Standard PC/SC Series IC Card Reader

General Technical Manual

(Revision 2.23)

Jinmuyu Electronics Co., Ltd

March 30, 2015

Please read this manual carefully before using. If any problem, please feel free to contact us, we will offer satisfied answer ASAP.

Contents

1 Introduction

This document is suitable for MR791, MR7911, MR801, MR811 and MR8111 and so on.

The above mentioned RFID Readers are designed according to USB PC/SC standard. It uses the Microsoft CCID driver and standard operation method, so you could refer to other standard PC/SC documents too.

2 Driver Installation and System Identification

If your PC system is no CCID driver, it will remind you to install the driver when the PC/SC Reader connects with your PC via USB interface at the first time. But no worry, we can offer you the CCID driver, you can get it from our website or we will send it to you by mail.

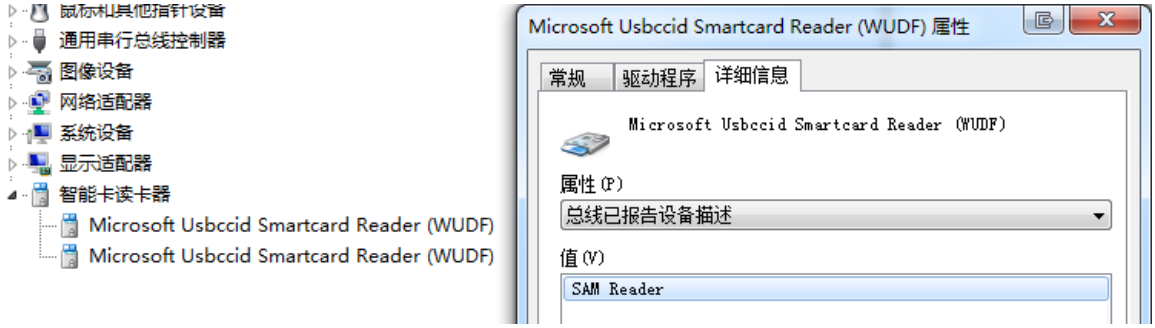
After installation successfully, it will show you two Smart Card Readers-- "Microsoft Usbccid Smartcard Reader (WUDF)" in your PC Device Manager, like the following picture.

2.1 Contactless Reader

The Reader can Read/Write the Contactless Smart Card and Memory Card within the Antenna fields.

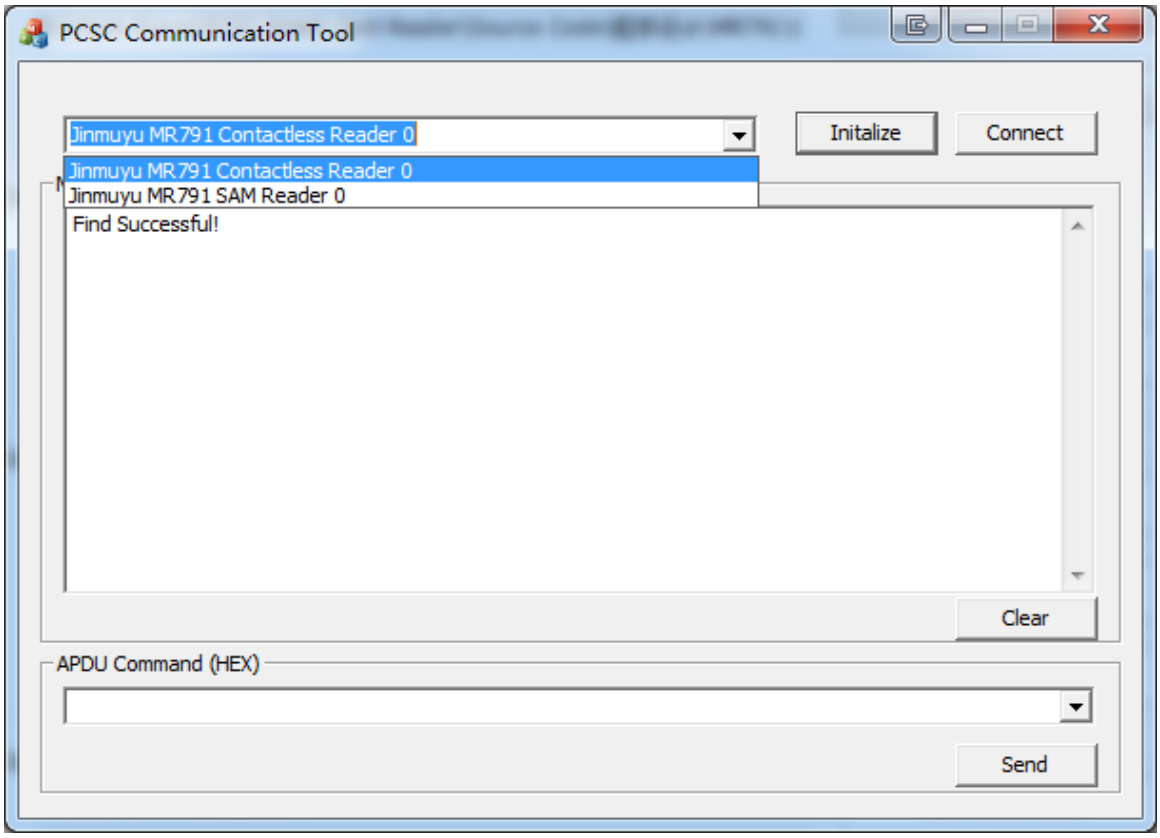
2.2 SAM Reader

Also it can operate ISO7816 SAM cards in the Reader internal SAM slots.



3 PC Software

First opening "PC/SC Communication Tool", then to click "Initialize" button, there are two Readers will be shown like the following picture.



4 PICC Interface Description

4.1 ATR Generation

If the reader detects a PICC, an ATR will be sent to the PCSC driver for identifying the PICC. Because these readers are standard PC/SC device, you could refer to other standard documents too.

4.2 ATR format for ISO 14443 Part 3 PICCs

Byte No.	Value(Hex)	Designation	Description
0	3B	Initial Header	
1	8N	T0	Higher nibble 8 means no TA1, TB1, and TC1 only TD1 is following. Lower nibble n is the number of historical bytes (HistByte 0 to HistByte n-1).
2	80	TD1	Higher nibble 8 means no TA2, TB2, and TC2 only TD2 is following. Lower nibble 0 means T = 0.
3	01	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1.
4 to 3+N	80	T1	Category indicator byte, 80 means A status indicator may be present in an optional COMPACT-TLV data object.
	4F	Tk	Application identifier Presence indicator
	0C		Length
	RID		Registered application provider identifier: (RID) # A0 00 00 03 06h
	SS		Byte for Standard
	C0…C1		Bytes for Card Name
	00 00 00 00h	RFU	RFU # 00 00 00 00h
4+N	UU	TCK	XOR of all the bytes T0 to Tk

4.3 ATR format for ISO 14443 Part 4 PICCs

Byte Nr	Value(Hex)	Designation	Description
0	3B	Initial Header	
1	8N	T0	Higher nibble 8 means no TA1, TB1, and TC1 only TD1 is following. Lower nibble n is the number of historical bytes (HistByte 0 to HistByte n-1).
2	80	TD1	Higher nibble 8 means no TA2, TB2, and TC2 only TD2 is following. Lower nibble 0 means T = 0.
3	01	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1.
4 to 3+N	XX	T1	Historical bytes: ISO14443A: The historical bytes from ATS response. Refer to the ISO14443-4 specification. ISO14443B: The higher layer response from the ATTRIB response. Refer to the ISO14443-3 specification.
	XX xx XX	Tk	
4+N	UU	TCK	XOR of bytes T0 to Tk

5 Contactless Reader Commands

5.1 Get Data

This command will retrieve the SNR or ATS of the present card.

Get UID APDU Format (5 Bytes)

Command	CLA	INS	P1	P2	Le
Get Data	0xFF	0xCA	0x00 0x01	0x00	0x00 (Full Length)

Get UID Response Format (UID + 2 Bytes) if P1 = 0x00

Response	Data					
Result	UID(LSB)	--	--	UID(MSB)	SW1	SW2

Get ATS of a ISO 14443 A card (ATS + 2 Bytes) if P1 = 0x01

Response	Data		
Result	ATS	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x6A 81	No such function

Example:

1. To get the connected PICC SNR

   APDU = 0xFF 0xCA 0x00 0x00 0x00

2. To get the ATS of the “connected ISO 14443A PICC”

   APDU = 0xFF 0xCA 0x01 0x00 0x00

5.2 MIFARE Classic Cards Commands（T=CL Emulation）

5.2.1 Load Authentication Keys

This command will just load (write) the keys in the IFD’s designated memory. The key will be of two different types; the reader key and the card key. This command can be used for all kinds of contactless cards.

Reader Key: This key will be used to protect the transmission of secured data e.g. card key from the application to the reader. Example: The application may encrypt the data with one of the reader keys:

It has to tell the reader that it has done so, and the number of the key used.

Card Key: This is the card specific key (e.g. for MIFARE it is MIFARE key). This key can be volatile or non-volatile.

The coding of the command provides the following mechanisms:

• Load keys in either container: Reader key container to be used for transmission protection and card key container for card authentication.

• Transmission of the loaded key in plain or encrypted using a key out of the reader key container: The key to be used is indicated in P1 by its number. P2 is indicating the address within the container, where the key shall be stored.

• The containers can be located in volatile or non-volatile memory.

Load Authentication Keys APDU Format（n byte):

Command	CLA	INS	P1	P2	Lc	Data
Load Authentication Keys	0xFF	0x82	Key Structure	Key Number	Key length	Key

Key Structure: 1byte

b7	b6	b5	b4	b3	b2	b1	b0	Description
X								0: Card Key; 1 Reader Key
	X							0: Plain Transmission, 1: Secured Transmission
		X						0: Keys are loaded into the IFD volatile memory 1: Keys are loaded into the IFD non-volatile memory.
			0					RFU
				000X				If b6 is set, it is the Reader Key number that has been used for the encryption, else it is ignored by the IFD. The maximum of 16-reader keys is possible. Typically an IFD uses two reader keys only.

The non-volatile Key, which is stored in the Flash of the Reader, has storage time limitation. Users need pay more attention to it.

Key Number: 1byte

When b7=0: Value: 0x00~07, The Reader can store 8 card's Keys.

When b7=1: Value: 0x00~01, The Reader can store 2 Reader's Keys.

Key Length: 1byte

When loading the Reader Key, the length of the Key must be 16bytes, or the Reader will return fail.

When loading the Card Key by way of plain text, the Reader no any restriction for the Key length.

When loading the Card Key by way of cipher text, the Key length must be 8bytes or 16bytes.

Key: N byte

Load the "Reader Key" or "Card Key" value into the Reader Writer.

Load Authentication Keys Response Format（2byte）

Response	Data Out	
Result	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x6A 81	No such function

Example:

1. Load Key {0x00 11 22 …. FF} in plain into the Reader's Key storage location 0x00

   APDU = 0xFF 82 A0 00 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

2. Load Key {0xFF FF FF FF FF FF} in plain into the Card's Key buffer location 0x00

   APDU = 0xFF 82 00 00 06 FF FF FF FF FF FF

3. Load Key {0xFF FF FF FF FF FF} in encrypted into Card's Key storage location 0x01. The encrypted key is {0x00 11 22 …. FF}

   The 6bytes Key adds extra following 2 bytes 0x00. The encrypted key encryption value (via using encrypted key) is 0xC0 D6 1E B0 84 F9 43 57

   APDU = 0xFF 82 60 01 08 C0 D6 1E B0 84 F9 43 57

5.2.2 Authentication

The application provides the number of the key used for the MIFARE 1K/4K card authentication. The specific key must be already in the reader. Two type authentication keys: TYPE_A and TYPE_B.

Load Authentication Keys APDU Format (6 Bytes) [Obsolete]

Command	CLA	INS	P1	P2	P3	Data
Authentication	0xFF	0x88	0x00	Block Number	Key Type	Key Number

Load Authentication Keys APDU Format (10 Bytes)

Command	CLA	INS	P1	P2	Lc	Data
Authentication	0xFF	0x86	0x00	0x00	0x05	Authenticate Data Bytes

Authenticate Data Bytes (5 Byte)

Byte1	Byte 2	Byte 3	Byte 4	Byte 5
Version01	0x00	Block Number	Key Type	Key Number

Block Number: 1 Byte. This is the memory block to be authenticated.

Key Type: 1 Byte

0x60 = Key is used as a TYPE A key for authentication.

0x61 = Key is used as a TYPE B key for authentication.

Key Number: 1 Byte

0x00 ~ 0x1F = Key Location.

Load Authentication Keys Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x6A 81	No such function

Example:

1. Authenticate the Block 0x04 with a {TYPE A, key number 0x00}. For PC/SC V2.01, Obsolete

   APDU = 0xFF 88 00 04 60 00

2. Authenticate the Block 0x04 with a {TYPE A, key number 0x00}. For PC/SC V2.07

   APDU = 0xFF 86 00 00 05 01 00 04 60 00

5.2.3 Read Binary Blocks

This command is used for retrieving “data blocks” from the PICC. The data block/trailer block must be authenticated first.

Read Binary APDU Format (5 Bytes)

Command	CLA	INS	P1	P2	Le
Read Binary Blocks	0xFF	0xB0	0x00	Block Number	Number of Bytes to Read

Reference:

Block Number (1 Byte): The block to be accessed

Number of Bytes to Read (1 Byte): Maximum 16 bytes

Read Binary Block Response Format (N + 2 Bytes)

Response	Data Out		
Result	N=0 ~ 16	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.

Error	0x6A 81	No such function

Example:

1. Read 16 bytes from the binary block 0x04 (MIFARE 1K or 4K)

   APDU = 0xFF B0 00 04 10

2. Read 4 bytes from the binary Page 0x04 (MIFARE Ultra light)

   APDU = 0xFF B0 00 04 04

3. Read 16 bytes starting from the binary Page 0x04 (MIFARE Ultra light) (Pages 4, 5, 6 and 7 will be read)

   APDU = 0xFF B0 00 04 10

5.2.4 Update Binary Blocks

This command is used for writing "data blocks" into the PICC. The data block/trailer block must be authenticated.

Update Binary APDU Format (4 or 16 + 5 Bytes)

Command	CLA	INS	P1	P2	Lc	Data
Update Binary Blocks	0xFF	0xD6	0x00	Block Number	Number of Bytes to Update	Block Data: MIFARE Ultra Light: 4bytes MIFARE 1K/4K: 16bytes

Reference:

Block Number (1 Byte): The starting block to be updated.

Number of Bytes to Update (1 Byte):

16 bytes for MIFARE 1K/4K;

4 bytes for MIFARE Ultra light;

Block Data (4 or 16 Bytes): The data will be written into the binary block/blocks.

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x6A 81	No such function

Example:

1. Update the binary block 0x04 of MIFARE 1K/4K with Data {00 01 ... 0F}

   APDU = 0xFF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

2. Update the binary block 0x04 of MIFARE Ultra light with Data {00 01 02 03}

   APDU = 0xFF D6 00 04 04 00 01 02 03

5.2.5 Value Block Operation

This command increments the value of a data object if the card supports this functionality.

Value Block Operation APDU Format (10 Bytes)

Command	CLA	INS	P1	P2	Lc	Data	
Value Block Operation	0xFF	0xD7	0x00	Block Number	0x05	VB_OP	VB_Value：4bytes（LSB..MSB）

Reference:

Block Number (1 Byte): The value block to be manipulated.

VB_OP (1 Byte):

0x00 = Store the VB_Value into the block. The block will then be converted to a value block.

0x01 = Increment the value of the value block by the VB_Value. This command is only valid for value block.

0x02 = Decrement the value of the value block by the VB_Value. This command is only valid for value block.

VB_Value (4 Bytes): The value used for value manipulation. The value is a signed long integer (4bytes).

Value Block Operation Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x6A 81	No such function

5.2.6 Read Value Block

This command is used for retrieving the value from the value block. This command is only valid for value block.

Read Value Block APDU Format (5 Bytes)

Command	CLA	INS	P1	P2	Lc
Read Value Block	0xFF	0xB1	0x00	Block Number	0x04

Reference:

Block Number (1 Byte): The value block to be accessed.

Read Value Block Response Format (4 + 2 Bytes)

Response	Data Out		
Result	Value(LSB…MSB)	SW1	SW2

Reference:

Value (4 Bytes): The value returned from the card. The value is a signed long integer (4 bytes).

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x6A 81	No such function

5.2.7 Restore Value Block

This command is used to copy a value from a value block to another value block.

Restore Value Block APDU Format (7 Bytes)

Command	CLA	INS	P1	P2	Lc	Data	
Restore Value Block	0xFF	0xD7	0x00	Source Block Number	0x02	0x03	Target Block Number

Reference:

Source Block Number (1 Byte): The value of the source value block will be copied to the target value block.

Target Block Number (1 Byte): The value block to be restored. The source and target value blocks must be in the same sector.

Restore Value Block Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x6A 81	No such function

Example:

1. Store a value “1” into block 0x04

   APDU = 0xFF D7 00 04 05 00 01 00 00 00

2. Read the value block 0x04

APDU = 0xFF B1 00 04 00

3. Copy the value from value block 0x04 to value block 0x05

APDU = 0xFF D7 00 04 02 03 05

4. Increment the value block 0x05 by “5”

APDU = 0xFF D7 00 05 05 01 05 00 00 00

5.3 Contactless Smart Card Operation Loop

5.3.1 ISO14443-4 Card Operation

Basic Operation Loop:

Step 1 Put the CPU card into Contactless Reader antenna field

Step 2 Connect Contactless Reader

Step 3 Send APDU command

Get 8bytes Random

--> 0x00 84 00 00 08

<-- 0x1A F7 F3 1B CD 2B A9 58 90 00

5.3.2 MIFARE 1K/4K Card Operation

Basic Operation Loop:

Step 1 Put the MIFARE 1K/4K card into Contactless Reader antenna field

Step 2 Connect Contactless Reader

Step 3 Send MIFARE 1K/4K card operation commands

Loading Key:

--> 0xFF 82 00 00 06 FFFFFFFFFFFF

<-- 0x90 00

Authenticate 0x04 block via Type A key which is stored in 0x00 position.

--> 0xFF 86 00 00 05 01 00 04 60 00

<-- 0x90 00

Read data from 0x04 block

--> 0xFF B0 00 04 10

<-- 0xF6 FF FF FF 09 00 00 00 F6 FF FF FF 04 FB 04 FB 90 00

Write data into 0x04 block

--> 0xFF D6 00 04 10 00112233445566778899AABBCCDDEEFF

<-- 0x90 00

Purse Initialization

--> 0xFF D7 00 04 05 00 00000000

<-- 0x90 00

Purse Increment

--> 0xFF D7 00 04 05 01 05000000

<-- 0x90 00

Purse Decrement

--> 0xFF D7 00 04 05 02 0A000000

<-- 0x90 00

Purse Copy

--> 0xFF D7 00 04 02 03 05

<-- 0x90 00

Read Purse Value

--> 0xFF B1 00 05 04

<-- 0xF6 FF FF FF 90 00

5.3.3 MIFARE Ultra Light Card Operation

Basic Operation Loop:

Step 1 Put the MIFARE Ultra Light card into Contactless Reader antenna field

Step 2 Connect Contactless Reader

Step 3 Send MIFARE Ultra Light card operation commands

Write data into block

--> 0xFF D6 00 04 04 00 11 22 33

<-- 0x90 00

Read data from block

--> 0xFF B0 00 04 04

<-- 0x00 11 22 33 90 00

6 SAM Reader Commands

6.1 Flash Commands

6.1.1 Read FLASH

This is used to read data from the internal Flash of the Reader. Note: the "offset address" (P2) + "reading length"(Le) can't be 256bytes, or the operation will fail.

Read FLASH APDU Format (5byte)

Command	CLA	INS	P1	P2	Le
Select SAM	0xFF	0xB0	0x00~FF	0x00~FF	0xXX

Reference:

P1: 1byte specified reading data page, 0x00~FF, total 256pages.

P2: 1byte the offset address of the "specified reading data page".

Le: 1byte reading data length, 0x01~FE. Max. Length: 0xFE

Select SAM Response Format (2bytes)

Response	Data Out		
Result	Data	SW1	SW2

Reference:

Data: the data to be read.

Response State:

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x6A 81	No such function

Example:

1. Read out 16bytes data from page3 in Flash. Note: Address0x00 is as beginning.

   APDU = 0xFF B0 03 00 10

6.1.2 Write FLASH

This is used to write data into the internal Flash of the Reader. Note: the "offset address"(P2) + "reading length"(Le) can't be 256bytes, or the operation will fail.

Write FLASH APDU Format (N+5bytes)

Command	CLA	INS	P1	P2	Lc	Data
Select SAM	0xFF	0xD6	0x00~FF	0x00~FF	0xXX	Data

Reference:

P1: 1byte specified writing data page, 0x00~FF, total 256pages.

P2: 1byte the offset address of the "specified writing data page".

Lc: 1byte writing data length, 0x01~FE.

Data: n byte data to be written.

Select SAM Response Format (2bytes)

Response	Data Out	
Result	SW1	SW2

Response State:

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x6A 81	No such function

Example:

1. Write 16bytes data "0x00 11 22 … FF" into page3 in Flash. Note: Address0x00 is as beginning.

   APDU = 0xFF B6 03 00 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

6.1.3 Select Operation Sector in Flash

This is used to select the operation sector in Flash.

Select FLASH Operation Sector APDU Format (7bytes)

Command	CLA	INS	P1	P2	Lc	Data
Select SAM	0xFF	0x00	0x80	0x00	0x02	Sector

Reference:

Sector: 2byte, specified Flash operation sector, 0x0000~0007, 8Sectors*256Pages*256Bytes = 512Kbytes.

Device offers 512Kbytes storage Flash.

Select FLASH Operation Sector Response Format (2bytes)

Response	Data Out	
Result	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x6A 81	No such function

Example:

1. Select the second sector in Flash.

   APDU = 0xFF 00 80 00 02 00 02

6.2 Private APDU

6.2.1 Reset SAM

This is used to reset SAM card and operate the selected response SAM card. Device can support 3 SAM slots, but each time just can operate one SAM card.

Reset SAM APDU Format (6bytes):

Command	CLA	INS	P1	P2	Lc	Data		
Reset SAM	0xFF	0x00	0x60	0x00	0x03	SAM Slot Number	Reset Baud Rate	Communication Baud Rate

Reference:

SAM Slot Number (1byte)

0x01 = SAM slot 1

0x02 = SAM slot 2

0x03 = SAM slot 3

0x04 = SAM slot 4, MR7911only

Reset Baud Rate (1 byte)

Communication Baud Rate (1 byte)

0x00 = 9600

0x01 = 19200

0x02 = 38400

0x03 = 55800

0x04 = 57600

0x05 = 115200

0x06 = 220400

Reset SAM Response Format (2bytes)

Response	Data Out		
Result	Reset Information	SW1	SW2

Reference:

Reset Information: Return SAM card reset information when operations succeed; if fail, "Reset Information" is no existing.

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x63 00	The operation is failed.
Error	0x67 00	Data length Error
Error	0x6A 81	No such function

Example:

1. Choose SAM slot 1 when resetting, Reset Baud Rate 0x00 (9600), Communication Baud Rate 0x00 (9600).

APDU = 0xFF 00 60 00 03 01 00 00

6.2.2 Select SAM

This is used to choose the operated SAM card.

Select SAM APDU Format (6bytes)

Command	CLA	INS	P1	P2	Lc	Data
Select SAM	0xFF	0x00	0x61	0x00	0x02	SAM Slot Number

Reference:

SAM Slot Number (1byte)

0x01 = SAM slot 1

0x02 = SAM slot 2

0x03 = SAM slot 3

0x04 = SAM slot 4, MR7911 support

Select SAM Response Format (2bytes)

Response	Data Out	
Result	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x6A 81	No such function

Example:

1. Choose SAM slot 1

   APDU = 0xFF 00 61 00 01 01

6.2.3 Read Device Version

This is used to read the Device firmware version.

Read Device Version APDU Format (5bytes)

Command	CLA	INS	P1	P2	Le
Select SAM	0xFF	0x00	0xFF	0x01	0x00

Reference:

Le: 1byte 0x00 = read the whole information.

Select SAM Response Format (2bytes)

Response	Data Out		
Result	Firmware Version	SW1	SW2

Response State

Result	SW1 SW2	Meaning
Success	0x90 00	The operation is completed successfully.
Error	0x6A 81	No such function

Example:

1. Read Firmware Version information

APDU = 0xFF 00 FF 01 00

6.3 SAM Card Operation Loop

6.3.1 SAM Card Operation

Basic Operation Loop:

Step 1 Connect SAM Reader

Step 2 Send Smart Card APDU command

Get 8 bytes Random

--> 0x00 84 00 00 08

<-- 0x1A F7 F3 1B CD 2B A9 58 90 00

6.3.2 SAM Card Operation Extension

Basic Operation Loop:

Step 1 Connect SAM Reader

Step 2 Send Private APDU command

--> 0xFF 00 60 00 03 02 00 00(reset and select current SAM card command)

<-- 0x3B XX… 90 00

Step 3 Send Smart Card APDU command

Get 8 bytes Random

--> 0x00 84 00 00 08

<-- 0x1A F7 F3 1B CD 2B A9 58 90 00

6.3.3 Device Flash Operation

Basic Operation Loop:

Step 1 Connect SAM Reader

Step 2 Choose Operation Sector in Flash

--> 0xFF 00 80 00 02 00 02 (Choose the Second Sector)

Step 3 Send Flash Read/Write commands

Write data into the Page3, 16 bytes data "0x00 11 22 … FF" from address 0x00 as beginning in Flash.

--> 0xFF D6 03 00 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

<-- 0x90 00

Read data out from the Page3, 16 bytes data "00 11 22 … FF" from address 0x00 as beginning in Flash.

--> 0xFF B0 03 00 10

<-- 0x00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 90 00

