

Министерство науки и высшего образования Российской Федерации

Амурский государственный университет

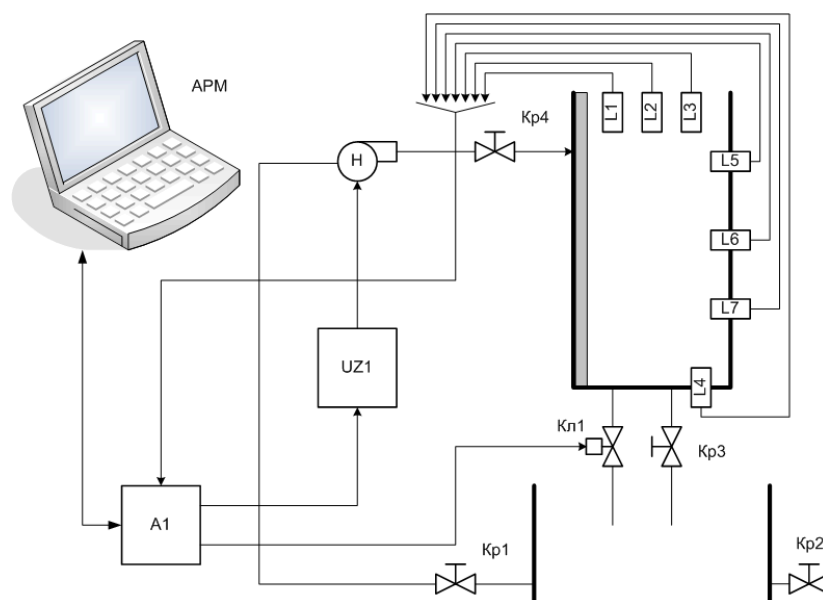
А.Н. Рыбалёв

**ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ
КОНТРОЛЛЕРЫ И АППАРАТУРА УПРАВЛЕНИЯ:
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

Часть 7.

**OMRON CP1E. ИЗМЕРЕНИЕ И РЕГУЛИРОВАНИЕ
УРОВНЯ**

Учебное пособие



Благовещенск
Издательство АмГУ
2024

УДК 62-5
ББК 32.965я73
Р 93

*Рекомендовано
учебно-методическим советом университета*

Рецензенты:

Протасов С.В., начальник монтажно-наладочного участка автоматики СП БТЭЦ;

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

Рыбалёв А.Н. Программируемые логические контроллеры и аппаратура управления: лабораторный практикум. Часть 7. OMRON CP1E. Измерение и регулирование уровня: учебное пособие. – Благовещенск: Амурский гос. ун-т, 2024. – 49 с.

Пособие предназначено для студентов 3 и 4 курсов направления подготовки «Автоматизация технологических процессов и производств», выполняющих работы на лабораторном стенде САУ-У-СК производства НПП «Учтех-Профи». Соответствует рабочим программам дисциплин «Средства автоматизации и управления», «Программное обеспечение систем управления», «Автоматизация технологических процессов и производств».

© Амурский государственный университет, 2024
© Рыбалёв А.Н., 2024

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	3
ВВЕДЕНИЕ.....	4
1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	5
1.1. Ввод-вывод	5
1.2. Организация памяти	11
1.3. Инструкции и операнды	14
2. ЛАБОРАТОРНЫЕ РАБОТЫ	19
2.1. Система ручного и дистанционного управления. Автоматическое управление по сигналам дискретных датчиков	19
2.2. Система измерения уровня. Обработка сигналов измерительных преобразователей	26
2.3. Система автоматического регулирования уровня	32
2.4. Система автоматического воспроизведения периодических сигналов.....	43
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	48

ВВЕДЕНИЕ

В настоящем пособии предлагаются дополнительные задания для лабораторных работ на стенде САУ-У-СК производства НПП «Учтех-Профи» Южно-Уральского государственного университета. Объектом управления на стенде является гидравлическая система, включающая питательный и приемный баки, водяной насос, оснащенный частотно-регулируемым приводом, и систему труб, снабженную запорной арматурой, в том числе электромагнитным клапаном на сливе из приемного бака. Регулируемая величина – уровень воды в приемном баке, который измеряется с помощью четырех измерительных преобразователей различного типа и контролируется тремя поплавковыми датчиками. Четвертый датчик выполняет защитную функцию, отключая подачу при недопустимо высоком уровне. Система управления построена на базе программируемого логического контроллера (ПЛК) CP1E производства компании Omron (Япония), оснащенного модулем ввода/вывода аналоговых сигналов CP1W-MAD11. Программирование ПЛК осуществляется при помощи программы CX-Programmer, запускаемой на персональном компьютере, входящем в состав стенда. Помимо системы программирования, ПЛК на компьютере также может быть запущена SCADA-система Trace Mode от российской компании Adastra, позволяющая визуализировать процессы измерения и управления. Подробное описание стенда, включая программное обеспечение, приведено в методических указаниях к проведению лабораторных работ [1], поставляемых вместе с лабораторной установкой.

В [1] предложены три лабораторные работы по следующим темам: снятие статических характеристик измерительных преобразователей, настройка систем автоматического регулирования, изучение основ программирования ПЛК CP1E. *Все эти работы обязательно должны быть выполнены обучающимися перед тем, как они приступят к выполнению заданий из данного пособия.*

Необходимость дополнения списка работ обусловлена тем, что имеющиеся далеко не в полной мере раскрывают потенциал стенда как средства обучения разработке программного обеспечения. Так, при выполнении первых двух работ используются уже готовые программы для ПЛК и SCADA-система, «внутренности» которых даже не описаны, а предлагаемые в третьей работе варианты довольно простых заданий никак не связаны с основной аппаратурой стенда и отлаживаются на имитаторе сигналов.

Все программное обеспечение, разрабатываемое при выполнении дополнительных лабораторных работ, создается «с нуля» и решает реальные задачи программно-логического управления, измерения и регулирования, взаимодействуя с измерительными преобразователями, датчиками и исполнительными механизмами стенда, а также с человеком-оператором. Особое внимание уделено организации разных уровней системы управления (ручное/местное, дистанционное, автоматическое).

1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Ввод-вывод

Задействованный в работах контроллер CP1E-NA20DT-D [2,3] имеет несколько областей памяти.

СЮ – это область ввода-вывода. Адресуется непосредственно указанием номера 16-разрядного слова и (при необходимости) через точку номера бита. Адресация слов и битов начинается с нуля. Примеры:

0.11 – одиннадцатый бит нулевого слова, адрес двенадцатого (по счету) встроенного дискретного входа, – к нему подключен один из поплавковых датчиков уровня;

91 – девяносто первое (девяносто второе по счету) слово, адрес второго встроенного аналогового входа, – к нему подключен измерительный преобразователь гидростатического давления DMP331.

СЮ охватывает не только встроенные входы и выходы ПЛК, но и весь остальной ввод-вывод, включая модули расширения и последовательный интерфейс:

область входов – слова от 0 до 99 (слова 90 и 91 закреплены за двумя встроенными аналоговыми входами ПЛК);

область выходов – слова от 100 до 199 (слово 190 закреплено за встроенным аналоговым выходом ПЛК);

последовательный интерфейс – слова от 200 до 289.

Контроллер CP1E-NA20DT-D имеет 12 дискретных входов, 8 дискретных выходов (транзисторного типа), 2 аналоговых входа и 1 аналоговый выход. Аналоговые входы и выходы рассчитаны на сигналы 0(1)...5В, 0...10В, -10...10В, 0(4)...20мА. Выбор вида сигнала (напряжение или ток) и диапазона определяется схемой подключения и программными настройками ПЛК.

Дискретные входы доступны по адресам от 0.00 до 0.11, дискретные выходы – по адресам от 100.00 до 100.07. Аналоговые входы имеют адреса 90 и 91, аналоговый выход – адрес 190.

При подключении модулей расширения их входам и выходам автоматически выделяются адреса, следующие за адресами встроенных входов и выходов с учетом выравнивания по длине слова. Поскольку для данной модели ПЛК считается, что ввод и вывод занимают по одному слову (встроенный аналоговый ввод/вывод не берется во внимание), адресация входов первого модуля расширения начинается с 1, а выходов – с 101.

На стенде контроллер оснащен единственным модулем расширения CP1W-MAD11, который имеет два аналоговых входа и один аналоговый выход. Вид и диапазоны сигналов – такие же, как у встроенных входов и выходов ПЛК. Вид сигнала определяется схемой подключения, а диапазон – программно, при инициализации модуля. Входы имеют адреса 1 и 2, выход – 101.

Информация обо всех входах и выходах системы приведена в табл.1.

Таблица 1. Входы и выходы системы

Адрес	Тип	Физ. подключение	Устройство
0.00	DI		Ультразвуковой датчик уровня P&F6GR6231-3RS00/ Тумблер 0.00 (в зависимости от положения переключателя «Режим»)
0.01	DI	ПЛК	Тумблер 0.01
0.02	DI	ПЛК	Тумблер 0.02
0.03	DI	ПЛК	Тумблер 0.03
0.04	DI	ПЛК	Тумблер 0.04
0.05	DI	ПЛК	Тумблер 0.05
0.06	DI	ПЛК	Тумблер 0.06
0.07	DI	ПЛК	Тумблер 0.07
0.09	DI	ПЛК	Поплавковый датчик уровня SQ3 (нижний)
0.10	DI	ПЛК	Поплавковый датчик уровня SQ2 (средний)
0.11	DI	ПЛК	Поплавковый датчик уровня SQ1 (верхний)
1	AI	Модуль, 0...10В	Емкостной датчик уровня LK3123
2	AI	Модуль, 4...20мА	Погружной датчик давления ПД100-ДГ
90	AI	ПЛК, 0...10В	Потенциометр СН90
91	AI	ПЛК, 4...20мА	Датчик гидростатического давления DMP331
100.00	DO	ПЛК	Пуск/останов работы преобразователя частоты, Светодиод 100.00
100.01	DO	ПЛК	Светодиод 100.01
100.02	DO	ПЛК	Включение электромагнитного клапана/ Светодиод 100.02
100.03	DO	ПЛК	Светодиод 100.03
100.04	DO	ПЛК	Светодиод 100.04
100.05	DO	ПЛК	Светодиод 100.05
100.06	DO	ПЛК	Светодиод 100.06
100.07	DO	ПЛК	Светодиод 100.07
101	АО	Модуль, 4...20мА	Задание частоты вращения электродвигателя
190	АО	ПЛК, 0...10В	Вольтметр СН190

Встроенные аналоговые входы и выходы ПЛК настраиваются следующим образом. Во-первых, при физическом подключении выбираются соответствующие клеммы в зависимости от того, какой используется сигнал – по напряжению или по току. Во-вторых, диапазон сигнала «уточняется» во вкладке Built-in AD/DA окна PLC Settings (рис. 1), которое можно открыть из дерева проекта программы CX-Programmer. *Необходимо отметить, что при загрузке программы в контроллер настройки (Settings) по умолчанию не включены в список обновления, поэтому для изменения конфигурации требуется поставить галочку в соответствующем пункте открывающегося окна (рис. 2).*

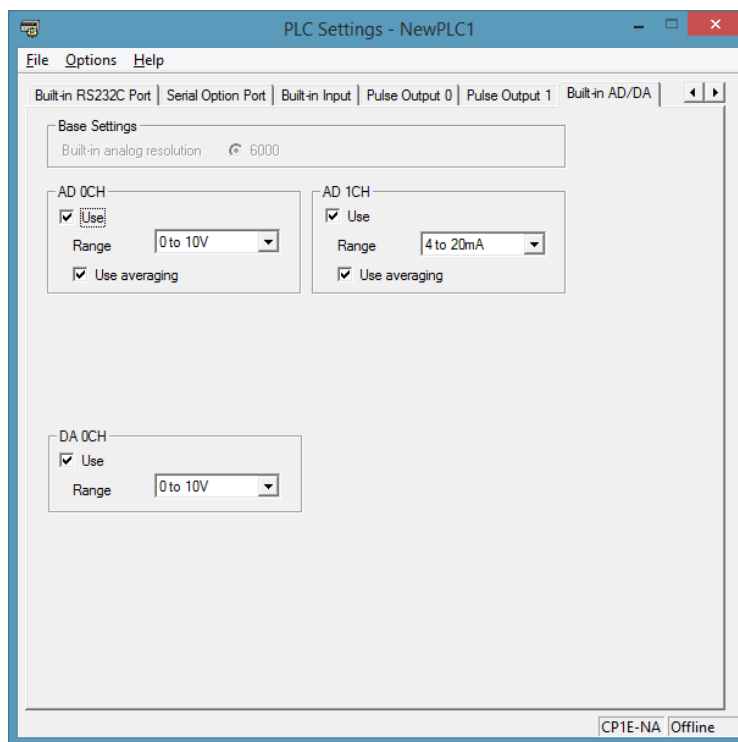


Рис. 1. Настройка встроенных аналоговых входов-выходов.

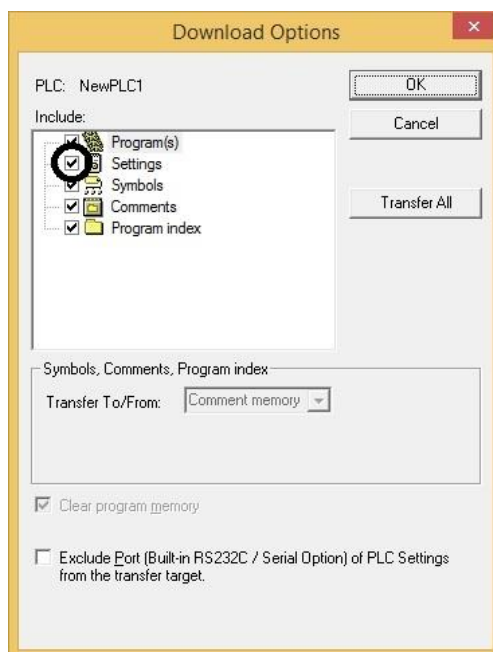


Рис. 2. Обновление настроек ПЛК при загрузке программы.

После записи новых настроек в ПЛК он должен быть перезагружен!

Настройка аналоговых входов и выхода модуля расширения производится программно, путем записи т.н. «кода диапазона» по адресу первого выходного слова модуля. В нашем случае это адрес 101. Код диапазона для модуля CP1W-MAD11 имеет следующий формат, рис. 3.



Рис. 3. Код диапазона модуля CP1W-MAD11.

Для каждого из двух входов и одного выхода закреплены по три бита, содержание которых определяет диапазон сигнала в соответствии с табл. 2.

Таблица 2. Содержание полей кода диапазона

Код диапазона	Диапазон сигнала аналогового входа 0	Диапазон сигнала аналогового входа 1	Диапазон сигнала аналогового выхода
000	-10...10В	-10...10В	-10...10В
001	0...10В	0...10В	0...10В
010	1...5В/4...20мА	1...5В/4...20мА	1...5В
011	0...5В/0...20мА	0...5В/0...20мА	0...20мА
100	–	–	4...20мА

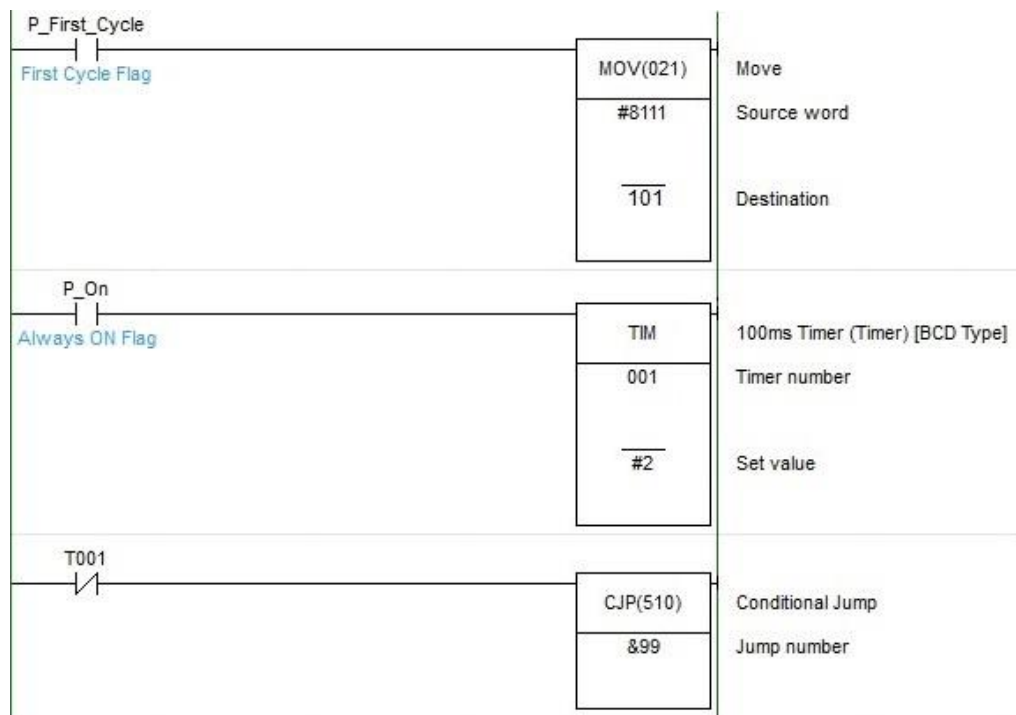
В нашем случае имеем (см. табл. 1) входы 0...10В и 4...20мА, выход – 4...20мА. Поэтому код диапазона будет следующим:

$$1000000(100)(010)(001) = (1000)(0001)(0001)(0001) = \#8111.$$

Код диапазона необходимо записать на первом цикле программы ПЛК, т.е. по «флагу» P_First_Circle. Однако после записи модулю требуется некоторое время (100 – 200 мс), чтобы провести инициализацию. В это время программа контроллера выполняться не должна. Для организации задержки следует использовать таймер. В «фирменной» программе «контакты» этого таймера помещены во все «цепи», поэтому они выполняются только по истечении указанного времени. Можно использовать другой, возможно, более эффективный подход, показанный на рис. 4.

Здесь задействованы команды CJR (Condition Jump – переход по условию) и JME (Jump End – «конец», точнее, точка, перехода) [4], которые позволяют «пропустить» выполнение «полезной программы» до срабатывания таймера. Параметр этих команд – просто номер перехода, он может быть любым числом в диапазоне от 0 до 127. Амперсанд (&) указывает на то, что это десятичное число (а не адрес и не шестнадцатеричное (#)).

До тех пор, пока не будет записан код диапазона, модуль аналоговых входов/выходов не приступит к преобразованию аналоговых сигналов. Пока не начато преобразование, результат преобразования содержит значение 0000, а на аналоговом выходе сохраняется уровень 0В или 0мА. По этой причине подключать преобразователь частоты к ПЛК следует только после запуска программы. В противном случае, обнаружив сигнал задания 0мА, преобразователь переходит в состояние ошибки. Сбросить ошибку можно с помощью кнопки STOP/RESET, предварительно отключив преобразователь от ПЛК. После инициализации модуля преобразователь можно вновь подключить к ПЛК.



< ПОЛЕЗНАЯ ПРОГРАММА >

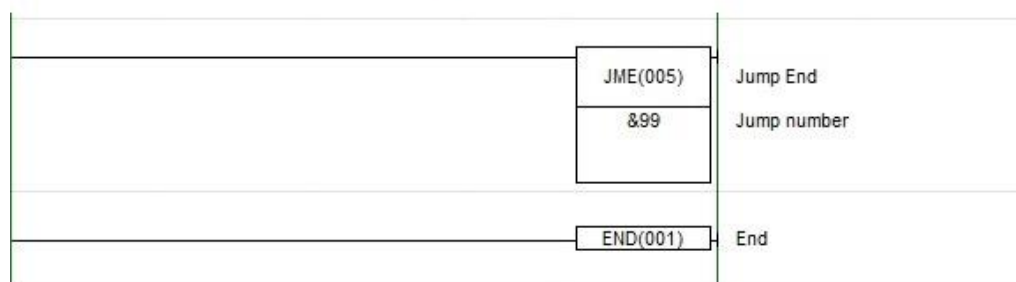


Рис. 4. Программная реализация инициализации модуля.

Сброс параметров модуля происходит только после отключения питания, поэтому повторная запись кода диапазона приводит не к «переинициализации» модуля, а к выдаче на аналоговый выход недопустимого значения. В этом случае ПЧ также переходит в состояние ошибки. Поэтому при каждом перезапуске программы следует временно отключать преобразователь от контроллера.

После инициализации модуля его переменным можно пользоваться так же, как и переменными самого ПЛК, применяя, например, команду MOV (21) с обращением по адресам 1, 2 для чтения и 101 для записи. Можно дать этим переменным имена, как это сделано в «фирменной» программе. Для этого следует использовать локальную «ветку» Symbols из дерева проекта.

Ниже описаны диапазоны изменения программных переменных аналогового ввода-вывода для сигналов 0...10В и 4...20мА, которые используются в лабораторном стенде.

Диапазоны входных аналоговых сигналов

Значениям напряжения в диапазоне от 0 до 10В соответствуют шестнадцатеричные значения от #0000 до #1770 (от 0 до 6000). Полный диапазон

значений: от #FED4 до #189C (от -300 до 6300). Для представления отрицательных значений напряжения используется дополнение до двух (рис. 5).

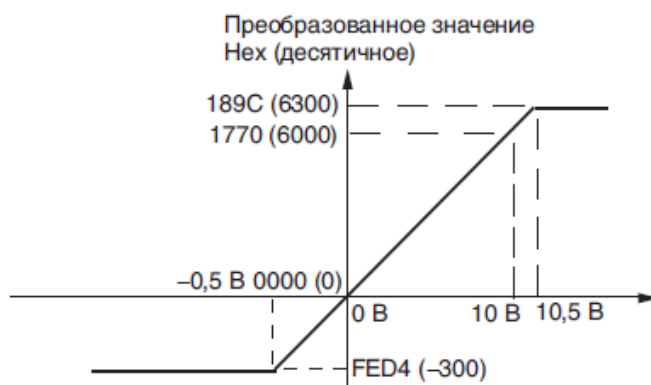


Рис. 5. Характеристика АЦП 0...10В.

Значениям тока в диапазоне от 4 до 20мА соответствуют шестнадцатеричные значения от #0000 до #1770 (от 0 до 6000). Полный диапазон значений: от #FED4 до #189C (от -300 до 6300). Для представления значений тока в диапазоне от 3,2 до 4 мА используется дополнение до двух (рис. 6). Если входной сигнал падает ниже уровня 3,2мА, срабатывает функция обнаружения разрыва цепи и в качестве результата преобразования устанавливается значение #8000.

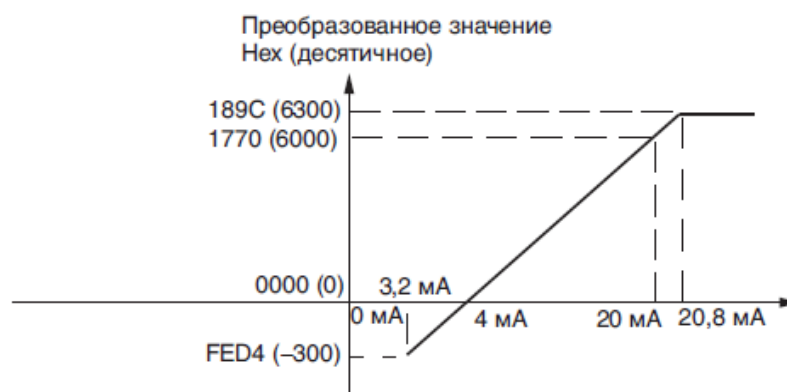


Рис. 6. Характеристика АЦП 4...20мА.

С помощью DIP-переключателя для входов может быть включена функция усреднения. Функция усреднения сохраняет в качестве результата преобразования среднее (текущее среднее) значение восьми последних входных значений. Данная функция позволяет сгладить изменения входного сигнала, происходящие в пределах короткого промежутка времени.

Диапазоны выходных аналоговых сигналов

Шестнадцатеричным значениям от #0000 до #1770 (от 0 до 6000) соответствуют значения напряжения в диапазоне от 0 до 10В. Полный диапазон сигнала: -0,5...10,5В. Для отрицательных значений напряжения используется дополнение до двух (рис. 7).

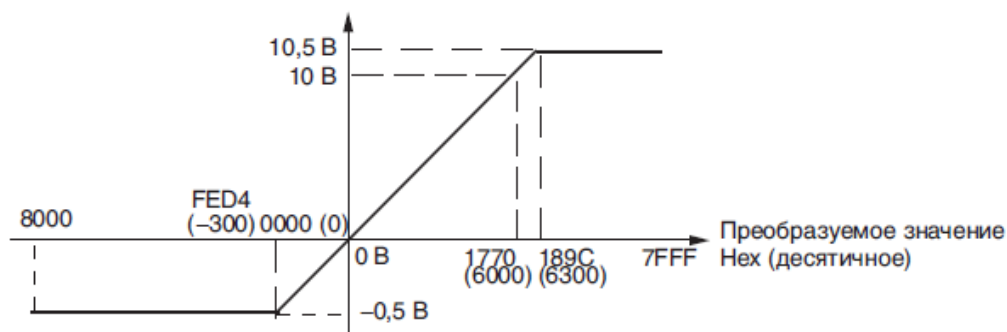


Рис. 7. Характеристика ЦАП 0...10В.

Шестнадцатеричным значениям от #0000 до #1770 (от 0 до 6000) соответствуют значения тока в диапазоне от 4 до 20мА. Полный диапазон выходного сигнала: 3,2...20,8мА (рис. 8).

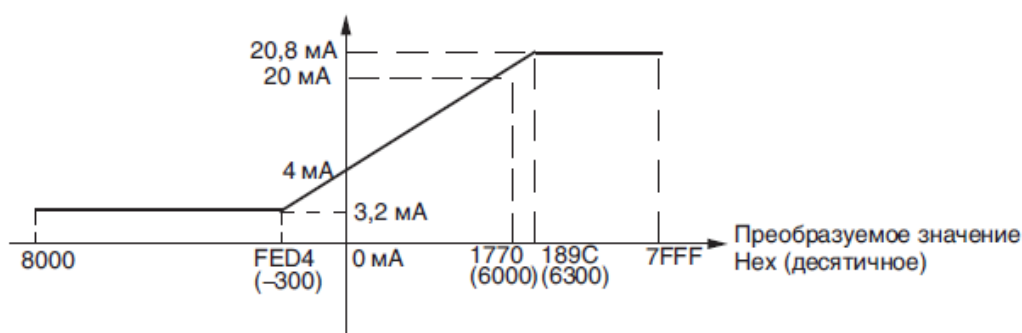


Рис. 8. Характеристика ЦАП 4...20мА.

1.2. Организация памяти

Внутренняя память ПЛК состоит из ППЗУ (EEPROM) и ОЗУ (RAM), между которыми происходит обмен данными, показанный на рис. 9 [3].

При отключении питания, а также в ряде других случаев (например, при перезагрузке) пользовательская программа и настройки ПЛК копируются в EEPROM, при включении они восстанавливаются в RAM. То же самое происходит с частью данных из области DM (этот процесс требует специальной настройки). Данные из других областей RAM в EEPROM не копируются и сохраняются только до тех пор, пока встроенный конденсатор не разрядится (согласно документации, 40 часов). ПЛК имеет отсек для установки дополнительной батареи питания, увеличивающей срок сохранности данных, однако на задействованном в лабораторной установке ПЛК батарея не установлена. Существует также так называемая *рабочая область* памяти W, которая всегда обнуляется сразу после отключения питания.

Настройка процесса загрузки программы производится во вкладке Startup/CPU Settings. Здесь можно предусмотреть принудительную очистку сохраняемых областей памяти и задать количество слов из области DM, копируемых в EEPROM (рис. 10). Помимо того, задается режим, в который переходит ПЛК при включении питания.

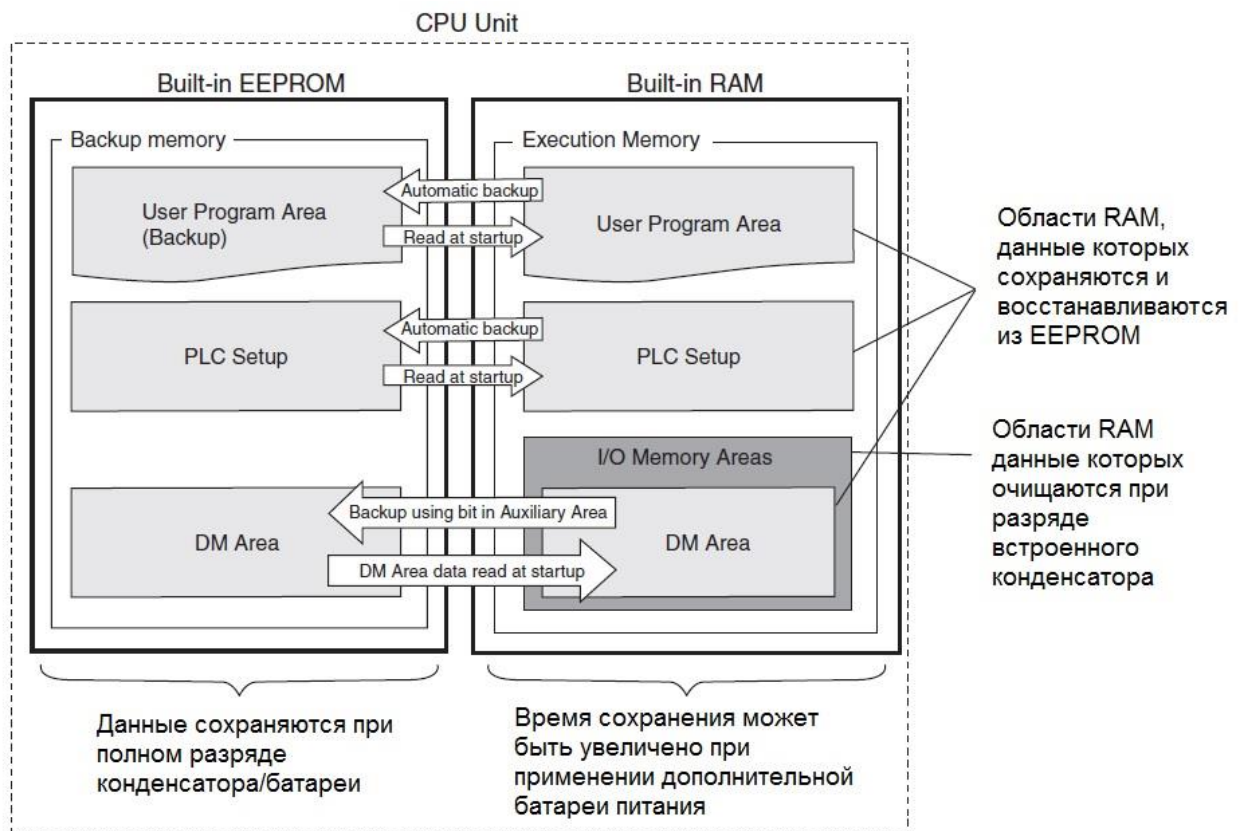


Рис. 9. Внутренняя память ПЛК.

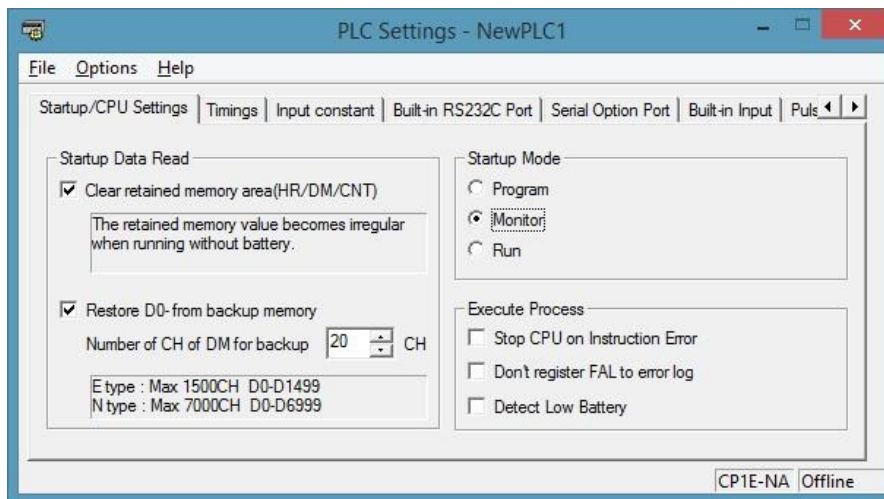


Рис. 10. Вкладка Startup/CPU Settings.

На рис. 11 показана общая схема памяти, так или иначе доступная пользовательской программе [3].

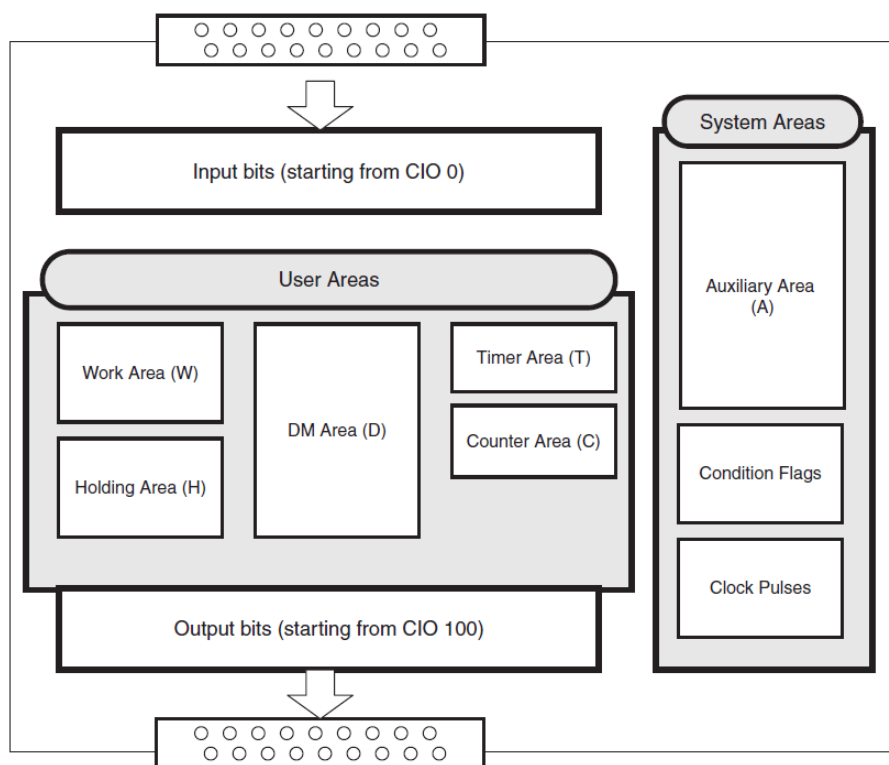


Рис. 11. Общая схема областей памяти RAM.

Область ввода-вывода CIO подробно рассмотрена в предыдущем пункте.

Work Area (W) – рабочая область. Может использоваться произвольным образом. Ограничения:

данные никогда не сохраняются при отключении питания;

недоступна при обмене данными по протоколу HOST LINK, используемому в SCADA-системе Trace Mode для связи с ПЛК.

Рекомендации по применению: для хранения вспомогательных, «недолго живущих» данных.

Holding Area (H) – область хранения. Может использоваться произвольным образом. Здесь рекомендуется хранить данные для обмена со SCADA-системой, особенно те, что требуют доступа к отдельным битам.

DM Area (D) – область данных. Может использоваться произвольным образом. Преимуществом этой области является возможность «резервного сохранения» данных в EEPROM (см. выше). Недостаток – невозможность обращения к отдельным битам.

Timer Area (T) – область памяти таймеров.

Counter Area (T) – область памяти счетчиков.

Auxiliary Area (A) – вспомогательная область. Относится к системной области, однако прикладная программа имеет к ней доступ. Слова и биты этой области имеют predetermined назначение. Например, слова A351...354 хранят информацию о дате и времени, A200.11 – флаг первого цикла (P_First_Cycle). Часть области допускает только чтение, другая – чтение и запись. В любом случае область A не может использоваться произ-

вольным образом, работать с ней можно только по назначению, тщательно изучив документацию.

Condition Flags – флаги условий. Автоматически устанавливаются системой по результатам выполнения команд. Например, P_ER – флаг ошибки (Error Flag), P_N – флаг отрицательного результата (Negative Flag). Для каждой команды закреплен свой список флагов, которые устанавливаются по результатам ее выполнения. Флаги можно использовать как контакты на лестничной диаграмме.

Clock Pulses – временные импульсы. Импульсы, автоматически генерируемые системным таймером. Например, P_1s – импульс, генерируемый раз в секунду. Импульсы доступны только для чтения. Как и флаги, их можно использовать в качестве контактов на лестничной диаграмме.

Все флаги и импульсы, а также часть битов области А имеют фиксированные имена, посмотреть которые можно в узле Symbols дерева проекта программы CX-Programmer.

Информация о размерах областей памяти приведена в табл. 3.

Таблица 3. Области памяти ПЛК CP1E-NA20DT-D

Область		Слово	Бит ("реле")		
			В CX-P		В CX-P
CIO Область	Область ввода/вывода	000 ... 199	0 ... 199	00000 ... 19915	0.00 ... 199.15
	Область последовательной связи ПЛК	200 ... 289 CH	200 ... 289	20000 ... 28915	200.00 ... 289.15
Рабочая область		W00 ... W099 CH	W00 ... W99	W0000 ... W9915	W0.00 ... W99.15
Область хранения		H00 ... H49 CH	H0 ... H49	H0000 ... H4915	H0.00 ... H49.15
Вспомогательная область		A000 ... A753 CH	A000 ... A753	A00000 ... A75315	A0.00 ... A753.15
Область DM		D00000 ... D08191	D0 ... D08191	-	-
Таймер		T000 ... T255	T0 ... T255	T000 ... T255	T0000 ... T0255
Счетчик		C000...C255	C000...C255	C000...C255	C0000...C0255

1.3. Инструкции и операнды

ПЛК CP1E в основном программируются на языке релейно-контактных схем, часто называемом языком «лестничных диаграмм». Имеется также возможность использовать язык мнемоник, но с ним работать труднее.

Помимо контактов и катушек, схемы содержат и более сложные команды [4]. С помощью операндов указываются параметры команды (ячейки «таблиц») (рис. 12).

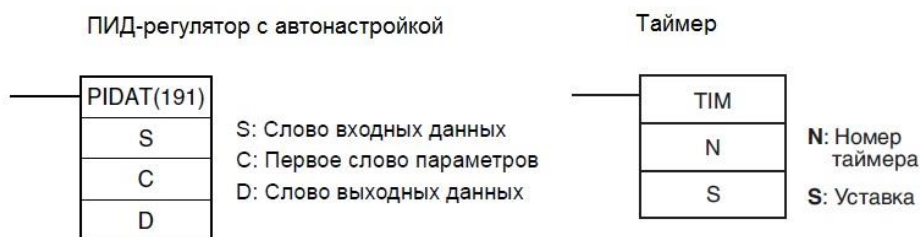


Рис. 12. Примеры команд.

Содержание операндов, естественно, зависит от команды, их общее назначение приведено в табл. 4.

Таблица 4. Назначение операндов

Типы операндов		Символическое обозначение	Описание	
Источник	Указывает адрес, по которому должны быть прочитаны данные, или константу.	S	Входной операнд	Любой входной операнд, кроме управляющих данных.
		C	Управляющие данные	Составные входные данные, значения и состояния битов которых определяют ход выполнения команды.
Выходной операнд (результат)	Указывает адрес, по которому должны быть записаны данные.	D (R)	---	
Число	Указывает определенное число, которое должно использоваться в команде (например, номер для перехода или номер подпрограммы).	N	---	

В качестве операндов могут выступать константы, адреса, косвенные адреса (указатели) и т.д. Ниже рассмотрены наиболее часто встречающиеся варианты задания операндов.

16-битные константы (см. табл. 5) могут иметь знак или не иметь его. Если константа имеет знак, то ее обычно записывают в десятичном формате с обязательным указанием знака («+» обязателен!). Константа без знака в десятичном формате снабжается префиксом «&», в шестнадцатеричном – префиксом «#». Если не поставить знак или префикс, компилятор воспримет введенное число не как константу, а как адрес (очень распространенная ошибка).

Таблица 5. 16-битовые константы


Данные	Операнд	Формат данных	Символ	Диапазон	Диапазон применения
16-битовая константа	Любые двоичные значения или ограниченный диапазон двоичных значений	Двоичное без знака	#	#0000...#FFFF	<div style="border: 1px solid black; padding: 2px; width: fit-content;">MOV(021)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">#5A</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">D100</div>
		Десятичное со знаком	±	-32768...+32767	<div style="border: 1px solid black; padding: 2px; width: fit-content;">+(400)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">D200</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">-128</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">D300</div>
		Десятичное без знака	&	&0...&65535	<div style="border: 1px solid black; padding: 2px; width: fit-content;">CMP(020)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">D400</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">&999</div>
	Любые двоично-десятичные значения или ограниченный диапазон двоично-десятичных значений	BCD	#	#0000...#9999	<div style="border: 1px solid black; padding: 2px; width: fit-content;">-B(414)</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">D500</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">#2000</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-top: 2px;">D600</div>

Некоторые команды работают не с двоичными, а с двоично-десятичными числами (BCD), такие числа также сопровождаются префиксом

«#». В отличие от «настоящих» шестнадцатеричных чисел в них присутствуют только цифры от 0 до 9. BCD-константа, по сути, правильным образом «расставляя» биты, выглядит, тем не менее, как десятичное число.

Указание адресов битов и слов показано в таблице 6.

Таблица 6. Адреса битов и слов

Операнд	Описание	Представление	Примеры применения
Указание адресов битов	<p>Для обращения к биту (биту входа) указываются непосредственно номер слова и номер бита.</p> <p>□□□□.□□</p> <p>Номер бита (00...15) Адрес слова</p> <p>Примечание. Для обращения к флагам завершения таймеров/счетчиков и для обращения к текущим значениям используются одни и те же адреса. Для флага задачи выделен только один адрес.</p>	<p>1.02</p> <p>Номер бита: 02 Номер слова: CIO 1</p>	<p>1.02</p> 
Указание адресов слов	<p>Для обращения к 16-битовому слову указывается непосредственно номер слова.</p> <p>□□□□</p> <p>Адрес слова</p>	<p>3</p> <p>Номер слова: 3</p> <p>D200</p> <p>Номер слова: D200</p>	

При обращении ко всем областям памяти, кроме CIO, буквенный префикс обязателен. Слова CIO адресуются «чистыми» номерами. Обратиться к отдельному биту слова можно во всех областях памяти, кроме DM. В области DM минимальной адресуемой единицей является 16-разрядное слово.

Другой особенностью данной области является то, что в ней можно использовать косвенную адресацию (указатели). Адрес представляет собой смещение от начала области и может быть записан как в двоичном, так и в двоично-десятичном виде. В первом случае для обращения по этому адресу используется префикс «@», во втором – «*» (табл. 7).

Многие команды ПЛК имеют несколько *вариантов выполнения*.

«Обычный» вариант предполагает выполнение команды на каждом цикле контроллера, пока действует *условие выполнения* (пока до команды «доходит ток»). Часто предпочтительным или даже необходимым является другой вариант: команда должна выполняться один раз при возникновении условия. Тогда перед именем команды необходимо поставить знак «@» (табл. 8). Возможны также ситуации, когда нужно выполнить команду «напоследок», когда условие снимается. В этом случае применяется префикс «%».

Таблица 7. Косвенная адресация слов в области DM

<p>Косвенное указание адресов области DM в двоичном коде</p>	<p>В качестве адреса указывается величина смещения от начала области. Двоичное значение (00000...32767), содержащееся по указанному адресу, является адресом слова в памяти данных (DM). Чтобы указать косвенный адрес в двоичном формате, впереди адреса следует добавить символ @.</p> <p>@D□□□□□</p> <p>↓</p> <p>Содержимое □□□□□ 00000...32767 (0000...7FFF hex)</p> <p>↓</p> <p>D □□□□□</p>	<p>@D300</p> <p>0 1 0 0 Содержимое</p> <p>Hex: 256</p> <p>↓</p> <p>Указывает D256.</p> <p>— Добавляется символ @.</p>	<p>MOV(021)</p> <p>#1</p> <p>@D300</p>
<p>Косвенное указание адресов области DM в двоично-десятичном коде</p>	<p>В качестве адреса указывается величина смещения от начала области. Двоично-десятичное значение (0000...9999), содержащееся по указанному адресу, является адресом слова в памяти данных (DM). Чтобы указать косвенный адрес в двоично-десятичном формате, спереди адреса следует добавить символ (*).</p> <p>*D□□□□□</p> <p>↓</p> <p>Содержимое □□□□□ 00000...9999 (BCD)</p> <p>↓</p> <p>D □□□□□</p>	<p>*D200</p> <p>0 1 0 0 Содержимое</p> <p>BCD: 100</p> <p>↓</p> <p>Указывает D100.</p> <p>— Добавляется символ (*).</p>	<p>MOV(021)</p> <p>#1</p> <p>*D200</p>

Таблица 8. Варианты выполнения команд

Варианты выполнения	Символ	Описание
<p>Определение направления переключения (различение фронта)</p>	ВКЛ	<p>@</p> <p>Команда выполняется по включению (т. е. по положительному фронту) условия выполнения.</p>
	ВЫКЛ	<p>%</p> <p>Команда выполняется по выключению (т. е. по отрицательному фронту) условия выполнения.</p>
<p>Мгновенное обновление</p>	!	<p>Обновление указанных операндами данных в области ввода/вывода или слов специального модуля ввода/вывода происходит при выполнении команды.</p>

Если команда работает с данными области ввода-вывода, то «обычный» вариант выполнения подразумевает, что настоящее обновление ввода/вывода произойдет только в конце контроллерного цикла, когда будет пересчитана вся программа. В некоторых случаях требуется, чтобы обновление происходило немедленно. При таком варианте вызова имя команды снабжается префиксом «!».

В общем случае варианты можно комбинировать, выставляя более одного префикса.

По результатам выполнения команд выставляются *флаги условий*.

Флаги условий используются совместно всеми командами и изменяют свое состояние в пределах цикла в зависимости от результатов выполнения отдельных команд. Следовательно, на разветвленном выходе с одним и тем же условием выполнения флаги условий следует располагать сразу же после команды, чтобы они отражали результаты ее выполнения (рис. 13). Не следу-

ет подключать флаг условия непосредственно к шине, поскольку в этом случае он будет отражать результаты выполнения других команд.

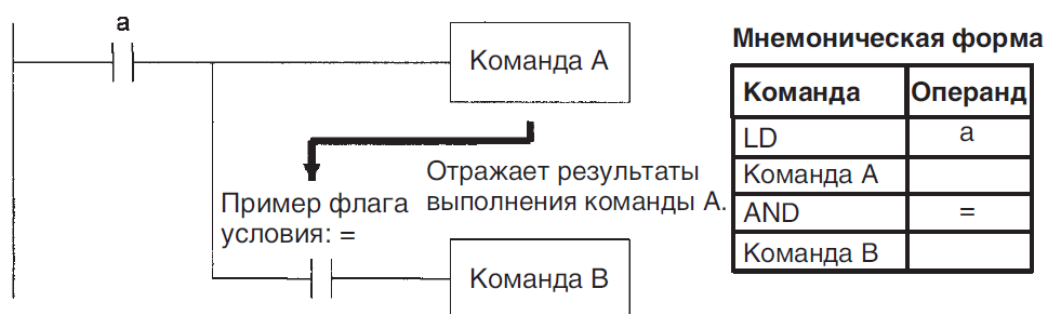


Рис. 13. Использование флагов условий.

Выполнение команды В зависит от результатов выполнения команды А, при этом для команд А и В используется общее условие выполнения (а). В этом случае команда В выполняется в соответствии с состоянием флага условия, только если была выполнена команда А.

Ниже рассмотрены некоторые флаги условий.

Флаг ошибки ER устанавливается при особых обстоятельствах, – например, когда для команды указан ошибочный операнд. При установленном флаге ER команда не выполняется. Перечень условий, которые приводят к включению флага ER, можно найти в описаниях отдельных команд. С помощью параметра в области настроек ПЛК можно указать, должно ли при включении флага ER прекратиться выполнение программы. По умолчанию включение флага ER не останавливает работу.

Флаг равенства EQ – это временный флаг для всех команд (кроме случая равенства операндов в командах сравнения). Он автоматически устанавливается системой и изменяется в процессе работы. Имеется также и *флаг неравенства NE*.

Флаг переноса CY используется в командах сдвига данных, в командах сложения и вычитания со входом переноса, в командах сложения и вычитания со входом заема, в командах ПИД-регулирования и др.

Флаги сравнения LT, LE, GT, GE используются в командах, выполняющих операции сравнения, а также в командах LMT, BAND, ZONE, PIDAT и др.

Флаг отрицательного значения N устанавливается, когда старший (крайний левый) бит слова результата выполнения определенных команд содержит «1», и сбрасывается без каких-либо условий для другой команды.

Флаг переполнения OF устанавливается математическими командами, когда результат операции не укладывается в разрядную сетку.

2. ЛАБОРАТОРНЫЕ РАБОТЫ

2.1. Система ручного и дистанционного управления. Автоматическое управление по сигналам дискретных датчиков

Постановка задачи

Система должна обеспечивать возможность ручного и дистанционного управления водяным насосом (с воздействием на преобразователь частоты) и сливным клапаном, а также осуществлять автоматическое управление насосом и клапаном по программе, изложенной далее.

В ручном режиме управление производится с помощью тумблеров 0.01 – 0.07 (тумблер 0.00 задействовать нельзя, поскольку он физически отключен от ПЛК). Контроль уровня выполняется по светодиодам SQ1 – SQ3 и непосредственно визуально.

Назначение тумблеров:

0.01 – пуск системы. При включенном состоянии разрешаются все дальнейшие действия. Если тумблер выключен, система находится в состоянии покоя: ПЧ – в режиме «стоп», сливной клапан закрыт;

0.02 – запуск программы автоматического управления (начинается со слива воды);

0.03 – переход в режим дистанционного управления со SCADA-системы. Тумблер имеет больший приоритет, чем 0.02, т.е. при одновременном включении управление все равно передается SCADA-системе;

0.04 – сигнал запуска ПЧ;

0.05, 0.06 – задание частоты по схеме:

НЕ 0.05 и НЕ 0.06 – 0 Гц;

0.05 и НЕ 0.06 – 16,7 Гц;

0.05 и 0.06 – 33,3 Гц;

НЕ 0.05 и 0.06 – 50 Гц;

0.07 – включение сливного клапана.

В дистанционном режиме управление и контроль осуществляются посредством элементов графического интерфейса SCADA-системы Trace Mode на экране компьютера. Примерный вид окна управления – на рис. 14.

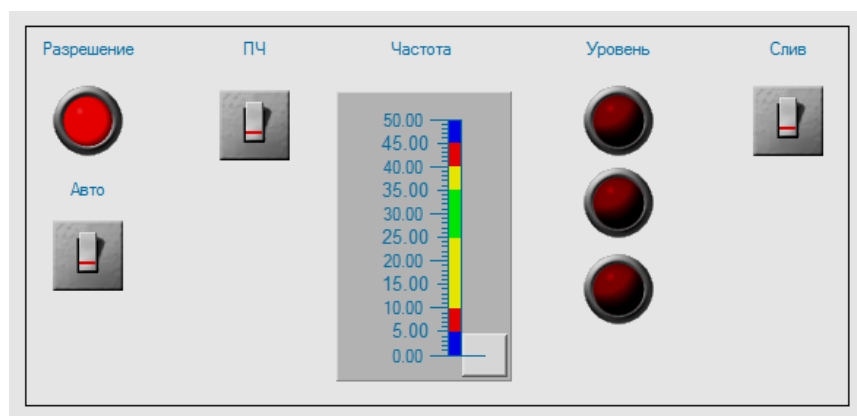


Рис. 14. Примерный вид окна SCADA.

Индикатор «Разрешение» загорается зеленым светом, когда SCADA-система допущена к управлению процессом. С помощью переключателя «Авто» запускается программа автоматического управления. Переключатель «ПЧ» подает команду пуска преобразователю частоты, а ползунок «Частота» служит одновременно для контроля и задания частоты. Индикаторы «Уровень» загораются при срабатывании соответствующих поплавковых датчиков. Переключатель «Слив» открывает сливной клапан.

В режиме автоматического управления происходит циклическое наполнение и опустошение резервуара водой по сигналам поплавковых датчиков. Процесс начинается со слива воды до момента осушения нижнего датчика. Далее, после выдержки времени 3с, включается насос с максимальной производительностью. При срабатывании среднего датчика скорость вращения двигателя насоса снижается вдвое. При срабатывании верхнего датчика насос отключается. Цикл повторяется после выдержки времени 3с.

Указания по написанию программы ПЛК

1. Программа должна инициализировать модуль расширения CP1W-MAD11 (см. п. 1.1).

2. Программа должна представлять собой конечный автомат [5] с определенным набором состояний, логикой смены состояний и логикой управления выходами в зависимости от входов и состояний. В качестве ориентира можно использовать следующий набор состояний:

- 0 – «Стоп»;
- 1 – «Ручное управление»;
- 2 – «Дистанционное управление»;
- 31 – «Автоматическое управление – слив»;
- 32 – «Автоматическое управление – наполнение».

Для хранения текущего состояния можно задействовать какое-либо слово, – например, из области памяти Н (пусть это будет Н0). На рис. 10 показаны переходы между состояниями.

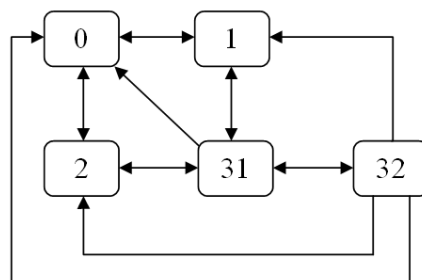


Рис. 15. Состояния и переходы.

Большинство переходов осуществляется при переключении тумблеров и не требует проверки текущего состояния. Так, переключение в состояние «Стоп» происходит при отключении тумблера 0.01, вне зависимости от текущего состояния (рис. 16).

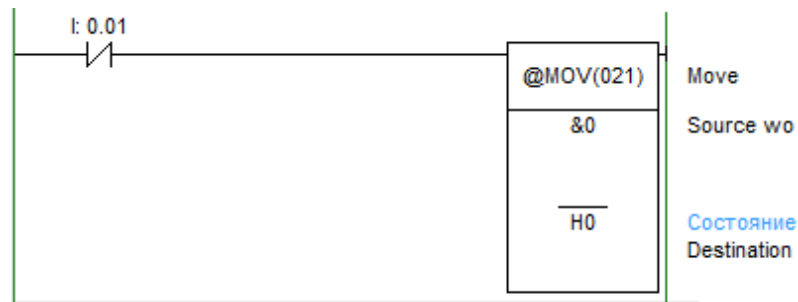


Рис. 16. Переход в состояние «Стоп».

Префикс @ перед командой MOV означает, что она будет выполняться по переднему фронту входного сигнала, т.е. *один раз* при отключении тумблера, а не *постоянно*, пока тумблер отключен. Такой подход позволяет ускорить выполнение кода и решает определенные проблемы при смене состояний.

Некоторые переходы, помимо контроля положения аппаратных тумблеров, требуют контролировать состояние переключателя «Авто» на экране SCADA. Протокол обмена SCADA и ПЛК обеспечивает передачу слов, но не битов. В простейшем случае под переключатель можно отвести целое слово, – например, N1, которое будет принимать значения 0 или 1 в зависимости от положения переключателя. Сравнить значение слова на равенство можно с помощью команды =(300), например:

```
= N1 &1 // N1==1?
```

Такой блок непосредственно вставляется в цепь программы.

Более эффективный подход состоит в том, чтобы отвести отдельные биты единственного слова под передачу всех входных дискретных сигналов (в нашем случае формируемых тремя переключателями экрана SCADA). Тогда эти биты можно «анализировать» с помощью обычных контактов. Однако не все области памяти ПЛК допускают обращение к отдельным битам, именно поэтому рекомендуется использовать область N, а не D, как это сделано в фирменном программном обеспечении. Управление отдельными битами в TRACE MODE осуществляется с помощью операции побитового исключающего ИЛИ (см. справку по графическому элементу «Выключатель»).

В любом случае, какой бы подход ни был принят, необходимо в первом цикле контроллера обнулить все принимаемые от SCADA переменные, иначе они будут содержать неопределенные значения. Следует также помнить, что по умолчанию в TRACE MODE отправка переменных по сети осуществляется по факту их изменения, а также при старте системы (при соответствующей настройке).

Переходы между состояниями 31 и 32 производятся «внутри» этих состояний и связаны с проверкой срабатывания таймера, запускаемого по сигналам соответствующих поплавковых датчиков.

Альтернативный и более эффективный способ хранения информации о текущем состоянии автомата состоит в том, чтобы для каждого состояния

выделить отдельный бит, – например:

Н0.00 – «Стоп»;

Н0.01 – «Ручное управление»;

Н0.02 – «Дистанционное управление»;

Н0.03 – «Автоматическое управление – слив»;

Н0.04 – «Автоматическое управление – наполнение».

В этом случае проверку состояния можно выполнять с помощью обычных контактов. Однако при таком подходе существует опасность того, что два или более бита по ошибке будут установлены одновременно, и тогда поведение программы станет непредсказуемым. Избежать этого можно, принудительно сбрасывая при переходах бит «прошлого» состояния (в общем случае всех или всех возможных «прошлых» состояний). Сброс бита осуществляется с помощью команды `RSET`. Для подстраховки можно также перед установкой «нового» бита обнулить все слово командой `MOV`.

Логiku управления выходами на языке лестничных диаграмм можно реализовать двумя путями.

«Классический» способ состоит в том, что каждый выход представлен в программе единственной «катушкой», активация которой происходит по многочисленным условиям, которые включают проверку состояний и входов (рис. 17).

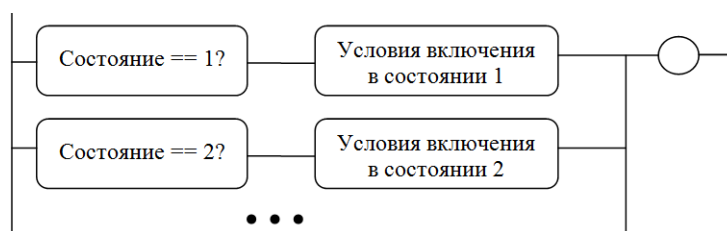


Рис. 17. Управление выходом.

Такой подход имеет очевидные недостатки: при большом числе состояний и условий цепи могут получиться достаточно громоздкими, а выполнение программы замедляется, поскольку на каждом цикле ей требуется пересчитывать все цепи.

Другой способ предполагает реализацию технологии `switch-case`, широко применяемой в текстовых языках, средствами языка лестничных диаграмм. Рассмотрим гипотетический код на языке С:

```
switch (State)
{
1: <Действия 1>; break;
2: <Действия 2>; break;
...
}
```

Здесь определенный набор действий выполняется в зависимости от состояния `State`.

На языке лестничных диаграмм программную конструкцию можно реализовать с помощью рассмотренных в п. 1.1. команд `SJP` и `JME` (рис. 18).

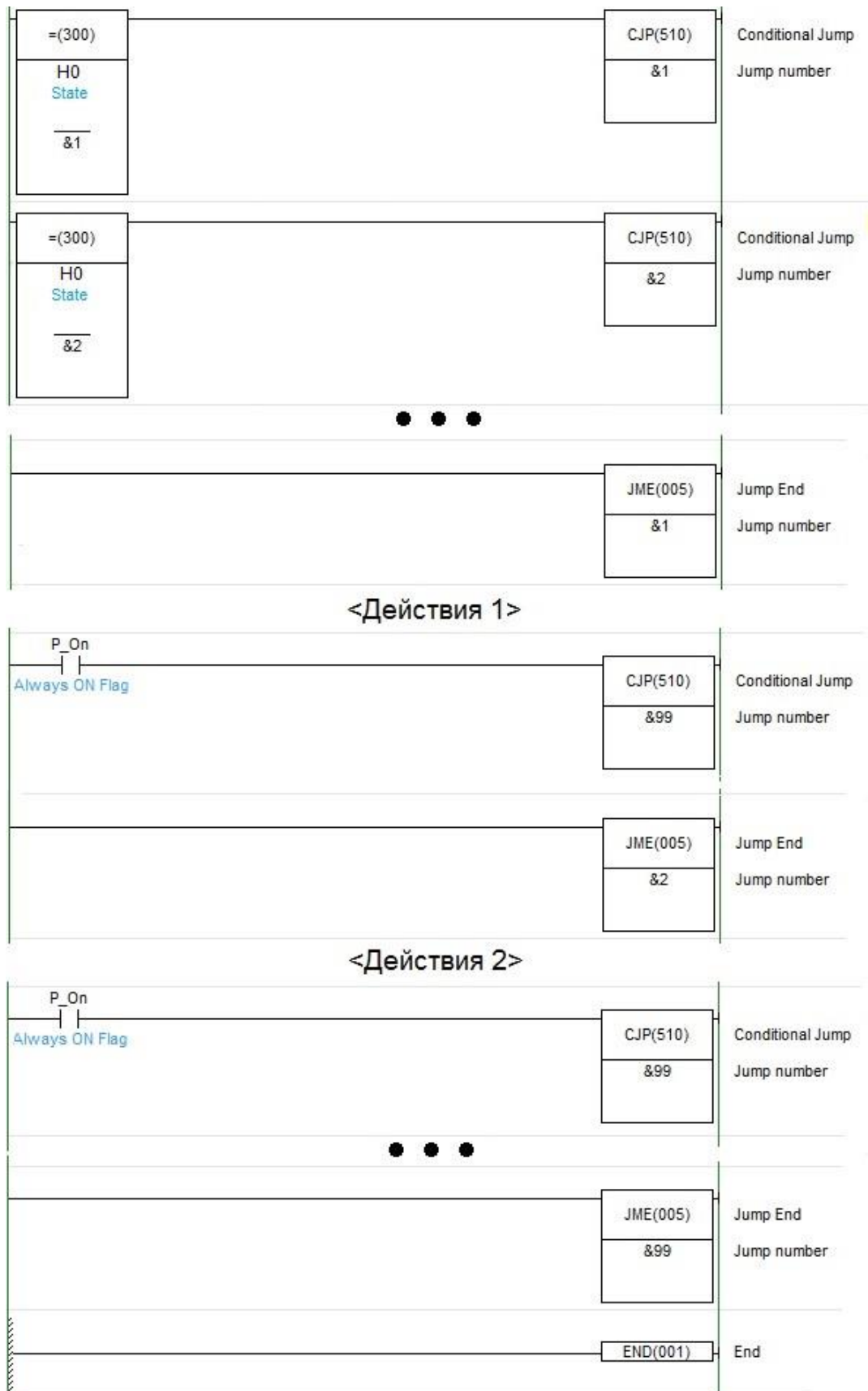


Рис. 18. Реализация конструкции switch-case.

Недостатком подхода является значительное увеличение длины программы, поскольку теперь цепи управления выходами необходимо поместить

в каждую «case»-секцию. (Тут следует отметить, что CX-Programmer при создании «дублирующих» катушек каждый раз «выносит предупреждения», которые нужно игнорировать).

Преимуществом является более высокая скорость выполнения программы, поскольку теперь ненужные в данный момент большие фрагменты кода просто не выполняются. Выполняются только те фрагменты, которые отвечают за текущее состояние. Программа легче составляется и документируется.

Поскольку в нашем случае разрабатываемая программа достаточно проста, при ее написании можно воспользоваться как «классической» технологией, так и технологией switch-case.

Указания по разработке SCADA-системы

1. Поскольку обмен с ПЛК будет выполняться по последовательному интерфейсу, необходимо в узле проекта создать сначала группу COM-порты, а потом и сам COM-порт, после чего произвести его настройку (рис. 19) .

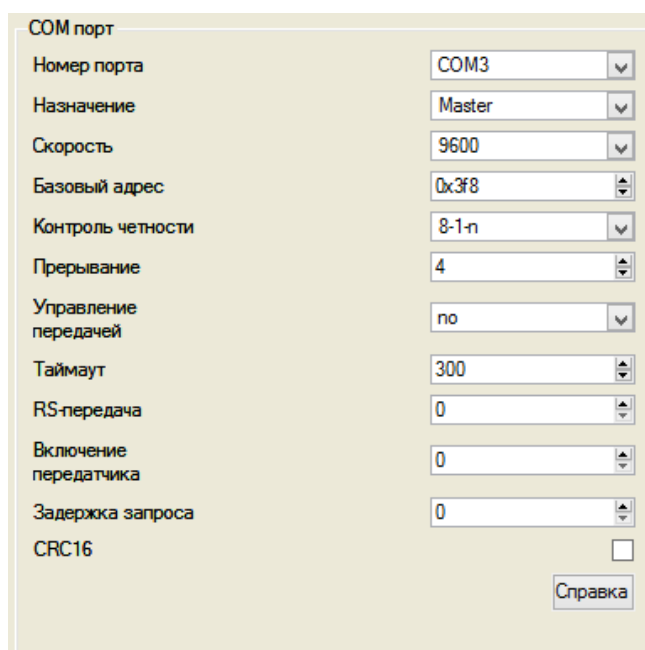


Рис. 19. Настройка COM-порта в Trace Mode.

В нашем случае мы имеем виртуальный порт COM 3, физически привязанный к USB. Необходимо согласовать параметры работы этого порта с параметрами порта RS232 ПЛК: скорость передачи и формат передаваемого слова (рис. 20).

2. При создании «Источников/приемников» (впрочем, как и других компонентов проекта) можно «в качестве образца» воспользоваться фирменным проектом, однако рекомендуется вместо каналов типа DM использовать каналы типа HR, адресующие область памяти N ПЛК. Как уже было сказано, в этом случае появляется возможность применения к соответствующим переменным битовых инструкций.

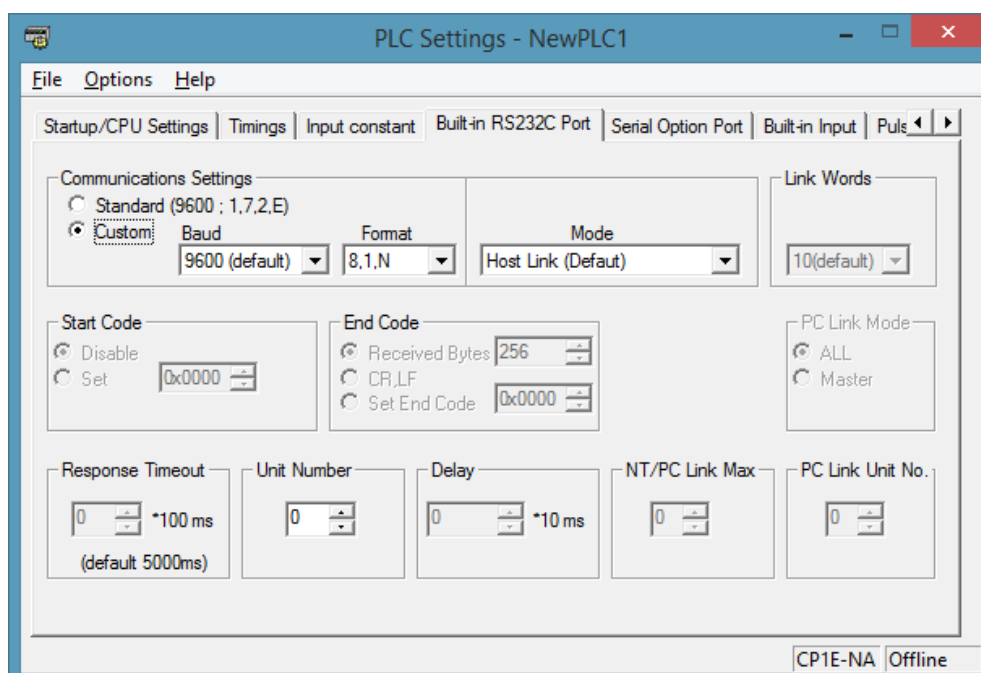


Рис. 20. Настройка COM-порта ПЛК.

3. Все входные для SCADA сигналы нужно провести через «Каналы», иначе корректная передача сигналов не гарантируется. Для создания канала достаточно перетащить Источник/Приемник в папку «Каналы», при этом канал соответствующего типа будет создан и привязан к Источнику/Приемнику автоматически. В дальнейшем графические элементы необходимо привязать к реальным значениям каналов, а не к Источникам/Приемникам.

4. Передача сигнала задания частоты должна производиться в двух направлениях: чтение и запись. Поэтому требуется создать два Источника/Приемника (точнее, источник и приемник). Чтение выполняется непосредственно из выходного слова ПЛК 101 (канал IR), запись – в какое-нибудь слово, – например, H2 (канал HR). Необходимо также создать каналы Trace Mode для чтения и записи (перетаскивая Источники/Приемники в папку «Каналы»). Каналы требуют настройки (масштабирования), поскольку SCADA оперирует значением частоты от 0 до 50 Гц, а ПЛК – значениями управляющего слова от 0 до 6000. *Необходимо также помнить о том, что запись должна осуществляться не в реальное, а во входное значение выходного канала.*

5. В фирменном проекте Trace Mode обработка сигналов поплавковых датчиков осуществляется некорректно, поскольку полученное от ПЛК слово 0 (входные дискретные сигналы) анализируется как целое число, а не побитно. Это, в частности, не позволяет выявить на экране аварию нижнего или среднего поплавкового датчика (соответствующие индикаторы будут все равно светиться, если сработал средний или верхний). Следует исправить ошибку, перейдя к побитовой обработке и применив вид индикации «Arg & Конст».

Указания по запуску системы

1. Для обеспечения взаимодействия ПЛК со SCADA-системой необходимо, чтобы:

присутствовало физическое соединение компьютера и ПЛК посредством кабеля USB-RS232 (он содержит встроенный преобразователь интерфейса);

ПЛК находился в режиме MONITOR (в режиме RUN запись данных в него невозможна, только чтение);

настройки портов компьютера и ПЛК соответствовали друг другу (см. выше) *Не забывайте при загрузке программы указать также необходимость загрузки настроек, если они были обновлены (см. п. 1.1).*

2. После того как система опробована, продемонстрируйте ее работу преподавателю.

2.2. Система измерения уровня. Обработка сигналов измерительных преобразователей

Постановка задачи

Система должна обеспечивать измерение уровня посредством всех четырех преобразователей, установленных на стенде, с отображением как числовых значений, так и графиков изменения сигналов. Примерный вид окна SCADA-системы показан на рис. 21.



Рис. 21. Окно SCADA-системы (в действии).

Окно содержит средства управления процессом, созданные в предыдущей работе. Под ними выводятся значения уровня, полученные путем обработки сигналов четырех измерительных преобразователей (в сантиметрах) и индикаторы достоверности показаний, которые меняют цвет на красный в случае, если сигналы выходят из допустимого диапазона. Элемент визуализации

зации «Тренд», расположенный справа, показывает графики изменения сигналов во времени.

Указания по написанию программы ПЛК

1. Программа должна инициализировать модуль расширения CP1W-MAD11 (см. п. 1.1).

2. Перед написанием программы следует изучить технические характеристики измерительных преобразователей и провести эксперименты, целью которых является установление диапазонов измерения для каждого канала.

Основные технические характеристики преобразователей приведены в [1]. Более подробные сведения представлены в электронных материалах на жестком диске персонального компьютера, входящего в состав лабораторной установки. При необходимости можно также обратиться к соответствующим Интернет-ресурсам. Следует обратить внимание на одно обстоятельство: материалы, посвященные погружному преобразователю давления, описывают целую серию преобразователей ПД100И. В настоящее время на стенде установлен измеритель ПД100И-ДГ0,04-167-0,5.3 с диапазоном измерения от 0 до 4 м вод. ст.

Диапазоны измерения каналов, которые требуется установить в результате экспериментов, должны быть привязаны к отсчетам измерительной линейки, находящейся на резервуаре. Погружной и гидростатический преобразователи, измеряющие давление, способны вести измерения начиная с нулевой отметки, тогда как емкостный и ультразвуковой преобразователи, хотя и рассчитаны на диапазон от 0 до 400 мм, измерять малые уровни не могут, поскольку смонтированы значительно выше отметки 400 мм. С другой стороны, имеются ограничения по максимуму. Это касается емкостного (по понятным причинам) и ультразвукового преобразователя, показания которого при больших уровнях становятся нестабильными (видимо, по причине нарушения условий отражения звуковой волны).

Результаты экспериментов необходимо свести в табл. 9.

Таблица 9. Результаты экспериментов по определению диапазонов измерения каналов

Измерительный преобразователь	Минимум			Максимум	
	Линейка	Физический сигнал	Программный сигнал	Физический сигнал	Программный сигнал
Ультразвуковой	? мм	? Гц	? ед	? Гц	? ед
Емкостный	? мм	? В	? ед	? В	? ед
Погружной	0 мм	? мА	? ед	? мА	? ед
Гидростатический	0 мм	? мА	? ед	? мА	? ед

Раздел таблицы «Максимум» не имеет столбца «Линейка», поскольку для всех преобразователей верхняя граница измерений находится выше предельной отметки на линейке (400 мм).

Значения физических сигналов фиксируются по показаниям измерителей стенда: частотомеру, вольтметру и двум миллиамперметрам. Значения

программных сигналов определяются в программе CX-Programmer. За исключением ультразвукового ИП, программные сигналы – это сигналы на входах модуля расширения и самого ПЛК. Их значения доступны по адресам 1 (емкостный ИП), 2 (погружной ИП), 91 (гидростатический ИП).

Программный сигнал по частоте (ультразвуковой ИП) формируется скоростным счетчиком ПЛК, подсчитывающим импульсы, поступающие на вход 0.0. Ультразвуковой ИП может формировать частотный сигнал в двух диапазонах от 40 до 400Гц (нижний диапазон) и от 160 до 1600Гц (верхний диапазон). Выбор диапазона определяется сигналом на так называемом «входе обучения» (Teach-in). На стенде используется верхний диапазон.

Скоростной счетчик должен быть активирован и настроен во вкладке Build-in Input окна PLC Settings (рис. 22). В нашем случае мы имеем простейший однотактный (с единственным счетным входом 0.0) инкрементный счетчик, работающий в циклическом режиме с программным сбросом.

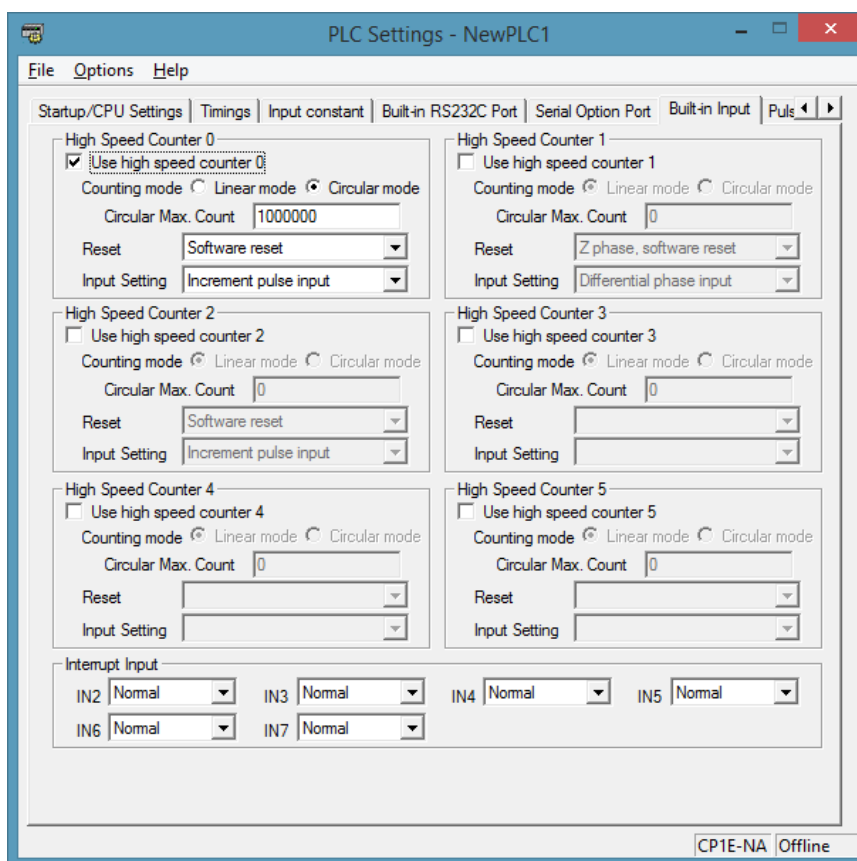


Рис. 22. Настройка скоростного счетчика.

После записи настроек в ПЛК скоростной счетчик автоматически приступает к работе. Каких-либо дополнительных действий по его настройке и запуску из программы не требуется.

Для работы со скоростными счетчиками (и импульсными выходами) используется команда PRV (881). Команда позволяет получать различную информацию от счетчиков и выходов и имеет два настраиваемых параметра: P – Port specifier (спецификатор порта) и C – Control data (управляющее слово).

В нашем случае $P = \#10$, что означает обращение к скоростному счетчику 0, $C = \#33$, это запрос частоты следования импульсов с интервалом 1 с (в «фирменной» программе установлено значение $C = \#23$ и интервал измерения равняется 100 мс, вследствие чего частота измеряется с погрешностью 10 Гц).

Результат выполнения команды `PRV` записывается в два слова, адрес первого (младшего) из которых передается команде в качестве параметра `D`. Здесь можно использовать любой адрес, – например, из области памяти `W`. Поскольку пределы изменения частоты в нашем случае небольшие, результат целиком укладывается в первое слово, а второе будет равно нулю.

Более подробную информацию о команде `PRV` можно получить из справочной системы.

Наблюдать значения переменных в программе `CX-Programmer` можно различными способами, в частности непосредственно на лестничной диаграмме, если эти переменные задействованы в программе, или с помощью средства просмотра и редактирования памяти `Memory` из дерева проекта. Однако наиболее удобным является средство `Time Chart Monitor`, доступное из меню `PLC`. Здесь можно просматривать численные значения и графики изменения специально отобранных переменных из различных областей памяти.

3. Программа должна производить обработку сигналов измерительных преобразователей. Результаты обработки должны заноситься в четыре переменные, которые будут считываться `SCADA`-системой и содержать значения уровня в миллиметрах. Разрешающая способность АЦП (0...6000) и скоростного счетчика (160...1600) в принципе позволяют обеспечить и большую дискретизацию, однако характеристики самих измерительных преобразователей делают это бессмысленным. Так, разрешающая способность самого точного, ультразвукового, измерителя согласно технической документации составляет 0,5 мм, погрешности остальных – еще больше.

Настройка каналов производится по их показаниям в двух точках: на *нижнем* и *верхнем* уровнях. Нижние уровни у преобразователей различаются. Для обеспечения «единства» показаний желательно выбрать для всех каналов один и тот же нижний уровень, – такой, на котором все измерители выдают «достоверный» сигнал, отступив немного вверх от максимального из всех каналов минимума. В качестве верхнего уровня для всех каналов, очевидно, следует принять уровень 400 мм.

Съем показаний каналов на нижнем и верхнем уровнях рекомендуется проводить следующим образом:

управляя вручную преобразователем частоты с помощью средств его лицевой панели, сначала вывести уровень немного выше заданного и дождаться успокоения колебаний;

приоткрывая ручной клапан слива, плавно вывести уровень на требуемую отметку и вновь дождаться успокоения колебаний;

зафиксировать показания каналов. Если показания «плывут», постараться зафиксировать минимум и максимум и определить среднее значение.

Программная обработка сигналов по показаниям в двух точках выполняется с помощью команды SCL (194). Входными переменными команды являются исходное слово S и адрес первого слова параметров P1. Результат попадает в слово результата R в двоично-десятичном виде (BCD).

В ПЛК семейства CP довольно широко используется двоично-десятичный формат представления данных (BCD – binary coded decimal). Каждый десятичный разряд в таком формате записывается в виде его четырехразрядного двоичного кода. При этом одно и то же число в десятичном и двоично-десятичном форматах «выглядит» в разных инструкциях одинаково (за исключением «префикса»), но хранится (и обрабатывается!) по-разному:

&1538 – 0000 0110 0000 0010 ($2^{10} + 2^9 + 2^1 = 1024+512+2=1538$)

#1538 – 0001 0101 0011 1000 (BCD)

Перед выполнением команды требуется подготовить непрерывную область памяти, состоящую из четырех слов. Пусть, например, эта область начинается с W0:

W0 – желаемое (преобразованное) значение в первой точке в формате BCD;

W1 – исходное (до преобразования) значение в первой точке в двоичном формате;

W2 – желаемое (преобразованное) значение во второй точке в формате BCD;

W3 – исходное (до преобразования) значение во второй точке в двоичном формате.

Далее, при вызове команды SCL слово W0 должно быть записано в параметр P1 (фактически записывается его адрес).

Допустим, мы имеем следующую ситуацию. Источник исходных данных – аналоговый вход, слово 1 (CIO). Первая точка (нижний уровень) – 80 мм, выдаваемое значение – 373. Вторая точка (верхний уровень) – 400 мм, выдаваемое значение – 5263. Результат преобразования необходимо записать в W4. Последовательность действий:

```
MOV(021) #080 W0
MOV(021) &373 W1
MOV(021) #400 W2
MOV(021) &5263 W3
SCL(194) 1 W0 W4
```

Результат выполнения команды SCL представлен в двоично-десятичном виде. Для его преобразования в двоичную форму необходимо использовать команду BIN (23):

```
BIN(023) W4 H5
```

Здесь результат преобразования заносится в слово H5.

В окончательной версии программы для обработки сигналов всех четырех каналов можно использовать одну и ту же рабочую область для всех четырех команд SCL (в приведенном выше примере это W0...W4). Однако на этапе отладки и настройки желательно все-таки задействовать для каждой команды SCL разные области памяти, чтобы в режиме мониторинга наблюдать актуальные значения настроечных переменных по каждому каналу (а не значения для последнего канала).

«Окончательные результаты» в любом случае, очевидно, следует отправлять по разным адресам, откуда они потом будут считываться SCADA-системой. Например, это могут быть слова H4...H7.

После настройки все четыре канала должны выдавать одни и те же показания (в миллиметрах) как минимум в двух точках. Если имеют место отклонения, а это вполне возможно для «плавающих» программных сигналов, необходимо подкорректировать настроечные значения (в примере это W1 и W3).

Полученную программу следует дополнить цепями проверки достоверности показаний, ориентируясь на данные табл. 3. Наиболее рационально эту проверку осуществлять еще до масштабирования сигналов, тогда в случае, если значение какого-либо входа находится вне допустимого диапазона, его масштабирование можно и не выполнять, ограничившись установкой «флага недостоверности», т.е. определенного бита в специально выделенном для этого слове. Это может быть, например, слово H8:

H8.00 – достоверность показаний ультразвукового ИП;

H8.01 – достоверность показаний емкостного ИП;

H8.02 – достоверность показаний погружного ИП;

H8.03 – достоверность показаний гидростатического ИП.

4. Дополнительное задание: создание каналов ввода/вывода задающего сигнала будущему регулятору уровня.

Разрабатываемая в данной лабораторной работе система измерения будет в дальнейшем положена в основу системы автоматического регулирования уровня. Для того, чтобы эта система могла функционировать и без участия SCADA-системы, необходимо обеспечить возможность ввода и отображения сигнала задания регулятору непосредственно на стенде. Для этого можно задействовать потенциометр CH90 и вольтметр CH190, доступные по адресам 90 и 190 соответственно. Требуется:

выполнить масштабирование входного сигнала, приведя его к диапазону 0...400;

обеспечить вывод показания на вольтметр в диапазоне 0...4В.

Указания по запуску и апробации системы

1. Апробируйте систему. Убедитесь в работоспособности всех элементов контроля и управления.

2. Продемонстрируйте работу системы преподавателю. При наличии замечаний устраните недостатки.

2.3. Система автоматического регулирования уровня

Постановка задачи

Система должна обеспечивать возможность ручного, дистанционного и автоматического регулирования уровня воды в резервуаре.

В *ручном режиме* управление осуществляется с помощью тумблеров 0.01 – 0.07 (тумблер 0.00 задействовать нельзя, поскольку он физически отключен от ПЛК) и потенциометра СН90, посредством которого задается электрическая частота питания двигателя насоса. Уровень контролируется непосредственно визуально по линейке, а частота – по показаниям вольтметра СН190 (изменению частоты от 0 до 50 Гц соответствует изменение напряжения от 0...5,0 В).

Назначение тумблеров:

0.01 – пуск системы. При включенном состоянии разрешаются все дальнейшие действия. В выключенном состоянии система находится в состоянии покоя: ПЧ – в режиме «стоп», сливной клапан закрыт;

0.02 – запуск процесса автоматического регулирования;

0.03 – переход в режим дистанционного управления со SCADA-системы. Тумблер имеет больший приоритет, чем 0.02, т.е. при одновременном включении управление все равно передается SCADA-системе;

0.04 – команда пуска ПЧ;

0.05, 0.06 – выбор измерительного преобразователя для системы автоматического регулирования:

HE 0.05 и HE 0.06 – гидростатический;

0.05 и HE 0.06 – погружной;

0.05 и 0.06 – емкостный;

HE 0.05 и 0.06 – ультразвуковой;

0.07 – включение сливного клапана.

В *дистанционном режиме* управление и контроль осуществляются посредством элементов графического интерфейса SCADA-системы Trace Mode на компьютере. Рекомендуется создать два экрана: основной, при помощи которого будет осуществляться собственно дистанционное управление, и дополнительный, предназначенный для работы в автоматическом режиме. Примерный вид основного экрана показан на рис. 23.

Индикатор «Разрешение» загорается зеленым цветом, когда система находится в режиме дистанционного управления. Кнопки «ПЧ» и «Слив» предназначены для пуска преобразователя и включения сливного клапана. Кнопки изменяют свое состояние (нажата/отпущена) во всех режимах (не только в дистанционном), информируя пользователя о состоянии аппаратуры. Кнопка «Авто» используется для перевода системы в режим автоматического регулирования, при ее нажатии открывается дополнительное окно.

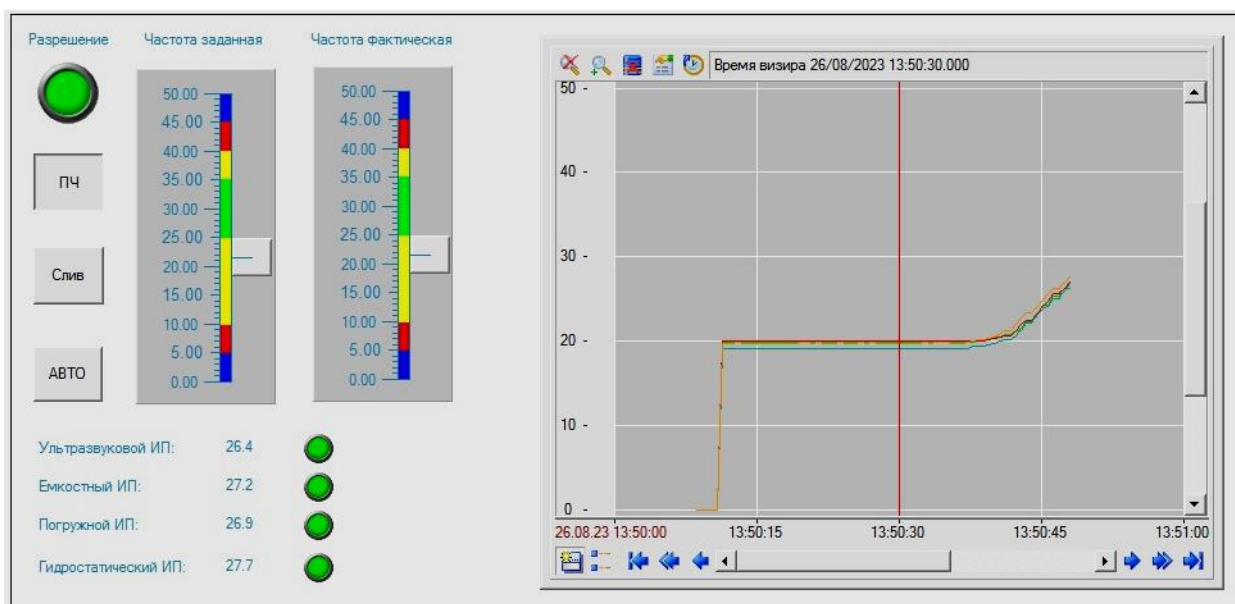


Рис. 23. Примерный вид основного экрана SCADA.

Два ползунка предназначены для ввода заданной частоты и просмотра текущей. В общем случае эти частоты совпадают только в дистанционном режиме (и только после окончания переходного процесса изменения частоты). С помощью правого ползунка можно наблюдать текущую частоту во всех режимах.

В левом нижнем углу экрана выводятся показания всех каналов измерения уровня с индикацией их достоверности. Элемент визуализации «Тренд», расположенный справа, показывает графики изменения сигналов во времени.

В режиме автоматического регулирования, который может быть запущен как из ручного, так и из дистанционного режима, регулированием занимается программный ПИД-регулятор, формирующий сигнал задания по частоте для ПЧ. Поскольку в отсутствии слива управляющее воздействие способно повысить уровень, но не понизить его, в автоматическом режиме слив должен иметь место. Поэтому предлагается одновременно с запуском автоматического регулятора включать электромагнитный клапан. Увеличить возмущение можно открывая ручной кран.

Если автоматическое регулирование запущено из ручного режима, задание регулятору уровня формируется потенциометром СН90 и контролируется по показаниям вольтметра СН190 (изменению заданного уровня от 0 до 400 мм соответствует изменение показаний от 0 до 4,0 В).

Если автоматическое регулирование запущено из дистанционного режима, в SCADA-системе открывается дополнительное окно, примерный вид которого показан на рис. 24.

Индикатор «Разрешение» загорается зеленым светом, когда система находится в режиме автоматического управления под контролем SCADA-системы. Слева от индикатора – кнопка, с помощью которой вводится и контролируется задание. Ниже располагаются «фиктивные» кнопки, на них вы-

водятся значения управляющего воздействия и регулируемой величины. Под ними выводятся показания всех каналов измерения уровня с индикацией достоверности. В отличие от основного окна здесь имеется возможность выбора измерительного канала, который будет задействован в системе регулирования. Выбранный канал выделяется цветом.

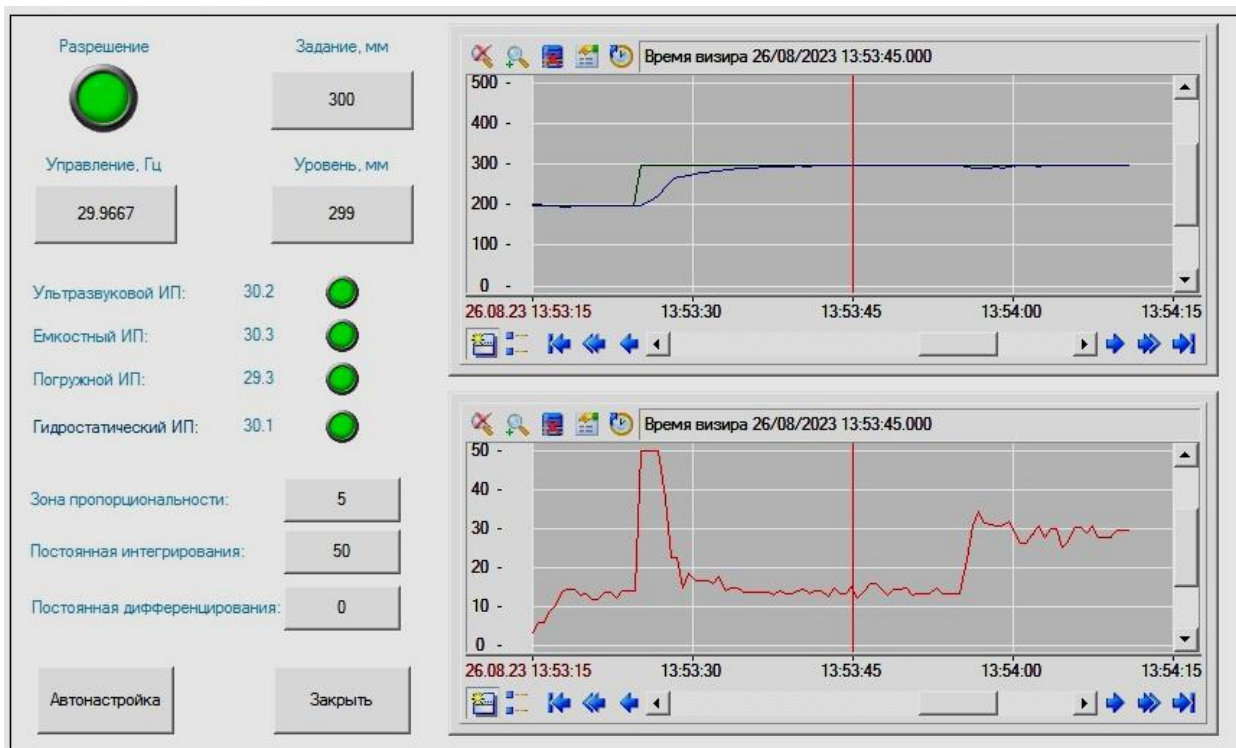


Рис. 24. Примерный вид дополнительного экрана SCADA.

Ниже располагаются графические элементы (надписи и кнопки), предназначенные для ввода настроек автоматического регулятора.

Кнопка «Автонастройка» запускает процесс автонастройки регулятора и фиксирует его протекание. После окончания процесса она автоматически отжимается. При нажатии на кнопку «Закрыть» дополнительное окно закрывается и открывается основное. При этом система переводится в режим дистанционного управления.

Справа располагаются два тренда. Верхний показывает графики изменения заданного и фактического уровней в миллиметрах. Нижний – график изменения управляющего воздействия (частоты).

Указания по написанию программы ПЛК

1. Программа должна инициализировать модуль расширения CP1W-MAD11 (см. п. 1.1).

2. Программа должна использовать созданную в предыдущей работе систему измерения уровня, т.е. по сути она будет «продолжением» программы, разработанной ранее. С другой стороны, будут полезны и некоторые наработки, полученные в ходе выполнения первой работы. Их можно перенести простым копированием кода, открыв одновременно два окна программы CX-Programmer.

3. На первом этапе разработки рекомендуется создать упрощенный «прототип» подсистемы автоматического регулирования и добиться его устойчивого функционирования.

В контроллере CP1E автоматическое ПИД-регулирование осуществляет команда PIDAT (191) – PID control with autotuning. В режиме собственно регулирования команда выполняет алгоритм, представленный на функциональной схеме (рис. 25).

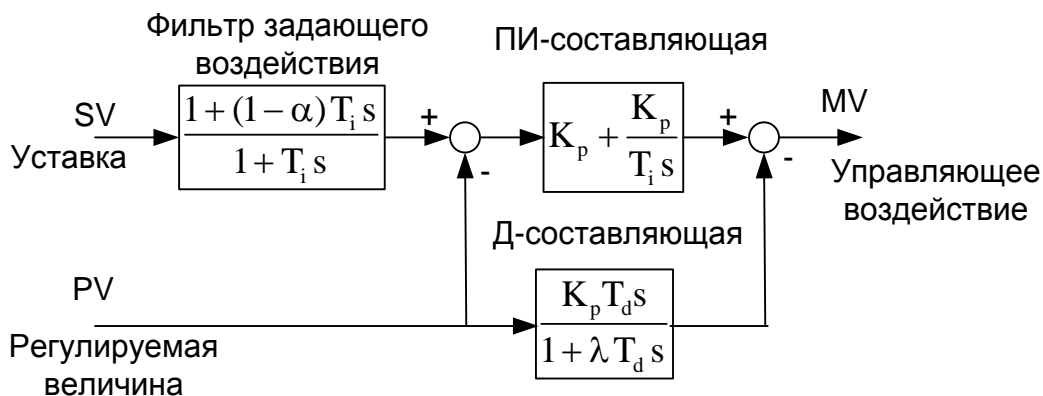


Рис. 25. Функциональная схема алгоритма.

Схема взята из справочной системы программы CX-Programmer, однако в ней были сделаны необходимые, по мнению автора, исправления (следует отметить, что в русскоязычном руководстве схема воспроизведена некорректно). Алгоритм реализует так называемое *ПИД-регулирование с двумя степенями свободы*. В отличие от стандартного ПИД-регулятора сигнал задания не дифференцируется, но пропускается через входной фильтр, представляющий собой интегро-дифференцирующее звено. Это позволяет уменьшить перерегулирование в переходных процессах, вызванных изменением задания, при сохранении хорошего качества переходных процессов по возмущению. Настраиваемым параметром фильтра является величина α , называемая в документации *2-ПИД-параметром* и *коэффициентом фильтрации задающего воздействия*. Этот коэффициент может быть настроен в диапазоне от 0 до 1. При $\alpha = 0$ фильтрация отключается совсем, при $\alpha = 1$ фильтр становится апериодическим звеном. Оптимальным обычно считается $\alpha = 0,65$, это значение и устанавливается по умолчанию.

Д-составляющая регулятора реализуется реальным дифференцирующим звеном. Параметр λ называется в справочной системе *коэффициентом неполной производной*. Как следует из документации, он не подлежит настройке и его значение, к сожалению, в документации не приведено.

Остальные параметры регулятора имеют «традиционный» смысл: K_p – коэффициент передачи (фактически настраивается обратная ему *зона пропорционального регулирования*); T_i – постоянная времени интегрирования; T_d – постоянная времени дифференцирования.

Команда PIDAT имеет следующий формат вызова:

PIDAT (191) S C D,

где S – слово входных данных (регулируемая величина); C – первое слово параметров (и «рабочей области») алгоритма; D – слово выходных данных (управляющее воздействие).

В параметр S следует записать некоторое слово, в которое, в свою очередь, попадет обработанное значение одного из измерителей уровня (в зависимости от выбора пользователя). Пусть это будет, например, $H9$ (рис. 26).

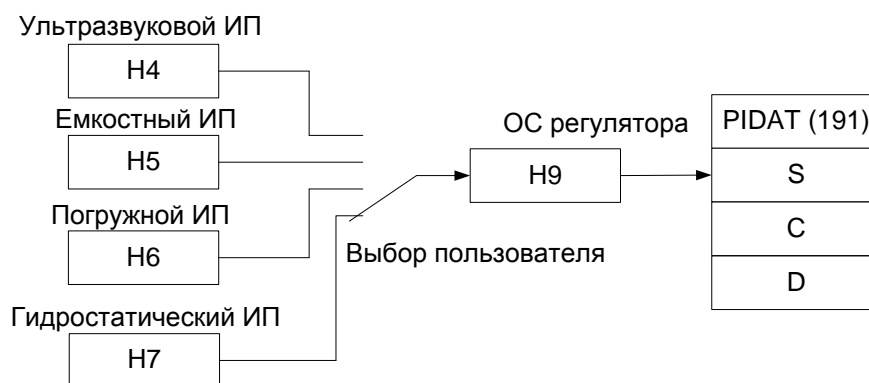


Рис. 26. Схема формирования «сигнала обратной связи» для регулятора.

Управляющее воздействие D – выход задания частоты 101.

Перед запуском команды PIDAT требуется тщательно подготовить довольно большую область памяти, начинающуюся с адреса, записываемого в параметр C , и занимающую 11 слов, – здесь будут находиться настройки регулятора. Кроме того, необходимо следить за тем, чтобы программа не затрагивала следующие за этими словами еще 30 слов, поскольку эта область будет использоваться регулятором для своих целей (рис. 27). Назначение, варианты и числовые диапазоны параметров приведены в табл. 10. Более подробную информацию можно получить в справочной системе.

Под параметры и рабочую область регулятора рекомендуется отвести слова из области памяти D , – например, $D0...D40$.

Заполнение слов на этапе отладки можно производить «вручную» с помощью средства просмотра и редактирования памяти Memory из дерева проекта. В окончательной версии заполнение, естественно, должно осуществляться программно, посредством стандартной команды MOV (021).

Настоятельно рекомендуется провести заполнение всех 11 слов на этапе инициализации системы, т.е. в первом цикле, по флагу P_Fisrt_Circle .

Задание ($D0$) можно обнулить или установить некоторое приемлемое значение. В качестве первого приближения рекомендуется принять следующие значения настроек регулятора:

- зона пропорциональности $P = 10$ ($D1$);
- постоянная интегрирования $T_i = 100$ ($D2$);
- постоянная дифференцирования $T_d = 0$ ($D3$).

Однако эти значения требуют коррекции, поскольку при них регулятор ведет себя довольно «вяло». Коррекция будет выполнена по результатам экспериментов и самонастройки регулятора.

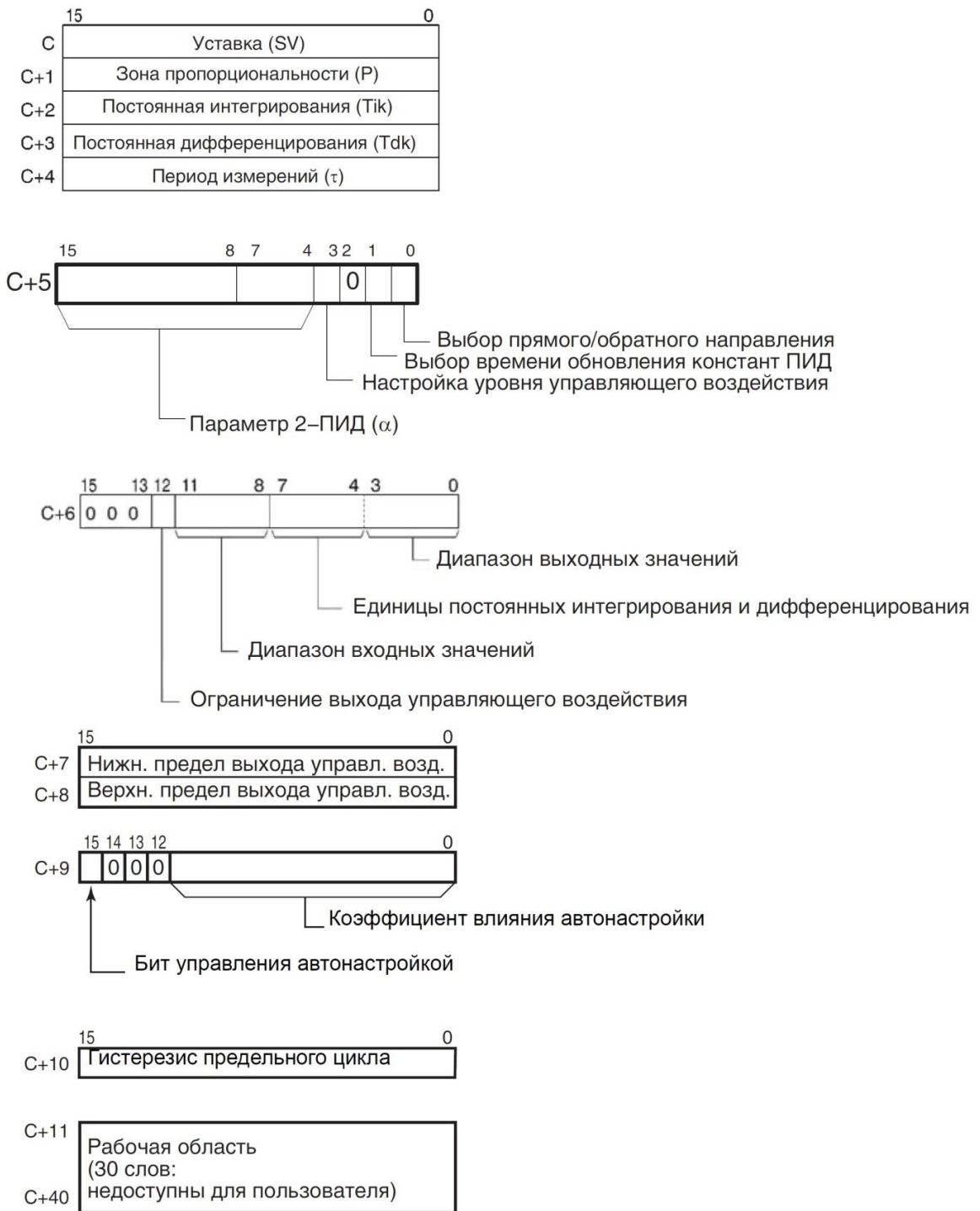


Рис. 27. Задание параметров ПИД-регулятора.

Период измерения (D4) рекомендуется установить равным 10 мс.

Параметр 2-ПИД (речь идет о слове D5) можно принять равным по умолчанию, уровень управляющего воздействия при нулевой ошибке – 0. Обновление ПИД-констант в каждом цикле следует разрешить, чтобы не требовалось каждый раз останавливать регулирование при изменении настроек. Направление пропорционального действия – обратное (что соответствует работе регулятора в режиме «нагреватель»).

Ограничение выхода управляющего воздействия (D6.12) рекомендуется включить. Диапазон входных данных в нашем случае небольшой и вполне укладывается в 10 бит (0...1023, фактически меньше), однако, как показывает опыт работы с регулятором, он должен быть не меньше чем выходной диапазон, который в нашем случае составляет 13 бит (0...8191, фактически 0...6000). Единицы постоянных интегрирования/дифференцирования – время (единица 100 мс), выходной диапазон 13 бит (0...8191, фактически 0...6000).

Нижний предел выхода управляющего воздействия (D7) – 0, верхний (D8) – 6000. Слова, отвечающие за самонастройку (D9, D10), на этапе инициализации следует обнулить, при этом все соответствующие параметры примут значения по умолчанию. В дальнейшем запуск самонастройки будет производиться установкой бита D9.15.

Таблица 10. Описание параметров алгоритма PIDAT

Управляющие данные	Параметр	Содержание	Диапазон значений	Изменение при включенном входном условии
1	2	3	4	5
C	Уставка (SV)	Заданное значение физической величины регулируемого техпроцесса.	Двоичное значение (с тем же числом битов, что указано для входного диапазона)	Разрешено
C+1	Зона пропорциональности	Параметр пропорционального воздействия, определяемый как отношение диапазона пропорционального регулирования к полному диапазону регулирования.	0001...270F hex (1...9999); (0,1%...999,9%, с шагом 0,1%)	Изменение при вкл. вх. условия возможно, если бит 1 слова C+5 = 1.
C+2	Tik Постоянная интегрирования	Константа, определяющая величину интегрального воздействия. С ростом этого параметра величина интегрального воздействия снижается.	0001...1FFF hex (1...8191); (9999 = интегральное регулирование не осуществляется) (см. примеч. 1.)	
C+3	Tdk Постоянная дифференцирования	Константа, определяющая величину воздействия по производной. С ростом этого параметра величина воздействия по производной снижается.	0001...1FFF hex (1...8191); (0000 = регулирование по производной не осуществляется) (см. примеч. 1.)	
C+4	Период измерения (τ)	Определяет периодичность, с которой выполняется команда ПИД-регулирования.	0001...270F hex (1...9999); (0,01...99,99 с, с шагом 10 мс)	Не допускается
Биты 04...15 слова C+5	Параметр 2-ПИД (α) Разрешение изменения ПИД-констант	Коэффициент фильтрации входного значения. Обычно следует использовать значение 0,65 (т. е. значение параметра 000). По мере приближения коэффициента фильтрации к 0 эффективность фильтрации снижается.	000 hex: $\alpha = 0,65$ Значениям от 100 до 163 hex соответствуют значения коэффициента от $\alpha = 0,00$ до $\alpha = 0,99$ (см. прим. 2).	
Бит 03 слова C+5		Определяет уровень управляющего воздействия при равенстве регулируемой переменной и уставки ($PV = SV$).	0: выход 0%; 1: выход 50%.	
Бит 01 слова C+5	Настройка уровня управляющего воздействия	Определяет, когда именно измененные значения констант (P, Tik и Tdk) вступают в силу и начинают использоваться для вычислений ПИД-регулятором	0: в начале выполнения команды ПИД-регулирования; 1: в начале выполнения команды ПИД-регулирования и в каждом цикле регулирования	Разрешено

1	2	3	4	5
Бит 00 слова С+5	Выбор прямого/ обратного направления для ПИД-регулятора	Определяет направление пропорционального воздействия.	0: обратное действие; 1: прямое действие.	Не допускается
Бит 12 слова С+6	Ограничение выхода управляющего воздействия	Определяет, должно ли применяться ограничение к выходу управляющего воздействия.	0: выключено (выход не ограничивается); 1: включено (выход ограничивается).	
Биты 08...11 слова С+6	Диапазон входных значений	Количество действительных битов в слове входного значения.	0: 8 битов; 5: 13 битов; 1: 9 битов; 6: 14 битов; 2: 10 битов; 7: 15 битов; 3: 11 битов; 8: 16 битов. 4: 12 битов;	
Биты 04...07 слова С+6	Единицы постоянных интегрирования и дифференцирования	Определяет единицы, в которых выражаются постоянные интегрирования и дифференцирования.	1: множитель к периоду измерения; 9: время (единица: 100 мс).	
Биты 00...03 слова С+6	Диапазон выходных значений	Количество действительных битов в слове выходного значения (количество битов выходного значения автоматически устанавливается равным количеству битов входного значения).	0: 8 битов; 5: 13 битов; 1: 9 битов; 6: 14 битов; 2: 10 битов; 7: 15 битов; 3: 11 битов; 8: 16 битов. 4: 12 битов;	
С+7	Нижний предел выхода управл. воздействия	Нижнее предельное значение управляющего воздействия при включенном ограничении выхода управляющего воздействия.	0000...FFFF (двоичн.) (См. примеч. 3)	
С+8	Верхний предел выхода управл. воздействия	Верхнее предельное значение управляющего воздействия при включенном ограничении выхода управляющего воздействия.	0000...FFFF (двоичн.) (См. примеч. 3)	
Бит 15 слова С+9	Бит управления автонастройкой	Этот бит служит для запуска автонастройки. • Для запуска автонастройки бит управления автонастройкой следует переключить в состояние «1» (автонастройку также можно запустить во время выполнения команды PIDAT(191)). • По завершении автонастройки этот бит автоматически выключается. Если бит управления автонастройкой будет выключен во время автонастройки (вручную), автонастройка будет прервана. В этом случае будут действовать ПИД-константы, вычисленные ранее (до запуска автонастройки).	В качестве бита управления: • 0 → 1: выполняется автонастройка. • 1 → 0: автонастройка прерывается (команда PID(191) выключает бит автоматически по завершении автонастройки). Применение в качестве флага: 0: автонастройка не выполняется; 1: в данный момент выполняется автонастройка.	

1	2	3	4	5
Биты 00...11 слова С+9	Коэффициент влияния автонастройки	<p>Данный параметр позволяет отрегулировать соотношение между вычисленными значениями ПИД-констант и фактически сохраняемыми значениями.</p> <p>В большинстве случаев следует оставить принимаемые по умолчанию значения (0000).</p> <ul style="list-style-type: none"> • Увеличьте это значение, чтобы повысить стабильность. • Уменьшите это значение, чтобы повысить скорость отклика. 	0000 hex: 1,00 (по умолчанию); 0001...03E8 hex (1...1000); (0,01...10,00, с шагом 0,01).	Разрешено (Значения данных параметров считываются при запуске автонастройки.)
С+10	Гистерезис предельного цикла	<p>Задаёт величину гистерезиса для предельного цикла. По умолчанию при обратном направлении управления управл. воздействие включается с гистерезисом SV–20%.</p> <p>Данный параметр рекомендуется повысить, если из-за нестабильности регулируемой переменной не удастся сформировать надлежащий предельный цикл. В то же время, при избыточной величине гистерезиса предельного цикла повышается погрешность автонастройки.</p>	0000 hex: 0,20% (по умолч); 0001...03E8 hex: 0,01...10,00% с шагом 0,01%; FFFF hex: 0,00%. Примечание. Значение в процентах от входного диапазона.	

Примечания к таблице:

(1) Если параметр выбора единиц измерения содержит «1», диапазон значений – от 1 до 8191 периодов измерения. Если параметр выбора единиц измерения содержит «9», диапазон значений – от 0,1 до 819,1 с. При значении «9» постоянные интегрирования и дифференцирования следует задавать так, чтобы они оставались в пределах от 1 до 8191 периодов измерения.

(2) При вводе значения «000» в параметр 2-ПИД (α) действует значение 0,65, являющееся нормой.

(3) Если включено ограничение выхода управляющего воздействия (т.е. выбрано «1»), значения следует задавать с соблюдением следующего правила: $0000 \leq$ нижн. предел выхода $MV \leq$ верхн. предел выхода $MV \leq$ макс. значение выходного диапазона.

4. Перед тем как приступить к разработке окончательной версии программы, необходимо определиться с ее структурой и назначением переменных.

Рекомендуется разбить программу на следующие секции:

1) «Инициализация» – здесь происходит первоначальная настройка модуля расширения CP1W-MAD11 (см. п. 1.1), регулятора (см. выше) и обнуление некоторых переменных области Н, которые будут задействованы в программе;

2) «Измерения» – здесь производится обработка сигналов измерительных преобразователей уровня. Итоговые значения (в миллиметрах) заносятся

по адресам Н4...Н7, а флаги достоверности – по адресу Н8. Фактически код этой секции был создан при выполнении предыдущей работы;

3) «Работа_с_режимами» – здесь будут фиксироваться режим работы системы и выполняться определенные действия в зависимости от установленного режима и поступающих команд. В частности, речь идет об управлении ПЧ (формирование команды «Пуск» и заданной частоты), электромагнитным клапаном слива, формировании задания для регулятора, обработке сигнала потенциометра СН90 и «управлении» вольтметром СН190;

4) «Регулирование» – здесь выбирается источник сигнала обратной связи и вызывается команда PIDAT.

Рекомендации по назначению переменных:

Н0 – режим работы системы. Одновременно установлен только один бит:

Н0.00 – «Стоп»;

Н0.01 – «Ручное управление»;

Н0.02 – «Дистанционное управление» (от SCADA-системы);

Н0.03 – «Автоматическое регулирование с щитовым контролем»;

Н0.04 – «Автоматическое регулирование под контролем SCADA»;

Н1 – дискретные сигналы от SCADA:

Н1.0 – команда включить сливной клапан;

Н1.1 – команда «Пуск» ПЧ;

Н1.2 – команда перехода на автоматическое регулирование;

Н2 – задание по частоте, формируемое в SCADA-системе (дистанционный режим);

Н3 – выбор измерительного преобразователя для автоматического регулирования. Одновременно установлен только один бит:

Н3.00 – ультразвуковой;

Н3.01 – емкостный;

Н3.02 – погружной;

Н3.03 – гидростатический;

Н4 – результат обработки сигнала ультразвукового ИП;

Н5 – результат обработки сигнала емкостного ИП;

Н6 – результат обработки сигнала погружного ИП;

Н7 – результат обработки сигнала гидростатического ИП;

Н8 – признаки достоверности показаний измерительных каналов с преобразователями:

Н3.00 – ультразвуковым;

Н3.01 – емкостным;

Н3.02 – погружным;

Н3.03 – гидростатическим;

Н9 – входной сигнал автоматического регулятора (регулируемая величина);

Н10 – задание (уставка) автоматическому регулятору от SCADA-системы.

Указания по разработке SCADA-системы

1. Все органы отображения информации о состоянии оборудования должны функционировать во всех режимах работы системы. Однако воздействие на процесс может быть оказано только в разрешенных режимах. Поэтому каждый элемент взаимодействия SCADA-системы с ПЛК должен использовать как минимум два канала (источника/приемника): входной, управляющий отображением, и выходной – для подачи команды.

Так, например, состояние экранной кнопки (нажата/отпущена), отвечающей за включение сливного клапана в дистанционном режиме (рис. 23) должно быть привязано к выходу ПЛК 100.02, что обеспечивает отображение состояния клапана во всех режимах. С другой стороны, при нажатии этой кнопки подается «команда» (устанавливается бит H1.0), которая выполняется только в режиме дистанционного управления, а в остальных игнорируется.

*Примененный подход имеет один недостаток. Совмещение функций отображения и управления приводит к ощутимой временной задержке между нажатием на кнопку и изменением ее состояния: системе требуется произвести два обмена данными между SCADA и ПЛК (запись в ПЛК команды и чтение из ПЛК результатов ее выполнения). Если вместо кнопки применить переключатель, как это было сделано ранее, то его состояние будет отражать только «желаемое» состояние клапана, а не фактическое, поскольку переключатель привязывается к единственному выходному каналу. Отправка команды производится в момент «переключения». Если в этот момент команда выполнена быть не может (не тот режим), то клапан не включится, а переключатель свое состояние изменит. Возможно, лучшим вариантом будет разделение ответственности между несколькими элементами визуализации: две кнопки без фиксации для **управления** клапаном («Включить» и «Выключить») и «сигнальная лампа» для индикации его состояния. Но такой вариант несколько загромождает экран.*

2. Все сигналы в данной системе рекомендуется провести через каналы Trace Mode. Если этого не делать и допускать привязку аргументов экрана непосредственно к «Источникам/Приемникам», то работоспособность SCADA-системы не гарантируется, о чем честно предупреждают разработчики Trace Mode (и что подтверждает опыт). Входные аргументы экранов привязываются к *реальным* (выходным) значениям каналов, выходные аргументы – к *входным* значениям.

При автопостроении создаются каналы типа Float. Однако если нужно оперировать битами, удобнее использовать каналы типа Hex16. Они создаются и привязываются к «Источникам/Приемникам» вручную. Аргумент экрана можно привязать к отдельному битовому атрибуту канала Hex16, как это показано на рис. 28. При этом необходимо помнить, что нумерация битов начинается с единицы.

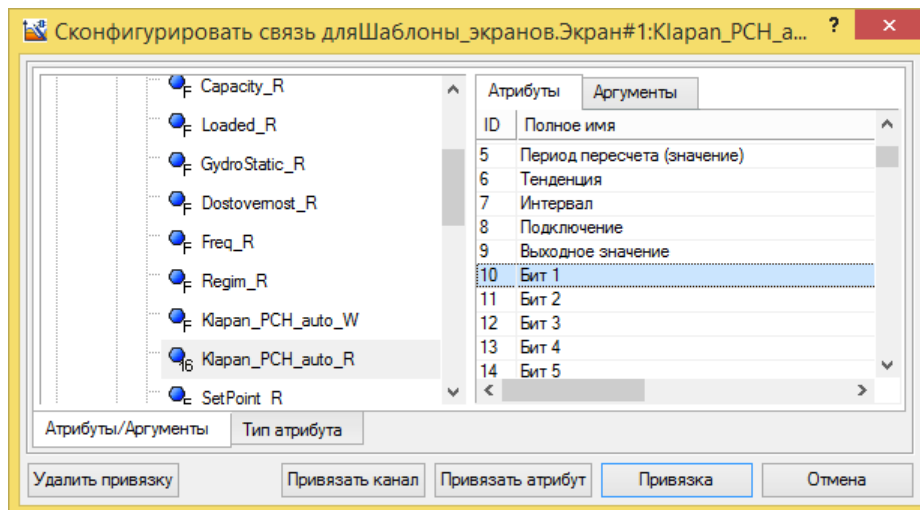


Рис. 28. Привязка аргумента экрана к атрибуту Бит 1 канала типа Нех16.

Указания по запуску и апробации системы

1. Апробируйте работу системы во всех режимах. Убедитесь в работоспособности всех элементов контроля и управления.

2. Апробируйте работу автоматического регулятора в сочетании со всеми измерительными преобразователями уровня. Особое внимание уделите ультразвуковому преобразователю: в разработанной ранее системе был установлен достаточно большой интервал измерения частоты следования импульсов (1 с). Существенная задержка в измерительном канале может привести к ухудшению качества процесса регулирования, вплоть до появления незатухающих колебаний. При необходимости уменьшите интервал измерения.

3. Изменяя настройки регулятора вручную, добейтесь наилучшего качества переходных процессов (максимальное быстродействие без существенного перерегулирования при изменении задания). После этого включите процесс автонастройки. Сравните поведение автоматически настроенной системы с поведением системы, настроенной вручную.

4. Продемонстрируйте работу системы преподавателю. При наличии замечаний устраните недостатки.

2.4. Система автоматического воспроизведения периодических сигналов

Постановка задачи

Система должна обеспечивать возможность формирования и воспроизведения периодических сигналов задания (синусоидального, пилообразного и сигнала прямоугольной формы) с настраиваемыми параметрами – частотой и амплитудой. Периодические сигналы должны формироваться в SCADA-системе. Программа ПЛК, созданная при выполнении предыдущей работы, изменению не подлежит.

Основное окно SCADA-системы, показанное на рис. 23, остается без изменений. Дополнительное окно, отвечающее за автоматическое регулирование, требует незначительной доработки (рис. 29).

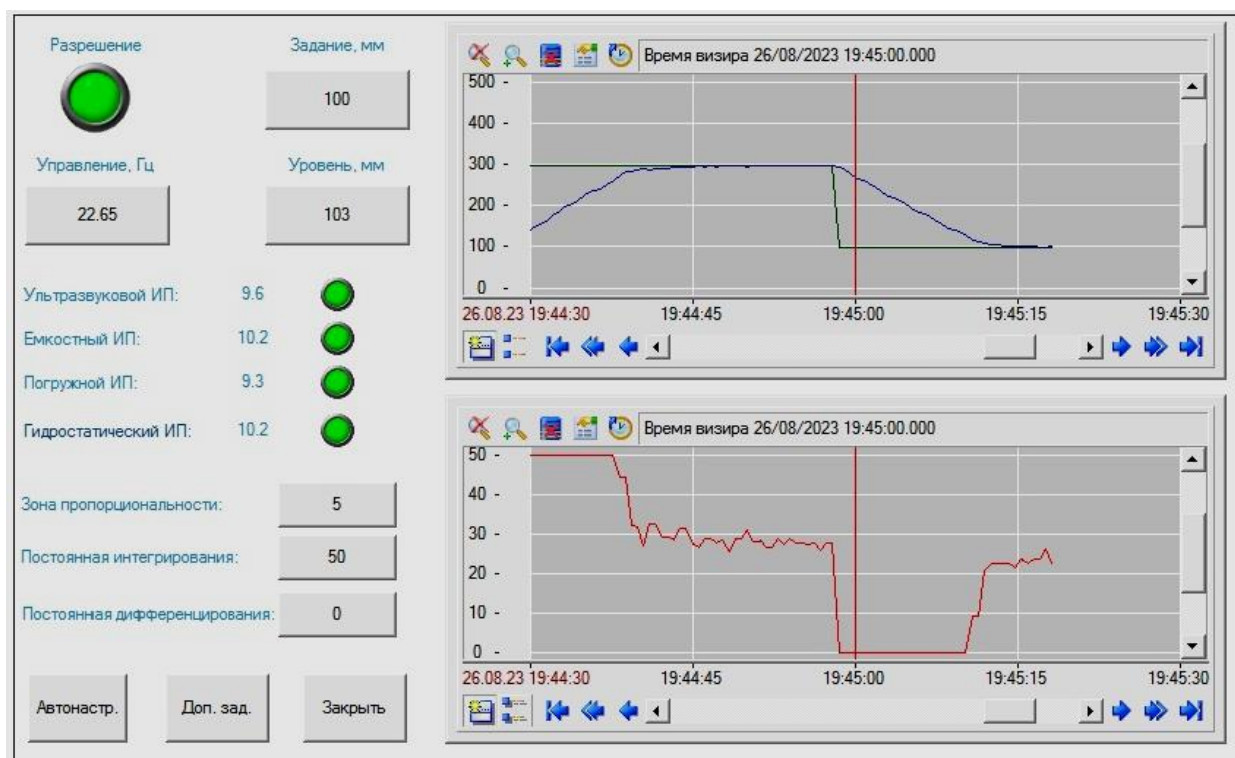


Рис. 29. Дополнительное окно.

Кнопка «Задание» по-прежнему отображает текущее задание. Несмотря на это, она теперь будет использоваться не для ввода задания как такового, а только его среднего уровня, относительно которого и будут проходить колебания. Вычислением самого задающего сигнала будет заниматься специальная программа, один из входных аргументов которой – полученное «от кнопки» значение.

Кнопка «Доп. Зад.» (дополнительное задание) открывает третье окно, в котором будет производиться выбор формы и ввод параметров периодического сигнала. Примерный вид окна показан на рис. 30.

Размещенные в верхнем левом углу фиктивные кнопки служат для отображения фактических значений уставки и регулируемой величины. Ниже расположены надписи и индикаторы, предназначенные для выбора формы сигнала, и средства ввода параметров колебаний.

Кнопка «Закреть» используется для возврата в предыдущее окно. Справа расположен тренд, на котором выводятся графики изменения сигналов задания и регулируемой величины.

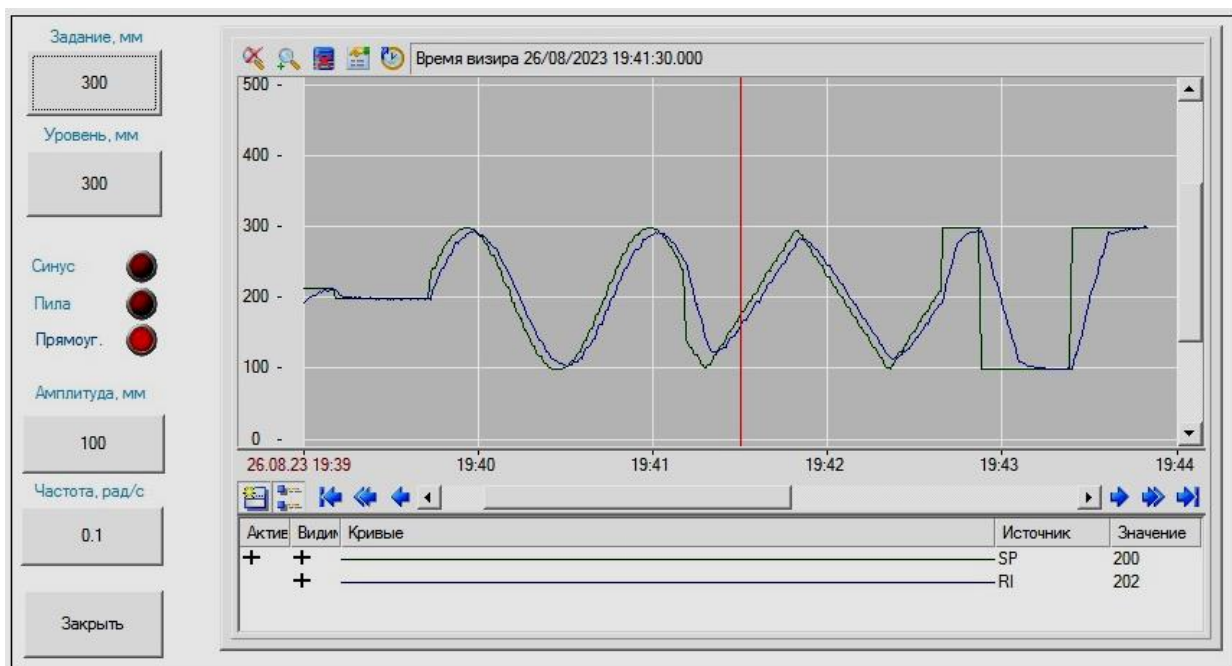


Рис. 30. Окно формирования периодического сигнала задания
Указания по разработке SCADA-системы

Основную сложность представляет разработка программы.

1. Программа должна формировать сигнал задания по текущему времени. Значение текущего времени в секундах с дробной частью может быть сформировано из значений системных переменных `@t_Set_Time` и `@t_Operation_Time`. Первая переменная представляет циклически изменяемое значение в диапазоне от 0 до 59999 мс, полученное из системного времени ОС, вторая – непрерывно увеличивающееся количество секунд с момента запуска монитора реального времени Trace Mode (в нашем случае – профайлера). Доступ к системным переменным устанавливается в Источниках/Приемниках, группа «Диагностика и сервис», подгруппа «Системные». Далее полученные значения требуется «пропустить» через каналы Trace Mode и «направить» во входные аргументы программы.

В программе формирование необходимого значения можно выполнить по следующей схеме.

Пусть, например, `@t_Set_Time = 12345`, `@t_Operation_Time = 67`. С момента запуска системы прошло 67с (целых). Сколько дополнительно прошло миллисекунд неизвестно, но это и не важно. Важно лишь, чтобы время изменялось быстрее, чем раз в секунду. Поэтому нас вполне устроит число 67,345. Сформировать его можно следующим образом:

- 1) $12345 \% 1000 = 345$ (остаток от деления);
- 2) $345 / 1000 = 0,345$;
- 3) $67 + 0,345 = 67,345$.

2. Помимо временных переменных, входными аргументами программы будут:

амплитуда, мм;

круговая частота колебаний, рад/с;
форма колебаний (может быть представлена целочисленной переменной, принимающей значения 0, 1, 2 в зависимости от выбора пользователя);
среднее значение (постоянная составляющая) сигнала задания.

Все эти аргументы нужно привязать к соответствующим аргументам экранов (рис. 29, 30).

Выходным аргументом программы будет сигнал задания. Этот аргумент должен быть привязан к входному значению соответствующего канала.

Кроме входных и выходных аргументов, рекомендуется объявить следующие *локальные* переменные:

время с момента запуска системы, рассчитываемое по алгоритму, приведенному выше;

длительность полупериода колебаний, рассчитываемая через круговую частоту. Эта переменная потребуется для формирования пилообразных и прямоугольных колебаний.

3. Расчет синусоидального сигнала не вызывает затруднений: достаточно использовать встроенную функцию `SIN`. Для формирования пилообразных и прямоугольных колебаний предлагается следующий подход.

Необходимо объявить две *глобальные* переменные (сохраняемые между вызовами):

время начала текущего полупериода;

«направление» сигнала. Для пилообразных колебаний эта переменная будет определять действительно направление (вверх или вниз), для прямоугольных – просто выбор одного из двух возможных значений (положительный/отрицательный). Естественным образом переменная должна принимать два значения: 1 и –1. Кроме того, ей обязательно нужно присвоить начальное значение, поскольку нулевое значение по умолчанию не подходит.

Эти переменные будут обновляться каждый раз, когда закончится очередной полупериод колебаний, т.е. когда разница между текущим временем и временем начала текущего полупериода достигнет значения самого полупериода. Текущее время будет сохранено как время начала текущего полупериода, а «направление» сигнала изменится на обратное.

Пользуясь этими переменными, несложно сформировать необходимые выражения для расчетов сигналов.

Выбор выражения в зависимости от формы колебаний рекомендуется оформить в виде программной конструкции `CASE OF`.

4. Если по каким-либо причинам рассчитанное значение задания окажется отрицательным, это может иметь неприятные последствия, поскольку отрицательные значения программа контроллера воспримет как большие положительные, и уровень начнет быстро подниматься до срабатывания предохранительного поплавка. Для предотвращения этого рекомендуется в конце программы добавить код, который обнулит задание при отрицательном результате расчета.

Указания по запуску и апробации системы

1. Рекомендуемые параметры колебаний:

средний уровень 200 мм;

амплитуда 100 мм;

частота 0,1 рад/с и ниже.

2. Регулирование уровня в режиме воспроизведения понижающегося задания обеспечивается совместной работой преобразователя частоты и системы слива. При достаточно большой частоте изменения задающего сигнала пропускная способность электромагнитного клапана не обеспечит необходимую скорость слива. В таком случае потребуется дополнительно открыть ручной сливной кран.

3. Апробируйте систему и продемонстрируйте ее работу преподавателю. При наличии замечаний устраните недостатки.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Система автоматического управления уровня. Методические указания к проведению лабораторных работ. – Челябинск: Научно-производственное предприятие «Учтех-Профи», 2014. – 68 с.
2. W479 SYSMAC CP Series CP1E CPU Unit Hardware User's Manual. Cat. No. W479-E1-04 – Omron, 2009. – 262 с.
3. SYSMAC CP Series CP1E CPU Unit Software User's Manual. Cat. No. W480-E1-04 – Omron, 2009. – 450 с.
4. SYSMAC CP Series CP1E CPU Unit Instructions Reference Manual. Cat. No. W483-E1-04. – Omron, 2009. – 466 с.
5. Лупал А.М. Теория автоматов. Учеб. Пособие – СПб.: СПбГУАП, 2000. – 119 с.

Адрес редакции и издателя:

675027, г. Благовещенск, Игнатьевское шоссе, 21, комн. 406.

Адрес типографии:

675000, г. Благовещенск, ул. Мухина, 150а

Адрес учредителя:

ФБГОУ ВО «Амурский государственный университет»
675027, г. Благовещенск, Игнатьевское шоссе, 21.

Андрей Николаевич Рыбалёв,

доцент кафедры АППиЭ АмГУ, канд. техн. наук.

Программируемые логические контроллеры и аппаратура управления:
лабораторный практикум. Часть 7. OMRON CP1E: Измерение и регулирова-
ние уровня. Учебное пособие

Издательство АмГУ. Подписано к печати __.2024. Редактор – *О.К. Ма-
монтова*. Формат 60 x 84/16. Усл. печ. л. 5,89. Тираж __. Заказ ____.