

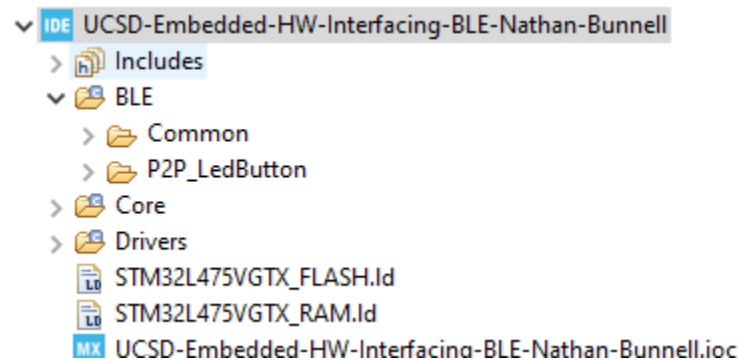
Date: 6/2/2021

Assignment 8: BLE Hands On

The following will document completion of the eighth assignment for ECE-40293, using the onboard hardware to complete the following task:

You are to implement the P2P_LedButton project that was covered in the Lesson Hands-On Section. The P2P_LedButton project is included in the STM32Cube repository.

To implement the P2P_LedButton project, we will generate our default project file as in all other lessons and then begin the process of importing the necessary support files and deleting unneeded or duplicate files. First, drag and drop the BLE folder from the projects file, using the same method we used to complete the WiFi project. We will only need the Common and P2P_LedButton folders out of this directory.

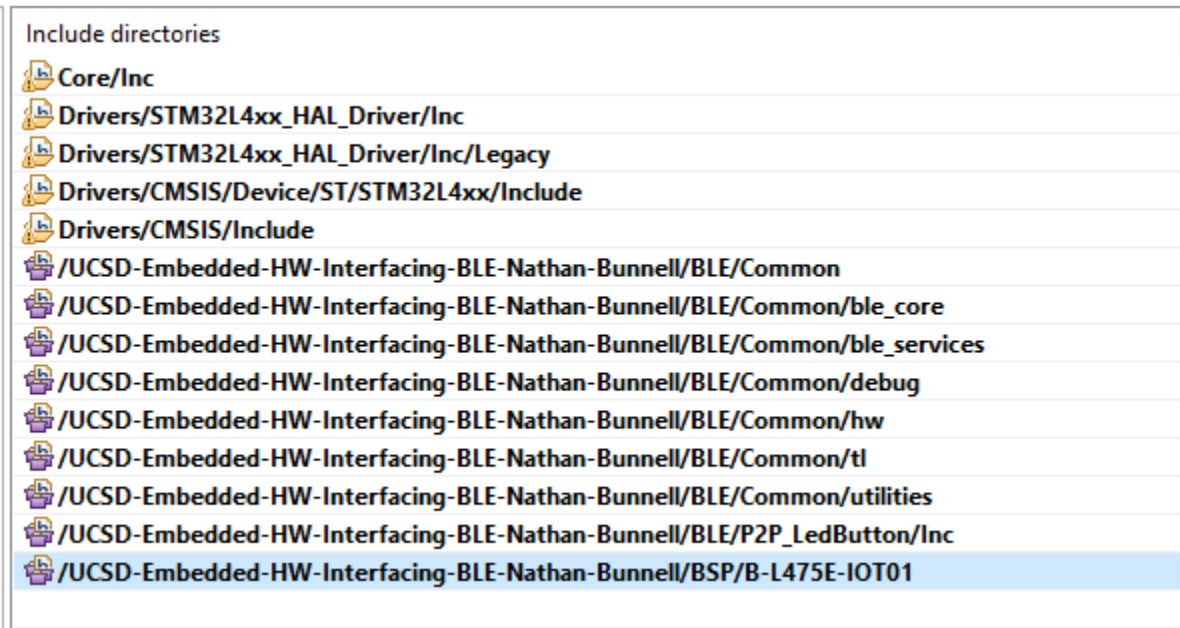


We will also need to update our source and include paths under the project properties to look at the new file locations:

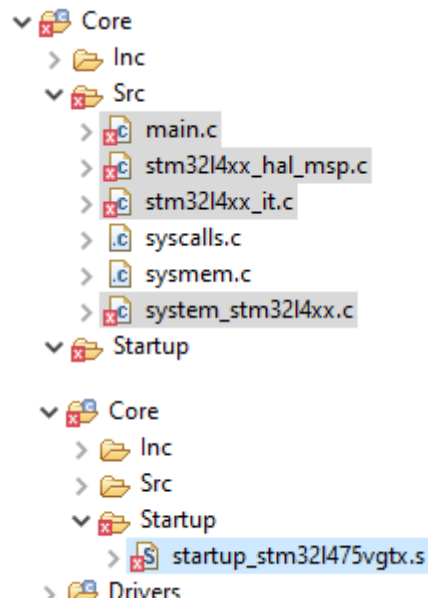
Source folders on build path:

- > /UCSD-Embedded-HW-Interfacing-BLE-Nathan-Bunnell/BLE
- > /UCSD-Embedded-HW-Interfacing-BLE-Nathan-Bunnell/BSP
- > /UCSD-Embedded-HW-Interfacing-BLE-Nathan-Bunnell/Core
- > /UCSD-Embedded-HW-Interfacing-BLE-Nathan-Bunnell/Drivers

Date: 6/2/2021



We can then begin to delete the default project files that will be replaced by the files contained in the BLE folder, deleting these files from the Core directory:



Date: 6/2/2021

We'll also pull in the BSP files appropriate for our board, ensuring that the correct firmware reference is taken, 1.17 in my case.

UCSD-Embedded-HW-Interfacing-BLE-Nathan-Bunnell > BSP		
	Name ^	Date modified
Bunnell, Inc	B-L475E-IOT01	6/1/2021
	Components	6/1/2021

Next, we move into the code and ensure the appropriate #defines and #includes are in place, in this case, enabling the RTC and adding a call to include the LL exti header file, as well as ensuring the device is configured to be in server mode when running.

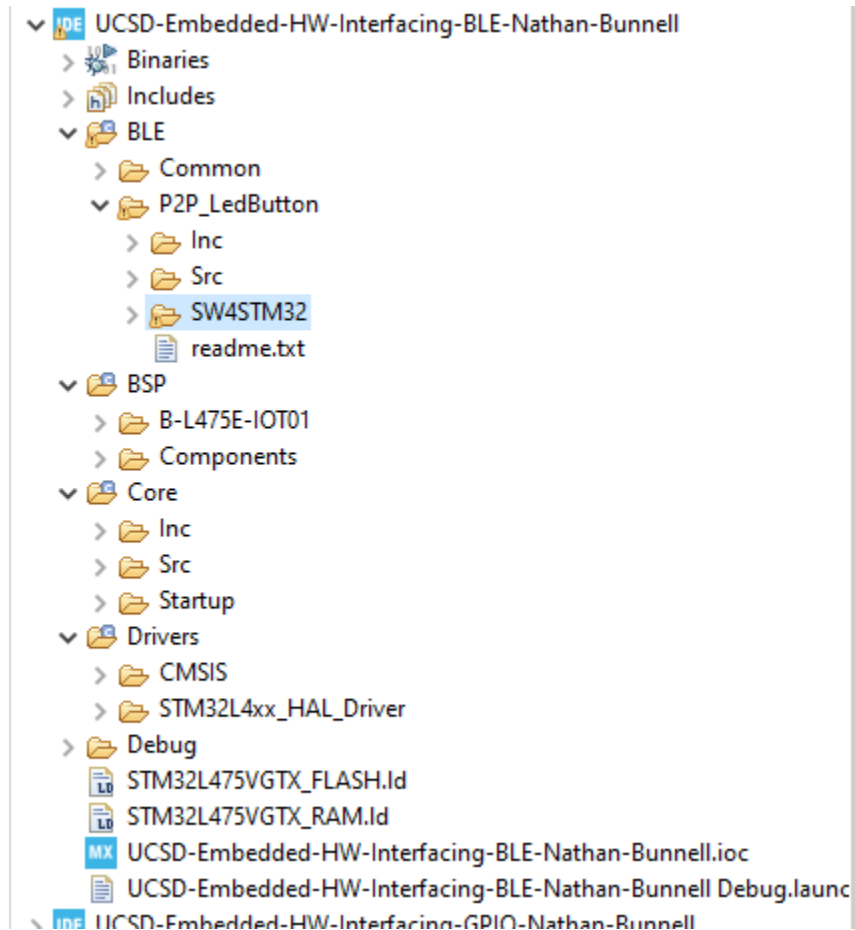
```
69 /*#define HAL_RNG_MODULE_ENABLED */
70 #define HAL_RTC_MODULE_ENABLED
71 /*#define HAL_SAI_MODULE_ENABLED */
72 /*#define HAL_SMARTCARD_MODULE_ENABLED */

23
24 #include "stm32l4xx_ll_exti.h"
25

46
47 #define LB_CLIENT 0 /*1 = Device is GAP Central & GATT Client */
48 #define LB_SERVER 1 /*1 = Device is GAP Peripheral & GATT SERVER*/
49
```

Date: 6/2/2021

Finally, we should have a project structure like this that will allow for a clean compilation and flash process:



```
CDT Build Console [UCSD-Embedded-HW-Interfacing-WiFi-Nathan-Bunnell]
06:54:41 **** Incremental Build of configuration Debug for project UCSD-Embedded-HW-Interfacing-WiFi-Nathan-Bunnell ****
make -j4 all
arm-none-eabi-size UCSD-Embedded-HW-Interfacing-WiFi-Nathan-Bunnell.elf
  text  data  bss   dec    hex filename
 37844  152   6288  44284  acfc UCSD-Embedded-HW-Interfacing-WiFi-Nathan-Bunnell.elf
Finished building: default.size.stdout

06:54:42 Build Finished. 0 errors, 0 warnings. (took 1s.23ms)
```

All these steps complete and the device flashed, we should be able to see the access point with a Bluetooth scan, seen here from my phone as "LB_P2P_SERVER":

Date: 6/2/2021

Available devices



LYWSD03MMC



LB_P2P_SERVER

Closing Thoughts

We've studied several layers of abstraction in this course thus far, be it LL, HAL, BSP and even the middleware needed for the USB and WiFi projects. Having struggled through and completed this project, I would call this code obfuscated more than abstracted. I'm confident that I wouldn't have been able to implement this on my own without the assistance of the video and discussion within the group forum. I've also got to express a large amount of confusion as to why the example project is provided but its so far from complete that it is essentially useless as is. I would have imagined that, considering the product is sold as an IOT-centric Discovery board, it would have been a key focus of ST's development team to ensure the reference project are functional and easy to use out of the gate.

Postscript: After completing the first draft of this report, I realized I had forgotten to get the project structure in a screenshot. Going back to capture this, and having made no changes, I discovered a set of compiler warnings related to a series missing folders in my C:\ST\STM32CubeMX\Drivers directory. Poking around, I couldn't find any other references to these "missing" files and while it caused no other problems in compilation, the same issue propagated out into other completed projects. I'll chalk it up to one more idiosyncrasy of the IDE until I figure out the root cause but wanted to explain away the discrepancy between the console log and the project having warning icons in it.