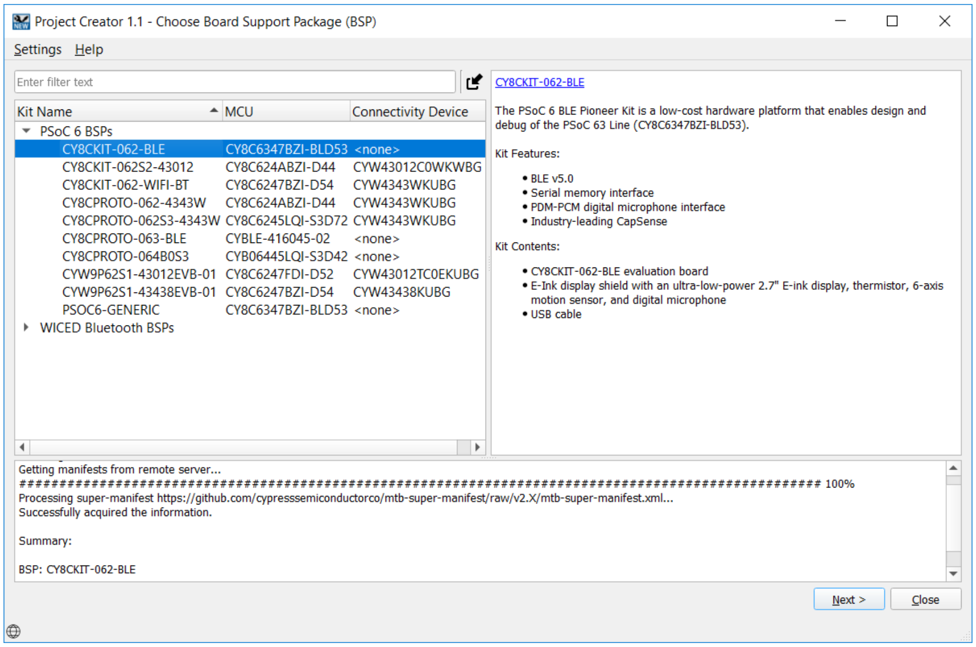
Creating a dual-core project in ModusToolbox 2.1

Creating the projects:

1. Create a new application using the project creator



Kit Name	MCU	Connectivity Device
PSoC 6 BSPs		
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	<none>
CY8CKIT-062S2-43012	CY8C624ABZI-D44	CYW43012C0WKWBG
CY8CKIT-062-WIFI-BT	CY8C6247BZI-D54	CYW4343WKUBG
CY8CPROTO-062-4343W	CY8C624ABZI-D44	CYW4343WKUBG
CY8CPROTO-062S3-4343W	CY8C6245LQI-S3D72	CYW4343WKUBG
CY8CPROTO-063-BLE	CYBLE-416045-02	<none>
CY8CPROTO-064B0S3	CYB06445LQI-S3D42	<none>
CYW9P62S1-43012EVB-01	CY8C6247FDI-D52	CYW43012TC0EKUBG
CYW9P62S1-43438EVB-01	CY8C6247BZI-D54	CYW43438KUBG
PSOC6-GENERIC	CY8C6347BZI-BLD53	<none>
WICED Bluetooth BSPs		

CY8CKIT-062-BLE

The PSoC 6 BLE Pioneer Kit is a low-cost hardware platform that enables design and debug of the PSoC 63 Line (CY8C6347BZI-BLD53).

Kit Features:

- BLE v5.0
- Serial memory interface
- PDM-PCM digital microphone interface
- Industry-leading CapSense

Kit Contents:

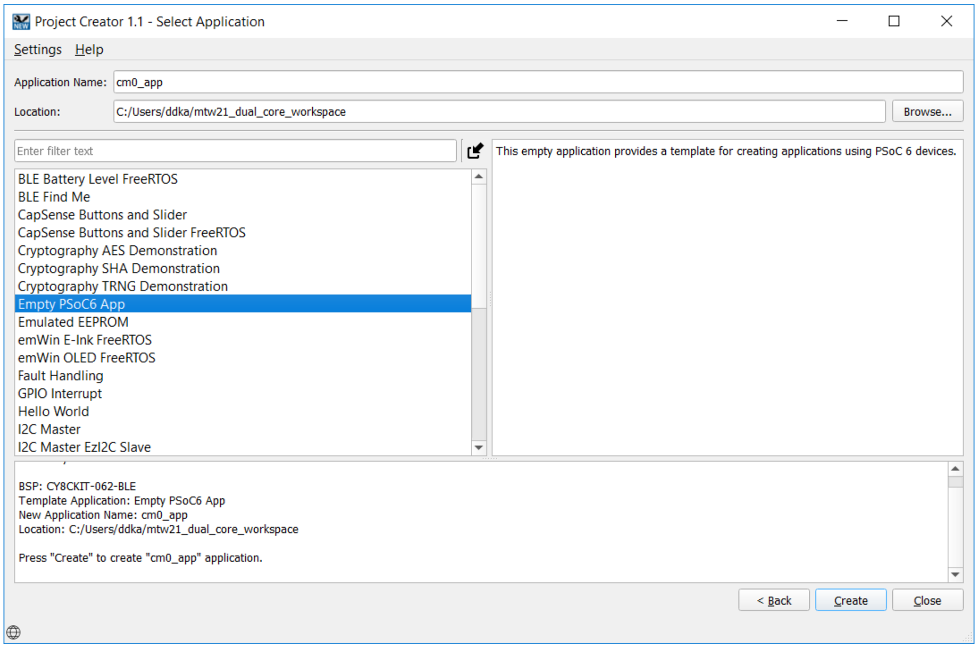
- CY8CKIT-062-BLE evaluation board
- E-Ink display shield with an ultra-low-power 2.7" E-ink display, thermistor, 6-axis motion sensor, and digital microphone
- USB cable

```
Getting manifests from remote server...
################################################################################################ 100%
Processing super-manifest https://github.com/cypresssemiconductorco/mtb-super-manifest/raw/v2.X/mtb-super-manifest.xml...
Successfully acquired the information.

Summary:

BSP: CY8CKIT-062-BLE
```

2. Choose “*Empty PSoC6 App*” Template, name the application as “*cm0p_app*” and click **Create**.



Application Name: cm0_app

Location: C:/Users/ddka/mtw21_dual_core_workspace

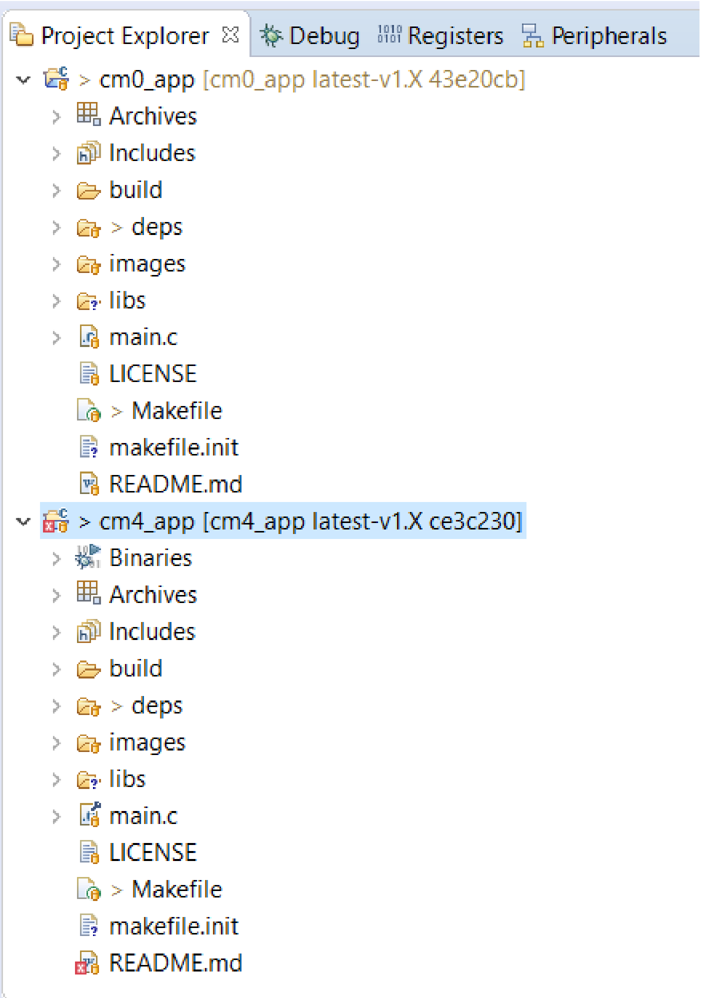
- BLE Battery Level FreeRTOS
- BLE Find Me
- CapSense Buttons and Slider
- CapSense Buttons and Slider FreeRTOS
- Cryptography AES Demonstration
- Cryptography SHA Demonstration
- Cryptography TRNG Demonstration
- Empty PSoC6 App
- Emulated EEPROM
- emWin E-Ink FreeRTOS
- emWin OLED FreeRTOS
- Fault Handling
- GPIO Interrupt
- Hello World
- I2C Master
- I2C Master EzI2C Slave

This empty application provides a template for creating applications using PSoC 6 devices.

```
BSP: CY8CKIT-062-BLE
Template Application: Empty PSoC6 App
New Application Name: cm0_app
Location: C:/Users/ddka/mtw21_dual_core_workspace

Press "Create" to create "cm0_app" application.
```

3. Repeat steps 1 and 2 to create another application called “*cm4_app*” using any application template.
4. You should see both the projects in the workspace as shown below:

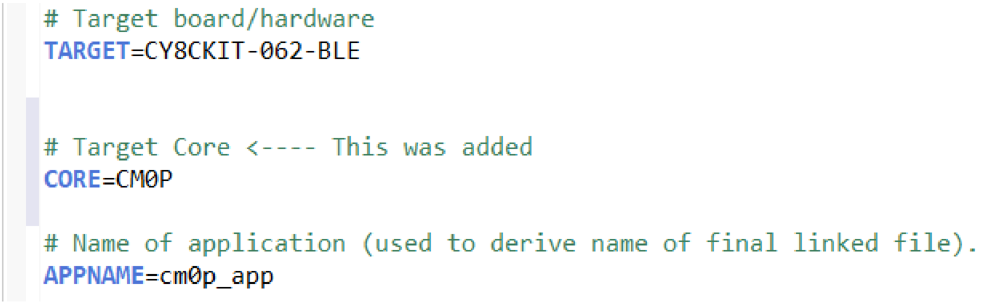


```
> cm0_app [cm0_app latest-v1.X 43e20cb]
    Archives
    Includes
    build
    > deps
    images
    libs
    main.c
    LICENSE
    > Makefile
    makefile.init
    README.md
> cm4_app [cm4_app latest-v1.X ce3c230]
    Binaries
    Archives
    Includes
    build
    > deps
    images
    libs
    main.c
    LICENSE
    > Makefile
    makefile.init
    README.md
```

Updating the CM0p Project:

1. Open main.c and remove the line

   #include “cyhal.h”
2. Open Makefile and change the following lines:
   - Add the line to target CM0p core:
     `CORE=CM0P`
   - Optionally change the APPNAME variable to make it easier to identify the cm0p_app as shown:
     `APPNAME=cm0p_app`



```
# Target board/hardware
TARGET=CY8CKIT-062-BLE


# Target Core <---- This was added
CORE=CM0P

# Name of application (used to derive name of final linked file).
APPNAME=cm0p_app
```

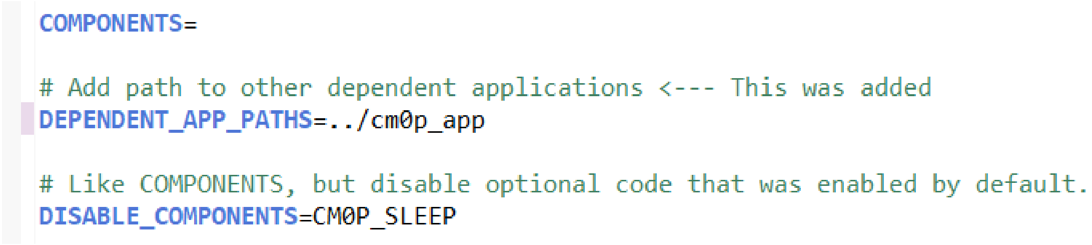
3. Update the linker script found in the path: *cm0_app\libs\TARGET_CY8CKIT-062-BLE\COMPONENT_CM0P\TOOLCHAIN_GCC_ARM.*
   The values changed for ram and flash regions are highlighted below:

```
    ram                 (rwx)   : ORIGIN = 0x08000000, LENGTH = 0x2000
   flash                (rx)    : ORIGIN = 0x10000000, LENGTH = 0x8000
```

4. Add your application code and then build your project. You cannot make use of HAL, so use only PDL APIs.

Updating the CM4 Project:

1. Open the Makefile and change:
   - Add “CM0P_SLEEP” to the DISABLE_COMPONENTS list.
     Eg: `DISABLE_COMPONENTS=CM0P_SLEEP`
   - Add a reference to the cm0p_app:
     Eg: `DEPENDENT_APP_PATHS=<relative path to cm0p_app application>`



```
COMPONENTS=

# Add path to other dependent applications <--- This was added
DEPENDENT_APP_PATHS=../cm0p_app

# Like COMPONENTS, but disable optional code that was enabled by default.
DISABLE_COMPONENTS=CM0P_SLEEP
```

2. Update the linker script:
   - Make sure to include the new cm0p_app image:
     Eg: `KEEP(*(.cm0p_app))`
     Note: the name is what is set as the APPNAME variable in the cm0p_app Makefile.
   - Also, update the value of FLASH_CM0P_SIZE based on cm0p_app image

```
 /* The size of the Cortex-M0+ application image at the start of FLASH
*/
 FLASH_CM0P_SIZE  = 0x8000;
```

3. Build the CM4 project.