C5545 BoosterPack Software

User's Guide

Literature Number: SPRUI92
October 2016

Contents

1 Introduction 4
- 1.1 Hardware Diagnostics 4
- 1.2 Out of Box Demo 4

2 Diagnostic Test Software 5
- 2.1 Overview 5
- 2.2 Folder Structure 6
- 2.3 Build Setup 7
- 2.4 Setup and Execution 8

3 Out of Box Demo 8
- 3.1 Out of Box Demo Features 9
- 3.2 Audio Demo 9
- 3.3 BLE Firmware 13
- 3.4 C5545 Boost Android App 15
- 3.5 BLE Firmware Upgrade 17
- 3.6 Setup and Execution 18

4 References 20

Appendix A Diagnostic Test Procedure 21
- A.1 Initial Steps 21

Appendix B CCS Target Configurations 45
- B.1 Connecting to C5545 DSP From CCS 45
- B.2 Connecting to CC2650 BLE From CCS 45
- B.3 Connecting to MSP432 LP From CCS 46

List of Figures

1 Folder Structure for C5545 BoosterPack Software Package 5
2 Folder Structure for the Diagnostic Software Package 6
3 Project Explorer Window 7
4 High-Level Blocks of the Demo 8
5 Software Components Involved in Audio Demo 9
6 Folder Structure of Audio Demo 11
7 CCS Workspace After Importing Audio Demo Projects 12
8 Folder Structure of the BLE Firmware 13
9 Project Explorer Window 14
10 Equalizer App Home Screen 15
11 BLE Scanner Screen 15
12 Scanner Screen After Scanning 15
13 Equalizer With Playback Stopped 15
14 Equalizer While Playing 16
15 Equalizer With Playback Paused 16
16 Folder Structure of Equalizer Android App 17
17 C5545 Boost Icon 19
18 Serial Console Output on Host PC 44

BoosterPack, Code Composer Studio are trademarks of Texas Instruments.
Bluetooth is a trademark of Bluetooth SIG, Inc.

User's Guide
SPRUI92–October 2016

C5545 BoosterPack Software

This document outlines the details of software modules that are provided with the C5545 BoosterPack™ kit.

1 Introduction

C5545 BoosterPack is TI’s C5545 DSP-based hardware platform for quick evaluation and software development on the C5545 digital signal processor (DSP). The C5545 BoosterPack also includes the CC2650 BLE chip for providing *Bluetooth*™ connectivity to the board. For more details on the hardware features, see the *TMS320C5545 BoosterPack Hardware User's Guide* (SPRUI90).

The C5545 BoosterPack comes with a set of software modules that enable quick evaluation of the hardware features and serves as reference for software development. Below is the list of software modules that are provided along with BoosterPack kit. Each of the software modules are described in more detail in the later sections of this document.

1.1 Hardware Diagnostics

Hardware diagnostics provide test software for verifying hardware interfaces on C5545 BoosterPack.

1.2 Out of Box Demo

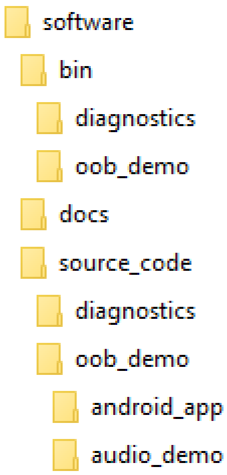
The out of box demo provides various software modules to demonstrate audio and BLE capabilities of C5545 BoosterPack. It includes an audio playback demo, BLE firmware and C5545 Boost Android app.

The audio playback demo runs on C5545 DSP of BoosterPack and supports wave file playback or audio streaming from LINE IN to HEADPHONE.

BLE firmware runs on the CC2650 BLE chip of BoosterPack, which acts as an interface between audio demo and Android app.

C5545 BoosterPack Android app provides a graphical user interface (GUI) for controlling equalizer parameters of the audio demo running on BoosterPack. It communicates with the BoosterPack through the BLE chip.

Figure 1 shows the folder structure for C5545 BoosterPack software package.



```
software
  bin
    diagnostics
    oob_demo
  docs
  source_code
    diagnostics
    oob_demo
      android_app
      audio_demo
```

Figure 1. Folder Structure for C5545 BoosterPack Software Package

- bin – contains the binaries for diagnostics and the out of box demo. C5545 BoosterPack is shipped with the out of box demo to boot from the SD card by default. Boot image for diagnostics can be placed in the root folder of SD card to boot the diagnostics.
- docs - contains the documents related to C5545 BoosterPack software
- source_code folder contains the source code for diagnostics and the out of box demo. This includes source code for the audio demo running on the BoosterPack and the C5545 BoosterPack equalizer app for the Android device.

2 Diagnostic Test Software

2.1 Overview

TI's C55xx Chip Support Library (CSL) is used for hardware access and configuration in the diagnostic test software. Diagnostic tests include three different types of interface tests, as described below.

2.1.1 DSP Interface Tests

DSP interface tests provide the test cases for verifying on-board interfaces connected to C5545 DSP.

- SPI Flash Test
- LED Test
- OLED Display Test
- Audio Tests
- Current Monitor Test
- RTC Test
- Push Button Test
- USB Test

2.1.2 Launchpad Interface Tests

C5545 BoosterPack provides hardware connectivity to interface with MSP432 LaunchPad. LaunchPad interface tests provide the test cases to verify the interfaces between C5545 DSP and MSP432/CC3200 LaunchPad.

Launchpad interface tests include:

- C5545 BP to MSP432 LP GPIO Test
- C5545 BP to CC3200 LP GPIO Test
- MSP-DSP inter-integrated circuit (I2C) Tests

For details on interfacing C5545 BoosterPack with MSP432 LaunchPad, see the *TMS320C5545 BoosterPack Hardware User's Guide* (SPRUI90).

2.1.3 BLE Interface tests

BLE interface tests include the test cases to verify the interfaces connected to the CC2650 BLE module on the BoosterPack.

BLE interface tests include:

- BLE to DSP general-purpose input/output (GPIO) Test
- BLE to MSP432 LP GPIO Test
- BLE universal asynchronous receiver/transmitter (UART) Test

2.2 Folder Structure

Figure 2 shows the folder structure of diagnostic software package.

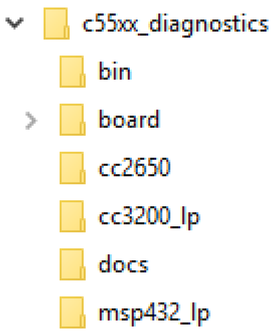


Figure 2. Folder Structure for the Diagnostic Software Package

- board – contains diagnostic source code for the tests running on C5545 DSP and MSP432 LaunchPad
- bin – contains C5545 DSP SD boot image for diagnostic tests
- cc2650 – contains test binaries for CC2560 BLE interface tests
- cc3200_lp – contains test binaries for CC3200 LaunchPad interface tests
- docs – contains diagnostic software documentation
- msp432_lp - contains test binaries for MSP432 LaunchPad interface tests

2.3 Build Setup

2.3.1 Prerequisites

Install and setup the following software tools for building the diagnostic software:

- CCSv6
- C55 Code gen tool – v4.4.1
- C55x CSL library
  - CSL library from TI website can be downloaded here

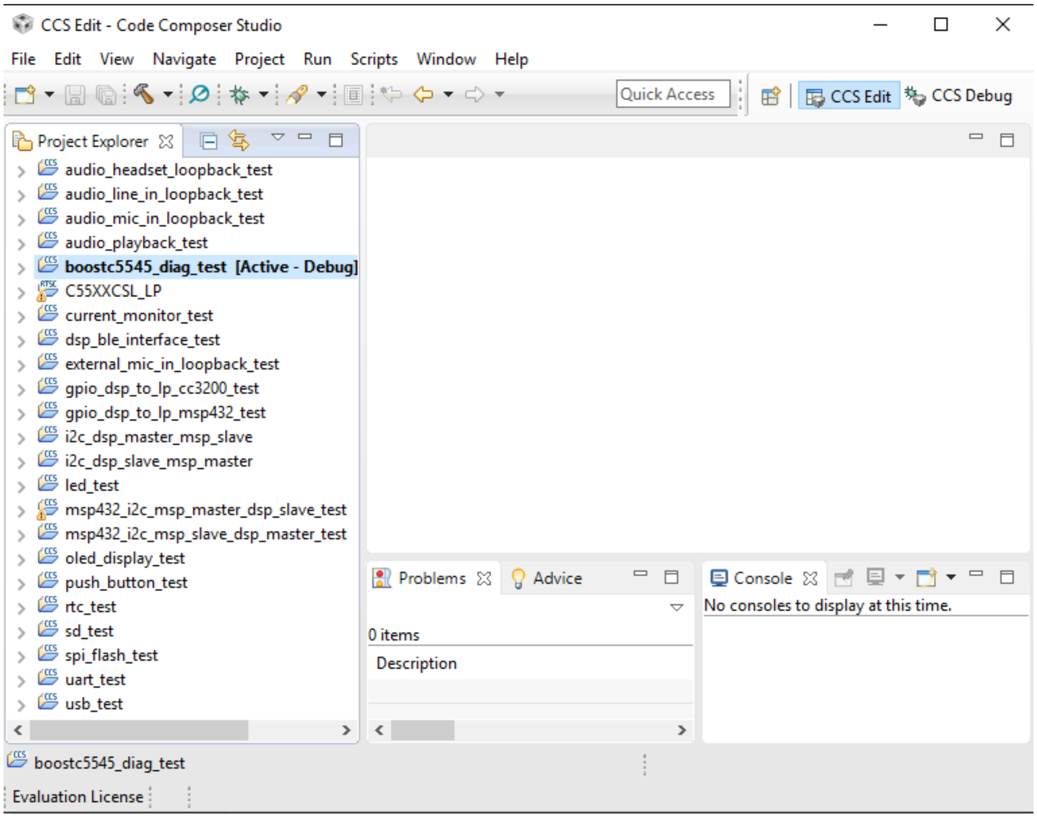
Setting up and getting started with Code Composer Studio™ (CCS) is beyond the scope of this document. For more information, see the getting started guides on the CCS download page. TI assumes that the user understands how to import, build, and run CCS projects.

2.3.2 Build Procedure

The diagnostic software package provides the CCS project setup for individual interface tests and an integrated test that can boot from the SD card and execute tests based on user selection from the menu. This section describes the procedure to build both type of projects.

Use the following steps to build the diagnostic test binaries:

1. Open CCS IDE and select the menu ‘Project → Import CCS Projects’.
2. In the new Window, click on the ‘Browse’ button, select the diagnostic root folder and click ‘OK’.
3. Select all the projects and click on the ‘Finish’ button. Projects should appear in the ‘Project Explorer’ window as shown in Figure 3.


CCS Edit - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Quick Access
CCS Edit
CCS Debug
Project Explorer
Problems
Advice
Console

audio_headset_loopback_test
audio_line_in_loopback_test
audio_mic_in_loopback_test
audio_playback_test
boostc5545_diag_test [Active - Debug]
C55XXCSL_LP
current_monitor_test
dsp_ble_interface_test
external_mic_in_loopback_test
gpio_dsp_to_lp_cc3200_test
gpio_dsp_to_lp_msp432_test
i2c_dsp_master_msp_slave
i2c_dsp_slave_msp_master
led_test
msp432_i2c_msp_master_dsp_slave_test
msp432_i2c_msp_slave_dsp_master_test
oled_display_test
push_button_test
rtc_test
sd_test
spi_flash_test
uart_test
usb_test

0 items
Description

No consoles to display at this time.

boostc5545_diag_test
Evaluation License

Figure 3. Project Explorer Window

4. Navigate to the C55xx CSL source folder (Default path - C:\ti\c55_lp) and import the following projects:
   (a) C55XXCSL_LP
   (b) atafs_bios_drv_lib
5. Navigate to C:\ti\c55_lp \ c55_csl_x.xx\inc\csl_general.h and change the following macro definitions:

```
#define CHIP_C5517 to //#define CHIP_C5517
#define CHIP_C5505_C5515 to //#define CHIP_C5505_C5515
#define CHIP_C5535 to //#define CHIP_C5535
```

6. Build C55XXCSL_LP and atafs_bios_drv_lib in both Debug and Release modes.

2.3.2.1 Building Diagnostic Test Binary

1. Right click on the project ‘boostc5545_diag_test’ and set ‘Build Configurations’ to ‘SD_BOOT’.
2. Right click on the project ‘boostc5545_diag_test’ and select the Build Project option.
3. bootimg.bin will be created under c55xx_diagnostics/bin folder.

2.3.2.2 Building Standalone Projects

1. Right click on any standalone test project and select the Build Project option.
2. Successful execution of build creates an .out that can be loaded and executed on BoosterPack

2.3.2.3 Building MSP432 Tests

1. Right click on the MSP432 diagnostic test projects (msp432_i2c_msp_master_dsp_slave_test/ msp432_i2c_msp_slave_dsp_master_test) and select the *Build Project* option.
2. Successful execution of build creates an .out that can be loaded and executed on MSP432 LaunchPad

2.4 Setup and Execution

For details on the test setup and procedure to run the diagnostics on C5545 BoosterPack, see Appendix A.

3 Out of Box Demo

The C5545 BoosterPack out of box demo is targeted to showcase the audio processing and wireless connectivity capabilities of the C5545 BoosterPack. High-level blocks of the demo are shown in Figure 4.

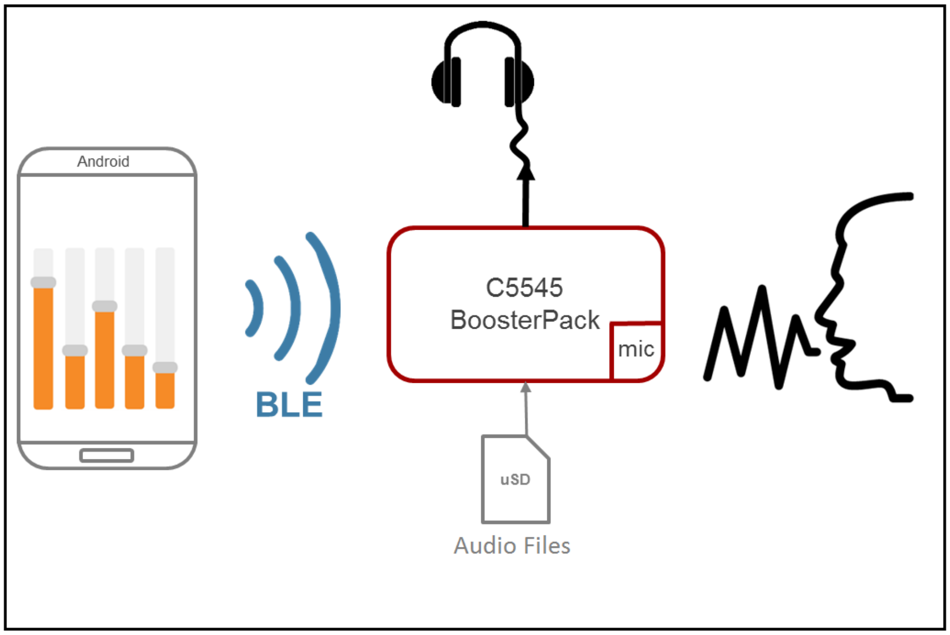


Figure 4. High-Level Blocks of the Demo

The out of box demo comprises streaming of audio samples to the headphone port of C5545 BoosterPack. The audio source will either be wave files stored on the SD card or input from LINE IN. The demo supports voice commands for playback control.

The C5545 BoosterPak Android app is a companion for the demo that acts as utility to control the equalizer settings of the playback. The audio demo running on C5545 DSP and the Android app will be communicating through the CC2650 BLE controller on the BoosterPack.

3.1 Out of Box Demo Features

Below are the features supported by the out of box demo on C5545 BoosterPack:

- Wave file playback from the SD card
  - Stereo and mono wave files with all standard sampling rates
- Audio loopback from the LINE IN to the HEADPHONE port
- Voice commands (play/pause/stop) to control the playback
- Push buttons to control the playback
- OLED display for play status indication
- Equalizer control from the C5545 Boost Android app
- Play status (playing/paused/stopped) indication on the Android app
- BLE firmware upgrade through SD card

3.2 Audio Demo

3.2.1 Overview

The audio demo is a core component of the out of box demo, which runs on C5545 DSP to stream the audio data to the HEADPHONE port of the BoosterPack. Figure 5 shows the software components involved in audio demo.

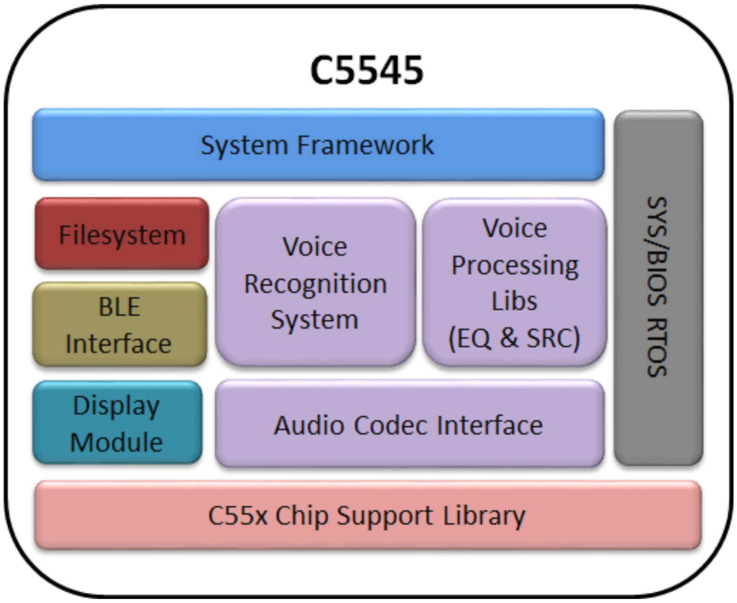


Figure 5. Software Components Involved in Audio Demo

3.2.1.1 C55x Chip Support Library

TI's C55x CSL provides low-level software libraries to configure and control hardware peripherals of the C55xx family of DSPs. It provides hardware abstraction to maintain high-level software compatibility between different C55x family DSPs. Out of box demo uses C55x CSL for hardware access, which makes it easily portable across various platforms of the family.

3.2.1.2 Audio Codec Interface

Audio codec interface handles sending and receiving data to and from audio codec on BoosterPack. Audio samples are sent to codec for playback. Audio samples are received from codec for voice commands or when LINE IN is selected as audio source.

3.2.1.3 Display Module

The display module outputs text on the OLED screen that indicates several status and information messages to the user. Messages displayed on OLED assist the user to select particular options and control the demo using the switches available on the BoosterPack board.

3.2.1.4 Voice Recognition System

The voice recognition system decodes voice command input from the user to control playback. The TIesr module is used for voice decoding. The voice recognition system integrates TIesr with the audio demo. The voice recognition system is disabled when LINE IN is selected as the audio source. TIesr voice trigger grammar phrases can be created by downloading the TIesr source from http://www.ti.com/tool/TIDEP0066.

3.2.1.5 Voice Processing Libraries

Voice processing libraries include equalizer and sample rate converter (SRC) modules. Equalizer library is used to control the audio effects based on the equalizer settings received from the Android app. The sampling rate of the hardware codec is fixed at 8 KHz for wave file playback and at 48 KHz for LINE IN loopback. The SRC library takes care of converting audio samples to 8 KHz to match with the hardware codec sampling rate.

3.2.1.6 BLE Interface

The BLE interface module handles the audio demo communication with the BLE controller. It uses the serial peripheral interface (SPI) interface of DSP to communicate with BLE. This module takes care of handling the requests coming from the Android app through BLE for equalizer control. This module also takes care of upgrading the BLE firmware.

3.2.1.7 Filesystem

TI's ATA filesystem is integrated with the audio demo to read the files from the SD card. Filesystem is used to read the audio samples or the BLE firmware from the SD card. It supports FAT16 and FAT32 formats.

3.2.1.8 DSP/BIOS RTOS

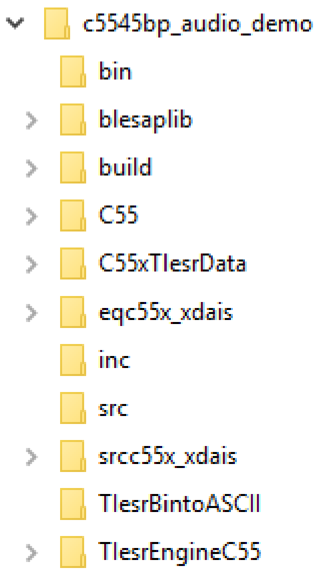
The out of box demo implements multiple software interrupts to handle different requests simultaneously. TI's DSP/BIOS RTOS should be used for scheduling and other OS related resources.

3.2.1.9 System Framework

System framework integrates and coordinates the modules described in the sections above. It takes care of handling interrupts, user requests and invoking respective modules to take necessary action. The system framework should take care of system initialization and initial task scheduling.

3.2.2 Folder Structure

Figure 6 shows the folder structure of the audio demo.



c5545bp_audio_demo
bin
blesaplib
build
C55
C55xTIesrData
eqc55x_xdais
inc
src
srcc55x_xdais
TIesrBintoASCII
TIesrEngineC55

Figure 6. Folder Structure of Audio Demo

- bin – contains boot image for audio demo
- build – contains CCS project setup for compiling and generating the executable binaries for audio demo. It also includes DSP/BIOS configuration files of the system.
- C55 – contains TIesr software for C55x platform
- C55xTIesrData – contains the grammar information for TIesr (see http://www.ti.com/tool/TIDEP0066)
- eqc55x_xdais – contains equalizer software library for C55x platform
- inc – contains the system software include files
- src – contains the system software source files
- srcc55x_xdais – contains SRC library for C55x platform
- TIesrEngineC55 – contains CCS project setup to build TIesr library

3.2.3 Build Setup

3.2.3.1 Prerequisites

Install and setup software tools listed below for building the audio demo:

- CCSv6
- C55 Code gen tool – v4.4.1
- DSP/BIOS – v5.42.1.09
- C55x CSL library
  - CSL library from TI website can be downloaded here

3.2.3.2 Build Procedure

Use the following steps to build the audio demo binaries:

1. Open CCS and select the menu ‘Project → Import CCS Projects’.
2. In the new Window, click on the ‘Browse’ button, select the audio demo root folder and click ‘OK’.
3. Select all the projects and click on the ‘Finish’ button. Projects should appear in the ‘Project Explorer’ window, as shown below.

4. Navigate to the C55xx CSL source folder (Default path - C:\ti\c55_lp) and import the following projects:
   (a) C55XXCSL_LP
   (b) atafs_bios_drv_lib

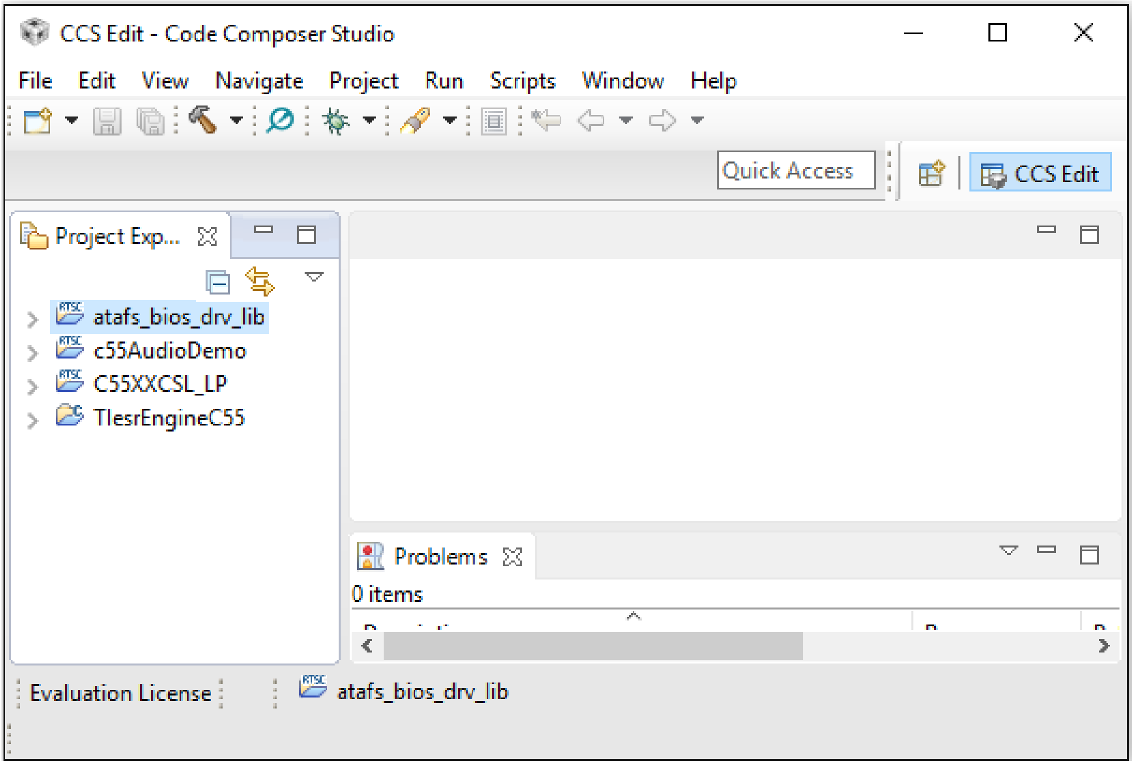
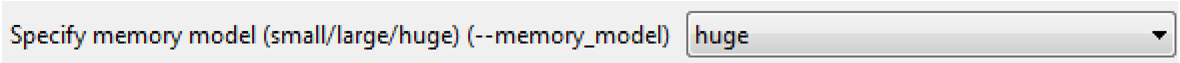


Figure 7. CCS Workspace After Importing Audio Demo Projects

5. Navigate to C:\ti\c55_lp \ c55_csl_x.xx\inc\csl_general.h and change the macro definitions below:

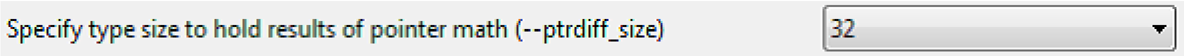
```
#define CHIP_C5517 to //#define CHIP_C5517
#define CHIP_C5505_C5515 to //#define CHIP_C5505_C5515
#define CHIP_C5535 to //#define CHIP_C5535
```

6. Navigate to Properties → Build → C5500 Compiler → Processor Options in atafs_bios_drv_lib and change Specify memory model to huge.



Specify memory model (small/large/huge) (--memory_model) huge

7. Change Specify type size to hold results of pointer math to 32 if not specified In Properties → Build → C5500 Compiler → Advanced Options → Runtime Model Options.



Specify type size to hold results of pointer math (--ptrdiff_size) 32

8. Uncheck use large memory model, if checked.
9. Repeat the above three steps for the C55XXCSL_LP project.
10. Build C55XXCSL_LP and atafs_bios_drv_lib in both Debug and Release modes.
11. Right click on the project ‘C55AudioDemo’ and select ‘build Project’.
12. Successful completion of the build creates boot image at the path c5545bp_audio_demo\bin.

To run the demo, see Section 3.6 for the procedure.

3.3 BLE Firmware

3.3.1 Overview

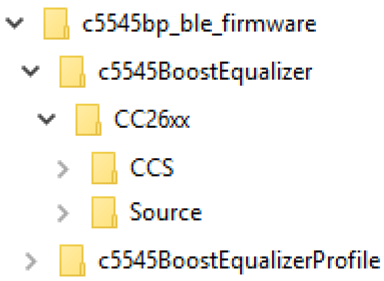
BLE firmware running on the CC2650 BLE chip provides Bluetooth connectivity to the C5545 BoosterPack. BLE firmware takes care of communicating with the Android app and sending the equalizer values received from Android app to the audio demo running on C5545 DSP. It also takes care of sending the playback status to the Android app based on the updates received from DSP. BLE firmware uses the BLE stack for CC26xx from Texas Instruments.

BLE firmware provides six different characteristics as part of the BLE profiles: five characteristics, one for each equalizer band, and sixth characteristics for play status notification. All of the equalizer band characteristics are configured for read and write access. Play status characteristic is configured for read and notify access. When scanned on C5545 Boost Android app, BoosterPack appears with the name ‘BoosterPack’.

BLE firmware communicates with the C5545 DSP over SPI interface. BLE works as the SPI slave and DSP works as the SPI master.

3.3.2 Folder Structure

Figure 8 shows the folder structure of BLE firmware.



```
c5545bp_ble_firmware
  c5545BoostEqualizer
    CC26xx
      CCS
      Source
  c5545BoostEqualizerProfile
```

Figure 8. Folder Structure of the BLE Firmware

- c5545BoostEqualizer – contains the CCS build setup and source code for the BLE firmware. Sub-folder ‘CCS’ contains the project build setup and ‘Source’ folder contains the source code.
- c5545BoostEqualizerProfile – contains source code for BLE profiles that are specific to the BoosterPack BLE app. New profiles can be added or existing profiles can be edited by changing this code.

3.3.3 Build Setup

3.3.3.1 Prerequisites

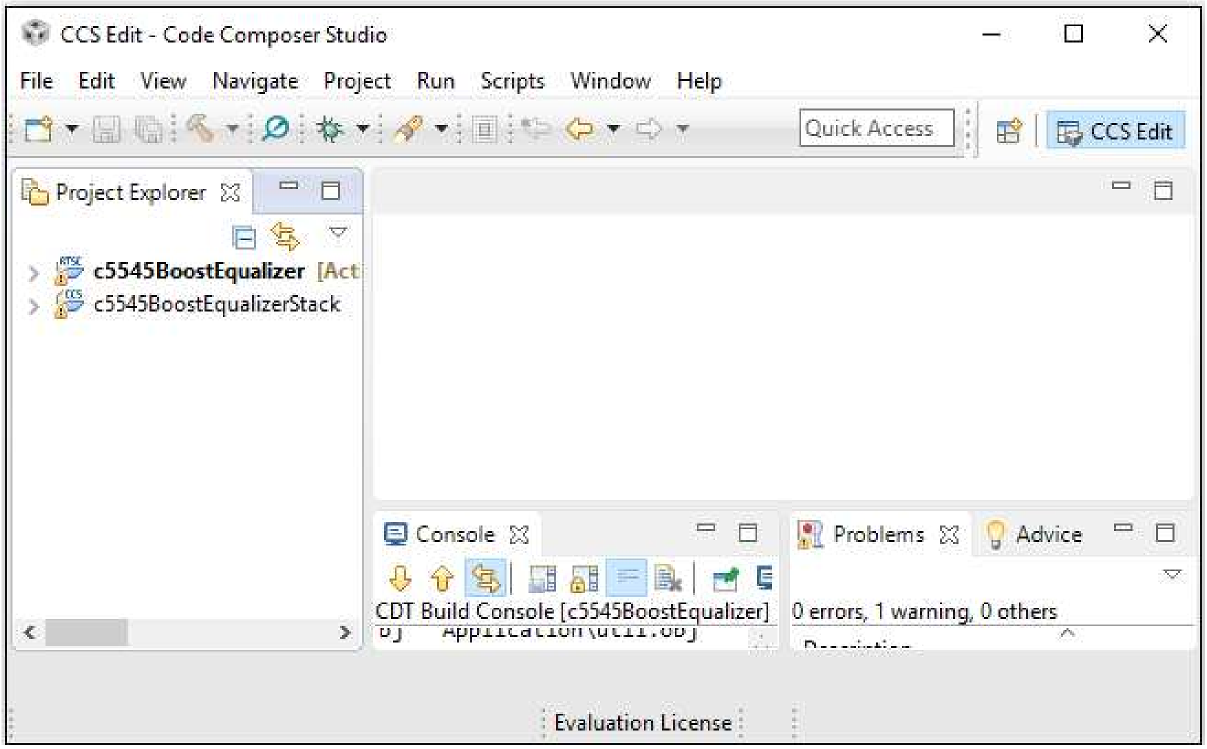
Install and setup the software tools listed below for building the BLE firmware:

- CCSv6
- CC26xx Code gen tool – v5.2.4 or later
- TI RTOS for SimpleLink – v2.13.00.06 or later
- TI BLE stack – v2.1.0 or later
- XDC tools – 3.31.1.33

3.3.3.2 Build Procedure

TI's Simplelink BLE stack is required to build the BLE firmware. Use the following steps to build the BLE firmware:

1. Download and install TI's Simplelink BLE stack from the TI Website (www.ti.com).
2. Copy the folder 'c5545BoostEqualizer' from the BLE firmware package to <Simplelink BLE stack Root>\Projects\ble.
3. Copy the folder 'c5545BoostEqualizerProfile' from the BLE firmware package to <Simplelink BLE stack Root>\Projects\ble\Profiles.
4. Open CCS and select the menu 'Project → Import CCS Projects'.
5. In the new Window, click on the 'Browse' button, select BLE firmware folder 'c5545BoostEqualizer' and click 'OK'.
6. Select all the projects and click on the 'Finish' button. Projects should appear in the 'Project Explorer' window as shown in Figure 9.



CDT Build Console [c5545BoostEqualizer]

0 errors, 1 warning, 0 others

Figure 9. Project Explorer Window

7. For building the BLE stack, right click on the project 'c5545BoostEqualizerStack' and select 'Build Project'. Successful build of stack creates c5545BoostEqualizerStack.out in the 'c5545BoostEqualizer\CC26xx\CCS\ c5545BoostEqualizerStack\FlashROM' folder.
8. For building the BLE app, right click on the project 'c5545BoostEqualizer' and select 'Build Project'. Successful build of stack creates c5545BoostEqualizer.out in the 'c5545BoostEqualizer\CC26xx\CCS\ c5545BoostEqualizer\FlashOnlyOAD' folder.

For the procedure to flash the BLE firmware onto the BoosterPack, see Section 3.6.

3.4 C5545 Boost Android App

The C5545 Boost Android app of C5545 BoosterPack out of box demo provides a GUI-based application running on Android devices to communicate with the BoosterPack over Bluetooth and control the equalizer parameters.

The C5545 Boost Android app is supported on the Android OS Lollipop and Marshmallow versions. Figure 10 through Figure 15 show different screens of the app.

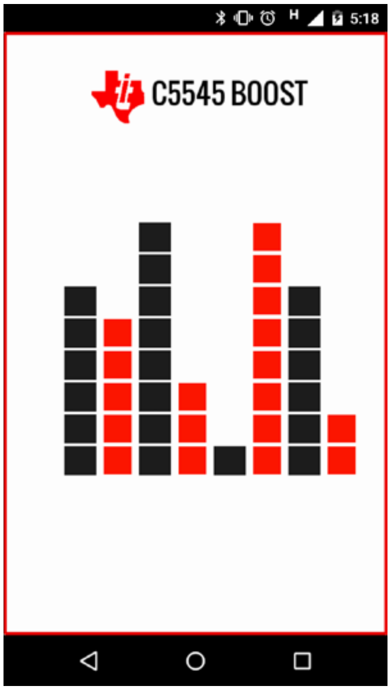
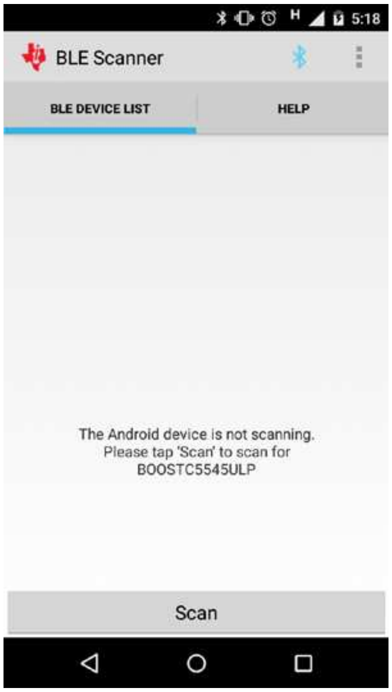


Figure 10. Equalizer App Home Screen



The Android device is not scanning.
Please tap 'Scan' to scan for
BOOSTC5545ULP

Figure 11. BLE Scanner Screen

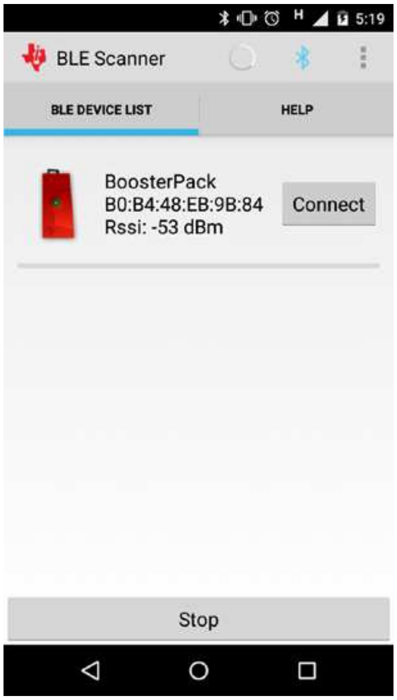


Figure 12. Scanner Screen After Scanning

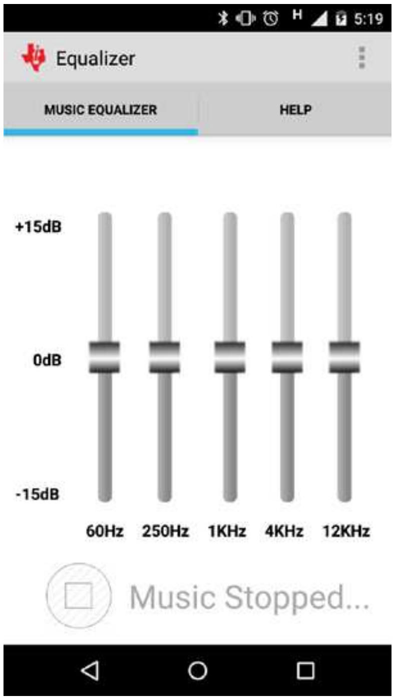


Figure 13. Equalizer With Playback Stopped

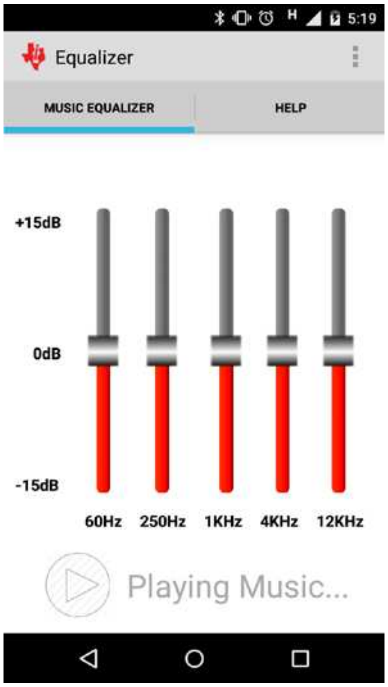


Figure 14. Equalizer While Playing

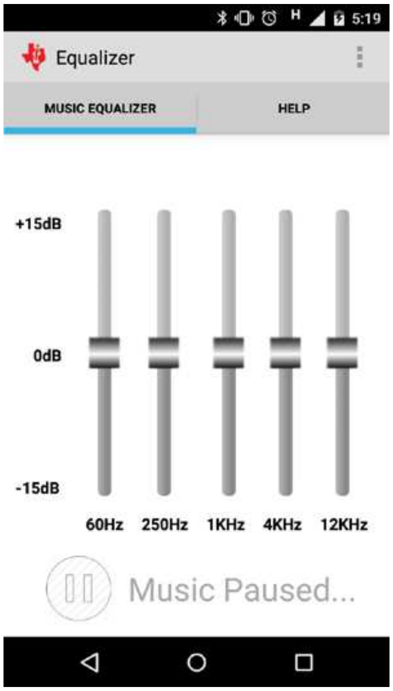


Figure 15. Equalizer With Playback Paused

3.4.1 Folder Structure

Figure 16 shows the folder structure of the equalizer Android app.

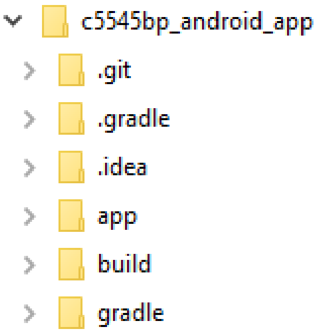


Figure 16. Folder Structure of Equalizer Android App

- app – contains Android app source code and project setup files
- build/gradle – contains setup for building the Android app

3.4.2 Build Setup

3.4.2.1 Prerequisites

- Android mobile/tablet with Lollipop or Marshmallow version
- Android mobile/tablet with BLE support
- Android studio with android SDK version 21-23

3.4.2.2 Build Procedure

The build process for a C5545 BoosterPack Android app module follows these steps:

1. Import the C5545 BoosterPack Android app source code to the android studio.
2. Compile the source code to generate the APK either in debug or release mode by selecting the Build Variant (Build → Select Build environment) in Android studio.
3. Before the application can be installed and deployed onto an Android device, the APK must be signed.
4. The APK Packager signs the APK using either the debug or release keystore. The application intended to be posted to the Play Store should be signed in release mode. Use the steps mentioned in the following link:
   https://developer.android.com/studio/publish/app-signing.html

3.5 BLE Firmware Upgrade

The BLE firmware upgrade feature of C5545 BoosterPack enables the users to upgrade the CC2650 BLE firmware from C5545 DSP.

Use the following steps to upgrade BLE firmware:

1. Rename the BLE app hex binary to bleApp.hex.
2. Rename the BLE stack hex binary to bleStack.hex.
3. Copy both the hex files to the C5545 BoostePack SD card having OoB demo binaries.
4. Insert the SD card to the BoosterPack and Power ON the board.
5. Wait for system initialization to complete and the BoosterPack to detect the BLE firmware.
6. When BoosterPack detects the firmware files, an option for the BLE firmware upgrade will be displayed on the OLED screen. Select SW2 to start the BLE firmware upgrade.
7. Wait until the BLE firmware upgrade is complete.
8. DSP reboots BLE after firmware upgrade, which starts executing the new firmware.

CAUTION

Bootloader back door should be enabled on the BLE chip for the BLE firmware upgrade feature to work. Ensure that any BLE app firmware being upgraded has the following CCFG configurations in the code.

```
#define SET_CCFG_BL_CONFIG_BOOTLOADER_ENABLE            0xC5
#define SET_CCFG_BL_CONFIG_BL_LEVEL                     0x1
#define SET_CCFG_BL_CONFIG_BL_PIN_NUMBER                0x2
#define SET_CCFG_BL_CONFIG_BL_ENABLE                    0xC5
```

3.6 Setup and Execution

This section describes the steps to program and run the out of box demo on C5545 BoosterPack. C5545 BoosterPack comes with the out of box demo binaries programmed to DSP and BLE by default. Use the steps in this section to reprogram the out of box demo binaries provided at 'software\bin\oob_demo':

3.6.1 Programming BLE Firmware

Use the following steps to flash the BLE firmware to BoosterPack:

1. Connect the emulator to the JTAG-CC2650 port on BoosterPack.
   (a) Pin converter may be needed while using standard emulators with 14/20 pin headers. Please check the references at the following links:
   http://www.spectrumdigital.com/14-pin-to-20-pin-cti-jtag-adapter
   http://www.spectrumdigital.com/cti20-pin-to-arm10-pin-jtag-adapter
2. Connect the micro USB cable to the DEBUG micro-USB port of BoosterPack and the host PC.
3. Open the CCS IDE and launch the target configuration for CC2650.
4. Use the following steps to create the target configuration file, if needed.
   (a) Select the menu 'File → New → Target Configuration File'
   (b) Give a name to the target configuration file and click 'Finish'.
   (c) In the new target configuration file created:
      (i) Select 'Connection' based on the emulator being used.
      (ii) Select 'Board or Device' as CC2650F128 and save the file.
      (iii) Select 'View → Target Configurations' to open the Target Configurations window.
   (d) Right click on the CC2650 Target Configuration file that was created and select 'Launch Selected Configuration'.
5. After target is launched, select 'Run → Connect Target', which connects to the target.
6. After target is connected, select 'Run → Load → Load Program' to load the program.
7. Browse and load the 'c5545BoostEqualizerStack.out' program.
8. Target will not halt after running the program above; ignore it and go to the next step.
9. Browse and load the 'c5545BoostEqualizer.out' program.
10. Disconnect the JTAG and reboot the board.

3.6.2 Programming DSP Firmware

1. Format a SD card using the SD formatter tool that can be downloaded from the following link:
   https://www.sdcard.org/downloads/formatter_4
2. Copy the audio demo boot image (bootimg.bin) to root directory of SD card.

NOTE: The SD card delivered with BoosterPack has correct formatting and the out of box demo binaries copied to the SD card. Skip this step while using BoostePack SD card as is.

3.6.3 Installing Android App

1. Copy or download the C5545BOOST.apk to the Android device.
2. Enable the settings of the Android device to install external Apps, if needed.
   (a) Go to Menu → Settings → Security → and check Unknown Sources.
3. Click on the C5545BOOST.apk downloaded to the Android device and select Install.
4. After successful installation, the C5545 Boost Android app appears in the app list with the icon shown in Figure 17.

Figure 17. C5545 Boost Icon

3.6.4 Running Out of Box Demo

3.6.4.1 Wave File Playback

1. Insert the SD card with the audio demo boot image and a few wave files to the microSD slot of the BoosterPack.
2. Connect headset to the HEADPHONE port of BoosterPack.
3. Power ON the BoosterPack.
4. Select Audio source as SD (SW3).
5. Wait until the system initialization is complete.
6. Launch the Equalizer app on the Android device.
7. Allow the app to enable Bluetooth, if not already enabled.
8. Press on the 'Scan' button to start scanning for the BoosterPack.
9. Press the 'Connect' button once the 'BoosterPack' device is detected by the App.
10. Start playback by pressing SW2 or by speaking 'play' near the BoosterPack on-board MIC.
11. Slide the equalizer bars on the app to change the equalizer values of playback.
12. Observe the change in audio at the headphone output while changing the equalizer values from the app.
13. Pause the playback by pressing SW2 or by speaking 'pause' near the BoosterPack on-board MIC.
14. Resume playback by pressing SW2 or by speaking 'play' near the BoosterPack on-board MIC.
15. Stop playback by pressing SW3 or by speaking 'stop' near the BoosterPack on-board MIC.
16. Observe the change in the C5545 Boost app playback status when playback is stopped/paused and playback is running.

3.6.4.2 LINE IN Loopback

1. Insert the SD card, with the audio demo boot image, to the microSD slot of BoosterPack.
2. Connect LINE IN cable between an audio device (PC/Mobile/audio player) and LINE IN port of BoosterPack.
3. Connect headset to the HEADPHONE port of the BoosterPack.
4. Power ON the BoosterPack.
5. Select Audio source as LINE IN (SW4).
6. Wait until the system initialization is complete.

7. Launch the Equalizer app on the Android device.
8. Allow the app to enable Bluetooth, if not already enabled.
9. Press the 'scan' button to start scanning for the BoosterPack.
10. Press the 'connect' button once the 'BoosterPack' device is detected by the app.
11. Play an audio file on the audio device connected to LINE IN.
12. Press the 'SW2' button on the BoosterPack to start playback.
13. Slide the equalizer bars on the app to change the equalizer values of playback.
14. Observe the change in audio at the headphone output while changing the equalizer values from the app.
15. Press the 'SW2' button to pause and the 'SW3' button to stop playback on the BoosterPack. Press the 'SW2' button to resume playback.
16. Observe the change in the C5545 Boost app playback status when playback is stopped/paused and playback is running.

NOTE: Voice command support is not available while running the demo with the audio source as LINE IN.

4 References

- *TMS320C5545 BoosterPack Hardware User's Guide* (SPRUI90)
- C5545 DSP
- CC2650 BLE
- MSP432 LaunchPad
- CC3200 Launchpad
- CCSv6
- TI E2E Community
- TI Wiki
- TI Codegen Tools
- DSP/BIOS RTOS
- TIesr

Appendix A

SPRUI92–October 2016

Diagnostic Test Procedure

Diagnostic test binaries referred in this document are available on the SD card at the software\bin\diagnostics path. Some of the tests may need CCS IDE to connect to different hardware platforms from CCS. For more details about connecting to different targets from CCS, see Appendix B.

A.1 Initial Steps

1. Copy the diagnostic boot image (software\bin\diagnostics\bootimg.bin) to the SD card root directory and insert the card into the SD slot of the BoosterPack.
2. Short pins 1 and 3 of the JP2 and JP3 ports on the board.
3. Short the pins on the J2 port. Use this jumper configuration as default unless a different setup is mentioned in the test procedure.
4. Connect the micro USB between the BoosterPack DEBUG microUSB port and the host PC.
5. Set SW6 to ON.
6. Open the serial console (example, Teraterm) on the host PC, connect to the COM port where the BoosterPack Debug port is connected and setup for the following configurations:
   (a) Baud rate - 115200
   (b) Data length - 8 bit
   (c) Parity - None
   (d) Stop bits - 1
   (e) Flow control - None

7. Press the RESET switch on the BoosterPack or press the Enter key on the serial console to display the diagnostic test menu on the serial console as shown below:

```
*******************************************************************
               C5545 BoosterPack HW Diagnostic Tests
*******************************************************************


Diagnostic Tests                         Pass            Fail
-----------------------                 ------          ------
  0 - Auto Run All Tests                  0               0
  1 - SPI Flash Test                      0               0
  2 - LED Test                            0               0
  3 - OLED Display Test                   0               0
  4 - Audio Playback Test                 0               0
  5 - Audio LINE IN Loopback Test         0               0
  6 - Audio On-board MIC Loopback Test    0               0
  7 - Audio Headset Loopback Test         0               0
  8 - Audio External MIC Loopback Test    0               0
  9 - Current Monitor Test                0               0
 10 - RTC Test                            0               0
 11 - Push Button Test                    0               0
 12 - USB Test                            0               0
 13 - I2C DSP Master - MSP Slave          0               0
 14 - I2C DSP Slave - MSP Master          0               0
 15 - DSP-BLE Interface Test              0               0

  q - Quit

Enter Desired Option:
```

A.1.1 Test Accessories

Below are the test accessories required for executing the diagnostic tests:

- BoosterPack
- Windows PC with CCS and TeraTerm installed
- Micro SD card
- Two Micro USB cables
- Mini USB cable
- Headset with MIC
- LINE IN Cable
- Micro Phone or headset with separate MIC port
- MSP432 LaunchPad
- CC3200 LaunchPad
- XDS110/XDS200 emulator
- CTI 20-pin/14-pin to ARM10-pin adapter for BLE programming
  - http://www.spectrumdigital.com/14-pin-to-20-pin-cti-jtag-adapter
  - http://www.spectrumdigital.com/cti20-pin-to-arm10-pin-jtag-adapter

A.1.2 DSP Interface Tests

A.1.2.1 SPI Test

A.1.2.1.1 Test Accessories

No additional accessories are required for this test.

A.1.2.1.2 Test Setup

No additional test setup is required.

A.1.2.1.3 Test Execution

SPI flash device data write, data read and block erase operations are verified during SPI flash test. The sample SPI test log is shown below:

```
********************************
         SPI FLASH Test
********************************

Running SPI flash erase test...

SPI flash erase test passed!!


Running SPI flash data read/write test...

SPI Write & Read buffer matching

SPI flash read write test Passed!!

SPI FLASH Test Completed!!
```

A.1.2.1.4 LED Test

A.1.2.1.4.1 Test Accessories

No additional accessories are required for this test.

A.1.2.1.4.2 Test Setup

No additional test setup is required.

A.1.2.1.4.3 Test Execution

LED test toggles three user LEDs on the BoosterPack. The sample test log for LED test is shown below:

```
**********************
       LED Test
**********************

Check if all three LED's on the BoosterPack are toggling
Press Y/y if three LED's are blinking properly, Any other key for failure:
y

LED Test Passed!

LED Test Completed!!
```

A.1.2.1.5 OLED Test

A.1.2.1.5.1 Test Accessories

No additional accessories are required for this test.

A.1.2.1.5.2 Test Setup

No additional test setup is required.

A.1.2.1.5.3 Test Execution

Test verifies device detection and configuration of OLED module on the BoosterPack by displaying test messages on to line1 and line2 of the OLED. Messages displayed on the screen should scroll at the end of the test. The test prompts the user to confirm the OLED display. The sample test log for the OLED test is shown below:

```
*************************************
              OLED TEST
*************************************
Running OLED Device Detect Test...
OLED detection Successful

Running OLED Display Test...

Displaying 'TI C5545 BP' on first row...

Displaying 'TI C5545 BP' on second row...

Displaying 'Texas Instruments' on first row and
'C5545 BoosterPack' on the second row

Scrolling OLED Display from Right to Left
OLED Display Test Completed

Press Y/y if the test messages are displayed properly on OLED,
any other key for failure:
y

 OLED Test Passed!

OLED Test Completed!!
```

A.1.2.1.6 Audio Playback Test

A.1.2.1.6.1 Test Accessories

Headset

A.1.2.1.6.2 Test Setup

Connect the headset to HEADPHONE port of the BoosterPack

A.1.2.1.6.3 Test Execution

Audio playback test verifies AIC3206 audio codec interfaces by sending an audio stream to the HEADPHONE port of the BoosterPack through AIC3206. This test verifies Tx path of on-board audio interface. Press the 'SW3' button to stop running the test. Press ‘y’ in case the audio output is proper or any other key for failure. The sample audio playback test log is shown below:

```
***************************************
          AUDIO PLAYBACK TEST
***************************************
Test outputs a sine tone on HEADPHONE port of BoosterPack

Connect the headset to the HEADPHONE port of the BoosterPack
Observe the Audio Tone at HEADPHONE port

Starting the Audio Tone Play...

Press SW3 on the BoosterPack for exiting from the test

Press Y/y if Audio output from the HEADPHONE port is proper,
any other key for failure:
y

 Audio Playback Test Passed!

Audio Playback Test Completed!
```

A.1.2.1.7 Audio LINE IN Loopback Test

A.1.2.1.7.1 Test Accessories

- 3.5 mm male to male audio (LINE IN) cable
- Headset
- Test PC / audio device

A.1.2.1.7.1.1 Test Setup

1. Connect the 3.5 mm male to the male audio (LINE IN) cable between audio OUT of the PC/audio device and LINE IN port of the BoosterPack.
2. Connect the headset to the HEADPHONE port of the BoosterPack.

A.1.2.1.7.2 Test Execution

The audio loopback test verifies the AIC3206 audio codec interfaces by receiving an audio stream on LINE IN and sending it back to the HEADPHONE port of the BoosterPack through AIC3206. This test verifies both Tx and Rx paths of on-board audio interfaces. Play a media file on the audio device or the PC, run the audio loopback test and observe that audio is played at the HEADPHONE port of the BoosterPack. Press the 'SW3' button to stop the test. Press ‘y’ in case the audio output is proper or any other key for failure.

The sample audio LINE IN loopback test log is shown below:

```
*******************************************
        AUDIO LINE IN LOOPBACK TEST
*******************************************
Test Receives audio samples from LINE IN and output the same
on HEADPHONE port

Connect headset to the HEADPHONE port of the BoosterPack

Connect a LINE-IN cable between the audio port of the
Test PC and LINE IN of the BoosterPack

Play any audio file from the Test PC and Check
Audio from the headset connected to BoosterPack

Press SW3 on the BoosterPack for exiting from the test

Press Y/y if the Audio stream from LINE IN is
observed at the headset connected to HEADPHONE port,
any other key for failure:
y
 Audio LINE IN Loopback Test Passed!

Audio LINE IN Loopback Test
                     Completed!!
```

A.1.2.1.8 Audio MIC IN Loopback Test

A.1.2.1.8.1 Test Accessories

Headset

A.1.2.1.8.2 Test Setup

Connect the headset to the HEADPHONE port of the BoosterPack

A.1.2.1.8.3 Test Execution

The audio MIC IN loopback test verifies the interface between the on-board mic and AIC3206 by receiving an audio stream from the on-board mic and sending it back to the HEADPHONE port of the BoosterPack. Press the 'SW3' button to stop the test. Press ‘y’ in case the audio output is proper or any other key for failure.

The sample audio MIC IN loopback test log is shown below:

```
********************************************
         AUDIO MIC IN LOOPBACK TEST
********************************************
Connect headset to the HEADPHONE port of the BoosterPack

This test loops back the audio input from the on-board MIC
of the BoosterPack to the headset connected to HEADPHONE port


Press SW3 on the BoosterPack for exiting from the test

Press Y/y if the Audio received from on-board MIC of the BoosterPack is
observed at the headset connected at the HEADPHONE port properly,
any other key for failure:
y

 Audio MIC IN Loopback Test Passed!

Audio MIC IN Loopback Test Completed!
```

A.1.2.1.9 Audio Headset Loopback Test

A.1.2.1.9.1 Test Accessories

Headset with MIC

A.1.2.1.9.2 Test Setup

Connect the headset to the HEADPHONE port of the BoosterPack.

A.1.2.1.9.3 Test Execution

The audio headset loopback test verifies the interface between the headset’s MIC and AIC3206 by receiving an audio stream from the headset with the MIC and sending it back to the HEADPHONE port of the BoosterPack. Press the 'SW3' button to stop the test. Press ‘y’ in case the audio output is proper or any other key for failure.

The sample log for the audio headset loopback test is shown below:

```
*****************************************
       AUDIO HEADSET LOOPBACK TEST
*****************************************
Test receives audio samples from HP MIC IN and outputs the
same on HEADPHONE port

Connect headset to the HEADPHONE port of the BoosterPack

Speak near the headset MIC and check the audio from headset output

Press SW3 on the BoosterPack for exiting from the test

Press Y/y if the Audio received from headphone MIC is
observed at the headset output properly
any other key for failure:
y

 Audio Headset Loopback Test Passed!

Audio Headset Loopback Test Completed!
```

A.1.2.1.10 External MIC IN Loopback Test

A.1.2.1.10.1 Test Accessories

PC microphone with one speaker jack and one audio jack

A.1.2.1.10.2 Test Setup

Connect the speaker jack of the PC microphone to the LINE IN and audio jack to the HEADPHONE port of the BoosterPack.

A.1.2.1.10.3 Test Execution

The external MIC IN loopback test verifies AIC3206 audio codec interfaces by receiving an audio stream on LINE IN and sending it back to the HEADPHONE port of the BoosterPack through AIC3206. This test verifies both Tx and Rx paths of on-board audio interfaces.

Speak something from the PC microphone’s mic and check whether you can hear the same on the HEADPHONE port of the BoosterPack. Press the 'SW3' button to stop the test. Press ‘y’ in case the audio output is proper or any other key for failure.

The sample log for the external MIC IN loopback test is shown below:

```
***************************************************
         AUDIO EXTERNAL MIC IN LOOPBACK TEST
***************************************************
Test Receives audio samples from PC microphone's MIC
and outputs the same on HEADPHONE port

Connect a PC microphone audio input jack to the LINE IN and
Speaker jack to the HEADPHONE port of the BoosterPack

Speak near the MIC of the PC microphone and
check Audio from the HEADPHONE port

Press SW3 on the BoosterPack for exiting from the test

Press Y/y if the Audio stream from the PC microphone
is properly Loopback to the HEADPHONE port of the BoosterPack,
any other key for failure:
y

 Audio External MIC In Loopback Test Passed!

Audio External MIC In Loopback Test Completed!
```

A.1.2.1.11 Current Monitor Test

A.1.2.1.11.1 Test Accessories

No additional accessories are required for this test.

A.1.2.1.11.2 Test Setup

No additional test setup is required.

A.1.2.1.11.3 Test Execution

This test verifies the four INA219 devices on the BoosterPack by reading the current, power, bus and shunt voltage values.

The sample log for the current monitor test is shown below:

```
************************************
        Current Monitor Test
************************************
Reading values from CVDD port
Shunt voltage - 34.250000mV
Bus voltage - 1.300000V
Power - 44.516346mW
Current - 35.407639mA

Reading values from LDOI port
Shunt voltage - 35.169998mV
Bus voltage - 1.756000V
Power - 57.752338mW
Current - 33.886391mA

Reading values from DSP_DVDDIO port
Shunt voltage - 3.490000mV
Bus voltage - 1.792000V
Power - 6.249617mW
Current - 3.469874mA

Reading values from VCC3V3_USB port
Shunt voltage - 327.639984mV
Bus voltage - 0.020000V
Power - 0.000000mW
Current - 35.430191mA


Current Monitor Test Completed!
```

A.1.2.1.12 RTC Test

A.1.2.1.12.1 Test Accessories

No additional accessories are required for this test.

A.1.2.1.12.2 Test Setup

No additional test setup is required.

A.1.2.1.12.3 Test Execution

This test verifies the RTC module by setting the Date, Time and generating an alarm interrupt.

The sample log for RTC test is shown below:

```
**************************
         RTC Test
**************************

RTC Set Time Test...


This test demonstrates RTC Time functionality
RTC Time will be set, read and displayed 10 times.
RTC interrupt will be generated for each Second

RTC_setCallback Successful
Setting RTC Time Successful
Setting RTC Date Successful
Setting RTC Events Successful

Starting the RTC

Iteration 1: Time and Date is : 12:12:12:0021, 16-10-08
Iteration 2: Time and Date is : 12:12:12:0026, 16-10-08
Iteration 3: Time and Date is : 12:12:12:0031, 16-10-08
Iteration 4: Time and Date is : 12:12:12:0036, 16-10-08
Iteration 5: Time and Date is : 12:12:12:0041, 16-10-08
Iteration 6: Time and Date is : 12:12:12:0046, 16-10-08
Iteration 7: Time and Date is : 12:12:12:0051, 16-10-08
Iteration 8: Time and Date is : 12:12:12:0056, 16-10-08
Iteration 9: Time and Date is : 12:12:12:0061, 16-10-08
Iteration 10: Time and Date is : 12:12:12:0067, 16-10-08

RTC Set Time Test Passed!!

RTC Test Completed!
```

A.1.2.1.13 Push Button Test

A.1.2.1.13.1 Test Accessories

No additional accessories are required for this test.

A.1.2.1.13.2 Test Setup

No additional test setup is required.

A.1.2.1.13.3 Test Execution

This test verifies the working of three push button switches on the BoosterPack by generating an interrupt to display the name of the switch when a push button switch is pressed.

Every press on the push button switch displays two options: either to continue or to exit, enter ‘y’ or ‘Y’ to continue (or) enter ‘x’ or ‘X’ to exit on the TeraTerm.

The sample test log for push button test is shown below:

```
********************************
        Push Button Test
********************************
Press any push button switch on the BoosterPack and check
if the corresponding switch name is displayed on the console

SW2 is pressed
Press X/x for exiting the test
Press Y/y to continue
y
continue
SW3 is pressed
Press X/x for exiting the test
Press Y/y to continue
y
continue
SW4 is pressed
Press X/x for exiting the test
Press Y/y to continue
x
Exiting from the push button test

Push Button Test Completed!
```

A.1.2.1.14 USB Test

A.1.2.1.14.1 Test Accessories

- Micro USB cable
- Micro USB EP Test Tool (c55xx_usb_ep_diag.exe)

A.1.2.1.14.1.1 Test Setup

Connect the micro USB cable between the Test PC and CLIENT Port of the BoosterPack.

A.1.2.1.14.1.2 Test Execution

This test verifies the USB interface by moving the mouse cursor left and right by pressing SW2 and SW4 respectively. SW3 is used to exit the test. Test will prompt for user input to confirm if the corresponding events are sent properly through USB.

The sample log for USB test on BoosterPack is shown below:

```
****************************
          USB Test
****************************
Press SW2 to move the mouse cursor to LEFT
Press SW4 to move the mouse cursor to RIGHT
Press SW3 to exit from the test

Press Y/y if the corresponding events are sent properly
through USB, any other key for failure
y
 USB Test Passed!

USB Test Completed!!
```

A.1.2.2 LaunchPad Interface Tests

This section describes the procedure for running the diagnostic tests between the C5545 BoosterPack and MSP432, CC3200 LaunchPads. Binaries for the MSP432 LaunchPad are available in the ‘msp432_lp’ folder and binaries for the CC3200 LaunchPad are available in the ‘cc3200_lp’ folder in the diagnostic binary package.

A.1.2.2.1 C5545 BP to MSP432 LP GPIO Test

A.1.2.2.1.1 Test Accessories

- Two micro USB cables
- MSP432 LaunchPad
- Energia IDE

A.1.2.2.1.2 Test Setup

1. Connect the MSP432 to the BoosterPack by using the Expansion connectors.
2. Connect one micro USB cable with the BoosterPack’s Debug port and the Test PC and the other micro USB cable with MSP432 LaunchPad and the Test PC.
3. Configure the UART jumpers on BoosterPack to enable the communication between DSP and LaunchPad:

   (a) JP2 – Short pin 3 and 4

   (b) JP3 – Short pin 1 and 2
4. Power ON the BoosterPack board.

A.1.2.2.1.3 Test Execution

This test verifies the GPIO lines between the DSP and the MSP432 LaunchPad.

1. MSP432 LP (from CCS):
   (a) Open CCS IDE and launch the target configuration file for MSP432 LP.
   (b) Connect to the target, load and run the gpio_launchpad_msp432_to_dsp_test.out program.
   (c) Press any key to continue the test after starting the DSP program, as described below.
2. BoosterPack (from CCS):
   (a) Open CCS IDE and launch the target configuration file for C5545 DSP.
   (b) Connect to the target, load and run the gpio_dsp_to_lp_msp432_test.out program.
   (c) Observe the logs on the CCS consoles of both MSP432 LP and C5545 BoosterPack.

The sample log for the test on BoosterPack is shown below:

```
***********************************************
       C5545 BP TO MSP432 LAUNCHPAD TEST
***********************************************


Writing HIGH to GPIO PINS 8 & 31
Waiting for LaunchPad to Write HIGH on GPIO PINS 6,7,9 & 30...

LaunchPad write completed DSP started reading...

DSP is able to read the GPIO PINS 6,7 & 9 as HIGH
Writing LOW to GPIO PINS 8 & 31

Waiting for LaunchPad to Write LOW on GPIO PIN 6,7,9 & 30...
LaunchPad write completed DSP started reading...

DSP is able to read the GPIO PINS 6,7 & 9 as LOW

GPIO C5545 BP TO LAUNCHPAD TEST Passed!
```

The sample log at LaunchPad is shown below:

```
***********************************************
        MSP432 LP to C5545 BP GPIO Test
***********************************************


Configured Uart_rx and I2s_data as input pins
Configured I2s_clk, Uart_tx, i2s_rx, i2s_fs data as output pins
Clearing Uart_tx, I2s_fx, I2s_clk and I2s_rx pins

Enter any character once DSP side code started running
1

Waiting for DSP to write HIGH for Uart_tx and I2s_data pins...
I2s_data is read HIGH
Uart_tx pin also read HIGH

Writing HIGH to the Uart_tx, I2s_fx, I2s_clk and I2s_rx

Waiting for DSP to write LOW for Uart_tx and I2s_data pins...
I2s_data pin is read LOW
Writing LOW to the Uart_tx[30], I2s_fx[7], I2s_clk[6] and I2s_rx[9]

MSP432 LP to C5545 BP GPIO Test Passed!

MSP432 LP to C5545 BP GPIO Test Completed!!
```

A.1.2.2.2 C5545 BP to CC3200 LP GPIO Test

A.1.2.2.2.1 Test Accessories

- Two micro USB cables
- CC3200 LaunchPad
- Energia IDE

A.1.2.2.2.2 Test Setup

1. Connect the CC3200 to BoosterPack by using the Expansion connectors.
2. Connect one micro USB cable with the BoosterPack’s Debug port and the Test PC and the other micro USB cable with CC3200 LaunchPad and the Test PC.
3. Configure the UART jumpers on BoosterPack to enable the communication between DSP and LaunchPad:
   (a) JP2 – Short pin 3 and 4
   (b) JP3 – Short pin 1 and 2
4. Power ON the BoosterPack board.

A.1.2.2.2.3 Test Execution

This test verifies the GPIO lines between the DSP and CC3200 LaunchPad.

1. CC3200 LP (from Energia):
   (a) Remove the jumper at J8 (TCK) and connect it to SOP2 for programming the LaunchPad.
   (b) Open Energia IDE, configure the board and serial port for CC3200 LP that is connected.
   (c) Open the sketch gpio_lp_cc3200_bp_test.ino, compile and upload it to CC3200 LP.
   (d) Put the jumper back from SOP2 to J8 for running the program.
   (e) Open the Energia serial monitor and press the 'Restart' button on LaunchPad to start running the program.
   (f) Press any key to continue the test after starting the DSP program, as described below.
2. BoosterPack (from CCS)
   (a) Open CCS IDE and launch the target configuration file for C5545 DSP.
   (b) Connect to the target, load and run the gpio_dsp_to_lp_cc3200_test.out program.
   (c) Observe the logs on the CCS and Energia consoles.

The sample log for the test on BoosterPack is shown below:

```
***********************************************
       C5545 BP TO CC3200 LAUNCHPAD TEST
***********************************************


Writing HIGH to GPIO PINS 8 & 31
Waiting for LaunchPad to Write HIGH on GPIO PINS 6,7,9 & 30...

LaunchPad write completed DSP started reading...
DSP is able to read the GPIO PINS 6,7 & 9 as HIGH

Writing LOW to GPIO PINS 8 & 31

Waiting for LaunchPad to Write LOW on GPIO PIN 6,7,9 & 30...
LaunchPad write completed DSP started reading...

DSP is able to read the GPIO PINS 6,7 & 9 as LOW

GPIO C5545 BP TO LAUNCHPAD TEST Passed!
```

The sample log at LaunchPad is shown below:

```
CC3200 LP to C5545 BP GPIO Test

Enter Any Character to Proceed further:
Waiting for DSP to write HIGH
Values are HIGH from DSP
i2s1dx    uartrx
1    1

Launchpad as Output, writing HIGH
uarttx    i2s1fs    i2s1clk    i2s1rx
1    1    1    1

Waiting for DSP to write LOW
Values are read LOW from DSP
i2s1dx    uartrx
0    0

Launchpad as Output, writing LOW
uarttx    i2s1fs    i2s1clk    i2s1rx
0    0    0    0

CC3200 LP to C5545 BP GPIO Test Passed!
```

A.1.2.2.3 MSP432 Slave DSP Master I2C Test

A.1.2.2.3.1 Test Accessories

MSP432 LaunchPad

A.1.2.2.3.2 Test Setup

1. Connect the MSP432 LP to the BoosterPack by using the Expansion connectors.
2. Connect one micro USB cable with the BoosterPack’s Debug port and the Test PC, and the other micro USB cable with MSP432 LaunchPad and the Test PC.
3. Configure the UART jumpers on BoosterPack to enable the communication between DSP and serial port:
   (a) JP2 – Short pin 1 and 3
   (b) JP3 – Short pin 1 and 3
4. Power ON the BoosterPack.

A.1.2.2.3.3 Test Execution

This test verifies the I2C interface between the MSP and DSP (while the MSP is acting as slave and DSP as master). During the test, master sends 16 bytes of data that should be received by slave properly.

1. MSP432 LP (from CCS)
   (a) Open CCS and connect to the MSP432 LaunchPad using the relevant target configuration file.
   (b) Load and run the msp_432_i2c_msp_slave_dsp_master_test.out program on MSP432.
   (c) Press any key to continue the test after DSP comes out of reset and the DSP diagnostic menu is accessible.
2. BoosterPack (from diagnostic serial console menu):
   (a) After starting the program on MSP432, confirm that the BoosterPack diagnostic menu is accessible on the serial console.
   (b) Run the ‘I2C DSP Master - MSP Slave‘ test from the diagnostic menu.
   (c) Observe the logs on CCS for MSP432 and the serial console for BoosterPack.
   (d) Verify that data sent from BoosterPack is received properly by MSP432.

The sample log for the test on BoosterPack is shown below:

```
*************************************************
        'DSP Master - MSP Slave' I2C Test
*************************************************

Sending 16 Bytes to Slave in blocks of 8
Data Sent to Slave:
0x0     0x1     0x2     0x3     0x4     0x5     0x6     0x7
0x8     0x9     0xa     0xb     0xc     0xd     0xe     0xf

'MSP Slave - DSP Master' I2C Test Completed!!
```

The sample log for the test on MSP432 is shown below:

```
*************************************************
        'MSP Slave - DSP Master' I2C Test
*************************************************
MSP432 I2C Slave Mode Test
MSP432 is Configured in Slave Mode with Address 0x38

Press Any Key to Continue After DSP is Out of Reset
1

Waiting for Data from Master...

Data Received from Master:
0x0    0x1    0x2    0x3    0x4    0x5    0x6    0x7
0x8    0x9    0xa    0xb    0xc    0xd    0xe    0xf

Data Received from Master Matched with Expected Data!

'MSP Slave - DSP Master' I2C Test Passed!
'MSP Slave - DSP Master' I2C Test Completed!!
```

A.1.2.2.4 MSP432 Master DSP Slave I2C Test

A.1.2.2.4.1 Test Accessories

MSP432 LaunchPad

A.1.2.2.4.2 Test Setup

1. Connect the MSP432 LP to the BoosterPack by using the Expansion connectors.
2. Connect one micro USB cable with the BoosterPack's Debug port and the Test PC, and the other micro USB cable with MSP432 LaunchPad and the Test PC.
3. Configure the UART jumpers on BoosterPack to enable the communication between DSP and the serial port:
   (a) JP2 – Short pin 1 and 3
   (b) JP3 – Short pin 1 and 3
4. Power ON the BoosterPack.

A.1.2.2.4.3 Test Execution

This test verifies the I2C interface between the MSP and DSP while the MSP is acting as master and DSP as slave. During the test, master sends 16 bytes of data thta should be received by slave properly.

1. MSP432 LP (from CCS):
   (a) Open CCS and connect to the MSP432 LaunchPad using the relevant target configuration file.
   (b) Load and run the msp432_i2c_msp_master_dsp_slave_test.out program on MSP432.
   (c) Press any key to continue the test after running the DSP side program.

2. BoosterPack (from Diagnostic serial console menu):
   (a) After starting the program on MSP432, confirm that the BoosterPack diagnostic menu is accessible on the serial console.
   (b) Run the ‘I2C DSP Slave - MSP Master‘ test from the diagnostic menu.
   (c) Observe the logs on CCS for MSP432 and serial console for the BoosterPack.
   (d) Verify that data sent from MSP432 is received properly by the BoosterPack.

The sample log for the test on BoosterPack is shown below:

```
*******************************************************
         'DSP Slave - MSP Master' I2C Test
*******************************************************
Waiting for the MSP to transfer data...

Data Received from Master:
0x0     0x1     0x2     0x3     0x4     0x5     0x6     0x7
0x8     0x9     0xa     0xb     0xc     0xd     0xe     0xf

Data Received from Master Matched with Expected Data!

'MSP Master- DSP Slave ' I2C Test Passed!!

'MSP Master - DSP Slave ' I2C Test Completed!!
```

The sample log for the test on MSP432 is shown below:

```
*******************************************************
         'MSP Master - DSP Slave' I2C Test
*******************************************************
Press Any Key to Continue After Running DSP Program
1

Sending 16 Bytes to Slave
0x0    0x1    0x2    0x3    0x4    0x5    0x6    0x7
0x8    0x9    0xa    0xb    0xc    0xd    0xe    0xf

'MSP Master - DSP Slave' I2C Test Completed!!
```

A.1.2.2.5 MSP432 INA Device Test

A.1.2.2.5.1 Test Accessories

MSP432 LaunchPad

A.1.2.2.5.2 Test Setup

- Connect the MSP432 LP to the BoosterPack by using the Expansion connectors.
- Connect one micro USB cable with the BoosterPack’s Debug port and the Test PC, and the other micro USB cable with MSP432 LaunchPad and the Test PC.

A.1.2.2.5.3 Test Execution

This test verifies accessing the INA device on the BoosterPack from the MSP432 LaunchPad.

1. MSP432 LP (from Energia):
   (a) Open Energia IDE, configure the board and serial port for MSP432 LP that is connected.
   (b) Open the sketch msp_ina_access.ino, compile and upload it to MSP432 LP.
   (c) Open Energia serial monitor and press restart button on LaunchPad to start running the program.

The sample test log is shown below:

```
MSP432 INA Device Test


Reading values from CVDD port

shunt Voltage -34.02mv
Bus Voltage - 1.31V
Power - 44.52mW
Current - 34.05mA

Reading values from LDOI port

shunt Voltage -34.81mv
Bus Voltage - 1.76V
Power - 50.46mW
Current - 28.74mA

Reading values from DSP_DVDDIO port

shunt Voltage -3.11mv
Bus Voltage - 1.79V
Power - 9.87mW
Current - 5.59mA

Reading values from VCC3V3_USB port

shunt Voltage -327.63mv
Bus Voltage - 0.02V
Power - 0.00mW
Current - 48.39mA

MSP432 INA Device Test Completed
```

A.1.2.3 CC2650 Interface Tests

A.1.2.3.1 DSP to CC2650 Interface Test

A.1.2.3.1.1 Test Accessories

- Mini USB Cable
- Two micro USB cables
- 14 to 10 pin JTAG converter
- XDS 100v2 emulator

A.1.2.3.1.2 Test Setup

1. Connect one micro USB cable with the BoosterPack's Debug port and the Test PC.
2. Connect the JTAG-CC2650 on the BoosterPack and Test PC with a mini USB cable using 14 to 10 pin converter and XDS 100v2 or XDS200 with CTI 20-pin to ARM 10-pin converter.
3. Power ON the BoosterPack.

A.1.2.3.1.3 Test Execution

This test verifies GPIO and SPI lines between CC2650 and C5545 DSP on BoosterPack.

1. CC2650 (from CCS):

   (a) Open CCS, launch the target configuration file for CC2650.

   (b) Connect to target, load and run the ble_dsp_interface_test.out program.

2. C5545 DSP (from the diagnostic serial console menu):
   (a) After the BLE firmware is flashed, select ‘DSP-BLE Interface Test’ from the DSP diagnostic test menu.
   (b) Check the CCS logs for CC2650 and on the serial console for DSP.

The sample log for the test on CC2650 is shown below:

```
**********************************
           BLE SPI Test
**********************************

Waiting to Read data from BP to BLE.....

read data 0x1
read data 0x2
read data 0x3
read data 0x4
 read data match rxbuffer
Write start:4 bytes

write data 0x5
write data 0x6
write data 0x7
write data 0x8

BLE Data Read and Transfer Test Passed!

BLE SPI Passed!
```

The sample log for the test on DSP is shown below:

```
************************************
       DSP-BLE Interface Test
************************************

Test verifies SPI & GPIO interfaces between BLE and DSP


Writing Four Bytes of Data to BLE
0x1     0x2     0x3     0x4

Write Successful

Waiting for Data from BLE

Data Read from BLE
0x5     0x6     0x7     0x8

Read data match with expected data

DSP-BLE Interface Test Passed!
```

A.1.2.3.2 MSP432 to CC2650 GPIO Test

A.1.2.3.2.1 Test Accessories

- MSP432 LaunchPad
- Mini USB Cable
- Two micro USB cables
- 14 to 10 pin JTAG converter
- XDS 100v2 emulator

A.1.2.3.2.2 Test Setup

1. Connect the MSP432 LP to the BoosterPack by using the Expansion connectors.
2. Connect one micro USB cable with the BoosterPack's Debug port and the Test PC, and the other micro USB cable with MSP432 LaunchPad and the Test PC.
3. Configure the UART jumpers on BoosterPack to enable the communication between CC2650 and LaunchPad:
   (a) JP2 – Short pin 2 and 4
   (b) JP3 – Short pin 2 and 4
4. Connect the JTAG-CC2650 on the BoosterPack and the Test PC with a mini USB cable using the 14- to 10-pin converter and XDS 100v2 or XDS200 with CTI 20-pin to ARM 10-pin converter.
5. Power ON the BoosterPack.

A.1.2.3.2.3 Test Execution

This test verifies GPIO lines between CC2650 on BoosterPack and MSP432 LaunchPad.

1. MSP432 LP (from CCS):
   (a) Open CCS and connect to MSP432 LaunchPad using the relevant target configuration file.
   (b) Load and run the gpio_launchpad_msp432_to_ble_test.out program on MSP432.
   (c) Press any key to continue the test after running the program on CC2650.
2. BoosterPack (from CCS):
   (a) Open CCS and connect to CC2650 using the relevant target configuration file.
   (b) Load and run the gpio_ble_to_msp432_lp_test.out program.
   (c) Observe the logs on both the CCS consoles.

The sample log for the test on CC2650 is shown below:

```
*******************************************************
          CC2650 BLE TO MSP432 LAUNCHPAD TEST
*******************************************************
Writting HIGH to the Uart_tx pin
Waiting for the Launchpad to write HIGH to the Uart_tx pin...
Yes, Uart_rx has read HIGH
Writting LOW to the Uart_tx pin
Waiting for the Launchpad to write LOW to Uart_tx pin...
Yes, Uart_rx has read LOW

GPIO BLE to LP Test Passed!
```

The sample log for the test on MSP432 is shown below:

```
******************************************
        Gpio Lp Msp432 To Ble Test
******************************************
Configured Uart_rx as input pins
Configured Uart_tx as output pins
Clearing the Uart_tx pin

Enter any character once DSP side code starting running
a
Waiting for DSP to write HIGH for Uart_tx pins...
Yes, Uart_tx pin is read as HIGH
Writting HIGH to the Uart_tx
Waiting for DSP to write LOW to Uart_tx pin...
Yes, Uart_rx pin read as LOW
Writting LOW to the Uart_tx


Gpio Lp Msp432 To Ble Completed!!
```

A.1.2.3.3 CC2650 UART Test

A.1.2.3.3.1 Test Accessories

- Mini USB Cable
- Micro USB cables
- 14 to 10 pin JTAG converter
- XDS 100v2 emulator
- Or, 100v2 or XDS200 with CTI 20-pin to ARM 10-pin converter

A.1.2.3.3.2 Test Setup

1. Connect the USB cable with the BoosterPack's Debug port and the Test PC.
2. Configure the UART jumpers on the BoosterPack to enable the communication between CC2650 and the serial port:
   (a) JP2 – Short pin 1 and 2
   (b) JP3 – Short pin 3 and 4
3. Connect JTAG-CC2650 on the the BoosterPack and Test PC with a mini USB cable using 14- to 10-pin converter and XDS 100v2.
4. Power ON the BoosterPack.
5. Open the serial console (example, Teraterm) on the host PC. Connect to the COM port where the BoosterPack Debug port is connected and setup for the following configurations:
   (a) Baud rate - 115200
   (b) Data length - 8 bit
   (c) Parity - None
   (d) Stop bits - 1
   (e) Flow control - None
6. Enable the local echo on the serial console.

A.1.2.3.3.3 Test Execution

This test verifies the CC26540 UART communication with the host PC.

1. Open CCS and connect to CC2650 using the relevant target configuration file.
2. Load and run the ble_uart_test.out program.
3. Check that the message 'CC2650 UART Test' is displayed on the serial console of the host PC.
4. Enter 10 characters on the serial console and check that the same is displayed on the CCS console.

The sample test log is shown below:

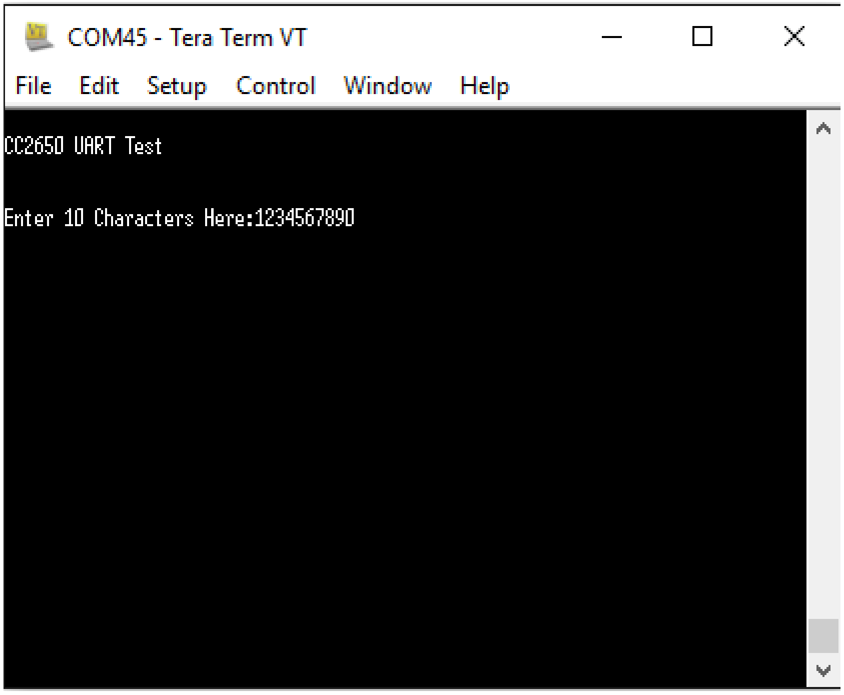
```
******************************
       CC2650 UART Test
******************************
Writing Data to Serial Console
Write Successful

Enter 10 Characters on the Serial Console
Waiting for Data from Serial Console...

Characters Received:1234567890

UART Test Passed!
```

The serial console output on the host PC is shown in Figure 18.



```
CC2650 UART Test

Enter 10 Characters Here:1234567890
```

Figure 18. Serial Console Output on Host PC

CCS Target Configurations

B.1 Connecting to C5545 DSP From CCS

1. Connect the micro USB cable to the DEBUG micro-USB port of BoosterPack and the host PC.
2. Open the CCS IDE and launch the target configuration for C5545.
3. Use the following steps to create the target configuration file, if needed.
   (a) Select the CCS menu 'File → New → Target Configuration File'.
   (b) Give a name to the target configuration file and click 'Finish'.
   (c) In the new target configuration file created:
      (i) Select 'Connection' as 'Texas Instruments XDS100v2 USB Debug Probe'.
      (ii) Select 'Board or Device' as TMS320C5545 and save the file.
4. Select 'View → Target Configurations' to open the Target Configurations window.
5. Right click on the C5545 Target Configuration file that is created and select 'Launch Selected Configuration'.
6. After target is launched, select 'Run → Connect Target' to connect to the target.
7. After target is connected, select 'Run → Load → Load Program' to load the program.
8. After program is loaded, select 'Run → Resume' to start running the program.

B.2 Connecting to CC2650 BLE From CCS

1. Connect the emulator to the JTAG-CC2650 port on the BoosterPack.
   (a) Pin converter may be needed while using the standard emulators with 14/20 pin headers. Check the references at the links below:
      http://www.spectrumdigital.com/14-pin-to-20-pin-cti-jtag-adapter
      http://www.spectrumdigital.com/cti20-pin-to-arm10-pin-jtag-adapter
2. Connect the micro USB cable to the DEBUG micro-USB port of the BoosterPack and host PC.
3. Open CCS IDE and launch the target configuration for CC2650.
   use the following steps to create the target configuration file, if needed:
   (a) Select the menu 'File → New → Target Configuration File'.
   (b) Give a name to the target configuration file and click 'Finish'.
   (c) In the new target configuration file created:
      (i) Select 'Connection' based on the emulator being used
      (ii) Select 'Board or Device' as CC2650F128 and save the file.
4. Select 'View → Target Configurations' to open the Target Configurations window.
5. Right click on the CC2650 Target Configuration file that is created and select 'Launch Selected Configuration'.
6. After target is launched, select 'Run → Connect Target' to connect to the target.
7. After target is connected, select 'Run → Load → Load Program' to load the program.
8. After program is loaded, select 'Run → Resume' to start running the program.

B.3 Connecting to MSP432 LP From CCS

1. Connect the micro USB cable to the USB port of MSP432 LaunchPad and the host PC.
2. Open CCS IDE and launch the target configuration for MSP432.
   Use the following steps to create the target configuration file, if needed:
   (a) Select the CCS menu 'File → New → Target Configuration File'.
   (b) Give a name to the target configuration file and click 'Finish'.
   (c) In the new target configuration file created:
      (i) Select 'Connection' as 'Texas Instruments XDS110 USB Debug Probe'.
      (ii) Select 'Board or Device' as MSP432P401R and save the file.
3. Select 'View → Target Configurations' to open the Target Configurations window.
4. Right click on the MSP432 Target Configuration file that is created and select 'Launch Selected Configuration'.
5. After target is launched, select 'Run → Connect Target' to connect to the target.
6. After target is connected, select 'Run → Load → Load Program' to load the program.
7. After program is loaded, select 'Run → Resume' to start running the program.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve 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 (also referred to herein as “components”) 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 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.

TI does not warrant or represent that any license, either express or implied, is granted under 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.

Reproduction of significant portions of TI information in TI data books or data sheets 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.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

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 which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm 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 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 a special agreement specifically governing such use.

Only those TI components which 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 which have ***not*** been so designated is solely at the Buyer's risk, and that 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.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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