



Система передачи извещений. **«Интегрированная система охраны 777»** ТУ РБ 37422747.001-98

Конфигурирование скриптовой логики системы ИСБ «777»

на базе программного обеспечения ИСБ «777» Комплекс ver.5.2 Руководство пользователя. Редакция 3

> Минск октябрь 2011



Оглавление

П	еречень	применяемых сокращений	3
1	Введ	цение	4
2.	Скри	пптовая логика	5
	2.1.	Назначение. Основные понятия.	5
	2.2.	Принцип работы	5
	2.3.	Особенности команд управления вспомогательными устройствами	6
	2.4.	Структура и синтаксис скриптов	6
	2.4.1.	Скрипт реакций системы	
	2.4.1.1.	Формат секций	8
		Описание события в системе	
	2.4.1.3.	Применение в описании события шаблонов	11
		Описание простой команды	
	2.4.1.5.	Использование в командах аргументов	14
		Использование в командах условий	
		Формирование секции	
		Особенности формирования скрипта системных реакций	
		Некоторые приёмы, используемые при конфигурировании функционирования пожарной автоматики	
		Скрипт системных состояний	
		Скрипт системных таймеров	
		*	





Перечень применяемых сокращений

```
АБ-4 - Адресный Блок (общее);
АБ-4(О) - Адресный Блок (Охранный);
АБ-4(П) - Адресный Блок (Пожарный);
АБ-4(У) - Адресный Блок (Управления);
АИУ - Адаптер Интерфейсов Универсальный;
АКБ – Аккумуляторная Батарея;
АСОД - Аппаратура Связи и Обработки Данных;
АСПС – Адресная Система Пожарной Сигнализации;
АПКП – Адресный Приёмно-Контрольный Прибор;
АРМ НСБ - Автоматизирование Рабочее Место Начальника Службы Безопасности;
АРМ ТВП - Автоматизирование Рабочее Место «Терминал Выдачи пропусков»;
АРМ ТДО - Автоматизирование Рабочее Место «Терминал Дежурного Оператора»;
АРМ УРВ - Автоматизирование Рабочее Место Учёта Рабочего Времени;
АУ - Адресное Устройство;
ББП – Блок Бесперебойного Питания;
БДЭП - База Данных Электронных Пропусков;
БИ - Блок Индикации;
БК-4 - Блок Коммутационный;
БМ - Блок Микрофонный;
БУ - Блок Управления;
БУН - Блок Управления Нагрузками;
ВКП - Выносная Контрольная Панель (общее);
ВКП(3) - Выносная Контрольная Панель Замка;
ВКП(О) - Выносная Контрольная Панель Охраны;
ВКП(Д) - Выносная Контрольная Панель Двойная (охраны и доступа);
ВПИУ-16 - Выносная Панель Индикации и Управления;
ВПИУ-16(А) - Выносная Панель Индикации и Управления (Автоматики);
ВПУ – Выносная Панель Управления;
ИБП - Источник Бесперебойного Питания;
ИСО(Б) - Интегрированная Система Охраны(Безопасности);
ИУ - Исполнительное Устройство;
КСО - Контроллер Сектора Охраны;
КСО.Д - Контроллер Сектора Доступа;
КСП-2 - Коммуникатор Сопряжения с пультом «Алеся»;
МК-01 – Контроллер АПКП «Бирюза»;
МЛС – Магистральная Линия Связи;
МАШ – Модуль Адресного Шлейфа;
МШ-4 – Модуль Шлейфа (общее);
МШ-4(М) – Модуль Шлейфа Модифицированный;
МШ-4(У) – Модуль Шлейфа Управления;
ОЛС – Объектовая Линия Связи;
ОУ – Объектовое Устройство;
ПАК – Программно-Аппаратный комплекс;
ППУ – Прибор Пожарный и Управления;
ПУ-40 – Панель управления АПКП «Бирюза»;
ПЦН - Пульт Централизованного Наблюдения;
Р485 - Репитер Универсальный;
СЗУ - Светозвуковое Устройство;
```



СП - Считыватель Пропусков; СПИ - Система Передачи Извещений.



1 Введение

Данный документ содержит сведения, необходимые для конфигурирования нестандартной логики функционирования системы на базе оборудования Интегрированной Системы Безопасности «777» и пакета программного обеспечения «ИСБ«777»Комплекс» ver.5.2 для наиболее широкого использования её технических возможностей.

К работе по конфигурированию системы должны допускаться лица, имеющие необходимую квалификацию и прошедшие соответствующее обучение у производителя системы.

В связи с постоянной работой по совершенствованию системы, повышению надежности и улучшению эксплуатационных параметров, в формат скриптовой логики могут быть внесены изменения, не отраженные в настоящей редакции руководства и не ухудшающие параметры системы. Проверить наличие и загрузить последнюю редакцию данного руководства, а также пакета программного обеспечения «ИСБ«777»Комплекс» можно в разделе «поддержка» на сайте www.rovalant.by.

Сведения по инсталляции и настройке пакета программного обеспечения «ИСБ«777» Комплекс» ver.5.2, а также сведения по работе с приложением dbForge Studio for MySQL приведены в «Руководстве «Инсталляция и настройка ПО «ИСБ 777 Комплекс» ver.5.2»».

Сведения о конфигурировании стандартной логики системы приведены в «Руководстве «Конфигуратор объекта ver.5.2»».

Подробные сведения о приложении «Конфигуратор извещений», используемого при конфигурировании скриптовой логики системы, приведены в «Руководстве «АРМ Терминал Дежурного Оператора» ver.5.2».





2. Скриптовая логика.

2.1. Назначение. Основные понятия.

Система ИСБ «777» строится на базе контроллеров того или иного типа в зависимости от круга решаемых задач по обеспечению безопасности объектов и персонала в соответствии с современными требованиями.

Каждый контроллер (КСО, КСО.Д, КСО-А, АПКП «Бирюза», БУ-40 «Гонг») является автономным устройством с энергонезависимой памятью, обеспечивающим работоспособность сектора, которым он управляет, вне зависимости от работоспособности других контроллеров а также работоспособности программно-аппаратных комплексов системы (серверов связи с оборудованием, удалённых терминалов наблюдения).

Режим работы контроллера, при котором он функционирует самостоятельно, без обмена данными с сервером связи называется автономным.

Несколько контроллеров могут объединяются в единую сеть, по средством сервера связи (Аппаратуры Связи о Обработки Данных «АСОД». При этом обеспечивается двусторонний приём-передача информации между каждым из контроллеров и сервером.

Режим работы контроллера в составе системы, при котором обеспечивается обмен данными с ПАК «АСОД» называется сетевым.

В простейшем случае, когда к серверу связи подключено несколько контроллеров, он осуществляет сбор информации от каждого контроллера по отдельности. При этом каждый из них функционирует по запрограммированной логике, описанной посредством приложения «Конфигуратор объекта», не влияя на функционирование других контроллеров. Для того, чтобы согласовать работу нескольких контроллеров между собой, необходимо провести конфигурирование скриптовой логики системы.

Основным назначением скриптовой логики является согласование работы нескольких контролеров, функционирующих в сетевом режиме, между собой. Т.е. применяя «скрипты», можно обеспечить взаимосвязь логики работы нескольких контроллеров, а соответственно также взаимосвязь логики работы подчинённого разным контроллерам оборудования системы (периферийного оборудования).

Помимо этого, используя скриптовую логику, можно реализовать практически любую нестандартную логику работы оборудования, которая недоступна при конфигурировании системы в приложении «Конфигуратор объекта», что позволяет строить систему любой сложности и обеспечивать практически любые индивидуальные требования к ней.

2.2. Принцип работы

В процессе работы системы формируются события – извещения о любом изменении состояния каждого из элементов системы, произошедшие по той или иной причине. К примеру, изменение состояния шлейфа пожарного адресного блока (шлейф перешёл в состояние «внимание») – событие, изменение состояния охранной зоны (переход зоны в состояние «неисправность») - событие, включение реле оповещения шлейфно-релейного модуля – тоже событие.

Программно-аппаратный комплекс «АСОД» (сервер связи с оборудованием) имеет набор команд, позволяющих параллельно внутренней логике контроллеров принудительно управлять многими элементами системы как вручную (по средством программно-аппаратного комплекса «ТДО» (приложения АРМ «Терминал Дежурного Оператора»)), так и автоматически - при помощи скриптовой логики.

Скрипт – есть сопоставление определённых событий и команд между собой. Иначе говоря, любому событию в системе, произошедшему с любым из его элементов, можно назначить набор любых команд, которые должны выполниться с любыми элементами данной системы при возникновении этого события.

Пример 1. Пожарная сигнализация (на базе АПКП «Бирюза») и система контроля доступа (на базе КСО) на объекте объединены в одну систему. При возникновении сигнала «пожар», необходимо открыть все электромагнитные замки в здании. В «скрипте» в данном случае любому событию о пожаре должны быть сопоставлены команды на открытие всех замков.

Пример 2. Объект оборудован пожарной сигнализацией на базе трёх контроллеров АПКП «Бирюза», системой дымоудаления в здании управляет оборудование, подчинённое первому контроллеру. Необходимо: при возникновении сигнала о пожаре в зонах путей эвакуации, которые защищают извещатели третьего контроллера обеспечить запуск системы дымоудаления. Скрипт в данном случае будет являться





сопоставлением событий о пожаре в зонах путей эвакуации третьего контроллера и команд на запуск устройств управления системой дымоудаления первого контроллера.

2.3.Особенности команд управления вспомогательными устройствами

В процессе функционирования каждый контроллер самостоятельно обеспечивает управление своими вспомогательными устройствами по заданной на этапе конфигурирования логике. Контроллер отслеживает состояние всех подчинённых устройств и в зависимости от этой логики осуществляет управление данными устройствами. Соответственно, для включения (выключения) какого либо устройства управления одной команды на включение (выключение) будет недостаточно, поскольку контроллер определит по своей логике, что устройство в данный момент должно быть выключено (включено) и сразу же произведёт его выключение (включение) обратно. Поэтому перед тем как посредством скрипта произвести включение/выключение какого-либо устройства управления, это устройство нужно заблокировать, иными словами запретить контроллеру управлять данным устройством. При этом оно перейдёт в состояние «заблокировано» и будет управляться только «АСОДом» до момента, пока не будет разблокировано.

Из вышесказанного следует, что если в скрипте присутствует команда включения/выключения вспомогательного устройства, то перед этой командой всегда должна присутствовать команда блокирования данного элемента. А если после выполнения команды управления необходимо отдать управление устройством обратно внутренней логике контроллера, после команды включения/выключения должна присутствовать команда разблокирования элемента.

При оперировании с периферийными элементами при формировании команд управления необходимо учитывать, что не для всех из них существуют такие команды. К примеру, для реле адресного блока управления, подчиняющемуся контроллеру сектора охраны КСО, не существует прямой команды «включить» (команда «включить» существует лишь для всего АБ-4(У), которое при включении запускает по запрограммированной логике процесс включения своих реле). В системе на базе АПКП «Бирюза» для всех устройств, входящих в состав зон управления (АБ-4(У), МШ-4(У), ППУ «Оберег» а также их реле), тоже не существует прямых команд управления (управление данными устройствами происходит посредством управления соответствующими направлениями автоматики (зонами управления)).

Наличие в системе для любого элемента необходимой команды можно проверить посредством приложения APM «Терминал Дежурного Оператора», кликнув по нему правой клавишей мыши в окне «структура системы». Все существующие для оперирования с данным элементом команды будут отображены в появившемся при этом контекстном меню управления.

2.4.Структура и синтаксис скриптов

Конфигурирование скриптовой логики системы производится посредством редактирования при помощи текстового редактора файлов commands.scr, events.scr и timers.scr (по умолчанию данные файлы располагаются в директории c:\rovalant\777\server\dbo сервера связи с оборудованием):

Commands.scr – «скрипт реакций системы»;

Events.scr – «скрипт системных состояний»;

Timers.scr – «скрипт системных таймеров».

Основным «скриптом», предназначенным для сопоставления событиям системы набора команд является файл **Commands.scr**. Данный файл описывает реакции системы на те или иные события, происходящие в ней.

В большинстве случаев в качестве события используется изменение состояния какого-либо элемента.

Имя	↑Тип	Размер
5		<dir></dir>
asod	ini	1 627
browser	ini	4 908
builder	ini	211
confmsg	ini	430
journal	ini	697
server	ini	2 595
video	ini	1 658
commands	SCF	2 679
events	SCF	2 039
timers	SCF	1 632

Пример 3. Есть набор пожарных зон, входящих в разные сектора. Если любая из них перейдёт в состояние «пожар», необходимо запустить определённое направление автоматики. Событием в данном случае будет являться переход любой из зон в состояние «пожар». Сопоставив для каждой зоны при переходе её в «пожар» команду на включение автоматики, мы посредством «скрипта реакций системы» **Commands.scr** опишем запуск от этих зон требуемого направления автоматики.





Существуют случаи, когда необходимо произвести определённое действие при условии, что несколько элементов системы находятся в определённом состоянии.

Продолжим пример 3.: после перехода всех сработавших пожарных зон в норму, необходимо выключить запущенное направление автоматики. В случае параллельного срабатывания нескольких пожарных зон сопоставление событию для каждой из них «переход зоны в состояние «норма»» команды выключения автоматики будет не совсем корректным, поскольку переход одной из зон в норму не означает перехода их всех в норму. Автоматику в данном случае нужно выключить лишь в случае, если каждая из требуемых зон перешла в норму. Для этого нам необходимо проверить состояние их всех и только в случае если каждая из них находятся в «норме», выключить автоматику.

Для формирования системного события при условии нахождения определённого набора элементов системы в определённом состоянии предназначен «скрипт системных состояний», описываемый посредством файла Events.scr.

В приведённом выше примере нам необходимо сформировать нестандартное для системы событие «нахождение каждой из набора пожарных зон в состоянии «норма»» и по этому событию выключить автоматику. Для этого мы посредством «скрипта системных состояний» **Events.scr** описываем данное событие, а затем этому событию посредством «скрипта реакций системы» **Commands.scr** сопоставляем команду на отключение автоматики.

Возможны случаи, когда необходимо автоматически выполнить какое-нибудь действие в системе в определённый момент времени суток.

Пример 4. В учреждении после окончания рабочего дня необходимо автоматически заблокировать все электронные замки и произвести автоматическую постановку всех помещений под охрану. Событием в данном случае будет являться наступление определённого времени суток (так называемый системный таймер).

Для формирования системного события при наступлении определённых значений времени суток предназначен «скрипт системных таймеров», описываемый посредством файла **Timers.scr**.

Далее рассмотрим подробно структуру и формат каждого из скриптов.

2.4.1.Скрипт реакций системы

Описание реакций системы на определённые события в ней производится посредством редактирования при помощи любого текстового редактора файла **Commands.scr** (по умолчанию файл находится в директории c:|vovalant|777|server|dbo сервера связи с оборудованием).

Информация в файле скрипта реакций системы должна быть представлена в следующем виде: первая строка файла содержит надпись "Скрипт Системных Реакций", удаление и редактирование которой не допускается. Затем разделённые как минимум одной пустой строкой располагаются секции скрипта, каждая из которых описывает реакции системы на определённое событие в ней. Количество секций в скрипте системных реакций неограниченно. Для удобства расшифровки информации между секциями имеется возможность оставлять комментарии. Для этого в начале строки, содержащей комментарий, должен быть установлен символ #.

Таким образом, в общем виде скрипт системных реакций имеет следующий формат:

```
''Скрипт Системных Реакций''
#Комментарии
S1
#Комментарии
S2
S3
.
.
```

, где Si — секции скрипта, N — их количество (количество событий в системе, при возникновении которых необходимо выполнить определённые команды).



SN



2.4.1.1. Формат секций

Каждая секция начинается со строки, содержащей открывающуюся фигурную скобку, и заканчивается строкой, содержащей закрывающуюся фигурную скобку. В первой строке секции содержится описание события в системе, на следующих строках — описание команд, которые должны выполниться при возникновении данного события. Количество команд в одной секции не должно превышать 255, т.е. при возникновении одного события имеется возможность выполнить до 255 команд. Комментарии, а также пустые строки в пределах секции не допускаются.

Таким образом, каждая секция Si имеет следующий формат:

```
{
E
C1
C2
.
.
.
.
.
.
.
.
.
.
```

, где E – событие системы, Ci – команды, которые должны выполниться при возникновении данного события.

2.4.1.2. Описание события в системе

Любое событие, произошедшее в системе, имеет свой источник.

Источником события может быть как логический элемент «структуры объекта», так и аппаратный элемент «структуры системы». К примеру, источником события «зона перешла в состояние «пожар»» является пожарная зона, которая перешла в «пожар». Источником события «переход шлейфа в состояние «неисправность»» является шлейф, который перешёл в «неисправность». Источником события «включено реле пожарной станции» является реле пожарной станции, которое включилось по каким-то причинам.

Формат строки E, описывающей событие в секции следующий:

?:X:Y:T-N

, где X – системный адрес элемента в структуре объекта, который является источником события;

Y – системный адрес элемента в структуре системы, который является источником события;

T – тип события;

N – номер события.

Каждый логический элемент из структуры объекта имеет свой индивидуальный четырёхзначный системный адрес в структуре, а аппаратный элемент из структуры системы семизначный системный адрес в своей структуре. Для того, чтобы определить номер элемента, необходимо запустить приложение «Конфигуратор объекта» (либо «Терминал Дежурного Оператора»), найти в окне «структура объекта/структура системы» необходимый элемент, выделить его левой клавишей мыши и в «окне конфигурирования» («окне состояния» для «ТДО») в поле «адрес» определить его системный адрес.

Источником конкретного события является лишь один элемент. При этом данный элемент относится либо к структуре объекта. Так событие «Переход зоны в состояние «Пожар»» относиться только к пожарной зоне – элементу структуры объекта, а «включение реле пожарной станции» к реле – элементу структуры системы.





Если описывается событие с элементом из структуры объекта, то адрес элемента в структуре системы в описании данного события маскируется адресом *255.255.255.255.255.255.255* и описание события при этом имеет вид:

?:X:255.255.255.255.255.255.255.T-N

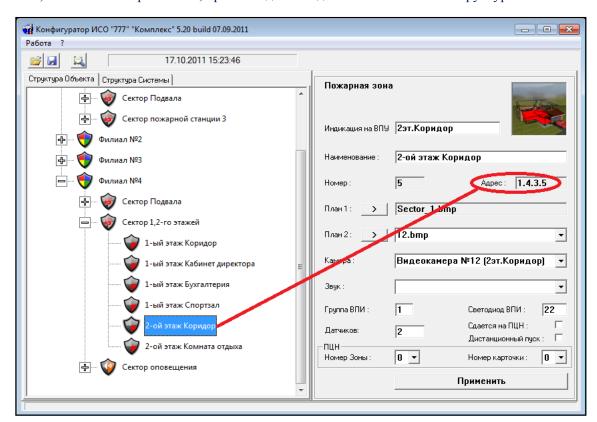
, где X – адрес элемента из структуры объекта, с которым произошло событие;

T,N – тип и номер события, произошедшего с данным элементом из структуры объекта.

Если же описывается событие с элементом из структуры системы, то адрес элемента в структуре объекта в данном событии маскируется адресом *255.255.255.255* и описание события имеет вид:

?:255.255.255.255:Y:T-N

, где Y – адрес элемента из структуры системы, с которым произошло событие; T , N – тип и номер события, произошедшего с данным элементом из структуры системы.



В примере, изображённом на рисунке, пожарная зона «2-ой этаж Коридор» имеет системный номер 1.4.3.5 (X=1.4.3.5). Здесь первая цифра — номер объединения объектов (всегда 1), вторая — номер объекта, третья — номер сектора данного объекта, четвёртая — номер зоны в данном секторе.

Т.к. данная зона является элементом структуры объекта, то в описании события адрес в структуре системы «маскируется». Т.е. событие, источником которого является данная зона, будет описываться следующим образом:

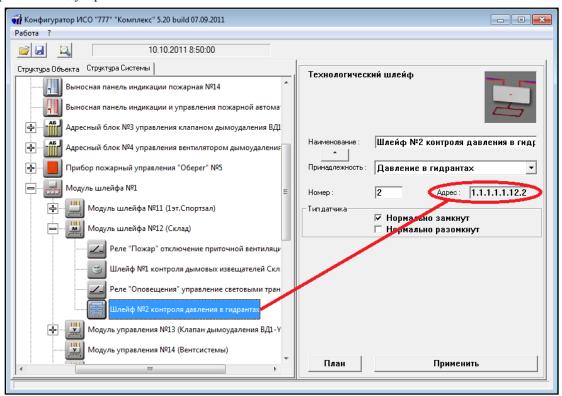
?: 1.4.3.5: 255.255.255.255.255.255.255.T-N

, где T, N – тип и номер события, произошедшего с пожарной зоной «2-ой этаж Коридор».





В примере на рисунке ниже технологический шлейф модифицированного шлейфно-релейного модуля, контролирующий давление в гидрантах имеет индивидуальный адрес 1.1.1.1.1.12.2. Здесь первая цифра – номер терминала дежурного оператора (всегда 1), вторая – номер «АСОДа», третья – номер магистральной линии связи «АСОДа», четвёртая – номер контроллера на данной линии связи, пятая – номер адресного (объектового) устройства, подключенного к данному контроллеру; шестая – номер периферийного устройства, подключенного к данному объектовому устройству; седьмая – номер подустройства данного периферийного подустройства.



Т.к. данный шлейф является элементом структуры системы, то в описании события адрес в структуре объекта «маскируется». Т.е. событие, происходящее с данным шлейфом будет описываться следующим образом:

?:255,255,255,255: 1.1.1.1.1.12.2:T-N

, где T, N – тип и номер события, произошедшего с данным технологическим шлейфом.

Список всех событий системы, а также их типы и номера перечислены в приложении «конфигуратор извещений». Данное приложение запускается как функция в режиме «администратор» программы APM «Терминал Дежурного Оператора», так и автономно (по умолчанию программа «Конфигуратор извещений» находиться в директории c:|rovalant|777|server| и представляет собой приложение confmsg.exe).

Для того, чтобы определить номер типа события и номер самого события достаточно отыскать в списке всех возможных извещений системы необходимое событие.

В примере, изображённом на рисунке ниже, событие «переход зоны в состояние «Пожар»», произошедшее в секторе пожарной станции «Бирюза» имеет тип **24**, номер **11**, а событие «переход шлейфа в состояние «сработка»» имеет тип **24**, номер **85**.









Итак, зная системный адрес элемента системы, номер и тип события, можно сформировать строку секции скрипта системных реакций, описывающую событие, произошедшее с данным элементом.

В приведённых выше примерах событие «Переход пожарной зоны «2-ой этаж Коридор»» будет описываться следующим образом:

?:<u>1.4.3.5</u>:255.255.255.255.255.255.255:<u>24-11</u>

А событие «Переход технологического шлейфа контроля давления в состояние «сработка»» будет описываться:

?:255.255.255.255:<u>1.1.1.1.1.12.2</u>:<u>24-85</u>

2.4.1.3. Применение в описании события шаблонов

Для упрощения написания скрипта системных реакций и сокращения количества в нём секций при описании одинаковых событий, происходящих со всеми элементами одного уровня, имеющими общий родительский элемент, используются шаблоны.

Шаблон «255» в разряде системного адреса элемента означает: «любой элемент на данной ветке системной иерархии».

Так, если адрес элемента в структуре объекта a.b.c.d, где a — номер объединения, b — номер объекта, c - номер объекта, d — номер зоны; то шаблон вида a.b.c.255 означает «любая зона» объекта b, сектора c. А шаблон вида a.b.255.255 означает - «любая зона любого сектора объекта b» либо «любой сектор объекта b».





К примеру, событие «Переход любой зоны седьмого сектора второго объекта в состояние «Пожар»» будет описываться следующим образом:

?: 1.2.7.255:255.255.255.255.255.255.255:24-11

Событие «Переход любой зоны второго объекта в состояние «Пожар»»:

?: 1.2.<u>255.255</u>:255.255.255.255.255.255.255:24-11

Событие «Перевод любого сектор второго объекта в ручной режим»:

?: 1.2.<u>255.255</u>:255.255.255.255.255.255.255:24-33

Аналогично для структуры системы. Адрес элемента имеет вид: a.b.c.d.e.f.g, где a - номер терминала дежурного оператора, b - номер «АСОДа», c - номер магистральной линии связи «АСОДа», d - номер контроллера на данной линии связи, e - номер адресного (объектового) устройства, подключенного на данный контроллер; f - номер периферийного устройства, подключенного к данному объектовому устройству; g - номер подустройства данного периферийного устройства.

Шаблон вида *a.b.c.d.e.f.255* означает «любое подустройство периферийного устройства *a.b.c.d.e.f..0*», шаблон *a.b.c.d.e.255.255* — «любое периферийное устройство любого устройства, подключенного к объектовому устройству с системным адресом *a.b.c.d.e.0.0*» либо «любое периферийное устройство, подключенное к объектовому устройству с адресом *a.b.c.d.e.0.0*» и так далее.

К примеру, событие «Переход любого технологического шлейфа шлейфно-релейного модуля №15, подключенного к модулю адресного шлейфа №2 контроллера АПКП «Бирюза» №12, который в свою очередь находится на магистральной линии №3 «АСОДа» №5, в «Пожар»» будет описываться следующим образом:

?:255.255.255.255:1.5.3.12.2.15.<u>255</u>:24-85

Событие «Переход любого шлейфа или извещателя любой пожарной станции пятого «АСОДа» в состояние «Неисправность»»:

?:255.255.255.255:1.5.255.255.255.255.255:24-26

При использовании шаблонов не допускается отсутствие шаблона «255» в разряде адреса дочернего элемента при наличии шаблона «255» в разряде адреса родительского элемента. Так, к примеру, шаблон, «1.1.255.3» (дословно обозначающий «третья зона любого сектора») неприменим. Секция, содержащая такое некорректное описание события функционировать не будет.





2.4.1.4. Описание простой команды

Управляющие команды могут быть адресованы только аппаратным элементам, входящих в структуру

Пример 5. Необходимо включить реле шлейфно-релейного модуля МШ-4. Команда в этом случае адресуется непосредственно данному реле и звучит как «Реле включить».

Команды управления логическими элементами структуры объекта производятся посредством команд, адресованных элементам структуры системы. При этом команды дополнительно содержат аргументы (об использовании в командах аргументов см.п.2.4.1.5).

Пример 6. Необходимо включить зону управления сектора пожарной станции. Команда в этом случае адресуется не зоне управления, а пожарной станции, контролирующей данный сектор, и звучит как «Пожарной станции включить зону управления».

Формат строки секции скрипта, описывающей простую команду без аргументов, следующий:

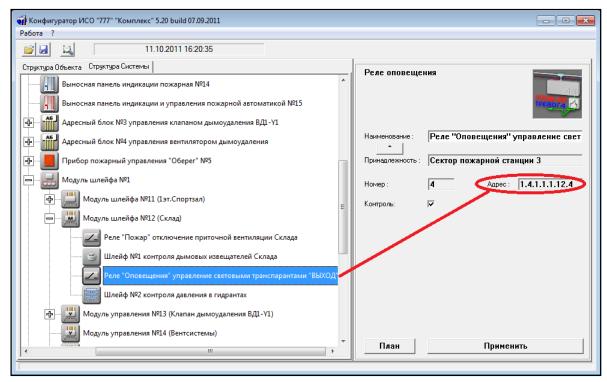
!:Z:T-N

, где Z – системный адрес элемента из структуры системы, которому адресована команда;

T – тип команды;

N – номер команды.

Как и в случае с событиями, адрес элемента, которому нужно отдать команду, определяется либо в «окне конфигурирования» приложения «Конфигуратор объекта» либо в «окне состояния» приложения «Терминал Дежурного Оператора», выделив необходимый элемент в окне «структура системы».



В примере на рисунке реле «оповещения» имеет системный адрес 1.4.1.1.1.12.4. Команда, адресованная данному реле, опишется следующим образом:

!:1.4.1.1.1.12.4:T-N

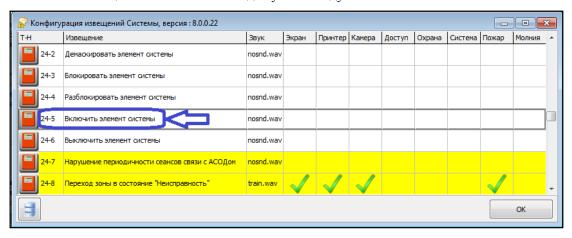
, где T — тип команды;

N – номер команды.





Так же как и список событий, список всех команд системы перечислен в приложении «Конфигуратор извещений». Для того, чтобы определить номер типа команды и номер самой команды достаточно отыскать в списке возможных извещений системы необходимую команду.



К примеру, для включения реле шлейфно-релейного модуля понадобится команда «включить элемент системы», имеющая тип 24 и номер 5.

В списке извещений «Конфигуратора извещений» не следует путать созвучные события и команды между собой. Так, к примеру, извещение «включить элемент системы» - является командой, а извещение «включен элемент системы» - событием.

Итак, зная адрес элемента в структуре системы, номер и тип команды, можно сформировать строку секции скрипта, описывающую команду, подаваемую данному элементу.

В приведённом выше примере команда «включить реле управления световыми указателями «ВЫХОД»» будет описываться следующим образом:

!: 1.4.1.1.1.12.4:24-5

2.4.1.5. Использование в командах аргументов

В некоторых случаях при передаче устройству управляющей команды ему необходимо дополнительно передать некоторые данные, сопряжённые с этой командой.

Пример 7. В секторе охраны и доступа необходимо поставить под охрану определённую охранную зону. В этом случае контроллеру КСО-515 отдаётся команда «поставить зону на оперативную охрану» и дополнительно, посредством аргумента передаётся номер зоны, которую нужно поставить под охрану.

Пример 8. В определённое время суток необходимо увеличить чувствительность периметральной сигнализации на базе адресного блока «Рубикон». В этом случае команде «установить уровень Рубикона» дополнительно посредством аргумента передаётся информация со значением устанавливаемого уровня чувствительности.

Таким образом, для передачи дополнительных данных в команде управления необходимо использовать аргументы. Формат команды с аргументами имеет следующий вид:

$!:Z:T-N[X_1-X_2-...-X_i]$

, где Z – системный адрес элемента в структуре системы, которому адресуется команда;

T – тип команды;

N – номер команды;

X— аргументы команды;

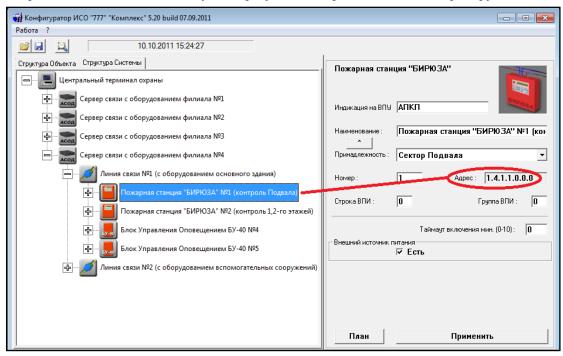
i – количество аргументов, дополнительно передаваемых в команде.





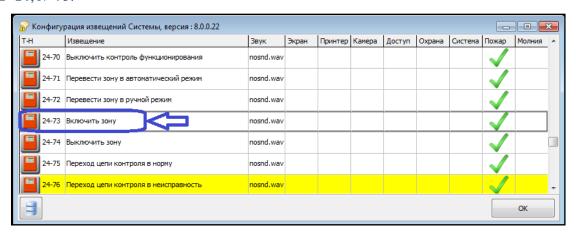
Таким образом, формат команды с аргументами имеет вид простой команды, после которой в квадратных скобках разделённые дефисом указываются дополнительно передаваемые аргументы.

К примеру, необходимо сформировать команду включения зоны управления, входящей в сектор пожарной станции «Бирюза». Команда в данном случае адресуется пожарной станции, контролирующей данный сектор.



Посредством окна конфигурирования приложения «Конфигуратор объекта» определяем адрес пожарной станции, контролирующей данный сектор: Z=1.4.1.1.0.0.0.

Далее посредством приложения «Конфигуратор извещений» определяем тип и номер команды «включить зону»: T=24, N=73.



В команде «включить зону» только один аргумент — номер включаемой зоны, поэтому количество аргументов: i=1.

Таким образом, команда данной пожарной станции, «включить зону управления» опишется следующим образом.

!: 1.4.1.1.0.0.0:24-73[X1]

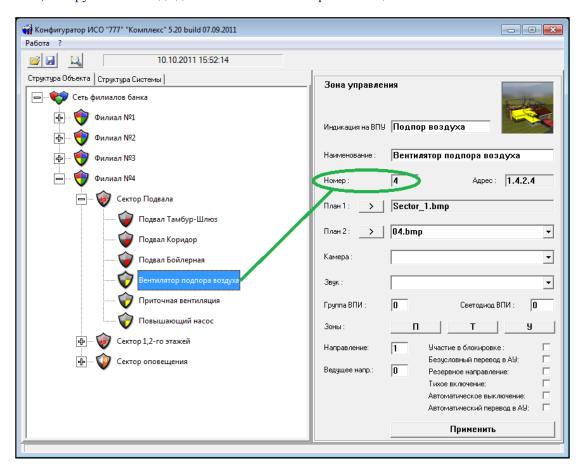
, где $\mathit{X1}$ – единственный аргумент команды – номер включаемой зоны.





Данный пример рассмотрен не случайно, поскольку в нём есть важная аппаратная особенность - зоны во внутренней логике работы контроллеров нумеруются от 0 до 255, а в устройствах индикации и программном обеспечении для удобства пользователя (чтобы исключить понятие «нулевая зона») - от 1 до 254. Поэтому первая зона в любом программном приложении соответствует нулевой зоне контроллера, вторая зона в программе — первой в контроллере, и так далее. Поскольку команда управления скрипта адресуется непосредственно контроллеру (аппаратной части), необходимо учитывать эту особенность и при указании в качестве аргумента номера зоны вводить аппаратный номер. Самый простой способ определения аппаратного номера — вычитание единицы из пользовательского номера.

Итак, посредством приложения «Конфигуратор объекта» определяем пользовательский номер зоны управления, которую по команде должна включить пожарная станция.



Учитывая указанную выше особенность адресации зон, четвёртой зоне управления в приложении «Конфигуратор объекта» будет соответствовать третья аппаратная зона контроллера, поэтому аргумент команды «включить зону» будет равен трём: $X_1 = 3$.

Итак, зная адрес элемента в структуре системы, номер и тип команды, количество и сами аргументы команды, можно сформировать строку секции скрипта, описывающую команду с аргументами, подаваемую данному элементу.

В приведённом примере команда пожарной станции «включить четвёртую зону управления вентилятором подпора воздуха» будет описываться следующим образом:

!: 1.4.1.1.0.0.0:24-73[3]





2.4.1.6. Использование в командах условий

Существуют ситуации, когда команда должна выполниться лишь при определённом условии – нахождении какого-либо из логических элементов системы (элементов «структуры объекта») в определённом состоянии.

Формат команды с условием выполнения, если определённый элемент структуры объекта находится в определённом состоянии, имеет следующий вид:

!: [Y:t=n]Z:T-N

, где Z – системный адрес элемента в структуре системы, которому адресуется команда;

T – тип команды;

N – номер команды;

Y – системный адрес элемента в структуре объекта, состояние которого нужно проанализировать ;

t — тип состояния;

n — номер состояния.

Формат команды с условием выполнения, если определённый элемент структуры объекта не находится в определённом состоянии, имеет следующий вид:

!: [Y:t!n]Z:T-N

, где Z – системный адрес элемента в структуре системы, которому адресуется команда;

T – тип команды;

N – номер команды;

Y – системный адрес элемента в структуре объекта, состояние которого нужно проанализировать;

t – тип состояния;

n — номер состояния;

Таким образом, формат команды с условием имеет вид простой команды (либо команды с аргументом), у которой перед указанием адреса элемента в структуре системы, которому адресуется команда, в квадратных скобках описывается условие нахождения определённого элемента структуры объекта в определённом состоянии.

Описание условия нахождения элемента структуры объекта в определённом состоянии имеет следующий вид:

[Y:t=n]

, гдеY – системный адрес элемента структуры объекта, состояние которого анализируется ;

t, n – тип и номер состояния, в котором должен находится данный элемент, чтобы выполнилось условие.

Описание условия не нахождения элемента структуры объекта в определённом состоянии имеет следующий вид:

[Y:t!n]

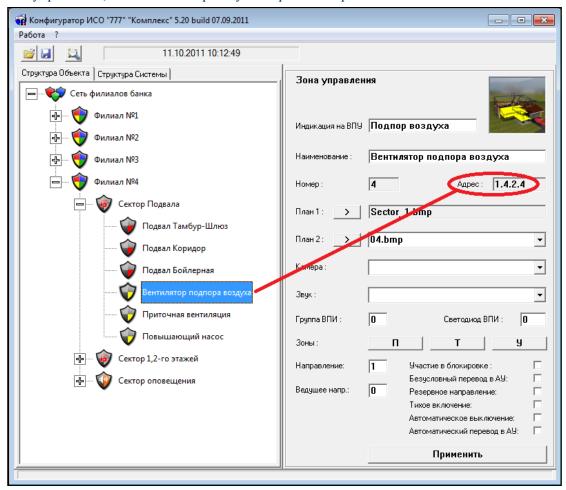
, гдеY – системный адрес элемента структуры объекта, состояние которого анализируется ;

t, n – тип и номер состояния, в котором не должен находится данный элемент, чтобы выполнилось условие.

Пример 9. Вентилятор подпора воздуха из примера, рассмотренного в п.2.4.1.5 должен запуститься лишь при условии, что направление автоматики (зона управления), в которую он входит, находится в автоматическом режиме функционирования. В данном примере нам необходимо в команду о включении добавить условие нахождения зоны управления «Вентилятор подпора воздуха» в автоматическом режиме.



Опишем данное условие. Посредством приложения «Конфигуратор объекта» определяем системный адрес зоны управления, состояние которой нужно проанализировать: *Y=1.4.2.4*.



Таким образом, условие нахождения данной зоны в определённом состоянии опишется как:

[1.4.2.4:t=n]

, где t, n — тип и номер состояния, в котором должна находится зона управления «Вентилятор подпора воздуха» для выполнения условия.

Если бы нам требовалось выполнения команды при условии, что данная зона не находится в каком-то определённом состоянии, то условие описывалось бы как:

[1.4.2.4:t!n]

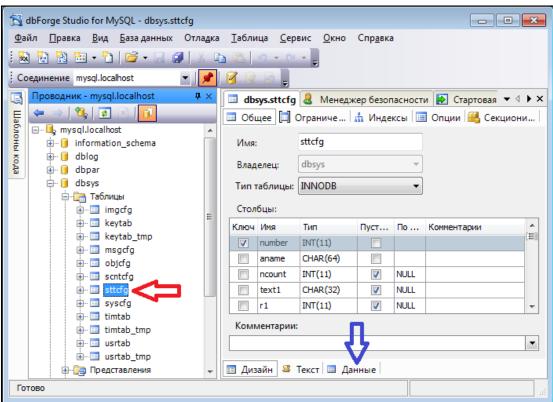
, где t, n — тип и номер состояния, в котором не должна находится зона управления «Вентилятор подпора воздуха» для выполнения условия.

Все виды состояний элементов в системе, а также их типы и номера содержатся в таблице «sttcfg» базы данных «dbsys» сервера управления базами данных MySQL. Доступ к данным таблицы производится посредством приложения dbForge Studio for MySQL.

Чтобы открыть данные таблицы «sttcfg» необходимо в окне проводника приложения dbForge Studio for MySQL двойным щелчком левой клавиши мыши выделить данную таблицу и в открывшемся окне «dbsys.sttcfg», содержащем информацию о таблице нажать кнопку «Данные». При этом в окне «dbsys.sttcfg» отобразятся данные таблицы.



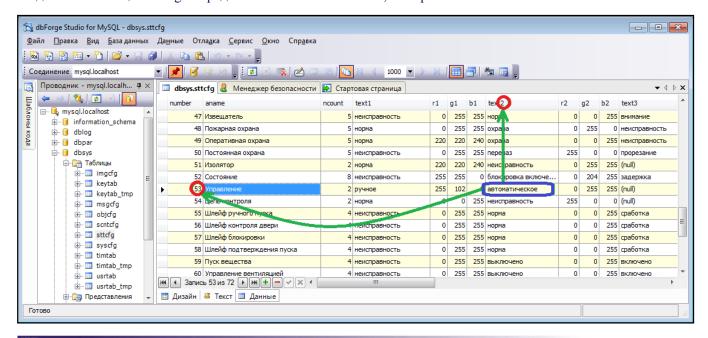




Строки таблицы содержат описание всех возможных состояний элементов системы.

В столбце **number** содержится порядковый номер параметра элементов, который может принимать то либо иное состояние (данный номер и является типом состояния t). В столбце **aname** содержится название данного параметра. Столбец **ncount** содержит значение количества состояний, которое может принимать параметр. Далее в столбцах **text**_n представлена текстовая расшифровка состояний, который может принимать данный параметр. Здесь n есть номер состояния, которое может принимать параметр.

В нашем примере мы описываем условие «зона управления находится в автоматическом режиме управления». Необходимый параметр — «управление», необходимое значение параметра «автоматическое». По данным таблицы «sttcfg» определяем тип состояния t=53, номер состояния n=2.







Теперь, зная системный номер элемента структуры объекта, тип и номер состояния, в котором должен находиться данный элемент можно описать условие. Для нашего примера условие «зона управления вентилятором подпора воздуха находится в автоматическом режиме управления» опишется следующим образом:

```
[1.4.2.4:53=2]
```

Добавив это условие в команду !:1.4.1.1.0.0.0:24-73[3] включения данной зоны управления, описанную в п.2.4.1.5, получим команду с условием: «включить вентилятор подпора воздуха при условии, что он находится в автоматическом режиме управления». Данная команда в итоге будет иметь вид:

!: [1.4.2.4:53=2]1.4.1.1.0.0.0:24-73[3]

2.4.1.7. Формирование секции

Секция имеет следующий формат:

{
E
C1
C2
.
.
.
.
.
.
.

, где E — описание события в системе, Ci — описания команд, которые должны выполниться при возникновении данного события.

Зная описание данного события и описание команд (реакций на него), можно сформировать секцию скрипта системных реакций.

Рассмотрим пример построения секции на основе примеров, приведённых в п.2.4.1.2, 2.4.1.4 и 2.4.1.6.

Задача: при возникновении пожара в коридоре 2-го этажа здания включить световые указатели «ВЫХОД» и запустить вентилятор подпора воздуха при условии, что он находится в автоматическом режиме.

Событие «переход зоны «2-ой этаж Коридор» в состояние «Пожар»» описано в примере п.2.4.1.2:

?:1.4.3.5:255.255.255.255.255.255.255:24-11

Команда «включить реле шлейфно-релейного модуля, управляющего световыми указателями «ВЫХОД»» описана в примере п.2.4.1.4:

!: 1.4.1.1.1.12.4:24-5

Для того, чтобы произвести управление реле шлейфно-релейного модуля, данное реле необходимо предварительно заблокировать (см.п.2.3). Команда «блокировать реле шлейфно-релейного модуля, управляющего световыми указателями «ВЫХОД»» опишется следующим образом:

!: 1.4.1.1.1.12.4:24-3

Команда «включить вентилятор подпора воздуха при условии, что он находится в автоматическом режиме управления» описана в примере п.2.4.1.6:







Перед прямым управлением блокируются только аппаратные элементы структуры системы, блокирование зоны управления не требуется (к тому же такой команды в системе для зон управления не существует).

Теперь сформируем секцию, которая выполнит поставленную задачу. Секция будет иметь следующий вид:

```
{
    ?:1.4.3.5:255.255.255.255.255.255.255:24-11
    !: 1.4.1.1.1.12.4:24-3
    !: 1.4.1.1.1.12.4:24-5
    !: [1.4.2.4:53=2]1.4.1.1.0.0.0:24-73[3]
}
```

Если необходимо включить данные устройства от каких-либо других пожарных зон, для каждой из них необходимо сформировать в скрипте системных реакций аналогичную секцию. В случае, если данные устройства будут включаться от всех пожарных зон сектора (либо нескольких секторов), при использовании шаблонов в описании события (см.п.2.4.1.3) для них всех достаточно одной секции.

При формировании в скрипте системных реакций секций автоматического включения устройств, необходимо также продумать, каким образом будет проводиться дальнейшее выключение этих устройств. В случае автоматического выключения необходимо дополнительно сформировать секции, описывающие данное выключение.

2.4.1.8. Особенности формирования скрипта системных реакций

Скрипт формируется набором секций, разделённых между собой пустыми строками.

При его формировании необходимо учитывать некоторые особенности.

Первые строки, описывающие события, в каждой секции должны быть уникальными, т.е. одно и то же событие, произошедшее с одним и тем же элементом не должно повторяться в разных секциях. При наличии секций с одинаковыми описаниями событий отработает только первая секция с данным событием и элементом, а последующие секции с таким же событием и адресом элемента в первой строке отрабатывать не будут.

Данное ограничение не распространяется на случаи, когда в первой строке одной из секций описано событие и адрес элемента, а в другой секции описано такое же событие и шаблон этого элемента. Т.е., к примеру, в разных секциях не может присутствовать одно и то же событие «переход зоны «2-ой этаж Коридор» в состояние «Пожар»». Но возможно в одной секции присутствие события «переход зоны «2-ой этаж Коридор» в состояние «Пожар»», в другой — «переход любой зоны данного сектора в состояние «Пожар»» а в третьей - «переход любой зоны объекта в состояние «Пожар»». Иначе говоря, наличие разных секций с одинаковыми описаниями событий не допускается.

Поскольку опрос и управление своими устройствами осуществляет только один определённый сервер связи (см.п.2.1), то событие должно происходить в пределах объекта, к которому относиться данный сервер («АСОД»), а команда управления должна формироваться для устройств, находящиеся в пределах данного сервера. Иначе говоря, скриптовая логика работает в пределах одного объекта и её функционирование между разными серверами связи с оборудованием невозможно.





2.4.1.9. Некоторые приёмы, используемые при конфигурировании функционирования пожарной автоматики

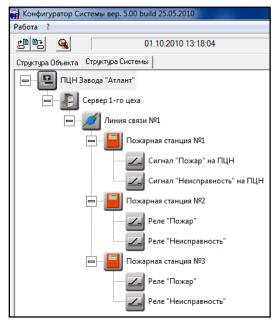
В системах пожарной сигнализации при необходимости управления устройствами пожарной автоматики посредством скрипта в качестве побудителя рекомендуется использование события из структуры объекта «переход зоны в состояние «Пожар»», поскольку событие из структуры системы «переход извещателя (шлейфа) в состояние «Пожар»» не всегда соответствует правильной логике работы пожарной автоматики.

К примеру, необходимо по скриптовой логике открыть клапан дымоудаления, а сигнал «Пожар» в зоне эвакуации формируется от двух адресных пожарных извещателей. В этом случае событие «переход извещателя в состояние «Пожар»» для запуска автоматики не подходит, поскольку после перехода одного извещателя в «Пожар» пожарная зона перейдёт только в состояние «Внимание», а это является недостаточным условием для запуска системы дымоудаления.

В некоторых случаях, когда нужно сформировать команду от события одного типа любого из элементов контроллера, удобно использование промежуточного элемента, который по своей внутренней логике реагирует на эти события.

Например, необходимо чтобы выход управления контроллера «неисправность» сработал не только от неисправности в секторе данного контроллера, но и от возникновения любой неисправности в секторах двух других контроллеров, а выход «пожар» от любого пожара на всех контроллерах. В данном случае, чтобы не описывать каждое событие с каждым типом неисправности, а также не описывать все события перехода пожарных зон в состояние «пожар», в качестве события «возникновения неисправности элементов второго (третьего) контроллера» можно использовать событие «включение реле «неисправность» второго (третьего) контроллера» соответственно. Аналогично с событиями «пожар» - включение реле «пожар» второго или третьего контроллера будет являться признаком возникновения пожара в любой из зон данных контроллеров. Кроме того событие «выключение реле «пожар» («неисправность»)» с большей долей вероятности скажет о переходе всех зон в «норму» нежели использование шаблона «переход любой зоны в норму».

Секции скрипта системных реакций, описывающие включения реле «пожар» первой станции от сигнала «пожар» на двух других станциях будет выглядеть следующим образом:



```
?:255.255.255.255: <u>1.1.1.2.0.1.0</u>:24-20 #включилось реле «пожар» 2-ой станции
!: 1.1.1.1.0.1.0:24-3
                                  #заблокировать реле «пожар» 1-ой станции
!: 1.1.1.1.0.1.0:24-5
                                       #включить реле «пожар» 1-ой станции
?:255.255.255.255: <u>1.1.1.3.0.1.0</u>:24-20 #включилось реле «пожар» 3-ей станции
!: 1.1.1.1.0.1.0:24-3
                                  #заблокировать реле «пожар» 1-ой станции
!: 1.1.1.1.0.1.0:24-5
                                       #включить реле «пожар» 1-ой станции
?:255.255.255: <u>1.1.1.2.0.1.0</u>:24-21 #выключилось реле «пожар» 2-ой станции
!: 1.1.1.1.0.1.0:24-6
                                      #выключить реле «пожар» 1-ой станции
!: 1.1.1.1.0.1.0:24-4
                                  #разблокировать реле «пожар» 1-ой станции
?:255.255.255.255: <u>1.1.1.3.0.1.0</u>:24-21</u> #выключилось реле «пожар» 3-ей станции
!: 1.1.1.1.0.1.0:24-6
                                       #выключить реле «пожар» 1-ой станиии
!: 1.1.1.1.0.1.0:24-4
                                  #разблокировать реле «пожар» 1-ой станции
```

В данном примере реле «пожар» первой станции от сигнала «пожар» на этой станции будет срабатывать по своей логике (внутренней логике контроллера), а от сигнала «пожар» на двух других станциях по скриптовой логике. В приведённом примере перед командами выключения реле первой станции отсутствует команда блокировки данного реле, поскольку предполагается, что после включения реле остаётся в заблокированном состоянии (разблокировка вручную через «Терминал Дежурного Оператора» не





происходит). В ином случае перед командой выключения дополнительно должна присутствовать команда блокировки.

Следует также отметить, что вариантом формирования секций скрипта, при котором команды одних секций адресуется элементу, который в других секциях сам является источником события, необходимо пользоваться, предусмотрев все возможные последствия в логике работы данного элемента. Так, в приведённом выше примере вариант включения реле каждой станции при включении реле любой другой станции не подходит. Этот вариант привёл бы к включению и блокированию реле первой и третьей станции в случае включения реле на второй. Включение реле на первой (третьей) станции в свою очередь приведёт к блокированию реле второй станции. В итоге реле всех трёх станций окажутся заблокированы, и выключить/разблокировать их можно будет только вручную (посредством АРМ «ТДО»).

При описании секции выключения пожарной автоматики в качестве события, напрашивается использование события «переход зоны в состояние «норма»», однако это будет не совсем корректно, если устройство запускается от нескольких побудительных пожарных зон, поскольку событие перехода одной пожарной зоны в состояние «норма» не означает перехода всех сработавших зон в норму. Поэтому, в случае одновременного срабатывания нескольких зон, а затем перехода одной из них в норму произойдёт выключение устройств пожарной автоматики, несмотря на то, что остальные пожарные зоны всё ещё находятся в «пожаре». В данном случае для выключения автоматики удобно задействовать скрипт системных состояний (см.п.2.4.2), который сформирует нестандартное событие при нахождении всех выбранных пожарных зон в состоянии «норма». Используя данное событие, в скрипте реакций системы можно сформировать секцию, описывающую корректное выключение устройств пожарной автоматики.

2.4.2. Скрипт системных состояний

«Скрипт системных состояний» предназначен для формирования системного события при условии нахождения набора определённых логических элементов системы в определённом состоянии.

Конфигурирование скрипта системных состояний производится посредством редактирования при помощи любого текстового редактора файла **Events.scr** (по умолчанию файл находится в директории *c:\rovalant\777\server\dbo* сервера связи с оборудованием).

Информация в файле скрипта системных состояний должна быть представлена в следующем виде: первая строка файла содержит надпись "Скрипт Системных Состояний", удаление и редактирование которой не допускается. Затем разделённые как минимум одной пустой строкой располагаются разделы, описывающие состояния определённого набора элементов системы, и формирующие при достижении данными элементами этих состояний, системные события «системное состояние і». Для удобства расшифровки информации между данными разделами имеется возможность оставлять комментарии. Для этого в начале строки, содержащей комментарий, должен быть установлен символ #.

В общем виде скрипт системных состояний имеет следующий формат:

```
"Скрипт Системных Состояний" #Комментарии S1 #Комментарии S2 S3 . . . . . . . . .
```

, где Si – разделы скрипта системных состояний, N – их количество (количество генерируемых в системе событий «системное состояние і», при достижении определёнными элементами нужных состояний).



SN



Каждый раздел начинается со строки, имеющей следующий вид:

!S:i

, где i – номер раздела и номер системного состояния (события «системное состояние i»).

Вторая строка раздела содержит открывающуюся фигурную скобку.

На последующих строках описываются элементы структуры объекта и их состояния в формате:

X:t-n

, где X – системный адрес элемента в структуре объекта;

t ,n — тип и номер состояния, в котором должен находится данный элемент, чтобы сформировалось событие «системное состояние i».

Заканчивается раздел строкой, содержащей закрывающуюся фигурную скобку.

Количество строк, описывающих состояния элементов, в одном разделе не должно превышать 255, т.е. сформировать событие «системное состояние і» возможно, при условии наличия у 255 элементов структуры объекта определённых состояний. Комментарии, а также пустые строки в пределах раздела не допускаются.

Продолжим пример, рассмотренный в п.2.4.1.7.

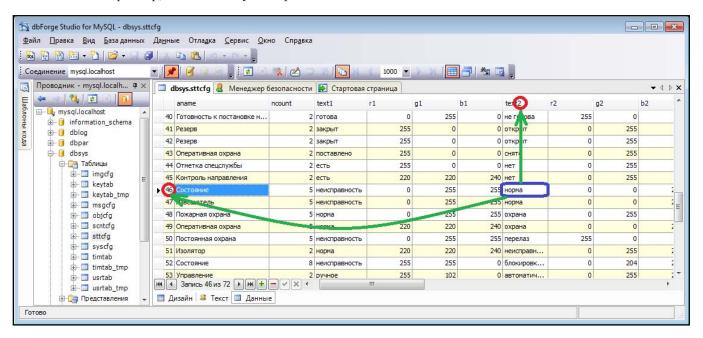
Задача: выключить световые указатели «ВЫХОД» и остановить вентилятор подпора воздуха при условии, что пожарные зоны №5,6 сектора 1,2-го этажа после срабатывания перешли в состояние «норма».

Для выполнения данной задачи необходимо посредством скрипта системных состояний сформировать событие «системное состояние і», означающее «нахождение выбранных зон в состояние «норма»» и затем сопоставить данному событию в скрипте реакций системы команды выключения устройств.

Сформируем раздел скрипта системных состояний, описывающий нахождение выбранных зон в состоянии «норма».

Согласно примеру из п.2.4.1.2 выбранные пожарные зоны имеют системные адреса 1.4.3.5, 1.4.3.6.

Согласно п.2.4.1.6 посредством приложения dbForge Studio for MySQL в таблице sttcfg определяем единственный параметр, имеющийся у пожарной зоны – «состояние».



Тип состояния в данном случае t=46, номер состояния «норма» n=2.





Зная номера требуемых элементов структуры объекта, тип и номер состояния, в котором они должны находиться, для формирования события «системное состояние i», можно сформировать раздел скрипта системных состояний.

Для приведённого примера раздел опишется:

```
!S:1
{
1.4.3.5:46-2
1.4.3.6:46-2
}
```

Данный раздел описывает первое (i=1) системное состояние «пожарные зоны с системными номерами 1.4.3.5 и 1.4.3.5 находятся в состоянии «норма»» и формирует по нему событие «системное состояние 1».

В скрипте реакций системы источником события «системное состояние і» всегда является элемент структуры системы «Терминал ДО», имеющий адрес 1.0.0.0.0.0.0. Тип события «системное состояние і» T=50. Таким образом, описание события «системное состояние і» в скрипте реакций системы согласно п.2.4.1.2 будет всегда выглядеть следующим образом:

```
?:255.255.255.255: 1.0.0.0.0.0.0:50-i
```

, где і – номер системного состояния.

В приведённом примере событие «пожарные зоны №5,6 сектора 1,2-го этажа после срабатывания перешли и находятся в состоянии «норма» в скрипте реакций системы опишется:

```
?:255.255.255.255: 1.0.0.0.0.0.0:50-1
```

Для выполнения описанной выше задачи сформируем секцию скрипта реакций системы.

Для того, чтобы произвести выключение реле шлейфно-релейного модуля, данное реле необходимо предварительно заблокировать (см.п.2.3). Команда «блокировать реле шлейфно-релейного модуля, управляющего световыми указателями «ВЫХОД»» согласно п.2.4.1.4 опишется следующим образом:

!: 1.4.1.1.1.12.4:24-3

Перед выключением команду «заблокировать» можно опустить в том случае, если элемент системы управляется только по скриптовой логике. В данном случае - если элементу МШ-4 в «Конфигураторе объекта» не назначена ни одна пожарная зона побуждения и предполагается, что включение реле шлейфнорелейного модуля могло произойти только посредством предыдущих команд скрипта реакций системы «заблокировать» и «включить», т.е. в данный момент реле уже находится в заблокированном состоянии.

Команда «выключить реле шлейфно-релейного модуля, управляющего световыми указателями «ВЫХОД»» согласно п.2.4.1.4 опишется:

!: 1.4.1.1.12.4:24-6

В том случае, если элемент системы управляется не только по скриптовой логике, но и по внутренней, после выключения его необходимо разблокировать. В данном случае - если элементу МШ-4 в «Конфигураторе объекта» назначены некоторые пожарные зоны побуждения и предполагается, что включение реле шлейфно-релейного модуля может произойти как посредством команд скрипта реакций системы «заблокировать» и «включить», так и посредством внутренней команды пожарной станции «включить». Для того, чтобы после выключения реле передать управление им пожарной станции, его необходимо разблокировать.





Команда «разблокировать реле шлейфно-релейного модуля, управляющего световыми указателями «ВЫХОД»» согласно п.2.4.1.4 опишется:

```
!: 1.4.1.1.1.12.4:24-4
```

Команда «выключить вентилятор подпора воздуха» согласно п.2.4.1.5 опишется:

```
!: 1.4.1.1.0.0.0:24-74[3]
```

Итак, описав в скрипте системных состояний формирование события «системное состояние 1», а также зная описание данного события, и команд которые должны по нему выполниться, можно сформировать секцию скрипта реакций системы, которая опишет поставленную задачу «выключить световые указатели «ВЫХОД» и остановить вентилятор подпора воздуха при условии, что пожарные зоны №5,6 сектора 1,2-го этажа после срабатывания перешли в состояние «норма»»:

```
{
?:255.255.255.255: 1.0.0.0.0.0.0:50-1
!: 1.4.1.1.1.12.4:24-3
!: 1.4.1.1.1.1.12.4:24-6
!: 1.4.1.1.1.1.12.4:24-4
!: 1.4.1.1.0.0.0:24-74[3]
}
```

При необходимости в скрипте системных состояний дополнительно можно описать «системное состояние 2», «системное состояние 3» и т.д., а затем в скрипте системных реакций описать производимые действия в системе при переходе выбранных элементов в данные состояния.

2.4.3. Скрипт системных таймеров

«Скрипт системных таймеров» предназначен для формирования системного события при наступлении определённых значений времени суток

Конфигурирование скрипта системных таймеров производится посредством редактирования при помощи любого текстового редактора файла **Timers.scr** (по умолчанию файл находится в директории *c:\rovalant\777\server\dbo* сервера связи с оборудованием).

Информация в файле скрипта системных таймеров должна быть представлена в следующем виде: первая строка файла содержит надпись "Скрипт Системных Таймеров", удаление и редактирование которой не допускается. Затем разделённые как минимум одной пустой строкой располагаются разделы, описывающие временные точки, при достижении которых генерируются системные события «системный таймер і». Для удобства расшифровки информации между данными разделами имеется возможность оставлять комментарии. Для этого в начале строки, содержащей комментарий, должен быть установлен символ #.

В общем виде скрипт системных таймеров имеет следующий формат:





, где Si – разделы скрипта системных таймеров, N – их количество (количество генерируемых в системе событий «системный таймер і», при достижении определённых временных точек).

Каждый раздел начинается со строки, имеющей следующий вид:

!T:i

, где i – номер раздела и номер системного таймера (события «системный таймер і»).

Вторая строка раздела содержит открывающуюся фигурную скобку.

На последующих строках в формате ЧЧ:ММ описываются временные точки, при наступлении которых должно сгенерироваться системное событие «системный таймер і»:

Заканчивается раздел строкой, содержащей закрывающуюся фигурную скобку.

Количество временных точек в одном разделе не должно превышать 255, т.е. сформировать событие «системный таймер і» возможно в 255 временных точках одних суток. Комментарии, а также пустые строки в пределах раздела не допускаются.

Пример 10. В административном здании необходимо закрыть электромагнитный замок двери доступа для посетителей в начале обеденного перерыва в 13-30 и по окончании рабочего дня в 18-00. Открыть данный замок необходимо по окончании обеда в 14-30 и в начале рабочего дня в 8-00.

Для выполнения данной задачи необходимо посредством скрипта системных таймеров сформировать два события «системный таймер 1» и «системный таймер 2». Первый системный таймер должен срабатывать в 13-30 и 18-00, второй – в 8-00 и 14-30.

Для приведённого примера разделы скрипта системных таймеров опишутся следующим образом:

```
!T:1
{
13:30
18:00
}
!T:2
{
8:00
14:30
}
```

Данные разделы в указанные моменты времени суток формируют события «системный таймер 1» (в 13-30и 18-00) и «системный таймер 2» (в 8-00 и 14-30).

В скрипте реакций системы источником события «системный таймер і» всегда является элемент структуры системы «Терминал ДО», имеющий адрес 1.0.0.0.0.0.0. Тип события «системный таймер і» T=51. Таким образом, описание события «системный таймер і» в скрипте реакций системы согласно п.2.4.1.2 будет всегда выглядеть следующим образом:

```
?:255.255.255.255: 1.0.0.0.0.0.0.0:51-i
```

, где і – номер системного таймера.

В приведённом примере событие «системный таймер 1» при котором необходимо закрыть электромагнитный замок опишется:

?:255.255.255.255: 1.0.0.0.0.0.0:51-1

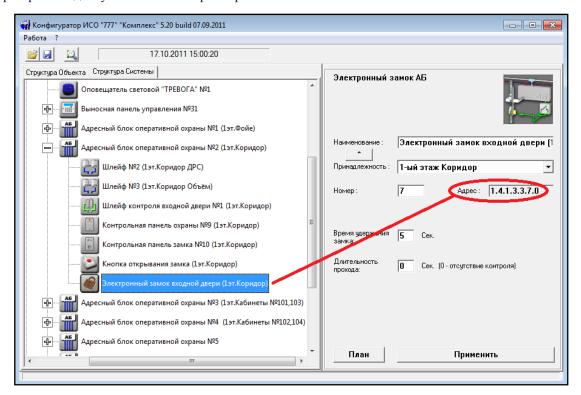




Событие «системный таймер 2» при котором необходимо открыть электромагнитный замок опишется:

?:255.255.255.255: 1.0.0.0.0.0.0:51-2

В примере электромагнитным замком управляет реле охранного адресного блока АБ-4(O), входящего в сектор охраны и доступа на базе контроллера КСО-512.



Сформируем секцию скрипта реакций системы.

Для того, чтобы произвести открытие замка, реле «электронный замок» необходимо предварительно заблокировать (см.п.2.3). Команда «блокировать реле АБ-4(O), управляющее ЭМЗ» согласно п.2.4.1.4 опишется следующим образом:

!: 1.4.1.3.3.7.0:34-20

Команда «открыть электромагнитный замок» («включить реле управления ЭМЗ») опишется:

!: 1.4.1.3.3.7.0:34-22

Команда «закрыть электромагнитный замок» («выключить реле управления ЭМЗ») опишется:

!: 1.4.1.3.3.7.0:34-23

Для того, чтобы сотрудники могли проходить через двери, используя личные электронные ключи в обеденное время и после окончания работы, реле управления замком после выключения необходимо разблокировать (передать управление данным реле контроллеру КСО-512). Команда разблокировки реле управления ЭМЗ опишется в данном случае:

!: 1.4.1.3.3.7.0:34-21





Итак, описав в скрипте системных таймеров формирование событий «системный таймер 1» и «системный таймер 2»,, а также зная описание данных событий, и команд которые должны по нему выполниться, можно сформировать секции скрипта реакций системы, которые опишут поставленные задачи.

Секция скрипта реакций системы, выполняющая закрытие электромагнитного замка в 13-30 и 18-00 опишется:

```
{
?:255.255.255.255: 1.0.0.0.0.0.0:51-1
!: 1.4.1.3.3.7.0:34-23
!: 1.4.1.3.3.7.0:34-21
}
```

Секция скрипта реакций системы, выполняющая открытие электромагнитного замка для свободного доступа посетителей через дверь в 8-00 и 14-30 опишется:

```
{
?:255.255.255.255: 1.0.0.0.0.0.0:51-2
!: 1.4.1.3.3.7.0:34-20
!: 1.4.1.3.3.7.0:34-22
}
```

При необходимости в скрипте системных таймеров дополнительно можно описать «системный таймер 3», «системный таймер 4» и т.д., а затем в скрипте системных реакций описать производимые действия в системе при «срабатывании» данных таймеров.

