TI Designs
Secure IoT Gateway Reference Design for Bluetooth® Low Energy, Wi-Fi® and Sub-1 GHz Nodes

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-TM4C129XGATEWAY	Design Folder
TM4C123GH6PM	Product Folder
TM4C129ENCPDT	Product Folder
DRV8833	Product Folder
CC3100	Product Folder
CC2650	Product Folder
EK-TM4C123GXL	Tools Folder
EK-TM4C129EXL	Tools Folder
DRV8833 EVM	Tools Folder
CC3100 BoosterPack	Tools Folder
CC2650EM	Product Folder
TRF7970A	Product Folder
RF430CL330H	Product Folder
EM Adapter BoosterPack	Tools Folder

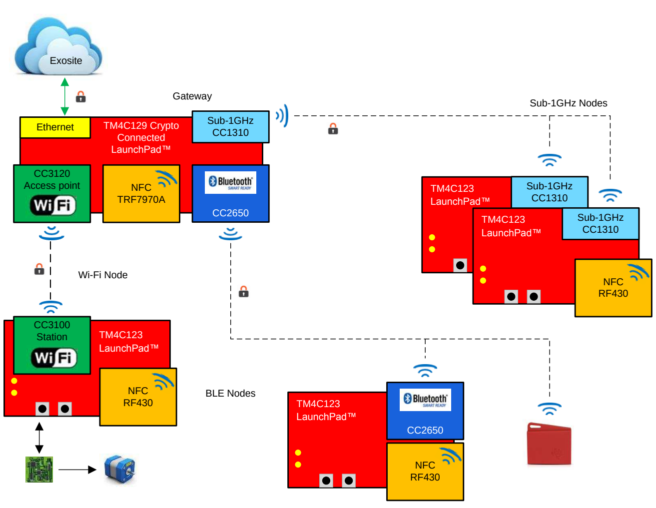
TI E2E™ Community — ASK Our E2E Experts

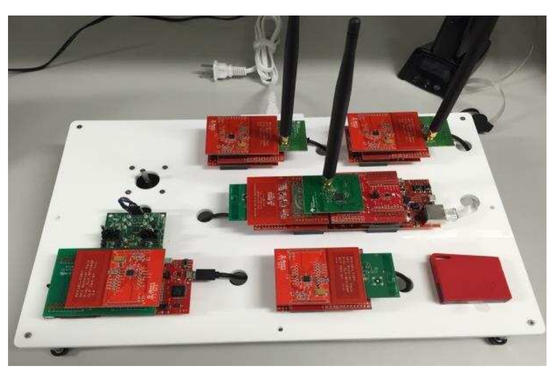
Design Features

- Exosite-Based Secure Cloud Connected Mid-Range Mulit-Protocol IoT Gateway Solution Using TM4C129Ex MCU, Which Connects Wi-Fi, BLE, and Sub-1GHz-Based Nodes to Cloud
- Supported Nodes Include Wi-Fi-Based Stepper Motor Control, BLE Sensor Tag, BLE Slave Node, and Sub-1GHz Slave Nodes
- Connection Between Nodes and Gateway Using NFC-Based Secure Out-of-Band Pairing
- Secure Data Communication Between Nodes and Gateway Using Hardware Crypto Blocks
- Secure Cloud Connection Using TI-RTOS NDK and wolfSSL Stack
- Modular Software Designed to Work on EK-TM4C129EXL (Crypto Connected LaunchPad™), EK-TM4C123GXL (Tiva LaunchPad), CC3100, CC2650, CC1310, and TRF7970A for Code Composer Studio™
- TI-RTOS Used for Task Scheduling

Featured Applications

- Industrial Application and Automation
- Smart Grid and Energy
- Precision Motion Control
- Test and Measurement
- Building Automation and Industrial IoT





All trademarks are the property of their respective owners.

1 System Description

This TI Design demonstrates the application of a TM4C129 high-performance microcontroller (MCU) as an IoT gateway securely connected to the cloud. This gateway system is capable of connecting to different wireless nodes like BLE, Wi-Fi, and Sub-1GHz and also enables their connectivity to the cloud. This demo features a stepper motor connected to the Wi-Fi node based on the TM4C123 and CC3100, a simple BLE node based on the TM4C123 and CC2650, a BLE SensorTag, and two Sub-1GHz nodes based on the TM4C123 and CC1310. This demo uses the services of Exosite as a cloud platform so that all the nodes connected to the gateway and the gateway itself can be controlled from an Exosite Dashboard GUI. The objective of this application demo is to provide a jumpstart to customers in creating their own IoT projects with TI's low cost MCUs and connectivity devices portfolio; all of these are easy to prototype and realize using TI’s LaunchPad and BoosterPack™ ecosystem.

This TI design was presented in a webinar titled "**Design a Cloud Connected IoT Gateway with Security Protection**". The video recording of this webinar is available as part of TI training - Design a Cloud Connected IoT Gateway with Security Protection

1.1 TM4C123GH6PM

The TM4C123GH6PM MCU is targeted for industrial applications including the following: remote monitoring, electronic point-of-sale machines, test equipment, measurement equipment, network appliances, switches, factory automation, HVAC, building control, gaming equipment, motion control, transportation, and security.

The TM4C123GH6PM is an 80-MHz high-performance MCU with up to 256KB on-chip Flash and 32KB on-chip SRAM. There are up to 43 GPIOs with programmable control for GPIO interrupts, pad configuration, and pin muxing. The MCU is integrated with six 32-bit general-purpose timers (up to twelve 16-bit timers), eight UARTs, four synchronous serial interface (SSI) modules, four inter-integrated circuit (I2C) modules, two 12-bit analog-to-digital converters (ADCs) with 12 analog input channels and a sample rate of one million samples per second, eight pulse width modulation (PWM) generator blocks, and two quadrature encoder interface (QEI) modules. The on-chip universal serial bus (USB) controller supports the USB OTG/Host/Device modes. The ARM® PrimeCell 32-channel configurable μDMA controller is also integrated to provide a method to offload data transfer tasks from the Cortex®-M4 processor and to efficiently use the processor and the bus bandwidth.

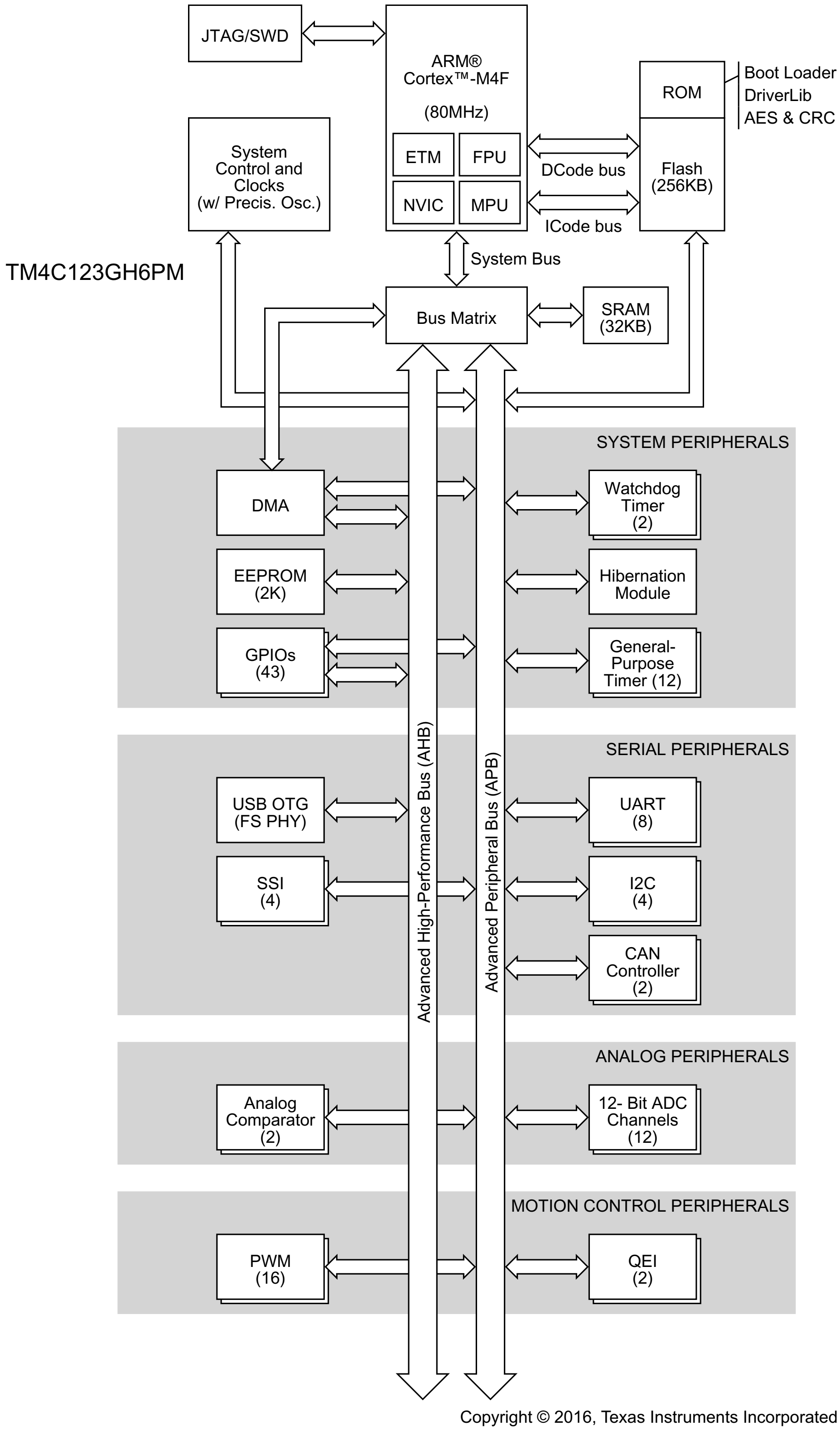


Figure 1. TM4C123GH6PM MCU High-Level Block Diagram

1.2 TM4C1294NCPDT

The TM4C1294ECPDT is a 120-MHz high-performance 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 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. The TM4C129ENCPDT MCU is code-compatible to all members of the extensive Tiva™ C Series, providing flexibility to fit precise needs.

This MCU is hardware encryption enabled. It provides security by its CRC hardware, AES hardware-accelerated data encryption, DES block cipher implementation, hashing hardware accelerator, and four tamper units along with tamper event response. Therefore, the TM4C1294ECPDT is ideally suited for developing secure cloud connected IoT systems to assist factory control or automation systems.

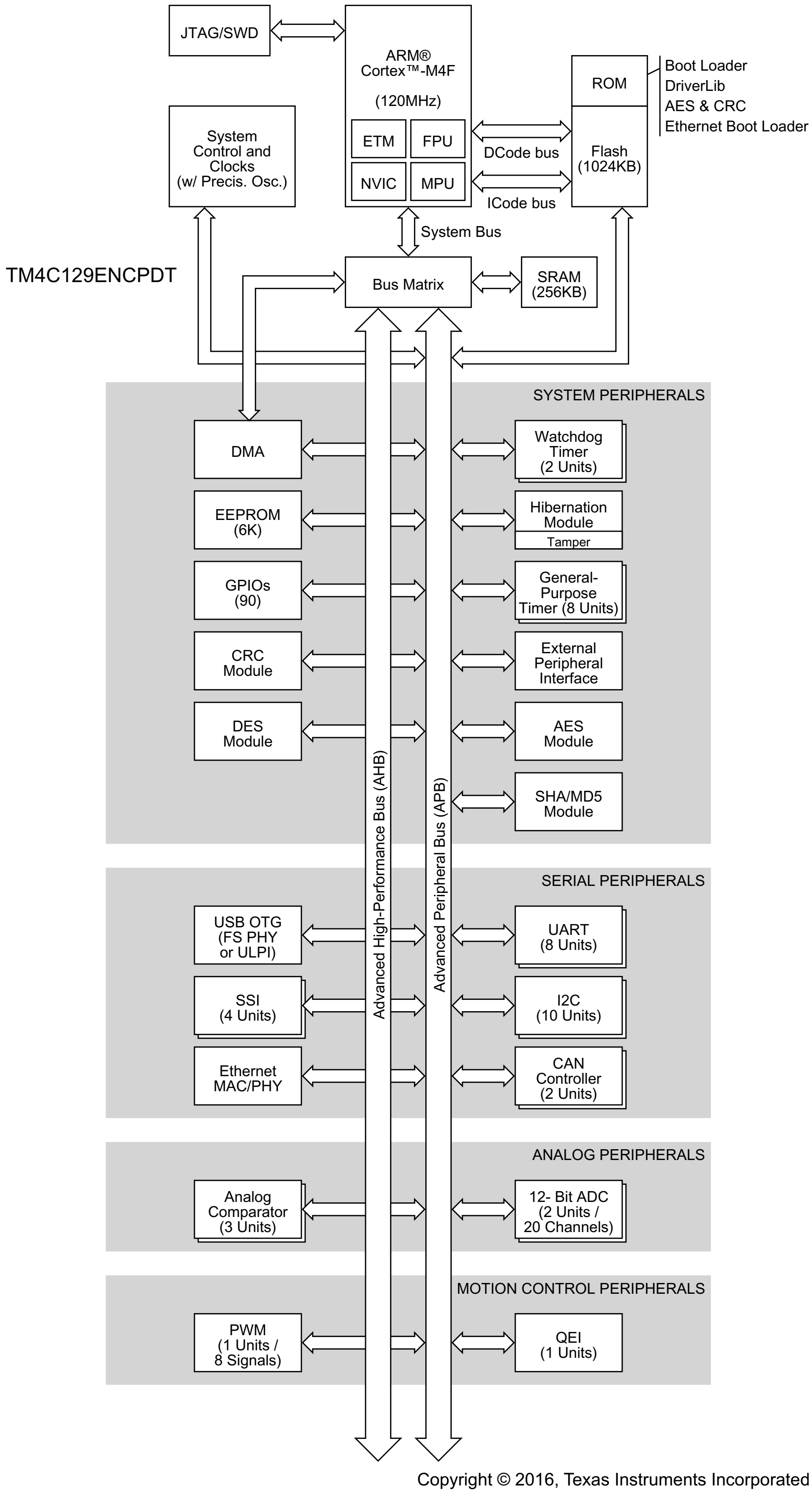


Figure 2. TM4C1294ECPDT MCU High-Level Block Diagram

1.3 DRV8833

The DRV8833 has two H-bridge drivers to drive a bipolar stepper motor, two DC brush motors, or other inductive loads. Aimed at driving 3.3-V and 5-V motors, this stepper driver with integrated FETs support up to 1.5 A_{RMS} with a low-power sleep mode to conserve power for battery-powered applications. Internal shutdown functions with a fault output pin protect the device from overcurrent, short-circuit, undervoltage lockout, and over temperature.

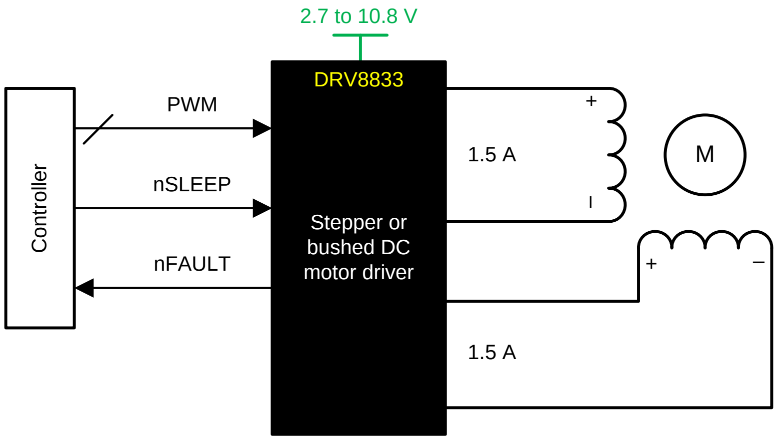


Figure 3. DRV833 Functional Block Diagram

1.4 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 supports Station, Access Point, and Wi-Fi Direct modes. The device also supportsWPA2 personal and enterprise security and WPS 2.0. This subsystem includes embedded TCP/IP and TLS/SSL stacks, HTTP server, and multiple Internet protocols.

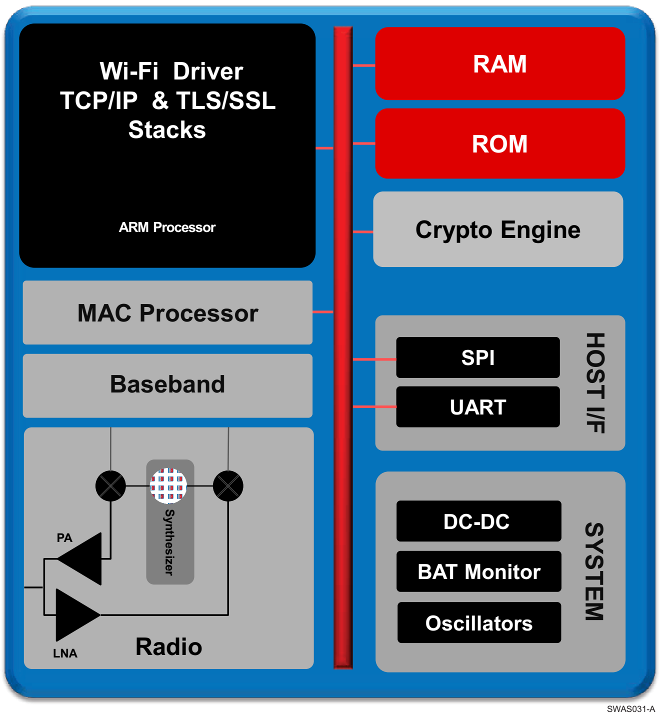


Figure 4. CC3100 Hardware Overview

1.5 CC2650

The CC2650 is a cost-effective, ultra-low-power, 2.4-GHz RF wireless MCU targeting *Bluetooth*® Smart, ZigBee® and 6LoWPAN, and ZigBee RF4CE remote control applications. A very low active RF and MCU current and low-power mode current consumption provides excellent battery lifetime, operates on small coin-cell batteries, and operates in energy-harvesting applications. The CC2650 contains a 32-bit ARM Cortex-M3 running at 48 MHz as the main processor and has a rich peripheral feature set, including an ultra-low-power sensor controller. The ultra-low-power sensor controller is ideal for interfacing external sensors or collecting analog and digital data while the rest of the system is in sleep mode. The *Bluetooth* low-energy (BLE) controller and the IEEE 802.15.4 MAC are embedded into ROM and are running partially on a separate ARM Cortex-M0 processor. This architecture improves overall system performance and power consumption and frees up flash memory for the application.

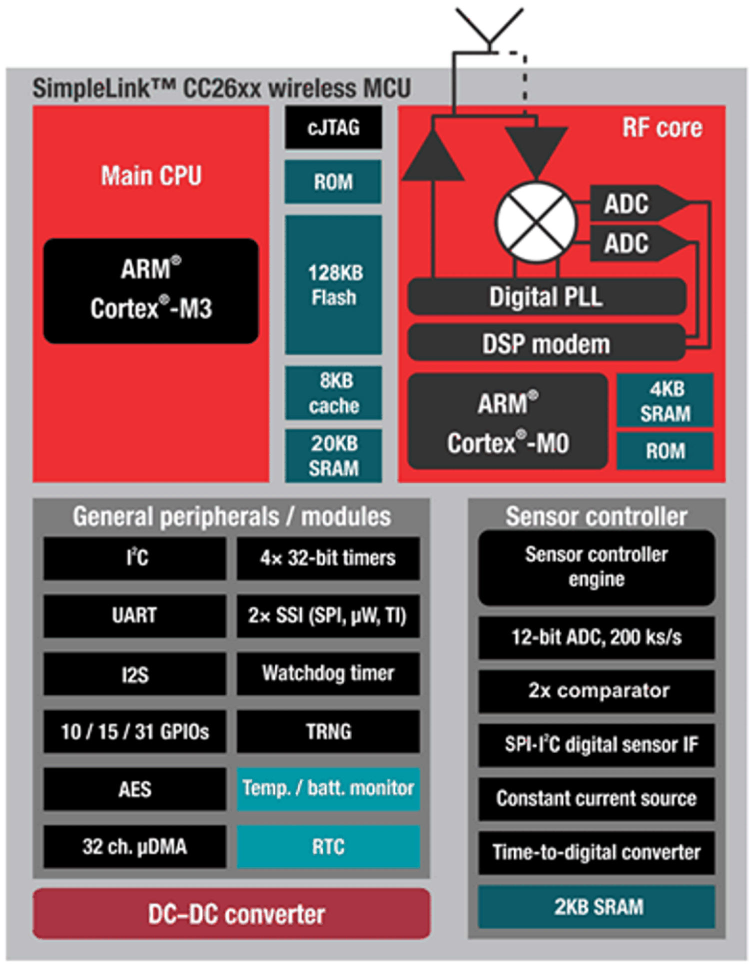


Figure 5. CC2650 Architectural Overview

1.6 BLE SensorTag

The SensorTag includes 10 low-power MEMS sensors in a tiny red package. It is expandable with DevPacks to make it easy to add more sensors or actuators. It can be connected to the cloud with *Bluetooth* Smart and sensor data is online in 3 minutes. The SensorTag is based on the CC2650 wireless MCU, offering 75% lower power consumption than previous *Bluetooth* Smart products. This allows the SensorTag to be battery powered and offer years of battery lifetime from a single coin cell battery. The *Bluetooth* Smart SensorTag includes iBeacon technology, which allows a phone to launch applications and customize content based on SensorTag data and physical location. Additionally, the SensorTag can be enabled with ZigBee®/6LoWPAN technology.

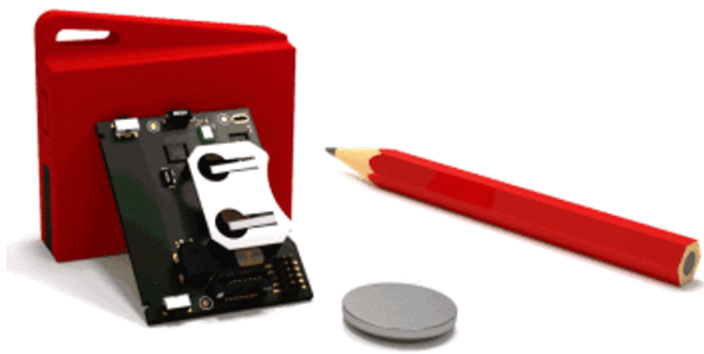


Figure 6. BLE SensorTag With Coin Cell Battery

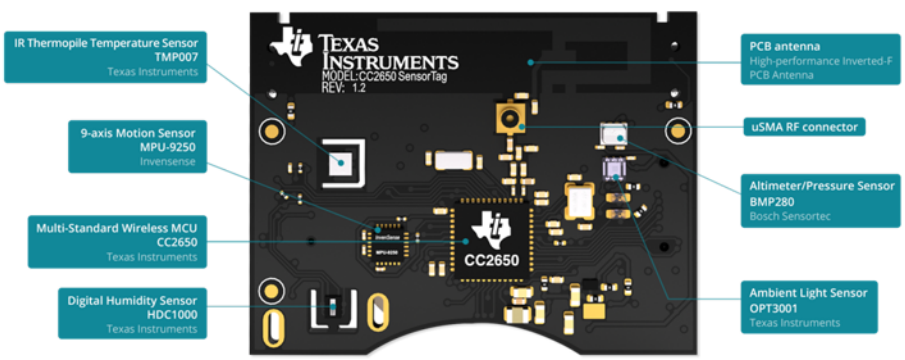


Figure 7. BLE SensorTag Internals (CC2650 Along With Sensors)

1.7 CC1310

The device is a member of the CC26xx and CC13xx family of cost-effective, ultra-low-power, 2.4-GHz and sub-1-GHz RF devices. Very low active RF, MCU current, and low-power mode current consumption provide excellent battery lifetime and allow operation on small coin-cell batteries and in energy-harvesting applications. The CC1310 is the first part in a Sub-1GHz family of cost-effective, ultra-low-power wireless MCUs. This device combines a flexible, very low-power RF transceiver with a powerful 48-MHz Cortex-M3 MCU in a platform supporting multiple physical layers and RF standards. A dedicated radio controller (Cortex-M0) handles low-level RF protocol commands that are stored in ROM or RAM, thus ensuring ultra-low power and flexibility. The low-power consumption of the CC1310 does not come at the expense of RF performance; the CC1310 has excellent sensitivity and robustness (selectivity and blocking) performance.

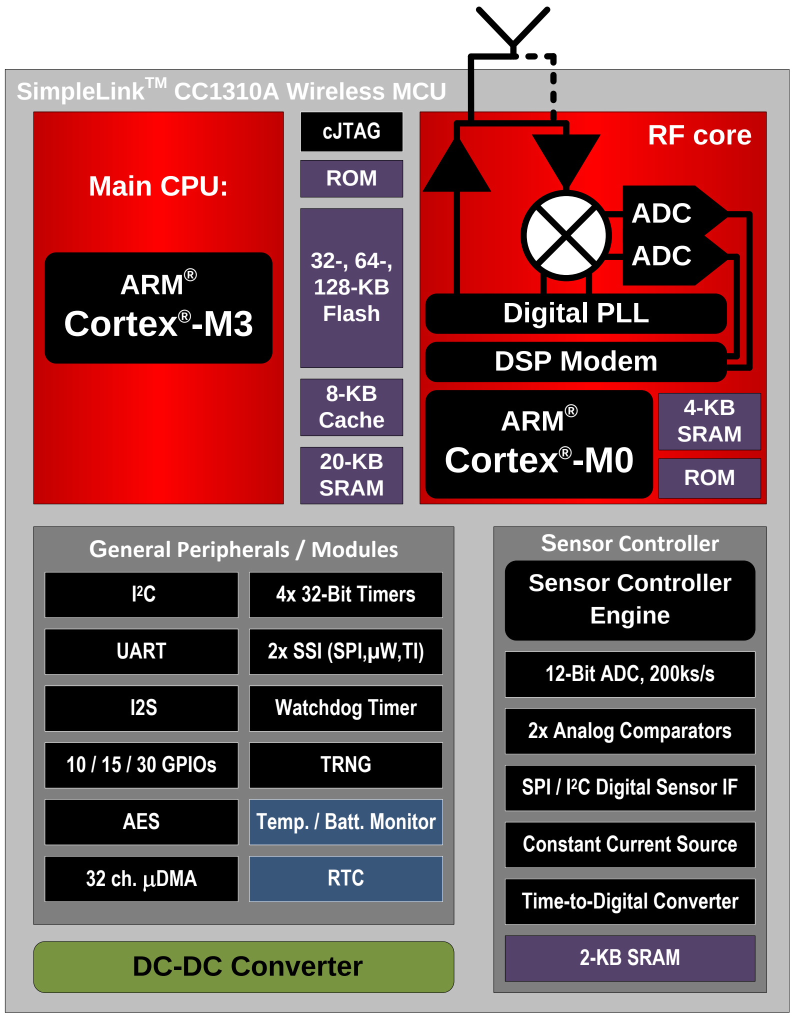


Figure 8. CC1310 Functional Block Diagram

1.8 TRF7970A

The TRF7970A is an integrated analog front-end and data-framing device for a 13.56-MHz RFID and near field communication (NFC) system. Built-in programming options make the device suitable for a wide range of applications for proximity and vicinity identification systems. The device can perform in one of three modes: RFID and NFC reader, NFC peer, or in card emulation mode. Built-in user-configurable programming options make the device suitable for a wide range of applications. The TRF7970A device is configured by selecting the desired protocol in the control registers. Direct access to all control registers allows fine tuning of various reader parameters as needed.

1.9 RF430CL330H

The Texas Instruments Dynamic NFC Interface Transponder RF430CL330H is an NFC Tag Type 4 device that combines a wireless NFC interface and a wired SPI or I^2C interface to connect the device to a host. The NDEF message in the SRAM can be written and read from the integrated SPI or I2C serial communication interface and can also be accessed and updated wirelessly through the integrated ISO14443B-compliant RF interface that supports up to 848 kbps. This operation allows NFC connection handover for an alternative carrier like BLE and Wi-Fi as an easy and intuitive pairing process or authentication process with only a tap. As a general NFC interface, the RF430CL330H enables end equipments to communicate with the fast-growing infrastructure of NFC-enabled smart phones, tablets, and notebooks.

1.10 TM4C123 Swizzle Adapter Board

The TM4C123 swizzle adapter board is a special purpose hardware adapter board to interface the TM4C123x LaunchPad with NFC and Wi-Fi BoosterPacks along with rendering necessary PWM outputs for the DRV8833 motor drive. Inorder to accommodate PWM pins, instead of using default TM4C123 SPI2, SPI0 is used to communicate to CC3100, hence CC3100 Booster pack cannot be mounted directly. Swizzle adapter board reroutes SPI0 lines to SPI2 position on a different header to facilitate CC3100 mounting, this can be done manually through Jumper wires too.

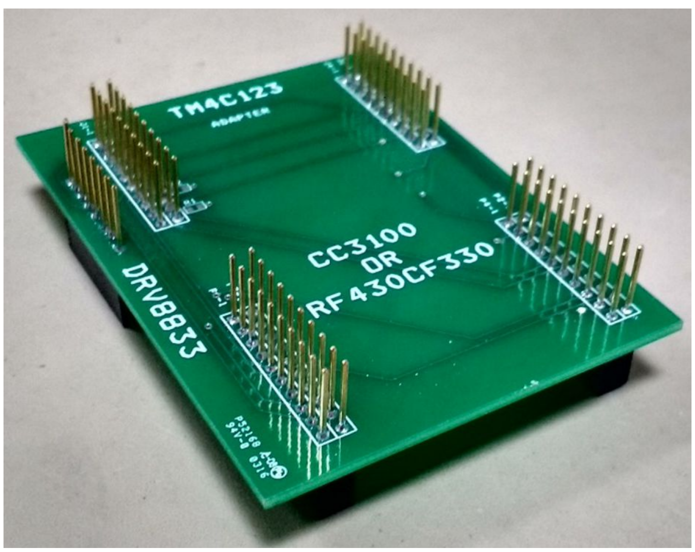


Figure 9. TM4C123 Swizzle Adapter Board

1.11 Exosite

Exosite is Internet of Things (IoT) software as a service (SaaS) company that develops software for companies that view and analyze data collected from sensors built into physical objects. Exosite's most basic concept is to make internet-connected physical things useful to people and businesses. Exosite's products help developers, companies, and organizations build IoT product solutions by providing pieces of the IoT system, including device code, a device connectivity and application platform, and hosted applications and services.

To get acquainted quickly with the way TM4C devices communicate with the Exosite, go through the "qs_iot" example project available in the TivaWare™. It requires the user to sign-up or log into the www.ti.exosite.com portal and then register the device on the server so that the device can be identified securely by the server and further communication can take place.

1.12 Stepper Motor Control

A stepper motor is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without feedback. The stepper motor is widely used in a wide range of applications involving precision motion control.

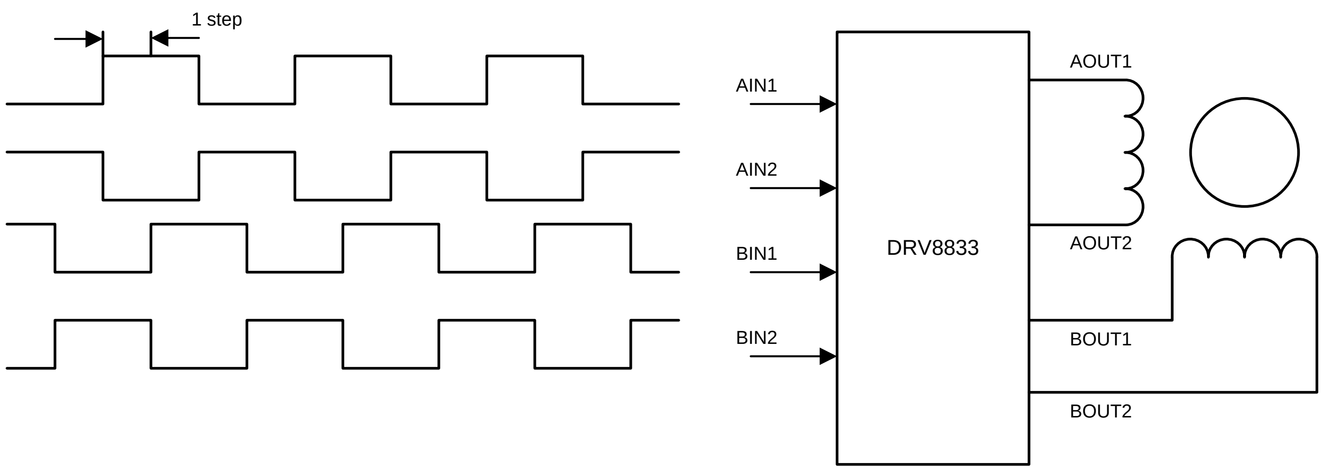


Figure 10. Driving the Stepper Motor in Full-Step Mode

2 Getting Started Hardware

2.1 List of Hardware Components

This application requires following hardware components. Some minor modifications are required in the hardware, and they must be done before proceeding further with the software setup.

Table 1. List of Hardware Components

BOARD NAME	INFO	QTY
TM4C129EXL	Tiva-C Crypto connected LaunchPad	1
TM4C123GXL	Tiva-C LaunchPad	4
CC3100BOOST	CC3100 Wi-Fi BoosterPack	2
BOOST-CCEMADAPTER	EM adapter BoosterPack	5
CC2650EM-4XD	CC2650 BLE device	2
CC1310EMK	CC1310 Sub-1GHz device	3
DLP-7970ABP	TRF7970A NFC transceiver BoosterPack	1
DLP-RF430BP	RF430CL330H NFC tag BoosterPack	4
BLE SensorTag	BLE SensorTag	1
TM4C123 swizzle adapter board	Specially designed board to interface multiple BoosterPacks	1
DRV8833-EVM	Driver board to run a stepper motor	1
Stepper motor	A stepper motor with 5- to 12-V input parameters	1
DC power supply	An external power supply to run the stepper motor	1

2.2 Hardware Configuration

2.2.1 Gateway

2.2.1.1 Setting up Gateway Hardware

The hardware components required to setup the gateway are listed in Table 2:

Table 2. Gateway Hardware Setup

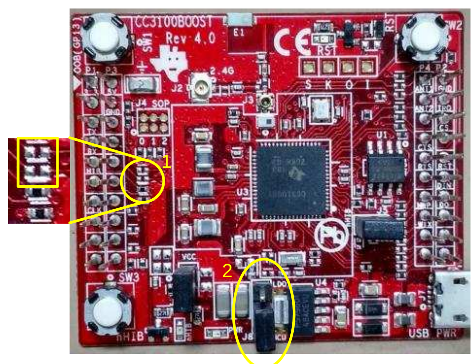
SR NO	COMPONENT NAME
1	**TM4C129EXL Crypto Connected LaunchPad** No modifications are required.
2	**CC3100 Wi-Fi BoosterPack** Mandatory configurations on the BoosterPack are as shown in Figure 11. Follow these steps: 1. Remove the two 0-Ω resistors as shown in Figure 11. These resistors are connected to the RX and TX pins on the BoosterPack header P1. Removing them ensures that the CC3100 does not interfere with its TX and RX pins. 2. Change the jumper J-8 setting to *MCU*.  **Figure 11. Hardware Configuration of CC3100 BoosterPack on Gateway**
3	**CC2650EM BLE Device** No modifications are required.
4	**CC1310EM Sub-1GHz Device** No modifications are required.

Table 2. Gateway Hardware Setup (continued)

<table>
<tr><th>SR NO</th><th>COMPONENT NAME</th></tr>
<tr><td>5</td><td>

EM Adapter BoosterPack (Two Numbers)

NOTE: Two BoosterPacks are required to connect to BLE and Sub-1GHz devices. Both follow the same configuration. In fact, all the EM adapter BoosterPacks used in this entire application demo follow same hardware configuration.

Mandatory configurations on the BoosterPacks are as shown in Figure 12. Follow these steps:

1. Remove all the 0-Ω resistors R-2 to R-20 except R-3 and R-4 as shown in Figure 12.
2. Connect the inner R-18 junction with the outer R-15 junction as shown in Figure 12. This change connects the TM4C and CC2650 and CC1310 RESET pins.

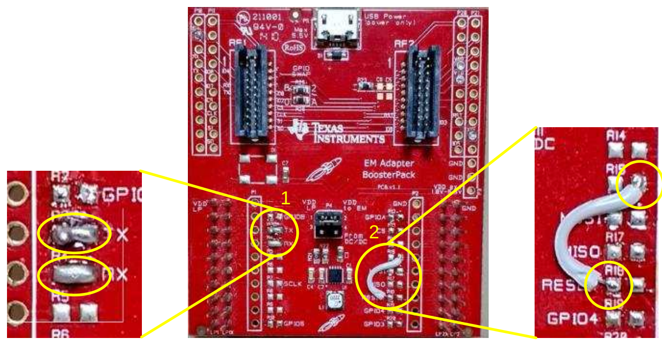


Figure 12. Hardware Configuration of EM Adapter BoosterPack on Gateway

NOTE: The RESET pin of the EM Adapter BoosterPack is not aligned with the RESET pin of the TM4C1294EXL on BoosterPack-1. Hence, hard-wire the RESET pin to avoid unknown observations.

</td></tr>
<tr><td>6</td><td>

TRF7970A NFC Transceiver BoosterPack

Mandatory configurations on the BoosterPack are as shown in Figure 13. Follow these steps:

1. Solder a 0-Ω resistor to connect the IRQ junction to the adjacent junction numbered as "2" as shown in Figure 13.

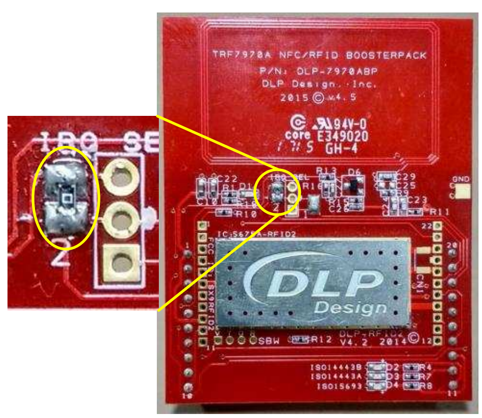


Figure 13. Hardware Configuration of TRF7970A BoosterPack on Gateway

</td></tr>
</table>

2.2.1.2 BoosterPack Signal Mapping

The header connections for setting up the gateway are shown in the following tables. Refer to these tables after programming all the hardware components with the necessary binaries.

Table 3. BoosterPack-1 Signal Mapping

BOOSTERPACK CONNECTOR	TM4C1294 CRYPTO CLP	EM ADAPTER BOOSTERPACK	CC2650EM	TRF7970A NFC BOOSTERPACK
A1-1	3.3 V	VDD_LP	VDD	1 (VDD)
A1-2	PE4	Unused	Unused	2 (Unused)
A1-3	PC4_U7RX	LP1.3	RF1.09	3 (Unused)
A1-4	PC5_U7TX	LP1.4	RF1.07	4 (Unused)
A1-5	PC6	Unused	Unused	5 (Unused)
A1-6	PE5	Unused	Unused	6 (Unused)
A1-7	PD3_SSI2CLK	Unused	Unused	7 (Unused)
A1-8	PC7	Unused	Unused	8 (IRQ)
A1-9	PB2	Unused	Unused	9 (SS)
A1-10	PB3	Unused	Unused	10 (EN)
D1-1	GND	GND	GND	20 (GND)
D1-2	PM3	Unused	Unused	19 (Unused)
D1-3	PH2	Unused	Unused	18 (Unused)
D1-4	PH3	Unused	Unused	17 (Unused)
D1-5	RESET	RESET(1)	RF2.15 (RESET)	16 (RESET)
D1-6	PD1_I2C7SDA	Unused	Unused	15 (MOSI)
D1-7	PD0_I2C7SCl	Unused	Unused	14 (MISO)
D1-8	PN2	Unused	Unused	13 (Unused)
D1-9	PN3	Unused	Unused	12 (Unused)
D1-10	PP2	Unused	Unused	11 (Unused)

(1) The RESET pin of the EM Adapter BoosterPack is not aligned with the RESET pin of the TM4C1294EXL on Boosterpack-1. Hence, hard-wire the RESET pin to avoid unknown observations.

Table 4. BoosterPack-2 Signal Mapping

BOOSTERPACK CONNECTOR	TM4C1294 CRYPTO CLP	CC3100 BOOSTERPACK	EM ADAPTER BOOSTERPACK	CC1310EM
A2-1	3.3 V	P1.1 (3.3 V)	VDD_LP	VDD
A2-2	PD2	Unused	Unused	Unused
A2-3	PP0	Unused	LP1.3	RF1.09
A2-4	PP1	Unused	LP1.4	RF1.07
A2-5	PD4	P1.5 (HIB)	Unused	Unused
A2-6	PD5	Unused	Unused	Unused
A2-7	PQ0	P1.7 (CLK)	Unused	Unused
A2-8	PP4	Unused	Unused	Unused
A2-9	PN5	Unused	Unused	Unused
A2-10	PN4	Unused	Unused	Unused
D2-1	GND	P2.1 (GND)	GND	GND
D2-2	PM7	P2.2 (IRQ)	Unused	Unused
D2-3	PP5	P2.3 (CS)	Unused	Unused
D2-4	PA7	Unused	Unused	Unused
D2-5	RESET	P2.5 (RESET)	RESET[1]	RF2.15 (RESET)
D2-6	PQ2	P2.6 (DIN)	Unused	Unused
D2-7	PQ3	P2.7 (DO)	Unused	Unused
D2-8	PP3	Unused	Unused	Unused
D2-9	PQ1	Unused	Unused	Unused
D2-10	PM6	Unused	Unused	Unused

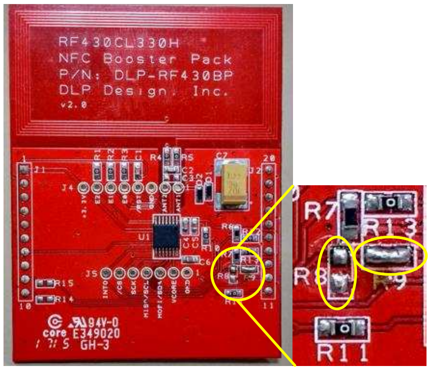
(1) The RESET pin of the EM Adapter BoosterPack is not aligned with the RESET pin of the TM4C1294EXL on BoosterPack-1. Hence, hard-wire the RESET pin to avoid unknown observations.

2.2.2 Wi-Fi Node

2.2.2.1 *Setting up Wi-Fi Node Hardware*

The hardware components required to setup the Wi-Fi node are listed in Table 5:

Table 5. Wi-Fi Node Hardware Setup

SR NO	COMPONENT NAME
1	**TM4C123GXL LaunchPad** No modifications are required.
2	**CC3100 Wi-Fi BoosterPack** Change the jumper J-8 setting to MCU.
3	**RF430CL330H BoosterPack** Mandatory configurations on the BoosterPack are as shown in Figure 14. Follow this step: 1. Remove the 0-Ω resistor at R-8 and place a 0-Ω resistor at R-9 as shown in Figure 14.  **Figure 14. Hardware Configuration of TRF7970A BoosterPack on Wi-Fi Node**
4	**TM4C123 Swizzle Adapter Board** No modifications are required.
5	**DRV8833 Stepper Motor Driver Board** No modifications are required.
6	**Stepper Motor** No modifications are required.
7	**External Power Supply (5 to 12 V)** No modifications are required.

2.2.2.2 Wi-Fi Node Signal Mapping

The header connections for setting up the Wi-Fi node are shown in the following tables. Refer to these tables after programming all the hardware components with the necessary binaries.

Table 6. Wi-Fi Node Signal Mapping

BOOSTERPACK CONNECTOR	CC3100 BOOSTERPACK (WI-FI)	DLP-RF430BP (NFC)	TM4C123 LAUNCHPAD CONNECTOR
P1-1	3.3 V	3.0 V	J1-1: 3.3 V
P1-2	Open	Unused	Open
P1-3	CC_UART1_TX	Unused	J1-3: PB0_U1RX
P1-4	CC_UART1_RX	Unused	J1-4: PB1_U1TX
P1-5	CC_nHIB	Unused	J3-7: PE1
P1-6	Open	Unused	J3-8: PE2
P1-7	CC_SPI_CLK	DATA_CLK	J2-10: PA2 _SSI0CLK
P1-8	Open	RESET	J3-3: PD0
P1-9	Test_3	Unused	J3-5: PD2
P1-10	FORCE_AP	Unused	J4-3: PB3
P3-1	5 V	N/A	J3-1: 5V
P3-2	GND	N/A	J3-2GND
P3-3	Open	N/A	
P3-4	Open	N/A	
P3-5	Open	N/A	
P3-6	Open	N/A	
P3-7	Open	N/A	
P3-8	Open	N/A	
P3-9	Open	N/A	
P3-10	Open	N/A	
P4-1	Test_29	N/A	J4-1: PF2
P4-2	Test_30	N/A	J4-2: PF3
P4-3	Open	N/A	
P4-4	CC_URT1_CTS	N/A	J2-4: PF0_U1RTS
P4-5	CC_UART1_RTS	N/A	J3-10: PF1_U1CTS
P4-6	Open	N/A	
P4-7	CC_NWP_UART_TX	N/A	J4-6: PC6_U3RX
P4-8	CC_WL_UART_TX	N/A	J1-5: PE4_U5RX
P4-9	CC_WLRS232_RX	N/A	J4-9: PD7_U2TX
P4-10	CC_WLRS232_TX	N/A	J4-8: PD6_U2RX
P2-1	GND	GND	J2-1: GND
P2-2	CC_IRQ	Unused	J3-6: PD3
P2-3	CC_SPI_CS	Unused	J2-3: PE0
P2-4	Open	Unused	Open
P2-5	MCU_RESET_IN	Unused	J2-5: RESET
P2-6	CC_SPI_DIN	MOSI/SDA	J1-8: PA5_SSI0TX, J1-10: PA7_I2C1SDA
P2-7	CC_SPI_DOUT	MISO/SCL	J2-8: PA4 _SSI0RXJ1-9: PA6_I2C1SCL
P2-8	Test_63	SPI_CS	J1-7: PB4
P2-9	Test_64	INTO	J4-7: PC7
P2-10	Test_18	Unused	J3-3: PD0

Table 7. Stepper Motor Drive

DRV8833	TM4C123 SWIZZLE ADAPTER BOARD	TM4C123
AIN1	J2-6	PB7_M0PWM1
AIN2	J1-2	PB5_M0PWM3
BIN2	J1-6	PE5_M0PWM5
BIN1	J4-5	PC5_M0PWM7
GND	GND	GND
VDD (external power supply)	Unused	Unused

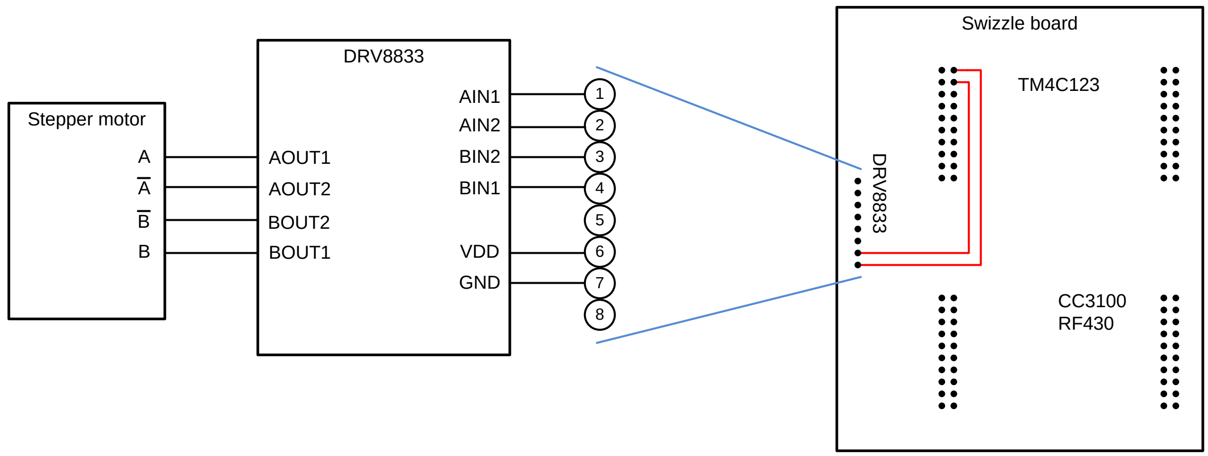


Figure 15. Block Diagram of Interface Between TM4C123 Swizzle Adapter Board and DRV8833

Note that the header on the TM4C123 swizzle adapter board can be directly plugged into the pins on the DRV8833.

2.2.3 BLE Node

2.2.3.1 Setting up BLE Node Hardware

The hardware components required to setup the BLE node are listed in Table 8:

Table 8. BLE Node Hardware Setup

<table>
<tr><th>SR NO</th><th>COMPONENT NAME</th></tr>
<tr><td>1</td><td>TM4C123GXL LaunchPad
No modifications are required.</td></tr>
<tr><td>2</td><td>CC2650EM BLE Device
No modifications are required.</td></tr>
<tr><td>3</td><td>EM Adapter BoosterPack
Mandatory configurations on the BoosterPacks are as shown in Figure 16. Follow these steps:
1. Remove all the 0-Ω resistors R-2 to R-20 except R-3 and R-4 as shown in Figure 16..
2. Connect the inner R-18 junction with the outer R-15 junction as shown in Figure 16. This change connects the TM4C123 and CC2650 RESET pins.

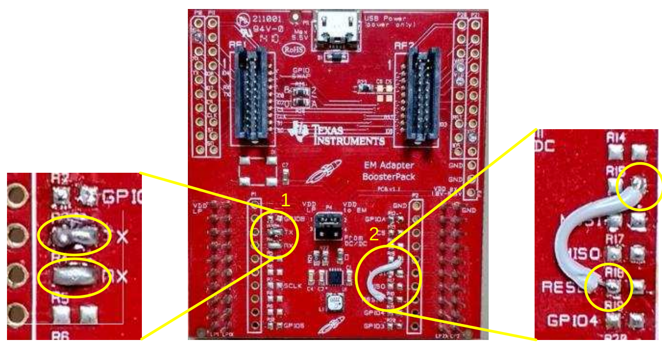



Figure 16. Hardware Configuration of EM Adapter BoosterPack on BLE Node
NOTE: The RESET pin of the EM Adapter BoosterPack is not aligned with the RESET pin of the TM4C1294EXL on BoosterPack-1. Hence, hard-wire the RESET pin to avoid unknown observations.</td></tr>
<tr><td>4</td><td>RF430CL330H BoosterPack
No modifications are required.</td></tr>
</table>

2.2.3.2 BLE Node Signal Mapping

The header connections for setting up the BLE node are shown in the following tables. Refer to Table 9 after programming all the hardware components with the necessary binaries.

Table 9. BLE Node Signal Mapping

BOOSTERPACK CONNECTOR	TM4C123 LAUNCHPAD	EM ADAPTER BOOSTERPACK	CC2650EM	RF430CL330H BOOSTERPACK
A-1	3.3 V	VDD_LP	VDD	VDD (J1.1)
A-2	PB5	Unused	Unused	Unused
A-3	PB0	LP1.3	RF1.09	Unused
A-4	PB1	LP1.4	RF1.07	Unused
A-5	PE4	Unused	Unused	Unused
A-6	PE5	Unused	Unused	Unused
A-7	PB4	Unused	Unused	Unused
A-8	PA5	Unused	Unused	Unused
A-9	PA6	Unused	Unused	Unused
A-10	PA7	Unused	Unused	Unused
D-1	GND	GND	GND	GND
D-2	PB2	Unused	Unused	Unused
D-3	PE0	Unused	Unused	Unused
D-4	PF0	Unused	Unused	Unused
D-5	RESET	RESET(1)	RF2.15 (RESET)	RESET (J2.16)
D-6	PB7	Unused	Unused	MOSI (J2.15)
D-7	PB6	Unused	Unused	MISO (J2.14)
D-8	PA4	Unused	Unused	Unused
D-9	PA3	Unused	Unused	INTO (J2.12)
D-10	PA2	Unused	Unused	Unused

(1) The RESET pin of the EM Adapter BoosterPack is not aligned with the RESET pin of the TM4C123GXL on BoosterPack-1. Hence, hard-wire the RESET pin to avoid unknown observations.

2.2.4 Sub-1GHz Node

2.2.4.1 Setting up Sub-1GHz Node Hardware

The hardware components required to setup the Sub-1GHz node are listed in Table 10:

Table 10. Sub-1GHz Node Hardware Setup

<table>
<tr><th>SR NO</th><th>COMPONENT NAME</th></tr>
<tr><td>1</td><td>TM4C123GXL LaunchPad
No modifications are required.</td></tr>
<tr><td>2</td><td>CC1310EM Sub-1GHz Device
No modifications are required.</td></tr>
<tr><td>3</td><td>EM Adapter BoosterPack
Mandatory configurations on the BoosterPacks are as shown in Figure 17. Follow these steps:
1. Remove all the 0-Ω resistors R-2 to R-20 except R-3 and R-4 as shown in Figure 17.
2. Connect the inner R-18 junction with the outer R-15 junction as shown in Figure 17. This change connects the TM4C123 and CC1310 RESET pins.

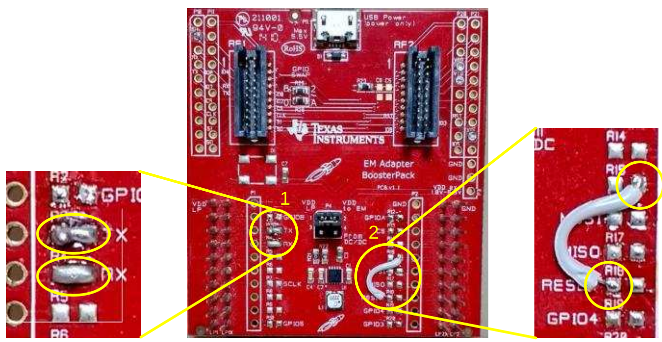



Figure 17. Hardware Configuration of EM Adapter BoosterPack on Sub-1GHz Node
NOTE: The RESET pin of the EM Adapter BoosterPack is not aligned with the RESET pin of the TM4C1294EXL on BoosterPack-1. Hence, hard-wire the RESET pin to avoid unknown observations.</td></tr>
<tr><td>4</td><td>RF430CL330H BoosterPack
No modifications are required.</td></tr>
</table>

2.2.4.2 Sub-1GHz Node Signal Mapping

The header connections for setting up the Sub-1GHz node are shown in Table 11. Refer to Table 11 after programming all the hardware components with the necessary binaries.

Table 11. Sub-1GHz Node Signal Mapping

BOOSTERPACK CONNECTOR	TM4C123 LAUNCHPAD	EM ADAPTER BOOSTERPACK	CC1310EM	RF430CL330H BOOSTERPACK
A-1	3.3 V	VDD_LP	VDD	VDD (J1.1)
A-2	PB5	Unused	Unused	Unused
A-3	PB0	LP1.3	RF1.09	Unused
A-4	PB1	LP1.4	RF1.07	Unused
A-5	PE4	Unused	Unused	Unused
A-6	PE5	Unused	Unused	Unused
A-7	PB4	Unused	Unused	Unused
A-8	PA5	Unused	Unused	Unused
A-9	PA6	Unused	Unused	Unused
A-10	PA7	Unused	Unused	Unused
D-1	GND	GND	GND	GND
D-2	PB2	Unused	Unused	Unused
D-3	PE0	Unused	Unused	Unused
D-4	PF0	Unused	Unused	Unused
D-5	RESET	RESET[(1)]	RF2.15 (RESET)	RESET (J2.16)
D-6	PB7	Unused	Unused	MOSI (J2.15)
D-7	PB6	Unused	Unused	MISO (J2.14)
D-8	PA4	Unused	Unused	Unused
D-9	PA3	Unused	Unused	INTO (J2.12)
D-10	PA2	Unused	Unused	Unused

(1) The RESET pin of EM Adapter BoosterPack is not aligned with the RESET pin of the TM4C123GXL on BoosterPack-1. Hence, hard-wire the RESET pin to avoid unknown observations.

3 Getting Started Software

3.1 Gateway Software Architecture

Figure 18 explains the architecture of the TM4C129x-based IoT gateway.

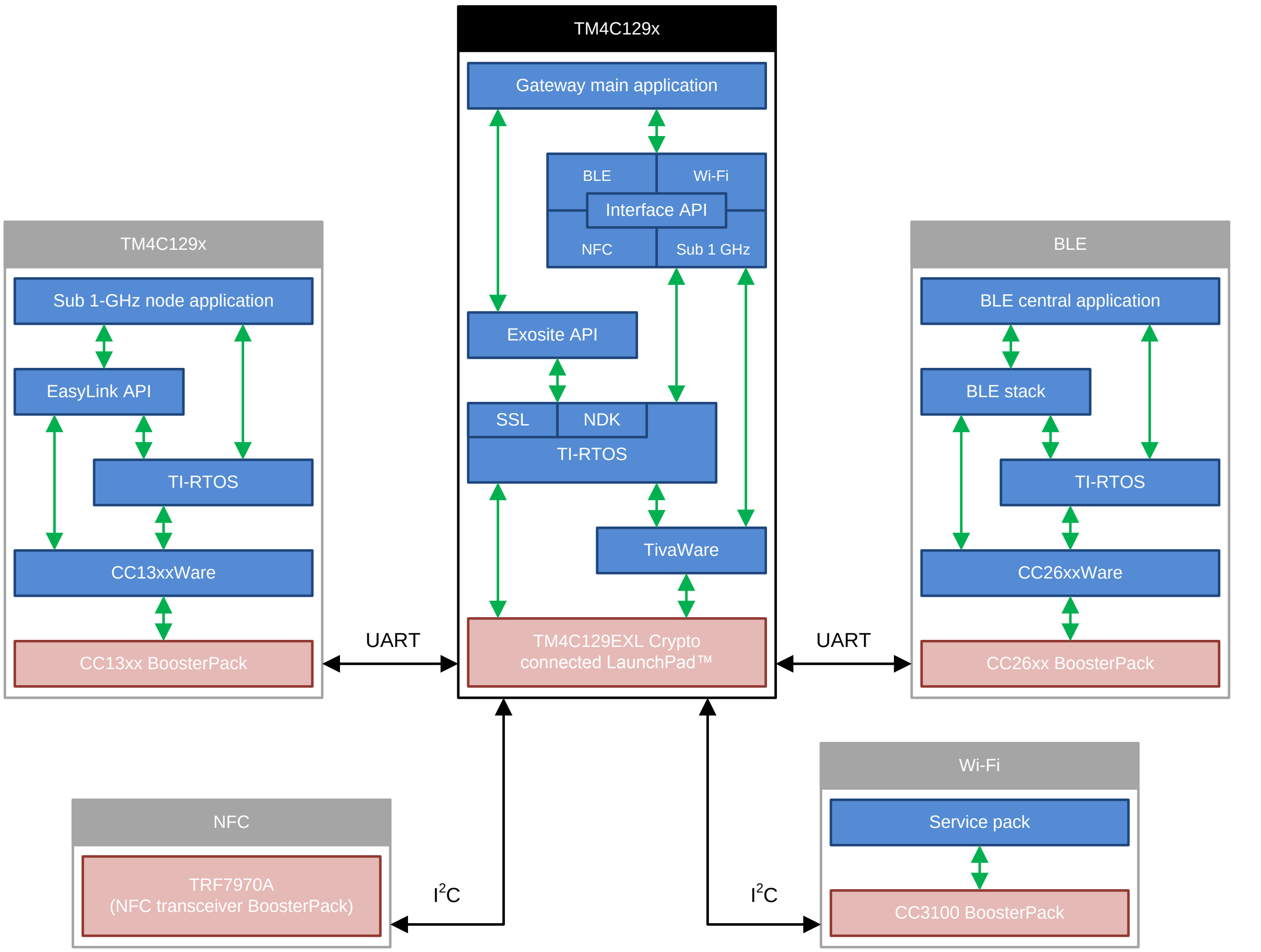


Figure 18. Gateway Software Architecture Diagram

TM4C129x software blocks:

- TivaWare C: for TM4C hardware register access and serial communications to other hardware through UART, SPI, and I^{2}C.
- Exosite API: a C language translation of the set of standard routines, which are required to connect and communicate with the Exosite Cloud Server. This implementation internally uses NDK and wolfSSL libraries of TI-RTOS.
- Interface API: allows the TM4C129x to communicate to the BLE, Wi-Fi, and Sub-1GHz hardware mounted on it using the onboard serial peripherals such as UART, SPI, and I^{2}C.
- TI-RTOS: used for scheduling tasks, which handle communication with:
  - Hardware peripherals (BLE, Wi-Fi, Sub-1GHz)
  - Exosite
  - Command line interface

BLE software blocks:

- TI-RTOS: for general scheduling
  - Manages the command and response handling over the UART
  - Operates the BLE peripheral
- CC26xxWare: performs CC26xx hardware access and the UART operation
- BLE stack: supports the BLE protocol

Sub-1GHz software blocks:

- TI-RTOS: for general scheduling
  - Manages the command and response handling over the UART
  - Operates the Sub-1GHz peripheral
- CC13xxWare: performs CC13xx hardware access and the UART operation
- EasyLink API: supports the Sub-1GHz protocol

Wi-Fi software blocks:

- There is no specific software required to be run on the CC3100 Wi-Fi module. However, for the sake of maintaining uniformity across platforms, the CC3100 is programmed with the latest CC3100SDK-SERVICEPACK.

NFC software blocks:

- There are no NFC software blocks in this design.

3.2 Wi-Fi Node Software Architecture

Figure 19 shows the software architecture of the Wi-Fi node and slave.

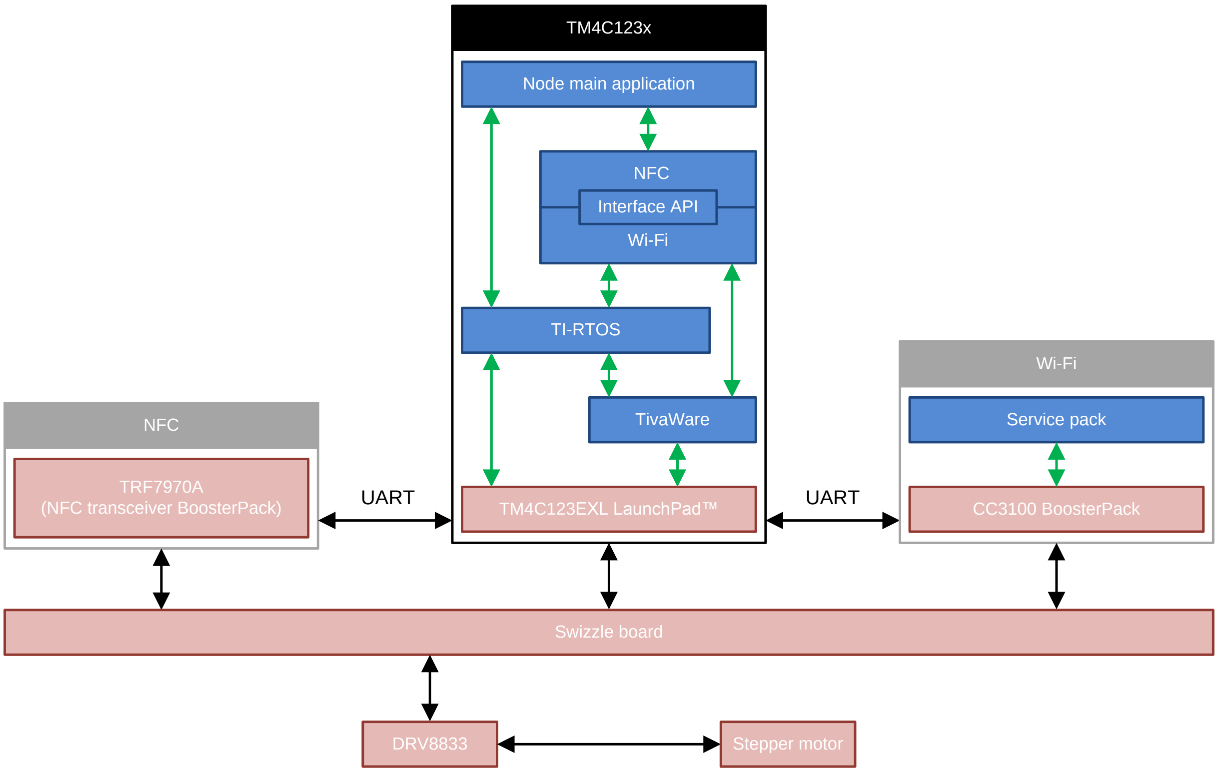


Figure 19. Wi-Fi Node Software Architecture

TM4C123x software blocks:

- TivaWare C: allows for TM4C hardware register access and serial communications to other hardware through SPI. Also, it controls the PWM signals from the onboard PWM modules, which are used to drive the motor.
- Interface API: allows the TM4C123x to communicate to the Wi-Fi, NFC hardware mounted on it using the onboard SPI
- TI-RTOS: for scheduling tasks, which handle communication with
  - Hardware peripherals (Wi-Fi, NFC)
  - Motor drive, sending desired PWM signals
  - Command line interface

Wi-Fi software blocks:

- There is no specific software required to be run on the CC3100 Wi-Fi module. However, for the sake of maintaining uniformity across platforms, the CC3100 is programmed with the service pack.

NFC software blocks:

- There are no NFC software blocks in this design.

3.3 BLE Node Software Architecture

Figure 20 shows the software architecture of the BLE node and slave.

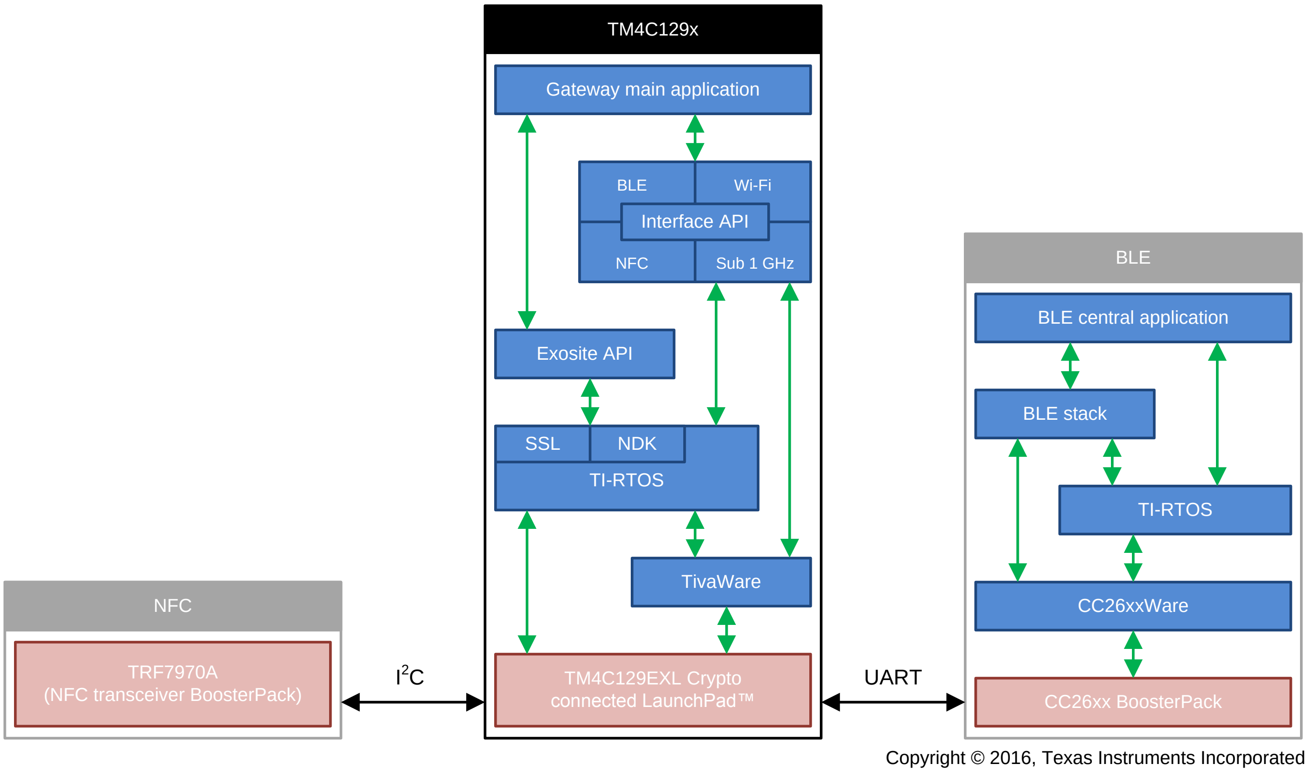


Figure 20. BLE Node Software Architecture Design

TM4C123x software blocks:

- TivaWare C: allows for TM4C hardware register access and serial communications to other hardware through UART and I^{2}C
- Interface API: allows the TM4C123x to communicate to the BLE, NFC hardware mounted on it using the onboard UART and I^{2}C serial peripherals
- TI-RTOS: used for scheduling tasks, which handle communication with
  - Hardware peripherals (BLE, NFC)
  - Command line interface

BLE software blocks:

- TI-RTOS: for general scheduling
  - Manages the command and response handling over the UART
  - Operates the BLE peripheral
- CC26xxWare: performs CC26xx hardware access and the UART operation
- BLE stack: supports the BLE protocol

3.4 Sub-1GHz Node Software Architecture

Figure 21 shows the software architecture of the BLE node and slave.

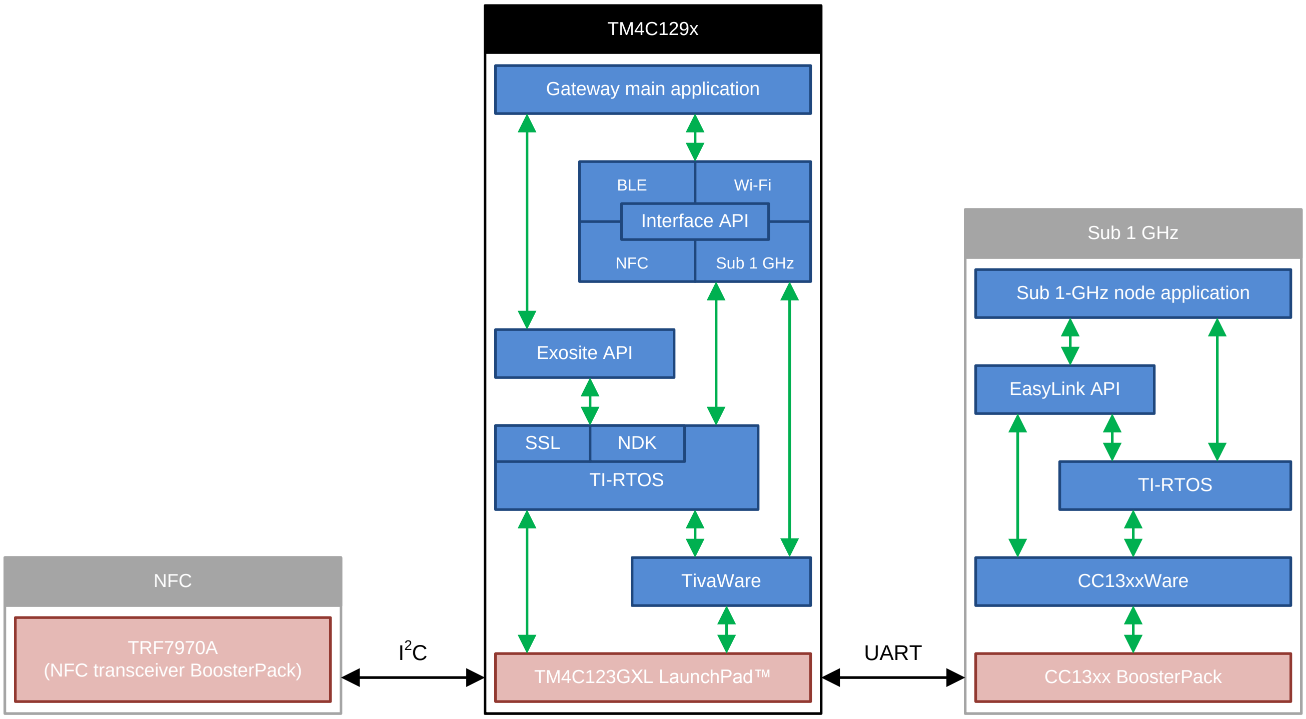


Figure 21. Sub-1GHz Node Software Architecture Design

TM4C123x software blocks:

- TivaWare C: allows for TM4C hardware register access and serial communications to other hardware through UART and I^2C
- Interface API: allows the TM4C123x to communicate to the Sub-1GHz, NFC hardware mounted on it using the on-board UART and I^2C serial peripherals
- TI-RTOS: used for scheduling tasks, which handle communication with
  - Hardware peripherals (Sub-1GHz, NFC)
  - Command line interface

Sub-1GHz software blocks:

- TI-RTOS: for general scheduling
  - Manages the command and response handling over the UART
  - Operates the Sub-1GHz peripheral
- CC13xxWare: performs CC13xx hardware access and the UART operation
- EasyLink API: supports the Sub-1GHz protocol

3.5 Exosite Architecture

3.5.1 SSL/TLS Security

The gateway connects to the Exosite using TI-RTOS NDK. This connection is secured using wolfSSL TLS routines.

3.5.2 Exosite CIK Infrastructure

The TM4C crypto connected LaunchPad EK-TM4C129EXL, which is used as a gateway for this demo, has to be registered on TI's Exosite portal as explained in Section 5.2. When a device is registered with Exosite, that device is allotted a CIK, which is a unique identifier of that device. Only a device having a valid CIK can connect to the Exosite cloud and exchange data.

3.5.3 Data Exchange Mechanism

Exosite can store the values in the form of dataports on the server. These dataports are basically the server's version of variables. The widgets on the portal dashboard (Exosite GUI) are used to modify these dataports or to display the value of these dataports on the same GUI.

The gateway periodically synchronizes with the cloud to achieve the following:

- Uploading local variables' values to the intended dataports on the Exosite server. These local variables contain the SensorTag data or data from BLE and Sub-1GHz nodes.
- Downloading the values of the intended dataports and copying them into local variables. Based on the values received from Exosite, a specific command is sent to the appropriate Wi-Fi or BLE. Sub-1GHz nodes to toggle LEDs, control motor, LED blinking rate, and so on.

Table 12. Dataports Used

<table>
<tr><th>NODE</th><th>DATAPORT NAME</th><th>DATAPORT VALUE FORMAT</th></tr>
<tr><td rowspan="2">Wi-Fi node</td><td>wifi_node1_e2g</td><td>LED2MOD1DIR1SPD30MSV100RFS100RUN2:
• LED—Toggle LED on Wi-Fi node (1: ON, 2: OFF)
• MOD—Change Mode for stepper motor (1: Full-step, 2: Half-step, 3: Micro-step)
• DIR—Change direction of stepper motor (1: Clockwise, 2: Counter-clockwise)
• SPD—Change the speed of stepper motor. (1 to 100)%
• MSV—Provide the value of microsteps if mode is "micro-stepping" (1 to 255)
• RFS—Value of fixed number of steps if it is run as such. [1-999]
• RUN—Choose if the motor is run freely or for fixed no of steps (1: Run freely, 2: Stop, 3: Run for fixed no of steps, 4: Run for fixed no of steps and reverse)</td></tr>
<tr><td>wifi_node1_g2e</td><td>1: Connected to gateway, 0: Disconnected</td></tr>
<tr><td rowspan="2">BLE SensorTag</td><td>ble_sentag_e2g</td><td>1: Connected to gateway, 2: Disconnected</td></tr>
<tr><td>ble_sentag_g2e</td><td>CON0AMT0.000IRT0.000HUM0.000BAR0.00LUX0.000
• CON—Connection status (1: Connected to gateway, 0: Disconnected)
• AMT—Ambient temperature value
• IRT—IR temperature value
• HUM—Humidity value
• BAR—Atmospheric pressure value
• LUX—Luminosity value</td></tr>
<tr><td rowspan="2">BLE node</td><td>ble_node1_e2g</td><td>LDB2ANM30:
• LDB—Toggle LED on BLE node (1: ON, 2: OFF)
• ANM—Change LED blinking rate (1 to 100)%</td></tr>
<tr><td>ble_node1_g2e</td><td>CON0BTA0BTB0TMC0TMF0:
• CON—Connection with gateway status (1: Connected, 0: Disconnected)
• BTA—Button1 press count
• BTB—Button2 press count
• TMC—Junction temperature in Celsius
• TMF—Junction temperature in Fahrenheit</td></tr>
</table>

Table 12. Dataports Used (continued)

NODE	DATAPORT NAME	DATAPORT VALUE FORMAT
Sub-1GHz node 1	sub_node1_e2g	LDB2ANM30: • LDB—Toggle LED on BLE node (1: ON, 2: OFF) • ANM—Change LED blinking rate (1 to 100)%
	sub_node1_g2e	CON0BTA0BTB0TMC0TMF0: • CON—Connection with gateway status (1: Connected, 0: Disconnected) • BTA—Button1 press count • BTB—Button2 press count • TMC—Junction temperature in Celsius • TMF—Junction temperature in Fahrenheit
Sub-1GHz node 2	sub_node2_e2g	LDB2ANM30: • LDB—Toggle LED on BLE node (1: ON, 2: OFF) • ANM—Change LED blinking rate (1 to 100)%
	sub_node2_g2e	CON0BTA0BTB0TMC0TMF0: • CON—Connection with gateway status (1: Connected, 0: Disconnected) • BTA—Button1 press count • BTB—Button2 press count • TMC—Junction temperature in Celsius • TMF—Junction temperature in Fahrenheit

NOTE: 'e2g' indicates that this dataport's value is meant to be sent from Exosite to the gateway. Similarly, 'g2e' indicates that the dataport value on Exosite is to be updated by the gateway.

4 Software Setup

4.1 Software Requirements

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)
- CC2650 BLE Stack-2 (http://www.ti.com/tool/ble-stack-archive)
- TI-RTOS for CC2650 v2.11.01.09. (C26xxWare is included) TI RTOS Download Link
- TI-RTOS for CC13xx v2.14.03.28. (C13xxWare is included) TI RTOS Download Link
- TivaWare_C v2. 1.1.71 (http://www.ti.com/tool/sw-tm4c)
- TI-RTOS for TIVA v2.14.00.10 TI RTOS Download Link
- wolfSSL for TI-RTOS (https://github.com/wolfSSL/wolfssl)

NOTE: The BLE demonstration is not compatible with BLE-STACK-2-1 (http://www.ti.com/tool/ble-stack).

The BLE demonstration is not compatible with tirtos_simplelink version 2.12.x, 2.13.x, or 2.14.x due to the UART driver changes in these releases. TI recommends using 2.11.01.09 for this demonstration inspite of installing version 2.14.03.28 that supports both CC13xx and CC26xx Family.

TI recommends installing these packages in the default location under C:\ti to avoid making any changes in the CCS project.

4.2 Building Software Stack

The required software with CCS projects for the demonstration of this application including both Gateway and the Node software can be downloaded from TIDM-TM4C129XGATEWAY Software. The projects under following folders are necessary to build specific subsystems:

- Gateway
- BLE_Node
- Sub1GHz_Node
- WIFI_Node

For an example, follow these steps to import projects into the CCS workspace to build the binaries required for the application:

1. Go to File → Import → CCS Project.
2. Browse folders to {TIDM_TM4C_Gateway_WiFi_BLE_Sub1GHz}\Project_Source\Gateway\TM4C129x\}.
3. Import all the projects into the workspace.
4. Build all the projects except Gateway_Main_App because Gateway_Main_App project depends on .lib output of other projects under folder Gateway. Then, build Gateway_Main_App. Executables can be found in debug folder of Gateway_Main_App.

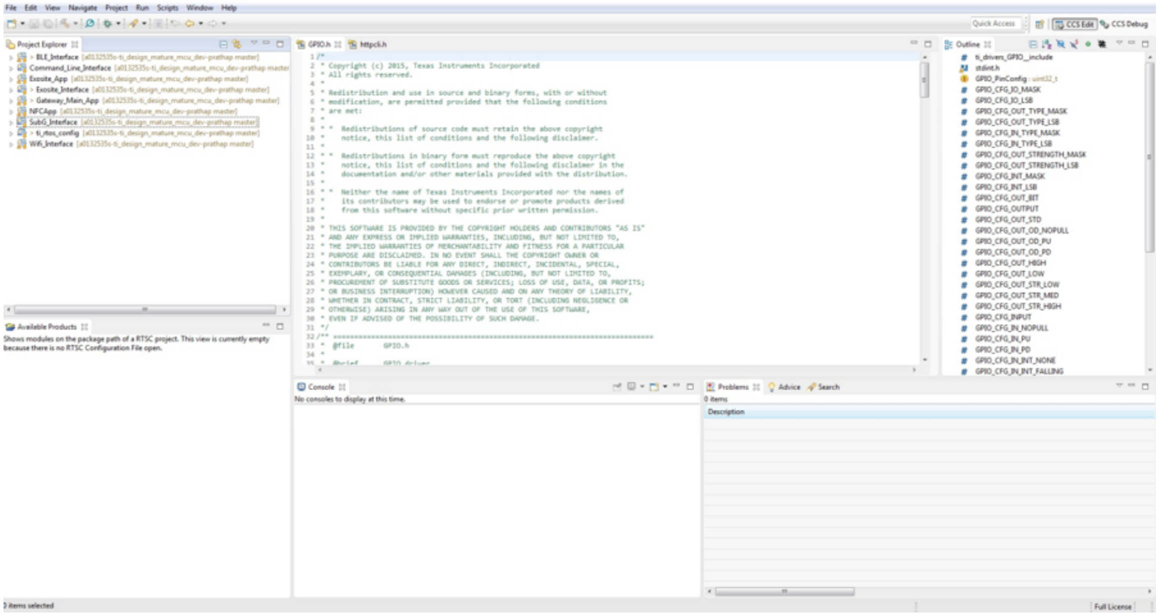


Figure 22. Importing CCS Projects for Gateway

4.3 Demo Executables List

The executable are also distributed along with the project source under folder Executables

Table 13. Gateway Executables

DEVICE	NAME OF EXECUTABLE
GATEWAY EXECUTABLES	
TM4C129EXL	Gateway_Main_App.out
CC2650EM	TM4C_CC26xx_Demo_CentralStack.out TM4C_CC26xx_Demo_Central.out
CC1310EM	CC13xx_Master.out
CC3100	Latest service pack
WI-FI NODE EXECUTABLES	
TM4C123GXL	wifi_microstepping_stepper_motor.out
CC3100	Latest service pack
BLE NODE EXECUTABLES	
TM4C123GXL	TM4C_BLE_NFC_Node.out
CC2650EM	TM4C_CC26xx_Demo_PeripheralStack.out TM4C_CC26xx_Demo_Peripheral.out
SUB-1GHz NODE EXECUTABLES	
TM4C123GXL	TM4C_SubG_NFC_Node.out
CC1310EM	CC13xx_Node.out

5 Installing the Demo

5.1 Setting up Subsystems

Table 14. Setting up Subsystems

PART	TO DO
GATEWAY SUBSYSTEM	
TM4C129EXL (Crypto Connected LaunchPad)	Flash <Gateway_Main_App.out> using CCS/Uniflash.
CC3100 (Wi-Fi BoosterPack)	Use Uniflash to program the CC3100 with the latest service pack. Visit the following URL and search for "Service Pack Programming". https://www.processors.wiki.ti.com/index.php/CC31xx_%26_CC32xx_UniFlash_Quick_Start_Guide
CC2650EM (BLE Device)	Use Uniflash/RFProgrammer and SmartRF06 to program TM4C_CC26xx_Demo_CentralStack.out and then TM4C_CC26xx_Demo_Central.out onto the CC2650 EM device.
CC13100EM (Sub-1GHz Device)	Use Uniflash/RFProgrammer and SmartRF06 to program CC13xx_Master.out onto the CC1310 device.
WI-FI NODE SUBSYSTEM	
TM4C123x (TIVA-C LaunchPad)	Flash wifi_microstepping_stepper_motor.out using CCS/Uniflash.
CC3100 (Wi-Fi BoosterPack)	Use Uniflash to program the CC3100 with the latest service pack. Visit the following URL and search for "Service Pack Programming". https://www.processors.wiki.ti.com/index.php/CC31xx_%26_CC32xx_UniFlash_Quick_Start_Guide
BLE NODE SUBSYSTEM	
TM4C123x (TIVA-C LaunchPad)	Flash TM4C_BLE_NFC_Node.out using CCS/Uniflash.
CC2650EM (BLE Device)	Use Uniflash or RF Programmer and SmartRF06 to program first TM4C_CC26xx_Demo_PeripheralStack.out and then TM4C_CC26xx_Demo_Peripheral.out onto the CC26750 EM device
SUB-1GHz NODE SUBSYSTEM	
TM4C123x (TIVA-C LaunchPad)	Flash TM4C_SubG_NFC_Node.out using CCS/Uniflash.
CC2650EM (BLE Device)	Use Uniflash or RF Programmer and SmartRF06 to program CC13xx_Node.out onto the CC1310 EM device.

5.2 Setting up Exosite

The Exosite cloud platform provides a dashboard to interact with the gateway and the slaves connected to the gateway. The user can send commands to execute tasks like toggling LED on nodes, drive a stepper motor connected to a node, and collect data from the nodes. The devices that run the gateway code (TM4C129x) need to be registered with Exosite because gateway handles all the communication with the Exosite cloud portal. The "MAC Address" of the gateway device (printed on the back of TM4C129x) is required for registration.

Cloud service is a trivial component in this demo, so go though the information provided in Section 1.11 before proceeding with setting up Exosite for the demo.

Follow these steps to register the device (TM4C129) to use as a gateway to the Exosite portal:

1. Go to https://ti.exosite.com.
2. Register an account for first-time users and log in.
3. Click on "Devices" in the left-hand menu and click on the "Add Device" button.
4. Select the "TM4C based Secure Cloud Connected IoT Gateway" device to add from the "Supported Devices" drop-down list. Then click "Continue".
5. Fill in the details of the board MAC, name, and location. Then click "Continue". Take note of the CIK being displayed and then click "Quit".

   The device is now registered with the Exosite server, but it still needs to be activated. The software running on the gateway will achieve this once it is up and running. The following sections talk about the connection, activation and execution of the demo.
6. Click on "Dashboards" in the left-hand side menu. Then, click on "IoT Gateway" under the "Portal Dashboards" section. The dashboard as shown in Figure 23 should appear.

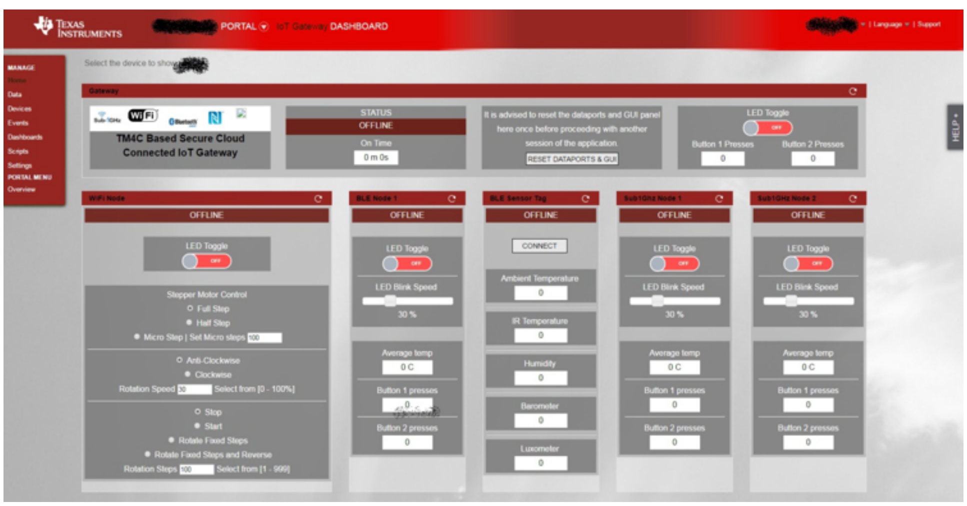


Figure 23. Exosite Dashboard Associated With Gateway Device Registered

NOTE: For first time users of Exosite with TM4C crypto connected launchpad Secure IoT Demo training video under section 2 will be a good start. This video contains information on how to connect the board to Exosite, UART console settings, proxy settings etc in detail.

Some countries and/or firewalls may block access to some internet based content such as google services. If this content is blocked, the Exosite based portion of the project will not work as intended. In such cases the demo can be executed through command line interface.

6 Demo Execution

All the individual subsystems of the demo must be set up and initialized before moving on to execute the demo.

6.1 Connecting Gateway to Cloud Using Ethernet

1. Power off the TM4C129x CCLP.
2. Connect the CC3100 Wi-Fi BoosterPack on the BoosterPack-2 interface on the TM4C129 CCLP. See Section 2.2.1 to verify the header connections.
3. Connect the EM Adapter BoosterPack to BoosterPack-1 Interface on TM4C129 CCLP. See Section 2.2.1 to verify the header connections. Connect the CC2650 BLE Device to the EM Adapter BoosterPack.
4. Connect the TRF7970A NFC Boosterpack to the EM Adapter BoosterPack, which is already connected to the TM4C129 CCLP in the previous step. See Section 2.2.1 verify the header connections.

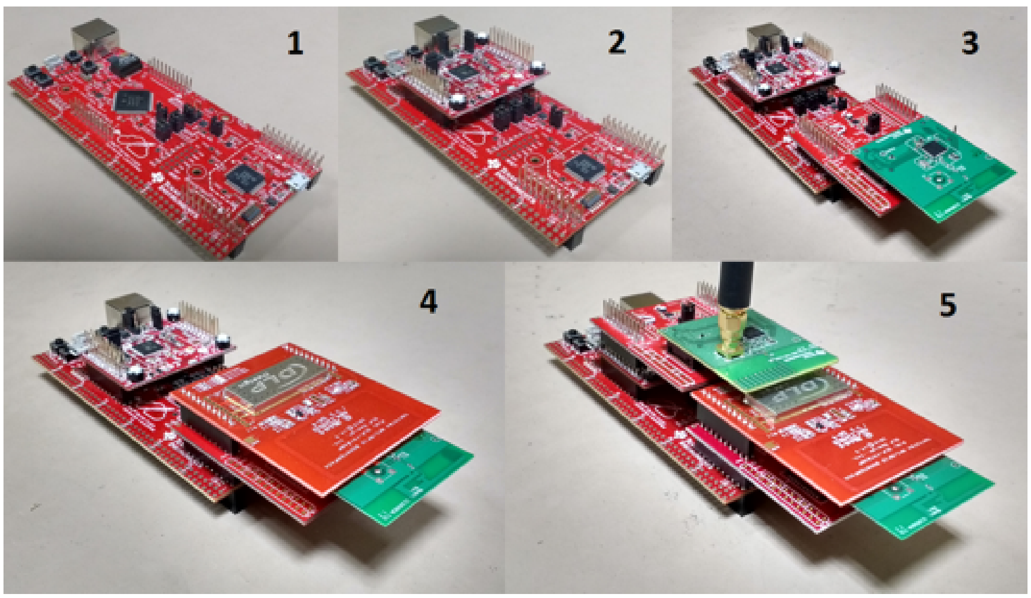


Figure 24. Connecting All Hardware Components of the Gateway

5. Connect the EM Adapter BoosterPack to BoosterPack-2 Interface on TM4C129 CCLP over the CC3100 BoosterPack, which was connected in Step 2. See Section 2.2.1 to verify the header connections. Connect the CC1310 Sub-1GHz device to this EM Adapter BoosterPack. Also mount the antenna on the Sub-1GHz device.
6. Connect an Ethernet LAN cable to connect the gateway to a LAN port with working internet connection.
7. Power on the board by connecting the TM4C129 CCLP to a power source using a USB cable to see the LED2, LED3, and LED4 with all three flashing at regular intervals. LED1 should start flashing after a few seconds once the gateway is connected to cloud. If LED1 does not flash or if the application is run behind a proxy network, configure Proxy and NTP. See Section 6.7 for command line options.
8. Now go to the dashboard "IoT Gateway", which should display the gateway as "ONLINE" as shown in Figure 25 if connected properly.

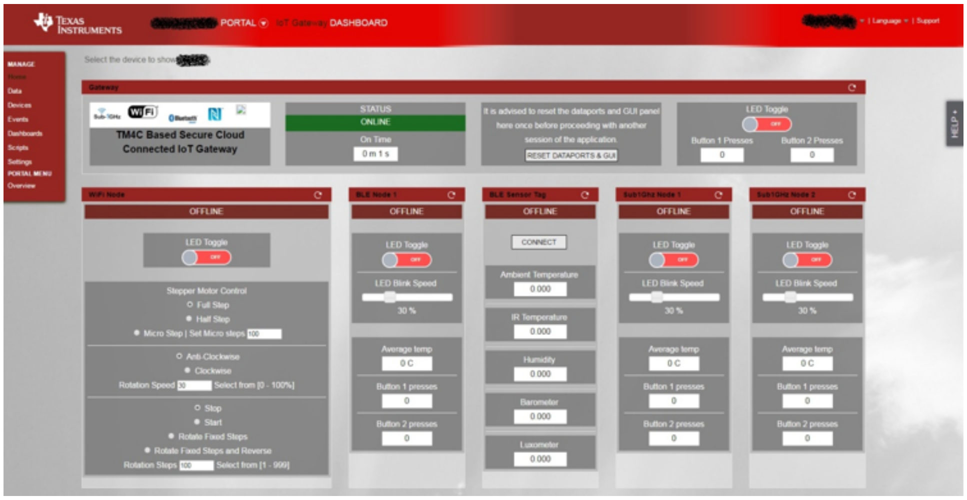


Figure 25. IoT Gateway Dashboard Showing Gateway ONLINE

6.2 Connecting Wi-Fi Node to Gateway

1. Make sure the TM4C123 LP is powered off before proceeding.
2. Connect the TM4C123 LP and CC3100 Wi-Fi BoosterPack to the TM4C123 swizzle adapter board at their designated BoosterPack interfaces. See Section 2.2.2 to verify header connections.
3. Connect the RF430 NFC Tag BoosterPack to the TM4C123 swizzle adapter board at its designated location (recommended as shown in Figure 26). See Section 2.2.2 to verify header connections.

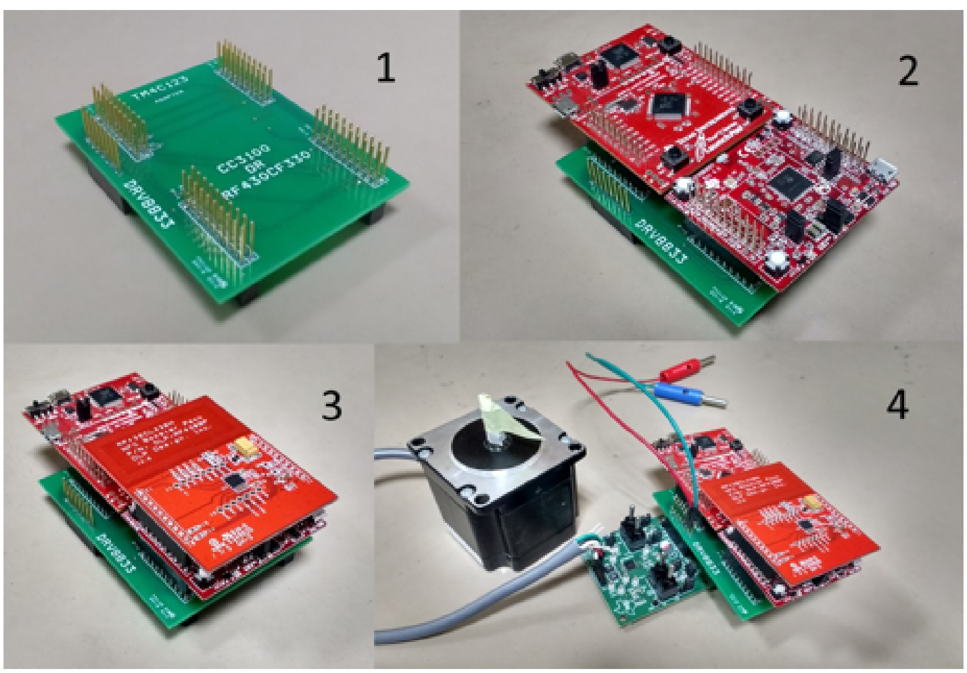


Figure 26. Connecting all Hardware Components of the Wi-Fi Node

4. Now connect the DRV8833 stepper motor drive to the TM4C123 swizzle adapter board as shown in Figure 27. Connect the +ve and –ve terminals of the external power supply to the DRV883 board using the pins on the swizzle board. The DRV883 can also be powered from the 5-V pin in the TM4C123 LaunchPad header as shown with the red lines in Figure 27, provided the USB power source can supply up to 2 A. Also, refer to the stepper motor's user guide to connect the correct terminals of the stepper motor (A, $\overline{A}$, B, $\overline{B}$) to their corresponding pins on DRV8833 board. See Section 2.2.2 to verify header connections.

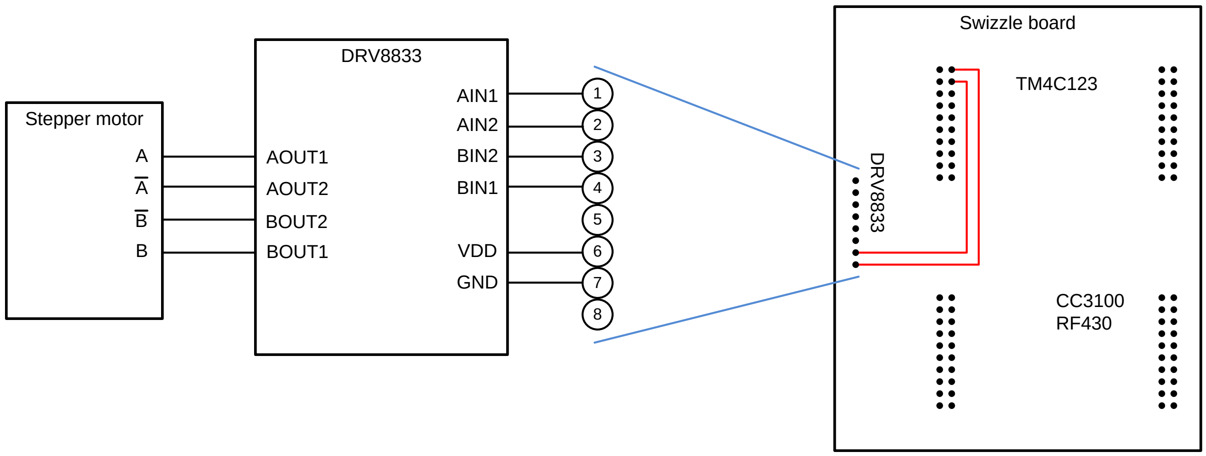


Figure 27. Connecting Swizzle Board, Stepper Motor, and External Power Supply to the DRV8833 Driver Board

5. Supply power to the TM4C123 LP by connecting it to a power source using the USB cable. The white color LED should flash once.
6. Tap the NFC tag on Wi-Fi node subsystem with the NFC transceiver on the gateway subsystem to exchange Wi-Fi credentials. The LED on the TM4C123x LP turns blue to indicate that connection to the gateway is in progress. Once the node connects to the gateway, this LED will turn to green; otherwise, if the connection is not successful, the LED turns red. Tap again to retry.

   On the gateway side, the LED3 should get switched on (green color) to indicate that the Wi-Fi slave is connected.

6.3 Connecting BLE Node to Gateway

1. Make sure that the TM4C123x LP is switched off before proceeding.
2. Connect the EM Adapter BoosterPack to the TM4C123x LP as shown in Figure 28.
3. Connect the CC2650 to the header present on the EM Adapter BoosterPack as shown.
4. Connect the RF430C330H NFC Tag to the EM Adapter BoosterPack as shown in Figure 28 (recommended).

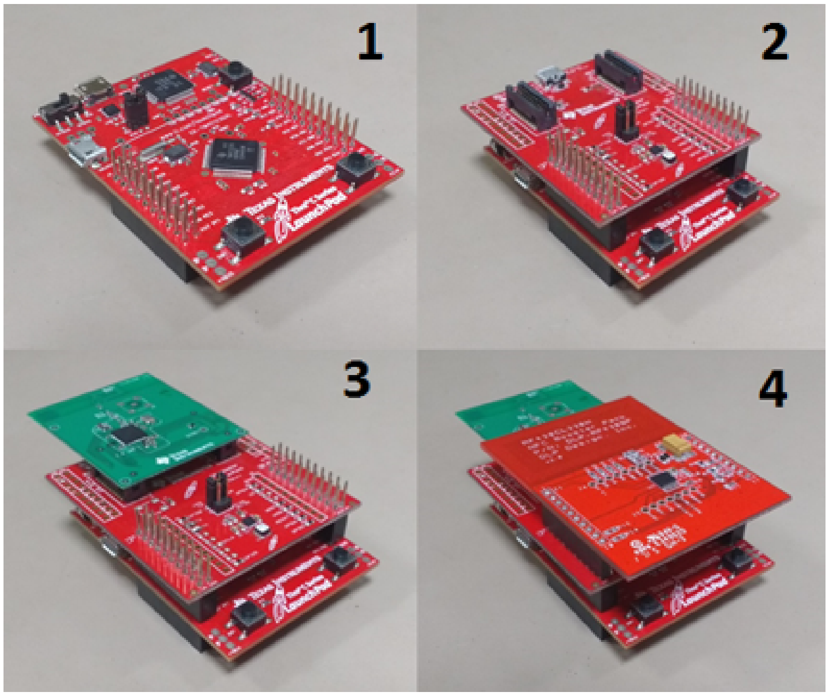


Figure 28. Connecting All Hardware Components of the BLE Node

5. Supply power to the TM4C123 LP by connecting it to a PC using the USB cable. The green LED should now flash at regular intervals.
6. Tap the NFC tag on the BLE node subsystem with the NFC transceiver on the gateway subsystem to exchange credentials. The LED on TM4C123x LP turns from green to blue (and continues to blink at regular intervals) to indicate that connection has been established with the gateway.

   On the gateway side, the LED2 should turn on (green color) to indicate that the BLE slave is connected.

6.4 Connecting SensorTag to Gateway

Connect the SensorTag to the gateway with the Exosite GUI and IoT gateway dashboard by clicking on the "Connect" button in the SensorTag widget. In a few seconds, the widget should show the SensorTag status as ONLINE. Before connecting to the gateway using Exosite GUI, the BLE SensorTag must be in advertising mode.

To connect the SensorTag using command line, see Section 6.7.

6.5 Connecting Sub-1GHz Nodes to Gateway

1. Make sure that the TM4C123x LP is switched off before proceeding.
2. Connect the EM Adapter BoosterPack to the TM4C123x, and then connect the CC1310 to the header present on the EM Adapter BoosterPack as shown in Figure 29.
3. Connect the RF430C330H NFC Tag to the EM Adapter BoosterPack as shown (recommended).

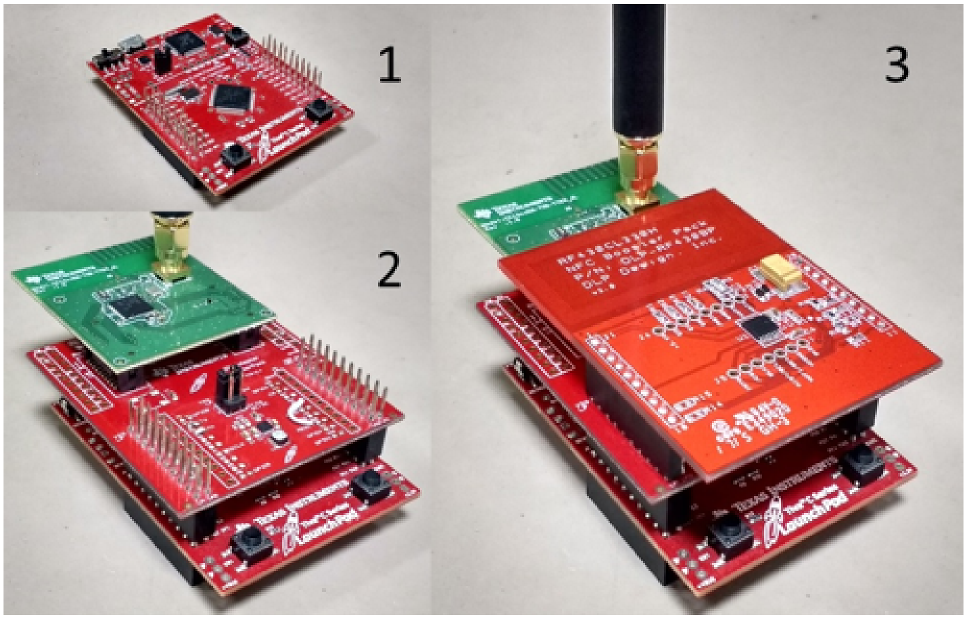


Figure 29. Connecting All Hardware Components of the Sub-1GHz Node

4. Supply power to the TM4C123 LP by connecting it to a power source using the USB cable. The green LED should now flash at regular intervals.
5. Tap the NFC tag on the Sub-1GHz node subsystem with the NFC transceiver on the gateway subsystem to exchange credentials.
6. The LED on the TM4C123x LP turns from green to blue (and continues to blink at regular intervals) to indicate that the connection has been established with the gateway.
7. On the gateway side, the LED3 switches on permanently to indicate that the Sub-1GHz slave is connected.

6.6 Using GUI to Control Nodes From Exosite

Once all the nodes are connected to the gateway and the gateway is connected to Exosite properly, then the IoT_Gateway dashboard on ti.exosite.com should resemble Figure 30. The widgets representing the corresponding nodes should display under the ONLINE banner.

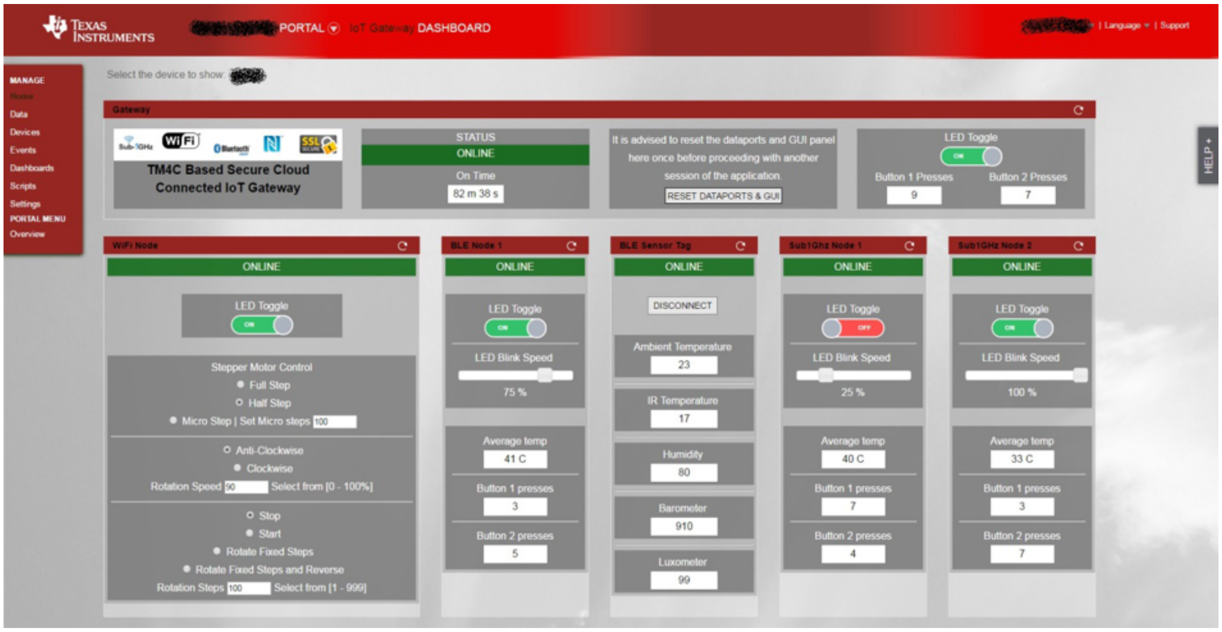


Figure 30. Exosite Dashboard With All Connected Widgets

6.6.1 Wi-Fi Node

The various controls on the Wi-Fi widget are as follows:

- LED toggle switch
- Motor mode switch [Half step | Full step | Micro-step]
- Micro-step value input [1 to 256] steps
- Motor direction switch [Clockwise | Anti-Clockwise]
- Motor speed input [0 to 100]%
- Motor run configuration switch [Start | Stop | Rotate fixed steps | Rotate fixed steps and reverse]
- Fixed step rotation value input [0 to 999]

6.6.2 BLE Node

The various controls on the BLE widget are as follows:

- LED toggle switch
- LED toggle rate input (animation value) [0 to 100]%
- Real-time TM4C123x junction temperature display
- Real-time button-press display

6.6.3 SensorTag

The various controls on the SensorTag widget are as follows:

- Connect or disconnect the SensorTag from the gateway
- Real-time sensor data display for Temperature, IR Temperature, Humidity, Atmospheric Pressure, and Luminosity

6.6.4 Sub-1GHz Node

The various controls on the Sub-1GHz widget are as follows:

- LED toggle switch
- LED toggle rate input (animation value) [0 to 100]%
- Real-time TM4C123x junction temperature display
- Real-time button-press display

6.7 Command Line Interface

The gateway demonstration can be performed through command line interface featuring the command set listed in Table 15.

To use the following commands, follow these steps:

1. Power up the gateway by connecting it to a PC using USB cable.
2. Open a console terminal using software like Tera Term or RealTerm or Putty with settings [Serial COM Port Connection | Bitrate 115200].
3. Press "Reset" on the gateway device to restart the gateway. The console will display some info as shown in Figure 31.

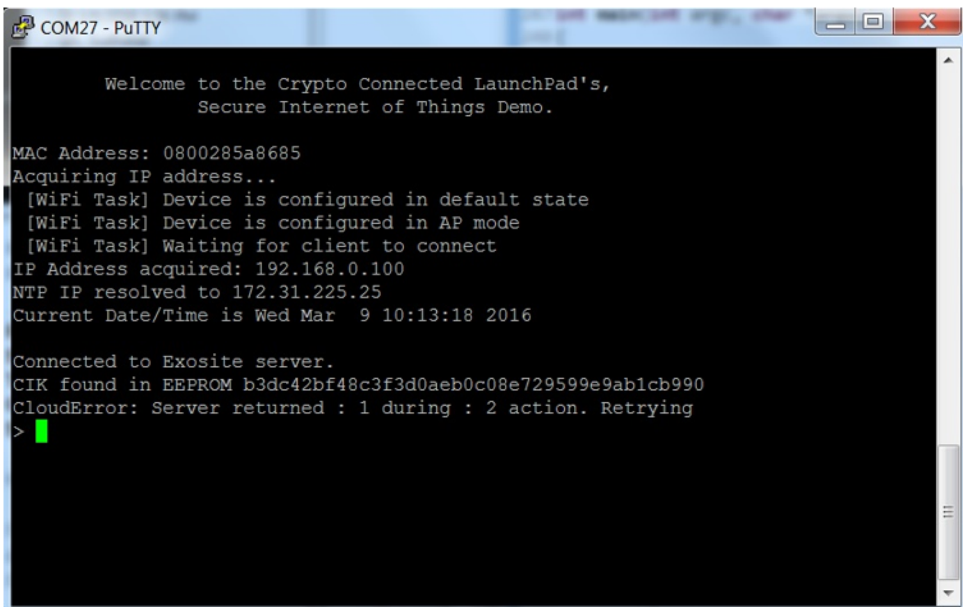


Figure 31. Gateway Console Debug Output

4. Hit "Enter" to see a command prompt as shown in Figure 32. See Section 6.7.1 for a list of valid commands accepted by the gateway.

6.7.1 Generic CLI Commands

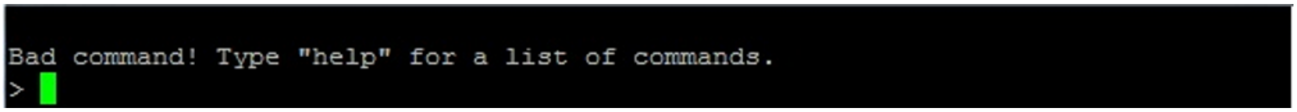


Figure 32. Command Prompt

Table 15. Generic CLI Commands

COMMAND	USAGE	DESCRIPTION
help	> help	Display list of commands
[h, ?]	> h > ?	Aliases for help
activate	> activate	Get a CIK from exosite
clear	> clear	Clear the display
connect	> connect	Tries to establish a connection with exosite
getmac	> getmac	Prints the current MAC address
ntp	> ntp *<NTP Server IP Address>*	Tries to connect to the provided IP during start-up to sync time.
proxy	> proxy *<Proxy IP Address> <Proxy Port Number>*	Set or disable a HTTP proxy server
led	> led on > led off	Toggle LED D1 on gateway. Type "led help" for more info.
wifi	> wifi ? > wifi <option1><option2> and so on	WiFi node control command. Type "wifi ?" or "wifi help" for usage info.
ble	> ble <option1> wifi ? > ble ?	BLE node control command. Type "ble ?" or "ble help" for usage info.
subg	> subg <option1><option2> and so on > subg ?	Sub-1GHz node control command. Type "subg ?" or "sub help" for usage info.

6.7.2 Controlling Nodes Using CLI Commands

The application demo can be executed locally in case the connection to the cloud is absent. Commands to achieve that are described in Table 16:

NOTE: Contents in the square brackets [ ] indicate the options available for the respective commands separated with a comma.

Table 16. Wi-Fi Node CLI Commands

COMMAND	DESCRIPTION
wifi led [on, off]	Toggle LED on BLE Node
wifi motor speed [0 to 100]	Change the speed of motor
wifi motor dir [clock, anti-clock]	Change the direction of motor
wifi motor mode [1 to 255]	Change the motor mode [1: Full Step, 2: Half Step, [3 to 255]: Micro Step]
wifi motor rfs [1 to 999]	Run fixed number of steps [Min: 1, Max: 999]
wifi motor rfr [1 to 999]	Run fixed number of steps and then reverse [Min: 1, Max: 999]
wifi motor run	Run the motor freely
wifi motor stop	Stop the motor

Table 17. BLE Node CLI Commands

COMMAND	DESCRIPTION
ble sensor-tag connect	Connect to SensorTag [SensorTag should be advertising]
ble sensor-tag disconnect	Disconnect form the SensorTag
ble sensor-tag status	Get the current status of SensorTag data
ble node led [on, off]	Toggle LED on BLE node
ble node animation [0 to 100]	Change the rate of blinking of led on BLE node
ble node status	Get the current status of Data coming from BLE node

Table 18. Sub-1GHz Node CLI Commands

COMMAND	DESCRIPTION
sub [node1, node2] led [on, off]	Toggle LED on one of the Sub-1GHz nodes
sub [node1, node2] animation [0 to 100]	Change the rate of blinking of led on one of the Sub-1GHz nodes
sub [node1, node2] status	Get the current status of data coming from one of the Sub-1GHz nodes

7 Design Files

7.1 Schematics

To download the schematics, see the design files at TIDM-TM4C129XGATEWAY.

7.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDM-TM4C129XGATEWAY.

7.3 PCB Layout Recommendations

Any additional note you think the customer would need to layout this board; also add details on the reasoning behind your layout (form factor, heat distribution, and so on.)

7.3.1 Layout Prints

To download the layer plots, see the design files at TIDM-TM4C129XGATEWAY.

7.4 Altium Project

To download the Altium project files, see the design files at TIDM-TM4C129XGATEWAY.

7.5 Gerber Files

To download the Gerber files, see the design files at TIDM-TM4C129XGATEWAY.

7.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDM-TM4C129XGATEWAY.

8 Software Files

To download the software files, see the design files at TIDM-TM4C129XGATEWAY.

9 References

1. Texas Instruments, *Tiva™ TM4C123GH6PM Microcontroller*, Datasheet (SPMS376)
2. Texas Instruments, *DRV8833 Dual H-Bridge Motor Driver*, DRV8833 Datasheet (SLVSAR1)
3. Texas Instruments, *CC3100 SimpleLink™ Wi-Fi® and IoT Solution Getting Started Guide*, User's Guide (SWRU375)
4. Texas Instruments, *ARM® Cortex®-M4F Based MCU TM4C123G LaunchPad™ Evaluation Kit*, EK-TM4C123GXL Product Page (http://www.ti.com/tool/EK-TM4C123GXL)
5. Texas Instruments, *DRV8833 Evaluation Module User's Guide* (SLVU498)
6. Texas Instruments, *SimpleLink™ Wi-Fi® CC3100 wireless network processor BoosterPack™ plug-in module* (http://www.ti.com/tool/cc3100boost)
7. Texas Instruments, *TivaWare™ Sensor Library*, User's Guide (SPMU371)
8. Texas Instruments, *SimpleLink Wi-Fi CC3100 SDK* (http://www.ti.com/tool/cc3100sdk)
9. Texas Instruments, *Stellaris® In-Circuit Debug Interface (ICDI) and Virtual COM Port Driver Installation Instructions*, Quick Start Guide (SPMU287)
10. Texas Instruments, *CC31xx & CC32xx UniFlash Quick Start Guide*, TI Wiki (http://processors.wiki.ti.com/index.php/CC31xx_%26_CC32xx_UniFlash_Quick_Start_Guide)
11. Texas Instruments, *High Resolution Microstepping Driver With the DRV88xx Series*, Application Report (SLVA416)
12. Texas Instruments, *TM4C1294x Wi-Fi Enabled IoT Node*, TIDM-TM4C129XWIFI Design Guide (TIDU992)

10 About the Author

SUDHAKAR SINGH is a software engineer in the Performance Microcontroller group at Texas Instruments, where he primarily works on TM4C software development, customer support, and reference design development. Sudhakar received his bachelor of engineering in computer science and engineering from the PEC University of Technology, India.

Revision A History

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

Changes from Original (August 2016) to A Revision	**Page**
• Changed title from *TM4C Based Secure Cloud Connected IoT Gateway for BLE, Wi-Fi, and Sub-1GHz Nodes*	1