CR14
ISO14443 type-B contactless coupler chip with anti-collision and CRC management

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
  - Automated ST anti-collision exchange
- I²C communication
  - Two-wire I²C serial interface
  - Supports 400 kHz protocol
  - 3 chip enable pins
  - Up to 8 CR14 connected on the same bus

SO16 (MQ)
150 mils width
Contents

1 Summary description ............................................. 6

2 Signal description ................................................. 9
   2.1 Oscillator (OSC1, OSC2) .................................. 9
   2.2 Antenna output driver (RF_OUT) ......................... 9
   2.3 Antenna input filter (RF_IN) ............................ 9
   2.4 Transmitter reference voltage (V_REF) ................ 9
   2.5 Serial clock (SCL) ...................................... 9
   2.6 Serial data (SDA) ....................................... 10
   2.7 Chip enable (E0, E1, E2) .............................. 10
   2.8 Power supply (V_CC, GND, GND_RF) .................. 10

3 CR14 registers .................................................... 12
   3.1 Parameter register (00h) ............................... 12
   3.2 Input/Output Frame Register (01h) .................... 13
   3.3 Slot marker register (03h) ............................. 14

4 CR14 I²C protocol description ................................. 16
   4.1 I²C start condition .................................. 16
   4.2 I²C stop condition ................................... 17
   4.3 I²C acknowledge bit (ACK) .......................... 17
   4.4 I²C data input ....................................... 17
   4.5 I²C memory addressing ............................... 18
   4.6 CR14 I²C write operations ......................... 19
   4.7 CR14 I²C read operations .......................... 20

5 Applying the I²C protocol to the CR14 registers .......... 22
   5.1 I²C parameter register protocol ....................... 22
   5.2 I²C input/output frame register protocol .......... 23
   5.3 I²C slot marker register protocol ..................... 24
   5.4 Addresses above location 06h ......................... 25
6 CR14 ISO14443 type-B radio frequency data transfer ..................... 26
   6.1 Output RF data transfer from the CR14 to the PICC (request frame) ... 26
   6.2 Transmission format of request frame characters .......................... 26
   6.3 Request start of frame ......................................................... 27
   6.4 Request end of frame .......................................................... 27
   6.5 Input RF data transfer from the PICC to the CR14 (answer frame) ...... 28
   6.6 Transmission format of answer frame characters ............................ 28
   6.7 Answer start of frame .......................................................... 29
   6.8 Answer end of frame ............................................................ 29
   6.9 Transmission frame ............................................................... 29
   6.10 CRC .......................................................... 30

7 Tag access using the CR14 coupler ............................................. 31
   7.1 Standard TAG command access description .................................. 31
   7.2 Anti-collision TAG sequence ..................................................... 32

8 Maximum rating ................................................................. 35

9 DC and AC parameters .......................................................... 36

10 Package mechanical ............................................................. 41

11 Ordering information .......................................................... 43

Appendix A ISO14443 type B CRC calculation ............................... 44

Revision history ................................................................. 46
List of tables

Table 1. Signal names ................................................................. 7
Table 2. CR14 control registers .................................................. 12
Table 3. Parameter register bits description ......................... 12
Table 4. Input/output frame register description ...................... 14
Table 5. Slot marker register description ................................ 14
Table 6. Device select code ..................................................... 16
Table 7. CR14 request frame character format ....................... 27
Table 8. Absolute maximum ratings .......................................... 35
Table 9. I²C AC measurement conditions ............................... 36
Table 10. I²C Input Parameters(1,2) ............................................ 36
Table 11. I²C DC characteristics ............................................... 36
Table 12. I²C AC characteristics .............................................. 38
Table 13. RFOUT AC characteristics ........................................ 39
Table 14. RFIN AC characteristics ........................................... 40
Table 15. SO16 narrow - 16 lead plastic small outline, 150 mils body width, package mechanical data .......... 42
Table 16. Ordering information scheme ............................... 43
Table 17. Document revision history ........................................ 46
List of figures

Figure 1. Logic diagram .............................................................. 7
Figure 2. Logic block diagram ................................................... 8
Figure 3. SO pin connections .................................................... 8
Figure 4. CR14 application schematic ......................................... 10
Figure 5. Maximum $R_L$ value versus bus capacitance ($C_{BUS}$) for an I²C bus ......................................................... 11
Figure 6. I²C bus protocol .......................................................... 18
Figure 7. CR14 I²C write mode sequence ..................................... 19
Figure 8. I²C polling flowchart using ACK .................................... 20
Figure 9. CR14 I²C read modes sequences .................................... 21
Figure 10. Host-to-CR14 transfer: I²C write to parameter register ...................... 22
Figure 11. CR14-to-host transfer: I²C random address read from parameter register ...................... 22
Figure 12. Host-to-CR14 transfer: I²C write to I/O frame register for ISO14443B ................. 23
Figure 13. CR14-to-host transfer: I²C random address read from I/O frame register for ISO14443B ................. 23
Figure 14. CR14-to-host transfer: I²C current address read from I/O frame register for ISO14443B ................. 24
Figure 15. Host-to-CR14 transfer: I²C write to slot marker register ...................... 24
Figure 16. CR14-to-host transfer: I²C random address read from slot marker register ...................... 24
Figure 17. CR14-to-host transfer: I²C current address read from slot marker register ...................... 25
Figure 18. Wave transmitted using ASK modulation ......................... 26
Figure 19. CR14 request frame character format ................................ 27
Figure 20. Request start of frame ................................................. 27
Figure 21. Request end of frame ................................................. 27
Figure 22. Wave received using BPSK sub-carrier modulation ................ 28
Figure 23. Answer start of frame ................................................. 29
Figure 24. Answer end of frame ................................................. 29
Figure 25. Example of a complete transmission frame ......................... 30
Figure 26. CRC transmission rules ............................................... 30
Figure 27. Standard TAG command: request frame transmission .................. 31
Figure 28. Standard TAG command: answer frame reception .................... 32
Figure 29. Standard TAG command: complete TAG access description .................. 32
Figure 30. Anti-collision ST short range memory sequence (1) .................... 33
Figure 31. Anti-collision ST short range memory sequence continued .............. 34
Figure 32. I²C AC testing I/O waveform ........................................ 36
Figure 33. I²C AC waveforms .................................................... 37
Figure 34. CR14 synchronous timing ............................................. 39
Figure 35. SO16 narrow - 16 lead plastic small outline, 150 mils body width, Package outline .............. 42
1 Summary description

The CR14 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 CR14 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 CR14 and the PICC is 106 Kbit/s in both transmission and reception modes.

The CR14 follows the ISO14443 type-B recommendation for Radio frequency power and signal interface.

The CR14 is specifically designed for short range applications that need disposable and reusable products.

The CR14 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 CR14 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 CR14 is fabricated in STMicroelectronics High Endurance Single Poly-silicon CMOS technology.

The CR14 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 CR14 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 CR14 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.
Table 1. Signal names

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF\textsubscript{OUT}</td>
<td>Antenna Output Driver</td>
</tr>
<tr>
<td>RF\textsubscript{IN}</td>
<td>Antenna Input Filter</td>
</tr>
<tr>
<td>OSC\textsubscript{1}</td>
<td>Oscillator Input</td>
</tr>
<tr>
<td>OSC\textsubscript{2}</td>
<td>Oscillator Output</td>
</tr>
<tr>
<td>E0, E1, E2</td>
<td>Chip Enable Inputs</td>
</tr>
<tr>
<td>SDA</td>
<td>I²C Bi-Directional Data</td>
</tr>
<tr>
<td>SCL</td>
<td>I²C Clock</td>
</tr>
<tr>
<td>V\textsubscript{CC}</td>
<td>Power Supply</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>V\textsubscript{REF}</td>
<td>Transmitter Reference Voltage</td>
</tr>
<tr>
<td>GND_RF</td>
<td>Ground for RF circuitry</td>
</tr>
</tbody>
</table>
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 CR14, 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 (RFOUT)

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

RFOUT has to be connected to the antenna circuitry as shown in Figure 4: CR14 application schematic. The LRC antenna circuitry must be connected across the RFOUT pin and GND.

2.3 Antenna input filter (RFIN)

The antenna input filter of the CR14, RFIN, 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 (VREF)

The Transmitter Reference Voltage input, VREF, 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 CR14. 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 VCC. (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 CR14. 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 VCC. (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 CR14 devices to be addressed on the same I²C bus. These inputs may be driven dynamically or tied to VCC or GND to establish the Device Select Code (note that the VIL and VIH 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 (VCC, GND, GND_RF)

Power is supplied to the CR14 using the VCC, GND and GND_RF pins. VCC is the Power Supply pin that supplies the power (+5V) for all CR14 operations.

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

Decoupling capacitors should be connected between the VCC Supply Voltage pin, the GND Ground pin and the GND_REF Ground pin to filter the power line, as shown in Figure 4.
Figure 5. Maximum $R_L$ value versus bus capacitance ($C_{BUS}$) for an I²C bus.
3 CR14 registers

The CR14 chip coupler contains six volatile registers. It is entirely controlled, at both digital and analog level, using the three registers listed below and shown in Table 2:

- Parameter Register
- Input/Output Frame 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. CR14 control registers

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Length</th>
<th>Access</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>Parameter Register</td>
<td>1 Byte</td>
<td>W</td>
<td>Set parameter register</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>Read parameter register</td>
</tr>
<tr>
<td>01h</td>
<td>Input/output Frame Register</td>
<td>36 Bytes</td>
<td>W</td>
<td>Store and send request frame to the PICC. Wait for PICC answer frame</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>Transfer PICC answered frame data to Host</td>
</tr>
<tr>
<td>02h</td>
<td>ST Reserved</td>
<td>NA</td>
<td>W</td>
<td>ST Reserved, must not be used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>03h</td>
<td>Slot Marker Register</td>
<td>1 Byte</td>
<td>W</td>
<td>Launch the automated anti-collision process from Slot_0 to Slot_15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>Return data FFh</td>
</tr>
<tr>
<td>04h</td>
<td>ST Reserved</td>
<td>NA</td>
<td>R and W</td>
<td>ST Reserved. Must not be used</td>
</tr>
<tr>
<td>05h</td>
<td>ST Reserved</td>
<td>NA</td>
<td>R and W</td>
<td>ST Reserved. Must not be used</td>
</tr>
</tbody>
</table>

3.1 Parameter register (00h)

The Parameter Register is an 8-bit volatile register used to configure the CR14, 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 CR14 in standard ISO14443 type-B configuration.

Table 3. Parameter register bits description

<table>
<thead>
<tr>
<th>Bit</th>
<th>Control</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>Frame Standard</td>
<td>0</td>
<td>ISO14443 type-B frame management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>RFU(1)</td>
</tr>
<tr>
<td>b1</td>
<td>RFU</td>
<td>0</td>
<td>Not used</td>
</tr>
</tbody>
</table>
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 CR14 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 CR14 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 CR14 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 CR14 provided that their data frame exchange is not longer than 35 Bytes in both request and answer.
### 3.3 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 CR14 and any ST short range memory present in the electromagnetic field. With one I²C access, the CR14 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 CR14 waits 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 4. Input/output frame register description

<table>
<thead>
<tr>
<th>Byte 0</th>
<th>Byte 1</th>
<th>Byte 2</th>
<th>Byte 3</th>
<th>...</th>
<th>Byte 34</th>
<th>Byte 35</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Length</td>
<td>First data Byte</td>
<td>Second data Byte</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<-------- Request and Answer Frame Bytes exchanged on the RF -------->

| 00h | No Byte transmitted |
| FFh | CRC Error |
| xxh | Number of transmitted Bytes |

#### Table 5. Slot marker register description

<table>
<thead>
<tr>
<th>Byte 0</th>
<th>b_7</th>
<th>b_6</th>
<th>b_5</th>
<th>b_4</th>
<th>b_3</th>
<th>b_2</th>
<th>b_1</th>
<th>b_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of stored Bytes: fixed to 18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Byte 1</th>
<th>Status Slot Bit 7</th>
<th>Status Slot Bit 6</th>
<th>Status Slot Bit 5</th>
<th>Status Slot Bit 4</th>
<th>Status Slot Bit 3</th>
<th>Status Slot Bit 2</th>
<th>Status Slot Bit 1</th>
<th>Status Slot Bit 0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Byte 2</th>
<th>Status Slot Bit 15</th>
<th>Status Slot Bit 14</th>
<th>Status Slot Bit 13</th>
<th>Status Slot Bit 12</th>
<th>Status Slot Bit 11</th>
<th>Status Slot Bit 10</th>
<th>Status Slot Bit 9</th>
<th>Status Slot Bit 8</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Byte 3</th>
<th>Slot_Register 0 = Chip_ID value detected in Slot 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 4</td>
<td>Slot_Register 1 = Chip_ID value detected in Slot 1</td>
</tr>
<tr>
<td>Byte 5</td>
<td>Slot_Register 2 = Chip_ID value detected in Slot 2</td>
</tr>
</tbody>
</table>

---

14/47  
Doc ID 11922 Rev 2
Table 5. Slot marker register description (continued)

<table>
<thead>
<tr>
<th>Byte 6</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Slot_Register 3 = Chip_ID value detected in Slot 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>..</td>
</tr>
<tr>
<td>Byte 17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Slot_Register 14 = Chip_ID value detected in Slot 14</td>
</tr>
<tr>
<td>Byte 18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Slot_Register 15 = Chip_ID value detected in Slot 15</td>
</tr>
</tbody>
</table>

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 CR14 I²C protocol description

The CR14 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 CR14 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 CR14 devices can be connected to the I²C bus, and selected individually.

The CR14 behaves as a slave device in the I²C protocol, with all CR14 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/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 CR14, 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 CR14 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 CR14 is always a slave device in all I²C communications. All data are transmitted Most Significant Bit (MSB) first.

<table>
<thead>
<tr>
<th>CR14 Select</th>
<th>Device code</th>
<th>Chip enable</th>
<th>RW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b7 b6 b5 b4</td>
<td>b3 b2 b1 b0</td>
<td></td>
</tr>
</tbody>
</table>

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 CR14 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 CR14 and the bus master.

A STOP condition at the end of an I²C Read command, after (and only after) a NoACK, forces the CR14 into its stand-by state.

A STOP condition at the end of an I²C Write command triggers the Radio Frequency data exchange between the CR14 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 CR14 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.
4.5 I²C memory addressing

To start up communication with the CR14, 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 RW bit.

According to the I²C bus definition, the seven most significant bits of the Device Select Code are the Device Type Identifier. For the CR14, these bits are defined as shown in Table 6.

The 8th bit is the Read/Write bit (RW). 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 CR14 device, the corresponding device returns an acknowledgment on the SDA bus during the 9th bit time.

The CR14 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 CR14 I²C write operations

The bus master sends a START condition, followed by a Device Select Code and the R/W bit set to ‘0’. The CR14 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 CR14 returns another ACK, and waits for the bus master to send the data Bytes that are to be written.

In the CR14 I²C Write mode, the bus master may sends one or more data Bytes depending on the selected register.

The CR14 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 Register causes the Radio Frequency data exchange between the CR14 and the PICC to be started.

During the Radio Frequency data exchange, the CR14 disconnects itself from the I²C bus. The time (tRFEX) 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/W bit).
- Step 2: if the CR14 is busy, no ACK is returned and the master goes back to Step 1. If the CR14 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. CR14 I²C write mode sequence
4.7 CR14 I²C read operations

To send a Read command, the bus master sends a START condition, followed by a Device Select Code and the R/W bit set to ‘1’.

The CR14 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 CR14 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 CR14 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 CR14 terminates the data transfer and switches to stand-by mode.

Figure 9. CR14 I²C read modes sequences
Applying the I²C protocol to the CR14 registers

5 Applying the I²C protocol to the CR14 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 CR14 sends and resends the Parameter Register contents until it receives a NoACK from the I²C Host.

The CR14 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-CR14 transfer: I²C write to parameter register

*Figure 11.* CR14-to-host transfer: I²C random address read from parameter register

*Figure 12.* CR14-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 CR14 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 CR14 continues to output data Bytes until a NoACK has been generated by the I²C Host, and received by the CR14. After all 36 Bytes have been output, the CR14 “rolls over”, and starts outputting from the start of the Input/Output Frame Register again.

The CR14 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-CR14 transfer: I²C write to I/O frame register for ISO14443B

**Figure 14.** CR14-to-host transfer: I²C random address read from I/O frame register for ISO14443B
5.3 \( \text{I}^2\text{C} \) slot marker register protocol

An \( \text{I}^2\text{C} \) 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 \( \text{I}^2\text{C} \) Slot Marker Register is not supported by the CR14. If the \( \text{I}^2\text{C} \) Host tries to read the Slot Marker Register, the CR14 will return the data value FFh in both Random Address and Current Address Read modes until NoACK is generated by the \( \text{I}^2\text{C} \) 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 \( \text{I}^2\text{C} \) Random Address Read.

Figure 16. Host-to-CR14 transfer: \( \text{I}^2\text{C} \) write to slot marker register

Figure 17. CR14-to-host transfer: \( \text{I}^2\text{C} \) random address read from slot marker register
5.4 Addresses above location 06h

In I²C Write mode, when the CR14 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 CR14 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.
6 CR14 ISO14443 type-B radio frequency data transfer

6.1 Output RF data transfer from the CR14 to the PICC (request frame)

The CR14 output buffer is controlled by the 13.56MHz clock signal generated by the external oscillator and by the request frame generator. The CR14 can be directly connected to an external matching circuit to generate a 13.56MHz sinusoidal carrier frequency on its antenna.

The current driven into the antenna coil is directly generated by the CR14 RFOUT output driver.

If the antenna is correctly tuned, it emits an H-field of a large enough magnitude to power a contactless PICC from a short distance. The energy received on the PICC antenna is converted to a Power Supply Voltage by a regulator, and turned into data bits by the ASK demodulator. The CR14 amplitude modulates the 13.56MHz wave by 10% as represented in Figure 19. The data transfer rate is 106 kbit/s.

![Wave transmitted using ASK modulation](image)

**Figure 19. Wave transmitted using ASK modulation**

Transfer time for one data bit is 1/106 kHz

6.2 Transmission format of request frame characters

The CR14 transmits characters of 10 bits, with the Least Significant Bit (b0) transmitted first, as shown in Figure 20.

Several 10-bit characters, preceded by the Start Of Frame (SOF) and followed by the End Of Frame (EOF), constitute a Request Frame, as shown in Figure 26.

A Request Frame includes the SOF, instructions, addresses, data, CRC and the EOF as defined in the ISO14443 type-B.

Each bit duration is called an Elementary Time Unit (ETU). One ETU is equal to 9.44μs (1/106kHz).
6.3 Request start of frame

The Start Of Frame (SOF) described in Figure 21 consists of:

- a falling edge,
- followed by ten Elementary Time Units (ETU) each containing a logical ‘0’
- followed by a single rising edge
- followed by two ETUs, each containing a logical ‘1’.

Figure 21. Request start of frame

6.4 Request end of frame

The End Of Frame (EOF) shown in Figure 22 consists of:

- a falling edge,
- followed by ten Elementary Time Units (ETU) containing each a logical ‘0’,
- followed by a single rising edge.

Figure 22. Request end of frame
6.5 Input RF data transfer from the PICC to the CR14 (answer frame)

The CR14 uses the ISO14443 type-B retro-modulation scheme which is demodulated and decoded by the \( RF_{\text{IN}} \) circuitry.

The modulation is obtained by modifying the PICC current consumption (load modulation). This load modulation induces an H-field variation, by coupling, that is detected by the CR14 \( RF_{\text{IN}} \) input as a voltage variation on the antenna. The \( RF_{\text{IN}} \) input demodulates this variation and decodes the information received from the PICC.

Data must be transmitted using a 847kHz, BPSK modulated sub-carrier frequency, \( f_s \), as shown in Figure 23, and as specified in ISO14443 type-B. In BPSK, all data state transitions (from ‘0’ to ‘1’ or from ‘1’ to ‘0’) are encoded by phase shift keying the sub-carrier.

Figure 23. Wave received using BPSK sub-carrier modulation

6.6 Transmission format of answer frame characters

The PICC should use the same character format as that used for output data transfer (see Figure 20).

An Answer Frame includes the SOF, data, CRC and the EOF, as illustrated in Figure 26. The data transfer rate is 106 kbit/s.

The CR14 will also accept Answer Frames that do not contain the SOF and EOF delimiters, provided that these Frames are correctly set in the Parameter Register. (See Figure 26).
6.7 Answer start of frame

The PICC SOF must be compliant with the ISO14443 type-B, and is shown in Figure 24:

- Ten or eleven Elementary Time Units (ETU) each containing a logical '0',
- Two ETUs containing a logical ‘1’.

Figure 24. Answer start of frame

6.8 Answer end of frame

The PICC EOF must be compliant with the ISO14443 type-B, and is shown in Figure 25:

- Ten or eleven Elementary Time Units (ETU) each containing a logical '0',
- Two ETUs containing a logical ‘1’

Figure 25. Answer end of frame

6.9 Transmission frame

The Request Frame transmission must be followed by a minimum delay, $t_0$ (see Table), in which no ASK or BPSK modulation occurs, before the Answer Frame can be transmitted. $t_0$ is the minimum time required by the CR14 to switch from transmission mode to reception mode, and should be inserted after each frame. After $t_0$, the 13.56MHz carrier frequency is modulated by the PICC at 847kHz for a minimum time of $t_1$ (see Table) to allow the CR14 to synchronize. After $t_1$, the first phase transition generated by the PICC represents the start bit ('0') of the Answer SOF (or the start bit '0' of the first data character in non SOF/EOF mode).
6.10 CRC

The 16-bit CRC used by the CR14 follows the ISO14443 type B recommendation. For further information, please see Appendix A on page 44.

The two CRC Bytes are present in all Request and Answer Frames, just before the EOF. The CRC is calculated on all the Bytes between the SOF and the CRC Bytes.

Upon transmission of a Request from the CR14, the PICC verifies that the CRC value is valid. If it is invalid, it discards the frame and does not answer the CR14.

Upon reception of an Answer from the PICC, the CR14 verifies that the CRC value is valid. If it is invalid, it stores the value FFh in the Input/Output Frame Register.

The CRC is transmitted Least Significant Byte first. Each Byte is transmitted Least Significant Bit first.

Figure 27. CRC transmission rules

<table>
<thead>
<tr>
<th>LSByte</th>
<th>MSByte</th>
<th>LSByte</th>
<th>MSByte</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC 16 (8 bits)</td>
<td>CRC 16 (8 bits)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7 Tag access using the CR14 coupler

In all the following I²C commands, the last three bits of the Device Select Code can be replaced by any of the three-bit binary values (000, 001, 010, 011, 100, 101, 110, 111). These values are linked to the logic levels applied to the E2, E1 and E0 pads of the CR14.

7.1 Standard TAG command access description

Standard PICC commands, like Read and Write, are generated by the CR14 using the Input/Output Frame Register.

When the host needs to send a standard frame command to the PICC, it first has to internally generate the complete frame, with the command code followed by the command parameters. Only the two CRC Bytes should not be generated, as the CR14 automatically adds them during the RF transmission.

When the frame is ready, the host has to write the request frame into the Input/Output Frame Register using the I²C write command specified in Figure 13 on page 23. After the I²C STOP condition, the CR14 inserts the I²C Bytes in the required ISO character format (Figure 20) and starts to transmit the request frame to the PICC. Once the RF transmission is over, the CR14 waits for the PICC to send an answer frame.

If the PICC answers, the characters received (Figure 26) are demodulated, decoded and stored into the Input/Output Frame Register, as specified in Table 4. During the entire RF transmission, the CR14 disconnects itself from the I²C bus. On reception of the PICC EOF, the CR14 checks the CRC and reconnects itself to the I²C bus.

The host can then get the PICC answer frame by issuing an Input/Output Frame Register Read on the I²C bus, as specified in Figures 14 and 15.

If no answer from the PICC is detected after a time-out delay, fixed in the Parameter Register (bits b5 and b6), the Input/Output Frame Register is set as specified in Table 4.
7.2 Anti-collision TAG sequence

The CR14 can identify an ST short range memory using a proprietary anti-collision system.

Issuing an I²C Write command to the Slot Marker Register (Figure 16) causes the CR14 TO automatically generate a 16-slot anti-collision sequence, and to store the identified Chip_ID in the Input/Output Frame Register, as specified in Table 4.

After receiving the Slot Marker Register I²C Write command, the CR14 generates an RF PCALL16 command followed by fifteen SLOT_MARKER commands, from SLOT_MARKER(1) to SLOT_MARKER(15). After each command, the CR14 waits for a tag answer. If the answer is correctly decoded, the corresponding Chip_ID is stored in the Input/Output Frame Register. If there is no answer, or if the answer is wrong (with a CRC error, for example), the CR14 stores an error code in the Input/Output Frame Register. At the end of the sequence, the host has to read the Input/Output Frame Register to retrieve all the identified Chip_IDs.
Figure 31. Anti-collision ST short range memory sequence (1)
Figure 32. Anti-collision ST short range memory sequence continued

<table>
<thead>
<tr>
<th>Slot</th>
<th>SOF</th>
<th>CRC</th>
<th>CRC</th>
<th>EOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>96h</td>
<td>CRC</td>
<td>CRC</td>
<td>EOF</td>
</tr>
<tr>
<td>11</td>
<td>56h</td>
<td>CRC</td>
<td>CRC</td>
<td>EOF</td>
</tr>
<tr>
<td>12</td>
<td>66h</td>
<td>CRC</td>
<td>CRC</td>
<td>EOF</td>
</tr>
<tr>
<td>13</td>
<td>76h</td>
<td>CRC</td>
<td>CRC</td>
<td>EOF</td>
</tr>
<tr>
<td>14</td>
<td>86h</td>
<td>CRC</td>
<td>CRC</td>
<td>EOF</td>
</tr>
<tr>
<td>15</td>
<td>96h</td>
<td>CRC</td>
<td>CRC</td>
<td>EOF</td>
</tr>
</tbody>
</table>
8 Maximum rating

Stressing the device above the rating listed in the Absolute Maximum Ratings table may cause permanent damage to the device. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.

Table 8. Absolute maximum ratings

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{\text{STG}} )</td>
<td>Storage Temperature</td>
<td>–65 to 150</td>
<td>°C</td>
</tr>
<tr>
<td>( V_{\text{IO}} )</td>
<td>Input or Output range (SDA)</td>
<td>–0.3 to 6.5</td>
<td>V</td>
</tr>
<tr>
<td>( V_{\text{IO}} )</td>
<td>Input or Output range (others pads)</td>
<td>–0.3 to ( V_{\text{CC}} + 0.3 )</td>
<td>V</td>
</tr>
<tr>
<td>( V_{\text{CC}} )</td>
<td>Supply Voltage</td>
<td>–0.3 to 6.5</td>
<td>V</td>
</tr>
<tr>
<td>( P_{\text{OUT}} )</td>
<td>Output Power on Antenna Output Driver (RF\text{OUT})</td>
<td>100</td>
<td>mW</td>
</tr>
<tr>
<td>( V_{\text{ESD}} )</td>
<td>Electrostatic Discharge Voltage (Human Body model) (^{1})</td>
<td>4000</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>Electrostatic Discharge Voltage (Machine model) (^{2})</td>
<td>500</td>
<td>V</td>
</tr>
</tbody>
</table>

1. MIL-STD-883C, 3015.7 (100 pF, 1500 \( \Omega \)).
2. EIAJ IC-121 (Condition C) (200 pF, 0 \( \Omega \)).
9  DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device. The parameters in the DC and AC Characteristic tables that follow are derived from tests performed under the Measurement Conditions summarized in the relevant tables. Designers should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

Table 9.  I²C AC measurement conditions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{CC}$ Supply Voltage</td>
<td>4.5</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td>Ambient Operating Temperature ($T_A$)</td>
<td>–20</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td>Input Rise and Fall Times</td>
<td>50</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Input Pulse Voltages</td>
<td>$0.2V_{CC}$</td>
<td>$0.8V_{CC}$</td>
<td>V</td>
</tr>
<tr>
<td>Input and Output Timing Reference Voltages</td>
<td>$0.3V_{CC}$</td>
<td>$0.7V_{CC}$</td>
<td>V</td>
</tr>
</tbody>
</table>

Figure 33.  I²C AC testing I/O waveform

![I²C AC testing I/O waveform](image)

Table 10.  I²C Input Parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{in}$</td>
<td>Input Capacitance (SDA)</td>
<td>8</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>$C_{in}$</td>
<td>Input Capacitance (SCL, E0, E1, E2))</td>
<td>6</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>$t_{NS}$</td>
<td>Low Pass Filter Input Time Constant (SCL &amp; SDA Inputs)</td>
<td>100</td>
<td>400</td>
<td>ns</td>
</tr>
</tbody>
</table>

1. Sampled only, not 100% tested.
2. $T_A = 25 \, ^\circ\text{C}$, $f = 400\text{kHz}$.

Table 11.  I²C DC characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Test condition</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{LI}$</td>
<td>Input Leakage Current (SCL, SDA, E0, E1, E2)</td>
<td>$0, V \leq V_{IN} \leq V_{CC}$</td>
<td>±2</td>
<td></td>
<td>µA</td>
</tr>
<tr>
<td>$I_{LO}$</td>
<td>Output Leakage Current (SCL, SDA, E0, E1, E2)</td>
<td>$0, V \leq V_{OUT} \leq V_{CC}$ in Hi-Z</td>
<td>±2</td>
<td></td>
<td>µA</td>
</tr>
</tbody>
</table>
Table 11. I²C DC characteristics (continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Test condition</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{CC}$</td>
<td>Supply Current</td>
<td>$V_{CC} = 5 \text{ V, } f_C = 400 \text{ kHz (rise/fall time &lt; 30ns), RF OFF}$</td>
<td>6 mA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{CC} = 5\text{ V, } f_C = 400 \text{ kHz (rise/fall time &lt; 30ns), RF ON}$</td>
<td>20 mA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$I_{CC1}$</td>
<td>Supply Current (Stand-by)</td>
<td>$V_{IN} = V_{SS}$ or $V_{CC}$, $V_{CC} = 5 \text{ V, RF OFF}$</td>
<td>5 mA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Input Low Voltage (SCL, SDA)</td>
<td>$-0.3 \text{ V to } 0.3V_{CC}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Input Low Voltage (E0, E1, E2)</td>
<td>$-0.3 \text{ V to } 0.3V_{CC}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>Input High Voltage (SCL, SDA)</td>
<td>$0.7V_{CC}$</td>
<td>$V_{CC} + 1 \text{ V}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Input High Voltage (E0, E1, E2)</td>
<td>$0.7V_{CC}$</td>
<td>$V_{CC} + 1 \text{ V}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Output Low Voltage (SDA)</td>
<td>$I_{OL} = 3 \text{ mA, } V_{CC} = 5 \text{ V}$</td>
<td>0.4 V</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 34. I²C AC waveforms
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Alt.</th>
<th>Parameter</th>
<th>Fast I²C 400 kHz</th>
<th>PC 100 kHz</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>$t_{CH1CH2}$ (1)</td>
<td>$t_R$</td>
<td>Clock Rise Time</td>
<td>300</td>
<td>1000 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{CL1CL2}$ (1)</td>
<td>$t_F$</td>
<td>Clock Fall Time</td>
<td>300</td>
<td>300 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{DH1DH2}$ (1)</td>
<td>$t_R$</td>
<td>SDA Rise Time</td>
<td>20</td>
<td>1000 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{DL1DL2}$ (1)</td>
<td>$t_F$</td>
<td>SDA Fall Time</td>
<td>20</td>
<td>300 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{CHDX}$ (2)</td>
<td>$t_{SU:STA}$</td>
<td>Clock High to Input Transition</td>
<td>600</td>
<td>4700 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{CHCL}$</td>
<td>$t_{HIGH}$</td>
<td>Clock Pulse Width High</td>
<td>600</td>
<td>4000 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{DLCI}$</td>
<td>$t_{HD:STA}$</td>
<td>Input Low to Clock Low (START)</td>
<td>600</td>
<td>4000 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{CLDX}$</td>
<td>$t_{HD:DAT}$</td>
<td>Clock Low to Input Transition</td>
<td>0</td>
<td>0 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{CLCH}$</td>
<td>$t_{LOW}$</td>
<td>Clock Pulse Width Low</td>
<td>1.3</td>
<td>4.7 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{DXCX}$</td>
<td>$t_{SU:DAT}$</td>
<td>Input Transition to Clock Transition</td>
<td>100</td>
<td>250 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{CHDH}$</td>
<td>$t_{SU:STO}$</td>
<td>Clock High to Input High (STOP)</td>
<td>600</td>
<td>4000 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{DHDL}$</td>
<td>$t_{BUF}$</td>
<td>Input High to Input Low (Bus Free)</td>
<td>1.3</td>
<td>4.7 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{CLQV}$</td>
<td>$t_{AA}$</td>
<td>Clock Low to Data Out Valid</td>
<td>1000</td>
<td>3500 ns</td>
<td></td>
</tr>
<tr>
<td>$t_{CLQX}$</td>
<td>$t_{DH}$</td>
<td>Data Out Hold Time After Clock Low</td>
<td>200</td>
<td>200 ns</td>
<td></td>
</tr>
<tr>
<td>$f_C$</td>
<td>$t_{SCL}$</td>
<td>Clock Frequency</td>
<td>400</td>
<td>100 kHz</td>
<td></td>
</tr>
</tbody>
</table>

1. Sampled only, not 100% tested.
2. For a reSTART condition, or following a write cycle.
Figure 35. CR14 synchronous timing

Table 13. RF OUT AC characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Condition</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{CC} )</td>
<td>External Oscillator Frequency</td>
<td>( V_{CC} = 5 ) V</td>
<td>13.553</td>
<td>13.567</td>
<td>MHz</td>
</tr>
<tr>
<td>MI(_{CARRIER} )</td>
<td>Carrier Modulation Index</td>
<td>( MI = (A-B)/(A+B) )</td>
<td>10</td>
<td>14</td>
<td>%</td>
</tr>
<tr>
<td>t(<em>{RFR}, t</em>{RFF} )</td>
<td>10% Rise and Fall time</td>
<td></td>
<td>0.5</td>
<td>1.5</td>
<td>( \mu )s</td>
</tr>
<tr>
<td>t(_{RFSBL} )</td>
<td>Pulse Width on RF(_{OUT} )</td>
<td>1 ETU = 128/( f_{CC} )</td>
<td>9.44</td>
<td></td>
<td>( \mu )s</td>
</tr>
<tr>
<td>t(_{JIT} )</td>
<td>ASK modulation bit jitter</td>
<td>CR14 to PICC</td>
<td>-0.5</td>
<td>0.5</td>
<td>( \mu )s</td>
</tr>
<tr>
<td>t(_{0} )</td>
<td>Antenna Reversal delay</td>
<td>Min = 64/( f_{S} )</td>
<td>75</td>
<td></td>
<td>( \mu )s</td>
</tr>
<tr>
<td>t(_{1} )</td>
<td>Synchronization delay</td>
<td>Min = 80/( f_{S} )</td>
<td>94</td>
<td></td>
<td>( \mu )s</td>
</tr>
</tbody>
</table>
### Table 13. RF<sub>OUT</sub> AC characteristics (continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Condition</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{WDG} )</td>
<td>Answer delay watchdog ((b_5=0, \ b_6=0))</td>
<td>Request EOF rising edge to first Answer start bit</td>
<td>500</td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{WDG} )</td>
<td>Answer delay watchdog ((b_5=0, \ b_6=1))</td>
<td></td>
<td>5</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>( t_{WDG} )</td>
<td>Answer delay watchdog ((b_5=1, \ b_6=0))</td>
<td></td>
<td>10</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>( t_{WDG} )</td>
<td>Answer delay watchdog ((b_5=1, \ b_6=1))</td>
<td></td>
<td>309</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>( t_{DR} )</td>
<td>Time Between Request characters</td>
<td>CR14 to PICC</td>
<td>9.44</td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( P_A )</td>
<td>RF&lt;sub&gt;OUT&lt;/sub&gt; output power</td>
<td></td>
<td>90</td>
<td></td>
<td>mW</td>
</tr>
<tr>
<td>( t_{POR} )</td>
<td>CR14 Power-On delay</td>
<td></td>
<td>20</td>
<td></td>
<td>ms</td>
</tr>
</tbody>
</table>

1. Data specified in the table above are estimated or target values. All values can be updated during product qualification.

### Table 14. RF<sub>IN</sub> AC characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter&lt;sup&gt;(1)&lt;/sup&gt;</th>
<th>Condition</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{RFSBL} )</td>
<td>PICC Pulse Width</td>
<td>( 1 ) ETU = ( 128/f_{CC} )</td>
<td>9.44</td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( f_S )</td>
<td>PICC Sub-carrier Frequency</td>
<td>( f_{CC}/16 )</td>
<td>847.5</td>
<td></td>
<td>KHz</td>
</tr>
<tr>
<td>( t_{DA} )</td>
<td>Time Between Answer characters</td>
<td>PICC to CR14</td>
<td></td>
<td>1, 2, 3</td>
<td>ETU</td>
</tr>
<tr>
<td>( V_{DYN} )</td>
<td>RF&lt;sub&gt;IN&lt;/sub&gt; Dynamic Voltage Level</td>
<td>( V_{DYN} ) Max for ( V_{OFFSET} = V_{CC}/2 )</td>
<td>0.5</td>
<td>( V_{CC}/2 )</td>
<td>V</td>
</tr>
<tr>
<td>( V_{OFFSET} )</td>
<td>RF&lt;sub&gt;IN&lt;/sub&gt; Offset Voltage Level</td>
<td></td>
<td>2</td>
<td>3</td>
<td>V</td>
</tr>
<tr>
<td>( V_{RET} )</td>
<td>RF&lt;sub&gt;IN&lt;/sub&gt; Retro-modulation Level</td>
<td></td>
<td>120</td>
<td></td>
<td>mV</td>
</tr>
</tbody>
</table>

1. Data specified in the table above are estimated or target values. All values can be updated during product qualification.
10 Package mechanical

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: www.st.com. ECOPACK® is an ST trademark.
Figure 36. SO16 narrow - 16 lead plastic small outline, 150 mils body width, Package outline

Table 15. SO16 narrow - 16 lead plastic small outline, 150 mils body width, package mechanical data

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Millimeters</th>
<th>Inches</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.75</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>0.1</td>
<td>0.25</td>
</tr>
<tr>
<td>A2</td>
<td>1.25</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>0.31</td>
<td>0.51</td>
</tr>
<tr>
<td>c</td>
<td>0.17</td>
<td>0.25</td>
</tr>
<tr>
<td>D</td>
<td>9.9</td>
<td>9.8</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>5.8</td>
</tr>
<tr>
<td>E1</td>
<td>3.9</td>
<td>3.8</td>
</tr>
<tr>
<td>e</td>
<td>1.27</td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>L</td>
<td>0.4</td>
<td>1.27</td>
</tr>
<tr>
<td>k</td>
<td>0°</td>
<td>8°</td>
</tr>
<tr>
<td>Tolerance</td>
<td>millimeters</td>
<td>inches</td>
</tr>
<tr>
<td>ccc</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>
11 Ordering information

Table 16. Ordering information scheme

<table>
<thead>
<tr>
<th>Device type</th>
<th>Package</th>
<th>Customer code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR14</td>
<td>MQ = SO16 Narrow (150 mils width)</td>
<td>XXX = Given by the issuer</td>
</tr>
<tr>
<td></td>
<td>MQP = SO16 Narrow (150 mils width) ECOPACK®</td>
<td></td>
</tr>
</tbody>
</table>

Example: CR14 – MQ / XXX

For a list of available options (speed, package, etc.) or for further information on any aspect of this device, please contact your nearest ST Sales Office.
Appendix A  ISO14443 type B CRC calculation

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#define BYTE unsigned char
#define USHORT unsigned short

unsigned short UpdateCrc(BYTE ch, USHORT *lpwCrc)
{
    ch = (ch^(BYTE)((*lpwCrc) & 0x00FF));
    ch = (ch^(ch<<4));
    *lpwCrc = (*lpwCrc >> 8)^((USHORT)ch << 8)^((USHORT)ch<<3)^((USHORT)ch>>4);
    return(*lpwCrc);
}

void ComputeCrc(char *Data, int Length, BYTE *TransmitFirst, BYTE *TransmitSecond)
{
    BYTE chBlock; USHORT wCrc;
    wCrc = 0xFFFF; // ISO 3309
    do
    {
        chBlock = *Data++;
        UpdateCrc(chBlock, &wCrc);
    } while (--Length);
    wCrc = ~wCrc; // ISO 3309
    *TransmitFirst = (BYTE) (wCrc & 0xFF);
    *TransmitSecond = (BYTE) ((wCrc >> 8) & 0xFF);
    return;
}

int main(void)
{
    BYTE BuffCRC_B[10] = {0x0A, 0x12, 0x34, 0x56}, First, Second, i;
    printf("Crc-16 G(x) = x^16 + x^12 + x^5 + 1");
printf("CRC_B of [ ");
for(i=0; i<4; i++)
    printf("%02X ",BuffCRC_B[i]);
ComputeCrc(BuffCRC_B, 4, &First, &Second);
printf("] Transmitted: %02X then %02X.", First, Second);
return(0);
}
## Revision history

### Table 17. Document revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-Dec-2005</td>
<td>1</td>
<td>Initial release.</td>
</tr>
<tr>
<td>19-Mar-2010</td>
<td>2</td>
<td>Updated <em>Figure 36</em> and <em>Table 15 on page 42</em></td>
</tr>
</tbody>
</table>