

**Комплекс интеллектуальной защиты
Mega SX-Light**

Приложение 1.

Построение профилей.

Содержание.

1. Общее описание структуры профиля.
2. События.
3. Типы данных и работа с ними.
4. Операторы.
5. Определение макросов и их использование.
6. Подключаемые файлы.
7. Механизмы устройства.
Входы.
Выходы.
Сеть 1.
Модем.
Озвучивание
DTMF
Модем GPRS.
Touch Memory.
Система.
Память.
Буфер текста.
8. События устройства.
9. Примеры обработчиков событий.

1. Общее описание структуры профиля.

Профиль – это описание алгоритма работы устройства. С точки зрения пользователя устройство представляет собой набор механизмов. Механизм – функциональный узел устройства программный или аппаратный. Например: модем, входы, выходы, память, таймеры и т.д. Механизмы имеют свои методы. Метод это способ воздействия на механизм для управления им, для его настройки или получения информации о нём. В процессе работы устройства механизмы могут вызывать различные события. Действия, которые выполняются при возникновении события – обработчик события.

Именно обработчики событий и составляют основную часть профиля. В общем виде структура любого профиля представляет собой следующее:

```
Событие_1
...
Обработчик события 1
```

```
...
Конец_события
```

```
Событие_2
...
Обработчик события 2
```

```
...
Конец_события
```

```
Событие_3
...
Обработчик события 3
```

```
...
Конец_события
```

Обработчик события может содержать в себе вызовы методов механизмов и данные. Все названия методов механизмов и событий, а так же все другие названия, которые могут быть использованы при построении профиля, называются **макросами**. Все макросы устройства сведены в программе редактора в общий список, который разделён на группы макросов для удобства поиска нужного названия. Названия макросов начинаются с буквы, состоят из букв и цифр, не имеют пробелов, для разделения слов используется символ подчёркивания `_`. Кроме этого профиль может содержать комментарии. Комментарии не влияют на алгоритм, а служат для пояснения текста профиля. Есть два вида комментариев. Строчный – обозначается символами `//` всё то что правее до конца строки комментарий. Блочный – начало обозначается символами `/*` а конец `*/` всё что находится между ними комментарий.

Примеры комментариев:

// Строчный комментарий

*/**

Блочный комментарий

**/*

2. События.

Все названия событий находятся в группе События. Так же в этой группе содержится макрос `Конец_события`. Концом события должен завершаться любой обработчик события.

Пример:

```
СОБЫТИЕ_Включение_питания
[Действия выполняемые при включении питания]
Конец_события
```

Один обработчик события может быть назначен на несколько событий, если на эти события должна быть одинаковая реакция.

Пример:

```
СОБЫТИЕ_Включение_питания
СОБЫТИЕ_Смена_программы
[Действия выполняемые при включении питания и при смене профиля]
Конец_события
```

Один обработчик события может являться частью обработчика другого события.

Пример:

```
СОБЫТИЕ_Включение_питания
[Действия выполняемые при включении питания]
СОБЫТИЕ_Смена_программы
[Действия выполняемые при включении питания и при смене профиля]
Конец_события
```

3. Типы данных и работа с ними.

Профиль может содержать данные следующих типов:

- Числовые.
- Строковые.
- Ссылочные.

Числовые данные представляют собой положительные числа в диапазоне 0...65535.

Числовые данные могут участвовать в математических и логических выражениях.

Пример:

```
10+5
3*5-1
```

Результат выражения может использоваться в дальнейших вычислениях.

Пример:

```
10+5 // Результат 15
+1 // Результат 15+1 = 16
```

Числовые данные могут служить аргументами для вызываемых методов механизмов, а так же могут являться результатом их работы.

Пример:

```
30 Выход_время_включения //Число 30 - аргумент макроса
```

```
Получить_напряжение_на_входе //После выполнения этого метода
//получим значения.
```

Строковые данные – представляют собой строки текста в кавычках размером до 70-ти символов. Строковые данные, не могут участвовать в выражениях. Они могут лишь выступать в роли аргумента методов механизмов и являться результатом их работы.

Пример:

“Строковой аргумент” Установить_текст_СМС //Строка - аргумент метода.

Получить_входящее_СМС //Получаем строку

Ссылочные данные – представляют собой ссылку на строковые или числовые данные, расположенные в другом месте профиля. Ссылка определяется названием ссылки и следующим за ним двоеточием. Ссылка объявляется перед теми данными, на которые она должна ссылаться. При использовании ссылки пишется её имя без двоеточия. При выполнении обработчика события, извлекаются данные, на которые указывает ссылка.

Пример:

Ссылка_на_строку: “Строка” //Объявление ссылки на строку

Ссылка_на_число: 123 //Объявление ссылки на число

СОБЫТИЕ_1

Ссылка_на_строку //Использование – текст “Строка”

//Получаем строку «Строка»

Ссылка_на_число //Использование – число 123

//Получаем число 123

Конец_события

Объявление ссылочных переменных следует производить не в обработчиках событий, а перед ними.

4. Операторы.

Для выполнения математических и логических операций с данными используются операторы. Все операторы сведены в таблицу.

Обозначение	Описание
+	Оператор математического сложения
-	Оператор математического вычитания
&	Оператор логическое И
	Оператор логическое ИЛИ
!	Оператор логическое НЕ
^	Оператор логическое исключ. ИЛИ
*	Оператор математического умножения
/	Оператор математического деления
>	Оператор сравнения на больше
<	Оператор сравнения на меньше
?	Оператор сравнения на равенство
()	Оператор изменения порядка вычислений
Если_верно Иначе Конец_если	Оператор условного выполнения

Математические и логические операторы должны располагаться между данными над которыми они производят операции.

Пример:

1+2 //Результат =3
6+5-1*2 //Результат =20

Результат оператора сравнения =1, если сравнение верно, и =0, если не верно.

Пример:

4>1 //Результат =1
7<10 //Результат =0

Все операции имеют одинаковый приоритет – исполняются последовательно, для изменения порядка вычислений используется оператор скобки ().

Пример:

6+5-1*2 //Результат =20
6+5-(1*2) //Результат =9

После выполнения операций над данными получаем результат выполнения операций.

Оператор условного выполнения используется для исполнения действий в зависимости от результата предыдущих операций. Формат этого оператора следующий:

argument

Если_верно

[список действий выполняемых, если argument не равен 0]

Иначе

[список действий выполняемых, если argument равен 0]

Конец_если

argument – предыдущий результат (может быть просто число)

Пример:

Получить_напряжение_на_входе > 50

Если_верно

Выход_выключить

Иначе

Выход_включить

Конец_если

Макрос Иначе может отсутствовать, если в нём нет необходимости.

Пример:

Получить_напряжение_на_входе > 50

Если_верно

Выход_выключить

Конец_если

Операторы Если_верно могут быть вложенными.

Пример:

Получить_напряжение_на_входе > 50

Если_верно

x1>10

Если_верно

```
        Выход_выключить
Иначе
        Выход_включить
Конiec_если
Конiec_если
```

5. Определение макросов и их использование.

Макрос это обозначение какого-либо выражения другим названием. Формат объявления нового макроса следующий:

```
Имя_макроса {Тело макроса}
```

Пользователем объявление макросов может быть использовано для следующих целей: обозначение названий методов механизмов и событий устройства своими названиями соответствующими конкретной задаче, обозначение некоторого набора действий одним названием. Использование макросов сделает текст профиля более понятным и читаемым.

Пример:

```
СОБЫТИЕ_Открыта_входная_дверь { СОБЫТИЕ_Сработал_вход_1 }
СОБЫТИЕ_Открыто_окно { Сработал_вход_2 }
```

```
Включить_сирену_на_10_секунд
{
1 Выход_выбрать
10 Выход_время_включения
}
```

```
СОБЫТИЕ_Открыто_окно
        Включить_сирену_на_10_секунд
Конiec_события
```

```
СОБЫТИЕ_Открыта_входная_дверь
        Включить_сирену_на_10_секунд
Конiec_события
```

Все объявляемые макросы разделяются по группам. Макросы, которые объявляет пользователь, попадают в группу Пользовательская. Для создания новой группы и размещения в ней своих макросов используется оператор Группа.

```
Группа Название_группы
```

Если группа с указанным названием уже существует, то новая группа не создается, а макросы объявляемые после указания существующей группы будут заноситься в эту группу.

Пример:

```
Группа Мои_макросы //Будет создана группа макросов Мои_макросы
Макрос1 {.....} //Макросы попадут в группу Мои_макросы
Макрос2 {.....}
```

```
Группа Выходы //Группа не будет создана – она уже есть
Включить_сирену {.....} //Макрос попадет в группу Выходы
```

6. Подключаемые файлы.

К основному файлу профиля могут подключаться другие файлы, содержащие, например, объявления макросов. Так в каждом профиле подключаются два файла: base.ev и Omega.dev эти файлы содержат объявления базовых макросов, а так же методов механизмов и событий устройства. Формат подключения файла следующий:

```
#F "Имя файла"
```

Пример:

```
#F "Мои макросы.ev"
```

Эта возможность может быть использована пользователем для того, чтобы разместить свои макросы в отдельном файле. Основной файл профиля при этом будет более понятным и читаемым.

7. Механизмы устройства.

Методы всех механизмов устройства разделены по группам. Каждому механизму соответствует своя группа. Их можно посмотреть в списке групп программы редактора. Методы механизмов могут иметь аргументы – входные значения. И могут возвращать результат – выходное значение. Далее будут рассмотрены все механизмы устройства и их методы и правила их использования.

Входы.

Входы – механизм объединяет в себе шесть входов устройства. Входы устройства аналоговые – они позволяют производить контроль напряжения или сопротивления. Если вход используется для контроля сопротивления, то на плате устройства должен быть включен соответствующий этому входу микропереключатель, подключающий к входу подтягивающий резистор сопротивлением 10 Ком к напряжению питания.

Для каждого входа можно задать диапазон допустимых значений напряжения или сопротивления при выходе за который, произойдет событие – срабатывание входа.

Для каждого из входов так же можно задать временные параметры работы входа.

Время активного уровня – это время, в течение которого сигнал на входе должен иметь значение, соответствующее уровню тревога, чтобы зафиксировалось срабатывание входа. Значение этого параметра по умолчанию 300 мс. Может задаваться от 0 до 25 секунд с дискретом 0,1 сек. Данный параметр служит для устранения реакции на короткие сигналы (устранениедребезга контактов).

Время неактивного уровня – это время, в течение которого сигнал на входе должен иметь значение, соответствующее уровню норма, чтобы разрешить повторные события срабатывания входа. По умолчанию данный параметр имеет значение 0. Может задаваться от 0 до 25 секунд с дискретом 0,1 сек. Этот временной интервал используется для исключения часто повторяющихся сигналов, например, с датчика движения.

Время паузы перед срабатыванием – это время паузы после фиксирования времени активного уровня сигнала соответствующего установленному перед срабатыванием входа. По умолчанию время паузы равно 0. Может изменяться от 0 до 650 сек с дискретом 0,1 сек. Применение – организация задержки после срабатывания датчика для снятия системы с охраны.

Для каждого из входов можно задать режим постоянного контроля. Если включен этот режим, то вход контролируется постоянно, если выключен, то только в режиме охраны.

Механизм Входы имеет следующие методы:

Выбрать_вход - выбор входа для дальнейшей работы с ним

Аргументы - номер выбираемого входа (1-6)

Результат - нет

Пример:

1 Выбрать_вход // Выбор входа 1

Установить_задержки_входа - установка временных параметров выбранного входа.

Аргументы - время активного уровня, время неактивного уровня, время паузы перед срабатыванием. (времена задаются в 0,1сек)

Результат - нет

Пример:

1 Выбрать_вход // Выбор входа 1
10 50 300 Установить_задержки_входа // Время активного уровня = 1 сек
// Время неактивного уровня = 5 сек
// Время паузы перед срабатыванием = 30 сек.

Установить_пороги_напряжения_входа - установка порогов напряжения срабатывания выбранного входа. (при контроле напряжения)

Аргументы - нижний порог, верхний порог (значения задаются в 0,1 вольт)

Результат - нет

Пример:

1 Выбрать_вход // Выбор входа 1
0 60 Установить_пороги_напряжения_входа // Пороги 0 вольт и 6 вольт

Установить_пороги_сопротивления_входа - установка порогов сопротивления срабатывания выбранного входа. (при контроле сопротивления)

Аргументы - нижний порог, верхний порог (значения задаются в 0,1 КОм)

Результат - нет

Пример:

1 Выбрать_вход // Выбор входа 1
40 70 Установить_пороги_сопротивления_входа // Пороги 4 Ком и 7 Ком

Получить_напряжение_на_входе - получение значения напряжения на выбранном входе.

Аргументы - нет

Результат - значение напряжения в 0,1вольт.

Пример:

1 Выбрать_вход // Выбор входа 1
Получить_напряжение_на_входе // Получим знач. Напряжения

Получить_сопротивление_на_входе - получение значения сопротивления на выбранном входе.

Аргументы - нет

Результат - значение сопротивления в 0,1КОм.

Пример:

1 Выбрать вход

// Выбор входа 1

Получить_сопротивление_на_входе

//Получим знач. сопротивления

Включить_вход - Включение контроля выбранного входа.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать вход

// Выбор входа 1

Включить_вход

// будет включен контроль входа

Выключить_вход - Выключение контроля выбранного входа.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать вход

// Выбор входа 1

Выключить_вход

// будет выключен контроль входа

Включить_режим_постоянного_контроля_входа – включение режима постоянного контроля выбранного входа.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать вход

// Выбор входа 1

Включить_режим_постоянного_контроля_входа

//включение пост.контроля

Выключить_режим_постоянного_контроля_входа - выключение режима постоянного контроля выбранного входа.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать вход

// Выбор входа 1

Выключить_режим_постоянного_контроля_входа

//выключение пост.контроля

Получить_состояние_входа_за_время – проверка того была ли тревога на выбранном входе в течение указанного времени (0-25 секунд).

Аргументы - значение интервала времени в 0,1 сек (0...255)

Результат - 0-норма(не было тревоги) 1-тревога

Пример:

1 Выбрать вход

// Выбор входа 1

200 Получить_состояние_входа_за_время

//проверка тревоги за последние 20 секунд

Механизм Входы вызывает следующие события:

СОБЫТИЕ_Сработал_вход_1	-	при срабатывании входа 1
СОБЫТИЕ_Сработал_вход_2	-	при срабатывании входа 2
СОБЫТИЕ_Сработал_вход_3	-	при срабатывании входа 3
СОБЫТИЕ_Сработал_вход_4	-	при срабатывании входа 4
СОБЫТИЕ_Сработал_вход_5	-	при срабатывании входа 5
СОБЫТИЕ_Сработал_вход_6	-	при срабатывании входа 6

7.2 Выходы.

Механизм выходы соответствует трём выходам устройства. Выходы построены по схеме открытый коллектор. Во включенном состоянии на выход подаётся минус источника питания, в выключенном состоянии на выход ничего не подаётся. Выходы обладают защитой от короткого замыкания, при обнаружении короткого замыкания все выходы отключаются до его исчезновения. Механизм Выходы предоставляет набор методов для включения/выключения выхода, получения его состояния, включения/выключения режима индикации состояния охраны, включения/выключения режима мигания. Включение выхода можно произвести с заданными временными параметрами: время включения и время паузы перед включением. Далее они подробно описаны:

Выход_выбрать - Выбор одного из выходов для дальнейшей работы с ним.

Аргументы - номер выбираемого выхода (1-3)

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

Выход_время_включения - Включение выбранного выхода на заданное время.

Аргументы - время включения в 0,1 сек

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

30 Выход_время_включения // Включение выхода на 3 секунды

Выход_время_включения_в_минутах - Включение выбранного выхода на заданное время.

Аргументы - время включения в минутах

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

30 Выход_время_включения_в_минутах // Включение выхода на 30 минут

Выход_время_включения_в_часах - Включение выбранного выхода на заданное время.

Аргументы - время включения в часах

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

30 Выход_время_включения_в_часах // Включение выхода на 30 часов

Выход_время_паузы_перед_включением – Установка времени паузы перед включением выбранного выхода.

Аргументы - время паузы в 0,1 сек

Результат - нет

Пример:

1 *Выбрать выход* // *Выбор выхода 1*

50 *Выход_время_паузы_перед_включением* // *Пауза перед включением 5 секунд*

30 *Выход_время_включения* // *Включение выхода на 3 секунды*

Выход_время_паузы_перед_включением_в_минутах – Установка времени паузы перед включением выбранного выхода.

Аргументы - время паузы в минутах

Результат - нет

Пример:

1 *Выбрать выход* // *Выбор выхода 1*

50 *Выход_время_паузы_перед_включением_в_минутах* // *Пауза перед включением 5 мин.*

30 *Выход_время_включения* // *Включение выхода на 3 секунды*

Выход_время_паузы_перед_включением_в_часах – Установка времени паузы перед включением выбранного выхода.

Аргументы - время паузы в часах

Результат - нет

Пример:

1 *Выбрать выход* // *Выбор выхода 1*

50 *Выход_время_паузы_перед_включением_в_часах* // *Пауза перед включением 5 часов.*

30 *Выход_время_включения* // *Включение выхода на 3 секунды*

Выход_включить - Включение выбранного выхода на неограниченное время.

Аргументы - нет

Результат - нет

Пример:

1 *Выбрать выход* // *Выбор выхода 1*

Выход_включить // *Включение*

Выход_выключить- Выключение выбранного выхода.

Аргументы - нет

Результат - нет

Пример:

1 *Выбрать выход* // *Выбор выхода 1*

Выход_выключить // *Выключение*

Выход_моргание_ВКЛ - Включение режима моргания.

Аргументы - нет

Результат - нет

Пример:

1 *Выбрать выход* // *Выбор выхода 1*

Выход_морганье_ВКЛ //Включение режима морганья

Выход_морганье_ВЫКЛ - Выключение режима морганья.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

Выход_морганье_ВЫКЛ //Выключение режима морганья

Выходы_Период_морганья - Установка периода морганья для всех выходов.

Аргументы - период морганья в 0,1 сек

Результат - нет

Пример:

10 Выход_Период_морганья //Период морганья = 1 секунда

Выходы_Длительность_импульса_морганья - Установка длительности импульса при моргании для всех выходов.

Аргументы - длительность импульса в 0,1 сек

Результат - нет

Пример:

2 Выходы_Длительность_импульса_морганья //длительность импульса = 0,2 секунды

Выход_индикация_режима_охраны_ВКЛ – включение режима индикации состояния охраны для выбранного выхода.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

Выход_индикация_режима_охраны_ВКЛ //выход1 – индикатор режима охраны

Выход_индикация_режима_охраны_ВЫКЛ – выключение режима индикации состояния охраны для выбранного выхода.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

Выход_индикация_режима_охраны_ВЫКЛ //выход1 – режим инд.охр. выключен

Выход_состояние - Получение состояния выбранного выхода. Возвращает 0, если выход выключен или не 0 если включен

Аргументы - нет

Результат - нет

Пример:

1 Выбрать выход // Выбор выхода 1

Выход_состояние // получим состояние выхода

Выход3 управляет светодиодом на боковой панели устройства. Он по умолчанию используется для индикации работы и настройки электронных ключей touch memory. Для выбора режима работы этого выхода используются следующие макросы

Выход3_обычный_режим - выход используется как обычный выход
Выход3_индикатор_TM - выход используется как индикатор работы интерфейса touch memory (по умолчанию)

Аргументы - нет
Результат - нет

Механизм Выходы не вызывает никаких событий.

7.3 Сеть1.

Механизм Сеть1 соответствует интерфейсу связи с внешними устройствами. Скорость обмена 9600 бит/сек. Имеется два выхода этого интерфейса RS и RS485. Одновременное подключение сразу к обоим выходам нежелательно. Выход интерфейса RS используется для связи с компьютером. Основное его назначение – запись профиля и файла звукового контейнера в устройство и обмен сообщениями с компьютером. Так же с помощью него могут объединяться устройства с таким же интерфейсом. Не предназначен для соединения большого количества устройств и устройств, находящихся на большом расстоянии друг от друга. Выход интерфейса RS-485 предназначен для обмена сообщениями между многими устройствами (до 32) и находящимися на большом расстоянии друг от друга.

Протокол интерфейса:

Допускает подключение к шине одновременно нескольких устройств до 32 (для выхода RS-485). Каждое устройство имеет свой адрес (0...31). Устройства могут обмениваться сообщениями, осылая его определённому устройству по его адресу или всем устройствам по общему адресу. Общим адресом является адрес=0.

Для пользователя предоставлена возможность обмена текстовыми сообщениями, длина передаваемых сообщений не должна превышать 21 символ. Подготавливать сообщения для отправки и анализировать принятые текстовые сообщения можно с помощью механизма Буффер_текста.

Сеть1_установить_свой_адрес - Установка своего адреса в сети1.

Аргументы - свой адрес 0...31
Результат - нет

Пример:

5 Сеть1_установить_свой_адрес // Устройство будет иметь адрес=5 для сети1

Сеть1_послать_сообщение - Отправка сообщения по адресу.

Аргументы - Сообщение (текст), адрес получателя
Результат - нет

Пример:

“Тревога” 2 Сеть1_послать_сообщение //устройству с адресом 2 отправить “Тревога”

Сеть1_послать_сообщение_всем - Отправка сообщения всем устройствам.

Аргументы - Сообщение (текст)
Результат - нет

Пример:

“Тревога” Сеть1_послать_сообщение_всем //Всем устройствам отправить “Тревога”

Сеть1_получить_входящее_сообщение - Получение входящего сообщения.

Аргументы - нет
Результат - Сообщение (текст)

Пример:

Сеть1_получить_входящее_сообщение //Получим принятое сообщение

!!! Пока не будет выполнен данный макрос, новые сообщения приниматься не будут.

Сеть1_получить_входящий_адрес - Получение адреса устройства, от которого было принято сообщение.

Аргументы - нет
Результат - адрес отправителя (0...31)

Пример:

Сеть1_получить_входящий_адрес //Получим адрес отправителя

Механизм Сеть1 вызывает следующее событие:

СОБЫТИЕ_Принято_Сообщение_по_RS_интерфейсу1 - при получении сообщения по сети1

7.4 Модем.

Механизм Модем позволяет использовать следующие функции GSM модуля:

- дозвон по списку абонентов
- отправка СМС по списку абонентов
- приём и отклонение входящих звонков
- приём входящих СМС
- звуковое оповещение при дозвоне
- отправка USSD запросов (запрос баланса)

Прежде чем использовать функцию дозвона, следует установить список номеров для оповещения и если требуется звуковую фразу, которая должна быть озвучена при соединении. Для отправки СМС так же требуется сначала установить список номеров для оповещения и текст СМС.

Методы механизма модем:

Установить_номера_для_оповещения - Установка списка номеров для дозвона и отправки СМС.

Аргументы - список номеров (номера в кавычках через запятую)
Результат - нет

Пример:

“+7111111111, +7222222222, +7333333333”

Установить_номера_для_оповещения // Установка трёх номеров для оповещения

!!! Если список номеров не изменяется то его настройку можно произвести только один раз, например, при включении питания

Установить_текст_СМС - Установка текста СМС

Аргументы - текст СМС (в кавычках)

Результат - нет

Пример:

“Это текст СМС”

Установить_текст_СМС // Установка текста СМС

Установить_звук_для_дозвона – задание фразы для озвучивания при дозвоне

Аргументы - ссылка на звуковую фразу (см. механизм Озвучивание)

Результат - нет

Пример:

Звук1: “Внимание” “Тревога” “”

Звук1 Установить_звук_для_дозвона

Произвести_дозвон - Дозвон по установленным номерам.

Аргументы - нет

Результат - нет

Пример:

“+7111111111, +7222222222, +7333333333”

Установить_номера_для_оповещения // Установка трёх номеров для оповещения

Произвести_дозвон //Дозвон на эти номера

Отправить_СМС - отправка установленного текста СМС на установленные номера.

Аргументы - нет

Результат - нет

Пример:

“+7111111111, +7222222222, +7333333333”

Установить_номера_для_оповещения // Установка трёх номеров для оповещения

“Это текст СМС”

Установить_текст_СМС // Установка текста СМС

Отправить_СМС //Отправка СМС

Принять_входящий_вызов - установление голосового соединения при входящем вызове.

Аргументы - нет

Результат - нет

Пример:

Принять_входящий_вызов //Приём вход. вызова

Прекратить_входящий_вызов - разрыв голосового соединения при входящем вызове.

Аргументы - нет

Результат - нет

Пример:

Прекратить_входящий_вызов //Отбой входящего вызова

Получить_входящее_СМС - получение текста последнего входящего СМС

Аргументы - нет

Результат - нет

Пример:

Получить_входящее_СМС // получим пришедшее СМС

!!! Для анализа текста СМС можно использовать механизм Буффер_текста

Получить_входящий_номер - получение телефонного номера последнего входящего вызова или входящей СМС.

Аргументы - нет

Результат - нет

Пример:

Получить_входящий_номер // получим номер входящего вызова или вх.СМС

Проверить_наличие_входящего_номера_в_списке – проверка присутствия телефонного номера входящего вызова или входящей СМС в списке номеров.

Аргументы - Список номеров (в кавычках через запятую)

Результат - =1 если номер есть в списке, =0 если нет в списке

Пример:

“+7111111111, +7222222222, +7333333333”

Проверить_наличие_входящего_номера_в_списке //получим результат поиска 1 или 0

!!! +7 и 8 в начале номера не участвуют в сравнении т.е. номера +7123456789 и 8123456789 считаются одинаковыми

Послать_USSD_запрос – отправка USSD запроса (запрос баланса сим карты).

Аргументы - Строка запроса

Результат - нет

Пример:

*“*100#” Послать_USSD_запрос*

При получении ответа на запрос произойдет событие:

СОБЫТИЕ_Входящее_USSD_сообщение

Получить_входящее_USSD_сообщение – получение ответа на USSD запрос.

Может использоваться только в обработчике события:

СОБЫТИЕ_Входящее_USSD_сообщение

Аргументы - нет
Результат - Строка – USSD сообщение
Пример:

Получить_время_голосового_соединения – получение продолжительности голосового соединения

Аргументы - нет
Результат - продолжительность соединения в 0,1сек (0...65535)

Пример:

Получить_время_голосового_соединения //Получим длительность соединения

Получить_мощность_GSM_сигнала – получение мощности сигнала GSM сети

Аргументы - нет
Результат - мощность сигнала в условных единицах (0...31)

Пример:

Получить_мощность_GSM_сигнала //Получим мощность GSM сигнала

Установить_максимальное_время_голосового_соединения – задание ограничения по времени голосового соединения

Аргументы - максимальное время соединения в 0,1 сек (0...65535)
Результат - нет

Пример:

300 Установить_максимальное_время_голосового_соединения //Ограничение 30 сек.

Включить_внешний_микрофон – Включение аудиовхода

Аргументы - нет
Результат - нет

Пример:

Включить_внешний_микрофон //Включение аудиовхода

Выключить_внешний_микрофон – Выключение аудиовхода

Аргументы - нет
Результат - нет

Пример:

Отключить_внешний_микрофон //Выключение аудиовхода

Механизм Модем вызывает следующие события:

СОБЫТИЕ_Входящий_звонок_начало	-	При начале входящего соединения
СОБЫТИЕ_Входящий_звонок_конец	-	При завершении входящего соединения
СОБЫТИЕ_Входящее_смс	-	При приёме входящей СМС
СОБЫТИЕ_Входящее_USSD_сообщение-	-	При приёме ответа на USSD запрос

7.5 Озвучивание.

Механизм Озвучивание позволяет управлять звуковым синтезатором Информатора. Звуки в памяти хранятся в виде слов, каждое из которых имеет своё название. Синтезатор может воспроизводить фразы, составленные из этих слов. Паузы между словами и между фразами могут настраиваться пользователем.

Методы механизма Озвучивание:

Воспроизвести_звук – Воспроизведение указанной фразы.

Аргументы	-	слово или ссылка на описание звуковой фразы
Результат	-	нет

Звуковая фраза представляет собой последовательность названий слов в кавычках и заканчивающаяся пустыми кавычками.

Примеры:

Звук1: "Внимание", "Тревога", ""

Звук1 Воспроизвести_звук

"Тревога" Воспроизвести_звук

Установить_количество_повторов_звука – Задание количества повторов озвучивания.

Аргументы	-	количество повторов
Результат	-	нет

Примеры:

3 Установить_количество_повторов_звука

"Тревога" Воспроизвести_звук //Произношение 3 раза

Установить_паузу_между_словами_звука – Задание паузы между словами.

Аргументы	-	длительность паузы в 0,1 сек
Результат	-	нет

Примеры:

5 Установить_паузу_между_словами_звука //Установка паузы 0,5 сек

Установить_паузу_между_фразами_звука – Задание паузы между словами.

Аргументы - длительность паузы в 0,1 сек
Результат - нет

Примеры:

15 Установить_паузу_между_фразами_звука //Установка паузы 1,5 сек

Механизм Озвучивание вызывает следующее событие:

СОБЫТИЕ_Окончание_озвучивания - при окончании озвучивания фразы

7.6 DTMF декодер.

Механизм DTMF декодер позволяет принимать DTMF сигналы, получаемые от другого телефона при голосовом соединении. DTMF сигналы могут быть вызваны нажатиями клавиш на управляющем телефоне в тоновом режиме или функцией быстрого набора. При приёме каждого DTMF сигнала запускается событие СОБЫТИЕ_DTMF_нажатие. Получить информацию о принятом сигнале можно двумя путями:

1. Получить код принятого нажатия, при этом нажатиям на цифровые клавиши соответствует соответствующее число, клавише # код 243, клавише * код 250.
2. Получить строку принятых нажатий, накопленных в буфере DTMF декодера. Размер буфера позволяет запоминать до 10-ти нажатий, при поступлении большего количества нажатий самое старое нажатие забывается. Имеется возможность очистить этот буфер.

Методы механизма DTMF декодер:

DTMF_получить_код_нажатия – получение цифрового кода последнего принятого DTMF сигнала

Аргументы - нет
Результат - код нажатия
Код 0...9 - нажатие 0...9,
Код 243 - нажатие #
Код 250 - нажатие *

Пример:

DTMF_получить_код_нажатия //Получим код последнего нажатия

DTMF_получить_строку_нажатий – получение содержимого буфера DTMF нажатий.

Аргументы - нет
Результат - строка DTMF нажатий

Пример:

*DTMF_получить_строку_нажатий //Получим строку
(пример: "12#" – получим при нажатии 1,2,#)*

DTMF_сбросить_строку_нажатий – очистка содержимого буфера DTMF нажатий.

Аргументы - нет
Результат - нет

Пример:
DTMF_сбросить_строку_нажатий

Механизм DTMF декодер вызывает следующее событие:

СОБЫТИЕ_DTMF_нажатие - При каждом принятом DTMF нажатии

7.7 Модем GPRS.

Механизм Модем_GPRS позволяет организовывать связь между устройствами, между устройствами и компьютерами через GPRS соединение по сети Internet. Для работы этой функции необходимо использование сервера (для передачи сообщений между устройствами по Internet). Для подключения к серверу устройство должно передать имя виртуальной сети, в которой оно желает обмениваться сообщениями и пароль виртуальной сети для доступа к ней. Обмен сообщениями возможен только между устройствами, подключенными к одной виртуальной сети. Он аналогичен обмену сообщениями через механизмы Сеть1 и Сеть2. Каждое устройство в виртуальной сети имеет свой адрес. Сообщения могут доставляться по адресу устройства или широковещательно.

Механизм Модем_GPRS функционирует только тогда, когда задана точка доступа (параметр GPRS соединения определяется оператором сотовой связи) и список IP адресов серверов. Список IP адресов может содержать один или несколько IP адресов. Устройство производит попытки подключения, начиная с первого адреса, и последовательно перебирает их при неудачной попытке соединения или при разрыве связи.

Методы механизма Модем_GPRS:

Установить_GPRS_точку_доступа - Задание точки доступа для GPRS соединения.

Аргументы - точка доступа (строка в кавычках)
Результат - нет

Пример:
“internet.mc”

Установить_GPRS_точку_доступа //Точка доступа internet.mc

Установить_GPRS_список_IP_адресов- Задание IP адресов серверов для GPRS соединения.

Аргументы - список IP адресов серверов (через запятую)
Результат - нет

Пример:
“212.111.0.1, 212.111.0.2”

Установить_GPRS_список_IP_адресов //задано два IP адреса

Установить_GPRS_пароль - Задание имени виртуальной сети и пароля.

Аргументы - имя виртуальной сети и пароль через пробел
Результат - нет

Пример:

"Net0001 password"

*Установить_GPRS_пароль //имя Вирт сети - Net0001
//пароль - password*

Установить_GPRS_адрес - Задание адреса устройства в виртуальной сети.

Аргументы - адрес устройства в виртуальной сети

Результат - нет

Пример:

10 Установить_GPRS_адрес //адрес устройства =10

Послать_GPRS_сообщение - отправка сообщения по адресу.

Аргументы - текст сообщения и адрес получателя

Результат - нет

Пример:

"Сообщение"

10 Послать_GPRS_сообщение //отправка сообщения по адресу 10

Получить_GPRS_сообщение - получение принятого сообщения

Аргументы - нет

Результат - текст сообщения

Пример:

Получить_GPRS_сообщение //принятое сообщение получим

Механизм Модем GPRS вызывает следующие события:

СОБЫТИЕ_Принято_Сообщение_по_GPRS - при приёме GPRS сообщения

7.8 Touch Memory.

Механизм Touch Memory содержит следующие функции:

- запрещение/разрешение запоминания электронных ключей.
- очистка памяти электронных ключей.
- получение номера электронного ключа.

Методы:

Запретить_запоминание_новых_эл_ключей – запрет добавления ключей.

Аргументы - нет

Результат - нет

Пример:

Запретить_запоминание_новых_эл_ключей //Новый ключи не могут быть добавлены

Запрет сохраняется после выключения питания.

Разрешить_запоминание_новых_эл_ключей – разрешение добавления ключей.

Аргументы - нет

Результат - нет

Пример:

Разрешить_запоминание_новых_эл_ключей // Новый ключи могут быть добавлены

Разрешение сохраняется после выключения питания.

Стереть_из_памяти_все_эл_ключи – забывание всех рабочих ключей.

Аргументы - нет

Результат - нет

Пример:

Стереть_из_памяти_все_эл_ключи // Все ключи будут забыты

Получить_номер_эл_ключа – получение строки номера электронного ключа.

Аргументы - нет

Результат - строка из 14 символов

Пример:

Получить_номер_эл_ключа // Получим строку номер ключа

// (пример: 00000CA6A56101)

Механизм Touch Memory вызывает следующее событие:

СОБЫТИЕ_Касание_ключом_TOUCH_MEMORY - при касании рабочим ключом

7.9 Система.

Механизм Система включает следующие функции:

- Управление режимом охраны и получение информации о нём.
- Управление таймерами для отсчёта временных интервалов.
- Контроль напряжения питания устройства.
- Управление режимом светодиода.
- Задание задержек в выполнении профиля.
- Получение номера версии Информатора.
- Изменение числовых и строковых данных в структуре профиля.

Режим охраны определяет работу входов. При включенном режиме охраны входы контролируются, при выключенном нет, за исключением тех, у которых включен режим постоянного контроля. Режим охраны может индцироваться выходами устройства, если для выхода включен режим индикации состояния охраны. Выход, используемый для индикации режима охраны, включен при включенной охране и выключен при выключенной охране, при постановке на охрану он мигает.

Методы для управления режимом охраны:

Включить_режим_охраны - Постановка на охрану через заданное время.

Аргументы - время постановки на охрану в 0,1 сек

Результат - нет

Пример:

50 Включить_режим_охраны //Постановка на охрану в течении 5 секунд

Выключить_режим_охраны - Снятие с охраны.

Аргументы - нет

Результат - нет

Пример:

Выключить_режим_охраны //Снятие с охраны

Изменить_режим_охраны - Постановка/снятие с охраны через заданное время.

Аргументы - время постановки на охрану в 0,1 сек

Результат - нет

Пример:

50 Изменить_режим_охраны //Постановка/снятие с охраны

!!! Если режим охраны был включен, то снятие происходит без задержки, если был выключен, то постановка производится с задержкой указанной в аргументе.

Состояние_режима_охраны - Получение информации о состоянии режима охраны.

Аргументы - нет

Результат - =1 если охрана вкл., =0 если охрана выкл,
>1 если постановка на охрану – время до постановки в 0,1 сек.

Пример:

Состояние_режима_охраны // получим состояние режима охраны

К режиму охраны относятся следующие события:

СОБЫТИЕ_Постановка_на_охрану - При постановке на охрану

СОБЫТИЕ_Снятие_с_охраны - При снятии с охраны

Таймеры служат для отсчёта временных интервалов от 0,1 сек до нескольких лет. Устройство содержит три таймера. По истечении установленного периода времени происходит событие срабатывания таймера, таймер при этом выключается. Для отсчёта следующего интервала времени нужно заново установить таймер. Таймер можно выключить и во время уже начатого отсчёта времени, другими словами остановить.

Методы для управления таймерами:

Выбрать_таймер - Выбор одного из пяти таймеров для последующей работы с ним.

Аргументы - номер таймера 1...5

Результат - нет

Пример:

1 Выбрать_таймер //выбор первого таймера

Установить_таймер - установка выбранного таймера на отсчёт временного интервала в 0,1 сек.

Аргументы - длительность временного интервала в 0,1 сек.

Результат - нет

Пример:

1 Выбрать_таймер //выбор первого таймера

100 Установить_таймер //установка таймера на отсчёт времени 10 секунд.

Установить_таймер_в_минутах - установка выбранного таймера на отсчёт временного интервала в минутах.

Аргументы - длительность временного интервала в минутах.

Результат - нет

Пример:

1 Выбрать_таймер //выбор первого таймера

100 Установить_таймер_в_минутах // отсчёт времени 100 минут.

Установить_таймер_в_часах - установка выбранного таймера на отсчёт временного интервала в часах.

Аргументы - длительность временного интервала в часах.

Результат - нет

Пример:

1 Выбрать_таймер //выбор первого таймера

100 Установить_таймер_в_часах // отсчёт времени 100 часов.

Выключить_таймер - Остановка выбранного таймера.

Аргументы - нет

Результат - нет

Пример:

1 Выбрать_таймер // выбор первого таймера

Выключить_таймер // останов первого таймера

Таймеры вызывают событие срабатывания:

СОБЫТИЕ_Сработал_таймер_1 - При истечении установленного времени

СОБЫТИЕ_Сработал_таймер_2 - При истечении установленного времени

СОБЫТИЕ_Сработал_таймер_3 - При истечении установленного времени

СОБЫТИЕ_Сработал_таймер_4 - При истечении установленного времени

СОБЫТИЕ_Сработал_таймер_5 - При истечении установленного времени

Контроль напряжения питания осуществляется следующим методом:

Получить_напряжение_питания - получение напряжения питания в 0,1 В.

Аргументы - нет
Результат - напряжение питания в 0,1 В.

Пример:

Получить_напряжение_питания // Получение напряжения питания

Для контроля напряжения питания есть два события:

СОБЫТИЕ_появление_питания - При появлении основного питания
СОБЫТИЕ_пропадание_питания - При пропадании основного питания

Задание задержек в выполнении профиля

– может потребоваться когда нужно выдержать небольшую паузу между выполняемыми действиями в одном обработчике события. Например, если требуется отправить несколько сообщений подряд по интерфейсу RS-485, то перед отправкой каждого следующего сообщения нужно выдержать небольшую паузу = 0,3 сек.

!!! Однако следует всегда помнить, что при организации задержек таким образом выполнение других событий в это время приостанавливается(они обработаются когда будет обработан текущий обработчик).

Метод:

Пауза – задание задержки в выполнении профиля.

Аргументы - длительность задержки в 0,01 сек (0...2,55сек)
Результат - нет

Пример:

10 Пауза // Пауза 0,1 секунда

Вызов события.

Вызвать_событие – Принудительный вызов события по его номеру

Аргументы - Номер вызываемого события (из группы Номера_Событий)
Результат - нет

Пример:

N_Событие_пользователя_1 Вызвать_событие

!!! Вызванное событие будет обработано после завершения обработки текущего события.

Получение номера версии Информатора.

Получить_номер_версии – получение номера версии

Аргументы - нет
Результат - номер версии (0...65535)

Пример:

Получить_номер_версии // Получим номер версии Информатора

Изменение числовых и строковых данных в структуре профиля. Эта возможность главным образом предусмотрена для изменения настроечных данных профиля без его перепрограммирования. Например, могут изменяться тексты СМС, номера для дозвона, значения разных временных задержек и т.д.

!!! не следует использовать эту функцию слишком часто. Т.к. при более чем 10000 раз использования сохранность содержимого памяти Информатора не гарантируется.

Методы для изменения параметров:

Выбрать_изменяемый_параметр – Указание на изменяемый параметр

Аргументы - ссылка на параметр строковой или числовой параметр
Результат - 0

Установить_значение_параметра – Изменение строкового или числового параметра

Аргументы - изменяемое значение число или строка
Результат - нет

Примеры:

Параметр_строка: “Первоначальное значение ”

Параметр_число: 1

*Параметр_строка Выбрать_изменяемый_параметр
“Новое значение” Установить_значение_параметра*

*Параметр_число Выбрать_изменяемый_параметр
2 Установить_значение_параметра*

!!! Примечание: Если размер новой строки превышает размер первоначальной строки (которая задавалась в профиле), то новая строка будет обрезана под первоначальный размер. Если в профиле числовой параметр был задан меньше чем 256, то он может быть изменён лишь на число в диапазоне 0...255.

Если планируется изменение некоторого параметра, то его размер в профиле должен соответствовать максимально допустимому размеру этого параметра. Строка должна быть дополнена пробелами до требуемой длины, а число иметь значение больше 255.

7.10 Память.

Механизм Память содержит шестнадцать ячеек, которые могут использоваться для хранения значений параметров или для хранения результатов промежуточных вычислений.

Методы:

x1, x2, x3...x16 - Получение значения ячейки памяти 1,2,3...16

Аргументы - нет

Результат - числовое значение памяти (0...65535)

Примеры:

*(x1+2)*x2 //Выражение использующее значения из памяти*

Запомнить_x1, Запомнить_x2, Запомнить_x3 ... Запомнить_x16 - Запоминание значения в ячейке памяти 1,2,3...16

Аргументы - запоминаемое значение (0...65535)

Результат - нет

Примеры:

10 Запомнить_x1 // В ячейку памяти x1 заносим число 10

x2+1 Запомнить_x2 // Значение ячейки памяти x2 увеличиваем на 1 и запоминаем

7.11 Буфер текста.

Механизм Буфер текста предназначен для обработки текстовой информации. С его помощью возможно создание строки из нескольких подстрок, добавление числа к строке, поиск в строке подстроки, получение из буфера подстроки, получение из буфера числа.

Добавление в текстовый буфер производится прибавлением добавляемого текста в конец того, что уже есть в буфере. Поиск осуществляется с позиции поиска. Позиция поиска обнуляется при очистке буфера или специальным методом. Если в процессе поиска искомый фрагмент найден то позиция поиска устанавливается после этого фрагмента. Получение подстроки и числа из буфера происходит с позиции поиска.

Методы механизма Текстовый буфер:

Очистить_текстовый_буфер - Очистка текстового буфера, сбрасывается позиция поиска.

Аргументы - нет

Результат - нет

Пример:

Очистить_текстовый_буфер

Поместить_текст_в_текстовый_буфер - Установка текста в буфер, сбрасывается позиция поиска.

Аргументы - текст помещаемый в буфер

Результат - нет

Пример:

“Текст” Поместить_текст_в_текстовый_буфер// Буфер текста содержит “Текст”

Добавить_строку_в_текстовый_буфер - Добавление строки в буфер текста.

Аргументы - текст добавляемый в буфер

Результат - нет

Пример:

“Текст” Поместить_текст_в_текстовый_буфер// Буфер текста содержит “Текст”

“ строка” Добавить_строку_в_текстовый_буфер // содержит “Текст строка”

Добавить_число_в_текстовый_буфер - Добавление числа в буфер текста.

Аргументы - число добавляемое в буфер

Результат - нет

Пример:

“Текст” Поместить_текст_в_текстовый_буфер // Буфер содержит “Текст”

123 Добавить_число_в_текстовый_буфер // содержит “Текст 123”

Сбросить_позицию_поиска_в_текстовом_буфере - сброс позиции поиска.

Аргументы - нет

Результат - нет

Пример:

Сбросить_позицию_поиска_в_текстовом_буфере

Найти_строку_в_текстовом_буфере - поиск строки с позиции поиска.

Аргументы - искомая строка

Результат - =1 если строка найдена (поз.поиска уст. после найденной строки)

=0 если не найдена (поз.поиска не изменяется)

Пример:

Сбросить_позицию_поиска_в_текстовом_буфере

“пароль=”

Найти_строку_в_текстовом_буфере //Ищем в буфере “пароль=”

Получить_слово_из_текстового_буфера_до_символа – извлечение текста с позиции поиска до заданного символа.

Аргументы - символ до которого извлекается текст (в кавычках)

Результат - получаем фрагмент текста буфера с позиции поиска до заданного символа

Пример:

//Пусть в буфере текста находится “Пароль=passw”

Сбросить_позицию_поиска_в_текстовом_буфере

“пароль=”

Найти_строку_в_текстовом_буфере

//Ищем в буфере “пароль=”

“ ” Получить_слово_из_текстового_буфера_до_символа // Получаем слово до пробела

// В итоге получим пароль – “passw”

!!! Если нужно получить текст до конца буфера текста, то в качестве символа конца нужно указать пустые кавычки "".

Получить_число_из_текстового_буфера – получение числа с позиции поиска.

Аргументы - нет
Результат - число (если с позиции поиска находится не число то результат = 0)

Пример:

//Пусть в буфере текста находится "Пароль=123"

Сбросить_позицию_поиска_в_текстовом_буфере

"пароль="

Найти_строку_в_текстовом_буфере

// Ищем в буфере "пароль="

Получить_число_из_текстового_буфера

// Получаем число

// В итоге получим пароль – число 123

Извлечь_текст_из_текстового_буфера – получение всего содержимого буфера текста.

Аргументы - нет
Результат - текст Буфера

Пример:

Извлечь_текст_из_текстового_буфера

Механизм Буфер текста не вызывает никаких событий.

8. События устройства.

Здесь приведён полный список событий устройства:

СОБЫТИЕ_Включение_питания	-	При включении питания
СОБЫТИЕ_Сработал_вход_1	-	При срабатывании входа 1
СОБЫТИЕ_Сработал_вход_2	-	При срабатывании входа 2
СОБЫТИЕ_Сработал_вход_3	-	При срабатывании входа 3
СОБЫТИЕ_Сработал_вход_4	-	При срабатывании входа 4
СОБЫТИЕ_Сработал_вход_5	-	При срабатывании входа 5
СОБЫТИЕ_Сработал_вход_6	-	При срабатывании входа 6
СОБЫТИЕ_Касание_ключом_TOUCH_MEMORY	-	Касание Touch memory
СОБЫТИЕ_Обнаружение_GSM_сети	-	Регистрация в GSM сети
СОБЫТИЕ_Пропадание_GSM_сети	-	Пропадание сигнала GSM сети
СОБЫТИЕ_Соединение_при_дозвоне	-	Установлении соединения при дозвоне
СОБЫТИЕ_DTMF_нажатие	-	При приёме DTMF сигнала
СОБЫТИЕ_Принято_Сообщение_по_RS_интерфейсу1	-	принято сообщение по сети1
СОБЫТИЕ_Входящий_звонок_начало	-	Входящий вызов
СОБЫТИЕ_Входящий_звонок_конец	-	Конец входящего соединения
СОБЫТИЕ_Входящее_смс	-	При приёме СМС
СОБЫТИЕ_Постановка_на_охрану	-	При постановке на охрану
СОБЫТИЕ_Снятие_с_охраны	-	При снятии с охраны
СОБЫТИЕ_Смена_программы	-	При смене профиля
СОБЫТИЕ_Сработал_таймер_1	-	Срабатывание таймера 1
СОБЫТИЕ_Сработал_таймер_2	-	Срабатывание таймера 2
СОБЫТИЕ_Сработал_таймер_3	-	Срабатывание таймера 3
СОБЫТИЕ_Сработал_таймер_4	-	Срабатывание таймера 4

СОБЫТИЕ_Сработал_таймер_5	-	Срабатывание таймера 5
СОБЫТИЕ_появление_питания	-	При появлении основного питания
СОБЫТИЕ_пропадание_питания	-	При пропадании основного питания
СОБЫТИЕ_Окончание_озвучивания	-	При окончании озвучивания фразы
СОБЫТИЕ_Принято_Сообщение_по_GPRS	-	При получении сообщения
СОБЫТИЕ_Входящее_USSD_сообщение	-	При получении ответа на USSD запрос

9. Примеры обработчиков событий.

Пример 1. Обработчик события включение питания.

*//Производится настройка входов и выходов, установка номеров для оповещения, адресов
//сетевых интерфейсов, параметров GPRS соединения.*

СОБЫТИЕ_Включение_питания

5 Выбрать_вход //5 вход – режим постоянного контроля
 Включить_режим_постоянного_контроля_входа
 4 Выход_выбрать
 Выход_индикация_режима_охраны_ВКЛ // 4 вых. индикатор режима охраны

“+712345678, +712345679”
 Установить_номера_для_оповещения

1 Сеть1_установить_свой_адрес //Адрес в сети1 =1
 1 Сеть2_установить_свой_адрес //Адрес в сети2 =1

“internet.mc” Установить_GPRS_точку_доступа
 “200.100.0.1” Установить_GPRS_список_IP_адресов
 “Му_Net 123” Установить_GPRS_пароль

Конец_события

Пример 2. Обработчик события срабатывания входа.

*//Начинается дозвон, отправляется СМС, отправляется сообщение компьютеру
//по RS-интерфейсу, отправляется сообщение по GPRS, включается выход на 10 сек.*

Текст_сообщения_при_тревоге: “Тревога!!!” //Далее используем эти данные по ссылке
 Звук_Тревога_входа1: “Внимание” ”Тревога” “Вход” “Номер” “Один” “”

СОБЫТИЕ_Сработал_вход_1

“+712345678, +712345679” Установить_номера_для_оповещения
 Текст_сообщения_при_тревоге Установить_текст_СМС
 Отправить_СМС
 Звук_Тревога_входа1 Установить_звук_для_дозвона
 Произвести_дозвон
 Текст_сообщения_при_тревоге Сеть1 послать_сообщение_всем
 Текст_сообщения_при_тревоге 0 Послать_GPRS_сообщение //Широковещательно
 1 Выход_выбрать
 100 Выход_время_включения

Конец_события

Пример 3. Обработчик события при входящем вызове.

/ Номер с которого произошёл вызов проверяется на наличие доступа, если доступ разрешён то устанавливается соединение, если не разрешён то происходит отбой вызова, а на определённый номер посылается СМС с сообщением о попытке несанкционированного доступа и номером с которого был звонок*/*

*Список_разрешённых_номеров:
"+7111111111, +7222222222, +7333333333"
Номер_для_сообщений: "+7111111111"*

*СОБЫТИЕ_Входящий_звонок_начало
Список_разрешённых_номеров
Проверить_наличие_входящего_номера_в_списке
Если_верно
Принять_входящий_вызов
Иначе
Прекратить_входящий_вызов
Номер_для_сообщений_Установить_номера_для_оповещения
"Была попытка доступа с номера: "
Поместить_текст_в_текстовый_буффер
Получить_входящий_номер
Добавить_строку_в_текстовый_буффер
Извлечь_текст_из_текстового_буффера
Установить_текст_СМС
Отправить_СМС
Конец_если
Конец_события*

Пример 4. Обработчик события при входящем СМС.

/ Номер с которого пришло СМС проверяется на наличие доступа, если доступ разрешён то проверяется наличие в СМС пароля, если пароль верный то происходит поиск команды управления выходом. Если доступа нет то на номер с которого получено СМС посылается ответное СМС с определённым текстом*/*

*Список_разрешённых_номеров:
"+7111111111, +7222222222, +7333333333"
Пароль: "АБВГДЕ"
Команда_ВКЛ: "Вкл"
СОБЫТИЕ_Входящее_смс
Список_разрешённых_номеров
Проверить_наличие_входящего_номера_в_списке
Если_верно
Получить_входящее_СМС
Поместить_текст_в_текстовый_буффер
Пароль_Найти_строку_в_текстовом_буффере
Если_верно*

Команда_ВКЛ Нйти_строку_в_текстовом_буфере
Если_верно
1 Выход_выбрать
100 Выход_время_включения
Конец_если
Конец_события
Конец_если
Конец_если

Получить_входящий_номер Установить_номера_для_оповещения
"В_доступе_отказано" Установить_текст_СМС
Отправить_СМС

Конец_события

Пример 5. Постановка/снятие с охраны с помощью Touch memory.

СОБЫТИЕ_Касание_ключом TOUCH_MEMORY
100 Изменить_режим_охраны //Время_постановки_10_секунд
Конец_события

Пример 6. Постановка/снятие с охраны с помощью входа.

// Для входа должен быть включен режим постоянного контроля

СОБЫТИЕ_Сработал_вход_5
100 Изменить_режим_охраны //Время_постановки_10_секунд
Конец_события

Пример 7. Оповещение о постановке/снятии с охраны с помощью СМС GPRS и выдача сообщения на ПК.

СОБЫТИЕ_Постановка_на_охрану
"+712345678" Установить_номера_для_оповещения
"Постановка_на_охрану" Установить_текст_СМС
Отправить_СМС
0 Послать_GPRS_сообщение //Широковещательно
Сеть1_послать_сообщение_всем
Конец_события

СОБЫТИЕ_Снятие_с_охраны
"+712345678" Установить_номера_для_оповещения
"Снятие_с_охраны" Установить_текст_СМС
Отправить_СМС
0 Послать_GPRS_сообщение //Широковещательно
Сеть1_послать_сообщение_всем

Конец_события

Пример 8. Оповещение при приёме по RS интерфейсу сообщения о тревоге.

*/*Оповещение производится через дозвон СМС и сообщением на ПК*/*

Текст_сообщения_при_тревоге: "Тревога"

СОБЫТИЕ_Принято_Сообщение_по_Сети1
Сеть1_получить_входящее_сообщение
Поместить_текст_в_текстовый_буффер
"Тр" Найти_строку_в_текстовом_буффере
Если_верно
 "+712345678, +712345679" Установить_номера_для_оповещения
 Текст_сообщения_при_тревоге Установить_текст_СМС
 Отправить_СМС
 Произвести_дозвон
 Сеть1_послать_сообщение_всем
Конец_если
Конец_события

Пример 9. Передача принятого GPRS сообщения по Сети1 на ПК.

СОБЫТИЕ_Принято_Сообщение_по_GPRS
Получить_GPRS_сообщение
Сеть1_послать_сообщение_всем
Конец_события

Пример 10. Передача по СМС напряжения на входе 1 каждый час.

СОБЫТИЕ_Сработал_таймер_1
1 Выбрать_таймер
1 Установить_таймер_в_часах // Устанавливаем следующий период 1 час
"Напр.на входе1=" Поместить_текст_в_текстовый_буффер
1 Вход_выбрать
Получить_напряжение_на_входе_1 Записать_число_в_ячейку_памяти
1 Считать_число_из_ячейки_памяти /10 //Целая часть напряжения
Добавить_число_в_текстовый_буффер
","Добавить_строку_в_текстовый_буффер //Запятая между целой и дробн.ч.
1 Считать_число_из_ячейки_памяти %10 //Десятая часть напряжения
Добавить_число_в_текстовый_буффер
" В." Добавить_строку_в_текстовый_буффер
Извлечь_текст_из_текстового_буффера

Установить_текст_СМС
"+712345678" Установить_номера_для_оповещения
Отправить_СМС

Конец_события

//В обработчике события включения питания следует первоначально запустить таймер:

// 1 Выбрать_таймер

// 1 Установить_таймер // Устанавливаем на срабатывание через 0,1 сек

WWW.PRO
ПРОДАЙНИК
E-RU.S.RU