

Introduction to *NFCLink*

**Texas Instruments
Safety and Security MCU
NFC/RFID Applications Team**

6/18/2013

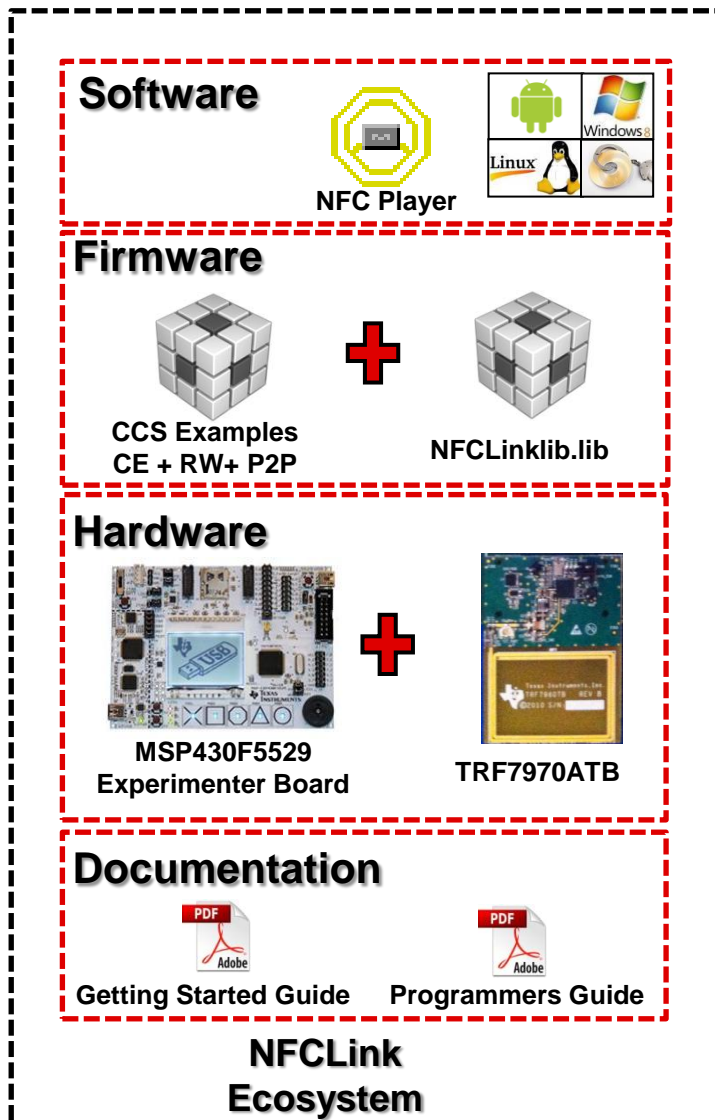
Agenda

- NFCLink Overview
- NFCStack + Evaluation Overview
- NFCLink Beta Release components
- Software + Hardware Roadmap
- NFCLink Demo – “Out of box experience”
 - Hardware + Software Configuration
 - Demonstration examples
 - Reader/Writer
 - Card Emulation
 - Peer to Peer
- NFCLink File Structure
- Porting to other MSP430s
- Memory Footprints
- Summary

NFCLINK OVERVIEW

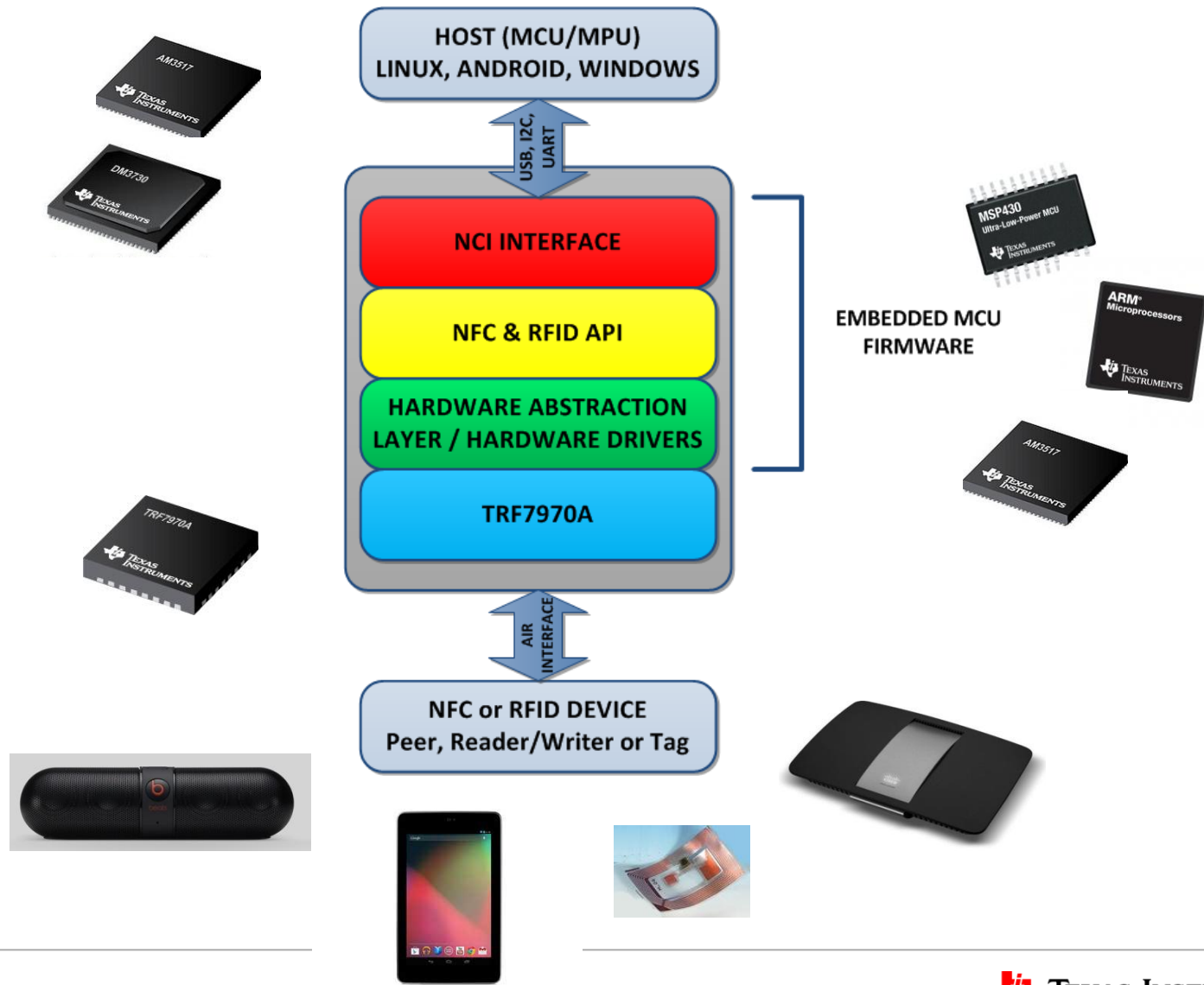
6/18/2013

What is *NFCLink* and what is it used for?



6/18/2013

NFCLink Architecture



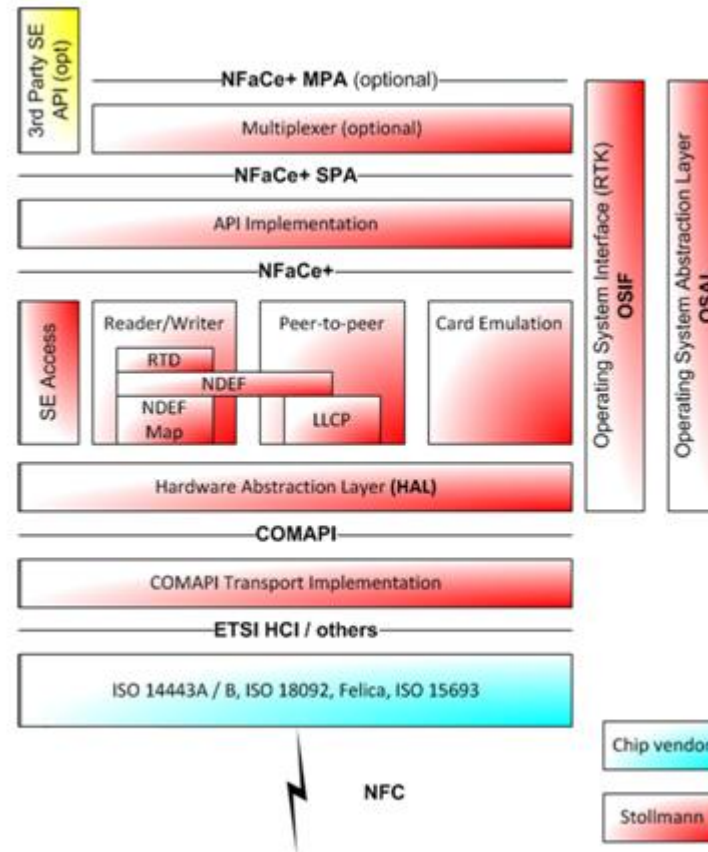
6/18/2013

NFCSTACK + EVALUATION **OVERVIEW**

6/18/2013

What is NFCStack+Eva and what is its Architecture?

- High level overview of the Windows GUI implementation.
In this case the TRF7970A + MSP430 would be the chip vendor.



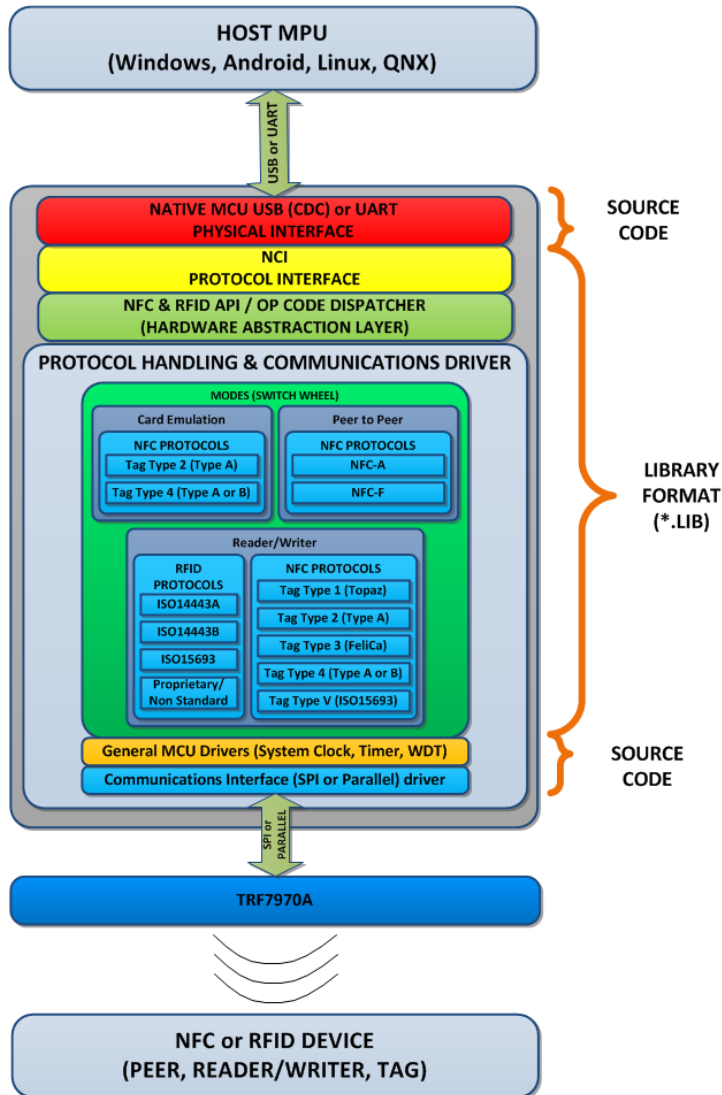
Where to Integrate the NFCStack

- In the first implementation which is releasing on June 18th, 2013 – the NFCStack+ is compiled as a Windows OS GUI with an installer, for demonstration purposes.
- 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 w/memory (commonly used for high volume/lower 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 NFC*Link* solution is currently.

NFC BETA RELEASE **COMPONENTS**

6/18/2013

What is going to be in June 18 release



Other components of the release will be:

- NFCStack+Eva_r6.0.47.5_RC.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
- Getting Started Guide (for the demo)

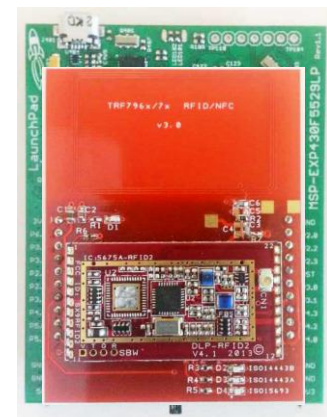
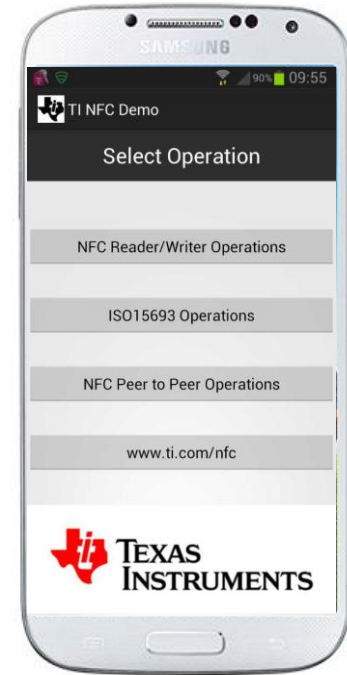
6/18/2013

SOFTWARE & HARDWARE **ROADMAP**

6/18/2013

Roadmap of what is coming after June 18th

- After June 18th:
 - Integration into MSP430Ware (next CCS release is August 2013)
 - Support for IAR IDE
 - Support for other TI MCU families will be available
 - Cortex M4 (Tiva)
 - Cortex A8 (Sitara)
 - After these MCUs are supported – next TI MCU platforms to be supported are TMS470/570 (Hercules) and other automotive qualified MCUs.
 - Android (2.x, 3.x and 4.x), Linux, QNX support
 - Support for MIFARE DESFire EV1 (in AES mode), Plus, and Classic
 - Support for NFC Tag Type 1 Platform (Topaz)
 - Quarterly updates with MSP430Ware
 - TI NFC/RFID Android Application
 - MSP430F5529 LaunchPad + TRF7970A BoosterPack (from DLP Design)



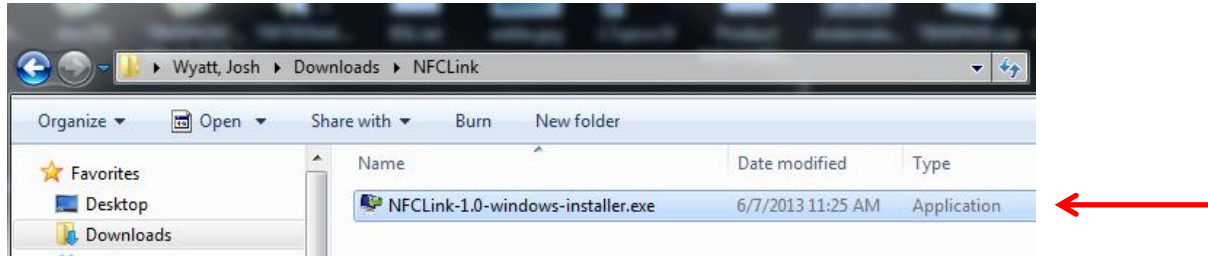
6/18/2013

NFCLINK DEMO **“OUT OF BOX EXPERIENCE”**

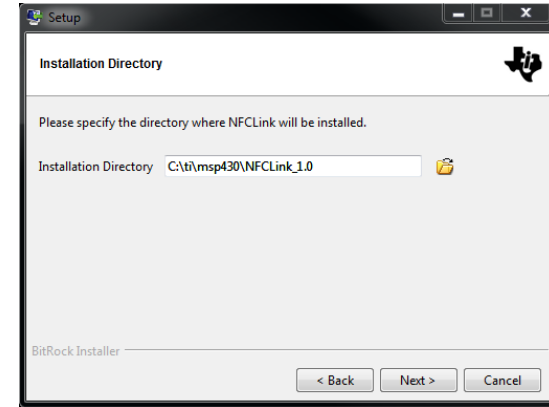
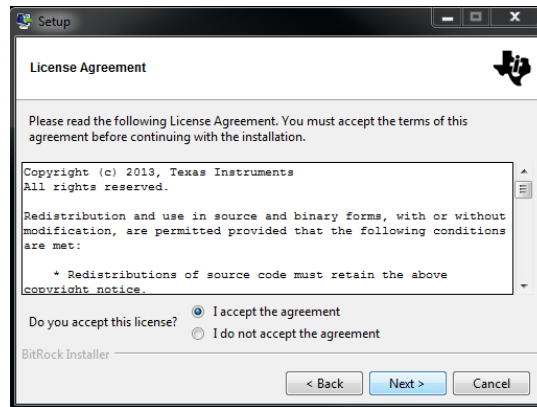
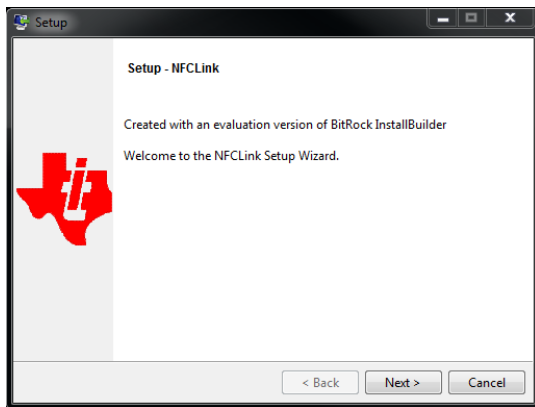
6/18/2013

Install NFCLink-1.0

- Use the NFCLink Installer executable to get the firmware code project and the NFCPlayer GUI loaded onto PC.

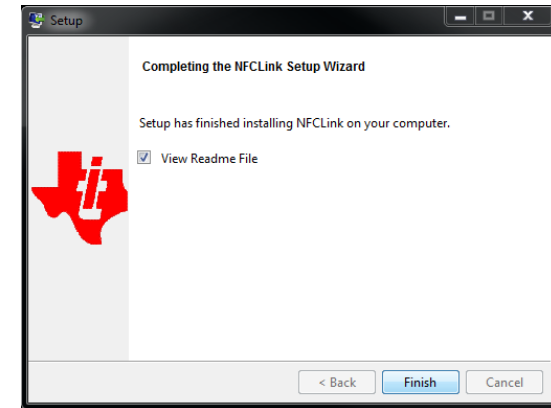
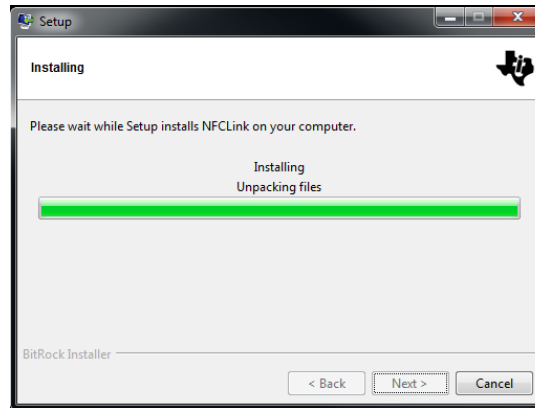
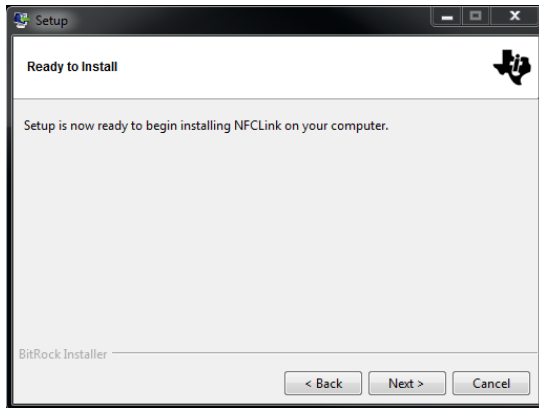


- Double click installer and follow instructions

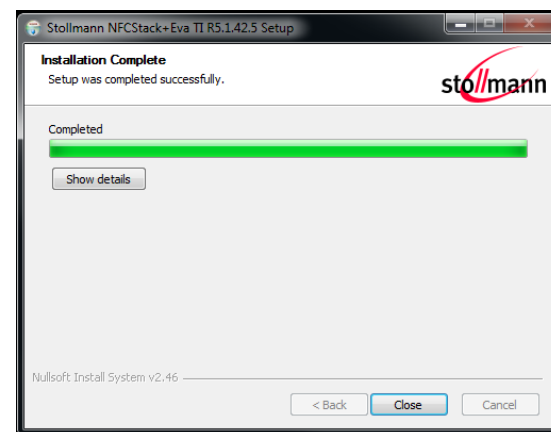
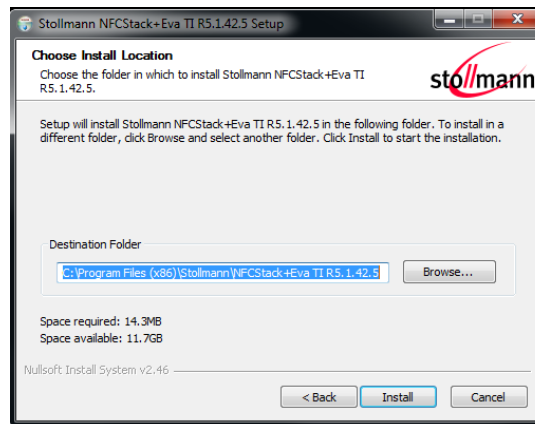
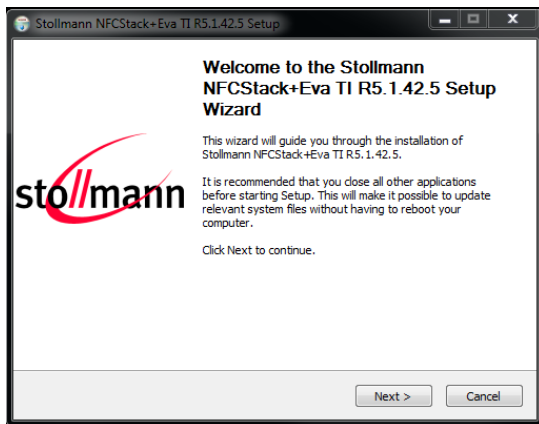


- Continues on next slide

Install NFCLink-1.0 (cont.)

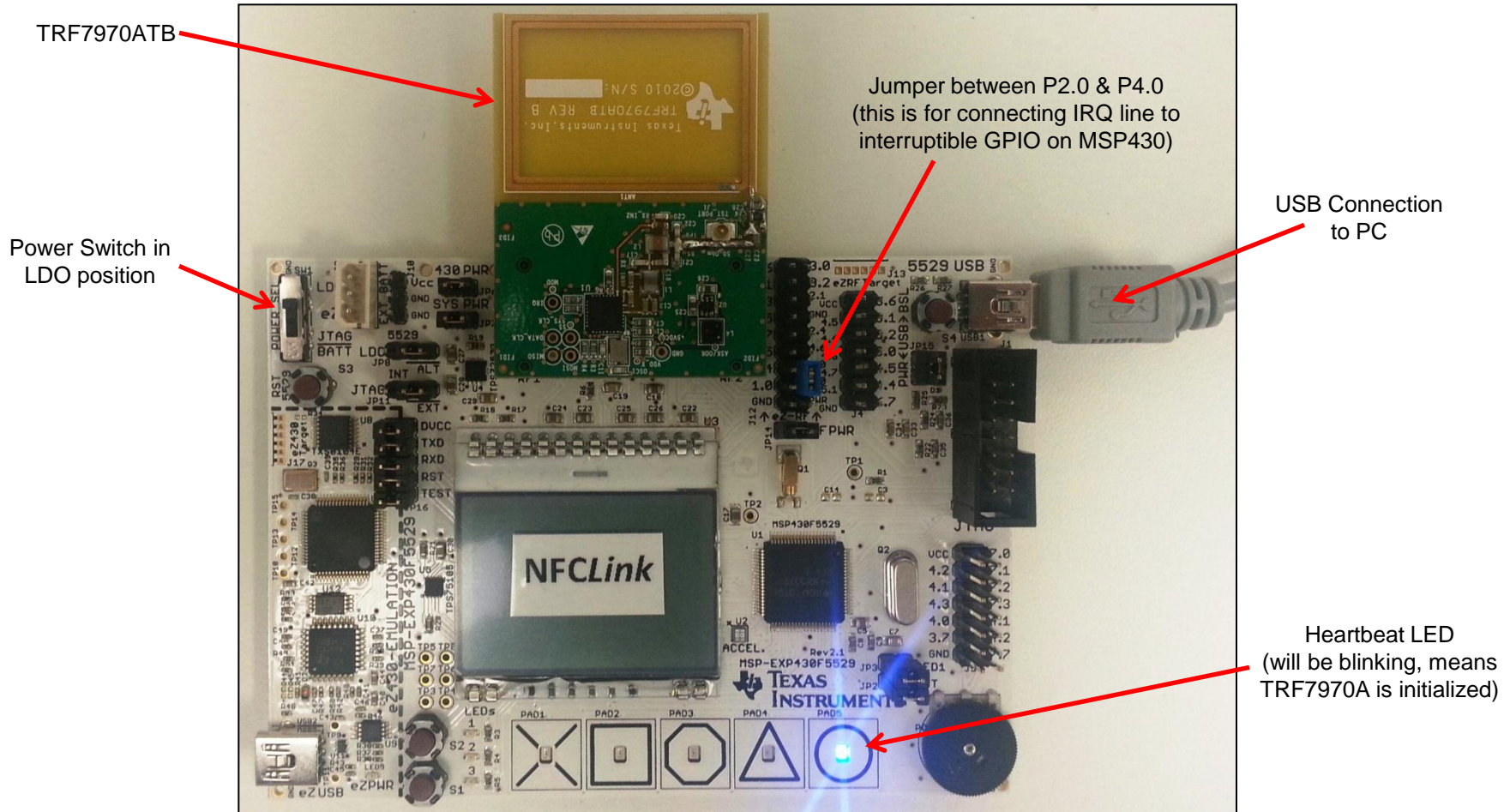


- Then the NFCPlayer GUI will begin installation automatically



- When complete, just press close button and plug the hardware into USB port on PC. ([see picture on next slide](#))

Hardware Configuration Image

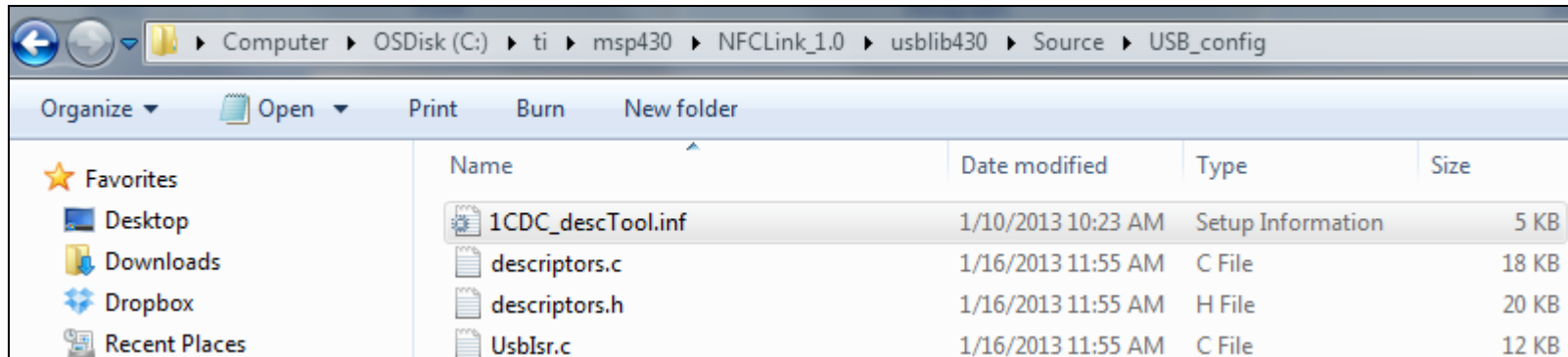


6/18/2013

Install USB Driver

- The first time the hardware is plugged into PC, the USB driver will need to be installed by pointing the wizard to the correct location/path.
- The file path for device manager wizard is:

C:\ti\msp430\NFCLink_1.0\usblib430\Source\USB_config

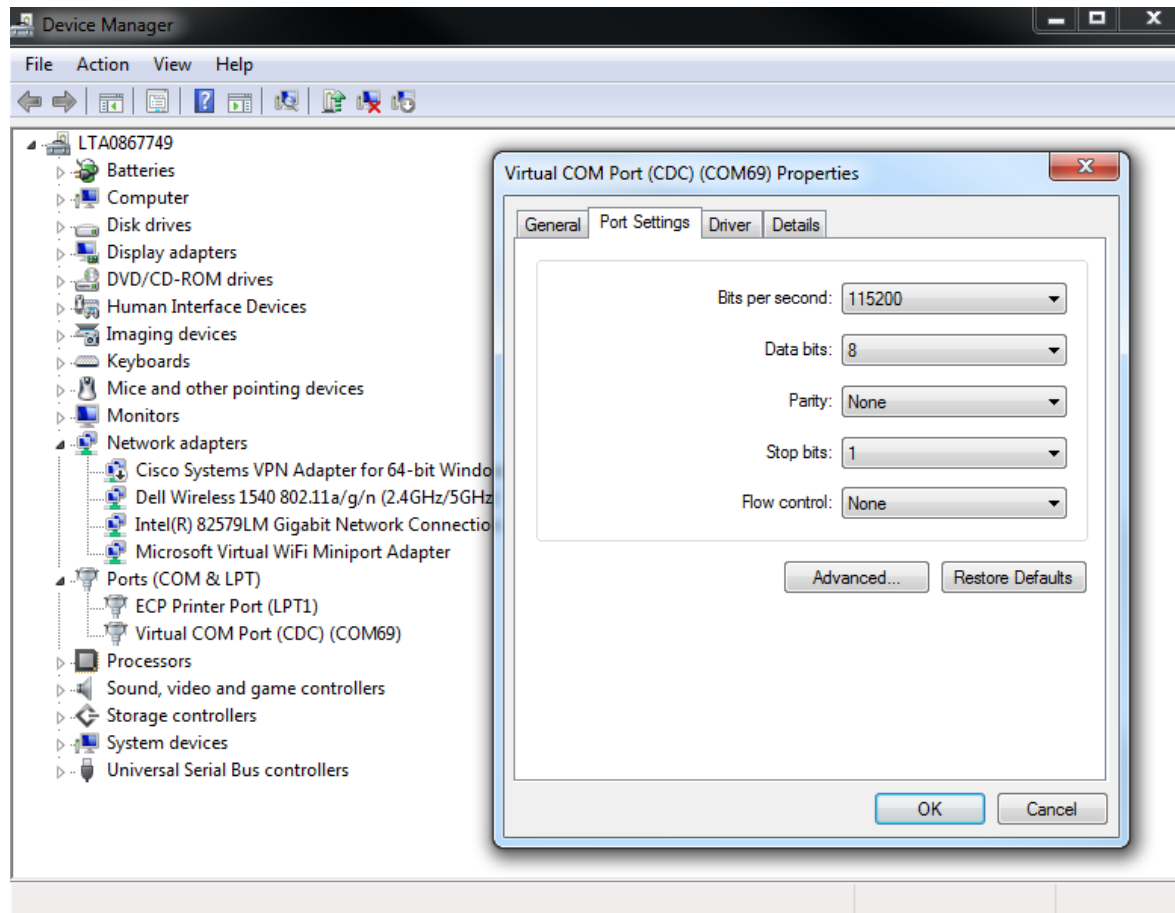


6/18/2013

COM Port Settings

NOTE: your COM port # most likely will be different

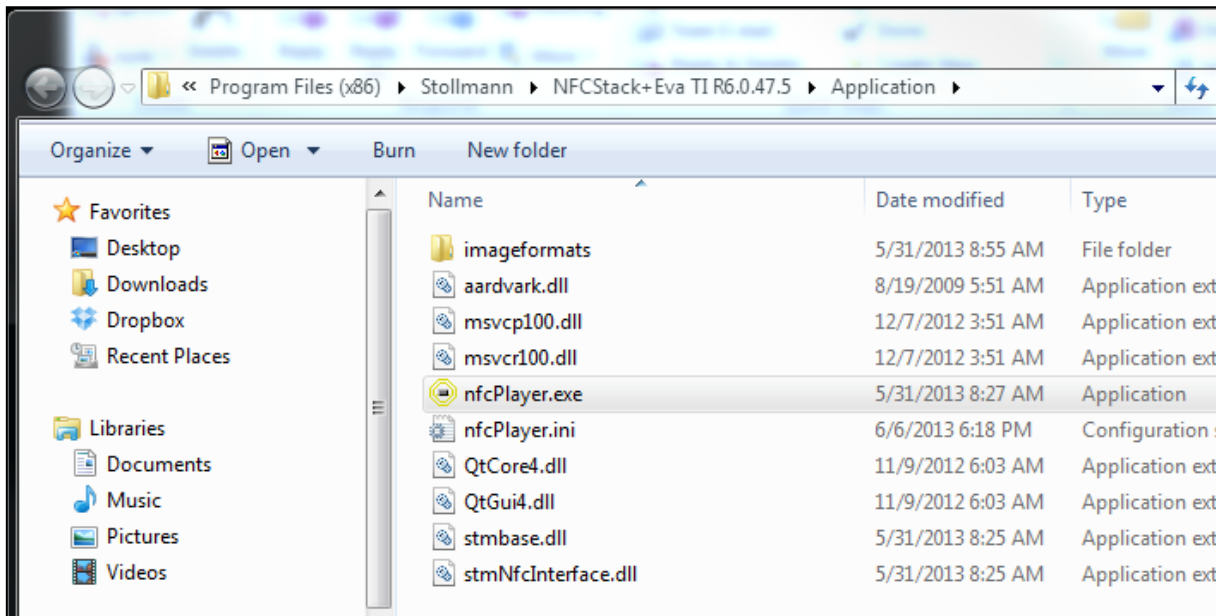
- Adjust Virtual COM Port settings for 115200, 8N1



6/18/2013

Execute NFCPLayer (PC GUI)

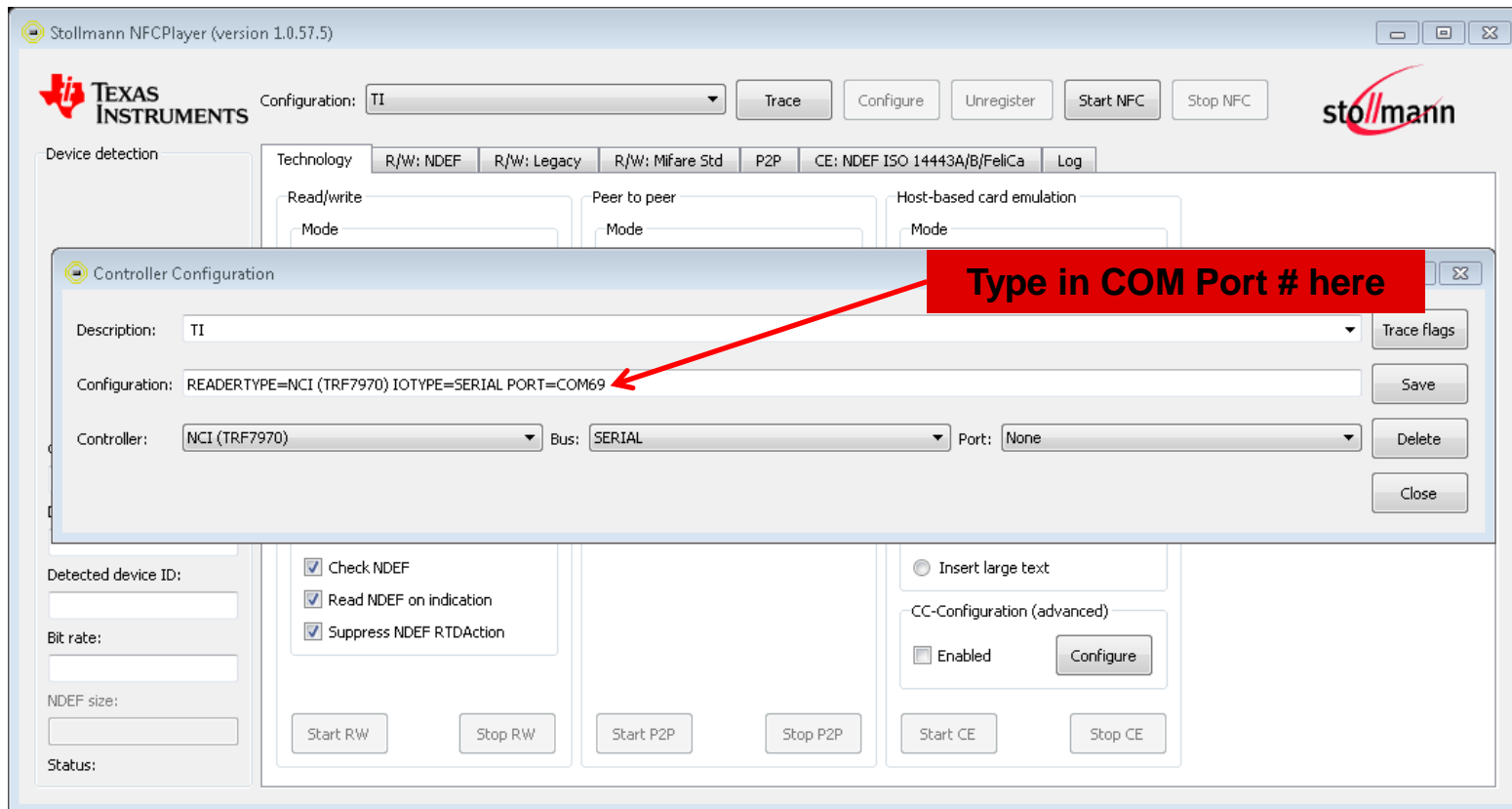
- Choose to execute NFCPlayer from Programs in PC start menu under Stollmann NFCStack+Eva TI R6.0.47.5 Folder by double clicking the nfcPlayer.exe file shown below.



- The GUI will open, see next slide.

PC GUI (configuration)

- After GUI opens, press the Configure button to pop up sub-screen and type in the number that the COM port enumerated to.
- Save this sub window, which will close it.

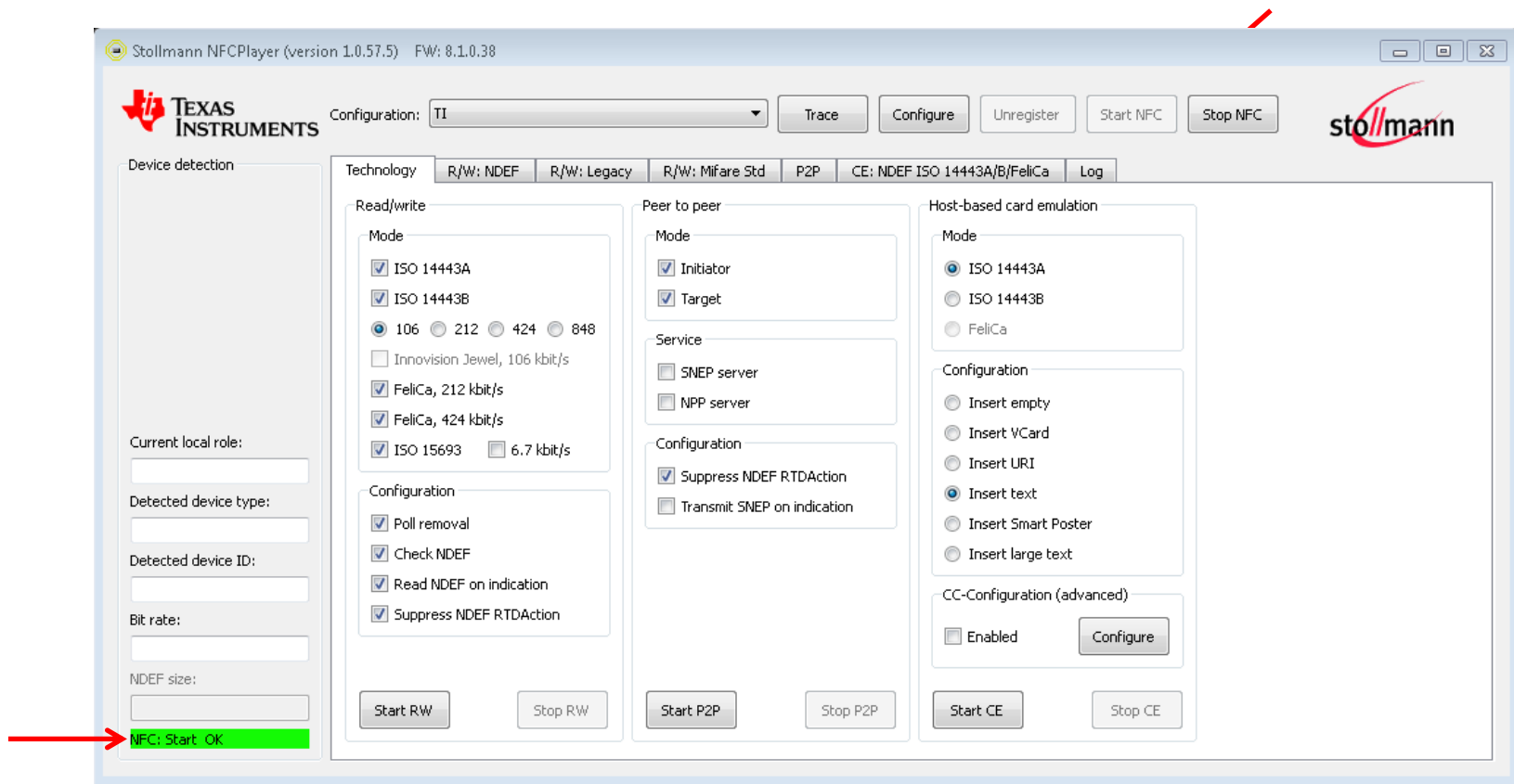


6/18/2013

PC GUI

(First Step after Config of the COM port)

- Press Start NFC Button, if correctly setup, then NFC: Start OK will appear in lower left hand window of the GUI.



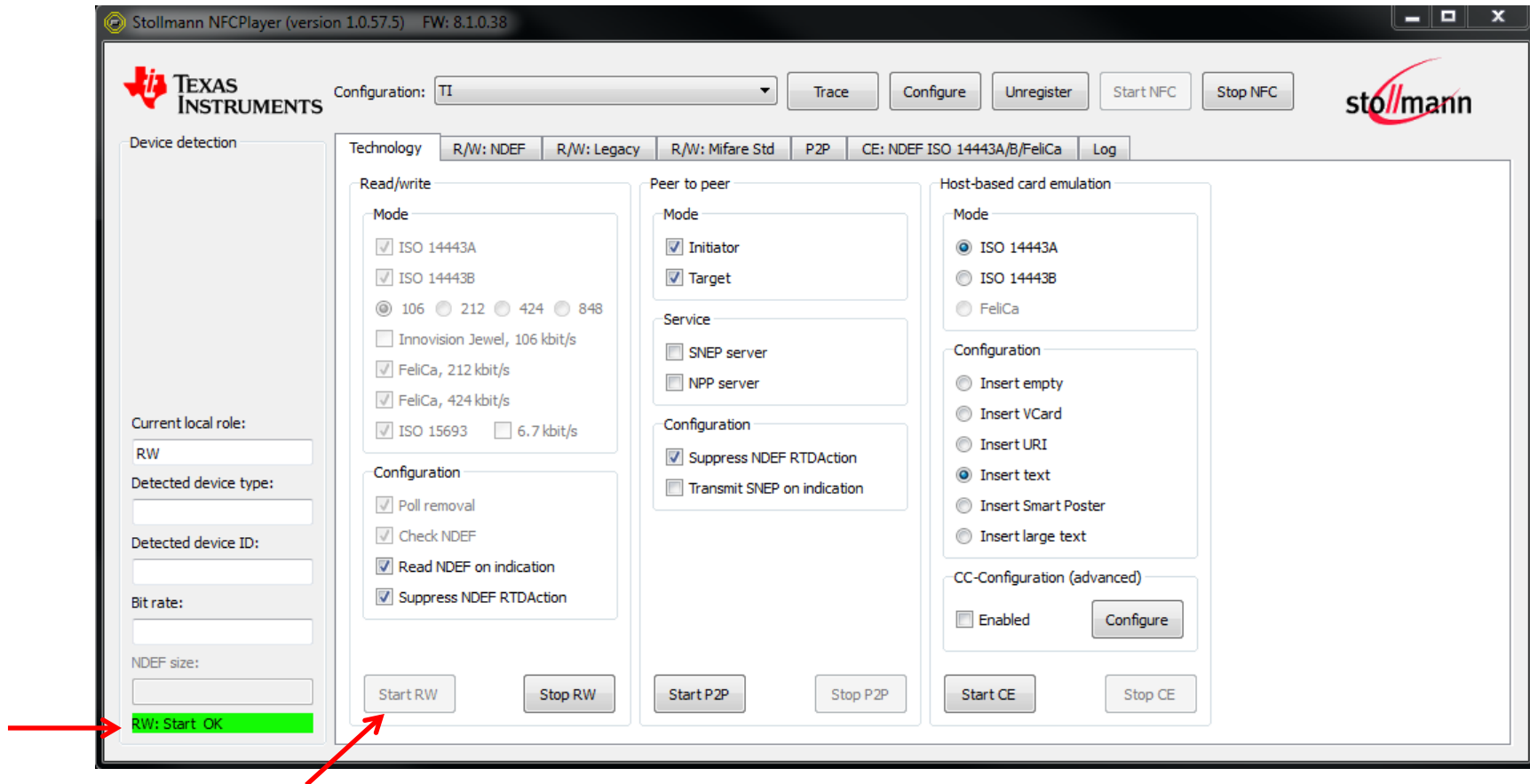
6/18/2013

NFC/RFID READER/WRITER MODE

6/18/2013

Reader/Writer Mode

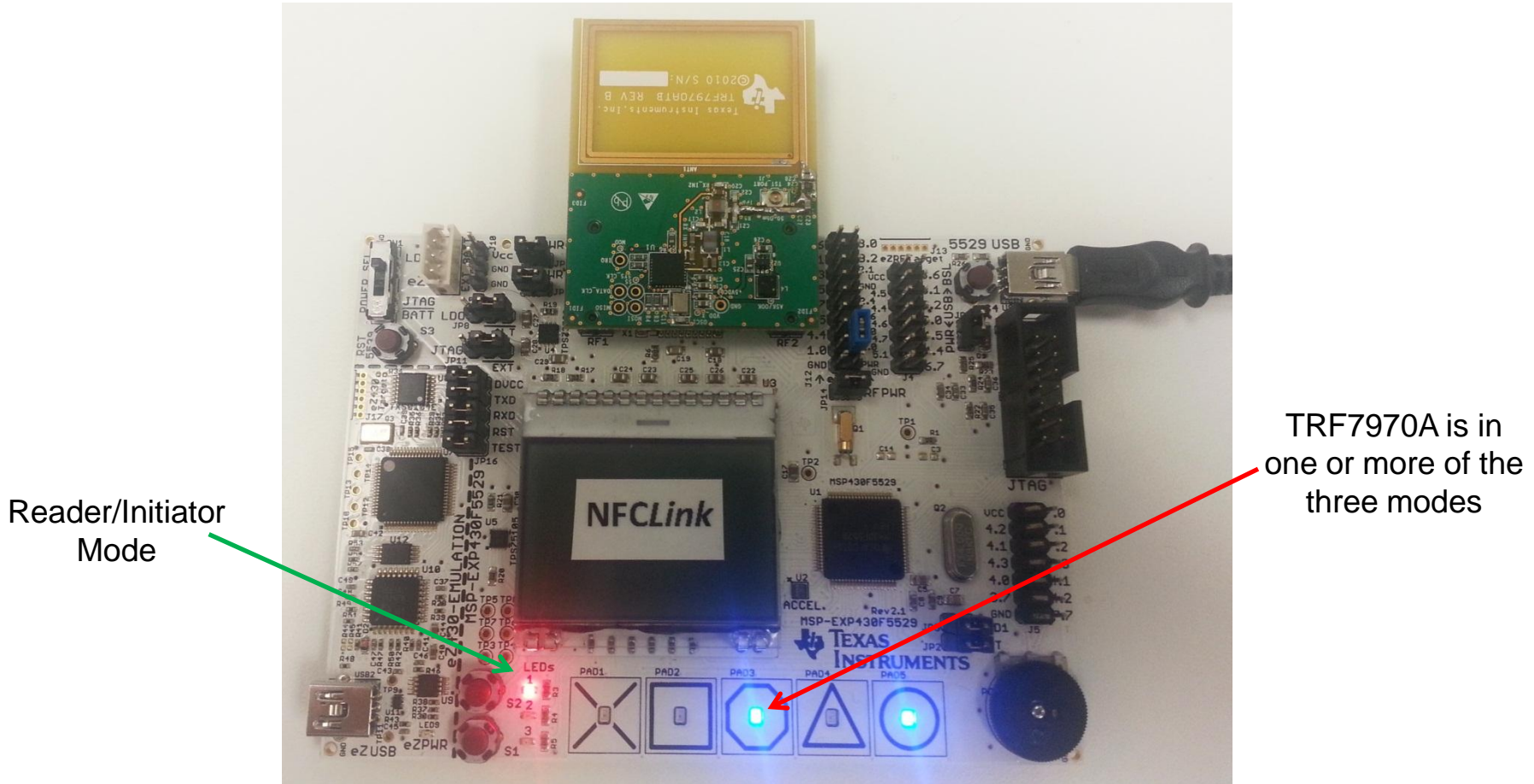
- For those wishing to use Reader/Writer Mode, press the Start RW button.
 - (RW: Start OK will appear in lower left hand window of the GUI)



6/18/2013

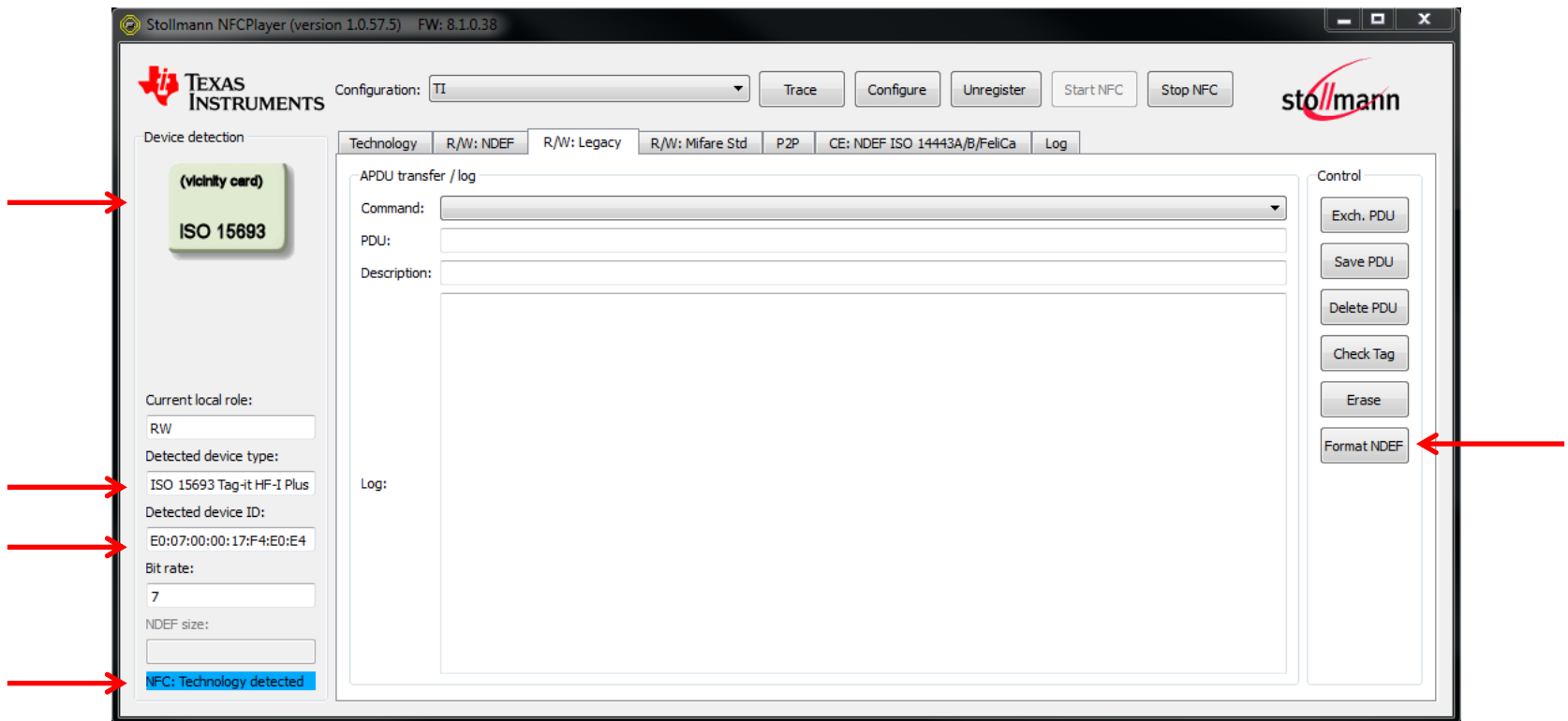
Hardware LED status indicators (Reader/Writer Mode)

- When hardware is in reader/writer mode, LED1 will be flashing and then go solid when an NFC/RFID tag is presented.



Reader/Writer Mode NFC-V tag

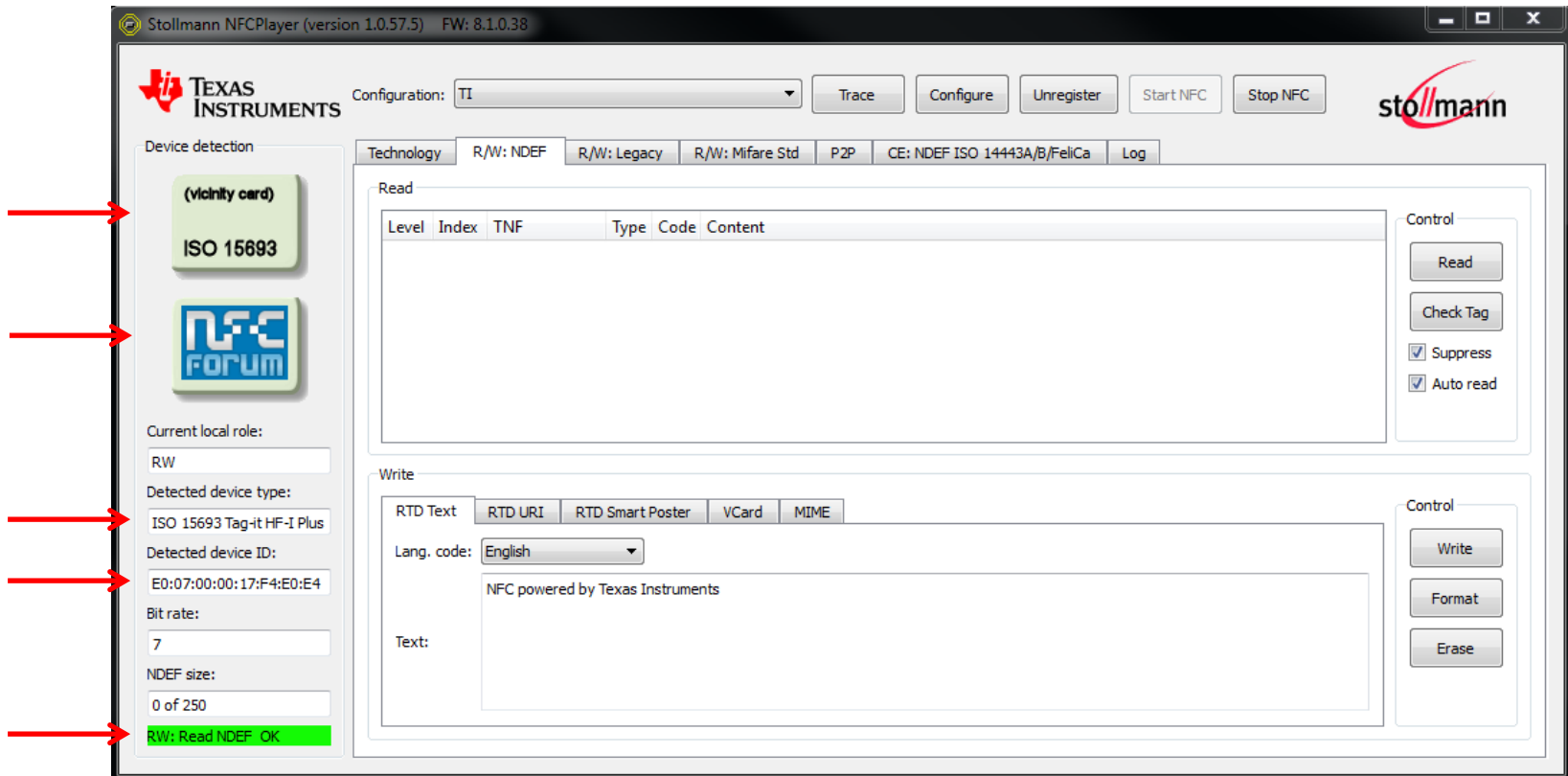
- After pressing the R/W button, present an NFC tag (formatted or not) – in this example we are presenting an unformatted NFC-V (ISO15693, TI HF-I) tag.
- Note the tab automatically flips to R/W: Legacy and the type of card is displayed graphically along with the Unique ID (in this case: E007000017F4E0E4)
- User can press the Format NDEF button to make the tag NFC-V type. (this puts Capability Container with empty NDEF in the user memory.)



6/18/2013

Reader/Writer Mode NFC-V tag (cont.)

- 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, below can be seen the results (formatted but empty tag)



6/18/2013

Reader/Writer Mode NFC-V tag (cont.)

- User can now enter data into the bottom 'Write' window and press the Write button to write NDEF message into the tag.
- When this occurs, screen will quickly flip to Write NDEF , then the user can 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

The screenshot shows the Stollmann NFCPlayer (version 1.0.57.5) FW: 8.1.0.38 interface. The 'Read' window is active, displaying the following NDEF message:

Level	Index	TNF	Type	Code	Content
0	0	RTDWellKnown	T	en	Welcome Texas Instruments Field Applications Engineers! NFC powered by Texas Instruments - NFCLink Rocks!!! NFC is a growth engine for Texas Instruments across all business units! We value your skills - t...

The 'Write' window is also visible, showing the 'RTD Text' tab and the 'Lang. code' set to 'English'. The 'Text' field contains the same message as the 'Read' window. The 'Control' buttons for 'Read', 'Check Tag', 'Write', 'Format', and 'Erase' are present. The 'Device detection' panel on the left shows the detected device as 'ISO 15693 Tag-it HF-I Plus' with ID 'E0:07:00:00:17:F4:E0:E4'. The 'RW: Read NDEF' status bar at the bottom indicates 'OK'.

6/18/2013

Reader/Writer Mode NFC Type 2 tag

- Here is Tag Type 2 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 www.ti.com/nfc web page.

The screenshot shows the Stollmann NFCPlayer (version 1.0.42.5) FW: 8.1.0.27 interface. The Texas Instruments logo is in the top left, and the Stollmann logo is in the top right. The Configuration dropdown is set to 'TI'. Buttons for Trace, Configure, Unregister, Start NFC, and Stop NFC are visible. The Technology tab is selected, showing R/W: NDEF, R/W: Legacy, R/W: Mifare Std, P2P, CE: NDEF ISO 14443A/B/FeliCa, CE: Embedded SE, and Log. The Read section displays a table of NDEF records:

Level	Index	TNF	Type	Code	Content
0	0	RTDWellKnown	Sp		
1	0	RTDWellKnown	T	en	Check this out - NFC powered by Texas Instruments!
1	1	RTDWellKnown	U		http://www.ti.com/nfc
1	2	RTDWellKnown	act		start application

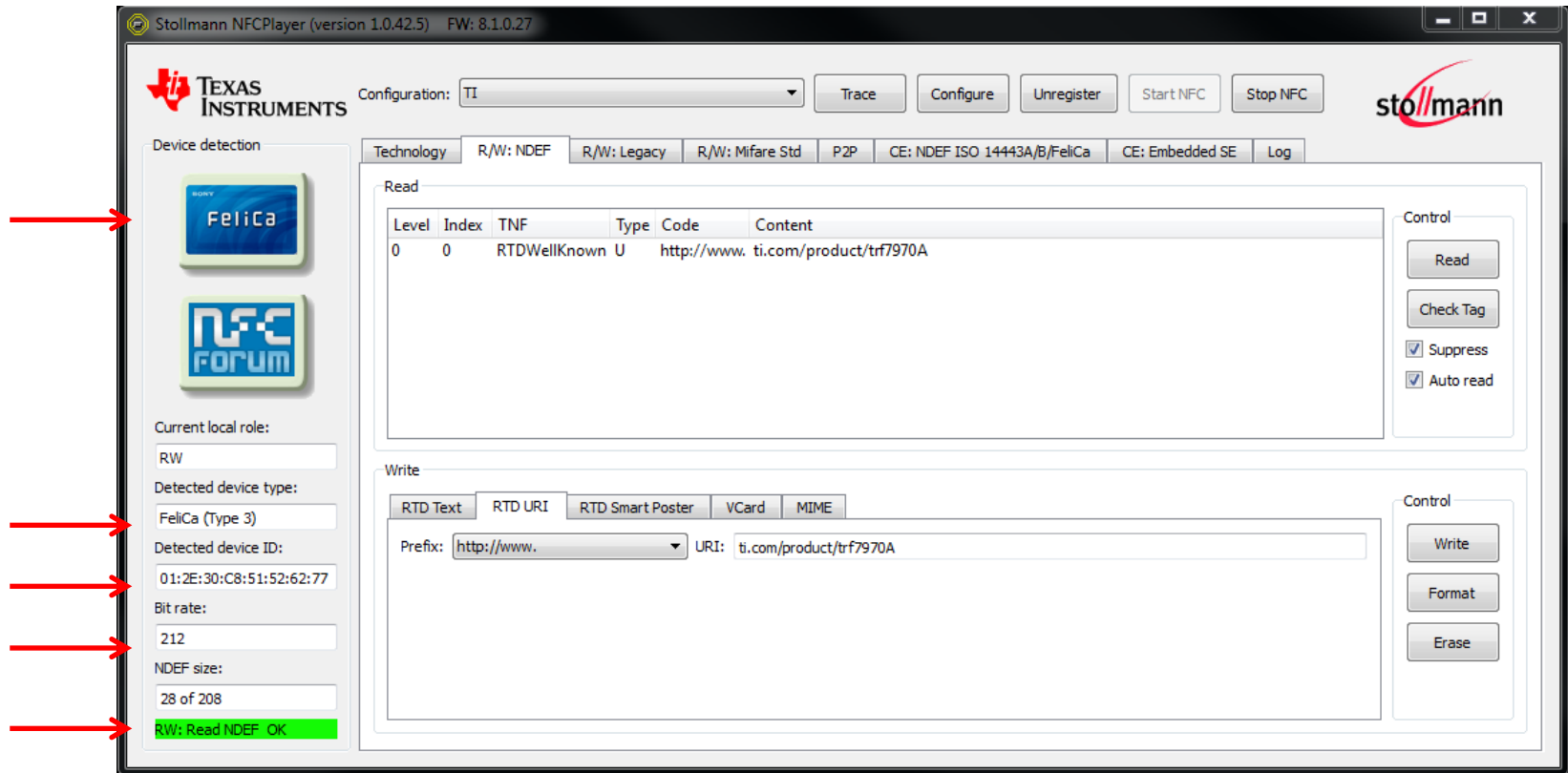
The Write section shows the RTD Smart Poster tab selected. The Lang. code is set to English, and the URI is <http://www.ti.com/nfc>. The Text field contains "Check this out - NFC powered by Texas Instruments!". The Control buttons for Read, Check Tag, Suppress, Auto read, Write, Format, and Erase are visible. The left sidebar shows device detection results for a Mifare UL tag, with red arrows pointing to the NXP Mifare UL and NFC Forum logos, and the detected device type, ID, bit rate, and NDEF size.

Current local role: RW
Detected device type: Mifare Ultralight (Type 2)
Detected device ID: 04:D0:9E:1A:43:28:80
Bit rate: 106
NDEF size: 84 of 137
RW: Read NDEF OK

6/18/2013

Reader/Writer Mode NFC Type 3 tag

- Here is Tag Type 3 which has been formatted with the tool and then programmed as RTD URI.



6/18/2013

Reader/Writer Mode NFC Type 4A tag

- Here is Tag Type 4A which has been formatted with the tool and then programmed as VCard.

The screenshot shows the Stollmann NFCPlayer (version 1.0.42.5) FW: 8.1.0.27 interface. The left sidebar displays the 'Device detection' section with the 'Desfire' tag icon highlighted by a red arrow. Below it, the 'Current local role' is set to 'RW'. The 'Detected device type' is 'Mifare DESFire (Type 4)', and the 'Detected device ID' is '04:58:17:3A:EF:22:80'. The 'Bit rate' is '848', and the 'NDEF size' is '339 of 2046'. A green status bar at the bottom left indicates 'RW: Read NDEF OK'. The main area shows the 'Read' section with a table of NDEF records. The first record is a MIME text/x-vCard with the following content:

Level	Index	TNF	Type	Code	Content
0	0	MIME	text/x-vCard		BEGIN:VCARD VERSION:3.0 N:Wyatt,Josh;;Applications Manager FN:Josh Wyatt ORG:Texas Instruments TITLE:NFC Expert URL:www.ti.com EMAIL;TYPE=INTERNET:josh.wyatt@ti.com TEL;TYPE=voice,work,pref:214-567-5124 TEL;TYPE=cell:214-567-5124

The 'Write' section is also visible, showing fields for 'RTD Text', 'RTD URI', 'RTD Smart Poster', 'VCard', and 'MIME'. The 'VCard' tab is selected, and the fields are populated with the contact information for Josh Wyatt at Texas Instruments. The 'Control' section on the right includes buttons for 'Read', 'Check Tag', 'Write', 'Format', and 'Erase', along with checkboxes for 'Suppress' and 'Auto read'.

6/18/2013

Reader/Writer Mode NFC Type 4B tag

- Here is Tag Type 4B (RF430CL330H) which has been formatted with the tool and then programmed for NFC Forum Bluetooth Connection Handover.

The screenshot shows the Stollmann NFCPlayer (version 1.0.42.5) interface. The left sidebar displays the Texas Instruments logo and the detected device: ISO 14443-B (Type 4). The main area shows the 'Read' and 'Write' sections. The 'Read' section displays a table of data:

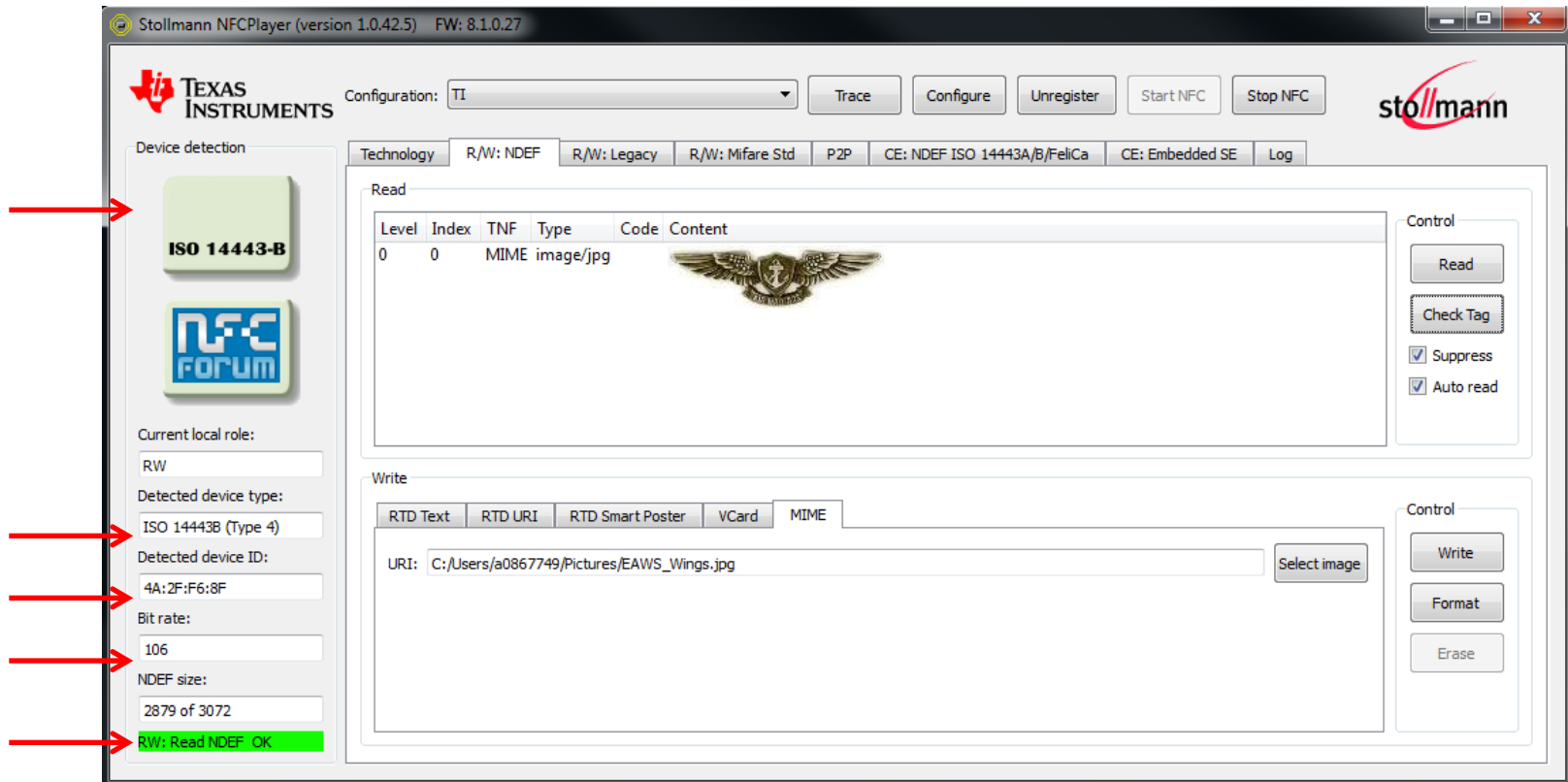
Level	Index	TNF	Type	Code	Content
0	0	MIME	application/vnd.bluetooth.ep.oob	21 00 06 05 04 03 02 01 0D 09 48 65 61 64 53 65 74 20 4E 61 6D 65 04 0D 04 04 20 05 03 1E 11 0B 11	

The 'Write' section shows the 'RTD Text' tab selected, with the language code set to 'English' and the text 'NFC powered by Texas Instruments'. The 'Control' buttons for 'Read', 'Check Tag', 'Write', 'Format', and 'Erase' are visible on the right. Red arrows point to the 'ISO 14443-B' tag, the 'Detected device type', 'Detected device ID', 'Bit rate', 'NDEF size', and the 'RW: Read NDEF OK' status bar.

6/18/2013

Reader/Writer Mode NFC Type 4B tag (cont.)

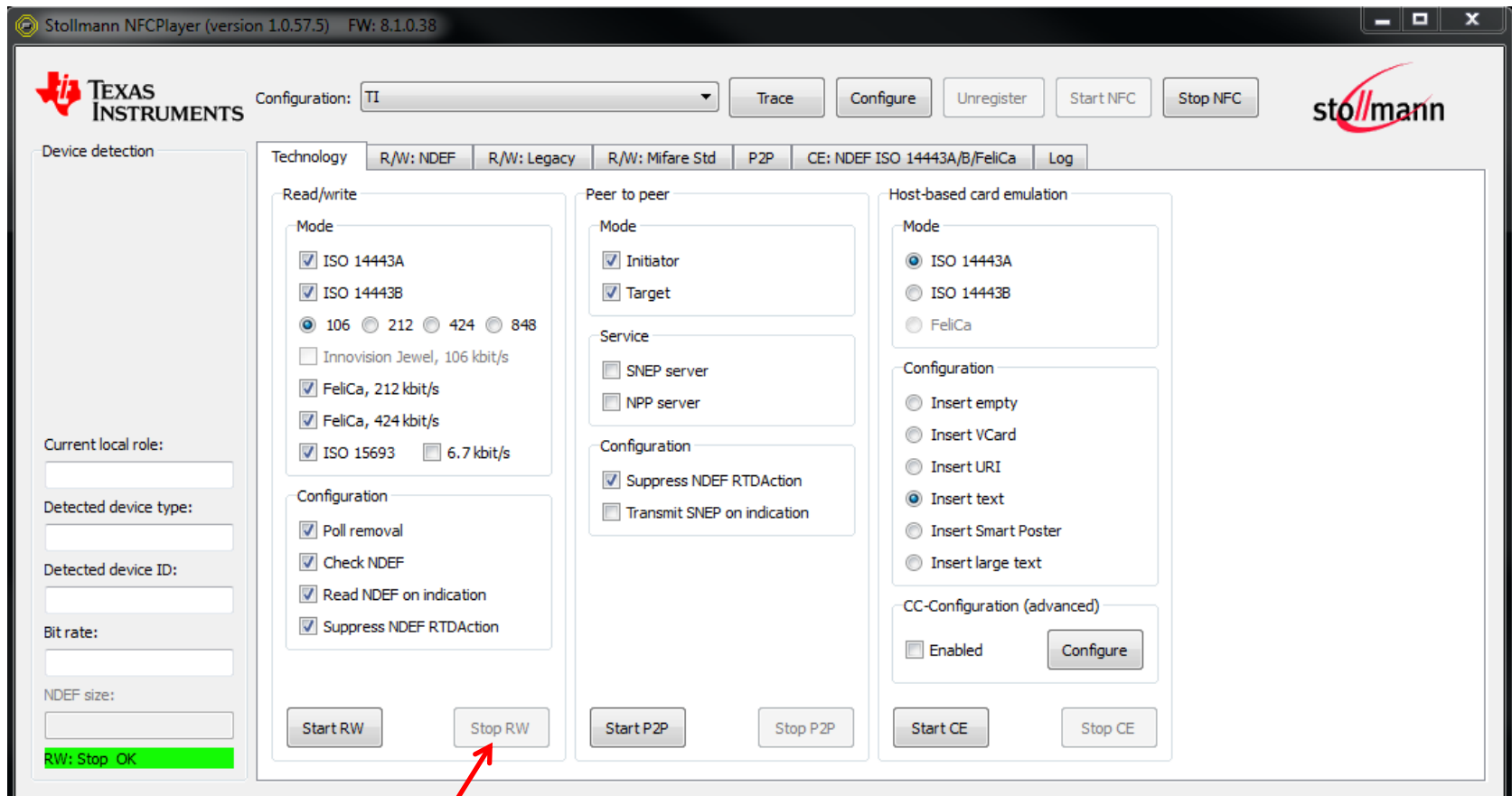
- Here is Tag Type 4B (RF430CL330H) which has been formatted with the tool and then programmed with an image (MIME).



6/18/2013

Exiting or Stopping the Reader/Writer Mode

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



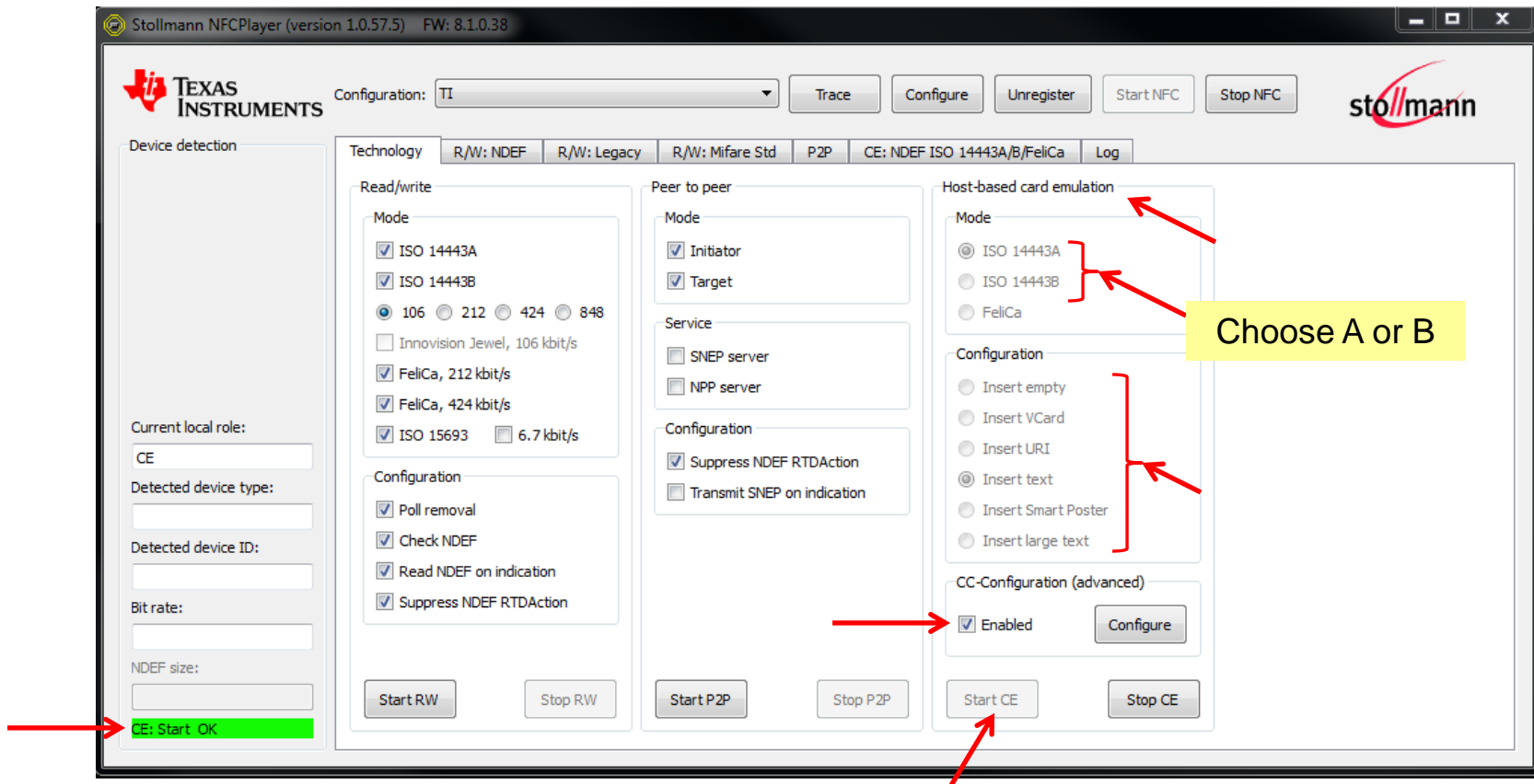
6/18/2013

NFC/RFID CARD EMULATION MODE

6/18/2013

Card Emulation Mode

- For those wishing to use Card Emulation Mode, under the Host-based card emulation, choose the Mode and the Configuration, then press the Start CE button.
 - (CE: Start OK will appear in lower left hand window of the GUI)

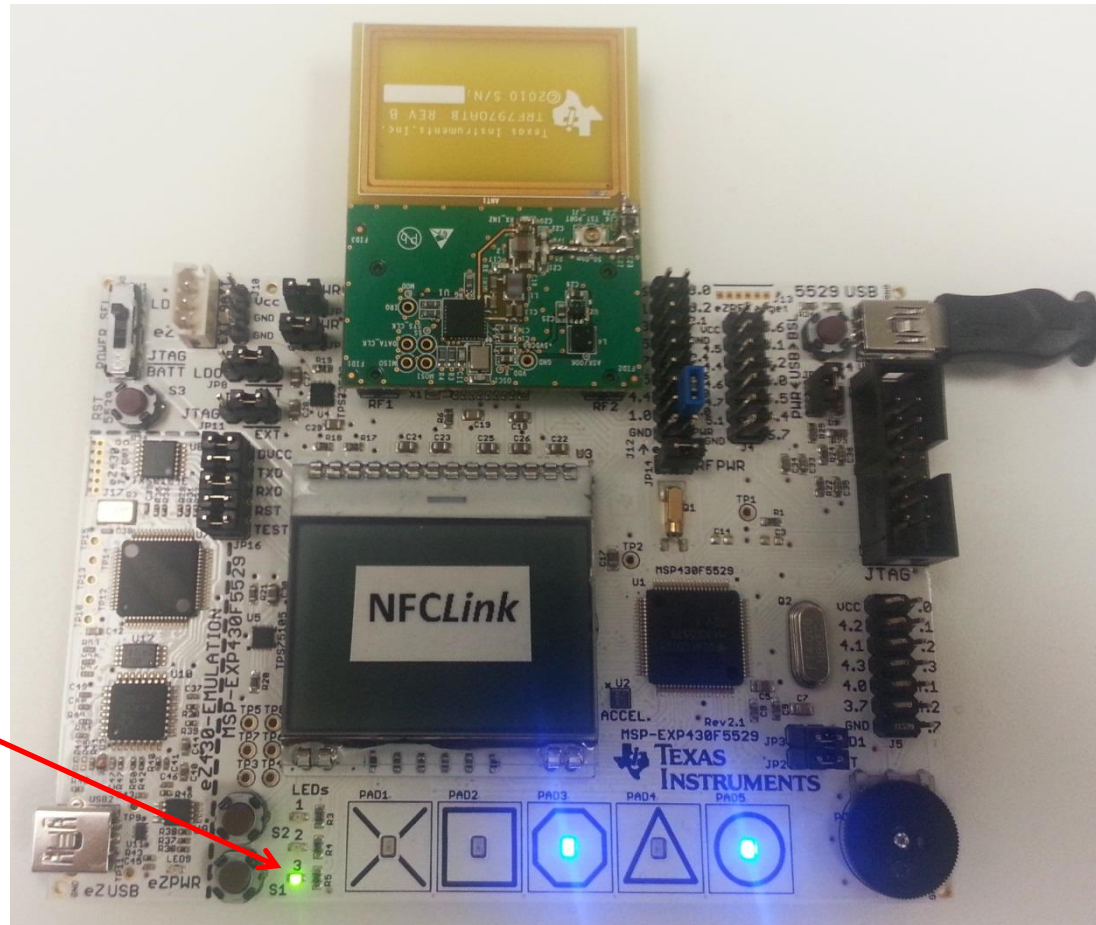


6/18/2013

Hardware LED status indicators (Card Emulation Mode)

- When hardware is in card emulation mode, LED3 will be solid.

Card Emulation
Mode

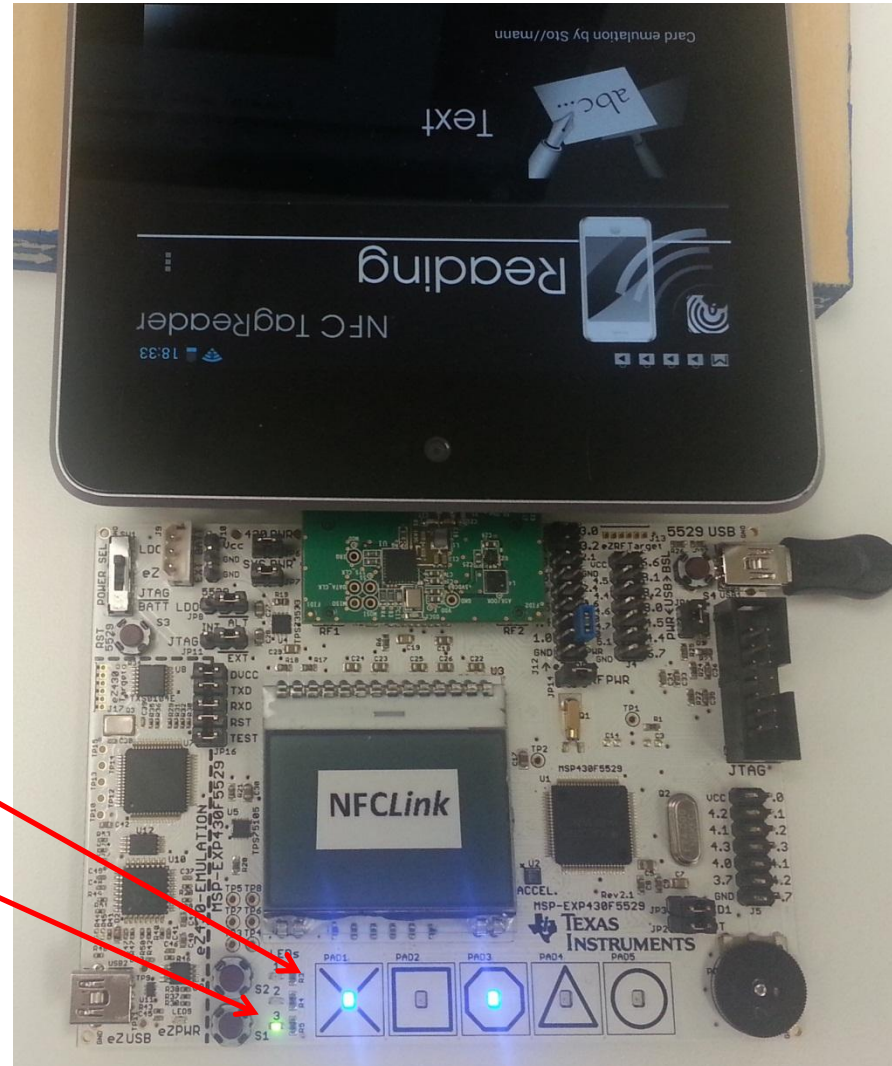


Hardware LED status indicators (Card Emulation Mode, cont.)

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

HF Field Present

Card Emulation
Mode



Card Emulation Mode (cont.)

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



VCard



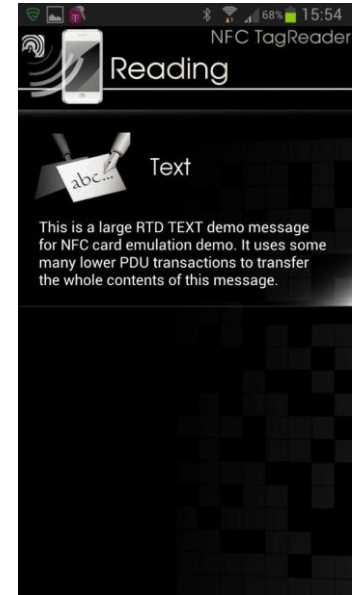
URI



text



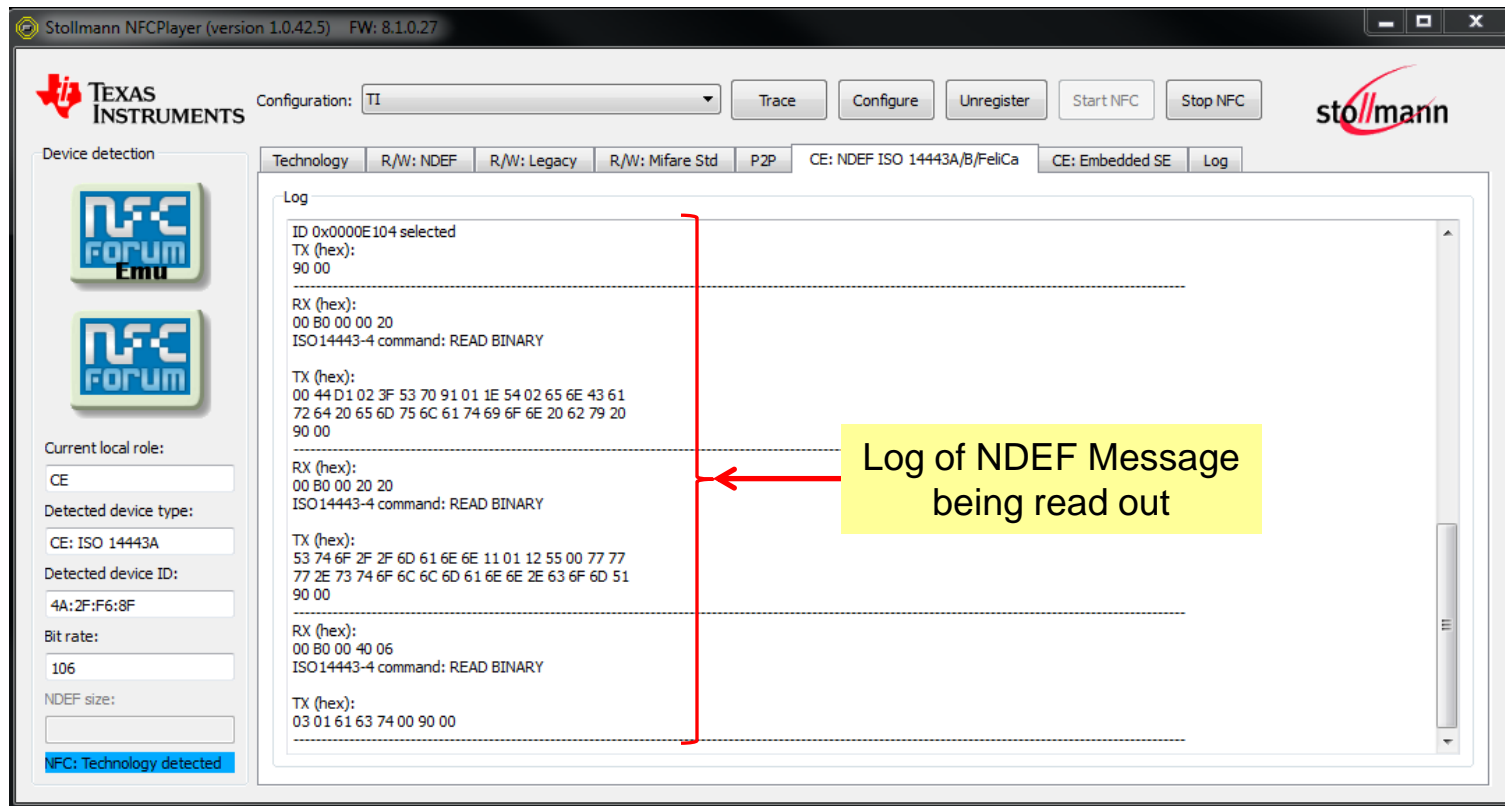
Smart Poster



Large text

Card Emulation Mode (cont.)

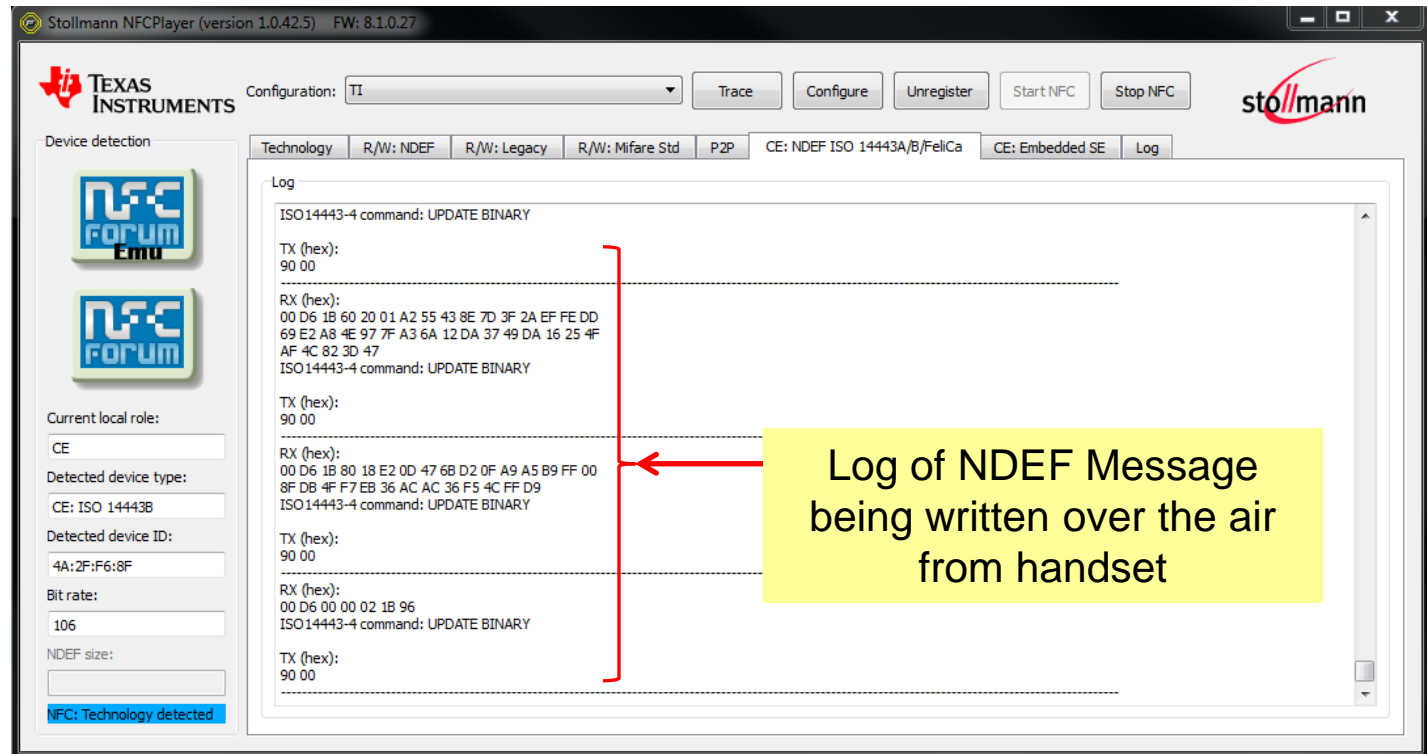
- When an NFC handset (or other NFC enabled reader/writer) is presented, the hardware will be read out just like a passive NFC/RFID tag.
- Here we show the GUI screen, as it is showing the packet activity between the NFC handset (or other reader/writer) and the hardware.



6/18/2013

Card Emulation Mode (cont.)

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



6/18/2013

Card Emulation Mode (cont.)

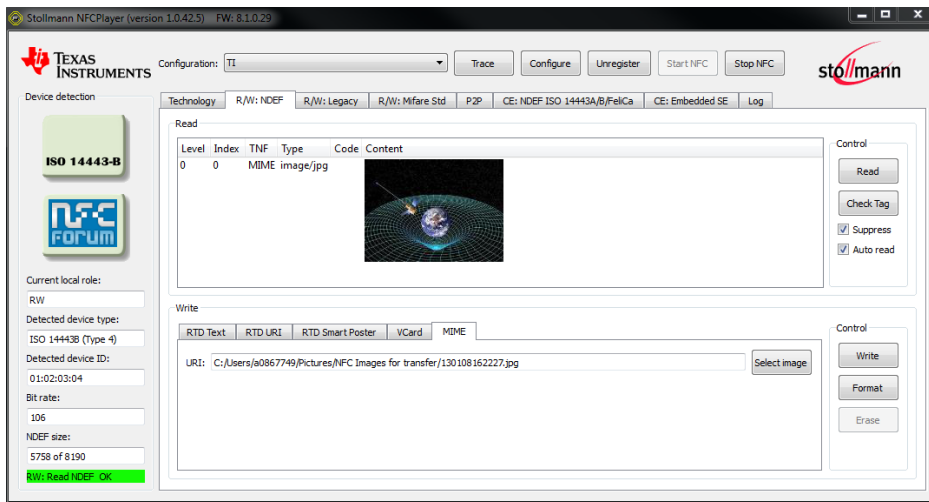
- Then of course the user can read back the image (or other data) from the hardware, using the application on the handset as well. (Master Yoda shown below in ~7kB form)



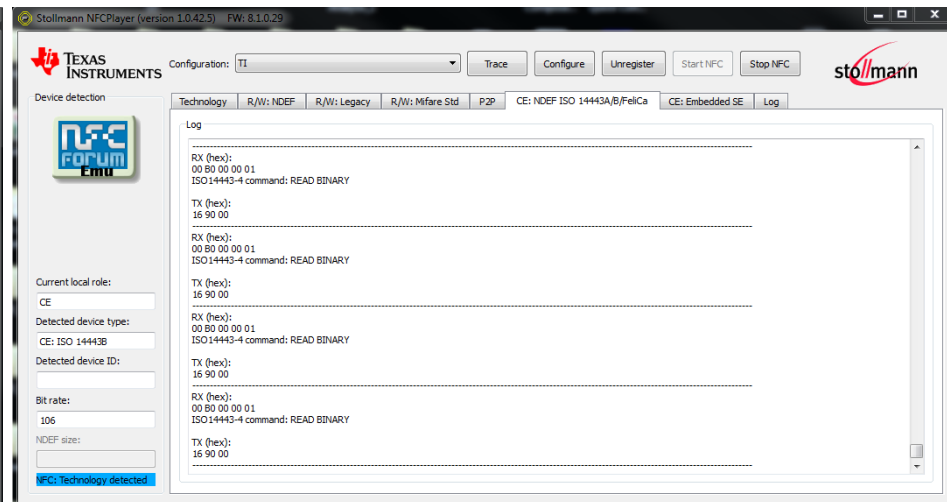
6/18/2013

R/W & Card Emulation Modes used in conjunction with one another

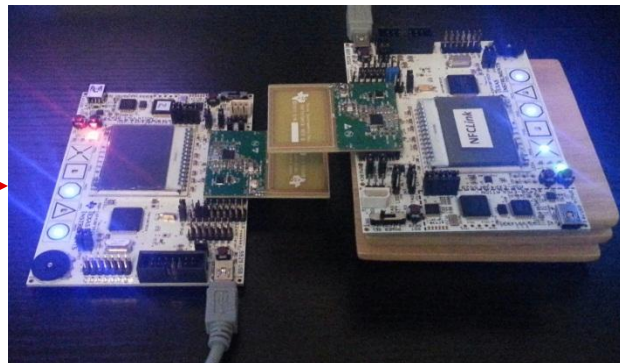
- User can also set up two of the hardware sets and run two instances of the GUI on the same PC (with one as reader/writer and the other in card emulation mode) then read/write data without using an NFC handset. (Here we have written ~5.7kB image and have read it back)



Hardware set in R/W mode



Hardware set in CE mode



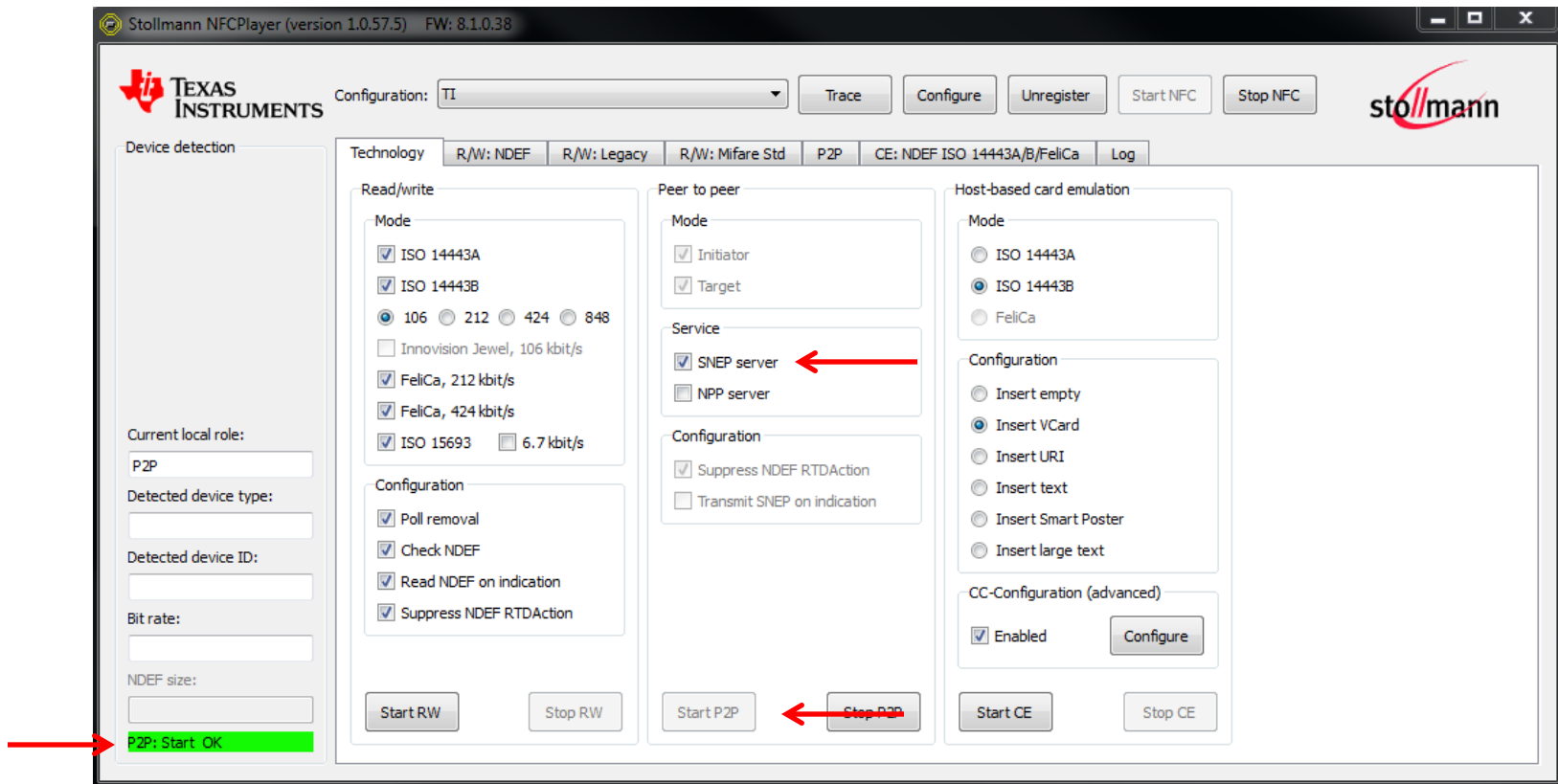
6/18/2013

NFC/RFID P2P MODE

6/18/2013

Peer to Peer Mode (P2P)

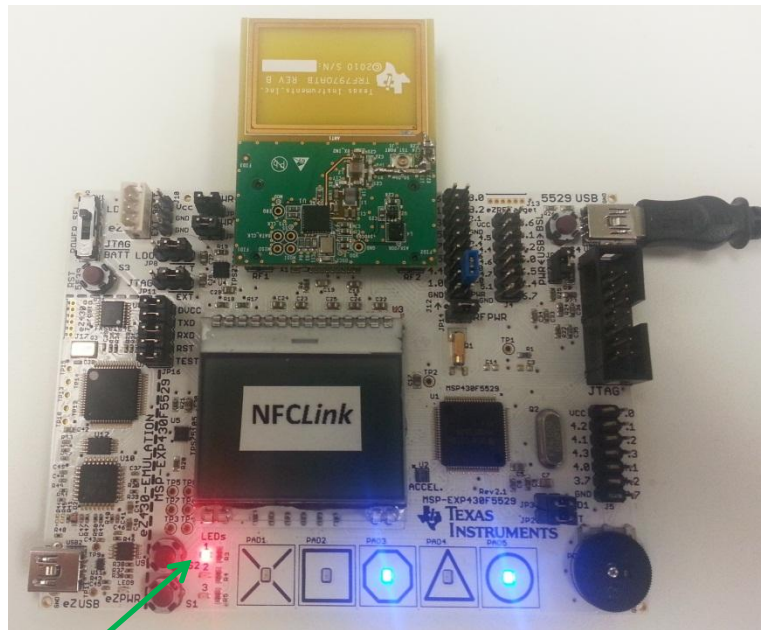
- For those wishing to use P2P Mode, under the Peer to Peer column, choose the SNEP box, then press the Start P2P button.
 - (P2P: Start OK will appear in lower left hand window of the GUI)



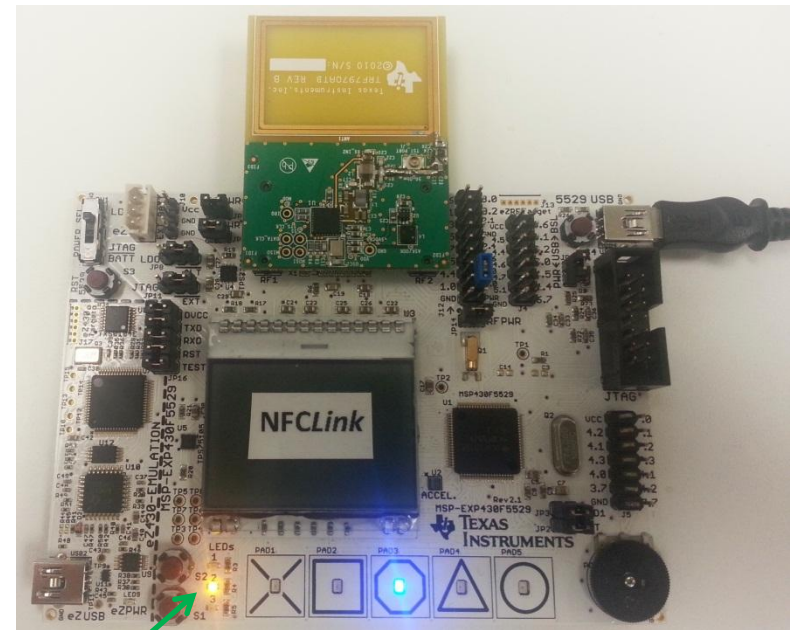
6/18/2013

Hardware LED status indicators (P2P Mode)

- When hardware is in P2P mode, LED1 & LED 2 will alternately be flashing.
- When NFC device is presented, the mode which is being used by the hardware (initiator or target) will go solid.
- After transfer is complete, the LEDs will go back to alternately flashing



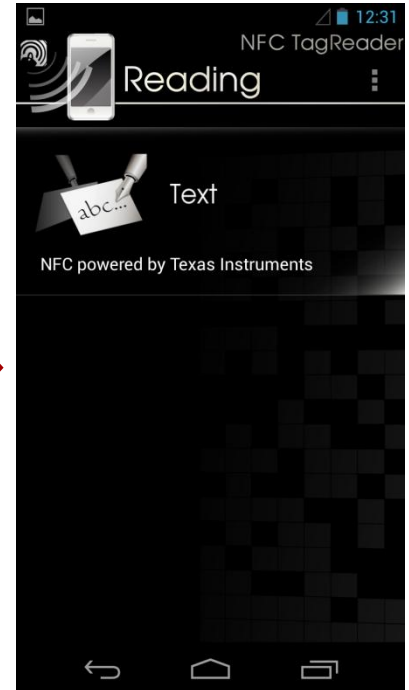
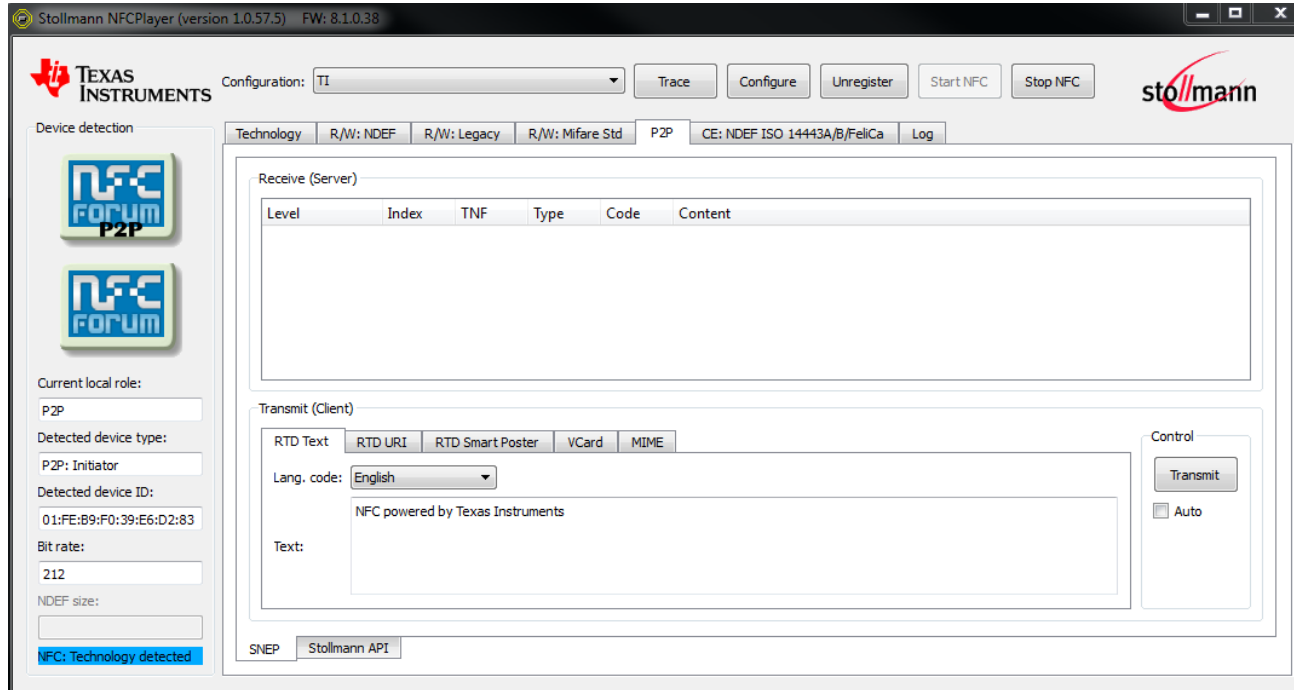
Reader/Initiator
Mode



Target
Mode

Peer to Peer Mode (P2P) (cont.)

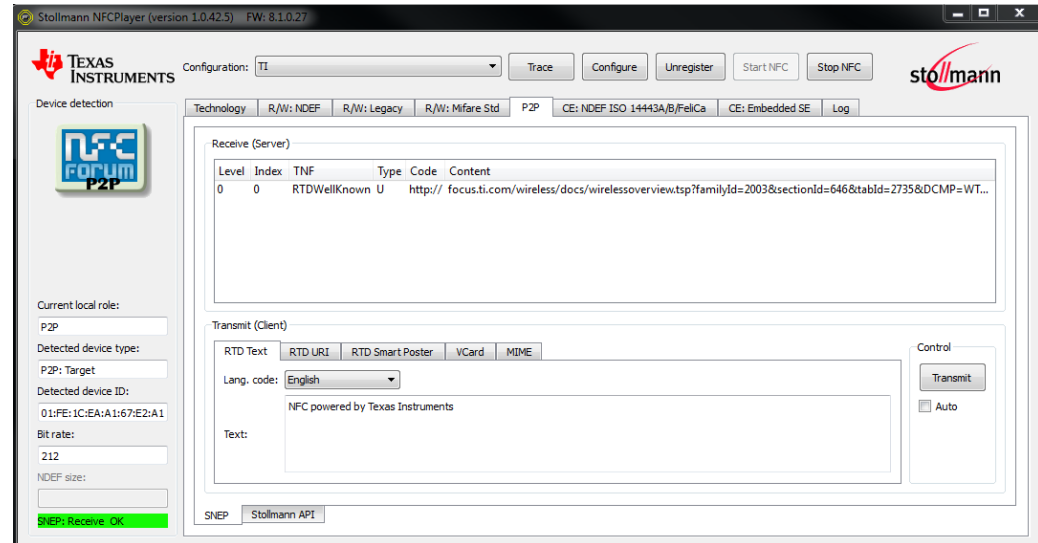
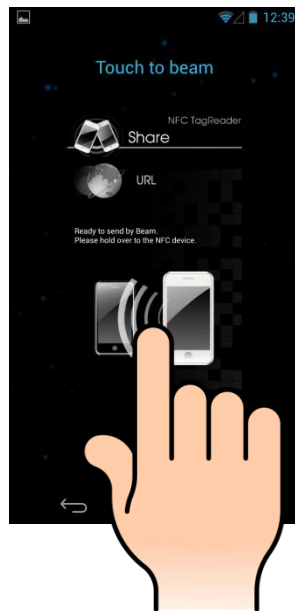
- When NFC Enabled device is presented (no app open in this case), the GUI screen flips over to P2P mode.
- Message can now be sent from GUI to the NFC enabled device using the Transmit button. This will open either the native app on the phone or a default one (NFC TagReader (from KDDI shown here))



6/18/2013

Peer to Peer Mode (P2P) (cont.)

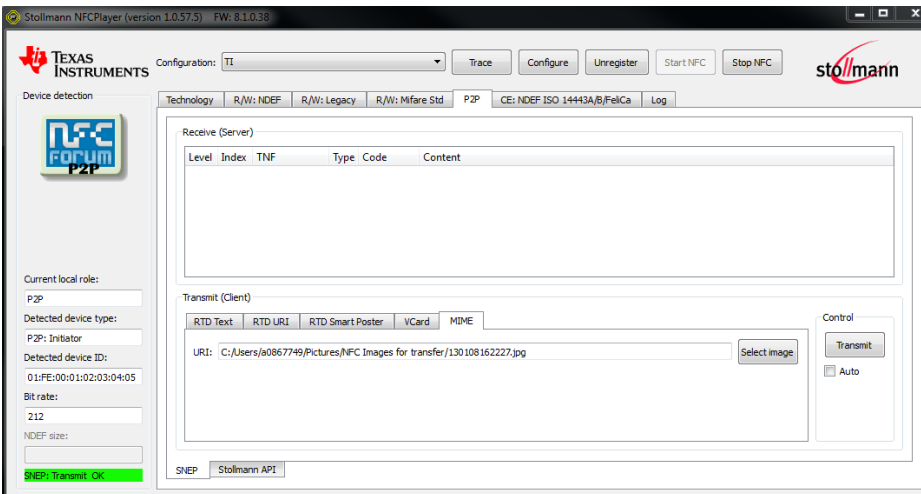
- When NFC Enabled device is with an application open for doing P2P, the GUI screen flips over to P2P mode as before.
- Message can now be sent from the NFC enabled device to the GUI.
- Here we are “Beaming” a URL to a TI website, other message types can also be sent over too.
 - Alternative Radio Handovers, text content, SmartPoster, Phone Numbers, Applications, Images, etc.



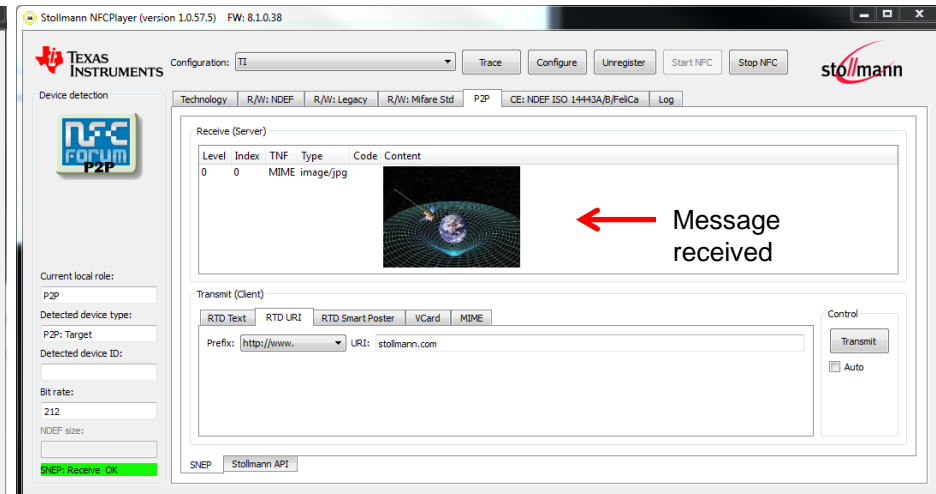
6/18/2013

Peer to Peer between two NFCLink hardware sets

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



Hardware set in P2P mode



Hardware set in P2P mode



6/18/2013

Peer to Peer between two NFCLink hardware sets (cont.)

The screenshot shows the Stollmann NFCPlayer (version 1.0.42.5) interface. The 'Receive (Server)' tab is active, displaying a table of received messages. A red arrow points to the table with the text 'Multiple Types of P2P messages received'.

Level	Index	TNF	Type	Code	Content
0	0	RTDWellKnown	U	http://www. stollmann.com	
0	0	RTDWellKnown	U	http://www. ti.com/nfc	
0	0	RTDWellKnown	Sp		
1	0	RTDWellKnown	T	en	Check this out - NFC powered by Texas Instruments!
1	1	RTDWellKnown	U	http://www. ti.com	
1	2	RTDWellKnown	act		start application

Transmit (Client) section shows fields for RTD Text, RTD URI, RTD Smart Poster, VCard, MIME, Lang. code (English), and a text area containing 'NFC powered by Stollmann'. A 'Transmit' button and an 'Auto' checkbox are also visible.

6/18/2013

Advanced Features with P2P

The screenshot displays the Stollmann NFCPlayer (version 1.0.57.5) FW: 8.1.0.38 interface. The main window shows configuration options for TI technology, with tabs for R/W: NDEF, R/W: Legacy, R/W: Mifare Std, and P2P. The P2P tab is selected, showing CE: NDEF ISO 14443A/B/FeliCa and a Log button. The interface is divided into two main sections: Connection 1 (SAP = 15) and Connection 2 (SAP = 14). Each connection section includes buttons for Start listener, Stop listener, Connect, and Disconnect, along with status indicators (Listener ready, Connected) and fields for Max. window size, MIU, and TX/RX bytes. Red arrows point to the RX bytes field in Connection 1 (41600) and the TX bytes field in Connection 2 (28544). A sidebar on the left shows device detection results, including NFC Forum P2P and NFC Forum tags. The bottom of the window shows the current local role (P2P) and detected device type (P2P: Target) for Connection 1, and P2P: Initiator for Connection 2. The Stollmann logo is visible in the top right corner.

Stollmann NFCPlayer (version 1.0.57.5) FW: 8.1.0.38

Configuration: TI

Trace Configure Unregister Start NFC Stop NFC

Technology R/W: NDEF R/W: Legacy R/W: Mifare Std P2P CE: NDEF ISO 14443A/B/FeliCa Log

Device detection

NFC Forum P2P

NFC Forum

Current local role: P2P

Detected device type: P2P: Target

Detected device ID: 08:14:44:3A

Bit rate: 212

NDEF size:

P2P: Transmit OK

Connection 1 (SAP = 15)

Start listener Stop listener Listener ready Max. window size: 1 Transmit RX bytes: 41600

Connect Disconnect Connected MIU: 128 Continuous Randomize size Echo server

Connection 2 (SAP = 14)

Start listener Stop listener Listener ready Max. window size: 1 Transmit RX bytes: 0

Connect Disconnect Connected MIU: 128 Continuous Randomize size Echo server

TX bytes: 28544

SNEP Stollmann API

Start NFC Stop NFC

Stollmann

Connection 1 (SAP = 15)

Start listener Stop listener Listener ready Max. window size: 1 Transmit RX bytes: 0

Connect Disconnect Connected MIU: 128 Continuous Randomize size Echo server

TX bytes: 41600

Connection 2 (SAP = 14)

Start listener Stop listener Listener ready Max. window size: 1 Transmit RX bytes: 28544

Connect Disconnect Connected MIU: 128 Continuous Randomize size Echo server

TX bytes: 0

Current local role: P2P

Detected device type: P2P: Initiator

Detected device ID: 01:FE:00:01:02:03:04:05

Stollmann

NFCLINK FILE STRUCTURE

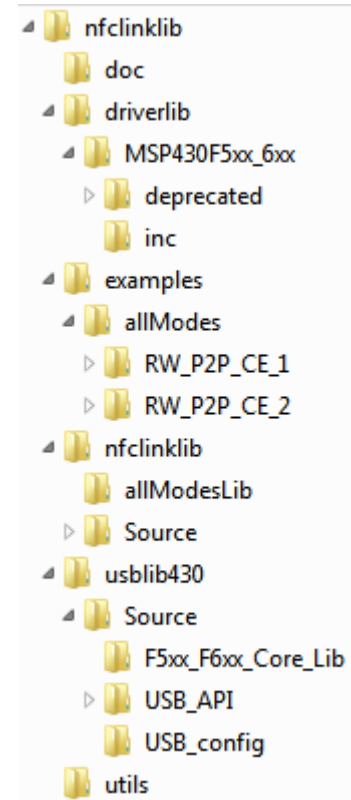
6/18/2013

What is being installed?

The NFCLink installer will include the following folders:

1. **doc** – User Guide.
2. **driverlib** – UART, GPIO and Timer drivers.
3. **examples/allModes** – Reader/Writer, Peer to Peer, and Card Emulation projects. (Only CCS for release)
 - a) **RW_P2P_CE_1** – USB (CDC) interface to the host.
 - b) **RW_P2P_CE_2** – UART Module interface to the host.
4. **nfclinklib** – NFC source + binary files used on the CCS projects.
 - a) **allModesLib** – Library File.
 - b) **Source** – TRF7970A drivers, and NFC Controller Header Files.
5. **usblib430** – USB drivers.
6. **utils** – NFC Player installer.

Note: The folder structure complies with MSP430Ware structure, and will be part of it on the upcoming August release.



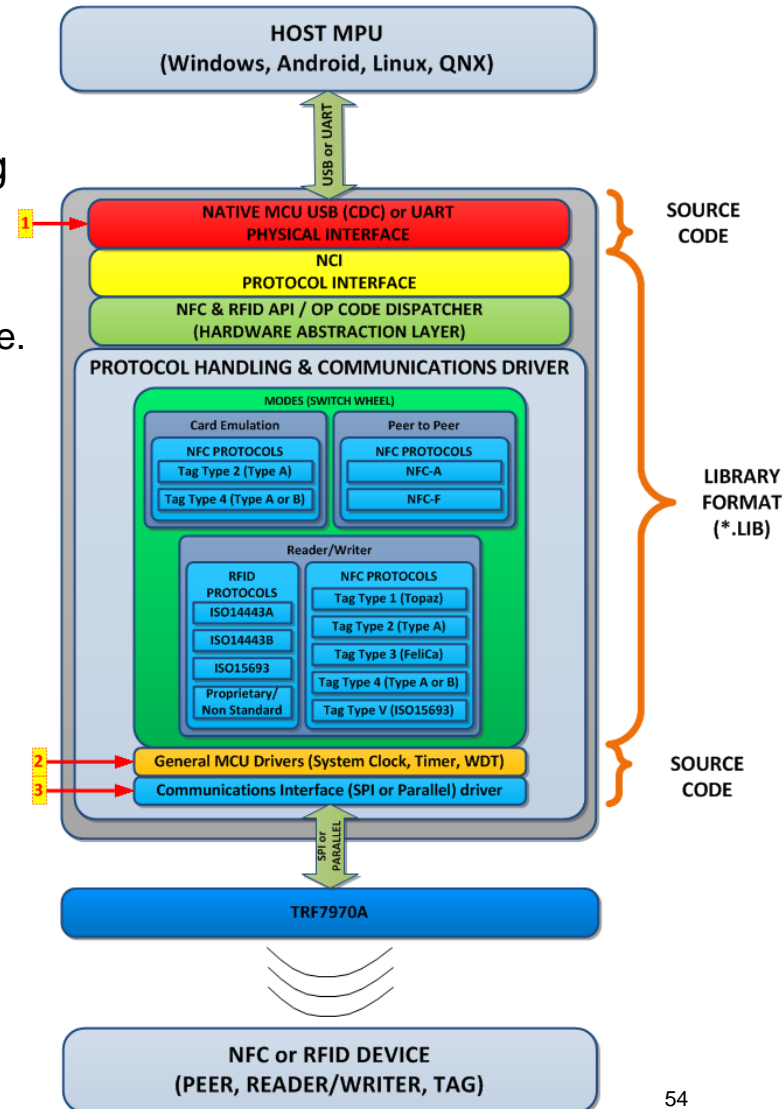
PORTING TO OTHER MSP430s

6/18/2013

MSP430s Porting Overview

The following layers will have to be modified when porting to other MSP430s:

1. **Host Interface** – Code examples will include USB / UART. Future releases will include SPI and I2C implementations – i.e. using an Aardvark™ I²C/SPI Host Adapter for testing.
2. **MSP430 Hardware** – Main application, MSP430 MCLK, Watch Dog Timer(WDT), GPIOs (for LEDs and debugging purposes), and one timer.
3. **TRF7970A transceiver Interface** – SPI w/ Slave Select module. Future releases will include Parallel.



Host Interface Modifications

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

1. **Host_Interface::Init()** –

Initialization for the module and RX ISR. For RW_P2P_CE1 the function initializes and configures the USB module.

2. **Host_Interface::Write()** –

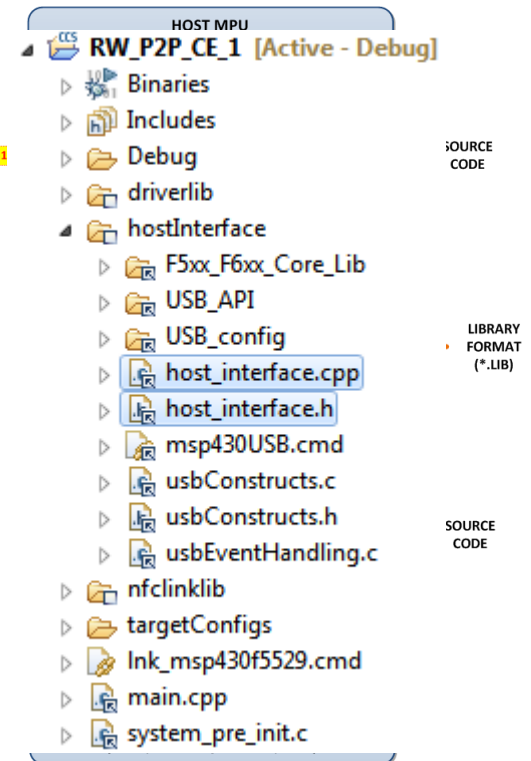
Function that transmits to the host. For RW_P2P_CE2 the function writes to the UART_TX buffer with len (length) bytes.

3. **The RX ISR for the host interface –**

```
BYTE USB CDC_handleDataReceived (BYTE intfNum) ← RW_P2P_CE1_Ex.  
__interrupt void USCI_A1_ISR(void)             ← RW_P2P_CE2_Ex.
```

Store incoming bytes into Host_Interface::recbuf.

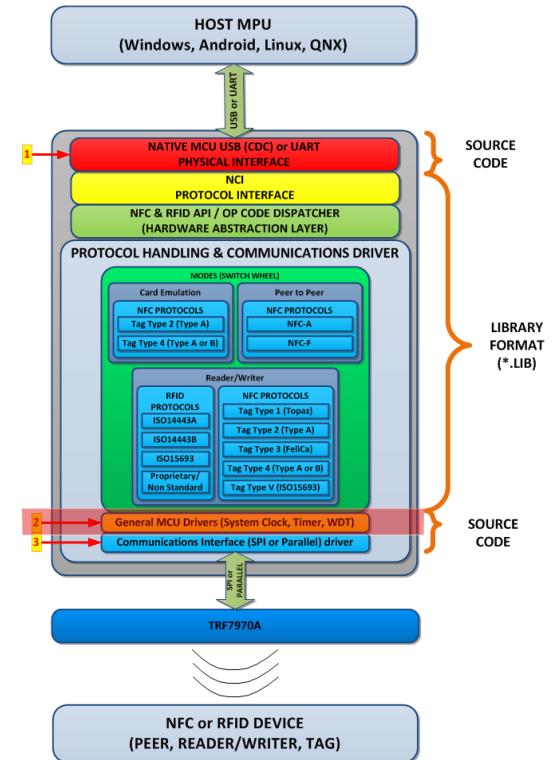
Note: Include necessary files for the Host_Interface module inside folder.



General MSP430 Modifications

There are **five** functions that **must** be reviewed :

1. **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 = 25MHz.
2. **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.
3. **HWTimer::Init()** –
Initialize Timer A using reference of ACLK (32.768kHz) running continuously.
4. **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. (i.e. For a 2MHz Clock → $TICKS_PER_MSEC = Ref. CLK / 1000 = 2 \times 10^6 / 1000 = 2000$)
5. **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.

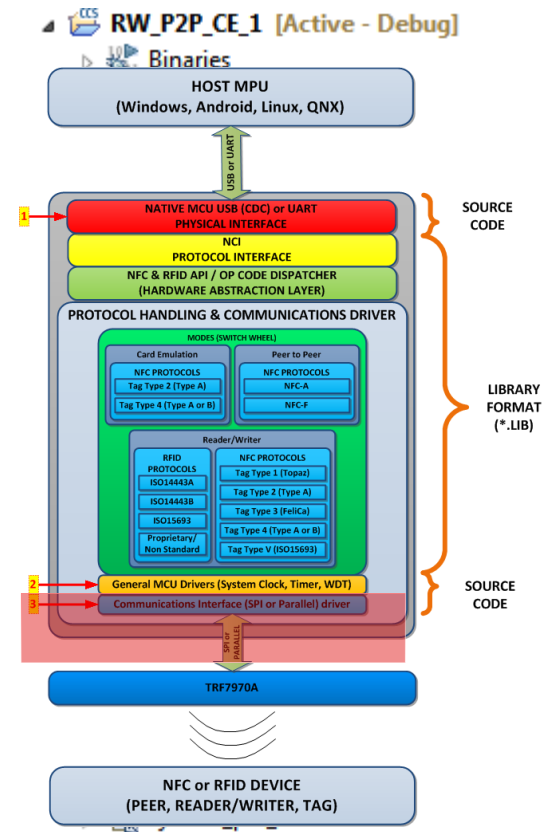


6/18/2013

TRF7970A Interface Modifications

There are **five** functions that **must** be reviewed:

1. **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/.h)
Note: For this release only SPI w/ SS will be supported.
2. **RFSPi::init()** –
Initializes the SPI module as 3 pin SPI, 8 bit Master, MSB, Clock Pol/ Phase = 0, SPI Clock ~ 4MHz using the SMCLK = 25MHz as reference. The Slave Select will be manually set – ensure that SLAVESELECT_OUTPUT is using the correct GPIO (SPI.cpp/.h)
Note: Please see the datasheet to match
3. **TRF797x_IRQ_ISR()** –
Ensure that the interrupt service routine for the IRQ pin (rising edge interrupt) is setup correctly. (RFID.cpp)
4. **RFSPi::waitBus()** –
Waits for the SPI module to be idle. (Needs to be modified depending on the USCI) (SPI.cpp)
5. **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)

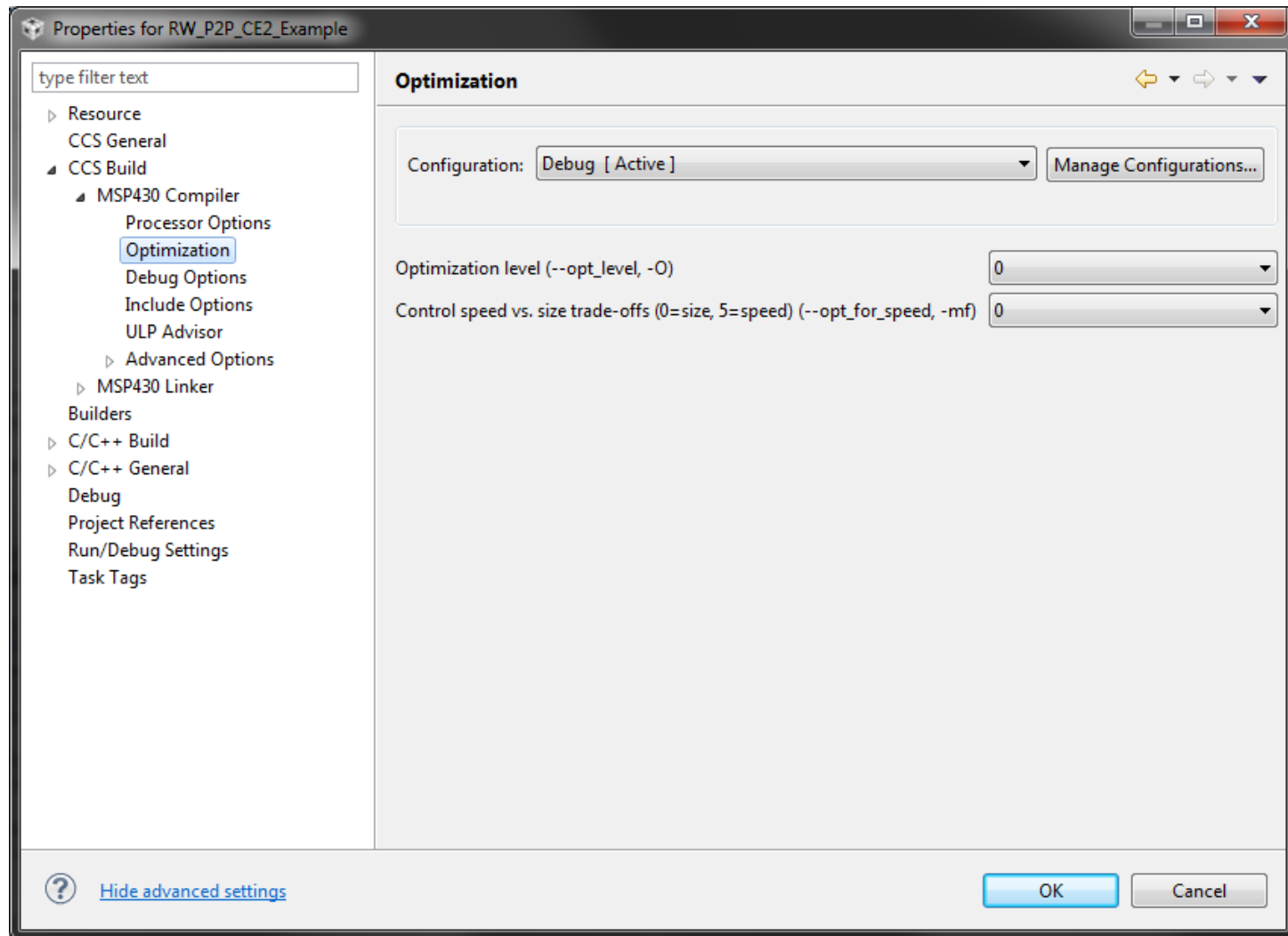


6/18/2013

MEMORY FOOTPRINTS

6/18/2013

CCS Optimization Options



NFCLink CCS Memory Footprint Overview

Mode + USB CDC (CCS Project stand alone w/o library)	Optimization Level (0 – 4)	Flash (kB)	RAM (kB)
Card Emulation	0	84.3	6.2
	4	62.3	6.2
Peer to Peer	0	71.2	6.2
	4	51.3	6.2
Reader/Writer	0	78.4	6.2
	4	57.2	6.2
All Modes	0	91.5	6.2
	4	68.1	6.2

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

Note 2: The memory footprints listed above are preliminary.

6/18/2013

RW_P2P_CE_1 Memory Footprint

RW_P2P_CE1_Ex Optimization Level (0 – 4)	nfcLink.lib Optimization Level (0 – 4)	Flash (kB)	RAM (kB)	
0	0	91.3	5.8	Larger Footprint (Development)
1	0	88.2	5.8	
2	0	86.5	5.8	
3	0	87	5.8	
4	0	85	5.8	
0	4	77.1	5.8	Smaller Footprint (Release)
1	4	74.1	5.8	
2	4	72.3	5.8	
3	4	72.8	5.8	
4	4	67.7	5.8	

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

/18/2013

RW_P2P_CE_2 Memory Footprint

RW_P2P_CE2_Ex Optimization Level (0 – 4)	nfcLink.lib Optimization Level (0 – 4)	Flash (kB)	RAM (kB)	
0	0	85.1	6	Larger Footprint (Development)
1	0	84.2	6	
2	0	83.1	6	
3	0	83.4	6	
4	0	81.2	6	
0	4	71	6	Smaller Footprint (Release)
1	4	70	6	
2	4	68.9	6	
3	4	69.3	6	
4	4	63.8	6	

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

/18/2013

SUMMARY

6/18/2013

Summary

- NFCLink Overview
- NFCStack + Evaluation Overview
- NFCLink Beta Release components
- Software + Hardware Roadmap
- NFCLink File Structure
- Update cycle - Quarterly
- For more information: www.ti.com/nfc

Backup

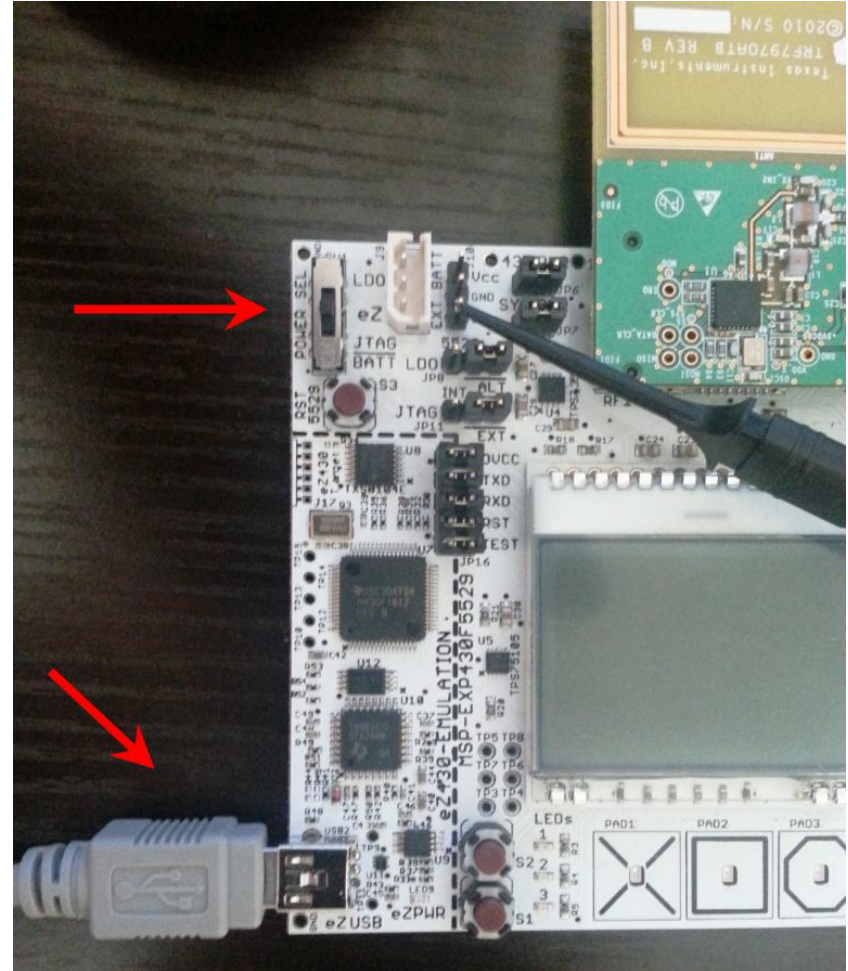
6/18/2013

Download Provided Firmware Image to MSP-EXP430F5529 board

- Download, Unzip and Install MSP430 Flash Programming Tool from Elprotrotronic website
- <http://www.elprotrotronic.com/files/FET-Pro430-Setup.zip>
- Connect USB-A end of USB cable to PC

Download Provided Firmware Image to MSP-EXP430F5529 board (cont.)

- Place POWER SEL switch on MSP-EXP430F5529 board (upper left hand side of the board) in the eZ position (middle)
- Connect mini-USB cable end of USB cable to ezUSB connector on MSP-EXP430F5529 board. (bottom left side of board)
- Open Elprotronic FET-Pro430 tool Graphical User Interface (GUI)



6/18/2013

Download Provided Firmware Image to MSP-EXP430F5529 board (cont.)

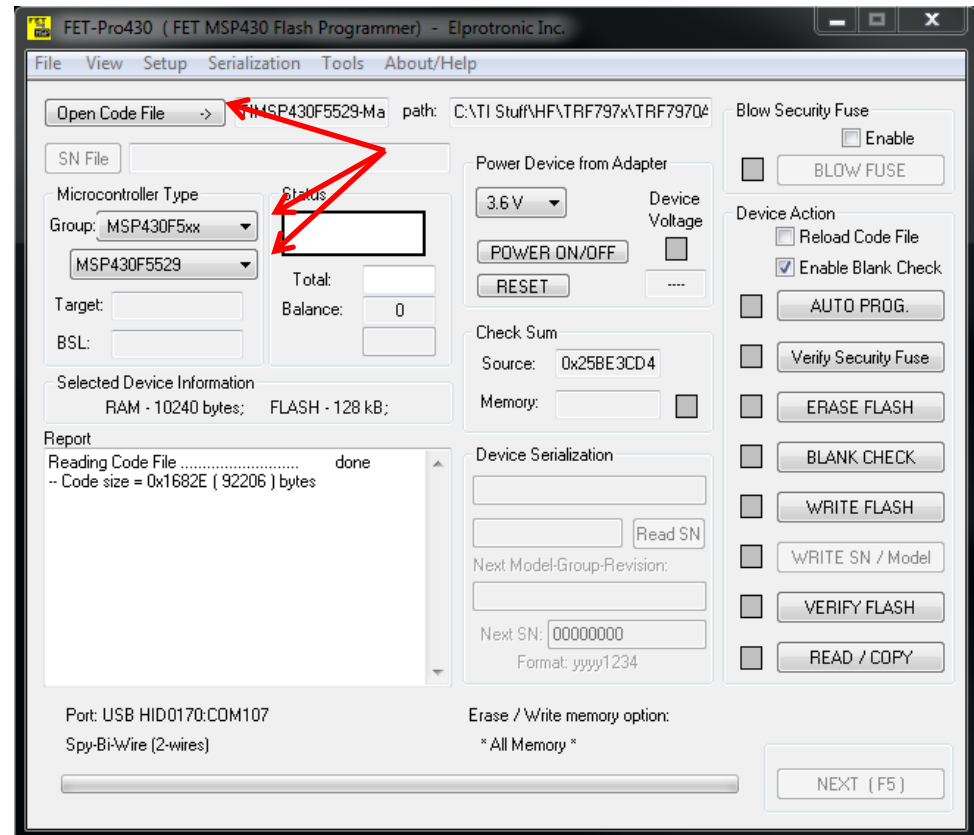
- In upper left corner of the GUI, press Open Code File Button and navigate to where provided FW image is stored on PC and select it.

– Examples/allModes/RW_P2P_CE_1/ccs/Debug/RW_P2P_CE_1.hex

or

– Examples/allModes/RW_P2P_CE_2/ccs/Debug/RW_P2P_CE_2.hex

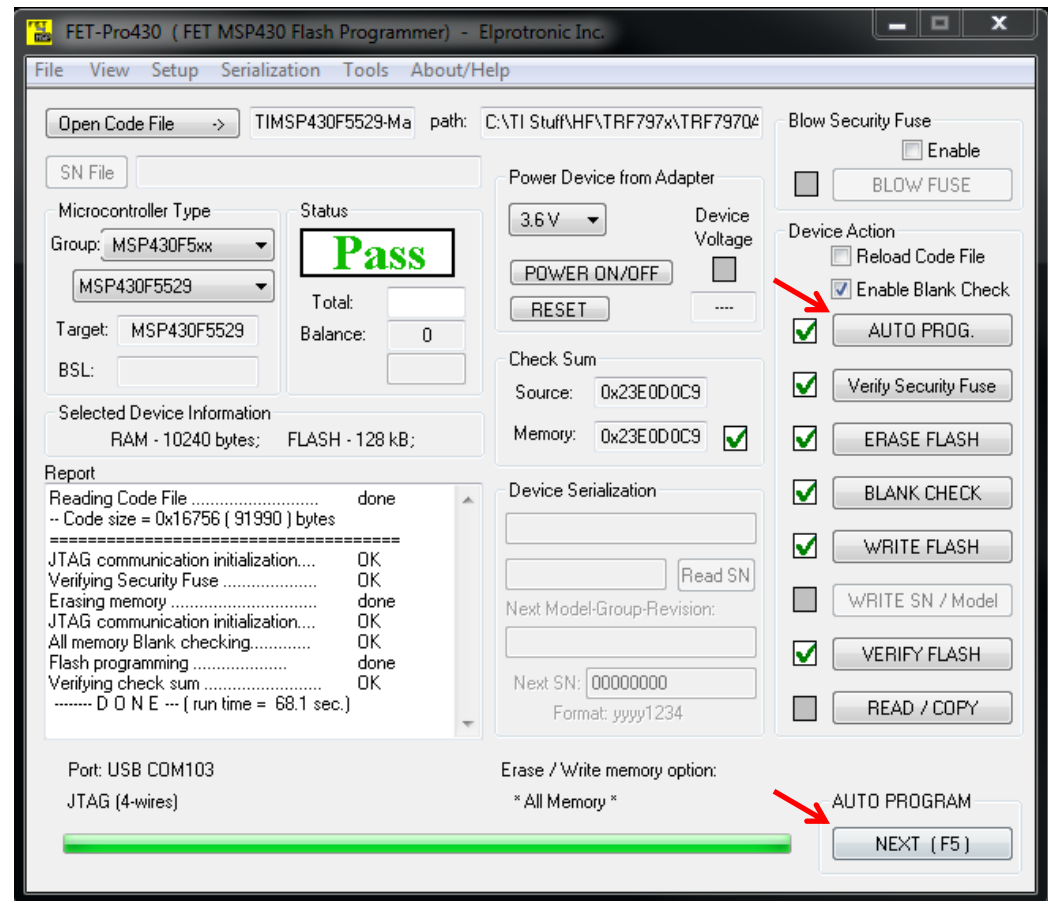
- Choose Microcontroller Type Group: (using dropdown) MSP430F5xx
- Choose (using dropdown) MSP430F5529



6/18/2013

Download Provided Firmware Image to MSP-EXP430F5529 board (cont.)

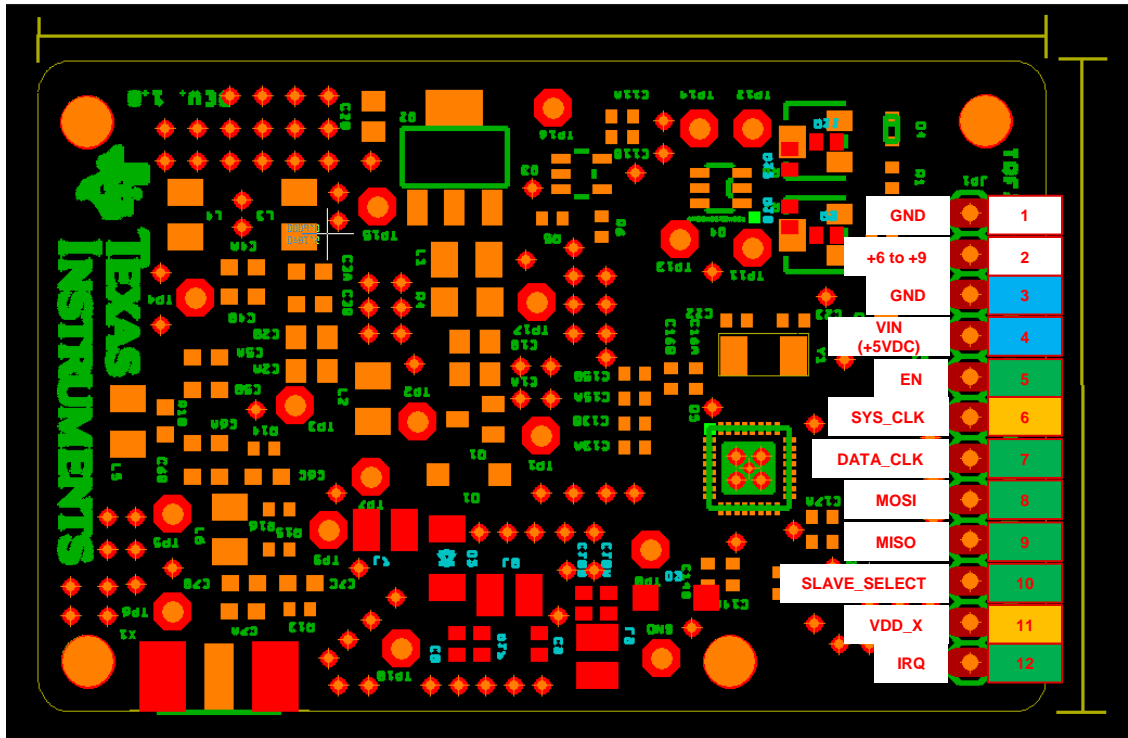
- Next, press the AUTO PROG. button in the Elprotronic tool and allow the tool to complete the steps.
- NOTE: If more than one board is to be programmed at this time, the NEXT (or F5) button can be used to program additional units instead repeating all the steps on previous slide.



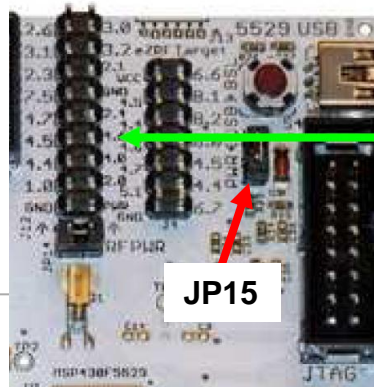
6/18/2013

NFCLINK + EMVCO AMPLIFIER EXAMPLE

NFCLink MCU Board to TRF7970A EMVCo Amplifier Hardware Connections



OTHER CONNECTION	MSP-EXP430F5529 HDR J12	SIGNAL NAME	TRF7970A AMP BOARD JP1
EXT. P/S	N/C	GND	PIN 1
EXT. P/S	N/C	+6VDC TO +9VDC	PIN 2
	GND	GND	PIN 3
JP15 (-F5529 BD.)	N/C	VIN (+5VDC)	PIN 4
	P2.3	EN	PIN 5
	N/C	SYS_CLK	PIN 6
	P3.2	DATA_CLK	PIN 7
	P3.0	MOSI	PIN 8
	P3.1	MISO	PIN 9
	P2.6	SLAVE_SELECT	PIN 10
	N/C	VDD_X	PIN 11
	P2.0	IRQ	PIN 12



Logic Analyzer and Debug Header J12

References

- ISO/IEC15693-3
- ISO/IEC14443-3, ISO/IEC14443-4
- ISO/IEC7816-4
- ISO/IEC18092
- NFC Forum Specifications
 - <http://www.nfc-forum.org/specs/>
- MSP-EXP430F5529 Users Guide
 - <http://www.ti.com/lit/pdf/slau330>
(schematic available in Section 5.3)
- TRF7970A Data Sheet
 - <http://www.ti.com/product/trf7970a>