

F780 Dual Face Magnetic CardReader **[Generic Version]**

Manual & Specification

Revision Notes

NO.	Date	Description	Ver.	By
1	2016-03-10	Initial Version	1.0.1	Leung

Overview

F780 Dual face magnetic card reader is developed under 32bit Cortex-M3 platform with working frequency 72MHz. It can read and decode the magnetic card , credit card compatible with ISO7811 standard rapidly and accurately.

F780 works under RS232 interface or USB interface. With the powerful setting utility , users can set the output format flexibly and expediently. F780 can be used in shopping mall , supermarket ,CVS , as well as being set into Kiosk terminals such as ATM, self-service terminals.

1. Technical Features

- Power supply: DC 5V/±10%
- Power Mode:
 - 1) From RS232 of some terminals
 - 2) From USB port.
 - 3) From external power adapter.
- Electric Current: stand by status :≤80mA . Crest value under working: ≤100mA
- Interface :RS232 , USB
- Magnetic Card reading features:

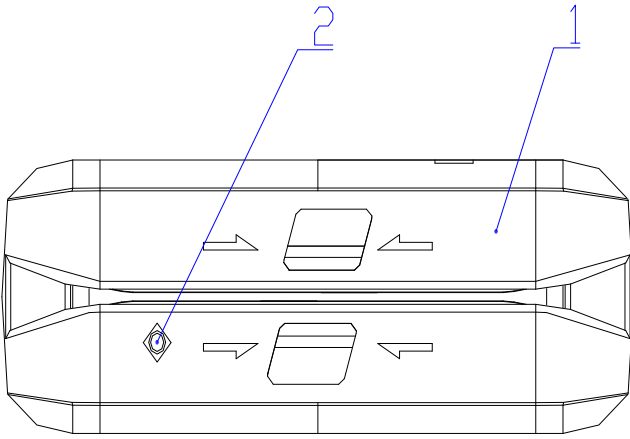
Support dual face swipe with two magnetic heads, support dual direction swipe.

Support ISO/IEC7811 standard.

Card swiping speed: 10cm/s~120cm/s.

Swiping lifetime: ≥600,000 passes.
- Working Environment: Temperature -10~50°C, Humidity 10%~90%.
- Support USB1.1/USB2.0 standard.
- Dimension (L×W×H): 116mm×36 mm×40mm

2. Structure Features



Pic 1

- 1. Main housing
- 2. Status LED (blue and red)

Picture:



3. DIP SWITCH Definition

There is a DIP Switch on the main board. The settings definition is following:

SW1	SW2	Interface Mode
ON	ON	USB Virtual COM Port
ON	OFF	USB HID Mode
OFF	ON	RS232 Mode (115200,8,N,1)
OFF	OFF	RS232 Mode (9600,8,N,1)

SW4	Status
ON	F/W updating mode
OFF	Working mode

SW3 has no definition.

4. Operation Manual

4.1. Communication Mode

Open the housing , there is a DIP switch on the main board, it controls the working mode of F780.

When SW1=ON:

If SW2=ON: Communication via USB virtual COM port, need install the USB-COM driver.

If SW2=OFF: Communication via USB, can be set to USB keyboard mode or HID mode by Setting utility.

When SW1=OFF:

Communication via RS232.

4.2. USB-HID Mode

Can set to USB keyboard or USB HID mode by setting utility on PC .

5. F780 Command system

Command format:

Sending Format:

Byte	1	2	3	4	5	Len + 1	Len + 2
Cmd	0x02	Len(H)	Len(L)	Cmd	Data	CHK	0x03

0x02 -----Start Character(1 byte)

Len(H)-----High-value of Length (1 byte)

Len(L)-----Low-value of Length (1 byte), Len= Cmd + the length of data

Cmd-----Byte of Command (1 byte)

Data-----Data transferred (For example: the card data, It will be null if no data transferred)

CHK-----CHKChecksum(1 byte), The exclusive or value of all bytes from Len(H) to data bytes.

0x03-----end character(1 byte)

Response Format:

Byte	1	2	3	4	Len -2	Len + 1	Len + 2
Response	0x02	Len(H)	Len(L)	Status	data	CHK	0x03

0x02 -----Start character (1 byte)

Len(H)-----High-value of Length (1 byte)

Len(L)-----Low-value of Length (1 byte) Len = Status+ the length of data

Status-----Status byte(1 byte)

data-----Data transferred (For example: the card data, It will be null if no data transferred)

CHK-----CHK Checksum (1 byte), The exclusive or value of all bytes from Len(H) to data bytes.

0x03-----end character(1 byte)

Command system:

5.1 Back to factory default settings

C:

Cmd	Data
0x0D	--

R:

Status	Data
0x00-Success	--
Others-failure	

Function: set back to factory default settings.

5.2 Write working parameters

C:

Cmd	Data
0x0E	StructWorkParam

R:

Status	Data
0x00-success	--
Others-failure	

Function: Write the data of working parameters to device.

For the definition of Working Parameter, please refer to Appendix A.

5.3 Read working parameter

C:

Cmd	Data
0x0F	

R:

Status	Data
0x00-success	StructWorkParam
others-failure	---

Function: read the working parameters from device.

For the definition of Working Parameter, please refer to Appendix A.

5.4 LED Control

C:

Cmd	Data
0x04	XY(1BYTE)

R:

Status	Data
--	--

Function: Control LED on/off

XY – 1byte LED status,

X –High 4 bytes: 0000–Control blue LED, 0001 –Control Red LED,

Y –Low 4 bytes: 0000 –off, 0001–on,

5.5 Buzzer Control

C:

Cmd	Data
0x05	[TimeH,TimeL]

R:

Status	Data
--	--

Function: Control the beep of Buzzer, the buzzer beeps time [TimeH,TimeL] then stops.

TimeH – the high byte of beeping time,

TimeL – the low byte of beeping time,

[TimeH,TimeL] means the beeping time, it is millisecond.

5.5 Get Ver. Number/UID Number

C:

Cmd	Data
0XF1	P1

R:

Status	Data
0x00-success	RespChar[]
other-failure	---

Function: Get the version number of firmware or UID number.

P1 – 1 byte parameter, when P1 = 0x00, get firmware version number.

when P1 = 0x01, get the device unique UID number.

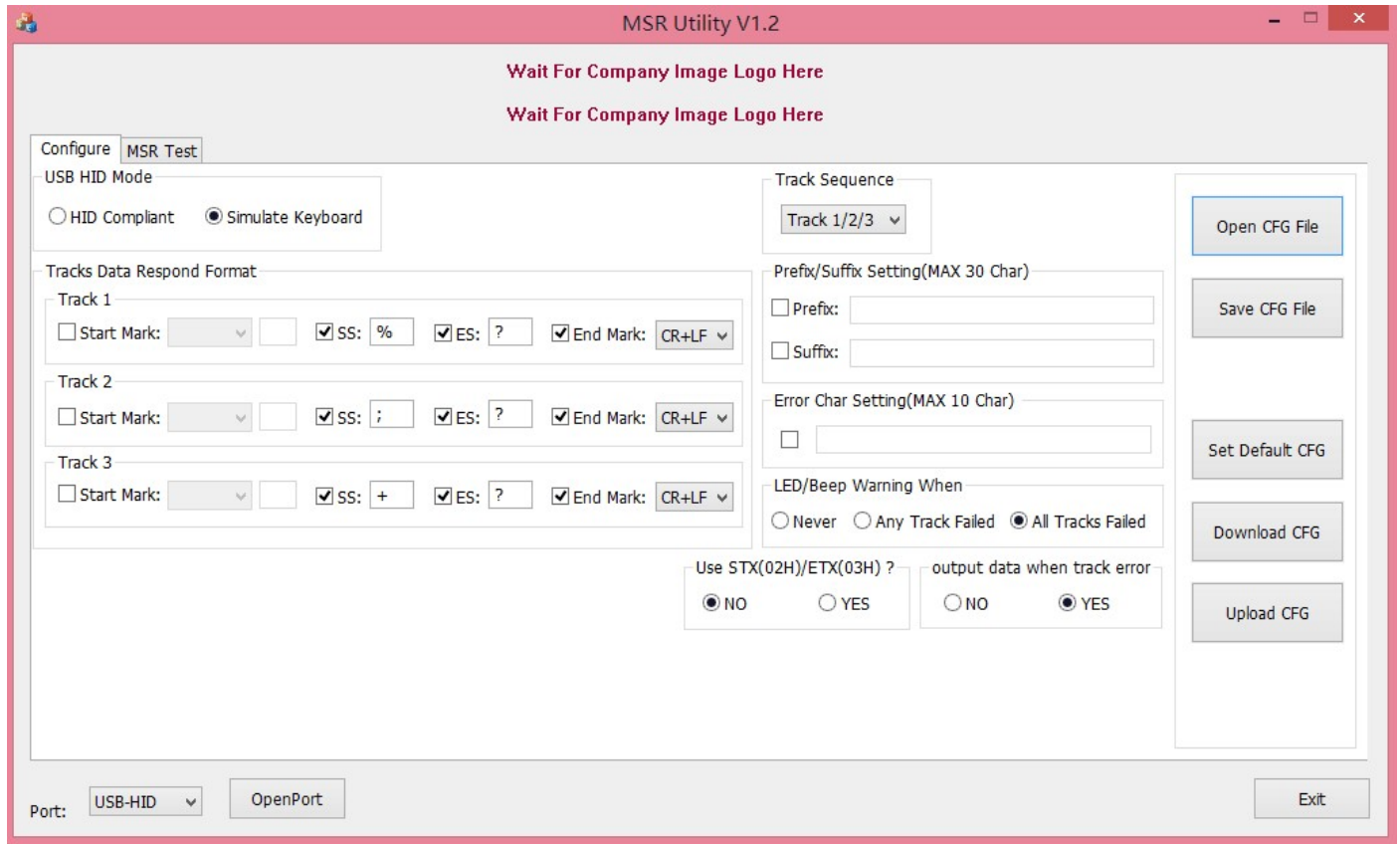
RespChar– Device firmware version number or UID number, It is ASCII Characters.

For examples : firmware such as "Neutral-V1.0.0.1-IAP"

UID such as "066FFF575351715087051731"

6 F780 setting utility specifications

The following is the window of the setting utility MSR Utility.exe , It supports the following settings:

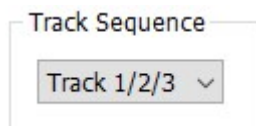


Pic 2 MSR Utility—F780 Setting utility



Pic 3

1. USBHID Mode: support USB HID Compliant mode and USB simulate Keyboard mode.



Pic 4

2. Select the output track sequence, support single/dual/triple tracks and the output sequence.

Tracks Data Respond Format

Track 1
 Start Mark: [dropdown] [input] SS: [%] ES: [?] End Mark: [CR+LF] [dropdown]

Track 2
 Start Mark: [dropdown] [input] SS: [;] ES: [?] End Mark: [CR+LF] [dropdown]

Track 3
 Start Mark: [dropdown] [input] SS: [+] ES: [?] End Mark: [CR+LF] [dropdown]

Pic 5

3. Set the output format of each track :

[Start control Chars] + [start mark] +track data + [End mark] + [End Separator]

The format the support characters of start control chars:

[Ctrol/Shift/Alt + chars]

[Ctrol + Shift + chars]

[F1~F12]

[ESC]

Start mark and end mark support all ASCII characters that can be displayed.

The characters and format of end separator:

CR + LF Enter + line feed

CR Enter

LF Line feed

TAB Tab key

SPACE space key

ESC Escap key

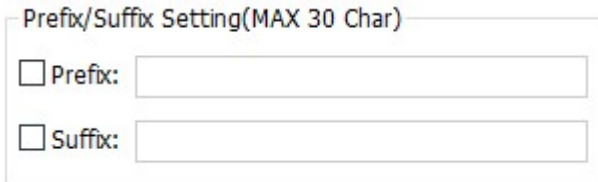
Use STX(02H)/ETX(03H) ?
 NO YES

Pic 6

4. Mark STX in front of the entire all tracks data , Mark ETX behind the entire all tracks data.

Eg: [STX] + track 1 data + track 2 data +track 3 data + [ETX]

“Track 1/2/3 data” means all the data got from the above point 4. Including"[Start control Chars] + [start mark] + track data + [End mark] + [End Separator]"



Pic 7

5. Mark prefix in front of the entire data , mark suffix behind the entire data , prefix and suffix support maximum 30 ASCII characters.

Eg:[Prefix] + [STX] + track 1 data + track 2 data + track 3 data + [ETX] + [Suffix]

“Track 1/2/3 data” means all the data got from the above point 4. Including "[Start control Chars] + [start mark] + track data + [End mark] + [End Separator]"

Summary: If all the characters from above points 4, 5,6 are set , the sequence of data output will be following:

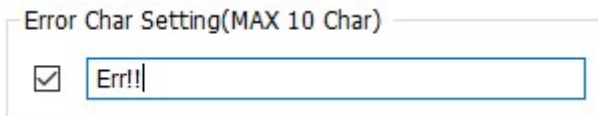
[Prefix] + [STX] +

[T1 start control chars] + [T1 start mark] + T1 data + [T1 end mark] + [T1 end separator] +

[T2 start control chars] + [T2 start mark] + T2 data + [T2 end mark] + [T2 end separator] +

[T3 start control chars] + [T3 start mark] + T3 data + [T3 end mark] + [T3 end separator] +

[ETX] + [Suffix]



Pic 8

6. When read tracks error, output the self-defined characters instead of the error tracks data. If this option is not set, the output error data is null .

Eg: As Pic 9, if read track1/2/3 error, output data: %Err!?!;Err!?!+Err!?!?



Pic 9

7. Set LED and Buzzer beep:

Never: No matter read card successfully or failed, the buzzer never beeps, the red LED never flashes.

Any Track Failed: Any track read failed, the buzzer beeps three sounds, the red LED is on. Otherwise when all set tracks read successfully, the buzzer beeps one sound ,the red LED does not flash.

All Tracks Failed: Only when all the set tracks read failed, the buzzer beeps three sounds, the red LED is on.

Otherwise if any of the set tracks read successfully , the buzzer beeps one sound, the red LED does not flash.

Summary: After all the parameters selected, Click “Download CFG” to download the setting to device.

Save CFG File

10. Save the current settings to .cfg file.

Open CFG File

11. Open the .cfg file to set the device.

Upload CFG

12. Read the current settings from device to setting utility.

Set Default CFG

13. set back to default factory settings.

Appendix A: Work parameters structure

//-----Work mode variable ,align at 4 bytes-----

```
typedef union _BITSTATUS
{
    struct _bitstatus
    {
        unsigned bOutSeq1           : 2;
        unsigned bOutSeq2           : 2;
        unsigned bOutSeq3           : 2;
        unsigned bLEDBeepReq        : 2;
        unsigned bDoubleTMode       : 1;
        unsigned bhSTXETX           : 1;
        unsigned bRev0              : 22;
    } bits;
    unsigned int value;
} BITSTATUS;
```

Bit Variable Instruction:

bOutSeq1 -the 1st output track, 0: no output, 1: output track 1, 2: output track 2, 3: output track 3.

bOutSeq2 -the 2nd output track, 0: no output, 1: output track 1, 2: output track 2, 3: output track 3.

bOutSeq3 -the 3rd output track, 0: no output, 1: output track 1, 2: output track 2, 3: output track 3.

bLEDBeepReq - Buzzer status 0: not beep, 1: beep three sounds if any track error,
2. Beep three sounds only when all tracks error.

bDoubleTModePosition of Magnetic head fixed, 0 -- tracks 1/2 fixed at the position of tracks 2/3,
1 -- Normal fixed position.

This setting is only for factory, users should not change this setting.

bhSTXETX Whether to add prefix (02) and suffix (03) .

typedef union WorkParam

```
{
    struct _workparam
    {
        BITSTATUS          BitStatus;           // Bit Variable

        unsigned char      T1Bit,T2Bit,T3Bit;   // BIT numbers coding
        unsigned char      T1CharSet,T2CharSet,T3CharSet; // character set coding
        unsigned char      T1SOM,T2SOM,T3SOM;   // Start mark coding
        unsigned char      T1EOM,T2EOM,T3EOM;   // end mark coding

        unsigned char      T1SOC,T2SOC,T3SOC;   // output start mark of tracks
        unsigned char      T1EOC,T2EOC,T3EOC;   // output end mark of tracks
        unsigned char      T3TrimSpace;         // reserved space in track 3      0xFF:
        reserve the original spaces, Others: reserve the spaces according to the value of T3RimSpace .
        unsigned char      HidMode;// the interface mode under HID ,0x21:HID compliant mode, 0x22: HID
        virtual keyboard mode.
    }
}
```



```

unsigned char    PrefixLen,Prefix[31];    // prefix string, maximum 31 bytes
unsigned char    SuffixLen,Suffix[31];    // suffix string, maximum 31 bytes

unsigned char    SSMark[3][MARKSIZE];    // start separator of each track
unsigned char    ESMark[3][MARKSIZE];    // end separator of each track

unsigned char    Customize;               // OEM customized mode
unsigned char    ErrorLen;                // length of data when error
unsigned char    ErrorStr[10];           // the data when read error, maximum 8 bytes.
}cell;
unsigned char collocate[sizeof(struct _workparam)];
}WORKPARAM;
extern WORKPARAM    WorkParam;

```

Appendix B:Coding sheet of character set

Chars	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ABA1	0	1	2	3	4	5	6	7	8	9	:	#	@	`	=	?
ABA2	0	1	2	3	4	5	6	7	8	9	:	;	<	=	=	?
ABA3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?