YFRF-200

Doc.	YFRF-200 Manual
Rev. N	1.12
Rev. D	04-22-2019

Rev	Summary	Write			Inspector		
		Pos	Name	Date	Pos	Name	Date
1.0	Initial		hh.choi	01.22.2019.		Bob.park	01.22.2019

내용

1.0 Introduction

The YFRF-200 is a PC-linked contactless smart card reader/writer used for accessing ISO 14443-4 Type A and Type B, MIFARE®. The YFRF-200 serves as the intermediary device between the computer and the contactless tag via the USB interface or RS-232. The reader carries out the command from the computer whether the command is used to communicate with a contactless tag, or control the device peripherals (LED or buzzer). This API document will discuss in detail how the commands were implemented for the contactless interface and device peripherals of the YFRF-200.

1.1 Features

- USB 2.0 Interface (Virtual Comport)
- or RS-223 or RS-232(TTL level)
- Smart Card Reader:

o Built-in antenna for contactless tag access, with card reading distance of up to 50 mm (depending on tag type)

o Support for ISO 14443 Part 4 Type A and B cards, MIFARE,

o Application Programming Interface:

o Custom Protocol : similar PC/SC

o Built-in Peripherals:

o User-controllable LED

o User-controllable buzzer

o Supports Android™ 5.0 and above

o Supports 2 SIM Socket

o Supports TMoney, Cashbee, and local cards (Can be developed by user request)

o Board Size : 50 x 100 mm including antenna Size : 50 x 60 mm

1.2 USB Interface

The YFRF-200 is connected to a computer through USB as specified in the USB Specification 2.0.

Pin	Signal	Function
1	V_{BUS}	+5 V power supply for the reader (Max. 200 mA, Normal 100 mA)
2	D-	Differential signal transmits data between YFRF-200 and PC
3	D+	Differential signal transmits data between YFRF-200 and PC
4	GND	Reference voltage level for power supply

Table 1: USB Interface

1.3 RS-232 Interface

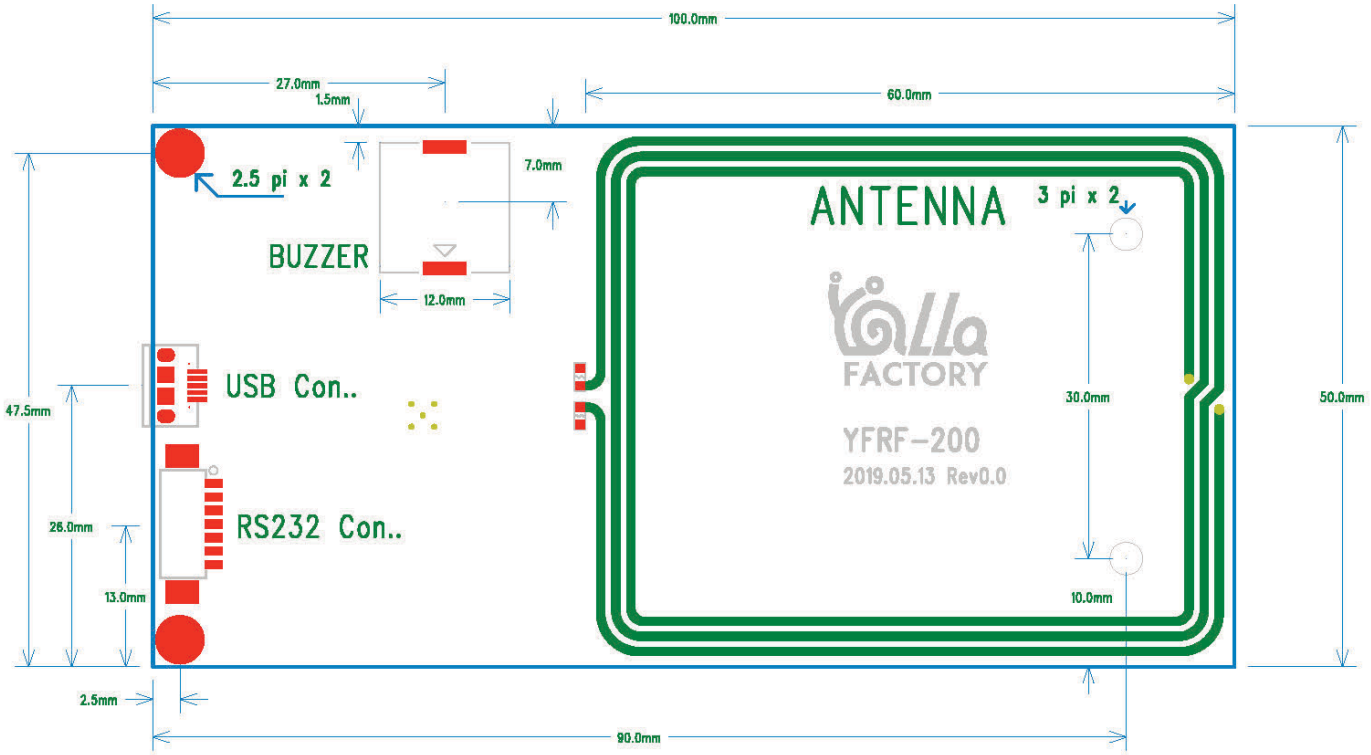
The YFRF-200 is connected to a computer through RS-232 with 115200bps

Pin	Signal	Function
1	V_{BUS}	+5 V power supply for the reader (Max. 200 mA, Normal 100 mA)
2	RX	TTL or RS-232 (Optional)
3	GND	Reference voltage level for power supply
4	TX	TTL or RS-232 (Optional)
5	GND	Reference voltage level for power supply
6	NC	
7	NC	

Table 2: RS-232 Interface

1.4 Dimensional Drawing

o Board Size : 50 x 100 mm including antenna Size : 48 x 60 mm



2.0. PICC Interface Description

2.1. ATR Generation

If the reader detects a PICC, an ATR will be sent to the PC/SC driver for identifying the PICC.

2.1.1. ATR format for ISO 14443 Part 3 PICCs

Byte	Value(Hex)	Designation	Description
0	3Bh	Initial Header	-
1	8Nh	T0	Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following. Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)
2	80h	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0
3	01h	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1
4 To 3+N	80h	T1	Category indicator byte, 80 means A status indicator may be present in an optional COMPACT-TLV data object
	4Fh	Tk	Application identifier Presence Indicator
	0Ch		Length
	RID		Registered Application Provider Identifier (RID) # A0 00 00 03 06h
	SS		Byte for standard
	C0 ... C1h		Bytes for card name
	00 00 00 00h	RFU	RFU # 00 00 00 00h
4+N	UUh	TCK	Exclusive-oring of all the bytes T0 to Tk

<**Table 3**: ATR format for ISO 14443 Part 3 PICCs>

Example:

ATR for MIFARE 1K = {3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6Ah}

ATR											
Initial Header	T0	TD1	TD2	T1	Tk	Length	RID	Standard	Card Name	RFU	TCK
3Bh	8Fh	80h	01h	80h	4Fh	0Ch	A0 00 00 03 06h	03h	00 01h	00 00 00 00h	6Ah

Where:

Length (YY) = 0Ch

RID = A0 00 00 03 06h (PC/SC Workgroup)

Standard (SS) = 03h (ISO 14443A, Part 3)

Card Name (C0 .. C1) = [00 01h] (MIFARE Classic® 1K)

Where, Card Name (C0 .. C1)

00 01h: MIFARE Classic 1K

00 02h: MIFARE Classic 4K

FFh [SAK]: Undefined

2.1.2. ATR format for ISO 14443 Part 4 PICCs

Byte	Value(Hex)	Designation	Description
0	3Bh	Initial Header	-
1	8Nh	T0	Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following. Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)
2	80h	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0
3	01h	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1
4 To 3+N	XXh	T1	Historical Bytes: ISO 14443A: The historical bytes from ATS response. Refer to the ISO14443-4 specification. ISO 14443B: The higher layer response from the ATTRIB response (ATQB). Refer to the ISO14443-3 specification.
	XXh XX XXh	Tkl	
4+N	UUh	TCK	Exclusive-oring of all the bytes T0 to Tk

<**Table 4**: ATR format for ISO 14443 Part 4 PICCs>

3.0. PICC Commands for General Purposes

3.1. Get Data

This command returns the serial number or ATS of the connected PICC.

Get UID APDU Format (5 bytes)

Command	Class	INS	P1	P2	Le
Get Data	FFH	CAh	00h 001	00h	00h (Full Length)

Get UID Response Format (UID + 2 bytes) if P1 = 00h

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

Get ATS of a ISO 14443 A card (ATS + 2 bytes) if P1 = 01h Response

Response	Data Out		
Result	ATS	SW1	SW2

Response Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully.
Error	63 00h	The operation failed.
Error	6A 81h	Function not supported.

Example:

1. To get the serial number of the connected PICC.
   UINT8 GET_UID[5]={FFh, CAh, 00h, 00h, 04h};
2. To get the ATS of the connected ISO 14443 A PICC.
   UINT8 GET_ATS[5]={FFh, CAh, 01h, 00h, 04h};

3.0. PICC Commands (T=CL Emulation) for MIFARE Classic Memory Cards

3.1. Load Authentication Keys

This command loads the authentication keys into the reader. The authentication keys are used to authenticate the particular sector of the MIFARE Classic 1K/4K memory card. Volatile authentication key location is provided.

Load Authentication Keys APDU Format (11 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Load Authentication	FFh	82h	Key Structure	Key Number	06h	Key (6 byte)

Where:

Key Structure 1 byte.
00h = Key is loaded into the reader volatile memory.
Other = Reserved.

Key Number 1 byte.
00h ~ 01h = Key Location. The keys will disappear once the reader is disconnected from the PC.

Key 6 bytes.
The key value loaded into the reader. e.g., {FF FF FF FF FF FFh}

Load Authentication Keys Response Format (2 Bytes)

Response	Data out	
Result	SW1	SW2

Response Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully
Error	63 00h	The operation failed.

Example: Load a key {FF FF FF FF FF FFh} into the key location 00h. APDU = {FF 82 00 00h 06 FF FF FF FF FF FFh}

3.2. Authentication

This command uses the keys stored in the reader to do authentication with the MIFARE 1K/4K card (PICC). Two types of authentication keys are used: TYPE_A and TYPE_B.

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

Command	Class	INS	P1	P2	P3	Data In
Authentication	FFh	88h	00h	Block Number	Key Type	Key Number

Load Authentication Keys APDU Format (10 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Authentication	FFh	86h	00h	00h	05h	Authenticate Data Bytes

Authenticate Data Bytes (5 bytes)

Byte1	Byte2	Byte3	Byte4	Byte5
Version 01h	00h	Block Number	Key Type	Key Number

Where:

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

Key Type 1 byte

60h = Key is used as a TYPE A key for authentication.

61h = Key is used as a TYPE B key for authentication.

Key Number 1 byte

00h ~ 01h = Key Location.

Note: For MIFARE Classic 1K Card, it has totally 16 sectors and each sector consists of 4 consecutive blocks. E.g. Sector 00h consists of Blocks {00h, 01h, 02h and 03h}; Sector 01h consists of Blocks {04h, 05h, 06h and 07h}; the last sector 0F consists of Blocks {3Ch, 3Dh, 3Eh and 3Fh}.

Once the authentication is done successfully, there is no need to do the authentication again if the blocks to be accessed belong to the same sector. Please refer to the MIFARE Classic 1K/4K specification for more details.

Load Authentication Keys Response Format (2 bytes)

Response	Data Out	
Result	SW1	SW2

Response Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully
Error	63 00h	The operation failed

Sectors (Total 16 sectors. Each sector consists of 4consecutive blocks)	Data Blocks (3 blocks, 16 bytes per block)	Trailer Block (1 block, 16 bytes)
Sector 0	00h ~ 02h	03h
Sector 1	04h ~ 06h	07h
..		
..		
Sector 14	38h ~ 0Ah	3Bh
Sector 15	3Ch ~ 3Eh	3Fh

1KB

Table 4: MIFARE 1K Memory Map

Sectors (Total 32 sectors. Each sector consists of 4 consecutive blocks)	Data Blocks (3 blocks, 16 bytes per block)	Trailer Block (1 block, 16bytes)	
Sector 0	00h ~ 02h	03h	2KB
Sector 1	04h ~ 06h	07h	
…			
…			
Sector 30	78h ~ 7Ah	7Bh	
Sector 31	7Ch ~ 7Eh	7Fh	
Sectors (Total 8 sectors. Each sector consists of 16 consecutive blocks)	**Data Blocks (15 blocks, 16 bytes per block)**	**Trailer Block (1 block, 16bytes)**	
Sector 32	80h ~ 8Eh	8Fh	2KB
Sector 33	90h ~ 9Eh	9Fh	
..			
..			
Sector 38	E0h ~ EEh	EFh	
Sector 39	F0h ~ FEh	FFh	

Table 5 : MIFARE Classic 4K Memory Map

Byte Number	0	1	2	3	Page
Serial Number	SN0	SN1	SN2	Bcc0	0
Serial Number	SN3	SN4	SN5	SN6	1
Internal/Lock	BCC1	Internal	Lock()	Lock1	2
OTP	OPT0	OPT1	OPT2	OPT3	3
Data read/write	Data0	Data1	Data2	Data3	4
Data read/write	Data4	Data5	Data6	Data7	5
Data read/write	Data8	Data9	Data10	Data11	6
Data read/write	Data12	Data13	Data14	Data15	7
Data read/write	Data16	Data017	Data18	Data19	8
Data read/write	Data20	Data21	Data22	Data23	9
Data read/write	Data24	Data25	Data26	Data27	10
Data read/write	Data28	Data29	Data30	Data31	11
Data read/write	Data32	Data33	Data34	Data35	12
Data read/write	Data36	Data37	Data38	Data39	13
Data read/write	Data40	Data41	Data42	Data43	14
Data read/write	Data44	Data45	Data46	Data47	15

512bits or 64 bytes

Table 6: MIFARE Ultralight Memory Map

Example:

1. To authenticate the Block 04h with a {TYPE A, key number 00h}. For PC/SC V2.01, Obsolete.
   APDU = {FF 88 00 04 60 00h};
2. To authenticate the Block 04h with a {TYPE A, key number 00h}. For PC/SC V2.07
   APDU = {FF 86 00 00 05 01 00 04 60 00h}

Note: MIFARE Ultralight does not need to do any authentication. The memory is free to access.

3.3. Read Binary Blocks

This command retrieves the data blocks from the PICC. The data block/trailer block must be authenticated first.

Read Binary APDU Format (5 bytes)

Command	Class	INS	P1	P2	Le
Read Binary Blocks	FFh	B0h	00h	Block Number	Number of Bytes to Read

Where:

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	0 <= N <= 16	SW1	SW2

Response Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully.
Error	63 00h	The operation failed.

Example:

1. Read 16 bytes from the binary block 04h (MIFARE Classic 1K or 4K)
   APDU = {FF B0 00 04 10h}
2. Read 4 bytes from the binary Page 04h (MIFARE Ultralight)
   APDU = {FF B0 00 04 04h}
3. Read 16 bytes starting from the binary Page 04h (MIFARE Ultralight) (Pages 4, 5, 6 and 7 will be read)
   APDU = {FF B0 00 04 10h}

Note: Please add a 2-second delay when reading NDEF messages in MIFARE Classic 4K cards.

3.4. Update Binary Blocks

This command writes data blocks into the PICC. The data block/trailer block must be authenticated.

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

Command	Class	INS	P1	P2	Lc	Data In
Update Binary Blocks	FFh	D6h	00h	Block Number	Number of Bytes to Update	Block Data 4Bytes for MIFARE Ultralight or 16 Bytes for MIFARE 1K/4K

Where:

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 Ultralight

Block Data 4 bytes or 16 bytes.
The data to be written into the binary block/blocks.

Response Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully.
Error	63 00h	The operation failed.

Example:

1. Update the binary block 04h of MIFARE Classic 1K/4K with Data {00 01 .. 0Fh}
   APDU = {FF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0Fh}
2. Update the binary block 04h of MIFARE Ultralight with Data {00 01 02 03}
   APDU = {FF D6 00 04 04 00 01 02 03h}

3.5. Value Block Related Commands

The data block can be used as value block for implementing value-based applications.

3.5.1. Value Block Operation

This command manipulates the value-based transactions (e.g., increment a value of the value block etc.)

Value Block Operation APDU Format (10 bytes)

Command	Class	INS	P1	P2	Lc	Data In	
Value Block Operation	FFh	D7h	00h	Block Number	05h	VB_OP	VB_Value (4bytes) {MSB…LSB}

Where:

Block Number 1 byte

The value block to be manipulated.

VB_OP 1 byte

00h = Store the VB_Value into the block. The block will then be converted to a value block. 01h = Increment the value of the value block by the VB_Value. This command is only valid for value block.

02h = 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 (4 bytes).

Example 1: Decimal –4 = {FFh, FFh, FFh, FCh}

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

Example 2: Decimal 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB			LSB
00h	00h	00h	01h

Value Block Operation Response Format (2 bytes)

Response	Data Out	
Result	SW1	Sw2

Response Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully
Error	63 00h	The operation failed.

3.5.2. Read Value Block

This command retrieves the value from the value block. This command is only valid for value block.

Read Value Block APDU Format (5 bytes)

Command	Class	INS	P1	P2	Le
Read Value Block	FFh	B1h	00h	Block Number	04h

Where:

Block Number 1 byte

The value block to be accessed.

Read Value Block Response Format (4 + 2 bytes)

Response	Data Out		
Result	Value {MSB ... LSB}	SW1	SW2

Where:

Value 4 bytes.

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

Example 1: Decimal –4 = {FFh, FFh, FFh, FCh}

Value			
MSB			LSB
FFH	FFH	FFh	FCh

Example 2: Decimal 1 = {00h, 00h, 00h, 01h}

Value			
MSB			LSB
00h	00h	00h	01h

Response Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully
Error	63 00h	The operation failed.

3.5.3. Restore Value Block

This command copies a value from a value block to another value block.

Restore Value Block APDU Format (7 bytes)

Command	Class	INS	P1	P2	Lc	Data In	
Restore Value Block	FFh	D7h	00h	Source Block Number	02h	03h	Target Blcok Number

Where:

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 Codes

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully.
Error	63 00h	The operation failed.

Example:

1. Store a value "1" into block 05h
   APDU = {FF D7 00 05 05 00 00 00 00 01h}
   Answer: 90 00h
2. Read the value block 05h
   APDU = {FF B1 00 05 00h}
   Answer: 00 00 00 01 90 00h [9000h]

3. Copy the value from value block 05h to value block 06h
   APDU = {FF D7 00 05 02 03 06h}
   Answer: 90 00h [9000h]
4. Increment the value block 05h by "5"
   APDU = {FF D7 00 05 05 01 00 00 00 05h}
   Answer: 90 00h [9000h]

4.0. Pseudo-APDU Commands

The pseudo-APDU commands are used for the following:

- Exchanging data with non-PC/SC–compliant tags
- Retrieving and setting the reader parameters
- Pseudo-APDUs can be sent through the "YFRF-200 PICC Interface" if the tag is already connected
- Pseudo-APDUs can be sent using "Escape Command" if the tag is not yet presented

4.1. Direct Transmit

This is the payload to be sent to the tag or reader.

Direct Transmit Command Format (Length of the Payload + 5 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Direct Transmit	FFh	00h	00h	00h	Number Of Bytes To send	Payload

Where:

Lc 1 byte.
Number of bytes to send
Maximum 255 bytes

Data In Response

Direct Transmit Response Format

Response	Data Out
Direct Transmit	Response Data

4.2. Bi-color LED and Buzzer Control

This command controls the states of the bi-color LED and Buzzer.

Bi-color LED and Buzzer Control Command Format (9 bytes)

Command	Class	INS	P1	P2	Lc	Data In (4bytes)
Bi-Color and Buzzer LED Control	FFh	00h	40h	LED State Control (Bit 7 --- Bit 0)	04h	Blinking Duration Control

Where:

P2 LED State Control

Bi-Color LED and Buzzer Control Format (1 byte)

CMD	Item	Description
Bit 0	Final State: Red LED	1 = On; 0 = Off
Bit 1	Final State: Green LED	1 = On; 0 = Off
Bit 2	State Mask: Red LED	1 = Update the State 0 = No change
Bit 3	State Mask: Green LED	1 = Update the State 0 = No change

Data Out SW1 SW2. Status Code returned by the reader.

Results	SW1	SW2	Meaning
Success	90h	Current LED State	The operation completed successfully.
Error	63h	00h	The operation failed.

Current LED State (1 byte)

Status	Item	Description
Bit 0	Current Red LED	1 = On; 0 = Off
Bit 1	Current Green LED	1 = On; 0 = Off
Bits 2 – 7	Reserved	

4.3. Get firmware version of the reader

This command retrieves the firmware version of the reader.

Command Format (5 bytes)

Command	Class	INS	P1	P2	Le

Get Firmware Version	FFh	00h	48h	00h	00h

Response Format (11 bytes)

Response	Data Out
Result	Firmware Version

Example:

Response = 59 46 52 46 32 30 30 5F 30 30 31h = YFRF200_001 (ASCII)

4.4. Get the PICC operating parameter

This command retrieves the PICC operating parameter of the reader.

Command Format (5 bytes)

Command	Class	INS	P1	P2	Le
Get PICC Operating Parameter	FFh	00h	50h	00h	00h

Response Format (2 bytes)

Response	Data Out	
Result	90h	PICC Operating Parameter

4.5. Set the PICC operating parameter This command sets the PICC operating parameter of the reader.

Command Format (5 bytes)

Command	Class	INS	P1	P2	Le
Set PICC Operating Parameter	FFh	00h	51h	New PICC Operating Parameter	00h

Response Format (2 bytes)

Response	Data Out	
Result	90h	PICC Operating Parameter

PICC Operating Parameter

Bit	Parameter	Description	Option
7	Auto PICC Polling	To enable the PICC Polling	1 = Enable 0 = Disable
6	Auto ATS Generation	To issue ATS Request whenever an ISO14443-4 Type A tag is activated	1 = Enable 0 = Disable
5	Polling Interval	To set the time interval between successive PICC Polling.	1 = 250ms 0 = 500ms
4		The Tag Types to be detected during PICC Polling.	
3			
2			
1	ISO 14443 Type B		1 = Detect 0 = Skip
0	ISO 14443 Type A #To detect the MIFARE Tags, the Auto ATS Generation must be disabled first.		1 = Detect 0 = Skip

Note: Default Value = FFh

4.6. Set Timeout Parameter

This command sets the timeout parameter of the contactless chip response time.

Command Format (5 bytes)

Command	Class	INS	P1	P2	Le
Set Timeout Parameter	FFh	00h	41h	Timeout Parameter (Unit: 5sec.)	00h

Where:

P2 Timeout Parameter

00h: No Timeout check

01h – FEh: Timeout with 5 second unit

FFh: Wait until the contactless chip responds

Response Format (2 bytes)

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully.
Error	63 00h	The operation failed.

4.7. Set buzzer output during card detection

This command sets the buzzer output during card detection. The default output is ON.

Command Format (5 bytes)

Command	Class	INS	P1	P2	Le
Set Buzzer Output during Card Detection	FFh	00h	52h	PollBuzzStatus	00h

Where:

P2 PollBuzzStatus

00h: Buzzer will NOT turn on when a card is detected

FFh: Buzzer will turn on when a card is detected

Response Format (2 bytes)

Results	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully
Error	63 00h	The operation failed.

5.0. Basic Program Flow for Contactless Applications

Step 0. Start the application. The reader will do the PICC Polling and scan for tags continuously. Once the tag is found and detected, the corresponding ATR will be sent to the PC. You must make sure that the PC/SC Escape Command has been set. See Appendix A for more details.

Step 1. The first thing is to connect the "YFRF-200 PICC Interface".

Step 2. Access the PICC by sending APDU commands.

:

:

Step N. Disconnect the "YFRF-200 PICC Interface". Shut down the application.

Notes:

1. The antenna can be switched off in order to save the power.
   - Turn off the antenna power: FF 00 00 00 04 D4 32 01 00h
   - Turn on the antenna power: FF 00 00 00 04 D4 32 01 01h
2. Standard and Non-Standard APDUs Handling.
   - PICCs that use Standard APDUs: ISO14443-4 Type A and B, MIFARE .. etc
   - PICCs that use Non-Standard APDUs: FeliCa, Topaz .. etc.
3. For the YFRF-200 PICC Interface, ISO 7816 T=1 protocol is used.
   - PC → Reader: Issue an APDU to the reader.
   - Reader → PC: The response data is returned.

5.1. How to access PC/SC-compliant tags (ISO 14443-4)?

Basically, all ISO 14443-4 compliant cards (PICCs) would understand the ISO 7816-4 APDUs. The YFRF-200 Reader just needs to communicate with the ISO 14443-4 compliant cards through exchanging ISO 7816-4 APDUs and Responses. YFRF-200 will handle the ISO 14443 Parts 1-4 Protocols internally.

MIFARE 1K, 4K, Mini and Ultralight tags are supported through the T=CL emulation. Simply treat the MIFARE tags as standard ISO 14443-4 tags. For more information, please refer to topic: **PICC Commands for MIFARE Classic Memory Tags.**

ISO 7816-4 APDU Format

Command	class	INS	P1	P2	Lc	Data In	Le
ISO 7816 Part 4 Command	-	-	-	-	Length Of the Data In	-	Expected Length of the Response Data

ISO 7816-4 Response Format (Data + 2bytes)

Response	Data Out		
Result	Response Data	SW1	SW2

Response Codes

Result	SW1 SW2	Meaning
Success	90 00h	The operation completed successfully.
Error	63 00h	The operation failed.

Typical sequence may be:

1. Present the Tag and Connect the PICC Interface
2. Read/Update the memory of the tag

1. Connect the Tag
2. Send an APDU, Get Challenge.

   << 00 84 00 00 08h

   >> 1A F7 F3 1B CD 2B A9 58h [90 00h]

Note: For ISO14443-4 Type A tags, the ATS can be obtained by using the APDU "FF CA 00 00 01h"