Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1581U
DualBoost III USB Dual Interface Reader

Reference Manual V1.03

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

Table of Contents

List of Figures

List of Tables

1.0. Introduction

Continuing the success of the ACR1281U, ACR1581U DualBoost III is the third generation product in the ACS's DualBoost Reader Series. ACR1581U is a Dual Interface Reader that is able to access any contact and contactless smart cards alike, and makes use of the USB CCID Class Driver and USB Interface to establish connection with PCs to accept card commands from the computer's application. Not only does it support traditional ISO 7816 MCU Cards, MIFARE® Cards and ISO 14443 Type A and B Contactless Cards, it also is able to implement additional support for FeliCa and ISO 15693 Cards.

The ACR1581U acts as the intermediary device between the computer and the card. The reader, which communicates with a contactless tag, MCU card, SAM card, or the device peripherals (LED or buzzer), will carry out commands issued from the computer. It has three interfaces: the PICC, ICC and SAM interface, which all follow the PC/SC specifications. The contact interface makes use of the APDU commands as defined in ISO 7816 specifications. For contact MCU card operations, please refer to the related card documentation and the PC/SC specifications.

This API document details how the PC/SC APDU commands are implemented for the contactless interface, contact memory card support and device peripherals of the ACR1581U.

2.0. Features

- USB Full Speed Interface
- CCID-compliant
- Smart Card Reader:
  - Contactless Interface:
    - Read/Write speed of up to 26kbps ISO 15693 & 848 kbps (ISO 14443) card types
    - Built-in antenna for contactless tag access, with card reading distance of up to 70 mm (depending on tag type)
    - Supports ISO 15693 card types
    - Supports ISO 14443 Part 4 Type A and B cards and MIFARE series
    - Built-in anti-collision feature
    - Supports extended APDU (max. 64 KB)
  - Contact Interface:
    - Supports ISO 7816 Class A, B and C (5 V, 3 V and 1.8 V)
    - Supports CAC (Common Access Card)
    - Supports PIV (Personal Identity Verification Card)
    - Supports microprocessor cards with T=0 or T=1 protocol
    - Supports PPS (Protocol and Parameters Selection)
    - Features Short Circuit Protection
    - Supports extended APDU (max. 64 KBytes for T=1; max. 512+10 Bytes for T=0)
  - SAM Interface:
    - One SAM Slot
    - Supports ISO 7816 Class A SAM cards
- Application Programming Interface:
  - Supports PC/SC
  - Supports CT-API (through wrapper on top of PC/SC)
- Built-in Peripherals:
  - Two user-controllable LEDs (Blue and Green)
  - User-controllable buzzer
- USB Firmware Upgradability
- Supports Android™ 3.1 and later[1]
- Compliant with the following standards:
  - ISO 14443
  - ISO 15693
  - ISO 7816
  - PC/SC
  - CCID
  - CE
  - UKCA
  - FCC
  - RoHS
  - REACH
  - Microsoft® WHQL

[1] *Uses an ACS-defined Android Library*

3.0. ACR1581U Architecture

3.1. Reader Block Diagram

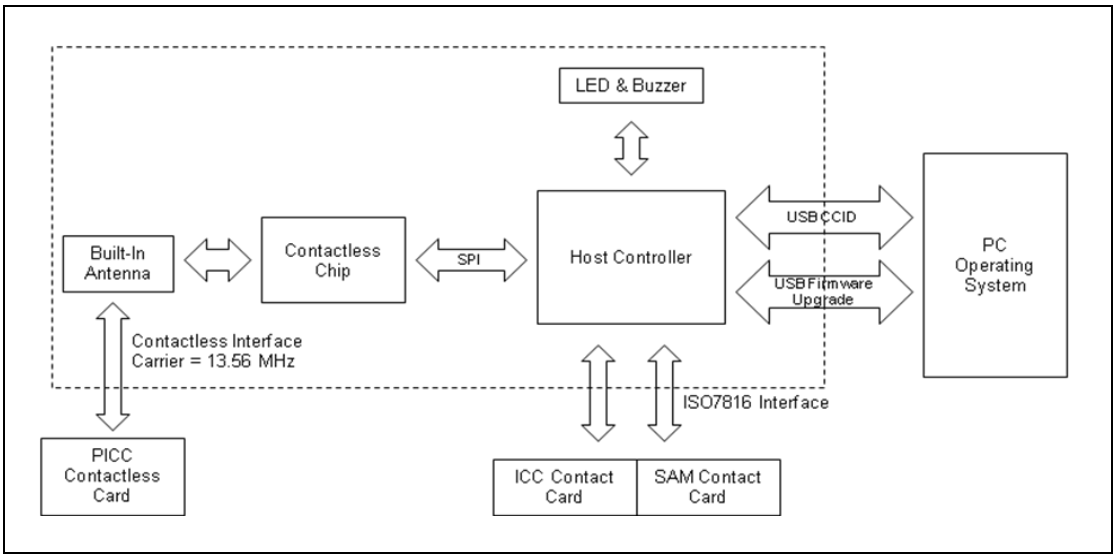


Figure 1: ACR1581U Reader Block Diagram

3.2. Communication between PC/SC driver and ICC, PICC and SAM

The protocol being used between the ACR1581U and the PC is CCID. All communications between ICC, PICC and SAM are PC/SC-compliant.

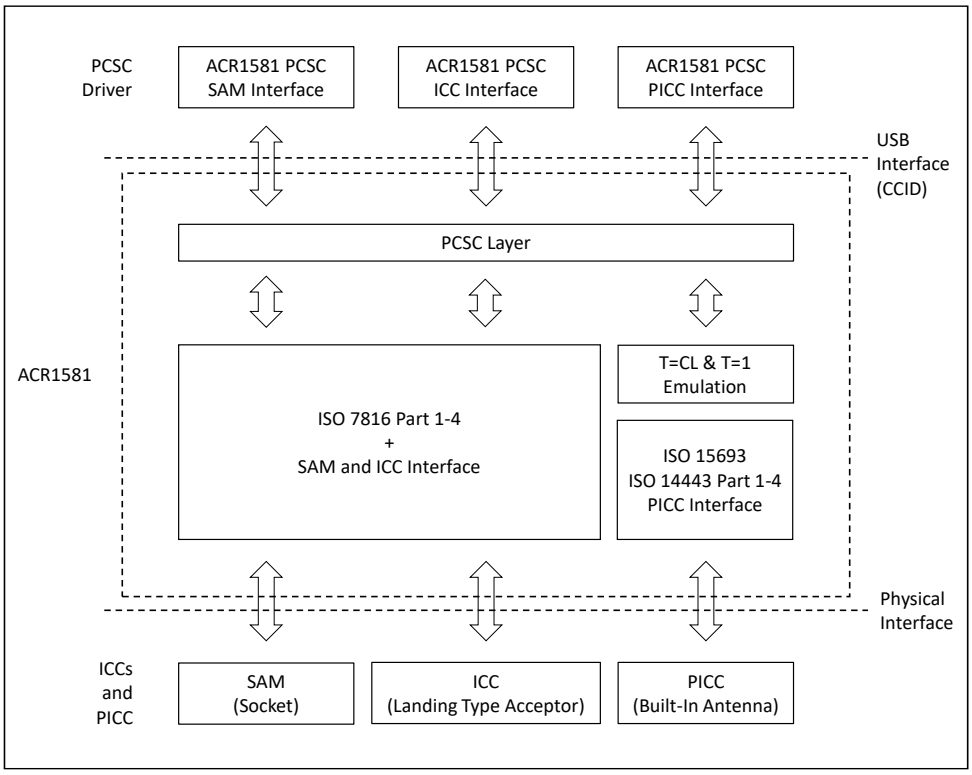


Figure 2: ACR1581U Architecture

4.0. Hardware Design

4.1. USB

The ACR1581U connects to a computer through USB following the USB standard.

4.1.1. Communication Parameters

The ACR1581U connects to a computer through USB as specified in the USB Specification 2.0. The ACR1581U works in full-speed mode, i.e. 12 Mbps.

Pin	Signal	Function
1	V_{BUS}	+5 V power supply for the reader
2	D-	Differential signal transmits data between ACR1581U-C1 and PC
3	D+	Differential signal transmits data between ACR1581U-C1 and PC
4	GND	Reference voltage level for power supply

Table 1: USB Interface Wiring

Note: *The device driver should be installed for the ACR1581U to function properly through USB interface.*

4.1.2. Endpoints

The ACR1581U uses the following endpoints to communicate with the host computer:

Control Endpoint – For setup and control purposes.

Bulk-OUT – For commands to be sent from the host to the ACR1581U (data packet size is 64 bytes).

Bulk-IN – For response to be sent from the ACR1581U to the host (data packet size is 64 bytes).

Interrupt-IN – For card status message to be sent from the ACR1581U to the host (data packet size is 8 bytes).

4.2. Contact Smart Card Interface

The interface between the ACR1581U and the inserted smart card follows the specifications of ISO 7816-3 with certain restrictions or enhancements to increase the practical functionality of the ACR1581U.

4.2.1. Smart Card Power Supply VCC (C1)

The current consumption of the inserted card must not be higher than 60 mA.

4.2.2. Card Type Selection

Before activating the inserted card, the controlling PC always needs to select the card type through the proper command sent to the ACR1581U. This includes both memory card and MCU-based cards.

For MCU-based cards the reader allows for the selection of the preferred protocol, T=0 or T=1. However, this selection is only accepted and carried out by the reader through the PPS if the card inserted in the reader supports both protocol types. Whenever an MCU-based card supports only one protocol type, T=0 or T=1, the reader automatically uses that protocol type, regardless of the protocol type selected by the application.

4.2.3. Interface for Microcontroller-based Cards

For microcontroller-based smart cards only the contacts C1 (VCC), C2 (RST), C3 (CLK), C5 (GND) and C7 (I/O) are used. A frequency of 5 MHz is applied to the CLK signal (C3).

4.3. Contactless Smart Card Interface

The interface between the ACR1581U and the contactless card follows the specifications of ISO 14443 with certain restrictions or enhancements to increase the practical functionality of the ACR1581U.

4.3.1. Carrier Frequency

The carrier frequency for the ACR1581U-C1 is 13.56 MHz.

4.3.2. Card Polling

The ACR1581U-C1 automatically polls the contactless cards that are within the field. ISO 14443-4 Type A, ISO 14443-4 Type B, ISO 15693 and MIFARE cards are supported.

4.4. User Interface

4.4.1. Buzzer and LED

The monotone buzzer and LEDs used for showing the state of the contact and contactless interfaces. The Blue LED is used for showing PICC status and Green LED for ICC.

Reader States	Buzzer	Green LED (ICC)	Blue LED (PICC)
1. Plug in the reader	Beep Once	● >> ● >> ●	
2. Standby (Contactless Polling, no ICC and PICC card)	Off	Off	●
3. Standby (No Polling, no ICC and PICC card)	Off	Off	Off
4. Contactless Card is tapped	Beep Once	Based on ICC status	●
5. Contactless Card is presence	Off	Based on ICC status	●
6. Contactless Card is removed	Off	Based on ICC status	Standby / Based on ICC status
7. Contactless Card is communicating	Off	Based on ICC status	Fast Blinking
8. Contact Card is inserted	Beep Once	●	Off / Based on PICC status
9. Contact Card is presence	Off	●	Off / Based on PICC status
10. Contact Card is removed	Off	Off	●
11. Contact Card is communicating	Off	Fast Blinking	Based on PICC status

Table 2: Buzzer and LED Indicator

5.0. Software Design

5.1. Contact Smart Card Protocol

5.1.1. Memory Card – 1/2/4/8/16 kb I2C Card

5.1.1.1. Select Card Type

The command is used to power down/up the selected card in the reader and performs a card reset afterword.

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FFh	A4h	00h	00h	01h	01h

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

5.1.1.2. Select Page Size

This command chooses the page size to read in the card. The default value is an 8-byte page write. It resets to the default value whenever the card is removed or the reader is turned off.

Command

Command	Class	INS	P1	P2	Lc	Data out
Select Page Size	FFh	01h	00h	00h	01h	Page Size

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

Page Size: 1 Byte

Status	Description
03h	8-byte page write
04h	16-byte page write
05h	32-byte page write
06h	64-byte page write
07h	128-byte page write

5.1.1.3. Read Memory Card

The command is used to read the memory card's content from a specified address.

Command

Command	Class	INS	P1	P2	Le
Read Memory Card	FFh	B0h	Memory Address		Length

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

5.1.1.4. Write Memory Card

The command is used to write the memory card's content from a specified address.

Command

Command	Class	INS	P1	P2	Le	Data In
Write Memory Card	FFh	D0h	Memory Address		Length	Data

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

5.1.2. Memory Card – 32/64/128/256/512/1024 kb I2C Card

5.1.2.1. Select Card Type

The command is used to power down/up the selected card in the reader and performs a card reset afterword.

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FFh	A4h	00h	00h	01h	02h

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

5.1.2.2. Select Page Size

This command chooses the page size to read in the card. The default value is an 8-byte page write. It resets to the default value whenever the card is removed or the reader is turned off.

Command

Command	Class	INS	P1	P2	Lc	Data out
Select Page Size	FFh	01h	00h	00h	01h	Page Size

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

Page Size: 1 Byte

Status	Description
03h	8-byte page write
04h	16-byte page write
05h	32-byte page write
06h	64-byte page write
07h	128-byte page write

5.1.2.3. Read Memory Card

The command is used to read the memory card's content from a specified address.

Command

Command	Class	INS	P1	P2	Le
Read Memory Card	FFh		Memory Address		Length

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

Note: INS *B0h = For 32, 64, 128, 256, 512 kb I2C card*

*1011 000*b; where * is the MSB of the 17 bit addressing = For 1024 kb I2C card*

5.1.2.4. Write Memory Card

The command is used to write the memory card's content from a specified address.

Command

Command	Class	INS	P1	P2	Le	Data In
Write Memory Card	FFh		Memory Address		Length	Data

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.

Note: INS *B0h = For 32, 64, 128, 256, 512 kb I2C card*

*1011 000*b; where * is the MSB of the 17 bit addressing = For 1024 kb I2C card*

5.1.3. Memory Card – ATMEL AT88SC153

5.1.3.1. Select card type

This command powers down/up the selected card inserted in the card reader and performs a card reset. It will also select the page size to be an 8-byte page write.

Command

Pseudo-APDU						
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**Card Type**
Select Card Type	FFh	A4h	00h	00h	01h	03h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.3.2. Read memory card

This command will read the Memory Card's Content from specified address.

Command

Pseudo-APDU					
Command	**Class**	**INS**	**P1**	**Byte Address**	**MEM_L**
Read Memory Card	FFh		00h		

Where:

INS (1 byte)
For reading zone 00b, INS = B0h
For reading zone 01b, INS = B1h
For reading zone 10b, INS = B2h
For reading zone 11b, INS = B3h
For reading fuse, INS = B4h

Byte Address (1 byte)
Memory address location of the memory card.

MEM_L (1 byte)
Length of data to be read from the memory card.

Response

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

Byte (1…N) Data read from memory card.

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.3.3. Write memory card

This command writes the memory card's content from a specified address.

Command

Pseudo-APDU									
Command	Class	INS	P1	Byte Address	MEM_L	Byte 1	…	…	Byte N
Write Memory Card	FFh		00h						

Where:

INS (1 byte)
For reading zone 00b, INS = D0h
For reading zone 01b, INS = D1h
For reading zone 10b, INS = D2h
For reading zone 11b, INS = D3h
For reading fuse, INS = D4h

Byte Address (1 byte)
Memory address location of the memory card.

MEM_L (1 byte)
Length of data to be written to the memory card

Byte (1…N) Data to be written to the memory card.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.3.4. Verify password

This command verifies whether the memory card's password matches the user's entered PIN.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**RP**	**PW (0)**	**PW (1)**	**PW (2)**
Verify Password	FFh	20h	00h		03h				

Where:

PW (0), PW (1), PW (2) = Password to be sent to memory card.

P2 (1 Byte)

= 0000 00r pb

Where the two bits "r p" indicates the password to compare

r = 0: Write password,

r = 1: Read password,

p = Password set number

r p = 01b for the secure code.

Response

Response	Data Out	
Result	SW1	ErrorCnt

Where:

SW1 = 90h

ErrorCnt (1 byte)

= Error Counter

FFh indicates the verification is correct. 00h indicates the password is locked (exceeded maximum number of retries). Other values indicate the current verification failed.

5.1.3.5. Initialize authentication

This command initializes the memory card's authentication.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**Q (0)**	**Q (1)**	**…**	**Q (7)**
Initialize Authentication	FFh	84h	00h	00h	08h				

Where:

Q (0…7) (8 bytes)

= Host random number

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.3.6. Verify authentication

This command verifies the memory card's authentication.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**Ch (0)**	**Ch (1)**	**…**	**Ch (7)**
Verify Authentication	FFh	82h	00h	00h	08h				

Where:

Ch (0…7) (8 bytes)

= Host challenge

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.4. Memory Card – ATMEL AT88SC1608

5.1.4.1. Select card type

This command powers down/up the selected card inserted in the card reader and performs a card reset. It also selects the page size to be a 16-byte page write.

Command

Pseudo-APDU						
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**Card Type**
Select Card Type	FFh	A4h	00h	00h	01h	04h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.4.2. Read memory card

This command reads the memory card's content from a specified address.

Command

Pseudo-APDU					
Command	**Class**	**INS**	**Zone Address**	**Byte Address**	**MEM_L**
Read Memory Card	FFh				

Where:

INS (1 byte)

For reading user zone, INS = B0h

For reading configuration zone or reading fuse, INS = B1h

Zone Address (1 byte)

= 00000 A10 A9 A8b, where A10 is the MSB of zone address

** don't care for reading fuse

Byte Address (1 byte)

= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

For reading fuse, Byte Address = 1000 0000b

MEM_L (1 byte)

Length of data to be read from the memory card.

Response

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

Byte (1…N) Data read from memory card.

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.4.3. Write to memory card

This command writes the memory card's content from a specified address.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**Zone Address**	**Byte Address**	**MEM_L**	**Byte 1**	**…**	**…**	**Byte N**
Write Memory Card	FFh								

Where:

INS (1 byte)

For reading user zone, **INS = D0h**

For reading configuration zone or reading fuse, **INS = D1h**

Zone Address (1 byte)

= 00000 A10 A9 A8b, where A10 is the MSB of zone address

** don't care for reading fuse

Byte Address (1 byte)

= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

For reading fuse, Byte Address = 1000 0000b

MEM_L (1 byte)

Length of data to be written to the memory card

Byte (1…N) Data to be written to the memory card.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.4.4. Verify password

This command verifies if the memory card's password matches the user's entered PIN.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**RP**	**PW (0)**	**PW (1)**	**PW (2)**
Verify Password	FFh	20h	00h	00h	04h				

Where:

PW (0), PW (1), PW (2) = Password to be sent to memory card.

RP (1 byte)

= 0000 r p2 p1 p0b

Where the two bits "r p2 p1 p0" indicate the password to compare

r = 0 : Write password,

r = 1: Read password,

p2 p1 p0 = Password set number

r p2 p1 p0 = 0111b for the secure code.

Response

Response	Data Out	
Result	SW1	ErrorCnt

Where:

SW1 = 90h

ErrorCnt (1 byte)

= Error Counter

FFh indicates the verification is correct. 00h indicates the password is locked (exceeded maximum number of retries). Other values indicate the current verification failed.

5.1.4.5. Initialize authentication

This command initializes the memory card's authentication.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**Q (0)**	**Q (1)**	**…**	**Q (7)**
Initialize Authentication	FFh	84h	00h	00h	08h				

Where:

Q (0…7) (8 bytes)

= Host random number

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.4.6. Verify authentication

This command verifies the memory card's authentication.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**Ch (0)**	**Ch (1)**	**…**	**Ch (7)**
Verify Authentication	FFh	82h	00h	00h	08h				

Where:

Ch (0…7) (8 bytes)

= Host challenge

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.5. Memory Card – SLE4418/SLE4428/SLE5518/SLE5528

5.1.5.1. Select card type

This command powers down/up the selected card in the reader, and then performs a card reset after.

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FFh	A4h	00h	00h	01h	05h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.5.2. Read memory card

This command reads the memory card's content from a specified address.

Command

Command	Class	INS	Byte Address		MEM_L
			MSB	LSB	
Read Memory Card	FFh	B0h			

Where:

MSB Byte Address (1 byte)

= 0000 00 A9 A8b is the memory address location of the memory card

LSB Byte Address (1 byte)

= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

MEM_L (1 byte)

Length of data to be read from the memory card

Response

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

Byte (1…N) Data read from memory card.

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.5.3. Read presentation error counter memory card (for SLE4428 and SLE5528 only)

This command reads the presentation error counter for the secret code.

Command

Command	Class	INS	P1	P2	MEM_L
Read Presentation Error Counter	FFh	B1h	00h	00h	03h

Response

Response	ErrCnt	Dummy 1	Dummy 2	SW1	SW2
Result					

Where:

ErrCnt (1 byte)

The value of the presentation error counter

FFh = indicates the verification is correct

00h = indicates the password is locked (exceeding the maximum number of retries)

Other values indicate the verification failed.

Dummy 1, Dummy 2 (2 bytes)

Dummy data read from the card

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.5.4. Read protection bit

This command reads the protection bit.

Command

Command	Class	INS	Byte Address		MEM_L
			MSB	LSB	
Read Protection Bit	FFh	B2h			

Where:

MSB Byte Address (1 byte)

The memory address location of the memory card

= 0000 00 A9 A8b

LSB Byte Address (1 byte)

The memory address location of the memory card

= A7 A6 A5 A4 A3 A2 A1 A0b

MEM_L (1 byte)

Length of protection bits read from the card, in multiples of 8 bits. The maximum value is 32.

MEM_L = 1 + INT ((number of bits – 1)/8)

For example, to read 8 protection bits starting from memory 0010h, the following pseudo-APDU should be issued:

FF B1 00 10 01h

Response

Response	PROT 1	…	…	PROT L	SW1	SW2
Result						

Where:

PROT (1..L) Bytes containing the protection bits.

SW1 SW2 = 90 00h if the operation was completed successfully.

The arrangement of the protection bits in the PROT bytes is as follows:

PROT 1								PROT 2								….							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

Where:

Px is the protection bit of byte *x* in response data:

0 = byte is write protected

1 = byte can be written

5.1.5.5. Write memory card

This command writes the memory card's content to a specified address.

Command

Command	Class	INS	Byte Address		MEM_L	Byte 1	…	…	Byte N
			MSB	LSB					
Write Memory Card	FFh	D0h							

Where:

MSB Byte Address (1 byte)

= 0000 00 A9 A8b is the memory address location of the memory card

LSB Byte Address (1 byte)

= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

MEM_L (1 byte)

Length of data to be written to the memory card

Byte (1…N) Data to be written to the memory card.

5.1.5.6. Write protection memory card

Each byte specified in the command is compared with the bytes stored in the specific address, and if the data matches, the corresponding protection bit is irreversibly programmed to '0'.

Command

Command	Class	INS	Byte Address		MEM_L	Byte 1	…	…	Byte N
			MSB	LSB					
Write Protection Memory Card	FFh	D1h							

Where:

MSB Byte Address (1 byte)

= 0000 00 A9 A8b is the memory address location of the memory card

LSB Byte Address (1 byte)

= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

MEM_L (1 byte)

=Length of data to be written to the memory card

Byte (1…N)

=Byte values compared with the data in the card starting at the Byte Address. Byte 1 is compared with the data at Byte Address; Byte N is compared with the data at Byte Address + N – 1.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.5.7. Present code memory card (for SLE44428 and SLE5528 only)

This command submits the secret code to the memory card to enable the write operation with the SLE4428 and SLE5528 cards. The following actions are executed:

1. Search a '1' bit in the presentation error counter and write the bit '0'.
2. Present the specified code to the card.
3. Try to erase the presentation error counter.

Command

Command	Class	INS	P1	P2	MEM_L	Code	
						Byte 1	Byte 2
Present Code Memory Card	FFh	20h	00h	00h	02h		

Where:

Code (3 bytes)

Secret code (PIN)

Response

Response	Data Out	
Result	90h	ErrorCnt

Where:

ErrorCnt (1 byte)

Error Counter

FFh = indicates the verification is correct.

00h = indicates the password is locked (exceeding maximum number of retries).

Other values indicate the verification failed.

5.1.6. Memory Card – SLE4432/SLE4442/SLE5532/SLE5542

5.1.6.1. Select card type

This command powers down/up the selected card in the reader, and then performs a card reset after.

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FFh	A4h	00h	00h	01h	06h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully

5.1.6.2. Read memory card

This command reads the memory card's content from a specified address.

Command

Command	Class	INS	P1	Byte Address	MEM_L
Read Memory Card	FFh	B0h	00h		

Where:

Byte Address (1 byte)

=A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

MEM_L (1 byte)

Length of data to be read from the memory card

Response

Response	Byte 1	…	…	Byte N	PROT1	PROT2	PROT3	PROT4	SW1	SW2
Result										

Where:

Byte (1…N) Data read from memory card.

PROT (1…4) Bytes containing the protections bits from protection.

SW1 SW2 = 90 00h if the operation was completed successfully.

The arrangement of the protection bits in the PROT bytes is as follows:

PROT 1								PROT 2								….							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

Where:

Px is the protection bit of byte *x* in response data:

0 = byte is write protected

1 = byte can be written

5.1.6.3. Read presentation error counter memory card (for SLE4442 and SLE5542 only)

This command reads the presentation error counter for the secret code.

Command

Command	Class	INS	P1	P2	MEM_L
Read Presentation Error Counter	FFh	B1h	00h	00h	04h

Response

Response	ErrCnt	Dummy 1	Dummy 2	Dummy 3	SW1	SW2
Result						

Where:

ErrCnt (1 byte)

The value of the presentation error counter

07h = indicates the verification is correct.

00h = indicates the password is locked (exceeded the maximum number of retries).

Other values indicate the verification failed.

Dummy 1, Dummy 2, Dummy 3 (3 bytes)

Dummy data read from the card

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.6.4. Read Protection Bit

This command reads the protection bits for the first 32 bytes.

Command

Command	Class	INS	P1	P2	MEM_L
Read Protection Bit	FFh	B2h	00h	00h	04h

Response

Response	PROT 1	PROT 2	PROT 3	PROT 4	SW1	SW2
Result						

Where:

PROT (1..4) Bytes containing the protection bits.

SW1 SW2 = 90 00h if the operation was completed successfully.

The arrangement of the protection bits in the PROT bytes is as follows:

PROT 1								PROT 2								….							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

Where:

Px protection bit of bytes in the response data:

0 = byte is write protected

1 = byte can be written

5.1.6.5. Write memory card

This command writes the memory card's content to a specified address.

Command

Command	Class	INS	P1	Byte Address	MEM_L	Byte 1	…	…	Byte N
Write Memory Card	FFh	D0h	00h						

Where:

Byte Address (1 byte)

= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

MEM_L (1 byte)

Length of data to be written to the memory card

Byte (1…N) Data to be written to the memory card.

Response

Response	Data Out	
Result	SW1	SW2

Where: **SW1 SW2** = 90 00h if the operation was completed successfully.

5.1.6.6. Write protection memory card

Each byte specified in the command is compared with the bytes stored in the specific address and if the data matches, the corresponding protection bit is irreversibly programmed to '0'.

Command

Command	Class	INS	P1	Byte Address	MEM_L	Byte 1	…	…	Byte N
Write Protection Memory Card	FFh	D1h	00h						

Where:

Byte Address (1 byte)

= 000A4 A3 A2 A1b (00h – 1Fh) is the protection memory address location of the memory card

MEM_L (1 byte)

Length of data to be written to the memory card

Byte (1…N) Byte values compared with the data in the card starting at the Byte Address. Byte 1 is compared with the data at Byte Address; Byte N is compared with the data at Byte Address + N – 1.

Response

Response	Data Out	
Result	SW1	SW2

Where: **SW1 SW2** = 90 00h if the operation was completed successfully.

5.1.6.7. Present code memory card (for SLE4442 and SLE5542 only)

This command submits the secret code to the memory card to enable the write operation with the SLE4442 and SLE5542 card. The following actions are executed:

1. Search a '1' bit in the presentation error counter and write bit '0'.
2. Present the specified code to the card.
3. Try to erase the presentation error counter.

Command

Command	Class	INS	P1	P2	MEM_L	Code		
						Byte 1	Byte 2	Byte 3
Present Code Memory Card	FFh	20h	00h	00h	03h			

Where:

Code (3 bytes)

Secret code (PIN)

Response

Response	Data Out	
Result	SW1	ErrorCnt

Where:

ErrorCnt (1 byte)

Error Counter

07h = indicates the verification is correct.

00h = indicates the password is locked (exceeded the maximum number of retries).

Other values indicate the verification failed.

5.1.6.8. Change code memory card (for SLE4442 and SLE5542 only)

This command writes the specified data as the new secret code in the card. The existing secret code must be presented to the card using the "Present Code" command prior to the execution of this command.

Command

Command	Class	INS	P1	P2	MEM_L	Code		
						Byte 1	Byte 2	Byte 3
Change Code Memory Card	FFh	D2h	00h	01h	03h			

Where:

Code (3 bytes)

Secret code (PIN)

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.7. Memory Card – SLE4406/SLE4436/SLE5536/SLE6636

5.1.7.1. Select card type

This command powers down/up the selected card in the reader, and then performs a card reset after.

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FFh	A4h	00h	00h	01h	07h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.7.2. Read Memory Card

This command reads the memory card's content from a specified address.

Command

Command	Class	INS	P1	Byte Address	MEM_L
Read Memory Card	FFh	B0h	00h		

Where:

Byte Address (1 byte)
Memory address location of the memory card

MEM_L (1 byte)
Length of data to be read from the memory card

Response

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

Byte (1…N) Data read from memory card.

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.7.3. Write one byte memory card

This command is used to write one byte to the specified address of the inserted card. The byte is written to the card with LSB first, i.e. the bit card address 0 is regarded as the LSB of byte 0.

Four different *write* modes are available for this card type, which are distinguished by a flag in the command data field:

a. Write

The byte value specified in the command is written to the specified address. This command can be used for writing personalization data and counter values to the card.

b. Write with carry

The byte value specified in the command is written to the specified address and the command is sent to the card to erase the next lower counter stage. This mode can therefore only be used for updating the counter value in the card.

c. Write with backup enabled (for SLE4436, SLE5536 and SLE6636 only)

The byte value specified in the command is written to the specified address. This command can be used for writing personalization data and counter values to the card. Backup bit is enabled to prevent data loss when card tearing occurs.

d. Write with carry and backup enabled (SLE4436, SLE5536 and SLE6636 only)

The byte value specified in the command is written to the specified address and the command is sent to the card to erase the next lower counter stage. This mode can therefore only be used for updating the counter value in the card. Backup bit is enabled to prevent data loss when card tearing occurs.

With all write modes, the byte at the specified card address is not erased prior to the write operation and hence, memory bits can only be programmed from '1' to '0'.
The backup mode available in the SLE4436 and SLE5536 card can be enabled or disabled in the write operation.

Command

Command	Class	INS	P1	Byte Address	MEM_L	Mode	Byte
Read Memory Card	FFh	D0h	00h		02h		

Where:

Byte Address (1 byte)
Memory address location of the memory card

Mode (1 byte)
Specifies the write mode and backup option
00h = Write.
01h = Write with carry.
02h = Write with backup enabled (for SLE4436, SLE5536 and SLE6636 only).
03h = Write with carry and with backup enabled (for SLE4436, SLE5536 and SLE6636 only).

Byte (1 byte)
Byte value to be written to the card

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.7.4. Present code memory card

This command submits the secret code to the memory card to enable card personalization mode. The following actions are executed:

1. Search a '1' bit in the presentation error counter and write bit '0'.
2. Present the specified code to the card.

Command

Command	Class	INS	P1	P2	MEM_L	Code			
						Addr	Byte 1	Byte 2	Byte 3
Present Code Memory Card	FFh	20h	00h	00h	04h	09h			

Where:

Addr (1 byte)

Byte address of the presentation counter in the card

Code (3 bytes)

Secret code (PIN)

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.7.5. Authenticate memory card (for SLE4436, SLE5536 and SLE6636 only)

This command reads the authentication certificate from the card. The following actions are executed:

1. Select Key 1 or Key 2 in the card as specified in the command.
2. Present the challenge data specified in the command to the card.
3. Generate the specified number of CLK pulses for each bit authentication data computed by the card.
4. Read 16 bits of authentication data from the card.
5. Reset the card to normal operation mode.

The authentication is performed in two steps. The first step is to send the Authentication Certificate to the card. The second step is to get back two bytes of authentication data calculated by the card.

Step 1: Send authentication certificate to the card.

Command

Command	Class	INS	P1	P2	MEM_L	Code				
						Key	CLK_CNT	Byte 1	…	Byte 6
Send Authentication Certificate	FFh	84h	00h	00h	08h					

Where:

Key (1 byte)

Key to be used for the computation of the authentication certificate

00h = Key 1 with no cipher block chaining.

01h = Key 2 with no cipher block chaining.

80h = Key 1 with cipher block chaining (for SLL5536 and SLE6636 only).

81h = Key 2 with cipher block chaining (for SLL5536 and SLE6636 only).

CLK_CNT (1 byte)

Number of CLK pulses to be supplied to the card for the computation of each bit of the authentication certificate. Typical value is 160 clocks (A0h).

Byte (1...6) Card challenge data.

Response

Response	SW1	SW2
Result	61h	02h

Step 2: Get the authentication data (Get Response).

Command

Command	Class	INS	P1	P2	MEM_L
Get Authentication Data	FFh	C0h	00h	00h	02h

Response

Response	Cert		SW1	SW2
Result				

Where:

Cert (2 bytes)

16 bits of authentication data computed by the card. The LSB of Byte 1 is the first authentication bit read from the card.

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.8. Memory Card – SLE4404

5.1.8.1. Select card type

This command powers down/up the selected card in the reader, and then performs a card reset after.

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FFh	A4h	00h	00h	01h	08h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.8.2. Read memory card

This command reads the memory card's content from a specified address.

Command

Command	Class	INS	P1	Byte Address	MEM_L
Read Memory Card	FFh	B0h	00h		

Where:

Byte Address (1 byte)
Memory address location of the memory card

MEM_L (1 byte)
Length of data to be read from the memory card

Response

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

Byte (1…N) Data read from memory card.

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.8.3. Write memory card

This command writes the memory card's content to a specified address. The byte is written to the card with LSB first, i.e. the bit at card address 0 is regarded as the LSB of byte 0.

The byte at the specified card address is not erased prior to the write operation and hence, memory bits can only be programmed from '1' to '0'.

Command

Command	Class	INS	P1	Byte Address	MEM_L	Byte 1	...	...	Byte N
Write Memory Card	FFh	D0h	00h						

Where:

Byte Address (1 byte)

Memory address location of the memory card

MEM_L (1 byte)

Length of data to be written to the memory card

Byte (1…N) Data to be written to the memory card.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.8.4. Erase scratch pad memory card

This command erases the data of the scratch pad memory of the inserted card. All memory bits inside the scratch pad memory will be programmed to the state of '1'.

Command

Command	Class	INS	P1	Byte Address	MEM_L
Erase Scratch Pad	FFh	D2h	00h		00h

Where:

Byte Address (1 byte)

Memory byte address location of the scratch pad. Typical value is 02h.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.8.5. Verify user code

This command submits the User Code (2 bytes) to the inserted card. The User Code enables access to the memory of the card.

The following actions are executed:

1. Present the specified code to the card.
2. Search a '1' bit in the presentation error counter and write the bit '0'.
3. Erase the presentation error counter. The Error User Counter can be erased when the submitted code is correct.

Command

Command	Class	INS	Error Counter LEN	Byte Address	MEM_L	Code	
						Byte 1	Byte 2
Verify User Code	FFh	20h	04h	08h	02h		

Where:

Error Counter LEN (1 byte)
Length of presentation error counter in bits

Byte Address (1 byte)
Byte address of the key in the card

Code (1 byte)
User Code

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.
= 63 00h if there are no more retries left.

Note: *After SW1 SW2 = 90 00h has been received, read back the User Error Counter to check whether the Verify_User_Code is correct. If the User Error Counter is erased and is equal to 'FFh', the previous verification was successful.*

5.1.8.6. Verify memory code

This command submits memory code (4 bytes) to the inserted card. The memory code is used to authorize the reloading of the user memory, together with the User Code.

The following actions are executed:

1. Present the specified code to the card.
2. Search a '1' bit in the presentation error counter and write the bit to '0'.
3. Erase the presentation error counter.

Note: *The Memory Error Counter cannot be erased.*

Command

Command	Class	INS	Error Counter LEN	Byte Address	MEM_L	Code			
						Byte 1	Byte 2	Byte 3	Byte 4
Verify Memory Code	FFh	20h	40h	28h	04h				

Where:

Error Counter LEN (1 byte)
Length of presentation error counter in bits

Byte Address (1 byte)
Byte address of the key in the card

Code (4 bytes)
Memory Code

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.
= 63 00h if there are no more retries left.

Note: *After SW1 SW2 = 90 00h has been received, read back the User Error Counter to check whether the Verify Memory Code is correct. If all data in Application Area is erased and is equal to 'FFh', the previous verification was successful.*

5.1.9. Memory Card – AT88SC101/AT88SC102/AT88SC1003

5.1.9.1. Select card type

This command powers down and up the selected card inserted in the card reader and performs a card reset.

Command

Pseudo-APDU						
Command	**Class**	**INS**	**P1**	**P2**	**Lc**	**Card Type**
Select Card Type	FFh	A4h	00h	00h	01h	09h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.9.2. Read Memory Card

This command reads the memory card's content from specified address.

Command

Pseudo-APDU					
Command	**Class**	**INS**	**P1**	**Byte Address**	**MEM_L**
Read Memory Card	FFh	B0h	00h		

Where:

Byte Address (1 byte)
Memory address location of the memory card.

MEM_L (1 byte)
Length of data to be read from the memory card.

Response

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

Byte (1…N) Data read from memory card.

SW1 SW2 = 90 00h if the operation was completed successfully.

5.1.9.3. Write Memory Card

This command writes data to the specified address of the inserted card. The byte is written to the card with LSB first, i.e., the bit at card address 0 is regarded as the LSB of byte 0.

The byte at the specified card address is not erased prior to the write operation and, hence, memory bits can only be programmed from '1' to '0'.

Command

Pseudo-APDU									
Command	**Class**	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**…**	**…**	**Byte N**
Write Memory Card	FFh	D0h	00h						

Where:

Byte Address (1 byte)

Memory address location of the memory card.

MEM_L (1 byte)

Length of data to be written to the memory card

Byte (1…N) Byte value to be written to the card.

Response

Response	Data Out	
Result	SW1	SW2

Where: **SW1 SW2** = 90 00h if the operation was completed successfully.

5.1.9.4. Erase non-application zone

This command erases the data in non-application zones. The EEPROM memory is organized into 16 bit words. Although erases are performed on single bits the ERASE operation clears an entire word in the memory. Therefore, performing an Erase on any bit in the word will clear All 16 bits of that word to the state of '1'.

To erase Error Counter or the data in Application Zones, please refer to:

- Erase Application Zone With Erase command as specified
- Erase Application Zone With Write and Erase command as specified
- Verify Security Code commands as specified

Command

Pseudo-APDU					
Command	**Class**	**INS**	**P1**	**Byte Address**	**MEM_L**
Erase Non-Application Zone	FFh	D2h	00h		00h

Where:

Byte Address (1 byte)

Memory byte address location of the word to be erased.

Response

Response	Data Out	
Result	SW1	SW2

Where: **SW1 SW2** = 90 00h if the operation was completed successfully.

5.1.9.5. Erase Application Zone with Erase

This command can be used in the following cases:

- AT88SC101: To erase the data in Application Zone with EC Function Disabled
- AT88SC102: To erase the data in Application Zone 1
- AT88SC102: To erase the data in Application Zone 2 with EC2 Function Disabled
- AT88SC1003: To erase the data in Application Zone 1
- AT88SC1003: To erase the data in Application Zone 2 with EC2 Function Disabled
- AT88SC1003: To erase the data in Application Zone 3

The following actions are executed for this command:

1. Present the specified code to the card.
2. Erase the presentation error counter. The data in corresponding Application Zone can be erased when the submitted code is correct.

Command

Pseudo-APDU										
Command	Class	INS	Error Counter LEN	Byte Address	MEM_L	CODE				
						Byte 1	Byte 2	…	…	Byte N
Erase Application Zone with Erase	FFh	20h	00h							

Where:

Error Counter LEN (1 byte)
= Length of presentation error counter in bits. The value should be 00h always.

Byte Address (1 byte)
= Byte address of the Application Zone Key in the card. Please refer to the table below for the correct value.

MEM_L (1 byte)
= Length of the Erase Key. Please refer to the table below for the correct value.

CODE (1…N) = Erase Key

Cases	Byte Address	LEN
AT88SC101: Erase Application Zone with EC function disabled	96h	04h
AT88SC102: Erase Application Zone 1	56h	06h
AT88SC102: Erase Application Zone 2 with EC2 function disabled	9Ch	04h
AT88SC1003: Erase Application Zone 1	36h	06h
AT88SC1003: Erase Application Zone 2 with EC2 function disabled	5Ch	04h
AT88SC1003: Erase Application Zone 3	C0h	06h

Response

Response	Data Out	
Result	SW1	SW2

Where: **SW1 SW2** = 90 00h if the operation was completed successfully.

Note: *After SW1SW2 = 90 00h been received, read back the data in Application Zone can check whether the Erase Application Zone with Erase is correct. If all data in Application Zone is erased and equals to "FFh", the previous verification was successful.*

5.1.9.6. Erase Application Zone with Write and Erase

This command can be used in the following cases:

- AT88SC101: To erase the data in Application Zone with EC Function Enabled
- AT88SC102: To erase the data in Application Zone 2 with EC2 Function Enabled
- AT88SC1003: To erase the data in Application Zone 2 with EC2 Function Enabled

With EC or EC2 Function Enabled (that is, ECEN or EC2EN Fuse is unblown and in "1" state), the following actions are executed:

1. Present the specified code to the card
2. Search a '1' bit in the presentation error counter and write the bit to '0'
3. Erase the presentation error counter. The data in corresponding Application Zone can be erased when the submitted code is correct.

Command

Pseudo-APDU									
Command	Class	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
						Byte 1	Byte 2	Byte 3	Byte 4
Erase Application Zone with Write and Erase	FFh	20h	80h		04h				

Where:

Error Counter LEN (1 byte)

= Length of presentation error counter in bits. The value should be 80h always.

Byte Address (1 byte)

= Byte address of the Application Zone Key in the card. Please refer to the table below for the correct value.

CODE (4 bytes)

= Erase Key

Cases	Byte Address
AT88SC101	96h
AT88SC102	9Ch
AT88SC1003	5Ch

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

= 63 00 if there are no more retries left.

Note: *After SW1SW2 = 90 00 has been received, read back the data in Application Zone can check whether the Erase Application Zone with Write and Erase is correct. If all data in Application Zone is erased and equals to "FFh", the previous verification was successful.*

5.1.9.7. Verify Security Code

This command submits Security Code (2 bytes) to the inserted card. Security Code is to enable the memory access of the card.

The following actions are executed:

1. Present the specified code to the card
2. Search a '1' bit in the presentation error counter and write the bit to '0'
3. Erase the presentation error counter. The Security Code Attempts Counter can be erased when the submitted code is correct.

Command

Pseudo-APDU							
Command	Class	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
						Byte 1	Byte 2
Verify Security Code	FFh	20h	08h	0Ah	02h		

Where:

Error Counter LEN (1 byte)

= Length of presentation error counter in bits.

Byte Address (1 byte)

= Byte address of the key in the card**.**

CODE (2 bytes)

= Security Code

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

= 63 00 if there are no more retries left.

Note: *After SW1SW2 = 90 00h been received, read back the Security Code Attempts Counter (SCAC) can check whether the Verify User Code is correct. If SCAC is erased and equals to "FFh", the previous verification was successful.*

5.1.9.8. Blow Fuse

This command blows the fuse of the inserted card. The fuse can be EC_EN Fuse, EC2EN Fuse, Issuer Fuse or Manufacturer's Fuse.

Note: *Blowing the fuse is an irreversible process.*

Command

Pseudo-APDU									
Command	Class	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
						Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of FUS Pin	State of RST Pin
Blown Fuse	FFh	05h	00h	00h	04h			01h	00h 01h

Where:

Fuse Bit Addr (2 bytes)

= Bit address of the fuse. Please refer to the table below for the correct value.

State of FUS Pin (1 byte)

= State of the FUS pin. Should be 01h always.

State of RST Pin (1 byte)

= State of the RST pin. Please refer to below table for the correct value.

		Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of RST Pin
AT88SC101	Manufacturer Fuse	05h	80h	01h
	EC_EN Fuse	05h	C9h	01h
	Issuer Fuse	05h	E0h	01h
AT88SC102	Manufacturer Fuse	05h	B0h	01h
	EC2EN Fuse	05h	F9h	01h
	Issuer Fuse	06h	10h	01h
AT88SC1003	Manufacturer Fuse	03h	F8h	00h
	EC2EN Fuse	03h	FCh	00h
	Issuer Fuse	03h	E0h	00h

Table 3: Blown Fuse Code Values

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h if the operation was completed successfully.

= 63 00 if there are no more retries left.

5.1.10. ACOS6-SAM Commands

This section contains SAM-specific commands.

***Note**: For complete information on ACOS6-SAM Commands and Scenarios, please contact an ACS representative for a copy of the ACOS6-SAM Reference Manual.*

5.1.10.1. Generate Key

This command is used to generate a diversified key to load into the ACOS3/6 card or other cards from deviation data such as a client card serial number. This command is catered for client card issuance purposes.

APDU	Description	
CLA	80h	
INS	88h	
P1	00h	Generate 8 Byte Key
	01h	Generate 16 Byte Key
	02h	Generate 24 Byte Key
P2	Key index of Master Key to generate Derived Key	
P3	08h	
Data	Input Data	

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3, must be 08h
6A 83h	Referenced key record not found in EF2
69 81h	Invalid EF2 (record size, file type, etc.)
6A 88h	EF2 not found
62 83h	Current DF is blocked; EF2 is blocked
69 83h	Usage counter is zero.
69 82h	Security condition not satisfied
6A 87h	Referenced Master Key is not capable of 3-DES encryption
61 08h	Command completed, issue GET RESPONSE to get the result

5.1.10.2. Diversify (or load) Key Data

This command prepares the SAM card to perform ciphering operations by diversifying and loading the key. It takes the serial number and CBC initial vector as command data input.

APDU	Description								
CLA	80h								
INS	72h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	Description
	-	0	0	0	0	0	0	1	Secret Code (Sc)
	-	0	0	0	0	0	1	0	Account Key (K_{ACCT})
	-	0	0	0	0	0	1	1	Terminal Key
	-	0	0	0	0	1	0	0	Card Key
	-	0	0	0	0	1	0	1	Bulk Encryption Key (Not diversified)
	-	0	0	0	0	1	1	0	Initial vector
	0	-	-	-	-	-	-	-	16-byte Key
	1	-	-	-	-	-	-	-	24-byte Key
P2	Index of Master Key: Bit7: 1 = local Key in current EF2; 0 = global KEY EF2 Bit6-Bit5: 00b - RFU Bit4-Bit0: Key Index								
P3	If P1 = 1-4, P3 = 8/16,(if algo is AES, P3 = 8/16) If P1 = 5, P3 = 0 If P1 = 6, P3 = 8 (Algo of Master Key is DES/ 3DES/ 3KDES) P3 = 16 (Algo of Master Key is AES)								
Data	If P1 = 1-4 Client card's Serial Number, (if algo is AES, Data is Client card's Serial Number or Client card's Serial Number append with "0000000000000000") If P1 = 5, No command data. If P1 = 6, DES/3DES/3KDES/AES CBC initial vector.								

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Wrong P1, P1 must be 1 to 6
67 00h	Wrong P3, P3 must be 8 (or 0)
62 83h	Current DF is blocked, or EF2 is blocked
69 82h	Security condition not satisfied
6A 88h	EF2 not found
6A 83h	Referenced Master Key in EF2 not found

SW1 SW2	Description
69 81h	Invalid EF2 (FDB, MRL, etc., not consistent)
6A 87h	Referenced KEY not capable of authentication
69 83h	Referenced Key is locked
90 00h	Target key generated, and ready in SAM memory

5.1.10.3. Encrypt

This command is used to encrypt data using DES or 3DES with either:

1. The session key created by the mutual authentication procedure with an ACOS3/6, DESFire®, DESFire® EV1 or MIFARE Plus card.
2. A diversified key (secret code).
3. A bulk encryption key.
4. Encrypt the diversified secret code with the session key.
5. Prepare ACOS3 secure messaging command given a non-SM command.

APDU	Description								
CLA	80h								
INS	74h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	Description
	-	0	0	0	0	0	0	-	ECB Mode
	-	0	0	0	0	0	1	-	CBC Mode
	-	0	0	0	0	1	0	-	Retail MAC Mode
	-	0	0	0	0	1	1	-	MAC Mode
	-	0	0	0	1	0	0	-	Prepare ACOS3 SM command.
	-	1	0	0	1	0	1	-	MIFARE DESFire Encryption
	-	1	0	0	1	1	0	-	MIFARE DESFire EV1 Encryption
	-	0	0	0	1	1	1	-	CMAC
	-	0	1	0	0	0	0		MIFARE Plus Command
	-	0	1	0	0	0	1		MIFARE Plus Response
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES
	1	-	-	-	-	-	-	1	AES
	-	-	-	-	-	-	-	-	All other values – RFU

APDU	Description
P2	P2 is derived key in SAM set using Load Key function: 1 – Encrypt Data with Session Key *Ks* 2 – Encrypt Data with Diversified Key *Sc* 3 – Encrypt Data with Bulk Encryption Key 0 – return ENC (*Sc*, *Ks*) If P1.b3 = 1 or b5=1, P2 must be 1 If P2 = 0h, P1 can be either 0 or 1
P3	P3 < 128 If bit 3 of P1 not equal to 1 and bit 5 of P1 not equal to 1 - If P2 = 1-3, multiple of 8 (DES/3DES/3KDES) or 16 (AES) up to 128 bytes - If P2 = 0, 0
Data	Plain text If P2 b6 = 1, The DATA format should be: • Length of Plain text data • Length of Command and Header of DESFire Card • Command and Header of DESFire Card • Plain text P1 = `A1h`, the encryption is for a MIFARE Plus command • if MFP Command is *value* operations command, the DATA format should be Command code(1 BYTE)+BlockNum(2/4 BYTE)+Value(4 BYTE). • if MFP Command is *Proximity Check*, the DATA format should be Command code(1 BYTE)+ PPS1(1 BYTE). • if MFP Command is *Read*, the DATA format should be Command code(1 BYTE)+ BlockNum(2 BYTE) • if MFP Command is *Write*, the DATA format should be Command code(1 BYTE)+ BlockNum(2 BYTE) +plaintext P1=`A3h`, • The data return by ICC (don't include SC code and don't include RMAC if RMAC exist)

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3
6A 83h	ACOS Target Key is not ready (use Diversify to generate the key)
61 XX	Encryption is done, use GET RESPONSE to get the result

5.1.10.4. Decrypt

This command is used to decrypt data using DES or 3DES or AES with either:

1. The session key created by the mutual authentication procedure with an ACOS3/6, MIFARE DESFire, MIFARE DESFire EV1 or MIFARE Plus card.
2. A diversified key (secret code).
3. A bulk encryption key.
4. Decrypt the diversified secret code with the session key.
5. Verify and Decrypt ACOS3 secure-messaging response.

Verify and Decrypt ACOS3 SM Response:

APDU	Description								
CLA	80h								
INS	76h								
P1	b7	b6	b5	b4	b3	b2	b1	b0	Description
	-	0	0	0	0	0	0	-	ECB Mode
	-	0	0	0	0	0	1	-	CBC Mode
	-	0	0	0	1	0	0	-	Verify and Decrypt ACOS3 SM Response
	-	1	0	0	1	0	1	-	MIFARE DESFire Decryption
	-	1	0	0	1	1	0	-	MIFARE DESFire EV1 Decryption
	-	0	1	0	0	1	0	-	MIFARE Plus Decryption
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES
	1	-	-	-	-	-	-	1	AES
	0	0	0	0	-	-	-	-	All other values - RFU
P2	P2 is derived key in SAM set using Load Key function: 1 – Decrypt Data with Session Key *Ks* 2 – Decrypt Data with Diversified Key *Sc* 3 – Decrypt Data with Bulk Encryption Key 0 – return DEC (*Sc*, *Ks*)								
P3	P3 < 128 If P1 = `A5h`, P3=16/32/48 If bit 3 of P1 not equal to 1 - If P2 = 1-3, multiple of 8 (DES/3DES/3KDES) or 16 (AES) up to 128 bytes - If P2 = 0, 0								
Data	Ciphertext If P1 = `A5h`, The DATA is Encrypted text If P2 b6 = 1, The DATA format should be: • Length of Plain text data, if unknown, use 00 • Length of Command and Header of DESFire Card • Command and Header of DESFire Card • Encrypted text								

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3
6A 83h	ACOS Target Key is not ready (use Diversify to generate the key)
61 XX	Decryption is done, use GET RESPONSE to get the result

5.1.10.5. Prepare Authentication

This command is used to authenticate the SAM card (as the terminal) to the ACOS 3/6 card or MIFARE Ultralight C/MIFARE DESFire Card/MIFARE Plus card.

APDU	Description
CLA	80h
INS	78h
P1	00h – 3DES 01h – DES 02h – 3KDES (MIFARE DESFire EV1/ACOS3) 03h – AES (MIFARE DESFire EV1/MIFARE Plus/ACOS3) 80h – 3DES (MIFARE DESFire Authenticate only) 81h – DES (MIFARE DESFire Authenticate only) Other – RFU
P2	0h – Verify ACOS3/6 Authenticate Return 01h – MIFARE Ultralight C/DESFire Authenticate by (Diversified) Terminal Key 05h – MIFARE Ultralight C/DESFire Authenticate by Bulk Encryption Key 02h – MIFARE Plus Authenticate. First Authenticate of SL1 to SL3 03h – MIFARE Plus Authenticate. Authentication in SL1 to SL2. 04h – MIFARE Plus Authenticate. Following Authenticate of SL2 to SL3.
P3	8 – (P1 = 00h, 01h, 02h, 80h, 81h) 16 – (P1 = 03h)
Data	Card Challenge Data

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3, must be 08h
6A 83h	ACOS Key (KT or KC) is not ready (use Diversify to generate this key)
69 82h	Security condition not satisfied
61 10h	Command completed, issue GET RESPONSE to get the result

5.1.10.6. Verify Authentication

This command is used to verify the ACOS 3/6, MIFARE Ultralight C, MIFARE DESFire/MIFARE DESFire EV1 or MIFARE Plus card to the terminal. The Session Key Ks would also be generated internally.

APDU	Description
CLA	80h
INS	7Ah
P1	00h – 3DES (P2 = 0) 01h – DES (P2 = 0) 02h – 3KDES (P2 = 0，ACOS3) 03h – AES (P2 = 0，ACOS3) Other – RFU
P2	00h – Verify ACOS3/6 Authenticate Return 01h – Verify MIFARE Ultralight C®/ DESFire®/ DESFire® EV1 Authenticate Return 02h – Verify MIFARE Plus Authenticate return
P3	08h – (P2 = 0, P2 = 1 and Session Key is DES/3DES) 16h – (P2 = 1 and Session Key is 3KDES/AES) 16h – (P2=02, and MIFARE Plus return data ek(RndA')) 32h – (P2=02, and MIFARE Plus return data ek(TI+PICCcap2+PCDcap2))
Data	ACOS 3/6: DES (K_S, RND_T) MIFARE DESFire/ DESFire EV1 return data: ek(RndA') MIFARE Plus return data ek(RndA') or ek(TI+PICCcap2+PCDcap2)

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3, must be 08h
6A 83h	ACOS-SAM Session Key or RND_T are not ready. Use PREPARE AUTHENTICATION to build these keys.
69 82h	Data is incorrect
90 00h	Data is correct, ACOS Mutual Authentication is successful

5.1.10.7. Verify ACOS Inquire Account

This command is used to verify the ACOS3/6 card's Inquire Account purse command. It would verify that the MAC checksum returned by ACOS3/6 are correct with the SAM's diversified key.

APDU	Description								
CLA	80h								
INS	7Ch								
P1	b7	b6	b5	b4	b3	b2	b1	b0	Description
	-	0	0	0	0	-	0	-	ACOS INQ_AUT is disabled
	-	0	0	0	0	-	1	-	ACOS INQ_AUT is enabled
	-	0	0	0	0	0	-	-	ACOS INQ_ACC_MAC is disabled
	-	0	0	0	0	1	-	-	ACOS INQ_ACC_MAC is enabled
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES (ACOS3 only)
	1	-	-	-	-	-	-	1	AES (ACOS3 only)
P2	0h								
P3	1Dh								
Data	Data Block returned by INQUIRE ACCOUNT of client ACOS card, see below.								

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3
6A 83h	ACOS Key K_S or K_{ACCT} are not ready; use DIVERSIFY command to generate K_{ACCT}; if applicable, use "Prepare Authentication" to generate K_S.
6F 00h	Data Block's MAC is incorrect
90 00h	Data Block's MAC is correct

5.1.10.8. Prepare ACOS Account Transaction

To create an ACOS3/6 Credit/Debit command, the MAC must be computed for ACOS3/6 to verify.

APDU	Description								
CLA	80h								
INS	7Eh								
P1	b7	b6	b5	b4	b3	b2	b1	b0	Description
	-	0	0	0	0	0	0	-	ACOS TRNS_AUT is disabled
	-	0	0	0	0	0	1	-	ACOS TRNS_AUT is enabled
	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES (ACOS3 only)
	1	-	-	-	-	-	-	1	AES (ACOS3 only)
P2	E2h: Credit E6h: Debit								
P3	0Dh								
Data	Data Block								

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3, must be 0Dh
6A 83h	ACOS Key K_S or K_{ACCT} are not ready; use DIVERSIFY command to generate K_{ACCT}; if applicable, use “Prepare Authentication” to generate K_S.
61 0Bh	Command completed, issue GET RESPONSE to get the result

5.1.10.9. Verify Debit Certificate

For ACOS3/6, if the DEBIT command has P1 = 1, a debit certificate is returned. The debit certificate can be checked by comparing the ACOS3 response to the result of this command.

APDU	Description
CLA	80h
INS	70h

APDU	Description								
	b7	b6	b5	b4	b3	b2	b1	b0	Description
	-	0	0	0	0	0	0	-	ACOS TRNS_AUT is disabled
	-	0	0	0	0	0	1	-	ACOS TRNS_AUT is enabled
P1	0	-	-	-	-	-	-	0	3DES
	0	-	-	-	-	-	-	1	DES
	1	-	-	-	-	-	-	0	3K DES (ACOS3 only)
	1	-	-	-	-	-	-	1	AES (ACOS3 only)
P2	0h								
P3	14h								
Data	Data Block								

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3, must be 14h
6A 83h	ACOS Key K_S or K_{ACCT} are not ready; use DIVERSIFY command to generate K_{ACCT}; if applicable, use PREPARE AUTHENTICATION to generate K_S.
69 82h	Security condition not satisfied
6F 00h	DEBIT CERTIFICATE is invalid
90 00h	Success, DEBIT CERTIFICATE is valid

5.1.10.10. Get Key

This command allows secure key injection from the current SAM's Key File (SFI=02h) into another ACOS6/ACOS6-SAM with or without key diversification. Using this ensures that the keys to be injected are protected by encryption and message authentication codes.

The Get Key command also allows secure key injection from the current SAM's Key File (SFI=02h) into ACOS7/10, MIFARE DESFire, MIFARE DESFire EV1 or MIFARE Plus card with key diversification. Using this ensures that the key to be injected is protected by encryption and message authentication codes.

If bit 7 of the Special Function Flag (Key Injection Only Flag) of the **Card Header Block** (Section 3.2 of ACOS6-SAM Reference Manual) has been set and the key file has been activated, Get Key must be used for loading or changing keys in the card. Setting this bit will disable Read Record command for the key file under any circumstances after activation.

Before this command is to be executed, a session key is already established with the target card with the mutual authentication procedure of **Mutual Authentication** (Section 5.3 of ACOS6-SAM Reference Manual) or the MIFARE Plus/MIFARE DESFire mutual authentication procedure.

Note: *The GET KEY command can only get the Key data.*

<table>
<tr><th>APDU</th><th colspan="4">Description</th></tr>
<tr><td>CLA</td><td colspan="4">80h</td></tr>
<tr><td>INS</td><td colspan="4">CAh</td></tr>
<tr><td rowspan="11">P1</td><td colspan="4">Get Key for ACOS card Set Key</td></tr>
<tr><td colspan="4">00h Response data is Key in MSAM
01h Response data is 16-byte Diversify Key
02h Response data is 24-byte Diversify Key
03h Response data is the Change Key command of MIFARE Plus Card</td></tr>
<tr><td colspan="4">Get Key for DESFire card Change Key, Response data for DESFire/DESFire EV1 Change Key</td></tr>
<tr><td></td><td>Card Type</td><td>Authenticate Key No. And Changing Key No.*</td><td>Key Length</td></tr>
<tr><td>80h</td><td>MIFARE DESFire</td><td>Are DIFFERENT in MIFARE DESFire card</td><td>16 bytes</td></tr>
<tr><td>81h</td><td>MIFARE DESFire EV1</td><td>Are DIFFERENT in MIFARE DESFire EV1 card</td><td>16 bytes</td></tr>
<tr><td>82h</td><td>MIFARE DESFire EV1</td><td>Are DIFFERENT in MIFARE DESFire EV1 card</td><td>24 bytes</td></tr>
<tr><td>88h</td><td>MIFARE DESFire</td><td>Are the SAME in MIFARE DESFire card</td><td>16 bytes</td></tr>
<tr><td>89h</td><td>MIFARE DESFire EV1</td><td>Are the SAME in MIFARE DESFire EV1 card</td><td>16 bytes</td></tr>
<tr><td>8Ah</td><td>MIFARE DESFire EV1</td><td>Are the SAME in MIFARE DESFire EV1 card</td><td>24 bytes</td></tr>
<tr><td>P2</td><td colspan="4">Key ID in SAM (New key for change)</td></tr>
<tr><td>P3</td><td colspan="4">If P1 = 00h, P3 is 08h
If P1 = 01/02h, P3 is 10h
If P1 = 03h, P3 is 0Bh
If P1 = 80/81/82/88/89/8Ah: P3 is 0Bh</td></tr>
<tr><td>Data</td><td colspan="4">If P1 = 00h, command data is RND_{Target}
If P1 = 01/02h, command data is RND_{Target} + serial (or batch) number of target card
If P1 = 03h
- Serial Number for target card (8 Byte)
- Write Command (A0 or A1) (1 Byte)
- BNr (2 Byte)
If P1 = 80/81/82/88/89/8Ah:
- Serial Number for target card (8 Byte)
- Original Key ID (Key in SAM card stored the Original key, 00 = Default Key of DESFire - Card)
- Key No. (DESFire Card Key No.)
- Key Version (DESFire Card Key Version, If not used, value = 00)</td></tr>
</table>

* This column points out if the listed cards have a distinct Change Key and Authenticate Key, or if they use the same value for both keys.

Specific Response Status Bytes

SW1 SW2	Description
69 85h	SAM Session Key not ready
62 83h	Current DF is blocked, or Target EF is blocked
69 86h	No DF selected
69 81h	Wrong file type of Key file, it should be Internal Linear Variable File
69 82h	Target file's header block has wrong checksum, or security condition not satisfied
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3
6A 83h	Target Key is not ready or Key Length less than 16
61 1Ch	Success, use GET RESPONSE to get the result

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 ISO14443 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 ~ 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 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 ISO14443 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>8Nh</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 ~ 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>Byte 1~4</th><th>Byte 5~7</th><th>Byte 8</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</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

5.2.2. Pseudo APDU for Contactless Interface

5.2.2.1. 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 ISO 14443A card(UID + 2 Bytes)

Response	Data Out		
Result	ATS	SW1	SW2

Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation was completed successfully.
Error	6Xh	XXh	Fail.

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};
```

5.2.2.2. Get PICC Data

This command returns the PICC data of the connected PICC.

Get PICC Data APDU Format (5 Bytes)

Command	Class	INS	P1	P2	Le
Get PICC Data	FFh	CAh	00h	02h	00h

If TypeA card, Get ATQA + UID + SAK Response Format (2 Bytes + 4/7/10 Bytes + 1 Byte + 2 Bytes)

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

If TypeB card, Get ATQB (12 Bytes + 2 Bytes)

Response	Data Out		
Result	ATQB	SW1	SW2

Response Codes

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

5.2.3. 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.

5.2.3.1. 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 1st data object 02 = error in the 2nd 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
XX 6A 80h	Data object XX with unexpected value
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 objects 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.

5.2.3.2. 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	Le
Manage Session	FFh	C2h	00h	00h	DataLen	DataObject (N bytes)	--/00h

Where:

Data Object (1 byte)

Tag	Data Object
81h	Start Transparent Session
82h	End Transparent Session
83h	Turn Off RF Field
84h	Turn On RF Field
5F 46h	Timer

Manage Session Response Data Object

Tag	Data Object
C0h	Generic Error status

5.2.3.2.1. 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	-

5.2.3.2.2. 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	-

5.2.3.2.3. Turn Off the RF Data Object

This command turns off the antenna field.

Turn off RF Field Data Object

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

5.2.3.2.4. Turn On the RF Data Object

This command turns on the antenna field.

Turn on the RF Field Data Object

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

5.2.3.2.5. Timer Data Object

This command creates a 32-bit timer data object in unit of 1 µs.

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

Timer Data Object

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

5.2.3.3. Transparent Exchange Command

This command transmits and receives any bit or bytes from ICC.

Transparent Exchange Command

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

Where **Data Object (1 byte):**

Tag	Data Object
90h	Transmission and Reception Flag
91h	Transmission Bit Framing
95h	Transceive – Transmit and Receive
5F 46h	Timer
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

5.2.3.3.1. Transmission and Reception Flag Data Object

This command defines the framing and RF parameters for the following transmission.

Transmission and Reception Flag Data Object

Tag	Length (1 byte)	Value		
		Byte 0		Byte 1
		bit	Description	
90h	02h	0	0 – append CRC in the transmit data 1 – do not append CRC in the transmit data	00h
		1	0 – discard CRC from the received data 1 – do not discard CRC from the received data (i.e. no CRC checking)	
		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)	
		5-7	RFU	

5.2.3.3.2. Transmission Bit Framing Data Object

This command defines 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.

5.2.3.3.3. Transceive Data Object

This command transmits and receives 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 µs.

Transceive Data Object

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

5.2.3.3.4. Timer Data Object

This command creates a 32-bit timer data object in unit of 1 µs.

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

Timer Data Object

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

5.2.3.3.5. Response Bit Framing Data Object

Inside the response, this command is used to notify the received transmission 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

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.

5.2.3.3.6. 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	RFU
		1	0 – no collision 1 – collision detected	
		2	0 – no parity error 1 – parity error detected	
		3	0 – no framing error 1 – framing error detected	
		4 – 7	RFU	

5.2.3.3.7. 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)

5.2.3.4.
5.2.3.4.
5.2.3.4.
5.2.3.4.
5.2.3.4.
5.2.3.4.
5.2.3.4.
5.2.3.4.
5.2.3.4.
5.2.3.4.

5.2.3.4. Switch Protocol Command

This command specifies the protocol and different layers of the standard within the transparent session.

Switch Protocol Command

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

Where:

Data Object (1 byte)

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

Switch Protocol Response Data Object

Tag	Data Object
C0h	Generic Error status
5F 51h	ICC response (ISO14443 Part 4)
8Fh	ICC response (ISO14443 Part 3, Felica, ISO15693)

5.2.3.4.1.
5.2.3.4.1.
5.2.3.4.1.

5.2.3.4.1. Switch Protocol Data Object

This command specifies 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

5.2.3.4.2. Response Data Object

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

Response Data Object

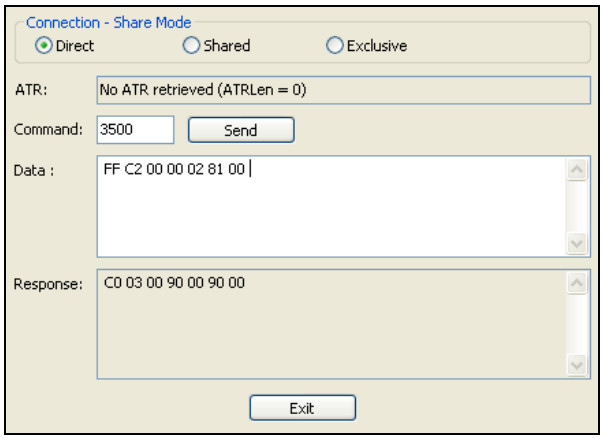
Tag	Length (1 byte)	Value
5F 51h	DataLen	ATR
8Fh	DataLen	Final SAK (if Type A part 3) or PI in ATQB (if Type B part 3).

5.2.3.5. PCSC 2.0 Part 3 Example

1. Start Transparent Session.

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

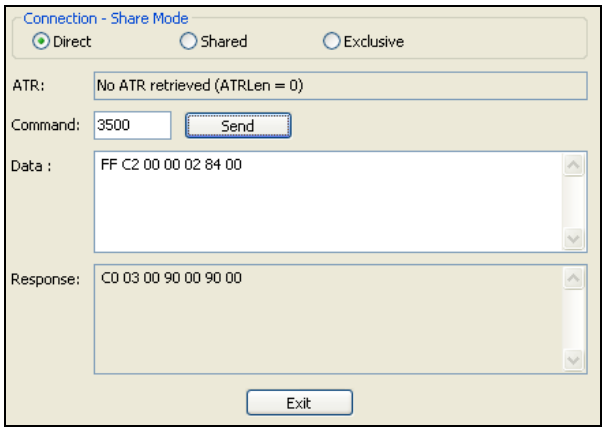
   Response: **C0 03 00 90 00 90 00**



2. Turn the Antenna Field on.

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

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

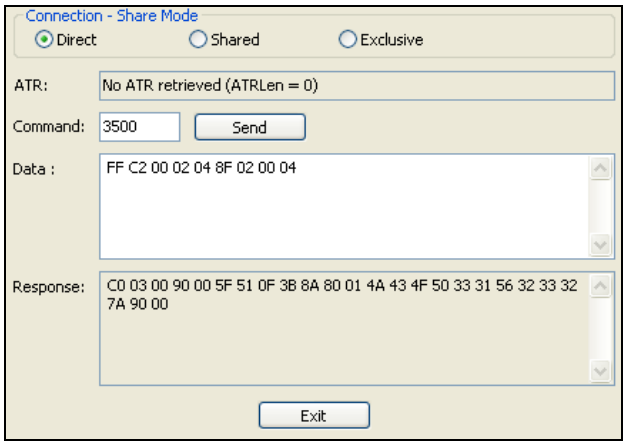


3. ISO 14443-4A Active.

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

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

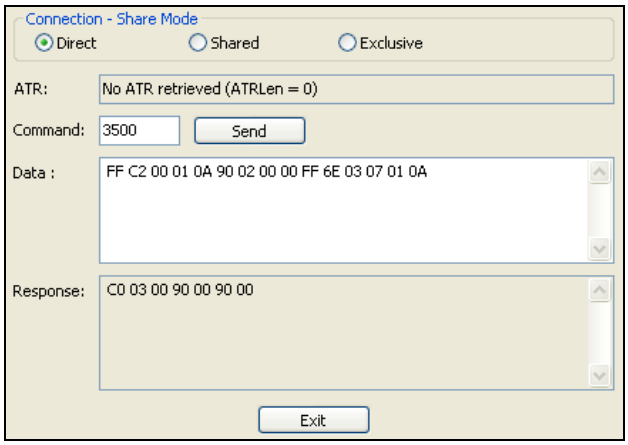
   C0 03 00 90 00 5F 51 [Len] [ATR] 90 00



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**

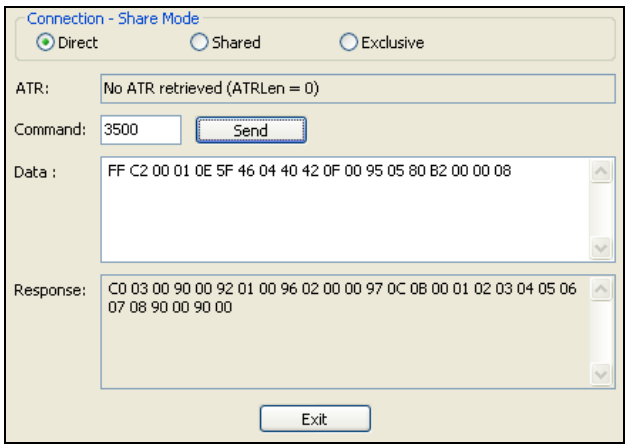
   Response: **C0 03 00 90 00 90 00**



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**

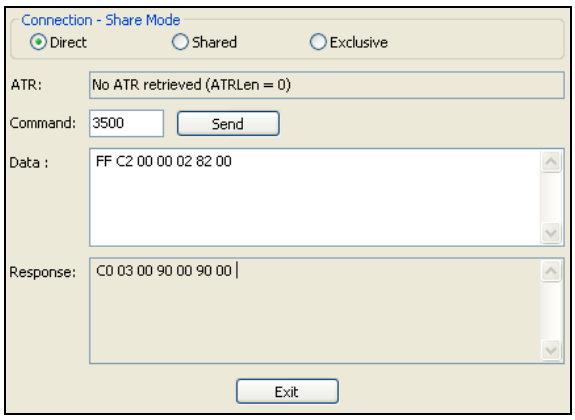
Response: **C0 03 00 90 00 92 01 00 96 02 00 00 97 0C [Card Response] 90 00**



6. End Transparent Session.

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

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



5.2.4. PICC Commands for MIFARE Classic (1k / 4k) Memory Cards

5.2.4.1. Load Authentication Keys

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

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 was completed successfully.
Error	63 00h	The operation 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}
```

5.2.4.2. 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 (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 was completed successfully.
Error	63h	00h	The operation 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
Sector 1	04h – 06h	07h
..	..	..
..	..	..
Sector 14	38h – 0Ah	3Bh
Sector 15	3Ch – 3Eh	3Fh

Table 4: 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	2 KB

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	
Sector 33	90h ~ 9Eh	9Fh	
..			
..			
Sector 38	E0h ~ EEh	EFh	
Sector 39	F0h ~ FEh	FFh	2 KB

Table 5: MIFARE Classic 4K Memory Map

Byte Number	0	1	2	3	Page
Serial Number	SN0	SN1	SN2	BCC0	0
Serial Number	SN3	SN4	SN5	SN6	1
Internal/Lock	BCC1	Internal	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 6: 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.*

5.2.4.3. Read Binary Blocks

This command retrieves multiple 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 was completed successfully.
Error	63h	00h	The operation 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
```

5.2.4.4. Update Binary Blocks

This command writes multiple data blocks on 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 Classic 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 was completed successfully.
Error	63	00h	The operation 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}
```

5.2.4.5. Write Value Block

This command is used to write a 4-byte value to a block in a card compatible with MIFARE Standard. User should perform succeed authentication to get the access right of the block before sending this command.

Write Value Block APDU Format (10 bytes)

Command	Class	INS	P1	P2	Lc	Data In	
Write Value Block	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 = Write the VB_Value into the block. The block will then be converted to a 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 was completed successfully.
Error	63	00h	The operation failed.

5.2.4.6. Read Value Block

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

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 was completed successfully.
Error	63	00h	The operation failed.

5.2.4.7. Decrement/Increment Value

This command is used to Decrement/Increment a 4-byte value from source block and stores the result to target block in a card compatible with MIFARE Standard. If user wants to store the result to the block same as source block, user can set the target block number equal to 0 or source block number. User should perform succeed authentication to get the access right of both source and target block before sending this command.

Command

Command	Class	INS	P1	P2	Lc	Data In
Debit/Credit Value	FFh	D7h	Target Block#	Source Block#	05h	See below

Command Data

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
01h	4 Bytes Increment Value with MSB first			
02h	4 Bytes Decrement Value with MSB first			

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

5.2.4.8. Copy Value Block

This command is used to copy the value from source block to target block in a card compatible with MIFARE Standard. User should perform succeed authentication to get the access right of both source and target block before sending this command.

Command

Command	Class	INS	P1	P2	Lc	Data In
Copy Value Block	FFh	D7h	Target Block#	Source Block#	02h	See below

Command Data

Byte 0	Byte 1
03h	Target Block#

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

5.2.5. Accessing PCSC-Compliant tags (ISO14443-4)

All ISO 14443-4 compliant cards (PICCs) understand the ISO 7816-4 APDUs. The ACR1581U reader just has to communicate with the ISO 14443-4 compliant cards by exchanging ISO 7816-4 APDUs and responses. The ACR1581U 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 **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 was completed successfully.
Error	63 00h	The operation failed.

Typical sequence may be:

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

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]

5.2.6. Accessing FeliCa tags

For FeliCa access, the command is different from the one used in 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

5.2.7. Supported PICC ATR

The following PICC type/technology are supported by default. The following ATR is returned to CCID Host on PC_to_RDR_IccPowerOn Command if the card is presented to the reader.

Card Type/Technology	ATR
MIFARE Std 1k3	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6A
MIFARE Std 4k3	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 02 00 00 00 00 69
MIFARE UltraLight3	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 03 00 00 00 00 68
MIFARE Plus SL1 2k3	Default: Same as MIFARE Std 1k Alternated: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 36 00 00 00 00 5D
MIFARE Plus SL1 4k3	Default: Same as MIFARE Std 4k Alternated: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 37 00 00 00 00 5C
MIFARE Plus SL2 2k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 38 00 00 00 00 53
MIFARE Plus SL2 4k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 39 00 00 00 00 52
MIFARE UltraLight C3	Default: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 3A 00 00 00 00 51 Alternated: Same as MIFARE UltraLight
SmartMX with MIFARE Std 1k Emulation3	Default: Same as MIFARE Std 1k Alternated: Same as ISO14443-4, Type A
SmartMX with MIFARE Std 4k Emulation[2]	Default: Same as MIFARE Std 4k Alternated: Same as ISO14443-4, Type A
ISO14443-4, Type A	3B 8n 80 01 T1 .. Tn Tck n = Number of Historical bytes in ATS T1 .. Tn = Historical bytes in ATS Tck = XOR of 8n 80 01 T1 .. Tn
ISO14443-4, Type B	3B 88 80 01 T1 .. T8 Tck T1 .. T4 = Application Data in ATQB T5 .. T7 = Protocol Info in ATQB T8 = MBLI in ATA Tck = XOR of 88 80 01 T1 .. T8
FeliCa	3B 8F 80 01 80 4F 0C A0 00 00 03 06 11 00 3B 00 00 00 00 42
ISO15693-3 Generic	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 00 00 00 00 00 63
Infineon My-D Vicinity (SRF55Vxxx)	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 0E 00 00 00 00 6D
ST LRI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 13 00 00 00 00 70
NXP I-Code SLI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 14 00 00 00 00 77
NXP I-Code SLIX/SLIX2	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 35 00 00 00 00 56

[2] Refer to "Param 2" in Set Operation Mode Escape command for configuration and drawback of the alternated ATR definition.

Card Type/Technology	ATR
PicoPass 2K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 17 00 00 00 00 79
PicoPass 2KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 18 00 00 00 00 76
PicoPass 16K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 19 00 00 00 00 77
PicoPass 16KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1A 00 00 00 00 74
PicoPass 16K (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1B 00 00 00 00 75
PicoPass 16KS (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1C 00 00 00 00 72
PicoPass 32KS (16 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1D 00 00 00 00 73
PicoPass 32KS (16 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1E 00 00 00 00 70
PicoPass 32KS (8x2 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1F 00 00 00 00 71
PicoPass 32KS (8x2 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 20 00 00 00 00 4E

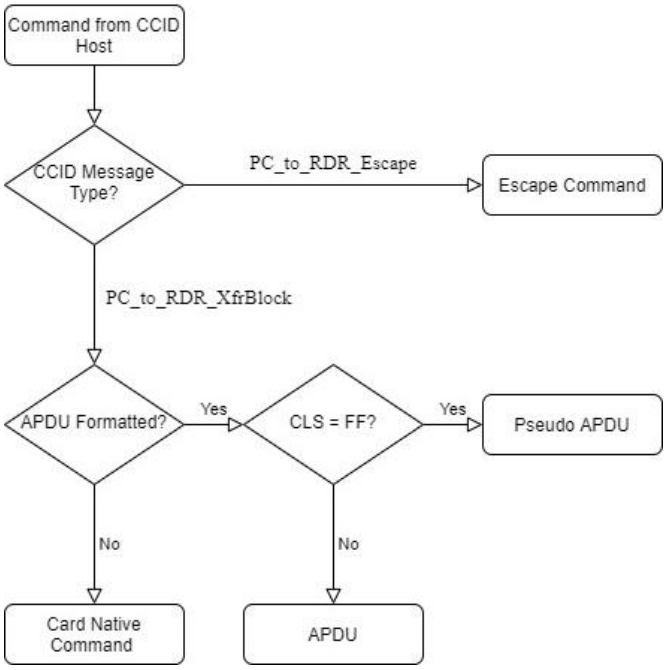
In order to reduce response time for generic application, the support of following PICC type/technology are disabled by default. User could enable the support of each Type/Technology by "Set operation Mode" Escape command. The following ATR is returned to CCID Host on PC_to_RDR_IccPowerOn Command if the card is presented to the reader and the corresponding Type/Technology is enabled.

Card Type/Technology	ATR
SRI (SRIX4K/SRT512)	3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 07 00 00 00 00 69
Topaz	3B 8F 80 01 80 4F 0C A0 00 00 03 06 02 00 30 00 00 00 00 5A
PicoPass 2K	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 17 00 00 00 00 75
PicoPass 2KS	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 18 00 00 00 00 7A
PicoPass 16K	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 19 00 00 00 00 7B
PicoPass 16KS	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1A 00 00 00 00 78
PicoPass 16K (8 x 2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1B 00 00 00 00 79
PicoPass 16KS (8 x 2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1C 00 00 00 00 7E
PicoPass 32KS (16 + 16)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1D 00 00 00 00 7F
PicoPass 32KS (16 + 8x2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1E 00 00 00 00 7C
PicoPass 32KS (8x2 + 16)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 1F 00 00 00 00 7D
PicoPass 32KS (8x2 + 8x2)	ISO15693: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 0A 00 20 00 00 00 00 42
Innovatron	3B 88 80 01 80 4F 05 F0 49 4E 4E 4F 35
CTS	3B 87 80 01 80 4F 04 F0 43 54 53 79

6.0.

6.0. Command Set

Commands from CCID Host could classify as Escape Command, Card Native Command, Pseudo APDU and APDU depend on its format and type of CCID Message used to send.



Escape Command is send by PC_to_RDR_Escape (corresponding to SCardControl() with SCARD_CTL_CODE(3500) in PCSC API) via the others are send by PC_to_RDR_XfrBlock (corresponding to SCardTransmit() in PCSC API).

6.1. Card Native Command and APDU

CCID Host could send Card Native Command or APDU to the Reader by using CCID Message PC_to_RDR_XfrBlock (corresponding to SCardTransmit() in PCSC API). For PICC, if the card support ISO14443 part 4 protocol or Innovatron protocol, the Reader will pack the Command/APDU into the protocol frame and send to the card directly without any interpretation of the Command/APDU. If the card do not support neither protocol, a message “6A 81” will return to CCID Host.

Note: Due to Microsoft Window Smart Card Plug and Play, Microsoft Window may send some APDU to a card at the time of card present. This action will make a DESFire card entering ISO APDU mode such that the card become fail to receive a native command until a card reset. Usually Microsoft Window will reset the card (by PC_to_RDR_IccPowerOff) after 10s of inactive.

6.2. PCSC Pseudo APDU (with Proprietary Extension) for PICC

The following Pseudo APDUs are provided to access a contactless card indirectly. CCID Host could send these APDUs to Reader by using CCID Message PC_to_RDR_XfrBlock (corresponding to SCardTransmit() in PCSC API). After receiving of a Pseudo APDU, it will be interpreted to generate low level card command(s) and then send to card. After the card handling those low level command(s), Reader collect the response(s) from the card and create a response to send back to CCID Host.

6.2.1. Get Data [FF CA …]

This command is used to read out the data obtained during activation process, such as serial number, protocol parameter and etc.

Command

Command	Class	INS	P1	P2	Le
Get Data	FFh	CAh	See below		00h (Full Length)

Command Parameter

P1	P2	Meaning
00h	00h	Get the UID/PUPI/SN of the Card
01h	00h	Get the ATS for Type A Part 4
00h	02h	Get the following Card Type related data in transmission order: Type A: 2 bytes ATQA/ATVA + 4/7/10 Bytes UID + 1 bytes Last SAK. Type B: 12 bytes ATQB
80h	00h	Get the following Card Type related data in transmission order: Type A: 2 bytes ATQA/ATVA + 4/7/10 Bytes UID + 1/2/3 bytes SAK. Type B: 12 bytes ATQB FeliCa: 17 byte ATQ (+ 6 byte ATTR if activated) SRI: 8 byte UID + 1 byte Chip ID. ISO15693: 1 byte DSFID + 8 byte UID CTS: 4 byte SN + 2 byte ATQT Innovatron: 4 byte SN + 1 byte tag address.

Response

Response	Data Out		
Result	Data	SW1	SW2

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.2.2. Load Key [FF 82 …]

This command is used to set the Key Data to the internal key buffer specified by Key Buffer Number. The key buffer is volatile and its content would be used during authentication. This command will not generate card data transfer.

Command

Command	Class	INS	P1	P2	Lc	Data In
Load Authentication Keys	FFh	82h	00h	Key Buffer Number (0 to 1)	Key Length	Key Data

Key Length/Data

Card Type	Key Length (Lc)	Key Data (in Transmission/Storing Order)
MIFARE Standard MIFARE Plus SL1	06h	6 Bytes Crypto1 Key A/B.
MIFARE Plus SL1 MIFARE Plus SL2	16h	6 Bytes Crypto1 Key A/B + 16 Bytes AES Key.
MIFARE Plus SL2	06h	6 Bytes Encrypted Crypto1 Key A/B.
MIFARE UltraLightC MIFARE DESFire	10h	16 Bytes 2K3DES Key.

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.2.3. Authenticate [FF 86 00 00 05 …]

This command is used to performing an authentication to the card to grand access of the protected blocks/pages. Before sending this command, User should use Load Key command to set the correct key data to the buffer specified by Key Buffer Number.

Command

Command	Class	INS	P1	P2	Lc	Data In
Authenticate	FFh	86h	00h	00h	05h	See Below

Command Data

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
01h	00h (RFU)	Address	Key Type	Key Buffer Number

Address and Key Type

Card Type	Address	Key Type
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	00h~FFh: Block 0~255	60h: Crypto1 Key A 61h: Crypto1 Key B
MIFARE UltraLightC	00h (RFU)	80h: 2K3DES
MIFARE DESFire	00h~0Eh: DESFire Key Number 0~14	0Ah: 2K3DES

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.2.4. Read Binary Blocks [FF B0 …]

This command is used to read specified number of byte of data from PICC starting from the specified block/page address. Depend on card type, user may need to perform authentication to get the access right of the required block(s)/page(s) before sending this command.

Command:

Command	Class	INS	P1	P2	Le
Read Binary Blocks	FFh	B0h	Mode and Address		Number of Bytes to Read

P1/P2 (Mode and Address)

Card Type	P1[7:4] Mode	P1[3:0] + P2[7:0] Starting Address (MSB First)
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	00h: Skip Trailers 08h: With Trailers	000h~0FFh: Block 0~255
MIFARE UltraLight MIFARE UltraLightC	00h (Reserved)	000h~02Fh: Page 0~47
SRIX4K/SRT512	00h (Reserved)	000h~07Fh: Block 0~127 0FFh: System Area
PicoPass	00h (Reserved)	000h~0FFh: Block 0~255
ISO15693	00h (Reserved)	000h~0FFh: Block 0~255
Topaz/NFC Type-1 Tag	00h (Reserved)	000h~7FFh: Byte Address

Le (Number of Bytes to Read)

Type	Byte 0	Byte 1	Byte 2
Short	00h: Read 256 bytes 01h~FFh: Read 1~255 bytes	--	
Extended	00h	0000h: Read 65536 bytes 0001h~FFFFh: Read 1~65535 bytes	

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.2.5. Update Binary Blocks [FF D6 …]

This command is used to write specified number (must be multiple of block/page size) of bytes to PICC starting from the specified block/page address. Depend on card type, user may need to perform authentication to get the access right of the required block(s)/page(s) before sending this command.

User should take a great care for writing to block/page that may change the security setting of the card (e.g. sector trailers of MIFARE card) as this may lock the card if incorrect data is written or operation is failed. As a result, to minimize the risk of card locking, it is not recommended to write to multiple block/page in a single APDU command if security block/page is involved.

Command

Command	Class	INS	P1	P2	Lc	Data In
Update Binary Blocks	FFh	D6h	Mode and Address		Number of Bytes to Write	Data Bytes

P1/P2 (Mode and Address) and Write Size alignment (Block/Page Size)

Card Type	P1[7:4] Mode	P1[3:0] + P2[7:0] Starting Address (MSB First)	Blk/Page Size (Bytes)
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	00h: Skip Trailers 08h: With Trailers	000h~0FFh: Block 0~255	16
MIFARE UltraLight MIFARE UltraLightC	00h: Reserved	000h~02Fh: Page 0~47	4
SRIX4K/SRT512	0x0 (Reserved)	SRIX4K/SRT512	4
PicoPass	0x0 (Reserved)	PicoPass	8
ISO15693	0x0 (Reserved)	ISO15693	1 ~ 32
Topaz/NFC Type-1 Tag	00h: with Erase	000h~7FFh: Byte Address	1(Addr 78h) or 8(Else)
	08h: without Erase		

Lc (Number of Bytes to Write)

Type	Byte 0	Byte 1	Byte 2
Short	01h~FFh: Write 1~255 bytes	--	
Extended	00h	0001h~FFFFh: Write 1~65535 bytes	

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.2.6. Manage Session [FF C2 00 00 …]

This command allows user to start a session with polling disable for the following communication. User should end the session as soon as those communications finished.

Please note, this command may make the reader fail detect a card present/absence if used incorrectly. This fail may be unable to recover automatically until a logical/physical reader disconnection.

Command

Command	Class	INS	P1	P2	Lc	Data In	Le
Manage Session	FFh	C2h	00h	00h	Cmd Data Length	Cmd TLV	--/00h

Response Code

Rsp Data	SW1 SW2	Meaning
--	90 00h	The operation was completed successfully.
Rsp TLV	90 00h	For Le = 0x00, One of Command TLV Fail. For Detail of Error, refer to Rsp TLV.
--	6X XXh	For Le = --, One of Command TLV Fail.

Cmd TLV

Cmd	Meaning
Start Session: 81 00h	Start a Session and Disable Polling.
RF Off: 83 00h	Turn off RF.
Timer: 5F 46 04h [TIME]	Set the sleep time before the next RF On/Off TLV. [TIME]: 4 byte value (MSB first) in range from 1000 to 100000 us. The actual sleep time will round up to nearest 1000us.
RF On: 84 00h	Turn on RF.
End Session: 82 00h	End a Session and Re-enable Polling.

Rsp TLV

Rsp	Meaning
TLV Error: C0 03 NN 6X XXh	Error in the NNth Command TLV.

6.2.7. Transparent Exchange [FF C2 00 01 …]

This command allows user transmit and receive any bit or bytes to/from card, with option to configure various link and transport layer (e.g. ISO14443 part 4) and some link layer redundancy (CRC and parity) optionally. User could embed any card specific raw data into this pseudo APDU and then send to the card.

Please note, this command may interference internal handling of card support, may change the card status without notification to the driver/firmware and may require a card reset and/or removal to bring the driver/firmware back to normal.

Command

Command	Class	INS	P1	P2	Lc	Data In	Le
Manage Session	FFh	C2h	00h	01h	Cmd Data Length	Cmd TLV	00h

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

Cmd TLV

Cmd	Meaning
Transceive Flag: 90 02 [Flag] 00h	Set the Flag for the following Transceive TLV. Flag[7:5]: RFU; Set to 0 Flag[4]: Set to disable ISO14443 Part 4 Flag[3]: Set to disable receiving parity handling Flag[2]: Set to disable transmitting parity handling Flag[1]: Set to disable receiving CRC handling Flag[0]: Set to disable transmitting CRC handling If this TLV is missing, the Flag value set in previous command is used. If Flag value is never set, current protocol value is used.
Transmit Bit Frame: 91 01h [NumBit]	Set the Bit Frame for the following Transceive TLV. If this TLV is missing, the default value is 0. NumBit[7:3]: RFU; Set to 0 NumBit[2:0]: Number of valid bits in last byte (0 means all valid).
Timer: 5F 46 04h [TIME]	Set the timeout for the following Transceive TLV. [TIME]: 4 byte value (MSB first) in range 1 us to 1000000 us. The actual timeout will round up to nearest 302.07 x 20~15 us. If this TLV is missing, the FWTI value set previously will be used as timeout.
Set FWTI: FF 6E 03 03 01h [FWTI]	Set FWT/Timeout for Transceive. If FWTI does not set by any previous "FF C2h ..." command, the default value is 0. FWTI: 0 ~ 15, FWT/Timeout = 302.07 x 2FWTI us
Transceive: 95h [Size] [Data]	Size: Size of Data coded in BER-TLV length field. Data: Data to be Transmit.

Rsp TLV

Rsp	Meaning
Receive Bit framing: 92 01h [NumBit]	NumBit[7:3]: RFU; Set to 0. NumBit[2:0]: Number of valid bits in last byte (0 means all valid).
Response: 97h [Size] [Data]	Size: Size of Data coded in BER-TLV length field. Data: Data Received.
Response Status: 96 02h [Status] 00h	Status[7:4]: RFU. Status[3]: Framing Error. Status[2]: Parity Error. Status[1]: RFU. Status[0]: CRC Error.

6.2.8. Switch Protocol [FF C2 00 02 …]

This command allows user to switch to specify protocol, select protocol layer and parameter.

Please note, this command may interference internal handling of card support, may change the card status without notification to the driver/firmware and may require a card reset and/or removal to bring the driver/firmware back to normal.

Command

Command	Class	INS	P1	P2	Lc	Data In	Le
Manage Session	FFh	C2h	00h	02h	Cmd Data Length	Cmd TLV	00h

Response Code

Rsp Data	SW1 SW2	Meaning
Rsp TLV	90 00h	Succeed with data.
--	90 00h	Succeed.
--	6X XXh	Fail.

Cmd TLV

Cmd	Meaning
Set Baud: FF 6E 03 05 01h [Baud]	Set the Baud for Part/Layer 4 to be applied during Switch Protocol. If [Baud] does not set by any previous "FF C2h ..." command, the default value is 98h (106 kbps). Baud[7:2]: RFU, Set to 100110b. Baud[1:0]: Baud to be set, 00b (106 kbps), 01b (212 kbps), 10b (424 kbps), 11b (848 kbps).
Switch Protocol: 8F 02h [RF] [Layer]	Switch the protocol to specified RF and/or Layer. [RF]: 00h: ISO14443A, 01h: ISO14443B 02h: ISO15693, 03h: FeliCa, FFh: Current RF Other: RFU [Layer]: 02h: Layer/Part 2, 03h: Layer/Part 3, 04h: Layer/Part 4 (For A/B Only) Other: RFU Note: It must be in a Transparent Session (Disable Polling) if switching to Layer/Part 2.

Rsp TLV

Rsp	Meaning
Response: 8Fh [Size] [Data]	Size: Size of Data coded in BER-TLV length field. Data: ATR (if Part 4) or Final SAK (if Type A part 3) or PI in ATQB (if Type B part 3).

6.3. Proprietary Pseudo APDU for PICC

The following Pseudo APDUs are provided as supplement to PCSC Pseudo APDUs to access a contactless card indirectly. The internally handling of these APDU is similar to PCSC Pseudo APDUs.

6.3.1. Read Value Block [FF B1 …]

This command is used to read a 4-byte value from a valid value block in a card compatible with MIFARE Standard. User should perform succeed authentication to get the access right of the block before sending this command.

Command

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

Response

Rsp Data	SW1 SW2	Meaning
4 Bytes Value with MSB first	90 00h	Succeed with data.
--	6X XXh	Fail.

6.3.2. Write Value Block [FF D7 …]

This command is used to write a 4-byte value to a block in a card compatible with MIFARE Standard. User should perform succeed authentication to get the access right of the block before sending this command.

Command

Command	Class	INS	P1	P2	Lc	Data In
Write Value Block	FFh	D7h	00h	Block Number	05h	See below

Command Data

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
00h	4 Bytes Value with MSB first			

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.3.3. Decrement/Increment Value [FF D7 …]

This command is used to decrement/Increment a 4-byte value from source block and stores the result to target block in a card compatible with MIFARE Standard. If user wants to store the result to the block same as source block, user can set the target block number equal to 0 or source block number. User should perform succeed authentication to get the access right of both source and target block before sending this command.

Command

Command	Class	INS	P1	P2	Lc	Data In
Decrement/Increment Value	FFh	D7h	Target Block#	Source Block#	05h	See below

Command Data

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
01h	4 Bytes Credit Value with MSB first			
02h	4 Bytes Debit Value with MSB first			

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.3.4. Copy Value Block [FF D7 …]

This command is used to copy the value from source block to target block in a card compatible with MIFARE Standard. User should perform succeed authentication to get the access right of both source and target block before sending this command.

Command

Command	Class	INS	P1	P2	Lc	Data In
Copy Value Block	FFh	D7h	Target Block#	Source Block#	02h	See below

Command Data

Byte 0	Byte 1
03h	Target Block#

Response Code

Results	SW1 SW2	Meaning
Success	90 00h	The operation was completed successfully.
Error	6X XXh	Fail.

6.4. Escape Command

The following commands are provided to configure PCD/NFC and to access special function of the reader. CCID Host could send these commands to reader by using CCID Message PC_to_RDR_Escape (corresponding to SCardControl() with SCARD_CTL_CODE(3500) in PCSC API). After receiving of an Escape Command, it will be interpreted to perform various operations and then generate a response to send back to CCID Host.

Note:
Should send these commands under correct interface. For example, E0 00 00 25 01 00 (Section 6.4.1.1) should send through PICC interface (Section 6.4.1). E0 00 00 2B 00 (Section 6.4.2.1) should send through ICC interface (Section 6.4.2).

6.4.1. Escape Command for PICC

6.4.1.1. RF Control [E0 00 00 25 01 …]

This command is used to set the RF control.

Command

Command	Class	INS	P1	P2	Lc	Data Out
RF Control	E0h	00h	00h	25h	01h	RF status

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	RF status

RF Status: 1 Byte

RF status	Description
00h	RF Off
01h	RF On, with Polling
02h	RF On, without Polling

Default Setting – 01h (RF On, with Polling)

6.4.1.2. Get PCD/PICC Status [E0 00 00 25 00]

This command is used to get the PCD/PICC status

Command

Command	Class	INS	P1	P2	Le
Get PCD/PICC Status	E0h	00h	00h	25h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Get PCD/PICC Status

PCD/PICC Status: 1 Byte

RF status	Description
00h	RF Off
01h	No PICC
02h	PICC Ready
03h	PICC Selected/Activated
FFh	Error

6.4.1.3. Get Polling/ATR Option [E0 00 00 23 00]

This command is used to set/get the Polling Option but save the setting without another command. This command should only be used for initial reader configuration.

Command

Command	Class	INS	P1	P2	Le
Get Polling/ATR Option	E0h	00h	00h	23h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	03h	01h	PICC Polling/ATR Option

6.4.1.4. Set Polling/ATR Option [E0 00 00 23 01 …]

This command is used to set the polling option.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Polling/ATR Option	E0h	00h	00h	23h	01h	PICC Polling/ATR Option

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	PICC Polling/ATR Option

PICC Polling/ATR Option - 1 Byte

Operating Parameter	Parameter	Description	Option
Bit 0	Enable Polling	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit 1	Enable RF Off		
Bit 2	RFU		
Bit 3	Enable extra MIFARE type identification for Part 3 card in ATR	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit 4 ~ 5	RF Off Interval		See below
Bit 6	RFU		
Bit 7	Enable Part 4 ATR for SmartMX/JCOS card with MIFARE emulation	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip

RF Off Interval – 2 Bit **Case 1:** Disabled RF Off (Bit 1 = 0)

Operating Parameter		USB Active (D0)	USB Suspend (D2)
Bit 5	Bit 4		
0	0	No RF Off	250 ms
0	1		500 ms
1	0		1000 ms
1	1		2500 ms

Case 2: Enabled RF Off (Bit 1 = 1)

Operating Parameter		USB Active (D0)	USB Suspend (D2)
Bit 5	Bit 4		
0	0	250 ms	500 ms
0	1	500 ms	1000 ms
1	0	1000 ms	2500 ms
1	1	2500 ms	2500 ms

Default Setting – 8Bh (Enabled Polling, Enabled RF Off, Enabled extra MIFARE type identification for Part 3 card in ATR, RF Off Interval[00], Enabled Part 4 ATR for SmartMX/JCOS card with MIFARE emulation)

6.4.1.5. Get PICC Polling Type [E0 00 01 20 00]

This command is used to get the allowed Technology/Polling Type but save the setting without another command. This command should only be used for initial reader configuration.

Command

Command	Class	INS	P1	P2	Le
Get PICC Polling Type	E0h	00h	01h	20h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E0h	00h	00h	00h	02h	PICC Polling Type

6.4.1.6. Set PICC Polling Type [E0 00 01 20 02 …]

This command is used to set the PICC polling type.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set PICC Polling Type	E0h	00h	01h	20h	02h	PICC Polling Type

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E0h	00h	00h	00h	02h	PICC Polling Type

PICC Polling Type - 2 Byte, LSB First, Bit Mask of following

Operating Parameter	Parameter	Description	Option
Bit 0	ISO 14443A Type A	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit 1	ISO 14443A Type B		
Bit 2	FeliCa		
Bit 5	Innovatron		
Bit 6	SRI/SRIX		
Bit 8	Picopass (ISO14443B)		
Bit 9	Picopass (ISO15693)		
Bit 10	ISO15693		
Bit 11	CTS		

Default Setting – 0705h (ISO14443 Type A, ISO14443 Type B, FeliCa, Picopass (ISO14443B), ISO15693)

6.4.1.7. Get Auto PPS [E0 00 00 24 00]

Whenever a PICC is recognized, the reader will try to change the communication speed between the PCD and PICC as defined by the maximum connection speed. If the card does not support the proposed connection speed, the reader will try to connect the card with a slower speed setting.

Command

Command	Class	INS	P1	P2	Le
Get Auto PPS	E0h	00h	00h	24h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In	
Result	E1h	00h	00h	00h	02h	Max Speed	Current Speed

6.4.1.8. Set Auto PPS [E0 00 00 24 01 …]

This command is used to set the auto PPS.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Auto PPS	E0h	00h	00h	24h	01h	Max Speed

Response Code

Response	Class	INS	P1	P2	Le	Data In	
Result	E1h	00h	00h	00h	02h	Max Speed	Current Speed

Speed of PPS

Speed	Description
00h	106 kbps; default setting, equal to No Auto PPS
01h	212 kbps
02h	424 kbps
03h	848 kbps

Default Setting – 02h (424 kbps)

Notes:

1. Normally, the application should know the maximum connection speed of the PICCs being used. The environment also affects the maximum achievable speed. The reader just uses the proposed communication speed to talk with the PICC. The PICC will become inaccessible if the PICC or environment does not meet the requirement of the proposed communication speed.

2. If the higher speed setting affects the performance of the reader, please switch back to a lower speed setting.

6.4.1.9. Read PICC Type [E0 00 00 35 00]

This command is used to read the PICC type.

Command

Command	Class	INS	P1	P2	Le
Get PICC Type	E0h	00h	00h	35h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In	
Result	E1h	00h	00h	00h	02h	Type	Status

Type: 1 Byte

Type	Description
CCh	No PICC
04h	Topaz
10h	MIFARE
11h	FeliCa
20h	Type A, Part 4
23h	Type B, Part 4
25h	Innovatron
28h	SRIX
30h	PicoPass
FFh	Other

Status: 1 Byte

Status	Description
00h	RF Off
01h	No PICC
02h	PICC Ready
03h	PICC Selected/Activated
FFh	Error

6.4.1.10.
6.4.1.10.
6.4.1.10.

6.4.1.10. Escape Command for PICC – HID Keyboard

6.4.1.10.1. Get Output Format

This command is used to get output format.

Command

Command	Class	INS	P1	P2	Le
Get Output Format	E0h	00h	00h	90h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In	
Result	E1h	00h	00h	00h	02h	Output Format	Output Order

6.4.1.10.2. Set Output Format

This command is used to set output format.

Command

Command	Class	INS	P1	P2	Lc	Data Out	
Set Output Format	E0h	00h	00h	90h	02h	Output Format	Output Order

Response Code

Response	Class	INS	P1	P2	Le	Data In	
Result	E1h	00h	00h	00h	02h	Output Format	Output Order

Output Format: 1 Byte

Operating Parameter	Parameter	Description	Option
Bit 7 ~ 4	Letter Case	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit 3 ~ 0	Display Mode		

Output Order: 1 Byte

Status	Description
00h	Default order (UID Byte 0, UID Byte 1 … UID Byte N) Example: aa cc bb dd (original /actual UID order)
01h	Reverse order (UID Byte N, UID Byte N-1 … UID Byte 0) Example: dd bb cc aa (reverse the UID order)

Letter Case: Upper 4 Bits (Bit 7 ~ 4)

Status (From bit 7~4)	Description (Don't care about x bit)
1xxx	Reserved
00x0	Lowercase
00x1	Uppercase
000x	Only Support 4 bytes UID
001x	Support 4, 7, 8, 10 bytes UID

Display Mode: Lower 4 Bits (Bit 3 ~ 0)

Status (From bit 7~4)	Description (Don't care about x bit)
0h	Hex
1h	Dec (byte by byte)
2h	Dec
3h	6H-6H
4h	8H-8H
5h	10H-10H
6h	14H-14H
7h	20H-20H
8h	6H-8D
9h	6H-10D
Ah	8H-10D
Bh	10H-14D
Ch	2H4H-8D
Dh	14H-17D

6.4.1.10.3. Get Character at Start, Between, at End UID

This command is used to get character at Start, Between, End UID.

Command

Command	Class	INS	P1	P2	Le
Get Character of UID	E0h	00h	00h	91h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In		
Result	E1h	00h	00h	00h	03h	Between	End	Start

6.4.1.10.4. Set Character at Start, Between, at End UID

This command is used to set character at Start, Between, End UID.

Command

Command	Class	INS	P1	P2	Lc	Data Out		
Set Character of UID	E0h	00h	00h	91h	03h	Between	End	Start

Response Code

Response	Class	INS	P1	P2	Le	Data In		
Result	E1h	00h	00h	00h	03h	Between	End	Start

Between: 1 Byte (The character between each UID)

Status	Description
FFh	No character in between
Other	Refer to Universal Serial Bus (USB) HID Usage Tables

End: 1 Byte (The character at the end of output)

Status	Description
FFh	No character in between
Other	Refer to Universal Serial Bus (USB) HID Usage Tables

Start: 1 Byte (The character at the start of output)

Status	Description
FFh	No character in between
Other	Refer to Universal Serial Bus (USB) HID Usage Tables

Notes:

1. *only the characters ";" "," "," "," "-" are supported in the AZERTY keyboard layout for the characters in between. Zero (0) and Backspace are NOT supported.*

6.4.1.10.5. Get Keyboard Layout Language

This command is used to get keyboard layout language.

Command

Command	Class	INS	P1	P2	Le
Get Keyboard Layout Language	E0h	00h	00h	92h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Keyboard Layout Language

6.4.1.10.6. Set Keyboard Layout Language

This command is used to set keyboard layout language.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Keyboard Layout Language	E0h	00h	00h	92h	01h	Keyboard Layout Language

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Keyboard Layout Language

Keyboard Layout Language: 1 Byte

Status	Description
00h	English
01h	French
02h	Reserved
03h	Lithuanian

6.4.1.10.7. Get Host Interface

This command is used to get host interface

Command

Command	Class	INS	P1	P2	Le
Get Host Interface	E0h	00h	00h	93h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Host Interface

6.4.1.10.8. Set Host Interface

This command is used to set host interface

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Host Interface	E0h	00h	00h	93h	01h	Host Interface

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Host Interface

Host Interface: 1 Byte

Status	Description
00h	Only HID Keyboard
01h	Only CCID Reader
02h	HID Keyboard + CCID Reader

6.4.1.11. Escape Command for PICC – Card Emulation

6.4.1.11.1. 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.*

Command

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

Response Code

Response	Class	INS	P1	P2	Le	Data In		
Result	E1h	00h	00h	00h	03h	NFC Mode	01h	01h

NFC Device Mode: 1 Byte

Status	Description
02h	NFC Forum Type 2 Tag Mode
03h	FeliCa
Other	Card Read/Write Mode

Note: *Please enter to Card Read/Write mode before switching to different card emulation mode. The response will be showed after the Card Emulation Mode initial is done.*

Byte Number	0	1	2	3	Byte Address access by USB
Serial Number	SN0	SN1	SN2	SN3	Nil
Reserved	Reserved	Reserved	Reserved	Reserved	Nil
Internal/Lock	Reserved	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
Data read/write	Data52	Data53	Data54	Data55	52-55
Data read/write			…		…
Data read/write	Data1996	Data1997	Data1998	Data1999	1996~1999

Accessible area (2000 bytes)

Table 7: NFC Forum Type 2 Tag Memory Map (2000 bytes)

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 8: 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).*

6.4.1.11.2.

6.4.1.11.2. Read Card Emulation Data (NFC Forum Type 2 Tag)

This command is used to read the emulated card content.

Command

Command	Class	INS	P1	P2	Lc	Data In			
Read Card Emulation Data	E0h	00h	00h	60h	04h	00h	NFC Mode	Start Offset	Length

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	Length	Data

6.4.1.11.3. Write Card Emulation Data (NFC Forum Type 2 Tag)

This command is used to write the emulated card content.

Command

Command	Class	INS	P1	P2	Lc	Data In				
Read Card Emulation Data	E0h	00h	00h	60h	Length + 04h	01h	NFC Mode	Start Offset	Length	Data

Response Code

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

NFC Device Mode: 1 Byte

Status	Description
02h	NFC Forum Type 2 Tag Mode
03h	FeliCa
Other	Card Read/Write Mode

Start Offset: 1 Byte – Address start to write

Length: 1 Byte – No. of byte to write

6.4.1.11.4. Read Card Emulation Data (NFC Forum Type 2 Tag)) (Extended)

This command is used to read the emulated card content.

Command

Command	Class	INS	P1	P2	Lc		Data In			
Read Card Emulation Data	E0h	00h	01h	60h	05h	00h	NFC Mode	Start Offset Bit[15:8]	Start Offset Bit[7:0]	Length

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	Length	Data

6.4.1.11.5. Write Card Emulation Data (NFC Forum Type 2 Tag) (Extended)

This command is used to write the emulated card content.

Command

Command	Class	INS	P1	P2	Lc		Data In				
Write Card Emulation Data	E0h	00h	01h	60h	Length + 05h	01h	NFC Mode	Start Offset Bit[15:8]	Start Offset Bit[7:0]	Length	Data

Response Code

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

NFC Device Mode: 1 Byte

Status	Description
02h	NFC Forum Type 2 Tag Mode
Other	Card Read/Write Mode

Start Offset: 2 Byte – Address start to write

Length: 1 Byte – No. of byte to write

6.4.1.11.6. Set Card Emulation of NFC Forum Type 2 Tag ID

This command sets the UID of the emulated MIFARE Ultralight card.

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Card Emulation Lock Data	E0h	00h	00h	61h	03h	3 bytes UID

Response Code

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

6.4.1.11.7. Set Card Emulation Lock Data in NFC

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

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Card Emulation Lock Data	E0h	00h	00h	65h	01h	Lock

Response Code

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

Lock: 1 Byte – Protect the data from being overwritten via NFC

Operating Parameter	Parameter	Description	Option
Bit 7 ~ 2	Reserved	Reserved	
Bit 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
Bit 0	NFC Forum Type 2 Tag Enable		

6.4.1.11.8. Get Card Emulation Status

This command is used to get the status of card emulation data in NFC communication.

Command

Command	Class	INS	P1	P2	Lc
Get Card Emulation Status	E0h	00h	00h	69h	00h

Response Code

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

Status: 1 Byte

Operating Parameter	Mode	Description
Bit 7 ~ 6	Reserved	Reserved
Bit 5	EmulatedCard is activated	1 = Activated
Bit 4	EmulatedCard is removed	1 = Card is removed
Bit 3	EmulatedCard is read all	1 = All data is read
Bit 2	EmulatedCard is read	1 = Data is read
Bit 1	EmulatedCard is written	1 = Data is written
Bit 0	EmulatedCard is detected	1 = Card is detecting

6.4.1.11.9. Example Command Set of Emulating NFC Forum Type 2 Tag Mode

The command set is to trigger ACS website https://www.acs.com.hk by using ACR1552U to emulate as the NFC forum type 2 tag mode. The steps are showed below:

1. Enter the card emulation mode with below command:
   - Send **Error! Reference source not found.**

     E0 00 00 40 03 01 00 00
2. Write the NDEF data with below command:
   - Send **Error! Reference source not found.**

     E0 00 00 60 1C 01 01 00 18 E1 10 20 0F 03 0F D1 01 0B 55 02 61 63 73 2E 63 6F 6D 2E 68 6B FE 00 00

Notes:
For more detailed information and specifications related to the NDEF (NFC Data Exchange Format), I would recommend referring to the NDEF specification. It provides comprehensive guidelines and details about the structure and usage of NDEF records, which are commonly used in NFC data exchange. The NDEF specification will provide a deeper understanding of how to interpret and utilize the NDEF command and data in the context of the ACR1552U device.

6.4.1.12. Escape Command for PICC – Discovery Mode

6.4.1.12.1. Enter Discovery Mode

This command is used to enter the discovery mode.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Enter Discovery Mode	E0h	00h	00h	6Ah	01h	Discovery Mode

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Discovery Mode

Discovery Mode: 1 Byte

Status	Description
00h	Card Reader Mode
02h	NFC Forum Type 2 Tag Mode
03h	FeliCa

6.4.2. Escape Command for ICC

6.4.2.1. Get Exclusive Mode [E0 00 00 2B 00]

This command is used to get the reader exclusive mode setting.

Command

Command	Class	INS	P1	P2	Le
Get Exclusive Mode	E0h	00h	00h	2Bh	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Exclusive Mode

6.4.2.2. Set Exclusive Mode [E0 00 00 2B 01 …]

This command is used to configure the reader in to/out from exclusive mode.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Exclusive Mode	E0h	00h	00h	2Bh	01h	Exclusive Mode

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Exclusive Mode

Exclusive Mode (1 byte)

Exclusive Mode	Description
00h	Share Mode: ICC and PICC interfaces can work at the same time.
01h	Exclusive Mode: PICC is disabled when Auto Polling and Antenna Power Off when ICC is inserted (Default).
Other	RFU

Default Setting – 01h (Exclusive Mode)

6.4.2.3. Get Card Power Config [E0 00 00 0B 00]

This command is used get the ICC Card Power Configuration. This command should only be used for initial reader configuration.

Command

Command	Class	INS	P1	P2	Le
Get Card Power Config	E0h	00h	00h	0Bh	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Card Power Config

6.4.2.4. Set Card Power Config [E0 00 00 0B 01 …]

This command is used set and save the ICC Card Power Configuration. This command should only be used for initial reader configuration.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Card Power Config	E0h	00h	00h	0Bh	01h	Config

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	Card Power Config

Card Power Config (1 byte)

Card Power Config	Description
00h	Auto Detect, 1.8V -> 3V -> 5V
01h	5V Only
02h	3V Only
03h	1.8V Only
04h	Auto Detect, 5V -> 3V -> 1.8V
Other	RFU

Default Setting – 04h (Auto Detect, 5V -> 3V -> 1.8V)

6.4.3. Escape Command for Peripheral Control and Other

6.4.3.1. Get Firmware Version [E0 00 00 18 …]

This command is used to get reader's firmware message.

Command

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

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	Length of Firmware Version	Firmware Version

Example:
Command: E0 00 00 18 00
Response Code: E1 00 00 00 12 41 43 52 31 35 38 31 20 46 57 20 31 2E 30 30
Firmware Version in Hex: 41 43 52 31 35 38 31 20 46 57 20 31 2E 30 30
Firmware Version in ASCII: ACR1581 FW 1.00

6.4.3.2. Get Serial Number [E0 00 00 33 00]

This command is used to get the serial number.

Command

Command	Class	INS	P1	P2	Le
Get Serial Number	E0h	00h	00h	33h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	Length of Serial No.	Serial No.

6.4.3.3. Set S/N in USB Descriptor [E0 00 00 F0]

This command is used to Set S/N in USB Descriptor.

Command

Command	Class	INS	P1	P2	Le	Data In	
Set S/N in USB Descriptor	E0h	00h	00h	F0h	02h	00h	Enable SN in USB Descriptor

Response Code

Response	Class	INS	P1	P2	Le	Data Out		
Result	E1h	00h	00h	00h	03h	Enable SN in USB Descriptor	90h	00h

Enable SN in USB Descriptor (1 byte)

Enable SN in USB Descriptor	Description
00h	Disable SN in USB Descriptor
01h	Enable SN in USB Descriptor

6.4.3.4. Set Buzzer Control - Single Time [E0 00 00 28 01 …]

This command is used to set a single buzzer

Command

Command	Class	INS	P1	P2	Lc	Data Out
Buzzer Control	E0h	00h	00h	28h	01h	BUZ Status

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	BUZ Status

Buzzer Status (1 byte)

Buzzer Status	Description
00h	Off
01 ~ FFh	On with duration in 10ms unit

6.4.3.5.

6.4.3.5. Set Buzzer Control - Repeatable [E0 00 00 28 03 …]

This command is used to set period of buzzer

Command

Command	Class	INS	P1	P2	Lc	Data Out
Buzzer Control	E0h	00h	00h	28h	03h	BUZ Status

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	03h	BUZ Status

Buzzer Status (3 byte)

Operating Parameter	Buzzer Status	Description
Param 1 – Byte 0	On Time Period	01 ~ FF: On Duration in 10ms unit
Param 2 – Byte 1	Off Time Period	01 ~ FF: Off Duration in 10ms unit
Param 3 – Byte 2	Time for Repeating	01 ~ FF: Number to Repeat

6.4.3.6. Get LED Status [E0 00 00 29 00]

This command is used to get the current LED status

Command

Command	Class	INS	P1	P2	Le
Get LED Status	E0h	00h	00h	29h	00h

Response Code

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

6.4.3.7. Set LED Control [E0 00 00 29 01 …]

This command is used to set LED control

Command

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

Response Code

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

LED Status (1 byte)

LED Status	Description
Bit 0 : Blue LED	1 = On ; 0 = Off
Bit 1 : Green LED	1 = On ; 0 = Off
RFU	Other

6.4.3.8. Get UI Behaviour [E0 00 00 21 00]

This command is used to get the PCD UI Behaviour but save the setting without another command. This command should only be used for initial reader configuration.

Command

Command	Class	INS	P1	P2	Le
Get PICC UI Behaviour	E0h	00h	00h	21h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E0h	00h	00h	00h	01h	PICC/ICC UI Behaviour

6.4.3.9. Set UI Behaviour [E0 00 00 21 01 …]

This command is used to set the PICC/ICC UI behaviour.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set PICC UI Behaviour	E0h	00h	00h	21h	01h	PICC/ICC UI Behaviour

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E0h	00h	00h	00h	01h	PICC/ICC UI Behaviour

UI Behaviour - 1 Byte, Bit Mask of following

Operating Parameter	Parameter	Description	Option
Bit 0	Accessing(LED Fast Blinking)	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit 3	Presence Event (Short Buzzer Beep)		
Bit 4	Card Removal Event (Short Buzzer Beep)		

Default Setting For PICC – 09h

Default Setting For ICC – 09h

Notes:

1. The Get/Set UI behaviour are excluding on SAM interface.