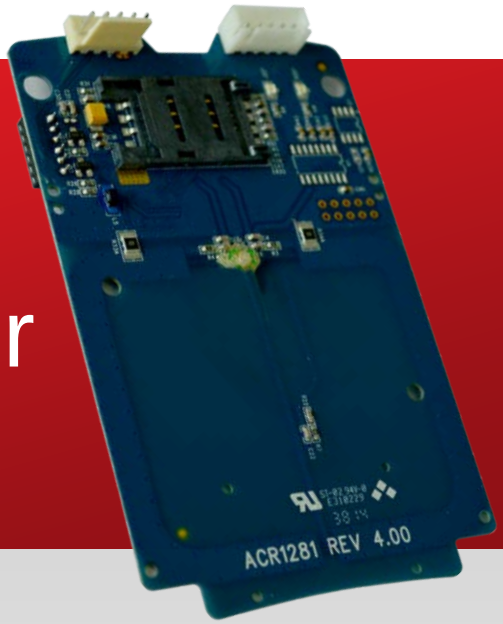
Advanced Card Systems Ltd.
Card & Reader Technologies

ACM1281U-C7

USB Contactless Reader Module with SAM Slot



Reference Manual V1.01

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

Revision History

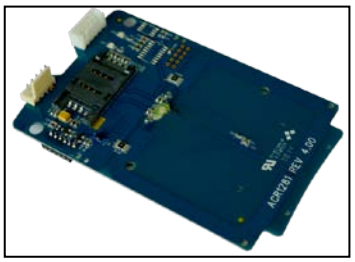
Release Date	Revision Description	Version Number
2015-04-22	• Initial Release	1.00
2017-05-31	• Updated Section 2.0 Features	1.01

Table of Contents

List of Figures

List of Tables

1.0. Introduction



The ACM1281U-C7 USB Contactless Reader Module with SAM Slot, running on the 13.56 MHz frequency, was designed for fast and easy integration to embedded systems. It makes use of the USB CCID class driver and accepts card commands from the computer application.

The ACM1281U-C7 has an integrated (on-board) antenna and comes with an optional USB cable and has additional features like firmware upgradability and extended APDU support.

It supports ISO 14443 Parts 1-4 Type A and B cards, and MIFARE Classic® series. 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 enhanced security. With a maximum of 848 Kbps high-speed communication ability for contactless cards making it suitable for highly demanding smart card applications like vending machine payment systems, kiosks, gaming machines, and other integrated systems.

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 ACM1281U-C7.

2.0. Features

- USB Full Speed Interface
- 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
    - ISO 7816-compliant Class A SAM cards
- Application Programming Interface:
  - Supports PC/SC
  - Supports CT-API (through wrapper on top of PC/SC)
- Built-in Peripherals:
  - Two user-controllable LEDs
  - User-controllable buzzer
- USB Firmware Upgradeability
- Supports Android™ 3.1 and later[1]
- Compliant with the following standards:
  - ISO 14443
  - ISO 7816
  - PC/SC
  - CCID
  - CE
  - FCC
  - RoHS 2
  - REACH
  - Microsoft® WHQL

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

3.0. ACM1281U-C7 Architecture

3.1. Reader Block Diagram

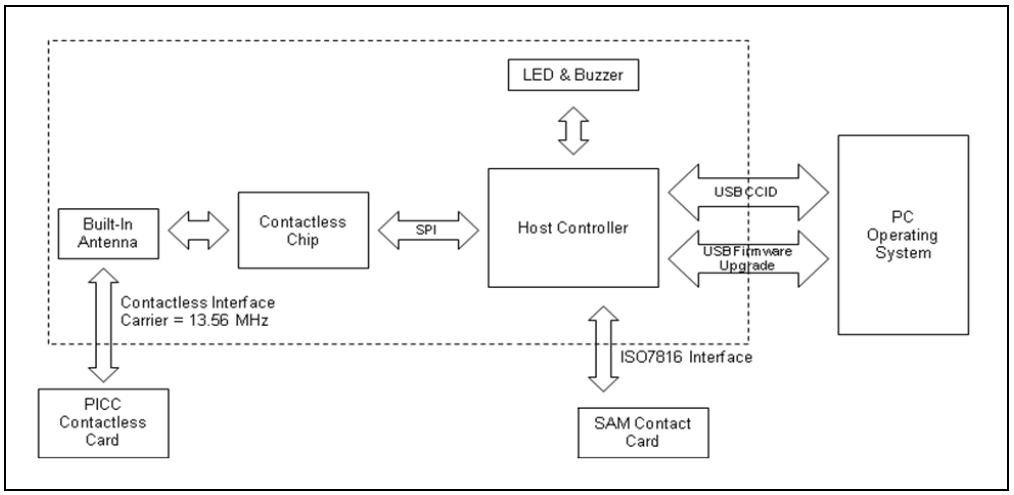


Figure 1: ACM1281U-C7 Reader Block Diagram

3.2. Communication between PC/SC driver PICC and SAM

The protocol being used between ACM1281U-C7 and the PC is CCID. All communications between PICC and SAM are PC/SC-compliant.

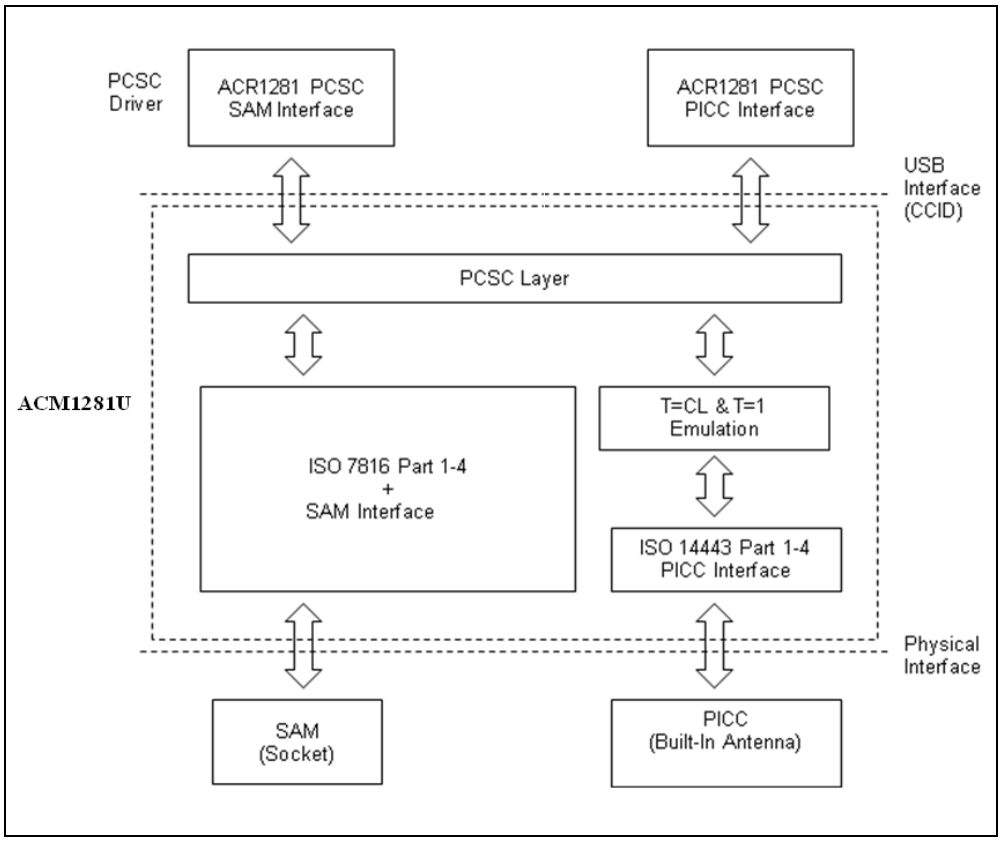


Figure 2: ACM1281U-C7 Architecture

4.0. Hardware Design

4.1. USB

The ACM1281U-C7 connects to a computer through USB following the USB standard.

4.1.1. Communication Parameters

The ACM1281U-C7 connects to a computer through USB as specified in the USB Specification 2.0. The ACM1281U-C7 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 ACM1281U-C7 and PC
3	D+	Differential signal transmits data between ACM1281U-C7 and PC
4	GND	Reference voltage level for power supply

Table 1: USB Interface Wiring

Note: *For ACM1281U-C7 to function properly through USB interface, the device driver should be installed.*

4.1.2. Endpoints

The ACM1281U-C7 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 ACM1281U-C7 (data packet size is 64 bytes).

Bulk-IN – For response to be sent from ACM1281U-C7 to host (data packet size is 64 bytes).

Interrupt-IN – For card status message to be sent from ACM1281U-C7 to host (data packet size is 8 bytes).

4.2. Contactless Smart Card Interface

The interface between the ACM1281U-C7 and the contactless card follows the specifications of ISO 14443 with certain restrictions or enhancements to increase the practical functionality of the ACM1281U-C7.

4.2.1. Carrier Frequency

The carrier frequency for ACM1281U-C7 is 13.56 MHz.

4.2.2. Card Polling

The ACM1281U-C7 automatically polls the contactless cards that are within the field. ISO 14443-4 Type A, ISO 14443-4 Type B and MIFARE cards are supported.

4.3. User Interface

4.3.1. Buzzer

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

Events	Buzzer
1. The reader is powered up and successfully initialized.	Beep
2. Card Insertion Event (PICC)	Beep
3. Card Removal Event (PICC)	Beep

Table 2: Buzzer Event

4.3.2. 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	N/A
2. PICC is present and activated.	ON	N/A
3. PICC is operating.	Blinking	N/A

Table 3: LED Indicator

5.0. Contactless Smart Card Protocol

5.1. ATR Generation

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

5.2. 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	UU	TCK	Exclusive-oring of all the bytes T0 to Tk

Table 4: ISO 14443 Part 3 ATR Format

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}

ATR											
Initial Header	**T0**	**TD1**	**TD2**	**T1**	**Tk**	**Length**	**RID**	**Standard**	**Card Name**	**RFU**	**TCK**
3Bh	8Fh	80h	01h	80h	4Fh	0Ch	A0 00 00 03 06h	03h	00h 01h	00 00 00 00h	6Ah

Where:

Length (YY) = 0Ch

RID = A0 00 00 03 06h (PC/SC Workgroup)

Standard (SS) = 03h (ISO 14443A, Part 3)

Card Name (C0 ... C1) = [00 01h] (MIFARE 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 undefined tags

5.3. 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>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>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><th>Byte1-4</th><th>Byte5-7</th><th>Byte8</th></tr><tr><td>Application Data from ATQB</td><td>Protocol Info Byte from ATQB</td><td>Higher nibble=MBLI from ATTRIB command
Lower nibble (RFU)=0</td></tr></table></td></tr>
<tr><td>XX
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 MIFARE® DESFire® as follows:

MIFARE DESFire (ATR) = 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. The ATS is returned for ISO 14443A-3 or ISO 14443B-3/4 PICCs.*

APDU Command = FF CA 01 00 00h

APDU Response = 06 75 77 81 02 90 00h

ATS = {06 75 77 81 02 80h}

Example 2: Consider the ATR from EZ-Link as follows:

EZ-Link (ATR) = 3B 88 80 01 1C 2D 94 11 F7 71 85 00 BEh

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

Protocol Information of ATQB = F7 71 85h

MBLI of ATTRIB = 00h

5.4. Pseudo APDUs for Contactless Interface

5.4.1. 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	FFh	CAh	00h 01h	00h	00h (Full Length)

Get UID Response if P1 = 00h

Response	UID	…	…	UID	SW1	SW2
Result	LSB			MSB		

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

Response	Data Out		
Result	ATS	SW1	SW2

Response Code

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

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

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

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

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

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

5.4.2.1. 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	FFh	82h	Key Structure	Key Number	06h	Key

Where:

Key Structure (1 Byte)

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

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

Other = Reserved

Key Number (1 Byte)

00h – 1Fh = 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.

20h (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 FFh.

Key (6 Bytes)

The key value loaded into the reader.

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h means the operation is completed successfully.

= 63 00h means the operation failed.

Example1:

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. *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 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 the "Internal RAM".*
3. *It is not recommended to use the "non-volatile key locations 00-1Fh" 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 20h" instead.*

5.4.2.2. 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)	FFh	88h	00h	Block Number	Key Type	Key Number

Command	Class	INS	P1	P2	Lc	Data In
Authentication 10 Bytes	FFh	86h	00h	00h	05h	Authenticate Data Bytes

Where:

Authenticate Data Bytes (5 Bytes)

Byte 1	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.

Note: *For MIFARE 1K card, it has a total of 16 sectors and each sector consists of 4 consecutive blocks. For 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 belong to the same sector. Please refer to the MIFARE 1K/4K specification for more details.

Key Type (1 Byte)

60h = Key is used as Key A key for authentication.

61h = Key is used as Key B key for authentication.

Key Number (1 Byte)

00h – 1Fh = 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.

20h (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 FFh.

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h means the operation is completed successfully.

= 63 00h 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	00h ~ 02h	03h
Sector 1	04h ~ 06h	07h
..		
..		
Sector 14	38h ~ 0Ah	3Bh
Sector 15	3Ch ~ 3Eh	3Fh

1 KB

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

2 KB

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 32	80h ~ 8Eh	8Fh
Sector 33	90h ~ 9Eh	9Fh
...		
...		
Sector 38	E0h ~ EEh	EFh
Sector 39	F0h ~ FEh	FFh

2 KB

Table 7: MIFARE 4K Memory Map

Example 1:

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

APDU = { FF 88 00 04 60 00h }

Example 2:

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

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

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

5.4.2.3. 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	FFh	B0h	00h	Block Number	Number of Bytes to Read

Where:

Block Number (1 Byte)

Starting Block

Number of Bytes to Read 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 (1 Byte).

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

Example 2: 40h (64 bytes). From 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 00h means the operation is completed successfully.

= 63 00h means the operation failed.

Example 1: Read 16 bytes from the binary block 04h (MIFARE 1K or 4K).

APDU = { FF B0 00 04 10h }

Example 2: Read 240 bytes starting from the binary block 80h (MIFARE 4K). Block 80h to Block 8Eh (15 blocks).

APDU = { FF B0 00 80 F0 }

5.4.2.4. 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.

Command

Command	Class	INS	P1	P2	Le	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)

Starting Block

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

Number of Bytes to Read 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 (1 Byte).

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: 10h (16 Bytes). Starting block only. (Single Block Mode)

Example 2: 30h (48 Bytes). From 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.*

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h means the operation is completed successfully.

= 63 00h means the operation failed.

Example 1: 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 }

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

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

5.4.2.5. 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	FFh	D7h	00h	Block Number	05h	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

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

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

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

VB_Value (4 bytes)

The value used for manipulation. The value is a signed long integer.

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

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

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

VB_Value			
MSB			LSB
00h	00h	00h	01h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h means the operation is completed successfully.

= 63 00h means the operation failed.

5.4.2.6. Read Value Block

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

Command

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.

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 FCh }

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

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

VB_Value			
MSB			LSB
00h	00h	00h	01h

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h means the operation is completed successfully.

= 63 00h means the operation failed.

5.4.2.7. 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	FFh	D7h	00h	Source Block Number	02h	03h	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 00h means the operation is completed successfully.

= 63 00h means the operation failed.

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

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

Example 2: Read the value block 05h

APDU = {FF B1 00 05 00h}

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

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

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

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

5.4.3. Access PC/SC-compliant tags (ISO 14443-4)

Basically, all ISO 14443-4 compliant cards (PICCs) can understand the ISO 7816-4 APDUs. The ACM1281U-C7 reader will only need to communicate with the ISO 14443-4 compliant cards through exchanging ISO 7816-4 APDUs and responses. ACM1281U-C7 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. Simply treat the MIFARE tags as standard ISO 14443-4 tags. For more information, see section 5.2.4.2 – PICC Commands for MIFARE 1K/4K Memory Cards.

Command

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

SW1 SW2 = 90 00h means the operation is completed successfully.

= 63 00h means the operation 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 58 [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: ISO 7816-4 APDU

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 07 [$90 00h]

6.0. 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.*

6.1. Get Firmware Version

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

Command

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

Response

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	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 37 30 32 2E 32

Firmware Version (HEX) = 41 43 52 31 32 38 31 55 5F 56 37 30 32 2E 32

Firmware Version (ASCII) = "ACR1281U_V702.2"

6.2. LED Control

This command is used to control the LEDs output.

Command

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

Response

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

6.3. LED Status

This command is used to check the existing LEDs status.

Command

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

Response

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

6.4. Buzzer Control

This command is used to control the buzzer output.

Command

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

Where:

Buzzer on Duration (1 Byte)

01 – FFh = Duration (unit: 10 ms)

Response

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

6.5. Set Default LED and Buzzer Behaviors

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

Command

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

Where:

Default Behaviors (1 Byte)

Default value = FBh.

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 make the LED blink whenever the card (PICC or ICC) is being accessed.

Response

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

6.6. Read Default LED and Buzzer Behaviors

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

Command

Command	Class	INS	P1	P2	Lc
Read Default LED and Buzzer Behaviors	E0h	00h	00h	21h	00h

Response

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

Where:

Default Behaviors (1 Byte)

Default value = FBh

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 make the LED blink whenever the card (PICC or ICC) is being accessed.

6.7. 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.

You can send a command to disable the PICC polling function by sending a command 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	E0h	00h	00h	23h	01h	Polling Setting

Response

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

Where:

Polling Setting (1 Byte)

Default value = FBh

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 ISO 14443A 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 to prevent the PICC from "warming up."*
2. *The longer the PICC Poll Interval, the more efficient it is for 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.*

6.8. Read Automatic PICC Polling

This command is used to check the current automatic PICC polling.

Command

Command	Class	INS	P1	P2	Lc
Read Automatic PICC Polling	E0h	00h	00h	23h	00h

Response

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

Where:

Polling Setting (1 Byte)

Default value = FBh

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 ISO 14443A Part 4	1 = Enable 0 = Disable

6.9. 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	E0h	00h	00h	22h	01h	0Ah

Response

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

Where:

Status (1 Byte)

00h = PICC is detected

FFh = No PICC is detected

6.10. Set 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	E0h	00h	00h	20h	01h	Operating Parameter

Response

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

Where:

Operating Parameter (1 Byte)

Default value = 03h

Operating Parameter	Parameter	Description	Option
Bit 0	ISO 14443 Type A	The tag types to be detected during PICC Polling	1 = Detect 0 = Skip
Bit 1	ISO 14443 Type B		1 = Detect 0 = Skip
Bit 2 – 7	RFU	RFU	RFU

6.11. Read 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	E0h	00h	00h	20h	00h

Response

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

Where:

Operating Parameter (1 byte)

Operating Parameter	Parameter	Description	Option
Bit 0	ISO 14443 Type A	The tag types to be detected during PICC Polling	1 = Detect 0 = Skip
Bit 1	ISO 14443 Type B		1 = Detect 0 = Skip
Bit 2 – 7	RFU	RFU	RFU

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. Accessing MIFARE DESFire tags (ISO 14443-4)

MIFARE 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 00h}

Class = 90h; INS = 0Ah (DESFire Instruction); P1 = 00h; P2 = 00h

Lc = 01h; Data In = 00h; Le = 00h (Le = 00h for maximum length)

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

Note: *Status Code {91 AFh} is defined in MIFARE DESFire specification. Please refer to MIFARE 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 00h} to get the first frame. INS=60h

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

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

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

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

Example 3: DESFire Native Command.

You 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 00h}

Answer: AF 25 9C 65 0C 87 65 1D D7h [$1DD7h]

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

The Data inside the blanket [$1DD7h] 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 {60h} to get the first frame. INS=60h

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

Step 2: Send an APDU {AFh} to get the second frame. INS=AFh

Answer: AF 04 01 01 00 06 18 05h [$1805h]

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

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

Note: *In DESFire Native Mode, the status code [90 00h] 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 00h] 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 XXh

CLA = 80h

INS = D2h

P1 = 00h

P2 = 00h

Data Len = XX XX XXh

Example 1: APDU length = 263 bytes

APDU Command:

80D20000000100001020304050607080 90A0B0C0D0E0F101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F40414243444546
4748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6
D6E6F707172737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F90919293
9495969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B
8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9D
ADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFD
FEFFh

Response:

000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F20212223242526
2728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4
D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717273
7475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F909192939495969798999A
9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDB
EBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE
0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF9000h

Example 2: APDU length = 775 bytes

APDU Command:

80D20000000300001020304050607080 90A0B0C0D0E0F101112131415161718191A1B1C1D1E1F
202122232425262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F40414243444546
4748494A4B4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6
D6E6F707172737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F90919293
9495969798999A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B
8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9D
ADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFD
FEFF000102030405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F2021222324
25262728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B
4C4D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717
2737475767778797A7B7C7D7E7F808182838485868788898A8B8C8D8E8F9091929394959697989
99A9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBC
BDBEBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDE
DFE0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF0001020
30405060708090A0B0C0D0E0F101112131415161718191A1B1C1D1E1F202122232425262728292
A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F50
515253545556575859 5A5B5C5D5E5F606162636465666768696A6B6C6D6E6F7071727374757677

78797A7B7C7D7E7F80818283848586878889 8A8B8C8D8E8F909192939495969798999A9B9C9D9
E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1
C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3
E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFFh

Response:

000102030405060708090A0B0C0D0E0F10111213141516171819 1A1B1C1D1E1F20212223242526
2728292A2B2C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4
D4E4F505152535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717273
7475767778797A7B7C7D7E7F80818283848586878889 8A8B8C8D8E8F909192939495969798999A
9B9C9D9E9FA0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDB
EBFC0C1C2C3C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE
0E1E2E3E4E5E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF00010203040
5060708090A0B0C0D0E0F10111213141516171819 1A1B1C1D1E1F202122232425262728292A2B2
C2D2E2F303132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F505152
535455565758595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F70717273747576777879
7A7B7C7D7E7F80818283848586878889 8A8B8C8D8E8F909192939495969798999A9B9C9D9E9FA
0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3
C4C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5
E6E7E8E9EAEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF000102030405060708090A
0B0C0D0E0F10111213141516171819 1A1B1C1D1E1F202122232425262728292A2B2C2D2E2F303
132333435363738393A3B3C3D3E3F404142434445464748494A4B4C4D4E4F50515253545556575
8595A5B5C5D5E5F606162636465666768696A6B6C6D6E6F707172737475767778797A7B7C7D7E
7F80818283848586878889 8A8B8C8D8E8F909192939495969798999A9B9C9D9E9FA0A1A2A3A4A
5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3C4C5C6C7C
8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9EA
EBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF9000h

Appendix D. Escape Command Example

Example: Get Firmware Version (using PCSCDirectCommand.exe).

Step 1: Plug in the ACM1281 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: “3500”

Enter Data: “E0 00 00 18 00” (APDU for Get Firmware Version)

Click enter to send to reader, then check the Response.

Appendix E. ACR128 Compatibility

Below is the list of ACR128 functions that are implemented differently or not supported by ACM1281U-C7.

Functions	ACR128	ACM1281U-C7
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

Functions	ACR128	ACM1281U-C7
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