

MiniSCADA Spica

Руководство программиста

АННОТАЦИЯ

Настоящий документ предназначен для программиста, использующего программу MiniSCADA Spica, работающую под управлением операционной системы Microsoft Windows и применяемую в автоматизированных системах управления предприятиями и технологическими процессами.

Настоящий документ может быть использован как руководство по организации человеко-машинного интерфейса для доступа к данным контроллеров и устройств по протоколам Modbus RTU, Modbus TCP, а также отображения этих данных на экране монитора в удобной и понятной форме.

В настоящем документе содержатся краткие сведения о программном и аппаратном обеспечении, которые необходимы для корректной работы программы MiniSCADA Spica, приведены сведения по установке и запуску ее на компьютере программиста, описаны способы и методы конфигурирования структур данных, используемых для визуализации технологического процесса и управления им.

СОДЕРЖАНИЕ

1. Назначение и условия применения программы	5
1.1. Назначение программы	5
1.2. Функции, выполняемые программой	5
1.3. Условия, необходимые для выполнения программы	5
1.3.1 Требования к составу и параметрам периферийных устройств	5
1.3.2 Требования к программному обеспечению	5
1.3.3 Требования к пользователю	6
2. Характеристика программы	6
2.1. Описание основных характеристик программы	6
2.1.1 Временные характеристики	6
2.2. Описание основных особенностей программы	7
2.2.1 Версии программы для различных операционных систем	7
2.2.2 Экранные формы	7
2.2.3 Протокол Modbus	7
2.2.4 Основные понятия	8
3. Обращение к программе	9
3.1. Установка программы	9
3.2. Загрузка и запуск программы	9
3.3. Выполнение программы	9
3.3.1 Создание конфигурации проекта	9
3.3.1.1 Общие сведения	9
3.3.1.2 Формат файлов конфигурации	13
3.3.1.3 Секция [port]. Настройка каналов связи	14
3.3.1.4 Секция [modbus]. Настройка протокола Modbus	15
3.3.1.5 Секция [pcu]. Добавление PCU	16
3.3.1.6 Секция [pw]. Настройка паролей доступа	17
3.3.1.7 Секция [color]. Создание цветов	18
3.3.1.8 Секция [bmp]. Добавление изображений	19
3.3.1.9 Секция [str]. Создание строк	20
3.3.1.10 Секция [seq]. Создание последовательностей	21
3.3.1.11 Секция [rect]. Создание боксов	22
3.3.1.12 Секция [font]. Создание шрифтов	23
3.3.1.13 Секция [action]. Создание действия	24
3.3.1.14 Секция [task]. Задание периодических действий	27
3.3.1.15 Секция [dialog]. Создание диалоговых окон	28
3.3.1.16 Секция [stamp]. Создание шаблонов	30
3.3.1.17 Секция [form]. Конфигурация основного окна проекта	31
3.3.1.18 Секция [start]. Задание действий при запуске	34
3.3.1.19 Секция [limit]. Задание пределов значений переменных	35
3.3.1.20 Секция [var]. Создание переменных	36
3.3.1.21 Секция [arc]. Функция архивирования	42
3.3.1.22 Секция [alarm]. Сводка тревог	44
3.3.1.23 Секция [time]. Синхронизация времени	44
3.3.1.24 Секция [print]. Создание команд печати	46
3.3.1.25 Секция [numpad]. Размер цифровой панели	51
3.3.1.26 Список зарезервированных обозначений	52
3.3.1.27 Формат файла инициализации Pult.ini	55
3.3.1.28 Формат файла инициализации Pult_xp.ini	56
3.4. Завершение программы	57
3.5. Организация используемой входной информации	58
3.6. Организация используемой выходной информации	58

4.	Сообщения.....	59
5.	Регистрация изменений.....	60

1. Назначение и условия применения программы

1.1. Назначение программы

Программа MiniSCADA Spica имеет следующее назначение:

- организация доступа к данным контроллеров и устройствам по протоколам Modbus RTU (с использованием интерфейса RS-232 или RS-485), Modbus TCP (с использованием интерфейса (сокета) TCP/IP, например, на базе Ethernet);
- предоставление информационных данных в реальном масштабе времени в различных видах – цифровом, графическом и т.п.;
- коррекция способов отображения информации (наличие встроенного интерпретатора настроечного файла);
- математическая обработка данных для их интерпретации;
- передача управления на нижестоящее контроллерное оборудование.

1.2. Функции, выполняемые программой

Программа MiniSCADA Spica предоставляет следующие возможности:

- функции HMI (человеко-машинный интерфейс) - отображение информации на экране монитора в удобной и понятной форме;
- контроль и управление - возможность оперативного вмешательства со стороны персонала;
- архивирование данных и перенос архивных файлов на резервный носитель;
- разграничение доступа к функциям посредством аутентификации (ввода пароля);
- гибкие решения для быстрого построения экранных форм, в том числе на основе макросов и шаблонов.

1.3. Условия, необходимые для выполнения программы

1.3.1 Требования к составу и параметрам периферийных устройств

Для соединения с контроллером необходимо наличие свободного порта COM или Ethernet.

Взаимодействие с программой осуществляется с помощью мыши или стилуса, в случае, если программа установлена на панельный компьютер.

Для написания конфигурации программы необходима клавиатура.

1.3.2 Требования к программному обеспечению

Функционирование программы MiniSCADA Spica осуществляется в среде операционных систем Windows CE, Windows XP, Windows XP Embedded, Windows 7, Windows Embedded Standart 7, Windows 10, Windows 10 IoT и в других совместимых операционных системах.

1.3.3 Требования к пользователю

Пользователь программы MiniSCADA Spica должен обладать следующими знаниями и навыками:

- иметь понятие о файлах и их спецификации;
- понимать принципы размещения файлов на накопителях и регистрации их в каталогах (папках);
- представлять иерархическую структуру папок (каталогов);
- иметь знание и умение представления чисел в различных системах исчисления;
- знать главные элементы графического интерфейса Windows;
- пользоваться оконным интерфейсом, владеть работой с окнами;
- осуществлять операции с текстовыми документами в текстовых редакторах;
- проводить операции с файлами всех типов (копирование, перемещение удаление, поиск по имени или содержимому и т.д.);
- осуществлять различными способами запуск программ;
- применять эргономические стандарты при разработке графического пользовательского интерфейса;
- знать протокол Modbus (характеристики, свойства и т.п.);
- иметь понятие о работе СОМ-порта, знать характеристики и основные настройки;
- иметь понимание принципов соединения устройств (компьютера с контроллером).

2. Характеристика программы

2.1. Описание основных характеристик программы

2.1.1 Временные характеристики

При передаче физических сигналов и во время их цифровой обработки неизбежно возникают задержки, обусловленные затратой необходимого времени на передачу сигналов и цифровых данных. Задержки происходят на всех уровнях обработки, но на некоторых из них являются наиболее заметными.

В программе **MiniSCADA Spica** основная причина задержки отображения данных – конечное время на передачу данных по каналу связи между контроллером и программой по протоколу Modbus, которое характеризуется т.н. временем цикла сканирования.

Время цикла сканирования в основном зависит от качества линии связи (отсутствия помех в линии связи), количества опрашиваемых регистров и наличия задержки ответа контроллера.

Передача команд управления (от программы к контроллеру) осуществляется на приоритетной основе без привязки к циклу сканирования.

2.2. Описание основных особенностей программы

2.2.1 Версии программы для различных операционных систем

Существует две версии программы MiniSCADA Spica:

- **MiniSCADA Spica (CE)** предназначена для работы под операционной системой Windows-CE. Программа этой версии состоит из исполняемого модуля **Pult.exe** и файла инициализации **Pult.ini**.
- **MiniSCADA Spica (XP)** предназначена для работы под операционными системами Windows-XP, Windows-XP-Embedded и под другими совместимыми операционными системами. Программа этой версии состоит из исполняемого модуля **Pult_xp.exe** и файла инициализации **Pult_xp.ini**.

Программа **MiniSCADA Spica (XP)** предоставляет расширенные возможности при разработке и отладке, поэтому при разработке конфигурации проекта удобнее использовать данную версию программы. Конфигурация одинакова для обеих версий программы.

2.2.2 Экранные формы

Экранные формы являются основным средством взаимодействия оператора технологического процесса и системы автоматизации.

На экранных формах наглядно в виде мнемосхем, таблиц и пр. отображается работа контролируемых технологических объектов. Изображения на экранных формах динамические, то есть всегда соответствуют актуальным (текущим) данным и результатам их алгоритмической обработки.

Данные экранных форм и их элементов, параметры связи с управляющими контроллерами, пароли доступа и прочий функционал проекта описываются в текстовом виде в двух файлах формата *.txt: Pult_hrd.txt и Pult_flx.txt. Используемые изображения хранятся в виде файлов формата *.bmp.

Конфигурация проекта состоит из файлов конфигурации Pult_hrd.txt и Pult_flx.txt и файлов используемых изображений. Конфигурация одинакова для обеих версий программы.

2.2.3 Протокол Modbus

Программа **MiniSCADA Spica** связывается с контроллером по протоколам:

- Modbus RTU с использованием интерфейса RS-232 или RS-485;
- Modbus TCP с использованием интерфейса (сокета) TCP/IP (например, на базе Ethernet).

Modbus работает с четырьмя типами буферов памяти:

- 0x – входной логический буфер контроллера;

- 1x – выходной логический буфер контроллера;
- 3x – выходной целый буфер контроллера;
- 4x – входной целый буфер контроллера.

Программа MiniSCADA Spica является master-устройством. Программа организует цикл опроса. Цикл начинается с младшего адреса буфера 1x. Далее опрашивает последовательно 3x, 0x и 4x. Программа опрашивает только те ячейки буфера, которые задействованы в проекте. Ячейки группируются в пакеты. В пакете ячейки идут строго подряд. Если существует пропуск ячеек (какие-то из ячеек буфера не задействованы), для передачи следующего набора ячеек будет сформирован следующий пакет.

Поэтому, для оптимизации работы программы, рекомендуется следить, чтобы адреса используемых переменных шли подряд.

2.2.4 Основные понятия

Фрейм – область на форме, которая характеризуется типом, положением, размером, условиями видимости, и другими параметрами, в числе которых действие, выполняющееся при нажатии на фрейм. Кроме того, для каждого фрейма определяется маска доступа к нему. Фрейм является базовой составляющей мнемосхем.

Шаблон – именованный набор фреймов, который предполагается использовать многократно. Может состоять из одного или нескольких фреймов.

Переменная – именованная изменяемая величина, содержащая значение. Переменная может быть связана с ячейками памяти буфера контроллера (входного и выходного).

Действие – именованная структура, представляющая собой *Условие* и два события: *Событие при выполнении условия* и *Событие при невыполнении условия*.

Контент – графический примитив (цвет, растровое изображение (bmp) или текст). Каждый компонент контента именован.

Последовательность – именованный список, из которого по определенному правилу выбирается компонент контента для вывода на экран или записи в файл. Используется при настройке бокса.

Бокс – именованный набор характеристик: фон, текст, выравнивание и параметры рамки. Используется при настройке фрейма. В качестве фона, текста, цвета текста и рамки могут быть заданы последовательности.

Задача – именованный перечень действий, выполняемых периодически в процессе работы программы.

Шрифт – именованный набор настроек начертания символов.

3. Обращение к программе

3.1. Установка программы

Для установки программы MiniScada Spica на компьютер программиста (при отсутствии проекта) необходимо скопировать исполняемый модуль и файл инициализации на компьютер (**Pult.exe** и **Pult.ini** для **Windows-CE**; **Pult_xp.exe** и **Pult_xp.ini** для **Windows-XP**).

В случае необходимости переноса на компьютер проекта (например, установки программы на компьютер оператора технологического процесса), необходимо скопировать на компьютер исполняемый модуль, файл инициализации, файлы конфигурации **Pult_hrd.txt** и **Pult_flx.txt**, а так же **bmp**-файлы изображений, используемых в проекте.

Исполняемый модуль и файл инициализации необходимо поместить в один каталог жесткого диска компьютера. Конфигурацию проекта (файлы конфигурации и изображения) можно поместить в любой каталог жесткого диска компьютера, путь к которому необходимо указать в файле инициализации.

3.2. Загрузка и запуск программы

Для запуска программы MiniSCADA Spica необходимо:

- 1) убедиться, что в файле инициализации (**Pult.ini** для **Windows-CE**; и **Pult_xp.ini** для **Windows-XP**) указаны необходимые параметры (см. формат файла инициализации);
- 2) запустить приложение (**Pult.exe** для **Windows-CE** или **Pult_xp.exe** для **Windows-XP**) стандартными средствами ОС **Windows**.

3.3. Выполнение программы

3.3.1 Создание конфигурации проекта

3.3.1.1 Общие сведения

Конфигурация проекта считывается программой MiniScada Spica из двух текстовых файлов **Pult_hrd.txt** и **Pult_flx.txt**, размещенных в определенном каталоге жесткого диска компьютера.

Файл **Pult_hrd.txt** содержит всю конфигурацию проекта.

Файл **Pult_flx.txt** содержит часть конфигурации, которая может измениться в процессе работы программы и перезаписывается самой программой (например, некоторые строки секции

[limit]). Если изменяемый средствами программы параметр не включен в файл Pult_flx.txt, он будет включен в файл автоматически.

Во время запуска программа читает файлы конфигурации и в соответствии с их содержимым создает необходимые структуры данных в памяти компьютера.

Программа устанавливает **связь с** одним или несколькими **контроллерами** (PCU), указанным в файле (секция [pcu], п. "Добавление PCU") по прописанным в файле адресам (секция [port], п. "Настройка каналов связи") и осуществляет обмен данными.

Программа осуществляет обмен данными с контроллером посредством **переменных** (секция [var], п. "Создание переменных"). Для переменной могут быть указаны адреса (ячейки буферов контроллера) для чтения и записи ее значения. Кроме того, для переменной указывается тип преобразования значения, полученного из контроллера, для вывода на экран. Если для переменной используется линейное преобразование, необходимо указать **пределы значений** переменной (секция [limit], п. "Задание пределов значений переменных").

Значения переменных могут отображаться в окне в виде цифр (например, в качестве значения одного из показателей технологического процесса), либо влиять на отображение **фреймов** (например, при различных значениях переменной на фрейме может отображаться различный текст).

Цвета, изображения и тексты, используемые при отображении фрейма, задаются соответственно в секциях [color], [bmp] и [str].

Для задания различных вариантов отображения фона, цвета рамки и текста, а также текста, выводимого на фрейме, служат **последовательности** (секция [seq], п. "Создание последовательностей"). Они ставят в соответствие возможным значениям переменных параметры вывода фрейма на экран.

Фон, цвет рамки и текста, текст, выводимый на фрейме, выравнивание текста и толщина рамки для многих фреймов могут совпадать, поэтому эти признаки задаются специальным элементом конфигурации – **боксом** (секция [rect], п. "Создание боксов"). Фон, цвет рамки и текста и текст, выводимый на фрейме, могут быть как статичными, так и изменяющимися в зависимости от значения переменной, указанной при создании фрейма. В этом случае в качестве их значения указывается **последовательность** (секция [seq], п. "Создание последовательностей").

Для фрейма может быть указано **действие**, которое будет выполнено при нажатии на фрейм. Список действий содержится в секции [actions] (п. "Создание действия").

Для фрейма, для которого указано действие при нажатии на фрейм, может быть указана маска доступа. Тогда для выполнения действия при нажатии будет необходимо наличие у оператора соответствующего **флага доступа** и **пароля доступа**. Флаги и пароли доступа задаются в секции [pw] (п. "Настройка паролей доступа").

Группу фреймов, которая с различными значениями некоторых настроек используется в проекте многократно, можно объединить и использовать как единое целое, создав **шаблон** (секция [stamp], п. "Создание шаблонов"). Описание шаблона – это список фреймов, с координатами, заданными относительно точки привязки шаблона (например, $x=0$, $y=0$). Изменяемые настройки фреймов задаются величинами, значения которых для каждого применения шаблона будут свои. Имена таких величин в настройках фрейма пишутся в круглых скобках, значения величин задаются в поле «Параметры» шаблона при его использовании на форме.

Проект может состоять из нескольких диалоговых окон. В окнах отображаются фреймы и шаблоны с заданными параметрами. Фреймы и шаблоны **основного окна** программы описывается в секции [form] (п. "Конфигурирование основного окна проекта"). Другие окна описываются в секции [dialog] (п. "Создание диалоговых окон").

Для записи десятичных дробей в качестве разделителя дробной и целой части используется символ «точка».

Файлы конфигурации состоят из секций:

- [port];
- [modbus];
- [pcu];
- [pw];
- [color];
- [bmp];
- [str];
- [seq];
- [rect];
- [font];
- [action];
- [task];
- [dialog];
- [stamp];
- [form];
- [start];
- [limit];
- [var];
- [arc];
- [alarm];
- [time];
- [print];
- [numpad].

Секции должны следовать в определённом порядке.

Файл Pult_flx.txt может не содержать ни одной секции. При внесении изменений в параметры конфигурации средствами программы MiniSCADA Spica изменяемые параметры будут добавлены в файл. При отсутствии файла он будет создан автоматически.

Файл Pult_hrd.txt может не содержать некоторых секций, таких как [color], [str], [bmp], [stamp], [dialog], [task], [print], [time], [arc], [alarm], [numpad] в случае, если функционал, задаваемый этими секциями, в проекте не используется.

3.3.1.2 *Формат файлов конфигурации*

Файлы конфигурации текстовые, имеют формат *.txt с кодировкой Windows-1251.

Строчный разделитель – последовательность 0x0D 0x0A.

Файлы конфигурации состоят из секций, каждая секция имеет свой формат данных.

Разделителями секций являются строки с названием секции. Название секции заключено в квадратные скобки. В строке названия секции никакие дополнительные символы не допустимы.

Каждая секция состоит из строк. Каждая строка состоит из полей. Разделителями между полями является символ '|'. Поле может состоять из нескольких подполей, разделителями между которыми является символ ';'. Некоторые, предусмотренные форматом, поля могут содержать пустую строку или прочерк '-'. В таком случае значение этого поля устанавливается из правил подстановки по умолчанию.

Секция может быть разделена на подсекции. Разделителем подсекций является строка, первым полем которой является символ '\$'. Такой секцией является, например, секция [stamp].

При описании элементов файла конфигурации, значение поля **«Наименование»** должно быть уникально. Поле «Наименование» может содержать:

- символы латинского алфавита ('a'...'z', 'A'...'Z');
- цифры ('0'...'9');
- символ '_'.

Первым символом поля **«Наименование»** должна быть буква.

Дробные числа (например, в секции [limit]) записываются в формате с плавающей точкой.

Разделитель целой и дробной части – точка (символ '.').

В тексте файлов конфигурации, для удобства чтения и редактирования допускаются комментарии.

Каждый комментарий должен занимать одну строку и начинаться с символов '//' (двойной слеш).

Допускается также пропуск строк.

3.3.1.3 Секция [port]. Настройка каналов связи

Для каналов связи (портов), используемых программой MiniSCADA Spica для связи с контроллерами, в конфигурации задаются различные настройки в зависимости от типа порта. Настройки связи с контроллером прописываются в секции [port].

Для связи с контроллером по интерфейсу **RS-232 (RS-485)** в конфигурации программы MiniSCADA Spica задается уникальное наименование порта, скорость передачи, проверка четности.

Формат строки секции [port] для интерфейса RS-232 (RS-485):

№	Наименование поля	Примечание
1	Наименование порта	Строка
2	Скорость передачи	Десятичное целое число
3	Проверка четности	Один из символов: е – контроль четности, о – контроль нечетности, п – нет контроля четности.

Поле 1 формируется, как обозначение типа порта COM с указанием номера порта в операционной системе (например, COM1).

Для связи с контроллером по интерфейсу **Ethernet** в конфигурации программы MiniSCADA Spica задается уникальное наименование порта и IP адрес контроллера.

Формат строки секции [port] для интерфейса Ethernet:

№	Наименование поля	Примечание
1	Наименование порта	Строка
2	Адрес IP	Строка

Поле 1 формируется, как обозначение типа порта IP с добавлением произвольного цифро-буквенного сочетания в целях создания уникального имени порта.

Поле 2 содержит строку – IP-адрес для связи с контроллером.

Пример секции [port]:

```
[port]
COM1|19200|n
COM5|38400|n
IP1|192.190.228.215
IP2|192.190.228.222
```

3.3.1.4 Секция [modbus]. Настройка протокола Modbus

Для протокола Modbus в конфигурации программы MiniSCADA Spica задается время ожидания. Эта настройка прописывается в секции [modbus]. Секция [modbus] содержит одну строку.

Формат строки секции [modbus]:

№	Наименование поля	Примечание
1	Время ожидания (мс)	Десятичное целое число

Пример секции [modbus]:

```
[modbus]  
1000
```

3.3.1.5 Секция [rsu]. Добавление PCU

Программа может осуществлять обмен данными с одним или несколькими контроллерами (PCU).

Для каждого PCU указывается наименование PCU, адрес в сети Modbus и наименование порта.

№	Наименование поля	Примечание
1	Наименование PCU	Строка
2	Адрес в сети Modbus	Целое число
3	Имя порта (канала связи)	Строка

Поле 1 содержит наименование PCU.

Поле 2 содержит адрес PCU в сети Modbus.

Поле 3 содержит наименование порта (секция [port]).

Примечание: Наличие описания PCU означает автоматическое создание соответствующей переменной с именем **PCU_LINE**, значение которой программа устанавливает в соответствии с состоянием связи:

- PCU_LINE = 1, если есть связь с PCU;
- PCU_LINE = 0, если нет связи с PCU.

Пример секции [rsu]:

```
[rsu]
PCU1 | 3 | IP1
PCU2 | 5 | COM5
```

3.3.1.6 Секция [pw]. Настройка паролей доступа

Для разграничения доступа к функционалу проекта в программе MiniSCADA Spica предусмотрено наличие паролей доступа. Пароли доступа задаются в секции [pw].

Формат строки секции [pw]:

№	Наименование поля	Примечание
1	Номер флага доступа	Целое число 1..8 или 99
2	Время действия (с)	Целое число 0..65535
3	Пароль в закодированном виде	Строка

Поле 1 содержит номер флага доступа. К флагу доступа применяется маска доступа из соответствующего поля секции [form].

Флаг 99 дает право доступа ко всем фреймам проекта, не зависимо от того, какую маску доступа имеет фрейм.

Поле 2 содержит время в секундах, в течение которого флаг доступа остается актуальным без признаков активности пользователя.

Поле 3 содержит пароль в закодированном виде. Данная строка формируется специальной программой «Генератор паролей» (файл Pult_PW.exe). Средствами этой программы введенный в строку «Фраза» пароль преобразуется в код. При работе с программой, при запросе пароля, необходимо ввести пароль в исходном (не закодированном) виде.

Если предполагается работа с проектом на компьютере, не оснащенном клавиатурой, в качестве пароля могут быть использованы только цифры.

Пример секции [pw]:

[pw]

1|300|690E6C9BD1

3.3.1.7 Секция [color]. Создание цветов

Цвета, используемые в проекте, задаются в секции [color].

Формат строки секции [color]:

№	Наименование поля	Примечание
1	Наименование контента	Строка
2	Цветовой код	Десятичное целое число

Цветовой код вычисляется по формуле: $B*65536+G*256+R$, где B, G, R – значения соответствующих цветовых компонент (синей, зеленой и красной), целые числа от 0 до 255.

Цвета, существующие по умолчанию:

red	Красный цвет
green	Зеленый цвет
blue	Синий цвет
white	Белый цвет
silver	Светло-серый цвет
black	Черный цвет

Цвета, задаваемые в секции [color] должны иметь наименования, отличные от имен существующих по умолчанию цветов.

Пример секции [color]:

```
[color]
yellow|65535
grey|8421504
violet|16711935
```

3.3.1.8 Секция [bmp]. Добавление изображений

В проекте MiniSCADA Spica в качестве фона фрейма может быть использовано изображение. Изображения должны быть в формате *.bmp. Для удобства переноса конфигурации проекта, рекомендуется все используемые в проекте изображения перед добавлением в проект поместить в одну папку, тогда путь к каталогу не указывается. Изображения, используемые в проекте, задаются в секции [bmp].

Формат строки секции [bmp]:

№	Наименование поля	Примечание
1	Наименование контента	Строка
2	Имя файла	Строка Содержит полное имя файла (с указанием каталога)

Пример секции [bmp]:

```
[bmp]
pic1|pic1.bmp
pic2|pic2.bmp
pic3|pic3.bmp
lamp_x|\HardDisk\lamp_x.bmp
lamp_red|\HardDisk\lamp_red.bmp
lamp_green|\HardDisk\lamp_green.bmp
```

3.3.1.9 Секция [str]. Создание строк

Строки, используемые в проекте, задаются в секции [str].

Формат строки секции [str]:

№	Наименование поля	Примечание
1	Наименование контента	Строка
2	Текст	Строка

При необходимости для обозначения места разрыва строки используется сочетание символов '\\\\' (три обратных слеша подряд).

Пример секции [str]:

```
[str]
s0|XXX
sx|X
s1|НЕОПР.
s2|ДИСТ.
s3|АВТО.
s4|МЕСТ.
```

3.3.1.10 Секция [seq]. Создание последовательностей

Последовательность используется при настройке бокса, а бокс – при настройке фрейма.

Последовательность служит для задания различных вариантов отображения фона, цвета рамки и текста, или текста, выводимого на фрейме. Последовательность ставит в соответствие возможным значениям переменной, указанной в настройках фрейма, параметр вывода фрейма на экран – определенный компонент контента (изображение, цвет или строку).

Последовательности задаются в секции [seq].

Формат строки секции [seq]:

№	Наименование поля	Примечание
1	Наименование последовательности	Строка
2	Имя контента	Строка
3	Имя контента	Строка
4	...	Строка

Первый контент соответствует состоянию «недостоверность», когда значение переменной не определено (например, когда отсутствует связь с контроллером).

Последующие контенты соответствуют значениям переменной от нуля и выше (0, 1, 2...). Для аналоговых переменных учитывается значение в формате данных контроллера.

Последний контент соответствует всем значениям переменной, превышающим количество контентов в последовательности минус 3. Например, если в последовательности 4 контента, первый будет соответствовать состоянию «недостоверность», второй – значению 0, третий – значению 1, а четвертый – всем прочим возможным значениям переменной.

Пример секции [seq]:

```
[seq]
clap_alarm_textclr|white|silver|red
clap_alarm_text|s0|s8|s14
clap_alarm_box_textclr|white|silver|red
clap_alarm_box_text|s0|s8|s9
PUMPMODE|s0|s1|s2|s3|s1
```

3.3.1.11 Секция [rect]. Создание боксов

Бокс представляет собой набор настроек отображения фреймов на экране: фон (цвет или изображение), цвет текста и рамки, текст, выводимый на фрейме, выравнивание текста на фрейме и толщина рамки. Боксы задаются в секции [rect].

Формат строки секции [rect]:

№	Наименование поля	Примечание
1	Наименование бокса	Строка
2	Фоновые контенты	Строка
3	Цветовые контенты	Строка
4	Текстовые контенты	Строка
5	Выравнивание	Строка
6	Толщина рамки	Десятичное целое число 0 – означает, что рамка отсутствует

Поле 1 содержит имя бокса.

Поле 2 содержит имя последовательности или компонента контента, используемого в качестве фонового цвета или изображения. Если фон не задан, он будет прозрачным.

Поле 3 содержит имя последовательности или компонента контента, используемого в качестве цвета текста и рамки, если она есть.

Поле 4 содержит имя последовательности или компонента контента, используемого в качестве текста.

Поле 5 содержит обозначение принципа выравнивания текста. Значение имеет первая буква обозначения:

- l – по левому краю;
- r – по правому краю;
- любая другая буква (по умолчанию) – по центру.

Пример секции [rect]:

```
[rect]
pic1_rect|pic1_bckgrnd|-|-|0
pic2_rect|pic2_bckgrnd|-|-|0
pic3_rect|pic3_bckgrnd|-|-|0
analog_rect_d|empty_bckgrnd|black|an_text|
analog_descr_rect|empty_bckgrnd|black|an_descr|
analog_units_rect|empty_bckgrnd|black|an_units|left
```

3.3.1.12 Секция [font]. Создание шрифтов

В проекте MiniSCADA Spica можно использовать различные настройки начертания символов для вывода строк на экран. Настройки начертания задаются специальными элементами конфигурации – шрифтами. Шрифты и их параметры задаются в секции [font].

Формат строки секции [font]:

№	Наименование поля	Примечание
1	Наименование шрифта	Строка
2	Системное имя шрифта	Строка
3	Ширина в точках	Десятичное целое число
4	Высота в точках	Десятичное целое число
5	Курсив	0 или 1
6	Жирность	0 или 1

Пример секции [font]:

```
[font]
font2|tahoma|06|14|0|1
font4|arial|06|14|0|0
font1|tahoma|07|14|0|1
font3|system|06|12|0|1
```

3.3.1.13 Секция [action]. Создание действия

Действия, используемые в проекте, задаются в секции [action].

Формат строки секции [action]:

№	Наименование поля	Примечание
1	Наименование действия	Строка
2	Условие	Строка
3	Событие при выполнении условия	Строка
4	Событие при невыполнении условия	Строка

Поле 1 содержит имя действия.

Поле 2 содержит условие. Если поле пропущено, условие считается истинным.

Формат записи условий:

$V=D$ - условие выполняется, если значение переменной V равно D

$V>D$ - условие выполняется, если значение переменной V больше D

$V<D$ - условие выполняется, если значение переменной V меньше D

$V!D$ - условие выполняется, если значение переменной V не равно D

перечисление нескольких условий, разделённых знаком «точка с запятой», означает проверку одновременного выполнения всех перечисленных условий (то есть условия складываются по «и»)

Например:

$V1<A;V2>B$ - условие выполняется, если значение переменной $V1$ меньше A **и** значение переменной $V2$ больше B

Поле 3 содержит событие, которое будет совершено при выполнении условия.

Поле 4 содержит событие, которое будет совершено при невыполнении условия.

Поля 3 и 4 могут подразделяться на дополнительные поля, тогда выполнение операций производится последовательно, в указанном порядке. Разделителем таких полей является символ ';':

В полях 3 и 4 допустимо указывать:

- операции archive, save, load, execute, print, arcsave, arcres (п. "Список зарезервированных обозначений");
- операции присваивания.

№	Тип присваивания	Пример	Результат
1	Непосредственное присваивание кода переменной	var=5	Код переменной var будет равен 5

№	Тип присваивания	Пример	Результат
2	Присваивание кода одной переменной от другой переменной	<code>var1=var2</code>	Код переменной var1 будет равен коду переменной var2
3	Присваивание кода переменной с арифметическими операциями	<code>var1=var2+1</code>	Код переменной var1 будет равен коду переменной var2 плюс 1
4	Установка признака «недостоверность» переменной	<code>var1#</code>	Код переменной var1 останется без изменения, переменная var1 становится недостоверной
5	Сброс признака «недостоверность» переменной	<code>var=</code>	Код переменной var1 останется без изменения, переменная var1 становится достоверной
6	Присваивание строкового контента	<code>str=произвольный текст</code>	Текст контента str будет текстом после знака равенства: «произвольный текст»
7	Присваивание строкового контента из последовательности по значению переменной	<code>str=var@seq</code>	Текст контента str становится таким же, как у контента последовательности seq, порядковый номер которого определяет код переменной var. Если переменной или последовательности с такими именами не существует, то текст контента str будет текстом после знака равенства: «var@seq»

Примечание 1: Значение физической величины (пункты 1-5) в результате присваивания кода переменной пересчитывается в соответствии с пределами из секции [limit].

Примечание 2: Под признаком «недостоверность» подразумевается, что значение переменной воспринимается как неопределенное, полученное в период неисправности измерительного прибора или канала связи. Признак недостоверности учитывается при создании последовательностей.

Пример секции [action]:

```
[action]
begin|-|n=1
Clap_avto|-|U(X)A=1
Clap_dist|-|U(X)D=1
Clap_mest|-|U(X)M=1
Clap_kvrit|-|U(X)K=1
//закрытие диалоговых окон при останове
close|PTB_SOST=4|cancel|
//отображение диалога управления заслонками
zasl_cmd|PTB_SOST!2|Zasl_Dialog
```

3.3.1.14 Секция [task]. Задание периодических действий

Последовательность действий, выполняемых периодически во время работы программы, задается в секции [task].

Строка описания задачи имеет следующий формат:

№	Наименование поля	Примечание
1	Период (мс)	Целое число
2	Наименование действия	Строка
3	Параметры	Строка

Поле 1 содержит период в миллисекундах, с которым выполняется действие, указанное в поле 2.

Поле 2 содержит имя действия из секции [action].

Поле 3 содержит параметры действия, указанного в поле 2.

Пример секции [task]:

```
[task]  
3000|action_time|time=50
```

3.3.1.15 Секция [dialog]. Создание диалоговых окон

Проект MiniSCADA Spica может быть многооконным. Основное окно программы описывается в секции [form]. Другие диалоговые окна проекта задаются в секции [dialog].

Для описания каждого диалогового окна секция [dialog] делится на подсекции. Первая строка подсекции содержит параметры отображения диалогового окна. Последующие строки подсекции содержат описание элементов (фреймов и шаблонов), присутствующих в диалоговом окне (аналогично секции [form], п. "Конфигурация основного окна проекта")

Формат *первой строки подсекции* – описания диалога:

№	Наименование поля	Примечание
1	Символ разделителя подсекции	Символ '\$'
2	Наименование диалога	Строка
3	Координата левой точки	Десятичное целое число
4	Координата верхней точки	Десятичное целое число
5	Ширина	Десятичное целое число
6	Высота	Десятичное целое число
7	Название диалога	Строка
8	Действие при вызове диалога	Строка
9	Параметры действия при вызове	Строка
10	Флаги	Целое число

Поле 1 содержит символ разделителя подсекции.

Поле 2 содержит наименование диалога – его уникальное имя.

Поля 3 и 4 содержат координаты левой верхней точки слева и сверху соответственно.

Поля 5 и 6 содержат размер диалогового окна – ширину и высоту соответственно.

Поле 7 содержит название диалога, которое будет отображаться в его заголовке.

Поле 8 содержит действие при вызове диалога.

Поле 9 содержит строку, содержащую список подстановок для действия из поля 8.

Поле 10 содержит специальные флаги, задающие особенности отображения. Поле не обязательно (по умолчанию 0), представляет собой поле битов:

- бит 1 (младший двоичный разряд) – запрет кнопки закрытия окна в верхнем правом углу;
- бит 2 и выше – резерв.

Следующие строки подсекции содержат описание фреймов и шаблонов в том же виде, что и в секции [form].

Пример секции [dialog]:

```
[dialog]
$|call_oper|260|170|230|090|Упр-е сиреной вызова оператора||
```

```
000000|000|000|250|180|-|-|-|backgrnd_rect|-|-|-|
//управление
000008|060|007|100|020|-|-|-|bbtn_rect|-|-|-|Включить|Com_siren_on|
000008|060|032|100|020|-|-|-|bbtn_rect|-|-|-|Отключить|Com_siren_off|
$|Кларан_Diag|260|070|300|500|Управление клапаном||
000001|000|378|300|014|-|-|-|label_c_rect|-|-|-|коэфф. регулирования:
000001|000|398|055|020|-|-|-|label_r_rect|-|-|-|KP
analog_TR|060|398|frame=(frame);analog_TR=Z(clap_name)KP
000001|000|423|055|020|-|-|-|label_r_rect|-|-|-|TI
analog_TR|060|423|frame=(frame);analog_TR=Z(clap_name)TI
000001|000|448|055|020|-|-|-|label_r_rect|-|-|-|TD
analog_TR|060|448|frame=(frame);analog_TR=Z(clap_name)TD
000001|150|398|025|020|-|-|-|label_r_rect|-|-|-|TZ
analog_TR|180|398|frame=(frame);analog_TR=Z(clap_name)TZ
000001|150|423|025|020|-|-|-|label_r_rect|-|-|-|ZN
analog_TR|180|423|frame=(frame);analog_TR=Z(clap_name)ZN
000001|150|448|025|020|-|-|-|label_r_rect|-|-|-|PF
analog_TR|180|448|frame=(frame);analog_TR=Z(clap_name)PF
```

3.3.1.16 Секция [stamp]. Создание шаблонов

Шаблон объединяет в своем составе группу фреймов, которая с заданием различных значений некоторых настроек используется в проекте многократно. Изменяемые настройки фреймов задаются величинами, значения которых для каждого применения шаблона будут свои. Имена таких величин в настройках фрейма пишутся в круглых скобках, значения величин задаются в поле «Параметры» шаблона при его использовании на форме.

Список шаблонов проекта задается в секции [stamp]. Для описания каждого шаблона секция [stamp] делится на подсекции.

Формат первой строки подсекции – описания шаблона:

№	Наименование поля	Примечание
1	Символ разделителя подсекции	Символ '\$'
2	Наименование шаблона	Строка

Следующие строки подсекции содержат описание фреймов в том же виде, что и в секции [form] (п. "Конфигурация основного окна проекта"), но координаты левой верхней точки указываются относительно точки привязки (обычно x=0; y=0), и в качестве настроек фрейма могут быть использованы значения подстановок из списка параметров. Чтобы использовать значение подстановки надо в качестве настройки указать имя подстановки в круглых скобках.

Пример секции [stamp]:

```
[stamp]
$|main_time
001|000|000|014|020|-|-|-|main_time_rect|-|-|(main_time_value)
$|analog_H2S
001|000|000|065|020|n=(frame)|-|-|-|analog_rect|-|dev_Ok|I(an_H2S)PV|
|Analog_H2S_Diag|Analog_name=(an_H2S);frame=(frame)
001|000|000|011|010|n=(frame)|-|-|-|HI_yellow_lim_rect|S(an_H2S)H|-|-|
001|000|000|011|010|n=(frame)|-|-|-|HH_lim_rect|S(an_H2S)HH|-|-|
001|000|000|011|020|n=(frame)|-|-|-|ERR_rect|S(an_H2S)E|-|-|
```

3.3.1.17 Секция [form]. Конфигурация основного окна проекта

Основное окно проекта описывается в секции [form]. Окно содержит фреймы и шаблоны. Формат строки для фреймов и шаблонов различен.

Последовательность описания элементов определяет порядок вывода их на экран. Это значит, что элементы, добавленные ранее, будут перекрыты более поздними элементами.

Строка описания фрейма имеет следующий формат:

№	Наименование поля	Примечание
1	Тип фрейма	Десятичное целое число
2	Координата левой точки	Десятичное целое число
3	Координата верхней точки	Десятичное целое число
4	Ширина	Десятичное целое число
5	Высота	Десятичное целое число
6	Условие видимости	Выражение условия
7	Условие выделения	Выражение условия
8	Шрифт нормального состояния	Строка
9	Шрифт выделенного состояния	Строка
10	Бокс	Строка
11	Переменная фона	Строка
12	Переменная цвета	Строка
13	Переменная текста	Строка
14	Статический текст	Строка
15	Действие при нажатии	Строка
16	Параметры действия при нажатии	Строка
17	Маска аутентификации	Десятичное целое число 1..255, по умолчанию =255

Поле 1 содержит тип фрейма, указывающий на особенности при его отображении.

При выводе на экран фрейма с типом 0 (фон) из области рисования извлекаются прямоугольники, в которых будут выведены другие фреймы. Это позволяет исключить эффект мерцания.

При выводе на экран изображения фрейма с типом 8 (кнопка) применяется «эффект объемности»: кнопка в нажатом состоянии изображается «утопленной», в не нажатом – «выпуклой».

Тип фрейма – десятичное целое число, определяющее назначение фрейма:

- 0 – фрейм используется как фон, на котором находятся другие фреймы.
- 1 – обычный фрейм.
- 5 – сводка тревог.
- 8 – кнопка.

Поля 2 и 3 содержат координаты левой верхней точки фрейма – расстояние от левого верхнего угла диалогового окна слева и сверху соответственно.

Поля 4 и 5 содержат соответственно ширину и высоту фрейма в точках.

Поле 6 содержит условие, при выполнении которого данный фрейм отображается на экране. Иначе фрейм не отображается.

Поле 7 содержит условие, при выполнении которого данный фрейм отображается как выделенный. Иначе фрейм отображается как обычный.

Формат записи условий:

$V=D$ - условие выполняется, если значение переменной V равно D

$V>D$ - условие выполняется, если значение переменной V больше D

$V<D$ - условие выполняется, если значение переменной V меньше D

$V!D$ - условие выполняется, если значение переменной V не равно D

перечисление нескольких условий, разделённых знаком «точка с запятой», означает проверку одновременного выполнения всех перечисленных условий (то есть условия складываются по «и»)

Например:

$V1<A;V2>B$ - условие выполняется, если значение переменной $V1$ меньше A и значение переменной $V2$ больше B

Поле 8 содержит шрифт, описанный в секции [font], который используется для вывода текста фрейма на экран, если не выполняется условие выделения.

Поле 9 содержит шрифт, описанный в секции [font], который используется для вывода текста фрейма на экран, если выполняется условие выделения.

Поле 10 содержит имя бокса, описанного в секции [rect], определяет все возможные варианты вывода на экран данного фрейма.

Поле 11 содержит имя переменной, значение которой определяет подстановку фонового контента при выводе фрейма на экран.

Поле 12 содержит имя переменной, значение которой определяет подстановку цветового контента при выводе фрейма на экран.

Поле 13 содержит имя переменной, значение которой определяет подстановку текстового контента при выводе фрейма на экран.

Поле 14 содержит текст, используемый для вывода на экран в случае, если не указан текстовый контент.

Поле 15 содержит строку, содержащую последовательность действий, выполняемых программой при нажатии на фрейм.

Поле 16 содержит строку, содержащую список подстановок для действия при нажатии.

Поле 17 содержит МД – число, представляющее собой поле битов (флагов), служащих для проверки возможности выполнения действия при нажатии в соответствии с разграничением доступа по паролям.

Действие выполняется, если:

- МД равно значению по умолчанию 255;
- результат операции ФД *and* МД отличен от нуля.

(где МД – маска доступа; ФД – флаг доступа; *and* – логическое побитовое «и»)

В противном случае – на экран выводится предложение ввести правильный пароль и, таким образом, повторить проверку, установив новое значение ФД.

Строка ссылки на шаблон имеет следующий формат:

№	Наименование поля	Примечание
1	Наименование шаблона	Строка
2	Координата левой точки	Десятичное целое число
3	Координата верхней точки	Десятичное целое число
4	Параметры	Строка

Поле 1 содержит имя используемого шаблона из секции [stamp].

Поле 2 содержит горизонтальную координату точки привязки фрейма.

Поле 3 содержит вертикальную координату точки привязки фрейма.

Поле 4 содержит список подстановок для формирования строк.

Пример секции [form]:

```
[form]
000001|209|382|040|016|n=1|-|font4|-|label_rect|-|-|Q тек.,м3/ч
000001|209|346|040|016|n=1|-|font4|-|label_rect|-|-|обводн.,%
000001|209|542|040|016|n=1|-|font4|-|label_rect|-|-|Q тек.,м3/ч
analog_H2S|704|072|frame=1;an_H2S=Q_HH_03
analog_H2S|704|094|frame=1;an_H2S=Q_HH_04
```

3.3.1.18 Секция [start]. Задание действий при запуске

Последовательность действий, выполняемых однократно при запуске программы, задается в секции [start]

Строка описания действия имеет следующий формат:

№	Наименование поля	Примечание
1	Наименование действия	Строка
2	Параметры	Строка

Поле 1 содержит имя действия из секции [action].

Поле 2 содержит параметры действия, указанного в поле 1.

Пример секции [start]:

```
[start]  
begin
```

3.3.1.19 Секция [limit]. Задание пределов значений переменных

Пределы значений задаются в физических единицах измерения и в формате данных контроллера (коды modbus). Пределы служат для вычисления коэффициентов линейного преобразования значения аналоговой переменной, полученной по протоколу modbus, в отображаемое на экране значение в физических единицах измерения, а также обратного преобразования значений, введенных в программе, для передачи по modbus на контроллер. Пределы задаются в секции [limit].

Формат строки секции [limit]:

№	Наименование поля	Примечание
1	Наименование предела	Строка
2	Нижний предел физической величины	Число с пл. зап.
3	Верхний предел физической величины	Число с пл. зап.
4	Нижний предел кода modbus	Целое число 0...65535 По умолчанию = 0
5	Верхний предел кода modbus	Целое число 0...65535 По умолчанию = 4095

Поле 1 содержит наименование предела значений.

Поля 2 и 3 содержат соответственно нижний и верхний пределы значений переменной в заданных для нее единицах измерения.

Поля 4 и 5 содержат соответственно нижний и верхний пределы значений переменной в формате данных контроллера (коды modbus). Значения этих полей неизменны (средствами программы), считываются только из постоянной части конфигурации (из файла Pult_hrd.txt).

Пример секции [limit]:

```
[limit]
IHOURKONTR|0|23|0|23
IMINKONTR|0|59|0|59
ISECKONTR|0|59|0|59
ITE_1PV|-55|150|0|4095
ZTE_1H|-55|150|0|4095
```

3.3.1.20 Секция [var]. Создание переменных

Для организации динамического отображения данных в программе используется концепция «переменных». В целом концепция соответствует использованию переменных в языках программирования высокого уровня.

Все переменные, используемые в системе Spica, образуют общий список переменных.

Переменные имеют обязательное свойство «имя», которое должно быть уникальным, поскольку используется для обращения к переменной, к её значению и другим свойствам.

Переменная может быть ассоциирована с буфером контроллера, тогда её значение устанавливается в процессе обмена данными. Для ассоциации переменной с одной или несколькими ячейками буферов контроллера, необходимо указать их адреса в соответствующих полях настройки переменной.

Так же переменная может быть использована без ассоциации с буфером контроллера для реализации внутреннего функционала проекта (например, участвовать при задании условия или события действия).

Значение переменной, её наименование, единица измерения, минимальный и максимальный предел значений, могут выступать в качестве строкового контента при настройке последовательностей и боксов.

Для переменной может быть задано преобразование – правило перевода значения в формате данных контроллера в значение в физических единицах измерения для вывода на экран. Если преобразование не задано, на экране будут отображаться данные с контроллера в неизменном виде (целые числа).

Переменные и их настройки задаются в секции [var].

Формат строки секции [var]:

№	Наименование поля	Примечание
1	Наименование переменной	Строка
2	Адреса чтения	Строка адреса
3	Адреса записи	Строка адреса
4	Обозначение переменной	Строка
5	Обозначение единицы измерения для аналоговых значений, либо перечень состояний для дискретных значений, либо имя последовательности с перечислением состояний для дискретных значений	Строка / Строка с разделителем
6	Преобразование	Строка
7	Количество знаков при выводе текста значения переменной	Целое число
8	Действие при изменении значения переменной	Строка
9	Параметры вызова действия	Строка
10	Критерии для занесения значений в архив	Целое число / Десятичная дробь

Поле 1 содержит имя переменной.

Поле 2 содержит адрес чтения значения переменной (адрес выходного буфера контроллера).
Массивы данных 1x или 3x.

Поле 3 содержит адрес записи значения переменной (адрес входного буфера контроллера).
Массивы данных 0x или 4x.

Каждая переменная может иметь связь с одним или несколькими адресами.

Формат адреса: **PCU:addr**, где

PCU – наименование контроллера (PCU) из секции [pcu].

addr – число или переменная, значение которой вычисляется по формуле:

*массив*10000+ссылка*

Примеры:

Строка **ZXXXPV|PCU1:30010|PCU1:40010** означает, что переменная ZXXXPV читается из контроллера PCU1 из массива данных 3x по ссылке 10, и записывается в массив 4x по ссылке 10.

Строка **SXXXPV|PCU1:10025;PCU1:10026|-** означает, что переменная SXXXPV читается из контроллера PCU1 из массива данных 1x по ссылкам 25 и 26. Первым указывается младший бит. В этом случае переменная будет иметь четыре возможных значения:

- Ссылка 25 содержит **0**, 26 – **0**: значение **0 (00)**
- Ссылка 25 содержит **1**, 26 – **0**: значение **1 (01)**
- Ссылка 25 содержит **0**, 26 – **1**: значение **2 (10)**
- Ссылка 25 содержит **1**, 26 – **1**: значение **3 (11)**

Запись в контроллер не осуществляется (прочерк).

Строка **UXXXXK|PCU2:33|PCU2:33** означает, что переменная UXXXXK читается из контроллера PCU2 из массива данных 0x по ссылке 33, и записывается в контроллер PCU2 в массив данных 0x по ссылке 33.

Строка **IYYYPV|PCU2:30010;PCU2:30011|-|xxxxxxx|xxx|FLOAT1** означает, что значение переменной IYYYPV формируется как число с плавающей точкой в формате IEEE-754 single precision размером 4 байта (sign bit, 8-bit exponent, 23-bit mantissa), и читается из контроллера PCU2 из массива данных 3x по ссылкам 10 и 11. Способ компоновки двух полученных значений в число с плавающей точкой указано в поле «Преобразование» в виде зарезервированного обозначения FLOAT1. Запись в контроллер не осуществляется (прочерк).

Примечание: На экран всегда выводится значение переменной, полученное с контроллера, в том числе при вводе нового значения переменной.

Поле 4 содержит текстовое обозначение переменной (комментарий).

Поле 5 используется в зависимости от типа переменной.

Если переменная служит для отображения аналоговых значений, то поле 5 содержит единицы измерения переменной.

Если переменная служит для отображения дискретных состояний, то поле 5 содержит либо имя последовательности с текстовыми обозначениями дискретных состояний, либо непосредственное перечисление текстовых обозначений дискретных состояний (не менее двух), разделённых символом точка с запятой. Эти обозначения будут отображаться в сводке тревог и в файле архива.

Поле 6 содержит тип преобразования значения, полученного с контроллера.

В поле 6 указывается либо наименование предела из секции [limit] (для линейного преобразования), либо обозначение специального преобразования.

Если указан предел, полученное по каналу связи значение, при выводе на экран преобразуется в число в формате с плавающей точкой по линейному закону. Коэффициенты линейного преобразования вычисляются на основании пределов значений переменной.

Во втором случае, полученные по каналу связи данные укладываются в непрерывный массив и интерпретируются особым образом, в соответствии с типом преобразования.

Специальные преобразования:

<i>Тип преобразования</i>	<i>Описание</i>
---------------------------	-----------------

FLOAT1	Массив данных интерпретируются как число с плавающей точкой однократной точности стандарта IEEE-754

Поле 7 содержит целое число, определяющее количество знаков для формирования текстового отображения значения переменной. Необязательный параметр. При его отсутствии количество знаков рассчитывается программой автоматически. Если преобразование не задано (то есть переменная содержит целые значения), то данное поле определяет общее количество выводимых знаков с учётом дополнительных нулей перед целым числом, иначе (если переменная содержит дробные значения) – поле определяет количество знаков после запятой.

Поле 8 содержит имя действия при изменении переменной.

Поле 9 содержит параметры для действия, имя которого указано в поле 8.

Поле 10 содержит число, которое представляет собой либо зону нечувствительности для функции сохранения архивных значений (для аналоговых переменных), либо кодовое обозначение классификации события (для логических переменных).

Значение по умолчанию: **-1** (активация тревоги и архивирование отсутствует)

Содержимое поля 10 должно соответствовать типу переменной и типу преобразования, указанному в поле 6:

<i>Тип переменной</i>	<i>Поле 10</i>	<i>Смысл</i>	<i>Размерность / код</i>
Аналоговая с линейным преобразованием	десятичное целое число	зона нечувствительности для функции архивирования	"сырые" коды АЦП
Аналоговая с преобразованием FLOAT1	десятичная дробь (разделитель точка)	зона нечувствительности для функции архивирования	физическая величина переменной.
Дискретная	десятичное целое число	код возникновения аварии или события	5 = событие перехода из 0 в 1 6 = событие ... из 1 в 0 9 = тревога ... из 0 в 1 10 = тревога ... из 1 в 0

			(аналогично полю Arg0 системы Антарес)
--	--	--	--

Пример секции [var]:

```
[var]
IHOURLKONTR|PCU1:30175|-|Время в контроллере|час|IHOURLKONTR
IMINKONTR|PCU1:30176|-|Время в контроллере|мин|IMINKONTR
ISECKONTR|PCU1:30177|-|Время в контроллере|сек|ISECKONTR
ITE_1PV|PCU1:30181|-|Т. подш. ПС. Н-1В|град.С|ITE_1PV|||10
ZTE_1H|PCU1:30001|PCU1:40001|Т. подш. ПС. Н-1В-макс|град.С|ITE_1PV
STE_1H|PCU1:10181|-|Т. подш. ПС. Н-1В|норма;ПЕРЕГРЕВ|||9
STE_2H|PCU1:10182|-|Т. подш. ПС. Н-2В|норма;ПЕРЕГРЕВ|||5
SM_PUMP1|PCU1:10188;PCU1:10189|-|Режим работы насоса Н1|PUMPMODE|||5
SM_PUMP2|PCU1:10190;PCU1:10191|-|Режим работы насоса Н2|НЕОПР;ДИСТ;АВТО;НЕОПР|||5
ITE_BVR1_PV|GZNU-10:30071;GZNU-10:30072|-|Температура газа|гр.С|FLOAT1|||0.1
```

Пояснение для второго поля (единицы измерения):

Переменная SM_PUMP2 связана с двумя адресами буфера 1х протокола ModBus.

Таким образом значение переменной формируется двумя битами информации.

Младший бит связан с адресом 10190 PCU1.

Старший бит связан с адресом 10191 PCU1.

Возможные значения переменной SM_PUMP2: «0»; «1»; «2»; «3».

Пояснение для пятого поля (единицы измерения):

Единица измерения переменной ITE_1PV – «град.С» (градусы Цельсия).

Переменная SM_PUMP2 отображает режимы работы насоса.

Значению переменной «1» соответствует режим «ДИСТ» (дистанционный)

Значению переменной «2» соответствует режим «АВТО» (автоматический)

Остальные значения («0» и «3») обозначены как – «НЕОПР» (неопределённый)

При описании переменной насоса STE_1H точно такие же состояния взяты из последовательности PUMPMODE, описанной в секции [seq].

Пояснение для десятого поля (архивирование):

Отсутствие поля 10 у переменных I HOURKONTR, IMINKONTR, ISECKONTR, ZTE_1H означает, что эти переменные не архивируются и не генерируют тревоги/события.

Число 10 в десятом поле переменной ITE_1PV означает, что значение переменной ITE_1PV архивируются каждый раз, когда изменение кода АЦП, передаваемого из регистра с адресом 30181, достигает 10 или более единиц (по сравнению с предыдущим сохранённым значением).

Число 9 в десятом поле переменной STE_1H означает, что каждый раз, когда переменная STE_1H меняет своё значение с «0» на «1», в сводке тревог будет появляться запись, помеченная красным цветом (тревога). Обратное изменение создаёт запись белого цвета. Все изменения по тегам, генерирующим тревоги, архивируются, следовательно, все изменения переменной STE_1H попадут в архив.

Число 5 в десятом поле переменной STE_2H означает формирование архивной записи для этой переменной при изменении её значения от 0 к 1. Это событие не отображаются в сводке тревог.

3.3.1.21 Секция [arc]. Функция архивирования

Архивирование – это сохранение значений переменных в файл с указанием даты и времени.

Имя файла архива формируется на основе сочетания символов «**arc**», двух последних цифр года и порядкового номера месяца в году по шаблону:

arc_YY_MM.txt

Файл архива имеет структуру текстового файла с кодировкой Windows-1251 и строчным разделителем в виде последовательности символов 0x0D 0x0A.

Каждая строка в файле - архивная запись.

Записи состоят из полей, разделённых символом Tab (0x09)

Поля по порядку:

<Дата>, <Время>, <Имя переменной>, <Комментарий к переменной>, <Значение>, <Единицы измерения>

Дата записывается в формате: DD/MM/YYYY

Время записывается в формате: hh:mm:ss

Значение записывается в формате, соответствующем типу переменной, с учётом параметра количества знаков.

Формирование архивных записей происходит в соответствии с критериями, заданными для переменной (поле 10 секции [var]). Так же существует возможность формирования архивных записей при выполнении специального действия **arcrec** (см. список зарезервированных обозначений).

Программа позволяет копировать файлы архивов в заранее определённое место для более длительного и надёжного хранения. Для этого предусмотрена команда **arcsave** (см. список зарезервированных обозначений).

Параметры функции архивирования задаются в секции [arc]

Первая строка секции – каталог архива (путь к рабочим файлам архива)

Вторая строка секции – каталог резервного хранилища (путь для копирования файлов архива командой **arcsave**)

Третья строка секции – целое число, указывающее максимальное количество рабочих файлов архива. Поскольку каждый файл соответствует месяцу работы системы, данное число фактически ограничивает размер архива по количеству месяцев работы.

Пример секции [arc]:

```
[arc]  
c:\Work\test\Arc\  
e:\Save\  
12
```

3.3.1.22 Секция [alarm]. Сводка тревог

При изменении значений логических переменных в соответствии с критериями, которые указаны в секции [var], программа формирует тревожные сообщения – «тревоги».

Тревоги отображаются в виде таблицы специальном фрейме (тип 5).

Фрейм представляет собой таблицу, где тревожное сообщение отображается в виде текста чёрного цвета на фоне красного или белого цвета. При изменении значения тега на тревожное, появляется соответствующая строка на красном фоне, изменение значения тега на нормальное состояние появляется соответствующая строка на белом фоне. Записи в сводке позволяют проследить всю хронологию тревожных событий на объекте, при этом на экране в одно и тоже время видны как момент возникновения события, так и момент возвращения в норму.

Фрейм сводки тревог (тип 5) имеет все свойства обычного фрейма – координаты, условия видимости и т.п.

Пример:

```
000005|005|600|1000|150|-|-|font2|-|bttn_rect|-|-|-|-|
```

Кроме обычных свойств, фрейм сводки тревог имеет два дополнительных свойства:

- Максимальный размер сводки (по умолчанию 10);
- Высота строки в пикселях (по умолчанию 18).

Максимальный размер сводки – это ограничение по количеству отображаемых тревог (максимальное количество строк).

Высота строки в пикселях - это вертикальный размер каждой отображаемой строки таблицы в пикселях.

Если необходимо задать особые параметры фрейма, это можно сделать, создав секцию [alarm]

Единственная строка секции [alarm] должна содержать два целых числа, разделённых символом «|». Первое число: максимальный размер сводки. Второе число: высота строки в пикселях.

Пример секции [alarm]:

```
[alarm]  
50|16
```

3.3.1.23 Секция [time]. Синхронизация времени

Программа осуществляет синхронизацию времени по значениям переменных, определённых в секции [time]. Системные часы ПЭВМ синхронизируются при каждом увеличении значения

переменной «минута» на единицу, если программа обнаруживает расхождение показаний системных часов и переменных даты/времени.

Секцию [time] необходимо располагать после секции [var]. В секции должна быть одна строка – это названия переменных даты и времени, используемых для синхронизации времени. Имена следуют через разделитель «|».

Порядок: год, месяц, день, час, минута

Пример секции [time]:

[time]

IYEARKONTR | IMONTHKONTR | IDAYKONTR | IHOURLKONTR | IMINKONTR

3.3.1.24 Секция [print]. Создание команд печати

В проекте может быть описан документ, который выводится на печать выполнением действия print. Документ описывается элементарными операциями – командами. Команды печати задаются в секции [print]. Каждая строка секции [print] – это одна команда. Существует несколько типов команд:

- Команда инициализации печати (первая строка секции);
- Команда установки текущей позиции;
- Команда выбора шрифта;
- Команда печати строки;
- Команда печати значения переменной;
- Команда печати значения строкового контента;
- Команда печати значения строкового контента из последовательности по значению переменной;
- Команда печати линии;
- Команда сохранения текущей позиции;
- Команда восстановления текущей позиции.

Команда инициализации печати – первая строка секции [print], используется единственный раз. Другие команды могут быть использованы несколько раз. Сочетание команд печати задает структуру документа. В процессе выполнения действия print программа последовательно выполняет все команды секции [print], осуществляя вывод информации на принтер.

1. Формат команды инициализации печати:

№	Наименование поля	Примечание
1	Имя устройства печати	Строка
2	Параметр масштабирования по горизонтали	Целое положительное число (по умолч. = 1)
3	Параметр масштабирования по вертикали	Целое положительное число (по умолч. = 1)

Команда инициализации печати содержит сетевое имя устройства для вывода на печать и общие параметры масштабирования.

Параметр масштабирования – это число, на которое делится соответствующая координата для получения координат в контексте устройства печати.

Каждая следующая команда начинается с названия и содержит ряд параметров.

Печать текста или линии требует указания координат привязки. Для текста это левый или правый (зависит от параметра «выравнивание») верхний угол, для линии это начало и конец линии. Текущая позиция может быть установлена специальной командой или является результатом выполнения предыдущей команды (печати текста или линии).

Указание координат производится одним из двух способов – абсолютным или относительным.

Абсолютный способ – это непосредственное указание координат в пикселях. Если целое число в поле *левый* или *верхний* написано без знака «+» или «-», то соответствующая координата указана абсолютным способом.

Относительный способ – это указание смещения координат в пикселях от текущей позиции. Если целое число в поле *левый* или *верхний* содержит слева знак «+» или «-», то соответствующая координата указана относительным способом.

Способ указания текущей позиции для каждой из двух координат (горизонтальной и вертикальной) выбирается независимо.

2. Формат команды установки текущей позиции:

№	Наименование поля	Примечание
1	Команда	<i>move</i>
2	Левый	Целое число (знак и целое число)
3	Верхний	Целое число (знак и целое число)

Выполнение команды установки текущей позиции приводит к установке новой текущей позиции и устанавливает точку возможного начала линии (для дальнейших команд *line*).

3. Формат команды выбора шрифта:

№	Наименование поля	Примечание
1	Команда	<i>font</i>
2	Название шрифта	Строка
3	Ширина шрифта	Целое число
4	Высота шрифта	Целое число
5	Жирный	1 или 0
6	Курсив	1 или 0

После выполнения команды выбора шрифта вывод текста осуществляется с использованием указанного шрифта. Текущая позиция не изменяется.

4. Формат команды печати строки:

№	Наименование поля	Примечание
1	Команда	<i>text</i>
2	Левый	Целое число (знак и целое число)
3	Верхний	Целое число (знак и целое число)
4	Выравнивание	Символ l или r
5	Текст	Строка

Выполнение команды печати строки приводит к печати строки из поля *текст*.

Если выбрано выравнивание слева (l), то текст будет напечатан вправо и вниз относительно точки привязки и текущая позиция устанавливается на месте правого верхнего угла напечатанного текста.

Если выбрано выравнивание справа (r), то текст будет напечатан влево и вниз относительно точки привязки и текущая позиция устанавливается на месте левого верхнего угла напечатанного текста.

5. Формат команды печати значения переменной:

№	Наименование поля	Примечание
1	Команда	<i>value</i>
2	Левый	Целое число (знак и целое число)
3	Верхний	Целое число (знак и целое число)
4	Выравнивание	Символ l или r
5	Имя переменной	Строка

Выполнение команды печати значения переменной приводит к печати значения переменной с заданным именем.

Выравнивание и установка текущей позиции происходит аналогично печати текста командой *text*.

6. Формат команды печати значения строкового контента:

№	Наименование поля	Примечание
1	Команда	<i>cont</i>
2	Левый	Целое число (знак и целое число)
3	Верхний	Целое число (знак и целое число)
4	Выравнивание	Символ l или r
5	Имя контента	Строка

Выполнение команды печати значения строкового контента приводит к печати значения строкового контента.

Выравнивание и установка текущей позиции происходит аналогично печати текста командой `text`.

7. Формат команды печати значения строкового контента из последовательности по значению переменной:

№	Наименование поля	Примечание
1	Команда	<i>seq</i>
2	Левый	Целое число (знак и целое число)
3	Верхний	Целое число (знак и целое число)
4	Выравнивание	Символ <code>l</code> или <code>r</code>
5	Имя переменной	Строка
6	Имя последовательности	Строка

Выполнение команды печати значения строкового контента из последовательности по значению переменной приводит к печати значения строкового контента, выбранного из заданной последовательности, по значению переменной с заданным именем.

Выравнивание и установка текущей позиции происходит аналогично печати текста командой `text`.

8. Формат команды печати линии:

№	Наименование поля	Примечание
1	Команда	<i>line</i>
2	Левый	Целое число (знак и целое число)
3	Верхний	Целое число (знак и целое число)

Координаты *левый* и *верхний* определяют точку конца линии.

Выполнение команды печати линии приводит к печати линии толщиной в 1 пиксель из текущей позиции в точку с указанными координатами. Устанавливает текущую позицию и возможное начало следующей линии (для дальнейших команд `line`).

Предусмотрена возможность сохранения в память и восстановления из памяти текущей позиции. Для этого в программе выделен массив размерностью в сто ячеек.

9. Формат команды сохранения текущей позиции:

№	Наименование поля	Примечание
1	Команда	<i>save</i>
2	Номер ячейки	Целое число (1...100)

Выполнение команды сохранения текущей позиции приводит к сохранению текущей позиции в ячейке памяти с указанным номером.

Данная команда не передается принтеру, и не производит никаких действий непосредственно связанных с печатью.

10. Формат команды восстановления текущей позиции из памяти:

№	Наименование поля	Примечание
1	Команда	<i>restore</i>
2	Номер ячейки	Целое число (1...100)

Выполнение команды восстановления текущей позиции приводит к установке текущей позиции из ячейки памяти с указанным номером.

Данная команда не передается принтеру, и не производит никаких действий непосредственно связанных с печатью.

Пример секции [print]:

```
[print]
HP LaserJet 2015
window|210|200
font|Arial|1600|3200|0|0
move|10000|05000
line|+100000|+0
line|+0|+80000
line|-100000|+0
line|+0|-80000
move|+35000|+01000
text|+0|+0|1|ЦПРС НГДУ "ДжН" ППТЖ N1
font|Arial|1600|3200|1|0
```

3.3.1.25 Секция [numrad]. Размер цифровой панели

Размер стандартной цифровой панели ввода описывается в секции [numrad]. Секция содержит целое число, определяющее размеры кнопок на цифровой панели. Необязательная секция. При её отсутствии размер цифровой панели рассчитывается автоматически.

Пример секции [numrad]:

```
[numrad]  
30
```

3.3.1.26 Список зарезервированных обозначений

Существует несколько особых обозначений, применяемых для составления файлов конфигурации. Они приведены в таблице ниже. Рекомендуется не использовать данные обозначения иначе, чем это предусмотрено данной таблицей.

Обозначение	Тип	Пояснение
exit	действие	Закрывает текущее верхнее окно программы
edit	действие	Вызывает окно редактирования значения переменной или текстового контента *
editMax	действие	Вызывает окно редактирования максимального предела переменной
editMin	действие	Вызывает окно редактирования минимального предела переменной
lists	действие	Вызывает появление окна ListBox выбора элемента из списка (текстовой последовательности). Порядковый номер выбранного элемента устанавливает новое значение переменной *
listv	действие	Вызывает появление окна ListBox выбора элемента из списка (текстовой последовательности). Текст выбранного элемента устанавливает новое значение переменной текста. *
archive	действие	Сохраняет текущие значения в файл *
save	действие	Сохраняет текущие значения в файл текущих значений *
load	действие	Считывает текущие значения из файла текущих значений. *
arcrec	действие	Архивирует значения переменных. Имена переменных следует перечислить в отдельном поле «параметров» действия через разделитель – символ «точка с запятой» (;)
arcsave	действие	Копирует файлы архива в резервную папку (или на резервный носитель)
execute	действие	Вызывает внешнее приложение на выполнение *
print	действие	Вывод на печать в соответствии с командами из секции [print]
dev_on_PCU	действие	Включает опрос контроллера с наименованием PCU
dev_off_PCU	действие	Выключает опрос контроллера с наименованием PCU
dev_Ok	контент	Обозначает состояние связи по всем PCU. dev_Ok = 1 – есть стабильная связь по всем PCU dev_Ok = 0 – нет связи ни с одним PCU Если связь есть, но она не стабильна, либо нет связи с частью PCU, то значение dev_Ok меняется с интервалом примерно каждые полсекунды.
value	контент	Значение переменной
valMin	контент	Минимальный предел значения переменной
valMax	контент	Максимальный предел значения переменной
valName	контент	Наименование параметра
valUnits	контент	Единицы измерения параметра
red	контент	Красный цвет

Обозначение	Тип	Пояснение
green	контент	Зеленый цвет
blue	контент	Синий цвет
white	контент	Белый цвет
silver	контент	Светло-серый цвет
black	контент	Черный цвет
AutoAction	действие	Выполняется при любом «клике» мышью
cancel	действие	Закрывает все открытые диалоги, оставляет только основное окно программы
FLOAT1	преобразование	При указании преобразования с таким именем в описании переменной (см. секция [var]) значение этой переменной формируется из двух составных частей, указанных в поле адреса.

* – команды могут (или должны) быть применены с параметрами.

Пояснение для применения команд с параметрами

Команда **edit** может применяться без параметров или с параметром отложенного вызова (в двойных кавычках), например:

edit

edit«actName»

edit«actName1;actName2;N=1»

где **actName**, **actName1**, **actName2** – имена действий из секции [action]. В кавычках указан перечень действий, которые будут вызваны на исполнение при завершении редактирования.

Команды **lists** и **listv** могут применяться без параметров, с параметрами указания на переменную и последовательность (разделитель – знак «^»), и/или с параметром отложенного вызова (в двойных кавычках), например:

lists

lists^var^seq«actName»

listv«actName1;actName2;N=1»

где **actName**, **actName1**, **actName2** – имена действий из секции [action], **var** – имя переменной из секции [var], **seq** – имя последовательности из секции [seq]. В кавычках указан перечень действий, которые будут вызваны на исполнение при выборе элемента из выпадающего списка.

Команда **execute** применяется с параметром командной строки для запуска (разделитель – знак «^»), например:

execute^explorer^http://ya.ru

Команды **archive**, **save**, **load** применяются с обязательным параметром имени последовательности (разделитель – знак «^»), например:

archive^seq1

save^seq2

load^seq2

где **seq1**, **seq2** – имена последовательностей из секции [seq].

Формат записи в файл при выполнении команд **save** и **archive**.

Команда **save** аналогична команде **archive**.

Отличие команд - в имени формируемого файла.

Команда **save** формирует имя файла на основе имени последовательности, указанной в параметрах команды: **<имя последовательности>.txt**

Команда **archive** формирует имя файла: **archive.txt**

Файлы сохраняются в каталог конфигурации проекта.

Формат файла – текстовый, в кодировке cp1251.

Разделитель строк – символы 13,10.

Разделитель полей – символ |.

Каждая запись содержит строку-идентификатор начала записи и количество строк, равное количеству элементов сохраняемой последовательности.

Строка-идентификатор содержит два поля: первое поле пустое, второе – имя сохраняемой последовательности.

Строка с текстом контента состоит из двух полей: первое поле – имя записываемого контента, второе – текст.

Строка со значением переменной состоит из трех полей: первое поле – имя записываемой переменной, второе – код, третье – значение физической величины в формате, определенном для данной переменной (как для вывода на экран).

Строки добавляются в конец файла.

3.3.1.27 *Формат файла инициализации Pult.ini*

Файл инициализации Pult.ini. служит для задания расположения конфигурации проекта.

Файл текстовый. Содержит одну секцию - **[Project]**.

В секции **[Project]** указывается расположение конфигурации проекта. Необходимо прописать полный путь (dir=\xxx\...). Путь должен начинаться и завершаться символом '\

Пример файла инициализации:

```
[Project]  
dir=\HardDisk\
```

3.3.1.28 Формат файла инициализации Pult_xp.ini

Файл инициализации Pult_xp.ini. служит для задания параметров основного окна программы MiniSCADA Spica (XP), а также параметры работы с проектом.

Файл текстовый. Состоит из двух секций - **[Window]** и **[Project]**.

Секция **[Window]** содержит параметры основного окна программы:

- положение окна на экране компьютера. Задается координатами его левого верхнего угла. X – слева, Y – сверху;
- размеры окна программы. W – ширина, H – высота.

Секция **[Project]** содержит параметры работы с проектом:

- расположение конфигурации проекта. Необходимо прописать полный путь (dir=x:\xxx\...\). Путь должен завершаться символом '\';
- режим работы программы. Возможны 2 режима: режим отладки и рабочий режим.

Режим отладки (mode=1) предназначен для разработки и проверки файлов конфигурации. В режиме отладки основное окно имеет системную область и зону настройки размеров. В нижней части окна выведены вспомогательные параметры – координаты последнего клика «мышью». При изменении размеров и положения окна, после закрытия программы новые размеры и координаты записываются в файл инициализации.

Рабочий режим (mode=0 – по умолчанию) предназначен для нормальной работы в составе системы автоматизации ТП. В рабочем режиме положение основного окна задаётся только параметрами секции **[Window]** файла инициализации и не может быть изменено в процессе работы.

Пример файла инициализации:

```
[Window]
X=0
Y=0
W=640
H=480
M=1
[Project]
dir=c:\Work\MiniSCADA Spica\Project\
mode=0
```

3.4. Завершение программы

Предусмотрено два способа для завершения программы:

- вызов действия **exit**;
- нажатие клавиш в сочетании **Alt+F4**.

При разработке файлов конфигурации **Pult_hrd.txt** рекомендуется предусмотреть место на экранной форме для кнопки с действием **exit**.

Если по каким-то причинам кнопка **exit** не доступна, можно воспользоваться клавиатурой - нажать клавишу Alt, а затем, удерживая её, нажать клавишу F4. Это приведёт к закрытию окна программы и завершению работы.

3.5. Организация используемой входной информации

Входные данные программы MiniSCADA Spica должны быть организованы следующим образом:

- текст, составленный согласно структурам, описанным в п."Создание конфигурации проекта" и оформленный в виде файлов конфигурации Pult_hrd.txt и Pult_flx.txt и инициализации Pult.ini;
- используемые в проекте графические изображения, хранящиеся в виде файлов формата *.bmp;
- данные, поступающие по Modbus.

3.6. Организация используемой выходной информации

Выходные данные программы MiniSCADA Spica организованы в двух видах:

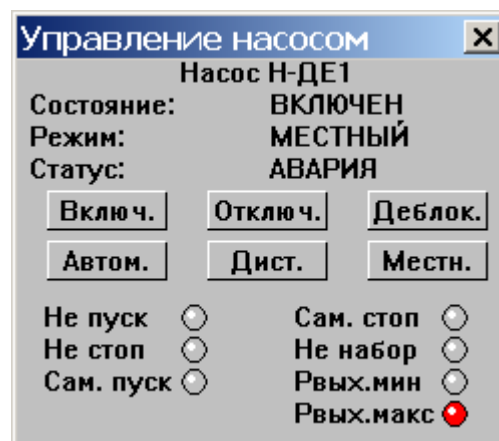
- экранные формы и диалоговые окна, являющиеся основным средством взаимодействия человека и системы автоматизации;
- управляющие воздействия, передаваемые по Modbus.

Примеры.

Экранная форма



Диалоговое окно



4. Сообщения

Во время запуска и работы программы возможно появление следующих сообщений:

<i>Заголовок</i>	<i>Сообщение</i>	<i>Вероятная причина сообщения</i>
Запуск невозможен!	Не удалось прочитать файл <имя файла>	Неправильно задан параметр dir файла ini
Внимание!	Невозможно открыть порт <имя порта>	Порт последовательного интерфейса, указанный в секции [port], не существует на данной ПЭВМ

