

# Smart Card Interface Design Example

## Table of Contents

Overview . . . . .	1
Design Description . . . . .	1
Interface Description . . . . .	6
Utilization Details . . . . .	6
Hardware Testing Scheme . . . . .	7
Software Interface and Design Details . . . . .	7
Timing Diagrams . . . . .	17
Application Area . . . . .	21
Conclusion . . . . .	21
List of Changes . . . . .	22

## Overview

Smart cards are credit card-sized plastic cards with embedded memory, widely used for applications such as access control, authentication, and security key storage. This document provides design details and usage for a smart card reader design example. This design is based on Electronic Signals and Transmission Protocols, SO 7816-3, which defines voltage and current requirements for the electrical contacts.

Associated files for this design example can be downloaded from the Microsemi website:

[http://soc.microsemi.com/download/rsc/?f=SmartCard\\_Reader\\_Interface\\_DF](http://soc.microsemi.com/download/rsc/?f=SmartCard_Reader_Interface_DF).

## Design Description

The design example is targeted to the Atmel® 16 Kbit AT88SC1616C Crypto Memory® smart card module. This design generates the necessary control signals at the smart card side. The commands/data for accessing the smart card module must be fed in either by a microcontroller interface or the configuration logic of the FPGA. In this design, Microsemi's embedded microcontroller, Core8051, is used for generating the commands to the smart card reader interface. The IP interface block diagram is shown in Figure 1.

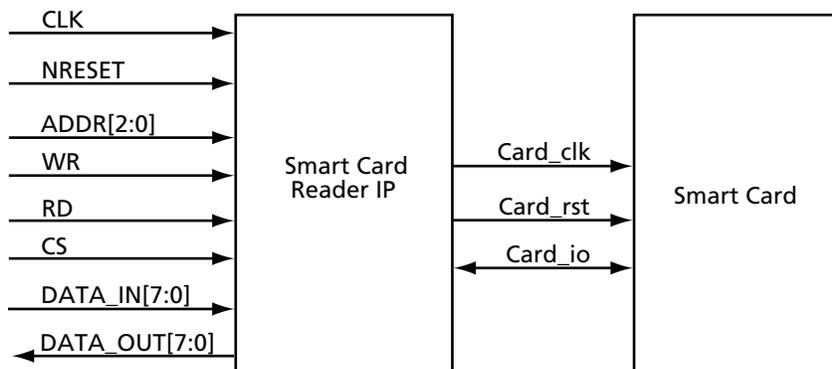


Figure 1 • IP Interface Diagram

The AT88SC1616C smart card module consists of two memory areas: the user zone and the configuration memory area.

The user zone is divided into 16 zones of 1,024 bits each. Access to the user zones is allowed only after security requirements have been met. These security requirements are defined by the user during the personalization of the device in the configuration memory.

The configuration memory consists of 2,048 bits of EEPROM memory used for storing passwords, keys, and codes, as well as for defining security levels to be used for each user zone. Access rights to the configuration memory are defined in the control logic and may not be altered by the end user.

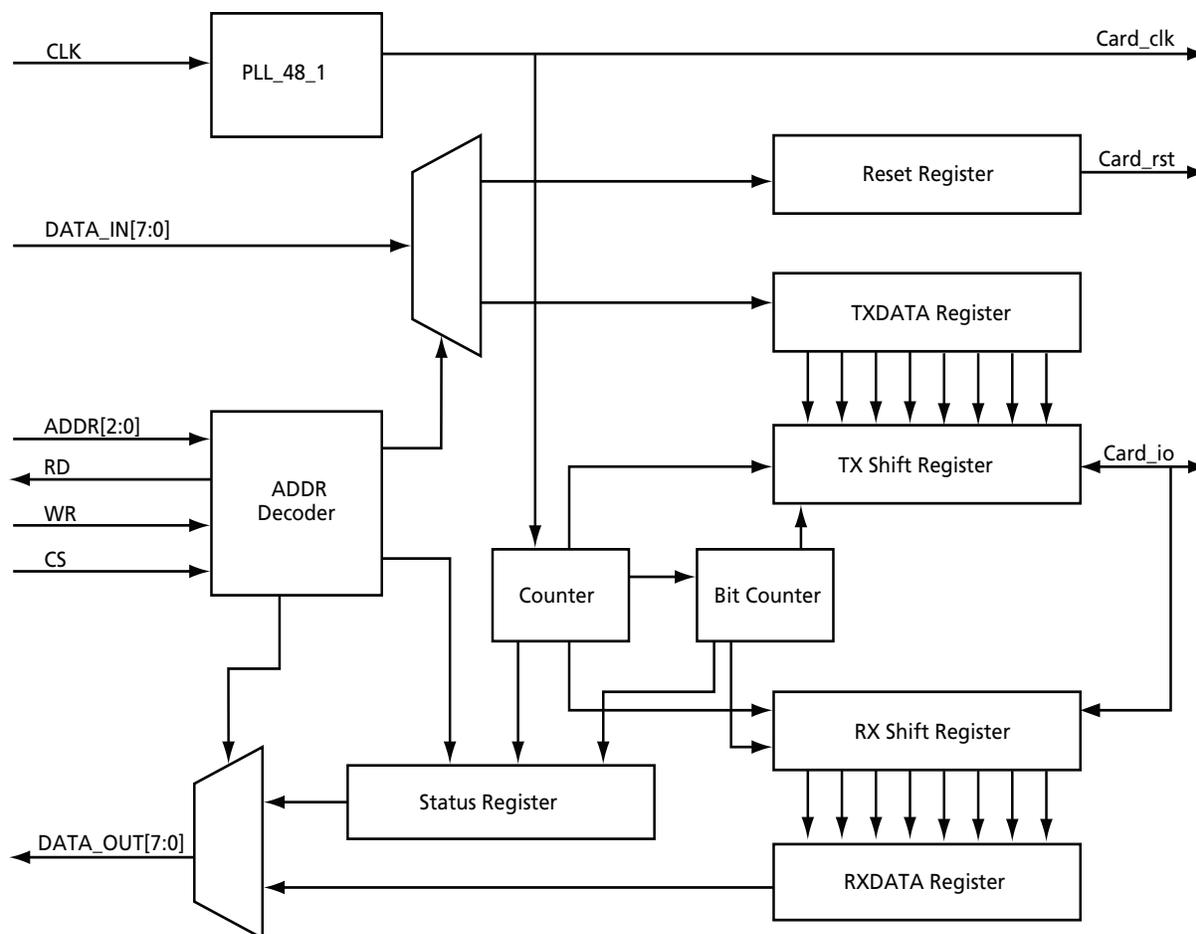
The key for accessing the supplied smart card module's configuration memory is 0xB6A405. This key is required for unlocking the configuration memory for any modification.

This document explains how to access the user zones and the configuration memory area.

As defined in the ISO 7816-3 standard, T=0 protocol, the operating procedure consists of the following consecutive operations:

1. Connection and activation of the contacts by the interface device
2. Reset of the card
3. Answer to reset by the card
4. Subsequent information exchange between the card and the interface device
5. Deactivation of the contacts by the interface device

The internal block diagram of the smart card reader IP is shown in [Figure 2](#).



**Figure 2 • Internal Block Diagram**

There are four internal registers for the design: Reset Register, Transmit Data Register, Receive Data Register, and Status Register.

- The Reset Register resets the smart card.
- The Transmit Register transmits data to the CARD\_IO bus.
- The Receive Register receives data from the CARD\_IO bus.
- The Status Register indicates the status of data transaction.

There are two counters: Counter and Bit Counter. Counter is used to sample each bit at the rate of  $f$  ( $f = 1$  MHz), at the middle of the bit. Bit Counter is used to count for 11 bits of a packet.

The offset addresses for the smart card reader registers are described in [Table 1](#).

**Table 1 • Smart Card Reader Registers**

Address ADDR[2:0]	Type	Width	Reset Value	Name
0x2	Write	8	0	Transmit Data Register
0x3	Read	8	0	Receive Data Register
0x4	Read	8	0	Status Register
0x5	Write	8	0	Reset Register

The Transmit Data Register contains the data that must be transmitted to the smart card. The Transmit Data Register bit settings are shown in [Table 2](#).

**Table 2 • Transmit Data Register Bit Position**

7	6	5	4	3	2	1	0
TX DATA[7]	TX DATA[6]	TX DATA[5]	TX DATA[4]	TX DATA[3]	TX DATA[2]	TX DATA[1]	TX DATA[0]

The Receive Data register contains the data that is received from the smart card. The Receive Data Register bit settings are shown in [Table 3](#).

**Table 3 • Receive Data Register Bit Position**

7	6	5	4	3	2	1	0
RX DATA[7]	RX DATA[6]	RX DATA[5]	RX DATA[4]	RX DATA[3]	RX DATA[2]	RX DATA[1]	RX DATA[0]

The Status Register bit settings are shown in [Table 4](#).

**Table 4 • Status Register Bit Position**

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TxRx_DONE

By default, the content of the Status Register is 0x00. When a byte of data is transferred out of the Transmit register and shifted out serially through the card I/O pin to the smart card, the content of the Status Register changes to 0x01. This bit must be monitored by the host to confirm that the data has been completely shifted out to the smart card. Once the complete byte is shifted out of the Transmit Register, this bit is set to 0x01. The content must be cleared to 0x00 before the next data transfer.

Similarly for the read operation, this register bit needs to be monitored to check whether any data is available in the Receive Register. Once the data is read from the smart card and a byte of information is available in the Receive Register, this bit becomes 1 and must be cleared by the host before receiving the next byte.

The Reset Register bit settings are shown in [Table 5](#).

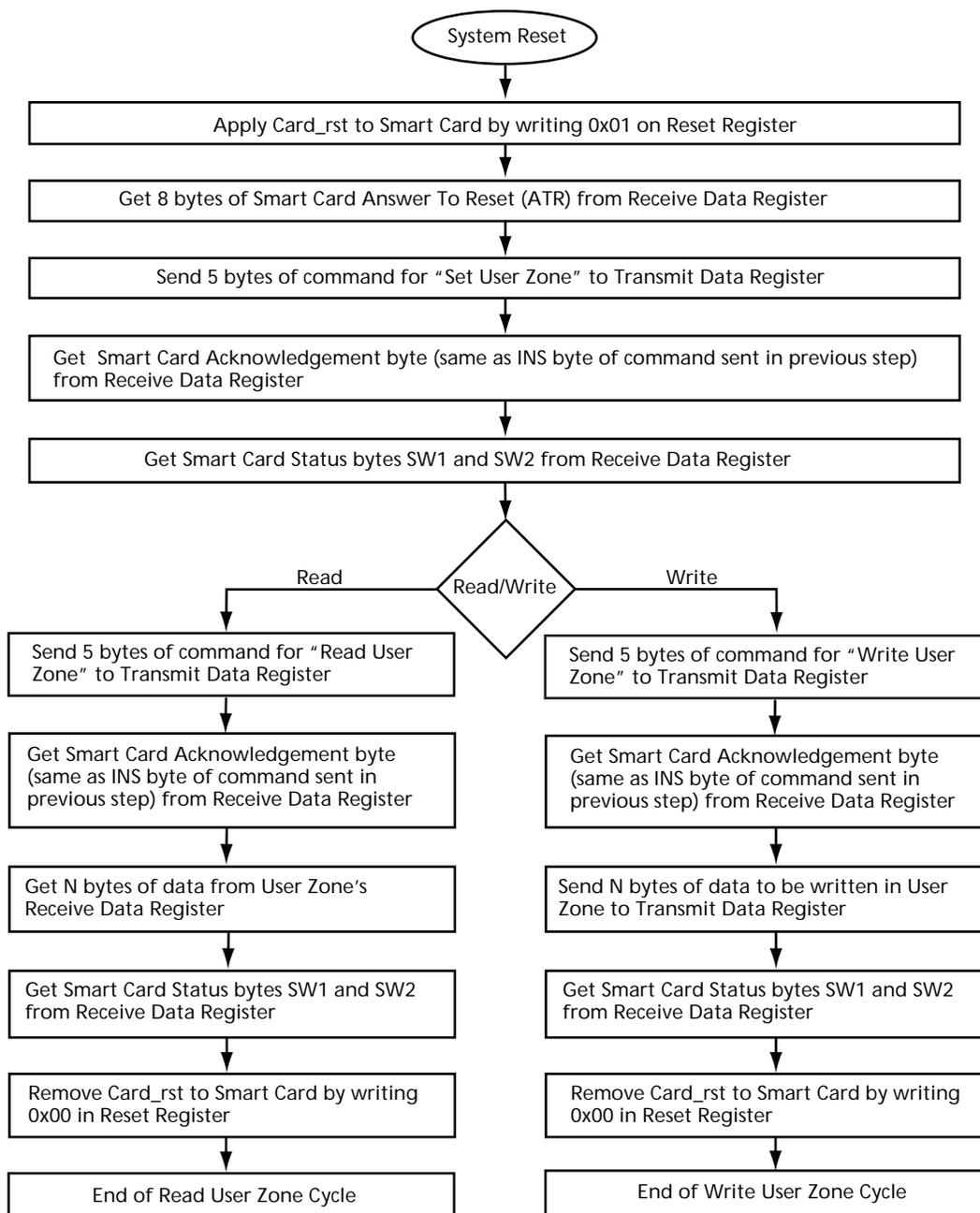
**Table 5 • Reset Register Bit Position**

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CARD_RST

By default, the Reset Register content will be 0x00. For any transaction to be performed, the initial step is to reset the card. For normal operation, this bit must be set to 1.

For more information on ATR, INS, two status words (SW1 and SW2), and command sets for reading and writing to the AT88SC1616C smart card module, refer to [www.atmel.com/dyn/resources/prod\\_documents/2030s.pdf](http://www.atmel.com/dyn/resources/prod_documents/2030s.pdf) and [www.atmel.com/dyn/resources/prod\\_documents/doc5025.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc5025.pdf).

Figure 3 shows the read/write cycle user zone flow.



**Figure 3 • Read/Write User Zone**

Figure 4 shows the read/write configuration memory flow.

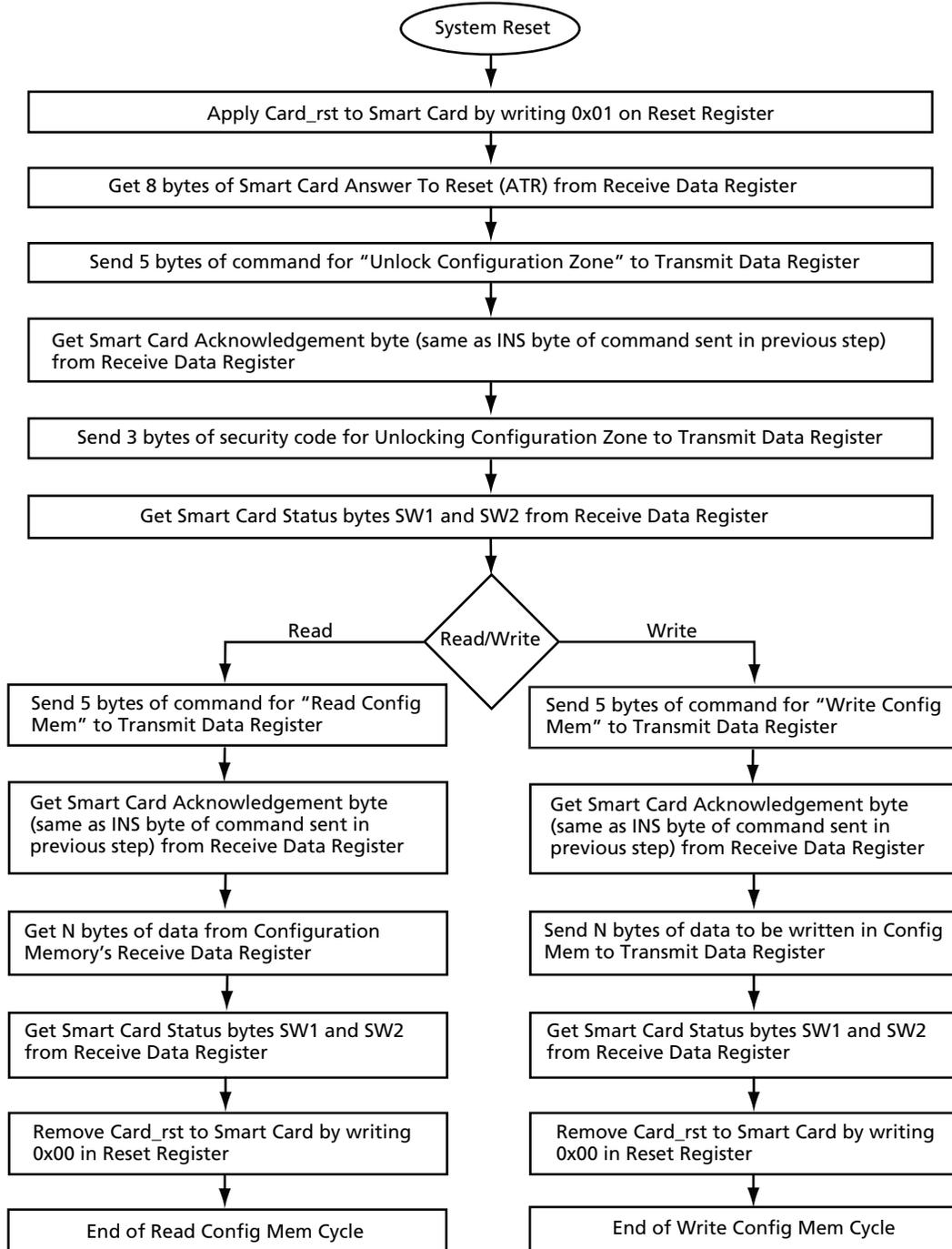


Figure 4 • Read/ Write Configuration Memory

## Interface Description

Table 6 described the interface details for the smart card reader interface.

**Table 6 • Ports Description**

Port	Direction	Description
NRESET	I	Active Low reset signal aborts current operation and resets all the control signals and registers.
CLK	I	Input Clock of 48 MHz
CARD_IO	I/O	Bidirectional serial data line to smart card
CARD_RST	O	Reset to smart card
CARD_CLK	O	Output clock to the smart card of 1 MHz
ADDR [2:0]	I	Address control signal to access various registers
DATA_IN [7:0]	I	Data input to smart card reader
DATA_OUT [7:0]	O	Data output from smart card reader
WR	I	Write enable signal
RD	I	Read enable signal
CS	I	Enable signal for smart card reader interface

## Utilization Details

This design can be implemented in Microsemi AGL250 or A3P250 devices. However, for testing purposes, this design was verified using Microsemi's AGL600V2-484 FBGA IGLOO<sup>®</sup> device. Table 7 shows the utilization details for this design example, including the smart card reader interface, Core8051, and other glue logic.

**Table 7 • Utilization Details**

Resource	Utilized	Total	Percentage
Core	5,721	13,824	41.38%
I/O (with clocks)	7	235	2.98%
Differential I/O	0	60	0.00%
Global (chip + quadrant)	4	18	22.22%
PLL	1	1	100.00%
RAM/FIFO	17	24	70.83%
Low static I <sub>CC</sub>	0	1	0.00%
FlashROM	0	1	0.00%
User JTAG	1	1	100.00%

## Hardware Testing Scheme

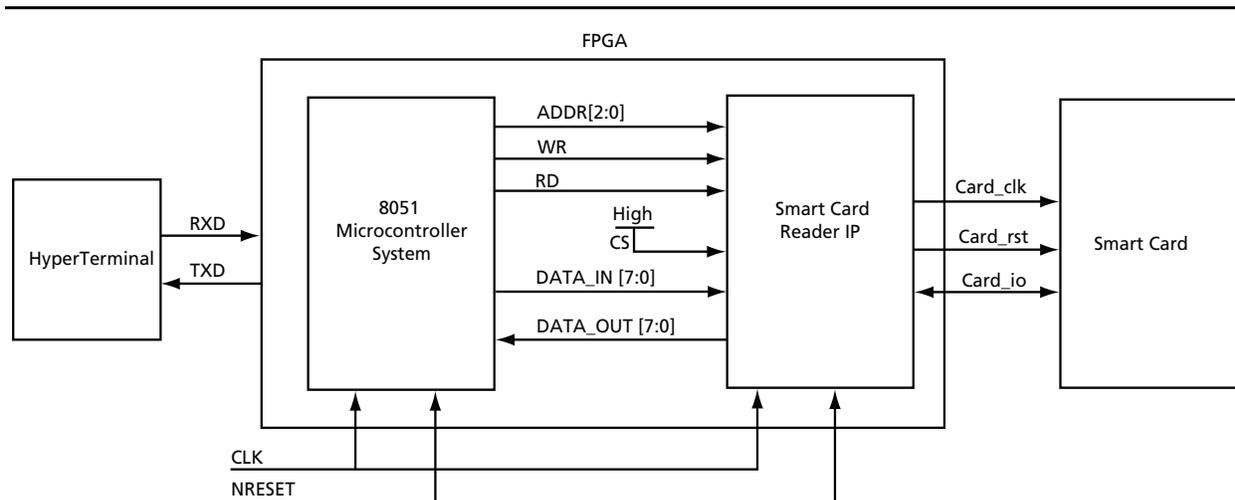
Testing of the logic is performed with the help of the Core8051 embedded processor on Microsemi's IGLOO Development Kit, along with a customized daughter board. Hardware verification was completed using the FS2 debugger and software executable. Figure 8 details the components used on the daughter board.

**Table 8 • Smart Card Connector Details**

Description	Manufacturer	Part Number
Smart card connector, 8-pin PCB mount	FCI	7434L0825S01LF

The smart card interface supports the international standard ISO 7816. The smart card supplied along with the daughter board is AT88SC1616C from Atmel.

Figure 5 shows the test setup, which consists of two sub units: the 8051 microcontroller system and the smart card reader.



**Figure 5 • Test Setup**

The Core8051 microcontroller system oversees the exchange of commands, acknowledgements, status, and data bytes with the smart card through the smart card reader interface. The design generates the timing for all signals interfacing with the smart card.

The PLL block within the FPGA generates the multiple clock frequencies required for proper access of the smart card. Normally, a smart card can operate at a maximum frequency of 5 MHz, but the PLL block can be reconfigured with a different frequency to support other cards. For this design, the transaction of each byte is done serially with a bit rate of  $372/f$  ( $f = 1$  MHz operating frequency), as specified in Electronic Signals and Transmission Protocols, ISO 7816-3.

## Software Interface and Design Details

The software modules consist of two applications to test and verify the smart card IP core:

- Application software: The software is written in C language for a Windows® platform and must run from a PC connected to the Microsemi IGLOO development board via a USB port. The program is used to send data to the smart card, read back the data from the smart card, and verify the data is correct. Both the configuration and user zones can be tested.
- Firmware software: The software is written in C language and can be used to write or read data from the smart card. Both user and configuration zone memories can be written and read back. The offset address corresponding to the different registers is hardcoded.

## Firmware Files

### **Main.c**

This file contains source code for the smart card drivers that handles the read user zone and write user zone commands. The source code also contains a UART interface for the user interface.

The file contains the following modules:

#### **Read\_UserZone**

This module contains the source code for reading data from the user zone of the smart card. The driver reads all 16 bytes of all 16 zones in the following sequence:

1. Resets the register, which resets the smart card, clock, and IP.
2. Verifies the answer to reset the 8-byte signal.
3. Writes the command to set the user zone.
4. Read acknowledgement.
5. Writes the user zone command.
6. Read acknowledgement.
7. Reads data from the user zone.
8. Read acknowledgement.

#### **Write\_UserZone**

This module contains the source code for writing data into the user zone of the smart card. The driver writes data into all the 16 bytes of all 16 zones in the following sequence:

1. Resets the register, which resets the smart card, clock, and IP.
2. Verifies the answer to reset the 8-byte signal.
3. Writes the command to set the user zone.
4. Read acknowledgement.
5. Writes the user zone command.
6. Read acknowledgement.
7. Writes data into the user zone.
8. Read acknowledgement.
9. Resets the Reset register.

#### **Read\_ConfZone**

This module contains the source code for reading data from the configuration zone of the smart card. The driver reads all the 240 bytes of configuration zone in the following sequence:

1. Resets the register, which resets the smart card, clock, and IP.
2. Verifies the answer to reset the 8-byte signal.
3. Writes the command to unlock the configuration memory.
4. Read acknowledgement.
5. Writes the command to read the configuration memory.
6. Read acknowledgement.
7. Reads data from the smart card.
8. Read acknowledgement.

## Write\_ConfZone

This module contains the source code for writing data into the configuration zone of the smart card. The driver writes manufacturer code, identification number, issuer code, and register code into the configuration memory zone in the following sequence:

1. Resets the register, which resets the smart card, clock, and IP.
2. Verifies the answer to reset the 8-byte signal.
3. Writes the command to unlock the configuration memory.
4. Read acknowledgement.
5. Writes the smart card manufacturing data.
6. Read acknowledgement.
7. Writes the card identification number.
8. Read acknowledgement.
9. Writes the card issuer code.
10. Read acknowledgement.
11. Writes the register code.
12. Read acknowledgement.
13. Resets the Reset register.

## Header.h

This is the smart card driver header file. [Table 9](#) describes the register mapping specified in this file.

**Table 9 • Register Mapping**

Address	Register Name	R/W	Description
0x0002	Base Address	R/W	Base address of IP code
0x0004	GPIO status	R	Reads the status register

## Application Files

### Main.c

This file provides the main functionality of the program. The user interaction, data validation, and communication with the USB port are done inside this file. The user input is validated and sent to the USB port sequentially.

### UsbCom.c

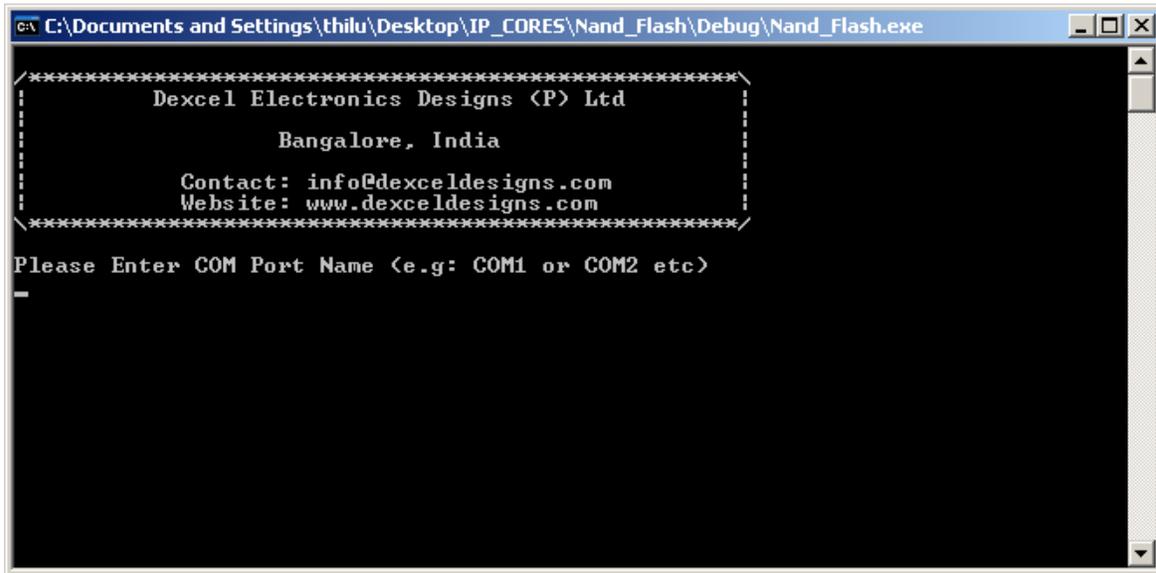
This source file takes care of the USB communication part.

### Program Execution (Smart\_Card.exe)

This executable runs from a Windows environment. The following menu options are available:

- Communication port: Select the USB port connected to the board.
- Operation: Select the operation (read, write, or verify).

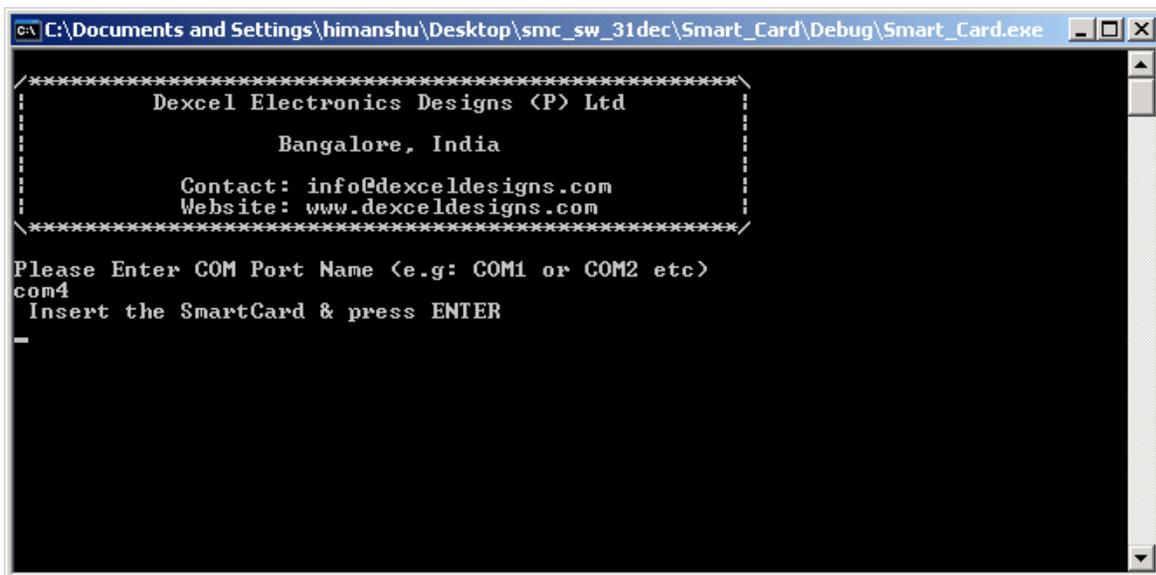
When Smart\_Card.exe is executed, the screen shown in Figure 6 appears.



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Nand_Flash\Debug\Nand_Flash.exe
/*****\
  Dexcel Electronics Designs (P) Ltd
      Bangalore, India
  Contact: info@dexceldesigns.com
  Website: www.dexceldesigns.com
/*****\
Please Enter COM Port Name (e.g: COM1 or COM2 etc)
_
```

**Figure 6 • Identifying the COM Port**

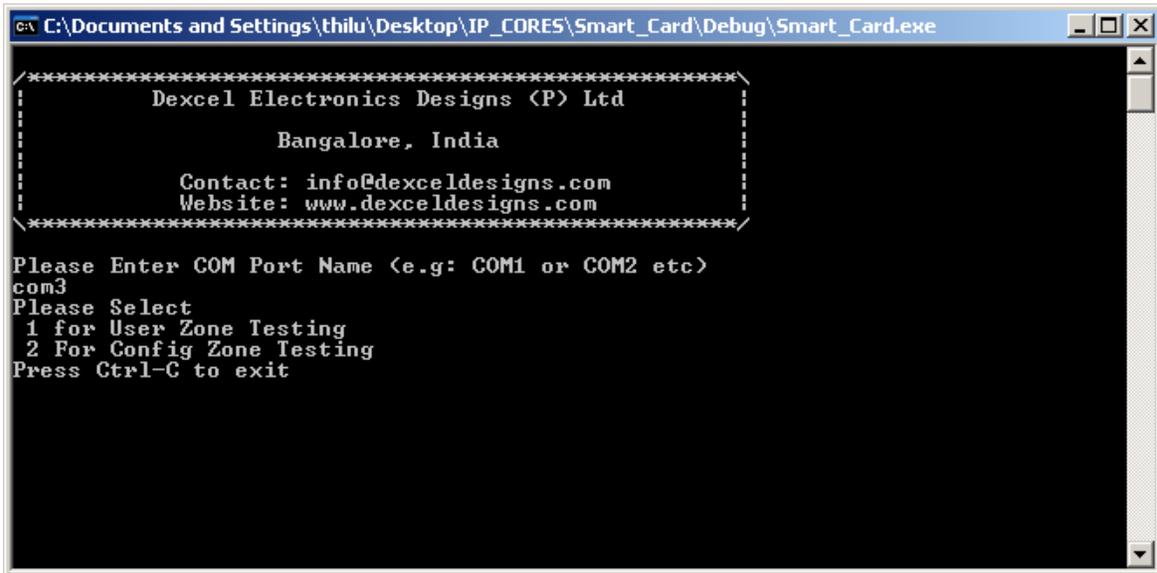
Enter the COM port name (available from the Device Manager). The screen shown in Figure 7 appears and requests you to insert the smart card.



```
C:\Documents and Settings\himanshu\Desktop\smc_sw_31dec\Smart_Card\Debug\Smart_Card.exe
/*****\
  Dexcel Electronics Designs (P) Ltd
      Bangalore, India
  Contact: info@dexceldesigns.com
  Website: www.dexceldesigns.com
/*****\
Please Enter COM Port Name (e.g: COM1 or COM2 etc)
com4
Insert the SmartCard & press ENTER
_
```

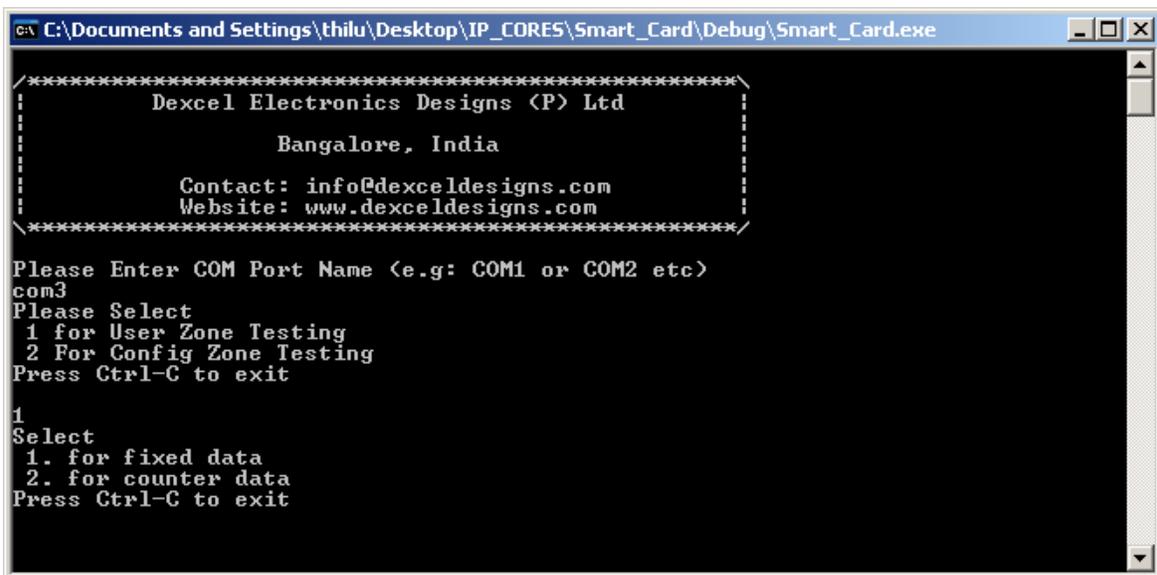
**Figure 7 • Prompting the User for Inserting the Smart Card**

You must have a smart card reader connected to the FPGA to perform the remaining steps. Insert the smart card with the correct orientation into the smart card reader and press **Enter**. The screen shown in [Figure 8](#) appears.



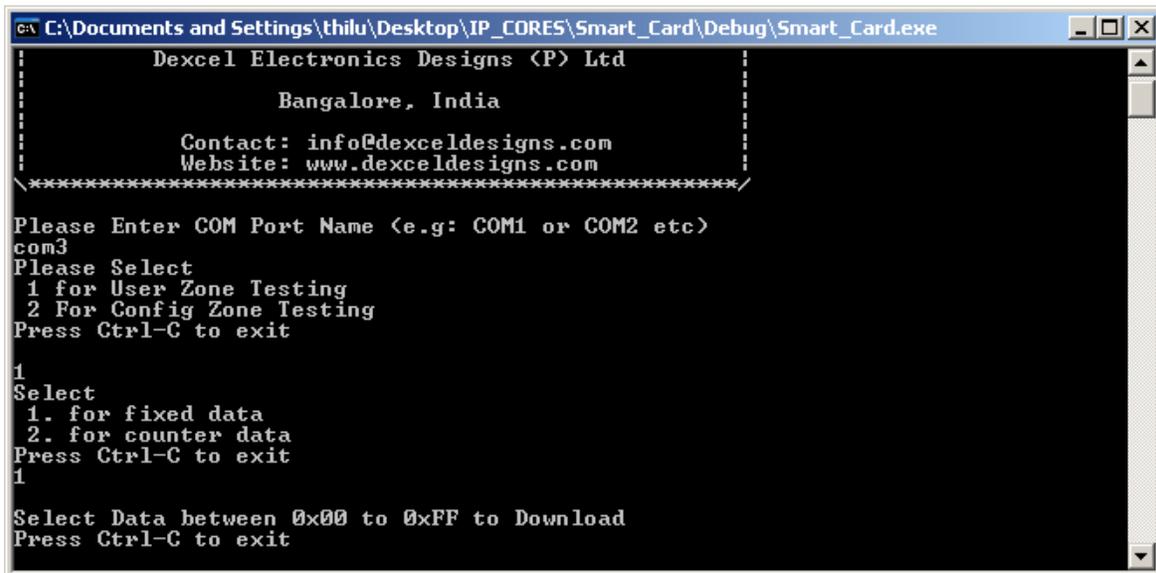
**Figure 8 • Selecting the Zone**

Select 1 for user zone, (2 for configuration zone testing). The screen shown in [Figure 6](#) appears.



**Figure 9 • Selecting Data Type**

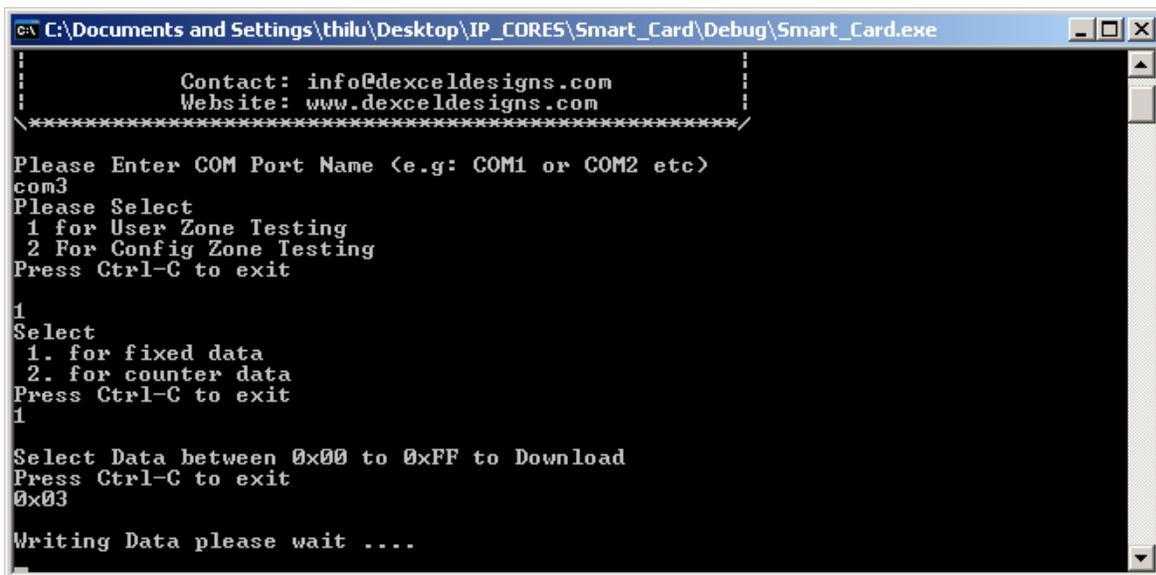
Select the data type. If 2 is selected, a counter value of 00 to 0xff will be written into the smart card user zone. If option 1 for fixed data is selected, the screen shown in [Figure 10](#) appears.



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
Dexcel Electronics Designs <P> Ltd
Bangalore, India
Contact: info@dexceldesigns.com
Website: www.dexceldesigns.com
*****/
Please Enter COM Port Name (e.g: COM1 or COM2 etc)
com3
Please Select
1 for User Zone Testing
2 For Config Zone Testing
Press Ctrl-C to exit
1
Select
1. for fixed data
2. for counter data
Press Ctrl-C to exit
1
Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
```

**Figure 10 • Selecting Data**

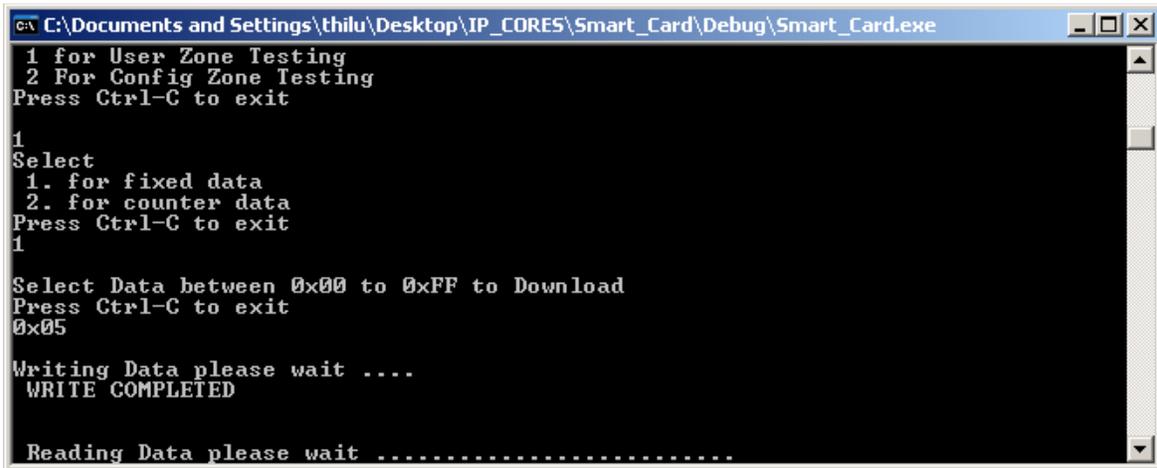
Select data between 00 to 0xff. The screen shown in [Figure 11](#) appears.



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
Contact: info@dexceldesigns.com
Website: www.dexceldesigns.com
*****/
Please Enter COM Port Name (e.g: COM1 or COM2 etc)
com3
Please Select
1 for User Zone Testing
2 For Config Zone Testing
Press Ctrl-C to exit
1
Select
1. for fixed data
2. for counter data
Press Ctrl-C to exit
1
Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
0x03
Writing Data please wait ....
```

**Figure 11 • User Zone Write Operation**

After completing the write operation, a window appears, indicating that writing is complete (Figure 12).



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
1 For User Zone Testing
2 For Config Zone Testing
Press Ctrl-C to exit

1
Select
 1. for fixed data
 2. for counter data
Press Ctrl-C to exit
1

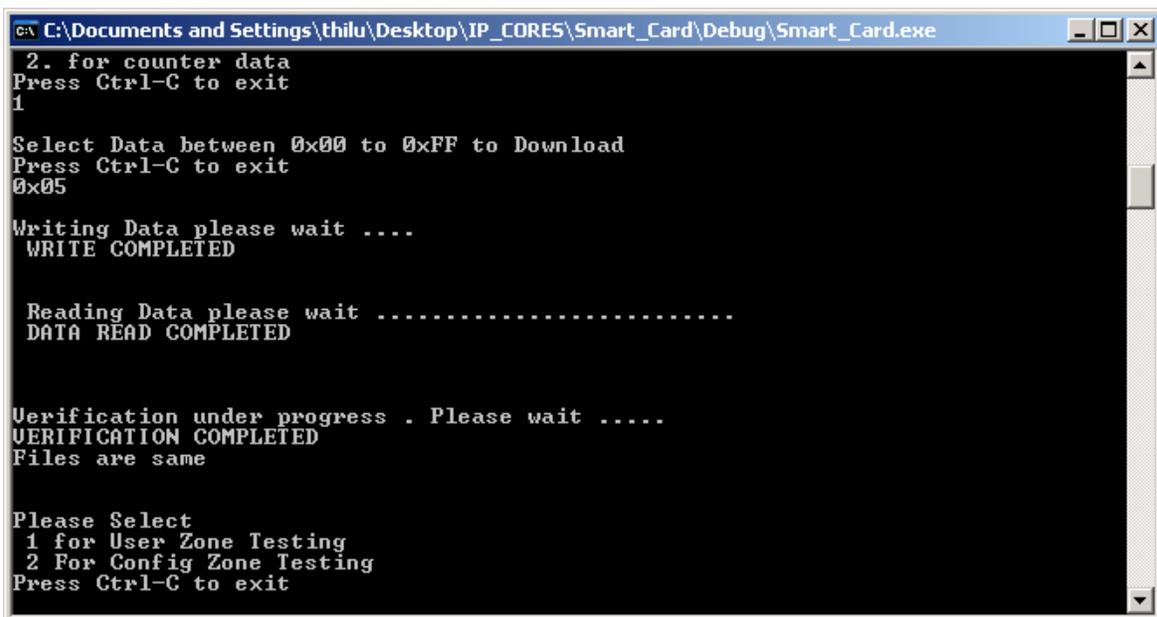
Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
0x05

Writing Data please wait ....
WRITE COMPLETED

Reading Data please wait .....
```

**Figure 12 • User Zone Read Operation**

After completing the read and verification operations, the screen shown in Figure 13 appears, indicating that the files are the same.



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
2. for counter data
Press Ctrl-C to exit
1

Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
0x05

Writing Data please wait ....
WRITE COMPLETED

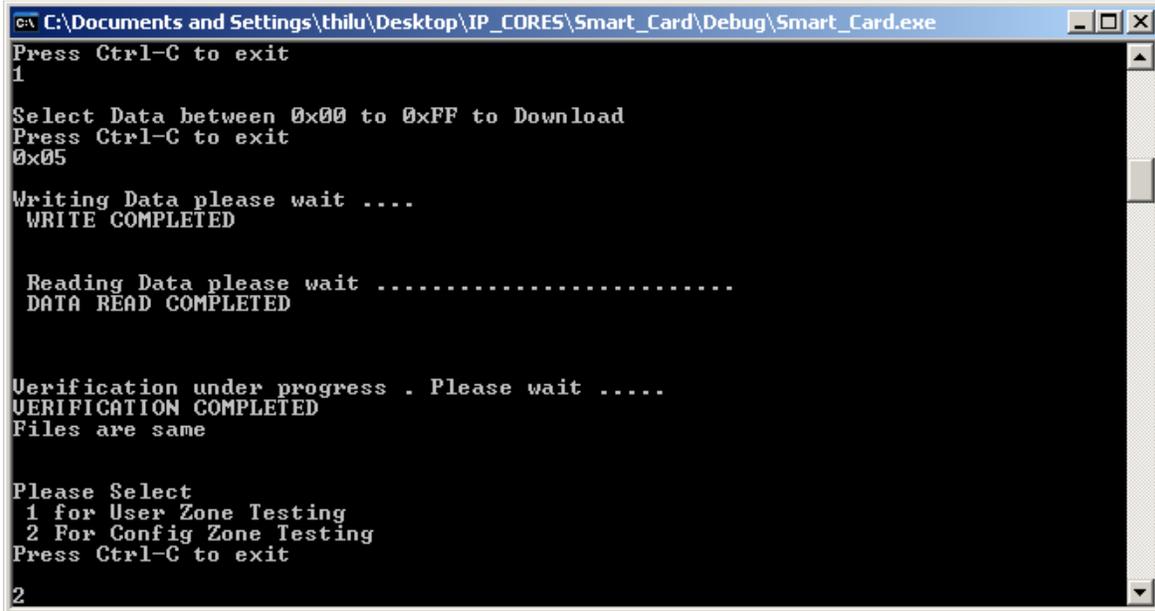
Reading Data please wait .....
DATA READ COMPLETED

Verification under progress . Please wait .....
VERIFICATION COMPLETED
Files are same

Please Select
 1 for User Zone Testing
 2 For Config Zone Testing
Press Ctrl-C to exit
```

**Figure 13 • Verifying the User Zone**

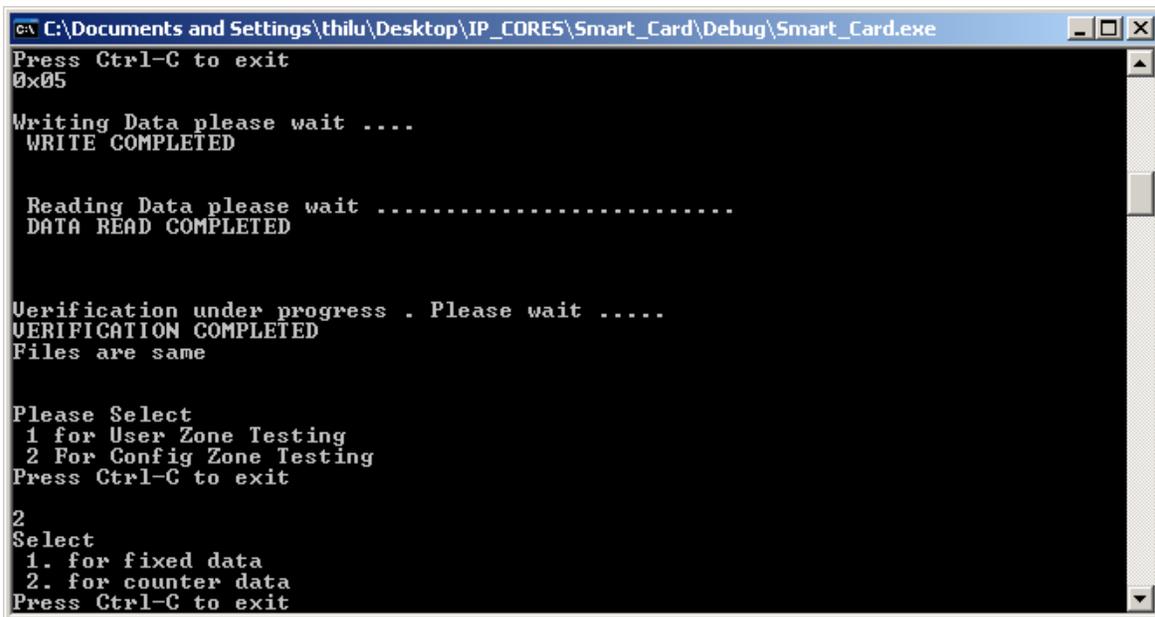
Select 2 to test the configuration zone (Figure 14).



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
Press Ctrl-C to exit
1
Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
0x05
Writing Data please wait ....
WRITE COMPLETED
Reading Data please wait .....
DATA READ COMPLETED
Verification under progress . Please wait .....
VERIFICATION COMPLETED
Files are same
Please Select
 1 for User Zone Testing
 2 For Config Zone Testing
Press Ctrl-C to exit
2
```

Figure 14 • Selecting Testing Type

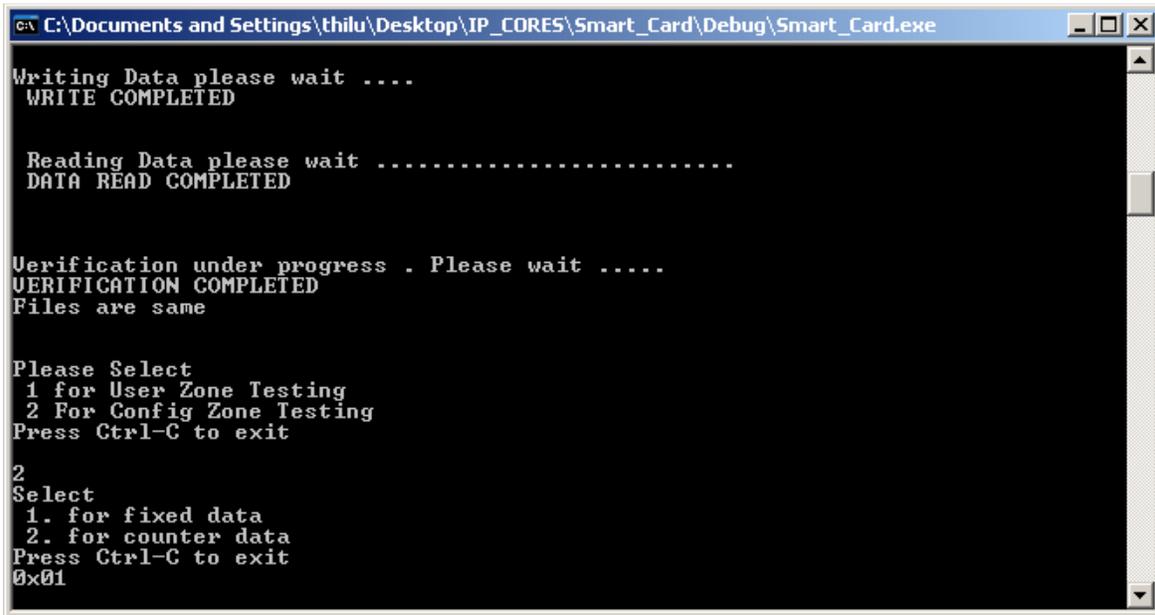
The screen shown in Figure 15 appears, requesting a data type selection. If option 2 is selected, a counter data value from 0x00 and sequential incrementing of the counter value is written.



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
Press Ctrl-C to exit
0x05
Writing Data please wait ....
WRITE COMPLETED
Reading Data please wait .....
DATA READ COMPLETED
Verification under progress . Please wait .....
VERIFICATION COMPLETED
Files are same
Please Select
 1 for User Zone Testing
 2 For Config Zone Testing
Press Ctrl-C to exit
2
Select
 1. for fixed data
 2. for counter data
Press Ctrl-C to exit
```

Figure 15 • Selecting Data Type for the Configuration Zone

If option 1 is selected, the screen shown in Figure 16 appears. Select a data value between 0x00 to 0xFF



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
Writing Data please wait ....
WRITE COMPLETED

Reading Data please wait .....
DATA READ COMPLETED

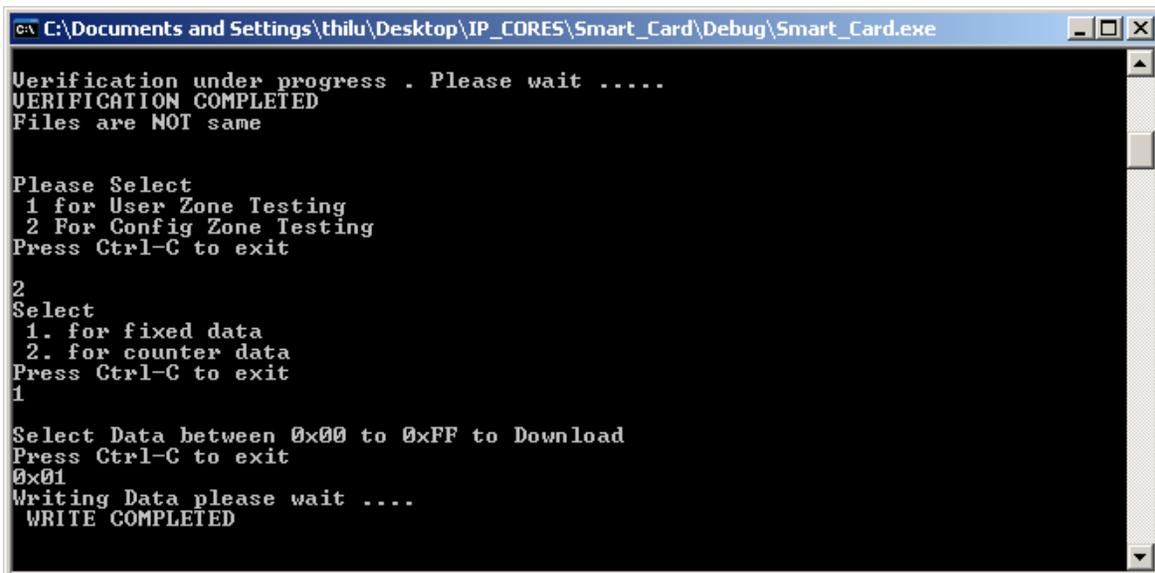
Verification under progress . Please wait .....
VERIFICATION COMPLETED
Files are same

Please Select
1 for User Zone Testing
2 For Config Zone Testing
Press Ctrl-C to exit

2
Select
1. for fixed data
2. for counter data
Press Ctrl-C to exit
0x01
```

**Figure 16 • Data for Configuration Zone**

The specified data value is written. The message shown in Figure 17 is displayed: “WRITE COMPLETED.”



```
C:\Documents and Settings\thilu\Desktop\IP_CORES\Smart_Card\Debug\Smart_Card.exe
Verification under progress . Please wait .....
VERIFICATION COMPLETED
Files are NOT same

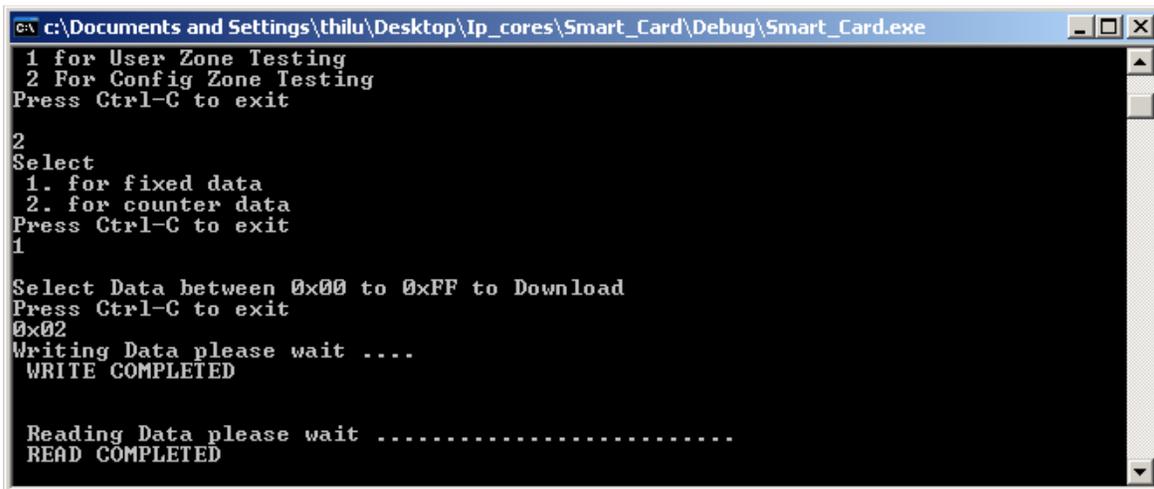
Please Select
1 for User Zone Testing
2 For Config Zone Testing
Press Ctrl-C to exit

2
Select
1. for fixed data
2. for counter data
Press Ctrl-C to exit
1

Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
0x01
Writing Data please wait ....
WRITE COMPLETED
```

**Figure 17 • Configuration Zone Write Status Message**

After the write operation is completed, a read operation is performed on the Configuration zone memory (Figure 18).



```
c:\Documents and Settings\thilu\Desktop\Ip_cores\Smart_Card\Debug\Smart_Card.exe
1 for User Zone Testing
2 For Config Zone Testing
Press Ctrl-C to exit

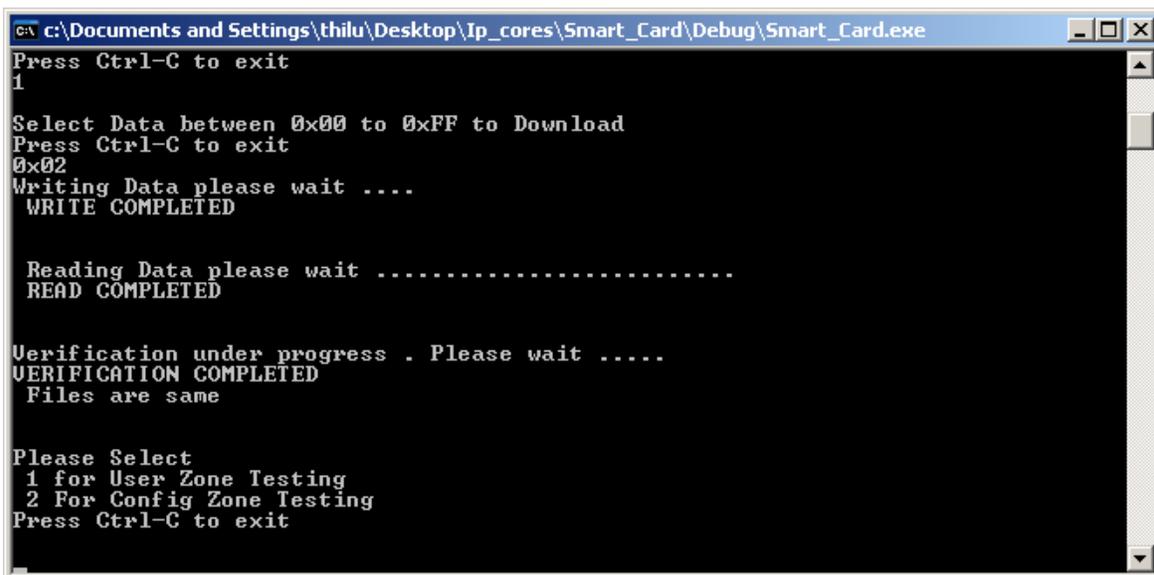
2
Select
 1. for fixed data
 2. for counter data
Press Ctrl-C to exit
1

Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
0x02
Writing Data please wait ....
WRITE COMPLETED

Reading Data please wait .....
READ COMPLETED
```

**Figure 18 • Configuration Zone Read Status Message**

After the read operation, file verification is performed. After verification, the result is displayed. In Figure 19, the result is shown as, “Files are same.”



```
c:\Documents and Settings\thilu\Desktop\Ip_cores\Smart_Card\Debug\Smart_Card.exe
Press Ctrl-C to exit
1

Select Data between 0x00 to 0xFF to Download
Press Ctrl-C to exit
0x02
Writing Data please wait ....
WRITE COMPLETED

Reading Data please wait .....
READ COMPLETED

Verification under progress . Please wait .....
VERIFICATION COMPLETED
Files are same

Please Select
 1 for User Zone Testing
 2 For Config Zone Testing
Press Ctrl-C to exit
```

**Figure 19 • Configuration Zone Result**

After the verification results, the menu screen appears.

During the write operation for the user zone, a text file named Write\_Data\_user.txt is created in the path from which Smart\_Card.exe was executed. Similarly, during the read operation, a file name Read\_Data\_user.txt is created. During verification, these two files are compared and the result is displayed.

Similarly, a text file name Write\_Data\_conf.txt is created during the write operation for configuration zone testing, and Read\_Data\_conf.txt is created during the read operation. During verification, these two files are compared and the result is displayed.

**Note:** As per the smart card datasheet, only a few locations of configuration memory can be written. Other locations are read only or secure areas for passwords and other access control. The test program is performing write access only to the Card Manufacturer Code (4 bytes, starting from location 0x0B).

For more details on memory organization, refer to the documents available on the Atmel website:

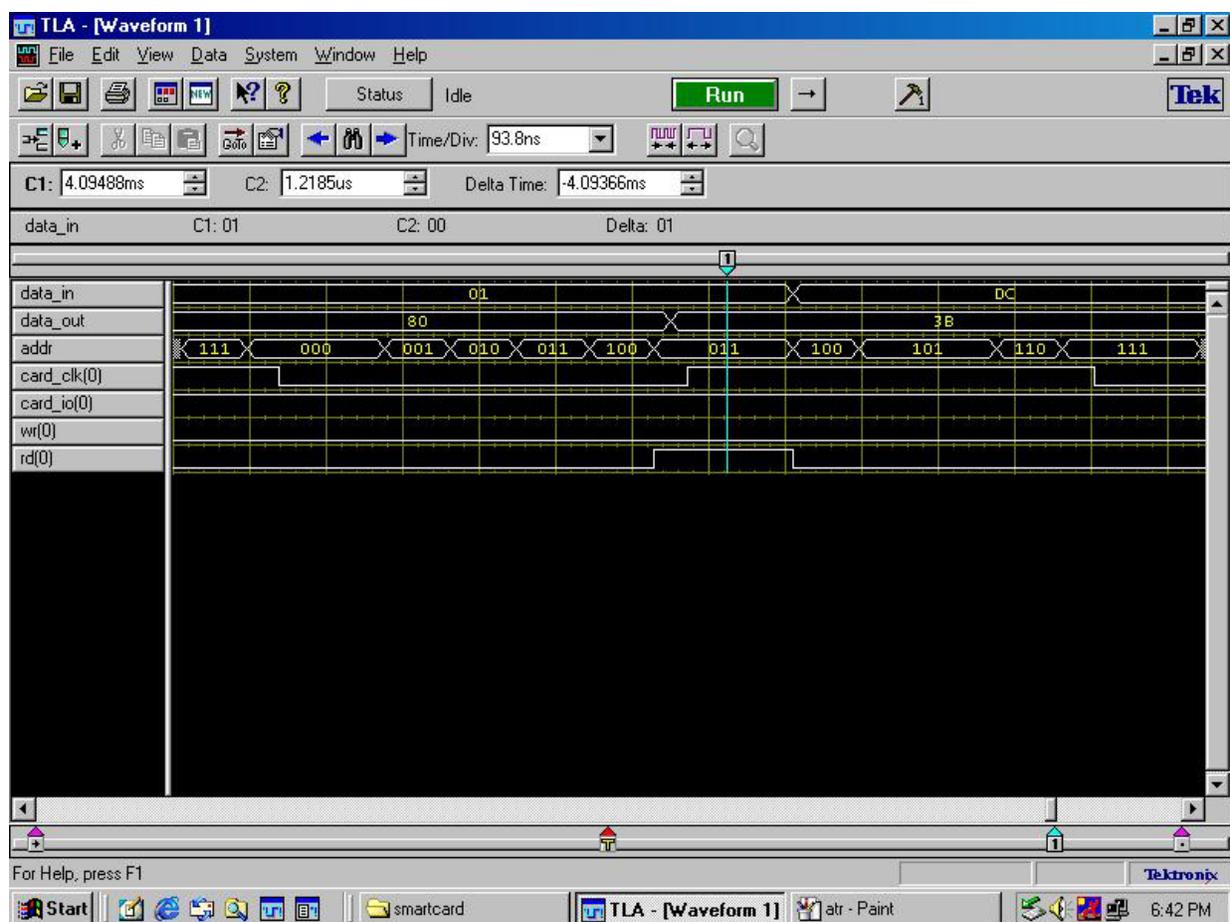
[http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=2385](http://www.atmel.com/dyn/products/product_card.asp?part_id=2385)

## Timing Diagrams

The result of each operation is captured on Logic Analyzer. The waveforms are shown in Figure 20 through Figure 24 on page 21.

### Answer To Reset (ATR)

The waveform in Figure 20 shows the first byte (0x3B) of the answer to reset (ATR) sent by the smart card. This byte is one of the eight bytes sent by the smart card as ATR when responding to a reset.



**Figure 20 • ATR Response from Smart Card**

For more information on ATR, refer the following documents:

[http://www.atmel.com/dyn/resources/prod\\_documents/2030s.pdf](http://www.atmel.com/dyn/resources/prod_documents/2030s.pdf)

[http://www.atmel.com/dyn/resources/prod\\_documents/doc5025.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc5025.pdf)

## User Zone Write Operation

The waveform in Figure 21 shows the data (0x8A) to be written into the user zone.

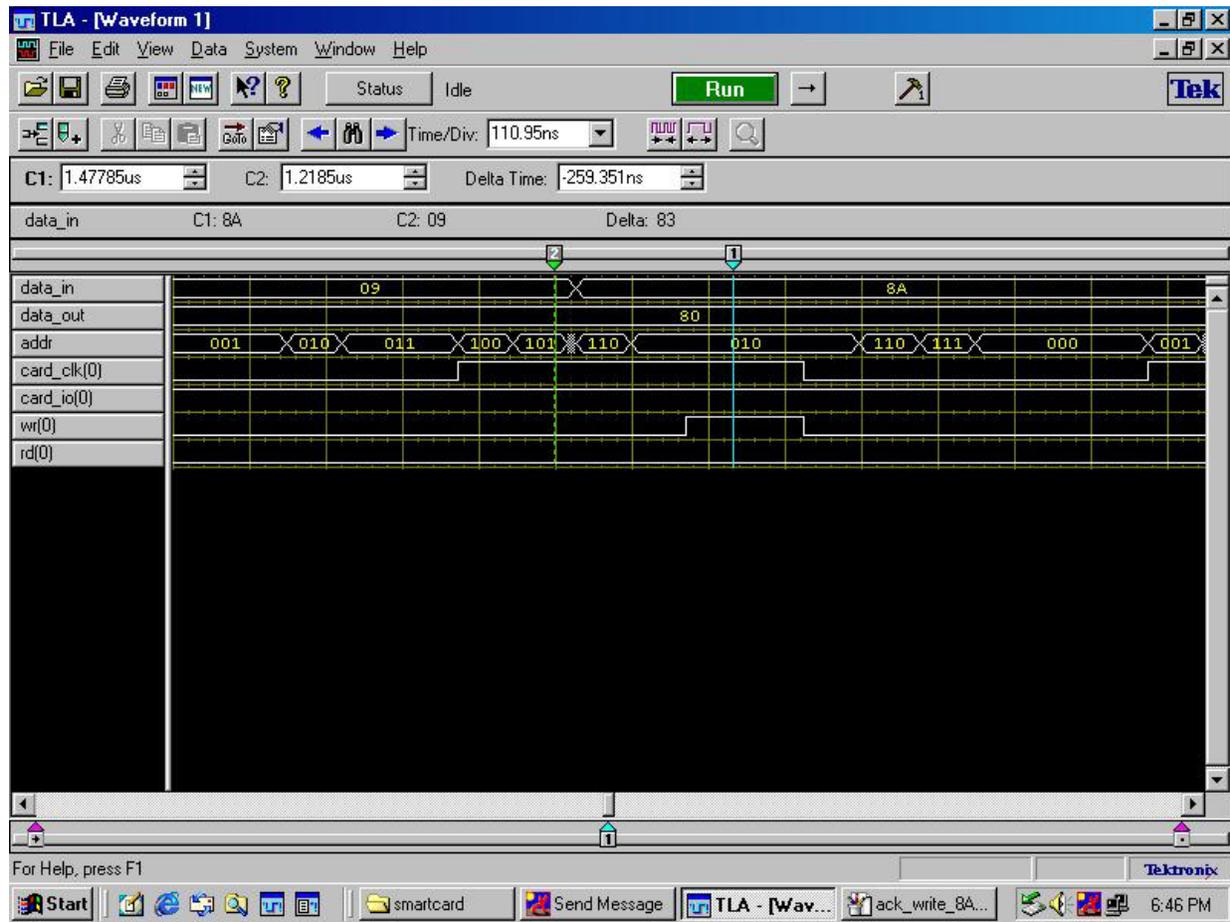


Figure 21 • User Zone Write Operation

## User Zone Read Operation

The waveform in Figure 22 shows the data (0x8A) read from user zone.

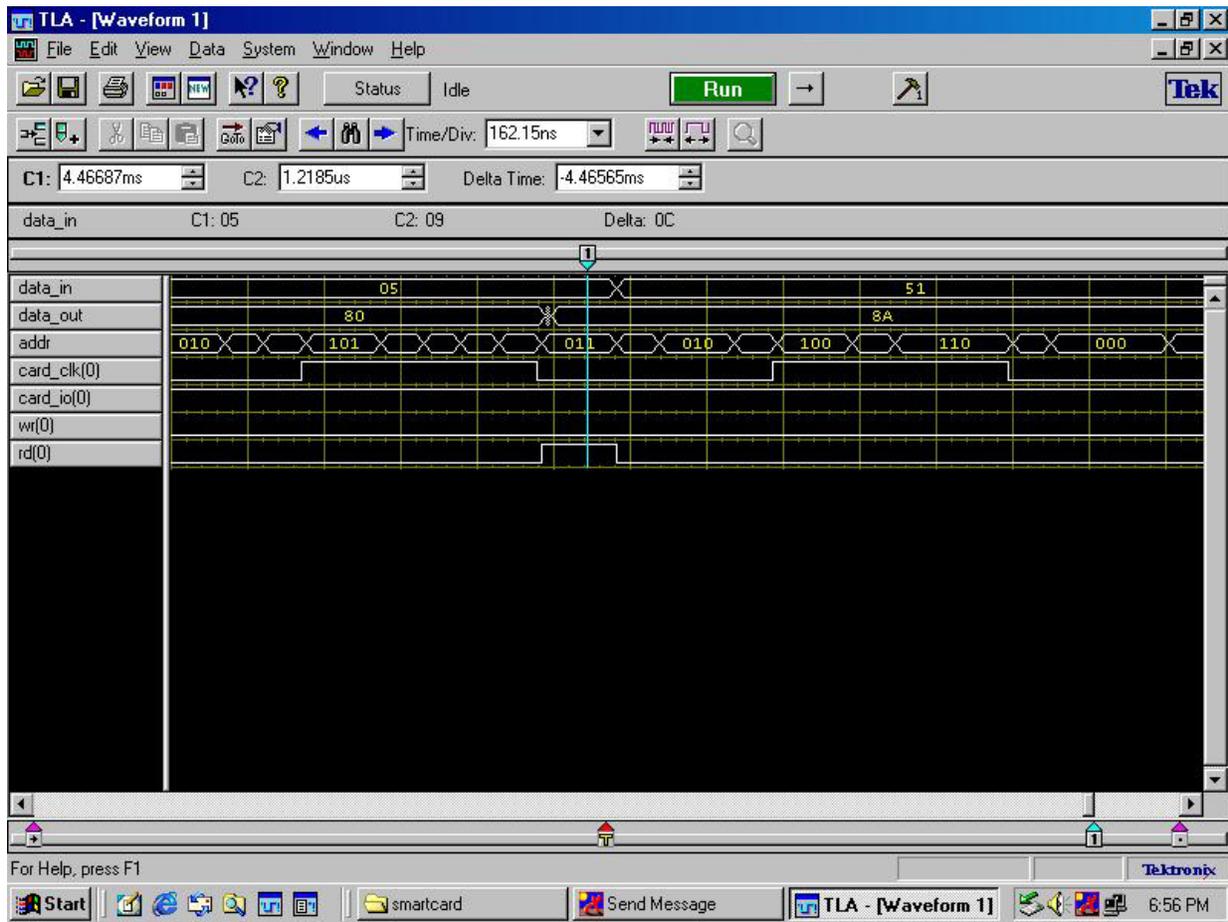
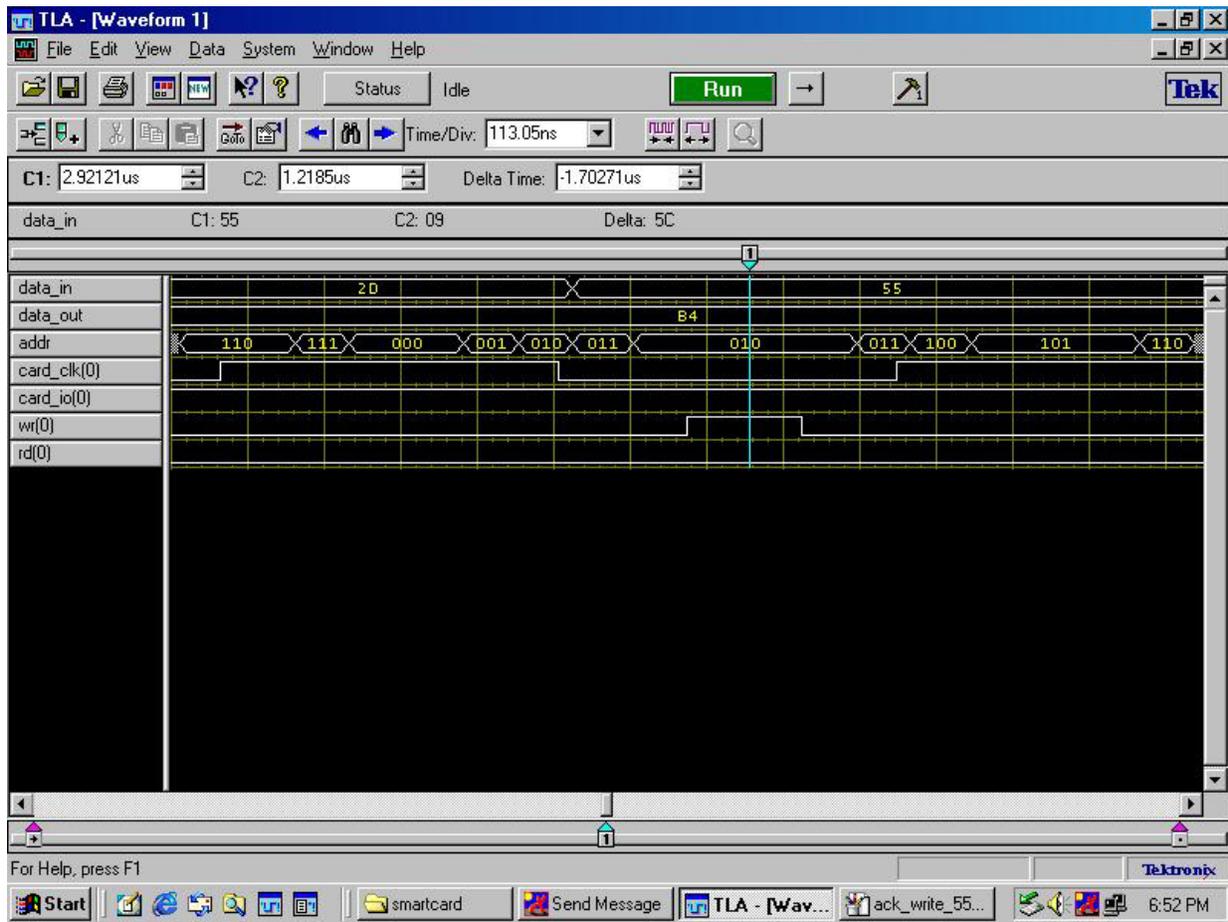


Figure 22 • User Zone Read Operation

## Configuration Memory Write Operation

The waveform in Figure 23 shows the data (0x55) to be written into the configuration zone.



**Figure 23 • Configuration Memory Write Operation**

## Configuration Memory Read Operation

The waveform in Figure 24 shows the data (0x55) read from configuration zone.

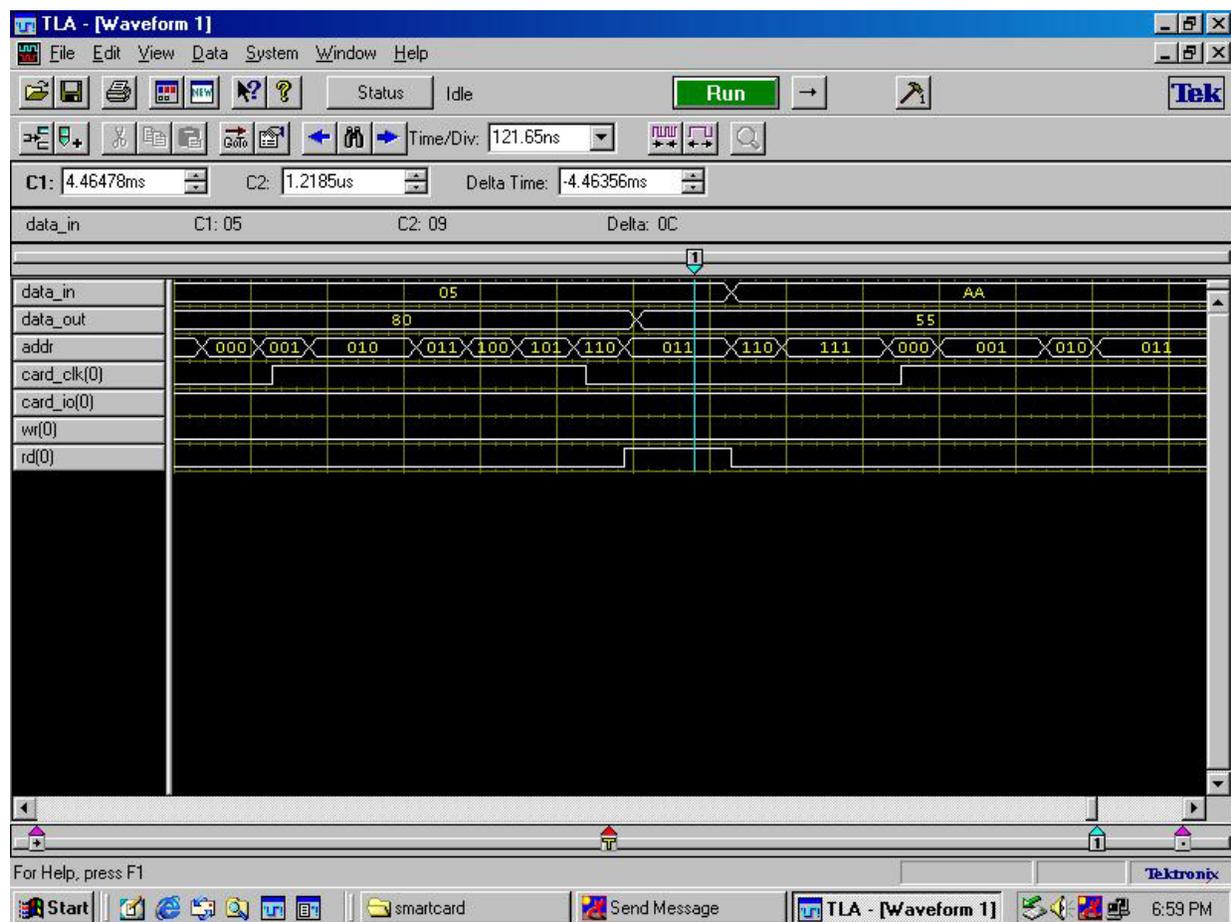


Figure 24 • Configuration Memory Read Operation

## Application Area

The smart card interface design discussed in this document can be used in a variety of security-dependent applications, including access control for a secure area, storage of security keys for authentication, as a re-chargeable travel ticket, and parking fee control.

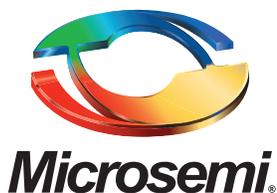
## Conclusion

Smart cards are known for their low cost and robust security. The smart card reader implementation using Microsemi's low-power FPGA eliminates the external microprocessor and the associated operating system requirement. This reduces the cost of the board and development, allowing products to get to market faster.

## List of Changes

The following table lists critical changes that were made in each revision of the document.

<b>Revision</b>	<b>Changes</b>	<b>Page</b>
Revision 1 (March 2015)	Non-Technical Updates.	N/A
Revision 0 (June 2009)	Initial Release.	N/A



**Microsemi Corporate Headquarters**  
One Enterprise, Aliso Viejo,  
CA 92656 USA

**Within the USA:** +1 (800) 713-4113  
**Outside the USA:** +1 (949) 380-6100  
**Sales:** +1 (949) 380-6136  
**Fax:** +1 (949) 215-4996

**E-mail:** [sales.support@microsemi.com](mailto:sales.support@microsemi.com)

© 2015 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at [www.microsemi.com](http://www.microsemi.com).

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.