

К. В. Вавилов

Контроллеры *SIMATIC S7-300* (*SIEMENS*)

Организация взаимодействия независимых локальных систем управления на основе автоматного подхода и функционального разделения автоматов управления

Санкт-Петербург

2005

Оглавление

1. ОСНОВНЫЕ РАБОТЫ И МАТЕРИАЛЫ	3
2. ЦЕЛЬ ДАННОЙ РАБОТЫ.....	3
3. ОБЪЕКТЫ И СРЕДСТВА УПРАВЛЕНИЯ	3
4. ТРЕБОВАНИЯ К АСУ В ЧАСТИ ЛОГИКИ УПРАВЛЕНИЯ	6
5. АЛГОРИТМЫ УПРАВЛЕНИЯ.....	6
5.1. БАЗОВЫЕ АВТОМАТЫ (КОНТРОЛЛЕРЫ S7-200)	7
5.1.1. Автомат управления нереверсивным приводом.....	7
5.1.2. Автомат управления реверсивным приводом.....	8
5.2. ТЕХНОЛОГИЧЕСКИЕ АВТОМАТЫ (КОНТРОЛЛЕР S7-200 ШАУ КО).....	10
5.2.1. Главный технологический автомат управления работой КО (ATMainCC).....	10
5.2.2. Технологический автомат управления фильтрацией КО (ATFCC).....	12
5.2.3. Технологический автомат управления промывкой фильтра КО (ATWCC).....	14
5.3. ТЕХНОЛОГИЧЕСКИЕ АВТОМАТЫ (КОНТРОЛЛЕР S7-200 ШАУ НА).....	16
5.3.1. Главный технологический автомат ШАУ НА (ATMainWP).....	16
5.3.2. Технологический автомат управления работой насосного агрегата (ATWPSFW).....	18
5.4. ТЕХНОЛОГИЧЕСКИЕ АВТОМАТЫ УПРАВЛЕНИЯ ПРОМЫВКОЙ (МАСТЕР-КОНТРОЛЛЕР S7-300).....	20
5.4.1. Главный технологический автомат управления (ATmain).....	20
5.4.3. Автомат управления промывкой осветлителя в очереди (ATWx).....	24
6. РЕАЛИЗАЦИЯ АЛГОРИТМОВ	26
ЗАКЛЮЧЕНИЕ.....	29
Приложение 1. Техническое задание “Процесс промывки БКО в автоматическом режиме”	30
П 1.1. Общие сведения	30
П 1.2. Описание алгоритма промывки	30
П 1.3. Нештатные ситуации.....	33
Приложение 2. Примеры сгенерированного кода автоматов проекта.....	36
П 2.1. Главный технологический автомат управления ATmain (рис. 17, 18)	36
П 2.2. Технологический автомат управления промывкой ATWmain (рис. 19, 20)	37
П 2.3. Автомат управления промывкой осветлителя в очереди ATWx (рис. 21, 22).....	40
Приложение 3. Примеры подпрограмм вызова автоматов	42
П 3.1. Вызов автомата ATmain.....	42
П 3.2. Вызов автомата ATWmain.....	43
П 3.3. Вызов автомата ATWx_curr.....	45
П 3.4. Вызов автомата ATWx_next.....	48
ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ	50

1. Основные работы и материалы

- [1] Вавилов К.В.
“Программируемые логические контроллеры SIMATIC S7-200 (SIEMENS).
Методика алгоритмизации и программирования задач логического управления”,
<http://is.ifmo.ru/progeny/metod065.pdf>
- [2] Вавилов К.В.
“LabVIEW и SWITCH-технология”,
<http://is.ifmo.ru/progeny/vavilov2.pdf.zip>
- [3] Материалы сайта <http://is.ifmo.ru/>
- [4] Шалыто А.А.
“SWITCH-технология. Алгоритмизация и программирование задач
логического управления“
<http://is.ifmo.ru/books/switch/1>

2. Цель данной работы

Работа является продолжением создания методик и примеров практического применения [1, 2] технологии автоматного программирования [3]. Она, также как и работа [2], призвана показать какой объем проектирования, реализации и тестирования можно реально выполнить, используя SWITCH-технология [4], эффективность которой с повышением сложности логической задачи возрастает.

В работе представлен пример реализации взаимодействия независимых локальных систем управления на основе автоматного подхода и функционального разделения автоматов управления, изложенного в работе [1].

3. Объекты и средства управления

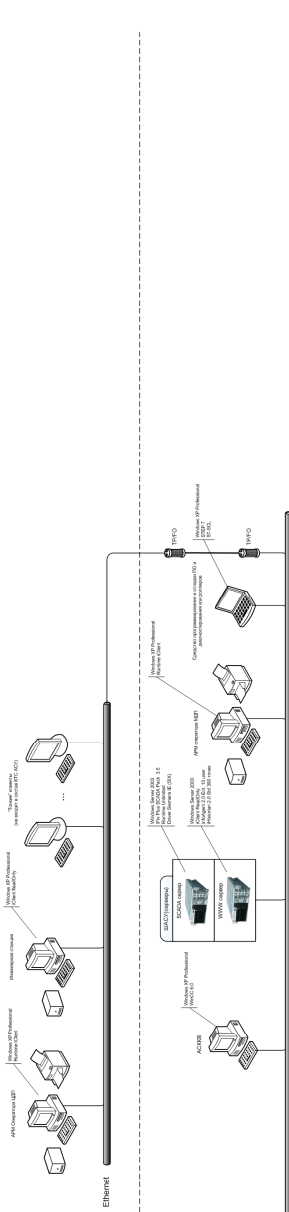
Данная работа является описанием решений для одной из частей проекта автоматизации – управления процессом промывки блока контактных осветителей (БКО) водопроводной станции. В целом проект был выполнен летом-осенью 2004 г. Отладка производилась осенью 2004 г. – летом 2005 г. Сначала она проводилась на полигоне, а затем проходили испытания на объекте.

Конечными объектами управления являлись 96 задвижек (по четыре задвижки 24-х контактных осветителей), насос и напорная задвижка (составляющие насосного агрегата подачи промывной воды).

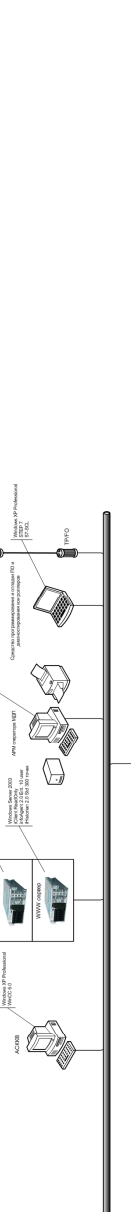
Контроллер S7-300 (модель S7-317-2DP) использовался как Мастер-контроллер сети ProfiBus. Каждое отдельное устройство управления задвижками контактного осветителя (КО) и насосным агрегатом (НА) представляло собой контроллер SIMATIC S7-200 (модель S7-224), связанный через интеллектуальный модуль по сети Profibus с Мастер-контроллером. Контроллеры КО и НА были помещены в шкафы автоматического управления (ШАУ), совмещавшие в себе функции как автоматического, так и местного управления. Схемы управления приводами размещались также в этих шкафах.

Структурная схема АСУ ТП приведена на рис. 1. Топология сети ProfiBus представлена на рис. 2. На этих рисунках жирным выделены устройства управления, используемые при управлении промывкой.

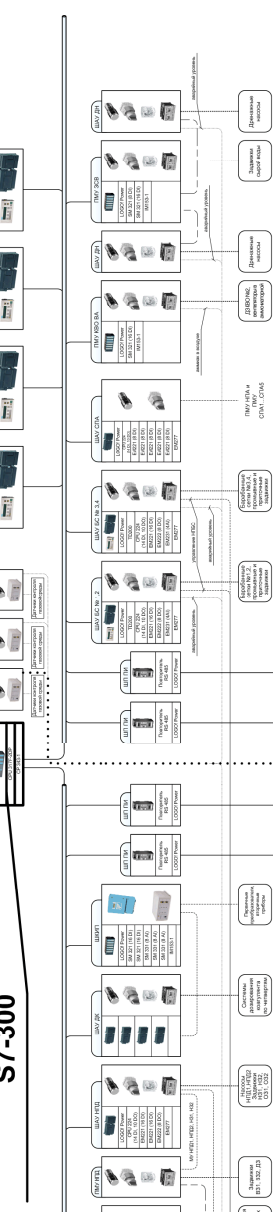
Уровень административного и диспетчерского управления



Уровень оперативного управления

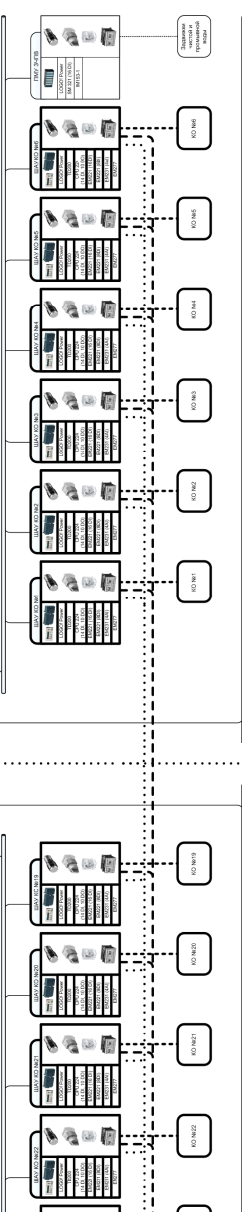


Базовый уровень автоматизации управления



Мастер-Контроллер S7-300

Контроллеры S7-200 ШАУ КО



Контроллеры S7-200 ШАУ НА

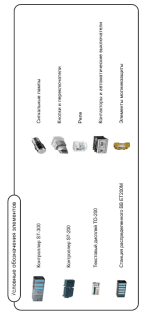
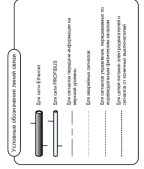
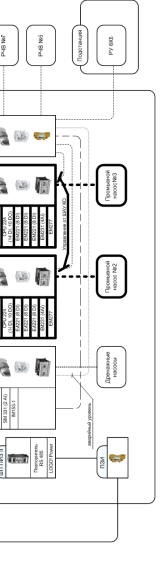
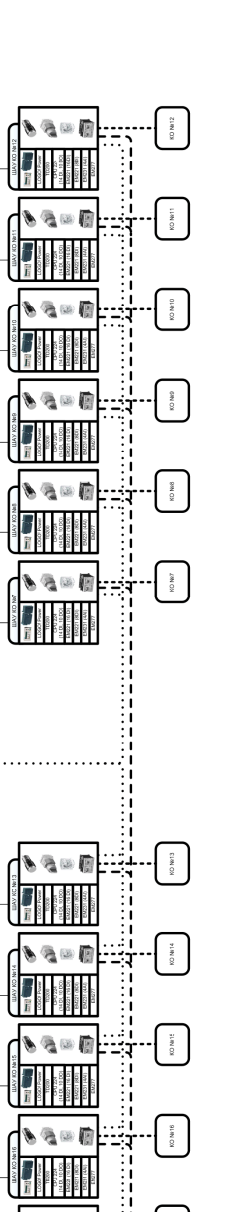


Рис. 1. Автоматизированная система управления БКО
Структурная схема комплекса технических средств

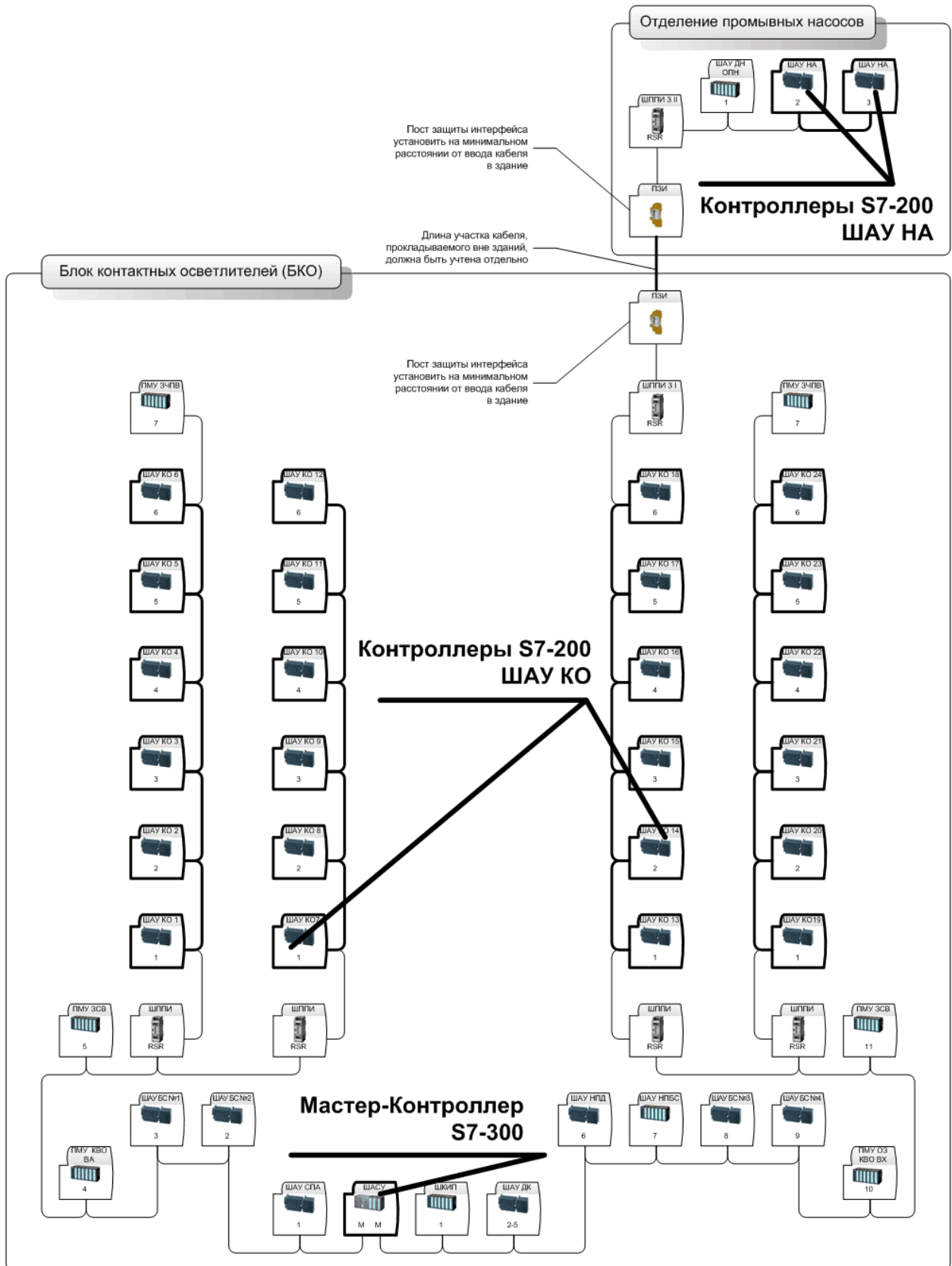


Рис.2. Топология сети ProfiBus

4. Требования к АСУ в части логики управления

Первоначальное техническое задание в виде описания технологии промывки было создано Заказчиком. После обсуждения было выработано окончательное техническое задание (ТЗ). Оно приведено в **Приложении 1**.

До ввода в действие автоматического режима управления промывкой (под управлением Мастер-контроллера), промывка выполнялась при местном режиме КО. При этом НА работал под управлением контроллера в автоматическом режиме (команды инициализации процессов пуска/останова подавались дистанционно с любого ШАУ КО). Таким образом, схема управления ШАУ и алгоритмы работы контроллеров КО и НА создавались с учетом такого предварительного режима работы.

Как следует из технического задания, в нем особое внимание уделялось ненормальной работе оборудования. Промывка должна прерываться только при наличии непреодолимых условий (например, при отсутствии связи с промываемым КО или ненормальной работе промывной задвижки при переходе с моющегося КО на подготовленный к промывке).

Не описанное в ТЗ понятие “отсутствие связи” означает не только потерю связи с интеллектуальным модулем контроллера ШАУ (отвечающим за выполнение коммуникационных функций), но и то, что контроллер находится в режиме STOP.

5. Алгоритмы управления

Один из вариантов применения **SWITCH-технологии** [4] (именно принципов применения, а не самой технологии) подробно описан в методике [1]. Следует отметить, что они действуют для ЛЮБОГО КОНТРОЛЛЕРА (не только SIMATIC).

Алгоритмы управления в соответствии с указанной методикой разделены на **базовые** и **технологические**. **Базовые** – это алгоритмы непосредственного управления приводами и выполняются (вызываются) постоянно. **Технологические** алгоритмы предназначены для формирования команд управления (в требуемый момент времени) и выполняются при автоматическом режиме управления (управлении от контроллера).

Как следует из структурной схемы (рис.1), система управления являлась распределенной. При этом необходимо было разработать механизм корректной работы (взаимодействия) параллельно работающих и независимых локальных систем управления (программ контроллеров ШАУ КО). Именно **функциональное разделение алгоритмов на базовые и технологические, которые представлены как автоматы**, позволило без труда решить указанную проблему. Получилась интересная, но вместе с тем вполне логичная ситуация – **принципы разделения автоматов**, помогавшие корректно управлять в локальной системе управления, **сработали и на более высоком уровне** – уровне взаимодействия нескольких локальных систем.

Согласно методике [1] исходными данными для технологического алгоритма являются не только входные сигналы в контроллер, но и не менее важна информация, формируемая базовыми алгоритмами – информация о ненормальной работе привода. В рассматриваемом проекте Мастер-контроллер не имел модулей ввода/вывода. Он получал ВСЮ информацию о поведении объектов управления от контроллеров КО и НА и, соответственно, не управлял приводами непосредственно. Для Мастер-контроллера разрабатывались только технологические алгоритмы, использующие всю полноту информации от **базовых автоматов подчиненных систем**.

Исходя из изложенного, для описания полной “картины” логики управления ниже приведены схемы связей и графы переходов ВСЕХ автоматов проекта в части управления промывкой, хотя в названии данной работы упоминается только контроллер S7-300 – главное управляющее устройство процесса промывки. Схемы и графы переходов разработаны с помощью пакета *MS Visio*.

5.1. Базовые автоматы (контроллеры S7-200)

5.1.1. Автомат управления нереверсивным приводом

Схема связей автомата приведена на рис. 3, а граф переходов – на рис. 4.



Рис. 3. Схема связей автомата управления нереверсивным приводом

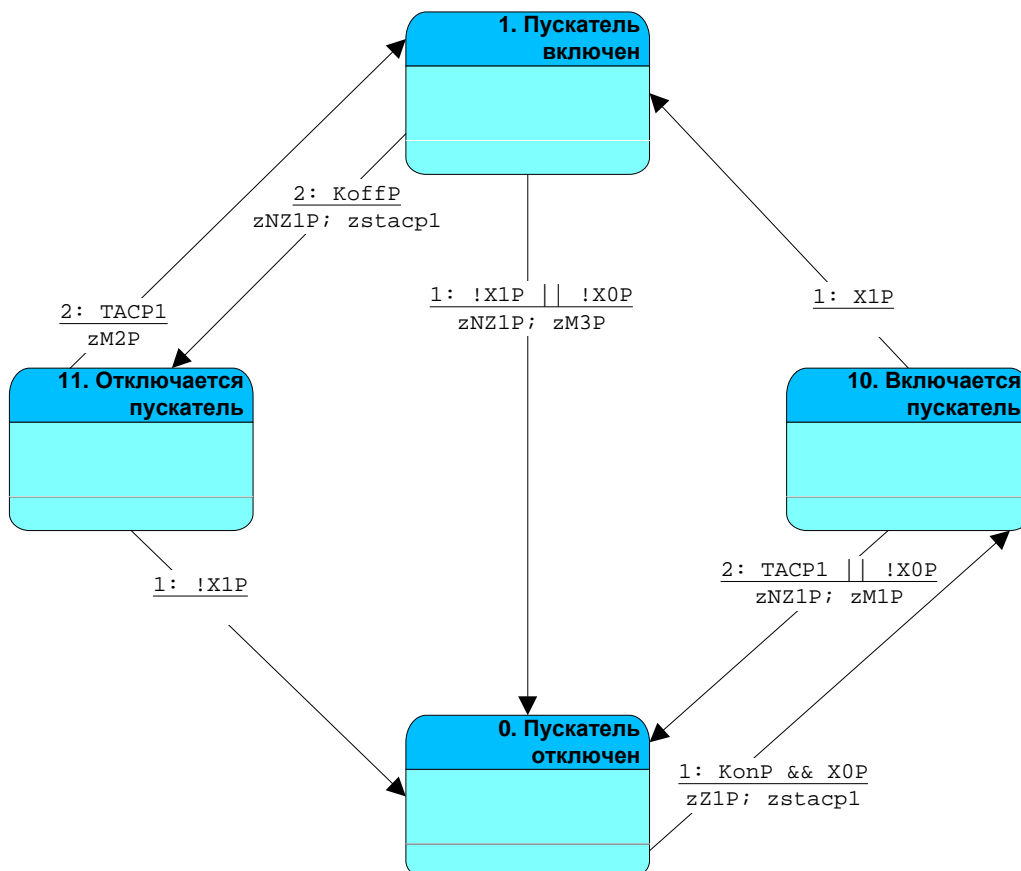


Рис. 4. Граф переходов автомата управления нереверсивным приводом

5.1.2. Автомат управления реверсивным приводом

Схема связей автомата приведена на рис. 5, а граф переходов – на рис. 6.

Автомат управления приводом затвора (ACSx)		
Включено питание привода	X0S	zZ1S
Включен на открытие	X1S	zNZ1S
Включен на закрытие	X2S	
Открыт	X3S	zZ2S
Закрыт	X4S	
Команда "Открыть"	KopS	zNZ2S
Команда "Закрыть"	KclS	
Команда "Остановить"	KoffS	
Пускатель на открытие отключен по внешней причине (проверено временем)	noneX1S_time	
Пускатель на закрытие отключен по внешней причине (проверено временем)	noneX2S_time	
Окончилось время контроля включения пускателя	TACS1	zstacs1
Окончилось время контроля открытия/закрытия затвора	TACS2	zstacs2
		zM1S
		zM2S
		zM3S
		zM4S
		zM7S
		zM8S
	zMopS	
	zMclS	

Рис. 5. Схема связей автомата управления реверсивным приводом

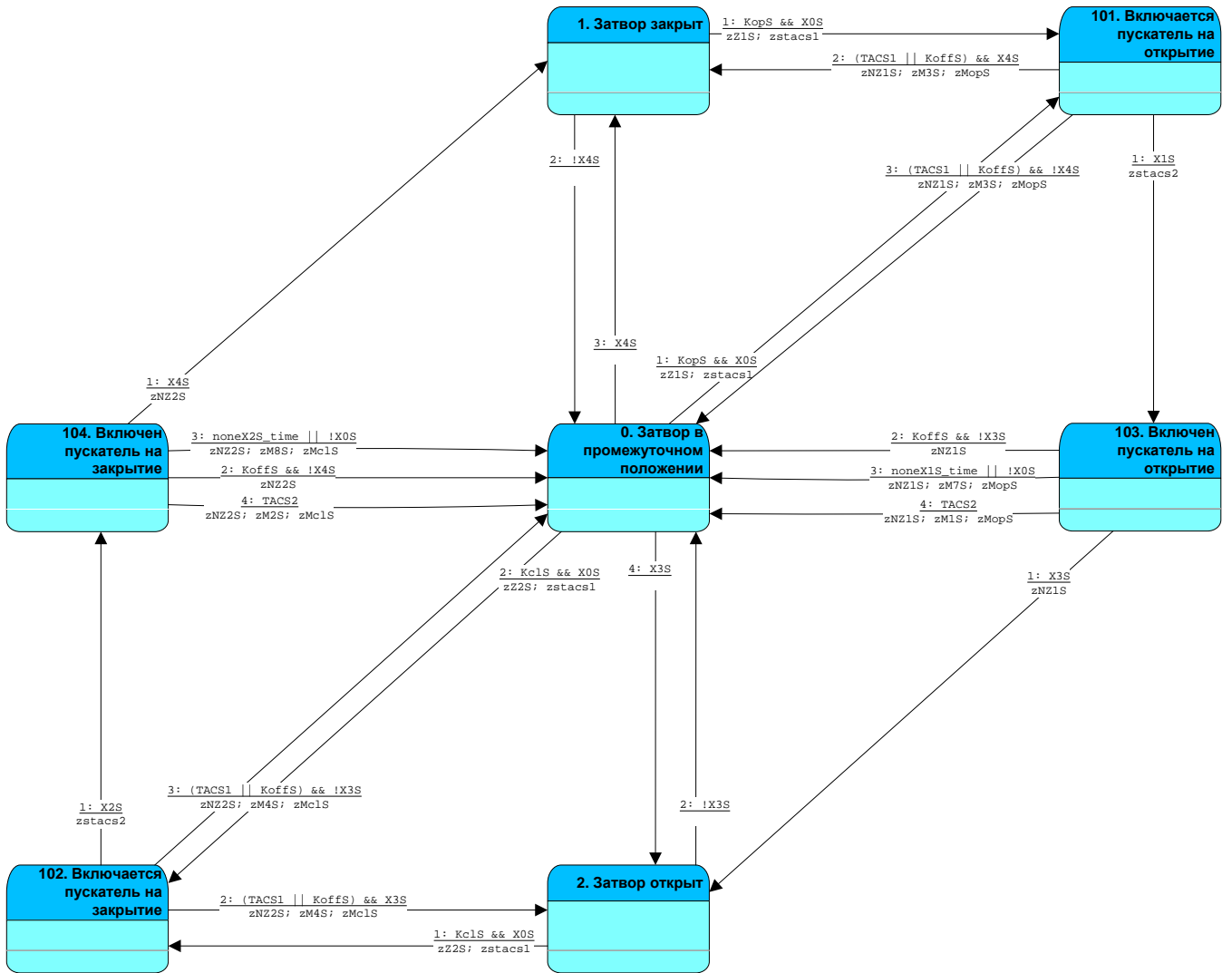


Рис. 6. Граф переходов автомата управления реверсивным приводом

5.2. Технологические автоматы (контроллер S7-200 ШАУ КО)

5.2.1. Главный технологический автомат управления работой КО (ATMainCC)

Схема связей автомата приведена на рис. 7, а граф переходов – на рис. 8.

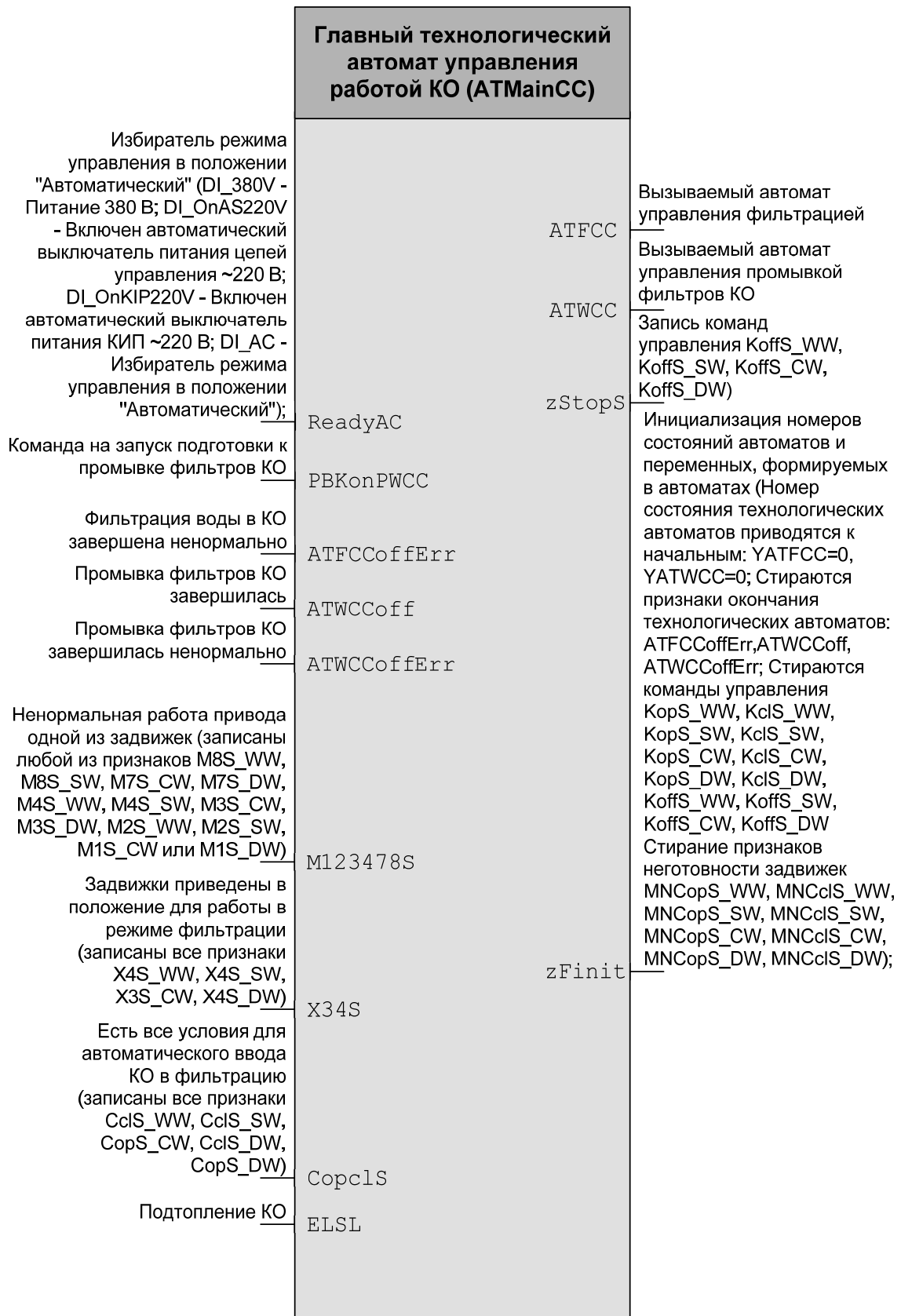


Рис. 7. Схема связей главного технологического автомата управления работой КО (ATMainCC)

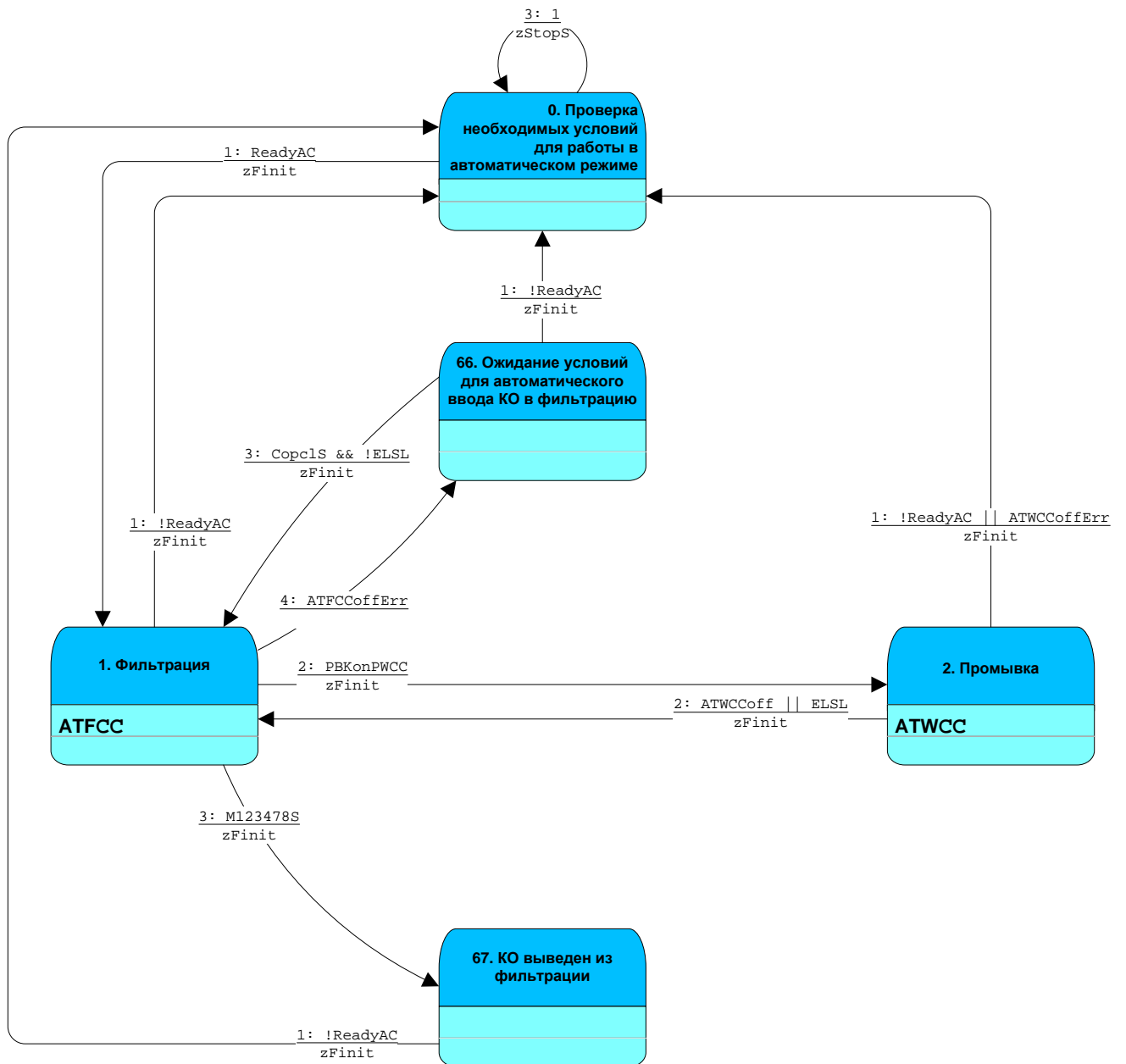


Рис. 8. Граф переходов технологического автомата управления работой КО (ATMainCC)

5.2.2. Технологический автомат управления фильтрацией КО (ATFCC)

Схема связей автомата приведена на рис. 9, а граф переходов – на рис. 10.

Технологический автомат управления фильтрацией КО (ATFCC)		
Ненормальное открытие задвижки сырой воды	MopS_DW	
Ненормальное закрытие задвижки сырой воды	MclS_DW	zKopS_DW
Ненормальное открытие задвижки чистой воды	MopS_CW	zKclS_DW
Ненормальное закрытие задвижки слива воды в канализацию	MclS_SW	zNKopS_DW
Ненормальное закрытие задвижки промывной воды	MclS_WW	zNKclS_DW
Задвижка сырой воды открыта	DDI_X3S_DW	zKoffS_SW
Задвижка сырой воды закрыта	DDI_X4S_DW	zKopS_CW
Задвижка чистой воды открыта	DDI_X3S_CW	zNKopS_CW
Задвижка слива воды в канализацию закрыта	DDI_X4S_SW	zKclS_SW
Задвижка промывной воды закрыта	DDI_X4S_WW	zNKclS_SW
Подтопление КО	ELSL	zNKclS_SW
Есть все условия для автоматического открытия задвижки сырой воды (X0 - Включено питание привода задвижки; !X2 - Не включен привод задвижки на закрытие; !X5 - Задвижка не заклинена; !HTD - Нет превышения температуры двигателя сырой воды)	CopS_DW	zKclS_WW
Есть все условия для автоматического закрытия задвижки сырой воды (X0 - Включено питание привода задвижки; !X1 - Не включен привод задвижки на открытие; !X5 - Задвижка не заклинена; !HTD - Нет превышения температуры двигателя сырой воды)	CopS_DW	zNKclS_WW
Есть все условия для автоматического открытия задвижки чистой воды	CclS_DW	zMNCopS_DW
Есть все условия для автоматического закрытия задвижки слива воды в канализацию И НЕТ ПОДТОПЛЕНИЯ	CopS_CW	zMNCclS_DW
Есть все условия для автоматического закрытия задвижки промывной воды	CclS_SW	zMNCopS_CW
	CclS_SW	zMNCclS_SW
	CclS_WW	zMNCclS_WW
Окончилось время контроля задержки входа в режим фильтрации	TDELAY	zATFCCoffErr
		zstdelay
		Запись команды "Открыть задвижку сырой воды"
		Запись команды "Закрыть задвижку сырой воды"
		Стирание команды "Открыть задвижку сырой воды"
		Стирание команды "Закрыть задвижку сырой воды"
		Запись команды "Остановить привод задвижки слива воды в канализацию"
		Запись команды "Открыть задвижку чистой воды"
		Стирание команды "Открыть задвижку чистой воды"
		Запись команды "Закрыть задвижку слива воды в канализацию"
		Стирание команды "Закрыть задвижку слива воды в канализацию"
		Запись команды "Закрыть задвижку промывной воды"
		Стирание команды "Закрыть задвижку промывной воды"
		Запись признака "Нет условий для автоматического открытия задвижки сырой воды"
		Запись признака "Нет условий для автоматического закрытия задвижки сырой воды"
		Запись признака "Нет условий для автоматического открытия задвижки чистой воды"
		Запись признака "Нет условий для автоматического закрытия задвижки слива воды в канализацию"
		Запись признака "Нет условий для автоматического закрытия задвижки промывной воды"
		Запись признака "Фильтрация воды в КО завершена ненормально"
		Начать контроль времени задержки входа в режим фильтрации

Рис. 9. Схема связей технологического автомата управления фильтрацией КО (ATFCC)

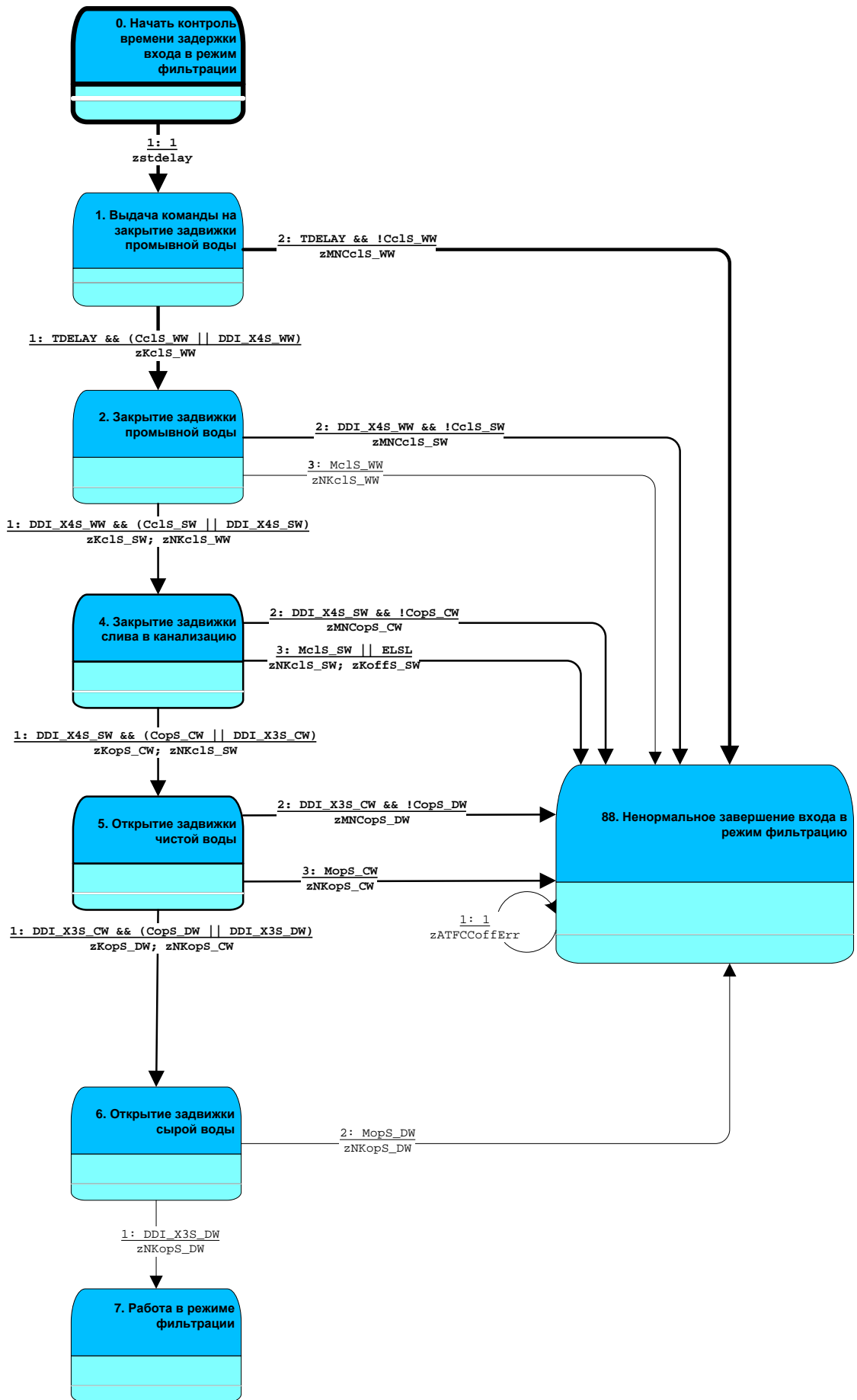


Рис. 10. Граф переходов технологического автомата управления фильтрацией КО (ATFCC)

5.2.3. Технологический автомат управления промывкой фильтра КО (ATWCC)

Схема связей автомата приведена на рис. 11, а граф переходов – на рис. 12.

Технологический автомат управления промывкой фильтра КО (ATWCC)			
Ненормальное открытие задвижки сырой воды	MopS_DW	zKopS_DW	Запись команды "Открыть задвижку сырой воды"
Ненормальное закрытие задвижки сырой воды	MclS_DW	zNKopS_DW	Стирание команды "Открыть задвижку сырой воды"
Ненормальное закрытие задвижки чистой воды	MclS_CW	zKclS_DW	Запись команды "Закрыть задвижку сырой воды"
Ненормальное открытие задвижки слива воды в канализацию	MopS_SW	zNKclS_DW	Стирание команды "Закрыть задвижку сырой воды"
Ненормальное открытие задвижки промывной воды	MopS_WW	zKclS_CW	Запись команды "Закрыть задвижку чистой воды"
Ненормальное закрытие задвижки промывной воды	MclS_WW	zNKclS_CW	Стирание команды "Закрыть задвижку чистой воды"
Задвижка сырой воды открыта	DDI_X3S_DW		Запись команды "Открыть задвижку слива воды в канализацию"
Задвижка сырой воды закрыта	DDI_X4S_DW		
Задвижка чистой воды закрыта	DDI_X4S_CW	zKopS_SW	Стирание команды "Открыть задвижку слива воды в канализацию"
Задвижка слива воды в канализацию открыта	DDI_X3S_SW	zNKopS_SW	Запись команды "Открыть задвижку промывной воды"
Задвижка промывной воды открыта	DDI_X3S_WW	zKopS_WW	Стирание команды "Открыть задвижку промывной воды"
Задвижка промывной воды закрыта	DDI_X4S_WW	zNKopS_WW	Запись команды "Закрыть задвижку промывной воды"
Есть все условия для автоматического открытия задвижки сырой воды (X0 - Включено питание привода задвижки; !X2 - Не включен привод задвижки на закрытие; !X5 - Задвижка не заклинена; !HTD - Нет превышения температуры двигателя сырой воды)	CopS_DW	zKclS_WW	Стирание команды "Закрыть задвижку промывной воды"
		zNKclS_WW	Запись команды "Остановить привод задвижки сырой воды"
		zKoffS_DW	Запись признака "Нет условий для автоматического открытия задвижки сырой воды"
Есть все условия для автоматического закрытия задвижки сырой воды (X0 - Включено питание привода задвижки; !X1 - Не включен привод задвижки на открытие; !X5 - Задвижка не заклинена; !HTD - Нет превышения температуры двигателя сырой воды)	CclS_DW	zMNCopS_DW	Запись признака "Нет условий для автоматического закрытия задвижки сырой воды"
		zMNCclS_DW	Запись признака "Нет условий для автоматического закрытия задвижки чистой воды"
		zMNCclS_CW	Запись признака "Нет условий для автоматического открытия задвижки слива воды в канализацию"
Есть все условия для автоматического открытия задвижки слива воды в канализацию	CopS_SW	zMNCopS_SW	Запись признака "Нет условий для автоматического открытия задвижки промывной воды"
		zMNCopS_WW	Запись признака "Нет условий для автоматического закрытия задвижки промывной воды"
Есть все условия для автоматического закрытия задвижки промывной воды	CclS_WW	zMNCclS_WW	Начать контроль времени для аварийного выхода из промывки фильтров КО
		zstew	Начать контроль времени сброса первого фильтрата
Команда на открытие промывной задвижки по сети PROFIBUS	PBWKopS_WW		
Команда на выход из режима промывки по сети PROFIBUS	PBKoffWCC	zstspff	
Не выполнять сброс первого фильтрата	PBNoSPF		Запись признака "Промывка фильтров КО завершена"
Окончилось время контроля для аварийного выхода из промывки фильтров КО	TEW	zATWCCoff	Запись признака "Промывка фильтров КО завершена ненормально"
Окончилось время контроля сброса первого фильтрата	TSPF	zATWCCoffErr	

Рис. 11. Схема связей технологического автомата управления промывкой фильтра КО (ATWCC)

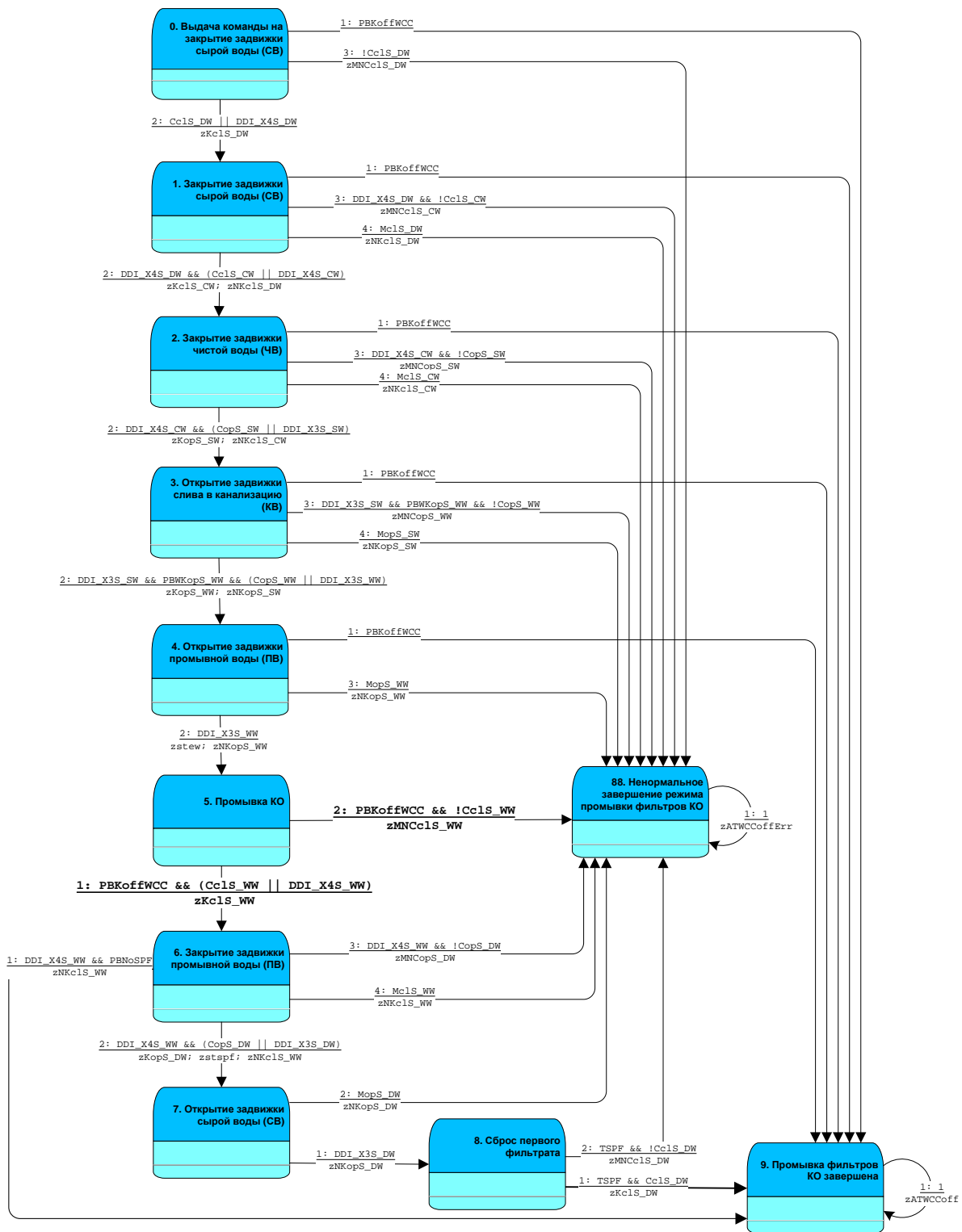


Рис. 12. Граф переходов технологического автомата управления промывкой фильтра КО (ATWCC)

5.3. Технологические автоматы (контроллер S7-200 ШАУ НА)

5.3.1. Главный технологический автомат ШАУ НА (ATMainWP)

Схема связей автомата приведена на рис. 13, а граф переходов – на рис. 14.

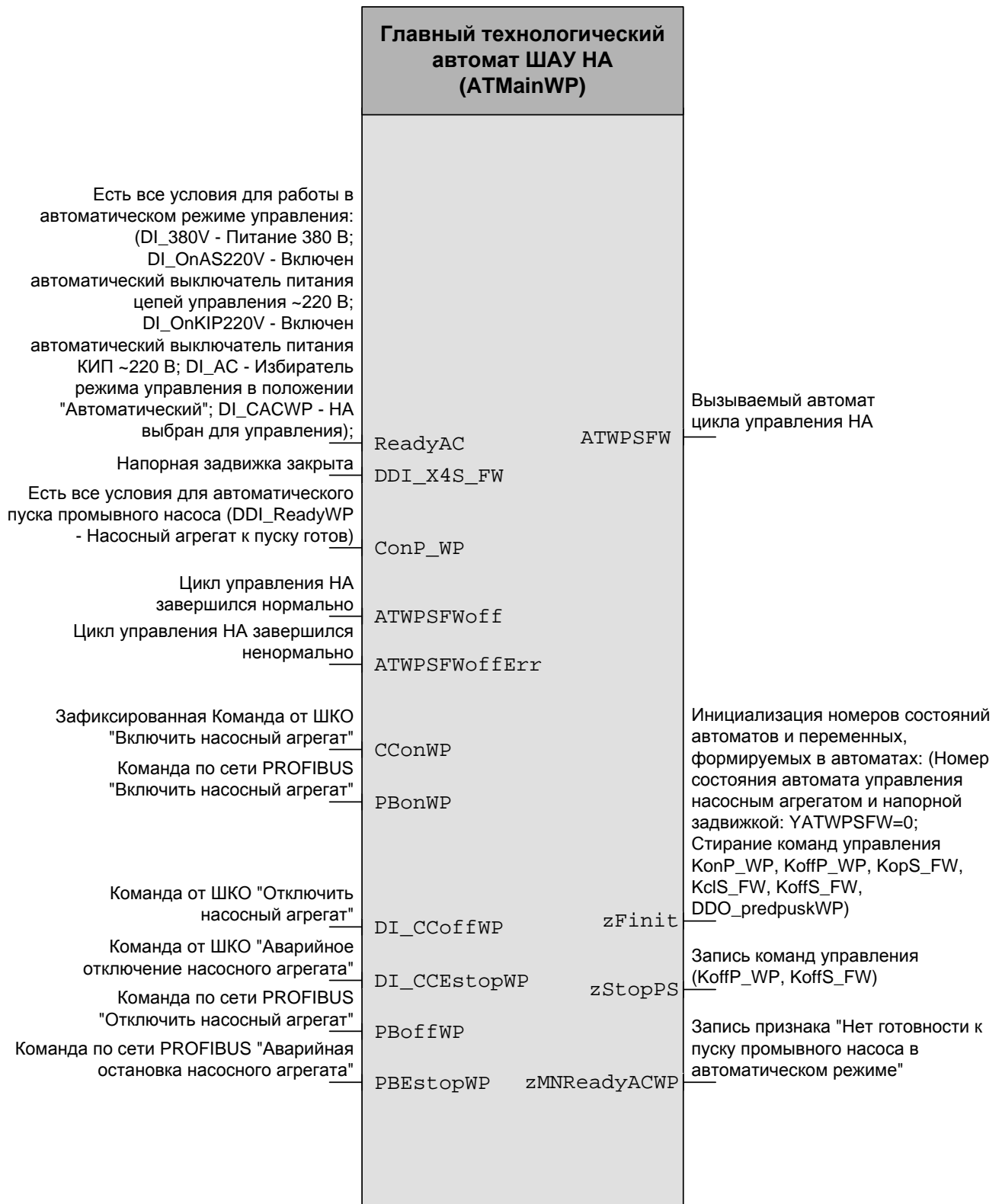


Рис. 13. Схема связей главного технологического автомата ШАУ НА (ATMainWP)

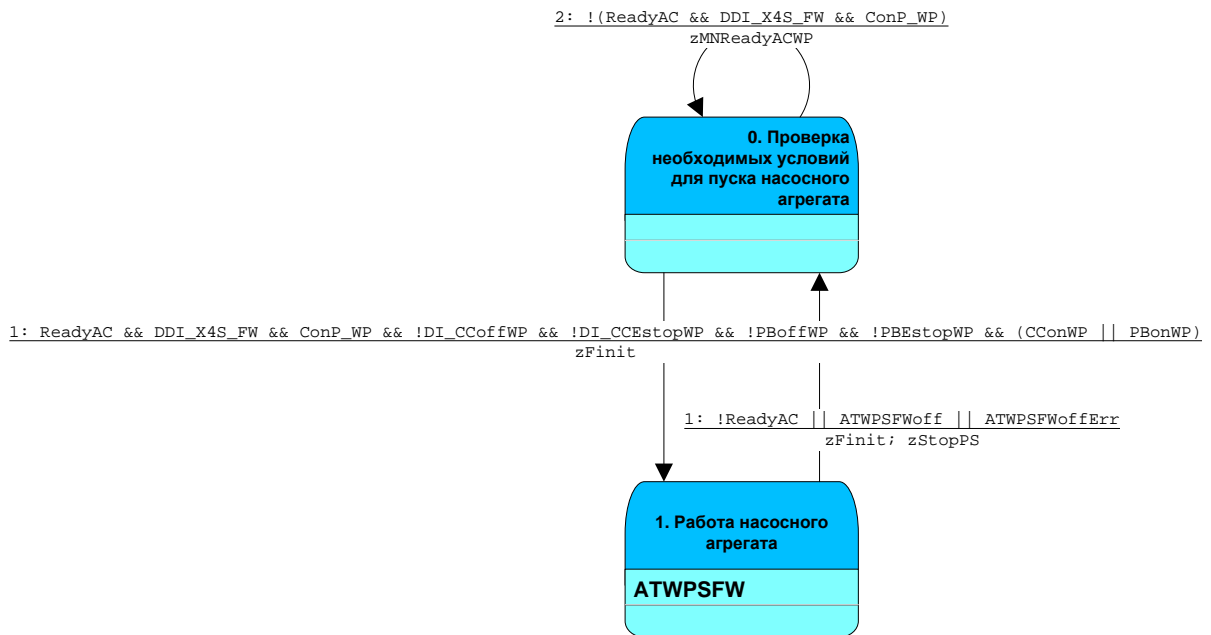


Рис. 14. Граф переходов главного технологического автомата ШАУ НА (*ATMainWP*)

5.3.2. Технологический автомат управления работой насосного агрегата (ATWPSFW)

Схема связей автомата приведена на рис. 15, а граф переходов – на рис. 16.

Технологический автомат управления работой насосного агрегата и напорной задвижки (ATWPSFW)			
Ненормальное открытие напорной задвижки	MopS_FW	zKopS_FW	Запись команды "Открыть напорную задвижку"
Ненормальное закрытие напорной задвижки	Mc1S_FW	zNKopS_FW	Стирание команды "Открыть напорную задвижку"
Напорная задвижка открыта	DDI_X3S_FW	zKc1S_FW	Запись команды "Закрыть напорную задвижку"
Напорная задвижка закрыта	DDI_X4S_FW	zNKc1S_FW	Стирание команды "Закрыть напорную задвижку"
Есть все условия для автоматического открытия напорной задвижки (X0 - Включено питание привода задвижки; !X2 - Не включен привод задвижки на закрытие; !X5 - Задвижка не заклинена; !HTD - Нет превышения температуры двигателя напорной задвижки)	CopS_FW	zKonP_WP	Запись команды "Включить привод насосного агрегата"
Есть все условия для автоматического закрытия напорной задвижки (X0 - Включено питание привода задвижки; !X1 - Не включен привод задвижки на открытие; !X5 - Задвижка не заклинена; !HTD - Нет превышения температуры двигателя напорной задвижки)	Cc1S_FW	zKoffP_WP	Запись команды "Отключить привод насосного агрегата"
Есть все условия для автоматического пуска промывного насоса (DDI_ReadyWP - Насосный агрегат к пуску готов)	ConP_WP	zMNCopS_FW	Запись признака "Нет условий для автоматического открытия напорной задвижки"
Нет включения пускателя привода промывного насоса за контрольное время	M1P_WP	zMNCc1S_FW	Запись признака "Нет условий для автоматического закрытия напорной задвижки"
Нет отключения пускателя привода промывного насоса за контрольное время	M2P_WP	zMNConP_WP	Запись признака "Нет готовности к пуску промывного насоса"
Насосный агрегат включен	DDI_X1P_WP	zDDO_predpuskWP	Запись команды "Предупреждение пуска"
Команда от ШКО "Отключить насосный агрегат"	DI_CCoffWP	zNDDO_predpuskWP	Стирание команды "Предупреждение пуска"
Команда от ШКО "Аварийное отключение насосного агрегата"	DI_CCEstopWP	zATWPSFWoff	Запись признака "Цикл управления НА завершился нормально"
Команда по сети PROFIBUS "Отключить насосный агрегат"	PBoffWP	zATWPSFWoffErr	Запись признака "Цикл управления НА завершился ненормально"
Команда по сети PROFIBUS "Аварийная остановка насосного агрегата"	PBEstopWP	zstps	Начать контроль времени предпусковой сигнализации
Окончилось время контроля предпусковой сигнализации	TPS		

Рис. 15. Схема связей технологического автомата управления работой насосного агрегата (ATWPSFW)

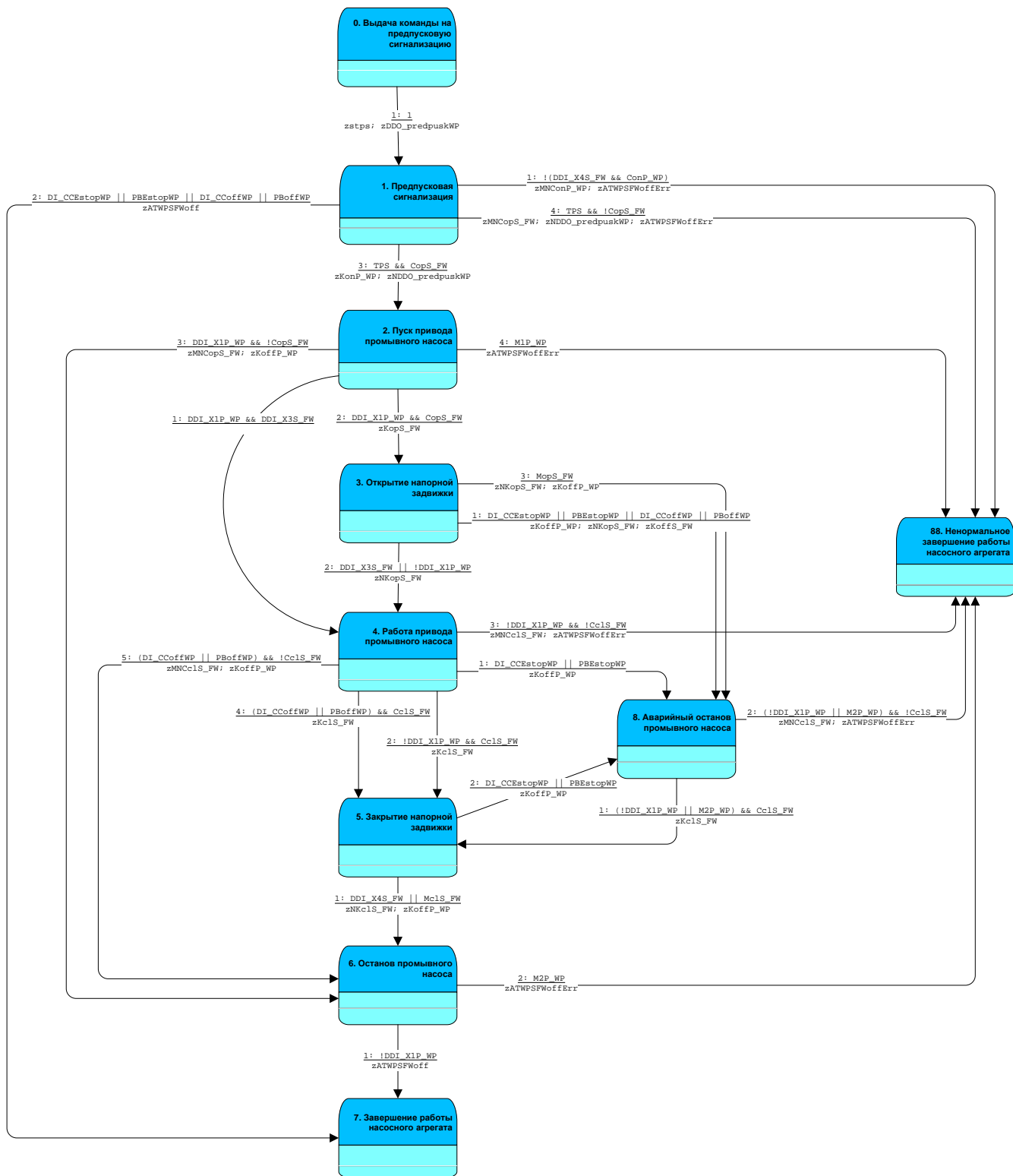


Рис. 16. Граф переходов технологического автомата управления работой насосного агрегата (ATWPSFW)

5.4. Технологические автоматы управления промывкой (Мастер-контроллер S7-300)

5.4.1. Главный технологический автомат управления (ATmain)

Схема связей автомата приведена на рис. 17, а граф переходов – на рис. 18.



Рис. 17. Схема связей главного технологического автомата управления (ATmain)

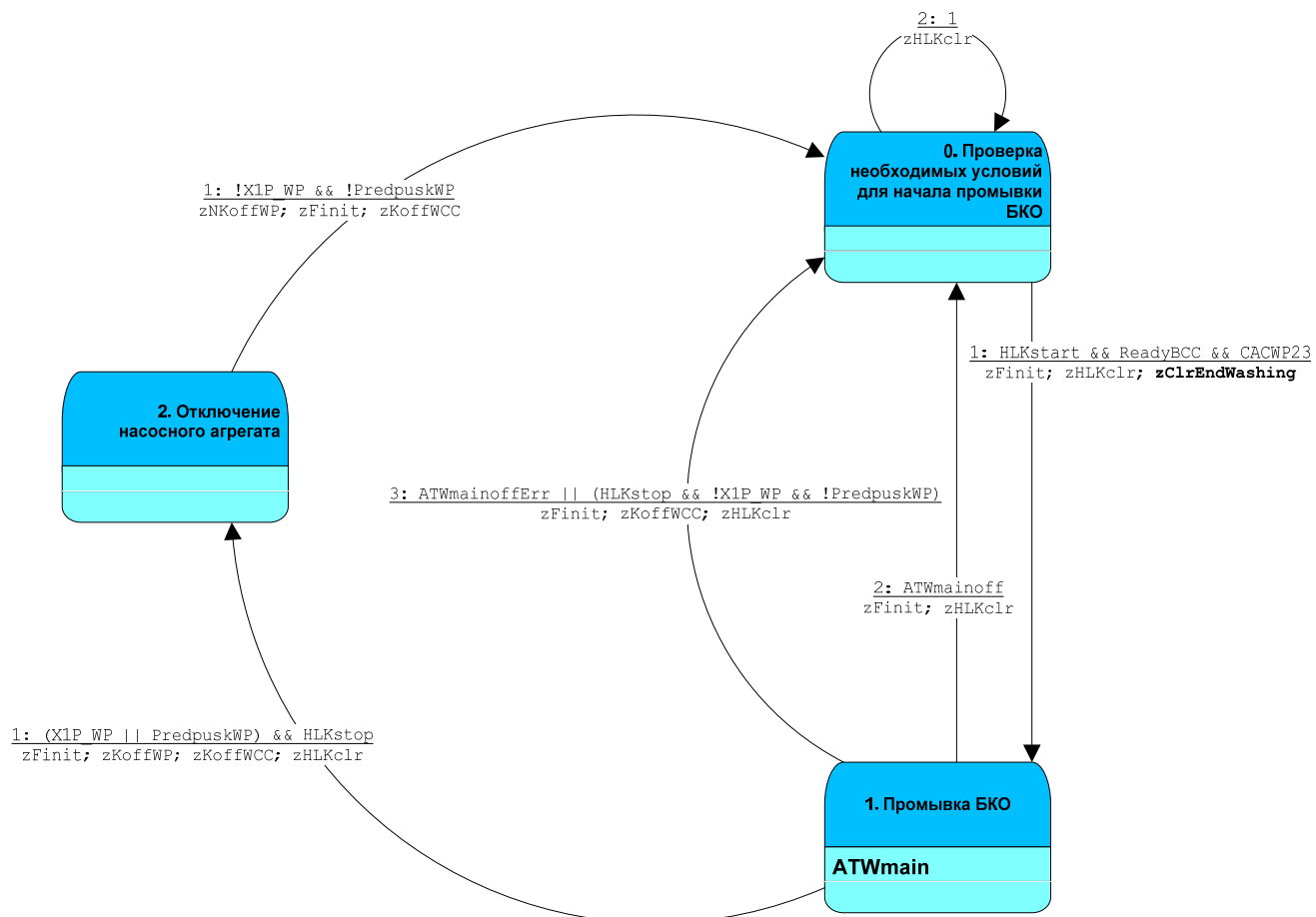


Рис. 18. Граф переходов главного технологического автомата управления (*ATmain*)

В этом графе переходов в первую вершину вложен автомат *ATWmain*.

5.4.2. Технологический автомат управления промывкой (ATWmain)

Схема связей автомата приведена на рис. 19, а граф переходов – на рис. 20.

Технологический автомат управления промывкой (ATWmain)		
Нет готовности к пуску промывного насоса в автоматическом режиме	MNReadyACWP	Вызываемый автомат управления промывкой текущего КО в очереди
	ATWCurr	Вызываемый автомат управления промывкой следующего КО в очереди
	ATWNext	
Есть все условия для работы текущего КО в автоматическом режиме (DI_380V – Питание 380В; DI_OnAS220V – Питание цепей управления ~220В; Включен автоматический выключатель питания цепей управления ~220В; DI_AC – Избиратель режима управления в положении «Автоматический», Нет NConnectCC – отсутствует признак отсутствия связи)	zKonPWCC_curr	Запись команды «Подготовить к промывке текущий КО»
	zKonPWCC_next	Запись команды «Подготовить к промывке следующий КО»
	ReadyAC_curr	Запись команды «Включить насосный агрегат»
	zKonWP	Стирание команды «Включить насосный агрегат»
	zNKonWP	
Есть все условия для работы следующего КО в автоматическом режиме (DI_380V – Питание 380В; DI_OnAS220V – Включен автоматический выключатель питания цепей управления ~220В; DI_AC – Избиратель режима управления в положении «Автоматический», Нет NConnectCC – отсутствует признак отсутствия связи)	zKoffWP	Запись команды «Отключить насосный агрегат»
	zNKoffWP	Стирание команды «Отключить насосный агрегат»
	ReadyAC_next	
Следующий осветлитель последний в очереди	LCC_next	
Текущий осветлитель последний в очереди	LCC_curr	Запись команды «Изменить индекс текущего КО на индекс следующего»
Окончилось время контроля промывки текущего КО	TW_curr	Запись команды «Увеличить индекс текущего КО на 1»
Окончилось время контроля для подготовки к промывке следующего КО	Tprep_curr	Запись команды «Увеличить индекс следующего КО на 1»
	zUpInd_next	
Задвижка промывной воды текущего КО открыта	X3S_WW_curr	Запись команды «Запомнить индекс текущего КО»
Задвижка промывной воды следующего КО открыта	X3S_WW_next	Запись команды «Запомнить индекс следующего КО»
Насосный агрегат включен	X1P_WP	
Напорная задвижка насосного агрегата открыта	X3S_FW	
Подготовка текущего КО к промывке не выполнена	ATWoffErrP_curr	zATWmainoff
Промывка текущего КО не выполнена	ATWoffErr_curr	zATWmainoffErr
Промывка текущего КО завершена	ATWoff_curr	
Подготовка следующего КО к промывке не выполнена	ATWoffErrP_next	zKOHAX_curr
Промывка следующего КО не выполнена	ATWoffErr_next	zKOHAX_next
Нет включения пускателя насосного агрегата за контрольное время	M1P_WP	zNKOHAX_curr
Ненормальное открытие напорной задвижки насосного агрегата	MopS_FW	
Признак «Не останавливать НА после завершения промывки»	NStopWP	zNKOHAX_next
Не получилось закрыть задвижку промывной воды на КО на котором произошло заклинивание задвижки промывной воды на открытие по причинам: MClS_WW – ненормальное закрытие задвижки промывной воды; MNCIS_WW – нет условий для автоматического закрытия задвижки промывной воды; отсутствие признака ReadyAC – есть все условия для работы КО в автоматическом режиме;	ErrClS_WW_mem	
Один из НА выбран для управления	CACWP23	
Предупреждение пуска промывного насоса	PredpuskWP	

Рис. 19. Схема связей технологического автомата управления промывкой (ATWmain)

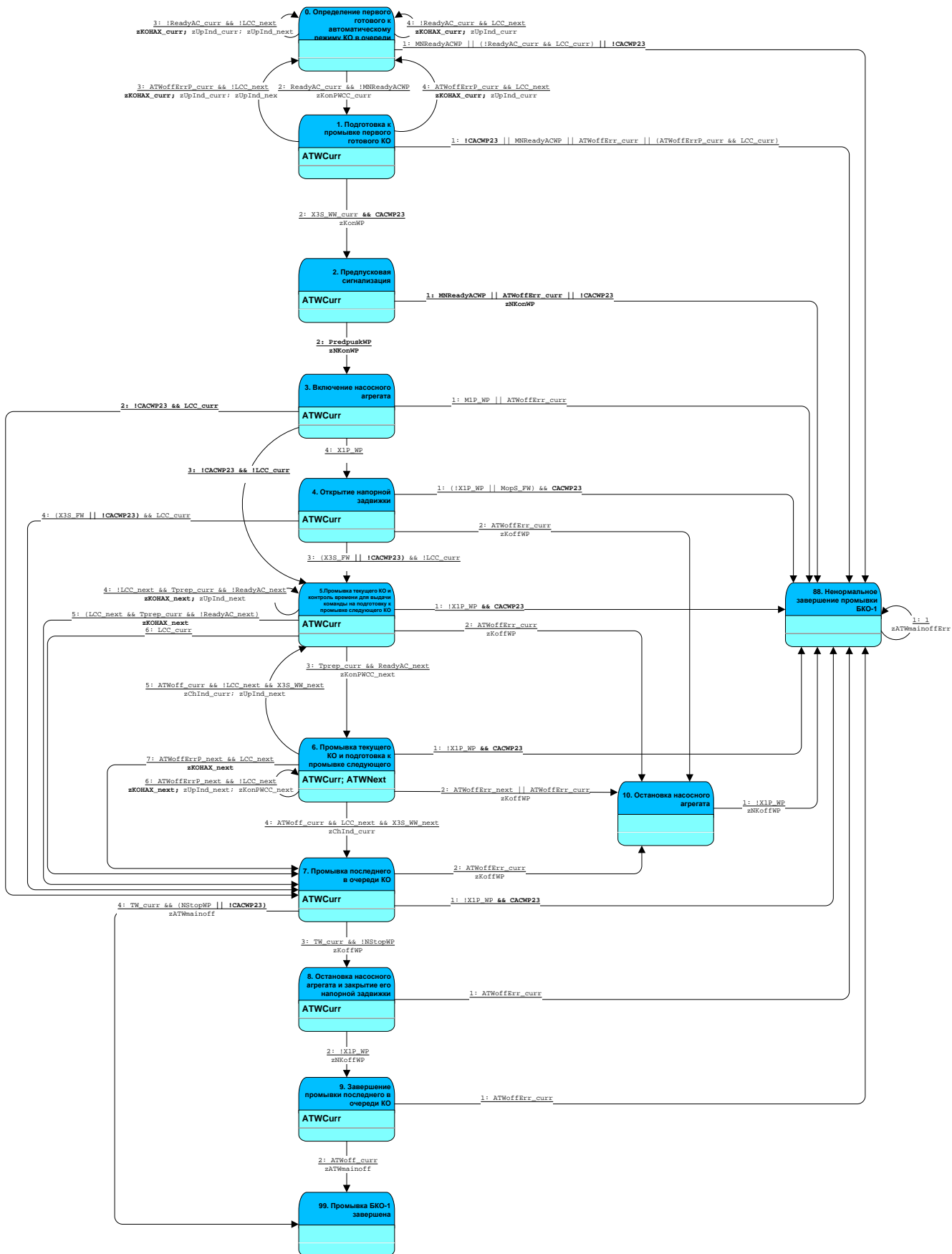


Рис. 20. Граф переходов технологического автомата управления промывкой (ATWmain)

5.4.3. Автомат управления промывкой осветлителя в очереди (ATWx)

Схема связей автомата приведена на рис. 21, а граф переходов – на рис. 22.

Автомат управления промывкой осветлителя в очереди (ATWx)		
Текущий КО первый в очереди	FCC	
Текущий КО последний в очереди	LCC	
Отсутствует связь с контроллером КО	NConnectCC	zNKonPWCC
Один из НА выбран для управления	CACWP23	zWKopS_WW
Есть все условия для работы КО в автоматическом режиме (DL_380V – Питание 380В; DL_OnAS220V – Включен автоматический выключатель питания цепей управления ~220В; DL_AC – Избиратель режима управления в положении «Автоматический», Нет NConnectCC – отсутствует признак отсутствия связи)	ReadyAC	zNWKopS_WW
Команда «Открыть промывную задвижку» (для следующего КО)	WKopS_WW_next	zKoffWCC
Задвижка сырой воды закрыта	X4S_DW	zNKoffWCC
Задвижка чистой воды закрыта	X4S_CW	
Задвижка промывной воды открыта	X3S_WW	
Задвижка промывной воды закрыта	X4S_WW	
Задвижка канализационной воды открыта	X3S_SW	
Насосный агрегат включен	X1P_WP	
Напорная задвижка насосного агрегата открыта	X3S_FW	
Задвижка сырой воды следующего КО закрыта	X4S_DW_next	
Задвижка чистой воды следующего КО закрыта	X4S_CW_next	
Задвижка канализационной воды следующего КО открыта	X3S_SW_next	
Окончилось время контроля для выдачи команды на подготовку следующего КО	Tprep	zstprep
Окончилось время контроля промывки	TW	zstw
Ненормальное закрытие задвижки сырой воды	MclS_DW	zATWoffErrP
Ненормальное закрытие задвижки чистой воды	MclS_CW	zATWoffErr
Ненормальное открытие задвижки канализационной воды	MopS_SW	
Ненормальное открытие задвижки промывной воды	MopS_WW	
Ненормальное закрытие задвижки промывной воды	MclS_WW	
Нет условий для автоматического закрытия задвижки сырой воды	MNCclS_DW	
Нет условий для автоматического закрытия задвижки чистой воды	MNCclS_CW	
Нет условий для автоматического открытия задвижки канализационной воды	MNCopS_SW	
Нет условий для автоматического открытия задвижки промывной воды	MNCopS_WW	
Нет условий для автоматического закрытия задвижки промывной воды	MNCclS_WW	
Не выполнена подготовка следующего КО к промывке	ATWoffErrP_next	
Следующий КО стал текущим (YATWMain=5, должен учитываться в программе только для следующего КО, а для текущего КО должен быть TRUE)	Next_to_Curr	

Рис. 21. Схема связей автомата управления промывкой осветлителя в очереди (ATWx)

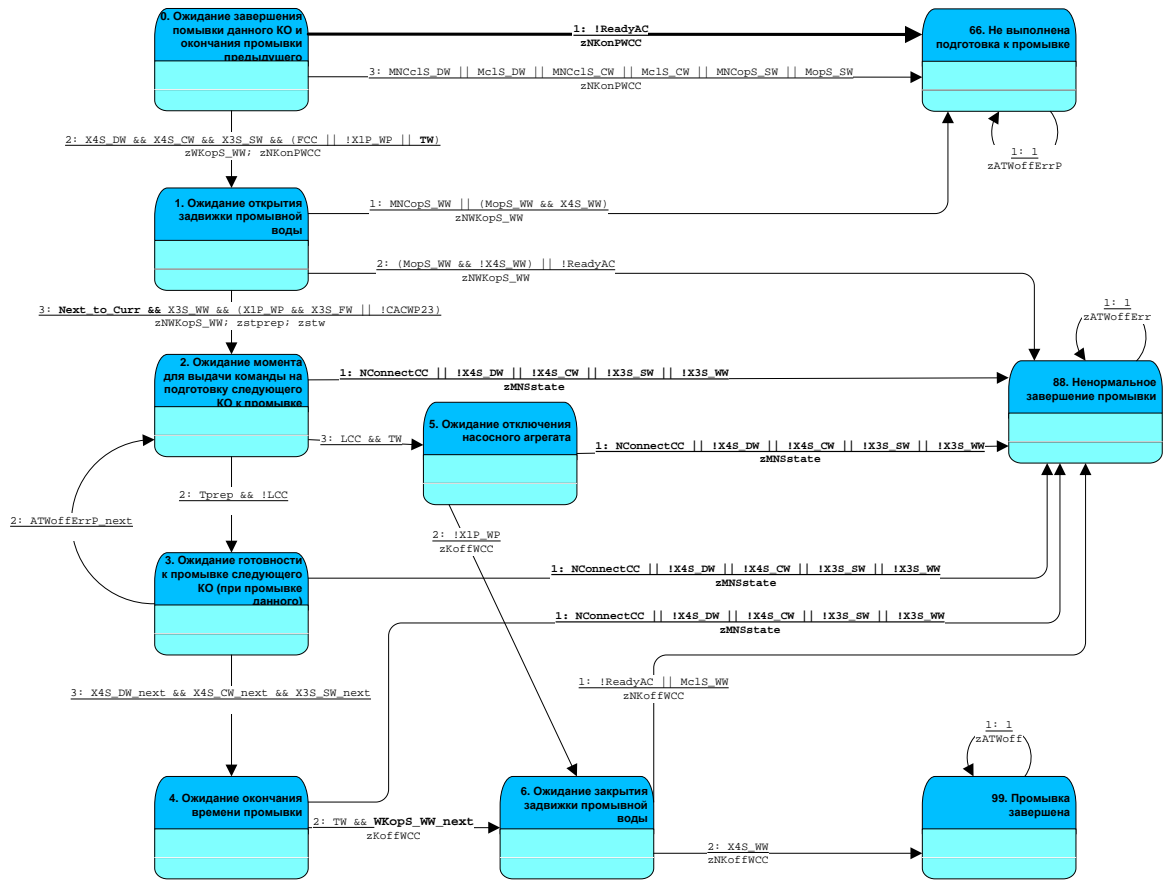


Рис. 22. Граф переходов автомата управления промывкой осветителя в очереди (ATWx)

6. Реализация алгоритмов

В соответствии со *SWITCH-технологией* процедура реализации программы предполагает **формальное и изоморфное программирование по заданным алгоритмам**. Алгоритм реализуется весьма просто: “вызвал – проверил – если выполнены необходимые условия, то сделал что надо (включая переход) – вышел”. Такие однородные действия позволяют не только выполнять программирование формально, но и **автоматически создавать код программы по графам переходов автоматов [1]**.

Подход к реализации автоматов в пакете *STEP 7* (стандартной среды для контроллеров *S7-300/400*) ничем не отличается от подхода, подробно изложенного в работе [1]. Единственным отличием является применение языка *SCL*, а не языка *STL*. Выбор языка *SCL* (заказываемого дополнительно к пакету *STEP 7*) обусловлен чисто практическим соображением – удобством работы с языком высокого уровня, так как язык *SCL* является Паскале-подобным.

В отличие от языка *STL* в языке *SCL* есть оператор, аналогичный оператору `switch` языка *C*. Это оператор **CASE**.

Код программы, реализующей произвольный автомат, состоит из нескольких частей:

- обнуление (сброс) команд, признаков и т.п., используемых в автомате, но не имеющих обнуления по условию;
- проверка условий и действий в состоянии 0;
- ...
- проверка условий и действий в состоянии N;
- обнуление (сброс) признаков вызова вложенных автоматов (если они используются);
- вызов вложенных автоматов, если текущее состояние равно 0;
- ...
- вызов вложенных автоматов, если текущее состояние равно N.

При этом шаблон для реализации произвольного автомата на языке *SCL* имеет следующий вид:

```
// Обнуление (сброс) команд, признаков и т.п., записываемых в автомате
Команда, признак (или признак выполнения процедуры над признаком_1) := FALSE;
...
Команда, признак (или признак выполнения процедуры над признаком_m) := FALSE;

// Проверка условий перехода
CASE номер_состояния OF
  Значение_0 : IF условие_1 THEN признак_выполнения_процедуры := TRUE;
               признак_выполнения_процедуры := TRUE;
               номер_состояния := новое_значение;

               ELSE IF условие_2 THEN признак_выполнения_процедуры := TRUE;
               признак_выполнения_процедуры := TRUE;
               номер_состояния := новое_значение;

               END_IF; END_IF;
  . . .
  Значение_n : IF условие_3 THEN признак_выполнения_процедуры := TRUE;
               признак_выполнения_процедуры := TRUE;
               номер_состояния := новое_значение;

               ELSE IF условие_4 THEN признак_выполнения_процедуры := TRUE;
               признак_выполнения_процедуры := TRUE;
               номер_состояния := новое_значение;

               END_IF; END_IF;

END_CASE;

// Обнуление признаков вызова вложенных автоматов
признак_вызова_вложенного_автомата_1 := FALSE;
. . .
признак_вызова_вложенного_автомата_n := FALSE;

// Вызов вложенных автоматов в Состоянии
CASE номер_состояния OF
  текущее_к_этому_моменту_значение: признак_вызова_вложенного_автомата_1 := TRUE;
  . . .
  текущее_к_этому_моменту_значение: признак_вызова_вложенного_автомата_n := TRUE;
ELSE: ;
END_CASE;
```

Как следует из приведенного шаблона, структура кода строгая и позволяет реализовать произвольный автомат. Программирование выполняется последовательно – состояние за состоянием, условие за условием и т.д. В итоге получается код программы, **ПОЛНОСТЬЮ соответствующий (изоморфный) графу переходов автомата (алгоритму).**

Если, а это **НАСТОЯТЕЛЬНО РЕКОМЕНДУЕТСЯ**, автоматизировать построение кода программы по графу переходов, то при тестировании не будет необходимости вообще заглядывать в текст программы. При этом надо будет только исправить граф переходов автомата и сгенерировать новый код.

Примеры сгенерированного кода приведены в **Приложении 2**.

Для завершения построения программы осталось проделать несложные операции над выходными переменными по сформированным в автомате признакам. Для того чтобы унифицировать общую структуру программы рекомендуется производить вызов автомата в отдельной подпрограмме. Там же можно выполнить операции над выходными переменными, операции с таймерами и вызов вложенных автоматов.

В методике для контроллера *S7-200* [1] предлагалось выделять по способу реализации следующие типы алгоритмов управления:

- **непосредственный**, который использует и формирует непосредственные данные, команды и сигналы;
- **подпрограмма** – алгоритм, использующий значения входных формальных параметров и формирующий значения выходных формальных параметров.

Строго говоря, в данном проекте **автоматы *ATmain*** (Главный технологический автомат управления) и ***ATWmain*** (Технологический автомат управления промывкой) являются **непосредственными**, а автомат ***ATWx*** (Автомат управления промывкой осветлителя в очереди) – **подпрограммой**.

Однако при использовании языка ***SCL*** (впрочем, как и любого другого языка высокого уровня) обычно применяются сложные структуры данных (для удобства). Соответственно обозначение переменной в тексте программы получается слишком длинным, и использовать такое обозначение в алгоритме чаще всего нереально (в отличие от программ на языке ***STL*** для контроллеров *S7-200*). Поэтому реализовывать все вызовы автоматов при применении языка ***SCL*** предлагается как стандартный вызов подпрограммы с передачей в нее значений непосредственных данных (переменных).

На языке ***SCL*** **шаблон вызова** выглядит следующим образом:

```
FUNCTION CALL_FB_обозначение_автомата:VOID

// Собственно вызов подпрограммы автомата
FB_обозначение_автомата.DB_обозначение_автомата(
    формальная_входная_переменная_1 := фактическая_переменная_1,
    . . .
    формальная_входная_переменная_n := фактическая_переменная_m
);

// Для ПРОТОКОЛИРОВАНИЯ
// для формирования очередной записи в протокол
// должно быть выполнено одно из следующих условий:
// - изменение номера состояния автомата;
// - изменение выходного воздействия, формируемого автоматом
//
IF одно_из_вышеуказанных_условий_выполнено THEN
    Процедура_формирования_очередной_записи_протокола
END_IF;

//Обработка результатов работы автомата.
//Формирование значений фактических переменных по признакам,
//сформированным в автомате
IF признак_выполнения_процедуры_над_фактической_переменной_1 THEN
    фактическая_переменная_1 := новое_значение;
END_IF;
. . .
IF признак_выполнения_процедуры_над_фактической_переменной_n THEN
    фактическая_переменная_n := новое_значение;
END_IF;

//Обработка результатов работы автомата.
//Вызов вложенных автоматов
IF признак_вызова_вложенного_автомата_1 THEN
    Вызов_вложенного_автомата_1;
END_IF;
. . .
IF признак_вызова_вложенного_автомата_n THEN
    Вызов_вложенного_автомата_n;
END_IF;

END_FUNCTION
```

Несколько слов по поводу операций с фактическими данными. Здесь действует следующее правило (аналогичное приведенному в методике для контроллеров *S7-200*): если переменная записывается и стирается, то перед присвоением нового значения необходимо следует проверить что **признак выполнения процедуры над фактической переменной** имеет значение TRUE. Если переменная только записывается, то значению **фактической переменной** присваивается значение **признака выполнения процедуры над фактической переменной**.

Значение **признака выполнения процедуры над такой фактической переменной** безусловно стирается в начале процедуры, реализующей автомат.

Примеры вызова подпрограмм автоматов приведены в **Приложении 3**.

Заключение

Разработка автоматов и программы для *S7-300* заняла две недели (**включая изучение совершенно незнакомого пакета *STEP 7* и языка *SCL***).

При этом отметим, что если при программировании контроллеров *S7-200* для реализации задач, описанных в методике [1], применялся язык инструкций *STL* (единственный текстовый язык, доступный в среде *MicroWin32*), то для контроллеров *S7-300/400* удобнее применять структурированный язык более высокого уровня – язык ***SCL***.

Применение автоматного подхода и принципа функционального разделения автоматов управления (на базовые и технологические) на локальном уровне позволяют легко организовать взаимодействие локальных систем.

Автоматный подход предоставляет разработчику программы организационного уровня (в данном проекте – это программа Мастер-контроллера *S7-300*) **эффективную технологию построения алгоритмов. В свою очередь, функциональное разделение** позволяет использовать те же данные о поведении объектов (данные, формируемые в базовых алгоритмах локальной системы), что и технологические алгоритмы локальной системы. Таким образом, **осуществляется своеобразная “синхронизация” работы** не только организующей программы с локальной, но и как следствие – **корректное взаимодействие нескольких локальных программ**.

Приложение 1. Техническое задание “Процесс промывки БКО в автоматическом режиме”

П 1.1. Общие сведения

В работу автоматической системы управления процессом промывки (АСУ ТП промывки) блока контактных осветлителей заложен алгоритм промывки, рассчитанный на основании технических и технологических параметров блока.

П 1.2. Описание алгоритма промывки

Алгоритм промывки блока контактных осветлителей (рис. П1) рассчитывается таким образом, чтобы обеспечить равномерную подачу промывной воды работающим промывным насосом. Порядок и время открытия и закрытия технологических задвижек на контактных осветлителях (КО) должны направлять поток промывной воды на один из промываемых КО, а при завершении промывки этого КО, плавно перенаправляется (перераспределяется) на следующий КО, стоящий в очереди на промывку.

Участие КО в процессе промывки включает в себя подготовку, вход в промывку (открытие задвижки промывной воды), собственно промывку и выход из промывки. Подготовительный период включает в себя время закрытия задвижки сырой воды, время закрытия задвижки чистой воды и время открытия задвижки канализационной воды – занимает по длительности время хода трех задвижек конкретного КО. Промывка КО идет при полностью закрытых задвижках сырой и чистой воды, и полностью открытых задвижках промывной и канализационной воды. По истечении заданного времени промывки (или по истечении времени аварийного выхода из промывки) закрывается задвижка промывной воды. После её закрытия команда на закрытие задвижки канализационной воды подается с задержкой на время, необходимое для сброса первого фильтрата (слив которого в канализационный канал производится при открытой задвижке сырой воды). По окончании сброса фильтрата одновременно закрываются задвижка сырой воды и задвижка канализационной воды, затем открывается задвижка чистой воды, и когда она откроется, открывается задвижка сырой воды. Контактный осветлитель переходит в рабочий режим фильтрации.

Для того чтобы обеспечить непрерывную и равномерную подачу промывной воды при переходе с одного КО на другой, необходимо одновременно начинать закрывать задвижку промывной воды КО, завершающего промывку, и открывать задвижку промывной воды следующего КО, начинающего промывку. Однако к этому времени следующий КО уже должен пройти подготовительный процесс закрытия задвижки сырой воды, закрытия задвижки чистой воды и открытия задвижки канализационной воды. Таким образом, заготовка следующего КО в промывку начинается заранее, за время равное сумме расчетного времени хода сырой, чистой и канализационной задвижек следующего КО до завершения промывки КО предыдущего. Максимальная величина времени на заготовку составляет сумму максимального времени хода сырой, чистой и канализационной задвижек КО (максимальное время хода задвижек равно времени контроля открытия/закрытия, индивидуального для каждой задвижки).

Промывной насос включается после того, когда первый в очереди на промывку КО пройдет период заготовки и промывная задвижка откроется. Включение промывного насоса предваряется звонком громкого боя. Напорная задвижка открывается после включения промывного насоса.

Когда закончится время промывки последнего в очереди КО, закрывается напорная задвижка, и выключается промывной насос. После этого закрывается промывная задвижка КО и продолжается выход КО из промывки.

Алгоритм промывки предусматривает возможность не останавливать насосный агрегат после окончания очереди КО, заказанных для автоматической промывки, для того, чтобы продолжить промывку в режиме ручного управления.

Если промывной насос решено не останавливать после завершения промывки, то после окончания времени промывки последнего КО, промывной насос остается включенным, напорная задвижка и промывная задвижка на последнем в очереди КО остается открытой. Оператору выдается сообщение «Промывка завершена. НА не выключен».

Положение технологических задвижек промываемых КО должно соответствовать алгоритму промывки.

В режиме фильтрации нормальное положение задвижек промывной и канализационной воды закрытое, а задвижек сырой и чистой воды – открытое.

Для расчета алгоритма промывки блока контактных осветлителей необходимы следующие параметры:

- технические параметры, задаваемые техническим персоналом;
- время хода задвижки сырой воды максимальное каждого КО;
- время хода задвижки чистой воды максимальное каждого КО;
- время хода задвижки промывной воды максимальное каждого КО;
- время хода задвижки канализационной воды максимальное каждого КО.

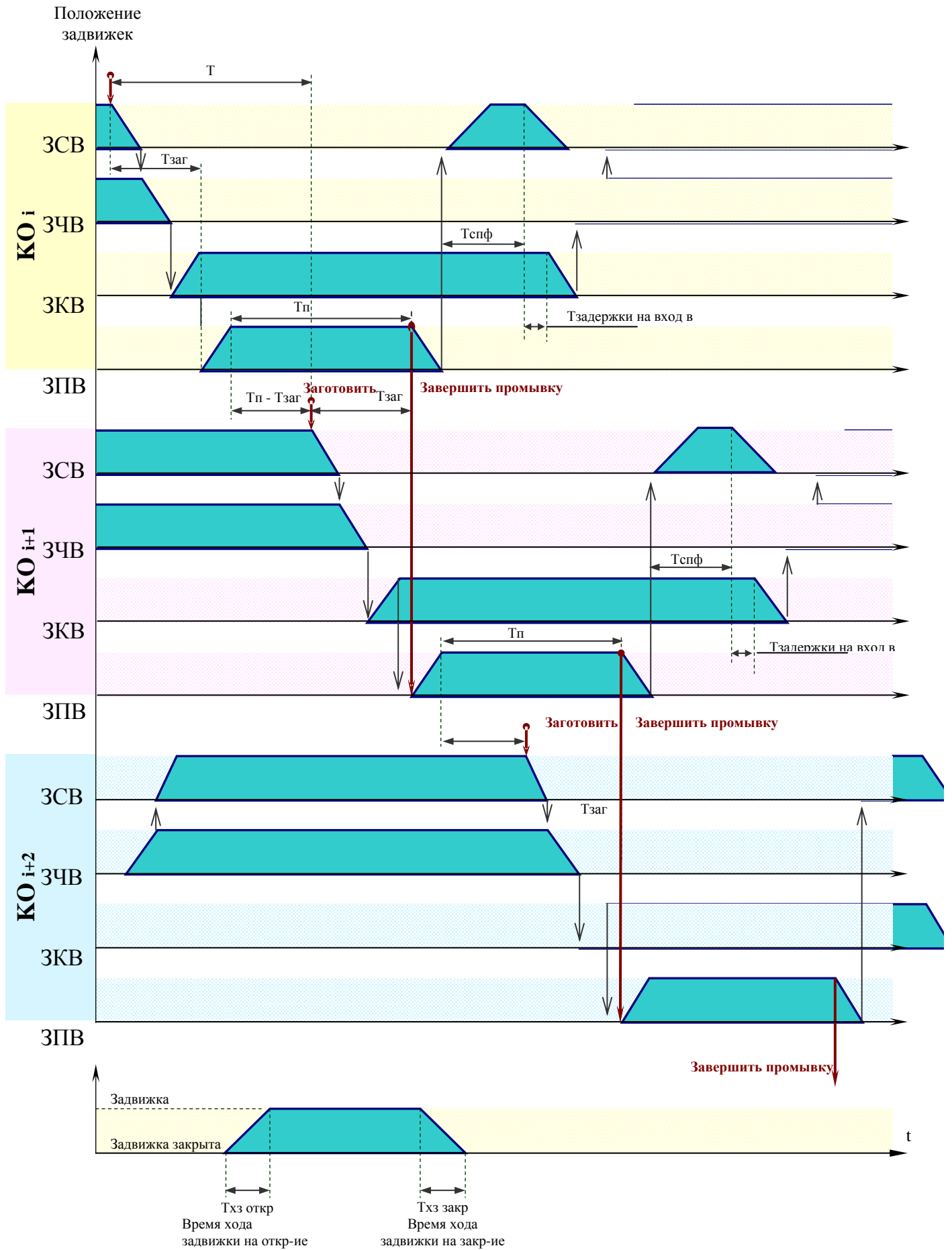


Рис. П1. Временная диаграмма промывки блока КО-1

Параметры режима промывки, задаваемые технологическим персоналом:

- номер промывного насоса (установкой переключателя на панели выбора промывного насоса в соответствующее положение);
- перечень промываемых КО в порядке их ввода в промывку;
- время промывки каждого КО (индивидуальное или общее);
- время сброса фильтрата каждого КО (индивидуальное или общее).

Для управления процессом промывки необходима выдача управляющих сигналов:

- открыть задвижку сырую / чистую / промывную / канализационную / напорную;
- закрыть задвижку сырую / чистую / промывную / канализационную / напорную;
- включить промывной насос;
- выключить промывной насос,

и получение контрольных сигналов:

- задвижка открывается сырая / чистая / промывная / канализационная / напорная;
- задвижка открыта сырая / чистая / промывная / канализационная / напорная;
- задвижка закрывается сырая / чистая / промывная / канализационная / напорная;
- задвижка закрыта сырая / чистая / промывная / канализационная / напорная;
- насос включен,
- насос выключен.

Также выполняется контроль за возникновением аварийных сигналов:

- подтопление;
- отключение привода (насоса или задвижки) по внешней причине.

До начала промывки блока контактных осветлителей в автоматическом режиме производится контроль готовности к промывке блока и КО, назначенных в промывку, по следующим параметрам:

- есть готовность к пуску промывного насоса в автоматическом режиме;
- один из промывных насосов выбран для управления;
- есть связь с контроллером выбранного промывного насоса;
- есть готовность к автоматическому управлению и готовность всех задвижек у всех КО, назначенных в промывку;
- закрыта ЗПВ у всех КО, назначенных в промывку;
- все КО, назначенные в промывку, находятся в режиме фильтрации;
- все КО, назначенные в промывку, не в ремонте.

Параметр «Готовность НА к пуску в автоматическом режиме» означает:

- данный НА выбран для управления;
- есть связь с контроллером;
- питание ~380В;
- включен автоматический выключатель «Питание цепей управления ~220В»;
- режим работы «Автоматический»;
- НА готов к пуску (сигнал от подстанции 6 кВ);
- напорная задвижка закрыта.

Параметр «Готовность КО к автоматическому управлению (АУ)» означает:

- есть связь с контроллером;
- питание ~380В;
- включен автоматический выключатель «Питание цепей управления ~220В»;
- режим работы «Автоматический»;

Параметр «Нет условий для открытия/закрытия задвижки» означает:

- не включен автоматический выключатель питания задвижки;
- задвижка заклинена;
- перегрев двигателя задвижки;
- задвижка движется в противоположном направлении.

Параметр «Ненормально открытие/закрытие задвижки» означает:

- нет включения пускателя за контрольное время;
- нет открытия/закрытия за контрольное время;
- отключение пускателя по внешним причинам.

Автоматическая система управления процессом промывки должна контролировать прохождение управляющих сигналов. Время ожидания открытия или закрытия задвижки после выдачи управляющего воздействия принимается равным максимальному времени хода задвижки, задаваемому индивидуально.

Любое отклонение от заданного режима считается нештатной ситуацией и требует анализа и специальной обработки, нарушающей прямой ход алгоритма промывки.

Система должна соответственно реагировать на возникновение нештатных ситуаций:

- включить звуковую сигнализацию;
- выдать сообщение о возникновении нештатной ситуации оператору;
- зарегистрировать нештатную ситуацию в журнале аварийных сообщений;
- по результату анализа события запустить алгоритм обработки конкретной ситуации.

Нештатные ситуации, приводящие к остановке НА или требующие вмешательства оператора в ход промывки, должны отображаться во всплывающем окне сообщений, с регистрацией в журнале промывки. При этом, установленный признак «Не останавливать НА после завершения промывки» не учитывается.

II 1.3. Нештатные ситуации

Перечень возможных нештатных ситуаций и алгоритм их обработки приведен в таблице 1.

Таблица 1

Нештатная ситуация	Этап промывки	Алгоритм обработки	Сообщение
Отсутствует связь с контроллером НА	Промывка в целом	Контроль за работой НА возлагается на оператора. При завершении промывки БКО насос необходимо отключить вручную (кнопками от ШАУ КО или ШАУ НА).	Постоянное отображение сообщения: «Отсутствует связь с НА. Контроль работы НА возлагается на оператора. При завершение промывки БКО насос необходимо отключить вручную».
Отсутствует готовность НА к пуску в автоматическом режиме или НА не включается	Подготовка к промывке первого КО	Первый КО переходит в режим фильтрации. Промывка не начинается.	
Отключение НА по внешним причинам	Промывка в целом	Напорная задвижка НА закрывается. КО, участвующие в промывке, переходят в режим фильтрации.	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.
НА не отключается		КО, участвующие в промывке, остаются в текущем состоянии.	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.
НЗ не открылась		Остановка НА. КО, подготовленный к промывке, переходят в режим фильтрации.	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.

Продолжение таблицы 1

Нештатная ситуация	Этап промывки	Алгоритм обработки	Сообщение
НЗ не закрылась			Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.
Подтопление любого КО	Промывка в целом	Аварийное отключение НА, напорная задвижка закрывается после отключения НА, открытие ЗКВ -релейная автоматика. КО, участвующие в промывке, переходят в режим фильтрации.	«Промывка в автоматическом режиме прервана из-за подтопления КО »
Нет готовности КО к АУ или нет условий для закрытия или Произошло ненормальное закрытие ЗСВ и ЗЧВ или нет условий для открытия или произошло ненормальное открытие ЗКВ	Подготовка к промывке последнего в очереди КО	Данный КО переходит в режим фильтрации и начинается подготовка к промывке следующего КО.	
	Подготовка к промывке не последнего в очереди КО	Данный КО переходит в режим фильтрации, исключается из очереди. Промываемый КО считается последним в очереди.	
Нет готовности КО к АУ или ЗПВ остановилась в промежуточном положении или не открылась за контрольное время	Вход в промывку (открытие ЗПВ)	Штатная остановка НА (с закрытием напорной задвижки). КО, участвующие в промывке переходят в режим фильтрации.	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.

Нештатная ситуация	Этап промывки	Алгоритм обработки	Сообщение
Нет условий для открытия ЗПВ или привод ЗПВ не включился на открытие (т.е. ЗПВ закрыта)	Вход в промывку (открытие ЗПВ) не последнего в очереди КО	Данный КО переходит в режим фильтрации и начинается подготовка к промывке следующего КО.	
	Вход в промывку (открытие ЗПВ) последнего в очереди КО	Штатная остановка НА (с закрытием напорной задвижки). КО, участвующие в промывке, переходят в режим фильтрации.	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.
Положение задвижек не соответствует режиму промывки или нет связи с контроллером	Промывка конкретного КО	Штатная остановка НА (с закрытием напорной задвижки). КО, участвующие в промывке переходят в режим фильтрации.	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.
Нет готовности КО к АУ или произошло ненормальное закрытие ЗПВ	Выход из промывки конкретного КО (закрытие ЗПВ) последнего в очереди КО	Штатная остановка НА (с закрытием напорной задвижки).	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.
	Выход из промывки конкретного КО (закрытие ЗПВ) не последнего в очереди КО	Штатная остановка НА (с закрытием напорной задвижки). Остальные КО, участвующие в промывке, переходят в режим фильтрации.	Промывка в автоматическом режиме прервана из-за возникшей нештатной ситуации.

Приложение 2. Примеры сгенерированного кода автоматов проекта

П 2.1. Главный технологический автомат управления *ATmain* (рис. 17, 18)

```
//*****  
//  
// СОЗДАНО с использованием КОНВЕРТОРА  
// графа переходов (нарисованного в Visio) в текст программы  
//  
//*****  
  
FUNCTION_BLOCK FB_ATmain //Главный технологический автомат управления  
VAR_INPUT  
    ATWmainoff:BOOL; //Промывка БКО завершена нормально  
    ATWmainoffErr:BOOL; //Промывка БКО завершена ненормально  
    ReadyBCC:BOOL; //Есть готовность БКО к промывке  
    CACWP23:BOOL; //Один из промывных насосов выбран для управления  
    HLKstart:BOOL; //Команда с верхнего уровня "Начать промывку БКО"  
    HLKstop:BOOL; //Команда с верхнего уровня "Остановить промывку БКО"  
    X1P_WP:BOOL; //Насосный агрегат включен  
    PredpuskWP:BOOL; //Предупреждение пуска промывного насоса  
    NStopWP:BOOL; //Признак «Не останавливать НА после завершения промывки»  
END_VAR  
  
VAR_OUTPUT  
    ATWmain:BOOL; //Вызываемый технологический автомат управления промывкой  
    zKoffWP:BOOL; //Запись команды "Отключить насосный агрегат"  
    zNKoffWP:BOOL; //Стирание команды "Отключить насосный агрегат"  
    zKoffWCC:BOOL; //Запись команды KoffWCC всех КО  
    zFinit:BOOL; //Инициализация номеров автоматов и переменных, формируемых в автоматах:  
                //Номера состояний приводятся к начальным (YATWmain:=0; все YATWx:=0),  
                //стираются признаки окончания технологических автоматов  
                //(ATWmainoff; ATWmainoffErr; все ATWoff; все ATWoffErr; все ATWoffErrP),  
                //стираются команды управления (все KonPWCC; все WKops_WW; все KoffWCC;  
                //все KonWP; все KoffWP),  
                //индексы текущего и следующего КО приводятся к начальным значениям  
                //(IndexCurr := 1, IndexNext := 2)  
    zHLKclr:BOOL; //Стирание команд HLKstart и HLKstop  
    zClrEndWashing:BOOL; // - Сброс признака состояния окончания предыдущей промывки  
  
END_VAR  
  
VAR_IN_OUT  
    YATmain:INT; //Номер состояния главного технологического автомата управления  
END_VAR  
  
//  
// Обнуление (сброс) команд, признаков и т.п., записываемых в автомате  
//  
zClrEndWashing := FALSE;  
zFinit := FALSE;  
zHLKclr := FALSE;  
zKoffWCC := FALSE;  
zKoffWP := FALSE;  
zNKoffWP := FALSE;  
  
// Реализация графа переходов  
  
CASE YATmain OF  
    0: IF ( HLKstart AND ReadyBCC AND CACWP23) THEN  
        zFinit := TRUE; zHLKclr := TRUE; zClrEndWashing := TRUE; YATmain := 1;  
    ELSE IF ( 1) THEN zHLKclr := TRUE; END_IF; END_IF;  
    1: IF ( (X1P_WP OR PredpuskWP) AND HLKstop) THEN  
        zFinit := TRUE; zKoffWP := TRUE; zKoffWCC := TRUE; zHLKclr := TRUE; YATmain := 2;  
    ELSE IF ( ATWmainoff ) THEN YATmain := 0; zFinit := TRUE; zHLKclr := TRUE;  
    ELSE IF ( ATWmainoffErr OR (HLKstop AND NOT X1P_WP AND NOT PredpuskWP) ) THEN  
        zFinit := TRUE; zKoffWCC := TRUE; zHLKclr := TRUE; YATmain := 0;  
    END_IF; END_IF; END_IF;  
    2: IF ( NOT X1P_WP AND NOT PredpuskWP) THEN  
        zNKoffWP := TRUE; zFinit := TRUE; zKoffWCC := TRUE; YATmain := 0; END_IF;  
    ELSE: ;  
END_CASE;
```

```

// Обнуление признаков вызова подАвтоматов
ATWmain := FALSE;

// Вызов вложенных автоматов в Состоянии
CASE YATmain OF
  1: ATWmain := TRUE;
  ELSE: ;
END_CASE;

END_FUNCTION_BLOCK

//*****

DATA_BLOCK DB_ATmain FB_ATmain //Экземплярный блок данных для FB_ATmain
BEGIN

END_DATA_BLOCK

```

II 2.2. Технологический автомат управления промывкой *ATWmain* (рис. 19, 20)

```

//*****
//
// СОЗДАНО с использованием КОНВЕРТОРА
// графа переходов(нарисованного в Visio) в текст программы
//
//*****

FUNCTION_BLOCK FB_ATWmain //Технологический автомат управления промывкой

VAR_INPUT
  MNReadyACWP:BOOL; //Нет готовности к пуску промывного насоса в автоматическом режиме
  ReadyAC_curr:BOOL; //Есть все условия для работы текущего КО в автоматическом режиме
  ReadyAC_next:BOOL; //Есть все условия для работы следующего КО в автоматическом режиме
  LCC_next:BOOL; //Следующий осветлитель последний в очереди
  LCC_curr:BOOL; //Текущий осветлитель последний в очереди
  TW_curr:BOOL; //Окончилось время контроля промывки текущего КО
  Tprep_curr:BOOL; //Окончилось время контроля для подготовки к промывке следующего КО
  X3S_WW_curr:BOOL; //Задвижка промывной воды текущего КО открыта
  X3S_WW_next:BOOL; //Задвижка промывной воды следующего КО открыта
  X1P_WP:BOOL; //Насосный агрегат включен
  X3S_FW:BOOL; //Напорная задвижка насосного агрегата открыта
  ATWoff_curr:BOOL; //Промывка текущего КО завершена
  ATWoffErrP_curr:BOOL; //Подготовка текущего КО к промывке не выполнена
  ATWoffErr_curr:BOOL; //Промывка текущего КО не выполнена
  ATWoffErrP_next:BOOL; //Подготовка следующего КО к промывке не выполнена
  ATWoffErr_next:BOOL; //Промывка следующего КО не выполнена

  M1P_WP:BOOL; //Нет включения пускателя насосного агрегата за контрольное время
  MopS_FW:BOOL; //Ненормальное открытие напорной задвижки насосного агрегата

  NStopWP:BOOL; //Признак «Не останавливать НА после завершения промывки»
  ErrCls_WW_mem:BOOL; //Не получилось закрыть задвижку промывной воды на КО
  //на котором произошло заклинивание задвижки промывной воды

  PredpuskWP:BOOL; //Предупреждение пуска промывного насоса
  SACWP23:BOOL; //Один из НА выбран для управления
END_VAR

VAR_OUTPUT
  ATWcurr:BOOL; //Вызываемый автомат управления промывкой текущего КО в очереди
  ATWNext:BOOL; //Вызываемый автомат управления промывкой следующего КО в очереди

  zKonPWCC_curr:BOOL; //Запись команды "Подготовить к промывке текущий КО"
  zKonPWCC_next:BOOL; //Запись команды "Подготовить к промывке следующий КО"
  zKonWP:BOOL; //Запись команды "Включить насосный агрегат"
  zNKonWP:BOOL; //Стирание команды "Включить насосный агрегат"
  zKoffWP:BOOL; //Запись команды "Отключить насосный агрегат"
  zNKoffWP:BOOL; //Стирание команды "Отключить насосный агрегат"

```

```

zChInd_curr:BOOL; //Запись команды "Изменить индекс текущего КО на индекс следующего"
zUpInd_curr:BOOL; //Запись команды "Увеличить индекс текущего КО на 1"
zUpInd_next:BOOL; //Запись команды "Увеличить индекс следующего КО на 1"

zATWmainoff:BOOL; //Запись признака "Промывка БКО-1 завершена нормально"
zATWmainoffErr:BOOL; //Запись признака "Промывка БКО-1 завершена ненормально"

ATWmainoff:BOOL; //Признак "Промывка БКО-1 завершена нормально"
ATWmainoffErr:BOOL; //Признак "Промывка БКО-1 завершена ненормально"

zKOHAX_curr:BOOL; //Запись признака «Текущий КО выведен из очереди на промывку»
zKOHAX_next:BOOL; //Запись признака «Следующий КО выведен из очереди на промывку»

END_VAR

VAR_IN_OUT
    YATWmain:INT; //Номер состояния технологического автомата управления
END_VAR

//
// Обнуление (сброс) команд, признаков и т.п., записываемых в автомате
//
zATWmainoff := FALSE;
zATWmainoffErr := FALSE;
zChInd_curr := FALSE;
zKOHAX_curr := FALSE;
zKOHAX_next := FALSE;
zKoffWP := FALSE;
zKonPWCC_curr := FALSE;
zKonPWCC_next := FALSE;
zKonWP := FALSE;
//zMemInd_curr := FALSE;
//zMemInd_next := FALSE;
zNKoffWP := FALSE;
zNKonWP := FALSE;
zUpInd_curr := FALSE;
zUpInd_next := FALSE;

// Реализация графа переходов

CASE YATWmain OF
0: IF ( MNReadyACWP OR (NOT ReadyAC_curr AND LCC_curr) OR NOT CACWP23 ) THEN YATWmain:=88;
    ELSE IF ( ReadyAC_curr AND NOT MNReadyACWP) THEN zKonPWCC_curr := TRUE; YATWmain := 1;
        ELSE IF ( NOT ReadyAC_curr AND NOT LCC_next) THEN
            zKOHAX_curr := TRUE; zUpInd_curr := TRUE; zUpInd_next := TRUE;
            ELSE IF ( NOT ReadyAC_curr AND LCC_next) THEN
                zKOHAX_curr := TRUE; zUpInd_curr := TRUE;
            END_IF; END_IF; END_IF; END_IF;
1: IF (NOT CACWP23 OR MNReadyACWP OR ATWoffErr_curr OR (ATWoffErrP_curr AND LCC_curr)) THEN
    YATWmain:=88;
    ELSE IF ( X3S_WW_curr AND CACWP23 ) THEN zKonWP := TRUE; YATWmain := 2;
        ELSE IF ( ATWoffErrP_curr AND NOT LCC_next) THEN
            zKOHAX_curr := TRUE; zUpInd_curr := TRUE; zUpInd_next := TRUE; YATWmain := 0;
            ELSE IF ( ATWoffErrP_curr AND LCC_next) THEN
                zKOHAX_curr := TRUE; zUpInd_curr := TRUE; YATWmain := 0;
            END_IF; END_IF; END_IF; END_IF;
2: IF ( MNReadyACWP OR ATWoffErr_curr OR NOT CACWP23) THEN YATWmain := 88; zNKonWP := TRUE;
    ELSE IF ( PredpuskWP) THEN zNKonWP := TRUE; YATWmain := 3; END_IF; END_IF;
3: IF ( M1P_WP OR ATWoffErr_curr) THEN YATWmain := 88;
    ELSE IF ( NOT CACWP23 AND LCC_curr) THEN YATWmain := 7;
        ELSE IF ( NOT CACWP23 AND NOT LCC_curr) THEN YATWmain := 5;
            ELSE IF ( X1P_WP) THEN YATWmain := 4;
        END_IF; END_IF; END_IF; END_IF;
4: IF ( (NOT X1P_WP OR Mops_FW) AND CACWP23) THEN YATWmain := 88;
    ELSE IF ( ATWoffErr_curr) THEN zKoffWP := TRUE; YATWmain := 10;
        ELSE IF ( (X3S_FW OR NOT CACWP23) AND NOT LCC_curr ) THEN YATWmain := 5;
            ELSE IF ( (X3S_FW OR NOT CACWP23) AND LCC_curr) THEN YATWmain := 7;
        END_IF; END_IF; END_IF; END_IF;
5: IF ( NOT X1P_WP AND CACWP23) THEN YATWmain := 88;
    ELSE IF ( ATWoffErr_curr) THEN zKoffWP := TRUE; YATWmain := 10;
        ELSE IF ( Tprep_curr AND ReadyAC_next ) THEN YATWmain := 6; zKonPWCC_next := TRUE;
            ELSE IF ( NOT LCC_next AND Tprep_curr AND NOT ReadyAC_next) THEN
                zKOHAX_next := TRUE; zUpInd_next := TRUE;
            END_IF;
        END_IF;
    END_IF;
END_CASE

```

```

ELSE IF ( (LCC_next AND Tprep_curr AND NOT ReadyAC_next)) THEN
    zKOHAX_next := TRUE; YATWmain := 7;
    ELSE IF ( LCC_curr) THEN YATWmain := 7;
END_IF; END_IF; END_IF; END_IF; END_IF; END_IF;
6: IF ( NOT X1P_WP AND CACWP23) THEN YATWmain := 88;
ELSE IF ( ATWoffErr_next OR ATWoffErr_curr) THEN zKoffWP := TRUE; YATWmain := 10;
    ELSE IF ( ATWoff_curr AND LCC_next AND X3S_WW_next) THEN
        zChInd_curr := TRUE; YATWmain := 7;
    ELSE IF ( ATWoff_curr AND NOT LCC_next AND X3S_WW_next) THEN
        zChInd_curr := TRUE; zUpInd_next := TRUE; YATWmain := 5;
    ELSE IF ( ATWoffErrP_next AND NOT LCC_next) THEN
        zKOHAX_next := TRUE; zUpInd_next := TRUE; zKonPWCC_next := TRUE;
    ELSE IF ( ATWoffErrP_next AND LCC_next) THEN
        zKOHAX_next := TRUE; YATWmain := 7;
    END_IF; END_IF; END_IF; END_IF; END_IF; END_IF;
7: IF ( NOT X1P_WP AND CACWP23) THEN YATWmain := 88;
ELSE IF ( ATWoffErr_curr) THEN zKoffWP := TRUE; YATWmain := 10;
    ELSE IF ( TW_curr AND NOT NStopWP) THEN zKoffWP := TRUE; YATWmain := 8;
    ELSE IF ( TW_curr AND (NStopWP OR NOT CACWP23)) THEN
        zATWmainoff := TRUE; YATWmain := 99;
    END_IF; END_IF; END_IF; END_IF;
8: IF ( ATWoffErr_curr) THEN YATWmain := 88;
    ELSE IF ( NOT X1P_WP) THEN zNKoffWP := TRUE; YATWmain := 9; END_IF; END_IF;
9: IF ( ATWoffErr_curr) THEN YATWmain := 88;
    ELSE IF ( ATWoff_curr) THEN zATWmainoff := TRUE; YATWmain := 99; END_IF; END_IF;
10: IF ( NOT X1P_WP) THEN zNKoffWP := TRUE; YATWmain := 88; END_IF;
88: IF ( 1) THEN zATWmainoffErr := TRUE; END_IF;
ELSE: ;
END_CASE;

```

```

// Обнуление признаков вызова вложенных автоматов
ATWCurr := FALSE;
ATWNext := FALSE;

```

```

// Вызов подАвтоматов в Состоянии

```

```

CASE YATWmain OF
    1: ATWCurr := TRUE;
    3: ATWCurr := TRUE;
    5: ATWCurr := TRUE;
    6: ATWCurr := TRUE; ATWNext := TRUE;
    7: ATWCurr := TRUE;
    8: ATWCurr := TRUE;
    9: ATWCurr := TRUE;
    4: ATWCurr := TRUE;
    2: ATWCurr := TRUE;
ELSE: ;
END_CASE;

```

```

END_FUNCTION_BLOCK

```

```

//*****

```

```

DATA_BLOCK DB_ATWmain FB_ATWmain //Экземплярный блок данных для FB_ATWmain
BEGIN

```

```

END_DATA_BLOCK

```

П 2.3. Автомат управления промывкой осветлителя в очереди ATWx (рис. 21, 22)

```
//*****  
//  
// СОЗДАНО с использованием КОНВЕРТОРА  
// графа переходов (нарисованного в Visio) в текст программы  
//  
//*****  
  
FUNCTION_BLOCK FB_ATWx //Автомат управления промывкой осветлителя в очереди  
VAR_INPUT  
  
    FCC:BOOL;           //Текущий КО первый в очереди  
    LCC:BOOL;           //Текущий КО последний в очереди  
    NConnectCC:BOOL;    //Отсутствует связь с контроллером КО  
    CACWP23:BOOL;       //Один из НА выбран для управления  
    ReadyAC:BOOL;       //Есть все условия для работы КО в автоматическом режиме  
  
    WKopS_WW_next:BOOL; //Команда «Открыть промывную задвижку» (для следующего КО)  
  
    X4S_DW:BOOL; //Задвижка сырой воды закрыта  
    X4S_CW:BOOL; //Задвижка чистой воды закрыта  
    X3S_WW:BOOL; //Задвижка промывной воды открыта  
    X4S_WW:BOOL; //Задвижка промывной воды закрыта  
    X3S_SW:BOOL; //Задвижка канализационной воды открыта  
    X1P_WP:BOOL; //Насосный агрегат включен  
    X3S_FW:BOOL; //Напорная задвижка насосного агрегата открыта  
  
    X4S_DW_next:BOOL; //Задвижка сырой воды следующего КО закрыта  
    X4S_CW_next:BOOL; //Задвижка чистой воды следующего КО закрыта  
    X3S_SW_next:BOOL; //Задвижка канализационной воды следующего КО открыта  
  
    Tprep:BOOL; //Окончилось время контроля для выдачи команды на подготовку следующего КО  
    TW:BOOL;     //Окончилось время контроля промывки  
  
    MclS_DW:BOOL; //Ненормальное закрытие задвижки сырой воды  
    MclS_CW:BOOL; //Ненормальное закрытие задвижки чистой воды  
    MopS_SW:BOOL; //Ненормальное открытие задвижки канализационной воды  
    MopS_WW:BOOL; //Ненормальное открытие задвижки промывной воды  
    MclS_WW:BOOL; //Ненормальное закрытие задвижки промывной воды  
  
    MNCclS_DW:BOOL; //Нет условий для автоматического закрытия задвижки сырой воды  
    MNCclS_CW:BOOL; //Нет условий для автоматического закрытия задвижки чистой воды  
    MNCopS_SW:BOOL; //Нет условий для автоматического открытия задвижки канализационной воды  
    MNCopS_WW:BOOL; //Нет условий для автоматического открытия задвижки промывной воды  
    MNCclS_WW:BOOL; //Нет условий для автоматического закрытия задвижки промывной воды  
  
    ATWoffErrP_next:BOOL; //Не выполнена подготовка следующего КО к промывке  
    Next_to_Curr:BOOL;     //Следующий КО стал текущим  
  
END_VAR  
VAR_OUTPUT  
    zNKonPWCC:BOOL; //Стирание команды "Подготовить к промывке текущий КО"  
    zWKops_WW:BOOL; //Запись команды "Открыть промывную задвижку"  
    zNWKops_WW:BOOL; //Стирание команды "Открыть промывную задвижку"  
    zKoffWCC:BOOL; //Запись команды "Завершить промывку текущего КО"  
    zNKoffWCC:BOOL; //Стирание команды "Завершить промывку текущего КО"  
    zstprep:BOOL; //Начать контроль времени для выдачи команды  
                    //на подготовку к промывке следующего КО  
    zstw:BOOL; //Начать контроль времени промывки  
    zMNSstate:BOOL; //Запись признака «Положение задвижек не соответствует режиму промывки»  
  
    zATWoffErrP:BOOL; //Запись признака "Подготовка текущего КО к промывке не выполнена"  
    zATWoffErr:BOOL; //Запись признака "Промывка текущего КО не выполнена"  
    zATWoff:BOOL; //Запись признака "Промывка текущего КО завершена"  
  
END_VAR  
  
VAR_IN_OUT  
    YATWx:INT; //Номер состояния автомата управления промывкой осветлителя в очереди  
END_VAR
```



```

//
// Обнуление (сброс) команд, признаков и т.п., записываемых в автомате
//
zATWoff := FALSE;
zATWoffErr := FALSE;
zATWoffErrP := FALSE;
zKoffWCC := FALSE;
zMNSstate := FALSE;
zNKoffWCC := FALSE;
zNKonPWCC := FALSE;
//zMNSstate := FALSE;
zNWKopS_WW := FALSE;
zWKopS_WW := FALSE;
zstprep := FALSE;
zstw := FALSE;

// Реализация графа переходов

CASE YATWx OF
  0: IF ( NOT ReadyAC) THEN zNKonPWCC := TRUE; YATWx := 66;
     ELSE IF ( X4S_DW AND X4S_CW AND X3S_SW AND (FCC OR NOT X1P_WP OR TW)) THEN
         zWKopS_WW := TRUE; zNKonPWCC := TRUE; YATWx := 1;
         ELSE IF ( MNCcls_DW OR Mcls_DW OR MNCcls_CW OR Mcls_CW OR MNCopS_SW OR MopS_SW ) THEN
             zNKonPWCC := TRUE; YATWx := 66; END_IF;
         END_IF; END_IF;
  1: IF ( MNCopS_WW OR (MopS_WW AND X4S_WW)) THEN zNWKopS_WW := TRUE; YATWx := 66;
     ELSE IF ( (MopS_WW AND NOT X4S_WW) OR NOT ReadyAC) THEN zNWKopS_WW := TRUE; YATWx := 88;
         ELSE IF ( Next_to_Curr AND X3S_WW AND (X1P_WP AND X3S_FW OR NOT CACWP23)) THEN
             zNWKopS_WW := TRUE; zstprep := TRUE; zstw := TRUE; YATWx := 2;
         END_IF; END_IF; END_IF;
  2: IF ( NConnectCC OR NOT X4S_DW OR NOT X4S_CW OR NOT X3S_SW OR NOT X3S_WW) THEN
         zMNSstate := TRUE; YATWx := 88;
         ELSE IF ( Tprep AND NOT LCC ) THEN YATWx := 3;
             ELSE IF ( LCC AND TW) THEN YATWx := 5;
         END_IF; END_IF; END_IF;
  3: IF ( NConnectCC OR NOT X4S_DW OR NOT X4S_CW OR NOT X3S_SW OR NOT X3S_WW) THEN
         zMNSstate := TRUE; YATWx := 88;
         ELSE IF ( ATWoffErrP_next) THEN YATWx := 2; //zstprep := TRUE;
             ELSE IF ( X4S_DW_next AND X4S_CW_next AND X3S_SW_next) THEN YATWx := 4;
         END_IF; END_IF; END_IF;
  4: IF ( NConnectCC OR NOT X4S_DW OR NOT X4S_CW OR NOT X3S_SW OR NOT X3S_WW) THEN
         zMNSstate := TRUE; YATWx := 88;
         ELSE IF ( TW AND WKopS_WW_next) THEN zKoffWCC := TRUE; YATWx := 6;
         END_IF; END_IF;
  5: IF ( NConnectCC OR NOT X4S_DW OR NOT X4S_CW OR NOT X3S_SW OR NOT X3S_WW) THEN
         zMNSstate := TRUE; YATWx := 88;
         ELSE IF ( NOT X1P_WP) THEN zKoffWCC := TRUE; YATWx := 6; END_IF; END_IF;
  6: IF ( NOT ReadyAC OR Mcls_WW) THEN YATWx := 88; zNKoffWCC := TRUE;
         ELSE IF ( X4S_WW) THEN zNKoffWCC := TRUE; YATWx := 99; END_IF; END_IF;
  66: IF ( 1) THEN zATWoffErrP := TRUE; END_IF;
  88: IF ( 1) THEN zATWoffErr := TRUE; END_IF;
  99: IF ( 1) THEN zATWoff := TRUE; END_IF;
     ELSE: ;
END_CASE;

END_FUNCTION_BLOCK

//*****

DATA_BLOCK DB_ATWx FB_ATWx //Экземплярный блок данных для FB_ATWx
BEGIN

END_DATA_BLOCK

```

Приложение 3. Примеры подпрограмм вызова автоматов

П 3.1. Вызов автомата *ATmain*

```
//*****
// ДАННЫЙ ПРИМЕР ПОДПРОГРАММЫ ВЫЗОВА АВТОМАТА
// включает в себя ОДИН ИЗ СПОСОБОВ ПРОТОКОЛИРОВАНИЯ
//*****

FUNCTION CALL_FB_ATmain:VOID //Вызов главного технологического автомата управления

// Переменные, введенные для обеспечения ПРОТОКОЛИРОВАНИЯ
VAR
    Frame:structDINT;
    Buffer AT Frame:DWORD;
END_VAR

VAR_TEMP
    i:INT;
    YATmain_prev:INT;
    tempRET_VAL:INT;
    tempCDT:DT;
END_VAR

// Для обеспечения ПРОТОКОЛИРОВАНИЯ
YATmain_prev:=DB_ATmain.YATmain;

FB_ATmain.DB_ATmain(
    ATWmainoff := DB_ATWmain.ATWmainoff,
    ATWmainoffErr := DB_ATWmain.ATWmainoffErr,
    ReadyBCC := DB_PW.ReadyBCC,
    CACWP23 := FC_CACWP23(),
    HLKstart := DB_PW.HLKstart,
    HLKstop := DB_PW.HLKstop,
    X1P_WP := DB_CC.ArrWP[DB_PW.NumberWP].X1P_WP,
    PredpuskWP := DB_CC.ArrWP[DB_PW.NumberWP].PredpuskWP,
    NStopWP := DB_PW.NStopWP
);

// ПРОТОКОЛИРОВАНИЕ
Frame.bit00:=DB_ATmain.zNKoffWP; //выходы автомата
Frame.bit01:=DB_ATmain.zFinit; //выходы автомата
Frame.bit02:=DB_ATmain.zKoffWP; //выходы автомата
Frame.bit03:=DB_ATmain.zKoffWCC; //выходы автомата
Frame.bit04:=DB_ATmain.zHLKclr; //выходы автомата
Frame.bit05:=DB_ATmain.ATWmain; //выходы автомата
Frame.bit06:=DB_ATmain.zClrEndWashing; //выходы автомата
Frame.bit07:=False;
. . .
Frame.bit31:=False;

IF YATmain_prev <> DB_ATmain.YATmain OR ATPROT.protATmain[ATPROT.INDEXprotATmain].OUTPUT <> Buffer
THEN
    IF ATPROT.INDEXprotATmain >= 20 THEN ATPROT.INDEXprotATmain := 0;
    ELSE ATPROT.INDEXprotATmain := ATPROT.INDEXprotATmain + 1; END_IF;

    ATPROT.protATmain[ATPROT.INDEXprotATmain].OUTPUT:=Buffer;
    tempRET_VAL := READ_CLK(RET_VAL := tempRET_VAL, CDT := tempCDT); //Текущее время
    ATPROT.protATmain[ATPROT.INDEXprotATmain].DTIME := DT_TOD(tempCDT);
    ATPROT.protATmain[ATPROT.INDEXprotATmain].Yprev:=YATmain_prev; //предыдущий номер состояния
    ATPROT.protATmain[ATPROT.INDEXprotATmain].Ycurr:=DB_ATmain.YATmain; //текущий номер состояния

    Frame.bit00:=DB_ATmain.ATWmainoff; //входы автомата
    Frame.bit01:=DB_ATmain.ATWmainoffErr; //входы автомата
    Frame.bit02:=False; //входы автомата
    Frame.bit03:=DB_ATmain.ReadyBCC; //входы автомата
    Frame.bit04:=DB_ATmain.CACWP23; //входы автомата
    Frame.bit05:=DB_ATmain.HLKstart; //входы автомата
    Frame.bit06:=DB_ATmain.HLKstop; //входы автомата
    Frame.bit07:=DB_ATmain.X1P_WP; //входы автомата
    Frame.bit08:=DB_ATmain.PredpuskWP; //входы автомата
    Frame.bit09:=DB_ATmain.NStopWP; //входы автомата
    Frame.bit10:=False;
    . . .
    Frame.bit31:=False;

    ATPROT.protATmain[ATPROT.INDEXprotATmain].INPUT:=Buffer;
END_IF;
```

```

//Обработка результатов работы автомата
IF DB_ATmain.zKoffWP THEN DB_CC.ArrWP[DB_PW.NumberWP].KoffWP := False; END_IF;
IF DB_ATmain.zFinit THEN
  DB_ATWmain.YATWmain := 0;
  DB_ATWmain.ATWmainoff := False;
  DB_ATWmain.ATWmainoffErr := False;
  DB_PW.IndexCurr := 1;
  DB_PW.IndexNext := 2;
  DB_PW.IndexMem := 0;
  DB_PW.Next_to_Curr := False; //Следующий КО стал текущим
  FOR i:=1 TO 24 BY 1 DO
    DB_CC.ArrCC[i].YATWx := 0;
    DB_CC.ArrCC[i].ATWoff := False;
    DB_CC.ArrCC[i].ATWoffErr := False;
    DB_CC.ArrCC[i].ATWoffErrP := False;
    DB_CC.ArrCC[i].KonPWCC := False;
    DB_CC.ArrCC[i].WKops_WW := False;
    DB_CC.ArrCC[i].KoffWCC := False;
  END_FOR;
  FOR i:=2 TO 3 BY 1 DO
    DB_CC.ArrWP[i].KonWP := False;
    DB_CC.ArrWP[i].KoffWP := False;
  END_FOR;
END_IF;

IF DB_ATmain.zClrEndWashing THEN
  DB_PW.EndWashing := 0;
  FOR i:=1 TO 24 BY 1 DO
    DB_PW.ArrCC_MNSstate[i] := False;
    DB_PW.ArrCC_KOHAX[i] := False;
  END_FOR;
END_IF;

IF DB_ATmain.zKoffWP THEN DB_CC.ArrWP[DB_PW.NumberWP].KoffWP := True; END_IF;
IF DB_ATmain.zKoffWCC THEN
  FOR i:=1 TO 24 BY 1 DO
    DB_CC.ArrCC[i].KoffWCC := True;
  END_FOR;
END_IF;

IF DB_ATmain.zHLKclr THEN DB_PW.HLKstart := False; DB_PW.HLKstop := False; END_IF;
IF DB_ATmain.ATWmain THEN CALL_FB_ATWmain();END_IF;

END_FUNCTION

```

П 3.2. Вызов автомата *ATWmain*

```

//*****
// ДАННЫЙ ПРИМЕР ПОДПРОГРАММЫ ВЫЗОВА АВТОМАТА
// включает в себя ОДИН ИЗ СПОСОБОВ ПРОТОКОЛИРОВАНИЯ
//*****

FUNCTION CALL_FB_ATWmain:VOID //Вызов технологического автомата управления промывкой

  // Переменные, введенные для обеспечения ПРОТОКОЛИРОВАНИЯ
  VAR
    Frame:structDINT;
    Buffer AT Frame:DWORD;
  END_VAR

  VAR_TEMP
    YATWmain_prev:INT;
    tempRET_VAL:INT;
    tempCDT:DT;
  END_VAR

  // Для обеспечения ПРОТОКОЛИРОВАНИЯ
  YATWmain_prev:=DB_ATWmain.YATWmain;

```

```

FB_ATWmain.DB_ATWmain(
  MNReadyACWP := DB_CC.ArrWP[DB_PW.NumberWP].MNreadyACWP,
  ReadyAC_curr := FC_ReadyAC(DB_PW.ArrHL[DB_PW.IndexCurr]),
  ReadyAC_next := FC_ReadyAC(DB_PW.ArrHL[DB_PW.IndexNext]),
  LCC_next := FC_LCC(DB_PW.IndexNext),
  LCC_curr := FC_LCC(DB_PW.IndexCurr),
  TW_curr := DB_ATWx.TW,
  Tprep_curr := DB_ATWx.Tprep,
  X3S_WW_curr := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].X3S_WW,
  X3S_WW_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X3S_WW,
  X1P_WP := DB_CC.ArrWP[DB_PW.NumberWP].X1P_WP,
  X3S_FW := DB_CC.ArrWP[DB_PW.NumberWP].X3S_FW,
  ATWoff_curr := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].ATWoff,
  ATWoffErrP_curr := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].ATWoffErrP,
  ATWoffErrP_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].ATWoffErrP,
  ATWoffErrP_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].ATWoffErr,
  ATWoffErr_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].ATWoffErr,
  M1P_WP := DB_CC.ArrWP[DB_PW.NumberWP].M1P_WP,
  MopS_FW := DB_CC.ArrWP[DB_PW.NumberWP].MopS_FW,
  NStopWP := DB_PW.NStopWP,
  ErrCls_WW_mem := FC_ErrCls_WW_mem(),
  CACWP23 := FC_CACWP23(),
  PredpuskWP := DB_CC.ArrWP[DB_PW.NumberWP].PredpuskWP
);

// ПРОТОКОЛИРОВАНИЕ
Frame.bit00:=DB_ATWmain.zKonWP; //выходы автомата
Frame.bit01:=DB_ATWmain.zNKonWP; //выходы автомата
Frame.bit02:=DB_ATWmain.zKoffWP; //выходы автомата
Frame.bit03:=DB_ATWmain.zNKoffWP; //выходы автомата
Frame.bit04:=false;//DB_ATWmain.zMemInd_curr; //выходы автомата
Frame.bit05:=false;//DB_ATWmain.zMemInd_next; //выходы автомата
Frame.bit06:=DB_ATWmain.zChInd_curr; //выходы автомата
Frame.bit07:=DB_ATWmain.zUpInd_curr; //выходы автомата
Frame.bit08:=DB_ATWmain.zUpInd_next; //выходы автомата
Frame.bit09:=DB_ATWmain.zKonPWCC_curr; //выходы автомата
Frame.bit10:=DB_ATWmain.zKonPWCC_next; //выходы автомата
Frame.bit11:=false;//DB_ATWmain.zPermOpS_WW_next; //выходы автомата
Frame.bit12:=false;//DB_ATWmain.zNPermOpS_WW_next; //выходы автомата
Frame.bit13:=DB_ATWmain.zATWmainoff; //выходы автомата
Frame.bit14:=DB_ATWmain.zATWmainoffErr; //выходы автомата
Frame.bit15:=DB_ATWmain.ATWcurr; //выходы автомата
Frame.bit16:=DB_ATWmain.ATWnext; //выходы автомата
Frame.bit17:=DB_ATWmain.zKOHAX_curr;
Frame.bit18:=DB_ATWmain.zKOHAX_next;
Frame.bit19:=False;
.
.
Frame.bit31:=False;

IF YATWmain_prev <> DB_ATWmain.YATWmain OR
ATPROT.protATWmain[ATPROT.INDEXprotATWmain].OUTPUT <> Buffer THEN

ATPROT.INDEXprotATWmain := ATPROT.INDEXprotATWmain + 1;
ATPROT.protATWmain[ATPROT.INDEXprotATWmain].OUTPUT:=Buffer;
tempRET_VAL := READ_CLK(RET_VAL := tempRET_VAL, CDT := tempCDT);//Текущее время
ATPROT.protATWmain[ATPROT.INDEXprotATWmain].DTIME := DT_TOD(tempCDT);
ATPROT.protATWmain[ATPROT.INDEXprotATWmain].Yprev:=YATWmain_prev; //предыдущий номер состояния
ATPROT.protATWmain[ATPROT.INDEXprotATWmain].Ycurr:=DB_ATWmain.YATWmain; //текущий номер состояния

Frame.bit00:=DB_ATWmain.MNReadyACWP; //входы автомата
Frame.bit01:=DB_ATWmain.ReadyAC_curr; //входы автомата
Frame.bit02:=DB_ATWmain.ReadyAC_next; //входы автомата
Frame.bit03:=DB_ATWmain.LCC_next; //входы автомата
Frame.bit04:=DB_ATWmain.LCC_curr; //входы автомата
Frame.bit05:=DB_ATWmain.TW_curr; //входы автомата
Frame.bit06:=DB_ATWmain.Tprep_curr; //входы автомата
Frame.bit07:=DB_ATWmain.X3S_WW_curr; //входы автомата
Frame.bit08:=DB_ATWmain.X3S_WW_next; //входы автомата
Frame.bit09:=DB_ATWmain.X1P_WP; //входы автомата
Frame.bit10:=DB_ATWmain.X3S_FW; //входы автомата
Frame.bit11:=DB_ATWmain.ATWoff_curr; //входы автомата
Frame.bit12:=DB_ATWmain.ATWoffErrP_curr; //входы автомата
Frame.bit13:=DB_ATWmain.ATWoffErrP_next; //входы автомата
Frame.bit14:=DB_ATWmain.ATWoffErrP_next; //входы автомата
Frame.bit15:=DB_ATWmain.ATWoffErr_next; //входы автомата
Frame.bit16:=DB_ATWmain.M1P_WP; //входы автомата
Frame.bit17:=DB_ATWmain.MopS_FW; //входы автомата
Frame.bit18:=DB_ATWmain.NStopWP; //входы автомата
Frame.bit19:=DB_ATWmain.ErrCls_WW_mem; //входы автомата

```

```

Frame.bit20:=DB_ATWmain.CACWP23; //входы автомата
Frame.bit21:=DB_ATWmain.PredpushkWP; //входы автомата
Frame.bit22:=False;
. . .
Frame.bit31:=False;

ATPROT.protATWmain[ATPROT.INDEXprotATWmain].INPUT := Buffer;

END_IF;

//Обработка результатов работы автомата
IF DB_ATWmain.zKonWP THEN DB_CC.ArrWP[DB_PW.NumberWP].KonWP :=True; END_IF;
IF DB_ATWmain.zNKonWP THEN DB_CC.ArrWP[DB_PW.NumberWP].KonWP :=False; END_IF;
IF DB_ATWmain.zKoffWP THEN DB_CC.ArrWP[DB_PW.NumberWP].KoffWP :=True; END_IF;
IF DB_ATWmain.zNKoffWP THEN DB_CC.ArrWP[DB_PW.NumberWP].KoffWP :=False; END_IF;
IF DB_ATWmain.zKOHAX_curr THEN
  DB_PW.ArrCC_KOHAX[DB_PW.ArrHL[DB_PW.IndexCurr]] := True; END_IF;
IF DB_ATWmain.zKOHAX_next THEN
  DB_PW.ArrCC_KOHAX[DB_PW.ArrHL[DB_PW.IndexNext]] := True; END_IF;
IF DB_ATWmain.zChInd_curr THEN DB_PW.IndexCurr := DB_PW.IndexNext; END_IF;
IF DB_ATWmain.zUpInd_curr THEN DB_PW.IndexCurr := DB_PW.IndexCurr + 1; END_IF;
IF DB_ATWmain.zUpInd_next THEN DB_PW.IndexNext := DB_PW.IndexNext + 1; END_IF;
IF DB_ATWmain.zKonPWCC_curr THEN
  DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].KonPWCC := True; END_IF;
IF DB_ATWmain.zKonPWCC_next THEN
  DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].KonPWCC := True; END_IF;
DB_ATWmain.ATWmainoff := DB_ATWmain.zATWmainoff;
IF DB_ATWmain.ATWmainoff THEN DB_PW.EndWashing := 1; END_IF;
DB_ATWmain.ATWmainoffErr := DB_ATWmain.zATWmainoffErr;
IF DB_ATWmain.ATWmainoffErr THEN DB_PW.EndWashing := 2; END_IF;
IF DB_ATWmain.ATWCurr THEN CALL_FB_ATWx_Curr(); END_IF;
IF DB_ATWmain.ATWnext THEN CALL_FB_ATWx_Next(); END_IF;

```

END_FUNCTION

П 3.3. Вызов автомата ATWx_curr

```

//*****
// ДАННЫЙ ПРИМЕР ПОДПРОГРАММЫ ВЫЗОВА АВТОМАТА
// включает в себя ОДИН ИЗ СПОСОБОВ ПРОТОКОЛИРОВАНИЯ
//*****

```

FUNCTION CALL_FB_ATWx_Curr:VOID //Вызов автомата управления промывкой текущего осветлителя

```

// Переменные, введенные для обеспечения ПРОТОКОЛИРОВАНИЯ
VAR
  Frame:structDINT;
  Buffer AT Frame:DWORD;
END_VAR

VAR_TEMP
  tempRET_VAL:INT;
  tempCDT:DT;
END_VAR

db500.temp_tprep := S_ODT(T_NO := 1,
  S := NOT DB_ATWx.zstprep,
  TV := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].STG_Tprep,
  BI:=db500.bival_tprep,
  Q := DB_ATWx.Tprep);

db500.temp_tw := S_ODT(T_NO := 2,
  S := NOT DB_ATWx.zstw,
  TV := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].STG_TW,
  BI:=db500.bival_tw,
  Q := DB_ATWx.TW);

```

```

FB_ATWx.DB_ATWx(
  YATWx:=DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].YATWx,
  FCC := FC_FCC(DB_PW.IndexCurr),
  LCC := FC_LCC(DB_PW.IndexCurr),
  NConnectCC := DB_NConnect.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]],
  CACWP23 := FC_CACWP23(),
  ReadyAC := FC_ReadyAC(DB_PW.ArrHL[DB_PW.IndexCurr]),
  WKops_WW_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].WKops_WW,
  X4S_DW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].X4S_DW,
  X4S_CW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].X4S_CW,
  X3S_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].X3S_WW,
  X4S_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].X4S_WW,
  X3S_SW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].X3S_SW,
  X1P_WP := DB_CC.ArrWP[DB_PW.NumberWP].X1P_WP,
  X3S_FW := DB_CC.ArrWP[DB_PW.NumberWP].X3S_FW,
  X4S_DW_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X4S_DW,
  X4S_CW_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X4S_CW,
  X3S_SW_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X3S_SW,
  MclS_DW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MclS_DW,
  MclS_CW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MclS_CW,
  MopS_SW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MopS_SW,
  MopS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MopS_WW,
  MclS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MclS_WW,
  MNCclS_DW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MNCclS_DW,
  MNCclS_CW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MNCclS_CW,
  MNCopS_SW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MNCopS_SW,
  MNCopS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MNCopS_WW,
  MNCclS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].MNCclS_WW,
  ATWoffErrP_next := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].ATWoffErrP,
  Next_to_Curr:=TRUE // Следующий КО стал текущим
                      // ДЛЯ ТЕКУЩЕГО ЭТОТ ПРИЗНАК НЕ ДОЛЖЕН МЕШАТЬ, ПОЭТОМУ TRUE
);

// ПРОТОКОЛИРОВАНИЕ
Frame.bit00:=DB_ATWx.zNKonPWCC;//выходы автомата
Frame.bit01:=DB_ATWx.zWKops_WW; //выходы автомата
Frame.bit02:=DB_ATWx.zNWKops_WW; //выходы автомата
Frame.bit03:=DB_ATWx.zKoffWCC; //выходы автомата
Frame.bit04:=DB_ATWx.zKoffFWCC; //выходы автомата
Frame.bit05:=DB_ATWx.zstprep; //выходы автомата
Frame.bit06:=DB_ATWx.zstw; //выходы автомата
Frame.bit07:=DB_ATWx.zATWoffErrP; //выходы автомата
Frame.bit08:=DB_ATWx.zATWoffErr; //выходы автомата
Frame.bit09:=DB_ATWx.zATWoff; //выходы автомата
Frame.bit10:=DB_ATWx.zMNSstate; //выходы автомата
Frame.bit11:=False;
. . .
Frame.bit31:=False;

IF ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].
(ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]]).Ycurr <> DB_ATWx.YATWx OR
ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].OUTPUT <> Buffer THEN

ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]] :=
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]] + 1;

ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].OUTPUT:=Buffer; //выходы автомата

tempRET_VAL := READ_CLK(RET_VAL := tempRET_VAL, CDT := tempCDT);//Текущее время

ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].DTIME := DT_TOD(tempCDT);

ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].Yprev:=
ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].
(ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]]-1).Ycurr; //предыдущий номер состояния

ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]].Ycurr:=DB_ATWx.YATWx; //текущий номер
состояния

```

```

Frame.bit00:=DB_ATWx.FCC; //входы автомата
Frame.bit01:=DB_ATWx.LCC; //входы автомата
Frame.bit02:=DB_ATWx.ReadyAC; //входы автомата
Frame.bit03:=DB_ATWx.WKopS_WW_next; //входы автомата
Frame.bit04:=DB_ATWx.X4S_DW; //входы автомата
Frame.bit05:=DB_ATWx.X4S_CW; //входы автомата
Frame.bit06:=DB_ATWx.X3S_WW; //входы автомата
Frame.bit07:=DB_ATWx.X4S_WW; //входы автомата
Frame.bit08:=DB_ATWx.X3S_SW; //входы автомата
Frame.bit09:=DB_ATWx.X1P_WP; //входы автомата
Frame.bit10:=DB_ATWx.X3S_FW; //входы автомата
Frame.bit11:=DB_ATWx.X4S_DW_next; //входы автомата
Frame.bit12:=DB_ATWx.X4S_CW_next; //входы автомата
Frame.bit13:=DB_ATWx.X3S_SW_next; //входы автомата
Frame.bit14:=DB_ATWx.MclS_DW; //входы автомата
Frame.bit15:=DB_ATWx.MclS_CW; //входы автомата
Frame.bit16:=DB_ATWx.MopS_SW; //входы автомата
Frame.bit17:=DB_ATWx.MopS_WW; //входы автомата
Frame.bit18:=DB_ATWx.MclS_WW; //входы автомата
Frame.bit19:=DB_ATWx.MNCclS_DW; //входы автомата
Frame.bit20:=DB_ATWx.MNCclS_CW; //входы автомата
Frame.bit21:=DB_ATWx.MNCopS_SW; //входы автомата
Frame.bit22:=DB_ATWx.MNCopS_WW; //входы автомата
Frame.bit23:=DB_ATWx.MNCclS_WW; //входы автомата
Frame.bit24:=DB_ATWx.ATWoffErrP_next; //входы автомата
Frame.bit25:=DB_ATWx.Next_to_Curr; //Следующий КО стал текущим
Frame.bit26:=DB_ATWx.Tprep; //входы автомата
Frame.bit27:=DB_ATWx.TW; //входы автомата
Frame.bit28:=DB_ATWx.NConnectCC;
Frame.bit29:=DB_ATWx.CACWP23;
Frame.bit30:=False;
Frame.bit31:=False;

```

```

ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexCurr],
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexCurr]]].INPUT := Buffer;

```

```

END_IF;

```

```

//Обработка результатов работы автомата

```

```

DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].YATWx:=DB_ATWx.YATWx;
IF DB_ATWx.zNKonPWCC THEN DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].KonPWCC := False;
END_IF;
IF DB_ATWx.zWKops_WW THEN DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].WKopS_WW :=True;
END_IF;
IF DB_ATWx.zNWKops_WW THEN DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].WKopS_WW := False;
END_IF;
IF DB_ATWx.zKoffWCC THEN DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].KoffWCC := True;
END_IF;
IF DB_ATWx.zNKoffWCC THEN DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].KoffWCC := False;
END_IF;

```

```

IF DB_ATWx.zstprep THEN db500.temp_tprep := S_ODT(T_NO := 1, S := NOT DB_ATWx.zstprep, TV
:= DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].STG_Tprep, BI:=db500.bival_tprep, Q :=
DB_ATWx.Tprep); END_IF;
IF DB_ATWx.zstw THEN db500.temp_tw := S_ODT(T_NO := 2, S := NOT DB_ATWx.zstw, TV :=
DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].STG_TW, BI:=db500.bival_tw, Q :=
DB_ATWx.TW);END_IF;

```

```

DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].ATWoffErrP := DB_ATWx.zATWoffErrP;
DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].ATWoffErr := DB_ATWx.zATWoffErr;
DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].ATWoff := DB_ATWx.zATWoff;
IF DB_ATWx.zMNSstate THEN DB_PW.ArrCC_MNSstate[DB_PW.ArrHL[DB_PW.IndexCurr]] := True;
END_IF;

```

```

END_FUNCTION

```

П 3.4. Вызов автомата *ATWx_next*

```
//*****
// ДАННЫЙ ПРИМЕР ПОДПРОГРАММЫ ВЫЗОВА АВТОМАТА
// включает в себя ОДИН ИЗ СПОСОБОВ ПРОТОКОЛИРОВАНИЯ
//*****

FUNCTION CALL_FB_ATWx_Next:VOID //Вызов автомата управления промывкой следующего осветлителя

// Переменные, введенные для обеспечения ПРОТОКОЛИРОВАНИЯ
VAR
    Frame:structDINT;
    Buffer AT Frame:DWORD;
END_VAR

VAR_TEMP
    tempRET_VAL:INT;
    tempCDT:DT;
END_VAR

FB_ATWx.DB_ATWx(
    YATWx:=DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].YATWx,
    FCC := FC_FCC(DB_PW.IndexNext),
    LCC := FC_LCC(DB_PW.IndexNext),
    NConnectCC := DB_NConnect.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]],
    CACWP23 := FC_CACWP23(),
    ReadyAC := FC_ReadyAC(DB_PW.ArrHL[DB_PW.IndexNext]),

    WKops_WW_next := False,

    //KoffWCC_prev := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexCurr]].KoffWCC,
    X4S_DW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X4S_DW,
    X4S_CW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X4S_CW,
    X3S_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X3S_WW,
    X4S_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X4S_WW,
    X3S_SW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].X3S_SW,
    X1P_WP := DB_CC.ArrWP[DB_PW.NumberWP].X1P_WP,
    X3S_FW := DB_CC.ArrWP[DB_PW.NumberWP].X3S_FW,
    X4S_DW_next := False,
    X4S_CW_next := False,
    X3S_SW_next := False,
    MclS_DW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MclS_DW,
    MclS_CW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MclS_CW,
    MopS_SW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MopS_SW,
    MopS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MopS_WW,
    MclS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MclS_WW,
    MNCclS_DW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MNCclS_DW,
    MNCclS_CW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MNCclS_CW,
    MNCopS_SW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MNCopS_SW,
    MNCopS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MNCopS_WW,
    MNCclS_WW := DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].MNCclS_WW,
    ATWoffErrP_next := False,
    Next_to_Curr:= (DB_ATWmain.YATWmain = 5) // ЭТО ДЕЛАЕТСЯ ДЛЯ СИНХРОНИЗАЦИИ С ATWmain,
                                           // ЧТОБЫ СМЕНА ИНДЕКСОВ ПРОИЗОШЛА
                                           // ДО ЗАПУСКА ТАЙМЕРА ПРОМЫВКИ

);

// ПРОТОКОЛИРОВАНИЕ
Frame.bit00:=DB_ATWx.zNKonPWCC;//выходы автомата
Frame.bit01:=DB_ATWx.zWKops_WW; //выходы автомата
Frame.bit02:=DB_ATWx.zNWKops_WW; //выходы автомата
Frame.bit03:=DB_ATWx.zKoffWCC; //выходы автомата
Frame.bit04:=DB_ATWx.zNKoffWCC; //выходы автомата
Frame.bit05:=DB_ATWx.zstprep; //выходы автомата
Frame.bit06:=DB_ATWx.zstw; //выходы автомата
Frame.bit07:=DB_ATWx.zATWoffErrP; //выходы автомата
Frame.bit08:=DB_ATWx.zATWoffErr; //выходы автомата
Frame.bit09:=DB_ATWx.zATWoff; //выходы автомата
Frame.bit10:=DB_ATWx.zMNSstate; //выходы автомата
Frame.bit11:=False;
. . .
Frame.bit31:=False;
```



```

IF ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]].Ycurr <> DB_ATWx.YATWx OR
ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]].OUTPUT <> Buffer THEN

    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]] :=
    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]] + 1;
    ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]].OUTPUT := Buffer;

    tempRET_VAL := READ_CLK(RET_VAL := tempRET_VAL, CDT := tempCDT); //Текущее время
    ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]].DTIME := DT_TOD(tempCDT);

    ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]].Yprev :=
    ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]]-1].Ycurr; //предыдущий номер состояния

    ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]].Ycurr:=DB_ATWx.YATWx; //текущий номер

    Frame.bit00:=DB_ATWx.FCC; //входы автомата
    Frame.bit01:=DB_ATWx.LCC; //входы автомата
    Frame.bit02:=DB_ATWx.ReadyAC; //входы автомата
    Frame.bit03:=DB_ATWx.WKopS_WW_next; //входы автомата
    Frame.bit04:=DB_ATWx.X4S_DW; //входы автомата
    Frame.bit05:=DB_ATWx.X4S_CW; //входы автомата
    Frame.bit06:=DB_ATWx.X3S_WW; //входы автомата
    Frame.bit07:=DB_ATWx.X4S_WW; //входы автомата
    Frame.bit08:=DB_ATWx.X3S_SW; //входы автомата
    Frame.bit09:=DB_ATWx.X1P_WP; //входы автомата
    Frame.bit10:=DB_ATWx.X3S_FW; //входы автомата
    Frame.bit11:=DB_ATWx.X4S_DW_next; //входы автомата
    Frame.bit12:=DB_ATWx.X4S_CW_next; //входы автомата
    Frame.bit13:=DB_ATWx.X3S_SW_next; //входы автомата
    Frame.bit14:=DB_ATWx.Mc1S_DW; //входы автомата
    Frame.bit15:=DB_ATWx.Mc1S_CW; //входы автомата
    Frame.bit16:=DB_ATWx.MopS_SW; //входы автомата
    Frame.bit17:=DB_ATWx.MopS_WW; //входы автомата
    Frame.bit18:=DB_ATWx.Mc1S_WW; //входы автомата
    Frame.bit19:=DB_ATWx.MNCc1S_DW; //входы автомата
    Frame.bit20:=DB_ATWx.MNCc1S_CW; //входы автомата
    Frame.bit21:=DB_ATWx.MNCopS_SW; //входы автомата
    Frame.bit22:=DB_ATWx.MNCopS_WW; //входы автомата
    Frame.bit23:=DB_ATWx.MNCc1S_WW; //входы автомата
    Frame.bit24:=DB_ATWx.ATWoffErrP_next; //входы автомата
    Frame.bit25:=DB_ATWx.Next_to_Curr; //Следующий КО стал текущим
    Frame.bit26:=DB_ATWx.Tprep; //входы автомата
    Frame.bit27:=DB_ATWx.TW; //входы автомата
    Frame.bit28:=DB_ATWx.NConnectCC;
    Frame.bit29:=DB_ATWx.CACWP23;
    Frame.bit30:=False;
    Frame.bit31:=False;

    ATPROT.protATWx[DB_PW.ArrHL[DB_PW.IndexNext],
    ATPROT.INDEXprotATWx[DB_PW.ArrHL[DB_PW.IndexNext]].INPUT := Buffer;

END_IF;

//Обработка результатов работы автомата
DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].YATWx := DB_ATWx.YATWx;
IF DB_ATWx.zNKonPWCC THEN
    DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].KonPWCC := False; END_IF;
IF DB_ATWx.zWKops_WW THEN
    DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].WKops_WW :=True; END_IF;
IF DB_ATWx.zNWKops_WW THEN
    DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].WKops_WW := False; END_IF;
IF DB_ATWx.zKoffWCC THEN
    DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].KoffWCC := True; END_IF;
IF DB_ATWx.zNKoffWCC THEN
    DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].KoffWCC := False; END_IF;
DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].ATWoffErrP := DB_ATWx.zATWoffErrP;
DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].ATWoffErr := DB_ATWx.zATWoffErr;
DB_CC.ArrCC[DB_PW.ArrHL[DB_PW.IndexNext]].ATWoff := DB_ATWx.zATWoff;
IF DB_ATWx.zMNSstate THEN DB_PW.ArrCC_MNSstate[DB_PW.ArrHL[DB_PW.IndexNext]] := True;
END_IF;

END_FUNCTION

```

Используемые сокращения

КО	- контактный осветлитель;
БКО	- блок контактных осветлителей
НА	- насосный агрегат;
ШАУ	- шкаф автоматического управления;
ТЗ	- техническое задание
STL	- язык инструкций (синтаксис языка напоминает Ассемблер)
SCL	- Structured Control Language , язык облегчающий программирование для <i>S7-300/400</i> и соответствующий <i>DIN EN 61131-3</i> (синтаксис языка напоминает Паскаль)
MicroWin32	- среда программирования контроллеров <i>S7-200</i>
STEP 7	- среда программирования контроллеров <i>S7-300/400</i>
ЗСВ	- задвижка сырой воды
ЗЧВ	- задвижка чистой воды
ЗКВ	- задвижка слива воды канализацию
ЗПВ	- задвижка промывной воды