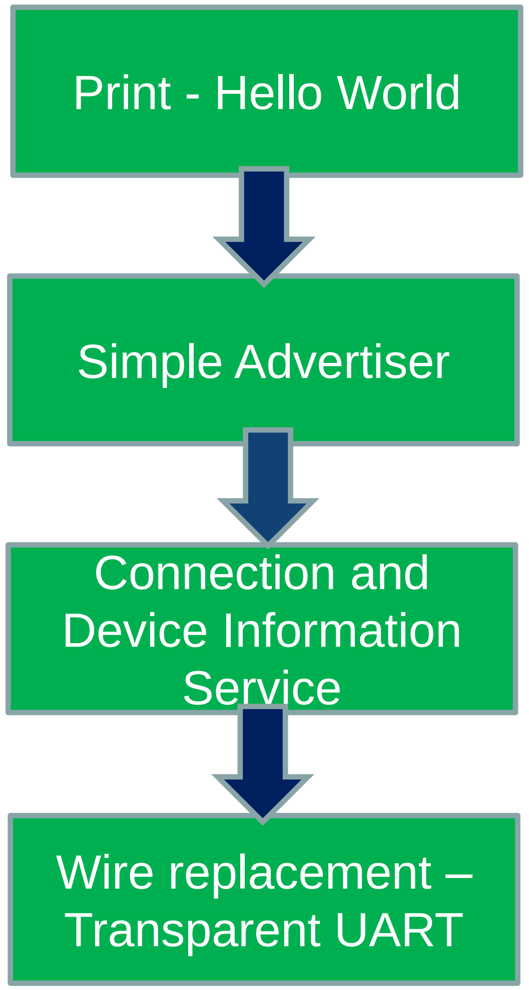
MICROCHIP
A Leading Provider of Microcontroller, Security,
Mixed-Signal, Analog & Flash-IP Solutions

Building a new BLE Application using MPLAB® Harmony v3

Goal

- Overview of Tools and Development Environment required for WBZ451/PIC32CXBZ2 devices
- Create BLE Application From Scratch
- 4 Tutorials created to help create app example that prints “Hello World” to BLE Transparent UART example

Building a new BLE App

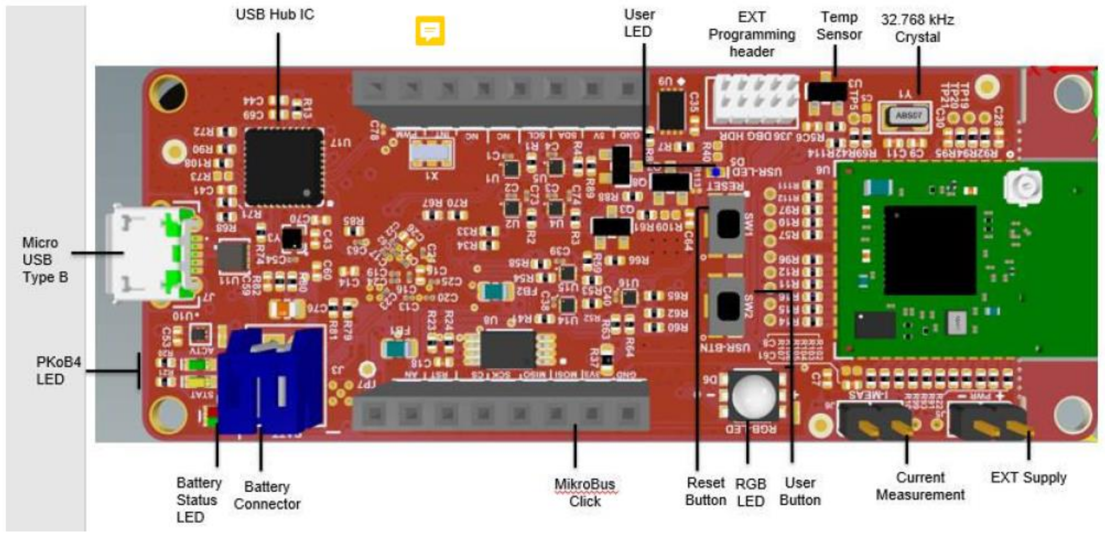

Print - Hello World
Simple Advertiser
Connection and Device Information Service
Wire replacement – Transparent UART

Building a new BLE App

- Development Environment
- Components of SDK
- Building a new project from scratch

Development Environment

- **Hardware**
  - WBZ451 Curiosity Board
    - External Customer Demo and Development
    - Build-in PKoB4 Program and Debug interface
    - Powered by USB or battery
    - Peripheral/Interface available for demo purpose



Development Environment

- **Software Prerequisites**
  - Integrated Development Environment
    - MPLAB X v5.50, XC32 v3.01
    - Harmony 3 Configurator plug-in
    - Microchip.PIC32CX-BZ_DFP-1.0.80 (Part Pack)
    - **Complete steps mentioned in PIC32CX-BZWBZ45x SDK Setup.pdf**

Component of SDK

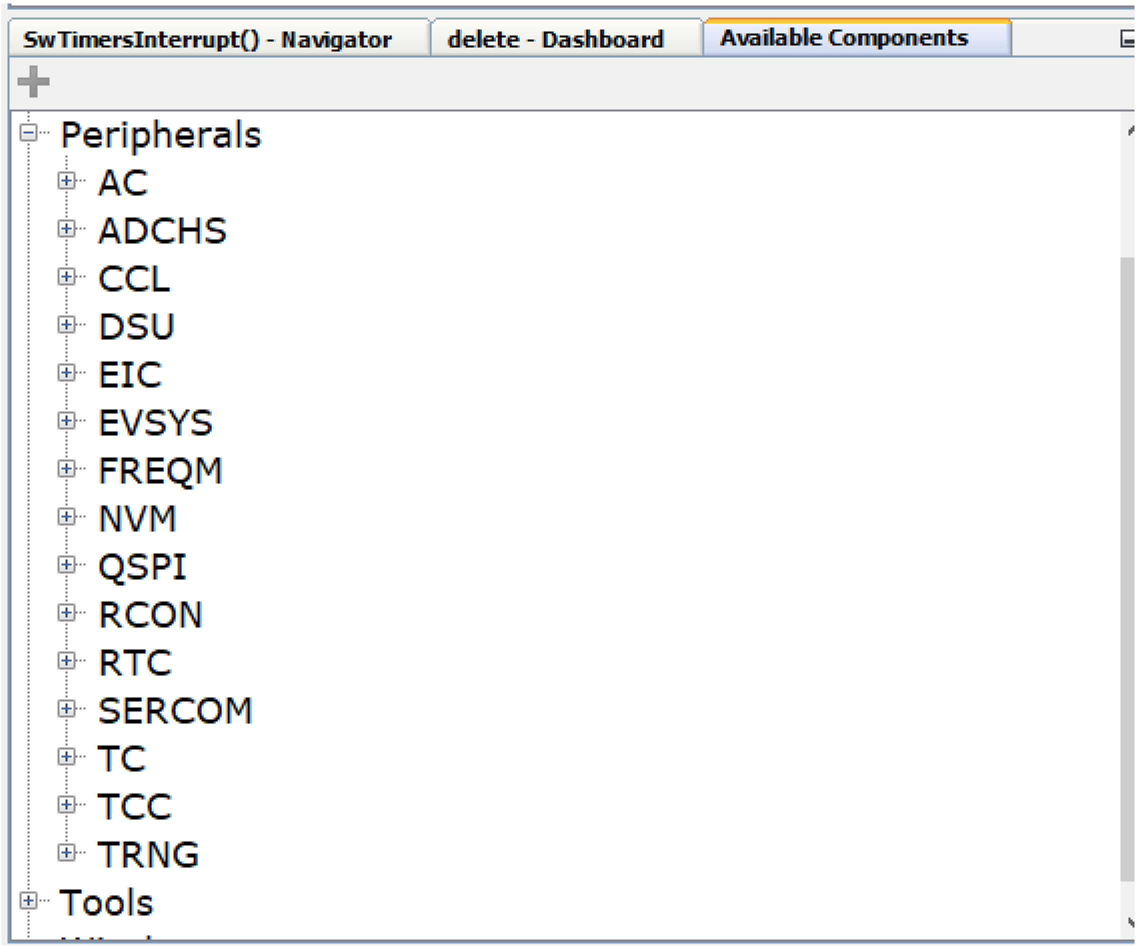
- **PIC32CX1012BZ25048/32 SDK consists of**
  - Real-Time Operating System
    - Multitasking
  - BLE Stack
    - Provided as library with documented interface
  - BLE Middleware
    - Provide easy access to BLE stack
  - BLE Build-in Services and Profiles
    - General purpose functions

Component of SDK

- **PIC32CX1012BZ25048/32 SDK consists of (cont…)**
  - RF Subsystem
    - Calibration and IB access
  - Software Framework
    - Interaction between application and BLE stack
  - Peripheral Libraries
    - Drivers for peripherals (Plibs)
  - Application
    - Partially generated by Harmony 3, implemented by user

Supported Peripheral

- **List of Supported Peripheral Drivers**



Peripherals
- AC
- ADCHS
- CCL
- DSU
- EIC
- EVSYS
- FREQM
- NVM
- QSPI
- RCON
- RTC
- SERCOM
- TC
- TCC
- TRNG

Tools

Shared Hardware Resources

- **FreeRTOS**
  - SysTick
- **BLE**
  - No Shared Resources with Application

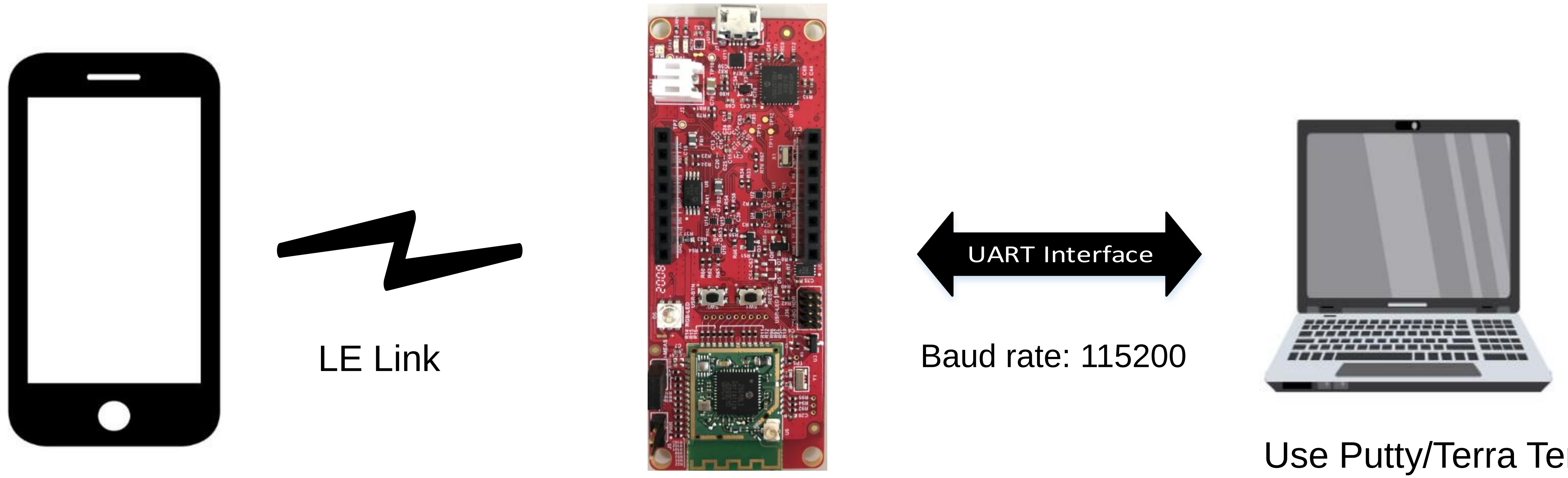
Interrupt Priorities

- **BLE and Zigbee Components use interrupt**
  - Used Interrupt Priority
    - FreeRTOS: 7
    - BLE: 4
    - Zigbee: 4
    - Arbiter: 1
  - Suggested Application Interrupt priority
    - USART: 3
    - Other priority: 7 or above

Harmony 3 Project

- **Plug in your Curiosity Board via USB**
- **This project takes you from ‘hello world’ to transparent UART in 4 steps**
- **Software Prerequisites**
  - Integrated Development Environment
    - MPLAB X v5.50, XC32 v3.01
    - Harmony 3 Configurator plug-in
    - Microchip.PIC32CX-BZ_DFP-1.0.80 (Part Pack)
    - Complete steps mentioned in PIC32CX-BZWBZ45x SDK Setup.pdf
- **Mobile App and Terminal Emulator:**
  - Android: LightBlue
  - iOS: LightBlue
  - PC serial terminal: Putty, Terra Term

HW Setup



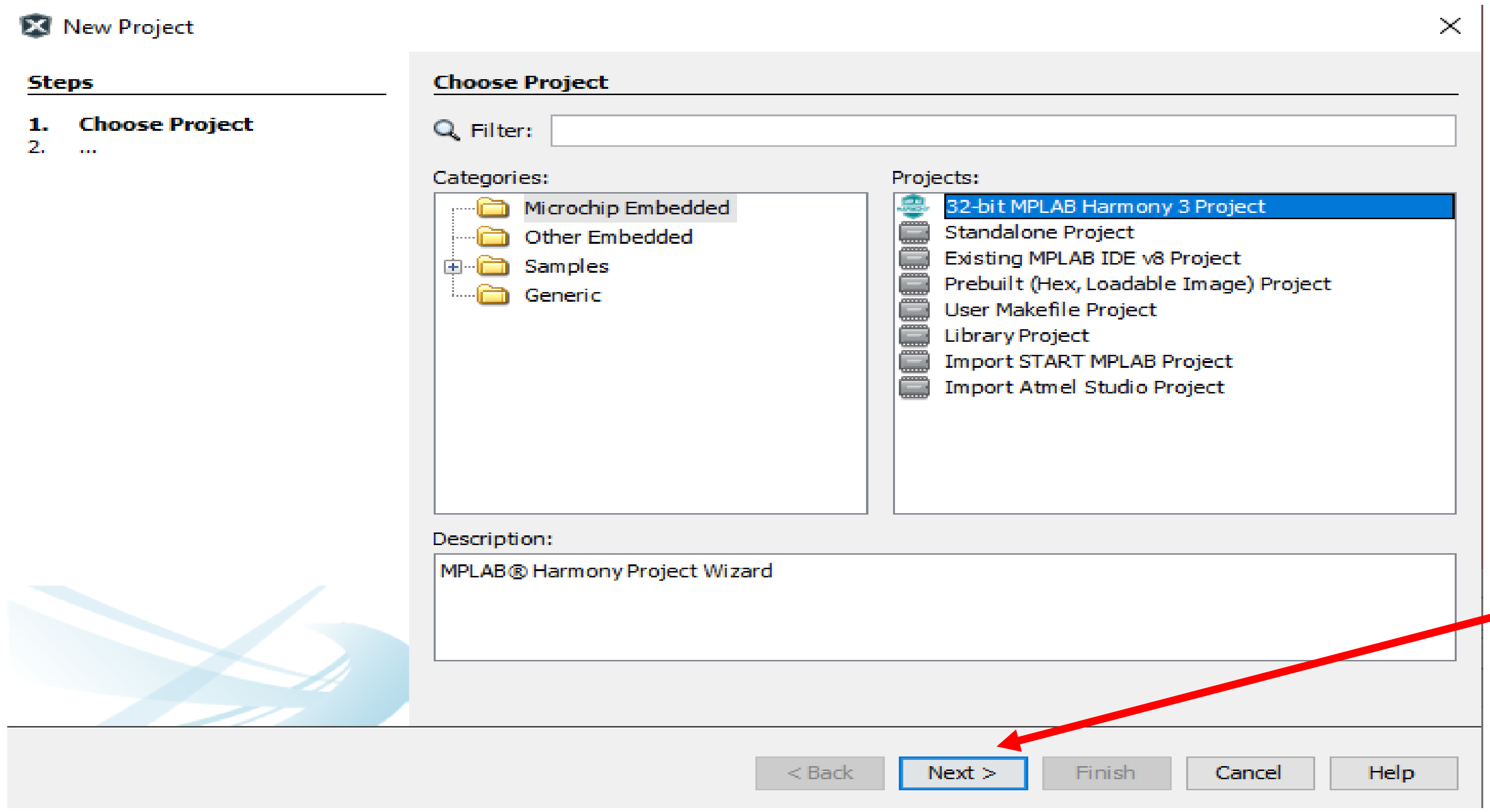
Baud rate: 115200

Android: LightBlue
iOS: LightBlue

Use Putty/Terra Term, other serial port emulator

New Project

- **File -> New Project -> 32-bit MPLAB Harmony 3 Project**



New Project

Steps

1. **Choose Project**
2. ...

Choose Project

Filter:

Categories:	Projects:
Microchip Embedded	32-bit MPLAB Harmony 3 Project
Other Embedded	Standalone Project
Samples	Existing MPLAB IDE v8 Project
Generic	Prebuilt (Hex, Loadable Image) Project
	User Makefile Project
	Library Project
	Import START MPLAB Project
	Import Atmel Studio Project

Description:

MPLAB® Harmony Project Wizard

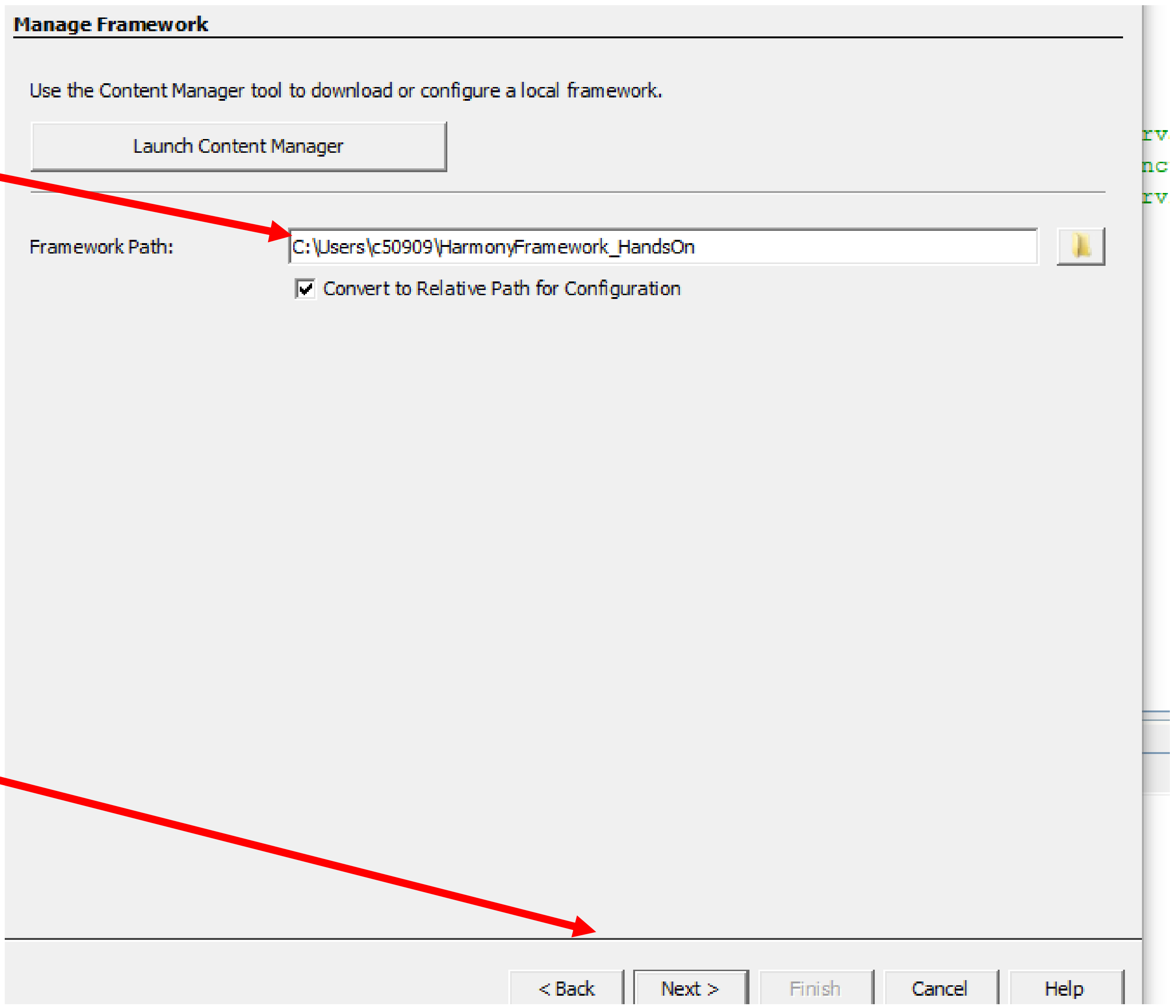
Then ‘Next>’

New Project

- **Framework Path**

(from previous setup)

Then ‘Next >’



Manage Framework

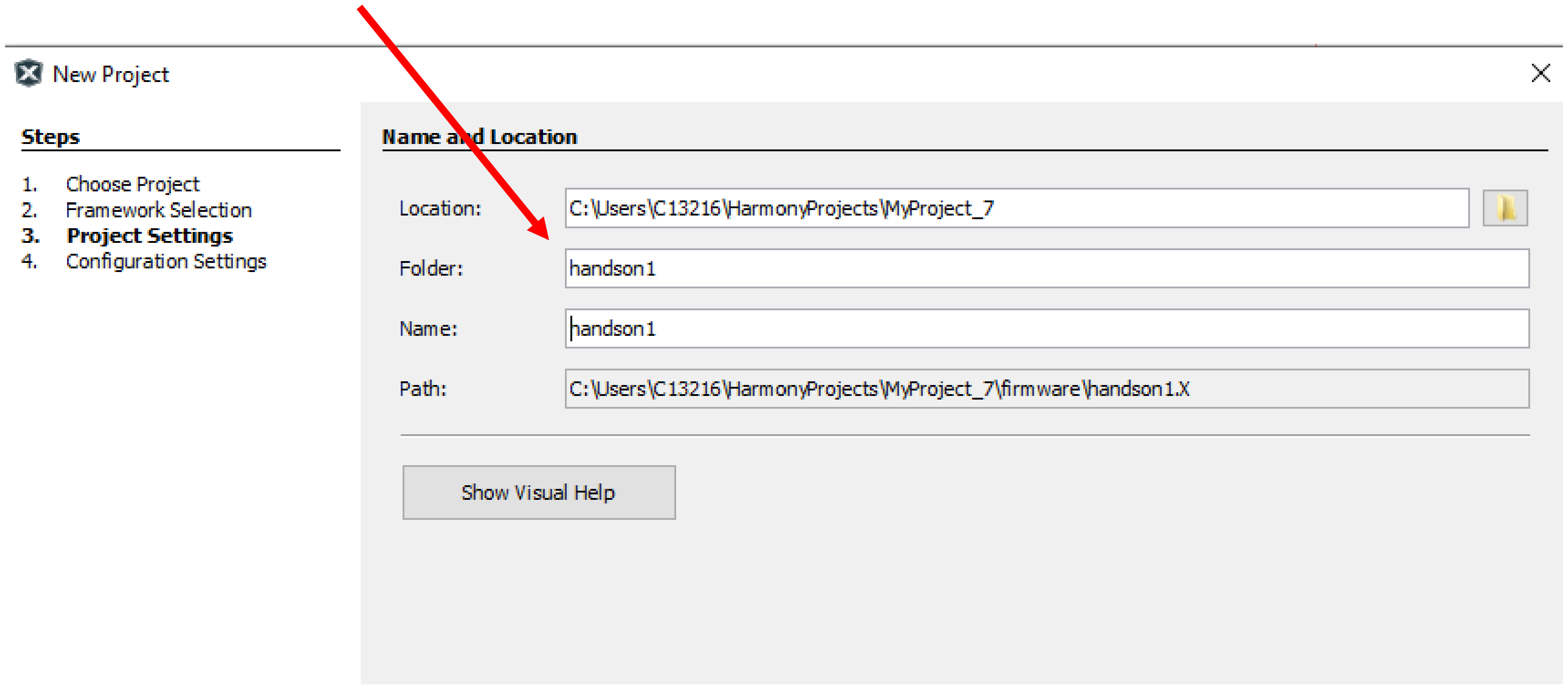
Use the Content Manager tool to download or configure a local framework.

Framework Path: C:\Users\c50909\HarmonyFramework_HandsOn

☑ Convert to Relative Path for Configuration

New Project

- **Select Project Folder Name, then 'Next'**



Location: C:\Users\C13216\HarmonyProjects\MyProject_7

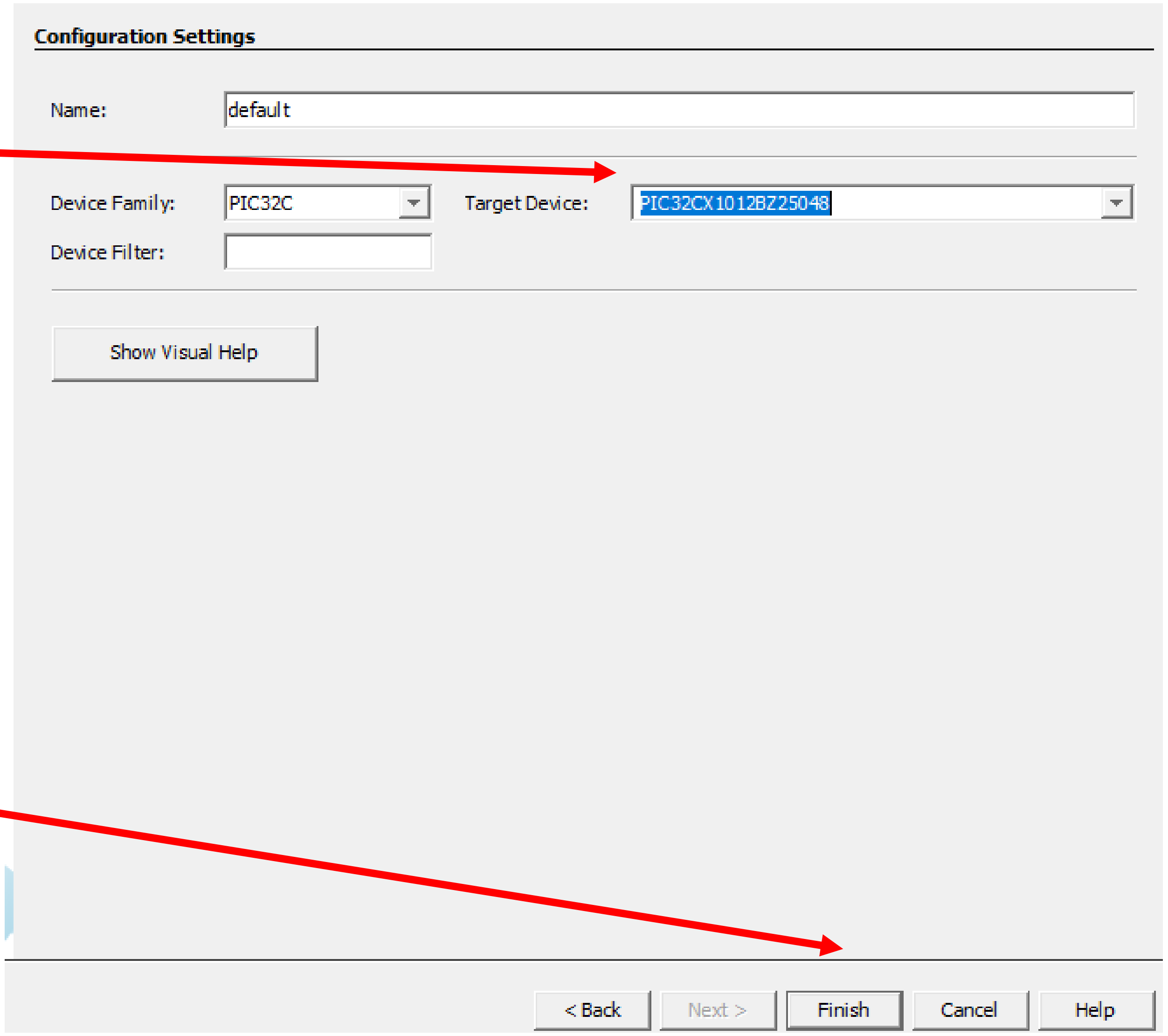
Folder: handson1

Name: handson1

Path: C:\Users\C13216\HarmonyProjects\MyProject_7\firmware\handson1.X

New Project

- **Select Device**



Configuration Settings

Name: default

Device Family: PIC32C

Target Device: PIC32CX1012BZ25048

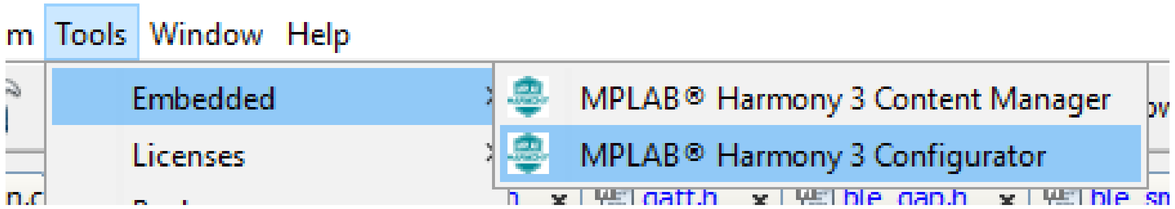
Device Filter:

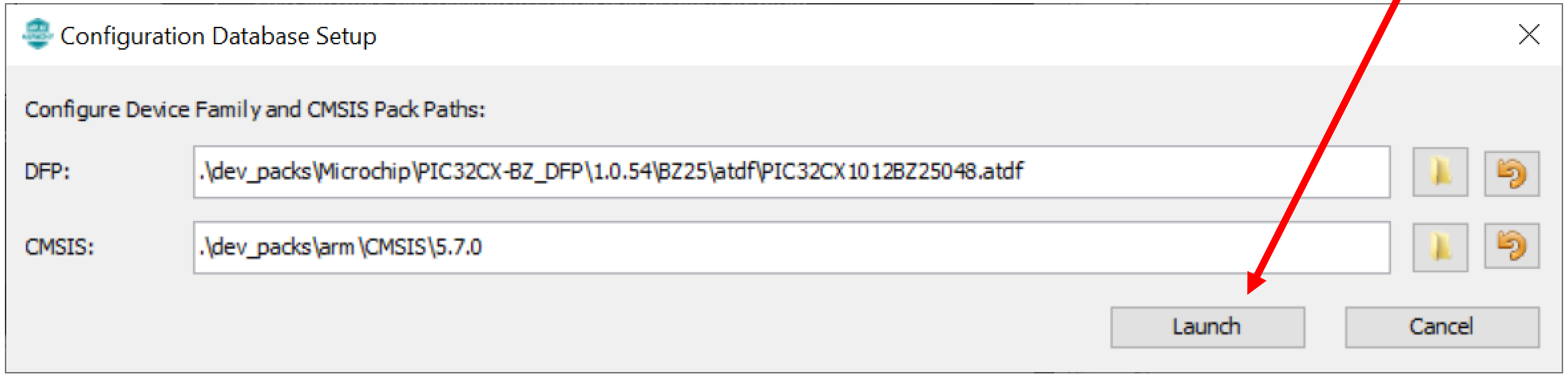
Then 'Finish'

Harmony 3 Configurator

- **GUI interface to help user**
  - Configure the components
  - Automatic apply dependency on modules
  - Automatic generate partial application code

Hit 'Launch'


m
Tools
Window
Help
Embedded
Licenses
MPLAB® Harmony 3 Content Manager
MPLAB® Harmony 3 Configurator


Configuration Database Setup

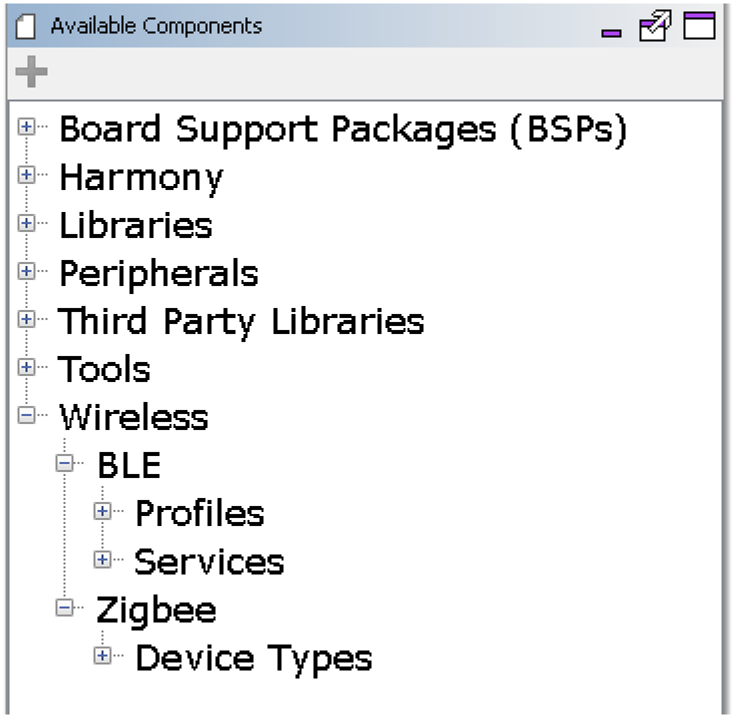
Configure Device Family and CMSIS Pack Paths:

DFP: .\dev_packs\Microchip\PIC32CX-BZ_DFP\1.0.54\BZ25\atdf\PIC32CX1012BZ25048.atdf

CMSIS: .\dev_packs\arm\CMSIS\5.7.0

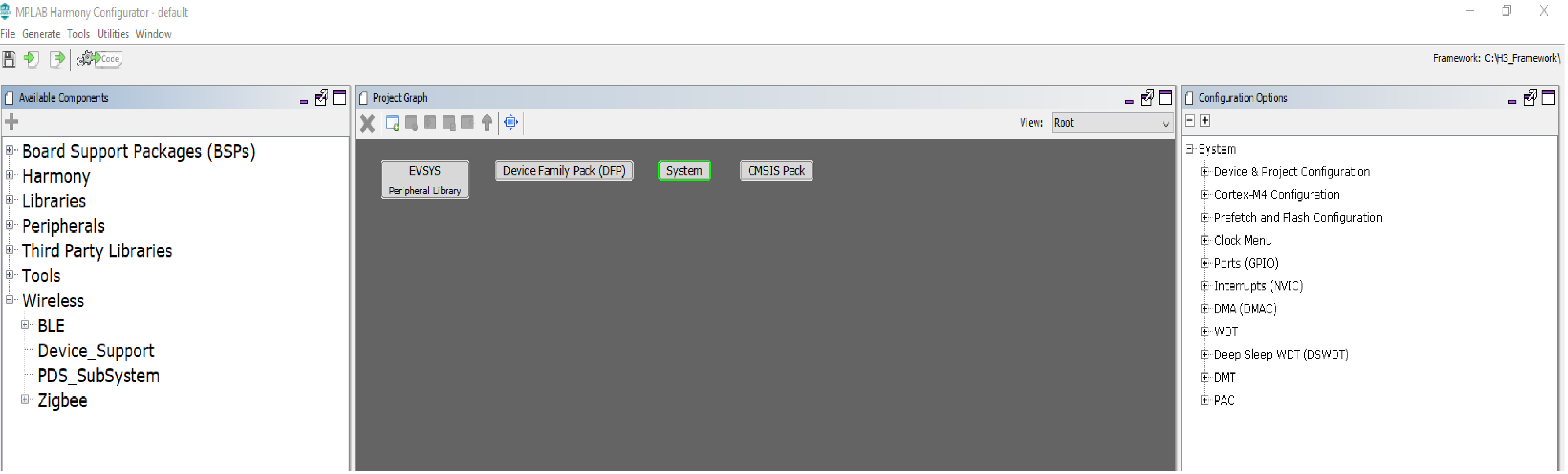
Harmony 3 Configurator

- **Available Components**
  - FreeRTOS
  - BLE_Stack
  - Profiles
  - Services
  - Peripherals
- **Drag or double-click component to add**


Available Components

- Board Support Packages (BSPs)
- Harmony
- Libraries
- Peripherals
- Third Party Libraries
- Tools
- Wireless
  - BLE
    - Profiles
    - Services
  - Zigbee
    - Device Types

Harmony 3 Configurator


MPLAB Harmony Configurator - default

File Generate Tools Utilities Window

Framework: C:\H3_Framework\

Available Components
- Board Support Packages (BSPs)
- Harmony
- Libraries
- Peripherals
- Third Party Libraries
- Tools
- Wireless
  - BLE
  - Device_Support
  - PDS_SubSystem
  - Zigbee

Project Graph

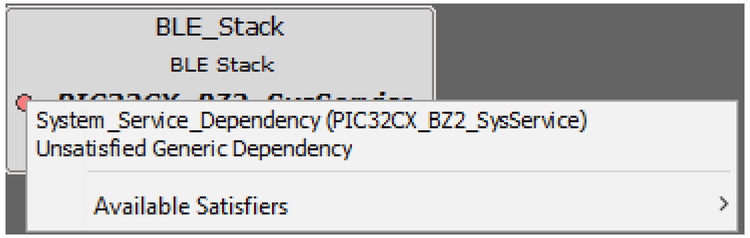
View: Root

EVSYS Peripheral Library | Device Family Pack (DFP) | System | CMSIS Pack

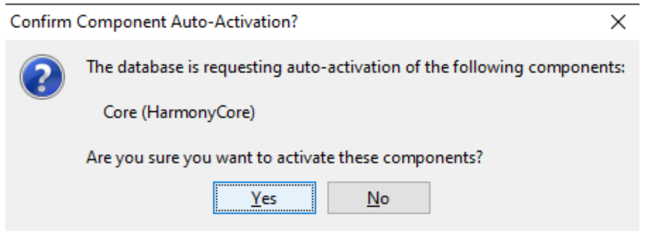
Configuration Options
- System
  - Device & Project Configuration
  - Cortex-M4 Configuration
  - Prefetch and Flash Configuration
  - Clock Menu
  - Ports (GPIO)
  - Interrupts (NVIC)
  - DMA (DMAC)
  - WDT
  - Deep Sleep WDT (DSWDT)
  - DMT
  - PAC

Harmony 3 Configurator

- **Try Drag BLE_Stack from “Available Component” to “Project Graph”**
  - Right click red dot PIC32CX_BZ2_SysService to add dependency


BLE_Stack
BLE Stack
System_Service_Dependency (PIC32CX_BZ2_SysService)
Unsatisfied Generic Dependency
Available Satisfiers

  - Add components with dependency

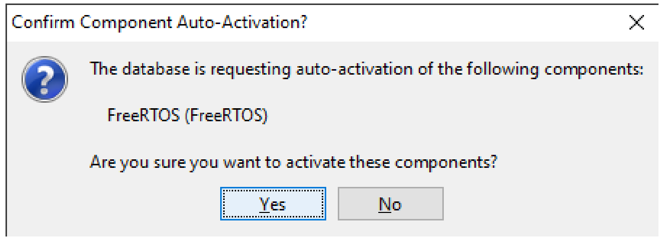

Confirm Component Auto-Activation?

The database is requesting auto-activation of the following components:

Core (HarmonyCore)

Are you sure you want to activate these components?

Yes No

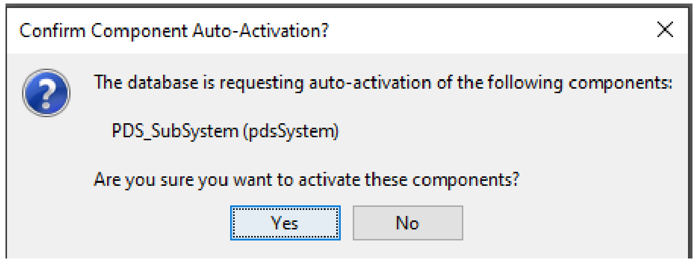

Confirm Component Auto-Activation?

The database is requesting auto-activation of the following components:

FreeRTOS (FreeRTOS)

Are you sure you want to activate these components?

Yes No

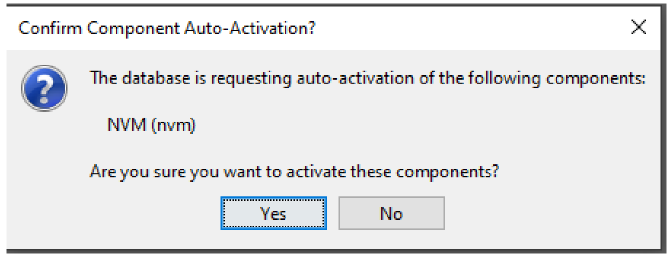

Confirm Component Auto-Activation?

The database is requesting auto-activation of the following components:

PDS_SubSystem (pdsSystem)

Are you sure you want to activate these components?

Yes No

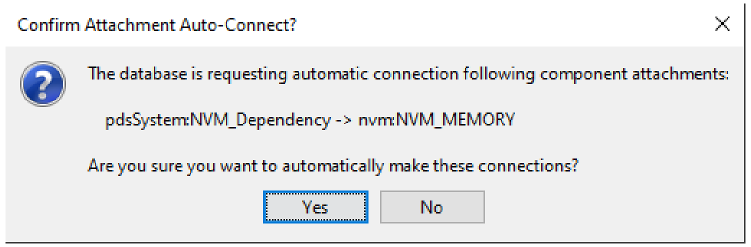

Confirm Component Auto-Activation?

The database is requesting auto-activation of the following components:

NVM (nvm)

Are you sure you want to activate these components?

Yes No


Confirm Attachment Auto-Connect?

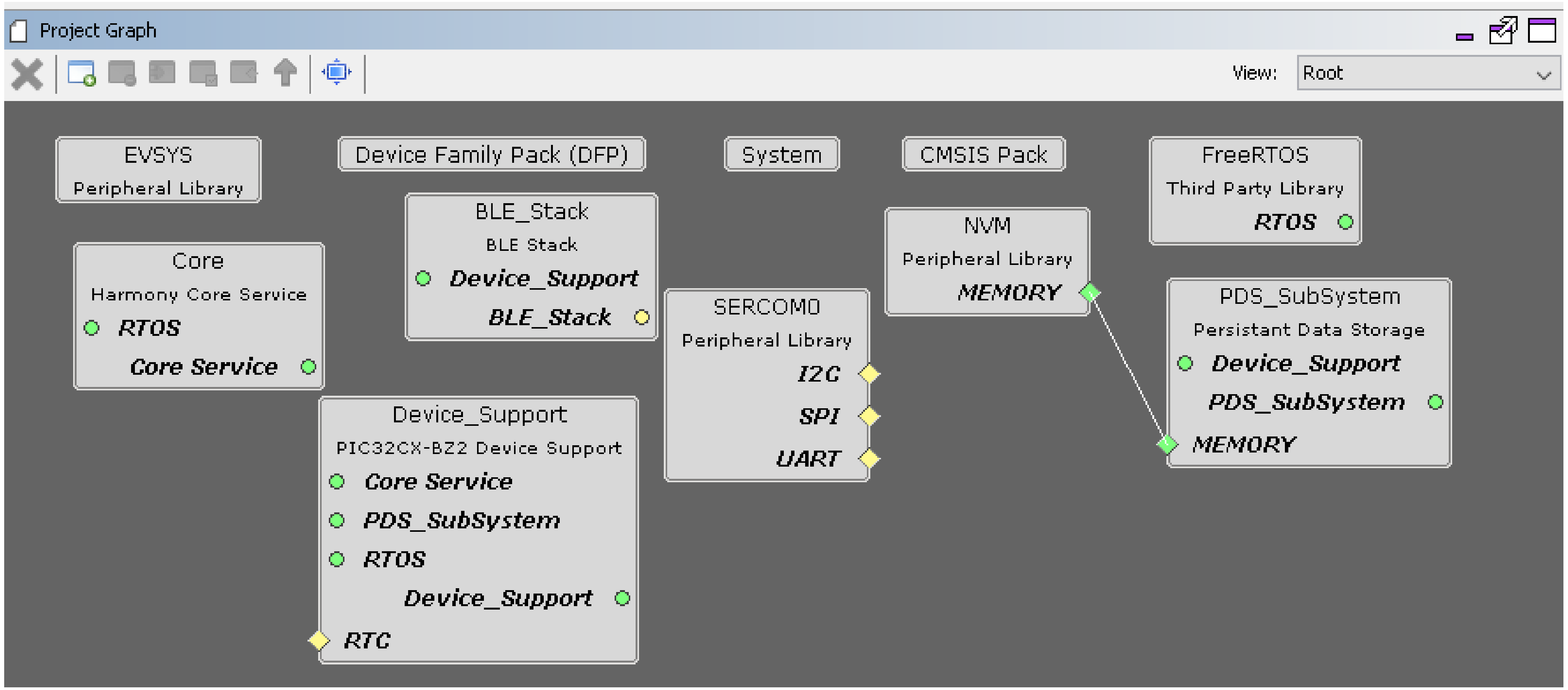
The database is requesting automatic connection following component attachments:

pdsSystem:NVM_Dependency -> nvm:NVM_MEMORY

Are you sure you want to automatically make these connections?

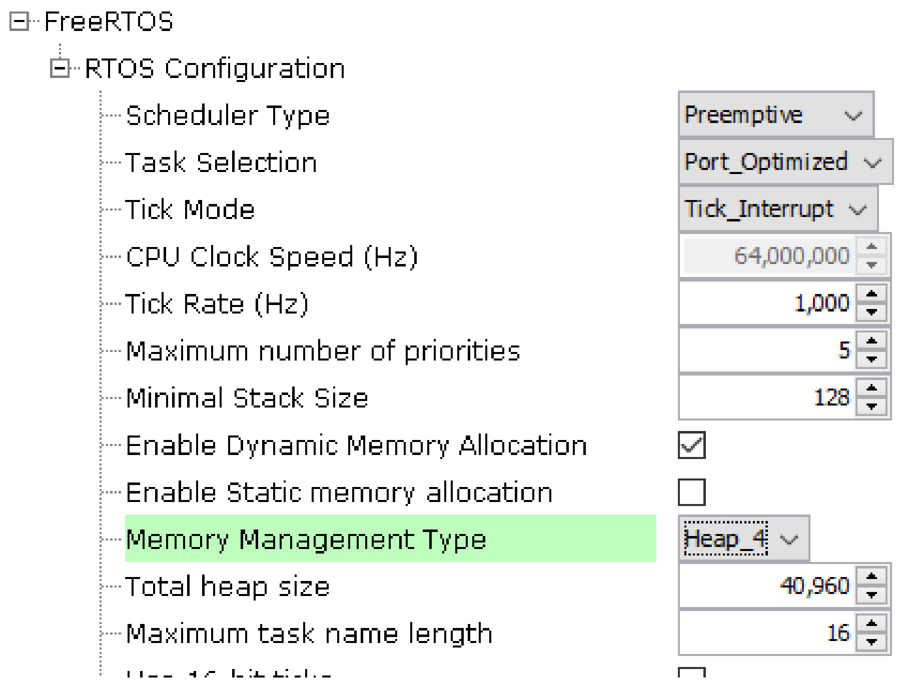
Yes No

Project graph after adding components



Harmony 3 Configurator

- **FreeRTOS Configurations**
  - Dynamic Memory Allocation

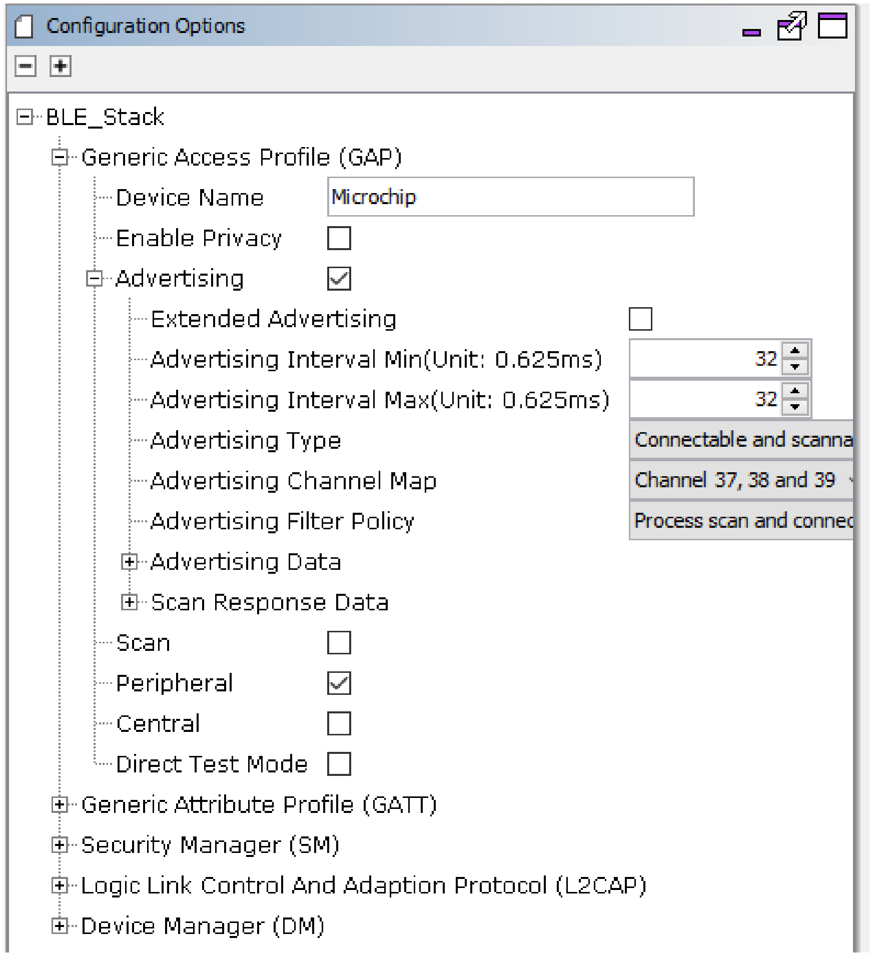


FreeRTOS
RTOS Configuration

Scheduler Type	Preemptive
Task Selection	Port_Optimized
Tick Mode	Tick_Interrupt
CPU Clock Speed (Hz)	64,000,000
Tick Rate (Hz)	1,000
Maximum number of priorities	5
Minimal Stack Size	128
Enable Dynamic Memory Allocation	☑
Enable Static memory allocation	☐
Memory Management Type	Heap_4
Total heap size	40,960
Maximum task name length	16

Harmony 3 Configurator

- **Drag BLE_Stack from “Available Component” to “Project Graph”**
  - Gap Configuration

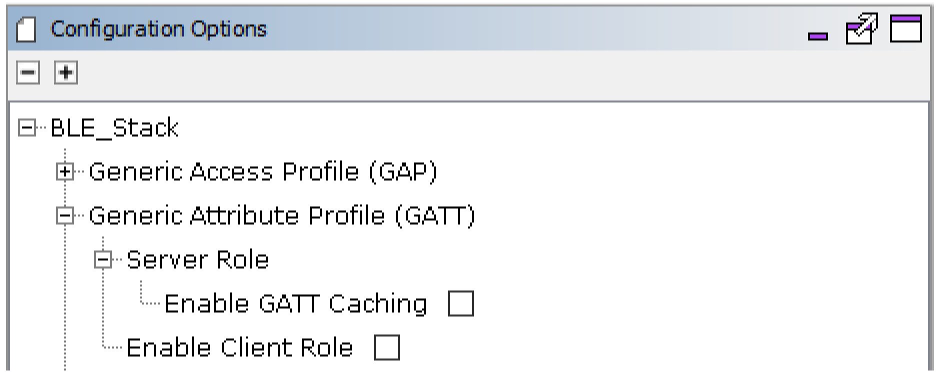


Configuration Options

- BLE_Stack
  - Generic Access Profile (GAP)
    - Device Name: Microchip
    - Enable Privacy: ☐
    - Advertising: ☑
      - Extended Advertising: ☐
      - Advertising Interval Min(Unit: 0.625ms): 32
      - Advertising Interval Max(Unit: 0.625ms): 32
      - Advertising Type: Connectable and scanna
      - Advertising Channel Map: Channel 37, 38 and 39
      - Advertising Filter Policy: Process scan and connec
      - Advertising Data
      - Scan Response Data
    - Scan: ☐
    - Peripheral: ☑
    - Central: ☐
    - Direct Test Mode: ☐
  - Generic Attribute Profile (GATT)
  - Security Manager (SM)
  - Logic Link Control And Adaption Protocol (L2CAP)
  - Device Manager (DM)

Harmony 3 Configurator

- **Drag BLE_Stack from “Available Component” to “Project Graph”**
  - GATT Configuration

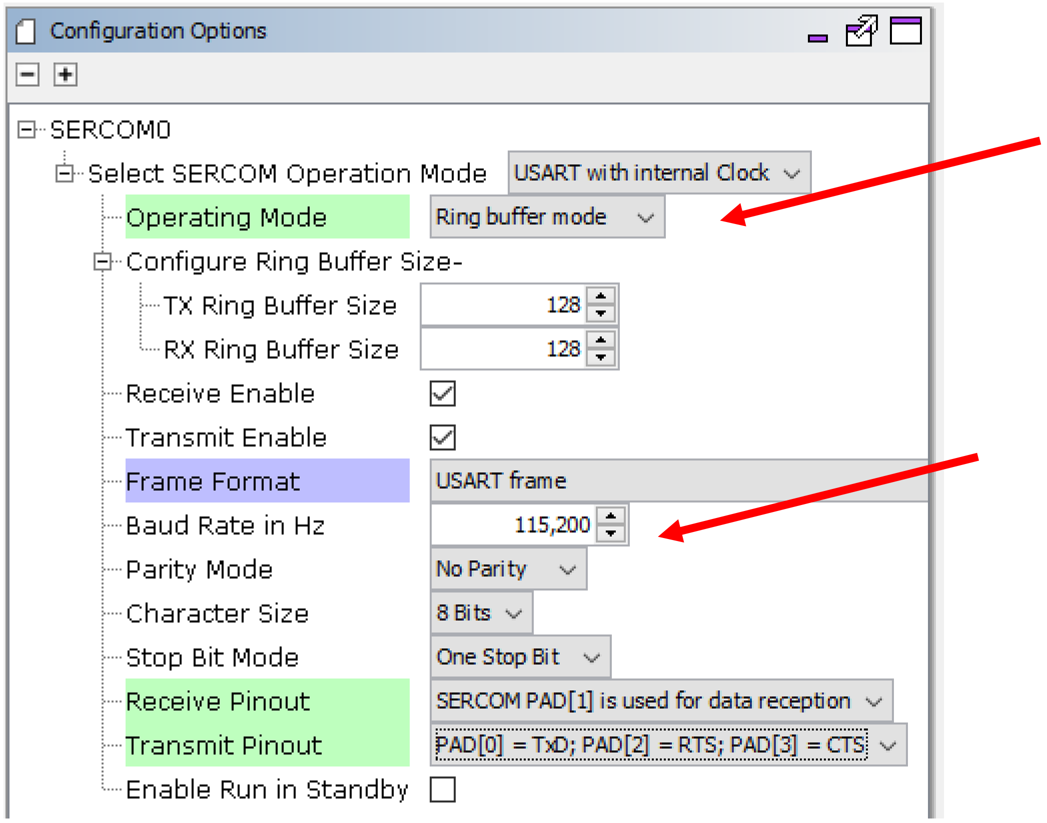


Configuration Options

- BLE_Stack
  - Generic Access Profile (GAP)
  - Generic Attribute Profile (GATT)
    - Server Role
      - Enable GATT Caching ☐
    - Enable Client Role ☐

Harmony 3 Configurator

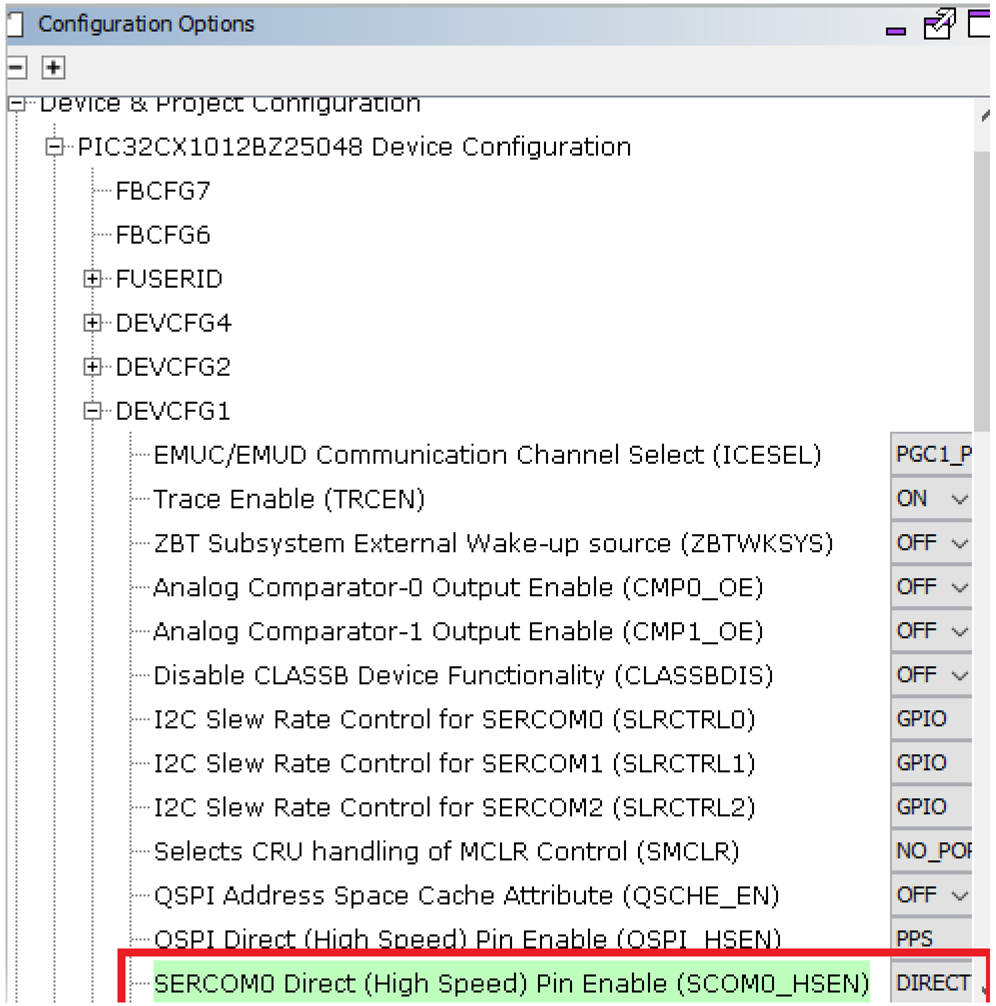
- **Peripheral Configuration**
  - Drag SERCOM0 from Peripherals/SERCOM
  - UART Configuration



SERCOM0	
Select SERCOM Operation Mode	USART with internal Clock
Operating Mode	Ring buffer mode
Configure Ring Buffer Size-	
TX Ring Buffer Size	128
RX Ring Buffer Size	128
Receive Enable	☑
Transmit Enable	☑
Frame Format	USART frame
Baud Rate in Hz	115,200
Parity Mode	No Parity
Character Size	8 Bits
Stop Bit Mode	One Stop Bit
Receive Pinout	SERCOM PAD[1] is used for data reception
Transmit Pinout	PAD[0] = TxD; PAD[2] = RTS; PAD[3] = CTS
Enable Run in Standby	☐

Harmony 3 Configurator

- **System Configurations**
  - Select SERCOM0(DIRECT) mode



Configuration Options

Device & Project Configuration

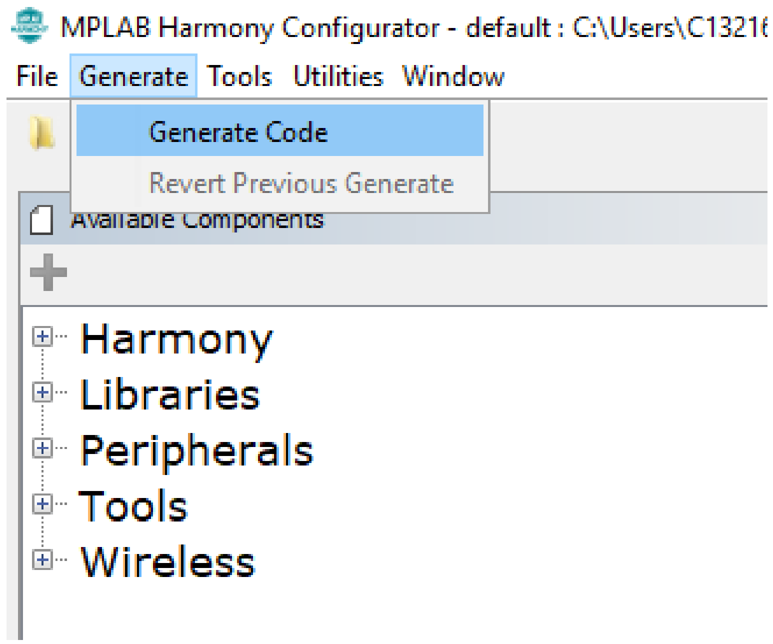
PIC32CX1012BZ25048 Device Configuration

- FBCFG7
- FBCFG6
- FUSERID
- DEVCFG4
- DEVCFG2
- DEVCFG1

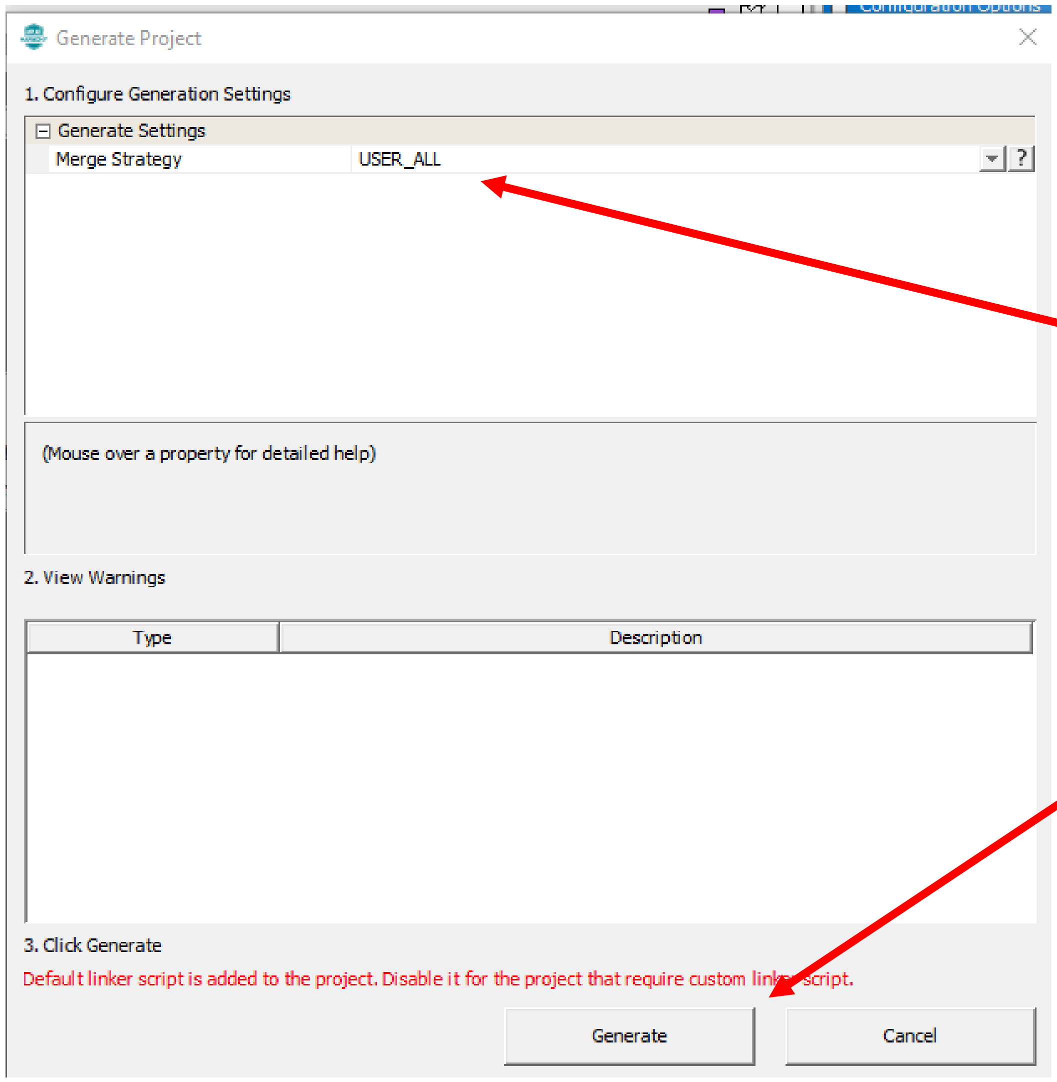
EMUC/EMUD Communication Channel Select (ICESEL)	PGC1_P
Trace Enable (TRCEN)	ON
ZBT Subsystem External Wake-up source (ZBTWKSYS)	OFF
Analog Comparator-0 Output Enable (CMP0_OE)	OFF
Analog Comparator-1 Output Enable (CMP1_OE)	OFF
Disable CLASSB Device Functionality (CLASSBDIS)	OFF
I2C Slew Rate Control for SERCOM0 (SLRCTRL0)	GPIO
I2C Slew Rate Control for SERCOM1 (SLRCTRL1)	GPIO
I2C Slew Rate Control for SERCOM2 (SLRCTRL2)	GPIO
Selects CRU handling of MCLR Control (SMCLR)	NO_POI
QSPI Address Space Cache Attribute (QSCHE_EN)	OFF
QSPI Direct (High Speed) Pin Enable (QSPI_HSEN)	PPS
SERCOM0 Direct (High Speed) Pin Enable (SCOM0_HSEN)	DIRECT

Harmony 3 Configurator

- **Generate Code**



Harmony 3 Configurator



1. Configure Generation Settings

Generate Settings	
Merge Strategy	USER_ALL

(Mouse over a property for detailed help)

2. View Warnings

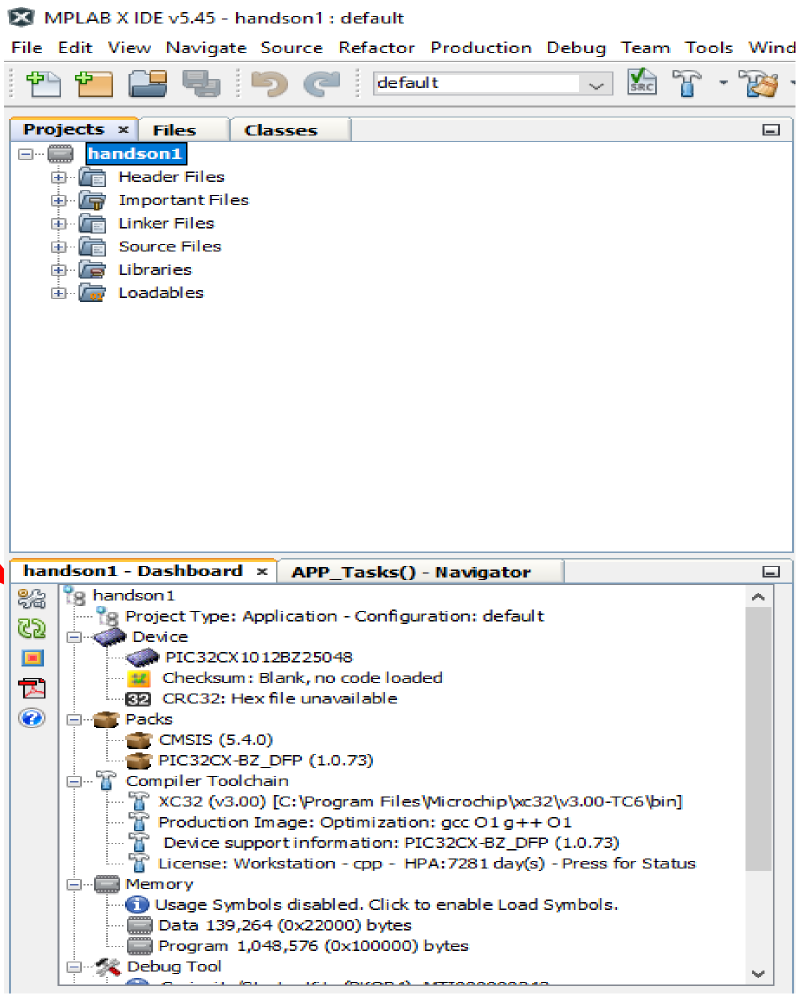
Type	Description

3. Click Generate

Default linker script is added to the project. Disable it for the project that require custom linker script.

First Program

- Minimize the MPLAB Harmony Configurator Window
- Switch to MPLABX Window
- Click Project Properties

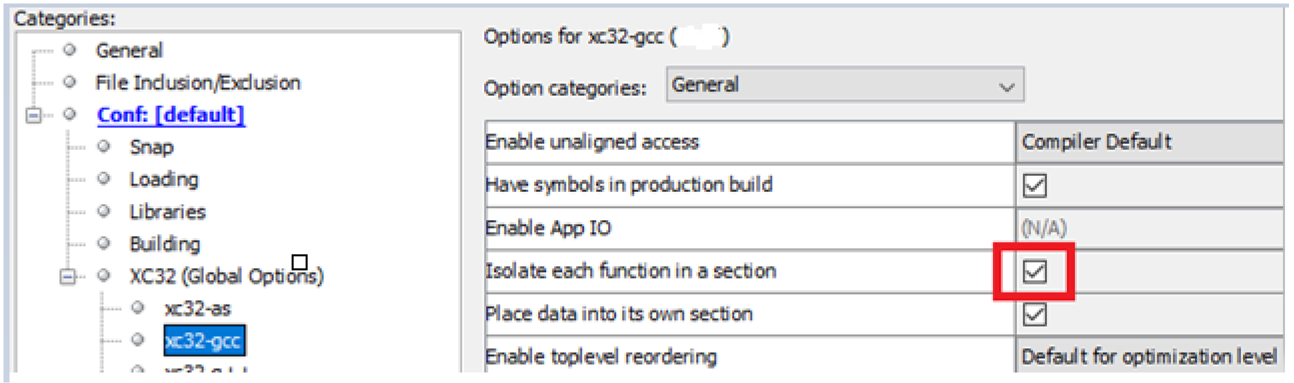

MPLAB X IDE v5.45 - handson1 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Wind
default
Projects
Files
Classes
handson1 - Dashboard
APP_Tasks() - Navigator

```
handson1
    Header Files
    Important Files
    Linker Files
    Source Files
    Libraries
    Loadables
```

```
handson1
  Project Type: Application - Configuration: default
  Device
    PIC32CX1012BZ25048
    Checksum: Blank, no code loaded
    CRC32: Hex file unavailable
  Packs
    CMSIS (5.4.0)
    PIC32CX-BZ_DFP (1.0.73)
  Compiler Toolchain
    XC32 (v3.00) [C:\Program Files\Microchip\xc32\v3.00-TC6\bin]
    Production Image: Optimization: gcc O1 g++ O1
    Device support information: PIC32CX-BZ_DFP (1.0.73)
    License: Workstation - cpp -  HPA:7281 day(s) - Press for Status
  Memory
    Usage Symbols disabled. Click to enable Load Symbols.
    Data 139,264 (0x22000) bytes
    Program  1,048,576 (0x100000) bytes
  Debug Tool
```

Compiler settings

- **Software**
  - Compiler
    - XC32 v3.01
      - Disable Optimization to Debug
      - Enable Optimization for minimum footprint

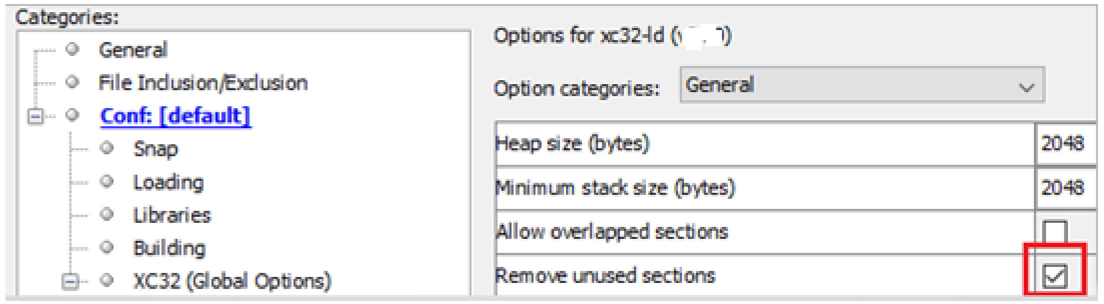


Categories:
- General
- File Inclusion/Exclusion
- Conf: [default]
  - Snap
  - Loading
  - Libraries
  - Building
  - XC32 (Global Options)
    - xc32-as
    - xc32-gcc

Options for xc32-gcc ( )

Option categories: General

Enable unaligned access	Compiler Default
Have symbols in production build	✓
Enable App IO	(N/A)
Isolate each function in a section	✓
Place data into its own section	✓
Enable toplevel reordering	Default for optimization level



Categories:
- General
- File Inclusion/Exclusion
- Conf: [default]
  - Snap
  - Loading
  - Libraries
  - Building
  - XC32 (Global Options)

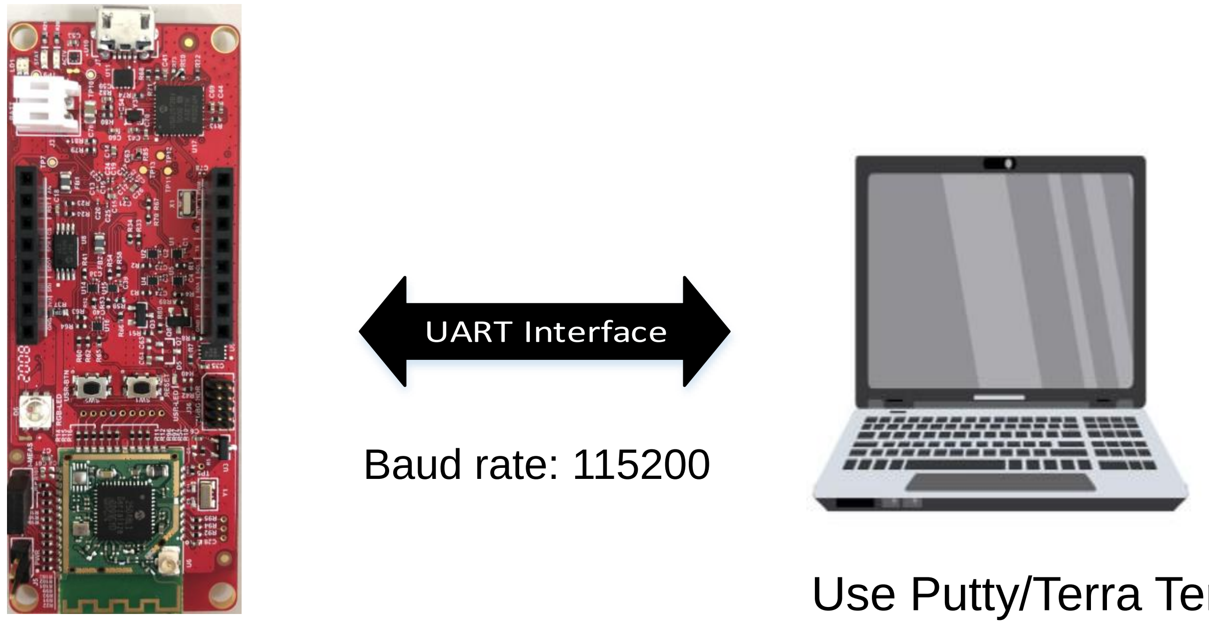
Options for xc32-ld ( )

Option categories: General

Heap size (bytes)	2048
Minimum stack size (bytes)	2048
Allow overlapped sections	☐
Remove unused sections	✓

First Program

- **Hello World**

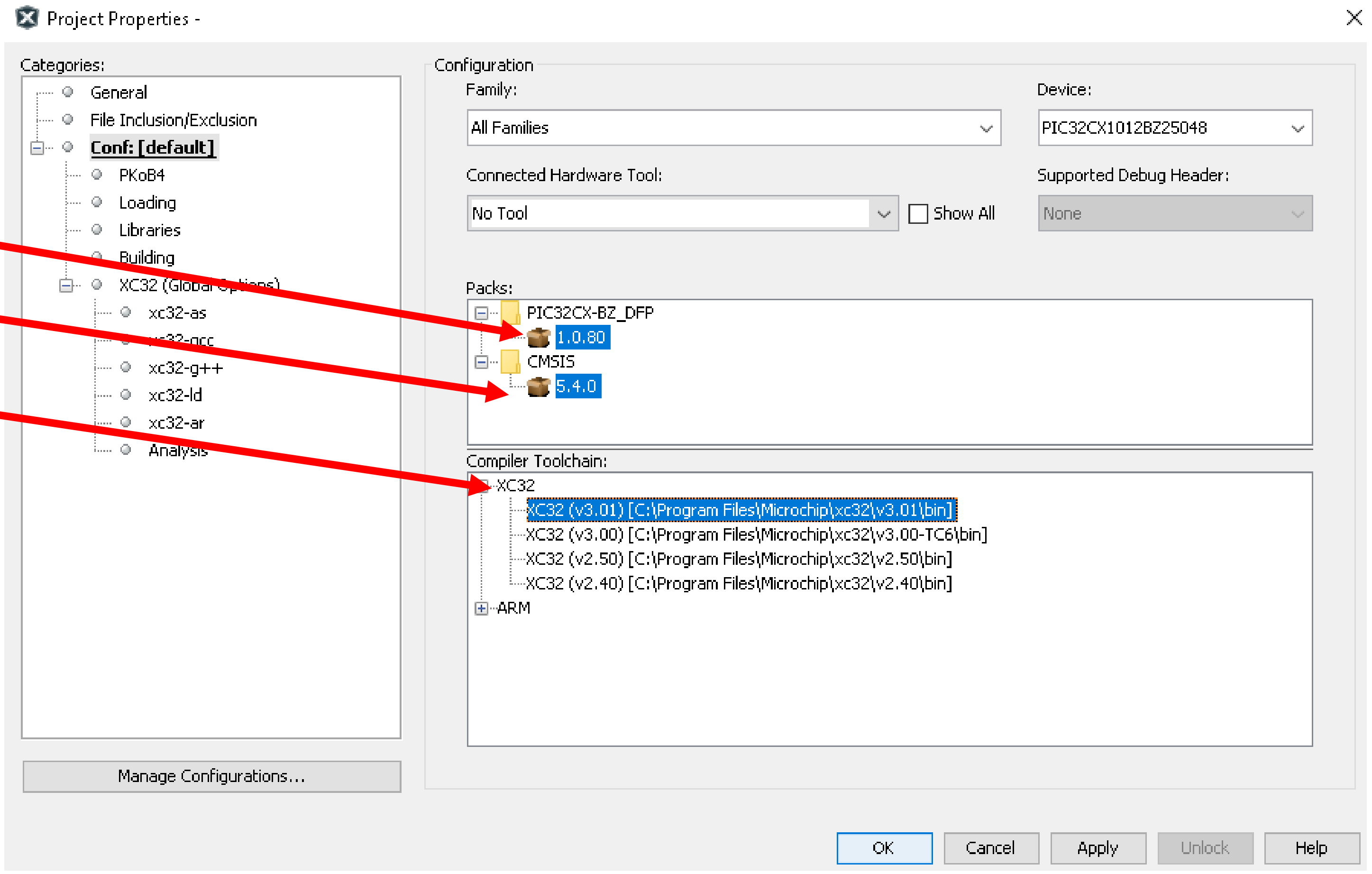


Baud rate: 115200

Use Putty/Terra Term, other serial port emulator

First Program

- **Verify that the XC32 compiler v3.01 and DFP 1.0.80 and PKOB4 is selected in MPLAB X IDE**


Project Properties -

Categories:
- General
- File Inclusion/Exclusion
- Conf: [default]
  - PKoB4
  - Loading
  - Libraries
  - Building
  - XC32 (Global Options)
    - xc32-as
    - xc32-gcc
    - xc32-g++
    - xc32-ld
    - xc32-ar
    - Analysis

Configuration
Family: All Families
Device: PIC32CX1012BZ25048
Connected Hardware Tool: No Tool
Show All
Supported Debug Header: None

Packs:
- PIC32CX-BZ_DFP
  - 1.0.80
- CMSIS
  - 5.4.0

Compiler Toolchain:
- XC32
  - XC32 (v3.01) [C:\Program Files\Microchip\xc32\v3.01\bin]
  - XC32 (v3.00) [C:\Program Files\Microchip\xc32\v3.00-TC6\bin]
  - XC32 (v2.50) [C:\Program Files\Microchip\xc32\v2.50\bin]
  - XC32 (v2.40) [C:\Program Files\Microchip\xc32\v2.40\bin]
- ARM

First Program

- **Modify the source code**
  - Open app.c
  - Add: Include header file ‘definitions.h’ -> just a bunch of more includes…

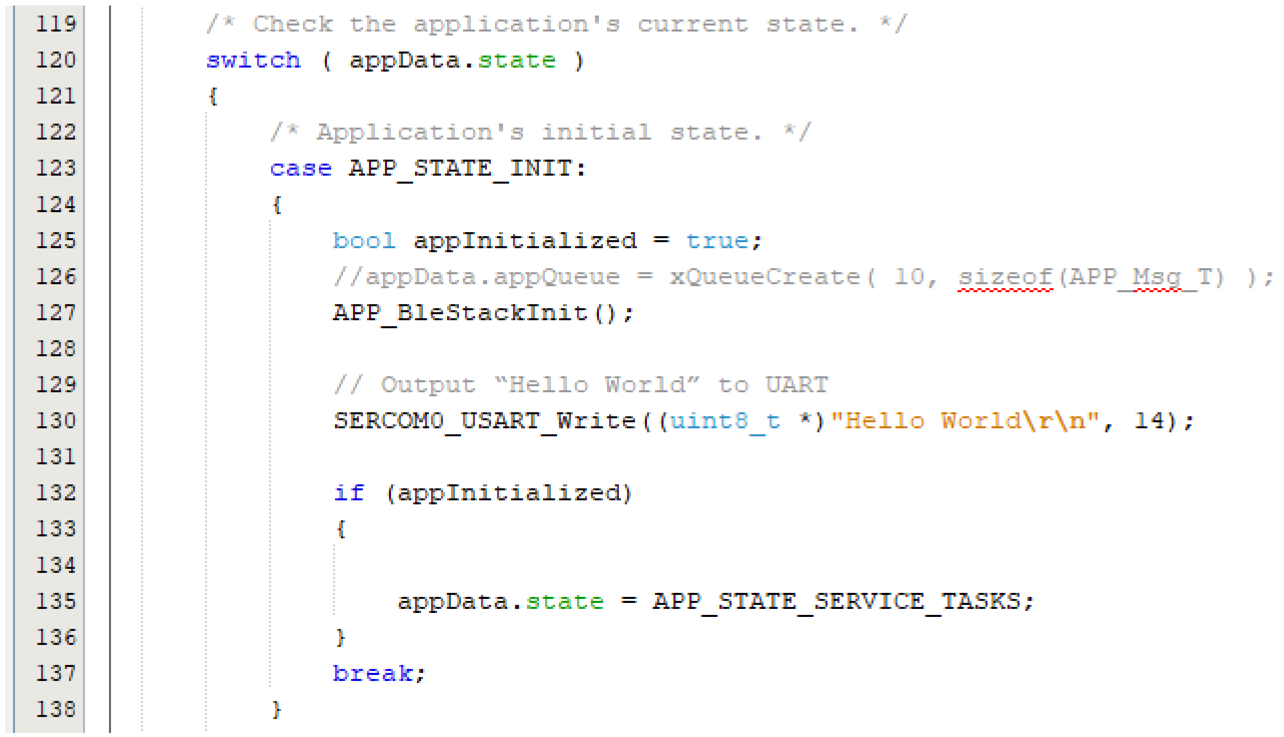
```
#include "definitions.h"
```

  - Search for (CTRL-F) “case APP_STATE_INIT:” (about line 129)
  - Then add following code to output “Hello, World”

```
// Output “Hello World” to UART
SERCOM0_USART_Write((uint8_t *)"Hello World\r\n", 14);
```

First Program

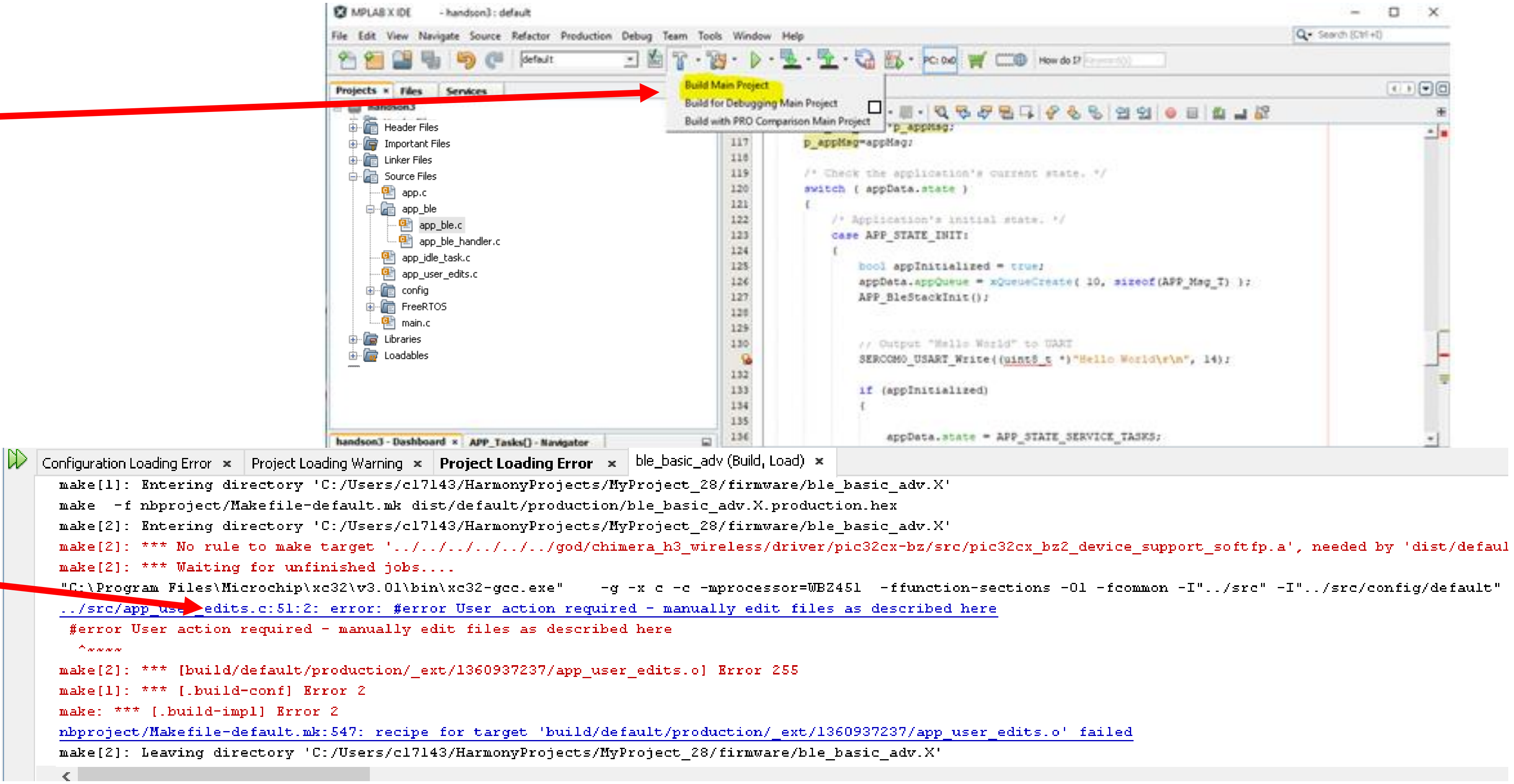
- **app.c now looks like this:**



```
/* Check the application's current state. */
switch ( appData.state )
{
    /* Application's initial state. */
    case APP_STATE_INIT:
    {
        bool appInitialized = true;
        //appData.appQueue = xQueueCreate( 10, sizeof(APP_Msg_T) );
        APP_BleStackInit();

        // Output "Hello World" to UART
        SERCOM0_USART_Write((uint8_t *)"Hello World\r\n", 14);

        if (appInitialized)
        {

            appData.state = APP_STATE_SERVICE_TASKS;
        }
        break;
    }
```

First Program

- **compile the code**



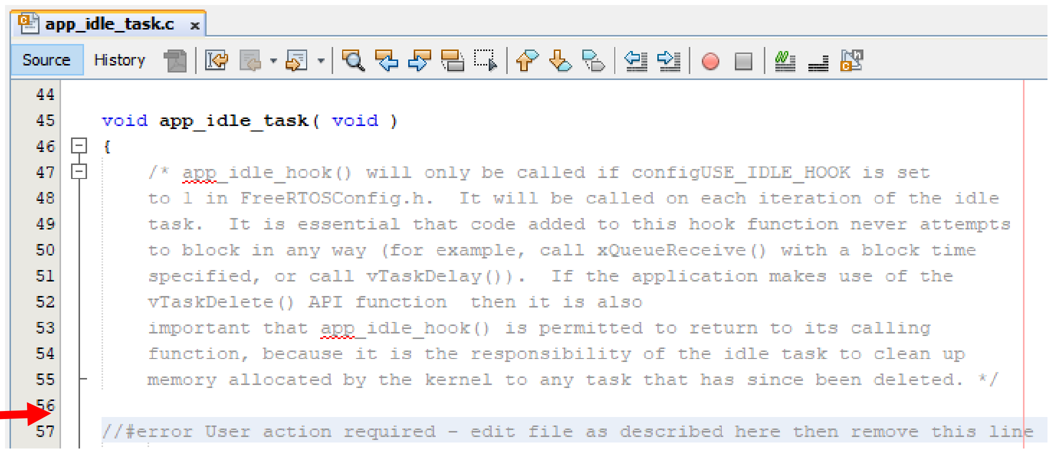
```
p_appMsg=appMsg;

/* Check the application's current state. */
switch ( appData.state )
{
    /* Application's initial state. */
    case APP_STATE_INIT:
    {
        bool appInitialized = true;
        appData.appQueue = xQueueCreate( 10, sizeof(APP_Msg_T) );
        APP_BleStackInit();


        // Output "Hello World" to UART
        SERCOM0_USART_Write((uint8_t *)"Hello World\r\n", 14);

        if (appInitialized)
        {

          appData.state = APP_STATE_SERVICE_TASKS;
```

```
make[1]: Entering directory 'C:/Users/c17143/HarmonyProjects/MyProject_28/firmware/ble_basic_adv.X'
make  -f nbproject/Makefile-default.mk dist/default/production/ble_basic_adv.X.production.hex
make[2]: Entering directory 'C:/Users/c17143/HarmonyProjects/MyProject_28/firmware/ble_basic_adv.X'
make[2]: *** No rule to make target '../../../../../../god/chimera_h3_wireless/driver/pic32cx-bz/src/pic32cx_bz2_device_support_softfp.a', needed by 'dist/defaul
make[2]: *** Waiting for unfinished jobs....
"C:\Program Files\Microchip\xc32\v3.01\bin\xc32-gcc.exe"    -g -x c -c -mprocessor=WBZ451  -ffunction-sections -O1 -fcommon -I"../src" -I"../src/config/default"
../src/app_user_edits.c:51:2: error: #error User action required - manually edit files as described here
 #error User action required - manually edit files as described here
  ^~~~~
make[2]: *** [build/default/production/_ext/1360937237/app_user_edits.o] Error 255
make[1]: *** [.build-conf] Error 2
make: *** [.build-impl] Error 2
nbproject/Makefile-default.mk:547: recipe for target 'build/default/production/_ext/1360937237/app_user_edits.o' failed
make[2]: Leaving directory 'C:/Users/c17143/HarmonyProjects/MyProject_28/firmware/ble_basic_adv.X'
```

Compile error

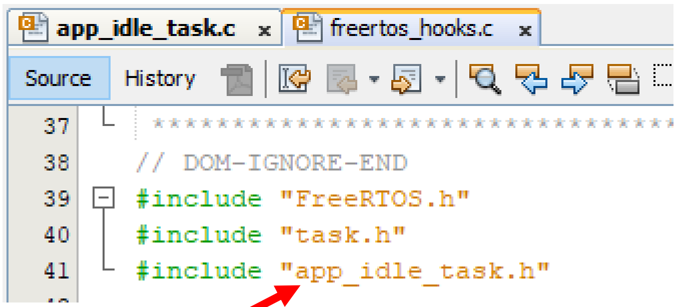
First Program

- **Fixing the compile error**



```
void app_idle_task( void )
{
    /* app_idle_hook() will only be called if configUSE_IDLE_HOOK is set
    to 1 in FreeRTOSConfig.h.  It will be called on each iteration of the idle
    task.  It is essential that code added to this hook function never attempts
    to block in any way (for example, call xQueueReceive() with a block time
    specified, or call vTaskDelay()).  If the application makes use of the
    vTaskDelete() API function  then it is also
    important that app_idle_hook() is permitted to return to its calling
    function, because it is the responsibility of the idle task to clean up
    memory allocated by the kernel to any task that has since been deleted. */

//#error User action required - edit file as described here then remove this line
```

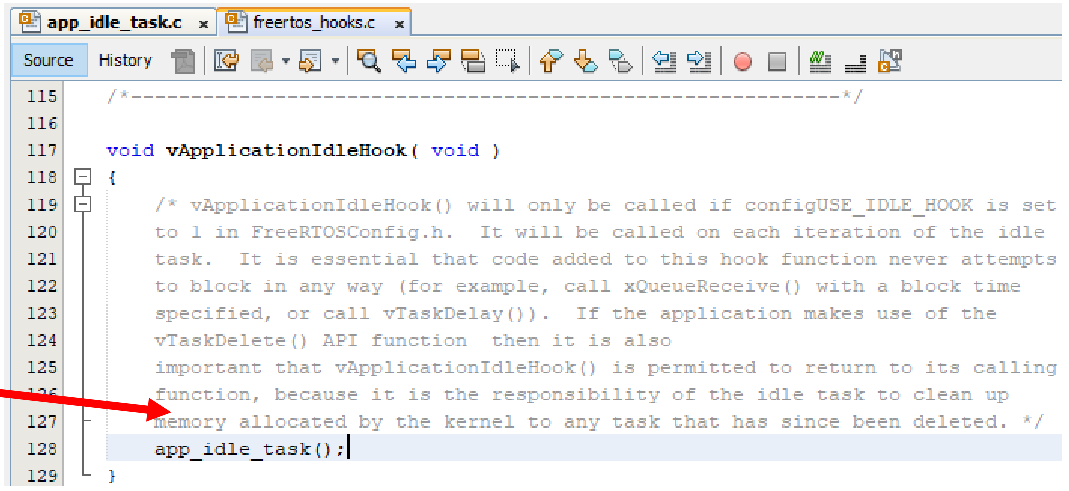
Comment out the #error



```
    *********************************
    // DOM-IGNORE-END
#include "FreeRTOS.h"
#include "task.h"
#include "app_idle_task.h"
```

Include – “app_idle_task.h”

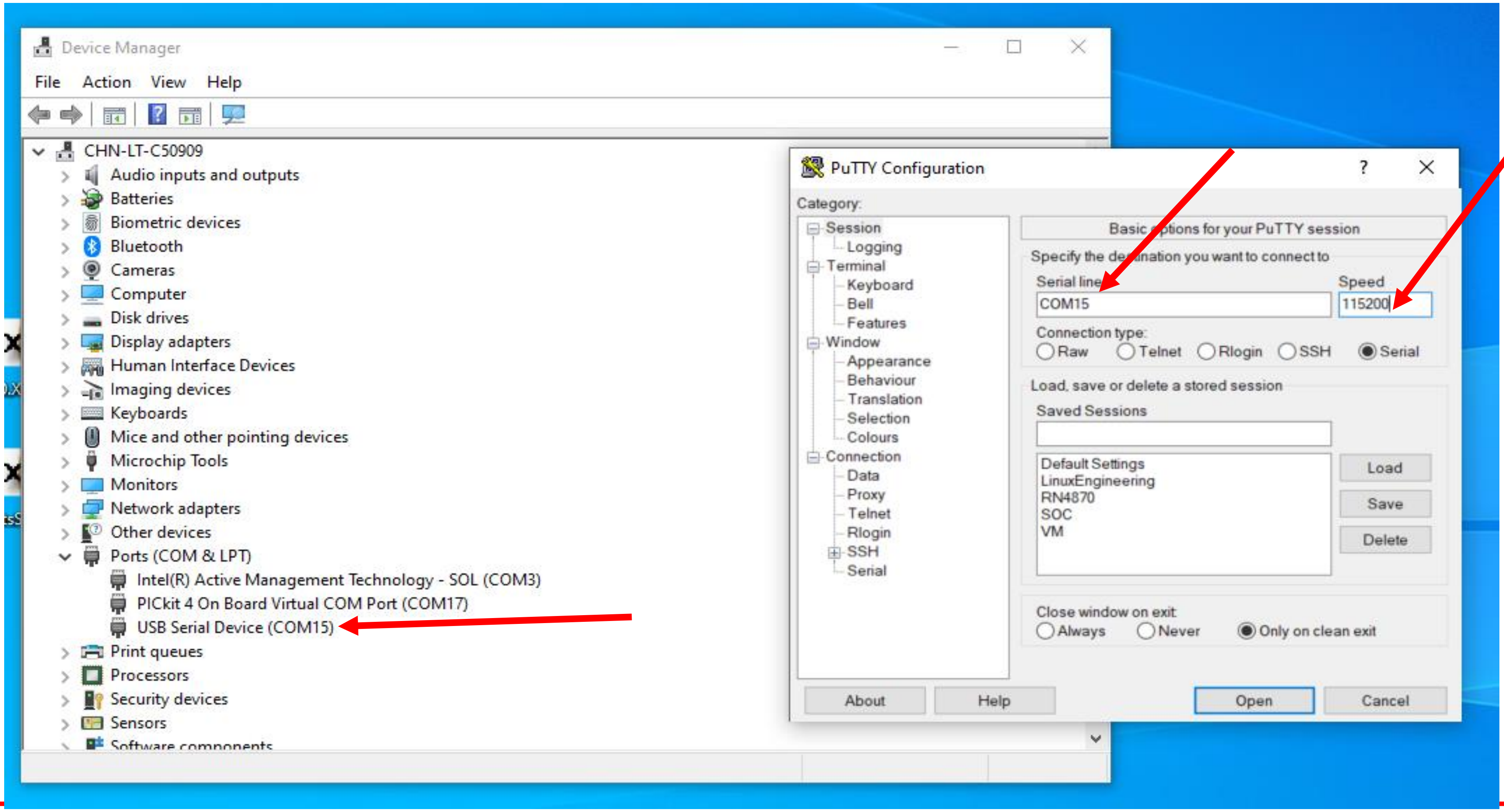
Add – app_idle_task()



```
/*-----------------------------------------------------------*/

void vApplicationIdleHook( void )
{
    /* vApplicationIdleHook() will only be called if configUSE_IDLE_HOOK is set
    to 1 in FreeRTOSConfig.h.  It will be called on each iteration of the idle
    task.  It is essential that code added to this hook function never attempts
    to block in any way (for example, call xQueueReceive() with a block time
    specified, or call vTaskDelay()).  If the application makes use of the
    vTaskDelete() API function  then it is also
    important that vApplicationIdleHook() is permitted to return to its calling
    function, because it is the responsibility of the idle task to clean up
    memory allocated by the kernel to any task that has since been deleted. */
    app_idle_task();
}
```

First Program

- **Open Terminal Emulator → Putty example. YOU NEED TO KNOW COM PORT, and set serial speed to 115200**



Device Manager

- CHN-LT-C50909
  - Audio inputs and outputs
  - Batteries
  - Biometric devices
  - Bluetooth
  - Cameras
  - Computer
  - Disk drives
  - Display adapters
  - Human Interface Devices
  - Imaging devices
  - Keyboards
  - Mice and other pointing devices
  - Microchip Tools
  - Monitors
  - Network adapters
  - Other devices
  - Ports (COM & LPT)
    - Intel(R) Active Management Technology - SOL (COM3)
    - PICkit 4 On Board Virtual COM Port (COM17)
    - USB Serial Device (COM15)
  - Print queues
  - Processors
  - Security devices
  - Sensors
  - Software components

PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM15

Speed: 115200

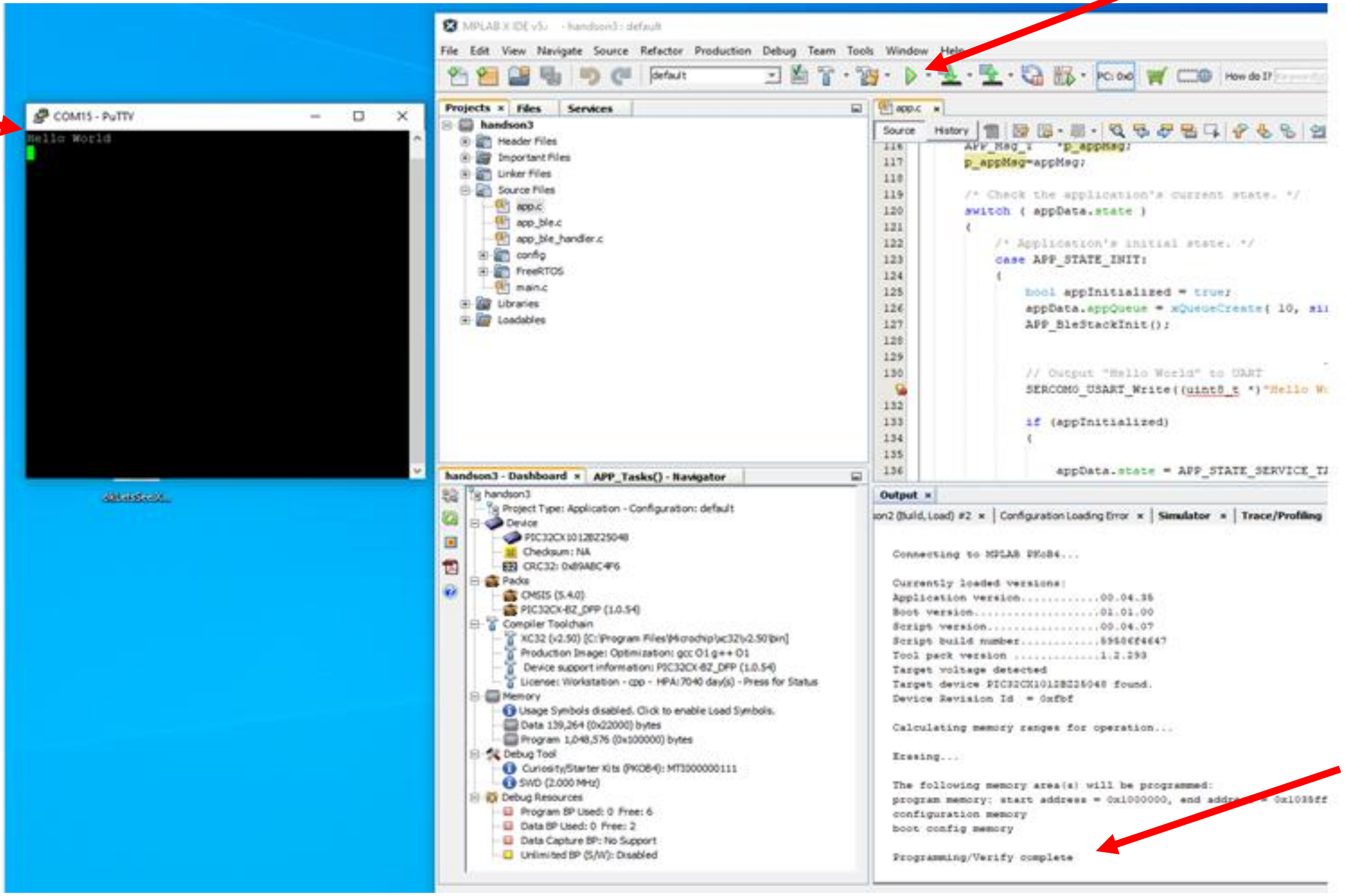
Connection type: Raw / Telnet / Rlogin / SSH / [✓ Serial]

Saved Sessions: Default Settings, LinuxEngineering, RN4870, SOC, VM

Close window on exit: Always / Never / [✓ Only on clean exit]

First Program

- **Run the Project and Observe Terminal emulator output**


COM15 - PuTTY
MPLAB X IDE v5.- handson3 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
Projects
Files
Services
app.c
Source
History
Output

```
Hello World
```

```
handson3
  Header Files
  Important Files
  Linker Files
  Source Files
    app.c
    app_ble.c
    app_ble_handler.c
    config
    FreeRTOS
    main.c
  Libraries
  Loadables
```

```
    APP_Msg_t      *p_appMsg;
    p_appMsg=appMsg;

    /* Check the application's current state. */
    switch ( appData.state )
    {
        /* Application's initial state. */
        case APP_STATE_INIT:
        {
            bool appInitialized = true;
            appData.appQueue = xQueueCreate( 10, si
            APP_BleStackInit();


            // Output "Hello World" to UART
            SERCOM0_USART_Write((uint8_t *)"Hello W

            if (appInitialized)
            {

                appData.state = APP_STATE_SERVICE_T
```

handson3 - Dashboard | APP_Tasks() - Navigator

```
handson3
  Project Type: Application - Configuration: default
  Device
    PIC32CX1012BZ25048
    Checksum: NA
    CRC32: 0x89ABC4F6
  Packs
    CMSIS (5.4.0)
    PIC32CX-BZ_DFP (1.0.54)
  Compiler Toolchain
    XC32 (v2.50) [C:\Program Files\Microchip\xc32\v2.50\bin]
    Production Image: Optimization: gcc O1 g++ O1
    Device support information: PIC32CX-BZ_DFP (1.0.54)
    License: Workstation - cpp - HPA:7040 day(s) - Press for Status
  Memory
    Usage Symbols disabled. Click to enable Load Symbols.
    Data 139,264 (0x22000) bytes
    Program 1,048,576 (0x100000) bytes
  Debug Tool
    Curiosity/Starter Kits (PKOB4): MT3000000111
    SWD (2.000 MHz)
  Debug Resources
    Program BP Used: 0  Free: 6
    Data BP Used: 0  Free: 2
    Data Capture BP: No Support
    Unlimited BP (S/W): Disabled
```

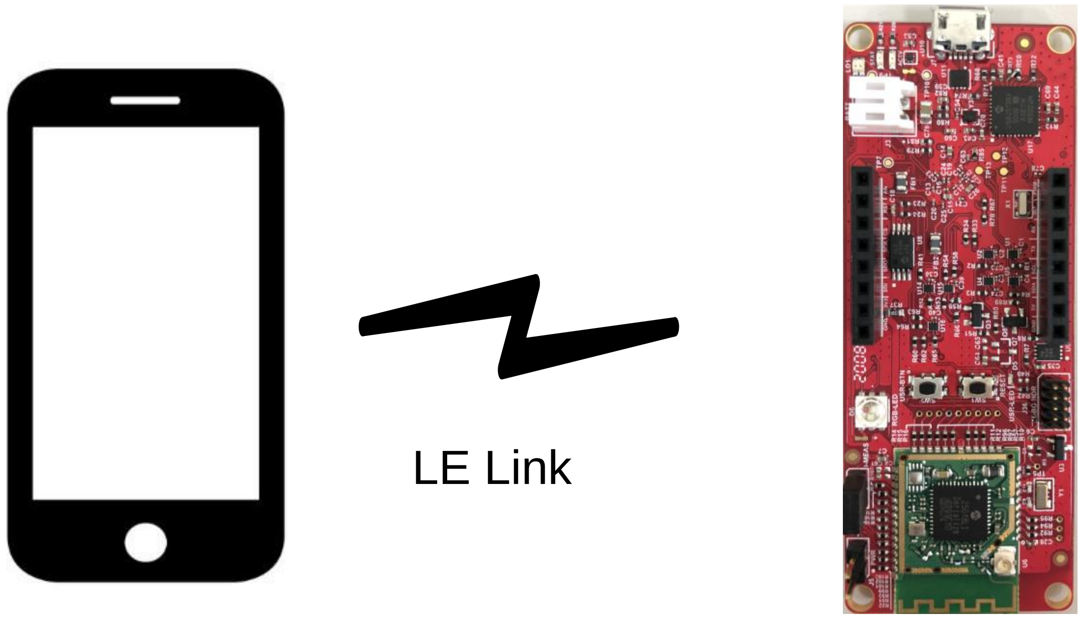
on2 (Build, Load) #2 | Configuration Loading Error | Simulator | Trace/Profiling

```
Connecting to MPLAB PKoB4...

Currently loaded versions:
Application version............00.04.36
Boot version...................01.01.00
Script version.................00.04.07
Script build number............5950ff4647
Tool pack version .............1.2.293
Target voltage detected
Target device PIC32CX1012BZ25048 found.
Device Revision Id  = 0xfbf

Calculating memory ranges for operation...

Erasing...

The following memory area(s) will be programmed:
program memory: start address = 0x1000000, end address = 0x1035ff
configuration memory
boot config memory

Programming/Verify complete
```

If you push the ‘Reset’ button on the Curiosity board it will print again

Second Program

- **Simple Advertiser**

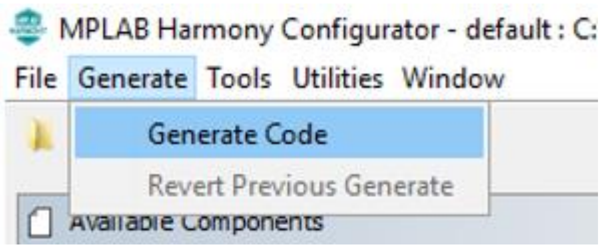


Android: LightBlue
iOS: LightBlue

Second Program

- After SERCOM0_USART_Write((uint8_t *)"Hello World\r\n", 14); open Harmony 3 configurator and configure advertisement data

- Generate Code

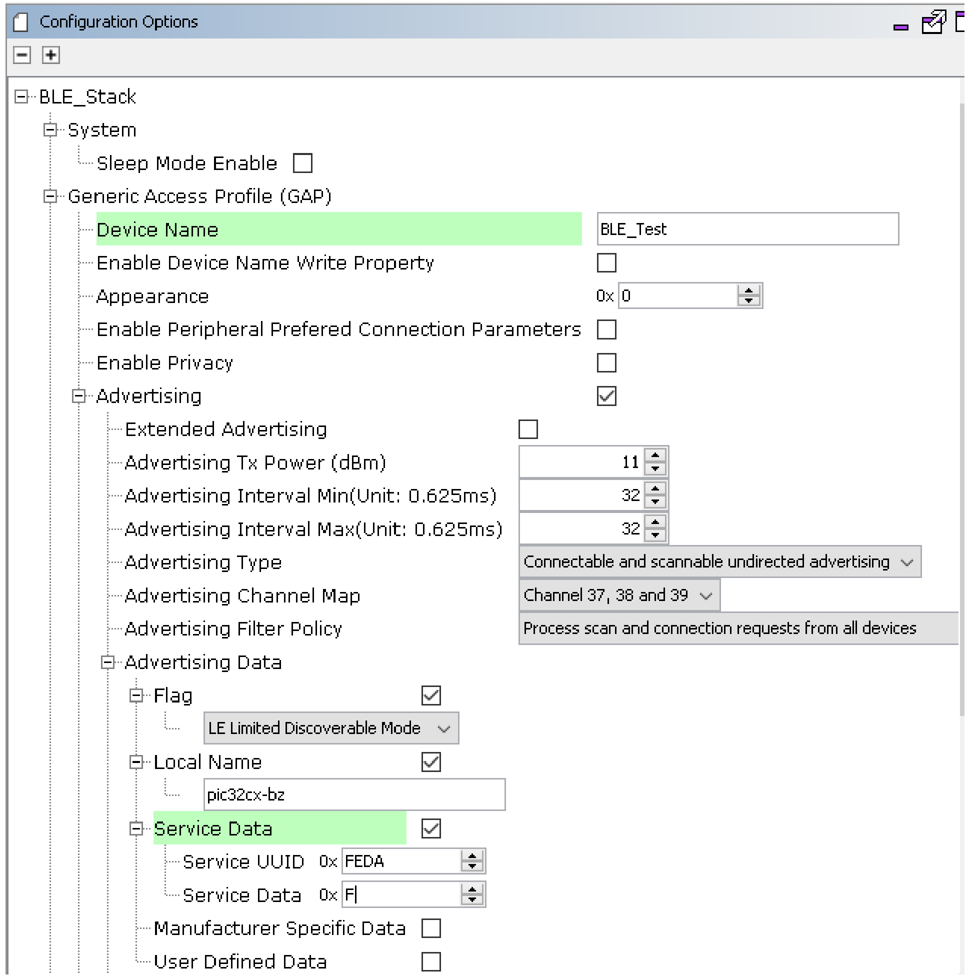

MPLAB Harmony Configurator - default : C:

File | Generate | Tools | Utilities | Window

Generate Code

Revert Previous Generate

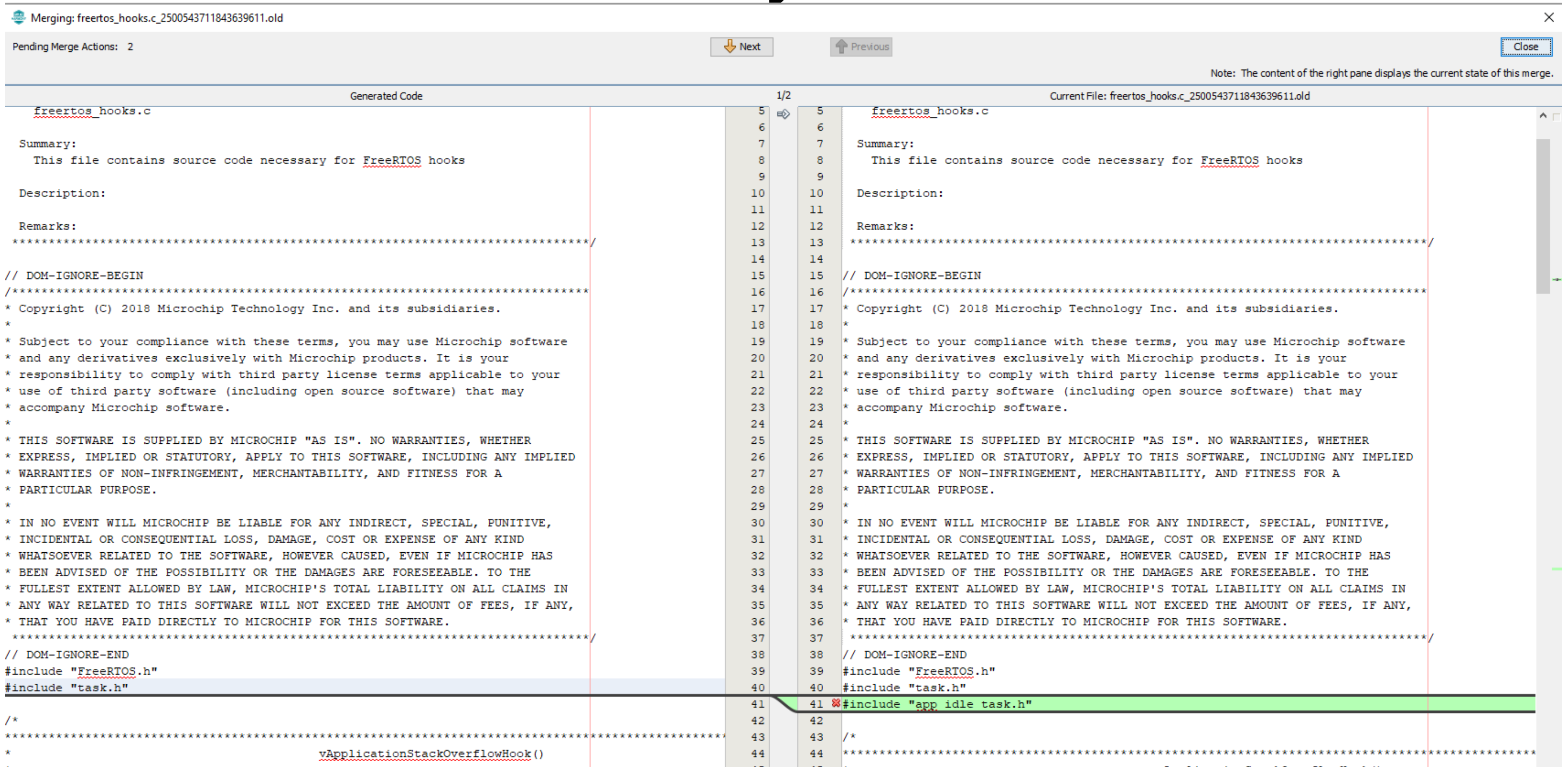
Available Components


Configuration Options

- BLE_Stack
  - System
    - Sleep Mode Enable ☐
  - Generic Access Profile (GAP)
    - Device Name: BLE_Test
    - Enable Device Name Write Property ☐
    - Appearance: 0x 0
    - Enable Peripheral Prefered Connection Parameters ☐
    - Enable Privacy ☐
    - Advertising ✓
      - Extended Advertising ☐
      - Advertising Tx Power (dBm): 11
      - Advertising Interval Min(Unit: 0.625ms): 32
      - Advertising Interval Max(Unit: 0.625ms): 32
      - Advertising Type: Connectable and scannable undirected advertising
      - Advertising Channel Map: Channel 37, 38 and 39
      - Advertising Filter Policy: Process scan and connection requests from all devices
      - Advertising Data
        - Flag ✓
          - LE Limited Discoverable Mode
        - Local Name ✓
          - pic32cx-bz
        - Service Data ✓
          - Service UUID 0x FEDA
          - Service Data 0x F
        - Manufacturer Specific Data ☐
        - User Defined Data ☐

Second Program

- MHC Merging window shows modified code
- Close the window to retain your edits



Merging: freertos_hooks.c_25005437118436396 11.old

Pending Merge Actions: 2

Note: The content of the right pane displays the current state of this merge.

Generated Code — 1/2 — Current File: freertos_hooks.c_2500543711843639611.old

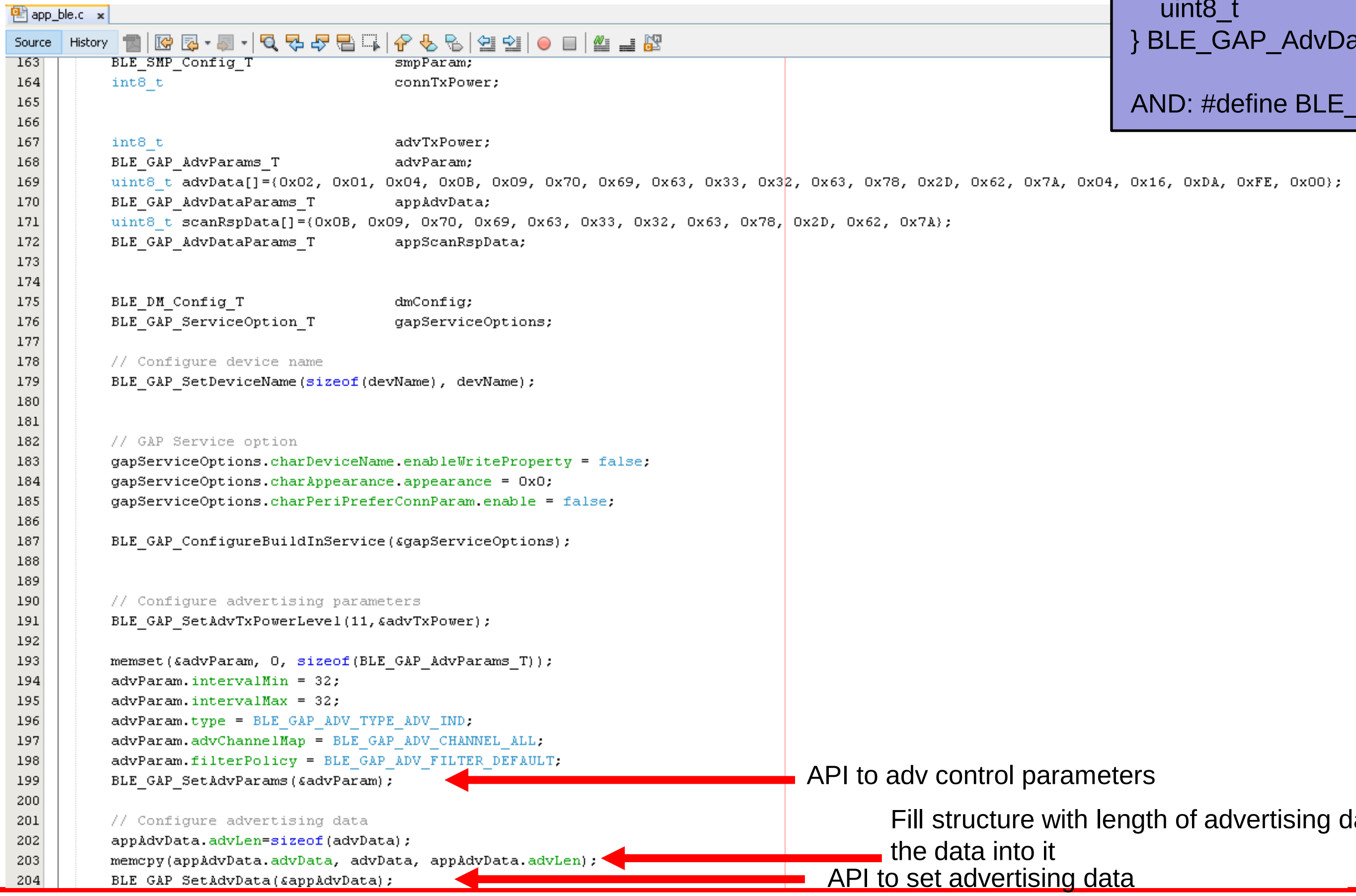
```
   freertos_hooks.c

  Summary:
    This file contains source code necessary for FreeRTOS hooks

  Description:

  Remarks:
 ******************************************************************************/

// DOM-IGNORE-BEGIN
/*******************************************************************************
* Copyright (C) 2018 Microchip Technology Inc. and its subsidiaries.
*
* Subject to your compliance with these terms, you may use Microchip software
* and any derivatives exclusively with Microchip products. It is your
* responsibility to comply with third party license terms applicable to your
* use of third party software (including open source software) that may
* accompany Microchip software.
*
* THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES, WHETHER
* EXPRESS, IMPLIED OR STATUTORY, APPLY TO THIS SOFTWARE, INCLUDING ANY IMPLIED
* WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A
* PARTICULAR PURPOSE.
*
* IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE,
* INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND
* WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS
* BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE
* FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN
* ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY,
* THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS SOFTWARE.
 ******************************************************************************/
// DOM-IGNORE-END
#include "FreeRTOS.h"
#include "task.h"

/*
*****************************************************************************************
*                                     vApplicationStackOverflowHook()
```

```
   freertos_hooks.c

  Summary:
    This file contains source code necessary for FreeRTOS hooks

  Description:

  Remarks:
 ******************************************************************************/

// DOM-IGNORE-BEGIN
/*******************************************************************************
* Copyright (C) 2018 Microchip Technology Inc. and its subsidiaries.
*
* Subject to your compliance with these terms, you may use Microchip software
* and any derivatives exclusively with Microchip products. It is your
* responsibility to comply with third party license terms applicable to your
* use of third party software (including open source software) that may
* accompany Microchip software.
*
* THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES, WHETHER
* EXPRESS, IMPLIED OR STATUTORY, APPLY TO THIS SOFTWARE, INCLUDING ANY IMPLIED
* WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A
* PARTICULAR PURPOSE.
*
* IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE,
* INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND
* WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS
* BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE
* FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN
* ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY,
* THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS SOFTWARE.
 ******************************************************************************/
// DOM-IGNORE-END
#include "FreeRTOS.h"
#include "task.h"
#include "app_idle_task.h"

/*
*****************************************************************************************
```

Second Program

- **Harmony 3 will autogenerate and set the advertisement data :**

```
/**@brief Advertising data. */
typedef struct BLE_GAP_AdvDataParams_T
{
    uint8_t             advLen;                              /**< Length of advertising data*/
    uint8_t             advData[BLE_GAP_ADV_MAX_LENGTH];      /**< Advertising data */
} BLE_GAP_AdvDataParams_T;

AND: #define BLE_GAP_ADV_MAX_LENGTH                           0x1F   // 31 bytes MAX
```



```
BLE_SMP_Config_T                smpParam;
int8_t                          connTxPower;


int8_t                          advTxPower;
BLE_GAP_AdvParams_T             advParam;
uint8_t advData[]={0x02, 0x01, 0x04, 0x0B, 0x09, 0x70, 0x69, 0x63, 0x33, 0x32, 0x63, 0x78, 0x2D, 0x62, 0x7A, 0x04, 0x16, 0xDA, 0xFE, 0x00};
BLE_GAP_AdvDataParams_T         appAdvData;
uint8_t scanRspData[]={0x0B, 0x09, 0x70, 0x69, 0x63, 0x33, 0x32, 0x63, 0x78, 0x2D, 0x62, 0x7A};
BLE_GAP_AdvDataParams_T         appScanRspData;


BLE_DM_Config_T                 dmConfig;
BLE_GAP_ServiceOption_T         gapServiceOptions;

// Configure device name
BLE_GAP_SetDeviceName(sizeof(devName), devName);


// GAP Service option
gapServiceOptions.charDeviceName.enableWriteProperty = false;
gapServiceOptions.charAppearance.appearance = 0x0;
gapServiceOptions.charPeriPreferConnParam.enable = false;

BLE_GAP_ConfigureBuildInService(&gapServiceOptions);


// Configure advertising parameters
BLE_GAP_SetAdvTxPowerLevel(11,&advTxPower);

memset(&advParam, 0, sizeof(BLE_GAP_AdvParams_T));
advParam.intervalMin = 32;
advParam.intervalMax = 32;
advParam.type = BLE_GAP_ADV_TYPE_ADV_IND;
advParam.advChannelMap = BLE_GAP_ADV_CHANNEL_ALL;
advParam.filterPolicy = BLE_GAP_ADV_FILTER_DEFAULT;
BLE_GAP_SetAdvParams(&advParam);

// Configure advertising data
appAdvData.advLen=sizeof(advData);
memcpy(appAdvData.advData, advData, appAdvData.advLen);
BLE_GAP_SetAdvData(&appAdvData);
```

Advertisement Data

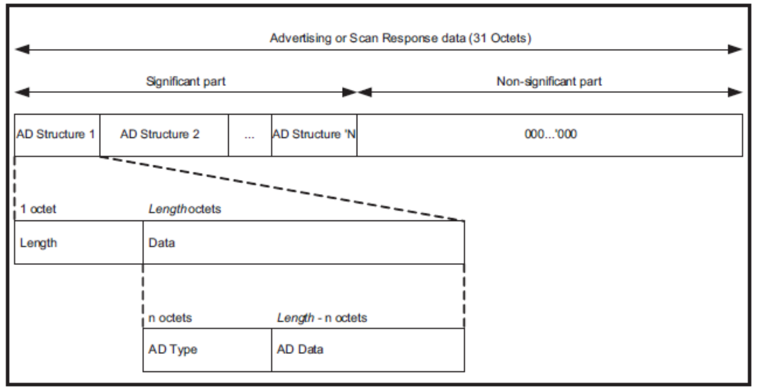
API to adv control parameters

Fill structure with length of advertising data then copy the data into it

API to set advertising data

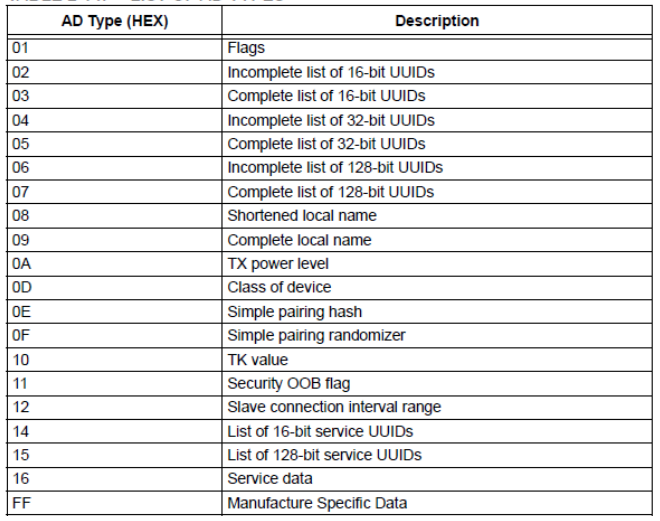
Second Program

- **AD Structures**



- **AD Types**

TABLE 2-14: LIST OF AD TYPES



AD Type (HEX)	Description
01	Flags
02	Incomplete list of 16-bit UUIDs
03	Complete list of 16-bit UUIDs
04	Incomplete list of 32-bit UUIDs
05	Complete list of 32-bit UUIDs
06	Incomplete list of 128-bit UUIDs
07	Complete list of 128-bit UUIDs
08	Shortened local name
09	Complete local name
0A	TX power level
0D	Class of device
0E	Simple pairing hash
0F	Simple pairing randomizer
10	TK value
11	Security OOB flag
12	Slave connection interval range
14	List of 16-bit service UUIDs
15	List of 128-bit service UUIDs
16	Service data
FF	Manufacture Specific Data

advData[]={0x02, 0x01, 0x04, 0x0A, 0x09, 0x4D, 0x69, 0x63, 0x72, 0x6F, 0x63, 0x68, 0x69, 0x70, 0x04, 0x16, 0xFE, 0xDA, 0x0F};

L-T-V format
Length Type Value

02-01-04 → Len=2,Type=Flags, Value =0b 00000100

04-16-FE-DA-0F → Len=4, Type=Service Data, Value=0xFEDA, 0x0F

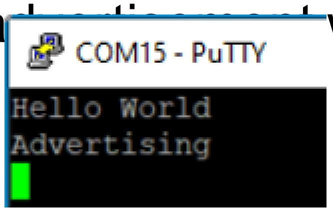
0A-09,'M','i','c','r','o','c','h','i',’p → Len=10,Type=Complete local name,Value=‘Microchip’

Second Program

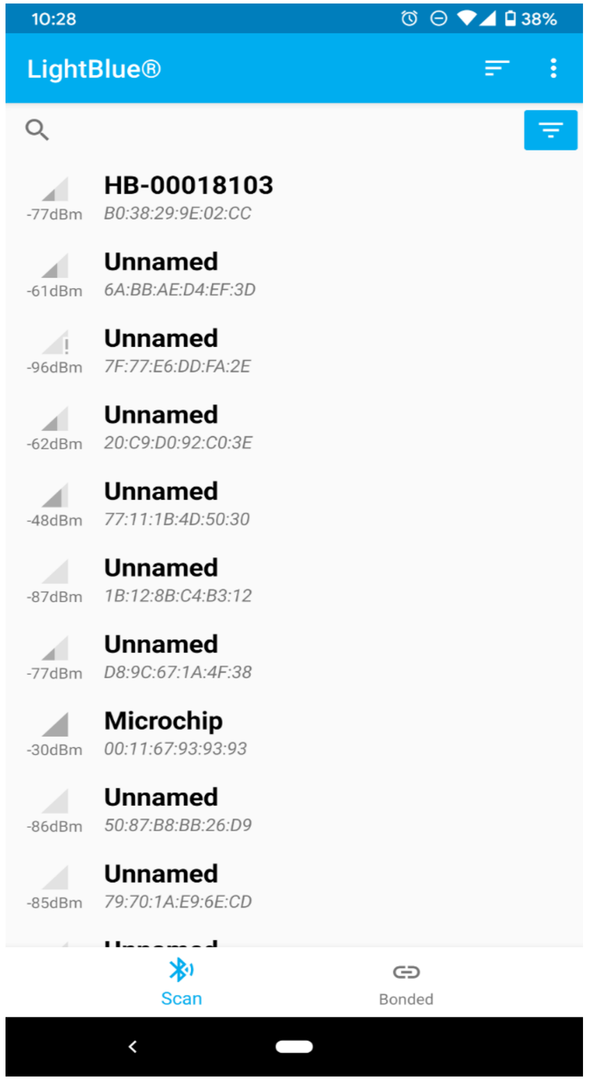
- **Enable Advertisement**
  - Add the following code around line 130 in app.c after “Hello World” print

```
// Start Advertisement
BLE_GAP_SetAdvEnable(0x01, 0x00);
SERCOM0_USART_Write((uint8_t *)"Advertising\r\n", 13);
```

- **Try to Advertise**
  - Compile and Run
  - Reset Bluetooth Settings on Smart Device (Avoid Caching)
  - Open phone BLE App to search the advertisement with device name “Microchip”



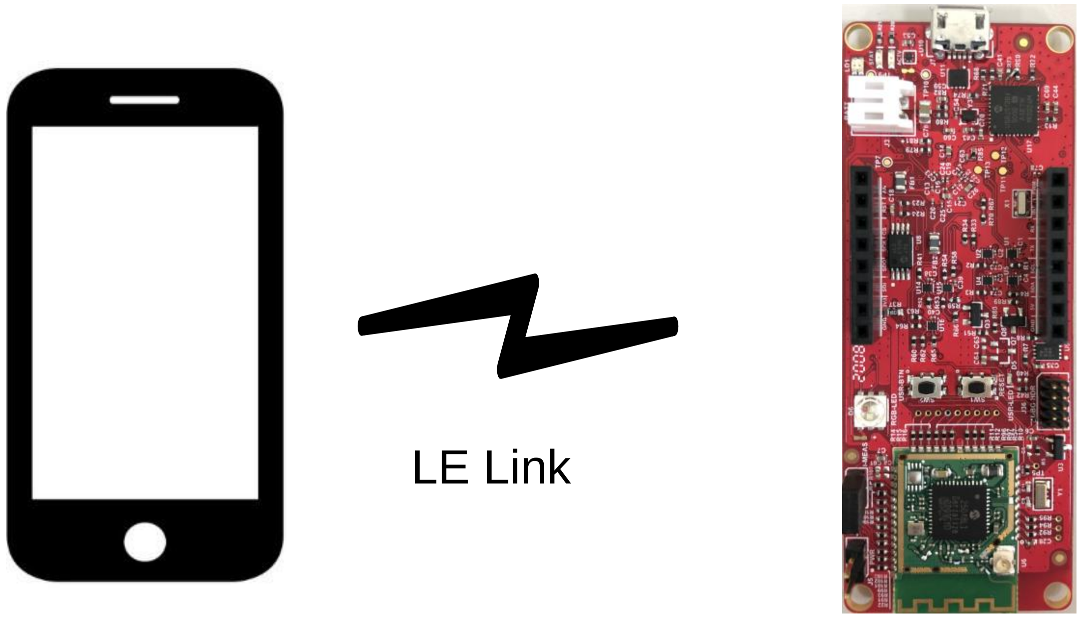
```
Hello World
Advertising
```



```
-77dBm  HB-00018103  B0:38:29:9E:02:CC
-61dBm  Unnamed      6A:BB:AE:D4:EF:3D
-96dBm  Unnamed      7F:77:E6:DD:FA:2E
-62dBm  Unnamed      20:C9:D0:92:C0:3E
-48dBm  Unnamed      77:11:1B:4D:50:30
-87dBm  Unnamed      1B:12:8B:C4:B3:12
-77dBm  Unnamed      D8:9C:67:1A:4F:38
-30dBm  Microchip    00:11:67:93:93:93
-86dBm  Unnamed      50:87:B8:BB:26:D9
-85dBm  Unnamed      79:70:1A:E9:6E:CD
```

Third Program

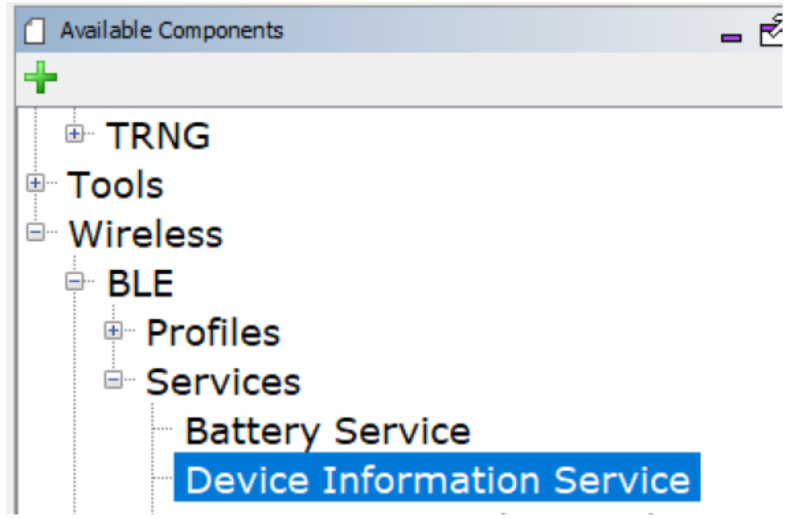
- **Add Device Information Service and display connection status**



Android: LightBlue
iOS: LightBlue

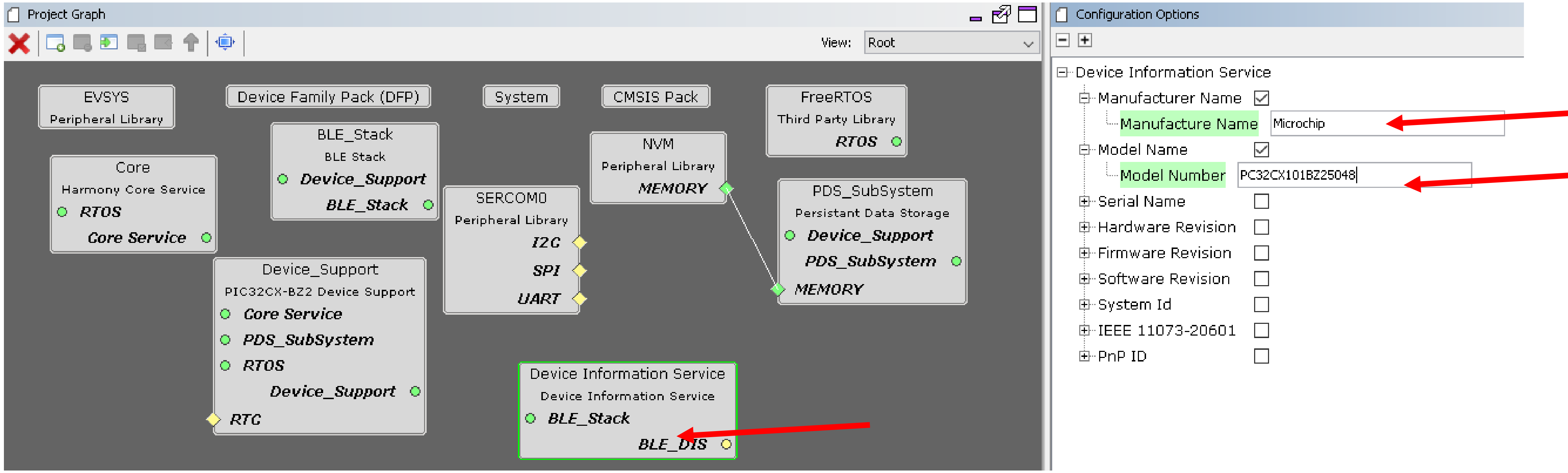
Third Program

- Open MHC window and look for ‘Available Components’ →add Wireless/BLE/Services/Device Information Service



Third Program

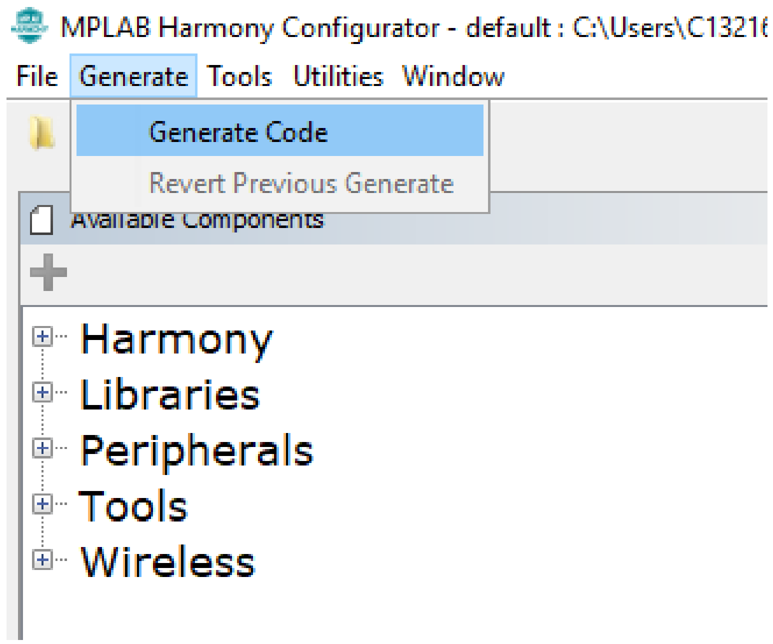
- Configure the BLE Device Info Service as shown



Device Information Service
- Manufacturer Name [✓]
  - Manufacture Name: Microchip
- Model Name [✓]
  - Model Number: PC32CX101BZ25048
- Serial Name [ ]
- Hardware Revision [ ]
- Firmware Revision [ ]
- Software Revision [ ]
- System Id [ ]
- IEEE 11073-20601 [ ]
- PnP ID [ ]

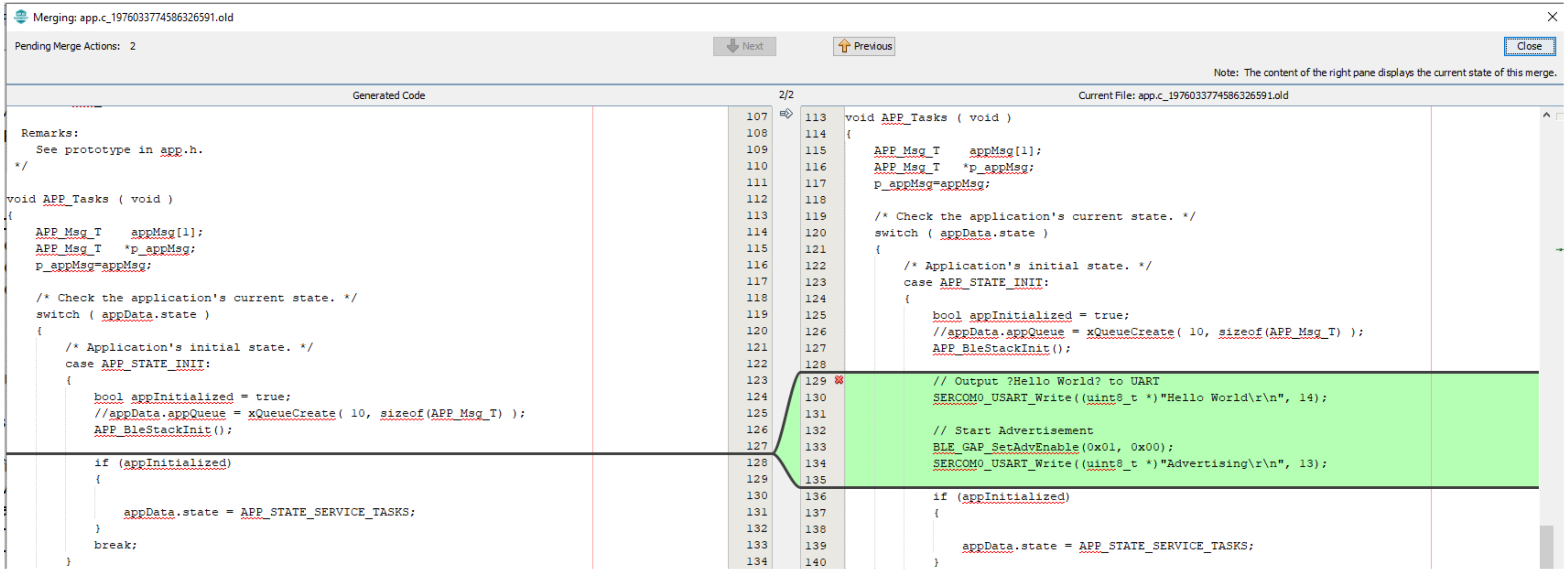
Third Program

- **Regenerate Code**



Third Program

- MHC Merging window shows modified code
- Close the window to retain your edits


Merging: app.c_1976033774586326591.old
Pending Merge Actions: 2
Next
Previous
Close
Note: The content of the right pane displays the current state of this merge.
Generated Code
2/2
Current File: app.c_1976033774586326591.old

```
 Remarks:
   See prototype in app.h.
 */

void APP_Tasks ( void )
{

    APP_Msg_T    appMsg[1];
    APP_Msg_T   *p_appMsg;
    p_appMsg=appMsg;

    /* Check the application's current state. */
    switch ( appData.state )
    {
        /* Application's initial state. */
        case APP_STATE_INIT:
        {
            bool appInitialized = true;
            //appData.appQueue = xQueueCreate( 10, sizeof(APP_Msg_T) );
            APP_BleStackInit();

            if (appInitialized)
            {

                appData.state = APP_STATE_SERVICE_TASKS;
            }
            break;
        }
```

```
void APP_Tasks ( void )
{
    APP_Msg_T    appMsg[1];
    APP_Msg_T   *p_appMsg;
    p_appMsg=appMsg;

    /* Check the application's current state. */
    switch ( appData.state )
    {
        /* Application's initial state. */
        case APP_STATE_INIT:
        {
            bool appInitialized = true;
            //appData.appQueue = xQueueCreate( 10, sizeof(APP_Msg_T) );
            APP_BleStackInit();

            // Output ?Hello World? to UART
            SERCOM0_USART_Write((uint8_t *)"Hello World\r\n", 14);

            // Start Advertisement
            BLE_GAP_SetAdvEnable(0x01, 0x00);
            SERCOM0_USART_Write((uint8_t *)"Advertising\r\n", 13);

            if (appInitialized)
            {

                appData.state = APP_STATE_SERVICE_TASKS;
            }
```

Third Program

In app.c, include header file

```
#include "ble_dis.h"
```

After printout of “Hello, World”, before starting advertisement, Add following code to register Device Information service

```
// Register Device Info Service
BLE_DIS_Add();
```

Third Program

Notification of Connection/Disconnection

In file app_ble_handler.c, include header file

```
#include "definitions.h"
```

Add this global variable in the Section:Global Variables near the top:

```
extern uint16_t conn_hdl = 0xFFFF;
```

We will need this later, but since we are here why not?

Third Program

Then find function APP_BleGapEvtHandler, and replace the BLE_GAP_EVT_CONNECTED: and BLE_GAP_EVT_DISCONNECTED: cases with this

```
case BLE_GAP_EVT_CONNECTED:
{
   SERCOM0_USART_Write((uint8_t *)"Connected\r\n", 11);
   conn_hdl = p_event->eventField.evtConnect.connHandle;
}
break;

case BLE_GAP_EVT_DISCONNECTED:
{
  SERCOM0_USART_Write((uint8_t *)"Disconnected\r\n", 14);
}
break;
```

Let's save this handle for later…

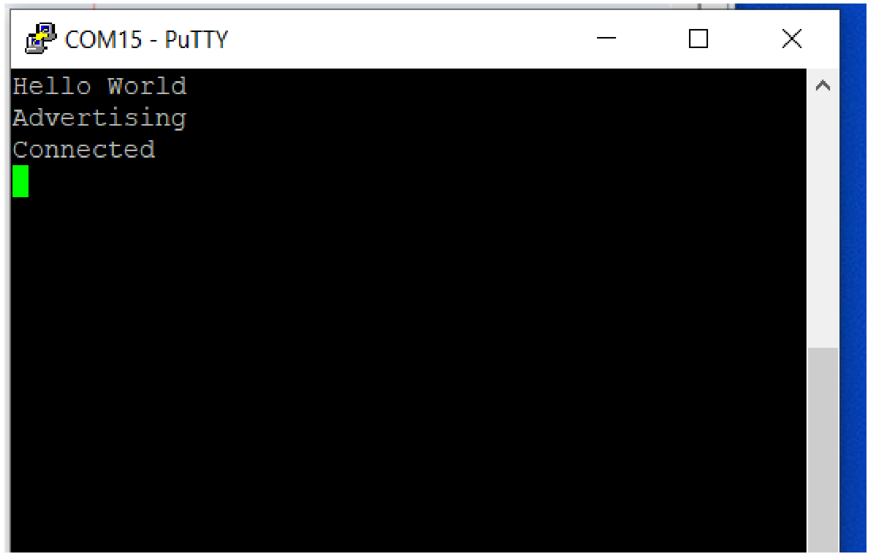
Third Program

Now your code should look like this:

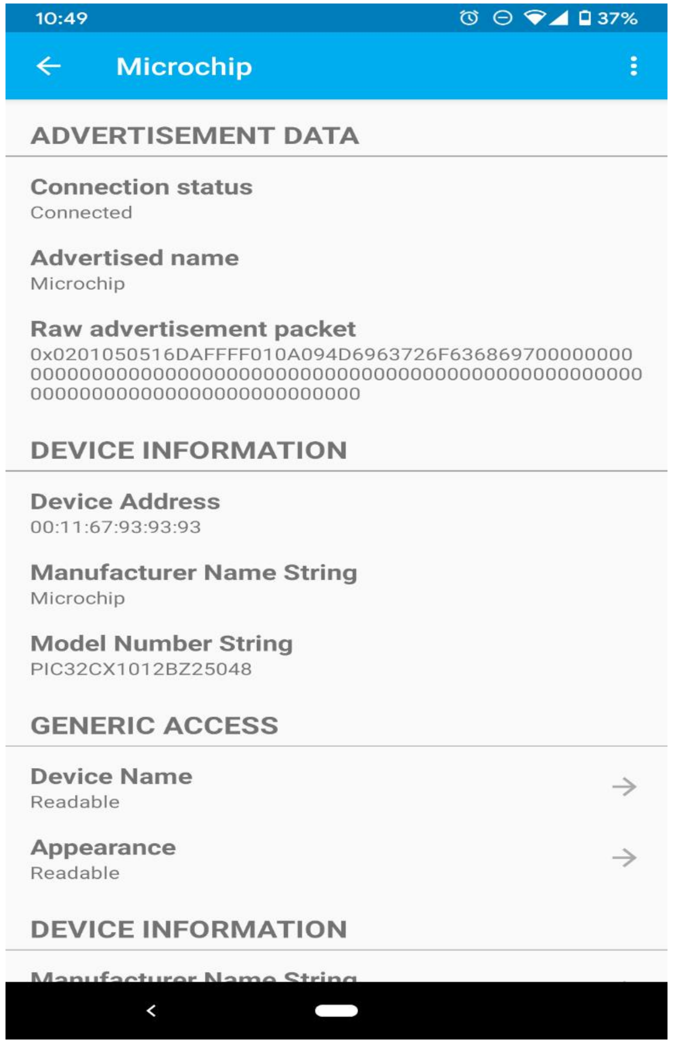
```
// Section: Global Variables
// *****************************************************************************
// *****************************************************************************

extern uint16_t conn_hdl = 0xFFFF;

// Section: Functions
// *****************************************************************************
// *****************************************************************************
void APP_BleGapEvtHandler(BLE_GAP_Event_T *p_event)
{
    switch(p_event->eventId)
    {
        case BLE_GAP_EVT_CONNECTED:
        {
            /* TODO: implement your application code.*/
          SERCOM0_USART_Write((uint8_t *)"Connected\r\n", 11);
          conn_hdl = p_event->eventField.evtConnect.connHandle;

        }
        break;

        case BLE_GAP_EVT_DISCONNECTED:
        {
            /* TODO: implement your application code.*/
          SERCOM0_USART_Write((uint8_t *)"Disconnected\r\n", 14);
        }
        break;
```

Third Program

- **Compile and Run the Project**
- **Reset Bluetooth Settings on Smart Device (Avoid Caching)**
- **Open smart phone App and try Connecting**
  - Check Connection Message on UART
  - Check Device Info Service display on Smart Device



```
Hello World
Advertising
Connected
```



ADVERTISEMENT DATA

Connection status
Connected

Advertised name
Microchip

Raw advertisement packet
0x0201050516DAFFFF010A094D6963726F636869700000000
000000000000000000000000000000000000000000000000
000000000000000000000000000000

DEVICE INFORMATION

Device Address
00:11:67:93:93:93

Manufacturer Name String
Microchip

Model Number String
PIC32CX1012BZ25048

GENERIC ACCESS

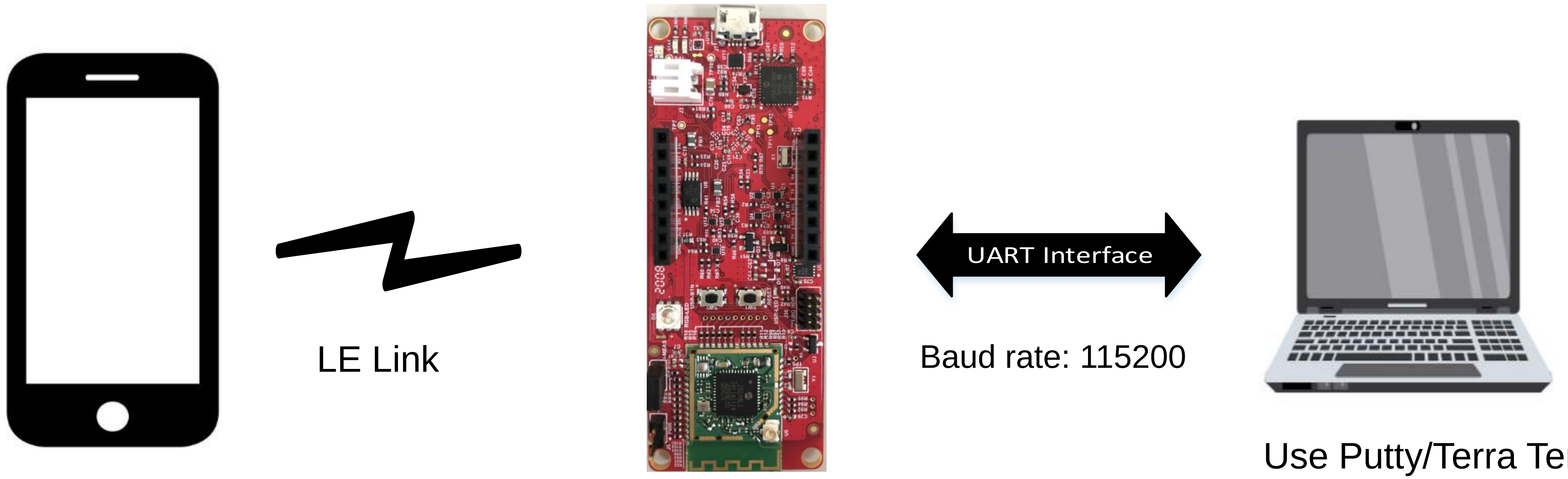
Device Name
Readable

Appearance
Readable

DEVICE INFORMATION

Fourth Program

- **Transparent UART service**



LE Link

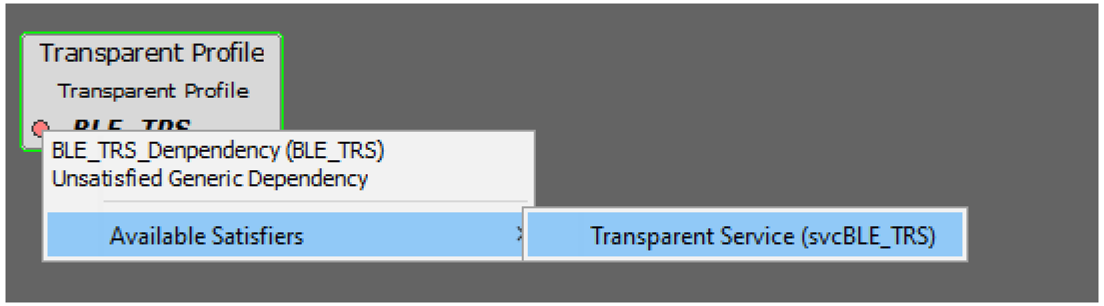
Baud rate: 115200

Android: LightBlue
iOS: LightBlue

Use Putty/Terra Term, other serial port emulator

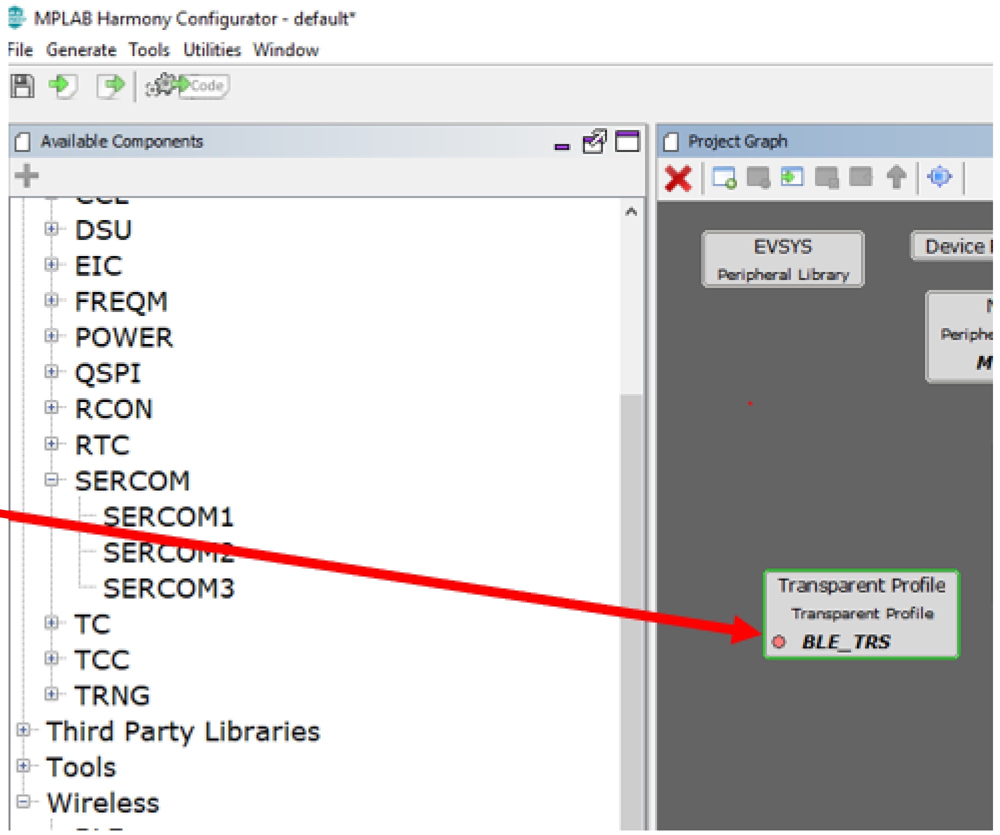
Fourth Program

- In MHC ‘Available Components’ add “Wireless/BLE/Profiles/Transparent Profile”.

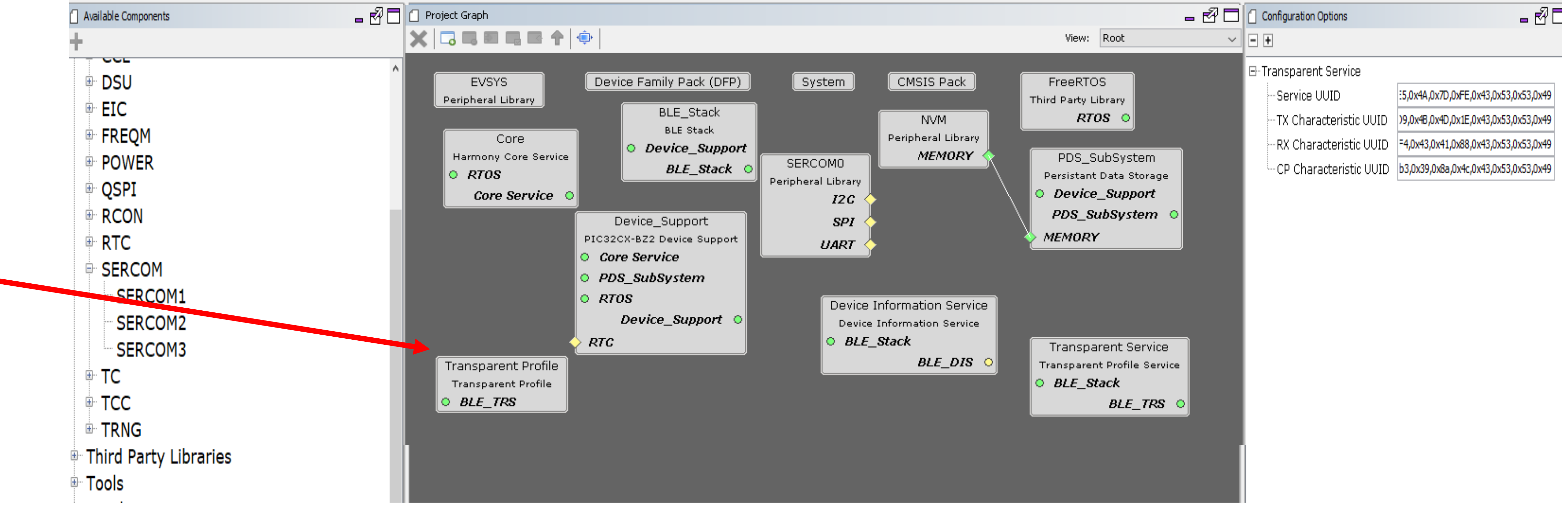

Transparent Profile
Transparent Profile
BLE_TRS
BLE_TRS_Denpendency (BLE_TRS)
Unsatisfied Generic Dependency
Available Satisfiers
Transparent Service (svcBLE_TRS)

Fourth Program

Right click the red dot to add dependency of Transparent Service



Fourth Program



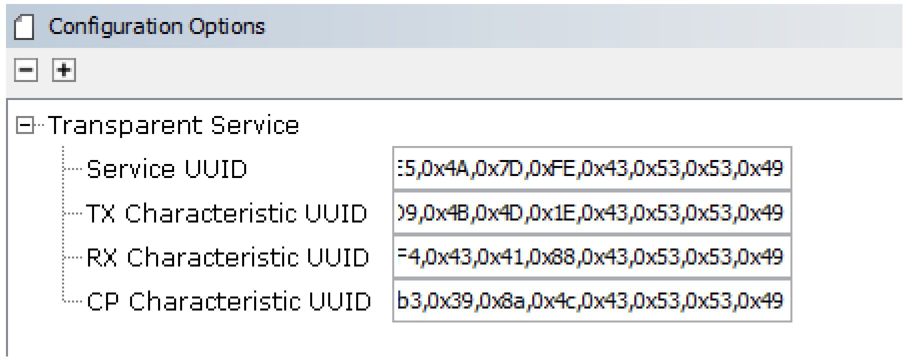
Transparent Service

Service UUID	:5,0x4A,0x7D,0xFE,0x43,0x53,0x53,0x49
TX Characteristic UUID	)9,0x4B,0x4D,0x1E,0x43,0x53,0x53,0x49
RX Characteristic UUID	F4,0x43,0x41,0x88,0x43,0x53,0x53,0x49
CP Characteristic UUID	b3,0x39,0x8a,0x4c,0x43,0x53,0x53,0x49

Fourth Program

- **Configure Transparent Service/Profile**

NO CHANGES

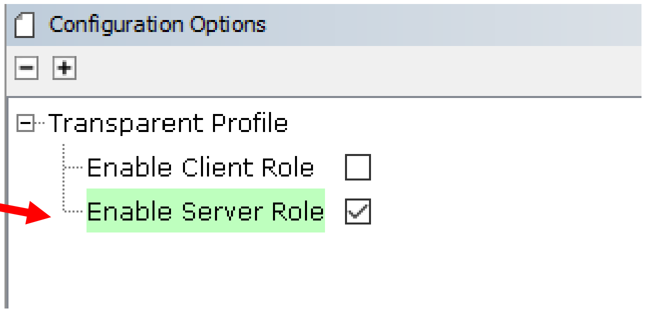


Configuration Options

Transparent Service

Service UUID	:5,0x4A,0x7D,0xFE,0x43,0x53,0x53,0x49
TX Characteristic UUID	)9,0x4B,0x4D,0x1E,0x43,0x53,0x53,0x49
RX Characteristic UUID	=4,0x43,0x41,0x88,0x43,0x53,0x53,0x49
CP Characteristic UUID	b3,0x39,0x8a,0x4c,0x43,0x53,0x53,0x49

ENABLE SERVER ROLE



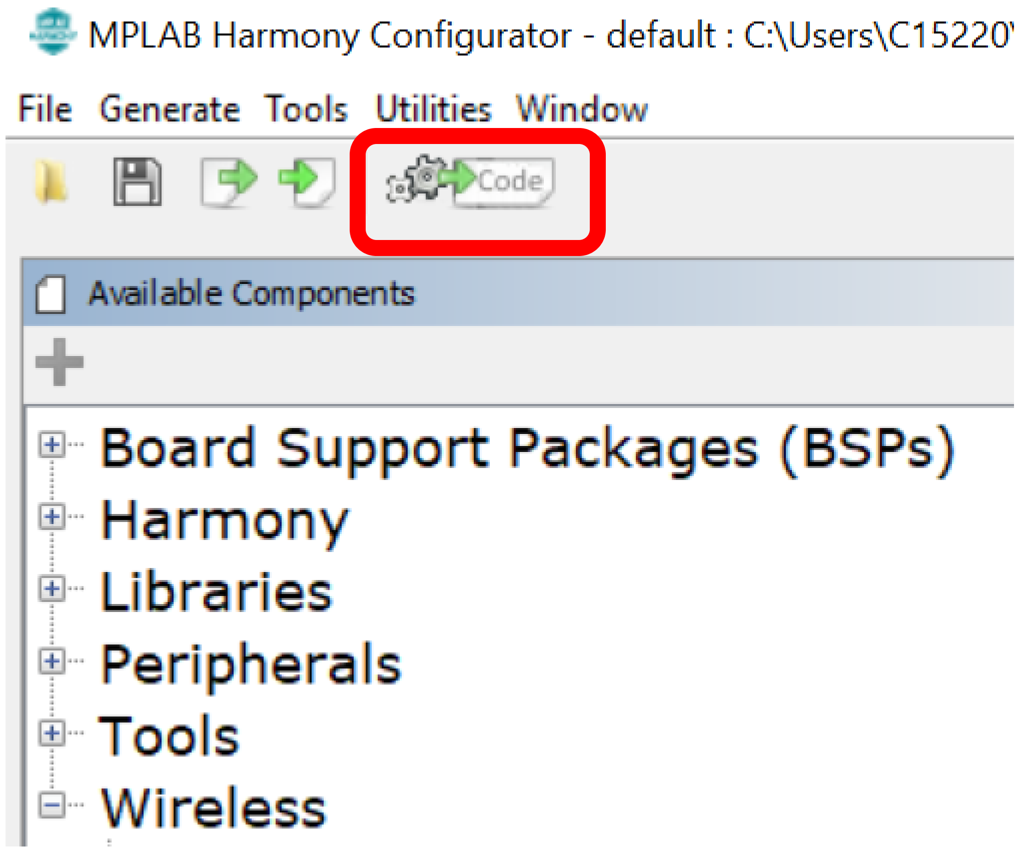
Configuration Options

Transparent Profile

- Enable Client Role ☐
- Enable Server Role ☑

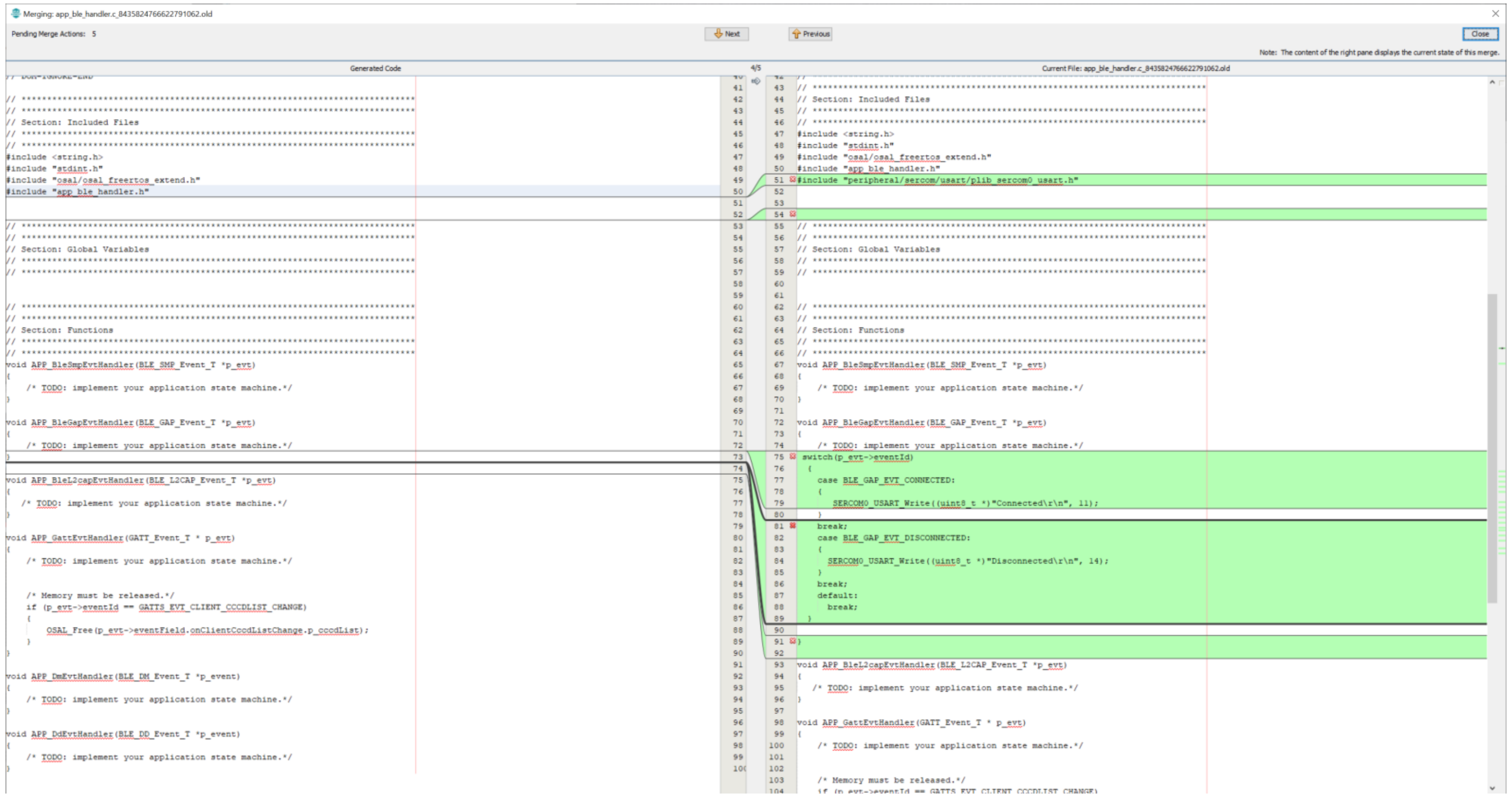
Fourth Program

- **Regenerate Code – you will have to hit 'Close' twice**


MPLAB Harmony Configurator - default : C:\Users\C15220\
File Generate Tools Utilities Window
Code
Available Components
Board Support Packages (BSPs)
Harmony
Libraries
Peripherals
Tools
Wireless

Fourth Program

- **MHC Merging windows shows modified code**
- **Close the windows to retain your edits**



Merging: app_ble_handler.c_8435824766622791062.old

Pending Merge Actions: 5

Note: The content of the right pane displays the current state of this merge.

Generated Code

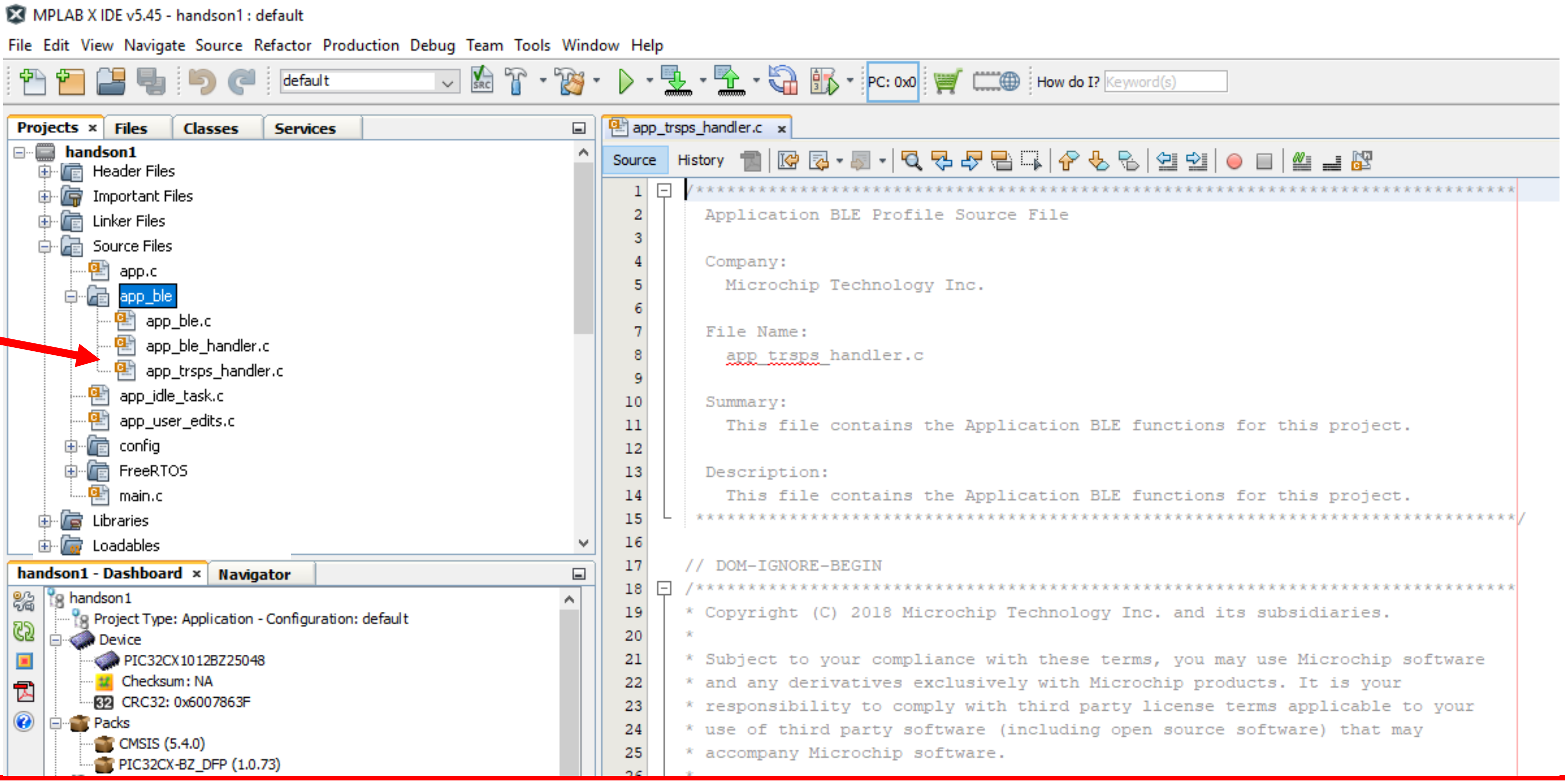
```
// ***************************************************************************
// ***************************************************************************
// Section: Included Files
// ***************************************************************************
// ***************************************************************************
#include <string.h>
#include "stdint.h"
#include "osal/osal_freertos_extend.h"
#include "app_ble_handler.h"


// ***************************************************************************
// ***************************************************************************
// Section: Global Variables
// ***************************************************************************
// ***************************************************************************


// ***************************************************************************
// ***************************************************************************
// Section: Functions
// ***************************************************************************
// ***************************************************************************
void APP_BleSmpEvtHandler(BLE_SMP_Event_T *p_evt)
{
    /* TODO: implement your application state machine.*/
}

void APP_BleGapEvtHandler(BLE_GAP_Event_T *p_evt)
{
    /* TODO: implement your application state machine.*/
}

void APP_BleL2capEvtHandler(BLE_L2CAP_Event_T *p_evt)
{
   /* TODO: implement your application state machine.*/
}

void APP_GattEvtHandler(GATT_Event_T * p_evt)
{
    /* TODO: implement your application state machine.*/


    /* Memory must be released.*/
    if (p_evt->eventId == GATTS_EVT_CLIENT_CCCDLIST_CHANGE)
    {
        OSAL_Free(p_evt->eventField.onClientCccdListChange.p_cccdList);
    }
}

void APP_DmEvtHandler(BLE_DM_Event_T *p_event)
{
    /* TODO: implement your application state machine.*/
}

void APP_DdEvtHandler(BLE_DD_Event_T *p_event)
{
    /* TODO: implement your application state machine.*/
}
```

Current File: app_ble_handler.c_8435824766622791062.old

```
// ***************************************************************************
// Section: Included Files
// ***************************************************************************
// ***************************************************************************
#include <string.h>
#include "stdint.h"
#include "osal/osal_freertos_extend.h"
#include "app_ble_handler.h"
#include "peripheral/sercom/usart/plib_sercom0_usart.h"



// ***************************************************************************
// ***************************************************************************
// Section: Global Variables
// ***************************************************************************
// ***************************************************************************


// ***************************************************************************
// ***************************************************************************
// Section: Functions
// ***************************************************************************
// ***************************************************************************
void APP_BleSmpEvtHandler(BLE_SMP_Event_T *p_evt)
{
    /* TODO: implement your application state machine.*/
}

void APP_BleGapEvtHandler(BLE_GAP_Event_T *p_evt)
{
    /* TODO: implement your application state machine.*/
 switch(p_evt->eventId)
  {
    case BLE_GAP_EVT_CONNECTED:
    {
      SERCOM0_USART_Write((uint8_t *)"Connected\r\n", 11);
    }
    break;
    case BLE_GAP_EVT_DISCONNECTED:
    {
     SERCOM0_USART_Write((uint8_t *)"Disconnected\r\n", 14);
    }
    break;
    default:
     break;
  }

}

void APP_BleL2capEvtHandler(BLE_L2CAP_Event_T *p_evt)
{
   /* TODO: implement your application state machine.*/
}

void APP_GattEvtHandler(GATT_Event_T * p_evt)
{
    /* TODO: implement your application state machine.*/


    /* Memory must be released.*/
    if (p_evt->eventId == GATTS_EVT_CLIENT_CCCDLIST_CHANGE)
```

Fourth Program

- Minimize MHC window and Maximize MPLABX window
- In Source Files find app trsps handler.c and open it



```
/*******************************************************************************
  Application BLE Profile Source File

  Company:
    Microchip Technology Inc.

  File Name:
    app_trsps_handler.c

  Summary:
    This file contains the Application BLE functions for this project.

  Description:
    This file contains the Application BLE functions for this project.
 *******************************************************************************/


// DOM-IGNORE-BEGIN
/*******************************************************************************
* Copyright (C) 2018 Microchip Technology Inc. and its subsidiaries.
*
* Subject to your compliance with these terms, you may use Microchip software
* and any derivatives exclusively with Microchip products. It is your
* responsibility to comply with third party license terms applicable to your
* use of third party software (including open source software) that may
* accompany Microchip software.
```

Fourth Program

In app_trsps_handler.c, include header file

```
#include "definitions.h"
```

Add this global variable in the Section:Global Variables near the top:

```
uint16_t conn_hdl;
```

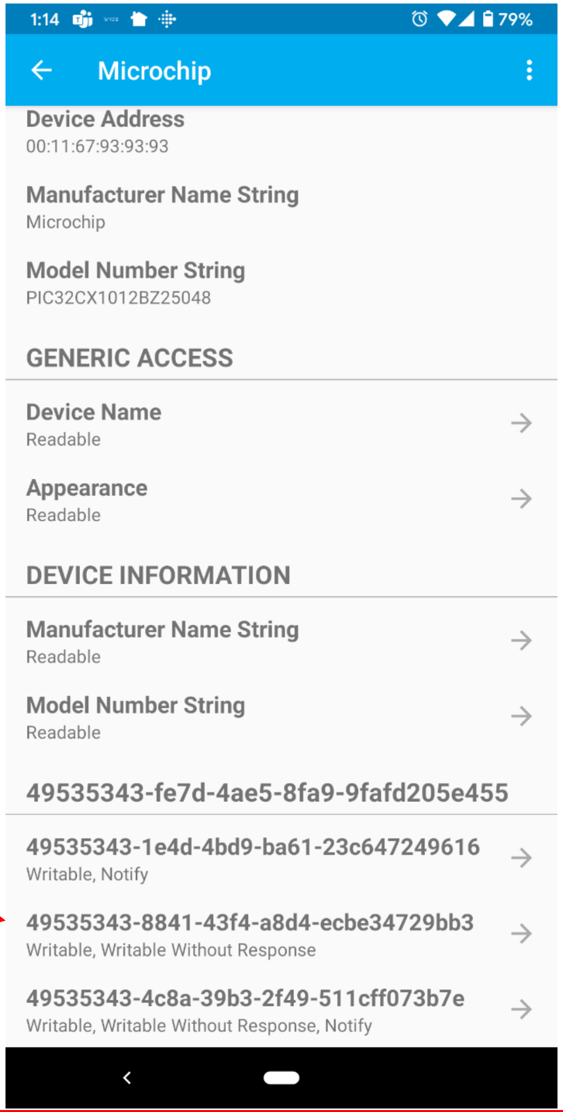
Add implementation of Transparent Service handling in callback function APP_TrspsEvtHandler

```
case BLE_TRSPS_EVT_RECEIVE_DATA:
   {
          uint16_t data_len;
          uint8_t *data;
          // Retrieve received data length
          BLE_TRSPS_GetDataLength(conn_hdl, &data_len);
          // Allocate memory according to data length
          data = OSAL_Malloc(data_len);
          if( data == NULL )
            break;
          // Retrieve received data
          BLE_TRSPS_GetData(conn_hdl, data);
          // Output received data to UART
          SERCOM0_USART_Write(data, data_len);
          // Free memory
          OSAL_Free(data);
    }
```

Fourth Program

- **Compile and Run the Project**
- **Reset Bluetooth Settings on Smart Device (Avoid Caching)**
- **Open LightBlue and connect to ‘Microchip’**
  - Check Connecting Message on UART
  - Using the SECOND UUID Characteristic (Writable,Writable Without Response), select Data Format UTF-8 String and type anything you want in Written Values followed by WRITE, the text should appear on the Terminal

Fourth Program


1:14 79%

Microchip

Device Address
00:11:67:93:93:93

Manufacturer Name String
Microchip

Model Number String
PIC32CX1012BZ25048

GENERIC ACCESS

Device Name
Readable

Appearance
Readable

DEVICE INFORMATION

Manufacturer Name String
Readable

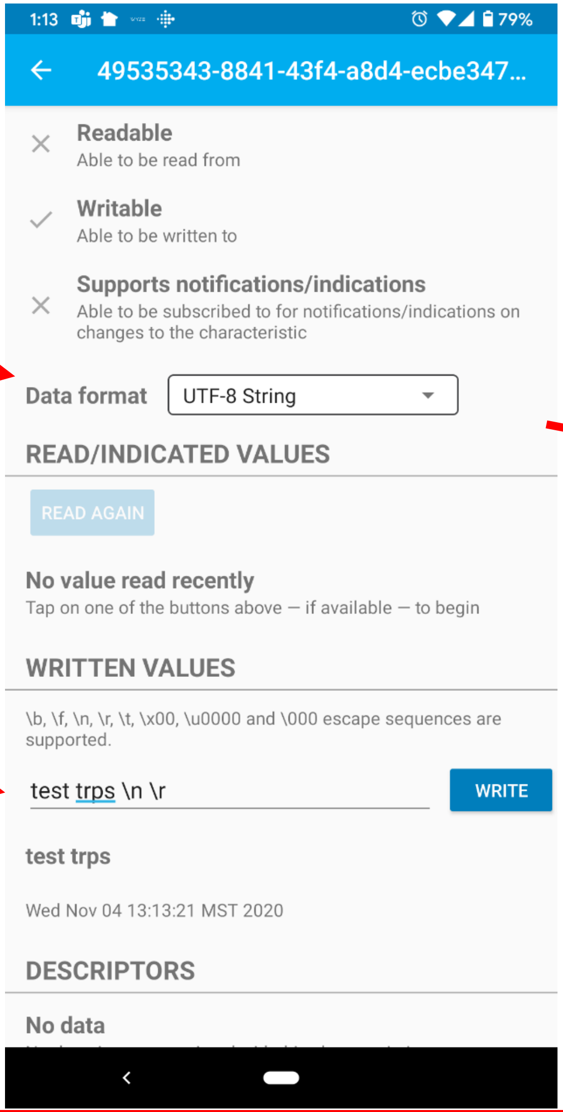
Model Number String
Readable

49535343-fe7d-4ae5-8fa9-9fafd205e455

49535343-1e4d-4bd9-ba61-23c647249616
Writable, Notify

49535343-8841-43f4-a8d4-ecbe34729bb3
Writable, Writable Without Response

49535343-4c8a-39b3-2f49-511cff073b7e
Writable, Writable Without Response, Notify


1:13 79%

49535343-8841-43f4-a8d4-ecbe347...

Readable
Able to be read from

Writable
Able to be written to

Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

Data format: UTF-8 String

READ/INDICATED VALUES

READ AGAIN

No value read recently
Tap on one of the buttons above — if available — to begin

WRITTEN VALUES

\b, \f, \n, \r, \t, \x00, \u0000 and \000 escape sequences are supported.

test trps \n \r — WRITE

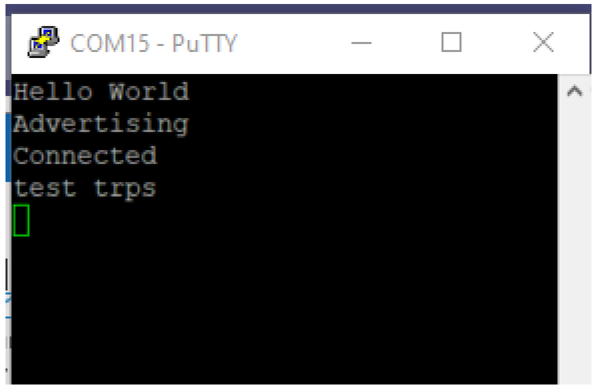
test trps

Wed Nov 04 13:13:21 MST 2020

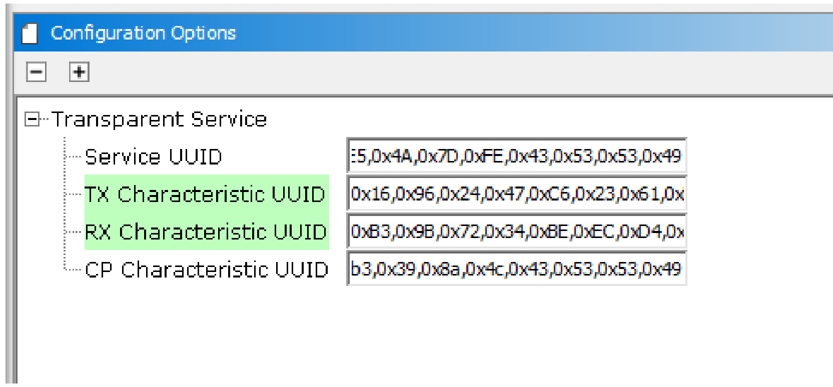
DESCRIPTORS

No data

YEAH!!!


COM15 - PuTTY

```
Hello World
Advertising
Connected
test trps
```


Configuration Options

Transparent Service	
Service UUID	5,0x4A,0x7D,0xFE,0x43,0x53,0x53,0x49
TX Characteristic UUID	0x16,0x96,0x24,0x47,0xC6,0x23,0x61,0x
RX Characteristic UUID	0xB3,0x9B,0x72,0x34,0xBE,0xEC,0xD4,0x
CP Characteristic UUID	b3,0x39,0x8a,0x4c,0x43,0x53,0x53,0x49

Fourth Program

- **Possible issues: Smart phone Bluetooth may need to be turned off/on to flush cache**

Fourth Program, cont’d

Now let’s have more fun with Transparent Service!

In app.c, before printout “Hello, World”, add UART RX initialization

```
// Enable UART Read
SERCOM0_USART_ReadNotificationEnable(true, true);
// Set UART RX notification threshold to be 1
SERCOM0_USART_ReadThresholdSet(1);
// Register the UART RX callback function
SERCOM0_USART_ReadCallbackRegister(uart_cb, (uintptr_t)NULL);
```

**** NOTE THE THRESHOLD = 1 ** means you can only type in 1 char at a time**

Fourth Program, con’td

In app.c, include header file

```
#include "ble_trsps.h"
```

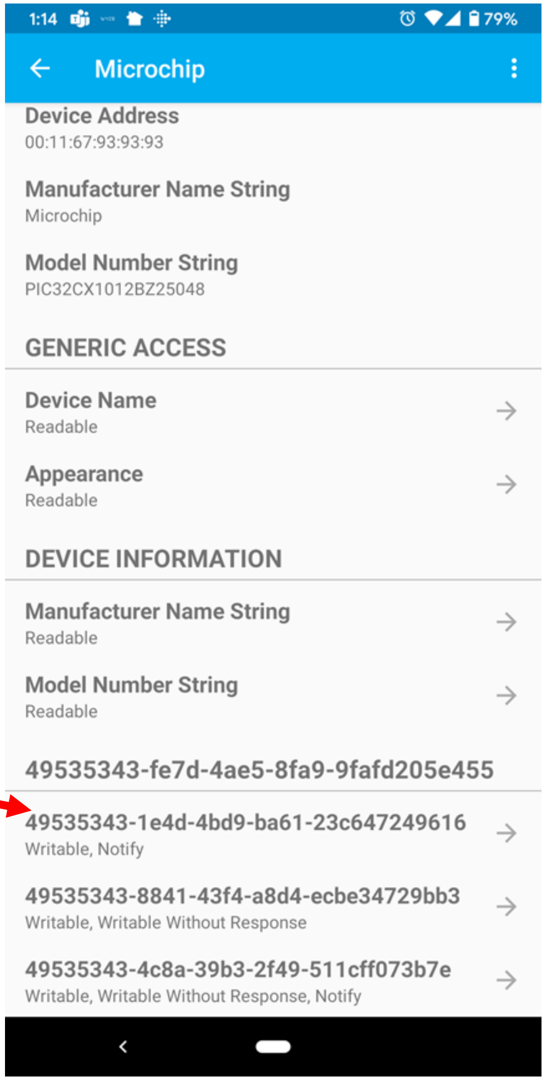
Add UART callback function after “Section: Application Callback Functions”

```
// refer to connection handle
uint16_t conn_hdl;

void uart_cb(SERCOM_USART_EVENT event, uintptr_t context)
{
  // If RX data from UART reached threshold (previously set to 1)
  if( event == SERCOM_USART_EVENT_READ_THRESHOLD_REACHED )
  {
    uint8_t data;
    // Read 1 byte data from UART
    SERCOM0_USART_Read(&data, 1);
    // Send the data from UART to connected device through Transparent service
    BLE_TRSPS_SendData(conn_hdl, 1, &data);
  }
}
```

Fourth Program, con’td

- **Compile and Run the Project**
- **Reset Bluetooth Settings on Smart Device (Avoid Caching)**
- **Open LightBlue and Try Connecting**
  - Check Connecting Message on UART
  - Subscribe to the FIRST UUID
  - Try Typing Characters in Terminal Emulator **1 per second (‘h, i’)**
  - Observe the Typed Characters Display on mobile phone app
- **See next slide for details**

Fourth Program, con’td



1:14 79%

Microchip

Device Address
00:11:67:93:93:93

Manufacturer Name String
Microchip

Model Number String
PIC32CX1012BZ25048

GENERIC ACCESS

Device Name
Readable

Appearance
Readable

DEVICE INFORMATION

Manufacturer Name String
Readable

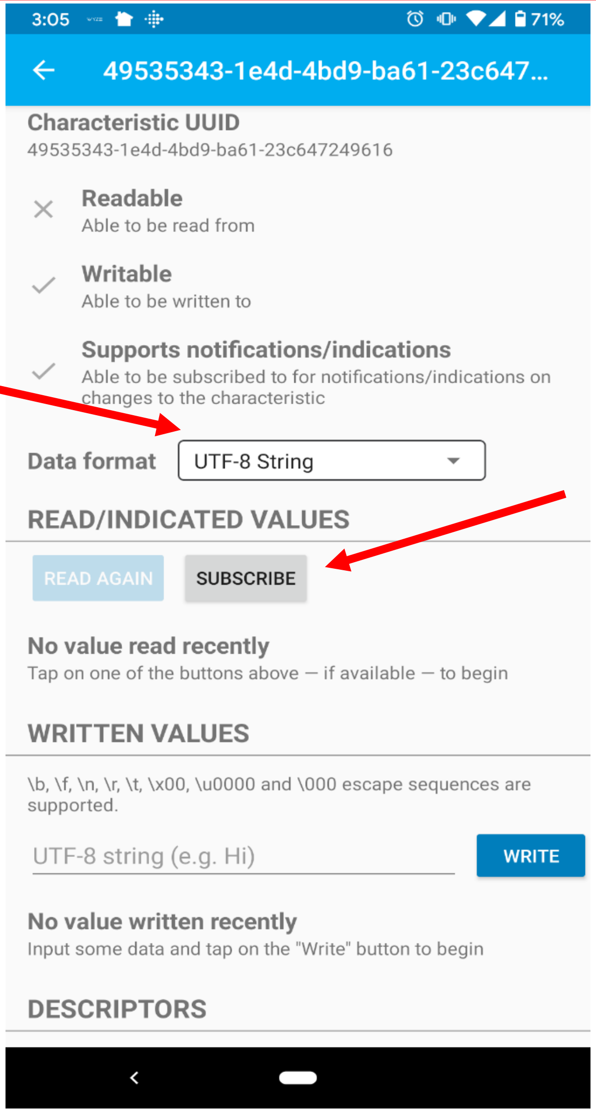
Model Number String
Readable

49535343-fe7d-4ae5-8fa9-9fafd205e455

49535343-1e4d-4bd9-ba61-23c647249616
Writable, Notify

49535343-8841-43f4-a8d4-ecbe34729bb3
Writable, Writable Without Response

49535343-4c8a-39b3-2f49-511cff073b7e
Writable, Writable Without Response, Notify



3:05 71%

49535343-1e4d-4bd9-ba61-23c647...

Characteristic UUID
49535343-1e4d-4bd9-ba61-23c647249616

✗ Readable
Able to be read from

✓ Writable
Able to be written to

✓ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

Data format: UTF-8 String

READ/INDICATED VALUES

READ AGAIN | SUBSCRIBE

No value read recently
Tap on one of the buttons above — if available — to begin

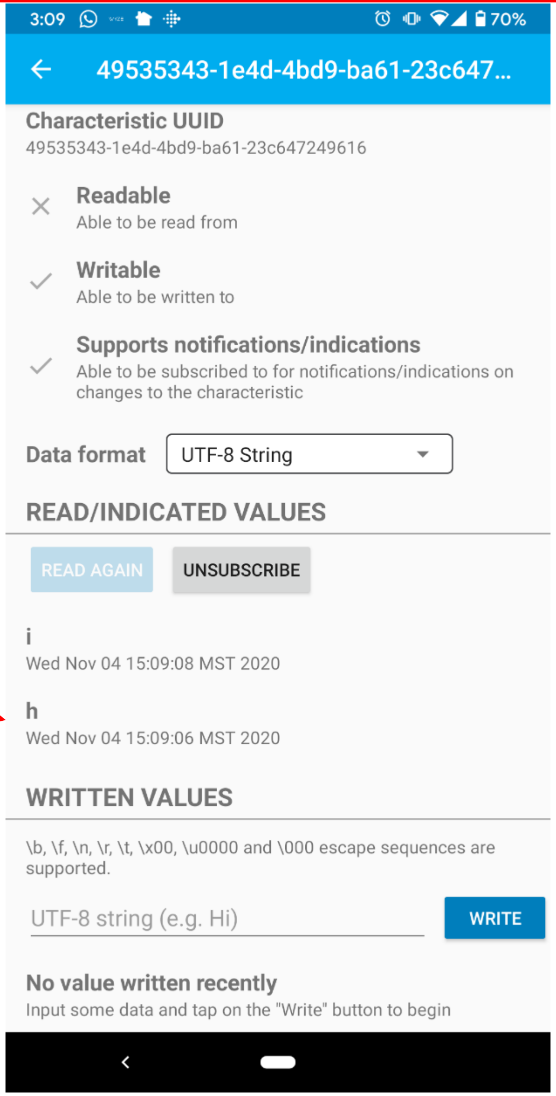
WRITTEN VALUES

\b, \f, \n, \r, \t, \x00, \u0000 and \000 escape sequences are supported.

UTF-8 string (e.g. Hi) | WRITE

No value written recently
Input some data and tap on the "Write" button to begin

DESCRIPTORS



3:09 70%

49535343-1e4d-4bd9-ba61-23c647...

Characteristic UUID
49535343-1e4d-4bd9-ba61-23c647249616

✗ Readable
Able to be read from

✓ Writable
Able to be written to

✓ Supports notifications/indications
Able to be subscribed to for notifications/indications on changes to the characteristic

Data format: UTF-8 String

READ/INDICATED VALUES

READ AGAIN | UNSUBSCRIBE

i
Wed Nov 04 15:09:08 MST 2020

h
Wed Nov 04 15:09:06 MST 2020

WRITTEN VALUES

\b, \f, \n, \r, \t, \x00, \u0000 and \000 escape sequences are supported.

UTF-8 string (e.g. Hi) | WRITE

No value written recently
Input some data and tap on the "Write" button to begin

End of Tutorial

- Thanks!