

---

**OpenDev**  
**ODRFID(-M,-N,-E,RS485,MODBUS)**

Open Development LLC

2022-01-13

## Contents

<b>1</b>	<b>Обзор</b>	<b>4</b>
<b>2</b>	<b>Возможности</b>	<b>4</b>
2.1	ODRFID	4
2.2	ODRFID-485	5
<b>3</b>	<b>ODRFID (прошивка CDC)</b>	<b>5</b>
3.1	Параметры порта	5
3.2	Формат пакета данных	5
3.3	Режимы устройства	6
3.4	Коды ошибок	7
3.5	Набор AT-команд.	8
3.5.1	Информация об устройстве.	8
3.5.2	Сброс буфера	8
3.5.3	Управление светодиодом	8
3.5.4	Управление зуммером	9
3.5.5	Управление режимом работы устройства	10
3.5.6	Управление полем	11
3.5.7	Сканирование всех меток	12
3.5.8	Сканирование первой метки	12
3.5.9	Сканирование следующей метки	13
3.5.10	Выбор метки по идентификатору	14
3.5.11	Проверка наличия метки	14
3.5.12	Информация о выбранной метке	15
3.5.13	Чтение блока данных	16
3.5.14	Запись блока данных	17
3.5.15	Блок значения	18
3.5.16	Смена идентификатора	18
3.5.17	Набор команд для Mifare Plus	19
3.5.18	Набор команд для меток 125kHz	21
3.5.19	Конфигурация устройства	23
3.5.20	Управление устройством	34
3.6	Errata	35
3.6.1	П.1 Информация об устройстве	35
3.6.2	П.2 Смена уровня безопасности	36
3.6.3	П.3 Управление трансивером 13.56MHz	36

3.6.4	П.4 Сброс аутентификации Mifare Plus . . . . .	37
<b>4</b>	<b>ODRFID (прошивка HID)</b>	<b>37</b>
<b>5</b>	<b>ODRFID-485 (протокол AT)</b>	<b>37</b>
<b>6</b>	<b>ODRFID-485 (прошивка Modbus)</b>	<b>37</b>
6.1	Соглашение о нумерации . . . . .	38
6.2	Регистры Modbus . . . . .	38
6.2.1	Регистры ввода . . . . .	38
6.2.2	Регистры хранения . . . . .	38
6.3	Пример ввода последовательности команд №1: . . . . .	40
6.4	Пример ввода последовательности команд №2: . . . . .	40
6.5	Настройки порта . . . . .	41
6.6	Формат пакета . . . . .	42
6.6.1	Сканирование всех меток. . . . .	42
6.6.2	Управление вторым светодиодом . . . . .	42
6.6.3	Управление выходом . . . . .	44
6.6.4	Чтение входного канала . . . . .	44
6.6.5	Управление UART/RS485 . . . . .	45
6.7	Errata . . . . .	46
6.7.1	П.1 Регистры хранения 131-141 . . . . .	46
6.7.2	П.2 Символ конца пакета CR . . . . .	46
<b>7</b>	<b>Синтаксис строки форматирования</b>	<b>46</b>
<b>8</b>	<b>Типы меток</b>	<b>47</b>
<b>9</b>	<b>Аппаратные ревизии</b>	<b>48</b>
<b>10</b>	<b>Ревизии документации</b>	<b>49</b>
<b>11</b>	<b>О сайте</b>	<b>50</b>

## 1 Обзор

В данном руководстве представлена подробная информация об устройствах, типах прошивок устройств, их возможностях, а также дается подробное описание протокола устройств для интеграции.

Если у вас остались вопросы, пожалуйста, посетите наш раздел с [документацией](#). Если вам нужна помощь или консультация по устройствам, пожалуйста, обратитесь к [технической поддержке](#).

ООО “Открытые Разработки” <https://open-dev.ru>

## 2 Возможности

### 2.1 ODRFID

- Считыватель является USB-устройством (USB 2.0, full speed) с разъемом micro-USB. Размеры устройства 10.3x6.8x1.1см.
- Индикация: зеленый светодиод (опционально) и зуммер.
- Типы встроенного ПО:
  - HID (эмуляция клавиатуры) - не требует установки дополнительных драйверов (\*в Microsoft Windows, Apple macOS и Linux);
  - CDC/ACM (виртуальный последовательный порт) - работает без дополнительных драйверов в Microsoft Windows 8 и выше, Apple macOS и Linux. Предоставляется драйвер для Microsoft Windows 7 (32/64 бит).
  - Совмещенная HID + CDC/ACM (определяется в системе как композитное устройство) - эмуляция клавиатуры (HID) и управление по CDC/ACM.
  - Последовательный порт (передача данных через RS485 линии). Поддерживает протокол, совместимый с версией ПО CDC/ACM, обернутый в бинарный протокол для передачи по линии RS485.
  - Протокол MODBUS (поддерживает протокол, совместимый с версией ПО CDC/ACM, обернутый в протокол MODBUS RTU. Также доступно ограниченное управление через регистры MODBUS).
- (опционально) Поддержка карт 125кГц стандарта EM41xx (и совместимых).
- (опционально) Поддержка карт 13.56 MHz стандарта MIFARE (Classic 1K/4K/Mini, Ultralight, NTAG, Plus).
- Поддерживается смена / обновление встроенного ПО посредством звгрузчика (HID USB или Mass-Storage).

- Аппаратные ревизии устройств и их различия описаны в разделе [Ревизии](#).

## 2.2 ODRFID-485

- Питание: 5-20В постоянного тока.
- Интерфейс данных: RS-485.
- Индикация: светодиоды (зеленый и красный) и зуммер.
- Типы встроенного ПО:
  - AT-команды поверх шины RS485.
  - протокол MODBUS-RTU.
- Поддержка карт: 13.56 MHz совместимые с MIFARE (Classic 1K/4K/Mini, Ultralight, NTAG, Plus)

## 3 ODRFID (прошивка CDC)

Устройство определяется в операционной системе как виртуальный последовательный порт (COM-порт) и следует спецификации USB Communications Device Class, Abstract Control Model (CDC/ACM). Большинство операционных систем (включая Microsoft Windows, Linux и Apple macOS) поддерживают работу с такими устройствами без необходимости установки дополнительных драйверов.

Различия ревизий устройств описаны в разделе [Аппаратные ревизии](#).

Описанный ниже набор AT-команд предназначен для опроса, чтения и записи бесконтактных карт, а также настройки устройства.

### 3.1 Параметры порта

Устройство является *виртуальным* последовательным портом. Соответственно, такие параметры как скорость передачи данных, четность, количество бит данных и.т.п. Таким образом, конфигурация параметров последовательного порта не требуется, устройство будет работать при любых настройках.

### 3.2 Формат пакета данных

- Пакет данных от ПК к устройству должен начинаться с ключевого слова [AT](#) и заканчиваться символом возврата каретки (десятичный ASCII код 13, шестнадцатеричный `0x0d`, строка

“\r” в большинстве языков программирования).

- Пакет данных от устройства на ПК начинается и заканчивается двумя символами - возврат каретки и перенос строки (десятичный ASCII код 13 и 10 соответственно, 0x0d и 0x0a в шестнадцатеричном представлении, строка “\r\n” в большинстве языков программирования).
- Ответ устройства на пакет от ПК состоит из одного или более пакетов, последний из которых \r\nOK\r\n (в случае успешного выполнения операции) или \r\nERROR\r\n (в случае возникновения ошибки при выполнении операции).
- Дополнительные символы в конце или начале пакета, такие как пробел, символ табуляции, перенос строки, возврат каретки и.т.п (помимо описанных выше) **не** допускаются. Они не будут отброшены анализатором команд со стороны устройства и, как следствие, приведут к ошибке обработки команды.

### 3.3 Режимы устройства

Устройство может находиться в одном из двух режимов: автоматическое сканирование (**SCAN1**) и ручное сканирование (посредством команд от ПК к устройству, **SCAN0**). Начальное состояние устройства (после подачи питания или перезагрузки) определяется соответствующей параметром в конфигурации устройства. Конфигурация по-умолчанию устанавливает автоматический режим **SCAN1**.

- **SCAN0** - устройство не осуществляет постоянное самостоятельное сканирование меток, находящихся в поле. Все действия выполняются только при поступлении соответствующей команды от ПК.
- **SCAN1** - устройство самостоятельно постоянно сканирует метки, находящиеся в поле. Интервал сканирования определяется соответствующим параметром в конфигурации устройства. При входе/выходе метки из поля на ПК отправляется пакет:
  - \r\nSCAN: +<HEX DATA>\r\n при обнаружении метки
  - \r\nSCAN: -<HEX DATA>\r\n при выходе метки из поля считывателя.
- **SCAN2** - (начиная с версии *1.3F Dec 17 2018*) - дополнительный режим автоматического сканирования, осуществляющий форматированный вывод. Устройство самостоятельно постоянно сканирует метки, находящиеся в поле. Интервал сканирования определяется соответствующим параметром в конфигурации устройства. При обнаружении новой метки в поле, на ПК будет отправлен пакет \r\n[formatted data]\r\n (содержимое пакета определяется данными на метке и строкой форматирования, заданной в конфигурации устройства). В отличие от режима **SCAN1**, при выходе метки из поля пакет не отправляется. Устройство не хранит историю меток - если метка ушла из поля, затем

была возвращена в поле, пакет будет отправлен заново.

Дополнение:

- (начиная с версии *1.0F Apr 3 2018*) HEX DATA в режиме SCAN1 определяется параметрами UID и UID в конфигурации устройства. Если передается UID, то к нему опционально (в зависимости от соответствующего значения в конфигурации устройства) добавляется SAK.
- (до версии *1.0F Apr 3 2018*) HEX DATA это UID + SAK.
- Начиная с версии 1.5 режим SCAN1 является устаревшим и фактически совпадает с режимом SCAN2 (форматированный вывод).

### 3.4 Коды ошибок

Если при операции с меткой возникла ошибка, передается дополнительный пакет с кодом ошибки следующего формата:

```
1 +CME ERROR: <code>
```

где code это десятичное число, значение которого это битовое поле (максимум 32 бита).

- 0x00000001 - Protocol error
- 0x00000002 - Parity error
- 0x00000004 - Checksum error
- 0x00000008 - Collision
- 0x00000010 - Buffer overflow
- 0x00000020 - Tear event
- 0x00000040 - IC overheated
- 0x00000080 - FIFO write error
- 0x00000100 - Operation timed out
- 0x00000200 - Mifare NAK
- 0x00000400 - Authentication failure
- 0x00000800 - Generic communication error
- 0x00001000 - The TAG returned more data than expected (v1.5+)
- 0x00002000 - The TAG reply integrity error (v1.5+)

Биты 16-31 зарезервированы для внутреннего использования и должны быть проигнорированы.

## 3.5 Набор AT-команд.

### 3.5.1 Информация об устройстве.

Осуществляет запрос информации об устройстве и версии встроенного ПО (прошивки).

пожалуйста, ознакомьтесь с п.1 в [Errata](#)

#### Синтаксис:

---

Запрос	<code>ATI\r</code>
Ответ	<code>&lt;product description&gt; &lt;firmware version/build date&gt;</code> <code>S/N &lt;serial number&gt;</code>

---

#### Пример:

---

Запрос	<code>ATI\r</code>
Ответ	<code>Open-Development RFID Reader (CDC-AT)1.0F Jan 17 2018</code> <code>S/N 220333635434B431500280010</code> <code>OK</code>

---

### 3.5.2 Сброс буфера

(с версии v1.6)

Команда сбрасывает (уничтожает) внутренний буфер устройства, ожидающий отправки на ПК.

#### Синтаксис:

---

Запрос	<code>ATH\r</code>
Ответ	<code>OK</code>

---

### 3.5.3 Управление светодиодом

Команда управляет состоянием диода (если имеется). Если светодиод отсутствует, устройство ответит пакетом “ошибка”.

Состояние светодиода:

- 0 = выключен,
- 1 = включен
- 2 = мигает

Если состояние не указано (т.е. *nil*<sup>1</sup>), управление светодиодом передается встроенному ПО (состояние по-умолчанию).

**Синтаксис:**

---

Запрос	<code>AT+D1=[0/1/2/nil]\r</code>
--------	----------------------------------

Ответ	<code>OK</code>
-------	-----------------

---

**Пример:**

---

Запрос	<code>AT+D1=1\r</code>	Включить светодиод
--------	------------------------	--------------------

Ответ	<code>OK</code>
-------	-----------------

Запрос	<code>AT+D1=\r</code>	Вернуть управление встроенному ПО
--------	-----------------------	-----------------------------------

Ответ	<code>OK</code>
-------	-----------------

Запрос	<code>AT+D1?\r</code>	Запрос состояния
--------	-----------------------	------------------

Ответ	<code>+D1=1</code>	Светодиод включен
-------	--------------------	-------------------

	<code>OK</code>	
--	-----------------	--

---

### 3.5.4 Управление зуммером

Команда управляет состоянием зуммера.

Состояние зуммера:

- 0 = выключен
- 1 = включен

---

<sup>1</sup>*nil* означает отсутствие поля данных (напр. `AT+D1=`).

Если состояние не указано (т.е. *nil*<sup>2</sup>), управление зуммером передается встроенному ПО (состояние по-умолчанию).

**Синтаксис:**

---

Запрос	<code>AT+B=[0/1/nil]\r</code>
Ответ	<code>OK</code>

---

**Пример:**

---

Запрос	<code>AT+B=1\r</code>	Включить зуммер
Ответ	<code>OK</code>	
Запрос	<code>AT+B=\r</code>	Вернуть управление встроенному ПО
Ответ	<code>OK</code>	
Запрос	<code>AT+B?\r</code>	Запрос состояния зуммера
Ответ	<code>+B=1</code>	Зуммер включен
	<code>OK</code>	

---

**3.5.5 Управление режимом работы устройства**

Данная команда переключает режим работы устройства (см. [Режимы устройства](#)).

**Синтаксис:**

---

Запрос	<code>AT+SCAN[0/1/2]\r</code>
Ответ	<code>OK</code>

---

**Пример:**

---

<sup>2</sup>*nil* означает отсутствие поля данных (напр. `AT+D1=`).

---

Запрос	<code>AT+SCAN0\r</code>	
Ответ	<code>OK</code>	Включен ручной режим

---



---

Запрос	<code>AT+SCAN?\r</code>	Запрос режима (поддерживается с версии 1.4F <sup>3</sup> )
Ответ	<code>+SCAN=0</code>	Текущий режим: ручной
	<code>OK</code>	

---

### 3.5.6 Управление полем

Данная команда включает / выключает радиочастотную антенну устройства. Устройство может сканировать метки только если антенна включена.

#### Заметка 1

- Для мульти-диапазонных считывателей (т.е. поддерживающих оба диапазона 125kHz и 13.56MHz) данная команда включает и выключает обе антенны, но только если соответствующий диапазон включен в настройках (см. команды `AT+h` и `AT+m`). Команда запроса состояния поля ненадежна и не должна использоваться.
- Для одно-диапазонных считывателей (только 125kHz или только 13.56MHz), команда включает и выключает антенну устройства, команды `AT+h` и `AT+m` недоступны.

#### Заметка 2

В режиме автоматического сканирования результат выполнения команды запроса состояния (`+RF?`) ненадежен. Данная команда выключена в считывателях, поддерживающих диапазон 13.56MHz в прошивках начиная с версии 2.2. Для мульти-диапазонных считывателей данная команда не имеет смысла и была выключена в прошивках начиная с версии 2.2.

#### Синтаксис:

---

Запрос	<code>AT+RF=[0/1]\r</code>	1 - включить 0 - выключить
Ответ	<code>OK</code>	

---

<sup>3</sup>режим 2 поддерживается с версии 1.3F, режим 1 устарел с версии 1.5F и объединен с 2.

**Пример:**


---

Запрос	<code>AT+RF=0\r</code>	
Ответ	<code>OK</code>	Антенна выключена

---

**3.5.7 Сканирование всех меток**

Запрос идентификаторов всех меток, находящихся в поле устройства. По завершении опроса, устройство возвращает ноль или более пакетов с идентификаторами обнаруженных меток. Последний пакет всегда “успех” (`OK`). Т.о. в ответ на команду придет один или более пакет. Все обнаруженные метки перезагружаются по завершении сканирования.

Устройству может понадобиться некоторое время для опроса всех меток. Следующая команда должна быть отправлена не ранее, чем получен ответ на предыдущую.

**Синтаксис:**


---

Запрос	<code>AT+I\r</code>	
Ответ	<code>+UID=&lt;HEX UID&gt;&lt;SAK</code>	Пакетов <code>+UID= . . .</code> может быть более одного или ни
	<code>&gt;</code>	одного.
	<code>OK</code>	

---

**Пример:**


---

Запрос	<code>AT+I\r</code>
Ответ	<code>+UID=EC6D140708</code>
	<code>+UID=343D7091725D8600</code>
	<code>OK</code>

---

**3.5.8 Сканирование первой метки**

Сканируется и активируется первая (согласно процедуре анти-коллизий) метка из находящихся в поле считывателя.

Если метка активирована, она помещается в спящее состояние (т.н. `halted state`).

Возвращается пакет с ее идентификатором и SAK и пакет “успех” (OK).

Если меток нет, команда считается выполненной и возвращается только пакет “успех”.

**Синтаксис:**

---

Запрос	<code>AT+i\r</code>	
Ответ	<code>+UID=&lt;HEX UID&gt;&lt;SAK&gt;</code>	Пакет <code>+UID</code> . . . может отсутствовать
	<code>OK</code>	

---

**Пример:**

---

Запрос	<code>AT+i\r</code>
Ответ	<code>+UID=EC6D140708</code>
	<code>OK</code>

---

### 3.5.9 Сканирование следующей метки

Деактивируется активная на данный момент метка и выбирается следующая (согласно процедуре анти-коллизий) 13.56MHz метка. Если метка активирована, она помещается в спящее состояние (т.н. *halted state*). Возвращается пакет с ее идентификатором и SAK и пакет “успех” (OK). Если меток нет, команда считается выполненной и возвращается только пакет “успех”.

‘-e’-ревизии устройств не поддерживают данную команду и возвращают пакет “ошибка”.

**Синтаксис:**

---

Запрос	<code>AT+n\r</code>	
Ответ	<code>+UID=&lt;HEX UID&gt;&lt;SAK&gt;</code>	Пакет <code>+UID</code> . . . может отсутствовать
	<code>OK</code>	

---

**Пример:**

---

Запрос	<code>AT+n\r</code>
Ответ	<code>+UID=343D7091725D8600</code>

---

---

OK

---

### 3.5.10 Выбор метки по идентификатору

Выбирается и активируется метка с заданным идентификатором. Команды работы с содержимым метки (чтение блока, запись блока и.т.д.) работают с “выбранной” меткой. В данном контексте “выбранная” означает “первая согласно процедуре анти-коллизии” либо заданная при помощи данной команды. Передача пустого UID (т.е. `nil`) деактивирует текущую метку (если есть).

Начиная с версии 1.6, под **выбранная** должно пониматься: ‘Метка, выбранная одной из команд `AT+i`, `AT+n` или `AT+SELECT=...`, которая была выполнена последней’ (**не** “первая согласно процедуре анти-коллизии”).

‘-e’-ревизии устройств не поддерживают данную команду и возвращают пакет “ошибка”.

#### Синтаксис:

---

Запрос	<code>AT+SELECT=&lt;HEX UID&gt;\r</code>	
Ответ	<code>OK</code>	Метка выбрана
	<code>ERROR</code>	Данная метка отсутствует

---

#### Пример:

---

Запрос	<code>AT+SELECT=343D7091725D86\r</code>	
Ответ	<code>OK</code>	Метка выбрана
Запрос	<code>AT+SELECT=343D7091725D87\r</code>	
Ответ	<code>ERROR</code>	Данная метка отсутствует

---

### 3.5.11 Проверка наличия метки

Данная команда проверяет, находится ли метка с указанным UID в поле устройства или нет. В отличие от команды `AT+SELECT=...`, проверяется только наличие. Если метка найдена, она **не** становится “выбранной”. После проверки, искомая метка (если найдена) помещается в

спящее состояние (т.н. `halted state`).

‘-e’-ревизии устройств не поддерживают данную команду и возвращают пакет “ошибка”.

**Синтаксис:**

---

Запрос	<code>AT+C=&lt;HEX UID&gt;\r</code>	
Ответ	<code>OK</code>	Метка присутствует
	<code>ERROR</code>	Метка отсутствует

---

**Пример:**

---

Запрос	<code>AT+C=343D7091725D86\r</code>	
Ответ	<code>OK</code>	Метка присутствует <i>Выбранная метка не изменена!</i>
Запрос	<code>AT+C=343D7091725D87\r</code>	
Ответ	<code>ERROR</code>	Метка отсутствует

---

### 3.5.12 Информация о выбранной метке

Запрос информации о выбранной метке: идентификатор UID, SAK, размер блока (BS), количество блоков (BC), тип метки (T). UID и SAK возвращаются в шестнадцатеричном формате, остальные поля - в десятичном.

Таблица поддерживаемых типов меток: см. [Типы меток](#)

**Синтаксис:**

---

Запрос	<code>AT+S\r</code>	
Ответ	<code>+UID=&lt;HEX UID&gt;&lt;SAK&gt;,BC=&lt;bc&gt;,BS=&lt;bs&gt;,T=&lt;t&gt;</code>	
	<code>OK</code>	
	<code>ERROR</code>	метка не ответила или

---

включен режим автоматического  
сканирования

---

**Пример:**


---

Запрос	AT+S\r
Ответ	+UID=EC6D140708,BC=64,BS=16,T=0
	OK

---

**3.5.13 Чтение блока данных**

*(только для Classic, Plus S/X (SL1), Ultralight и NTAG-ов)*

Команда запрашивает блок данных с метки. Номер блока должен быть в пределах, поддерживаемых данной меткой (см. поле *BC* в предыдущей команде). Блок передается в десятичном формате. В случае успешного чтения, возвращается номер блока и данные в шестнадцатеричном формате. Количество байт данных (каждый байт представлен двумя шестнадцатеричными цифрами) равно размеру блока (см. поле *BS* в предыдущей команде). Метка должна быть выбрана заранее одной из команд *AT+i*, *AT+n*, или *AT+SELECT*.

‘-e’-ревизии устройств не поддерживают данную команду и возвращают пакет “ошибка”.

Начиная с версии 3.2 блок может быть передан в шестнадцатеричном формате: *AT+R10* или *AT+R0xA*. Ответ считывателя по-прежнему всегда содержит блок в десятичном формате.

**Синтаксис:**


---

Запрос	AT+R<block number>\r	
Ответ	+DATA <block number>:<hex data>	
	OK	
	+CME ERROR: <code>	Метка не ответила, ошибка чтения/авторизации или
	ERROR	включен режим автоматического сканирования

---

**Пример:**


---

```
Запрос  AT+R0\r
Ответ   +DATA 0:EC6D1407920804009944314230353913
        ОК
```

---

**3.5.14 Запись блока данных**

(только для *Classic, Plus S/X (SL1), Ultralight* и *NTAG-ов*)

Команда пишет данные в указанный блок метки. Номер блока должен быть в пределах, поддерживаемых данной меткой (см. поле *BC* в команде *AT+S*). Блок передается в десятичном формате. Данные передаются после блока через двоеточие в шестнадцатеричном формате. Количество байт данных равно размеру блока (см. поле *BS* в команде *AT+S*). Т.е. должно быть передано  $2 \cdot BS$  шестнадцатеричных цифр.

‘-e’-ревизии устройств не поддерживают данную команду и возвращают пакет “ошибка”.

Начиная с версии 3.2 блок может быть передан в шестнадцатеричном формате: *AT+W10:...* или *AT+W0xA:....*

**Синтаксис:**


---

```
Запрос  AT+W<block number>:<
        HEX DATA>

Ответ   ОК

        +CME ERROR: <code>      Метка не ответила, ошибка чтения/авторизации,
        ERROR                    ошибка синтаксиса или включен режим
                                автоматического сканирования
```

---

**Пример:**


---

```
Запрос  AT+W1:000102030405060708090A0B0C0D0E0F\r
Ответ   ОК
```

---

### 3.5.15 Блок значения

(поддерживается с версии 1.6, только для Classic, Plus S/X (SL1))

Увеличить/уменьшить значение в блоке значение метки типа Mifare Classic или совместимой.

‘-e’-ревизии устройств не поддерживают данную команду и возвращают пакет “ошибка”.

Начиная с версии 3.2 блок может быть передан в шестнадцатеричном формате: AT+V<I/D>>10:... или AT+V<I/D>0x0a:....

#### Синтаксис:

---

Запрос AT+V<I/D><block number>  
>:<value>

Ответ OK

+CME ERROR: <code>            Ошибка чтения/записи/аутентификации (если присутствует +CME...)

ERROR                            ошибка синтаксиса или включен автоматический режим

---

Для увеличения значения необходимо указать I, для уменьшения - D. Отрицательные числа не поддерживаются. Номер блока и значение должны передаваться в десятичном виде.

#### Пример:

---

Запрос AT+VI1:42\r

Ответ OK

---

### 3.5.16 Смена идентификатора

Не все метки поддерживают изменение идентификатора (т.е. перезаписываемый блок 0). Это опасная операция. В лучшем случае ничего не изменится, в худшем - метка перестанет работать. Вы выполняете данную операцию на свой страх и риск. Данная команда поддерживается начиная с версии 1.5.

‘-e’-ревизии устройств не поддерживают данную команду и возвращают пакет “ошибка”.

#### Синтаксис:

Запрос	AT+~<HEX UID>	(*)
Ответ	OK	Успех

\* Длина нового идентификатора **должна** быть равна длине старого. Изменение длины идентификатора не поддерживается. Поддерживаются только метки типа Mifare Classic.

### 3.5.17 Набор команд для Mifare Plus

(поддерживается с версии v1.6)

Все '-e' ревизии и часть '-m' ревизий устройств не поддерживают данные команды и вернут пакет "ошибка".

**3.5.17.1 Аутентификация** Выполняет аутентификацию Mifare Plus для заданного блока.

пожалуйста, обратитесь к п.4 в [Errata](#)

**Синтаксис:**

Запрос	AT+N<b\lock>	
Ответ	OK	Успех
	+CME ERROR: <code>	Ошибка аутентификации или
	ERROR	ошибка синтаксиса

<b\lock> должен быть десятичным числом. Для сброса аутентификации необходимо передать ноль (0).

Начиная с версии 3.2 блок может быть передан в шестнадцатеричном формате: AT+N10 или AT+N0x0a.

**3.5.17.2 Чтение блока данных** Команда запрашивает блок данных с метки. Предполагается, что перед этой командой была успешно проведена авторизация для соответствующего блока.

**Синтаксис:**

---

Запрос	<code>AT+M&lt;block number&gt;\r</code>	
Ответ	<code>+DATA &lt;block number&gt;:&lt;hex data&gt;</code>	
	<code>&gt;</code>	
	<code>OK</code>	
	<code>+CME ERROR: &lt;code&gt;</code>	Метка не ответила, вернула ошибку или
	<code>ERROR</code>	включен автоматический режим

---

**Пример:**


---

Запрос	<code>AT+M1\r</code>
Ответ	<code>+DATA 1:EC6D1407920804009944314230353913</code>
	<code>OK</code>

---

Номер блока должен быть передан в десятичном формате.

Начиная с версии 3.2 блок может быть передан в шестнадцатеричном формате: `AT+M10` или `AT+M0xA`. Ответ считывателя по-прежнему всегда содержит блок в десятичном формате.

**3.5.17.3 Запись блока данных** Команда пишет данные в указанный блок метки. Предполагается, что перед этой командой была успешно проведена авторизация для соответствующего блока. Номер блока должен быть в пределах, поддерживаемых данной меткой (см. поле *BC* в команде `AT+S`). Блок передается в десятичном формате. Данные передаются после блока через двоеточие в шестнадцатеричном формате. Количество байт данных равно размеру блока (16 для меток Mifare Plus). Т.е. должно быть передано 32 шестнадцатеричных цифры.

Начиная с версии 3.2 блок может быть передан в шестнадцатеричном формате: `AT+E10:...` или `AT+E0xA:....`

**Синтаксис:**


---

Запрос	<code>AT+E&lt;block number&gt;:&lt;HEX DATA&gt;</code>	
Ответ	<code>OK</code>	Успех
	<code>+CME ERROR: &lt;code&gt;</code>	Метка не ответила, ошибка чтения/авторизации,

---

---

**ERROR**ошибка синтаксиса или включен режим  
автоматического сканирования

---

**Пример:**

---

Запрос	<code>AT+E1:000102030405060708090A0B0C0D0E0F\r</code>
Ответ	<code>OK</code>

---

**3.5.17.4 Смена уровня безопасности** Повышение уровня безопасности метки Mifare Plus. Новый уровень должен быть передан как десятичное число.

**Синтаксис:**

---

Запрос	<code>AT+J&lt;level&gt;</code>
--------	--------------------------------

Ответ	<code>OK</code>
-------	-----------------

Успех.

<code>+CME ERROR: &lt;code&gt;</code>
---------------------------------------

Метка не ответила, ошибка чтения/авторизации,

<code>ERROR</code>
--------------------

ошибка синтаксиса или включен режим автоматического  
сканирования

---

**3.5.18 Набор команд для меток 125kHz**

Поддерживается только ревизиями `-e/m`.

**3.5.18.1 Запись меток типа T55x7** Устройство позволяет программировать метки типа T55x7 (T5557/67/... и совместимые с ними на уровне протокола) в режиме эмуляции метки Em-Marine. При последующем считывании метка будет определяться как метка типа EM4100 с указанным идентификатором.

**Бифазное кодирование**

В режиме совместимости по-умолчанию для чипов T5577 (т.н. *e5555-compatibility mode*) бифазное кодирование **инвертировано** по сравнению со стандартом EM4100. Поэтому при использовании бифазной модуляции, устройство попытается запрограммировать чип T5577 в расширенном режиме, в котором доступно совместимое бифазное кодирование.

Данная операция может завершиться с ошибкой, поэтому рекомендуется использовать манчестерское кодирование.

#### Синтаксис:

---

Запрос	<code>AT+@НННННННННН\</code>	<code>НН. .НН</code> - 5 байт (10 шестнадцатеричных символов): Customer ID (1 байт) + UID (4 байта)
Ответ	<code>OK</code>	

---

Если перед этим было выполнено сканирование метки и была выбрана EM-Marine метка, 10 символов идентификатора могут быть опущены - тогда будет произведено клонирование метки.

#### Пример:

---

Запрос	<code>AT+@4201020304\</code>	
Ответ	<code>OK</code>	Метка запрограммирована

---

#### Пример 2:

---

Запрос	<code>AT+i\</code>	
Ответ	<code>+UID=1011121314FFOK</code>	Обнаружена метка EM-Marine Уберите старую метку и поднесите новую
Запрос	<code>AT+@\</code>	
Ответ	<code>OK</code>	Новая метка запрограммирована и имеет идентификатор старой

---

Устройство не имеет возможности проверить, что программирование завершено успешно (возможности проверить наличие совместимой метки в поле считывателя также нет). В связи с этим пакет “успех” будет отправлен устройством вне зависимости от того, была ли метка запрограммирована на самом деле или нет. Проверить успех выполнения операции можно, например, последующей командой [сканирование метки](#).

**3.5.18.2 Запись меток типа T55x7 с защитой паролем.** Команда аналогична предыдущей, но в дополнение к записи идентификатора, T55x7-совместимая метка также защищается паролем от несанкционированного доступа

**Синтаксис:**

---

Запрос	AT+xPPPPPPP,nnnnnnnnn\r	PP..PP - 32х битное беззнаковое целое.
Ответ	OK	

---

Запятая и последующие 10 символов данных могут быть опущены для проведения операции клонирования (см. комментарий к предыдущей команде). Пароль и данные должны быть переданы а шестнадцатеричном формате.

**3.5.18.3 Сброс пароля метки T55x7  
Угроза безопасности**

В результате данной операции будет снята только *защита* метки паролем. Сам пароль и данные останутся в памяти метки - и будут свободно **доступны** для чтения.

Пожалуйста, имейте в виду, что устройство не имеет возможности проверить что: \* совместимая метка находится в поле \* указан корректный пароль \* защита была на самом деле снята.

**Синтаксис:**

---

Запрос	AT+uPPPPPPP\r	PP..PP - 32х битное беззнаковое целое.
Ответ	OK	

---

Пароль должен быть передан а шестнадцатеричном формате, так же, как и при записи.

**3.5.19 Конфигурация устройства**

Последующие команды предназначены для изменения параметров устройства. Они выполняются вне зависимости от **режима работы** устройства. Отдельные команды изменяют только временные настройки. Для сохранения временных настроек в энергонезависимую память необходимо выполнить команду AT+P.

**3.5.19.1 Светодиод** Данная команда включает/выключает управление светодиодом. Данный параметр влияет только на то, будет ли встроенное ПО включать светодиод при поднесении метки или нет. Прямое управление светодиодом (команда AT+D1...) доступно вне зависимости от данного параметра.

**Синтаксис:**

---

Запрос	<code>AT+L[0/1/?]\r</code>	? - запрос текущего значения 0 - не включать светодиод 1 - включать светодиод
Ответ	<code>OK</code>	

---

**Пример:**

---

Запрос	<code>AT+L?\r</code>	
Ответ	<code>+L1</code>	Встроенное ПО будет включать/выключать светодиод
	<code>OK</code>	

---

**3.5.19.2 Зуммер** Данная команда включает/выключает управление зуммером. Данный параметр влияет только на то, будет ли встроенное ПО включать зуммер при поднесении метки или нет. Прямое управление зуммером (команда `AT+B...`) доступно вне зависимости от данного параметра.

**Синтаксис:**

---

Запрос	<code>AT+Z[0/1/?]\r</code>	? - запрос текущего значения 0 - не включать зуммер 1 - включать зуммер
Ответ	<code>OK</code>	

---

**Пример:**

---

Запрос	<code>AT+Z?\r</code>	
Ответ	<code>+Z1</code>	Встроенное ПО будет включать зуммер
	<code>OK</code>	

---

**3.5.19.3 Чувствительность приемника** Чувствительность измеряется в *dBm* и может принимать значения в диапазоне [18;48] *dBm*. Значение по-умолчанию - 33*dBm*. Данный

параметр влияет только на чувствительность 13.56МГц приемника.

Поддерживаемые значения:

- 18dBm
- 23dBm
- 33dBm
- 38dBm
- 43dBm
- 48dBm

Если передано значение, отличное от перечисленных выше, будет использовано ближайшее значение с округлением в бо́льшую сторону.

#### Синтаксис:

---

Запрос	<code>AT+G=[18-48]\r</code>	? - запрос текущего значения
		+G= - сброс к значению по-умолчанию (33)
Ответ	<code>OK</code>	

---

#### Пример:

---

Запрос	<code>AT+G=24\r</code>	
Ответ	<code>OK</code>	Значение будет округлено
		Значение будет применено позже.

---

Запрос	<code>AT+G?\r</code>	
Ответ	<code>+G=33</code>	Значение 24 было округлено.
	<code>OK</code>	

---

**3.5.19.4 Частота опроса меток** Если устройство находится в режиме автоматического сканирования, оно опрашивает метку раз в заданное количество миллисекунд. Данная команда задает этот интервал. Допустимый диапазон значений - [100 - 65535] мс. Значения меньше минимального будут приняты и минимально допустимое значение (100мс) будет установлено. Значения больше максимального не будут приняты и вызовут ошибку.

Значение по-умолчанию: \* до версии 1.4 - 1000 мс. \* начиная с версии 1.4: 200 мс.

**Синтаксис:**

---

Запрос	<code>AT+T[?/&lt;value&gt;]\r</code>	? - запрос текущего значения
Ответ	<code>OK</code>	

---

**Пример:**

---

Запрос	<code>AT+T?\r</code>	
Ответ	<code>+T1000</code>	Текущий интервал 1000ms
	<code>OK</code>	

---

Запрос	<code>AT+T1500\r</code>	
Ответ	<code>OK</code>	Интервал установлен в 1500ms

---

**3.5.19.5 Ключи авторизации** Метки типа Mifare Classic/Plus требуют ключ авторизации для доступа к содержимому метки. Данная команда устанавливает ключ авторизации и тип ключа (для Mifare Classic). Значение по-умолчанию: все байты равны 0xFF.

Ключи записываются в устройство и считываются с устройства согласно настоящему протоколу в текстовом виде (без кодирования, защиты и.т.п.). При сохранении настроек в энергонезависимую память, ключи также будут сохранены в текстовом виде и доступны для последующего считывания согласно настоящему протоколу. Если сохранение конкретного ключа нежелательно, установите его в значение по-умолчанию (FF . . FF) перед выполнением команды сохранения настроек AT+P.

**Синтаксис:**

---

Запрос	<code>AT+K&lt;type&gt;&lt;hex data&gt;\r</code>	тип может быть A или B
Ответ	<code>OK</code>	

---

**3.5.19.5.1 Ultralight** Начиная с версии 1.5:

Тип **U** задает пароль авторизации для меток типа Ultralight EV1 и NTAG's:

---

Запрос	<code>AT+KU&lt;hex data&gt;\r</code>	4 байта (8 шестнадцатеричных символов)
Ответ	<code>OK</code>	

---

Тип **P** производит авторизацию и запрашивает *PACK* текущей метки:

---

Запрос	<code>AT+KP\r</code>	
Ответ	<code>+P&lt;hex data&gt;</code>	2 байта (4 шестнадцатеричных символа)
	<code>OK</code>	

---

### 3.5.19.5.2 Mifare Plus

Начиная с версии 1.6:

Тип **X** задает ключ авторизации (16 байт - 32 шестнадцатеричных символа) для меток типа Mifare Plus.

---

Запрос	<code>AT+KX&lt;hex data&gt;\r</code>
Ответ	<code>OK</code>

---

#### Пример:

---

Запрос	<code>AT+KA000102030405\r</code>	
Ответ	<code>OK</code>	Задан ключ A: 00 . . 05

---

Запрос	<code>AT+K?</code>	
Ответ	<code>+A000102030405\r</code>	Возвращает текущие тип и ключ Mifare Classic.
	<code>+UFFFFFFFF</code>	(с 1.5) возвращает пароль Ultralight
	<code>+XFF . . FF</code>	(с 1.6) возвращает ключ Mifare Plus
	<code>OK</code>	

---

### 3.5.19.6 Управление виртуальной клавиатурой

Данный раздел применим только к третьему поколению устройств (ODRFID-N/M/E с прошивками версий 3.0 и выше).

Устройства третьего поколения имеют комбинированный USB интерфейс - виртуальный последовательный порт CDC (протокол описан настоящим документом) и HID (определяется на ПК как клавиатура). В отличие от предыдущих поколений, интерфейс HID работает только как клавиатура и не предоставляет собственного протокола для команд.

Клавиатура может быть включена (состояние по-умолчанию) или выключена. Обратите внимание, что даже при выключенной клавиатуре, со стороны ПК устройство все равно будет определяться как клавиатура, но она ничего не будет “печатать”.

Если клавиатура включена, в режиме автоматического сканирования устройство отправляет форматированный вывод на ПК как будто данные “печатая” на клавиатуре и **не** дублируется в последовательный порт. Если клавиатура выключена, форматированный вывод в режиме автоматического сканирования отправляется через последовательный порт согласно настоящему документу.

При “печати” на клавиатуре, например, если необходимо отправить символ “d”, будет нажата клавиша “d” (она же “в” в русской раскладке). Какой символ будет напечатан (d, в или другой соответствующий) и куда он будет отправлен целиком зависит от текущего состояния операционной системы.

#### Синтаксис:

Запрос	<code>AT+H[0/1/?]\r</code>	? - запрос текущего значения 0 - выключить клавиатуру 1 - включить клавиатуру
Ответ	<code>OK</code>	

#### Пример:

Запрос	<code>AT+H?\r</code>	
Ответ	<code>+H1</code>	Клавиатура включена
	<code>OK</code>	
Запрос	<code>AT+H0\r</code>	

---

Ответ	OK	The keyboard is now disabled
-------	----	------------------------------

---

### 3.5.19.7 Управление трансивером 13.56MHz (только для -m-ревизий устройств)

Включение/выключение трансивера 13.56MHz. По-умолчанию: включено.

Если трансивер выключен, соответствующие метки не будут сканироваться.

метки 125kHz всегда сканируются **первыми** если соответствующий трансивер включен.

обратитесь к п.3 в [Errata](#)

#### Синтаксис:

---

Запрос	AT+h[0/1/?]\r	? - запрос текущего значения
		0 - выключить
		1 - включить
Ответ	OK	

---

#### Пример:

---

Запрос	AT+h?\r	
Ответ	+h1	трансивер 13.56МГц включен
	OK	

---

### 3.5.19.8 Управление трансивером 125kHz (только для -m-ревизий устройств)

Включение/выключение трансивера 125kHz. По-умолчанию: включено.

Если трансивер выключен, соответствующие метки не будут сканироваться.

метки 125kHz всегда сканируются **первыми** если соответствующий трансивер включен.

#### Синтаксис:

---

Запрос	AT+m[0/1/?]\r	? - запрос текущего значения
		0 - выключить
		1 - включить

---



---

Запрос	<code>AT+A[0/1/?]\r</code>	? - запрос текущего значения 0 - не добавлять SAK 1 - добавлять SAK
Ответ	<code>OK</code>	

---

**Пример:**


---

Запрос	<code>AT+A?\r</code>	
Ответ	<code>+A=1</code>	SAK будет добавлен
	<code>OK</code>	

---

**3.5.19.11 Выбор данных в режиме сканирования** (для версий *1.0F Apr 3 2018* и старше, не поддерживается начиная с версий *1.5F*)

Данный параметр определяет, какие данные будут выданы в режиме автоматического сканирования: \* 1 - идентификатор (UID), \* 0 - блок (номер блока определяется параметром [Номер блока](#))

**Синтаксис:**


---

Запрос	<code>AT+U[0/1/?]\r</code>	? - запрос текущего значения 0 - печатать блок 1 - печатать UID (* по-умолчанию)
Ответ	<code>OK</code>	

---

**Пример:**


---

Запрос	<code>AT+U?\r</code>	
Ответ	<code>+U=1</code>	Будет напечатан UID
	<code>OK</code>	

---

### 3.5.19.12 Номер блока (в режиме сканирования) (для версий 1.0F Apr 3 2018 и старше, не поддерживается начиная с версий 1.5F)

Данный параметр определяет, какой блок будет прочитан и выдан в режиме автоматического сканирования, если выбрана опция “выдать блок” в параметре [Выбор данных](#)

Если блок не существует или не может быть прочитан (например, вследствие ошибки авторизации), ничего не будет выдано.

#### Синтаксис:

Запрос	<code>AT+b[?/&lt;value&gt;]\r</code>	? - запрос текущего значения
Ответ	<code>OK</code>	

#### Пример:

Запрос	<code>AT+b?\r</code>	
Ответ	<code>+b=0</code>	Будет напечатано содержимое блока 0
	<code>OK</code>	
Запрос	<code>AT+b1\r</code>	
Ответ	<code>OK</code>	Будет напечатано содержимое блока 1

### 3.5.19.13 Формат выдачи данных (режим SCAN2) (только для версий 1.3F Dec 17 2018 и выше)

Строка форматирования имеет одинаковый формат для версий прошивок CDC и HID. Подробнее см. в [описании строки форматирования](#) Формат по-умолчанию: `HU*\n`. Режим по-умолчанию (до того, как первый раз встретится один из ключей-символов `a`, `d`, `h`, или `H`): шестнадцатеричный вывод заглавными буквами.

все последовательности “\r” и “\n” в строке форматирования (т.е. два символа ‘\’ и последующий ‘r’ или ‘n’) будут приняты и выведены при использовании команд `AT+F...`, однако при печати (через последовательный порт) они будут проигнорированы, поскольку конфликтуют с разделителем пакетов. Символ ; зарезервирован для разделителя команд, его использование в строке форматирования приведет к ошибке или неопределенному поведению.

(начиная с версии 3.2) при задании строки форматирования последовательности вида `&#xHH` будут преобразованы в символ с ASCII кодом `0xHH` (в частности, для использования символа `;` ; необходимо передать последовательность `&#3B`).

**Синтаксис:**


---

Запрос	<code>AT+F[?/=&lt;format&gt;]\r</code>	? - запрос строки форматирования
Ответ	<code>OK</code>	

---

**Пример:**


---

Запрос	<code>AT+F?\r</code>	
Ответ	<code>+F=HU*\n</code>	Напечатать UID как заглавные шестнадцатеричные символы
	<code>OK</code>	

Запрос	<code>AT+F=hB1%*\r</code>	
Ответ	<code>OK</code>	Напечатать содержимое блока 1 как строчные шестнадцатеричные символы

---

**3.5.19.14 Сохранение настроек** Данная команда сохраняет текущую конфигурацию устройства в энергонезависимую память

**Синтаксис:**


---

Запрос	<code>AT+P\r</code>
Ответ	<code>OK</code>

---

**3.5.20 Управление устройством**

Следующие команды управляют самим устройством и доступны в режиме автоматического сканирования.

**3.5.20.1 Перезагрузка** Перезагрузить устройство

**Синтаксис:**

---

Запрос	AT+Q\r
Ответ	OK

---

**3.5.20.2 Обновление встроенного ПО** Перезагрузить устройство и перейти во встроенный загрузчик. Встроенный загрузчик доступен не на всех устройствах. Если загрузчик недоступен, команда аналогична AT+Q.

**Синтаксис:**

---

Запрос	AT+X\r
Ответ	OK

---

## 3.6 Errata

### 3.6.1 П.1 Информация об устройстве

---

---

Версии ПО	до v3.1 (включительно)
Действие	Обработка ответа на команду 'AT1'
Ожидаемый ответ	3 пакета - описание продукта, серийный номер, OK
Полученный ответ	2 пакета - описание продукта + серийный номер, OK. Между описанием продукта и серийным номером соединится только один разделитель \r\n вместо двух

## Решение

1. Не использовать данную команду.; 2. Настроить парсер на разделитель `\r\n` (один) и пропускать пустые пакеты (между двумя последовательными разделителями). На данный момент все устройства имеют сторонние методы контроля целостности пакета, поэтому между двумя пакетами не может быть “лишних” символов; 3. Настроить парсер на распознавание последовательности символов `\r\nS/N` и исправлять ее на `\r\n\r\nS/N`

### 3.6.2 П.2 Смена уровня безопасности

Версии ПО	с v3.0 до v3.1
Действие	Переключение с уровня 1 на уровень 3 командой 'AT+J3'
Ожидаемый результат	Метка переключается с уровня 1 на уровень 3
Полученный результат	Метка остается на уровне 1, устройство перестает сканировать метки до перезагрузки
Решение	Нет. Пожалуйста, обновитесь до последней версии ПО

Переключение с уровня 0 на уровень 1 работает как и должно.

### 3.6.3 П.3 Управление трансивером 13.56MHz

Версии ПО	с 2.0 до 2.1
Действие	установка параметра “включение трансивера”
Ожидаемый результат	действие выполнено

---

Полученный результат возможно, действие не выполнено

Решение После отправки команды `AT+h...` необходимо проверить результат выполнение командой `AT+h?`. Если первая команда не выполнена, необходимо отправить команду повторно.

---

### 3.6.4 П.4 Сброс аутентификации Mifare Plus

---

Версии ПО	все до 3.1 включительно
Действие	Сброс аутентификации командой <code>AT+N0</code>
Ожидаемый результат	Аутентификация сброшена, получен пакет “успех”
Полученный результат	Аутентификация не сброшена, получен пакет “ошибка”
Решение	Нет. Пожалуйста, обновитесь до последней версии ПО

---

## 4 ODRFID (прошивка HID)

Русская версия документа недоступна. Пожалуйста, обратитесь к английской версии.

## 5 ODRFID-485 (протокол AT)

Русская версия документа недоступна. Пожалуйста, обратитесь к английской версии.

## 6 ODRFID-485 (прошивка Modbus)

Устройство принимает команды и отправляет ответ через интерфейс RS485 по протоколу Modbus RTU.

Устройство использует набор AT команд, используемых в протоколе CDC, с минимальными отличиями.

Устройство использует/принимает только часть функций протокола Modbus RTU.

- Чтение регистров ввода (код функции 4)

- Чтение регистров хранения (код функции 3)
- Установка одного регистра хранения (код функции 6)
- Установка нескольких регистров хранения (код функции 16)
- Все остальные коды функций не поддерживаются и вызовут исключение

## 6.1 Соглашение о нумерации

Согласно спецификации протокола MODBUS, регистры хранения занимают адресное пространство 40001-49999. Однако передаваемый адрес является относительным (0 для 40001, 1 для 40002 и т. д.). Здесь и далее в этом документе используются относительные адреса (начиная с нуля).

## 6.2 Регистры Modbus

### 6.2.1 Регистры ввода

- **0** - возвращает размер (в байтах) внутренних данных, ожидающих чтения
- **1** - индикатор наличия метки:
  - 0 - метки нет.
  - любое другое значение - метка есть.
  - поддерживается с версии 3.1. Предыдущие версии вернут исключение.
  - в ручном режиме - регистр хранит результат последней операции сканирования ( $AT+i$ ,  $AT+n$ , и т.д.). В действительности, метки уже может не быть - поскольку устройство не осуществляет самостоятельное сканирование, оно об этом не может узнать.
  - в автоматическом режиме - операция сканирования осуществляется раз в  $T$  миллисекунд (см. [Конфигурация устройства](#)). Регистр хранит результат последнего сканирования и обновляется, соответственно, раз в  $T$  миллисекунд. После перехода в режим автоматического сканирования необходимо подождать не менее  $T$  миллисекунд (рекомендуется  $2 * T$ ) прежде чем опрашивать данный регистр.
- Все остальные регистры не поддерживаются, попытка прочитать их вызовет исключение

### 6.2.2 Регистры хранения

- 0-125 ( $0x00$  –  $0x7D$ ) - внутренний буфер данных

– *запись*: отправить AT-команду на устройство. Начальный адрес не важен (рекомендуется использовать 0), до тех пор пока данные укладываются в указанный диапазон.

– *чтение*: возвращает запрашиваемую часть внутреннего буфера данных.

Чтобы узнать количество доступных данных, ожидающих чтения, запросите первый входной регистр (адрес 0). Доступные данные начинаются с адреса 0 и занимают N байт ( $N = [\text{значение 1-го регистра ввода}]$ ) - т.е. K регистров, где  $K = \text{ceil}([\text{значение 1-го регистра ввода}] / 2)$ .

Устройство не будет проверять, являются ли запрошенные внутренние данные действительными или нет, и всегда будет возвращать запрошенное количество байт. Операция чтения регистров не обнуляет количество доступных байт, то есть значение 1-го регистра ввода останется неизменным после операции чтения.

- 126 (0x7E) - записать любое значение в этот регистр для обнуления числа байт, ожидающих чтения (1-й регистр ввода будет установлен в 0). Команда **игнорируется** если в данный момент осуществляется сканирование всех меток. В этом случае буфер не обнуляется.
- 127 (0x7F) - записать любое значение в этот регистр, чтобы перезагрузить устройство.
- 128 (0x80) - записать любое значение в этот регистр, чтобы перезапустить устройство и перевести его в режим DFU.

протокол DFU несовместим с протоколом Modbus RTU. Вы не сможете обновить прошивку устройства, не отключив его от линии Modbus.

- 130 (0x82) - только начиная с версии 2.1. (только чтение в версии 2.1, чтение-запись начиная с 2.2)
  - *чтение*: размер UID последней отсканированной карты
  - *запись*: сбросить значение регистров 130-141 в ноль.
- **131-141 (0x83 - 0x8D)** - только начиная с версии 2.1 (*только чтение*)
  - 131 (0x83) - тип последней отсканированной карты (см. [типы меток](#))
  - 132 - первый байт последней отсканированной карты
  - 133 - второй байт и.т.д.
  - 132-141 (0x84-0x8D) - первый, второй .и.т.д. байты UID последней отсканированной карты (количество регистров с данными определяется значением регистра 130)

пожалуйста, обратитесь к п.1 в [Errata](#)

- **143 (0x8F)** - **только с версии 3.1 (только запись)** - запись значения 0x4242 (16962) приведет к сбросу устройства к заводским настройкам (в.т.ч. адрес, скорость передачи данных, конфигурация устройства и.т.д.) и последующей перезагрузке.
- Все остальные регистры не поддерживаются, попытка чтения/записи в них вызовет исключение.

**Важно:** регистры 130-141 активны только в режиме автоматического сканирования. Они **не будут** обновляться в ручном режиме.

### 6.3 Пример ввода последовательности команд №1:

Адрес устройства по умолчанию (идентификатор ведомого устройства Modbus) - 95 (0x5F)

1. Удалите все ожидающие чтения данные: запишите 1 в регистр хранения 126

```
>> 5F06007E0000E4AC
```

```
<< 5F06007E0000E4AC
```

2. Отправьте команду AT (например, AT+SCAN0): запишите AT+SCAN0 в регистры хранения 0, 1, 2, 3 (т. к.  $K = \text{sizeof(AT+SCAN0)} / 2 = 8 / 2 = 4$ . Если команда занимает 7 байтов, она должна быть дополнена одним нулём - 0x00)

```
>> 5F10000000040841542B5343414E304A48
```

```
<< 5F1000000004CCB4
```

Устройство вернуло код успеха (0x10 вместо 0x90 для исключения), поэтому нет необходимости проверять ответ `\r\nOK\r\n` (или `\r\nERROR\r\n`), и он не передается устройством.

3. Убедитесь, что нет данных, ожидающих чтения (их не должно быть, это действие показано здесь только в демонстрационных целях)

```
>> 5F04000000013CB4
```

```
<< 5F0402000010FD
```

### 6.4 Пример ввода последовательности команд №2:

1. Удалите все ожидающие чтения данные: запишите 1 в регистр хранения 126

```
>> 5F06007E0000E4AC
```

```
<< 5F06007E0000E4AC
```

- Отправьте команду AT (например, AT+G?): Запишите AT+G?\0 в регистры хранения 0, 1, 2 (т. к.  $K = \text{sizeof(AT+G?)} / 2 = 5 / 2 = 3$  - с округлением в большую сторону).

```
>> 5F10000000030641542B473F00A682
```

```
<< 5F10000000038D76
```

Устройство вернуло код успеха, теперь мы можем запросить результат команды

- Проверьте количество байт, ожидающих чтения: прочитайте значение первого регистра ввода

```
>> 5F04000000013CB4
```

```
<< 5F04020009D0FB
```

Устройство сообщает о 9 байтах, ожидающих чтения. Нам нужно прочитать 5 регистров хранения ( $\text{ceil}(9/2) = 5$ ).

- Прочитать внутренние данные в ожидании: прочитайте соответствующее количество регистров хранения.

```
>> 5F030000000588B7
```

```
<< 5F030A0D0A2B473D33330D0A30F147
```

Обратите внимание на мусор (0x30) в конце данных - мы прочитали один дополнительный байт, который должен быть сброшен обрабатывающим программным обеспечением. Значимые данные составляют 9 байтов (0A0D0A2B473D33330D0A, т. е.  $\backslash r \backslash n + G = 33 \backslash r \backslash n$ ).

- Важно:** нужно сбросить ожидающие чтения данные (это то же самое, что и первый шаг. Программное обеспечение мастера может очищать данные как до, так и после команды (до тех пор, пока внутренний буфер не переполняется), но хотя бы один из шагов (1 либо 4) должен быть выполнен.

```
>> 5F06007E0000E4AC
```

```
<< 5F06007E0000E4AC
```

## 6.5 Настройки порта

Настройки по умолчанию: 115200 бит/с, 8N1 (8 бит, 1 стоповый бит, без контроля четности), адрес устройства 0x5F (95). Эти настройки можно изменить с помощью команды AT+[, описанной ниже. Новые настройки сохраняются в энергонезависимой памяти и применяются после сброса устройства.

## 6.6 Формат пакета

(Это относится к данным, отправляемым и получаемым с помощью команд Чтение/Установка регистров хранения)

- Данные мастер-устройство должны начинаться с ключевого слова AT и могут заканчиваться символом ноль (0x00), чтобы данные помещались в целое количество 16-битных регистров.

в отличие от протокола CDC, символ возврата каретки (`\r`) опционален.

пожалуйста, обратитесь у п.2 в [Errata](#).

- Пакеты между устройствами и хостом начинаются и заканчиваются символом возврата каретки, за которым следует символ перевода строки (ASCII-коды 13,10 десятичные, 0x0d, 0x0a шестнадцатеричные, `\r\n` как строковый литерал C).
- Ответ от устройства мастеру на запрос хост-устройства состоит из одного или нескольких пакетов.

*Примечание:* в отличие от протокола CDC, пакеты `\r\nOK\r\n` и `\r\nERROR\r\n` не передаются. Код успеха или ошибки передаются в ответном пакете Modbus согласно протоколу Modbus RTU: MSB функционального кода сигнализирует об ошибке.

- Не добавляйте ненужные символы пробела (пробел, табуляция, перевод строки и т. д.) к данным запроса - они не будут удалены анализатором командной строки, что приведет к ответу **ОШИБКА**.

Полное описание команд верхнего уровня см. в справочнике [ODRFID \(CDC\)](#). Обратите внимание, что пакеты `\r\nOK\r\n` и `\r\nERROR\r\n` отсутствуют. Различия перечислены ниже.

### 6.6.1 Сканирование всех меток.

Команда является асинхронной - UID отсканированного тега будет помещен в буфер (регистры хранения 0-125) после завершения процесса сканирования. Устройство не будет отвечать на запросы до тех пор, пока идет сканирование.

### 6.6.2 Управление вторым светодиодом

В дополнение к первому (зеленому) светодиоду, управляемому командой `AT+D1`, данное устройство имеет второй (красный) светодиод и поддерживает команду `AT+D2`.

**Синтаксис:**

---

Запрос	<code>AT+D2=[0 1]</code>	<i>0</i> - выключить второй светодиод, <i>1</i> - включить второй светодиод
Ответ	Код успеха согласно протоколу Modbus RTU	

---

### 6.6.3 Управление выходом

Данная команда (`AT+Y=[0|1]`) позволяет управлять состоянием выходного сигнала OUT (открытый коллектор, без подтяжки)

#### Синтаксис:

---

Запрос	<code>AT+Y=[0 1]</code>	Задать состояние OUT <i>0</i> - отключено (не подтянуто), <i>1</i> - включено (подключено к земле)
Ответ	Код успеха согласно протоколу Modbus RTU	

---

#### Пример

---

Запрос	<code>AT+Y=1\r</code>	Вывод OUT подключен к земле
Ответ	Код успеха согласно протоколу Modbus RTU	
Запрос	<code>AT+Y=0\r</code>	Вывод OUT не подтянут
Ответ	Код успеха согласно протоколу Modbus RTU	
Запрос	<code>AT+Y?\r</code>	Запрос состояния вывода OUT
Ответ	<code>+Y=0</code>	Вывод OUT не подтянут

---

### 6.6.4 Чтение входного канала

Данная команда (`AT+y?`) позволяет запрашивать состояние входа (если присутствует).

#### Синтаксис:

---

Запрос	AT+y?	Запрос состояния входа
Ответ	+y=0 или +y=1 OK	0 - подтянут к земле, 1 - подтянут к питанию

---

### 6.6.5 Управление UART/RS485

Эта команда запрашивает и устанавливает параметры, связанные с UART (скорость передачи, биты, четность и т. Д.). Возвращаемые значения - данные, которые сохранены в энергонезависимой памяти. Они могут отличаться от реальных в том случае, если они были изменены без последующей перезагрузки. Новые настройки сохраняются в энергонезависимой памяти немедленно и применяются при следующей перезагрузке.

Формат команды: AT+ [=<a>, <s>, <p>, <b>, <r>, <t>

Значения, используемые в запросе и возвращаемые в ответе

- <a> - адрес (1..247) в десятичном формате.
- <s> - стоп биты
  - 10 - 1 стоп бит
  - 15 - 1.5 стоп бита
  - 20 - 2 стоп бита
- <p> - четность
  - 0 - none
  - 1 - odd (с версии v 2.0)
  - 2 - even (с версии v 2.0)
- <b> - количество бит данных (поддерживается только 8 бит, остальные значения принимаются, но игнорируются)
- <r> - бодрейт
- <t> - межсимвольный интервал

При изменении настроек, касающихся UART/RS485, все настройки (включая конфигурацию устройства) будут записаны (вызывается команда AT+P).

#### Пример:

---

Запрос	AT+[? \r	
Ответ	+ [=95, 10, 0, 8, 115200, 10	115200 8N1, адрес 0x5F, таймаут 10ms

---

## 6.7 Errata

### 6.7.1 П.1 Регистры хранения 131-141

---

Версии ПО	v2.1, v2.2
Действие	Чтение произвольного количества регистров в указанном диапазоне
Ожидаемый ответ	Тип и идентификатор метки
Полученный ответ	Неверные данные
Решение	При чтении регистром с адресом больше 131, необходимо всегда читать регистры начиная с адреса <b>130</b> используя функциональный код 3
Пример	Чтение длины и идентификатора из регистров 132, 133, 134, 135
<b>неверно</b>	чтение 4-х регистров - код 3, адрес 132, количество 4
<b>верно</b>	чтение 6-ти регистров - код 3, адрес 130, количество 6

---

### 6.7.2 П.2 Символ конца пакета CR

---

Версии ПО	все версии до v2.2 включительно
Действие	завершение пакета от мастера устройству символом \r
Ожидаемый результат	нормальная обработка пакета согласно протоколу
Полученный результат	сообщение об ошибке modbus устройства
Решение	Не добавлять символ \r в конец фрейма. Если необходимо дополнить фрейм символом для четности, необходимо использовать символ ноль \0

---

## 7 Синтаксис строки форматирования

Пожалуйста, обратитесь к онлайн-версии: [Строка форматирования](#)

## 8 Типы меток

- 0 - Classic 1K
- 1 - Classic 4K
- 2 - Classic Mini
- 3 - Ultralight<sup>4</sup>
- 4 - Ultralight C (только UID)
- 5 - Ultralight EV1 (640 бит)
- 6 - Ultralight EV1 (1312 бит)
- 7 - Plus S 2K (SL1)<sup>5</sup>
- 8 - Plus S 4K (SL1)<sup>6</sup>
- 9 - DesFire (только UID)
- 10 - DesFire<sup>7</sup>
- 11 - DesFire<sup>8</sup>
- 12 - Classic 2K
- 13 - Plus X 2K (SL1)<sup>9</sup>
- 14 - Plus X 4K (SL1)<sup>10</sup>
- 15 - Plus X (SL0)<sup>11</sup>
- 16 - Plus X 2K (SL2)<sup>12</sup>
- 17 - Plus X 4K (SL2)<sup>13</sup>
- 18 - Plus X (SL3)<sup>14</sup>
- 19 - Plus S (SL0)<sup>15</sup>
- 20 - RFU
- 21 - RFU
- 22 - Plus S (SL3)<sup>16</sup>

<sup>4</sup>все Ultralight метки (Ultralight, Ultralight EV1) и NTAG метки будут идентифицированы как *Ultralight* (код 3) в прошивках до версии 1.5.

<sup>5</sup>до версии 1.5 - любая метка Plus S 2K/4K или Plus X 2K/4K соответственно

<sup>6</sup>до версии 1.5 - любая метка Plus S 2K/4K или Plus X 2K/4K соответственно

<sup>7</sup>метки DESFire не поддерживаются (доступно только чтение UID) - все метки типа DESFire будут идентифицированы как тип 9 с версии 1.5, как 9, 10, или 11 до версии 1.5.

<sup>8</sup>метки DESFire не поддерживаются (доступно только чтение UID) - все метки типа DESFire будут идентифицированы как тип 9 с версии 1.5, как 9, 10, или 11 до версии 1.5.

<sup>9</sup>до версии 1.5 - любая метка Plus S 2K/4K или Plus X 2K/4K соответственно

<sup>10</sup>до версии 1.5 - любая метка Plus S 2K/4K или Plus X 2K/4K соответственно

<sup>11</sup>с версии 1.5

<sup>12</sup>с версии 1.5

<sup>13</sup>с версии 1.5

<sup>14</sup>с версии 1.5

<sup>15</sup>с версии 1.5

<sup>16</sup>все Ultralight метки (Ultralight, Ultralight EV1) и NTAG метки будут идентифицированы как *Ultralight* (код 3) в

- 23 - NTAG213<sup>17</sup>
- 24 - NTAG215<sup>18</sup>
- 25 - NTAG216<sup>19</sup>
- 26 - Ultralight Nano<sup>20</sup>
- 27 - EM41xx 125kHz и совместимые
- 28 - NTAG413
- 29 - NTAG424
- 30 - Plus X (SL3, 4K)<sup>21</sup>
- 31 - Plus S (SL3, 4K)<sup>22</sup>
- 32 - Совместимая с протоколом ISO 14443-4 / ISO 7816-4
- 33 - EMV<sup>23</sup>
- 34-254 Зарезервировано<sup>24</sup>
- 255 - Неизвестно

## 9 Аппаратные ревизии

Устройство	Ревизия	Код прошивки	Частоты	Примечания
ODRFID	nil	F	13.56MHz	Поддержка меток Mifare Classic/Ultralight, индикация - светодиод и зуммер
	-M	m	13.56MHz / 125kHz	Поддержка меток Mifare Classic/Ultralight/NTAG/Plus (Mifare Plus поддерживается только в режиме совместимости с Classic), Em-Marine (чтение), T5577 (запись), индикация - зуммер

прошивках до версии 1.5.

<sup>17</sup>с версии 1.5

<sup>18</sup>с версии 1.5

<sup>19</sup>с версии 1.5

<sup>20</sup>с версии 1.5

<sup>21</sup>все Plus X/S 2K/4K с уровнем защиты 3 будут идентифицированы как 18 и 22 соответственно для поддержания совместимости.

<sup>22</sup>все Plus X/S 2K/4K с уровнем защиты 3 будут идентифицированы как 18 и 22 соответственно для поддержания совместимости.

<sup>23</sup>EMV-совместимые карты (код 33) являются подмножеством ISO 14443-4 / ISO 7816-4 карт (тип 32) и будут идентифицированы как тип 32 до того, как будет прочитано их содержимое.

<sup>24</sup>Метки EM41xx не имеют SAK. Считыватель присваивает им SAK=0xFF для поддержания совместимости с метками 13.56МГц.

Устройство	Ревизия	Код прошивки	Частоты	Примечания
	-N	n	13.56MHz	Поддержка меток Mifare Classic/Ultralight/NTAG/Plus, индикация - зуммер
	-E	e	125kHz	Поддержка меток Em-Marine (чтение), T5577 (запись), индикация - зуммер
ODRFID (2021)	-M (r3)	m	13.56MHz / 125kHz	Поддержка меток Mifare Classic/Ultralight/NTAG/Plus, индикация - светодиод, зуммер
	-N	n	13.56MHz	Поддержка меток Mifare Classic/Ultralight/NTAG/Plus, индикация - светодиод, зуммер
	-E	e	125kHz	Поддержка меток Em-Marine (чтение), T5577 (запись), индикация - светодиод, зуммер

### Прошивки. Пожалуйста, прочтите данный раздел

- Все ревизии устройств ODRFID имеют два варианта прошивки: 'CDC' (последовательный порт) и 'HID'. Один вариант можно сменить на другой путем обновления прошивки (даже если версии совпадают). Однако ревизии -M, -E и -N **нельзя** превратить одну в другую обновлением прошивки. Пожалуйста, убедитесь, что вы выбрали соответствующую прошивку перед обновлением. Некорректный выбор приведет к **полной неработоспособности** устройства. В этом случае придется отправлять устройство на ремонт в авторизованный сервисный центр.
- Ревизии -M и -E могут читать метки EM41xx и совместимые с ними (частота 125kHz) с манчестерским и бифазным кодированием и с 32/64 переходами на бит. Согласно спецификации EM4100 при бифазном кодировании используется режим "middle-bit transition is zero". Устройство способно читать оба варианта кодирования ("middle-bit transition is zero" и "middle-bit transition is one").
- Третье поколение устройств (2021) имеет одну прошивку с двумя интерфейсами (CDC и HID). Перепрошивка для смены интерфейса не требуется.

## 10 Ревизии документации

---

Ревизия	Дата	Примечания
1.0	2018-01-22	Первая версия
1.1	2018-04-27	Обновление протокола: Поддержка выбора UID/блок
1.2	2019-04-07	Добавлен протокол RS485
1.3	2019-08-20	Новые возможности прошивки 1.6
1.4	2019-10-29	Новое устройство: RS485-IO
1.5	2020-01-27	Поддержка устройств с диапазоном 125kHz
1.6	2020-12-08	Прошивка MODBUS v2.1
1.7	2021-03-16	Исправлено описание команд HID 125kHz
1.8	2021-08-26	Обновление протокола для устройств третьего поколения
1.9	2022-01-11	Errata. Перевод документации на русский

---

## 11 О сайте

ООО “Открытые Разработки” [open-dev.ru](https://open-dev.ru)

© 2018-2022. Все права защищены.

Тема MkDocs: [mkdocs-material](#) (Martin Donath)

Тема Pandoc: [EisVogel](#) (Pascal Wagler и John MacFarlane)