

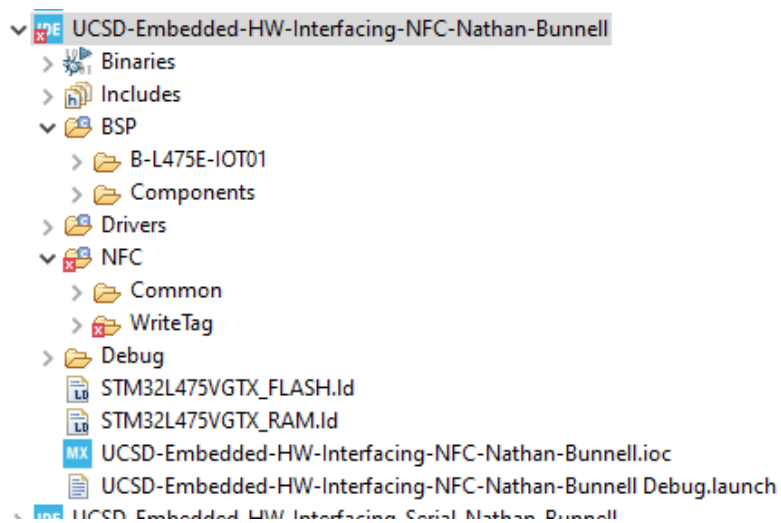
Date: 6/2/2021

## Assignment 9: NFC Hands ON

The following will document completion of the ninth assignment for ECE-40293, which will include the following goal:

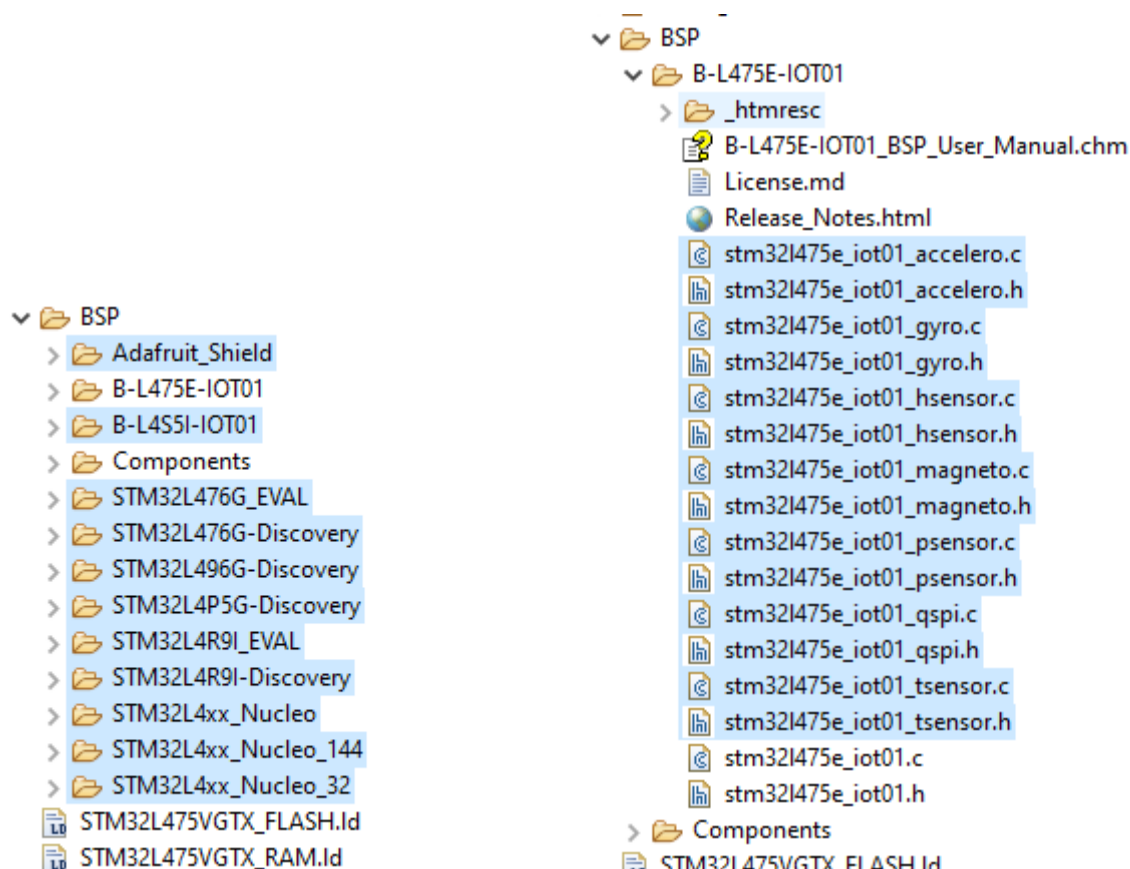
*You are to implement the WriteTag project that was covered in the Lesson Hands-On Section. The WriteTag project is included in the STM32Cube repository.*

Similar to the methods used in the WiFi and BLE assignments, we will generate our default project and then begin the process of pulling the needed support files, deleting unnecessary or duplicate files, and bit by bit, getting to a clean compiled and flashed program. Our final project structure should look like this (ignore the error, it was corrected by re-entering an include path later):

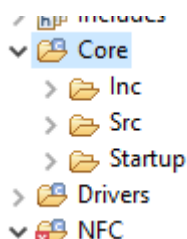


Date: 6/2/2021

To get here, we'll pull in the BSP folders as in previous assignments, making sure to pull the correct FW rev, 1.17 in my case, and then deleting all the extra files that are not required for this example:

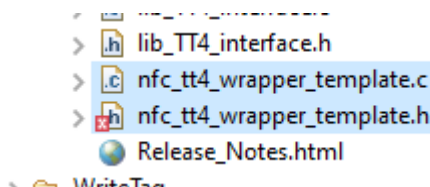


We can also delete the entirety of the Core directory as the NFC dir contains everything we will need:



Date: 6/2/2021

Finally, we will pull in the NFC example files from the reference location and can trim out the unused WrAARtoRunBLEapp and the EWARM & MDK-ARM sub-dirs under the WriteTag folder. The very last files to delete after this will be under NFC/Common/NDEF\_TagType4\_lib.



After updating the source and include paths in the project properties, we can compile and flash our Disco board.

Source folders on build path:

- > /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/BSP
- > /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/Drivers
- > /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/NFC

Include directories

- Drivers/CMSIS/Include
- Drivers/CMSIS/Device/ST/STM32L4xx/Include
- Drivers/STM32L4xx\_HAL\_Driver/Inc
- Drivers/STM32L4xx\_HAL\_Driver/Inc/Legacy
- /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/NFC/Common/M24SR
- /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/NFC/Common/NDEF\_TagType4\_lib
- /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/NFC/WriteTag/Inc
- /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/BSP/B-L475E-IOT01
- /UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell/BSP/Components/m24sr

CDT Build Console [UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell]

20:19:41 \*\*\*\* Incremental Build of configuration Debug for project UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell \*\*\*\*

make -j4 all

arm-none-eabi-size UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell.elf

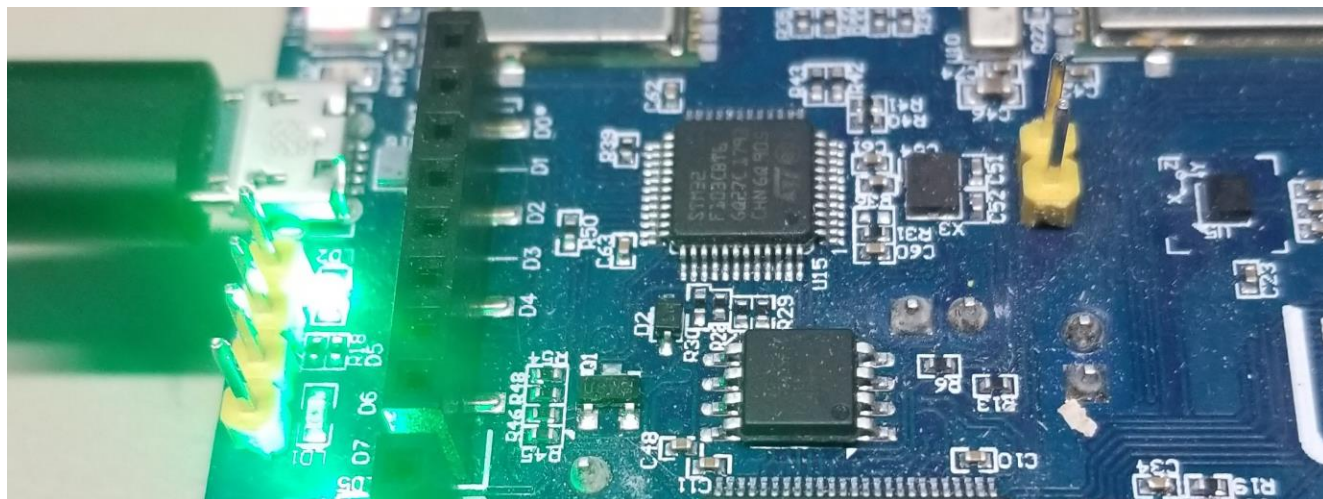
text	data	bss	dec	hex	filename
31132	668	4932	36732	8f7c	UCSD-Embedded-HW-Interfacing-NFC-Nathan-Bunnell.elf

Finished building: default.size.stdout

20:19:42 Build Finished. 0 errors, 0 warnings. (took 1s.33ms)

Date: 6/2/2021

When the program is running, we can see LED2 blinking, and the rate changing with presses of the blue user button.



Conveniently, one of the first results in the app store for an NFC reader was an ST solution specifically targeted to reading this hardware. Installed and enabled, I was able to place my phone on the board and read the device data off it.



Date: 6/2/2021

### **Closing Thoughts**

Compared to the BLE project, implementing NFC was much easier. I'm still confused on why these IOT projects aren't better developed "out of the box". Regardless, this was an interesting assignment. I cant think of any use cases for myself offhand but I would like to play around further with reading and writing the device data via the phone app or primary MCU.