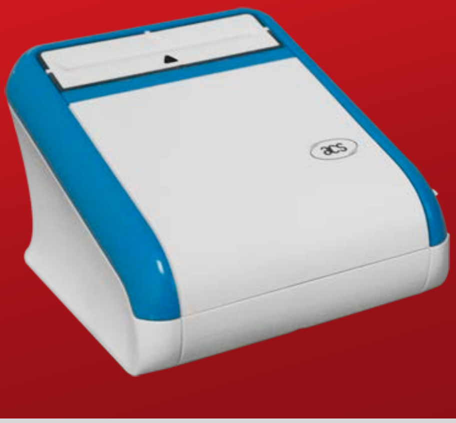
Advanced Card Systems Ltd.
Card & Reader Technologies

ACR33U-A1 SmartDuo

Smart Card Reader



Reference Manual

Subject to change without prior notice

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

Table of Contents

1.0. Introduction 3
2.0. Features 4
3.0. Smart Card Support 5
3.1. MCU Cards 5
3.2. Memory-based Smart Cards....5
4.0. Smart Card Interface 6
4.1. Smart Card Power Supply VCC (C1) 6
4.2. Programming Voltage VPP (C6)....6
4.3. Card Type Selection 6
4.4. Interface for Microcontroller-based Cards 6
4.5. Card Tearing Protection....6
5.0. Power Supply....7
5.1. Status LED....7
6.0. USB Interface....8
6.1. Communication Parameters 8
6.2. Endpoints 8
7.0. Communication Protocol 9
8.0. Commands....11
8.1. CCID Command Pipe Bulk-OUT Messages....11
8.1.1. PC_to_RDR_IccPowerOn....11
8.1.2. PC_to_RDR_IccPowerOff....11
8.1.3. PC_to_RDR_GetSlotStatus 12
8.1.4. PC_to_RDR_XfrBlock 12
8.1.5. PC_to_RDR_GetParameters....13
8.1.6. PC_to_RDR_ResetParameters 13
8.1.7. PC_to_RDR_SetParameters 14
8.1.8. PC_to_RDR_Escape 16
8.2. CCID Bulk-IN Messages....18
8.2.1. RDR_to_PC_DataBlock 18
8.2.2. RDR_to_PC_SlotStatus 18
8.2.3. RDR_to_PC_Parameters....19
8.2.4. Memory Card Access via PC_to_RDR_XfrBlock....22
8.2.5. Memory Card – 1, 2, 4, 8, 16 kbit I2C card 22
8.2.6. RDR_to_PC_Escape 25
8.2.7. Memory Card – 32, 64, 128, 256, 512, 1024 kbit I2C card....26
8.2.8. Memory Card – ATMEL AT88SC153....29
8.2.9. Memory Card – ATMEL AT88SC1608....33
8.2.10. Memory Card – SLE4418/SLE4428/SLE5518/SLE5528 36
8.2.11. Memory Card – SLE4432/SLE4442/SLE5532/SLE5542 41
8.2.12. Memory Card – SLE4406/SLE4436/SLE5536/SLE6636....45
8.2.13. Memory Card – AT88SC101 / AT88SC102 / AT88SC1003 49
Appendix A. CCID Response Error Codes....56

Tables

Table 1: USB Interface Wiring 8
Table 2: ACR33U-A1 SmartDuo Communication Endpoints....8

1.0. Introduction

The ACR33U-A1 SmartDuo PC-Linked Reader acts as an interface for the communication between a computer and a smart card. Different types of smart cards have different commands and different communication protocols, which, in most cases, prevent direct communication between a smart card and a computer. The ACR33U-A1 SmartDuo Smart Card Reader establishes a uniform interface from the computer to the smart card for a wide variety of cards. By taking care of the card's particulars, it releases the computer software programmer from being responsible with smart card operations' technical details, which in many cases, are not relevant to the implementation of a smart card system.

2.0. Features

- USB Full Speed Interface
- Plug-and–Play – CCID support brings utmost compatibility
- Dual Slots for Full-Sized Smart Cards
- 3 SAM (Secure Access Module) Card Slots
- Smart card reader:
    - Supports ISO 7816 Class A (5V) smart cards
    - Reads and writes onto all microprocessor cards with T=0 and T=1 protocol
    - Supports memory cards (Atmel AT88SC153 and AT88SC1608)
    - Supports SLE 4406/18/28/32/36/42, SLE 5518/28/32/36/42, SLE 6636
    - Features Short Circuit Protection
    - Supports PPS (Protocol and Parameters Selection)
- Tri-Color LED (Green, Red, Blue) and Buzzer for Status Indication
- Compliant with the following international standards:
    - PC/SC
    - CCID
    - Microsoft WHQL
    - CE
    - FCC
    - RoHS
    - VCCI

3.0. Smart Card Support

3.1. MCU Cards

The ACR33U-A1 SmartDuo Smart Card Reader is a PC/SC compliant smart card reader that supports ISO7816 Class A (5 V) smart card. It also works with MCU cards following either the T=0 and T=1 protocol.

3.2. Memory-based Smart Cards

The ACR33U-A1 SmartDuo Smart Card Reader works with several memory-based smart cards such as:

- Cards following the I2Cbus protocol (free memory cards) with maximum 128 bytes page with capability, including:

  Atmel: AT24C01/02/04/08/16/32/64/128/256/512/1024

- Cards with secure memory IC with password and authentication, including:

  Atmel: AT88SC153 and AT88SC1608

- Cards with intelligent 1k bytes EEPROM with write-protect function, including:

  Infineon: SLE4418, SLE4428, SLE5518 and SLE5528

- Cards with intelligent 256 bytes EEPROM with write-protect function, including:

  Infineon: SLE4432, SLE4442, SLE5532 and SLE5542

- Cards with ‘104’ type EEPROM non-reloadable token counter cards, including:

  Infineon: SLE4406, SLE4436, SLE5536 and SLE6636

- Cards with Security Logic with Application Zone(s), including:

  Atmel: AT88SC101, AT88SC102 and AT88SC1003

4.0. Smart Card Interface

The interface between the ACR33U-A1 SmartDuo Smart Card Reader and the inserted smart card follows the specifications of ISO 7816-3 with certain restrictions or enhancements to increase the practical functionality of the ACR33U-A1 SmartDuo.

4.1. Smart Card Power Supply VCC (C1)

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

4.2. Programming Voltage VPP (C6)

According to ISO 7816-3, the smart card contact C6 (VPP) supplies the programming voltage to the smart card. Since all common smart cards in the market are EEPROM based and do not require the provision of an external programming voltage, the contact C6 (VPP) has been implemented as a normal control signal in the ACR33U-A1 SmartDuo. The electrical specifications of this contact are identical to those of the signal RST (at contact C2).

4.3. Card Type Selection

The controlling PC must always select the card type through the proper command sent to the ACR33U-A1 SmartDuo prior to activating the inserted card. This includes both the memory cards and MCU-based cards.

For MCU-based cards the reader allows to select the preferred protocol, T=0 or T=1. However, this selection is only accepted and carried out by the reader through the PPS when the card inserted in the reader supports both protocol types. Whenever 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.

4.4. Interface for Microcontroller-based Cards

For microcontroller-based smart cards only the contacts C1 (VCC), C2 (RST), C3 (CLK), C5 (GND) and C7 (I/O) are used. A frequency of 4 MHz is applied to the CLK signal (C3).

4.5. Card Tearing Protection

The ACR33U-A1 SmartDuo provides a mechanism to protect the inserted card when it is suddenly withdrawn while it is powered up. The power supply to the card and the signal lines between the ACR33U-A1 SmartDuo and the card is immediately deactivated when the card is removed. However, as a rule to avoid any electrical damage, a card should only be removed from the reader while it is powered down.

> **Note:** ACR33U-A1 SmartDuo does not switch on the power supply to the inserted card by itself. This can be done by the controlling computer through the proper command sent to the reader.

5.0. Power Supply

The ACR33U-A1 SmartDuo requires a voltage of 5 VDC, 100 mA, regulated, power supply. The ACR33U-A1 SmartDuo gets the power supply from PC (through the cable supplied along with each type of reader).

5.1. Status LED

The LED indicates the activation status of the smart card interface:

- **Flashing slowly (turns on 200 ms every 2 seconds)**
  Indicates ACR33U-A1 SmartDuo is powered up and in the standby state. Either the smart card has not been inserted or the smart card has not been powered up (if it is inserted).
- **Lighting up**
  Indicates power supply to the smart card is switched on, i.e., the smart card is activated.
- **Flashing quickly**
  Indicates there are communications between ACR33U-A1 SmartDuo and smart card.

The different LED colors indicate the different states of the ACR33U-A1 SmartDuo, where:

- **Red LED**
  Power status
- **Green LED**
  Main card slot status
- **Blue LED**
  Slave card slot status

6.0. USB Interface

The ACR33U-A1 SmartDuo is connected to a computer through a USB following the USB standard.

6.1. Communication Parameters

The ACR33U-A1 SmartDuo is connected to a computer through USB as specified in the USB Specification 2.0. The ACR33U-A1 SmartDuo is working in full speed mode, i.e. 12 Mbps.

Pin	Signal	Function
1	V_{BUS}	+5 V power supply for the reader
2	D-	Differential signal transmits data between ACR33U-A1 and PC.
3	D+	Differential signal transmits data between ACR33U-A1 and PC.
4	GND	Reference voltage level for power supply

Table 1: USB Interface Wiring

Note: In order for the ACR33U-A1 SmartDuo to function properly through USB interface, either ACS proprietary device driver or ACS PC/SC device driver has to be installed.

6.2. Endpoints

The ACR33U-A1 SmartDuo uses the following endpoints to communicate with the host computer:

Control Endpoint	For setup and control purpose
Bulk OUT	For command to be sent from host to ACR33U-A1 (data packet size is 64 bytes)
Bulk IN	For response to be sent from ACR33U-A1 to host (data packet size is 64 bytes)
Interrupt IN	For card status message to sent from ACR33U-A1 to host (data packet size is 8 bytes)

Table 2: ACR33U-A1 SmartDuo Communication Endpoints

7.0. Communication Protocol

ACR33U-A1 SmartDuo shall interface with the host thru the USB connection. A specification, namely CCID, has been released within the industry defining such a protocol for the USB chip-card interface devices. CCID covers all the protocols required for operating smart cards.

The configurations and usage of USB endpoints on ACR33U-A1 SmartDuo shall follow CCID section 3. An overview is summarized below:

1. *Control Commands* are sent on control pipe (default pipe). These include class-specific requests and USB standard requests. Commands that are sent on the default pipe report information back to the host on the default pipe.
2. *CCID Events* are sent on the interrupt pipe.
3. *CCID Commands* are sent on BULK-OUT endpoint. Each command sent to ACR33U-A1 SmartDuo has an associated ending response. Some commands can also have intermediate responses.
4. *CCID Responses* are sent on BULK-IN endpoint. All commands sent to ACR33U-A1 SmartDuo have to be sent synchronously. (i.e. bMaxCCIDBusySlots is equal to 01h for ACR33U-A1 SmartDuo)

The supported CCID features by ACR33 are indicated in its Class Descriptor:

Offset	Field	Size	Value	Description
0	`bLength`	1	`36h`	Size of this descriptor, in bytes
1	`bDescriptorType`	1	`21h`	CCID Functional Descriptor type
2	`bcdCCID`	2	`0100h`	CCID Specification Release Number in Binary-Coded decimal
4	`bMaxSlotIndex`	1	`05h`	2 big slots and 4 SAM slots are available on ACR33U-A1
5	`bVoltageSupport`	1	`01h`	ACR33U-A1 can supply 5.0V to its slot
6	`dwProtocols`	4	`00000003h`	ACR33U-A1 supports T=0 and T=1 Protocol
10	`dwDefaultClock`	4	`00000FA0h`	Default ICC clock frequency is 4 MHz
14	`dwMaximumClock`	4	`00000FA0h`	Maximum supported ICC clock frequency is 4 MHz
18	`bNumClockSupported`	1	`00h`	Does not support manual setting of clock frequency
19	`dwDataRate`	4	`00002A00h`	Default ICC I/O data rate is 10752 bps
23	`dwMaxDataRate`	4	`00054024h`	Maximum supported ICC I/O data rate is 344100 bps

Offset	Field	Size	Value	Description
27	bNumDataRatesSupported	1	00h	Does not support manual setting of data rates
28	dwMaxIFSD	4	00000Feh	Maximum IFSD supported by ACR33U-A1 for protocol T=1 is 254
32	dwSynchProtocols	4	00000000h	ACR33U-A1 does not support synchronous card
36	dwMechanical	4	00000000h	ACR33U-A1 does not support special mechanical characteristics
40	dwFeatures	4	000204B0h	ACR33U-A1 supports the following features: • Automatic ICC clock frequency change according to parameters • Automatic baud rate change according to frequency and FI,DI parameters • Automatic PPS made by the CCID according to the active parameters • Automatic IFSD exchange as first exchange (T=1 protocol in use) • Short APDU level exchange with CCID
44	dwMaxCCIDMessageLength	4	0000010Fh	Maximum message length accepted by ACR33U-A1 is 271 bytes
48	bClassGetResponse	1	00h	Insignificant for TPDU level exchanges
49	bClassEnvelope	1	00h	Insignificant for TPDU level exchanges
50	wLCDLayout	2	0000h	No LCD
52	bPINSupport	1	03h	With PIN Verification and Modification
53	bMaxCCIDBusySlots	1	01h	Only 1 slot can be simultaneously busy

8.0. Commands

8.1. CCID Command Pipe Bulk-OUT Messages

ACR33U-A1 SmartDuo shall follow the CCID Bulk-OUT Messages as specified in CCID section 4. In addition, this specification defines some extended commands for operating additional features. This section lists the CCID Bulk-OUT Messages to be supported by ACR33U-A1 SmartDuo.

8.1.1. PC_to_RDR_IccPowerOn

Activate the card slot and return ATR from the card.

Offset	Field	Size	Value	Description
0	bMessageType	1	62h	
1	dwLength	4	00000000h	Size of extra bytes of this message
2	bSlot	1	00-05h	Identifies the slot number for this command
5	bSeq	1	00-FFh	Sequence number for command
6	bPowerSelect	1	01h	Voltage that is applied to the ICC 01h – 5 volts
7	abRFU	2		Reserved for future use

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

8.1.2. PC_to_RDR_IccPowerOff

Deactivate the card slot.

Offset	Field	Size	Value	Description
0	bMessageType	1	63h	
1	dwLength	4	00000000h	Size of extra bytes of this message
5	bSlot	1	00-05h	Identifies the slot number for this command
6	bSeq	1	00-FFh	Sequence number for command
7	abRFU	3		Reserved for future use

The response to this message is the RDR_to_PC_SlotStatus message.

8.1.3. PC_to_RDR_GetSlotStatus

Get current status of the slot.

Offset	Field	Size	Value	Description
0	bMessageType	1	65h	
1	dwLength	4	00000000h	Size of extra bytes of this message
5	bSlot	1	00-05h	Identifies the slot number for this command
6	bSeq	1	00-FFh	Sequence number for command
7	abRFU	3		Reserved for future use

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

8.1.4. PC_to_RDR_XfrBlock

Transfer data block to the ICC.

Offset	Field	Size	Value	Description
0	bMessageType	1	6Fh	
1	dwLength	4		Size of abData field of this message
5	bSlot	1	00-05h	Identifies the slot number for this command
6	bSeq	1	00-FFh	Sequence number for command
7	bBWI	1	00-FFh	Used to extend the CCIDs Block Waiting Timeout for this current transfer. The CCID will timeout the block after “this number multiplied by the Block Waiting Time” has expired.
8	wLevelParameter	2	0000h	RFU (TPDU exchange level)
10	abData	Byte array		Data block sent to the CCID. Data is sent “as is” to the ICC (TPDU exchange level)

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

8.1.5. PC_to_RDR_GetParameters

Get slot parameters.

Offset	Field	Size	Value	Description
0	bMessageType	1	6Ch	
1	DwLength	4	00000000h	Size of extra bytes of this message
5	BSlot	1	00-05h	Identifies the slot number for this command
6	BSeq	1	00-FFh	Sequence number for command
7	AbRFU	3		Reserved for future use

The response to this message is the `RDR_to_PC_Parameters` message.

8.1.6. PC_to_RDR_ResetParameters

Reset slot parameters to default value.

Offset	Field	Size	Value	Description
0	bMessageType	1	6Dh	
1	DwLength	4	00000000h	Size of extra bytes of this message
5	BSlot	1	00-05h	Identifies the slot number for this command
6	BSeq	1	00-FFh	Sequence number for command
7	AbRFU	3		Reserved for future use

The response to this message is the `RDR_to_PC_Parameters` message.

8.1.7. PC_to_RDR_SetParameters

Set slot parameters.

Offset	Field	Size	Value	Description
0	bMessageType	1	61h	
1	dwLength	4		Size of extra bytes of this message
5	bSlot	1	00-05h	Identifies the slot number for this command
6	bSeq	1	00-FFh	Sequence number for command
7	bProtocolNum	1	00h, 01h	Specifies what protocol data structure follows. 00h = Structure for protocol T=0 01h = Structure for protocol T=1 The following values are reserved for future use: 80h = Structure for 2-wire protocol 81h = Structure for 3-wire protocol 82h = Structure for I2C protocol
8	abRFU	2		Reserved for future use
10	abProtocolDataStructure	Byte array		Protocol Data Structure

Protocol Data Structure for Protocol T=0 (dwLength=00000005h)

Offset	Field	Size	Value	Description
10	bmFindexDindex	1		B7-4 – FI – Index into the table 7 in ISO/IEC 7816-3:1997 selecting a clock rate conversion factor B3-0 – DI - Index into the table 8 in ISO/IEC 7816-3:1997 selecting a baud rate conversion factor
11	bmTCCKST0	1		B0 – 0b, B7-2 – 000000b B1 – Convention used (b1=0 for direct, b1=1 for inverse) Note: The CCID ignores this bit.
12	bGuardTimeT0	1	00-FFh	Extra Guardtime between two characters. Add 0 to 254 etu to the normal guardtime of 12etu. FFh is the same as 00h.

Offset	Field	Size	Value	Description
13	bWaitingIntegerT0	1	00-FFh	WI for T=0 used to define WWT
14	bClockStop	1	00-03h	ICC Clock Stop Support 00h = Stopping the Clock is not allowed 01h = Stop with Clock signal Low 02h = Stop with Clock signal High 03h = Stop with Clock either High or Low

Protocol Data Structure for Protocol T=1 (dwLength=00000007h)

Offset	Field	Size	Value	Description
10	bmFindexDindex	1		B7-4 – FI – Index into the table 7 in ISO/IEC 7816-3:1997 selecting a clock rate conversion factor B3-0 – DI - Index into the table 8 in ISO/IEC 7816-3:1997 selecting a baud rate conversion factor
11	BmTCCKST1	1		B7-2 – 000100b B0 – Checksum type (b0=0 for LRC, b0=1 for CRC B1 – Convention used (b1=0 for direct, b1=1 for inverse) Note: The CCID ignores this bit.
12	BGuardTimeT1	1	00-FFh	Extra Guardtime (0 to 254 etu between two characters). If value is FFh, then guardtime is reduced by 1 etu.

13	BwaitingIntegerT1	1	00-9Fh	B7-4 = BWI values 0-9 valid
				B3-0 = CWI values 0-Fh valid

14	bClockStop	1	00-03h	ICC Clock Stop Support 00h = Stopping the Clock is not allowed 01h = Stop with Clock signal Low 02h = Stop with Clock signal High 03h = Stop with Clock either High or Low
15	bIFSC	1	00-FFh	Size of negotiated IFSC
16	bNadValue	1	00h	Only support NAD = 00h

The response to this message is the RDR_to_PC_Parameters message.

8.1.8. PC_to_RDR_Escape

Define and access extended features.

Offset	Field	Size	Value	Description
0	bMessageType	1	6Bh	
1	dwLength	4		Size of abData field of this message
5	bSlot	1	00-05h	Identifies the slot number for this command
6	bSeq	1	00-FFh	Sequence number for command
7	abRFU	3		Reserved for future use
10	abData	Byte array		Data block sent to the CCID

LED

Offset	Field	Size	Value	Description
10	bcmdCode	1	01h	
11	wcmdLength	2	0001h	
13	abRFU	2		Reserved for future use
15	abData	1	00000 XYZb	00000xxx for 3 led, XYZ: 000 => 3 leds off XYZ: 001 => led1 on, green for 1 Sec XYZ: 010 => led 2 on, red for 1 Sec XYZ: 100 => led 3 on, blue for 1 Sec

BUZZER

Offset	Field	Size	Value	Description
10	bcmdCode	1	08h	
11	wcmdLength	2	0001h	
13	abRFU	2		Reserved for future use
15	abData	1	XXh	XXh for buzzer on or off, YZ: 5A => buzzer on for 1 Sec, conform to will's version YZ: A5 => buzzer off

Get Firmware Version

Offset	Field	Size	Value	Description
10	bcmdCode	1	04h	
11	wcmdLength	2	0000h	
13	abRFU	2		Reserved for future use

The response to this command message is the `RDR_to_PC_Escape` response message.

8.2. CCID Bulk-IN Messages

The Bulk-IN messages are used in response to the Bulk-OUT messages. ACR33U-A1 SmartDuo shall follow the CCID Bulk-IN Messages as specified in CCID section 4. This section lists the CCID Bulk-IN Messages to be supported by ACR33U-A1 SmartDuo.

Note: The values of bSlot and bSeq are the same as Bulk-out message.

8.2.1. RDR_to_PC_DataBlock

This message is sent by ACR33U-A1 SmartDuo in response to the command message:

PC_to_RDR_IccPowerOn and PC_to_RDR_XfrBlock.

Offset	Field	Size	Value	Description
0	bMessageType	1	80h	Indicates that a data block is being sent from the CCID
1	dwLength	4		Size of extra bytes of this message
5	bSlot	1		Same value as in Bulk-OUT message
6	bSeq	1		Same value as in Bulk-OUT message
7	bStatus	1		Slot status register as defined in CCID section 4.2.1
8	bError	1		Slot error register as defined in CCID section 4.2.1
9	bChainParameter	1	00h	RFU (TPDU exchange level)
10	abData	Byte array		This field contains the data returned by the CCID

8.2.2. RDR_to_PC_SlotStatus

This message is sent by ACR33U-A1 SmartDuo in response to `PC_to_RDR_IccPowerOff`, `PC_to_RDR_GetSlotStatus`, `PC_to_RDR_Abort` messages and Class specific ABORT request.

Offset	Field	Size	Value	Description
0	bMessageType	1	81h	
1	dwLength	4	00000000h	Size of extra bytes of this message
5	bSlot	1		Same value as in Bulk-OUT message
6	bSeq	1		Same value as in Bulk-OUT message
7	bStatus	1		Slot status register as defined in CCID section 4.2.1

Offset	Field	Size	Value	Description
8	bError	1		Slot error register as defined in CCID section 4.2.1
9	bClockStatus	1		value = 00h Clock running 01h Clock stopped in state L 02h Clock stopped in state H 03h Clock stopped in an unknown state All other values are RFU.

8.2.3. RDR_to_PC_Parameters

This message is sent by ACR33U-A1 SmartDuo in response to `PC_to_RDR_GetParameters`, `PC_to_RDR_ResetParameters` and `PC_to_RDR_SetParameters` messages.

Offset	Field	Size	Value	Description
0	bMessageType	1	82h	
1	dwLength	4		Size of extra bytes of this message
5	bSlot	1		Same value as in Bulk-OUT message
6	bSeq	1		Same value as in Bulk-OUT message
7	bStatus	1		Slot status register as defined in CCID section 4.2.1
8	bError	1		Slot error register as defined in CCID section 4.2.1
9	bProtocolNum	1		Specifies what protocol data structure follows. 00h = Structure for protocol T=0 01h = Structure for protocol T=1 The following values are reserved for future use. 80h = Structure for 2-wire protocol 81h = Structure for 3-wire protocol 82h = Structure for I2C protocol
10	abProtocolDataStructure	Byte array		Protocol Data Structure as summarized in CCID section 10.1.6

Protocol Data Structure for Protocol T=0 (bProtocolNum=0, dwLength=00000005h)

Offset	Field	Size	Value	Description
10	bmFindexDindex	1		B7-4 – FI – Index into the table 7 in ISO/IEC 7816-3:1997 selecting a clock rate conversion factor B3-0 – DI - Index into the table 8 in ISO/IEC 7816-3:1997 selecting a baud rate conversion factor
11	bmTCCKST0	1	00h, 02h	For T=0, B0 – 0b, B7-2 – 000000b B1 – Convention used (b1=0 for direct, b1=1 for inverse)
12	bGuardTimeT0	1	00-FFh	Extra Guardtime between two characters. Add 0 to 254 etu to the normal guardtime of 12etu. FFh is the same as 00h.
13	bWaitingIntegerT0	1	00-FFh	WI for T=0 used to define WWT
14	bClockStop	1	00-03h	ICC Clock Stop Support 00h = Stopping the Clock is not allowed 01h = Stop with Clock signal Low 02h = Stop with Clock signal High 03h = Stop with Clock either High or Low

Protocol Data Structure for Protocol T=1 (bProtocolNum=1, dwLength=00000007h)

Offset	Field	Size	Value	Description
10	bmFindexDindex	1		B7-4 – FI – Index into the table 7 in ISO/IEC 7816-3:1997 selecting a clock rate conversion factor B3-0 – DI - Index into the table 8 in ISO/IEC 7816-3:1997 selecting a baud rate conversion factor
11	BmTCCKST1	1	10h, 11h, 12h, 13h	For T-1, B7-2 – 000100b B0 – Checksum type (b0=0 for LRC, b0=1 for CRC B1 – Convention used (b1=0 for direct, b1=1 for inverse)
12	BGuardTimeT1	1	00-FFh	Extra Guardtime (0 to 254 etu between two characters). If value is FFh, then guardtime is reduced by 1.

13	BwaitingIntegerT1	1	00-9Fh	B7-4 = BWI
				B3-0 = CWI

14	bClockStop	1	00-03h	ICC Clock Stop Support 00h = Stopping the Clock is not allowed 01h = Stop with Clock signal Low 02h = Stop with Clock signal High 03h = Stop with Clock either High or Low
15	bIFSC	1	00-FFh	Size of negotiated IFSC
16	bNadValue	1	00h	Only support NAD = 00h

8.2.4. Memory Card Access via PC_to_RDR_XfrBlock

Memory cards can be accessed via `PC_to_RDR_XfrBlock` command. All memory card functions are mapped into pseudo-APDUs.

8.2.5. Memory Card – 1, 2, 4, 8, 16 kbit I2C card

8.2.5.1. SELECT_CARD_TYPE

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

> **NOTE**: This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF H	A4 H	00 H	00 H	01 H	01 H

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.5.2. SELECT_PAGE_SIZE

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Page size
FF H	01 H	00 H	00 H	01 H	

Page size = 03 H for 8-byte page write

= 04 H for 16-byte page write

= 05 H for 32-byte page write

= 06 H for 64-byte page write

= 07 H for 128-byte page write

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.5.3. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF H	B0 H			

Byte Address Memory address location of the memory card.

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE x Data read from memory card

SW1, SW2 = 90$_H$ 00$_H$ if no error

8.2.5.4. WRITE_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	....	....	Byte n
		MSB	LSB					
FF$_H$	D0$_H$							

Byte Address Memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

Byte x Data to be written to the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90$_H$ 00$_H$ if no error

8.2.6. RDR_to_PC_Escape

This message is sent by ACR33U-A1 SmartDuo in response to `PC_to_RDR_Escape` messages.

Offset	Field	Size	Value	Description
0	bMessageType	1	83h	
1	dwLength	4		Size of abData field of this message
5	bSlot	1		Same value as in Bulk-OUT message
6	bSeq	1		Same value as in Bulk-OUT message
7	bStatus	1		Slot status register as defined in CCID section 4.2.1
8	bError	1		Slot error register as defined in CCID section 4.2.1
9	bRFU	1	00h	RFU (TPDU exchange level)
10	abData	Byte array		This field contains the data returned by the CCID

LED

Offset	Field	Size	Value	Description
10	bcmdCode	1	81h	
11	wcmdLength	2	0000h	
13	abStatus	2	00XXh	XXh for SW2: 00 : success 01 : bad parameter

BUZZER

Offset	Field	Size	Value	Description
10	bcmdCode	1	88h	
11	wcmdLength	2	0000h	
13	abStatus	2	00XXh	XXh: 00 : success 01 : bad parameter

Get Firmware Version

Offset	Field	Size	Value	Description
10	bcmdCode	1	88h	
11	wcmdLength	2	0000h	
13	abStatus	2	00XXh	XXh: 00 : success 01 : bad parameter
15	abData	4	001Ch	0x30 0x30 0x31 0x43: 001C

8.2.7. Memory Card – 32, 64, 128, 256, 512, 1024 kbit I2C card

8.2.7.1. SELECT_CARD_TYPE

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

Note: This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF H	A4 H	00 H	00 H	01 H	02 H

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.7.2. SELECT_PAGE_SIZE

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Page size
FF H	01 H	00 H	00 H	01 H	

Data TPDU to be sent to the card

Page size = 03 H for 8-byte page write

= 04 H for 16-byte page write

= 05 H for 32-byte page write

= 06 H for 64-byte page write

= 07 H for 128-byte page write

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.7.3. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF H				

INS = B0 H for 32,64,128,256,512kbit iic card

= 1011 000* b for 1024kbit iic card, where * is the MSB of the 17 bit addressing

Byte Address Memory address location of the memory card.

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE x Data read from memory card

SW1, SW2 = 90 H 00 H if no error

8.2.7.4. WRITE_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	....	....	Byte n
		MSB	LSB					
FF H								

INS = D0 H for 32,64,128,256,512kbit iic card

= 1101 000* b for 1024kbit iic card, where * is the MSB of the 17 bit addressing

Byte Address Memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

Byte x Data to be written to the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.8. Memory Card – ATMEL AT88SC153

8.2.8.1. SELECT_CARD_TYPE

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

> **Note:** This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF H	A4 H	00 H	00 H	01 H	03 H

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.8.2. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF H		00 H		

INS = B0 H for reading zone 00 b

= B1 H for reading zone 01 b

= B2 H for reading zone 10 b

= B3 H for reading zone 11 b

= B4 H for reading fuse

Byte Address Memory address location of the memory card.

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE x Data read from memory card

SW1, SW2 = 90 H 00 H if no error

8.2.8.3. WRITE_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	P1	Bye Address	MEM_L	Byte 1	....	....	Byte n
FF H		00 H						

INS = D0 H for writing zone 00 b

= D1 H for writing zone 01 b

= D2 H for writing zone 10 b

= D3 H for writing zone 11 b

= D4 H for writing fuse

Byte Address Memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

MEM_D Data to be written to the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.8.4. VERIFY_PASSWORD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU							
CLA	INS	P1	P2	Lc	Pw(0)	Pw(1)	Pw(2)
FF $_H$	20 $_H$	00 $_H$		03 $_H$			

Pw(0),Pw(1),Pw(2) Passwords to be sent to memory card.

P2 = 0000 00rp $_b$

where the two bits "rp" indicate the password to compare

r = 0: Write password,

r = 1: Read password,

p = Password set number,

rp = 01 for the secure code.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2 ErrorCnt
90 $_H$	

SW1 = 90 $_H$

SW2 (ErrorCnt) = Error Counter. FF$_H$ indicates the verification is correct. 00$_H$ indicates the password is locked (exceeded the maximum number of retries). Other values indicate the current verification has failed.

8.2.8.5. INITIALIZE_AUTHENTICATION

Command format (abData field in the PC_to_RDR_XfrBlock)

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

Q(0),Q(1)…Q(7) Host random number, 8 bytes.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.8.6. VERIFY_AUTHENTICATION

Command format (abData field in the PC_to_RDR_XfrBlock)

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

Ch(0),Ch(1)…Ch(7) Host challenge, 8 bytes.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.9. Memory Card – ATMEL AT88SC1608

8.2.9.1. SELECT_CARD_TYPE

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

Note: This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF H	A4 H	00 H	00 H	01 H	04 H

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.9.2. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	Zone Address	Byte Address	MEM_L
FF H				

INS = B0 H for reading user zone

= B1 H for reading configuration zone or reading fuse

Zone Address = 0000 0$A_{10}A_9A_8$ b, where A_{10} is the MSB of zone address

= don't care for reading fuse

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ b is the memory address location of the memory card.

= 1000 0000 b for reading fuse

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE x Data read from memory card

SW1, SW2 = `90` H `00` H if no error

8.2.9.3. WRITE_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	Zone Address	Byte Address	MEM_L	Byte 1	....	....	Byte n
FF H								

INS = `D0` H for writing user zone

= `D1` H for writing configuration zone or writing fuse

Zone Address = 0000 0$A_{10}A_9A_8$ b, where A_{10} is the MSB of zone address

= don't care for writing fuse

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ b is the memory address location of the memory card.

= 1000 0000 b for writing fuse

MEM_L Length of data to be written to the memory card.

Byte x Data to be written to the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = `90` H `00` H if no error

8.2.9.4. VERIFY_PASSWORD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Data			
FF H	20 H	00 H	00 H	04 H	RP	Pw(0)	Pw(1)	Pw(2)

Pw(0),Pw(1),Pw(2) Passwords to be sent to memory card.

RP = 0000 $rp_2p_1p_0$ b

where the four bits "$rp_2p_1p_0$" indicate the password to compare:

r = 0: Write password,

r = 1: Read password,

$p_2p_1p_0$: Password set number.

($rp_2p_1p_0$ = 0111 for the secure code).

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2 ErrorCnt
90 H	

SW1 = 90 H

SW2 (ErrorCnt) = Error Counter. FF_H indicates the verification is correct. 00_H indicates the password is locked (exceeded the maximum number of retries). Other values indicate the current verification has failed.

8.2.9.5. INITIALIZE_AUTHENTICATION

Command format (abData field in the PC_to_RDR_XfrBlock)

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

Byte Address Memory address location of the memory card.

Q(0),Q(1)…Q(7) Host random number, 8 bytes.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.9.6. VERIFY_AUTHENTICATION

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	P1	P2	Lc	Q1(0)	Q1(1)	…	Q1(7)
FF H	82 H	00 H	00 H	08 H				

Byte Address Memory address location of the memory card.

Q1(0),Q1(1)…Q1(7) Host challenge, 8 bytes.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

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

8.2.10.1. SELECT_CARD_TYPE

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

Note: This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF H	A4 H	00 H	00 H	01 H	05 H

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.10.2. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF $_H$	B0 $_H$			

MSB Byte Address = 0000 00A_9A_8 $_b$ is the memory address location of the memory card.

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ $_b$ is the memory address location of the memory card.

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE x Data read from memory card

SW1, SW2 = 90 $_H$ 00 $_H$ if no error

8.2.10.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (only SLE4428 and SLE5528)

To read the presentation error counter for the secret code.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FF $_H$	B1 $_H$	00 $_H$	00 $_H$	03 $_H$

Response data format (abData field in the RDR_to_PC_DataBlock)

ERR CNT	DUMMY 1	DUMMY 2	SW1	SW2

ERRCNT The value of the presentation error counter. FF$_H$ indicates the last verification is correct. 00$_H$ indicates the password is locked (exceeded the maximum number of retries). Other values indicate the last verification has failed.

DUMMY Two bytes dummy data read from the card.

SW1, SW2 = 90 $_H$ 00 $_H$ if no error

8.2.10.4. READ_PROTECTION_BIT

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF $_H$	B2 $_H$			

MSB Byte Address = 0000 00$A_9A_8{}_b$ is the memory address location of the memory card.

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0{}_b$ is the memory address location of the memory card.

MEM_L Length of protection bits to be read from the card, in multiples of 8 bits. Maximum value is 32.

MEM_L = 1 + INT((number of bits-1)/8)
For example, to read eight protection bits starting from memory `0x0010`, the following pseudo-APDU should be issued:

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

Response data format (abData field in the RDR_to_PC_DataBlock)

PROT 1	…	…	PROT L	SW1	SW2

PROT y Bytes containing the protection bits

SW1,SW2 = `90` $_H$ `00` $_H$ if no error

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

PROT 1								PROT 2								….							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

Px is the protection bit of BYTE x in the response data

'0' byte is write protected

'1' byte can be written

8.2.10.5. WRITE_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	....	....	Byte N
		MSB	LSB					
FF $_H$	D0 $_H$							

MSB Byte Address = 0000 00A_9A_8 $_b$ is the memory address location of the memory card.

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ $_b$ is the memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

Byte x Data to be written to the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 $_H$ 00 $_H$ if no error

8.2.10.6. WRITE_PROTECTION_MEMORY_CARD

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	Byte Address		MEM_L	Byte 1	....	....	Byte N
		MSB	LSB					
FF $_H$	D1 $_H$							

MSB Byte Address = 0000 00A_9A_8 $_b$ is the memory address location of the memory card.

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ $_b$ is the memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

Byte x Byte values to be compared with the data in the card starting at Byte Address. BYTE 1 is compared with the data at Byte Address; BYTE N is compared with the data at (Byte Address+N-1).

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = `90` H `00` H if no error

8.2.10.7. PRESENT_CODE_MEMORY_CARD (only SLE 4428 and SLE5528)

To submit the secret code to the memory card to enable the write operation with the SLE4428 and SLE5528 card, the following actions are executed:

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

Command format (abData field in the PC_to_RDR_XfrBlock)

<table>
<tr><th colspan="7">Pseudo-APDU</th></tr>
<tr><th rowspan="2">CLA</th><th rowspan="2">INS</th><th rowspan="2">P1</th><th rowspan="2">P2</th><th rowspan="2">MEM_L</th><th colspan="2">CODE</th></tr>
<tr><th>Byte 1</th><th>Byte 2</th></tr>
<tr><td>FF H</td><td>20 H</td><td>00 H</td><td>00 H</td><td>02 H</td><td></td><td></td></tr>
</table>

CODE Two bytes secret code (PIN)

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2 ErrorCnt
90 H	

SW1 = `90` H

SW2 (ErrorCnt) = Error Counter. `FF`H indicates the verification is correct. `00`H indicates the password is locked (exceeded the maximum number of retries). Other values indicate the current verification has failed.

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

8.2.11.1. SELECT_CARD_TYPE

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

> **Note:** This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF $_H$	A4 $_H$	00 $_H$	00 $_H$	01 $_H$	06 $_H$

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 $_H$ 00 $_H$ if no error

8.2.11.2. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF $_H$	B0 $_H$	00 $_H$		

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ $_b$ is the memory address location of the memory card.

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE x Data read from memory card

SW1, SW2 = 90 $_H$ 00 $_H$ if no error

8.2.11.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (only SLE4442 and SLE5542)

To read the presentation error counter for the secret code.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FF H	B1 H	00 H	00 H	04 H

Response data format (abData field in the RDR_to_PC_DataBlock)

ERR CNT	DUMMY 1	DUMMY 2	DUMMY 3	SW1	SW2

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

DUMMY Three bytes dummy data read from the card.

SW1, SW2 = 90 H 00 H if no error

8.2.11.4. READ_PROTECTION_BITS

To read the protection bits for the first 32 bytes.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FF H	B2 H	00 H	00 H	04 H

Response data format (abData field in the RDR_to_PC_DataBlock)

PROT 1	PROT 2	PROT 3	PROT 4	SW 1	SW 2

PROT y Bytes containing the protection bits from protection memory

SW1, SW2 = 90 H 00 H if no error

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

PROT 1								PROT 2								…							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

Px is the protection bit of BYTE x in the response data

‘0’ byte is write protected

‘1’ byte can be written

8.2.11.5. WRITE_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	....	....	Byte N
FF $_H$	D0 $_H$	00 $_H$						

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$ $_b$ is the memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

Byte x Data to be written to the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 $_H$ 00 $_H$ if no error

8.2.11.6. WRITE_PROTECTION_MEMORY_CARD

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	....	....	Byte N
FF $_H$	D1 $_H$	00 $_H$						

Byte Address = $000A_4$ $A_3A_2A_1A_0$ $_b$ (00 $_H$ to 1F $_H$) is the protection memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

Byte x Byte values to be compared with the data in the card starting at Byte Address. BYTE 1 is compared with the data at Byte Address; BYTE N is compared with the data at (Byte Address+N-1).

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = `90` H `00` H if no error

8.2.11.7. PRESENT_CODE_MEMORY_CARD (only SLE 4442 and SLE5542)

To submit the secret code to the memory card to enable the write operation with the SLE4442 and SLE5542 card, the following actions are executed:

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
`FF` H	`20` H	`00` H	`00` H	`03` H			

CODE Three bytes secret code (PIN)

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2 ErrorCnt
`90` H	

SW1 = `90` H

SW2 (ErrorCnt) = Error Counter. `07`$_H$ indicates the verification is correct. `00`$_H$ indicates the password is locked (exceeded the maximum number of retries). Other values indicate the current verification has failed.

8.2.11.8. CHANGE_CODE_MEMORY_CARD (only SLE 4442 and SLE5542)

To write the specified data as new secret code in the card.

The current secret code must have been presented to the card with the PRESENT_CODE command prior to the execution of this command.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FF H	D2 H	00 H	01 H	03 H			

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

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

8.2.12.1. SELECT_CARD_TYPE

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

Note: This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF H	A4 H	00 H	00 H	01 H	07 H

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.12.2. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF H	B0 H	00 H		

Byte Address = Memory address location of the memory card.

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

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

BYTE x Data read from memory card

SW1, SW2 = 90 H 00 H if no error

8.2.12.3. WRITE_ONE_BYTE_MEMORY_CARD

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

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

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 write mode can therefore only be used for updating the counter value in the card.

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

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

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 write mode can therefore only be used for updating the counter value in the card. Backup bit is enabled to prevent data loss when card tearing occurs.

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

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU						
CLA	INS	P1	Byte Address	MEM_L	MODE	BYTE
FF$_H$	D0$_H$	00$_H$		02$_H$		

Byte Address = Memory address location of the memory card

MODE Specifies the write mode and backup option

00$_H$: write

01$_H$: write with carry

02$_H$: write with backup enabled (SLE4436, SLE5536 and SLE6636 only)

03$_H$: write with carry and with backup enabled (SLE4436, SLE5536 and SLE6636 only)

BYTE Byte value to be written to the card

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90$_H$ 00$_H$ if no error

8.2.12.4. PRESENT_CODE_MEMORY_CARD

To submit the secret code to the memory card to enable the card personalization mode, the following actions are executed:

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

The ACR33U-A1 SmartDuo does not try to erase the presentation counter after the code submission! This must be done by the application software through a separate 'Write with carry' command.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	P1	P2	MEM_L	CODE			
					ADDR	Byte 1	Byte 2	Byte 3
FF$_H$	20$_H$	00$_H$	00$_H$	04$_H$	09$_H$			

ADDR Byte address of the presentation counter in the card

CODE Three bytes secret code (PIN)

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = `90` H `00` H if no error

8.2.12.5. AUTHENTICATE_MEMORY_CARD (SLE4436, SLE5536 and SLE6636 only)

To read a card authentication certificate from a SLE5536 or SLE6636 card, the following actions are executed by the ACR33U-A1 SmartDuo:

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

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

Step 1: Send Authentication Certificate to the Card

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU											
CLA	INS	P1	P2	MEM_L	CODE						
					KEY	CLK_ CNT	Byte 1	Byte 2	… …	Byte 5	Byte 6
FF H	84 H	00 H	00 H	08 H							

KEY Key to be used for the computation of the authentication certificate:

`00`$_H$: key 1 with no cipher block chaining

`01`$_H$: key 2 with no cipher block chaining

`80`$_H$: key 1 with cipher block chaining (SLE5536 and SLE6636 only)

`81`$_H$: key 2 with cipher block chaining (SLE5536 and SLE6636 only)

CLK_CNT Number of CLK pulses to be supplied to the card for the computation of each bit of the authentication certificate. Typical value is 160 clocks (A0$_H$)

BYTE 1...6 Card challenge data

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2
61$_H$	02$_H$

SW1, SW2 = 61$_H$ 02$_H$ if no error, meaning two bytes of authentication data are ready. The authentication data can be retrieved by "Get_Response" command.

Step 2: Get back the Authentication Data (Get_Response)

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FF$_H$	C0$_H$	00$_H$	00$_H$	02$_H$

Response data format (abData field in the RDR_to_PC_DataBlock)

CERT		SW1	SW2

CERT 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$_H$ 00$_H$ if no error

8.2.13. Memory Card – AT88SC101 / AT88SC102 / AT88SC1003

8.2.13.1. SELECT_CARD_TYPE

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

Note: This command can only be used after the logical smart card reader communication has been established using the SCardConnect( ) API. For details of ScardConnect( ) API, please refer to PC/SC specifications.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF$_H$	A4$_H$	00$_H$	00$_H$	01$_H$	09$_H$

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90$_H$ 00$_H$ if no error

8.2.13.2. READ_MEMORY_CARD

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF H	B0 H	00 H		

Byte Address = Memory address location of the memory card.

MEM_L Length of data to be read from the memory card.

Response data format (abData field in the RDR_to_PC_DataBlock)

BYTE 1	…	…	BYTE N	SW1	SW2

BYTE x Data read from memory card

SW1, SW2 = 90 H 00 H if no error

8.2.13.3. WRITE_MEMORY_CARD

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

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	P1	Byte Address	MEM_L	Byte 1	....	....	Byte N
FF H	D0 H	00 H						

Byte Address = Memory address location of the memory card.

MEM_L Length of data to be written to the memory card.

BYTE Byte value to be written to the card

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error

8.2.13.4. ERASE_NON_APPLICATION_ZONE

To erase 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 in Section 8.2.13.5
- `ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE` command as specified in Section 8.2.13.6
- `VERIFY_SECURITY_CODE` commands as specified in Section 8.2.13.7

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
`FF` H	`D2` H	`00` H		`00` H

Byte Address = Memory byte address location of the word to be erased.

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = `90` H `00` H if no error

8.2.13.5. 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:

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU									
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE				
					Byte 1	Byte 2	…….	……	Byte N
FF H	20 H	00 H							

Error Counter LEN Length of presentation error counter in bits. The value should be 0x00 always.

Byte Address Byte address of the Application Zone Key in the card. Please refer to the table below for the correct value.

MEM_L Length of the Erase Key. Please refer to the table below for the correct value.

CODE N bytes of Erase Key

	Byte Address	LEN
AT88SC101: Erase Application Zone with EC function disabled	96 H	04 H
AT88SC102: Erase Application Zone 1	56 H	06 H
AT88SC102: Erase Application Zone 2 with EC2 function disabled	9C H	04 H
AT88SC1003: Erase Application Zone 1	36 H	06 H
AT88SC1003: Erase Application Zone 2 with EC2 function disabled	5C H	04 H
AT88SC1003: Erase Application Zone 3	C0 H	06 H

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error.

Note: After SW1SW2 = 0x9000 has 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 “0xFF”, the previous verification is success.

8.2.13.6. 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 undamaged and in “1” state), the following actions are executed:

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

Command format (abData field in the PC_to_RDR_XfrBlock)

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

Error Counter LEN Length of presentation error counter in bits. The value should be 0x80 always.

Byte Address Byte address of the Application Zone Key in the card.

	Byte Address
AT88SC101	96 H
AT88SC102	9C H
AT88SC1003	5C H

CODE 4 bytes Erase Key

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90 H 00 H if no error.

= 63 H 00 H if there is no more retry chance

Note: After SW1SW2 = 0x9000 has 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 “0xFF”, the previous verification is success.

8.2.13.7. VERIFY_SECURITY_CODE

To submit Security Code (2 bytes) to the inserted card. Security Code is to enable the memory access of the card.

The following actions are executed:

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU						
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
FF$_H$	20$_H$	08$_H$	0A$_H$	02$_H$		

Error Counter LEN Length of presentation error counter in bits.

Byte Address Byte address of the key in the card.

CODE 2 bytes Security Code

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90$_H$ 00$_H$ if no error.

= 63$_H$ 00$_H$ if there is no more retry chance

Note: After SW1SW2 = `0x9000` has 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 "`0xFF`", the previous verification is success.

8.2.13.8. BLOWN_FUSE

To blow the fuse of the inserted card. The fuse can be EC_EN Fuse, EC2EN Fuse, Issuer Fuse or Manufacturer’s Fuse.

NOTE : Blowing of the Fuse is an irreversible process.

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of FUS Pin	State of RST Pin
FF$_H$	05$_H$	00$_H$	00$_H$	04$_H$			01$_H$	00$_H$ or 01$_H$

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

State of FUS Pin State of the FUS pin. Should be 0x01 always.

State of RST Pin 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	05$_H$	80$_H$	01$_H$
	EC_EN Fuse	05$_H$	C9$_H$	01$_H$
	Issuer Fuse	05$_H$	E0$_H$	01$_H$
AT88SC102	Manufacturer Fuse	05$_H$	B0$_H$	01$_H$
	EC2EN Fuse	05$_H$	F9$_H$	01$_H$
	Issuer Fuse	06$_H$	10$_H$	01$_H$
AT88SC1003	Manufacturer Fuse	03$_H$	F8$_H$	00$_H$
	EC2EN Fuse	03$_H$	FC$_H$	00$_H$
	Issuer Fuse	03$_H$	E0$_H$	00$_H$

Response data format (abData field in the RDR_to_PC_DataBlock)

SW1	SW2

SW1, SW2 = 90$_H$ 00$_H$ if no error

Appendix A. CCID Response Error Codes

The following table summarizes the possible error code returned by the ACR33U-A1 SmartDuo:

Error Code	Status
FF_h	SLOTERROR_CMD_ABORTED
FE_h	SLOTERROR_ICC_MUTE
FD_h	SLOTERROR_XFR_PARITY_ERROR
FC_h	SLOTERROR_XFR_OVERRUN
FB_h	SLOTERROR_HW_ERROR
$F8_h$	SLOTERROR_BAD_ATR_TS
$F7_h$	SLOTERROR_BAD_ATR_TCK
$F6_h$	SLOTERROR_ICC_PROTOCOL_NOT_SUPPORTED
$F5_h$	SLOTERROR_ICC_CLASS_NOT_SUPPORTED
$F4_h$	SLOTERROR_PROCEDURE_BYTE_CONFLICE
$F3_h$	SLOTERROR_DEACTIVATED_PROTOCOL
$F2_h$	SLOTERROR_BUSY_WITH_AUTO_SEQUENCE
$E0_h$	SLOTERROR_CMD_SLOT_BUSY