*TI Designs*

# *Wi-Fi® Enabled IoT Node With NFC Connection Handover on High Performance Microcontrollers*

## TI Designs

TI Designs provide the foundation that you need including methodology, testing and design files to quickly evaluate and customize the system. TI Designs help *you* accelerate your time to market.

## Design Resources

| | |
|---|---|
| TIDM-TM4C129XNFC | Design Folder |
| TM4C1294NCPDT | Product Folder |
| SimpleLink CC3100 | Product Folder |
| TRF7970A | Product Folder |
| RF430CL330H | Product Folder |
| EK-TM4C1294XL | Tools Folder |
| SimpleLink CC3100 BoosterPack | Tools Folder |
| NFC Transceiver BoosterPack | Tools Folder |
| NFC Transponder BoosterPack | Tools Folder |
| CC31XXEMUBOOST | Tools Folder |

TI E2E™ Community

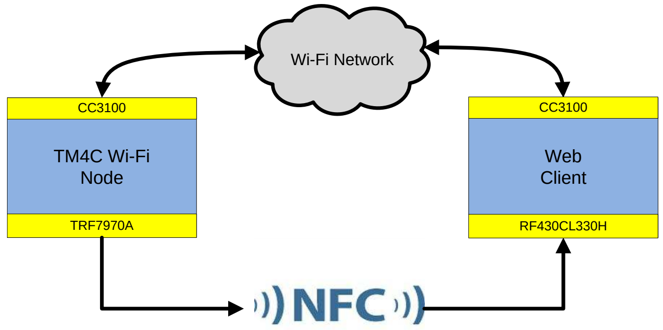
ASK Our E2E Experts
WEBENCH® Calculator Tools

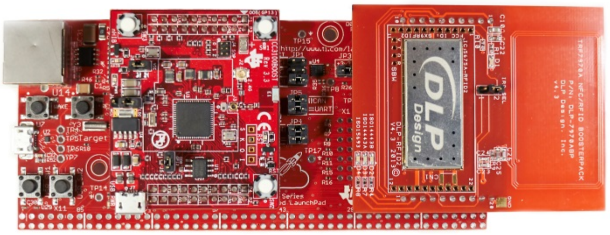
## Design Features

- TM4C1294 MCU Host and CC3100 Network Processor as Wi-Fi HTTP Server
- Wi-Fi HTTP Server Works as Access Point or Station
- CC3100 Executes Wi-Fi Stack, Simplifying TM4C1294 Host Processor Application
- HTML Code Remotely Controls TM4C1294 MCU From Web Browser
- TRF7970A NFC Transceiver Shares Pairing Information and HTTP Server URL in Access Point Mode
- RF430CL330H NFC Transponder Receives Pairing Information and Shares HTTP Server URL in Station Mode
- TI-RTOS Used for Task Scheduling and Peripheral Access

## Featured Applications

- Industrial Application and Automation
- Home Automation
- Smart Grid and Energy
- Test and Measurement





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

All trademarks are the property of their respective owners.

# 1 System Description

One of the most common wireless connectivity technologies used today is Wi-Fi. Configuring Wi-Fi network connection parameters is straight forward in user oriented devices such as laptop computers, smartphones, or tablets that include an intuitive user interface. Configuring embedded applications such as Industrial and Home Automation, Smart Grid and Energy, or Test and Measurement applications without a user interface requires a connection to a computer or other device to enter the required pairing information. Near field communication (NFC) can complete the configuration process with a simple tap.

This reference design illustrates NFC connection handover (pairing) and URL sharing with a Wi-Fi node using a TM4C1294 high-performance microcontroller and a CC3100 network processor. This reference design builds on the Wi-Fi Enabled IoT Node with High Performance MCU Reference Design.

The software accompanying this design requires an EK-TM4C1294XL LaunchPad™, a CC3100 BoosterPack (Wi-Fi), and a TRF7970A BoosterPack (NFC, access point) or RF430CL330H (NFC, station). For remote access to the Wi-Fi node, it is required to program the included HTML files to the CC3100 using the TI Uniflash tool.

TI-RTOS has been used for scheduling the various tasks. The use of a RTOS is highly recommended to distribute the load and make the application easily scalable.

## 1.1 *TM4C1294NCPDT*

The TM4C1294NCPDT is a 120-MHz high-performance microcontroller with a 1MB on-chip Flash and 256KB on-chip SRAM and features an integrated Ethernet MAC+PHY for connected applications. The device has high bandwidth interfaces like a memory controller and a high-speed USB2.0 digital interface. Integrating a number of low- to mid-speed serials, up to a 4MSPS 12-bit ADC, and motion control peripherals, this device makes for a unique solution for a variety of applications ranging from industrial communication equipment to Smart Energy or Smart Grid applications.

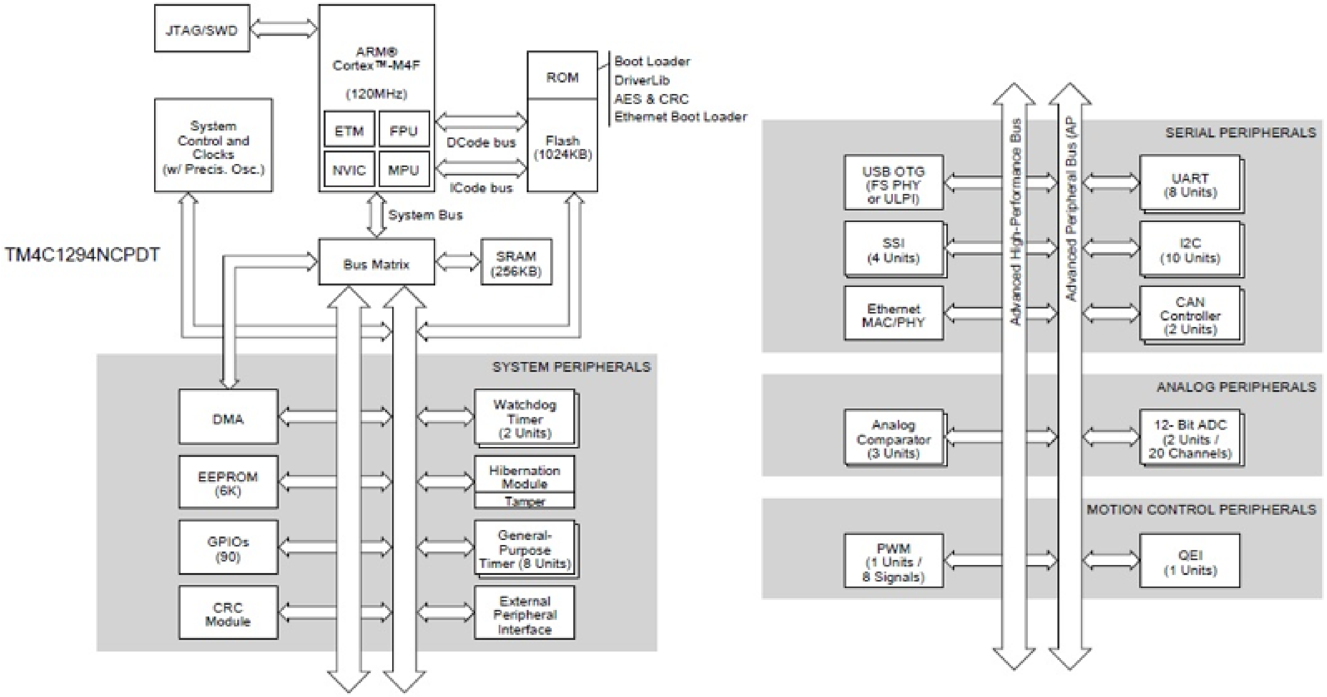


**Figure 1. TM4C1294NCPDT Microcontroller High-Level Block Diagram**

## 1.2 CC3100

The CC3100 Wi-Fi network processor subsystem features a Wi-Fi Internet-on-a-Chip™ and contains an additional dedicated ARM® MCU that completely offloads the host MCU. This subsystem includes an 802.11 b/g/n radio, baseband, and MAC with a powerful crypto engine for fast, secure Internet connections with 256-bit encryption. The CC3100 supports Station, Access Point, and Wi-Fi Direct modes. The device also supports WPA2 personal and enterprise security and WPS 2.0. This subsystem includes embedded TCP/IP and TLS/SSL stacks, HTTP server, and multiple Internet protocols.

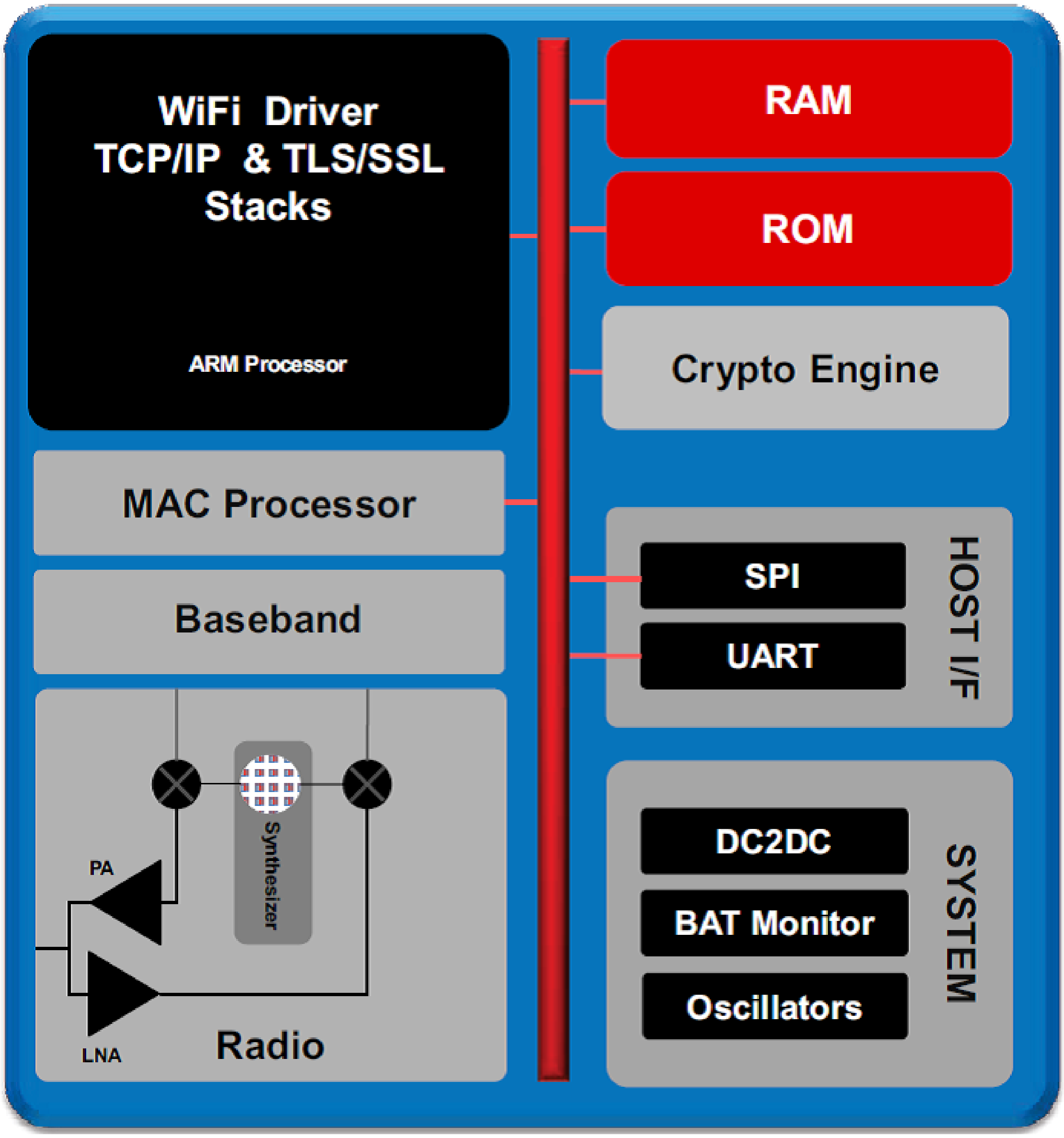


**Figure 2. CC3100 Hardware Overview**

## 1.3 TM4C1294/CC3100 Interface

The interface between the TM4C1294 and CC3100 is illustrated in Figure 3. The TM4C1294 can reset the CC3100 through the Reset pin. The TM4C1294 can also enable and disable the CC3100 by controlling the nHIB pin. When the CC3100 needs service, it toggles the CC_IRQ signal to notify the TM4C1294. The TM4C1294 then transfers information with the CC3100 over the SPI bus. The TI SimpleLink™ library driver processes the message and controls the communication.

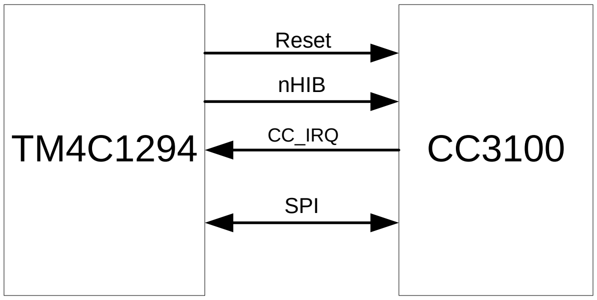


**Figure 3. TM4C1294/CC3100 Interface Overview**

## 1.4 TRF7970A

The TRF7970A is a high-performance, 13.56-MHz HF RFID and NFC Transceiver IC composed of an integrated analog front end (AFE) and a built-in data framing engine for the ISO15693, ISO14443A/ISO14443B, and FeliCa. This includes data rates up to 848 kbps for the ISO14443A/ISO14443B with all framing and synchronization tasks on board (in default mode). The TRF7970A also supports NFC Tag Type 1, 2, 3, and 4 operations. This architecture enables the customer to build a complete cost-effective yet high-performance, multi-protocol, 13.56-MHz RFID and NFC system together with a low-cost microcontroller.

## 1.5 TM4C1294/TRF7970A Interface

The interface between the TM4C1294 and TRF7970A is illustrated in Figure 4. The TM4C1294 can enable and disable the TRF7970A by controlling the Enable pin. When the TRF7970A needs service, it toggles the IRQ signal to notify the TM4C1294. The TM4C1294 then transfers information with the TRF7970A over the SPI bus. The TI NFC stack software processes the message and controls the communication based on the TI-RTOS SPI driver.

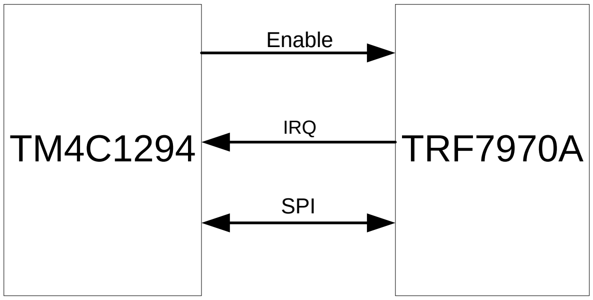


**Figure 4. TM4C1294/TRF7970A Interface Overview**

## 1.6 RF430CL330H

The TI Dynamic NFC Interface Transponder RF430CL330H is an NFC Tag Type 4 device that combines a wireless NFC interface and a wired SPI or I$^2$C interface to connect the device to a host. The NFC Data Exchange Format (NDEF) message in the SRAM can be written and read from the integrated SPI or I$^2$C serial communication interface and can also be accessed and updated wirelessly through the integrated ISO14443B-compliant RF interface that supports up to 848 kbps.

## 1.7 TM4C1294/RF430CL330H Interface

The interface between the TM4C1294 and RF430CL330H is illustrated in Figure 5. After any activity occurs in the RF430CL330H (for example, tag has being read or written), it notifies the TM4C1294 by toggling the INT0 signal. The TM4C1294 then exchanges information with the RF430CL330H over the I$^2$C bus to identify the interrupt cause and perform any required actions. The communication is based on TI RTOS I$^2$C driver.

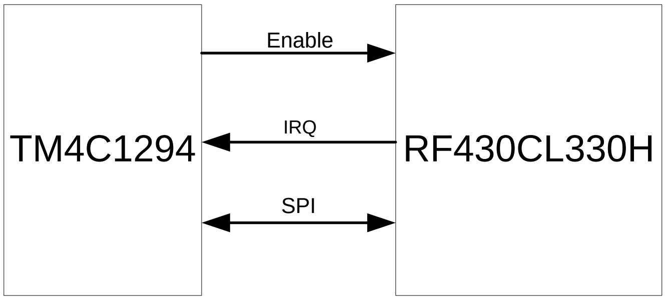


**Figure 5. TM4C1294/RF430CL330H Interface Overview**

## 2 System Functionality Block Diagram

The TM4C Wi-Fi Node can be configured in two modes: Access Point and Station.

### 2.1 TM4C Wi-Fi Node as an Access Point

When the TM4C Wi-Fi node is configured as a Wi-Fi access point, the Wi-Fi web client is directly connected to the TM4C Wi-Fi node as shown in Figure 6. There is no connection to the Internet through Wi-Fi. The Wi-Fi node serves as a host for a web server and allows remote control of the Wi-Fi Node LaunchPad.

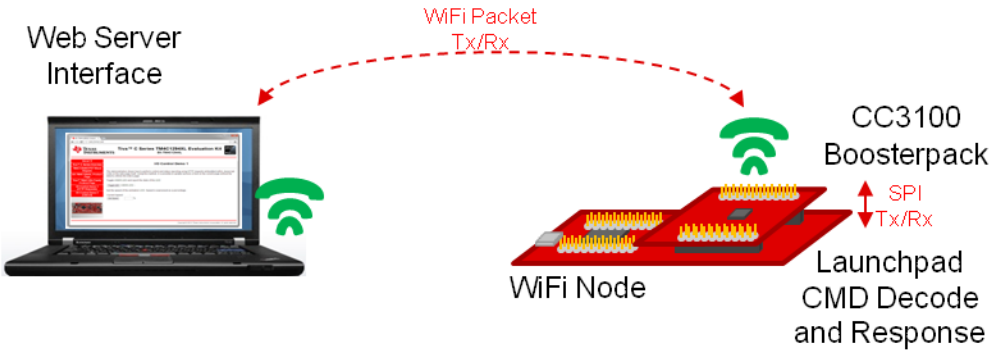


**Figure 6. General Setup and Data Flow for Wi-Fi Node Access Point**

NFC has two different purposes in an Access Point reference design:

1. Wi-Fi connection pairing
2. HTTP server URL sharing

While there is no client device connected to the Access Point, the TRF7970A NFC device will share the Wi-Fi connection details (SSID, Security Type, and Security Key) using both peer-to-peer (P2P) and read/write (R/W) NFC modes (R/W to interact with the RF430CL330H). Once a client device is connected, the P2P NFC message is replaced to share the HTTP server URL.

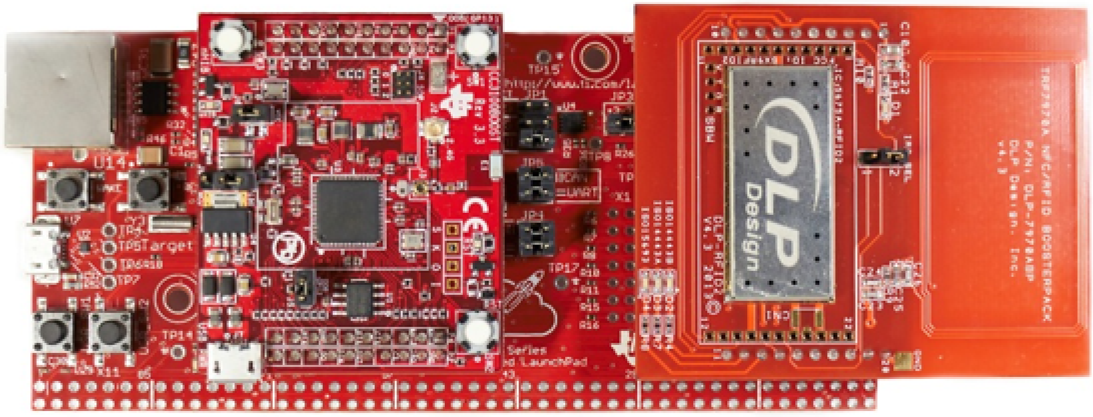


**Figure 7. Access Point Hardware**

## 2.2 TM4C Wi-Fi Node as a Wi-Fi Station

When configured as a station, the Wi-Fi node needs to connect with an existing network and acquire an IP address. Once connected, the Wi-Fi node can be accessed from a PC or smartphone connected to the same network through an internet browser such as Internet Explorer®, Chrome®, or Firefox® as a web server. The web server will be the GUI to enable control of the LaunchPad functions. When a particular function is asserted through the web server, a command is sent to the CC3100 BoosterPack, which then passes the command to the LaunchPad. Upon receipt of the command, the LaunchPad executes the action or task associated with the command and responds with the appropriate data or acknowledgment as needed.

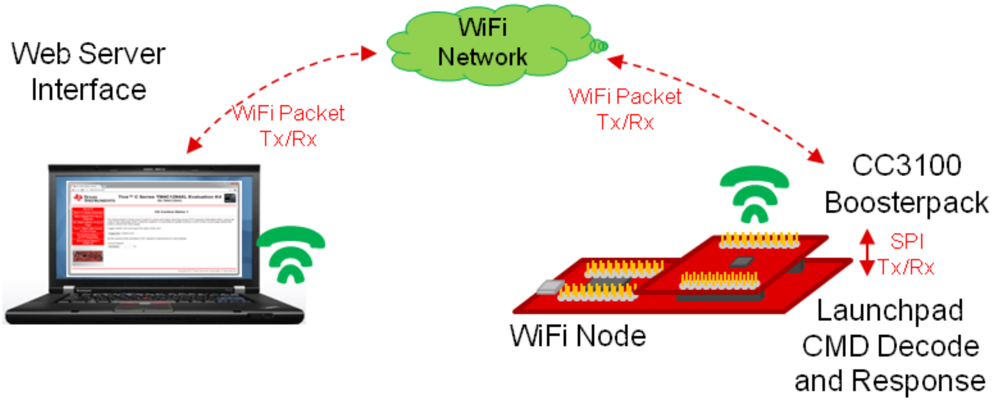


**Figure 8. General Setup and Data Flow for Wi-Fi Node Station**

The NFC interface has two different purposes in a station reference design:

1. Receive Wi-Fi pairing information
2. Share HTTP server URL

For this application, the RF430CL330H is being used instead of the TRF7970A as an Access Point. The RF430CL330H typically provides information as an NFC tag only, but the capability to upload the NDEF message wirelessly provides a simple way to receive the Wi-Fi pairing information. This device is affordable, requires a simpler and smaller software in the host microcontroller, and consumes less energy when compared to the TRF7970A. See more details of these two modes in Section 6.2.

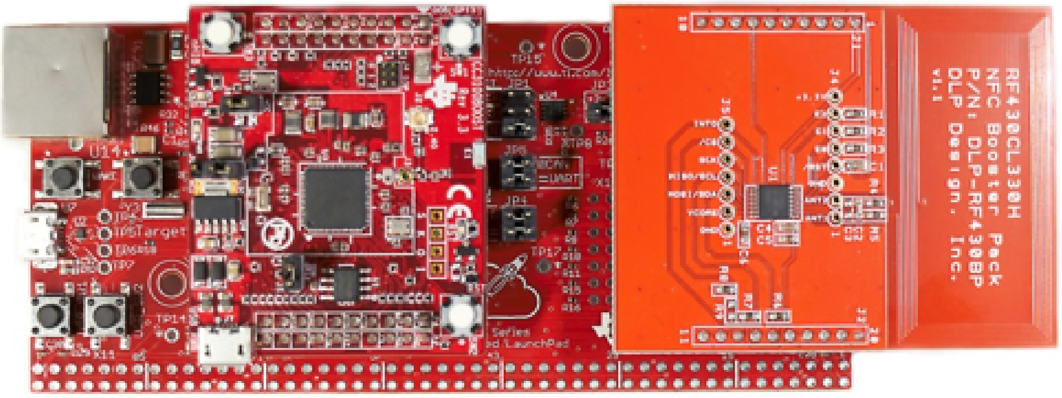


**Figure 9. Station Hardware**

## 3 Getting Started Hardware

For hardware, both modes (Access Point and Station) use a EK-TM4C1294XL Connected LaunchPad as the main board and the CC3100 BoosterPack for Wi-Fi interface. These two boards are connected through BoosterPack connector 2. The signal mapping is illustrated in Table 1:

**Table 1. EK-TM4C1294XL CC3100BP Signal Mapping**

| BOOSTERPACK CONNECTOR | CC3100 BOOSTERPACK | TM4C1294 LAUNCHPAD |
|---|---|---|
| A2-1 | 3.3 V | 3.3 V |
| A2-2 | Open | PD2 |
| A2-3 | CC_UART1_TX | PP0_U6RX |
| A2-4 | CC_UART1_RX | PP1_U6TX |
| A2-5 | CC_nHIB | PD4 |
| A2-6 | Open | PD5 |
| A2-7 | CC_SPI_CLK | PQ0_SSI3CLK |
| A2-8 | Open | PP4 |
| A2-9 | Test_3 | PN5 |
| A2-10 | FORCE_AP | PNA |
| B2-1 | 5 V | 5 V |
| B2-2 | GND | GND |
| B2-3 | Open | PB4 |
| B2-4 | Open | PB5 |
| B2-5 | Open | PK0 |
| B2-6 | Open | PK1 |
| B2-7 | Open | PK2 |
| B2-8 | Open | PK3 |
| B2-9 | Open | PA4 |
| B2-10 | Open | PA5 |
| C2-1 | Test_29 | PG1 |
| C2-2 | Test_30 | PK4 |
| C2-3 | Open | PK5 |
| C2-4 | CC_URT1_CTS | PM0 |
| C2-5 | CC_UART1_RTS | PM1 |
| C2-6 | Open | PM2 |
| C2-7 | CC_NWP_UART_TX | PH0 |
| C2-8 | CC_WL_UART_TX | PH1 |
| C2-9 | CC_WLRS232_RX | PK6 |
| C2-10 | CC_WLRS232_TX | PK7 |
| D2-1 | GND | GND |
| D2-2 | CC_IRQ | PM7 |
| D2-3 | CC_SPI_CS | PP5 |
| D2-4 | Open | PA7 |
| D2-5 | MCU_RESET_IN | RESET |
| D2-6 | CC_SPI_DIN | PQ2_SSI3MOSI |
| D2-7 | CC_SPI_DOUT | PQ3_SSI3MISO |
| D2-8 | Test_63 | PP3 |
| D2-9 | Test_64 | PQ1 |
| D2-10 | Test_18 | PM6 |

The TM4C1294 and CC3100 interface is illustrated by the block diagram in Figure 3. The TM4C1294 controls the enable/disable of the CC3100. When the CC3100 needs service, it toggles the "CC_IRQ" pin to notify the TM4C1294. The TM4C1294 reads the CC3100 request through the SPI port and perform the rest of communication. The following TM4C1294 I/O pins are used as the interface to CC3100:

- SSI3 is in three-pin SPI mode (CLK, MOSI, SOMI).
- GPIO output pin PP5 is manually controlled as SPI CS for the CC3100.
- GPIO input interrupt on pin PM7 is enabled to capture the request from the CC3100.
- GPIO output pin PD4 is configured to enable/disable the CC3100.

For the Access Point reference design, the EK-TM4C1294XL Connected LaunchPad board is connected to the DLP-7970ABP board through the BoosterPack connector 1. The signal mapping is illustrated in Table 2:

**Table 2. EK-TM4C1294XL DLP-7970ABP Signal Mapping**

| BOOSTERPACK CONNECTOR | DLP-7970ABP | TM4C1294 LAUNCHPAD |
|---|---|---|
| A1-1 | 3.0 V | 3.3 V |
| A1-2 | Unused | PE4 |
| A1-3 | Unused | PC4_U7RX |
| A1-4 | Unused | PC5_U7TX |
| A1-5 | Unused | PC6 |
| A1-6 | Unused | PE5 |
| A1-7 | SPI_CLK | PD3_SSI2CLK |
| A1-8 | IRQ (IRQ_SEL 2) | PC7 |
| A1-9 | SPI_CS | PB2 |
| A1-10 | EN | PB3 |
| D1-1 | GND | GND |
| D1-2 | Unused | PM3 |
| D1-3 | XOUT (IRQ_SEL 1) | PH2 |
| D1-4 | Unused | PH3 |
| D1-5 | RESET | RESET |
| D1-6 | MOSI | PD1_SSI2XDAT0 |
| D1-7 | MISO | PD0_SSIXDATA1 |
| D1-8 | LED (ISO15693) | PN2 |
| D1-9 | LED (ISO14443A) | PN3 |
| D1-10 | LED (ISO14443B) | PP2 |

The TM4C1294/TRF7970A interface is illustrated by the block diagram in Figure 4. The TM4C1294 controls the enable/disable of the TRF7970A. When the TRF7970A needs service, it toggles the "IRQ" pin to notify the TM4C1294. The TM4C1294 will read the TRF7970A request through the SPI port and perform the rest of communication. The following TM4C1294 I/O pins are used as the interface to CC3100:

- SSI2 is in three pin SPI mode (CLK, MOSI, SOMI).
- GPIO output pin PB2 is manually controlled as SPI CS for the TRF7970A.
- GPIO input interrupt on pin PC7 is enabled to capture the request from the TRF7970A.
- GPIO output pin PB3 is configured to enable/disable the TRF7970A.

For the Station reference design, the EK-TM4C1294XL Connected LaunchPad board is connected to the DLP-RF430BP board through the BoosterPack connector 1. The signal mapping is illustrated in Table 3:

**Table 3. EK-TM4C1294XL DLP-RF430BP Signal Mapping**

| BOOSTERPACK CONNECTOR | DLP-RF430BP | TM4C1294 LAUNCHPAD |
|---|---|---|
| A1-1 | 3.0 V | 3.3 V |
| A1-2 | Unused | PE4 |
| A1-3 | Unused | PC4_U7RX |
| A1-4 | Unused | PC5_U7TX |
| A1-5 | Unused | PC6 |
| A1-6 | Unused | PE5 |
| A1-7 | DATA_CLK | PD3_SSI2CLK |
| A1-8 | RESET | PC7 |
| A1-9 | Unused | PB2 |
| A1-10 | Unused | PB3 |
| D1-1 | GND | GND |
| D1-2 | Unused | PM3 |
| D1-3 | Unused | PH2 |
| D1-4 | Unused | PH3 |
| D1-5 | Unused | RESET |
| D1-6 | MOSI/SDA | PD1_I2C7SDA |
| D1-7 | MISO/SCL | PD0_I2C7SCL |
| D1-8 | SPI_CS | PN2 |
| D1-9 | INTO | PN3 |
| D1-10 | Unused | PP2 |

The TM4C1294/RF430CL330H interface is illustrated by the block diagram in Figure 5. When the RF430CL330H needs service, it toggles the "INT0" pin to notify the TM4C1294. The TM4C1294 will read the RF430CL330H request through the $I^2C$ port and perform the rest of communication. The following TM4C1294 I/O pins are used as the interface to RF430CL330H:

- I2C7 to read/write RF430CL330H registers and SRAM containing NDEF message.
- GPIO input interrupt on pin PN3 to capture any service request from RF430CL330H.

The following additional TM4C1294 peripherals are also enabled in the system:

- Timer 0 to sample buttons and internal temperature sensor every 10 ms
- Timer 1 to generate 1-ms system tick
- Timer 2 to toggle LED with variable periods
- Timer 4 for TRF7970A timing control
- GPIO output pins PN0, PN1, PF0, and PF4 to control user LEDs
- GPIO input pins PJ0 and PJ1 to sample the button states
- ADC0 channel 3 to read the internal temperature sensor
- UART 0 is configured to 8N1 and baud rate of 115200 as a diagnostic port
- EEPROM to store the received pairing information in Station device

# 4 Getting Started Software

## 4.1 Access Point Software Architecture

The architecture of the TM4C1294 Access Point software is illustrated in Figure 10. The TI SimpleLink library driver processes the message between the TM4C1294 and CC3100. The NFC stack software is used to process communication between the TM4C1294 and TRF7970A (NFC stack has been adapted to work on the TM4C1294 microcontroller using TI-RTOS). TI-RTOS is used for task scheduling, intertask communication, and interrupt handling. NFC/pairing-related tasks interact with the hardware using TI-RTOS provided drivers (which rely over TivaWare™ library). Wi-Fi server-related tasks use the TivaWare library directly to access the corresponding hardware.

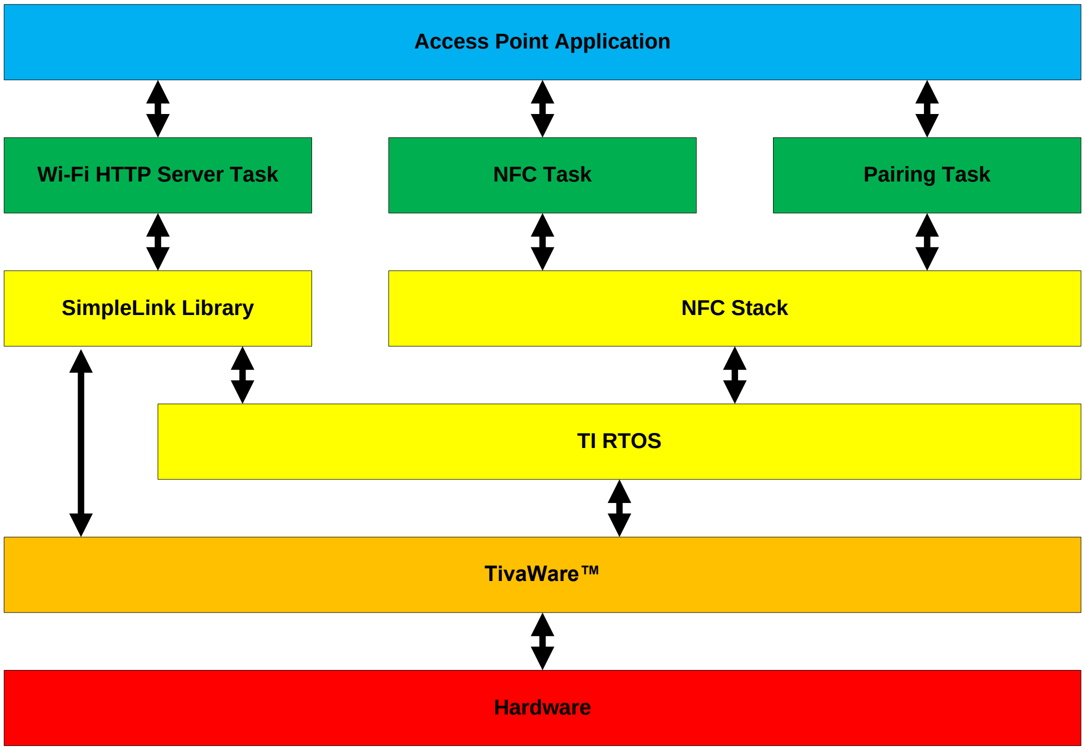


**Figure 10. Access Point Architecture Block Diagram**

The following TI-RTOS functions are statically configured in the TI-RTOS configuration file:

- CLK0: 1-ms system tick generated by Timer 1
- HWI_cc3100_req: GPIO interrupt on PM7 for capturing CC3100 request and setting semaphore
- HWI_timer0a: 10-ms timer 0 interrupt for sampling buttons and internal temperature sensor
- HWI_timer2a: Timer 2 interrupt with variable period for toggling LED
- Queue_Station_Conn: Queue to pass information from Wi-Fi task to NFC task about client connection status
- Semaphore_NFC: Semaphore to resume NFC task after an interrupt (TRF7970A IRQ or Timer)
- Semaphore_Tag: Semaphore to resume Pairing Task if a Type 4 Tag is detected
- Semaphore_CC3100_Req: Semaphore to resume Wi-Fi task after a CC3100 interrupt
- Task_HeartBeat: Toggle LED every second to indicate the device is running
- Task_HttpServer: Perform all actions related to Wi-Fi functionality
- Task_Nfc: Perform all actions related to NFC functionality
- Task_Pairing: Decode message from a NFC Type 4 Tag indicating the client device is ready to receive Wi-Fi Handover information and invoke NFC Writer Mode
- Timer_NFC: Timer to control the required synchronization with the TRF7970A

### 4.2 Station Software Architecture

For Station implementation, the NFC architecture is simplified by the use of the RF430CL330H transponder. This device does not require any special NFC software running in the main microcontroller other than an I$^2$C interface and a simple command/response implementation.

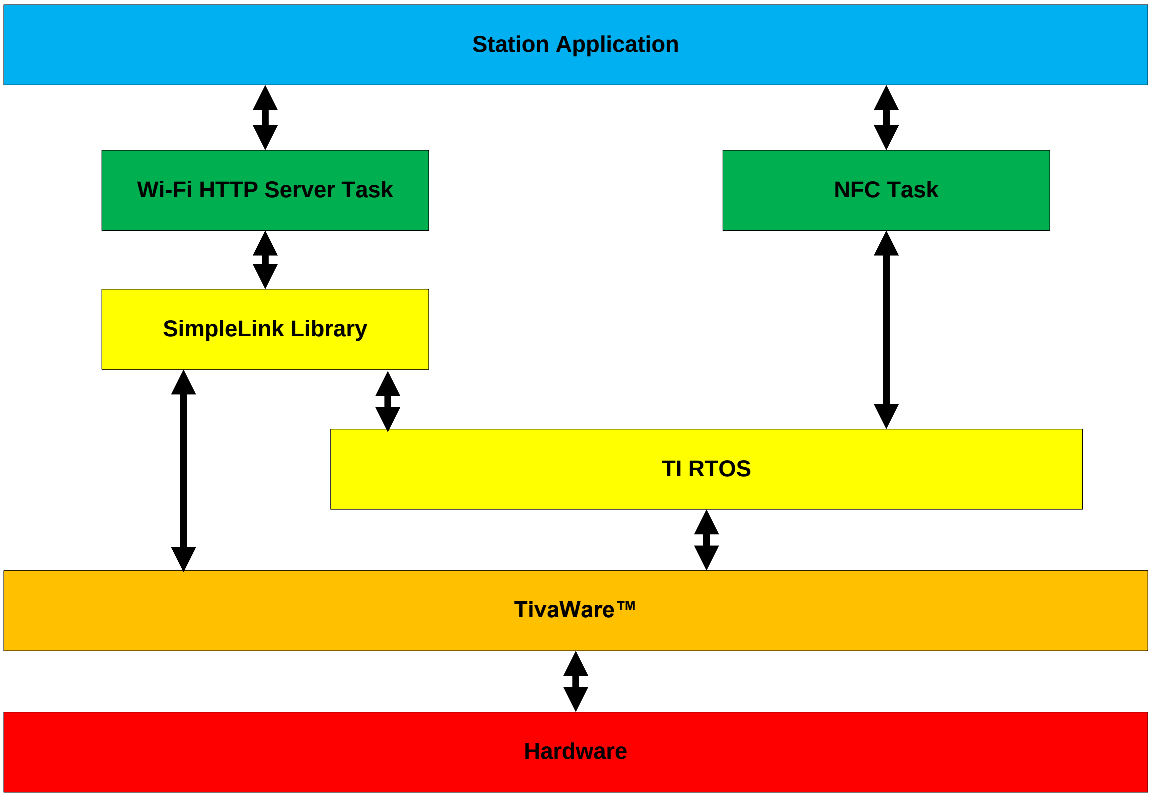


**Figure 11. Station Architecture Block Diagram**

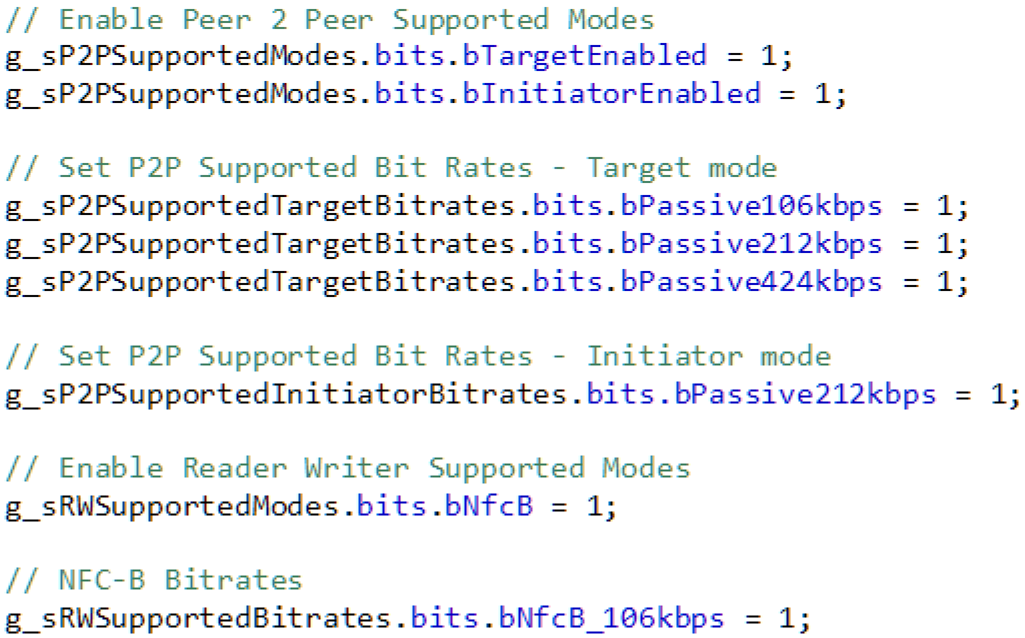
The following TI-RTOS functions are statically configured in the TI-RTOS configuration file:

- Clk0: 1-ms system tick generated by Timer 1
- Hwi_cc3100_req: GPIO interrupt on PM7 for capturing CC3100 request and setting semaphore
- Hwi_rf430_req: GPIO interrupt on PN3 for capturing RF430 request and setting semaphore
- Hwi_timer0a: 10-ms timer 0 interrupt for sampling buttons and internal temperature sensor
- Hwi_timer2a: Timer 2 interrupt with variable period for toggling LED
- Mailbox_Assigned_IP: Mailbox to pass the assigned IP address from Wi-Fi task to NFC task to create NDEF message
- Semaphore_RF430_Req: Semaphore to resume NFC task after an interrupt
- Semaphore_WiFi_Pairing: Semaphore to indicate Wi-Fi task that new pairing information is available in EEPROM
- Semaphore_CC3100_Req: Semaphore to resume Wi-Fi task after a CC3100 interrupt
- Task_HeartBeat: Toggles LED every second to indicate the device is running
- Task_HttpServer: Performs all actions related to Wi-Fi functionality
- Task_Nfc: Performs all actions related to NFC functionality
- Task_NfcNdefReplace: Replaces default NDEF message by dynamically created message containing IP address

## 4.3 NFC Communication Details

### 4.3.1 NFC Modes

For Access Point implementation, NFC is configured to work on two different modes, P2P and R/W. Supported NFC modes are setup in NFC_configuration API at nfc_utils.c source code file. For this application case, the following modes are enabled:



```
// Enable Peer 2 Peer Supported Modes
g_sP2PSupportedModes.bits.bTargetEnabled = 1;
g_sP2PSupportedModes.bits.bInitiatorEnabled = 1;

// Set P2P Supported Bit Rates - Target mode
g_sP2PSupportedTargetBitrates.bits.bPassive106kbps = 1;
g_sP2PSupportedTargetBitrates.bits.bPassive212kbps = 1;
g_sP2PSupportedTargetBitrates.bits.bPassive424kbps = 1;

// Set P2P Supported Bit Rates - Initiator mode
g_sP2PSupportedInitiatorBitrates.bits.bPassive212kbps = 1;

// Enable Reader Writer Supported Modes
g_sRWSupportedModes.bits.bNfcB = 1;

// NFC-B Bitrates
g_sRWSupportedBitrates.bits.bNfcB_106kbps = 1;
```

**Figure 12. NFC Mode Selection Code**

For Station implementation, the RF430CL330H is a NFC Tag Type 4 device in compliance with the ISO14443B RF interface. There are no configurable modes.

### 4.3.2 NDEF Message Definition

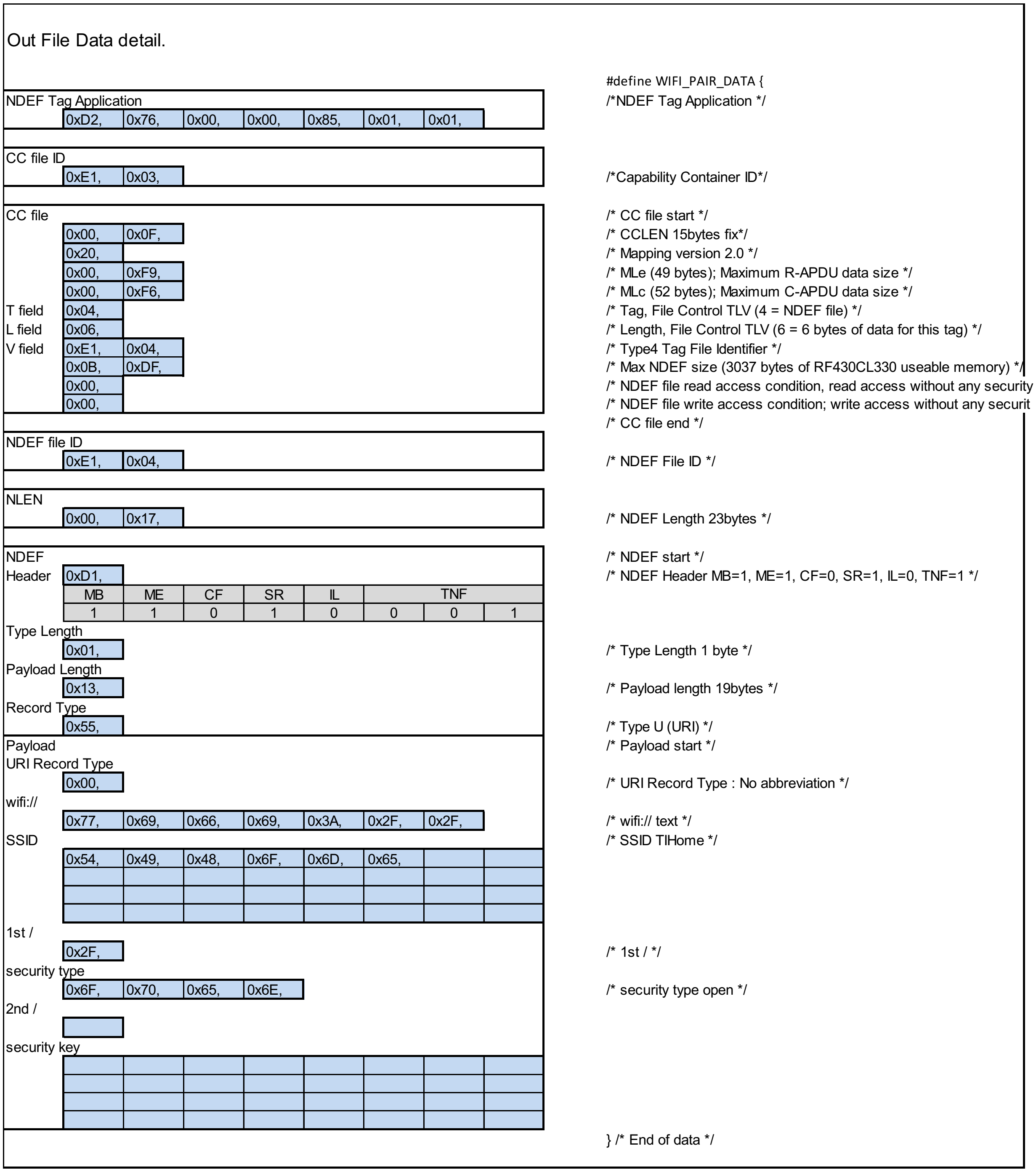
The NDEF message used to share Wi-Fi pairing details is a customized URI record following this format:

**wifi://[network ssid]/[wep|wpa|open]/[network key]**

By default, the Wi-Fi pairing parameters are defined as follows:

**wifi://TIHome/open**

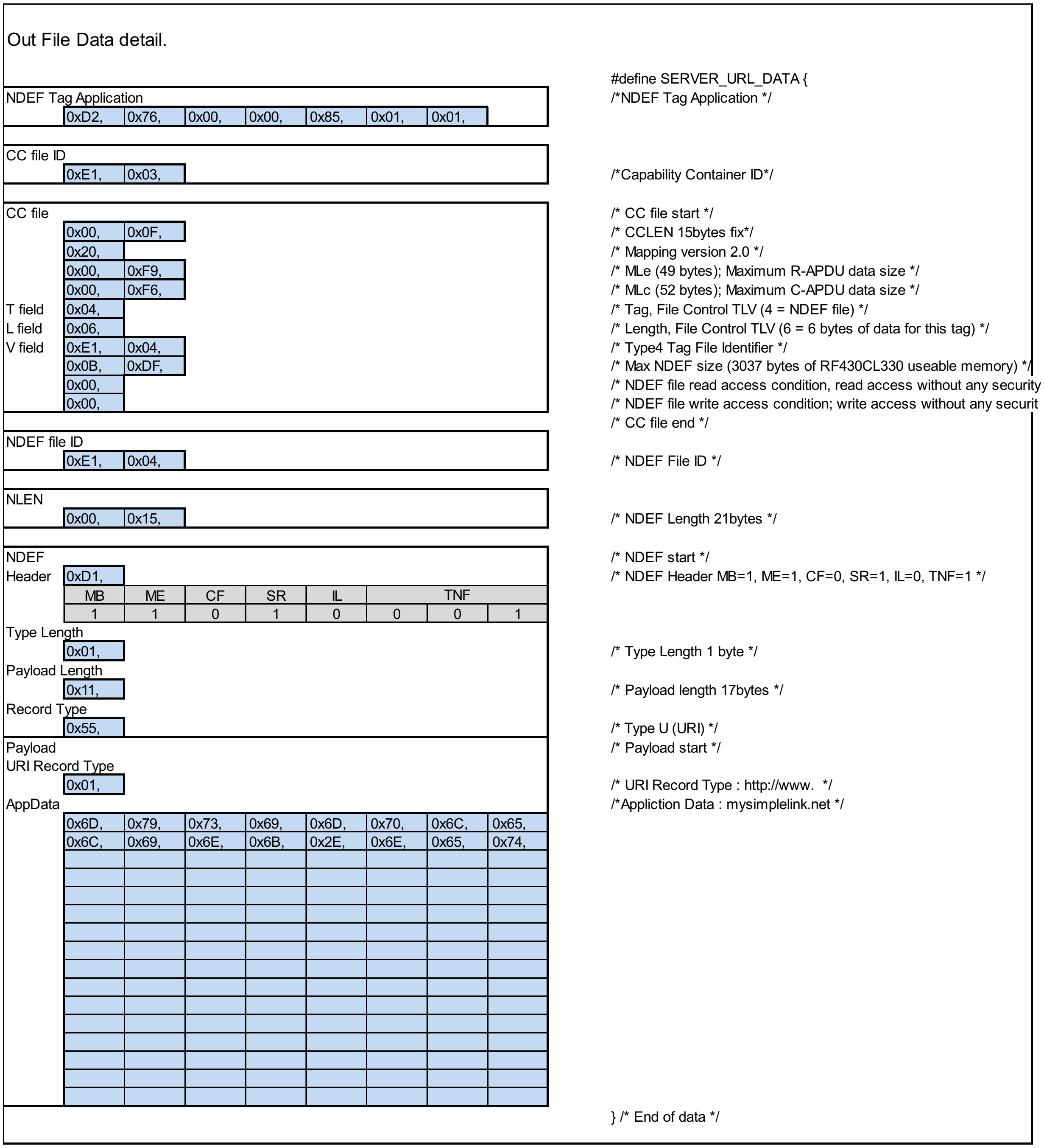
For an NDEF message definition, the user can use the NDEF Maker application found here (http://www.ti.com/lit/zip/sloa187) to create the following message:



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Out File Data detail. | | | | | | | | | |
| | | | | | | | | | #define WIFI_PAIR_DATA { |
| NDEF Tag Application | | | | | | | | | /*NDEF Tag Application */ |
| | 0xD2, | 0x76, | 0x00, | 0x00, | 0x85, | 0x01, | 0x01, | | |
| CC file ID | | | | | | | | | |
| | 0xE1, | 0x03, | | | | | | | /*Capability Container ID*/ |
| CC file | | | | | | | | | /* CC file start */ |
| | 0x00, | 0x0F, | | | | | | | /* CCLEN 15bytes fix*/ |
| | 0x20, | | | | | | | | /* Mapping version 2.0 */ |
| | 0x00, | 0xF9, | | | | | | | /* MLe (49 bytes); Maximum R-APDU data size */ |
| | 0x00, | 0xF6, | | | | | | | /* MLc (52 bytes); Maximum C-APDU data size */ |
| T field | 0x04, | | | | | | | | /* Tag, File Control TLV (4 = NDEF file) */ |
| L field | 0x06, | | | | | | | | /* Length, File Control TLV (6 = 6 bytes of data for this tag) */ |
| V field | 0xE1, | 0x04, | | | | | | | /* Type4 Tag File Identifier */ |
| | 0x0B, | 0xDF, | | | | | | | /* Max NDEF size (3037 bytes of RF430CL330 useable memory) */ |
| | 0x00, | | | | | | | | /* NDEF file read access condition, read access without any security |
| | 0x00, | | | | | | | | /* NDEF file write access condition; write access without any securit |
| | | | | | | | | | /* CC file end */ |
| NDEF file ID | | | | | | | | | |
| | 0xE1, | 0x04, | | | | | | | /* NDEF File ID */ |
| NLEN | | | | | | | | | |
| | 0x00, | 0x17, | | | | | | | /* NDEF Length 23bytes */ |
| NDEF | | | | | | | | | /* NDEF start */ |
| Header | 0xD1, | | | | | | | | /* NDEF Header MB=1, ME=1, CF=0, SR=1, IL=0, TNF=1 */ |
| | MB | ME | CF | SR | IL | TNF | | | |
| | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | |
| Type Length | | | | | | | | | |
| | 0x01, | | | | | | | | /* Type Length 1 byte */ |
| Payload Length | | | | | | | | | |
| | 0x13, | | | | | | | | /* Payload length 19bytes */ |
| Record Type | | | | | | | | | |
| | 0x55, | | | | | | | | /* Type U (URI) */ |
| Payload | | | | | | | | | /* Payload start */ |
| URI Record Type | | | | | | | | | |
| | 0x00, | | | | | | | | /* URI Record Type : No abbreviation */ |
| wifi:// | | | | | | | | | |
| | 0x77, | 0x69, | 0x66, | 0x69, | 0x3A, | 0x2F, | 0x2F, | | /* wifi:// text */ |
| SSID | | | | | | | | | /* SSID TIHome */ |
| | 0x54, | 0x49, | 0x48, | 0x6F, | 0x6D, | 0x65, | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| 1st / | | | | | | | | | |
| | 0x2F, | | | | | | | | /* 1st / */ |
| security type | | | | | | | | | |
| | 0x6F, | 0x70, | 0x65, | 0x6E, | | | | | /* security type open */ |
| 2nd / | | | | | | | | | |
| | | | | | | | | | |
| security key | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | } /* End of data */ |

**Figure 13. Wi-Fi Pairing NDEF Definition**

For URL sharing in Access Point, the default value is defined as http://www.mysimplelink.net. The following NDEF URI record type message is defined to share this information:



Out File Data detail.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | #define SERVER_URL_DATA { |
| NDEF Tag Application | | | | | | | | | /*NDEF Tag Application */ |
| | 0xD2, | 0x76, | 0x00, | 0x00, | 0x85, | 0x01, | 0x01, | | |
| CC file ID | | | | | | | | | |
| | 0xE1, | 0x03, | | | | | | | /*Capability Container ID*/ |
| CC file | | | | | | | | | /* CC file start */ |
| | 0x00, | 0x0F, | | | | | | | /* CCLEN 15bytes fix*/ |
| | 0x20, | | | | | | | | /* Mapping version 2.0 */ |
| | 0x00, | 0xF9, | | | | | | | /* MLe (49 bytes); Maximum R-APDU data size */ |
| | 0x00, | 0xF6, | | | | | | | /* MLc (52 bytes); Maximum C-APDU data size */ |
| T field | 0x04, | | | | | | | | /* Tag, File Control TLV (4 = NDEF file) */ |
| L field | 0x06, | | | | | | | | /* Length, File Control TLV (6 = 6 bytes of data for this tag) */ |
| V field | 0xE1, | 0x04, | | | | | | | /* Type4 Tag File Identifier */ |
| | 0x0B, | 0xDF, | | | | | | | /* Max NDEF size (3037 bytes of RF430CL330 useable memory) */ |
| | 0x00, | | | | | | | | /* NDEF file read access condition, read access without any security |
| | 0x00, | | | | | | | | /* NDEF file write access condition; write access without any securit |
| | | | | | | | | | /* CC file end */ |
| NDEF file ID | | | | | | | | | |
| | 0xE1, | 0x04, | | | | | | | /* NDEF File ID */ |
| NLEN | | | | | | | | | |
| | 0x00, | 0x15, | | | | | | | /* NDEF Length 21bytes */ |
| NDEF | | | | | | | | | /* NDEF start */ |
| Header | 0xD1, | | | | | | | | /* NDEF Header MB=1, ME=1, CF=0, SR=1, IL=0, TNF=1 */ |
| | MB | ME | CF | SR | IL | TNF | | | |
| | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | |
| Type Length | | | | | | | | | |
| | 0x01, | | | | | | | | /* Type Length 1 byte */ |
| Payload Length | | | | | | | | | |
| | 0x11, | | | | | | | | /* Payload length 17bytes */ |
| Record Type | | | | | | | | | |
| | 0x55, | | | | | | | | /* Type U (URI) */ |
| Payload | | | | | | | | | /* Payload start */ |
| URI Record Type | | | | | | | | | |
| | 0x01, | | | | | | | | /* URI Record Type : http://www.  */ |
| AppData | | | | | | | | | /*Appliction Data : mysimplelink.net */ |
| | 0x6D, | 0x79, | 0x73, | 0x69, | 0x6D, | 0x70, | 0x6C, | 0x65, | |
| | 0x6C, | 0x69, | 0x6E, | 0x6B, | 0x2E, | 0x6E, | 0x65, | 0x74, | |
| | | | | | | | | | } /* End of data */ |

**Figure 14. HTTP URL NDEF Definition**

For a Station application, two NDEF messages are defined:

1. Text Record Type message (not supported in NDEF Maker application previously mentioned) containing the word "WIFI". The purpose of this message is to identify the device as a valid Station and activate NFC Reader/Writer mode on the Access Point device to rewrite the information contained in the RF430CL330H with the corresponding Wi-Fi pairing information.
2. URI Record Type message containing the web server URL based on the assigned IP address. The message is defined as "http://aaa.bbb.ccc.ddd".

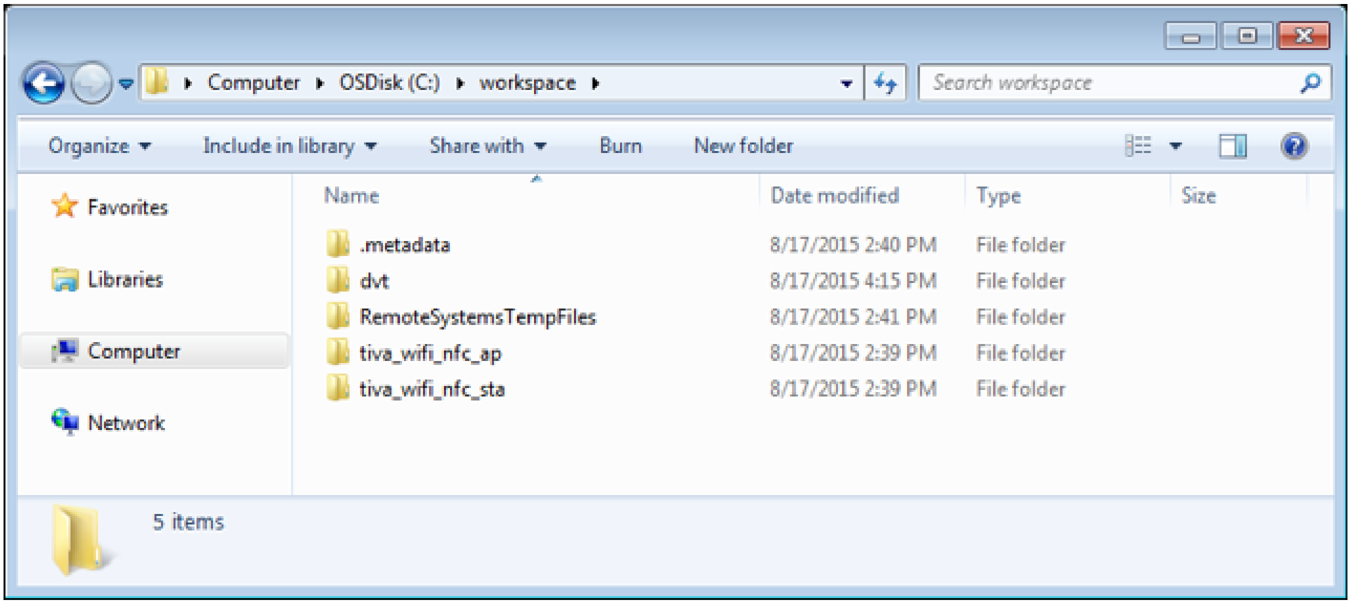
## 5 Software Setup

These tools and software packages are required to build and test Access Point and Station projects:

- Code Composer Studio™ (http://www.ti.com/tool/ccstudio)
- Uniflash for CC3100/CC3200 (http://www.ti.com/tool/uniflash)
- CC3100 SDK v1.1.0 (http://www.ti.com/tool/cc3100sdk)
- TI-RTOS for TIVA v2.14.0.10 (CCS App Center); TivaWare is included.

It is recommend to install these packages in the default location to avoid making any changes in the CCS project. Once the previous tools are installed, follow these steps:

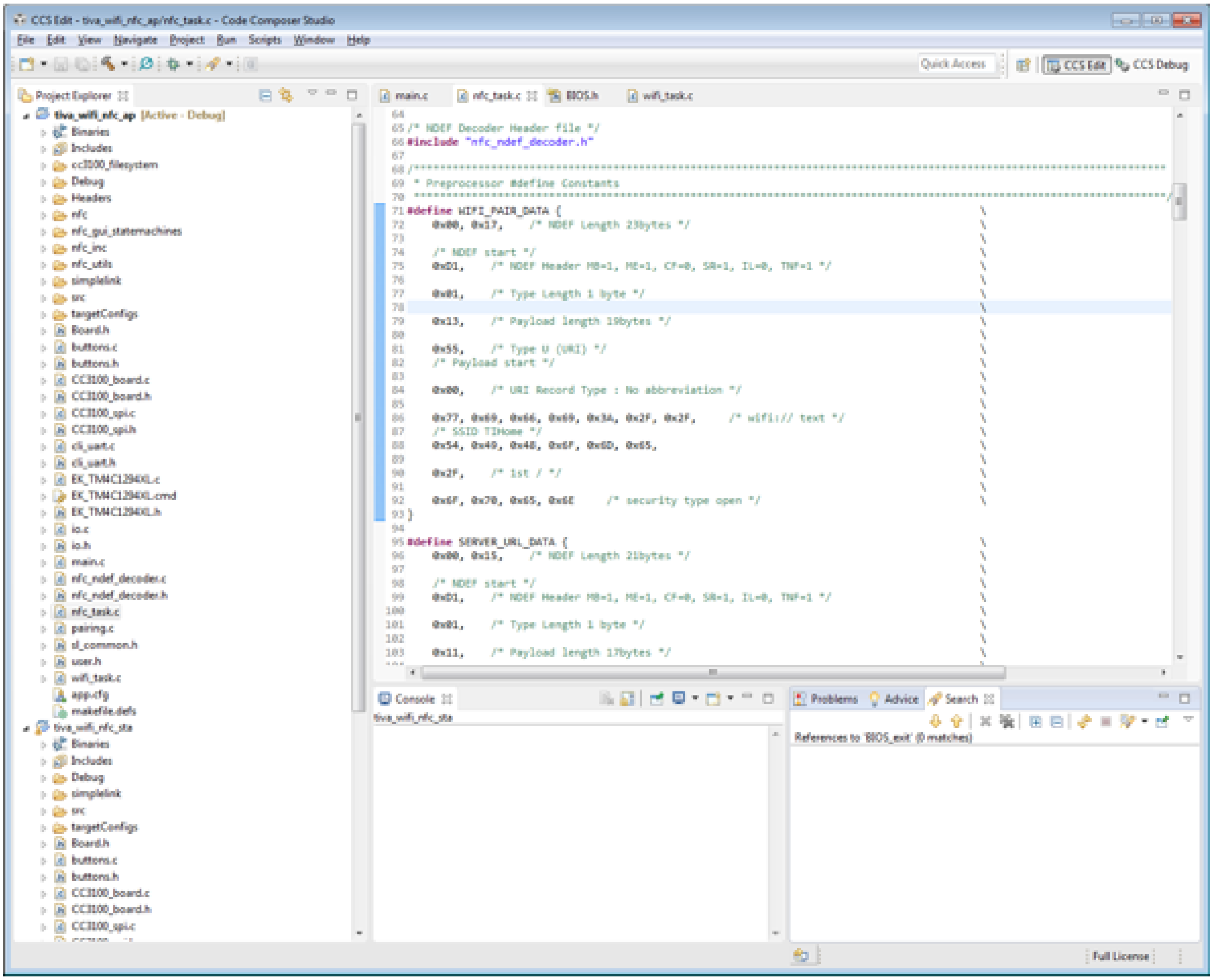
1. Unzip the software release zip file. Place the extracted "wifi_nfc_ap" and "wifi_nfc_sta" directories in your workspace.



| Name | Date modified | Type | Size |
|---|---|---|---|
| .metadata | 8/17/2015 2:40 PM | File folder | |
| dvt | 8/17/2015 4:15 PM | File folder | |
| RemoteSystemsTempFiles | 8/17/2015 2:41 PM | File folder | |
| tiva_wifi_nfc_ap | 8/17/2015 2:39 PM | File folder | |
| tiva_wifi_nfc_sta | 8/17/2015 2:39 PM | File folder | |

**Figure 15. CCS Workspace**

2. Import both projects into CCS.

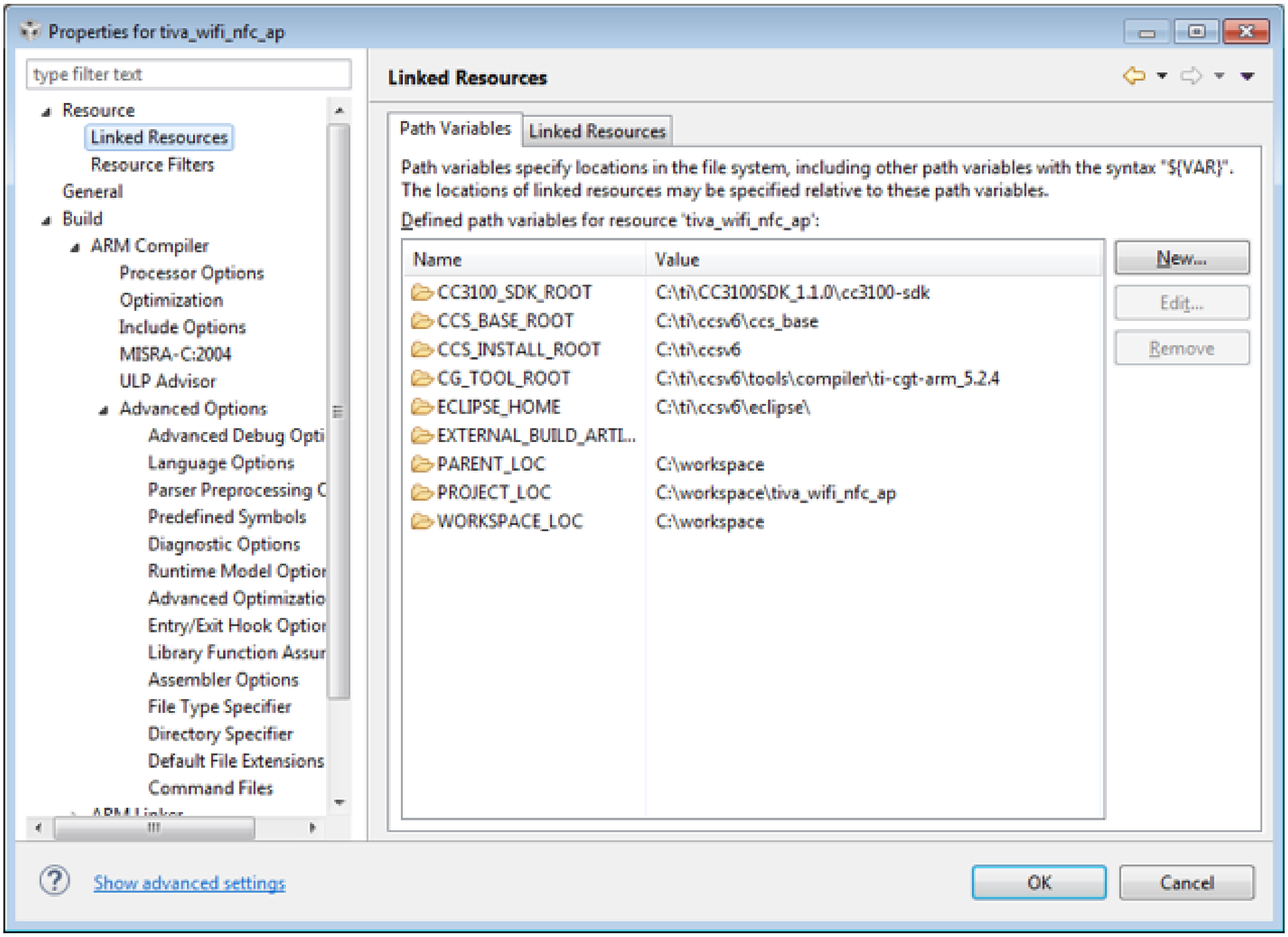


```

/* NDEF Decoder Header file */
#include "nfc_ndef_decoder.h"

/*****************************************************************************************************************
 * Preprocessor #define Constants
 *****************************************************************************************************************/
#define WIFI_PAIR_DATA {                                                                                   \
    0x00, 0x17,     /* NDEF Length 23bytes */                                                            \
                                                                                                         \
    /* NDEF start */                                                                                     \
    0xD1,     /* NDEF Header MB=1, ME=1, CF=0, SR=1, IL=0, TNF=1 */                                    \
                                                                                                         \
    0x01,     /* Type Length 1 byte */                                                                  \
                                                                                                         \
    0x13,     /* Payload length 19bytes */                                                              \
                                                                                                         \
    0x55,     /* Type U (URI) */                                                                        \
    /* Payload start */                                                                                  \
                                                                                                         \
    0x00,     /* URI Record Type : No abbreviation */                                                   \
                                                                                                         \
    0x77, 0x69, 0x66, 0x69, 0x3A, 0x2F, 0x2F,      /* wifi:// text */                                  \
    /* SSID TIHome */                                                                                    \
    0x54, 0x49, 0x48, 0x6F, 0x6D, 0x65,                                                                  \
                                                                                                         \
    0x2F,     /* 1st / */                                                                               \
                                                                                                         \
    0x6F, 0x70, 0x65, 0x6E      /* security type open */                                                \
}

#define SERVER_URL_DATA {                                                                                  \
    0x00, 0x15,     /* NDEF Length 21bytes */                                                            \
                                                                                                         \
    /* NDEF start */                                                                                     \
    0xD1,     /* NDEF Header MB=1, ME=1, CF=0, SR=1, IL=0, TNF=1 */                                    \
                                                                                                         \
    0x01,     /* Type Length 1 byte */                                                                  \
                                                                                                         \
    0x11,     /* Payload length 17bytes */                                                              \
```

References to 'BIOS_exit' (0 matches)

**Figure 16. CCS Projects**

3. Check the "Linked Resources Path Variables" option to confirm those correspond to the actual folders in the current setup.



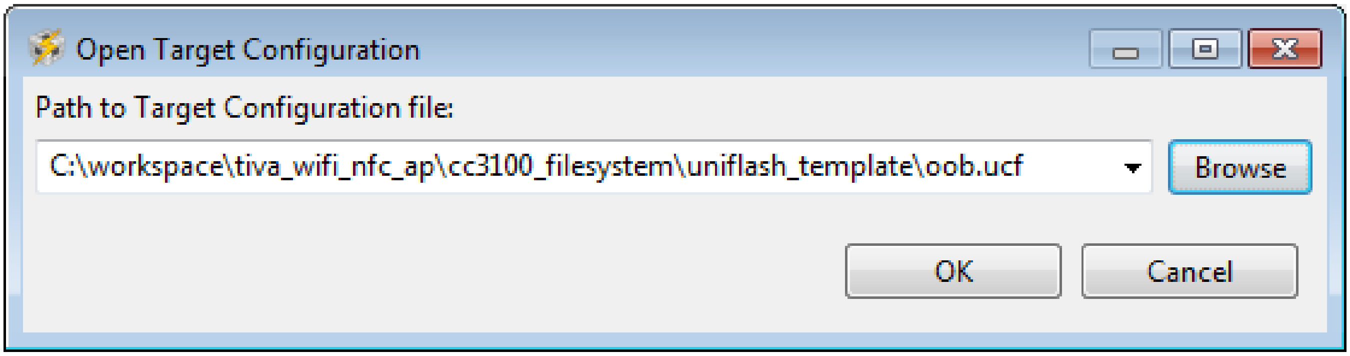
Path variables specify locations in the file system, including other path variables with the syntax "${VAR}". The locations of linked resources may be specified relative to these path variables.

Defined path variables for resource 'tiva_wifi_nfc_ap':

| Name | Value |
|---|---|
| CC3100_SDK_ROOT | C:\ti\CC3100SDK_1.1.0\cc3100-sdk |
| CCS_BASE_ROOT | C:\ti\ccsv6\ccs_base |
| CCS_INSTALL_ROOT | C:\ti\ccsv6 |
| CG_TOOL_ROOT | C:\ti\ccsv6\tools\compiler\ti-cgt-arm_5.2.4 |
| ECLIPSE_HOME | C:\ti\ccsv6\eclipse\ |
| EXTERNAL_BUILD_ARTI... | |
| PARENT_LOC | C:\workspace |
| PROJECT_LOC | C:\workspace\tiva_wifi_nfc_ap |
| WORKSPACE_LOC | C:\workspace |

**Figure 17. Path Variables**

4. Program the CC3100 using the CC31XXEMUBOOST board. The HTML code for this example is included in the "cc3100_filesystem" subdirectory. Program the code into the CC3100 BoosterPack using Uniflash with the project file oob.ucf found in the following directory: "cc3100_filesystem\uniflash_template\oob.ucf".



Path to Target Configuration file: C:\workspace\tiva_wifi_nfc_ap\cc3100_filesystem\uniflash_template\oob.ucf

**Figure 18. Uniflash Target Configuration**

# 6 Demo Execution

For details about the HTTP server demo execution, please check the WIFI Enabled IoT Node with High Performance MCU Reference Design.

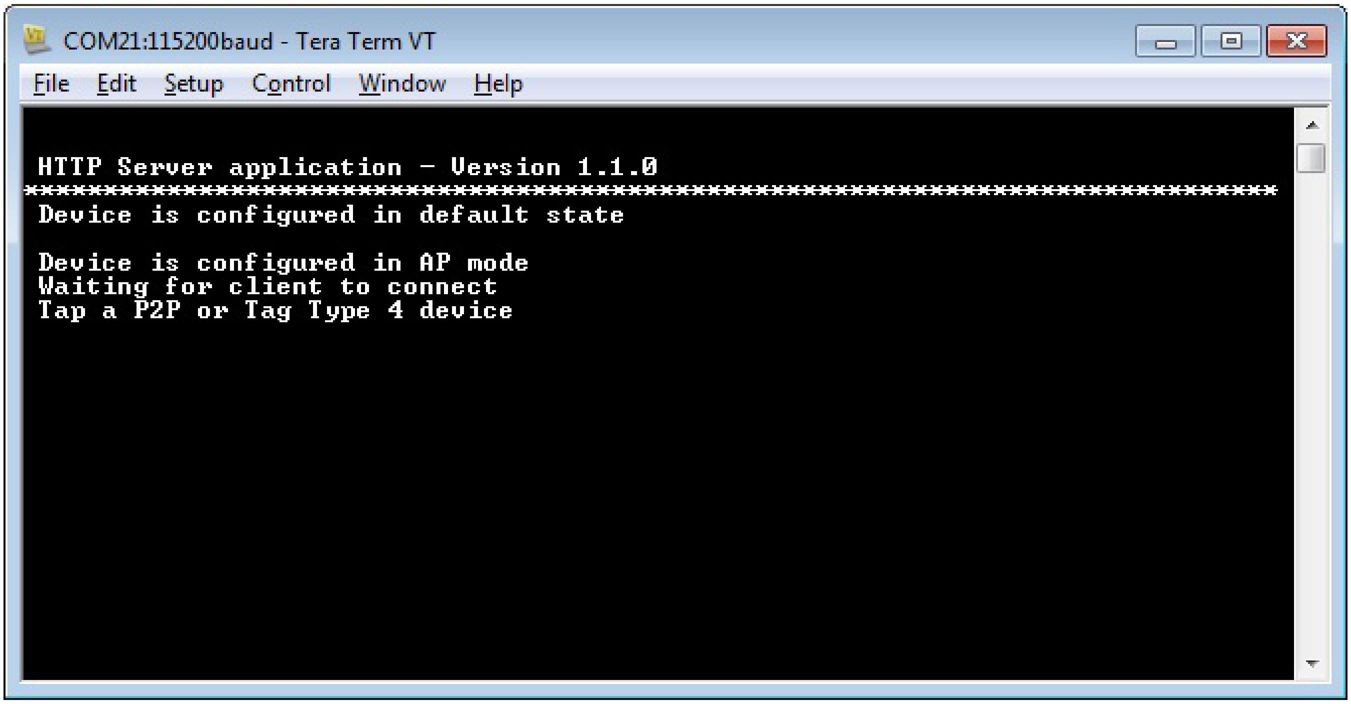
To access the debug port, follow these steps:

1. Connect the EK-TM4C1294XL LaunchPad to the PC using the debug USB port and identify the assigned COM port number in Device Manager.
2. Open a terminal window (like Hyperterminal or TeraTerm) and connect to the Stellaris Virtual Serial Port COM port.
3. Select the Baud Rate as 115200, Data Bits as 8, Parity as none, Stop Bits as 1, and Flow Control as none.

## 6.1 Access Point

When the TM4C Wi-Fi node works as a Wi-Fi access point, the Wi-Fi client is directly connected to the TM4C Wi-Fi node as shown in Figure 4. After device initialization, these are the performed actions:

1. Display information in the debug port.


COM21:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
HTTP Server application - Version 1.1.0
*******************************************************************************
Device is configured in default state

Device is configured in AP mode
Waiting for client to connect
Tap a P2P or Tag Type 4 device
```

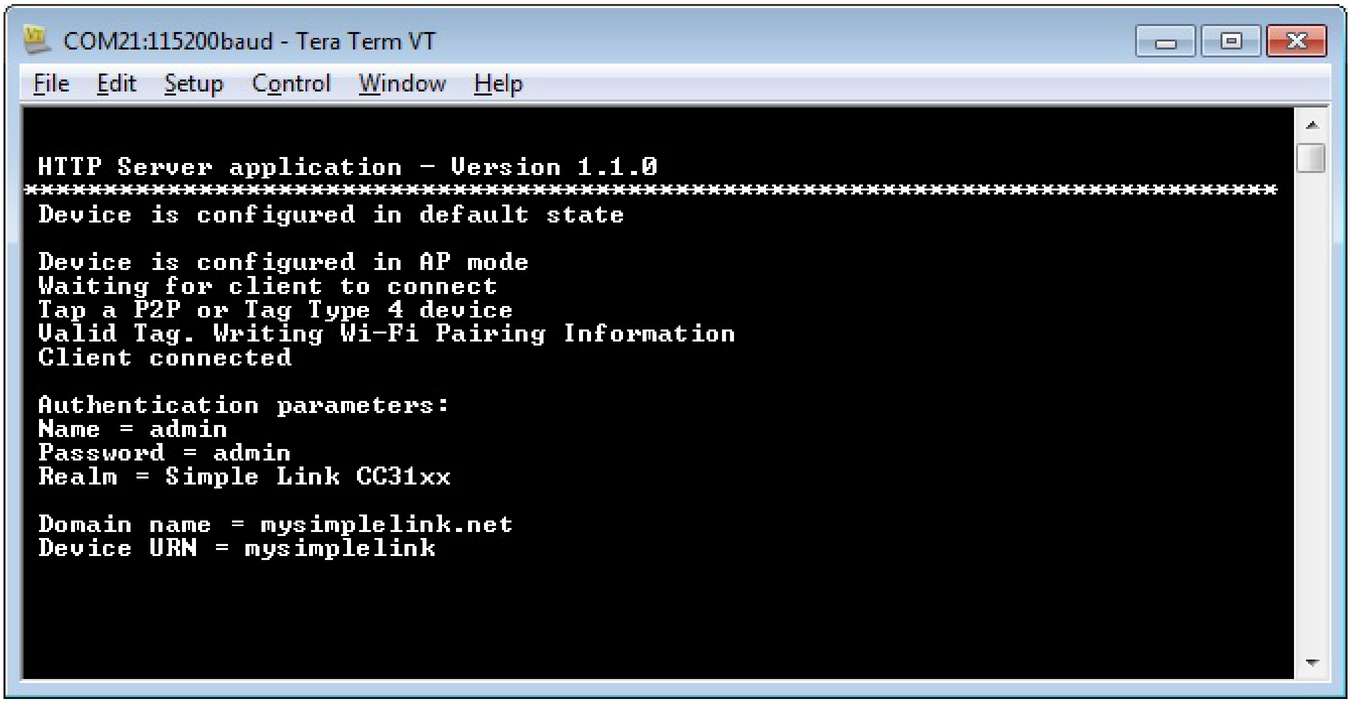
**Figure 19. Access Point Startup Information**

2. Start the Wi-Fi application, start broadcasting the SSID, and wait for a client to be connected.
3. Start the NFC application to look for Type 4 Tags or detect a P2P capable device to share the pairing information.

To connect a client device, follow any of these steps:

1. Using an Android® smartphone, install a Wi-Fi Handover application able to decode the NDEF message format mentioned in Section 4.3.2. Once installed, a simple tap with the TRF7970A BoosterPack antenna will complete the connection to the TI Home network.
2. Enable Wi-Fi connectivity in a computer or smartphone, look for TI Home SSID, and connect to that network.

Once a client is connected, it will display the following information:


COM21:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
HTTP Server application - Version 1.1.0
******************************************************************************
Device is configured in default state

Device is configured in AP mode
Waiting for client to connect
Tap a P2P or Tag Type 4 device
Valid Tag. Writing Wi-Fi Pairing Information
Client connected

Authentication parameters:
Name = admin
Password = admin
Realm = Simple Link CC31xx

Domain name = mysimplelink.net
Device URN = mysimplelink
```

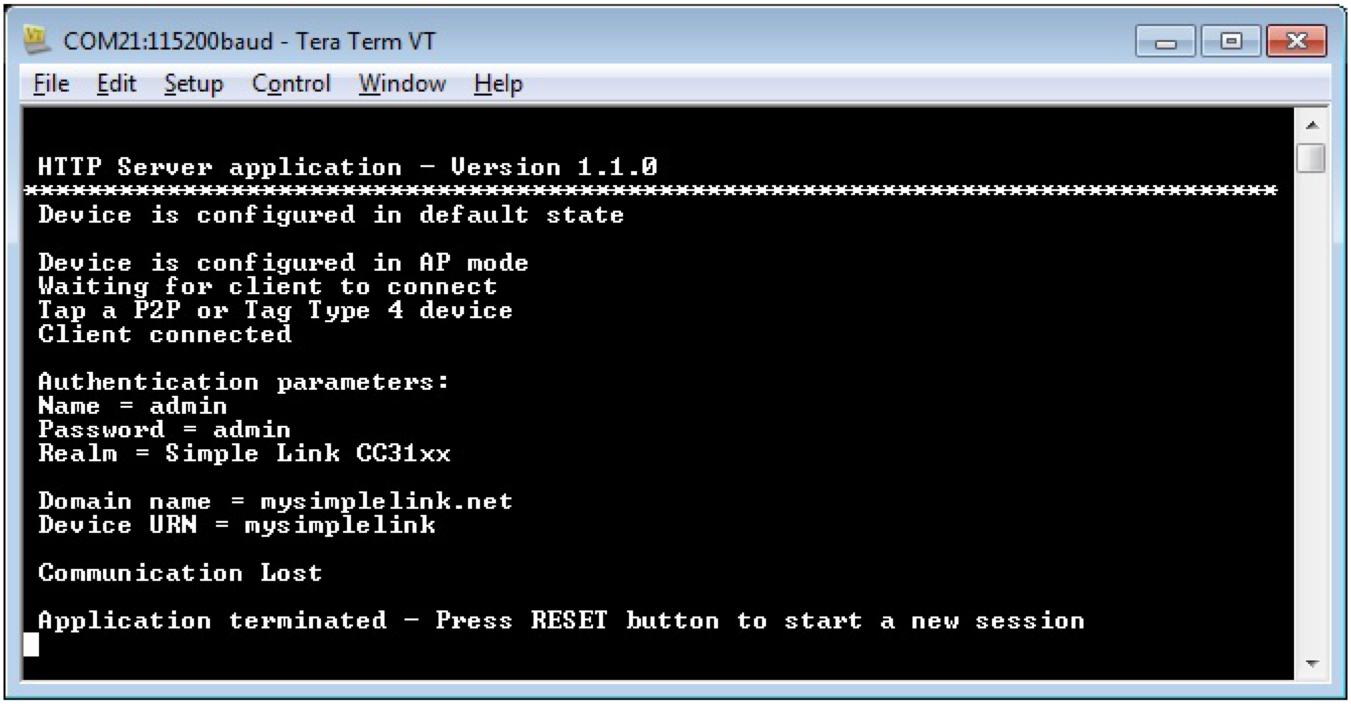
**Figure 20. Access Point With Client Connected**

To open the web server application, there are two options:

1. Open a browser window and enter the following URL: http://www.mysimplelink.net.
2. Perform a second tap with a NFC smartphone. After a client is connected, the NDEF message is replaced by a URL message containing the link mentioned in Figure 14.

In both cases, it is required to enter a login ID and password. The default information is the word "admin" for both.

For this reference system, the application is terminated as soon as the client is disconnected from the Access Point. Once the client is disconnected, it will display the following information:


COM21:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
 HTTP Server application - Version 1.1.0
*******************************************************************************
 Device is configured in default state

 Device is configured in AP mode
 Waiting for client to connect
 Tap a P2P or Tag Type 4 device
 Client connected

 Authentication parameters:
 Name = admin
 Password = admin
 Realm = Simple Link CC31xx

 Domain name = mysimplelink.net
 Device URN = mysimplelink

 Communication Lost

 Application terminated - Press RESET button to start a new session
```

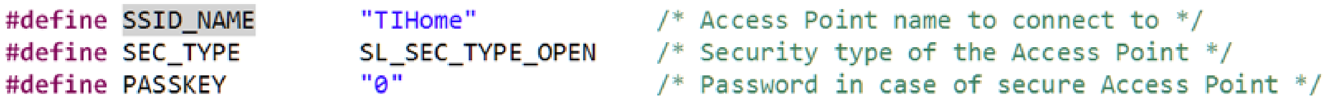
**Figure 21. Access Point Application Terminated**

**NOTE:** Wi-Fi pairing information is defined in two different places. For Wi-Fi setup (CC3100), it is defined in sl_common.h (SSID_AP_MODE, PASSWORD_AP_MODE, and SEC_TYPE_AP_MODE). For NFC setup (TRF7970A), it is defined at nfc_task.c (WIFI_PAIR_DATA). Both definitions need to be updated if a change in pairing information is required. A WPA2 example is provided by removing WIFI_OPEN_SECURITY from compiler predefined symbols.

## 6.2 Station

When the TM4C Wi-Fi node works as a Wi-Fi station, both the TM4C Wi-Fi node and the Wi-Fi client (laptop, smartphone) need to connect to the same network as shown in Figure 8 (an external router or access point is required). There are two options to connect the station to the desired network:

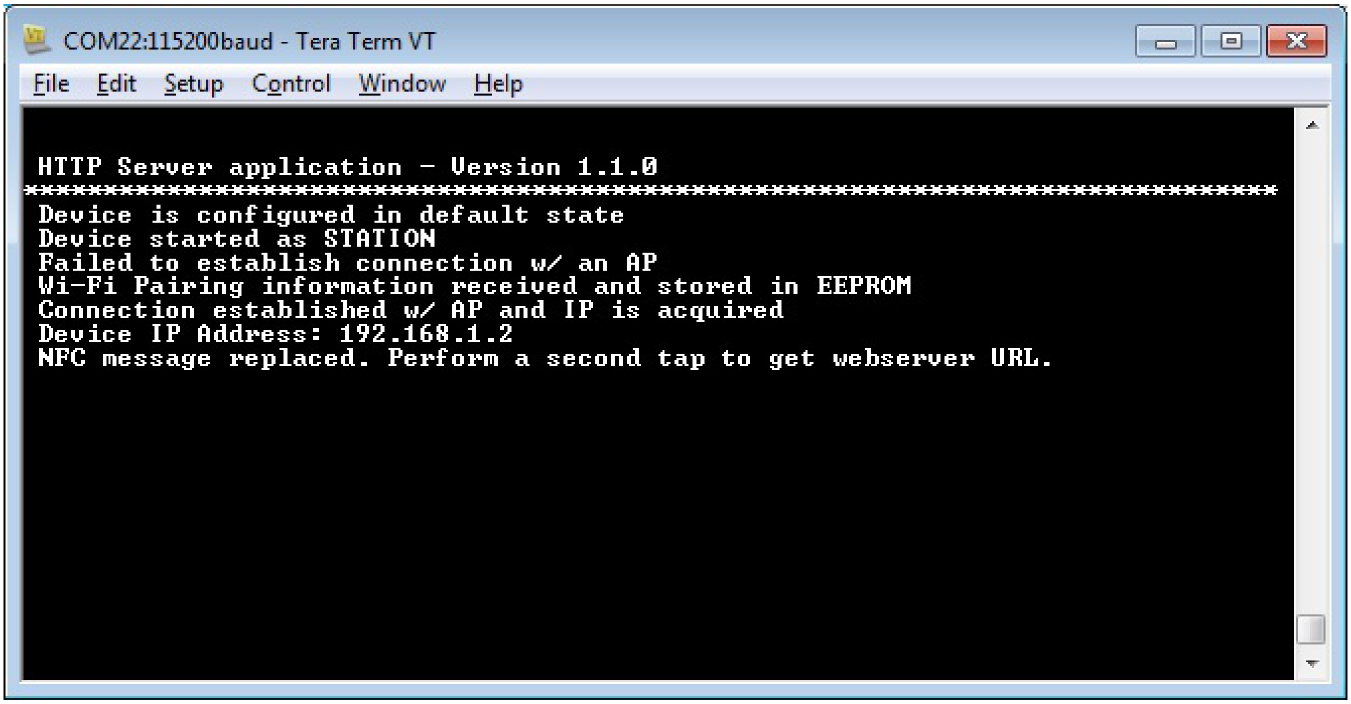
1. Hard coded pairing information in Flash. To enable this mode, define the symbol "USE_HARDCODED_SSID" in the compiler options. After that, define the desired parameters in sl_common.h header file. These are the default values:



```
#define SSID_NAME        "TIHome"           /* Access Point name to connect to */
#define SEC_TYPE         SL_SEC_TYPE_OPEN   /* Security type of the Access Point */
#define PASSKEY          "0"                /* Password in case of secure Access Point */
```

**Figure 22. Wi-Fi Network Configuration Code**

Once the device is connected to the network, observe the following information:



```
HTTP Server application - Version 1.1.0
********************************************************************************
Device is configured in default state
Device started as STATION
Failed to establish connection w/ an AP
Wi-Fi Pairing information received and stored in EEPROM
Connection established w/ AP and IP is acquired
Device IP Address: 192.168.1.2
NFC message replaced. Perform a second tap to get webserver URL.
```

**Figure 23. Station Connected**

In this case, the NFC interface is used to provide the web server URL (IP address). A tap with a smartphone connected to the same network will be enough to open the web server page.

2. Receive pairing information through NFC. To enable this mode, remove the symbol "USE_HARDCODED_SSID" in the compiler options. For this purpose, use an NFC writer device (for example, a smartphone with a Tag Writer application) and rewrite the RF430CL330H with an NDEF message following the format mentioned in Section 4.3.2.
3. After the device is connected to the network and an IP address has been assigned, RF430CL330H content is replaced with an NDEF message containing the web server URL based on the assigned IP address. A second tap with a smartphone connected to the same network will open a browser window to access the web server interface. Both login and password are set to "admin".

**NOTE:** The Access Point system described in this document has the NFC writer capability, and it can be use to pair the Station device. The default NFC message in the Station device is the word "WIFI". This word is used as an authentication method by the Access Point to enable writer mode and send its pairing information to the Station. A simple tap of the antennas of the TRF7970A board in the Access Point device and the RF430CL330H board in station device will cause the two devices to connect to each other. The "Connection Established" messages will be displayed in both UART interfaces.

## 7 Resources

To download the resource files for this reference design, see the product page at http://www.ti.com/tool/TIDM-TM4C129XNFC.

## 8 References

1. Texas Instruments, TivaWare for C Series: http://www.ti.com/tool/SW-TM4C
2. Texas Instruments, *Stellaris® In-Circuit Debug Interface (ICDI) and Virtual COM Port*, Driver Installation Instructions (SPMU287)
3. Texas Instruments, TI-RTOS: http://www.ti.com/tool/ti-rtos
4. Texas Instruments, SimpleLink SDK: http://www.ti.com/tool/cc3100sdk
5. Texas Instruments, Near Field Communication (NFC) Card Emulation Reference Design: http://www.ti.com/tool/TIDM-NFC-CE

## 9 About the Author

**ENRIQUE LIZÁRRAGA** is a Software Design Engineer at Texas Instruments for TM4C family of high performance microcontrollers. For over 10 years, Enrique has designed and developed embedded software for automotive and communication applications. Enrique holds a B.S. degree on digital systems and communication engineering from UACJ, México.

# Revision History

| Changes from Original (September 2015) to A Revision | Page |
|---|---|
| • Changed from "Resource Explorer in CCS" to "CCS App Center" | 17 |

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

## IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers (“Buyers”) who are developing systems that incorporate TI semiconductor products (also referred to herein as “components”). Buyer understands and agrees that Buyer remains responsible for using its independent analysis, evaluation and judgment in designing Buyer’s systems and products.

TI reference designs have been created using standard laboratory conditions and engineering practices. **TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.** TI may make corrections, enhancements, improvements and other changes to its reference designs.

Buyers are authorized to use TI reference designs with the TI component(s) identified in each particular reference design and to modify the reference design in the development of their end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, IS GRANTED HEREIN, including but not limited to 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.

TI REFERENCE DESIGNS ARE PROVIDED "AS IS". TI MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. TI DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT, QUIET POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO TI REFERENCE DESIGNS OR USE THEREOF. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY BUYERS AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON A COMBINATION OF COMPONENTS PROVIDED IN A TI REFERENCE DESIGN. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF TI REFERENCE DESIGNS OR BUYER’S USE OF TI REFERENCE DESIGNS.

TI reserves 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 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 for TI components 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.

Reproduction of significant portions of TI information in TI data books, data sheets or reference designs 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.

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 that anticipate dangerous failures, monitor failures and their consequences, lessen the likelihood of dangerous failures 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 Buyer’s 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 an agreement specifically governing such use.

Only those TI components that 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 that have ***not*** been so designated is solely at Buyer's risk, and 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.

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