TERIDIAN SEMICONDUCTOR CORP.

Simplifying System Integration™

Using Synchronous Smart Cards with the 73S12xxF

December 18, 2008
Rev. 1.0
UG_12xxF_018

Teridian Semiconductor Corp., 6440 Oak Canyon, Suite 100, Irvine, CA 92618
TEL (714) 508-8800, FAX (714) 508-8877, http://www.teridian.com

Table of Contents

Figures

1 Introduction

This document describes how to use synchronous smart cards with the 73S12xxF family. The smart card block of the 73S12xxF family contains specific logic for running synchronous smart cards. There are three primary sync smart card types; 2-wire, 3-wire and I^2C smart cards. Each of these card types will be described and how to operate these cards with the 73S12xxF sync card logic is detailed.

This document is intended for the application developer that needs to support synchronous smart cards. Understanding of basic Smart Card operation is helpful and required. In addition, familiarity with the Teridian 73S12xxF smart card block is assumed for Asynchronous smart card operation. Refer to the 73S12xxF data sheets for a detailed description of the smart card logic and control registers.

2 Synchronous Smart Card Basics

Most smart cards are the asynchronous type that operates similar to a standard serial UART for data transfer. Async cards contain a microcontroller and a real time operating system (RTOS) to provide all the functionality of the smart card according to the ISO-7816 specification.

Sync cards are typically thought of as memory cards that do not require the microcontroller functionality of Async cards. The sync card only needs to read and write from the card's memory and, for some cards, provide simple security functions.

The async smart card interface requires a clock line (CLK) signal. This clock is not used to clock the data transfers, but is supplied as a common timebase for the serial transfer logic and the microcontroller clock source. Sync cards, as implied, use the CLK signal to clock the data one bit at a time for data transfers.

The 2-wire and 3-wire sync cards require the use of the Reset (RST) signal on the smart card interface to control the reset function of the cards, and in the case of the 3-wire card, differentiate the card command from the card output.

This document describes the format of the three types of cards based on the following:

2-wire card – SLE4432/SLE4442
3-wire card – SLE4418/SLE4428
I^2C card – AT24C1024

Refer to the card data sheets for complete information.

2.1 2-Wire Cards

2.1.1 Answer to Reset

2-wire synchronous smart cards use the VCC, ground, I/O and RST smart card interface signals. A card session begins with a card reset followed by an Answer To Reset (ATR). The ATR of a sync card is fixed at 32 bits (4 bytes). The activation sequence is shown in Figure 1.

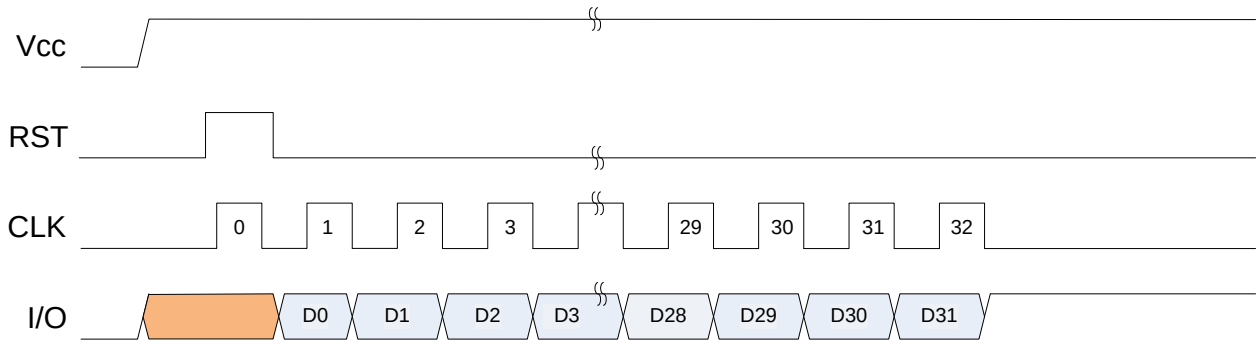


Figure 1: ATR Sequence

The session starts with turning on the power to the card (Vcc). Once the Vcc is stable, the RST signal is asserted high and the CLK will output a single pulse. After the falling edge of CLK, the RST signal should be de-asserted. The CLK should then be allowed to run for 32 cycles to capture all 32 bits of the ATR. After the falling edge of the last clock pulse, the card will release the I/O line.

2.1.2 Command and Response Processing

After the ATR response has been processed, the card is now ready to accept a command. Commands are three bytes in length and are formatted as shown in Figure 2.

Byte 1	Byte 2	Byte 3
Command	Address	Data

Figure 2: Command Packet Format

Each byte is transmitted with least significant bit first. The command byte is sent first and specifies the type of command to be executed. This consists of read, write and security commands. The second byte specifies the 8-bit address for the command to operate upon. Since the address is limited to 8-bits, the size of the memory is 256-bytes max. The last byte corresponds to a single byte of data. Write operations are limited to single bytes writes. Read operations ignore the data field and are considered don't care.

Each command begins with a start bit. The start bit is defined as a falling edge on I/O while the CLK is high. The command is transferred on 24 clocks and then followed by a stop bit. Every command expects data (in case of a read operation) or an indication that it is processing the command (in the case of a write or security operation). Figure 2 shows the command read operation and Figure 3 shows the write and security command operations. Since these types of cards were based on an old flash based technology, they required a long write time. To prevent any commands from being executed while the previous one is being processed, the status can be monitored as shown in Figure 3 or a delay can be executed to allow for the processing to complete.

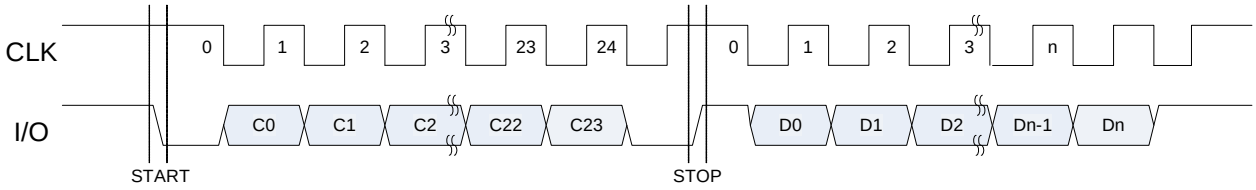


Figure 3: Read Command Operation

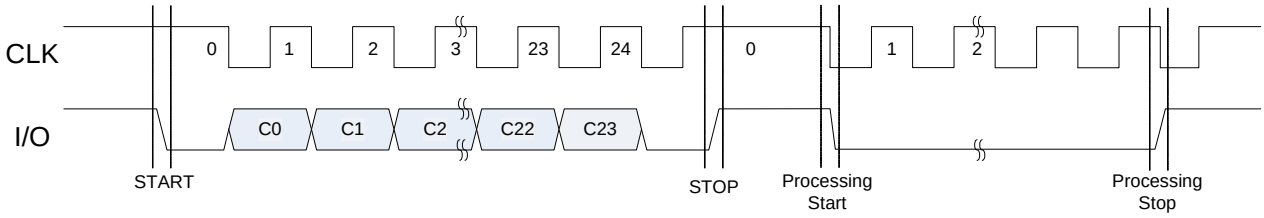


Figure 4: Write and Security Command Operation

The RST signal is not used for normal transactions, but can be used as a break signal by applying a positive pulse on RST for a minimum duration.

2.2 3-Wire Cards

2.2.1 Answer to Reset

The ATR sequence is identical to the 2-wire cards as outlined in Section 2.1.1.

2.2.2 Command and Response Processing

3-wire cards do not use start and stop bits. Instead they use the RST signal to frame the command transactions. Like the 2-wire cards, the commands are formatted in the same manner as shown in Figure 2. The command byte differs from the 2-wire cards in that it uses the two LSBs of the command byte as address MSBs. As a result, the 3-wire cards have 10 bits of address so the size of the memory is 1K bytes. The read operation is shown in Figure 5. As noted before, the write security operations take some time to complete. Like the 2-wire cards, the processing time can be monitored as shown in Figure 6 or a delay can be added to allow the processing to terminate before issuing the next command. The write and security operations are shown in Figure 6.

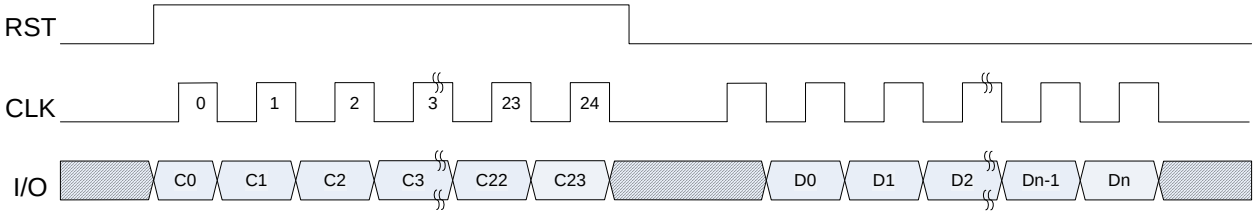


Figure 5: Read Command Operations

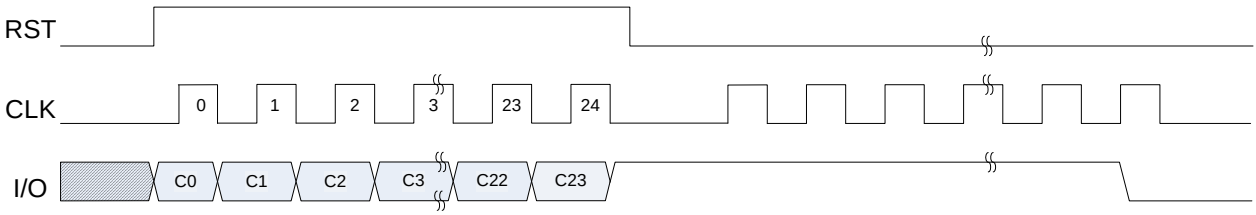


Figure 6: Write and Security Command Operations

2.3 I²C Cards

I²C cards do not support an ATR response. They operate similar to a 2-wire card where start and stop bits are used. They do not use commands, but use a combination of the read/write bit and the start bit to configure the type of operation to be performed. They do require the transmission of an I²C address and a read/write bit for each transaction. In addition, an acknowledge (ACK) bit is required after transferring every byte. For every byte transmitted by a sender an active low ACK bit must be sent by the receiver to insure proper reception. Figure 7 shows the data structure of a data packet.

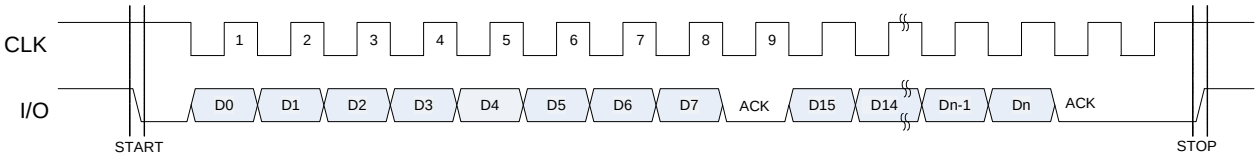


Figure 7: I²C Data Structure

The first byte of a packet must contain the device address byte. The 6 MSB must hold the I²C device address and contain a value of “1 0 1 0 0 0”. Bit 1 contains the MSB of the 17-bit address or a ‘0’ if data is being read back on the following byte. Bit 0 contains the read/write bit. Figure 8 shows the device address. The device address will always begin with a start bit and be followed by either a memory address or read data depending on the operation being performed. All write operations will require that the memory address is specified in the second byte. Read operations can be configured to use the last used address or use a command to write the address for the read and then read the actual data. This application note only considers use of the read operations with the address specified.

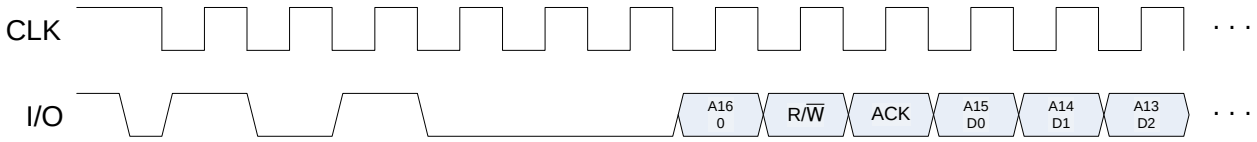


Figure 8: Device Address

2.3.1 I²C Write Operations

Write operations are formatted with the device address as the first byte. Bit 1 of the device address contains the MSB of the memory address. This is bit A16 so the memory configuration of this card is 131Kb. The second byte contains address bits A15 to A8 and the third byte contains the address bits A7 to A0. The fourth byte will contain the data to write to the address specified in the command. If a single byte is to be written, then a stop bit will follow the ACK bit. If more than one byte is to be written to successive memory locations, then the data is written in successive bytes of the command until the last one is written and then a stop bit follows it. Figure 9 shows an I²Cwrite transaction. As with all memory type cards, write operations must allow some time for the write operation to complete.

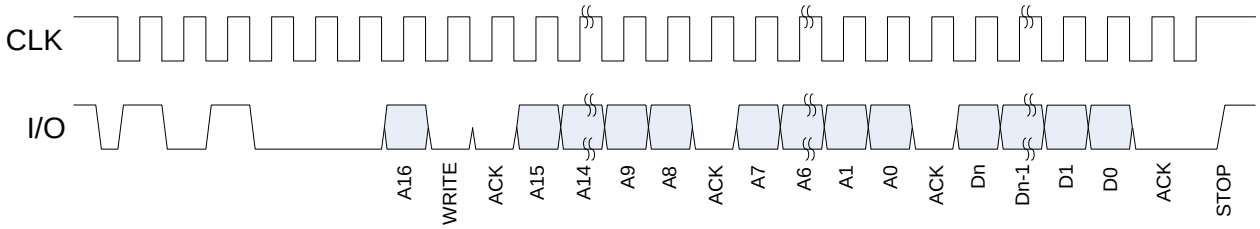


Figure 9: I²C Write Operation

2.3.2 I²C Read Operations

Read operations are a bit more complex. Read operations can start at the current address from a previous transaction or the address can be specified in the current command. When the address is specified in the current transaction, a dummy write sequence is required and then a read sequence is executed. A dummy write will look identical to the sequence specified in Figure 9 up to the ACK bit after the LSB of the address is transferred. Figure 10 shows the I²C read operation resuming from the last two bits of the dummy write. A single byte can be read followed by a negative ACK (NACK) and a stop bit. If more than one byte is to be read, then all received bytes will respond with an ACK until the last byte which is then followed by a NACK. This protocol allows the reader to determine when the last byte is to be received.

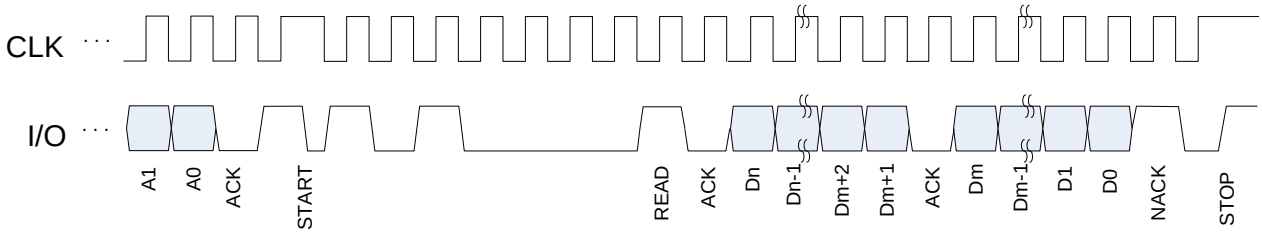


Figure 10: I²C Read Operation

3 Using Synchronous Smart Cards with the 73S12xxF

The smart card block within the 73S12xxF device contains specific support for synchronous smart cards. See the “Synchronous Operation Mode” section of the specific 73S12xxF data sheet for a description of the differences between Asynchronous and Synchronous operation within the smart card block. The purpose of this application note is to describe how the smart card block should be used to operate a synchronous smart card.

3.1 Special Precautions when Using Synchronous Smart Cards

3.1.1 Synchronous Mode Clock

The CLK output is sourced by the ETU clock. The ETU clock is a free running clock that can’t be started or stopped. As a result, the CLK output is not initialized and must be set to a high or low level before any synchronous transaction can begin. The level is determined by the card type being used. The CLK output must be started and allowed to run for a full clock cycle and then stopped at the desired logic level manually. This is not desirable during a card session, so it is recommended to perform this initialization before the card is activated. This can be done before turning on VCC to the card. The CLK output will always be at ‘0’ when VCC is off, but the smart card logic remains active. The procedure for initializing the CLK output is as follows: Note: this procedure is only necessary from a power on or device reset condition as the CLK will be stopped after any card transaction.

1. Turn on the CLK by writing ‘0’ to the CLKOFF bit in the SCCtl register.
2. Wait a minimum of 1 ETU. This is necessary to ensure that both CLK edges are seen by the logic and allowed to stop the output at the desired level.
3. Set CLKLVL to the desired logic level and write ‘1’ to the CLKOFF bit to stop the CLK output at the desired level.
4. Turn on VCC to the desired voltage.

For all synchronous transactions, the data is shifted to/from the data FIFOs on the falling edge of CLK and the data is read or written on the rising edge. The CLK logic must output a falling edge on CLK to properly configure the FIFOs for read or write operations. If the first edge of the CLK output is rising (after being started), this edge is ignored and the next rising edge will sample the first bit of the data. It should be noted that the RLEN counter is incremented on the falling edge of CLK.

3.1.2 Proper Use of Interrupts in Synchronous Mode

When operating with Asynchronous cards, the transmit buffer empty and receive register full are used to control the interrupt processing for data transfer. When using synchronous cards, these interrupts should not be used. Instead, the clock counter (RLEN) should be used to keep track of the number of clock pulses output on CLK. The best way to perform data transactions is to send data on a byte by byte basis since the data FIFOs are byte wide. Setting the RLEN register to 8 will accomplish this. This way the interrupts are generated when the FIFO is emptied or filled. The value that is written to RLEN will need to be something other than 8 due to CLK limitations described above. These exceptions will be explained later in this document.

3.2 73S12xxF Operation

The following sections describe how the three different synchronous smart card types operate with the 73S12xxF. Each section describes how the ATR is retrieved (when applicable), how to send commands to the card and read data from the card. Refer to the flowcharts in Appendix A.

3.2.1 2-Wire Card Operation

3.2.1.1 ATR Retrieval

The description of the ATR sequence for 2-wire cards is described in Section 2.2.1. A scope capture for the ATR is shown in Figure 11. Note that the RST is high during the first clock pulse. As mentioned before, since the first edge of the CLK is a rising one, the data on the I/O is undefined. This is no problem as it is a don't care during reset. The first falling edge will configure the RX FIFO to receive the first bit from the card and the next rising edge will sample the data. The flowchart for retrieving the ATR is shown in Figure 25. This flowchart includes the steps necessary for initializing the CLK output low, making sure the card is inserted (or activation is blocked by hardware), turning on the card supply (VCC), generation of the RST pulse, reading the four byte ATR value and stopping the CLK when done.

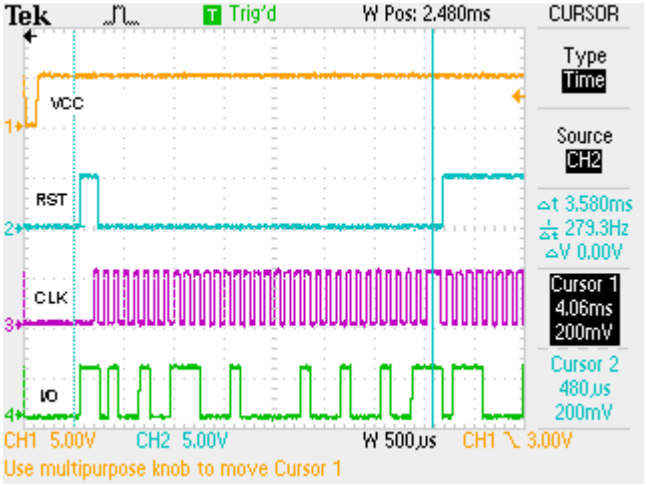


Figure 11: ATR Scope Capture

3.2.1.2 Sending Commands

2-wire card commands must follow the format as described in Section 2.1.2. The sequence must begin with a start bit and end with a stop bit. An entire command frame shown in Figure 12. A zoom of the start bit located at the left cursor of Figure 12 is shown in Figure 13. A zoom of the stop bit located at the right cursor of Figure 12 is shown in Figure 14.

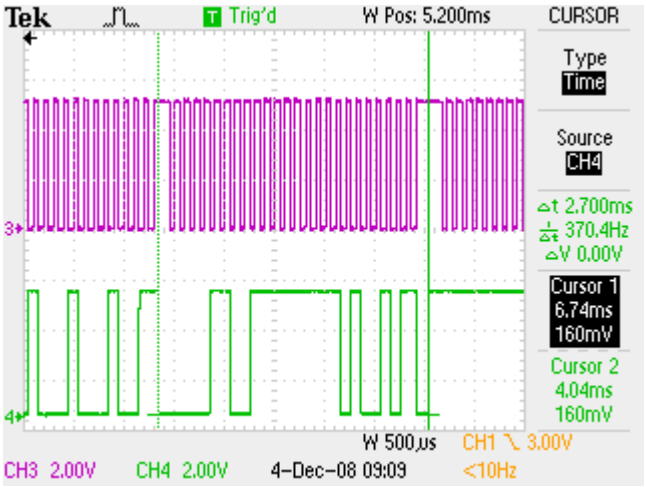


Figure 12: Command Frame

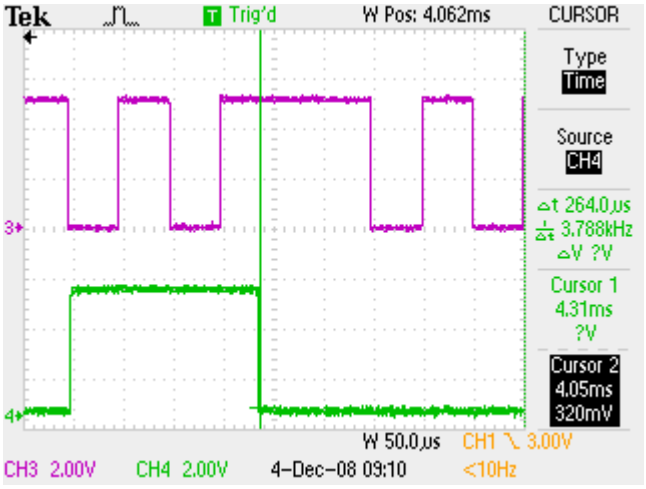


Figure 13: Start Bit Zoom

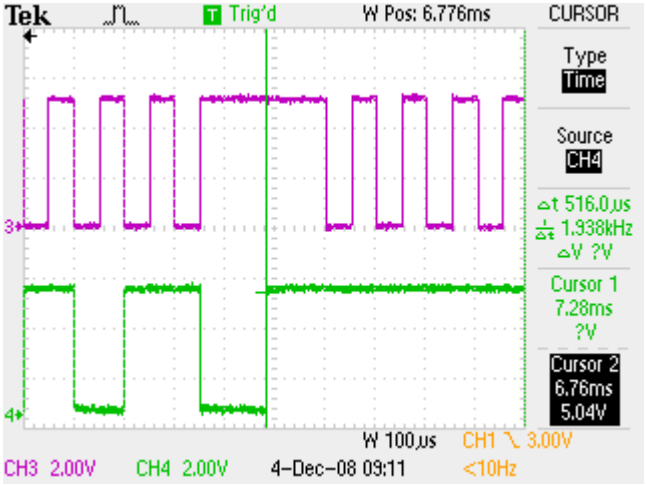


Figure 14: Stop Bit Zoom

The last byte of the command must load the RLEN register with a value of 9 to insure the last bit is clocked properly. This is necessary because the first edge on CLK (after it is started) is falling. The next rising edge will clock in the first bit of data. Since RLEN was loaded with 8, the interrupt will occur on the 8th falling edge. The WAITTO (CLK counter = RLEN) ISR will load the address byte to the TX FIFO, but it won't be placed on the I/O line until the next falling edge on CLK. The last bit of the command byte is placed on the I/O line when the interrupt is generated and is clocked on the rising edge of CLK. Figure 15 shows the start bit (left cursor) and the command byte of a command frame. The right cursor shows when the WAITTO is generated. The next rising edge on CLK is actually clocking the MSB of the command byte. In order not to miss clocking the last bit of the data byte, the last byte of a command frame must have RLEN set to 9.

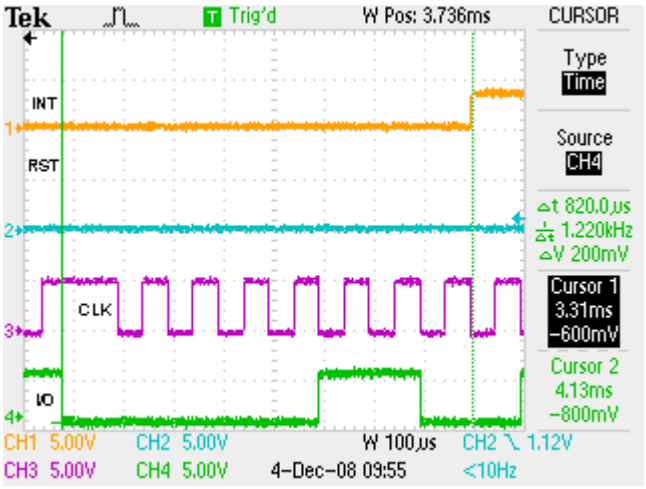


Figure 15: Start Bit and WAITTO

When reading the ATR response, setting RLEN to 9 on the last byte is not necessary, because RLEN is set to 1 to generate the card reset and then RLEN is loaded with 8 for the four bytes of the ATR. This sequence inherently provides the extra clock for the card reset. This is shown in Figure 16. The left cursor shows where the WAITTO is generated for RLEN = 1 and the right cursor shows where the interrupt is generated for RLEN = 8.

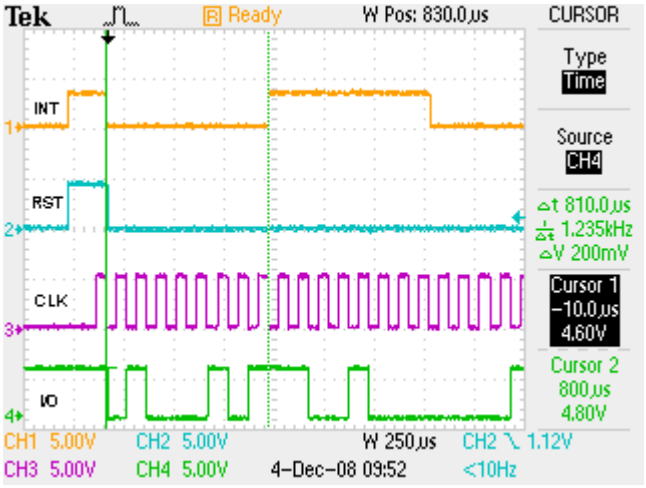


Figure 16: ATR Interrupt Timing

Figure 26 shows the send command flowchart.

3.2.1.3 Reading Data from the Card

When data is to be read from the card, the data must be clocked in after the stop bit. The first byte of the data being read must set the RLEN counter at 9. Since the CLK was stopped high after the stop bit, the first falling edge on CLK will configure the RX FIFO. The RLEN counter is now at 8 and the data byte is ready to be read into the RX FIFO. The subsequent bytes can set the RLEN at 8 during the ISR. After the last byte is received, the CLK should be stopped high to prepare for the next start bit at the beginning of the next command frame. The read data flowchart is shown in Figure 27.

3.2.2 3-Wire Card Operation

3.2.2.1 ATR Retrieval

The ATR retrieval for 3-wire cards is exactly the same as for 2-wire cards. See Section 3.2.1.1.

3.2.2.2 Sending Commands

3-wire card commands must follow the format as described in Section 2.2.2. Instead of using the start and stop bits as 2-wire cards, the 3-wire cards surround the command frame with the RST output set high. 3-wire card command frames are tricky with the 73S12xxF smart card logic as they expect the CLK to be stopped low and then start CLK so the first edge output is a rising edge. As explained previously, the transmit and receive FIFOs require a falling edge on CLK to be configured before a rising edge will clock the data properly. To work around this, the CLK should be high before starting the command frame. In this way, the CLK will start with a falling edge so the FIFOs are ready. A scope capture of the command frame is shown in Figure 17.

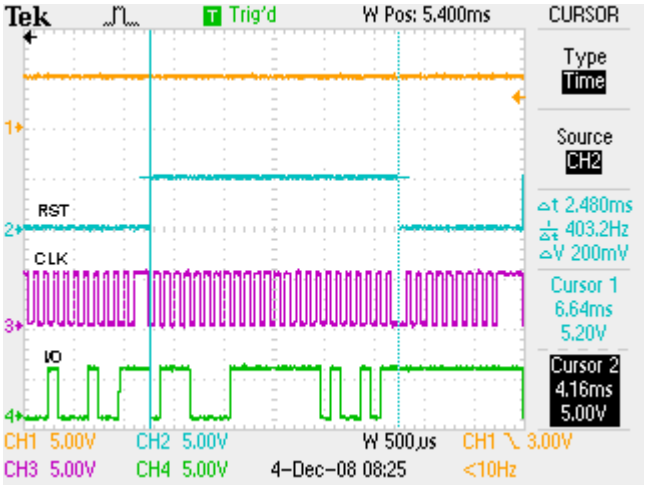


Figure 17: 3-Wire Command Frame

The proper timing of the 3-wire card has the CLK low when the RST is taken high. Unfortunately, to ensure the first edge output on CLK is a falling one, the RST can't be taken high while CLK is high. To work around this, the CLK must be started and be allowed to go low before taking RST high to start the command frame. By setting RLEN at 1 before starting CLK, the first falling edge of CLK will generate an interrupt and the ISR can then set RST high. The ISR should then set RLEN to 7 and continue. The WAITTO interrupt timing is shown in Figure 18. The Left cursor shown the interrupt when RLEN = 1 and the right cursor shows the interrupt when RLEN = 7. Figure 19 shows a zoom in of the CLK falling edge and the RST being set soon after by the ISR. When the WAITTO interrupt occurs, the command byte has been sent and the address byte should be written to the TX FIFO. From this point on, the RLEN configuration is identical to the 2-wire cards, where the address byte is sent to the TX FIFO with RLEN set at 8 and the data byte (last byte of the command frame) is sent with RLEN set to 9.

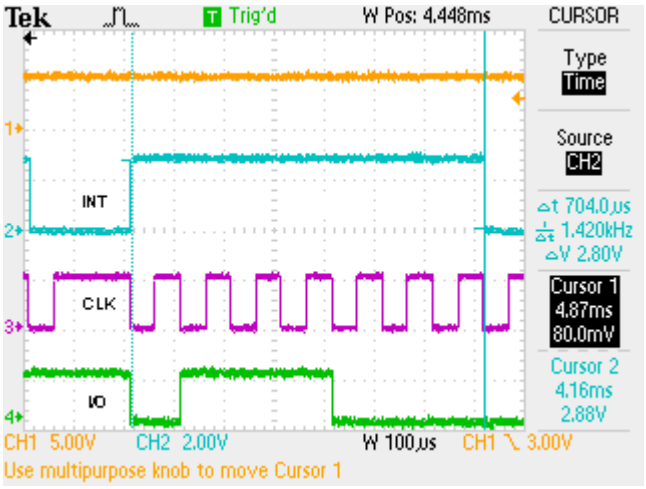


Figure 18: WAITTO Interrupt Timing on the Command Byte

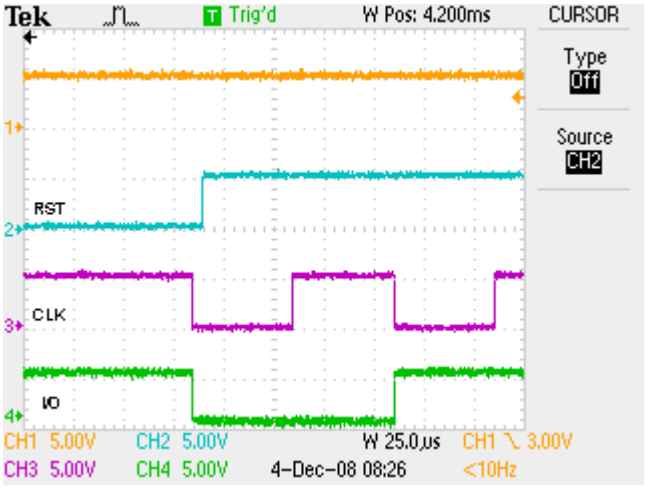


Figure 19: Zoom in on the Start of the Command Frame

After the command frame is complete, the CLK should be stopped low. This is shown in Figure 20. The flowchart for sending command frames is shown in Figure 28.

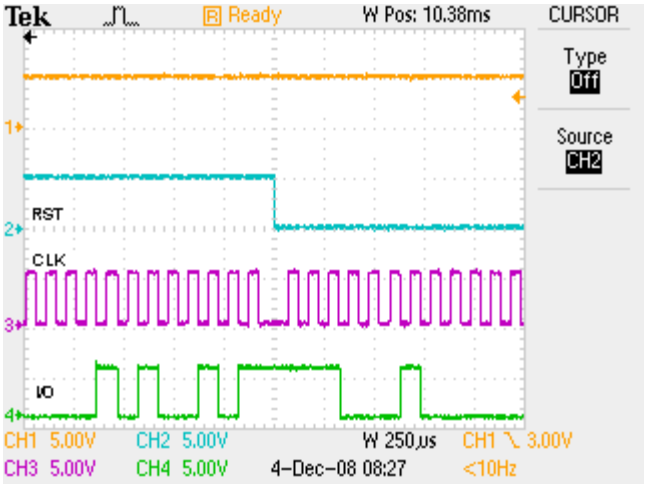


Figure 20: Zoom in on the End of the Command Frame

3.2.2.3 Reading Data from the Card

Reading data from a 3-wire card is a bit unusual. The CLK is low when the reading process begins. When CLK starts, the first clock pulse is ignored. Refer to Figure 5 to see this. This follows the same logic as the smart card block and therefore doesn't require any special processing except reading the first byte must load RLEN with 9 and then 8 thereafter. The flowchart for reading data from a card is shown in Figure 29.

3.2.3 I²C Card Operation

As mentioned above, the I²C cards operate similar to the 2-wire cards. They require the use of start and stop bits to begin and end transactions. They differ in that they do not use specific commands or return an ATR and require ACK and NAK bits as part of the protocol. The 73S12xxF smart card logic has provisions for I²C operation and contains a control bit (I2CMODE in the STXCTL register) to configure the smart card logic for I²C cards. By setting this control bit, the logic is configured to switch the I/O data from the FIFOs to the I/O bit in the SCCtl (or SCECtl for the external interface) register when RLEN is reached or is set to 0. When the interrupt is generated, the firmware must control the ACK bit manually through I/O and IOD (I/O direction control) bits.

When sending data to the card every byte must be followed with a valid ACK from the card. The card will set the ACK bit low after the last bit is sent. This occurs on the 9th falling edge of CLK. As a result, RLEN should be set to 9 for all bytes read and written from/to a card. Figure 21 shows a transaction start with device address and memory address. The left cursor shows the start bit. The right cursor shows the ACK bit returned from the card. Notice that the bit preceding the ACK bit (R/W bit) is low so the next two bytes must be the memory address. When a WAITTO interrupt is generated, the firmware must set the IOD bit for input, write the next byte to the TX FIFO, set RLEN to 9 and set the IOD bit low so the ACK bit can be read. The ACK bit should be sampled in the middle of the ACK bit so it should be sampled ½ ETU after the interrupt. A proper ACK is low, so if the ACK bit is high, there is an error and the proper error recovery algorithm should be executed. The CLK has not been stopped so the next falling edge of CLK will automatically switch the I/O path back to the FIFO since the value of RLEN is no longer at the max value or 0 (since it was reloaded). The setting of the IOD bit is overridden in the hardware so no reconfiguration is required. At the beginning of the frame, the IOD bit is set for output to properly configure the start bit. The IOD bit is overridden by hardware when the CLK is running and the value of the RLEN counter is not at the limit or 0. At the end of a card write operation, the CLK should be stopped high and the IOD bit should be set for output and I/O set low. A minimum 1 ETU delay should be allowed for the CLK output to stop high and then the I/O should be taken high to create the stop bit. Figure 22 shows a capture of the stop bit. The left cursor shows the last ACK and the right cursor shows the stop bit. The I²C write data flow chart is shown in Figure 31.

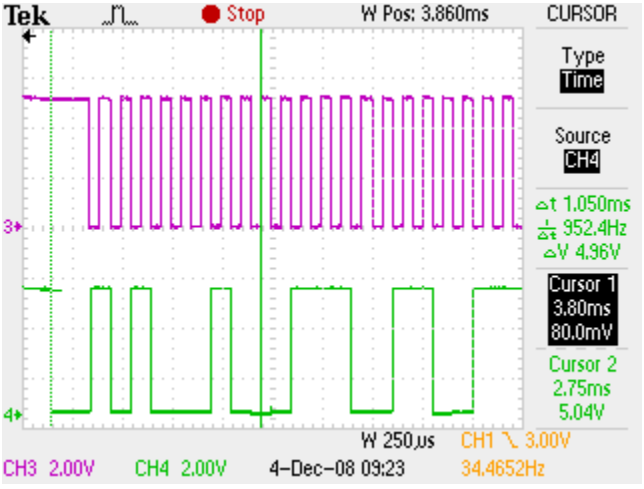


Figure 21: I²C Transaction Start with Start Bit, Device Address and Memory Address

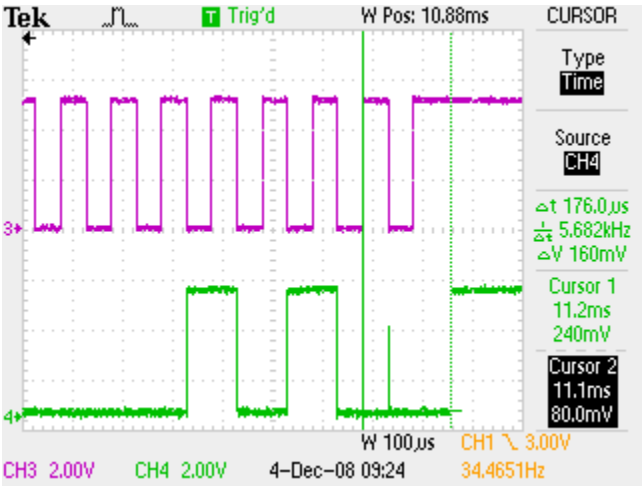


Figure 22: I²C Transaction End with Stop Bit

When reading data from the card, the device address has to be written to the card with or without the memory address. If the R/W bit is set high, then the next byte after the ACK (designated as ACK_W) will be the first bit of the read data. If the R/W bit is low, then the memory address will follow the device address. At the end of this frame and after the ACK bit, the CLK and the I/O should be stopped high to prepare for the start bit that must precede any card read where the memory address is specified. The device address must be sent again. This time with the R/W bit set high (for read). At this point, the data can be read from the card.

Before reading data from the card, the ACK bit output (that follows the read data byte that will be designated as ACK_R) should be configured by setting the IOD for output and the I/O bit low. Waiting for the WAITTO may be too late as the ISR has to be serviced and then the ACK_R bit has to be set. To ensure that the ACK_R bit is output to the card immediately after the last data bit is read, the IOD and I/O bits should be configured before the WAITTO interrupt is generated. Setting this during the ACK_W bit preceding the reading of the first bit of the read data stream may be too early. To insure all the ACKs are properly configured, the first read ACK_R bit should be configured somewhere in between these times. This is easily accomplished by waiting at least 1 ETU after the WAITTO interrupt for the ACK_W input that precedes the first data read. By waiting for at least 1 ETU, the last ACK_W bit is allowed to fully complete before configuring the IOD and IO bits. When the WAITTO ISR is serviced, RLEN should be set for 9 which will automatically connect the I/O to the RX FIFO on the next falling edge of CLK. Waiting at least 1 ETU insures that this edge is output before configuring the IOD and IO bits. This method insures the ACK bits are not during its bit time and the first ACK_R is clean.

Figure 10 shows the proper format for an I²C read operation. It begins with a start bit followed by the device address with the R/W bit set low to indicate that the memory address is to be written. Figure 23 shows an example of a read operation. The left cursor shows the start bit followed by the device address and a memory address of 0. After the memory address is written, another start bit is generated (note: no stop was generated at the end of the memory address). The right cursor of Figure 23 shows this start bit. The device address is sent again, this time with the R/W bit set high to indicate that the data can now be read from the card. After the ACK bit is verified, the smart card block is configured to receive data and set RLEN to 9. Up to this point the operation of the ACK bit is the same as for the write operations.

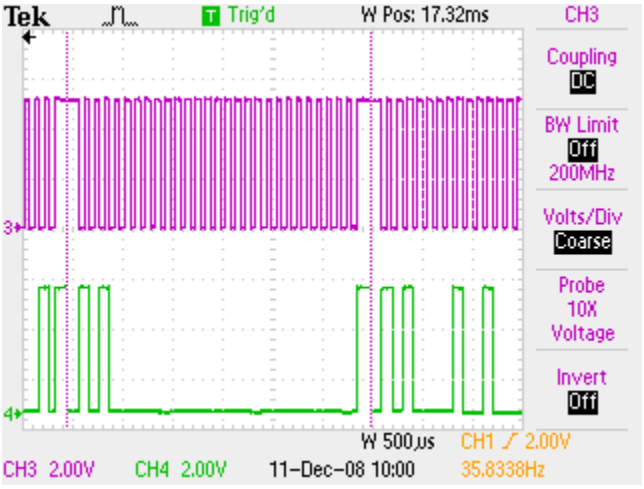


Figure 23: I²C Data Read

The read operations can read continuous consecutive memory locations as long as the ACK from the reader is low. The IO and IOD bits need to be set low after this ACK bit completes (device address with R/W set high). This configures the ACK bit to output a logic zero for the next byte read after the WAITTO interrupt is generated. The ISR will read the byte and set the RLEN to 9 to read the next byte. To signal the card that the reader wants to stop reading, the last ACK must be set high. During the ISR, if the next byte is to be the last one read, the IOD bit must be set for output and the I/O bit must be set high to generate the NAK after the previous ACK bit time is complete. After the next WAITTO interrupt, the ACK bit will output a logic one (NAK) to signal the card that the read operation is complete. The CLK should be stopped high with the IOD and I/O bits set low after the NAK bit time completes to prepare the stop bit. The I/O bit should then be set high to output the stop bit. The read operation is now complete. Figure 24 shows an example of the end of a read operation. The left cursor marks the next to last ACK bit and the right cursor shows the last NAK bit. The stop bit is shown after the NAK bit.

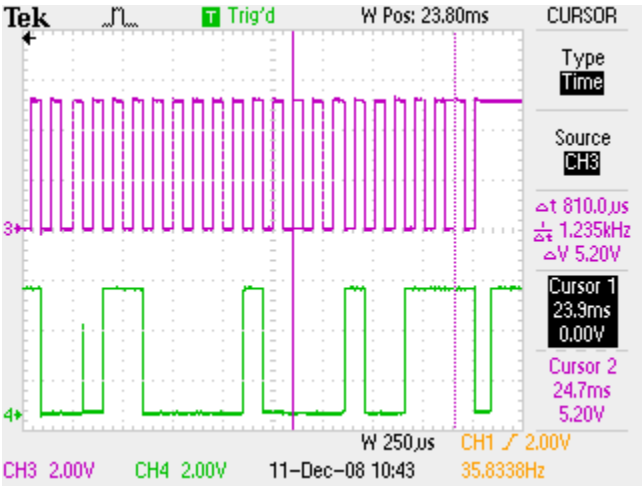


Figure 24 : I²C Read Operation Termination

4 Related Documentation

The following 73S12xxF documents are available from Teridian Semiconductor Corporation:

71S1210F Data Sheet
71S1209F Data Sheet
71S1215F Data Sheet
71S1217F Data Sheet

5 Contact Information

For more information about Teridian Semiconductor products or to check the availability of the 73S12xxF, contact us at:

6440 Oak Canyon Road
Suite 100
Irvine, CA 92618-5201

Telephone: (714) 508-8800
FAX: (714) 508-8878
Email: scr.support@teridian.com

For a complete list of worldwide sales offices, go to http://www.teridian.com.

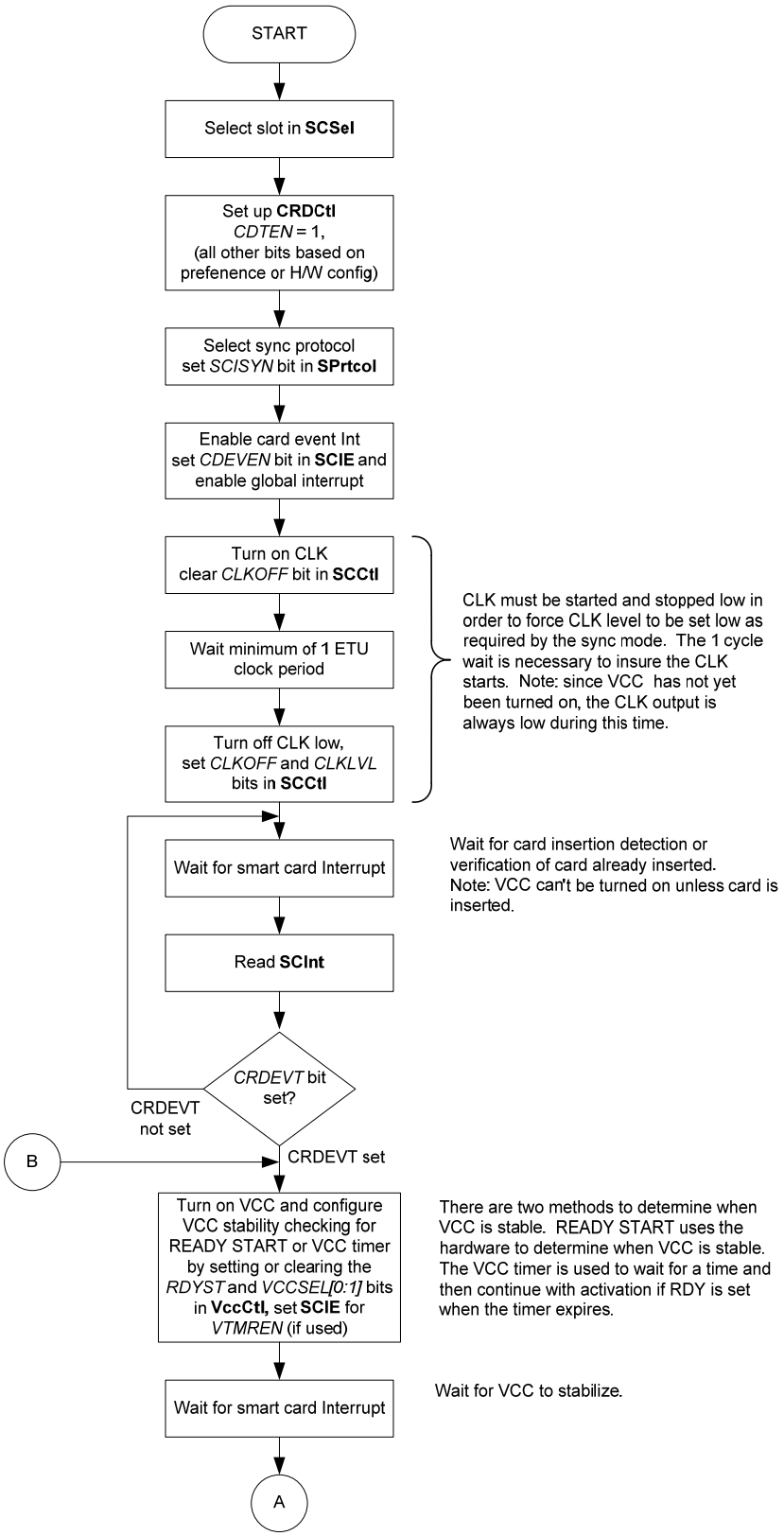
Appendix A

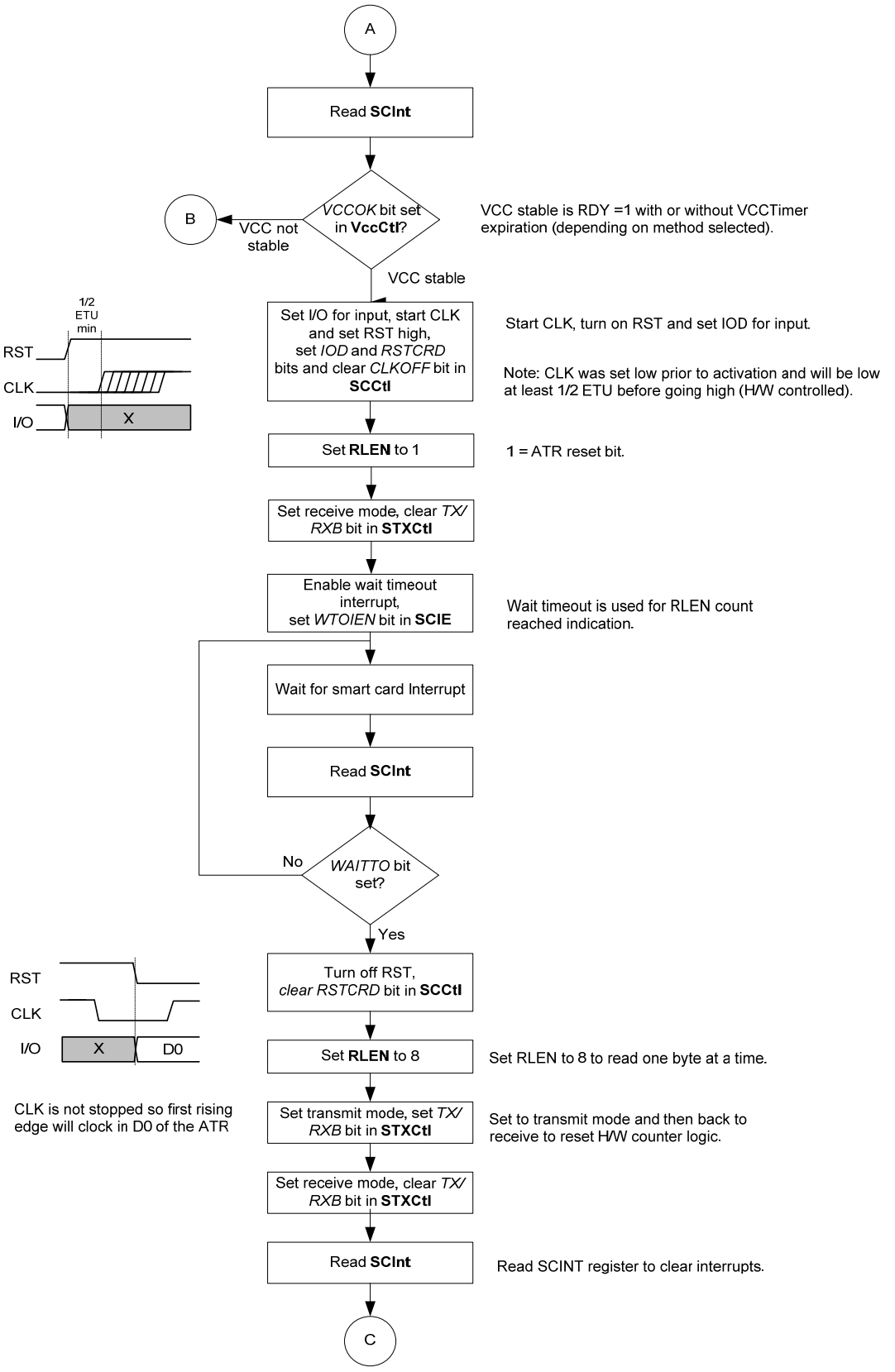
The following figures show the flowcharts for each type of synchronous card.

Bold typeface denotes a register name. *Italic* typeface denotes a register bit name.

Note: Only changes to registers are indicated. Other bits remain unchanged.

A.1 2-Wire Card Activation and ATR Retrieval




A
Read SCInt
B
VCC not stable
VCCOK bit set in VccCtl?
VCC stable is RDY =1 with or without VCCTimer expiration (depending on method selected).
VCC stable
1/2 ETU min
RST
CLK
I/O
X
Set I/O for input, start CLK and set RST high, set IOD and RSTCRD bits and clear CLKOFF bit in SCCtl
Start CLK, turn on RST and set IOD for input.
Note: CLK was set low prior to activation and will be low at least 1/2 ETU before going high (H/W controlled).
Set RLEN to 1
1 = ATR reset bit.
Set receive mode, clear TX/RXB bit in STXCtl
Enable wait timeout interrupt, set WTOIEN bit in SCIE
Wait timeout is used for RLEN count reached indication.
Wait for smart card Interrupt
Read SCInt
No
WAITTO bit set?
Yes
RST
CLK
I/O
X
D0
Turn off RST, clear RSTCRD bit in SCCtl
Set RLEN to 8
Set RLEN to 8 to read one byte at a time.
CLK is not stopped so first rising edge will clock in D0 of the ATR
Set transmit mode, set TX/RXB bit in STXCtl
Set to transmit mode and then back to receive to reset H/W counter logic.
Set receive mode, clear TX/RXB bit in STXCtl
Read SCInt
Read SCINT register to clear interrupts.
C

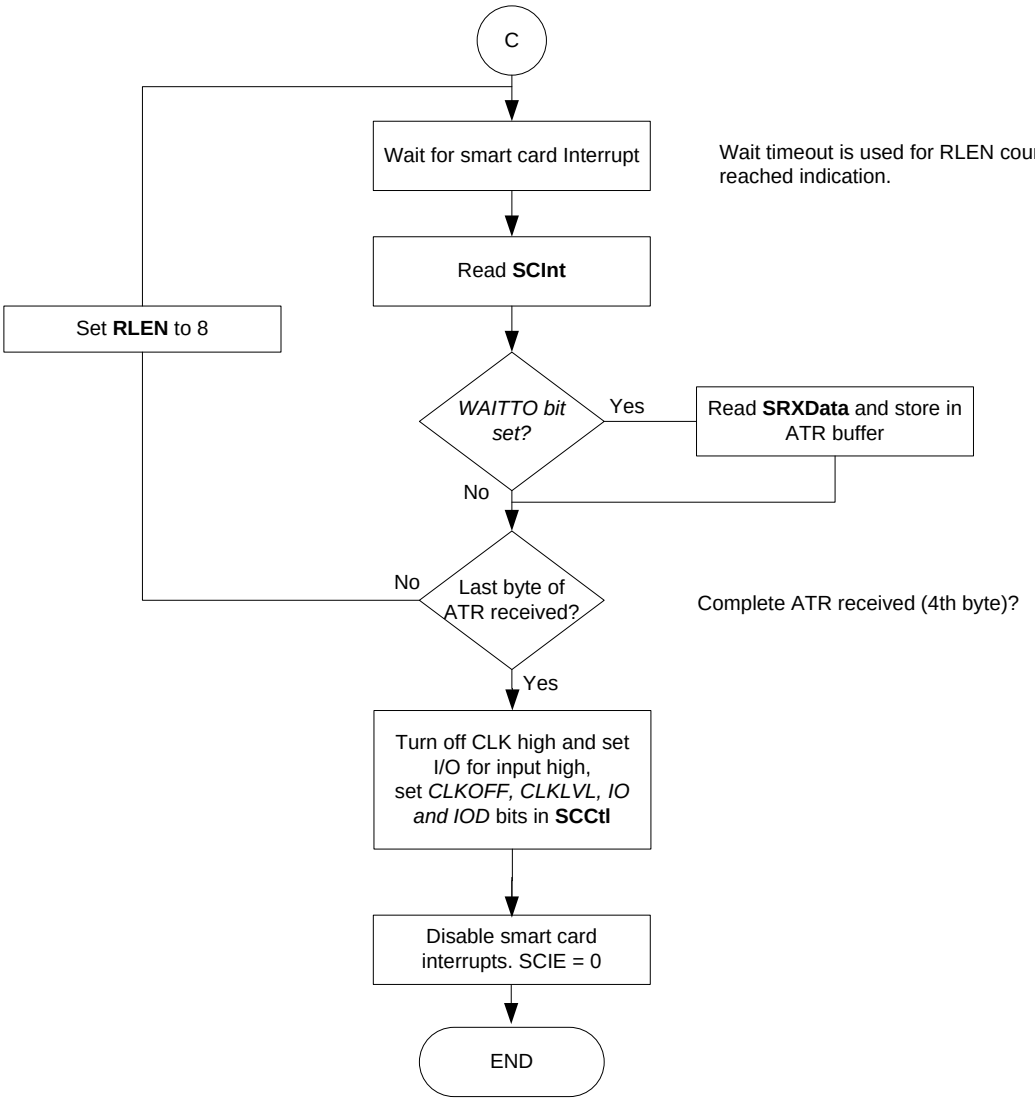
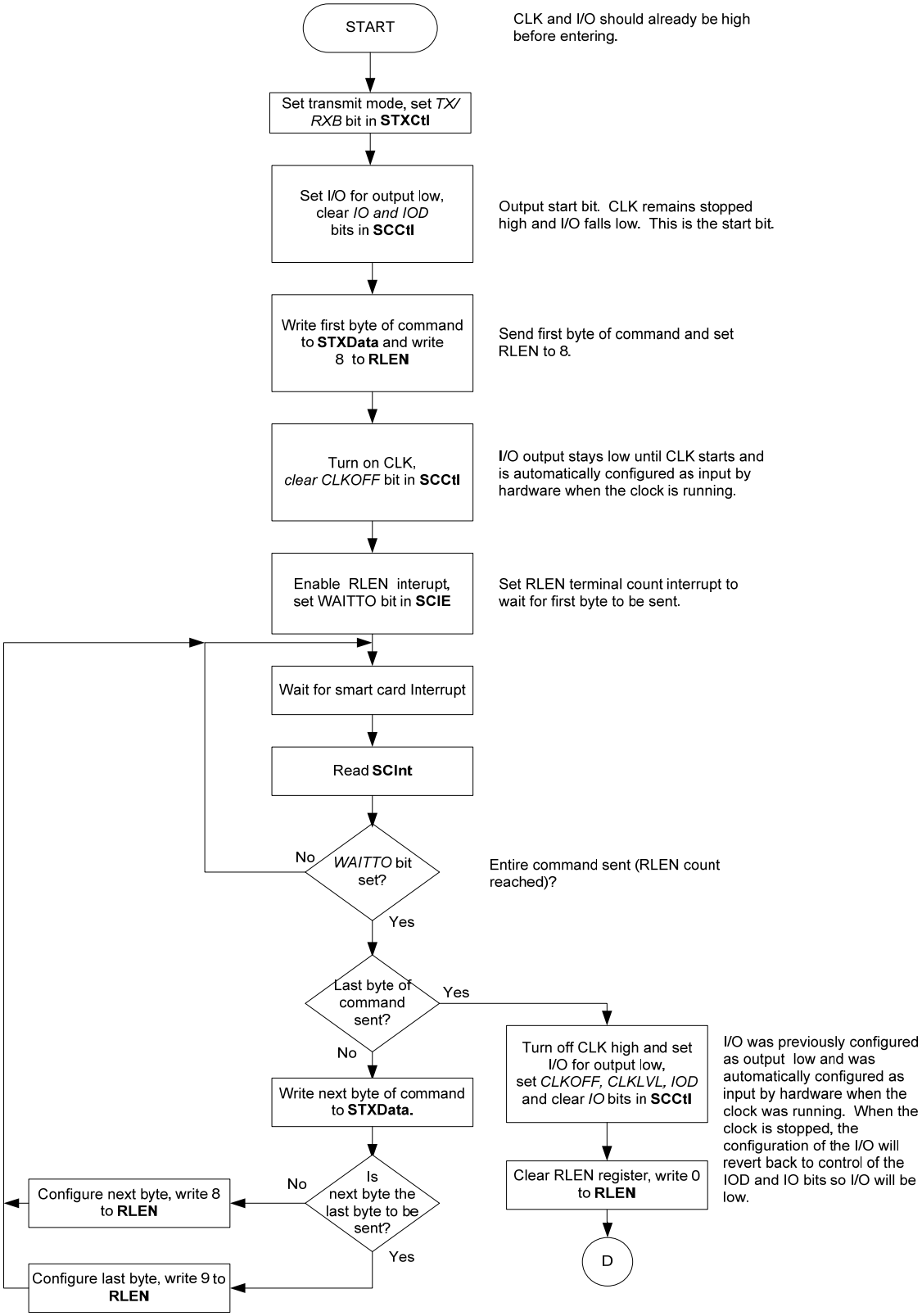


Figure 25: 2-Wire Card Activation and ATR Retrieval Flowchart

A.2 2-Wire Send Command to Sync Card with Start and Stop Bits



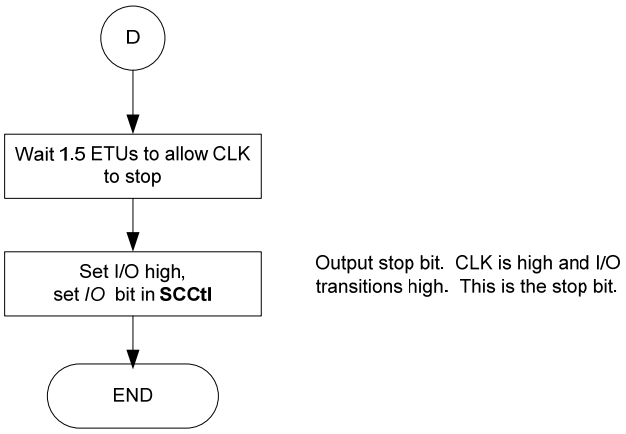


Figure 26: 2-Wire Send Command Flowchart

A.3 2-Wire Read Data from Sync Card

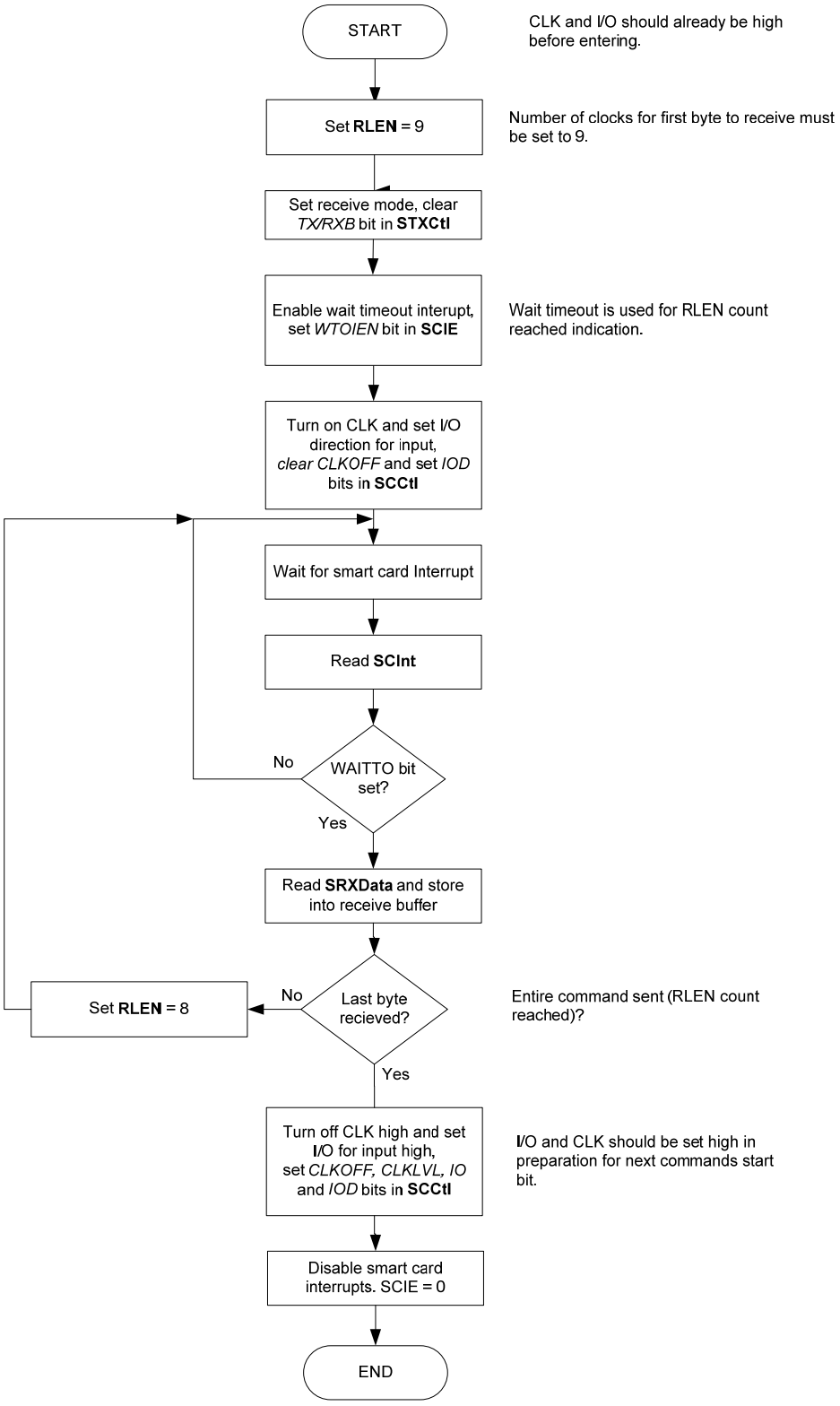
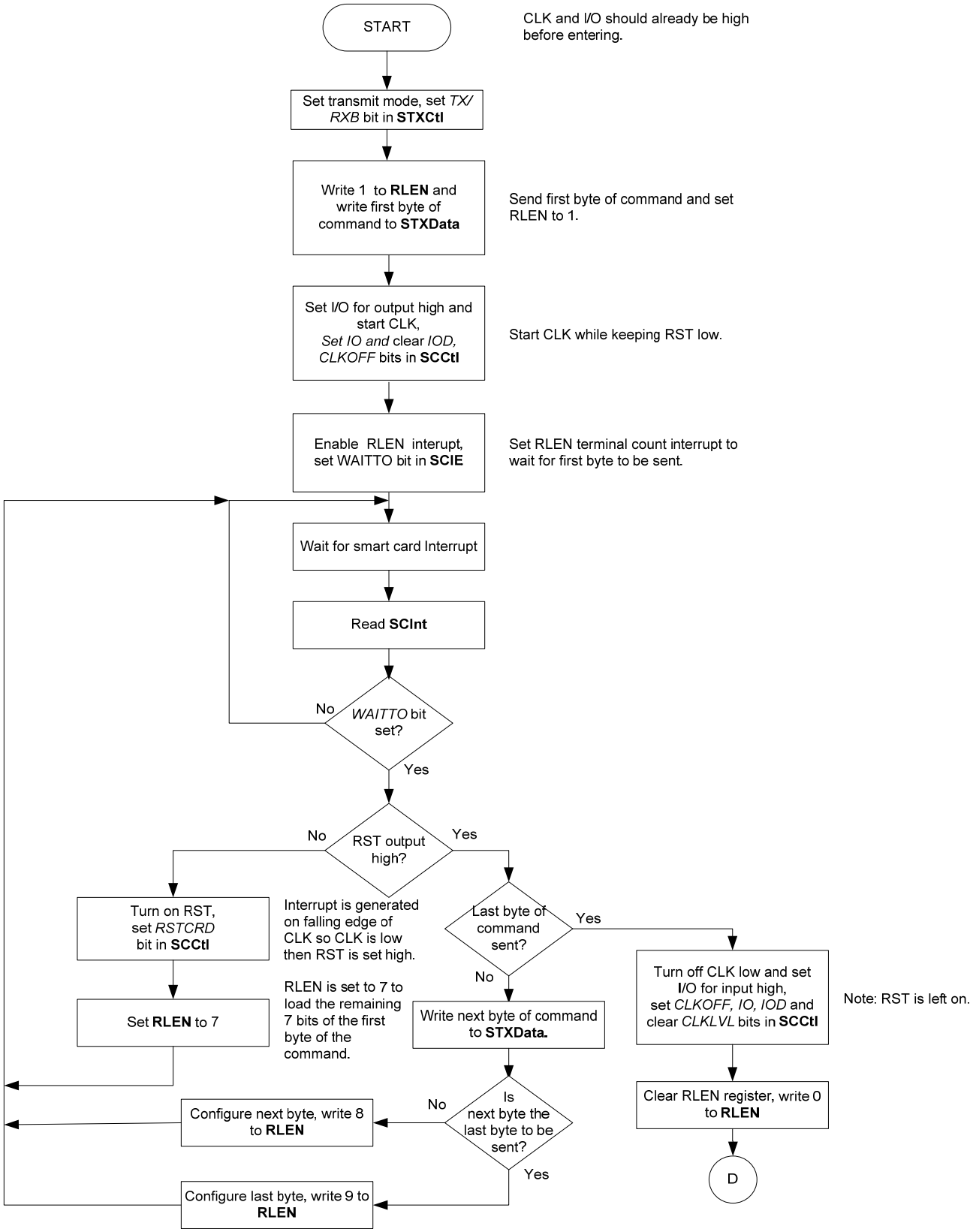


Figure 27: 2-Wire Read Data Flowchart

A.4 3-Wire Send Command to Sync Card with Start and Stop Bits



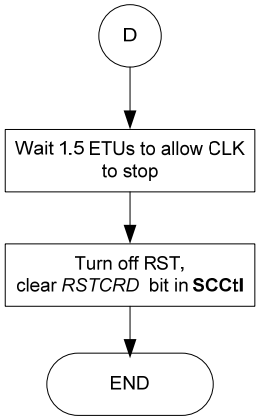


Figure 28: 3-Wire Send Command Flowchart

A.5 3-Wire Read Data from Sync Card

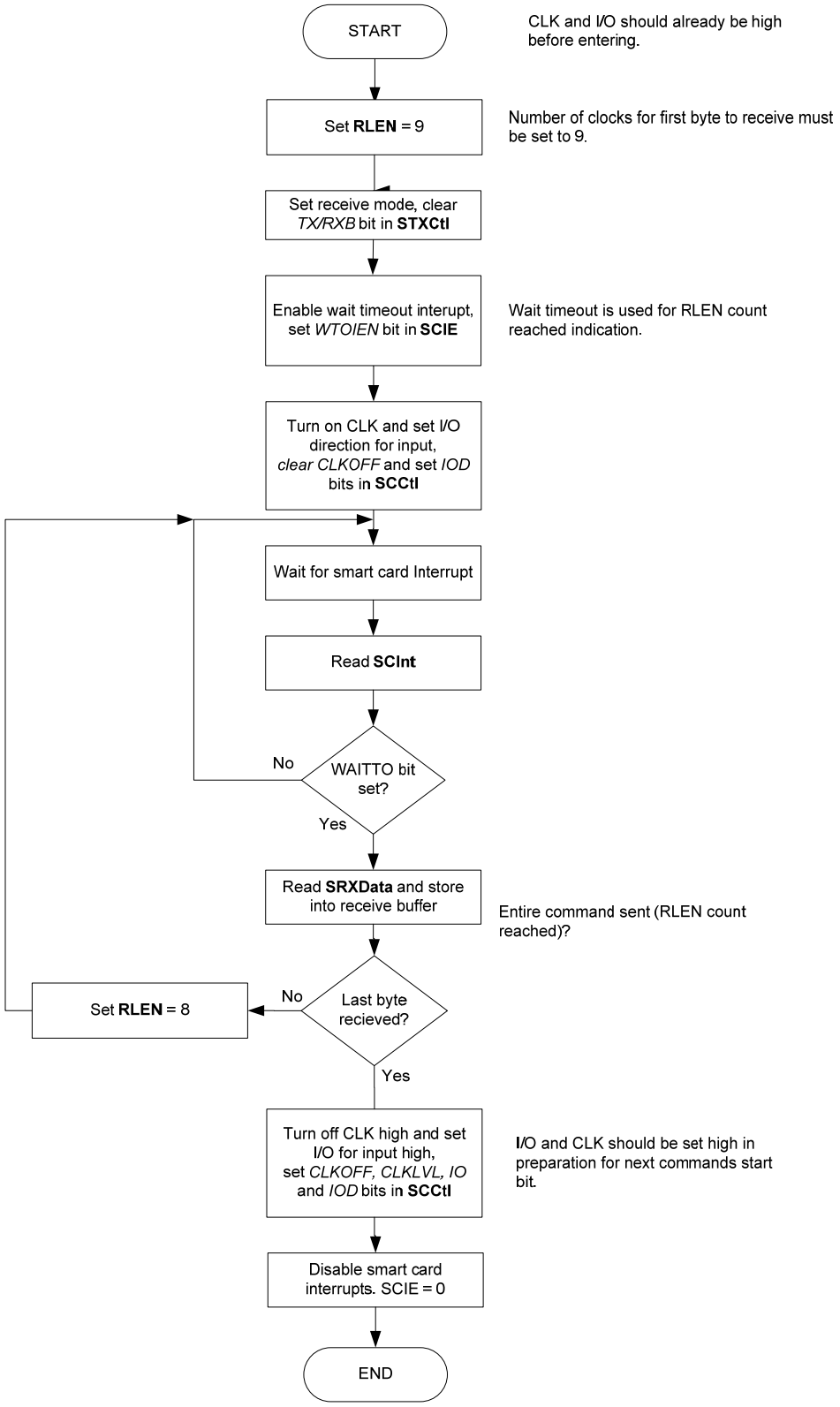
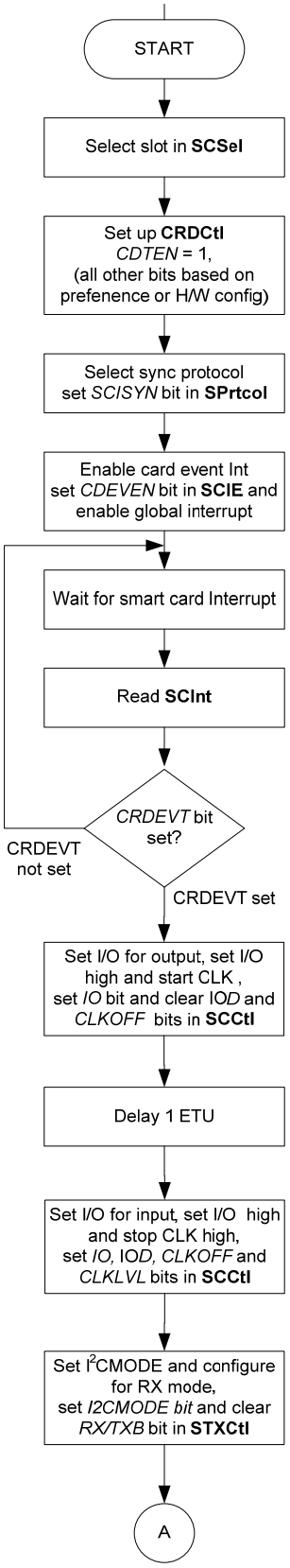


Figure 29: 3-Wire Read Data Flowchart

A.6 Activate 1²C Type of Smart Card



Wait for card insertion detection or verification of card already inserted. Note: VCC can't be turned on unless card is inserted.

The CLK signal output logic is normally low upon power on reset. A valid I2C start bit requires that the CLK signal is high. The CLK output can not be set to a certain level directly. The clock must be allowed to run for a full cycle (min 1 ETU) and then stopped at the desired level. As a result, the CLK circuit should be started and set high prior to turning on power (Vcc) to the card. If Vcc is off, then the single clock pulse will not be output on the CLK output. When Vcc is turned on, the CLK output will start high.

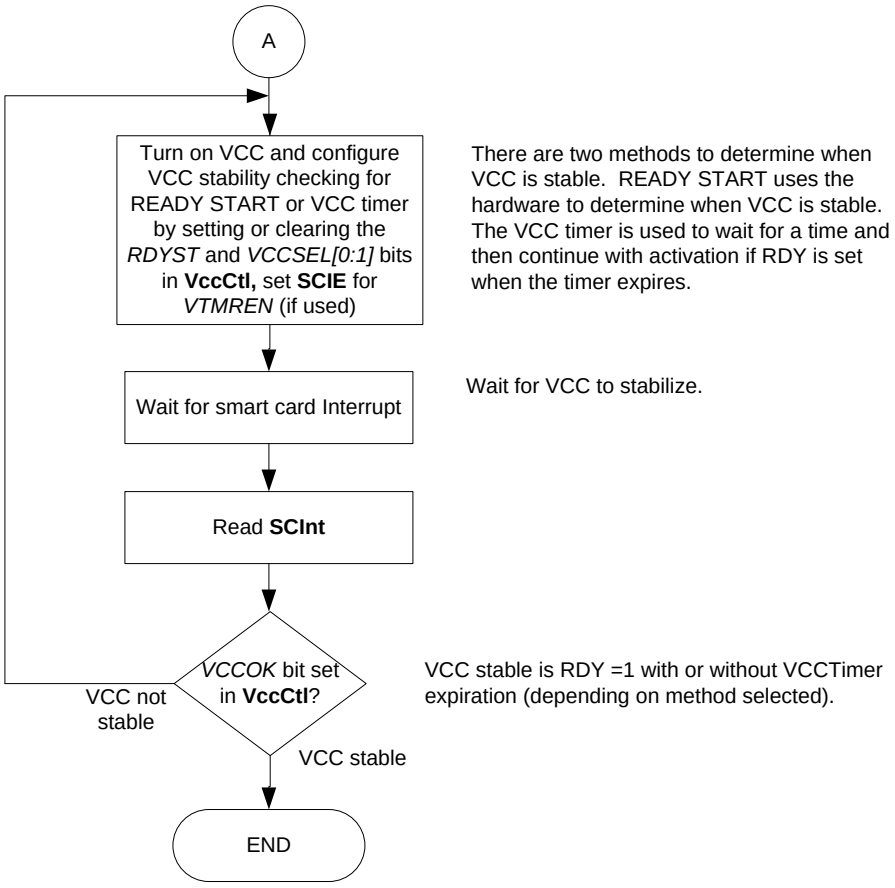
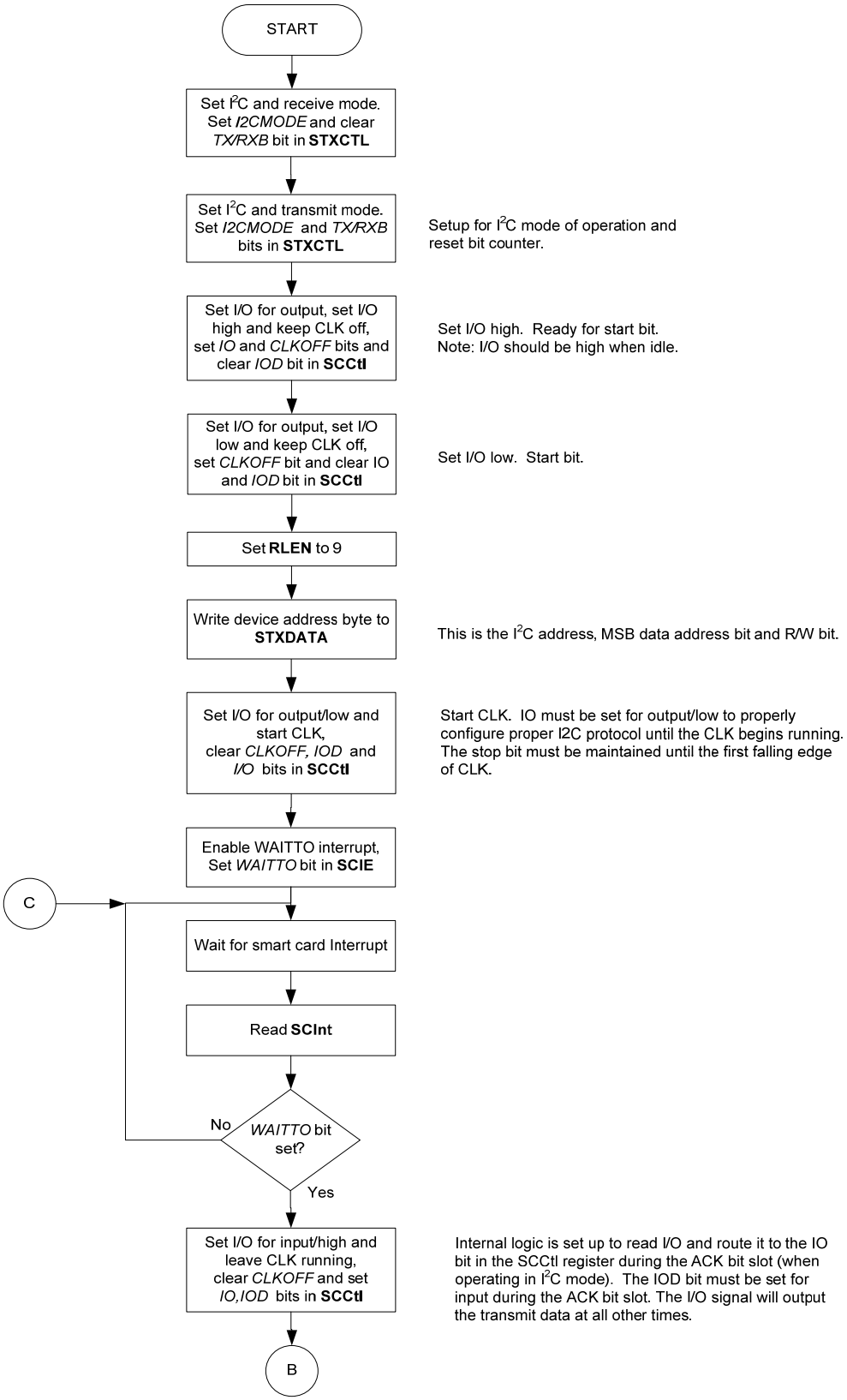


Figure 30: I²C Card Activation Flowchart

A.7 Perform Page Write to 1²C Mode Sync Card



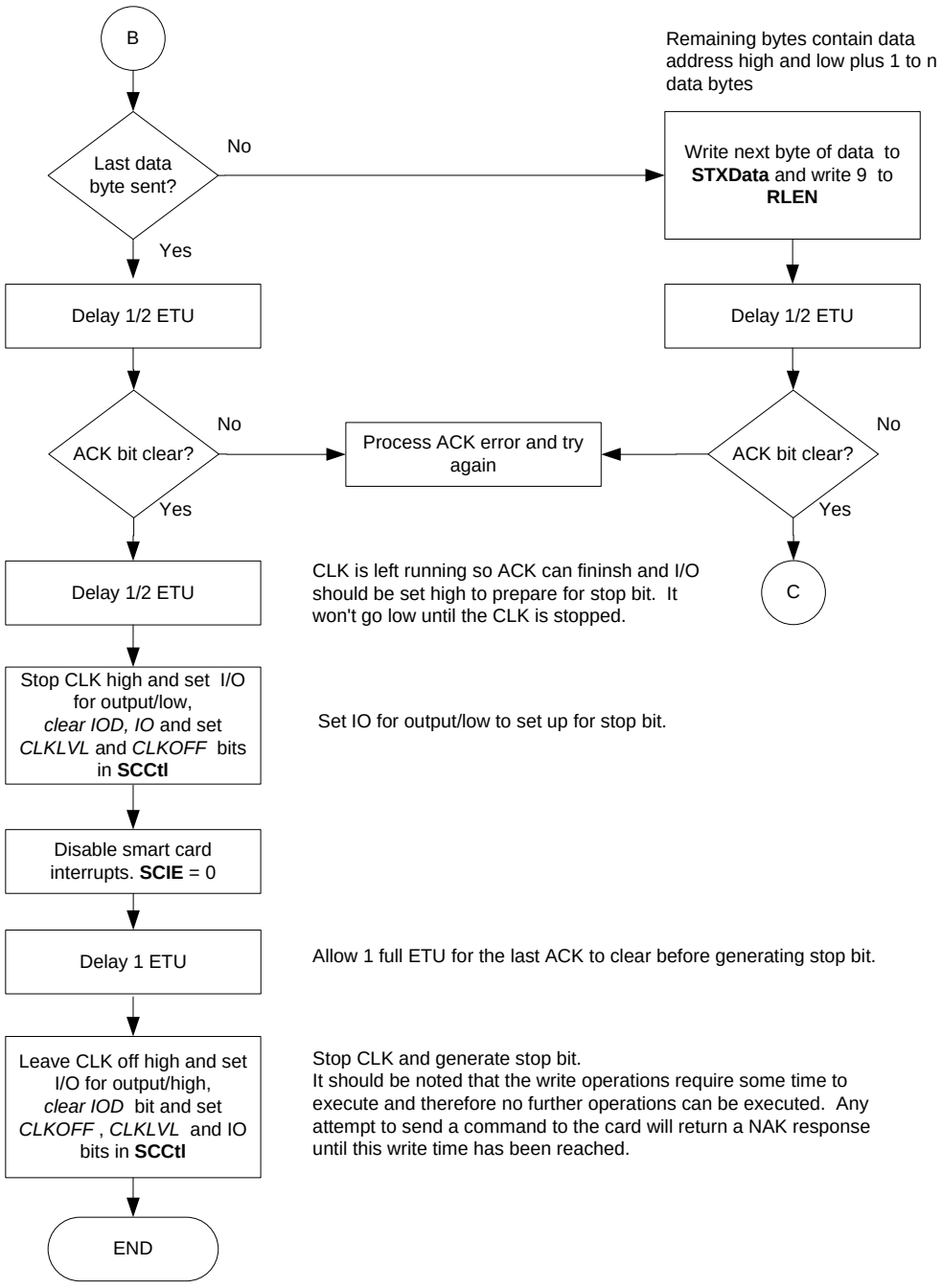
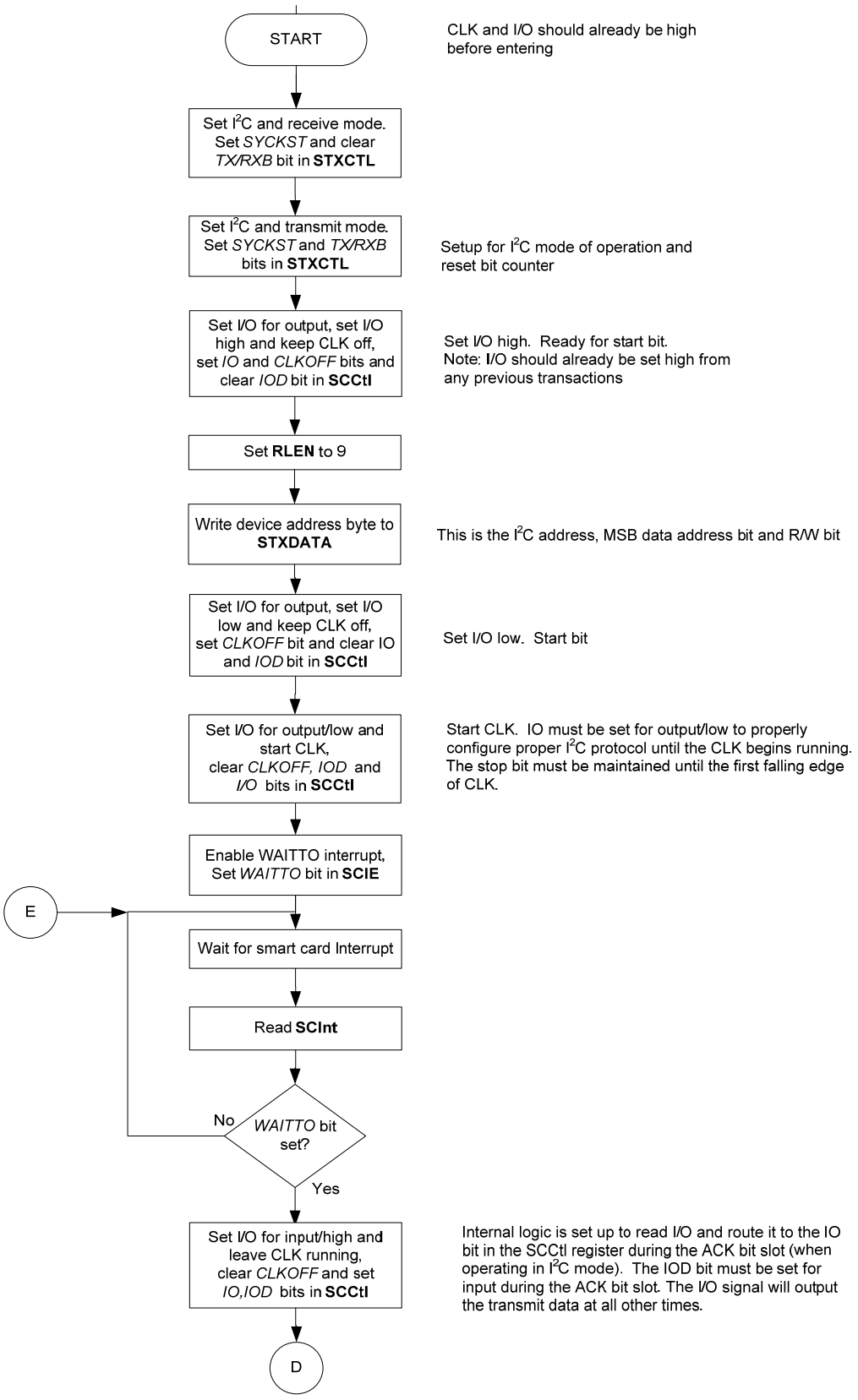
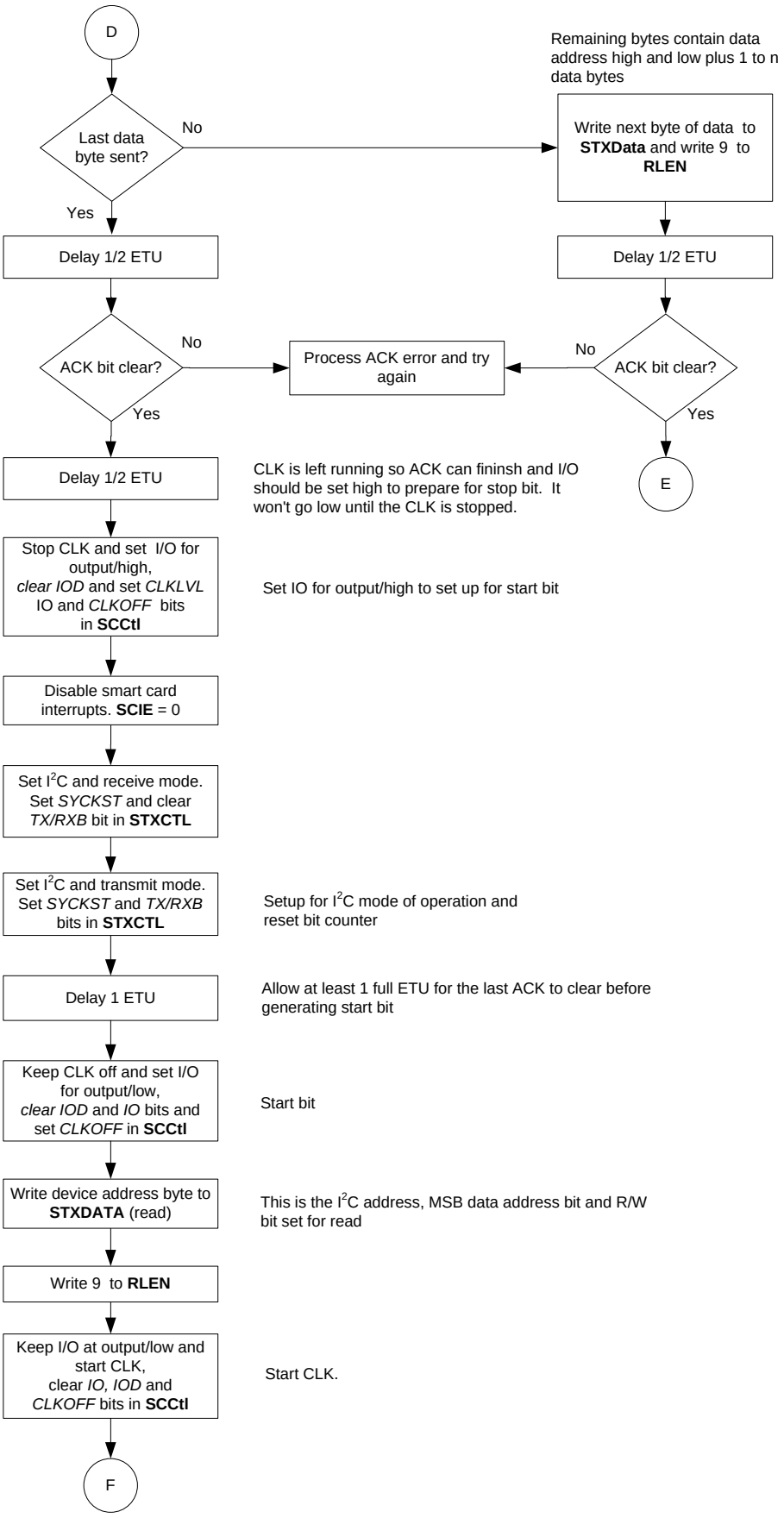
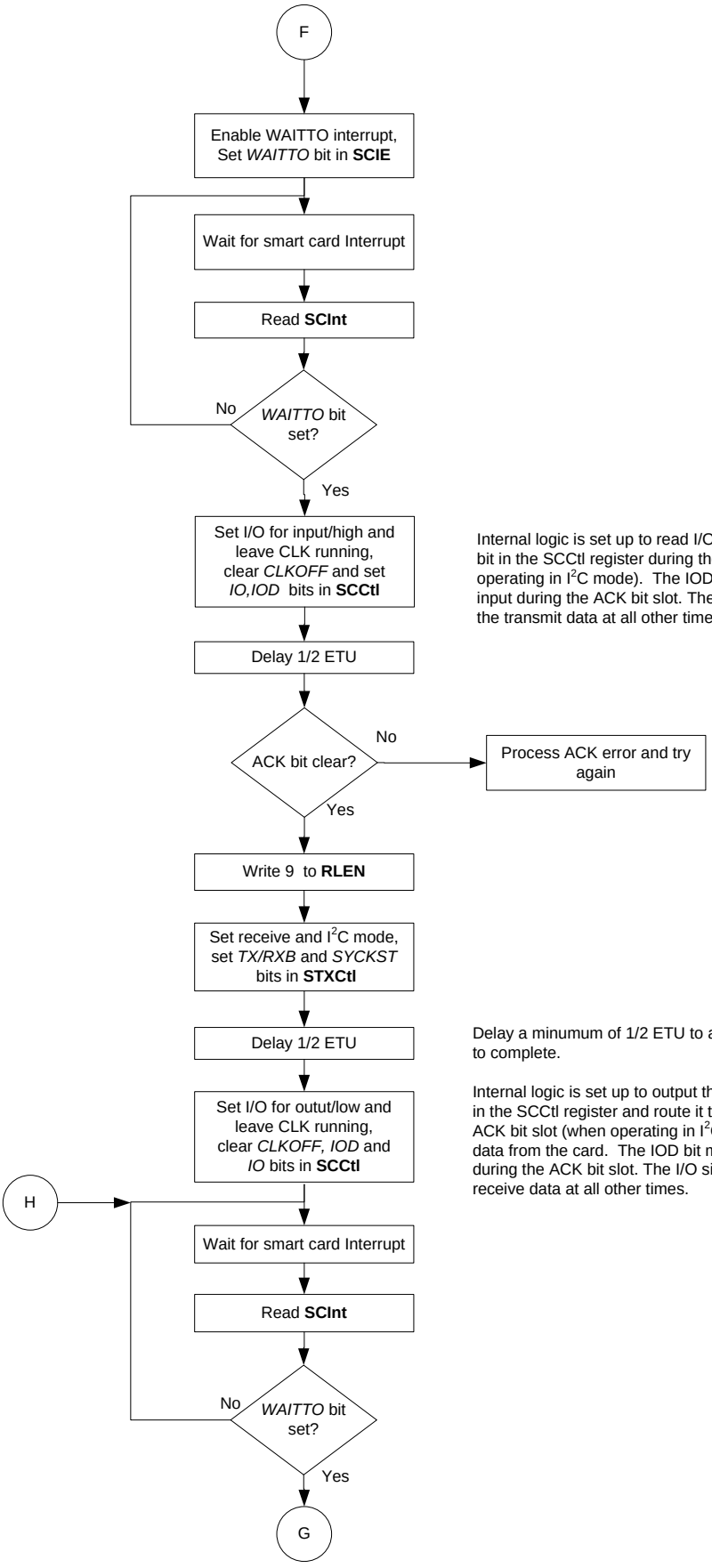


Figure 31: I²C Write Data Flowchart

A.8 Read Data from 1^{2}C Type Sync Card




D
Last data byte sent?
No
Yes
Remaining bytes contain data address high and low plus 1 to n data bytes
Write next byte of data to **STXData** and write 9 to **RLEN**
Delay 1/2 ETU
Delay 1/2 ETU
ACK bit clear?
No
Yes
Process ACK error and try again
No
ACK bit clear?
Yes
E
Delay 1/2 ETU
CLK is left running so ACK can fininsh and I/O should be set high to prepare for stop bit. It won't go low until the CLK is stopped.
Stop CLK and set I/O for output/high, *clear IOD* and set *CLKLVL* IO and *CLKOFF* bits in **SCCtl**
Set IO for output/high to set up for start bit
Disable smart card interrupts. **SCIE** = 0
Set I2C and receive mode. Set *SYCKST* and clear *TX/RXB* bit in **STXCTL**
Set I2C and transmit mode. Set *SYCKST* and *TX/RXB* bits in **STXCTL**
Setup for I2C mode of operation and reset bit counter
Delay 1 ETU
Allow at least 1 full ETU for the last ACK to clear before generating start bit
Keep CLK off and set I/O for output/low, *clear IOD* and *IO* bits and set *CLKOFF* in **SCCtl**
Start bit
Write device address byte to **STXDATA** (read)
This is the I2C address, MSB data address bit and R/W bit set for read
Write 9 to **RLEN**
Keep I/O at output/low and start CLK, clear *IO, IOD* and *CLKOFF* bits in **SCCtl**
Start CLK.
F



Internal logic is set up to read I/O and route it to the IO bit in the SCCtl register during the ACK bit slot (when operating in I^2C mode). The IOD bit must be set for input during the ACK bit slot. The I/O signal will output the transmit data at all other times.

Delay a minumum of 1/2 ETU to allow ACK bit slot to complete.

Internal logic is set up to output the contents of the I/O bit in the SCCtl register and route it to the IO pin during the ACK bit slot (when operating in I^2C mode) when reading data from the card. The IOD bit must be set for output during the ACK bit slot. The I/O signal will input the receive data at all other times.

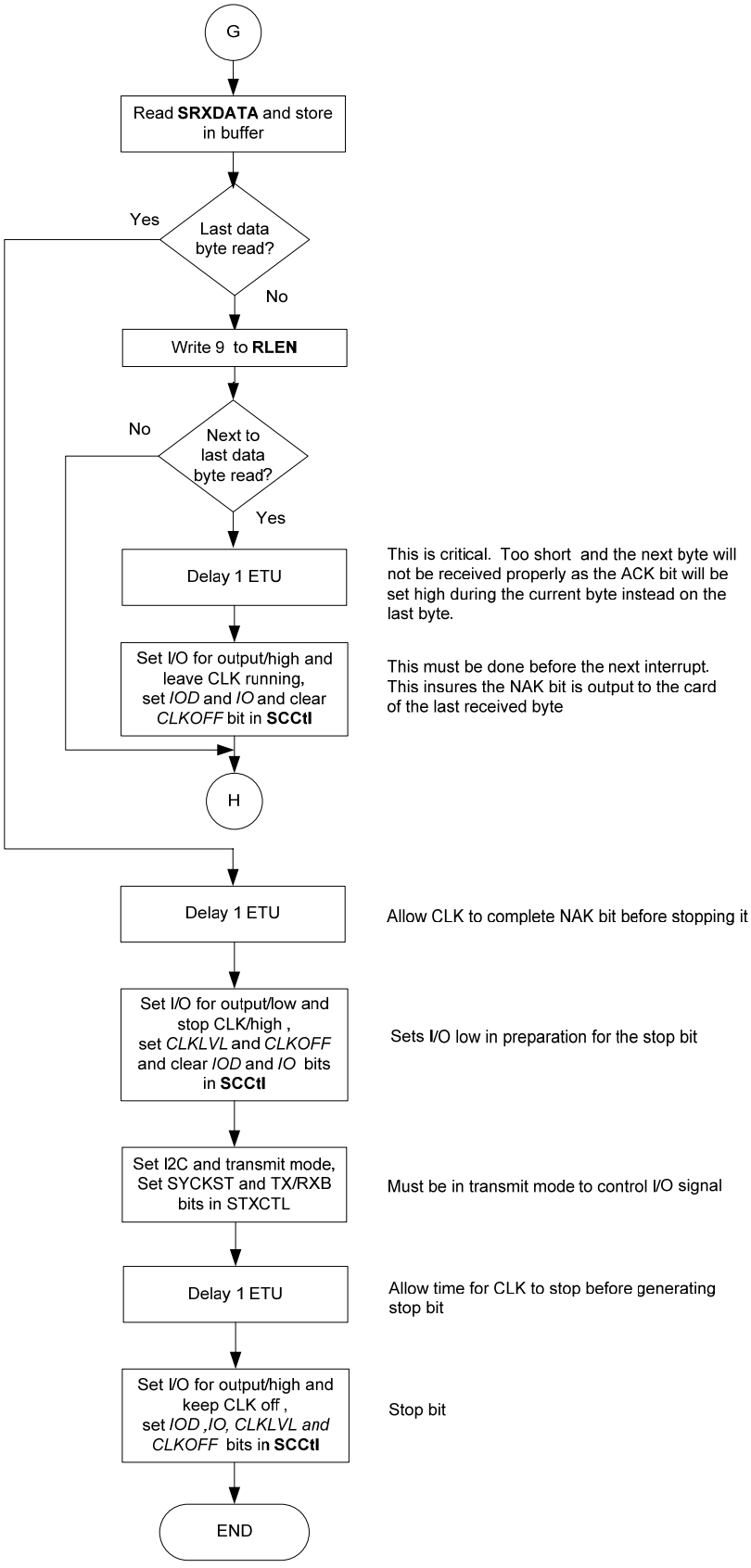


Figure 32: I²C Read Data Flowchart

Revision History

Revision	Date	Description
1.0	12/18/2008	First publication.