Application Report

SLOA233–October 2016

RF430FRL152H NFC Sensor Tag Application Example With MSP430™ Microcontrollers

Ralph Jacobi

Safety and Security (S2) NFC/RFID Applications

ABSTRACT

Analog and digital sensors coupled together with wireless technology is an expanding use case across many different areas of growing technology such as diagnostics, data logging, and medical. Near Field Communication (NFC) technology has emerged as a viable option for many of those applications that can benefit from utilizing a contactless short-range wireless solution.

Texas Instruments offers the RF430FRL15xH family of dynamic NFC sensor transponders that allow multiple analog and digital sensors to be interfaced to a single chip that can read sensor data and then transmit it over the air using NFC communication. The RF430FRL15xH devices use the ISO/IEC 15693 protocol for over-the-air communication, has a 14-bit sigma-delta ADC, and an I^2C or SPI data bus.

This application report describes how to use an MSP430FR4133 16-bit FRAM microcontroller and a TRF7970A NFC transceiver to read data from the RF430FRL152H sensor patch. The application report also includes a brief description of how to use TI's MSP-IQMATHLIB to calculate the temperature read by the thermistor while minimizing software overhead and power consumption.

The sample code that is described in this document can be downloaded from http://www.ti.com/lit/zip/sloa233.

Contents

List of Figures

1 Introduction

The RF430FRL15xH series of devices is a good fit for applications where sensors are used and benefit from having a wireless interface. The RF430FRL15xH uses NFC/RFID communication to transmit data wirelessly over short distances (typical range is 1 to 5 cm). The RF430FRL15xH can be powered by an RF field to allow for battery-less applications that use one or more sensors. This document describes the fundamental features of the RF430FRL15xH parts and the basics of the NFC/RFID communication that they support.

To receive sensor data from a tag using the RF430FRL152H, an NFC reader supporting ISO/IEC 15693 must be used. Additionally, depending on the sensor, calculations may be required to convert the raw values into practical data. Therefore, this document also discusses methods that can be implemented to achieve this functionality with minimal MCU resources.

The firmware example included with this application report is designed to implement these concepts to read sensor data from the RF430-TMPSNS-EVM (see Figure 1). The RF430-TMPSNS-EVM evaluation board and design files can be found at http://www.ti.com/tool/TIDM-RF430-TEMPSENSE.

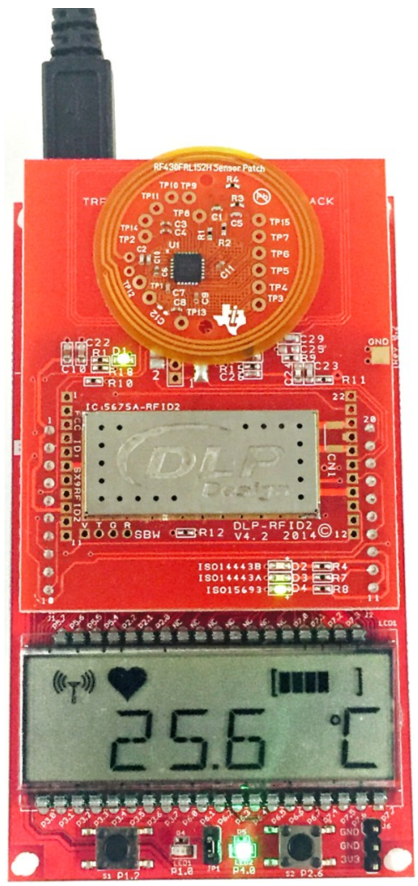


Figure 1. RF430FRL152H Sensor Tag and MSP430FR4133 Reader

2 Out of the Box Experience

This section describes the steps required to program and use the MSP-EXP430FR4133 and DLP-7970ABP to read the data from an RF430-TMPSNS-EVM.

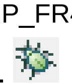
Before following these steps, make sure to extract the contents of the sloa233.zip from Section 1 and make sure that either Uniflash V3.4 or newer, or Code Composer Studio™ IDE V6.0.1 is installed on the PC. If neither are installed, Uniflash V3.4 is recommended for simple flash programming.

2.1 Using Uniflash

1. Connect the DLP-7970ABP BoosterPack™ plug-in module to the MSP430FR4133 LaunchPad™ development kit.
2. Connect a USB cable from the PC to the LaunchPad development kit.
3. Open Uniflash V3.4 or newer.
4. Go to File → New Configuration.
5. Select *Connection* = TI MSP430™ USB1 and *Board or Device* = MSP430FR4133.
   (a) Apply Firmware Update for Emulator if prompted.
6. Go to Program → Load Program → Browse.
7. Select the .out file found at [Install Path]\TRF7970ABP_FR4133_FRL_Reader\Debug.
8. Click OK and wait for the program to finish loading.
9. Reset the board by power cycling it.
10. Present the RF430-TMPSNS-EVM to the antenna on the top of the DLP-7970ABP.
11. Read the temperature data from the LCD display.
    (a) The LCD display also has the following features (see Figure 2):
        (i) Heart Symbol – This is a heartbeat to indicate the code is running
        (ii) Wireless Connectivity Symbol – This indicates the TRF7970A is able to communicate with the RF430FRL152H.
        (iii) Signal Strength Bar – This indicates how strong or weak the RF communication strength is from the tag.
    (b) The push button S2 can be used to toggle additional LCD display features.
        (i) Button S2 (P2.6) toggles the displayed temperature result units between Fahrenheit and Celsius. The display defaults to degrees Celsius.

2.2 Using Code Composer Studio™ IDE

Instead of programming the firmware with CCS Uniflash, the Code Composer Studio IDE can also be used to program the firmware. This option also allows users to modify and debug the firmware for additional custom applications.

1. Connect the DLP-7970ABP to the MSP430FR4133 LaunchPad development kit.
2. Connect a USB cable from the PC to the LaunchPad development kit.
3. Open Code Composer Studios V6.0.1 or newer on the PC.
4. Go to Project → Import CCS Project → Browse.
5. Navigate to contents extracted from the sloa233.zip file.
6. Import the TRF7970ABP_FR4133_FRL_Reader project.
7. Click the Debug button. 
8. When the code finishes downloading, click the Stop Button. 
9. Reset the board by power cycling it and then resume from Step 10 in Section 2.1.

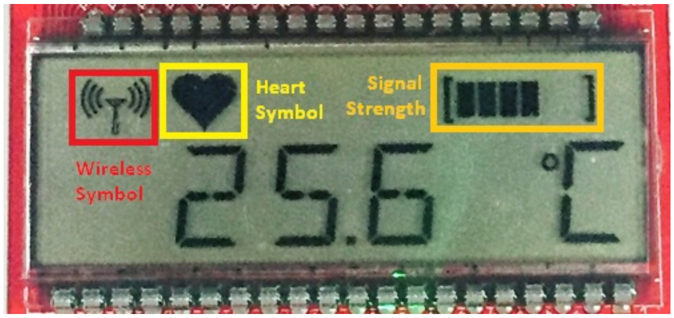


Figure 2. Additional LCD Display Features

3 RF430FRL152H NFC ISO/IEC 15693 Sensor Transponder

3.1 Features

The RF430FRL15xH family of devices has a programmable 16-bit MSP430 core that is interfaced with an analog front end (AFE) to make a single chip solution. This allows for the interfacing of multiple analog or digital sensors while also supporting the transmission of sensor data over the short range wireless NFC technology.

The internal MSP430 core has 2KB of FRAM and 4KB of SRAM available for programming and data storage. It supports SPI and I^2C communication, and has a 3-channel 14-bit sigma-delta ADC. The device runs at a clock speed of 2 MHz.

The RF430FRL15xH devices operate at a 1.5-V voltage level that allows for the use of low-voltage coin cell batteries. It is also possible to passively power the device by drawing energy from the RF field of any reader that is communicating with an RF430FRL15xH.

The RF430FRL15xH portfolio consists of three parts: the RF430FRL152H, the RF430FRL153H, and the RF430FRL154H. The 152H supports all of the hardware features mentioned above. However, the 153H does not have the SPI and I^2C interface, and the 154H does not have the sigma-delta ADC.

This application report describes the fully featured RF430FRL152H exclusively.

3.1.1 Sensor Application

The RF430FRL15xH has a Sensor Application ROM coded into the device that allows for seamless handling of multiple sensors by simply configuring a set of registers.

Features that are supported by the Sensor Application include:

- Number of measurement cycles
- Skipping of specific sensor measurements based on the cycle count
- Measurement frequency (either predefined time steps or a custom time steps)
- Which samples are stored (all, first, last, highest, lowest, or average)
- Sensor configurations
  - Gain
  - Filter Type (CIC or Moving Average)
  - Oversampling Rate
- Monitor and Alarm Conditions

All of these features are handled by the ROM code and can be enabled, disabled, and configured with simple register writes that can be part of the FRAM firmware or done over the air with NFC technology.

3.1.2 14-Bit Sigma-Delta ADC

The 14-bit sigma-delta ADC allows up to three analog sensors to be interfaced to the RF430FRL152H. The ADC also has 2 channels (ADC1 and ADC2) that can measure resistance that allows for the use of thermistors.

The sigma-delta ADC is a slow-acquisition ADC that has ultra-low-power input current, offset, and low-noise thresholds. This allows users to minimize the power required for taking samples with the device. Due to the nature of how sigma-delta ADCs operate, it takes additional time to get high-accuracy samples. For example, a full 14-bit conversion takes approximately 2 seconds. Much faster samples can be taken at the cost of accuracy.

3.1.3 Memory Structure

The RF430FRL15xH has a total of 2KB of FRAM, 4KB of SRAM, and 8KB of ROM. The ROM contains the Sensor App, ISO/IEC 15693 RF Stack, and the Boot Code for the device, and cannot be modified. The 4KB of SRAM is used to store sensor data and consists of 8 sectors of 512 bytes each. Individual sections can be powered down to save power, but all data stored in a powered down section of SRAM is lost.

The 2KB of FRAM can be used to write custom applications and store data. This includes applications that manage the communication to digital sensors through serial protocols. FRAM is an ultra-low-power nonvolatile memory that stores data even when powered down. The FRAM can be partitioned to function as additional RAM memory to store extra sensor data, and it is partitioned in 4 blocks of 512 bytes each.

3.1.3.1 NFC/RFID Block Format

The RF430FRL15xH has a total of 0xF2 (242) blocks of memory that contain NFC/RFID readable data. Each block is 8 bytes wide, so there is a total of 1936 bytes of data available for use with NFC/RFID.

3.2 RF430FRL152H Sensor Tag

The RF430FRL152H has been used to create the Battery-less NFC/RFID Temperature Sensing Patch TI Design. This patch is made on a flexible PCB and includes a single thermistor interfaced to the RF430FRL152H to provide temperature readings. The tag is powered by harvesting energy from the RF field generated by NFC/RFID readers.

3.2.1 Tag Design

The sensor patch includes an RF430FRL152H IC chip, passive components used to properly connect the RF430FRL152H pins with power and ground connections, and the circuitry for the thermistor including a reference resistor.

The RFID antenna is included in the PCB on the outside rim and surrounds the entire patch. It is a 3-turn coil tuned by two capacitors.

A number of test points are also included that can be used to program the RF430FRL152H with firmware, connect the device through SPI or I^2C lines, or connect a battery to the patch.

3.2.2 Thermistor

The 100-kΩ thermistor used for the sensor patch design is a Panasonic Negative Temperature Coefficient (NTC) Thermistor. The part number for it is ERT-J1VS104FA. The thermistor functions accurately from –40°C to 125°C and provides accurate measurements within 0.5°C. A 100-kΩ reference resistor is also used for higher accuracy.

3.2.3 Block Diagram

Figure 3 shows a simple block diagram of the key components. The test points are used to access pins for debugging and providing an option to JTAG program the RF430FRL152H. These are recommended for testing new designs. Full schematics can be found in the Battery-less NFC/RFID Temperature Sensing Patch TI Design.

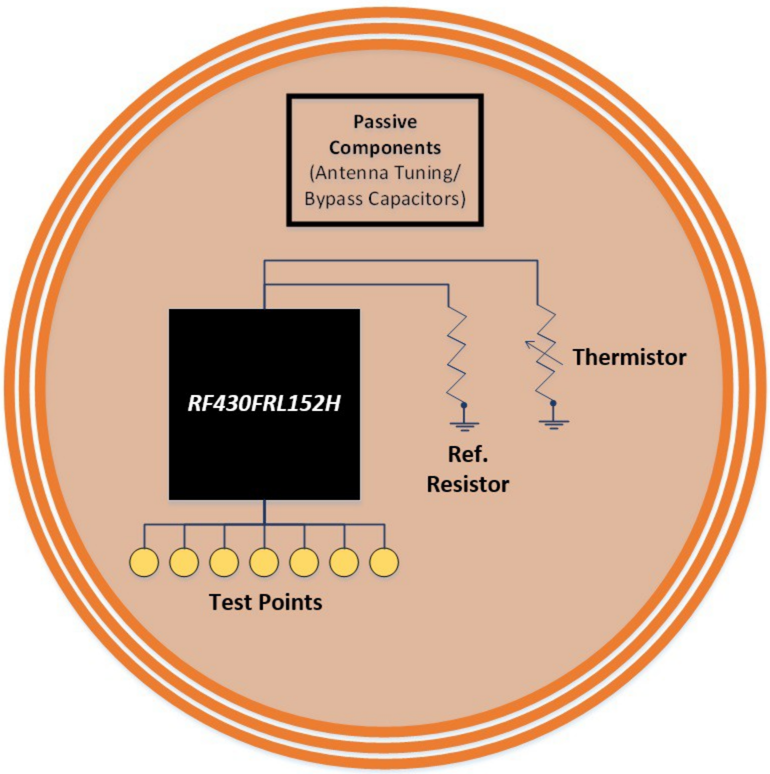


Figure 3. Block Diagram for RF430FRL152 Sensor Tag

4 ISO/IEC 15693 Standard

The ISO/IEC 15693 standard is for Vicinity Integrated Circuit Cards (VICC), which support greater reader ranges than Proximity Integrated Circuit Cards (PICC) such as those that use the ISO14443 standard. All ISO/IEC 15693 compliant devices operate at a radio frequency of 13.56 MHz.

The over-the-air data rates for all ISO/IEC 15693 reader commands are either 26.48 kbit/s or 6.62 kbit/s. ISO/IEC 15693 readers use Amplitude-Shift Keying (ASK) for communication at a modulation index of either 10% or 100%.

Whether the low or high data rate is used in a reply is determined by the ISO/IEC 15693 reader. The reader can request either high or low data rates from tag responses based on the request flags of the command being sent. For more details about request flags, see the official ISO/IEC 15693-3 specifications.

4.1 Anticollision

The ISO/IEC 15693 standard includes a method that can be used to handle tag collisions. The ISO/IEC 15693 anticollision procedure is a robust solution to the problem of how to detect and communicate with a single NFC/RFID tag when more than one NFC/RFID tag is in the RF field.

When a single ISO/IEC 15693 tag is presented to an ISO/IEC 15693 capable NFC/RFID reader, a single slot Inventory command can be used to initiate communication with the tag. The tag replies with its Unique Identifier (UID) and Data Storage Format Identified (DSFID)

However, when multiple ISO/IEC 15693 tags are presented at the same time, a collision occurs as each tag is trying to reply to the same command.

To resolve the collision of multiple ISO/IEC 15693 tags, a 16-slot Inventory command is issued instead. This command includes a mask to aid the resolution of tag collisions. Each slot of the 16 slots corresponds to the least significant nibble of the UID for an ISO/IEC 15693 tag, and the start of each slot after slot 0 (which is the Inventory command itself) is indicated by a specific modulated signal known as a slot marker. When a tag receives the command or the appropriate slot marker for the slot it falls into, it then replies with its UID and DSFID.

However, if two tags share the same least significant nibble or nibbles, they will still collide. To handle this, masking is used. When tags collide at a slot, that slot is recorded and then another 16-slot Inventory command is issued, this time with the mask including the known nibbles of the tags that collided. This forces only the tags whose least significant nibbles match the mask to reply, and they reply at a slot based on their least significant unmasked nibble. This process continues to loop through until all tag collisions are resolved.

Figure 4 shows the full software flow for the ISO/IEC 15693 Anticollision routine included in the firmware example.

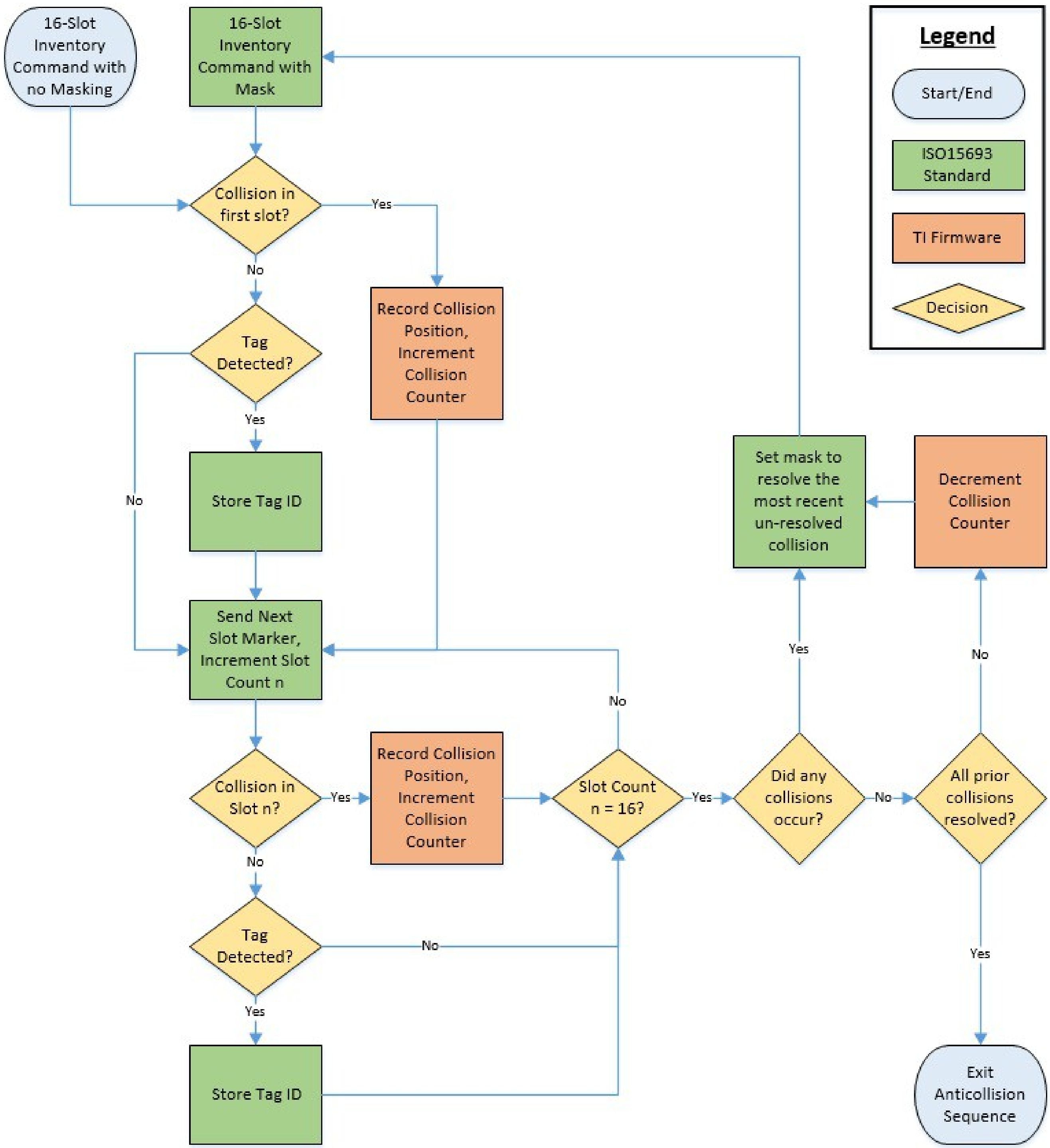


Figure 4. ISO/IEC 15693 Anticollision Flowchart

4.2 Addressed Commands and Nonaddressed Commands

The ISO/IEC 15693 standard does not require a specific flow of commands to select and activate ISO/IEC 15693 compliant tags, unlike some other HF RFID standards. This gives users flexibility in how to implement an application but also requires a method to allow for very robust applications to be developed. This is managed through the use of addressed commands.

Most ISO/IEC 15693 commands allow for the optional inclusion of a tag UID to be issued along with the command. This is known as an addressed command. Only the tag with the matching UID can reply to an addressed command. The tag detects whether or not a command is addressed based on the request flag indicator. For further details on how request flags are used, see the ISO/IEC 15693-3 standard.

4.3 Identifying the RF430FRL152H ISO/IEC 15693 Tag

The ISO/IEC 15693 standard supports the identification of a specific ISO/IEC 15693 tag through a combination of the tag UID and programmable information, such as DSFIDs and Application Family Identifiers (AFIs). Given that each ISO/IEC 15693 tag has its own UID, and most applications interact with a significant number of tags, it is usually not feasible to exclusively use the UID as a method of identifying a specific ISO/IEC 15693 tag. In this case, the Get System Information command can be used to get additional tag information such as the DSFID, AFI, and also the memory format information for the tag.

For the RF430FRL152H, the default response to the Get System Information command provides the memory format information, but no DSFID or AFI data. Using a combination of the Get System Information response along with the tag UID allows any NFC/RFID reader to identify a tag as an RF430FRL152H.

The first 2 UID bytes are the mandatory 0xE0 byte (required for all ISO/IEC 15693 compliant tags) and the manufacturer identification byte, which for Texas Instruments products is 0x07. The Get System Information response for memory format returns 0xF2 for the number of blocks and 0x07 for the size of each block. These values are unique to the RF430FRL15xH family of devices in the Texas Instruments NFC/RFID portfolio. Therefore, checking for a combination of those values can allow for the identification of any RF430FRL152H tag.

4.4 Reading and Writing ISO/IEC 15693 Tag Data

To read data from an ISO/IEC 15693 tag, either a Read Single Block or Read Multiple Block command can be used. To write data to an ISO/IEC 15693 tag, a Write Single Block command can be used. For this application report, only the Read Single Block and Write Single Block commands are discussed.

The Read Single Block command can be issued with the optional inclusion of a specific ISO/IEC 15693 tag UID. If the UID is included, this is known as an addressed read and only the tag with that specific UID replies. By using the process of identifying the RF430FRL152H tag first, addressed reads can be used to ensure that the data being received is only from it.

The same process can be used for the Write Single Block command to ensure that only the RF430FRL152H is having data written to it. TI recommends using addressed commands for Write Single Block to only write to the intended tag and avoid accidently overwriting data on other nearby tags.

5 TRF7970A NFC Transceiver

The TRF7970A is an NFC/RFID transceiver that supports the use of all three NFC modes: Reader/Writer, Peer-to-Peer, and Card Emulation. For Reader/Writer, the supported technologies are ISO14443A (NFC Tag Type 2 and Type 4A), ISO14443B (NFC Tag Type 4B), FeliCa (NFC Tag Type 3), and ISO/IEC 15693 (NFC Tag Type 5). For Peer-to-Peer, the supported technologies are NFC-A (106 kbps) and NFC-F (212 or 424 kbps), and the TRF7970A can serve as an active or passive device as well as an initiator or a target device. For Card Emulation, the supported technologies are NFC Type 4A and NFC Type 4B tags.

The TRF7970A is controlled by sending direct commands and control register configurations to the device through either a SPI or parallel interface. The TRF7970A has a 127-byte FIFO register that is used to transmit and receive data. The device can automatically handle the decoding, encoding, and data framing for all data packets being transmitted and received for the configured technology type, as well as automatic parity and CRC checks.

This document focuses exclusively on Reader/Writer mode for ISO/IEC 15693 (NFC Type 5) tags. For details on how to use the TRF7970A for any of the other modes mentioned above, see the appropriate TRF7970A application report:

- Reader/Writer: NFC/HF RFID Reader/Writer Using the TRF7970A
- Peer-to-Peer: NFC Active and Passive Peer-to-Peer Communication Using the TRF7970A
- Card Emulation: NFC Card Emulation Using the TRF7970A

5.1 TRF7970A Configuration

For the example firmware provided, the TRF7970A is being configured to use the ISO/IEC 15693 high data rate option for communication with the RF430FRL152H tag. The specific data rate is 26.48 kbps, and is set by writing 0x02 to the ISO Control Register (Register 0x01) of the TRF7970A.

The TRF7970A also includes a number of other registers that can be modified on an application by application basis. The firmware provided is designed just for ISO/IEC 15693 NFC/RFID Reader functionality, so there is no need to modify these registers manually. The TRF7970A automatically optimizes the relevant registers for a specific technology type when the ISO Control Register is written.

The MSP430 LaunchPad development kit with DLP-7970ABP hardware that is used for the example firmware runs at a 3.3-V logic level, so the TRF7970A is configured for 3.3-V operation (Bit 0 in the Chip Status Control Register (0x00)). However, it would be possible to use a 5-V setting with custom hardware or when using an evaluation platform that supports 5 V. Using 5 V allows for the read range to be extended further at the cost of greater power consumption.

6 MSP430 IQmathLib

To calculate the temperature based on the raw result from the ADC measurement of the thermistor, a series of math operations such as multiplication, division, logarithms, and exponentials need to be executed. These calculations are traditionally handled in microcontrollers (MCUs) by using floating-point arithmetic. However, floating point is inherently resource intensive for many smaller MCUs that typically can feature NFC functionality without major code space issues. The calculations also use up a large number of cycles, which can cause issues for optimizing systems for low power as well.

A method to address those issues and allow for less-expensive lower-resource MCUs to be used is to include the MSP430 IQmathLib and use the fixed-point arithmetic capabilities that are included with the library. Fixed-point arithmetic allows for an efficient trade-off between code size and accuracy of the result, and also significantly reduces the number of the cycles needed for calculations.

The basic concept of fixed-point arithmetic is to map floating point numbers into integers and then handle calculations such as multiplication, division, logarithms, and exponentials by using functions that only use integer addition, subtraction, and logical operators (AND, OR, bit shifting, and so on) instead of the memory-intensive floating-point functions. A drawback is that there is reduced resolution in the decimal values when they are mapped from floating point to fixed point. Another issue is that the maximum value is also limited, so it is possible to unintentionally overflow the fixed-point integers, which results in any following calculations being erroneous.

The mapping method for fixed point allows a fair amount of flexibility in terms of how much overhead is permitted for large calculations versus how much accuracy is afforded for the resolution of decimal values. It is possible to have very large numbers with just a few digits of resolution on the decimal value or smaller numbers with multiple digits of resolution for decimal values.

7 Hardware Description

7.1 MSP-EXP430FR4133

The MSP-EXP430FR4133 LaunchPad development kit is an easy-to-use Evaluation Module (EVM) for the MSP430FR4133 microcontroller. It contains everything needed to start developing on the MSP430 ultra-low-power (ULP) FRAM-based microcontroller (MCU) platform, including on-board emulation for programming, debugging, and energy measurements. The board features on-board buttons and LEDs for quick integration of a simple user interface as well as a liquid crystal display (LCD) display that showcases the integrated driver with flexible software-configurable pins. The MSP430FR4133 device features embedded FRAM (ferroelectric random access memory), a nonvolatile memory known for its ultra-low power, high endurance, and high-speed write access. Rapid prototyping is simplified by the 20-pin BoosterPack plug-in module headers, which support a wide range of available BoosterPack plug-in modules. You can quickly add features like wireless connectivity, graphical displays, environmental sensing, and much more.

7.2 RF430-TMPSNS-EVM

Battery-less sensor measurements that use energy harvesting from the RF field can be implemented in applications including medical, health and fitness, and industrial where instantaneous measurements are required and a battery is not feasible or desired. This design implements a single-chip NFC/RFID field powered temperature sensor system with the RF430FRL152H.

7.3 DLP-7970ABP

The third-party provider DLP Design NFC/RFID BoosterPack plug-in module (DLP-7970ABP) is an add-on board designed to fit all of TI's MCU LaunchPad development kits. This BoosterPack plug-in module allows the software application developer to get familiar with the functionalities of TRF7970A multiprotocol fully integrated 13.56-MHz NFC/HF RFID IC on the TI embedded microcontroller platform of choice without having to worry about developing the RF section.

The TRF7970A device is an integrated analog front end and data-framing device for a 13.56-MHz RFID and NFC system. Built-in programming options make the device suitable for a wide range of applications for proximity and vicinity identification systems.

8 RF430FRL152H Reader Firmware Example

This section provides a high-level overview of the firmware flow and structure for the MSP430FR4133 firmware example that reads data from the RF430FRL152H, calculates the temperature based on the data, and displays it on the LCD screen of the MSP-EXP430FR4133 LaunchPad development kit.

The firmware example can be downloaded from http://www.ti.com/lit/zip/sloa233.

8.1 Firmware Summary

Figure 5 shows the flowchart that the firmware follows. The flow starts with initializations before moving into the main loop to execute the application firmware.

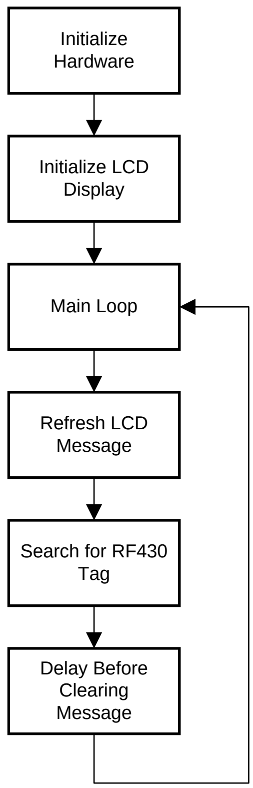


Figure 5. Main.c Flowchart

8.1.1 Hardware Initialization

The initialization sequence for the MSP430FR4133 (see Figure 6) begins with setting up the hardware. This process includes setting up all peripherals for the MSP430 as well as powering up the TRF7970A and initializing the relevant registers for it. The LCD also needs to be initialized through the LCD driver peripheral of the MSP430 MCU.

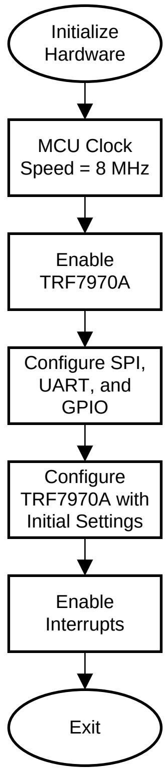


Figure 6. Hardware Initialization Flowchart

After initializations are complete, then the main loop begins to execute. The main loop consists of three steps.

1. Refresh LCD message to a default message.
2. Scan for NFC tags to find any RF430FRL15xH tags and read data from them.

   If an RF430FRL15xH tag is found and temperature data is received, the data is displayed on the LCD during this process, which is detailed below.
3. Execute a delay to allow users to read the temperature data.

   This delay can also be seen as a simulation for additional application-specific processes that would take up time between NFC tag detection.

8.1.2 RF430 Tag Detection

Figure 7 shows the full firmware flow for searching for the RF430FRL15xH transponder.

The NFC command for initial tag detection is the Inventory command, which can be used for both single tag and multiple tag detection. In the event of multiple tag detection, the anticollision algorithm manages the multiple tags and resolves all collisions.

Once a tag or multiple tags have been identified, the Get System Information NFC command is used to determine if a tag is a TI RF430FRL15xH transponder. If no tag is detected, then the process exits.

If an RF430FRL15xH transponder has been detected, then it is configured for sensor measurements using the Write Single Block NFC command. The example firmware sends configurations specifically for thermistors, but these configurations can be modified to support other custom applications.

After configuration, the sensor data is read out when ready using the Read Single Block NFC command. At this time, the NFC communication is complete. With the raw data from the RF430FRL15xH transponder, it is possible convert the ADC values to temperature data using the IQmathLib on the MSP430FR4133.

For this example, the temperature of the thermistor is calculated and then the resulting value is sent to the LCD to display the value to the user. When the LCD is updated, then the process exits.

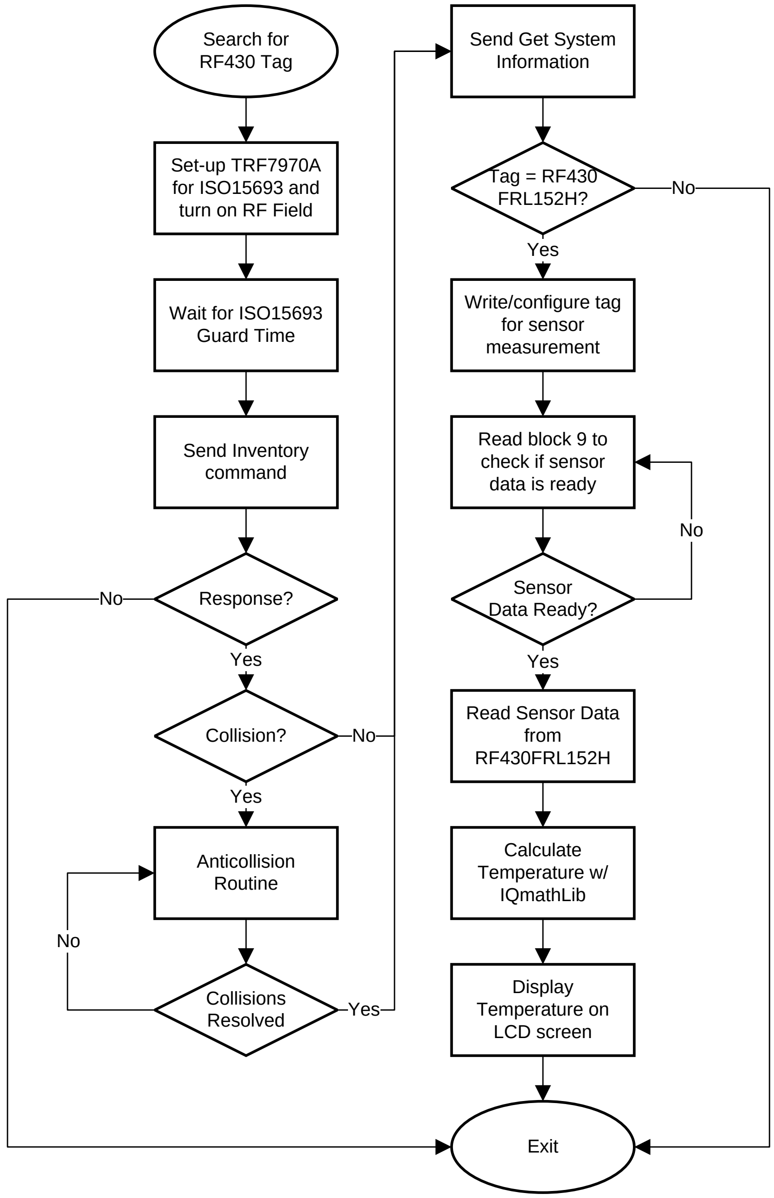


Figure 7. Search for RF430FRL152H Tag Flowchart

8.2 Additional Firmware Comments

The hardware GPIO pin configurations for the DLP-7970ABP BoosterPack plug-in module can be found in the *mcu.h* file. The pins are set by #define statements to allow for easy modification when porting the firmware example to another MCU. Additionally, throughout the firmware example there are a few #define's that can be modified to customize the application.

The first is the #define for ENABLE_HOST in *uart.h*, which is used to toggle all UART functionality. Commenting the #define disables UART functionality. UART allows for NFC tag response data to be sent over UART and viewed with any UART terminal software on a host machine. The drawbacks are that the UART slows down RF communication and that the messages for the UART output are stored in Flash memory and reduce the amount of available Flash memory for application code. TI recommends using this option only when debugging tag responses or when a UART terminal output is applicable for an application. Otherwise it should be disabled to save time and flash memory. The default setting for this is disabled.

The next is the #define for NFC_FIFO_SIZE in *trf79xxa.h*, which determines how large of a buffer is reserved to store FIFO data from the TRF7970A. Changing this allows the TRF79xxA driver to adjust how it handles incoming and outgoing data. The default value for this is set to 120 bytes.

The last is the #define for T5T_MAX_TAGS_DETECTABLE in *iso15693.h*, which determines how many different NFC-V/Type 5 Tags can be detected successfully by the anticollision algorithm. Each tag requires 8 bytes of SRAM memory so the number can be increased or decreased based on available RAM. The default value for this is set to 8 tags, which requires 64 bytes of SRAM memory.

8.3 API Document

An API guide detailing all functions that are used in the firmware example can be found in the associated zip file.

For information specifically about the IQmathLib functions, see the MSP430 IQmathLib User's Guide.

8.4 Hardware Connections

The example software is configured for use with the MSP430FR4133 LaunchPad development kit along with the DLP-7970ABP Boosterpack.

For DLP-7970ABP boards that are older than Version 4.5, the IRQ jumper that is on the board should be set in Position 2.

Table 1 lists the connections between the MSP430FR4133 and the TRF7970A.

Table 1. DLP-7970ABP and MSP-EXP430FR4133 Hardware Connection

DLP-7970ABP Pin	MSP-EXP430FR4133 Pin
MOSI	P5.2
MISO	P5.3
CLK	P5.1
SS	P8.2
EN	P8.3
IRQ	P1.6
ISO14443B LED	P1.5
ISO14443A LED	P1.4
ISO/IEC 15693 LED	P1.3

9 Conclusion

The RF430FRL15xH series of devices have a broad range of uses for various sensor measurements. This application report describes how to implement one common use case for the RF430FRL15xH from both the NFC transponder and NFC reader side. Understanding this specific use case enables users to explore additional and different applications involving the use of sensors with the NFC technology through the RF430FRL15xH devices.

10 References

1. TRF7970A Multiprotocol Fully Integrated 13.56-MHz RFID and NFC Transceiver IC
2. MSP430FR413x Mixed-Signal Microcontrollers
3. RF430FRL15xH NFC ISO 15693 Sensor Transponder
4. NFC/HF RFID Reader/Writer Using the TRF7970A
5. NFC Active and Passive Peer-to-Peer Communication Using the TRF7970A
6. NFC Card Emulation Using the TRF7970A
7. RF430FRL152H Battery-less NFC/RFID Temperature Sensing Patch
8. MSP IQMath Users Guide
9. Panasonic ERT-J1VS104FA data sheet
10. ISO/IEC15693-2:2006(E) (http://www.ansi.org)
11. ISO/IEC15693-3:2009(E) (http://www.ansi.org)
12. NFCForum-TS-DigitalProtocol-1.0 (Digital Protocol) (http://www.nfc-forum.org)
13. NFCForum-TS-Activity-1.0 (Activity Protocol) (http://www.nfc-forum.org)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265