Keywords: DS8007,smart card,DS5002,DS5250,microcontroller,secure microcontroller,uC,multiprotocol,ISO 7816,EMV,Integrated Circuit Card, IC card,POS terminal,banking terminal,ATM,payment terminal,PIN pad,access control,pay tv,set top box,STB

APPLICATION NOTE 4029

The DS8007 and Smart Card Interface Fundamentals

Dec 05, 2007

Abstract: The DS8007 is a multiprotocol, low-cost, dual, smart card interface that supports all ISO 7816, EMV™, and GSM11-11 requirements. This one mixed-signal peripheral manages all the details of the interface between a microcontroller and two, independent smart cards. This application note describes some of the fundamentals of smart cards and how to communicate with them. Software is provided that uses the DS8007 to interface a smart card with a DS5002 Secure Microprocessor.

Overview

What is a smart card? A smart card is generally defined as any pocket-sized card containing an embedded integrated circuit. Because of the embedded integrated circuit, smart cards are sometimes referred to as Integrated Circuit Cards, or ICCs. **Figure 1** shows a typical example. Used in widely varying applications, these cards replace the familiar payment (debit or credit) cards that use a magnetic stripe to store information about the card account. The transition to smart cards in payment applications is occurring primarily because of increased functionality, and especially because of the improved security possible with this technology. These latter capabilities must, however, be evaluated against the smart card's higher cost.

Figure 1. Smart card example.

The integrated circuits embedded in smart cards can be either simple, nonvolatile memory devices or something as sophisticated as a microcontroller capable of performing complex operations. A simple nonvolatile memory device in a payment card can replace the magnetic stripe for storing data. In many

such devices, the memory is combined with additional logic to restrict access to some, or all of the memory. However, the real power of smart cards lies in the ability of an embedded microcontroller to perform data processing and/or encryption functions. This processing ability allows enhanced security capabilities. Yet as complexity rises, so does the cost of the card. The cost of a smart card with an embedded processor ranges between $7.00 to $15.00 (USD), while a payment card with a magnetic stripe can cost as little as $0.75[1] (USD). This higher cost of smart cards has slowed the universal conversion from simpler technology, but as the requirements for security increase, so will the need for smart cards.

The DS8007 provides all electrical signals necessary to physically interface a microcontroller with two separate smart cards. The device contains a dedicated internal sequencer that controls automatic card activation and deactivation, and an ISO UART for data communication. Charge pumps and voltage regulators allow the device to operate from a 2.7V to 6.0V supply voltage, and to produce two independent smart card supply voltages, either of which can be 1.8V, 3.0V, or 5V. Communication with the microcontroller is provided by a standard, parallel 8-bit bus that carries either data in a nonmultiplexed configuration or data and address in multiplexed configuration.

Smart Card Details

While the most familiar smart card form factor is a credit-card-size device, the term "smart card" also applies to a Subscriber Identification Module (SIM), which is about the size of a postage stamp and frequently found in cellular phones. This SIM form factor is also used in payment terminals to provide the terminal with specific payment-system data and detailed application information. The card portion of the credit-card-size device is generally made of polyvinyl chloride (PVC), and is typically embossed with the account number and possibly an expiration date. Regardless of the form factor, all electromechanical specifications are based on the ISO 7816 series of standards. In addition, a consortium of the EuroCard®, MasterCard®, and Visa® (EMV) corporations has developed a set of standards specifically addressing smart cards and their application to payment systems. The EMV specifications are generally based on the ISO 7816 documents.

Smart Card Contacts

The number, location, and function of the contacts on a smart card are explicitly defined by the standards mentioned above. The location of the integrated circuit on a standard smart card and the contact dimensions are shown in **Figure 2**. There are eight possible contact locations defined by ISO 7816. Of these eight locations, five are presently used in EMV applications. The names and functions of these contacts are given in **Table 1**. While contact C6 is defined as V_{PP} by ISO 7816, this programming voltage is not used on current cards, according to the EMV specification. Contacts C4 and C8 are not used, and need not be physically present. A more detailed discussion of the individual contacts specified in the EMV specifications follows.

Table 1. Smart Card Contacts

Contact Name	Contact Function
C1	Supply voltage to card (V_{CC})
C2	Reset (RST)
C3	Clock (CLK)
C4	Provided on the DS8007; not used in EMV
C5	Ground (GND)
C6	V_{PP}; not used in EMV
C7	Input/output (I/O)
C8	Provided on the DS007; not used in EMV

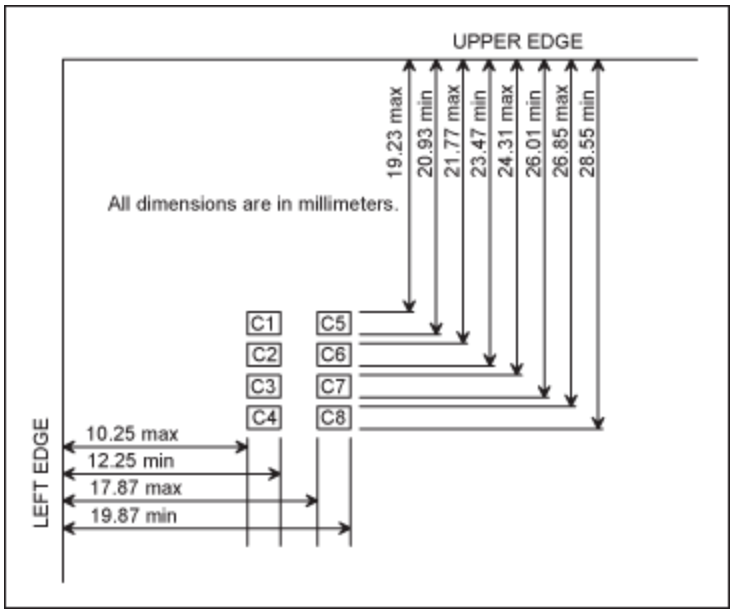

Figure 2. Contact dimensions and location.

V_{CC} Contact (C1)

This contact provides the power-supply voltage to the card. The original specifications for V_{CC} only included 5V DC ±10%. However, there is at present a phased migration to lower voltage cards. Cards that only support this original specification, called class A cards, will be replaced by class AB or class ABC cards by the end of June 2009. The V_{CC} specifications for these card classes follow:

Class A Cards: $4.5V \le V_{CC} \le 5.5V$ at ≤ 50mA
Class AB Cards: $2.70V \le V_{CC} \le 3.3V$ at ≤ 50mA
Class ABC Cards: $1.62V \le V_{CC} \le 1.98V$ at ≤ 30mA

The DS8007 card interface contains charge pumps and voltage regulators that can supply the appropriate voltages for any of the three card classes when the device is operating from a 2.6V to 6.0V power supply.

I/O Contact (C7)

The I/O contact on a smart card is used as an input (reception mode) to receive data from the terminal or as an output (transmission mode) to transmit data to the terminal.

In reception mode, the card will recognize valid data when the input conforms to the following specification.

Class A Cards

Input High Voltage: $0.7 \times V_{CC} \le V_{IH} \le V_{CC}$
Input Low Voltage: $0.0 \le V_{IL} \le 0.8V$
Rise Time/Fall Time: ≤ 1µs

Class AB or ABC Cards

Input High Voltage: $0.7 \times V_{CC} \le V_{IH} \le V_{CC}$
Input Low Voltage: $0.0 \le V_{IL} \le 0.2 \times V_{CC}$
Rise Time/Fall Time: ≤ 1µs

In transmission mode, the I/O contact will provide signal levels as follows:

Class A Cards

Output High Voltage: $0.7 \times V_{CC} \le V_{OH} \le V_{CC}$, $-20\mu A < I_{OH} < 0$, V_{CC} = min
Output Low Voltage: $0.0 \le V_{OL} \le 0.4V$, $0 < I_{OL} < 1mA$, V_{CC} = min
Rise Time/Fall Time: ≤ 1.0µs

Class AB or ABC Cards

Output High Voltage: $0.7 \times V_{CC} \le V_{OH} \le V_{CC}$, $-20\mu A < I_{OH} < 0$, V_{CC} = min
Output Low Voltage: $0.0 \le V_{OL} \le 0.15 \times V_{CC}$, $0 < I_{OL} < 1mA$, V_{CC} = min
Rise Time/Fall Time: ≤ 1.0µs

The EMV specifications state that the smart card's I/O contact driver will be set to reception mode unless the I/O contact is transmitting.

CLK Contact (C3)
The CLK contact is an input sourced by the interfacing terminal, i.e., by the DS8007. This signal is used to control the timing of data transfer during the transaction process. The frequency range is specified between 1.0MHz and 5.0MHz. This contact has the following electrical specifications:

Class A Cards

Input High Voltage: $V_{CC} - 0.7 \le V_{IH} \le V_{CC}$
Input Low Voltage: $0.0 \le V_{IL} \le 0.5V$
Rise Time/Fall Time: ≤ 9% of clock period

Class AB or ABC Cards

Input High Voltage: $0.7 \times V_{CC} \le V_{IH} \le V_{CC}$
Input Low Voltage: $0.0 \le V_{IL} \le 0.2 \times V_{CC}$
Rise Time/Fall Time: ≤ 9% of clock period

RST Contact (C2)
The RST contact is an input to the card sourced by the interfacing terminal. This signal is active-low, and will cause an asynchronous reset of the card. This contact has the same electrical specifications as the CLK contact, but with a maximum rise and fall time of 1.0µs.

As seen from the above specifications, the terminal that interfaces with a smart card must provide various supply voltages and signal levels. The interface specifications also require that the terminal withstand a short-circuit between any two of the card's contacts. For these reasons, using a dedicated device to provide the necessary supply voltages and signal levels is clearly more advantageous than assembling a large number of discrete analog ICs for the task. The DS8007 is such a dedicated device. Besides the analog circuitry necessary to provide these functions, it also contains a FIFO and other digital-control logic to provide the state sequencing and timing necessary to support a complete card session.

Terminal Interface Requirements

All card sessions consist of the following steps.

1. Insertion of the card into the terminal; connection and activation of the contacts
2. Reset of the card; establishment of communication between the terminal and the card (ATR sequence—see below)
3. Execution of the transaction(s)
4. Deactivation of the contacts; removal of the card

Following the initial reset of the card after insertion, the card responds with a series of characters called the Answer to Reset, or ATR. This series of characters establishes the initial communication details, including the specific protocol, bit timing, and data transfer details for all subsequent communications. While subsequent data transfers can change certain communications parameters, the ATR establishes initial communications conditions. The ATR is discussed extensively in the section(s) below.

Individual Character Details

During the interface between the smart card and the terminal, information is communicated serially over the bidirectional I/O contact. The bit duration is defined as the Elemental Time Unit, or ETU. The time period for the ETU has a direct linear relationship to the clock signal provided by the terminal on the CLK contact. The bit timing of the characters during the ATR is called the Initial ETU. This Initial ETU is defined by the following equation:

$$\text{Initial ETU} = 372/f \text{ seconds} \qquad \text{(Eq.1)}$$

where f is the frequency of the clock signal in hertz.

After the ATR, the bit duration is called the Current ETU, which is a function of parameters F and D and the clock frequency. (Parameters F and D are discussed in more detail in the TA1 Character section below.)

$$\text{Current ETU} = F/(Df) \text{ seconds} \qquad \text{(Eq.2)}$$

where f is the frequency of the clock signal in hertz.

Each character in any communication consists of ten bits, resulting in a duration of 10 ETUs. The first bit of a character is called the Start Bit; it is always low. Preceding the Start Bit, the I/O line is kept in its default high state. The last bit of a character is the Parity Bit; it is either high or low as determined by the source, so that the total number of 1s in the character is even. An illustration of this bit pattern is shown in **Figure 3**.

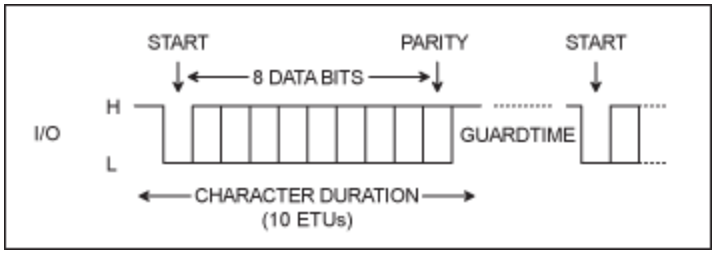

Figure 3. A 10-bit character frame.

Smart Card Communication Protocol

Within the ISO 7816 specifications, four bits are used to select the communications protocol for a card session. Presently, 2 of the possible 16 protocols are in use. They are referred to as T=0 and T=1. Both protocols are half-duplex (one direction at a time), asynchronous communications. The T=0 protocol is a character-based format, while T=1 is a block-based format. All EMV-compliant smart cards must support the T=0 or T=1 protocols, while terminals must support both.

Immediately after a card is inserted into a terminal and while all contacts are maintained in a "low" state, supply voltage is applied to the card's V_{CC} contact. After the terminal verifies that the voltage is steady and within the specified limits, the terminal's I/O contact driver is placed in reception mode and a clock signal is applied to the card's CLK contact. Within 200 cycles of the initiation of the clock signal, the terminal will have placed its I/O line in reception mode, and the card will put its I/O line in transmission mode. After an interval of between 40,000 and 45,000 clock periods, the terminal applies an active-high signal on the card's RST contact. Between 400 and 40,000 clock periods later, the card responds with a series of characters called the ATR. The ATR includes information detailing how subsequent communication will be conducted, including the T=0 or T=1 protocol selection. If no protocol is specified, T=0 is assumed. (The complete details of the ATR and the information contained within are described below.)

Answer to Reset (ATR)

After initially being reset by the terminal, the EMV smart card responds with a string of characters known as the Answer to Reset, or ATR. These characters consist of an initial character, TS, followed by a maximum of 32 additional characters. Together, these characters provide information to the terminal about how to communicate with the card for the remainder of the session. Each character is described in the following sections.

The contents of the ATR defined by the EMV specification for protocol T=0 are given in **Table 2**, and for protocol T=1 in **Table 3**.

Table 2. Basic EMV ATR for T=0 Only

Character	Value	Remarks
TS	'3B' or '3F'	Indicates direct (3B) or inverse (3F) convention.
T0	'6x	TB1 and TC1 present, TA1 and TD1 absent; x indicates the number of historical bytes present.
TB1	'00'	V_{PP} is not required.
TC1	'00' to 'FF'	Indicates the amount of extra guard time required. Value 'FF' has a special meaning. (See TC1 description below.)

Table 3. Basic EMV ATR for T=1 Only

Character	Value	Remarks
TS	'3B' or '3F'	Indicates direct (3B) or inverse (3F) convention.
T0	'Ex'	TB1, TC1, and TD1 present, TA1 is absent; x indicates the number of historical bytes present.
TB1	'00'	V_{PP} is not required.
TC1	'00' to 'FF'	Indicates the amount of extra guard time required.
TD1	'81'	TA2, TB2, and TC2 absent; TD2 present; T=1 to be used.
TD2	'31'	TA3 and TB3 present; TC3 and TD3 absent; T=1 to be used.
TA3	'10' to 'FE'	Returns IFSI, which indicates the initial value for the card's information field size and IFSC of 16 bytes to 254 bytes.
TB3	m.s. nibble* '0' to '4'; l.s. nibble '0' to '5'	BWI = 0 to 4 CWI = 0 to 5
TCK		Check character. Exclusive ORing of all ATR bytes from T0 to TCK inclusive is null.

*Note: m.s. nibble = most significant nibble; l.s. nibble = least significant nibble.

TS Initial Character

The first character of the ATR sequence is defined as the initial character, TS. By virtue of its bit pattern, this character synchronizes information and defines the polarity of all subsequent characters. The first four bits of TS consist of a low start bit, followed by two high bits, followed by an additional low bit. This fixed-bit pattern allows timing synchronization. The following three bits are either all high to indicate direct convention, or all low to indicate inverse convention. For direct convention, a high state on the I/O line is equivalent to logic 1, and the data is transmitted least significant bit first. For the inverse convention, a low state on the I/O line is equivalent to logic 1, and the data is transferred most significant bit first. While the specifications allow inverse convention, EMV recommends that the direct convention be used for all current card designs. The final three bits are two low bits followed by a high bit. The last bit in this, or any other 10-bit character frame, is the parity bit; it will be set or cleared to make the number of 1s in the frame an even number.

T0 Format Character

The second character of the ATR sequence is defined as the Format Character, and is called T0. This character contains two parts, both of which determine what characters are contained in the remaining ATR sequence. The most significant four bits are referred to as Y1, and they indicate whether TA1, TB1, TC1, or TD1 will be transmitted. For each logic 1 of Y1, the presence of the respective character is determined as follows:

Bit 8 (msb) = 1 indicates character TD1 will be transmitted
Bit 7 = 1 indicates character TC1 will be transmitted
Bit 6 = 1 indicates character TB1 will be transmitted
Bit 5 = 1 indicates character TA1 will be transmitted

The least significant four bits of T0 are referred to as K. These bits determine the number, 0 to 15, of "historical bytes" that will be contained in the remaining ATR sequence. Historical bytes convey general information about the card such as the card manufacturer, the chip in the card, the masked ROM in the chip, or the card's state of life. Neither the ISO 7816 nor EMV specifications define precisely what (if any) information is conveyed.

As can be seen in Table 2 above, Y1 bits b7 and b6 are high and bits b8 and b5 are low ('6x'). This

indicates that TC1 and TB1 will be transmitted, and characters TA1 and TD1 will not (as indicated in the table). For protocol T=0, characters TB1 and TC1 complete the basic ATR sequence. In Table 3, bit 8 of Y1 is also high, so character TD1 will also be transmitted for protocol T=1.

TA1 Character
While the character TA1 is not transmitted in the basic EMV ATR response for either the T=0 or T=1 protocols, it is defined in the ISO 7816 specifications for other communications. When used, TA1 is broken into upper and lower nibbles. The upper nibble determines the clock-rate conversion factor, F, that is used to modify the frequency of the clock signal. The lower nibble determines D, the bit-rate-adjustment factor that can be used to adjust the bit duration subsequent to the ATR. Use of these parameters is shown in Equation 2 above. The default values of F = 372 and D = 1 are used for the Initial ETU value during the ATR, and will continue to be used during subsequent exchanges unless changed outside the basic ATR.

TB1 Character
The TB1 character conveys information on the smart card's programming voltage requirements. Bits b1 to b5 (called PI1) convey the programming voltage, and bits b6 and b7 (called II) convey the maximum programming current required by the smart card. For the basic ATR, TB1 = '00' indicates that the V_{PP} pin is not connected in the smart card.

TC1 Character
The TC1 character conveys the value of N, which determines the extra guard time to be added between consecutive characters sent to the smart card from the terminal. This value does not apply to characters sent from the card to the terminal, or to two characters sent in opposite directions. N is a binary number representing the additional ETUs to be added as extra guard time. When TC1 = 'FF', the minimum delay between characters should be used. For protocol T=0, this is 12 ETUs, and 11 for T=1. The value of N can be anything between 0 and 255; if TC1 is not returned in the ATR, the terminal will continue as if a value of 00 had been received. Since this value can add time to character transmission, it should be minimized to speed transactions.

TD1 Character
The TD1 character indicates if any further interface bytes are to be transmitted, and if so, which protocol will be used. The character TD1 is a specific instance of the generalized character, TDx. The most significant nibble of TDx indicates whether TA(x + 1), TB(x + 1), TC(x + 1), or TD(x + 1) will be transmitted. For each logic 1, the presence of the respective character in subsequent transmissions is determined as follows:

Bit 8 (msb) = 1 indicates character TD(x + 1) will be transmitted
Bit 7 = 1 indicates character TC(x + 1) will be transmitted
Bit 6 = 1 indicates character TB(x + 1) will be transmitted
Bit 5 = 1 indicates character TA(x + 1) will be transmitted

The least significant nibble of the TD1 character (TDx generalized) contains either the value 0x0 or 0x1, indicating protocol T=0 or T=1 respectively.

If protocol T=0 is used, the character TD1 will not be included in the ATR sequence; protocol T=0 will be used for all subsequent transmissions. If protocol T=1 is used, TD1 will be included and will contain the value of 0x81. This latter value indicates that TD2 will be present and protocol T=1 will be used for all subsequent transmissions.

TA2 Character
While the character TA2 is not transmitted in the basic EMV ATR response for either the T=0 or T=1

protocols, it is defined in the ISO 7816 specifications. The presence or absence of TA2 determines whether the smart card will operate in specific mode or negotiable mode, respectively, following the ATR. The absence of TA2 indicates that the negotiable mode of operation will be used.

TB2 Character
While the character TB2 is not transmitted in the basic EMV ATR response for either the T=0 or T=1 protocols, it is defined in the ISO 7816 specifications. The character TB2 conveys PI2, which determines the value of programming voltage required by the smart card. The value of PI1 in character TB1 is superceded when the character TB2 is present.

TC2 Character
While the character TC2 is not transmitted in the basic EMV ATR response for either the T=0 or T=1 protocols, it is defined in the ISO 7816 specifications. When present, TC2 is specific to protocol type T=0. TC2 conveys the work waiting-time integer (WI) that determines the maximum interval between the leading edge of the start bit of any character sent by the smart card and the leading edge of the start bit of the previous character sent either by the card or the terminal. The value of the work waiting time is given as:

$$\text{Work Waiting Time} = 960 \times D \times WI \qquad \text{(Eq.3)}$$

where D is the bit-rate adjustment factor (see description in TA1 above).

When TC2 is not contained in the ATR sequence, the default value of WI = 0x0A is assumed.

TD2 Character
The TD2 character has the same function as the TD1 character. For details, see the TD1 description above. In Table 3 for protocol T=1, TD2 is present and contains the value 0x31. This value indicates that: TA3 and TB3 will be present, TC3 and TD3 will be absent, and the protocol type will be T=1.

TA3 Character
The TA3 character conveys the Information Field Size Integer (IFSI) for the smart card. IFSI determines the Information Field Size for the smart card which is the maximum length of the Information Field (INF) of blocks that can be received by the card. The Field Size can be any value between 0x01 and 0xFE. Values of 0x0 and 0xFF are reserved for future use. In the basic ATR and using the T=1 protocol, TA3 will have a value in the range of 0x10 to 0xFE, thus indicating an IFSC in the range of 16 to 254 bytes. For an ATR not containing TA3, the terminal will assume a default value of 0x20.

TB3 Character
The TB3 character indicates the value of the Character Waiting Time Integer (CWI) and the Block Waiting Time Integer (BWI) used to compute the Character Waiting Time (CWT) and Block Waiting Time (BWT). The least significant nibble of TB3 (b1 to b4) indicates the value of CWI; the most significant nibble (b5 to b8) indicates the value of BWI. In the basic ATR for the T=1 protocol, the TB3 character will have the least significant nibble in the range of 0 to 5 (CWI = 0 to 5), and the most significant nibble in the range 0 to 4 (BWI = 0 to 4).

TC3 Character
While the character TC3 is not transmitted in the basic EMV ATR response for either the T=0 or T=1 protocols, it is defined in the ISO 7816 specifications. When TC3 is present, it indicates the type of block-error detection to be used. When TC3 is not present, the default longitudinal redundancy check (LRC) is the block-error checking used.

TCK Character
The TCK character is the check character, and has a value that allows the integrity of the data sent in the ATR to be verified. The value of TCK can be anything, as long as the exclusive ORing of all bytes from T0 to TCK inclusive is zero. TCK is not used for T=0, but will be returned in the ATR in all other cases.

ATR Summary
After the necessary parameters have been transferred from the card to the terminal following the terminal's reception of the last character in the ATR sequence, any necessary adjustments to the interface parameters can be made to the DS8007. Further communications can then commence.

Application Protocol Data Unit (APDU)

As previously indicated, the next phase of a card session is the Execution of the Transaction(s). The specific operations performed during a transaction depend on the type of card and account (credit, debit, etc.) and the user's request. Regardless of the specific operations, the transactions are accomplished by issuing commands from the terminal to the smart card. The smart card performs the requested operation(s) and potentially communicates a result. The card's operation can be as simple as reading a location in memory or as complex as performing a cryptographic operation. Regardless of the operation, the communication between the terminal and the card is conducted using Application Protocol Data Units, or APDUs.

To run an application, the smart card and terminal must exchange information. This sharing of information is accomplished in a command-response data exchange. The terminal creates and sends a command to the smart card, which then interprets the command and sends a response. This command-response message pair is known as an Application Protocol Data Unit (APDU). A specific command message sent by the terminal (C-APDU) will have a specific response message from the card (R-APDU). These messages are referred to as APDU command-response pairs. The EMV specification details the format of both of these message types, and their formats are described below.

C-APDU Format
The terminal initiates all Command APDUs. They consist of a required 4-byte header followed by an optional body of variable length that can contain data. The number of data bytes contained in the C-APDU is specified in the command byte Lc; the number of bytes which the terminal expects to receive from the card's response is specified in the command byte Le. **Table 4** shows the C-APDU format, and the characters are described in **Table 5**.

Table 4. Command APDU Structure

CLA	INS	P1	P2	Lc	Data	Le
←Mandatory Header→				←Conditional Body→		

Table 5. Command APDU Content Description

Code	Description	Length
CLA	Class of instruction	1
INS	Instruction code	1
P1	Instruction parameter 1	1
P2	Instruction parameter 2	1
Lc	Number of bytes present in command date field	0 or 1
Data	String of data bytes sent in command (= Lc)	Variable
Le	Maximum number of data bytes expected in data field of response	0 or 1

The first byte of the Command APDU is defined as the instruction class, and is called CLA. This byte can take any 8-bit value except 0xFF. At present, however, only the values of the most significant nibble of 0 and 8 are used. The most significant nibble with a value of 0 is defined as an interindustry command, and the value of 8 is proprietary to the EMV specification.

The second byte of the Command APDU is the instruction code, and is called INS. This byte is valid only if the least significant bit is 0 and the most significant nibble is neither a 6 nor a 9.

The P1 and P2 bytes of the mandatory header contain parameters for the specific command, and can be any value. If not used, the parameter byte must have the value of 0x00.

R-APDU
After receiving and interpreting the APDU command from the terminal, the smart card will return a response. As defined in the specifications, this response consists of an optional body of variable length followed by a required trailer consisting of two bytes. This format is illustrated in **Table 6**, and the contents of the APDU response are described in **Table 7**.

Table 6. APDU Card Response Format

Data	SW1	SW2
←Body→	←Trailer→	

Table 7. APDU Command Response Contents

Code	Description	Length
Data	String of data bytes received in APDU response	Var (= Lr)
SW1	Command processing status	1
SW2	Command processing qualifier	1

The expected length of the card's response is transmitted as the Le code portion of the APDU command. The actual length of the response is called Lr. Although the card does not transmit the value of Lr, the terminal can calculate it if needed for the application.

For normal completion of a command, a smart card will return SW1 with a value of 0x90 and SW2 with a value of 0x00. Any other response indicates that either an error or warning occurred[2].

Example Code

The software provided with this application note is contained in the downloadable file an4029_sw.zip. This file contains all of the C (main.c, ds8007.c, LCD_Funct.c) and assembly language (Startup.a51) source code required to produce the executable hex file (ds8007.hex). The code was compiled and linked using the Keil PK51 Professional Developer's Kit and the µVision® Integrated Development Environment (IDE). The µVision project file (ds8007.Uv2) is also included in the .zip file. The .HEX file was loaded and run on the DS8007 Smart Card Interface board in the DS8007 evaluation (EV) kit, available from Maxim. The example software implements a complete smart card session including power-up, ATR, APDU, and power-down operations. When connected to a dumb terminal, the board and software will produce an RS-232 serial output at 38,400 baud. This output is shown in **Figure 4** below. Detailed descriptions of this software are beyond the scope of this application note, but the source code provided can be the basis of a complete smart card interface using the DS5002 Secure Microprocessor and the DS8007 Multiprotocol Dual Smart Card Interface chip.

```
DS8007 Basic Test

DS8007 Version: 3
Compiled Crystal Frequency: 12000000
Waiting for smartcard presence
Found smartcard in slot 1
Power up return: 0
ATR: 3b be 11 00 00 41 01 38 25 00 03 00 00 00 00
00 01 90 00

APDU Response: 10
CB C4 BD D5 A4 7E 36 3F 90 0


Waiting for smartcard removal from slot 1
Smartcard removed
Waiting for smartcard presence
```

Figure 4. Software output.

For testing the example software, a microcontroller-based smart card from Advanced Card Systems (ACS) was used. As a microcontroller-based device, this card executes functions of its embedded operating system called ACS Smart Card Operating Systems Version 1, or ACOS1. This card has the following features.

- 8kB of EEPROM memory for application data
- Compliance with ISO 7816-3, T=0 protocol
- DES and MAC capabilities
- Session key based on random numbers
- PIN, changeable by card holder
- Key pair for mutual authentication

The ACOS1 documentation indicates that this card will respond to reset with 19 bytes of data. As can be seen in Figure 4, the value returned for T0 is 0xBE. The high nibble (0xB) indicates that TA1, TB1, and TD1 will be included in the ATR along with TS. The low nibble contains 0xE, indicating that there will be 14 historical bytes included in the ATR response (see character T0 description above). Therefore, a total of 19 bytes were included in the ATR from the card.

For this example software, one ADPU, the Start Session command is executed. This command has the following format.

CLA	INS	P1	P2	P3
0x80	0x84	0x00	0x00	0x08

The response to the Start Session command has the following format.

Data	SW1	SW2
RNDc (8 bytes from card)	Status	Status

The software's output shown in Figure 4 indicates that the random number returned from the card was 0xCB, 0xC4, 0xBD, 0xD5, 0xA4, 0x7E, 0x36, and 0x3F. It also shows the status returned was 0x90, 0x00, which indicates a successful completion of the command.

Conclusion

The DS8007 is an mixed-signal peripheral that relieves the burden of interfacing a microcontroller with smart cards. It provides all the electrical signals necessary to physically interface with two separate smart cards. A dedicated internal sequencer controls automatic card activation and deactivation, as well as an ISO UART for data communication. Charge pumps and voltage regulators allow the DS8007 to operate from a 2.7V to 6.0V supply voltage, while simultaneously producing two independent smart card supply voltages either of which can be 1.8V, 3.0V, or 5V. Communication with the microcontroller is provided by a standard, parallel 8-bit bus that carries data in a nonmultiplexed configuration or data and address in multiplexed configuration. The software provided implements a complete card session using the DS5002 Secure Microprocessor and the DS8007 as its smart card interface. The characters returned by the smart card in its ATR are output on the board's serial port at 38400 baud, and the card is sent a "Start Session" Command APDU. The resulting random number is also output on the serial port.

[1]Source: Gartner Group.
[2]A complete description of Command APDU response error or warning status codes can be found in the Book-1 Part II Section 6 of the EMV specifications.

A Chinese version of this article was published in the September 2007 edition of *EDN China*.

Related Parts		
DS5002	Secure Microprocessor Chip	Free Samples
DS5002FP	Secure Microprocessor Chip	Free Samples
DS5250	High-Speed Secure Microcontroller	
DS8007	Multiprotocol Dual Smart Card Interface	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 4029: http://www.maximintegrated.com/an4029
APPLICATION NOTE 4029, AN4029, AN 4029, APP4029, Appnote4029, Appnote 4029