
(* раздел кода *)
IF val THEN
  BIT_LOAD_W := in OR SHL(dat, pos);
ELSE
  BIT_LOAD_W := in AND (NOT SHL(dat, pos));
END_IF;

FUNCTION DWORD_OF_BYTE : DWORD
VAR_INPUT
  B3 : BYTE;
  B2 : BYTE;
  B1 : BYTE;
  B0 : BYTE;
END_VAR
(* раздел кода *)
DWORD_OF_BYTE := SHL(SHL(SHL(BYTE_TO_DWORD(B3), 8) OR
  BYTE_TO_DWORD(B2), 8) OR
  BYTE_TO_DWORD(B1), 8) OR
  BYTE_TO_DWORD(B0);

FUNCTION_BLOCK IP_CONTROL2
VAR_IN_OUT
  IP_C : IP_C;
  S_BUF : NETWORK_BUFFER_SHORT;
  R_BUF : NETWORK_BUFFER_SHORT;
END_VAR
VAR_INPUT
  IP : DWORD;
  PORT : WORD;
  TIME_OUT : TIME;
END_VAR
VAR

```

```

c_time : UDINT;
s_time : UDINT;
r_time : UDINT;
c_enable : BOOL;
c_ip : DWORD;
c_port : WORD;
c_mode : BYTE;
c_status : BYTE;
c_ready : BOOL;
c_ready_old : BOOL;
c_select : BOOL;
c_fdWrite : SOCKET_FD_SET;
c_Timeout : SOCKET_TIMEVAL;
s_total : INT;
s_cur_pos : INT;
s_cur_size : INT;
s_max_size : INT;
s_start : BOOL;
s_active : BOOL;
s_req : BOOL;
s_done : BOOL;
s_status : BYTE;
r_status : BYTE;
r_count : INT;
r_offset : INT;
r_max_size : INT;
new_connection : BOOL;
tx : DWORD;
error_time : DWORD;
state: BYTE;
server_socket : DINT:= SOCKET_INVALID;
socket : DINT:= SOCKET_INVALID;
sockaddr : SOCKADDRESS;
sockaddr_size : DWORD;
plc_841 : BOOL;
plc_881 : BOOL;
dint_true : DINT := 1;
bytes_received : DINT;
bytes_sent : DINT;

```

```

udp_mode : BOOL;
END_VAR
VAR CONSTANT
STOP : BYTE := 0;
TS_INIT : BYTE := 1;
TS_OPEN : BYTE := 2;
UDP_INIT : BYTE := 31;
TC_INIT : BYTE := 51;
TC_CONNECT : BYTE := 52;
C_WAIT : BYTE := 250;
C_CLOSE : BYTE := 251;
END_VAR
(* раздел кода *)
tx := T_PLC_MS ();
IF IP_C.C_PORT = 0 THEN
    IP_C.C_PORT := PORT;
END_IF;
IF IP_C.C_IP = 0 AND IP_C.C_MODE < 4 THEN
    IP_C.C_IP := IP;
END_IF;
new_connection := c_ip <> IP_C.C_IP OR c_mode <> IP_C.C_MODE OR c_port <> IP_C.C_PORT;
c_enable := IP_C.C_ENABLE AND NOT new_connection;
IF IP_C.C_ENABLE AND IP_C.ERROR = 0 AND state = STOP THEN
    c_ip := IP_C.C_IP;
    c_mode := IP_C.C_MODE;
    c_port := IP_C.C_PORT;
    plc_841 := ADR(%MW0)=16#30000000;
    plc_881 := ADR(%MW0)=16#20000000;
    udp_mode := C_MODE.0;
    error_time := TIME_TO_DWORD(MAX(T#200ms,TIME_OUT));
    IP_C.TIME_RESET := TRUE;
    r_max_size := SIZEOF(R_BUF.BUFFER);
    s_max_size := SIZEOF(S_BUF.BUFFER);
    IF C_MODE.0 THEN
        state := UDP_INIT;
    ELSIF C_MODE = 0 THEN
        state := TC_INIT;
    ELSE

```

```

state := TS_INIT;
END_IF;
END_IF;
IF IP_C.TIME_RESET THEN
  IP_C.TIME_RESET := FALSE;
  c_time := tx;
  s_time := tx;
  r_time := tx;
  c_status := BYTE#0;
  s_status := BYTE#0;
  r_status := BYTE#0;
END_IF;
CASE state OF
UDP_INIT:
  socket := SysSockCreate(SOCKET_AF_INET, SOCKET_DGRAM, SOCKET_IPPROTO_IP);
  IF socket < 0 THEN
    c_status := 1;
    state := C_CLOSE;
  ELSE
    SysSockSetOption(socket, SOCKET_SOL, SOCKET_SO_BROADCAST, ADR(dint_true), SIZEOF(dint_true));
    SysSockIoctl(socket, SOCKET_FIONBIO, ADR(dint_true));
    sockaddr.sin_family := SOCKET_AF_INET;
    sockaddr.sin_port := SysSockHtons(c_port);
    sockaddr.sin_addr := SEL(c_mode = 1, SOCKET_INADDR_ANY, SysSockNtohl(c_ip));
    IF c_mode >= 2 THEN
      IF SysSockBind(socket, ADR(sockaddr), SIZEOF(sockaddr)) THEN
        c_ready := TRUE;
        state := C_WAIT;
      ELSE
        c_status := 2;
        state := C_CLOSE;
      END_IF;
    ELSE
      c_ready := TRUE;
      state := C_WAIT;
    END_IF;
  END_IF;
END_IF;
TC_INIT:

```

```

socket := SysSockCreate(SOCKET_AF_INET, SOCKET_STREAM, SOCKET_IPPROTO_IP);
IF socket < 0 THEN
    c_status := 1;
    state := C_CLOSE;
ELSE
    IF plc_841 OR plc_881 THEN
        SysSockSetOption(socket,6,SOCKET_TCP_NODELAY, ADR(dint_true), SIZEOF(dint_true));
    END IF;
    SysSockIoctl(socket, SOCKET_FIONBIO, ADR(dint_true));
    sockaddr.sin_family := SOCKET_AF_INET;
    sockaddr.sin_port := SysSockNtohs(c_port);
    sockaddr.sin_addr := SysSockNtohl(c_ip);
    c_select := FALSE;
    state := TC_CONNECT;
END IF;
TC_CONNECT:
IF NOT c_enable OR c_status = 255 THEN
    state:= C_CLOSE;
ELSE
    IF c_select = FALSE THEN
        IF SysSockConnect(socket,ADR(sockaddr),SIZEOF(sockaddr)) THEN
            c_ready := TRUE;
            state:= C_WAIT;
        ELSE
            c_select := SYSLIBSOCKETS_OPTION.0;
        END IF;
    ELSE
        c_Timeout.tv_sec := 0;
        c_Timeout.tv_usec := 0;
        c_fdWrite.fd_count := 1;
        c_fdWrite.fd_array[0] := socket;
        IF SysSockSelect(SOCKET_FD_SETSIZE,0,ADR(c_fdWrite),0, ADR(c_Timeout)) > 0 THEN
            c_ready := TRUE;
            state:= C_WAIT;
        END IF;
    END IF;
END IF;
TS_INIT:

```

```

server_socket := SysSockCreate(SOCKET_AF_INET, SOCKET_STREAM, SOCKET_IPPROTO_IP);
IF server_socket = SOCKET_INVALID THEN
  c_status := 1;
  state:= C_CLOSE;
ELSE
  SysSockSetOption( server_socket, SOCKET_SOL, SOCKET_SO_REUSEADDR, ADR(dint_true), SIZEOF(dint_true) );
  sockaddr.sin_family:= SOCKET_AF_INET;
  sockaddr.sin_port := SysSockHtons(c_port);
  sockaddr.sin_addr := SOCKET_INADDR_ANY;
  IF NOT SysSockBind(server_socket, ADR(sockaddr), SIZEOF(sockaddr)) THEN
    c_status := 2;
    state := C_CLOSE;
  ELSIF NOT SysSockListen(server_socket, 1) THEN
    c_status := 3;
    state:= C_CLOSE;
  ELSE
    SysSockIoctl(server_socket, SOCKET_FIONBIO, ADR(dint_true));
    state:= TS_OPEN;
  END_IF;
END_IF;
TS_OPEN:
sockaddr_size := SIZEOF(sockaddr);
socket := SysSockAccept(server_socket, ADR(sockaddr), ADR(sockaddr_size));
IF socket <> SOCKET_INVALID THEN
  IF c_mode = BYTE#2 AND sockaddr.sin_addr <> SysSockNtohl(c_ip) THEN
    c_status:= C_CLOSE;
  ELSE
    IF plc_841 OR plc_881 THEN
      SysSockSetOption(socket, 6, SOCKET_TCP_NODELAY, ADR(dint_true), SIZEOF(dint_true));
    END_IF;
    SysSockIoctl(socket, SOCKET_FIONBIO, ADR(dint_true));
    c_ready := TRUE;
    state := C_WAIT;
  END_IF;
  ELSIF NOT c_enable THEN
    state:= C_CLOSE;
  END_IF;
END_IF;
C_WAIT:

```



```

IF NOT c_enable OR NOT c_ready THEN
    state:= c_close;
END_IF
C_CLOSE:
    c_ready := FALSE;
    IF socket >= 0 THEN
        IF (NOT plc_841) AND (NOT plc_881) THEN
            SysSockShutdown(socket, 0);
        END_IF;
        SysSockClose(socket);
        socket := SOCKET_INVALID;
    END_IF
    IF server_socket >= 0 THEN
        IF (NOT plc_841) AND (NOT plc_881) THEN
            SysSockShutdown(server_socket, 0);
        END_IF;
        SysSockClose(server_socket);
        server_socket := SOCKET_INVALID;
    END_IF
    state:= STOP;
END_CASE;
IF c_ready THEN
    IF NOT c_ready_old THEN
        IP_C.C_STATE := 254;
        IP_C.TIME_RESET := TRUE;
    ELSE
        IP_C.C_STATE := 255;
    END_IF;
ELSE
    IF c_ready_old THEN
        IP_C.C_STATE := 1;
        IP_C.MAILBOX[1] := 0;
        IP_C.MAILBOX[2] := 0;
        IP_C.MAILBOX[3] := 0;
        IF NOT new_connection THEN
            s_active := FALSE;
            s_req := FALSE;
            S_BUF.SIZE := 0;
        END_IF;
    END_IF;

```

```

ELSE
    IP_C_C_STATE := 0;
    END_IF;
    END_IF;
    c_ready_old := c_ready;
    IF c_ready AND NOT s_active AND IP_C.MAILBOX[3] = 0 THEN
        IF R_BUF.SIZE >= INT_TO_UINT(r_max_size) THEN
            R_BUF.SIZE := UINT#0;
            r_status := 254;
        END_IF;
        r_offset := SEL(udp_mode, UINT_TO_INT(R_BUF.SIZE), 0);
        r_count := r_max_size - r_offset;
        IF udp_mode THEN
            bytes_received := SysSockRecvFrom(socket, ADR(R_BUF.BUFFER), r_count, 0, ADR(sockaddr), SIZEOF(sockaddr));
            IF c_mode = BYTE#3 AND sockaddr.sin_addr <> SysSockNtohl(c_ip) THEN
                bytes_received := 0;
                R_BUF.SIZE := UINT#0;
                sockaddr.sin_port := SysSockNtohs(c_port);
                sockaddr.sin_addr := SysSockNtohl(c_ip);
            END_IF;
        ELSE
            bytes_received := SysSockRecv(socket, ADR(R_BUF.BUFFER[r_offset]), r_count, 0);
        END_IF;
        CASE bytes_received OF
            -2147483648..-1: ;
            0: IF NOT udp_mode THEN
                    c_status := 253;
                    c_ready := FALSE;
                END_IF;
            ELSE
                r_time := tx;
                R_BUF.SIZE := INT_TO_UINT(r_offset + DINT_TO_UINT(bytes_received));
                IP_C.MAILBOX[1] := IP_C.MAILBOX[1] + 1;
                IF IP_C.MAILBOX[1] = 0 THEN IP_C.MAILBOX[1] := 1; END_IF;
            END_CASE
        END_IF;
        IF S_BUF.SIZE > 0 THEN
            IF c_ready AND c_enable THEN
                IF NOT s_active AND IP_C.MAILBOX[2] = 0 THEN

```

```

s_total := LIMIT(0,UINT_TO_INT(S_BUF.SIZE),r_max_size);
s_cur_pos := 0;
s_cur_size := 0;
s_start := TRUE;
s_active := TRUE;
END_IF;
IF s_done OR s_start THEN
  s_start := FALSE;
  s_cur_pos := s_cur_pos + s_cur_size;
  IF s_total > s_cur_pos THEN
    s_cur_size := s_total - s_cur_pos;
    s_cur_size := LIMIT(0, s_cur_size, s_max_size);
    s_time := tx;
    r_time := tx;
    s_req := TRUE;
  ELSE
    s_req := FALSE;
    s_active := FALSE;
    S_BUF.SIZE := 0;
  END_IF;
END_IF;
END_IF;
IF s_req THEN
  s_done := FALSE;
  IF udp_mode THEN
    bytes_sent:=SysSockSendTo(socket,ADR(S_BUF.BUFFER[s_cur_pos]),s_cur_size,0,ADR(sockaddr),SIZEOF(sockaddr));
  ELSE
    bytes_sent := SysSockSend(socket, ADR(S_BUF.BUFFER[s_cur_pos]), s_cur_size, 0);
  END_IF;
CASE bytes sent OF
-2147483648..-1: ;
0: IF NOT udp_mode THEN
  c_status := 253;
  c_ready := FALSE;
END_IF;
ELSE

```

```

s_done := TRUE;
s_cur_size := DINT_TO_INT(bytes_sent);
END_CASE
END_IF;
IF IP_C.R_OBSERVE = FALSE OR s_active THEN
  r_time := tx;
END_IF;
IF c_status = 0 AND tx - c_time > error_time AND NOT c_ready AND c_mode < 2 THEN
  c_status := 255;
END_IF;
IF s_status = 0 AND tx - s_time > error_time AND s_active THEN
  s_status := 255;
END_IF;
IF r_status = 0 AND tx - r_time > error_time AND c_ready AND IP_C.R_OBSERVE THEN
  r_status := 255;
END_IF;
IP_C.ERROR := DWORD_OF_BYTE(c_status,s_status,r_status,0);

FUNCTION_BLOCK IP_FIFO
VAR_IN_OUT
  FIFO : IP_FIFO_DATA;
  ID : BYTE;
  STATE : BYTE;
END_VAR
VAR
  tmp : INT;
END_VAR
(* раздел кода *)
IF NOT FIFO.INIT THEN
  FIFO.INIT := TRUE;
  FIFO.NW := 1;
  FIFO.NR := 1;
  FIFO.EMPTY := TRUE;
  FIFO.FULL := FALSE;
  FIFO.TOP := 128;
  FIFO.MAX_ID := BYTE#1;
END_IF;
IF ID = BYTE#00 THEN

```

```

IF FIFO.ID < INT_TO_BYTE(FIFO.TOP) THEN
  ID := USINT_TO_BYTE(BYTE_TO_USINT(FIFO.ID) + USINT#1);
  FIFO.ID := ID;
ELSIF STATE < BYTE#200 THEN
  STATE := BYTE#255;
  RETURN;
END_IF;
END_IF;
IF STATE = BYTE#1 AND NOT FIFO.FULL THEN
  tmp := BYTE_TO_INT(ID);
  IF FIFO.Y[tmp] < FIFO.MAX_ID THEN
    FIFO.Y[tmp] := USINT_TO_BYTE(BYTE_TO_USINT(FIFO.Y[tmp]) + USINT#1);
    tmp := FIFO.NW;
    FIFO.X[tmp] := ID;
    IF FIFO.NW = FIFO.TOP THEN FIFO.NW := 1; ELSE FIFO.NW := FIFO.NW + 1; END_IF;
    FIFO.FULL := FIFO.NW = FIFO.NR;
    FIFO.EMPTY := FALSE;
    STATE := BYTE#2;
  END_IF;
END_IF;
IF STATE = BYTE#2 AND NOT FIFO.EMPTY THEN
  tmp := FIFO.NR;
  IF ID = FIFO.X[tmp] THEN
    STATE := BYTE#3;
  END_IF;
END_IF;
IF STATE = BYTE#4 AND NOT FIFO.EMPTY THEN
  tmp := FIFO.NR;
  IF ID = FIFO.X[tmp] THEN
    tmp := BYTE_TO_INT(ID);
    FIFO.Y[tmp] := USINT_TO_BYTE(BYTE_TO_USINT(FIFO.Y[tmp]) - USINT#1);
    IF FIFO.NR = FIFO.TOP THEN FIFO.NR := 1; ELSE FIFO.NR := FIFO.NR + 1; END_IF;
    FIFO.EMPTY := FIFO.NR = FIFO.NW;
    FIFO.FULL := FALSE;
  END_IF;
  STATE := BYTE#5;
END_IF;

```

```

FUNCTION_BLOCK MB_SERVER
VAR_IN_OUT
    IP_C : IP_C;
    S_BUF : NETWORK_BUFFER_SHORT;
    R_BUF : NETWORK_BUFFER_SHORT;
    VMAP : ARRAY [1..10] OF VMAP_DATA;
    DATA : ARRAY [0..255] OF WORD;
END_VAR
VAR_INPUT
    DATA_SIZE : INT;
    ENABLE : BOOL;
    UDP : BOOL;
END_VAR
VAR_OUTPUT
    ERROR : DWORD;
END_VAR
VAR
    MB_VMAP : MB_VMAP;
    adr1 : INT;
    adr2 : INT;
    points : INT;
    points2 : INT;
    coils : BYTE;
    mask : BYTE;
    idx1 : INT;
    idx2 : INT;
    bit_pos : INT;
    count : INT;
    fc : INT;
    i : INT;
    state : INT;
    ip_state : BYTE;
    ip_id : BYTE;
    IP_FIFO : IP_FIFO;
    w_tmp : WORD;
    last_cycle : TIME;
    tx : TIME;
    t : TON;
    ip_error : BOOL;

```

```

END_VAR
(* раздел кода *)
tx := DWORD TO TIME(T_PLC_MS());
FOR i := 1 TO 10 DO
  IF VMAP[i].TIME_OUT > T#0s THEN
    VMAP[i].TIME_OUT := VMAP[i].TIME_OUT + tx - last_cycle;
  END_IF;
END_FOR;
last_cycle := tx;
ip_error := IP_C.ERROR > DWORD#0;
CASE state OF
00:
  IF ENABLE THEN
    state := 10;
    ip_state := BYTE#1;
  END_IF;
10:
  IF ip_state = BYTE#3 THEN
    IP_C.C_PORT := WORD#0;
    IP_C.C_IP := DWORD#0;
    IP_C.C_MODE := SEL(UDP,BYTE#4,BYTE#5);
    IP_C.TIME_RESET := TRUE;
    IP_C.C_ENABLE := TRUE;
    IP_C.R_OBSERVE := FALSE;
    state := 20;
  END_IF;
20:
  IF NOT ip_error THEN
    IF S_BUF.SIZE = UINT#0 AND R_BUF.SIZE > UINT#6 THEN
      IF R_BUF.SIZE = WORD_TO_UINT(WORD_OF_BYTE(R_BUF.BUFFER[4],R_BUF.BUFFER[5])) + UINT#6 THEN
        ERROR := DWORD#00;
        FOR i := 0 TO 13 DO
          S_BUF.BUFFER[i] := R_BUF.BUFFER[i];
        END_FOR;
        fc := BYTE_TO_INT(R_BUF.BUFFER[7]);
        adr1 := WORD_TO_INT(WORD_OF_BYTE(R_BUF.BUFFER[8],R_BUF.BUFFER[9]));
        points := WORD_TO_INT(WORD_OF_BYTE(R_BUF.BUFFER[10],R_BUF.BUFFER[11]));
        MB_VMAP(VMAP:=VMAP,FC:=fc,V_ADR:=adr1,V_CNT:=points,SIZE:=DATA_SIZE);
        adr1:=MB_VMAP.P_ADR;
      END_IF;
    END_IF;
  END_IF;
END_IF;

```

```

bit_pos:=MB_VMAP.P_BIT;
ERROR:=BYTE_TO_DWORD(MB_VMAP.ERROR);
IF ERROR = DWORD#0 THEN
CASE fc OF
1..2 :
count := 9;
coils := BYTE#0;
mask := BYTE#1;
FOR i := 1 TO points DO
IF mask = BYTE#0 THEN
mask := BYTE#1;
S_BUF.BUFFER[count] := coils;
coils := BYTE#0;
count := count + 1;
END_IF;
IF (DATA[adr1] AND SHL(WORD#1,bit_pos)) > WORD#0 THEN
coils := coils OR mask;
END_IF;
bit_pos := bit_pos + 1;
IF bit_pos > 15 THEN
ADR1 := ADR1 + 1;
bit_pos := 0;
END_IF;
mask := SHL(mask,1);
END_FOR;
S_BUF.BUFFER[Count] := coils;
S_BUF.BUFFER[8] := INT_TO_BYTE(count - 8);
S_BUF.SIZE := INT_TO_UINT(count + 1);
3..4 :
idx1 := 07;
count := adr1 + points - 1;
FOR i := adr1 TO count DO
idx1 := idx1 + 2;
idx2 := idx1 + 1;
w_tmp := DATA[i];
S_BUF.BUFFER[idx1] := WORD_TO_BYTE(SHR(w_tmp,8));
S_BUF.BUFFER[idx2] := WORD_TO_BYTE(w_tmp);
END_FOR;
S_BUF.BUFFER[8] := SHL(INT_TO_BYTE(points),1);

```



```

5 : S_BUF.SIZE := INT_TO_UINT (idx2 + 1);
DATA[adr1]:=BIT_LOAD_W(DATA[adr1],R_BUF.BUFFER[10] > BYTE#0,bit_pos);
S_BUF.SIZE := UINT#12;
6 : DATA[adr1] := WORD_OF_BYTE (R_BUF.BUFFER[10],R_BUF.BUFFER[11]);
S_BUF.SIZE := UINT#12;
15 : mask := BYTE#1;
idx1 := 13;
FOR i := 1 TO points DO
DATA[adr1]:=BIT_LOAD_W(DATA[adr1],(R_BUF.BUFFER[idx1]AND mask)> BYTE#0,bit_pos);
bit_pos := bit_pos + 1;
IF bit_pos > 15 THEN
adr1:= adr1 + 1;
bit_pos := 0;
END_IF;
mask := ROL(mask,1);
IF mask = BYTE#1 THEN
idx1 := idx1 +1;
END_IF;
END_FOR;
S_BUF.SIZE := UINT#12;
16 : idx1 := 11;
count := adr1 + points - 1;
FOR i := adr1 TO count DO
idx1 := idx1 + 2;
idx2 := idx1 + 1;
Data[i] := WORD_OF_BYTE (R_BUF.BUFFER[idx1],R_BUF.BUFFER[idx2]);
END_FOR;
S_BUF.SIZE := UINT#12;
22 : w_tmp := WORD_OF_BYTE (R_BUF.BUFFER[10],R_BUF.BUFFER[11]);
DATA[adr1] := (DATA[adr1] AND w_tmp) OR
(WORD_OF_BYTE (R_BUF.BUFFER[12],R_BUF.BUFFER[13]) AND (NOT w_tmp));
S_BUF.SIZE := UINT#14;
23 :

```

```

adr2 := WORD_TO_INT(WORD_OF_BYTE(R_BUF.BUFFER[12], R_BUF.BUFFER[13]));
points2 := WORD_TO_INT(WORD_OF_BYTE(R_BUF.BUFFER[14], R_BUF.BUFFER[15]));
MB_VMAP (VMAP:=VMAP, FC:=16, V_ADR:=adr2, V_CNT:=points2, SIZE:=DATA_SIZE);
adr2:=MB_VMAP.P_ADR;
ERROR:=BYTE_TO_DWORD(MB_VMAP.ERROR);
IF ERROR = DWORD#0 THEN
    idx1 := 15;
    count := adr2 + points2 - 1;
    FOR i := adr2 TO count DO
        idx1 := idx1 + 2;
        idx2 := idx1 + 1;
        Data[i] := WORD_OF_BYTE(R_BUF.BUFFER[idx1], R_BUF.BUFFER[idx2]);
    END FOR;
    idx1 := 07;
    count := adr1 + points - 1;
    FOR i := adr1 TO count DO
        idx1 := idx1 + 2;
        idx2 := idx1 + 1;
        w_tmp := DATA[i];
        S_BUF.BUFFER[idx1] := WORD_TO_BYTE(SHR(w_tmp, 8));
        S_BUF.BUFFER[idx2] := WORD_TO_BYTE(w_tmp);
    END FOR;
    S_BUF.BUFFER[8] := SHL(INT_TO_BYTE(points), 1);
    S_BUF.SIZE := INT_TO_UINT(idx2 + 1);
END_IF;
ELSE
    ERROR := DWORD#01;
END_CASE;
END_IF;
END_IF;
IF ERROR > DWORD#0 THEN
    S_BUF.BUFFER[7] := S_BUF.BUFFER[7] OR BYTE#2#1000_0000;
    S_BUF.BUFFER[8] := DWORD_TO_BYTE(ERROR);
    S_BUF.SIZE := UINT#9;
END_IF;
S_BUF.BUFFER[4] := BYTE#0;
S_BUF.BUFFER[5] := UINT_TO_BYTE(S_BUF.SIZE - UINT#6);
END_IF;
ELSE

```

```

        ERROR := IP_C.ERROR;
    END_IF;
    t(IN:= ip_error, PT:=T#5s);
    IF t.Q THEN
        IP_C.TIME_RESET := TRUE;
    END_IF;
    R_BUF.SIZE := UINT#0;
    IF ENABLE = FALSE THEN
        ip_state := BYTE#4;
        state := 00;
    END_IF;
END_CASE;
IP_FIFO(FIFO:=IP_C.FIFO, STATE:=ip_state, ID:=ip_id);

FUNCTION BLOCK MB_VMAP
VAR_IN_OUT
    VMAP : ARRAY [1..10] OF VMAP_DATA;
END_VAR
VAR_INPUT
    FC : INT;
    V_ADR : INT;
    V_CNT : INT;
    SIZE : INT;
END_VAR
VAR_OUTPUT
    P_ADR : INT;
    P_BIT : INT;
    ERROR : BYTE;
END_VAR
VAR
    i : INT;
    vopt : DWORD;
    vadr : INT;
    padr : INT;
    mask : DWORD;
    vsize : INT;
    w_tmp : WORD;
    i_tmp : INT;
    init : BOOL;

```

```

END_VAR
(* раздел кода *)
IF init = FALSE THEN
  init := TRUE;
  IF VMAP[1].FC = DWORD#0 THEN
    VMAP[1].FC := DWORD#16#FFFF_FFFF;
    VMAP[1].V_ADR := 0;
    VMAP[1].V_SIZE := 256;
    VMAP[1].P_ADR := 0;
  END_IF;
END_IF;
ERROR := BYTE#02;
mask := SHL(DWORD#1,FC);
IF (mask AND DWORD#2#00000000_01000000_00000000_01100000) <> DWORD#0 THEN
  V_CNT := 1;
ELSIF V_CNT = 0 THEN
  RETURN;
END_IF;
FOR i := 1 TO 10 DO
  vopt := VMAP[i].FC;
  IF (vopt AND mask) <> DWORD#0 THEN
    vadr := VMAP[i].V_ADR;
    vsize := VMAP[i].V_SIZE;
    padr := VMAP[i].P_ADR;
    IF (mask AND DWORD#2#00000000_00000000_10000000_00100110) <> DWORD#0 THEN
      i_tmp := WORD_TO_INT(SHR(INT_TO_WORD(V_ADR + V_CNT - 1), 4)) + 1;
      IF i_tmp <= SIZE AND i_tmp <= vsize THEN
        w_tmp := INT_TO_WORD(V_ADR);
        P_ADR := WORD_TO_INT(SHR(w_tmp, 4)) + padr;
        P_BIT := WORD_TO_INT(w_tmp AND WORD#16#000F);
        ERROR := BYTE#0;
        EXIT;
      END_IF;
    ELSIF (mask AND DWORD#2#00000000_11000001_00000000_01011000) <> DWORD#0 THEN
      IF V_ADR >= vadr AND V_ADR + V_CNT <= vadr + vsize THEN
        P_ADR := V_ADR - vadr + padr;
        IF P_ADR + V_CNT <= SIZE THEN
          ERROR := BYTE#0;
        END_IF;
      END_IF;
    END_IF;
  END_IF;

```

```

EXIT;
END_IF;
END_IF;
ELSE
  ERROR := BYTE#01;
  RETURN;
END_IF;
END_IF;
END_FOR;
IF ERROR = BYTE#0 THEN
  IF (mask AND DWORD#2#00000000_11000001_10000000_01100000) <> DWORD#0 THEN
    VMAP[i].TIME_OUT := T#1ms;
  END_IF;
END_IF;
END_IF;

FUNCTION T_PLC_MS : DWORD
VAR CONSTANT
  debug : BOOL := 0;
  N : INT := 0;
  offset : DWORD := 0;
END_VAR
VAR
  tx : TIME;
END_VAR
(* раздел кода *)
tx := TIME();
T_PLC_MS := TIME_TO_DWORD(Tx);
IF debug THEN
  T_PLC_MS := (SHL(T_PLC_MS,N) OR SHL(DWORD#1,N)-1) + OFFSET;
END_IF;

FUNCTION WORD_OF_BYTE : WORD
VAR INPUT
  B1 : BYTE;
  B0 : BYTE;
END_VAR
(* раздел кода *)
WORD_OF_BYTE := SHL(BYTE_TO_WORD(B1),8) OR BYTE_TO_WORD(B0);

```

Программа MODBUS, составленная на языке CFC, показана на рис. 54.

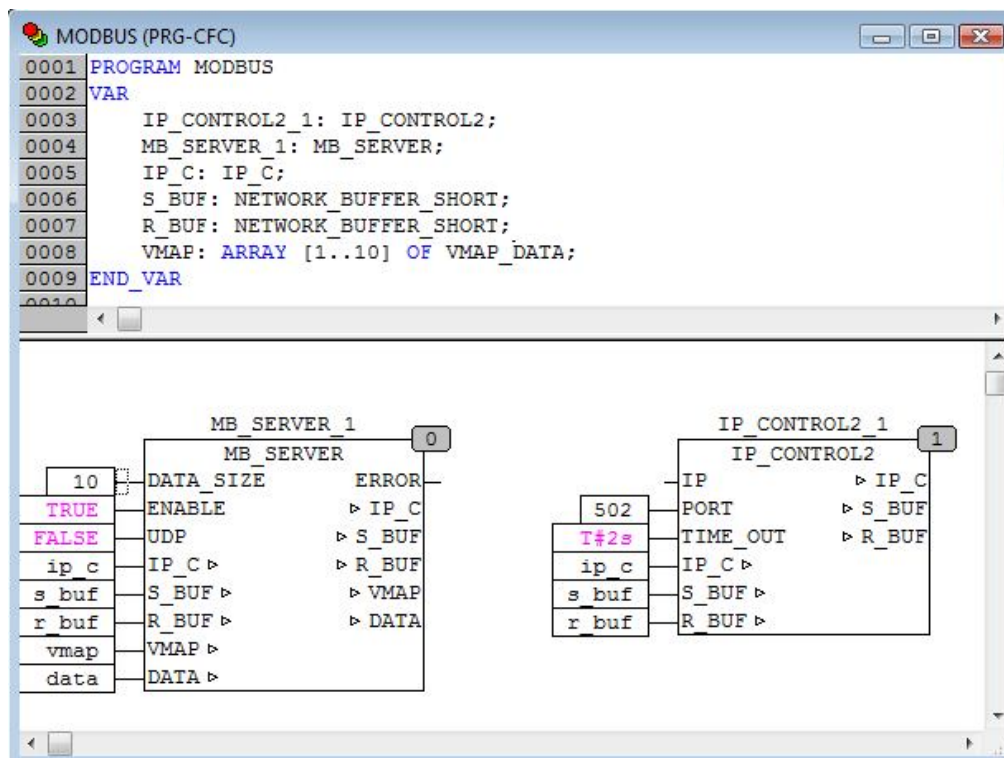


Рис. 54. Программа MODBUS.

Программа осуществляет обмен по протоколу ModbusTCP/IP, используя приведенные ранее функции и функциональные блоки. Переменные обмена («регистры Modbus») сосредоточены в глобальном массиве слов DATA.

В программе будут задействованы следующие элементы массива:

DATA[0] (слово) – задание регулятору, после необходимых преобразований записывается в переменную *yzad* программы PLC\_PRG (см. рис.45 и ниже);

DATA[1] (слово) – сигнал ручного управления, после необходимых преобразований записывается в переменную *u\_manual* программы PLC\_PRG;

DATA[2].0 (бит) – сигнал перехода на ручного управления, после необходимых преобразований записывается в переменную *mode* программы PLC\_PRG;

DATA[3] (слово) – регулируемая величина, в эту переменную записывается пересчитанное значение переменной *u* программы PLC\_PRG, полученное из Simulink-модели объекта;

DATA[4] (слово) – выход регулятора, в эту переменную записывается пересчитанное значение переменной *u* программы PLC\_PRG.

Начатая ранее программа PLC\_PRG дополняется вызовом программы MODBUS и цепями пересчета переменных для согласования диапазонов их изменения в контроллере и модели с одной стороны и панели оператора с другой (рис. 55, табл. 20).

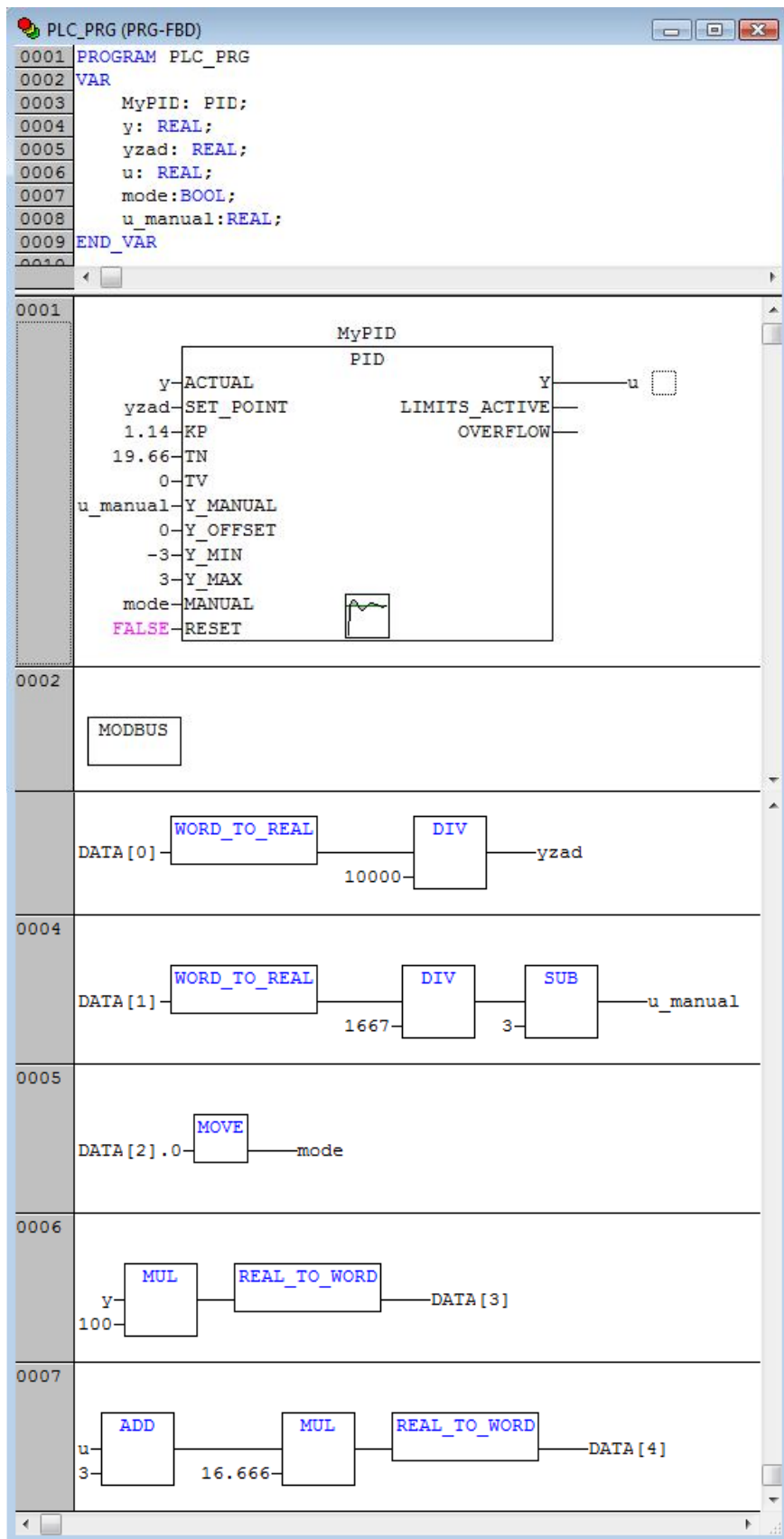


Рис. 55. Программа PLC\_PRG.

### Диапазоны изменение переменных

Величина	Переменная контроллера/модели	Переменная панели/Modbus
Регулируемая величина	$y, 0...1$	DATA[3], 0...100
Управление (выход регулятора)	$u, -3...3$	DATA[4], 0...100
Задание	$yzad, 0...1$	DATA[0], 0...10000
Ручное управление	$u_{manual}, -3...3$	DATA[1], 0...10000

#### Интерфейс пользователя

Пользовательский интерфейс имитирует работу панели оператора WeintekMT8070iE. Для работы с панелью и симуляциииспользуется программный пакет EasyBuilderPro.

После запуска программы требуется указать модель панели и режим (ориентацию) дисплея. Далее автоматически появляется окно «Системные настройки» (рис. 56), в котором необходимо указать все устройства, с которыми панель будет обмениваться данными. В нашем случае такое устройство одно, это «локальный» ПЛК типа MODBUS TCP/IP.

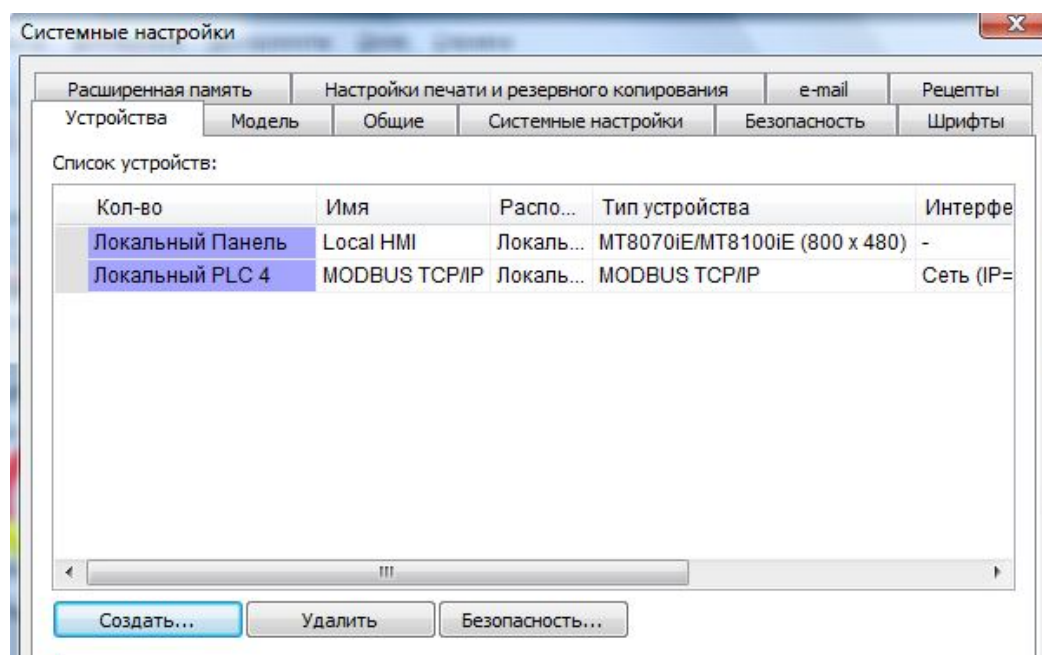


Рис. 56. Системные настройки панели.

При настройке интерфейса (рис. 57) требуется указать IP-адрес ПЛК (это адрес самого компьютера, его можно узнать в командной строке с помощью команды ipconfig) и порт (значение по умолчанию 502 – стандартный порт MODBUS, изменять не нужно).



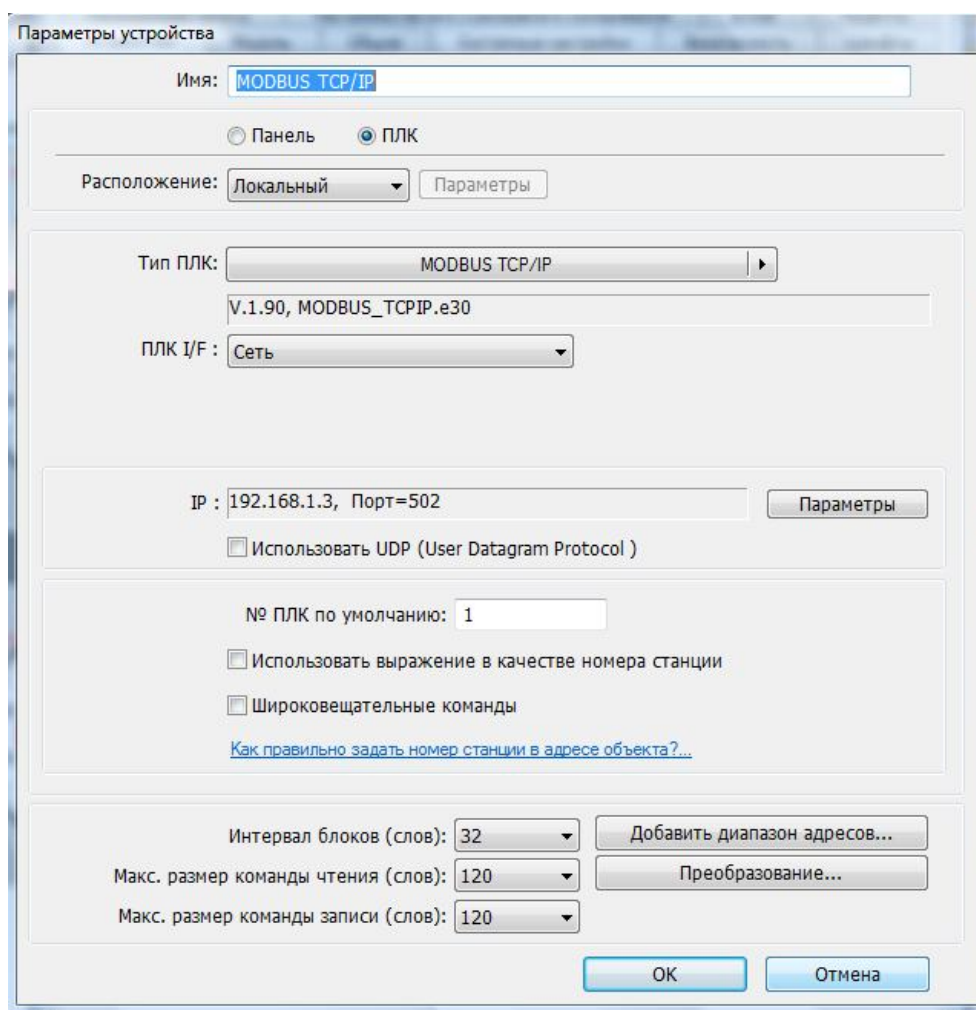


Рис. 57. Настройка интерфейса ПЛК.

В пользовательском окне 10 (рис. 58) разместите элементы визуализации процесса регулирования:

два компонента «Числа» для ввода и отображения текущих задания и сигнала ручного управления;

два «Индикатора» для отображения текущих значений регулируемой величины и сигнала управления;

битовый переключатель («Тумблер») для управления режимом работы регулятора (автомат/ручное);

текстовые поля («Текст») для поясняющих надписей;

«Графические выборки» для отображения графиков изменения регулируемой величины и управляющего сигнала.

Перед размещением последнего компонента требуется сконфигурировать объект «Выборки данных» с помощью окна (рис.59), вызываемого одной из кнопок нижнего ряда панель инструментов. По нажатию кнопки «Создать» открывается окно настроек выборки (рис. 60). Здесь требуется указать начальный номер регистра MODBUS ПЛК, форматы и количество считываемых переменных (рис. 61).

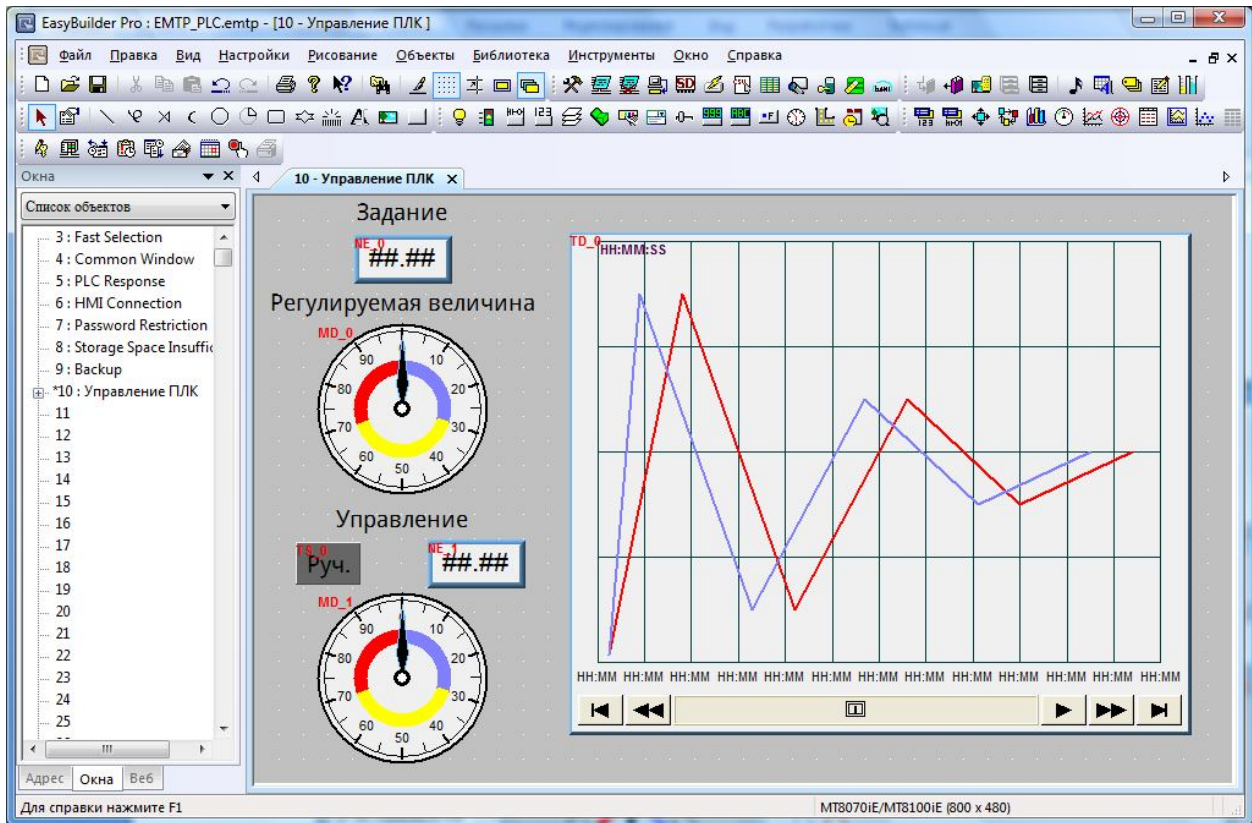


Рис. 58. Настройка интерфейса ПЛК.

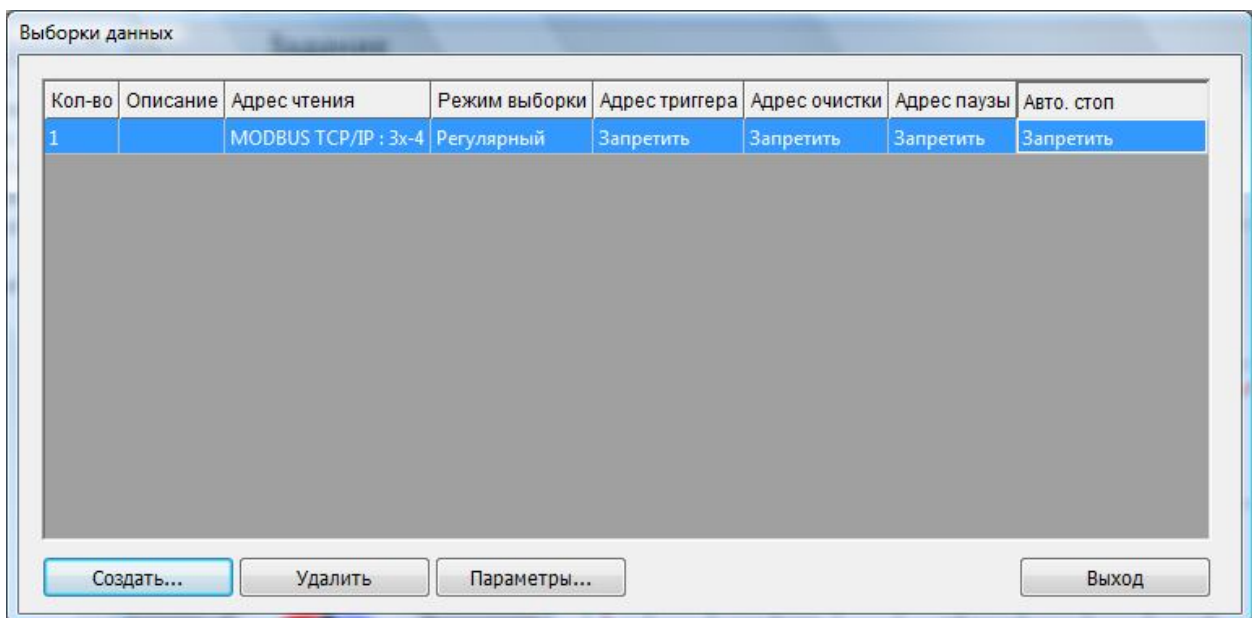


Рис. 59. Окно «Выборки данных».

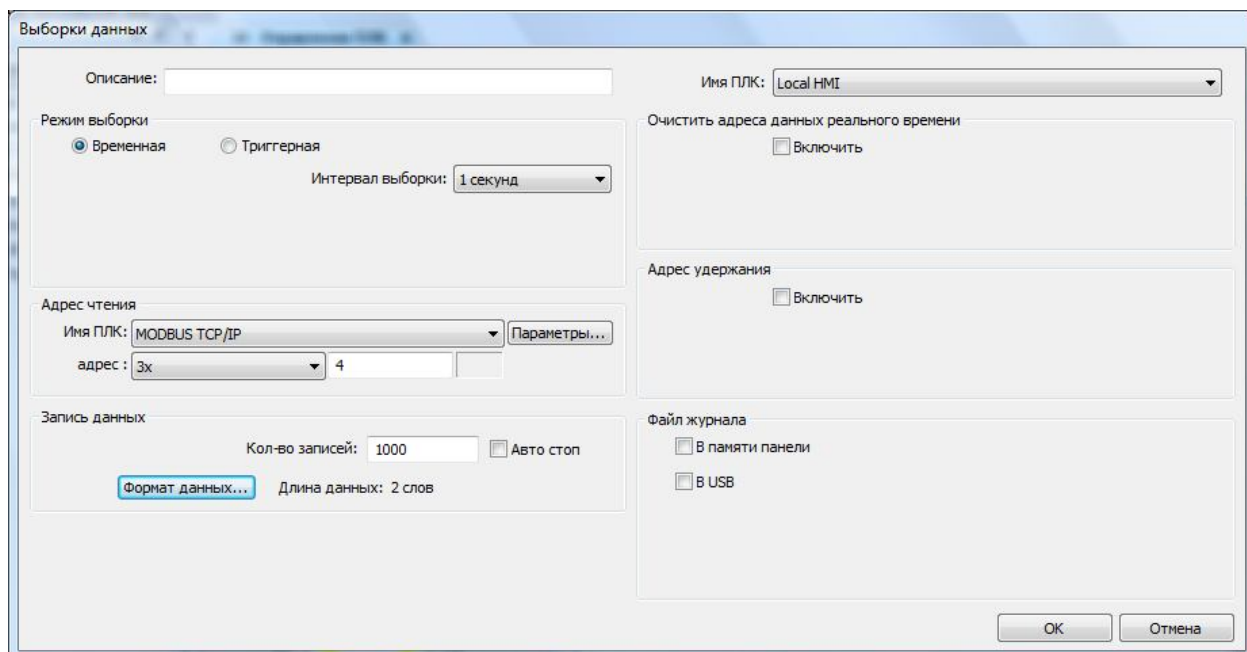


Рис. 60. Настройка выборки.

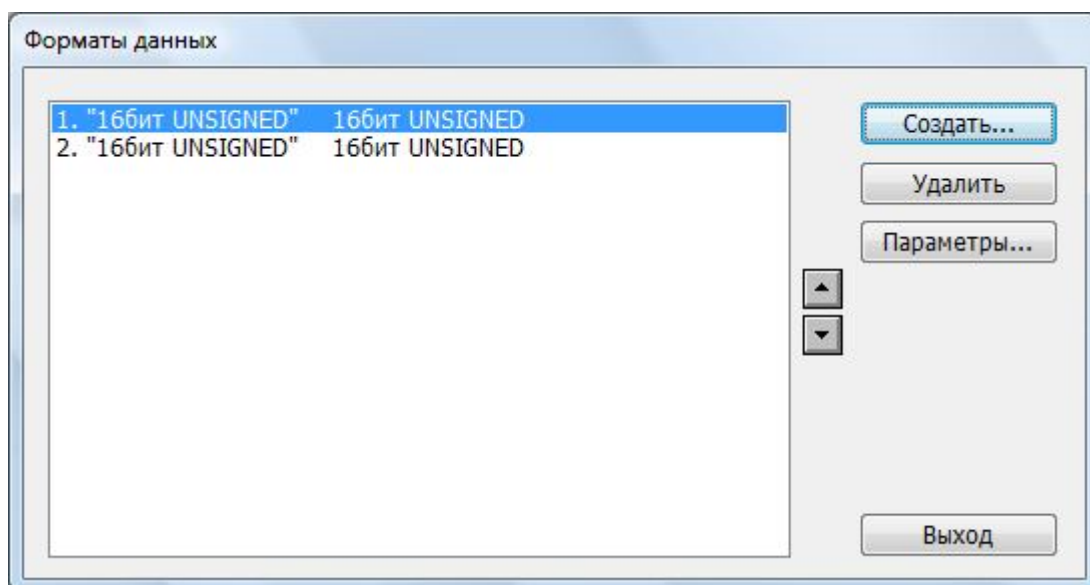


Рис. 61. Форматы данных выборки.

При настройке всех компонентов визуализации (кроме текста) требуется указать адреса чтения/записи. В EasyBuilderPro используются следующие типы адресов протокола MODBUS: 0x, 1x, 3x, 4x, 5x, 6x, 3x\_bit и 4x\_bit.

0x: Регистры флагов. Устройство чтения и записи. При чтении бита данного типа устройства, функциональный код – 01H. При записи бита, функциональный код – 05H. При записи нескольких битов, функциональный код – 0FH.

1x: Дискретный ввод. Устройство только для чтения. При чтении бита, функциональный код – 02Н.

3x: Регистры ввода. Устройство только для чтения. При чтении данных, функциональный код – 04Н.

4x: Регистр хранения. Устройство чтения и записи. При чтении данных, функциональный код – 03Н. При записи данных, функциональный код – 10Н.

5x: Функциональные коды идентичны 4x. Разница в том, что 5x делает замену мест двойных слов при формате данных 32-bit unsigned. Если данные, считываемые 4x – 0x1234, то данные, считываемые 5x – 0x3412.

6x: Устройство чтения и записи. При чтении данных, функциональный код – 03Н. Отличие от 4x в том, что при записи данных, функциональный код – 06Н, запись одного регистра.

3x\_bit: Функциональный код идентичен 3x. Разница в том, что 3x\_bit считывает из данных единичный бит.

4x\_bit: Функциональный код идентичен 4x. Разница в том, что 4x\_bit считывает из данных единичный бит.

В разрабатываемой системе визуализации для всех компонентов, кроме битового переключателя, используется тип адреса 4x. Нумерация регистров начинается с единицы. Таким образом, обращение к элементу массива DATA[0] программы для ПЛК производится как к первому регистру, DATA[1] – как ко второму и т.д.

Компонент «Тумблер» использует тип адресации 0x. Биты нумеруются, начиная с единицы. Поэтому бит DATA[2].0 в EasyBuilderPro будет иметь номер 33.

С остальными настройками компонентов разберитесь самостоятельно. При установке пределов изменения величин ориентируйтесь на данные табл. 1.

После настройки всех компонентов экрана с помощью одной из кнопок верхнего ряда панели инструментов запустите процесс «интерактивного моделирования», убедившись предварительно в нормальной работе Simulink-модели виртуального контроллера PLCWinNT.

Если все настройки были сделаны верно, проект будет откомпилирован и появится окно симуляции. Об установлении связи с контроллером будет свидетельствовать появление на экране всех компонентов визуализации (рис. 62).

Опробуйте органы управления и наблюдайте ход процессов регулирования:

измените задание и наблюдайте переходный процесс по заданию;

измените (с помощью блока Slider Gain Simulink-модели) возмущение и наблюдайте переходный процесс по возмущению;

перейдите на ручное управление и попытайтесь вручную вывести объект на новый режим.

К сожалению, в режиме «On-line Simulation» предел работы составляет 10 минут, однако перезапуск решает эту проблему.

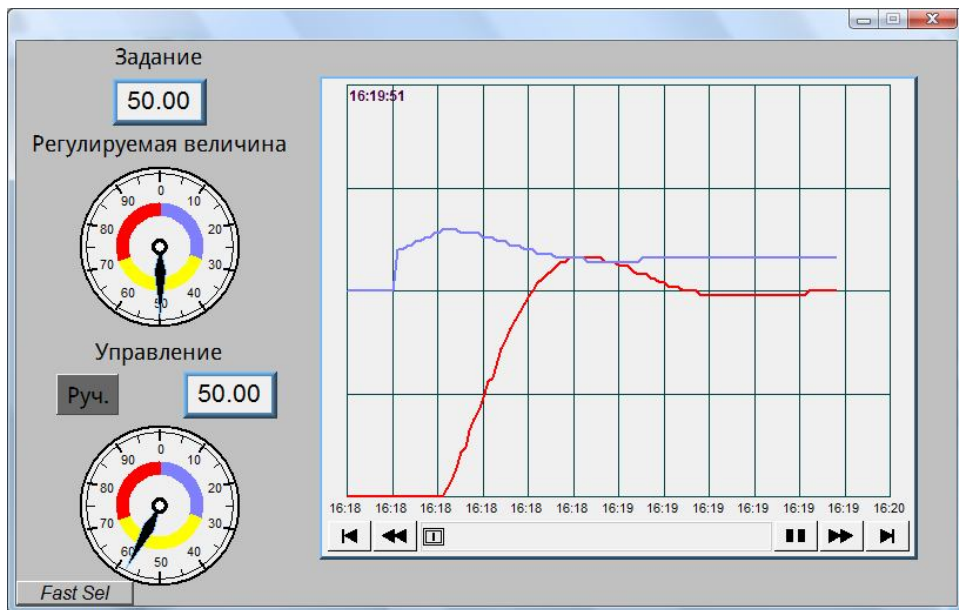


Рис. 62. Окно симулятора панели в работе.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. БУСТ. Блок управления тиристорами и симисторами. Руководство по эксплуатации. – ОВЕН, 2007. – 24 с.
2. ВМА8. Модуль ввода аналоговый измерительный. Руководство по эксплуатации. – ОВЕН, 2007. – 88 с.
3. ВМУ8. Модуль вывода. Паспорт и руководство по эксплуатации. – ОВЕН, 2007. – 64 с.
4. ИП 320. Панель оператора. Паспорт и руководство по эксплуатации. – ОВЕН, 2008. – 21 с.
5. Конфигурирование области ввода/вывода ПЛК. Руководство пользователя. – ОВЕН, 2007. – 24 с.
6. ОВЕН ПЛК150. Контроллер программируемый логический. Паспорт и руководство по эксплуатации. – ОВЕН, 2007. – 38 с.
7. Панель оператора ИП320. Конфигурирование. Руководство пользователя. – ОВЕН, 2007 – 41 с.
8. Преобразователи частоты HITACHI. Инструкция по эксплуатации. Серия SJ100-...NFE/HFE. – ВЭМЗ-СПЕКТР, 1999. – 82 с.
9. Программный пакет EasyBuilder 8000. Руководство пользователя. (рус.). – [http://plcsystems.ru/catalog/Weintek/doc/EB8000\\_rus.pdf](http://plcsystems.ru/catalog/Weintek/doc/EB8000_rus.pdf)
10. Руководство пользователя по программированию ПЛК в CoDeSys 2.3. – 3S Smart Software Solution GmbH, Русская редакция: ПК Пролог, 2006. – 453 с.

Андрей Николаевич Рыбалев  
доц. кафедры АПП и Э АмГУ, канд. техн. наук, доцент

**Программируемые логические контроллеры и аппаратура управления:  
лабораторный практикум. Часть 5. Панели оператора.**

Учебное пособие

---

Изд-во АмГУ. Подписано к печати ???.2015. Формат 60x84/16. Усл. печ. ???  
Тираж ???.