



RFID library for SR176 tag management with STR71x

Introduction

This document describes a software and hardware RFID interface using the STR71x I²C peripheral and the CRX14 RFID transceiver.

The main purpose of this software and hardware package is to provide resources to assist the development of an application using an RFID tag.

The software interface is composed of library source files, include files and some application template source files.

Contents

1	File organization of the RFID library	3
2	Overview of radio frequency identification	4
3	Operating ranges and applications	5
3.1	Close coupling	5
3.2	Short range	5
3.3	Long range	5
3.4	Extended or very long range	5
4	ISO standards	6
4.1	ISO compatibility for ST devices	6
5	ST short-range contactless memories	7
5.1	SR176	7
5.2	SR1X4K	7
5.3	SR1X512	7
6	ST short range coupler: CRX14	8
7	Overview of RFID reader using STR710	9
7.1	CRX14 demonstration kit	9
7.2	Interfacing with STR7xx using I2C peripheral	12
8	RFID library	14
8.1	RFID reader/writer: the application layer	14
8.2	SR176 commands layer	17
8.3	CRX14 commands layer	17
8.4	I2C commands layer	20
9	Revision history	21

1 File organization of the RFID library

The following table presents the library modules:

Table 1. RFID library modules

File	Description
<i>Device_RFID.h</i>	RFID definitions, type definitions and function prototypes
<i>Device_RFID.c</i>	Protocol management to interface the CRX14 transceiver

2 Overview of radio frequency identification

Countless business applications, from product manufacturing to end-customer sales, can benefit from radio frequency identification (RFID). With RFID, logistics becomes more effective, stock handling is easier, traceability is guaranteed, cloning and counterfeiting can be avoided, antitheft protection is enabled, and warranty problems can be easily eliminated.

What is radio frequency identification (RFID)?

RFID technology uses radio communications to uniquely identify items, such as goods, people, or animals. It enables automated collection of information on identified items independently from their position, without the need of a direct optical link or human intervention. RFID also liberates the user from harsh or dirty environment concerns that restrict other automatic identification solutions such as barcodes. On top of that, RFID is used as a bi-directional data carrier since it allows information attached to the item to be written and updated on the fly.

What does a RFID system consist of?

An RFID system consists of several components including one or multiple contactless memory tags (or transponders), attached to the items to be labeled, identified, and traced, and handheld or fixed contactless read/write reader station units or coupling devices exchanging data with the transponder(s) by means of RF signals.

Contactless memory tags

Contactless memory tags (transponders) are the backbone of the RFID system since they contain data that allow the items to be identified. A contactless memory tag is made of a non-volatile memory associated with a radio frequency communication block, performing RF signal modulation, demodulation, power supply regulation, and an antenna. A transponder can be read-only (ROM), one-time-programmable (OTP), or read/write (EEPROM), and it can be embedded in various package formats (inlays, cards, etc.).

The reader

The **reader** (or reader station unit) is connected to an antenna that transmits and receives the radio frequency signal from and to the transponder(s). The reader sends an RF signal to the transponder's antenna. This signal generates a voltage that is rectified and provides a power source to the transponder. The power transfer from the reader to the transponder is performed through an inductive coupling between the two coils in 125 kHz and 13.56 MHz systems. For ultra-high frequencies (UHF), the electric field generated by the reader is used to power the tag. In other words, the transponder does not need a dedicated power supply.

Passive and active tags

Two main families of transponders exist: Passive (battery-less) and Active. Adding a battery can bring several advantages, such as improvement of the reading range or continuous monitoring of temperature and pressure etc., but adds an extra cost. During the reader-transponder communication, the RF signal generated by the reader is modulated according to the data to be sent. In the transponder-reader direction, no modulation is applied on the reader RF signal and the communication is done through load modulation in inductive coupling systems. The reader interfaces with the RFID system control host through a serial interface such as RS232, RS422/485, or USB.

3 Operating ranges and applications

The RFID operating ranges are defined by the distance between the transponder(s) and the reader.

3.1 Close coupling

The **close-coupling** range includes the transponder-reader distances up to 5 cm. Brand protection, anti-counterfeiting of electronic goods and food consumables, as well as electronic purchases at vending machines operate in the close-coupling range. These applications employ low-cost readers operating with a limited number of transponders and require security features such as password, anti-clone, or cryptographic capabilities.

3.2 Short range

In addition, RF technology systems are a universal solution for various ticketing applications, regardless of the means of transport used (public transportation or private car). Contactless systems can also be used in a wide range of access-control applications, including entrance systems to and within public buildings (in offices, for example), time and attendance systems, room logistics at hotels, locking systems, and amusement parks. These **short-range** or proximity-range applications use low-cost transponders embedded in plastic cards or paper tickets and operate in the range up to 20 cm. Readers and transponders from different suppliers must be fully interoperable. Required security levels are satisfied through password, anti-clone, or cryptographic capabilities.

3.3 Long range

RF transponders are a low-cost universal solution for tracking a wide variety of valuable assets (e.g. package delivery, animal tracking) and improving the product supply chain management from manufacturing to stock handling to distribution. Capable of storing traceability history, transponders provide accurate real-time information on each tagged item. Similarly, they can be used for anti-theft applications (using Electric Article Surveillance) or access control from a longer distance.

These long-range or vicinity-range applications operate within 1 m from the reader and typically involve a large number of low-cost transponders (100+), requiring reliable anti-collision mechanisms and transponder-reader interoperability.

3.4 Extended or very long range

Extended or very long-range (up to 10 meters) RFID systems will be instrumental in creating low-cost contactless solutions for new supply chains, logistics, and tracking applications that can offer improved efficiency, accuracy, and security standards to major manufacturers, retailers, and their customers.

4 ISO standards

The ISO organization devotes itself to definition and promotion of worldwide-recognized standards in the contactless area. 3 standards for objects operating in the 13.56 MHz frequency range have already been approved (ISO SC17 on identification cards and related devices):

- ISO10536 identification cards and contactless integrated circuits cards:
 - Close-coupling cards (reading range below 2 cm).
- ISO 14443 identification cards and contactless integrated circuits cards:
 - Proximity cards (reading range up to 20 cm).
- ISO 15693 identification cards and contactless integrated circuits cards:
 - Vicinity cards (reading range up to 1 m).

This standard is part of the future SC31/ISO18000 standard, currently under discussion.

ISO standards are recognized worldwide and guarantee the full interoperability of transponders and readers from different suppliers.

4.1 ISO compatibility for ST devices

ST contactless memories are ISO-compatible:

- ISO 14443 type B for short-range devices
- ISO 15693 or ISO18000-3 Mode 1 for long-range devices.

This application note refers to ST ISO 14443B short range RFID.

5 ST short-range contactless memories

5.1 SR176

ST's SR176 is an ISO14443 Type B device. It offers 176 bits of EEPROM organized in eleven 16-bit blocks, a 64-bit UID and an 8-bit Chip_ID. Targeted markets are cost-sensitive applications such as industrial identification, access control, and low-value tickets that do not need to be modified more than a few times.

5.2 SRIX4K

ST's SRIX4K includes 4096 bits of non-volatile memory organized in three areas: the first area is an OTP (One Time Programmable) bit zone, in which bits can only be switched from 1 to 0; the second area provides two 32-bit binary down counters; and the last area which is a non-volatile memory split into 32-bit blocks (9 blocks are lockable). The SRIX4K supports France Telecom's proprietary anti-clone function and anti-collision mechanism based on an 8-bit Chip_ID.

The SRI4K is based on the SRIX4K, excluding the support of France Telecom's proprietary anti-clone function.

5.3 SRIX512

The SRIX512 is a light version of the SRIX4K with 512 bits of non-volatile memory. All three devices (SRIX4K, SRI4K, SRIX512) are designed for short-range applications such as consumable products identification, anti-counterfeiting, or high-end ticketing solutions.

6 ST short range coupler: CRX14

The CRX14 is a short-range contactless chip coupler compliant with the ISO14443 type B standard. It interfaces with the host CPU that manages the communications protocol in both directions through the industry-standard I²C serial bus. The coupler is equipped with France Telecom's proprietary anti-clone function and anti-collision mechanism. The CRX14 is supplied in SO16N packages.

A complete solution combining the CRX14 coupler and SRxx transponders operates in the short range and ensures a total compatibility in terms of memory tag protocol access. In addition, the combination of the CRX14 and SRIX512 or SRIX4K memories guarantees a high level of system security for short-range and low-cost applications. Optimized for interoperability, these products can also work independently, with any ISO-standard chips.

The CR14 is based on the CRX14, excluding the support of France Telecom's proprietary anti-clone function.

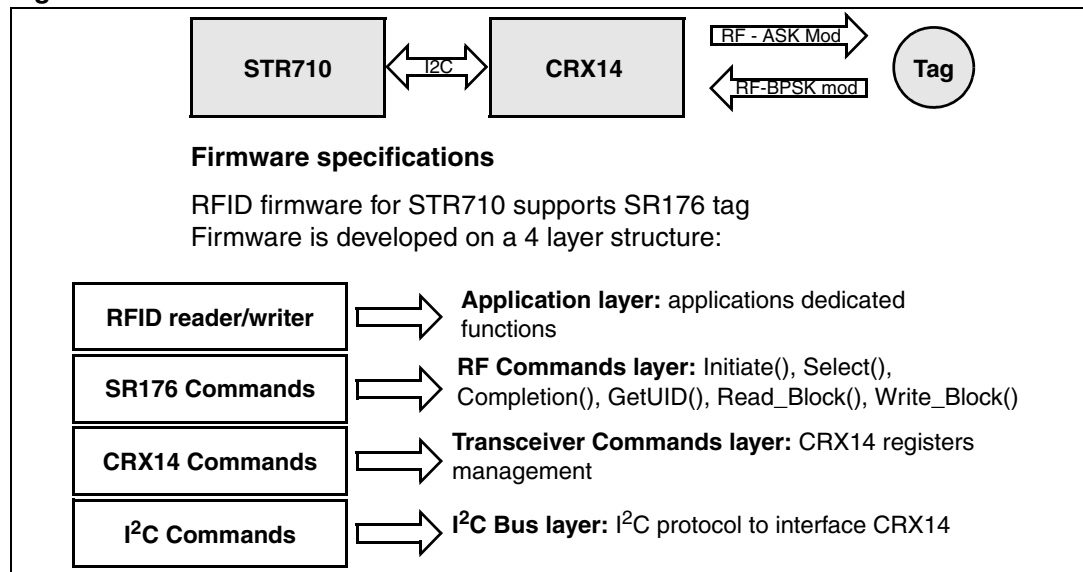
7 Overview of RFID reader using STR710

The RFID reader is developed using the STR710 ARM7TDMI powered microcontroller and the CRX14 low cost contactless coupler chip. Communication between the microcontroller and the coupler is achieved by an I²C bus.

The RFID library has been developed to support **SR176 tag**.

As depicted in picture below, the RFID library has been developed on a 4-layer structure. Using the application layer, the tag may be used as an EEPROM with read/write operation. It is not necessary to know how STR710 communicates with the CRX14 coupler chip and how CRX14 communicates with SR176 tag.

Figure 1. Architecture details



7.1 CRX14 demonstration kit

A demonstration kit has been created to interface CRX14 with the STR710 microcontroller via I²C. Further details can be found in the application note *AN1806*. The demonstration kit schematic and layout are shown below.

Figure 2. CRX14 demonstration kit schematic

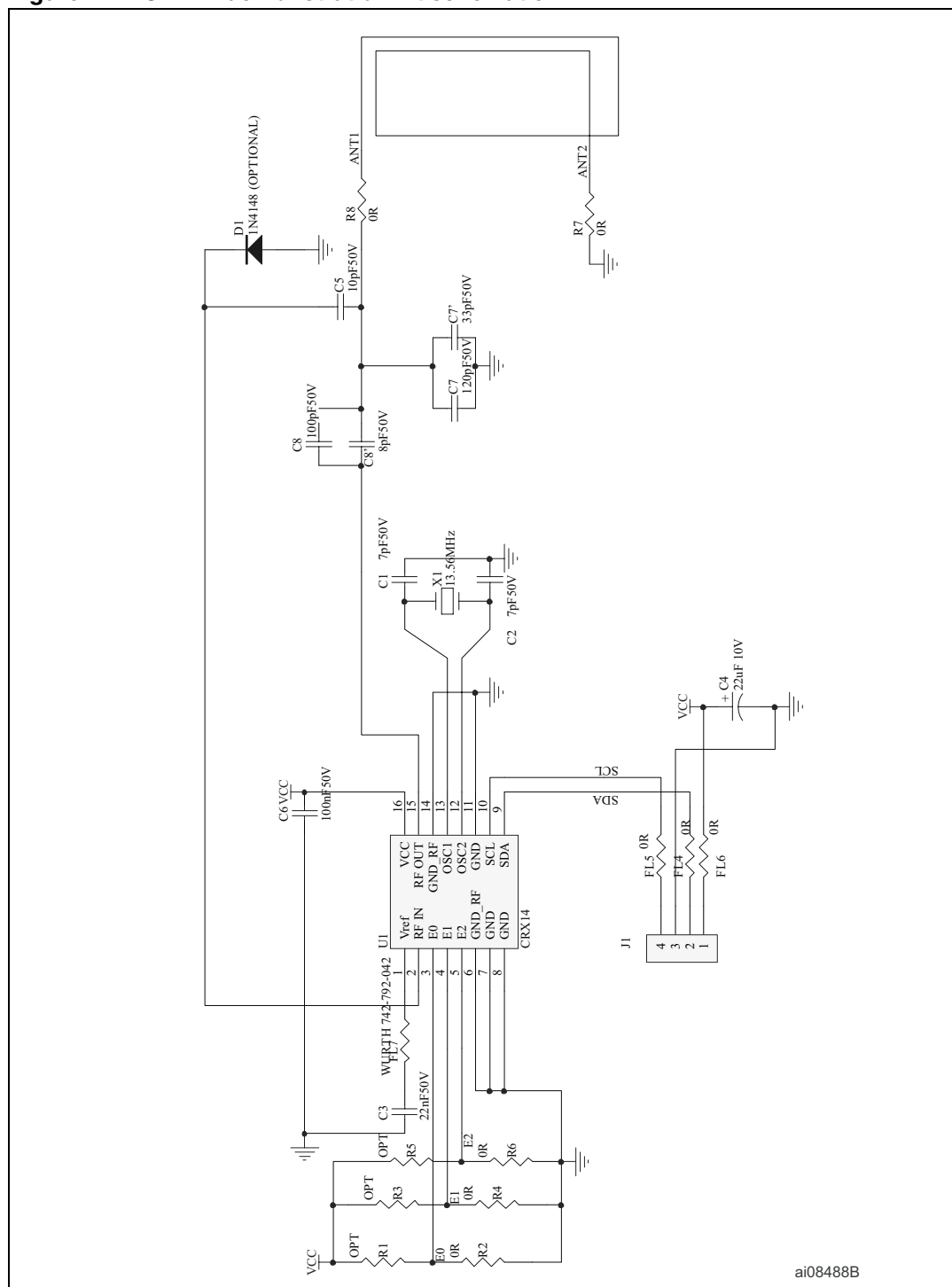
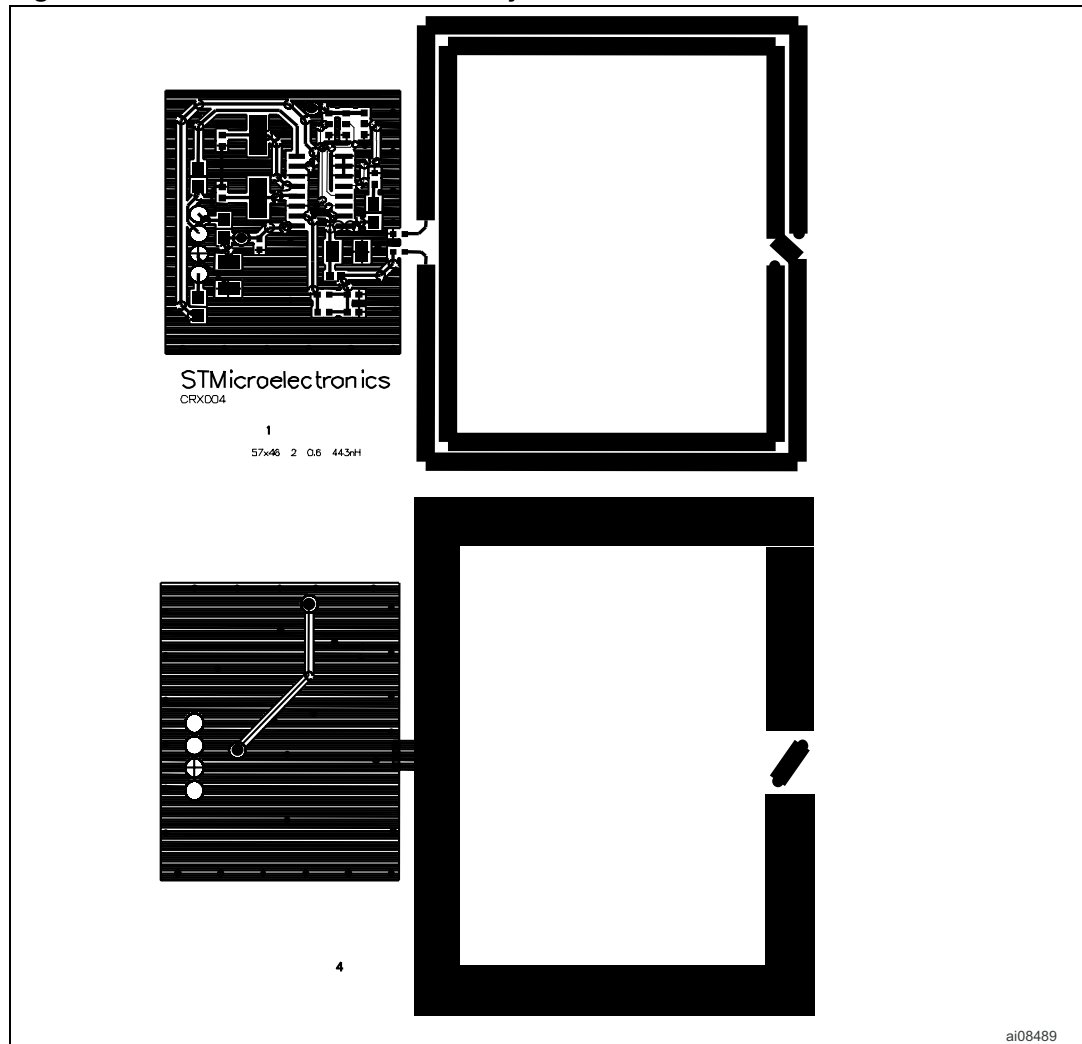


Figure 3. CRX14 demonstration kit layout



ST has designed the CRX14, a short range contactless coupler chip, compliant with the ISO14443 type B proximity standard. The CRX14 generates a 13.56 MHz signal. Designed to deliver an RF power of 100 mW, it operates in the Short Range on contactless memory tags (provided that they, too, are compliant with ISO14443 type B).

The CRX14 features the ST anti-collision mechanism, which allows the reader to detect and identify all the tags that are present in the operating range, and to access them individually. Because the CRX14 implements the France Telecom-proprietary anti-clone function, the reader can also perform authentication of tags that are equipped with the France Telecom anti-clone capability.

The CRX14 coupler interfaces between:

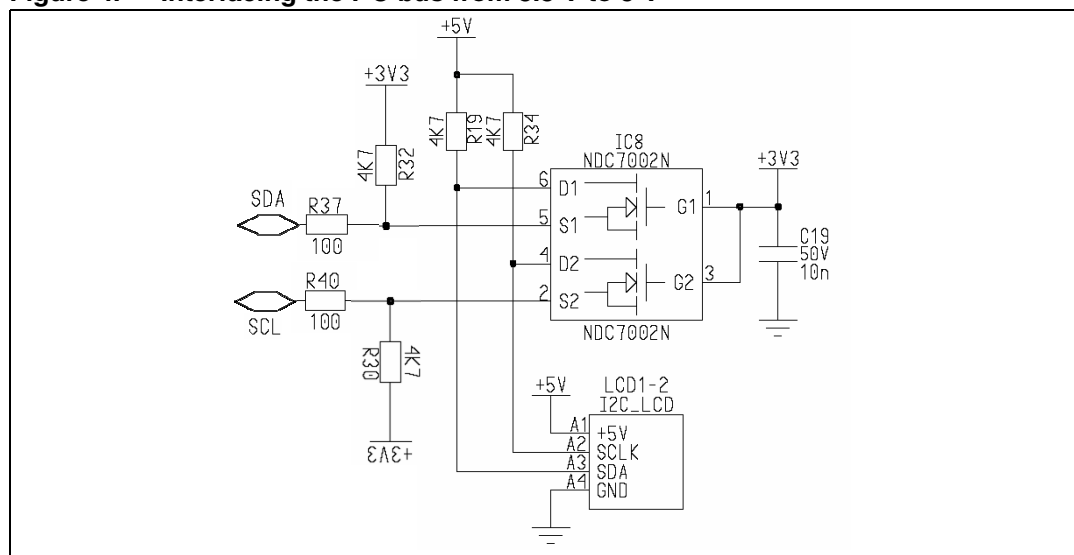
- the memory tags, on one side, through input/output buffers and the ISO14443 type B radio frequency protocol, and
- the system master processor, on the other side, through a 400 kHz I²C bus.

Operating from a 5 V power supply, and delivered in a SO16N package, the CRX14 coupler chip is an excellent solution for building contactless readers, embedded in the final equipment, and offering a good compromise between operating range and cost.

7.2 Interfacing with STR7xx using I2C peripheral

Due to the fact that the STR7xx microcontroller has a 3.3 voltage supply and the CRX14 device work at 5 V, a level shifter has to be inserted on the I²C bus in order to adjust the voltage levels.

Figure 4. Interfacing the I²C bus from 3.3 V to 5 V



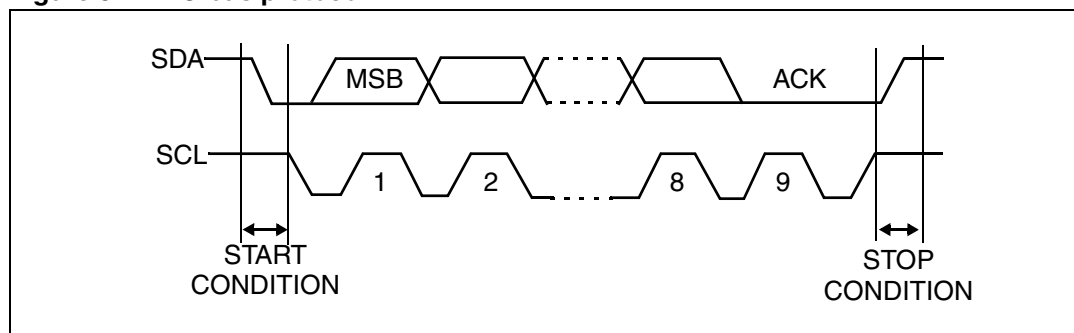
The I²C Bus Interface on STR71x provides both multimaster and slave functions, and controls all I²C bus-specific sequencing, protocol, arbitration and timing. It also supports fast I²C mode (400 kHz).

For the purpose of this application note, the I²C peripheral works as Master transmitter/receiver. In Master mode, it initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated in master mode by hardware as soon as the Master mode is selected.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the start condition contain the address (one in 7-bit mode, two in 10-bit mode).

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to [Figure 5](#).

Figure 5. I²C bus protocol



Finally, I²C peripheral configuration has to be set as follows:

- data width: 8-bit
- Ack bit: enabled
- Speed config: the user must configure it via *I2C_frequency* define in *device_RFID.h* file.

```
void init_CRX14_I2C(void){
    u32 f_RFID=I2C_frequency;

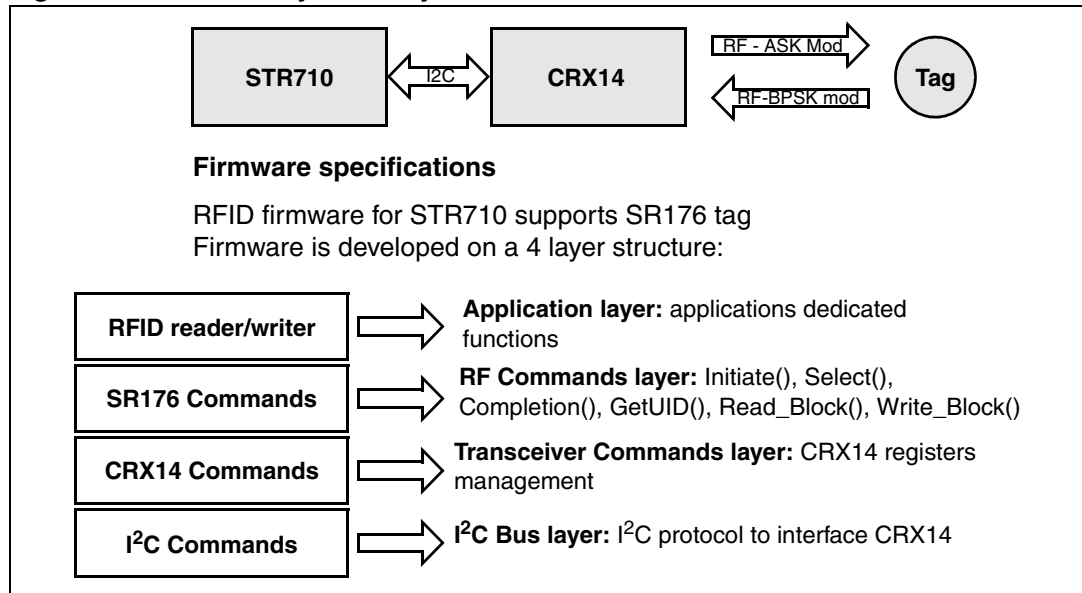
    /*-----
    -----
    Configure the SDA And the SCL pin of the I2C0 and I2C1 to alternate function Open
    Drain
    -----
    -----*/
    GPIO_Config (GPIO1,I2C0_SCL,GPIO_AF_OD);
    GPIO_Config (GPIO1,I2C0_SDA,GPIO_AF_OD);

    /*-----
    -----
    Configure I2C0 module
    -----
    -----*/
    I2C_Init (I2C0);
    I2C_FCLKConfig (I2C0);
    I2C_OnOffConfig (I2C0, ENABLE);//Enable the I2C0
    I2C_SpeedConfig (I2C0, f_RFID);//Configure the I2C0 speed
    I2C_AddressConfig (I2C0,0xA2, I2C_Mode7);
    //Configure the I2C0 address (to be used specially when this cell has to be adressed)
    I2C_AcknowledgeConfig (I2C0, ENABLE);
    //set str7 to send ack when receive data
}
```

8 RFID library

RFID library is developed using a 4-layer architecture. This section provides a brief description of the layer functionality.

Figure 6. RFID library: the 4 layer structure



8.1 RFID reader/writer: the application layer

The RFID EEPROM memory may be accessed directly using the application layer. It allows read/write operations on RFID EEPROM with different access options. Moreover, the application layer implements the Finite State Machine (FSM) which allows each device to be searched for and selected close to the antenna reader.

```

/*-----
RFID Application Layer
-----*/

Input:
- RFID_Action: 3 bit
    bit 0: 1=ENABLE 64UID request, 0=DISABLE 64UID request
    bit 1: 1=ENABLE READ request, 0=DISABLE READ request
    bit 2: 1=ENABLE WRITE request, 0=DISABLE WRITE request
- TX_frame: TX buffer pointer (write operations)
- TX_lenght: TX buffer lenght (write operations)
- TX_offset: TX offset for write operations
- RX_frame: RX buffer pointer (read operations)
- RX_lenght: RX buffer lenght (read operations)
- RX_offset: RX offset for read operations
- RFID_UID: 64 bits RFID UID
- RFID_timeout: timeout for RFID polling (number of times STR7 is trying to search tag)

Output:
- RX_frame: RX buffer pointer (read operations)
- RFID_UID: 64 bits RFID UID

Return:
- Status:

```

```
bit0: 1-No tag selected; 0-tag selected
bit1: 1-Error on 64UID; 0-no error
bit2: 1-Error on READ; 0-no error
bit3: 1-Error on WRITE; 0-no error
bit4: 1-Error on Completion; 0-no error
bit5: 1-Error on RF_OFF; -no error
-----*/
RFID_status = RFID_Handler(RFID_Action,
&TX_frame[0],
TX_lenght,
TX_offset,
&RX_frame[0],
RX_lenght,
RX_offset,
&RFID_UID[0],
RFID_Polling_timeout);
```

8.1.1 RFID actions

Three main actions are available:

- Read: RFID EEPROM
- Write: RFID EEPROM
- Get 64-bit UID (unique identifier)

The user can choose actions to perform on the device using the **RFID_Action** parameter: 3 bits allow each action to be chosen separately.

8.1.2 RFID memory access

Two different ways to access RFID EEPROM in read/write mode are available. The user can choose to:

- access (read/write) all 10 words (20 bytes) available on SR176 tag starting from base address 0x0;
- access (read/write) a portion of memory (one or more words) starting from an offset address.

The following must be provided by the user:

- a pointer related to the buffer he want to read/write;
- the buffer length;
- the offset where he want to start to read/write.

8.1.3 RFID polling

The user can choose how many times to search for a tag device close to the antenna reader using the **RFID_Polling_timeout** parameter.

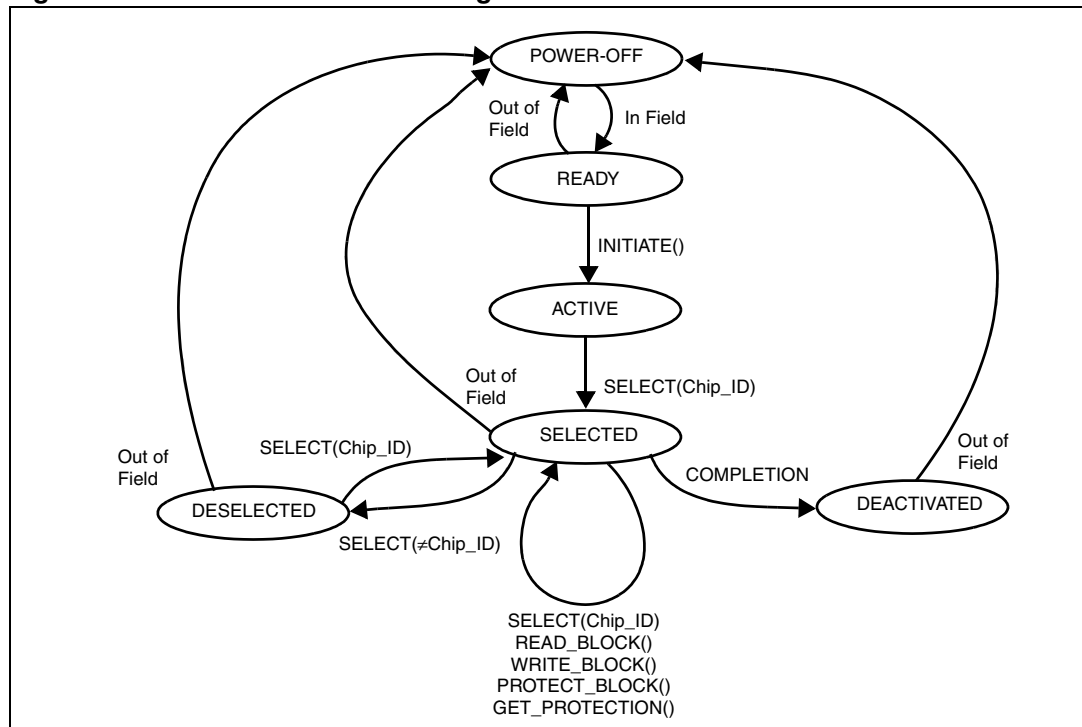
8.1.4 RFID communication status report

The communication status can be checked using the **RFID_Handler** return parameter. It returns:

- Presence/absence of tag into the space covered from magnetic field.
- Error on reading/writing RFID EEPROM
- Error getting 64 bits UID
- Error on cut-off RF carrier.

In order to access an RFID EEPROM, the application layer function has to implement a Finite State Machine (FSM) as depicted in the following picture:

Figure 7. RFID: state transition diagram



When invoking the application layer, FSM searches for a tag (for a period of time specified in **RFID_Polling_timeout**); when a tag device is found, the FSM attempts to recover the tag *CHIP_ID* using the “*Initiate()*” command. If the tag responds with its own *CHIP_ID* the FSM attempts to select it in order to perform actions according the *RFID_Action* parameter.

The FSM performs these actions using the “*init_RFID()*”. It describes all states from POWER OFF to SELECTED.

When selected, the application layer FSM uses three functions to perform the requested actions:

- `GetSR176_64UID(&RFID_UID[0]);`
- `ReadSR176_EEPROM(&RX_frame[0],RX_offset,RX_lenght);`
- `WriteSR176_EEPROM(&TX_frame[0],TX_offset,TX_lenght);`

When all requested actions are performed, the FSM deselects the tag device via a “*Completion()*” command.

Every time the RFID_Handler is invoked it powers the antenna with the 13.56 MHz carrier; before exiting, it cuts-off the RF carrier.

8.2 SR176 commands layer

In order to manage the RFID tag, the CRX14 uses a set of commands. They are both used to activate and select each device and to access (read/write or get 64 bits UID) the tag EEPROM.

Here are all the functions used for reported RFID tags:

- RF_ON(); // power on carrier frequency
- RF_OFF(); // power off carrier frequency
- Initiate(&RX_frame[0]); // search a tag in the field
- Select (RFID_CHIPID,&RX_frame[0]); // activate a tag in the field
- Completion (&RX_framebuf[0]); // deselect tag
- Time_Read(); // program time for read
- ReadBlock (address, &RX_frame[0]); // read a word
- Time_Write(); // program time for write
- WriteBlock (address, &TX_frame[0], &RX_frame[0]); // write a word

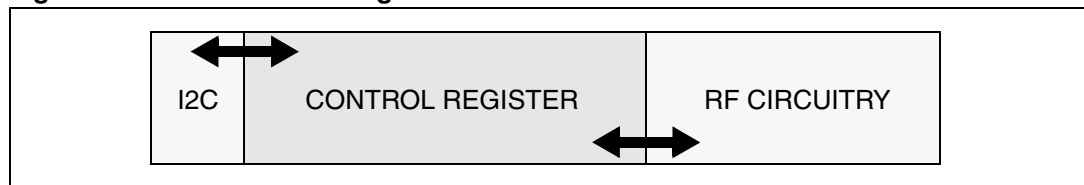
Each RF command has a dedicated 16-bit code (see SR176 and CRX14 datasheet). Tags in the field recognize it and perform the required operation (activation, read, write, deactivation, etc.).

RF Commands functions use CRX14 commands to access CRX14 registers. Basically each RF command performs a write operation and a read operation to a CRX14 register.

8.3 CRX14 commands layer

This layer provides all functions needed to interface the STR710 microcontroller with the CRX14 coupler device. CRX14 device uses 6 control registers to communicate both with the microcontroller (via I²C) and the tag (via RF circuitry).

Figure 8. CRX14 control register access scheme

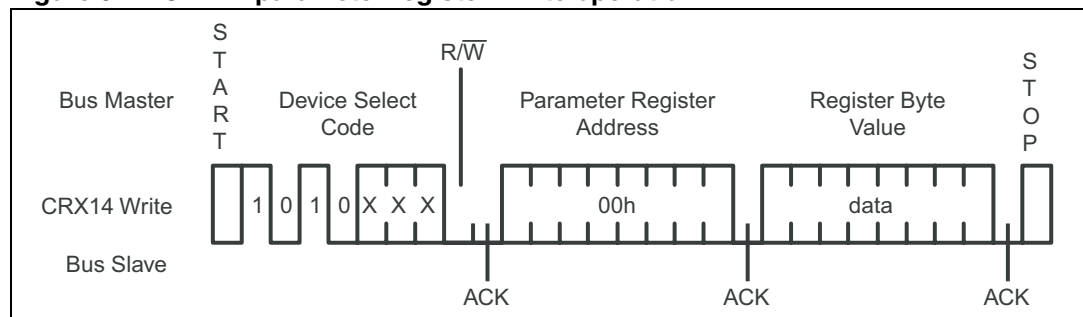
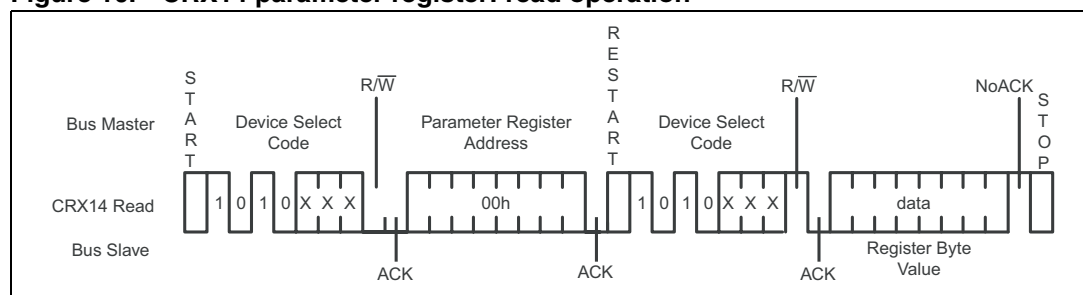


Depending on which register the user would like to access in read or write mode, several functions are implemented.

Table 2. Control registers

Address	Register	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 process
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

8.3.1 CRX14 parameter register access

Figure 9. CRX14 parameter register: write operation**Figure 10. CRX14 parameter register: read operation**

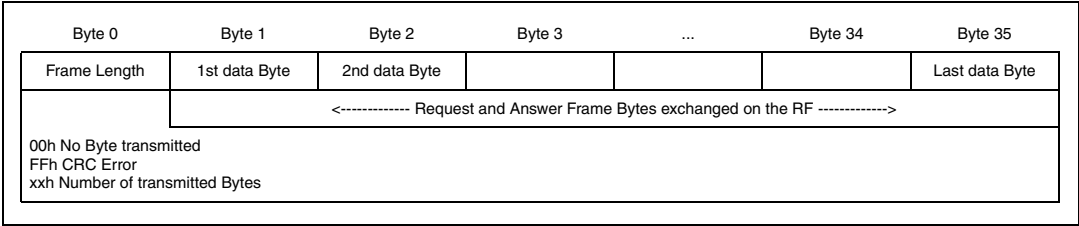
Here are the functions used for parameter register access:

```
int Write_CRX14_PR(UCHAR *DATA)
int Read_CRX14_PR(UCHAR *DATA)
```

8.3.2 CRX14 I/O frame register access

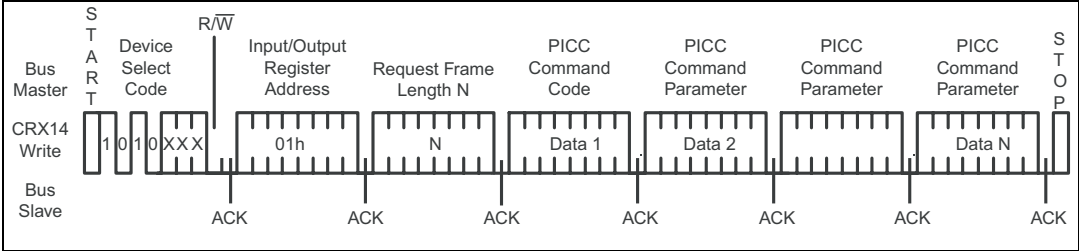
The Input/Output Frame Register is a 36-byte buffer that is accessed serially from Byte 0 through to Byte 35 (see [Figure 11](#)). It is located at the I²C address 01h. The Input/Output Frame Register is the buffer in which the CRX14 stores the data Bytes of the request frame to be sent to the PICC. 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.

Figure 11. CRX14 I/O frame register description



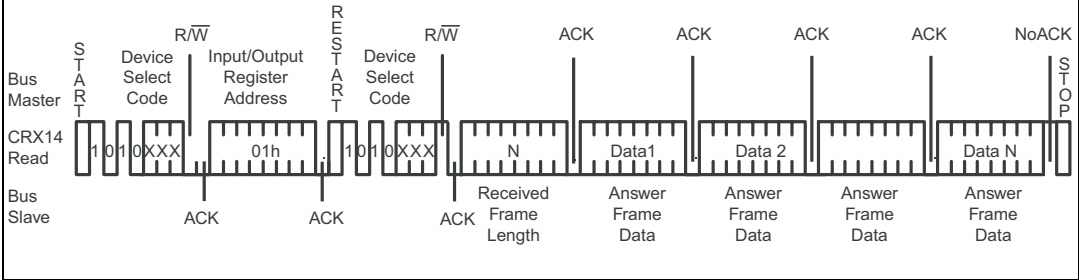
When accessed in I²C Write mode, the register stores the request frame Bytes that are to be transmitted to the PICC. Byte 0 must be set with the request frame length (in Bytes) and the frame is stored from Byte 1 onwards. At the end of the transmission, the 16-bit CRC is automatically added. After the transmission, CRX14 waits 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.

Figure 12. CRX14 I/O frame register: write operation



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.

Figure 13. CRX14 I/O frame register: read operation



Here are the functions used for I/O Frame register access:

```
int Write_CRX14_IOFR(UCHAR *DATA)
int Read_CRX14_IOFR(UCHAR *DATA)
```

8.4 I²C commands layer

The I²C layer provides all the functionalities to access and manage the I²C peripheral on STR710. It is a part of the STR710 library. Some functionality was added in order to receive a status when data is sent or received on the I²C bus; two programmable timeouts (I2C_TIMEOUT_SEND and I2C_TIMEOUT_RECEIVE) inform the application layer about any send/receive failure on I²C bus.

Below the code added for I2C_ByteSend and I2C_ByteReceive functions.

8.4.1 I2C_ByteSend_with_status function

```
/******
 * Function Name : I2C_ByteSend
 * Description : Send a single byte of data.
 * Input : I2Cx ( I2C0 or I2C1 )
 * Data : the byte to be sent to the slave
 * Return : status: 0-byte sent, 1-transmission error.
 *****/
u8 I2C_ByteSend_with_status (I2C_TypeDef *I2Cx, u8 Data)
{
    u8 status=0;
    u32 timeout=I2C_TIMEOUT_SEND;
    //Wait until the I2C_BTF bit is set
    while (((I2Cx->SR1 & 0x08)==0)&&timeout--);
    if(!timeout) status=1;
    //Write in the DR register the byte to be sent
    I2Cx->DR = Data;
    return status;
}
```

8.4.2 I2C_ByteReceive_with_status function

```
/******
 * Function Name : I2C_ByteReceive
 * Description : Returns the received byte.
 * Input : I2Cx ( I2C0 or I2C1 )
 * Output : Data ( variable where store read data )
 * Return : status: 0-byte sent, 1-transmission error.
 *****/
u8 I2C_ByteReceive_with_status (I2C_TypeDef *I2Cx, u8 *Data)
{
    u8 status=0;
    u32 timeout=I2C_TIMEOUT_RECEIVE;
    //Wait till I2C_BTF bit is set
    while (((I2Cx->SR1 & 0x08)==0)&&timeout--);
    if(!timeout) status=1;
    (*Data)=I2Cx->DR;
    return status;
}
```

9 Revision history

Table 3. Document revision history

Date	Revision	Changes
31-Jan-2008	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2008 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com