SMART ARM-based Microcontroller

AT10700: Smart Card Interface using USART ISO7816 in SAM L22 MCU

APPLICATION NOTE

Introduction

This application note explains about interfacing smart card with Atmel® | SMART SAM L22 Microcontroller (MCU) using ISO7816 mode in SERCOM USART Module.

This application note explains the hardware / software setup used in this smart card application, SAM L22 device configurations required to enable ISO7816 mode in SERCOM peripheral for this application. It also provides an overview of ISO7816 protocol and introduction to AT88SC1616C smart card module.

Table of Contents

1. SAM L22 Device Configuration for Smart Card Interface

Smart card can be interfaced with MCU using ISO7816 protocol, a standard for electronic identification cards with contacts. This section explains the MCU interface detail such as required I/O pins, USART Clock details and ISO7816 Mode in SERCOM Module. ISO7816 protocol related information is explained in Overview of ISO7816 Protocol on page 6.

The following hardware and software used are used in this application:

1. **Hardware:**
   – SAM L22 Xplained Pro kit
   – Smart card Xplained Pro extension wing
   – Smart card (AT88SC1616C Atmel CryptoMemory in ISO Module form)
2. **Software:**
   – Smart card Quick Start Example for SAM L22 Xplained Pro kit is available in Atmel Studio 7

1.1. Pin Interface Details

ISO7816 mode in SERCOM USART Module is a two pin interface (Data & Clock). TXD of USART is used for Data line and CLK pin is used for Clock. In ISO7816 mode, Receiver and Transmitter uses the same I/O pin for Data as this is half duplex communication. Reset pin of the smart card should be connected to an I/O pin of MCU to provide reset signal to the card.

Other I/O pins can be used by card holder or socket to provide VCC supply, to detect card insertion etc.

The SERCOM pins used are multiplexed with other I/O functionalities. User must first program the I/O pin controller to assign the required I/O pins to SERCOM peripheral function. If I/O lines of the SERCOM are not used by the application, they can be used for other functionality.

Following table shows the I/O pins used to interface the smart card with SAM L22 MCU.

Table 1-1. Pin Interface Details for Smart Card with SAM L22 MCU

SAM L22 I/O Pin	Smart Card Pin	Function
USART_CLK	CLK	Smart Card Clock
USART_TXD	I/O	Smart Card Data USART Input / Output (Bidirectional)
Any GPIO (For Reset)	RST	Smart Card Reset
Any GPIO (For card detection)	Not Applicable	Connected to card holder /socket to indicate card insertion
VCC	VCC	Supply
GND	GND	Ground

1.2. USART Clock Configuration

The Baud rate generator in SERCOM module generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

This ISO7816 interface is synchronous communication. For synchronous communication, the /2 (divide-by-2) output is used. This functionality is automatically configured, depending on the selected operating mode(CTRLA.CMODE).

Refer to SERCOM chapter, Clock Generation – Baud-Rate Generator section in SAM L22 datasheet for details on configuring the baud rate.

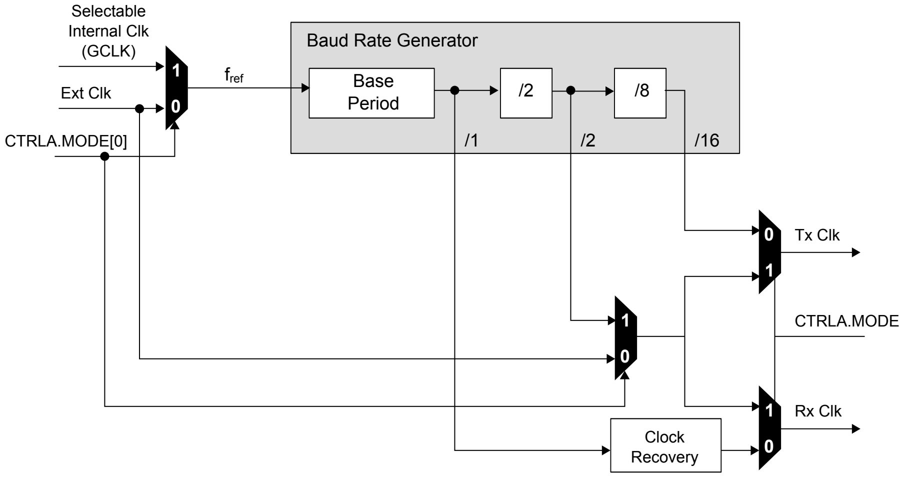
Following table shows the equation of baud rate generator.

Table 1-2. Baud Rate Equation

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

Following figure is the block diagram of Baud rate generator.

Figure 1-1. Baud Rate Generator



1.3. SERCOM USART Module - ISO7816 Mode for Smart Card Interface

Following information is available in SAM L22 datasheet for ISO7816 configuration in SERCOM USART peripheral.

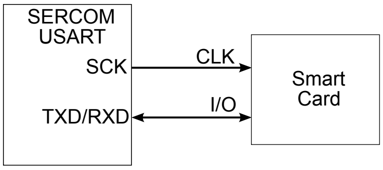
This description contains register *(REGISTER_NAME.BITFIELD_NAME)* level configuration details. Refer SAM L22 datasheet for detailed information on Register and Bit field descriptions.

The SERCOM USART features an ISO/IEC 7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO 7816 link. Both T=0 and T=1 protocols defined by the ISO 7816 specification are supported.

ISO 7816 is available with the following configuration:

- ISO 7816 format (CTRLA.FORM = 0x07)
- Inverse transmission and reception (CTRLA.RXINV=1 and CTRLA.TXINV=1)
- Single bidirectional data line (CTRLA.TXPO and CTRLA.RXPO configured to use the same data pin)
- Even parity (CTRLB.PMODE=0)
- 8-bit character size (CTRLB.CHSIZE=0)
- T=0 (CTRLA.CMODE=1) or T=1 (CTRLA.CMODE=0)

Figure 1-2. Connection of a Smart Card to the SERCOM USART



Protocol T=0

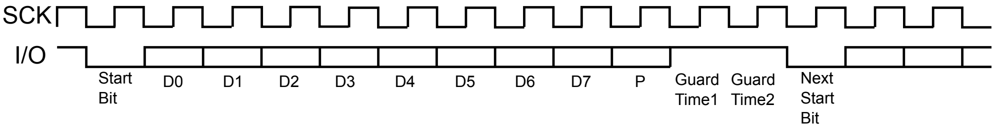
In T=0 protocol, a character is made up of:

- one start bit,
- eight data bits,
- one parity bit
- and one guard time, which lasts two bit times.

The transfer is synchronous (CTRLA.CMODE=1). The transmitter shifts out the bits and does not drive the I/O line during the guard time. Additional guard time can be added by programming the Guard Time (CTRLC.GTIME).

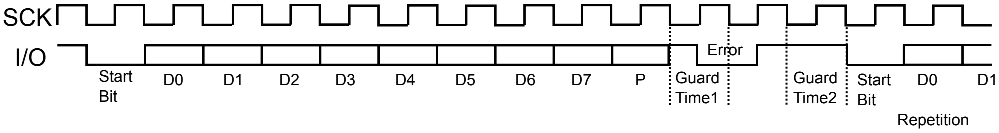
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the figure below.

Figure 1-3. T=0 Protocol without Parity Error



If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in the next figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time, which lasts 1 bit time.

Figure 1-4. T=0 Protocol with Parity Error



When the USART is the receiver and it detects a parity error, the parity error bit in the Status Register (STATUS.PERR) is set and the character is not written to the receive FIFO.

2. Overview of ISO7816 Protocol

ISO/IEC 7816 is an international standard related to electronic identification cards with contacts, especially Smart Cards, managed jointly by the International Organization for Standardization (ISO) and the International Electro technical Commission (IEC).

The high level information of some of the ISO7816 standards include following:

- **7816-1:** Physical characteristics
- **7816-2:** Cards with contacts — Dimensions and location of the contacts
- **7816-3:** Cards with contacts — Electrical interface and transmission protocols
- **7816-4:** Organization, security and commands for interchange

ISO/IEC 7816-1 defines the physical characteristics of the Smart Card.

ISO/IEC7816-2 defines eight contacts for the Smart Card which six of them are usually used for communication.

Following sections gives an overview of ISO/IEC7816-3 and ISO/IEC7816-4 protocol standard. For detailed information about ISO/IEC7816-3 and ISO/IEC7816-4 standards, refer to ISO7816 reference documents in References on page 18.

2.1. Smart Card Operation Principles

The Smart Card operations begin from the mechanical insertion of the Smart Card. The interactions between the interface device and the Smart Card shall be conducted through the following operations:

- **Smart Card device configuration**
  - Detection of the Smart Card insertion to the interface device (optional)
  - Warm reset of the smart card by the Interface device
  - Answer-to-Reset (ATR) response by the Smart Card to the interface device
  - Protocol and Parameter Selection (PPS) exchange
- **Information exchange**
  - Execution of the commands between the Smart Card and the Interface device

2.2. Smart Card Device Configuration

The ISO/IEC 7816-3 specifies electrical interface and transmission protocols for asynchronous cards, which includes the ATR mechanism. ATR establishes a basic communication channel between the Smart Card and the reader. The Protocol and PPS is used for setting the parameters responded by ATR.

2.2.1. Answer-To-Reset(ATR)

ATR is a series of characters responded to by the card after the successful reset operation. ATR characters determine the initial communication parameters, bit timing, and the data transfer details between the card and the interfacing device.

The ATR message, which less than 33 characters, consists of these fields:

- Initial byte (TS) (mandatory)
- Format byte (T0) (mandatory)
- Interface bytes (TAi,TBi,TCi and TDi) (optional)
- Historical bytes (T1,T2,...TK) (optional)
- Check bytes (TCK) (conditional)

The TS is used to establish the bit-signaling and bit-ordering conventions between the interface device and the card. TS have two possible patterns which should both be supported by the interface device.

The T0 byte is used to signal the presence or absence of subsequent interface bytes or Historical bytes. Interface bytes are used to tailor the characteristics of the I/O channel, including the specific protocol used by the card and interface device during subsequent exchange of commands and responses.

The historical bytes, if present, are used to convey manufacturer information. There is no established standard for the information presented in the historical bytes.

2.2.2. PPS request and Response

The PPS request and response consists of an initial byte, PPSS, followed by a format byte PPS0, three optional parameter bytes, PPS1, PPS2, PPS3, and a check byte PCK as the last byte.

- PPSS identifies the PPS request or response and is set to 0xFF
- In PPS0 byte, bit 8 is reserved for future use and set to 0. Each bit 5, bit 6, or bit 7 set to '1' indicate the presence of an optional byte, PPS1, PPS2, PPS3, respectively. Bit 4 to bit 1 encode the protocol type T to propose a transmission protocol. (i.e., T = 0 or T = 1 or T = 2, etc.)
- PPS1 is encoded in the same way as in TA1 (Interface byte)
- PPS2 is encoded in the same way as the first TB byte for T = 15
- PPS3 is reserved for future use
- Exclusive-ORing all the bytes PPSS to PCK inclusive shall result as 0x00, any other value is invalid

In the most common cases, the PPS response is identical to the PPS request.

For more information on PPS exchange, refer to ISO7816 reference documents in References on page 18.

2.3. Smart Card Data Exchange

After the ATR and PPS exchange between the card and the interfacing device, the next step is to execute the command(s) between the card and the interfacing device.

Currently, there are two protocols, that are widely used for Smart Card communication:

- T = 0 (asynchronous half-duplex character transmission protocol)
- T = 1 (asynchronous half-duplex block transmission protocol)

2.3.1. Application Protocol Data Unit

The Smart Card performs the requested operations and communicates the result as a response from the Smart Card.

This command response message pair is known as an Application Protocol Data Unit (APDU).

Table 2-1. Command APDU Structure

Command Header				Common Body		
CLA	INS	P1	P2	[Lc Field]	[Data Field]	[Le Field]

Preceding table defines the structure of command APDU. APDU consists of a command header and a command-body. The command header includes CLA, INS, P1 and P2 fields. As in the T=0 protocol, CLA indicates the class of the command.

INS indicates the command to process. P1 and P2 are the parameter bytes and indicate controls and options for processing the command. The body of the APDU can vary in size and is used to transmit data to the card's APDU processor as part of a command or to convey a response from the card to the interface device. The Lc field specifies the number of bytes to be transmitted to the card as part of the instruction, i.e., the length of the data field. The data field contains information that must be sent to the card to allow its APDU processor to execute the command specified in the APDU. The *Le field* specifies the number of bytes that will be returned to the reader in the response APDU.

The body of the APDU can take four different forms:

- **Case 1**: No data is transferred to or from the card, so the APDU only contains the header
- **Case 2**: No data is transferred to the card, but data is returned from the card. The body of the APDU only contains a non-null Le field
- **Case 3**: Data is transferred to the card, but none is returned from it. The body of the APDU includes the Lc and data fields
- **Case 4**: Data is transferred to the card and is also returned from the card as a result of the command. The body of the APDU includes the Lc, data and Le fields

Table 2-2. Response APDU Structure

Body	Common Body	
Data Field	SW1	SW2

Preceding table defines the response APDU structure. It consists of a body and a trailer. The body is either null or it includes a data field – depending on the specific command. The length of the data field is determined by the Le field in the corresponding command APDU. The trailer consists of up to two fields of status bytes called SW1 and SW2. These fields return a status code in which one byte is used to specify an error category and the other is used to specify a command-specific status or error indication.

2.3.2. T=0 Protocol

The T=0 protocol is a byte-oriented protocol where a character is transmitted across the channel between the interface device and the card.

Error handling is performed on each byte by looking at the parity bit. If the actual parity bit does not correspond to the parity of the transmitted data, an error must have occurred. In the T=0 protocol, the receiving side signals require the byte to be retransmitted in case of detecting a parity error. This is done by holding the I/O line low (normally the I/O line is set high before the transfer of a byte). When the transmitting side detects the I/O line low, it resends the byte that was not received by the receiving side correctly.

The APDU for T=0 protocol consists of two distinct structures:

- A command sent from the interface device to the card
- A response sent from the card to the interface device

The command header includes the following five fields; the length of each field is one byte:

- **CLA:** class designation of the command set to establish a collection of instructions
- **INS:** specifies a specific instruction from within the set of instructions
- **P1:** used to specify the addressing used by the [CLA, INS] instruction
- **P2:** also used to specify the addressing used by the [CLA, INS] instruction
- **P3:** specifies the number of data bytes transferred to or from the card as part of the [CLA, INS] instruction execution

Each value of CLA defines an application-specific set of instructions.

2.3.3. T=1 Protocol

The T=1 protocol is a block-oriented protocol where a block is transmitted across the channel between the interface device and the card. A block is a byte string conveyed in asynchronous characters.

The T=1 transmission protocol defines three types of blocks.

- An information block (I-block) is used to convey information for use by the application layer. In addition, it conveys a positive or negative acknowledgment.
- A receive ready block (R-block) is used to convey a positive or negative acknowledgment. Its information field shall be absent.
- A supervisory block (S-block) is used to exchange control information between the interface device and the card. Its information field may be present depending on its controlling function.

Table 2-3. Block Frame Structure

Prologue field			Information field	Epilogue field
Node address	Protocol control byte	Data length	Optional (INF)	Error detection (EDC)
1 byte	1 byte	1 byte	0-254 Bytes	1 or 2 Bytes

Preceding table defines the block frame structure. The three fields involved in the block frames are:

- Prologue field
- Information field
- Epilogue field

For more information about the block frame structure, refer ISO7816 reference documents in References on page 18.

3. Atmel CryptoMemory (AT88SC1616C) Smart Card

A smart card is a type of chip card, typically plastic card. This Smart card contains an embedded computer chip either a memory or microprocessor type that stores and transacts data. This data is usually associated with either value, information, or both and is stored and processed within the card's chip. The card data is transacted via a reader that is part of a computing system. Systems that are enhanced with smart cards are in use today throughout several key applications including healthcare, banking, entertainment, and transportation.

Atmel® AT88SCxxxxC is a family of high-performance secure memory devices providing 1kbit to 256kbits of user memory with advanced built-in security and cryptographic features. This card also comes in ISO module form that can be used for smart card applications. The memory is divided into 4, 8, or 16 user zones each of which may be individually set with different security access rights or used together to effectively provide space for one or multiple data files.

Atmel CryptoMemory® offers the ability to communicate with virtually any smart card reader using the asynchronous T=0 protocol defined in ISO 7816-3. For devices with 32kbits of user memory and larger, communication speeds up to 153,600 baud are supported by utilizing ISO 7816-3 protocol and parameter selection. All CryptoMemory devices in smart card module form will also communicate using a synchronous 2-Wire serial interface.

3.1. Package Information and Pin Configuration

The package detail and pin assignment information of Atmel CryptoMemory are as follows:

Figure 3-1. ISO Smart Card Module

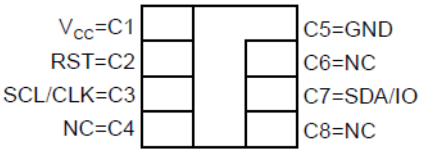


Table 3-1. ISO Module Package Pin Assignment

Pad	Description	ISO module
VCC	Supply Voltage	C1
GND	Ground	C5
SCL/CLK	Serial Clock Input	C3
SDA/IO	Serial Data Input/Output	C7
RST	Reset Input	C2

3.2. Reset Sequence

The power-up sequence which complies with ISO 7816-3 for a cold reset in smart card applications is as follows:

- Power up VCC
- RST, I/O, and CLK are low
- Set I/O in Receive mode
- Provide a clock signal to CLK

- RST goes high after 400 clock cycles.

CryptoMemory provides an ISO 7816-3 compliant asynchronous Answer-To-Reset (ATR) sequence. When the reset sequence is activated, the device will output the data programmed into the 64-bit ATR register.

3.3. Device ATR

The device will respond with a 64-bit ATR code including historical bytes to indicate the memory density within the CryptoMemory family.

The 64-bit ATR code comes from a register which contains the characters shown in following table. The historical bytes (T1, T2, T3) show the density of the CryptoMemory device. This register may be modified during customization.

Table 3-2. ATR Code for AT88SC1616C Atmel CryptoMemory

Device	TS	T0	TA(1)	TB(1)	TD(1)	TA(2)	T1	T2
AT88SC1616C	$3B	$B2	$11	$00	$10	$80	$00	$16

3.4. Memory Test Zone (MTZ)

The Atmel CryptoMemory have different memory zones and we are using Memory Test Zone. This field is a 16-bit wide register with open read/write access privileges at all times for testing basic communication to the device. This field is free of all security constraints at all times.

3.5. Command Set

The command set of CryptoMemory is provided in this section. Each instruction sent to the CryptoMemory must have four bytes:

- Command
- Address 1
- Address 2
- N

The last byte, N, defines the number of any additional data bytes to be sent or received from the CryptoMemory device. Complete command set is available in Table Atmel CryptoMemory Synchronous Command Set in CryptoMemory datasheet. The following table shows some of the commands for AT88SC1616C device.

Table 3-3. CryptoMemory(AT88SC1616C) Synchronous Command Set

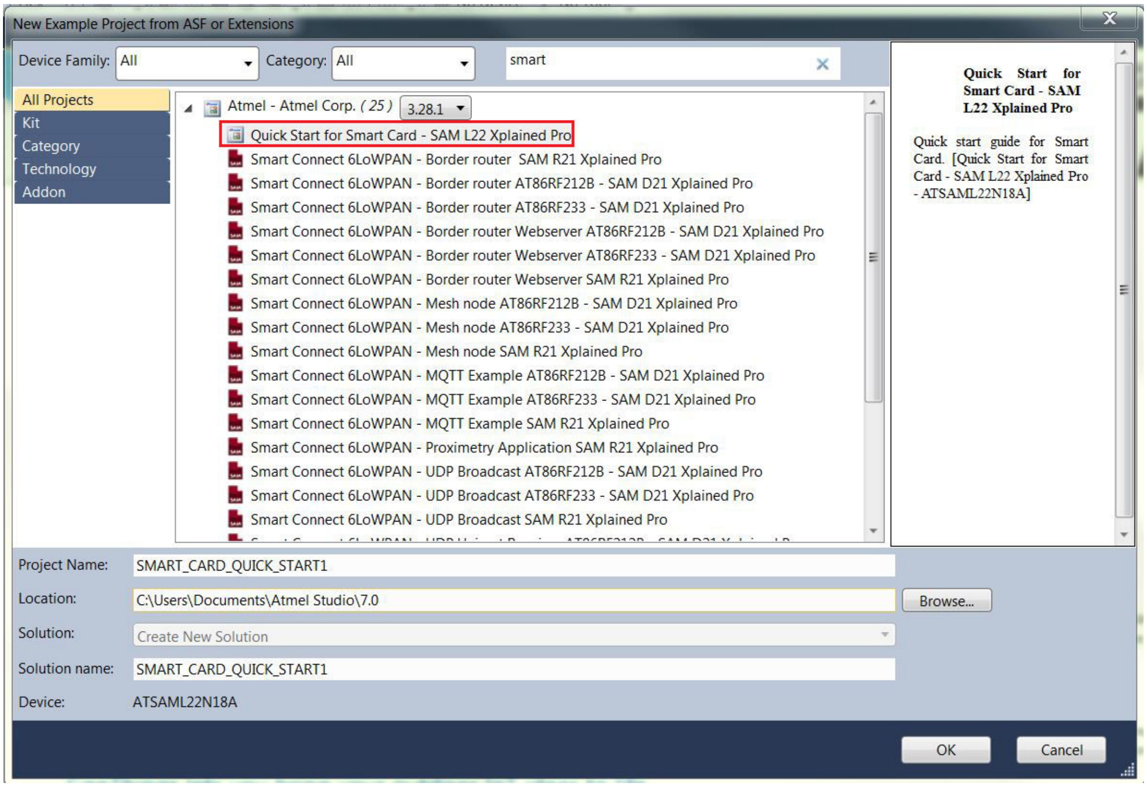
Command description		Command	ADDR 1	ADDR 2	N	Data (N)
Write User Zone	Normal	$B0	ADDR	ADDR	N≤$10	N-bytes
Read User Zone		$B2	ADDR	ADDR	N	
System Write	Write Config Zone	$B4	$00	ADDR	N≤$10	N-bytes
	Write Fuses	$B4	$01	Fuse ID	$00	
	Send Checksum	$B4	$02	$00	$02	2-bytes
	Set User Zone	$B4	$03	Zone	$00	

Command description		Command	ADDR 1	ADDR 2	N	Data (N)
System Read	Read Config Zone	$B6	$00	ADDR	N	
	Read Fuse Byte	$B6	$01	$00	$01	
	Read Checksum	$B6	$02	$00	$02	

4. ASF Quick Start Example for Smart Card

Atmel Studio 7 with ASF 3.27.3 or above has Quick start Example for Smart card. In this application, Write and Read command to Memory test zone has been performed and verified. User Inputs and Output console is demonstrated through Virtual COM port of Embedded Debugger (EDBG) present in SAM L22 Xplained Pro.

Figure 4-1. Smart Card Example in ASF



Device Family: All
Category: All
smart

Atmel - Atmel Corp. (25) 3.28.1

- Quick Start for Smart Card - SAM L22 Xplained Pro
- Smart Connect 6LoWPAN - Border router SAM R21 Xplained Pro
- Smart Connect 6LoWPAN - Border router AT86RF212B - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - Border router AT86RF233 - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - Border router Webserver AT86RF212B - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - Border router Webserver AT86RF233 - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - Border router Webserver SAM R21 Xplained Pro
- Smart Connect 6LoWPAN - Mesh node AT86RF212B - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - Mesh node AT86RF233 - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - Mesh node SAM R21 Xplained Pro
- Smart Connect 6LoWPAN - MQTT Example AT86RF212B - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - MQTT Example AT86RF233 - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - MQTT Example SAM R21 Xplained Pro
- Smart Connect 6LoWPAN - Proximetry Application SAM R21 Xplained Pro
- Smart Connect 6LoWPAN - UDP Broadcast AT86RF212B - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - UDP Broadcast AT86RF233 - SAM D21 Xplained Pro
- Smart Connect 6LoWPAN - UDP Broadcast SAM R21 Xplained Pro

Quick Start for Smart Card - SAM L22 Xplained Pro

Quick start guide for Smart Card. [Quick Start for Smart Card - SAM L22 Xplained Pro - ATSAML22N18A]

Project Name: SMART_CARD_QUICK_START1
Location: C:\Users\Documents\Atmel Studio\7.0
Solution: Create New Solution
Solution name: SMART_CARD_QUICK_START1
Device: ATSAML22N18A

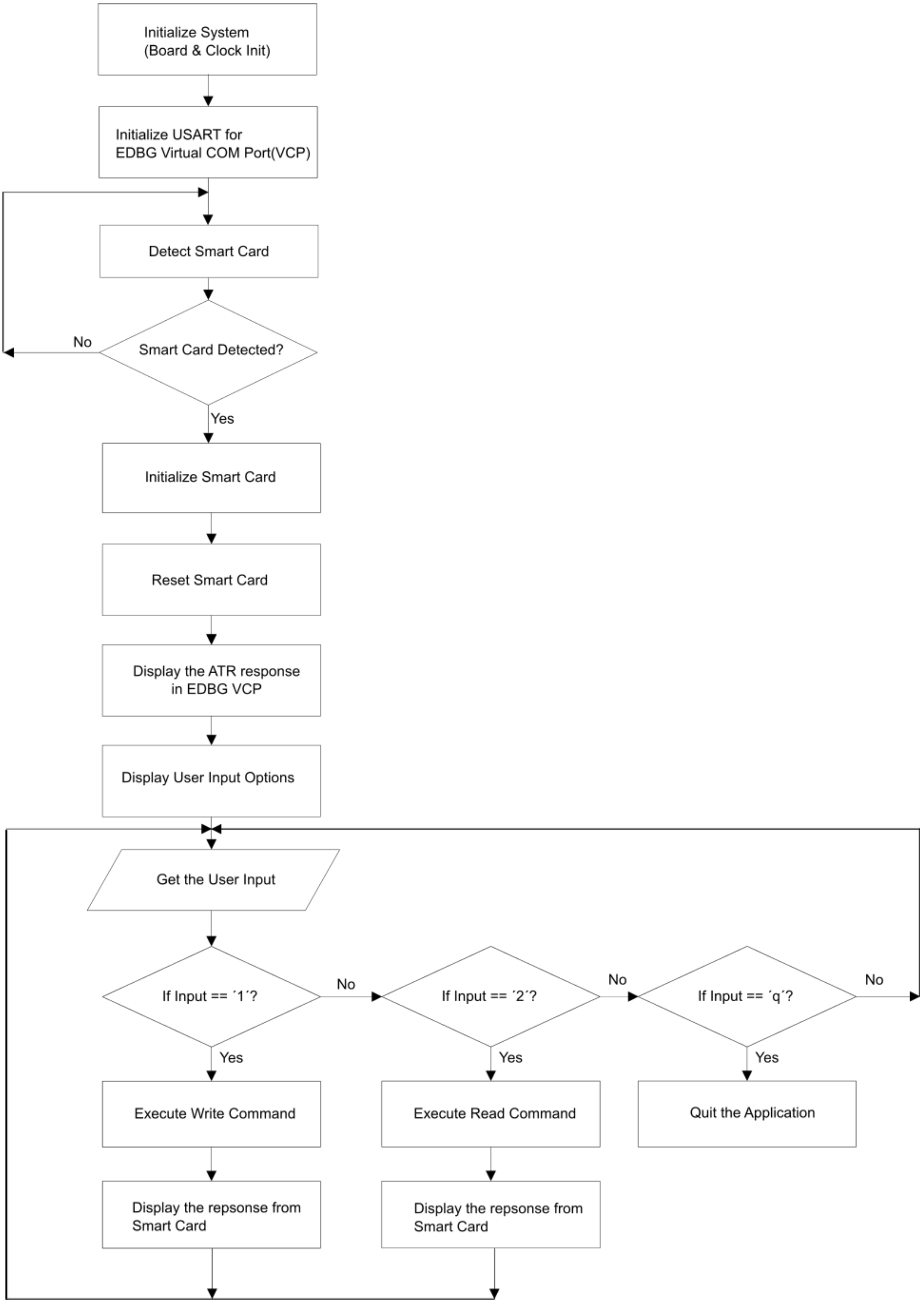
4.1. Software Overview

In the Quick start example, following are the relevant files required for this example. Apart from these files, example also uses other drivers from ASF such as board support, clock, SERCOM, etc.

- `qs_smart_card.c` (main file)
- `iso7816.c`(ISO7816 driver source file available in ASF path: `ASF\sam0\components\smart_card`)
- `iso7816.h` (ISO7816 driver header file available in ASF path: `ASF\sam0\components\smart_card`)
- `conf_iso7816.h` (configuration file)

The application flow is shown in the following illustration.

Figure 4-2. Application Flowchart



The following table shows the functions available in ISO7816 driver files in ASF that are being used in the example.

Table 4-1. Functions Available in ISO7816 Driver Files

Function name	Description
`void iso7816_init(struct usart_module *const module, uint32_t pin_rst, uint32_t clock_get_hz);`	Function to Initialize ISO7816 mode in USART Module
`void iso7816_cold_reset(void);`	Function to perform a cold reset sequence to smart card
`void iso7816_warm_reset(void);`	Function to perform a warm reset sequence to smart card
`enum status_code iso7816_send_char(uint8_t uc_char);`	Function to send a character
`static enum status_code iso7816_get_char(uint8_t *p_char_received);`	Function to get a character
`uint16_t iso7816_xfr_block_tpdu_t0(const uint8_t *p_apdu, uint8_t *p_message, uint16_t us_length);`	Function to send a TPDU command to a smart card

Commands Used in Code to Write and Read 2 bytes in Memory test Zone:

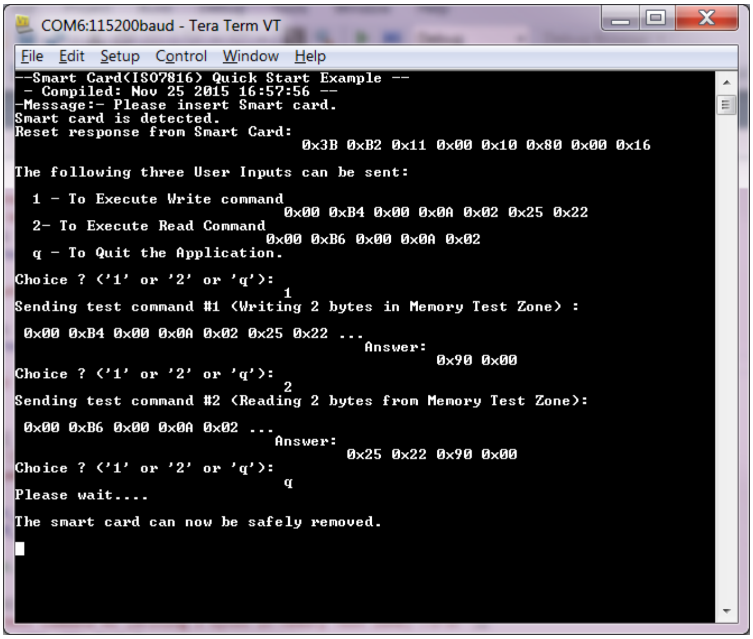
```
/* Test command #1 - Write to memory Test Zone {CLA, INS, Add, Add, No of bytes,
Data}*/
const uint8_t test_cmd1[CMD1_LEN] = {0x00, 0xB4, 0x00, 0x0A, 0x02, 0x55, 0xAA};

/* Test command #2 - Read from Memory Test Zone {CLA, INS, Add, Add, No of bytes}*/
const uint8_t test_cmd2[CMD2_LEN] = {0x00, 0xB6, 0x00, 0x0A, 0x02};
```

Decode Information of the Commands:

1. CLA - 0x00 (ISO 7816-4 instructions (files and security)).
2. 0xB4 – System Write Command for AT88SC1616C.
3. 0xB6 – System Read Command for AT88SC1616C.
4. 0x00 0x0A – Address bytes.
5. 0x55 0xAA – Data written in Memory Test Zone.

The console output of the application is provided for reference.

Figure 4-3. Console Output


COM6:115200baud - Tera Term VT
File Edit Setup Control Window Help

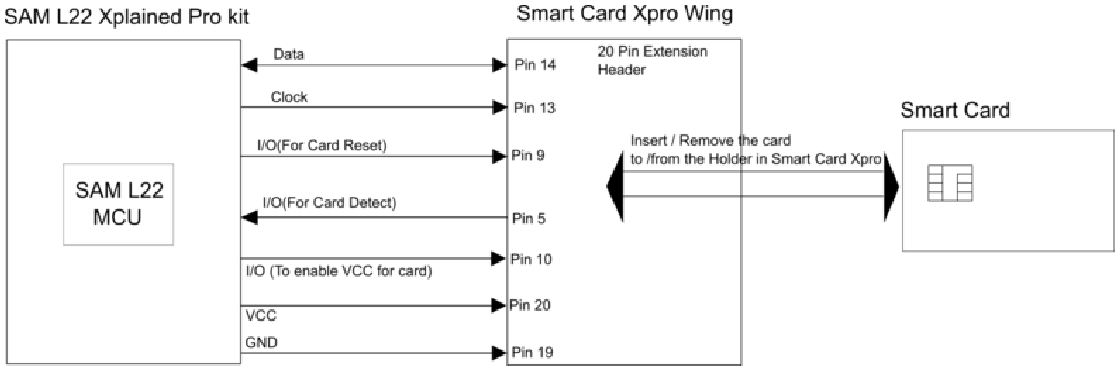
```
--Smart Card(ISO7816) Quick Start Example --
 - Compiled: Nov 25 2015 16:57:56 --
-Message:- Please insert Smart card.
Smart card is detected.
Reset response from Smart Card:
                                0x3B 0xB2 0x11 0x00 0x10 0x80 0x00 0x16

The following three User Inputs can be sent:

  1 - To Execute Write command
                              0x00 0xB4 0x00 0x0A 0x02 0x25 0x22
  2- To Execute Read Command
                            0x00 0xB6 0x00 0x0A 0x02
  q - To Quit the Application.

Choice ? ('1' or '2' or 'q'):
                              1
Sending test command #1 (Writing 2 bytes in Memory Test Zone) :

 0x00 0xB4 0x00 0x0A 0x02 0x25 0x22 ...
                                       Answer:
                                               0x90 0x00
Choice ? ('1' or '2' or 'q'):
                              2
Sending test command #2 (Reading 2 bytes from Memory Test Zone):

 0x00 0xB6 0x00 0x0A 0x02 ...
                             Answer:
                                     0x25 0x22 0x90 0x00
Choice ? ('1' or '2' or 'q'):
                              q
Please wait....

The smart card can now be safely removed.
```

4.2. Hardware Platform Overview

SAM L22 Xplained Pro with Smart Card XPRO wing is the hardware used. Smart Card XPRO wing is connected to EXT3 Connector in SAM L22 Xplained Pro board. The micro-USB cable is connected to the EDBG USB port. This cable powers the setup as well as used for displaying user messages through EDBG Virtual COM port.

The block diagram of hardware pin connection is as follows.

Figure 4-4. Pin Connection of Smart Card Application



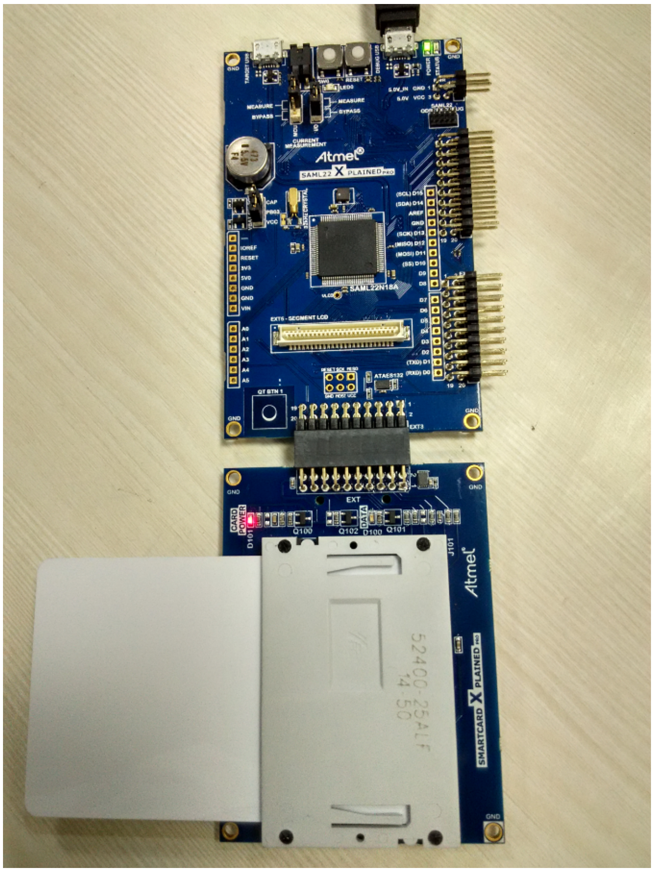
The pin details of SAM L22 device used in this application are presented in the table below:

SAML22 pin detail	SAM L22 Xplained Pro extension header 3 pin number	Smart Card Xplained Pro pin description
PA15 (USART_CLK)	Pin 13	Connected to Smart Card Clock
PA14 (USART_TXD)	Pin 14	Connected to Smart Card Data - USART Input / Output (Bidirectional)

SAML22 pin detail	SAM L22 Xplained Pro extension header 3 pin number	Smart Card Xplained Pro pin description
PB18 (GPIO)	Pin 9	Connected to Smart Card Reset
PC16 (GPIO)	Pin 5	Connected to card holder /socket to indicate card insertion
PB19 (GPIO)	Pin 10	Connected to a switch which turns on VCC to Smart Card
VCC	Pin 20	Supply
GND	Pin19	Ground

The following image illustrates the hardware setup.

Figure 4-5. Hardware Setup



5. References

1. SAM L22 Device datasheet: http://www.atmel.com/Images/Atmel-42402-SAM-L22_Datasheet.pdf
2. SAML22 Xplained Pro User guide: http://www.atmel.com/Images/Atmel-42474-SAM-L22-Xplained-Pro_User-Guide.pdf
3. Atmel CryptoMemory Datasheet: http://www.atmel.com/Images/Atmel-5211-CryptoMem-Full-Specification-Datasheet.pdf
4. Atmel CryptoMemory Application note for Smart Card applications: http://www.atmel.com/Images/doc5025.pdf
5. ISO Reference Links:
   - 5.1. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=35&ics2=240&ics3=15&csnumber=45989
   - 5.2. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38770
   - 5.3. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=36134

6. Revision History

Doc. Rev.	Date	Comments
42641A	02/2016	Initial document release.

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (“Safety-Critical Applications”) without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.