Lab Exercises

Supported Targets:

- PSoC6 BLE Pioneer Kit

Can be easily ported to other PSoC 6 family devices.

Software:

ModusToolbox™ 2.1

Repository Link:

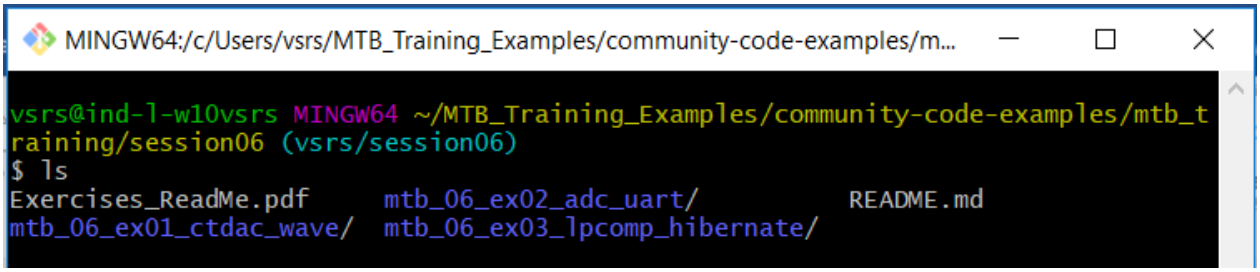
https://github.com/cypresssemiconductorco/Community-CodeExamples/tree/master/mtb_training/session06

Importing the exercises:

Here are the steps to import the projects into Eclipse IDE workspace

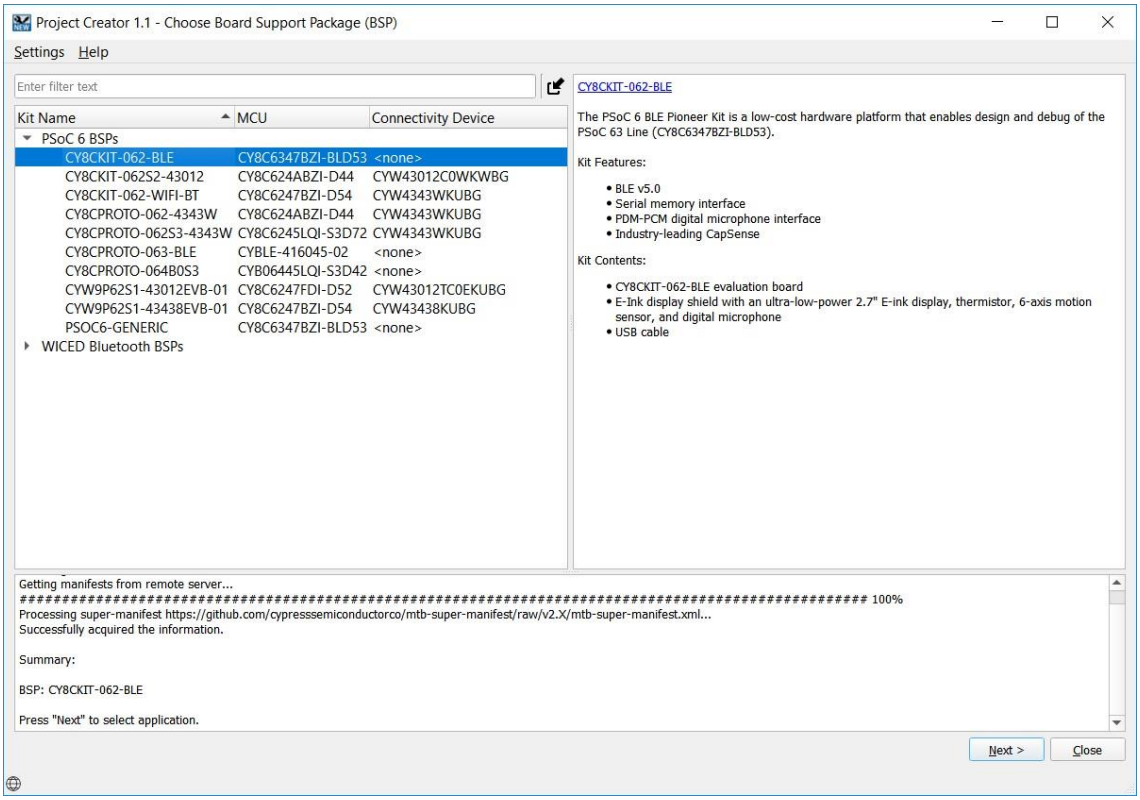
1. Clone the projects using the command: git clone https://github.com/cypresssemiconductorco/Community-CodeExamples.git

2. You will find all the exercises in the path *mtb_training/session06/* as shown:

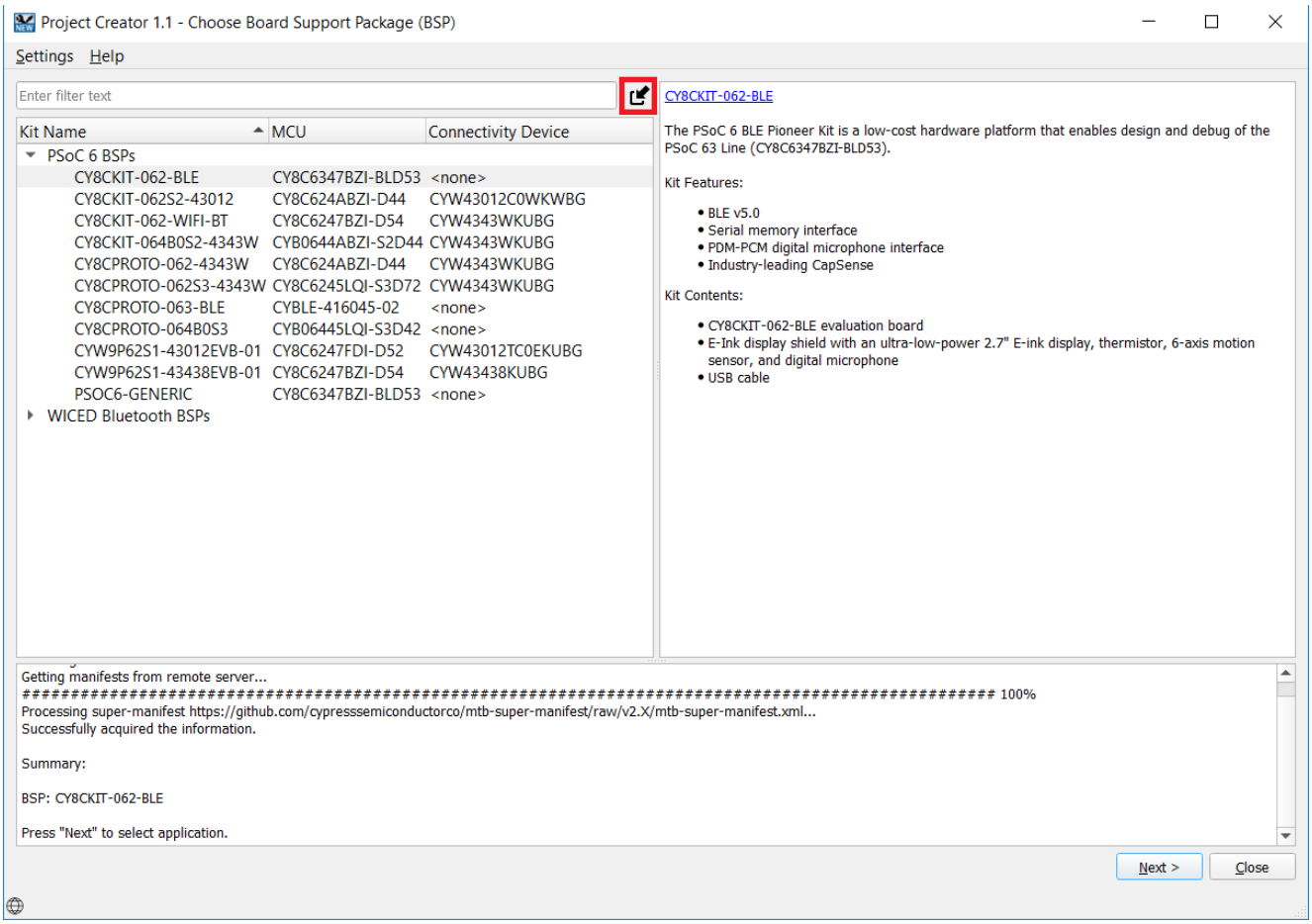


3. Open ModusToolbox and create a workspace folder of choice.

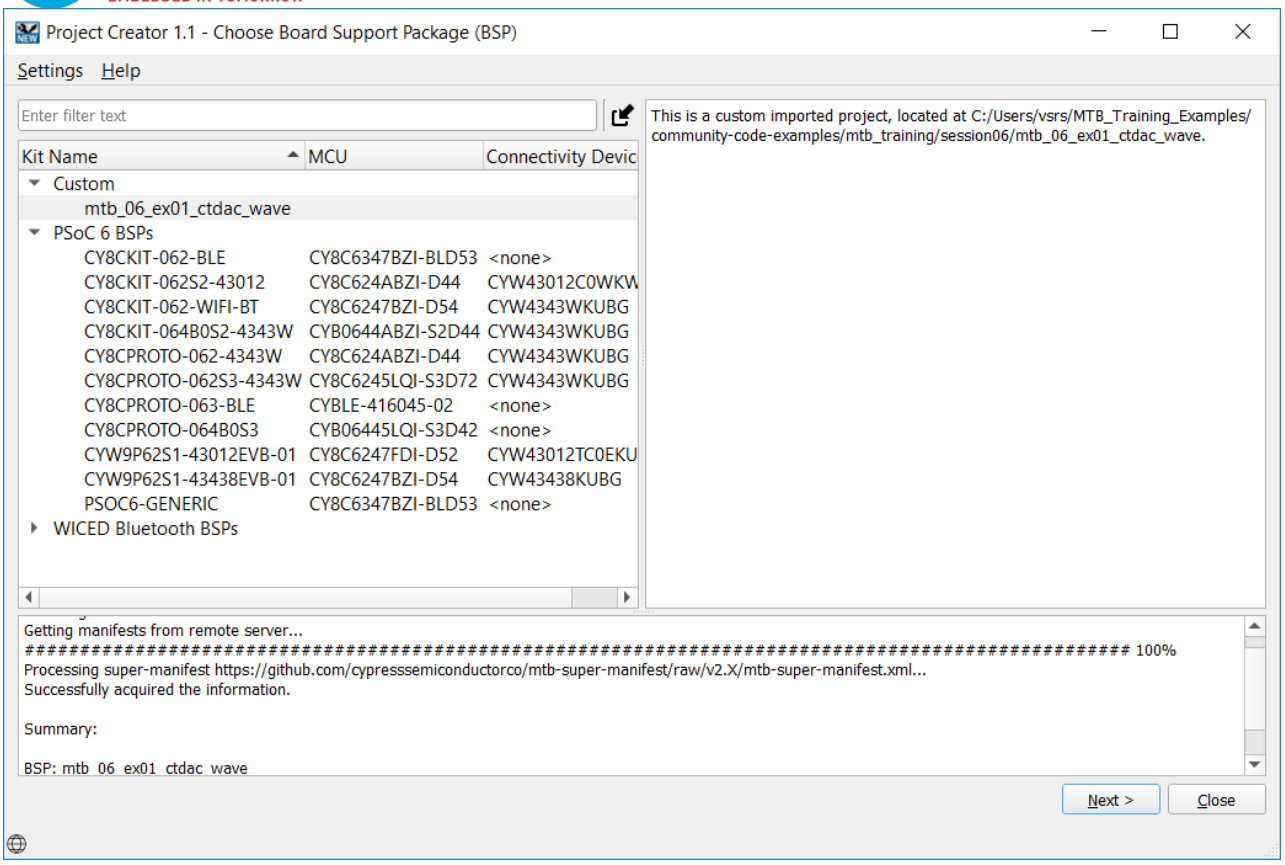
4. Once the Eclipse IDE is open, click **New Application > Choose BSP.**



5. Click the Import symbol as shown below:

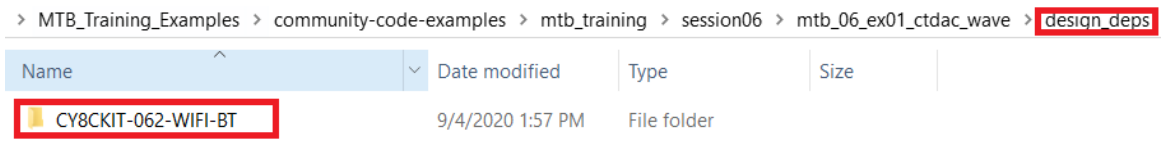


6. Point to the directory of the exercise containing the Makefile. Once selected you should see the name of the exercise visible under Application Name as shown below:

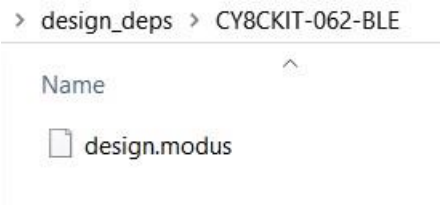


7. Click **Create** and then **Close**.

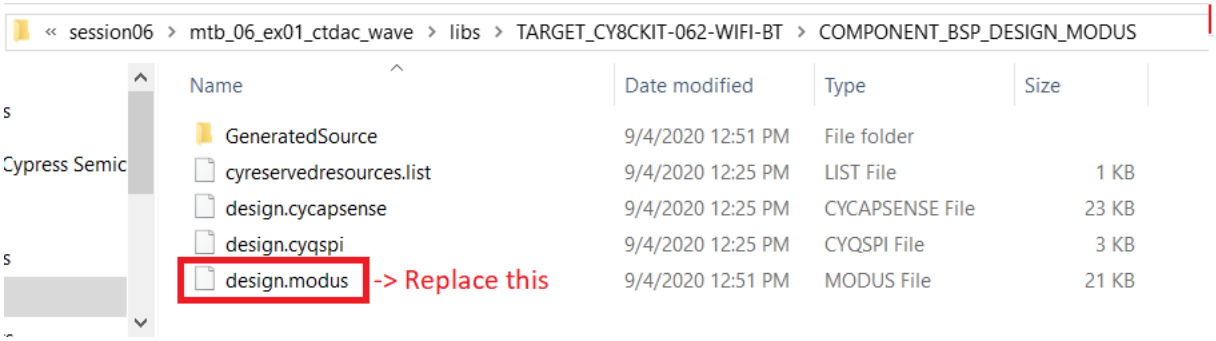
8. If the exercise you are working with uses PDL APIs and device configurator, then it will contain a folder named "*design_deps*". It contains the supported target folders. This step is not needed if device configurator is not used.



The target will contain a *design.modus* file as shown below.



9. Copy the *design.modus* file of the target board you are working with and paste it into the following directory:
<exercise_name>\libs\TARGET_<BOARD_NAME>\COMPONENT_BSP_DESIGN_MO DUS



10. Delete the GenerateSource folder in the path: *<exercise_name>\libs\TARGET_<BOARD_NAME>\COMPONENT_BSP_DESIGN_MODUS*

Running the exercises:

1. Open the Library Manager to choose the **Active BSP** and then click **Apply**.

2. Clean your application (mandatory step)

3. Build your application.

4. You can find the project description in the file *main.c* which explains the project and explains the pin connections and the expected output.

main.c

```
/* Project Description
*
* In this project, the CTDAC has been used to generate a sine wave
*
* (1) 12 bit unsigned sinewave input is stored in a lookup table is transferred using DMA
* (2) The CTDAC voltage reference is internal bandgap reference (1.2V)
* (3) The internal reference is buffered through CTBm block before being supplied to CTDAC
*
* To observe the outputs, connect pin 9[6] to an oscilloscope and see the waveform.
*
* Connections:
* P9[6] ---> CTDAC output pin

*
* Output: You should see a sine wave with amplitude ~1.2V(Internal Vref voltage)
*
* */

#include "cy_pdl.h"
#include "cyhal.h"
#include "cybsp.h"

/* Lookup table for a sine wave in unsigned format. */
uint32_t sineWaveLUT[] = {0x7FF, 0x880, 0x900, 0x97F, 0x9FC, 0xA78, 0xAF1, 0xB67, 0xBD9, 0xC48,
                          0xCB2, 0xD18, 0xD79, 0xDD4, 0xE29, 0xE77, 0xEC0, 0xF01, 0xF3C, 0xF6F,
                          0xF9A, 0xFBE, 0xFDA, 0xFEE, 0xFFA, 0xFFF, 0xFFA, 0xFEE, 0xFDA, 0xFBE,
                          0xF9A, 0xF6F, 0xF3C, 0xF01, 0xEC0, 0xE77, 0xE29, 0xDD4, 0xD79, 0xD18,
                          0xCB2, 0xC48, 0xBD9, 0xB67, 0xAF1, 0xA78, 0x9FC, 0x97F, 0x900, 0x880,
                          0x7FF, 0x77E, 0x6FE, 0x67F, 0x602, 0x586, 0x50D, 0x497, 0x425, 0x3B6,
                          0x34C, 0x2E6, 0x285, 0x22A, 0x1D5, 0x187, 0x13E, 0x0FD, 0x0C2, 0x08F,
                          0x064, 0x040, 0x024, 0x010, 0x004, 0x000, 0x004, 0x010, 0x024, 0x040,
```

5. Program the device.