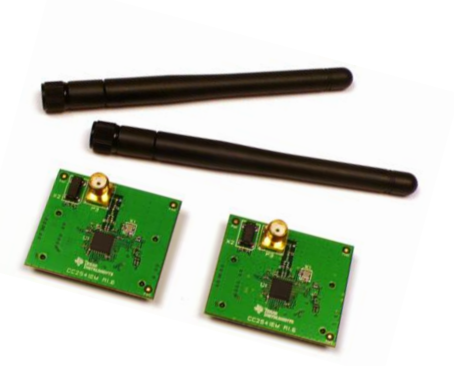
Bluetooth® Low Energy

CC2540 Development Kit

CC2541 Evaluation Module Kit

User's Guide



Document Number: SWRU301A

Development Kit Part Number: CC2540DK, CC2541EMK

TABLE OF CONTENTS

1. References

The following references provide additional information on the CC2540/41, the Texas Instruments *Bluetooth®* low energy (BLE) stack and the BLE specification in general. (All path and file references in this document assume that the BLE development kit software has been installed to the default path C:\Texas Instruments\BLE-CC254X-1.1B\)

1.1 Printed Copy Included in the Box with CC2540DK

[1] CC2540 Development Kit Quick Start Guide (SWRU300)

1.2 Printed Copy Included in the Box with CC2541EMK

[2] CC2541 Evaluation Module Kit Quick Start Guide (SWRU311)

1.3 Included with Texas Instruments *Bluetooth* Low Energy Software Installer

(The software installer is available for download at **http://www.ti.com/blestack**)

[3] Texas Instruments *Bluetooth®* Low Energy Software Developer's Guide (SWRU271A)
C:\Texas Instruments\BLE-CC254X-1.1b\Documents\TI_BLE_Software_Developer's_Guide.pdf

[4] TI BLE Vendor Specific HCI Reference Guide
C:\Texas Instruments\BLE-CC254X-1.1b\Documents\TI_BLE_Vendor_Specific_HCI_Guide.pdf

[5] Texas Instruments BLE Sample Applications Guide (SWRU297)
C:\Texas Instruments\BLE-CC254X-1.1b\Documents\TI_BLE_Sample_Applications_Guide.pdf

1.4 Available from *Bluetooth* Special Interest Group (SIG)

[6] *Specification of the Bluetooth System*, Covered Core Package version: 4.0 (30-June-2010)
https://www.bluetooth.org/technical/specifications/adopted.htm

2. Introduction

Thank you for purchasing a Texas Instruments (TI) *Bluetooth®* low energy (BLE) Kit. The purpose of this document is to give an overview of the hardware and software included in the CC2540 Development Kit (CC2540DK) and the add-on CC2541 Evaluation Module Kit (CC2541EMK).

The information in this guide will get you up and running with the kit; however for more detailed information on BLE technology and the TI BLE protocol stack, please consult the Texas Instruments *Bluetooth®* Low Energy Software Developer's Guide [3].

2.1 CC2540DK Contents Overview

The CC2540DK contains the following hardware components:

- **2 x SmartRF05 Evaluation Boards (SmartRF05EB)**
- **2 x CC2540 Evaluation Modules (CC2540EM)**
- **2 x Pulse Antennas**
- **1 x CC2540 USB Dongle**
- **Cables**

Figure 1 – CC2540DK

2.2 CC2541EMK Contents Overview

The CC2541EMK contains the following hardware components:

- **2 x CC2541 Evaluation Modules (CC2541EM)**
- **2 x Pulse W1010 Antennas**
- **Cables**

The kit is FCC and IC certified and tested/complies with ETSI/R&TTE over temperature from 0 to +35°C. The antenna, W1010 from Pulse, is a ¼ wave dipole antenna with 2 dBi gain.

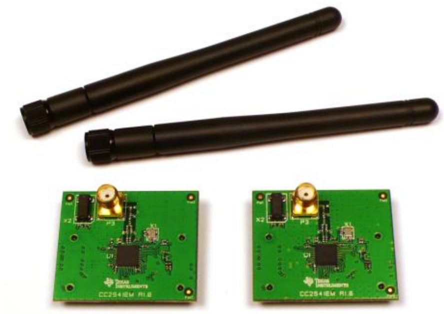


Figure 2 - CC2541EMK

Caution! The kit contains ESD sensitive components. Handle with care to prevent permanent damage. To minimize risk of injury, avoid touching components during operation if symbolized as hot.

2.3 System Requirements

To use the TI BLE software, a PC running Microsoft Windows (XP or later; 32-bit support only) is required, as well as Microsoft .NET Framework 3.5 Service Pack 1 (SP1) or greater.

In order to check whether your system has the appropriate .NET Framework, open up the Windows Control Panel, and select “Add or Remove Programs”. Amongst the list of currently installed programs, you should see “Microsoft .NET Framework 3.5 SP1”, as shown in Figure 3.

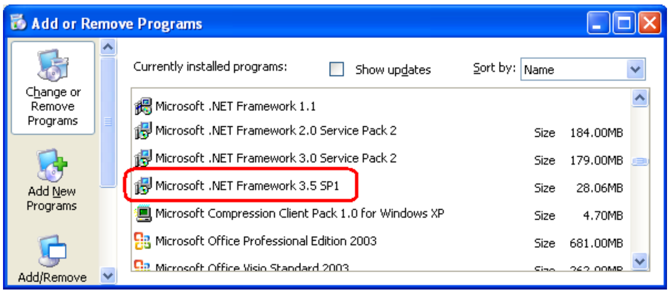


Figure 3 System Requirements, .NET Framework 3.5 SP1

If you do not see it in the list, you can download the framework from Microsoft.

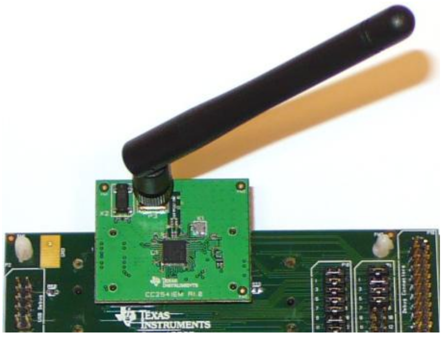
From a hardware standpoint, the Windows PC must contain at least one, and up to three, free USB ports. With one free port, a single CC2540/41 device can be flashed or debugged, or the BLE sniffer can be used. In order to simultaneously flash or debug both evaluation modules (EMs) while running the BLE sniffer software, three USB ports are required.

IAR Embedded Workbench for 8051 development environment is required in order to make changes to the BLE software. More information on IAR can be found in the Texas Instruments *Bluetooth*® Low Energy Software Developer’s Guide [3].

For the CC2541EMK, it is required to use SmartRF05 Boards **Rev. 1.8.1 or later.** More information about the SmartRF05EB can be found in www.ti.com/lit/swru210. The CC2541EM boards can also be plugged into a battery board (see www.ti.com/tool/soc-bb) for standalone operation.

3. Getting Started with the SimpleBLE Demo

3.1 Hardware Setup



Connect the antennas to the SMA connectors on the EMs. Tighten the antenna's screw firmly on to the SMA connector. If not properly connected, you might see reduced RF performance. Next, mount the EMs firmly on to connectors P5 and P6 on the SmartRF05EB.

3.2 Power Options

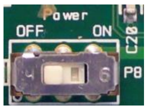
There are several ways of applying power to the SmartRF05EB.

- 2 x 1.5 V AA Batteries
- USB
- External Power Supply

For the batteries and USB, there are voltage regulators on the SmartRF05EB that will set the on-board voltage to 3.3 V. The external power supply should set a voltage that does not exceed 3.3 V.

Note that there should only be one active power source at any time. To minimize risk of personal injury or property damage, **never use rechargeable batteries** to power the board.

3.3 Power the Boards



Find jumper P11 on the top side of each SmartRF05EB. This jumper is used to set the power source for the board. Set P11 to "1-2" if you are using battery power. Set P11 to "2-3" if you are using USB or an external power supply.

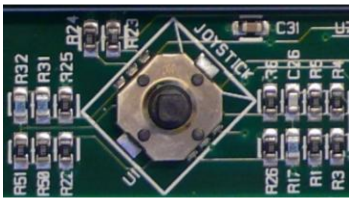
Once you have set P11, find switch P8 on the top side of each SmartRF05EB. To power up the boards, flip the switch from the "OFF" to "ON".

3.4 Start-up Screen

One of the EMs will be pre-loaded with the SimpleBLECentral application, while the other will be pre-loaded with the SimpleBLEPeripheral application. The LCD screens on the two SmartRF05EBs should display messages similar to those below:

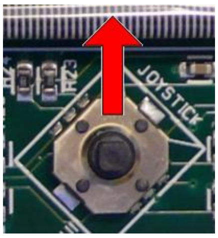
The "0x…" value displayed on each board is the device address. Every CC2540/41 device has a unique address.

3.5 Using the Joystick



The SimpleBLEPeripheral application runs autonomously and does not require any user interaction. The SimpleBLECentral application, however, requires user interaction by means of joystick U1. Find joystick U1 on the top side of the SmartRF05EB, immediately below the LCD. The joystick has five different movements: it can be moved up, down, left, right, and it can be pressed in, just like a button. Each movement performs different actions depending on the state of the device.

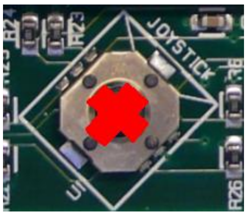
3.6 Device Discovery



Before the two devices can connect, the central device must first discover the peripheral device. To perform device discovery, press up on joystick U1 once. The LCD on the central device should display “Discovering…”.

After a few seconds, it should display “Devices Found 1 / <- To Select”. This means that the central device successfully discovered the peripheral. Press left on joystick U1 to view the address of the peripheral device. This address should match the address seen on the peripheral’s LCD.

3.7 Establish Connection



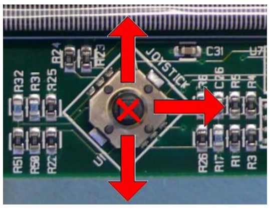
To establish a connection with the peripheral, press joystick U1 in towards the board (push it inwards, like it is a button). Once the connection is established, the central device will automatically perform service discovery on the peripheral using the BLE GATT protocol. This should complete within a few seconds.

The two LCD screens should appear as in the images below, with the central still displaying the peripheral’s address and the peripheral having changed from “Advertising” to “Connected”:

Be careful that you don’t “double tap” U1 which would terminate the connection immediately, giving *Disconnected Reason: 22.*

3.8 Connected Operations

Once the connection has been established and service discovery is complete, you can perform the following operations using **joystick U1** on the central device:



UP: Read / Write Data
RIGHT: Connection Parameter Update
DOWN: RSSI Monitoring
IN: Terminate Link

3.9 Read / Write Data

Pressing up on U1 will send a read request to the peripheral device. One byte of data will be read, and the value will be displayed. Pressing up again will send a write request, and one byte of data will be written to the peripheral. The peripheral’s LCD should display the written value each time this is done.

Bluetooth low energy is an ideal technology for transmission of small amounts of data between two devices while consuming very little power, as is demonstrated here.

Continuing to press up on the joystick will alternate between reads and writes, with the value incrementing each time.

3.10 Monitor RSSI

Pressing down on U1 will turn on RSSI (received signal strength indication) monitoring. The RSSI will be displayed on the LCD in units of negative **dBm**.

If the boards are moved farther apart from each other, the RSSI will drop (since the value is negative, a higher number means lower RSSI). If they are moved closer together, the RSSI should rise.

Pressing down on U1 again will turn off RSSI monitoring.

3.11 Connection Parameter Update

Pressing right on U1 will send a connection parameter update request to the peripheral to use a longer connection interval. This will result in much longer latency when performing data reads and writes; however the power consumed by both devices is significantly reduced.

3.12 Terminate Link

Pressing U1 in towards the board will terminate the link. The peripheral will return to an advertising state. The central device will display a “Reason” code, which indicates why the disconnection occurred (values are defined in the BLE stack API).

In this case, the reason code of 22 indicates that the link termination was initiated by the central device. In the event that the peripheral device goes out of range or has power disconnected from it, you will see a reason code of 8 which indicates that a link timeout has occurred.

You can now perform device discovery and re-connect to the peripheral if desired.

3.13 SimpleBLE Demo Source Code

The project and source code files for these applications (as well as many others) are included with the Bluetooth low energy (BLE) stack from Texas Instruments, which can be downloaded at www.ti.com/blestack.

The two projects implementing this demo are called SimpleBLECentral (Master configuration) and SimpleBLEPeripheral (Slave configuration). These can be modified as desired, and should provide a good framework for developing your own custom BLE applications.

More details on these projects can be found within the BLE Software Developer’s Guide [3], which is included with the stack.

4. Using BTool

BTool is a PC Application that allows a user to form a connection between two BLE devices. BTool works by communicating with a CC2540/41 acting as a network processor by means of HCI vendor specific commands. You will be able to perform the same functions very similar to those in the SimpleBLE demo; however with BTool you will have much broader control of the central device.

More information on the network processor configuration and the HostTestRelease project can be found in the Texas Instruments *Bluetooth*® Low Energy Software Developer's Guide [3]. More information on the HCI interface, as well as details on the HCI vendor specific commands that are used by the CC2540/41, can be found in the TI BLE Vendor Specific HCI Reference Guide [4].

For this section, a PC running windows 7 has been used, but the procedures are essentially the same for other windows version, such as XP.

Host Board in the following context refers to the hardware platform consisting of a SmartRF05+EM setup or a CC2540USB Dongle, since either can be used.

4.1 Using SmartRF05EB + CC2540EM/CC2541EM as Host Board

4.1.1 Load HostTestRelease Project on EM using SmartRF05EB

Before using BTool, the HostTestRelease (network processor) application must be loaded onto the central device. A hex file containing the HostTestRelease application can be found at the following location:

C:\...\BLE-CC254X-1.1b\Accessories\Hex_Files\CC254*X*_ble_SmartRF_HostTestRelease_Master.hex

For instructions on loading a hex file, please see section 5. It is recommended that you load this firmware on the SmartRF05EB/EM boards that were previously running the SimpleBLECentral application, and leave the SimpleBLEPeripheral application on the other set of boards.

4.1.2 Connect SmartRF05EB to PC

BTool, running on a Windows PC, communicates with the EM by means of the serial port (RS-232) interface on the SmartRF05EB. With a standard DB-9 serial cable, connect the SmartRF05EB to your PC's serial port. If the PC does not have a serial port, a USB-to-serial converter will be required.

4.2 Determining the COM Port

You will need to know which COM port Windows has assigned to the serial port. To find out, right-click on the "Computer" icon on your Start and select "Properties", shown in Figure 4.

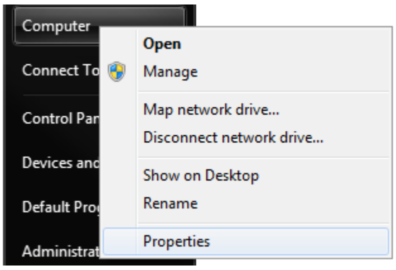


Figure 4 Win7 PC, Finding Computer Properties

The "System" window should open up. Click "Device Manager":

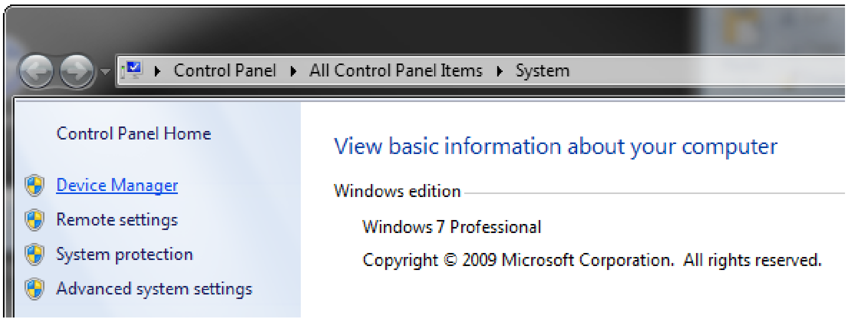


Figure 5 Win7 PC, Finding Device Manager

A list of all hardware devices should appear. Under the section "Ports (COM & LPT)", the port should appear. Next to the name should be the port number (for example, the CC2540USB Dongle uses COM10 in Figure 6):

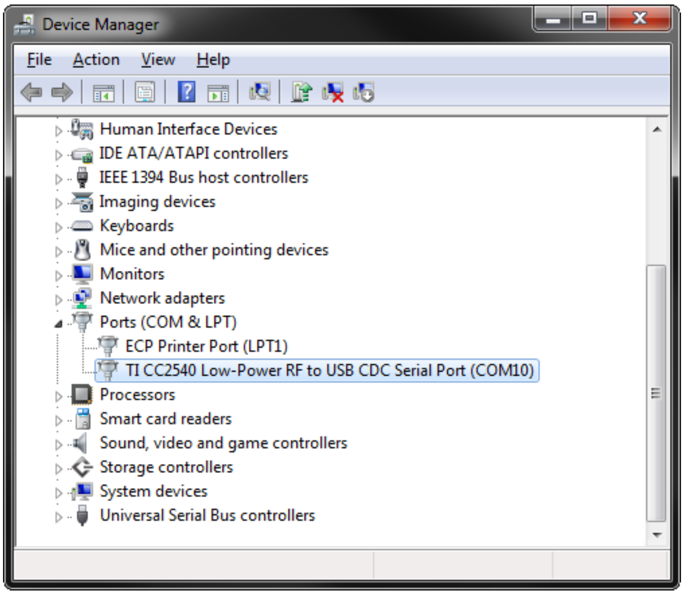


Figure 6 Win7 PC, Connected Ports List

Take note of this port number, as it will be needed in order to use BTool. You may close the device manager at this point.

4.3 Starting the Application

To start the application go into your programs by choosing Start > All Programs > Texas Instruments > Bluetooth-LE-1.1b > BTool. On Start-up you should be able to set the Serial Port Settings. Set the “Port” value to the COM port earlier noted in Section 4.2. For the other settings, use the default values as shown in Figure 7. Press “OK” to connect to the Host Board.

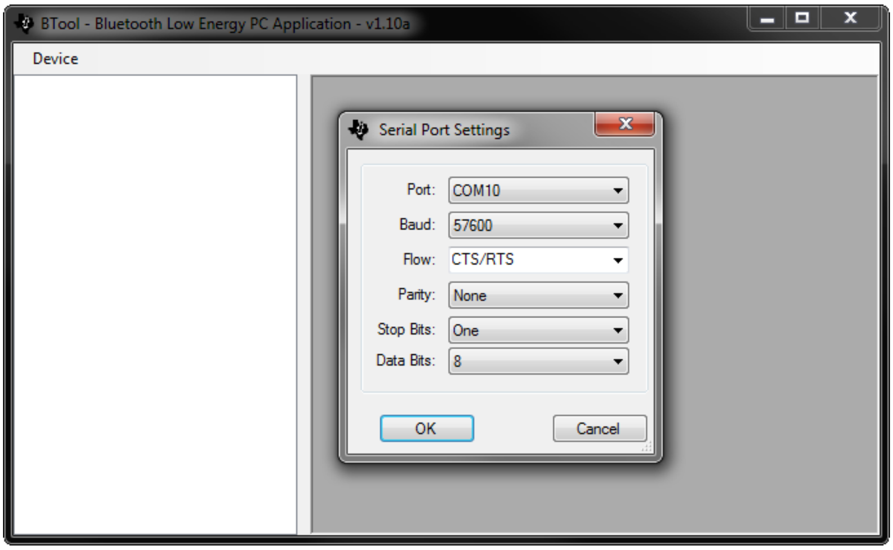


Figure 7 BTool, Serial Port settings

When connected you should see the screen presented in Figure 8. The screen indicates that you now have a serial port connection to the Host Board. The screen is divided up into a few sections: the left sidebar contains information on the Host Board status. The left side of the sub-window contains a log of all messages sent from the PC to the Host Board and received by the PC from the Host Board. The right side of the sub-window contains a GUI for control of the Host Board.

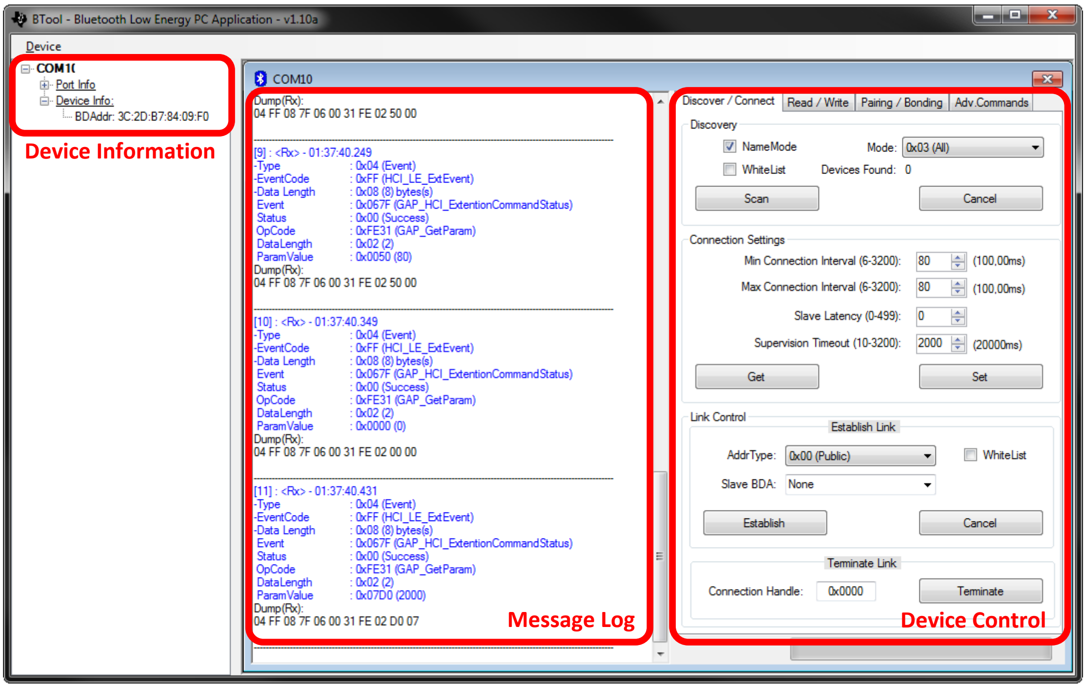


Figure 8 BTool, Overview

4.4 Creating a BLE Connection between Central and Peripheral Devices

At this point the central device (connected to the PC) is ready to discover other BLE devices that are advertising. If you have left the SimpleBLEPeripheral application running on one SmartRF05EB, you should be ready to use BTool. As long as the SmartRF05EB running the SimpleBLEPeripheral is powered up and un-connected, it should be in discoverable (advertising) mode.

4.4.1 Scanning for Devices

Press the “Scan” button under the “Discover / Connect” tab as shown in Figure 9.

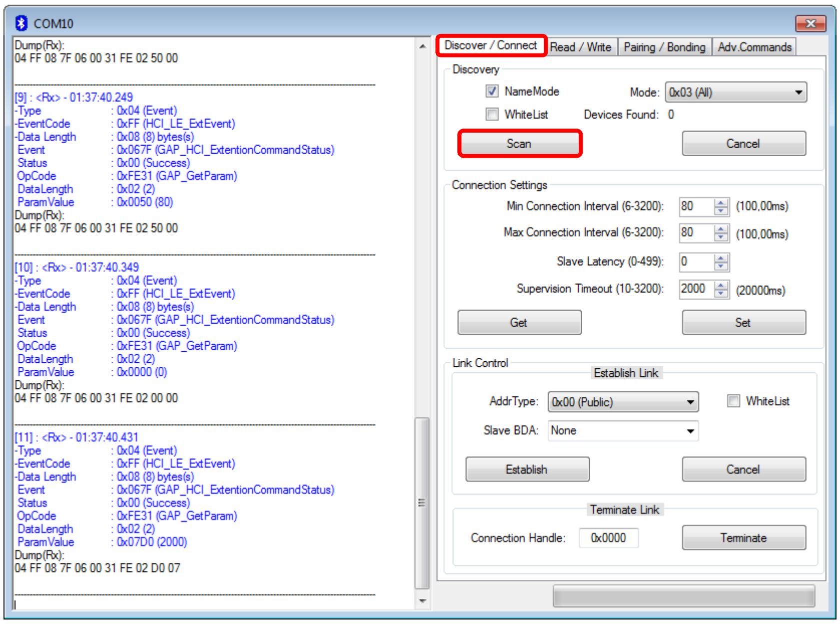


Figure 9 BTool, Scan for Devices

The central device will begin search for other BLE devices. As devices are found, the log on the left side of the screen will display the devices discovered. After 10 seconds, the device discovery process will complete, and the central device will stop scanning. A summary of all the scanned devices will be displayed in the log window. In the example in Figure 10, one peripheral device was discovered while scanning. If you do not want to wait through the full 10 seconds of scanning, the “Cancel” button can be pressed alternatively, which will stop the device discovery process. The address of any scanned devices will appear in the “Slave BDA” section of the “Link Control” section in the bottom right corner of the sub-window.

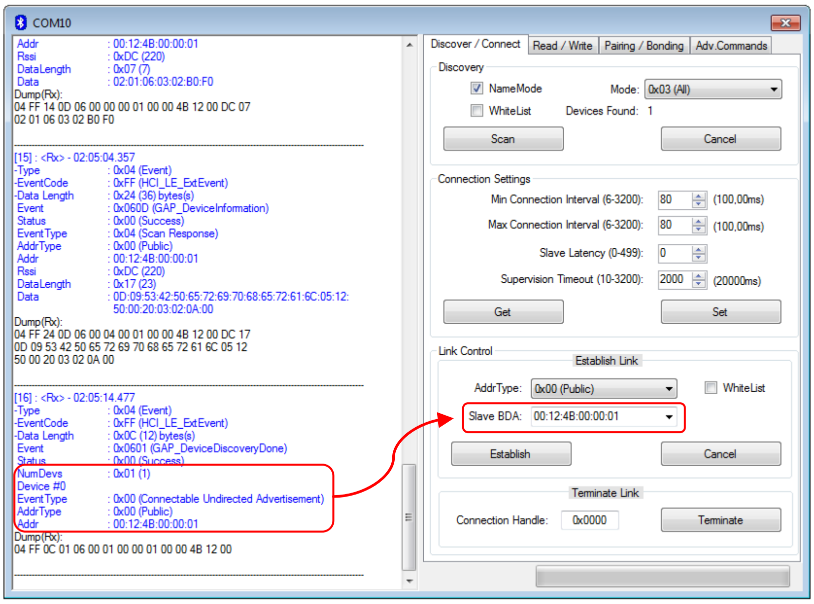


Figure 10 BTool, Slave Address

4.4.2 Selecting Connection Parameters

Before establishing a connection, you will want to set up the desired connection parameters. The default values of 100ms connection interval, 0 slave latency, and 20000ms supervision timeout should serve as a good starting point; however for different applications you may want to experiment with these values.

Once the desired values have been set, be sure to click the “Set” button; otherwise the settings will not be saved. Note that the connection parameters must be set before a connection is established; changing the values and clicking the “Set” button while a connection is active will not change the settings of an active connection. The connection must be terminated and re-established to use the new parameters. (The *Bluetooth* specification does support connection parameter updates while a connection is active; however this must be done using either an L2CAP connection parameter update request, or using a direct HCI command. More information can be found in the Specification of the *Bluetooth* System [6])

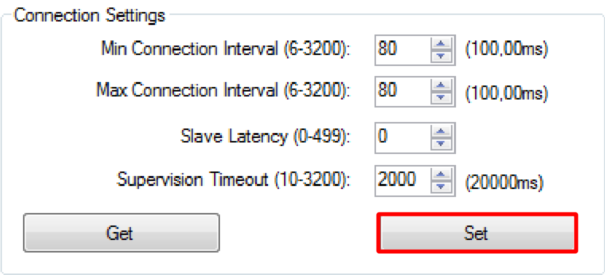


Figure 11 BTool, Connection Settings

4.4.3 Establishing a Connection

To establish a connection with the peripheral device, select the address of the device to connect with, and click the “Establish” button as shown in Figure 12.

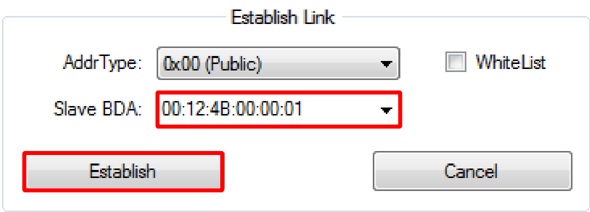


Figure 12 BTool, Establish Connection

If the set of connection parameters are invalid (for example, if the combination of connection parameters violates the specification), the message window will return a “GAP_EstablishLink” event message with a “Status” value of “0x12 (Not setup properly to perform that task)”, as shown in Figure 13. The parameters will have to be corrected before a connection can be established.

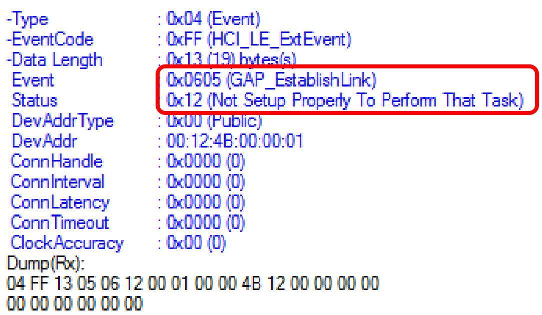


Figure 13, BTool, Invalid Connection Parameters

As long as the peripheral is powered-up and still in discoverable mode, a connection should be established immediately. Once a connection is established, the message window will return a “GAP_EstablishLink” event message with a “Status” value of “0x00 (Success)”:

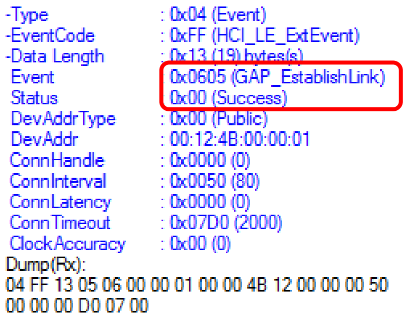


Figure 14 BTool, Link Established

The LCD screen on the peripheral SmartRF05EB should display "Connected". In BTool, you can see your connected peripheral device in the Device Information field, as shown in Figure 15.

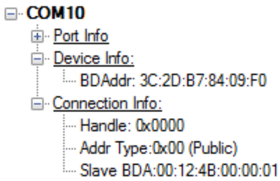


Figure 15 BTool, Device Information

4.5 Using the Simple GATT Profile

The SimpleBLEPeripheral software contains one sample GATT service profile (More information on the SimpleGATTProfile can be found in the Texas Instruments *Bluetooth*® Low Energy Software Developer's Guide [3]). GATT services contain data values known as "characteristic values". All application data that is being sent or received in BLE must be contained within characteristic value. This section details a step-by-step process that demonstrates several processes for reading, writing, discovering, and notifying GATT characteristic values using BTool.

Note that the types (UUIDs) of the five characteristic values (0xFFF1, 0xFFF2, 0xFFF3, 0xFFF4, and 0xFFF5), as well as the simple profile primary service UUID value (0xFFF0), do not conform to any specifications in the *Bluetooth* SIG. They are simply used as a demonstration.

The tables in Figure 16 and Figure 17 below show the SimpleBLEPeripheral complete attribute table, and can be used as a reference. Services are shown in yellow, characteristics are shown in blue, and characteristic values / descriptors are shown in grey. When working with the SimpleBLEPeripheral application, it might be useful to print out the table as a reference.

SimpleBLEPeripheral Application: Complete Attribute Table						
handle (hex)	**handle (dec)**	**Type (hex)**	**Type (#DEFINE)**	**Hex / Text Value (default)**	**GATT Server Permissions**	**Notes**
0x1	1	0x2800	GATT_PRIMARY_SERVICE_UUID	0x1800 (GAP_SERVICE_UUID)	GATT_PERMIT_READ	Start of GAP Service (Mandatory)
0x2	2	0x2803	GATT_CHARACTER_UUID	02 (properties: read only) 03 00 (handle: 0x0003) 00 2A (UUID: 0x2A00)	GATT_PERMIT_READ	Device Name characteristic declaration
0x3	3	0x2A00	GAP_DEVICE_NAME_UUID	"Simple BLE Peripheral"	GATT_PERMIT_READ	Device Name characteristic value
0x4	4	0x2803	GATT_CHARACTER_UUID	02 (properties: read only) 05 00 (handle: 0x0005) 01 2A (UUID: 0x2A01)	GATT_PERMIT_READ	Appearance characteristic declaration
0x5	5	0x2A01	GAP_APPEARANCE_UUID	0x0000	GATT_PERMIT_READ	Appearance characteristic value
0x6	6	0x2803	GATT_CHARACTER_UUID	0A (properties: read/write) 07 00 (handle: 0x0007) 02 2A (UUID: 0x2A02)	GATT_PERMIT_READ	Peripheral Privacy Flag characteristic declaration
0x7	7	0x2A02	GAP_PERI_PRIVACY_FLAG_UUID	0x00 (GAP_PRIVACY_DISABLED)	GATT_PERMIT_READ | GATT_PERMIT_WRITE	Peripheral Privacy Flag characteristic value
0x8	8	0x2803	GATT_CHARACTER_UUID	0A (properties: read/write) 09 00 (handle: 0x0009) 03 2A (UUID: 0x2A03)	GATT_PERMIT_READ	Reconnection address characteristic declaration
0x9	9	0x2A03	GAP_RECONNECT_ADDR_UUID	00:00:00:00:00:00	GATT_PERMIT_READ | GATT_PERMIT_WRITE	Reconnection address characteristic value
0xA	10	0x2803	GATT_CHARACTER_UUID	02 (properties: read only) 0B 00 (handle: 0x000B) 04 2A (UUID: 0x2A04)	GATT_PERMIT_READ	Peripheral Preferred Connection Parameters characteristic declaration
0xB	11	0x2A04	GAP_PERI_CONN_PARAM_UUID	50 00 (100ms preferred min connection interval) A0 00 (200ms preferred max connection interval) 00 00 (0 preferred slave latency) E8 03 (10000ms preferred supervision timeout)	GATT_PERMIT_READ	Peripheral Preferred Connection Parameters characteristic declaration
0xC	12	0x2800	GATT_PRIMARY_SERVICE_UUID	0x1801 (GATT_SERVICE_UUID)	GATT_PERMIT_READ	Start of GATT Service (mandatory)
0xD	13	0x2803	GATT_CHARACTER_UUID	20 (properties: indicate only) 0E 00 (handle: 0x000E) 05 2A (UUID: 0x2A05)	GATT_PERMIT_READ	Service Changed characteristic declaration
0xE	14	0x2A05	GATT_SERVICE_CHANGED_UUID	(null value)	(none)	Service Changed characteristic value
0xF	15	0x2800	GATT_PRIMARY_SERVICE_UUID	0x180A (DEVINFO_SERV_UUID)	GATT_PERMIT_READ	Start of Device Information Service
0x10	16	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 11 00 (handle 0x0011) 23 2A (UUID 0x2A23)	GATT_PERMIT_READ	System ID characteristic declaration
0x11	17	0x2A23	DEVINFO_SYSTEM_ID_UUID	xx xx xx 00 00 xx xx xx (xx's are IEEE address)	GATT_PERMIT_READ	System ID
0x12	18	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 13 00 (handle 0x0013) 24 2A (UUID 0x2A24)	GATT_PERMIT_READ	Model Number String characteristic declaration
0x13	19	0x2A24	DEVINFO_MODEL_NUMBER_UUID	"Model Number"	GATT_PERMIT_READ	Model Number String
0x14	20	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 15 00 (handle 0x0015) 25 2A (UUID 0x2A25)	GATT_PERMIT_READ	Serial Number String characteristic declaration
0x15	21	0x2A25	DEVINFO_SERIAL_NUMBER_UUID	"Serial Number"	GATT_PERMIT_READ	Serial Number String
0x16	22	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 17 00 (handle 0x0017) 26 2A (UUID 0x2A26)	GATT_PERMIT_READ	Firmware Revision String characteristic declaration
0x17	23	0x2A26	DEVINFO_FIRMWARE_REV_UUID	"Firmware Revision"	GATT_PERMIT_READ	Firmware Revision String
0x18	24	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 19 00 (handle 0x0019) 27 2A (UUID 0x2A27)	GATT_PERMIT_READ	Hardware Revision String characteristic declaration
0x19	25	0x2A27	DEVINFO_HARDWARE_REV_UUID	"Hardware Revision"	GATT_PERMIT_READ	Hardware Revision String
0x1A	26	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 1B 00 (handle 0x001B) 28 2A (UUID 0x2A28)	GATT_PERMIT_READ	Software Revision String characteristic declaration
0x1B	27	0x2A28	DEVINFO_SOFTWARE_REV_UUID	"Software Revision"	GATT_PERMIT_READ	Software Revision String
0x1C	28	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 1D 00 (handle 0x001D) 29 2A (UUID 0x2A29)	GATT_PERMIT_READ	Manufacturer Name String characteristic declaration
0x1D	29	0x2A29	DEVINFO_MANUFACTURER_NAME_UUID	"Manufacturer Name"	GATT_PERMIT_READ	Manufacturer Name String
0x1E	30	0x2803	GATT_CHARACTER_UUID	02 (read permissions) 1F 00 (handle 0x001F) 2A 2A (UUID 0x2A2A)	GATT_PERMIT_READ	IEEE 11073-20601 Regulatory Certification Data List characteristic declaration
0x1F	31	0x2A2A	DEVINFO_11073_CERT_DATA_UUID	FE 00 65 78 70 65 72 69 6D 65 6E 74 61 6C	GATT_PERMIT_READ	IEEE 11073-20601 Regulatory Certification Data List

Figure 16, SimpleBLEPeripheral Attribute Table

SimpleBLEPeripheral Application: Complete Attribute Table						
handle (hex)	handle (dec)	Type (hex)	Type (#DEFINE)	Hex / Text Value (default)	GATT Server Permissions	Notes
0x20	32	0x2800	GATT_PRIMARY_SERVICE_UUID	0xFFF0 (SIMPLEPROFILE_SERV_UUID)	GATT_PERMIT_READ	Start of Simple GATT Profile Service
0x20	32	0x2803	GATT_CHARACTER_UUID	0A (properties: read/write) 22 00 (handle: 0x0022) F1 FF (UUID: 0xFFF1)	GATT_PERMIT_READ	Characteristic 1 declaration
0x22	34	0xFFF1	SIMPLEPROFILE_CHAR1_UUID	1 (1 byte)	GATT_PERMIT_READ | GATT_PERMIT_WRITE	Characteristic 1 value
0x23	35	0x2901	GATT_CHAR_USER_DESC_UUID	"Characteristic 1" (17 bytes)	GATT_PERMIT_READ	Characteristic 1 user description
0x24	36	0x2803	GATT_CHARACTER_UUID	02 (properties: read only) 25 00 (handle: 0x0025) F2 FF (UUID: 0xFFF2)	GATT_PERMIT_READ	Characteristic 2 declaration
0x25	37	0xFFF2	SIMPLEPROFILE_CHAR2_UUID	2 (1 byte)	GATT_PERMIT_READ	Characteristic 2 value
0x26	38	0x2901	GATT_CHAR_USER_DESC_UUID	"Characteristic 2" (17 bytes)	GATT_PERMIT_READ	Characteristic 2 user description
0x27	39	0x2803	GATT_CHARACTER_UUID	08 (properties: write only) 28 00 (handle: 0x0028) F3 FF (UUID: 0xFFF3)	GATT_PERMIT_READ	Characteristic 3 declaration
0x28	40	0xFFF3	SIMPLEPROFILE_CHAR3_UUID	3 (1 byte)	GATT_PERMIT_WRITE	Characteristic 3 value
0x29	41	0x2901	GATT_CHAR_USER_DESC_UUID	"Characteristic 3" (17 bytes)	GATT_PERMIT_READ	Characteristic 3 user description
0x2A	42	0x2803	GATT_CHARACTER_UUID	10 (properties: notify only) 2B 00 (handle: 0x002B) F4 FF (UUID: 0xFFF4)	GATT_PERMIT_READ	Characteristic 4 declaration
0x2B	43	0xFFF4	SIMPLEPROFILE_CHAR4_UUID	4 (1 byte)	(none)	Characteristic 4 value
0x2C	44	0x2902	GATT_CLIENT_CHAR_CFG_UUID	00:00 (2 bytes)	GATT_PERMIT_READ | GATT_PERMIT_WRITE	Characteristic 4 configuration
0x2D	45	0x2901	GATT_CHAR_USER_DESC_UUID	"Characteristic 4" (17 bytes)	GATT_PERMIT_READ	Characteristic 4 user description
0x2E	46	0x2803	GATT_CHARACTER_UUID	02 (properties: read only) 2F 00 (handle: 0x002F) F5 FF (UUID: 0xFFF5)	GATT_PERMIT_READ	Characteristic 5 declaration
0x2F	47	0xFFF5	SIMPLEPROFILE_CHAR5_UUID	01:02:03:04:05 (5 bytes)	GATT_PERMIT_ AUTHEN_READ	Characteristic 5 value
0x30	48	0x2901	GATT_CHAR_USER_DESC_UUID	"Characteristic 5" (17 bytes)	GATT_PERMIT_READ	Characteristic 5 user description

Figure 17 SimpleBLEPeripheral Attribute Table

4.5.1 Reading a Characteristic Value by UUID

The first characteristic of the SimpleGATTProfile service has both read and write permissions, and has a UUID of 0xFFF1. The simplest way to read its value is to use the “Read Characteristic by UUID” sub-procedure. To do this, you will first need to click the “Read / Write” tab in BTool. Select the option “Read Using Characteristic UUID” under the “Sub-Procedure” option in the “Characteristic Read” section at the top of the screen. Enter “F1:FF” (note that the LSB is entered first, and the MSB is entered last) in the “Characteristic UUID” box, and click the “Read” button as shown in Figure 18.

An attribute protocol *Read by Type Request* packet gets sent over the air from the central device to the peripheral device, and an attribute protocol *Read by Type Response* packet gets sent back from the peripheral device to the central device. The value “01” is displayed in the “Value” box, and “Success” is displayed in the “Status” box. In addition, the message window will display information on the *Read by Type Response* packet that was received by the central device. The message includes not only the characteristic’s data value, but also the handle of the characteristic value (0x0022 in this case).

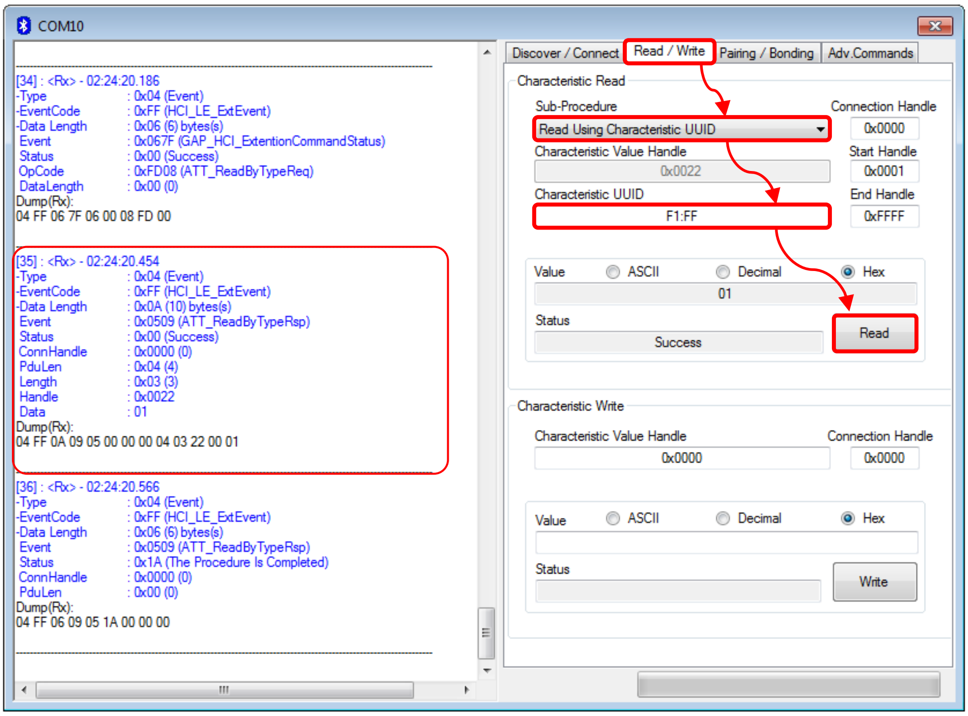


Figure 18 BTool, Read a Characteristic Value by UUID

4.5.2 Writing a Characteristic Value

In the previous section, the handle of the first characteristic in the SimpleGATTProfile was found to be 0x0022. Knowing this, and based on the fact that the characteristic has both read and write permissions, it is possible for us to write a new value. Enter "0x0022" into the "Characteristic Value Handle" box in the "Characteristic Write" section, and enter any 1-byte value in the "Value" section (the format can be set to either "Decimal" or "Hex"). Click the "Write" button as shown in Figure 19.

An attribute protocol *Write Request* packet gets sent over the air from the central device to the peripheral device, and an attribute protocol *Write Response* packet gets sent back from the peripheral device to the central device. The status box will display "Success", indicating that the write was successful.

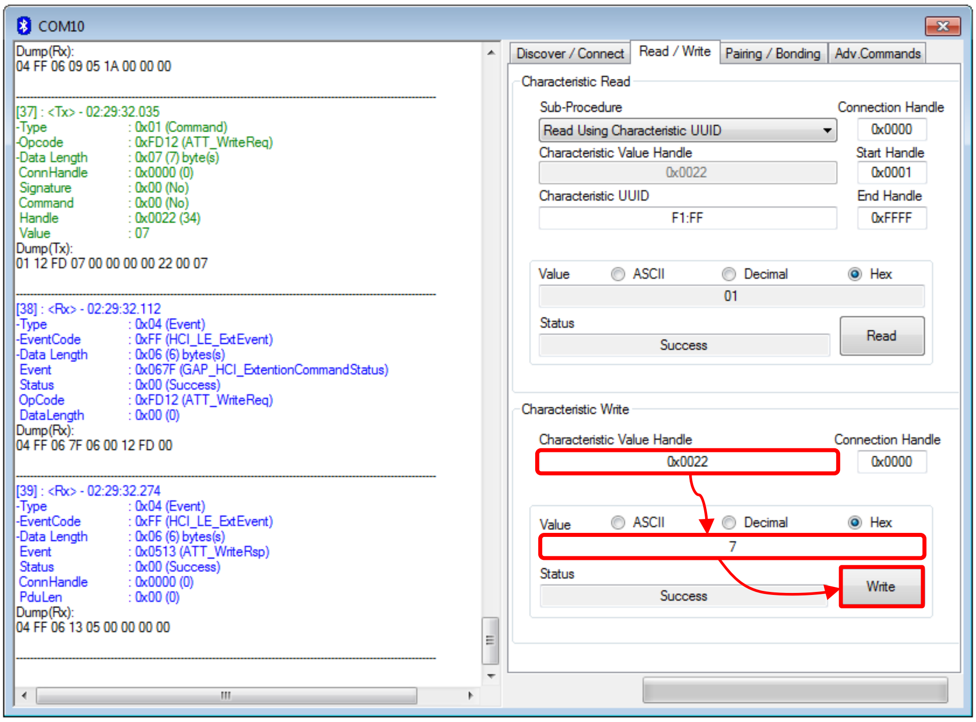


Figure 19 BTool, Write a Characteristic Value

The LCD screen on the peripheral SmartRF05EB should display "Char 1:", and the value written in decimal format.

4.5.3 Reading a Characteristic Value by Handle

After writing a new value to the first characteristic in the profile, we can read the value back to verify the write. This time, instead of reading the value by its UUID, the value will be read by its handle. Select the option "Read Characteristic Value / Descriptor" under the "Sub-Procedure" option in the "Characteristic Read" section. Enter "0x0022" in the "Characteristic Value Handle" box, and click the "Read" button as shown in Figure 20.

An attribute protocol *Read Request* packet gets sent over the air from the central device to the peripheral device, and an attribute protocol *Read Response* packet gets sent back from the peripheral device to the central device. The new value is displayed in the "Value" box, and "Success" is displayed in the "Status" box. This value should match the value that was written in the previous step.

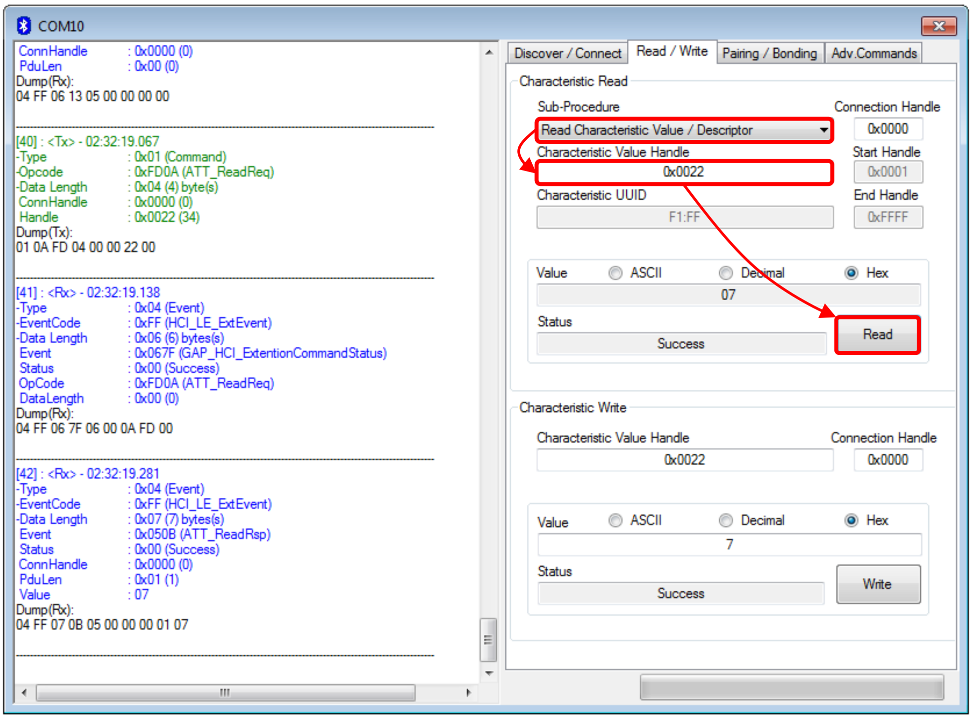


Figure 20 BTool, Read a Characteristic Value by Handle

4.5.4 Discovering a Characteristic by UUID

The next thing to do is to discover a characteristic by its UUID. By doing this, we will not only get the handle of the UUID, but we will also get the properties of the characteristic. The UUID of the second characteristic in the SimpleGATTProfile is 0xFFF2. Select the option "Discover Characteristic by UUID" under the "Sub-Procedure" option in the "Characteristic Read" section at the top of the screen. Enter "F2:FF" in the "Characteristic UUID" box, and click the "Read" button as shown in Figure 21.

A series of attribute protocol *Read by Type Request* packets get sent over the air from the central device to the peripheral device, and for each request an attribute protocol *Read by Type Response* packet gets sent back from the peripheral device to the central device. Essentially, the central device is reading every attribute on the peripheral device with a UUID of 0x2803 (this is the UUID for a characteristic declaration as defined in *Specification of the Bluetooth System* [6]), and checking the "Characteristic Value UUID" portion of each declaration to see if it matches type 0xFFF2. The procedure is complete once every characteristic declaration has been read.

The procedure will find one instance of the characteristic with type 0xFFF2, and display "02 25 00 F2 FF" (the value of the declaration) in the "Value" box, with "Success" displayed in the "Status" box. As per the *Bluetooth* specification, the first byte "02" tells us that the properties of the characteristic are read-only. The second and third bytes "25 00" tell us that the handle of the characteristic value is 0x0025. The fourth and fifth bytes tell the UUID of the characteristic, 0xFFF2.

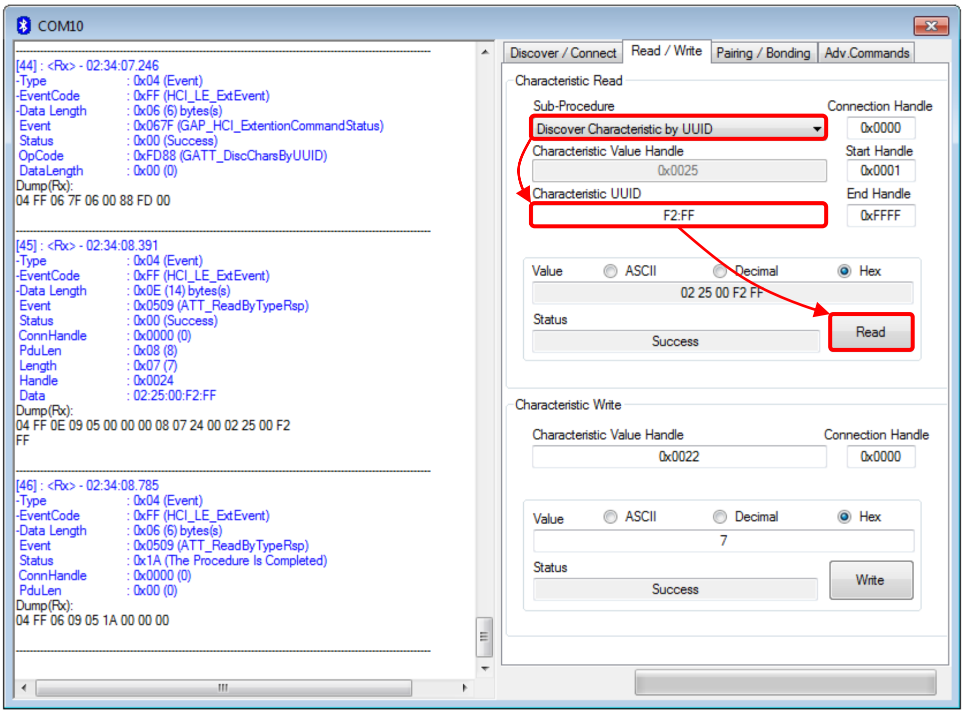


Figure 21 BTool, Discover a Characteristic by UUID

4.5.5 Reading Multiple Characteristic Values

It is also possible to read multiple characteristic values with one request, as long as the handle of each value is known. To read the values of both of the characteristics that we previously read, select the option "Read Multiple Characteristic Values" under the "Sub-Procedure" option in the "Characteristic Read" section at the top of the screen. Enter "0x0022;0x0025" in the "Characteristic Value Handle" box, and click the "Read" button as shown in Figure 22.

An attribute protocol *Read Multiple Request* packet gets sent over the air from the central device to the peripheral device, and an attribute protocol *Read Multiple Response* packet gets sent back from the peripheral device to the central device. The values of the two characteristics are displayed in the "Value" box, and "Success" is displayed in the "Status" box. This first byte should match the value that was written in the previous step, and the second byte should be "02".

One important note about reading multiple characteristic values in a single request is that the response will not parse the separate values. This means that the size of each value being read must be fixed, and must be known by the client. In the example here, this is not an issue since there are only two bytes in the response; however care must be taken when using this command.

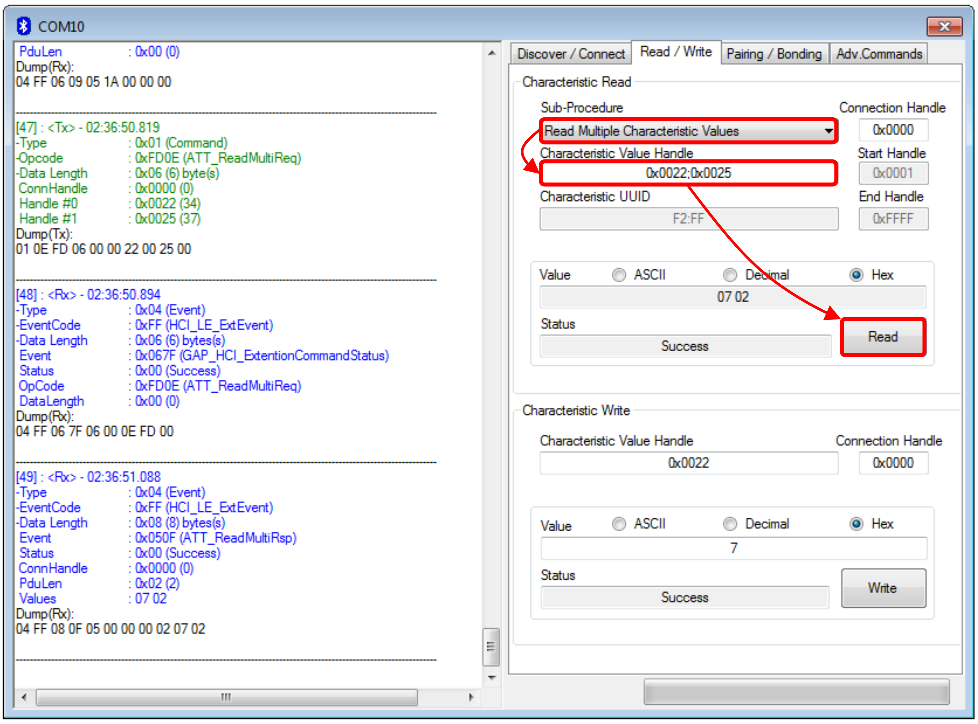


Figure 22 BTool, Read Multiple Characteristic Values

4.5.6 Enabling Notifications

In BLE, it is possible for a GATT server device to “push” characteristic value data out to a client device, without being prompted with a read request. This process is called a “characteristic value notification”. Notifications are useful in that they allow a device in a BLE connection to send out as much or as little data as required at any point in time. In addition, since no request from the client is required, the overhead is reduced and the data is transmitted more efficiently. The SimpleBLEPeripheral software contains an example in which notifications can be demonstrated.

The third characteristic in the SimpleGATTProfile has write-only properties, while the fourth characteristic in the profile has notify-only properties. Every five seconds, the SimpleBLEPeripheral application will take the value of the third characteristic and copy it into the fourth characteristic. Each time the fourth characteristic value gets set by the application, the profile will check to see if notifications are enabled. If they are enabled, the profile will send a notification of the value to the client device.

Before notifications can be enabled, the handle of the fourth characteristic must be found. This can be done by using the “Discover Characteristic by UUID” process (see section 4.5.4), with the UUID value set to “F4:FF”. The procedure will find one instance of the characteristic with type 0xFFF4, and display “10 2B 00 F4 FF” (the value of the declaration) in the “Value” box, with “Success” displayed in the “Status” box. As per the *Bluetooth* specification, the first byte “10” tells us that the properties of the characteristic are notify-only. The second and third bytes “2B 00” tell us that the handle of the characteristic value is 0x002B. The fourth and fifth bytes tell the UUID of the characteristic, 0xFFF4.

In order to enable notifications, the client device must write a value of 0x0001 to the client characteristic configuration descriptor for the particular characteristic. The handle for the client characteristic configuration descriptor immediately follows the characteristic value’s handle. Therefore, a value of 0x0001 must be written to handle 0x002C. Enter “0x002C” into the “Characteristic Value Handle” box in the “Characteristic Write” section, and enter “01:00” in the “Value” section (note that the LSB is entered first, and the MSB is entered last). Click the “Write Value” button. The status box will display “Success”, indicating that the write was successful.

Every five seconds, an attribute protocol *Handle Value Notification* packet gets sent from the peripheral device to the central device. With each notification, the value of the characteristic at handle is displayed in the log window.

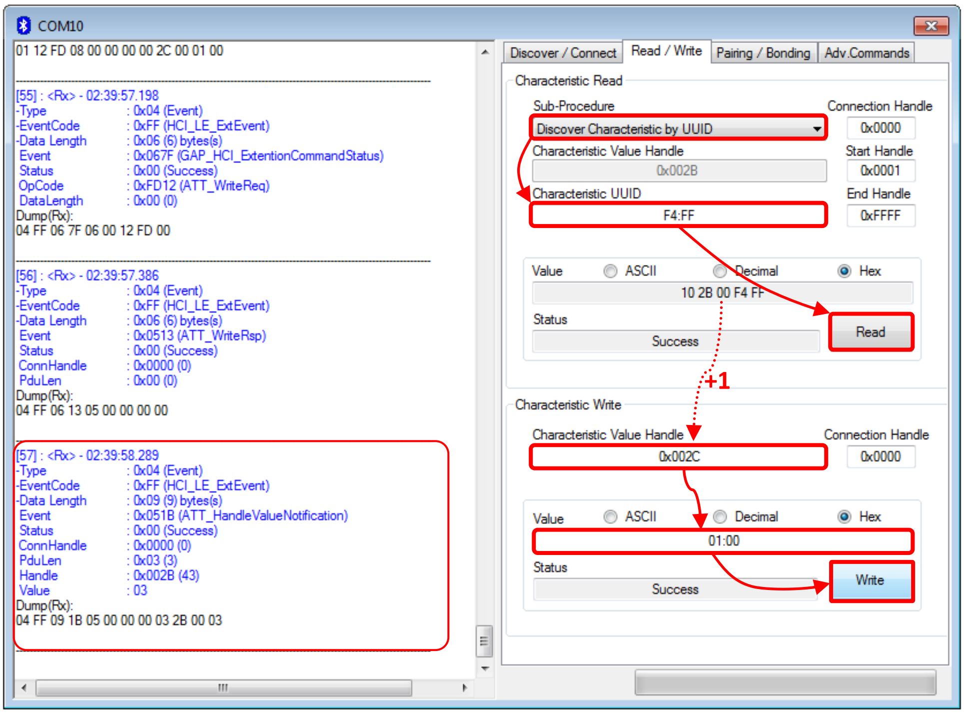


Figure 23 BTool, Enable Notifications

The value should be “03” in each notification, since it is copied from the value of the third characteristic in the profile (which has a default value of 3). The third characteristic has write-only properties, and therefore can be changed. By following the procedure from section 4.5.4, the handle of the third characteristic can be found to be 0x0028. By following the procedure from section 4.5.2, a new value can be written to handle 0x0028. The LCD screen on the peripheral SmartRF05EB should display “Char 3:”, and the value written in decimal format. Once the write is complete, the value of the fourth characteristic will change. This new value is reflected in the incoming notification messages.

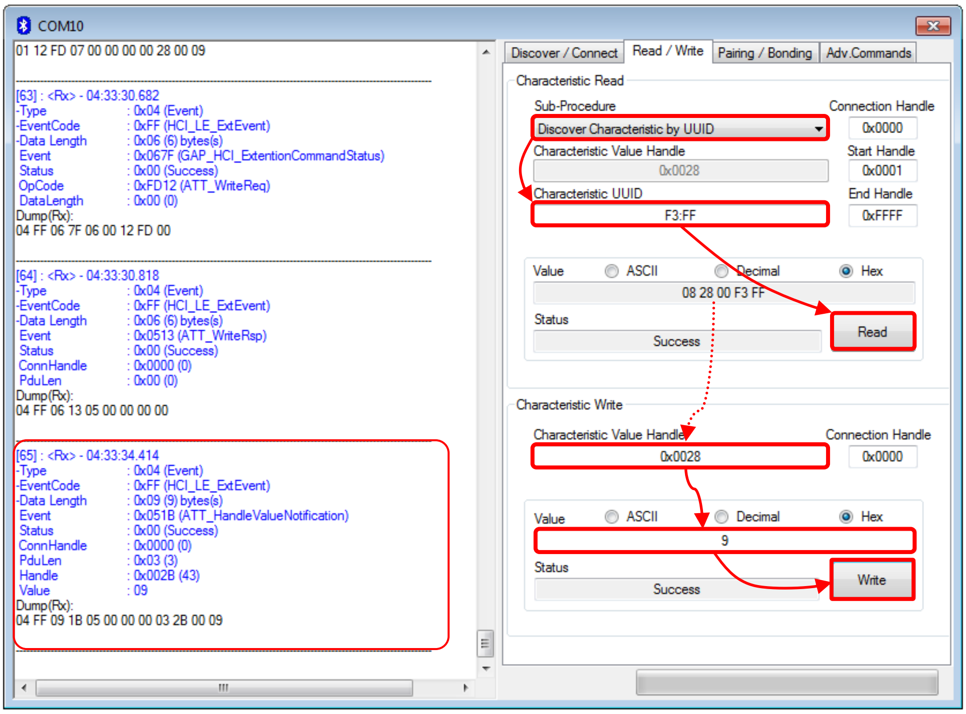


Figure 24 BTool, Write Value to Trigger Notification

It is important to note that the simple GATT profile included with the BLE development kit does not conform to any standard profile specification available from the *Bluetooth* SIG. The profile, including the GATT characteristic definition, the UUID values, and the functional behavior, was developed by Texas Instruments for use with the CC2540DK or CC2542EMK development kit, and is intended as a demonstration of the capabilities of the *Bluetooth* low energy protocol.

4.6 Using BLE Security

BTool also includes the ability to make use of security features in BLE, including encryption, authentication, and bonding.

4.6.1 Encrypting the Connection

The SimpleGATTProfile contains a fifth characteristic with a UUID of 0xFFF5. Like the second characteristic, this characteristic has read-only permissions; however this characteristic can only be read if the link is encrypted.

Using the same discovery process as before with the "Discover Characteristic by UUID" command, it can be determined that the handle of the fifth characteristic value is 0x002F. If you attempt to read this characteristic, however, an error will occur with a status of "INSUFFICIENT_ENCRYPTION".

To encrypt the link, the pairing process must be initiated. Click on the "Pairing / Bonding" tab in BTool. In the "Initiate Pairing" section at the top of the screen, check the boxes labeled "Bonding Enabled" and "Authentication (MITM) Enabled", and click the button "Send Pairing Request", as shown in Figure 25. This will send the request to the peripheral device.

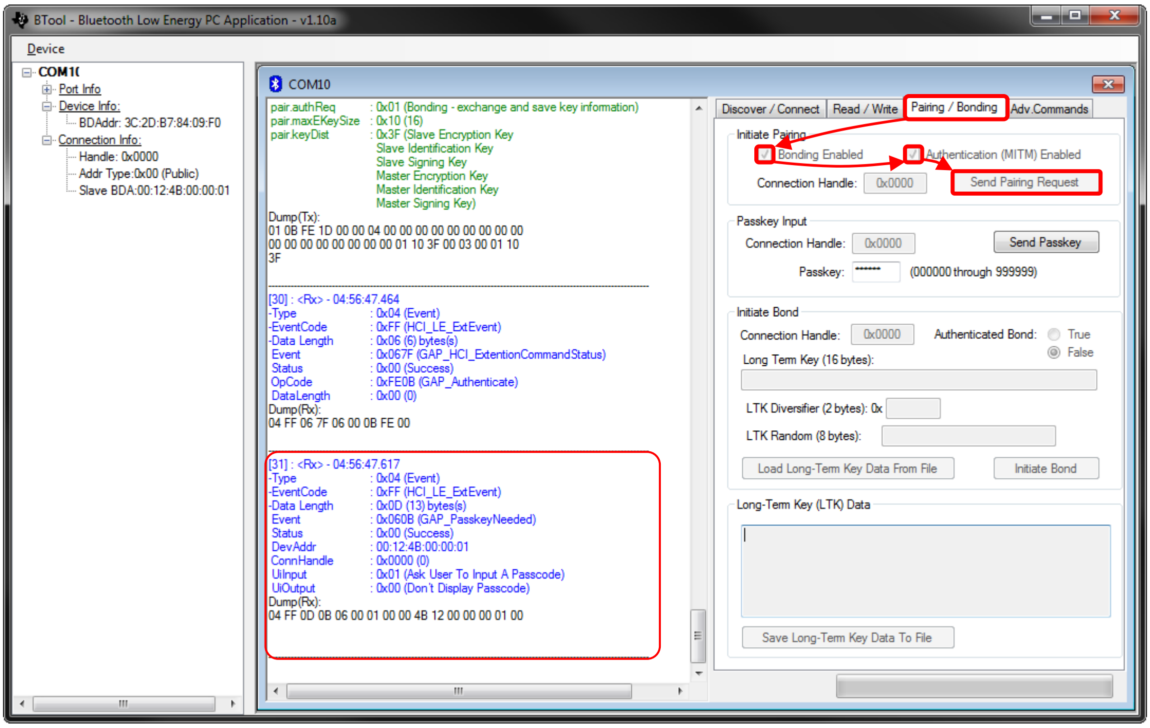


Figure 25 BTool, Send Pairing Request

The peripheral will send a pairing response in return, which will require a six-digit passcode to be entered by the user in order to complete the process. Typically, this passcode is intended to be used by a peripheral device containing a display. By displaying the passkey on the peripheral device and requiring the user to enter it in on the central device's user interface, the link is authenticated, in that it has been verified that the connection has not been hijacked using a man-in-the-middle (MITM) attack.

In the case of the SimpleBLEPeripheral software, a fixed passcode "000000" is used (this value can be modified in the source code). In the box labeled "Passkey" in the "Passkey Input" section, enter the value "000000" and click the "Send Passkey" button, as shown in Figure 26. Note that if you do not send the passkey within 30 seconds after receiving the pairing response message, the pairing process will fail, and you will need to re-send the pairing request.

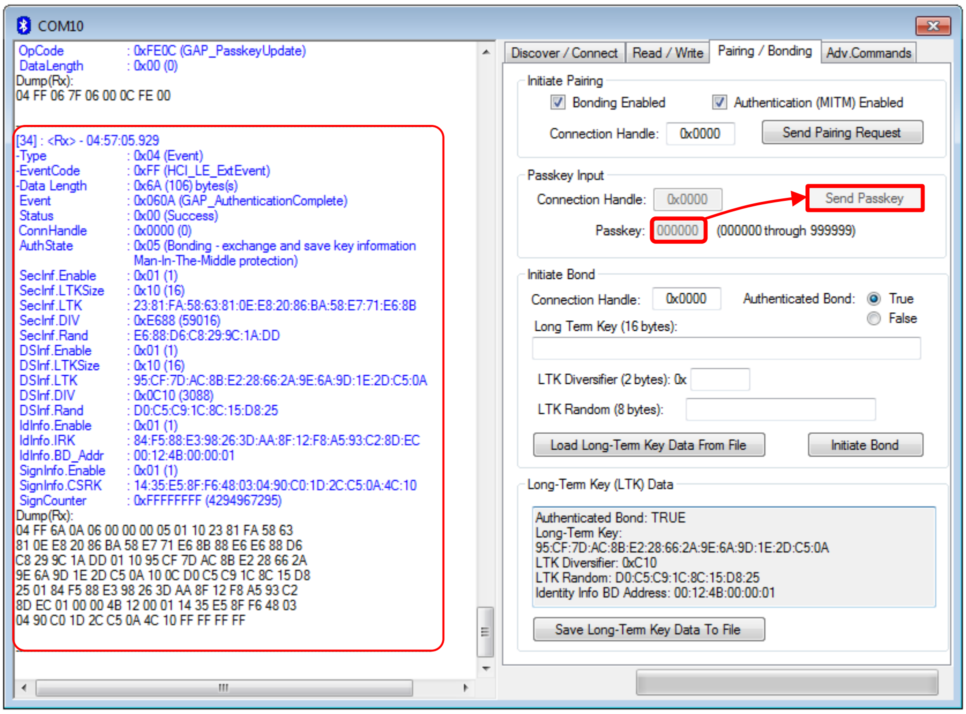


Figure 26 BTool, Send Passkey

When pairing is successfully completed, you will see a "GAP_AuthenticationComplete" event in the log window, with a "Success" status. The BLE connection is now encrypted. You will now be able to read the fifth characteristic value (handle 0x002F) from the peripheral. The five-byte value of the characteristic is "01 02 03 04 05".

4.6.2 Using Bonding and Long-Term Keys

Bonding is a feature in BLE that allows a device, after initial pairing with a peer, to remember specific information about that peer device. In particular, the long-term key data that is generated during the initial pairing process can be stored locally. If the connection is then terminated and the two devices later reconnect, this data can be used to quickly re-initiate encryption without needing to go through the full pairing process and/or use a passkey. In addition, if a client device had enabled notifications of any characteristics on the server device while the two devices were bonded, the server device will remember the setting and the client will not have to re-enable them.

After pairing has been completed with bonding enabled, the "Long-Term Key (LTK) Data" will be populated with some of the data from the "GAP_AuthenticationComplete" event that was generated during the encryption process. This data is required for re-initiating encryption upon reconnect. Click the "Save Long-Term Key Data to File" button to save this information to file, as shown in Figure 27. The data is saved as in a "comma separated value" (CSV) format as simple text, and can be store anywhere on disk. Be sure to note the location that the file is stored.

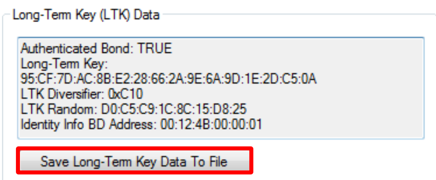


Figure 27 BTool, Save Long-Term Key Data to File

Within the peripheral device, a similar process is going on, in that the SimpleBLEPeripheral software contains a bond manager that is storing the long-term key data that it had generated during encryption. Since the SimpleBLEPeripheral does not have a file system, it is simply storing the data in the nonvolatile memory of the CC2540/41. More information on the bond manager can be found in Texas Instruments *Bluetooth®* Low Energy Software Developer's Guide [3].

With a bond now active, you can enable notifications of a characteristic value and have that setting remembered for later. Note that if notifications were enabled before going through the pairing process, then the setting will not be stored. Therefore, you will need to re-write the value “01:00” to a client characteristic configuration descriptor. For example, write “01:00” to handle 0x002C to enable the periodic notifications, as was done in section 4.5.6. You should now be receiving a notification once every five seconds. Because the devices are paired with bonding enabled, the bond manager in the SimpleBLEPeripheral software will store the client characteristic configuration descriptor data in nonvolatile memory.

To verify that bonding worked, you will need to disconnect and re-connect. Click on the “Discover / Connect” tab and click the “Terminate” button at the bottom of the screen to disconnect from the peripheral device, as shown in Figure 28. The message window will show a “GAP_TerminateLink” event with “Success” status. In addition, the connection information in the upper-left corner of the screen will disappear.

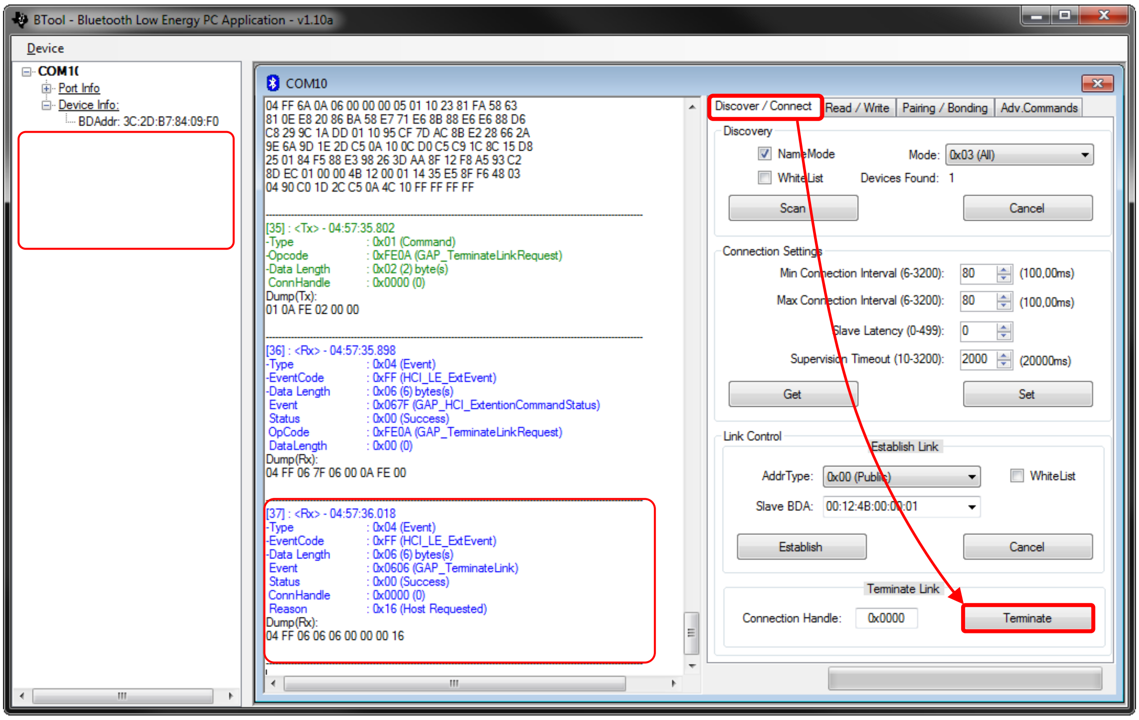


Figure 28 BTool, Terminate Link

At a later time, re-connect with the peripheral device following the procedure in section 4.4.3. Once connected, you will notice that the periodic notifications are no longer enabled. This is because the Simple GATT profile will always reset the value of the client characteristic configuration descriptor back to “00:00” if a connection is terminated or if the device resets.

To re-initiate encryption and re-enable the periodic notifications, return to the “Pairing / Bonding” tab. In the “Initiate Bond” section, click the “Load Long-Term Key Data From File” button, and select the file in which the data was previously stored. The data fields will get automatically populated from the data in the file. Click the “Initiate Bond” button to re-enable encryption, as shown in Figure 29.

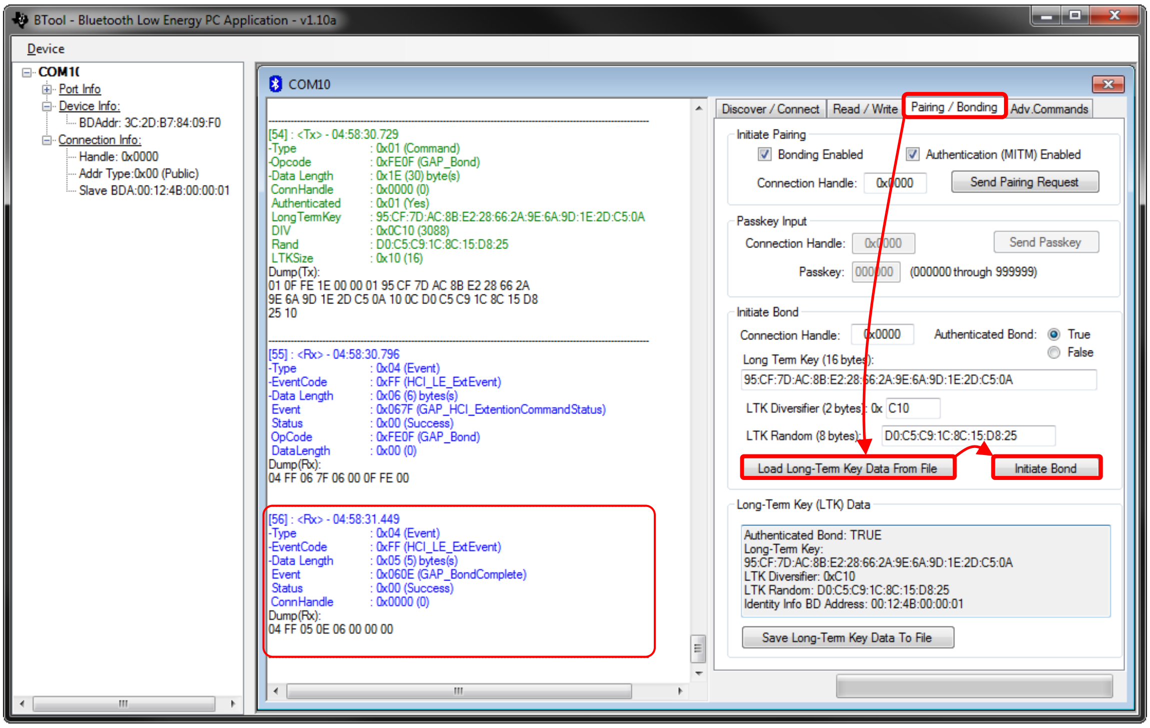


Figure 29 BTool, Re-initiate Encryption

A "GAP_BondComplete" event with "Success" status will be displayed in the message window. This indicates that the link has been re-encrypted, which can be verified by reading the fifth characteristic value in the SimpleGATTProfile at handle 0x002F. You will also now be receiving periodic notifications of the fourth characteristic value, as the client characteristic configuration descriptor value of the characteristic has been restored. Any changes to the client characteristic configuration descriptor value (i.e. turning off notifications) will be saved to nonvolatile memory and remembered for next time that encryption is initiated using the long-term key.

4.7 Additional Sample Applications

In addition to the SimpleBLEPeripheral application, the BLE software development kit includes project and source code files for several additional applications and profiles, including:

- Blood Pressure Sensor- with simulated measurements
- Heart Rate Sensor- with simulated measurements
- Health Thermometer- with simulated measurements

More information on these projects can be found in Texas Instruments BLE Sample Applications Guide [5].

5. Program / Debug the CC254x

The SmartRF05EB allows for debugging using IAR Embedded Workbench, as well as for reading and writing hex files to the CC254x flash memory using the SmartRF Flash Programmer software. SmartRF Flash Programmer also has the capability to change the IEEE address of the CC254x device. The BLE software development kit includes hex files for SimpleBLEPeripheral, SimpleBLECentral, and HostTestRelease (Master Configuration) projects. This section details on using SmartRF Flash Programmer. Information on using IAR Embedded Workbench for debugging can be found in [3]

5.1 Hardware Setup

In order to program or debug the CC254x, the CC254xEM board must be plugged in to the SmartRF05EB. Connect the SmartRF05EB to your PC using a standard USB cable.

5.2 Using SmartRF Flash Programmer Software

Note: the instructions in the section apply to the latest version of SmartRF Flash Programmer (version 1.11.1 Rev. M), which is available at the following URL:

http://focus.ti.com/docs/toolsw/folders/print/flash-programmer.html

To start the application go into your programs by choosing Start > Programs > Texas Instruments > SmartRF Flash Programmer > SmartRF Flash Programmer. The program should open up the following window:

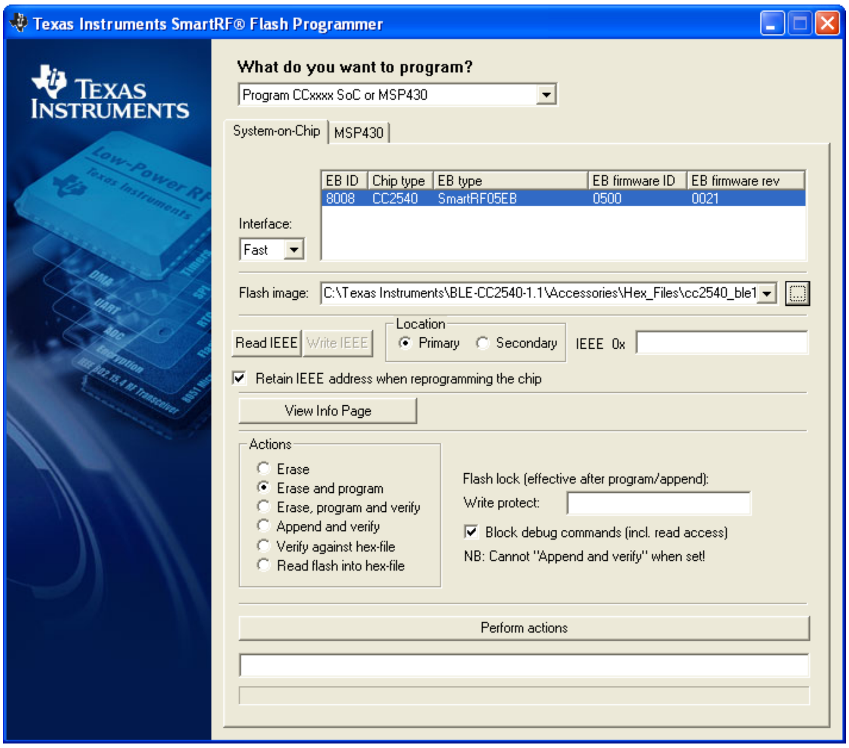


Figure 30

5.2.1 Reading or Writing a Hex File to the CC254x

To read or write a hex file to the CC254x, select the option “Program CCxxxx SoC or MSP430” in the drop box at the top. The connected CC254x should be detected and show up in the list of devices. Under “Flash image” select the desired hex file that you would like to write to the device. If you are reading from the CC254x, under “Flash image” enter the desired path and filename for the hex file. To write to the CC254x, under “Actions” select “Erase, program and verify”. To read from the CC254x, under “Actions” select “Read flash into hex-file”. To begin the read or write, click the button “Perform actions”.

If the action completes successfully, you should see the progress bar at the bottom of the window fill up, and either one of the following two messages, depending on whether a write or a read was performed: “CC254*x* – ID8008: Erase, program and verify OK” or “CC254*x* – ID8008: Flash read OK”.

You may see the following error message:

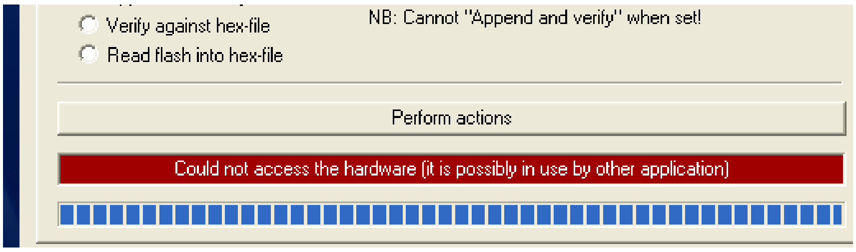


Figure 31

If this comes up, it most likely means that you have IAR open and are debugging. You will need to stop debugging before you can use SmartRF Flash Programmer to communicate with the SmartRF05EB.

5.2.2 Reading or Writing the CC254X Device Address

Every CC254x device comes pre-programmed with a unique 48-bit IEEE address. This is referred to as the device’s “primary address”, and cannot be changed. It is also possible to set a “secondary address” on a device, which will override the primary address upon power-up. SmartRF Flash Programmer can be used to read the primary address, as well as to read or write the secondary address.

To read back the primary address of a device connected to the CC Debugger, select “Primary” under the “Location” option, and click the “Read IEEE” button. The primary device address should appear in the box on the right. Click the “Perform Actions” button at the bottom to perform the read.

To read back the secondary address, select “Secondary” under the “Location” option, and click the “Read IEEE” button. The secondary device address should appear in the box on the right. Click the “Perform Actions” button at the bottom to perform the read.

To set a new secondary address, select “Secondary” under the “Location” option, and enter the desired address in the box on the right. Click the “Perform Actions” button at the bottom to perform the write. If the secondary device is set to “FF FF FF FF FF FF”, the device will use the primary address. If the secondary device is set to anything else, the secondary address will be used.

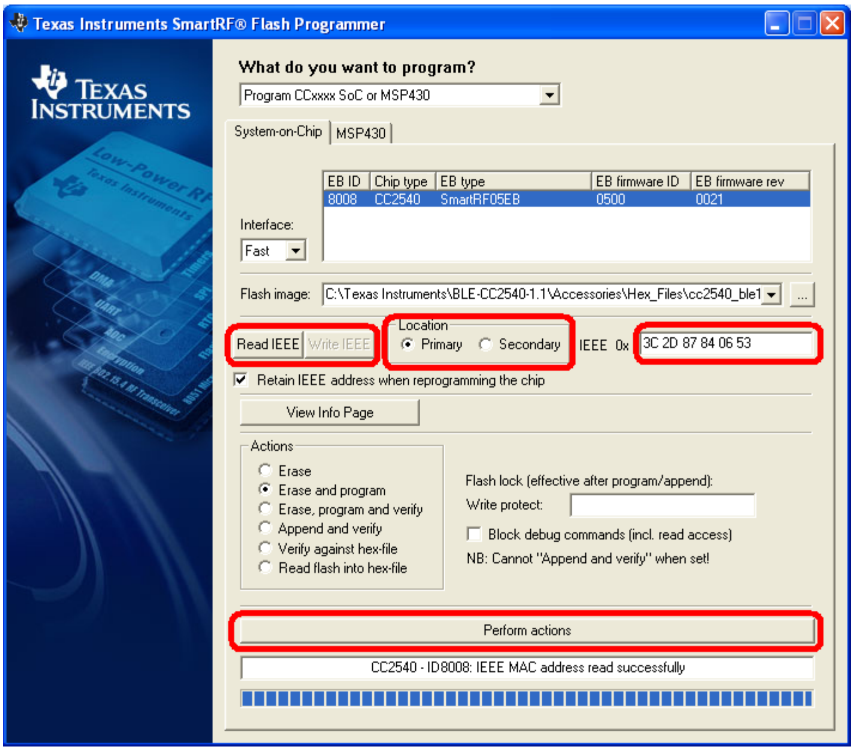


Figure 32

Note that every time you re-program the device using SmartRF Flash Programmer, the secondary address of the device will get set to FF:FF:FF:FF:FF:FF. This can be avoided by selecting the option "Retain IEEE address when reprogramming the chip". A similar situation exists when a device is reprogrammed through IAR Embedded Workbench, in that the secondary address will get set to FF:FF:FF:FF:FF:FF each time. To avoid this, the IAR option "Retain unchanged memory", under the "Debugger" > "Texas Instruments" project option can be selected.

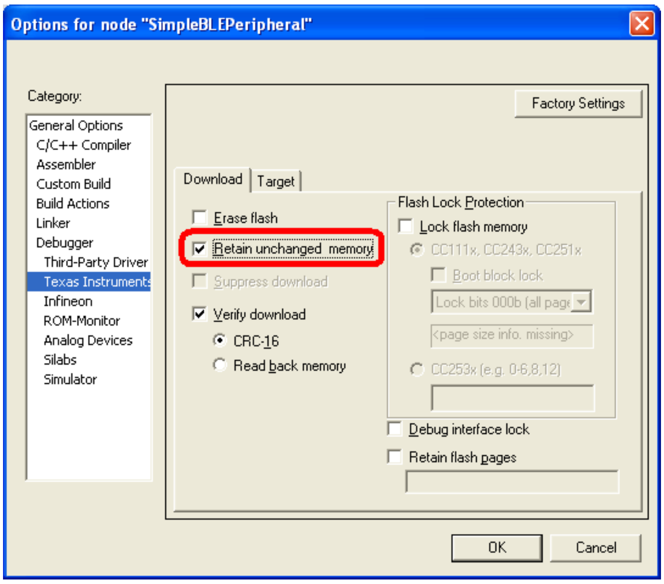


Figure 33

6. SmartRF™ Packet Sniffer

The SmartRF™ Packet Sniffer is a PC software application used to display and store RF packets captured with a listening RF hardware node. Various RF protocols are supported, included BLE. The Packet Sniffer filters and decodes packets and displays them in a convenient way, with options for filtering and storage to a binary file format.

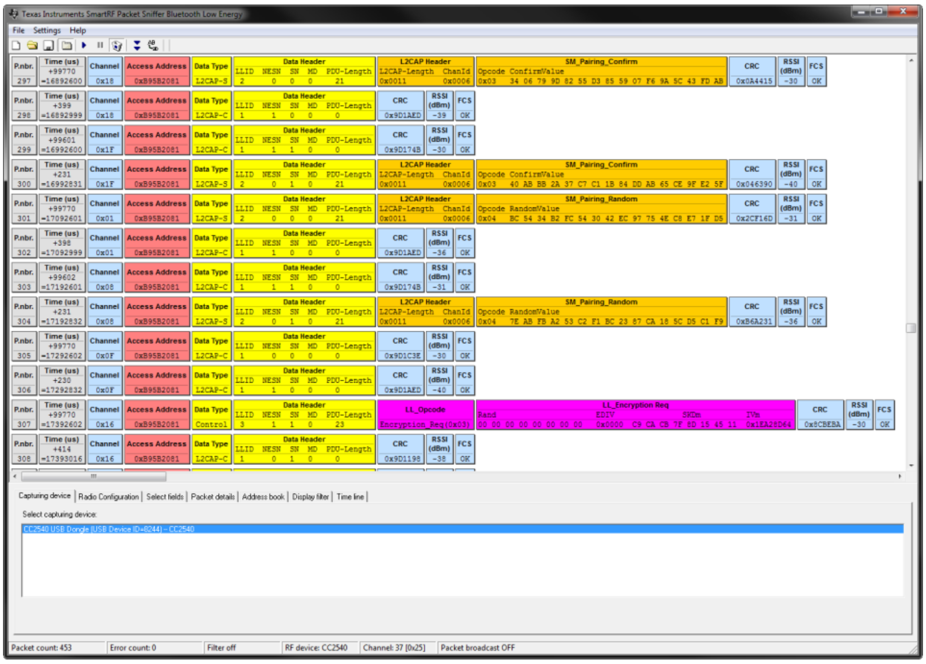


Figure 34

The USB Dongle included with the CC2540DK Development Kit can be used as the listening hardware node, and can be useful when debugging BLE software applications. The SmartRF™ Packet Sniffer software can be downloaded at the following link:

http://focus.ti.com/docs/toolsw/folders/print/packet-sniffer.html

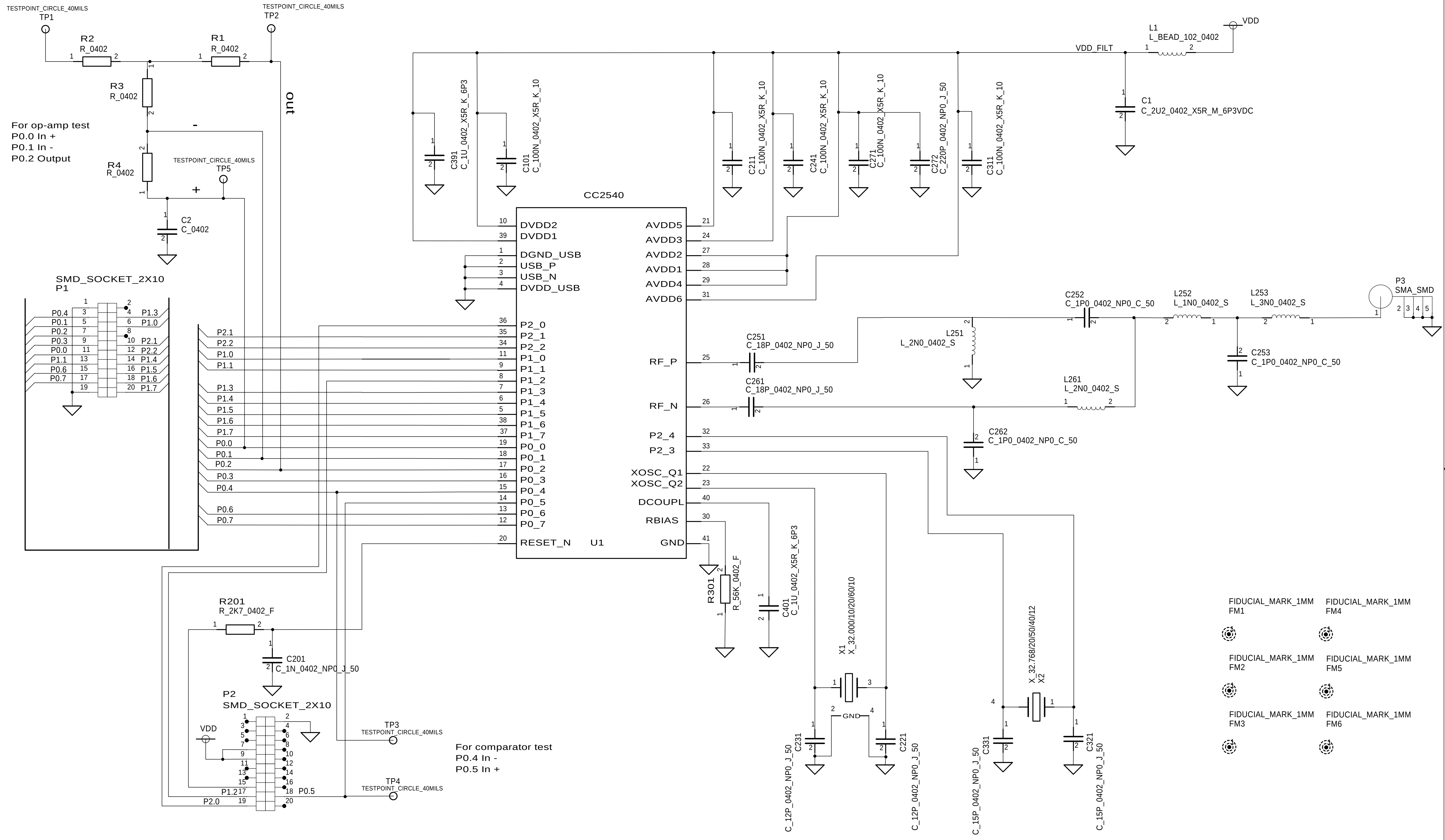
7. General Information

7.1 Document History

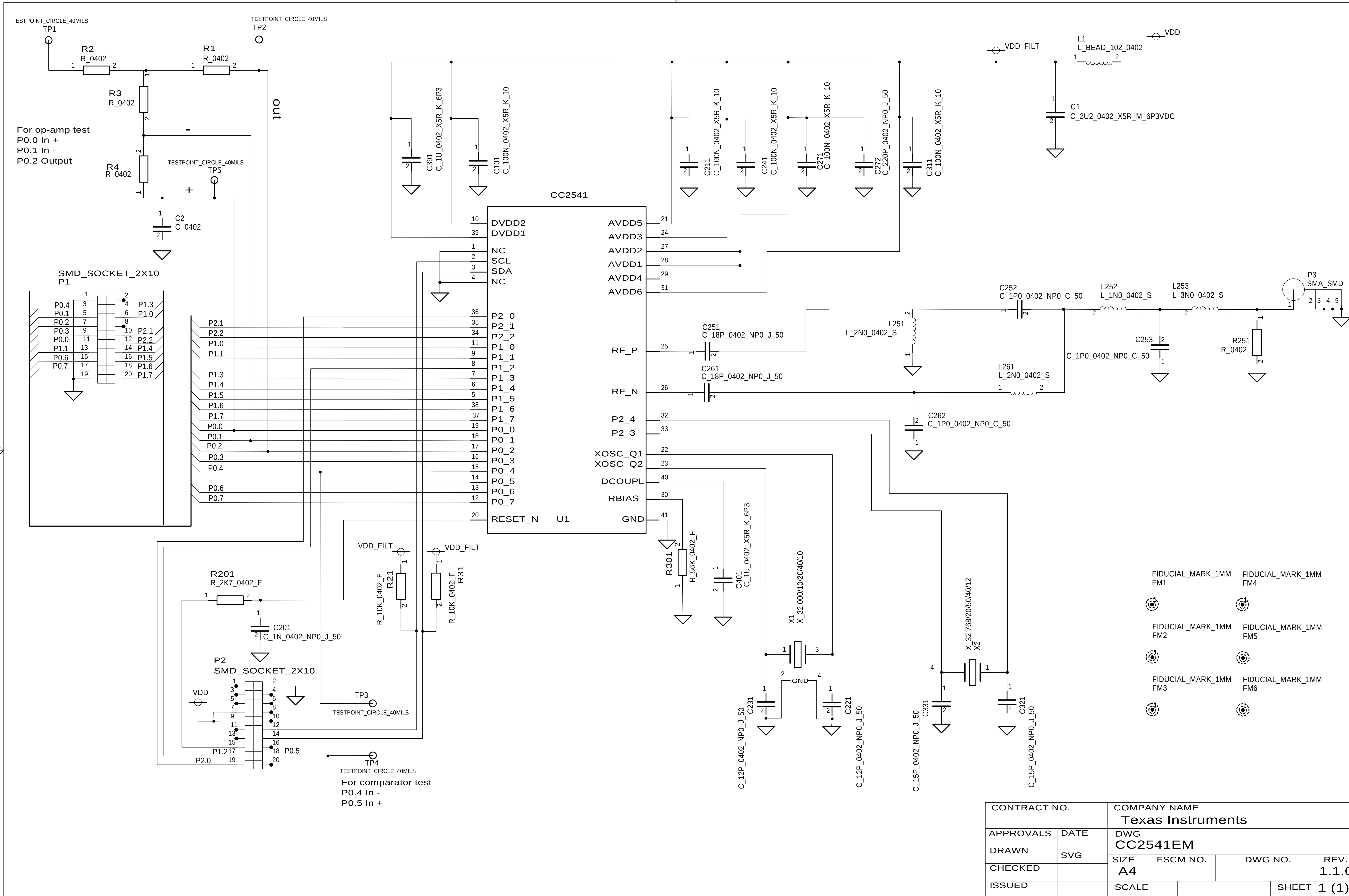
Revision	Date	Description/Changes
SWRU301	2011-08-22	Initial release with BLE software release v1.1
SWRU301A	2012-01-10	Updated with CC2541EMK , for BLE software release v1.1b

Appendix A
Schematics

CC2540/41 Evaluation Module



CONTRACT NO.		COMPANY NAME Texas Instruments			
APPROVALS	DATE	DWG CC2540EM Discrete			
DRAWN	SVG				
CHECKED	NN	SIZE A4	FSCM NO.	DWG NO.	REV. 1.5.1
ISSUED		SCALE		SHEET	1 (1)



CC2541
U1
For op-amp test
P0.0 In +
P0.1 In -
P0.2 Output
For comparator test
P0.4 In -
P0.5 In +
SMD_SOCKET_2X10
P1
P2
TESTPOINT_CIRCLE_40MILS
TP1
TP2
TP3
TP4
TP5
out
VDD_FILT
VDD
L1
L_BEAD_102_0402
C1
C_2U2_0402_X5R_M_6P3VDC
C391
C_1U_0402_X5R_K_6P3
C101
C_100N_0402_X5R_K_10
C211
C241
C271
C272
C_220P_0402_NP0_J_50
C311
DVDD2
DVDD1
NC
SCL
SDA
AVDD5
AVDD3
AVDD2
AVDD1
AVDD4
AVDD6
P2_0
P2_1
P2_2
P1_0
P1_1
P1_2
P1_3
P1_4
P1_5
P1_6
P1_7
P0_0
P0_1
P0_2
P0_3
P0_4
P0_5
P0_6
P0_7
RESET_N
RF_P
RF_N
P2_4
P2_3
XOSC_Q1
XOSC_Q2
DCOUPL
RBIAS
GND
C252
C_1P0_0402_NP0_C_50
L252
L_1N0_0402_S
L253
L_3N0_0402_S
P3
SMA_SMD
C251
C_18P_0402_NP0_J_50
C261
L251
L_2N0_0402_S
L261
C253
R251
R_0402
C262
R1
R2
R3
R4
C2
C_0402
R201
R_2K7_0402_F
C201
C_1N_0402_NP0_J_50
R21
R31
R_10K_0402_F
R301
R_56K_0402_F
C401
C_1U_0402_X5R_K_6P3
X1
X_32.000/10/20/40/10
X2
X_32.768/20/50/40/12
C231
C221
C_12P_0402_NP0_J_50
C331
C321
C_15P_0402_NP0_J_50
FIDUCIAL_MARK_1MM
FM1
FM2
FM3
FM4
FM5
FM6
CONTRACT NO.
COMPANY NAME
Texas Instruments
APPROVALS
DATE
DWG
CC2541EM
DRAWN
SVG
CHECKED
SIZE
A4
FSCM NO.
DWG NO.
REV.
1.1.0
ISSUED
SCALE
SHEET
1 (1)

EVALUATION BOARD/KIT/MODULE (EVM) ADDITIONAL TERMS

Texas Instruments (TI) provides the enclosed Evaluation Board/Kit/Module (EVM) under the following conditions:

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies TI from all claims arising from the handling or use of the goods.

Should this evaluation board/kit not meet the specifications indicated in the User’s Guide, the board/ kit may be returned within 30 days from the date of delivery for a full refund. THE FOREGOING LIMITED WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY SELLER TO BUYER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages. For additional information on TI's environmental and/or safety programs, please visit www.ti.com/esh or contact TI.

No license is granted under any patent right or other intellectual property right of TI covering or relating to any machine, process, or combination in which such TI products or services might be or are used. TI currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

REGULATORY COMPLIANCE INFORMATION

As noted in the EVM User’s Guide and/or EVM itself, this EVM and/or accompanying hardware may or may not be subject to the Federal Communications Commission (FCC) and Industry Canada (IC) rules.

For EVMs **not** subject to the above rules, this evaluation board/kit/module is intended for use for ENGINEERING DEVELOPMENT, DEMONSTRATION OR EVALUATION PURPOSES ONLY and is not considered by TI to be a finished end product fit for general consumer use. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC or ICES-003 rules, which are designed to provide reasonable protection against radio frequency interference. Operation of the equipment may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

General Statement for EVMs including a radio

User Power/Frequency Use Obligations: This radio is intended for development/professional use only in legally allocated frequency and power limits. Any use of radio frequencies and/or power availability of this EVM and its development application(s) must comply with local laws governing radio spectrum allocation and power limits for this evaluation module. It is the user’s sole responsibility to only operate this radio in legally acceptable frequency space and within legally mandated power limitations. Any exceptions to this is strictly prohibited and unauthorized by Texas Instruments unless user has obtained appropriate experimental/development licenses from local regulatory authorities, which is responsibility of user including its acceptable authorization.

For EVMs annotated as FCC – FEDERAL COMMUNICATIONS COMMISSION Part 15 Compliant

Caution

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

FCC Interference Statement for Class A EVM devices

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

REGULATORY COMPLIANCE INFORMATION (continued)

FCC Interference Statement for Class B EVM devices

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

For EVMs annotated as IC – INDUSTRY CANADA Compliant

This Class A or B digital apparatus complies with Canadian ICES-003.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Concerning EVMs including radio transmitters

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Concerning EVMs including detachable antennas

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.

This radio transmitter has been approved by Industry Canada to operate with the antenna types listed in the user guide with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Cet appareil numérique de la classe A ou B est conforme à la norme NMB-003 du Canada.

Les changements ou les modifications pas expressément approuvés par la partie responsable de la conformité ont pu vider l'autorité de l'utilisateur pour actionner l'équipement.

Concernant les EVMs avec appareils radio

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

Concernant les EVMs avec antennes détachables

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une communication satisfaisante.

Le présent émetteur radio a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés dans le manuel d'usage et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

【Important Notice for Users of this Product in Japan】

This development kit is NOT certified as Confirming to Technical Regulations of Radio Law of Japan

If you use this product in Japan, you are required by Radio Law of Japan to follow the instructions below with respect to this product:

1. Use this product in a shielded room or any other test facility as defined in the notification #173 issued by Ministry of Internal Affairs and Communications on March 28, 2006, based on Sub-section 1.1 of Article 6 of the Ministry's Rule for Enforcement of Radio Law of Japan,
2. Use this product only after you obtained the license of Test Radio Station as provided in Radio Law of Japan with respect to this product, or
3. Use of this product only after you obtained the Technical Regulations Conformity Certification as provided in Radio Law of Japan with respect to this product. Also, please do not transfer this product, unless you give the same notice above to the transferee. Please note that if you could not follow the instructions above, you will be subject to penalties of Radio Law of Japan.

Texas Instruments Japan Limited
(address) 24-1, Nishi-Shinjuku 6 chome, Shinjukku-ku, Tokyo, Japan

http://www.tij.co.jp

【ご使用にあたっての注】

本開発キットは技術基準適合証明を受けておりません。

本製品のご使用に際しては、電波法遵守のため、以下のいずれかの措置を取っていただく必要がありますのでご注意ください。

1. 電波法施行規則第6条第1項第1号に基づく平成18年3月28日総務省告示第173号で定められた電波暗室等の試験設備でご使用いただく。
2. 実験局の免許を取得後ご使用いただく。
3. 技術基準適合証明を取得後ご使用いただく。

なお、本製品は、上記の「ご使用にあたっての注意」を譲渡先、移転先に通知しない限り、譲渡、移転できないものとします。

上記を遵守頂けない場合は、電波法の罰則が適用される可能性があることをご留意ください。

日本テキサス・インスツルメンツ株式会社
東京都新宿区西新宿６丁目２４番１号
西新宿三井ビル
http://www.tij.co.jp

EVALUATION BOARD/KIT/MODULE (EVM) WARNINGS, RESTRICTIONS AND DISCLAIMERS

For Feasibility Evaluation Only, in Laboratory/Development Environments. Unless otherwise indicated, this EVM is not a finished electrical equipment and not intended for consumer use. It is intended solely for use for preliminary feasibility evaluation in laboratory/development environments by technically qualified electronics experts who are familiar with the dangers and application risks associated with handling electrical mechanical components, systems and subsystems. It should not be used as all or part of a finished end product.

Your Sole Responsibility and Risk. You acknowledge, represent and agree that:

1. You have unique knowledge concerning Federal, State and local regulatory requirements (including but not limited to Food and Drug Administration regulations, if applicable) which relate to your products and which relate to your use (and/or that of your employees, affiliates, contractors or designees) of the EVM for evaluation, testing and other purposes.
2. You have full and exclusive responsibility to assure the safety and compliance of your products with all such laws and other applicable regulatory requirements, and also to assure the safety of any activities to be conducted by you and/or your employees, affiliates, contractors or designees, using the EVM. Further, you are responsible to assure that any interfaces (electronic and/or mechanical) between the EVM and any human body are designed with suitable isolation and means to safely limit accessible leakage currents to minimize the risk of electrical shock hazard.
3. You will employ reasonable safeguards to ensure that your use of the EVM will not result in any property damage, injury or death, even if the EVM should fail to perform as described or expected.
4. You will take care of proper disposal and recycling of the EVM’s electronic components and packing materials.

Certain Instructions. It is important to operate this EVM within TI’s recommended specifications and environmental considerations per the user guidelines. Exceeding the specified EVM ratings (including but not limited to input and output voltage, current, power, and environmental ranges) may cause property damage, personal injury or death. If there are questions concerning these ratings please contact a TI field representative prior to connecting interface electronics including input power and intended loads. Any loads applied outside of the specified output range may result in unintended and/or inaccurate operation and/or possible permanent damage to the EVM and/or interface electronics. Please consult the EVM User's Guide prior to connecting any load to the EVM output. If there is uncertainty as to the load specification, please contact a TI field representative. During normal operation, some circuit components may have case temperatures greater than 60oC as long as the input and output are maintained at a normal ambient operating temperature. These components include but are not limited to linear regulators, switching transistors, pass transistors, and current sense resistors which can be identified using the EVM schematic located in the EVM User's Guide. When placing measurement probes near these devices during normal operation, please be aware that these devices may be very warm to the touch. As with all electronic evaluation tools, only qualified personnel knowledgeable in electronic measurement and diagnostics normally found in development environments should use these EVMs.

Agreement to Defend, Indemnify and Hold Harmless. You agree to defend, indemnify and hold TI, its licensors and their representatives harmless from and against any and all claims, damages, losses, expenses, costs and liabilities (collectively, "Claims") arising out of or in connection with any use of the EVM that is not in accordance with the terms of the agreement. This obligation shall apply whether Claims arise under law of tort or contract or any other legal theory, and even if the EVM fails to perform as described or expected.

Safety-Critical or Life-Critical Applications. If you intend to evaluate the components for possible use in safety critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, such as devices which are classified as FDA Class III or similar classification, then you must specifically notify TI of such intent and enter into a separate Assurance and Indemnity Agreement.

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