Advanced Card Systems Ltd.
Card & Reader Technologies

ACR100I SIMFlash II (CCID)

SIMFlash with Embedded MIFARE®

Reference Manual V1.00

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

Table of Contents

List of Figures

1.0. Introduction

ACR100I SIMFlash II (CCID) is not an average smart card reader. Its memory storage comes with NAND flash memory for your high capacity data storage needs. This can be partitioned up to three sections as desired by the user. The ACR100I SIMFlash II (CCID) also has an embedded MIFARE® Classic (1K) chip for various contactless card functions, such as logical and physical access.

1.1. SIM-sized Smart Card Reader

With its steadfast support for ISO 7816 microprocessor smart cards, the ACR100I SIMFlash II (CCID) is a powerful reader. It works with most memory cards and microprocessor cards with the T=0 and T=1 protocol.

1.2. Memory Storage Device

Aside from its smart card reading ability, the ACR100I SIMFlash II (CCID) is also a storage device. Users can now store personal files in a secured way with the NAND flash onboard the ACR100I. Various options for partitioning the flash drive are also available: Private/Security, Public and CD-ROM/Autorun, and hidden areas are all accessible in the same device.

1.3. Contactless Feature

ACR100I SIMFlash II (CCID)'s embedded MIFARE Classic (1K) chip allows the reader to be used in contactless applications – allowing flexibility in using the device in various functions.

1.4. Ease of Integration

With ACR100I SIMFlash II (CCID) being compliant with CCID (Chip/Smart Card Interface Devices) and PC/SC standards, it is easier to integrate in a computer-based environment by eliminating driver installation prior to use. In addition, ACR100I SIMFlash II (CCID) may now be used on mobile devices running the Android™ platform with versions 3.1 and above.

With its wide array of features, ACR100I SIMFlash II (CCID) can be used in various application areas, such as PKI (Public Key Infrastructure), network security and GSM management.

2.0. Features

- USB Combo Device – Works as a smart card reader and mass storage
- USB 2.0 High Speed Interface
- Plug and Play – CCID support brings utmost mobility
- Extractable USB Connector
- Smart Card Reader:
  - Supports ISO 7816 Class A, B and C (5 V, 3 V, 1.8 V) SIM-sized cards
  - Supports microprocessor cards with T=0 or T=1 protocol
  - Supports memory cards using Synchronous Card APDU
  - Supports Specification 11.11-compliant GSM 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)
- Flash Drive:
  - Built-in NAND flash memory
  - Maximum of built-in 8 GB flash memory
  - Division of up to three partitions (Private/Security, Public and CD-ROM/Autorun, and Hidden)
- Contactless Feature:
  - Embedded MIFARE Classic (1K) chip
- Supports Android™ 3.1 and above[1]
- Compliant with the following standards:
  - ISO 7816
  - CE
  - FCC
  - VCCI
  - PC/SC
  - CCID
  - Microsoft® WHQL
  - RoHS 2
  - REACH

[1] *PC/SC and CCID support are not applicable*

3.0. System Block Diagram

The USB Hub Controller is the communication interface between the computer and the MCU of the smart card and the flash memory via USB port connection. The flash memory is available for the end-user to use as storage. In Windows Explorer, the device is detected as a removable disk. The ACR100I is powered from the USB port without other external power supply.

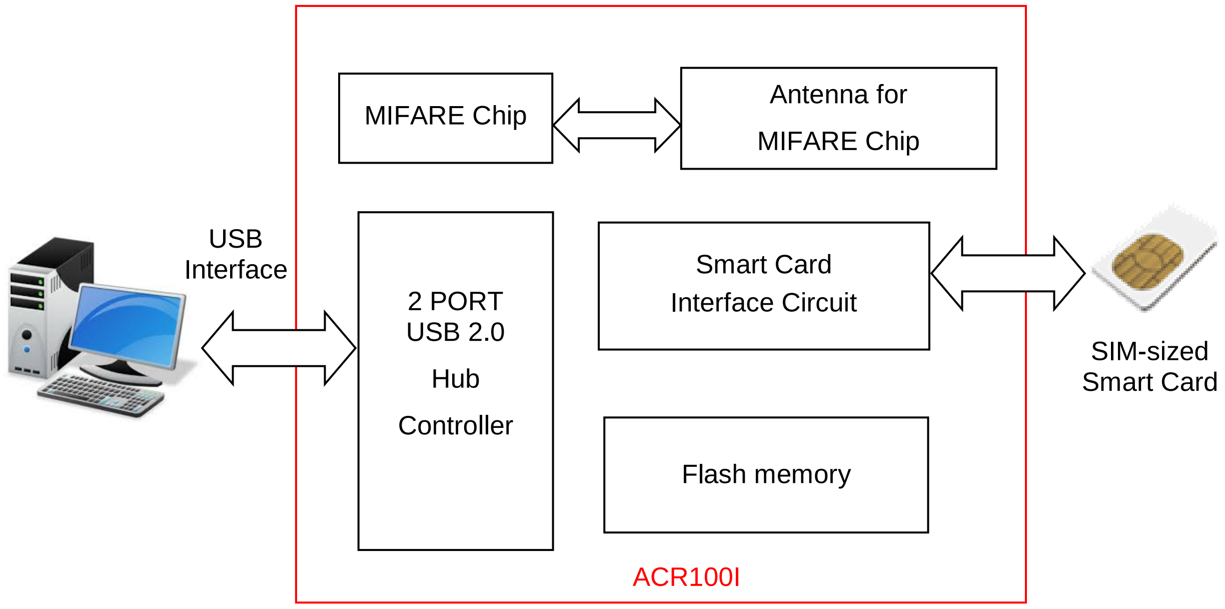

Figure 1: ACR100I System Block Diagram

4.0. Power Supply

The ACR100I requires a voltage of 5 V DC, 300 mA regulated power supply, and gets the power supply from computer.

4.1. Status LED

Bicolor LED in front of the reader indicates the activation status of the smart card and flash memory interface.

GREEN LED:

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

Indicates that the ACR100I 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 is a communication between ACR100I and smart card.

RED LED:

Lighting up

Indicates there is a communication between ACR100I and flash memory.

4.2. Embedded MIFARE Chip

The ACR100I has an embedded MIFARE Classic chip with a memory size of 1K.

5.0. Smart Card Interface

The interface between the ACR100I and the inserted smart card follows the specifications of ISO 7816-3.

5.1. Smart Card Power Supply VCC (C1)

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

5.2. 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 ACR100I. The electrical specifications of this contact are identical to those of the single RST (at contact C2).

5.3. Card Type Selection

The controlling computer has to always select the card type through the proper command sent to the ACR100I prior to activating the inserted card. This includes both the memory cards and MCU-based cards. For MCU-based cards, the reader allows to select the preferred protocol, T=0 or T=1, however, this selection is only accepted and carried out by the reader through the PPS when the card inserted in the reader supports both protocol types. Whenever an MCU-based card supports only one protocol type, T=0 or T=1, the reader automatically uses that protocol type, regardless of the protocol type selected by the application.

5.4. 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).

5.5. Card Tearing Protection

The ACR100I 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 ACR100I and are immediately deactivated when the card is being removed. As a rule, however, to avoid any electrical damage, **a card should only be removed from the reader while it is powered down.**

Note: *The ACR100I never switches on the power supply to the inserted card by itself. This must be explicitly be done by controlling the computer through the proper command sent to the reader.*

6.0. USB Interface

The connection of the ACR100I to a computer through a USB port follows a USB Standard.

6.1. Communication Parameters

The ACR100I is connected to a computer through USB as specified in the USB Specification 2.0. The ACR100I is working in high-speed mode, i.e. 480 Mbps.

Pin	Signal	Function
1	V_{BUS}	+5 V power supply for the reader (Max 500 mA, Normal 300 mA)
2	D-	Differential signal transmits data between ACR100I and computer.
3	D+	Differential signal transmits data between ACR100I and computer.
4	GND	Reference voltage level for power supply

6.2. Endpoints

The ACR100I uses the following endpoints to communicate with the host computer.

6.2.1. Smart Card Reader

Control Endpoint	For setup and control purpose
Bulk OUT	For command to be sent from host to ACR100I (data packet size is 64 bytes).
Bulk IN	For response to be sent from ACR100I to host (data packet size is 64 bytes).
Interrupt IN	For card status message to be sent from ACR100I to host (data packet size is 8 bytes).

6.2.2. Mass Storage

Control Endpoint	For setup and control purpose
Bulk OUT	For command to be sent from host to Device (data packet size is 512 bytes)
Bulk IN	For response to be sent from Device to host (data packet size is 512 bytes)

7.0. Communication Protocol

ACR100I shall interface with the host through USB connection. A specification, namely CCID, has been released within the industry defining such protocol for the USB chip-card interface devices. CCID covers all the protocols required for operating smart cards and PIN.

The configurations and usage of USB endpoints on ACR100I shall follow CCID Section 3. An overview is summarized below:

1. **Control Commands** are sent on a control pipe (default pipe). These include class-specific requests and USB standard requests. Commands that are sent on the Default Pipe Report Information back to the host on the default pipe.
2. **CCID Events** are sent on the interrupt pipe.
3. **CCID Commands** are sent on *Bulk-OUT* endpoint. Each command sent to ACR100I has an associated ending response. Some commands can also have intermediate responses.
4. **CCID Responses** are sent on *Bulk-IN* endpoint. All commands sent to ACR100I have to be sent synchronously. (i.e. bMaxCCIDBusySlots is equal to 1 for ACR100I)

The supported CCID features by ACR100I are indicated in its Class Descriptor:

Offset	Field	Size	Value	Description
0	*bLength*	1	36h	Size of this descriptor, in bytes
1	*bDescriptorType*	1	21h	CCID Functional Descriptor type
2	*bcdCCID*	2	0100h	CCID Specification Release Number in binary-coded decimal
4	*bMaxSlotIndex*	1	00h	One slot is available on ACR100I
5	*bVoltageSupport*	1	07h	ACR100I can supply 1.8 V, 3.0 V and 5.0 V to its slot
6	*dwProtocols*	4	00000003h	ACR100I supports T=0 and T=1 Protocol
10	*dwDefaultClock*	4	00000FA0h	Default ICC clock frequency is 4 MHz
14	*dwMaximumClock*	4	00000FA0h	Maximum supported ICC clock frequency is 4 MHz
18	*bNumClockSupported*	1	00h	Does not support manual setting of clock frequency
19	*dwDataRate*	4	00002A00h	Default ICC I/O data rate is 10752 bps
23	*dwMaxDataRate*	4	0001F808h	Maximum supported ICC I/O data rate is 250000 bps
27	*bNumDataRatesSupported*	1	00h	Does not support manual setting of data rates
28	*dwMaxIFSD*	4	00000Feh	Maximum IFSD supported by ACR100I for protocol T=1 is 254
32	*dwSynchProtocols*	4	00000000h	ACR100I does not support synchronous card
36	*dwMechanical*	4	00000000h	ACR100I does not support special mechanical characteristics

Offset	Field	Size	Value	Description
40	*dwFeatures*	4	00010030h	ACR100I supports the following features: • Automatic ICC clock frequency change according to parameters • Automatic baud rate change according to frequency and FI,DI parameters • TPDU level exchange with ACR100I
44	*dwMaxCCIDMessageLength*	4	0000010Fh	Maximum message length accepted by ACR100I is 271 bytes
48	*bClassGetResponse*	1	00h	Insignificant for TPDU level exchanges
49	*bClassEnvelope*	1	00h	Insignificant for TPDU level exchanges
50	*wLCDLayout*	2	0000h	No LCD
52	*bPINSupport*	1	00h	No PIN Verification
53	*bMaxCCIDBusySlots*	1	01h	Only 1 slot can be simultaneously busy

7.1. Command to the ACR100I

7.1.1. CCID Command Pipe Bulk-OUT Messages

ACR100I shall follow the *CCID Bulk-OUT Messages* as specified in CCID section 4. In addition, this specification defines some extended commands for operating additional features. This section lists the CCID Bulk-OUT messages to be supported by ACR100I.

7.1.1.1. PC_to_RDR_IccPowerOn

Activate the card slot and return ATR from the card

Offset	Field	Size	Value	Description
0	*bMessageType*	1	62h	
1	*dwLength*	4	00000000h	Size of extra bytes of this message
2	*bSlot*	1		Identifies the slot number for this command
5	*bSeq*	1		Sequence number for command
6	*bPowerSelect*	1		Voltage that is applied to the ICC 00h – Automatic Voltage Selection 01h – 5 volts 02h – 3 volts
7	*abRFU*	2		Reserved for future use

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

7.1.1.2. PC_to_RDR_IccPowerOff

Deactivate the card slot.

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

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

7.1.1.3. PC_to_RDR_GetSlotStatus

Gets the current status of the slot.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	65h	
1	*dwLength*	4	00000000h	Size of extra bytes of this message
5	*bSlot*	1		Identifies the slot number for this command
6	*bSeq*	1		Sequence number for command

Offset	Field	Size	Value	Description
7	*abRFU*	3		Reserved for future use

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

7.1.1.4. PC_to_RDR_XfrBlock

Transfer data block to the ICC.

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

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

7.1.1.5. PC_to_RDR_GetParameters

Get slot parameters.

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

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

7.1.1.6. PC_to_RDR_ResetParameters

Reset slot parameters to default value.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	6Dh	
1	*DwLength*	4	00000000h	Size of extra bytes of this message
5	*BSlot*	1		Identifies the slot number for this command

Offset	Field	Size	Value	Description
6	*BSeq*	1		Sequence number for command
7	*AbRFU*	3		Reserved for future use

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

7.1.1.7. PC_to_RDR_SetParameters

Set slot parameters.

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

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

Offset	Field	Size	Value	Description
10	*bmFindexDindex*	1		B7-4 – FI – Index into the Table 7 in ISO/IEC 7816-3:1997 selecting a clock rate conversion factor. B3-0 – DI – Index into the Table 8 in ISO/IEC 7816-3:1997 selecting a baud rate conversion factor.
11	*bmTCCKST0*	1		B0 – 0b, B7-2 – 000000b B1 – Convention used (b1=0 for direct, b1=1 for inverse) ***Note:** The CCID ignores this bit.*
12	*bGuardTimeT0*	1		Extra Guardtime between two characters. Add 0 to 254 etu to the normal guardtime of 12 etu. FFh is the same as 00h.
13	*bWaitingIntegerT0*	1		WI for T=0 used to define WWT

Offset	Field	Size	Value	Description
14	*bClockStop*	1		ICC Clock Stop Support 00h = Stopping the Clock is not allowed 01h = Stop with Clock signal Low 02h = Stop with Clock signal High 03h = Stop with Clock either High or Low

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

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

13	*BWaitingIntegerT1*	1		B7-4 = BWI values 0-9 valid
				B3-0 = CWI values 0-Fh valid

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

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

7.1.2. CCID Bulk-IN Messages

The Bulk-IN messages are used in response to the Bulk-OUT messages. ACR100I shall follow the *CCID Bulk-IN Messages* as specified in CCID Section 4. This section lists the CCID Bulk-IN messages to be supported by ACR100I.

7.1.2.1. RDR_to_PC_DataBlock

This message is sent by ACR100I in response to *PC_to_RDR_IccPowerOn*, *PC_to_RDR_XfrBlock* and *PC_to_RDR_Secure* messages.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	80h	Indicates that a data block is being sent from the CCID
1	*dwLength*	4		Size of extra bytes of this message
5	*bSlot*	1		Same value as in Bulk-OUT message
6	*bSeq*	1		Same value as in Bulk-OUT message
7	*bStatus*	1		Slot status register as defined in CCID Section 4.2.1
8	*bError*	1		Slot error register as defined in CCID Section 4.2.1 and its specifications in Section 5.2.8
9	*bChainParameter*	1	00h	RFU (TPDU exchange level)
10	*AbData*	Byte array		This field contains the data returned by the CCID

7.1.2.2. RDR_to_PC_SlotStatus

This message is sent by ACR100I in response to *PC_to_RDR_IccPowerOff*, *PC_to_RDR_GetSlotStatus*, *PC_to_RDR_Abort* messages and class specific *ABORT* request.

Offset	Field	Size	Value	Description
0	*bMessageType*	1	81h	
1	*dwLength*	4	00000000h	Size of extra bytes of this message
5	*bSlot*	1		Same value as in Bulk-OUT message
6	*bSeq*	1		Same value as in Bulk-OUT message
7	*bStatus*	1		Slot status register as defined in CCID Section 4.2.1
8	*bError*	1		Slot error register as defined in CCID Section 4.2.1 and its specifications in Section 5.2.8
9	*bClockStatus*	1		Value = 00h Clock running 01h Clock stopped in state L 02h Clock stopped in state H 03h Clock stopped in an unknown state All other values are RFU.

7.1.2.3. RDR_to_PC_Parameters

This message is sent by ACR100I in response to *PC_to_RDR_GetParameters*, *PC_to_RDR_ResetParameters* and *PC_to_RDR_SetParameters* messages

Offset	Field	Size	Value	Description
0	*bMessageType*	1	82h	
1	*dwLength*	4		Size of extra bytes of this message
5	*bSlot*	1		Same value as in Bulk-OUT message
6	*bSeq*	1		Same value as in Bulk-OUT message
7	*bStatus*	1		Slot status register as defined in CCID Section 4.2.1
8	*bError*	1		Slot error register as defined in CCID Section 4.2.1 and its specification in Section 5.2.8
9	*bProtocolNum*	1		Specifies what protocol data structure follows. 00h = Structure for protocol T=0 01h = Structure for protocol T=1 The following values are reserved for future use. 80h = Structure for 2-wire protocol 81h = Structure for 3-wire protocol 82h = Structure for I2C protocol
10	*abProtocolDataStructure*	Byte array		Protocol Data Structure as summarized in Section 5.2.3.

7.1.3. Commands Accessed via PC_to_RDR_XfrBlock

7.1.3.1. GET_READER_INFORMATION

This command returns relevant information about the particular ACR100I model and the current operating status, such as the firmware version number; the maximum data length of a command and response; the supported card types; and whether a card is inserted and powered up or not.

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

Command format (abData field in the PC_to_RDR_XfrBlock)

Pseudo-APDU				
CLA	INS	P1	P2	Lc
FFh	09h	00h	00h	10h

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 ACR100I. This data field is a bitmap with each bit representing a particular card type. A bit set to '1' means the corresponding card type is supported by the reader and can be selected with the *SELECT_CARD_TYPE* command. The bit assignment is as follows:

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

See ***Appendix A*** 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: 00h: no card inserted 01h: card inserted, not powered up 03h: card powered up

7.2. Mass Storage

Mass Storage Device Class specifies all protocols required for data transaction between the Host (computer) and storage devices. The configurations and usage of USB endpoints on ACR100I shall follow *Mass Storage Class Bulk-Only Transport* in Section 3 (Protocol Code) of the *USB Mass Storage Device Specification*. This document is available at: www.usb.org.

An overview of this specification is summarized below:

1. **Control Commands** are sent on control pipe (default pipe). It is shared with the CCID interface.
2. **Data-Out Command Protocol** uses the *Bulk-OUT* endpoint to transfer data from the host to the device.
3. **Data-In Command Protocol** uses the *Bulk-IN* endpoint to transfer data from the device or to return status about the device.

Appendix A. Supported Card Types

The following table is a list of the card types returned by *GET_READER_INFORMATION* corresponding with the respective card type code:

Card type code	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 SLE4442
07h	Infineon® SLE4406, SLE4436 and SLE5536
08h	Infineon® SLE4404
09h	Atmel® AT88SC101, AT88SC102 and AT88SC1003
0Ch	MCU-based cards with T=0 communication protocol
0Dh	MCU-based cards with T=1 communication protocol

Appendix B. Response Error Codes

The following table summarizes the possible error code returned by ACR100I:

Error Code	Status
FFh	SLOTERROR_CMD_ABORTED
FEh	SLOTERROR_ICC_MUTE
FDh	SLOTERROR_XFR_PARITY_ERROR
FCh	SLOTERROR_XFR_OVERRUN
FBh	SLOTERROR_HW_ERROR
F8h	SLOTERROR_BAD_ATR_TS
F7h	SLOTERROR_BAD_ATR_TCK
F6h	SLOTERROR_ICC_PROTOCOL_NOT_SUPPORTED
F5h	SLOTERROR_ICC_CLASS_NOT_SUPPORTED
F4h	SLOTERROR_PROCEDURE_BYTE_CONFLICE
F3h	SLOTERROR_DEACTIVATED_PROTOCOL
F2h	SLOTERROR_BUSY_WITH_AUTO_SEQUENCE
E0h	SLOTERROR_CMD_SLOT_BUSY