Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1251U USB NFC Reader with SAM Slot

Application Programming Interface V1.07

info@acs.com.hk
www.acs.com.hk

Table of Contents

List of Figures

List of Tables

1.0. Introduction

The ACR1251U is a PC-linked NFC smart card reader with SAM (Secure Access Module) slot developed based on the 13.56 MHz contactless technology. Following the ACR122U, ACS's successful NFC reader and also the world's first CCID-compliant contactless reader, the ACR1251U offers more advanced features. It is designed to support not only ISO 14443 Type A and B cards, but also MIFARE®, FeliCa and all four types of NFC tags and devices.

ACR1251U acts as the intermediary device between the computer and the card. The reader, which specifically communicates with the contactless tag, SAM card or the device peripherals (LED or buzzer), will carry out a command issued from the computer. It has two reader interfaces, namely the PICC and SAM interface, and both interface follow the PC/SC specifications. This API document will discuss in detail how the PC/SC APDU commands were implemented for the contactless interface and device peripherals of ACR1251U.

2.0. Features

- USB 2.0 Full Speed Interface
- CCID Compliance
- Smart Card Reader:
  - Read/Write speed of up to 424 Kbps
  - Built-in antenna for contactless tag access, with card reading distance of up to 50 mm (depending on tag type)
  - Support for ISO 14443 Part 4 Type A and B cards, MIFARE, FeliCa, and all four types of NFC (ISO/IEC 18092 tags)
  - Built-in anti-collision feature (only one tag is accessed at any time)
  - NFC Support:
    - Card reader/writer mode
    - Peer-to-Peer mode
  - ISO 7816-compliant SAM slot
- Application Programming Interface:
  - Supports PC/SC
  - Supports CT-API (through wrapper on top of PC/SC)
- Built-in Peripherals:
  - User-controllable bi-color LED
  - User-controllable buzzer
- USB Firmware Upgradability
- Supports Android™ 3.1 and above
- Compliant with the following standards:
  - ISO 18092
  - ISO 14443
  - ISO 7816
  - LASCOM
  - CE
  - FCC
  - VCCI
  - MIC
  - KC
  - PC/SC
  - CCID
  - Microsoft® WHQL
  - RoHS 2
  - REACH

3.0. Acronyms and Abbreviations

Acronym/Abbreviation	Description
ATR	Attribute Request and Attribute Response
DEP	Data Exchange Protocol Request and Data Exchange Protocol Response
DSL	Deselect Request and Deselect Response
PSL	Parameter Selection Request and Parameter Selection Response
RLS	Release Request and Release Response
WUP	Wakeup Request and Wakeup Response
DID	Device ID
BS	Sending bit duration
BR	Receiving bit duration
PP	Protocol Parameters
Gi	Optional information field for Initiator
PFB	Control information for transaction
FSL	maximum value for the Frame Length
LLCP	Logical Link Control Protocol

Table 1: Acronyms and Abbreviations

4.0. Architecture

For communication architecture, the protocol used between ACR1251U reader and the computer is CCID protocol. All communications between PICC and SAM are PCSC-compliant.

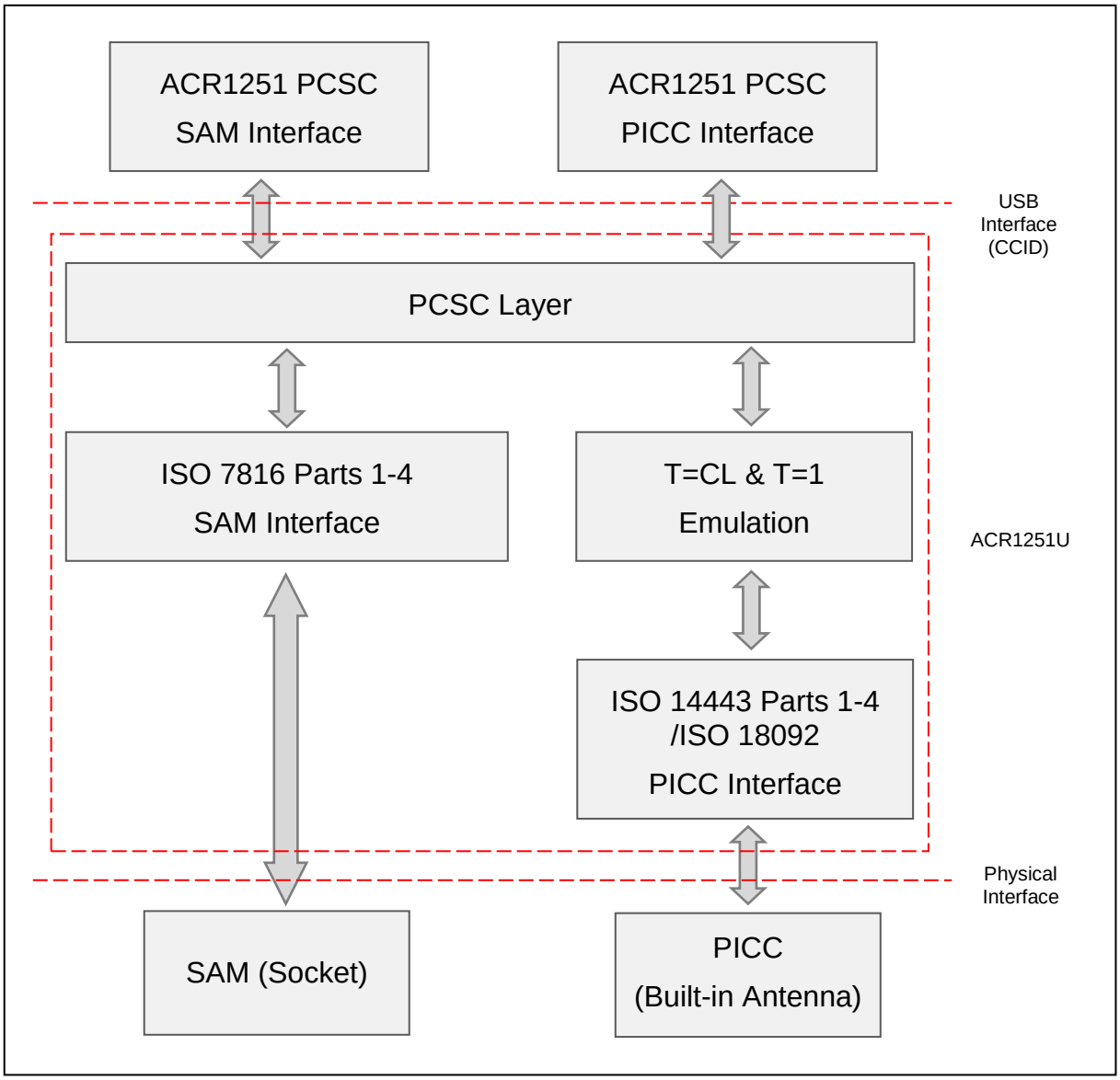

Figure 1: ACR1251U Architecture

5.0. Host Programming (PC-linked) API

5.1. PCSC API

This section will describe some of the PCSC API commands for application programming usage. For more details, please refer to Microsoft MSDN Library or PCSC workgroup.

5.1.1. SCardEstablishContext

The **SCardEstablishContext** function establishes the resource manager context within which database operations are performed.

Refer to: http://msdn.microsoft.com/en-us/library/windows/desktop/aa379479%28v=vs.85%29.aspx

5.1.2. SCardListReaders

The **SCardListReaders** function provides the list of readers within a set of named reader groups, eliminating duplicates.

The caller supplies a list of reader groups, and receives the list of readers within the named groups. Unrecognized group names are ignored. This function only returns readers within the named groups that are currently attached to the system and available for use.

Refer to: http://msdn.microsoft.com/en-us/library/windows/desktop/aa379793%28v=vs.85%29.aspx

5.1.3. SCardConnect

The **SCardConnect** function establishes a connection (using a specific resource manager context) between the calling application and a smart card contained by a specific reader. If no card exists in the specified reader, an error is returned.

Refer to: http://msdn.microsoft.com/en-us/library/windows/desktop/aa379473%28v=vs.85%29.aspx

5.1.4. SCardControl

The **SCardControl** function gives you direct control of the reader. You can call it any time after a successful call to **SCardConnect** and before a successful call to **SCardDisconnect**. The effect on the state of the reader depends on the control code.

Refer to: http://msdn.microsoft.com/en-us/library/windows/desktop/aa379474%28v=vs.85%29.aspx

Note: *Commands from* ***Section 5.6 – Peripherals Control*** *are using this API for sending.*

5.1.5. ScardTransmit

The **SCardTransmit** function sends a service request to the smart card and expects to receive data back from the card.

Refer: http://msdn.microsoft.com/en-us/library/windows/desktop/aa379804%28v=vs.85%29.aspx

Note: *APDU Commands (i.e. the command sent to connected card and* ***Section 5.3 – PICC Commands****) are using this API for sending.*

5.1.6. ScardDisconnect

The **SCardDisconnect** function terminates a connection previously opened between the calling application and a smart card in the target reader.

Refer to: http://msdn.microsoft.com/en-us/library/windows/desktop/aa379475%28v=vs.85%29.aspx

5.1.7. APDU Flow

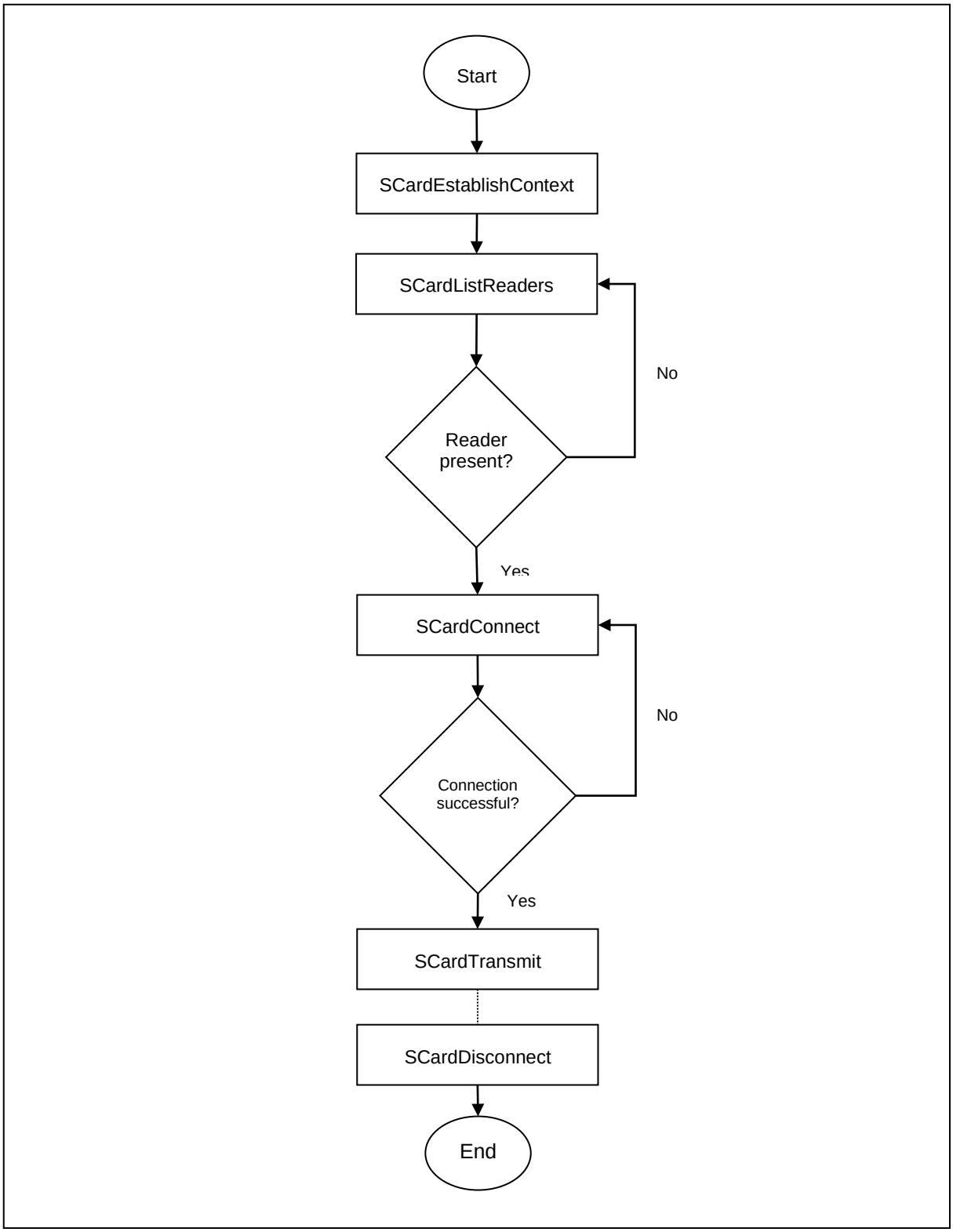

Figure 2: ACR1251U APDU Flow

5.1.8. Escape Command Flow

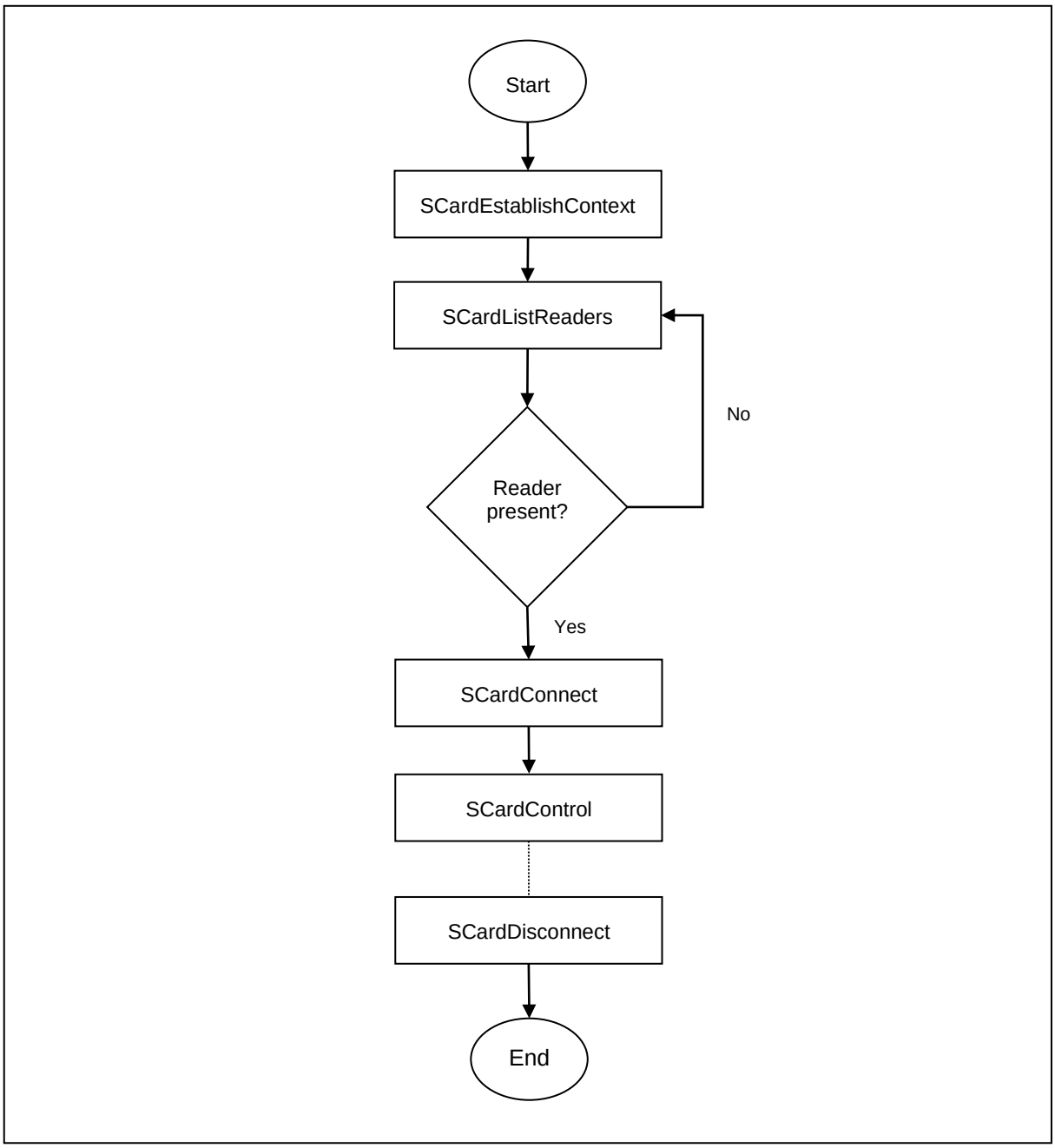

Figure 3: ACR1251U Escape Command Flow

5.2. Contactless Smart Card Protocol

5.2.1. ATR Generation

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

5.2.1.1. ATR Format for ISO 14443 Part 3 PICCs

Byte	Value	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 06
	SS		Byte for standard.
	C0 .. C1h		Bytes for card name.
	00 00 00 00h	RFU	RFU # 00 00 00 00
4+N	UU	TCK	Exclusive-oring of all the bytes T0 to Tk

Example:

ATR for MIFARE® Classic 1K = {3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 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)

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

= 11h: FeliCa

Card Name (C0 .. C1)	00 01: MIFARE Classic 1K	00 30: Topaz and Jewel
	00 02: MIFARE Classic 4K	00 3B: FeliCa
	00 03: MIFARE Ultralight®	FF 28: JCOP 30
	00 26: MIFARE Mini	FF [SAK]: undefined tags

5.2.1.2. ATR Format for ISO 14443 Part 4 PICCs

<table>
<tr><th>Byte</th><th>Value</th><th>Designation</th><th>Description</th></tr>
<tr><td>0</td><td>3Bh</td><td>Initial Header</td><td></td></tr>
<tr><td>1</td><td>8N</td><td>T0</td><td>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)</td></tr>
<tr><td>2</td><td>80h</td><td>TD1</td><td>Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following.
Lower nibble 0 means T = 0</td></tr>
<tr><td>3</td><td>01h</td><td>TD2</td><td>Higher nibble 0 means no TA3, TB3, TC3, TD3 following.
Lower nibble 1 means T = 1</td></tr>
<tr><td rowspan="2">4
to
3 + N</td><td>XX</td><td>T1</td><td rowspan="2">Historical Bytes:

ISO 14443-A:
The historical bytes from ATS response. Refer to the ISO 14443-4 specification.

ISO 14443-B:<table><tr><th>Byte1-4</th><th>Byte5-7</th><th>Byte8</th></tr><tr><td>Application Data from ATQB</td><td>Protocol Info Byte from ATQB</td><td>Higher nibble=MBLI from ATTRIB command
Lower nibble (RFU)=0</td></tr></table></td></tr>
<tr><td>XX
XX
XX</td><td>Tk</td></tr>
<tr><td>4+N</td><td>UU</td><td>TCK</td><td>Exclusive-oring of all the bytes T0 to Tk</td></tr>
</table>

Example 1: ATR for MIFARE® DESFire® = {3B 81 80 01 80 80h} // 6 bytes of ATR

Note: *Use the APDU "FF CA 01 00 00h" to distinguish the ISO 14443A-4 and ISO 14443B-4 PICCs, and retrieve the full ATS if available. ISO 14443A-3 or ISO 14443B-3/4 PICCs do have ATS returned.*

APDU Command = FF CA 01 00 00h

APDU Response = 06 75 77 81 02 80 90 00h

ATS = {06 75 77 81 02 80h}

Example 2: ATR for EZ-link = {3B 88 80 01 1C 2D 94 11 F7 71 85 00 BEh}

Application Data of ATQB = 1C 2D 94 11h

Protocol Information of ATQB = F7 71 85h

MBLI of ATTRIB = 00h

5.3. Pseudo APDU for Contactless Interface

5.3.1. 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 01h	00h	00h (Max Length)

If P1 = 00h, Get UID Response Format (UID + 2 Bytes)

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

If P1 = 01h, Get ATS of a ISO 14443 A card (ATS + 2 Bytes)

Response	Data Out		
Result	ATS	SW1	SW2

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation is completed successfully.
Warning	62h	82h	End of UID/ATS reached before Le bytes (Le is greater than UID Length).
Error	6Ch	**XX**h	Wrong length (wrong number Le: ‘**XX**’ encodes the exact number) if Le is less than the available UID length.
Error	63h	00h	The operation is failed.
Error	6Ah	81h	Function not supported

Examples:

To get the serial number of the “connected PICC”:

```
UINT8 GET_UID[5] = {FF, CA, 00, 00, 00};
```

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

```
UINT8 GET_ATS[5] = {FF, CA, 01, 00, 00};
```

5.4. PICC Commands (T=CL Emulation) for MIFARE® Classic (1K/4K) memory cards

5.4.1. Load Authentication Keys

This command loads the authentication keys to the reader. The authentication keys are used to authenticate the particular sector of the MIFARE Classic (1K/4K) memory card. Two kinds of authentication key locations are provided: volatile and non-volatile key locations.

Load Authentication Keys APDU Format (11 bytes)

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

Where:

Key Structure 1 byte.

00h = Key is loaded into the reader volatile memory.

Other = Reserved.

Key Number 1 byte.

00h – 01h = Non-volatile memory for storing keys. The keys are permanently stored in the reader and will be retained in the reader's memory even if the reader is disconnected from the PC. It can store up to 32 keys inside the reader non-volatile memory.

Note: *The default value is FF FF FF FF FF FFh.*

Key 6 bytes.

The key value loaded into the reader. Example: FF FF FF FF FF FFh

Load Authentication Keys Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Load Authentication Keys Response Codes

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

Example:

// Load a key {FF FF FF FF FF FFh} into the volatile memory location 00h.

APDU = {FF 82 00 00 06 FF FF FF FF FF FFh}

5.4.2. Authentication for MIFARE® Classic (1K/4K)

This command uses the keys stored in the reader to do authentication with the MIFARE Classic (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	Byte 2	Byte 3	Byte 4	Byte 5
Version 01h	00h	Block Number	Key Type	Key Number

Where:

Block Number 1 byte. The memory block to be authenticated.

For MIFARE Classic 1K card, it has totally 16 sectors and each sector consists of four 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 0Fh consists of blocks {3Ch, 3Dh, 3Eh and 3Fh}. Once the authentication is done successfully, there is no need to do the authentication again provided that the blocks to be accessed are belonging to the same sector. Please refer to the MIFARE Classic (1K/4K) specification for more details.

Note: *Once the block is authenticated successfully, all the blocks belonging to the same sector are accessible.*

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.

00 ~ 01h = Volatile memory for storing keys. The keys will disappear when the reader is disconnected from the PC. Two volatile keys are provided. The volatile key can be used as a session key for different sessions.

Load Authentication Keys Response Format (2 bytes)

Response	Data Out	
Result	SW1	SW2

Load Authentication Keys Response Codes

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

Examples:

//Authenticate the Block 04h with a {TYPE A, key number 00h}. For PC/SC V2.01, Obsolete.

APDU = {FF 88 00 04 60 00h};

//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}

Sectors (Total 16 sectors. Each sector consists of 4 consecutive 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

1K bytes

Table 2: MIFARE® Classic 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, 16 bytes)
Sector 0	00h – 02h	03h
Sector 1	04h – 06h	07h
..	..	..
..	..	..
Sector 30	78h – 7Ah	7Bh
Sector 31	7Ch – 7Eh	7Fh
Sector 32	80h – 8Eh	8Fh
Sector 33	90h – 9Eh	9Fh
..	..	..
..	..	..
Sector 38	E0h – EEh	EFh
Sector 39	F0h – FEh	FFh

4K bytes

Table 3: MIFARE® Classic 4K Memory Map

Examples:

// To authenticate the Block 04h with a {TYPE A, key number 00h}.

// PC/SC V2.01, Obsolete

APDU = FF 88 00 04 60 00h

// To authenticate the Block 04h with a {TYPE A, key number 00h}.

// PC/SC V2.07

APDU = FF 86 00 00 05 01 00 04 60 00h

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	Lock0	Lock1	2
OTP	OPT0	OPT1	OTP2	OTP3	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	Data17	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

512 bits or 64 bytes

Table 4: MIFARE Ultralight® Memory Map

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

5.4.3. Read Binary Blocks

This command is used for retrieving a multiple of "data blocks" from the PICC. The data block/trailer block must be authenticated first before executing this command.

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 starting block.

Number of Bytes to Read 1 byte.

Multiple of 16 bytes for MIFARE Classic (1K/4K) or Multiple of 4 bytes for MIFARE Ultralight

Maximum of 16 bytes for MIFARE Ultralight.

Maximum of 48 bytes for MIFARE Classic 1K. (Multiple Blocks Mode; 3 consecutive blocks)

Maximum of 240 bytes for MIFARE Classic 4K. (Multiple Blocks Mode; 15 consecutive blocks)

Example 1: 10h (16 bytes). The starting block only. (Single Block Mode)

Example 2: 40h (64 bytes). From the starting block to starting block+3. (Multiple Blocks Mode)

Note: *For security reasons, the Multiple Block Mode is used for accessing Data Blocks only. The Trailer Block is not supposed to be accessed in Multiple Blocks Mode. Please use Single Block Mode to access the Trailer Block.*

Read Binary Block Response Format (Multiply of 4/16 + 2 bytes)

Response	Data Out		
Result	Data (Multiple of 4/16 Bytes)	SW1	SW2

Read Binary Block Response Codes

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

Examples:

// Read 16 bytes from the binary block 04h (MIFARE Classic 1K/ 4K)

APDU = FF B0 00 04 10h

// Read 240 bytes starting from the binary block 80h (MIFARE Classic 4K)

// Block 80h to Block 8Eh (15 blocks)

APDU = FF B0 00 80 F0h

5.4.4. Update Binary Blocks

This command is used for writing a multiple of “data blocks” into the PICC. The data block/trailer block must be authenticated first before executing this command.

Update Binary APDU Format (Multiple of 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 (Multiple of 16 Bytes)

Where:

Block Number — 1 byte. The starting block to be updated.

Number of bytes to update — 1 byte.

- Multiply of 16 bytes for MIFARE Classic (1K/4K) or 4 bytes for MIFARE Ultralight.
- Maximum 48 bytes for MIFARE Classic 1K. (Multiple Blocks Mode; 3 consecutive blocks)
- Maximum 240 bytes for MIFARE Classic 4K. (Multiple Blocks Mode; 15 consecutive blocks)

Block Data — Multiple of 16 + 2 Bytes, or 6 bytes. The data to be written into the binary block/blocks.

Example 1: 10h (16 bytes). The starting block only. (Single Block Mode)

Example 2: 30h (48 bytes). From the starting block to starting block +2. (Multiple Blocks Mode)

Note: *For safety reasons, the Multiple Block Mode is used for accessing data blocks only. The Trailer Block is not supposed to be accessed in Multiple Blocks Mode. Please use Single Block Mode to access the Trailer Block.*

Update Binary Block Response Codes (2 bytes)

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

Examples:

// 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}

// Update the binary block 04h of MIFARE Ultralight with Data {00 01 02 03h}

APDU = {FF D6 00 04 04 00 01 02 03h}

5.4.5. Value Block Operation (INC, DEC, STORE)

This command is used for manipulating value-based transactions (e.g., increment a value of the value block).

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 (4 Bytes) {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

Value Block Operation Response Codes

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

5.4.6. 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	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

Read Value Block Response Codes

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

5.4.7. Copy Value Block

This command is used for copying a value from a value block to another value block.

Copy Value Block APDU Format (7 bytes)

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

Copy Value Block Response Format (2 bytes)

Response	Data Out	
Result	SW1	SW2

Copy Value Block Response Codes

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

Examples:

// Store a value "1" into block 05h

APDU = {FF D7 00 05 05 00 00 00 00 01h}

// Read the value block 05h

APDU = {FF B1 00 05 04h}

// Copy the value from value block 05h to value block 06h

APDU = {FF D7 00 05 02 03 06h}

// Increment the value block 05h by "5"

APDU = {FF D7 00 05 05 01 00 00 00 05h}

5.5. Accessing PCSC-compliant tags (ISO 14443-4)

Basically, all ISO 14443-4 compliant cards (PICCs) understands the ISO 7816-4 APDUs. The ACR1251U reader just has to communicate with the ISO 14443-4 compliant cards through exchanging ISO 7816-4 APDUs and responses. ACR1251U will handle the ISO 14443 Parts 1-4 Protocols internally.

MIFARE Classic (1K/4K), MIFARE Mini and MIFARE Ultralight tags are supported through the T=CL emulation. Just simply treat the MIFARE tags as standard ISO 14443-4 tags. For more information, please refer to **Section 5.3**.

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 + 2 bytes)

Response	Data Out		
Result	Response Data	SW1	SW2

Common ISO 7816-4 Response Codes

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

Typical sequence may be:

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

To do this:

1. Connect the tag.

   The ATR of the tag is 3B 88 80 01 00 00 00 00 33 81 81 00 3Ah.

   In which,

   The Application Data of ATQB = 00 00 00 00, protocol information of ATQB = 33 81 81. It is an ISO 14443-4 Type B tag.

2. Send an APDU, Get Challenge.

   << 00 84 00 00 08h

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

   Note: *For ISO 14443-4 Type A tags, the ATS can be obtained by using the APDU "FF CA 01 00 00h."*

Example:

// Read 8 bytes from an ISO 14443-4 Type B PICC (ST19XR08E)

APDU = {80 B2 80 00 08h}

Class = 80h

INS = B2h

P1 = 80h

P2 = 00h

Lc = None

Data In = None

Le = 08h

Answer: 00 01 02 03 04 05 06 07h [$9000h]

5.6. Accessing FeliCa tags

For FeliCa access, the command is different with PCSC-compliant tags and MIFARE. The command follows the FeliCa specification with an added header.

FeliCa Command Format

Command	Class	INS	P1	P2	Lc	Data In
FeliCa Command	FFh	00h	00h	00h	Length of the Data In	FeliCa Command (start with Length Byte)

FeliCa Response Format (Data + 2 bytes)

Response	Data Out
Result	Response Data

Read Memory Block Example:

1. Connect the FeliCa.

   The ATR = 3B 8F 80 01 80 4F 0C A0 00 00 03 06 11 00 3B 00 00 00 00 42h

   In which, 11 00 3Bh = FeliCa

2. Read FeliCa IDM.

   CMD = FF CA 00 00 00h

   RES = [IDM (8bytes)] 90 00h

   e.g., FeliCa IDM = 01 01 06 01 CB 09 57 03h

3. FeliCa command access.

   Example: “Read” Memory Block.

   CMD = FF 00 00 00 10 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

   where:

   Felica Command = 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

   IDM = 01 01 06 01 CB 09 57 03h

   RES = Memory Block Data

5.7. Peripherals Control

The reader's peripherals control commands are implemented by using *PC_to_RDR_Escape*.

5.7.1. Get Firmware Version

This command is used for getting the reader's firmware message.

Get Firmware Version Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Get Firmware Version	E0h	00h	00h	18h	00h

Get Firmware Version Response Format (5 bytes + Firmware Message Length)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	Number of bytes to receive	Firmware Version

Example:

Response = E1 00 00 00 0F 41 43 52 31 32 35 31 55 5F 56 32 30 34 2E 30

Firmware Version (HEX) = 41 43 52 31 32 35 31 55 5F 56 32 30 34 2E 30

Firmware Version (ASCII) = "ACR1251U_V204.0"

5.7.2. LED Control

This command is used for controlling the LED's output.

LED Control Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
LED Control	E0h	00h	00h	29h	01h	LED Status

LED Control Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	LED Status

LED Status (1 byte)

LED Status	Description	Description
Bit 0	RED LED	1 = ON; 0 = OFF
Bit 1	GREEN LED	1 = ON; 0 = OFF
Bit 2 - 7	RFU	RFU

5.7.3. LED Status

This command is used for checking the existing LED's status.

LED Status Format (5 bytes)

Command	Class	INS	P1	P2	Lc
LED Status	E0h	00h	00h	29h	00h

LED Status Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	LED Status

LED Status (1 byte)

LED Status	Description	Description
Bit 0	RED LED	1 = ON; 0 = OFF
Bit 1	GREEN LED	1 = ON; 0 = OFF
Bit 2 - 7	RFU	RFU

5.7.4. Buzzer Control

This command is used for controlling the buzzer output.

Buzzer Control Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Buzzer Control	E0h	00h	00h	28h	01h	Buzzer On Duration

Where:

Buzzer On Duration 1 byte.
00h = Turn OFF
01 to FFh = Duration (unit: 10 ms)

Buzzer Control Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	00h

5.7.5. Buzzer Status

This command is used for checking the existing buzzer status.

Buzzer Status Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Buzzer Status	E0h	00h	00h	28h	00h

Buzzer Status Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	00h

5.7.6. Set LED and Buzzer Status Indicator Behavior

This command is used for setting the behaviors of LEDs and buzzer as status indicators. The command format in this section is applicable to firmware version 205 and below only.

Note: *The setting will be saved into non-volatile memory.*

Set LED and Buzzer Status Indicator Behavior Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set LED and Buzzer Status Indicator Behavior	E0h	00h	00h	21h	01h	Behavior

Behavior (1 byte)

Behavior	MODE	Description
Bit 0	SAM Activation Status LED	To show the activation status of the SAM interface. 1 = Enable; 0 =Disable
Bit 1	PICC Polling Status LED	To show the PICC polling status. 1 = Enable; 0 =Disable
Bit 2	PICC Activation Status LED	To show the activation status of the PICC interface. 1 = Enable; 0 =Disable
Bit 3	Card Insertion and Removal Events Buzzer	To make a beep whenever a card insertion or removal event is detected. (For PICC). 1 = Enable; 0 =Disabled
Bit 4 – 6	RFU	RFU
Bit 7	Card Operation Blinking LED	To blink the LED whenever the card (PICC) is being accessed.

Note: *Default value of behavior = 8Fh.*

Set LED and Buzzer Status Indicator Behavior Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	Default Behaviors

5.7.7. Read LED and Buzzer Status Indicator Behavior

This command is used for reading the current default behaviors of LEDs and buzzer. The command format in this section is applicable to firmware version 205 and below only.

Read LED and Buzzer Status Indicator Behavior Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read LED and Buzzer Status Indicator Behavior	E0h	00h	00h	21h	00h

Read LED and Buzzer Status Indicator Behavior Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	Behaviors

Behavior (1 byte)

Behavior	MODE	Description
Bit 0	SAM Activation Status LED	To show the activation status of the SAM interface. 1 = Enable; 0 =Disable
Bit 1	PICC Polling Status LED	To show the PICC polling status. 1 = Enable; 0 =Disable
Bit 2	PICC Activation Status LED	To show the activation status of the PICC interface. 1 = Enable; 0 =Disable
Bit 3	Card Insertion and Removal Events Buzzer	To make a beep whenever a card insertion or removal event is detected. (For PICC) 1 = Enable; 0 =Disabled
Bit 4 – 6	RFU	RFU
Bit 7	Card Operation Blinking LED	To make the LED blink whenever the card (PICC) is being accessed.

Note: *Default value of Behavior = 8Fh.*

5.7.8. Set LED and Buzzer Status Indicator Behavior for PICC interface

This command is used for setting the behaviors of LEDs and buzzer as status indicators for PICC interface. The command format in this section is applicable to firmware version 206 and above only.

Note: *The setting will be saved into non-volatile memory.*

Set LED and Buzzer Status Indicator Behavior for PICC interface Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set LED and Buzzer Status Indicator Behavior for PICC interface	E0h	00h	00h	21h	01h	Behavior

Behavior (1 byte)

Behavior	MODE	Description
Bit 0	Card Operation Blinking LED	To blink the LED whenever the PICC card is being accessed. 1 = Enable; 0 =Disable
Bit 1	PICC Polling Status LED	To show the PICC polling status. 1 = Enable; 0 =Disable
Bit 2	PICC Activation Status LED	To show the activation status of the PICC interface. 1 = Enable; 0 =Disable
Bit 3	Card Insertion and Removal Events Buzzer	To make a beep whenever a card insertion or removal event is detected. (For PICC). 1 = Enable; 0 =Disabled
Bit 4	RFU	RFU
Bit 5	PN512 Reset Indication Buzzer	To make a beep when the PN512 is reset. 1 = Enable; 0 =Disabled
Bit 6	Color Select (GREEN)	Green LED for status change 1 = Enable; 0 =Disabled
Bit 7	Color Select (RED)	RED LED for status change 1 = Enable; 0 =Disabled

Note: *Default value of behavior = 7Fh.*

Set LED and Buzzer Status Indicator Behaviors for PICC interface Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	Default Behaviors

5.7.9. Read LED and Buzzer Status Indicator Behavior for PICC interface

This command is used for reading the current default behaviors of LEDs and buzzer for PICC interface. The command format in this section is applicable to Firmware version 206 and above only.

Read LED and Buzzer Status Indicator Behavior for PICC interface Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read LED and Buzzer Status Indicator Behavior for PICC interface	E0h	00h	00h	21h	00h

Read LED and Buzzer Status Indicator Behavior for PICC interface Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	Behaviors

Behavior (1 byte)

Behavior	MODE	Description
Bit 0	Card Operation Blinking LED	To blink the LED whenever the PICC card is being accessed. 1 = Enable; 0 =Disable
Bit 1	PICC Polling Status LED	To show the PICC polling status. 1 = Enable; 0 =Disable
Bit 2	PICC Activation Status LED	To show the activation status of the PICC interface. 1 = Enable; 0 =Disable
Bit 3	Card Insertion and Removal Events Buzzer	To make a beep whenever a card insertion or removal event is detected. (For PICC). 1 = Enable; 0 =Disabled
Bit 4	RFU	RFU
Bit 5	PN512 Reset Indication Buzzer	To make a beep when the PN512 is reset. 1 = Enable; 0 =Disabled
Bit 6	Color Select (GREEN)	Green LED for status change 1 = Enable; 0 =Disabled
Bit 7	Color Select (RED)	RED LED for status change 1 = Enable; 0 =Disabled

Note: *Default value of Behavior = 7Fh.*

5.7.10. Set Automatic PICC Polling

This command is used for setting the reader's polling mode.

Whenever the reader is connected to the PC, the PICC polling function will start the PICC scanning to determine if a PICC is placed on/removed from the built-antenna.

You can send a command to disable the PICC polling function. The command is sent through the PCSC Escape command interface. To meet the energy saving requirement, special modes are provided for turning off the antenna field whenever the PICC is inactive, or no PICC is found. The reader will consume less current in power saving mode.

Note: *The setting will be saved into non-volatile memory.*

Set Automatic PICC Polling Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set Automatic PICC Polling	E0h	00h	00h	23h	01h	Polling Setting

Set Automatic PICC Polling Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	Polling Setting

Polling Setting (1 byte)

Polling Setting	Mode	Description
Bit 0	Auto PICC Polling	1 = Enable; 0 =Disable
Bit 1	Turn off Antenna Field if no PICC is found.	1 = Enable; 0 =Disable
Bit 2	Turn off Antenna Field if the PICC is inactive.	1 = Enable; 0 =Disable
Bit 3	Activate the PICC when detected.	1 = Enable; 0 =Disable
Bit 5 .. 4	PICC Poll Interval for PICC	<Bit 5 – Bit 4> <0 – 0> = 250 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	
Bit 7	Enforce ISO 14443-A Part 4	1= Enable; 0= Disable.

Note: *Default value of Polling Setting = 8Fh.*

Reminders:

1. *It is recommended to enable the option "Turn Off Antenna Field if the PICC is inactive", so that the "Inactive PICC" will not be exposed to the field all the time to prevent the PICC from "warming up".*
2. *The longer the PICC Poll Interval, the more efficient of energy saving. However, the response time of PICC Polling will become longer. The Idle Current Consumption in Power Saving Mode is about 60 mA, while the Idle Current Consumption in Non-Power Saving mode is about 130mA.* ***Note:*** *Idle Current Consumption = PICC is not activated.*
3. *The reader will activate the ISO 14443A-4 mode of the "ISO 14443A-4 compliant PICC" automatically. Type B PICC will not be affected by this option.*
4. *The JCOP30 card comes with two modes: ISO 14443A-3 (MIFARE Classic 1K) and ISO 14443A-4 modes. The application has to decide which mode should be selected once the PICC is activated.*

5.7.11. Read Automatic PICC Polling

This command is used for checking the current PICC polling setting.

Read Automatic PICC Polling Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read Automatic PICC Polling	E0h	00h	00h	23h	00h

Read the Configure Mode Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	Polling Setting

Polling Setting (1 byte)

Polling Setting	Mode	Description
Bit 0	Auto PICC Polling	1 = Enable; 0 =Disable
Bit 1	Turn off Antenna Field if no PICC is found.	1 = Enable; 0 =Disable
Bit 2	Turn off Antenna Field if the PICC is inactive.	1 = Enable; 0 =Disable
Bit 3	Activate the PICC when detected.	1 = Enable; 0 =Disable
Bit 5 .. 4	PICC Poll Interval for PICC	<Bit 5 – Bit 4> <0 – 0> = 250 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	
Bit 7	Enforce ISO 14443-A Part 4	1= Enable; 0= Disable.

Note: *Default value of Polling Setting = 8Fh.*

5.7.12. Set PICC Operating Parameter

This command is used for setting the PICC operating parameter.

Note: *The setting will be saved into non-volatile memory.*

Set the PICC Operating Parameter Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set the PICC Operating Parameter	E0h	00h	00h	20h	01h	Operation Parameter

Set the PICC Operating Parameter Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1	00h	00h	00h	01h	Operation Parameter

Operating Parameter (1 byte)

Operating Parameter	Parameter	Description	Option
Bit 0	ISO 14443 Type A	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit 1	ISO 14443 Type B		1 = Detect 0 = Skip
Bit 2	FeliCa 212 Kbps		1 = Detect 0 = Skip
Bit 3	FeliCa 424 Kbps		1 = Detect 0 = Skip
Bit 4	Topaz		1 = Detect 0 = Skip
Bit 5 - 7	RFU	RFU	RFU

Note: *Default value of Operation Parameter = 1Fh.*

5.7.13. Read PICC Operating Parameter

This command is used for checking the current PICC operating parameter.

Read the PICC Operating Parameter Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read the PICC Operating Parameter	E0h	00h	00h	20h	00h

Read the PICC Operating Parameter Response Format (6 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	Operation Parameter

Operating Parameter (1 byte)

Operating Parameter	Parameter	Description	Option
Bit 0	ISO 14443 Type A	The Tag Types to be detected during PICC polling.	1 = Detect 0 = Skip
Bit 1	ISO 14443 Type B		1 = Detect 0 = Skip
Bit 2	FeliCa 212 Kbps		1 = Detect 0 = Skip
Bit 3	FeliCa 424 Kbps		1 = Detect 0 = Skip
Bit 4	Topaz		1 = Detect 0 = Skip
Bit 5 - 7	RFU	RFU	RFU

Note: *Default value of Operation Parameter = 1Fh.*

5.8. NFC Peer-to-Peer Related Commands

5.8.1. Initiator Mode-related Commands

This section provides the commands that can be used in Initiator Mode. The figure below shows the peer-to-peer flow of commands for this mode.

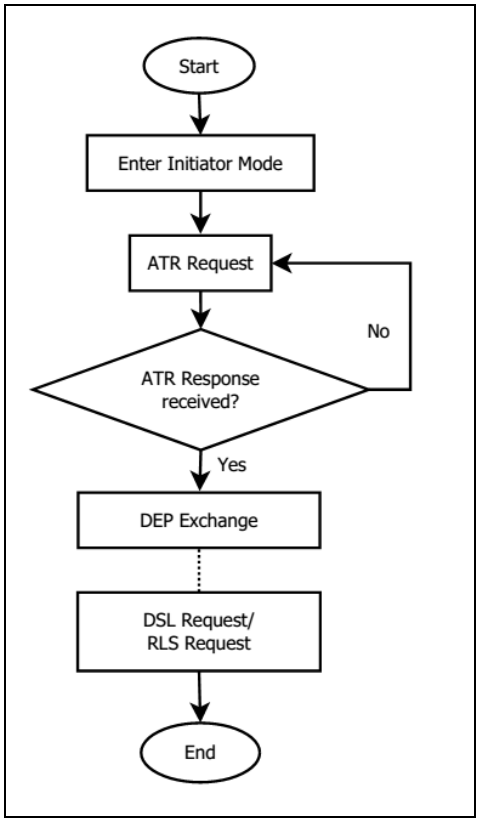

Figure 4: Peer-to-Peer Flow for Initiator Mode

5.8.1.1. Set Initiator Mode Timeout

This command is to set the timeout for Initiator Mode. Once the reader enters Initiator, it will retry 5 times (each time with 250 ms interval) in order to success exchange SNEP message.

Set Initiator Mode Timeout Command Format (7 bytes)

Command	Class	INS	P1	P2	Lc	Data In	
Set Initiator Mode	E0h	00h	00h	41h	02h	Timeout (MSB)	Timeout (LSB)

Note: *Unit = 10 ms, default value of Initiator Mode Timeout = 00 64h (100 * 10 ms = 1000 ms).*

Set Initiator Mode Timeout Response Format (7 bytes)

Response	Class	INS	P1	P2	Le	Data Out	
Result	E1h	00h	00h	00h	02h	Timeout (MSB)	Timeout (LSB)

Where:

Timeout 2 bytes. Timeout for Initiator Mode (10 ms).

5.8.1.2. Enter Initiator Mode

This command is to set the reader into Initiator Mode in order to send out SNEP Message.

Enter Initiator Mode Command Format (8 bytes)

Command	Class	INS	P1	P2	Lc	Data In		
Enter Initiator Mode	E0h	00h	00h	40h	03h	NfcMode	OpMode	Speed

Enter Initiator Mode Response Format (8 bytes)

Response	Class	INS	P1	P2	Le	Data Out		
Result	E1h	00h	00h	00h	03h	NfcMode	OpMode	Speed

Where:

NfcMode 1 byte. NFC Device Mode.
08h = Peer-to-Peer Initiator Mode
00h = Card Reader/Write Mode

OpMode 1 byte. Active Mode/Passive Mode.
01h = Active Mode
02h = Passive Mode

Speed 1 byte. Communication Speed.
01h = 106 Kbps
02h = 212 Kbps
03h = 424 Kbps

5.8.1.3. Send ATR Request

This command is used to poll peer-to-peer Target Mode device within the field.

ATR Request Command Format

Command	Class	INS	P1	P2	Lc	Data In				
ATR Request	E0h	00h	00h	42h	Len	11h	Mode (1 byte)	Speed (1 byte)	NFCID (10 bytes)	DID (1 byte)

Data In				
BS (1 byte)	BR (1 byte)	PP (1 byte)	LLCP Parameter	
			GiLen (1 byte)	Gi (GiLen bytes)

ATR Request Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	Len	ATR Response (Len bytes)

Where:

Mode 1 byte. Operation Mode.
01h = Active
02h = Passive

Speed 1 byte. Communication Speed.
01h = 106 Kbps
02h = 212 Kbps
03h = 424 Kbps

NFCID 10 bytes. Initiator device's NFCID.

DID 1 byte. Initiator device's device identification.

BS 1 byte. Initiator device's support send-bit rates.

BR 1 byte. Initiator device's support bit rates.

PP 1 byte. Initiator device's optional parameters.

Gi N byte. LLCP parameter.

5.8.1.4. Exchange DEP

This command is used to exchange DEP with target device.

DEP Exchange Command Format

Command	Class	INS	P1	P2	Lc	Data In			
DEP Exchange	E0h	00h	00h	43h	Len	11h	PFB (1 byte)	DepLen (1 byte)	Dep (N bytes)

DEP Exchange Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	Len	Dep Response (Len bytes)

Where:

PFB 1 byte. Controls the data transmission and error recovery.

DepLen 1 byte. DEP message length.

Dep N bytes. DEP message for peer-to-peer communication.

5.8.1.5. Send DSL Request

This command is used to send DSL request to target device.

DSL request Command Format

Command	Class	INS	P1	P2	Lc	Data In	
DSL request	E0h	00h	00h	44h	02h	11h	DID (1 byte)

Where:

DID 1 byte. Device Identification.

DSL request Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.1.6. Send RLS Request

This command is used to send RLS request to target device.

RLS request Command Format

Command	Class	INS	P1	P2	Lc	Data In	
RLS request	E0h	00h	00h	45h	02h	11h	DID (1 byte)

Where:

DID 1 byte. Device Identification.

RLS request Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.1.7. Send PSL Request

This command is used to send PSL request to target device.

PSL request Command Format

Command	Class	INS	P1	P2	Lc	Data In			
PSL request	E0h	00h	00h	46h	04h	11h	DID (1 byte)	BRS (1 byte)	FSL (1 byte)

Where:

DID 1 byte. Device Identification.

BRS 1 byte. Bit rates for Initiator and Target Device.

FSL 1 byte. Frame Length.

PSL request Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	01h	DID (1 byte)

Where:

DID 1 byte. Device Identification.

5.8.2. Target Mode-related Commands

This section provides the commands that can be used in Target Mode. The figure below shows the peer-to-peer flow of commands for this mode.

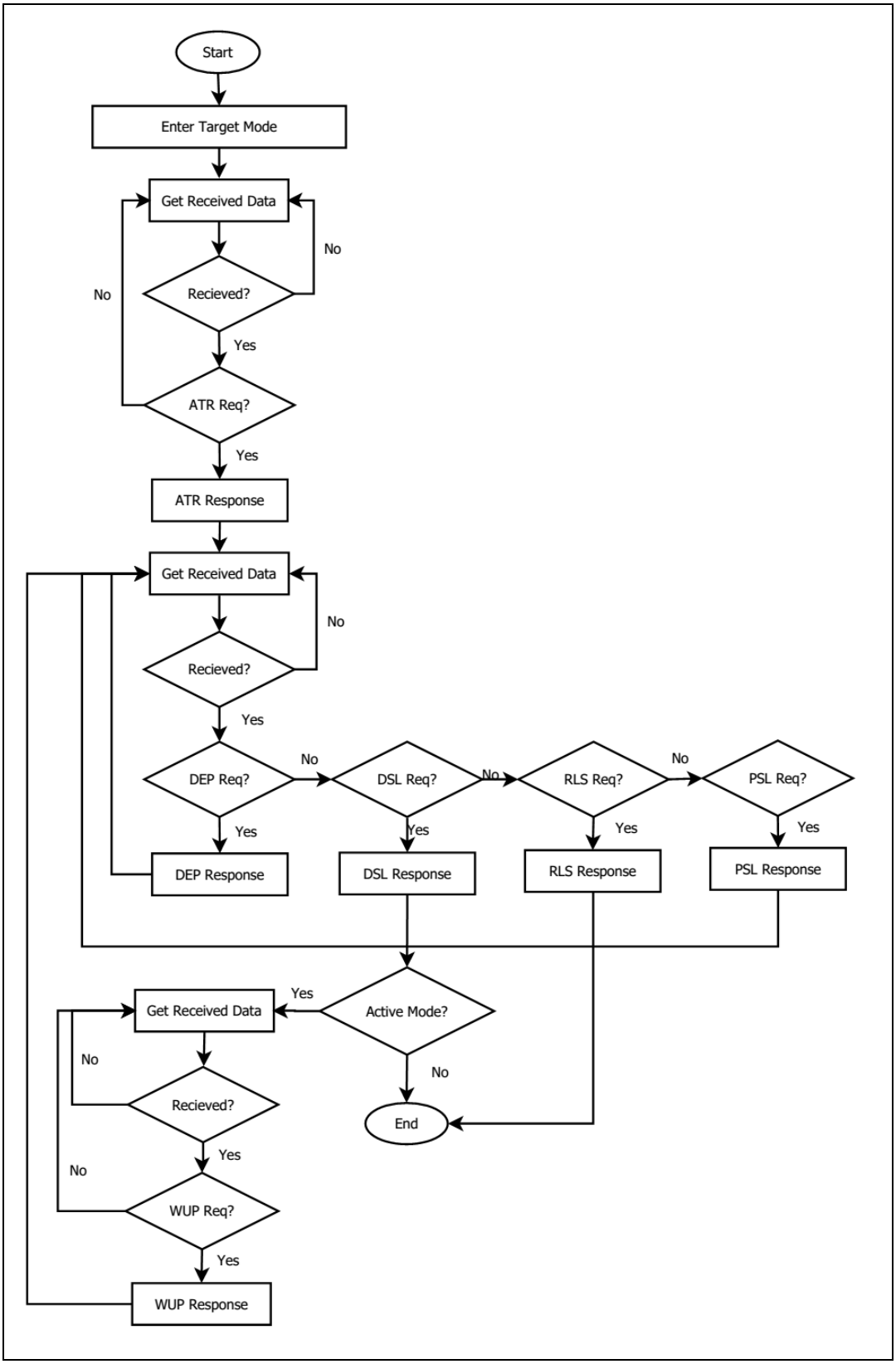

Figure 5: Peer-to-Peer Flow for Target Mode

5.8.2.1. Set Target Mode Timeout

This command is used to set the timeout for Target Mode.

Set Target Timeout Command Format

Command	Class	INS	P1	P2	Lc	Data In	
Set Target Timeout	E0h	00h	00h	59h	02h	Timeout (MSB)	Timeout (LSB)

Note: *Unit 100 µs, default value of Target Timeout = 00 C8h (200 * 100 µs = 20 ms).*

Set Target Timeout Response Format

Response	Class	INS	P1	P2	Le	Data Out	
Result	E1h	00h	00h	00h	02h	Timeout (MSB)	Timeout (LSB)

Where:

Timeout 2 bytes. Timeout for Initiator Mode (unit = 100 µs).

5.8.2.2. Enter Target Mode

This command is used to configure the reader into Target Mode to receive SNEP message.

Enter Target Mode Command Format

Command	Class	INS	P1	P2	Lc
Enter Target Mode	E0h	00h	00h	51h	00h

To set into the Target Mode with baud rate 106 Kbps and Passive Mode

or

Command	Class	INS	P1	P2	Lc	Data In	
Enter Target Mode	E0h	00h	00h	51h	02h	Speed	OpMode

Enter Target Mode Response Format

Response	Class	INS	P1	P2	Le	Data Out	
Result	E1h	00h	00h	00h	02h	Speed	OpMode

Where:

Speed 1 byte. Communication Speed.
01h = 106 Kbps
02h = 212 Kbps
03h = 424 Kbps

OpMode 1 byte. Active Mode/Passive Mode.
01h = Active Mode
02h = Passive Mode

5.8.2.3. Send ATR Response

This command is used to send ATR Response for the Initiator's ATR request.

ATR Response Command Format

Command	Class	INS	P1	P2	Lc	Data In
ATR Response	E0h	00h	00h	52h	Len	LLCP Parameter (N bytes)

Where:

LLCP Parameter N bytes. ATR response's General Byte.

ATR Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.2.4. Send DEP Response

This command is used to send DEP Response for the Initiator's DEP request.

DEP Response Command Format

Command	Class	INS	P1	P2	Lc	Data In	
DEP Response	E0h	00h	00h	53h	Len	PFB (1 byte)	DEP Message (N bytes)

Where:

PFB 1 byte. Controls the data transmission and error recovery.

DEP Message N bytes. DEP response.

DEP Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.2.5. Send DSL Response

This command is used to send DSL Response for the Initiator's DSL request.

DSL Response Command Format

Command	Class	INS	P1	P2	Lc
DSL Response	E0h	00h	00h	54h	00h

DSL Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.2.6. Send RLS Response

This command is used to send RLS Response for the Initiator's RLS request.

RLS Response Command Format

Command	Class	INS	P1	P2	Lc
RLS Response	E0h	00h	00h	55h	00h

RLS Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.2.7. Send PSL Response

This command is used to send PSL Response for the Initiator's PSL request.

PSL Response Command Format

Command	Class	INS	P1	P2	Lc	Data In	
PSL Response	E0h	00h	00h	56h	02h	BRS (1 byte)	FSL (1 byte)

Where:

BRS 1 byte. BRS Parameter.

FSL 1 byte. FSL Parameter.

PSL Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.2.8. Send WUP Response

This command is used to send WUP Response for the Initiator's WUP request.

WUP Response Command Format

Command	Class	INS	P1	P2	Lc
WUP Response	E0h	00h	00h	55h	00h

WUP Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	Return Code (2 bytes)

Return Code

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

5.8.2.9. Get Received Data

This command is used to get the message from the Initiator Mode device.

Get Received Data Command Format

Command	Class	INS	P1	P2	Lc
Get Received Data	E0h	00h	00h	58h	00h

Get Received Data Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	Len	Received message (N bytes)

5.9. ACR122U Compatible Commands

5.9.1. Bi-color LED and Buzzer Control

This command is used for controlling 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 (4 bytes)
Bi-color LED and Buzzer Control	FFh	00h	40h	LED State Control	04h	Blinking Duration Control

P2 LED State Control

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

CMD	Item	Description
Bit 0	Final Red LED State	1 = On; 0 = Off
Bit 1	Final Green LED State	1 = On; 0 = Off
Bit 2	Red LED State Mask	1 = Update the State 0 = No change
Bit 3	Green LED State Mask	1 = Update the State 0 = No change
Bit 4	Initial Red LED Blinking State	1 = On; 0 = Off
Bit 5	Initial Green LED Blinking State	1 = On; 0 = Off
Bit 6	Red LED Blinking Mask	1 = Blink 0 = Not Blink
Bit 7	Green LED Blinking Mask	1 = Blink 0 = Not Blink

Data In Blinking Duration Control

Bi-color LED Blinking Duration Control Format (4 bytes)

Byte 0	Byte 1	Byte 2	Byte 3
T1 Duration Initial Blinking State (unit = 100 ms)	T2 Duration Toggle Blinking State (unit = 100 ms)	Number of repetition	Link to Buzzer

Where:

Byte 3 Link to Buzzer. Control the buzzer state during the LED Blinking.

00h = The buzzer will not turn on.

01h = The buzzer will turn on during the T1 Duration.

02h = The buzzer will turn on during the T2 Duration.

03h = The buzzer will turn on during the T1 and T2 Duration.

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

Status Code

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

Reminders:

1. *The LED State operation will be performed after the LED Blinking operation is completed.*
2. *The LED will not change if the corresponding LED Mask is not enabled.*
3. *The LED will not blink if the corresponding LED Blinking Mask is not enabled. Also, the number of repetition must be greater than zero.*
4. *T1 and T2 duration parameters are used for controlling the duty cycle of LED blinking and Buzzer Turn-On duration. For example, if T1=1 and T2=1, the duty cycle = 50%.*

   Note: *Duty Cycle = T1 / (T1 + T2).*
5. *To control only the buzzer, just set the P2 "LED State Control" to zero.*
6. *To make the buzzer operate, the "number of repetition" must greater than zero.*
7. *To control only the LED, just set the parameter "Link to Buzzer" to zero.*

5.9.2. Get Firmware Version

This command is used for retrieving the firmware version of the reader.

Get Firmware Version Command Format (5 bytes)

Command	Class	INS	P1	P2	Le
Get Firmware	FFh	00h	48h	00h	00h

Get Firmware Version Response Format (X bytes)

Response	Data Out
Result	Firmware Version

Example:

Response = 41 43 52 31 32 35 31 55 5F 56 32 30 34 2E 30h = ACR1251U_V204.0 (ASCII)

5.9.3. Get PICC Operating Parameter

This command is used for getting the PICC operating parameter of the reader.

Get the PICC Operating Parameter Command Format (5 bytes)

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

Get the PICC Operating Parameter Response Format (2 byte)

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 ISO 14443-4 Type A tag is activated.	1 = Enable 0 = Disable
5	Polling Interval	To set the time interval between successive PICC polling.	1 = 250 ms 0 = 500 ms
4	FeliCa 424 Kbps	The Tag Types to be detected during PICC polling.	1 = Detect 0 = Skip
3	FeliCa 212 Kbps		1 = Detect 0 = Skip
2	Topaz		1 = Detect 0 = Skip
1	ISO 14443 Type B		1 = Detect 0 = Skip
0	ISO 14443 Type A ***Note:*** *To detect the MIFARE tags, the Auto ATS Generation must be disabled first.*		1 = Detect 0 = Skip

5.9.4. Set PICC Operating Parameter

This command is used for setting the PICC operating parameter of the reader.

Set PICC operation Parameter Command Format (5 bytes)

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

Set PICC operation Parameter Response Format (2 byte)

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 ISO 14443-4 Type A tag is activated.	1 = Enable 0 = Disable
5	Polling Interval	To set the time interval between successive PICC polling.	1 = 250 ms 0 = 500 ms
4	FeliCa 424 Kbps	The Tag Types to be detected during PICC polling.	1 = Detect 0 = Skip
3	FeliCa 212 Kbps		1 = Detect 0 = Skip
2	Topaz		1 = Detect 0 = Skip
1	ISO 14443 Type B		1 = Detect 0 = Skip
0	ISO 14443 Type A ***Note:*** *To detect the MIFARE tags, the Auto ATS Generation must be disabled first.*		1 = Detect 0 = Skip

Appendix A. SNEP Message

For the data format, please refer to NFC Forum NFC Data Exchange Format (NDEF) Specifications 1.0.

Example:

SNEP Message = {D1 02 0F 53 70 D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6Bh}

Offset	Content	Length	Description
0	D1	1	NDEF header. TNF = 01h, SR=1, MB=1, ME=1
1	02	1	Record name length (2 bytes)
2	0F	1	Length of the Smart Poster data (15 bytes)
3	53 70 (“Sp”)	2	Record name
5	D1	1	NDEF header. TNF = 01h, SR=1, MB=1, ME=1
6	01	1	Record name length (1 byte)
7	0B	1	The length of the URI payload (11 bytes)
8	55 (“U”)	1	Record type: “U”
9	01	1	Abbreviation: “http://www.”
10	61 63 73 2E 63 6F 6D 2E 68 6B	10	The URL itself. “acs.com.hk”