Lib160 API Guide

Version 1.6

2019-04-12

Revision History

Revision	Date	Description	By
1.0	03/11/2012	Initial Release,including IC ,magnetic,RFID cards interface description	zhengw
1.1	05/12/2013	Modify IC card section api describes	zhengw
1.2	05/04/2014	Integrated Serial Interface,add the encryption interface, modify magnetic cards, proximity cards section interface description	zhengw
1.2	24/04/2014	Increase SLE4442 card interface	zhengw
1.3	24/04/2014	Increase SLE4428 card interface	zhengw
1.4	29/04/2014	After increasing the portion of the interface functions for hardware upgrades, modified proximity card section describes	zhengw
1.4.2	05/05/2014	Modify the non-contact IC card section Interface Description	zhengw
1.4.3	06/05/2014	Integrated serial port function, modify the card section Interface Description	zhengw
1.4.4	03/11/2014	Increase Ultralight card reader function	Feng
1.5	27/03/2019	Increase Desfire card reader function	liangc
1.6	04/04/2019	Increase pinpad function	liangc
1.7	12/04/2019	Increase emv function	liangc

Directory

Foreword....4
1.HAL api spec....4
1.1 Function List....4
1.1.0 init device....4
1.1.1 close device....5
1.1.2 Get the firmware version....5
1.1.3 Set work mode....5
1.1.4 Download 3DES key....5
1.1.5 Get the SDK version....5
1.2 CPU IC card function....6
1.2.1 Init IC card....6
1.2.2 close slot....6
1.2.3 IC card Write/Read....6
1.2.4 Detection card....6
1.3 M1 card function....7
1.3.1 Find the card and return the card's serial number....7
1.3.2 Select RFID card....7
1.3.3 RFID card authorization....7
1.3.4 Read block data....7
1.3.5 write block data....8
1.3.6 power off rfid....8
1.3.7 read sector data....8
1.3.8 write sector data....9
1.4 Magnetic Stripe card....9
1.4.1 power on....9
1.4.2 power off....10
1.4.3Reset head....10
1.4.4 Detect whether swipe the card....10
1.4.5 read card data....10
1.5 EMV Card function....11
1.5.1 Read EMV Card Number....11
1.5.2 EMV Core Init....11
1.5.3 Add AIDList....11
1.5.4 Add Capk....12
1.5.5 EMV Transaction....13
1.5.6 Get TLV data....13
1.5.7 Get TermParam Pointer....14
1.5.8 Get TransParam Pointer....14
1.5.9 Set TLV data....15
1.5.10 Set CallBackFun....15
1.5.11 Clear AIDList....16
1.5.12 Clear CAPKList....16

1.6 Desfire Card function....17
1.6.1 Request Desfire Card....17
1.6.2 Activation Desfire Card....17
1.6.3 Power off Desfire Card....17
1.6.4 Authentication key....17
1.6.5 Update key....18
1.6.6 Get Key Setting....18
1.6.7 Change Key Setting....18
1.6.8 Get Key Version....18
1.6.9 Select Application....19
1.6.10 Create Application....19
1.6.11 Delete Application....19
1.6.12 Get the Application IDentifiers....19
1.6.13 Format Card....20
1.6.14 Get Card Information....20
1.6.15 Get File IDentifiers....20
1.6.16 Get File Setting....20
1.6.17 Change File Setting....21
1.6.18 Create StdDataFile....21
1.6.19 Create BackupDataFile....21
1.6.20 Create ValueFile....21
1.6.21 Create LinearRecordFile....22
1.6.22 Create CyclicRecordFile....22
1.6.23 Delete File....22
1.6.24 Write DataFile....23
1.6.25 Read DataFile....23
1.6.26 Get ValueFile....23
1.6.27 PlusValueFile....23
1.6.28 MinusValueFile....24
1.6.29 Write Record....24
1.6.30 Read Record....24
1.6.31 Clear RecordFile....24
1.6.32 Commit Transaction....25
1.6.33 Abort Transaction....25
1.6.34 Error Codes....25
1.7 PinPad function....26
1.7.1 Set Port number....26
1.7.2 Get Port number....26
1.7.3 PinPadInit....27
1.7.4 Get PinPad Information....27
1.7.5 Load MasterKey....27
1.7.6 Load WorkKey....27
1.7.7 Get PINBlock....28
1.7.8 GetMac....28

1.7.9 CalcDES..... 29
Note..... 29

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

Type	unsigned char __stdcall InitDev (unsigned char comport,long BaudRate)
Description	Set the serial port number, baud rate
Example	ucRet=InitDev(COM1,115200)
Response	0x00----------init ok. 0xff----------init error
Note	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

Type	void DelDev ()
Description	close com

1.1.2 Get the firmware version

Type	void ReadSN(unsigned char *SN)
Description	Get the Firmware version
Parameters	SN-data buffer

1.1.3 Set work mode

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

1.1.4 Download 3DES key

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

1.1.5 Get the SDK version

Type	unsigned char __stdcall GetSDKVersion(char *OutVer)
Description	Get the SDKversion
Parameters	OutVer-sdk version buffer
Response	0x00 read ok Other read fail

1.2 CPU IC card function

1.2.1 Init IC card

Type	unsigned char IccInit(unsigned char slot, unsigned char *ATR)
Description	Init and reset ic card
Parameters	Slot-0~4 or 5(Contactless) ATR – Answer To Reset Result.（need min 64+1bytes buffer） ATR[0] ATR length. ATR[1]~ATR[ATR[0]] IC Reset result.
Response	0x00----------init ok. 0x01----------Card out 0xf0----------slot error 0x06----------Communication failure

1.2.2 close slot

Type	void IccClose (unsigned char slot)
Description	close slot and power off ic card
Parameters	slot – 0-4 or 5(Contactless)
Response	0x00---------- ok. 0x01----------Card out 0xf0----------slot error 0x06----------Communication failure

1.2.3 IC card Write/Read

Type	unsigned char IccIsoCommand2(unsigned char slot, unsigned char *Send ,int sendLen, unsigned char *Recv,int &rcvLen)
Description	IC card operation function This function supports IC cartoon with interface protocol (T = 0 and T = 1)
Parameters	slot-0~4 or 5(Contactless) Send-APDU data to be sent sendLen-APDU data length to be sent Recv-Response data rcvLen-Response data length
Response	0x00---------Successful implementation； 0xff---------Can not communicate with or without power.

1.2.4 Detection card

Type	unsigned char IccDetect(unsigned char slot)

Description	Detection card exists
Parameters	slot – 0-4
Response	0 -yes other -no

1.3 M1 card function

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

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

1.3.2 Select RFID card

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

1.3.3 RFID card authorization

Type	unsigned char M1Authority(unsigned char type,unsigned char block ,unsigned char *pwd)
Parameters	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)
Response	0x00--------ok

1.3.4 Read block data

Type	unsigned char M1ReadBlock(unsigned char block,unsigned char *pck)
Description	read out one block data of card,16 bytes
	Block-The absolute block number IC card, IC card when you need to

Parameters	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.
Response	0x00--------ok

1.3.5 write block data

Type	unsigned char M1WriteBlock(unsigned char block,unsigned char *pck)
Description	write data to one block of rfid card,16bytes
Parameters	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
Response	0x00--------ok

1.3.6 power off rfid

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

1.3.7 read sector data

Type	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)
Description	read whole one sector data
Parameters	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
Response	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

Type	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)
Description	write data to whole one sector
Parameters	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
Response	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

Type	void MagOpen(void)
Description	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

Type	void MagClose(void)
Description	close magnetic

1.4.3Reset head

Type	void MagReset(void)
Description	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

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

1.4.5 read card data

Type	unsigned char __stdcall MagRead_DES(unsigned char *Track1,unsigned char *Track2,unsigned char *Track3)
Description	Read magnetic data(plain text or encryption)
Parameters	Track1 - Store a pointer to the data track 1,the first byte is the data length Track2 - Store a pointer to the data track 2,the first byte is the data length Track3 - Store a pointer to the data track 3,the first byte is the data length
Response	0x00 read Card Error bit0 = 1 Correctly read track 1 data bit1 = 1 Correctly read track 2 data bit2 = 1 Correctly read track3 data

1.5 EMV Card function

1.5.1 Read EMV Card Number

Type	int _stdcall MSR_ICCardID(unsigned char slot,char *data)
Description	Read EMV Card Number
Parameters	Slot- 0 or 5 Data- Card Number data, the first byte is the data length
Response	0x00 read ok Other read fail

1.5.2 EMV Core Init

Type	int _stdcall EmvCoreCbInit(void)
Description	Emv CoreCbInit
Parameters	none
Response	0x00 read ok Other read fail

1.5.3 Add AIDList

Type	int _stdcall EmvAddAIDList(T_EMV_APP_LIST* pEmvAppList)
Description	Add AIDList
Parameters	pEmvAppList - AIDList data;
Response	0x00 read ok Other read fail

note	typedef struct
	{
	u8 ucAID[16]; //9F06
	u8 ucAIDLen; //
	u8 ucSelectIndicator; //DF01
	u8 ucTargetPercentage; //DF17
	u8 ucMaxTargetPercentage; //DF16
	u8 ucTermFloor[4]; //9F1B

	u32 uiThresholdValue; //DF15
	u8 ucOnLinePINFlag; //DF18
	u8 ucTACDefault[5]; //DF11
	u8 ucTACDenial[5]; //DF13
	u8 ucTACOnline[5]; //DF12 联 机ú
	u8 ucDdolLen; // DDOL
	u8 ucDdol[252]; //DF14 DDOL
	u8 ucTdolLen; // TDOL
	u8 ucTdol[252]; //97
	u8 ucTermAVM[2]; //9F09
	u8 ucRFTxnLmt[6]; //DF20
	u8 ucRFFLmt[6]; //DF19
	u8 ucRFCVMLmt[6]; //DF21
	u8 ucECTranLmt[6]; //9F7B
	ó----------
	u8 ucMerchantNameLocation[40+1];//9F4E
	u8 ucMerchantCode[2]; //9F15
	u8 ucMerchantID[15+1]; //9F16
	u8 ucAcquirerID[6]; //9F01
	u8 ucTermID[8+1]; //9F1C
	u8 ucTranRefCurrExp; //9F3D
	u8 ucTranRefCurr[2]; //9F3C
	u8 ucTranCurrExp; //5F36
	u8 ucTranCurrCode[2]; //5F2A
	//----------
	
	//PAYPASS----------
	u8 ucUdolLen; //UDOL
	u8 ucUdol[240]; //9F69
	u8 MagStripeInd; // Mag Stripe
	u8 MagStripeVer[2]; //9F6D
	u8 TermCapNoCVMReq[3]; //NoCVM
	u8 TermCapCVMReq[3]; //CVM
	u8 ucPayPassAddTermCapa[5]; //9F40
	u8 ucPayPassTermType; //9F35
	u8 PaypassRFU[32]; //
	//----------
	u16 unCrc; //Reserved
	}T_EMV_APP_LIST;

1.5.4 Add Capk

Type	int _stdcall EmvADDCAPKList(T_EMV_TERM_CAPK* pEmvTermCapk)
Description	Add Capk
Parameters	pEmvTermCapk- Capk data;
	0x00 read ok

Response	Other read fail

note	typedef struct
	{
	u8 ucRID[5]; //9F06
	u8 ucIndex; //9F22
	u8 ucHashIndicator; //DF06
	u8 ucCAPKIndicator; //DF07
	u8 ucModulusLen;
	u8 ucModulus[248]; //DF02
	u8 ucExponentLen;
	u8 ucExponent[3]; //DF04
	u8 ucExpDate[3]; //DF05 (bcd yyyymmdd)
	u8 ucCheckSum[20]; //DF03
	u16 unCrc; // Reserved
	}T_EMV_TERM_CAPK;

1.5.5 EMV Transaction

Type	int _stdcall EmvCoreTrans(u8 Slot,u8 *pAmountAuth,u8 *pucTransResult)
Description	EMV Transaction
Parameters	Slot- 0(contact) or 5(contactless) pAmountAuth - BCD format (6byte,E.g 10.00 is \x00\x00\x00\x00\x10\x00) pucTransResult- Transaction Result 0x40 :Approve 0x00 :Refuse 0x80 :Online
Response	0x00 read ok Other read fail

1.5.6 Get TLV data

Type	int _stdcall EmvCoreGetTagData(u32 uiTag,u8 *pucDataOut,u32 *puiDataOutLen)
Description	GetTagData
Parameters	uiTag - Tag pucDataOut - Out Tag value puiDataOutLen- Out Tag value len
Response	0x00 read ok Other read fail

1.5.7 Get TermParam Pointer

Type	T_EMV_TERM_PARAM *EmvCoreGetTermParam(void)
Description	Get TermParam Pointer，Let the upper application get the parameter pointer so that it can be set
Parameters	none
Response	TermParam variable pointer

note	typedef struct
	{
	u8 ucIFD[8+1]; //9F1E
	u8 ucTerminalCountry[2]; //9F1A
	u8 ucTermType; //9F35
	u8 ucTermCapa[3]; //9F33
	u8 ucAddTermCapa[5]; //9F40
	u8 ucMerchantNameLocation[40+1];//9F4E
	u8 ucMerchantCode[2]; //9F15
	u8 ucMerchantID[15+1]; //9F16
	u8 ucAcquirerID[6]; //9F01 BCD
	u8 ucTermID[8+1]; //9F1C ASC
	u8 ucTranRefCurrExp; //9F3D
	u8 ucTranRefCurr[2]; //9F3C
	u8 ucTranCurrExp; //5F36
	u8 ucTranCurrCode[2]; //5F2A
	u8 ucCapture; //
	u8 ucTermSMSupportIndicator; //
	u8 ucPaypassImplementationOptions; //
	u8 ucTermFLmtFlg; //
	u8 ucRFTxnLmtFlg; //
	u8 ucRFFLmtFlg; //
	u8 ucRFCVMLmtFlg; //
	u8 ucRFStatusCheckFlg; //
	u8 ucRFZeroAmtNoAllowed; //
	u8 ucUseFangba; //
	u8 ucPrintfDebugInfo; //
	u8 ucHostType; //
	u8 ucEmvTest; //
	u8 ucUseCallBackApdu; //
	}T_EMV_TERM_PARAM;

1.5.8 Get TransParam Pointer

Type	T_EMV_TRANS_PARAM *EmvCoreGetTransParam(void)
Description	Get TransParam Pointer，Let the upper application get the parameter pointer so that it can be set
	none

Parameters	
Response	TransParam variable pointer

note	typedef struct
	{
	u8 ucTransKernalType; //
	u8 ucIsForceOnline; //
	u8 ucIsSimpleFlow; //
	u8 ucOption; //
	u8 ucEcTerminalSupportIndicator;//
	u8 ucReaderTTQ[4];
	u8 ucTransNo[4]; //9F41
	u8 ucTransDate[3]; //9A BCD YYMMDD
	u8 ucTransTime[3]; //9F21 BCD HHMMS
	u8 ucAmountAuth[6]; //9F02
	u8 ucAmountOther[6]; //9F03
	u8 ucTransType; //9C
	}T_EMV_TRANS_PARAM;

1.5.9 Set TLV data

Type	int _stdcall EmvCoreSetTagData(u32 uiTag,u8 *pucDataIn,u32 uiDataInLen)
Description	SetTagData,
Parameters	uiTag - Tag pucDataIn- Set Tag value uiDataInLen- Set Tag value len
Response	0x00 read ok Other read fail

1.5.10 Set CallBackFun

Type	int _stdcall EmvCoreSetCallBackFun(T_EMVCORE_CALLBACK *ptEmvCallback)
Description	Set CallBackFun
Parameters	ptEmvCallback- Callback function structure to be set
Response	0x00 read ok Other read fail

note	typedef struct
	{
	//Print log callback function

	s32 (*EmvCbDebugPritLogFun)(u8 *phexIn, u32 uiLen);//
	//Display app tag list
	s32 (*EmvCbSelAppFun)(u8 ucIsFirstSelect,s8 *pcAppLabelList[], s32 iAppNum);
	//Display string ucStr
	s32 (*EmvCbShowHintFun)(s8 *pcStr,s32 iClearFlag,s32 iDisplayTime);
	//Enter PIN
	s32 (*EmvCbInputPINFun)(u8 ucPINType,u8 *pucOutPINBlock);
	//Cardholder ID verification
	s32 (*EmvCbCertVerifyFun)(void);
	//Get the cumulative amount of the transaction for a card
	s32 (*EmvCbGetSumLogByPANFun)(u8 *pucPAN,u32 uiPANLen,u32 *puiOutAmount);
	//Online processing of sending and receiving
	s32 (*EmvCbOnlineProcFun)(void);
	//
	s32 (*EmvCbReferProcFun)(void);
	s32 (*EmvCbAdviceProcFun)(void);
	s32 (*EmvCbReSwipeCardFun)(void);
	
	//Get public key collection list based on RID and public key index, find return 0
	s32 (*EmvCbLoadRevocListFun)(u8 *pucRID,u8 ucCAPKIndex,u8 *pucCertSerial);
	//Find blacklist for exception file check, found return 0
	s32 (*EmvCbSearchExceptionListFun)(u8 *pucPAN,u32 uiPANLen,u8 ucPANSeq);
	}T_EMVCORE_CALLBACK;

1.5.11 Clear AIDList

Type	int _stdcall EmvClearAIDList(void)
Description	Clear EMV AIDList
Parameters	none
Response	0x00 read ok Other read fail

1.5.12 Clear CAPKList

Type	int _stdcall EmvClearCAPKList(void)
Description	Clear EMV CAPKList
Parameters	none

Response	0x00 read ok Other read fail

1.6 Desfire Card function

1.6.1 Request Desfire Card

Type	unsigned char _stdcall DFRequest(unsigned char* buffer)
Description	Find the card and return the card's serial number
Parameters	buffer- card's serial number data, the first byte is the data length
Response	0x00 read ok Other read fail

1.6.2 Activation Desfire Card

Type	unsigned char _stdcall DFReset(unsigned char* buffer)
Description	Activation Desfire Card and return the card's ATS
Parameters	buffer- card's ATS data, the first byte is the data length
Response	0x00 read ok Other read fail

1.6.3 Power off Desfire Card

Type	unsigned char _stdcall DFHalt(void)
Description	Power off Desfire Card
Parameters	no
Response	0x00 read ok Other read fail

1.6.4 Authentication key

Type	unsigned char _stdcall DFAuthenKey(int index,unsigned char* Key)
Description	Authentication key

Parameters	Index -Key number (00 ~0d) Key- key value pointer (16byte)
Response	0x00 read ok Other read fail

1.6.5 Update key

Type	unsigned char _stdcall DFUpdateKey(int index,unsigned char* newPass,unsigned char* oldPass)
Description	Authentication key
Parameters	Index -Key number (00 ~0d) newPass- new key value pointer (16byte) oldPass - old key value pointer (16byte)
Response	0x00 read ok Other read fail

1.6.6 Get Key Setting

Type	unsigned char _stdcall DFGetKeySetting(unsigned char* result)
Description	Get the master key configuration settings depending on the currently selected AID
Parameters	result- out keysetting value pointer (2byte)
Response	0x00 read ok Other read fail

1.6.7 Change Key Setting

Type	unsigned char _stdcall DFChangeKeySetting(unsigned char keysetting)
Description	Changes the master key configuration settings depending on the currently selected AID
Parameters	keysetting- setting key value (1byte)
Response	0x00 read ok Other read fail

1.6.8 Get Key Version

Type	unsigned char _stdcall DFGetKeyVersion(int keyNum,unsigned char *OutVer)
Description	Get the current key version of any key stored on the card.

Parameters	keyNum- Key number (00 ~0d) OutVer - out key version pointer (1byte)
Response	0x00 read ok Other read fail

1.6.9 Select Application

Type	unsigned char _stdcall DFSelectAID(unsigned char* aid)
Description	Select one specific application for further access.
Parameters	aid- application identifier pointer (3byte Low byte first)
Response	0x00 read ok Other read fail

1.6.10 Create Application

Type	unsigned char _stdcall DFCreateAID(unsigned char* aid,unsigned char setting,unsigned char keynum)
Description	Create new applications on the card
Parameters	aid- application identifier pointer (3byte Low byte first) Setting - Application Master Key Settings (1byte) Keynum - Number of Keys (1byite 01~0d)
Response	0x00 read ok Other read fail

1.6.11 Delete Application

Type	unsigned char _stdcall DFDeleteApp(unsigned char* aid)
Description	Delete applications on the card
Parameters	aid- application identifier pointer (3byte Low byte first)
Response	0x00 read ok Other read fail

1.6.12 Get the Application IDentifiers

Type	unsigned char _stdcall DFGetApplicationIDs(unsigned char *aidnum,unsigned char* aids)
Description	Get the Application IDentifiers of all active applications on a card.
Parameters	Aidnum - out application identifier number (1byte) aids- out application identifier pointer (Low byte first)

Response	0x00 read ok Other read fail

1.6.13 Format Card

Type	unsigned char _stdcall DFFormatCard(void)
Description	Releases the card user memory
Parameters	No
Response	0x00 read ok Other read fail

1.6.14 Get Card Information

Type	unsigned char _stdcall DFGetInfo(unsigned char* result)
Description	Get manufacturing related data of the card
Parameters	result- out manufacturing related data pointer (28byte)
Response	0x00 read ok Other read fail

1.6.15 Get File IDentifiers

Type	unsigned char _stdcall DFGetFileIDs(unsigned char *filenum,unsigned char*fileIDs)
Description	Get the File IDentifiers of all active files within the currently selected application
Parameters	filenum- out file identifier number (1byte) fileIDs - out fileidentifier pointer
Response	0x00 read ok Other read fail

1.6.16 Get File Setting

Type	unsigned char _stdcall DFGetFileSetting(int index,unsigned char* filesetting,unsigned char *Outlen)
Description	Get information on the properties of a specific file
Parameters	index- file number (1byte) Filesetting - out file information Outlen - out file information len
Response	0x00 read ok Other read fail

1.6.17 Change File Setting

Type	unsigned char _stdcall DFChangeFileSetting(int index,unsigned char* setting)
Description	Changes the access parameters of an existing file
Parameters	index- file number (1byte) setting- 3 bytes, the first byte is communication settings, then 2, 3 bytes are new Access Rights(Low byte first)
Response	0x00 read ok Other read fail

1.6.18 Create StdDataFile

Type	unsigned char _stdcall DFCreateSTDFile(int index,unsigned char* settings,unsigned char* filesize)
Description	Create a new standard data file under the current application of the card
Parameters	index- file number (1byte) setting- 3 bytes, the first byte is communication settings, then 2, 3 bytes are new Access Rights(Low byte first) filesize - file size (3byte Low byte first)
Response	0x00 read ok Other read fail

1.6.19 Create BackupDataFile

Type	unsigned char _stdcall DFCreateBackupDataFile(int index,unsigned char* settings,unsigned char* filesize)
Description	Create new data files under the current application of the card, support backup mechanism
Parameters	index- file number (1byte) setting- 3 bytes, the first byte is communication settings, then 2, 3 bytes are new Access Rights(Low byte first) filesize - file size (3byte Low byte first)
Response	0x00 read ok Other read fail

1.6.20 Create ValueFile

Type	unsigned char _stdcall DFCreateValueFile(int index,unsigned char* settings,unsigned char* lowsb,unsigned char* highsb,unsigned char* initsb,unsigned char limiten)
Description	Create a new value file under the current application of the card
Parameters	index- file number (1byte) setting- 3 bytes, the first byte is communication settings, then 2, 3 bytes are new Access Rights(Low byte first)

	lowsb- 4 byte length(Low byte first) and codes the lower limit which is valid for this file highsb - 4 byte length(Low byte first) and codes the upper limit which is valid for this file initsb - 4 byte length(Low byte first)the initial value of the value file limiten - the activation of the LimitedCredit feature, 0x00 means that LimitedCredit is disabled and 0x01 enables this feature
Response	0x00 read ok Other read fail

1.6.21 Create LinearRecordFile

Type	unsigned char _stdcall DFCreateLinearRecordFile(int index,unsigned char* settings,unsigned char* filesize,unsigned char* number)
Description	Create a new linear record file in the current application directory of the card
Parameters	index- file number (1byte) setting- 3 bytes, the first byte is communication settings, then 2, 3 bytes are new Access Rights(Low byte first) filesize- The size of a record,3 byte length(Low byte first) number- Maximum number of records,3 byte length(Low byte first)
Response	0x00 read ok Other read fail

1.6.22 Create CyclicRecordFile

Type	unsigned char _stdcall DFCreateCyclicRecordFile(int index, unsigned char* settings,unsigned char* filesize,unsigned char* number)
Description	Create a new cyclic record file in the current application directory of the card
Parameters	index- file number (1byte) setting- 3 bytes, the first byte is communication settings, then 2, 3 bytes are new Access Rights(Low byte first) filesize- The size of a record,3 byte length(Low byte first) number- Maximum number of records,3 byte length(Low byte first)
Response	0x00 read ok Other read fail

1.6.23 Delete File

Type	unsigned char _stdcall DFDeleteFile(int index)
Description	Delete the specified file in the current application directory of the card
Parameters	index- file number (1byte)

Response	0x00 read ok Other read fail

1.6.24 Write DataFile

Type	unsigned char _stdcall DFWriteDataFile(int index,unsigned char* address,unsigned char* length,unsigned char*data)
Description	Write data to Standard Data Files or Backup Data Files.
Parameters	index- file number (1byte) address -File offset address(3byte Low byte first) length - Written data length(3byte Low byte first) data - Written data
Response	0x00 read ok Other read fail

1.6.25 Read DataFile

Type	unsigned char _stdcall DFReadDataFile(int index,unsigned char* offset,unsigned char* len,unsigned char* OutData,unsigned int *DataLen)
Description	Read data from Standard Data Files or Backup Data Files.
Parameters	index- file number (1byte) offset-File offset address(3byte Low byte first) len- Read data length(3byte Low byte first) OutData- Out data DataLen -Out data len
Response	0x00 read ok Other read fail

1.6.26 Get ValueFile

Type	unsigned char _stdcall DFGetValueFile(int index,unsigned char* OutValue)
Description	Read the currently stored value from Value Files.
Parameters	index- file number (1byte) OutValue- The current value of the read file(4byte low byte first)
Response	0x00 read ok Other read fail

1.6.27 PlusValueFile

Type	unsigned char _stdcall DFPlusValueFile(int index,unsigned char* value)
Description	Increase a value stored in a Value File.
Parameters	index- file number (1byte) value- Increased value(4byte low byte first)

Response	0x00 read ok Other read fail

1.6.28 MinusValueFile

Type	unsigned char _stdcall DFMinusValueFile(int index,unsigned char* value)
Description	Decrease a value stored in a Value File.
Parameters	index- file number (1byte) value- decrease value(4byte low byte first)
Response	0x00 read ok Other read fail

1.6.29 Write Record

Type	unsigned char _stdcall DFWriteRecord(int index,unsigned char* offset,unsigned char* len,unsigned char* data)
Description	Write data to a record in a Cyclic or Linear Record File.
Parameters	index- file number (1byte) offset-offset within one single record (3byte Low byte first) length - length of data which is to be written to the record file(3byte Low byte first) data - Written data
Response	0x00 read ok Other read fail

1.6.30 Read Record

Type	unsigned char _stdcall DFReadRecord(int index,unsigned char* offset,unsigned char* recordnum,unsigned char* OutData,unsigned int *DataLen)
Description	Read out a set of complete records from a Cyclic or Linear Record File..
Parameters	index- file number (1byte) offset-offset of the newest record which is read out (3byte Low byte first) recordnum- number of records to be read (3byte Low byte first) OutData- out data DataLen - out data len
Response	0x00 read ok Other read fail

1.6.31 Clear RecordFile

Type	unsigned char _stdcall DFClearRecordFile(int index)
Description	Reset a Cyclic or Linear Record File to the empty state.

Parameters	index- file number (1byte)
Response	0x00 read ok Other read fail

1.6.32 Commit Transaction

Type	unsigned char _stdcall DFCommitTransion(void)
Description	Submit all previous write access operations of the backup data file, value file and record file in the current directory, so that the previous modification is valid
Parameters	no
Response	0x00 read ok Other read fail

1.6.33 Abort Transaction

Type	unsigned char _stdcall DFAbortTransion(void)
Description	Cancel all previous write access operations of the backup data file, value file and log file in the current directory, invalidating the previous modification
Parameters	no
Response	0x00 read ok Other read fail

1.6.34 Error Codes

0x00 Successful operation

0x0C No changes done to backup files, CommitTransaction /AbortTransaction not necessary

0x0E Insufficient NV-Memory to complete command

0x1C Command code not supported

0x1E CRC or MAC does not match data Padding bytes not valid

0x40 Invalid key number specified

0x7E Length of command string invalid

0x9D Current configuration / status does not allow the requested command

0x9E Value of the parameter(s) invalid

0xA0 Requested AID not present on PICC

0xA1 Unrecoverable error within application, application will be disabled

0xAE Current authentication status does not allow the requested command

0xAF Additional data frame is expected to be sent

0xBE Attempt to read/write data from/to beyond the file's/record's limits. Attempt to exceed the limits of a value file.

0xC1 P Unrecoverable error within PICC, PICC will be disabled

0xCA Previous Command was not fully completed Not all Frames were requested or

provided by the PCD

0xCD was disabled by an unrecoverable error

0xCE Number of Applications limited to 28, no additional CreateApplication possible

0xDE Creation of file/application failed because file/application with same number already exists

0xEE Could not complete NV-write operation due to loss of power,internal backup/rollback mechanism activated

0xF0 Specified file number does not exist

0xF1 Unrecoverable error within file, file will be disabled

1.7 PinPad function

1.7.1 Set Port number

Type	int _stdcall PinPadSetCom(unsigned char ComId)
Description	Set the serial port number
Parameters	ComId- Com number
Response	0x00 read ok Other read fail

1.7.2 Get Port number

Type	int _stdcall PinPadGetCom(unsigned char *ComId)
Description	Get the serial port number
Parameters	ComId- Out Com number
Response	0x00 read ok Other read fail

1.7.3 PinPadInit

Type	int _stdcall PinPadInit(void)
Description	Initialization PinPad
Parameters	none
Response	0x00 read ok Other read fail

1.7.4 Get PinPad Information

Type	int _stdcall PinPadGetInfo(unsigned char *OutVer,unsigned char *OutSN)
Description	Get PinPad Information
Parameters	OutVer- Out Firmware version (Maximum 48 bytes of space required) OutSN- Out SerialNo (Maximum 40 bytes of space required)
Response	0x00 read ok Other read fail

1.7.5 Load MasterKey

Type	int _stdcall PinpadLoadMasterKey(unsigned int mkeyID, unsigned char keyType, unsigned char *keybuf, unsigned char keylen)
Description	Load MasterKey
Parameters	mkeyID- KeyNo (1--50) keyType- KEYTYPE_PINEK (PinMasterKEY) KEYTYPE_DATAEK(EncryptMasterKey) KEYTYPE_DATADK(DecryptMasterKey) Keybuf - Plain Master key data Keylen - Master key data len
Response	0x00 read ok Other read fail

1.7.6 Load WorkKey

Type	int _stdcall PinpadLoadWorkKey(unsigned int wkeyID, unsigned int mkeyID, unsigned char keyType, unsigned char *keybuf, unsigned char keylen,unsigned char *kcv,unsigned char kcvlen)
Description	Load WorkKey
Parameters	wkeyID- KeyNo (1--50) keyType- KEYTYPE_PINEK (PinWorkKEY) KEYTYPE_DATAEK(EncryptWorkKey) KEYTYPE_DATADK(DecryptWorkKey)

	Keybuf - Cipher Work key data(Encrypted by the corresponding master key) Keylen - Work key data len Kcv - kcv data (Encrypt 8 bytes 00 using the plain work key) Kcvlen - kcv data len (kcvlen==0,Then no KCV verification)
Response	0x00 read ok Other read fail

1.7.7 Get PINBlock

Type	int _stdcall PinPadGetPin(unsigned char *Title, unsigned char Mode, unsigned char *PAN, unsigned char *PinData, unsigned char MinPINLen, unsigned char MaxPINLen, unsigned short TimeOut, unsigned short KeyIndex)
Description	Get pinblock
Parameters	Title - Amount(String format E.g"1234" is 12.34) Mode - 0：ISO9564_format0 1：ISO9564_format0 without master account operation PAN - Complete card number PinData - Out Cipher PINBlock MinPINLen - Password minimum length 4 MaxPINLen - Password maximum length 12 TimeOut- Enter the timeout period in seconds KeyIndex - KeyNo (1--50)
Response	0x00 read ok Other read fail

1.7.8 GetMac

Type	int _stdcall PinPadGetMac(unsigned char CalMode, unsigned char *pMac, unsigned char *Data,unsigned int DataLen, unsigned int KeyIndex)
Description	Get mac
Parameters	CalMode - 0: MAC_X99 1:MAC_X919 pMac- Out mac data Data- In data DataLen- In data len KeyIndex - KeyNo (1--50)
Response	0x00 read ok Other read fail

1.7.9 CalcDES

Type	int _stdcall PinPadCalcDES(unsigned char CalcMode, unsigned char *OutData, unsigned char *InData,unsigned int InDataLen, unsigned int KeyIndex, unsigned char CbcMode,unsigned char *InvData)
Description	Calc des
Parameters	CalMode - 0: ENCRYPT 1: DECRYPT OutData- Out calc data InData- In data InDataLen- In data len (Must be a multiple of 8) KeyIndex - KeyNo (1--50) CbcMode - 0: ECB mode 1: CBC mode InvData - Initial data only valid in CBC mode
Response	0x00 read ok Other read fail

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.