AN3255
Application note

Building an RFID short-range reader using the STM8S-DISCOVERY

Application overview

High frequency 13,56 MHz RFID solutions offer ideal close-proximity identification for product authentication, parcel tracking, document management, library and ticketing applications.

This application note describes how to build an RFID (radio frequency identification) short-range reader using STMicroelectronics STM8S-DISCOVERY and ISO14443 type-B CR14 contactless coupler. The STM8S-DISCOVERY and the CR14 communicate through an I^2C bus.

The resulting RFID reader can exchange data with ISO 14443-2 type-B proximity PICCs (proximity integrated coupling cards) also called tags. Communications are possible only when the tags are present in the electromagnetic field generated by the reader built-in antenna.

Once the STM8S-DISCOVERY is powered up through a USB cable connected to the host PC, an electromagnetic field is generated by the RFID reader. A beep is emitted and the LED LD1 briefly lights up when an ISO 14443-2 type-B proximity tag is detected by the reader and its unique identifier (UID) successfully read.

The STM8S-DISCOVERY can be used to evaluate the main features of all STM8S MCUs, even if it is built on an STM8S105C6T6.

Reference documents

- STM8S-DISCOVERY evaluation board user manual (UM0817).
- Developing and debugging your STM8S-DISCOVERY application code (UM0834).
- User manual “CR14 and CRX14 reference design PCB Gerber files” (UM0672) and RFID Gerber files.
- CR14 datasheet: “Low cost ISO14443 type-B contactless coupler chip with anti-collision and CRC management”
- SRI2K datasheet: “13,56 MHz short-range contactless memory chip with 2048 bit EEPROM and anti-collision functions”
- Application note “Antenna (and associated components) matching-circuit calculation of the CRX14 coupler” (AN1806)
- Application note “How to design a 13.56 MHz customized antenna” (AN2866)

These documents can be downloaded from http://www.st.com.

Contents

List of tables

List of figures

1 Application description

1.1 Overview

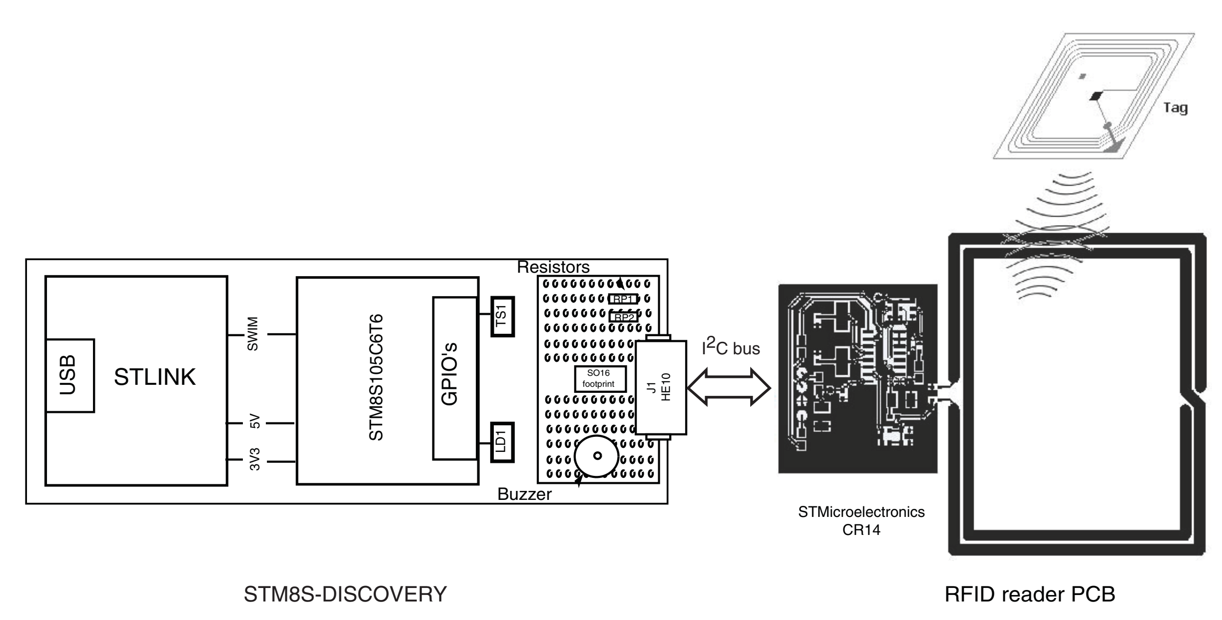
This application is built around a short range RFID reader PCB developed by STMicroelectronics. The PCB Gerber files and the user manual "CR14 and CRX14 reference design PCB Gerber files" (UM0672) are available from http://www.st.com.

This board is designed to be connected to a digital host, in this case an STM8S-DISCOVERY, which manages data transmission and reception through an I^2C interface (see *Figure 1*).

The tags supported by this application must be based on ST contactless memories compliant with ISO 14443 part2 type-B standard for the radio-frequency power and signal interface.

This application has been tested using a tag based on ST SRI2K short range contactless EEPROM. Refer to application note AN2866 explaining how to design a 13.56 MHz tag antenna.

Figure 1. Short range RFID reader solution



1.2 Short-range contactless communication principles

1.2.1 CR14 contactless coupler

The CR14 is the main component of the RFID reader PCB. It interfaces with the following components:

- The contactless tags

  The data frames exchanged with the tags are compliant with ISO14443 type-B radio frequency protocol. Data are stored in the CR14 input/output Frame registers (see *Table 1: CR14 control registers*).
- The STM8S105C6T6 through the I^2C bus.

The CR14 is organized as 4 functional blocks:

- **The I²C bus controller**

  It handles the serial connection with the STM8S105C6 application host, and controls the read/write accesses to all CR14 registers. It is compliant with the 400 KHz I^2C bus specification.
- **The RAM buffer**

  The RAM buffer is bidirectional. It stores all the request frame bytes to be transmitted to the tag, plus all the received bytes sent back by the tag on the answer frame.
- **The transmitter**

  It powers the tag by generating a 13,56 MHz signal on an external antenna. The resulting field is 10% modulated using ASK (amplitude shift keying) modulation to transmit data.
- **The receiver**

  It demodulates the signal generated on the antenna by the load variation of the tag. The resulting signal is decoded by an 847 KHz BPSK (binary phase shift keying) sub-carrier decoder.

The CR14 generates an electromagnetic field which is rectified to power the tag. The reader transmits information to the tag by modulating the carrier wave. To transmit information back to the reader, the tag backscatters the carrier wave by modifying its own impedance thereby perturbing the field.

The CR14 chip contains six volatile registers of which three allow to configure the CR14 and to transmit/receive frames to/from the tag (see *Table 1*):

- Parameter register
- Input/Output Frame register
- Slot Marker register

For details regarding registers description and CR14 I^2C protocol, refer to the CR14 datasheet.

Table 1. CR14 control registers

Address	Description		Access	Purpose
00h	Parameter register	1 byte	W	Set parameter register
			R	Read parameter register

Table 1. CR14 control registers (continued)

Address	Description		Access	Purpose
01h	Input/Output Frame register	36 bytes	W	Store and send request frame to the tag. Wait for tag answer frame
			R	Transfer tag answered frame data to host
02h	ST reserved	N/A	W	ST reserved, must not be used
			R	
03h	Slot Marker register	1 byte	W	Launch the automated anti-collision process from Slot_0 to Slot_15
			R	Return FFh
04h	ST reserved	N/A	R and W	ST reserved, must not be used
05h	ST reserved	N/A	R and W	ST reserved, must not be used

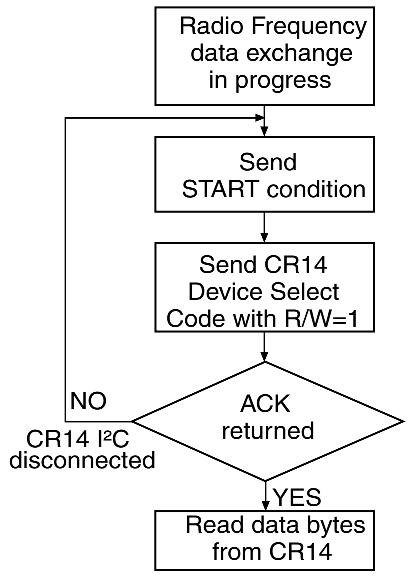
1.2.2 I²C polling using Ack

During radio frequency data exchange, the CR14 disconnects itself from the I²C bus. The time needed to complete the exchange is not fixed as it depends on the tag command format. To know when the exchange is complete before starting reading the data in the Input/Output Frame register, the bus master uses an Ack polling sequence that performs the following actions:

1. **Initial condition**: a radio frequency data exchange is in progress.
2. **Step 1**: the master issues a START condition followed by the first byte of the new instruction (Device Select Code plus R/W bit = 1) (see *Appendix A: I2C memory addressing*).
3. **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 by sending back an ACK, thus indicating that it is ready to receive the second part of the next instruction (the first byte of this instruction has been sent during Step 1).

Figure 2 shows the detailed I²C Ack polling flowchart.

Figure 2. Ack Polling flowchart



1.2.3 Reader-tag protocol

Standard tag commands such as Read and Write are generated by the CR14 using its Input/Output Frame register. To send a command to the tag, the STM8S105C6 host first generates internally the complete frame containing the command code followed by the command parameters (if required). Only the 2 CRC bytes must not be generated as the CR14 automatically adds them during the RF transmission.

Once the frame is ready, the host writes it into the Input/Output Frame register using an I^2C Write command.

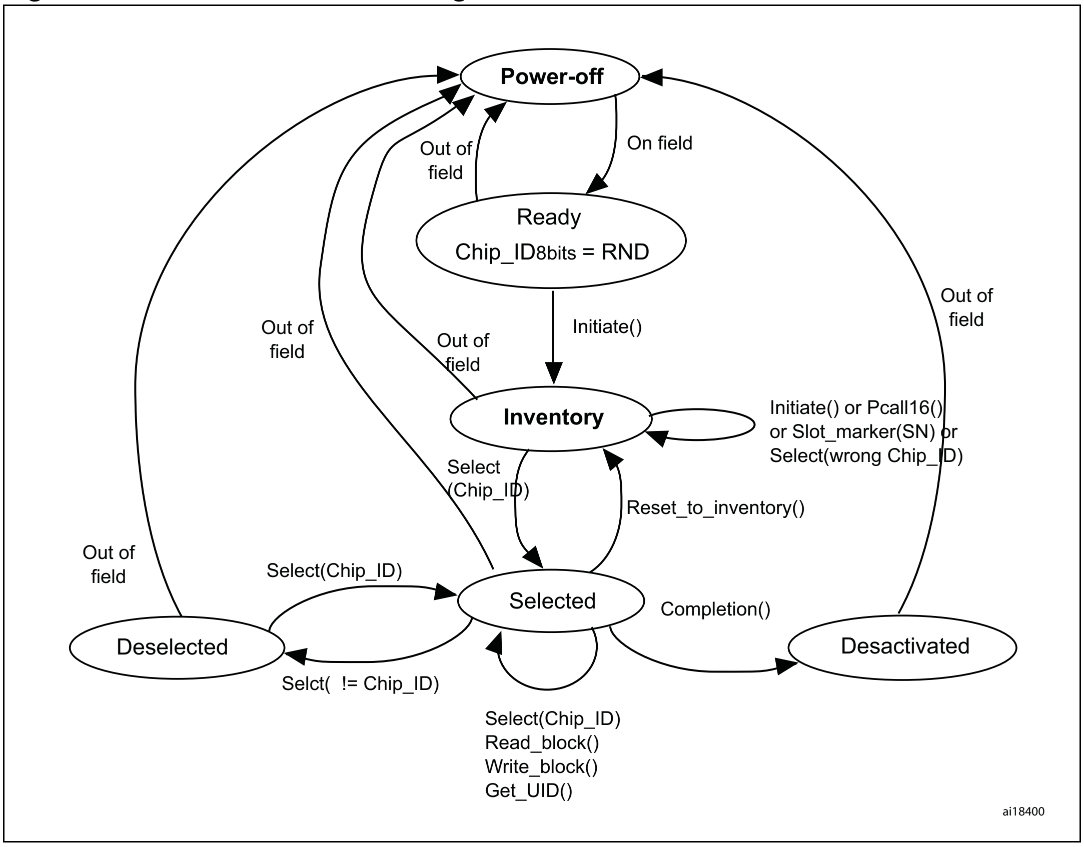
If the tag answers, the characters received are demodulated, decoded and stored into the input/output frame buffer. During the entire RF transmission, the CR14 disconnects itself from the I^2C bus. On reception of the tag EOF, the CR14 checks the CRC and reconnects to the I^2C bus. The host can then get the tag answer frame by issuing an Input/Output Frame Register Read command on the I^2C bus.

Refer to the CR14 datasheet for details on Read and Write commands.

1.2.4 Commands and tag states

The tag can be switched into different states (see *Figure 3*). The tag only answers specific commands depending on its current state. These states are specified by the ISO 15693 standard. For details concerning these states, refers to one of ST short range contactless EEPROM datasheet (for example SRI2K) available from http://www.st.com.

Figure 3. TAG state transition diagram



1.3 Hardware requirements

The following STM8S-DISCOVERY on-board resources are used:

- LED LD1

Refer *Table 2*, *Table 3*, *Table 4*, and *Table 5* for the list of additional hardware required to make this application software run on the STM8S-DISCOVERY.

Table 2. List of PCB passive components

Component description	Value	Comment
Resistors		
R1	N/A	Hardwired addressing of the CR14 (bits b3, b2, and b1 of the 7-bit I²C Device Select Code. Allow up to 7 CR14 readers to be addressed on the same I²C bus
R2	0 Ω	
R3	N/A	
R4	0 Ω	
R5	N/A	
R6	0 Ω	
R7	0 Ω	-

Table 2. List of PCB passive components (continued)

Component description	Value	Comment
R8	0 Ω	Serial resistor allowing to fine tune the quality factor of the reader antenna
Capacitor		
C1, C2	7 pF	crystal oscillator capacitors
C3	22 nF	-
C4	22 µF	-
C5	5 pF (50 V)	-
C6	100 nF	-
C7, C7'	220 pF (50 V), 56 pF (50 V)	Capacitors allowing to fine tune the RLC resonant frequency
C8, C8'	82 pF (50 V), 22 pF (50 V)	impedance adaptation of the RLC resonant circuit.
Diode		
D1 (N4148)		Optional
Ferrite		
L3,L4,L5,L6, L7 multilayer SMD ferrites		Removal of parasites for tag data reception and I^2C

Table 3. List of PCB packaged components

Part name	Component name	Description	Package
CR14/CRX14	Short-range RFID couplers	Contactless coupler compliant with short-range ISO14443 type-B standard	SO16
13.56 MHz crystal	Crystal	Generates a 13.56 MHz carrier frequency	XTALCMS

Table 4. List of tag packaged components

Part name	Component name	Description	Package
SRI2K	13,56 MHZ short-range contactless memory chip	Short-range contactless memory used to build a tag	SBN18

Table 5. List of other passive components

Component description	Value	Comment
Resistor		
R_{P1}, R_{P2} resistors	4,7 KΩ	Pull-up for I^2C open-drain

Table 5. List of other passive components (continued)

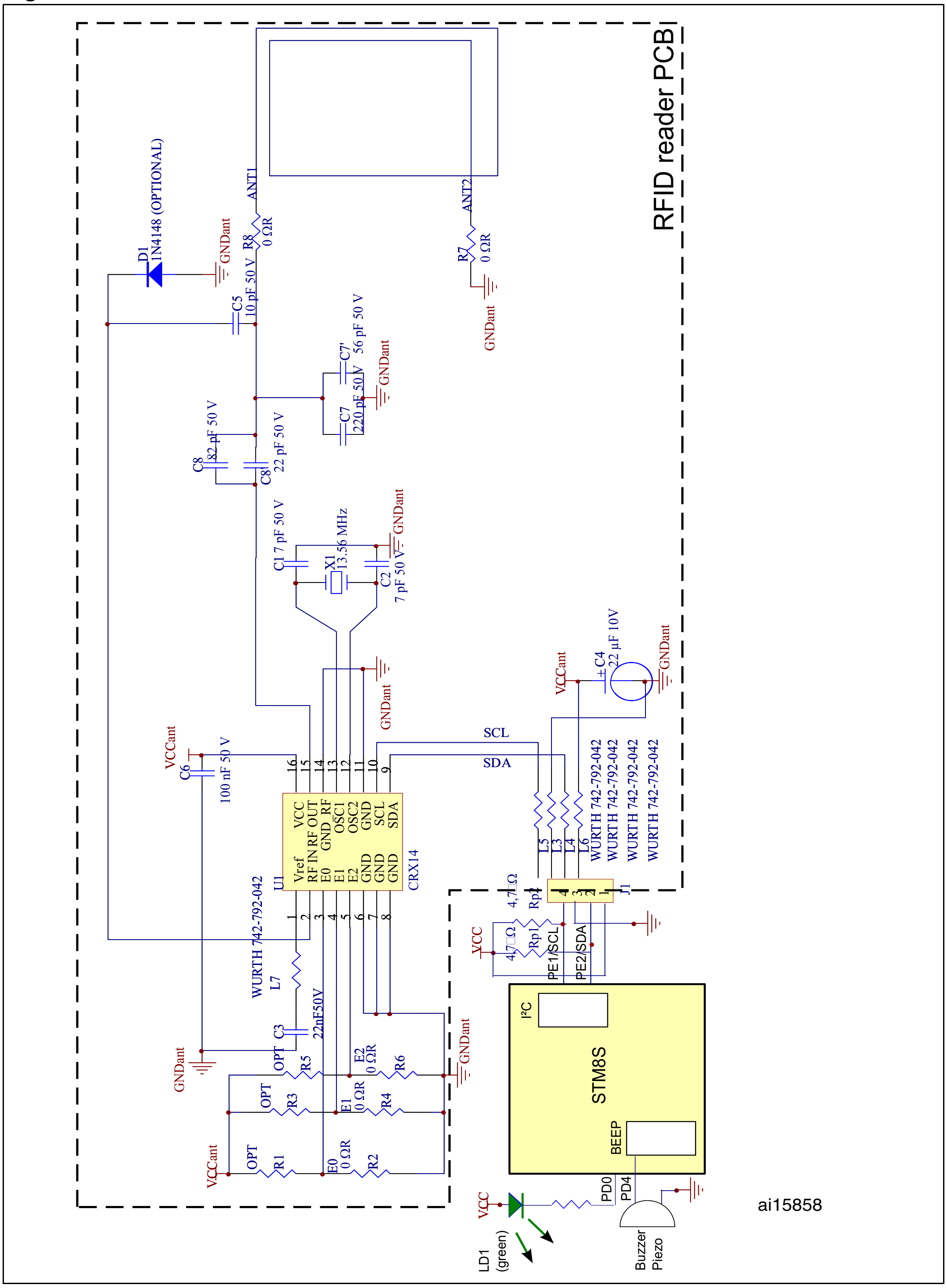
Component description	Value	Comment
Other		
Piezo buzzer	5 V operating voltage	Supports 4 KHz input frequency

1.4 Application schematics

Figure 4 shows the contactless reader implementation schematics. The reader is made of two parts:

- STMicroelectronics RFID reader PCB based on the CR14 short range contactless coupler (see *Section 1.2.1*).
- The STM8S-DISCOVERY which STM8S105C6T6 microcontroller controls the bidirectional communications with the CR14 through the I^2C serial bus (refer to the CR14 datasheet).

Figure 4. Contactless RFID reader schematics



The efficiency of data transfers between the RFID reader and tag depends on the tuning of their respective antenna. This is done by adjusting the following components:

- C_S is the serial capacitance used to adapt the impedance of the reader antenna RLC equivalent circuit:

  C_S = C8 + C8’
- C_P is the parallel capacitance used to tune the resonant frequency of the reader RLC equivalent circuit:

  C_P = C7 + C7’
- R_S is the serial resistor used to fine tune the quality factor of the reader RLC resonant equivalent circuit:

  R_S = R8

Refer to application note AN1806 for details on how to design a reader antenna and infer the values of C_P, C_S and R_S. This document is associated with a software tool using the Grover method to calculate the inductance of rectangular planar antennas.

Warning: The values of C_S, C_P, and R_S are dependant. Tuning one of them impacts the 2 others. The best compromise must be found to achieve a good tuning for the reader antenna (refer to AN1806).

2 STM8S-DISCOVERY configuration

2.1 Power supply configuration

The CR14 of the RFID reader PCB must be supplied from 5 V± 500 mV. This board being powered from STM8S-DISCOVERY V_{DD}, the jumper JP1 of the STM8S-DISCOVERY must be set to V_{DD} = 5 V (see UM0817).

2.2 Option byte configuration

The STM8S105C6T6 Beeper output is enabled through the alternate function remapping option (AFR7) of the OPT2 option byte:

AFR7 = 0: port D4 alternate function is TIM2_CH1 (default)

AFR7 = 1: port D4 alternate function is BEEP (required)

For details on the option byte and alternate function remapping, refer to UM0834 and to the STM8S105xx datasheet, respectively.

3 Software description

3.1 STM8S peripherals used by the application

The application software uses STM8S standard firmware library to control general purpose functions. These peripheral functions are the following:

- Clock (CLK)

  The clock control enables and delivers the correct clock frequency to the CPU and peripherals. It configures the HSI prescaler division factor to 4. The I^2C input clock frequency is 4 MHz to be able to generate correct timings compliant with Fast mode.
- GPIOs

  They drive the MCU I/Os to interface with external hardware. They configure PD0 port as output push-pull high to drive LD1 and switch it off at initialization. PD4 port is configured through alternate function remapping to enable the Beeper output pin.
- I^2C

  This peripheral handles the serial connection with the CR14 contactless coupler of the RFID reader board. It controls the read/write access to the CR14 registers.
- Auto wake-up (AWU)

  This peripheral is used to provide an internal wake-up timebase that is used when the MCU goes into Active-halt power saving mode. It is configured to wake up the MCU after 512 ms, which is a good trade-off between the time during which the microcontroller remains in Active-halt power saving mode and the time required by the RFID reader to identify the tag.
- Beeper

  This peripheral drives the Beeper output pin with a signal of 4 KHz for sound generation.
- TIM3

  The TIM3 timer is used to measure the LSI frequency with Input Capture 1 to reach with a better accuracy the standard Beeper frequency outputs.

3.2 Configuring the STM8S standard firmware library

The *stm8s_conf.h* file of the STM8S standard firmware library is used to configure the library by enabling the peripherals used by the application.

The following define statements must be present:

`#define _CLK 1` enables the clock control (CLK),
`#define _GPIO 1` enables the GPIOs,
`#define _I2C 1` enables the I^2C interface,
`#define _AWU 1` enables the Auto wake-up,
`#define _BEEP 1` enables the beeper,
`#define _TIM3 1` enables timer 3.

3.3 Application principle

This application initiates communications with the tags present in the range of the electromagnetic field generated by the reader antenna, and read the tag UIDs. If the operation succeeds, the STM8S105C6T6 emits a beep and blinks LD1.

When running the code in debug mode from STVD development tool, the UID value can be displayed in the **Watch** windows.

The application uses the I^2C protocol to configure the CR14 embedded on the RFID reader board, and to transmit/receive frames to/from the tags (see *Table 6*). Communications with the tags are managed by the state transition diagram shown in *Figure 3*.

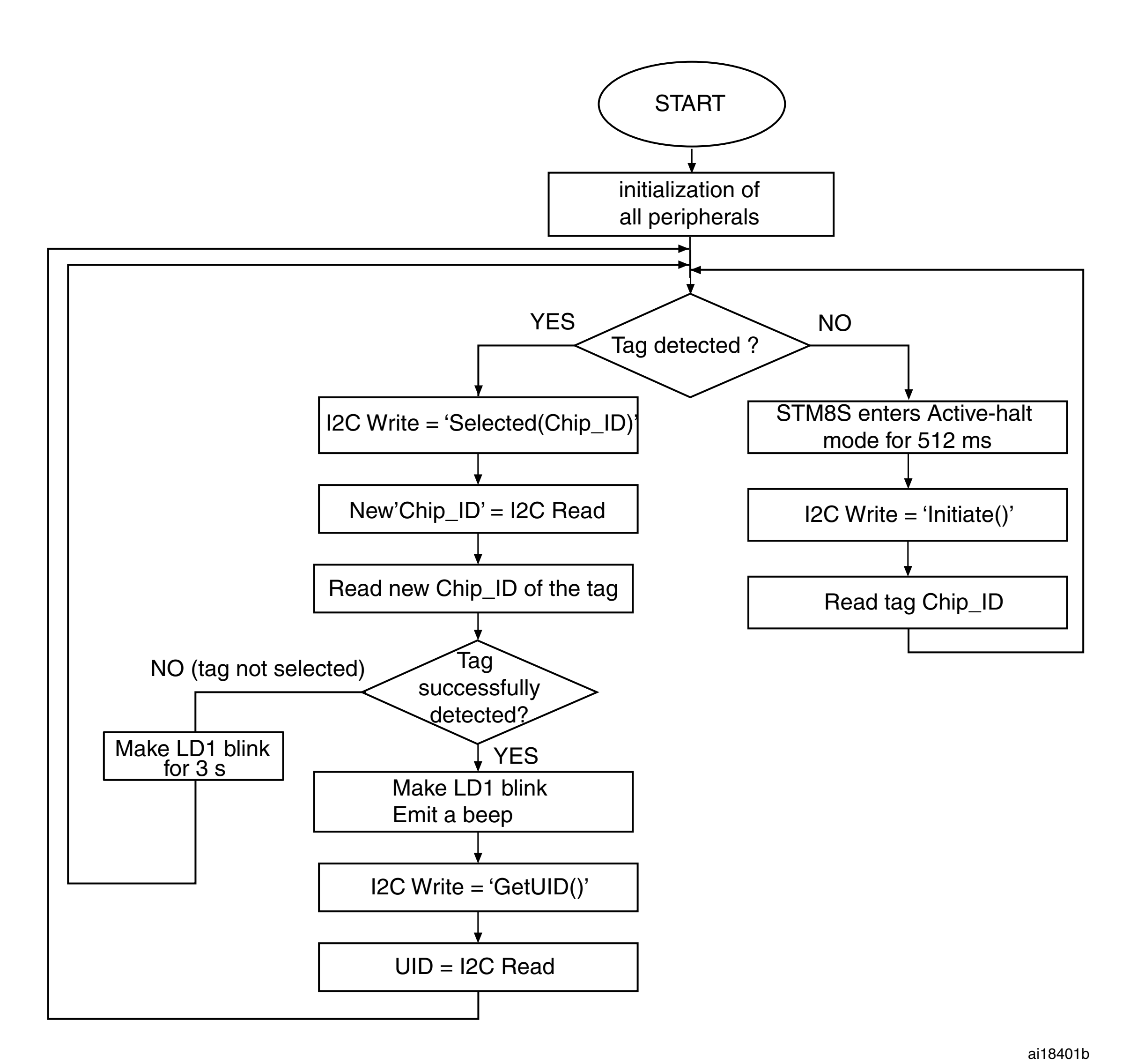
A possible way to improve this application would be to use an LCD screen to display the tag unique identifier (UID). Another improvement could be to implement the CR14 anti-collision capability which allows the reader to select up to 16 tags (one at a time) during predefined time slots.

Refer to *Figure 5* for the flowchart of the application software main loop.

Table 6. TAG commands used within this application

Tag commands	Description
Initiate()	To detect if a tag in *Ready* state is present in the reader field range
Select(Chip_ID)	Allow the tag to enter the *Selected* state
Get_UID()	On receiving this command, the tag returns its 8 UID bytes

Figure 5. Main application loop flowchart



During peripheral initialization, the application first configures the CR14 parameter register to generate the 13,56 MHz RF field on the reader antenna.

Prior to issuing a Read/Write command (such as Get_UID()) to the memory tag, the tag state machine must be put in the *Selected* state by sending a Select(Chip_ID) command. All commands sent to the tag before this command is issued are ignored.

Once the peripheral initialization has completed, the application code enters a loop in which it checks if a tag is present in the RF field by writing periodically the Initiate() command in the CR14 Input/Output Frame register (see *Appendix A: I2C memory addressing*). If no tag is present in the electromagnetic field, the STM8S105C6T6 enters the Active-halt power saving mode for 512 ms and automatically wakes up to issue a new Initiate() command.

As soon as a tag is present in the electromagnetic field, it automatically enters the *Ready* state in which its 8-bit random Chip_ID is initialized. When receiving the Initiate() command, the tag then switches to the *Inventory* state in which its new 8-bit Chip_ID random value is set and returned to the CR14. Since this application software does not implement the anti-collision mechanism, the Chip_ID is stored by the host and sent as parameter to the Select(Chip_ID) command issued by the host to the tag. The tag then switches directly to the *Selected* state.

The Get_UID() command is then sent to the tag that answers by returning its 8 UID bytes. The host emits a beep and lights LD1 for about 1 s.

Appendix A I^2C memory addressing

To start communicating with the CR14, the bus master initiates a START condition and sends 8 bits (with Most Significant Bit first) on the serial data line SDA. These bits contain the Device Select Code (7 bits) and the RWbar bit.

According to the I^2C bus definition, the seven Most Significant Bits of the Device Select Code are the Device Type Identifier. For the CR14 these bits definition is given in *Table 7*.

Table 7. Device Select code

	Device Code				Chip Enable			RWbar
	b7	b6	b5	b4	b3	b2	b1	b0
CR14 select	1	0	1	0	E2	E1	E0	RWbar

The 8th bit is the Read/Write bit (RWbar). It is set to ‘1’ for I^2C read and to ‘0’ for I^2C write operations.

If the data sent by the bus master matches the Device Select Code of the CR14, it returns an acknowledgement on the bus during the 9th bit time. The CR14 device generates a NACK if its Device Select Code does not correspond to the data sent. It deselects himself from the bus and goes in standby mode.

Refer to the CR14 datasheet for details regarding the CR14 I^2C protocol, and to the STM8S reference manual for a functional description on I^2C peripheral.

Appendix B I^2C Read and Write functions

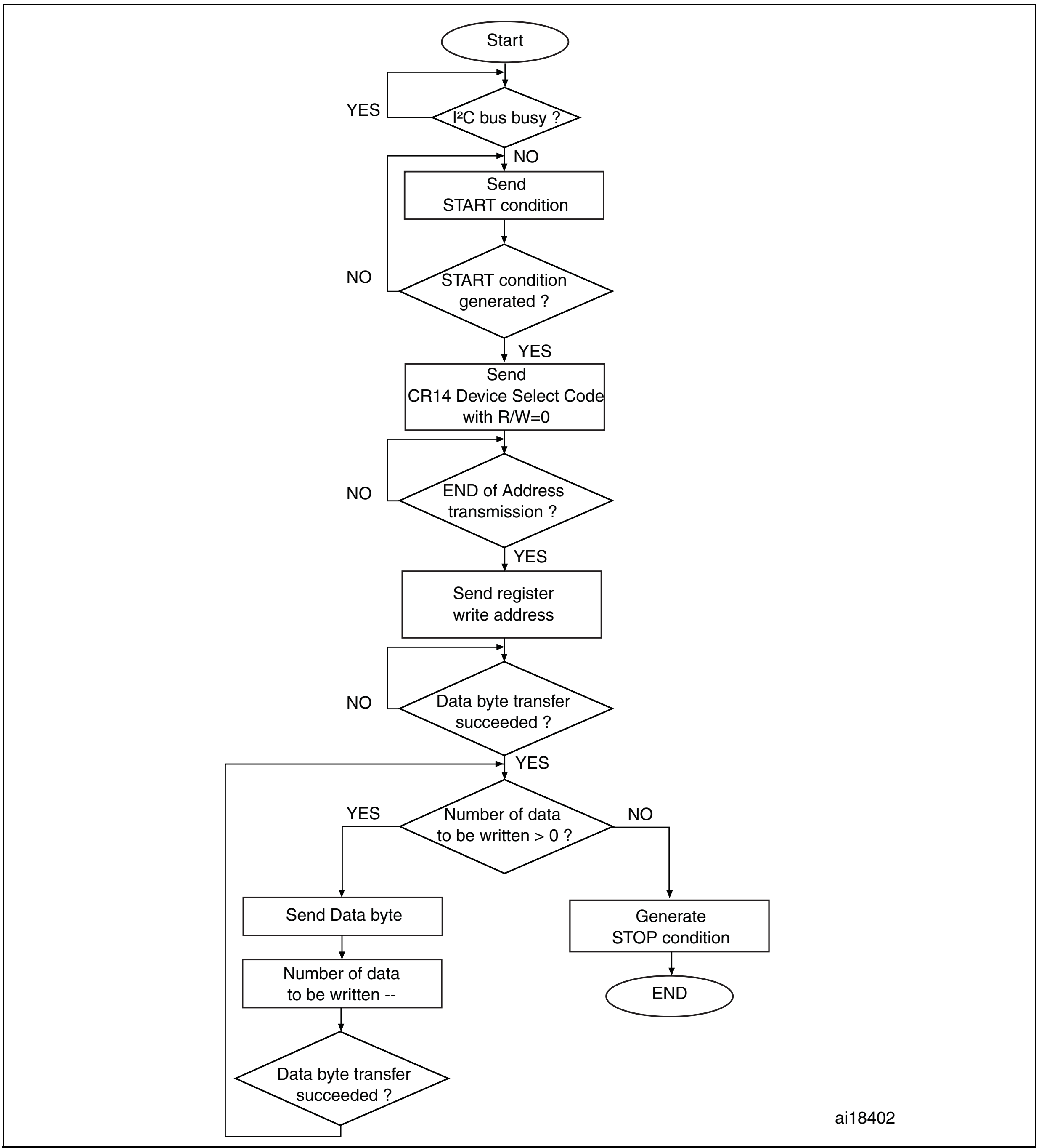
This section gives an overview of the main functions controlling the read and write accesses to the CR14.

B.1 I^2C Page Write flowchart

The Page Write function performs write accesses to the CR14 registers. Its parameters are the buffer containing the bytes to be written to the CR14, the CR14 write address and the number of bytes to be written.

Figure 6 shows the detailed flowchart of the `I2C_Page_Write` function.

Figure 6. I^2C Page Write flowchart

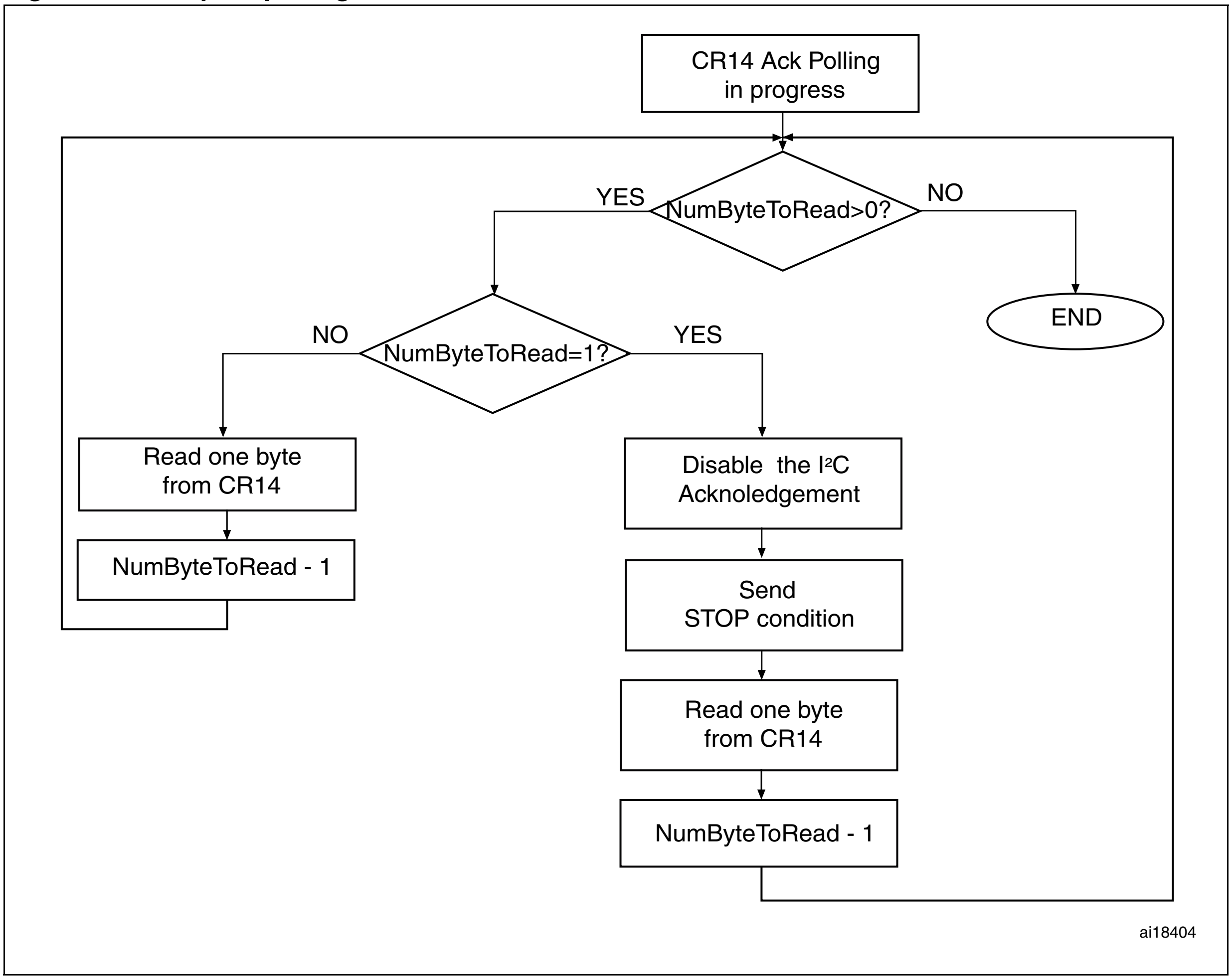


B.2 I²C post polling buffer read flowchart

This function has been implemented to be used together with the I²C polling function. Once an ACK has been returned by `I2C_CR14_EE_AckPolling`, the `I2C_CR14_EE_PostPolling` function reads one by one all the bytes that are available in the Input/ Output Frame register of the CR14.

Figure 7 shows the detailed flowchart of the `I2C_CR14_EE_PostPolling` function.

Figure 7. I²C post polling buffer read flowchart



Revision history

Table 8. Document revision history

Date	Revision	Changes
12-Oct-2010	1	Document migrated from UM0927 rev 1.