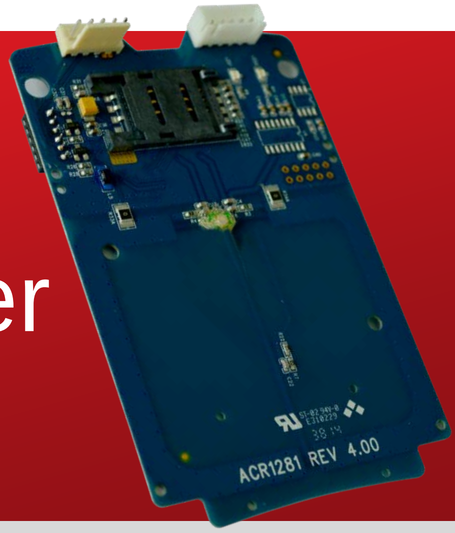
Advanced Card Systems Ltd.
Card & Reader Technologies

ACM1281S-C7

Serial Contactless Reader Module with SAM Slot



Reference Manual V1.01

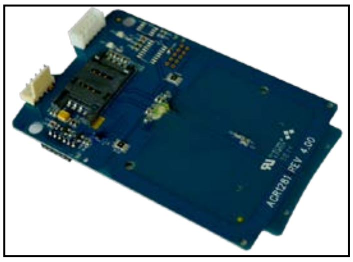
Revision History

Release Date	Revision Description	Version Number
2015-04-22	• Initial Release	1.00
2017-05-30	• Updated Section 2.0 Features	1.01

Table of Contents

List of Tables

1.0. Introduction



The ACM1281S-C7 Serial Contactless Reader Module with SAM Slot was designed based on the 13.56 MHz technology. It supports ISO 14443 Parts 1-4 Type A and B cards, and MIFARE® Classic series with a card reading distance of up to 50 mm (depending on tag type).

The ACM1281S-C7 has an integrated (on-board) antenna and comes with an optional serial cable and has additional features like USB firmware upgradability and extended APDU support.

The ACM1281S-C7 is a plug-and-play device that does not require any driver installation and is specifically designed for fast and easy integration to embedded systems. It also has a built-in ISO 7816 Compliant Class A SAM (Secure Access Module) slot which can be used together with a SAM card for high-level security in contactless transactions. It also makes use of high-speed communication for contactless cards at a maximum of 848 Kbps, which makes it suitable for highly demanding applications such as vending machine payment systems, kiosks, gaming machines and other integrated systems which have different serial ports.

This Reference Manual will discuss in detail how the PC/SC APDU commands are implemented for the contactless interface, SAM card support and device peripherals of ACM1281S-C7.

2.0. Features

- Serial RS-232 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
- Smart Card Reader:
  - Contactless Interface:
    - 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 Classic series
    - Built-in anti-collision feature (only one tag is accessed at any time)
    - Supports extended APDU (Max. 64 KB)
  - SAM Interface:
    - One SAM slot
    - Supports ISO 7816-compliant Class A SAM cards
- Built-in Peripherals:
  - Two user-controllable LEDs
  - User-controllable buzzer
- USB Firmware Upgradeability
- Compliant with the following standards:
  - ISO 14443
  - ISO 7816
  - PC/SC
  - CE
  - FCC
  - RoHS 2
  - REACH

2.1. Serial Interface

The ACM1281S-C7 is connected to a computer through a serial interface (RS-232 or RS-485).

2.1.1. Communication Parameters

The ACM1281S-C7 is connected to a host through serial interface (RS-232 or RS-485), 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: RS-232 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: RS-485 Interface Wiring

2.2. Serial Protocol

ACM1281S-C7 shall interface with the host with serial connection. CCID-like format is used for communication.

Command Format

STX (02h)	Bulk-OUT Header	APDU Command or Parameters	Checksum	ETX (03h)
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 02h

ETX – End of Text, tells the reader the command ended, must equal to 03h

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}

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

After ACM1281S receives the command, it will first response the status frame to tell the host the command status.

The Status Frame Format as below:

STX (02h)	Status	Checksum	ETX (03h)
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 ACM1281S 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 “03h”.

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 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 ACM1281S followed.

The Response Frame Format as below:

STX (02h)	Bulk-IN Header	APDU Response or abData	Checksum	ETX (03h)
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 02h

ETX – End of Text, tells the host the response ended, must be equal to 03h

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}

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

2.3. CCID-like Commands

2.3.1. Bulk-OUT Messages

The ACM1281S shall follow the CCID Bulk-OUT Messages as specified in CCID Section 6.1. In addition, this specification defines some extended commands for operating additional features. This section lists the CCID Bulk-OUT Messages to be supported by ACM1281S.

2.3.1.1. 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.*

2.3.1.2. 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.
7	*abRFU*	3		Reserved for future use.

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

2.3.1.3. 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.

2.3.1.4. 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.

2.3.1.5. 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.

Offset	Field	Size	Value	Description
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.

2.3.2. Bulk-IN Messages

The Bulk-IN messages are used in response to the Bulk-OUT messages. ACM1281S shall follow the CCID Bulk-IN Messages as specified in CCID section 6.2. This section lists the CCID Bulk-IN Messages to be supported by ACM1281S.

2.3.2.1. RDR_to_PC_DataBlock

This message is sent by ACM1281S in response to *PC_to_RDR_IccPowerOn* and *PC_to_RDR_XfrBlock* messages.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	80h	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 Spec Section 6.2.6
8	*bError*	1		Slot error register as defined in CCID Spec Section 6.2.6
9	*bChainParameter*	1	00h	RFU (TPDU exchange level).
10	*abData*	Byte array		This field contains the data returned by the CCID.

2.3.2.2. RDR_to_PC_Escape

This message is sent by ACM1281S 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 Spec Section 6.2.6
8	*bError*	1		Slot error register as defined in CCID Spec Section 6.2.6
9	*bRFU*	1	00h	RFU.
10	*abData*	Byte array		This field contains the data returned by the CCID.

2.3.2.3. RDR_to_PC_SlotStatus

This message is sent by ACM1281S 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 Spec Section 6.2.6
8	*bError*	1		Slot error register as defined in CCID Spec Section 6.2.6
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.

3.0. Contactless Smart Card Protocol

3.1. ATR Generation

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

3.1.1. 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) = 0Ch

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
00 36h: MIFARE PLUS SL1_2K
00 37h: MIFARE PLUS SL1_4K
00 38h: MIFARE PLUS SL2_2K
00 39h: MIFARE PLUS SL2_4K
00 3Ah: MIFARE Ultralight C
FF 28h: JCOP 30
FF [SAK]h: undefined tags

3.1.2. 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 MIFARE DESFire = {3B 81 80 01 80 80h} // 6 bytes of ATR

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

APDU Command = FF CA 01 00 00h

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

ATS = {06 75 77 81 02 80h}

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

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

Protocol Information of ATQB = F7 71 85h

MBLI of ATTRIB = 00h

3.2. 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.

3.2.1. Get Data

This command returns the serial number or ATS of the "connected PICC".

Get UID APDU Format (5 bytes)

Command	Class	INS	P1	P2	Le
Get Data	FFh	CAh	00h 01h	00h	00h (Max Length)

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

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

If P1 = 01h, Get ATS of 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]={FFh, CAh, 00h, 00h, 00h};

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

UINT8 GET_ATS[5]={FFh, CAh, 01h, 00h, 00h};

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

3.2.2.1. Load Authentication Keys

This 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)
00h = Key is loaded into the reader volatile memory
20h = Key is loaded into the reader non-volatile memory
Other = Reserved

Key Number (1 byte)
00h ~ 1Fh = 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.

20h (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 05h.

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 20h.

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

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 20h" 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 00h ~ 1Fh" 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 020h".*

3.2.2.2. 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 bytes):

Byte1	Byte 2	Byte 3	Byte 4	Byte 5
Version 01h	00h	Block Number	Key Type	Key Number

Where:

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

Example: Sector 00h consists of Blocks {00h, 01h, 02h and 03h}; Sector 01h consists of Blocks {04h, 05h, 06h and 07h}; the last sector 0Fh consists of Blocks {3Ch, 3Dh, 3Eh and 3Fh}. Once the authentication is done successfully, there is no need to do the authentication again provided that the blocks to be accessed are belonging to the same sector. Please refer to the MIFARE 1K/4K specification for more details.

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

Key Type (1 byte) 60h = Key is used as a TYPE A key for authentication

61h = Key is used as a TYPE B key for authentication

Key Number (1 byte) 00h ~ 1Fh = 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.

20h (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	00h ~ 02h	03h
Sector 1	04h ~ 06h	07h
..		
..		
Sector 14	38h ~ 0Ah	3Bh
Sector 15	3Ch ~ 3Eh	3Fh

1 KB

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	00h ~ 02h	03h
Sector 1	04h ~ 06h	07h
..		
..		
Sector 30	78h ~ 7Ah	7Bh
Sector 31	7Ch ~ 7Eh	7Fh

2 KB

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	80h ~ 8Eh	8Fh	2 KB
Sector 33	90h ~ 9Eh	9Fh	
..			
..			
Sector 38	E0h ~ EEh	EFh	
Sector 39	F0h ~ FEh	FFh	

Examples:

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

// PC/SC V2.01, Obsolete

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

<Similarly>

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

// PC/SC V2.07

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

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

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

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

3.2.2.3. Read Binary Blocks

This 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. Multiple 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: 10h (16 bytes). The starting block only. (Single Block Mode)

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

***Note:** For 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 04h (MIFARE 1K or 4K)

APDU = {FF B0 00 04 10h}

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

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

APDU = {FF B0 00 80 F0h}

3.2.2.4. 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.

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

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

Note: *For safety 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 Multiple 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 04h 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 04 of MIFARE Ultralight with Data {00 01 02 03h}

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

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

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

- 00h = Store the VB_Value into the block and will be converted to a value block.
- 01h = Increment the value of the value block by the VB_Value. This command is only valid for value block.
- 02h = Decrement the value of the value block by the VB_Value. This command is only valid for value block.

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

Example 1: Decimal 4 = {FFh, FFh, FFh, FCh}

VB_Value			
MSB		LSB	
FFh	FFh	FFh	FCh

Example 2: Decimal 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB		LSB	
00h	00h	00h	01h

Value Block Operation Response Format (2 Bytes)

Response	Data Out	
Result	SW1	SW2

Value Block Operation Response Codes

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

3.2.2.6. Read Value Block

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

Read Value Block APDU Format (5 bytes)

Command	Class	INS	P1	P2	Le
Read Value Block	FFh	B1h	00h	Block Number	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 = {FFh, FFh, FFh, FCh}

Value			
MSB			LSB
FFh	FFh	FFh	FCh

Example 2: Decimal 1 = {00h, 00h, 00h, 01h}

Value			
MSB			LSB
00h	00h	00h	01h

Read Value Block Response Codes

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

3.2.2.7. Copy Value Block

This 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 05h

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

// Read the value block 05h

APDU = {FF B1 00 05 00h}

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

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

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

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

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

All ISO 14443-4 compliant cards (PICCs) would understand the ISO 7816-4 APDUs. The ACM1281S reader just has to communicate with the ISO 14443-4 compliant cards through exchanging ISO 7816-4 APDUs and responses. ACM1281S will handle the ISO 14443 Parts 1-4 Protocols internally.

MIFARE 1K, MIFARE 4K, MIFARE Mini and MIFARE 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 = 80h

INS = B2h

P1 = 80h

P2 = 00h

Lc = None

Data In = None

Le = 08h

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. Get Firmware Version

This 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	E0h	00h	00h	18h	00h

Get Firmware Version Response Format (Firmware Message Length)

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	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 38 2E 30h

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

Firmware Version (ASCII) = "ACR1281S_V308.0"

4.2. LED Control

This command is used to control the LEDs' output.

LED Control Format (6 bytes)

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

LED Control Response Format (6 bytes)

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

LED Status (1 byte) – LED Control

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

4.3. LED Status

This command is used to check the existing LEDs' status.

LED Status Format (5 bytes)

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

LED Status Response Format (6 bytes)

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

LED Status (1 byte) – LED Status

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

4.4. Buzzer Control

This command is used to control the buzzer output.

Buzzer Control Format (6 bytes)

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

Where:

Buzzer On Duration 1 byte.

- 00h = Turn OFF
- 1 to FFh = Duration (unit: 10 ms)

Buzzer Control Response Format (6 bytes)

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

4.5. Set Default LED and Buzzer Behaviors

This command is used to configure the default behaviors for LEDs and buzzer of the card reader.

Set Default LED and Buzzer Behaviors Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set Default LED and Buzzer Behaviors	E0h	00h	00h	21h	01h	Default Behaviors

Default Behaviors (1 byte)

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.

Note: *Default value of Default Behaviors = FBh.*

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

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

4.6. Read Default LED and Buzzer Behaviors

This 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	E0h	00h	00h	21h	00h

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

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

Default Behaviors (1 byte)

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.

Note: *Default value of Default Behaviors = FBh.*

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

4.7. Initialize Card's Detection Counter

This command is used to initialize the card's detection counter.

Initialize Card's Detection Counter Format (9 bytes)

Command	Class	INS	P1	P2	Lc	Data In			
Initialize Card's Detection Counter	E0h	00h	00h	09h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

Initialize Card's Detection Counter Response Format (9 bytes)

Response	Class	INS	P1	P2	Lc	Data Out			
Result	E1h	00h	00h	00h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

Where:

PICC Cnt (LSB) 1 byte. PICC Detection Counter (LSB)

PICC Cnt (MSB) 1 byte. PICC Detection Counter (MSB)

4.8. Read Card's Detection Counter

This command is used to check the card's detection counter value.

Read Card's Detection Counter Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read Card's Detection Counter	E0h	00h	00h	09h	00h

Read Card's Detection Counter Response Format (9 bytes)

Response	Class	INS	P1	P2	Lc	Data Out			
Result	E1h	00h	00h	00h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

Where:

PICC Cnt (LSB) 1 byte. PICC Detection Counter (LSB)

PICC Cnt (MSB) 1 byte. PICC Detection Counter (MSB)

4.9. Update Card's Detection Counter

This command is used to update the card's detection counter value.

Update Card's Detection Counter Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Update Card's Detection Counter	E0h	00h	00h	0Ah	00h

Update Card's Detection Counter Response Format (9 bytes)

Response	Class	INS	P1	P2	Lc	Data Out			
Result	E1h	00h	00h	00h	04h	RFU	RFU	PICC Cnt (LSB)	PICC Cnt (MSB)

Where:

PICC Cnt (LSB) 1 byte. PICC Detection Counter (LSB)

PICC Cnt (MSB) 1 byte. PICC Detection Counter (MSB)

4.10. 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 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	E0h	00h	00h	23h	01h	Polling Setting

Set Automatic PICC Polling Response Format (6 bytes)

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

Polling Setting (1 byte)

Polling Setting	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 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	-
Bit 7	Enforce ISO 14443A Part 4	1= Enable; 0= Disable.

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

Reminders:

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

4.11. Read Automatic PICC Polling

This 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	E0h	00h	00h	23h	00h

Read the Configure mode Response Format (6 bytes)

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

Polling Setting (1 byte)

Polling Setting	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 ms <0 – 1> = 500 ms <1 – 0> = 1000 ms <1 – 1> = 2500 ms
Bit 6	RFU	-
Bit 7	Enforce ISO 14443A Part 4	1= Enable; 0= Disable.

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

4.12. Set the PICC Operating Parameter

This 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	E0h	00h	00h	20h	01h	Operation Parameter

Set the PICC Operating Parameter Response Format (6 bytes)

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

Operating Parameter (1 byte)

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

Note: *Default value of Operation Parameter = 03h*

4.13. Read the PICC Operating Parameter

This 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	E0h	00h	00h	20h	00h

Read the PICC Operating Parameter Response Format (6 bytes)

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

Operating Parameter (1 byte)

Operating Parameter	Parameter	Description	Option
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

4.14. 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	E0h	00h	00h	24h	01h	Max Speed

Set Auto PPS Response Format (9 bytes)

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

Where:

Max Speed 1 byte. Maximum Speed.

Current Speed 1 byte. Current Speed.

Value can be:

- 106 Kbps = 00h (No Auto PPS; default setting)
- 212 Kbps = 01h
- 424 Kbps = 02h
- 848 Kbps = 03h

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.*

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

4.15. Read Auto PPS

This command is used to check current auto PPS setting.

Read Auto PPS Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read Auto PPS	E0h	00h	00h	24h	00h

Set Auto PPS Response Format (9 bytes)

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

Where:

Max Speed 1 byte. Maximum Speed.

Current Speed 1 byte. Current Speed.

Value can be:

- 106 Kbps = 00h (No Auto PPS; default setting)
- 212 Kbps = 01h
- 424 Kbps = 02h
- 848 Kbps = 03h

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

4.16. Antenna Field Control

This 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	E0h	00h	00h	25h	01h	Status

Antenna Field Control Response Format (6 bytes)

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

Where:

Status 1 byte.

01h = Enable Antenna Field

00h = Disable Antenna Field

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

4.17. Read Antenna Field Status

This 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	E0h	00h	00h	25h	00h

Read Antenna Field Status Response Format (6 bytes)

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

Where:

Status 1 byte.

- 01h = Enable Antenna Field
- 00h = Disable Antenna Field

4.18. Set Serial Communication Mode

This 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	44h	Mode Select

Set Serial Communication Mode Response Format (2 bytes)

Response	Byte 0	Byte 1
Result	90h	Mode Select

Mode Select (1 byte) – Communication Speed and Mode Selection

Offset	Parameter	Description
Bit 0-3	Serial Communication Speed	000b= 9600 bps(Default) 001b= 19200 bps 010b= 38400 bps 011b= 57600 bps 100b= 115200 bps 101b= 128000 bps 110b= 230400 bps 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).

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