Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1281U-C1 Dual Interface Reader

Application Programming Interface

Table of Contents

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

Figures

Tables

1.0. Introduction

ACR1281U-C1 DualBoost II is the second generation of ACS's ACR128 DualBoost Reader. ACR1281U-C1 is a powerful and efficient dual interface smart card reader, which can be used to access ISO 7816 MCU cards and Mifare, ISO 14443 Type A and B Contactless Cards. It makes use of the USB CCID class driver and USB interface to connect to a PC and accept card commands from the computer application.

ACR1281U-C1 acts as the intermediary device between the PC and the card. The reader, specifically to communicate with a contactless tag, MCU card, SAM card, or the device peripherals (LED or buzzer), will carry out a command issued from the PC. It has three interfaces namely the PICC, ICC and SAM interfaces, and all these interfaces 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, refer to the related card documentation and the PC/SC specifications. This API document will discuss in detail how the PC/SC APDU commands were implemented for the contactless interface, contact memory card support and device peripherals of ACR1281U-C1.

1.1. Features

The ACR1281U-C1 Dual Interface Smart Card Reader has the following features:

- A built-in antenna is provided for PICC applications.
- A standard ICC landing type card acceptor.
- A SAM socket is provided for highly secure applications.
- It is ISO 14443 Parts 1-4 compliant for Contactless Smart Card Interface.
- Its contactless interface supports ISO 14443 Part 4 Type A & B and Mifare Classics.
- It uses the T=CL emulation for Mifare 1K/4K PICCs.
- It supports Extended APDU with a maximum of 64 kbytes.
- It is ISO 7816 Parts 1-4 compliant for Contact Smart Card Interface and supports memory cards.
- It has User-Controllable Peripherals such as LED and Buzzer.
- It has energy saving modes to turn off the antenna field whenever the PICC is inactive, or no PICC is found to prevent the PICC from being exposed to the field all the time.
- The device is PC/SC compliant for three interfaces namely Contact, Contactless, and SAM Interface.
- The device makes use of the Microsoft CCID class driver for both Contactless and Contact interfaces.
- It makes use of USB V2.0 Interface (12 Mbps).
- It is firmware upgradeable through the USB Interface.

2.0. Architecture of ACR1281U

2.1. Reader Block Diagram

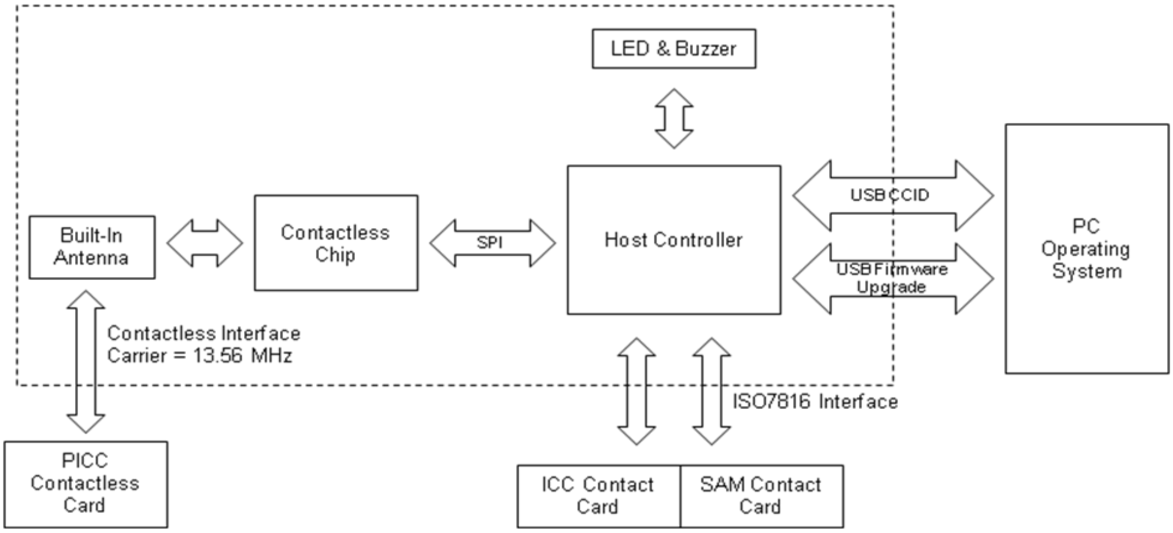


Figure 1: ACR1281U Reader Block Diagram

2.2. Communication between the PC/SC Driver and the ICC, PICC & SAM

The protocol between ACR1281U-C1 and the PC is using CCID protocol. All the communication between ICC, PICC and SAM are PC/SC compliant.

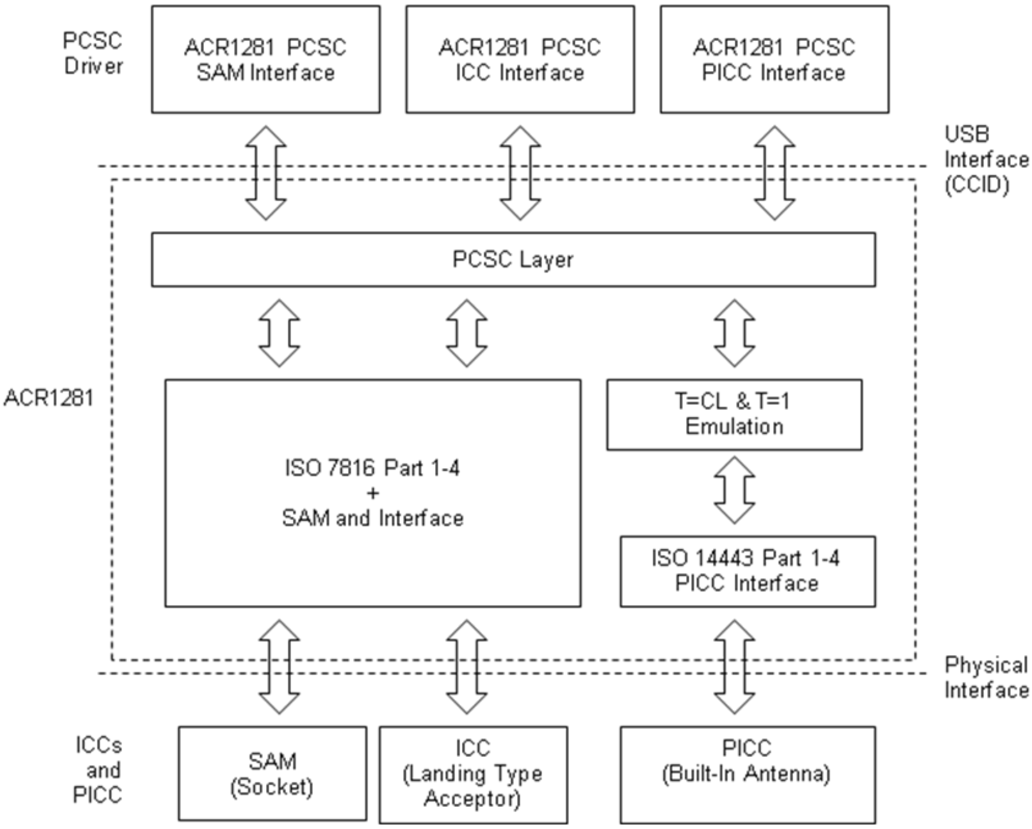


Figure 2: ACR1281U Architecture

3.0. Hardware Description

3.1. USB

The ACR1281U-C1 connects to a computer through a USB following the USB standard.

3.1.1. Communication Parameters

The ACR1281U-C1 connects to a computer through USB as specified in the USB Specification 2.0. The ACR1281U-C1 is working 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 ACR1281U-C1 and PC.
3	D+	Differential signal transmits data between ACR1281U-C1 and PC.
4	GND	Reference voltage level for power supply

Table 1: USB Interface Wiring

Note: In order for the ACR1281U-C1 to function properly through USB interface, the device driver should be installed.

3.1.2. Endpoints

The ACR1281U-C1 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 host to ACR1281U-C1 (data packet size is 64 bytes)

Bulk IN – For response to be sent from ACR1281U-C1 to host (data packet size is 64 bytes)

Interrupt IN – For card status message to be sent from ACR1281U-C1 to host (data packet size is 8 bytes)

3.2. Contact Smart Card Interface

The interface between the ACR1281U-C1 and the inserted smart card follows the specifications of ISO 7816-3 with certain restrictions or enhancements to increase the practical functionality of the ACR1281U-C1.

3.2.1. Smart Card Power Supply VCC (C1)

The current consumption of the inserted card must not be any higher than 50 mA.

3.2.2. 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 ACR1281U-C1. This includes both memory card and MCU-based cards.

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

3.2.3. 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 4.8 MHz is applied to the CLK signal (C3).

3.3. Contactless Smart Card Interface

The interface between the ACR1281U-C1 and the Contactless Card follows the specifications of ISO 14443 with certain restrictions or enhancements to increase the practical functionality of the ACR1281U-C1.

3.3.1. Carrier Frequency

The carrier frequency for ACR1281U is 13.56 MHz.

3.3.2. Card Polling

The ACR1281U-C1 automatically polls the contactless cards that are within the field. ISO 14443-4 Type A, ISO 14443-4 Type B and Mifare are supported.

3.4. User Interface

3.4.1. Buzzer

A monotone buzzer is used to show the “Card Insertion” and “Card Removal” events.

Events	Buzzer
1. The reader powered up and initialization success.	Beep
2. Card Insertion Event (ICC or PICC)	Beep
3. Card Removal Event (ICC or PICC)	Beep

Table 2: Buzzer Event

3.4.2. LED

The LEDs are used for showing the state of the contact and contactless interfaces. The Red LED is used for showing PICC status and Green LED for ICC.

Reader States	Red LED PICC Indicator	Green LED ICC Indicator
1. No PICC Found or PICC present but not activated.	A single pulse per ~ 5 seconds	
2. PICC is present and activated	ON	
3. PICC is operating	Blinking	
4. ICC is present and activated		ON
5. ICC is absent or not activated		OFF
6. ICC is operating		Blinking

Table 3: LED Indicator

4.0. Software Design

4.1. Contact Smart Card Protocol

4.1.1. Memory Card – 1/2/4/8/16 kbits I2C Card

4.1.1.1. Select Card Type

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

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FF_H	$A4_H$	00_H	00_H	01_H	01_H

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = 90 00_H if the operation is completed successfully

4.1.1.2. Select Page Size

This command will choose the page size to read the smart card. The default value is 8-byte page write. It will reset to default value whenever the card is removed of the reader is powered off.

Command

Command	Class	INS	P1	P2	Lc	Page Size
Select Page Size	FF_H	01_H	00_H	00_H	01_H	

Where:

Page Size: 1 Byte.

03_H = 8-byte page write

04_H = 16-byte page write

05_H = 32-byte page write

06_H = 64-byte page write

07_H = 128-byte page write

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = `90 00`$_H$ if the operation is completed successfully

4.1.1.3. Read Memory Card

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

Command

Command	Class	INS	Byte Address		MEM_L
			MSB	LSB	
Read Memory Card	FF$_H$	B0$_H$			

Where:

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

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

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

SW1, SW2 = `90 00`$_H$ if the operation is completed successfully

4.1.1.4. Write Memory Card

This command will write 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	FF$_H$	D0$_H$							

Where:

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.

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = `90 00`$_H$ if the operation is completed successfully

4.1.2. Memory Card – 32/64/128/256/512/1024 kbits I2C Card

4.1.2.1. Select Card Type

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

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FF$_H$	A4$_H$	00$_H$	00$_H$	01$_H$	02$_H$

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = `90 00`$_H$ if the operation is completed successfully

4.1.2.2. Select Page Size

This command will choose the page size to read the smart card. The default value is 8-byte page write. It will reset to default value whenever the card is removed of the reader is powered off.

Command

Command	Class	INS	P1	P2	Lc	Page Size
Select Page Size	FF$_H$	01$_H$	00$_H$	00$_H$	01$_H$	

Where:

Page Size: 1 byte.

03$_H$ = 8-byte page write

04$_H$ = 16-byte page write

05$_H$ = 32-byte page write

06$_H$ = 64-byte page write

07$_H$ = 128-byte page write

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = `90 00`$_H$ if the operation is completed successfully

4.1.2.3. Read Memory Card

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

Command

Command	Class	INS	Byte Address		MEM_L
			MSB	LSB	
Read Memory Card	FF$_H$				

Where:

INS: 1 Byte

`B0`$_H$ = For 32, 64, 128, 256, 512 kbit I2C card

`1011 000*`$_b$; where * is the MSB of the 17 bit addressing = For 1024 kbit I2C card

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

Response	Byte 1	…	…	Byte N	SW1	SW2
Result						

Where:

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

SW1, SW2 = `90 00`$_H$ if the operation is completed successfully

4.1.2.4. Write Memory Card

This command will write 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	FF$_H$								

Where:

INS: 1 Byte.

$D0_H$ = For 32, 64, 128, 256, 512 kbit I2C card

$1101\ 000*_b$; where * is the MSB of the 17 bit addressing = For 1024 kbit I2C card

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.

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = $90\ 00_H$ if the operation is completed successfully

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

4.1.3.1. Select Card Type

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

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FF_H	$A4_H$	00_H	00_H	01_H	05_H

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = $90\ 00_H$ if the operation is completed successfully

4.1.3.2. Read Memory Card

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

Command

Command	Class	INS	Byte Address		MEM_L
			MSB	LSB	
Read Memory Card	FF_H	$B0_H$			

Where:

MSB Byte Address: 1 byte.

= $0000\ 00\ A9\ A8_b$ is the memory address location of the memory card

LSB Byte Address: 1 byte.

= $A7\ A6\ A5\ A4\ A3\ A2\ A1\ A0_b$ 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 = $90\ 00_H$ if the operation is completed successfully

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

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

Command

Command	Class	INS	P1	P2	MEM_L
Presentation Error Counter	FF_H	$B1_H$	00_H	00_H	03_H

Response

Response	ErrCnt	Dummy1	Dummy2	SW1	SW2
Result					

Where:

ErrCnt: 1 byte. The value of the presentation error counter.

FF_H = indicates the verification is correct

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

Other values indicate the verification failed

Dummy1, Dummy 2: 2 Bytes. Dummy data read from the card

SW1, SW2 = `90 00`$_H$ if the operation is completed successfully

4.1.3.4. Read Protection Bit

This command is used to read the protection bit.

Command

Command	Class	INS	Byte Address		MEM_L
			MSB	LSB	
Read Protection Bit	FF$_H$	B2$_H$			

Where:

MSB Byte Address: 1 Byte. The memory address location of the memory card.

= `0000 00 A9 A8`$_b$

LSB Byte Address: 1 Byte. The memory address location of the memory card.

= `A7 A6 A5 A4 A3 A2 A1 A0`$_b$

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 `0x0010`, the following pseudo-APDU should be issued:

`0xFF 0xB1 0x00 0x10 0x01`

Response

Response	PROT 1	…	…	PROT L	SW1	SW2
Result						

Where:

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

SW1, SW2 = `90 00`$_H$ if the operation is 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

4.1.3.5. Write Memory Card

This command will write 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	FF_H	$D0_H$							

Where:

MSB Byte Address: 1 Byte.

= $0000\ 00\ A9\ A8_b$ is the memory address location of the memory card

LSB Byte Address: 1 Byte.

= $A7\ A6\ A5\ A4\ A3\ A2\ A1\ A0_b$ 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 = $90\ 00_H$ if the operation is completed successfully

4.1.3.6. Write Protection Memory Card

Each of the bytes specified in the command is compared with the bytes stored in the specific address and if the data match, 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	FF_H	$D1_H$							

Where:

MSB Byte Address: 1 Byte.

= $0000\ 00\ A9\ A8_b$ is the memory address location of the memory card

LSB Byte Address: 1 Byte.

= $A7\ A6\ A5\ A4\ A3\ A2\ A1\ A0_b$ 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 00_H if the operation is completed successfully

4.1.3.7. Present Code Memory Card (for SLE44428 and SLE5528 only)

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:

- Search a ‘1’ bit in the presentation error counter and write the bit ‘0’
- Present the specified code to the card
- Try to erase the presentation error counter

Command

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

Where:

Code: 2 Byte. Secret code (PIN).

Response

Response	Data Out	
Result	90_H	ErrorCnt

Where:

ErrorCnt: 1 Byte. Error Counter.

FF_H = indicates the verification is correct.

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

Other values indicate the verification failed.

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

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

4.1.4.1. Select Card Type

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

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FF_H	$A4_H$	00_H	00_H	01_H	06_H

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = $90\ 00_H$ if the operation is completed successfully

4.1.4.2. Read Memory Card

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

Command

Command	Class	INS	P1	Byte Address	MEM_L
Read Memory Card	FF_H	$B0_H$	00_H		

Where:

Byte Address: 1 Byte.

= A7 A6 A5 A4 A3 A2 A1 $A0_b$ 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	PROT 1	PROT 2	PROT 3	PROT 4	SW1	SW2
Result										

Where:

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

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

SW1, SW2 = $90\ 00_H$ if the operation is 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

4.1.4.3. Presentation Error Counter Memory Card (for SLE4442 and SLE5542 only)

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

Command

Command	Class	INS	P1	P2	MEM_L
Presentation Error Counter	FF_H	$B1_H$	00_H	00_H	04_H

Response

Response	ErrCnt	Dummy1	Dummy2	Dummy 3	SW1	SW2
Result						

Where:

ErrCnt: 1 Byte. The value of the presentation error counter.

07_H = indicates the verification is correct

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

Other values indicate the verification failed

Dummy1, Dummy 2, Dummy3: 3 Bytes. Dummy data read from the card

SW1, SW2 = $90\ 00_H$ if the operation is completed successfully

4.1.4.4. Read Protection Bit

This command is used to read the protection bits for the first 32 bytes.

Command

Command	Class	INS	P1	P2	MEM_L
Read Protection Bit	FF_H	$B2_H$	00_H	00_H	04_H

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

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 00`$_H$ if the operation is 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 bytes in the response data:

0 = byte is write protected

1 = byte can be written

4.1.4.5. Write Memory Card

This command will write 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	`FF`$_H$	`D0`$_H$	`00`$_H$						

Where:

Byte Address: 1 Byte.

= `A7 A6 A5 A4 A3 A2 A1 A0`$_b$ 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 00`$_H$ if the operation is completed successfully

4.1.4.6. Write Protection Memory Card

Each of the bytes specified in the command is compared with the bytes stored in the specific address and if the data match, 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	FF_H	$D1_H$	00_H						

Where:

Byte Address: 1 Byte.

= 000A4 A3 A2 A1$_b$ (00_H to $1F_H$) 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 00_H if the operation is completed successfully

4.1.4.7. Present Code Memory Card (for SLE4442 and SLE5542 only)

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:

- Search a '1' bit in the presentation error counter and write bit '0'
- Present the specified code to the card
- 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	FF_H	20_H	00_H	00_H	03_H			

Where:

Code: 3 Bytes. Secret Code (PIN).

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

Response

Response	Data Out	
Result	SW1	ErrorCnt

Where:

ErrorCnt: 1 Byte. Error Counter.

07_H = indicates the verification is correct.

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

Other values indicate the verification failed.

4.1.4.8. Change Code Memory Card (for SLE4442 and SLE5542 only)

This command is used to write 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	FF_H	$D2_H$	00_H	01_H	03_H			

Where:

Code: 3 Bytes. Secret Code (PIN).

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = $90\ 00_H$ if the operation is completed successfully

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

4.1.5.1. Select Card Type

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

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FF_H	$A4_H$	00_H	00_H	01_H	07_H

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = `90 00`H if the operation is completed successfully

4.1.5.2. Read Memory Card

This command will read 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 00`H if the operation is completed successfully

4.1.5.3. 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	FF_H	$D0_H$	00_H		02_H		

Where:

Byte Address: 1 Byte. Memory address location of the memory card

Mode: 1 Byte. Specifies the write mode and backup option

00_H = write

01_H = write with carry

02_H = write with backup enabled (for SLE4436, SLE5536 and SLE6636 only)

03_H = 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 = $90\ 00_H$ if the operation is completed successfully

4.1.5.4. Present Code Memory Card

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

- Search a ‘1’ bit in the presentation error counter and write bit ‘0’
- 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	FF_H	20_H	00_H	00_H	04_H	09_H			

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 = $90\ 00_H$ if the operation is completed successfully

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

This command is used to read the authentication certificate from the card. The following actions are executed:

- Select Key 1 or Key 2 in the card as specified in the command
- Present the challenge data specified in the command to the card
- Generate the specified number of CLK pulses for each bit authentication data computed by the card
- Read 16 bits of authentication data from the card
- 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	FF_H	84_H	00_H	00_H	08_H					

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

Where:

Key: 1 Byte. Key to be used for the computation of the authentication certificate.

00_H = key1 with no cipher block chaining

01_H = key2 with no cipher block chaining

80_H = key1 with cipher block chaining (for SLL5536 and SLE6636 only)

81_H = key2 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 ($A0_H$).

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

Response

Response	SW1	SW2
Result	61_H	02_H

Step 2: Get the Authentication Data (Get_Response)

Command

Command	Class	INS	P1	P2	MEM_L
Get Authentication Data	FF_H	$C0_H$	00_H	00_H	02_H

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 = $90\ 00_H$ if the operation is completed successfully

4.1.6. Memory Card – SLE4404

4.1.6.1. Select Card Type

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

Command

Command	Class	INS	P1	P2	Lc	Card Type
Select Card Type	FF_H	$A4_H$	00_H	00_H	01_H	08_H

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = $90\ 00_H$ if the operation is completed successfully

4.1.6.2. Read Memory Card

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

Command

Command	Class	INS	P1	Byte Address	MEM_L
Read Memory Card	FF_H	$B0_H$	00_H		

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 = $90\ 00_H$ if the operation is completed successfully

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

4.1.6.3. Write Memory Card

This command will write 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	FF_H	$D0_H$	00_H						

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 = $90\ 00_H$ if the operation is completed successfully

4.1.6.4. 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 the state of '1'.

Command

Command	Class	INS	P1	Byte Address	MEM_L
Erase Scratch Pad	FF_H	$D2_H$	00_H		00_H

Where:

Byte Address: 1 Byte. Memory byte address location of the scratch pad. (Typical value is 02_H)

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = 90 00_H if the operation is completed successfully

4.1.6.5. Verify User Code

This command is used to submit 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:

- Present the specified code to the card
- Search a '1' bit in the presentation error counter and write the bit '0'
- 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	FF_H	20_H	04_H	08_H	02_H		

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 00`$_H$ if the operation is completed successfully

= `63 00`$_H$ if there are no more retries left

Note: After SW1 SW2 = `90 00`$_H$ 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 equals to 'FF$_H$', the previous verification is successful.

4.1.6.6. Verify Memory Code

This command is used to submit 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:

- Present the specified code to the card
- Search a '1' bit in the presentation error counter and write the bit to '0'
- Erase the presentation error counter. Note that 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	FF$_H$	20$_H$	40$_H$	28$_H$	04$_H$				

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 00`$_H$ if the operation is completed successfully

= `63 00`$_H$ if there are no more retries left

Note: After SW1 SW2 = `90 00`$_H$ 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 equals to 'FF$_H$', the previous verification is successful.

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

4.2. Contactless Smart Card Protocol

4.2.1. ATR Generation

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

4.2.2. ATR format for ISO 14443 Part 3 PICCs.

Byte	Value (Hex)	Designation	Description
0	3B$_H$	Initial Header	
1	8N$_H$	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	80$_H$	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0
3	01$_H$	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1
4 To 3+N	80$_H$	T1	Category indicator byte, 80 means A status indicator may be present in an optional COMPACT-TLV data object
	4F$_H$	Tk	Application identifier Presence Indicator
	0C$_H$		Length
	RID		Registered Application Provider Identifier (RID) # A0 00 00 03 06
	SS		Byte for standard
	C0.. C1		Bytes for card name
	00 00 00 00$_H$	RFU	RFU # 00 00 00 00
4+N	UU	TCK	Exclusive-oring of all the bytes T0 to Tk

Table 4: ISO 14443 Part 3 ATR Format

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

Example:

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

ATR											
Initial Header	T0	TD1	TD2	T1	Tk	Length	RID	Standard	Card Name	RFU	TCK
$3B_H$	$8F_H$	80_H	01_H	80_H	$4F_H$	$0C_H$	A0 00 00 03 06_H	03_H	00_H 01_H	00 00 00 00_H	$6A_H$

Where:

Length (YY) = 0C
RID = A0 00 00 03 06 (PC/SC Workgroup)
Standard (SS) = 03 (ISO 14443A, Part 3)
Card Name (C0 ... C1) = [00 01] (Mifare 1K)
[00 02] (Mifare 4K)
[00 03] (Mifare Ultralight)
[00 26] (Mifare Mini)
[FF 28] JCOP 30
FF SAK undefined tags

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

4.2.3. 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>3B</td><td>Initial Header</td><td></td></tr>
<tr><td>1</td><td>8N</td><td>T0</td><td>Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following.
Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)</td></tr>
<tr><td>2</td><td>80</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>01</td><td>TD2</td><td>Higher nibble 0 means no TA3, TB3, TC3, TD3 following.
Lower nibble 1 means T = 1</td></tr>
<tr><td rowspan="2">4
to
3 + N</td><td>XX</td><td>T1</td><td rowspan="2">Historical Bytes:
ISO 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>XX
XX
XX</td><td>Tk</td></tr>
<tr><td>4+N</td><td>UU</td><td>TCK</td><td>Exclusive-oring of all the bytes T0 to Tk</td></tr>
</table>

Table 5: ISO 14443 Part 4 ATR Format

Example 1: Consider the ATR from DESFire as follows:

DESFire (ATR) = `3B 81 80 01 80 80` (6 bytes of ATR)

> **Note:** Use the APDU “`FF CA 01 00 00`” to distinguish the ISO 14443A-4 and ISO 14443B-4 PICCs and retrieve the full ATS if available. The ATS is returned for ISO 14443A-3 or ISO 14443B-3/4 PICCs.
>
> APDU Command = `FF CA 01 00 00`
>
> APDU Response = `06 75 77 81 02 90 00`
>
> ATS = `{06 75 77 81 02 80}`

Example 2: Consider the ATR from EZLink as follows:

EZLink (ATR) = `3B 88 80 01 1C 2D 94 11 F7 71 85 00 BE`

> Application Data of ATQB = `1C 2D 94 11`
>
> Protocol Information of ATQB = `F7 71 85`
>
> MBLI of ATTRIB = `00`

4.2.4. Pseudo APDUs for Contactless Interface

4.2.4.1. Get Data

This command is used to return the serial number or ATS of the "connected PICC".

Command

Command	Class	INS	P1	P2	Le
Get Data	FF_H	CA_H	00_H 01_H	00_H	00_H (Full Length)

Get UID Response if P1 = 00_H

Response	UID	…	…	UID	SW1	SW2
Result	LSB			MSB		

Get ATS Response if P1 = 01_H (for ISO 14443A cards only)

Response	Data Out		
Result	ATS	SW1	SW2

Response Code

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

Example 1: To get the serial number of the connected PICC

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

Example 2: To get the ATS of the connected ISO 14443-A PICC

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

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

4.2.4.3. Load Authentication Keys

This command is used to load the authentication keys into the reader. The authentication keys are used to authenticate the specified sector of the Mifare 1K/4K Memory Card. Two kinds of authentication key locations are provided, volatile and non-volatile key locations.

Command

Command	Class	INS	P1	P2	Le	Data In
Load Authentication Keys	FF_H	82_H	Key Structure	Key Number	06_H	Key

Where:

Key Structure: 1 Byte.

00_H = Key is loaded into the reader's volatile memory

20_H = Key is loaded into the reader's non-volatile memory

Other = Reserved.

Key Number: 1 Byte.

00_H – $1F_H$ = Non-volatile memory for storing keys. The keys are permanently stored in the reader and will not be erased even if the reader is disconnected from the PC. It can store up to 32 keys inside the reader non-volatile memory.

20_H (Session Key) = Volatile memory for temporarily storing keys. The keys will be erased when the reader is disconnected from the PC. Only one volatile memory is provided. The volatile key can be used as a session key for different sessions. Default value = FF FF FF FF FF FF_H.

Key: 6 Bytes. The key value loaded into the reader

E.g. {FF FF FF FF FF FF}

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = 90 00_H means the operation is completed successfully

= 63 00_H means the operation failed

Example1:

Load a key { FF FF FF FF FF FF } into the non-volatile memory location 05_H.

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

Load a key { FF FF FF FF FF FF } into the volatile memory location 20_H.

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

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 any application.
2. The content of the volatile memory "Session Key 20_H" 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 the "Internal RAM".
3. It is not recommended to use the "non-volatile key locations $00\text{-}1F_H$" to store any "temporary key" that will be changed frequently. 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, store the "key value" to the "volatile key location 20_H" instead.

4.2.4.3.1. Authentication for Mifare 1K/4K

This command is used to authenticate the Mifare 1K/4K card (PICC) using the keys stored in the reader. Two types of authentication keys are used Type_A and Type_B.

Command

Command	Class	INS	P1	P2	P3	Data In
Authentication 6 Bytes (Obsolete)	FF_H	88_H	00_H	Block Number	Key Type	Key Number

Command	Class	INS	P1	P2	Lc	Data In
Authentication 10 Bytes	FF_H	86_H	00_H	00_H	05_H	Authenticate Data Bytes

Where:

Authenticate Data Bytes: 5 Bytes.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Version 01_H	00_H	Block Number	Key Type	Key Number

Block Number: 1 Byte. The memory block to be authenticated.

Note: For Mifare 1K Card, it has a total of 16 sectors and each sector consists of 4 consecutive blocks. Ex. Sector 00_H consists of Blocks {00, 01, 02 and 03_H}; Sector 01_H consists of Blocks {04, 05, 06 and 07_H}; the last sector $0F_H$ consists of Blocks {3C, 3D, 3E and $3F_H$}.

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

Key Type: 1 Byte.

60_H = Key is used as Type A key for authentication

61_H = Key is used as Type B key for authentication

Key Number: 1 Byte.

00_H – $1F_H$ = Non-volatile memory for storing keys. The keys are permanently stored in the reader and will not be erased even if the reader is disconnected from the PC. It can store up to 32 keys inside the reader non-volatile memory.

20_H (Session Key) = Volatile memory for temporarily storing keys. The keys will be erased when the reader is disconnected from the PC. Only 1 volatile memory is provided. The volatile key can be used as a session key for different sessions. Default value = FF FF FF FF FF FF_H.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = 90 00_H means the operation is completed successfully

= 63 00_H means the operation failed

Sectors (Total of 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	0x00 ~ 0x02	0x03
Sector 1	0x04 ~ 0x06	0x07
..		
..		
Sector 14	0x38 ~ 0x0A	0x3B
Sector 15	0x3C ~ 0x3E	0x3F

1K Bytes

Table 6: Mifare 1K Memory Map

Sectors (Total of 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	0x00 ~ 0x02	0x03
Sector 1	0x04 ~ 0x06	0x07
...		
...		
Sector 30	0x78 ~ 0x7A	0x7B
Sector 31	0x7C ~ 0x7E	0x7F

2K Bytes

Sectors (Total of 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	0x80 ~ 0x8E	0x8F
Sector 33	0x90 ~ 0x9E	0x9F
...		
...		
Sector 38	0xE0 ~ 0xEE	0xEF
Sector 39	0xF0 ~ 0xFE	0xFF

2K Bytes

Table 7: Mifare 4K Memory Map

Example1: To authenticate Block 04_H with the following characteristics: Type A, key number 00_H, from PC/SC V2.01 (Obsolete).

APDU = { FF 88 00 04 60 00 }

Example2: Similar to the previous example, to authenticate Block 04_H with the following characteristics: Type A, key number 00_H, from PC/SC V2.07.

APDU = { FF 86 00 00 05 01 00 04 60 00 }

Note: Mifare Ultralight does not need authentication since it provides free access to the user data area.

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 8: Mifare Ultralight Memory Map

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

4.2.4.3.2. Read Binary Blocks

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

Command

Command	Class	INS	P1	P2	Le
Read Binary Blocks	FF_H	$B0_H$	00_H	Block Number	Number of Bytes to Read

Where:

Block Number: 1 Byte. Starting Block.

Number of Bytes to Read: 1 Byte. The length of the bytes to be read can be a multiple of 16 bytes for Mifare 1K/4K or a multiple of 4 bytes for Mifare Ultralight

Maximum of 16 bytes for Mifare Ultralight.

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

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

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

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

Note: For security considerations, 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.

Response

Response	Data Out		
Result	Data (Multiple of 4 or 16 bytes)	SW1	SW2

Where:

SW1, SW2 = 90 00_H means the operation is completed successfully

= 63 00_H means the operation failed

Example 1: Read 16 bytes from the binary block 04_H (Mifare 1K or 4K)

APDU = { FF B0 00 04 10_H }

Example 2: Read 240 bytes starting from the binary block 80_H (Mifare 4K). Block 80_H to Block $8E_H$ (15 blocks)

APDU = { FF B0 00 80 F0 }

4.2.4.3.3. Update Binary Blocks

This 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.

Command

Command	Class	INS	P1	P2	Le	Data In
Update Binary Blocks	FF_H	$D6_H$	00_H	Block Number	Number of Bytes to Update	Block Data (Multiple of 16 Bytes)

Where:

Block Number: 1 Byte. Starting Block.

Number of Bytes to Read: 1 Byte. The length of the bytes to be read can be a multiple of 16 bytes for Mifare 1K/4K or a multiple of 4 bytes for Mifare Ultralight

Maximum of 16 bytes for Mifare Ultralight.

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

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

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

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

Note: For security considerations, 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.

Block Data: Multiple of 16 + 2 Bytes, or 6 Bytes. Data to be written into the binary blocks.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = 90 00_H means the operation is completed successfully

= 63 00_H means the operation failed

Example 1: Update the binary block 04_H of Mifare 1K/4K with Data {00 01 .. 0F}

APDU = { FF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F }

Example 2: Update the binary block 04_H of Mifare Ultralight with Data { 00 01 02 03_H }

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

4.2.4.3.4. Value Block Operation (Increment, Decrement, Store)

This command is used to manipulate value-based transactions (e.g. increment a value block, etc.).

Command

Command	Class	INS	P1	P2	Lc	Data In	
Value Block Operation	FF_H	$D7_H$	00_H	Block Number	05_H	VB_OP	VB_Value (4 Bytes) {MSB…LSB}

Where:

Block Number: 1 Byte. Value Block to be manipulated

VB_OP: 1 Byte. Value block operation.

00_H = Store VB_Value into the block. The block will then be converted to a value block.

01_H = Increment the value of the value block by the VB_Value. This command is only valid for value blocks.

02_H = Decrement the value of the value block by the VB_Value. This command is only valid for value blocks.

VB_Value: 4 Byte. The value used for manipulation. The value is a signed long integer.

Example 1: Decimal - 4 = { $FF\ FF\ FF\ FC_H$ }

VB_Value			
MSB			LSB
FF_H	FF_H	FF_H	FC_H

Example 2: Decimal 1 = { $00\ 00\ 00\ 01_H$ }

VB_Value			
MSB			LSB
00_H	00_H	00_H	01_H

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = $90\ 00_H$ means the operation is completed successfully

= $63\ 00_H$ means the operation failed

4.2.4.3.5. Read Value Block

This command is used to for retrieving the value from the value block. This command is only valid for value blocks.

Command

Command	Class	INS	P1	P2	Le
Read Value Block	FF_H	$B1_H$	00_H	Block Number	00_H

Where:

Block Number. 1 Byte. The value block to be accessed.

Response

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

Where:

Value. 4 Bytes. The value returned from the cards. The value is a signed long integer.

Example 1: Decimal - 4 = { FF FF FF FC_H }

VB_Value			
MSB			LSB
FF_H	FF_H	FF_H	FC_H

Example 2: Decimal 1 = { 00 00 00 01_H }

VB_Value			
MSB			LSB
00_H	00_H	00_H	01_H

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = 90 00_H means the operation is completed successfully

= 63 00_H means the operation failed

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

4.2.4.3.6. Copy Value Block

This command is used to copy a value from a value block to another value block.

Command

Command	Class	INS	P1	P2	Lc	Data In	
Copy Value Block	FF_H	$D7_H$	00_H	Source Block Number	02_H	03_H	Target Block Number

Where:

Source Block Number: 1 Byte. Block number where the value will come from and copied to the target value block.

Target Block Number: 1 Byte. Block number where the value from the source block will be copied to. The source and target value blocks must be in the same sector.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = $90\ 00_H$ means the operation is completed successfully

= $63\ 00_H$ means the operation failed

Example 1: Store a value "1" into block 05_H

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

Example 2: Read the value block 05_H

APDU = {FF B1 00 05 00_H }

Example 3: Copy the value from value block 05_H to value block 06_H

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

Example 4: Increment the value block 05_H by "5"

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

4.2.4.4. Access PC/SC Compliant Tags (ISO14443-4)

Basically, all ISO 14443-4 compliant cards (PICCs) would understand the ISO 7816-4 APDUs. The ACR1281U-C1 Reader needs to communicate with the ISO 14443-4 compliant cards through exchanging ISO 7816-4 APDUs and Responses. ACR1222U will handle the ISO 14443 Parts 1-4 Protocols internally.

The Mifare 1K, 4K, MINI and 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 topic "PICC Commands for Mifare Classic Memory Tags".

Command

Command	Class	INS	P1	P2	Lc	Data In	Le
ISO7816 Part 4 Command					Length of the Data In		Expected Length of the Response Data

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1, SW2 = `90 00`H means the operation is completed successfully

= `63 00`H means the operation failed

Typical sequence may be:

- Present the Tag and Connect the PICC Interface
- 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 3A`H

In which,

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

Step 2) **Send an APDU, Get Challenge.**

`<< 00 84 00 00 08`H

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

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

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

Example: ISO 7816-4 APDU

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

APDU = { 80 B2 80 00 08_H }

Class = 80_H; INS = $B2_H$; P1 = 80_H; P2 = 00_H;

Lc = None; Data In = None; Le = 08_H

Answer: 00 01 02 03 04 05 06 07 [$90 00_H]

4.3. Peripherals Control

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

Note: The driver will add the Class, INS and P1 automatically.

4.3.1. Get Firmware Version

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

Command

Command	Class	INS	P1	P2	Lc
Get Firmware Version	$E0_H$	00_H	00_H	18_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	Number of Bytes to be Received	Firmware Version

Example:

Response = E1 00 00 00 0F 41 43 52 31 32 38 31 55 5F 56 35 30 33 2E 31

Firmware Version (HEX) = 41 43 52 31 32 38 31 55 5F 56 35 30 33 2E 31

Firmware Version (ASCII) = "ACR1281U_V503.1"

4.3.2. LED Control

This command is used to control the LEDs output.

Command

Command	Class	INS	P1	P2	Lc	Data In
LED Control	$E0_H$	00_H	00_H	29_H	01_H	LED Status

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	LED Status

Where:

LED Status: 1 Byte.

LED Status	Description	Description
Bit 0	Red LED	1 = ON 0 = OFF
Bit 1	Green LED	1 = ON 0 = OFF
Bit 2 – 7	RFU	RFU

4.3.3. LED Status

This command is used to check the existing LEDs status.

Command

Command	Class	INS	P1	P2	Lc
LED Status	$E0_H$	00_H	00_H	29_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	LED Status

Where:

LED Status: 1 Byte.

LED Status	Description	Description
Bit 0	Red LED	1 = ON 0 = OFF
Bit 1	Green LED	1 = ON 0 = OFF
Bit 2 – 7	RFU	RFU

4.3.4. Buzzer Control

This command is used to control the buzzer output.

Command

Command	Class	INS	P1	P2	Lc	Data In
Buzzer Control	$E0_H$	00_H	00_H	28_H	01_H	Buzzer On Duration

Where:

Buzzer On Duration: 1 Byte.

01 – FF_H = Duration (unit: 10 ms)

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	00_H

4.3.5. Set Default LED and Buzzer Behaviors

This command is used to set the default behavior for the LEDs and buzzer.

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Default LED and Buzzer Behaviors	$E0_H$	00_H	00_H	21_H	01_H	Default Behaviors

Where:

Default Behaviors: 1 Byte. Default value = $8F_H$.

LED Status	Description	Description
Bit 0	ICC Activation Status LED	To show the activations 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	RFU	RFU
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 = Disable
Bit 5	Contactless Chip Reset Indication Buzzer	To make a beep when the contactless chip is reset. 1 = Enable 0 = Disable
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 blink the LED whenever the card (PICC or ICC) is being accessed.

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

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	E1$_H$	00$_H$	00$_H$	00$_H$	01$_H$	Default Behaviors

4.3.6. Read Default LED and Buzzer Behaviors

This command is used to set the read the current default behaviors for LEDs and buzzer.

Command

Command	Class	INS	P1	P2	Lc
Read Default LED and Buzzer Behaviors	E0$_H$	00$_H$	00$_H$	21$_H$	00$_H$

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	E1$_H$	00$_H$	00$_H$	00$_H$	01$_H$	Default Behaviors

Where:

Default Behaviors: 1 Byte. Default value = 8F$_H$.

LED Status	Description	Description
Bit 0	ICC Activation Status LED	To show the activations 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	RFU	RFU
Bit 3	RFU	RFU

LED Status	Description	Description
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 = Disable
Bit 5	Contactless Chip Reset Indication Buzzer	To make a beep when the contactless chip is reset. 1 = Enable 0 = Disable
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 blink the LED whenever the card (PICC or ICC) is being accessed.

4.3.7. Initialize Cards Insertion Counter

This command is used to initialize the cards insertion/detection counter.

Command

Command	Class	INS	P1	P2	Lc	Data In			
Initialize Cards Insertion Counter	$E0_H$	00_H	00_H	09_H	04_H	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).

Response

Response	Class	INS	P1	P2	Le
Result	$E1_H$	00_H	00_H	00_H	00_H

4.3.8. Read Cards Insertion Counter

This command is used to check the cards insertion/detection counter value.

Command

Command	Class	INS	P1	P2	Lc
Read Cards Insertion Counter	$E0_H$	00_H	00_H	09_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out			
Result	$E1_H$	00_H	00_H	00_H	04_H	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).

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

4.3.9. Update Cards Insertion Counter

This command is used to update the cards insertion/detection counter value.

Command

Command	Class	INS	P1	P2	Lc
Update Cards Insertion Counter	$E0_H$	00_H	00_H	$0A_H$	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out			
Result	$E1_H$	00_H	00_H	00_H	04_H	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).

4.3.10. Set Automatic PICC Polling

This 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 PC/SC Escape Command interface. To meet the energy saving requirement, special modes are provided for turning off the antenna field whenever the PICC is inactive, or no PICC is found. The reader will consume less current in power saving mode.

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Automatic PICC Polling	$E0_H$	00_H	00_H	23_H	01_H	Polling Setting

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Polling Setting

Where:

Polling Setting: 1 Byte. Default value = $8F_H$.

Polling Setting	Description	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	RFU	RFU
Bit 5 – 4	PICC Polling Interval for PICC	Bit 5 – Bit 4: 0 – 0 = 250 ms 0 – 1 = 500 ms 1 – 0 = 1000 ms 1 – 1 = 2500 ms
Bit 6	RFU	RFU
Bit 7	Enforce ISO14443A Part 4	1 = Enable 0 = Disable

Notes:

1. It is recommended to enable the option "Turn off Antenna Field is 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 60 mA, while the Idle Current Consumption in Non-Power Saving mode is about 130 mA. Idle Current Consumption = PICC is not activated.
3. The reader will activate the ISO 14443A-4 mode of the "ISO14443A-4 compliant PICC" automatically. Type B PICC will not be affected by this option.
4. The JCOP30 card comes with two modes: ISO14443A-3 (Mifare 1K) and ISO14443A-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

4.3.11. Read Automatic PICC Polling

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

Command

Command	Class	INS	P1	P2	Lc
Read Automatic PICC Polling	$E0_H$	00_H	00_H	23_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Polling Setting

Where:

Polling Setting: 1 Byte. Default value = $8F_H$.

Polling Setting	Description	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	RFU	RFU
Bit 5 – 4	PICC Polling Interval for PICC	Bit 5 – Bit 4: 0 – 0 = 250 ms 0 – 1 = 500 ms 1 – 0 = 1000 ms 1 – 1 = 2500 ms
Bit 6	RFU	RFU
Bit 7	Enforce ISO14443A Part 4	1 = Enable 0 = Disable

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

4.3.12. Manual PICC Polling

This command is used to determine if any PICC is within the detection range of the reader. This command can be used if the Automatic PICC Polling function is disabled.

Command

Command	Class	INS	P1	P2	Lc	Data In
Manual PICC Polling	$E0_H$	00_H	00_H	22_H	01_H	$0A_H$

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Status

Where:

Status: 1 Byte.

00_H = PICC is detected

FF_H = No PICC is detected

4.3.13. Set the PICC Operating Parameter

The command is used to set the PICC Operating Parameter.

Command

Command	Class	INS	P1	P2	Lc	Data In
Set the PICC Operating Parameter	$E0_H$	00_H	00_H	20_H	01_H	Operating Parameter

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Operating Parameter

Where:

Operating Parameter: 1 Byte. Default value = 03_H.

Operating Parameter	Parameter	Description	Option
Bit 0	ISO14443 Type A	The tag types to be detected during PICC Polling	1 = Detect 0 = Skip
Bit 1	ISO14443 Type B		1 = Detect 0 = Skip
Bit 2 – 7	RFU	RFU	RFU

4.3.14. Read the PICC Operating Parameter

This command is used to check current PICC Operating Parameter.

Command

Command	Class	INS	P1	P2	Lc
Read the PICC Operating Parameter	$E0_H$	00_H	00_H	20_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Operating Parameter

Where:

Operating Parameter: 1 Byte.

Operating Parameter	Parameter	Description	Option
Bit 0	ISO14443 Type A	The tag types to be detected during PICC Polling	1 = Detect 0 = Skip
Bit 1	ISO14443 Type B		1 = Detect 0 = Skip
Bit 2 – 7	RFU	RFU	RFU

4.3.15. Set the Exclusive Mode

This command is used to set the reader into / out from Exclusive mode.

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Exclusive Mode	$E0_H$	00_H	00_H	$2B_H$	01_H	New Mode Configuration

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	Mode Configuration	Current Mode Configuration

Where:

Exclusive Mode: 1 Byte.

00_H = Share Mode: ICC and PICC interfaces can work at the same time.

01_H = Exclusive Mode: PICC is disabled when Auto Polling and Antenna Power Off when ICC is inserted (Default).

4.3.16. Read the Exclusive Mode

This command is used to check current Exclusive mode setting.

Command

Command	Class	INS	P1	P2	Lc
Read Exclusive Mode	$E0_H$	00_H	00_H	$2B_H$	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	Mode Configuration	Current Mode Configuration

Where:

Exclusive Mode: 1 Byte.

00_H = Share Mode: ICC and PICC interfaces can work at the same time.

01_H = Exclusive Mode: PICC is disabled when Auto Polling and Antenna Power Off when ICC is inserted (Default).

4.3.17. 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,

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Auto PPS	$E0_H$	00_H	00_H	24_H	01_H	Max Speed

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	Max Speed	Current Speed

Where:

Max Speed: 1 Byte. Maximum Speed.

Current Speed: 1 Byte. Current Speed.

00_H = 106 kbps; default setting, equal to No Auto PPS

01_H = 212 kbps

02_H = 424 kbps

03_H = 848 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 is the PICC or environment does not meet the requirement of the proposed communication speed.
2. The reader supports different speed between sending and receiving.

4.3.18. Read Auto PPS

This command is used to check the current Auto PPS Setting.

Command

Command	Class	INS	P1	P2	Lc
Read Auto PPS	$E0_H$	00_H	00_H	24_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	Max Speed	Current Speed

Where:

Max Speed: 1 Byte. Maximum Transmission Speed.

Current Speed: 1 Byte. Current Transmission Speed.

00_H = 106 kbps; default setting; equal to No Auto PPS

01_H = 212 kbps

02_H = 424 kbps

03_H = 848 kbps

4.3.19. Set Antenna Field

This command is used for turning on/off the antenna field.

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Antenna Field	$E0_H$	00_H	00_H	25_H	01_H	Status

Where:

Status: 1 Byte.

00_H = Disable Antenna Field

01_H = Enable Antenna Field

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Status

Where:

Status: 1 Byte.

00_H = PICC Power Off

01_H = PICC Idle

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

4.3.20. Read Antenna Field Status

This command is used to check the current Antenna Field status.

Command

Command	Class	INS	P1	P2	Lc
Read Antenna Field	$E0_H$	00_H	00_H	25_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Status

Where:

Status: 1 Byte.

00_H = PICC Power Off

01_H = PICC Idle. Ready to poll contactless tag, but not detected.

02_H = PICC Ready. PICC Request (Refer to ISO 14443) Success, i.e. contactless tag detected

03_H = PICC Selected. PICC Select (Refer to ISO 14443) Success.

04_H = PICC Activated. PICC Activation (Refer to ISO 14443) Success, ready for APDU exchange.

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

4.3.21. Set User Extra Guard Time Setting

This command is used to set the extra guard time for ICC communication. The user extra guard time will be stored into EEPROM.

Command

Command	Class	INS	P1	P2	Lc	Data In	
Set User Extra Guard Time Setting	$E0_H$	00_H	00_H	$2E_H$	02_H	ICC User Guard Time	SAM User Guard Time

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	ICC User Guard Time	SAM User Guard Time

Where:

ICC User Guard Time: 1 Byte. User Guard Time value for ICC.

SAM User Guard Time: 1 Byte. User Guard Time value for SAM.

4.3.22. Read User Extra Guard Time

This command is used to read the set extra guard time for ICC communication.

Command

Command	Class	INS	P1	P2	Lc
Read User Extra Guard Time Setting	$E0_H$	00_H	00_H	$2E_H$	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	ICC User Guard Time	SAM User Guard Time

Where:

ICC User Guard Time: 1 Byte. User Guard Time value for ICC.

SAM User Guard Time: 1 Byte. User Guard Time value for SAM.

4.3.23. Set "616C" Auto Handle Option Setting

This command is used to set the "616C" auto handle option. This command optional for T=0, ACOS5.

Command

Command	Class	INS	P1	P2	Lc	Data In	
Set "616C" Auto Handle Option	$E0_H$	00_H	00_H	32_H	02_H	ICC Option	SAM Option

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	ICC User Guard Time	SAM User Guard Time

Where:

ICC / SAM Option: 1 Byte.

FF_H = Enable "616C" Auto Handle

00_H = Disable "616C" Auto Handle (Default)

4.3.24. Read "616C" Auto Handle Option

This command is used to read the "616C" auto handle option.

Command

Command	Class	INS	P1	P2	Lc
Read "616C" Auto Handle Option	$E0_H$	00_H	00_H	32_H	00_H

Response

Response	Class	INS	P1	P2	Le	Data Out	
Result	$E1_H$	00_H	00_H	00_H	02_H	ICC User Guard Time	SAM User Guard Time

Where:

ICC / SAM Option: 1 Byte.

FF_H = Enable "616C" Auto Handle

00_H = Disable "616C" Auto Handle (Default)

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

4.3.25. Refresh Interface Status

This command is used to refresh the specified interface.

Command

Command	Class	INS	P1	P2	Lc	Data In
Refresh Interface Status	$E0_H$	00_H	00_H	$2D_H$	01_H	Interface No.

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	$E1_H$	00_H	00_H	00_H	01_H	Interface No.

Where:

Interface No.: 1 Byte. Interface to be refreshed.

01_H = ICC Interface

02_H = PICC Interface

04_H = SAM Interface

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

Appendix A. Basic Program Flow for Contactless Applications

Step 0. Start the application. The reader will do the PICC Polling and scan for tags continuously. Once the tag is found and detected, the corresponding ATR will be sent to the PC.

Step 1. Connect the "ACR1281U PICC Interface" with T=1 protocol.

Step 2. Access the PICC by exchanging APDUs.

..

Step N. Disconnect the "ACR1281U PICC Interface". Shut down the application.

Appendix B. Access DESFire Tags (ISO 14443-4)

The DESFire supports ISO 7816-4 APDU Wrapping and Native modes. Once the DESFire Tag is activated, the first APDU sent to the DESFire Tag will determine the "Command Mode". If the first APDU is "Native Mode", the rest of the APDUs must be in "Native Mode" format. Similarly, if the first APDU is "ISO 7816-4 APDU Wrapping Mode", the rest of the APDUs must be in "ISO 7816-4 APDU Wrapping Mode" format.

Example 1: DESFire ISO 7816-4 APDU Wrapping.

To read 8 bytes random number from an ISO 14443-4 Type A PICC (DESFire)

APDU = {90 0A 00 00 01 00 00}

Class = 0x90; INS = 0x0A (DESFire Instruction); P1 = 0x00; P2 = 0x00

Lc = 0x01; Data In = 0x00; Le = 0x00 (Le = 0x00 for maximum length)

Answer: 7B 18 92 9D 9A 25 05 21 [$91AF]

> **Note:** Status Code {91 AF} is defined in DESFire specification. Please refer to the DESFire specification for more details.

Example 2: DESFire Frame Level Chaining (ISO 7816 wrapping mode)

In this example, the application has to do the "Frame Level Chaining".

To get the version of the DESFire card:

Step 1: Send an APDU {90 60 00 00 00} to get the first frame. INS=0x60

Answer: 04 01 01 00 02 18 05 91 AF [$91AF]

Step 2: Send an APDU {90 AF 00 00 00} to get the second frame. INS=0xAF
Answer: 04 01 01 00 06 18 05 91 AF [$91AF]

Step 3: Send an APDU {90 AF 00 00 00} to get the last frame. INS=0xAF

Answer: 04 52 5A 19 B2 1B 80 8E 36 54 4D 40 26 04 91 00 [$9100]

Example 3: DESFire Native Command.

We can send Native DESFire Commands to the reader without ISO 7816 wrapping if we find that the Native DESFire Commands are easier to handle.

To read 8 bytes random number from an ISO 14443-4 Type A PICC (DESFire)

APDU = {0A 00}

Answer: AF 25 9C 65 0C 87 65 1D D7 [$1DD7]

In which, the first byte "AF" is the status code returned by the DESFire Card.

The Data inside the blanket [$1DD7] can simply be ignored by the application.

Example 4: DESFire Frame Level Chaining (Native Mode)

In this example, the application has to do the "Frame Level Chaining".

To get the version of the DESFire card:

Step 1: Send an APDU {60} to get the first frame. INS=0x60

Answer: AF 04 01 01 00 02 18 05 [$1805]

Step 2: Send an APDU {AF} to get the second frame. INS=0xAF
Answer: AF 04 01 01 00 06 18 05 [$1805]

Step 3: Send an APDU {AF} to get the last frame. INS=0xAF

Answer: 00 04 52 5A 19 B2 1B 80 8E 36 54 4D 40 26 04[$2604]

> **Note:** In DESFire Native Mode, the status code [90 00] will not be added to the response if the response length is greater than 1. If the response length is less than 2, the status code [90 00] will be added in order to meet the requirement of PC/SC. The minimum response length is 2.

Appendix C. Extended APDU Example

Card: ACOS7 (supports Extended APDU, echo response)

Write CMD: 80 D2 00 00 XX XX XX$_{H}$

CLA = 80$_{H}$

INS = D2$_{H}$

P1 = 00$_{H}$

P2 = 00$_{H}$

Data Len = XX XX XX$_{H}$

Example1: APDU length = 263 bytes

APDU Command:

80D20000000100000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1
E1F202122232425262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F40414243
4445464748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F6061626364656667686
96A6B6C6D6E6F707172737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E
8F909192939495969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B
4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9
DADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEF
F$_{H}$

Response:

000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223242
5262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A
4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F7
0717273747576777879 7A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495
969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BAB
BBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0
E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF9000$_{H}$

Example2: APDU length = 775 bytes

APDU Command:

80D20000000300000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1
E1F202122232425262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F40414243
4445464748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F6061626364656667686
96A6B6C6D6E6F707172737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E
8F909192939495969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B
4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9
DADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEF
F000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F2021222324
25262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F40414243444546474849 4
A4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F
707172737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F90919293949
5969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BA
BBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE
0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF000102030405
060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2
B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F50

5152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F7071727374757
67778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A9B
9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C
1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5E6
E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF$_H$

Response:

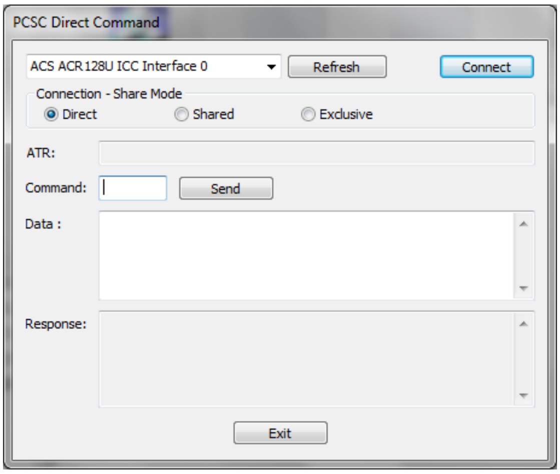
000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223242
5262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A
4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F7
0717273747576777879 7A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495
969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BAB
BBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0
E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF0001020304050
60708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2B
2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F505
152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717273747576
7778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A9B9
C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1
C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5E6E
7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF000102030405060708090A0B0C
0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F30313
233343536373839 3A3B3C3D3E3F404142434445464748494A4B4C4D4E4F5051525354555657
58595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F707172737475767778797A7B7C7
D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A9B9C9D9E9FA0A1A2
A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C
8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECED
EEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF9000$_H$

Appendix D. Escape Command Example

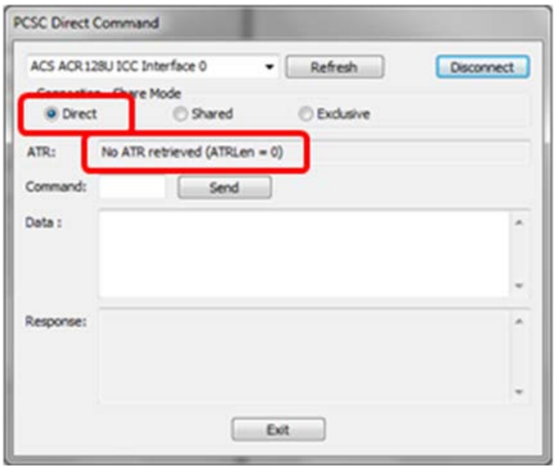
Example: Get Firmware Version (Using PCSCDirectCommand.exe)

Step 1: Plug in the ACR1281 Reader to PC

Step 2: Open the PCSCDirectCommand.exe



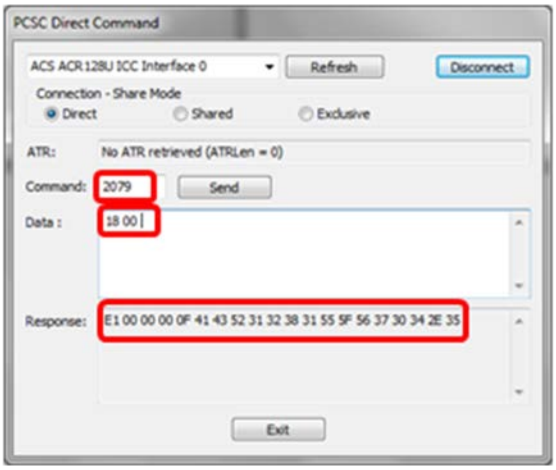
Step 3: Connect the reader in Direct mode. The ATR will be displayed (if a card is present) or "No ATR retrieved (ATRLen = 0)" will be displayed (if no card).



Step 4: Enter Command: "2079"

Enter Data: "18 00" (APDU for Get Firmware Version)

Click enter to send to reader, then check the Response



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

Appendix E. 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
00_H	Auto-select T=0 or T=1 communication protocol
01_H	I2C memory card (1k, 2k, 4k, 8k and 16k bits)
02_H	I2C memory card (32k, 64k, 128k, 256k, 512k and 1024k bits)
03_H	RFU
04_H	RFU
05_H	Infineon SLE4418 and SLE4428
06_H	Infineon SLE4432 and SLE4442
07_H	Infineon SLE4406, SLE4436 and SLE5536
08_H	Infineon SLE4404
09_H	RFU

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

Appendix F. ACR128 Compatibility

Below is the list of ACR128 functions that are implemented differently or not supported by ACR1281U-C1.

Functions	ACR128	ACR128U-C1
1. Change the default FWI and Transmit Frame Size of the activated PICC	1F 03 [Data: 3 bytes]	Not supported.
2. Transceiver Setting	20 04 06 [Data: 3 bytes]	Not supported.
3. PICC Setting	2A 0C [Data: 12 bytes]	Not supported.
4. PICC T=CL Data Exchange Error Handling	2C 02 [Data:1 byte]	Not supported.
5. Read Register	19 01 [Reg. No.]	Not supported.
6. Update Register	1A 02 [Reg. No.] [Value]	Not supported.
7. PICC Polling for Specific Types	20 02 [Data: 1 byte] FF	20 01 [Data: 1 byte]
8. Buzzer Control	28 01 [Duration] Duration: 00 = Turn Off 01 – FE = Duration x 10 ms FF = Turn On	28 01 [Duration] Duration: 01 – FF = Duration x 10 ms

9. Set/Read Default LED and Buzzer Behaviors	Set: 21 01 [Data: 1 byte] Read: 21 00 Data: Bit 0 = ICC Activation Status Bit 1 = PICC Polling Status LED Bit 2 = PICC Activation Status Buzzer Bit 3 = PICC PPS Status Buzzer Bit 4 = Card Insertion and Removal Events Buzzer Bit 5 = Contactless Chip Reset Indication Buzzer Bit 6 = Exclusive Mode Status Buzzer Bit 7 = Card Operation Blinking LED	Set: 21 01 [Data: 1 byte] Read: 21 00 Data: Bit 0 = ICC Activation Status Bit 1 = PICC Polling Status LED Bit 2 = RFU Bit 3 = RFU Bit 4 = Card Insertion and Removal Events Buzzer Bit 5 = Contactless Chip Reset Indication Buzzer Bit 6 = Exclusive Mode Status Buzzer Bit 7 = Card Operation Blinking LED
10. Set/Read Automatic PICC Polling	Set: 23 01 [Data: 1 byte] Read: 23 00 Data: Bit 0 = Auto PICC Polling Bit 1 = Turn off Antenna Field if no PICC is found Bit 2 = Turn off Antenna Field if the PICC is inactive Bit 3 = Activate the PICC when detected Bit 4..5 = PICC Poll Interval for PICC Bit 6 = Test Mode Bit 7 = Enforce ISO 14443A Part 4	Set: 23 01 [Data: 1 byte] Read: 23 00 Data: Bit 0 = Auto PICC Polling Bit 1 = Turn off Antenna Field if no PICC is found Bit 2 = Turn off Antenna Field if the PICC is inactive Bit 3 = RFU Bit 4..5 = PICC Poll Interval for PICC Bit 6 = RFU Bit 7 = Enforce ISO 14443A Part 4

Table 9: ACR128U Compatibility