Building Firmware For CC264BPA

User Guide

Revision 01

Overview

This user guide shows how existing Simplelink CC2640 projects (e.g. TI's sample project like (*simple_peripheral, key_fob, heart_rate, hid_emu_kbd*, etc) can be built for CC2640BPA module hardware under BLE SDK v2.2.1. It also shows how custom board files can be created for new hardware where new or existing projects can run on.

GT-tronics' CC2640BPA module is a very cost competitive, small, ultra-low power, and pre-certified BLE module using TI's state of the art Simplelink CC2640 SoC. Customers can easily drop this module in their design to enable BLE functionality without worrying the radio performance optimization and certification challenges that the chip level integration would normally be encountered. The modules also come in different development boards that help customers to evaluate and prototype their applications quickly.

The following information links pertain to CC264BPA and its related hardware:

CC2640BPA / CC2640BPA-S Modules

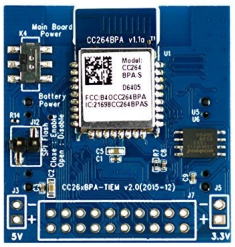
Online Order:

- http://makerspot.com/cc2640-bluetooth-low-energy-pre-certified-iot-wireless-module-5-pack/
- http://makerspot.com/cc2640-bluetooth-low-energy-iot-wireless-module-5-pack/

Data Sheet:

- http://www.gt-tronics.com/sites/default/files/website/downloads/CC26xBxA_Bluetooth_Smart_and_IoT_Module_DataSheet.pdf

CC2640BPA-TIEM Breakout Board (and EM for SmartRF06)



Online Order:

- http://makerspot.com/cc2640-bluetooth-low-energy-iot-wireless-module-evaluation-module/

User Guide Package:

- http://www.gt-tronics.com/sites/default/files/website/downloads/CC26xBPA-TIEM_User_Guide.zip

CC2640BPA-UDOG USB Dongle

Online Order:

- http://makerspot.com/cc2640-bluetooth-low-energy-ble-4-2-usb-hid-dongle-bluegiga-bled112/

User Guide Package:

- http://www.gt-tronics.com/sites/default/files/website/downloads/CC26xBPA-UDOG_User_Guide.zip

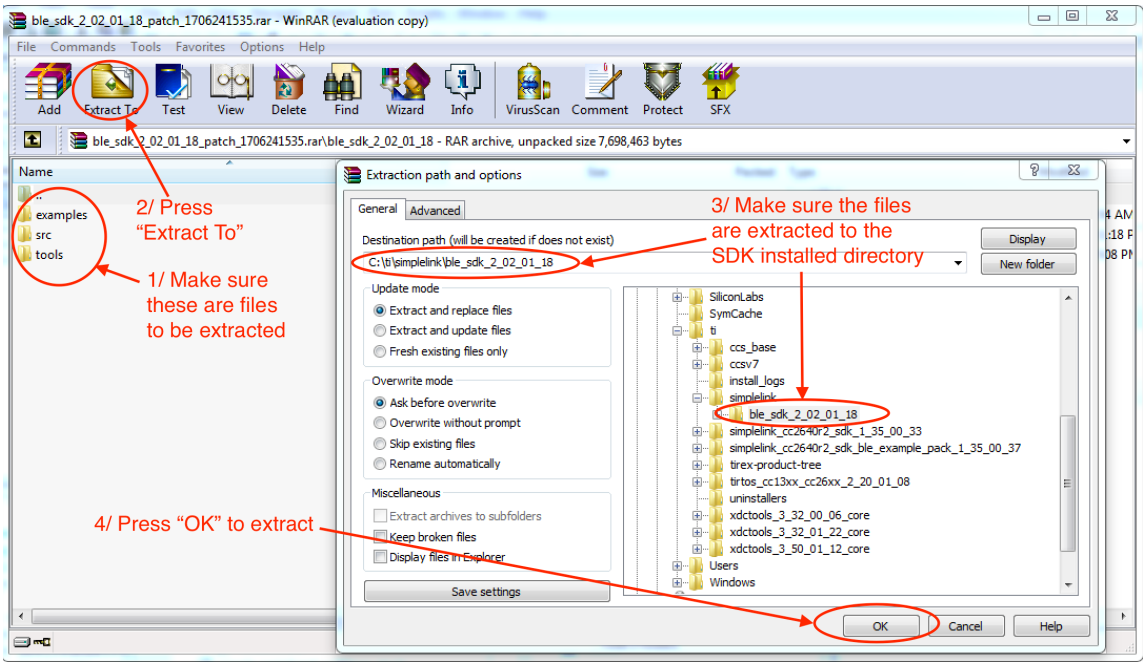
Preparation

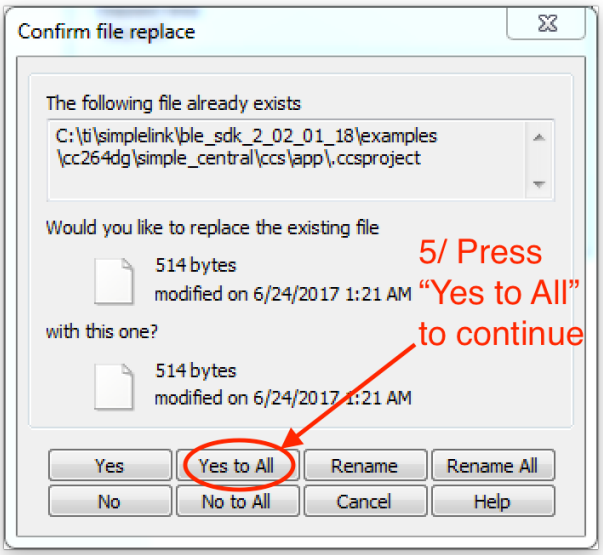
Prerequisites:

1. TI Simplelink CC2640 BLE SDK 2.2.1 installed
2. TI Code Composer Studio (CCS v7) installed.

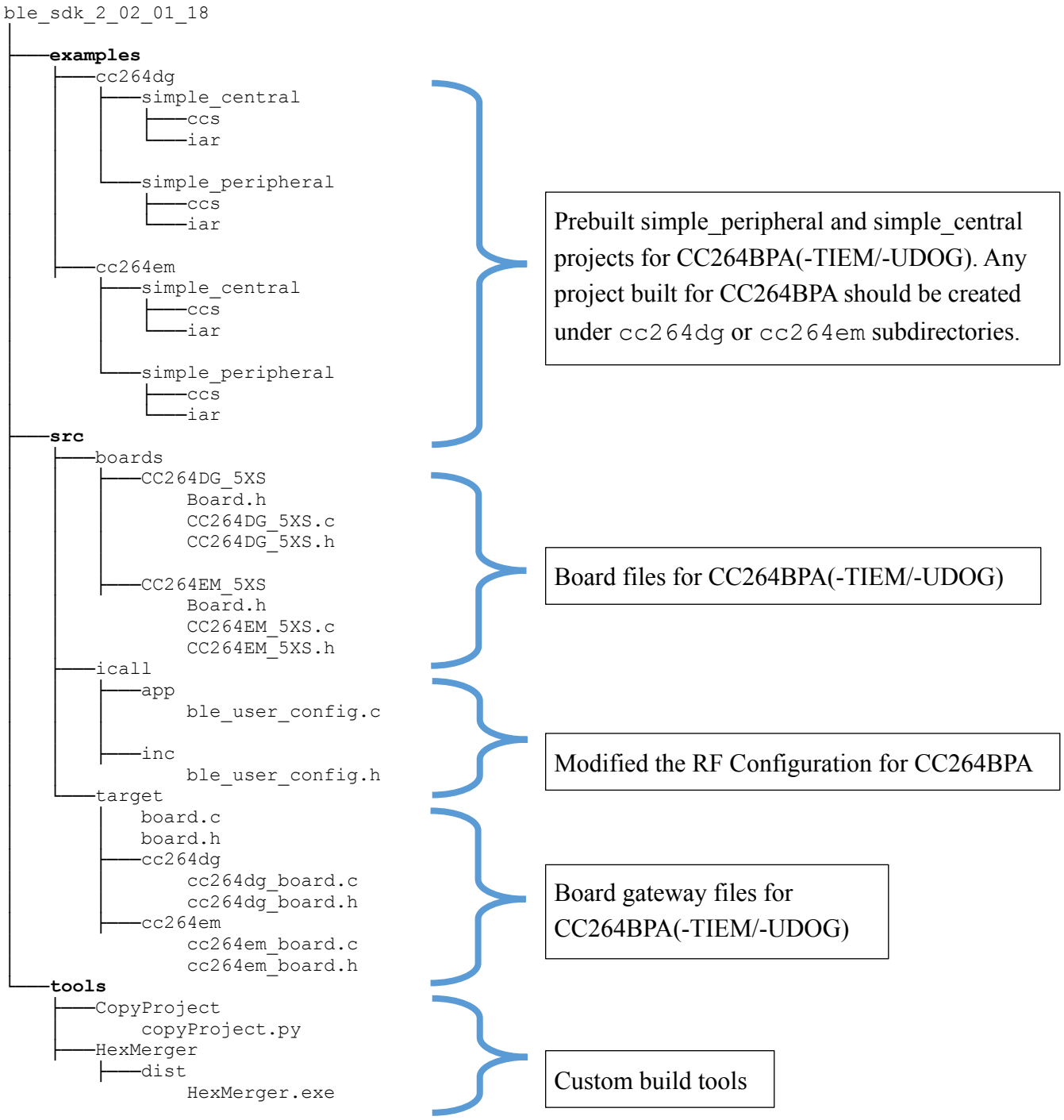
You can find those software links through TI's BluetoothLE Wiki.

After installing the prerequisites software, you can apply the patch file `ble_sdk_2_02_01_18_patch_<timestamp>.rar` to the installed SDK directory. You can find the patch file in CC264BPA-TIEM User Guide or CC264BPA-UDOG User Guide packages.





Understand The Patch File



Building Simple_Peripheral Project

Both *simple_peripheral* projects in `examples\cc264em` and `examples\cc264dg` are copied from the original `examples\cc2650em\simple_peripheral` and reconfigured for CC264BPA-TIEM and CC264BPA-UDOG. These projects are the examples of how users should create or port their own projects to run on CC264BPA hardware.

We create a `copyproject.py` python tool to facilitate the copying/porting process. This tool requires python 2.7 to run.

The following steps are used to port the *simple_peripheral* project from CC2650EM hardware to CC264BPA-TIEM hardware. The steps can be used to port other projects.

1. Start a DOS command window and execute the following in the window:
```
cd c:\ti\simplelink\ble_sdk_2_02_01_18\tools\CopyProject
python copyproject.py -r cc2650em=cc264em,CC2650DK_7ID=CC264EM_5XS c:\ti
\simplelink\ble_sdk_2_02_01_18\examples\cc2650em\simple_peripheral c:\ti
\simplelink\ble_sdk_2_02_01_18\examples\cc264em\simple_peripheral
```

2. Open CCS and import CCS project from `c:\ti\simplelink\ble_sdk_2_02_01_18\examples\cc264em\simple_peripheral`

3. In Project Properties->Build->ArmCompiler->AdvanceOptions->PredefinedSymbols, reconfigure or add the following symbols:
   - `xDisplay_DISABLE_ALL`
   - `BOARD_DISPLAY_EXCLUDE_LCD`
   - `xBOARD_DISPLAY_EXCLUDE_UART`

4. In simple_peripheral.c, replace `Display_Type_LCD` with `Display_Type_UART.`

5. Rebuild the stack project first, then the app project.

Explanations:

Step 1:

- The copyproject.py script can copy any existing project to run on CC264BPA hardware.
- The command format is:

```
python copyproject.py -r<replaceKeyWords> <copyFromDirectory>
<copyToDirectory>
```

- The “-r” option is to replace any keyword found in the file content and file name. In this case any `cc2650em` will be replaced with `cc264em`, and `CC2650DK_7ID` will be replaced with `CC264EM_5XS`.
- The keyword `CC264EM_5XS` is a predefined symbol for determining which board file (for CC264BPA-TIEM) to be used in building the firmware.

Step 2:

- Import CCS project

Step 3:

- CC264BPA(-TIEM/-UDOG) supports only UART display.
- `xDisplay_DISABLE_ALL` to enable the display subsystem
- `BOARD_DISPLAY_EXCLUDE_LCD` to disable LCD
- `xBOARD_DISPLAY_EXCLUDE_UART` to enable UART

Step 4:

- The default display configuration in *simple_peripheral* project is LCD. Need to change it to UART.

Step 5:

- Always build the stack project 1st before building the app project. However, once the stack project has been built once and has not been reconfigured, building stack project can be skipped.

Important Tips

1. Follow the same steps above to copy other projects (e.g. TI’s sample projects – *simple_central, key_fob, heart_rate, hid_emu_kbd*, etc) to run on CC264BPA hardware.
2. Replace `CC2650DK_7ID` (or equivalent if not copying from `examples\cc2650em`) with `CC264EM_5XS` or `CC264DG_5XS` for CC264BPA-TIEM or CC264BPA-UDOG hardware, respectively. This can be specified in `copyproject.py` command or replace it directly in the predefined symbol list.
3. Make sure the display subsystem is configured as per your project requirements.

Creating Custom Board Files

If you are using CC264BPA module or CC264BPA-TIEM and expanding it with new hardware components, likely you will want to create a new custom board files.

Follow the steps below to create a new set of gateway and board files for a new hardware called *CC264CU_5XS* (for example).

1. Use copyproject.py python script to create the new board gateway files
   ```
   cd c:\ti\simplelink\ble_sdk_2_02_01_18\tools\CopyProject
   python copyproject.py -r cc264em=cc264cu,CC264EM=CC264CU c:\ti
   \simplelink\ble_sdk_2_02_01_18\src\target\cc264em c:\ti\simplelink
   \ble_sdk_2_02_01_18\src\target\cc264cu
   ```
2. Use copyproject.py python script to create the new board files
   ```
   cd c:\ti\simplelink\ble_sdk_2_02_01_18\tools\CopyProject
   python copyproject.py -r cc264em=cc264cu,CC264EM=CC264CU c:\ti
   \simplelink\ble_sdk_2_02_01_18\src\boards\CC264EM_5XS c:\ti\simplelink
   \ble_sdk_2_02_01_18\src\boards\CC264CU_5XS
   ```
3. Modify the new board files according to your new hardware requirements
   a. Redefine the IO pin map (`CC264CU_5XS.h`)
   b. Redefine the initial pin configuration and state (`CC264CU_5XS.c`)
   c. Add new defines for the new hardware (`Board.h`)
   d. Enable/Disable the driver blocks (`CC264CU_5XS.c/h`)
4. Modify the `src/target/board.c` and `src/target/board.h` head board gateway files so the new board files can be included properly
   a. Add the following in `src/target/board.c`:
      ```
      #elif defined(CC264CU_5XS)
          #include "./cc264cu/cc264cu_board.c"
      ```
   b. Add the following in `src/target/board.h`:
      ```
      #elif defined(CC264CU_5XS)
          #include "./cc264cu/cc264cu_board.h"
      ```

Finally, create a new directory `examples/cc264cu` for existing (or new) projects. Then follow the previous section to copy projects for this new hardware.

Revision History

Rev.	Date	Description	By
01	2017-06-24	Initial release	ML

DISCLAIMER

The information disclosed to you hereinabove (the “Materials”) is provided solely for the selection and use of GT-tronics industrial, evaluation, and do-it-yourself products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, GT-tronics hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) GT-tronics shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or GT-tronics had been advised of the possibility of the same. GT-tronics assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of GT-tronics’s limited warranty, please refer to GT-tronics’s Terms of Sale which can be viewed at http://www.gt-tronics.com/terms.htm GT-tronics industrial, evaluation, and do-it-yourself products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of GT-tronics products in such critical applications, please refer to GT-tronics’s Terms of Sale which can be viewed at http://www.gt-tronics.com/terms.htm