



**Advanced Card Systems Ltd.**  
Card & Reader Technologies

# ACR38 Smart Card Reader



Application Note: Memory Card Access Manual



## Table of Contents

<b>1.0.</b>	<b>Introduction .....</b>	<b>5</b>
1.1.	Supported Memory Cards.....	5
<b>2.0.</b>	<b>Memory Card Type Selection.....</b>	<b>6</b>
2.1.	Memory Card Type Selection for Firmware 1.12c.....	6
2.1.1.	By Programmatic Method .....	6
2.2.	Memory Card Type Selection for Firmware 1.10.....	6
2.2.1.	By Property Sheet.....	6
2.2.2.	By Programmatic Method .....	7
2.2.3.	Logical Flow of Memory Card Functions .....	9
<b>3.0.</b>	<b>ACR38 Firmware 1.12c Memory Card Command Set .....</b>	<b>10</b>
3.1.	Recollection card – 1, 2, 4, 8 and 18 Kbit I2C Card .....	10
3.1.1.	SELECT_CARD_TYPE .....	10
3.1.2.	SELECT_PAGE_SIZE .....	10
3.1.3.	READ_MEMORY_CARD.....	11
3.1.4.	WRITE_MEMORY_CARD .....	11
3.2.	Memory Card – 32, 64, 128, 256, 512, and 1024 Kbit I2C Card.....	12
3.2.1.	SELECT_CARD_TYPE .....	12
3.2.2.	SELECT_PAGE_SIZE .....	12
3.2.3.	READ_MEMORY_CARD.....	13
3.2.4.	WRITE_MEMORY_CARD .....	13
3.3.	Memory Card – ATMEL AT88SC153 .....	14
3.3.1.	SELECT_CARD_TYPE .....	14
3.3.2.	READ_MEMORY_CARD.....	14
3.3.3.	WRITE_MEMORY_CARD .....	15
3.3.4.	VERIFY_PASSWORD .....	15
3.3.5.	INITIALIZE_AUTHENTICATION.....	16
3.3.6.	VERIFY_AUTHENTICATION .....	16
3.4.	Memory Card – ATMEL AT88C1608.....	16
3.4.1.	SELECT_CARD_TYPE .....	16
3.4.2.	READ_MEMORY_CARD.....	17
3.4.3.	WRITE_MEMORY_CARD .....	18
3.4.4.	VERIFY_PASSWORD .....	18
3.4.5.	INITIALIZE_AUTHENTICATION.....	19
3.4.6.	VERIFY_AUTHENTICATION .....	19
3.5.	Memory Card – SLE 4418 / SLE 4428 / SLE 5518 / SLE 5528 .....	19
3.5.1.	SELECT_CARD_TYPE .....	19
3.5.2.	READ_MEMORY_CARD.....	20
3.5.3.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4428 and SLE 5528) 20	
3.5.4.	READ_PROTECTION_BIT .....	21
3.5.5.	WRITE_MEMORY_CARD .....	22
3.5.6.	WRITE_PROTECTION_MEMORY_CARD .....	22
3.5.7.	PRESENT_CODE_MEMORY_CARD (SLE 4428 and SLE 5528).....	23
3.6.	Memory Card – SLE 4432 / SLE 4442 / SLE 5532 / SLE 5542 .....	23
3.6.1.	SELECT_CARD_TYPE .....	23
3.6.2.	READ_MEMORY_CARD.....	24
3.6.3.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4442 and SLE 5542) 24	
3.6.4.	READ_PROTECTION_BITS .....	25
3.6.5.	WRITE_MEMORY_CARD .....	25
3.6.6.	WRITE_PROTECTION_MEMORY_CARD .....	26
3.6.7.	PRESENT_CODE_MEMORY_CARD (SLE 4442 and SLE 5542).....	26
3.6.8.	CHANGE_CODE_MEMORY_CARD (SLE 4442 and SLE 5542) .....	27
3.7.	Memory Card – SLE 4406 / SLE 4436 / SLE 5536 / SLE 6636 .....	27
3.7.1.	SELECT_CARD_TYPE .....	27



3.7.2.	READ_MEMORY_CARD.....	27
3.7.3.	WRITE_ONE_BYTE_MEMORY_CARD.....	28
3.7.4.	PRESENT_CODE_MEMORY_CARD.....	29
3.7.5.	AUTHENTICATE_MEMORY_CARD (SLE 4436, SLE 5536 and SLE 6636).....	29
3.8.	Memory Card – SLE 4404.....	30
3.8.1.	SELECT_CARD_TYPE.....	30
3.8.2.	READ_MEMORY_CARD.....	31
3.8.3.	WRITE_MEMORY_CARD.....	31
3.8.4.	ERASE_SCRATCH_PAD_MEMORY_CARD.....	32
3.8.5.	VERIFY_USER_CODE.....	32
3.8.6.	VERIFY_MEMORY_CODE.....	33
3.9.	Memory Card – AT88SC101 / AT88SC102 / AT88SC1003.....	33
3.9.1.	SELECT_CARD_TYPE.....	33
3.9.2.	READ_MEMORY_CARD.....	34
3.9.3.	WRITE_MEMORY_CARD.....	34
3.9.4.	ERASE_NON_APPLICATION_ZONE.....	35
3.9.5.	ERASE_APPLICATION_ZONE_WITH_ERASE.....	35
3.9.6.	ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE.....	36
3.9.7.	VERIFY_SECURITY_CODE.....	37
3.9.8.	BLOWN_FUSE.....	38
3.10.	Other Commands Access via PC_to_RDR_XfrBlock.....	39
3.10.1.	GET_READER_INFORMATION.....	39
3.11.	Supported Card Types.....	40
3.12.	Response Error Codes.....	40
<b>4.0.</b>	<b>ACR38 Firmware 1.10 Memory Card Command Set.....</b>	<b>41</b>
4.1.	Memory Card – 1, 2, 4, 8, and 16 Kbit I2C Card.....	41
4.1.1.	SELECT_PAGE_SIZE.....	41
4.1.2.	READ_MEMORY_CARD.....	41
4.1.3.	WRITE_MEMORY_CARD.....	42
4.2.	Memory Card – 32, 64, 128, 256, 512, and 1024 Kbit I2C Card.....	42
4.2.1.	SELECT_PAGE_SIZE.....	42
4.2.2.	READ_MEMORY_CARD.....	43
4.2.3.	WRITE_MEMORY_CARD.....	43
4.3.	Memory Card – ATMEL AT88SC153.....	44
4.3.1.	READ_MEMORY_CARD.....	44
4.3.2.	WRITE_MEMORY_CARD.....	44
4.3.3.	VERIFY_PASSWORD.....	45
4.3.4.	INITIALIZE_AUTHENTICATION.....	45
4.3.5.	VERIFY_AUTHENTICATION.....	46
4.4.	Memory Card – ATMEL AT88SC1608.....	46
4.4.1.	READ_MEMORY_CARD.....	46
4.4.2.	WRITE_MEMORY_CARD.....	47
4.4.3.	VERIFY_PASSWORD.....	47
4.4.4.	INITIALIZE_AUTHENTICATION.....	48
4.4.5.	VERIFY_AUTHENTICATION.....	48
4.5.	Memory Card – SLE 4418 / SLE 4428 / SLE 5518 / SLE5528.....	48
4.5.1.	READ_MEMORY_WITH_PROTECT_BIT_CARD.....	48
4.5.2.	READ_MEMORY_WITHOUT_PROTECT_BIT_CARD.....	49
4.5.3.	WRITE_MEMORY_CARD.....	50
4.5.4.	WRITE_PROTECTION_MEMORY_CARD.....	50
4.5.5.	PRESENT_CODE_MEMORY_CARD (SLE 4428 and SLE 5528).....	51
4.5.6.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4428 and SLE 5528).....	51
4.6.	Memory Card – SLE 4432 / SLE 4442 / SLE 5532 / SLE 5542.....	52
4.6.1.	READ_MEMORY_CARD.....	52
4.6.2.	WRITE_MEMORY_CARD.....	52
4.6.3.	WRITE_PROTECTION_MEMORY_CARD.....	53
4.6.4.	PRESENT_CODE_MEMORY_CARD (SLE 4442 and SLE 5542).....	53
4.6.5.	READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE 4442 and SLE 5542).....	53



SLE 5542) 54

4.6.6. CHANGE\_CODE\_MEMORY\_CARD (SLE 4442 and SLE 5542) .....54

4.7. Memory Card – SLE 4406 / SLE 4436 / SLE 5536 / SLE 6636 .....55

4.7.1. READ\_MEMORY\_CARD.....55

4.7.2. WRITE\_ONE\_BYTE\_MEMORY\_CARD.....55

4.7.3. PRESENT\_CODE\_MEMORY\_CARD .....56

4.7.4. AUTHENTICATE\_MEMORY\_CARD (SLE 4436, SLE 5536 and SLE 6636).....57

## Figures

Figure 1: ACR38 Reader Setting Property Sheet..... 7

## Tables

Error! No table of figures entries found.



## 1.0. Introduction

The ACR38 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 and this prevents in most cases, the direct communication between the smart card and the 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's specific particulars, it releases the computer software programmer from getting involved with the technical details of the smart card operation which are in many cases, not relevant in the implementation of a smart card system.

This document contains the PC/SC Memory Card Command set for ACR38 Firmware 1.12c and 1.10. The ACR38 Firmware 1.12C is backward compatible with the ACR38 Firmware 1.10 with minor changes to the Memory Card Command Set. In this document, we will refer to the ACR38 with firmware 1.12c as ACR38 FW1.12c while ACR38 with firmware 1.10 will be referred to ACR38 FW1.10.

### 1.1. Supported Memory 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	x	✓
7. Cards with Security Logic with Application Zone(s), including: Atmel: AT88SC101, AT88SC102 and AT88SC1003	x	✓



## 2.0. Memory Card Type Selection

### 2.1. Memory Card Type Selection for Firmware 1.12c

#### 2.1.1. By Programmatic Method

The SELECT\_CARD\_TYPE command must be executed first before other memory card commands. This command powers down and up the selected card inserted in the card reader and performs a card reset. 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. For the Memory Card Command Set for ACR38 FW1.12c, please refer to Section 3.0.

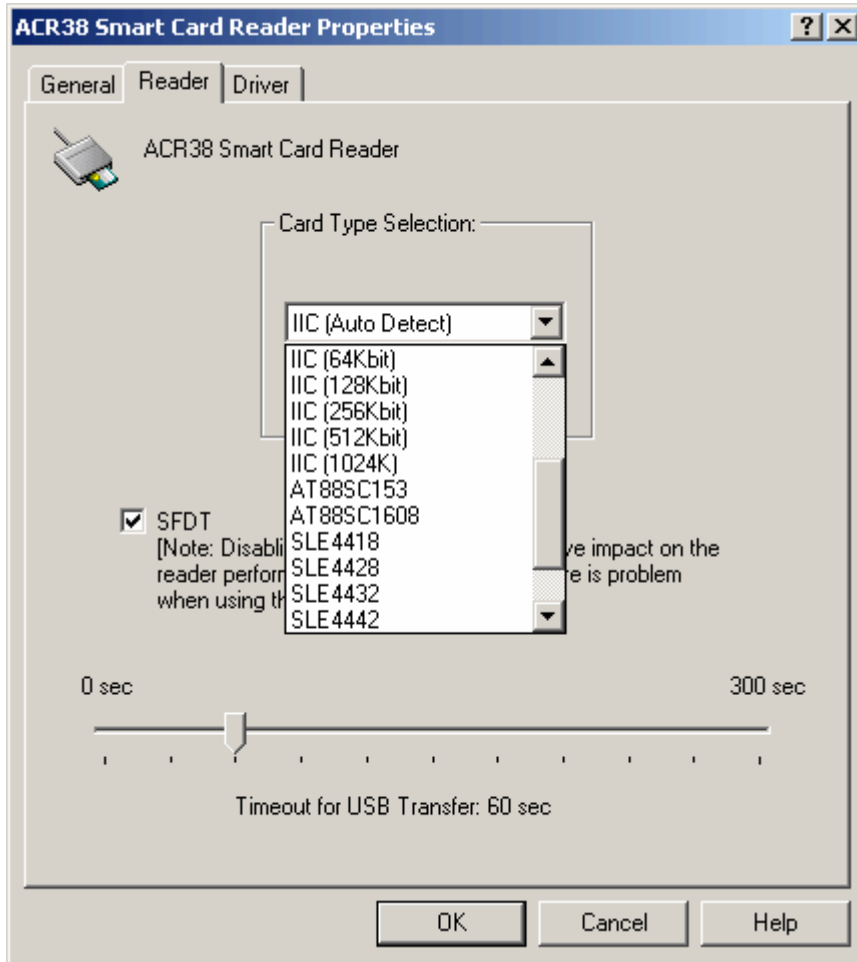
A code snippet for the program flow is given below to demonstrate how to select the memory card type in ACR38 FW1.12c:

```
SCARDCONTEXT hContext;  
SCARDHANDLE hCard;  
unsigned long dwActProtocol;  
SCARD_IO_REQUEST ioRequest;  
DWORD size = 64, SendLen = 6, RecvLen = 255, retCode;  
byte cardType;  
  
//Establish PC/SC Connection  
retCode = SCardEstablishContext (SCARD_SCOPE_USER, NULL, NULL, &hContext);  
  
//List all readers in the system  
retCode = SCardListReaders (hContext, NULL, readerName, &size);  
  
//Connect to the reader  
retCode = SCardConnect(hContext, readerName, SCARD_SHARE_SHARED,  
SCARD_PROTOCOL_T0, &hCard, &dwActProtocol);  
  
//Select Card Type  
unsigned char SendBuff[] = {0xFF,0xA4,0x00,0x00,0x01,cardType};  
retCode = SCardTransmit( hCard, &ioRequest, SendBuff, SendLen, NULL,  
RecvBuff, &RecvLen);  
  
//Disconnect from the reader  
retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD);  
  
//End the established context  
retCode = SCardReleaseContext(hContext);
```

### 2.2. Memory Card Type Selection for Firmware 1.10

#### 2.2.1. By Property Sheet

User could invoke the reader setting property sheet by selecting “property” of “ACR38 Smart Card Reader” device under the Device Manager. Figure 1 shows the reader setting property sheet.



**Figure 1:** ACR38 Reader Setting Property Sheet

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

### 2.2.2. By Programmatic Method

The card type can also be changed at 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 a 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 dwActiveProtocol;
    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);

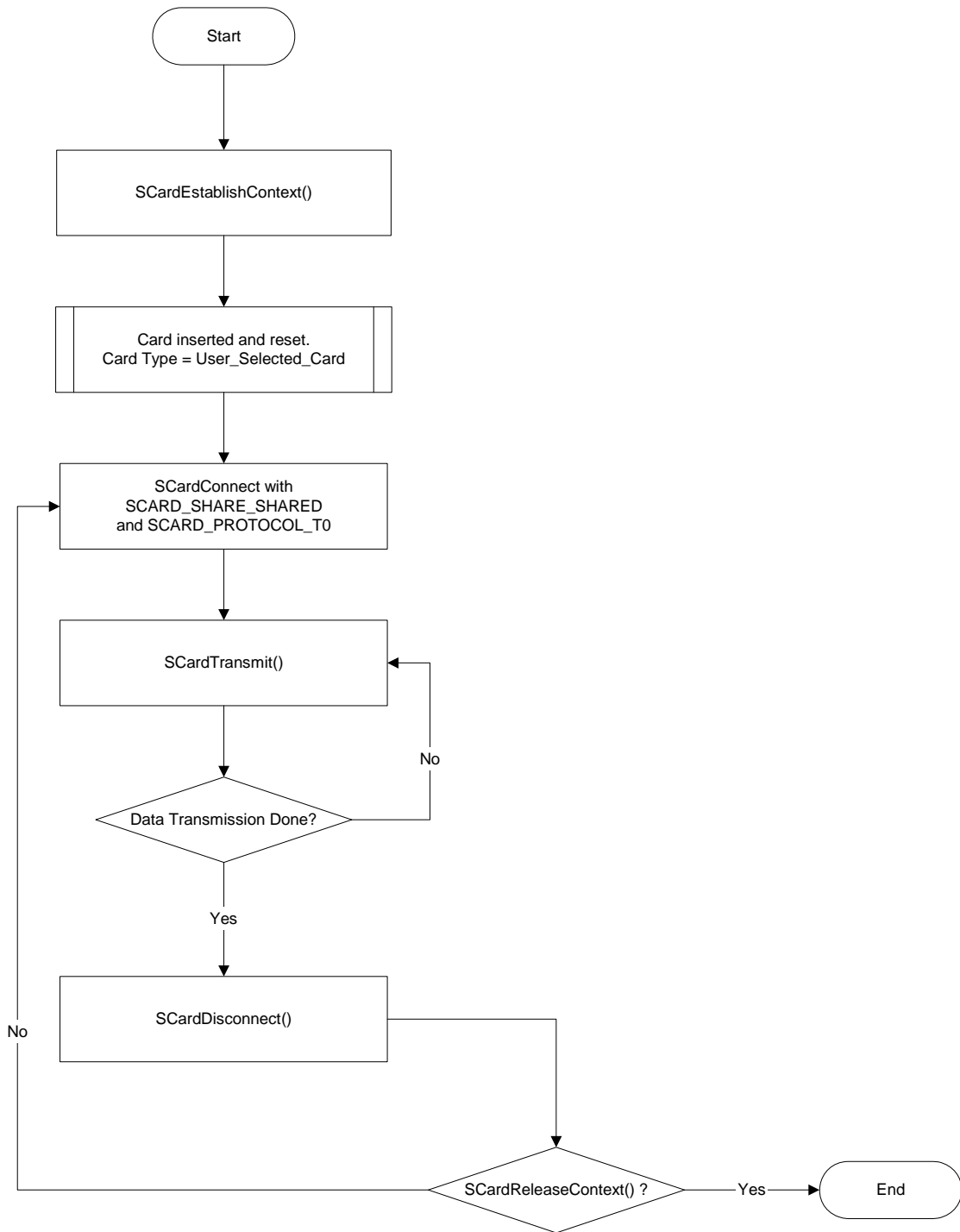
if (rv != SCARD_S_SUCCESS)
{
    // error handling ...
    return rv;
}

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
. . .
}
```



### 2.2.3. Logical Flow of Memory Card Functions





### 3.0. ACR38 Firmware 1.12c Memory Card Command Set

This section contains the Memory Card Command Set for ACR38 FW1.12c.

#### 3.1. Recollection card – 1, 2, 4, 8 and 18 Kbit I2C Card

##### 3.1.1. 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 specification.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF <sub>H</sub>	A4 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	01 <sub>H</sub>

Response Data Format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

##### 3.1.2. 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 <sub>H</sub>	01 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	

- Page size**
- = 03<sub>H</sub> for 8-byte page write
  - = 04<sub>H</sub> for 16-byte page write
  - = 05<sub>H</sub> for 32-byte page write
  - = 06<sub>H</sub> for 64-byte page write
  - = 07<sub>H</sub> for 128-byte page write



Response Data Format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.1.3. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF <sub>H</sub>	B0 <sub>H</sub>			

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.1.4. 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 <sub>H</sub>	D0 <sub>H</sub>							

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.2. Memory Card – 32, 64, 128, 256, 512, and 1024 Kbit I2C Card

#### 3.2.1. 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 specification.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

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

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

SW1, SW2 = 90<sub>H</sub> 00<sub>H</sub> if no error

#### 3.2.2. 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 <sub>H</sub>	01 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	

**Data** TPDU to be sent to the card

**Page size**

- = 03<sub>H</sub> for 8-byte page write
- = 04<sub>H</sub> for 16-byte page write
- = 05<sub>H</sub> for 32-byte page write
- = 06<sub>H</sub> for 64-byte page write
- = 07<sub>H</sub> for 128-byte page write



Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.2.3. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF <sub>H</sub>				

**INS** = B0<sub>H</sub> for 32,64,128,256,512kbit iic card

= 1011 000\*<sub>b</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.2.4. 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 <sub>H</sub>								

**INS** = D0<sub>H</sub> for 32,64,128,256,512kbit iic card

= 1101 000\*<sub>b</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.3. Memory Card – ATMEL AT88SC153

#### 3.3.1. 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 specification.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

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

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

#### 3.3.2. READ\_MEMORY\_CARD

Command Format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF <sub>H</sub>		00 <sub>H</sub>		

**INS** = B0<sub>H</sub> for reading zone 00<sub>b</sub>  
 = B1<sub>H</sub> for reading zone 01<sub>b</sub>  
 = B2<sub>H</sub> for reading zone 10<sub>b</sub>  
 = B3<sub>H</sub> for reading zone 11<sub>b</sub>  
 = B4<sub>H</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.3.3. 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 <sub>H</sub>		00 <sub>H</sub>						

**INS** = D0<sub>H</sub> for writing zone 00<sub>b</sub>

= D1<sub>H</sub> for writing zone 01<sub>b</sub>

= D2<sub>H</sub> for writing zone 10<sub>b</sub>

= D3<sub>H</sub> for writing zone 11<sub>b</sub>

= D4<sub>H</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.3.4. 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 <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>		03 <sub>H</sub>			

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

**P2** = 0000 00<sub>rp</sub><sub>b</sub>

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 <sub>H</sub>	

**SW1** = 90<sub>H</sub>

**SW2 (ErrorCnt)** = Error Counter. FF<sub>H</sub> indicates the verification is correct. 00<sub>H</sub> indicates the password is locked (exceed maximum number of retries). Other values indicate the current verification is failed.

### 3.3.5. 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 <sub>H</sub>	84 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.3.6. 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 <sub>H</sub>	82 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

**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<sub>H</sub> 00<sub>H</sub> if no error

## 3.4. Memory Card – ATMEL AT88C1608

### 3.4.1. 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 specification.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

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

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.4.2. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	Zone Address	Byte Address	MEM_L
FF <sub>H</sub>				

**INS** = B0<sub>H</sub> for reading user zone

= B1<sub>H</sub> for reading configuration zone or reading fuse

**Zone Address** = 0000 0A<sub>10</sub>A<sub>9</sub>A<sub>8</sub> b, where A<sub>10</sub> is the MSB of zone address

= don't care for reading fuse

**Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> 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<sub>H</sub> 00<sub>H</sub> if no error



### 3.4.3. 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 <sub>H</sub>								

- INS** = D0<sub>H</sub> for writing user zone  
= D1<sub>H</sub> for writing configuration zone or writing fuse
- Zone Address** = 0000 0A<sub>10</sub>A<sub>9</sub>A<sub>8</sub> b, where A<sub>10</sub> is the MSB of zone address  
= don't care for writing fuse
- Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> b is the memory address location of the memory card.  
= 1000 0000<sub>b</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.4.4. VERIFY\_PASSWORD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

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

- Pw(0),Pw(1),Pw(2)** Passwords to be sent to memory card.
- RP** = 0000 rp<sub>2</sub>p<sub>1</sub>p<sub>0</sub> b  
where the four bits “rp<sub>2</sub>p<sub>1</sub>p<sub>0</sub>” indicate the password to compare:  
r = 0: Write password,  
r = 1: Read password,  
p<sub>2</sub>p<sub>1</sub>p<sub>0</sub>: Password set number.  
(rp<sub>2</sub>p<sub>1</sub>p<sub>0</sub> = 0111 for the secure code).

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2 ErrorCnt
90 <sub>H</sub>	

- SW1** = 90<sub>H</sub>
- SW2 (ErrorCnt)** = Error Counter. FF<sub>H</sub> indicates the verification is correct. 00<sub>H</sub> indicates the password is locked (exceed maximum number of retries). Other values



indicate the current verification is failed.

### 3.4.5. 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 <sub>H</sub>	84 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.4.6. 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 <sub>H</sub>	82 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

**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<sub>H</sub> 00<sub>H</sub> if no error

## 3.5. Memory Card – SLE 4418 / SLE 4428 / SLE 5518 / SLE 5528

### 3.5.1. 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 specification



Command format (abData field in the PC\_to\_RDR\_XfrBlock)

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

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.5.2. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF <sub>H</sub>	B0 <sub>H</sub>			

**MSB Byte Address** = 0000 00A<sub>9</sub>A<sub>8</sub> b is the memory address location of the memory card.

**LSB Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.5.3. READ\_PRESENTATION\_ERROR\_COUNTER\_MEMORY\_CARD (SLE 4428 and SLE 5528)

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 <sub>H</sub>	B1 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	03 <sub>H</sub>

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

**ERRCNT** The value of the presentation error counter. FF<sub>H</sub> indicates the last verification is correct. 00<sub>H</sub> indicates the password is locked (exceed maximum number of retries). Other values indicate the last verification is failed.

**DUMMY** Two bytes dummy data read from the card.

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.5.4. READ\_PROTECTION\_BIT

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FF <sub>H</sub>	B2 <sub>H</sub>			

**MSB Byte Address** = 0000 00A<sub>9</sub>A<sub>8</sub> b is the memory address location of the memory card.

**LSB Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> 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.

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

For example, to read eight protection bits starting from memory 0x0010, the following pseudo-APDU should be issued:

0xFF 0xB2 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<sub>H</sub> 00<sub>H</sub> 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

### 3.5.5. 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 <sub>H</sub>	D0 <sub>H</sub>							

**MSB Byte Address** = 0000 00A<sub>9</sub>A<sub>8</sub> b is the memory address location of the memory card.

**LSB Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.5.6. 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 <sub>H</sub>	D1 <sub>H</sub>							

**MSB Byte Address** = 0000 00A<sub>9</sub>A<sub>8</sub> b is the memory address location of the memory card.

**LSB Byte Address** = A<sub>7</sub>A<sub>6</sub>A<sub>5</sub>A<sub>4</sub> A<sub>3</sub>A<sub>2</sub>A<sub>1</sub>A<sub>0</sub> 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.5.7. PRESENT\_CODE\_MEMORY\_CARD (SLE 4428 and SLE 5528)

To submit the secret code to the memory card to enable the write operation with the SLE 4428 and SLE 5528 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

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU						
CLA	INS	P1	P2	MEM_L	CODE	
					Byte 1	Byte 2
FF <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	02 <sub>H</sub>		

**CODE** Two bytes secret code (PIN)

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2
	ErrorCnt
90 <sub>H</sub>	

**SW1** = 90<sub>H</sub>

**SW2 (ErrorCnt)** = Error Counter. FF<sub>H</sub> indicates the verification is correct. 00<sub>H</sub> indicates the password is locked (exceed maximum number of retries). Other values indicate the current verification is failed.

## 3.6. Memory Card – SLE 4432 / SLE 4442 / SLE 5532 / SLE 5542

### 3.6.1. 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 specification.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF <sub>H</sub>	A4 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	06 <sub>H</sub>



Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.6.2. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF <sub>H</sub>	B0 <sub>H</sub>	00 <sub>H</sub>		

**Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> 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	PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

**BYTE x** Data read from memory card

**PROT y** Bytes containing the protection bits from protection memory

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> 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

### 3.6.3. 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.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FF <sub>H</sub>	B1 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	04 <sub>H</sub>

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

ERRCNT	DUMMY 1	DUMMY 2	DUMMY 3	SW1	SW2



- ERRCNT** The value of the presentation error counter. 07<sub>H</sub> indicates the last verification is correct. 00<sub>H</sub> indicates the password is locked (exceed maximum number of retries). Other values indicate the last verification is failed.
- DUMMY** Three bytes dummy data read from the card.
- SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.6.4. 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 <sub>H</sub>	B2 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	04 <sub>H</sub>

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

PROT 1	PROT 2	PROT3	PROT 4	SW1	SW2

- PROT y** Bytes containing the protection bits from protection memory
- SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> 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

### 3.6.5. 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 <sub>H</sub>	D0 <sub>H</sub>	00 <sub>H</sub>						

- Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> 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<sub>H</sub> 00<sub>H</sub> if no error



### 3.6.6. 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 <sub>H</sub>	D1 <sub>H</sub>	00 <sub>H</sub>						

**Byte Address** = 000A4 A3A2A1A0<sub>b</sub> (00<sub>H</sub> to 1F<sub>H</sub>) 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.6.7. PRESENT\_CODE\_MEMORY\_CARD (SLE 4442 and SLE 5542)

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 and write the bit to '0'
2. Present the specified code to the card
3. 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 <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	03 <sub>H</sub>			

**CODE** Three bytes secret code (PIN)

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2
	ErrorCnt
90 <sub>H</sub>	

**SW1** = 90<sub>H</sub>

**SW2 (ErrorCnt)** = Error Counter. 07<sub>H</sub> indicates the verification is correct. 00<sub>H</sub> indicates the password is locked (exceed maximum number of retries). Other values indicate the current verification is failed.



### 3.6.8. 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 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 <sub>H</sub>	D2 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	03 <sub>H</sub>			

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

SW1, SW2 = 90<sub>H</sub> 00<sub>H</sub> if no error

## 3.7. Memory Card – SLE 4406 / SLE 4436 / SLE 5536 / SLE 6636

### 3.7.1. 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 specification.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

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

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

SW1, SW2 = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.7.2. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF <sub>H</sub>	B0 <sub>H</sub>	00 <sub>H</sub>		

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<sub>H</sub> 00<sub>H</sub> if no error

### 3.7.3. 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 (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 (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU						
CLA	INS	P1	Byte Address	MEM_L	MODE	BYTE
FF <sub>H</sub>	D0 <sub>H</sub>	00 <sub>H</sub>		02 <sub>H</sub>		

**Byte Address** = Memory address location of the memory card.

**MODE** Specifies the write mode and backup option

00<sub>H</sub>: write

01<sub>H</sub>: write with carry

02<sub>H</sub>: write with backup enabled (SLE 4436, SLE 5536 and SLE 6636 only)

03<sub>H</sub>: write with carry and with backup enabled (SLE 4436, SLE 5536 and SLE 6636 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<sub>H</sub> 00<sub>H</sub> if no error

### 3.7.4. PRESENT\_CODE\_MEMORY\_CARD

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

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

The ACR38 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 <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	04 <sub>H</sub>	09 <sub>H</sub>			

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.7.5. AUTHENTICATE\_MEMORY\_CARD (SLE 4436, SLE 5536 and SLE 6636)

To read a card authentication certificate from a SLE 5536 or SLE 6636 card, the following actions are executed by the ACR38:

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

The authentication 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 <sub>H</sub>	84 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>							

**KEY** Key to be used for the computation of the authentication certificate:  
 00<sub>H</sub> : key 1 with no cipher block chaining  
 01<sub>H</sub> : key 2 with no cipher block chaining  
 80<sub>H</sub> : key 1 with cipher block chaining (SLE 5536 and SLE 6636 only)  
 81<sub>H</sub> : 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. Typical value is 160 clocks (A0<sub>H</sub>)

**BYTE 1...6** Card challenge data

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2
61 <sub>H</sub>	02 <sub>H</sub>

**SW1, SW2** = 61<sub>H</sub> 02<sub>H</sub> 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 <sub>H</sub>	C0 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	02 <sub>H</sub>

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<sub>H</sub> 00<sub>H</sub> if no error

### 3.8. Memory Card – SLE 4404

#### 3.8.1. 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 specification.



Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF <sub>H</sub>	A4 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	08 <sub>H</sub>

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.8.2. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF <sub>H</sub>	B0 <sub>H</sub>	00 <sub>H</sub>		

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.8.3. WRITE\_MEMORY\_CARD

This command is used 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 <sub>H</sub>	D0 <sub>H</sub>	00 <sub>H</sub>						

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.8.4. ERASE\_SCRATCH\_PAD\_MEMORY\_CARD

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

To erase error counter or user area, please use the VERIFY\_USER\_CODE command as specified in the Section 3.8.5.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF <sub>H</sub>	D2 <sub>H</sub>	00 <sub>H</sub>		00 <sub>H</sub>

**Byte Address** = Memory byte address location of the scratch pad.  
Typical value is 0x02.

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.8.5. VERIFY\_USER\_CODE

This command is used to submit User Code (2 bytes) to the inserted card. User Code is to enable the memory access of the card.

The following actions are executed:

1. Present the specified code to the card
2. Search a '1' bit in the presentation error counter and write the bit to '0'
3. Erase the presentation error counter. The User Error 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 <sub>H</sub>	20 <sub>H</sub>	04 <sub>H</sub>	08 <sub>H</sub>	02 <sub>H</sub>		

**Error Counter LEN** Length of presentation error counter in bits.

**Byte Address** Byte address of the key in the card.

**CODE** 2 bytes User Code

Response data format (abData field in the RDR\_to\_PC\_DataBlock)



SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error.  
= 63<sub>H</sub> 00<sub>H</sub> if there is no more retry chance

**Note:** After SW1SW2 = 0x9000 has been received, read back the User Error Counter can check whether the VERIFY\_USER\_CODE is correct. If User Error Counter is erased and equals to "0xFF", the previous verification is success.

### 3.8.6. VERIFY\_MEMORY\_CODE

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

The following actions are executed:

1. Present the specified code to the card
2. Search a '1' bit in the presentation error counter and write the bit to '0'
3. Erase the presentation error counter. Please note that Memory Error Counter cannot be erased.

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 <sub>H</sub>	20 <sub>H</sub>	40 <sub>H</sub>	28 <sub>H</sub>	04 <sub>H</sub>				

**Error Counter LEN** Length of presentation error counter in bits.  
**Byte Address** Byte address of the key in the card.  
**CODE** 4 bytes Memory Code

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error  
= 63<sub>H</sub> 00<sub>H</sub> if there is no more retry chance

**Note:** After SW1SW2 = 0x9000 has been received, read back the Application Area can check whether the VERIFY\_MEMORY\_CODE is correct. If all data in Application Area is erased and equals to "0xFF", the previous verification is success.

## 3.9. Memory Card – AT88SC101 / AT88SC102 / AT88SC1003

### 3.9.1. 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 specification.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)



Pseudo-APDU					
CLA	INS	P1	P2	Lc	Card Type
FF <sub>H</sub>	A4 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	09 <sub>H</sub>

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 3.9.2. READ\_MEMORY\_CARD

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF <sub>H</sub>	B0 <sub>H</sub>	00 <sub>H</sub>		

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.9.3. WRITE\_MEMORY\_CARD

This command is used 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 <sub>H</sub>	D0 <sub>H</sub>	00 <sub>H</sub>						

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.9.4. ERASE\_NON\_APPLICATION\_ZONE

This command is used 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:

1. ERASE\_APPLICATION\_ZONE\_WITH\_ERASE command as specified in Section 3.9.5
2. ERASE\_APPLICATION\_ZONE\_WITH\_WRITE\_AND\_ERASE command as specified in Section 3.9.6
3. VERIFY\_SECURITY\_CODE commands as specified in Section 3.9.7

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	Byte Address	MEM_L
FF <sub>H</sub>	D2 <sub>H</sub>	00 <sub>H</sub>		00 <sub>H</sub>

**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<sub>H</sub> 00<sub>H</sub> if no error

### 3.9.5. ERASE\_APPLICATION\_ZONE\_WITH\_ERASE

This command can be used in the following cases:

1. AT88SC101: To erase the data in Application Zone with EC Function Disabled
2. AT88SC102: To erase the data in Application Zone 1
3. AT88SC102: To erase the data in Application Zone 2 with EC2 Function Disabled
4. AT88SC1003: To erase the data in Application Zone 1
5. AT88SC1003: To erase the data in Application Zone 2 with EC2 Function Disabled
6. AT88SC1003: To erase the data in Application Zone 3

The following actions are executed for this command:

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



Command 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 <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>							

**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.

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

**MEM\_L** Length of the Erase Key. Please refer to the table above for the correct value.

**CODE** N bytes of Erase Key

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> 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.

### 3.9.6. ERASE\_APPLICATION\_ZONE\_WITH\_WRITE\_AND\_ERASE

This command can be used in the following cases:

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

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

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



Command 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 <sub>H</sub>	20 <sub>H</sub>	80 <sub>H</sub>		04 <sub>H</sub>				

**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 <sub>H</sub>
AT88SC102	9C <sub>H</sub>
AT88SC1003	5C <sub>H</sub>

**CODE** 4 bytes Erase Key

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error.

= 63<sub>H</sub> 00<sub>H</sub> 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\_WRITE\_AND\_ERASE is correct. If all data in Application Zone is erased and equals to "0xFF", the previous verification is success.

### 3.9.7. VERIFY\_SECURITY\_CODE

This command is used 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:

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

Command 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 <sub>H</sub>	20 <sub>H</sub>	08 <sub>H</sub>	0A <sub>H</sub>	02 <sub>H</sub>		

**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<sub>H</sub> 00<sub>H</sub> if no error.  
= 63<sub>H</sub> 00<sub>H</sub> 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.

### 3.9.8. BLOWN\_FUSE

This command is used to blow the fuse of the inserted card. The fuse can be EC\_EN Fuse, EC2EN Fuse, Issuer Fuse or Manufacturer's Fuse.

**Note:** The blowing of 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			
					Ffuse Bit Addr (High)	Fuse Bit Addr (Low)	State of FUS Pin	State of RST Pin
FF <sub>H</sub>	05 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	04 <sub>H</sub>			01 <sub>H</sub>	00 <sub>H</sub> or 01 <sub>H</sub>

**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 always be 0x01.

**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 <sub>H</sub>	80 <sub>H</sub>	01 <sub>H</sub>
	EC_EN Fuse	05 <sub>H</sub>	C9 <sub>H</sub>	01 <sub>H</sub>
	Issuer Fuse	05 <sub>H</sub>	E0 <sub>H</sub>	01 <sub>H</sub>
AT88SC102	Manufacturer Fuse	05 <sub>H</sub>	B0 <sub>H</sub>	01 <sub>H</sub>
	EC2EN Fuse	05 <sub>H</sub>	F9 <sub>H</sub>	01 <sub>H</sub>
	Issuer Fuse	06 <sub>H</sub>	10 <sub>H</sub>	01 <sub>H</sub>
AT88SC1003	Manufacturer Fuse	03 <sub>H</sub>	F8 <sub>H</sub>	00 <sub>H</sub>
	EC2EN Fuse	03 <sub>H</sub>	FC <sub>H</sub>	00 <sub>H</sub>
	Issuer Fuse	03 <sub>H</sub>	E0 <sub>H</sub>	00 <sub>H</sub>

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



### 3.10. Other Commands Access via PC\_to\_RDR\_XfrBlock

#### 3.10.1. GET\_READER\_INFORMATION

This command returns relevant information about the particular ACR38 Firmware 1.12c 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.

**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.

Command format (abData field in the PC\_to\_RDR\_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	P2	Lc
FF <sub>H</sub>	09 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	10 <sub>H</sub>

Response data format (abData field in the RDR\_to\_PC\_DataBlock)

FIRMWARE	MAX_C	MAX_R	C_TYPE	C_SEL	C_STAT

**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 ACR38U-112c. 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

Refer to the next section for the correspondence between these bits and the respective card types.

**C\_SEL** The currently selected card type. 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<sub>H</sub>: no card inserted

01<sub>H</sub>: card inserted, not powered up

03<sub>H</sub>: card powered up



### 3.11. Supported Card Types

The following table summarizes the card type returned by GET\_READER\_INFORMATION correspond with the respective card type.

Byte	Card Type
00 <sub>H</sub>	Auto-select T=0 or T=1 communication protocol
01 <sub>H</sub>	I2C memory card (1k, 2k, 4k, 8k and 16k bits)
02 <sub>H</sub>	I2C memory card (32k, 64k, 128k, 256k, 512k and 1024k bits)
03 <sub>H</sub>	Atmel AT88SC153 secure memory card
04 <sub>H</sub>	Atmel AT88SC1608 secure memory card
05 <sub>H</sub>	Infineon SLE 4418 and SLE 4428
06 <sub>H</sub>	Infineon SLE 4432 and SLE 4442
07 <sub>H</sub>	Infineon SLE 4406, SLE 4436 and SLE 5536
08 <sub>H</sub>	Infineon SLE 4404
09 <sub>H</sub>	Atmel AT88SC101, AT88SC102 and AT88SC1003
0C <sub>H</sub>	MCU-based cards with T=0 communication protocol
0D <sub>H</sub>	MCU-based cards with T=1 communication protocol

### 3.12. Response Error Codes

The following table summarizes the possible error code returned by the ACR38 (FW 1.12c)

Error Code	Status
FF <sub>n</sub>	SLOTERROR_CMD_ABORTED
FE <sub>n</sub>	SLOTERROR_ICC_MUTE
FD <sub>n</sub>	SLOTERROR_XFR_PARITY_ERROR
FC <sub>n</sub>	SLOTERROR_XFR_OVERRUN
FB <sub>n</sub>	SLOTERROR_HW_ERROR
F8 <sub>n</sub>	SLOTERROR_BAD_ATR_TS
F7 <sub>n</sub>	SLOTERROR_BAD_ATR_TCK
F6 <sub>n</sub>	SLOTERROR_ICC_PROTOCOL_NOT_SUPPORTED
F5 <sub>n</sub>	SLOTERROR_ICC_CLASS_NOT_SUPPORTED
F4 <sub>n</sub>	SLOTERROR_PROCEDURE_BYTE_CONFLICE
F3 <sub>n</sub>	SLOTERROR_DEACTIVATED_PROTOCOL
F2 <sub>n</sub>	SLOTERROR_BUSY_WITH_AUTO_SEQUENCE
E0 <sub>n</sub>	SLOTERROR_CMD_SLOT_BUSY



## 4.0. ACR38 Firmware 1.10 Memory Card Command Set

### 4.1. Memory Card – 1, 2, 4, 8, and 16 Kbit I2C Card

#### 4.1.1. 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 reader is powered off.

Send Buffer Format

SCardTransmit Send Buffer					
CLA	INS	P1	P2	Lc (P3)	Page size
FF <sub>H</sub>	01 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	

**Page size**

- = 03<sub>H</sub> for 8-byte page write
- = 04<sub>H</sub> for 16-byte page write
- = 05<sub>H</sub> for 32-byte page write
- = 06<sub>H</sub> for 64-byte page write
- = 07<sub>H</sub> for 128-byte page write

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

#### 4.1.2. READ\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Byte Address		MEM_L (P3)
		MSB (P1)	LSB (P2)	
FF <sub>H</sub>	B0 <sub>H</sub>			

**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

**BYTE x** Data read from memory card

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



### 4.1.3. WRITE\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	Byte Address		MEM_L (P3)	Byte 1	....	....	Byte n
		MSB (P1)	LSB (P2)					
FF <sub>H</sub>	D0 <sub>H</sub>							

**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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

## 4.2. Memory Card – 32, 64, 128, 256, 512, and 1024 Kbit I2C Card

### 4.2.1. 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 reader is powered off.

Send Buffer Format

SCardTransmit Send Buffer					
CLA	INS	P1	P2	Lc (P3)	Page size
FF <sub>H</sub>	01 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	

**Data** TPDU to be sent to the card

**Page size**  
 = 03<sub>H</sub> for 8-byte page write  
 = 04<sub>H</sub> for 16-byte page write  
 = 05<sub>H</sub> for 32-byte page write  
 = 06<sub>H</sub> for 64-byte page write  
 = 07<sub>H</sub> for 128-byte page write

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



### 4.2.2. READ\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Byte Address		MEM_L (P3)
		MSB (P1)	LSB (P2)	
FF <sub>H</sub>				

**INS** = B0<sub>H</sub> for 32,64,128,256,512kbit iic card  
 = 1011 000\*<sub>b</sub> 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 Buffer Format

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

**BYTE x** Data read from memory card

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.2.3. WRITE\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	Byte Address		MEM_L (P3)	Byte 1	....	....	Byte n
		MSB (P1)	LSB (P2)					
FF <sub>H</sub>								

**INS** = D0<sub>H</sub> for 32,64,128,256,512kbit iic card  
 = 1101 000\*<sub>b</sub> 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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



### 4.3. Memory Card – ATMEL AT88SC153

#### 4.3.1. READ\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)
FF <sub>H</sub>		00 <sub>H</sub>		

- INS**
- = B0<sub>H</sub> for reading zone 00 b
  - = B1<sub>H</sub> for reading zone 01 b
  - = B2<sub>H</sub> for reading zone 10 b
  - = B3<sub>H</sub> for reading zone 11 b
  - = B4<sub>H</sub> 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

**BYTE x** Data read from memory card

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

#### 4.3.2. WRITE\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)	Byte 1	....	....	Byte n
FF <sub>H</sub>		00 <sub>H</sub>						

- INS**
- = D0<sub>H</sub> for writing zone 00 b
  - = D1<sub>H</sub> for writing zone 01 b
  - = D2<sub>H</sub> for writing zone 10 b
  - = D3<sub>H</sub> for writing zone 11 b
  - = D4<sub>H</sub> 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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.3.3. VERIFY\_PASSWORD

Send Buffer Format

SCardTransmit Send Buffer							
CLA	INS	P1	P2	Lc (P3)	Pw(0)	Pw(1)	Pw(2)
FF <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>		03 <sub>H</sub>			

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

**P2** = 0000 00rp<sub>b</sub>

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

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.3.4. INITIALIZE\_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (P3)	Q(0)	Q(1)	...	Q(7)
FF <sub>H</sub>	84 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

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

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



### 4.3.5. VERIFY\_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (p3)	Ch(0)	Ch(1)	...	Ch(7)
FF <sub>H</sub>	82 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

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

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

## 4.4. Memory Card – ATMEL AT88SC1608

### 4.4.1. READ\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Zone Address (P1)	Byte Address (P2)	MEM_L (P3)
FF <sub>H</sub>				

**INS** = B0<sub>H</sub> for reading user zone

= B1<sub>H</sub> for reading configuration zone or reading fuse

**Zone Address** = 0000 0A10A9A8<sub>b</sub>, where A10 is the MSB of zone address

= don't care for reading fuse

**Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> is the memory address location of the memory card.

= 1000 0000<sub>b</sub> 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

**BYTE x** Data read from memory card

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



#### 4.4.2. WRITE\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer									
CLA	INS	Zone Address (P1)	Byte Address (P2)	MEM_L (P3)	Byte 1	....	....	....	Byte n
FF <sub>H</sub>									

- INS** = D0<sub>H</sub> for writing user zone  
= D1<sub>H</sub> for writing configuration zone or writing fuse
- Zone Address** = 0000 0A10A9A8<sub>b</sub>, where A10 is the MSB of zone address  
= don't care for writing fuse
- Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> is the memory address location of the memory card.  
= 1000 0000<sub>b</sub> 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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

#### 4.4.3. VERIFY\_PASSWORD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (P3)	Data			
FF <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	04 <sub>H</sub>	RP	Pw(0)	Pw(1)	Pw(2)

- Pw(0),Pw(1),Pw(2)** Passwords to be sent to memory card.
- RP** = 0000 rp2p1p0<sub>b</sub>  
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

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



#### 4.4.4. INITIALIZE\_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (P3)	Q(0)	Q(1)	...	Q(7)
FF <sub>H</sub>	84 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

**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

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

#### 4.4.5. VERIFY\_AUTHENTICATION

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	P2	Lc (P3)	Q1(0)	Q1(1)	...	Q1(7)
FF <sub>H</sub>	82 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>				

**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

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.5. Memory Card – SLE 4418 / SLE 4428 / SLE 5518 / SLE5528

#### 4.5.1. READ\_MEMORY\_WITH\_PROTECT\_BIT\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Byte Address		MEM_L (P3)
		MSB (P1)	LSB (P2)	
FF <sub>H</sub>	B0 <sub>H</sub>			

**MSB Byte Address** = 0000 00A9A8<sub>b</sub> is the memory address location of the memory card.



**LSB Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> is the memory address location of the memory card.

**MEM\_L** Length of data to be read from the memory card. (Max. allowable size is EC<sub>H</sub>.)

Response Buffer Format

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

**BYTE x** Data read from memory card

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

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> 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:

$$L = 1 + \text{INT}(N/8)$$

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

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

'0' byte is write protected

'1' byte can be written

### 4.5.2. READ\_MEMORY\_WITHOUT\_PROTECT\_BIT\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	Byte Address		MEM_L (P3)
		MSB (P1)	LSB (P2)	
FF <sub>H</sub>	B2 <sub>H</sub>			

**MSB Byte Address** = 0000 00A9A8<sub>b</sub> is the memory address location of the memory card.

**LSB Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> 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

**BYTE x** Data read from memory card

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



### 4.5.3. WRITE\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	Byte Address		MEM_L (P3)	Byte 1	....	....	Byte N
		MSB (P1)	LSB (P2)					
FF <sub>H</sub>	D0 <sub>H</sub>							

**MSB Byte Address** = 0000 00A9A8<sub>b</sub> is the memory address location of the memory card.

**LSB Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> 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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.5.4. 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 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)					
FF <sub>H</sub>	D1 <sub>H</sub>							

**MSB Byte Address** = 0000 00A9A8<sub>b</sub> is the memory address location of the memory card.

**LSB Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> 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

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



#### 4.5.5. PRESENT\_CODE\_MEMORY\_CARD (SLE 4428 and SLE 5528)

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
FF <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	02 <sub>H</sub>		

**CODE** Two bytes secret code (PIN)

Response Buffer Format

ERRCNT	CODE		SW1	SW2
	Byte 1	Byte 2		

**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<sub>H</sub> 00<sub>H</sub> if no error

If the correct code has been presented to the card, the value of ERRCNT is FF<sub>H</sub> and the value of CODE is identical to the code data specified in the command.

#### 4.5.6. READ\_PRESENTATION\_ERROR\_COUNTER\_MEMORY\_CARD (SLE 4428 and SLE 5528)

To read the presentation error counter for the secret code.

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	P2	MEM_L (P3)
FF <sub>H</sub>	B1 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>

Response Buffer Format

SCardTransmit Receive Buffer				
ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

**ERRCNT** The value of the presentation error counter.

**DUMMY** Three bytes dummy data read from the card.

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



## 4.6. Memory Card – SLE 4432 / SLE 4442 / SLE 5532 / SLE 5542

### 4.6.1. READ\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)
FF <sub>H</sub>	B0 <sub>H</sub>	00 <sub>H</sub>		

**Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> 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

**BYTE x** Data read from memory card

**PROT y** Bytes containing the protection bits from protection memory

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> 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

### 4.6.2. WRITE\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)	Byte 1	...	....	Byte N
FF <sub>H</sub>	D0 <sub>H</sub>	00 <sub>H</sub>						

**Byte Address** = A7A6A5A4 A3A2A1A0<sub>b</sub> 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 Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



### 4.6.3. 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'.

Send Buffer Format

SCardTransmit Send Buffer								
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)	Byte 1	....	....	Byte N
FF <sub>H</sub>	D1 <sub>H</sub>	00 <sub>H</sub>						

Byte Address = 000A4 A3A2A1A0<sub>b</sub> (00<sub>H</sub> to 1F<sub>H</sub>) 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

SW1,SW2 = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.6.4. 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
FF <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	03 <sub>H</sub>			

CODE Three bytes secret code (PIN)

Response Buffer Format

ERRCNT	CODE			SW1	SW2
	Byte 1	Byte 2	Byte 3		

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<sub>H</sub> 00<sub>H</sub> if no error

If the correct code has been presented to the card, the value of ERRCNT is 07<sub>H</sub> and the value of CODE is identical to the code data specified in the command.



#### 4.6.5. READ\_PRESENTATION\_ERROR\_COUNTER\_MEMORY\_CARD (SLE 4442 and SLE 5542)

To read the presentation error counter for the secret code.

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	P2	MEM_L (P3)
FF <sub>H</sub>	B1 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>

Response Buffer Format

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

**ERRCNT** The value of the presentation error counter.

**DUMMY** Three bytes dummy data read from the card.

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

#### 4.6.6. 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 have been 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
FF <sub>H</sub>	D2 <sub>H</sub>	00 <sub>H</sub>	01 <sub>H</sub>	03 <sub>H</sub>			

Response Buffer Format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



## 4.7. Memory Card – SLE 4406 / SLE 4436 / SLE 5536 / SLE 6636

### 4.7.1. READ\_MEMORY\_CARD

Send Buffer Format

SCardTransmit Send Buffer				
CLA	INS	P1	Byte Address (P2)	MEM_L (P3)
FF <sub>H</sub>	B0 <sub>H</sub>	00 <sub>H</sub>		

**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

**BYTE x** Data read from memory card

**SW1,SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.7.2. 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
FF <sub>H</sub>	D0 <sub>H</sub>	00 <sub>H</sub>		02 <sub>H</sub>		

**Byte Address** = Memory address location of the memory card.

**LEN** = 5 + MEM\_L

**MODE** Specifies the write mode and backup option

00<sub>H</sub> : write

01<sub>H</sub> : write with carry

02<sub>H</sub> : write with backup enabled (SLE 4436, SLE 5536 and SLE 6636 only)

03<sub>H</sub> : write with carry and with backup enabled (SLE 4436, SLE 5536 and SLE 6636 only)

**BYTE** Byte value to be written to the card

Response data format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error

### 4.7.3. 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 ACR38 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
FF <sub>H</sub>	20 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	04 <sub>H</sub>				

**ADDR** Byte address of the presentation counter in the card

**CODE** Three bytes secret code (PIN)

Response data format

SCardTransmit Receive Buffer	
SW1	SW2

**SW1, SW2** = 90<sub>H</sub> 00<sub>H</sub> if no error



#### 4.7.4. 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 ACR38:

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 ACR38 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
FF <sub>H</sub>	84 <sub>H</sub>	00 <sub>H</sub>	00 <sub>H</sub>	08 <sub>H</sub>							

- KEY** Key to be used for the computation of the authentication certificate:
  - 00<sub>H</sub>: key 1 with no cipher block chaining
  - 01<sub>H</sub>: key 2 with no cipher block chaining
  - 80<sub>H</sub>: key 1 with cipher block chaining (SLE 5536 and SLE 6636 only)
  - 81<sub>H</sub>: 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

- 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<sub>H</sub> 00<sub>H</sub> if no error