Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1281S-C1 Serial Dual Interface Reader

Communication Protocol V1.01

Table of Contents

List of Tables

1.0. Introduction

The ACR1281S-C1 serial protocol defines the interface between the PC and reader, as well as the communication channel between the PC and the supported cards – ISO 14443 compliant contactless cards (PICC) and ISO 7816 compliant full-sized (ICC) and SIM-sized (SAM) contact cards.

1.1. Features

- Serial RS232 Interface: Baud Rate = 9.6 kbps (default), 19.2 kbps, 38.4 kbps, 57.6 kbps, 115.2 kbps, 230.4 kbps
- USB interface for power supply
- CCID-like frame format (Binary format)
- Contactless Smart Card Reader:
    - Read/write speed of up to 848 kbps
    - Built-in antenna for contactless tag access, with card reading distance of up to 50 mm (depending on tag type)
    - Supports ISO 14443 Part 4 Type A and B cards and Mifare series
    - Built-in anti-collision feature (only one tag is accessed at any time)
    - Supports extended APDU (max. 64 kbytes)
- Contact Smart Card Reader:
    - Supports ISO 7816 Class A, B and C (5 V, 3V and 1.8 V)
    - Supports microprocessor cards with T=0 or T=1 protocol
    - Supports memory cards
    - ISO 7816 compliant SAM slot
- Built-in Peripherals:
    - Two user-controllable LEDs
    - User-controllable buzzer
- USB Firmware Upgradability
- Compliant with the following standards:
    - ISO 14443
    - ISO 7816
    - CE
    - FCC
    - RoHS

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

1.2. Serial Interface

The ACR1281S-C1 is connected to a computer through a Serial Interface (RS232 or RS485).

1.2.1. Communication Parameters

The ACR1281S-C1 is connected to a host through serial interface (RS232 or RS485), Supported Baud Rate: 9,600 bps (default), 19,200 bps, 38,400 bps, 57,600 bps, 115,200 bps and 230,400 bps.

Pin	Signal	Function
1	VCC	+5 V power supply for the reader
2	TXD	The signal from the host to the reader
3	RXD	The signal from the reader to the host
4	GND	Reference voltage level for power supply

Table 1: RS232 Interface Wiring

Pin	Signal	Function
1	VCC	+5 V power supply for the reader
2	A	Differential signal transmits data between the reader and host
3	B	Differential signal transmits data between the reader and host
4	GND	Reference voltage level for power supply

Table 2: RS485 Interface Wiring

1.3. Serial Protocol

ACR1281S-C1 shall interface with the host with serial connection. CCID-like format is used for communication.

The Command Format as below:

STX (0x02h)	Bulk-OUT Header	APDU Command or Parameters	Checksum	ETX (0x03h)
1 Byte	10 Bytes	M Bytes (if applicable)	1 Byte	1 Byte

Where:

STX – Start of Text, tells the reader start to receive the command, must equal to 0x02h

ETX – End of Text, tells the reader the command ended, must equal to 0x03h

Bulk-OUT Header – 10bytes CCID-liked Header

APDU Command or Parameter – APDU command or parameter for accessing reader and card

Checksum – error checking, equal to XOR {Bulk-OUT Header, APDU Command or Parameters}

After ACR1281S receives the command, ACR1281S will first response the status frame to tell the host the command status.

The Status Frame Format as below:

STX (0x02h)	Status	Checksum	ETX (0x03h)
1 Byte	1 Byte	1 Byte	1 Byte

Note: *Checksum = Status*

There are several cases that may occur:

Case1 ACK Frame = {02 00 00 03h}
Inform the HOST that the frame is correctly received. The HOST has to wait for the response of the command. The ACR1281S will not receive any more frames while the command is being processed.

Case2 Checksum Error Frame = {02 FF FF 03h}
The received data checksum is incorrect.

Case3 Length Error Frame = {02 FE FE 03h}
The data length is greater than 275 bytes.

Case4 ETX Error Frame = {02 FD FD 03h}
The last byte is not equal to ETX “0x03h”.

Case5 Time out Error Frame = {02 99 99 03h}
No data receive for a long time.

NAK Frame = {02 00 00 00 00 00 00 00 00 00 00 00 03h} // 11 zeros

Used by the HOST to get the last response or card insertion/ removal event messages.

If the frame is correctly received (e.g., ACK Frame received by Host), the response frame will be sent by ACR1281S followed.

The Response Frame Format as below:

STX (0x02h)	Bulk-IN Header	APDU Response or abData	Checksum	ETX (0x03h)
1 Byte	10 Bytes	N Bytes (If applicable)	1 Byte	1 Byte

Where:

STX – Start of Text, tells the host to receive the response, must be equal to 0x02h

ETX – End of Text, tells the host the response ended, must be equal to 0x03h

Bulk-IN Header – 10bytes CCID-like header, please refer to **Section 1.4 – CCID-like Commands**

APDU Response or abData – APDU response or data from accessed command

Checksum – error checking, equal to XOR {Bulk-OUT Header, APDU Response or abData}

1.4. CCID-like Commands

1.4.1. Bulk-OUT Messages

ACR1281S shall follow the CCID Bulk-OUT Messages as specified in CCID Section 4. In addition, this specification defines some extended commands for operating additional features. This section lists the CCID Bulk-OUT Messages to be supported by ACR1281S.

1.4.1.1. PC_to_RDR_IccPowerOn

This command activates the card slot and returns ATR from the card.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	62h	
1	*dwLength*	4	00000000h	Size of extra bytes of this message.
2	*bSlot*	1		Identifies the slot number for this command. For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
5	*bSeq*	1		Sequence number for command.
6	*bPowerSelect*	1		Voltage that is applied to the ICC. 00h – Automatic Voltage Selection 01h – 5 V 02h – 3 V
7	*abRFU*	2		Reserved for future use.

The response to this message is the *RDR_to_PC_DataBlock* message and the data returned is the *Answer to Reset (ATR)* data.

Note: *The ICC and SAM interface must be activated before accessing contact cards.*

1.4.1.2. PC_to_RDR_IccPowerOff

This command deactivates the card slot.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	63h	
1	*dwLength*	4	00000000h	Size of extra bytes of this message.
5	*bSlot*	1		Identifies the slot number for this command For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
6	*bSeq*	1		Sequence number for command.

Offset	Field	Size	Value	Description
7	*abRFU*	3		Reserved for future use.

The response to this message is the *RDR_to_PC_SlotStatus* message.

1.4.1.3. PC_to_RDR_GetSlotStatus

This command gets the current status of the slot.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	65h	
1	*dwLength*	4	00000000h	Size of extra bytes of this message.
5	*bSlot*	1		Identifies the slot number for this command. For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
6	*bSeq*	1		Sequence number for command.
7	*abRFU*	3		Reserved for future use.

The response to this message is the *RDR_to_PC_SlotStatus* message.

1.4.1.4. PC_to_RDR_XfrBlock

This command transfers data block to the ICC.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	6Fh	
1	*dwLength*	4		Size of abData field of this message.
5	*bSlot*	1		Identifies the slot number for this command. For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
6	*bSeq*	1		Sequence number for command.
7	*bBWI*	1		Used to extend the CCIDs Block Waiting Timeout for this current transfer. The CCID will timeout the block after “this number multiplied by the Block Waiting Time” has expired.
8	*wLevelParameter*	2	0000h	RFU (TPDU exchange level).
10	*abData*	Byte array		Data block sent to the CCID. Data is sent “as is” to the ICC (TPDU exchange level).

The response to this message is the *RDR_to_PC_DataBlock* message.

1.4.1.5. PC_to_RDR_Escape

This command is used to access extended features.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	6Bh	
1	*dwLength*	4		Size of *abData* field of this message.
5	*bSlot*	1		Identifies the slot number for this command. For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
6	*bSeq*	1		Sequence number for command.
7	*abRFU*	3		Reserved for future use.
10	*abData*	Byte array		Data block sent to the CCID.

The response to this command message is the *RDR_to_PC_Escape* response message

1.4.2. Bulk-IN Messages

The Bulk-IN messages are used in response to the Bulk-OUT messages. ACR1281S shall follow the CCID Bulk-IN Messages as specified in CCID section 4. This section lists the CCID Bulk-IN Messages to be supported by ACR1281S.

1.4.2.1. RDR_to_PC_DataBlock

This message is sent by ACR1281S in response to *PC_to_RDR_IccPowerOn* and *PC_to_RDR_XfrBlock* messages.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	8h	Indicates that a data block is being sent from the CCID.
1	*dwLength*	4		Size of extra bytes of this message.
5	*bSlot*	1		Same value as in Bulk-OUT message. For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
6	*bSeq*	1		Same value as in Bulk-OUT message.
7	*bStatus*	1		Slot status register as defined in CCID Section 4.2.1.
8	*bError*	1		Slot error register as defined in CCID Section 4.2.1.
9	*bChainParameter*	1	00h	RFU (TPDU exchange level).
10	*abData*	Byte array		This field contains the data returned by the CCID.

1.4.2.2. RDR_to_PC_Escape

This message is sent by ACR1281S in response to *PC_to_RDR_Escape* messages.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	83h	
1	*dwLength*	4		Size of abData field of this message.
5	*bSlot*	1		Same value as in Bulk-OUT message. For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
6	*bSeq*	1		Same value as in Bulk-OUT message.
7	*bStatus*	1		Slot status register as defined in CCID Section 4.2.1.
8	*bError*	1		Slot error register as defined in CCID Section 4.2.1.
9	*bRFU*	1	00h	RFU.
10	*abData*	Byte array		This field contains the data returned by the CCID.

1.4.2.3. RDR_to_PC_SlotStatus

This message is sent by ACR1281S in response to *PC_to_RDR_IccPowerOff*, *PC_to_RDR_GetSlotStatus* messages and Class specific ABORT request.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	81h	
1	*dwLength*	4	00000000h	Size of extra bytes of this message.
5	*bSlot*	1		Same value as in Bulk-OUT message. For SAM interface, *bSlot* = 2. For ICC interface, *bSlot* = 1. For PICC interface, *bSlot* = 0.
6	*bSeq*	1		Same value as in Bulk-OUT message.
7	*bStatus*	1		Slot status register as defined in CCID Section 4.2.1.
8	*bError*	1		Slot error register as defined in CCID Section 4.2.1.
9	*bClockStatus*	1		Value: 00h = Clock running 01h = Clock stopped in state L 02h = Clock stopped in state H 03h = Clock stopped in an unknown state All other values are RFU.

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

2.0. Contact Smart Card Protocol

Pseudo APDUs are for accessing memory tag communication and peripherals.

The pseudo APDUs should be sent via *PC_to_RDR_XfrBlock* with *bSlot* = 1.

2.1.1. Memory Card – 1, 2, 4, 8, 16 kbits I2C Card

2.1.1.1. Select Card Type

This command powers down and up the selected card that is inserted to the card reader, and performs a card reset.

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	01h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.1.2. Select Page Size

This command chooses the page size to read the smart card. The default value is an eight-byte page write. It will reset to default value whenever the card is removed or the reader is powered off.

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Page size
FFh	01h	00h	00h	01h	

Where:

Page size (1Byte) = 03h for 8-byte page write
= 04h for 16-byte page write
= 05h for 32-byte page write
= 06h for 64-byte page write
= 07h for 128-byte page write

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.1.3. Read Memory Card

This command reads the memory card from a specified address location.

Command Format

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B0h			

Where:

Byte Address (2 bytes) = Memory address location of the memory card

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

Response Data Format

BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.1.4. Write Memory Card

This command writes on the memory card from a specified address location.

Command Format

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	...	...	Byte N
		MSB	LSB					
FFh	D0h							

Where:

Byte Address (2 Bytes) = Memory address location of the memory card

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

BYTE (1…N) = Data to be written to the memory card

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.2. Memory Card – 32, 64, 128, 256, 512, 1024 kbits I2C Card

2.1.2.1. Select Card Type

This command powers down and up the selected card that is inserted to the card reader, and performs a card reset.

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	02h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.2.2. Select Page Size

This command chooses the page size to read the smart card. The default value is an eight-byte page write. It will reset to default value whenever the card is removed or the reader is powered off.

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Page size
FFh	01h	00h	00h	01h	

Where:

Page size (1Byte) = 03h for 8-byte page write
= 04h for 16-byte page write
= 05h for 32-byte page write
= 06h for 64-byte page write
= 07h for 128-byte page write

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.2.3. Read Memory Card

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

Command Format

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh				

Where:

INS (1 byte): For 32, 64, 128, 256, 512 kbit I2C card, INS = 0xB0h
For 1024kbit I2C card, INS = 1011 000* b
where * is the MSB of the 17 bit addressing

Byte Address (2 Bytes) = Memory address location of the memory card

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

Response Data Format

BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.2.4. Write Memory Card

This command writes on the memory card in a specified address location.

Command Format

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	…	…	Byte N
		MSB	LSB					
FFh								

Where:

INS (1 byte): For 32, 64, 128, 256, 512 kbit I2C card, INS = 0xD0h
For 1024 kbit I2C card, INS = 1101 000* b
where * is the MSB of the 17 bit addressing

Byte Address (2 Bytes) = 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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.3. Memory Card – ATMEL AT88SC153

2.1.3.1. Select Card Type

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

Command Format

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

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.3.2. Read Memory Card

This command reads the memory card in a specified address location.

Command Format

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh		00h		

Where:

INS (1 Byte): For reading zone 00b, INS = 0xB0h
For reading zone 01b, INS = 0xB1h
For reading zone 10b, INS = 0xB2h
For reading zone 11b, INS = 0xB3h
For reading fuse, INS = 0xB4h

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

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

Response Data Format

BYTE 1	...	...	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.3.3. Write Memory Card

This command writes on the memory card in a specified address location.

Command Format

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	…	…	Byte N
FFh		00h						

Where:

INS (1 Byte): For reading zone 00b, INS = 0xD0h
For reading zone 01b, INS = 0xD1h
For reading zone 10b, INS = 0xD2h
For reading zone 11b, INS = 0xD3h
For reading fuse, INS = 0xD4h

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

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

BYTE (1…N) = Data to be written to the memory card

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.3.4. Verify Password

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

Command Format

Pseudo-APDU							
CLA	INS	P1	P2	Lc	PW (0)	PW (1)	PW (2)
FFh	20h	00h		03h			

Where:

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

P2 (1 Byte) = 0000 00r p b

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 = 01 for the secure code

Response Data Format

SW1	ErrorCnt
90	

Where:

ErrorCnt (1 Byte) = Error Counter

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

2.1.3.5. Initialize Authentication

This command initializes the memory card's authentication.

Command Format

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Q (0)	Q (1)	…	Q (7)
FFh	84h	00h	00h	08h				

Where:

Q (0…7) = Host random number, 8 bytes

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.3.6. Verify Authentication

This command verifies the memory card's authentication.

Command Format

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Ch (0)	Ch (1)	…	Ch (7)
FFh	82h	00h	00h	08h				

Where:

Ch (0…7) = Host challenge, 8 bytes

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.4. Memory Card – ATMEL AT88SC1608

2.1.4.1. Select Card Type

This command powers down and up the selected card that is inserted in the card reader, and performs a card reset. It will also select the page size to be 16-byte page write.

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	04h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.4.2. Read Memory Card

This command reads the memory card in the specified address location.

Command Format

Pseudo-APDU				
CLA	INS	Zone Address	Byte Address	MEM_L
FFh				

Where:

INS (1 Byte): For reading user zone, INS = 0xB0h

For reading configuration zone or reading fuse, INS = 0xB1h

Zone Address (1Byte) = 00000 A10 A9 A8b, where A10 is the MSB of zone address

**don't care for reading fuse

Byte Address (1Byte) = 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 (1Byte) = Length of data to be read from the memory card

Response Data Format

BYTE 1	...	...	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.4.3. Write Memory Card

This command writes to the memory card on a specified address location.

Command Format

Pseudo-APDU								
CLA	**INS**	**Zone Address**	**Byte Address**	**MEM_L**	**Byte 1**	**...**	**...**	**Byte N**
FFh								

Where:

INS (1 Byte): For reading user zone, INS = 0xD0h

For reading configuration zone or reading fuse, INS = 0xD1h

Zone Address (1Byte) = 00000 A10 A9 A8b, where A10 is the MSB of zone address

** don't care for reading fuse

Byte Address (1Byte) = 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 (1Byte) = Length of data to be written to the memory card

BYTE (1…N) = Data to be written to the memory card

Response Data Format

SW1	**SW2**

Where:

SW1, SW2 = 90 00h if no error

2.1.4.4. Verify Password

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

Command Format

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

Where:

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

RP (1 Byte) = 0000 r p2 p1 p0 b

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 = 0111 for the secure code

Response Data Format

SW1	ErrorCnt
90h	

Where:

ErrorCnt (1 Byte) = Error Counter

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

2.1.4.5. Initialize Authentication

This command initializes the memory card's authentication.

Command Format

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Q (0)	Q (1)	…	Q (7)
FFh	84h	00h	00h	08h				

Where:

Q (0…7) = Host random number, 8 bytes

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.4.6. Verify Authentication

This command verifies the memory card's authentication.

Command Format

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Ch (0)	Ch (1)	…	Ch (7)
FFh	82h	00h	00h	08h				

Where:

Ch (0…7) = Host challenge, 8 bytes

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

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

2.1.5.1. Select Card Type

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

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	05h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.5.2. Read Memory Card

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

Command Format

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B0h			

Where:

MSB Byte Address (1Byte) = 0000 00 A9 A8b is the memory address location of the memory card

LSB Byte Address (1Byte) = A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

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

Response Data Format

BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.5.3. Presentation Error Counter Memory Card (only SLE4428 and SLE5528)

This command is used to read the presentation error counter for the secret code.

Command Format

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	B1h	00h	00h	03h

Response Data Format

ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

Where:

ERRCNT (1Byte) = the value of the presentation error counter. "FFh" indicates the last verification is correct. "00h" indicates the password is locked (exceeded the maximum number of retries). Other values indicate the last verification has failed

DUMMY1, DUMMY2 (2Byte) = Two bytes dummy data read from the card

SW1, SW2 = 90 00h if no error

2.1.5.4. Read Protection Bit

This command is used to read the protection bit.

Command Format

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B2h			

Where:

MSB Byte Address (1Byte) = 0000 00 A9 A8b is the memory address location of the memory card

LSB Byte Address (1Byte) = A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

MEM_L (1Byte) = Length of protection bits to be read from the card, in multiple of 8 bits (Maximum value is 32)

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

For example, to read eight protection bits starting from memory 0x0010h, the following pseudo-APDU should be issued: 0xFFh 0xB1h 0x00h 0x10h 0x01h.

Response Data Format

PROT 1	…	…	PROT L	SW1	SW2

Where:

PROT (1…L) = Bytes containing the protection bits

SW1, SW2 = 90 00h if no error

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 the response data

'0' byte is write protected

'1' byte can be written

2.1.5.5. Write Memory Card

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

Command Format

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	…	...	Byte N
		MSB	LSB					
FFh	D0h							

Where:

MSB Byte Address (1Byte) = 0000 00 A9 A8b is the memory address location of the memory card

LSB Byte Address (1Byte) = A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card

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

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

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.5.6. Write Protection Memory Card

Each of the bytes specified in the command is internally in the card compared with the byte stored at the specified address. If the data match, the corresponding protection bit is irreversibly programmed to '0'.

Command Format

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	...	...	Byte N
		MSB	LSB					
FFh	D1h							

Where:

MSB Byte Address (1Byte)	= 0000 00 A9 A8b is the memory address location of the memory card
LSB Byte Address (1Byte)	= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card
MEM_L (1Byte)	= Length of data to be written to the memory card
Byte (1…N)	= Byte values to be compared with the data in the card starting at 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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.5.7. Present Code Memory Card (only SLE 4428 and SLE5528)

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

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

Command Format

Pseudo-APDU						
CLA	INS	P1	P2	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	00h	00h	02h		

Where:

CODE (2 Bytes) = secret code (PIN)

Response Data Format

SW1	ErrorCnt
90h	

Where:

ErrorCnt (1 Byte) = Error Counter. "FFh" indicates the verification is correct. "00h" indicates the password is locked (exceeded the maximum number of retries). Other values indicate the current verification has failed.

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

2.1.6.1. Select Card Type

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

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	06h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.6.2. Read Memory Card

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

Command Format

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
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 Data Format

BYTE 1	…	…	BYTE N	PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

PROT (1…4) = Bytes containing the protection bits from protection memory

SW1, SW2 = 90 00h if no error

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 the response data

'0' byte is write protected

'1' byte can be written

2.1.6.3. Read Present Error Counter Memory Card (only SLE4442 and SLE5542)

This command is used to read the presentation error counter for the secret code.

Command Format

Pseudo-APDU				
CLA	**INS**	**P1**	**P2**	**MEM_L**
FFh	B1h	00h	00h	04h

Response Data Format

ERRCNT	DUMMY 1	DUMMY 2	DUMMY 3	SW1	SW2

Where:

ERRCNT (1 Byte) = The value of the presentation error counter. "07h" indicate the last verification is correct. "00h" indicates the password is locked (exceeded the maximum number of retries). Other values indicate the last verification has failed.

DUMMY1, DUMMY2, DUMMY3 (3 Byte) = dummy data read from the card

SW1, SW2 = 90 00h if no error

2.1.6.4. Read Protection Bits

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

Command Format

Pseudo-APDU				
CLA	**INS**	**P1**	**P2**	**MEM_L**
FFh	B2h	00h	00h	04h

Response Data Format

PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

Where:

PROT (1…4) = Bytes containing the protection bits from protection memory

SW1, SW2 = 90 00h if no error

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 the response data

'0' byte is write protected

'1' byte can be written

2.1.6.5. Write Memory Card

This command writes on the memory card's content in a specified address location.

Command Format

Pseudo-APDU								
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**...**	**...**	**Byte N**
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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.6.6. Write Protection Memory Card

Each of the bytes specified in the command is internally in the card compared with the byte stored at the specified address. If the data match, the corresponding protection bit is irreversibly programmed to '0'.

Command Format

Pseudo-APDU								
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**...**	**...**	**Byte N**
FFh	D1h	00h						

Where:

Byte Address (1 Byte) = 000A4 A3 A2 A1 A0b (00h to 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 to be compared with the data in the card starting at 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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.6.7. Present Code Memory Card (only SLE 4442 and SLE5542)

This command is used to submit 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 the bit to '0'.
2. Present the specified code to the card.
3. Try to erase the presentation error counter.

Command Format

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	03h			

Where:

CODE (3 Byte) = secret code (PIN)

Response Data Format

SW1	ErrorCnt

Where:

ErrorCnt (1 Byte) = Error Counter. "07h" indicate the verification is correct. "00h" indicates the password is locked (exceeded the maximum number of retries). Other values indicate the current verification has failed.

2.1.6.8. Change Code Memory Card (only SLE 4442 and SLE5542)

This command is used to write the specified data as new secret code in the card. The current secret code must be presented to the card with the PRESENT_CODE command prior to the execution of this command.

Command Format

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	D2h	00h	01h	03h			

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

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

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

2.1.7.1. Select Card Type

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

Command Format

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	07h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.7.2. Read Memory Card

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

Command Format

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	B0h	00h		

Where:

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

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

Response Data Format

BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.7.3. 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 at 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:

1. **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.
2. **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.
3. **Write with backup enabled (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.
4. **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 Format

Pseudo-APDU						
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**MODE**	**BYTE**
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

0x00h: write

0x01h: write with carry

0x02h: write with backup enabled (SLE4436, SLE5536 and SLE6636 only)

0x03h: write with carry and with backup enabled (SLE4436, SLE5536 and SLE6636 only)

BYTE (1 Byte) = Byte value to be written to the card

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.7.4. Present Code Memory Card

The command is used to submit the secret code to the memory card to enable the card personalization mode. The following actions are executed:

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

The ACR1281S does not try to erase the presentation counter after the code submission. This must be done by the application software through a separate '*Write with carry*' command.

Command Format

Pseudo-APDU								
CLA	INS	P1	P2	MEM_L	CODE			
					ADDR	Byte 1	Byte 2	Byte 3
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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.7.5. Authenticate Memory Card (SLE4436, SLE5536 and SLE6636 only)

This command is used to read a card authentication certificate from a SLE5536 or SLE6636 card. The following actions are executed by the ACR1281S:

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 of 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 has to be 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 Format

Pseudo-APDU											
CLA	INS	P1	P2	MEM_L	CODE						
					KEY	CLK_CNT	Byte1	Byte 2	…	Byte 5	Byte 6
FFh	84h	00h	00h	08h							

Where:

KEY (1 Byte) = Key to be used for the computation of the authentication certificate:

0x00h = key 1 with no cipher block chaining
0x01h = key 2 with no cipher block chaining
0x80h = key 1 with cipher block chaining (SLE5536 and SLE6636 only)
0x81h = key 2 with cipher block chaining (SLE5536 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 Data Format

SW1	SW2
61h	02h

If there is no error, it means two bytes of authentication data are ready. The authentication data can be retrieved by *GET_RESPONSE* command.

Step 2: Get back the *Authentication Data* (GET_RESPONSE).

Command Format

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	C0h	00h	00h	02h

Response Data Format

CERT		SW1	SW2

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 no error

2.1.8. Memory Card – SLE4404

2.1.8.1. Select Card Type

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

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	08h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.8.2. Read Memory Card

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

Command Format

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
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 Data Format

BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.8.3. Write Memory Card

This command is used to write data on a 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. Thus, memory bits can only be programmed from state '1' to state '0'.

Command Format

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	...	...	Byte N
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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.8.4. Erase Scratch Pad Memory Card

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

Command Format

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FFh	D2h	00h		00h

Where:

Byte Address (1 Byte) = Memory byte address location of the scratch pad
Typical value is 0x02h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.8.5. Verify User Code

This command is used to submit a User Code (2 bytes) to the inserted card. The User Code is used 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 User Error Counter can be erased when the submitted code is correct.

Command Format

Pseudo-APDU						
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
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 Data Format

SW1	SW2

SW1, SW2 = 90 00h if no error.

= 63 00h if there is no more retry chance

Note: *After SW1SW2 = 90 00h is received, read back the User Error Counter to check if the VERIFY_USER_CODE is correct. If User Error Counter is erased and is equal to "FFh," the previous verification is successful.*

2.1.8.6. Verify Memory Code

This command is used to submit Memory Code (4 bytes) to the inserted card. 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. Please note that Memory Error Counter cannot be erased.

Command Format

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
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 Byte) = Memory Code

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

= 63 00h if there is no more retry chance

Note: *After SW1SW2 = 0x9000h is received, read back the Application Area to check if the VERIFY_MEMORY_CODE is correct. If all data in Application Area is erased and is equal to "FFh," the previous verification is successful.*

2.1.9. Memory Card – AT88SC101/AT88SC102/AT88SC1003

2.1.9.1. Select Card Type

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

Command Format

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FFh	A4h	00h	00h	01h	09h

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.9.2. Read Memory Card

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

Command Format

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
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 Data Format

BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE (1…N) = Data read from memory card

SW1, SW2 = 90 00h if no error

2.1.9.3. Write Memory Card

This command is used to write 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. Thus, memory bits can only be programmed from '1' to '0'.

Command Format

Pseudo-APDU								
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**…**	**...**	**Byte N**
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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.9.4. Erase Non-Application Zone

This command is used to erase the data in Non-Application Zones. The EEPROM memory is organized into 16-bit words. Although erasures are performed on a single bit, the ERASE operation clears an entire word in the memory. Therefore, performing an Erase command 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 Format

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	D2h	00h		00h

Where:

Byte Address (1 Byte) = Memory byte address location of the word to be erased

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

2.1.9.5. 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 Format

Pseudo-APDU									
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE				
					Byte 1	Byte 2	…	…	Byte N
FFh	20h	00h							

Where:

Error Counter LEN (1 Byte) = Length of presentation error counter in bits. The value should be 0x00h 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

Case	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 Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

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

2.1.9.6. 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 un-blown 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 Format

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
FFh	20h	80h		04h				

Where:

Error Counter LEN (1 Byte) = Length of presentation error counter in bits. The value should always be 80h.

Byte Address (1 Byte) = Byte address of the Application Zone Key in the card.

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

Where:

CODE (4 Byte) = Erase Key

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

= 63 00h if there is no more retry chance

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

2.1.9.7. Verify Security Code

This command is used to submit Security Code (2 bytes) to the inserted card. Security Code is used 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 Format

Pseudo-APDU						
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
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 Byte) = Security Code

Response Data Format

SW1	SW2

Where:

SW1, SW2 = 90 00h if no error

= 63 00h if there is no more retry chance

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

2.1.9.8. Blown Fuse

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

Note: *The blowing of Fuse is an irreversible process.*

Command Format

Pseudo-APDU								
CLA	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
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 always be 0x01h.

State of RST Pin (1 Byte) = State of the RST pin. Please refer to below table for the correct value.

Case	Fuse	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

Response Data Format

SW1	SW2

Where

SW1, SW2 = 90 00h if no error

3.0. Contactless Smart Card Protocol

3.1.1. ATR Generation

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

3.1.1.1. ATR format for ISO 14443 Part 3 PICCs

Byte	Value (Hex)	Designation	Description
0	3Bh	Initial Header	
1	8Nh	T0	Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following. Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)
2	80h	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0
3	01h	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1
4 to 3+N	80h	T1	Category indicator byte, 80 means A status indicator may be present in an optional COMPACT-TLV data object
	4Fh	Tk	Application identifier Presence Indicator
	0Ch		Length
	RID		Registered Application Provider Identifier (RID) A0 00 00 03 06h
	SS		Byte for standard
	C0h .. C1h		Bytes for card name
	00 00 00 00h	RFU	RFU 00 00 00 00h
4+N	UUh	TCK	Exclusive-oring of all the bytes T0 to Tk

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

Length (YY) = 0x0Ch
RID = {A0 00 00 03 06h} (PC/SC Workgroup)
Standard (SS) = 03 (ISO 14443A, Part 3)
Card Name (C0 .. C1) = {00 01h} (Mifare 1K)

00 01h: Mifare 1K
00 02h: Mifare 4K
00 03h: Mifare Ultralight
00 26h: Mifare Mini
FF 28h: JCOP 30
FF [SAK]h: undefined tags

3.1.1.2. ATR format for ISO 14443 Part 4 PICCs

<table>
<tr><th>Byte</th><th>Value (Hex)</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 to 3 + N</td><td>XXh</td><td>T1</td><td rowspan="2">Historical Bytes:

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

ISO 14443B:
<table><tr><td>Byte1-4</td><td>Byte5-7</td><td>Byte8</td></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>XXh
XXh
XXh</td><td>Tk</td></tr>
<tr><td>4+N</td><td>UUh</td><td>TCK</td><td>Exclusive-oring of all the bytes T0 to Tk</td></tr>
</table>

Example 1: ATR for 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

3.1.2. Pseudo APDUs for Contactless Interface

Pseudo APDUs are used for accessing contactless tag communication and peripherals.

The pseudo APDUs should be sent via *PC_to_RDR_XfrBlock* with *bSlot* = 0.

3.1.2.1. 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 = 0x00h, Get UID Response Format (UID + 2 Bytes)

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

If P1 = 0x01h, Get ATS of a ISO 14443 A card (ATS + 2 Bytes)

Response	Data Out		
Result	ATS	SW1	SW2

Response Codes

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

Examples:

// To get the serial number of the "connected PICC"

UINT8 GET_UID[5]={0xFFh, 0xCAh, 0x00h, 0x00h, 0x00h};

// To get the ATS of the "connected ISO 14443 A PICC"

UINT8 GET_ATS[5]={0xFFh, 0xCAh, 0x01h, 0x00h, 0x00h};

3.1.2.2. PICC Commands (T=CL Emulation) for Mifare 1K/4K Memory Cards

3.1.2.2.1. Load Authentication Keys

The ***Load Authentication Keys*** command loads the authentication keys into the reader. The authentication keys are used to authenticate a particular sector of the Mifare 1K/4K memory card. Two kinds of authentication key locations are provided: volatile and non-volatile key locations.

Load Authentication Keys APDU Format (11 Bytes)

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

Where:

Key Structure (1 Byte): 0x00h = Key is loaded into the reader volatile memory
0x20h = Key is loaded into the reader non-volatile memory
Other = Reserved

Key Number (1 Byte): 0x00h ~ 0x1Fh = Non-volatile memory is used for storing keys. The keys are permanently stored in the reader and will not disappear even the reader is disconnected from the PC. It can store up to 32 keys inside the reader non-volatile memory.

0x20h (Session Key) = Volatile memory is used for storing a temporary key. The key will disappear once the reader is disconnected from the PC. Only one (1) volatile key is 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. Example: {FF FF FF FF FF FFh}

Load Authentication Keys Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Load Authentication Keys Response Codes

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

Example 1:

// Load a key {FF FF FF FF FF FFh} into the non-volatile memory location 0x05h.

APDU = {FF 82 20 05 06 FF FF FF FF FF FFh}

// Load a key {FF FF FF FF FF FFh} into the volatile memory location 0x20h.

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

APDU = {FF 82 00 20 06 FF FF FF FF FF FFh}

Example 2:

Notes:

1. *Basically, the application should know all the keys being used. It is recommended to store all the required keys to the non-volatile memory for security reasons. The contents of both volatile and non-volatile memories are not readable by the outside world.*
2. *The content of the volatile memory "Session Key 0x20h" will remain valid until the reader is reset or powered off. The session key is useful for storing any key value that is changing from time to time. The session key is stored in the "Internal RAM", while the non-volatile keys are stored in "EEPROM" that is relatively slower than "Internal RAM".*
3. *It is not recommended to use the "non-volatile key locations 0x00h ~ 0x1Fh" to store any "temporary key value" that will be changed so often. The "non-volatile keys" are supposed to be used for storing any "key value" that will not change frequently. If the "key value" is supposed to be changed from time to time, please store the "key value" to the "volatile key location 0x020h".*

3.1.2.2.2. Authentication for Mifare 1K/4K

The ***Authentication*** command uses the keys stored in the reader to perform authentication with the Mifare 1K/4K card (PICC). Two types of authentication keys are used: TYPE_A and TYPE_B.

Load Authentication Keys APDU Format (6 Bytes) (Obsolete)

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

Load Authentication Keys APDU Format (10 Bytes)

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

Authenticate Data Bytes (5 Byte):

Byte1	Byte 2	Byte 3	Byte 4	Byte 5
Version 0x01h	0x00h	Block Number	Key Type	Key Number

Where:

Block Number (1 Byte): The memory block to be authenticated. For Mifare 1K Card, it has a total of 16 sectors and each sector consists of four (4) consecutive blocks.

Example: Sector 0x00h consists of Blocks {0x00h, 0x01h, 0x02h and 0x03h}; Sector 0x01h consists of Blocks {0x04h, 0x05h, 0x06h and 0x07h}; the last sector 0x0Fh consists of Blocks {0x3Ch, 0x3Dh, 0x3Eh and 0x3Fh}. 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 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): 0x60h = Key is used as a TYPE A key for authentication

0x61h = Key is used as a TYPE B key for authentication

Key Number (1 Byte): 0x00h ~ 0x1Fh = Non-volatile memory is used for storing keys. The keys are permanently stored in the reader and will not disappear even the reader is disconnected from the PC. It can store 32 keys into the non-volatile memory of the reader.

0x20h (Session Key) = Volatile memory is used for storing keys. The keys will disappear when the reader is disconnected from the PC. Only one (1) volatile key is provided. The volatile key can be used as a session key for different sessions.

Load Authentication Keys Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Load Authentication Keys Response Codes

Results	SW1	SW2	Meaning
Success	90h	00h	The operation is completed successfully.
Error	63h	00h	The operation has 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	0x00h ~ 0x02h	0x03h
Sector 1	0x04h ~ 0x06h	0x07h
..		
..		
Sector 14	0x38h ~ 0x0Ah	0x3Bh
Sector 15	0x3Ch ~ 0x3Eh	0x3Fh

1K Bytes

Table 3: Mifare 1K Memory Map

Sectors (Total 32 sectors. Each sector consists of 4 consecutive blocks)	Data Blocks (3 blocks, 16 bytes per block)	Trailer Block (1 block, 16 bytes)
Sector 0	0x00h ~ 0x02h	0x03h
Sector 1	0x04h ~ 0x06h	0x07h
..		
..		
Sector 30	0x78h ~ 0x7Ah	0x7Bh
Sector 31	0x7Ch ~ 0x7Eh	0x7Fh

2K Bytes

Table 4: Mifare 4K Memory Map

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	0x80h ~ 0x8Eh	0x8Fh
Sector 33	0x90h ~ 0x9Eh	0x9Fh
..		
..		
Sector 38	0xE0h ~ 0xEEh	0xEFh
Sector 39	0xF0h ~ 0xFEh	0xFFh

2K Bytes

Examples:

// To authenticate the Block 0x04h with a {TYPE A, key number 0x00h}.

// PC/SC V2.01, Obsolete

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

<Similarly>

// To authenticate the Block 0x04h with a {TYPE A, key number 0x00h}.

// 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.*

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 5: Mifare Ultralight Memory Map

3.1.2.2.3. Read Binary Blocks

The ***Read Binary Blocks*** command is used for retrieving multiple "data blocks" from the PICC. The data block/trailer block must be authenticated first before executing the *Read Binary Blocks* command.

Read Binary Block 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): Multiply of 16 bytes for Mifare 1K/4K or Multiply of 4 bytes for Mifare Ultralight

- Maximum 16 bytes for Mifare Ultralight
- Maximum 48 bytes for Mifare 1K. (Multiple Blocks Mode; 3 consecutive blocks)
- Maximum 240 bytes for Mifare 4K. (Multiple Blocks Mode; 15 consecutive blocks)

Example 1: 0x10h (16 bytes). The starting block only. (Single Block Mode)

Example 2: 0x40h (64 bytes). From the starting block to starting block +3. (Multiple Blocks Mode)

Note: For safety reason, 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 (Multiply of 4/16 Bytes)	SW1	SW2

Read Binary Block Response Codes

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

Examples:

// Read 16 bytes from the binary block 0x04h (Mifare 1K or 4K)

APDU = {FF B0 00 04 10h}

// Read 240 bytes starting from the binary block 0x80h (Mifare 4K)

// Block 0x80h to Block 0x8Eh (15 blocks)

APDU = {FF B0 00 80 F0h}

3.1.2.2.4. Update Binary Blocks

The ***Update Binary Blocks*** command is used for writing multiple "data blocks" into the PICC. The data block/trailer block must be authenticated first before executing the *Update Binary Blocks* command.

Update Binary APDU Format (Multiple of 16 + 5 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Update Binary Blocks	FFh	D6h	00h	Block Number	Number of Bytes to Update	Block Data (Multiple of 16 Bytes)

Where:

Block Number (1 Byte): The starting block to be updated.

Number of Bytes to Update (1 Byte):

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

Example 1: 0x10h (16 bytes). The starting block only. (Single Block Mode)

Example 2: 0x30h (48 bytes). From the starting block to starting block+2. (Multiple Blocks Mode)

Note: *For safety reason, the Multiple Blocks 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.*

Block Data (Multiply of 16 + 2 Bytes, or 6 bytes): The data to be written into the binary block/blocks.

Update Binary Block Response Codes (2 Bytes)

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

Examples:

// Update the binary block 0x04h of Mifare 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 0x04 of Mifare Ultralight with Data {00 01 02 03h}

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

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

The ***Value Block*** command is used for manipulating value-based transactions.E.g., Increment a value of the value block, etc.

Value Block Operation APDU Format (10 Bytes)

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

Where:

Block Number (1 Byte): The value block to be manipulated.

VB_OP (1 Byte): 0x00h = Store the VB_Value into the block and will be converted to a value block.

0x01h = Increment the value of the value block by the VB_Value. This command is only valid for value block.

0x02h = Decrement the value of the value block by the VB_Value. This command is only valid for value block.

VB_Value (4 Bytes): The value used for value manipulation. The value is a signed long integer (4 bytes).

Example 1: Decimal 4 = {0xFFh, 0xFFh, 0xFFh, 0xFCh}

VB_Value			
MSB		LSB	
FFh	FFh	FFh	FCh

Example 2: Decimal 1 = {0x00h, 0x00h, 0x00h, 0x01h}

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

3.1.2.2.6. Read Value Block

The ***Read Value Block*** command is used for retrieving the value from the value block. This command is only valid for value block.

Read Value Block APDU Format (5 Bytes)

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

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 = {0xFFh, 0xFFh, 0xFFh, 0xFCh}

Value			
MSB			LSB
FFh	FFh	FFh	FCh

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

Example 2: Decimal 1 = {0x00h, 0x00h, 0x00h, 0x01h}

Value			
MSB			LSB
00h	00h	00h	01h

Read Value Block Response Codes

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

3.1.2.2.7. Copy Value Block

Copy Value Block command is used to copy a value from a value block to another value block.

Copy Value Block APDU Format (7 Bytes)

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

Where:

Source Block Number (1 Byte) = The value of the source value block will be copied to the target value block.

Target Block Number (1 Byte) = The value block to be restored. The source and target value blocks must be in the same sector.

Copy Value Block Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Copy Value Block Response Codes

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

Examples:

// Store a value "1" into block 0x05h

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

// Read the value block 0x05h

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

APDU = {FF B1 00 05 00h}

// Copy the value from value block 0x05h to value block 0x06h

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

// Increment the value block 0x05h by "5"

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

3.1.2.3. Access PC/SC Compliant Tags (ISO 14443-4)

All ISO 14443-4 compliant cards (PICCs) would understand the ISO 7816-4 APDUs. The ACR1281S reader has to communicate with the ISO 14443-4 compliant cards through exchanging ISO 7816-4 APDUs and responses. ACR1281S will handle the ISO 14443 Parts 1-4 Protocols internally.

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

ISO 7816-4 APDU Format

Command	Class	INS	P1	P2	Lc	Data In	Le
ISO 7816 Part 4 Command					Length of the Data In		Expected length of the Response Data

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

Response	Data Out		
Result	Response Data	SW1	SW2

Common ISO 7816-4 Response Codes

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

Typical sequence may be:

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

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

Step 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:

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

APDU ={80 B2 80 00 08h}

Class = 0x80h

INS = 0xB2h

P1 = 0x80h

P2 = 0x00h

Lc = None

Data In = None

Le = 0x08h

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

4.0. Peripherals Control

Accessing peripherals should be sent via *PC_to_RDR_Escape* with bSlot = 0

4.1.1. Get Firmware Version

Get Firmware Version command is used to get the reader's firmware message.

Get Firmware Version Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Get Firmware Version	0xE0h	0x00h	0x00h	0x18h	0x00h

Get Firmware Version Response Format (Firmware Message Length)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	Number of Bytes to be Received	Firmware Version

Sample Response = E1 00 00 00 0F 41 43 52 31 32 38 31 53 5F 56 33 30 33 2E 30h

Firmware Version (HEX) = 41 43 52 31 32 38 31 53 5F 56 33 30 33 2E 30h

Firmware Version (ASCII) = "ACR1281S_V303.0"

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

4.1.2. LED Control

LED Control command is used to control the LEDs' output.

LED Control Format (6 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
LED Control	0xE0h	0x00h	0x00h	0x29h	0x01h	LED Status

LED Control Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	LED Status

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

Table 6: LED Status (1 Byte) – LED Control

4.1.3. LED Status

LED Status command is used to check the existing LEDs' status.

LED Status Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
LED Status	0xE0h	0x00h	0x00h	0x29h	0x00h

LED Status Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	LED Status

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

Table 7: LED Status (1 Byte) – LED Status

4.1.4. Buzzer Control

Buzzer Control command is used to control the buzzer output.

Buzzer Control Format (6 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Buzzer Control	0xE0h	0x00h	0x00h	0x28h	0x01h	Buzzer On Duration

Buzzer On Duration (1Bytes): 0x00h = Turn OFF
0x01 to 0xFFh = Duration (unit: 10ms)

Buzzer Control Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	00

4.1.5. Set Default LED and Buzzer Behaviors

Set Default LED and Buzzer Behaviors command is used to configure the *Set the Default Behaviors for LEDs and Buzzer* card reader feature.

Set Default LED and Buzzer Behaviors Format (6 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set Default LED and Buzzer Behaviors	0xE0h	0x00h	0x00h	0x21h	0x01h	Default Behaviors

Default Behaviors	Mode	Description
Bit 0	ICC Activation Status LED	To show the activation status of the ICC interface. 1 = Enable; 0 =Disable
Bit 1	PICC Polling Status LED	To show the PICC Polling Status. 1 = Enable; 0 =Disable
Bit 2	PICC Activation Status LED	To show the activation status of the PICC interface 1 = Enable; 0 =Disable
Bit 3	RFU	RFU
Bit 4	Card Insertion and Removal Events Buzzer	To make a beep whenever a card insertion or removal event is detected. (For both ICC and PICC) 1 = Enable; 0 =Disabled
Bit 5	RC531 Reset Indication Buzzer	To make a beep when the RC531 is reset. 1 = Enable; 0 =Disabled
Bit 6	Exclusive Mode Status Buzzer. Either ICC or PICC interface can be activated.	To make a beep when the exclusive mode is activated. 1 = Enable; 0 =Disable
Bit 7	Card Operation Blinking LED	To make the LED blink whenever the card (PICC or ICC) is being accessed.

Table 8: Default Behaviors (1Byte)

Note: **Default value of Default Behaviors = 0xFBh.*

Set Default LED and Buzzer Behaviors Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Default Behaviors

4.1.6. Read Default LED and Buzzer Behaviors

Read Default LED and Buzzer Behaviors command is used to configure the *Read the current Default Behaviors for LEDs and Buzzer* card reader feature.

Read Default LED and Buzzer Behaviors Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read Default LED and Buzzer Behaviors	0xE0h	0x00h	0x00h	0x21h	0x00h

Read Default LED and Buzzer Behaviors Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Default Behaviors

Default Behaviors	Mode	Description
Bit 0	ICC Activation Status LED	To show the activation status of the ICC interface. 1 = Enable; 0 =Disable
Bit 1	PICC Polling Status LED	To show the PICC Polling Status. 1 = Enable; 0 =Disable
Bit 2	PICC Activation Status LED	To show the activation status of the PICC interface 1 = Enable; 0 =Disable
Bit 3	RFU	RFU
Bit 4	Card Insertion and Removal Events Buzzer	To make a beep whenever a card insertion or removal event is detected. (For both ICC and PICC) 1 = Enable; 0 =Disabled
Bit 5	RC531 Reset Indication Buzzer	To make a beep when the RC531 is reset. 1 = Enable; 0 =Disabled
Bit 6	Exclusive Mode Status Buzzer. Either ICC or PICC interface can be activated.	To make a beep when the exclusive mode is activated. 1 = Enable; 0 =Disable
Bit 7	Card Operation Blinking LED	To make the LED blink whenever the card (PICC or ICC) is being accessed.

Table 9: Default Behaviors (1Byte)

Note: *Default value of Default Behaviors = 0xFBh.*

4.1.7. Initialize Cards Insertion Counter

Initialize Cards Insertion Counter command is used to initialize the card's insertion/detection counter.

Initialize Cards Insertion Counter Format (9 Bytes)

Command	Class	INS	P1	P2	Lc	Data In			
Initialize Cards Insertion Counter	0xE0h	0x00h	0x00h	0x09h	0x04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

Initialize Cards Insertion Counter Response Format (9 Bytes)

Response	Class	INS	P1	P2	Lc	Data Out			
Result	0xE1h	0x00h	0x00h	0x00h	0x04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

Where:

ICC Cnt (LSB) (1 Byte) = ICC Insertion Counter (LSB)

ICC Cnt (MSB) (1 Byte) = ICC Insertion Counter (MSB)

PICC Cnt (LSB) (1 Byte) = PICC Insertion Counter (LSB)

PICC Cnt (MSB) (1 Byte) = PICC Insertion Counter (MSB)

4.1.8. Read Cards Insertion Counter

Read Cards Insertion Counter command is used to check the card's insertion/detection counter value.

Read Cards Insertion Counter Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read Cards Insertion Counter	0xE0h	0x00h	0x00h	0x09h	0x00h

Read Cards Insertion Counter Response Format (9 Bytes)

Response	Class	INS	P1	P2	Lc	Data Out			
Result	0xE1h	0x00h	0x00h	0x00h	0x04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

Where:

ICC Cnt (LSB) (1 Byte) = ICC Insertion Counter (LSB)

ICC Cnt (MSB) (1 Byte) = ICC Insertion Counter (MSB)

PICC Cnt (LSB) (1 Byte) = PICC Insertion Counter (LSB)

PICC Cnt (MSB) (1 Byte) = PICC Insertion Counter (MSB)

4.1.9. Update Cards Insertion Counter

Update Cards Insertion Counter command is used to update the card's insertion/detection counter value.

Update Cards Insertion Counter Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Update Cards Insertion Counter	0xE0h	0x00h	0x00h	0x0Ah	0x00h

Update Cards Insertion Counter Response Format (9 Bytes)

Response	Class	INS	P1	P2	Lc	Data Out			
Result	0xE1h	0x00h	0x00h	0x00h	0x04h	ICC Cnt (LSB)	ICC Cnt (MSB)	PICC Cnt (LSB)	PICC Cnt (MSB)

Where:

ICC Cnt (LSB) (1 Byte) = ICC Insertion Counter (LSB)

ICC Cnt (MSB) (1 Byte) = ICC Insertion Counter (MSB)

PICC Cnt (LSB) (1 Byte) = PICC Insertion Counter (LSB)

PICC Cnt (MSB) (1 Byte) = PICC Insertion Counter (MSB)

4.1.10. Set Automatic PICC Polling

Set Automatic PICC Polling command is used to set the reader's polling mode.

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

We can send a command to disable the PICC polling function. The command is sent through the *PCSC Escape* command interface.

Note: *To meet the energy saving requirement, special modes are provided for turning off the antenna field whenever the PICC is inactive, or no PICC is found. The reader will consume less current in power saving mode.*

Set Automatic PICC Polling Format (6 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set Automatic PICC Polling	0xE0h	0x00h	0x00h	0x23h	0x01h	Polling Setting

Set Automatic PICC Polling Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Polling Setting

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

Table 10: Polling Setting (1Byte)

Note: *Default value of Polling Setting = 0x8Fh.*

Notes:

1. *It is recommended to enable the option **"Turn Off Antenna Field if the PICC is inactive"**, so that the **"Inactive PICC"** will not be exposed to the field all the time so as to prevent the PICC*

from "warming up".

2. *The longer the PICC Poll Interval, the more efficient of energy saving. However, the response time of PICC Polling will become longer. The Idle Current Consumption in Power Saving Mode is about 60mA, while the Idle Current Consumption in Non-Power Saving mode is about 130mA. Idle Current Consumption = PICC is not activated.*
3. *The reader will activate the ISO 14443A-4 mode of the "ISO 14443A-4 compliant PICC" automatically. Type B PICC will not be affected by this option.*
4. *The JCOP30 card comes with two modes: ISO 14443A-3 (Mifare 1K) and ISO 14443A-4 modes. The application has to decide which mode should be selected once the PICC is activated.*

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

4.1.11. Read Automatic PICC Polling

Read the Automatic PICC Polling command is used to check the current *Automatic PICC Polling Setting*.

Read Automatic PICC Polling Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read Automatic PICC Polling	0xE0h	0x00h	0x00h	0x23h	0x00h

Read the Configure mode Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Polling Setting

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

Table 11: Polling Setting (1Bytes)

Note: *Default value of Polling Setting = 0x8Fh.*

4.1.12. Set the PICC Operating Parameter

Set the PICC Operating Parameter command is used to configure the PICC Operating Parameter.

Set the PICC Operating Parameter Format (6 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set the PICC Operating Parameter	0xE0h	0x00h	0x00h	0x20h	0x01h	Operation Parameter

Set the PICC Operating Parameter Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Operation Parameter

Operating Parameter	Parameter	Description	Option
Bit0	ISO 14443 Type A	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit1	ISO 14443 Type B		1 = Detect 0 = Skip
Bit2 - 7	RFU	RFU	RFU

Table 12: Operating Parameter (1 Byte)

Note: *Default value of Operation Parameter = 0x03h.*

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

4.1.13. Read the PICC Operating Parameter

Read the PICC Operating Parameter command is used to check current PICC Operating Parameter.

Read the PICC Operating Parameter Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read the PICC Operating Parameter	0xE0h	0x00h	0x00h	0x20h	0x00h

Read the PICC Operating Parameter Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Operation Parameter

Operating Parameter	Parameter	Description	Option
Bit0	ISO 14443 Type A	The Tag Types to be detected during PICC Polling.	1 = Detect 0 = Skip
Bit1	ISO 14443 Type B		1 = Detect 0 = Skip
Bit2 - 7	RFU	RFU	RFU

Table 13: Operating Parameter (1 Byte)

4.1.14. Set the Exclusive Mode

Set the Exclusive Mode command is used to set the reader into/out from Exclusive Mode.

Set the Exclusive Mode Format (6 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set the Exclusive Mode	0xE0h	0x00h	0x00h	0x2Bh	0x01h	Exclusive mode

Set the Exclusive Mode Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Exclusive mode

Where:

Exclusive Mode (1Bytes): 0x00h = Share Mode, ICC and PICC Interface work together

0x01h = Exclusive Mode, PICC disable Auto Poll and Antenna power off, when ICC inserted (Default)

4.1.15. Read the Exclusive Mode

Read the Exclusive Mode command is used to check current Exclusive Mode setting.

Read the Exclusive Mode Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read the Exclusive Mode	0xE0h	0x00h	0x00h	0x2Bh	0x00h

Set the Exclusive Mode Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Exclusive Mode

Where:

Exclusive mode (1Bytes): 0x00 = Share Mode, ICC and PICC Interface work together

0x01 = Exclusive Mode, PICC disable Auto Poll and Antenna power off, when ICC inserted (Default)

4.1.16. Set Auto PPS

Whenever a PICC is recognized, the reader will try to change the communication speed between the PCD and PICC 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.

Set Auto PPS Format (7 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set Auto PPS	0xE0h	0x00h	0x00h	0x24h	0x01h	Max Speed

Set Auto PPS Response Format (9 Bytes)

Response	Class	INS	P1	P2	Le	Data Out	
Result	0xE1h	0x00h	0x00h	0x00h	0x02h	Max Speed	Current Speed

Where:

Max Speed (1 Byte) = Maximum Speed

Current Speed (1 Byte) = Current Speed

Value can be: 106k bps = 0x00h -> equal to No Auto PPS (default setting)

212k bps = 0x01h

424k bps = 0x02h

848k bps = 0x03h

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. *The reader supports different speed between sending and receiving.*

4.1.17. Read Auto PPS

Read Auto PPS command is used to check current *Auto PPS Setting*.

Read Auto PPS Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read Auto PPS	0xE0h	0x00h	0x00h	0x24h	0x00h

*Set Auto P*PS Response Format (9 Bytes)

Response	Class	INS	P1	P2	Le	Data Out	
Result	0xE1h	0x00h	0x00h	0x00h	0x02h	Max Speed	Current Speed

Where:

Max Speed (1 Byte) = Maximum Speed

Current Speed (1 Byte) = Current Speed

Value can be: 106k bps = 0x00h -> equal to No Auto PPS (default setting)

212k bps = 0x01h

424k bps = 0x02h

848k bps = 0x03h

4.1.18. Antenna Field Control

Antennal Field Control command is used for turning on/off the antenna field.

Antenna Field Control Format (6 Bytes)

Command	Class	INS	P1	P2	Lc	Data In
Antenna Field Control	0xE0h	0x00h	0x00h	0x25h	0x01h	Status

Antenna Field Control Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Status

Where:

Status (1 Byte): 0x01h = Enable Antenna Field

0x00h = Disable Antenna Field

Note: *Make sure the Auto PICC Polling is disabled before turning off the antenna field.*

4.1.19. Read Antenna Field Status

Read Antenna Field Status command is used to check current *Antenna Field Status*.

Read Antenna Field Status Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read Antenna Field Status	0xE0h	0x00h	0x00h	0x25h	0x00h

Read Antenna Field Status Response Format (6 Bytes)

Response	Class	INS	P1	P2	Le	Data Out
Result	0xE1h	0x00h	0x00h	0x00h	0x01h	Status

Where:

Status (1 Byte): 0x01h = Enable Antenna Field

0x00h = Disable Antenna Field

4.1.20. User Extra Guard Time Setting

User Extra Guard Time Setting is used to set the extra guard time for ICC and SAM communication.

Note: *The user extra guard time value will be stored into EEPROM.*

User Extra Guard Time Setting Format (7 Bytes)

Command	Class	INS	P1	P2	Lc	Data In	
User Extra Guard Time Setting	0xE0h	0x00h	0x00h	0x2Eh	0x02h	ICC UserGuardTime	SAM UserGuardTime

User Extra Guard Time Setting Response Format (7 Bytes)

Response	Class	INS	P1	P2	Le	Data Out	
Result	0xE1h	0x00h	0x00h	0x00h	0x02h	ICC UserGuardTime	SAM UserGuardTime

Where:

ICC UserGuardTime (1 Byte) = User Guard Time value for ICC Slot

SAM UserGuardTime (1 Byte) = User Guard Time value for SAM Slot

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

4.1.21. Read User Extra Guard Time

Read User Extra Guard Time command is used to read the set extra guard time for ICC and SAM communication.

Read User Extra Guard Time Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read User Extra Guard Time	0xE0h	0x00h	0x00h	0x2Eh	0x00h

Read User Extra Guard Time Response Format (7 Bytes)

Response	Class	INS	P1	P2	Le	Data Out	
Result	0xE1h	0x00h	0x00h	0x00h	0x02h	ICC UserGuardTime	SAM UserGuardTime

Where:

ICC UserGuardTime (1 Byte) = User Guard Time value for ICC Slot

SAM UserGuardTime (1 Byte) = User Guard Time value for SAM Slot

4.1.22. "616C" Auto Handle Option Setting

The ***"616C" Auto Handle Option Setting*** command is used to configure the *"616C" Auto Handle Option*.

* Optional for T=0 ACOS5

"616C" Auto Handle Option Setting Format (7 Bytes)

Command	Class	INS	P1	P2	Lc	Data In	
"616C" Auto Handle Option Setting	0xE0h	0x00h	0x00h	0x32h	0x02h	ICC Option	SAM Option

"616C" Auto Handle Option Setting Response Format (7 Bytes)

Response	Class	INS	P1	P2	Le	Data Out	
Result	0xE1h	0x00h	0x00h	0x00h	0x02h	ICC Option	SAM Option

Where:

ICC Option (1 Byte): User Guard Time value for ICC Slot
0xFFh = Enable "616C" Auto Handle
0x00h = Disable "616C" Auto Handle (Default)

SAM Option (1 Byte): User Guard Time value for SAM Slot
0xFFh = Enable "616C" Auto Handle
0x00h = Disable "616C" Auto Handle (Default)

4.1.23. Read "616C" Auto Handle Option

Read "616C" Auto Handle Option command is used to read the *"616C" Auto Handle Option*.

Read "616C" Auto Handle Option Format (5 Bytes)

Command	Class	INS	P1	P2	Lc
Read "616C" Auto Handle Option	0xE0h	0x00h	0x00h	0x32h	0x00h

Read "616C" Auto Handle Option Response Format (7 Bytes)

Response	Class	INS	P1	P2	Le	Data Out	
Result	0xE1h	0x00h	0x00h	0x00h	0x02h	ICC Option	SAM Option

Where:

ICC Option (1 Byte): User Guard Time value for ICC Slot
0xFFh = Enable "616C" Auto Handle
0x00h = Disable "616C" Auto Handle (Default)

SAM Option (1 Byte): User Guard Time value for SAM Slot
0xFFh = Enable "616C" Auto Handle
0x00h = Disable "616C" Auto Handle (Default)

4.1.24. Set Serial Communication Mode

Set Serial Communication Mode command is used to configure the communication speed and communication mode.

Set Serial Communication Mode Format (2 Bytes)

Command	Byte 0	Byte 1
Set Serial Communication Mode	0x44h	Mode Select

Set Serial Communication Mode Response Format (2 Bytes)

Response	Byte 0	Byte 1
Result	0x90h	Mode Select

Offset	Parameter	Description
Bit 0-3	Serial Communication Speed	000b= 9600bps(Default) 001b= 19200bps 010b= 38400bps 011b= 57600bps 100b= 115200bps 101b= 128000bps 110b= 230400bps Other value reserve for future use.
Bit 4 - 6	RFU	RFU
Bit 7	Interrupt-In Message(CCID-like Format)	1 = Report Interrupt-In Message. 0 = Not report (Default).

Table 14: Mode Select (1 Byte) – Communication Speed and Mode Selection

Note: *After the communication speed is changed successfully, the program has to adjust its communication speed to continue the rest of the data exchanges.*

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

Appendix A. Supported Card Types

The following table summarizes the card type returned by *GET_READER_INFORMATION* correspond with the respective card type.

Card Type Code	Card Type
00h	Auto-select T=0 or T=1 communication protocol
01h	I2C memory card (1k, 2k, 4k, 8k and 16k bits)
02h	I2C memory card (32k, 64k, 128k, 256k, 512k and 1024k bits)
03h	Atmel AT88SC153 secure memory card
04h	Atmel AT88SC1608 secure memory card
05h	Infineon SLE4418 and SLE4428
06h	Infineon SLE4432 and SLE4442
07h	Infineon SLE4406, SLE4436 and SLE5536
08h	Infineon SLE4404
09h	Atmel AT88SC101, AT88SC102 and AT88SC1003

Table 15: Supported Card Types