

Wabash Capstone Project Notes 09

By: John Danison

What Did I Say I Would Do This Week?



Find electrical schematic software & get working locally.



Begin writing firmware outline & find proper libraries.

What I Have Done Over The Past Several Weeks



Wrote test firmware modules for each sensor we are using. Created a simple working prototype. See slide 4.



Found electrical schematic software in the forms of Fritzing and Altium Designer. Both options are installed and working. See slide 5.



Created more electrical documentation for new circuits. Also did calculations for battery selection and circuit power. See slide 7.



Max found strain gauges, so I created the circuitry and code needed to test these for the first time. See slide 11.

Breakdown of Code Modules

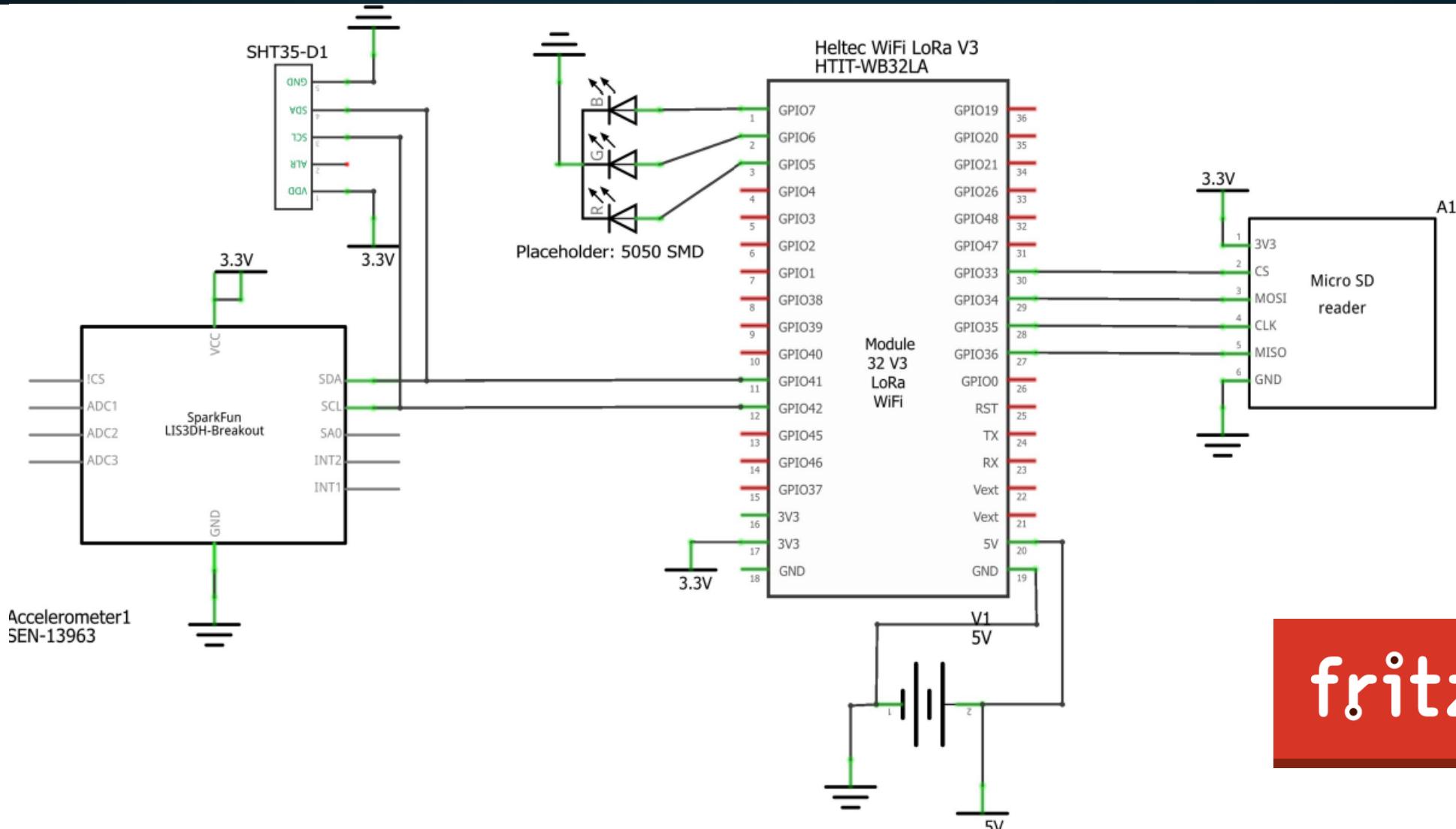
The screenshot shows the VS Code interface with the following details:

- EXPLORER** pane on the left:
 - RECEI... (with a plus icon)
 - .pio
 - .vscode
 - data
 - include (highlighted in grey)
 - lib
 - src
 - LIS3DH_Module.cpp
 - LIS3DH_Module.h
 - main.cpp
 - main.h
 - NAU7802_Module.cpp
 - NAU7802_Module.h
 - OLEDDisplay_Modul...
 - OLEDDisplay_Modul...
 - SDCard_Module.cpp
 - SDCard_Module.h
 - SHT45_Module.cpp
 - SHT45_Module.h
 - test
 - .gitignore
 - platformio.ini
 - Temp.txt
- Editor** pane on the right:
 - platformio.ini X
 - NAU7802_Module.h

Content of platformio.ini:

```
1 ; PlatformIO Project Configuration
2 ;
3 ; Build options: build flags, ...
4 ; Upload options: custom upload...
5 ; Library options: dependencies...
6 ; Advanced options: extra script...
7 ;
8 ; Please visit documentation for...
9 ; https://docs.platformio.org/pa...
10
11 [env:heltec_wifi_lora_32_V3]
12 platform = espressif32
13 board = heltec_wifi_lora_32_V3
14 framework = arduino
15 monitor_speed = 115200
16 build_flags =
17     -D CONFIG_FATFS_LFN_HEAP
18     -D CONFIG_FATFS_EXFAT_ENABLED
19 lib_deps =
20     heltecautomation/Heltec_ESP32
21     jgromes/RadioLib@^6.4.2
```
- Terminal** pane at the bottom (not visible in the screenshot).

Electrical Software 1 – Fritzing



fritzing

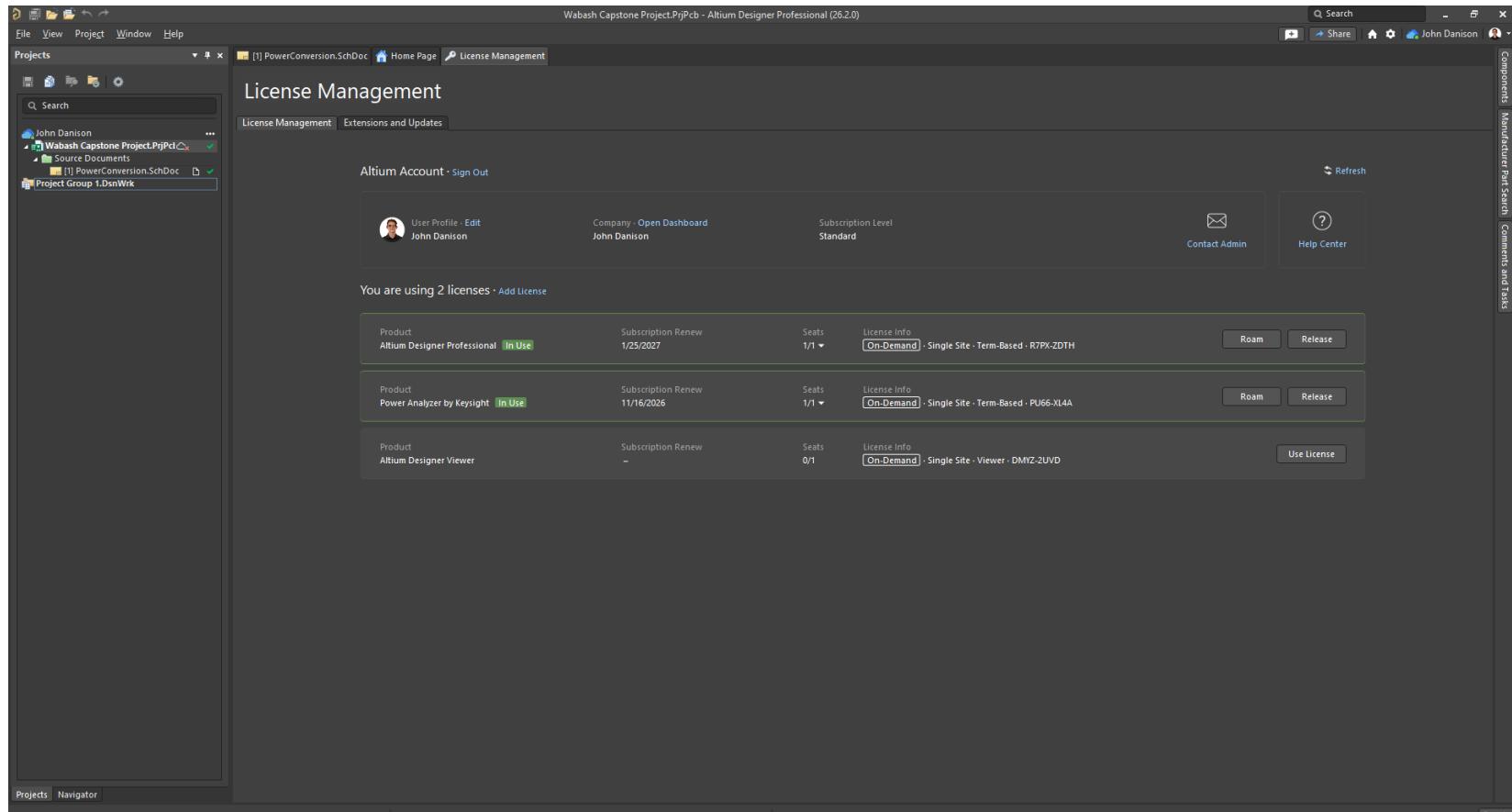
Electrical Software 2 – Altium Designer



Going forward, Altium Designer will be used.

Fritzing is good for quick simple drawings, Altium is PHENOMINAL at schematic making, PCB layouts, and more.

Nothing has been done in Altium so far, I just got it working today.



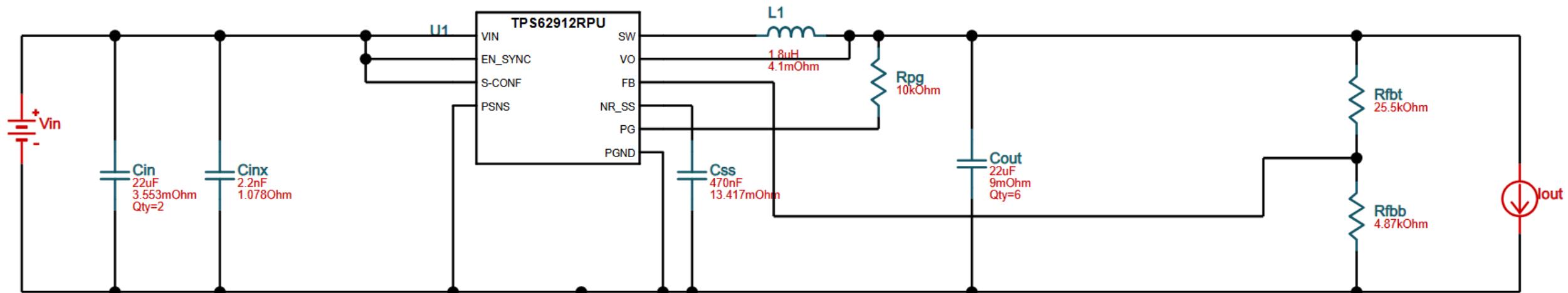
Summary of Current Breakdown & Battery Calculations

Current Breakdown	
Component	Current (mA)
ESP32	20
LoRa Radio (idle)	46
LIS3DH	0.011
SHT35	0.1
microSD	2
Total	68.211
+10% Overhead	75.0321

Battery Calculations		
Calculation Step	Formula	Result
mAh @ 5V	$75.0321 \text{ mA} \times 6 \text{ months} \times 30 \text{ days} \times 24 \text{ hr}$	324.139 Ah
Wh @ 5V	$324.139 \text{ Ah} \times 5 \text{ V}$	1620.699 Wh
Wh w/ battery loss	$1620.699 \text{ Wh} \div 0.90$	1800.770 Wh
Ah from 12V battery	$1800.770 \text{ Wh} \div 12.8 \text{ V}$	140.685 Ah

Sample Designed Power Converter

Sample 12V : 5V DC Buck Converter



Efficiency: 96.6%

BOM Cost: \$3.07

Footprint: 169 mm²

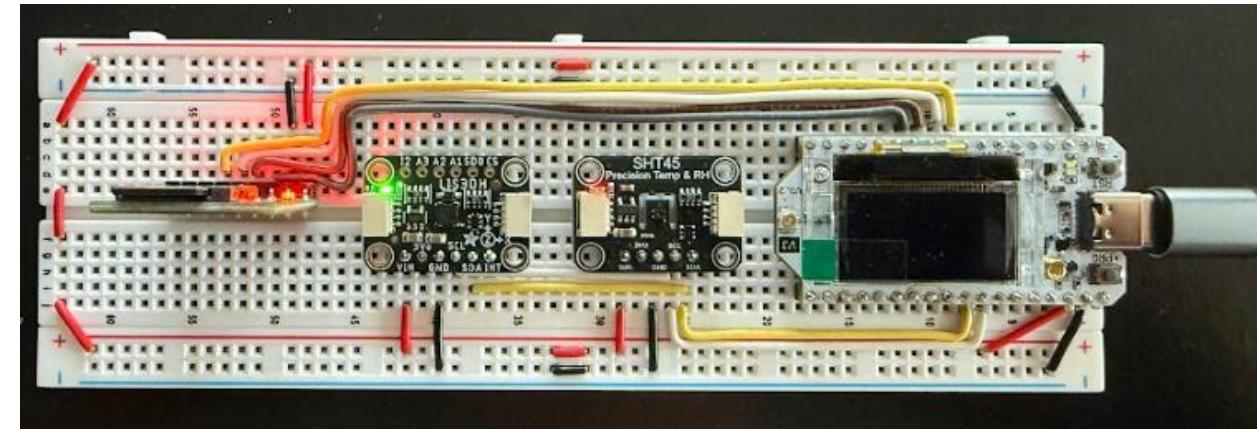
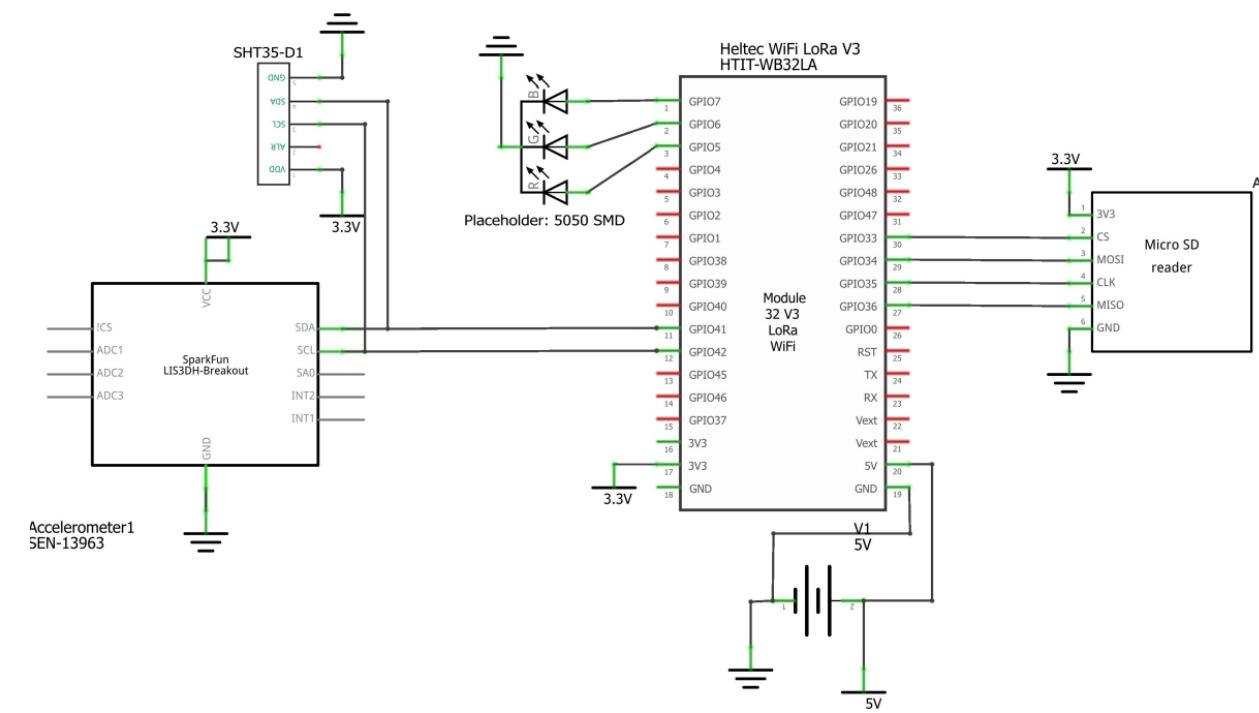
BOM Count: 15

Topology: Buck

Frequency: 2.2 MHz IC Cost: \$0.84 | 1ku



Comparison of Circuit Software & Hardware Layouts



Comparison of Software Data Acquisition

Data Sample Expectations

Parameter	Value / Calculation	Result
Sensors	3-axis accel + 1 strain (high-rate) + Temp & RH (1 sample)	—
Sampling Rate (Accel/Strain)	2,000 samples/sec	—
Data Format	16-bit (2 bytes/sample)	—
Data Rate	$2,000 \times 4 \times 2$	16 kB/s
Burst Duration	—	2 s
Burst Size (Accel + Strain)	$16 \text{ kB/s} \times 2 \text{ s}$	≈ 32 kB
Temp + RH Samples	2 sensors × 2 bytes each	+ 4 B
Total per Burst	$32 \text{ kB} + 4 \text{ B}$	≈ 32 kB
128 GB microSD Capacity	≈ 137,000 MB	—
Bursts Storable	$137,000 \text{ MB} \div 0.031 \text{ MB}$	≈ 4.3 million bursts

Unoptimized Data Sample Results

```

1 === EVENT 3 ===
2 Timestamp: Time not set
3 Temperature: 24.64 C
4 Humidity: 23.93 %
5
6 Accelerometer Samples (20):
7 Sample, X(g), Y(g), Z(g)
8 1, 1.864, -2.034, 1.334
9 2, 1.952, -0.415, 1.257
10 3, 1.996, 0.813, 1.649
11 4, 2.018, 1.427, 1.845
12 5, 2.029, 1.734, 1.898
13 6, 2.035, 1.887, 0.912
14 7, 2.038, 1.964, 0.656
15 8, 2.039, 2.002, 0.229
16 9, 2.040, 2.021, 0.477
17 10, 2.040, 2.031, 1.259
18 11, 2.040, 0.088, 1.650
19 12, 2.040, 0.088, 1.650
20 13, 2.033, -0.976, 1.845
21 14, 1.223, -1.508, 1.943
22 15, -0.070, -1.774, 1.667
23 16, -0.733, -1.907, 0.834
24 17, -0.213, -1.973, -0.100
25 18, 0.914, -2.006, -1.070
26 19, 1.477, -1.620, -1.555
27 20, 1.759, 0.210, -1.797
28

```

Normal text length: 622 lines: 29 Ln:1 Col:1 Pos:1 Unix (LF) UTF-8 INS

New Strain Gauge Circuitry & Demo

The screenshot shows a Windows desktop with a MSI logo. On the screen is the PlatformIO Development Environment. The left sidebar has sections like 'Upload and Monitor', 'Clean', 'Devices', 'Platform', 'Program Size', 'Upload Filesystem', 'Erase Flash', 'Dependencies', 'Advanced', 'Remote', and 'Miscellaneous'. Under 'PIO Home', it shows 'Open', 'PIO Account', 'Inspect', 'Projects & Config...', 'Libraries', 'Boards', 'Platforms', and 'Devices'. Under 'Debug', it shows 'Start Debugging' and 'Toggle Debug Con...'. Under 'Miscellaneous', it lists 'Serial & UDP Plotter', 'PlatformIO Core CLI', 'Clone Git Project', 'New Terminal', 'Upgrade Platform...', and 'Show Release Notes'. The main area shows C++ code for strain gauge monitoring, terminal output for task execution, and a table of strain measurement data.

```
int32_t raw = nau7802.readRaw();
Serial.printf(" Single: %ld\n", raw);

Serial.println("\nFiltered readings (16 samples each):");
int32_t avg = nau7802.readAverage(16);
int32_t median = nau7802.readMedian(9);
int32_t filtered = nau7802.readFiltered(16);

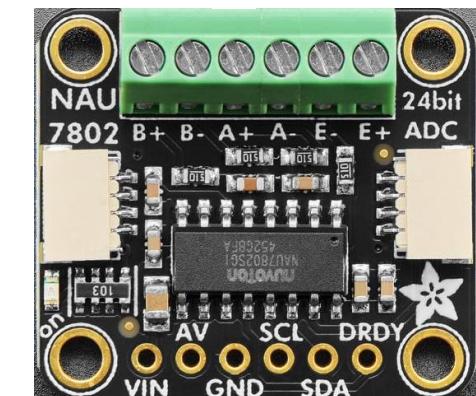
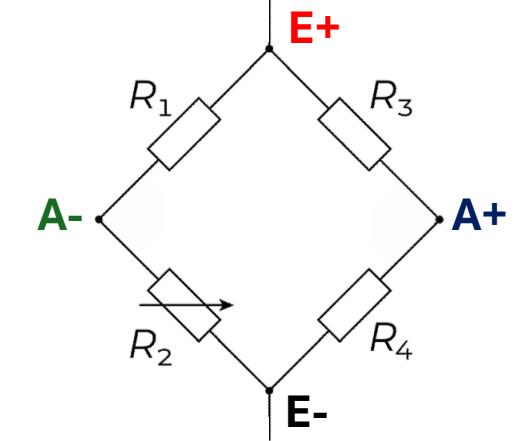
Serial.printf(" Average: %ld\n", avg);
Serial.printf(" Median: %ld\n", median);
Serial.printf(" Filtered: %ld (outliers removed)\n", filtered);

// Show zeroed readings
int32_t reading = nau7802.getReading(); // Uses single raw read
float offset = nau7802.calculateOffset(filtered);

==== TARING STRAIN GAUGE ===
NAU7802: Taring with 20 samples...
NAU7802: Zero offset set to 6526588
Strain gauge zeroed successfully!
=====

==== CONTINUOUS STRAIN MONITORING ===
Monitoring strain in real-time...
Apply load to the strain gauge now!
Press any key to stop.

Time(s), Raw, Filtered, Zeroed, Strain(με)
1.08, 6525534, 6528253, 7446, 55.48
2.38, 6525296, 6525955, 4628, 34.42
3.68, 6525676, 6527842, -271, -2.02
4.99, 6531346, 6527368, -6077, -45.28
6.29, 6531612, 6528445, 1972, 14.69
7.59, 6526986, 6526873, -2078, -15.42
8.89, 6528760, 6526944, 9787, 72.32
10.19, 6521834, 6528758, 8257, 61.52
11.49, 6528966, 6525065, 7208, 53.64
12.79, 6520184, 6528148, 11771, 87.70
```



Did I Do What I Say I Would Do?

I believe that I covered everything that I said I would do. I created the firmware and circuitry required to get all sensors working.

I showed up to every lecture, lab, and meeting and contributed my knowledge with everyone.

Plans For Upcoming Weeks

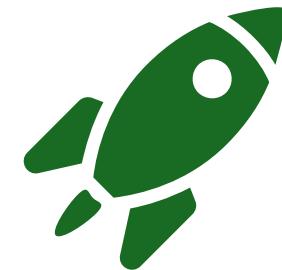


High Level

Start to refine system sensors & configuration.

Learn how to use Altium Designer. Will follow tutorial videos.

Continue to show up to class.



By EOW:

Continue to refine the noise readings in strain gauge circuit. Figure out a reliable solution.

Begin creating schematics in Altium Designer. Need to start PCB design ASAP.