OpenDev RFID USB Reader

OPEN DEVELOPMENT LLC

2018-06-25

1 Overview

This reference manual provides operating instructions, command references for different firmwares, and other detailed product information. If you have any questions, please visit our Knowledge Base, and if you need further assistance, send us a Technical assistance request.

Open Development LLC https://open-dev.ru

2 Features

- Reader is a bus-powered USB 2.0 full-speed device with micro-USB connector. The device dimensions are 10.3x6.8x1.1cm.
- Indication: a green LED and a buzzer.
- Available firmware types include: HID (emulates a keyboard) - out-of-the-box OS support. No driver installation on Microsoft Windows, Apple macOS and Linux; CDC/ACM (virtual serial port) - works out of the box in Microsoft Windows 8 and 10, Apple macOS and Linux, a driver for Microsoft Windows 7 (32 bit and 64 bit) is available.
- Supports 13.56 MHz RFID MIFARE tags (Classic 1K/4K/Mini and Ultralight)
- Extensible and future-proof design: the device firmware can be easily changed or updated using built-in HID USB Bootloader.

3 CDC Firmware

The device appears as a virtual serial port (COM port) by implementing the USB Communications Device Class, Abstract Control Model (CDC/ACM) specification. All major operating systems, including Windows, Linux, and MacOS support such devices out-of-the-box.

An AT command set is provided to query tags, read and write data, and configure the device itself.

3.1 Port settings

The device represents itself as a virtual COM port. As such, port settings, like communication speed, parity or stop bit settings will be ignored. No specific configuration is necessary and the port can be operated at any settings (without influencing the communication speed).

3.2 General Frame Format

- Host-to-Device data must start with an `AT` keyword and end with a carriage-return character (ASCII code 13 decimal, 0x0d hexadecimal, *"\r"* as C string literal)
- Device-to-Host frames start and end with a carriage-return character followed by a line-feed character (ASCII codes 13,10 decimal, 0x0d,0x0a hexadecimal, *"\r\n"* as C string literal).
- Device-to-Host response to a Host-to-Device request consists of one or more frames, the last one being `\r\nOK\r\n` or `\r\nERROR\r\n`, depending on whether the request/command has been completed successully or failed.
- Do **not** add any unnesseccary white space characters (space, back-space, tab, carriage-return, line-feed, etc.) to the request data, they will not be removed by the command-line parser leading to *"ERROR"* response.

3.3 Device modes

RFID device can be in two modes: scanning (`SCAN1`) and application controlled mode (`SCAN0`). The default mode (i.e. the device mode after POR) is selected via the device settings. Factory defaults set it to *scanning* mode.

- `SCAN1` - the device continuously scans whether an RFID tag is present or not (actually it queries the tag presence once every N ms, factory default - 1000ms) and reports it to the PC:
  - `SCAN:+<HEX DATA>` when a new tag is detected
  - `SCAN:-<HEX DATA>` when a tag is no longer present
- `SCAN0` - the device does not scan for RFID tags on its own, only when a corresponding request arrives.

All commands except LED control, BUZZ control, settings related, and device reboot will fail with `ERROR` while in `SCAN1` mode

- (v. *1.0F Apr 3 2018* and above) `DATA` is determined by the UID and Block Number settings. If `DATA` is the UID, SAK is optionally appended based on the corresponding device setting.
- (prior to v. *1.0F Apr 3 2018*) `DATA` is the UID + SAK

3.4 Error codes

When a tag reports an error, an additional frame describing the error can be returned before the `ERROR` frame. This additional frame has the following format:

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

where code is the last error reported by RFID IC. It is a 32bit unsigned integer in decimal format with the following bit fields:

- `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

3.5 CDC/ACM AT Command Set. API Referece.

3.5.1 Device Information

This command queries the product information and firmware version of the device.

Syntax:

Request	`ATI\r`
Response	`<product descrition> <firmware version/build date>`
	`S/N <serial number>`

Example:

Request	`ATI\r`
Response	`Open-Development RFID Reader (CDC-AT)1.0F Jan 17 2018`
	`S/N 220333635434B431500280010`

3.5.2 LED Control

This command controls the state of the light-emitting diode (LED).

Syntax:

Request	`AT+D1=[0|1|2|nil]\r`	Changes the state of the LED to
		0=off, *1=on*, *2=blinking*.
		If no state is specified (i.e. `nil`),
		the application relinquishes control to the firmware
Response	`OK`	

Example:

Request	`AT+D1=1\r`	Turn the LED on
Response	`OK`	
Request	`AT+D1=\r`	Return control to the firmware
Response	`OK`	
Request	`AT+D1?\r`	Query LED state
Response	`+D1=1`	LED is on
	`OK`	

3.5.3 Buzzer Control

This command controls the state of the buzzer

Syntax:

Request	`AT+B=[0|1|nil]\r`	Changes the state of the buzzer to
		0=off, *1=on*.
		If no state is specified (i.e. `nil`),

	the application relinquishes control to the firmware
Response	`OK`

Example:

Request	`AT+B=1\r`	Turn the buzzer on
Response	`OK`	
Request	`AT+B=\r`	Return control to the firmware
Response	`OK`	
Request	`AT+B?\r`	Query the buzzer state
Response	`+B=1`	Buzzer is on
	`OK`	

3.5.4 RFID Mode Control

This command switches the device mode.

Syntax:

Request	`AT+SCAN[0|1]\r`
Response	`OK`

Example:

Request	`AT+SCAN0\r`	
Response	`OK`	Scanning disabled

3.5.5 Enable/Disable RF Field

This command switches the RF field, generated by the IC, *on* or *off*

Syntax:

Request	`AT+RF=[0|1]\r`	`1` - switch on
		`0` - switch off
Response	`OK`	

Example:

Request	`AT+RF=0\r`	
Response	`OK`	RF field is off
Request	`AT+RF?\r`	Query the RF field state
Response	`+RF=0`	RF field is currently off
	`OK`	

3.5.6 Inventory scan

Requests an inventory scan. Returns a list of UIDs (may be empty) of active tags. All the discovered tags are scanned and halted/deselected.

Syntax:

Request	`AT+I\r`	
Response	`+UID=<HEX UID><SAK>`	possibly repeated or absent
	`OK`	

Example:

Request	`AT+I\r`
Response	`+UID=EC6D140708`

	+UID=343D7091725D8600
	OK

3.5.7 Inventory Scan (without Anti-Collision)

Requests an inventory scan without anti-collision (selects and returns the first detected tag, if any).

Syntax:

Request	AT+i\r	
Response	+UID=<HEX UID><SAK>	possibly absent
	OK	

Example:

Request	AT+i\r
Response	+UID=EC6D140708
	OK

3.5.8 Inventory Scan (select next tag)

Halts the currently selected tag and requests an inventory scan without anti-collision (i.e select the next tag)

Syntax:

Request	AT+n\r	
Response	+UID=<HEX UID><SAK>	possibly absent
	OK	

Example:

Request	AT+n\r

Response	+UID=343D7091725D8600
	OK

3.5.9 Select Tag

Select a single tag by UID for further processing. The tag access commands (read/write block) address the *current* tag. *current* means either the first one according to the anti-collision algorithm, or the one, selected with this command. Passing an empty UID will deselect the current tag (if any).

Syntax:

Request	AT+SELECT=<HEX UID>\r	
Response	OK	Tag selected
	ERROR	Tag is not present

Example:

Request	AT+SELECT=343D7091725D86\r	
Response	OK	Tag selected
Request	AT+SELECT=343D7091725D87\r	
Response	ERROR	Tag is not present

3.5.10 Check Tag Presence

Check whether a tag with the specified UID is currently present. Unlike the AT+SELECT=... command this one just checks the tag presence, it does select the tag and does not make it the current one.

Syntax:

Request	AT+C=<HEX UID>\r	
Response	OK	Tag is present
	ERROR	Tag is not present

Example:

Request	`AT+C=343D7091725D86\r`	
Response	`OK`	Tag is present
		The current UID has not changed!

Request	`AT+C=343D7091725D87\r`	
Response	`ERROR`	Tag is not present

3.5.11 Get Current Tag Information

Request detailed information about the current tag: UID, block size, block count, type. UID us returned in hexadecimal format, BS (block size), BC (block count), T (type) in decimal format.

Recognized tag types

- 0 - Classic 1K
- 1 - Classic 4K
- 2 - Classic Mini
- 3 - Ultralight
- 4 - Ultralight C
- 5 - Ultralight C EV1 (640 bits)
- 6 - Ultralight C EV1 (1312 bits)
- 7 - Plus S 2K
- 8 - Plus S 4K
- 9 - DesFire 2K
- 10 - DesFire 4K
- 11 - DesFire 8K
- 12 - Classic 2K
- 13 - Plus X 2K
- 14 - Plus X 4K
- 15-254 RFU
- 255 - Unknown

Syntax:

Request	`AT+S\r`

Response	+UID=<HEX UID>,BC=<bc>,BS=<bs>,T=<t>	
	OK	
	ERROR	tag did not respond
		or SCAN1 mode is enabled

Example:

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

3.5.12 Read Data Block

Read data block from tag memory. The block number must be within the range supported by the tag (see **BC** field in the AT+S\r request), and passed in decimal format. Returns the block number, followed by the actual data in hexadecimal format. Number of bytes in the data equals the tag block size (see **BS** field in the AT+S\r request). A tag must be selected prior to executing this command (either by calling AT+i,AT+S, or AT+SELECT)

Syntax

Request	AT+R0\r	
Response	+DATA <number of bytes>:<hex data>	
	OK	
	+CME ERROR: <code>	tag did not respond, reported an error
	ERROR	or SCAN1 mode is enabled

Example

Request	AT+R0\r
Response	+DATA 0:EC6D1407920804009944314230353913

OK

3.5.13 Write Data Block

Write data block to tag memory. The block number must be within the range supported by the tag (see **BC** field in the AT+S\r request), and passed in decimal format. The data must be passed in hexadecimal format. The length of the data must match the size of a data block as returned in the **BS** parameter (i.e. pass *2*BS* hexadecimal characters).

Syntax

Request	AT+W<block number>:<HEX DATA>	
Response	OK	
	+CME ERROR: <code>	write or auth error (if +CME... is reported)
	ERROR	syntax error or SCAN1 mode is enabled (if no +CME...)

Example

Request	AT+W1:000102030405060708090A0B0C0D0E0F\r
Response	OK

3.5.14 Device Settings

The following commands change various device settings. The are executed regardless of the device mode. Individual commands change only in-RAM settings. In order to save the current settings to ROM, execute the AT+P command.

Enable/Disable the LED

This command enables/disables the LED. This setting only affects whether the firmware switches the LED when a tag is detected/lost. Regardless of this setting, you can switch the LED on and off using AT+D1... command.

Syntax:

Request	`AT+L[0|1|?]\r`	`?` - query the current value
		`0` - do not switch LED on/off
		`1` - switch LED on/off
Response	`OK`	

Example:

Request	`AT+L?\r`	
Response	`+L=1`	LED will be switched on/off
	`OK`	

Enable/Disable the Buzzer

This command enables/disables the buzzer. This setting only affects whether the firmware will make a *beep* when a tag is detected or not. Regardless of this setting, you can switch the buzzer on and off using `AT+B...` command.

Syntax:

Request	`AT+Z[0|1|?]\r`	`?` - query the current value
		`0` - do not make a *beep*
		`1` - make a *beep*
Response	`OK`	

Example:

Request	`AT+Z?\r`	
Response	`+Z=1`	The device will beep when a tag is detected
	`OK`	

RFID Receiver Gain

The recevier gain is measured in *dBm*. The valid range is `[18;48]dBm`. The default value is `33dBm`.

The valid gain values are

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

If any other value is passed, it will be rounded (to the nearest available value greater than the one pased).

Syntax:

Request	`AT+G=[18-48]\r`	? - query the current value
		+G= - revert to default (33)
Response	`OK`	

Example:

Request	`AT+G=24\r`	
Response	`OK`	The value will be rounded
		The value will not be applied immediately

Request	`AT+G?\r`	
Response	`+G=33`	The value (24) has been rounded.
	`OK`	

Tag Presence Query Frequency

When in `SCAN1` mode, the device queries the tag presence every N ms (default value: 1000ms). This command controls the query frequency (i.e. the interval N). Valid range is [250-65535] ms. Values smaller than the minimum one will be accepted and silently increased to fit in the range. Values greater then the aximum one will result in an error.

Syntax:

Request	`AT+T[?|<value>]\r`	`?` - query the current value
Response	`OK`	

Example:

Request	`AT+T?\r`	
Response	`+T=1000`	The current interval value is 1000ms
	`OK`	

Request	`AT+T1500\r`	
Response	`OK`	The current interval value set to 1500ms

Mifare Authentication Key

Mifare Classic tags need an authentication key to be read/written. The following command sets the key (the key type and 6 bytes long key itself), which will be used to authenticate connected Mifare Classic tags. The default value is `FF`

Syntax:

Request	`AT+K<type><hex data>\r`	type can be `A` or `B`
Response	`OK`	

Example:

Request	`AT+KA000102030405\r`	
Response	`OK`	Key set to `00..05` type `A`
Request	`AT+K?`	
Response	`+A000102030405\r`	returns current key type and value

	OK

SAK

(for version *1.0F Apr 3 2018* and above)

By default Mifare SAK (Select Acknowledge Code) is appended to the UID in the scanning mode. This can be disabled/enabled using the `+A` command.

Syntax:

Request	`AT+A[0|1|?]\r`	`?` - query the current value
		`0` - do not append SAK
		`1` - append SAK
Response	`OK`	

Example:

Request	`AT+A?\r`	
Response	`+A=1`	SAK will be appended
	`OK`	

Data selection (scanning mode)

(for version *1.0F Apr 3 2018* and above)

This setting controls data (UID or block) printed when a new tag is detected. `1` - UID, `0` - block, determined by the Block Number setting

Syntax:

Request	`AT+U[0|1|?]\r`	`?` - query the current value
		`0` - print block
		`1` - print UID (the default)
Response	`OK`	

Example:

Request	`AT+U?\r`	
Response	`+U=1`	UID will be printed
	`OK`	

Block number selection (scanning mode)

(for version *1.0F Apr 3 2018* and above)

In scanning mode the device continuously scans whether an RFID tag is present and prints UID or the selected block. This setting controls the block number being printed:

If the block does not exist or cannot be read (ex. due to an authorization failure) nothing is printed.

Syntax:

Request	`AT+b[?|<value>]\r`	? - query the current value
Response	`OK`	

Example:

Request	`AT+b?\r`	
Response	`+b=0`	Block 0 contents will be printed
	`OK`	

Request	`AT+b1\r`	
Response	`OK`	Block 1 contents will be printed

Write settings to ROM

This command saves the current in-RAM setting to ROM, i.e. makes them permanent

Syntax:

Request	`AT+P\r`

Response	`OK`

3.5.15 Device Control

The following commands manage the RFID reader device itself:

Device Reboot

Reboot the device

Syntax:

Request	`AT+Q\r`
Response	`OK`

Firmware Update

Reboot the device and start the bootloader (this command may not be supported by some devices. If not supported, acts as `AT+Q\r`).

Syntax:

Request	`AT+X\r`
Response	`OK`

4 HID Firmware

The HID firmware acts as a USB HID Keyboard and a Vendor-Specific HID device. The device scans for new tags and types the UID (or UID+SAK or the specified block) of a found tag by using virtual keystrokes.

4.1 Device modes

RFID device can be in two modes: scanning and application controlled mode. The default mode (i.e. then device mode after POR) is selected via the device settings. Factory defaults set it to *scanning*

mode.

- When in scanning mode the device scans for new tags and types the UID (UID+SAk / block) of a found tag by using virtual keystrokes. One can control how the data is printed via the device settings (upper case / lower case, carriage return, etc). The tag presence is queried once every N ms. Factory default is 1000ms. Custom HID commands (except the one switching the mode and controlling the device settings) are ignored in the *scanning* mode.
- When the device in the application controlled mode, the user controlls the device behavior via custom HID commands.

4.2 HID Reports

This firmware has two reports

- Report `0` is used to send keyboard events when the device is in scanning mode and follows the standard HID Keyboard Report foramt
- Report `1` is used to configure the device, switching scan modes and query/read/write tags. Both *IN-* and *OUT* reports with ID `1` always consist of 63 bytes.

4.3 Custom HID commands

Custom HID commands are sent and received via HID report with ID `1`. The report always consists of 63 bytes. The report ID is prepended to the data, therefore the full report length is 64 bytes. The first byte (after the ID one) is the command ID. Valid commands are listed below. The next bytes are command parameters. If the command parameters occupy less than 62 bytes, the remaining unused bytes are padding and can have any arbitrary values.

The device answers with a report, where the first byte (after the report ID) is the command id (the same as the request), the next 4 bytes are the error code (32bit unsigned integer, little endian), and the following bytes are command specific. If the command parameters occupy less than 62 bytes, the remaining unused bytes are padding and can have any arbitrary values.

The error code has the following bit fields:

- `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

The error code of `0` is `OK` (i.e. *no error*). **ATTENTION:** the data following the error code bytes is valid only if the error code is 0.

Command list:

- `0x01` - Device Information
- `0x02` - LED Control
- `0x03` - Buzzer Control
- `0x10` - Inventory Scan
- `0x11` - Inventory Scan (w/o Anti-Collision)
- `0x12` - Inventory Scan (select next tag)
- `0x13` - Check Tag Presence
- `0x20` - Select Tag
- `0x21` - Tag Details
- `0x22` - Read Data Block
- `0x23` - Write Data Block
- `0x30` - Settings: LED
- `0x31` - Settings: Buzzer
- `0x32` - Settings: Gain
- `0x33` - Settings: Query Frequency
- `0x34` - Settings: Mifare Key
- `0x35` - Settings: UID Case (KBD)
- `0x36` - Settings: UID Carriage Return (KBD)
- `0x37` - Settings: SAK (KBD)
- `0x38` - Settings: Scan Type
- `0x39` - Settings: Scan Block Number
- `0x3F` - Settings: Save to ROM
- `0x40` - RFID Antenna State
- `0x41` - RFID Mode Control
- `0x80` - Reboot Device
- `0x81` - Reboot Device (DFU)
- Other values are **RFU**

Abberviations:

- *IN* - device to host
- *OUT* - host to device

Several commands have one 1byte parameter (*bValue*), that controlls a device state or setting. It can be one of the following:

`QUERY`	0	query the current state/value
`ON`	1	enable
`OFF`	2	disable
`SWITCH`	3	switch state
`DEFAULT`	4	relinquish control to the firmware
`-----`	other values	treated as `QUERY`

4.3.1 Device Information

- Command code: 0x01
- Params (OUT): none
- Params (IN):
  - 4 bytes: error code
  - 1 byte: firmware info length
  - N bytes: firmware info

This command queries the firmware version of the device.

4.3.2 LED Control

- Command code: 0x02
- Params (OUT): *bValue*
  - `ON` - LED on
  - `OFF` - LED off
  - `SWITCH`- blink
  - `DEFAULT` - relinquish control to the firmware
  - other - query state
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current LED state (1 - on, 0 - off)

4.3.3 Buzzer Control

- Command code: 0x03
- Params (OUT): *bValue*
  - `ON` - buzzer on
  - `OFF` - buzzer off
  - `SWITCH`,`DEFAULT` - relinquish control to the firmware
  - other - query state
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current buzzer state (1 - on, 0 - off)

4.3.4 RFID Mode Control

- Command code: 0x41
- Params (OUT): *bValue*
  - `ON` - enable scan mode
  - `OFF` - disable scan mode
  - `QUERY` - query the current mode
  - other - **error** (will return `INVARG` error)
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current mode (1 - scan mode enabled, 0 - scan mode disabled)

4.3.5 Enable/Disable RF Field

- Command code: 0x42
- Params (OUT): *bValue*
  - `ON` - enable RF field
  - `OFF` - disable RF field
  - other - query the current mode
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current RF field state (1 - enabled, 0 - disabled)

4.3.6 Inventory Scan

- Command code: 0x10

- Params(OUT): none

One or mode *IN* reports will be sent by the device. Params: * 4 byte: error code * 1 byte: *L* - UID length (0 for the last report) * *L* bytes: UID * 1 byte: SAK

4.3.7 Inventory Scan (without Anti-Collision)

- Command code: 0x11
- Params (OUT): none
- Params (IN):
  - 4 bytes: error code
  - 1 byte: *L* - UID length
  - *L* bytes: UID
  - 1 byte: SAK

4.3.8 Inventory Scan (select next tag)

- Command code: 0x12
- Params (OUT): none
- Params (IN):
  - 4 bytes: error code
  - 1 byte: *L* - UID length
  - *L* bytes: UID
  - 1 byte: SAK

4.3.9 Check Tag Presence

- Command code: 0x13
- Params (OUT): none
- Params (IN):
  - 4 bytes: error code
  - 1 byte: *L* - UID length
  - *L* bytes: UID
  - 1 byte: SAK

4.3.10 Select Tag

- Command code: 0x20

- Params (OUT): none
- Params (IN):
  - 4 bytes: error code

4.3.11 Get Current Tag Information

- Command code: 0x21
- Params (OUT): none
- Params (IN):
  - 4 bytes: error code
  - 1 byte: *L* - UID length (0 for the last report)
  - *L* bytes: UID
  - 1 byte: SAK
  - 2 bytes: block count (16bit unsigned integer, little endian)
  - 1 byte: block size
  - 1 byte: tag type

Recognized tag types

- 0 - Classic 1K
- 1 - Classic 4K
- 2 - Classic Mini
- 3 - Ultralight
- 4 - Ultralight C
- 5 - Ultralight C EV1 (640 bits)
- 6 - Ultralight C EV1 (1312 bits)
- 7 - Plus S 2K
- 8 - Plus S 4K
- 9 - DesFire 2K
- 10 - DesFire 4K
- 11 - DesFire 8K
- 12 - Classic 2K
- 13 - Plus X 2K
- 14 - Plus X 4K
- 15-254 RFU
- 255 - Unknown

4.3.12 Read Data Block

A tag must be selected prior to executing this command (either by calling Inventory Scan w/o Anti-Collision, Inventory Scan - next tag, Select Tag, or Tag Info). The block number must be within the range supported by the tag.

- Command code: 0x22
- Params (OUT): 1 byte block number
- Params (IN):
  - 4 bytes: error code
  - 1 byte: block number
  - *N* bytes: block data (*N* is equal to the value of the *block size* field returned by the tag info command)

4.3.13 Write Data Block

A tag must be selected prior to executing this command (either by calling Inventory Scan w/o Anti-Collision, Inventory Scan - next tag, Select Tag, or Tag Info). The block number must be within the range supported by the tag. Data size must be equal to the block size value returned by the tag info command.

- Command code: 0x23
- Params (OUT):
  - 1 byte: block number
  - 1 byte: data size (*N*)
  - N bytes: data
- Params (IN):
  - 4 bytes: error code
  - 1 byte: block number

4.3.14 Device Settings

The following commands change various device settings. The are executed regardless of the scanning mode. Individual commands change only in-RAM settings. In order to save the current settings to ROM, execute the Settings Write command.

Enable/Disable the LED

This command enables/disables the LED. This setting only affects whether the firmware switches the LED when a tag is detected/lost. Regardless of this setting, you can switch the LED on and off using LED Control command.

- Command code: 0x30
- Params (OUT): *bValue*
  - `ON` - enable LED switching
  - `OFF` - disable LED switching
  - other - query the current setting
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current LED control setting (1 - enabled, 0 - disabled)

Enable/Disable the Buzzer

This command enables/disables the buzzer. This setting only affects whether the firmware switches the buzzer on when a tag is detected or not. Regardless of this setting, you can switch the buzzer on and off using BUZZ Control command.

- Command code: 0x31
- Params (OUT): *bValue*
  - `ON` - enable buzzer
  - `OFF` - mute buzzer
  - other - query the current setting
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current buzzer control setting (1 - enabled, 0 - disabled)

RFID Receiver Gain

The recevier gain is measured in *dBm*. The valid range is `[18;48]dBm`. The default value is `33dBm`.

The valid gain values are

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

If any other value is passed, it will be rounded (to the nearest available value greater than the one pased).

- Command code: 0x32
- Params (OUT): (1 byte) 0 - query the current value, >0 - set new gain value (in dBm)
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current gain value in dBm.

Tag Presence Query Frequency

When in scanning mode, the device queries the tag presence every N ms (default value: 1000ms). This command controls the query frequency (i.e. the interval N). Valid range is [250-65535] ms. Values smaller than the minimum one will be accepted and silently increased to fit in the range. Values greater then the aximum one will result in an error.

- Command code: 0x33
- Params (OUT): (2 bytes - 16bit unsigned little endian integer) 0 - query the current value, >0 - set new interval value (in ms)
- Params (IN):
  - 4 bytes: error code
  - 2 bytea: the current interval value in ms.

Mifare Authentication Key

Mifare Classic tags need an authentication key to be read/written. The following command sets the key (the key type and 6 bytes long key itself), which will be used to authenticate connected Mifare Classic tags. The default value is FF

- Command code: 0x34
- Params (OUT):
  - 1 byte: key type - “A” (ascii code 65) or “B” (ascii code 66). Any other value - query the current key and key type.
  - 6 bytes: key
- Params (IN):
  - 1 byte: the current key type - “A” (ascii code 65) or “B” (ascii code 66)
  - 6 bytes: the current key

UID in Uppercase/Lowercase (keyboard mode)

- Command code: 0x35

- Params (OUT): *bValue*
  - `ON` - print UID in upper case
  - `OFF` - print UID in lower case
  - other - query the current setting
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current case control setting (1 - upper case, 0 - lower case)

Append/Omit Carriage Return (keyboard mode)

- Command code: 0x36
- Params (OUT): *bValue*
  - `ON` - print carriage return after UID
  - `OFF` - do not print carriage return after UID
  - other - query the current setting
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current CR control setting (1 - append, 0 - omit)

Append/Omit SAK (keyboard mode)

- Command code: 0x37
- Params (OUT): *bValue*
  - `ON` - append SAK to UID
  - `OFF` - do not append SAK to UID
  - other - query the current setting
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current SAK control setting (1 - append, 0 - omit)

Data selection (scanning mode)

This setting controls data (UID or block) printed when a new tag is detected.

- Command code: 0x38
- Params (OUT): 1 byte value
  - `1` - print UID
  - `2` - print the contents of a block determined by the Block Number
  - other - query the current setting

- Params (IN):
  - 4 bytes: error code
  - 1 byte: current value

Block number selection (scanning mode)

In scanning mode the device continuously scans whether an RFID tag is present and prints UID or the selected block. This setting controls the block number being printed:

If the block does not exist or cannot be read (ex. due to an authorization failure) nothing is printed.

- Command code: 0x39
- Params (OUT):
  - 1 byte: 0 - query, >0 - set
  - 1 byte: new block number (if `set` mode is selected)
- Params (IN):
  - 4 bytes: error code
  - 1 byte: current value

Write settings to ROM

This command saves the current in-RAM setting to ROM, i.e. makes them permanent

- Command code: 0x3F
- Params (OUT): none
- Params (IN): (4 bytes) error code

4.3.15 Device Control

The following commands manage the RFID reader device itself:

Device Reboot

Reboot the device.

- Command code: 0x80
- Params (OUT): none
- Params (IN): (4 bytes) error code

Firmware Update

Reboot the device and start the bootloader (this command may not be supported by some devices. If not supported, acts as Device Reboot).

- Command code: 0x81
- Params (OUT): none
- Params (IN): (4 bytes) error code

4.3.16 Other commands - RFU

*Params (IN) (4 bytes) error code `0x800` - Generic communication error.

5 Revisions

Revision	Date	Remarks
1.0	2018-01-22	Initial Version
1.1	2018-04-27	Protocol update • Support UID / Block selection in scan mode

6 About

Open Developent LLC open-dev.ru

Pandoc theme: *EisVogel* by Pascal Wagler and John MacFarlane