TI Designs
TM4C1294x Wi-Fi Enabled IoT Node

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-TM4C129XWIFI	Tool Folder Containing Design Files
TM4C1294NCPDT	Product Folder
CC3100	Product Folder
CC3100BOOST	Tool Folder
EK-TM4C1294XL	Tool Folder

TI E2E™ Community

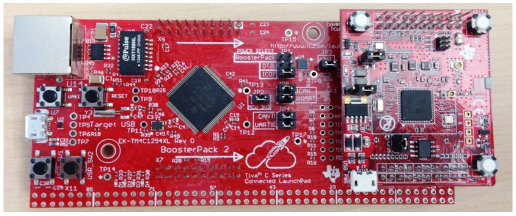
ASK Our E2E Experts
WEBENCH® Calculator Tools

Design Features

- TM4C1294 MCU Integrated With CC3100 Network Processor as Wi-Fi HTTP Server
- Wi-Fi HTTP Server Works as Access Point or Station
- Wi-Fi Stack Runs on CC3100
- TM4C1294 Functions as Host Processor Handling Wi-Fi Callback Functions and Control Functions
- HTML Code Enables User to Remotely Control Operation of TM4C1294 Through Web Browser
- Operation on TM4C1294XL LaunchPad Includes LED Toggling, Internal Temperature Reading, and Button Press Recording
- Software Designed to Work on EK- TM4C1294XL (Connected LaunchPad) and CC3100 BoosterPack for Code Composer Studio™ Software
- TI-RTOS Used for Task Scheduling

Featured Applications

- Factory Automation and Control
- Building Automation
- Smart Grid and Energy



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

1 System Description

This system example illustrates how to build a Wi-Fi node with the TM4C1294 high-performance microcontroller and CC3100 network processor. A user can remotely control the TM4C1294 through a web browser. The functionalities used for this project can be extended in real applications requiring the high performance features of the TM4C device and connection to a wireless domain.

The software accompanying this design works on an EK-TM4C1294XL LaunchPad and CC3100 BoosterPack. For remote access to the Wi-Fi node, the user must program the included *.html* files to the CC3100 using the TI Uniflash tool.

1.1 TM4C1294NCPDT

The TM4C1294NCPDT is a 120-MHz, high-performance microcontroller (MCU) 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 such as a memory controller and a USB 2.0 High-speed digital interface. With the integration of a number of low-to-mid speed serials (up to 4 million samples per second (MSPS)), a 12-bit ADC, and motion control peripherals, the TM4C1294CPDT microcontroller makes for a unique solution for a variety of applications ranging from industrial communication equipment to smart energy or smart grid applications. Figure 1 shows the block diagram for the TM4C1294NCPDT MCU.

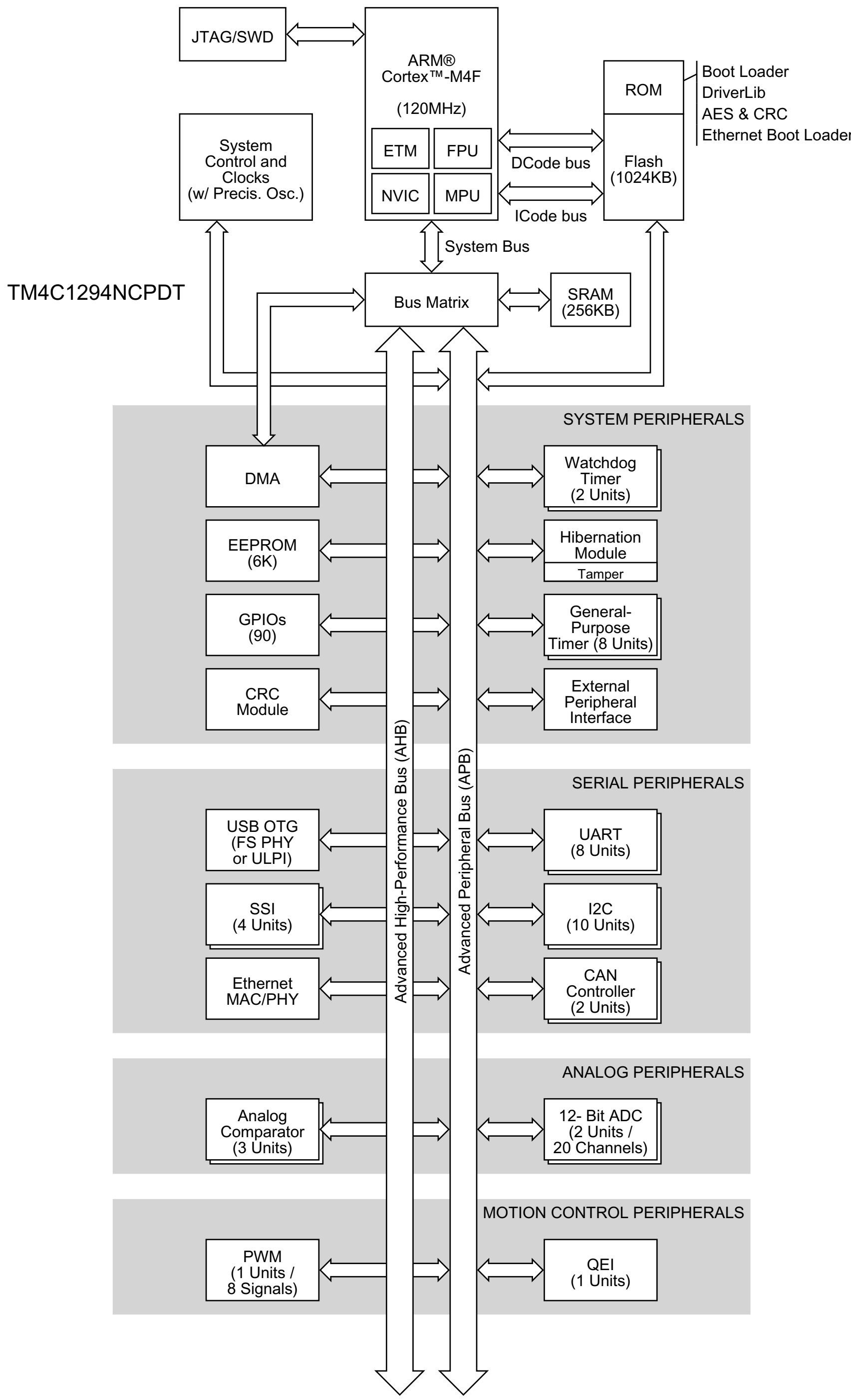


Figure 1. TM4C1294NCPDT Microcontroller High-Level Block Diagram

1.2 CC3100

The CC3100 Wi-Fi network processor subsystem features a Wi-Fi Internet-on-a-chip™ integrated circuit 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 device 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.

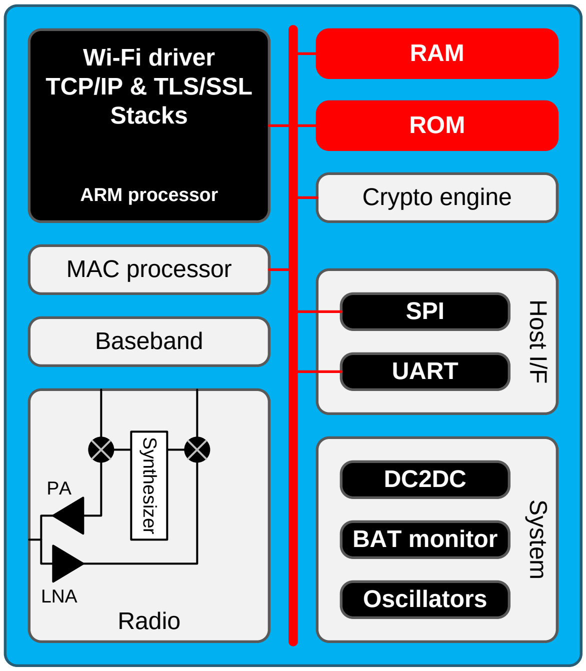


Figure 2. CC3100 Hardware Overview

1.3 TM4C129/CC3100 Interface

Figure 3 shows the interface between the TM4C1294 and CC3100 devices. The TM4C1294 can reset CC3100 through the reset pin. The TM4C1294 can also enable and disable the CC3100 by controlling the nHIB pin. When the CC3100 requires service, it toggles the CC_IRQ signal to notify the TM4C1294 device. The TM4C1294 then transfers information with the CC3100 over the serial peripheral interface (SPI) bus. The TI SimpleLink library driver is used to process the message and control the communication. The user can process the request from the CC3100 device as http callback events without the knowledge of the SPI messaging format.

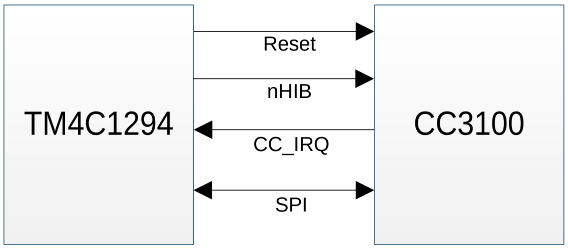


Figure 3. TM4C129 and CC3100 Interface Overview

2 Functional Block Diagram

The TM4C Wi-Fi node is configurable in two modes: Access Point and Station.

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

The Wi-Fi web client is directly connected to the TM4C Wi-Fi node when the TM4C Wi-Fi node is configured as a Wi-Fi access point (see Figure 4). This configuration does not allow 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.

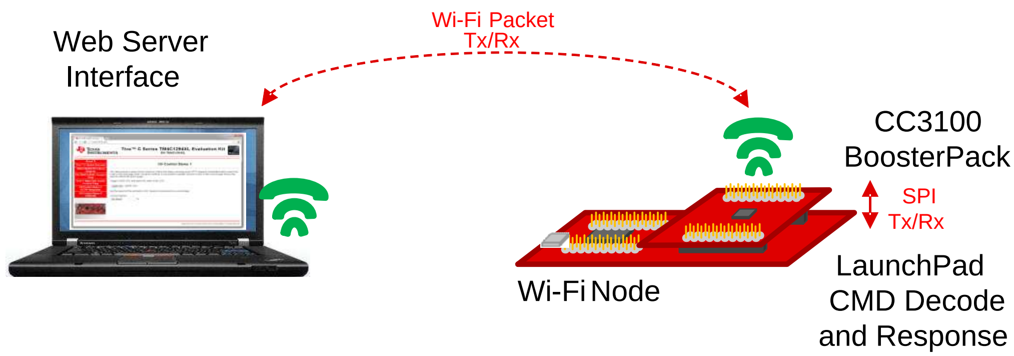


Figure 4. General Setup and Dataflow for Wi-Fi Node Access Point

2.2 TM4C Wi-Fi Node as Wi-Fi Station

When configured as a station, the Wi-Fi node connects with the existing network and acquires an IP address. The Wi-Fi node is accessible using an Internet browser such as Microsoft Internet Explorer®, Google Chrome™, or Mozilla Firefox® as a web server. The web server enables 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 required.

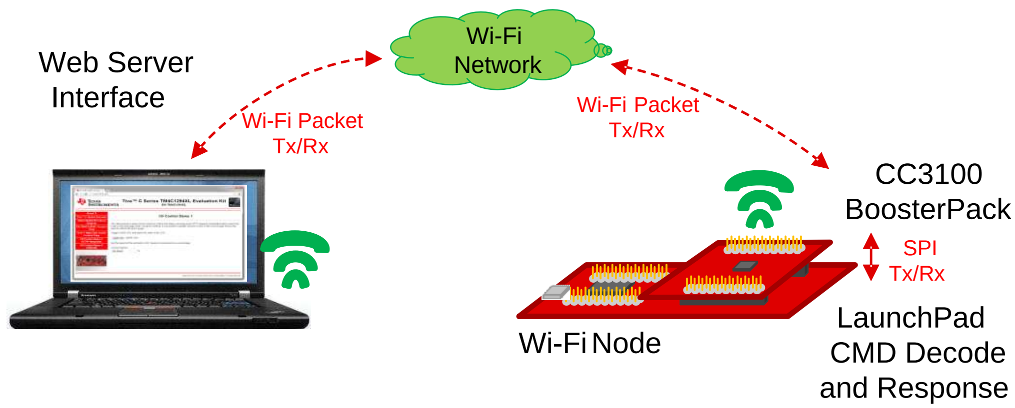


Figure 5. General Setup and Dataflow for Wi-Fi Node Station

3 Getting Started Hardware

The hardware used in this example is the EK-TM4C129XL Connected LaunchPad and the CC3100 BoosterPack. The EK-TM4C129XL Connected LaunchPad board is connected to the SimpleLink CC3100 BoosterPack board through the BoosterPack connector 1 on EK-TM4C129XL. Table 1 shows the signal mapping.

Table 1. Signal Mapping

BOOSTERPACK CONNECTOR	CC3100 BOOSTERPACK	TM4C1294 CONNECTED LAUNCHPAD
P1.1 (1)	3.3 V	3.3 V
P1.2	Open	PE4
P1.3	CC_UART1_TX	PC4_U7RX
P1.4	CC_UART1_RX	PC5_U7TX
P1.5 (1)	CC_nHIB	PC6
P1.6	Open	PE5
P1.7 (1)	CC_SPI_CLK	PD3_SSI2CLK
P1.8	Open	PC7
P1.9	Test_3	PB2
P1.10	FORCE_AP	PB3
P2.1 (1)	GND	GND
P2.2 (1)	CC_IRQ	PM3
P2.3 (1)	CC_SPI_CS	PH2
P2.4	Open	PH3
P2.5 (1)	MCU_RESET_IN	RESET
P2.6 (1)	CC_SPI_DIN	PD1_SSI2XDAT0
P2.7 (1)	CC_SPI_DOUT	PD0_SSIXDATA1
P2.8	Test_63	PN2
P2.9	Test_64	PN3
P2.10	Test_18	PP2
P3.1 (1)	5V	5V
P3.2 (1)	GND	GND
P3.3	Open	PE0
P3.4	Open	PE1
P3.5	Open	PE2
P3.6	Open	PE3
P3.7	Open	PD7
P3.8	Open	PA6
P3.9	Open	PM4
P3.10	Open	PM5
P4.1	Test_29	PF1
P4.2	Test_30	PF2
P4.3	Open	PF3
P4.4	CC_URT1_CTS	PG0
P4.5	CC_UART1_RTS	PL4
P4.6	Open	PL5
P4.7	CC_NWP_UART_TX	PL1
P4.8	CC_WL_UART_TX	PL2
P4.9	CC_WLRS232_RX	PL3
P4.10	CC_WLRS232_TX	PL4

(1) Denotes that this pin is required for this design.

The block diagram in Figure 3 shows the TM4C1294/CC3100 interface. The TM4C1294 device controls the reset and enable/disable switches of the CC3100 device. When the CC3100 device requires service, it toggles the “CC_IRQ” pin to notify the TM4C1294. The TM4C129 reads the CC3100 request through the SPI port and performs the rest of the communication. The following TM4C1294 I/O pins are used as the interface to CC3100:

- SSI2 in three-pin SPI mode (CLK, MOSI, and SOMI)
- GPIO output pin PH2 is manually controlled as SPI CS for CC3100
- GPIO input interrupt on pin PM3 is enabled to capture the request from CC3100
- GPIO output pin PC6 is configured to enable or disable CC3100

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

- Timer 1 to generate 1-ms system tick
- Timer 2 to toggle LED with variable periods
- Timer 0 to sample buttons and internal temperature sensor every 10 ms
- GPIO output pin PN0 to toggle LED
- 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 a baud rate of 115200 as a diagnostic port

4 Getting Started Software

Figure 6 shows the architecture of the TM4C1294 software. The TI SimpleLink library driver processes the message between the TM4C1294 and CC3100 devices. Use TI's real time operating system (RTOS) to schedule the processing. Use the TI TivaWare ™ library to control the hardware on the TM4C1294 device.

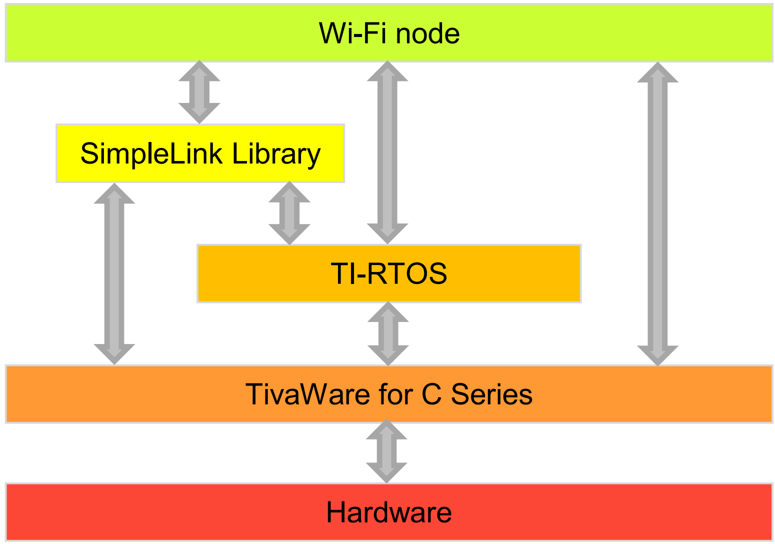


Figure 6. TM4C1294 Software Architecture Block Diagram

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

- CLK0: 1-ms system tick generated by timer 1
- HWI0: GPIO interrupt on PM3 for capturing the CC3100 request and setting semaphore
- HWI1: Timer 2 interrupt with variable period for toggling the LED
- HWI2: 10-ms timer 0 interrupt for sampling buttons and internal temperature sensor
- TASK0: Perform all tasks requested by CC3100 (controlled by the semaphore set in HWI 0)

The software was developed as an extension of the CC3100 SDK examples. The SimpleLink library itself is not device specific. All device specific information must be included in the application program interface (API) functions defined in the following files:

- *board.c*: APIs for setting up hardware on the board
- *spi.c*: APIs SPI communication
- *cli_uart.c*: APIs for the universal asynchronous receiver and transmitter (UART) port (debugging)

Two additional files (*io.c* and *buttons.c*) contain APIs for controlling the peripherals used for the demo. The Wi-Fi processing functions are included in the file main.c. The software release is in the *.zip* file format. The Access Point Mode and Station Mode are in separate projects for TI's Code Composer Studio™ (CCS) software.

5 Demo Installation

To rebuild the demo the user must install the following TI tools and software packages.

- TM4C1294 Connected LaunchPad
- CC3100 BoosterPack
- Code Composer Studio (CCS) v6.0.1 or above
- Uniflash for CC3100/CC3200 v3.2.0
- CC3100 SDK 1.1.0
- TI-RTOS for TIVA v2.12.01.33 (TivaWare v2.1.0.12573 is included)

Unzip the software release *.zip* file. Place the extracted “wifi_node_ap_tirtos” and “wifi_node_station_tirtos” directories under the “EK_TM4C1294XL” subdirectory in the TI-RTOS installation (see Figure 7).

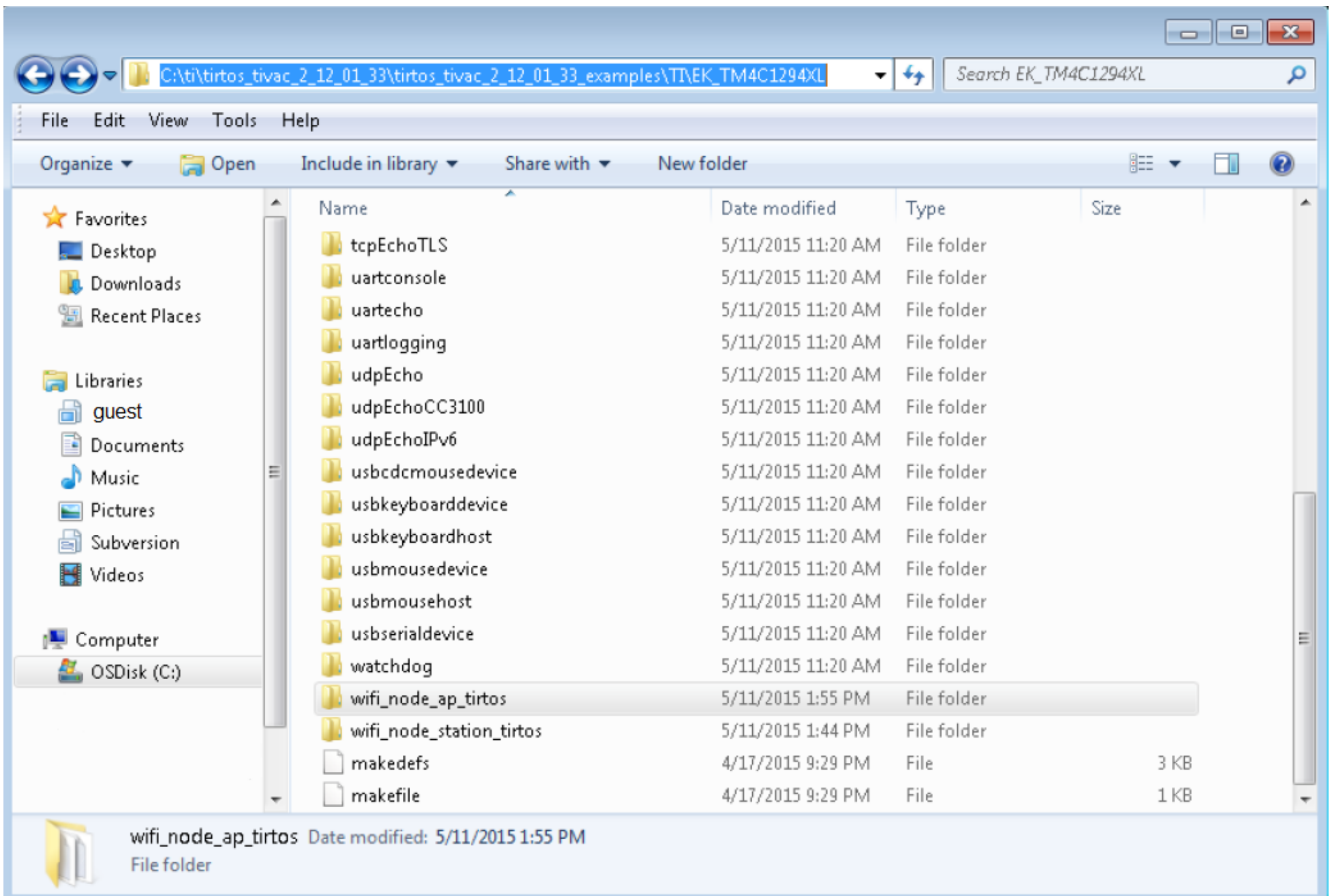


Figure 7. Placement of Extracted Directories

Figure 8 shows the details of the project directory. The http callback function is in *main.c* and the *wifi_node_ap.cfg* is the TI-RTOS configuration file for the project. The *sl_common.h* contains information about the service set identifier (SSID) and security key for the access point.

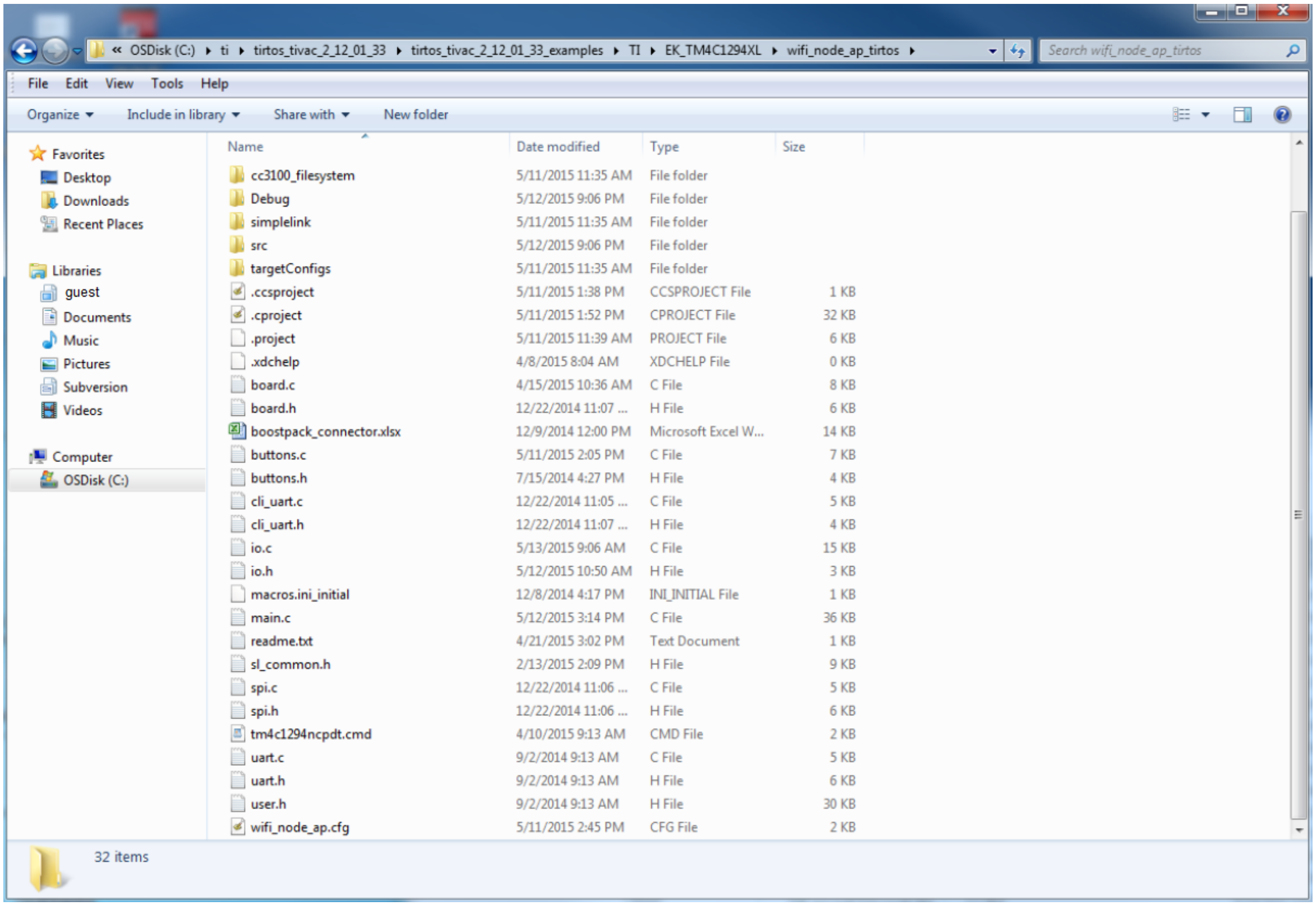


Figure 8. Content of “wifi_node_ap_tirtos” Directory

Figure 9 shows the linked path for compiling the project. The paths in Figure 9 are obtained using the default locations for the software installation. If the user does not install the software to the default location, then the linked path must be changed to match the actual installation location.

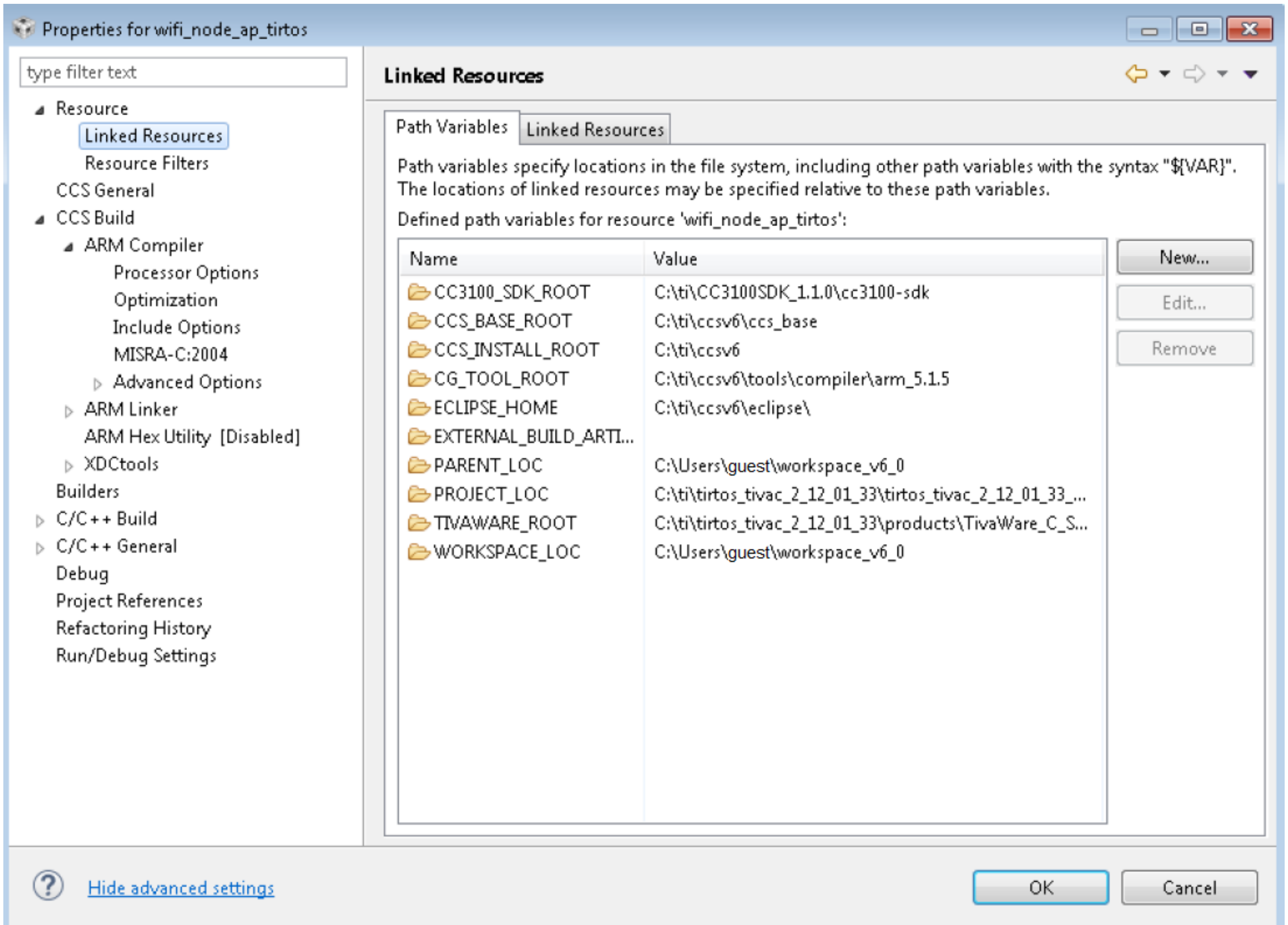


Figure 9. Linked Paths for Project

The HTML code for this example is also included in the "cc3100_filesystem" subdirectory (see Figure 8). 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".

To generate the HTTP "get" event with the CC3100 device, the "Ajax" method is required. The following code passage shows an example of the "Ajax" function for generating an HTTP "get" event:

```
function getTokenValue(paramPage, tokenId, successFn, failFn)
    {
        $.ajax({
                "type": "GET".
                "url": paramPage,
                "cache": false,
                "dataType": "html"
        })
        // define callback for get request success
        .done( function(getdata, status, xhr)
        {
           successFn($(getdata).filter('#'+tokenId).text());
        })
        .fail(function(jqxhr, settings, exception)
        {
           failFn();
        });
    }
```

The following function uses the getTokenValue() function to generate an HTTP "get" event to acquire the updated number of button 1 presses.

```
function get_button1()
{
    getTokenValue
    (  'io_http_params.htm', 'button1_count_status',
       // successFn callback, val is the token value
       function(val)
       {
           document.getElementByld("button1_count").value = val;
       },
       // failFn callback
       function()
       {
          _count = _count + 1;
       }
    );
}
```

In the get_button1() function, the getTokenValue function is called to generate an HTTP "get" event of "button1_count_status" (defined in a separate .html file *io_http_params.htm*). The returned value is saved in variable "val". Then "val" is used to update the display of the "button1_count" window in the website. View further details in the released source files.

6 Demo Execution

The TM4C Wi-Fi node can work as an access point or a station. There are separate CCS projects for each mode. The website is the same for both modes. Figure 10 shows the client website.

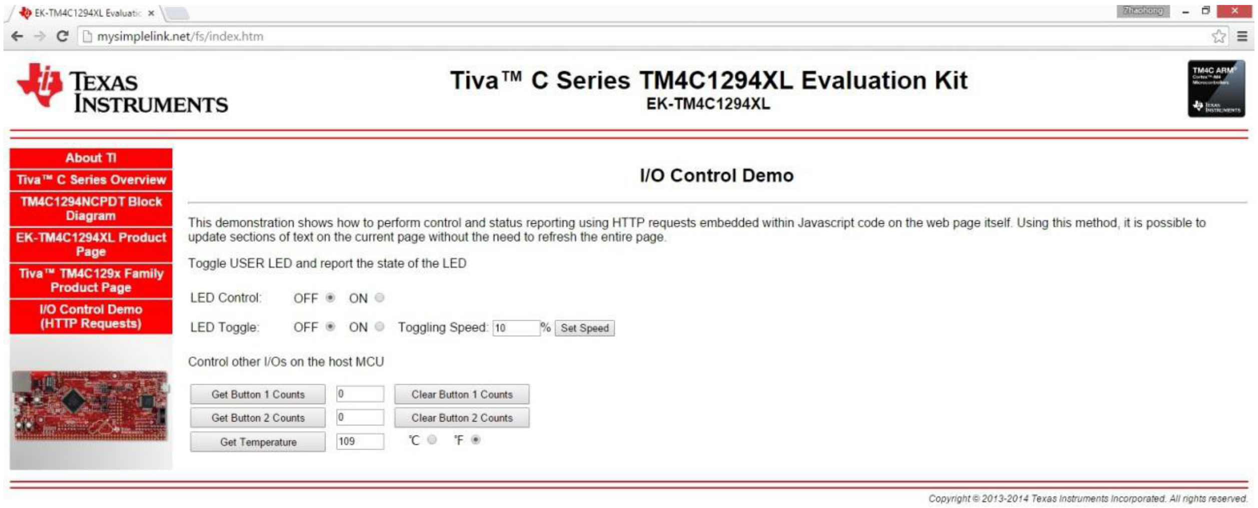


Figure 10. Website for Wi-Fi Node

The “LED Control” field determines the state of LED1 on the EK-TM4C129XL board. The “LED Toggle” field determines if LED1 is at a steady state or toggling between “ON” and “OFF”. At 1% of the toggling speed, LED1 switches at a period of 2 seconds. At 100% of the toggling speed, LED1 switches at a period of 20 milliseconds. LED1 remains “OFF” if the LED control is “OFF”, regardless of the “LED Toggle”.

The website can also request the counts of button presses and the reading from the on-chip temperature sensor. The counts of button presses can be cleared from the website. The temperature readings can be displayed in Celsius or Fahrenheit.

6.1 Access Point

When the TM4C Wi-Fi node functions as a Wi-Fi access point, the Wi-Fi client can connect directly to the TM4C Wi-Fi node, as Figure 4 shows. For this selected configuration there is no internet connection. Access the TM4C Wi-Fi node website by searching for “mysimpleLink.net” in a web browser. The login ID and password are all “admin”.

6.2 Station

When the TM4C Wi-Fi node functions as a Wi-Fi station, both the TM4C Wi-Fi node and the Wi-Fi client must be connected to the Internet (see Figure 5). After connecting the Internet, the TM4C Wi-Fi node dynamically acquires an IP address. This information outputs on the diagnostic UART port. Access the TM4C Wi-Fi node website by searching for this IP address from a web browser. The login ID and password are still all “admin”.

7 Software and Resource Files

To download the software files and resource files for this reference design, please visit the following website: http://www.ti.com/tool/TIDM-TM4C129XWIFI.

8 References

1. Texas Instruments, *Driver Installation Instructions SPMU287B – August 2012 – Revised January 2014* , Driver Installation Instructions (SPMU287)

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