User's Guide
SLOU386A–May 2014–Revised November 2014

NFCLink Firmware User's Guide

NFCLink is a complete Near Field Communications (NFC) firmware and software solution that Texas Instruments has developed in conjunction with a third party. The hardware used is either a combination of MSP-EXP430F5529 and TRF7970ATB, or it is a combination of the MSP-EXP430F5529LP and DLP-7970ABP. The total solution is intended to be used for developing applications that require any or all of the possible NFC modes, with a variety of operating systems such as Windows®, Android™, and Ubuntu™.

Integration of NFC into end equipment is growing rapidly (on a worldwide scale) and spanning across consumer, medical, retail, industrial, automotive, and smart grid application spaces. This solution is a modular platform (firmware and hardware) designed to not only make it easier for the field to promote the TRF7970A and Texas Instruments microcontrollers in these mentioned areas, but also to serve as a powerful development tool for customers to use immediately for integration into their product. The firmware is provided as a project library with API stack examples for all three modes of NFC operations.

Contents

1 Export Control Notice

Recipient agrees to not knowingly export or re-export, directly or indirectly, any product or technical data (as defined by the U.S., EU, and other Export Administration Regulations) including software, or any controlled product restricted by other applicable national regulations, received from disclosing party under nondisclosure obligations (if any), or any direct product of such technology, to any destination to which such export or re-export is restricted or prohibited by U.S. or other applicable laws, without obtaining prior authorization from U.S. Department of Commerce and other competent Government authorities to the extent required by those laws.

2 Description

NFCLink is the market name for a TI based hardware, firmware library, and software stack solution for Near Field Communications (NFC). Integration of NFC into end equipment is growing rapidly (on a worldwide scale) and spanning across consumer, medical, retail, industrial, automotive, and smart grid application spaces.

This solution is a modular platform (firmware and hardware) designed to not only make it easier for the field to promote the TRF7970A and TI microcontrollers in these mentioned areas, but also serve as a powerful development tool for customers to use immediately for integration into their product.

The firmware is provided as a project library with API stack examples for all three modes of NFC operations. The hardware required to demonstrate the capabilities of the solution consists of the following:

- An MSP430F5529 USB Experimenter's Board (MSP-EXP430F5529) and a TRF7970ATB Evaluation Module (plugged onto MSP-EXP430F5529 RF1 and RF2 headers)
- An easy-to-use GUI running on Windows PC for demonstrating the NFC/RFID reader/writer, peer-to-peer, and card emulation modes.
- The OS stack (NFCStack+Eva, from our partner Stollmann) current offerings are running on Windows Mobile 6.5, Windows XP, Windows 7 and Windows 8.

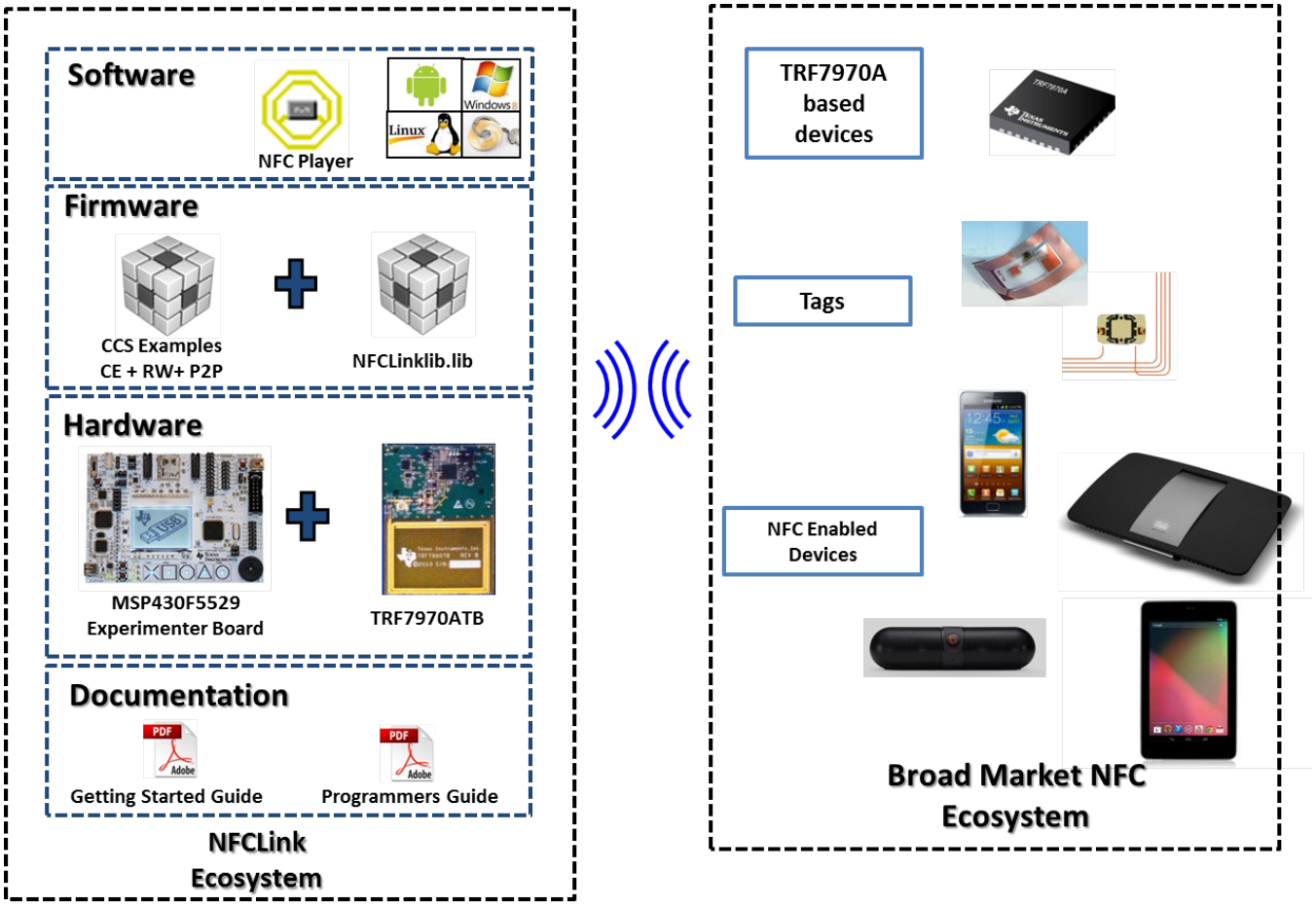

Figure 1. Kit Contents and Target Applications

3 Hardware Requirements

The hardware required to demonstrate the capabilities of the solution consists of the following:

- MSP-EXP430F5529 Experimenters board
- TRF7970ATB (plugged onto MSP-EXP430F5529 RF1 and RF2 headers)
- CCS Project including nfclinklib.lib (for developers)

  OR

  A TI provided binary file (loaded onto the MSP430F5529, for initial demo purposes)
- Instructions for downloading firmware (see Section 7.3.1)
- USB cable (USB-A to Mini-B)
- NFCLink (downloaded from http://www.ti.com/tool/nfclink)

4 Stack Definition and Function

The Stollmann NFaCe+SPA Evaluation Application demonstrates the functionality of the Stollmann NFC stack (NFaCe+SPA) on an application user interface. It supports Intel, ARM, Cortex and other CPUs. NFCLink uses NFCStack + Eva (Evaluation Application).

The package contains the graphical user interface (GUI) and currently supports the following:

- NFC Modes
  - NFC read/write operations
  - Peer-to-peer operations, including SNEP
  - Host-based card emulation
- NFC Tag Platform Types:
  - Type 2 (MIFARE Ultralight™, MIFARE Ultralight C™, Infineon my-d™ move, Infineon my-d move NFC)
  - Type 3 (Sony FeliCa™ Lite (RC-S965), FeliCa Lite-S (RC-S966))
  - Types 4A and 4B (ISO14443A/B)
  - Type V (ISO15693)
- NFC Controller Interface (NCI), with extensions

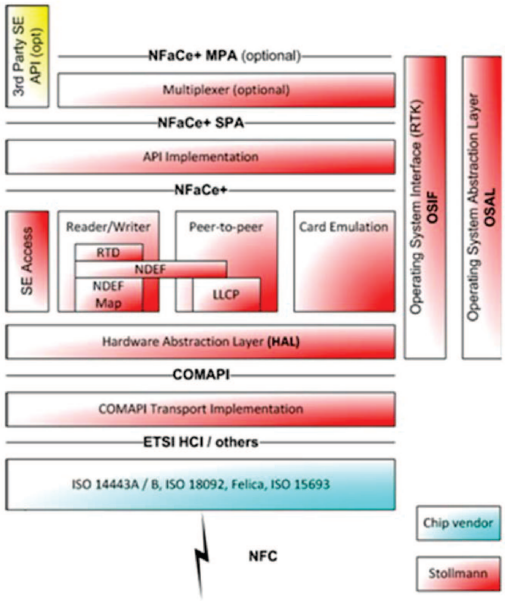

Figure 2. NFCStack + Eva Architecture

5 Integrating the NFCLink Stack

In an actual application, the NFCStack+ would be compiled as:

- A driver component into the OS using applicable compiler and the supplied source code. (host integration / integrated host) NFC module – Run full or parts of the protocol stack on a dedicated CPU withmemory (commonly used for high-volume low-cost, automotive, or POS applications)
- OS to NFC Controller (embedded MCU + TRF7970A) – same as first example, but running the stack on the OS.
  - This is basically what the NFCLink solution is currently.

The simplified block diagram of the NFCLink Architecture is shown in Figure 3, where the host could be A8, A9, ARM, or other MPU, the embedded MCU could be ARM or MSP430™, and the TRF7970A is the NFC transceiver. NFC or RFID devices could be passive tags, other NFC devices (for example, speakers), or NFC-enabled handsets or tablets.

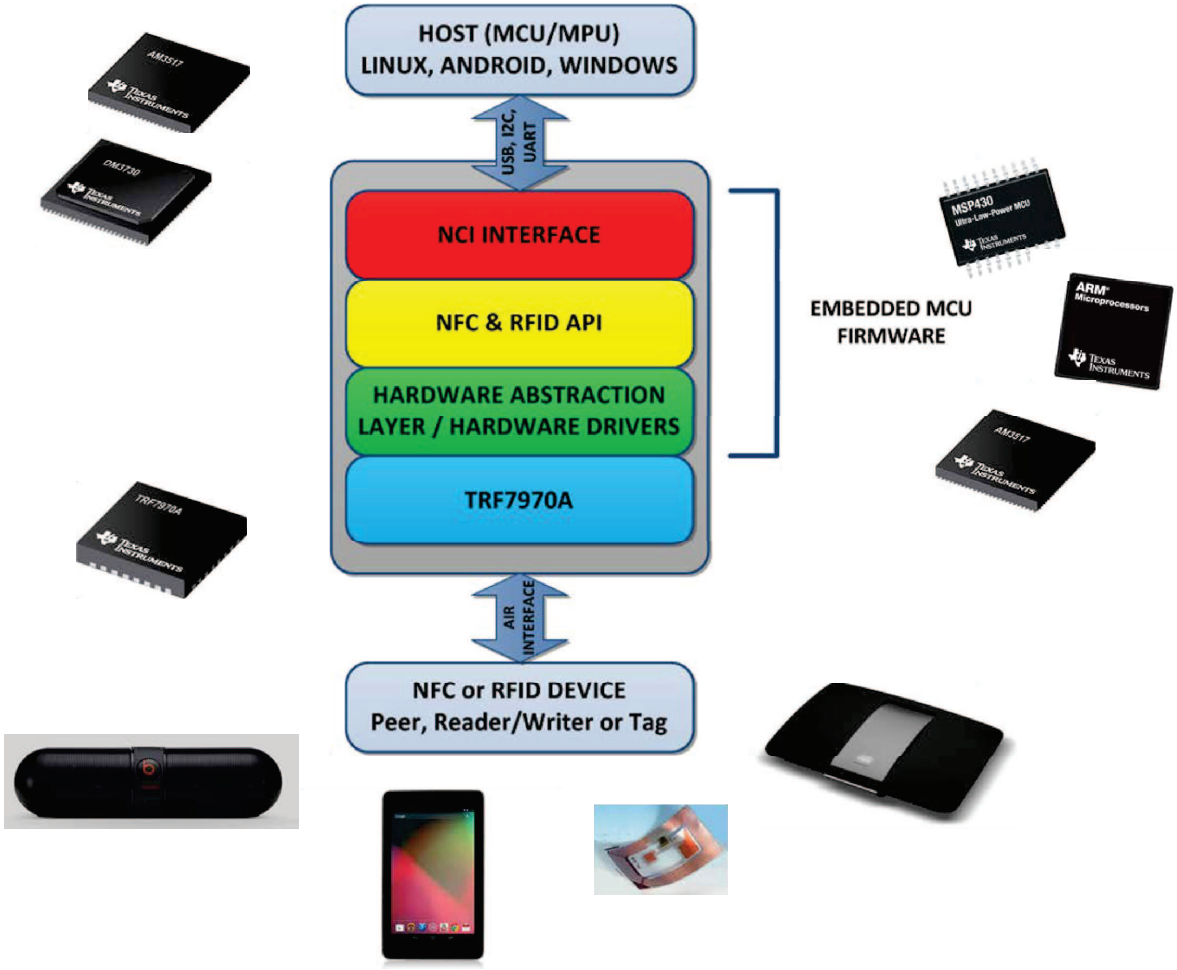

Figure 3. Simplified Block Diagram of NFCLink Architecture

There are MSP430 object code and source code components, with the source code components being the necessary ones needed by the developer to modify for different MSP430 MCUs, while the object code portions are specific to NFC and RFID functions that are completed, thus making it simple for the customer developing a solution just to use it and not have to research and learn all about the low level details of how NFC or RFID works.

Other components of the release are:

- NFCStack+Eva_Windows_R6.1.153.5-Setup.exe – Windows GUI installer
- NFCLink compiled library (nfclinklib.lib)
- CCS Projects
  - RW_P2P_CE1_Example – for USB CDC applications
  - RW_P2P_CE2_Example – for UART applications

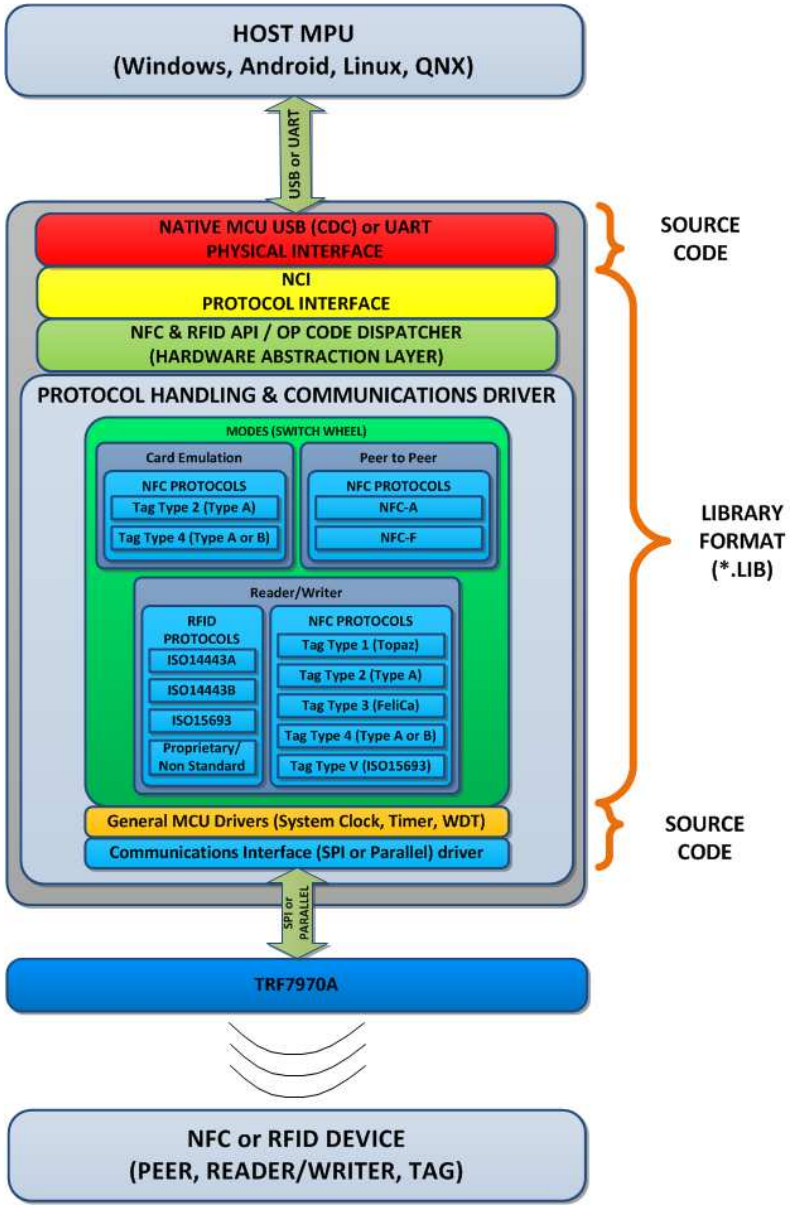

Figure 4. Delivery Format of Available Code

6 Hardware Configuration

Figure 5 shows the NFCLink hardware configuration using the MSP-EXP430F559 and the TRF7970ATB.

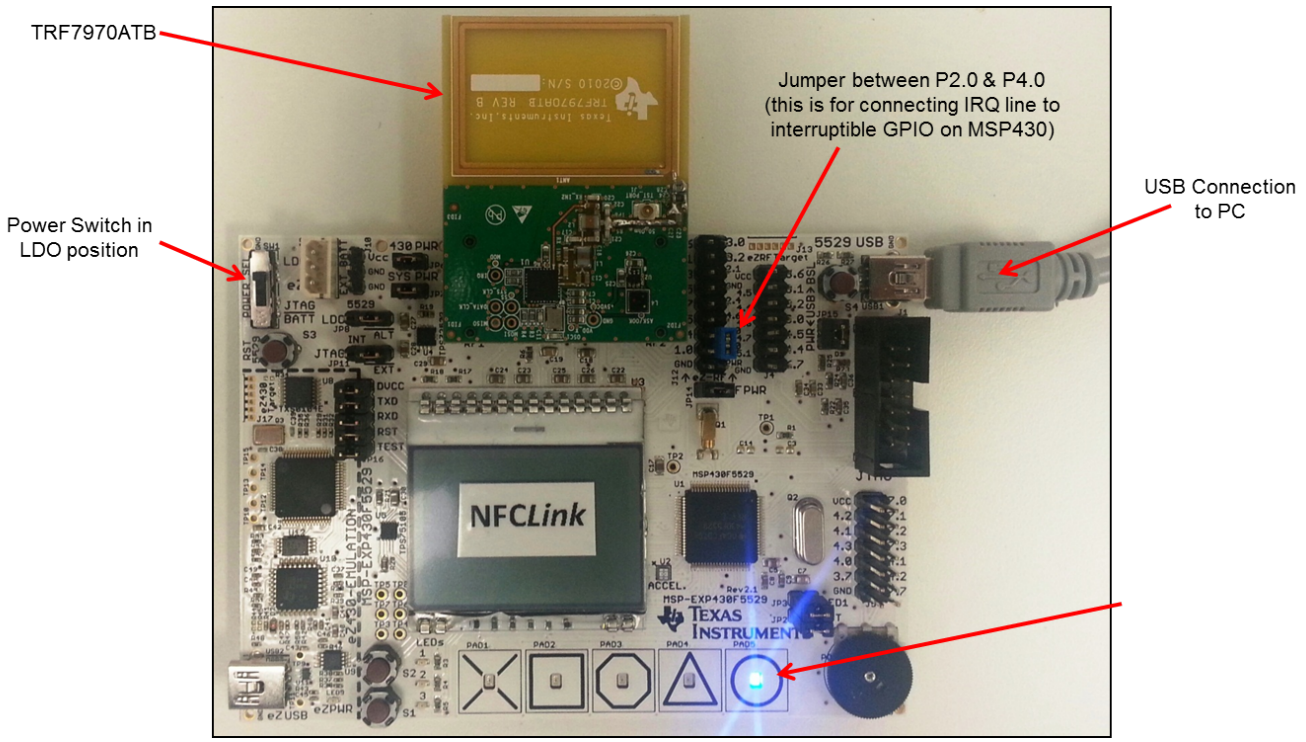

Figure 5. NFCLink Hardware

Figure 6 shows the DLP-7970ABP BoosterPack.

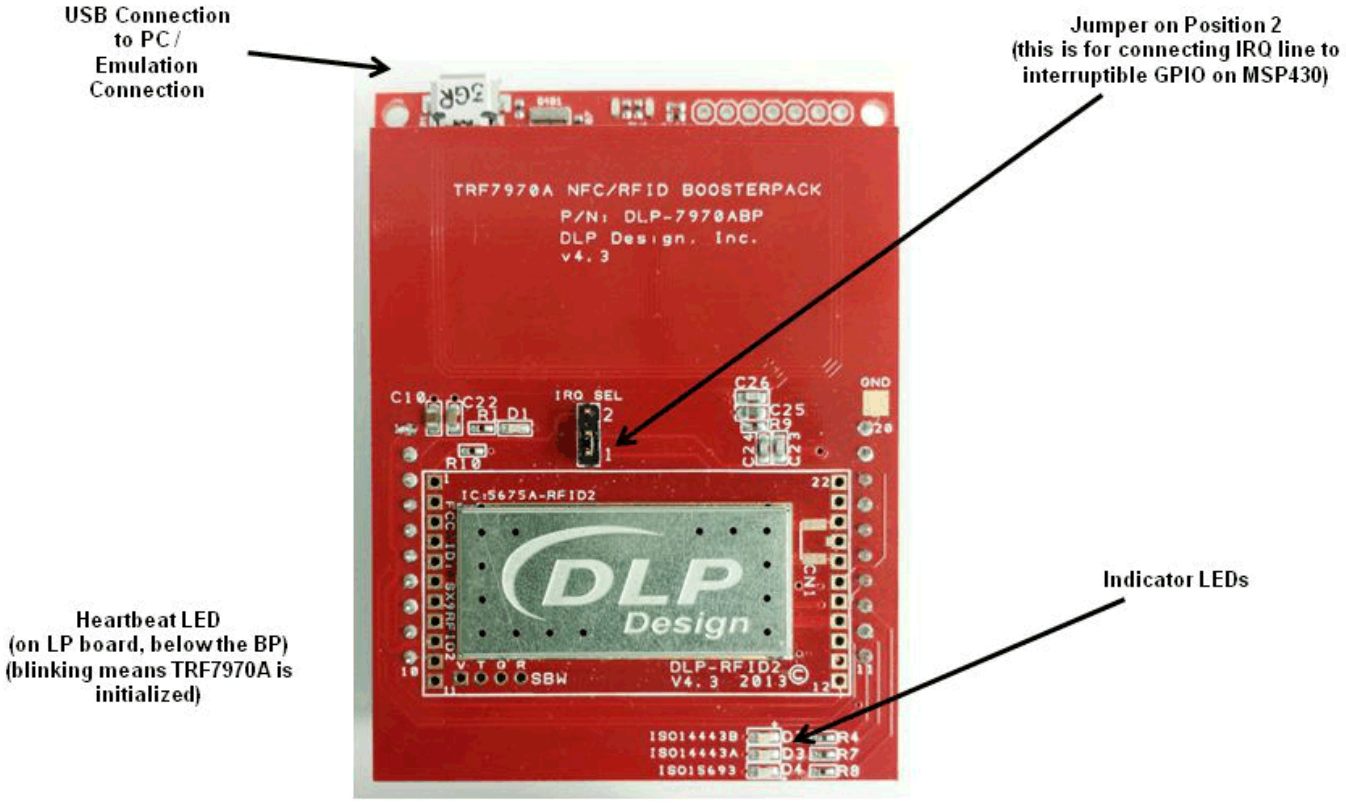

Figure 6. DLP-7970ABP BoosterPack

7 Loading the Code Project

7.1 Download the Flash Programming Tool

Users can download, unzip and install one of two tools to load the binary file into the hardware.

- http://processors.wiki.ti.com/index.php/Category:CCS_UniFlash
  Or
- http://www.elprotronic.com/files/FET-Pro430-Lite-Setup.zip

7.2 Download and Install the NFCLink Package

First, download the NFCLink Installer Package. Then, follow the installer instructions.

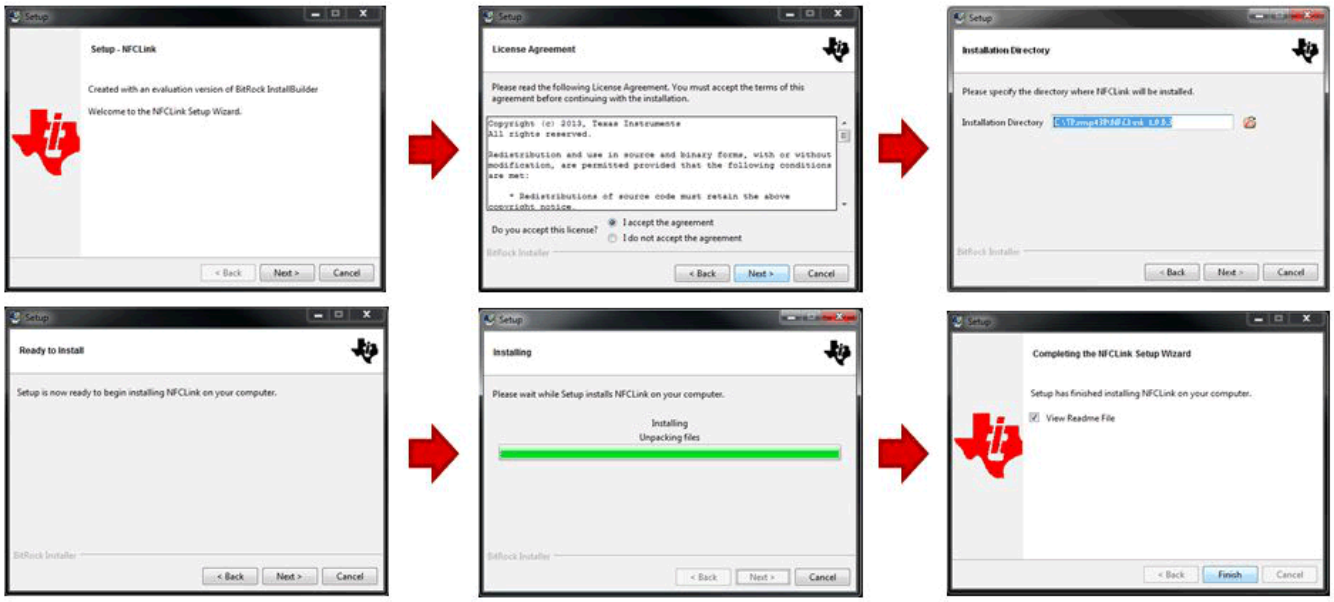

Figure 7. NFCLink Installation Process

After NFCLink package has finished installation, NFCPlayer GUI installation will begin in a new window, as seen in Figure 8.

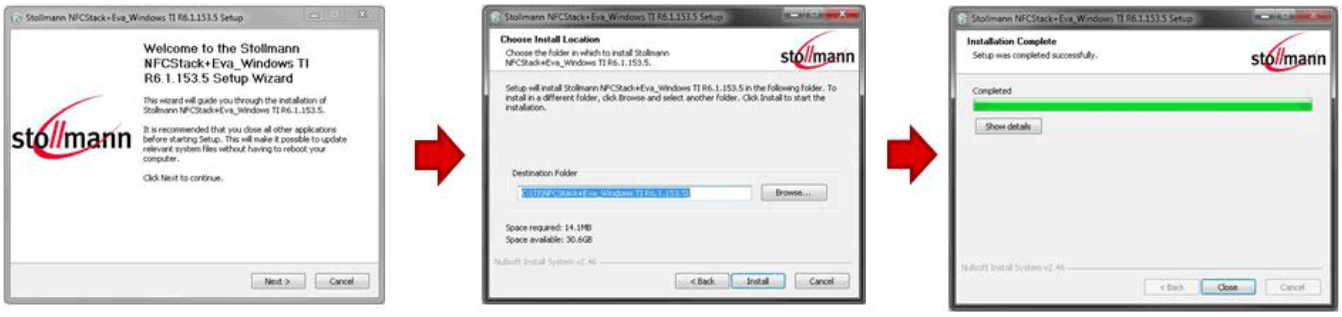

Figure 8. NFCPlayer GUI Installation Process

7.3 Download the Code Project Onto the MSP-EXP430F5529

After retrieving and installing one of the Flash Programming tools, connect the USB-A end of provided USB cable to PC and mini-USB end of cable to the ezUSB connector on MSP-EXP430F5529 board. (see Figure 9 below).

Switch the POWER SEL switch on MSP-EXP430F559 board to the eZ position (middle position, see Figure 9)

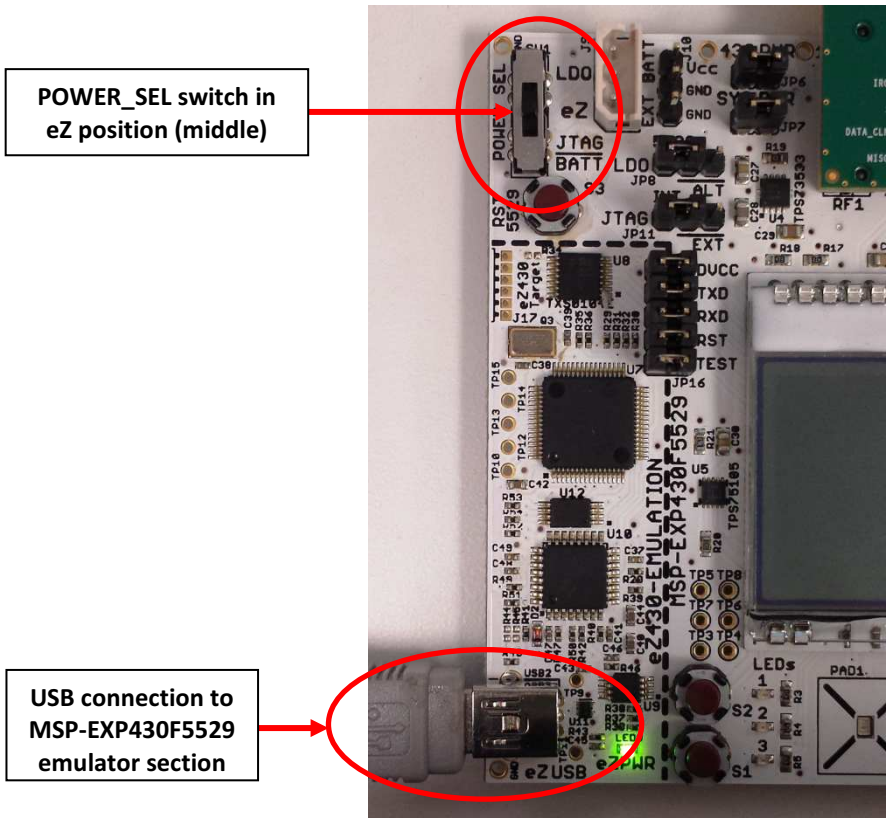

Figure 9. Connection and Configuration for Programming MSP430 Flash

Open Flash Programming tool of choice (both tools will be covered here)

7.3.1 Using Uniflash

1. Open Uniflash tool.
2. Click File, and then choose New Target Configuration.

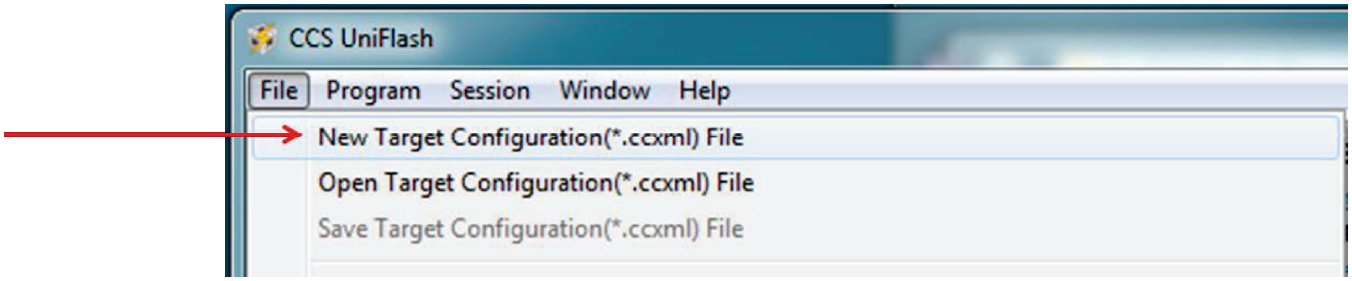

Figure 10.

3. Select Connection (USB1) and select MSP430F5529 Device.

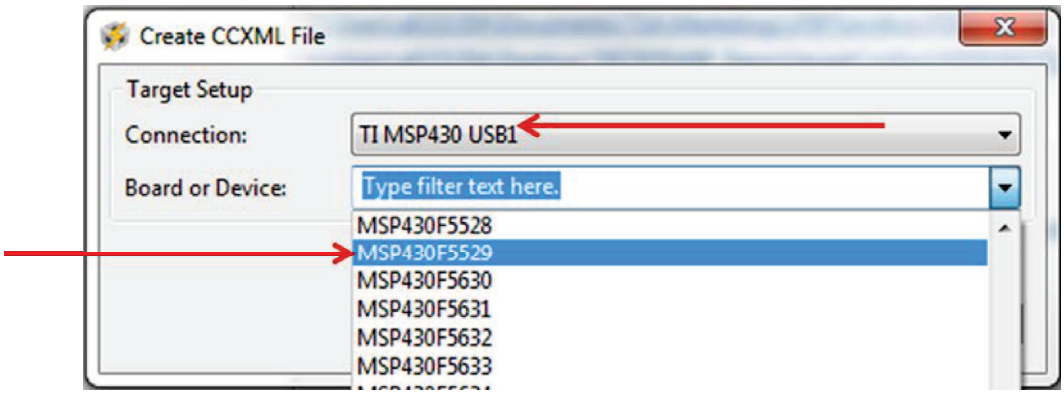

Figure 11.

4. Load the Target Binary by clicking Program and choosing Load Program.

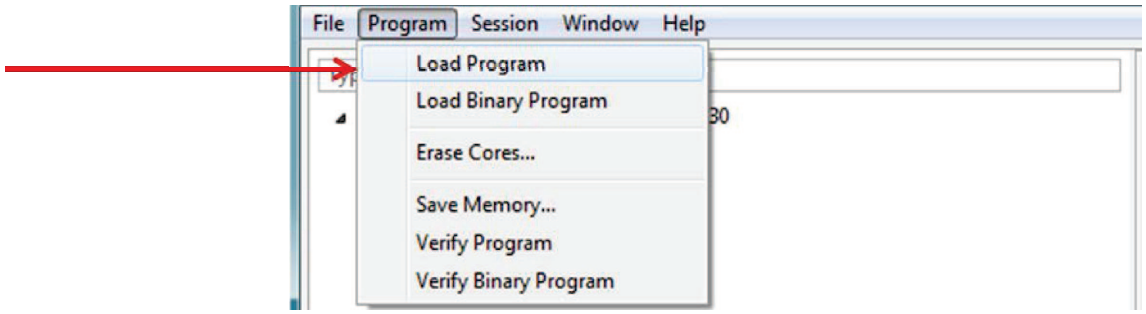

Figure 12.

5. Select path.

   (a) For the MSP430F5529 LaunchPad USB demo, select the path:

   ```
   C:\TI\msp430\NFCLink_1.0.0.3\examples\boards\msp430f5529_lp\allModes\RW_P2P_CE_1\ccs\Debug\RW_P2P_CE_1.hex
   ```

   (b) For the MSP430F5529 Experimenter's Board USB demo, select the path:

   ```
   C:\TI\msp430\NFCLink_1.0.0.3\examples\boards\msp430f5529_exp\allModes\RW_P2P_CE_1\ccs\Debug\RW_P2P_CE_1.hex
   ```

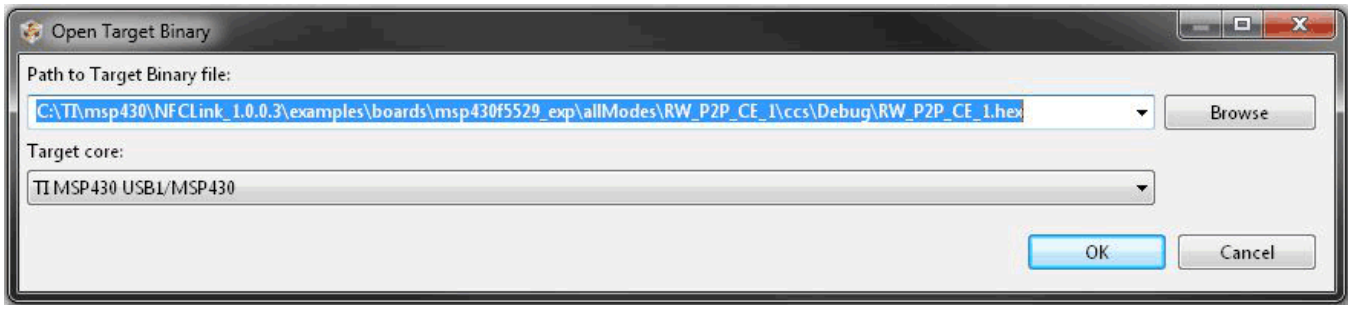

Figure 13.

6. Click OK to start programming the board.

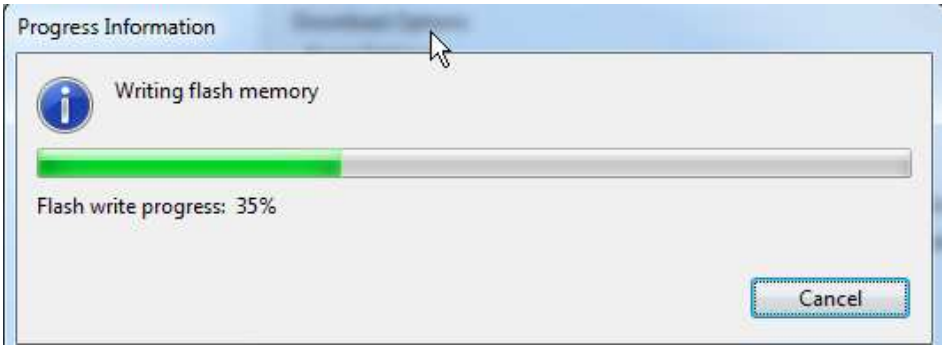

Figure 14.

7.3.2 Using Elprotronic

1. Open the Elprotronic tool.
2. In the top left corner of the GUI, click the Open Code File button, navigate to the folder in which the provided firmware image is stored on the PC, and select it.
   - Path is:

   `C:\TI\msp430\NFCLink_1.0.0.3\examples\boards\msp430f5529_exp\allModes\RW_P2P_CE_1\ccs\Debug\RW_P2P_CE_1.hex`
3. Choose Microcontroller Type Group: (using dropdown) MSP430F5xx
4. Choose (using dropdown) MSP430F5529

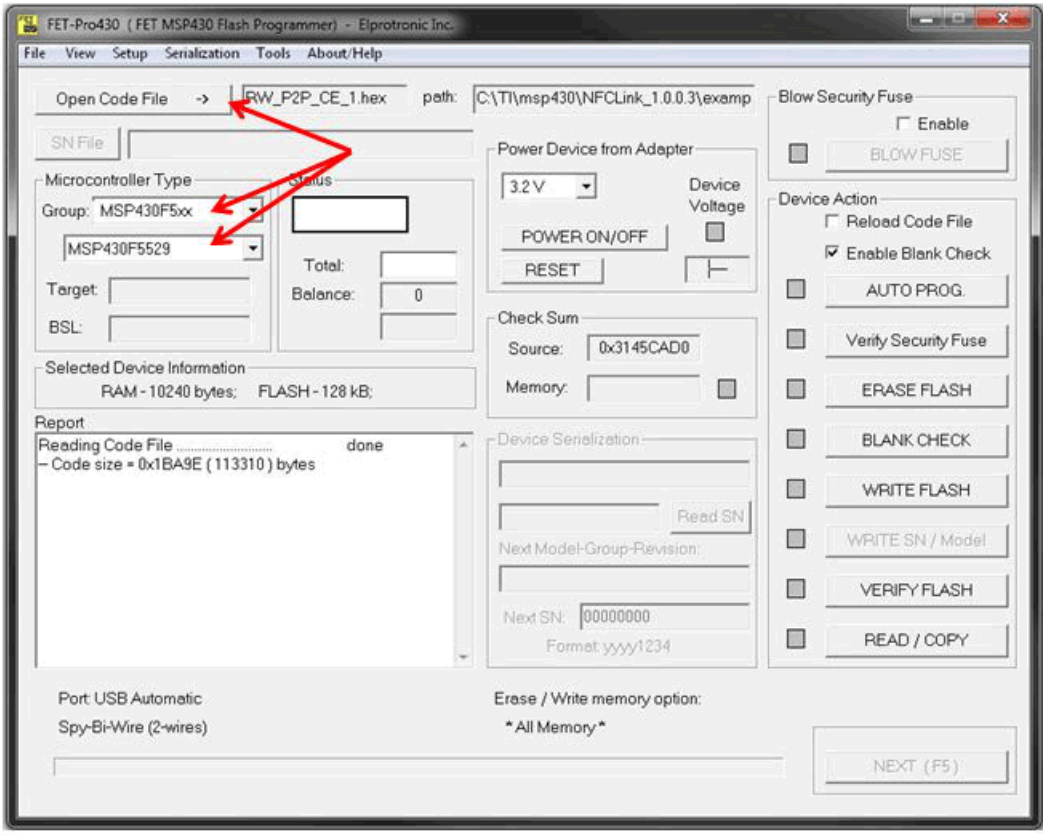

Figure 15.

5. Press the AUTO PROG. button in the Elprotronic tool and allow the tool to complete the steps.

NOTE: If more than one board is to be programmed at this time, the NEXT (or F5) button can be used to program additional units instead repeating all the steps.

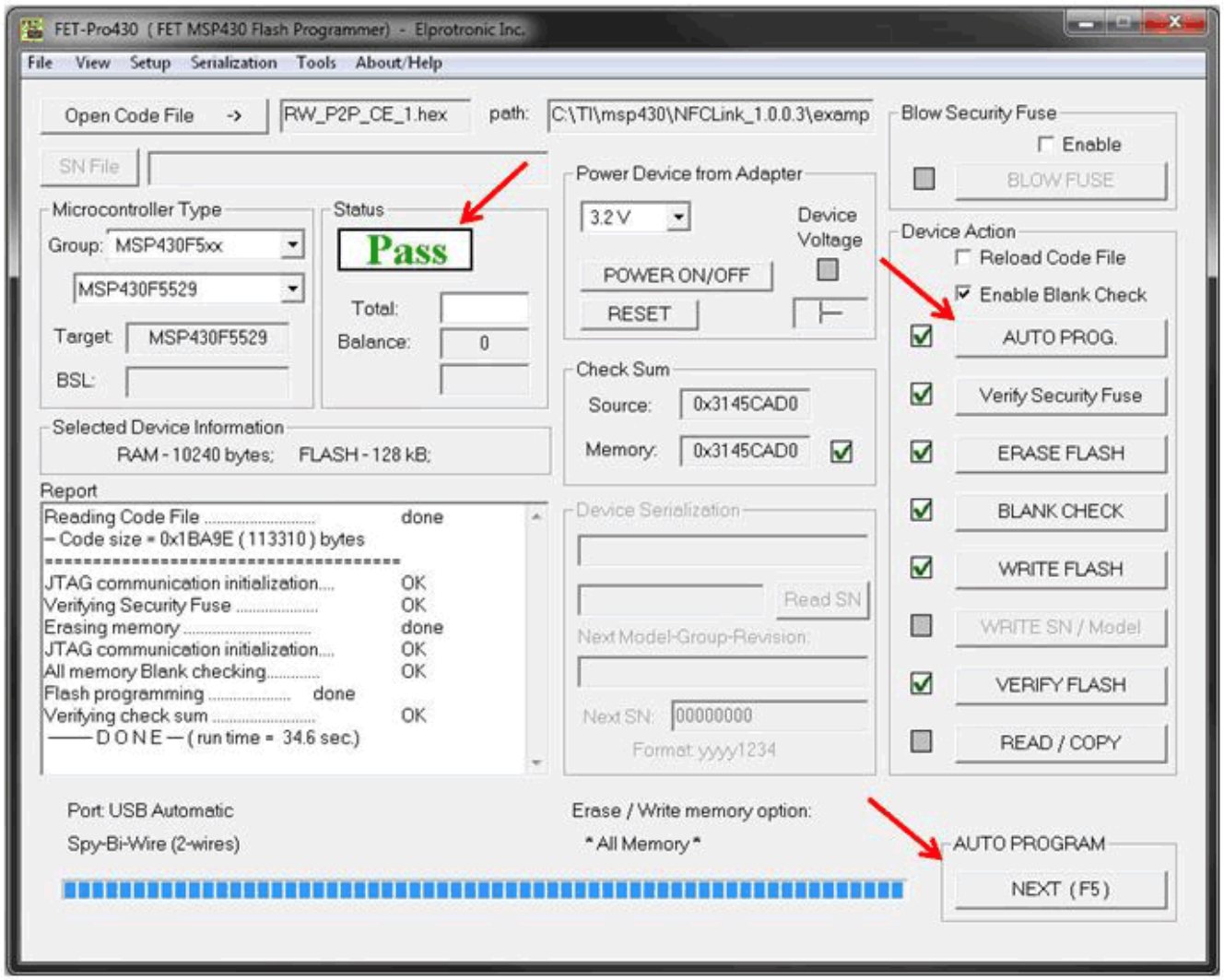

Figure 16.

After flashing the MSP430F5529 (with either tool), disconnect the mini-USB cable connector from the eZ USB board connector, move the POWER SEL switch to the LDO position (topmost), and then connect the mini-USB cable connector to 5529 USB board connector as shown in Figure 17.

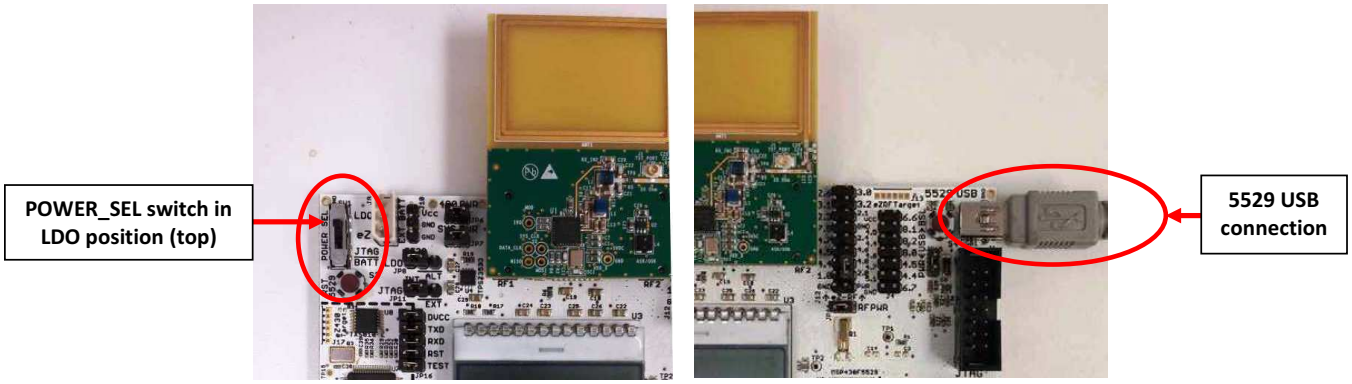

Figure 17. Connecting Mini-USB Cable to 5529 USB Board

8 Installing USB Driver

If the USB driver is not automatically detected and used when the 5529 USB connection is made, then PC can be pointed to USB driver .inf file (C0_SimpleSend.inf) which is located in this directory (after NFCLink is installed):

```
C:\TI\msp430\NFCLink_1.0.0.3\usblib430\Source\USB_config
```

9 COM Port Settings

1. Proceed to the Device Manager from the PC Control Panel to determine which COM Port the board has enumerated; that port information is needed to configure the COM port correctly. This step will only need to be done the first time the board is enumerated on a given PC system. The board will enumerate as a virtual COM port (USB CDC). in the example shown in Figure 18 the virtual COM port is shown as COM61.

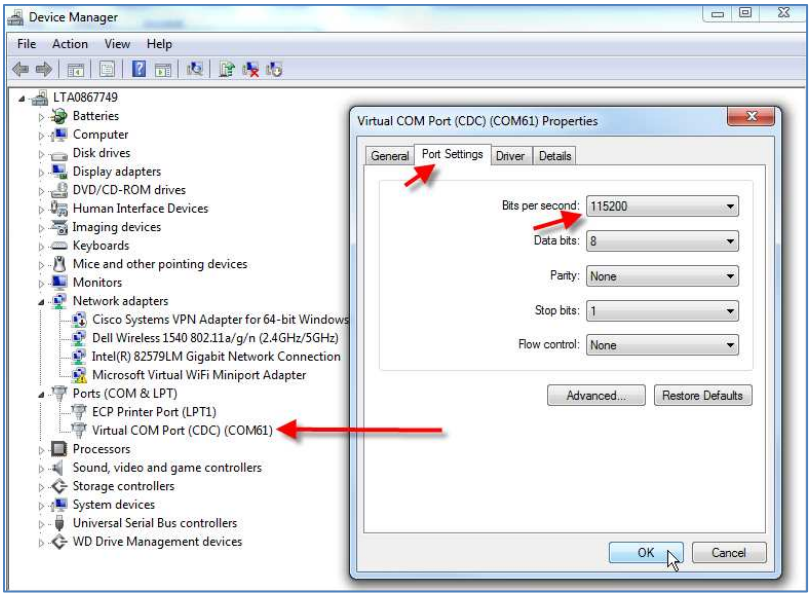

Figure 18.

2. Double click on the port in the Device Manager menu window
3. Navigate to Port Settings tab and ensure the baud rate is set for 115200, 8 data bits, no parity and 1 stop bit (8N1)
4. Click OK.

10 PC GUI

10.1 Installation

Installation of the PC GUI should have occurred in Section 7.2 of this document. If for some reason the installation was declined or stopped or cancelled during that step, the following steps should be followed to complete the installation of the GUI.

1. Navigate to NFCLink directory named 'utils', double click the executable highlighted in Figure 19.

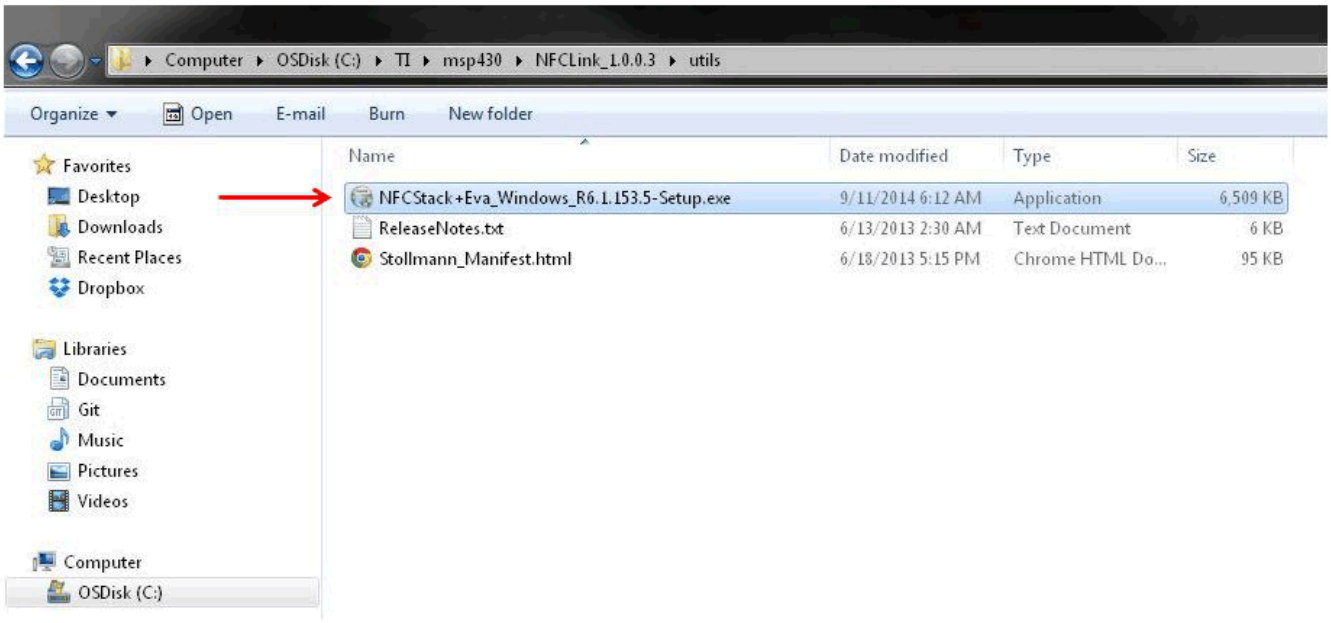

Figure 19.

2. Follow the instructions of the installer, as shown in Figure 20.

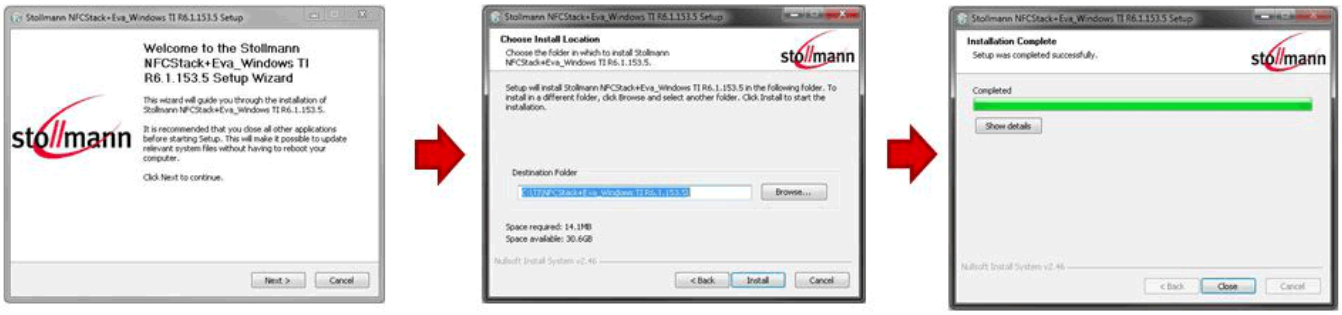

Figure 20.

3. Execute the NFCPlayer by navigating to the "Stollmann NFCStack+Eva_Windows TI R6.1.153.5" folder (should be in the "Program Files" directory), and double click the NFCPlayer Icon as shown in Figure 21.

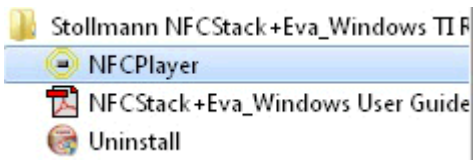

Figure 21.

10.2 GUI Configuration

1. When the GUI opens, navigate to the Configure button.
2. Provide a Description in the Description field
3. Type in the enumerated COM port values.
4. Click save.

NOTE: The configuration steps only need to be done once for a given set of hardware on a PC if step 3 (saving the configuration) is done. Henceforth, if multiple platforms have been set up and saved on the same PC, the description field will be a dropdown menu with all previous selections available to choose from; these steps are illustrated in Figure 22.

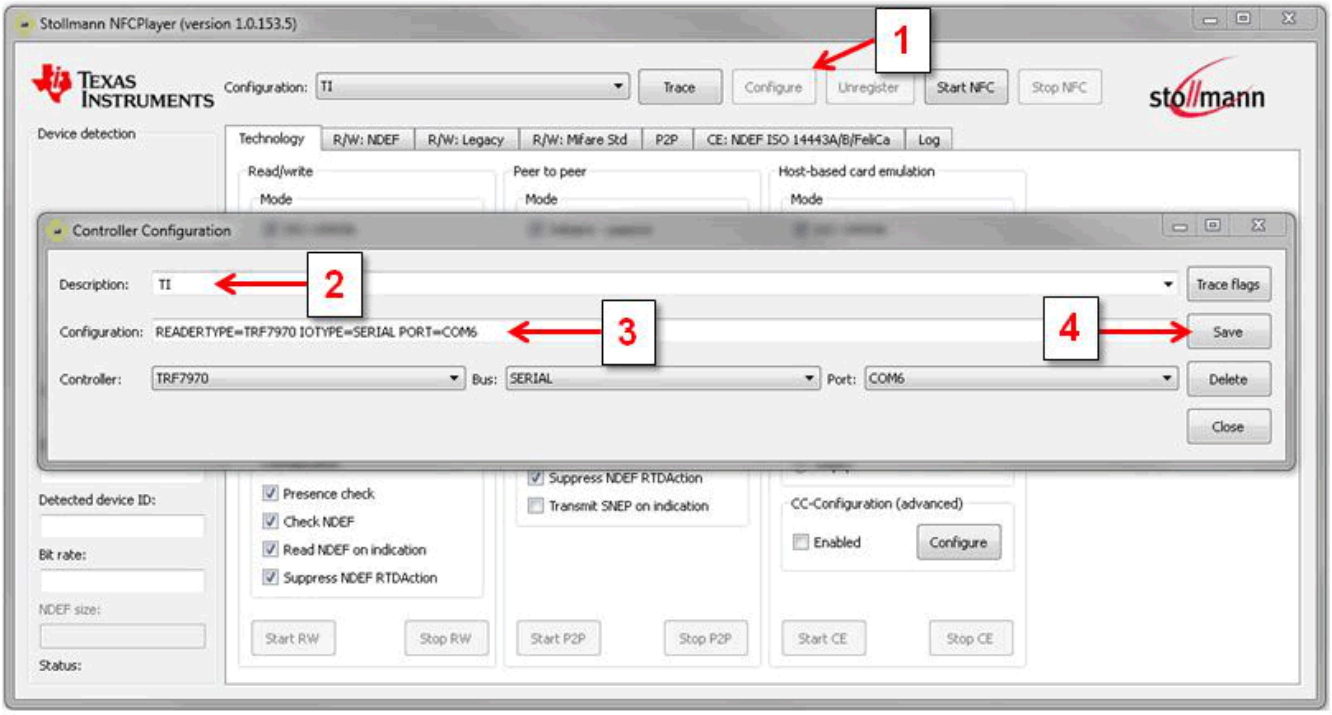

Figure 22. Steps for Configuring the GUI

The window in Figure 23 is now shown.

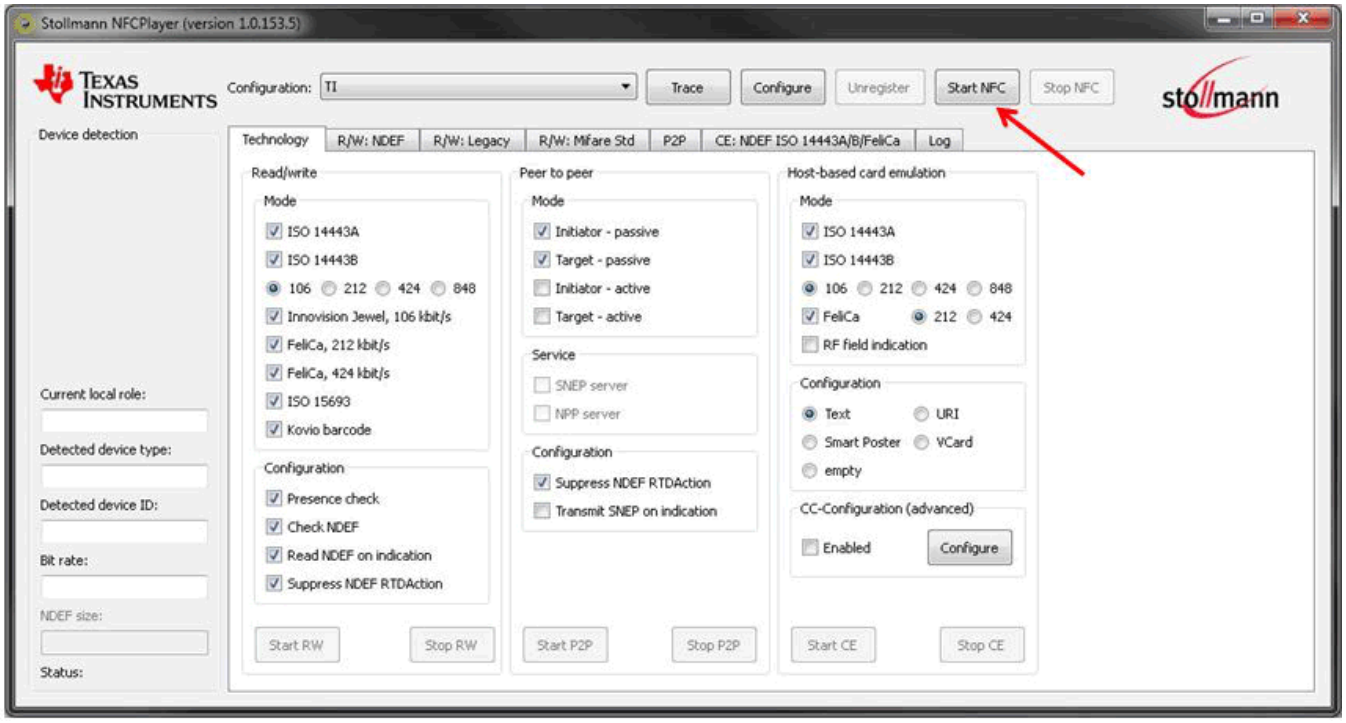

Figure 23.

When NFCPlayer is started, the status (NFC: Start OK) will be indicated as shown in Figure 24.

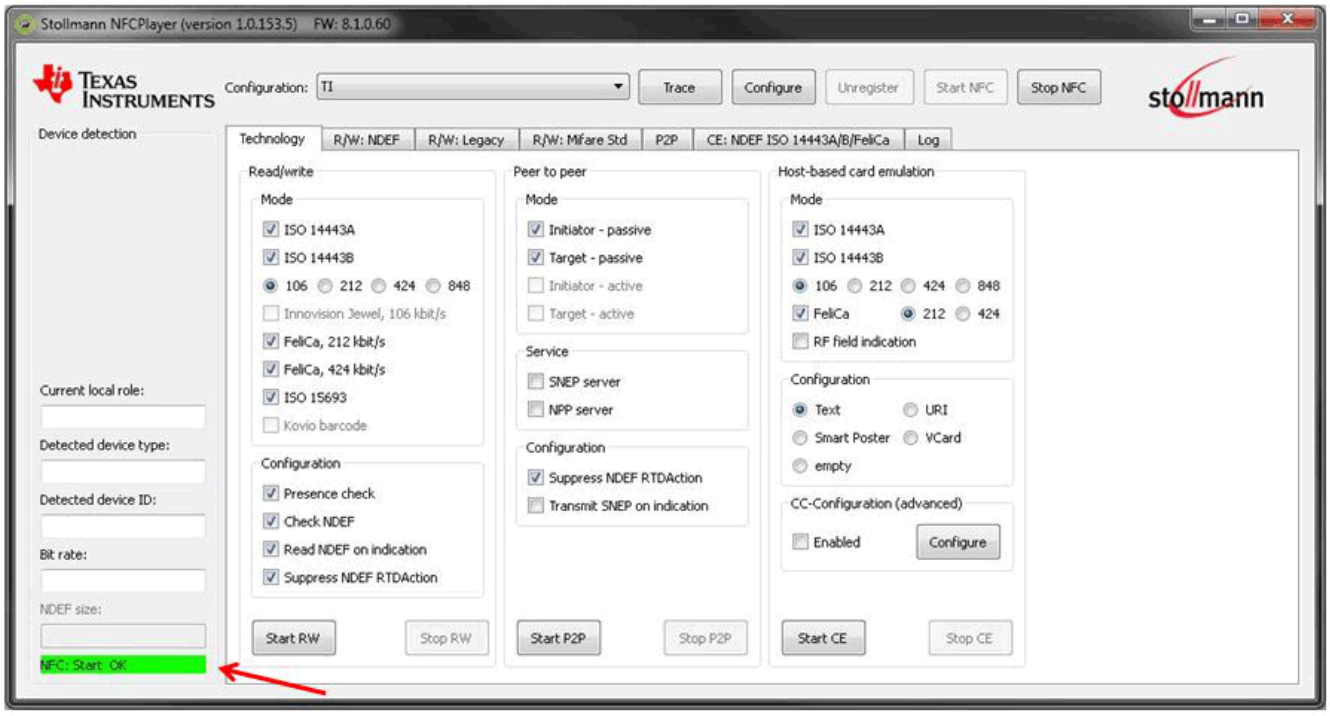

Figure 24.

11 NFC or RFID Reader and Writer Mode Overview

To use the Reader/Writer Mode, press the Start RW button. RW: Start OK will appear in the NFCPlayer status window and the Current local role window will show RW, as shown in Figure 25.

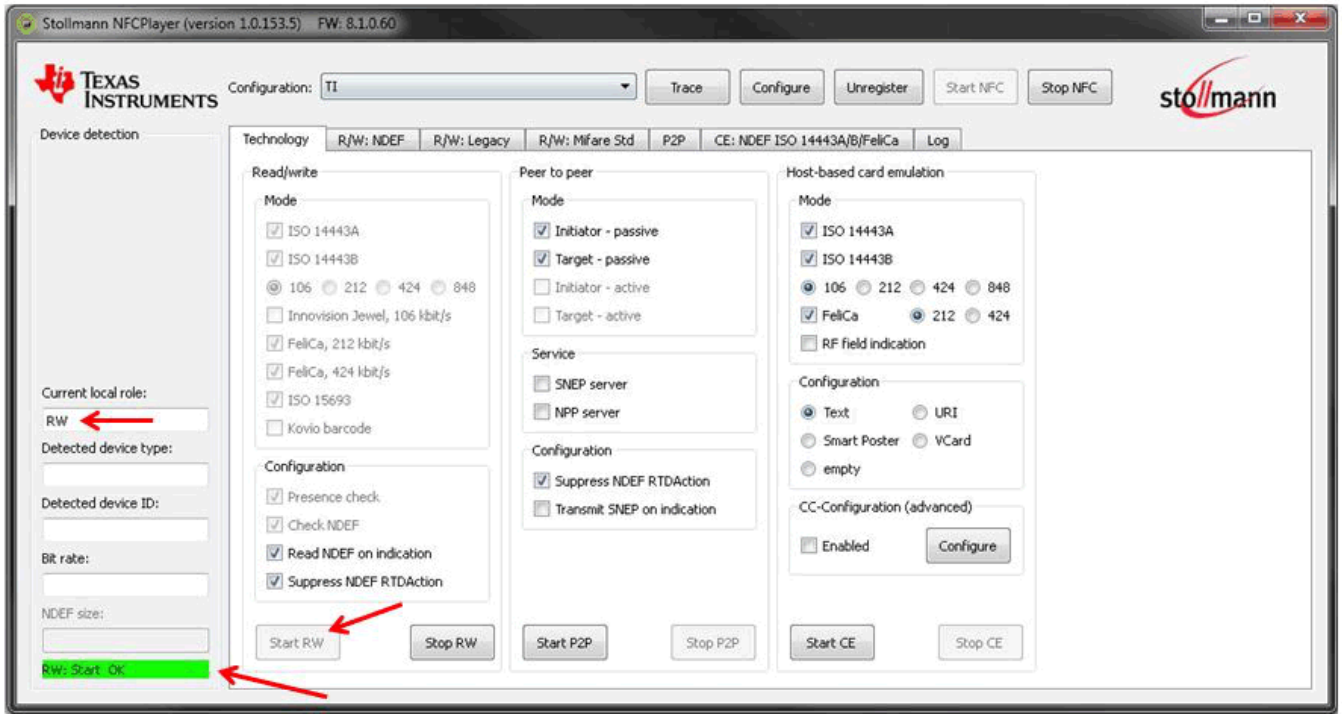

Figure 25.

When hardware is in reader/writer mode, LED1 will be flashing and then go solid, as shown in Figure 26) when an NFC/RFID transponder is presented.

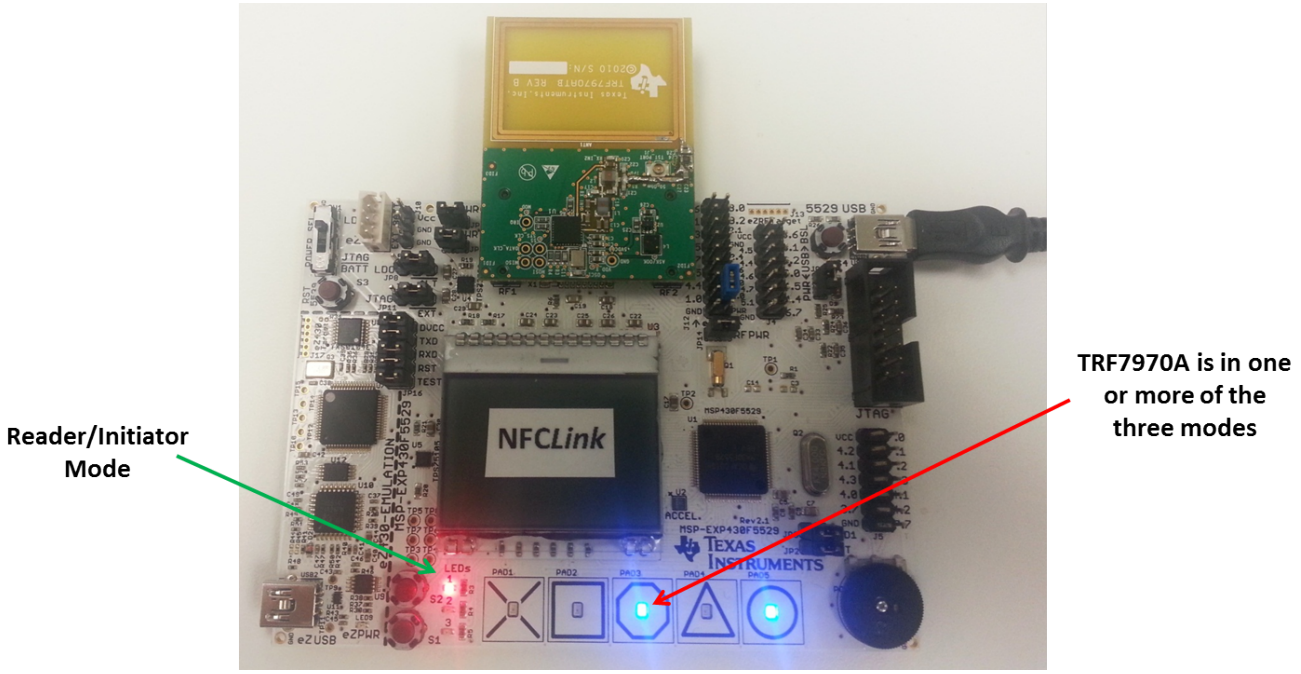

Figure 26.

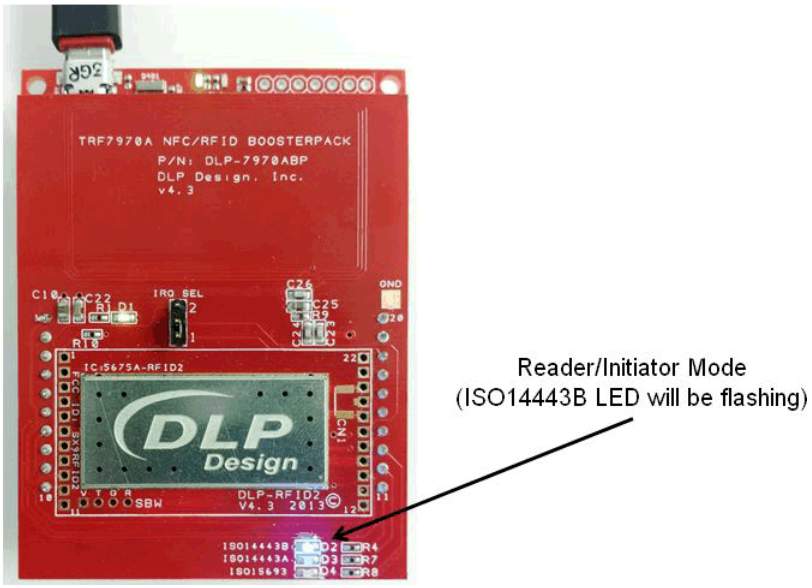

Figure 27.

12 NFC or RFID Peer-to-Peer (P2P) Mode Overview

To use the Peer-to-Peer Mode, press the Start P2P button. P2P: Start OK will appear in the NFCPlayer status window and Current local role window will show P2P, as shown in Figure 28.

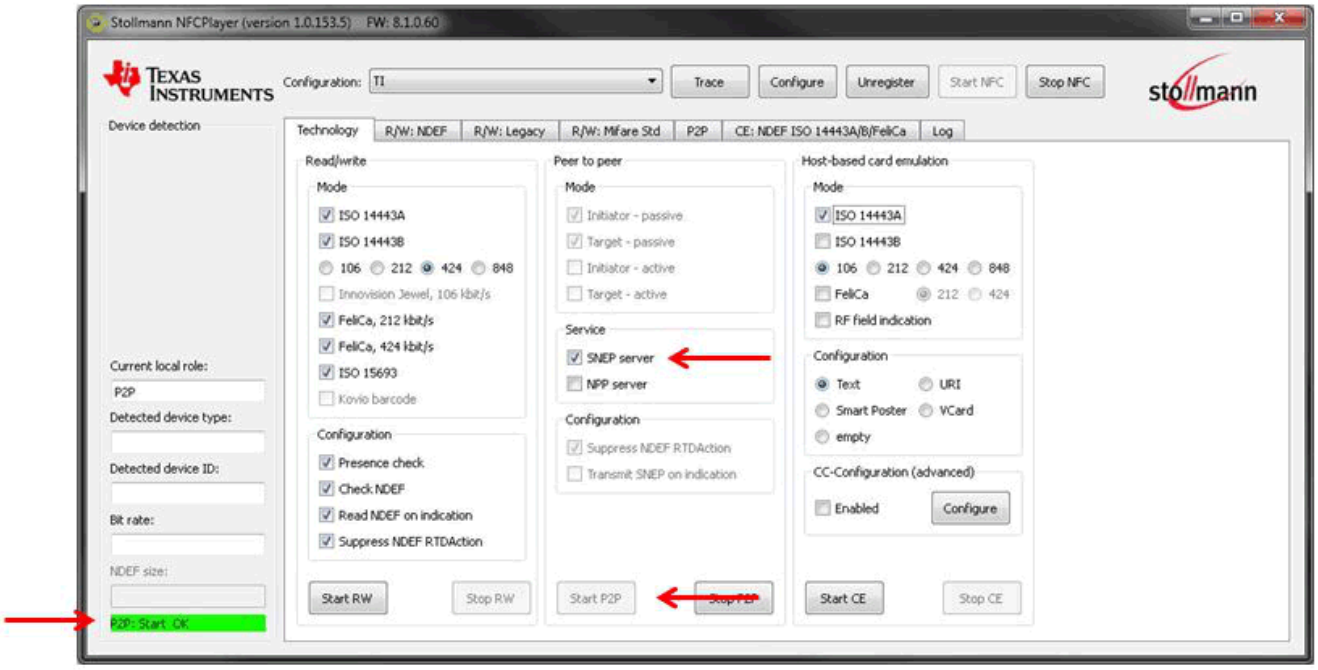

Figure 28.

When the hardware is in P2P Mode, LED1 and LED2 flash alternately. When Peer NFC Device is presented, the mode being used by the hardware (Initiator or Target) is on solid. After transfer of data is complete and the peer is removed, the LEDs again flash alternately (see Figure 29).

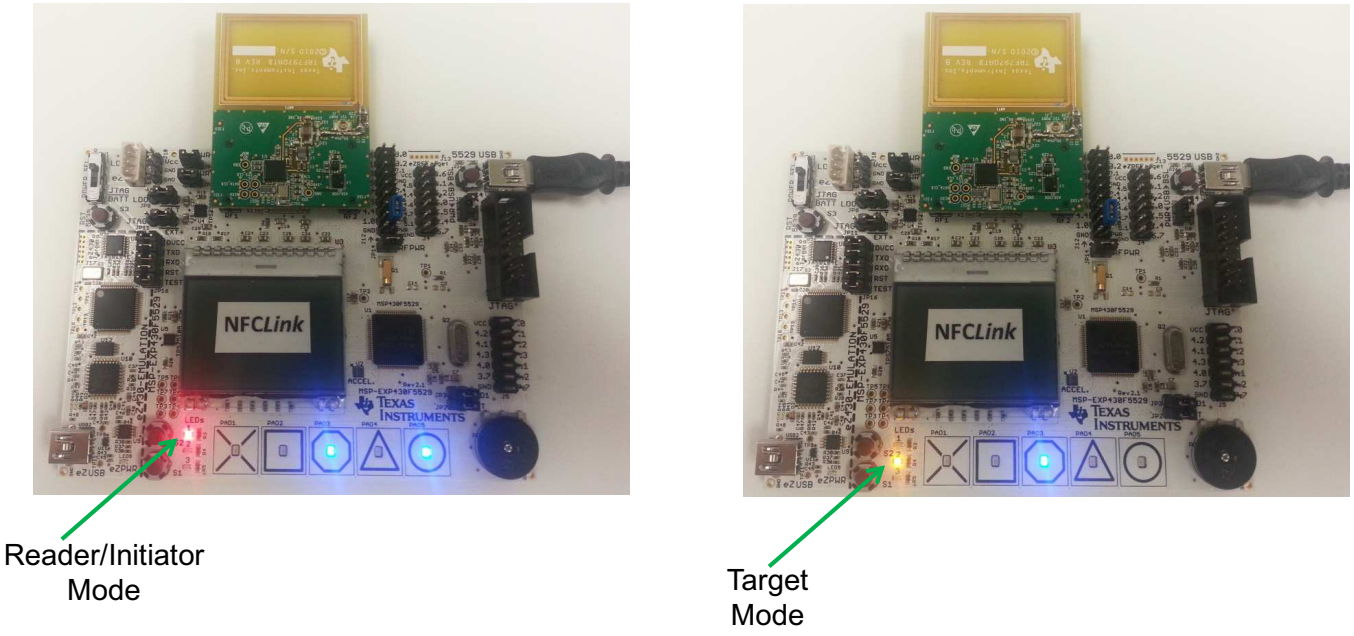

Figure 29.

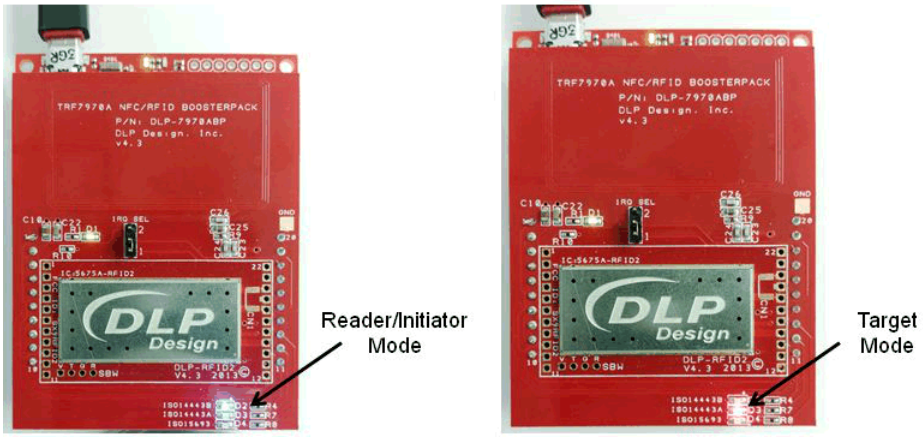

Figure 30.

13 NFC or RFID Card Emulation (CE) Mode Overview

To use the Card Emulation Mode, press the Start CE button. "CE: Start OK" is shown in the NFCPlayer status window, and the Current local role window shows "CE", as shown in Figure 31.

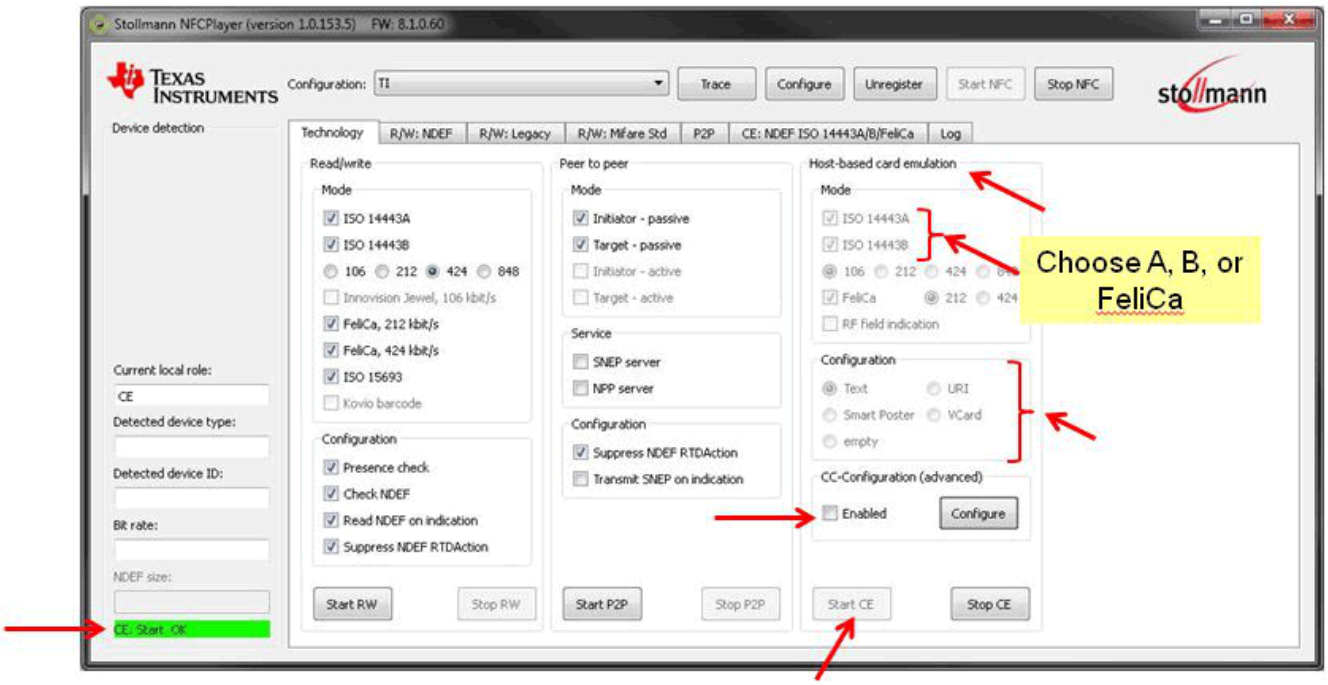

Figure 31.

When the hardware is in CE mode, LED3 is on solid.

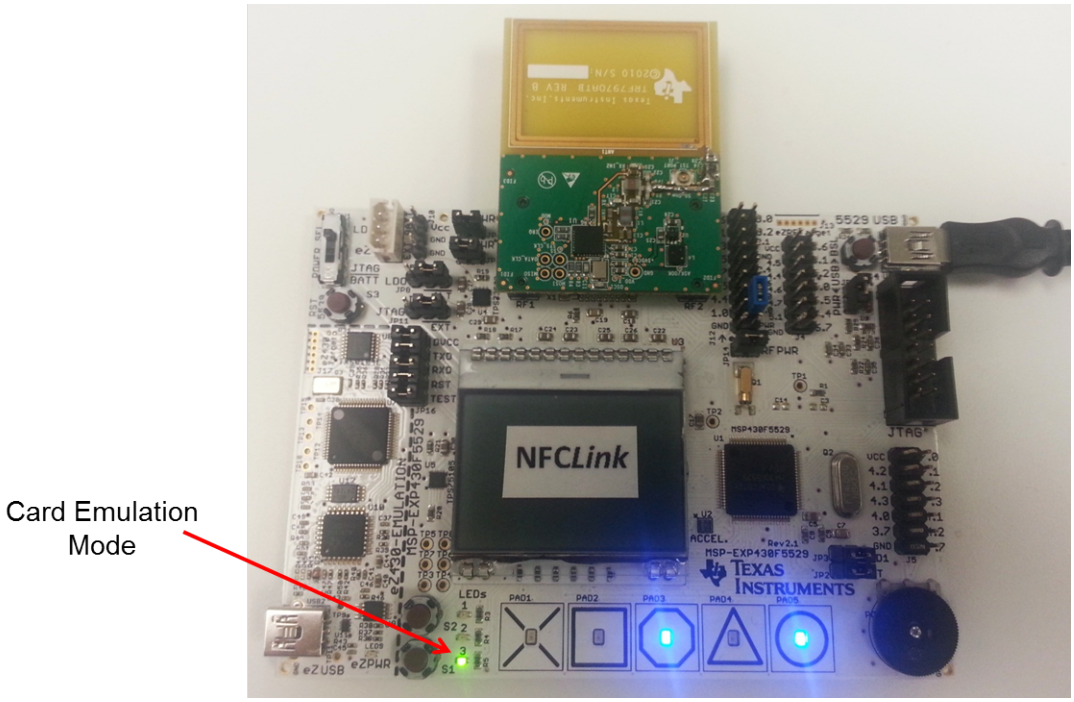

Figure 32.

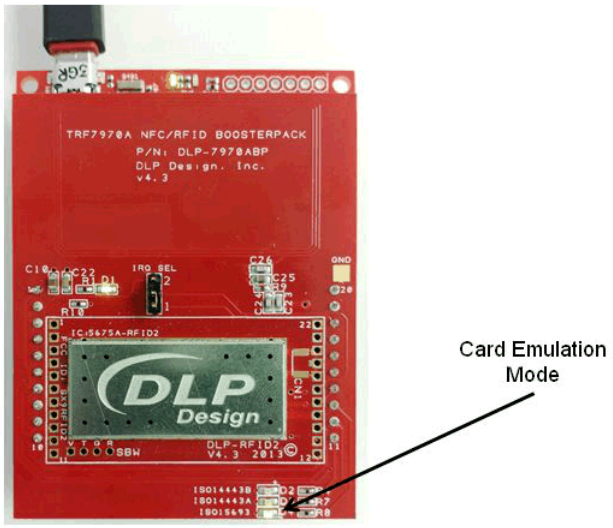

Figure 33.

14 NFC or RFID Reader and Writer (RW) Mode Details

Reader and Writer modes demonstrated and explained by showing the Tag Types and the different types of common data stored.

RTD types:

- Text
- URI
- Smart Poster
- Vcard
- MIME

Tag Platforms Types supported are:

- NFC-5 (TI ISO15693)
- NFC-2 (MF UL)
- NFC-3 (Sony FeliCa)
- NFC 4A (DESFire EV1)
- NFC 4B (TI Dynamic Tag, the RF430CL330H)

14.1 NFC Type 2 Tag Platform

Figure 34 shows an NFC Type 2 Tag Platform which has been formatted with the tool and then programmed as RTD Smart Poster. If Suppress NDEF RTD Action (in front panel) box is unchecked, presenting this tag will open browser up and will be routed to http://www.ti.com/tool/nfclink web page.

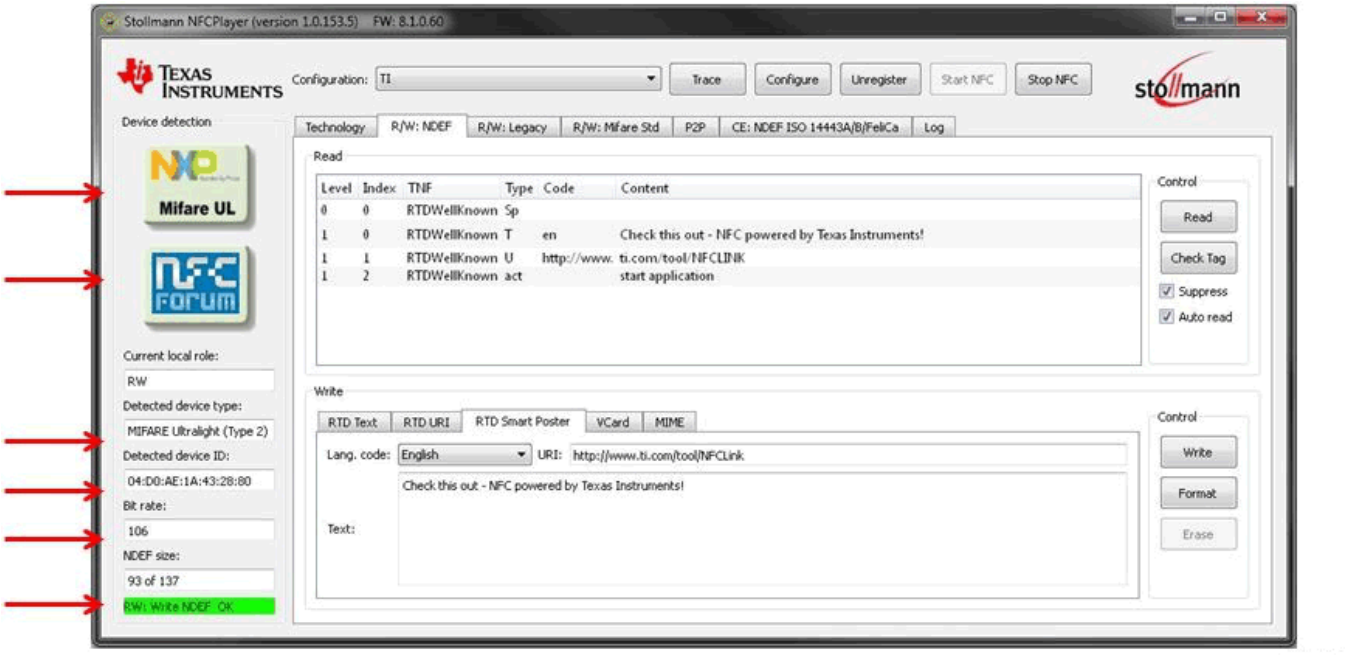

Figure 34.

14.2 NFC Type 3 Tag Platform

Figure 35 shows an NFC Type 3 Tag Platform which has been formatted with the tool and then programmed with an RTD URI.

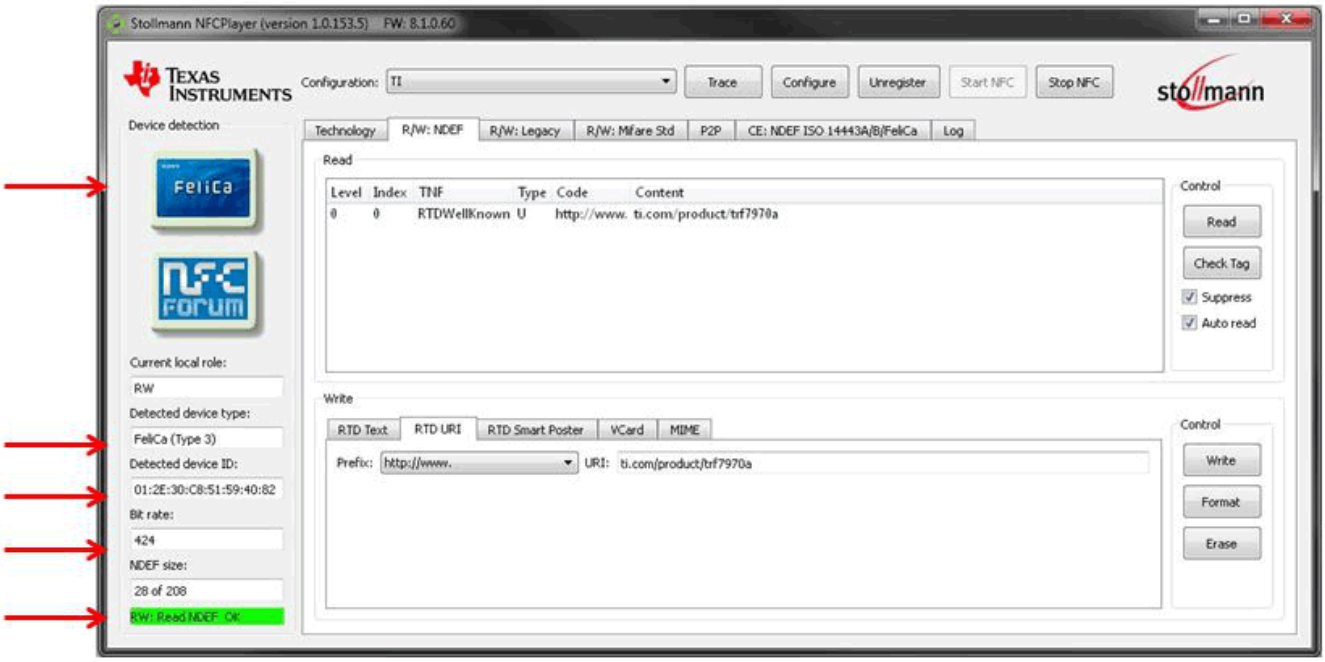

Figure 35.

14.3 NFC Type 4A Tag Platform

Figure 36 shows an NFC Type 4A Tag Platform which has been formatted with the tool and then programmed as a VCard.

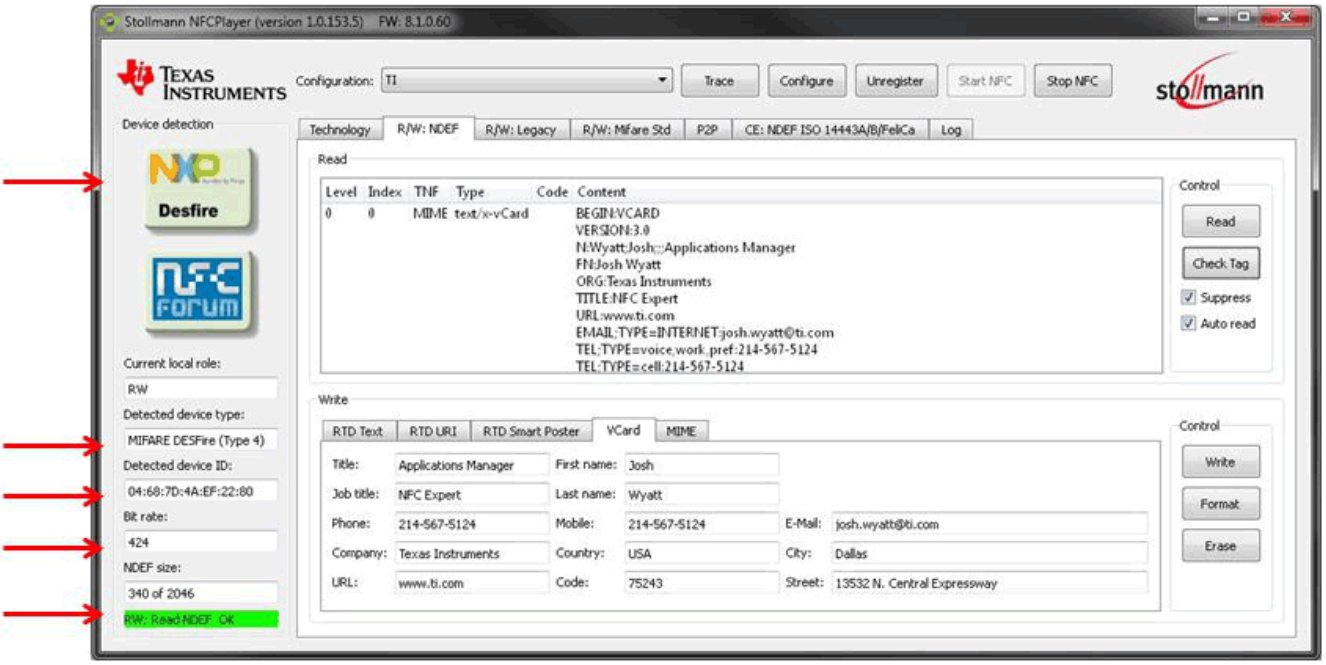

Figure 36.

14.4 NFC Type 4B Tag Platform

Figure 37 shows a Tag Type 4B that has been formatted with the tool and then programmed for NFC Forum *Bluetooth*® Connection Handover.

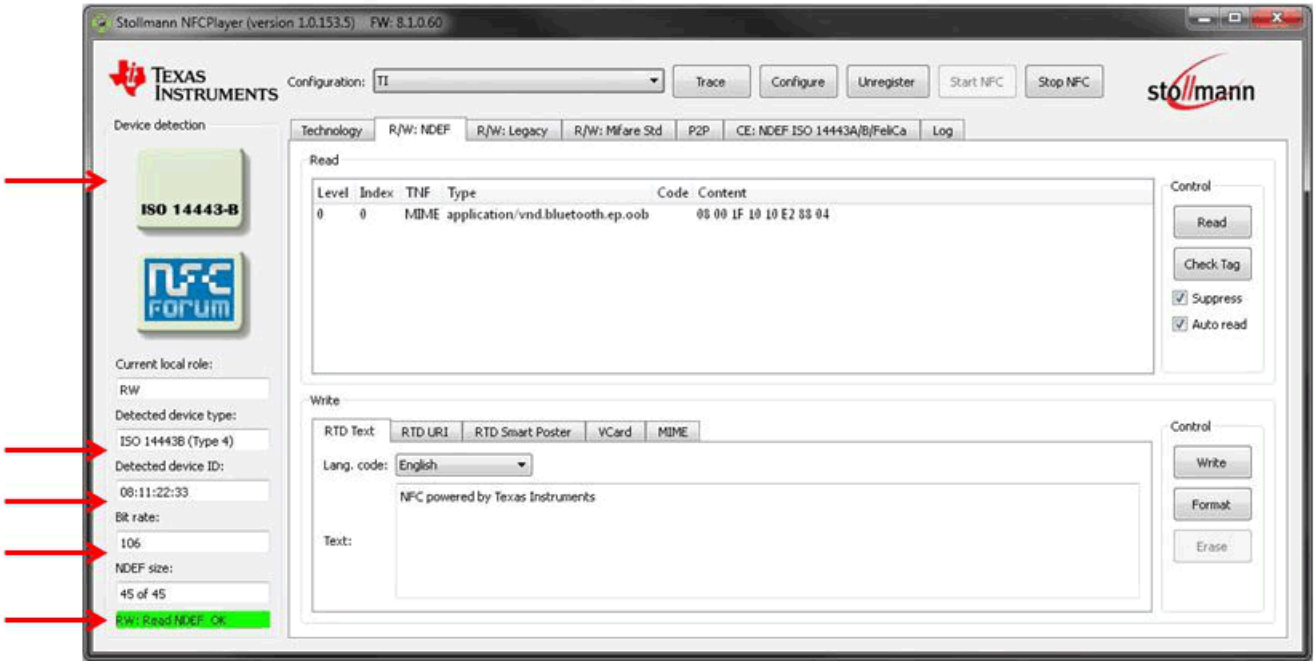

Figure 37.

Figure 38 shows the same Tag Type 4B (RF430CL330H) when it has been formatted with the tool and then programmed with an image (MIME). This demonstrates a larger file being stored, which could then be extracted out of the tag by another NFC device or by the host over I2C or SPI (wired connection).

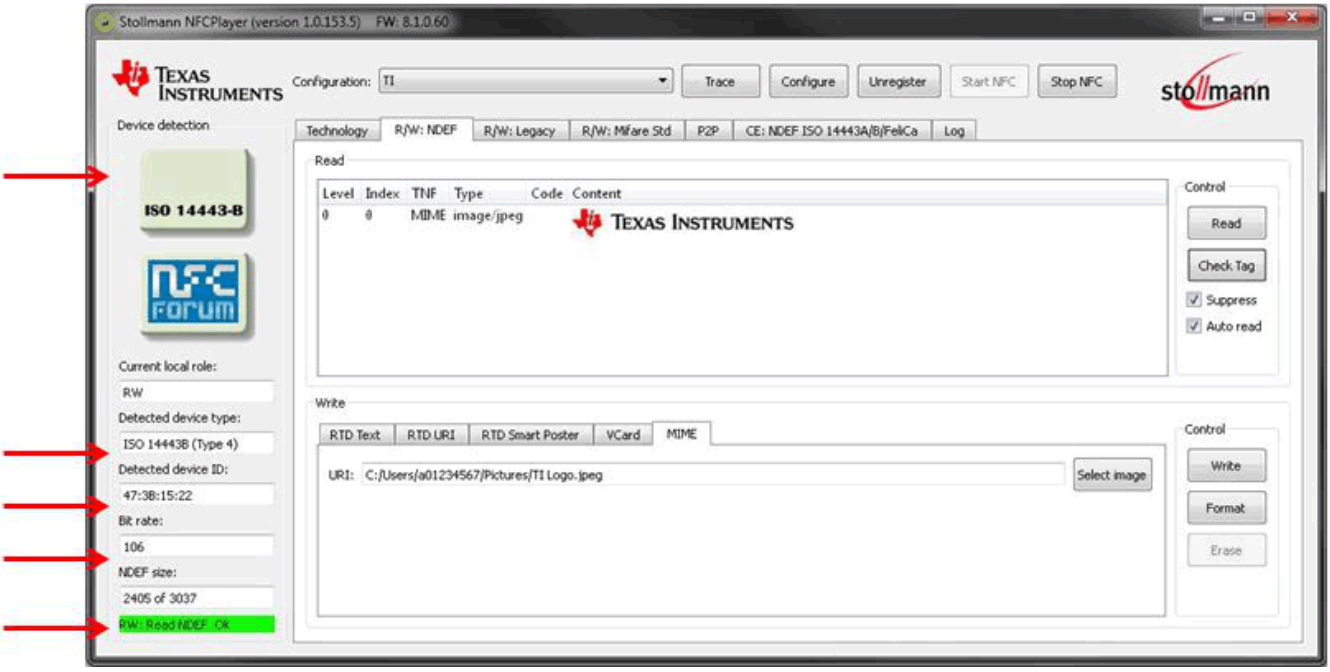

Figure 38.

14.5 NFC Type 5 Tag Platform

After pressing the R/W button, present a NFC tag (doesn't have to be formatted). in this example we are presenting an unformatted NFC Type 5 (ISO15693, TI HF-I) tag. Note that the tab automatically flips to R/W: Legacy and the type of card is displayed graphically along with the Unique ID (in this case, E00700002A044954). Press the Format NDEF button to make the tag NFC-V type. This puts a Capability Container with empty NDEF in the user memory.

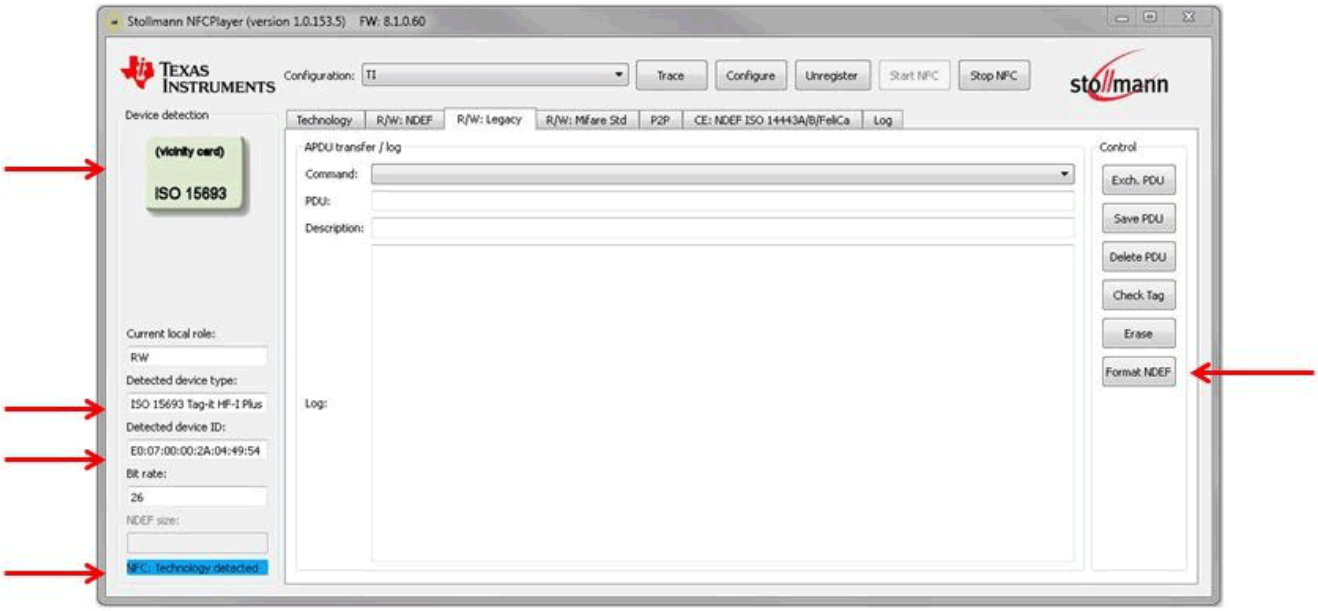

Figure 39.

After the tag is formatted, the GUI status window will quickly flip to Format NDEF OK, then present the R/W NDEF with the status changing to RW: Read NDEF OK. Figure 40 shows the results (formatted, but with an empty tag)

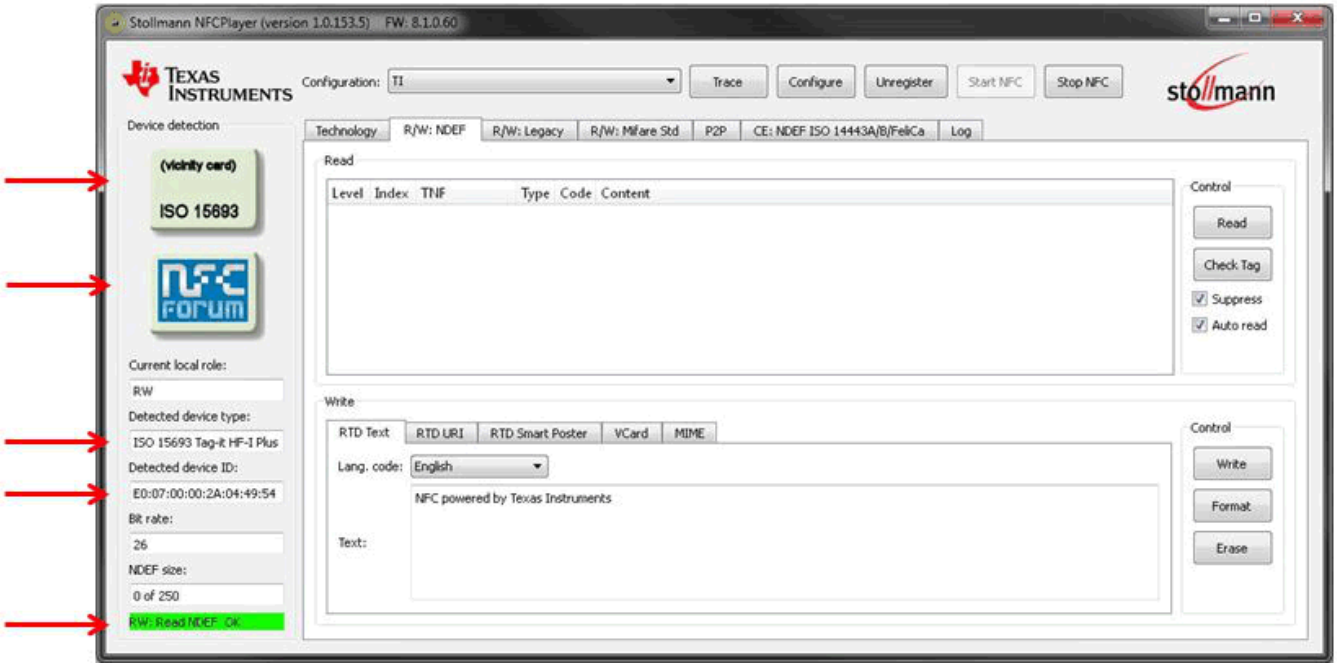

Figure 40.

User can now enter data into the bottom 'Write' window and press the Write button to write a NDEF message into the tag. When this occurs, the screen will quickly flip to Write NDEF. You can then either press the Read button, or remove the tag from the reader/writer antenna and re-present the tag to get the NDEF message read back.

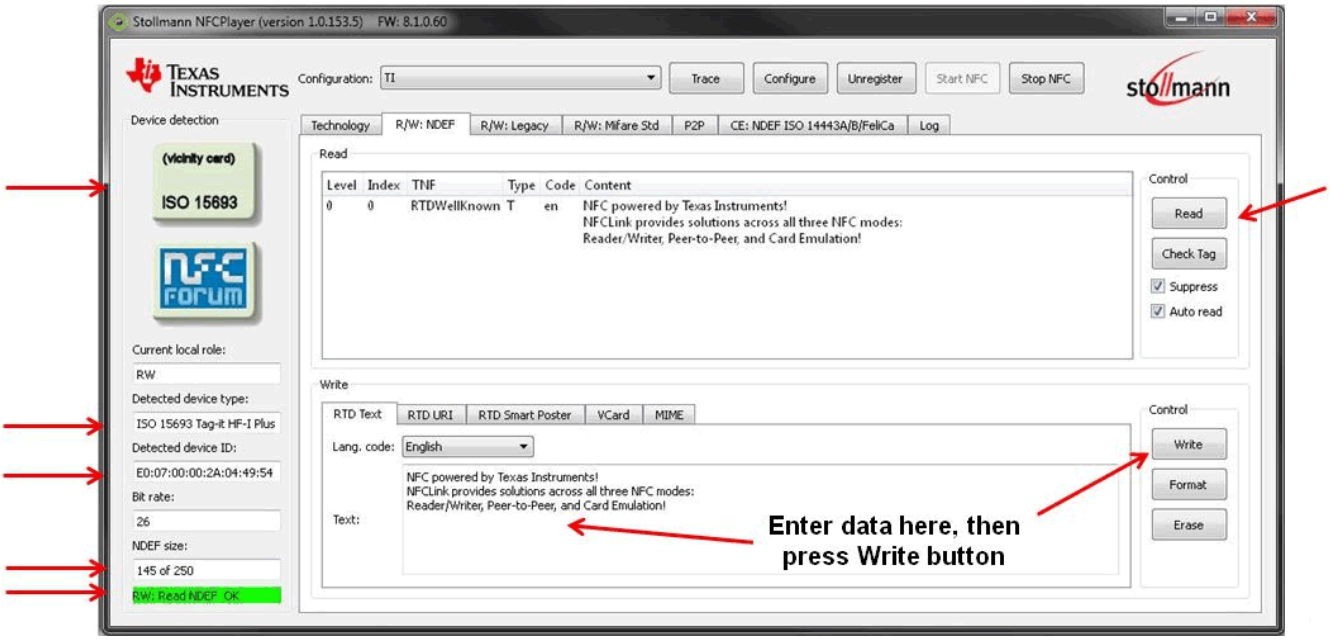

Figure 41.

To stop the R/W mode, remove the tag from the field and press the Stop RW button.

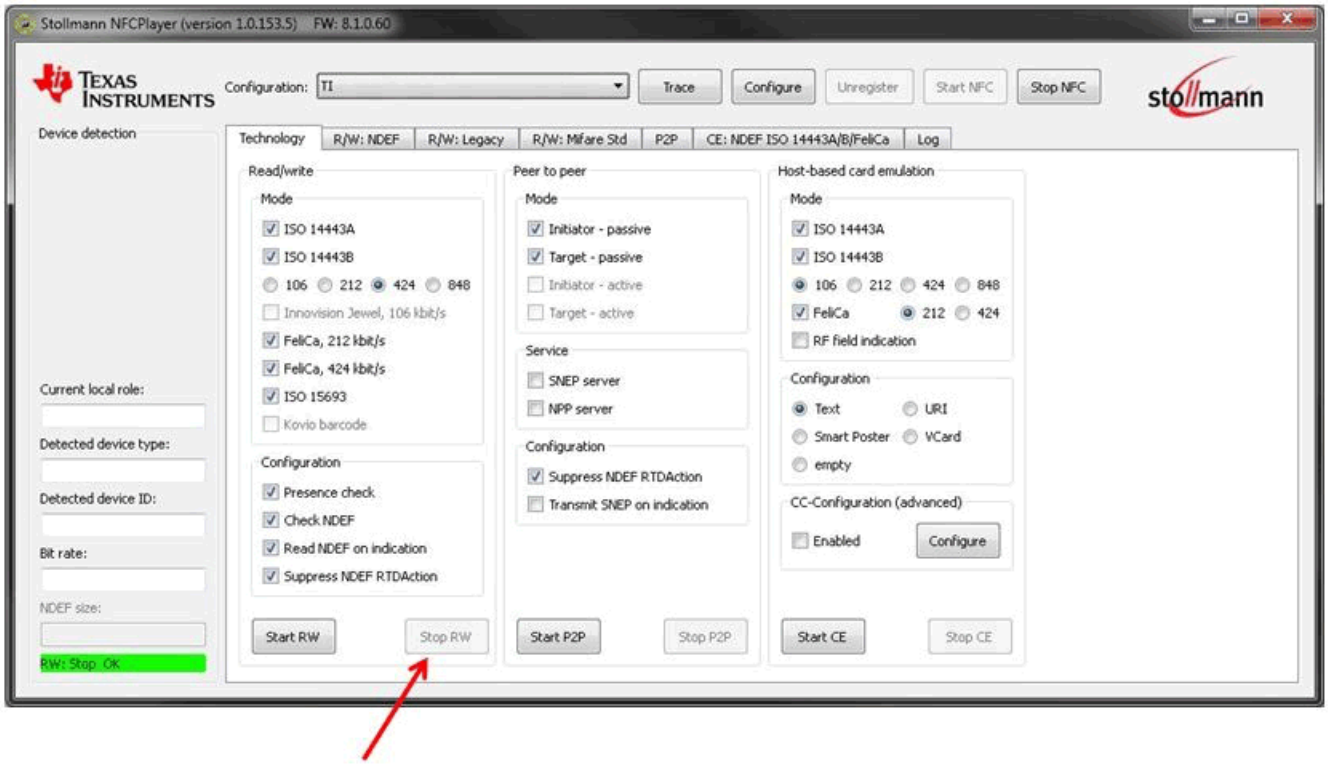

Figure 42.

15 NFC or RFID Peer-to-Peer (P2P) Mode Details

The TRF7970A is capable of being a Type-A or Type-F peer (target or initiator) at 106 kbps, 212 kbps, or 424 kbps. This section describes how to accomplish this with the use of the host stack and the embedded MCU library. This NFC mode can be used for a variety of applications in which a data transport system is needed for data exchange, firmware updates, or simplified setup of alternative radios (for example, *Bluetooth*, Wi-Fi®, or RF4CE). This differs from card emulation, because after the connection is established, data can be passed either way. Also, multiple channels can be set up to provide for high-speed full-duplex data transmission schemes over the magnetic link. Early versions of the NFC devices used the NDEF Push Protocol (called NPP), but now the NFC Forum has a newer specification called Simple NDEF (NFC Data Exchange Format) Protocol or "SNEP". Both are supported with the embedded firmware and the host stack offered by Texas Instruments.

When NFC Enabled device is presented (no app open in this case), the GUI screen flips over to P2P mode. Message can now be sent from a GUI to the NFC-enabled device using the Transmit button. This opens either the native app on the phone or a default one (the NFC TagReader from KDDI is shown in Figure 43).

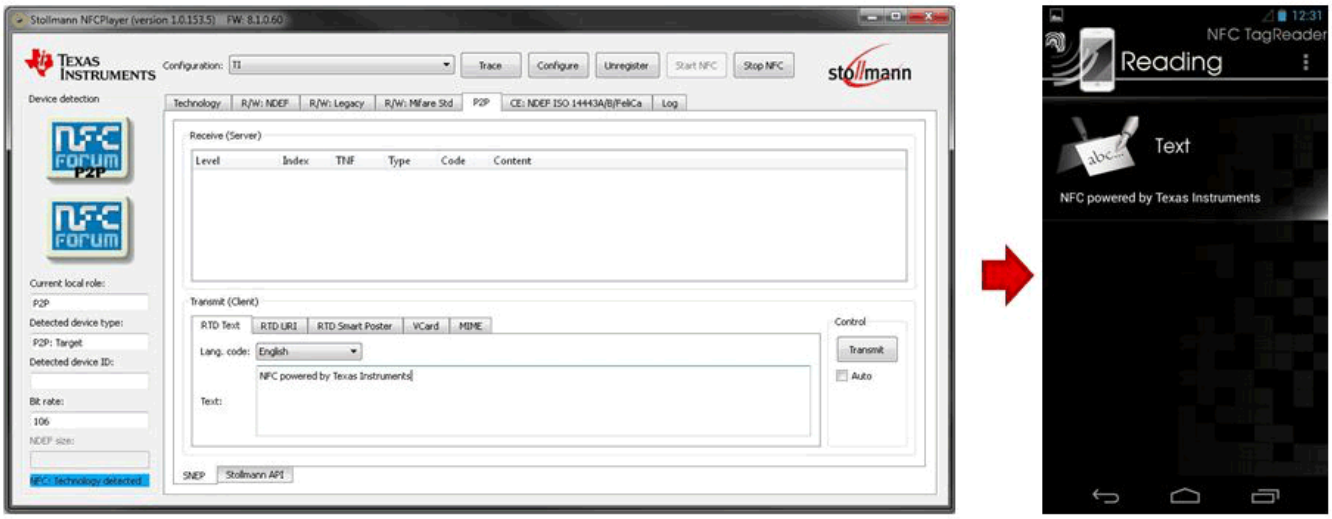

Figure 43.

When NFC-enabled device has application open for doing P2P, the GUI screen flips over to P2P mode, as before. Messages can now be sent from the NFC-enabled device to the GUI.

In Figure 44 the device is "beaming" a URL to a Texas Instruments website. Other message types can be sent as well, such as alternative radio handovers, text content, SmartPoster, phone numbers, applications, or images.

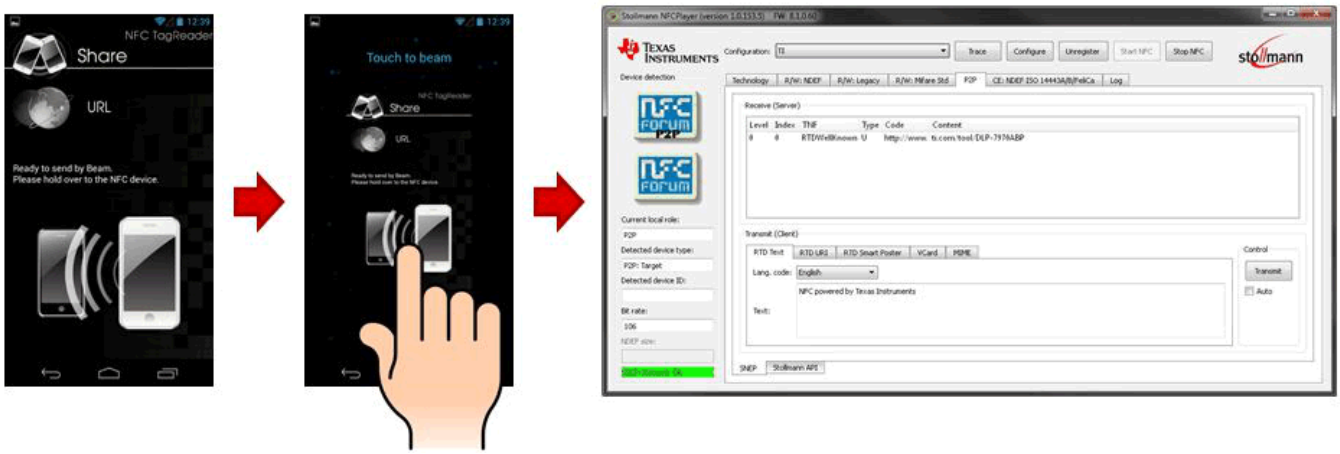

Figure 44.

You can also set up two of the hardware sets and run two instances of the GUI on the same PC, then demonstrate the P2P functionality without using a NFC handset. In Figure 45 we have sent an RTD MIME message (in the form of an image) from one hardware set to another using SNEP)

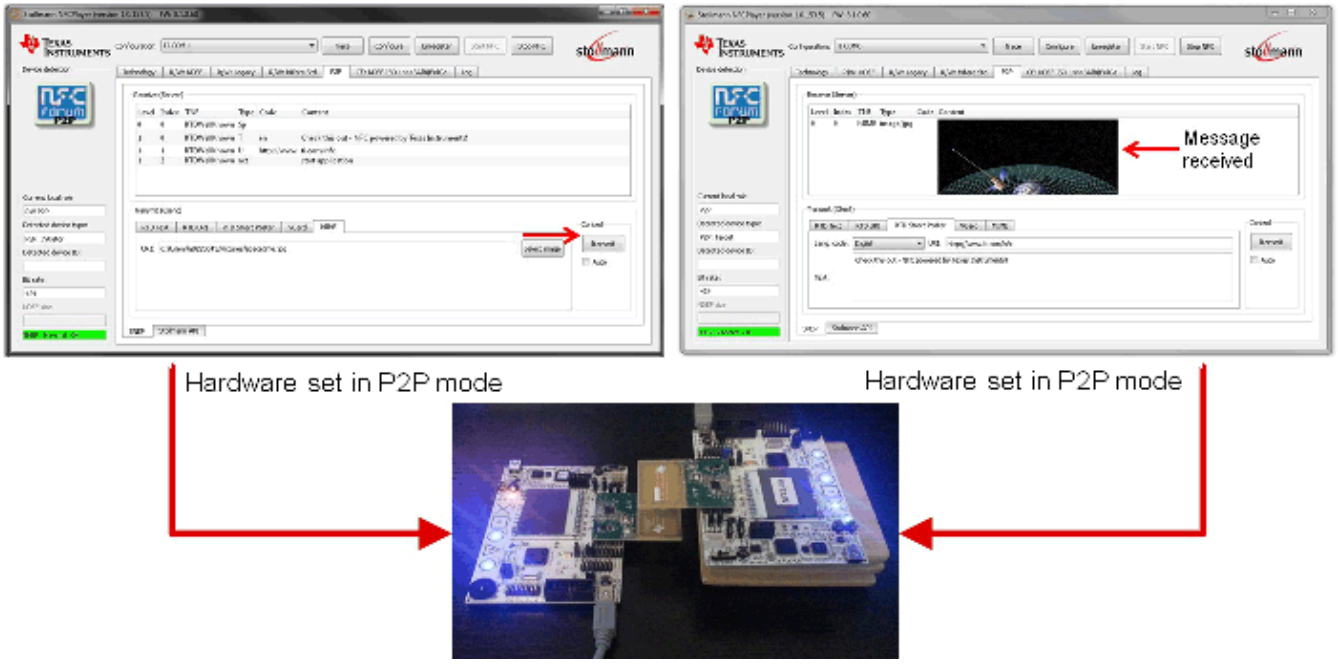

Figure 45.

User can also send over and receive all the other types with this hardware setup described. Here we have sent a RTD URI, SmartPoster, VCard from one hardware set to another (also using SNEP). It's important to realize here that once connection and symmetry are established, the device on either side can send a message to the other device, without changing any settings.

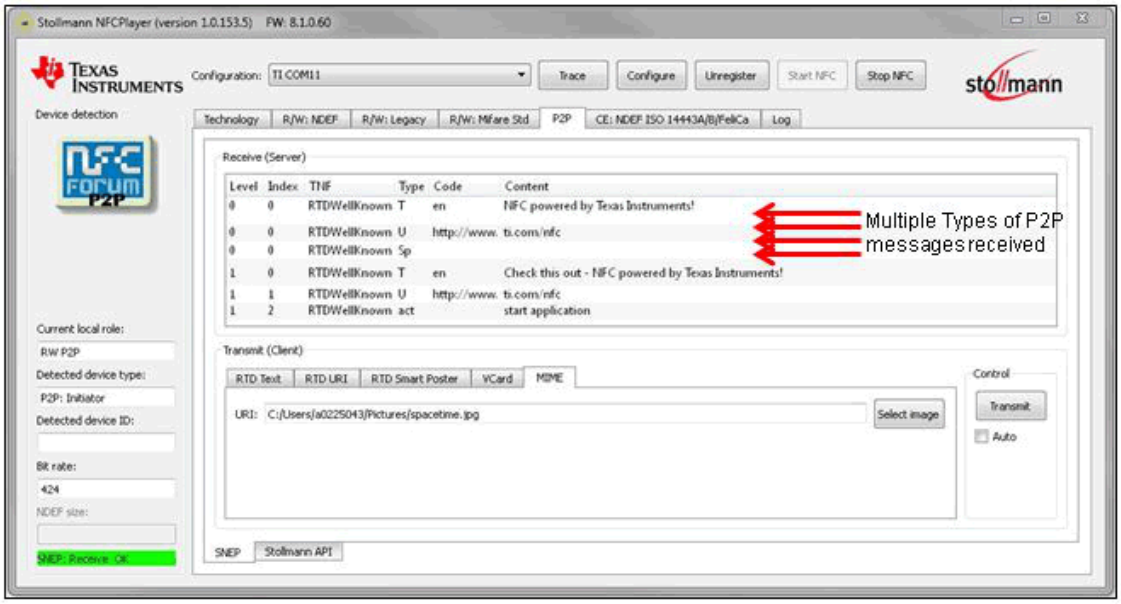

Figure 46.

Demonstrating the advanced features that NFC Peer-to-Peer offers, Figure 47 shows that you can see (once P2P connection is established) that a two-channel, bi-directional communication link can be arranged between two kits. This is useful for firmware updates or large data transfers where a user can set the device down for at least a few seconds (NFC Landing Pads…). Figure 47 shows two devices that are set up (one as the target the other as the initiator) and that have communication symmetry.

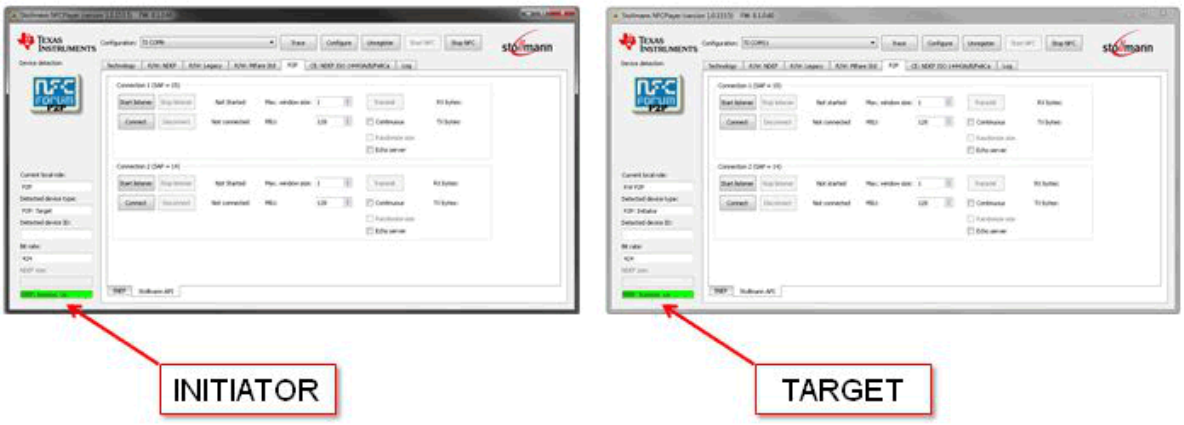

Figure 47.

We have chosen the Stollman API tab and started the listeners. The next step is to do a connect, on either side. Then, on either side channel – choose continuous under a TRANSMIT button and depress the TRANSMIT button. This will start a continuous stream of data to begin flowing over the NFCLink in two directions, using a NFC magnetic field based on the TRF7970A + MCU device, on either side.

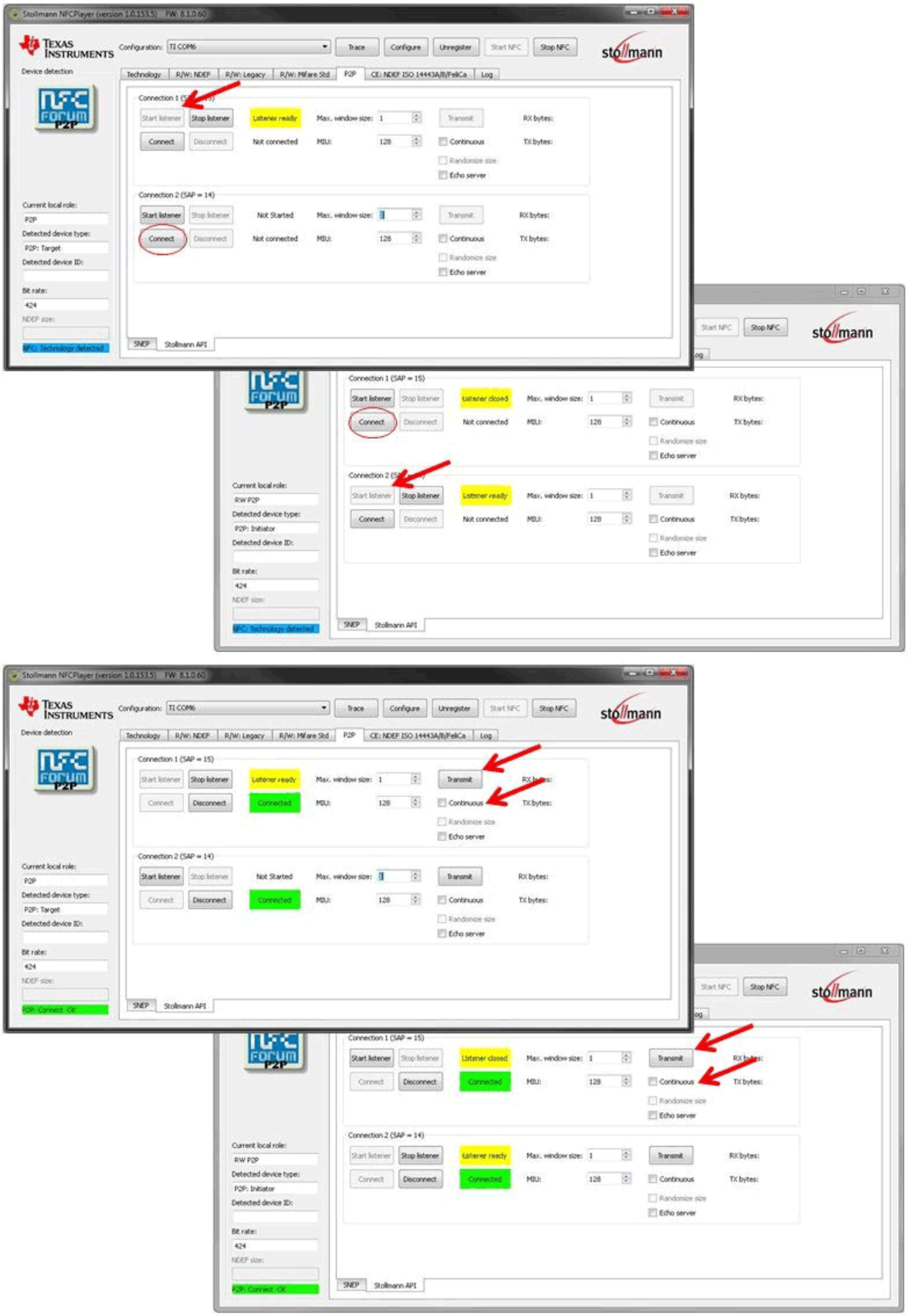

Figure 48.

Then the user can see data amounts streaming and in what direction.

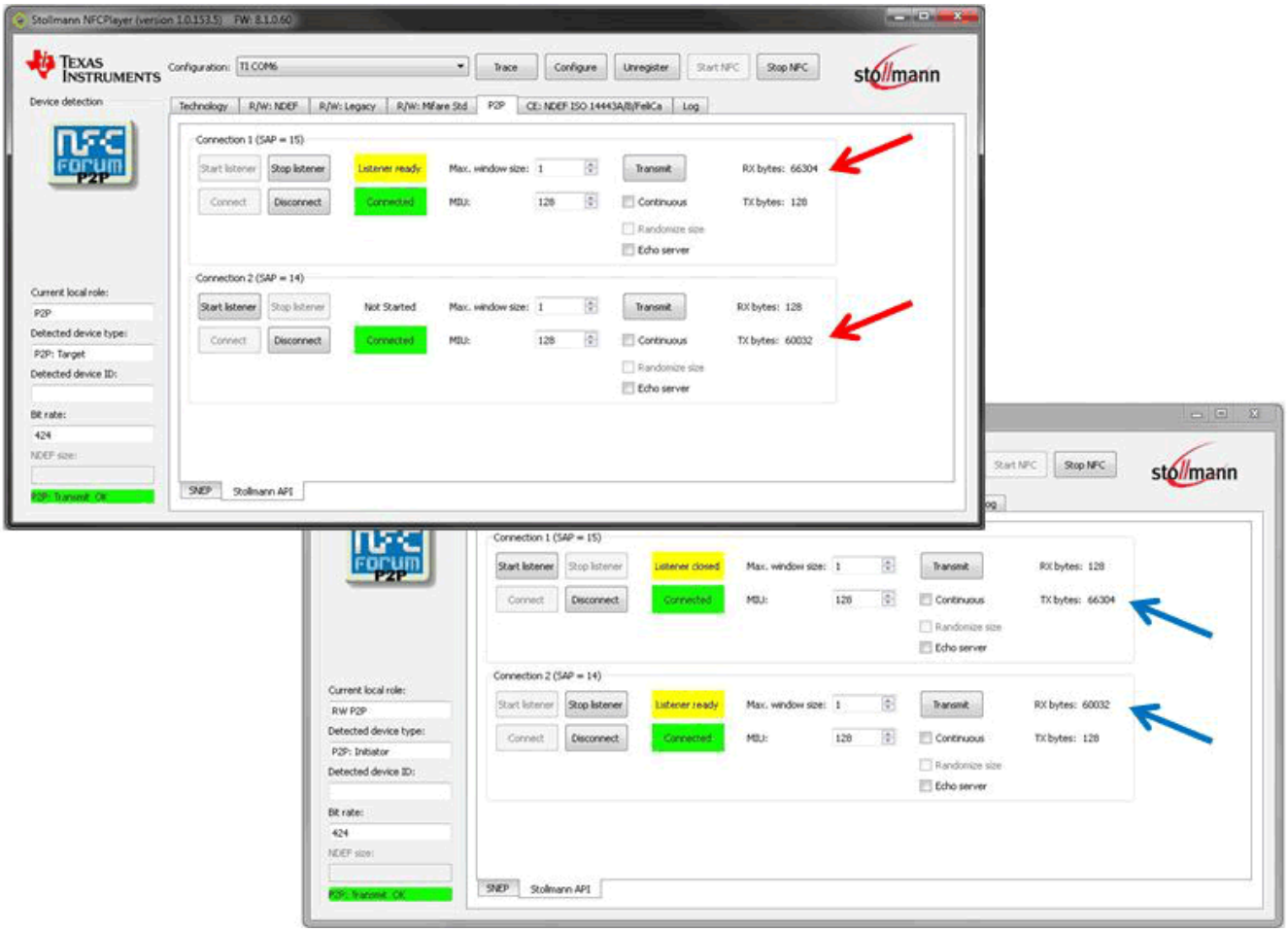

Figure 49.

16 NFC or RFID Card Emulation (CE) Mode Details

The TRF7970A can emulate (or be) a Type A or Type B card at 106 kbps. This section describes how to accomplish this with the use of the host stack and the embedded MCU library. This NFC mode is used for a variety of applications where larger tag sizes are needed or more advanced applications need to pass data sizes or methods which exceed the capabilities of passive tags. In a previous section, we showed the RF430CL330H, which is a dynamic tag with 3kB of memory space available, here we are showing something similar, but with 8kB of space. (the size of the emulated card is really limited only by the settings in the system, meaning, the card could be indicated to be much larger)

When hardware is in card emulation mode, and a NFC-enabled device acting as reader or writer is presented, the LED in the center of PAD1 will illuminate. The PAD1 LED will flash as the device approaches, then, when the field strength is sufficient it will stay on solid.

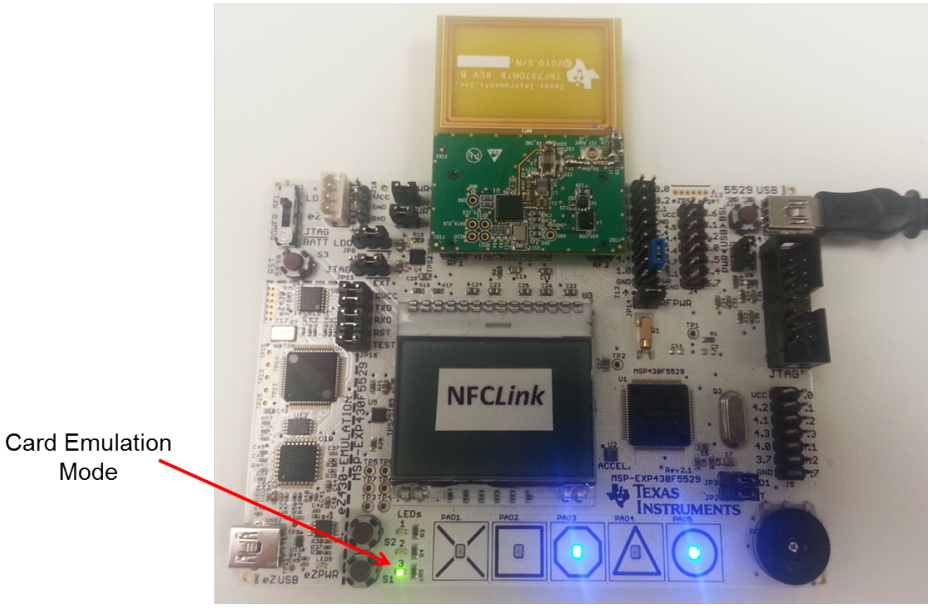

Figure 50.

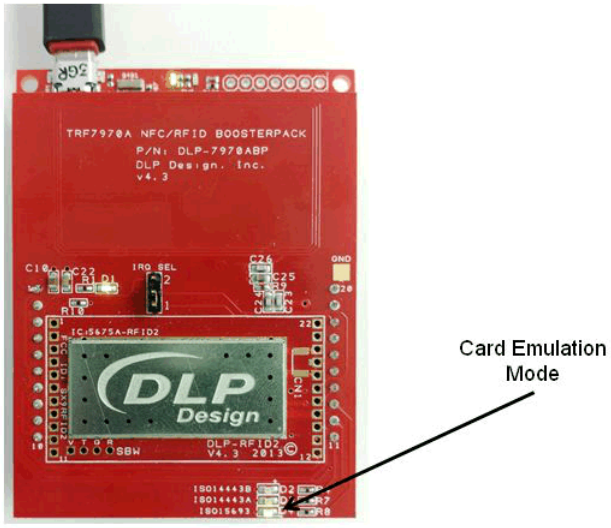

Figure 51.

When an NFC handset (or other NFC enabled reader or writer) is presented, the hardware will be read out just like a passive NFC or RFID tag. Figure 52 shows various screen captures from handset application called NFC TagReader (from KDDI)

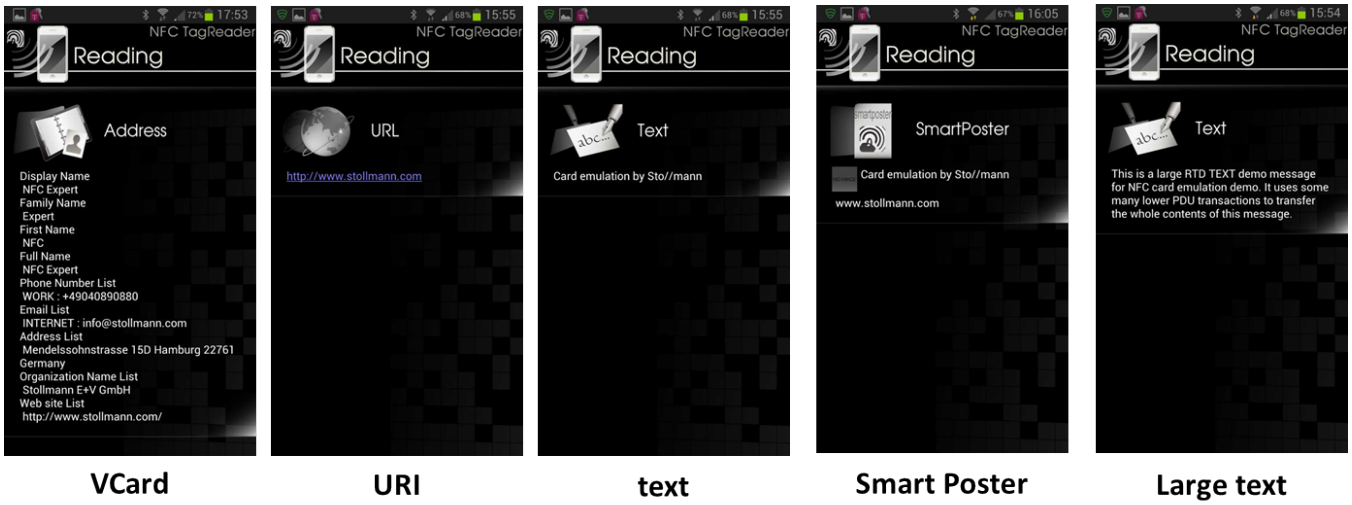

Figure 52.

When an NFC handset (or other NFC-enabled reader or writer) is presented, the hardware will be read out like a passive NFC or RFID tag. Figure 53 shows the GUI screen, as appears when showing the packet activity between the NFC handset (or other reader/writer) and the hardware.

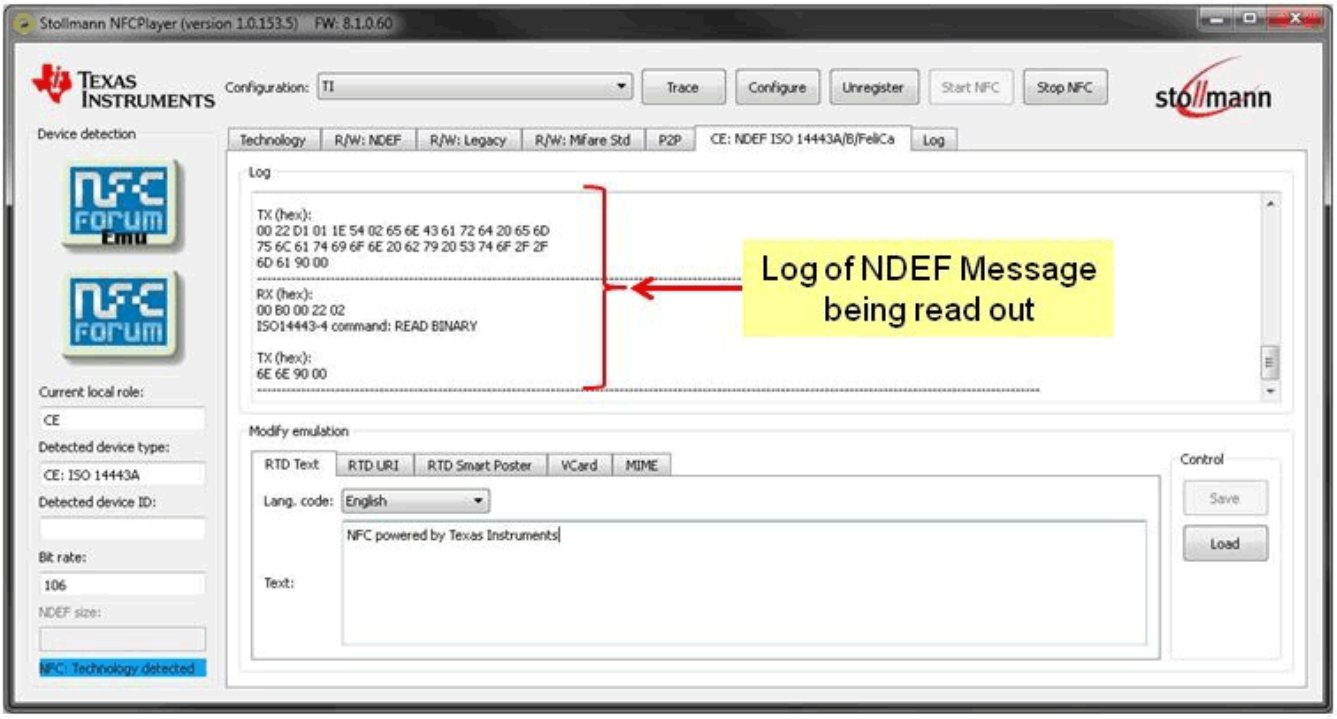

Figure 53.

You can also use a handset application (like the KDDI one) to write an NDEF message (an image for example) to the hardware.

It is possible to customize the contents of the different RTD types being emulated. To do this, open up the CE: NDEF ISO14443A/B/FeliCa tab when Card Emulation is enabled (see Figure 54). There are tabs for each RTD type that NFCLink supports where it is possible to input personalized data for emulation.

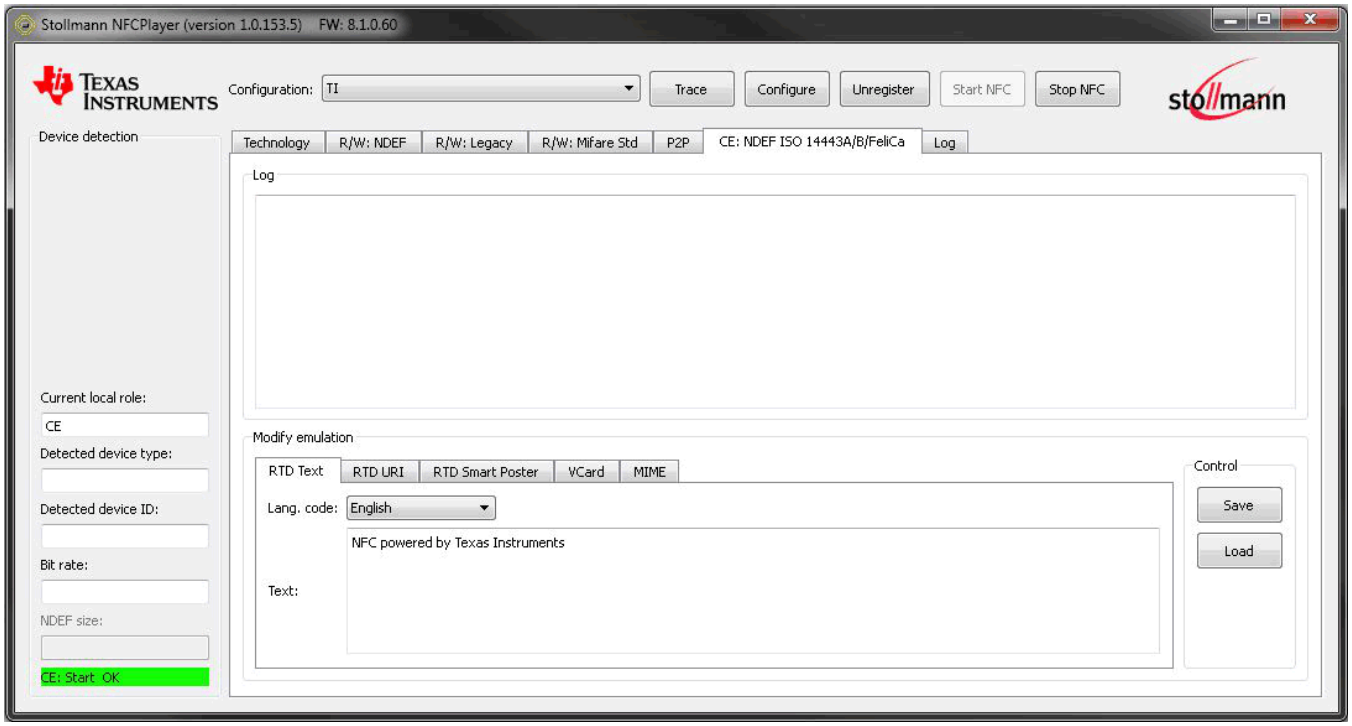

Figure 54.

To store the edited RTD for use by the GUI, the Save button must be pressed. This also overwrites the default RTD type chosen when Start CE is pressed. To see the currently loaded RTD type and the contents of its message, click the Load button.

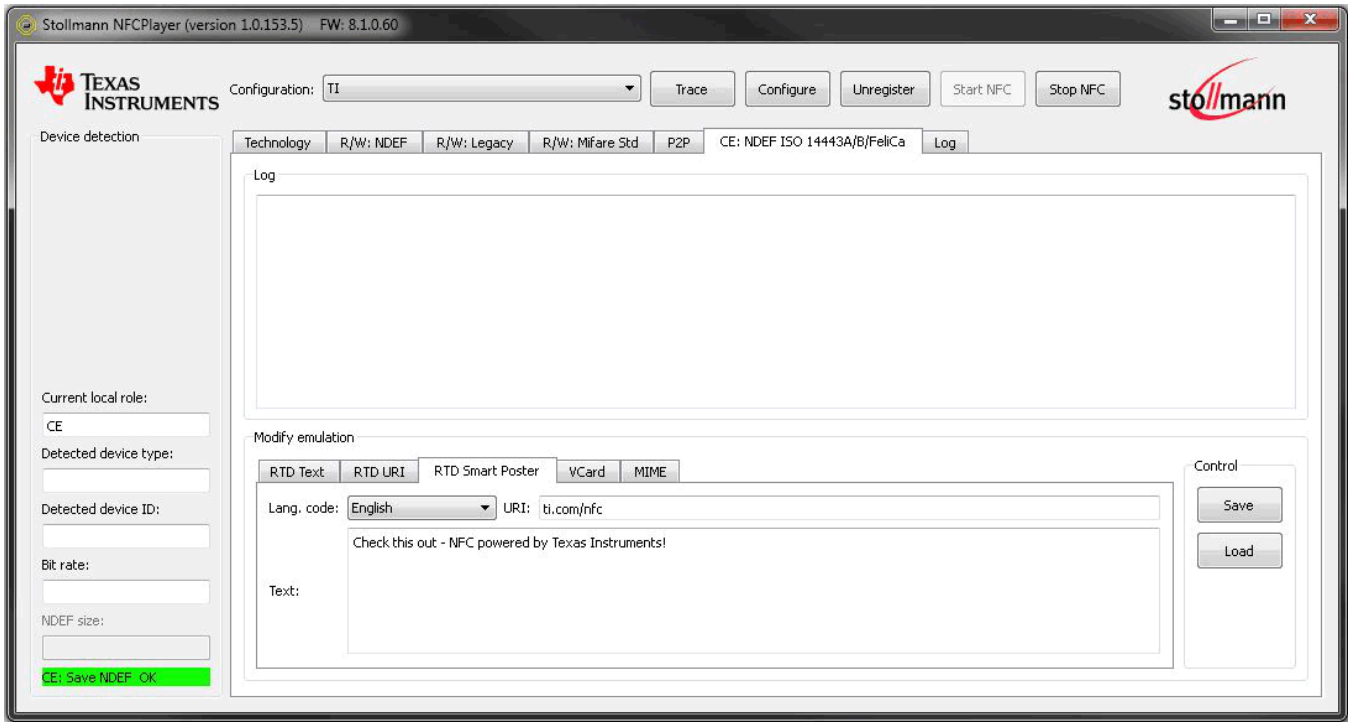

Figure 55.

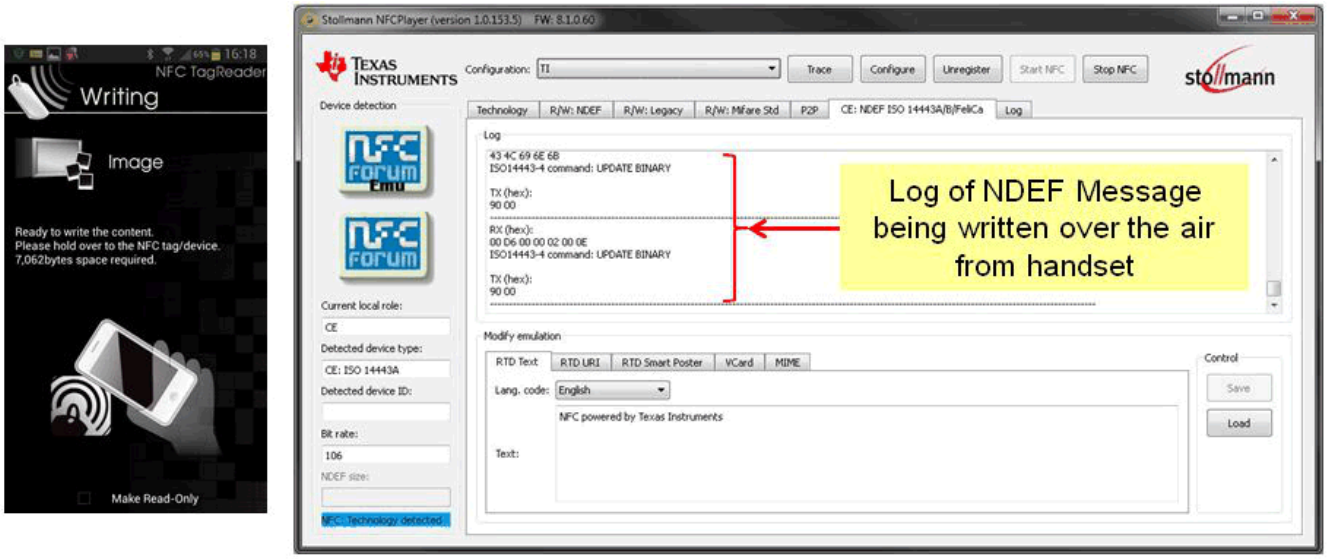

Figure 56.

Then, the application can read back the image (or other data) from the hardware.

You can also set up two of the hardware sets and run two instances of the GUI on the same PC (with one as the reader and writer and the other in card emulation mode) then read and write data without using a NFC handset. (Figure 57 we have written an image of approximately 5.7kB and have read it back)

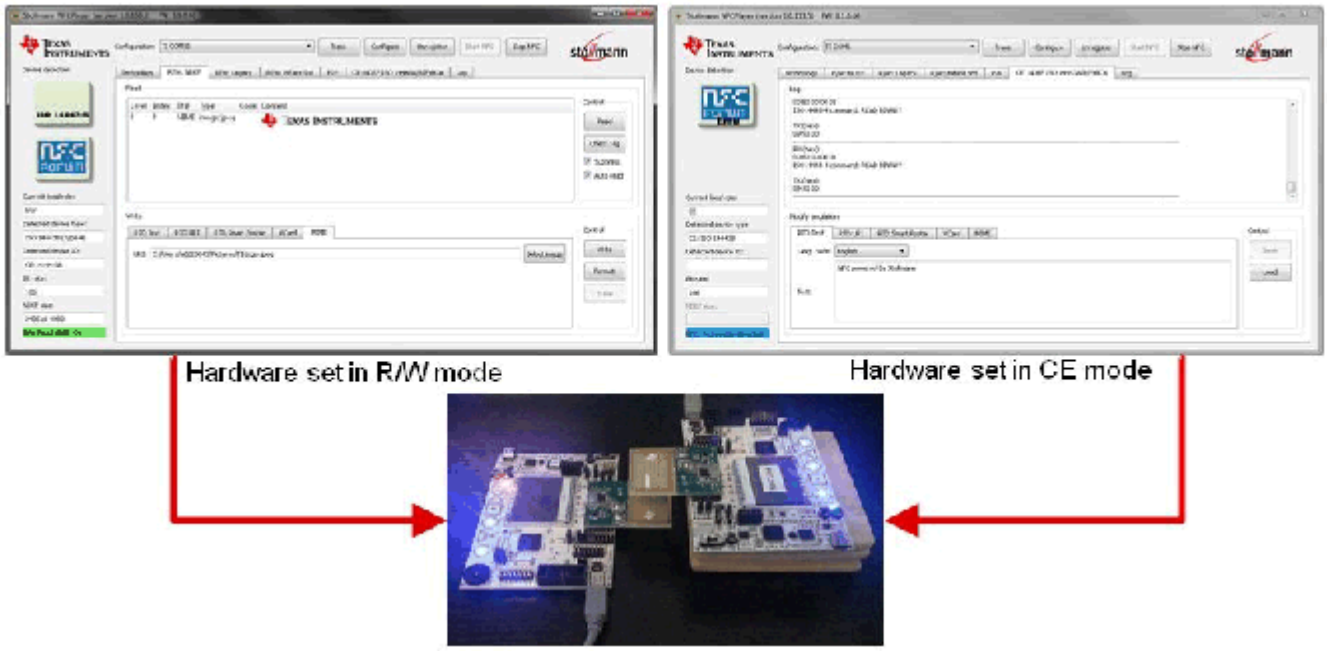

Figure 57.

17 Project File Structure

The NFCLink installation places the following folders at this location:

`C:\TI\msp430\NFCLink_1.0.0.3`

doc — User Guide\Quick Start Presentation. (similar to this document)

examples\allModes — Reader/Writer, Peer to Peer, and Card Emulation projects.

RW_P2P_CE_1: USB (CDC) interface to the host.

RW_P2P_CE_2: UART Module interface to the host.

Libraries — The libraries used by the CCS projects.

driverlib – UART, GPIO and Timer drivers.

nfclinklib – library for NFC operations

usblib430 – USB drivers for the MSP430F5529.

utils — NFC Player installer – Host executable for Windows OS. The program will be installed in `C:\TI\NFCStack+Eva_Windows TI R6.1.153.5`.

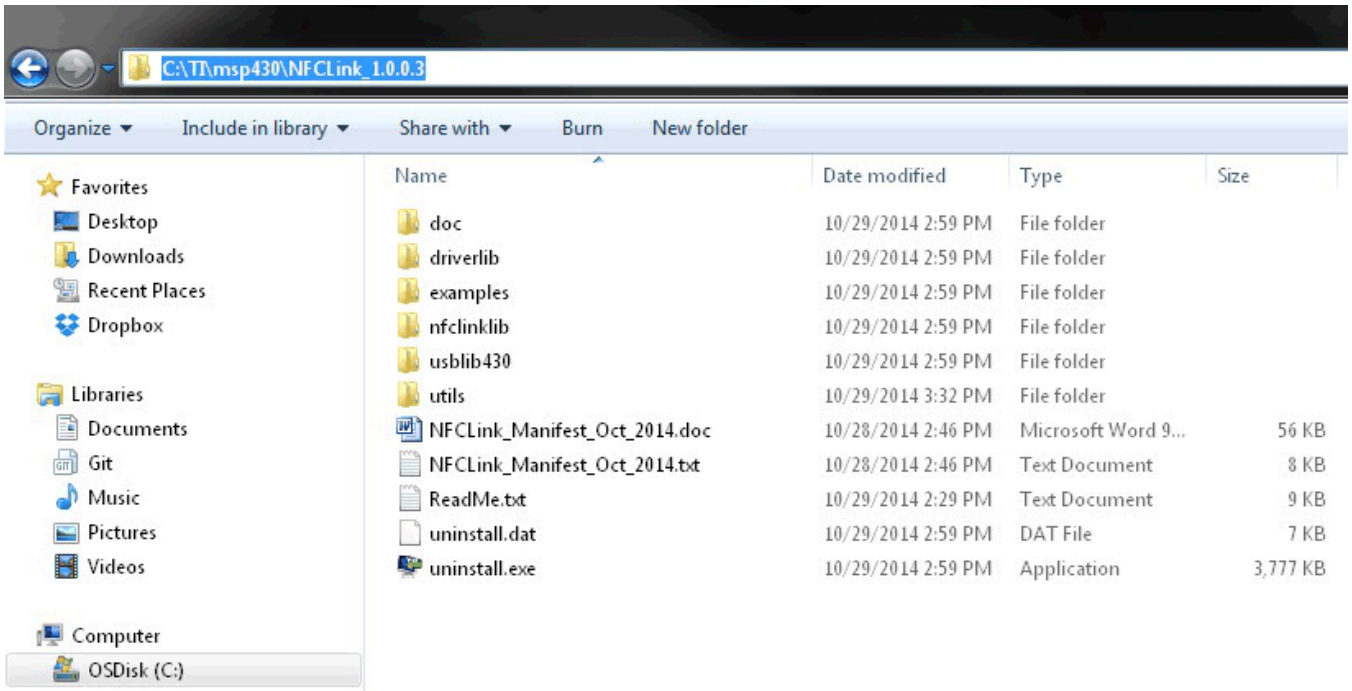

Figure 58.

18 Porting to Other MCUs

The following layers must be modified when porting to other MSP430 MCUs:

1. **Host interface –** Code examples include USB and UART. Future releases will include SPI and I^2C implementations – for example, using an Aardvark I2C/SPI Host Adapter for testing.
2. **MSP430 hardware –**Main application, MSP430 MCLK, Watchdog Timer (WDT), GPIOs (for LEDs and debugging purposes), and one timer.
3. **TRF7970A transceiver interface –** SPI with Slave Select module.

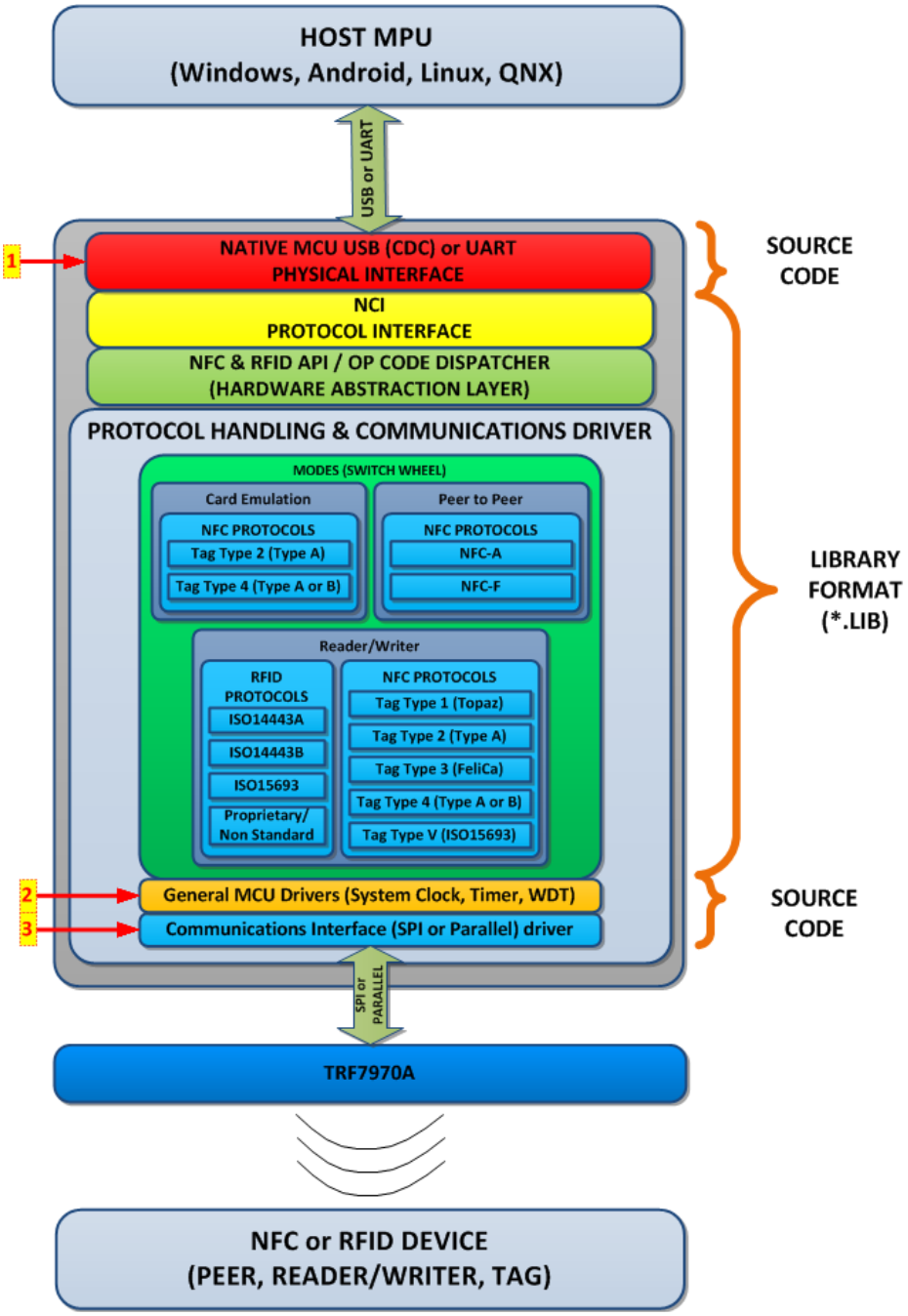

Figure 59.

18.1 Host Interface Modifications

There are three functions that must be modified to interface with the host.

Host_Interface::Init() — Initialization for the module and RX ISR. For RW_P2P_CE1 the function initializes and configures the USB module.

Host_Interface::Write() — Function that transmits to the host. For RW_P2P_CE2 the function writes to the UART_TX buffer with len bytes.

The RX ISR for the host interface— Store incoming bytes into Host_Interface::recbuf.

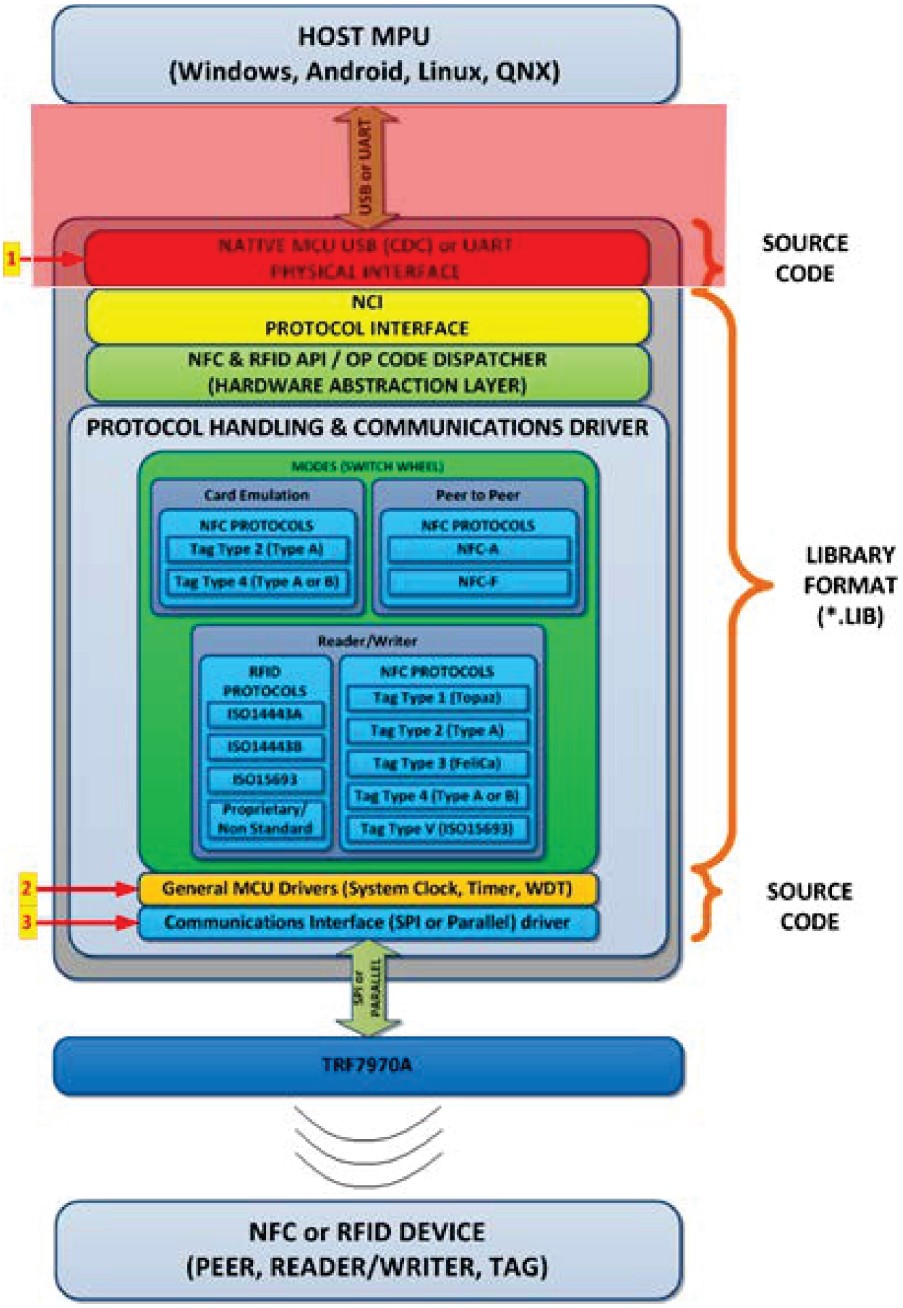

Figure 60.

18.2 General MSP430 Modifications

There are five functions that must be reviewed:

HW_Config::Init() — Disables the WDT. Sets up the frequency of the MCLK – the current implementation uses the 32.768-kHz crystal (ACLK) , MCLK = DCO = 25 MHz.

HW_Config::MCU_Reset() — Reset the MSP430 by setting the BOR flag (this can be modified to a software power-on reset depending on the MCU). When a host reset command is received, this function is used to reset the MCU.

HWTimer::Init() — Initialize Timer A using reference of ACLK (32.768kHz) running continuously.

HWTimer::ticks32() — Returns and stores the value of the timer's counter into timervalue.f[0]. Furthermore **TICKS_PER_MSEC** (inside MSP430_hardware.h) needs to be updated based on the CLK being used as reference (for example, for a 2-MHz Clock → TICKS_PER_MSEC = Reference CLK / 1000 = 2×10^6 / 1000 = 2000)

GPIO::Init() — Initializes the External Field LED (P1.1), External Field debug pin (P4.1), Any Mode LED (P1.3), RW/Initiator Mode LED (1.0), P2P LED (P8.1), CE (8.2), Serial TX debug pin(4.3) and Serial RX debug pin (4.2). These GPIOs will be helpful to provide feedback to our team.

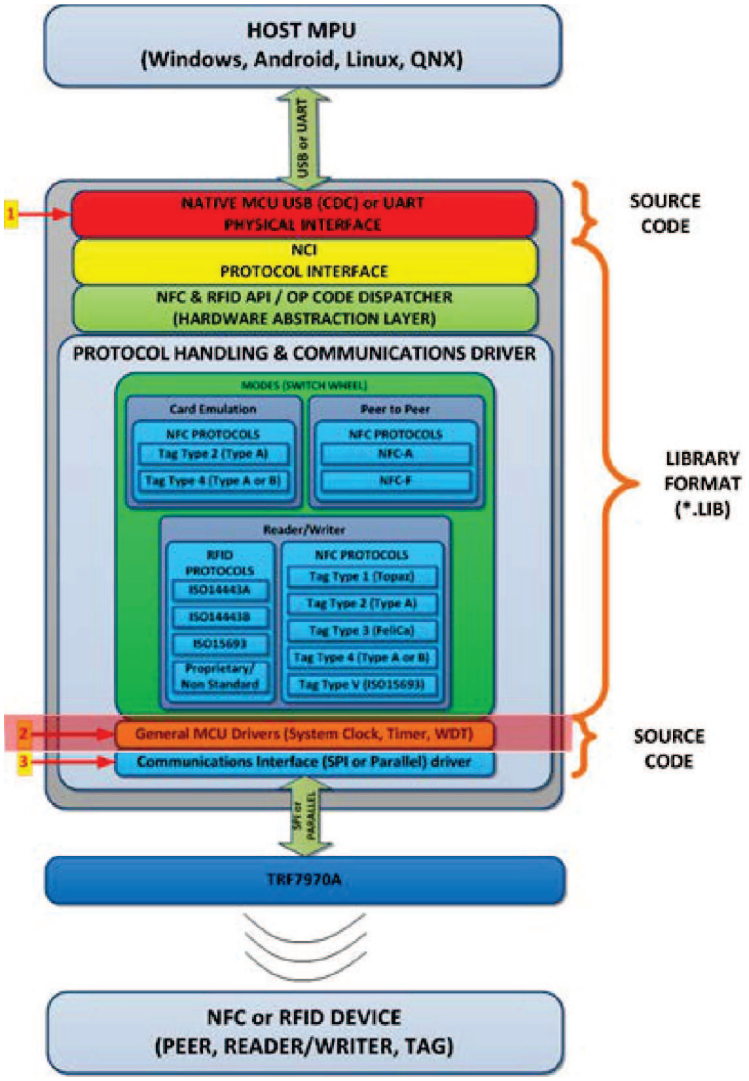

Figure 61.

18.3 TRF7970A SPI Driver Modifications

There are five functions that must be reviewed:

TRF797x_setup() — Initializes the interface to the TRF7970A (SPI / Parallel). Initializes the TRF7970 EN pin, then sets up the IRQ pin with a rising edge interrupt. Afterwards, it writes to the TRF7970 to ensure it has been initialized properly (RFID.cpp and RFID.h).

Note: At this time, only SPI with SS is supported.

RFSPI::init() — Initializes the SPI module as 3-pin SPI, 8-bit Master, MSB, Clock Phase = 0, SPI Clock ≈ 4 MHz using the SMCLK = 25 MHz as reference. The Slave Select is manually set – ensure that SLAVESELECT _OUTPUT is using the correct GPIO (SPI.cpp and SPI.h)

Note: See the datasheet to match

TRF797x_IRQ_ISR() — Ensure that the interrupt service routine for the IRQ pin (rising edge interrupt) is setup correctly. (RFID.cpp)

RFSPI::waitBus() — Waits for the SPI module to be idle. (Needs to be modified depending on the USCI) (SPI.cpp)

RFSPI::transfer() — Writes to the SPI TX register, and returns the value of the SPI RX register. (Ensure USPITXBUF and USPIRXBUF are defined appropriately). (SPI.h)

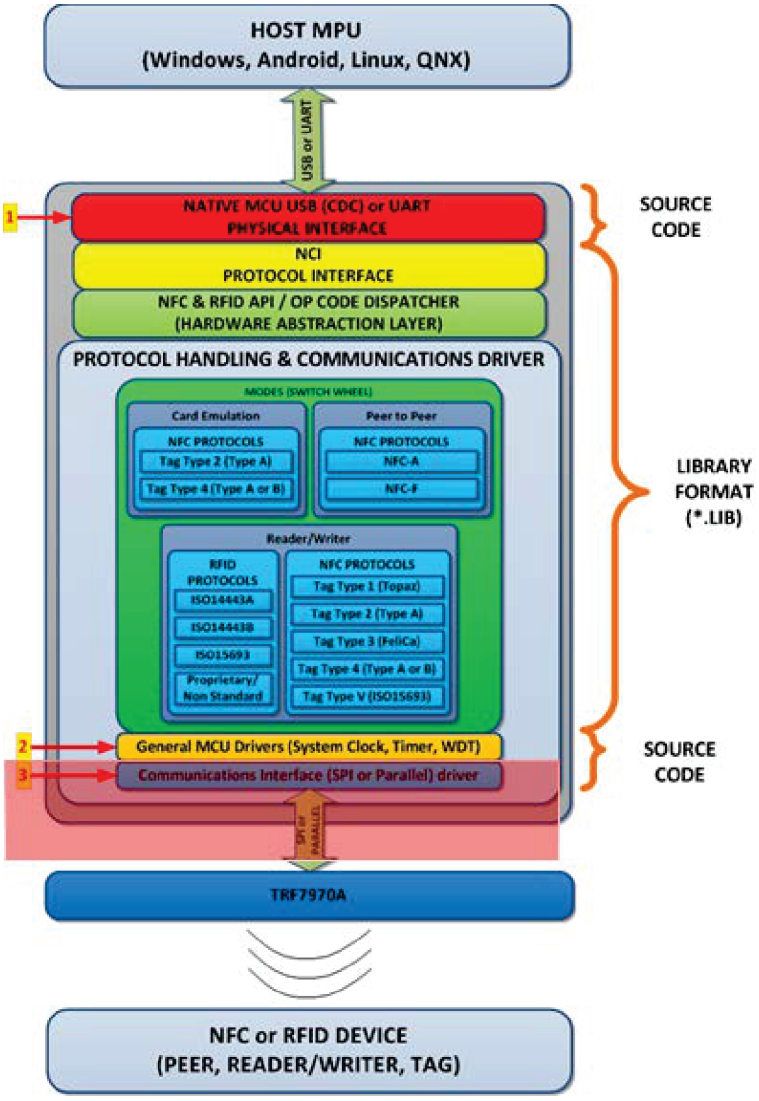

Figure 62.

19 Memory Footprints

19.1 CCS Optimization Options

There are two knobs available for optimizing the application:

1. Optimization Level (range 0 to 4)
2. Control speed vs. size (range 0 to 5)

Note: For the smallest memory footprint, the control speed vs. size must be set to 0.

The following section describe a memory footprint for the examples in the release and the internal development version – by only modifying the optimization level. Because the library and the CCS project can be optimized independently, the table for the USB includes the best and worst cases scenarios.

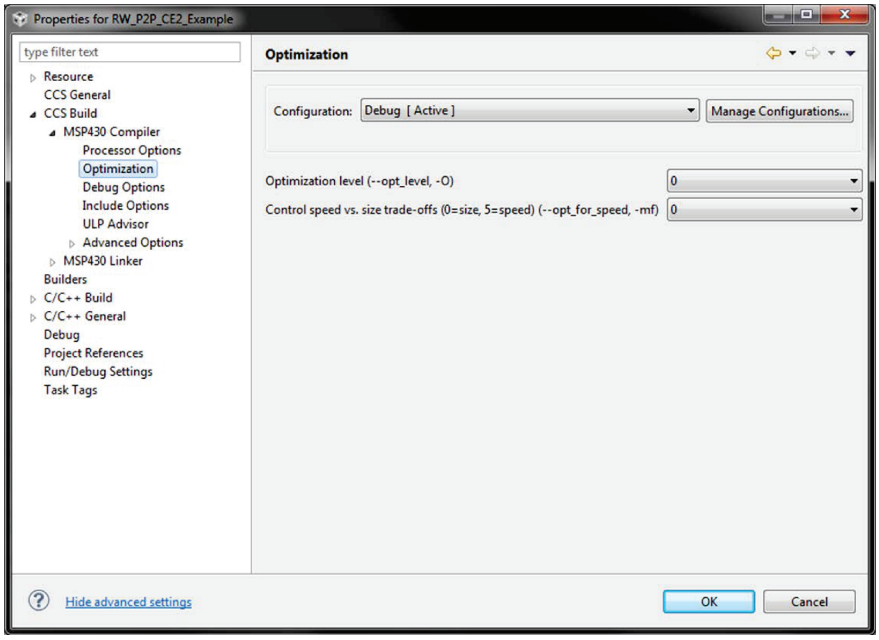

Figure 63.

Table 1. CCS Memory Footprint

Mode + USB CDC[(1)]	Optimization Level (0-4)	Flash (kB)	RAM (kB)
All Modes	0	113.2	5.7
	0	104.6	5.7

(1) CCS project stand alone with library

Note: The control speed and size tradeoffs (0 = size, 5 = speed) for all the memory footprints was set to 0 – optimized for smallest footprint (size).

20 References

- *TRF7970A Multi-Protocol Fully Integrated 13.56-MHz RFID and NFC Transceiver IC* (SLOS743)
- *MSP-EXP430F5529 Experimenter Board User's Guide* (SLAU330)
- ISO14443-3, ISO14443-3, ISO14443-4 (ISO/IEC 14443-2:2010, ISO/IEC 14443-3:2011, ISO/IEC 14443-4:2008)
- ISO15693-2, ISO15693-3 (ISO/IEC 15693-2:2006, ISO15693-3:2009)
- ISO7816-4 (ISO/IEC 7816-4:2013)
- ECMA340 (NFCIP-1)
- ISO18092 (NFCIP-1)
- ECMA352 (NFCIP-2)
- ISO21481 (NFCIP-2)
- NFC Forum Specifications (NFC Forum Specs)

Revision History

Changes from Original (May 2014) to A Revision **Page**

- Changed the exe name in the first list item following "Other components of the release are:" 6
- Added Figure 6 7
- Changed Figure 7 9
- Changed Figure 8 9
- Changed step (5) 11
- Changed Figure 13 11
- Changed the path in of Section 7.3.2 12
- Changed Figure 15 12
- Changed Figure 16 13
- Changed the inf file name and path in Section 8 14
- Changed Figure 19 15
- Changed Figure 20 15
- Changed the folder name in step (3) 15
- Changed Figure 21 15
- Changed Figure 22 16
- Changed Figure 23 17
- Changed Figure 24 17
- Changed Figure 25 18
- Added Figure 27 19
- Changed Figure 28 20
- Added Figure 30 21
- Changed Figure 31 22
- Added Figure 33 23
- Changed Figure 34 24
- Changed Figure 35 25
- Changed Figure 36 25
- Changed the first paragraph of Section 14.4 26
- Changed Figure 37 26
- Changed Figure 38 26
- Changed the "Unique ID" in the first paragraph of Section 14.5 27
- Changed Figure 39 27
- Changed Figure 40 27
- Changed Figure 41 28
- Changed Figure 42 28
- Changed Figure 43 29
- Changed Figure 44 30
- Changed Figure 45 30
- Changed Figure 46 31
- Changed Figure 47 31
- Changed Figure 48 33
- Changed Figure 49 34
- Added Figure 51 35
- Changed Figure 53 37
- Added the paragraph that starts "It is possible to customize..." through Figure 55 37
- Changed Figure 56 39
- Changed Figure 57 39
- Changed the path in the first paragraph of Section 17 40
- Changed the path in the "utils" entry 40
- Changed Figure 58 40

- Changed all contents of Table 1 45
- Removed *Table 2 RW_P2P_CE_1 Memory Footprint* and *Table 3 RW_P2P_CE Memory Footprint* 45

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

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