Advanced Card Systems Ltd.
Card & Reader Technologies

ACR38x
Smart Card Reader

Reference Manual V6.03

Subject to change without prior notice

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

Table of Contents

List of Figures

No table of figures entries found.

List of Tables

1.0. Introduction

The ACR38x PC-linked Smart Card 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 ACR38x 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 for being responsible with smart card operations' technical details, which in many cases, are not relevant to the implementation of a smart card system.

1.1. Reference Documents

The following related documents are available from www.usb.org:

- Universal Serial Bus Specification 2.0 (also referred to as the USB specification), April 27, 2000
- Universal Serial Bus Common Class Specification 1.0, December 16, 1997

The following related documents can be ordered through www.ansi.org:

- ISO/IEC 7816-1; Identification Cards – Integrated circuit(s) cards with contacts - Part 1: Physical Characteristics
- ISO/IEC 7816-2; Identification Cards – Integrated circuit(s) cards with contacts - Part 2: Dimensions and Locations of the contacts
- ISO/IEC 7816-3; Identification Cards – Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols

1.2. Symbols and Abbreviations

Abbreviation	Description
ATR	Answer-To-Reset
ICC	Integrated Circuit Cards
NAD	Node Address
PPS	Protocol and Parameters Selection
TPDU	Transport Protocol Data Unit
USB	Universal Serial Bus

Table 1: Symbols and Abbreviations

2.0. Features

- USB 2.0 Full Speed Interface
- Smart Card Reader:
    - Supports ISO 7816 Class A, B and C (5 V, 3 V, 1.8 V) cards
    - Supports microprocessor cards with T=0 or T=1 protocol
    - Supports memory cards
    - Supports PPS (Protocol and Parameters Selection)
    - Features Short Circuit Protection
- Application Programming Interface:
    - Supports PC/SC
    - Supports CT-API (through wrapper on top of PC/SC)
- Supports Android™ 3.1 and above[1]
- Compliant with the following standards:
    - EN60950/IEC 60950
    - ISO 7816
    - CE
    - FCC
    - PC/SC
    - EMV 2000 Level 1
    - Microsoft® WHQL
    - RoHS 2
    - REACH

[1] *PC/SC support is not applicable*

3.0. Smart Card Support

3.1. MCU Cards

The ACR38x is a PC/SC-compliant smart card reader that supports ISO 7816 Class A, B and C (5 V, 3 V, and 1.8 V) smart cards. It also works with MCU (MicroController Unit) cards following the T=0 and T=1 protocol. For the SAM (Secure Access Module) slot, only MCU cards following T=0 protocol is supported.

The card ATR indicates the specific operation mode (TA2 present; bit b5 of TA2 must be 0) and when that particular mode is not supported by the ACR38x, the reader will reset the card to a negotiable mode. If the card cannot be set to negotiable mode, the reader will then reject the card.

When the card ATR indicates the negotiable mode (TA2 not present) and communication parameters other than the default parameters, the ACR38x will execute the PPS and try to use the communication parameters that the card suggested in its ATR. If the card does not accept the PPS, the reader will use the default parameters (F=372, D=1).

Note: *For the meaning of the aforementioned parameters, please refer to* ***ISO 7816-3****.*

3.2. Memory-based Smart Cards

The ACR38x works with several memory-based smart cards such as:

- Cards following the I2C bus 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
- Cards with secure memory IC with password and authentication, including:
  - Atmel®: AT88SC153 and AT88SC1608
- Cards with intelligent 1 KB 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

4.0. Smart Card Interface

The interface between the ACR38x and the inserted smart card follows the specification of ISO 7816-3 with certain restrictions or enhancements to increase the practical functionality of ACR38x.

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 ACR38x. The electrical specifications of this contact are identical to those of the signal RST (at contact C2).

4.3. Card Type Selection

The controlling personal computer must always select the card type through the proper command sent to the ACR38x prior to activating the inserted card. This includes both the memory cards and MCU-based cards.

For MCU-based cards, the reader is allowed 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 ACR38x 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 ACR38x and the card is immediately deactivated when the card is being 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: *The ACR38x never switches on the power supply to the inserted card by itself. The controlling computer through the proper command sent to the reader must explicitly do this.*

5.0. Power Supply

The ACR38x requires a voltage of 5 V DC, 100 mA, regulated, power supply. The ACR38x gets power supply from the computer (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 ACR38x 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 ACR38x and smart card.

6.0. USB Interface

6.1. Communication Parameters

The ACR38x is connected to a computer through USB as specified in the USB Specification 2.0. The ACR38x is working in full speed mode (e.g. 12 Mbps).

Pin	Signal	Function
1	V_{BUS}	+5 V power supply for the reader
2	D-	Differential signal transmits data between ACR38x and PC
3	D+	Differential signal transmits data between ACR38x and PC
4	GND	Reference voltage level for power supply

Table 2: USB Interface Wiring

***Note:** In order for the ACR38x to function properly through USB interface, either ACS PC/SC driver has to be installed.*

6.2. Endpoints

The ACR38x 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 ACR38x (data packet size is 64 bytes)
Bulk IN	For response to be sent from ACR38x to host (data packet size is 64 bytes)
Interrupt IN	For card status message to be sent from ACR38x to host (data packet size is 8 bytes)

7.0. Communication Protocol

During normal operation, the ACR38x acts as a slave device with regards to the communication between a computer and the reader. The communication is carried out in the form of success 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 ACR38x only after the response to the previous command has been received.

There are two cases where the reader transmits data without receiving a command from the computer:

1. Reset Message
2. Card Status Message

7.1. Command to Reader

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		

Where:

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

Instruction The instruction code of the command to be carried out by the ACR38x.

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 ACR38x through the BULK OUT endpoint.*

7.2. Response from Reader

The response from the ACR38x to any command depends on whether the command has been received by the ACR38x without error (e.g. checksum error).

The response from the ACR38x 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		

Where:

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

Note: *A table listing the possible values of the status byte and the corresponding meaning is given in* ***Appendix B****.*

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 ACR38x to the host computer through BULK IN endpoint.*

7.3. Card Status Message

When a card is being inserted into the ACR38x or an inserted card is being removed from the ACR38x while the ACR38x is in idle mode (i.e. not executing a command), the ACR38x transmits a Card Status Message to notify the host computer of the change in the card insertion status.

The Card Status Message consists of the following structure and contents:

Card Status Message for Card Insertion

Byte	1	2	3	4
	Header	Status	Data Length	
	01h	C1h	00h	00h

Card Status Message for Card Removal

Byte	1	2	3	4
	Header	Status	Data Length	
	01h	C0h	00h	00h

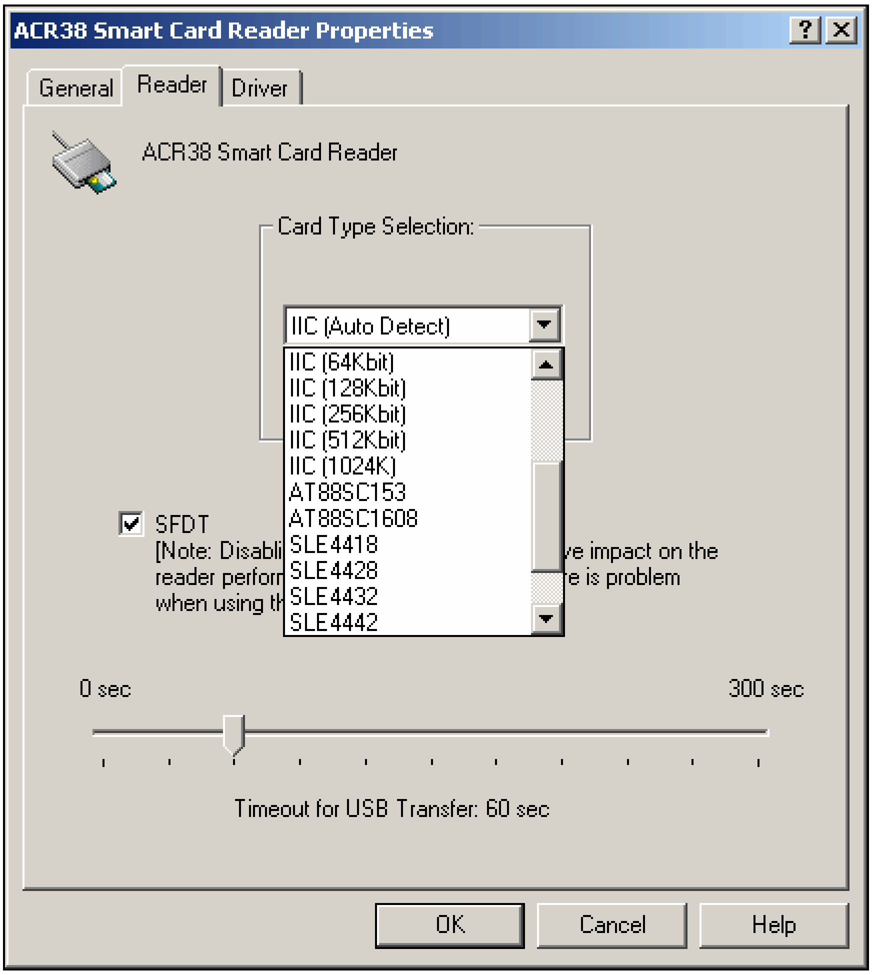
A Card Status Message is transmitted only once for every card insertion or removal. The ACR38x does not expect an acknowledgement signal from the computer. After transmitting a status message, the ACR38x waits for the next command from the computer.

Note: *Card Status Messages are sent from ACR38x to the host computer through INTERRUPT IN endpoint.*

8.0. Memory Card Type Selection

8.1. By Property Sheet

User could invoke the reader setting property sheet by selecting the **Properties** of ACR38 Smart Card Reader device under the Device Manager.



ACR38 Smart Card Reader Properties

ACR38 Smart Card Reader

Card Type Selection: IIC (Auto Detect)

☑ SFDT

[Note: Disabli… …ve impact on the reader perfor… …e is problem when using th…

Timeout for USB Transfer: 60 sec

Figure 1: ACR38x Reader Setting Property Sheet

The ACR38x needs to be removed, and then reconnected to the computer in order for the change to take effect.

8.2. By Programmatic Method

The card type can also be changed the program run-time using Vendor Specific extension API of PC/SC.

Application programs are required to include the following MACRO in one of the source header file:

```
#define IOCTL_SMARTCARD_SET_CARD_TYPE  SCARD_CTL_CODE(2060)
```

Applications should connect to PC/SC using the SCARD_SHARE_DIRECT protocol. After which, invoke the *SCardControl()* and use IOCTL_SMARTCARD_SET_CARD_TYPE for the *dwControlCode* parameter to inform the driver of new card type. The input buffer will be a LONG variable storing the desired card type. The return value is either SCARD_S_SUCCESS or a WIN32 Error (ERROR_INSUFFICIENT_BUFFER).

Example:

```
int main()
{
   long rv;
   long nCardType = 15; // SLE4418 – refer to inf for more info
   BYTE cbOutBuffer[10];
   SCARDCONTEXT hctx;
   SCARDHANDLE hsc;
   DWORD dwActievProtocol;
   DWORD dwBytesRet;

   rv = SCardEstablishContext(SCARD_SCOPE_SYSTEM,NULL,NULL,&hctx);
   if (rv != SCARD_S_SUCCESS)
      return rv;

   rv = SCardConnect(
hctx,
”ACS ACR38U 0”,
SCARD_SHARE_DIRECT,  // This allows apps to connect to
            // PC/SC even without card inserted
0,
&hsc,
&dwActiveProtocol);

if (rv != SCARD_S_SUCCESS)
{
      // error handling …
      return rv;
}

rv = SCardControl(hsc, IOCTL_SMARTCARD_SET_CARD_TYPE,
&nCardType, sizeof(nCardType), cbOutBuffer, 10,
&dwBytesRet);

if (  rv == SCARD_S_SUCCESS && cbOutBuffer[0] == 0x90 && cbOutBuffer[1] ==
0x00)
{
// OK
   }
else . . .  // other error handling
   . . .
}
```

9.0. Commands

9.1. Control Commands

The Control Commands are in charge of the internal operation of the ACR38x. They do not directly affect the card inserted in the ACR38x and are therefore independent of the selected card type.

9.1.1. GET_ATR_STAT

This command returns relevant information about the particular ACR38x 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	
01h	01h	00h	00h

Response Data Format

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

Where:

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 ACR38x. 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 A** 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:

00h: No card inserted

01h: Card inserted, not powered up

03h: Card powered up

9.1.2. SELECT_CARD_TYPE

This command sets the required card type. The firmware in ACR38x 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
01h	02h	00h	01h	

Where:

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

Response Data Format

Header	Status	Data length	
		LEN	
01h			

9.1.3. SET_OPTION

This command selects the options for the ACR38x.

Command Format

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

Where:

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	
01h			

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
01h	0Ah	MSB	LSB	

Where:

LEN Length of PPS request. Typical value is “4”.

PPS Request PPS Request to send to the card.

Note: *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=94h (62500 baud at 4 MHz) is: FF 11 94 7Ah

Response Data Format

Header	Status	Data length		Data				
		LEN						
01h							…	

9.1.5. SET_READER_PPS

This command sends PPS Response to the ACR38x and asks the ACR38x to switch its protocol and/or speed to communicate with the smart card. This command should work in pair with SET_CARD_PPS.

Command Format

Header	Instruction	Data length		Data
		LEN		PPS Response
01h	0Bh	MSB	LSB	

Where:

LEN Length of PPS response; Typical value is “4”.

PPS Response PPS Response received from the card.

***Note:** 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 ACR38x. The ACR38x 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	
01h			

9.2. Card Commands

The Card Commands are directed toward the card inserted in the ACR38x. The structure of these commands and the data transmitted in the commands and responses depend on the selected card type.

9.2.1. MCU Card Command Set

9.2.1.1. RESET_WITH_5_VOLTS_DEFAULT

This command powers up the card inserted in the ACR38x 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	
01h	80h	00h	00h

Response Data Format

Header	Status	Data length		ATR				
		LEN						
01h							…	

Where:

ATR Answer-To-Reset as transmitted by the card according to ISO 7816-3.

Note: *ATR is only returned in the reader response if the communication protocol of the card is compatible with the ACR38x (i.e. if the card can be processed by the reader). Otherwise, ACR38x returns an error status and deactivates the smart card interface.*

9.2.1.2. RESET_WITH_SPECIFIC_VOLTAGE

This command powers up the card inserted in the ACR38x 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		Data
		LEN		
01h	80h	00h	01h	

Where:

Data = 00h for automatic voltage detection.
= 01h for 5-volt card.
= 02h for 3-volt card.
= 03h for 1.8-volt card.

Response data format

Header	Status	Data length		ATR				
		LEN						
01h							…	

Where:

ATR Answer-To-Reset as transmitted by the card according to ISO 7816-3.

Note: *The ATR is only returned in the reader response if the communication protocol of the card is compatible with the ACR38x (i.e. if the card can be processed by the reader). Otherwise, ACR38x returns an error status and deactivates the smart card interface.*

9.2.1.3. POWER_OFF

This command powers off the card inserted in the ACR38x.

Command Format

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

Response Data Format

Header	Status	Data length	
		LEN	
01h			

9.2.1.4. EXCHANGE_TPDU_T0

This command exchanges an APDU command/response pair between the card inserted in the ACR38x and the host computer.

Command Format

Header	Instruction	Data length LEN		Data				
		MSB	LSB	T0 TPDU				
01h	A0h						…	

Where:

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		BYTE 1	…	…	BYTE N	SW1	SW2
		LEN							
01h									

Where:

BYTE x Response data from card (if any).

SW1 SW2 Status code returned by the card.

9.2.1.5. EXCHANGE_TPDU_T1

This command exchanges an APDU command/response pair between the cards inserted in the ACR38x and the host computer using T1 protocol.

Command Format

Header	Instruction	Data length LEN		Data				
		MSB	LSB	T1 TPDU Frame				
01h	A1h						…	

Where:

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.

Note: *Please refer to* ***ISO/IEC 7816:3:1997(E) (Section 9.4)*** *for detailed information.*

Response Data Format

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

Where:

BYTE x Response T1 Block from card (if any). The response should include NAD, PCB, LEN, INF and EDC fields.

Note: *Please refer to* ***ISO/IEC 7816:3:1997(E) (Section 9.4)*** *for detailed information.*

9.2.2. Memory Card Command Set

9.2.2.1. Memory Card – 1, 2, 4, 8, and 16 kilobit I2C Card

9.2.2.1.1. 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 ACR38x is powered off.

Send Buffer Format

SCardTransmit Send Buffer					
CLA	**INS**	**P1**	**P2**	**Lc (P3)**	**Page size**
FFh	01h	00h	00h	01h	

Where:

Page size = 03h for 8-byte page write.
= 04h for 16-byte page write.
= 05h for 32-byte page write.
= 06h for 64-byte page write.
= 07h for 128-byte page write.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	**SW2**

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.1.2. READ_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Byte Address		MEM_L (P3)
		MSB (P1)	LSB (P2)	
FFh	B0h			

Where:

Byte Address Memory address location of the memory card.

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

Response Data Format

SCardTransmit Receive Buffer					
BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE x Data read from memory card.

SW1 SW2 = 90 00h if no error.

9.2.2.1.3. WRITE_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	Byte Address		MEM_L (P3)	Byte 1	....	....	Byte n
		MSB (P1)	LSB (P2)					
FFh	D0h							

Where:

Byte Address Memory address location of the memory card.

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

Byte x Data to be written to the memory card.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.2. Memory Card – 32, 64, 128, 256, 512, and 1024 kilobit I2C Card

9.2.2.2.1. 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 ACR38x is powered off.

Send Buffer Format

SCardTransmit Send Buffer					
CLA	INS	P1	P2	Lc (P3)	Page size
FFh	01h	00h	00h	01h	

Where:

Data TPDU to be sent to the card.

Page size = 03h for 8-byte page write.
= 04h for 16-byte page write.
= 05h for 32-byte page write.
= 06h for 64-byte page write.
= 07h for 128-byte page write.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.2.2. READ_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Byte Address		MEM_L (P3)
		MSB (P1)	LSB (P2)	
FFh				

Where:

INS = B0h for 32, 64, 128, 256, 512 kilobit iic card.

= 1011 000*b for 1024 kilobit 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 Buffer Format

SCardTransmit Receive Buffer					
BYTE 1	...	...	BYTE N	SW1	SW2

Where:

BYTE x Data read from memory card.

SW1 SW2 = 90 00h if no error.

9.2.2.2.3. WRITE_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	Byte Address		MEM_L (P3)	Byte 1	....	....	Byte n
		MSB (P1)	LSB (P2)					
FFh								

Where:

INS = D0h for 32, 64, 128, 256, 512 kilobit iic card.

= 1101 000*b for 1024 kilobit 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 in the memory card.

Byte x Data to be written to the memory card.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.3. Memory Card – ATMEL AT88SC153

9.2.2.3.1. READ_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	**INS**	**P1**	**Byte Address (P2)**	**MEM_L (P3)**
FFh		00h		

Where:

INS = B0h for reading zone 00b.
= B1h for reading zone 01b.
= B2h for reading zone 10b.
= B3h for reading zone 11b.
= B4h 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 Buffer Format

SCardTransmit Receive Buffer					
BYTE 1	**…**	**…**	**BYTE N**	**SW1**	**SW2**

Where:

BYTE x Data read from memory card.

SW1 SW2 = 90 00h if no error.

9.2.2.3.2. WRITE_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	**INS**	**P1**	**Bye Address (P2)**	**MEM_L (P3)**	**Byte 1**	**....**	**....**	**Byte n**
FFh		00h						

Where:

INS = D0h for writing zone 00b.
= D1h for writing zone 01b.
= D2h for writing zone 10b.
= D3h for writing zone 11b.
= D4h for writing fuse.

Byte Address Memory address location of the memory card.

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

MEM_D Data to be written to the memory card.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	**SW2**

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.3.3. VERIFY_PASSWORD

Send Buffer Format

SCardTransmit Send Buffer							
CLA	**INS**	**P1**	**P2**	**Lc (P3)**	**Pw(0)**	**Pw(1)**	**Pw(2)**
FFh	20h	00h		03h			

Where:

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

P2 = 0000 00rpb

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 Buffer Format

SCardTransmit Receive Buffer	
SW1	**SW2**

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.3.4. INITIALIZE_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	**INS**	**P1**	**P2**	**Lc (P3)**	**Q(0)**	**Q(1)**	**…**	**Q(7)**
FFh	84h	00h	00h	08h				

Where:

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

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	**SW2**

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.3.5. VERIFY_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	**INS**	**P1**	**P2**	**Lc (P3)**	**Ch(0)**	**Ch(1)**	**…**	**Ch(7)**
FFh	82h	00h	00h	08h				

Where:

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

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	**SW2**

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.4. Memory Card – ATMEL AT88SC1608

9.2.2.4.1. READ_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	**INS**	**Zone Address (P1)**	**Byte Address (P2)**	**MEM_L (P3)**
FFh				

Where:

INS = B0h for reading user zone.

= B1h for reading configuration zone or reading fuse.

Zone Address = 0000 0A10A9A8b, where A10 is the MSB of zone address.

= don't care for reading fuse.

Byte Address = A7A6A5A4 A3A2A1A0b is the memory address location of the memory card.

= 1000 0000b for reading fuse.

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

Response Buffer Format

SCardTransmit Receive Buffer					
BYTE 1	**…**	**…**	**BYTE N**	**SW1**	**SW2**

Where:

BYTE x Data read from memory card.

SW1 SW2 = 90 00h if no error.

9.2.2.4.2. WRITE_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	**INS**	**Zone Address (P1)**	**Byte Address (P2)**	**MEM_L (P3)**	**Byte 1**	**....**	**....**	**Byte n**
FFh								

Where:

INS = D0h for writing user zone.

= D1h for writing configuration zone or writing fuse.

Zone Address = 0000 0A10A9A8b, where A10 is the MSB of zone address.

= don't care for writing fuse.

Byte Address = A7A6A5A4 A3A2A1A0b is the memory address location of the memory card.

= 1000 0000b for writing fuse.

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

Byte x Data to be written to the memory card.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	**SW2**

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.4.3. VERIFY_PASSWORD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (P3)	Data			
FFh	20h	00h	00h	04h	RP	Pw(0)	Pw(1)	Pw(2)

Where:

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

RP = 0000 rp2p1p0b

where the four bits "rp2p1p0" indicate the password to compare:

r = 0: Write password

r = 1: Read password

p2p1p0: Password set number

(rp2p1p0 = 0111 for the secure code)

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.4.4. INITIALIZE_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (P3)	Q(0)	Q(1)	…	Q(7)
FFh	84h	00h	00h	08h				

Where:

Byte Address Memory address location of the memory card.

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

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.4.5. VERIFY_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (P3)	Q1(0)	Q1(1)	…	Q1(7)
FFh	82h	00h	00h	08h				

Where:

Byte Address Memory address location of the memory card.

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

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.5. Memory Card – SLE 4418/SLE 4428/SLE 5518/SLE5528

9.2.2.5.1. READ_MEMORY_WITH_PROTECT_BIT_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	**INS**	**Byte Address**		**MEM_L (P3)**
		MSB (P1)	**LSB (P2)**	
FFh	B0h			

Where:

MSB Byte Address = 0000 00A9A8b is the memory address location of the memory card.

LSB Byte Address = A7A6A5A4 A3A2A1A0b is the memory address location of the memory card.

MEM_L Length of data to be read from the memory card. (Maximum allowable size is ECh).

Response Buffer Format

SCardTransmit Receive Buffer									
BYTE 1	**…**	**…**	**BYTE N**	**PROT 1**	**…**	**…**	**PROT L**	**SW1**	**SW2**

Where:

BYTE x Data read from memory card.

PROT y Bytes containing the protection bits of the data bytes read.

SW1 SW2 = 90 00h if no error.

The number L of protection bytes returned in the response is determined by the number N of data bytes read from the card as follows:

L = 1 + INT(N/8)

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

Where:

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

‘0’ byte is write protected.

‘1’ byte can be written.

9.2.2.5.2. READ_MEMORY_WITHOUT_PROTECT_BIT_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Byte Address		MEM_L (P3)
		MSB (P1)	LSB (P2)	
FFh	B2h			

Where:

MSB Byte Address = 0000 00A9A8b is the memory address location of the memory card.

LSB Byte Address = A7A6A5A4 A3A2A1A0b is the memory address location of the memory card.

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

Response Buffer Format

SCardTransmit Receive Buffer					
BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE x Data read from the memory card.

SW1 SW2 = 90 00h if no error.

9.2.2.5.3. WRITE_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	Byte Address		MEM_L (P3)	Byte 1	....	....	Byte N
		MSB (P1)	LSB (P2)					
FFh	D0h							

Where:

MSB Byte Address = 0000 00A9A8b is the memory address location of the memory card.

LSB Byte Address = A7A6A5A4 A3A2A1A0b is the memory address location of the memory card.

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

Byte x Data to be written in the memory card.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.5.4. WRITE_PROTECTION_MEMORY_CARD

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

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	Byte Address		MEM_L (P3)	Byte 1	....	....	Byte N
		MSB (P1)	LSB (P2)					
FFh	D1h							

Where:

MSB Byte Address= 0000 00A9A8b is the memory address location of the memory card.

LSB Byte Address = A7A6A5A4 A3A2A1A0b 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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.5.5. PRESENT_CODE_MEMORY_CARD (SLE 4428 and SLE 5528)

This command is used to submit the secret code to the memory card to enable the write operation with the SLE 4428 card. The following actions are executed:

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

Send Buffer Format

SCardTransmit Send Buffer						
CLA	INS	P1	P2	MEM_L (P3)	CODE	
					Byte 1	Byte 2
FFh	20h	00h	00h	02h		

Where:

CODE Two bytes secret code (PIN).

Response Buffer Format

ERRCNT	CODE		SW1	SW2
	Byte 1	Byte 2		

Where:

ERRCNT The value of the presentation error counter after the code presentation.

CODE The two bytes secret code read from the card.

SW1 SW2 = 90 00h if no error.

If the correct code has been presented to the card, the value of ERRCNT is FFh and the value of CODE is identical to the code data specified in the command.

9.2.2.5.6. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4428 and SLE 5528)

This command is used to read the presentation error counter for the secret code.

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	P2	MEM_L (P3)
FFh	B1h	00h	00h	00h

Response Buffer Format

SCardTransmit Receive Buffer				
ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

Where:

ERRCNT	The value of the presentation error counter.
DUMMY	Three bytes dummy data read from the card.
SW1 SW2	= 90 00h if no error.

9.2.2.6. Memory Card – SLE 4432/SLE 4442/SLE 5532/SLE 5542

9.2.2.6.1. READ_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)
FFh	B0h	00h		

Where:

Byte Address = A7A6A5A4 A3A2A1A0b is the memory address location of the memory card.

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

Response Buffer Format

SCardTransmit Send Buffer									
BYTE 1	…	…	BYTE N	PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

Where:

BYTE x Data read from memory card.

PROT y Bytes containing the protection bits from protection memory.

SW1 SW2 = 90 00h 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

Where:

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

‘0’ byte is write protected.

‘1’ byte can be written.

9.2.2.6.2. WRITE_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)	Byte 1	....	....	Byte N
FFh	D0h	00h						

Where:

Byte Address = A7A6A5A4 A3A2A1A0b is the memory address location of the memory card.

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

Byte x Data to be written in the memory card.

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.6.3. WRITE_PROTECTION_MEMORY_CARD

Each byte 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’.

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)	Byte 1	....	....	Byte N
FFh	D1h	00h						

Where:

Byte Address = 000A4 A3A2A1A0b (00h to 1Fh) 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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.6.4. PRESENT_CODE_MEMORY_CARD (SLE 4442 and SLE 5542)

This command is used to submit the secret code to the memory card to enable the write operation with the SLE 4442 and SLE 5542 card. The following actions are executed:

1. Search a '1' bit in the presentation error counter
2. Write the bit to '0' present the specified code to the card try to erase the presentation error counter

Send Buffer Format

SCardTransmit Send Buffer							
CLA	INS	P1	P2	MEM_L (P3)	CODE		
					Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	03h			

Where:

CODE Three bytes secret code (PIN).

Response Buffer Format

SCardTransmit Receive Buffer					
ERRCNT	CODE			SW1	SW2
	Byte 1	Byte 2	Byte 3		

Where:

ERRCNT The value of the presentation error counter after the code presentation.

CODE The three bytes secret code read from the card.

SW1 SW2 = 90 00h if no error.

If the correct code has been presented to the card, the value of ERRCNT is 07h and the value of CODE is identical to the code data specified in the command.

9.2.2.6.5. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4442 and SLE 5542)

This command is used to read the presentation error counter for the secret code.

Send Buffer Format

SCardTransmit Send Buffer				
CLA	**INS**	**P1**	**P2**	**MEM_L (P3)**
FFh	B1h	00h	00h	00h

Response Buffer Format

SCardTransmit Receive Buffer					
ERRCNT	**DUMMY 1**	**DUMMY 2**	**DUMMY 3**	**SW1**	**SW2**

Where:

ERRCNT	The value of the presentation error counter.
DUMMY	Three bytes dummy data read from the card.
SW1 SW2	= 90 00h if no error.

9.2.2.6.6. CHANGE_CODE_MEMORY_CARD (SLE 4442 and SLE 5542)

This command is used to write the specified data as new secret code in the card.

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

Send Buffer Format

SCardTransmit Send Buffer							
CLA	INS	P1	P2	MEM_L (P3)	CODE		
					Byte 1	Byte 2	Byte 3
FFh	D2h	00h	01h	03h			

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.7. Memory Card – SLE 4406/SLE 4436/SLE 5536/SLE 6636

9.2.2.7.1. READ_MEMORY_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)
FFh	B0h	00h		

Where:

Byte Address = Memory address location of the memory card.

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

Response Buffer Format

SCardTransmit Receive Buffer					
BYTE 1	…	…	BYTE N	SW1	SW2

Where:

BYTE x Data read from memory card.

SW1 SW2 = 90 00h if no error.

9.2.2.7.2. WRITE_ONE_BYTE_MEMORY_CARD

This command is used to write one byte to the specific 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 (SLE 4436, SLE 5536 and SLE 6636 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 (SLE 4436, SLE 5536 and SLE 6636 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 SLE 4436 and SLE 5536 card can be enabled or disabled in the write operation.

Command Format

SCardTransmit SendBuffer						
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**MODE**	**BYTE**
FFh	D0h	00h		02h		

Where:

Byte Address = Memory address location of the memory card.

LEN = 5 + MEM_L

MODE Specifies the write mode and backup option:

00h: Write

01h: Write with carry

02h: Write with backup enabled (SLE 4436, SLE 5536 and SLE 6636 only)

03h: Write with carry and with backup enabled (SLE 4436, SLE 5536 and SLE 6636 only)

BYTE Byte value to be written in the card.

Response Data Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.7.3. PRESENT_CODE_MEMORY_CARD

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

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

The ACR38x 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

SCardTransmit Send Buffer								
CLA	INS	P1	P2	MEM_L	CODE			
					ADDR	Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	04h				

Where:

ADDR Byte address of the presentation counter in the card.

CODE Three bytes secret code (PIN).

Response Data Format

SCardTransmit Receive Buffer	
SW1	SW2

Where:

SW1 SW2 = 90 00h if no error.

9.2.2.7.4. AUTHENTICATE_MEMORY_CARD (SLE 4436, SLE 5536 and SLE 6636)

This command is used to read a card authentication certificate from an SLE 5536 or SLE 6636 card. The following actions are executed by the ACR38x:

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 ACR38x returns the 16 bits of authentication data calculated by the card in the response.

Command format

SCardTransmit Send Buffer											
CLA	**INS**	**P1**	**P2**	**MEM_L**	**CODE**						
					KEY	**CLK_CNT**	**Byte1**	**Byte 2**	**……**	**Byte 5**	**Byte 6**
FFh	84h	00h	00h	08h							

Where:

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

00h: Key 1 with no cipher block chaining

01h: Key 2 with no cipher block chaining

80h: Key 1 with cipher block chaining (SLE 5536 and SLE 6636 only)

81h: Key 2 with cipher block chaining (SLE 5536 and SLE 6636 only)

CLK_CNT Number of CLK pulses to be supplied to the card for the computation of each bit of the authentication certificate.

BYTE 1...6 Card challenge data.

Response data format

SCardTransmit Receive Buffer			
CERT		**SW1**	**SW2**

Where:

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 00h if no error.

Appendix A. Supported Card Types

The following table shows the value 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.

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

Table 3: Supported Card Types

Appendix B. Response Status Codes

The following table is a list of the possible status code returned by the ACR38x:

Status Code	Status
00h	OK – command successfully executed
F4h	SLOTERROT_PROCEDURE_BYTE_CONFLICT
F6h	SLOTERROR_BAD_LENGTH
F7h	SLOTERROR_BAD_FIDI
F8h	SLOTERROR_BAD_ATR_TS
F9h	SLOTERROR_ICC_NOT_POWERED_UP
FAh	SLOTERROR_ICC_NOT_INSERTED
FBh	SLOTERROR_HW_ERROR
FCh	SLOTERROR_XFE_OVERRUN
FDh	SLOTERROR_XFE_PARITY_ERROR
FEh	SLOTERROR_ICC_MUTE
FFh	SLOTERROR_CMD_ABORTED

Table 4: Response Status Codes