Advanced Card Systems Ltd.
Card & Reader Technologies

ACR38
Smart Card Reader

Reference Manual

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

Table of Contents

Tables

1.0. Introduction

This document contains information regarding the ACR38 with firmware 1.12c and 1.10 using the PC/SC platform. The ACR38 with firmware 1.12c uses CCID interface to communicate with the USB port. CCID refers to the Device Class Specification for USB chip/Smart Card Interface Devices and defines the communication protocol and commands for the USB chip-card interface devices. The ACR38 FW1.12c is backward compatible with the ACR38 FW1.10 for smart card applications using the PC/SC platform and MCU cards.

The ACR38 acts as an interface for the communication between a computer (for example, a PC) and a smart card. Different types of smart cards have different commands and different communication protocols which prevents, in most cases the direct communication between a smart card and a computer. The ACR38 establishes a uniform interface from the computer to the smart card for a wide variety of cards. By taking care of the card specific particulars, it releases the computer software programmer from getting involved with the technical details of the smart card operation, which are not relevant in many cases of the implementation of smart card system.

> **Note**: Although the ACR38 is a true *card reader/writer* as it can read and write data from and to smart cards. The terms *card reader* or *reader* will be used indifferently to refer to the ACR38. These designations are commonly used for this kind of devices. We will also refer to ACR38 with firmware 1.12c as ACR38 FW1.12c while ACR38 with firmware 1.10 will be referred to as ACR38 FW1.10.

2.0. Features

The following are the features of the ACR38:

Features	Firmware 1.10	Firmware 1.12c
1. PC/SC	✓	✓
2. CCID	×	✓
3. WHQL Certified Drivers	✓	✓
4. CE and FCC	✓	✓
5. RoHS	✓	✓
6. EMV Level 1	✓	✓
7. ISO 7816 (Class A, B, C)	✓	✓
8. MCU Card Support (T=0, T=1)	✓	✓
9. Memory Card Support	✓	✓
10. USB Full Speed	✓	✓
11. Short Circuit Protection	✓	✓

3.0. Smart Card Support

3.1. MCU Cards

The ACR38 Series is a family PC/SC compliant smart card readers that support ISO 7816 5V, 3V and 1.8 (Class A, B, and C) smart cards. The ACR38 Series also works with MCU cards following either the T=0 and T=1 protocol.

3.2. Memory-based Smart Cards

The ACR38 Series supports the following memory cards:

Types of Memory Cards	Firmware 1.10	Firmware 1.12c
1. 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 SGS-Thomson: ST14C02C, ST14C04C Gemplus: GFM1K, GFM2K, GFM4K, GFM8K	✓	✓
2. Cards with secure memory IC with password and authentication, including: Atmel: AT88SC153 and AT88SC1608	✓	✓
3. Cards with intelligent 1k bytes EEPROM with write-protect function, including: Infineon: SLE4418, SLE4428, SLE5518 and SLE5528	✓	✓
4. Cards with intelligent 256 bytes EEPROM with write-protect function, including: Infineon: SLE4432, SLE4442, SLE5532 and SLE5542	✓	✓
5. Cards with '104' type EEPROM non-reloadable token counter cards, including: Infineon: SLE4406, SLE4436, SLE5536 and SLE6636	✓	✓
6. Cards with Intelligent 416-Bit EEPROM with internal PIN check, including: Infineon: SLE4404	×	✓
7. Cards with Security Logic with Application Zone(s), including: Atmel: AT88SC101, AT88SC102 and AT88SC1003	×	✓

4.0. Smart Card Interface

The interface between the ACR38 and the inserted smart card follows the specifications of ISO 7816-3 with certain restrictions or enhancements to increase the practical functionality of ACR38.

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 ACR38. 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 has to always select the card type through the proper command sent to the ACR38 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 ACR38 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 ACR38 and the card is immediately deactivated when the card is being removed. As a general rule however, to avoid any electrical damage, a card should only be removed from the reader while it is powered down.

> **Note:** The ACR38 does never by itself switch on the power supply to the inserted card. This must explicitly be done by the controlling computer through the proper command sent to the reader.

5.0. Power Supply

The ACR38 requires a voltage of 5 V DC, 100 mA, regulated, power supply. The ACR38 Smart Card Reader gets power supply from a PC (through the cable supplied along with each type of reader).

5.1. Status LED

The Green LED on the front of the reader indicates the activation status of the smart card interface:

- **Flashing slowly (turns on 200ms for every 2 seconds)**

  Indicates that the ACR38 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 that the power supply to the smart card is switched on, i.e., the smart card is activated.
- **Flashing quickly**

  Indicates there is communication between the ACR38 and a smart card.

6.0. USB Interface

The ACR38 is connected to a computer through a USB following the USB standard.

6.1. Communication Parameters

The ACR38 is connected to a computer through USB as specified in the USB Specification 1.1. The ACR38 is working in full speed mode, i.e. 12 Mbps.

Pin	Signal	Function
1	V_{BUS}	+5V power supply for the reader
2	D-	Differential signal transmits data between ACR38 and PC.
3	D+	Differential signal transmits data between ACR38 and PC.
4	GND	Reference voltage level for power supply

Table 1. USB Interface Wiring

Note: In order for the ACR38 with FW 1.10 to function properly through USB interface, the ACS PC/SC Device Driver has to be installed while for ACR38 FW 1.12c the ACS CCID driver or the Microsoft CCID Driver should be installed.

6.2. Endpoints

The ACR38 uses the following endpoints to communicate with the host computer:

Control Endpoint	For setup and control purposes
Bulk OUT	For command to be sent from host to ACR38 (data packet size is 64 bytes)
Bulk IN	For response to be sent from ACR38 to host (data packet size is 64 bytes)
Interrupt IN	For card status message to be sent from ACR38 to host (data packet size is 8 bytes)

7.0. Communication Protocol

7.1. ACR38 FW1.12c Communication Protocol

The ACR38 with Firmware 1.12c shall interface with the host thru 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 and PIN.

The configurations and usage of USB endpoints on ACR38 FW1.12c 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 ACR38 FW1.12c 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 ACR38 FW1.12c have to be sent synchronously. (i.e. `bMaxCCIDBusySlots` is equal to 1 for ACR38 FW1.12c)

The supported CCID features by ACR38 FW1.12c 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	00h	One slot is available on ACR38 FW1.12c.
5	bVoltageSupport	1	07h	ACR38 FW1.12c can supply 1.8V, 3.0V and 5.0V to its slot.
6	dwProtocols	4	00000003h	ACR38 FW1.12c supports T=0 and T=1 Protocol
10	dwDefaultClock	4	00000FA0h	Default ICC clock frequency is 4MHz
14	dwMaximumClock	4	00000FA0h	Maximum supported ICC clock frequency is 4MHz
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	0001F808h	Maximum supported ICC I/O data rate is 344 kbps
27	bNumDataRatesSupported	1	00h	Does not support manual setting of data rates
28	dwMaxIFSD	4	00000Feh	Maximum IFSD supported by ACR38 FW1.12c for protocol T=1 is 254
32	dwSynchProtocols	4	00000000h	ACR38 FW1.12c does not support synchronous card
36	dwMechanical	4	00000000h	ACR38 FW1.12c does not support special mechanical characteristics
40	dwFeatures	4	00010030h	ACR38 FW1.12c supports the following features: • Automatic ICC clock frequency change according to parameters • Automatic baud rate change according to frequency and FI,DI parameters • TPDU level exchange with ACR38 FW1.12c

44	dwMaxCCIDMessageLength	4	0000010Fh	Maximum message length accepted by ACR38 FW1.12c 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	00h	No PIN Verification
53	bMaxCCIDBusySlots	1	01h	Only 1 slot can be simultaneously busy

7.2. ACR38 FW1.10 Communication Protocol

During normal operation, the ACR38 acts as a slave device with regard to the communication between a computer and the reader. The communication is carried out in the form of successive command-response exchanges. The computer transmits a command to the reader and receives a response from the reader after the command has been executed. A new command can be transmitted to the ACR38 only after the response to the previous command has been received.

There are two cases where the reader transmits data without having received a command from the computer, namely the Reset Message and the Card Status Message.

7.2.1. Command to ACR38

A command consists of six protocol bytes and a variable number of data bytes with the following structure:

Byte 1 2 3 4 5 ... N+4 (N>0)

Header	Instruction	Data Length = N	Data
01H		Data Length N	

Header Always 01H to indicate the start of a command.

Instruction The instruction code of the command to be carried out by the ACR38.

Data Length Number of subsequent data bytes, and is encoded in 2 bytes. The first byte (MSB) and second byte (LSB) represent data length N.

Data Data contents of the command.

For a READ command, for example, the data bytes would specify the start address and the number of bytes to be read. For a WRITE command, the data bytes would specify the start address and the data to be written to the card.

The data bytes can represent values to be written to a card and/or command parameters such as an address, a counter, etc.

Note: Commands are sent from host computer to ACR38 through the BULK OUT endpoint.

7.2.2. Response from ACR38

The response from the ACR38 to any command depends on whether the command has been received by the reader without error (e.g., checksum error).

The response by the ACR38 to a correctly received command consists of three protocol bytes, two status bytes and a variable number of data bytes with the following structure:

Byte 1 2 3 4 5 ... N+4 (N>0)

Header	Status	Data Length = N	Data
01H		Data Length N	

Header Always 01H to indicate the start of the response.

Status	Indicates the command execution status: 00H = command successfully executed Otherwise = error in command data, or command cannot be executed A table listing the possible values of the status byte and the corresponding meaning is given in Appendix B.2.
Data Length	Number of subsequent data bytes, and is encoded in 2 bytes. The first byte (MSB) and second byte (LSB) represent data length N.
Data	Data contents of the command. For a READ_DATA command, for example, the data bytes would contain the contents of the memory addresses read from the card. The data bytes can represent values read from the card and/or status information.

Note: Responses are sent from ACR38 to the host computer through BULK IN endpoint.

7.2.3. Card Status Message

When a card is being inserted into the reader or an inserted card is being removed from the reader while the reader is idle, i.e., not executing a command, the reader transmits a Card Status Message to notify the host computer of the change in the card insertion status.

The Card Status Messages have the following structure and contents:

Card Status Message for Card Insertion

Byte	1	2	3	4
	Header	Status	Data Length	
	01 H	C1 H	00 H	00 H

Card Status Message for Card Removal

Byte	1	2	3	4
	Header	Status	Data Length	
	01 H	C0 H	00 H	00 H

A card status message is transmitted only once for every card insertion or removal event. The reader does not expect an acknowledge signal from the computer. After transmitting a status message, the reader waits for the next command from the computer.

Note: Card status messages are sent from ACR38 to the host computer through `INTERRUPT IN` endpoint.

8.0. ACR38 FW1.12c Commands

8.1.1. CCID Command Pipe Bulk-OUT Messages

The ACR38 FW1.12c 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 ACR38 FW1.12c.

8.1.1.1. 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		Identifies the slot number for this command
5	bSeq	1		Sequence number for command
6	bPowerSelect	1		Voltage that is applied to the ICC 00h – Automatic Voltage Selection 01h – 5 volts 02h – 3 volts
7	abRFU	2		Reserved for future use

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

8.1.1.2. 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		Identifies the slot number for this command
6	bSeq	1		Sequence number for command
7	abRFU	3		Reserved for future use

The response to this message is the RDR_to_PC_SlotStatus message.

8.1.1.3. 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		Identifies the slot number for this command
6	bSeq	1		Sequence number for command
7	abRFU	3		Reserved for future use

The response to this message is the RDR_to_PC_SlotStatus message.

8.1.1.4. 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		Identifies the slot number for this command
6	bSeq	1		Sequence number for command
7	bBWI	1		Used to extend the CCIDs Block Waiting Timeout for this current transfer. The CCID will timeout the block after "this number multiplied by the Block Waiting Time" has expired.
8	wLevelParameter	2	0000h	RFU (TPDU exchange level)
10	abData	Byte array		Data block sent to the CCID. Data is sent "as is" to the ICC (TPDU exchange level)

The response to this message is the RDR_to_PC_DataBlock message.

8.1.1.5. 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		Identifies the slot number for this command
6	BSeq	1		Sequence number for command
7	AbRFU	3		Reserved for future use

The response to this message is the RDR_to_PC_Parameters message.

8.1.1.6. 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		Identifies the slot number for this command
6	BSeq	1		Sequence number for command
7	AbRFU	3		Reserved for future use

The response to this message is the RDR_to_PC_Parameters message.

8.1.1.7. 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		Identifies the slot number for this command
6	bSeq	1		Sequence number for command
7	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
8	abRFU	2		Reserved for future use
10	abProtocolData Structure	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		Extra Guardtime between two characters. Add 0 to 254 etu to the normal guardtime of 12 etu. FFh is the same as 00h.
13	bWaitingInteger T0	1		WI for T=0 used to define WWT
14	bClockStop	1		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		Extra Guardtime (0 to 254 etu between two characters). If value is FFh, then guardtime is reduced by 1 etu.

Offset	Field	Size	Value	Description
13	BwaitingIntegerT1	1		B7-4 = BWI values 0-9 valid B3-0 = CWI values 0-Fh valid
14	bClockStop	1		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		Size of negotiated IFSC
16	bNadValue	1	00h	Only support NAD = 00h

The response to this message is the RDR_to_PC_Parameters message.

8.1.2. CCID Bulk-IN Messages

The Bulk-IN messages are used in response to the Bulk-OUT messages. ACR38 FW1.12c shall follow the CCID Bulk-IN Messages as specified in section 4. This section lists the CCID Bulk-IN Messages to be supported by ACR38 FW1.12c.

8.1.2.1. RDR_to_PC_DataBlock

This message is sent by ACR38 FW1.12c in response to PC_to_RDR_IccPowerOn, PC_to_RDR_XfrBlock and PC_to_RDR_Secure messages.

Offset	Field	Size	Value	Description
0	bMessageType	1	80h	Indicates that a data block is being sent from the CCID
1	dwLength	4		Size of extra bytes of this message
5	bSlot	1		Same value as in Bulk-OUT message
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 and this specification section 5.2.8
9	bChainParameter	1	00h	RFU (TPDU exchange level)
10	abData	Byte array		This field contains the data returned by the CCID

8.1.2.2. RDR_to_PC_SlotStatus

This message is sent by ACR38 FW1.12c 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
8	bError	1		Slot error register as defined in CCID section 4.2.1 and this specification section 5.2.8
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.

8.1.2.3. RDR_to_PC_Parameters

This message is sent by ACR38 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 and this specification section 5.2.8
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	abProtocolData Structure	Byte array		Protocol Data Structure as summarized in section 5.2.3.

8.1.3. Commands Accessed via PC_to_RDR_XfrBlock

8.1.3.1. GET_READER_INFORMATION

This command returns relevant information about the particular ACR38 model and the current operating status such as the firmware revision number, the maximum data length of a command and response, the supported card types, and whether a card is inserted and powered up or not.

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 specification.

Pseudo-APDU				
CLA	INS	P1	P2	Lc
FF $_H$	09 $_H$	00 $_H$	00 $_H$	10 $_H$

Table 2. Command format (abData field in the PC_to_RDR_XfrBlock)

FIRMWARE	MAX_C	MAX_R	C_TYPE	C_SEL	C_STAT

Table 3. Response data format (abData field in the RDR_to_PC_DataBlock)

FIRMWARE 10 bytes data for firmware version

MAX_C The maximum number of command data bytes.

MAX_R The maximum number of data bytes that can be requested to be transmitted in a response.

C_TYPE The card types supported by the ACR38 FW1.12c. This data field is a bitmap with

each bit representing a particular card type. A bit set to '1' means the corresponding card type is supported by the reader and can be selected with the *SELECT_CARD_TYPE* command. The bit assignment is as follows:

Byte	1								2							
card type	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

See Appendix A.1 for the correspondence between these bits and the respective card types.

C_SEL The currently selected card type. A value of 00_H means that no card type has been selected.

C_STAT Indicates whether a card is physically inserted in the reader and whether the card is powered up:

00_H: no card inserted

01_H: card inserted, not powered up

03_H: card powered up

9.0. ACR38 FW1.10 Commands

The commands executed by the ACR38 can generally be divided into two categories, namely Control Commands and Card Commands.

Control Commands are in charge of the internal operation of the ACR38. They do not directly affect the card inserted in the reader and are therefore independent of the selected card type.

Card Commands are directed toward the card inserted in the ACR38. The structure of these commands and the data transmitted in the commands and responses depend on the selected card type.

9.1. Control Commands

9.1.1. GET_ACR_STAT

This command returns relevant information about the particular ACR38 model and the current operating status such as the firmware revision number, the maximum data length of a command and response, the supported card types, and whether a card is inserted and powered up or not.

Command format

Header	Instruction	Data length	
01_H	01_H	00_H	00_H

Response data format

Header	Status	Data length LEN		INTERNAL								MAX_C	MAX_R	C_TYPE		C_SEL	C_STAT
01_H																	

INTERNAL 10 bytes data for internal use only

MAX_C The maximum number of command data bytes.

MAX_R The maximum number of data bytes that can be requested to be transmitted in a response.

C_TYPE The card types supported by the ACR38. This data field is a bitmap with each bit representing a particular card type. A bit set to '1' means the corresponding card type is supported by the reader and can be selected with the `SELECT_CARD_TYPE` command. The bit assignment is as follows:

Byte	1								2							
card type	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

See Appendix B.1 for the correspondence between these bits and the respective card types.

C_SEL The currently selected card type as specified in a previous `SELECT_CARD_TYPE` command. A value of `00H` means that no card type has been selected.

C_STAT Indicates whether a card is physically inserted in the reader and whether the card is powered up:

00_H: no card inserted

01_H: card inserted, not powered up

03_H: card powered up

9.1.2. SELECT_CARD_TYPE

This command sets the required card type. The firmware in ACR38 adjusts the communication protocol between reader and the inserted card according to the selected card type.

Command format

Header	Instruction	Data length		Data
		LEN		TYPE
01 H	02 H	00 H	01 H	

TYPE See Appendix B.1 for the value to be specified in this command for a particular card to be used.

Response data format

Header	Status	Data length	
		LEN	
01_H			

9.1.3. SET_OPTION

This command selects the options for the reader.

Command format

Header	Instruction	Data length		Data
		LEN		Option
01 H	07 H	00 H	01 H	

Option Bit 4 : Select for EMV mode

Specifies whether the reader is in EMV mode

0 : Reader not in EMV mode (default)

1 : Reader in EMV mode

Bit 5 : Select for memory card mode

Specifies whether the reader is in memory card mode

0 : reader not in memory card mode (default)

1 : reader in memory card mode

Bit 0, 1, 2, 3, 6 and 7: Reserved

Response data format

Header	Status	Data length	
		LEN	
01 H			

9.1.4. SET_CARD_PPS

This command sends PPS Request to the smart card. This command should work in pair with `SET_READER_PPS`.

Command format

Header	Instruction	Data length		Data
		LEN		PPS Request
01 H	0A H	MSB	LSB	

LEN Length of PPS request. Typical value is "4"

PPS Request PPS Request to send to the card (Please refer to ISO/IEC 7816-3:1997 Section 7 for details of PPS request)

A typical PPS request to select T=1 protocol and FD=0x94 (62500 baud at 4MHz) is: `0xFF 0x11 0x94 0x7A`

Response data format

Header	Status	Data length		Data				
		LEN						
01 H							…	

9.1.5. SET_READER_PPS

This command sends PPS Response to the reader and asks the reader to switch its protocol and/or speed to communication with the smart card. This command should work in pair with `SET_CARD_PPS`.

Command format

Header	Instruction	Data length		Data
		LEN		PPS Response
01 H	0B H	MSB	LSB	

LEN Length of PPS response; Typical value is "4".

PPS Response PPS Response received from the card (Please refer to ISO/IEC 7816-3:1997 Section 7 for details of PPS response). After the driver or the application validates the PPS Response, it should send the PPS Response to the reader. The reader can then switch the protocol and/or speed.

A typical PPS response should be the same as PPS Request.

Response data format

Header	Status	Data length	
		LEN	
01 H			

9.2. MCU Card Commands

9.2.1. RESET_WITH_5_VOLTS_DEFAULT

This command powers up the card inserted in the card reader and performs a card reset. If the card is powered up when the command is being issued, only a reset of the card is carried out. The power supply to the card is not switched off.

Command format

Header	Instruction	Data length LEN	
01 H	80 H	00 H	00 H

Response data format

Header	Status	Data length LEN		ATR				
01 H							.....	

ATR Answer-To-Reset as transmitted by the card according to ISO7816-3.

Note: ATR is only returned in the ACR38 response if the communication protocol of the card is compatible with the reader, i.e., if the card can be processed by the ACR38. Otherwise, the ACR38 returns an error status and deactivates the smart card interface.

9.2.2. RESET_WITH_SPECIFIC_VOLTAGE

This command powers up the card inserted in the card reader and performs a card reset. If the card is powered up when the command is being issued, only a reset of the card is carried out. The power supply to the card is not switched off.

Command format

Header	Instruction	Data length LEN		Data
01 H	80 H	00 H	01 H	

Data = 00 H for automatic voltage detection

= 01 H for 5-volt card

= 02 H for 3-volt card

= 03 H for 1.8-volt card

Response data format

Header	Status	Data length LEN		ATR				
01 H							.....	

ATR Answer-To-Reset as transmitted by the card according to ISO7816-3.

NOTE: The ATR is only returned in the ACR38 response if the communication protocol of the card is compatible with the reader, i.e., if the card can be processed by the ACR38. Otherwise, the ACR38 returns an error status and deactivates the smart card interface.

9.2.3. POWER_OFF

This command powers off the card inserted in the card reader.

Command format

Header	Instruction	Data length LEN	
01 H	81 H	00 H	00 H

Response data format

Header	Status	Data length LEN	
01 H			

9.2.4. EXCHANGE_TPDU_T0

To exchange an APDU (Application Protocol Data Unit) command/response pair between the MCU card inserted in the ACR38 and the host computer.

Command format

Header	Instruction	Data length LEN		Data				
		MSB	LSB	T0 TPDU				
01 H	A0 H						.....	

LEN	Length of APDU command data, N
Data	T0 TPDU to be sent to the card
	Case 1: CLA INS P1 P2
	Case 2: CLA INS P1 P2 Le
	Case 3: CLA INS P1 P2 Lc Data
	Case 4: Not supported. The driver/application should break case 4 command into case 3 + case 2 commands.

Response data format

Header	Status	Data length LEN		BYTE 1	…	…	BYTE N	SW1	SW2
01 H									

BYTE x	Response data from card (if any).
SW1, SW2	Status code returned by the card.

9.2.5. EXCHANGE_TPDU_T1

To exchange an APDU (Application Protocol Data Unit) command/response pair between the MCU cards inserted in the ACR38 and the host computer using T1 protocol.

Command format

Header	Instruction	Data length LEN		Data				
		MSB	LSB	T1 TPDU Frame				
01 H	A1 H	MSB	LSB				.....	

LEN Length of APDU command data, N

Data T1 TPDU frame to be sent to the card. It should include NAD, PCB, LEN, INF and EDC fields. Please refer to ISO/IEC 7816:3:1997(E) Section 9.4 for detailed information.

Response data format

Header	Status	Data length LEN		BYTE 1	…	…	BYTE N
01 H							

BYTE x Response T1 Block from card (if any). The response should include NAD, PCB, LEN, INF and EDC fields. Please refer to ISO/IEC 7816:3:1997(E) Section 9.4 for detailed information.

Appendix A. ACR38 FW1.12c

Appendix A.1. Supported Card Types

This table lists the card types returned by `GET_READER_INFORMATION` corresponding with the respective card type code:

Card type code	Card Type
00_H	Auto-select T=0 or T=1 communication protocol
01_H	I2C memory card (1k, 2k, 4k, 8k and 16k bits)
02_H	I2C memory card (32k, 64k, 128k, 256k, 512k and 1024k bits)
03_H	Atmel AT88SC153 secure memory card
04_H	Atmel AT88SC1608 secure memory card
05_H	Infineon SLE4418 and SLE4428
06_H	Infineon SLE4432 and SLE4442
07_H	Infineon SLE4406, SLE4436 and SLE5536
08_H	Infineon SLE4404
09_H	Atmel AT88SC101, AT88SC102 and AT88SC1003
$0C_H$	MCU-based cards with T=0 communication protocol
$0D_H$	MCU-based cards with T=1 communication protocol

Appendix A.2. Response Error Codes

This table lists the error codes that may be returned by the ACR38:

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`

Appendix B. ACR38 FW1.10

Appendix B.1. Supported Card Types

The following table shows the values that must be specified in the *SET_CARD_TYPE* command for a particular card type to be used, and how the bits in the response to the *GET_ACR_STAT* command correspond with the respective card types.

Card Type	Card Type
00_H	Auto-select T=0 or T=1 communication protocol
01_H	I2C memory card (1k, 2k, 4k, 8k and 16k bits)
02_H	I2C memory card (32k, 64k, 128k, 256k, 512k and 1024k bits)
03_H	Atmel AT88SC153 secure memory card
04_H	Atmel AT88SC1608 secure memory card
05_H	Infineon SLE4418 and SLE4428
06_H	Infineon SLE4432 and SLE5542
07_H	Infineon SLE4406, SLE4436 and SLE5536
$0C_H$	MCU-based cards with T=0 communication protocol
$0D_H$	MCU-based cards with T=1 communication protocol

Appendix B.2. Response Status Codes

The following table is a list of the possible status code returned by the ACR38:

Status Code	Status
00	OK – command successfully executed
F4	SLOTERROT_PROCEDURE_BYTE_CONFLICT
F6	SLOTERROR_BAD_LENGTH
F7	SLOTERROR_BAD_FIDI
F8	SLOTERROR_BAD_ATR_TS
F9	SLOTERROR_ICC_NOT_POWERED_UP
FA	SLOTERROR_ICC_NOT_INSERTED
FB	SLOTERROR_HW_ERROR
FC	SLOTERROR_XFE_OVERRUN
FD	SLOTERROR_XFE_PARITY_ERROR
FE	SLOTERROR_ICC_MUTE
FF	SLOTERROR_CMD_ABORTED