

# Lib160 API Guide

Version 1.2

2014-05-07

## Revision History

Revision	Date	Description	By
----------	------	-------------	----

---



# Directory

Foreword.....	3
1.HAL api spec.....	3
1.1 Function List.....	3
1.1.0 init device.....	3
1.1.1 close device.....	3
1.1.2 Get the serial number.....	3
1.1.3 Set work mode.....	4
1.1.4 Download 3DES key.....	4
1.2 IC card function.....	4
1.2.1 Init IC card.....	4
1.2.2 close slot.....	4
1.2.3 IC card Write/Read.....	5
1.3 RFID card function.....	6
1.3.1 Find the card and return the card's serial number.....	6
1.3.2 Select RFID card.....	6
1.3.3 RFID card authorization.....	6
1.3.4 Read block data.....	6
1.3.5 write block data.....	7
1.3.6 power off rfid.....	7
1.3.7 read sector data.....	7
1.3.8 write sector data.....	8
1.4 Magnetic Stripe card.....	8
1.4.1 power on.....	8
1.4.2 power off.....	8
1.4.3Reset head.....	9
1.4.4 Detect whether swipe the card.....	9
1.5 Data Structures.....	10
1.5.1 IC Card data struct.....	10
Note.....	10

## Foreword

160Lib support PSAM cards, IC cards, contactless IC cards, magnetic stripe cards, providing WINDOWS dynamic library (Lib160.dll, Lib160.lib) when the user development, users should target machine dynamic library files into the appropriate directory.

## 1. HAL api spec

### 1.1 Function List

#### 1.1.0 init device

<b>Type</b>	unsigned char __stdcall InitDev (unsigned char comport,long BaudRate)
<b>Description</b>	Set the serial port number, baud rate
<b>Example</b>	ucRet=InitDev(COM1,115200)
<b>Response</b>	0x00-----init ok. 0xff-----init error
<b>Note</b>	USB interface or serial interface to share this function to open the device. When the device is a USB interface, comport need to assign more than 200 number, BaudRate any value. When the device for the serial interface, the actual port and baud rate settings. (Please refer Demo)

#### 1.1.1 close device

<b>Type</b>	void DelDev ()
<b>Description</b>	close com

#### 1.1.2 Get the serial number

<b>Type</b>	void ReadSN(unsigned char *SN)
<b>Description</b>	Get the serial number
<b>Parameters</b>	SN-data buffer

### 1.1.3 Set work mode

<b>Type</b>	int _stdcall setWorkMode(int mode)
<b>Description</b>	Switching device work mode (encryption devices only support)
<b>Parameters</b>	mode-5: plain text (not encrypted), 6:3DS encrypted communication
<b>Response</b>	0x00-----set success. 0xff-----set failure

### 1.1.4 Download 3DES key

<b>Type</b>	unsigned char _stdcall set3DESKey(unsigned char* oldKey,unsigned char* newKey)
<b>Description</b>	Update 3DES key (encryption devices only support)
<b>Parameters</b>	oldKey-old 3des key(default is 16 bytes 0x00), newKey:new 3des key
<b>Response</b>	0x00-----set success. 0xff-----set failure

## 1.2 IC card function

### 1.2.1 Init IC card

<b>Type</b>	unsigned char IccInit(unsigned char slot, unsigned char *ATR)
<b>Description</b>	Init and reset ic card
<b>Parameters</b>	Slot-0~5 ATR – Answer To Reset Result. (need min 32+1bytes buffer) ATR[0] ( ATR length. ATR[1]~ATR[ATR[0]](IC Reset result.
<b>Response</b>	0x00-----init ok. 0x01-----Card out 0xf0-----slot error 0x06-----Communication failure

### 1.2.2 close slot

<b>Type</b>	void IccClose (unsigned char slot)
<b>Description</b>	close slot and power off ic card
<b>Parameters</b>	slot - 0-4
<b>Response</b>	0x00-----init ok. 0x01-----Card out 0xf0-----slot error 0x06-----Communication failure

### 1.2.3 IC card Write/Read

<b>Type</b>	unsigned char IccIsoCommand(unsigned char slot,APDU_SEND *ApduSend,APDU_RESP *ApduRecv)
<b>Description</b>	IC card operation function This function supports IC cartoon with interface protocol (T = 0 and T = 1)
<b>Parameters</b>	<p>slot-0~5</p> <p>APDU_SEND struct:</p> <pre>struct{ unsigned char    Command[4]; unsigned int     Lc; unsigned char    DataIn[512]; unsigned int     Le; }; Command[] = {CLA, INS, P1, P2}。 Lc = DataIn length。 DataIn = To send a pointer to the data to the IC card。 Le =Length of data expected to return。 Case1: Lc=0; Le=0 Neither data nor transmit data back Case2: Lc=0; Le&gt;0 However, no data is sent to return the desired data, if in practice the number of data expected to return the terminal is unknown, set Le = 256; otherwise determined values. Case3: Lc&gt;0; Le=0 Data is sent without the desired data returns Case4: Lc&gt;0; Le&gt;0 A transmission data and has the desired data returns, if in practice the number of data expected to return the terminal is unknown, set Le = 256; otherwise determined values. APDU_RESP struct: struct{ unsigned int LenOut; unsigned char DataOut[512]; unsigned char    SWA; unsigned char    SWB; }; LenOut = The actual data returned from the IC card length。 DataOut = The data returned from the IC card pointer。 SWA = Status byte 1. SWB = Status byte 2.</pre>
<b>Response</b>	<p>0x00-----Successful implementation;</p> <p>0xff-----Can not communicate with or without power.</p>

## 1.3 RFID card function

### 1.3.1 Find the card and return the card's serial number

<b>Type</b>	unsigned char M1Request(unsigned char type,unsigned char *rsp)
<b>Description</b>	Find the card and return the card's serial number
<b>Parameters</b>	Type-0x0A Type A, 0x0B TYPEB Rsp-at least 6 bytes
<b>Response</b>	0x00-----ok

### 1.3.2 Select RFID card

<b>Type</b>	unsigned char M1Select(unsigned char *SerialNo)
<b>Description</b>	serialNo, 4 bytes
<b>Parameters</b>	SerialNo-serial number of card
<b>Response</b>	0x00-----ok

### 1.3.3 RFID card authorization

<b>Type</b>	unsigned char M1Authority(unsigned char type,unsigned char block ,unsigned char *pwd)
<b>Parameters</b>	type-password type 0x0A pass A, 0x0B pass B Block- block number Pwd- Pointing to an array of storage card password (6 char array password)
<b>Response</b>	0x00-----ok

### 1.3.4 Read block data

<b>Type</b>	unsigned char M1ReadBlock(unsigned char block,unsigned char *pck)
<b>Description</b>	read out one block data of card,16 bytes
<b>Parameters</b>	Block-The absolute block number IC card, IC card when you need to read the x-y area's first block, the block number must be an absolute block = x * 4 + y. Pck-Subscript number greater than 16 points of the array, as the return of the card 16 bytes of data cache.
<b>Response</b>	0x00-----ok

### 1.3.5 write block data

<b>Type</b>	unsigned char M1WriteBlock(unsigned char block,unsigned char *pck)
<b>Description</b>	write data to one block of rfid card,16bytes
<b>Parameters</b>	Block- The absolute block number IC card, IC card when you need to write the first x y block area first, the absolute block number must be block = x * 4 + y. Pck-Subscript number greater than 16 points of the array, as the return of the card 16 bytes of data cache
<b>Response</b>	0x00-----ok

### 1.3.6 power off rfid

<b>Type</b>	unsigned char M500PiccHalt(void)
<b>Description</b>	Let rfid card dormant
<b>Response</b>	0x00-----ok

### 1.3.7 read sector data

<b>Type</b>	unsigned char M1ReadSec(unsigned char cardtype,unsigned char *pwd,unsigned char keyAB,unsigned char sector,unsigned char *buf,unsigned char mode ,unsigned char *snr,unsigned char timeout)
<b>Description</b>	read whole one sector data
<b>Parameters</b>	Cardtype-card type, 0x0A type A card, 0x31 type B card Pwd:-Pointing to an array of storage card password (6 char array password) keyAB-0x0A pass A, 0x0B pass B Sector-sector number Buf-read buffer, >=42 bytes Mode-Reserved Snr-Card serial number is returned timeout-Timeout
<b>Response</b>	0x08-Look for card error, there is no card in the induction area。 0x10-The card may have been dormant, not selected, but the card has been read out the serial number 0x12-Password authentication failed 0x01-0~2Block did not read out, swipe too fast 0x00-Operation is successful, the read data valid 0xff-Unknown error

### 1.3.8 write sector data

<b>Type</b>	unsigned char M1WriteSec(unsigned char cardtype,unsigned char *pwd,unsigned char keyAB,unsigned char sector,unsigned char *buf,unsigned char len,unsigned char mode ,unsigned char *snr,unsigned char timeout)
<b>Description</b>	write data to whole one sector
<b>Parameters</b>	Parameters-cardtype: type, 0x0A A card, 0x31 B card Pwd- Pointing to an array of storage card password (6 char array password) keyAB-0x0A: passA, 0x0B passB Sector-sector number Buf-write buffer Len-data length Mode-Reserved Snr- return serial number Timeout- timeout
<b>Response</b>	0x08-Look for card error, there is no card in the induction area。 0x10-The card may have been dormant, not selected, but the card has been read out the serial number 0x12-Password authentication failed 0x01-0~2Block did not read out, swipe too fast 0x00-Operation is successful, the read data valid 0xff-Unknown error

## 1.4 Magnetic Stripe card

### 1.4.1 power on

<b>Type</b>	void MagOpen(void)
<b>Description</b>	open magnetic. Read magnetic data using interrupt work, once open magnetic card reader, even without calling Read function, as long as the credit card, the same can read magnetic head data, so no need to use magnetic card reader, magnetic card reader is best to turn off

### 1.4.2 power off

<b>Type</b>	void MagClose(void)
<b>Description</b>	close magnetic

### 1.4.3 Reset head

<b>Type</b>	void MagReset(void)
<b>Description</b>	Reset heads, and clears the buffer data card. The head has been on the case of electricity, the function resets the head, remove card data buffer;No power at the head of the case, only clears the buffer data card.To ensure that the data read head is the latest data, the cycle of test card, it is best to call this function once to clear the buffer data card.

### 1.4.4 Detect whether swipe the card

<b>Type</b>	unsigned char MagSwiped(void)
<b>Description</b>	Detect whether swipe the card Regardless of whether the credit card, the function will return immediately.
<b>Response</b>	0 -yes 0xff -no

### 1.4.5 read card data

<b>Type</b>	unsigned char MagRead(unsigned char *Track1, unsigned char *Track2,unsigned char *Track3)
<b>Description</b>	1,2,3 track magnetic read data buffer
<b>Parameters</b>	Track1 - Store a pointer to the data track 1 Track2 - Store a pointer to the data track 2 Track3 - Store a pointer to the data track 3
<b>Response</b>	0x00 read Card Error bit0 = 1 Correctly read track 1 data bit1 = 1 Correctly read track 2 data bit2 = 1 Correctly read track3 data bit4 = 1 1Track data with parity error bit5 = 1 2Track data with parity error bit6 = 1 3Track data with parity error

## 1.5 MEMORY card operate function

### 1.5.1 MEMORY card read IO status

<b>Type</b>	unsigned char Mc_Io_Read(unsigned char slot)
<b>Description</b>	Selected slot, card read for MEMORY
<b>Parameters</b>	Slot:Need to set up a cartoon channel numbers
<b>Response</b>	0:low 1:high

### 1.5.2 MEMORY card write IO

<b>Type</b>	void Mc_Io_Write(unsigned char slot,unsigned char mode)
<b>Description</b>	MEMORY card operations IO port, set the high and low
<b>Parameters</b>	Slot:Need to set up a cartoon channel numbers Mode:1respect high,0 respect low
<b>Response</b>	no

## Note

The library suitable for our USB interface card, proximity card, IC card reader and other devices, but it does not represent all devices support all the features described in this document. Which part of the function of specific needs, only the reference interface documentation to describe the functional part. If in doubt, and want to get the latest version of the document, please contact us.