



Advanced Card Systems Ltd.
Card & Reader Technologies

ACR1581U

USB Dual Interface Reader



Reference Manual V1.06



Table of Contents

1.0.	Introduction	5
2.0.	Features	6
3.0.	ACR1581U Architecture	7
3.1.	Reader Block Diagram	7
3.2.	Communication between PC/SC driver and ICC, PICC and SAM	7
4.0.	Hardware Design	8
4.1.	USB	8
4.1.1.	Communication Parameters	8
4.1.2.	Endpoints	8
4.2.	Contact Smart Card Interface	8
4.2.1.	Smart Card Power Supply VCC (C1)	8
4.2.2.	Card Type Selection	8
4.2.3.	Interface for Microcontroller-based Cards	8
4.3.	Contactless Smart Card Interface	9
4.3.1.	Carrier Frequency	9
4.3.2.	Card Polling	9
4.4.	User Interface	9
4.4.1.	Buzzer and LED	9
5.0.	Software Design	10
5.1.	PCSC API	10
5.1.1.	SCardEstablishContext	10
5.1.2.	SCardListReaders	11
5.1.3.	SCardConnect	12
5.1.4.	SCardControl	13
5.1.5.	SCardTransmit	15
5.1.6.	SCardDisconnect	17
5.1.7.	APDU Flow	18
5.1.8.	Escape Command Flow	19
5.2.	Contact Smart Card Protocol	20
5.2.1.	Memory Card – 1/2/4/8/16 kb I2C Card	20
5.2.2.	Memory Card – 32/64/128/256/512/1024 kb I2C Card	22
5.2.3.	Memory Card – ATMEL AT88SC153	24
5.2.4.	Memory Card – ATMEL AT88SC1608	28
5.2.5.	Memory Card – SLE4418/SLE4428/SLE5518/SLE5528	32
5.2.6.	Memory Card – SLE4432/SLE4442/SLE5532/SLE5542	37
5.2.7.	Memory Card – SLE4406/SLE4436/SLE5536/SLE6636	41
5.2.8.	Memory Card – SLE4404	46
5.2.9.	Memory Card – AT88SC101/AT88SC102/AT88SC1003	50
5.2.10.	ACOS6-SAM Commands	56
5.3.	Contactless Smart Card Protocol	68
5.3.1.	ATR Generation	68
5.3.2.	APDU, Pseudo APDU and Card Native Command	70
5.3.3.	PCSC Pseudo APDU (with Proprietary Extension) for PICC	70
5.3.4.	Pass Through Command	78
5.3.5.	APDU Commands for PCSC 2.0 Part 3 (Version 2.02 or above)	83
5.3.6.	Proprietary Pseudo APDU for PICC	93
5.3.7.	Accessing PCSC-Compliant tags (ISO14443-4)	95
5.3.9.	Supported PICC ATR	97



6.0.	Escape Command	100
6.1.	Escape Command for PICC	100
6.1.1.	RF Control [E0 00 00 25 01 ...].....	100
6.1.2.	Get PCD/PICC Status [E0 00 00 25 00].....	100
6.1.3.	Get Polling/ATR Option [E0 00 00 23 00]	101
6.1.4.	Set Polling/ATR Option [E0 00 00 23 01 ...]	101
6.1.5.	Get PICC Polling Type [E0 00 01 20 00].....	102
6.1.6.	Set PICC Polling Type [E0 00 01 20 02 ...]	102
6.1.7.	Get Auto PPS [E0 00 00 24 00].....	103
6.1.8.	Set Auto PPS [E0 00 00 24 01 ...]	103
6.1.9.	Read PICC Type [E0 00 00 35 00].....	104
6.1.10.	Get RF Power Setting [E0 00 00 50 00]	105
6.1.11.	Set RF Power Setting [E0 00 00 50 01 ...].....	105
6.1.12.	Get Selective Suspend Setting [E0 00 00 E5 00]	106
6.1.13.	Set Selective Suspend Setting [E0 00 00 E5 01 ...].....	106
6.1.14.	Escape Command for PICC – HID Keyboard.....	107
6.1.15.	Escape Command for PICC – Card Emulation	113
6.1.16.	Escape Command for PICC – Discovery Mode	119
6.2.	Escape Command for ICC	120
6.2.1.	Get Exclusive Mode [E0 00 00 2B 00]	120
6.2.2.	Set Exclusive Mode [E0 00 00 2B 01 ...]	120
6.2.3.	Get Card Power Config [E0 00 00 0B 00]	121
6.2.4.	Set Card Power Config [E0 00 00 0B 01 ...].....	121
6.3.	Escape Command for Peripheral Control and Other	122
6.3.1.	Get Firmware Version [E0 00 00 18 ...]	122
6.3.2.	Get Serial Number [E0 00 00 33 00].....	122
6.3.3.	Set S/N in USB Descriptor [E0 00 00 F0].....	122
6.3.4.	Set Buzzer Control - Single Time [E0 00 00 28 01 ...].....	123
6.3.5.	Set Buzzer Control - Repeatable [E0 00 00 28 03 ...]	123
6.3.6.	Get LED Status [E0 00 00 29 00].....	124
6.3.7.	Set LED Control [E0 00 00 29 01 ...]	124
6.3.8.	Get UI Behaviour [E0 00 00 21 00]	124
6.3.9.	Set UI Behaviour [E0 00 00 21 01 ...].....	125
Appendix A.	NDEF Message	126
Appendix B.	ACR1281U Compatibility	127
Appendix B.1.	Load Authentication Keys	127
Appendix B.2.	Authentication for MIFARE Classic (1K/4K)	128
Appendix B.3.	Manual PICC Polling	129
Appendix B.4.	Read PICC Operating Parameter	129
Appendix B.5.	Set PICC Operating Parameter	130
Appendix B.6.	Initialize Card Insertion Counter	130
Appendix B.7.	Read Card Insertion Counter	131
Appendix B.8.	Update Card Insertion Counter	131
Appendix B.9.	Compatibility and Migration Notes	132

List of Figures

Figure 1 :	ACR1581U Reader Block Diagram	7
Figure 2 :	ACR1581U Architecture	7



Figure 3 : ACR1581U APDU Flow 18
Figure 4 : ACR1581U Escape Command Flow..... 19
Figure 5 : ACR1581U Transparent Session Flow 83

List of Tables

Table 1 : USB Interface Wiring 8
Table 2 : Buzzer and LED Indicator 9
Table 3 : Blown Fuse Code Values 55
Table 4 : MIFARE Classic 1K Memory Map 73
Table 5 : MIFARE Classic 4K Memory Map 73
Table 6 : MIFARE Ultralight Memory Map 74
Table 7 : NFC Forum Type 2 Tag Memory Map (2000 bytes) 113
Table 8 : FeliCa Memory Map (160 bytes) 114



1.0. Introduction

Continuing the success of the ACR1281U is the ACR1581 DualBoost III, third generation product in the ACS's DualBoost Reader Series. ACR1581 is a Dual Interface Reader that is able to access any contact and contactless smart cards alike, and makes use of the USB CCID Class Driver and USB Interface to establish connection with PCs to accept card commands from the computer's application. Not only does it support traditional ISO 7816 MCU Cards, MIFARE® Cards and ISO 14443 Type A and B Contactless Cards, it also is able to implement additional support for FeliCa and ISO 15693 Cards.

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

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



2.0. Features

- USB Full Speed Interface
- CCID-compliant
- Smart Card Reader:
 - Contactless Interface:
 - Read/Write speed of up to 26kbps ISO 15693 & 848 kbps (ISO 14443) card types
 - Built-in antenna for contactless tag access, with card reading distance of up to 70 mm (depending on tag type)
 - Supports ISO 15693 card types
 - Supports ISO 14443 Part 4 Type A and B cards and MIFARE series
 - Built-in anti-collision feature
 - Supports extended APDU (max. 64 KB)
 - Contact Interface:
 - Supports ISO 7816 Class A, B and C (5 V, 3 V and 1.8 V)
 - Supports CAC (Common Access Card)
 - Supports PIV (Personal Identity Verification Card)
 - Supports microprocessor cards with T=0 or T=1 protocol
 - Supports PPS (Protocol and Parameters Selection)
 - Features Short Circuit Protection
 - Supports extended APDU (max. 64 Kbytes for T=1; max. 512+10 Bytes for T=0)
 - SAM Interface:
 - One SAM Slot
 - Supports ISO 7816 Class A SAM cards
- Application Programming Interface:
 - Supports PC/SC
 - Supports CT-API (through wrapper on top of PC/SC)
- Built-in Peripherals:
 - Two user-controllable LEDs (Blue and Green)
 - User-controllable buzzer
- USB Firmware Upgradability
- Supports Android™ 3.1 and later¹
- Compliant with the following standards:
 - ISO 14443
 - ISO 15693
 - ISO 7816
 - PC/SC
 - CCID
 - CE
 - UKCA
 - FCC
 - RoHS
 - REACH
 - Microsoft® WHQL

¹ Uses an ACS-defined Android Library

3.0. ACR1581U Architecture

3.1. Reader Block Diagram

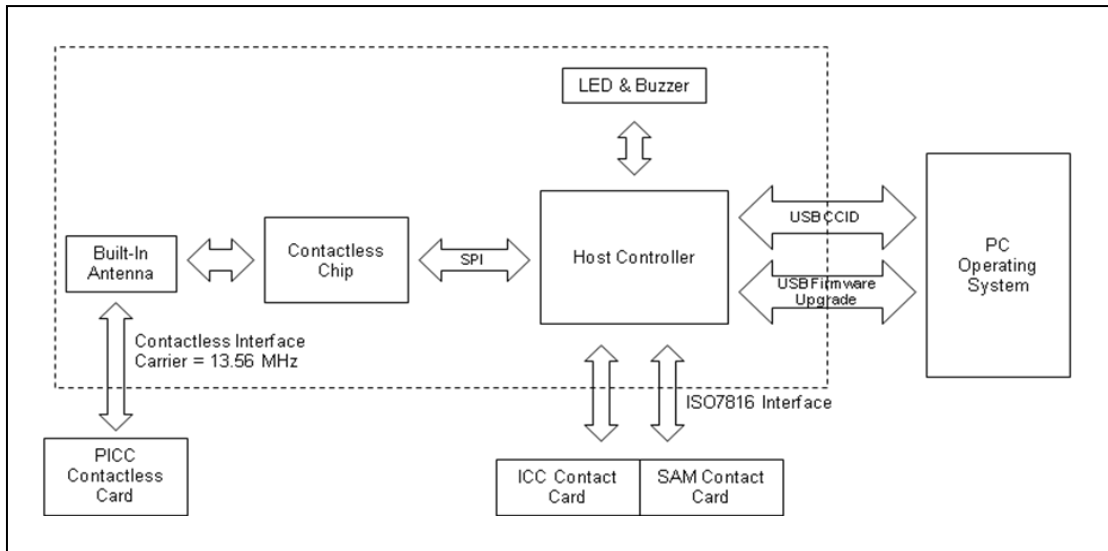


Figure 1: ACR1581U Reader Block Diagram

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

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

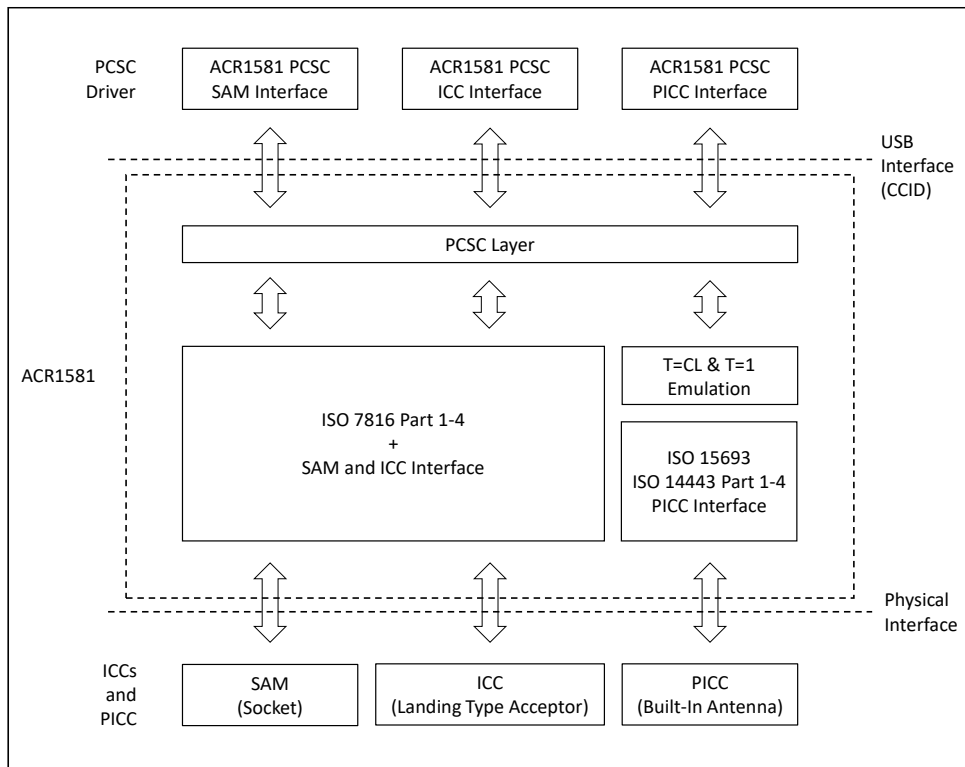


Figure 2: ACR1581U Architecture

4.0. Hardware Design

4.1. USB

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

4.1.1. Communication Parameters

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

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

Table 1: USB Interface Wiring

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

4.1.2. Endpoints

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

Control Endpoint – For setup and control purposes.

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

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

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

4.2. Contact Smart Card Interface

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

4.2.1. Smart Card Power Supply VCC (C1)

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

4.2.2. Card Type Selection

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

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

4.2.3. Interface for Microcontroller-based Cards

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



4.3. Contactless Smart Card Interface

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

4.3.1. Carrier Frequency

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

4.3.2. Card Polling

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

4.4. User Interface

4.4.1. Buzzer and LED

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

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

Table 2: Buzzer and LED Indicator



5.0. Software Design

5.1. PCSC API

This section will describe some of the PCSC API for application programming usage. For more details, please refer to Microsoft MSDN Library or PCSC workgroup.

5.1.1. SCardEstablishContext

The SCardEstablishContext function establishes the resource manager context within which database operations are performed.

Refer to: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa379479%28v=vs.85%29.aspx>

This function should be performed first before any other PCSC operation.

Example:

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT hContext;
int retCode;
void main ()
{
    // To establish the resource manager context and assign it to "hContext"
    retCode = SCardEstablishContext(SCARD_SCOPE_USER,
                                   NULL,
                                   NULL,
                                   &hContext);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Establishing resource manager context failed
    }
    else
    {
        // Establishing resource manager context successful
        // Further PCSC operation can be performed
    }
}
```



5.1.2. SCardListReaders

The SCardListReaders function provides the list of readers within a set of named reader groups, eliminating duplicates.

The caller supplies a list of reader groups, and receives the list of readers within the named groups. Unrecognized group names are ignored. This function only returns readers within the named groups that are currently attached to the system and available for use.

Refer to: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa379793%28v=vs.85%29.aspx>

Example:

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT hContext; // Resource manager context
int retCode;
char readerName [256]; // List reader name

void main ()
{
    // To establish the resource manager context and assign to "hContext"
    retCode = SCardEstablishContext(SCARD_SCOPE_USER,
                                   NULL,
                                   NULL,
                                   &hContext);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Establishing resource manager context failed
    }
    else
    {
        // Establishing resource manager context successful
        // List the available reader which can be used in the system
        retCode = SCardListReaders (hContext,
                                    NULL,
                                    readerName,
                                    &size);
        if (retCode != SCARD_S_SUCCESS)
        {
            // Listing reader fail
        }
        if (readerName == NULL)
        {
            // No reader available
        }
        else
        {
            // Reader listed
        }
    }
}
}
```



5.1.3. SCardConnect

The SCardConnect function establishes a connection (using a specific resource manager context) between the calling application and a smart card contained by a specific reader. If no card exists in the specified reader, an error is returned.

Refer to: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa379473%28v=vs.85%29.aspx>

Example:

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT      hContext;          // Resource manager context
SCARDHANDLE       hCard;            // Card context handle
unsigned long     dwActProtocol;     // Establish active protocol
int               retCode;
char              readerName [256]; // List reader name
char              rName [256];      // Reader name for connection

void main ()
{
    ...
    if (readerName == NULL)
    {
        // No reader available
    }
    else
    {
        // Reader listed
        rName = "ACS ACR1581 1S Dual Reader PICC 0"; // Depends on what
                                                    // reader be used
                                                    // Should connect to
                                                    // PICC interface

        retCode = SCardConnect(hContext,
                                rName,
                                SCARD_SHARE_SHARED,
                                SCARD_PROTOCOL_T0,
                                &hCard,
                                &dwActProtocol);
        if (retCode != SCARD_S_SUCCESS)
        {
            // Connection failed (May be because of incorrect reader
            // name, or no card was detected)
        }
        else
        {
            // Connection successful
        }
    }
}
```



5.1.4. SCardControl

The SCardControl function gives you direct control of the reader. You can call it any time after a successful call to SCardConnect and before a successful call to SCardDisconnect. The effect on the state of the reader depends on the control code.

Refer to: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa379474%28v=vs.85%29.aspx>

Note: Commands from **Escape Command** use this API for sending.

Example:

```
#define SCARD_SCOPE_USER 0

#define EscapeCommand 0x310000 + 3500*4
SCARDCONTEXT hContext; // Resource manager context
SCARDHANDLE hCard; // Card context handle
unsigned long dwActProtocol; // Established active protocol
int retCode;
char readerName [256]; // Lists reader name
char rName [256]; // Reader name for connection
BYTE SendBuff[262], // APDU command buffer
RecvBuff[262]; // APDU response buffer
BYTE FWVersion [20], // For storing firmware
version message
BYTE ResponseData[50]; // For storing card response
DWORD SendLen, // APDU command length
RecvLen; // APDU response length

void main ()
{
    ...
    rName = "ACS ACR1581 1S Dual Reader PICC 0"; // Depends on what
reader will be used
// Should connect to
PICC interface

    retCode = SCardConnect(hContext,
        rName,
        SCARD_SHARE_DIRECT,
        SCARD_PROTOCOL_T0| SCARD_PROTOCOL_T1,
        &hCard,
        &dwActProtocol);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Connection failed (may be because of incorrect reader
name, or no card was detected)
    }
    else
    {
        // Connection successful
        RecvLen = 262;
        // Get firmware version
        SendBuff[0] = 0xE0;
        SendBuff[1] = 0x00;
        SendBuff[2] = 0x00;
        SendBuff[3] = 0x18;
        SendBuff[4] = 0x00;
    }
}
```



```
SendLen = 5;
retCode = SCardControl ( hCard,
    EscapeCommand,
    SendBuff,
    SendLen,
    RecvBuff,
    RecvLen,
    &RecvLen);
if (retCode != SCARD_S_SUCCESS)
{
    // APDU sending failed
    return;
}
else
{
    // APDU sending successful
    // The RecvBuff stores the firmware version message.
    for (int i=0;i< RecvLen-5;i++)
    {
        FWVersion[i] = RecvBuff [5+i];
    }
}
// Connection successful
RecvLen = 262;

// Turn Green LED on, turn Red LED off
SendBuff[0] = 0xE0;
SendBuff[1] = 0x00;
SendBuff[2] = 0x00;
SendBuff[3] = 0x29;
SendBuff[4] = 0x01;
SendBuff[5] = 0x02; // Green LED On, Red LED off
SendLen = 6;
retCode = SCardControl ( hCard,
    EscapeCommand,
    SendBuff,
    SendLen,
    RecvBuff,
    RecvLen,
    &RecvLen);
if (retCode != SCARD_S_SUCCESS)
{
    // APDU sending failed
    return;
}
else
{
    // APDU sending success
}
```



5.1.5. SCardTransmit

The SCardTransmit function sends a service request to the smart card and expects to receive data back from the card.

Refer to: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa379804%28v=vs.85%29.aspx>

Note: APDU Commands (i.e. the commands sent to connected card, PCSC Pseudo APDU (with Proprietary Extension) for PICC, and PCSC 2.0 Part 3 Example

1. Start Transparent Session.
Command: **FF C2 00 00 02 81 00**
Response: **C0 03 00 90 00 90 00**
2. Turn the Antenna Field off.
Command: **FF C2 00 00 02 83 00**
Response: **C0 03 00 90 00 90 00**
3. Turn the Antenna Field on.
Command: **FF C2 00 00 02 84 00**
Response: **C0 03 00 90 00 90 00**
4. ISO 14443-4A Active.
Command: **FF C2 00 02 04 8F 02 00 04**
Response: **C0 03 01 64 01 90 00** (if no card present)
C0 03 00 90 00 5F 51 [Len] [ATR] 90 00
5. Set the PCB to 0Ah and enable the CRC, parity and protocol prologue in the transmit data.
Command: **FF C2 00 01 0A 90 02 00 00 FF 6E 03 07 01 0A**
Response: **C0 03 00 90 00 90 00**
6. Send the APDU "80B2000008" to card and get response.
Command: **FF C2 00 01 0E 5F 46 04 40 42 0F 00 95 05 80 B2 00 00 08**
Response: **C0 03 00 90 00 92 01 00 96 02 00 00 97 0C [Card Response] 90 00**
7. End Transparent Session.
Command: **FF C2 00 00 02 82 00**
Response: **C0 03 00 90 00 90 00**

Proprietary Pseudo APDU for PICC) use this API for sending.



Example:

```
#define SCARD_SCOPE_USER      0

SCARDCONTEXT      hContext;          // Resource manager context
SCARDHANDLE       hCard;             // Card context handle
unsigned long     dwActProtocol;     // Established active protocol
int               retCode;
char              readerName [256]; // List reader name
char              rName [256];      // Reader name for connect
BYTE              SendBuff[262];    // APDU command buffer
BYTE              RecvBuff[262];    // APDU response buffer
BYTE              CardID [8],       // For storing the FeliCa IDM/
                                   MIFARE UID
BYTE              ResponseData[50]; // For storing card response
DWORD            SendLen,           // APDU command length
DWORD            RecvLen;           // APDU response length
SCARD_IO_REQUEST  ioRequest;

void main ()
{
    ...
    rName = "ACS ACR1581 1S Dual Reader PICC 0"; // Depends on what
                                                  // reader should be used
                                                  // Should connect to PICC
                                                  // interface

    retCode = SCardConnect (hContext,
                            rName,
                            SCARD_SHARE_SHARED,
                            SCARD_PROTOCOL_T0,
                            &hCard,
                            &dwActProtocol);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Connection failed (May be because of incorrect reader
        // name, or no card was detected)
    }
    else
    {
        // Connection successful
        ioRequest.dwProtocol = dwActProtocol;
        ioRequest.cbPciLength = sizeof(SCARD_IO_REQUEST);
        RecvLen = 262;
    }
}
```



```
// Get MIFARE UID/ FeliCa IDM
SendBuff[0] = 0xFF;
SendBuff[1] = 0xCA;
SendBuff[2] = 0x00;
SendBuff[3] = 0x00;
SendBuff[4] = 0x00;
SendLen = 5;
retCode = SCardTransmit( hCard,
                        &ioRequest,
                        SendBuff,
                        SendLen,
                        NULL,
                        RecvBuff,
                        &RecvLen);

if (retCode != SCARD_S_SUCCESS)
{
    // APDU sending failed
    return;
}
else
{
    // APDU sending successful
    // The RecvBuff stores the IDM for FeliCa / the UID for
    MIFARE.
    // Copy the content for further FeliCa access
    for (int i=0;i< RecvLen-2;i++)
    {
        CardID [i] = RecvBuff[i];
    }
}
```



5.1.6. SCardDisconnect

The **SCardDisconnect** function terminates a connection previously opened between the calling application and a *smart card* in the target *reader*.

Refer to: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa379475%28v=vs.85%29.aspx>

This function is used to end the PCSC Operation.

Example:

```
#define SCARD_SCOPE_USER 0

SCARDCONTEXT      hContext;          // Resource manager context
SCARDHANDLE        hCard;            // Card context handle
unsigned long      dwActProtocol;    // Established active protocol
int                retCode;

void main ()
{
    ...
    // Connection successful
    ...
    retCode = SCardDisconnect(hCard, SCARD_RESET_CARD);
    if (retCode != SCARD_S_SUCCESS)
    {
        // Disconnection failed
    }
    else
    {
        // Disconnection successful
    }
}
}
```

5.1.7. APDU Flow

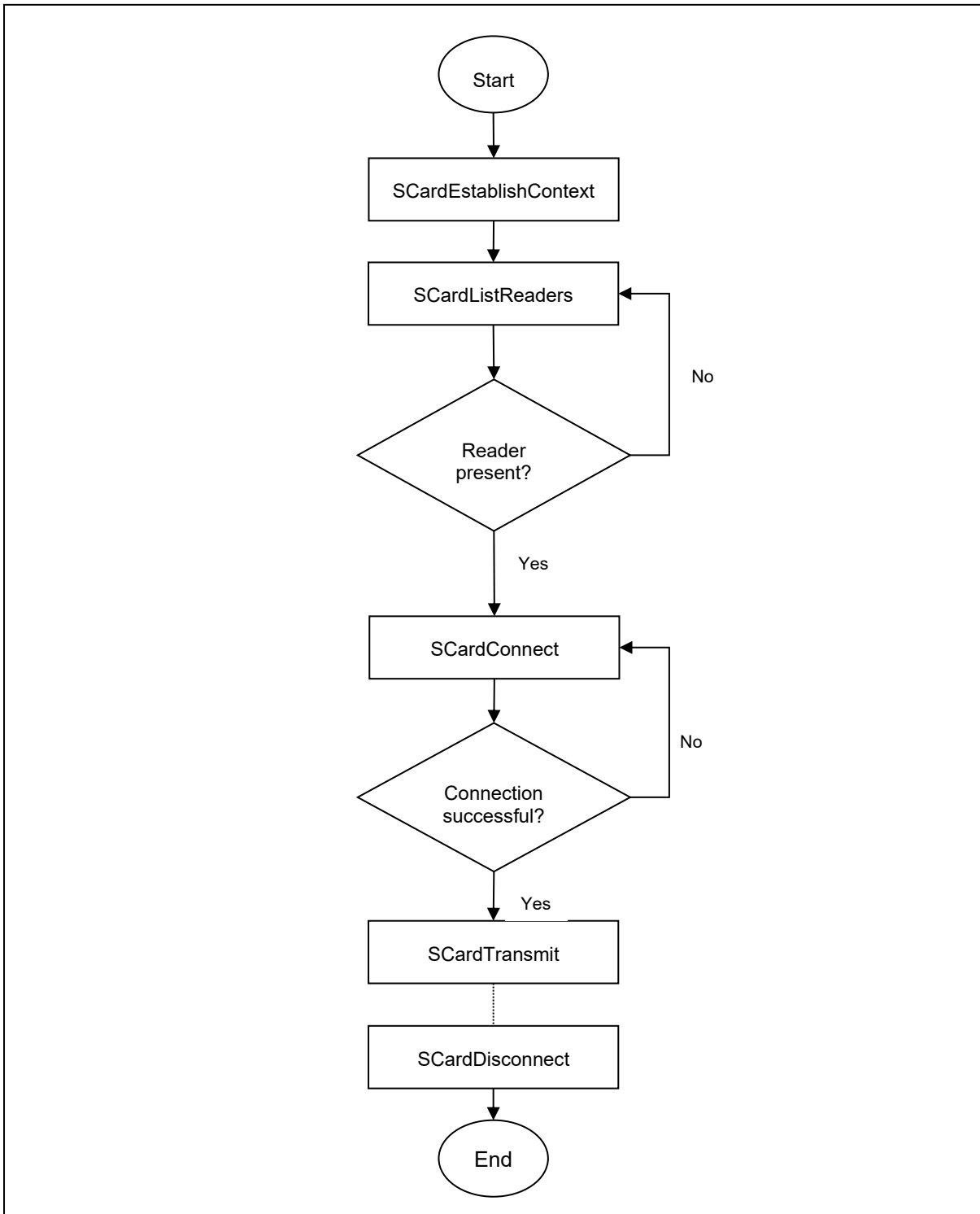


Figure 3: ACR1581U APDU Flow

5.1.8. Escape Command Flow

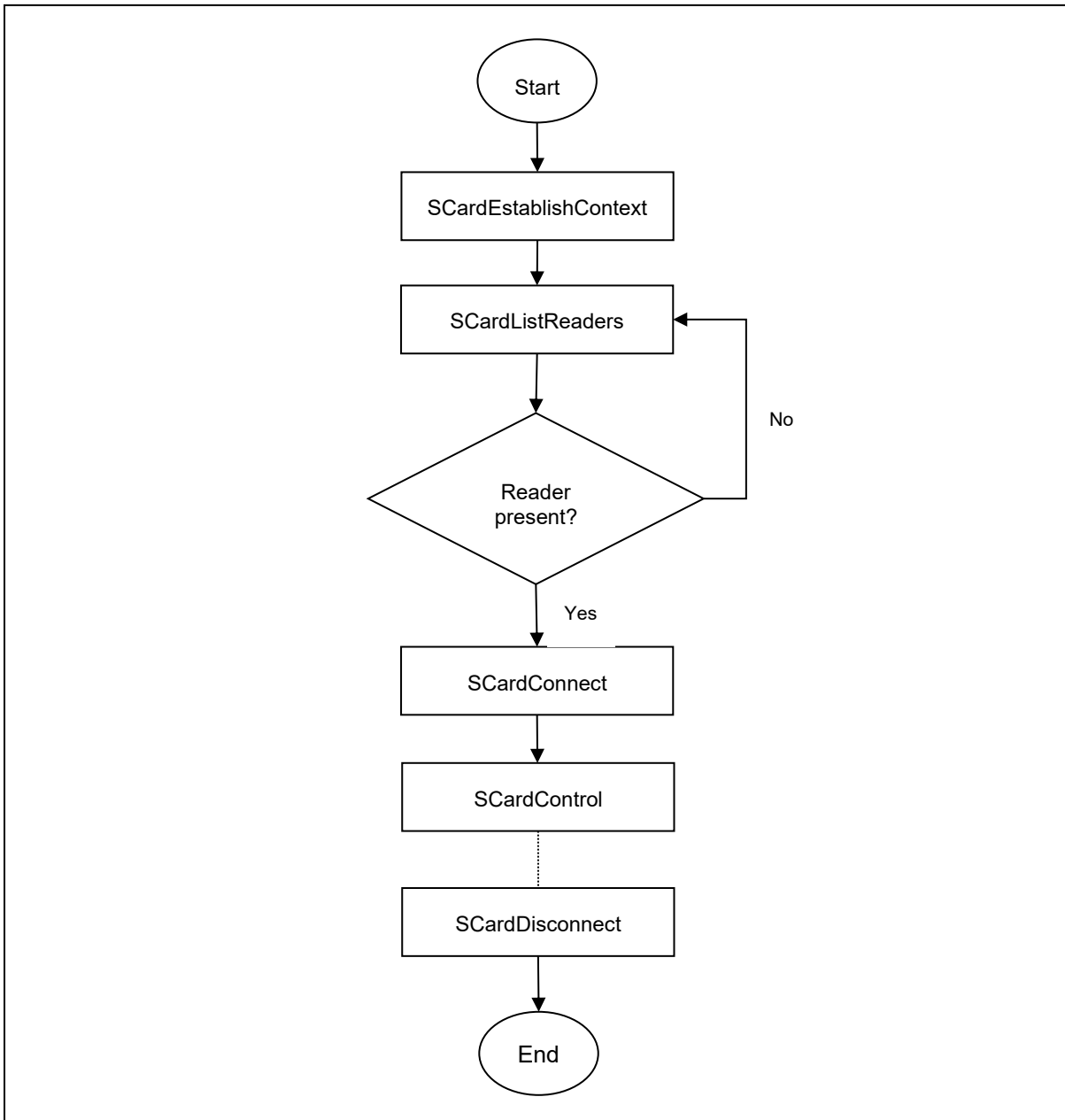


Figure 4: ACR1581U Escape Command Flow



5.2. Contact Smart Card Protocol

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

5.2.1.1. Select Card Type

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

Command

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

Response Codes

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

5.2.1.2. Select Page Size

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

Command

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

Response Codes

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

Page Size: 1 Byte

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



5.2.1.3. Read Memory Card

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

Command

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

Response Codes

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

5.2.1.4. Write Memory Card

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

Command

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

Response Codes

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



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

5.2.2.1. Select Card Type

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

Command

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

Response Codes

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

5.2.2.2. Select Page Size

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

Command

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

Response Codes

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

Page Size: 1 Byte

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



5.2.2.3. Read Memory Card

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

Command

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

Response Codes

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

Note: *INS* B0h = For 32, 64, 128, 256, 512 kb I2C card
1011 000*b; where * is the MSB of the 17 bit addressing = For 1024 kb I2C card

5.2.2.4. Write Memory Card

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

Command

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

Response Codes

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

Note: *INS* B0h = For 32, 64, 128, 256, 512 kb I2C card
1011 000*b; where * is the MSB of the 17 bit addressing = For 1024 kb I2C card



5.2.3. Memory Card – ATMEL AT88SC153

5.2.3.1. Select card type

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

Command

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

5.2.3.2. Read memory card

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

Command

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

Where:

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

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

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

Response

Response	Byte 1	Byte N	SW1	SW2
Result						

Where:

Byte (1...N) Data read from memory card.
SW1 SW2 = 90 00h if the operation was completed successfully.



5.2.3.3. Write memory card

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

Command

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

Where:

- INS** (1 byte)
For reading zone 00b, INS = D0h
For reading zone 01b, INS = D1h
For reading zone 10b, INS = D2h
For reading zone 11b, INS = D3h
For reading fuse, INS = D4h
- Byte Address** (1 byte)
Memory address location of the memory card.
- MEM_L** (1 byte)
Length of data to be written to the memory card
- Byte (1...N)** Data to be written to the memory card.

Response

Response	Data Out	
Result	SW1	SW2

Where:

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



5.2.3.4. Verify password

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

Command

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

Where:

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

P2 (1 Byte)
= 0000 00r pb

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

r = 0: Write password,

r = 1: Read password,

p = Password set number

r p = 01b for the secure code.

Response

Response	Data Out	
Result	SW1	ErrorCnt

Where:

SW1 = 90h

ErrorCnt (1 byte)
= Error Counter

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

5.2.3.5. Initialize authentication

This command initializes the memory card's authentication.

Command

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

Where:

Q (0...7) (8 bytes)
= Host random number

Response

Response	Data Out
Result	SW1 SW2

Where:

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

5.2.3.6. Verify authentication

This command verifies the memory card's authentication.

Command

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

Where:

Ch (0...7) (8 bytes)
= Host challenge

Response

Response	Data Out
Result	SW1 SW2

Where:

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



5.2.4. Memory Card – ATMEL AT88SC1608

5.2.4.1. Select card type

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

Command

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

Response

Response	Data Out
Result	SW1 SW2

Where:

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

5.2.4.2. Read memory card

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

Command

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

Where:

INS (1 byte)

For reading user zone, INS = B0h

For reading configuration zone or reading fuse, INS = B1h

Zone Address (1 byte)

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

** don't care for reading fuse

Byte Address (1 byte)

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

For reading fuse, Byte Address = 1000 0000b

MEM_L (1 byte)

Length of data to be read from the memory card.

Response

Response	Byte 1	Byte N	SW1	SW2
Result						

Where:

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

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



5.2.4.3. Write to memory card

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

Command

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

Where:

- INS** (1 byte)
For reading user zone, **INS = D0h**
For reading configuration zone or reading fuse, **INS = D1h**
- Zone Address** (1 byte)
= 0000 A10 A9 A8b, where A10 is the MSB of zone address
** don't care for reading fuse
- Byte Address** (1 byte)
= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card
For reading fuse, Byte Address = 1000 0000b
- MEM_L** (1 byte)
Length of data to be written to the memory card
- Byte (1...N)** Data to be written to the memory card.

Response

Response	Data Out	
Result	SW1	SW2

Where:

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



5.2.4.4. Verify password

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

Command

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

Where:

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

RP (1 byte)
= 0000 r p2 p1 p0b

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

r = 0 : Write password,

r = 1: Read password,

p2 p1 p0 = Password set number

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

Response

Response	Data Out	
Result	SW1	ErrorCnt

Where:

SW1 = 90h

ErrorCnt (1 byte)

= Error Counter

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



5.2.4.5. Initialize authentication

This command initializes the memory card's authentication.

Command

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

Where:

Q (0...7) (8 bytes)
= Host random number

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

5.2.4.6. Verify authentication

This command verifies the memory card's authentication.

Command

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

Where:

Ch (0...7) (8 bytes)
= Host challenge

Response

Response	Data Out	
Result	SW1	SW2

Where:

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



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

5.2.5.1. Select card type

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

Command

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

5.2.5.2. Read memory card

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

Command

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

Where:

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

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

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

Response

Response	Byte 1	Byte N	SW1	SW2
Result						

Where:

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

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



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

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

Command

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

Response

Response	ErrCnt	Dummy 1	Dummy 2	SW1	SW2
Result					

Where:

- ErrCnt** (1 byte)
The value of the presentation error counter
FFh = indicates the verification is correct
00h = indicates the password is locked (exceeding the maximum number of retries)
Other values indicate the verification failed.
- Dummy 1, Dummy 2** (2 bytes)
Dummy data read from the card
- SW1 SW2** = 90 00h if the operation was completed successfully.



5.2.5.4. Read protection bit

This command reads the protection bit.

Command

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

Where:

MSB Byte Address (1 byte)

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

LSB Byte Address (1 byte)

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

MEM_L (1 byte)

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

$$\text{MEM_L} = 1 + \text{INT}((\text{number of bits} - 1)/8)$$

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

FF B1 00 10 01h

Response

Response	PROT 1	PROT L	SW1	SW2
Result						

Where:

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

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

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

PROT 1								PROT 2																
P 8	P 7	P 6	P 5	P 4	P 3	P 2	P 1	P1 6	P1 5	P1 4	P1 3	P1 2	P1 1	P1 0	P 9	P1 8	P1 7

Where:

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

0 = byte is write protected

1 = byte can be written



5.2.5.5. Write memory card

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

Command

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

Where:

MSB Byte Address (1 byte)

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

LSB Byte Address (1 byte)

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

MEM_L (1 byte)

Length of data to be written to the memory card

Byte (1...N)

Data to be written to the memory card.

5.2.5.6. Write protection memory card

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

Command

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

Where:

MSB Byte Address (1 byte)

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

LSB Byte Address (1 byte)

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

MEM_L (1 byte)

=Length of data to be written to the memory card

Byte (1...N)

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

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



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

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

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

Command

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

Where:

Code (3 bytes)
Secret code (PIN)

Response

Response	Data Out	
Result	90h	ErrorCnt

Where:

ErrorCnt (1 byte)
Error Counter
FFh = indicates the verification is correct.
00h = indicates the password is locked (exceeding maximum number of retries).
Other values indicate the verification failed.



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

5.2.6.1. Select card type

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

Command

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

Response

Response	Data Out
Result	SW1 SW2

Where:

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

5.2.6.2. Read memory card

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

Command

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

Where:

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

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

Response

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

Where:

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

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

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

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

PROT 1								PROT 2																
P ₈	P ₇	P ₆	P ₅	P ₄	P ₃	P ₂	P ₁	P ₁ ₆	P ₁ ₅	P ₁ ₄	P ₁ ₃	P ₁ ₂	P ₁ ₁	P ₁ ₀	P ₉	P ₁ ₈	P ₁ ₇

Where:

P_x is the protection bit of byte x in response data:

0 = byte is write protected

1 = byte can be written



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

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

Command

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

Response

Response	ErrCnt	Dummy 1	Dummy 2	Dummy 3	SW1	SW2
Result						

Where:

- ErrCnt** (1 byte)
The value of the presentation error counter
07h = indicates the verification is correct.
00h = indicates the password is locked (exceeded the maximum number of retries).
Other values indicate the verification failed.
- Dummy 1, Dummy 2, Dummy 3** (3 bytes)
Dummy data read from the card
- SW1 SW2** = 90 00h if the operation was completed successfully.

5.2.6.4. Read Protection Bit

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

Command

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

Response

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

Where:

- PROT (1..4)** Bytes containing the protection bits.
- SW1 SW2** = 90 00h if the operation was completed successfully.

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

PROT 1								PROT 2																
P	P	P	P	P	P	P	P	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1
8	7	6	5	4	3	2	1	6	5	4	3	2	1	0	9	8	7

Where:

- Px** protection bit of bytes in the response data:
0 = byte is write protected
1 = byte can be written



5.2.6.5. Write memory card

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

Command

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

Where:

- Byte Address** (1 byte)
= A7 A6 A5 A4 A3 A2 A1 A0b is the memory address location of the memory card
- MEM_L** (1 byte)
Length of data to be written to the memory card
- Byte (1...N)** Data to be written to the memory card.

Response

Response	Data Out	
Result	SW1	SW2

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

5.2.6.6. Write protection memory card

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

Command

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

Where:

- Byte Address** (1 byte)
= 000A4 A3 A2 A1b (00h – 1Fh) is the protection memory address location of the memory card
- MEM_L** (1 byte)
Length of data to be written to the memory card
- Byte (1...N)** Byte values compared with the data in the card starting at the Byte Address. Byte 1 is compared with the data at Byte Address; Byte N is compared with the data at Byte Address + N – 1.

Response

Response	Data Out	
Result	SW1	SW2

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



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

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

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

Command

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

Where:

Code (3 bytes)
Secret code (PIN)

Response

Response	Data Out
Result	SW1 ErrorCnt

Where:

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

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

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

Command

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

Where:

Code (3 bytes)
Secret code (PIN)

Response

Response	Data Out
Result	SW1 SW2

Where:

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



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

5.2.7.1. Select card type

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

Command

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

5.2.7.2. Read Memory Card

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

Command

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

Where:

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

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

Response

Response	Byte 1	Byte N	SW1	SW2
Result						

Where:

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

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

5.2.7.3. Write one byte memory card

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

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

a. Write

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

b. Write with carry

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

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

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

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

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

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

The backup mode available in the SLE4436 and SLE5536 card can be enabled or disabled in the write operation.

Command

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

Where:

Byte Address	(1 byte) Memory address location of the memory card
Mode	(1 byte) Specifies the write mode and backup option 00h = Write. 01h = Write with carry. 02h = Write with backup enabled (for SLE4436, SLE5536 and SLE6636 only). 03h = Write with carry and with backup enabled (for SLE4436, SLE5536 and SLE6636 only).
Byte	(1 byte) Byte value to be written to the card

Response

Response	Data Out	
Result	SW1	SW2

Where:

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



5.2.7.4. Present code memory card

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

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

Command

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

Where:

- Addr** (1 byte)
Byte address of the presentation counter in the card
- Code** (3 bytes)
Secret code (PIN)

Response

Response	Data Out	
Result	SW1	SW2

Where:

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



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

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

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

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

Step 1: Send authentication certificate to the card.

Command

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

Where:

- Key** (1 byte)
Key to be used for the computation of the authentication certificate
00h = Key 1 with no cipher block chaining.
01h = Key 2 with no cipher block chaining.
80h = Key 1 with cipher block chaining (for SLL5536 and SLE6636 only).
81h = Key 2 with cipher block chaining (for SLL5536 and SLE6636 only).
- CLK_CNT** (1 byte)
Number of CLK pulses to be supplied to the card for the computation of each bit of the authentication certificate. Typical value is 160 clocks (A0h).
- Byte (1...6)** Card challenge data.

Response

Response	SW1	SW2
Result	61h	02h



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

Command

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

Response

Response	Cert	SW1	SW2
Result			

Where:

- Cert** (2 bytes)
16 bits of authentication data computed by the card. The LSB of Byte 1 is the first authentication bit read from the card.
- SW1 SW2** = 90 00h if the operation was completed successfully.



5.2.8. Memory Card – SLE4404

5.2.8.1. Select card type

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

Command

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

5.2.8.2. Read memory card

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

Command

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

Where:

Byte Address (1 byte)

Memory address location of the memory card

MEM_L (1 byte)

Length of data to be read from the memory card

Response

Response	Byte 1	Byte N	SW1	SW2
Result						

Where:

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

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



5.2.8.3. Write memory card

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

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

Command

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

Where:

- Byte Address** (1 byte)
Memory address location of the memory card
- MEM_L** (1 byte)
Length of data to be written to the memory card
- Byte (1...N)** Data to be written to the memory card.

Response

Response	Data Out
Result	SW1 SW2

Where:

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

5.2.8.4. Erase scratch pad memory card

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

Command

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

Where:

- Byte Address** (1 byte)
Memory byte address location of the scratch pad. Typical value is 02h.

Response

Response	Data Out
Result	SW1 SW2

Where:

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



5.2.8.5. Verify user code

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

The following actions are executed:

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

Command

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

Where:

- Error Counter LEN** (1 byte)
Length of presentation error counter in bits
- Byte Address** (1 byte)
Byte address of the key in the card
- Code** (1 byte)
User Code

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

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



5.2.8.6. Verify memory code

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

The following actions are executed:

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

Note: The Memory Error Counter cannot be erased.

Command

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

Where:

- Error Counter LEN** (1 byte)
Length of presentation error counter in bits
- Byte Address** (1 byte)
Byte address of the key in the card
- Code** (4 bytes)
Memory Code

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

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



5.2.9. Memory Card – AT88SC101/AT88SC102/AT88SC1003

5.2.9.1. Select card type

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

Command

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

5.2.9.2. Read Memory Card

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

Command

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

Where:

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

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

Response

Response	Byte 1	Byte N	SW1	SW2
Result						

Where:

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

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

5.2.9.3. Write Memory Card

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

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

Command

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

Where:

- Byte Address** (1 byte)
Memory address location of the memory card.
- MEM_L** (1 byte)
Length of data to be written to the memory card
- Byte (1...N)** Byte value to be written to the card.

Response

Response	Data Out
Result	SW1 SW2

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

5.2.9.4. Erase non-application zone

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

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

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

Command

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

Where:

- Byte Address** (1 byte)
Memory byte address location of the word to be erased.

Response

Response	Data Out
Result	SW1 SW2

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

5.2.9.5. Erase Application Zone with Erase

This command can be used in the following cases:

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

The following actions are executed for this command:

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

Command

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

Where:

- Error Counter LEN** (1 byte)
= Length of presentation error counter in bits. The value should be 00h always.
- Byte Address** (1 byte)
= Byte address of the Application Zone Key in the card. Please refer to the table below for the correct value.
- MEM_L** (1 byte)
= Length of the Erase Key. Please refer to the table below for the correct value.
- CODE (1...N)**
= Erase Key

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

Response

Response	Data Out	
Result	SW1	SW2

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

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

5.2.9.6. Erase Application Zone with Write and Erase

This command can be used in the following cases:

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

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

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

Command

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

Where:

- Error Counter LEN** (1 byte)
= Length of presentation error counter in bits. The value should be 80h always.
- Byte Address** (1 byte)
= Byte address of the Application Zone Key in the card. Please refer to the table below for the correct value.
- CODE** (4 bytes)
= Erase Key

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

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

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



5.2.9.7. Verify Security Code

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

The following actions are executed:

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

Command

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

Where:

- Error Counter LEN** (1 byte)
= Length of presentation error counter in bits.
- Byte Address** (1 byte)
= Byte address of the key in the card.
- CODE** (2 bytes)
= Security Code

Response

Response	Data Out	
Result	SW1	SW2

Where:

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

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



5.2.9.8. Blow Fuse

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

Note: Blowing the fuse is an irreversible process.

Command

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

Where:

Fuse Bit Addr (2 bytes)
= Bit address of the fuse. Please refer to the table below for the correct value.

State of FUS Pin (1 byte)
= State of the FUS pin. Should be 01h always.

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

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

Table 3: Blown Fuse Code Values

Response

Response	Data Out	
Result	SW1	SW2

Where:

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



5.2.10. ACOS6-SAM Commands

This section contains SAM-specific commands. CCID Host could send Card Native Command or APDU to the Reader by using SCardTransmit() in PCSC API.

Note: Please note that the ACOS6-SAM component is only included in the ACR1581U SDK package bundled with the reader. It is not part of the standard retail version of the reader. For complete information on ACOS6-SAM Commands and Scenarios, please contact an ACS representative for a copy of the ACOS6-SAM Reference Manual.

5.2.10.1. Generate Key

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

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

Specific Response Status Bytes

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



5.2.10.2. Diversify (or load) Key Data

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

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

Specific Response Status Bytes

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



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

5.2.10.3. Encrypt

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

1. The session key created by the mutual authentication procedure with an ACOS3/6, DESFire®, DESFire® EV1/EV2/Light or MIFARE Plus card.
2. A diversified key (secret code).
3. A bulk encryption key.
4. Encrypt the diversified secret code with the session key.
5. Prepare ACOS3 secure messaging command given a non-SM command.

APDU	Description																																																																																																																																																
CLA	80h																																																																																																																																																
INS	74h																																																																																																																																																
	<table border="1"> <thead> <tr> <th>b7</th> <th>b6</th> <th>b5</th> <th>b4</th> <th>b3</th> <th>b2</th> <th>b1</th> <th>b0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>ECB Mode</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>-</td> <td>CBC Mode</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>-</td> <td>Retail MAC Mode</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>-</td> <td>MAC Mode</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>-</td> <td>Prepare ACOS3 SM command.</td> </tr> <tr> <td>-</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>-</td> <td>MIFARE DESFire Encryption</td> </tr> <tr> <td>-</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>-</td> <td>MIFARE DESFire EV1/EV2/Light Encryption</td> </tr> <tr> <td>P1</td> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>CMAC</td> </tr> <tr> <td></td> <td>-</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td></td> <td>MIFARE Plus Command</td> </tr> <tr> <td></td> <td>-</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>MIFARE Plus Response</td> </tr> <tr> <td></td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3DES</td> </tr> <tr> <td></td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>DES</td> </tr> <tr> <td></td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3K DES</td> </tr> <tr> <td></td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>AES</td> </tr> <tr> <td></td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>All other values – RFU</td> </tr> </tbody> </table>	b7	b6	b5	b4	b3	b2	b1	b0	Description	-	0	0	0	0	0	0	-	ECB Mode	-	0	0	0	0	0	1	-	CBC Mode	-	0	0	0	0	1	0	-	Retail MAC Mode	-	0	0	0	0	1	1	-	MAC Mode	-	0	0	0	1	0	0	-	Prepare ACOS3 SM command.	-	1	0	0	1	0	1	-	MIFARE DESFire Encryption	-	1	0	0	1	1	0	-	MIFARE DESFire EV1/EV2/Light Encryption	P1	-	0	0	0	1	1	1	CMAC		-	0	1	0	0	0		MIFARE Plus Command		-	0	1	0	0	0	1	MIFARE Plus Response		0	-	-	-	-	-	0	3DES		0	-	-	-	-	-	1	DES		1	-	-	-	-	-	0	3K DES		1	-	-	-	-	-	1	AES		-	-	-	-	-	-	-	All other values – RFU
b7	b6	b5	b4	b3	b2	b1	b0	Description																																																																																																																																									
-	0	0	0	0	0	0	-	ECB Mode																																																																																																																																									
-	0	0	0	0	0	1	-	CBC Mode																																																																																																																																									
-	0	0	0	0	1	0	-	Retail MAC Mode																																																																																																																																									
-	0	0	0	0	1	1	-	MAC Mode																																																																																																																																									
-	0	0	0	1	0	0	-	Prepare ACOS3 SM command.																																																																																																																																									
-	1	0	0	1	0	1	-	MIFARE DESFire Encryption																																																																																																																																									
-	1	0	0	1	1	0	-	MIFARE DESFire EV1/EV2/Light Encryption																																																																																																																																									
P1	-	0	0	0	1	1	1	CMAC																																																																																																																																									
	-	0	1	0	0	0		MIFARE Plus Command																																																																																																																																									
	-	0	1	0	0	0	1	MIFARE Plus Response																																																																																																																																									
	0	-	-	-	-	-	0	3DES																																																																																																																																									
	0	-	-	-	-	-	1	DES																																																																																																																																									
	1	-	-	-	-	-	0	3K DES																																																																																																																																									
	1	-	-	-	-	-	1	AES																																																																																																																																									
	-	-	-	-	-	-	-	All other values – RFU																																																																																																																																									



APDU	Description
P2	<p>P2 is derived key in SAM set using Load Key function:</p> <ul style="list-style-type: none"> 1 – Encrypt Data with Session Key <i>Ks</i> 2 – Encrypt Data with Diversified Key <i>Sc</i> 3 – Encrypt Data with Bulk Encryption Key 0 – return ENC (<i>Sc</i>, <i>Ks</i>) <p>If P1.b3 = 1 or b5=1, P2 must be 1 If P2 = 0h, P1 can be either 0 or 1</p>
P3	<p>P3 < 128 If bit 3 of P1 not equal to 1 and bit 5 of P1 not equal to 1</p> <ul style="list-style-type: none"> - If P2 = 1-3, multiple of 8 (DES/3DES/3KDES) or 16 (AES) up to 128 bytes - If P2 = 0, 0
Data	<p>Plain text If P2 b6 = 1, The DATA format should be:</p> <ul style="list-style-type: none"> • Length of Plain text data • Length of Command and Header of DESFire Card • Command and Header of DESFire Card • Plain text <p>P1 = A1h, the encryption is for a MIFARE Plus command</p> <ul style="list-style-type: none"> • if MFP Command is <i>value</i> operations command, the DATA format should be Command code(1 BYTE)+BlockNum(2/4 BYTE)+Value(4 BYTE). • if MFP Command is <i>Proximity Check</i>, the DATA format should be Command code(1 BYTE)+ PPS1(1 BYTE). • if MFP Command is <i>Read</i>, the DATA format should be Command code(1 BYTE)+ BlockNum(2 BYTE) • if MFP Command is <i>Write</i>, the DATA format should be Command code(1 BYTE)+ BlockNum(2 BYTE) +plaintext <p>P1=A3h,</p> <ul style="list-style-type: none"> • The data return by ICC (don't include SC code and don't include RMAC if RMAC exist) <p>If authenticate with Authenticate EV2 First/ Authenticate EV2NonFirst command before, then P1 should be CDh or 8Fh.</p> <ul style="list-style-type: none"> • If P1=CDh, the encryption is for a DESFire EV2/Light command. The DATA format should be: Plain text data • If P1=8Fh, the CMAC is for a DESFire EV2/Light command or response. The DATA format should be: <ul style="list-style-type: none"> ➢ EV2 Command: The whole command without the CMAC data ➢ EV2 Response: The whole response from EV2 ➢ Light Command: The whole command without the CMAC data and the CLA/P1/P2/Lc/Le bytes. ➢ Light Response: The SW2 and Response data from Light. <p>If the EV2/Light card return the status code and plain text data only, then the data should be 00h, which should be sent to the SAM card.</p>



Specific Response Status Bytes

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

5.2.10.4. Decrypt

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

1. The session key created by the mutual authentication procedure with an ACOS3/6, MIFARE DESFire, MIFARE DESFire EV1/EV2/Light or MIFARE Plus card.
2. A diversified key (secret code).
3. A bulk encryption key.
4. Decrypt the diversified secret code with the session key.
5. Verify and Decrypt ACOS3 secure-messaging response.

Verify and Decrypt ACOS3 SM Response:

APDU	Description																																																																																																												
CLA	80h																																																																																																												
INS	76h																																																																																																												
	<table border="1"> <thead> <tr> <th>b7</th> <th>b6</th> <th>b5</th> <th>b4</th> <th>b3</th> <th>b2</th> <th>b1</th> <th>b0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>ECB Mode</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>-</td> <td>CBC Mode</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>-</td> <td>Verify and Decrypt ACOS3 SM Response</td> </tr> <tr> <td>-</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>-</td> <td>MIFARE DESFire Decryption</td> </tr> <tr> <td>P1</td> <td>-</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>MIFARE DESFire EV1 Decryption</td> </tr> <tr> <td>-</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>-</td> <td>MIFARE Plus Decryption</td> </tr> <tr> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3DES</td> </tr> <tr> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>DES</td> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3K DES</td> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>AES</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>All other values - RFU</td> </tr> </tbody> </table>	b7	b6	b5	b4	b3	b2	b1	b0	Description	-	0	0	0	0	0	0	-	ECB Mode	-	0	0	0	0	0	1	-	CBC Mode	-	0	0	0	1	0	0	-	Verify and Decrypt ACOS3 SM Response	-	1	0	0	1	0	1	-	MIFARE DESFire Decryption	P1	-	1	0	0	1	1	0	MIFARE DESFire EV1 Decryption	-	0	1	0	0	1	0	-	MIFARE Plus Decryption	0	-	-	-	-	-	-	0	3DES	0	-	-	-	-	-	-	1	DES	1	-	-	-	-	-	-	0	3K DES	1	-	-	-	-	-	-	1	AES	0	0	0	0	-	-	-	-	All other values - RFU
b7	b6	b5	b4	b3	b2	b1	b0	Description																																																																																																					
-	0	0	0	0	0	0	-	ECB Mode																																																																																																					
-	0	0	0	0	0	1	-	CBC Mode																																																																																																					
-	0	0	0	1	0	0	-	Verify and Decrypt ACOS3 SM Response																																																																																																					
-	1	0	0	1	0	1	-	MIFARE DESFire Decryption																																																																																																					
P1	-	1	0	0	1	1	0	MIFARE DESFire EV1 Decryption																																																																																																					
-	0	1	0	0	1	0	-	MIFARE Plus Decryption																																																																																																					
0	-	-	-	-	-	-	0	3DES																																																																																																					
0	-	-	-	-	-	-	1	DES																																																																																																					
1	-	-	-	-	-	-	0	3K DES																																																																																																					
1	-	-	-	-	-	-	1	AES																																																																																																					
0	0	0	0	-	-	-	-	All other values - RFU																																																																																																					

P2 is derived key in SAM set using Load Key function:

- | | |
|----|--|
| P2 | 1 – Decrypt Data with Session Key Ks
2 – Decrypt Data with Diversified Key Sc
3 – Decrypt Data with Bulk Encryption Key
0 – return DEC (Sc, Ks) |
|----|--|



APDU	Description
P3	<p>P3 < 128 If P1 = A5h, P3=16/32/48 If bit 3 of P1 not equal to 1</p> <ul style="list-style-type: none"> - If P2 = 1-3, multiple of 8 (DES/3DES/3KDES) or 16 (AES) up to 128 bytes - If P2 = 0, 0
Data	<p>Ciphertext If P1 = A5h, The DATA is Encrypted text If P2 b6 = 1, The DATA format should be:</p> <ul style="list-style-type: none"> • Length of Plain text data, if unknown, use 00 • Length of Command and Header of DESFire Card • Command and Header of DESFire Card • Encrypted text <p>If authenticate with Authenticate EV2 First/ Authenticate EV2NonFirst command before, and the EV2/Light card return the encrypted and CMAC data, then P1 should be CDh. The DATA format should be:</p> <ul style="list-style-type: none"> • EV2 Response: The whole response from EV2 • Light Response: The SW2 and Response data from Light.

Specific Response Status Bytes

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

5.2.10.5. Prepare Authentication

This command is used to authenticate the SAM card (as the terminal) to the ACOS 3/6 card or MIFARE Ultralight C/MIFARE DESFire Card/MIFARE DESFire EV1/EV2/Light/MIFARE Plus card.

APDU	Description
CLA	80h
INS	78h
P1	<p>00h – 3DES 01h – DES 02h – 3KDES (MIFARE DESFire EV1/ACOS3) 03h – AES (MIFARE DESFire EV1/MIFARE Plus/ACOS3) 80h – 3DES (MIFARE DESFire Authenticate only) 81h – DES (MIFARE DESFire Authenticate only) Other – RFU</p>
P2	0h – Verify ACOS3/6 Authenticate Return



APDU	Description
	01h – MIFARE Ultralight C/DESFire Authenticate by (Diversified) Terminal Key
	05h – MIFARE Ultralight C/DESFire Authenticate by Bulk Encryption Key
	02h – MIFARE Plus Authenticate. First Authenticate of SL1 to SL3
	03h – MIFARE Plus Authenticate. Authentication in SL1 to SL2.
	04h – MIFARE Plus Authenticate. Following Authenticate of SL2 to SL3.
	06h – MIFARE DESfire EV2/Light AuthenticateEV2First/AuthenticateEV2NonFirst/Mifare Plus EV1 secure messaging
P3	8 – (P1 = 00h, 01h, 02h, 80h, 81h) 16 – (P1 = 03h)
Data	Card Challenge Data

Specific Response Status Bytes

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

5.2.10.6. Verify Authentication

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

APDU	Description
CLA	80h
INS	7Ah
	00h – 3DES (P2 = 0) 01h – DES (P2 = 0)
P1	02h – 3KDES (P2 = 0 · ACOS3) 03h – AES (P2 = 0 · ACOS3) Other – RFU
P2	00h – Verify ACOS3/6 Authenticate Return 01h – Verify MIFARE Ultralight C®/ DESFire®/ DESFire® EV1 Authenticate Return 02h – Verify MIFARE Plus Authenticate return Or MIFARE DESFire EV2/Light AuthenticateEV2First/AuthenticateEV2NonFirst Return
P3	08h – (P2 = 0, P2 = 1 and Session Key is DES/3DES) 16h – (P2 = 1 and Session Key is 3KDES/AES) 16h – (P2=02, and MIFARE Plus return data ek(RndA'))



APDU	Description
	32h – (P2=02, and MIFARE Plus return data ek(TI+PICCcap2+PCDcap2))
	ACOS 3/6: DES (K _s , RND _T)
Data	MIFARE DESFire/ DESFire EV1/EV2/Light return data: ek(RndA') MIFARE Plus return data ek(RndA') or ek(TI+PICCcap2+PCDcap2)

Specific Response Status Bytes

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

5.2.10.7. Verify ACOS Inquire Account

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

APDU	Description																																																																																										
CLA	80h																																																																																										
INS	7Ch																																																																																										
	<table border="1"> <thead> <tr> <th></th> <th>b7</th> <th>b6</th> <th>b5</th> <th>b4</th> <th>b3</th> <th>b2</th> <th>b1</th> <th>b0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>0</td> <td>-</td> <td>ACOS INQ_AUT is disabled</td> </tr> <tr> <td></td> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>1</td> <td>-</td> <td>ACOS INQ_AUT is enabled</td> </tr> <tr> <td></td> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>-</td> <td>ACOS INQ_ACC_MAC is disabled</td> </tr> <tr> <td>P1</td> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>-</td> <td>-</td> <td>ACOS INQ_ACC_MAC is enabled</td> </tr> <tr> <td></td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3DES</td> </tr> <tr> <td></td> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>DES</td> </tr> <tr> <td></td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3K DES (ACOS3 only)</td> </tr> <tr> <td></td> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>AES (ACOS3 only)</td> </tr> </tbody> </table>		b7	b6	b5	b4	b3	b2	b1	b0	Description		-	0	0	0	0	-	0	-	ACOS INQ_AUT is disabled		-	0	0	0	0	-	1	-	ACOS INQ_AUT is enabled		-	0	0	0	0	0	-	-	ACOS INQ_ACC_MAC is disabled	P1	-	0	0	0	0	1	-	-	ACOS INQ_ACC_MAC is enabled		0	-	-	-	-	-	-	0	3DES		0	-	-	-	-	-	-	1	DES		1	-	-	-	-	-	-	0	3K DES (ACOS3 only)		1	-	-	-	-	-	-	1	AES (ACOS3 only)
	b7	b6	b5	b4	b3	b2	b1	b0	Description																																																																																		
	-	0	0	0	0	-	0	-	ACOS INQ_AUT is disabled																																																																																		
	-	0	0	0	0	-	1	-	ACOS INQ_AUT is enabled																																																																																		
	-	0	0	0	0	0	-	-	ACOS INQ_ACC_MAC is disabled																																																																																		
P1	-	0	0	0	0	1	-	-	ACOS INQ_ACC_MAC is enabled																																																																																		
	0	-	-	-	-	-	-	0	3DES																																																																																		
	0	-	-	-	-	-	-	1	DES																																																																																		
	1	-	-	-	-	-	-	0	3K DES (ACOS3 only)																																																																																		
	1	-	-	-	-	-	-	1	AES (ACOS3 only)																																																																																		
P2	0h																																																																																										
P3	1Dh																																																																																										
Data	Data Block returned by INQUIRE ACCOUNT of client ACOS card.																																																																																										

Specific Response Status Bytes

SW1 SW2	Description
69 86h	No DF selected



SW1 SW2	Description
6A 86h	Invalid P1 or P2
67 00h	Incorrect P3
6A 83h	ACOS Key K _s or K _{ACCT} are not ready; use DIVERSIFY command to generate K _{ACCT} ; if applicable, use "Prepare Authentication" to generate K _s .
6F 00h	Data Block's MAC is incorrect
90 00h	Data Block's MAC is correct

5.2.10.8. Prepare ACOS Account Transaction

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

APDU	Description																																				
CLA	80h																																				
INS	7Eh																																				
	<table border="1"> <thead> <tr> <th>b7</th> <th>b6</th> <th>b5</th> <th>b4</th> <th>b3</th> <th>b2</th> <th>b1</th> <th>b0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>ACOS TRNS_AUT is disabled</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>-</td> <td>ACOS TRNS_AUT is enabled</td> </tr> </tbody> </table>	b7	b6	b5	b4	b3	b2	b1	b0	Description	-	0	0	0	0	0	0	-	ACOS TRNS_AUT is disabled	-	0	0	0	0	0	1	-	ACOS TRNS_AUT is enabled									
b7	b6	b5	b4	b3	b2	b1	b0	Description																													
-	0	0	0	0	0	0	-	ACOS TRNS_AUT is disabled																													
-	0	0	0	0	0	1	-	ACOS TRNS_AUT is enabled																													
P1	<table border="1"> <tbody> <tr> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3DES</td> </tr> <tr> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>DES</td> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3K DES (ACOS3 only)</td> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>AES (ACOS3 only)</td> </tr> </tbody> </table>	0	-	-	-	-	-	-	0	3DES	0	-	-	-	-	-	-	1	DES	1	-	-	-	-	-	-	0	3K DES (ACOS3 only)	1	-	-	-	-	-	-	1	AES (ACOS3 only)
0	-	-	-	-	-	-	0	3DES																													
0	-	-	-	-	-	-	1	DES																													
1	-	-	-	-	-	-	0	3K DES (ACOS3 only)																													
1	-	-	-	-	-	-	1	AES (ACOS3 only)																													
P2	E2h: Credit E6h: Debit																																				
P3	0Dh																																				
Data	Data Block																																				

Specific Response Status Bytes

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



5.2.10.9. Verify Debit Certificate

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

APDU	Description																																				
CLA	80h																																				
INS	70h																																				
	<table border="1"> <thead> <tr> <th>b7</th> <th>b6</th> <th>b5</th> <th>b4</th> <th>b3</th> <th>b2</th> <th>b1</th> <th>b0</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>-</td> <td>ACOS TRNS_AUT is disabled</td> </tr> <tr> <td>-</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>-</td> <td>ACOS TRNS_AUT is enabled</td> </tr> </tbody> </table>	b7	b6	b5	b4	b3	b2	b1	b0	Description	-	0	0	0	0	0	0	-	ACOS TRNS_AUT is disabled	-	0	0	0	0	0	1	-	ACOS TRNS_AUT is enabled									
b7	b6	b5	b4	b3	b2	b1	b0	Description																													
-	0	0	0	0	0	0	-	ACOS TRNS_AUT is disabled																													
-	0	0	0	0	0	1	-	ACOS TRNS_AUT is enabled																													
P1	<table border="1"> <tbody> <tr> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3DES</td> </tr> <tr> <td>0</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>DES</td> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>0</td> <td>3K DES (ACOS3 only)</td> </tr> <tr> <td>1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>1</td> <td>AES (ACOS3 only)</td> </tr> </tbody> </table>	0	-	-	-	-	-	-	0	3DES	0	-	-	-	-	-	-	1	DES	1	-	-	-	-	-	-	0	3K DES (ACOS3 only)	1	-	-	-	-	-	-	1	AES (ACOS3 only)
0	-	-	-	-	-	-	0	3DES																													
0	-	-	-	-	-	-	1	DES																													
1	-	-	-	-	-	-	0	3K DES (ACOS3 only)																													
1	-	-	-	-	-	-	1	AES (ACOS3 only)																													
P2	0h																																				
P3	14h																																				
Data	Data Block																																				

Specific Response Status Bytes

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

5.2.10.10. Get Key

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

The Get Key command also allows secure key injection from the current SAM's Key File (SFI=02h) into ACOS7/10, MIFARE DESFire, MIFARE DESFire EV1/EV2/Light or MIFARE Plus card with key diversification. Using this ensures that the key to be injected is protected by encryption and message authentication codes.

If bit 7 of the Special Function Flag (Key Injection Only Flag) of the **Card Header Block** (Section 4.2 of ACOS6-SAM Reference Manual) has been set and the key file has been activated, Get Key must be used for loading or changing keys in the card. Setting this bit will disable Read Record command for the key file under any circumstances after activation.



Before this command is to be executed, a session key is already established with the target card with the mutual authentication procedure of **Mutual Authentication** (Section 6.3 of ACOS6-SAM Reference Manual) or the MIFARE Plus/MIFARE DESFire mutual authentication procedure.

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

APDU	Description																												
CLA	80h																												
INS	CAh																												
	Get Key for ACOS card Set Key																												
	00h Response data is Key in MSAM																												
	01h Response data is 16-byte Diversify Key																												
	02h Response data is 24-byte Diversify Key																												
	03h Response data is the Change Key command of MIFARE Plus Card																												
	Get Key for DESFire card Change Key, Response data for DESFire/DESFire EV1 Change Key																												
	<table border="1"> <thead> <tr> <th></th> <th>Card Type</th> <th>Authenticate Key No. And Changing Key No.*</th> <th>Key Length</th> </tr> </thead> <tbody> <tr> <td>P1</td> <td>80h MIFARE DESFire</td> <td>Are DIFFERENT in MIFARE DESFire card</td> <td>16 bytes</td> </tr> <tr> <td></td> <td>81h MIFARE DESFire EV1</td> <td>Are DIFFERENT in MIFARE DESFire EV1 card</td> <td>16 bytes</td> </tr> <tr> <td></td> <td>82h MIFARE DESFire EV1</td> <td>Are DIFFERENT in MIFARE DESFire EV1 card</td> <td>24 bytes</td> </tr> <tr> <td></td> <td>88h MIFARE DESFire</td> <td>Are the SAME in MIFARE DESFire card</td> <td>16 bytes</td> </tr> <tr> <td></td> <td>89h MIFARE DESFire EV1</td> <td>Are the SAME in MIFARE DESFire EV1 card</td> <td>16 bytes</td> </tr> <tr> <td></td> <td>8Ah MIFARE DESFire EV1</td> <td>Are the SAME in MIFARE DESFire EV1 card</td> <td>24 bytes</td> </tr> </tbody> </table>		Card Type	Authenticate Key No. And Changing Key No.*	Key Length	P1	80h MIFARE DESFire	Are DIFFERENT in MIFARE DESFire card	16 bytes		81h MIFARE DESFire EV1	Are DIFFERENT in MIFARE DESFire EV1 card	16 bytes		82h MIFARE DESFire EV1	Are DIFFERENT in MIFARE DESFire EV1 card	24 bytes		88h MIFARE DESFire	Are the SAME in MIFARE DESFire card	16 bytes		89h MIFARE DESFire EV1	Are the SAME in MIFARE DESFire EV1 card	16 bytes		8Ah MIFARE DESFire EV1	Are the SAME in MIFARE DESFire EV1 card	24 bytes
	Card Type	Authenticate Key No. And Changing Key No.*	Key Length																										
P1	80h MIFARE DESFire	Are DIFFERENT in MIFARE DESFire card	16 bytes																										
	81h MIFARE DESFire EV1	Are DIFFERENT in MIFARE DESFire EV1 card	16 bytes																										
	82h MIFARE DESFire EV1	Are DIFFERENT in MIFARE DESFire EV1 card	24 bytes																										
	88h MIFARE DESFire	Are the SAME in MIFARE DESFire card	16 bytes																										
	89h MIFARE DESFire EV1	Are the SAME in MIFARE DESFire EV1 card	16 bytes																										
	8Ah MIFARE DESFire EV1	Are the SAME in MIFARE DESFire EV1 card	24 bytes																										
P2	Key ID in SAM (New key for change)																												
	If P1 = 00h, P3 is 08h																												
P3	If P1 = 01/02h, P3 is 10h																												
	If P1 = 03h, P3 is 0Bh																												
	If P1 = 80/81/82/88/89/8Ah: P3 is 0Bh or 0Dh																												



APDU	Description
	If P1 = 00h, command data is RND _{Target}
	If P1 = 01/02h, command data is RND _{Target} + serial (or batch) number of target card
	If P1 = 03h
	- Serial Number for target card (8 Byte)
	- Write Command (A0 or A1) (1 Byte)
	- BNr (2 Byte)
Data	If P1 = 80/81/82/88/89/8Ah:
	- Serial Number for target card (8 Byte)
	- Original Key ID (Key in SAM card stored the Original key, 00 = Default Key of DESFire - Card)
	- Key No. (DESFire Card Key No.)
	- Key Version (DESFire Card Key Version, If not used, value = 00)
	- Command:C6 (Optional)
	- KeySetNo (Optional)

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

Specific Response Status Bytes

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



5.3. Contactless Smart Card Protocol

5.3.1. ATR Generation

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

5.3.1.1. ATR Format for ISO14443 Part 3 PICCs

Byte	Value	Designation	Description
0	3Bh	Initial Header	
1	8Nh	T0	Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following. Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)
2	80h	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0
3	01h	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1
4 ~ 3+N	80h	T1	Category indicator byte, 80 means A status indicator may be present in an optional COMPACT-TLV data object
	4Fh	Tk	Application identifier Presence Indicator
	0Ch		Length
	RID		Registered Application Provider Identifier (RID) # A0 00 00 03 06
	SS		Byte for standard
	C0 .. C1h		Bytes for card name
	00 00 00 00h	RFU	RFU # 00 00 00 00
4+N	UU	TCK	Exclusive-oring of all the bytes T0 to Tk

Example:

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

Where:

- Length (YY) = 0Ch
- RID = {A0 00 00 03 06h} (PC/SC Workgroup)
- Standard (SS) = 03h (ISO 14443A, Part 3)
- Card Name (C0 .. C1) = {00 01h} (MIFARE Classic 1K)
- Standard (SS) = 03h: ISO 14443A, Part 3
= 11h: FeliCa

Card Name (C0 .. C1):

- 00 01: MIFARE Classic 1K
- 00 02: MIFARE Classic 4K
- 00 03: MIFARE Ultralight®
- 00 26: MIFARE Mini®
- 00 3A: MIFARE Ultralight® C
- 00 36: MIFARE Plus® SL1 2K
- 00 37: MIFARE Plus® SL1 4K
- 00 38: MIFARE Plus® SL2 2K
- 00 39: MIFARE Plus® SL2 4K
- 00 30: Topaz and Jewel
- 00 3B: FeliCa
- FF 28: JCOP 30
- FF [SAK]: undefined tags



5.3.1.2. ATR Format for ISO14443 Part 4 PICCs

Byte	Value	Designation	Description						
0	3Bh	Initial Header							
1	8Nh	T0	Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following. Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)						
2	80h	TD1	Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following. Lower nibble 0 means T = 0						
3	01h	TD2	Higher nibble 0 means no TA3, TB3, TC3, TD3 following. Lower nibble 1 means T = 1						
4 ~ 3+N	XX	T1	Historical Bytes: ISO 14443-A: The historical bytes from ATS response. Refer to the ISO 14443-4 specification. ISO 14443-B: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Byte 1~4</th> <th style="width: 33%;">Byte 5~7</th> <th style="width: 33%;">Byte 8</th> </tr> </thead> <tbody> <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> </tbody> </table>	Byte 1~4	Byte 5~7	Byte 8	Application Data from ATQB	Protocol Info Byte from ATQB	Higher nibble=MBLI from ATTRIB command Lower nibble (RFU)=0
	Byte 1~4	Byte 5~7		Byte 8					
	Application Data from ATQB	Protocol Info Byte from ATQB		Higher nibble=MBLI from ATTRIB command Lower nibble (RFU)=0					
XX	Tk								
4+N	UU	TCK	Exclusive-oring of all the bytes T0 to Tk						

Example 1:

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

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

APDU Command = FF CA 01 00 00h
 APDU Response = 06 75 77 81 02 80 90 00h
 ATS = {06 75 77 81 02 80h}

Example 2:

ATR for EZ-Link = {3B 88 80 01 1C 2D 94 11 F7 71 85 00 BEh}
 Application Data of ATQB = 1C 2D 94 11h
 Protocol Information of ATQB = F7 71 85h
 MBLI of ATTRIB = 00h



5.3.2. APDU, Pseudo APDU and Card Native Command

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

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

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

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

5.3.3.1. Get Data [FF CA ...]

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

Command

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

Command Parameter

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



P1	P2	Meaning
		CTS: 4 byte SN + 2 byte ATQT Innovatron: 4 byte SN + 1 byte tag address.

Response

Response	Data Out		
Result	Data	SW1	SW2

Response Code

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

Examples:

To get the serial number of the “connected PICC”:

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

To get the ATS of the “connected ISO 14443 A PICC”:

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

5.3.3.2. Load Key [FF 82 ...]

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

Command

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

Key Length/Data

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



Response Code

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

Example:

// Load a key {FF FF FF FF FF FFh} into the volatile memory location 00h.

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

5.3.3.3. Authenticate [FF 86 00 00 05 ...]

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

Command

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

Command Data

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

Address and Key Type

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

Response Code

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



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 4: MIFARE Classic 1K Memory Map

Sectors (Total 32 sectors. Each sector consists of 4 consecutive blocks)	Data Blocks (3 blocks, 16 bytes per block)	Trailer Block (1 block, 16 bytes)
Sector 0	00h ~ 02h	03h
Sector 1	04h ~ 06h	07h
..		
..		
Sector 30	78h ~ 7Ah	7Bh
Sector 31	7Ch ~ 7Eh	7Fh

} 2 KB

Sectors (Total 8 sectors. Each sector consists of 16 consecutive blocks)	Data Blocks (15 blocks, 16 bytes per block)	Trailer Block (1 block, 16 bytes)
Sector 32	80h ~ 8Eh	8Fh
Sector 33	90h ~ 9Eh	9Fh
..		
..		
Sector 38	E0h ~ EEh	EFh
Sector 39	F0h ~ FEh	FFh

} 2 KB

Table 5: MIFARE Classic 4K Memory Map



Byte Number	0	1	2	3	Page
Serial Number	SN0	SN1	SN2	BCC0	0
Serial Number	SN3	SN4	SN5	SN6	1
Internal/Lock	BCC1	Internal	Lock0	Lock1	2
OTP	OPT0	OPT1	OTP2	OTP3	3
Data read/write	Data0	Data1	Data2	Data3	4
Data read/write	Data4	Data5	Data6	Data7	5
Data read/write	Data8	Data9	Data10	Data11	6
Data read/write	Data12	Data13	Data14	Data15	7
Data read/write	Data16	Data17	Data18	Data19	8
Data read/write	Data20	Data21	Data22	Data23	9
Data read/write	Data24	Data25	Data26	Data27	10
Data read/write	Data28	Data29	Data30	Data31	11
Data read/write	Data32	Data33	Data34	Data35	12
Data read/write	Data36	Data37	Data38	Data39	13
Data read/write	Data40	Data41	Data42	Data43	14
Data read/write	Data44	Data45	Data46	Data47	15

} 512 bits
or
64 bytes

Table 6: MIFARE Ultralight Memory Map

Examples:

// To authenticate the Block 04h with a {TYPE A, key number 00h}. PC/SC V2.07

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

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

5.3.3.4. Read Binary Blocks [FF B0 ...]

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

Command:

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

P1/P2 (Mode and Address)

Card Type	P1[7:4] Mode	P1[3:0] + P2[7:0] Starting Address (MSB First)
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	0h: Skip Trailers 8h: With Trailers	000h~0FFh: Block 0~255
MIFARE Ultralight MIFARE Ultralight C	0h (Reserved)	000h~02Fh: Page 0~47
SRIX4K/SRT512	0h (Reserved)	000h~07Fh: Block 0~127 0FFh: System Area
PicoPass	0h (Reserved)	000h~0FFh: Block 0~255
ISO15693	0h (Reserved)	000h~7FFh: Block 0~2047
Topaz/NFC Type-1 Tag	0h (Reserved)	000h~7FFh: Byte Address
CTS	0h (Reserved)	000h~01Fh: Block 0~31

Le (Number of Bytes to Read)

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

Response Code

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



Examples:

// Read 16 bytes from the binary block 04h (MIFARE Classic 1K or 4K)

APDU = FF B0 00 04 10h

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

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

APDU = FF B0 00 80 F0h

5.3.3.5. Update Binary Blocks [FF D6 ...]

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

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

Command

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

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

Card Type	P1[7:4] Mode	P1[3:0] + P2[7:0] Starting Address (MSB First)	Blk/Page Size (Bytes)
MIFARE Standard MIFARE Plus SL1 MIFARE Plus SL2	0h: Skip Trailers 8h: With Trailers	000h~0FFh: Block 0~255	16
MIFARE Ultralight MIFARE UltraLightC	0h (Reserved)	000h~02Fh: Page 0~47	4
SRIX4K/SRT512	0h (Reserved)	SRIX4K/SRT512	4
PicoPass	0h (Reserved)	PicoPass	8
ISO15693	0h (Reserved)	000h~7FFh: Block 0~2047	ISO15693
Topaz/NFC Type-1 Tag	0h: with Erase 8h: without Erase	000h~7FFh: Byte Address	1(Addr 78h) or 8(Else)
CTS	0h (Reserved)	000h~01Fh: Block 0~31	CTS



Lc (Number of Bytes to Write)

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

Response Code

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

Examples:

// Update the binary block 04h of MIFARE Classic 1K/4K with Data {00 01 .. 0Fh}

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

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

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



5.3.4. Pass Though Command

5.3.4.1. Accessing ISO14443-3 Tags

This section demonstrates how to use the pass-through command format to send commands compliant with ISO14443-3, the firmware requirements are illustrated as follows:

- ACR1581U-C FW 1.09.00 or above

Command

Command	Class	INS	P1	P2	Lc	Data In
ISO14443-3 Command	FFh	00h	00h	00h	Data Length	Data

Response

Response	Data Out		
Result	Data	SW1	SW2

Response Code

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

5.3.4.1.1. Transfer ISO14443-3 Native Command to Pass Through Command

This section explains how to convert ISO14443 native commands into the pass-through format.

Example:

Mifare Ultralight AES Get Version

60

Pass Through Format

FF 00 00 00 01 60



5.3.4.2. Accessing ISO15693 Tags

This section demonstrates how to use the pass-through command format to send commands compliant with ISO15693, the firmware requirement are illustrate as follow:

- ACR1581U-C FW 1.08.00 or above

Command

Command	Class	INS	P1	P2	Lc	Data In		Le
ISO15693 Command	FFh	FBh	00h	Flags	Data Length+1	Command Code	Data	--/00h

Where:

Flags: 00h (Default): Sets the flags to 22h. In this case, the data field does not need to include the UID.

Note: This option cannot be used with Custom or Proprietary commands.

Others: use the value as flags value

Response

Response	Data Out		
Result	Data	SW1	SW2

Response Code

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

5.3.4.2.1. Transfer ISO15693 Command to Pass Though Command

This section explains how to convert ISO15693 commands into the pass-through format. Please note that the pass-through command supports only commands operating at 26 kbps. For commands that operate at 53 kbps, refer to *Error! Reference source not found.*

Example 1:

Stay Quiet (Mandatory Command)

22 02 XX XX XX XX XX XX XX XX

Where:

22 Flags

02 Stay quiet command code

XX XX XX XX XX XX XX XX 8 bytes UID

Pass Though Format

FF FB 00 22 09 02 XX XX XX XX XX XX XX XX

Or



FF FB 00 00 01 02

Example 2:

Read Multiple Blocks with non-addressed (Optional Command)

02 23 00 01

Where:

- 02 Flags
- 23 Read multiple blocks command code
- 00 First block number
- 01 Read 2 blocks

Pass Though Format

FF FB 00 02 03 23 00 01

Example 3:

Get Random Number in Selected state (Custom Command for NXP SLIX2)

12 B2 04

Where:

- 12 Flags
- B2 Get random number command code
- 04 NXP manufacturer code

Pass Though Format

FF FB 00 12 02 B2 04

5.3.4.2.2. Supported Commands list

The following table lists all commands specified in ISO15693-3. Commands marked as N/A indicate that testing has not yet been performed. For inquiries regarding these commands, please contact info@acs.com.hk. Please note that ISO15693 cards may not support all optional commands. Refer to the respective card's datasheet to confirm command support.

Command Code	Type	Function	Support
01h	Mandatory	Inventory	Yes
02h	Mandatory	Stay Quiet	Yes
20h	Optional	Read Single Block	Yes
21h	Optional	Write Single Block	Yes
22h	Optional	Lock Block	Yes
23h	Optional	Read Multiple Blocks	Yes
24h	Optional	Write Multiple Blocks	Yes
25h	Optional	Select	Yes
26h	Optional	Reset to Ready	Yes
27h	Optional	Write AFI	Yes
28h	Optional	Lock AFI	Yes
29h	Optional	Write DSFID	Yes



Command Code	Type	Function	Support
2Ah	Optional	Lock DSFID	Yes
2Bh	Optional	Get System Information	Yes
2Ch	Optional	Get Multiple Block Security Status	Yes
2Dh	Optional	Fast Read Multiple Blocks	No
30h	Optional	Extended Read Single Block	Yes
31h	Optional	Extended Write Single Block	Yes
32h	Optional	Extended Lock Block	Yes
33h	Optional	Extended Read Multiple Blocks	Yes
34h	Optional	Extended Write Multiple Blocks	Yes
35h	Optional	Authenticate	Yes
36h	Optional	KeyUpdate	N/A
37h	Optional	AuthComm Crypto Format Indicator	N/A
38h	Optional	SecureComm Crypto Format Indicator	N/A
39h	Optional	Challenge	No
3Ah	Optional	ReadBuffer	Yes
3Bh	Optional	Extended Get System Information	Yes
3Ch	Optional	Extended Get Multiple Block Security Status	Yes
3Dh	Optional	Fast Extended Read Multiple Blocks	No
A0-DFh	Custom	IC Mfg Dependent	N/A
E0-FFh	Proprietary	IC Mfg Dependent	N/A

5.3.4.3. Accessing FeliCa tags

For FeliCa access, the command is different from the one used in PCSC-compliant and MIFARE tags. The command follows the FeliCa specification with an added header.

FeliCa Command Format

Command	Class	INS	P1	P2	Lc	Data In
FeliCa Command	FFh	00h	00h	00h	Length of the Data In	FeliCa Command (start with Length Byte)

FeliCa Response Format (Data + 2 bytes)

Response	Data Out
Result	Response Data

Read Memory Block Example:

1. Connect the FeliCa.

The ATR = 3B 8F 80 01 80 4F 0C A0 00 00 03 06 **11 00 3B** 00 00 00 00 42h

In which, **11 00 3B** = FeliCa

2. Read FeliCa IDM.

CMD = FF CA 00 00 00h



RES = [IDM (8bytes)] 90 00h
e.g., FeliCa IDM = 01 01 06 01 CB 09 57 03h

3. FeliCa command access.

Example: "Read" Memory Block.

CMD = FF 00 00 00 10 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

where:

Felica Command = 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

IDM = 01 01 06 01 CB 09 57 03h

RES = Memory Block Data

5.3.5. APDU Commands for PCSC 2.0 Part 3 (Version 2.02 or above)

PCSC2.0 Part 3 commands are used to transparently pass data from an application to a contactless tag, return the received data transparently to the application and protocol, and switch the protocol simultaneously.

5.3.5.1. PCSC 2.0 Part 3 Flow

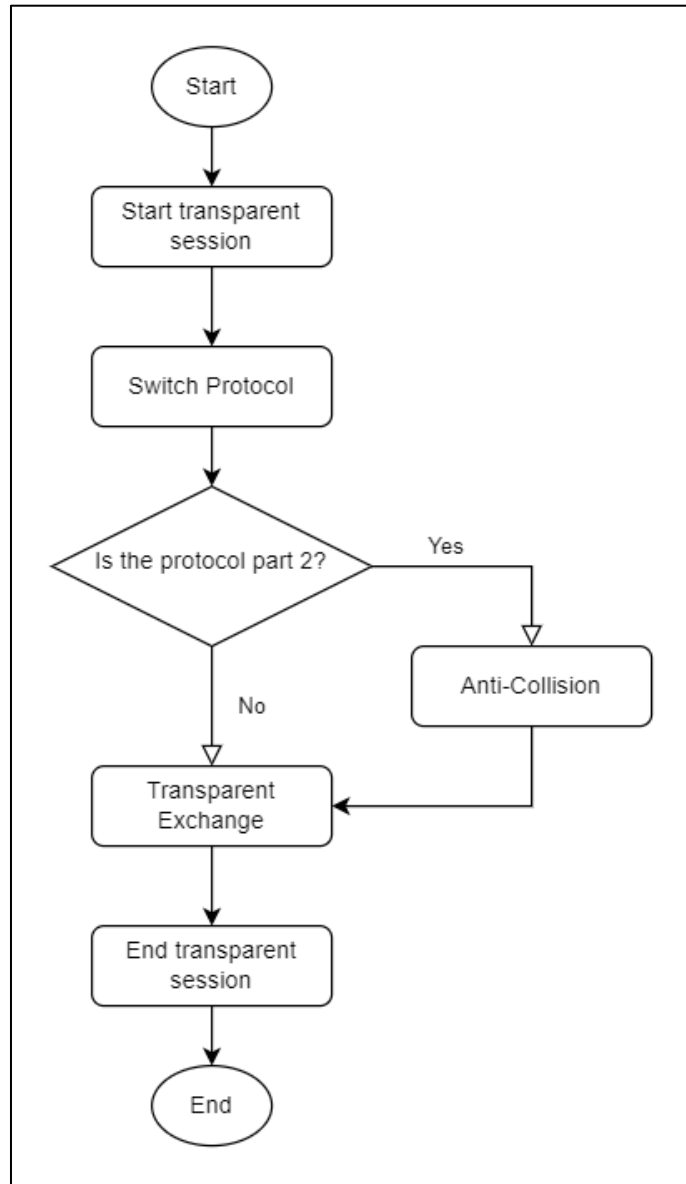


Figure 5: ACR1581U Transparent Session Flow



5.3.5.2. Command and Response APDU Format

Command Format

CLA	INS	P1	P2	Lc	Data In
FFh	C2h	00h	Function	DataLen	Data[DataLen]

Where Functions (1 byte):

- 00h = Manage Session
- 01h = Transparent Exchange
- 02h = Switch Protocol
- Other = RFU

Response Format

Data Out	SW1	SW2
Data Field BER-TLV encoded		

Every command returns SW1 and SW2 together with the response data field (if available). The SW1 SW2 is based on ISO 7816. SW1 SW2 from the C0 data object below should also be used.

C0 data element Format

Tag	Length (1 byte)	SW2
C0h	03h	Error Status

Error Status Description

Error Status	Description
XX SW1 SW2	XX = number of the bad data object in the APDU 00 = general error of APDU 01 = error in the 1 st data object 02 = error in the 2 nd data object
00 90 00h	No error occurred
XX 62 82h	Data object XX warning, requested information not available
XX 63 00h	No information
XX 63 01h	Execution stopped due to failure in other data object
XX 6A 81h	Data object XX not supported
XX 67 00h	Data object XX with unexpected length
XX 6A 80h	Data object XX with unexpected value
XX 64 00h	Data Object XX execution error (no response from IFD)
XX 64 01h	Data Object XX execution error (no response from ICC)
XX 6F 00h	Data object XX failed, no precise diagnosis

The first value byte indicates the number of the erroneous data object XX, while the last two bytes indicate the explanation of the error. SW1 SW2 values based on ISO 7816 are allowed.

If there are more than one data objects in the C-APDU field and one data object failed, IFD can process the following data objects if they do not depend on the failed data objects.



5.3.5.3. Manage Session [FF C2 00 00 ...]

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

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

Command

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

Response Code

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

Cmd TLV

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

Rsp TLV

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



5.3.5.3.1. Start Session Data Object

This command is used to start a transparent session. Once the session has started, auto-polling will be disabled until the session is ended.

Start Session Data Object

Tag	Length (1 byte)	Value
81h	00h	-

5.3.5.3.2. End Session Data Object

This command ends the transparent session. The auto-polling will be reset to the state before the session has started.

End Session Data Object

Tag	Length (1 byte)	Value
82h	00h	-

5.3.5.3.3. Turn Off the RF Data Object

This command turns off the antenna field.

Turn off RF Field Data Object

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

5.3.5.3.4. Turn On the RF Data Object

This command turns on the antenna field.

Turn on the RF Field Data Object

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

5.3.5.3.5. Timer Data Object

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

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

Timer Data Object

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

5.3.5.4. Transparent Exchange [FF C2 00 01 ...]

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

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

Command

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

Response Code

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

Cmd TLV

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



Cmd	Meaning
	FWTI: 0 ~ 15, FWT/Timeout = 302.07 x 2FWTI us
Transceive: 95h [Size] [Data]	Size: Size of Data coded in BER-TLV length field. Data: Data to be Transmit.

Rsp TLV

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

5.3.5.4.1. Transmission and Reception Flag Data Object

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

Transmission and Reception Flag Data Object

Tag	Length (1 byte)	Value		
		Byte 0		Byte 1
		bit	Description	
90h	02h	0	0 – append CRC in the transmit data 1 – do not append CRC in the transmit data	00h
		1	0 – CRC checking from the received data 1 – no CRC checking from the received data	
		2	0 – insert parity in the transmit data 1 – do not insert parity	
		3	0 – expect parity in received date 1 – do not expect parity (i.e. no parity checking)	
		4	0 – append protocol prologue in the transmit data or discard from the response 1 – do not append or discard protocol prologue if any (e.g. PCB, CID, NAD)	
		5-7	RFU	

5.3.5.4.2. Transmission Bit Framing Data Object

This command defines the number of valid bits of the last byte of data to transmit or transceive.

Transmission bit Framing Data Object

Tag	Length (1 byte)	Value	
		bit	Description
91h	01h	0-2	Number of valid bits of the last byte (0 means all bits are valid)
		3-7	RFU

Transmission bit framing data object shall be together with “transmit” or “transceive” data object only. If this data object does not exist, it means all bits are valid.

5.3.5.4.3. Transceive Data Object

This command transmits and receives data from the ICC. After transmission is complete, the reader will wait until the time given in the timer data object.

If no timer data object was defined in the data field, the reader will wait for the duration given in the Set Parameter FWTI Data Object. If no FWTI is set, the reader will wait for about 302 μ s.

Transceive Data Object

Tag	Length		Value
95h	01-7Fh		Data (1~127 Bytes)
95h	81h	80h or above	Data (128~N Bytes)

5.3.5.4.4. Timer Data Object

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

Example: : If there is a timer data object with 5000 μ s, the reader will wait the following Transceive TLV for about 5000 μ s before timeout.

Timer Data Object

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

5.3.5.4.5. Response Bit Framing Data Object

Inside the response, this command is used to notify the received transmission bit Framing Data Object

Tag	Length (1 byte)	Value	
		bit	Description
92h	01h	0-2	Number of valid bits of the last byte (0 means all bits are valid)
		3-7	RFU

Transmission bit framing data object shall be together with “transmit” or “transceive” data object only. If this data object does not exist, it means all bits are valid.

5.3.5.4.6. Response Status Data Object

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

Response Status Data Object

Tag	Length (1 byte)	Value		
		Byte 0		Byte 1
		Bit	Description	
96h	02h	0	0 – CRC is OK or no checked 1 – CRC check fail	RFU
		1	0 – no collision 1 – collision detected	
		2	0 – no parity error 1 – parity error detected	
		3	0 – no framing error 1 – framing error detected	
		4 - 7	RFU	

5.3.5.4.7. Response Data Object

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

Response Data Object

Tag	Length		Value
97h	01-7Fh		ReplyData (1~127 Bytes)
	81h	80 - FFh	ReplyData (128~255 Bytes)
	82h	-	ReplyData (256~N Bytes)

5.3.5.5. Switch Protocol [FF C2 00 02 ...]

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

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

Command

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

Response Code

Rsp Data	SW1 SW2	Meaning
Rsp TLV	90 00h	Succeed with data.



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

Cmd TLV

Cmd	Meaning
Set Baud: FF 6E 03 05 01h [Baud]	Set the Baud for Part/Layer 4 to be applied during Switch Protocol. If [Baud] does not set by any previous "FF C2h ..." command, the default value is 98h (106 kbps). ISO14443: 98h (106 kbps), 99h (212 kbps), 9A (424 kbps), 9B (848 kbps). ISO15693: 80h (26 kbps), 08h (53 kbps)
Switch Protocol: 8F 02h [RF] [Layer]	Switch the protocol to specified RF and/or Layer. [RF]: 00h: ISO14443A, 01h: ISO14443B 02h: ISO15693, 03h: FeliCa, FFh: Current RF Other: RFU [Layer]: 02h: Layer/Part 2, 03h: Layer/Part 3, 04h: Layer/Part 4 (For A/B Only) Other: RFU Note: It must be in a Transparent Session (Disable Polling) if switching to Layer/Part 2.

Rsp TLV

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

5.3.5.5.1. Switch Protocol Data Object

This command specifies the protocol and different layers of the standard.

Switch Protocol Data Object

Tag	Length (1 byte)	Value	
		Byte 0	Byte 1
8Fh	02h	00h – ISO/IEC14443 Type A 01h – ISO/IEC14443 Type B 02h – ISO15693 03h – FeliCa	02h – Switch to Layer 2 03h – Switch or activate to layer 3 04h – Activate to layer 4 Other - RFU



Tag	Length (1 byte)	Value	
		Byte 0	Byte 1
		Other – RFU	

5.3.5.5.2. Response Data Object

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

Response Data Object

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

5.3.5.6. PCSC 2.0 Part 3 Example

8. Start Transparent Session.
Command: **FF C2 00 00 02 81 00**
Response: **C0 03 00 90 00 90 00**
9. Turn the Antenna Field off.
Command: **FF C2 00 00 02 83 00**
Response: **C0 03 00 90 00 90 00**
10. Turn the Antenna Field on.
Command: **FF C2 00 00 02 84 00**
Response: **C0 03 00 90 00 90 00**
11. ISO 14443-4A Active.
Command: **FF C2 00 02 04 8F 02 00 04**
Response: **C0 03 01 64 01 90 00** (if no card present)
C0 03 00 90 00 5F 51 [Len] [ATR] 90 00
12. Set the PCB to 0Ah and enable the CRC, parity and protocol prologue in the transmit data.
Command: **FF C2 00 01 0A 90 02 00 00 FF 6E 03 07 01 0A**
Response: **C0 03 00 90 00 90 00**
13. Send the APDU “80B2000008” to card and get response.
Command: **FF C2 00 01 0E 5F 46 04 40 42 0F 00 95 05 80 B2 00 00 08**
Response: **C0 03 00 90 00 92 01 00 96 02 00 00 97 0C [Card Response] 90 00**
14. End Transparent Session.
Command: **FF C2 00 00 02 82 00**
Response: **C0 03 00 90 00 90 00**



5.3.6. Proprietary Pseudo APDU for PICC

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

5.3.6.1. Write Value Block [FF D7 ...]

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

Command

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

Command Data

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

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

Response Code

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



5.3.6.2. Read Value Block [FF B1 ...]

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

Command

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

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

Response

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

5.3.6.3. Decrement/Increment Value [FF D7 ...]

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

Command

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

Command Data

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



Response Code

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

5.3.6.4. Copy Value Block [FF D7 ...]

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

Command

Command	Class	INS	P1	P2	Lc	Data In
Copy Value Block	FFh	D7h	00	Source Block#	02h	See below

Command Data

Byte 0	Byte 1
03h	Target Block#

Response Code

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

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

All ISO 14443-4 compliant cards (PICCs) understand the ISO 7816-4 APDUs. The ACR1581U reader just has to communicate with the ISO 14443-4 compliant cards by exchanging ISO 7816-4 APDUs and responses. The ACR1581U will handle the ISO 14443 Parts 1-4 Protocols internally.

MIFARE Classic (1K/4K), MIFARE Mini and MIFARE Ultralight tags are supported through the T=CL emulation. Just simply treat the MIFARE tags as standard ISO 14443-4 tags. For more information, please refer to **PCSC Pseudo APDU (with Proprietary Extension) for PICC**.

ISO 7816-4 APDU Format

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

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

Response	Data Out		
Result	Response Data	SW1	SW2



Common ISO 7816-4 Response Codes

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

Typical sequence may be:

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

To do this:

1. Connect the tag.

The ATR of the tag is 3B 88 80 01 00 00 00 00 33 81 81 00 3Ah.

In which,

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

2. Send an APDU, Get Challenge.

<< 00 84 00 00 08h

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

Note: For ISO 14443-4 Type A tags, the ATS can be obtained by using the APDU “FF CA 01 00 00h.”

Example:

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

APDU = {80 B2 80 00 08h}

Class = 80h

INS = B2h

P1 = 80h

P2 = 00h

Lc = None

Data In = None

Le = 08h

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



5.3.8. Supported PICC ATR

The following PICC type/technology are supported by default. The following ATR is returned to CCID Host on SCardStatus() in PCSC API if the card is presented to the reader.

Card Type/Technology	ATR
MIFARE Std 1k ²	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6A
MIFARE Std 4k ²	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 02 00 00 00 00 69
MIFARE UltraLight ²	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 03 00 00 00 00 68
MIFARE Ultralight EV1 ²	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 3D 00 00 00 00 56
MIFARE Plus SL1 2k ²	Default: Same as MIFARE Std 1k Alternated: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 36 00 00 00 00 5D
MIFARE Plus SL1 4k ²	Default: Same as MIFARE Std 4k Alternated: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 37 00 00 00 00 5C
MIFARE Plus SL2 2k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 38 00 00 00 00 53
MIFARE Plus SL2 4k	3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 39 00 00 00 00 52
MIFARE Ultralight C ²	Default: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 3A 00 00 00 00 51 Alternated: Same as MIFARE Ultralight
SmartMX with MIFARE Std 1k Emulation ²	Default: Same as MIFARE Std 1k Alternated: Same as ISO14443-4, Type A
SmartMX with MIFARE Std 4k Emulation ²	Default: Same as MIFARE Std 4k Alternated: Same as ISO14443-4, Type A
ISO14443-4, Type A	3B 8n 80 01 T1 .. Tn Tck n = Number of Historical bytes in ATS T1 .. Tn = Historical bytes in ATS Tck = XOR of 8n 80 01 T1 .. Tn
ISO14443-4, Type B	3B 88 80 01 T1 .. T8 Tck T1 .. T4 = Application Data in ATQB T5 .. T7 = Protocol Info in ATQB T8 = MBLI in ATA Tck = XOR of 88 80 01 T1 .. T8
FeliCa	3B 8F 80 01 80 4F 0C A0 00 00 03 06 11 00 3B 00 00 00 00 42
ISO15693-3 Generic	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 00 00 00 00 00 63
Infineon My-D Vicinity (SRF55Vxxx)	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 0E 00 00 00 00 6D
ST LRI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 13 00 00 00 00 70
NXP I-Code SLI	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 14 00 00 00 00 77

² Refer to “Bit 3” and “Bit 7” in [Set Polling/ATR Option \[E0 00 00 23 01 ...\]](#) Escape command for configuration and drawback of the alternated ATR definition.



Card Type/Technology	ATR
NXP I-Code SLIX/SLIX2	3B 8F 80 01 80 4F 0C A0 00 00 03 06 0B 00 35 00 00 00 00 56
PicoPass 2K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 17 00 00 00 00 79
PicoPass 2KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 18 00 00 00 00 76
PicoPass 16K	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 19 00 00 00 00 77
PicoPass 16KS	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1A 00 00 00 00 74
PicoPass 16K (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1B 00 00 00 00 75
PicoPass 16KS (8 x 2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1C 00 00 00 00 72
PicoPass 32KS (16 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1D 00 00 00 00 73
PicoPass 32KS (16 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1E 00 00 00 00 70
PicoPass 32KS (8x2 + 16)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 1F 00 00 00 00 71
PicoPass 32KS (8x2 + 8x2)	ISO14443B: 3B 8F 80 01 80 4F 0C A0 00 00 03 06 06 00 20 00 00 00 00 4E



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

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



6.0. Escape Command

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

Note:

Should send these commands under correct interface. For example, E0 00 00 25 01 00 (Section 6.1.1) should send through PICC interface (Section 6.0).

6.1. Escape Command for PICC

6.1.1. RF Control [E0 00 00 25 01 ...]

This command is used to set the RF control.

Command

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

Response Code

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

RF Status: 1 Byte

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

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

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

This command is used to get the PCD/PICC status

Command

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

Response Code

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

PCD/PICC Status: 1 Byte

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



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

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

Command

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

Response Code

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

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

This command is used to set the polling option.

Command

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

Response Code

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

PICC Polling/ATR Option - 1 Byte

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

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

Operating Parameter		USB Active (D0)
Bit 5	Bit 4	
0	0	No RF Off
0	1	
1	0	
1	1	

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

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

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



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

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

Command

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

Response Code

Response	Class	INS	P1	P2	Le	Data In	
						Byte 1	Byte 0
Result	E1h	00h	00h	00h	02h	PICC Polling Type	

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

This command is used to set the PICC polling type.

Command	Class	INS	P1	P2	Lc	Data Out	
						Byte 1	Byte 0
Set PICC Polling Type	E0h	00h	01h	20h	02h	PICC Polling Type	

Response Code

Response	Class	INS	P1	P2	Le	Data In	
						Byte 1	Byte 0
Result	E1h	00h	00h	00h	02h	PICC Polling Type	

PICC Polling Type - 2 Byte, Bit Mask of following

Bytes	Operating Parameter	Parameter	Description	Option
Byte 1	Bit 0	ISO 14443A	The Tag Types to be detected during PICC Polling. RFU bit should be set to 0.	1 = Detect 0 = Skip
	Bit 1	ISO 14443B		
	Bit 2	FeliCa		
	Bit 3	RFU		
	Bit 4	Topaz		
	Bit 5	Innovatron		
	Bit 6	SRI/SRIX		
	Bit 7	RFU		
Byte 0	Bit 0	Picopass (ISO14443B)		
	Bit 1	Picopass (ISO15693)		



Bytes	Operating Parameter	Parameter	Description	Option
	Bit 2	ISO15693		
	Bit 3	CTS		
	Bit 4-7	RFU		

Default Setting – Byte 1: 07h (ISO14443A, ISO14443B, FeliCa)

Byte 0: 05h (Picopass (ISO14443B), ISO15693)

Example:

Command: E0 00 01 20 02 07 05

Response: E1 00 00 00 02 07 05

Polling Type: Byte 1 = 07h = 0000 0111b = ISO14443A, ISO14443B, FeliCa

Byte 0 = 05h = 0000 0101b = Picopass (ISO14443B), ISO15693

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

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

Command

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

Response Code

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

6.1.8. Set Auto PPS [E0 00 00 24 01 ...]

This command is used to set the auto PPS.

Command

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

Response Code

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



Speed of PPS

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

Default Setting – 02h (424 kbps)

Notes:

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

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

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

This command is used to read the PICC type.

command

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

Response Code

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

Type: 1 Byte

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



Status: 1 Byte

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

6.1.10. Get RF Power Setting [E0 00 00 50 00]

This command is used to read the RF power setting, the firmware requirement are illustrate as follow:

- ACR1581U-C FW 1.09.00 or above

Command

Command	Class	INS	P1	P2	Le
Get RF Power Setting	E0h	00h	00h	50h	00h

Response Code

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

6.1.11. Set RF Power Setting [E0 00 00 50 01 ...]

This command is used to set the RF power setting, the firmware requirement is as same as Get RF Power Setting.

Notice: After set RF setting the reader would reset once.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set RF Power Setting	E0h	00h	01h	50h	01h	RF Power

Response Code

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

Percentage mode

RF Power - 1 Byte

Parameter	Description
00h	Disable manual RF Power setting
01h	20%



Parameter	Description
02h	40%
03h	60%
04h	80%
05h	100%

Default Setting – 00h

* RF Power value in Percentage mode may not have effective due to hardware limitation.

6.1.12. Get Selective Suspend Setting [E0 00 00 E5 00]

This command is used to read the selective suspend setting, the firmware requirement are illustrate as follow:

- ACR1581U-C FW 1.09.00 or above

Command

Command	Class	INS	P1	P2	Lc
Get Selective Suspend Setting	E0h	00h	00h	E5h	00h

Response Code

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

6.1.13. Set Selective Suspend Setting [E0 00 00 E5 01 ...]

This command is used to set the selective suspend setting, the firmware requirement is as same as Get Selective Suspend Setting.

Notice: Selective suspend can not be enable while the reader is in keyboard emulation mode and vice versa.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set Selective Suspend Setting	E0h	00h	00h	E5h	01h	Selective Suspend

Response Code

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

Selective Suspend - 1 Byte



Parameter	Description
00h	Disable
01h	Enable

Default Setting – 00h

6.1.14. Escape Command for PICC – HID Keyboard

6.1.14.1. Get Output Format [E0 00 00 90 00]

This command is used to get output format.

Command

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

Response Code

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

6.1.14.2. Set Output Format [E0 00 00 90 02 ...]

This command is used to set output format.

For Output Order 02h and 04h are available for the following firmware:

- ACR1581U-C FW 1.09.00 or above

Command

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

Response Code

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

Output Format: 1 Byte

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



Output Order: 1 Byte

Status	Description
00h	Default setting (UID Byte 0, UID Byte 1 ... UID Byte N) Example: aa cc bb dd (original /actual UID order)
01h	Reverse all card type order (UID Byte N, UID Byte N-1 ... UID Byte 0) Example: dd bb cc aa (reverse the original / actual UID order)
02h	Reverse ISO 14443 and Felica only Example: dd bb cc aa (reverse the selected Card Type original / actual UID order)
04h	Reverse ISO 15693 only Example: dd bb cc aa (reverse the selected Card Type original / actual UID order)

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

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

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

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



Status (From bit 7~4)	Description (Don't care about x bit)
9h	6H-10D
Ah	8H-10D
Bh	10H-14D
Ch	2H4H-8D
Dh	14H-17D

Default Setting – Output Format: 30h

Output Order: 00h

6.1.14.3. Get Character at Start, Between, at End UID [E0 00 00 91 00]

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

Command

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

Response Code

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

6.1.14.4. Set Character at Start, Between, at End UID [E0 00 00 91 03 ...]

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

Command

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

Response Code

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

Between: 1 Byte (The character between each UID)

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



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

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

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

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

Notes:

- only the characters “,” “,” “,” “,” “-” are supported in the AZERTY keyboard layout for the characters in between. Zero (0) and Backspace are NOT supported.

6.1.14.5. Get Keyboard Layout Language [E0 00 00 92 00]

This command is used to get keyboard layout language.

Command

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

Response Code

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

6.1.14.6. Set Keyboard Layout Language [E0 00 00 92 01 ...]

This command is used to set keyboard layout language.

Command

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

Response Code

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



Keyboard Layout Language: 1 Byte

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

Default Setting – 00h (English)

6.1.14.7. Get Host Interface [E0 00 00 93 00]

This command is used to get host interface

Command

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

Response Code

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

6.1.14.8. Set Host Interface [E0 00 00 93 01 ...]

This command is used to set host interface command

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

Response Code

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

Host Interface: 1 Byte

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

Default Setting – 01h (Only CCID Reader)



6.1.14.9. Get On-Screen Keyboard Setting[E0 00 00 94 00]

This command is used to get on-screen keyboard behaviour, the firmware requirement are illustrate as follow:

- ACR1581U-C FW 1.09.00 or above

Command

Command	Class	INS	P1	P2	Lc
Get On-Screen Keyboard Setting	E0h	00h	00h	94h	00h

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	On-Screen Keyboard Setting

6.1.14.10. Set On-Screen Keyboard Setting[E0 00 00 94 01 ...]

This command is used to set on-screen keyboard behaviour, the firmware requirement is as same as Get On-Screen Keyboard Setting.

Command

Command	Class	INS	P1	P2	Lc	Data Out
Set On-Screen Keyboard Setting	E0h	00h	00h	94h	01h	On-Screen Keyboard Setting

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	On-Screen Keyboard Setting

On-Screen Keyboard Setting: 1 Byte

Status	Description
00h	Disable Keyboard Popup
01h	Enable Keyboard Popup

Default Setting – 00h (Disable Keyboard Popup)



6.1.15. Escape Command for PICC – Card Emulation

6.1.15.1. Enter Card Emulation Mode [E0 00 00 40 03 ...]

This command is used to set the reader into card emulation mode in order to emulate a a NFC Forum Type 2 Tag or a FeliCa Card.

Note: Lock byte is not supported in emulated NFC Forum Type 2 Tag. UID is user programmable.

Command

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

Response Code

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

NFC Device Mode: 3 Byte

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

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

Byte Number	0	1	2	3	Byte Address access by USB
Serial Number	SN0	SN1	SN2	BCC0	Nil
Reserved	Reserved	Reserved	Reserved	Reserved	Nil
Internal/Lock	Reserved	Internal	Lock0	Lock1	Nil
Data read/write	Data0	Data1	Data2	Data3	0-3
Data read/write	Data4	Data5	Data6	Data7	4-7
Data read/write	Data8	Data9	Data10	Data11	8-11
Data read/write	Data12	Data13	Data14	Data15	12-15
Data read/write	Data16	Data17	Data18	Data19	16-19
Data read/write	Data20	Data21	Data22	Data23	20-23
Data read/write	Data24	Data25	Data26	Data27	24-27
Data read/write	Data28	Data29	Data30	Data31	28-31
Data read/write	Data32	Data33	Data34	Data35	32-35
Data read/write	Data36	Data37	Data38	Data39	36-39
Data read/write	Data40	Data41	Data42	Data43	40-43
Data read/write	Data44	Data45	Data46	Data47	44-47
Data read/write	Data48	Data49	Data50	Data51	48-51
Data read/write	Data52	Data53	Data54	Data55	52-55
Data read/write
Data read/write	Data1984	Data1985	Data1986	Data1987	1984~1987

Accessible area (1988 bytes)

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



Memory	1 Block data (16 Byte)	Byte Address access by USB
Data read/write	Block 0	0-15
Data read/write	Block 1	16-31
Data read/write	Block 2	32-47
Data read/write	Block 3	48-63
Data read/write	Block 4	64-79
Data read/write	Block 5	80-95
Data read/write	Block 6	96-111
Data read/write	Block 7	112-127
Data read/write	Block 8	128-143
Data read/write	Block 9	144-159

Table 8: FeliCa Memory Map (160 bytes)

Where:

Default: Block 0 data: {10h, 01h, 01h, 00h, 09h, 00h, 00h, 00h, 00h, 00h, 01h, 00h, 00h, 00h, 00h, 1Ch}

Default Block 0 data NFC Type3 Tag Attribute Information Block

Notes:

1. *FeliCa card emulation support Read/Write without Encryption*
2. *FeliCa Card Identification Number in IDm is user programmable while Manufacturer Code is fixed at (03 88).*



6.1.15.2. Read Card Emulation Data (NFC Forum Type 2 Tag) [E0 00 00 60 04 ...]

This command is used to read the emulated card content.

Command

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

Response Code

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

Start Offset: 1 Byte – Address start from Data0 in **Table 7**

Length: 1 Byte – No. of byte

6.1.15.3. Write Card Emulation Data (NFC Forum Type 2 Tag) [E0 00 00 60 ...]

This command is used to write the emulated card content.

Command

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

Response Code

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

NFC Device Mode: 1 Byte

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

Start Offset: 1 Byte – Address start from Data0 in **Table 7**

Length: 1 Byte – No. of byte



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

This command is used to read the emulated card content.

Command

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

Response Code

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

Start Offset: 2 Byte – Address start to read from SN0 in **Table 7**

Length: 1 Byte – No. of byte to read

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

This command is used to write the emulated card content.

Command

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

Response Code

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

NFC Device Mode: 1 Byte

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

Start Offset: 2 Byte – Address start to write from SN0 in **Table 7**

Length: 1 Byte – No. of byte to write

6.1.15.6. Set Card Emulation of NFC Forum Type 2 Tag ID [E0 00 00 61 03 ...]

This command sets the UID of the emulated NFC Forum Type 2 Tag.

Command

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

Response Code

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



6.1.15.7. Set Card Emulation Lock Data in NFC [E0 00 00 65 01 ...]

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

Command

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

Response Code

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

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

Operating Parameter	Parameter	Description	Option
Bit 7 ~ 2	Reserved	Reserved	
Bit 1	FeliCa Lock Enable	Data cannot be modified via NFC. The data can still be modified by using the USB escape command.	0: Lock disable 1: Lock enable
Bit 0	NFC Forum Type 2 Tag Enable		

6.1.15.8. Set Card Emulation FeliCa IDm [E0 00 00 64 06 ...]

This command sets the 6-byte FeliCa Card Identification number on emulated FeliCa card.

Command

Command	Class	INS	P1	P2	Lc	Data In
Set Card Emulation FeliCa IDm	E0h	00h	00h	64h	06h	IDm

Response Code

Response	Class	INS	P1	P2	Le	Data Out
Result	E1h	00h	00h	00h	06h	IDm

Where:

IDm 6 bytes.



6.1.15.9. Get Card Emulation Status [E0 00 00 69 00]

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

Command

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

Response Code

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

Status: 1 Byte

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

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

The command set is to trigger ACS website <https://www.acs.com.hk> by using ACR1581U to emulate as the NFC forum type 2 tag mode. The steps are showed below:

1. Enter the card emulation mode with below command:

- Send Enter Card Emulation Mode

E0 00 00 40 03 02 00 00

2. Write the NDEF data with below command:

- Send Write Card Emulation Data (NFC Forum Type 2 Tag)

E0 00 00 60 1A 01 02 00 18 E1 10 F4 00 03 0F D1 01 0B 55 02 61 63 73 2E 63 6F 6D 2E 68 6B FE

The command set is to trigger an sample long URL website

<https://www.example.com/this/is/a/very/long/url/that/keeps/going/on/and/on/with/even/more/segments/added/to/make/sure/it/exceeds/the/typical/length/limit/of/260/bytes/which/is/surprisingly/easy/to/do/if/you/keep/adding/more/and/more/segments/like/this/one/and/even/more>

by using ACR1581U to emulate as the NFC forum type 2 tag mode. The steps are showed below:

1. Enter the card emulation mode with below command:

- Send Enter Card Emulation Mode

E0 00 00 40 03 02 00 00



2. Write the NDEF data with below command:

- Send Write Card Emulation Data (NFC Forum Type 2 Tag). Since the length of the NDEF message is longer than 256 bytes, it needs to be spilt into two parts to send to the NFC Forum Type 2 Tag.

```
E0 00 00 60 AC 01 02 00 A8 E1 10 F4 00 03 FF 01 09 C1 01 00 00 01 02 55 02 65 78
61 6D 70 6C 65 2E 63 6F 6D 2F 74 68 69 73 2F 69 73 2F 61 2F 76 65 72 79 2F 6C 6F
6E 67 2F 75 72 6C 2F 74 68 61 74 2F 6B 65 65 70 73 2F 67 6F 69 6E 67 2F 6F 6E 2F
61 6E 64 2F 6F 6E 2F 77 69 74 68 2F 65 76 65 6E 2F 6D 6F 72 65 2F 73 65 67 6D 65
6E 74 73 2F 61 64 64 65 64 2F 74 6F 2F 6D 61 6B 65 2F 73 75 72 65 2F 69 74 2F 65
78 63 65 65 64 73 2F 74 68 65 2F 74 79 70 69 63 61 6C 2F 6C 65 6E 67 74 68 2F 6C
69 6D 69 74 2F 6F 66 2F 32 36 30 2F 62 79 74
```

```
E0 00 00 60 6E 01 02 A8 6A 65 73 2F 77 68 69 63 68 2F 69 73 2F 73 75 72 70 72 69
73 69 6E 67 6C 79 2F 65 61 73 79 2F 74 6F 2F 64 6F 2F 69 66 2F 79 6F 75 2F 6B 65
65 70 2F 61 64 64 69 6E 67 2F 6D 6F 72 65 2F 61 6E 64 2F 6D 6F 72 65 2F 73 65 67
6D 65 6E 74 73 2F 6C 69 6B 65 2F 74 68 69 73 2F 6F 6E 65 2F 61 6E 64 2F 65 76 65
6E 2F 6D 6F 72 65 FE
```

Notes:

For more detailed information and specifications related to the NDEF (NFC Data Exchange Format), I would recommend referring to the NDEF specification. It provides comprehensive guidelines and details about the structure and usage of NDEF records, which are commonly used in NFC data exchange. The NDEF specification will provide a deeper understanding of how to interpret and utilize the NDEF command and data in the context of the ACR1581U device.

6.1.16. Escape Command for PICC – Discovery Mode

6.1.16.1. Enter Discovery Mode [E0 00 00 6A 01 ...]

This command is used to enter the discovery mode.

Command

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

Response Code

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

Discovery Mode: 1 Byte

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



6.2. Escape Command for ICC

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

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

Command

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

Response Code

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

6.2.2. Set Exclusive Mode [E0 00 00 2B 01 ...]

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

Command

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

Response Code

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

Exclusive Mode (1 byte)

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

Default Setting – 01h (Exclusive Mode)



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

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

Command

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

Response Code

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

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

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

Command

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

Response Code

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

Card Power Config (1 byte)

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

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



6.3. Escape Command for Peripheral Control and Other

6.3.1. Get Firmware Version [E0 00 00 18 ...]

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

Command

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

Response Code

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

Example:

Command: E0 00 00 18 00

Response Code: E1 00 00 00 12 41 43 52 31 35 38 31 20 46 57 20 31 2E 30 30

Firmware Version in Hex: 41 43 52 31 35 38 31 20 46 57 20 31 2E 30 30

Firmware Version in ASCII: ACR1581 FW 1.00

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

This command is used to get the serial number.

Command

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

Response Code

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

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

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

Command

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

Response Code

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



Enable SN in USB Descriptor (1 byte)

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

Default Setting For FW1.09.00 or below – 00h

Default Setting For FW1.09.01 or above – 01h

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

This command is used to set a single buzzer

Command

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

Response Code

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

Buzzer Status (1 byte)

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

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

This command is used to set period of buzzer

Command

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

Response Code

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



Buzzer Status (3 byte)

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

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

This command is used to get the current LED status

Command

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

Response Code

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

6.3.7. Set LED Control [E0 00 00 29 01 ...]

This command is used to set LED control

Command

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

Response Code

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

LED Status (1 byte)

LED Status	Description
Bit 0: Blue LED	1 = On; 0 = Off
Bit 1: Green LED	1 = On; 0 = Off
Bit 2-7: RFU	Other

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

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



Command

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

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	PICC UI Behaviour

6.3.9. Set UI Behaviour [E0 00 00 21 01 ...]

This command is used to set the PICC UI behaviour.

For UI Behaviour bit 1, bit 2 and bit 5 are available for the following firmware:

- ACR1581U-C FW 1.09.00 or above

Command

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

Response Code

Response	Class	INS	P1	P2	Le	Data In
Result	E1h	00h	00h	00h	01h	PICC UI Behaviour

UI Behaviour - 1 Byte, Bit Mask of following

Operating Parameter	Parameter	Description	Option
Bit 0	Accessing(LED Fast Blinking)	The UI behaviour of the reader	1 = Enable 0 = Disable
Bit 1	PICC Polling Status LED		
Bit 2	PICC Activation Status LED		
Bit 3	Presence Event (Short Buzzer Beep)		
Bit 4	Card Removal Event (Short Buzzer Beep)		
Bit 5	Power Up Event (Short Buzzer Beep)		

Default Setting For PICC – 2Fh

Notes:

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



Appendix A. NDEF Message

This section shows how to use NDEF message to encode the URL onto the Ntag.

For the data format, please refer to NFC Forum NFC Data Exchange Format (NDEF) Specifications 1.0.

Example:

NDEF Message = { D1 01 0B 55 02 61 63 73 2E 63 6F 6D 2E 68 6Bh}

Offset	Content	Length	Description
0	D1	1	NDEF header. TNF = 01h, SR=1, MB=1, ME=1
1	01	1	Record name length (1 byte)
2	0B	1	The length of the URI payload (11 bytes)
3	55 ("U")	1	Record type: "U"
4	02	1	Abbreviation: "https://www."
5	61 63 73 2E 63 6F 6D 2E 68 6B	10	The URL itself. "acs.com.hk"

Encode to Ntag = {03 0F D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6B FEh}

Offset	Content	Length	Description
0	03	1	TLV header. 03h = NDEF message
1	0F	1	The length of the NDEF message (15 byte)
2	D1 01 0B 55 01 61 63 73 2E 63 6F 6D 2E 68 6B	15	NDEF Message
17	FE	1	TLV header. FEh = End of record



Appendix B. ACR1281U Compatibility

This section shows the commands that compatible to ACR1281U, the firmware require 1.08.01 or above.

Appendix B.1. Load Authentication Keys

This command loads 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 was completed successfully.
- = 63 00h means the operation failed.



Appendix B.2. Authentication for MIFARE Classic (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

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 was completed successfully.
= 63 00h means the operation failed.

Appendix B.3. Manual PICC Polling

This command determines if any PICC is within the detection range of the reader. This command can be used if the automatic PICC polling function is disabled.

Manual PICC Polling Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Manual PICC Polling	E0h	00h	00h	22h	01h	0Ah

Manual PICC Polling Response Format (6 bytes)

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

Appendix B.4. Read PICC Operating Parameter

This command checks the current PICC operating parameter.

Read PICC Operating Parameter Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read PICC Operating Parameter	E0h	00h	00h	20h	00h

Read PICC Operating Parameter Response Format (6 bytes)

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

Appendix B.5. Set PICC Operating Parameter

The command sets the PICC operating parameter.

Set PICC Operating Parameter Format (6 bytes)

Command	Class	INS	P1	P2	Lc	Data In
Set PICC Operating Parameter	E0h	00h	00h	20h	01h	Operating Parameter

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

Note: Default value of Behaviors = 03h.

Set PICC Operating Parameter Response Format (6 bytes)

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

Appendix B.6. Initialize Card Insertion Counter

This command initializes the cards insertion/detection counter.

Initialize Card Insertion Counter Format (9 bytes)

Command	Class	INS	P1	P2	Lc	Data In			
Initialize Card Insertion Counter	E0h	00h	00h	09h	04h	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)



Initialize Card Insertion Counter Response Format (5 bytes)

Response	Class	INS	P1	P2	Le
Result	E1h	00h	00h	00h	00h

Appendix B.7. Read Card Insertion Counter

This command checks the cards insertion/detection counter value.

Read Card Insertion Counter Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Read Card Insertion Counter	E0h	00h	00h	09h	00h

Read Card Insertion Counter Response Format (9 bytes)

Response	Class	INS	P1	P2	Le	Data Out			
Result	E1h	00h	00h	00h	04h	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)

Appendix B.8. Update Card Insertion Counter

This command updates the cards insertion/detection counter value.

Update Card Insertion Counter Format (5 bytes)

Command	Class	INS	P1	P2	Lc
Update Card Insertion Counter	E0h	00h	00h	0Ah	00h

Update Card Insertion Counter Response Format (9 bytes)

Response	Class	INS	P1	P2	Le	Data Out			
Result	E1h	00h	00h	00h	04h	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)

Appendix B.9. Compatibility and Migration Notes

As part of product improvement, certain commands supported in ACR1281U have been updated or replaced in ACR1581U.

The table below provides a clear mapping to help customers transition smoothly between the two models.

Most commonly used functions such as LED control and PICC polling remain supported, although some commands may use updated naming or configuration structures.

For any migration or integration questions, our technical team will be happy to assist at info@acs.com.hk.

Note:

The Card Insertion Counter feature available in ACR1281U is not supported in ACR1581U.

Applications relying on this feature may require modification.

ACR1281U API	ACR1281U Command	ACR1581U Supported	ACR1581U Equivalent	Notes
LED Control	E0 00 00 29 01	Yes	<u>Set LED Control [E0 00 00 29 01 ...]</u>	Same concept, different LED colour (Blue/Green instead of Red/Green)
LED Status	E0 00 00 29 00	Yes	<u>Get LED Status [E0 00 00 29 00]</u>	Same as above
Set Default LED and Buzzer Behaviours	E0 00 00 21 01	Yes	<u>Set UI Behaviour [E0 00 00 21 01 ...]</u>	Bit definition changed, remove the exclusive mode buzzer and add Power on buzzer control
Read Default LED and Buzzer Behaviours	E0 00 00 21 00	Yes	<u>Get UI Behaviour [E0 00 00 21 00]</u>	Same as above
Set Automatic PICC Polling	E0 00 00 23 01	Yes	<u>Set Polling/ATR Option [E0 00 00 23 01 ...]</u>	Bit definition modified, remove turn off antenna when PICC inactive and add extra MIFARE type identification for Part 3 card in ATR
Read Automatic PICC Polling	E0 00 00 23 00	Yes	<u>Get Polling/ATR Option [E0 00 00 23 00]</u>	Same as above