

# SirenaBT5

Описание протокола обмена

## Принцип работы

Протокол обмена SirenaBT5 аналогичен протоколу **AT**. Формат данных текстовой, каждая строка заканчивается символом **CRLF**. Канал связи работает в полном дуплексе. Некоторые ответы могут приходить без предварительного запроса командой. Команды требовательны к регистру символов. Канал связи Bluetooth профиль SPP-C (виртуальный COM порт). Команды можно отправлять подряд, не дожидаясь ответа, буфер составляет 2048 байт.

Формат команды:

[команда][CRLF]

[команда]=[аргументы][CRLF]

Несколько аргументов разделяются символом ;

Формат ответа:

+ [команда]:[ответ][CRLF]

Ответ может содержать:

- ключевые слова: **OK, ERROR, BUSY, READY, NULL, TRUE, FALSE**
- аргументы разделённые символом ;
- ключевое слово совместно с аргументом
- **JSON**

### Список команд

- SPEAK
- PLAY\*
- PLAYSTOP\*
- LIST\*
- GETMELODY\*
- SETMELODY\*
- DATAMELODY\*
- ENDMELODY\*
- DELETEMELODY\*
- CLEARMELODY\*
- GETEVENT\*
- SETEVENT\*
- DELETEEVENT\*
- CLEAREVENT\*
- SAVEFS\*
- FS
- INFO
- GETCONFIG
- SETCONFIG
- SETBUTTON
- SETMEASURE
- SETPRESSMAX
- SETIDLEMAX
- SETDEVIATION
- PINCODE
- SAVECONFIG

\*Команды зависят от состояния файловой системы

### Список асинхронных ответов

- +FS
- +EVENT
- +PLAYSTOP

## Команда SPEAK

Воспроизведение потокового аудио

Формат потока WAVE 8kHz 8bit, закодирован для передачи в WAVE7.

Команда записывает семплы в буфер и тут же начинает воспроизведение, воспроизведение продолжается до опустошения буфера. В ответе приходит количество записанных семплов в буфер, если количество записанных семплов меньше, чем было отправлено, то необходимо повторно отправить не записанные семплы. Порядок семплов слева на право.

Рекомендуется отправлять семплы пакетами по 1024 байт, размер пакета считается в незакодированном виде.

Отправка производится последовательно запрос/ответ, до тех пор, пока не закончатся аудио данные.

Запрос:

SPEAK=<семплы в кодировке WAVE7>

Ответ:

+SPEAK: {"writed":<записано семплов>}

## Команда PLAY

Воспроизведение записанной мелодии

Доступные индексы узнаются командой LIST.

Ответ приходит при начале воспроизведения и конце воспроизведения.

Ответ +PLAYSTOP приходит без запроса.

Запрос:

PLAY =<index>

Ответ:

+PLAY: {"index":<index>}

+PLAYSTOP: {"index":<index>}

## Команда PLAYSTOP

Остановить воспроизведение мелодии

Останавливает воспроизведение мелодии.

Если воспроизведения не было, то ответ с числом мелодии не придёт.

Запрос:

PLAYSTOP

Ответ:

+PLAYSTOP:OK

+PLAYSTOP: {"index":<index>}

## Команда LIST

Запрос таблиц мелодий и событий

Возвращает массивы индексов мелодий, настроенных событий и максимальный размер таблиц.

Если файловая система не инициализирована, возвращает NULL.

Запрос:

LIST

Ответ:

+LIST:NULL

или

+LIST: {"melody":[],"event":[],"melodyMax":40,"eventMax":30}



## Команда GETMELODY

Получить информацию о мелодии

Доступные индексы узнаются командой LIST.

Возвращает название мелодии в формате UTF-8 в кодировке BASE64.

Возвращает null, в случае, если мелодия с этим индексом не существует.

Запрос:

GETMELODY=<index>

Ответ:

+GETMELODY:{"index":<index>,"name": "<base64>"}

или

+GETMELODY:{"index":<index>,"name":null}

**Команда SETMELODY**

Начать запись мелодии

Запрос на начало записи мелодии, параметры:

length	Размер аудио данных от 1 до 1Мб
samples	Дискретизация от 8 до 48kHz
base64	Название в UTF-8 не более 512 байт, кодировка BASE64

Возвращает индекс мелодии или ошибку:

PARAM	Ошибка в параметрах
BUSY	Уже был запрос на запись
FULL	Нет места в таблице
IO	Сбой ввода/вывода

Запрос:

SETMELODY=&lt;length&gt;;&lt;samples&gt;;&lt;base64&gt;

Ответ:

+SETMELODY:{"index":&lt;index&gt;}

или

+SETMELODY:ERROR;&lt;error&gt;

## Команда DATAMELODY

Отправка пакетов данных мелодии

Отправка данных осуществляется пакетами по 512 байт, последний пакет не обязан быть 512 байт. Данные кодируются в BASE64.

Ответ ОК означает успешное завершение записи данных, ждёт следующий пакет. END означает конец приёма.

Возможные ошибки:

DATA	Данные более 512 байт или ещё не завершилась прошлая операция
WRITE	Ошибка записи, сбой ввода/вывода

Запрос:

DATAMELODY=<base64>

Ответ:

+DATAMELODY:OK

или

+DATAMELODY:END

или

+DATAMELODY:ERROR;<error>

**Команда ENDMELODY**

Завершить запись мелодии

Закрывает поток записи мелодии, если поток не был открыт, возвращает FALSE, иначе TRUE.

Запрос:

ENDMELODY

Ответ:

+ENDMELODY:TRUE

или

+ENDMELODY:FALSE

**Команда DELETEMELODY**

Удалить мелодию

Запрос:

DELETEMELODY=<index>

Ответ:

+DELETEMELODY:{"index":<index>,"delete":<bool>}

## Команда CLEARMELODY

Очистить таблицу мелодий

Запрос:

CLEARMELODY

Ответ:

+CLEARMELODY:OK

## Команда GETEVENT

Получить информацию о событии

Тип null, означает событие отсутствует.

Запрос:

GETEVENT=<index>

Ответ:

+GETEVENT:<json>

JSON:

index	0	
type	null, cyclic, press, pulse	
chIndex	1	
melodyIndex	-1	
poweroff	<bool>	
pulse	[{}]	До 3х объектов

pulse:

width	100
count	1

## Команда SETEVENT

Установить или изменить событие

В отличие от SETMELODY, в этой команде контроль над свободными индексами должен производиться в программе.

Параметры:

index	Индекс события
melodyIndex	Индекс мелодии (-1 мелодия не выбрана)
poweroff	0 – 1
chIndex	1 – 3
type	null, pulse, cyclic, 1 – 3  null аналогичен DELETEEVENT  если число, то задаются типы width и count
width	Ширина импульса
count	Количество импульсов

Запрос:

```
SETEVENT=<index>;<melodyIndex>;<poweroff>;<chIndex>;<type>;<
width>;<count>;[...]]
```

Ответ:

```
+SETEVENT: {"index":<index>,"change":<bool>}
```

или

```
+SETEVENT:ERROR;<error>
```



**Команда DELETEEVENT**

Сбросить событие

Запрос:

DELETEEVENT=<index>

Ответ:

+DELETEEVENT:{"index":<index>,"delete":<bool>}

## Команда CLEAREVENT

Очистить таблицу событий

Запрос:

CLEAREVENT

Ответ:

+CLEAREVENT:OK

## Команда SAVEFS

Сохранить таблицы мелодий и событий

Возвращает, TRUE было сохранено, FALSE не удалось сохранить, BUSY файловая система не инициализирована.

Запрос:

SAVEFS

Ответ:

+SAVEFS:TRUE

или

+SAVEFS:FALSE

или

+SAVEFS:BUSY

## Команда FS

Состояние файловой системы

READY файловая система готова к операциям ввода/вывода.

BUSY ошибка инициализации файловой системы.

Ответ со значением BUSY может придти без запроса.

Запрос:

FS

Ответ:

+FS:READY

или

+FS:BUSY

## Команда INFO

Запрос состояния устройства

Запрос:

INFO

Ответ:

+INFO:<json>

JSON:

fsReady	<bool>	Аналог команды FS
version	5.0	Версия программы
build	1000	Номер сборки
pincode	"0000"	Аналог команды PINCODE

**Команда GETCONFIG**

Запрос конфигурации

Настройки 3х каналов в JSON.

Запрос:

GETCONFIG

Ответ:

+GETCONFIG:[{}],{}]

Объект:

pressMax	<1 - 2000>
----------	------------

idleMax	<1 - 2000>
---------	------------

pulsesMax	<1 - 20>
-----------	----------

deviation	<1 - 50>
-----------	----------

button	<bool>
--------	--------

measure	<bool>
---------	--------

**Команда SETCONFIG**

Установка конфигурации

Индекс это номер канала -1, то есть счет с 0.

Значения BOOL задаются цифрами 0-1.

Запрос:

```
SETCONFIG=<index>;<pressMax>;<idleMax>;<pulsesMax>;<deviation>;<button>;<measure>
```

Ответ:

```
+SETCONFIG:{"index":<index>}
```

## Команда SETBUTTON

Установка частичной конфигурации

Запрос:

SETBUTTON=<index>;<button>

Ответ:

+SETBUTTON:{"index":<index>}



**Команда SETMEASURE**

Установка частичной конфигурации

Запрос:

SETMEASURE=<index>;<measure>

Ответ:

+SETMEASURE:{"index":<index>}

**Команда SETPRESSMAX**

Установка частичной конфигурации

Запрос:

SETPRESSMAX=<index>;<pressMax>

Ответ:

+SETPRESSMAX:{"index":<index>}

**Команда SETIDLEMAX**

Установка частичной конфигурации

Запрос:

SETIDLEMAX=<index>;<idleMax>

Ответ:

+SETIDLEMAX:{"index":<index>}

**Команда SETPULSES MAX**

Установка частичной конфигурации

Запрос:

SETPULSES MAX=<index>;<pulsesMax>

Ответ:

+SETPULSES MAX:{"index":<index>}

## Команда SETDEVIATION

Установка частичной конфигурации

Запрос:

SETDEVIATION=<index>;<deviation>

Ответ:

+SETDEVIATION:{"index":<index>}

## Команда PINCODE

Запрос и установка пин-кода

Пин-код фиксированной длины 4 символа, символы только числа 0 – 9.

Запрос:

PINCODE

Ответ:

+PINCODE:0000

Запрос:

PINCODE=0000

Ответ:

+PINCODE:OK

**Команда SAVECONFIG**

Сохранение конфигураций

TRUE конфигурация сохранена, FALSE ошибка сохранения.

Запрос:

SAVECONFIG

Ответ:

+SAVECONFIG:TRUE

или

+SAVECONFIG:FALSE

**Ответ +EVENT**

Произошло событие на канале

Ответ:

+EVENT:<json>

JSON:

index	1 – 3	канал
type	pulse, press, release, cyclic, measure, errMeasure	
pulse	[{}]	до 3х объектов
measure	{}	

pulse:

width	100
count	1

measure:

width	100
idle	101



**Формат WAVE7**

Принцип формата состоит в удалении одного бита из 8 битного семпла и переводе числа 0-127 в строковые данные.

Удаление бита:

$\text{Sample7} = \text{Sample8} \gg 1;$

Перевод в символы ANSI:

0 – 25	65 – 90	(A-Z)
26 – 51	97 – 122	(a-z)
52 – 61	48 – 57	(0-9)
62 – 125	192 – 255	(А-Яа-я)
126	43	(+)
127	47	(/)

### Принцип взаимодействия

Буфера приёма и отправки 2048 байт. Символ конца строки CRLF.  
Таймаут отправки 1000 мс. Таймаут ответа 3000 мс.

Если произошёл таймаут или другая фатальная ошибка при передаче, то следует переподключиться и произвести инициализацию по новой.

Ожидание приёма должно быть асинхронным, а не синхронным ожиданием ответа после отправки.

Контроль таймаута ответа производится следующим способом: есть счетчик выполняемых команд, при каждом запросе команды счетчик инкрементируется и запускается счетчик таймаута. В процедуре приёма производится разбор входящих данных на ответы, каждый ответ декрементируется счетчик команд и перезагружает счетчик таймаута. Как только счетчик команд равен 0 отключает счетчик таймаута. Необходимо учесть, что есть асинхронные ответы не требующие запроса, и в случае если счетчик равен 0, такие ответы не должны декрементировать счетчик команд и перезагружает счетчик таймаута.

### Подключение и инициализация

После подключения необходимо отправить команду INFO, если файловая система не готова к работе выдаём сообщение об ошибке и периодически переподключаемся.

Далее запрашиваем таблицы мелодий и событий командой LIST, после ответа запрашиваем информацию о мелодиях и событиях командами GETMELODY и GETEVENT.

Запрашиваем конфигурации командой GETCONFIG.

На этом инициализация закончена, остальные команды отправляются от взаимодействия пользователя с программой.

### **Запись мелодии**

Запрашиваем поток записи командой SETMELODY.

Блокируем приложение и показываем прогресс.

Далее синхронно отправляем данные командой DATAMELODY, до момента получения ответа END.

Закрываем поток ENDMELODY.

Разблокируем приложение.

### **Потоковое аудио**

Отправляются синхронно данные командой SPEAK, в ответе приходит количество записанных данных, не записанные данные нужно отправить повторно. Процедура повторяется, пока данные не будут закончены.

Запись данных с микрофона нужно ограничить 10 секундами.

Нажимается кнопка, производится запись с микрофона до отпускания кнопки или по истечению времени ограничения.

Блокируется кнопки и начинается передача в сирену, по завершению передачи кнопка разблокируется.