Application Report
SWRA534–January 2017

Running Standalone Bluetooth® low energy Applications on CC2650 Module

Nathan Siegel

ABSTRACT

This application report details how to run standalone *Bluetooth*® low energy (BLE) applications on the SimpleLink™ Bluetooth low energy wireless MCU Module (CC2650MODA). In addition to the simple network-processor configuration included in the Bluetooth low energy software stack, the CC2650 module can be used for standalone single-chip applications (such as a BLE peripheral or beacon) with minimal modification to the project. The necessary changes to the board files and the setup of the hardware and software are detailed in this document.

Contents

Trademarks

1 Introduction

The *TI SimpleLink™ Bluetooth® low energy CC2650 Module BoosterPack™ Plug-in Module* (BOOSTXL-CC2650MA) allows users to quickly and easily add Bluetooth low energy to a LaunchPad™ development kit for developing network processer-based BLE applications. The CC2650 module can also be used to function as a standalone, System-on-Chip (SoC) device that can run additional sample applications using TI's royalty-free Bluetooth low energy software stack (BLE-Stack) software development kit (SDK). This application report details how to run these standalone applications on the CC2650 module BoosterPack™, or on any custom board that incorporates the CC2650 module.

The CC2650 module contains the CC2650 wireless microcontroller (MCU) with an integrated antenna, an ARM® Cortex®-M3 32-bit MCU, in-system flash memory, 15 I/Os, and is precertified for FCC/IC, CE, and ARIB radio standards. When used with TI's BLE-Stack, the module also has a Bluetooth RF-PHY qualified component that reduces the Bluetooth qualification testing. In addition, the module is pretested and uses an optimized antenna layout, reducing the need for expensive board redesigns and performing time-consuming RF tests with expensive test equipment on the production line. See *CC2650MODA SimpleLink™ Bluetooth® low energy Wireless MCU Module* for details on precertification.

It is assumed that the reader has working knowledge of the Bluetooth core specification version 4.2 [2] and is familiar with the concepts and build procedure documented in TI's BLE software developer's guide [3] provided with the BLE-Stack SDK [4]. This application report assumes usage of BLE-STACK 2.2.1 which supports all Bluetooth 4.2 LE features. This application report uses the simple BLE peripheral sample application that is provided with the BLE-STACK SDK as a functional example. Because all device firmware resides and executes internally on the CC2650 wireless MCU, other BLE sample applications can run on the CC2650 module if the project is configured to account for the correct mapping of external peripheral I/O connections. For example, button presses and LEDs (used for demonstration purposes in some of the sample applications provided with the BLE-STACK SDK) require a specific I/O configuration mapping that is development kit specific.

TI recommends that standalone BLE applications are developed on the CC2650 LaunchPad (LAUNCHXL-CC2650) [5], then this application report can be used to provide guidance on porting those applications to the CC2650 module for product deployment. The CC2650 LaunchPad is a complete development kit featuring an integrated XDS110 JTAG debugger and access to all device I/Os. In addition, the XDS110 on the LaunchPad can function as a JTAG programmer for external CC26xx wireless MCUs. The LaunchPad includes a CC2650 wireless MCU in a 7 × 7 QFN package with 31 I/Os, and the CC2650 module uses a 5 × 5 QFN package with 15 I/Os. Aside from the number of I/Os and the RF configuration, all software is compatible and exchangeable across all CC2650 and CC2640 package configurations.

2 Device Setup

This section details the procedure for setting up the hardware and software to run sample applications on the CC2650 module. This example uses the CC2650 Module BoosterPack, and it is provided as an example of the required changes when using the CC2650 module placed on a custom board.

2.1 Required Hardware and Software

Hardware:

- 1 CC2650 Module BoosterPack
- 1 CC2650 LaunchPad
- 10-pin JTAG debug cable (provided with the CC2650 Module BoosterPack)

Software:

- *Bluetooth low energy software stack* (BLE-STACK-2-2-1)
- *IAR Embedded Workbench*

  or
- *Code Composer Studio™* (CCS) (see *CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack 2.2.1* for version requirements)

2.2 Hardware Setup

Use the following instructions to set up the hardware.

1. Connect the 10-pin JTAG cable to the JTAG pins on the CC2650 module, then connect the other end of the cable to the XDS110 Out pins on the CC2650 LaunchPad (see Figure 1).

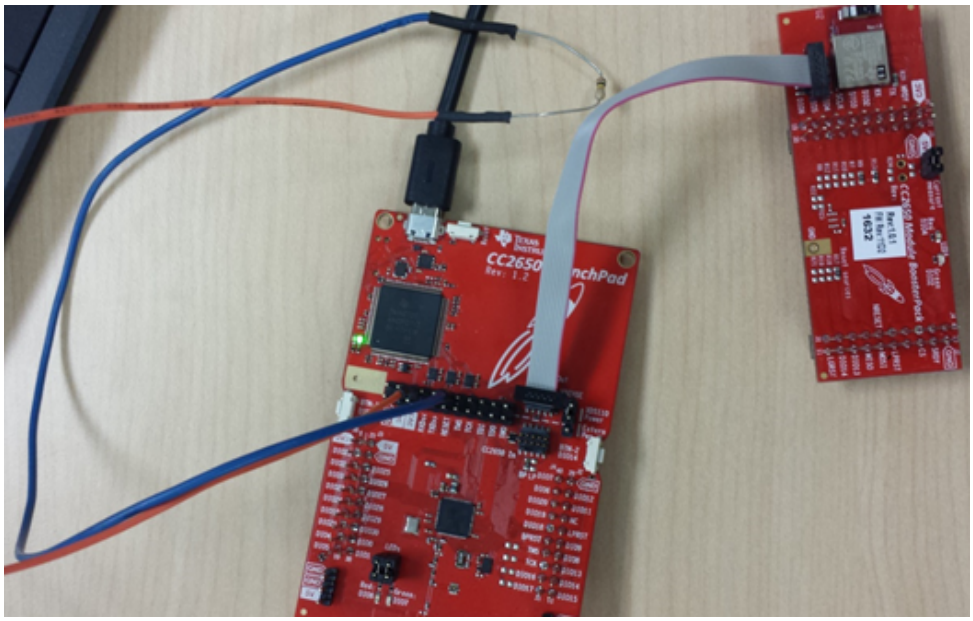

Figure 1. CC2650 Module to CC2650 LaunchPad Connection

2. Ensure that the necessary jumpers are removed to isolate the XDS110 from the onboard CC2650 of the LaunchPad (see the yellow box in Figure 2). Also, verify that the XDS110 power jumper is selected to supply power to the CC2650 module.

   Complete step 3 only if using revision 1.2 or later of the CC2650 LaunchPad.

3. Insert a 100-k resistor between the 3V3 and reset pins on the XDS side of the JTAG-CC2650 jumper header on the launchpad (see the orange and blue headers in Figure 1).

   Step 3 is required with revision 1.2 or later because the CC2650 Module BoosterPack was designed as a BoosterPack to work with a LaunchPad.

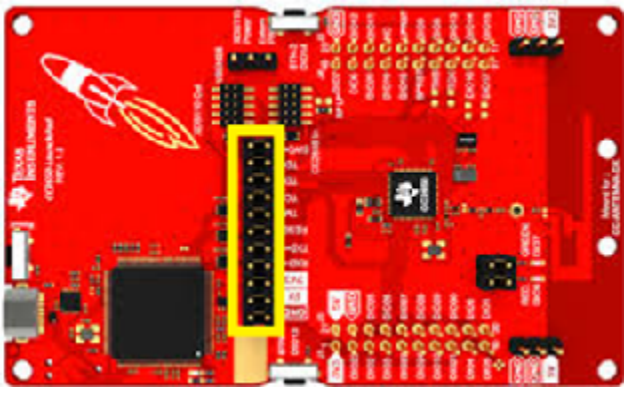

Figure 2. Jumpers to Remove on CC2650 LaunchPad

After completing the previous steps, the provided micro-USB cable can be used to power and program the CC2650 module as detailed in the software developer's guide.

3 Running Sample Applications on CC2650 Module

The sample applications provided with the Bluetooth low energy software stack will run on the CC2650 module, but they require some changes because of the differences in pin mappings and external peripherals. This section details the process of running the sample applications on the device and the necessary changes to the board files. A simple BLE peripheral sample application is used as an example.

3.1 Downloading and Running Sample Applications

Refer to *CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack 2.2.1* for information on building and downloading the sample applications using IAR Embedded Workbench or CCS. If using the CC2650 LaunchPad to program the module, the project must be configured to use the XDS110 debugger. All required changes in this application report are based on the simple BLE peripheral (simple_peripheral) sample application that is configured for the CC2650 LaunchPad development kit.

3.2 Required Board-File Changes

The board files for the application project must be modified because of the differences in pin mappings and external peripherals. This application report shows these modifications by changing the existing LaunchPad board files, but custom board files may also be created. Refer to the peripherals and drivers section of *CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack 2.2.1* for details on using custom board files.

The CC2650_LAUNCHXL.h file must be updated to contain the correct pin mappings, as well as the RF front end and bias configuration. This file is located in the following location (if the BLE-STACK is installed to the default location):
C:\ti\simplelink\ble_sdk_2_02_01_18\src\boards\CC2650_LAUNCHXL\CC2650_LAUNCHXL.h.

NOTE: The CC2650 LaunchPad and CC2650 module use different RF front end and bias configurations. Ensure that CC2650EM_5XD is defined in the board file to match the externally biased, differential output RF configuration of the module. An incorrect RF configuration results in substantially degraded RF performance.

Refer to the following code snippet for required changes to the CC2650_LAUNCHXL.h file.

```
/* RF Configuration definition */
#define CC2650EM_5XD     /* RF Configuration for CC2650 Module */
// #define CC2650EM_7ID  /* Default RF config for CC2650 LaunchPad */

/* Mapping of pins to board signals using general board aliases
 *      <board signal alias>        <pin mapping>
*/

/* Discrete outputs */
#define Board_RLED                  IOID_4
#define Board_GLED                  IOID_2
#define Board_LED_ON                1
#define Board_LED_OFF               0

/* Discrete inputs */
#define Board_BTN1                  PIN_UNASSIGNED
#define Board_BTN2                  PIN_UNASSIGNED

/* UART Board */
#define Board_UART_RX               IOID_1          /* RXD  */
#define Board_UART_TX               IOID_0          /* TXD  */
#define Board_UART_CTS              PIN_UNASSIGNED  /* CTS  */
#define Board_UART_RTS              PIN_UNASSIGNED  /* RTS  */

/* SPI Board */
#define Board_SPI0_MISO             IOID_12         /* RF1.20 */
#define Board_SPI0_MOSI             IOID_11         /* RF1.18 */
#define Board_SPI0_CLK              IOID_10         /* RF1.16 */
#define Board_SPI0_CSN              PIN_UNASSIGNED
#define Board_SPI1_MISO             PIN_UNASSIGNED
#define Board_SPI1_MOSI             PIN_UNASSIGNED
#define Board_SPI1_CLK              PIN_UNASSIGNED
#define Board_SPI1_CSN              PIN_UNASSIGNED

/* I2C */
#define Board_I2C0_SCL0             PIN_UNASSIGNED
#define Board_I2C0_SDA0             PIN_UNASSIGNED

/* SPI */
#define Board_SPI_FLASH_CS          PIN_UNASSIGNED
#define Board_FLASH_CS_ON           0
#define Board_FLASH_CS_OFF          1
```

These changes work with the CC2650 Module BoosterPack. If the CC2650 module is placed on a custom board, it will have no external I/O peripheral connections by default, therefore the remaining pins must be set to PIN_UNASSIGNED. The use of a custom board with the CC2650 module will require changes similar to the previous code snippet. For custom board layouts, refer to the board schematic for determining the correct pin configuration.

Next, the CC2650_LAUNCHXL.c board file must be updated to contain the correct fields in the PIN_Config GPIO initialization table. This file is located in the following location (if the BLE-STACK is installed to the default location): C:\ti\simplelink\ble_sdk_2_02_01_18\src\boards\CC2650_LAUNCHXL\CC2650_LAUNCHXL.h.

See the following code snippet for required changes to the table.

```
const PIN_Config BoardGpioInitTable[] = {

    Board_RLED      | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,      /*
LED initially off */
    Board_GLED      | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,      /*
LED initially off */
    Board_UART_RX   | PIN_INPUT_EN | PIN_PULLDOWN,                                            /*
UART RX via debugger back channel */
    Board_UART_TX   | PIN_GPIO_OUTPUT_EN | PIN_GPIO_HIGH | PIN_PUSHPULL,                     /*
UART TX via debugger back channel */
    Board_SPI0_MOSI | PIN_INPUT_EN | PIN_PULLDOWN,                                           /*
SPI master out - slave in */
    Board_SPI0_MISO | PIN_INPUT_EN | PIN_PULLDOWN,                                           /*
SPI master in - slave out */
    Board_SPI0_CLK  | PIN_INPUT_EN | PIN_PULLDOWN,                                           /*
SPI clock */

    PIN_TERMINATE
};
```

These changes work with the CC2650 Module BoosterPack. If the CC2650 module is placed on a custom board, the only value in the table shall be PIN_TERMINATE unless the table is configured to match the board schematic. Do not exceed the maximum number of I/Os (15) for the CC2650 module 5 × 5 package.

If the application has any predefined symbols that would cause the application to expect to use any inputs or outputs (such as the display driver), these must be disabled in the project, typically in the Predefined Symbols of the project. The simple BLE peripheral-project configuration for the LaunchPad does not have any of these enabled by default. The accessing preprocessor symbols section of *CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack 2.2.1* details how to access the predefined symbols fo IAR Embedded Workbench and CCS.

4 References

1. *TI SimpleLink™ Bluetooth® low energy CC2650 Module BoosterPack™ Plug-in Module*
2. *Bluetooth Core Specification, Version 4.2*
3. *CC2640 and CC2650 SimpleLink™ Bluetooth® low energy Software Stack 2.2.1*
4. *Bluetooth low energy software stack*
5. *SimpleLink™ CC2650 Wireless MCU LaunchPad™ Kit*
6. *IAR Embedded Workbench*
7. *Code Composer Studio™*

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY 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 products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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 RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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