IC READER

DLL Library Reference Manual

(ISO14443 Type A)

Contents

Contents 2
API Function Sets 3
1.1 System Function 3
1.1.1 API_GetSysComm 3
1.1.2 API_OpenComm 3
1.1.3 API_CloseComm 3
1.1.4 API_SetDeviceAddress 4
1.1.5 API_SetBaudrate 4
1.1.6 API_SetSerNum 5
1.1.7 API_GetSerNum 5
1.1.8 API_GetVersionNum 5
1.1.9 API_ControlLED 6
1.1.10 API_ControlBuzzer 6
1.2 ISO14443 Type-A Function 7
1.2.4 API_MF_Halt 7
1.2.5 API_MF_Read 7
1.2.6 API_MF_Write 8
1.2.7 API_MF_InitVal 8
1.2.8 API_MF_Dec 9
1.2.9 API_MF_Inc 10
1.3 Mifare UltraLight 11
1.31 ULRead 11
1.3.2 ULWrite 11
1.3.3 getULCardSN 12
2. APPENDIX 13

API Function Sets

1.1 System Function

1.1.1 API_GetSysComm

Function	Get COM port information from system
Prototype	int API_GetSysComm(unsigned char *Buffer)
Description	Input parameter： none Output parameter： Buffer[0]：serial port numbers had been detected in system Buffer[1]：if Buffer[0] > 0, Buffer[1] is kept as port number Buffer[2]：if Buffer[0] > 1, Buffer[2] is kept as port number … Buffer[N]：if Buffer[0] > N-1, Buffer[2] is kept as port number
Return	return as 0 when succeed, otherwise will return 1

1.1.2 API_OpenComm

Function	Open assigned Com port and set baudrate
Prototype	HANDLE API_OpenComm(int nCom, int nBaudrate)
Description	Input parameter： nCom：assigned Com number nBaudrate：baudrate (9600,19200,38400,57600,115200） Output parameter： none
Return	return with Com handle when succeed, otherwise will be 0

1.1.3 API_CloseComm

Function	Close assigned Com
Prototype	BOOL API_CloseComm(HANDLE commHandle)
Description	Input parameter： commHandle：assigned Com Handle Output parameter： none
Return	return as TRUE when succeed, otherwise will be FALSE

1.1.4 API_SetDeviceAddress

Function	Set reader device address
Prototype	int API_SetDeviceAddress(HANDLE commHandle, int DeviceAddress, unsigned char NewAddress, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress: device address, range from 0~255 NewAddress：new address, range from 0~255 Output parameter： Buffer If succeed, Buffer[0] will be the new address after setted If failed, Buffer[0] is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.1.5 API_SetBaudrate

Function	Set reader's communication baudrate
Prototype	int API_SetBaudrate(HANDLE commHandle, int DeviceAddress, unsigned char NewBaud, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0 ~ 255 NewBaud：new baudrate code 0x00 – 9600 bps 0x01 – 19200 bps 0x02 – 38400 bps 0x03 – 57600 bps 0x04 – 115200 bps Other value--9600 bps Output parameter： Buffer If succeed, Buffer[0] will be the new baudrate code after setted If failed, Buffer[0]is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.1.6 API_SetSerNum

Function	Set device serial number
Prototype	int API_SetSerNum(HANDLE commHandle, int DeviceAddress, unsigned char *NewValue, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0 ~ 255 NewValue：new serial number（8byte） Output parameter： Buffer If succeed, Buffer[0]=0x80 If failed，Buffer[0] is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.1.7 API_GetSerNum

Function	Get device serial number
Prototype	int API_GetSerNum(HANDLE commHandle, int DeviceAddress, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0~255 Output parameter： Buffer If succeed,Buffer[0] is the reader's address, then Buffer[1-8] is the serial number If failed, Buffer[0] is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.1.8 API_GetVersionNum

Function	Get reader version number
Prototype	int API_GetVersionNum(HANDLE commHandle, int DeviceAddress, char *VersionNum)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0~255

	Output parameter： VersionNum If succeed，VersionNum[0-7] is the version If failed， VersionNum[0] is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.1.9 API_ControlLED

Function	Control reader's LED
Prototype	int API_ControlLED(HANDLE commHandle, int DeviceAddress, unsigned char freq, unsigned char duration, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0~255 freq：cycle count of LED lighting in one cycle（20ms for one cycle ） duration：cycle time of LED status(1seconds for one cycle） Output parameter： Buffer If succeed，Buffer[0]=0x80 If failed，Buffer[0] is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.1.10 API_ControlBuzzer

Function	Control reader's buzzer
Prototype	int API_ControlBuzzer(HANDLE commHandle, int DeviceAddress, unsigned char freq, unsigned char duration, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0~255 freq： cycle counts of Buzzer making sound in one cycle(100ms for one second) duration：cyclic time of the buzzer status (1 second for one cycle) Output parameter： Buffer If succeed,Buffer[0]=0x80 If failed，Buffer[0] is the status code returned from reader, detail definition, please refer to APPENDIX

Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.2 ISO14443 Type-A Function

1.2.4 API_MF_Halt

Function	Operate to halt the reader
Prototype	int API_MF_Halt(HANDLE commHandle, int DeviceAddress)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address,range from 0x00~0xFF Output parameter： none
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.2.5 API_MF_Read

Function	Integrated functions with request, anticollision, select card, password authentication,, etc in one command to finish reading card
Prototype	int API_MF_Read(HANDLE commHandle, int DeviceAddress, unsigned char Mode, unsigned char Blk_add, unsigned char Num_blk, unsigned char *Key, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0x00~0xFF Mode： control reading mode 0x00：Idle mode + Key A 0x01：All mode + Key A 0x02：Idle mode + Key B 0x03：All mode + Key B Blk_add：start address of the block to be read, range from 0x00~0xFE Num_blk：lengths of block to be read, range from 1~4（Mifare 1k card) Key[0-5]：6 bytes key Output parameter： Buffer： If succeed, Buffer[0]: return data length Buffer[1-4]：4 bytes card serial number (from low to high)) Buffer[5-N]： data of card If failed, Buffer[0]: is the status code returned from reader, detail

	definition, please refer to APPENDIX Note：this function enable to read block of this sector only, because each sector has its own unique key
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.2.6 API_MF_Write

Function	Integrated functions with request, anticollision, select card, password authentication, write card, etc in one command, to finish writing operation
Prototype	int API_MF_Write(HANDLE commHandle, int DeviceAddress, unsigned char Mode, unsigned char Blk_add, unsigned char Num_blk, unsigned char *Key, unsigned char *Senddata, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0x00~0xFF Mode：control reading mode 0x00：Idle mode + Key A 0x01：All mode + Key A 0x02：Idle mode + Key B 0x03：All mode + Key B Blk_add：start address of block to be written, range from 0x00~0xFE Num_blk：numbers of blocks to be written,range from 1~4（Mifare 1K） Key[0-5]：6 bytes Key Senddata[0-M]：data to be written（16 *Num_blk byte） Output parameter： Buffer： If succeed，Buffer[0]: length of data be return Buffer[1-4]：4 bytes card serial number (low byte in front) If failed， Buffer[0]: is the status code returned from reader, detail definition, please refer to APPENDIX Note：this function enable to read block of this sector only, because each sector has its own unique key
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.2.7 API_MF_InitVal

Function	E-wallet initialize
Prototype	int API_MF_InitVal(HANDLE commHandle, int DeviceAddress, unsigned char Mode, unsigned char Sec_num, unsigned char *Key, unsigned char *Value, unsigned char *Buffer)

Description	Input parameter： commHandle：Com handle DeviceAddress：device address, range from 0x00~0xFF Mode：（operation mode） 0x00: Idle mode + Key A 0x01: All mode + Key A 0x02: Idle mode + Key B 0x03: All mode + Key B Sec_num：Sector Number Key[0-5]：6 bytes key Value[0-3]：initialize value Output parameter： Buffer： If succeed，Buffer[0]: length of data be returned Buffer[1-N]: data returned If failed，Buffer[0]: is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.2.8 API_MF_Dec

Function	E-wallet decrement
Prototype	int API_MF_Dec(HANDLE commHandle, int DeviceAddress, unsigned char Mode, unsigned char Sec_num, unsigned char *Key, unsigned char *Value, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address, device address, range from 0x00~0xFF Mode：（operation mode） 0x00: Idle mode + Key A 0x01: All mode + Key A 0x02: Idle mode + Key B 0x03: All mode + Key B Sec_num：Sector Number Key[0-5]：6 bytes key Value[0-3]：value to be decreased Output parameter： Buffer： If succeed，Buffer[0]: data length returned Buffer[1-N]: data returned If failed，Buffer[0]:is the status code returned from reader, detail definition, please refer to APPENDIX。

Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.2.9 API_MF_Inc

Function	E-wallet increment
Prototype	int API_MF_Inc(HANDLE commHandle, int DeviceAddress, unsigned char Mode, unsigned char Sec_num, unsigned char *Key, unsigned char *Value, unsigned char *Buffer)
Description	Input parameter： commHandle：Com handle DeviceAddress：device address，range from 0x00~0xFF Mode：（operation mode） 0x00: Idle mode + Key A 0x01: All mode + Key A 0x02: Idle mode + Key B 0x03: All mode + Key B Sec_num：Sector Number Key[0-5]：6 bytes key Value[0-3]：value to be increased Output parameter： Buffer： If succeed，Buffer[0]: data length returned Buffer[1-N]: data returned If failed， Buffer[0]: is the status code returned from reader, detail definition, please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.3 Mifare UltraLight

1.31 ULRead

Function	To read one block of Mifare Ultralight card
Prototype	int ULRead(int handle,int address,byte blk_add,byte[] SN,byte[] buffer);
Description	Read one block of Mifare Ultralight card Input parameter： commHandle — Com handle DeviceAddress — device address, range from 0—255 blk_add Start address of block to be read, range from 0--63 Output parameter snr buffer If succeed, buffer[0-4]: data returned from card(4 byte) If failed, buffer[0]: is the status returned from reader, detail description please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.3.2 ULWrite

Function	To write one block of Mifare Utralight
Prototype	int ULWrite(int handle,int address,byte blk_add,byte[] SN,byte[] buffer);
Description	Input parameter： commHandle — Com handle DeviceAddress — device address, range from 0—255 blk_add Start address of block to be written, range from 0--63 SN Reserved buffer Data to be written (4 bytes) Output parameter: buffer If succeed，buffer[0-6]: 7 bytes card UID (low byte in front)

	If failed， buffer[0]: is the status returned from reader, detail description please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

1.3.3 getULCardSN

Function	Integrated functions with request, anticollision, select card, etc in one command, to finish Get card SN
Prototype	int getULCardSN(int handle,int address,byte[] SN);
Description	I Input parameter： commHandle — Com handle DeviceAddress — device address, range from 0—255 Output parameter： SN— succeed, return card number 0x00 – one card be detected 0x01 – several cards be detected.(but another card UID detected when invoking again) If failed，SN[0]: is the status returned from reader, detail description please refer to APPENDIX
Return	Return as 0 when succeed, otherwise will be not 0, detail definition please refer to APPENDIX

2 APPENDIX

API Function return value

Return	Description
0x00	CMD execute succeed
0x02	Data length received not matching up
0x03	Serial port transmitting failed
0x04	No data received by serial port
0x05	Device address not matching up
0x07	Checksum error
0x0A	Input parameter error, please refer to unspecific functions declaration

Reader Status return code

Status	Description
0x00	CMD execute succeed
0x01	CMD operate failed（detail description please refer to functions)
0x80	Parameter setup succeed
0x81	Parameter setup failed
0x82	Communication timeout
0x83	Card not existing
0x84	Receiving card data error
0x85	Input parameter or input CMD format error
0x87	Unknown error
0x89	input parameter or input CMD format error
0x8A	Block initializing mistake
0x8B	Get wrong serial number when anticollision
0x8C	Password anthentication error
0x8f	Input command code not existing
0x90	command not supported by card
0x91	command format mistake