Design Guide: TIDM-NFC-RW
Near Field Communication (NFC) Reader/Writer Reference Design

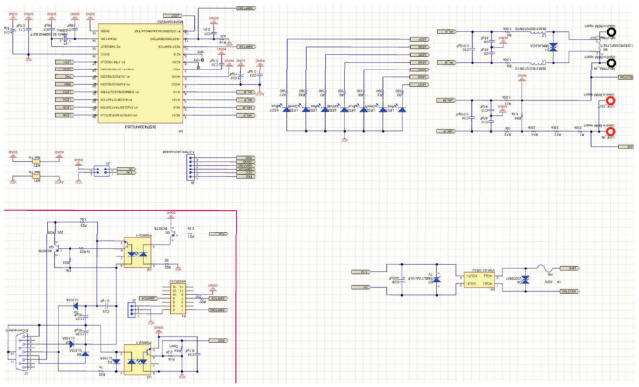
Description

This Near Field Communication (NFC) reference design provides a firmware example for the implementation of an NFC Reader/Writer application using the TRF7970A NFC transceiver. This reference design provides a number of easy-to-use Application Programming Interfaces (APIs), which let users quickly implement NFC Reader/Writer functionality. The API documentation, hardware, and example C code provided lets designers to develop NFC Reader/Writer applications with an MSP430™/MSP432 or easily ported to another MCU of choice.

Resources

TIDM-NFC-RW	Design Folder
TRF7970A, TRF7964A	Product Folder
MSP430F5529, MSP432P401R	Product Folder
TRF7970A BoosterPack	Tool Folder
TRF7970A Target Board	Tool Folder
MSP430F5529 LaunchPad	Tool Folder
MSP430F5529 Experimenters Board	Tool Folder
MSP432P401R LaunchPad	Tool Folder

Ask our TI E2E™ support experts

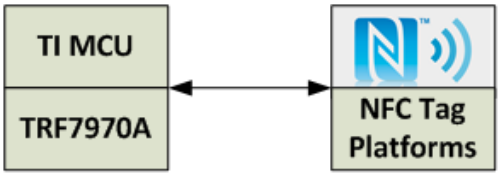


Features

- Able to Read and Write NFC Type 2, 3, 4A, 4B, and 5 Tag Platforms
- Provides Examples of How to Properly Read and Write NDEF-Formatted RTDs for Each Supported NFC Tag Platform
- Includes an Easy to Use GUI to Select Between Individual NFC Modes
- Offers a Flexible Firmware Structure That Allows for Configurable NDEF Applications
- Offers Builds for MSP-EXP430F5529, MSP-EXP430F5529LP, and MSP-EXP432P401R
- Tested for Existing NFC-Enabled Device Interoperability and Against NFC Forum Test Suite.
- Includes Firmware (With GUI), Schematics, and User's Guide

Applications

- Smart Grid Prepayment Infrastructure
- Set-Top Box and Multimedia Configuration
- Automotive Infotainment
- Access Control and Ticketing





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

Copyright © 2015–2020, Texas Instruments Incorporated

1 System Description

The TRF7970A NFC transceiver supports three modes:

- Reader/Writer
- Card Emulation
- Peer-to-Peer

This design focuses on how to use the TRF7970A in Reader/Writer mode. This mode allows an NFC-enabled system to activate and read or write NFC tag platforms. For the TRF7970A, the following tag platforms can be read.

- Type 2 (T2T)
- Type 3 (T3T)
- Type 4A or B (T4T)
- ISO 15693 (T5T)

NFC transceivers that support Reader/Writer mode typically poll to check if a tag or NFC peer is present every 500 ms. This time is known as the period time, which includes the time the transceiver is polling (searching for a tag) plus the time the transceiver is waiting to be activated with the receivers turned on (if it is emulating a tag or is in Peer-to-Peer Target mode). In Figure 1, the transceiver is first polling for Active A and F technology (Active Peer-to-Peer). After it polls for passive A technology (ISO 14443A-3), passive B (ISO 14443B-3), passive F (ISO 18092), and then passive V (ISO 15693).

The system in this design builds on and includes two previous TI Designs (see the following links) and is the third installment of the 3-part series of the NFCLink (stand-alone) offering from TI.

- NFCLink (stand-alone) P2P: http://www.ti.com/tool/TIDM-NFC-P2P
- NFCLink (stand-alone) Card Emulation: http://www.ti.com/tool/TIDM-NFC-CE

Copyright © 2015–2020, Texas Instruments Incorporated

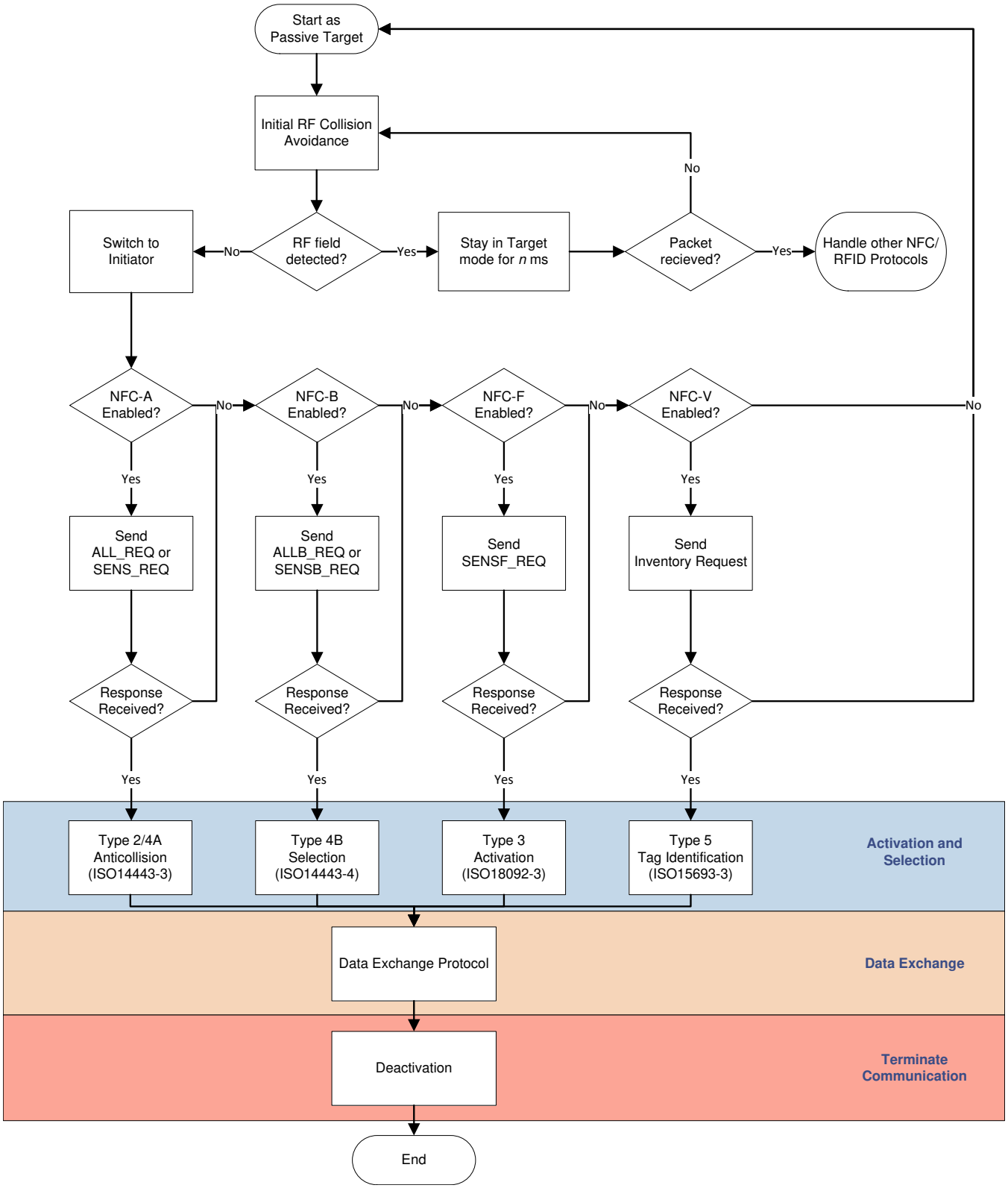


Figure 1. : TIDM-NFC-RW System Flow Diagram

Copyright © 2015–2020, Texas Instruments Incorporated

1.1 NFC/HF RFID Transceiver

1.1.1 TRF7970A

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

The device can perform in one of three modes:

- NFC or HF RFID reader
- NFC peer
- Card emulation mode

Built-in, user-configurable programming options suit the device for a wide range of applications. In this reference design, the TRF7970A device is configured by the firmware automatically for each protocol and then used for reading or writing the NFC tag platforms presented into the vicinity or proximity of the resonant coil antenna connected to the TRF7970A.

The TRF7964A is also a pin-for-pin compatible device, which can drop into this reference design directly. The TRF7964A is used only for the reference design discussed in this guide (Reader/Writer mode).

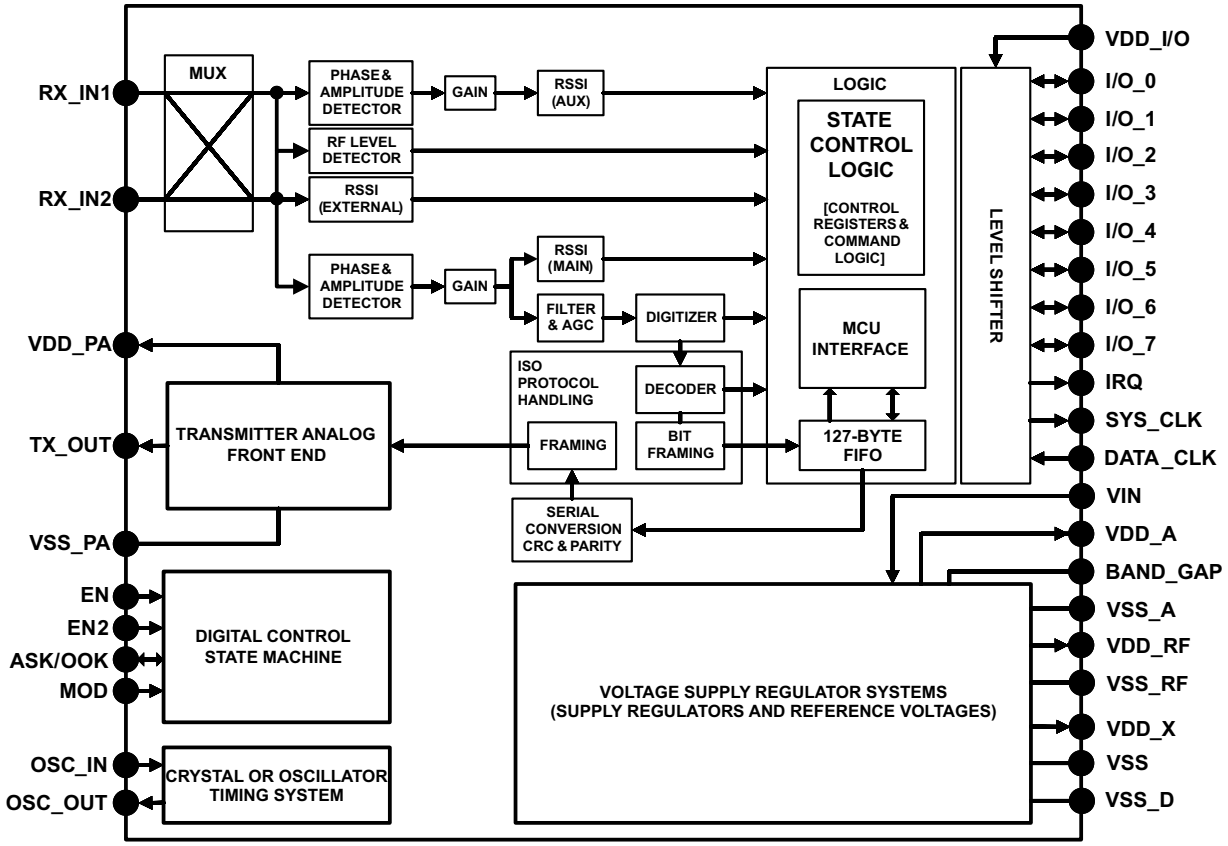


Figure 2. TRF7970A Block Diagram

Copyright © 2015–2020, Texas Instruments Incorporated

1.2 Microcontroller Units (MCUs)

1.2.1 MSP430F5529

The MSP430F5xx family features a 16-bit microcontroller to operate the NFC firmware stack, using the eUSCI_B0 SPI peripheral to interface with the TRF7970A. Figure 3 shows the MSP430F5xx block diagram.

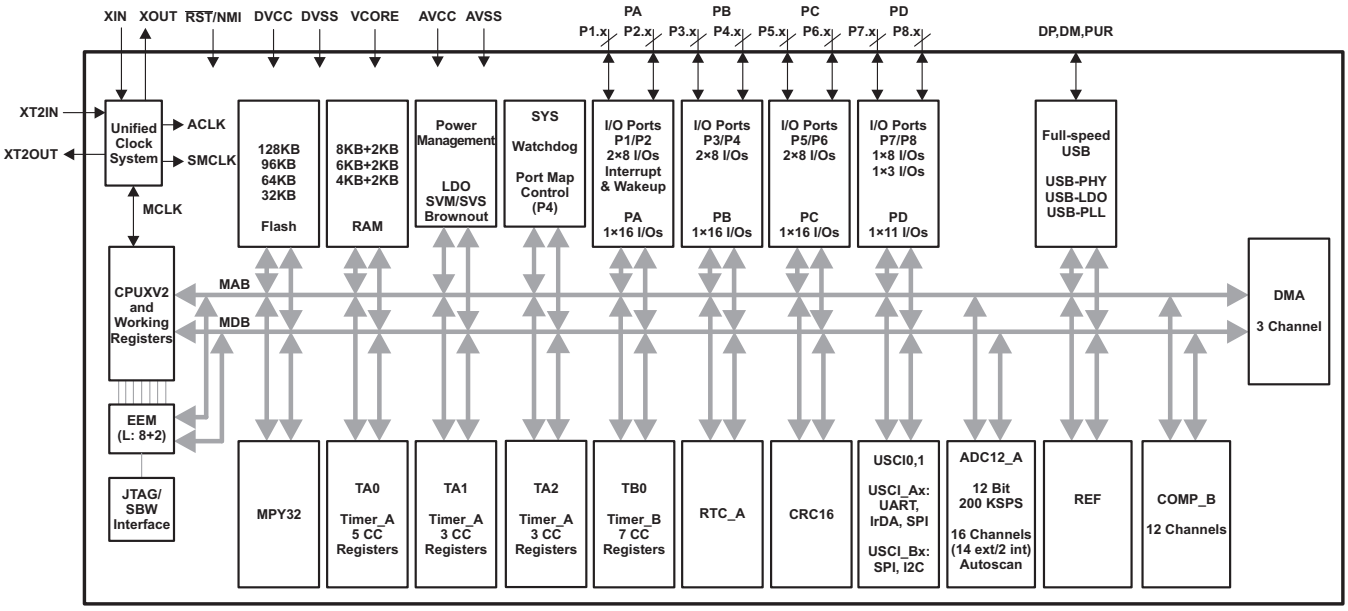


Figure 3. MSP430F5xx Block Diagram

Copyright © 2015–2020, Texas Instruments Incorporated

1.2.2 MSP432P401M

The MSP432P401x family features the ARM® Cortex®-M4 processor to operate the NFC firmware stack. This 32-bit MCU also uses eUSCI_B0 SPI peripheral to interface with the TRF7970A. Figure 4 shows the MSP432 block diagram.

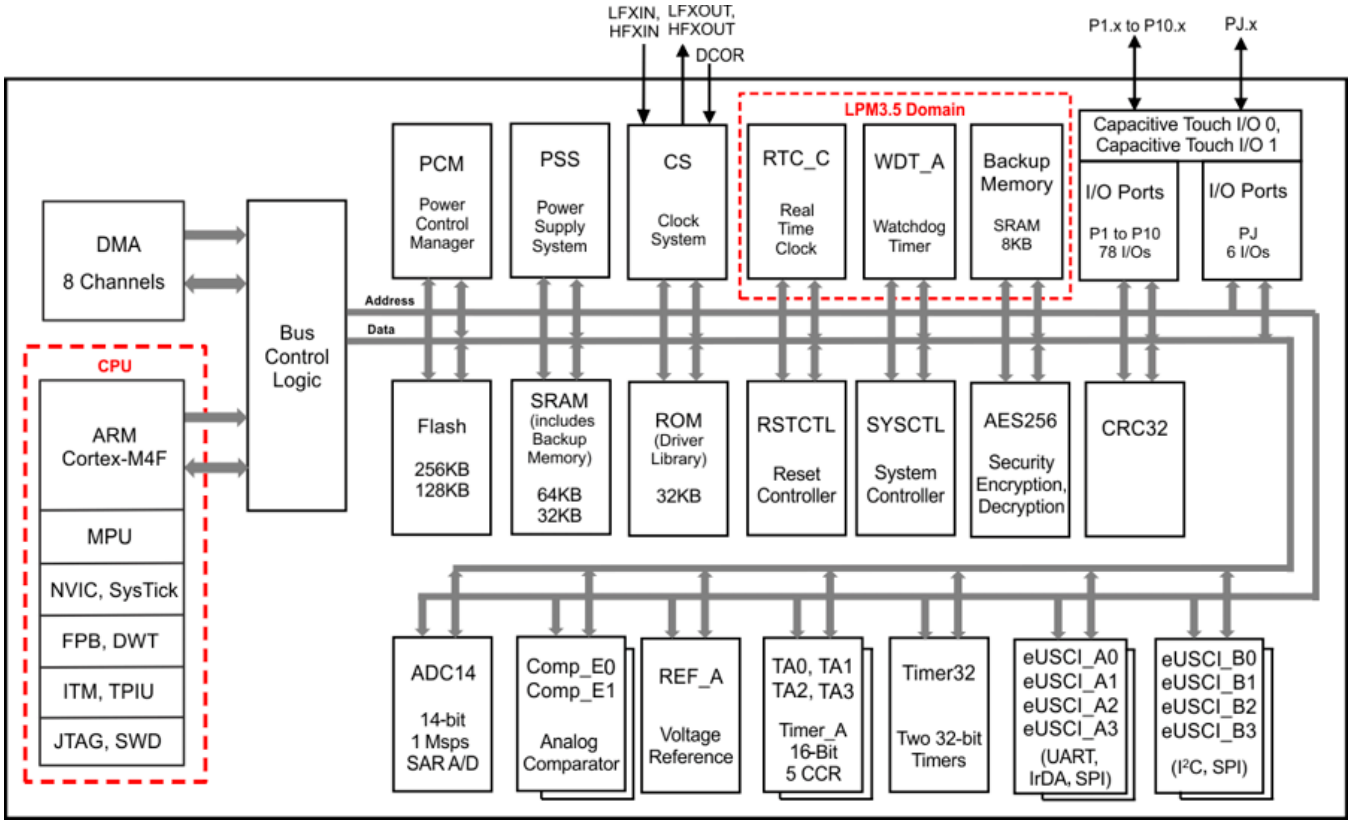


Figure 4. MSP432 Block Diagram

Copyright © 2015–2020, Texas Instruments Incorporated

1.3 Boost Converter

1.3.1 TPS61222DCKT

The TPS61222DCKT is a high-performance and high-efficiency boost converter used on the TRF7970ATB plug-in module to take in 3.3-V DC from the MSP-EXP430F5529 board and create 5-V DC. This is used for VIN potential to the TRF7970A, which allows the transceiver to be operated at full power out (200 mW). The –DCKT variant of the device is used to keep the bill of materials of the module to a minimum. Figure 5 shows the TPS61222DCKT functional block diagram.

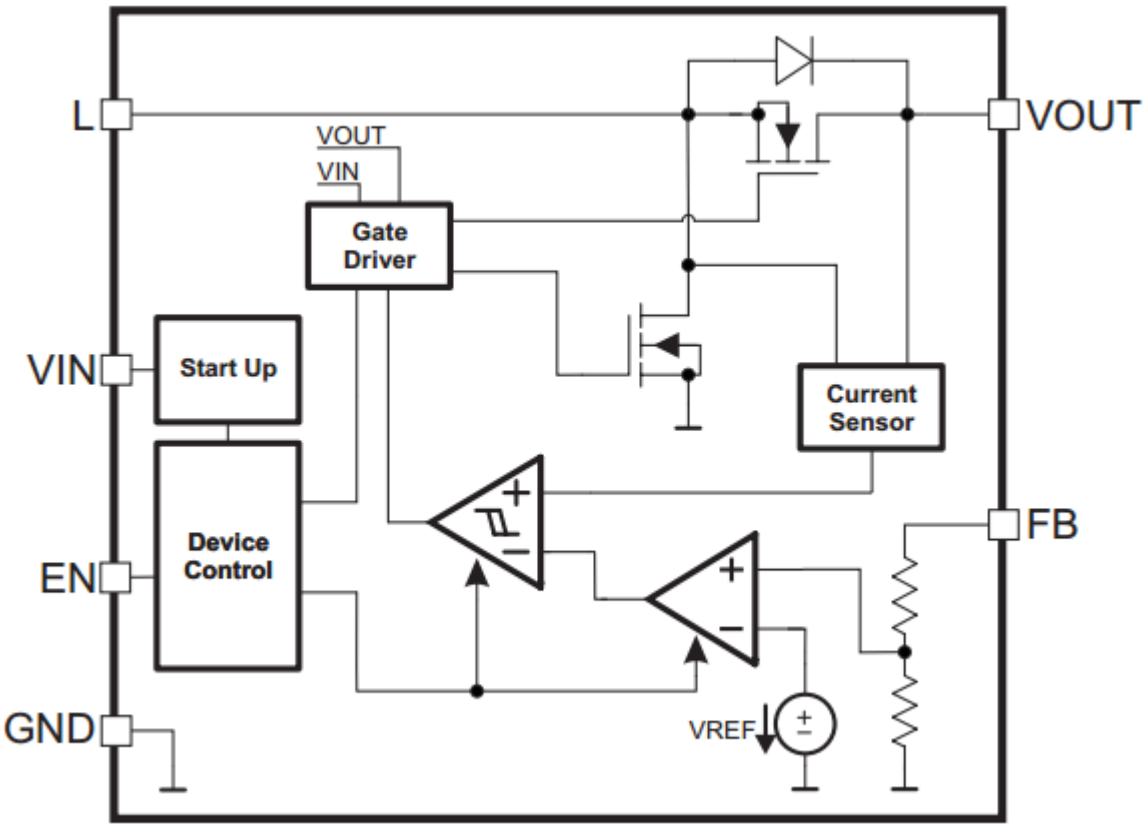


Figure 5. TPS61222DCKT Functional Block Diagram (Fixed Output Version)

2 System Block Diagrams

Figure 6 and Figure 7 show the system hardware block diagram and system firmware layers including PHY (TRF7970A), respectively.

Copyright © 2015–2020, Texas Instruments Incorporated

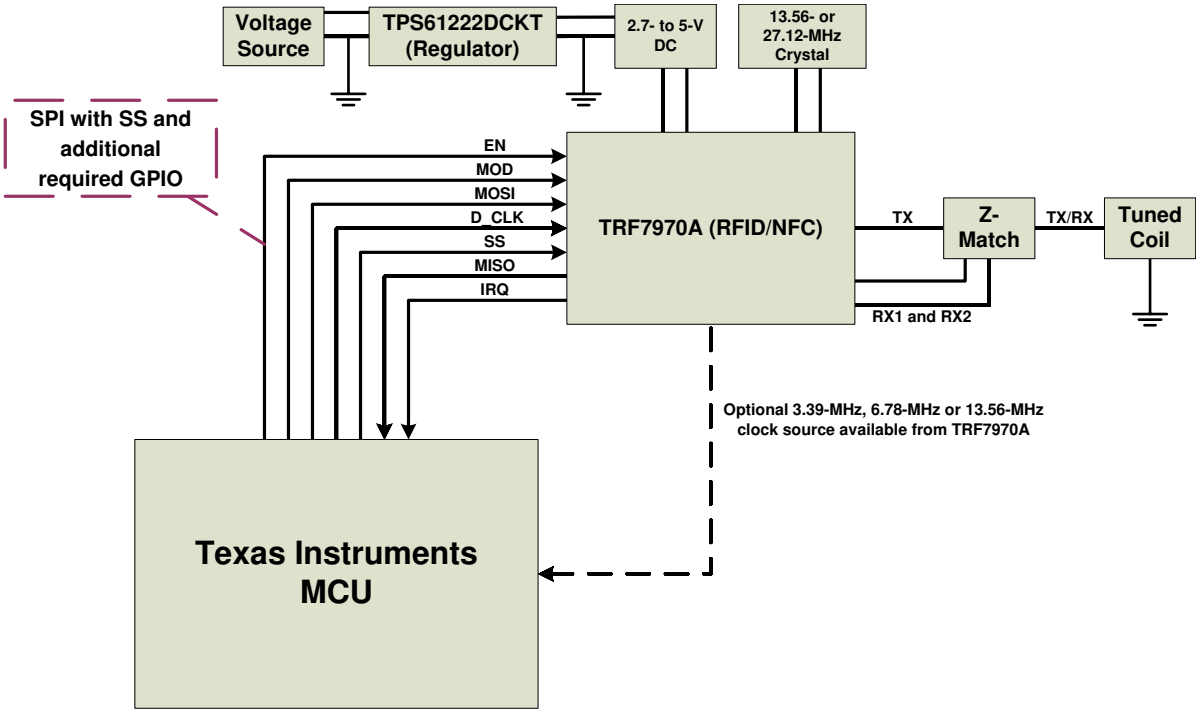


Figure 6. System Hardware Block Diagram

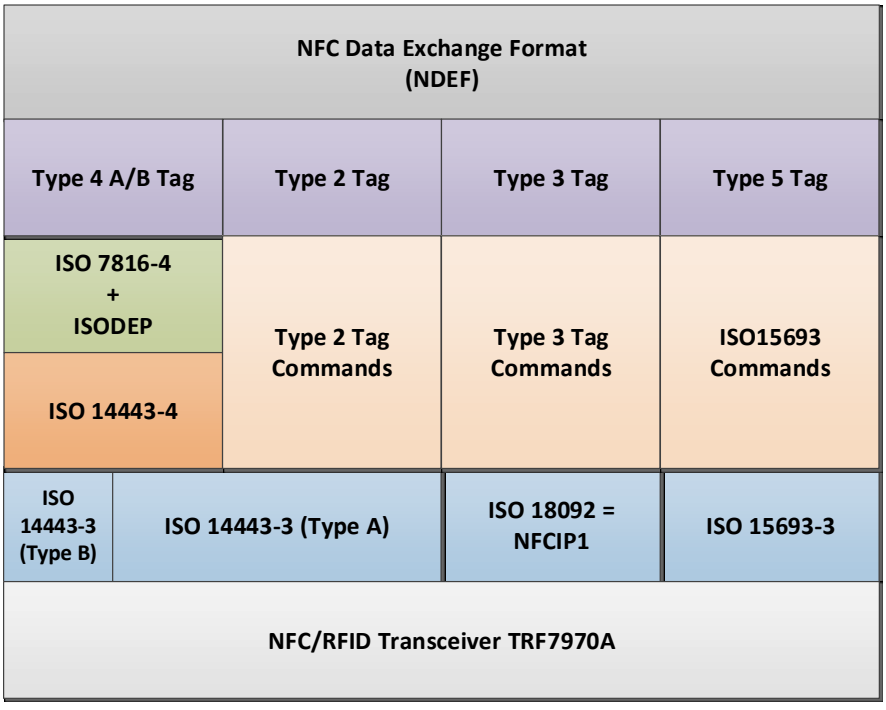


Figure 7. System Firmware Layers, Including PHY (TRF7970A)

Copyright © 2015–2020, Texas Instruments Incorporated

3 Theory of Operation

3.1 Operational Overview

The current Reader/Writer demo on the MSP430F5529 device has two modes:

- A stand-alone mode
- A host mode which requires a system (PC typically) to run the TI NFC Tool GUI

The stand-alone mode can be configured for specific bit rates for each supported tag technology:

- Type 2/4A @ 106/212/424/848 kbps (data rate used depends on if tag supports)
- Type 3 @ 212/424 kbps
- Type 4B @ 106/212/424/848 kbps (data rate used depends on if tag supports)
- Type 5 @ 26.48 kbps

The order in which the firmware polls for the different tag technologies is NFC-A (Type 2/4A), NFC-B (Type 4B), NFC-F (Type 3), and NFC-V (Type 5). If no tags are detected, the polling circles back to NFC-A (Type 2/4A) (see Figure 27 for the switching mechanism while in standalone mode). Polling commands for NFC-A and NFC-B must be sent at 106 kbps, and then the bit rate can be increased after anticollision is completed. For more information, see Figure 8.

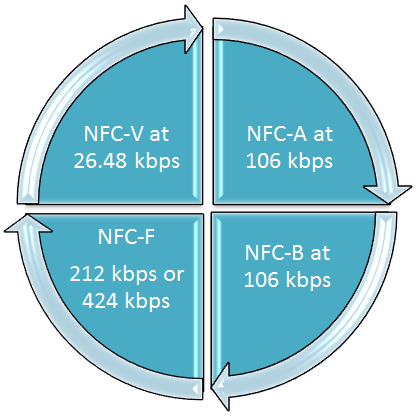


Figure 8. Reader/Writer Polling Mechanism

The second mode requires a host (PC) to run the TI NFC Tool GUI and then connect to either the MSP430F5529 Experimenter's Board or LaunchPad™ through the USB CDC (see Figure 28), or the MSP432P401 LaunchPad through the back channel UART interface (see Figure 29). The GUI lets the user select which modes to enable or disable. When a connection is established with an NFC-enabled device, the GUI switches to the Reader/Writer tab that lets the user view the tag data and send a customized Text or URI RTD for NDEF-formatted tags.

Copyright © 2015–2020, Texas Instruments Incorporated

3.2 MSP430F5529-Based System

The MSP430F5529-based systems are using a built-in USB feature of the MCU to communicate with the host, SPI module, and GPIOs to communicate with the TRF7970A and GPIOs to control the LEDs onboard the BoosterPack™ and the LaunchPad used by this reference design. For more information, see Figure 9.

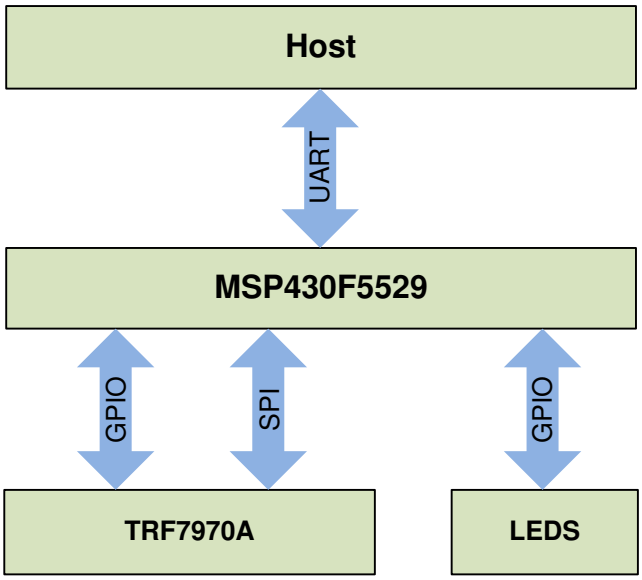


Figure 9. Reader/Writer Demo System Block Diagram for MSP430F5529

Copyright © 2015–2020, Texas Instruments Incorporated

3.3 MSP432P401-Based System

The MSP432P401-based systems are using a built-in UART feature of the MCU to communicate with the host, SPI module, and GPIOs to communicate with the TRF7970A and GPIOs to control the LEDs onboard the BoosterPack and the LaunchPad used by this reference design. For more information, see Figure 10.

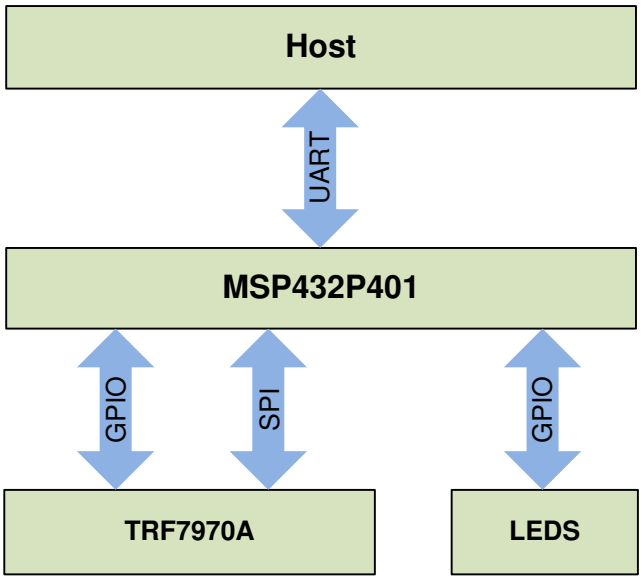


Figure 10. Reader/Writer Demo System Block Diagram for MSP432P401

Copyright © 2015–2020, Texas Instruments Incorporated

4 Getting Started Hardware

4.1 LaunchPad/BoosterPack Configurations

4.1.1 MSP-EXP430F5529LP + DLP-7970ABP

The MSP-EXP430F5529LP LaunchPad is an easy-to-use evaluation module for the MSP430F5529 USB microcontroller. The EVM contains everything needed to start development, including onboard emulation for programming and debugging, as well as onboard buttons and LEDs for quickly adding a simple user interface. Rapid prototyping is easy, thanks to 40-pin access headers and a wide range of BoosterPack plug-in modules. This enables technologies such as wireless, display drivers, temperature sensing, and much more.

The DLP Design NFC/RFID BoosterPack (DLP-7970ABP) is an add-on board designed to fit all of TI's MCU LaunchPads. This BoosterPack lets the software application developer get familiar with the functionalities of TRF7970A Multi-Protocol Fully Integrated 13.56-MHz NFC/HF RFID IC on their TI embedded microcontroller platform of choice without designing the RF section.



Figure 11. MSP-EXP430F5529LP + DLP-7970ABP Configuration

See the TI store bundle at https://store.ti.com/nfclink-bndl.aspx.

Copyright © 2015–2020, Texas Instruments Incorporated

4.1.2 MSP-EXP432P401R + DLP-7970ABP

The MSP432P401R LaunchPad enables you to develop high-performance applications that benefit from low-power operation. The device features the MSP432P401R, which includes a 48-MHz ARM-Cortex M4F processor, 95-µA/MHz active power and 850-nA RTC operation, 14-bit 1MSPS differential SAR ADC and AES256 accelerator. This LaunchPad includes an on-board emulator with EnergyTrace™+ Technology, which means projects can be programmed and debugged without the need for additional tools, while also measuring total system energy consumption

DLP-7970ABP is also used with this LaunchPad, as previously mentioned; the DLP-7970ABP is a part of the LaunchPad/BoosterPack eco-system. For more information, see Figure 12.

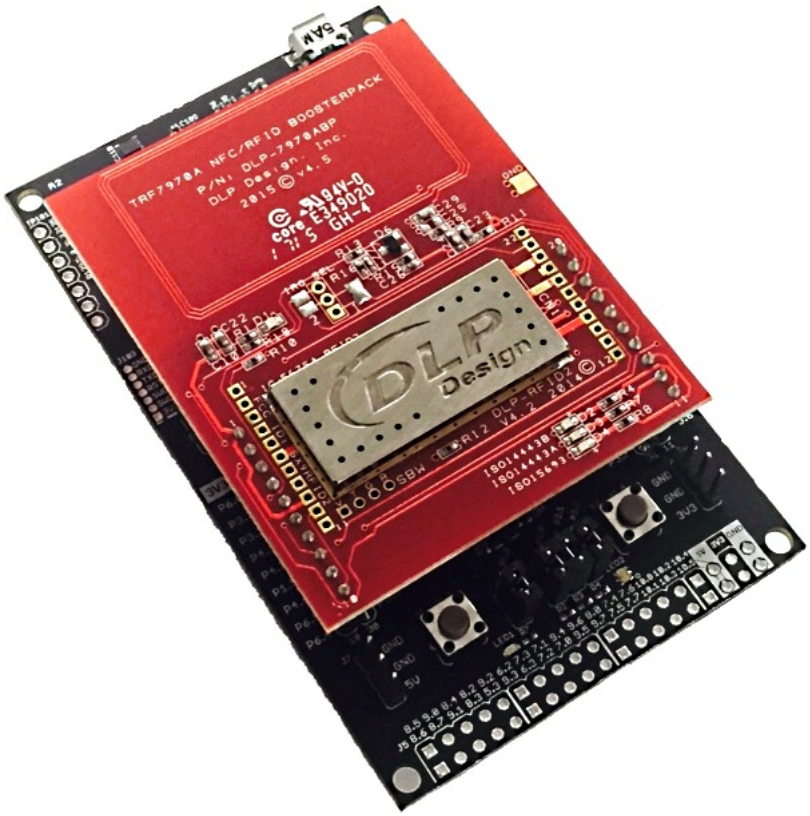


Figure 12. MSP-EXP432P401RLP + DLP-7970ABP Configuration

See the TI store bundle at https://store.ti.com/NFCLINK-BNDL-MSP432.aspx.

Copyright © 2015–2020, Texas Instruments Incorporated

4.2 Experimenter's Board and Target Board Configuration

4.2.1 MSP-EXP430F5529 + TRF7970ATB

The MSP430F5529 Experimenter's Board (MSP-EXP430F5529) is a development platform for the MSP430F5529 device, from the latest generation of MSP430™ devices with integrated USB. The board is compatible with many TI low-power RF wireless evaluation modules such as the TRF7970ATB module. The Experimenter Board helps designers quickly learn and develop using the new F55xx MCUs, which provide the lowest active power consumption of the industry, integrated USB, and more memory and leading integration for other applications, such as energy harvesting, wireless sensing, and automatic metering infrastructure (AMI).

The TRF7970ATB Evaluation Module helps the software application developer become familiar with the functionalities of TRF7970A Multi-Protocol Fully Integrated 13.56-MHz NFC/HF RFID IC with the freedom to develop with any TI MCU. The MSP-EXP430F5529 Experimenter's Board is an example of a development platform. Alternatively, use any other TI embedded microcontroller platform with the EM socket headers populated. The TRF7970ATB is hardwired for SPI communications, supports slave select, TRF7970A Direct Mode 2 (default), Direct Mode 1, and Direct Mode 0 operations. You can also access and gain full control over the TRF7970A EN2 and EN lines, allowing for design and development of ultralow-power, high-frequency (HF) RFID/NFC systems. The module has an onboard boost converter (TPS61222DCKT) that boosts 3.3 VDC to 5 VDC out to TRF7970A IC for 23-dBm (full transmitter power out) operations. For more information, see Figure 13.

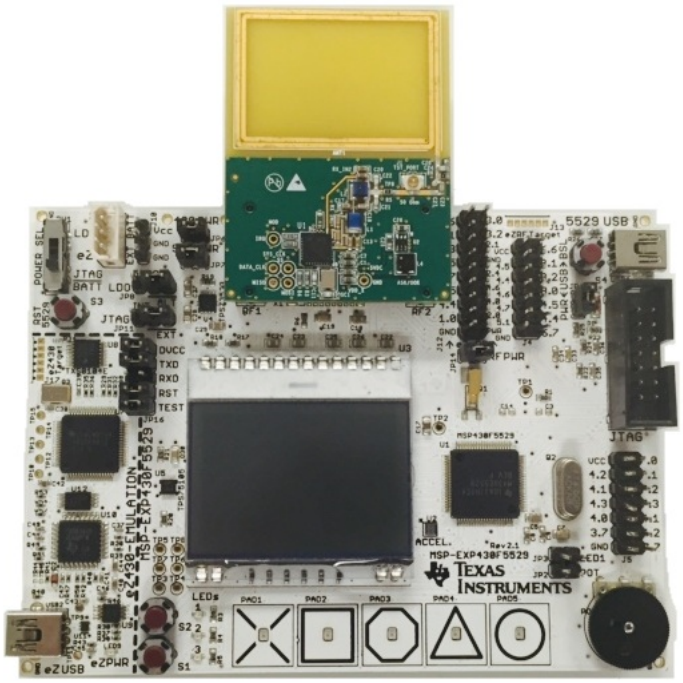


Figure 13. MSP-EXP430F5529 + TRF7970ATB Configuration

See the TI store bundle at https://store.ti.com/NFCLink-Evalution-Kit-P4617.aspx.

Copyright © 2015–2020, Texas Instruments Incorporated

5 Out-of-the-Box Experience

This section covers the steps to load and run the TI NFC Tool Reader/Writer demo with the MSP-EXP430F5529 and DLP-7970ABP. Use these steps for the other hardware platforms.

1. Download the installer from http://www.ti.com/lit/zip/sloa227.
2. Extract the installer.
3. Open the installer.
4. Connect the DLP-7970ABP to the MSP430F5529 LaunchPad (see Figure 14).

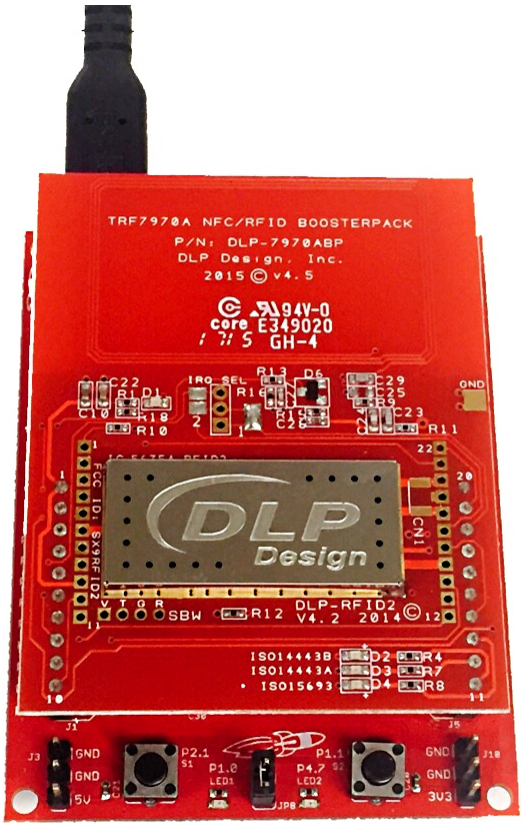


Figure 14. Hardware Configuration (MSP-EXP430F5529LP + DLP-7970ABP)

5. Open Code Composer Studio.
6. Import the Reader/Writer project from the location where the installer placed the project (for example, the default location is: C:\ti\msp430\TRF7970A_RW_1.02.03\examples\boards\MSP-EXP430F5529LP for the other hardware configurations back up one to …\boards).
7. Click  (this button compiles the code and downloads it to the MSP430).
8. Click  when the project is loaded (this executes the code).
9. Click  when the code is running.

NOTE: If CCS is unavailable, use Uniflash (http://processors.wiki.ti.com/index.php/Category:CCS_UniFlash) to load the complied code project into the target MCU board.

10. Run Uniflash when it is installed.

Copyright © 2015–2020, Texas Instruments Incorporated

11. Navigate to File→New Configuration (see Figure 15).

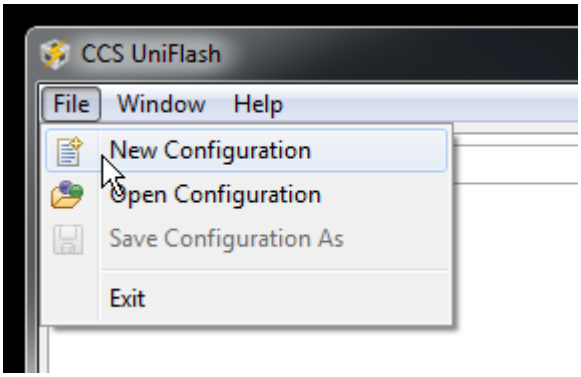


Figure 15. UniFlash New Configuration Drop-down Menu

12. Choose TI MSP430 USB1 and MSP430F5529 (or other target MCUs).
13. Click OK (see Figure 16).

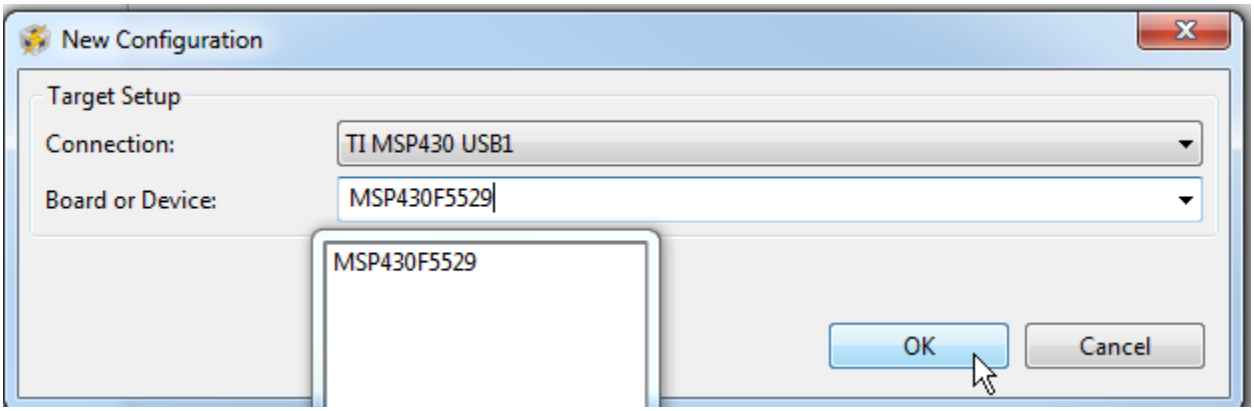


Figure 16. UniFlash Target Setup Menu

NOTE: If UniFlash requests to update the emulator, update it.

14. Choose Load Program from the Program drop-down menu (see Figure 17).

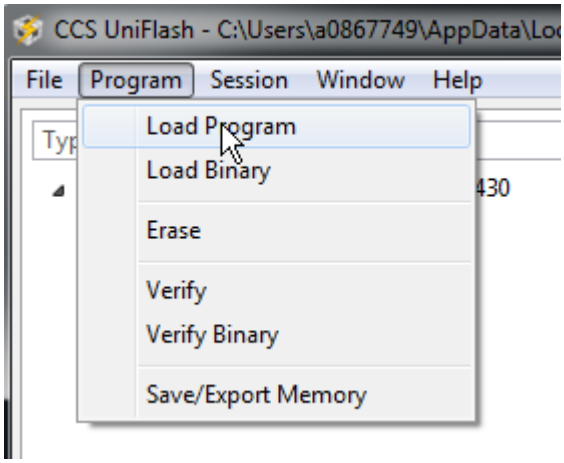


Figure 17. UniFlash Program Dropdown Menu

15. Navigate to the path for Target Binary File (in this case for MSP430F5529 on LaunchPad).

Copyright © 2015–2020, Texas Instruments Incorporated

16. Click OK (see Figure 18).

NOTE: The installer uses the following default location: C:\ti\msp430\TRF7970A_RW_1.02.03\examples\boards\MSP-EXP430F5529LP\F5529LP_TRF7970A_ALL_NFC_MODES\Debug

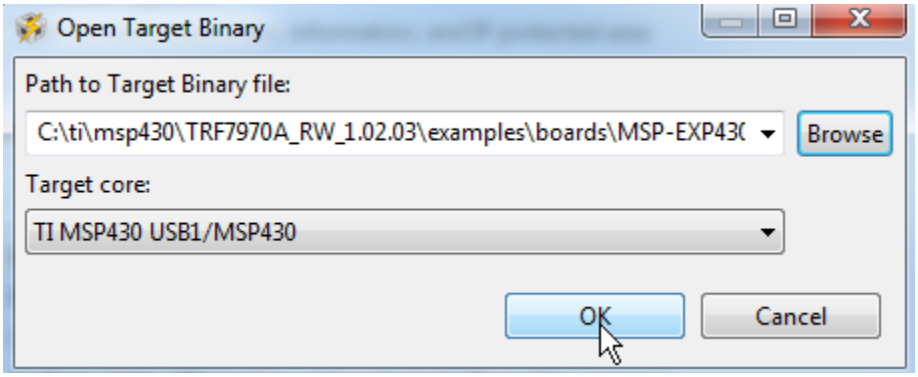


Figure 18. Path to Target Binary File

NOTE: The UniFlash tool loads the program into the MCU flash.

17. Exit the program.
18. Remove the USB connector.
19. Replace the USB connector.
20. Choose from the following modes:
    - For the basic Stand-alone Reader mode, do as follows:
        1. Present any supported NFC/HF RFID tag to the DLP-7970ABP antenna.
        2. Observe that D2 (blue LED, silkscreened ISO 14443B) on DLP-7970ABP is lighted while reading any supported NFC/HF RFID tag.
    - For advanced Stand-alone Reader mode, use the NFC TI Tool GUI as follows:
        – Option 1
            1. Open the NFC TI Tool GUI.

NOTE: By default, the installer places the NFC TI Tool GUI in a folder named C:\ti\msp430\TRF7970A_RW_1.02.03\tools\TI NFC Tool\TI NFC Tool\bin\Debug. This folder is in a folder named Texas Instruments on the Programs file list. See Figure 19.

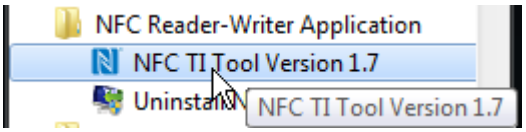


Figure 19. NFC TI Tool v1.7 Executable

2. Select the COM Port from drop-down menu.

Copyright © 2015–2020, Texas Instruments Incorporated

3. Click Connect (see Figure 20).

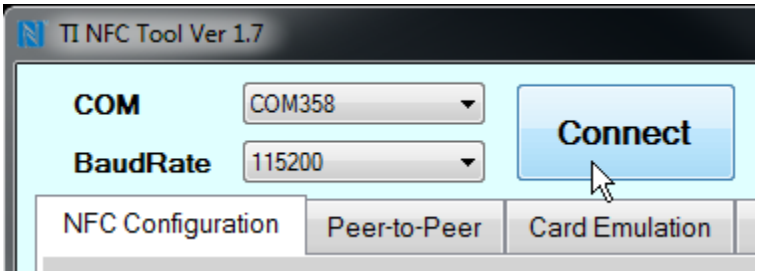


Figure 20. Manual COM Port Selection

– Option 2

1. Select Hardware platform from drop-down menu.
2. Click Auto Connect (see Figure 21).

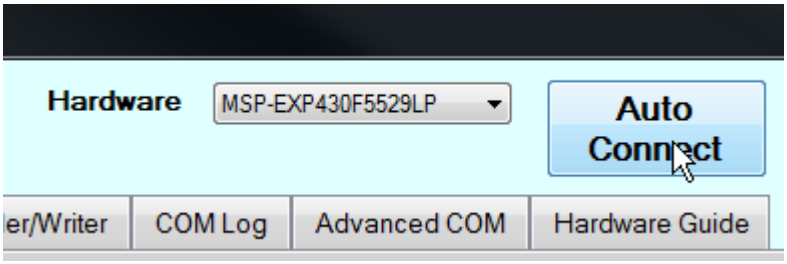


Figure 21. Automatic COM Port Selection

NOTE: When Hardware platform and GUI are connected, the bottom-left corner of the screen shows this status and the firmware version loads. See Figure 22.

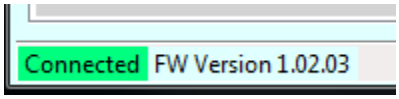


Figure 22. Connection Status With Firmware Version

3. Select the Reader/Writer Protocol modes.

Copyright © 2015–2020, Texas Instruments Incorporated

4. Click Start RW (see Figure 23 where all modes are selected).

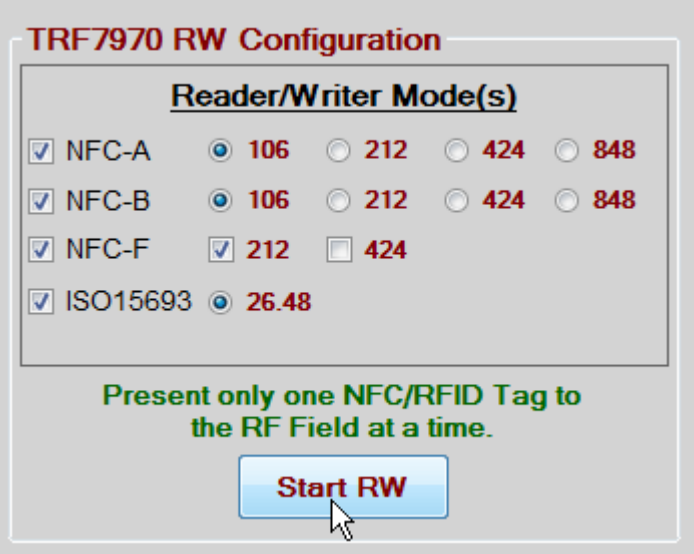


Figure 23. Reader/Writer Protocol Mode Selection

5. Present any NDEF-formatted NFC to the DLP-7970A BoosterPack Antenna (see Figure 24).

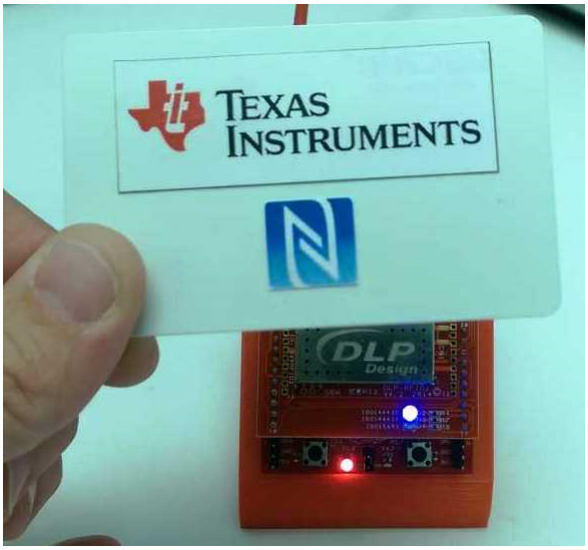


Figure 24. NFC Type 5 Tag Platform Presented to Hardware Platform Antenna

Copyright © 2015–2020, Texas Instruments Incorporated

NOTE: Reading Tag Data: Tag data is output into the Tag Content pane in the Reader/Writer tab of the GUI. NDEF messages that are text or URI RTDs are automatically parsed. All other messages display in raw hex. Tag Type, tag response data rate, technology, RSSI value and UID, PUPI, or ID2 are displayed on upper left-hand side of the Reader/Writer tab window. See Figure 25.

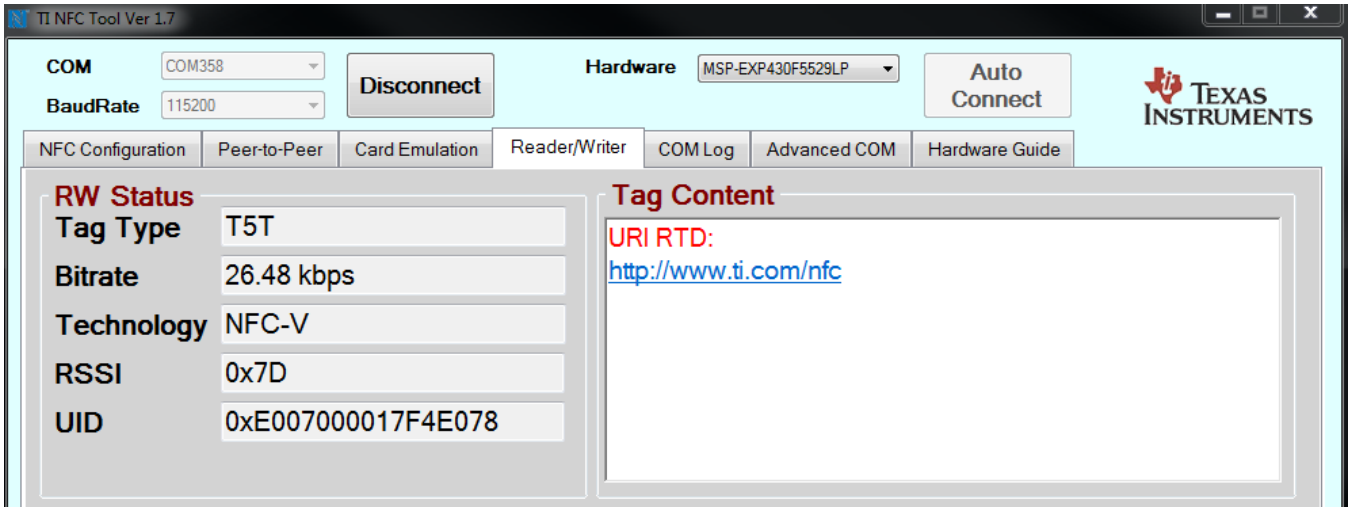


Figure 25. Reading NDEF Content of NFC Type 5 Tag Platform (URI RTD)

6. Select either the Text or the URI tab in the Tag Write pane within the Reader/Writer tab of the GUI.
7. Type a message.
8. Click Write.

NOTE: If the write was successful, a message appears in the Tag Content pane.

9. Remove the tag from the field.

Copyright © 2015–2020, Texas Instruments Incorporated

10. Present the tag to read the new NDEF message (see Figure 26 and Figure 27).

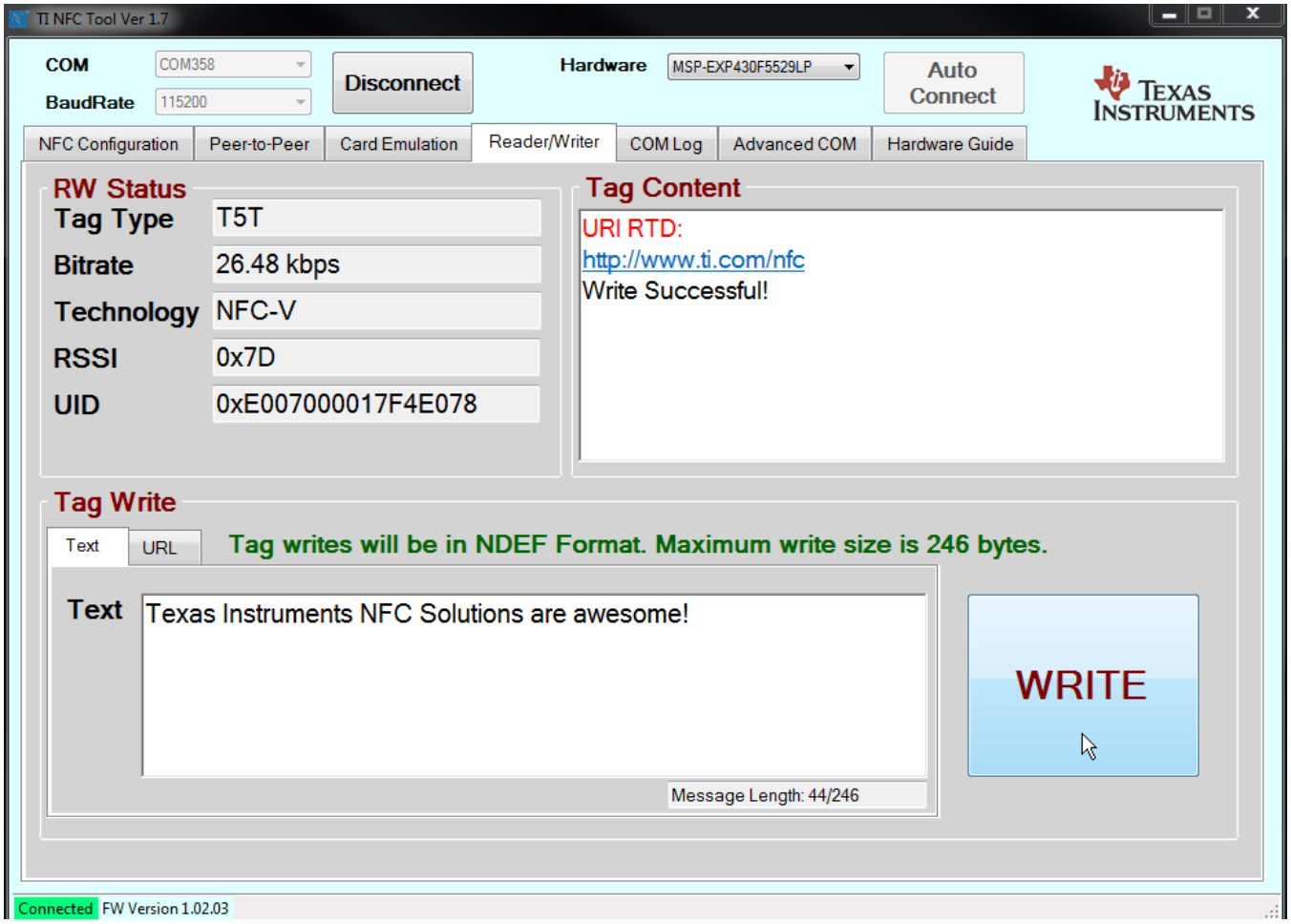


Figure 26. Writing New NDEF Message and RTD Type (Text RTD)

Copyright © 2015–2020, Texas Instruments Incorporated

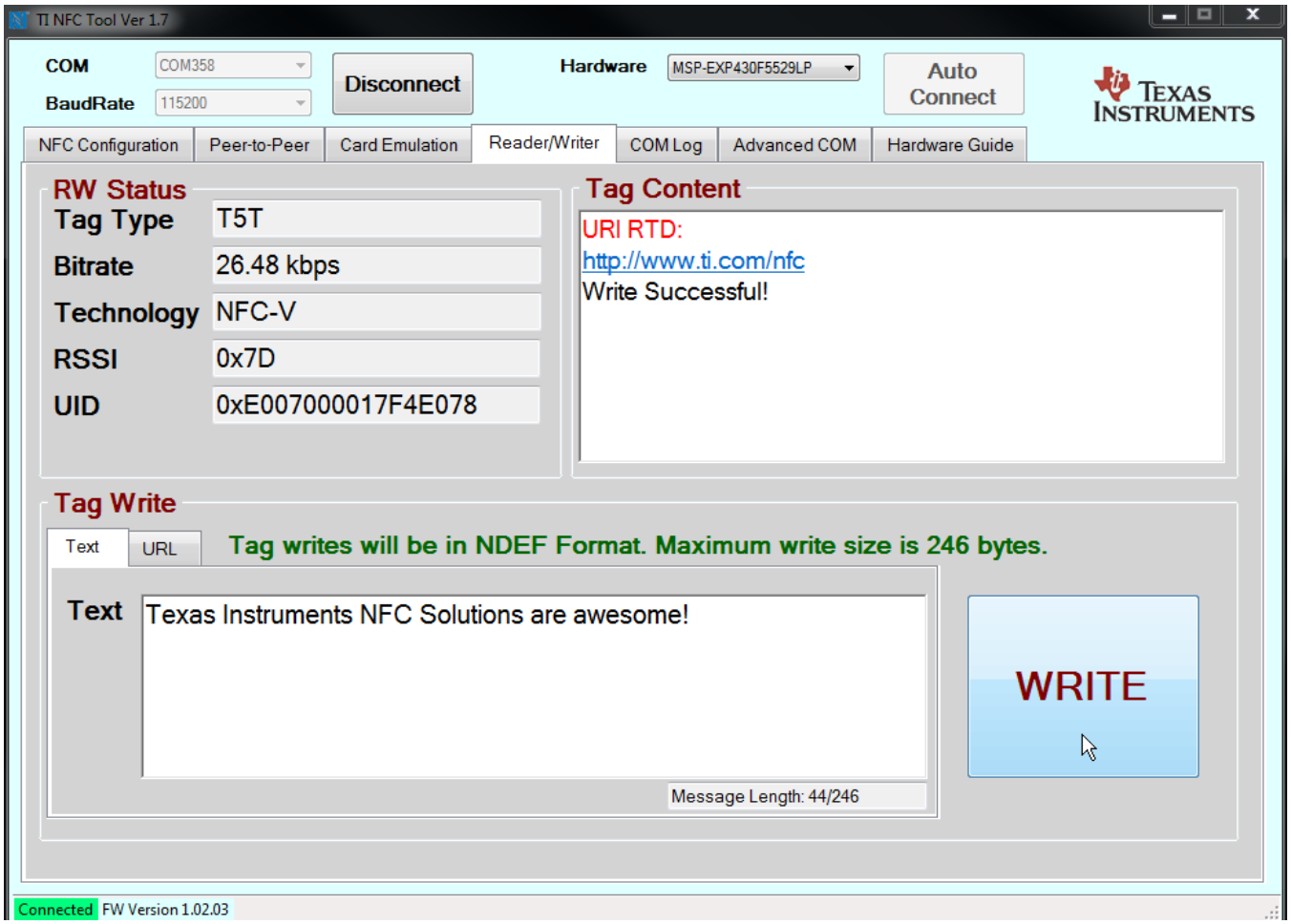


Figure 27. Reading the New NDEF Message

NOTE: When you press Disconnect, the Reader/Writer modes that are enabled are retained and the system operate as it does in standalone mode.

Copyright © 2015–2020, Texas Instruments Incorporated

11. Press the RST button or power cycle the USB connection to reset the board and restore the default program (see Figure 28).

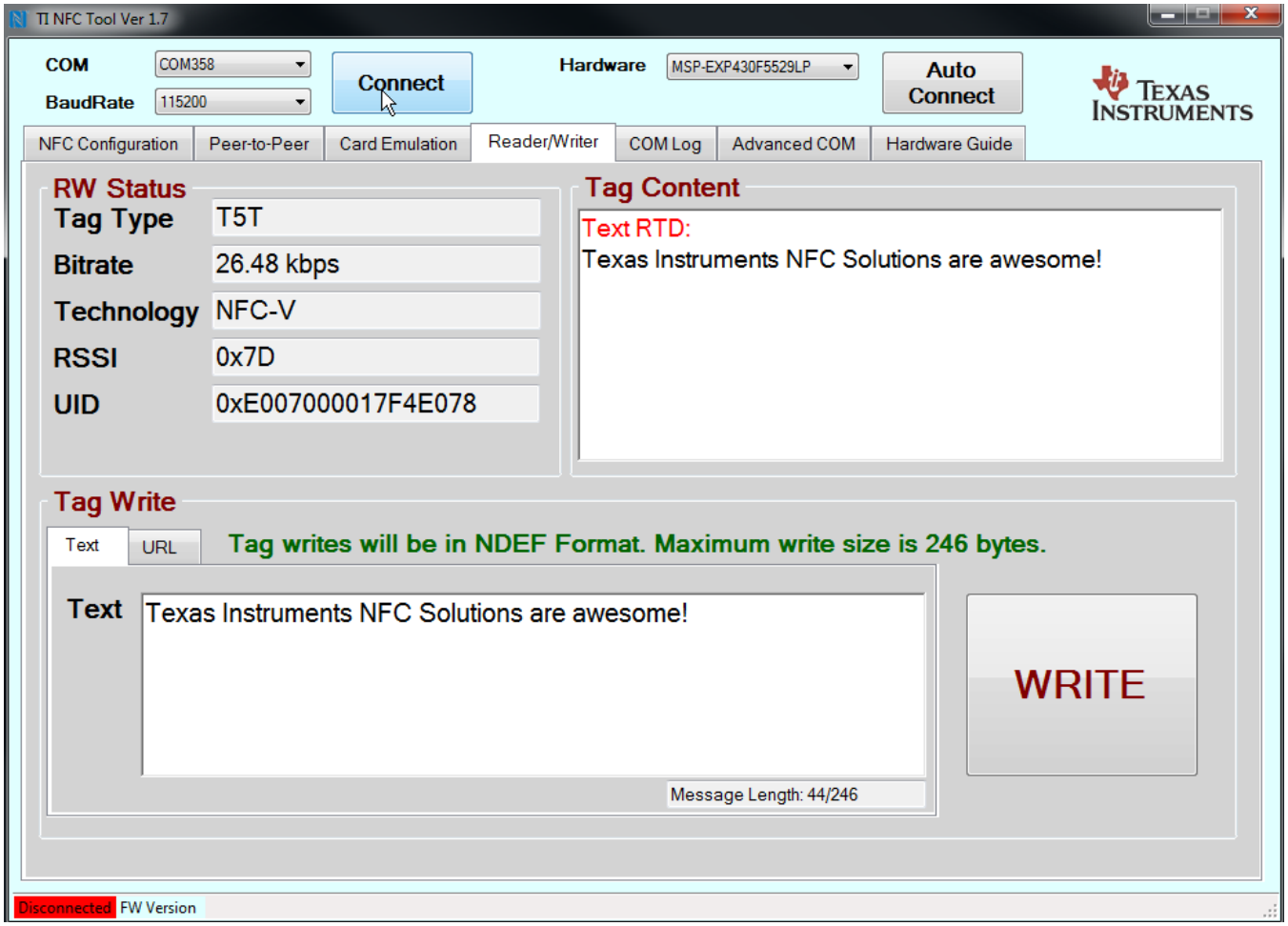


Figure 28. GUI Disconnected from NFC Hardware Platform

Copyright © 2015–2020, Texas Instruments Incorporated

6 Getting Started With the NFCLink (Stand-alone) Tool Firmware

6.1 Introduction

This section explains the APIs in the NFC/RFID layer (see Figure 29) that initializes and handles the Reader/Writer communication. This section also covers how to implement a Reader/Writer application that can send and receive NDEF message to and from an NFC-enabled device.

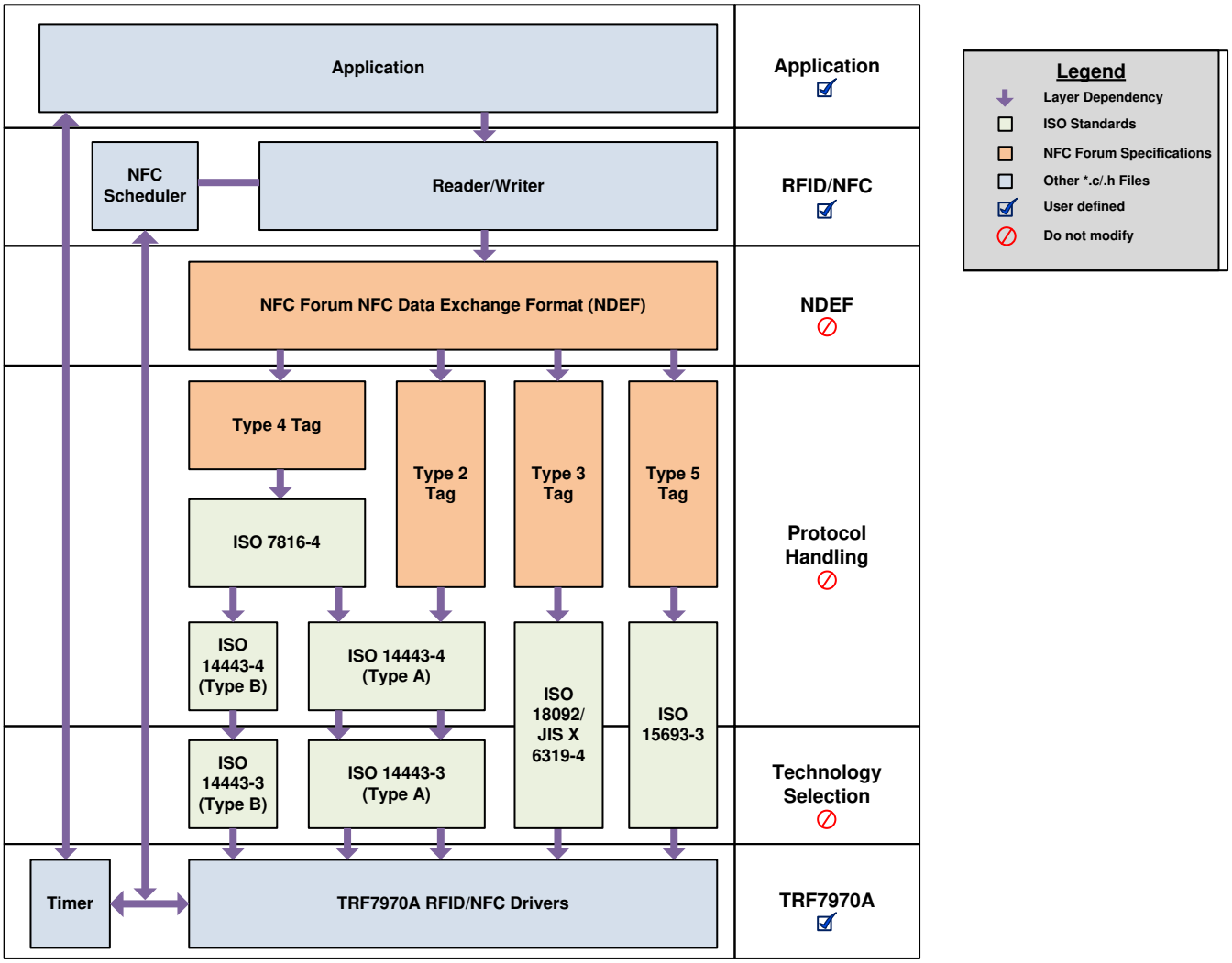


Figure 29. NFCLink (Standalone) R/W NFC Stack Architecture

Copyright © 2015–2020, Texas Instruments Incorporated

6.2 Using and Implementing the Reader/Writer Sample Application

This section explains how to implement a Reader/Writer sample application that uses the S1 and S2 buttons on the MSP430F5529 Experimenter's Board to send different NDEF messages to an NFC-enabled device. Table 1 and Table 2 list the connections between the MSP430F5529 and the TRF7970A for the different MSP432F5529 evaluation platforms. Table 3 lists the connections between the MSP432P401R and the TRF7970A for the MSP430P401R LaunchPad.

Table 1. DLP-7970ABP BoosterPack + MSP-EXP430F5529LP Hardware Connections

DLP-7970ABP Pins	MSP430F5529 LaunchPad Pins
TRF7970A EN1	P4.1
TRF7970A EN2	N/A
TRF7970A IRQ	P2.2
MOSI	P3.0
MISO	P3.1
CLK	P3.2
Slave Select	P4.2
I/O_2	P6.6(1)
I/O_3	P2.0(1)
I/O_5	P1.6(1)

(1) Pin is only needed for using Special Direct Mode.

Table 2. TRF7970ATB + MSP-EXP430F5529 Experimenter's Board Hardware Connections

TRF7970ATB Pins	MSP430F5529 EXP Board Pins
TRF7970A EN1	P2.3
TRF7970A EN2	N/A
TRF7970A IRQ	P2.0(1)
MOSI	P3.0
MISO	P3.1
CLK	P3.2
Slave Select	P2.6
MOD	P2.1
ASK/OOK	P4.7

(1) Requires a jumper between P2.0 and P4.0 on the Experimenter's Board.

Table 3. DLP-7970ABP + MSP-EXP432P401R LaunchPad Hardware Connections

DLP-7970ABP Pins	MSP432P401R LaunchPad Pins
TRF7970A EN1	P6.2
TRF7970A EN2	N/A
TRF7970A IRQ	P3.0
MOSI	P1.6
MISO	P1.7
CLK	P1.5
Slave Select	P6.5
I/O_2	P4.3(1)
I/O_3	P2.5(1)
I/O_5	P4.1(1)

(1) Pin is needed only for using Special Direct Mode.

Copyright © 2015–2020, Texas Instruments Incorporated

6.3 Low-Level Initialization

For the low-level initialization, set the main clock frequency of the MSP430F5529 to 25 MHz to initialize the MCU in MCU_init(). The hardware connections of the TRF7970A and the SPI module of the MSP430F5529 (SPI clock running at 2 MHz) is initialized in TRF79x0_Init(). The NFC_init must then be called to initialize all variables within the NFC stack. When the NFC stack is initialized, calling the NFC_configuration function that sets up the NFC stack to enable the TRF7970A for the desired modes of communication to configure the stack. You can customize the enabled NFC modes by either modifying the function within the firmware or by using the PC GUI interface. See the following MCU and TRF7970A initialization code snippet for an example of how to initialize the TRF7970A in the main(void) program of a C project.

```
#include "msp430.h"
#include "nfc_controller.h"
#include "ndef_image.h"
#include "tag_header.h"

uint8_t g_ui8SerialBuffer[265];

t_sNfcRWMode g_sRWSupportedModes;
t_sNfcRWCommBitrate g_sRWSupportedBitrates;
t_sIsoDEP_RWSetup g_sRWSetupOptions;
uint8_t g_ui8IsoDepInitiatorDID;

uint8_t g_ui8TxBuffer[256];
uint8_t g_ui8TxLength;

void main(void)
{
                tNfcState eTempNFCState;
                tNfcState eCurrentNFCState;
                char pcBytesReceivedString[5];

                            // Reader/Writer Variables
                            t_sNfcRWMode sRWMode;
                            t_sNfcRWCommBitrate sRWBitrate;

                            // Initialize MCU
                            MCU_init();

                            // Enable interrupts globally
                            __enable_interrupt();

                            // Initialize USB Communication
                            Serial_init();

                            // Initialize TRF7970
                            TRF79x0_init();

#ifdef MSP430F5529_EXP_BOARD_ENABLED
                            Buttons_init(BUTTON_ALL);
                            Buttons_interruptEnable(BUTTON_ALL);
#endif

                            TRF79x0_idleMode();

                            // Initialize the NFC Controller
                            NFC_init();

                            // This function will configure all the settings for each protocol
                            NFC_configuration();

                            // Initialize the RW T2T, T3T, T4T and T5 state machines
                            T2T_init(g_ui8TxBuffer,256);
                            T3T_init(g_ui8TxBuffer,256);
```


Copyright © 2015–2020, Texas Instruments Incorporated

```
T4T_init(g_ui8TxBuffer,256);
T5T_init(g_ui8TxBuffer,256);
```

6.4 Reader/Writer NFC Stack Configuration

Set the bits for each Reader/Writer mode with the sRWSupportedModes variable in the NFC_configuration function to initialize the Reader/Writer NFC stack. For each NFC mode, customize which bit rates are supported by the firmware. For the Reader/Writer Stack Configuration code, the sRWSupportedModes variable enables ISO 14443A/NFC-A with bit rates of 106 kbps and 848 kbps selected and ISO 15693 with a bit rate of 26.48 kbps selected.

For NFC-A and NFC-B, use the 106-kbps bit rate even if you desire higher bit rate because the tags can be selected only with 106-kbps communication. When the tag selection is complete, the firmware increases the bit rate provided if the higher bit rate is supported by the tag.

```
// Enable Reader Writer Supported Modes
g_sRWSupportedModes.bits.bNfcA = 1;
g_sRWSupportedModes.bits.bNfcB = 0;
g_sRWSupportedModes.bits.bNfcF = 0;
g_sRWSupportedModes.bits.bISO15693 = 1;

// NFC-A Bit rates
g_sRWSupportedBitrates.bits.bNfcA_106kbps = 1;    // Must be enabled if bNfcA is set
g_sRWSupportedBitrates.bits.bNfcA_212kbps = 0;
g_sRWSupportedBitrates.bits.bNfcA_424kbps = 0;
g_sRWSupportedBitrates.bits.bNfcA_848kbps = 1;

// NFC-B Bit rates
g_sRWSupportedBitrates.bits.bNfcB_106kbps = 0;                // Must be enabled if bNfcB is set
g_sRWSupportedBitrates.bits.bNfcB_212kbps = 0;
g_sRWSupportedBitrates.bits.bNfcB_424kbps = 0;
g_sRWSupportedBitrates.bits.bNfcB_848kbps = 0;

// NFC-F Bit rates
g_sRWSupportedBitrates.bits.bNfcF_212kbps = 0;
g_sRWSupportedBitrates.bits.bNfcF_424kbps = 0;

// ISO15693 Bit rates
g_sRWSupportedBitrates.bits.bISO15693_6_7kbps = 0;    // Not supported
g_sRWSupportedBitrates.bits.bISO15693_26_48kbps = 1;
```

6.5 Activation

After configuration, the NFC_run function is called to run the NFC stack that polls for the enabled technologies and then goes through activation and selection for the first tag from which it receives a reply. When the tag is properly activated and selected, the application calls the state machines required to read the data from the tag.

6.6 Reading and Writing Tags

Four state machines are available for reading and writing tags:

- T2T_stateMachine
- T3T_stateMachine
- T4T_stateMachine
- T5T_stateMachine

Because Type 4A and Type 4B tags use the same memory structure, the same process can be used to read and write either T4TA or T4TB platforms after activation and selection.

When a state machine is called, it reads a tag of that technology by default. The state machine checks for NDEF content on the tag. If the state machine finds an NDEF message, it reads the message. If no NDEF content is found, the state machine reads the raw data from the tag. The firmware outputs the received data to the USB interface to display the data on the TI NFC Tool GUI.

Copyright © 2015–2020, Texas Instruments Incorporated

If you must access the read data for an application-specific purpose, you can copy the data read from the tag by finding the correct read within the state machine. TI designed each state machine to output-received tag data to the USB interface. You can find the most recently received data at any Serial_printBuffer function call.

To keep the memory sizes at reasonable levels, the amount of data stored from reading tags at any time is limited. At the top of each state machine is a declaration for buffer labeled as g_pui8TXTRxBuffer, where the X represents the tag type number. When a tag is read, the state machine goes into an idle state. During this state, the state machine can enter a write state to write new data onto the tag. This behavior can be modified.

When the tag state machine has finished reading and writing the tag, the machine exits and returns to the main application code. No further polling, reads, or writes occur until the tag is removed from the RF field. When the tag is removed, the reader reinitializes the state machines to reset them to resume polling for each enabled technology.

7 Reader/Writer Interoperability Results

This section describes the results of the interoperability between the existing TRF7970A Reader/Writer stack and the list of NFC/RFID tags referenced in the *NFC/HF RFID Reader/Writer* application report (SLOA227) . Use the legend in Table 4 for the results in Table 5.

Table 4. Legend for NFC/RFID Tag Support

Legend	
Supported	√
Not supported	X

Table 5 contains the results from the tests to validate the interoperability of the Reader/Writer firmware stack with various existing tag technology types and NFC devices currently on the market.

Table 5. NFC/RFID Tags Currently Supported by Reader/Writer Mode

NFC/RFID Tags	Read	Write
Type 1	X	X
Type 2	√	√
Type 3	√	√
Type 4A	√	√
Type 4B	√	√
RF430CL33xH	√	√
Type 5	√	√
RF430FRL15xH	√	√
Tag-It HF-I Standard	√	√
Tag-It HF-I Pro	√	√
Tag-It HF-I Plus	√	√

Copyright © 2015–2020, Texas Instruments Incorporated

8 Design Files

8.1 Schematics

To download the schematics for each board, see the design files at http://www.ti.com/tool/TIDM-NFC-RW.

8.2 Bill of Materials

To download the bill of materials (BOM), see the design files at http://www.ti.com/tool/TIDM-NFC-RW.

8.3 Software Files

To download the software files, see the design files at http://www.ti.com/tool/TIDM-NFC-RW and http://www.ti.com/lit/zip/sloa227.

9 References

1. *TRF7970A* data sheet (SLOS743)
2. *NFC/HF RFID Reader/Writer* application report (SLOA227)
3. *MSP430F5529* data sheet (SLAS590)
4. *MSP432P401* data sheet (SLAS826)
5. ISO/IEC 18092/ECMA-340 (NFCIP – 1) (http://www.ecma-international.org)
6. ISO/IEC 21481/ECMA-352 (NFCIP – 2) (http://www.ecma-international.org)
7. ISO/IEC 7816-4:2005 (E) (ISO7816-4)
8. JIS X 6319-4 (http://www.webstore.jsa.or.jp/)
9. ISO/IEC 14443-3:2009 (E) (ISO14443-3)
10. ISO/IEC 14443-4:2008 (E) (ISO14443-4)
11. ISO/IEC 15693-2:2006 (E) (ISO15693-2)
12. ISO/IEC 15693-3:2009 (E) (ISO15693-3)
13. NFCForum-TS-DigitalProtocol-1.0 (Digital Protocol) (http://www.nfc-forum.org)
14. NFCForum-TS-Activity-1.0 (Activity Protocol) (http://www.nfc-forum.org)
15. NFCForum-TS-RTD_1.0 (NFC Record Type Definition [RTD]) (http://www.nfc-forum.org)
16. NFCForum-TS-RTD_Text_1.0 (Text Record Type Definition) (http://www.nfc-forum.org)
17. NFCForum-TS-RTD_URI_1.0 (URI Record Type Definition) (http://www.nfc-forum.org)
18. NFCForum-TS-NDEF_1.0 (NFC Data Exchange Format [NDEF]) (http://www.nfc-forum.org)
19. NFC Forum T2TOP (Type 2 Tag Operation) (http://www.nfc-forum.org)
20. NFC Forum T3TOP (Type 3 Tag Operation) (http://www.nfc-forum.org)
21. NFC Forum T4TOP (Type 4 Tag Operation) (http://www.nfc-forum.org)
22. NFC Forum Logo (http://nfc-forum.org/our-work/nfc-branding/n-mark/the-n-mark-license/)
23. TI NFC/E2E Community (https://e2e.ti.com/support/wireless_connectivity/f/667)

9.1 Trademarks

MSP430, TI E2E, LaunchPad, BoosterPack, EnergyTrace are trademarks of Texas Instruments.
ARM, Cortex are registered trademarks of ARM Limited.

10 Terminology

ALL_REQ— NFC-A Polling command, equivalent to ISO 14443-3 short frame command WupA (0x52)

ALLB_REQ— NFC-B Polling command, equivalent to ISO 14443-3 command WupB

APDU— Application Protocol Data Unit, used in command-response pairs to exchange I (information), R (receive ready) or S (supervisory) blocks

Copyright © 2015–2020, Texas Instruments Incorporated

ATTRIB— PICC Selection Command, Type B

ATQA— Answer To ReQuest A, ISO 14443-3 term, equivalent to NFC term SENS_RES

ATQB— Answer To ReQuest B, ISO 14443-3 term, equivalent to NFC term SENSB_RES

CC— Capability Container, contains management data and is stored inside a read-only EF file. This EF is inside the NDEF tag application. Default (or basic) file ID for this file (for NFC) is 0xE103. See NFC Type 4 Tag Operation Specification and ISO/IEC 7816-4 for more information.

CE— Card Emulation, one of the three modes offered by NFC devices. In this optional mode of NFC operation, an NFC Forum device is considered as a card emulation platform only when it is emulating a Type 3, Type 4A ,or Type 4B Tag Platform. Emulation of any other cards or tags is beyond the scope of the NFC Forum Brand Promise.

DEP— Data Exchange Protocol, an abstracted operational layer which is using either ISO or NFC protocols to exchange data. Thus, two terms can be created with this acronym: ISO-DEP and NFC-DEP. In the context of this document, after activation and selection of the emulated card and before deactivation, ISO-DEP is used exclusively to exchange data between Reader/Writer (PCD) and Tag Platform (PICC).

EF— Elementary File, a set of data objects, records or units sharing the same file identifier and the same security attributes

File Identifier—Two-byte data element used to address a file

fc— Carrier frequency, in the context of NFC/HF RFID, is 13.56 MHz, ±7 kHz. This is the fundamental transmit frequency of the Reader/Writer (also called PCD).

Initiator— Generator of the RF field and source of the beginning of the NFCIP-1 communication

MIME— Multipurpose Internet Mail Extensions

Modulation Index, m—Signal amplitude ratio of $\frac{peak - minimum}{peak + minimum}$ or $\frac{1-b}{1+b}$, where b is the ratio ratio between the modulated amplitude and the initial signal amplitude. The index m, is defined per protocol type, for both downlink and uplink and is generally expressed as a percentage (for example, Type A uses m = 100%, Type B uses m = 8 to 14%).

NDEF— NFC Data Exchange Format

NFC— Near Field Communication

PCB— Protocol Control Byte, used for conveying information required to control data transmission of blocks during the exchange of command-response APDU pairs. Bit coding of this byte and usage rules are found in ISO/IEC FDIS 14443-4.

PCD— Proximity Coupling Device (also commonly referred to as Reader/Writer)

PICC— Proximity Integrated Circuit Card (also commonly referred to as tag, tag platform, or transponder)

PUPI— Pseudo Unique PICC Identifier (randomly generated or static number returned by PICC as part of the response to REQB, WupB, ALLB_REQ or SENSB_REQ)

RTD— Record Type Definition

SAK— Select AcKnowledge (from ISO 14443-3); in NFC terms this is also called SEL_RES

SDD_RES— Equivalent to ISO 14443-3 response to SDD_REQ, and is complete NFCID1 CLn + BCC (if cascade level 1 [single size UID] or indicates NFCID1 is incomplete in the response and further cascade levels must be completed to obtain complete NFCID1 + BCC.

SDD_REQ— Equivalent to ISO 14443-3 Type A anticollision sequence. Comprised of SEL_CMD, SEL_PAR, and n data bits coded based on cascade level specified by SEL_CMD and calculated by value of SEL_PAR

Copyright © 2015–2020, Texas Instruments Incorporated

SEL_RES— Equivalent to ISO 14443-3 SAK response

SENS_REQ— NFC-A Polling command, equivalent to ISO 14443-3 short frame command REQ_A (0x26)

SENS_RES— NFC-A Polling Response, equivalent to ISO 14443-3 ATQA

SENSB_REQ— NFC-B Polling command, equivalent to ISO 14443-3 command REQ_B

SENSB_RES— NFC-B Polling command response, equivalent to ISO 14443-3 ATQB

SENSF_REQ— NFC-F Polling command

SENSF_RES— NFC-F Polling response

Single Device Detection (SDD)— Algorithm used by the Reader/Writer to detect one of several targets in its RF field (Type A anticollision, from ISO/IEC 14443-3)

T2T— Type 2 Tag

T3T— Type 3 Tag

T4T— Type 4 Tag

T5T— Type 5 Tag

Tag— See Tag Platform

Tag Platform— Responds to Reader/Writer (PCD) commands by using load modulation scheme (passive operation) Throughout the document the use the terms *tag(s)*, *tag platform(s)*, and *transponder(s)* all have the same meaning.

TLV— Type Length Value

Transponder— See Tag Platform

UID— Unique IDentifier

VICC— Vicinity Integrated Circuit Card

Copyright © 2015–2020, Texas Instruments Incorporated

11 About the Authors

ERICK MACIAS is an applications engineer at TI working on the NFC/RFID applications team. His efforts are focused on an embedded firmware and software design. For more than 4 years, Erick has designed and developed reference designs, customer-specific dedicated solutions, and extensive training material for NFC/HF RFID products using the extensive portfolio of TI microcontroller and processor platforms. Additionally, Erick provides in-depth customer-focused support through field sales and applications teams and the E2E Forum. Erick holds a BS in Computer Engineering from the University of Florida and is also pursuing his MS in same discipline.

RALPH JACOBI is an applications engineer at TI working on the NFC/RFID applications team, focused on embedded firmware and application software as it relates to digital and analog hardware. Ralph holds a BS in Electrical Engineering from the University of Florida. He is an Eagle Scout and past president of the Eta Kappa Nu, Epsilon Sigma Chapter. Ralph has been at TI since 2014.

JOSH WYATT is currently the NFC/RFID applications manager at TI. His early background was in airborne, long-range, high-power active and passive detection and tracking systems, radios, magnetic and inertial navigation, computers, and cryptographic instruments. Since 1997, he has been working with ground-based LF, HF, and UHF RFID, contactless payment, and NFC systems. Josh has been at TI since 2002.

Copyright © 2015–2020, Texas Instruments Incorporated

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (October 2015) to A Revision	Page
• Deleted footnote: IRQ is defaulted to P3.0 for DLP_7970ABP V4.5 and newer (SLOA226)	25
• Deleted footnote: IRQ is defaulted to P3.0 for DLP_7970ABP V4.5 and newer (SLOA226)	25

Copyright © 2015–2020, Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2020, Texas Instruments Incorporated