**SDK Instruction Manual for External Reader-writer of**

**Shanghai Sunmi Technology Co., Ltd**

Revision 2.0.0

Shanghai Sunmi Technology Co., Ltd

**Developer Data**

Contents

[1 Introduction 4](#_Toc493751602)

[2 Applicable Environment 4](#_Toc493751603)

[3 Performance parameters of the reader-writer 4](#_Toc493751604)

[4 SDK Document 5](#_Toc493751605)

[5 Development configuration of SDK document 5](#_Toc493751606)

[6 Function definition 7](#_Toc493751607)

[6.1 open reader-writer 7](#_Toc493751608)

[6.2 close reader-writer 7](#_Toc493751609)

[6.3 check whether the reader-writer has been opened or not 7](#_Toc493751610)

[6.4 reset reader-wirter 7](#_Toc493751611)

[6.5 reset the system 8](#_Toc493751612)

[6.6 control the buzzer 8](#_Toc493751613)

[6.7 control the buzzing time of the buzzer 8](#_Toc493751614)

[6.8 control LED light 9](#_Toc493751615)

[6.9 control the flashing of LED light 9](#_Toc493751616)

[6.10 control the radio-frequency singnal 9](#_Toc493751617)

[6.11 obtain the model of the reader-writer 10](#_Toc493751618)

[6.12 obtain the software version of the reader-writer 10](#_Toc493751619)

[6.13 obtain the hardware version of the reader-writer 10](#_Toc493751620)

[6.14 ISO15693 tag query 11](#_Toc493751621)

[6.15 ISO15693 tag’s reading block data 11](#_Toc493751622)

[6.16 ISO15693 tag’s writing block data 12](#_Toc493751623)

[6.17 ISO15693 tag’s locking data block 12](#_Toc493751624)

[6.18 ISO15693 tag’s writing AFI value 12](#_Toc493751625)

[6.19 ISO15693 tag’s locking AFI value 13](#_Toc493751626)

[6.20 ISO15693 tag’s writing DSFID value 13](#_Toc493751627)

[6.21 ISO15693 tag’s locking DSFID value 14](#_Toc493751628)

[6.22 ISO15693 tag’s obtaining information 14](#_Toc493751629)

[6.23 transparent transmission of ISO15693 tag’s user command frame 14](#_Toc493751630)

[6.24 ISO14443A tag query 15](#_Toc493751631)

[6.25 ISO14443A M1 authentication key 15](#_Toc493751632)

[6.26 Read ISO14443A M1 block data 16](#_Toc493751633)

[6.27 write ISO14443A M1 block data 16](#_Toc493751634)

[6.28 ISO14443A M1 initialized purse value 17](#_Toc493751635)

[6.29 read ISO14443A M1 purse value 17](#_Toc493751636)

[6.30 ISO14443A M1 added purse value 17](#_Toc493751637)

[6.31 ISO14443A M1 deducted purse value 18](#_Toc493751638)

[6.32 ISO14443A M1 backup purse value 18](#_Toc493751639)

[6.33 read ISO14443A M0 data page 18](#_Toc493751640)

[6.34 write ISO14443A M0 data page 19](#_Toc493751641)

[6.35 ISO14443A CPU’s sending RATS command 19](#_Toc493751642)

[6.36 ISO14443A CPU’s sending APDU command 19](#_Toc493751643)

[6.37 ISO14443A tag suspend 20](#_Toc493751644)

[6.38 ISO14443A tag’s deselect 20](#_Toc493751645)

[6.39 transparent transmission of ISO14443A tag’s user command frame 20](#_Toc493751646)

[6.40 ISO14443B tag query 20](#_Toc493751647)

[6.41 ISO14443B tag’s sending APDU command 21](#_Toc493751648)

[6.42 transparent transmission of ISO14443B tag’s user command frame 21](#_Toc493751649)

[6.43 ISO14443B tag suspend 22](#_Toc493751650)

[6.44 felica tag query 22](#_Toc493751651)

[6.45 transparent transmission of felica tag’s user command frame 22](#_Toc493751652)

[6.46 detect card slot 23](#_Toc493751653)

[6.47 control card slot 23](#_Toc493751654)

[6.48 send APDU command to ISO7816 24](#_Toc493751655)

[6.49 read magnetic strip card data 24](#_Toc493751656)

[6.50 SLE4418/SLE4428/SLE5518/SLE5528 card’s select 25](#_Toc493751657)

[6.51 SLE4418/SLE4428/SLE5518/SLE5528 card’s reading data 25](#_Toc493751658)

[6.52 SLE4418/SLE4428/SLE5518/SLE5528 card’s writing data 26](#_Toc493751659)

[6.53 SLE4428/SLE5528 card’s count of reading remaining errors 26](#_Toc493751660)

[6.54 SLE4418/SLE4428/SLE5518/SLE5528 card’s reading protected area data 26](#_Toc493751661)

[6.55 SLE4418/SLE4428/SLE5518/SLE5528 card’s writing protected area data 27](#_Toc493751662)

[6.56 SLE4428/SLE5528 card’s reading secret key 27](#_Toc493751663)

[6.57 SLE4428/SLE5528 card’s writing secret key 28](#_Toc493751664)

[6.58 SLE4428/SLE5528 card’s authentication key 28](#_Toc493751665)

[6.59 SLE4432/SLE4442/SLE5532/SLE5542 card’s select 28](#_Toc493751666)

[6.60 SLE4432/SLE4442/SLE5532/SLE5542 card’s reading data 28](#_Toc493751667)

[6.61 SLE4432/SLE4442/SLE5532/SLE5542 card’s writing data 29](#_Toc493751668)

[6.62 SLE4442/SLE5542 card’s count of reading remaining errors 29](#_Toc493751669)

[6.63 SLE4432/SLE4442/SLE5532/SLE5542 card’s reading protected area data 29](#_Toc493751670)

[6.64 SLE4432/SLE4442/SLE5532/SLE5542 card’s writing protected area data 30](#_Toc493751671)

[6.65 SLE4442/SLE5542 card’s reading secret key 30](#_Toc493751672)

[6.66 SLE4442/SLE5542 card’s writing secret key 31](#_Toc493751673)

[6.67 SLE4442/SLE5542 card’s authentication key 31](#_Toc493751674)

[6.68 AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 card’s select 31](#_Toc493751675)

[6.69 AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 card’s reading data 32](#_Toc493751676)

[6.70 AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 card’s writing data 32](#_Toc493751677)

[6.71 AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 card’s select 32](#_Toc493751678)

[6.72 AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 card’s reading data 33](#_Toc493751679)

[6.73 AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 card’s writing data 33](#_Toc493751680)

[6.74 AT88SC102 card’s select 33](#_Toc493751681)

[6.75 AT88SC102 card’s reading data 34](#_Toc493751682)

[6.76 AT88SC102 card’s writing data 34](#_Toc493751683)

[6.77 AT88SC102 card’s authentication key 34](#_Toc493751684)

[6.78 AT88SC102 card fuse wire 35](#_Toc493751685)

[6.79 AT88SC1604 card’s select 35](#_Toc493751686)

[6.80 AT88SC1604 card’s reading data 35](#_Toc493751687)

[6.81 AT88SC1604 card’s writing data 36](#_Toc493751688)

[6.82 AT88SC1604 card’s authentication key 36](#_Toc493751689)

[6.83 AT88SC1604 card fuse wire 37](#_Toc493751690)

[6.84 AT45DB card’s select 37](#_Toc493751691)

[6.85 AT45DB card’s reading data 37](#_Toc493751692)

[6.86 AT45DB card’s writing data 37](#_Toc493751693)

[7 error code 38](#_Toc493751694)

# Introduction

This manual defines & describes the usage method for Sunmi T1’s external USB reader-writer SDK, its content includes:

* Function definition；
* Error code definition;

# Applicable Environment

This SDK（hereinafter referred to as SDK）is applicable to ：

* Android 3.0 and above Android 3.0 version

Hardware environment:

* T1 device of Shanghai Sunmi Technology Co., Ltd
* UM002 reader-writer of Shanghai Esystech
* T10 reader-writer of Shenzhen Deka Technology Co., Ltd
* ACR1281U-C1 reader-writer of Hongkong Long Jie Smart Card Co., Ltd
* ACR1281U-K1 reader-writer of Hongkong Long Jie Smart Card Co., Ltd

# Performance Parameters of Reader-writer

* UM002 card reader of Shanghai Esystech

Operating frequency：13.56MH

Supported types of non-contact protocols：ISO14443A protocol、ISO14443B protocol、ISO15693 protocol and ISO18092 protocol

Communication mode：USB HID port, driver free

Card reading distance：50mm~100mm

Card reading time：<200ms

Power supply mode：supplied from USB port 5V

Supported types of contact protocols：ISO7816（namely one PSAM card slot, power supply voltage：5V，communication BPS：9600bps）

Contact person：Mingyue Yang

Contact phone number：13386007233

* T10 reader-writer of Shenzhen Deka Technology Co., Ltd

Operating frequency：13.56MH

Supported types of non-contact protocols：ISO14443A protocol and ISO14443B protocol

Communication mode：USB HID port, driver free

Card reading distance：50mm~100mm

Card reading time：<200ms

Power supply mode：external power supply 9V

Supported types of contact protocols：ISO7816 protocol（namely 4 PSAM card slots, power supply voltage：5V 3.3V 1.8V optional, communication BPS：

9600bps 19200bps 38400bps 57600bps 115200bps optional）

Support plug-in card slot：ISO7816 protocol（power supply voltage：5V 3.3V 1.8V optional, communication BPS：9600bps 19200bps 38400bps

57600bps 115200bps optional）

Support contact-type storage card：namely plug-in card slot, SLE4428、SLE4442、AT24C01、AT24C02、AT24C04、AT24C08、AT24C16、AT24C64、

AT88SC102、AT88SC1604、AT45DB

Other supported cards：magnetic strip card

Contact person：Chen Song

Contact phone number：13311667181

* ACR1281U-C1 reader-writer of Hongkong Long Jie Smart Card Co., Ltd

Operating frequency：13.56MH

Supported types of non-contact protocols：ISO14443A protocol（not supporting CPU card）

Communication mode：USB PC/SC port, driver free

Card reading distance：50mm~100mm

Card reading time：<200ms

Power supply mode：supplied from USB port 5V

Support plug-in card slot：ISO7816 protocol Class A, B and C

Support contact-type storage card：namely plug-in card slot，SLE4428、SLE4442、AT24C01、AT24C02、AT24C04、AT24C08、AT24C16、AT24C32、

AT24C64, etc.

* ACR1281U-K1 reader-writer of Hongkong Long Jie Smart Card Co., Ltd

Operating frequency：13.56MH

Supported types of non-contact protocols：ISO14443A protocol

Communication mode：USB PC/SC port, driver free

Card reading distance：50mm~100mm

Card reading time：<200ms

Power supply mode：supplied from USB port 5V

Supported types of contact protocols：ISO7816 protocol Class A, B and C（namely one PSAM card slot）

Support plug-in card slot：ISO7816 protocol Class A, B and C

Support contact-type storage card：namely plug-in card slot，SLE4428、SLE4442、AT24C01、AT24C02、AT24C04、AT24C08、AT24C16、AT24C32、

AT24C64, etc.

# SDK Document

This SDK（hereinafter referred to as SDK）includes the following documents：

* acssmc-1.1.3.jar：Driver java package of Shanghai Sunmi Technology Co., Ltd’s T1 device: external PC/SC reader-writer.
* SunmiReader.jar：Development java package of Shanghai Sunmi Technology Co., Ltd’s T1 device: external reader-writer SDK.

# Development Configuration of SDK Document

Take Android Studio project as an example as follows：

* Enter app directory according to directory hierarchy as shown in the the following figure

****

* Create new libs directory according to app directory hierarchy as shown in the the following figure (part circled in red)

****

* Place SunmiReader.jar and acssmc-1.1.3.jar development package under newly created libs directory, as shown in the following figure:

****

* Finally, import in turn SunmiReader.jar and acssmc-1.1.3.jar development package into the project, right click on SunmiReader.jar or acssmc-1.1.3.jar development package, then click Add As Library in the pop-up menu，as shown in the following figure：

** **

# Function Definition

## Open Reader-writer

|  |  |
| --- | --- |
| Function prototype | int open (Context context) |
| Function description | This function is used to open reader-writer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| Context context | Context | Omitted |

 |
| Return value | Return type | int |
| Return value | 0: succeed in opening reader-writer-1: reader-writer already opened -2: failure in opening reader-writer  |
| Instructions | This function must be executed before operating IC card or controlling reader-writer, and it is only necessary to successfully execute once |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Close Reader-writer

|  |  |
| --- | --- |
| Function prototype | void close () |
| Function description | This function is used to close reader-writer |
| Parameter description | None |
| Return value | None |
| Instructions | App can carry out repeated execution on this function. Invloking IC card or controlling the interface function of reader-writer will cause response error  |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Check Whether the Reader-writer Has Been Opened or Not

|  |  |
| --- | --- |
| Function prototype | boolean isOpen () |
| Function description | This function is used to check whether the reader-writer has been opened or not |
| Parameter description | None  |
| Return value | Return type | boolean |
| Return value | true：the reader-writer has been opened, and the communication work is OK.false：the reader-writer has been closed or not opened. |
| Instructions | If the reader-writer has explicitly used Open function to open successfully, but the connection is lost (e.g., the communication between the reader-writer & T1 is interrupted), this function will automatically close the reader-writer handle and release the memory space, then the function will return false. At this time, it is necessary for the App to explicitly use Open function and try once more to open the reader-writer.  |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Reset Reader-writer

|  |  |
| --- | --- |
| Function prototype | int reset () |
| Function description | This function is used to reset the reader-writer |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | After resetting the reader-writer system, it is necessary for the App to explicitly use Close function to close the reader-writer. Then, it is necessary for the App to explicitly use Open function to reopen the reader-writer. |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Reset the System

|  |  |
| --- | --- |
| Function prototype | int systemReset () |
| Function description | This function is used to reset the reader-writer system |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Control the Buzzer

|  |  |
| --- | --- |
| Function prototype | int beep (boolean beep) |
| Function description | This function is used to control buzzer on/off |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| boolean beep | Control mode of the buzzer | beep = true：turn on the buzzerbeep = false：turn off the buzzer |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | After turning on the buzzer, it is necessary to explicitly invoke interface to turn off the buzzer. |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Control the Buzzing Time of the Buzzer

|  |  |
| --- | --- |
| Function prototype | int beepTime (char time) |
| Function description | This function is used to control the buzzing time of the buzzer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| char time | Time | Buzzer time, its unit is mse.g.：time = 200 indicates 200ms |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | The buzzer begins to buzz until reaching the buzzer time, then the buzzer will be automatically turned off. |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Control LED Light

|  |  |
| --- | --- |
| Function prototype | int led (byte index, boolean mode) |
| Function description | This function is used to control LED light of the reader-writer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte index | Index | index = 0x00：all the indicator lightsindex = 0x01：processing indicator lightsindex = 0x02：status indicator lightsindex = 0x03：communication indicator lights |
| boolean mode | Control LED | mode = true：LED light is on mode = false：LED light is off |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Control the Flashing of LED Light

|  |  |
| --- | --- |
| Function prototype | int ledTwinkle (byte index) |
| Function description | This function is used to control the flashing of LED light of the reader-writer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte index | Index | index = 0x00：all the indicator lightsindex = 0x01：processing indicator lightsindex = 0x02：status indicator lightsindex = 0x03：communication indicator lights |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Control the Radio-frequency Signal

|  |  |
| --- | --- |
| Function prototype | int controlRf (byte mode) |
| Function description | This function is used to control the radio-frequency signal of the reader-writer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte mode | Control mode of radio-frequency signals | 0x00：turn off RF signal0x01：turn on RF signal0x02：reset RF signal (turn it off for 20ms，then turn it on again) |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | Successfully turn off RF signal, then successfully turn on RF signal, and reset IC card status within the RF range. Reset the RF signal and also IC card status within the RF range |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Obtain the Model of the Reader-writer

|  |  |
| --- | --- |
| Function prototype | int getReaderType (byte [] type) |
| Function description | This function is used to obtain the model of the reader-writer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] type | The model of the reader-writer | Allocated memory >= 50 byte，e.g.：byte [] type = new byte [50]. |

 |
| Return value | Return type | int |
| Return value | >= 0：effective data length of the reader-writer model（succeed in execution）< 0：failure in execution |
| Instructions | None |
| Inculded in | libSUNMIANDROIDUSBHIDREADER.so |

## Obtain the Software Version of the Reader-writer

|  |  |
| --- | --- |
| Function prototype | int getReaderSoftwareVersion (byte [] sv) |
| Function description | This function is used to obtain the software version of the reader-writer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] sv | Software version of the reader-writer | Allocated memory >= 50 byte，e.g.：byte [] sv = new byte [50]. |

 |
| Return value | Return type | int |
| Return value | >= 0：effective data length of the software version of the reader-writer (succeed in execution)< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Obtain the Hardware Version of the Reader-writer

|  |  |
| --- | --- |
| Function prototype | int getReaderHardwareVersion (byte [] hv) |
| Function description | This function is used to obtain the hardware version of the reader-writer |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] hv | Hardware version of the reader-writer | Allocated memory >= 50 byte，e.g.：byte [] hv = new byte [50]. |

 |
| Return value | Return type | int |
| Return value | >= 0：effective data length of the hardware version of the reader-writer (succeed in execution)< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag Query

|  |  |
| --- | --- |
| Function prototype | int iso15693CheckCard (boolean mode, byte [] uid) |
| Function description | This function is used to query & activate ISO15693 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| boolean mode | Query mode | false：normal modetrue：repeat mode, repeat the query about the tag that was got in the last query  |
| byte [] uid | UID of the tag | Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | To operate ISO15693 tag, it is necessary for the App to succeed first in explicitly using iso15693CheckCard function before using ISO15693 tag’s other interface functions |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Reading Block Data

|  |  |
| --- | --- |
| Function prototype | int iso15693ReadBlock (byte [] uid, byte addr, byte [] block) |
| Function description | This function is used to read the block data of specified ISO15693 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID  | Tag UID to read data.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |
| byte addr | Block address | Block address to read data. |
| byte [] block | Block data | Block data that has been readAllocated memory = 4 byte，e.g.：byte [] block = new byte [4]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Writing Block Data

|  |  |
| --- | --- |
| Function prototype | int iso15693WriteBlock (byte [] uid, byte addr, byte [] block) |
| Function description | This function is used to write block data to the specified ISO15693 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID  | Tag UID to write data.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |
| byte addr | Block address | Block address to write data. |
| byte [] block | Block data | Data to write.Allocated memory = 4 byte，e.g.：byte [] block = new byte [4]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Locking Data Block

|  |  |
| --- | --- |
| Function prototype | int iso15693LockBlock (byte [] uid, byte addr) |
| Function description | This function is used to lock the data block of ISO15693 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID  | Tag UID to lock data block.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |
| byte addr | Block address | Block address to be locked. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Writing AFI Value

|  |  |
| --- | --- |
| Function prototype | int iso15693WriteAfi (byte [] uid, byte afi) |
| Function description | This function is used to write AFI value to specified ISO15693 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID  | Tag UID to write AFI value.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |
| byte afi | AFI value | AFI value to write in |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Locking AFI Value

|  |  |
| --- | --- |
| Function prototype | int iso15693LockAfi (byte [] uid) |
| Function description | This function is used to lock ISO15693 tag’s AFI value  |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID | Tag UID to lock AFI value.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Writing DSFID Value

|  |  |
| --- | --- |
| Function prototype | int iso15693WriteDsfid (byte [] uid, byte dsfid) |
| Function description | This function is ued to write DSFID value to specified ISO15693 tag  |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID | Tag UID to write DSFID value.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |
| byte dsfid | DSFID value | DSFID value to write |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Locking DSFID Value

|  |  |
| --- | --- |
| Function prototype | int iso15693LockDsfid (byte [] uid) |
| Function description | This function is used to lock ISO15693 tag’s DSFID value |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID | Tag UID to lock DSFID value.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO15693 Tag’s Obtaining Information

|  |  |
| --- | --- |
| Function prototype | int iso15693ReadSystemInfo (byte [] uid, ISO15693Info info) |
| Function description | This function is used to obtain ISO15693 tag information |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] uid | Tag UID | Tag UID to obtain tag information.Allocated memory = 8 byte，e.g.：byte [] uid = new byte [8]. |
| ISO15693Info info | Tag information | Tag informationISO15693Info info = new ISO15693Info ();info.infoFlag：tag information flaginfo.dsfid：tag DSFIDinfo.afi：tag AFIinfo.blockNum：number of tag blockinfo.blockSize：size of tag blockinfo.ic：tag IC reference |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Transparent Transmission of ISO15693 Tag’s User Command Frame

|  |  |
| --- | --- |
| Function prototype | int iso15693Dtu (byte [] cmd, int cmdlen, byte [] rsp) |
| Function description | This function is used for transparent transmission of ISO15693 tag’s user command frame |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] cmd | Request frame | Request frame data |
| int cmdlen | Length | Effective length of request frame data |
| byte [] rsp | Response frame | Response frame dataAllocated memory >= 1024 byte，e.g.：byte [] rsp = new byte [1024]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of response frame data（succeed in execution）< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A Tag Query

|  |  |
| --- | --- |
| Function prototype | int iso14443ACheckCard (boolean mode, byte [] uid, byte [] type) |
| Function description | This function is used to query and activate ISO14443A tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| boolean mode | Query tag mode | false：read free tagtrue：read all the tags |
| byte [] uid | Tag UID | Allocated memory = 10 byte，e.g.：byte [] uid = new byte [10]. |
| byte [] type | Card type | Allocated memory = 2 byte，e.g.：byte [] type = new byte [2].0x0400：Mifare\_One(S50) card0x0200：Mifare\_One(S70) card 0x0800：Mifare\_Pro card0x0403：Mifare\_ProX card 0x4403：Mifare\_DESFire card 0x4400：Mifare\_UltraLight card |

 |
| Return value | Return type | int |
| Return value | > 0：UID effective length of the tag（succeed in execution）<= 0：failure in execution |
| Instructions | To operate ISO14443A tag, it is necessary for the App to succeed first in explicitly using iso14443CheckCard function before using ISO14443A tag’s other interface functions |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A M1 Authentication Key

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1Auth (byte keytype, byte addr, byte [] key) |
| Function description | This function is used to authenticate ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte keytype | Secret key type | Type of authentication key.keytype = 0; indicating authentication with KeyA；keytype = 1; indicating autrhentication with KeyB |
| byte addr | Block address | Block address of the authentication |
| byte [] key | Secret key | authentication key.Allocated memory = 6 byte，e.g.：byte [] key = new byte [6]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Read ISO14443A M1 Block Data

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1ReadBlock (byte addr, byte [] block) |
| Function description | This function is used to read the specified block data of ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | The block address to read data |
| byte [] block | Block data | Block data to be readAllocated memory = 16 byte，e.g.：byte [] block = new byte [16]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Write ISO14443A M1 Block Data

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1WriteBlock (byte addr, byte [] block) |
| Function description | This function is used to write block data to the specified ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | The block address to write data |
| byte [] block | Block data | Block data to writeAllocated memory = 16 byte，e.g.：byte [] block = new byte [16]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instrructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A M1 Initialized Purse Value

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1InitValue (byte addr, int value) |
| Function description | This function is used to intitialized purse value of the specified ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | The block address to initialize purse value |
| int value | Purse value | Initialized purse value |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Read ISO14443A M1 Purse Value

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1ReadValue (byte addr, M1ReadValue value) |
| Function description | This function is used to read the purse value of the specified ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | The block address to read purse value |
| M1ReadValue value | Purse value to be read | Tag informationM1ReadValue value = new M1ReadValue value ();value.value：purse value |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A M1 Added Purse Value

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1AddValue (byte addr, int value) |
| Function description | This function is used to add the purse value of the specified ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | The block address to add purse value |
| int value | Purse value | Purse value to be added |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A M1 Deducted Purse Value

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1MinValue (byte addr, int value) |
| Function description | This function is used to deduct the purse value of the specified ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | The block address to deduct purse value |
| int value | Purse value | The purse value to be deducted |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A M1 Backup Purse Value

|  |  |
| --- | --- |
| Function prototype | int iso14443AM1BackupValue (byte addr, byte addrbak) |
| Function description | This function is used to backup the purse value of the specified ISO14443A M1 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Original address | Original purse block address to backup |
| byte addrbak | New address | New purse block address |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A M0’s Reading Data Page

|  |  |
| --- | --- |
| Function prototype | int iso14443AM0ReadBlock (byte addr, byte [] block) |
| Function description | This function is used to read the data page of the specified ISO14443A M0 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | Block address to read data page |
| byte [] block | Block data | Data to be readAllocated memory = 4 byte，e.g.：byte [] block = new byte [4]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A M0’s Writing Data Page

|  |  |
| --- | --- |
| Function prototype | int iso14443AM0WriteBlock (byte addr, byte [] block) |
| Function description | This function is used to write data page to the specified ISO14443A M0 tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Block address | Block address to write data page |
| byte [] block | Block data | Data to be writtenAllocated memory = 4 byte，e.g.：byte [] block = new byte [4]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A CPU’s Sending rats Command

|  |  |
| --- | --- |
| Function prototype | int iso14443ARats (byte [] rats) |
| Function description | This function is used to send rats command to the specified ISO14443A CPU tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] rats | Rats data | CPU card rats dataAllocated memory >= 255 byte，e.g.：byte [] rats = new byte [255]. |

 |
| Return value | Return type | int |
| Return value | >= 0：rats data length (succeed in execution)< 0：failure in execution |
| Instructions | Send rats command first before operating CPU card |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A CPU’s Sending APDU Command

|  |  |
| --- | --- |
| Function prototype | int iso14443AAPDU (byte [] cmd, int cmdlen, byte [] rsp) |
| Function description | This function is used to send APDU command to the specified ISO14443A CPU tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] cmd | APDU command | APDU command data |
| int cmdlen | Length  | Effective length of APDU command data |
| byte [] rsp | APDU response | APDU response dataAllocated memory >= 1024 byte，e.g.：byte [] rsp = new byte [1024]. |

 |
| Return value | Return type | int |
| Return value | >= 0：Length of APDU response data (succeed in execution)< 0：failure in execution |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A Tag Suspend

|  |  |
| --- | --- |
| Function prototype | int iso14443AHalt () |
| Function description | This function is used to suspend ISO14443A tag（Halt） |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443A Tag’s Deselect

|  |  |
| --- | --- |
| Function prototype | int iso14443ASDsel () |
| Function description | This function is used to deselect ISO14443A tag |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Transparent Transmission of ISO14443A Tag’s User Command Frame

|  |  |
| --- | --- |
| Function prototype | int iso14443ADtu (byte [] cmd, int cmdlen, byte [] rsp) |
| Function description | This function is used to control the transparent transmission of ISO14443A tag’s user command frame |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] cmd | Request frame | Request frame data |
| int cmdlen | Length | Effective length of request frame data  |
| byte [] rsp | Response frame | Response frame dataAllocated memory >= 1024 byte，e.g.：byte [] rsp = new byte [1024]. |

 |
| Return value | Return type | int |
| Return value | >= 0：Length of reponse frame data (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | This port is temporarily reserved |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443B Tag Query

|  |  |
| --- | --- |
| Function prototype | int iso14443BCheckCard (byte mode, byte [] pupi, byte [] appfield, byte [] protocol) |
| Function description | This function is used to query & activate ISO14443B tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte mode | RFU | ISO14443B tag RFUe.g.：mode = 0x00 |
| byte [] pupi | PUPI | Allocated memory = 4 byte，e.g.：byte [] pupi = new byte [4]. |
| byte [] appfield | App information | Allocated memory = 4 byte，e.g.：byte [] appfield = new byte [4]. |
| byte [] protocol | Protocol information | Allocated memory = 4 byte，e.g.：byte [] protocol = new byte [4]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution |
| Instructions | To control ISO14443B tag, it is necessary for the App to succeed first in explicitly using iso14443BCheckCard function before using ISO14443B tag’s other interface functions |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443B Tag’s Sending APDU Command

|  |  |
| --- | --- |
| Function prototype | int iso14443BAPDU (byte [] cmd, int cmdlen, byte [] rsp) |
| Function description | This function is used to send APDU command to the specified ISO14443B CPU tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] cmd | APDU command | APDU command data |
| int cmdlen | Length | Effective length of APDU command data |
| byte [] rsp | APDU response | APDU response dataAllocated memory >= 1024 byte，e.g.：byte [] rsp = new byte [1024]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of APDU response data (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Transparent Transmission of ISO14443B Tag’s User Command Frame

|  |  |
| --- | --- |
| Function prototype | int iso14443BDtu (byte [] cmd, int cmdlen, byte [] rsp) |
| Function description | This function is used for the transparent transmission of ISO14443B tag’s user command frame |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] cmd | Request frame | Request frame data |
| int cmdlen | Length | Effective length of request frame data |
| byte [] rsp | Response frame | Response frame dataAllocated memory >= 1024 byte，e.g.：byte [] rsp = new byte [1024]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of response frame data (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## ISO14443B Tag Suspend

|  |  |
| --- | --- |
| Function prototype | int iso14443BHalt () |
| Function description | This function is used to suspend ISO14443B tag |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Felica Tag Query

|  |  |
| --- | --- |
| Function prototype | int felicaCheckCard (byte mode, byte [] uid) |
| Function description | This function is used to query & activate felica tag  |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte mode | RFU | Felica tag RFUe.g.：mode = 0x00 |
| byte [] uid | Tag UID | Allocated memory = 16 byte，e.g.：byte [] uid = new byte [16]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | To control felica tag, it is necessary for the App to succeed first in explicitly using felicaCheckCard function before using felica tag’s other interface functions |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Transparent Transmission of Felica Tag’s User Command Frame

|  |  |
| --- | --- |
| Function prototype | int felicaDtu (byte [] cmd, int cmdlen, byte [] rsp) |
| Function description | This function is used to control the transparent transmission of user command frame |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] cmd | Request frame | Request frame data |
| int cmdlen | Length | Effective length of request frame data |
| byte [] rsp | Response frame | Response frame dataAllocated memory >= 1024 byte，e.g.：byte [] rsp = new byte [1024]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of response frame data (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Detect Card Slot

|  |  |
| --- | --- |
| Function prototype | int cardSlotCheck (byte index) |
| Function description | This function is used to detect whether there is a card in the card slot. |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte index | Index | Card slot location indexindex = 0x00 indicates the communication with secondary card holderindex = 0x02 indicates the communication with SAM1index = 0x03 indicates the communication with SAM2index = 0x04 indicates the communication with SAM3index = 0x05 indicates the communication with SAM4 |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution (with card)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | It is necessary to succeed first in explicitly using cardSlotCheck function before using contact-type IC card’s other interface functions. |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Control Card Slot

|  |  |
| --- | --- |
| Function prototype | int cardSlotControl (byte index, byte state, byte baud, byte [] atr) |
| Function description | This function is used to control the power on, power off or reset of the card slot |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte index | Index | Card slot location indexindex = 0x00 indicates the communication with secondary card holderindex = 0x02 indicates the communication with SAM1index = 0x03 indicates the communication with SAM2index = 0x04 indicates the communication with SAM3index = 0x05 indicates the communication with SAM4 |
| byte state | Control | Card slot controlstate = 0x00 indicates turning off the power supply；state = 0x01 indicates turning on the power supply；state = 0x02 indicates reset. |
| byte baud | Baud rate | Communication Baud rate of the card slotbaud = 0x00 indicates 9600baud = 0x01 indicates 19200baud = 0x02 indicates 38400baud = 0x03 indicates 57600baud = 0x04 indicates 115200 |
| byte [] atr | ATR | ATR data.Allocated memory >= 255 byte，e.g.：byte [] atr = new byte [255].Turn off the power supply when state = 0x00, without ATR data |

 |
| Return value | Return type | int |
| Return value | >= 0：length of ATR data (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Send APDU Command to ISO7816

|  |  |
| --- | --- |
| Function prototype | int iso7816APDU (byte index, byte [] cmd, int cmdlen, byte [] rsp) |
| Function description | This function is used to send APDU command to the specified ISO7816 CPU tag |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte index | Index | Card slot location indexindex = 0x00 indicates the communication with secondary card holderindex = 0x02 indicates the communication with SAM1index = 0x03 indicates the communication with SAM2index = 0x04 indicates the communication with SAM3index = 0x05 indicates the communication with SAM4 |
| byte [] cmd | APDU command | APDU command data |
| int cmdlen | Length | Effective length of APDU command data |
| byte [] rsp | APDU response | APDU response dataAllocated memory >= 1024 byte，e.g.：byte [] rsp = new byte [1024]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of APDU response data (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## Read Magnetic Strip Card Data

|  |  |
| --- | --- |
| Function prototype | int magneticStripCardReadData (byte second, TrackData trackdata) |
| Function description | This function is used to read the magnetic track data of the magnetic strip card |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte second | Time | Overtime of reading the magnetic strip card dataUnit: second |
| TrackData trackdata | Magnetic track data | Magnetic track data that has been readAllocated memory as follows:TrackData trackdata = new TrackData ();trackdata.trackdata1 = new byte[79];trackdata.trackdata2 = new byte[40];trackdata.trackdata3 = new byte[107];trackdata.len1：length of magnetic track 1 datatrackdata.trackdata1：magnetic track 1 datatrackdata.len2：length of magnetic track 2 datatrackdata.trackdata2：magnetic track 2 datatrackdata.len3：length of magnetic track 3 datatrackdata.trackdata3：magnetic track 3 data |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4418/SLE4428/SLE5518/SLE5528 Card’s Select

|  |  |
| --- | --- |
| Function prototype | int sle4418TO5528Select () |
| Function description | This function is used to select SLE4418/SLE4428/SLE5518/SLE5528 card |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4418/SLE4428/SLE5518/SLE5528 Card’s Reading Data

|  |  |
| --- | --- |
| Function prototype | int sle4418TO5528ReadData (int addr, int len, byte [] data) |
| Function description | This function is used to read the data of SLE4418/SLE4428/SLE5518/SLE5528 card  |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to read data |
| int len | Length | Length of the data to be read |
| byte [] data | Data | Data to be readAllocated memory >= len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4418/SLE4428/SLE5518/SLE5528 Card’s Writing Data

|  |  |
| --- | --- |
| Function prototype | int sle4418TO5528WriteData (int addr, int len, byte [] data) |
| Function description | This function is used to write the data of SLE4418/SLE4428/SLE5518/SLE5528 card |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to write data |
| int len | Length | Length of the data to be written |
| byte [] data | Data | Data to be writtenAllocated memory >= len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4428/SLE5528 Card’s Count of Reading Remaining Errors

|  |  |
| --- | --- |
| Function prototype | int sle4428And5528ReadErrCnt () |
| Function description | This function is used to count the reading of SLE4428/SLE5528 card’s remaining errors |
| Parameter description | None |
| Return value | Return type | int |
| Return value | >= 0：authentication of the count of remaining errors (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4418/SLE4428/SLE5518/SLE5528 Card’s Reading Protected Area Data

|  |  |
| --- | --- |
| Function prototype | int sle4418TO5528ReadProtectData (int addr, int len, byte [] protect) |
| Function description | This function is used to read SLE4418/SLE4428/SLE5518/SLE5528 card’s protected area data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to read protected area data |
| int len | Length | Length of the protected area data to be read |
| byte [] protect | Data | Protected area data to be readAllocated memory >= len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the protected area data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4418/SLE4428/SLE5518/SLE5528 Card’s Writing Protected Area Data

|  |  |
| --- | --- |
| Function prototype | int sle4418TO5528WriteProtectData (int addr, int len, byte [] protect) |
| Function description | This function is used to write SLE4418/SLE4428/SLE5518/SLE5528 card’s protected arera data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to write protected area data |
| int len | Length | Length of the protected area data to be written |
| byte [] protect | Data | Protected area data to be writtenAllocated memory >= len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4428/SLE5528 Card’s Reading Secret Key

|  |  |
| --- | --- |
| Function prototype | int sle4428And5528ReadKey (byte [] key) |
| Function description | This function is used to read SLE4428/SLE5528 card’s secret key |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] key | Secret key | Secret key value to be readAllocated memory = 2 byte，e.g.：byte [] key = new byte [2]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4428/SLE5528 Card’s Writing Secret Key

|  |  |
| --- | --- |
| Function prototype | int sle4428And5528WriteKey (byte [] key) |
| Function description | This function is used to write SLE4428/SLE5528 card’s secret key |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] key | Secret key | Secret key value to be writtenAllocated memory = 2 byte，e.g.：byte [] key = new byte [2]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4428/SLE5528 Card’s Authentication Key

|  |  |
| --- | --- |
| Function prototype | int sle4428And5528Auth (byte [] key) |
| Function description | This function is used to authenticate SLE4428/SLE5528 card |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] key | Secret key | Secret key valueAllocated memory = 2 byte，e.g.：byte [] key = new byte [2]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4432/SLE4442/SLE5532/SLE5542 Card’s Select

|  |  |
| --- | --- |
| Function prototype | int sle4432TO5542Select () |
| Function description | This function is used to select SLE4432/SLE4442/SLE5532/SLE5542 card |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4432/SLE4442/SLE5532/SLE5542 Card’s Reading Data

|  |  |
| --- | --- |
| Function prototype | int sle4432TO5542ReadData (byte addr, byte len, byte [] data) |
| Function description | This function is used to read SLE4432/SLE4442/SLE5532/SLE5542 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Address | Starting address to read data |
| byte len | Length | Length of the data to be read |
| byte [] data | Data | Data to be readAllocated memory >= len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4432/SLE4442/SLE5532/SLE5542 Card’s Writing Data

|  |  |
| --- | --- |
| Function prototype | int sle4432TO5542WriteData (byte addr, byte len, byte [] data) |
| Function description | This function is used to write SLE4432/SLE4442/SLE5532/SLE5542 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Address | Starting address to write data |
| byte len | Length | Length of the data to be written |
| byte [] data | Data | Data to be writtenAllocated memory >= len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4442/SLE5542 Card’s Count of Reading Remaining Errors

|  |  |
| --- | --- |
| Function prototype | int sle4442And5542ReadErrCnt () |
| Function description | This function is used to count the reading of SLE4442/SLE5542 card’s remaining errors |
| Parameter description | None |
| Rreturn value | Return type | int |
| Rreturn value | >= 0：authenticate the count of remaining errors (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4432/SLE4442/SLE5532/SLE5542 Card’s Reading Protected Area Data

|  |  |
| --- | --- |
| Function prototype | int sle4432TO5542ReadProtectData (byte addr, byte len, byte [] protect) |
| Function description | This function is used to read SLE4432/SLE4442/SLE5532/SLE5542 card’s protected area data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Address | Starting address to read protected area data |
| byte len | Length | Length of the protected area data to be read |
| byte [] protect | Protected area data | Protected area data to be readAllocated memory >= len byte，e.g.：byte [] protect = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the protected area data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4432/SLE4442/SLE5532/SLE5542 Card’s Writing Protected Area Data

|  |  |
| --- | --- |
| Function prototype | int sle4432TO5542WriteProtectData (byte addr, byte len, byte [] protect) |
| Function description | This function is used to write SLE4432/SLE4442/SLE5532/SLE5542 card’s protected area data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Address | Starting address to write protected area data |
| byte len | Length | Length of the protected area data to be written |
| byte [] protect | Data | Protected area data to be writtenAllocated memory >= len byte，e.g.：byte [] protect = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4442/SLE5542 Card’s Reading Secret Key

|  |  |
| --- | --- |
| Function prototype | int sle4442And5542ReadKey (byte [] key) |
| Function description | This function is used to read SLE4442/SLE5542 card’s secret key |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] key | Secret key | The secret key value to be readAllocated memory = 3 byte，e.g.：byte [] key = new byte [3]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4442/SLE5542 Card’s Writing Secret Key

|  |  |
| --- | --- |
| Function prototype | int sle4442And5542WriteKey (byte [] key) |
| Function description | This function is used to write SLE4442/SLE5542 card’s secret key |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] key | Secret key | Secret key value to be writtenAllocated memory = 3 byte，e.g.：byte [] key = new byte [3]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## SLE4442/SLE5542 Card’s Authentication Key

|  |  |
| --- | --- |
| Function prototype | int sle4442And5542Auth (byte [] key) |
| Function description | This function is used to authenticate SLE4442/SLE5542 card |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte [] key | Secret key | Secret key value to be authenticatedAllocated memory = 3 byte，e.g.：byte [] key = new byte [3]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 Card’s Select

|  |  |
| --- | --- |
| Function prototype | int at24C01to16Select () |
| Function description | This function is used to select AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 card |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 Card’s Reading Data

|  |  |
| --- | --- |
| Function prototype | int at24C01to16ReadData (int addr, int len, byte [] data) |
| Function description | This function is used to read AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 card’s data  |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to read data |
| int len | Length | Length of the data to be read |
| byte [] data | Data | Data to be readAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 Card’s Writing Data

|  |  |
| --- | --- |
| Function prototype | int at24C01to16WriteData (int addr, int len, byte [] data) |
| Function description | This function is used to write AT24C01/AT24C02/AT24C04/AT24C08/AT24C16 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to write data |
| int len | Length | Length of the data to be written |
| byte [] data | Data | Data to be writtenAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 Card’s Select

|  |  |
| --- | --- |
| Function prototype | int at24C32to1024Select () |
| Function description | This function is used to select AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 card |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 Card’s Reading Data

|  |  |
| --- | --- |
| Function prototype | int at24C32to1024ReadData (int addr, int len, byte [] data) |
| Function description | This function is used to read AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to read data |
| int len | Length | Length of the data to be read |
| byte [] data | Data | Data to be readAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 Card’s Writing Data

|  |  |
| --- | --- |
| Function prototype | int at24C32to1024WriteData (int addr, int len, byte [] data) |
| Function description | This function is used to write AT24C32/AT24C64/AT24C128/AT24C256/AT24C512/AT24C1024 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to write data |
| int len | Length | Length of the data to be written |
| byte [] data | Data | Data to be writtenAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC102 Card’s Select

|  |  |
| --- | --- |
| Function prototype | int at88SC102Select () |
| Function description | This function is used to select AT88SC102 card |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC102 Card’s Reading Data

|  |  |
| --- | --- |
| Function prototype | int at88SC102ReadData (byte addr, byte len, byte [] data) |
| Function description | This function is used to read AT88SC102 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Address | Starting address to read data |
| byte len | Length | Length of the data to be read |
| byte [] data | Data | Data to be readAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC102 Card’s Writing Data

|  |  |
| --- | --- |
| Function prototype | int at88SC102WriteData (byte addr, byte len, byte [] data) |
| Function description | This function is used to write AT88SC102 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Address | Starting address to write data |
| byte len | Length | Length of the data to be written |
| byte [] data | Data | Data to be writtenAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC102 Card’s Authentication Key

|  |  |
| --- | --- |
| Function prototype | int at88SC102Auth (byte addr, byte len, byte [] key) |
| Function description | This function is used to authenticate AT88SC102 card |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte addr | Address | Address of the secret key to be authenticated |
| byte len | Length | Length of the secret key to be authenticated  |
| byte [] key | Secret key | Secret key dataAllocated memory >= len byte，e.g.：byte [] key = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC102 Card Fuse Wire

|  |  |
| --- | --- |
| Function prototype | int at88SC102Fuse (byte mode) |
| Function description | This function is used for AT88SC102 card fuse wire |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| byte mode | Fuse wire mode | Fuse wire modemode = 0 indictaes：manufacturer area mode = 1 indictaes：error counter in App area 2mode = 2 indictaes：issuer |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC1604 Card’s Select

|  |  |
| --- | --- |
| Function prototype | int at88SC1604Select () |
| Function description | This function is used to select AT88SC1604 card |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC1604 Card’s Reading Data

|  |  |
| --- | --- |
| Function prototype | int at88SC1604ReadData (int addr, int len, byte [] data) |
| Function description | This function is used to read AT88SC1604 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to read data |
| int len | Length | Length of the data to be read |
| byte [] data | Data | Data to be readAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC1604 Card’s Writing Data

|  |  |
| --- | --- |
| Function prototype | int at88SC1604WriteData (int addr, int len, byte [] data) |
| Function description | This function is used to write AT88SC1604 card’s data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to write data |
| int len | Length | Length of the data to be written |
| byte [] data | Data | Data to be writtenAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC1604 Card’s Authentication Key

|  |  |
| --- | --- |
| Function prototype | int at88SC1604Auth (int addr, int len, byte [] key) |
| Function description | This function is used to authenticate AT88SC1604 card |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Address of the secret key to be authenticated |
| int len | Length | Length of the secret key to be authenticated |
| byte [] key | Secret key | Secret key dataAllocated memory >= len byte，e.g.：byte [] key = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT88SC1604 Card Fuse Wire

|  |  |
| --- | --- |
| Function prototype | int at88SC1604Fuse () |
| Function description | This function is used for AT88SC1604 card fuse wire |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT45DB Card’s Select

|  |  |
| --- | --- |
| Function prototype | int at45DBSelect () |
| Function description | This function is used to select AT45DB card |
| Parameter description | None |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT45DB Card’s Reading Data

|  |  |
| --- | --- |
| Function prototype | int at45DBReadData (int addr, int len, byte [] data) |
| Function description | This function is used to read AT45DB card data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to read data |
| int len | Length | Length of the data to be read |
| byte [] data | Data | Data to be readAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | >= 0：length of the data to be read (succeed in execution)< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

## AT45DB Card’s Writing Data

|  |  |
| --- | --- |
| Function prototype | int at45DBWriteData (int addr, int len, byte [] data) |
| Function description | This function is used to write AT45DB card data |
| Parameter description

|  |  |
| --- | --- |
| Parameter type & parameters | Detailed description |
| int addr | Address | Starting address to write data |
| int len | Length | Length of the data to be written |
| byte [] data | Data | Data to be writtenAllocated memory = len byte，e.g.：byte [] data = new byte[len]. |

 |
| Return value | Return type | int |
| Return value | 0：succeed in execution< 0：failure in execution-7：indicating that the reader-writer doesn’t support this command |
| Instructions | None |
| Included in | libSUNMIANDROIDUSBHIDREADER.so |

# Error Code

|  |  |
| --- | --- |
| **Code** | **Definition** |
| 0 | Succeed in the execution of the function |
| -1 | The reader-writer is not opened, jint layer’s data conversion error, wrongful pass of parameter to java layer or length error of the paramter’s memory allocation passed to java layer |
| -2 | Reader-wirter response overtime, validation error or length error of the data frame in reader-writer response |
| -3 | The reader-writer response data do not match the reader-writer command data |
| -4 | Reader-wirter response code error |
| -5 | The format of the reader-writer response data does not mztch thr response data format specified by the reader-writer command protocol |
| -6 | The length of the reader-writer response data is beyond the border of data cache |
| -7 | Mode or command not supported by the reader-writer |

Document Control Page

|  |
| --- |
| **Historic Records of the Document** |
| Date | Name | Update Records of the Version |
| 2017-04-16 | Jinxin Zhang | V1.0.0Created. |
| 2017-05-16 | Yunting Xu | V1.0.1Update the style |
| 2017-06-26 | Jinxin Zhang | V1.0.21．Add development configuration of SDK document2．Add the performance parameters of the reader-wirter |
| 2017-09-21 | Jinxin Zhang | V2.0.0 |