Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Team 34: Portable High Energy Experiment (PHEE) DAQ

ECEN 404 Final Presentation

Team Members: Ethan Barnes
John Sabra
Sang Hoon Chung

Sponsor: Michelle Chatter
Sandia National Laboratories

TA: Logan Smith

Project Summary

Sandia National Laboratories

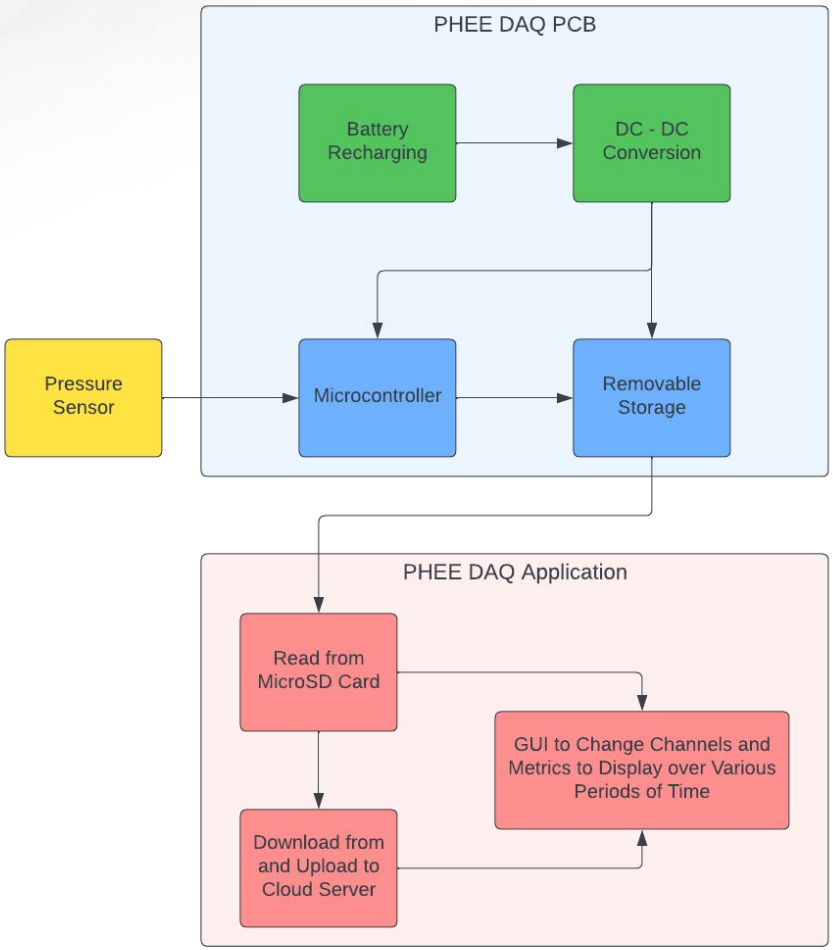
Problem statement:

- The United States military possesses about 5500 nuclear weapons in its stockpile
- The security of these weapons and equipment is paramount when they are transported for storage and testing
- Sandia is interested in detecting explosive-type events in sensitive areas
  - Exact application for DAQ system may not be disclosed

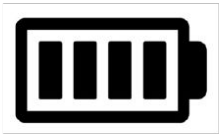
The Portable High Energy DAQ System will:

- Protect government equipment by detecting and identifying explosives within a 100 ft range
  - Utilizes a pressure sensor to classify if an explosive event occurred
- Write output to removable storage device
  - User will be able to refer to and perform analysis on past data

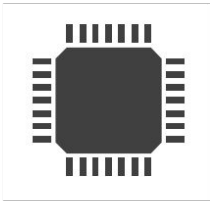
Integrated Project Diagram



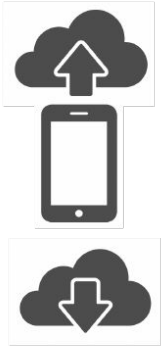
Johnny Sabra



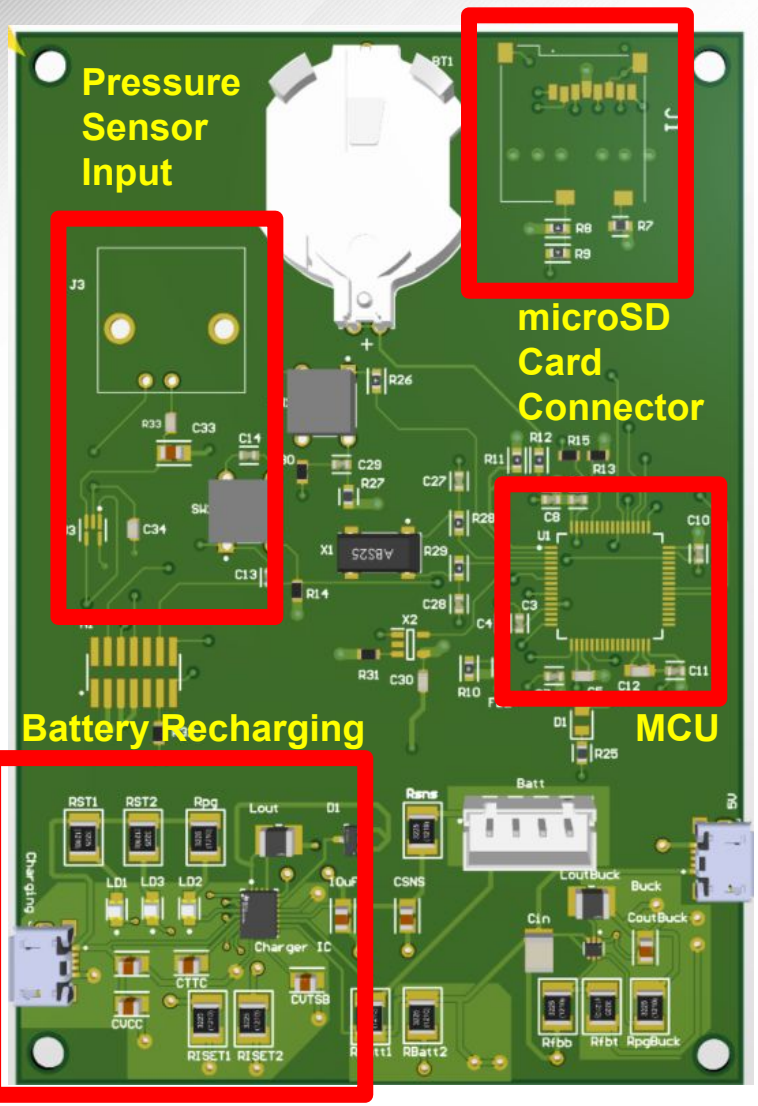
Ethan Barnes

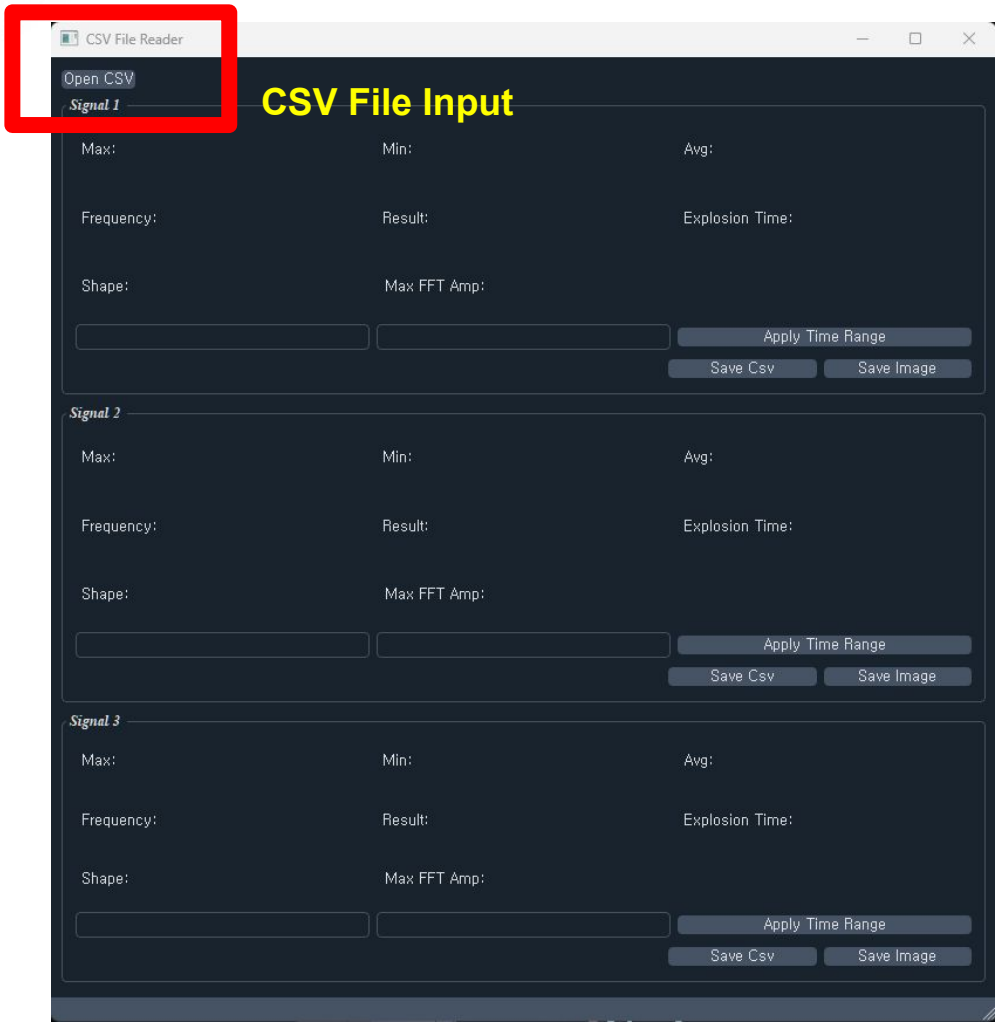


Sang Hoon Chung

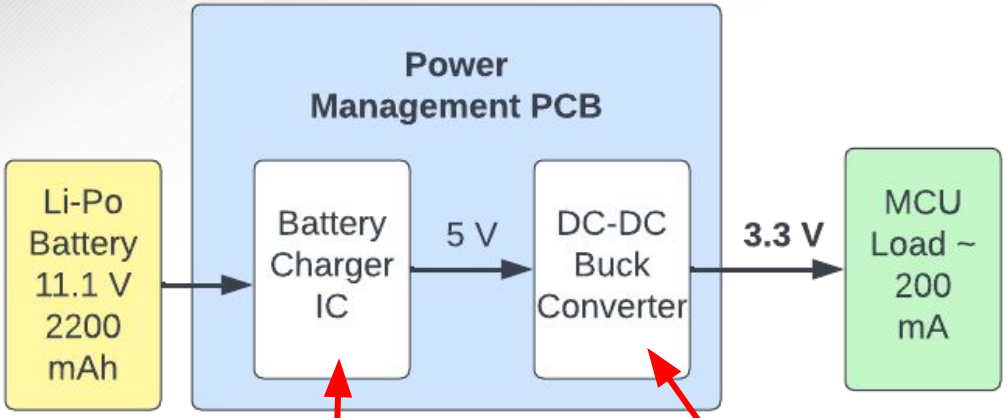


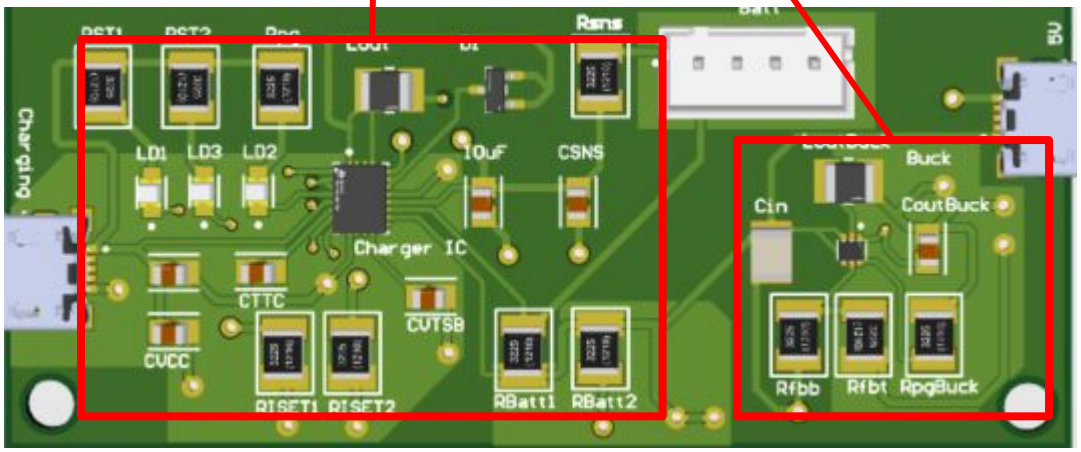
Integrated Project Diagram





DC Power Supply



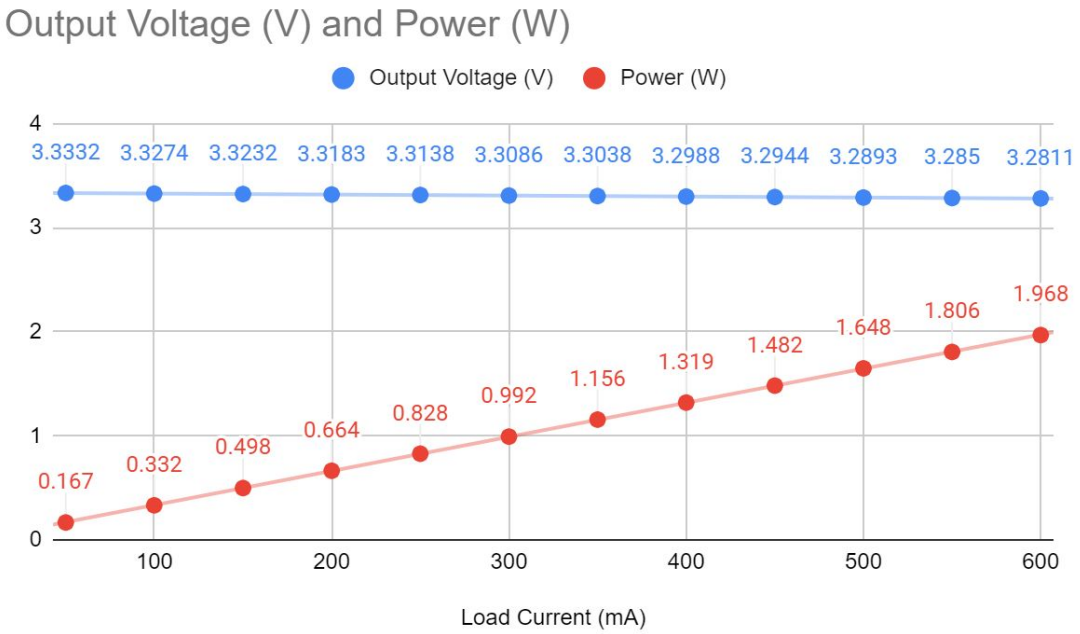


- Battery Charger IC
  - Takes 11.1V input from LiPo Battery and regulates to 5V
  - Utilizes DC Input (5 - 16V) to charge battery
- DC-DC Buck Converter
  - Utilizes 5V input to generate required supply voltage of 3.3V for the MCU

Design requirements:

Supply Voltage	Load Current
3.3 V	~ 200 mA

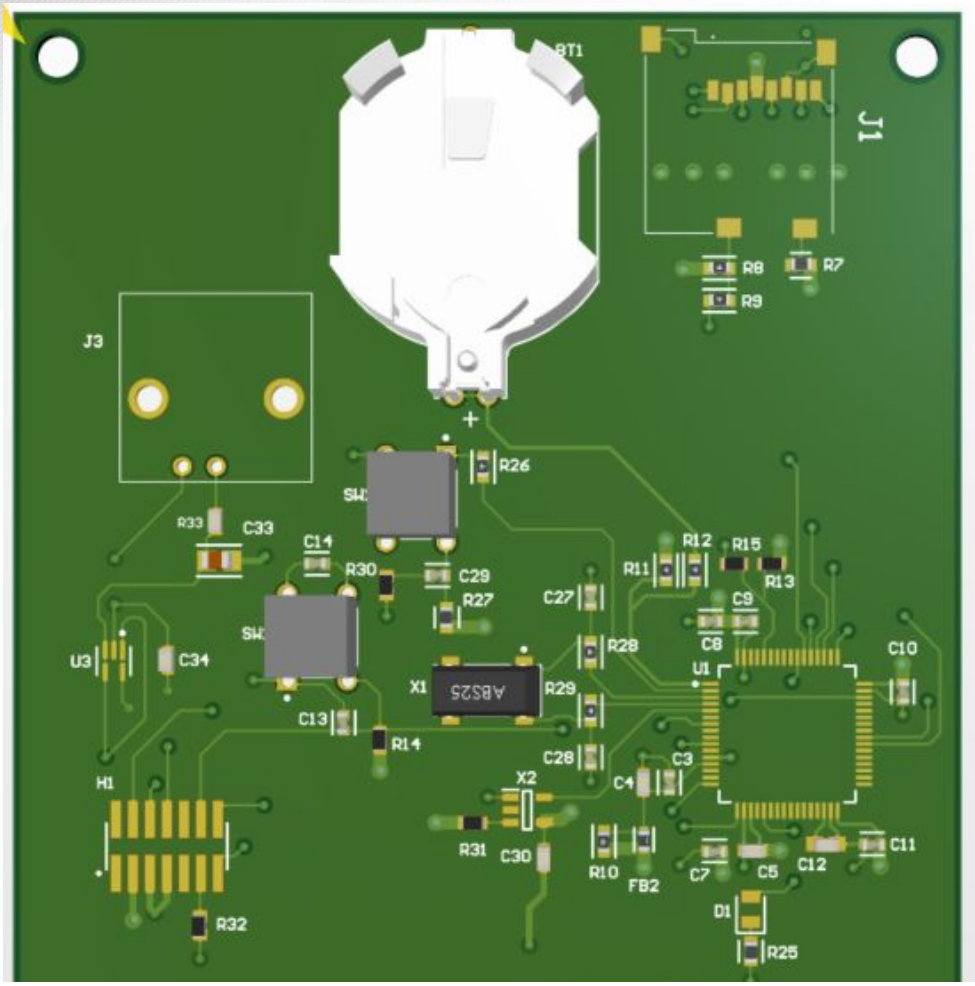
Validation



- DC-DC buck converter output validated at 3.3138 V under normal load.
- **Challenge**: Battery charger output tested at 2.8 V(outside buck converter input range)
  - solution to switch from 3-cell battery to 1-cell battery pending

Average Output Voltage	**3.3064 V**
Output Voltage Under Load	**3.3183 V**
Average Power	**1.071666667 W**

Microcontroller PCB



Device	Measured Voltage	Expected Voltage
Micro USB Output	5.001 V	5 V
LDO Regulator Output	3.318 V	3.3 V
MCU VDD Input	3.3179 V	3.3 V
microSD I/O Input	3.3027 V	3.3 V

- PCB is powered with 5V USB output from laptop or 5V output from Power Supply Subsystem
- MCU reads data from pressure sensor

Microcontroller PCB

```
Log output file:   C:\Users\Ethan\AppData\Local\Temp\STM32CubeProgrammer_a17700.log
ST-LINK SN  : 004D004933315112343338 34
ST-LINK FW  : V3J10M3B5S1
Board       : STLINK-V3SET
Voltage     : 3.30V
SWD freq    : 24000 KHz
Connect mode: Under Reset
Reset mode  : Hardware reset
Device ID   : 0x421
Revision ID : Rev A
Device name : STM32F446xx
Flash size  : 512 KBytes
Device type : MCU
Device CPU  : Cortex-M4
BL Version  : 0x90


Memory Programming ...
Opening and parsing file: ST-LINK_GDB_server_a17700.srec
  File          : ST-LINK_GDB_server_a17700.srec
  Size          : 57.45 KB
  Address       : 0x08000000


Erasing memory corresponding to segment 0:
Erasing internal memory sectors [0 3]
Download in Progress:


File download complete
Time elapsed during download operation: 00:00:01.311
```

- Connection to debugger and MCU on PCB successful
- Code successfully flashed onto MCU

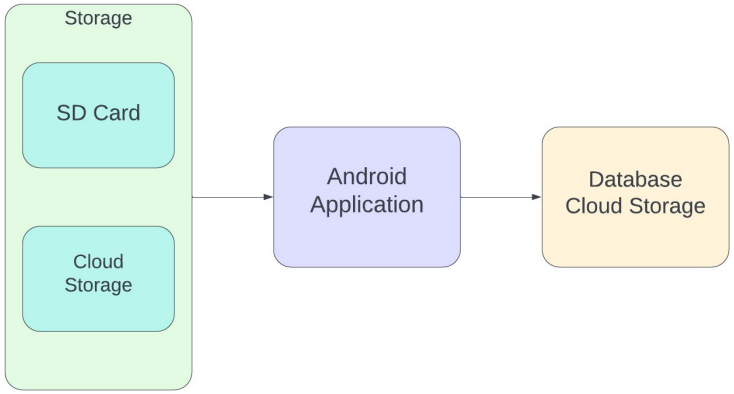
```
Port 0 ×
SD card mounted successfully...
*adc_data.csv* created successfully
*adc_data.csv* created successfully
*adc_data.csv* created successfully
*adc_data.csv* created successfully
Total time to write 30000 samples to SD card (WITH printf, not in ms yet): 10901 ms
Samples per second: 2752.041
```

- Expected sample rate (>2 ksps)

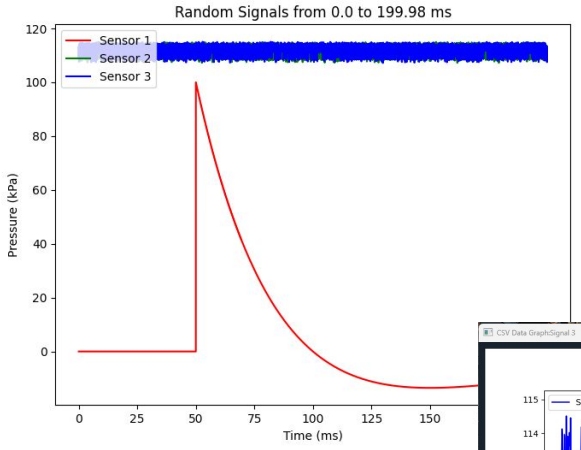
Application

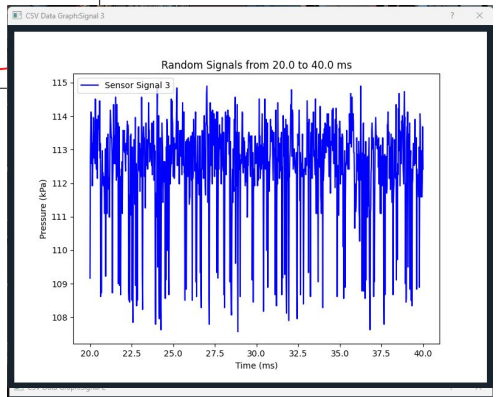
Challenge 1: Develop a GUI to configure DAQ settings and output “quick looks” of data

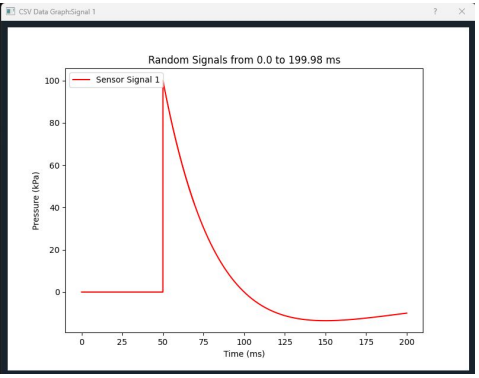
Challenge 2: Accessing and managing signal data from various sources.

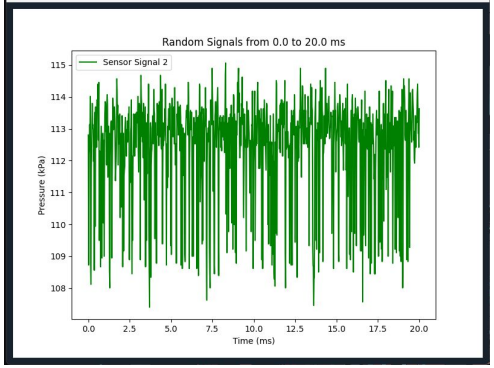


Application









Application

```python
def open_csv(self):
    csv_filename, _ = QFileDialog.getOpenFileName(None, "Open Image File", "", "*.csv")
    if csv_filename == "":
        QtWidgets.QMessageBox.warning(self, "Warning", "Please load data first", QtWidgets.QMessageBox.Cancel)
        return
    # Check if the extension is .csv
    if not os.path.splitext(csv_filename.lower())[1] == ".csv":
        QtWidgets.QMessageBox.warning(self, "Warning", "Please load CSV format file", QtWidgets.QMessageBox.Cancel)
        return
    # Try to read the CSV file
    try:
        df = pd.read_csv(csv_filename)
    except pd.errors.EmptyDataError:
        QtWidgets.QMessageBox.warning(self, "Warning", f"document {csv_filename} is an empty file", QtWidgets.QMessageBox.Cancel)
        return False
    except FileNotFoundError:
        QtWidgets.QMessageBox.warning(self, "Warning", f"document {csv_filename} does not exist", QtWidgets.QMessageBox.Cancel)
        return False

    # Check if required columns are included
    required_columns = ["time", "amplitude 1", "amplitude 2", "amplitude 3"]
    missing_columns = [col for col in required_columns if col not in df.columns]

    if missing_columns:
        QtWidgets.QMessageBox.warning(self, "Warning", f"document {csv_filename} The following columns are missing: {', '.join(missing_columns)}",
                                      QtWidgets.QMessageBox.Cancel)
        return False
    self.data = df
```

Figure 1

```python
def list_files(self):
    service = self.connect_to_drive()
    results = service.files().list(pageSize=10, fields="nextPageToken, files(id, name)").execute()
    items = results.get('files', [])
    for item in items:
        print(u'{0} ({1})'.format(item['name'], item['id']))
```

Figure 2

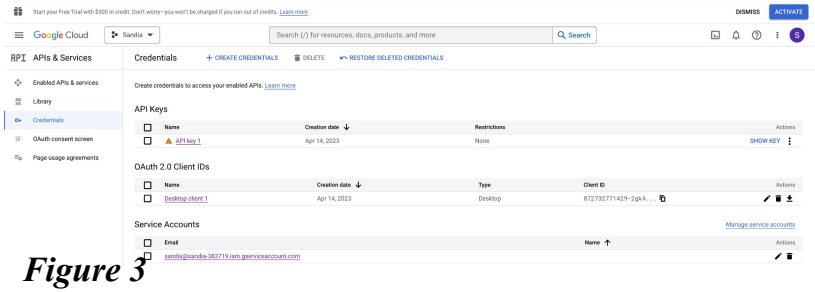


Figure 3

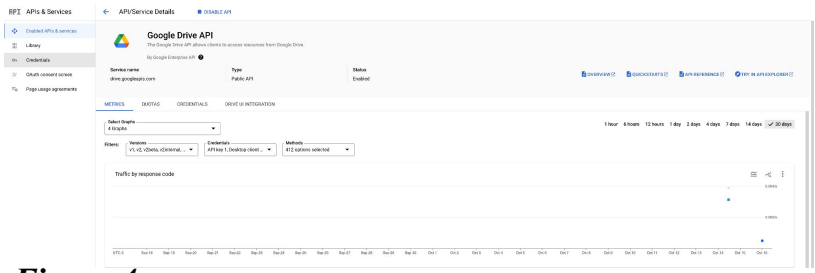


Figure 4

figure 1: The script to access to file explorer(.csv file)
figure 2: The script to list the Google Drive file(.csv).
figure 3 and figure 4: These figures show the API is working while I run the script for the application.

Integrated System Results

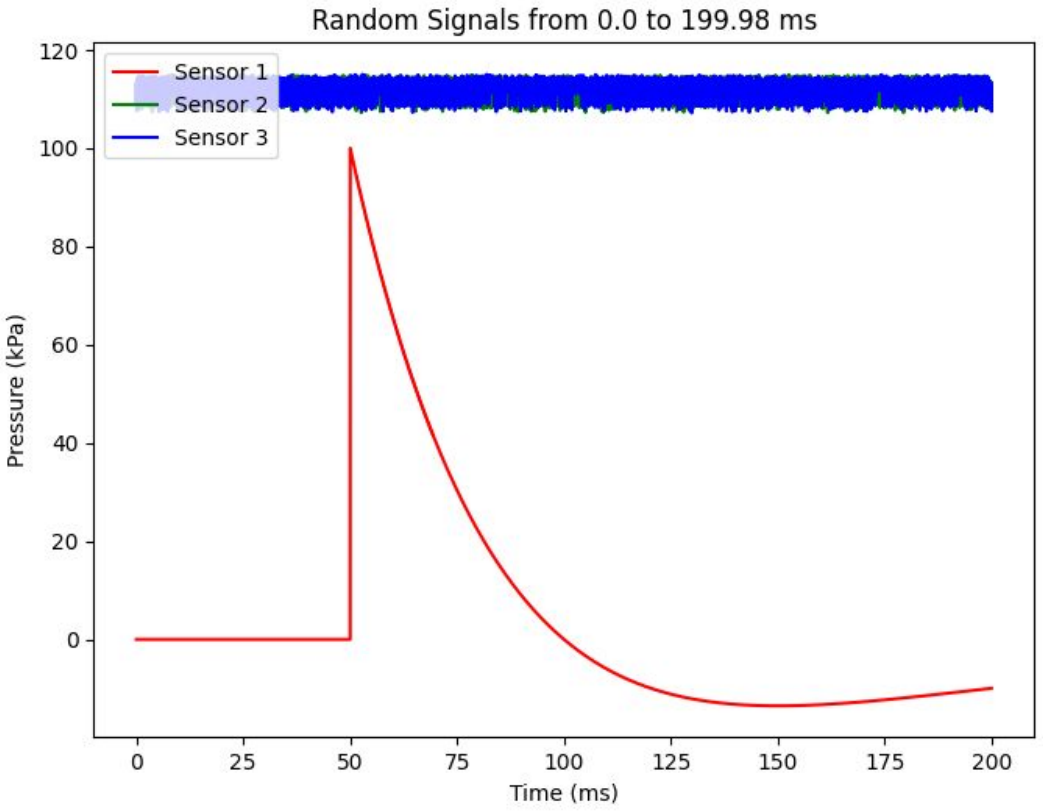
Metric	Actual Results	Expected Results
Sampling Rate	~2752 sps	> 2000 sps
MCU Supply Voltage	3.3179 V	3.3 V
Pressure Sensor Reading	Constant ~1.65 V (Constant 112.8 kPa)	0 V - 2.193 V (0 kPa - 150 kPa)
Power Supply Voltage	3.3138 V Output Voltage under load	3.3 V @ 120 mA

Integrated System Results

BATCH_0.CSV

File Edit View

```
batch,cluster,explosion,pressure,delta_pressure
0,0,0,2048.000,0.000
0,0,0,2012.000,0.000
0,0,0,2013.000,1.000
0,0,0,2031.000,18.000
0,0,0,2042.000,11.000
0,0,0,2054.000,12.000
0,0,0,2056.000,2.000
0,0,0,2052.000,0.000
0,0,0,2045.000,0.000
0,0,0,2043.000,0.000
0,0,0,2043.000,0.000
0,0,0,2036.000,0.000
0,0,0,2049.000,13.000
0,0,0,2051.000,2.000
0,0,0,2061.000,10.000
0,0,0,2038.000,0.000
0,0,0,2039.000,1.000
0,0,0,2032.000,0.000
0,0,0,1965.000,0.000
0,0,1,2017.000,52.000
0,0,1,2064.000,47.000
0,0,0,2081.000,17.000
0,0,0,2086.000,5.000
0,0,0,2057.000,0.000
0,0,0,2060.000,3.000
0,0,0,2051.000,0.000
0,0,0,2075.000,24.000
0,0,0,2059.000,0.000
0,0,0,2063.000,4.000
0,0,0,1983.000,0.000
0,0,1,2037.000,54.000
0,0,0,2053.000,16.000
0,0,0,2056.000,3.000
0,0,0,2031.000,0.000
0,0,0,2035.000,4.000
0,0,0,2014.000,0.000
0,0,0,2028.000,14.000
0,0,0,2029.000,1.000
0,0,0,2032.000,3.000
0,0,0,2025.000,0.000
0,0,0,2035.000,10.000
0,0,0,1964.000,0.000
0,0,1,2025.000,61.000
0,0,0,2047.000,22.000
0,0,0,2060.000,13.000
```



The result of the integration. Because of the PCB saturation, the real data from PCB looks bad. The plot from sensor 1 is the ideal result and sensor 2 and 3 is the result from the PCB.

Conclusions

Issues/Revisions	Status
The 3-cell battery of the power supply is outside the input range of the buck converter. Battery charger is regulating to incorrect voltage.	Power supply was designed to support a 5V DC input via Micro USB to mitigate issues with the battery charger voltage output.
Initially implemented pressure, acceleration, and audio sensors.	Currently only supports pressure sensor input because pressure alone is highly indicative of explosive events.
Op-amp input terminals are swapped.	Input signal needs to enter the positive terminal, but currently enters the negative terminal. Op-amp output (ADC signal input) always saturated because of this. Will order final PCB before 11/8 deadline.
The outputs of the original three sensors are plotted in the GUI.	After we made the decision to only use the pressure sensor, the application doesn't need to plot multiple sensors' data.