Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

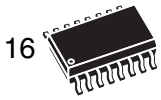
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

CRX14

ISO14443 type-B contactless coupler chip with anti-collision, CRC management and anti-clone function

Features

- Single 5 V ±500 mV supply voltage
- SO16N package
- Contactless communication
  - ISO14443 type-B protocol
  - 13.56MHz carrier frequency using an external oscillator
  - 106 Kbit/s data rate
  - 36-byte input/output frame register
  - Supports frame answer with/without SOF/EOF
  - CRC generation and check
  - France Telecom proprietary anti-clone function
  - Automated ST anti-collision exchange
- I²C communication
  - Two-wire I²C serial interface
  - Supports 400 kHz protocol
  - 3 chip enable pins
  - Up to 8 CRX14 connected on the same bus



SO16 (MQ)
150 mils width

Contents

List of tables

List of figures

1 Summary description

The CRX14 is a contactless coupler that is compliant with the short range ISO14443 type-B standard. It is controlled using the two wire I²C bus.

The CRX14 generates a 13.56 MHz signal on an external antenna. Transmitted data are modulated using Amplitude Shift Keying (ASK). Received data are demodulated from the PICC (Proximity integrated Coupling Card) load variation signal, induced on the antenna, using Bit Phase Shift Keying (BPSK) of a 847kHz sub-carrier. The Transmitted ASK wave is 10% modulated. The Data transfer rate between the CRX14 and the PICC is 106 Kbit/s in both transmission and reception modes.

The CRX14 follows the ISO14443 type-B recommendation for Radio frequency power and signal interface.

The CRX14 is specifically designed for short range applications that need disposable or secure and reusable, products.
The CRX14 includes an automated anti-collision mechanism that allows it to detect and select any ST short range memories that are present at the same time within its range. The anti-collision mechanism is based on the STMicroelectronics probabilistic scanning method.

The CRX14 provides an anti-clone function, from FRANCE TELECOM, which allows the authentication of the ST short range memories. Using the CRX14 single chip coupler, therefore, it is easy to design a reader, with authentication capability and to build an end application with a high level of security at low cost.

The CRX14 provides a complete analog interface, compliant with the ISO14443 type-B recommendations for Radio-Frequency power and signal interfacing. With it, any ISO14443 type-B PICC products can be powered and have their data transmission controlled via a simple antenna.

The CRX14 is fabricated in STMicroelectronics High Endurance Single Poly-silicon CMOS technology.

The CRX14 is organized as 4 different blocks (see *Figure 2*):

- The I²C bus controller. It handles the serial connection with the application host. It is compliant with the 400kHz I²C bus specification, and controls the read/write access to all the CRX14 registers.
- The RAM buffer. It is bi-directional. . It stores all the request frame Bytes to be transmitted to the PICC, and all the received Bytes sent by the PICC on the answer frame.
- The transmitter. It powers the PICCs by generating a 13.56MHz signal on an external antenna. The resulting field is 10% modulated using ASK (amplitude shift keying) for outgoing data.
- The receiver. It demodulates the signal generated on the antenna by the load variation of the PICC. The resulting signal is decoded by a 847kHz BPSK (binary phase shift keying) sub-carrier decoder.

The CRX14 is designed to be connected to a digital host (Microcontroller or ASIC). This host has to manage the entire communication protocol in both transmit and receive modes, through the I²C serial bus.

Figure 1. Logic diagram

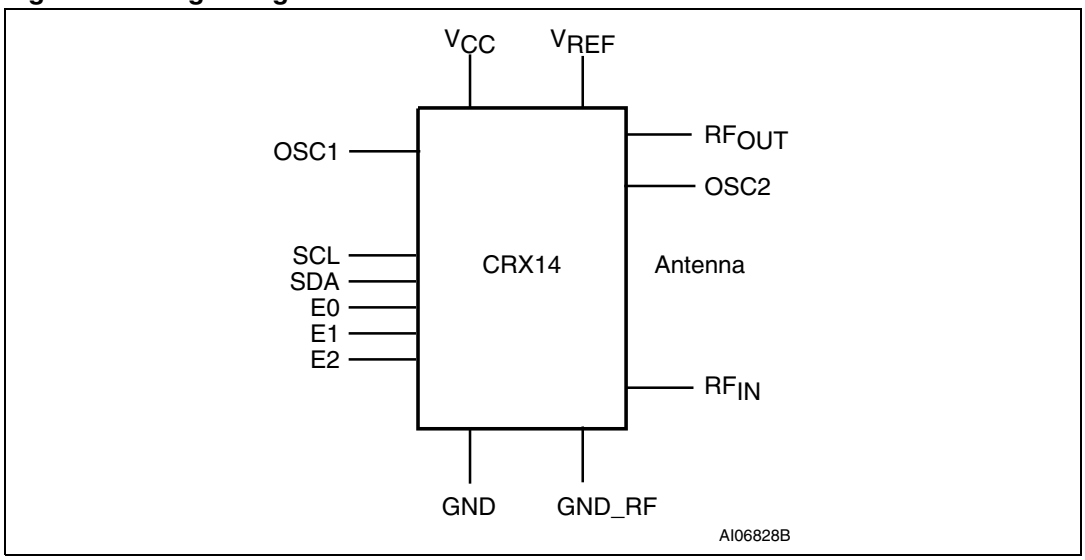


Table 1. Signal names

Signal	Description
RF_{OUT}	Antenna Output Driver
RF_{IN}	Antenna Input Filter
OSC1	Oscillator Input
OSC2	Oscillator Output
E0, E1, E2	Chip Enable Inputs
SDA	I²C Bi-Directional Data
SCL	I²C Clock
V_{CC}	Power Supply
GND	Ground
V_{REF}	Transmitter Reference Voltage
GND_RF	Ground for RF circuitry

Figure 2. Logic block diagram

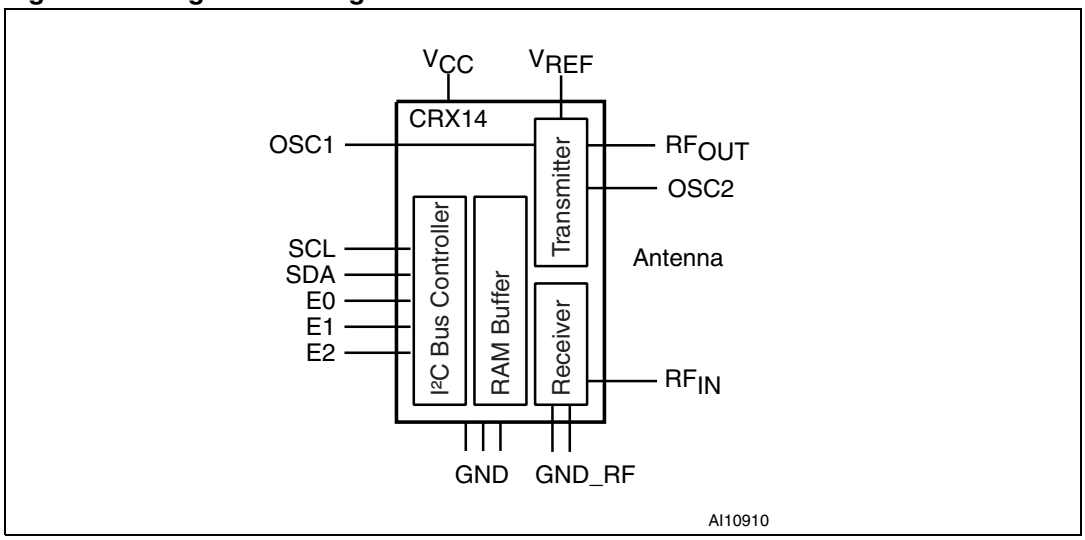
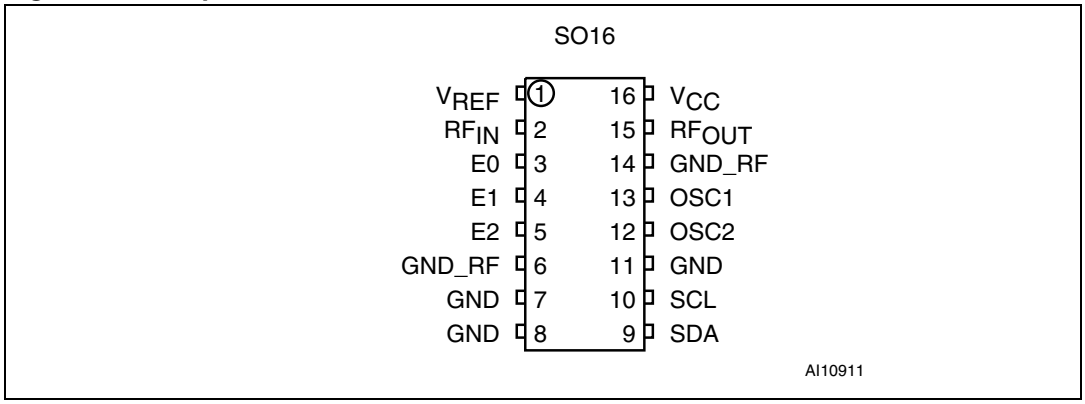


Figure 3. SO pin connections



2 Signal description

See *Figure 1: Logic diagram*, and *Table 1: Signal names*, for an overview of the signals connected to this device.

2.1 Oscillator (OSC1, OSC2)

The OSC1 and OSC2 pins are internally connected to the on-chip oscillator circuit. The OSC1 pin is the input pin, the OSC2 is the output pin. For correct operation of the CRX14, it is required to connect a 13.56MHz quartz crystal across OSC1 and OSC2. If an external clock is used, it must be connected to OSC1 and OSC2 must be left open.

2.2 Antenna output driver (RF_{OUT})

The Antenna Output Driver pin, RF_{OUT}, generates the modulated 13.56MHz signal on the antenna. Care must be taken as it will not withstand a short-circuit.

RF_{OUT} has to be connected to the antenna circuitry as shown in *Figure 4: CRX14 application schematic* The LRC antenna circuitry must be connected across the RF_{OUT} pin and GND.

2.3 Antenna input filter (RF_{IN})

The antenna input filter of the CRX14, RF_{IN}, has to be connected to the external antenna through an adapter circuit, as shown in *Figure 4*.

The input filter demodulates the signal generated on the antenna by the load variation of the PICC. The resulting signal is then decoded by the 847kHz BPSK decoder.

2.4 Transmitter reference voltage (V_{REF})

The Transmitter Reference Voltage input, V_{REF}, provides a reference voltage used by the output driver for ASK modulation.

The Transmitter Reference Voltage input should be connected to an external capacitor, as shown in *Figure 4*.

2.5 Serial clock (SCL)

The SCL input pin is used to strobe all I²C data in and out of the CRX14. In applications where this line is used by slave devices to synchronize the bus to a slower clock, the master must have an open drain output, and a pull-up resistor must be connected from the Serial Clock (SCL) to V_{CC}. (*Figure 5* indicates how the value of the pull-up resistor can be calculated).

In most applications, though, this method of synchronization is not employed, and so the pull-up resistor is not necessary, provided that the master has a push-pull (rather than open drain) output.

2.6 Serial data (SDA)

The SDA signal is bi-directional. It is used to transfer I²C data in and out of the CRX14. It is an open drain output that may be wire-OR'ed with other open drain or open collector signals on the bus. A pull-up resistor must be connected from Serial data (SDA) to V_{CC}. (*Figure 5* indicates how the value of the pull-up resistor can be calculated).

2.7 Chip enable (E0, E1, E2)

The Chip Enable inputs E0, E1, E2 are used to set and reset the value on the three least significant bits (b3, b2, b1) of the 7-bit I²C Device Select Code. They are used for hardwired addressing, allowing up to eight CRX14 devices to be addressed on the same I²C bus. These inputs may be driven dynamically or tied to V_{CC} or GND to establish the Device Select Code (note that the V_{IL} and V_{IH} levels for the inputs are CMOS compatible, not TTL compatible).

When left open, E0, E1 and E2 are internally read at the logic level 0 due to the internal pull-down resistors connected to each inputs.

2.8 Power supply (V_{CC}, GND, GND_RF)

Power is supplied to the CRX14 using the V_{CC}, GND and GND_RF pins.

V_{CC} is the Power Supply pin that supplies the power (+5V) for all CRX14 operations.

The GND and GND_RF pins are ground connections. They must be connected together.

Decoupling capacitors should be connected between the V_{CC} Supply Voltage pin, the GND Ground pin and the GND_REF Ground pin to filter the power line, as shown in *Figure 4*.

Figure 4. CRX14 application schematic

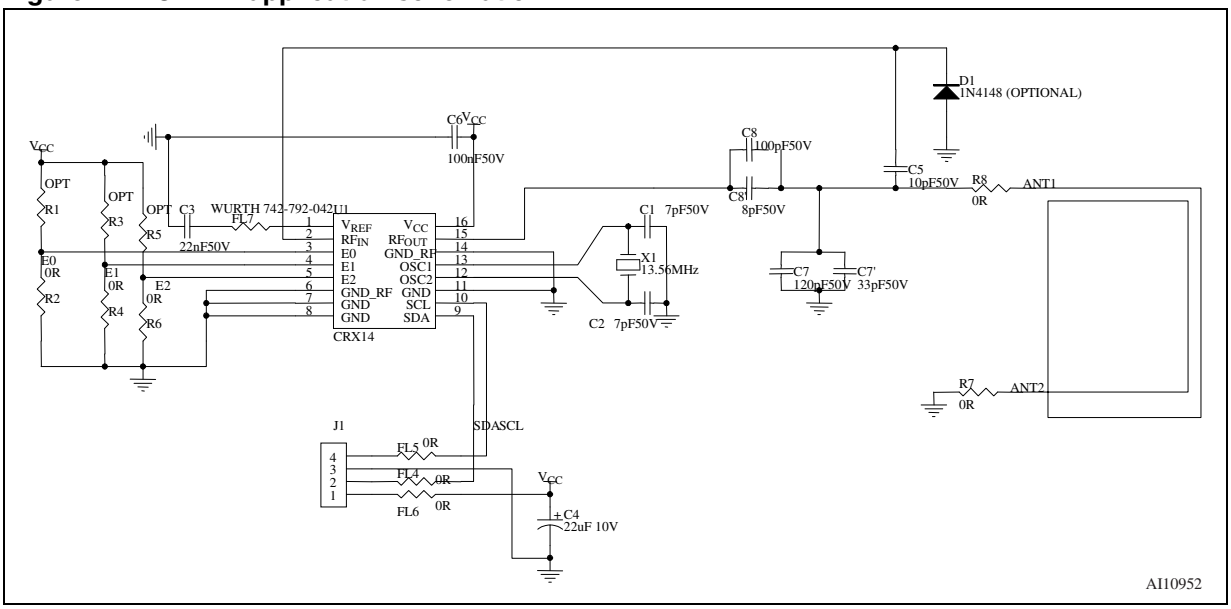
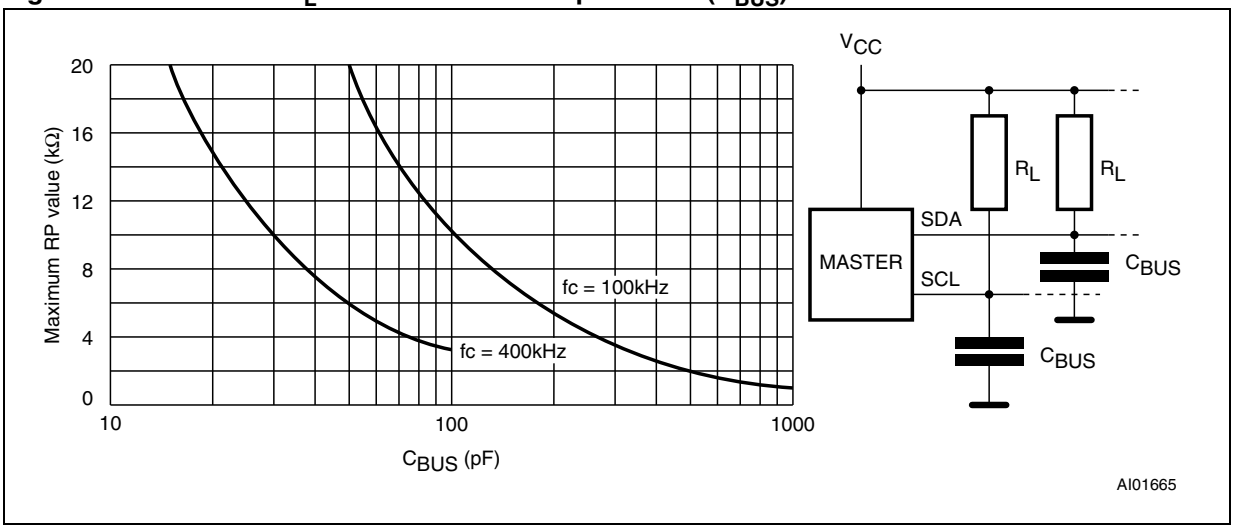


Figure 5. Maximum R_L value versus bus capacitance (C_{BUS}) for an I^2C bus



3 CRX14 registers

The CRX14 chip coupler contains six volatile registers. It is entirely controlled, at both digital and analog level, using the four registers listed below and shown in *Table 2*:

- Parameter Register
- Input/Output Frame Register
- Authentication Register
- Slot Marker Register

The other 3 registers are located at addresses 02h, 04h and 05h. They are “ST Reserved”, and must not be used in end-user applications.

In the I²C protocol, all data Bytes are transmitted Most Significant Byte first, with each Byte transmitted Most significant bit first.

Table 2. CRX14 control registers

Address		Length	Access	Purpose
00h	Parameter Register	1 Byte	W	Set parameter register
			R	Read parameter register
01h	Input/output Frame Register	36 Bytes	W	Store and send request frame to the PICC. Wait for PICC answer frame
			R	Transfer PICC answered frame data to Host
02h	Authenticate Register	NA	W	Start the Authentication process
			R	Get the Authentication status
03h	Slot Marker Register	1 Byte	W	Launch the automated anti-collision process from Slot_0 to Slot_15
			R	Return data FFh
04h	ST Reserved	NA	R and W	ST Reserved. Must not be used
05h	ST Reserved	NA	R and W	ST Reserved. Must not be used

3.1 Parameter register (00h)

The Parameter Register is an 8-bit volatile register used to configure the CRX14, and thus, to customize the circuit behavior. The Parameter Register is located at the I²C address 00h and it is accessible in I²C Read and Write modes. Its default value, 00h, puts the CRX14 in standard ISO14443 type-B configuration.

Table 3. Parameter register bits description

Bit	Control	Value	Description
b_0	Frame Standard	0	ISO14443 type-B frame management
		1	RFU(1)
b_1	RFU	0	Not used

Table 3. Parameter register bits description (continued)

Bit	Control	Value	Description
b_2	Answer Frame Format	0	Answer PICC Frames are delimited by SOF and EOF
		1	Answer PICC Frames do not provide SOF and EOF delimiters
b_3	ASK Modulation Depth	0	10% ASK modulation depth mode
		1	RFU
b_4	Carrier Frequency	0	13.56MHz carrier on RF OUT is OFF
		1	13.56MHz carrier on RF OUT is ON
b_5	t_{WDG} Answer delay watchdog	b5=0, b6=0: Watchdog time-out = 500µs to be used for read b5=0, b6=1: Watchdog time-out = 5ms to be used for authentication b5=1, b6=0: Watchdog time-out = 10ms to be used for write b5=1, b6=1: Watchdog time-out = 309ms to be used for MCU timings	
b_6			
b_7	RFU	0	Not used

1. RFU = Reserved for Future Use.

3.2 Input/output frame register (01h)

The Input/Output Frame Register is a 36-Byte buffer that is accessed serially from Byte 0 through to Byte 35 (see *Table 4*). It is located at the I²C address 01h.

The Input/Output Frame Register is the buffer in which the CRX14 stores the data Bytes of the request frame to be sent to the PICC. It automatically stores the data Bytes of the answer frame received from the PICC. The first Byte (Byte 0) of the Input/Output Frame Register is used to store the frame length for both transmission and reception.

When accessed in I²C Write mode , the register stores the request frame Bytes that are to be transmitted to the PICC. Byte 0 must be set with the request frame length (in Bytes) and the frame is stored from Byte 1 onwards. At the end of the transmission, the 16-bit CRC is automatically added. After the transmission, the CRX14 wait for the PICC to send back an answer frame. When correctly decoded, the PICC answer frame Bytes are stored in the Input/Output Frame Register from Byte 1 onwards. Byte 0 stores the number of Bytes received from the PICC.

When accessed in I²C Read mode, the Input/Output Register sends back the last PICC answer frame Bytes, if any, with Byte 0 transmitted first. The 16-bit CRC is not stored, and it is not sent back on the I²C bus.

The Input/Output Frame Register is set to all 00h between transmission and reception. If there is no answer from the PICC, Byte 0 is set to 00h. In the case of a CRC error, Byte 0 is set to FFh, and the data Bytes are discarded and not appended in the register.

The CRX14 Input/Output Frame Register is so designed as to generate all the ST short range memory command frames. It can also generate all standardized ISO14443 type-B command frames like REQB, SLOT-MARKER, ATTRIB, HALT, and get all the answers like ATQB, or answer to ATTRIB. All ISO14443 type-B compliant PICCs can be accessed by the CRX14 provided that their data frame exchange is not longer than 35 Bytes in both request and answer.

Table 4. Input/output frame register description

Byte 0	Byte 1	Byte 2	Byte 3	...	Byte 34	Byte 35
Frame Length	First data Byte	Second data Byte				Last data Byte
	<---------- Request and Answer Frame Bytes exchanged on the RF ---------->					
00h No Byte transmitted FFh CRC Error						
xxh Number of transmitted Bytes						

3.3 Authenticate register (02h)

The Authenticate Register is used to trigger the complete authentication exchange between the CRX14 and the secured ST short range memory. It is located at the I²C address 02h.

The Authentication system is based on a proprietary challenge/response mechanism that allows the application software to authenticate a secured ST short range memory of the SRXxxx family. A reader designed with the CRX14 can check the authenticity of a memory device and protect the application system against silicon copies or emulators.

A complete description of the Authentication system is available under Non Disclosure Agreement (NDA) with STMicroelectronics. For more details about this CRX14 function, please contact the nearest STMicroelectronics sales office.

3.4 Slot marker register (03h)

The slot Marker Register is located at the I²C address 03h. It is used to trigger an automated anti-collision sequence between the CRX14 and any ST short range memory present in the electromagnetic field. With one I²C access, the CRX14 launches a complete stream of commands starting from PCALL16(), SLOT_MARKER(1), SLOT_MARKER(2) up to SLOT_MARKER(15), and stores all the identified Chip_IDs into the Input/Output Frame Register (I²C address 01h).

This automated anti-collision sequence simplifies the host software development and reduces the time needed to interrogate the 16 slots of the STMicroelectronics anti-collision mechanism.

When accessed in I²C Write mode, the Slot Marker Register starts generating the sequence of anti-collision commands. After each command, the CRX14 wait for the ST short range memory answer frame which contains the Chip_ID. The validity of the answer is checked and stored into the corresponding Status Slot Bit (Byte 1 and Byte 2 as described in *Table 5*). If the answer is correct, the Status Slot Bit is set to ‘1’ and the Chip_ID is stored into the corresponding Slot_Register. If no answer is detected, the Status Slot Bit is set to ‘0’, and the corresponding Slot_Register is set to 00h. If a CRC error is detected, the Status Slot Bit is set to ‘0’, and the corresponding Slot_Register is set to FFh.

Each time the Slot Marker Register is accessed in I²C Write mode, Byte 0 of the Input/Output Frame Register is set to 18, Bytes 1 and 2 provide Status Bits Slot information, and Bytes 3 to 18 store the corresponding Chip_ID or error code.

The Slot Marker Register cannot be accessed in I²C Read mode. All the anti-collision data can be accessed by reading the Input/Output Frame Register at the I²C address 01h.

Table 5. Slot marker register description

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Byte 0	Number of stored Bytes: fixed to 18							
Byte 1	Status Slot Bit 7	Status Slot Bit 6	Status Slot Bit 5	Status Slot Bit 4	Status Slot Bit 3	Status Slot Bit 2	Status Slot Bit 1	Status Slot Bit 0
Byte 2	Status Slot Bit 15	Status Slot Bit 14	Status Slot Bit 13	Status Slot Bit 12	Status Slot Bit 11	Status Slot Bit 10	Status Slot Bit 9	Status Slot Bit 8
Byte 3	Slot_Register 0 = Chip_ID value detected in Slot 0							
Byte 4	Slot_Register 1 = Chip_ID value detected in Slot 1							
Byte 5	Slot_Register 2 = Chip_ID value detected in Slot 2							
Byte 6	Slot_Register 3 = Chip_ID value detected in Slot 3							
Byte n	.....							
Byte 17	Slot_Register 14 = Chip_ID value detected in Slot 14							
Byte 18	Slot_Register 15 = Chip_ID value detected in Slot 15							

Status bit value description:
1: No error detected. The Chip_ID stored in the Slot register is valid.
0: Error detected
– Slot register = 00h: No answer frame detected from ST short range memory
– Slot register = FFh: Answer Frame detected with CRC error. Collision may have occurred

4 CRX14 I²C protocol description

The CRX14 is compatible with the I²C serial bus memory standard, which is a two-wire serial interface that uses a bi-directional data bus and serial clock.

The CRX14 has a pre-programmed, 4-bit identification code, ’1010’ (as shown in *Table 6*), that corresponds to the I²C bus definition. With this code and the three Chip Enable inputs (E2, E1, E0) up to eight CRX14 devices can be connected to the I²C bus, and selected individually.

The CRX14 behaves as a slave device in the I²C protocol, with all CRX14 operations synchronized to the serial clock.

I²C Read and Write operations are initiated by a START condition, generated by the bus master.

The START condition is followed by the Device Select Code and by a Read/Write bit (R/$\overline{W}$). It is terminated by an acknowledge bit. The Device Select Code consists of seven bits (as shown in *Table 6*):

- the Device Code (first four bits)
- plus three bits corresponding to the states of the three Chip Enable inputs, E2, E1 and E0, respectively

When data is written to the CRX14, the device inserts an acknowledge bit (9th bit) after the bus master’s 8-bit transmission.

When the bus master reads data, it also acknowledges the receipt of the data Byte by inserting an acknowledge bit (9th bit).

Data transfers are terminated by a STOP condition after an ACK for Write, or after a NoACK for Read.

The CRX14 supports the I²C protocol, as summarized in *Figure 6*.

Any device that sends data on to the bus, is defined as a transmitter, and any device that reads the data, as a receiver.

The device that controls the data transfer is known as the master, and the other, as the slave. A data transfer can only be initiated by the master, which also provides the serial clock for synchronization. The CRX14 is always a slave device in all I²C communications. All data are transmitted Most Significant Bit (MSB) first.

Table 6. Device select code

	Device code				Chip enable			R$\overline{W}$
	b7	b6	b5	b4	b3	b2	b1	b0
CRX14 Select	1	0	1	0	E2	E1	E0	R$\overline{W}$

4.1 I²C start condition

START is identified by a High-to-Low transition of the Serial Data line, SDA, while the Serial Clock, SCL, is stable in the High state. A START condition must precede any data transfer command.

The CRX14 continuously monitors the SDA and SCL lines for a START condition (except during Radio Frequency data exchanges), and will not respond unless one is sent.

4.2 I²C stop condition

STOP is identified by a Low-to-High transition of the Serial Data line, SDA, while the Serial Clock, SCL, is stable in the High state.

A STOP condition terminates communications between the CRX14 and the bus master.

A STOP condition at the end of an I²C Read command, after (and only after) a NoACK, forces the CRX14 into its stand-by state.

A STOP condition at the end of an I²C Write command triggers the Radio Frequency data exchange between the CRX14 and the PICC.

4.3 I²C acknowledge bit (ACK)

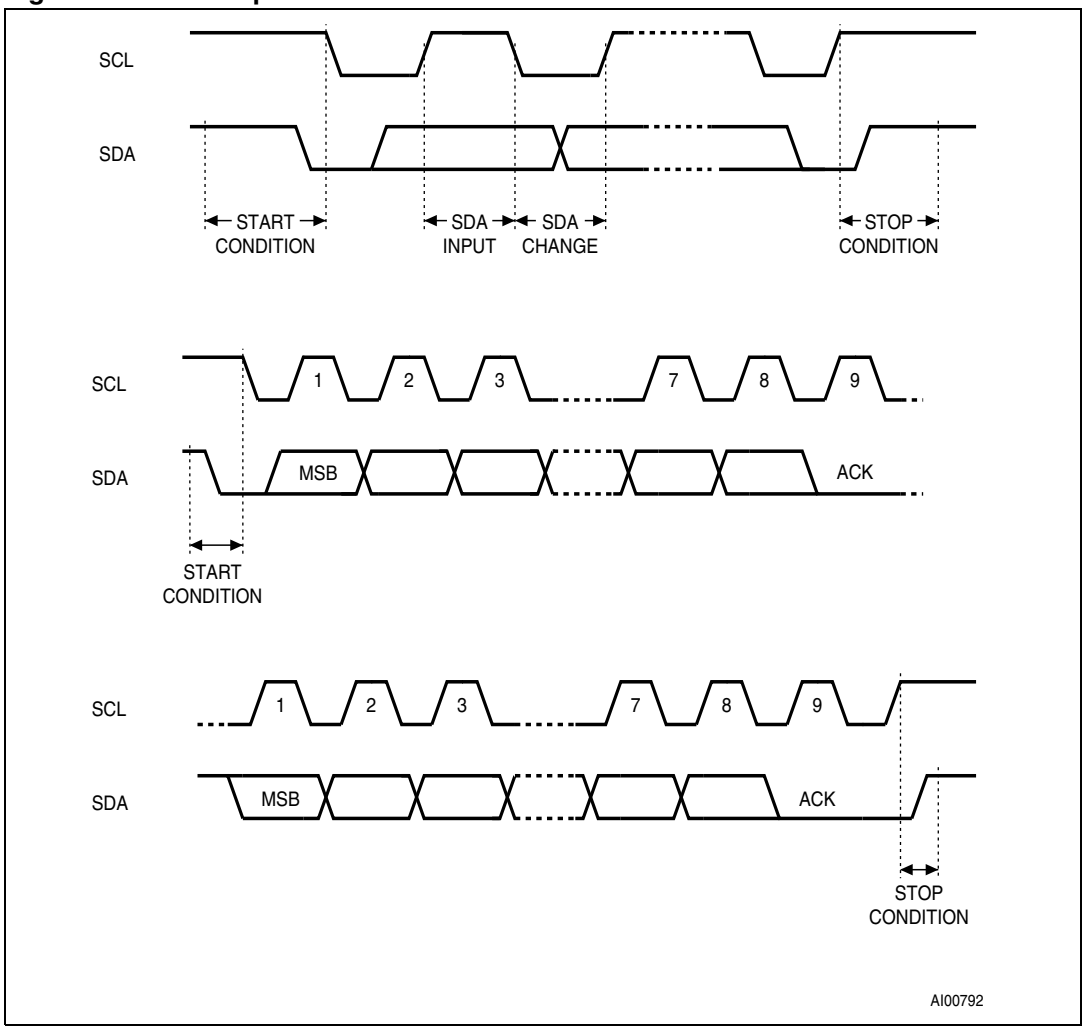
An acknowledge bit is used to indicate a successful data transfer on the I²C bus.

The bus transmitter, either master or slave, releases the Serial Data line, SDA, after sending 8 bits of data. During the 9th clock pulse the receiver pulls the SDA line Low to acknowledge the receipt of the 8 data bits.

4.4 I²C data input

During data input, the CRX14 samples the SDA bus signal on the rising edge of the Serial Clock, SCL. For correct device operation, the SDA signal must be stable during the Low-to-High Serial Clock transition, and the data must change only when the SCL line is Low.

Figure 6. I²C bus protocol



4.5 I²C memory addressing

To start up communication with the CRX14, the bus master must initiate a START condition. Then, the bus master sends 8 bits (with the most significant bit first) on the Serial Data line, SDA. These bits consist of the Device Select Code (7 bits) plus a $R\overline{W}$ bit.

According to the I²C bus definition, the seven most significant bits of the Device Select Code are the Device Type Identifier. For the CRX14, these bits are defined as shown in *Table 6*.

The 8th bit is the Read/Write bit ($R\overline{W}$). It is set to '1' for I²C Read, and to '0' for I²C Write operations.

If the data sent by the bus master matches the Device Select Code of a CRX14 device, the corresponding device returns an acknowledgment on the SDA bus during the 9th bit time.

The CRX14 devices whose Device Select Codes do not correspond to the data sent, generate a No-ACK. They deselect themselves from the bus and go into stand-by mode.

4.6 CRX14 I²C write operations

The bus master sends a START condition, followed by a Device Select Code and the R/$\overline{W}$ bit set to ’0’. The CRX14 that corresponds to the Device Select Code, acknowledges and waits for the bus master to send the Byte address of the register that is to be written to. After receipt of the address, the CRX14 returns another ACK, and waits for the bus master to send the data Bytes that are to be written.

In the CRX14 I²C Write mode, the bus master may sends one or more data Bytes depending on the selected register.

The CRX14 replies with an ACK after each data Byte received. The bus master terminates the transfer by generating a STOP condition.

The STOP condition at the end of a Write access to the Input/Output Frame, Authenticate or Anti-Collision Register causes the Radio Frequency data exchange between the CRX14 and the PICC to be started.

During the Radio Frequency data exchange, the CRX14 disconnects itself from the I²C bus. The time (t_{RFEX}) needed to complete the exchange is not fixed as it depends on the PICC command format. To know when the exchange is complete, the bus master uses an ACK polling sequence as shown in *Figure 8*. It consists of the following:

- Initial condition: a Radio Frequency data exchange is in progress.
- Step 1: the master issues a START condition followed by the first Byte of the new instruction (Device Select Code plus R/$\overline{W}$ bit).
- Step 2: if the CRX14 is busy, no ACK is returned and the master goes back to Step 1. If the CRX14 has completed the Radio Frequency data exchange, it responds with an ACK, indicating that it is ready to receive the second part of the next instruction (the first Byte of this instruction being sent during Step 1).

Figure 7. CRX14 I²C write mode sequence

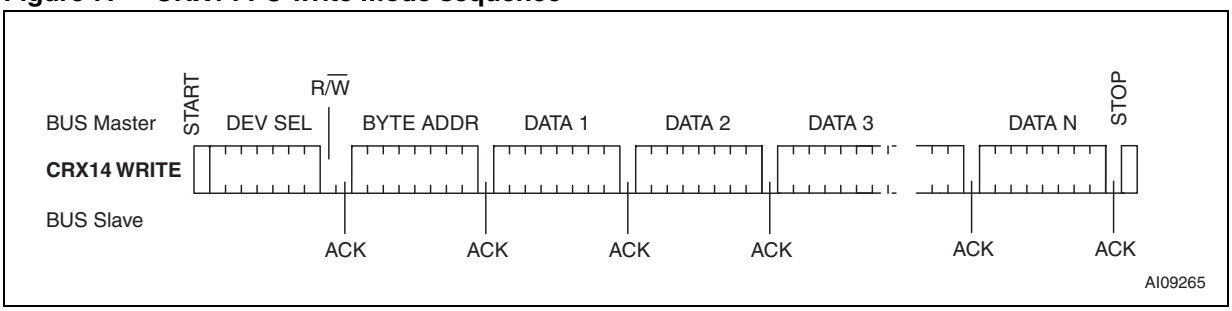
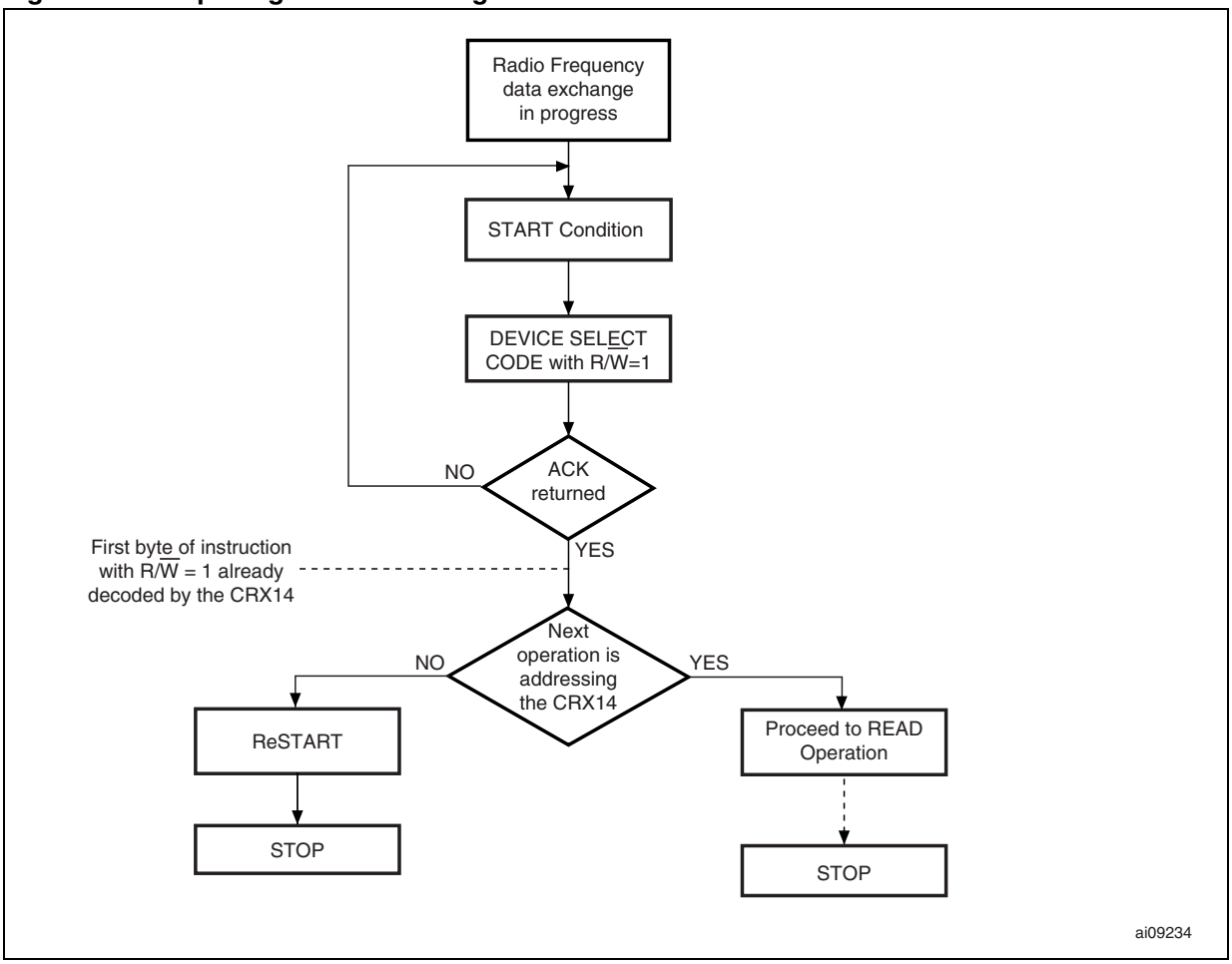


Figure 8. I²C polling flowchart using ACK



4.7 CRX14 I²C read operations

To send a Read command, the bus master sends a START condition, followed by a Device Select Code and the R/$\overline{W}$ bit set to '1'.

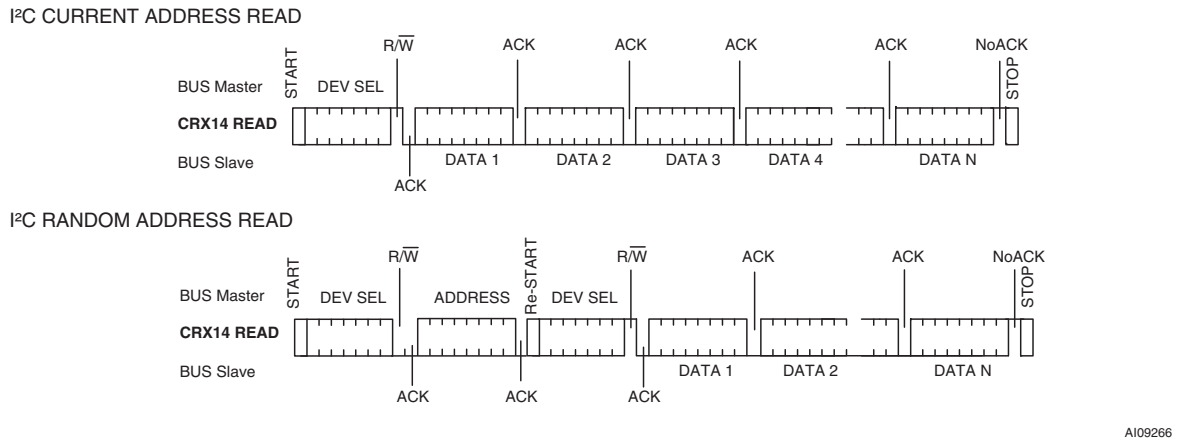
The CRX14 that corresponds to the Device Select Code acknowledges and outputs the first data Byte of the addressed register.

To select a specific register, a dummy Write command must first be issued, giving an address Byte but no data Bytes, as shown in the bottom half of *Figure 9*. This causes the new address to be stored in the internal address pointer, for use by the Read command that immediately follows the dummy Write command.

In the I²C Read mode, the CRX14 may read one or more data Bytes depending on the selected register. The bus master has to generate an ACK after each data Byte to read all the register data in a continuous stream. Only the last data Byte should not be followed by an ACK. The master then terminates the transfer with a STOP condition, as shown in *Figure 9*.

After reading each Byte, the CRX14 waits for the master to send an ACK during the 9th bit time. If the master does not return an ACK within this time, the CRX14 terminates the data transfer and switches to stand-by mode.

Figure 9. CRX14 I²C read modes sequences



5 Applying the I²C protocol to the CRX14 registers

5.1 I²C parameter register protocol

Figure 10 shows how new data is written to the Parameter Register. The new value becomes active after the I²C STOP condition.

Figure 11 shows how to read the Parameter Register contents. The CRX14 sends and re-sends the Parameter Register contents until it receives a NoACK from the I²C Host.

The CRX14 supports the I²C Current Address and Random Address Read modes. The Current Address Read mode can be used if the previous command was issued to the register where the Read is to take place.

Figure 10. Host-to-CRX14 transfer: I²C write to parameter register

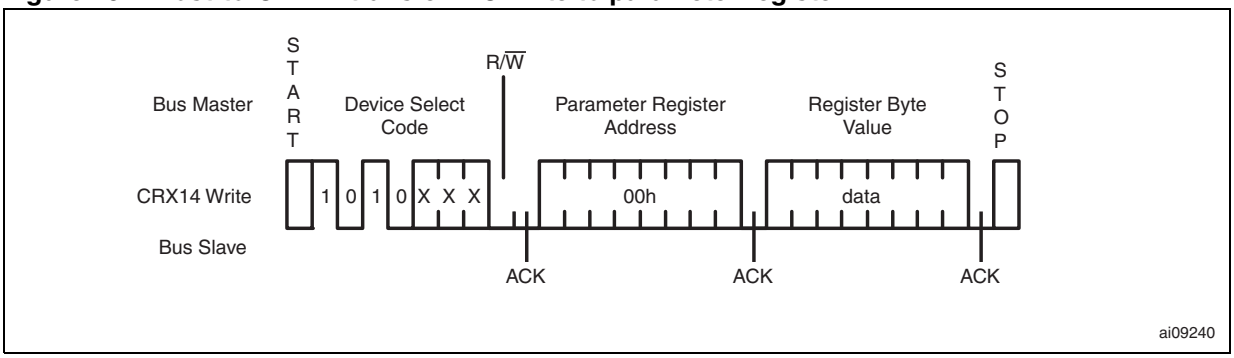


Figure 11. CRX14-to-host transfer: I²C random address read from parameter register

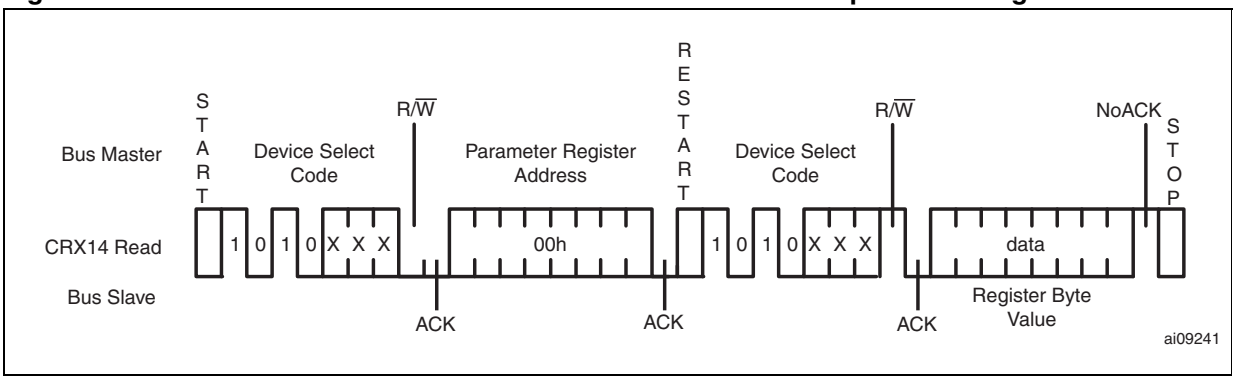
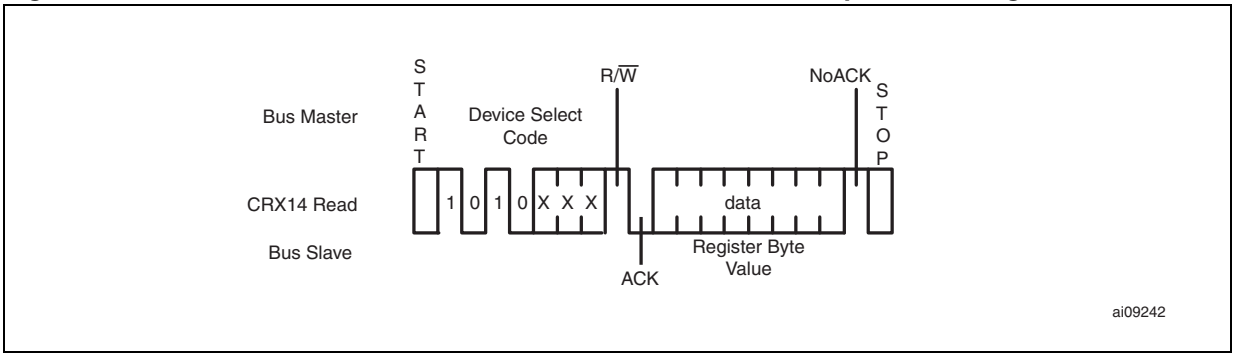


Figure 12. CRX14-to-host transfer: I²C current address read from parameter register



5.2 I²C input/output frame register protocol

Figure 13 shows how to store a PICC request frame command of *N* Bytes into the Input/Output Frame Register.

After the I²C STOP condition, the request frame is RF transmitted in the ISO14443 type-B format. The CRX14 then waits for the PICC answer frame which will also be stored in the Input/Output Frame Register. The request frame is over-written by the answer frame.

Figure 14 shows how to read an *N*-Byte PICC answer frame.

The two CRC Bytes generated by the PICC are not stored.

The CRX14 continues to output data Bytes until a NoACK has been generated by the I²C Host, and received by the CRX14. After all 36 Bytes have been output, the CRX14 “rolls over”, and starts outputting from the start of the Input/Output Frame Register again.

The CRX14 supports the I²C Current Address and Random Address Read modes. The Current Address Read mode can be used if the previous command was issued to the register where the Read is to take place.

Figure 13. Host-to-CRX14 transfer: I²C write to I/O frame register for ISO14443B

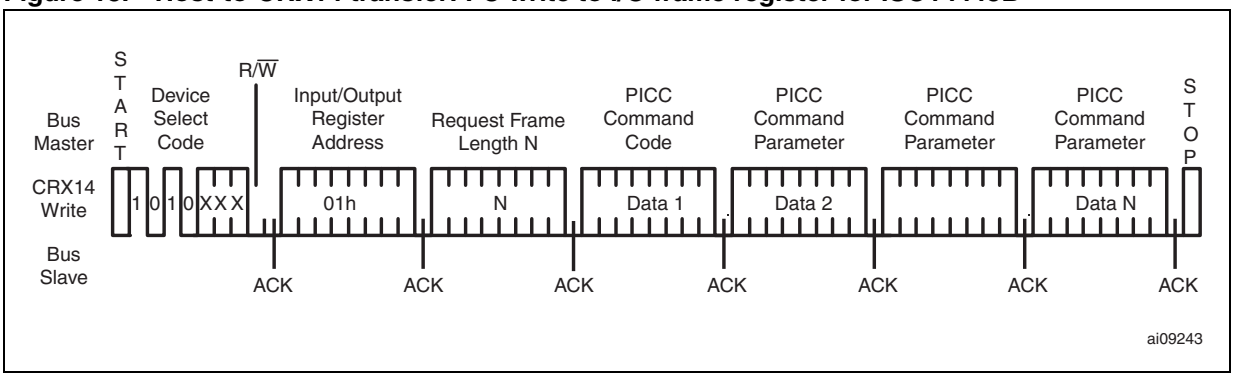


Figure 14. CRX14-to-host transfer: I²C random address read from I/O frame register for ISO14443B

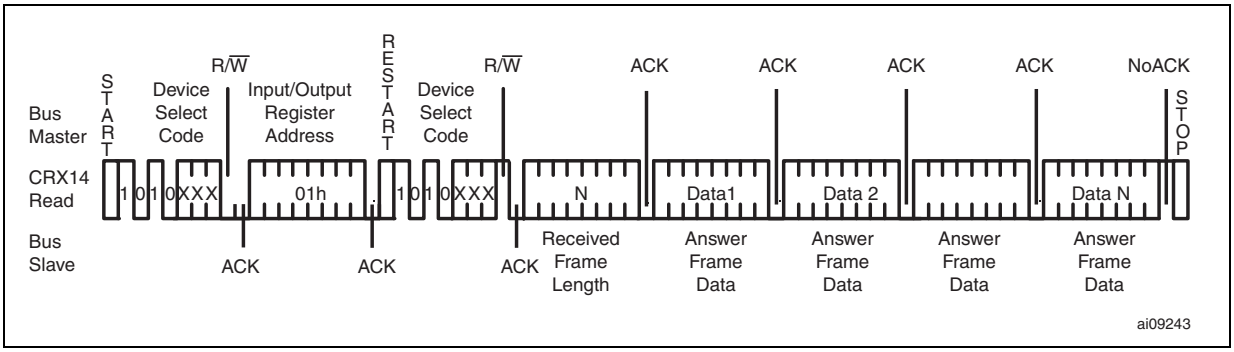
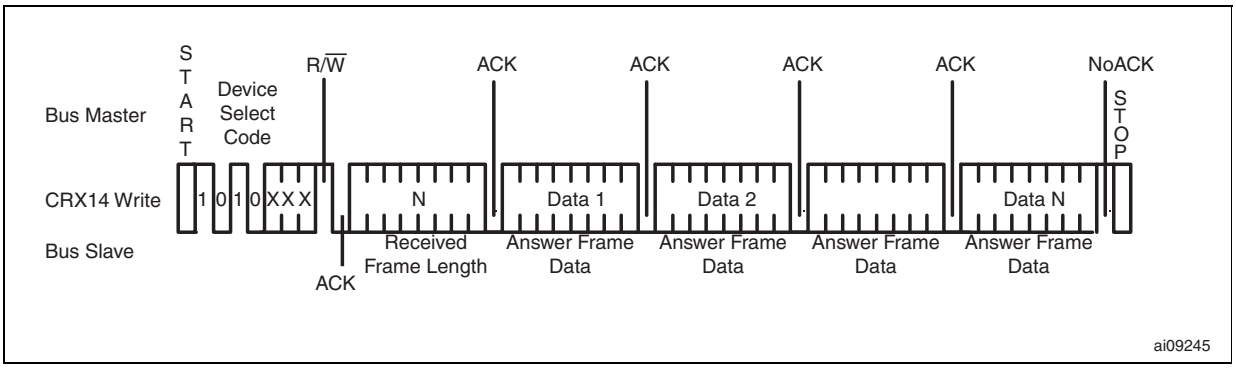


Figure 15. CRX14-to-host transfer: I²C current address read from I/O frame register for ISO14443B



5.3 I²C authenticate register protocol

For information please contact your nearest STMicroelectronics sales office.

5.4 I²C slot marker register protocol

An I²C Write command to the Slot Marker Register generates an automated sixteen-command loop (See *Figure 16* for a description of the command).

All the answers from the ST short range memory devices that are detected, are written in the Input/Output Frame Register.

Read from the I²C Slot Marker Register is not supported by the CRX14. If the I²C Host tries to read the Slot Marker Register, the CRX14 will return the data value FFh in both Random Address and Current Address Read modes until NoACK is generated by the I²C Host.

The result of the detection sequence is stored in the Input/Output Frame Register. This Register can be read by the host by using I²C Random Address Read.

Figure 16. Host-to-CRX14 transfer: I²C write to slot marker register

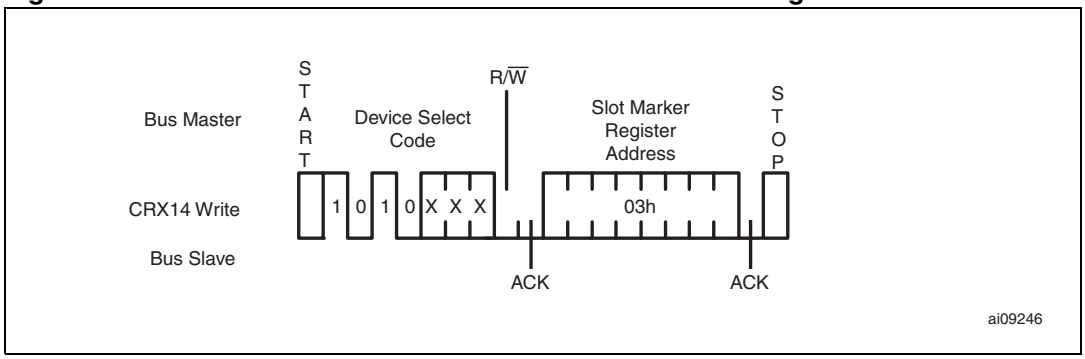


Figure 17. CRX14-to-host transfer: I²C random address read from slot marker register

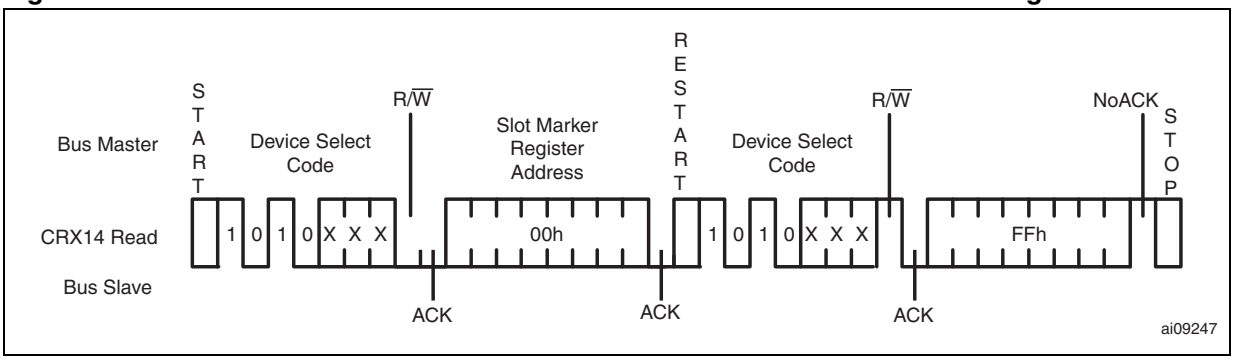
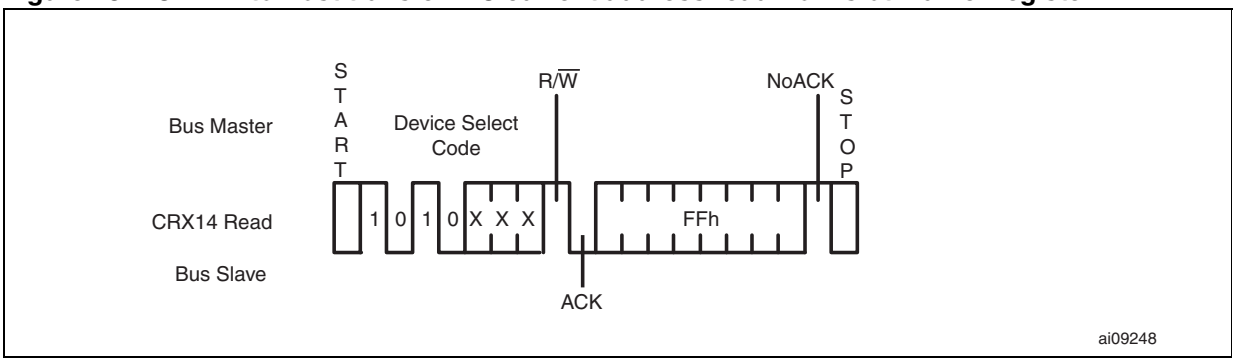


Figure 18. CRX14-to-host transfer: I²C current address read from slot marker register



5.5 Addresses above location 06h

In I²C Write mode, when the CRX14 receives the 8-bit register address, and the address is above location 06h, the device does not acknowledge (NoACK) and deselects itself from the bus. The Serial Data line, SDA, stays at logic '1' (pull-up resistor), and the I²C Host receives a NoACK during the 9th bit time. The SDA line stays High until the STOP condition is issued.

In the I²C Current and Random Address Read modes, when the CRX14 receives the 8-bit register address, and the address is above location 06h, the device does not acknowledge the Device Select Code after the START condition, and deselects itself from the bus.