## TI Designs BLE Enabled IoT Node on High-Performance Microcontrollers

# Texas Instruments

### **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-TM4C129XBLE	Desi
TM4C1294NCPDT	Proc
SimpleLink CC2650	Proc
EK-TM4C1294XL	Tool
CC2650EMK	Tool
BOOST-CCEM ADAPTER	Tool
BLE-STACK-2	Tool
CC2650DK	Tool

TI E2E™ Community Design Folder Product Folder Product Folder Tools Folder Tools Folder Tools Folder Tools Folder Tools Folder

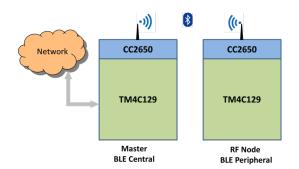
ASK Our E2E Experts WEBENCH® Calculator Tools

#### **Design Features**

- The Master is a TM4C1294 MCU and the CC2650 as BLE Central.
- The Slave is a TM4C1294 MCU and the CC2650 as BLE Peripheral.
- LWIP-Based Web Server Runs on the Master Side TM4C1294 MCU.
- The BLE Stack Runs on CC2650.
- The TM4C1294 Works as the Host Processor Performing a Demo Application.
- HTML Code Remotely Controls Slave Operation of the TM4C1294 From a Web Browser.
- Code Composer Studio<sup>™</sup> is for Development and Debugging.
- TI RTOS is 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.

Code Composer Studio, LaunchPad, SimpleLink, BoosterPack, TivaWare, Tiva are trademarks of Texas Instruments. Stellaris is a registered trademark of Texas Instruments. ARM, Cortex are registered trademarks of ARM Limited. *Bluetooth* is a registered trademark of Bluetooth SIG. ZigBee is a registered trademark of Zigbee Alliance. All other trademarks are the property of their respective owners.

ÆΑ



#### 1 System Description

Bluetooth<sup>®</sup> low energy, also known as Bluetooth Smart (BLE), is one of the most common low-power wireless connectivity technologies. BLE is the intelligent, power-friendly version of Bluetooth wireless technology. While the power-efficiency of BLE is perfect for devices that run off a tiny battery for long periods, the magic of BLE is its ability to work with an application on a smartphone or tablet you already own. BLE lets developers and OEMs easily create solutions that work with the billions of Bluetooth-enabled products already in the market.

This reference design shows how to create a BLE node using the TM4C129 high-performance microcontroller and the single-mode CC2650. For more information about this BLE node application, refer to BLE-Enabled IoT Node With High-Performance MCU Reference Design.

The software accompanying this design works on an EK-TM4C1294XL LaunchPad<sup>™</sup> integrated with a CC2650EMK.

TIRTOS schedules various tasks. TI recommends using RTOS to distribute the load and make the application easily scalable.

#### 1.1 TM4C1294NCPDT

The TM4C1294NCPDT device is a 120-MHz high-performance microcontroller with 1MB of 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 high-speed USB2.0 digital interface. With the integration of numerous low-to-mid speed serials (up to 4 million samples per second [MSPS]), a 12-bit ADC, and motion control peripherals, the TM4C1294NCPDT microcontroller is ideal for use with industrial communication equipment applications to Smart Energy or Smart Grid applications.

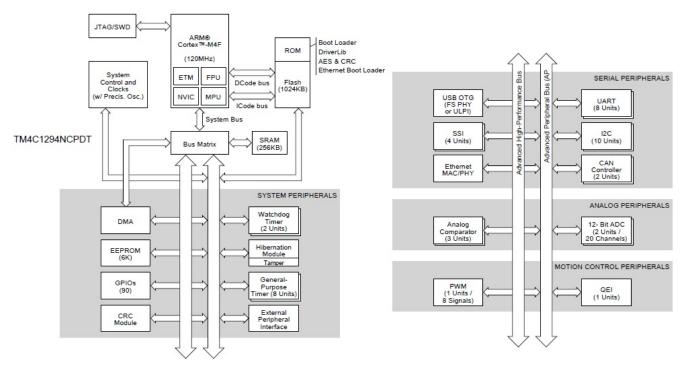


Figure 1. TM4C1294NCPDT Microcontroller High-Level Block Diagram



#### System Description

#### 1.2 CC2650

The CC2650 is a cost-effective, ultralow 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 48MHz as the main processor and has a rich peripheral feature set, including an ultralow power sensor controller. The ultralow 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 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.

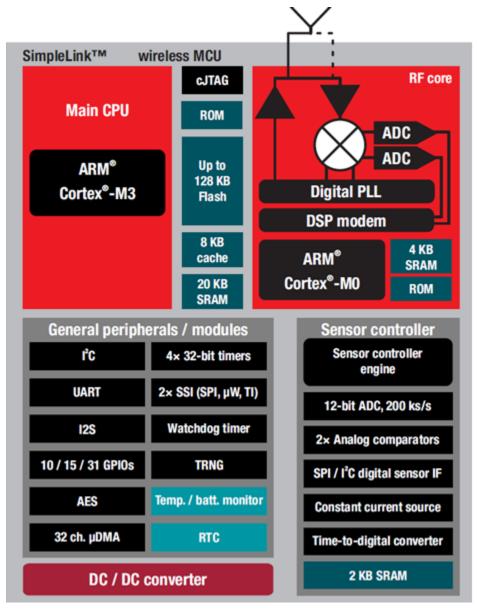


Figure 2. CC2650 Hardware Overview



System Description

#### 1.3 TM4C129 and CC265 Interface

Figure 3 illustrates the interface between the TM4C1294 and the CC2650. The TM4C1294 communicates to the CC2650 through UART. A simple command response protocol is implemented for this demonstration. UART0 is used in CC2650 and UART7 is used in TM4C of this demonstration.

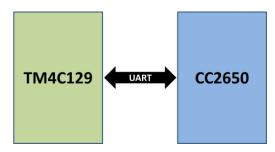
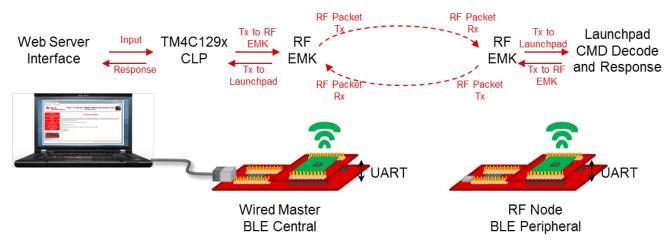


Figure 3. TM4C129 and CC2650 Interface Overview

#### 2 System Functionality Block Diagram

The TM4C BLE node can be configured in two modes: central and peripheral. The wired master node acts as BLE central and RF node acts as BLE peripheral. For this demonstration, the CC2650 is configured in single-processor mode and the TM4C acts as the application controller. The demo BLE profiles and services are in the CC2650. Network processor architecture is beyond the scope for this software design.





### 2.1 TM4C BLE Node as a Central

4

The TM4C performs the following tasks:

- Runs LWIP Ethernet-based Web server.
- Displays assigned IP addresses through UART0.
- Receives user commands requested through a hosted Web page.
- Sends user commands to the CC2650 device through UART, and waits for a response.
- Responds to the web page http request for the selected demo command.



The CC2650 performs the following tasks:

- Waits for commands from the TM4C through UART.
- Discovers and connects to the BLE Peripheral node after receiving a Connect command.
- Interacts with the BLE peripheral node depending on the requested command. Read and Write characteristics value requests are sent to the BLE peripheral node.
- Responds to the TM4C with the data read from the TM4C BLE peripheral node.

### 2.2 TM4C BLE Node as a Peripheral

The CC2650 performs the following tasks:

- Advertises primary characteristics for connection.
- Connects to the TM4C BLE central on request.
- Receives Read or Write characteristics value request coming from the TM4C BLE central.
- Sends user commands to the TM4C device through UART, and waits for a response.
- Responds to the Read or Write characteristic value request coming from the TM4C BLE central

The TM4C performs the following tasks:

- Receives demo commands coming through the BLE peripheral.
- · Performs actions or tasks associated with the user demo command.
- Responds with the appropriate data or acknowledgment to the CC2650 BLE peripheral through TM4C UART7.

Quesse         Diry         Under Quesse         Value (Qer Quess)         Cost of Variant Prime         Permission         Permission         Permission         Permission           10         00200         QUESSE	handle	Туре	_			Application	GATT Server	_
18         0.283         GATOMPACTE_UND         0 [0 [0 monthms notify only]) 10 [0 (0 UND 0.4710)         D. DOGULED         Net / Wire         Ne			Type (#DEFINE)	Value (default)	Local Parameter Name			Description
10         0.2803         0.ATT_OARACTER_UD         1F 0 Quancie bc01911         0.TF 0 Quancie bc01911 <td>15</td> <td>0x2800</td> <td>GATT_PRIMARY_SERVICE_UUID</td> <td></td> <td></td> <td></td> <td>Read</td> <td></td>	15	0x2800	GATT_PRIMARY_SERVICE_UUID				Read	
IndexIndexInternational (International)International (International)International (International)International 	16	0x2803	GATT CHARACTER LIND				Read	
17         0xFP0         TD_TOGGLELED_UND         0 (1 byte)         TD_TOGGLELED         Nead / White         Nead / White <t< td=""><td>10</td><td>072000</td><td></td><td></td><td></td><td></td><td>1 loud</td><td></td></t<>	10	072000					1 loud	
Image: Control Contro Control Control Control Control Control Control Control Control C	17	0xEED1		0 (1 hvte)	TD TOGGLELED	Read / Write	Notify	LED Toggle
10         002202         CHT_CLENT_CARR_CRE_CUUD         CDD Toggle * (10 bytes)         PRRM         CDD Toggle * (10 bytes)           10         0.2201         GATT_CHAR_USER_DESC_LUD         TLD Toggle * (10 bytes)         Read         Read         CDD Toggle * (10 bytes)           20         0.2203         GATT_CHAR_CER_ULD         0 (10 bytes)         TD_SETLEDSPEED         Read         SetLED Speed           21         0.6702         TD_SETLEDSPEED_LUD         0 (10 bytes)         TD_SETLEDSPEED         Read         SetLED Speed           22         0.2202         GATT_CHAR_CER_ULD         0 (10 bytes)         TD_SETLEDSPEED         Read         SetLED Speed           23         0.2203         GATT_CHAR_CER_ULD         0 (10 bytes)         TD_SETLEDSPEED         Read         SetLED Speed           24         0.2203         GATT_CHAR_CER_ULD         0 (10 bytes)         TD_GETBUTTONIOUNT         Read         Control SetLED Speed           25         0.4753         TD_GETBUTTONIOUNT_LD         0 (11 bytes)         TD_GETBUTTONIOUNT         Read         Control SetLED Speed           26         0.4203         GATT_CHAR_USER_DESC_UDD         Col Buton 1 Press         Control Control SetLED SetL		UXIT DT		0 (1 5)(0)			r totar y	Characteristic Value
10         0.4201         GATT_GHAR_USER_DESC_UND         TED Togge "(10 byte)         Pased         Pased         Characteristic Configuration Configuration           20         0.4203         GATT_GHARACTER_UND         10 (properties: notify only) DEF (10 top barRel)         TD, SETLESPEED         Read         Characteristic Configuration           21         0.47T2         TD_SETLESPEED_LUD         0.119/s)         TD_SETLESPEED         Read         Mathy         SELESPEED         Read         Characteristic Value Characteristic Value Characteristic Value         Read         Mathy         SELESPEED         Read         Characteristic Value Characteristic Value Characteristic Value Characteristic Value Characteristic Value Characteristic Value         Read         Mathy         SELESPEED         Read         Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic	18	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)			Read	
10         0x201         CATT_OHAR_USB_DESC_UUD         TED Toggle * (10 bytes)         Amount of the state of the								
20         0.203         GAT_GHARACTER_UUI         T0 (properties: notify only) DE FF_UUD 0. FFL02         DD_SETLEDSPEED_UD0         Read         Generation Declaration           21         0.4702         D_SETLEDSPEED_UD0         0 (1 byle)         D_SETLEDSPEED_UD0         Read         Minito Declaration         Read         Generations Declaration           22         0.4200         GAT_GHARACTER_UU0         TED Anmaton * (13 bytes)         Read         Read         Generations Declaration           24         0.2201         GAT_GHARACTER_UU0         TED Anmaton * (13 bytes)         Read         Read         Generations Declaration           25         0.4703         TD_GETBUTTONICOUNT_UD0         0 (1 byle)         TD_GETBUTTONICOUNT_UD0         Classical field         Contractoristic Declaration         Generations Declaration           26         0.4202         GATT_GHARACTER_UU0         Classical field         TD_GETBUTTONICOUNT_UD0         0 (1 byle)         TD_GETBUTTONICOUNT_Read         Read         Contractoristic Declaration           27         0.4201         GATT_GHAR_USER_DESC_UD0         Generation field         TD (properties: notify only) tf0 (properties	19	0x2901	GATT_CHAR_USER_DESC_UUID	"LED Toggle " (10 bytes)			Read	
20         0x2000         0ATT_CHARACTEP_LUD         1F 00 (handle: 0x00 FP)         Peed         Duracersetic Duracersetic Declaration           21         0x2000         TD_SETLEDSPEED_LUD         0 (1 byle)         TD_SETLEDSPEED         Read         SetLEDS Speed           22         0x2000         CATT_CLENT_CHAR_CF0_LUD         10 (0 popertise: notify only) 10 (popertise: notify only) 11 (popertise: notify only) 12 (popertise: notify only) 13 (popertise: notify only) 13 (popertise: notify only) 14 (popertise: notify only) 15 (popertise: notify o								
Image: Constraint of the	20	0,2002					Pood	
11         0xFFD2         TD_SEFLEDSPEED_UU0         0(1 byte)         TD_SEFLEDSPEED         Read         Watery         SefLEDSpeed Oracteristic Value           22         0x2902         GATT_CLENT_CHAR_GFG_UUD         00:00 (2 bytes)         Image: Comparison of the comparis	20	072003					Neau	
22         0.2020         GATT_CLENT_OHAR_CFG_UUD         0000 (2 bytes)         1         0         1         0         0         0         0000 (2 bytes)           23         0:2001         GATT_CHAR_USE_DESC_UUD         TLED Animation * (13 bytes)         Read         0         Control tents	21					Pood / Write	Notify	
22         Ox2-000         CATL_CHAP_LEND_LEND_CHUD         Observations         Paralest         Observations           23         0x2-001         GATL_CHAP_LUSER_DESC_UUU         TED Animation '(13 bytes)         Read         Read         Cate Station 1 Press Configuration           24         0x2-003         GATL_CHAP_RCTER_UUU         TO (properties: notify only) IF (UUD.OKFPD)         TD_GETBUTTON1COUNT_UUD         Read         Cate Station 1 Press Count Characteristic Ualue         Cate Station 1 Press Count Characteristic Value         Read         Cate Station 1 Press Count Characteristic Value         Count Characteristic	21	UXITUZ		o (T byte)		Read / White	Notify	Characteristic Value
23         0x2901         GATT_OHAR_USER_DESC_UUD         LED Annation *(13 bytes)         Pead         Set LED Speed Obracetristic Configuration           24         0x2803         GATT_OHARACTER_UUD         10 (properties: notify only) 1F 00 (pandie: 0x001F) 05 (UD 0x1FD3)         D_GETBUTTONICOUNT         Read         Set LED Speed Declaration           25         0x5FD3         TD_GETBUTTONICOUNT_UUD         0 (1 byte)         TD_GETBUTTONICOUNT         Read         Notify         Get Button 1 Press Count Characteristic Value           26         0x2901         GATT_OLAR_CFG_UUD         0000 (2 bytes)         TD_GETBUTTONICOUNT         Read         Get Button 1 Press Count Characteristic Value           27         0x2901         GATT_OLAR_USER_DESC_UUD         "Get Button 1 Press Count" (24 bytes)         Read         Get Button 1 Press Count Characteristic Configuration           28         0x2902         GATT_OLAR_USER_DESC_UUD         "Get Button 1 Press Count" (24 bytes)         D_GETBUT TONICOUNT         Read         Get Button 1 Press Count Characteristic Configuration           29         0x5FD4         TD_GERBUTTONICOUNT_UUD         0 (1 byte)         TD_GLRBUTTONICOUNT_Read         Read         Count Characteristic Count Characteristi	22	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)			Read	
23         0x2001         GATE_OARRUSER_DESC_UUD         LDD Animation*(13 bytes)         Read         Omatomic Configuration Configuration           24         0x2003         GATE_OARRACTER_UUD         If (0roperties: notify only) (15 (roperties): notify only)         To_GETBUTTONICOUNT_UUD         (10 (roperties): notify only)         Read         Configuration Configuration           25         0x4P03         To_GETBUTTONICOUNT_UUD         0(1 (byte)         To_GETBUTTONICOUNT_PUD         (1 (byte)         To_GETBUTTONICOUNT_PUD         (2 (byte)         (								
24         0x2803         GATT_CHARACTER_UUD         10 (properties: notify only) (pO handle: 0x001F) DS FF (UUD: 0xFFD3)         The analysis of t	23	0x2901	GATT_CHAR_USER_DESC_UUID	"LED Animation " (13 bytes)			Read	Characteristic
24     0x2803     CATT_CHARACTER_UUU     1F 00 (handle cho0)Fp.     number of the product of the produc				10 (				
Image: Control of the second	24	0x2802					Read	
25         0xFED3         TD_GETBUTTONICOUNT_LUDD         0 (1 by/e)         TD_GETBUTTONICOUNT         Read / Write         Nutry         Count Characteristic Value           26         0x2902         GATT_GLART_CHAR_CFG_UUD         0:00 (2 byres)         Read / Write         Nutry         Count Characteristic Value         Get Button 1 Press Count Characteristic           27         0x2901         GATT_GHAR_USER_DESC_UUD         TGEt Button 1 Press Count Characteristic         Read         Get Button 1 Press Count Characteristic           28         0x2803         GATT_GHAR_USER_DESC_UUD         TGC Button 1 Press Count Characteristic         Read         Count Characteristic Count Characteristic           29         0xFED4         TD_GLRBUTTONICOUNT_LUDD         0 (1 by/e)         TD_GLRBUTTONICOUNT         Read / Write         Nutry         Count Characteristic           30         0x2902         GATT_GHAR_USER_DESC_UUD         0:00 (2 byres)         TD_GLRBUTTONICOUNT         Read / Write         Nutry         Count Characteristic           31         0:2901         GATT_GHAR_USER_DESC_UUD         0:00 (2 byres)         TD_GETBUTTONICOUNT         Read / Write         Nutry         Count Characteristic           32         0:2901         GATT_GHAR_USER_DESC_UUD         0:00 (2 byres)         TD_GETBUTTONICOUNT         Read / Write         Nutry	24	072003					Neau	
Image: Constraint of the second sec				-/				
26         0x2902         GATT_CLENT_CHAR_CFG_UUD         00:00 (2 bytes)         Read         Get Button 1 Press Court Characteristic Value           27         0x2901         GATT_CHAR_USER_DESC_UUD         "Get Button 1 Press Count" (24 bytes)         Read         Read         Court Characteristic Configuration           28         0x2803         GATT_CHARACTER_UUD         10 (properties : notify only) 1F 00 (nande: 0x001F)         Read         Clear Button 1 Press Characteristic           29         0xFFD4         TD_CLRBUTTONICOUNT_UUD         0 (1 byte)         TD_CLRBUTTONICOUNT         Read / Write         Notify Value           30         0x2902         GATT_CHAR_USER_DESC_UUD         0 (1 byte)         TD_CLRBUTTONICOUNT         Read / Write         Notify Value         Clear Button 1 Press Court Characteristic Value           31         0x2901         GATT_CHAR_USER_DESC_UUD         "Clear Button 1 Press Court Characteristic Value         Read         Clear Button 1 Press Court Characteristic Value           32         0x2903         GATT_CHAR_USER_DESC_UUD         "Clear Button 1 Press Court Characteristic Value         Clear Button 1 Press Court Characteristic Value         Clear Button 1 Press Court Characteristic Value           33         0xFFD5         TD_GETBUTTON2COUNT_UUD         0 (1 bytes)         TD_GETBUTTON2COUNT         Read         Clear Button 2 Press Court Characteristic Value	25	0xFFD3	TD_GETBUTTON1COUNT_UUID	0 (1 byte)	TD_GETBUTTON1COUNT	Read / Write	Notify	Count Characteristic
26     0x2902     GATT_CLENT_CHAR_CFG_UUD     00:00 (2 bytes)     Read     Count Characteristic Count Characteristic Out C								
Image: Constraint of the second state second state second state state of the second state state state o	26	0x2902	GATT CLIENT CHAR CFG UUID	00:00 (2 bytes)			Read	
27     0x2901     GATT_CHAR_USER_DESC_UUD     Get Button 1 Press Count * (24 bytes)     Read     Count Characteristic Charac								
27       0:2201       GATT_CHAR_USER_DESC_UDD       bytes)       Image: control processing to control procesing to control procesing to control procontrol processing to contro				"Get Button 1 Press Count " (24				
28         0x2803         GATT_CHARACTER_UUD         10 (properties: notify only) 1F 00 (handle: 0x001F) D4 FF (UUD: 0xFFD4)         TD         Read         Coar Button 1 Press Councersitie Declaration           29         0xFFD4         TD_CLRBUTTONICOUNT_UUD         0 (1 byte)         TD_CLRBUTTONICOUNT         Read / Write         Notify         Clear Button 1 Press Counc Characteristic Value           30         0x2902         GATT_CLENT_OHAR_CFG_UUD         00:00 (2 bytes)         TD_CLRBUTTONICOUNT         Read         Count Characteristic Value         Count Characteristic Value           31         0x2901         GATT_CHAR_USER_DESC_UUD         "Clear Button 1 Press Count " (28 bytes)         Read         Count Characteristic Configuration         Count Characteristic Configuration           32         0x2803         GATT_CHAR_USER_DESC_UUD         "Clear Button 1 Press Count Characteristic Configuration         Read         Cell Button 2 Press Count Characteristic Configuration           33         0xFFD5         TD_GETBUTTON2COUNT_UUD         0 (1 byte)         TD_GETBUTTON2COUNT         Read / Write         Notify         Cell Button 2 Press Count Characteristic Value           34         0x2902         GATT_CHAR_USER_DESC_UUD         "Get Button 2 Press Count "(24 bytes)         Read         Get Button 2 Press Count Characteristic Count Characteristic Count Characteristic Count Characteristic Count Characteristic Count Characteristic Count Char	27	0x2901	GATT_CHAR_USER_DESC_UUID				Read	
28       0x2803       GATT_CHARACTER_UUD       1F 00 (handle: 0x01F)       Pead       Pead       Ourraceristic Declaration         29       0xFFU4       TD_CLRBUTTONICOUNT_UUD       0 (1 byte)       TD_CLRBUTTONICOUNT       Read / Write       Notify       Ourraceteristic Value         30       0x2902       GATT_CHAR_CFG_UUD       00:00 (2 bytes)       TD_CLRBUTTONICOUNT       Read / Write       Notify       Ourraceteristic Value         31       0x2901       GATT_CHAR_USEP_DESC_UUD       "Clear Button 1 Press Count"       Read       Clear Button 1 Press Count (28 bytes)       Read       Court Characteristic Count Characteristic Court Characteristic Court Characteristic Count Characteristic Count Characteristic Count Characteristic Count Characteristic Count Characteristic Declaration       Get Button 2 Press Count * (28 bytes)       Read       Ourraceteristic Declaration         32       0x2802       GATT_CHAR_USEP_DESC_UUD       0 (1 byte)       TD_GETBUTTON2COUNT       Read       Notify       Count Characteristic Value         33       0xFFD5       TD_GETBUTTON2COUNT_UUD       0 (1 byte)       TD_GETBUTTON2COUNT       Read       Count Characteristic Value       Count Characteristic Value       Count Characteristic Count Characteristic Count Characteristic Value       Count Charact				10 (properties: notify only)				
29         0xFFD4         TD_CLRBUTTONICOUNT_LUUD         0 (1 byte)         TD_CLRBUTTONICOUNT         Read / Write         Notify         Gear Button 1 Press Count Characteristic Value           30         0x2902         GATT_CLENT_CHAR_CFG_UUD         0:00 (2 bytes)         Read         Cear Button 1 Press Count Characteristic Value           31         0x2901         GATT_CHAR_USER_DESC_UUD         "Gear Button 1 Press Count" (28 bytes)         Read         Cear Button 1 Press Count Characteristic Configuration           32         0x2903         GATT_CHARACTER_UUD         "Gear Button 1 Press Count" (28 bytes)         Read         Gear Button 2 Press Count Characteristic Configuration           32         0x2903         GATT_CHARACTER_UUD         0 (1 byte)         TD_GETBUTTON2COUNT (28 bytes)         Read         Get Button 2 Press Count Characteristic Declaration           33         0xFFD5         TD_GETBUTTON2COUNT_UUD         0 (1 byte)         TD_GETBUTTON2COUNT         Read         Notify         Get Button 2 Press Count Characteristic Value           34         0x2902         GATT_CHAR_USER_DESC_UUD         0:00 (2 bytes)         TD_GETBUTTON2COUNT         Read         Count Characteristic Value           35         0x2903         GATT_CHAR_USER_DESC_UUD         "Get Button 2 Press Count * (24 bytes)         Read         Count Characteristic Configuration         Cear Button 2 Pres	28	0x2803	GATT_CHARACTER_UUID				Read	
29       0xFFD4       TD_CLRBUTTONICOUNT_UUD       0 (1 byte)       TD_CLRBUTTONICOUNT       Read / Write       Notify       Count Characteristic Value         30       0x2902       GATT_CLENT_CHAR_CFG_UUD       0:00 (2 bytes)       Image: Count Characteristic Value       Count Characteristic Value       Count Characteristic Value         31       0x2901       GATT_CHAR_USER_DESC_UUD       "Clear Button 1 Press Count" (28 bytes)       Read       Cerr Button 1 Press Count (28 bytes)       Count Characteristic Configuration         32       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) if 50 (nharde: 0x001F), D5 FF (UUD: 0xFFD5)       TD_GETBUTTON2COUNT       Read       Get Button 2 Press Count (24 bytes)         33       0xFFD5       TD_GETBUTTON2COUNT_UUD       0 (1 byte)       TD_GETBUTTON2COUNT       Read       Get Button 2 Press Count (24 bytes)         34       0x2902       GATT_CLENT_CHAR_CFG_UUD       0(1 byte)       TD_GETBUTTON2COUNT       Read       Get Button 2 Press Count (24 bytes)       <				D4 FF (UUID: 0xFFD4)				
Image: Construction of the construc	20			0(1  byto)		Pood / M/rito	Notify	
30         0x2902         GATT_CLENT_CHAR_CFG_UUD         00:00 (2 bytes)         Read         Clear Button 1 Press Count Characteristic Value           31         0x2901         GATT_CHAR_USER_DESC_UUD         "Clear Button 1 Press Count" (28 bytes)         Read         Clear Button 1 Press Count Characteristic Ontiguration           32         0x2803         GATT_CHARACTER_UUD         "D (properties: notify only) 1F 00 (handle: 0x001F) D5 FF (UUD: 0xFFD5)         Read         Clear Button 2 Press Count Characteristic Ontroperties: Declaration           33         0xFFD5         TD_GETBUTTON2COUNT_UUD         0 (1 byte)         TD_GETBUTTON2COUNT D (bytes)         Read         Get Button 2 Press Count Characteristic Value           34         0x2901         GATT_CLENT_CHAR_CFG_UUD         0 (1 byte)         TD_GETBUTTON2COUNT Person 2 (24 bytes)         Read         Get Button 2 Press Count Characteristic Value           35         0x2901         GATT_CLENT_CHAR_CFG_UUD         0:00 (2 bytes)         Read         Get Button 2 Press Count Characteristic Value         Get Butt	29	UXFFD4		u (T byte)		Reau / White	NOULY	
Image: Construction of the construc								Clear Button 1 Press
31       0x2901       GATT_CHAR_USER_DESC_UUID       "Clear Button 1 Press Count " (28 bytes)       Read       Clear Button 1 Press Count Characteristic Configuration         32       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D5 FF (UUD: 0xFFD5)       Read       Cert Button 2 Press Characeristic Declaration         33       0xFFD5       TD_GETBUTTON2COUNT_UUID       0 (1 byte)       TD_GETBUTTON2COUNT       Read       Cert Button 2 Press Count Characteristic Value         34       0x2902       GATT_CLENT_CHAR_CFG_UUD       00:00 (2 bytes)       TD_GETBUTTON2COUNT       Read       Count Characteristic Value         35       0x2901       GATT_CHAR_USER_DESC_UUID       "Get Button 2 Press Count " (24 bytes)       Read       Count Characteristic Value         36       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       TD_CLRBUTTON2COUNT       Read       Count Characteristic Value         37       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Cear Button 2 Press Count Characteristic Value         38       0x2902       GATT_CLENT_CHAR_CFG_UUD       00:00 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Cear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHARACTER_UUID	30	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)			Read	Count Characteristic
31       0x2901       GATT_CHAR_USER_DESC_UUD       "Cear Button 1 Press Count" (28 bytes)       Read       Count Characteristic Configuration         32       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) 1F 00 (handle: 0x001F) D5 FF (UUD 0xFFD6)       Read       Read       Cet Button 2 Press Count Characteristic Declaration         33       0xFFD5       TD_GETBUTTON2COUNT_UUD       0 (1 byte)       TD_GETBUTTON2COUNT       Read       Write       Notify       Count Characteristic Declaration         34       0x2902       GATT_CLENT_CHAR_OFG_UUD       00:00 (2 bytes)       TD_GETBUTTON2COUNT       Read       Count Characteristic Value         35       0x2901       GATT_CHAR_USER_DESC_UUD       "Get Button 2 Press Count" (24 bytes)       Read       Read       Count Characteristic Value         36       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       Read       Read       Count Characteristic Configuration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Count Characteristic Count Characteristic Count Characteristic         38       0x2902       GATT_CHAR_CFG_UUD       00:00 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Cear Button 2 Press Count Characteristic       Count Characteristic Count Characteristic<								
12       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) 10 (properties: notify only) DS FF (UUD: 0xFFD5)       Read       Cet Button 2 Press Declaration         33       0xFFD5       TD_GETBUTTON2COUNT_UUD       0 (1 byte)       TD_GETBUTTON2COUNT       Read / Write       Notify       Count Characteristic Value         34       0x2902       GATT_CLENT_CHAR_CFG_UUD       00:00 (2 bytes)       TD_GETBUTTON2COUNT       Read / Write       Notify       Count Characteristic Value         35       0x2901       GATT_CHAR_USER_DESC_UUD       "Get Button 2 Press Count" (24 bytes)       Read       Cet Button 2 Press Count Characteristic Configuration         36       0x2803       GATT_CHARACTER_UUD       "Get Button 2 Press Count" (24 bytes)       Read       Cet Button 2 Press Count Characteristic Configuration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Cear Button 2 Press Count Characteristic Configuration         38       0x2902       GATT_CHARACTER_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Cear Button 2 Press Count Characteristic Configuration         39       0x2902       GATT_CLENT_CHAR_CFG_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Cear Button 2 Press Count Characteristic Value       Cear Button 2	31	0x2901	GATT CHAR USER DESC UUID				Read	Count Characteristic
32       0x2803       GATT_CHARACTER_UUID       1F 00 (handle: 0x001F) D5 FF (UUD: 0xFFD5)       Read       Characeristic Declaration         33       0xFFD5       TD_GETBUTTON2COUNT_UUID       0 (1 byte)       TD_GETBUTTON2COUNT       Read / Write       Notify       Get Button 2 Press Count Characteristic Value         34       0x2902       GATT_CLENT_CHAR_CFG_UUD       00:00 (2 bytes)       Image: Count (24 bytes)       Read / Write       Read       Get Button 2 Press Count Characteristic Value         35       0x2901       GATT_CHAR_USER_DESC_UUD       "Get Button 2 Press Count " (24 bytes)       Read       Read       Get Button 2 Press Count Characteristic Configuration         36       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       TD_CLRBUTTON2COUNT       Read       Gear Button 2 Press Count Characteristic Configuration         37       0xFFD5       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Gear Button 2 Press Count Characteristic Value       Gear Button 2 Press Count Characteristic Configuration       Gear Button 2 Press Count Characteristic Value       <				(28 bytes)				
Image: state stat								
33       0xFFD5       TD_GETBUTTON2COUNT_UUID       0 (1 byte)       TD_GETBUTTON2COUNT       Read / Write       Notify       Get Button 2 Press Count Characteristic Value         34       0x2902       GATT_CLIENT_CHAR_CFG_UUID       00:00 (2 bytes)       Read       Get Button 2 Press Count Characteristic Value         35       0x2901       GATT_CHAR_USER_DESC_UUID       "Get Button 2 Press Count " (24 bytes)       Read       Read       Get Button 2 Press Count Characteristic Value         36       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUID: 0xFFD6)       Read       Read       Characteristic Configuration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Clear Button 2 Press Count Characteristic Value         38       0x2902       GATT_CLENT_CHAR_CFG_UUID       0:00 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUID       "Clear Button 2 Press Count " (28 bytes)       TD_GETTEMP       Read       Clear Button 2 Press Count Characteristic Configuration         40       0x2803       GATT_CHAR_LISER_DESC_UUID       10 (properties: notify only) 1F 00 (pnacteries: notify only)       TD_GETTEMP       Read       Clear Button 2 Press C	32	0x2803	GATT_CHARACTER_UUID				Read	
33       0xFFD5       TD_GETBUTTON2COUNT_UUID       0 (1 byte)       TD_GETBUTTON2COUNT       Read / Write       Notify       Count Characteristic Value         34       0x2902       GATT_CLIENT_CHAR_CFG_UUD       0::00 (2 bytes)       Image: Count Characteristic Value       Get Button 2 Press         35       0x2901       GATT_CHAR_USER_DESC_UUD       "Get Button 2 Press Count " (24 bytes)       Read       Read       Get Button 2 Press         36       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) IF 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       Read       Read       Clear Button 2 Press Count * (24 cont Characteristic Configuration 2 Press Cont Characteristic Configuration 2 Press Cont Characteristic Configuration 2 Press         37       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) IF 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       Read       Read       Clear Button 2 Press Count * Value         38       0x2902       GATT_CLENT_CHAR_CFG_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count * Value         39       0x2901       GATT_CLENT_CHAR_CFG_UUD       0::00 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count * Value         39       0x2901       GATT_CLENT_CHAR_CFG_UUD       0::00 (2 bytes)       Image: Clear Button 2 Press Count * (28 bytes)       Clear Button 2 Press Count *								
34       0x2902       GATT_CLIENT_CHAR_CFG_UUID       00:00 (2 bytes)       Read       Get Button 2 Press Count Characteristic Value         35       0x2901       GATT_CHAR_USER_DESC_UUID       "Get Button 2 Press Count " (24 bytes)       Read       Read       Get Button 2 Press Count Characteristic Configuration         36       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUID: 0xFFD6)       Read       Clear Button 2 Press Count Characteristic Declaration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Notify       Clear Button 2 Press Count Characteristic Declaration         38       0x2902       GATT_CLIENT_CHAR_CFG_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Odear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUID       "Clear Button 2 Press Count " (28 bytes)       TD_GETTEMP       Read       Clear Button 2 Press Count Characteristic Value         40       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)       TD_GETTEMP       Read       Get Temperature da Characceristic Declaration	33	0xFFD5	TD_GETBUTTON2COUNT_UUID	0 (1 byte)	TD_GETBUTTON2COUNT	Read / Write	Notify	Count Characteristic
34       0x2902       GATT_CLIENT_CHAR_CFG_UUD       00:00 (2 bytes)       Read       Count Characteristic Value         35       0x2901       GATT_CHAR_USER_DESC_UUD       "Get Button 2 Press Count " (24 bytes)       Read       Get Button 2 Press Count Characteristic Configuration         36       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       Read       Cear Button 2 Press Count Characteristic Declaration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Cear Button 2 Press Count Characteristic Declaration         38       0x2902       GATT_CLIENT_CHAR_CFG_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Cear Button 2 Press Count Characteristic Value         39       0x2902       GATT_CLIENT_CHAR_CFG_UUD       0:0:0 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Cear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUD       "Clear Button 2 Press Count " (28 bytes)       Read       Cear Button 2 Press Count Characteristic Configuration         40       0x2803       GATT_CHAR_USER_DESC_UUD       "Clear Button 2 Press Count " (28 bytes)       Cear Button 2 Press Count Characteristic Configuration       Count Characteristic Count Characteristic Configuration         41       0x2803       GAT								
Image: state in the state	34	0x2902	GATT CLIENT CHAR CEG LILIID	00:00 (2 bytes)			Read	
35       0x2901       GATT_CHAR_USER_DESC_UUID       "Get Button 2 Press Count " (24 bytes)       Read       Get Button 2 Press Count Characteristic Configuration         36       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUID: 0xFFD6)       Read       Characeristic Declaration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count Characteristic Declaration         38       0x2902       GATT_CLIENT_CHAR_CFG_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CLIENT_CHAR_CFG_UUID       00:00 (2 bytes)       Read       Clear Button 2 Press Count Characteristic Value         40       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)       Read       Get Temperature da Characteristic Configuration         41       0xFED7       TD_GETTEMP_UUID       00:00 (2 byte)       TD_GETTEMP_E       Read (Write       Notify	04	012002						
33       0X2901       GATT_CHAR_USER_DESC_UUD       bytes)       Count Characteristic Configuration         36       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       Read       Clear Button 2 Press Characteristic Declaration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUD       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Count Characteristic Declaration         38       0x2902       GATT_CLIENT_CHAR_CFG_UUD       00:00 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUD       "Clear Button 2 Press Count " (28 bytes)       Read       Clear Button 2 Press Count Characteristic Value         40       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUD: 0xFFD7)       TD_GETTEMP       Read       Clear Button 2 Press Count Characteristic Value         41       0xFED7       TD_GETTEMP_UUD       00:00 (2 byte)       TD_GETTEMP       Read       Clear Button 2 Press Count Characteristic Value				"Get Button 2 Press Count " (24				Get Button 2 Press
36       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D6 FF (UUID: 0xFFD6)       Read       Clear Button 2 Press Characeristic Declaration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Clear Button 2 Press Count Characeristic Value         38       0x2902       GATT_CLIENT_CHAR_CFG_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUID       "Clear Button 2 Press Count " (28 bytes)       Clear Button 2 Press Count Characteristic Value       Clear Button 2 Press Count Characteristic Value         40       0x2803       GATT_CHAR_USER_DESC_UUID       "Clear Button 2 Press Count " (28 bytes)       Read       Clear Button 2 Press Count Characteristic Configuration         40       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)       TD_GETTEMP       Read       Get Temperature da Characeristic Declaration         41       0xFED7       TD_GETTEMP_UUID       00:00 (2 byte)       TD_GETTEMP       Read //Write       Notify	35	0x2901	GATT_CHAR_USER_DESC_UUID				Read	Count Characteristic
36       0x2803       GATT_CHARACTER_UUID       1F 00 (handle: 0x001F) D6 FF (UUD: 0xFFD6)       Read       Characeristic Declaration         37       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Clear Button 2 Press Count Characteristic Value         38       0x2902       GATT_CLIENT_CHAR_CFG_UUID       00:00 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUID       "Clear Button 2 Press Count " (28 bytes)       Clear Button 2 Press Count Characteristic Value       Clear Button 2 Press Count Characteristic Configuration       Clear Button 2 Press Count Characteristic Configuration       Clear Button 2 Press Count Characteristic Value       Clear Button 2 Press Count Characteristic Configuration       Clear Button 2 Press Count Characteristic Configurat				10 (properties: notify only)				
1       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Clear Button 2 Press Count Characteristic Value         38       0x2902       GATT_CLIENT_CHAR_CFG_UUID       00:00 (2 bytes)       TD_CLRBUTTON2COUNT       Read       Clear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CLIENT_CHAR_CFG_UUID       00:00 (2 bytes)       Image: Clear Button 2 Press Count Characteristic Value       Clear Button 2 Press Count Characteristic Value       Clear Button 2 Press Count Characteristic Value         40       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)       TD_GETTEMP       Read       Get Temperature da Characeristic Declaration         41       0xFED7       TD_GETTEMP_UUID       00:00 (2 byte)       TD_GETTEMP       Read (Write       Notify	36	0x2803	GATT_CHARACTER_UUID				Read	
37       0xFFD6       TD_CLRBUTTON2COUNT_UUID       0 (1 byte)       TD_CLRBUTTON2COUNT       Read / Write       Notify       Count Characteristic Value         38       0x2902       GATT_CLIENT_CHAR_CFG_UUD       00:00 (2 bytes)       Image: Count Characteristic Value       Cear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUD       "Clear Button 2 Press Count"       Read       Cear Button 2 Press Count Characteristic Value         40       0x2803       GATT_CHAR_ACTER_UUD       "Clear Button 2 Press Count"       Read       Read       Cear Button 2 Press Count Characteristic Configuration 2 Press Count (28 bytes)         40       0x2803       GATT_CHARACTER_UUD       "Clear Button 2 Press Count"       Read       Read       Cear Button 2 Press Count Characteristic Configuration 2 Press Count (28 bytes)         41       0x2803       GATT_CHARACTER_UUD       "In (properties: notify only)"       For (handle: 0x001F)       Read       Get Temperature da Characeristic Configuration         41       0xFED7       TD_GETTEMP_UUD       00:00 (2 byte)       TD_GETTEMP_R       Read / Write       Notify       Get Temperature da			_	D6 FF (UUID: 0xFFD6)				
Image: state in the state	07	0		0 (1 hute)		Deed (14/2)	N Indife .	Clear Button 2 Press
38       0x2902       GATT_CLIENT_CHAR_CFG_UUD       00:00 (2 bytes)       Read       Clear Button 2 Press Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUD       "Clear Button 2 Press Count " (28 bytes)       Read       Clear Button 2 Press Count Characteristic Configuration         40       0x2803       GATT_CHARACTER_UUD       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUD: 0xFFD7)       Read       Get Temperature da Characeristic Declaration         41       0xFED7       TD_GETTEMP_UUD       00:00 (2 byte)       TD_GETTEMP_UE       Read / Write       Notify	37	UXFFD6	ID_CLRBUTTON2COUNT_UUID	u (1 byte)	ID_CLKBUTION2COUNT	rkead / Write	NOTITY	
38       0x2902       GATT_CLIENT_CHAR_CFG_UUD       00:00 (2 bytes)       Read       Count Characteristic Value         39       0x2901       GATT_CHAR_USER_DESC_UUD       "Clear Button 2 Press Count"       Read       Clear Button 2 Press Count " (28 bytes)         40       0x2803       GATT_CHARACTER_UUD       "I0 (properties: notify only) IF 00 (handle: 0x001F) D7 FF (UUD: 0xFFD7)       Read       Read       Cert Temperature da Characteristic Configuration         41       0xFED7       TD_GETTEMP_UUD       00:00 (2 byte)       TD_GETTEMP_N       Read (Write Notify       Get Temperature da								Clear Button 2 Press
39       0x2901       GATT_CHAR_USER_DESC_UUID       "Clear Button 2 Press Count " (28 bytes)       Read       Clear Button 2 Press Count Characteristic Configuration         40       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)       Read       Clear Button 2 Press Count Characteristic Configuration         41       0xFED7       TD_GETTEMP_UUID       00:00 (2 byte)       TD_GETTEMP_N       Read / Write       Notify	38	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)			Read	Count Characteristic
39       0x2901       GATT_CHAR_USER_DESC_UUD       Clear Button 2 Press Count (28 bytes)       Read       Count Characteristic Configuration         40       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)       Read       Get Temperature da Characeristic Declaration         41       0xFED7       TD_GETTEMP_UUID       00:00 (2 byte)       TD_GETTEMP_R       Read / Write       Notify								
40       0x2803       GATT_CHARACTER_UUID       10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)       10 (properties: notify only) Read       Read       Get Temperature da Characeristic Declaration         41       0xFED7       TD_GETTEMP_UUID       00:00 (2 byte)       TD_GETTEMP_N       Read / Write       Notify	30	0x2001		"Clear Button 2 Press Count "			Pood	
40     0x2803     GATT_CHARACTER_UUID     10 (properties: notify only) 1F 00 (handle: 0x001F) D7 FF (UUID: 0xFFD7)     Read     Get Temperature da Characeristic Declaration       41     0xFED7     TD_GETTEMP_UUID     00:00 (2 byte)     TD_GETTEMP_R     Read / Write     Notify	39	072901	GATI_UNAR_USER_DESU_UUID	(28 bytes)			Reau	
D7 FF (UUID: 0xFFD7) D2 GETTEMP UUID 00:00 (2 byte) TD GETTEMP Read / Write Notify Get Temperature da				10 (properties: notify only)				Get Temperature dat
41 0xFED7 TD GETTEMP UUD 00:00 (2 byte) TD GETTEMP Read / Write Notify Get Temperature da	40	0x2803	GATT_CHARACTER_UUID	1F 00 (handle: 0x001F)			Read	
41 (XEE)/ I) (FELEMP UUL) IO(0) (2 byte) III) (FELEMP IRead / Write INotity I				D7 FF (UUID: 0xFFD7)				Declaration
			TD GETTEMP LILIID	00.00 (2 byte)	TD GETTEMP	Read / Write	Notify	Get Temperature data

### Figure 5. TM4C BLE Peripheral Profile Table



### 3 Getting Started Hardware

For both the master central and slave peripheral nodes, the hardware is the EK-TM4C129XL-connected LaunchPad and the CC2650 EMK board. The EK-TM4C129XL-connected LaunchPad board is connected to the SimpleLink<sup>™</sup> CC2650 BoosterPack<sup>™</sup> board through the BoosterPack connector 1 and an adapter board. The communication channel is UART in 2-pin standard mode. Table 1 lists the necessary signal mapping for the demonstration.

BoosterPack Connector	TM4C1294 LaunchPad	CC2650EMK	EM Adapter BoosterPack
A1-1	3.3 V	3.3 V	VDD_LP
A1-2	PE4	Unused	PE4
A1-3	PC4_U7RX	IOID_2	LP1-3
A1-4	PC5_U7TX	IOID_3	LP1-4
A1-5	PC6	Unused	Unused
A1-6	PE5	Unused	Unused
A1-7	PD3_SSI2CLK	Unused	Unused
A1-8	PC7	Unused	Unused
A1-9	PB2	Unused	Unused
A1-10	PB3	Unused	Unused
D1-1	GND	GND	GND
D1-2	PM3	Unused	Unused
D1-3	PH2	Unused	Unused
D1-4	PH3	Unused	Unused
D1-5	RESET	Unused	Unused
D1-6	PD1_I2C7SDA	Unused	Unused
D1-7	PD0_I2C7SCL	Unused	Unused
D1-8	PN2	Unused	Unused
D1-9	ON3	Unused	Unused
D1-10	PP2	Unused	Unused

### Table 1. Signal Mapping



Getting Started Software

#### 4 Getting Started Software

#### 4.1 TM4C BLE Node as a Central Software Architecture

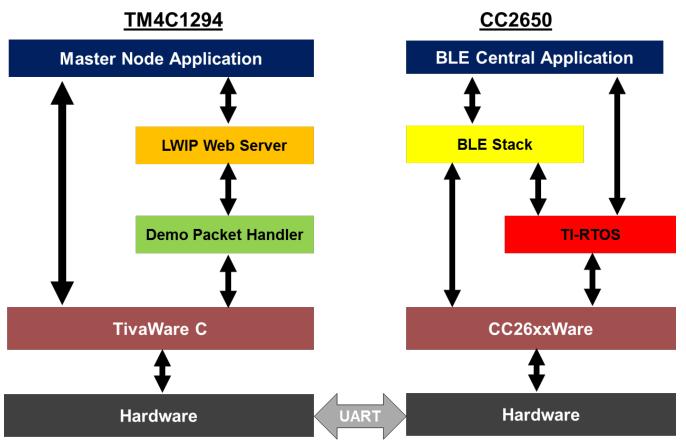
Figure 6 illustrates the architecture of the TM4C1294 BLE central node.

TM4C software blocks:

- TivaWare<sup>™</sup> C Allows for TM4C hardware register access and UART communication.
- LWIP Ethernet stack for Web server
- Demo packet handler
  - Converts http requests to UART demo commands.
  - Converts UART command responses to string format for Web page display.

CC2650 software blocks:

- TI RTOS for general scheduling
  - (a) Manages the demo command or response handling over UART.
  - (b) Operates the BLE central.
- CC26xxWare Performs CC2650 hardware access and UART operation.
- BLE Stack Supports BLE protocol.



#### Figure 6. TM4C BLE Central Architecture Block Diagram

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

- cmdReceived\_sem: Waits until the demo command is received.
- cmdResponse\_sem: Waits until the demo response is ready.
- Task\_ externalMCUComm: Performs UART command receive and response transmit.



### 4.2 TM4C BLE Node as a Peripheral Software Architecture

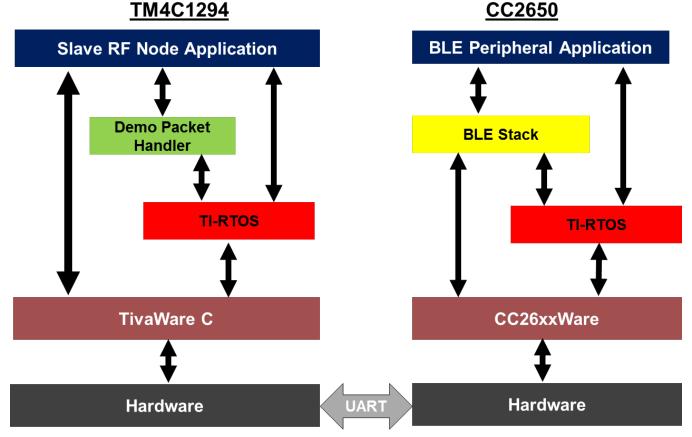
Figure 7 illustrates the architecture of the TM4C1294 BLE peripheral node.

CC2650 software blocks:

- TI RTOS for general scheduling
  (a) Manages the demo command or response handling over UART.
  (b) Operates the BLE peripheral.
- CC26xxWare Performs CC2650 hardware access and UART operation.
- BLE Stack Supports the BLE protocol.

TM4C software blocks:

- TI RTOS for general scheduling
  - (a) Manages the demo command and response handling over UART.
  - (b) Runs the demo application that performs LED control, temperature, button counts, and so forth
- TivaWare C Performs TM4C hardware register access and UART communication.
- Demo packet handler Decodes UART-based demo commands to demo tasks.



### Figure 7. TM4C BLE Peripheral Architecture Block Diagram

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

- TM4C:
  - ledAnimationClock: LED animation control
  - updateTempClock: Updates the temperature value every second.
  - updateButtonCountclock: Updates the button press status every 10 ms.
  - cmdReceived\_sem: Waits until the demo command is received.



- cmdResponse\_sem: Waits until the demo response is ready.
- Task\_ uartCommand: Performs UART command receive and response transmit.
- Task\_ demoRFnode: Performs demo applications such as LED control, temperature, update button counts, and so forth.
- CC2650:
  - cmdReceived\_sem: Waits until demo command is received.
  - cmdResponse\_sem: Waits until demo response is ready.
  - Task\_ externalMCUComm: Performs UART command receive and response transmit.



#### 5 Software Setup

These tools and software packages are required to build and test access point and station projects:

Software Setup

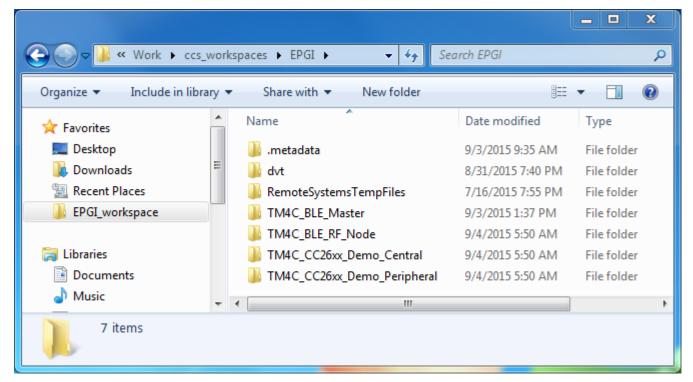
- Composer Studio (<u>http://www.ti.com/tool/ccstudio</u>)
- CC2650 BLE Stack-2 (http://www.ti.com/tool/ble-stack-archive)
- TI-RTOS for CC2650 v2\_11\_01\_0910 (Part of CC2650 BLE Stack-2 Installer). CC26xxWare is included.
- TivaWare\_C v2.1.1.71 (http://www.ti.com/tool/sw-tm4c)
- TI-RTOS for TIVA v2.14.0.10 (Resource Explorer in CCS)

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

The demonstration is not compatible with tirtos\_simplelink version  $2_{12_x}$ ,  $2_{13_x}$ ,  $2_{14_x}$  due to the UART driver changes in these releases. TI recommends using  $2_{11_01_0910}$  for this demonstration.

TI recommends installing these packages in the default location under C:\ti to avoid making any changes in the CCS project. When the previous tools are installed, follow these steps:

- 1. Unzip the software release zip file.
- 2. Place the extracted TM4C\_CC26xx\_Demo\_Central, TM4C\_CC26xx\_Demo\_Peripheral, TM4C\_BLE\_Master, and TM4C\_BLE\_RF\_Node directories in your workspace. See Figure 8.



#### Figure 8. CCS Workspace



3. Import all projects into CCS (see Figure 9).

CCS Edit - TM4C_BLE_Master/TM4C_BLE_Master.c - C	code Composer Studio	■ X
File Edit View Navigate Project Run Scripts	Window Help	
➡ = © < ▼ Ø * ▼ Ø * ■ ©		CS Debug
<ul> <li>Project Explorer X  • • • • • • • • • • • • • • • • • •</li></ul>	<pre>TM4C_BLE_Master.c %  TM4C_BLE_Master.c %  TM4C</pre>	
0 items selected	😥 Full Lice	ense

Figure 9. CCS Projects

4. Check the Linked Resources Path Variables (see Figure 10) to confirm that they correspond to the actual folders in the current setup.

ype filter text	Linked Resources		← → ⇒ →
Resource Linked Resources	Path Variables Linked Resources	]	
Resource Filters CCS General		' the file system, including other path variables with the syntax "S{VAR}". may be specified relative to these path variables.	
CCS Build	Defined path variables for resource	e 'TM4C_BLE_Master':	
<ul> <li>ARM Compiler</li> <li>ARM Linker</li> </ul>	Name	Value	New
<ul> <li>ARM Linker</li> <li>ARM Hex Utility [Disabled]</li> <li>Builders</li> </ul>	CCS_BASE_ROOT	C:\ti\ccsv6\ccs_base C:\ti\ccsv6	Edit
⊳ C/C++ Build	CG_TOOL_ROOT	C:\ti\ccsv6\tools\compiler\ti-cgt-arm_5.2.4	Remove
C/C++ General Debug	ECLIPSE_HOME	C:\ti\ccsv6\eclipse\	
Project References Run/Debug Settings	ORIGINAL_PROJECT_ROOT PARENT_LOC	C:\ti\TivaWare_C_Series-2.1.1.71\examples\boards\ek-tm4c1294xl\TM4C_BLE_Master\ccs D:\Work\ccs_workspaces\EPGI	
	PROJECT_LOC SW ROOT	D:\Work\ccs_workspaces\EPGI\TM4C_BLE_Master \${ORIGINAL_PROJECT_ROOT}\\\\	
	WORKSPACE_LOC	D:\Work\ccs_workspaces\EPGI	
			]

#### Figure 10. Path Variables – TM4C Master



type filter text	Linked Resources		(⇒
Resource     Linked Resources     Resource Filters     CCS General     CCS Build		the file system, including other path variables with the syntax "\${VAR}". nay be specified relative to these path variables. :TM4C_CC26xx_Demo_Central:	
<ul> <li>ARM Compiler</li> <li>ARM Linker</li> <li>ARM Hex Utility [Disabled]</li> <li>XDCtools</li> <li>Builders</li> <li>C/C++ Build</li> <li>C/C++ General</li> <li>Debug</li> <li>Project References</li> <li>Run/Debug Settings</li> </ul>	Name	Value C:\ti\titos_simplelink_2_11_01_09\products\cc26xxware_2_20_06_14829 C:\ti\ccsv6\ccs_base C:\ti\ccsv6 C:\ti\ccsv6\ccols\compiler\ti-cgt-arm_5.2.4 C:\ti\ccsv6\eclipse\ \$(ORIGINAL_PROJECT_ROOT)\\.\JAR\Application\CC2640 C:\ti\simplelink\ble_cc26xx_2_00_00_42893\Projects\ble\TM4C_CC26xx_Demo_Central\CC26xx\CCS\TM4C_CC26xx_Demo_Central D:\Work\ccs_workspaces\EPGI D:\Work\ccs_workspaces\EPGI D:\Work\ccs_workspaces\EPGI	Edit
Hide advanced settings		OK	Cancel

Figure 11. Path Variables – CC2650 Central

pe filter text	Linked Resources		$\langle \neg \neg \neg \rangle$
Resource Linked Resources Resource Filters CCS General CCS Build	The locations of linked resources r	the file system, including other path variables with the syntax "S(VAR)". may be specified relative to these path variables. = "TM4C_CC26xx_Demo_CentralStack':	
ARM Compiler	Name	Value	New
<ul> <li>ARM Linker</li> <li>ARM Hex Utility [Disabled]</li> <li>Builders</li> </ul>	CC26XXWARE	C:\ti\titcs_simplelink_2_11_01_09\products\cc26xxware_2_20_06_14829 C:\ti\ccsx6\ccs_base	Edit
<ul> <li>C/C++ Build</li> <li>C/C++ General</li> </ul>	CCS_INSTALL_ROOT CG_TOOL_ROOT	C:\ti\ccsv6 C:\ti\ccsv6\tools\compile^ti-cgt-arm_5.2.4	Remove
Debug Project References	ECLIPSE_HOME	C:\ti\ccsv6\eclipse\	
Run/Debug Settings	CRG_PROJ_DIR	\${ORIGINAL_PROJECT_ROOT}\\\JAR\Stack\CC2640 C:\ti\simplelink\ble_cc26xx_2_00_00_42893\Projects\ble\TM4C_CC26xx_Demo_Central\CC26xx\CC5\TM4C_CC26xx_Demo_CentralStack	
	PARENT_LOC	D:\Work\ccs_workspaces\EPGI D:\Work\ccs_workspaces\EPGI\TM4C_CC26xx_Demo_CentralStack	
	>WORKSPACE_LOC	D:\Work\ccs_workspaces\EPGI	

Figure 12. Path Variables – CC2650 Central Stack



#### Software Setup

www.ti.com

type filter text	Linked Resources		(> ▼ ⊂>
Resource Linked Resources	Path Variables Linked Resources		
Resource Filters CCS General		the file system, including other path variables with the syntax "\${VAR}". nay be specified relative to these path variables.	
CCS Build	Defined path variables for resource	: 'TM4C_CC26xx_Demo_Peripheral':	
<ul> <li>ARM Compiler</li> <li>ARM Linker</li> </ul>	Name	Value	New
ARM Hex Utility [Disabled]	CC26XXWARE	C:\ti\tirtos_simplelink_2_11_01_09\products\cc26xxware_2_20_06_14829 C:\ti\ccsx6\ccs base	Edit
Builders	CCS INSTALL ROOT	C:\ti\ccsv6	Remove
C/C++ Build	CG_TOOL_ROOT	C:\ti\ccsv6\tools\compiler\ti-cgt-arm_5.2.4	
C/C++ General	ECLIPSE_HOME	C:\ti\ccsv6\eclipse\	
Debug	EXTERNAL_BUILD_ARTIFACT		
Project References	CRG_PROJ_DIR	\${ORIGINAL_PROJECT_ROOT}\\.IAR\Application\CC2640	
Run/Debug Settings	CRIGINAL_PROJECT_ROOT	C:\ti\simplelink\ble_cc26xx_2_00_00_42893\Projects\ble\TM4C_CC26xx_Demo_Peripheral\CC26xx\CCS\TM4C_CC26xx_Demo_Peripheral	
	DARENT_LOC	D:\Work\ccs_workspaces\EPGI	
	➢ PROJECT_LOC	D:\Work\ccs_workspaces\EPGI\TM4C_CC26xx_Demo_Peripheral	
	TI_RTOS_DRIVERS_BASE	C:\ti\tirtos_simplelink_2_11_01_09\packages	
	BWORKSPACE_LOC	D:\Work\ccs_workspaces\EPGI	

### Figure 13. Path Variables – CC2650 Peripheral

ype filter text	Linked Resources		<-> → →
<ul> <li>Resource</li> <li>Linked Resources</li> <li>Resource Filters</li> </ul>	Path Variables Linked Resources	the file system, including other path variables with the syntax "\${VAR}".	
CCS General		and the system, including other part variables with the system system system.	
▲ CCS Build		- 'TM4C_CC26xx_Demo_PeripheralStack':	
<ul> <li>ARM Compiler</li> <li>ARM Linker</li> </ul>	Name	Value	New
ARM Hex Utility [Disabled]	🗁 CC26XXWARE	C:\ti\tirtos_simplelink_2_11_01_09\products\cc26xxware_2_20_06_14829	Edit
Builders	CCS_BASE_ROOT	C:\ti\ccsv6\ccs_base	
C/C++ Build	CCS_INSTALL_ROOT	C:\ti\ccsv6	Remove
> C/C++ General	CG_TOOL_ROOT	C:\ti\ccsv6\tools\compiler\ti-cgt-arm_5.2.4	
Debug	🗁 ECLIPSE_HOME	C:\ti\ccsv6\eclipse\	
Project References	EXTERNAL_BUILD_ARTIFACT		
Run/Debug Settings	🗁 ORG_PROJ_DIR	\${ORIGINAL_PROJECT_ROOT}\\\JAR\Stack\CC2640	
	ORIGINAL_PROJECT_ROOT PARENT LOC	C\ti\simplelink\ble_cc26xx_2_00_00_42893\Projects\ble\TM4C_CC26xx_Demo_Peripheral\CC26xx\CCS\TM4C_CC26xx_Demo_PeripheralStack D\Work\ccs_workspaces\EPGI	
	> PROJECT_LOC	D:\Work\ccs_workspaces\EPGI\TM4C_CC26xx_Demo_PeripheralStack	
	TI RTOS DRIVERS BASE	C:\ti\tirtos_simplelink_2_11_01_09\packages	
	WORKSPACE_LOC	D:\Work\ccs_workspaces\EPGI	

### Figure 14. Path Variables – CC2650 Peripheral Stack



ype filter text	Linked Resources		(-+)
<ul> <li>Resource         <ul> <li>Linked Resources</li> <li>Resource Filters</li> <li>CCS General</li> <li>CCS Build</li> </ul> </li> </ul>	The locations of linked resource	s in the file system, including other path variables with the sy es may be specified relative to these path variables.	ntax "\${VAR}".
ARM Compiler	Defined path variables for reso		
ARM Linker	Name	Value	New
ARM Hex Utility [Disabled]	CCS_BASE_ROOT	C:\ti\ccsv6\ccs_base	Edit
> XDCtools	CCS_INSTALL_ROOT	C:\ti\ccsv6	
Builders	CG_TOOL_ROOT	C:\ti\ccsv6\tools\compiler\ti-cgt-arm_5.2.4	Remove
C/C++ Build	ECLIPSE_HOME	C:\ti\ccsv6\eclipse\	
C/C++ General	EXTERNAL_BUILD_ARTIF		
Debug	➢ PARENT_LOC	D:\Work\ccs_workspaces\EPGI	
Project References	➢ PROJECT_LOC	D:\Work\ccs_workspaces\EPGI\TM4C_BLE_RF_Node	
Run/Debug Settings	>WORKSPACE_LOC	D:\Work\ccs workspaces\EPGI	
?) Hide advanced settings		ОК	Cancel

### Figure 15. Path Variables – TM4C Slave

5. Check the TI-RTOS version and platform selection (see Figure 16 and Figure 17).

Properties for TM4C_CC26xx_Demo_Cer	tral	_ <b>—</b> X
type filter text	CCS General	↓ ↓ ↓ ↓
<ul> <li>Resource</li> <li>CCS General</li> <li>CCS Build</li> <li>ARM Compiler</li> <li>ARM Linker</li> <li>ARM Hex Utility [Disabled]</li> <li>XDCtools</li> <li>Builders</li> <li>C/C++ Build</li> <li>C/C++ General</li> </ul>	Configuration: FlashROM [Active]  Main Main RTSC XDCtools version: 3.30.6.67_core A Products and Repositories & Order	age Configurations
Debug Project References Run/Debug Settings	<ul> <li>▷ SYS/BIOS</li> <li>■ TI-RTOS for SimpleLink Wireless MCUs</li> <li>■ 2.13.1.09</li> <li>♥ 2.11.1.09</li> <li>▷ ■ TI-RTOS for TivaC</li> <li>■ Other Repositories</li> <li>♥ ⊕ S{TARGET_CONTENT_BASE} [C:/ti/ccsv6/ccs_base]</li> </ul>	Add Edit Remove Select All Deselect All
	Target:     ti.targets.arm.elf.M3       Platform:     ti.platforms.simplelink:CC2650F128       Build-profile:     release	•
Hide advanced settings	ОК	Cancel

Figure 16. TI-RTOS Product Selection – CC2650



Software Setup

www.ti.com

Properties for TIVA_RF_Node		
type filter text A Resource Linked Resources Resource Filters CCS General CCS Build ARM Compiler ARM Linker ARM Hex Utility [Disabled] XDCtools Builders C/C++ Build C/C++ General Debug Project References Run/Debug Settings	CCS General         Configuration:       Debug [Active]         Main       RTSC         XDCtools version:       3.31.1.33_core         Products and Repositories       Order         Products and Repositories       Products and Repositories         Image:       TI-RTOS for SimpleLink Wireless MCUs         Image:       Y14.0.0         Image:       Y14.0.10         Image:       Y14.0.10	
Hide advanced settings	OK	Cancel

#### Figure 17. TI-RTOS Product Selection – TM4C Slave

- 6. Demo executable list
  - TM4C:
    - Wired Master TM4C\_BLE\_Master.out
    - RF Node Slave TM4C\_BLE\_RF\_Node.out
  - CC2650:
    - Central
      - TM4C\_CC26xx\_Demo\_CentralStack.out
      - TM4C\_CC26xx\_Demo\_Central.out
    - Peripheral RF Node
      - TM4C\_CC26xx\_Demo\_PeripheralStack.out
      - TM4C\_CC26xx\_Demo\_Peripheral.out

**NOTE:** <project\_name>Stack.out must be be loaded before loading the <project\_name>.out.



#### 6 Demonstration Execution

For details about demonstration execution, see <u>BLE-Enabled IoT Node With High-Performance MCU</u> <u>Reference Design</u>.

#### 6.1 Debug Port Setup for TM4C Wired Master Device

- 1. Open a terminal window (like Hyperterminal or TeraTerm) and connect to the Stellaris® Virtual Serial Port COM port corresponding to the TM4C-wired master device.
- 2. Select the baud rate as 9600, data bits as 8, parity as none, stop bits as 1, and flow control as none.
- 3. When the Ethernet cable is connected to TM4C-wired master device, the IP address is acquired and displayed on the debug terminal.

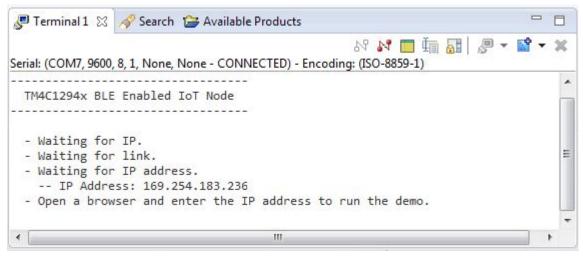


Figure 18. Debug Terminal

#### 6.2 Demonstration Execution on the Web Page

- 1. Power on the peripheral RF node.
- 2. From a PC connected to the same network as the central node, open a browser and enter the assigned IP address to open the Web page.
- 3. Press Connect and wait until the status displays as CONNECTED (can take up to 8 seconds).
- 4. Demo 1: **Toggle LED** Pressing Toggle LED toggles the LED1 on the Slave TM4C-Connected LaunchPad.
- 5. Demo 2: LED Animation Controls the LED animation speed on the Slave TM4C. 0% is 1-s period and 100% is 20 ms.
- 6. Demo 3: Get and Clear Button Press Count
  - Get Button 1 Count displays the press count of the button SW1 on the slave TM4C board.
  - Clear Button 1 Count clears the press count of the button SW1 on the slave TM4C board.
  - Get Button 2 Count displays the press count of the button SW2 on the slave TM4C board.
  - Clear Button 2 Count clears the press count of the button SW2 on the slave TM4C board.
- 7. Demo 4: **Get Temperature** Get Temperature gives the device junction temperature of the Slave TM4C microcontroller.

TEXAS INSTRUMENTS

www.ti.com

Demonstration Execution

TEXAS INSTRUM	Tiva <sup>™</sup> C Series TM4C1294XL Evaluation Kit EK-TM4C1294XL
About TI iva™ C Series Overview M4C1294NCPDT Block	TM4C1294x BLE Enabled IoT Node
Diagram K-TM4C1294XL Product Page	This demonstration shows how to perform control and status reporting using HTTP requests embedded within Javascript code on the web page itself. Using this method is possible to update sections of text on the current page without the need to refresh the entire page.
Tīva™ TM4C129x Family Product Page	BLE Slave Connection
M4C1294x BLE Enabled IoT Node (HTTP Requests)	Connect BLE Slave : NOT CONNECTED
	*** Note: Below Demo's can be performed only when connected to BLE Slave.
	Toggle USER LED and report the state of the LED
	Toggle LED USER LED: OFF
	Set the speed of the animation LED. Speed is expressed as a percentage.
	Current Speed: 10% Set Speed 10 %
	Control other I/O on the Slave MCU.
	Get Button 1 Counts0Clear Button 1 CountsGet Button 2 Counts0Clear Button 1 Counts
	Get Temprature

Figure 19. Tiva™ C Series TM4C1294XL Evaluation Kit



#### 7 Resources

To download the resource files for this reference design, refer to <u>http://www.ti.com/tool/TIDM-TM4C129XBLE</u>.

#### 8 References

- 1. TivaWare for C Series: http://www.ti.com/tool/SW-TM4C
- Stellaris In-Circuit Debug Interface (ICDI) and Virtual COM Port Driver Installation Instructions (SPMU287)
- 3. TI RTOS: http://www.ti.com/tool/ti-rtos
- 4. BLE Stack-2: http://www.ti.com/tool/ble-stack-archive
- 5. EK-TM4C1294XL LaunchPad: http://www.ti.com/tool/ek-tm4c1294xl
- 6. CC2650 Development Kit: http://www.ti.com/tool/cc2650dk

#### **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 Copyright © 2015, Texas Instruments Incorporated