Руководство программиста

miniSCADA Spica

Содержание

1 Описание программы miniSCADA Spica

Программа **miniSCADA Spica** – утилита, осуществляющая функции человеко-машинного интерфейса.

Основным средством взаимодействия человека (оператора технологического процесса) и машины (системы автоматизации) являются экранные формы, на которых наглядно, в виде мнемосхем, отображается работа контролируемых технологических объектов. Изображения мнемосхем динамические, то есть всегда соответствуют актуальным (текущим) данным и результатам их алгоритмической обработки.

Программа позволяет с рабочего места оператора осуществлять управляющие воздействия – передавать на технологические объекты команды, изменять режимы, параметры работы объектов и прочее.

Программа позволяет осуществлять разграничение доступа к своим функциям посредством аутентификации (ввода пароля).

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.

Существует две версии программы miniSCADA Spica:

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

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

Программа **miniSCADA Spica (XP)** предоставляет расширенные возможности при разработке и отладке, поэтому при разработке конфигурации проекта удобнее использовать данную версию.

Программа **miniSCADA Spica** связывается с CPU по протоколам:

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

2 Необходимое программное и аппаратное обеспечение

Программа miniSCADA Spica работает под операционными системами Windows-CE, Windows-XP, Windows-XP-Embedded, и под другими совместимыми операционными системами.

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.

3 Протокол Modbus

Программа **miniSCADA Spica** связывается с CPU 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. Она 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.

4 Установка программы miniSCADA Spica

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 прописать в файле инициализации.

5 Запуск программы miniSCADA Spica

Для запуска программы miniSCADA Spica необходимо:

1. Убедиться, что в файле инициализации указаны необходимые параметры (п. 8 Формат файла инициализации)
2. Запустить приложение Pult.exe (для Windows-CE) или Pult_xp.exe (для Windows-XP) стандартными средствами Windows.

6 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.

7 Создание конфигурации проекта miniSCADA Spica

Программа miniSCADA Spica осуществляет визуализацию технологического процесса и обеспечивает управление им.

Конфигурация считывается 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. 7.6 Добавление PCU) по прописанным в файле адресам (секция [port], п. 7.2 Настройка каналов связи), и осуществляет обмен данными.

Программа осуществляет обмен данными с контроллером посредством **переменных** (секция [var], п. 7.7 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. 7.5 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. 7.11 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. 7.12 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 таком случае, в качестве их значения указывается **последовательность**.

Для фрейма может быть указано **действие**, которое будет выполнено при нажатии на фрейм. Список действий содержится в секции [actions] (п. 7.14 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. 7.4 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. 7.15 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. 7.17 Конфигурирование основного окна проекта). **Другие окна** описываются в секции [dialog] (п. 7.16 Создание диалоговых окон).

Файлы конфигурации состоят из секций:

- [port];
- [modbus];
- [pw];
- [limit];
- [pcu];
- [var];
- [color];
- [str];
- [bmp];

- [seq];
- [rect];
- [font];
- [action];
- [stamp];
- [dialog];
- [form];
- [start];
- [task];
- [print].

Порядок следования секций соблюдать обязательно.

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], в случае, если функционал, задаваемый этими секциями, в проекте не используется.

7.1 Формат файлов конфигурации

Файлы конфигурации текстовые, имеют формат *.txt.

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.

7.2 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: e – контроль четности, o – контроль нечетности, n – 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 контролером.

Пример секции [port]:

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

7.3 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

7.4 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

7.5 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

7.6 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 [pcu]:

[pcu]
PCU1|3|IP1
PCU2|5|COM5

7.7 Создание переменных

Основное назначение переменных – организация обмена данными с контроллером. 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 для реализации внутреннего функционала проекта (например, участвовать при задании условия или события действия).

Значение переменной, ее 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	Строка
6	Преобразование	Строка
7	Количество знаков до/после запятой при выводе значения переменной	Целое число
8	Действие при изменении значения переменной	Строка
9	Параметры вызова действия	Строка

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 **IXXXPV|PCU1:30010|PCU1:40010** означает, что переменная IXXXPV 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 **UXXXK|PCU2:10033|PCU2:33** означает, что переменная UXXXK читается из контроллера PCU2 из массива данных 1x 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 содержит единицы измерения переменной.

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 типа 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. Например, если в этом поле указано обозначение **FLOAT1**, то полученный массив интерпретируются как число с плавающей точкой однократной точности стандарта IEEE-754.

Поле 7 содержит количество знаков после запятой.

Количество знаков до/после запятой – необязательный параметр. При его отсутствии количество знаков после запятой рассчитывается программой автоматически. Если преобразование не задано (то есть переменная содержит целые значения), то данное поле определяет количество знаков до запятой, иначе (переменная содержит дробные значения) – после запятой.

Поле 8 содержит имя действия при изменении переменной.

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

Пример секции [var]:

[var]
IHOURKONTR|PCU1:30175|-|Время в контроллере|час|IHOURKONTR
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
ZTE_1H|PCU1:30001|PCU1:40001|Т.подш.ПС.Н-1В-макс|град.С|ITE_1PV

7.8 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
gray|8421504
violet|16711935

7.9 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.

7.10 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. 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|\HardDisk\pic1.bmp
pic2|\HardDisk\pic2.bmp
pic3|\HardDisk\pic3.bmp
lamp_x|\HardDisk\lamp_x.bmp
lamp_red|\HardDisk\lamp_red.bmp
lamp_green|\HardDisk\lamp_green.bmp

7.11 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

7.12 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

7.13 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

7.14 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

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 (п. 7.21 Список зарезервированных обозначений);
- операции присваивания.

№	Тип присваивания	Пример	Результат
1	Непосредственное присваивание кода переменной	var=5	Код переменной var будет равен 5
2	Присваивание кода одной переменной от другой переменной	var1=var2	Код переменной var1 будет равен коду переменной var2
3	Присваивание кода переменной с арифметическими операциями	var1=var2+1	Код переменной var1 будет равен коду переменной var2 плюс 1

№	Тип присваивания	Пример	Результат
4	Установка признака «недостоверность» переменной	var1#	Код переменной var1 останется без изменения, переменная var1 становится недостоверной
5	Сброс признака «недостоверность» переменной	var=	Код переменной var1 останется без изменения, переменная var1 становится достоверной
6	Присваивание строкового контента	str=произвольный текст	Текст контента str будет текстом после знака равенства: «произвольный текст»
7	Присваивание строкового контента из последовательности по значению переменной	str=var@seq	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_kvit|-|U(X)K=1

7.15 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. 7.17 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
000001|000|000|014|020|-|-|-|-|main_time_rect|-|-|(main_time_value)
$|analog_H2S
000001|000|000|065|020|n=(frame)|-|-|-|analog_rect|-
|dev_Ok|I(an_H2S)PV||Analog_H2S_Diag|Analog_name=(an_H2S);frame=(frame)
000001|000|000|011|010|n=(frame)|-|-|-|HI_yellow_lim_rect|S(an_H2S)H|-|-|
000001|000|000|011|010|n=(frame)|-|-|-|HH_lim_rect|S(an_H2S)HH|-|-|
000001|000|000|011|020|n=(frame)|-|-|-|ERR_rect|S(an_H2S)E|-|-|
```

7.16 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. 7.17 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|-|-|-|-|bttn_rect|-|-|-|Включить|Com_siren_on|
000008|060|032|100|020|-|-|-|-|bttn_rect|-|-|-|Отключить|Com_siren_off|
$|Klapan_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

7.17 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.
- 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

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

7.18 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

7.19 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

7.20 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	***move***
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	***font***
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	***text***
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	***value***
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	***cont***
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	***seq***
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
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	***line***
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	***save***
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

№	Наименование поля	Примечание
1	Команда	***restore***
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|l|ЦПРС НГДУ "ДжН" ППТЖ N1
font|Arial|1600|3200|1|0

7.21 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 архива. *
save	действие	Сохраняет текущие значения в файл текущих значений. *
load	действие	Считывает текущие значения из файла текущих значений. *
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
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

Обозначение	Тип	Пояснение
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.

Файлы сохраняются в каталог конфигурации проекта.

Имя файла архива: archive.txt.

Имя файла текущих значений последовательности: <имя последовательности>.txt.

Формат файла – текстовый, в кодировке ср1251.

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.

8 Формат файла инициализации

8.1 Формат файла инициализации Pult.ini

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

Файл текстовый. Содержит одну секцию.

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

Пример файла инициализации:

[Project]
dir=\HardDisk

8.2 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.

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
```