Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

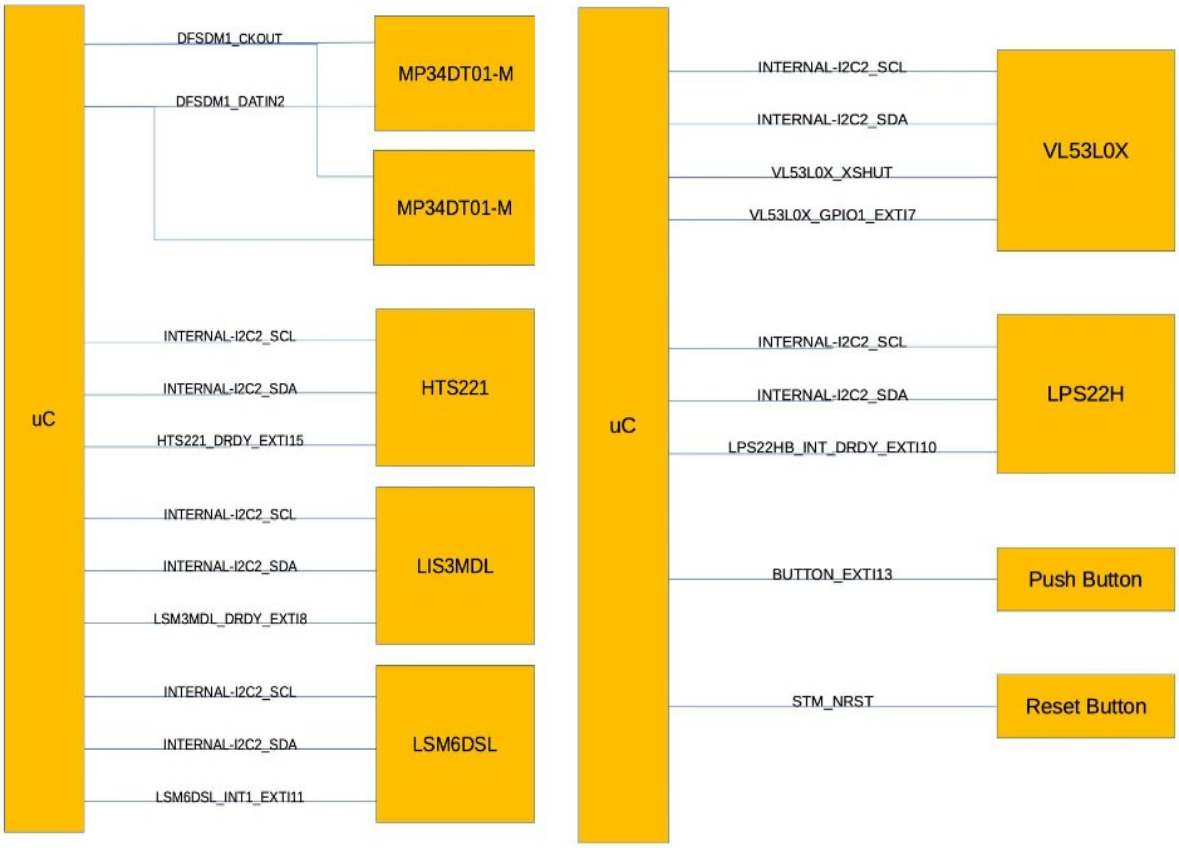
Project Report

- **Requirements Specification**
  - Schematic Design starts after consideration and approval of the MRD (Market Required Document), Functional Requirements and Engineering Specifications. These specifications and requirements can be categorized as the following points, which comes from lecture one's reading assignment:
    - Product requirements: Describe what the product is.
    - Functional Requirements: Describe what the product must do.
    - Engineering Specification: Describes how the design will be implemented and how the requirements will be met.
    - Hardware Specifications: Describe how specific hardware is designed.
    - Firmware Specifications: Describe how the firmware for specific processors will be designed.
    - Test Specifications: Describe what must be tested and how to verify that the system operates correctly.
  - Schematic Design includes assigning attributes (properties) to components (reference, value/name, footprint), performing electric rule checking.
  - The outcome of Schematic Design are generated BOM and Netlist – for PCB design.
  - PCB Design includes drawing PCB outline, selecting number of layers, placing components, layout (drawing traces, copper islands, ground/power layers), and performing DRC (Design Rule Check).
  - The outcome of PCB Design are generated Gerber, drill and assembly files.
  - BOM, Geber/drill/assembly files, Assembly Instructions, Test Procedures and Report template are sent to the manufacturer.
  - Manufacturer sends to the customer a ready-to-go product along with test results and a quality certification report.

Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

- Block Diagram



  - For sensors:
    - Noted that for sensors I2C bus, the SCL and SDA line are the same, total only two I2C bus goes to uC, the block diagram is just for reference only, since it is too hard to draw two lines and link to all the sensors.
    - IRQ output is a separate pin from every sensor chip.

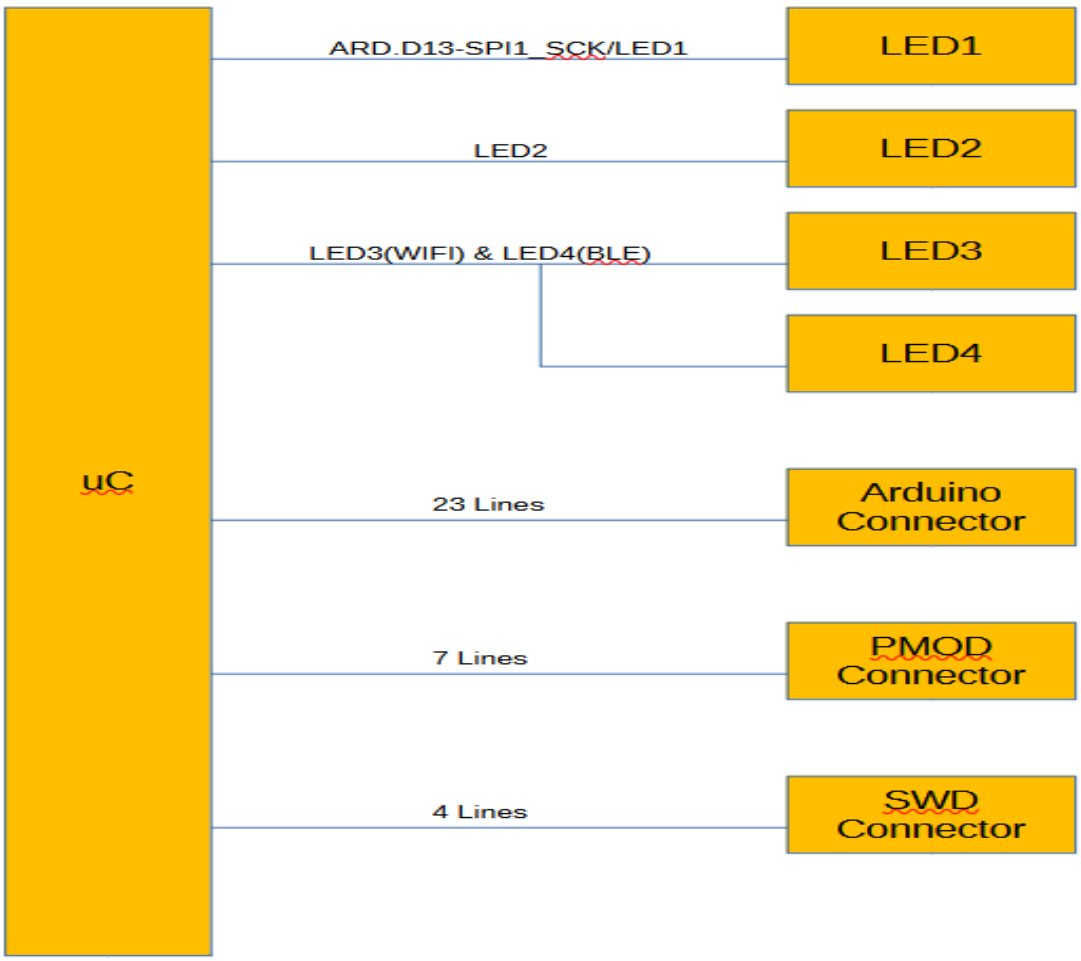
Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

- For RF modules:
  - All RF Modules are connected to the same SPI bus. From the bus only three lines go to uC. The block diagram is just for reference only, since it is too hard to draw same SPI bus lines and link to all the RF modules.
  - The signal /CS or CSN is a separate one for every RF Module.

Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

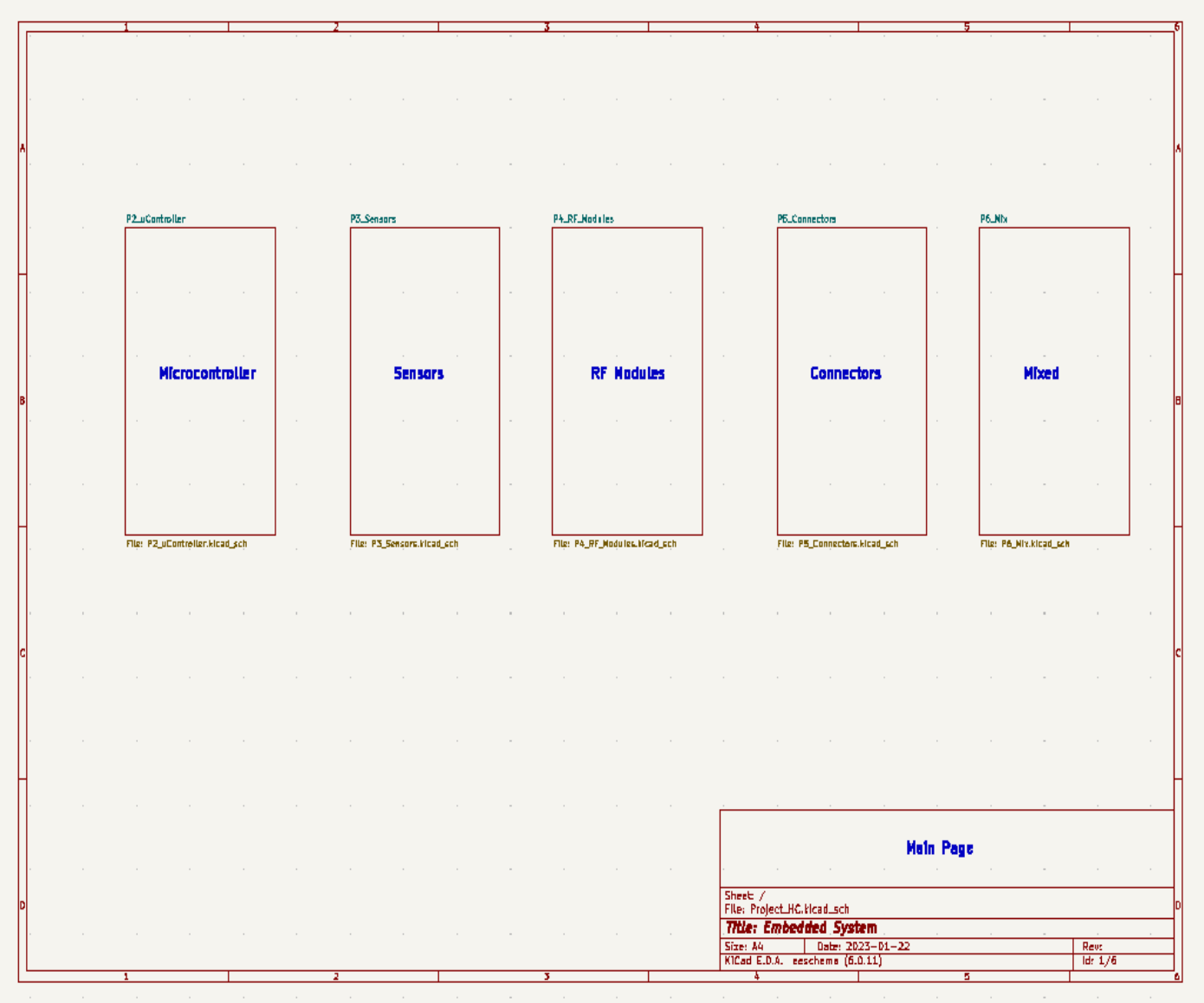


- For connectors and LED:
  - For LED's, total 3 lines are connected to micro-controller
  - For Arduino, total 23 lines connect to micro-controller that spread out in 4 connectors
  - For PMOS, total 7 lines connect to micro-controller
  - For SWD, total 4 lines connect to micro-controller

Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

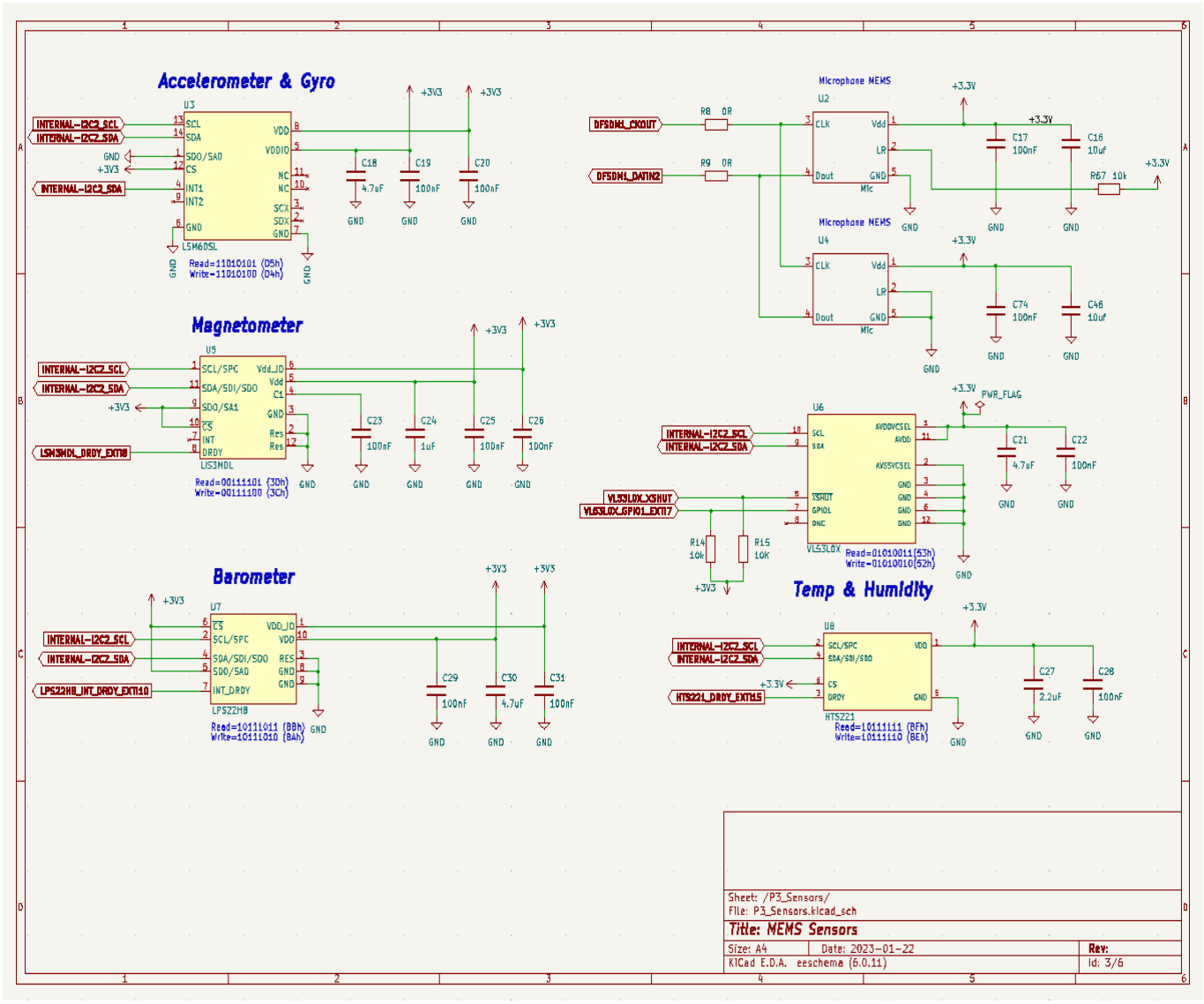
- **Hierarchical Schematic Creation (KiCAD)**



Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

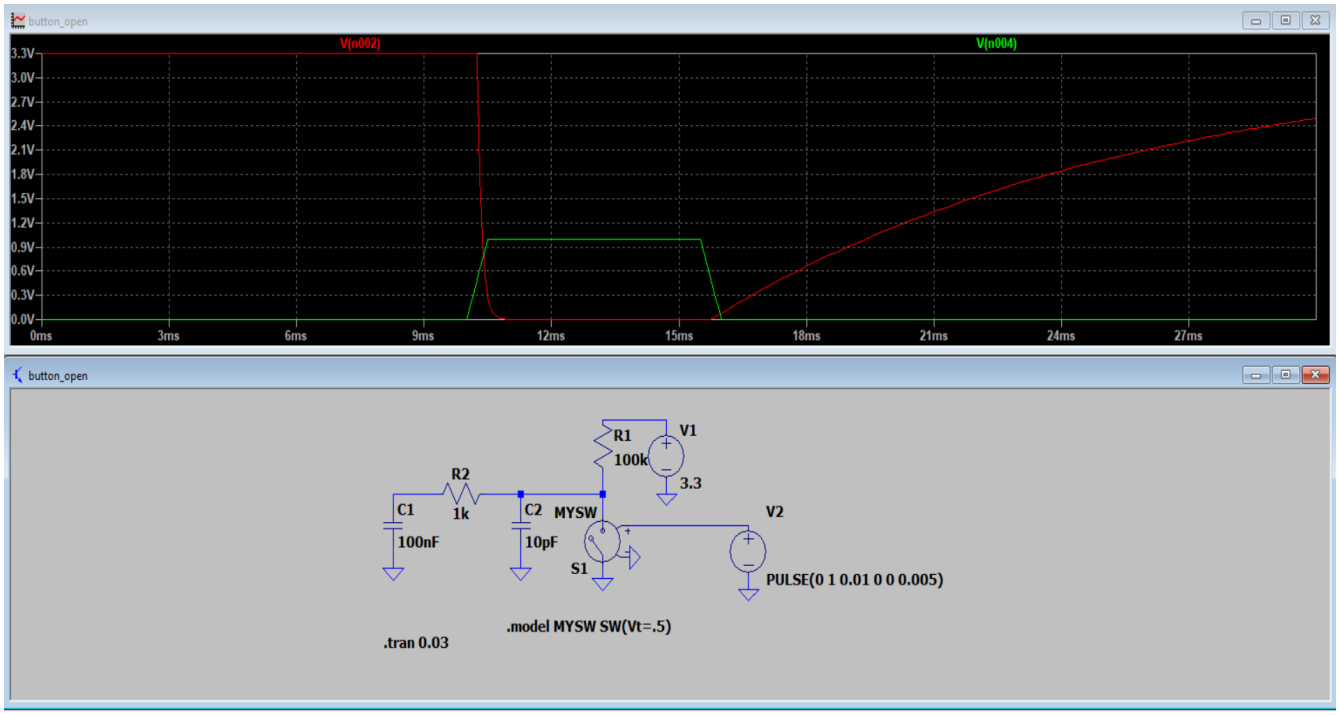
- **Example of Schematic Sheet (Page3)**



Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

- **Simulation of charging the cap 0.1uF on the input BUTTON_EXTI13 in LTSpice to calculate the interrupt delay (Fig.20 of STM32L475 DS)**



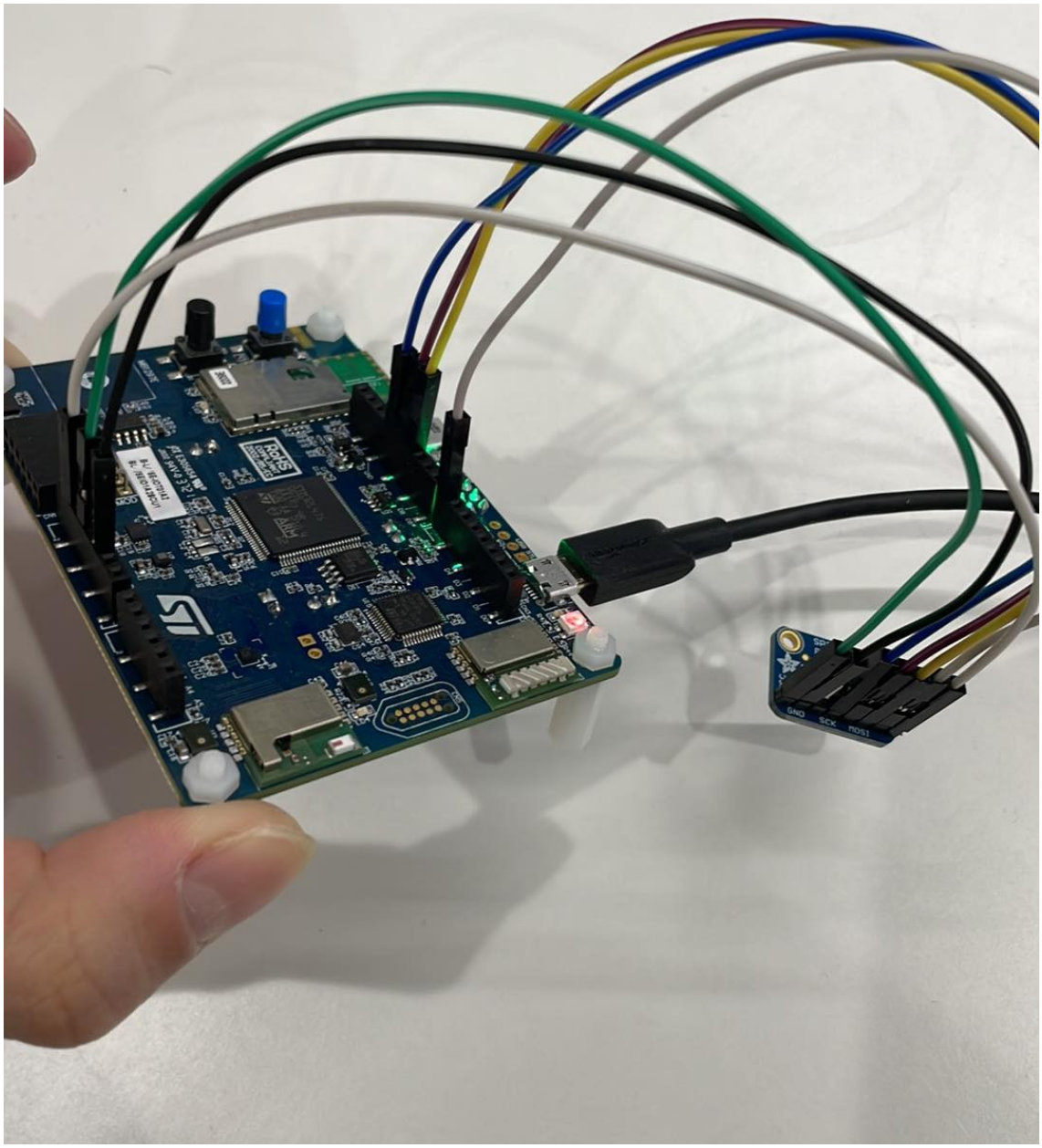
From the simulation in LTSpice, the red line is the voltage for cap 0.1uF, and green light is for the control for button press and open. From the STM32L475 data sheet, we use 3.3V supply voltage. V_IL = 0.39 * Vcc - 0.06 = 1.23 V, V_IH = 0.49 * Vcc + 0.26 = 1.88 V, so in order for the micro-controller to read back high, the voltage value has to be greater than 1.88V, it takes nearly 8ms for the cap to be greater than 1.88V.

- **ERC - Fixing Errors and Warnings. Generating BOM and Netlist. Why some errors may be ignored?**
  - Done, please see the attached project file
  - Error in current schematic:
    - Two outputs connected together: This is false negative statement, it relates to two MISO pin of RF modules connected
  - Warnings in current schematic:
    - Warnings are all about symbol has been modified in library, it requires seems I need to reorganize the symbol for better connection and placement

Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

- **External FRAM Module Connection**

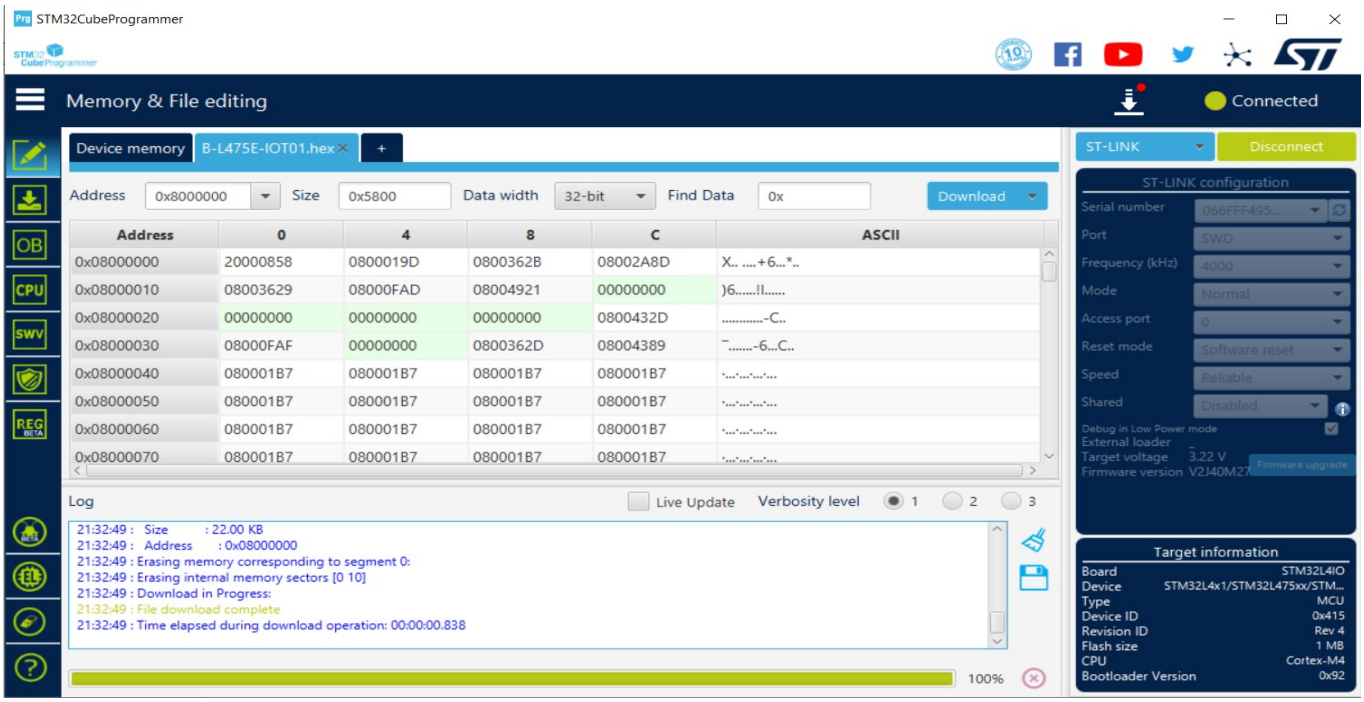


Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

IoT board	FRAM Module MB85RS64V
CN1 - PA5 - SPI1_SCK (D13 on the board)	SCK
CN1-PA6 - SPI1_MISO (D12 on the board)	MISO
CN1 - PA7 - SPI1_MOSI (D11 on the board)	MOSI
CN3 - PA4 - SPI1_NSS (D7 on the board)	CS
CN2 – 3.3V	VCC
CN2 - GND	GND
3.3V of the STM IoT board	NWP
unconnected	NHOLD

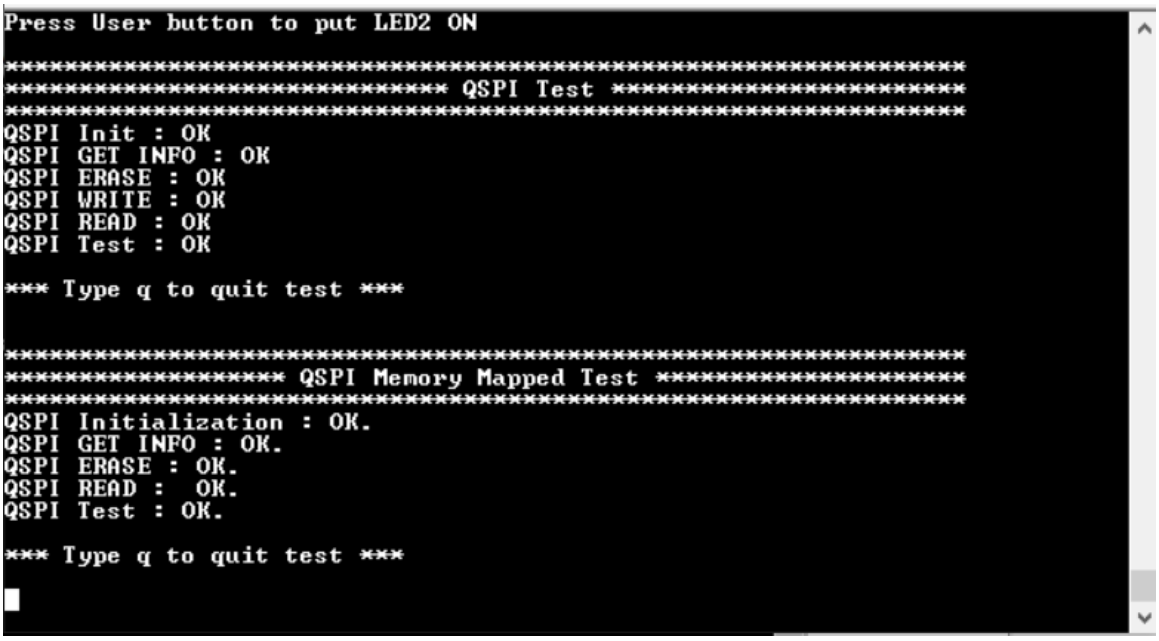
- **Programming uC for Testing the Sensors**



Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

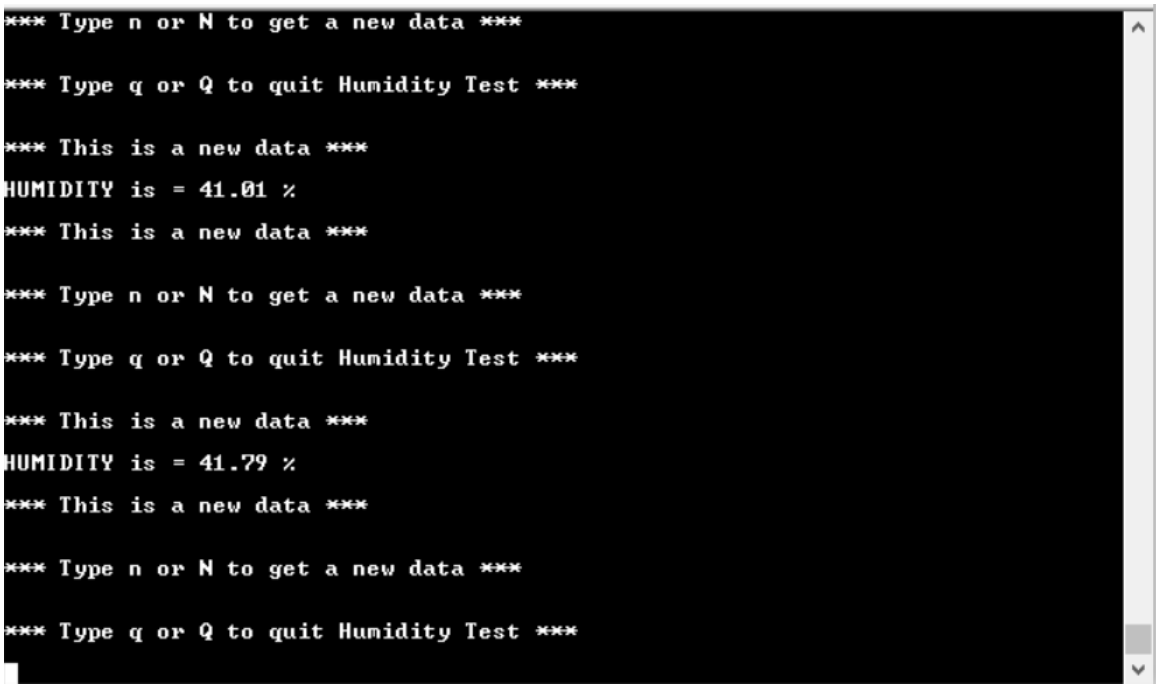
- **Results of all tests w. detailed comments**
  - Start with the QSPI testing



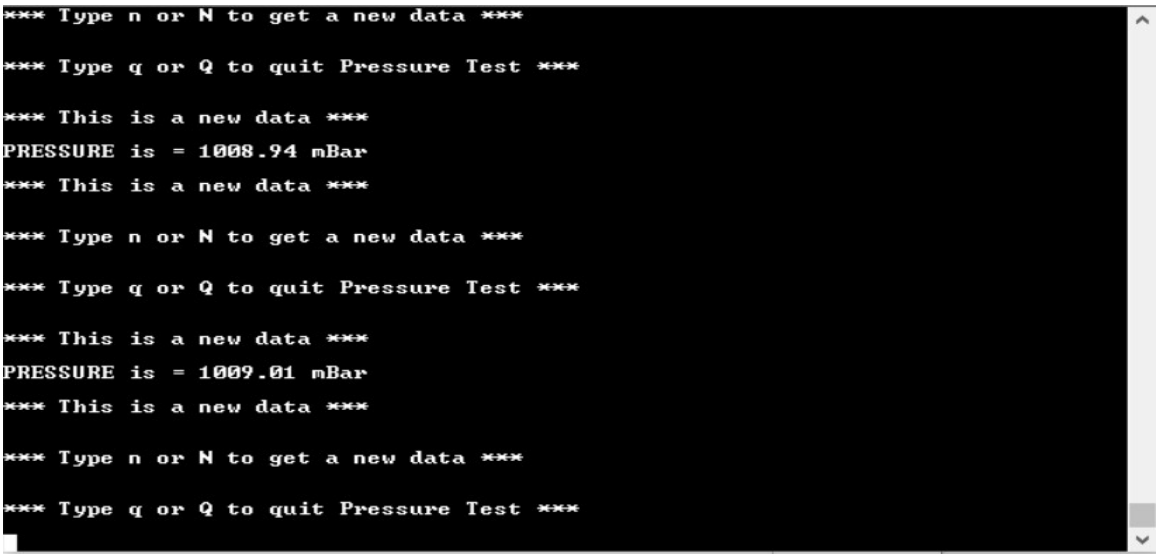
  - Test the temperature sensor – when I put my hand close to the temperature sensor, the temperature goes up
  - Test the humidity sensor – When I breath close to the humidity sensor, the humidity value goes up

Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023



- Test the atmospheric pressure sensor – High position average is 1008.68 mbar, low position average is 1008.97 mbar, (testing 3 points at high and low position and take the average)



Kai(Hsuankai) Chang
Name of the Course: 168584 Embedded Systems Hardware Design
Course Number: ECE-40292
Student ID: U09475562

Date: 03/09/2023

- **Conclusion:**

  After finishing this course, I have learned lots of new and interesting knowledge related to embedded system, especially in the hardware perspective. At the beginning of the course, I learned that in order to design a good embedded hardware system, different requirements and specifications have to be finalized and discussed. For instance, what type of function and market requirements does this embedded system need to fulfill, if going deeper, what type of processor, communication interface, and sensors need to be used in order to meet the requirements. It really give me a bigger picture when selecting embedded hardware device. This course also taught me how to use EDA tools like KiCad, which is very beneficial to my career. I learned how to draw schematic from scratch, choosing the right symbol, assigning value and footprint. I also learned how to use various website, like snapEDA, to get the symbol and footprint I need, or create it by myself. I can really feel the difference in my everyday work life, since right now I can read the schematic from our power electronics hardware team more easily, and I know where to find the connections and modules I need. Besides creating schematics, the weekly quiz is very helpful to refresh my electrical engineering knowledge. From capacitor, inductor, and resistor formula or circuit calculation, to basic embedded system brain storming questions, all benefits a lot in my learning journey. I also spent quite amount of time learning how to use LTSpice to simulate the circuit, which I even use it on my daily work now. Last but not least, the FRAM module project helps me refresh my knowledge in embedded programming, and I also reviewed how to use STM32CubeIDE to auto generate code and start testing the FRAM module. In sum, I think this course is definitely what I am looking for, as a firmware engineer dealing with programming everyday, sometimes hardware issue will get neglected, and most of the time the bug is not only in software, but hardware. After taking course, I think I have equipped with more tools and knowledge to continue my embedded career, and I am ready to my next adventure.