



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1252U

NFC Forum Certified Reader



Application Programming Interface V1.08



Table of Contents

1.0.	Introduction	4
2.0.	Features	5
3.0.	Acronyms and Abbreviations	6
4.0.	Architecture	7
5.0.	Host Programming (PC-linked) API.....	8
5.1.	PCSC API	8
5.1.1.	SCardEstablishContext.....	8
5.1.2.	SCardListReaders.....	8
5.1.3.	SCardConnect.....	8
5.1.4.	SCardControl	8
5.1.5.	ScardTransmit.....	8
5.1.6.	ScardDisconnect.....	8
5.1.7.	APDU Flow.....	9
5.1.8.	Escape Command Flow.....	10
5.2.	Contactless Smart Card Protocol	11
5.2.1.	ATR Generation	11
5.3.	Pseudo APDU for Contactless Interface	14
5.3.1.	Get Data.....	14
5.4.	APDU commands for PCSC 2.0 Part 3 (version 2.02 or above)	15
5.4.1.	Command and Response APDU Format.....	15
5.4.2.	Manage Session Command.....	17
5.4.3.	Transparent Exchange Command.....	21
5.4.4.	Switch Protocol Command.....	25
5.4.5.	PCSC 2.0 Part 3 Example	26
5.5.	PICC Commands for MIFARE® Classic (1K/4K) Memory Cards.....	29
5.5.1.	Load Authentication Keys	29
5.5.2.	Authentication for MIFARE® Classic (1K/4K).....	30
5.5.3.	Read Binary Blocks.....	33
5.5.4.	Update Binary Blocks.....	34
5.5.5.	Value Block Operation (INC, DEC, STORE)	35
5.5.6.	Read Value Block.....	36
5.5.7.	Copy Value Block.....	37
5.6.	Accessing PCSC-compliant tags (ISO 14443-4).....	38
5.7.	Accessing FeliCa tags	40
5.8.	Peripherals Control	41
5.8.1.	Get Firmware Version	41
5.8.2.	LED Control.....	42
5.8.3.	LED Status	43
5.8.4.	Buzzer Control	44
5.8.5.	Buzzer Status.....	45
5.8.6.	Set LED and Buzzer Status Indicator Behavior for PICC Interface	46
5.8.7.	Read LED and Buzzer Status Indicator Behavior for PICC Interface.....	47
5.8.8.	Set Automatic PICC Polling	48
5.8.9.	Read Automatic PICC Polling	50
5.8.10.	Set PICC Operating Parameter	51
5.8.11.	Read PICC Operating Parameter	52
5.8.12.	Set Serial Number.....	53
5.8.13.	Set and lock Serial Number	54
5.8.14.	Read Serial Number.....	55
5.8.15.	Unlock Serial Number	56
5.9.	NFC Peer-to-Peer Mode-related Commands	57
5.9.1.	Initiator Mode-related Commands.....	57
5.9.2.	Target Mode-related Commands.....	61
5.10.	NFC Card Emulation Mode-related Commands.....	67
5.10.1.	Enter Card Emulation Mode.....	67



5.10.2.	Read Card Emulation Data (MIFARE® Ultralight® or FeliCa).....	69
5.10.3.	Write Card Emulation Data (MIFARE® Ultralight® or FeliCa).....	70
5.10.4.	Set Card Emulation of MIFARE® Ultralight® UID	70
5.10.5.	Set Card Emulation FeliCa IDm.....	71
5.10.6.	Set Card Emulation Lock Data in NFC	71
5.11.	ACR122U Compatible Commands	72
5.11.1.	Bi-color LED and Buzzer Control	72
5.11.2.	Get Firmware Version	74
5.11.3.	Get the PICC Operating Parameter	75
5.11.4.	Set the PICC Operating Parameter	76
Appendix A. SNEP Message		77

List of Figures

Figure 1 :	ACR1252U Architecture.....	7
Figure 2 :	ACR1252U APDU Flow	9
Figure 3 :	ACR1252U Escape Command Flow.....	10
Figure 4 :	Peer-to-Peer Flow for Initiator Mode.....	57
Figure 5 :	Peer-to-Peer Flow for Target Mode	61

List of Tables

Table 1 :	Acronyms and Abbreviations.....	6
Table 2 :	MIFARE® Classic 1K Memory Map	31
Table 3 :	MIFARE® Classic 4K Memory Map	31
Table 4 :	MIFARE® Ultralight® Memory Map	32
Table 5 :	MIFARE® Ultralight® Memory Map (52 bytes)	68
Table 6 :	FeliCa Memory Map (160 bytes)	69



1.0. Introduction

The ACR1252U NFC Forum Certified reader is a USB PC-linked contactless card reader/writer with a SAM (Secure Access Module) slot, which can be used together with a SAM card for high level security in contactless transactions. It is an NFC card reader/writer, and capable to support card emulation and peer-to-peer communication modes. The ACR1252U reader is also NFC library compliant to support Bluetooth and Wi-Fi NFC pairing/log-in.

ACR1252U is compliant to ISO 14443 Parts 1 to 4 supporting contactless card, MIFARE® cards, FeliCa cards and ISO 18092 NFC tags.

ACR1252U has two reader interfaces, namely the PICC and SAM interface. Both interfaces 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 ACR1252U.



2.0. Features

- USB 2.0 Full Speed (12 Mbps)
- Microsoft® CCID Compliant for both SAM and PICC interface
- 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® Classic, MIFARE® Mini, MIFARE® Ultralight®, FeliCa, Topaz, and all four types of NFC (ISO/IEC 18092 tags)
 - Built-in anti-collision feature (only one tag is accessed at a time)
 - NFC Support
 - Card reader/writer mode
 - Peer-to-Peer mode
 - Card emulation mode
 - ISO 7816-compliant SAM slot (Class A)
- 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 later
- Compliant with the following standards:
 - ISO 18092 (NFCIP-1)
 - ISO 14443 Parts 1 -4
 - ISO 7816 Class A (for SAM slot)
 - NFC Forum
 - FeliCa Performance Certification
 - PC/SC
 - CCID
 - CE
 - FCC
 - RoHS 2
 - REACH
 - J-LIS (Japan)
 - VCCI (Japan)
 - MIC (Japan)
 - KC (Korea)
 - Microsoft® WHQL



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 ACR1252U and the computer is CCID protocol. All communications between PICC and SAM are PC/SC compliant.

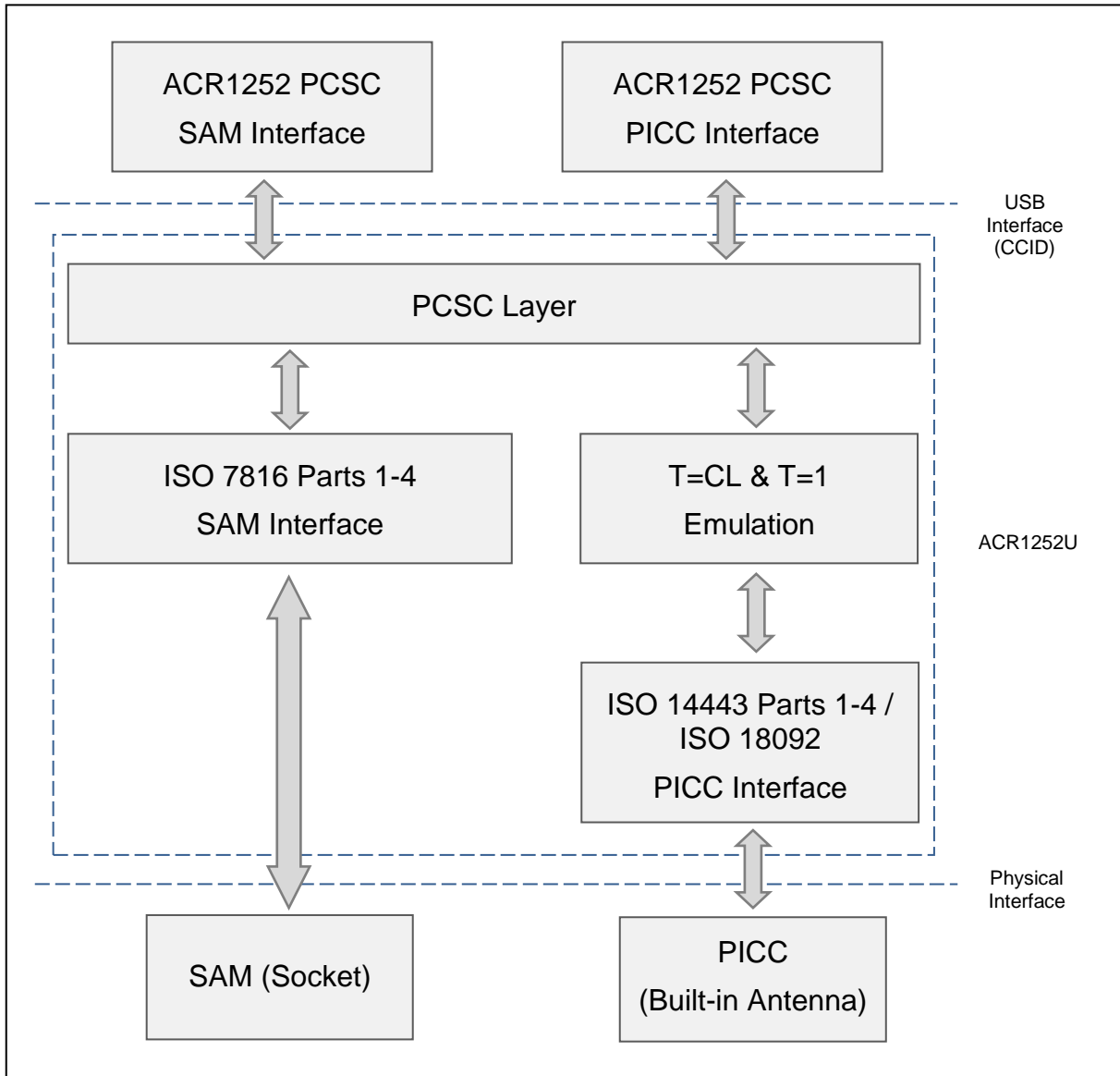


Figure 1: ACR1252U Architecture



5.0. Host Programming (PC-linked) API

5.1. PCSC API

This section will describe some of the PCSC API 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 5.8 – 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 to: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa379804%28v=vs.85%29.aspx>

Note: APDU Commands (i.e. the commands 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

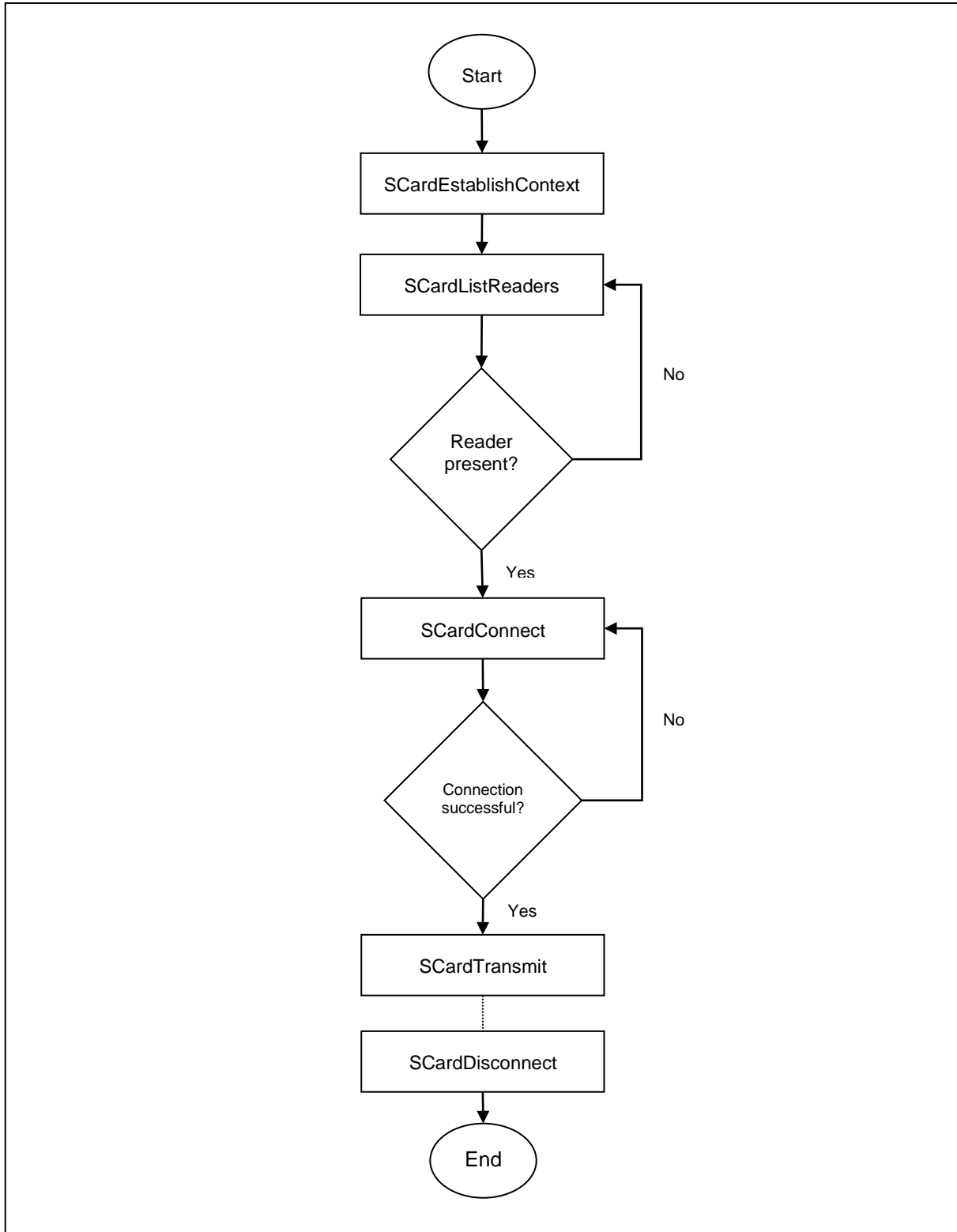


Figure 2: ACR1252U APDU Flow

5.1.8. Escape Command Flow

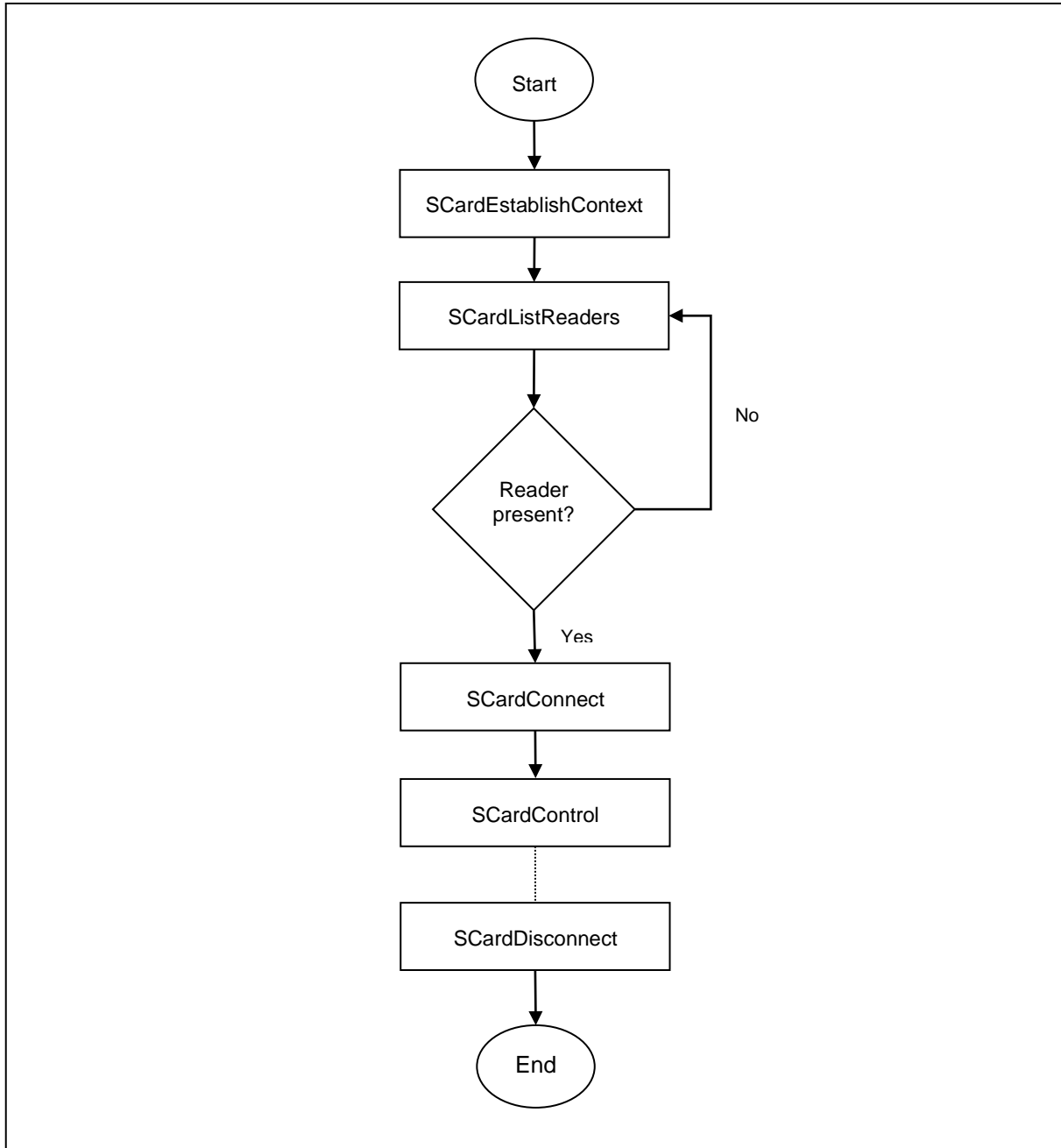


Figure 3: ACR1252U Escape Command Flow

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
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 38: MIFARE Plus SL2 2K
00 02: MIFARE Classic 4K	00 39: MIFARE Plus SL2 4K
00 03: MIFARE Ultralight	00 30: Topaz and Jewel
00 26: MIFARE Mini	00 3B: FeliCa
00 3A: MIFARE Ultralight C	FF 28: JCOP 30
00 36: MIFARE Plus SL1 2K	FF [SAK]: undefined tags
00 37: MIFARE Plus SL1 4K	

5.2.1.2. ATR Format for ISO 14443 Part 4 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	XX	T1	Historical Bytes: ISO 14443-A: The historical bytes from ATS response. Refer to the ISO14443-4 specification. ISO 14443-B:					
	XX XX XX	Tk		<table border="1"> <thead> <tr> <th>Byte1-4</th> <th>Byte5-7</th> <th>Byte8</th> </tr> </thead> <tbody> <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> </tbody> </table>	Byte1-4	Byte5-7	Byte8	Application Data from ATQB
Byte1-4	Byte5-7	Byte8						
Application Data from ATQB	Protocol Info Byte from ATQB	Higher nibble=MBLI from ATTRIB command Lower nibble (RFU)=0						
4+N	UU	TCK	Exclusive-oring of all the bytes T0 to Tk					



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	XXh	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. APDU commands for PCSC 2.0 Part 3 (version 2.02 or above)

PCSC2.0 Part 3 commands are used to transparently pass data from an application to a contactless tag, return the received data transparently to the application and protocol, and switch the protocol simultaneously.

5.4.1. Command and Response APDU Format

Command Format

CLA	INS	P1	P2	Lc	Data In
FFh	C2h	00h	Function	DataLen	Data[DataLen]

Where:

Functions 1 byte.
 00h = Manage Session
 01h = Transparent Exchange
 02h = Switch Protocol
 Other = RFU

Response Format

Data Out	SW1	SW2
Data Field BER-TLV encoded		

Every command returns SW1 and SW2 together with the response data field (if available). The SW1 SW2 is based on ISO 7816. SW1 SW2 from the C0 data object below should also be used.

C0 data element Format

Tag	Length (1 byte)	SW2
C0h	03h	Error Status

Error Status Description

Error Status	Description
XX SW1 SW2	XX = number of the bad data object in the APDU 00 = general error of APDU 01 = error in the 1 st data object 02 = error in the 2 nd data object
00 90 00h	No error occurred
XX 62 82h	Data object XX warning, requested information not available
XX 63 00h	No information
XX 63 01h	Execution stopped due to failure in other data object
XX 6A 81h	Data object XX not supported
XX 67 00h	Data object XX with unexpected length



Error Status	Description
XX 6A 80h	Data object XX with unexpected vale
XX 64 00h	Data Object XX execution error (no response from IFD)
XX 64 01h	Data Object XX execution error (no response from ICC)
XX 6F 00h	Data object XX failed, no precise diagnosis

The first value byte indicates the number of the erroneous data object XX, while the last two bytes indicate the explanation of the error. SW1 SW2 values based on ISO 7816 are allowed.

If there are more than one data object in the C-APDU field and one data object failed, IFD can process the following data objects if they do not depend on the failed data objects.



5.4.2. Manage Session Command

This command is used to manage the transparent session. This includes starting and ending a transparent session. Through this command, you can also manage the operation environment and the capabilities of the IFD within the transparent session.

Manage Session Command

Command	Class	INS	P1	P2	Lc	Data In
Manage Session	FFh	C2h	00h	00h	DataLen	DataObject (N bytes)

Where:

Data Object (1 byte)

Tag	Data Object
80h	Version Data Object
81h	Start Transparent Session
82h	End Transparent Session
83h	Turn Off RF Field
84h	Turn On RF Field
5F 46h	Timer
FF 6Dh	Get Parameter
FF 6Eh	Set Parameter

Manage Session Response Data Object

Tag	Data Object
C0h	Generic Error status
80h	Version data object
FF 6Dh	IFD parameter data object

5.4.2.1. Start Session Data Object

This command is used to start a transparent session. Once the session has started, auto-polling will be disabled until the session is ended.

Start Session Data Object

Tag	Length (1 byte)	Value
81h	00h	-



5.4.2.2. End Session Data Object

This command ends the transparent session. The auto-polling will be reset to the state before the session has started.

End Session Data Object

Tag	Length (1 byte)	Value
82h	00h	-

5.4.2.3. Version Data Object

This command is used to return the version number of the IFD handler.

Version Data Object

Tag	Length (1 byte)	Value		
80h	03h	Major	Minor	Build

5.4.2.4. Turn Off the RF Data Object

This command is used to turn off the antenna field.

Turn off RF Field Data Object

Tag	Length (1 byte)	Value
83h	00h	-

5.4.2.5. Turn On the RF Data Object

This command is used to turn on the antenna field.

Turn on the RF Field Data Object

Tag	Length (1 byte)	Value
84h	00h	-

5.4.2.6. Timer Data Object

This command is used to create a 32-bit timer data object in unit of 1 μ s.

Example: If there is timer data object with 5000 μ s between RF Turn Off Data Object and RF Turn On Data Object, the reader will turn off the RF field for about 5000 μ s before it is turned on.

Timer Data Object

Tag	Length (1 byte)	Value
5F 46h	04h	Timer (4 bytes)

5.4.2.7. Get Parameter Data Object

This command is used to get the different parameters from the IFD.

Get Parameter Data Object

Tag	Length (1 byte)	Value		
		Tag	Len	Value
FF 6Dh	Var	TLV_Objects		

TLV_Objects

Parameters Requested	Tag	Length
Frame size for IFD integer (FSDI)	01h	00h
Frame size for ICC integer (FSCI)	02h	00h
Frame waiting time integer (FWTI)	03h	00h
Max. Communication Speed supported by the IFD	04h	00h
Communication Speed of the ICC	05h	00h
Modulation Index	06h	00h
PCB for ISO/IEC14443	07h	00h
CID for ISO/IEC14443	08h	00h
NAD for ISO/IEC14443	09h	00h
Param 1 – 4 for for ISO/IEC14443 type B	0Ah	00h

5.4.2.8. Set Parameter Data Object

This command is used to set different parameters from the IFD.

Set Parameter Data Object

Tag	Length (1 byte)	Value		
		Tag	Len	Value
FF 6Eh	Var	TLV_Objects		

TLV_Objects

Parameters Requested	Tag	Length
Frame size for IFD integer (FSDI)	01h	01h
Frame size for ICC integer (FSCI)	02h	01h
Frame waiting time integer (FWTI)	03h	01h
Max. Communication Speed supported by the IFD	04h	01h
Communication Speed of the ICC	05h	01h
Modulation Index	06h	01h



Parameters Requested	Tag	Length
PCB for ISO/IEC14443	07h	01h
CID for ISO/IEC14443	08h	01h
NAD for ISO/IEC14443	09h	01h
Param 1 – 4 for for ISO/IEC14443 type B	0Ah	04h



5.4.3. Transparent Exchange Command

This command is used to transmit and receive any bit or bytes from ICC.

Transparent Exchange Command

Command	Class	INS	P1	P2	Lc	Data In
TranspEx	FFh	C2h	00h	01h	DataLen	DataObject (N bytes)

Where:

Data Object (1 byte)

Tag	Data Object
90h	Transmission and Reception Flag
91h	Transmission Bit Framing
92h	Reception Bit Framing
93h	Transmit
94h	Receive
95h	Transceive – Transmit and Receive
FF 6Dh	Get Parameter
FF 6Eh	Set Parameter

Transparent Exchange Response Data Object

Tag	Data Object
C0h	Generic Error status
92h	Number of valid bits in the last byte of received data
96h	Response Status
97h	ICC response
FF 6Dh	IFD parameter data object

5.4.3.1. Transmission and Reception Flag Data Object

This command is used to define the framing and RF parameters for the following transmission.

Transmission and Reception Flag Data Object

Tag	Length (1 byte)	Value	
		bit	Description
90h	02h	0	0 – append CRC in the transmit data 1 – do not append CRC in the transmit data
		1	0 – discard CRC from the received data 1 – do not discard CRC from the received data (i.e. no CRC checking)



Tag	Length (1 byte)	Value	
		2	0 – insert parity in the transmit data 1 – do not insert parity
		3	0 – expect parity in received date 1 – do not expect parity (i.e. no parity checking)
		4	0 – append protocol prologue in the transmit data or discard from the response 1 – do not append or discard protocol prologue if any (e.g. PCB, CID, NAD)
		5-15	RFU

5.4.3.2. Transmission Bit Framing Data Object

This command is used to define the number of valid bits of the last byte of data to transmit or transceive.

Transmission bit Framing Data Object

Tag	Length (1 byte)	Value	
		bit	Description
91h	01h	0-2	Number of valid bits of the last byte (0 means all bits are valid)
		3-7	RFU

Transmission bit framing data object shall be together with “transmit” or “transceive” data object only. If this data object does not exist, it means all bits are valid.

5.4.3.3. Reception bit Framing Data Object

For the command APDU, this data object defines the number of expected valid bits of the last byte of data received.

For the response APDU, this data object mentions the number of valid bits in the last byte of received data.

Reception bit Framing Data Object

Tag	Length (1 byte)	Value	
		bit	Description
92h	01h	0-2	Number of valid bits of the last byte (0 means all bits are valid)
		3-7	RFU

If this data object does not exist, it means all bits are valid.



5.4.3.4. Transmit Data Object

This command is used to transmit the data from IFD to the ICC. No response is expected from the ICC after transmit is complete.

Transmit Data Object

Tag	Length (1 byte)	Value
93h	DataLen	Data (N bytes)

5.4.3.5. Receive Data Object

This command is used to force the reader into receiving mode within the time, given in the following timer object.

Receive Data Object

Tag	Length (1 byte)	Value
94h	00h	-

5.4.3.6. Transceive Data Object

This command is used to transmit and receive data from the ICC. After transmission is complete, the reader will wait until the time given in the timer data object.

If no timer data object was defined in the data field, the reader will wait for the duration given in the Set Parameter FWTI Data Object. If no FWTI is set, the reader will wait for about 302 μ s.

Transceive Data Object

Tag	Length (1 byte)	Value
95h	DataLen	Data (N Bytes)

5.4.3.7. Response Status Data Object

Inside the response, this command is used to notify the received data status.

Response Status Data Object

Tag	Length (1 byte)	Value		
		Byte 0		Byte 1
		Bit	Description	
96h	02h	0	0 – CRC is OK or no checked 1 – CRC check fail	If a collision is detected, these bytes will tell the collision position. Otherwise, "00h" will be shown.
		1	0 – no collision 1 – collision detected	
		2	0 – no parity error 1 – parity error detected	



Tag	Length (1 byte)	Value	
		3	0 – no framing error 1 – framing error detected
		4 - 7	RFU

5.4.3.8. Response Data Object

Inside the response, this command is used to notify the received data status.

Response Data Object

Tag	Length (1 byte)	Value
97h	DataLen	ReplyData (N Byte)



5.4.4. Switch Protocol Command

This command is used to specify the protocol and different layers of the standard within the transparent session.

Switch Protocol Command

Command	Class	INS	P1	P2	Lc	Data In
SwProtocol	FFh	C2h	00h	02h	DataLen	DataObject (N bytes)

Where:

Data Object (1 byte)

Tag	Data Object
8Fh	Switch Protocol Data Object
FF 6Dh	Get Parameter
FF 6Eh	Set Parameter

Switch Protocol Response Data Object

Tag	Data Object
C0h	Generic Error status
FF 6Dh	IFD parameter data object

5.4.4.1. Switch Protocol Data Object

This command is used to specify the protocol and different layers of the standard.

Switch Protocol Data Object

Tag	Length (1 byte)	Value	
		Byte 0	Byte 1
8Fh	02h	00h – ISO/IEC14443 Type A 01h – ISO/IEC14443 Type B 03h – FeliCa Other – RFU	00h – If no layer separation 02h – Switch to Layer 2 03h – Switch or activate to layer 3 04h – Activate to layer 4 Other - RFU

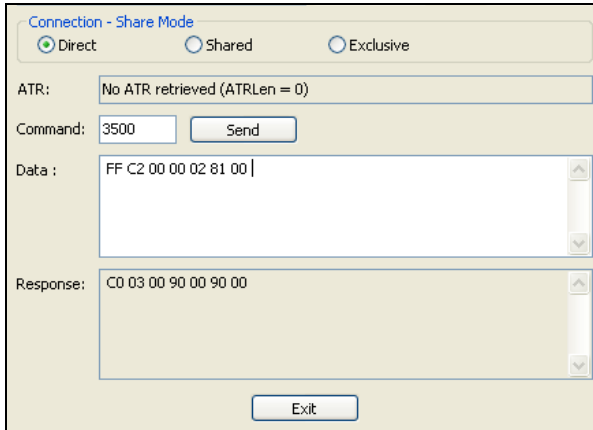


5.4.5. PCSC 2.0 Part 3 Example

1. Start Transparent Session.

Command: **FF C2 00 00 02 81 00**

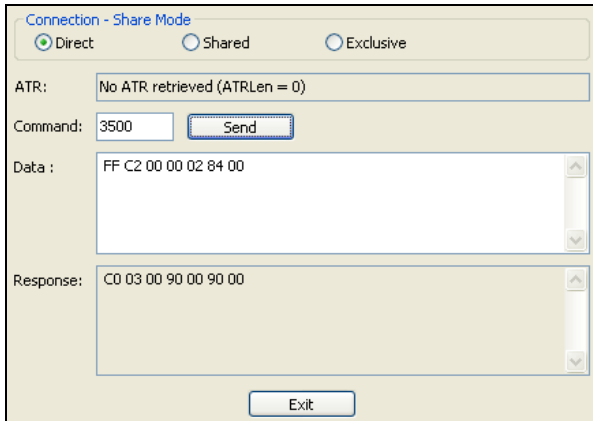
Response: **C0 03 00 90 00 90 00**



2. Turn the Antenna Field on.

Command: **FF C2 00 00 02 84 00**

Response: **C0 03 00 90 00 90 00**



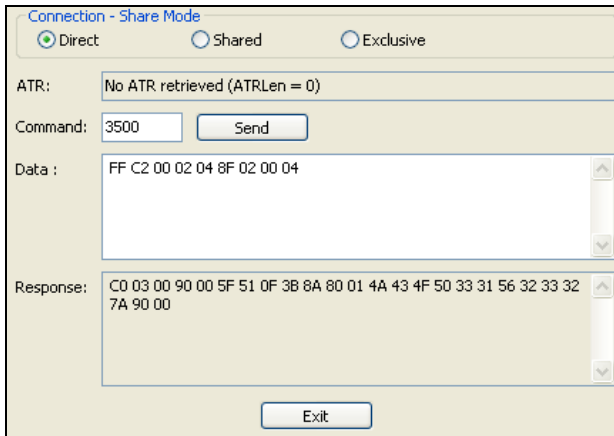


3. ISO14443-4A Active.

Command: **FF C2 00 02 04 8F 02 00 04**

Response: **C0 03 01 64 01 90 00** (if no card present)

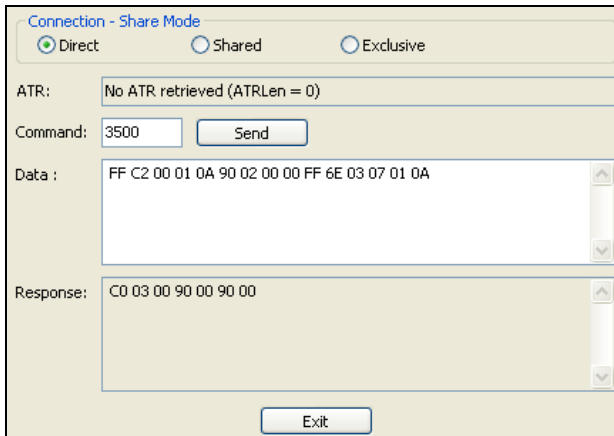
C0 03 00 90 00 5F 51 [ATR] 90 00



4. Set the PCB to 0Ah and enable the CRC, parity and protocol prologue in the transmit data.

Command: **FF C2 00 01 0A 90 02 00 00 FF 6E 03 07 01 0A**

Response: **C0 03 00 90 00 90 00**

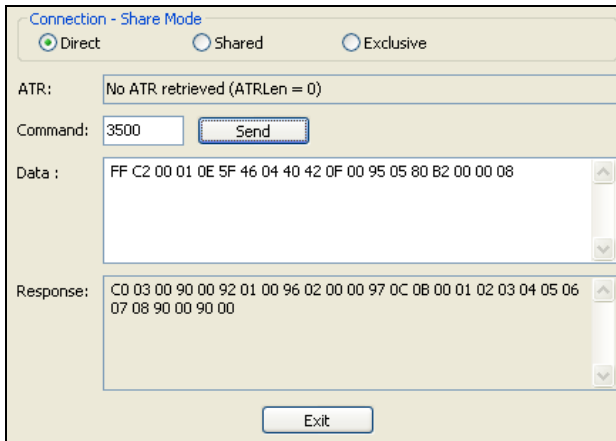




- 5. Send the APDU “80B2000008” to card and get response.

Command: **FF C2 00 01 0E 5F 46 04 40 42 0F 00 95 05 80 B2 00 00 08**

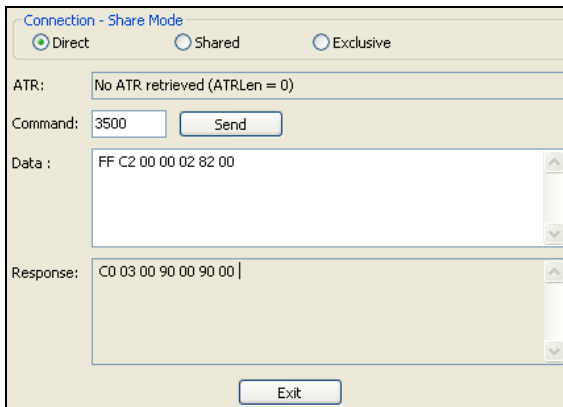
Response: **C0 03 00 90 00 92 01 00 96 02 00 00 97 0C [Card Response] 90 00**



- 6. End Transparent Session.

Command: **FF C2 00 00 02 82 00**

Response: **C0 03 00 90 00 90 00**





5.5. PICC Commands for MIFARE® Classic (1K/4K) Memory Cards

5.5.1. Load Authentication Keys

This command will load the authentication keys into the reader. The authentication keys are used to authenticate the particular sector of the MIFARE Classic 1K/4K Memory Card.

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 memory.
Other = Reserved.
- Key Number** 1 byte.
00h ~ 01h = Volatile memory for storing a temporary key. The key will disappear once the reader is disconnected from the computer. Two volatile keys are provided. The volatile key can be used as a session key for different sessions. *Default Value = {FF FF FF FF FF FFh}*
- 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

Load Authentication Keys Response Codes

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



5.5.2. 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 a total of 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 computer. 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	90h	00h	The operation is completed successfully.
Error	63h	00h	The operation is failed.

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	} 1 KB
Sector 1	04h – 06h	07h	
..	
..	
Sector 14	38h – 0Ah	3Bh	
Sector 15	3Ch – 3Eh	3Fh	

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	} 2 KB
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, 16 bytes)	
Sector 32	80h ~ 8Eh	8Fh	} 2 KB
Sector 33	90h ~ 9Eh	9Fh	
..			
..			
Sector 38	E0h ~ EEh	EFh	
Sector 39	F0h ~ FEh	FFh	

Table 3: 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	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

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}

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

5.5.3. 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 the “Read Binary Blocks” 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 or 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



5.5.4. 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.
 - Multiple 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 4K (Multiple Blocks Mode; 15 consecutive blocks).
- Block Data** Multiple of 16 bytes, or 4 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}
```

5.5.5. 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.



5.5.6. 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	90	00h	The operation is completed successfully.
Error	63	00h	The operation is failed.



5.5.7. 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}



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

All ISO 14443-4 compliant cards (PICCs) understands the ISO 7816-4 APDUs. The ACR1252U reader just has to communicate with the ISO 14443-4 compliant cards through exchanging ISO 7816-4 APDUs and responses. ACR1252U 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 5.5 – PICC Commands for MIFARE® Classic (1K/4K) Memory Cards.**

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 SW2	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]
```



5.7. Accessing FeliCa tags

For FeliCa access, the command is different with PCSC-compliant and MIFARE tags. 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



5.8. Peripherals Control

The reader's peripherals control commands are implemented by using **SCardControl** with Control Code **SCARD_CTL_CODE(3500)**.

5.8.1. 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 32 55 5F 56 31 30 30 2E 31

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

Firmware Version (ASCII) = "ACR1252U_V100.1"



5.8.2. 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



5.8.3. 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



5.8.4. 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



5.8.5. 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



5.8.6. 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.

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	Card Operation Blinking LED	To blink the LED whenever the PICC card is being accessed
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 = Disable
Bit 6	Color Select (GREEN)	GREEN LED for status change 1 = Enable; 0 = Disable
Bit 7	Color Select (RED)	RED LED for status change 1 = Enable; 0 = Disable

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



5.8.7. 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.

Read LED and Buzzer Status Indicator Behavior Format for PICC Interface (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 for PICC Interface (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
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 = Disable
Bit 6	Color Select (GREEN)	GREEN LED for status change 1 = Enable; 0 = Disable
Bit 7	Color Select (RED)	RED LED for status change 1 = Enable; 0 = Disable

Note: Default value of Behavior = 7Fh



5.8.8. Set Automatic PICC Polling

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

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



5.8.9. 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.



5.8.10. 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.



5.8.11. 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.



5.8.12. Set Serial Number

This command is used for setting the Serial Number.

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

Set the Serial Number Format

Command	Class	INS	P1	P2	Lc	Data In
Set the Serial Number	E0h	00h	00h	DAh	Len	Serial Number (N bytes)

Where:

Serial Number N bytes. Maximum 20 bytes.

Set the Serial Number Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1	00h	00h	00h	Len	Serial Number / Return Code (N bytes) / (2 bytes)

Return Code

Results	SW1 SW2	Meaning
Lock	90 00h	The serial number is already locked.
Error	63 00h	The serial number is too long.



5.8.13. Set and lock Serial Number

This command is used to **set** and **lock** serial number. The setting will be saved **permanently** into non-volatile memory.

Note: This will lock the serial number memory which unable to re-write a new serial number. The serial number will be locked and cannot re-write until used the **unlock** command.

Set the Serial Number Format

Command	Class	INS	P1	P2	Lc	Data In
Set the Serial Number	E0h	00h	00h	D9h	Len	Serial Number (N bytes)

Where:

Serial Number N bytes. Maximum 20 bytes.

Set the Serial Number Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1	00h	00h	00h	Len	Serial Number / Return Code (N bytes) / (2 bytes)

Return Code

Results	SW1 SW2	Meaning
Lock	90 00h	The serial number is already locked.
Error	63 00h	The serial number is too long.



5.8.14. Read Serial Number

This command is used for reading the serial number.

Read the Serial Number Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read the Serial Number	E0h	00h	00h	33h	00h

Read the Serial Number Response Format

Response	Class	INS	P1	P2	Le	Data Out
Result	E1	00h	00h	00h	Len	Serial Number (N bytes)



5.8.15. Unlock Serial Number

This command is used to **unlock** the serial number if the serial number has been set and locked. This will unlock and re-write new serial number if the existing serial number has been locked and needs to be reconfigured.

Note: For the “unlock serial command”, please contact us at info@acs.com.hk or contact an Advanced Card Systems Ltd. Sales Representative for the details.

5.9. NFC Peer-to-Peer Mode-related Commands

5.9.1. 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 Initiator Mode.

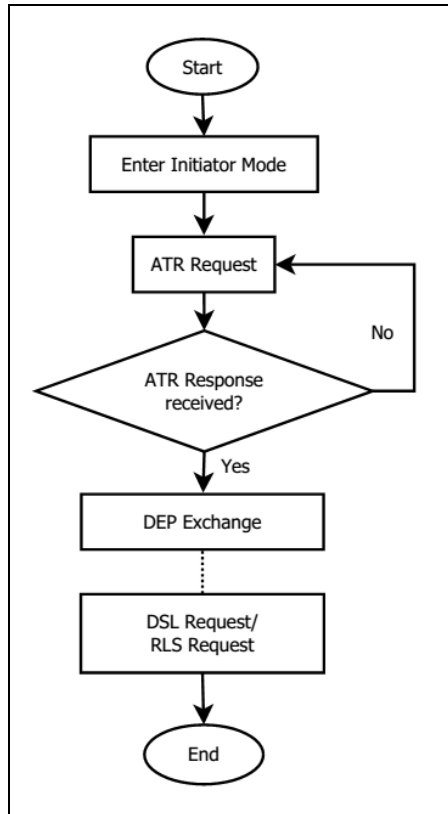


Figure 4: Peer-to-Peer Flow for Initiator Mode

5.9.1.1. Set Initiator Mode Timeout

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

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 (unit = 10 ms).

5.9.1.2. 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.
 01h = MIFARE Ultralight Card Emulation Mode
 03h = FeliCa Card Emulation Mode
 08h = Peer-to-Peer Initiator Mode
 00h = Card Read/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

After executing Enter Initiator Mode command, the reader will wait for the NFC device, which in Target Mode, will present and send out the pre-set SNEP Message to it. The reader will stop all other tasks until the SNEP Message is sent successfully.

5.9.1.3. Send ATR Request

This command is used to send an ATR_REQ to 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 bytes. LLCP parameter.

5.9.1.4. 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. Control the data transmission and error recovery.
- DepLen** 1 byte. DEP message length.
- Dep** N bytes. DEP message for peer-to-peer communication.



5.9.1.5. 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.

5.9.1.6. 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.

5.9.2. Target Mode-related Commands

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

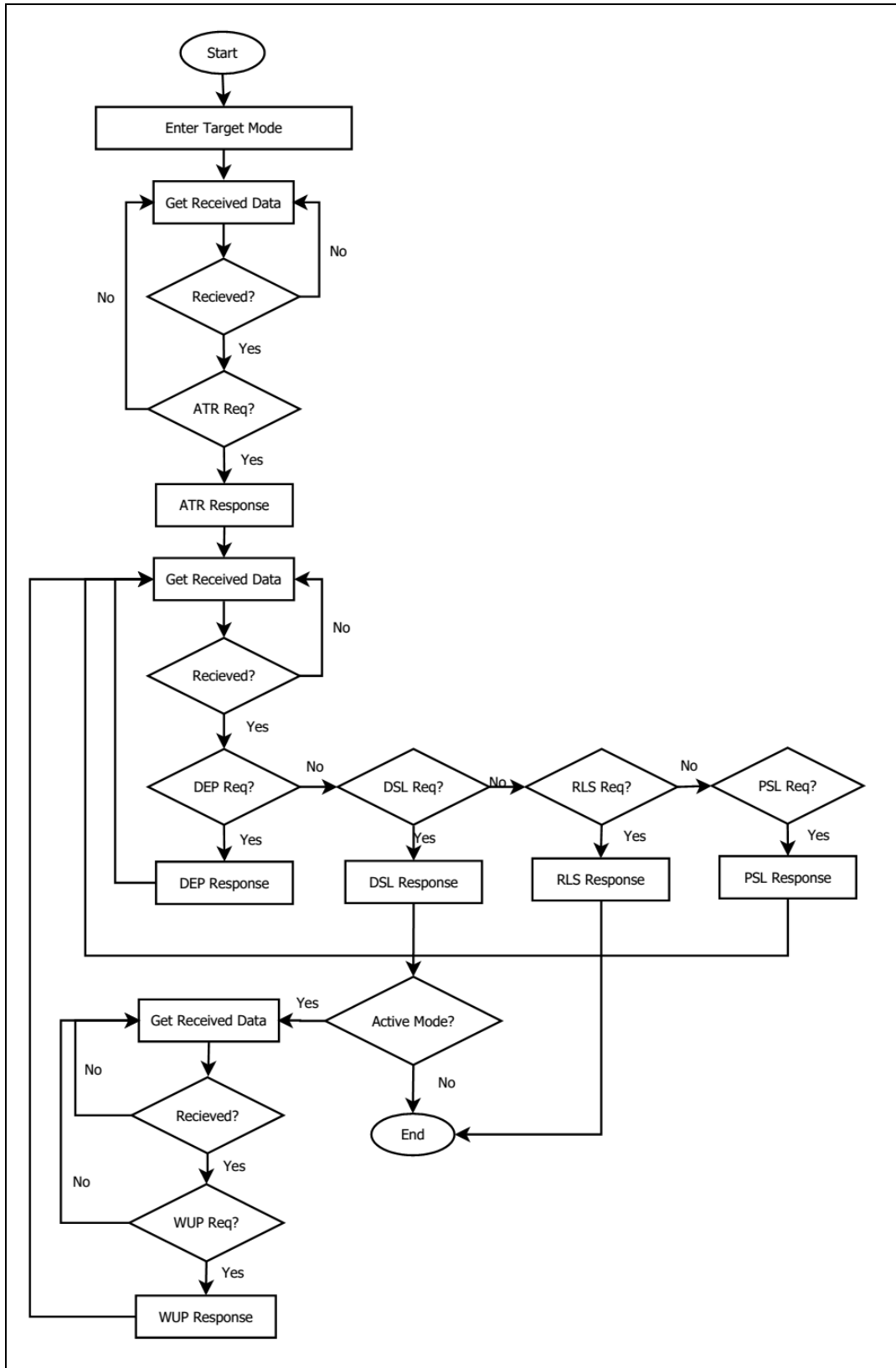


Figure 5: Peer-to-Peer Flow for Target Mode



5.9.2.1. Set Target Mode Timeout

This command is used to set the timeout for the reader in 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 μ s, default value of Target Timeout = 00 C8h (200 * 100 μ s = 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 Target Mode (unit =100 μ s).

5.9.2.2. Enter Target Mode

This command is used to set the reader into Target Mode to receive SNEP Message.

Enter Target Mode Command Format (8 bytes)

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

Enter Target Mode Response Format (8 bytes)

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

After executing Enter Target Mode, the reader will wait for the NFC device, which in Initiator Mode, will present and receive the SNEP message. The reader will stop all other tasks until the SNEP Message is exchanged successfully.

5.9.2.3. 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.

5.9.2.4. 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. Control 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.

5.9.2.5. 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.

5.9.2.6. 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.

5.9.2.7. 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.

5.9.2.8. 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	57h	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.

5.9.2.9. Get Received Data

This command is used to get the data received from NFC Initiator device.

Get Received Data Command Format (5 bytes)

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

Get Received Data Response Format (11 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	SNEP Message Len	SNEP Message

Where:

- SNEP Message Len** 1 byte. Length of the received SNEP Message.
- SNEP Message** Received SNEP message from Initiator devices.



5.10. NFC Card Emulation Mode-related Commands

5.10.1. Enter Card Emulation Mode

This command is used to set the reader into card emulation mode, in order to emulate a MIFARE Ultralight or a FeliCa Card.

Note: Lock byte is not supported in emulated MIFARE Ultralight. UID is user programmable.

Enter Card Emulation Mode Command Format (8 bytes)

Command	Class	INS	P1	P2	Lc	Data In		
Enter Card Emulation Mode	E0h	00h	00h	40h	03h	NFCMode	00h	00h

Enter Card Emulation Mode Response Format (8 bytes)

Response	Class	INS	P1	P2	Le	Data Out		
Result	E1h	00h	00h	00h	03h	NFCMode	01h	01h

Where:

- NFCMode** 1 byte. NFC Device Mode.
 01h = MIFARE Ultralight Card Emulation Mode
 03h = FeliCa Card Emulation Mode
 06h = Peer-to-Peer Initiator Mode
 Other = Card Read/Write Mode



Byte Number	0	1	2	3	Byte Address access by USB
Serial Number	SN0	SN1	SN2	BCC0	Nil
Serial Number	SN3	SN4	SN5	SN6	Nil
Internal/Lock	BCC1	Internal	Lock0	Lock1	Nil
Data read/write	Data0	Data1	Data2	Data3	0-3
Data read/write	Data4	Data5	Data6	Data7	4-7
Data read/write	Data8	Data9	Data10	Data11	8-11
Data read/write	Data12	Data13	Data14	Data15	12-15
Data read/write	Data16	Data17	Data18	Data19	16-19
Data read/write	Data20	Data21	Data22	Data23	20-23
Data read/write	Data24	Data25	Data26	Data27	24-27
Data read/write	Data28	Data29	Data30	Data31	28-31
Data read/write	Data32	Data33	Data34	Data35	32-35
Data read/write	Data36	Data37	Data38	Data39	36-39
Data read/write	Data40	Data41	Data42	Data43	40-43
Data read/write	Data44	Data45	Data46	Data47	44-47
Data read/write	Data48	Data49	Data50	Data51	48-51

} Accessible area
(52 bytes)

Table 5: MIFARE® Ultralight® Memory Map (52 bytes)

Where:

Default SN[0-6] {04h, 96h, 50h, 01h, F4h, 02h, 80h}

Default Data[0-3] {E1h, 10h, 06h, 00h} //NFC Type2Tag



Memory	1 Block data (16 Byte)	Byte Address access by USB
Data read/write	Block 0	0-15
Data read/write	Block 1	16-31
Data read/write	Block 2	32-47
Data read/write	Block 3	48-63
Data read/write	Block 4	64-79
Data read/write	Block 5	80-95
Data read/write	Block 6	96-111
Data read/write	Block 7	112-127
Data read/write	Block 8	128-143
Data read/write	Block 9	144-159

Table 6: FeliCa Memory Map (160 bytes)

Where:

Default: Block 0 data: {10h, 01h, 01h, 00h, 09h, 00h, 00h, 00h, 00h, 00h, 01h, 00h, 00h, 00h, 00h, 1Ch}

Default Block 0 data NFC Type3 Tag Attribute Information Block

Notes:

1. FeliCa card emulation support Read/Write without Encryption
2. FeliCa Card Identification Number in IDm is user programmable while Manufacturer Code is fixed at (03 88).

5.10.2. Read Card Emulation Data (MIFARE® Ultralight® or FeliCa)

This command is used to read the emulated card content.

Read Card Emulation Data Command Format (9 bytes)

Command	Class	INS	P1	P2	Lc	Data In			
Read Card Emulation Data	E0h	00h	00h	60h	04h	00h	NFCMode	StartOffset	Length

Read Card Emulation Data Response Format (Data + 5 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	Length	Data being read

Where:

NFCMode 1 byte. NFC Device Mode.
01h = MIFARE Ultralight Card Emulation Mode
03h = FeliCa Card Emulation Mode



- StartOffset** 1 byte. Address start to read.
- Length** 1 byte. Number of byte going to read.
- Data being read** The output data.

5.10.3. Write Card Emulation Data (MIFARE® Ultralight® or FeliCa)

This command is used to write to the emulated card.

Write Card Emulation Data Command Format

Command	Class	INS	P1	P2	Lc	Data In				
Write Card Emulation Data	E0h	00h	00h	60h	Length + 4	01h	NFCMode	StartOffset	Length	Data to write

Write Card Emulation Data Response Format (8 bytes)

Response	Class	INS	P1	P2	Le	Data Out		
Result	E1h	00h	00h	00h	03h	Length	90h	00h

Where:

- NfcMode** 1 byte. NFC Device Mode.
01h = MIFARE Ultralight Card Emulation Mode
03h = FeliCa Card Emulation Mode
- StartOffset** 1 byte. Address start to read.
- Length** 1 byte. Number of bytes to write.
- Data to Write** The binary data to write.

5.10.4. Set Card Emulation of MIFARE® Ultralight® UID

This command is used to set the UID of the emulated MIFARE Ultralight card.

Set Card Emulation MIFARE Ultralight UID Command Format (12 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set Card Emulation	E0h	00h	00h	61h	07h	7 bytes UID

Set Card Emulation MIFARE Ultralight UID Response Format (7 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	02h	90h 00h

Where:

- UID** 7 bytes. 7 bytes MIFARE UID.



5.10.5. Set Card Emulation FeliCa IDm

This command is to set the 6-byte FeliCa Card Identification number on emulated FeliCa card.

Set Card Emulation FeliCa Card Identification number Command Format (11 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Enter Initiator Mode	E0h	00h	00h	64h	06h	IDm

Set Card Emulation FeliCa Card Identification number Response Format (11 bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	06h	IDm

Where:

IDm 6 bytes.

5.10.6. Set Card Emulation Lock Data in NFC

This command is to set the lock for card emulation data in NFC communication. If the data is locked, it is protected from being overwritten via NFC.

Set Card Emulation Lock Data in NFC Command Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Enter Initiator Mode	E0h	00h	00h	65h	01h	Lock

Set Card Emulation lock data in NFC Response Format (7 bytes)

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

Where:

Lock 1 byte. Protect the data from being overwritten via NFC.

Lock Parameter

Bit	Parameter	Description	Option
7-2	Reserved	Reserved	
1	FeliCa Lock Enable	Data cannot be modified via NFC. The data can still be modified by using the USB escape command.	0: Lock disable 1: Lock enable
0	MIFARE Ultralight Lock Enable		0: Lock disable 1: Lock enable



5.11. ACR122U Compatible Commands

5.11.1. Bi-color LED and Buzzer Control

This command is used to control 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	63h	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	RFU	RFU

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 the buzzer only, 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 the LED only, just set the parameter “Link to Buzzer” to zero.



5.11.2. Get Firmware Version

This command is used to retrieve 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 32 55 5F 56 31 30 30 2E 31h = ACR1252U_V100.1 (ASCII)



5.11.3. Get the PICC Operating Parameter

This command is used to get 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 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 = 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	ISO14443 Type B		1 = Detect 0 = Skip
0	ISO14443 Type A <i>Note: To detect the MIFARE Tags, the Auto ATS Generation must be disabled first.</i>		1 = Detect 0 = Skip



5.11.4. Set the PICC Operating Parameter

This command is used to set 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 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 = 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	ISO14443 Type B		1 = Detect 0 = Skip
0	ISO14443 Type A <i>Note: To detect the MIFARE tags, the Auto ATS Generation must be disabled first.</i>		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"

Android is trademark of Google Inc.
Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other countries.
MIFARE, MIFARE Classic, MIFARE DESFire and MIFARE Ultralight are registered trademarks of NXP B.V.