Portable High Energy Experiment DAQ

Ethan Barnes
Sang Hoon Chung
John Sabra

FINAL REPORT

REVISION – 1
3 December 2023

Final Report
FOR
Portable High Energy Experiment DAQ

Team 34

Approved by:

_____
Project Leader Date

_____
Prof. Kalafatis Date

_____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	12/03/2023	Ethan Barnes		Final Report

Portable High Energy Experiment DAQ

Ethan Barnes
Sang Hoon Chung
John Sabra

SCHEDULE AND VALIDATION PLAN

REVISION – 3
3 December 2023

Schedule:

ECEN 403:

Task	Deadline	Owner	Status
Write FSR, ICD, Milestones, Validation	Feb 22, 2023	All	Complete
Present Midterm presentations	Mar 8, 2023	All	Complete
Present the status update presentation	Mar 29, 2023	All	Complete
Present the final presentation	Apr 19, 2023	All	Complete
Perform final demo	Apr 28, 2023	All	Complete
Decide on microcontroller	Feb 19, 2023	Ethan Barnes	Complete
Interface I/O with microcontroller	Apr 19, 2023	Ethan Barnes	Complete
Identify battery type for power supply	Feb 19, 2023	John Sabra	Complete
Identify voltage regulation method	Feb 19, 2023	John Sabra	Complete
Decide on integrated design or separate power supply	Feb 19, 2023	John Sabra	Complete
Begin power supply PCB design	Apr 28, 2023	Complete	Complete
Pick types of sensors for design	Feb 19, 2023	All	Complete
Order Parts	Feb 24, 2023	All	Complete
Receive Sensors form Sandia	Mar 5, 2023	All	Complete
Develop GUI for users to interact with application	Mar 20, 2023	Sang Hoon Chung	Complete

Read data from SD card and compute metrics on data	Mar 27, 2023	Sang Hoon Chung	Complete
Set up the cloud server account and file directory	Apr 3, 2023	Sang Hoon Chung	Complete
Connect from the cloud server to app	Apr 10, 2023	Sang Hoon Chung	Complete
Upload and download the data to and from cloud server	Apr 17, 2023	Sang Hoon Chung	Complete
Complete Final Report	Apr 29, 2023	ALL	Complete

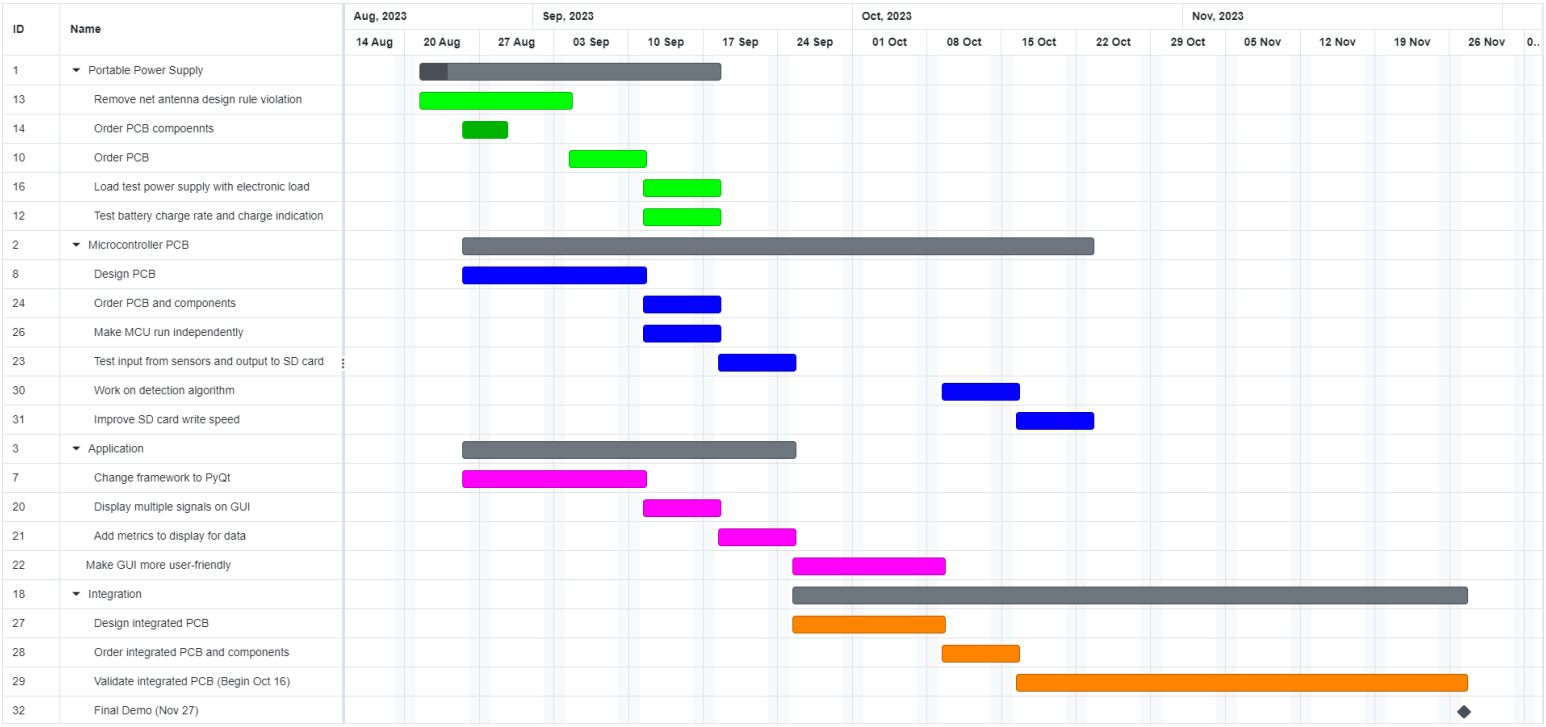
Validation Plan:

ECEN 403:

Test	Owner	Status
Able to read digital signals from sensors	Ethan Barnes	COMPLETE
Able to write to SD card	Ethan Barnes	COMPLETE
Sample signals at least 2 kHz	Ethan Barnes	COMPLETE
Write to buffer file rate 2 kHz	Ethan Barnes	INCOMPLETE
Power supply can convert input voltage to 3.3 V and 5 V	John Sabra	COMPLETE
Batteries will recharge	John Sabra	COMPLETE
Power supply maintains 3.3V at 100 mA load current	John Sabra	COMPLETE
Build the app for GUI	Sang Hoon Chung	COMPLETE
Set up the cloud server to interact with application	Sang Hoon Chung	COMPLETE
Debug for the application	Sang Hoon Chung	COMPLETE

Schedule:

ECEN 404:



Validation Plan:

ECEN 404:

Team Member	9/10	9/24	10/8	10/22	11/5
Johnny	PCB Ordered	Power supply PCB validated	Charging feature validated	Integrated PCB design complete	Integrated PCB is functional
	Zero design violations in Altium	Regulates 3.3 V under 40mA load and 5 V under 120 mA load (estimated load current of MCU)	Can charge a 3-cell lipo battery at 1C rate		

Ethan	PCB design complete	MCU runs independently	PCB is functional	Zero design violations in Altium	Can be powered with 3.3V source and can read from sensors and write to SD card at more than 2 kHz Integrated PCB is portable with power supply regulating battery voltage to 3.3 and 5V under load
	Zero design violations in Altium	Program will start and stop from GPIO outside of debug mode	Powered with 3.3 V source, can write to SD card at ~2 kHz Pressure sensors test pending		
Sang Hoon	Change Framework to PyQt and Design multiple signals on GUI	Add more metrics to display for data	Make GUI more user- friendly	Finalizing the program and adopting the software	Validate and test communication between subsystems
	Check the application shows whole sensors data	Add additional metrics and check availability of the result Display via the application at next column	Ask opinions to other people about the design of GUI	Check the software works when we download the software itself as .exe file	Application can read data from SD card in desired format (time, explosion, pressure, acceleration)

Portable High Energy Experiment DAQ

Ethan Barnes
Sang Hoon Chung
John Sabra

CONCEPT OF OPERATIONS

REVISION – 3
29 April 2023

Concept of Operations
for
Portable High Energy Experiment DAQ

Team <34>

Approved by:

_____
Project Leader Date

_____
Prof. Kalafatis Date

_____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	02/10/2023	Ethan Barnes		Revision 1
2	02/22/2023	John Sabra		Revision 2
3	04/29/2023	Ethan Barnes		Revision 3

Table of Contents

List of Tables

No table of figures entries found.

List of Figures

1. Executive Summary

This project aims to build a device able to detect the occurrence of low blast (5 lb. charge or smaller) within a 100 ft area. Our device or The Portable High Energy Experiment Data Acquisition System (PHEE DAQ) will continuously gather real time data using a sensor array and subsequently analyze this data within a processor to identify the occurrence of an explosive event. We plan to accurately detect an explosion within 100ft by utilizing acoustic, pressure, and vibration sensors. Since this device is expected to gather data continuously, the design will be self-powered with a chargeable battery. To accommodate real time data acquisition, the microcontroller will capture the rise of the event with sampling at fast rates (2 kHz or higher). The output of the microcontroller will identify whether an explosion has occurred and will be stored onto onboard storage. The main purpose of this device is for the protection of equipment such as weapons and vehicles from the damage incurred after an explosive event.

2. Introduction

The purpose of this document is to propose the design of a device capable of detecting a low blast (5 lb. charge or smaller) within 100ft. This PHEE DAQ is a system capable of gathering and analyzing data to determine the occurrence of explosive events in its surroundings by using a pressure sensor connected to a microcontroller unit. This system will be portable and will feature removable data storage so that it can be recovered at any time.

2.1. Background

Terrorism and acts of violence occur daily all over the world. Currently, in places where many US troops are deployed, the risk of bombing is always inherent. Frequently, the size of the explosive charges used in these attacks are not large. However, despite this, the power of these explosives is far greater than we can imagine, and they have the potential to harm many people [1]. To reduce the damage of these explosives, it is advantageous to detect these explosions and minimize their damage.

Sandia National Laboratories aims to design a portable device that can detect explosive charges that are 5 lbs. or less in size from within 100 ft. The working concept of this device is as follows: it gathers data continuously in real time, and the moment it detects a high energy event, it outputs whether it has detected an explosion. Subsequently, this data is saved in onboard storage for reference at any time.

Our team's goal is to design this device. We plan to design a device that can detect an explosive event by selecting the appropriate sensors, designing software that uses sampling, and filters so that the device can analyze the signal in real time as soon as it is detected [2]. This process will require a stable power source, removable storage that can store and utilize sufficient data, and integrate them onto a custom PCB. The portability of this design along with the versatility of its data inputs, making it an optimal choice for any type of data acquisition especially high energy experiments.

2.2. Overview

The network used in this device detects the explosion of various sensors (acoustic [3], pressure [4], vibration sensor [5], etc.) connected to the microcontroller. It moves the data to the microcontroller, where the filter is coded and can be sampled in real-time for analysis. This data will be saved continuously on a removable storage device. Developing the GUI for configuring DAQ settings will give the output “quick looks” of data. The Portable High Energy Experiment Data Acquisition System (PHEE DAQ) calculates the time and the weight of the explosive. This allows the user to determine which areas explosions have been detected and pinpoint areas that need to be carefully monitored.

2.3. Referenced Documents and Standards

[1]: Chong S, Long B, Maddry JK, Bebarta VS, Ng P. Acute C4 Ingestion and Toxicity: Presentation and Management. Cureus. 2020 Mar 16;12(3):e7294. doi: 10.7759/cureus.7294.
PMID: 32313735; PMCID: PMC7163342.

[2]: Skotak Maciej, Alay Eren, Chandra Namas, “On the Accurate Determination of Shock Wave Time-Pressure Profile in the Experimental Models of Blast-Induced Neurotrauma”
Accessed at https://www.frontiersin.org/articles/10.3389/fneur.2018.00052.

[3]: R. Showen, C. Dunson, G.H. Woodman, S. Christopher, T. Lim, S.C. Wilson, Locating fish bomb blasts in real-time using a networked acoustic system, Accessed at https://doi.org/10.1016/j.marpolbul.2018.01.029.

[4]: Ma, Xuejiao, Deren Kong, and Yucheng Shi. 2022. "Measurement and Analysis of Shock Wave Pressure in Moving Charge and Stationary Charge Explosions" *Sensors* 22, no. 17: 6582. https://doi.org/10.3390/s22176582

[5]: Nguyen, Hoang, Yosoon Choi, Xuan-Nam Bui, and Trung Nguyen-Thoi. 2020. "Predicting Blast-Induced Ground Vibration in Open-Pit Mines Using Vibration Sensors and Support Vector Regression-Based Optimization Algorithms" *Sensors* 20, no. 1: 132. https://doi.org/10.3390/s20010132

3. Operating Concept

3.1. Scope

The Portable High Energy Experiment Data Acquisition System (PHEE DAQ) will determine: (1) if an explosive event occurred and (2) the general size of the explosive charge in pounds (lbs) as a stretch goal. To accomplish this, various sensors will record an environment and send signals to a microcontroller which will condition and then process the signals in real time. The processed signals will then be analyzed based on an algorithm we will create and then the outputs of this algorithm will be stored locally on the DAQ along with the timestamp of the event. The local storage device will be removable so that the user can see the results.

3.2. Operational Description and Constraints

The PHEE DAQ will be used for explosives testing in controlled outdoor environments. The DAQ and its sensors will be placed in close proximity to an explosive charge where it will remain until after the explosive event occurs.

The PHEE DAQ has specific constraints specified by Sandia National Laboratories:

- It will be used in ideal environments where the only phenomenon that occurs is a single explosive event.
- The sensors on the DAQ will be at a maximum of 100 ft away from the explosive charge and at a minimum adjacent to the charge. In the case where the DAQ itself might be damaged by the explosive event, long wires will be used to connect the sensors to the DAQ which will be out of range of and protected from the blast.
- The processor on the DAQ should be capable of sampling at a rate of 2 kHz or higher.
- The processor on the DAQ also must be able to classify if an event is explosive in real time.
- A signal conditioner must be integrated into the design.

3.3. System Description

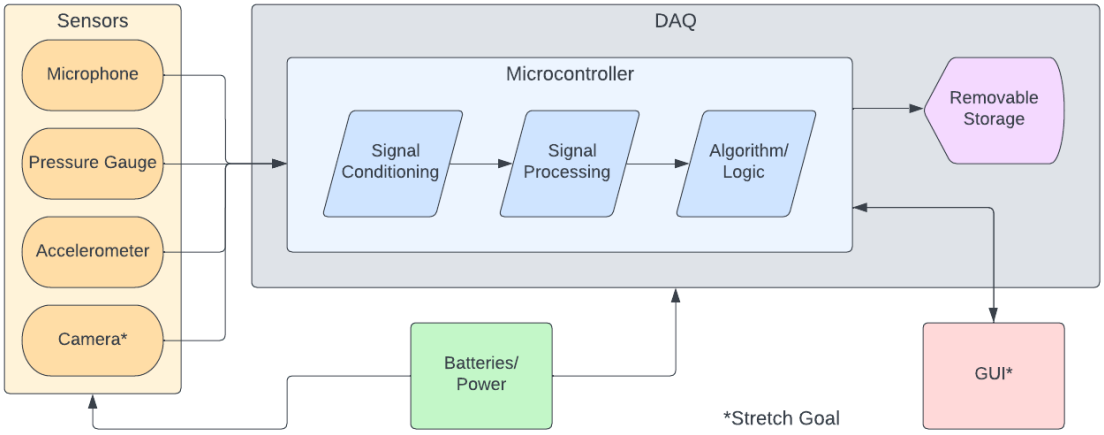


Figure 1: PHEE DAQ Block Diagram

The PHEE DAQ will consist of sensors, a microcontroller, and a removable storage device all integrated on a custom PCB which will be powered by batteries through a power supply. A GUI that can be used to receive data from and change settings on the DAQ may be added to the system if time and resources permit.

Sensors: Data will be acquired by our system through sensors that will measure the following:

- Audio: We will use a microphone that can pick up high-amplitude sound adequately to record the sound of the blast. Frequency will be important in determining if an explosive event occurred, but amplitude will be a more valuable metric considering how loud explosions are compared to other phenomena.
- Pressure: We will use either a diffusion silicon pressure sensor or a ceramic pressure sensor to measure the pressure of the air from the blast and shock waves due to the explosion. The blast wave produced from explosions, which consists of highly pressurized air, is a defining characteristic of explosions. We will use the pressure sensor to measure the high pressure from the shock front of the blast wave and the dynamic pressure of the air behind the shock front.
- Vibrations: We will use either an inertial measurement unit (IMU) or an accelerometer to measure vibrations from the blast wave. This sensor will be able to gather data on the force due to the high pressure of the shock front.
- Camera: A stretch goal may be using a camera to record video of the explosive event and using computer vision to analyze the light emitted from the explosion.

Signal Conditioning/Processing and Detection Algorithm: The data acquired through our sensors will then be sent to a microcontroller where we will use digital signal conditioning to enhance the signals by amplifying and filtering them. Then we will use digital signal processing to extract useful information from the signals to make them easier to analyze. Once our signals are in a format that is easy to understand, we will develop an algorithm that will analyze each metric in relation to time and decide if an explosive event occurred at that specific time.

Microcontroller: The microcontroller will be the brain of the entire system. We will program it to interact with the sensors, removable storage, and possibly the GUI. All our signal conditioning/processing and detection algorithm code will be run on this microcontroller and will be sent to the removable storage device and possibly the GUI. The removable storage device we will use is an SD card, so an SD card reader will be integrated into the circuit so that data can be written to it from the microcontroller.

Custom PCB: One option is that there will be a main PCB interfacing the microcontroller, sensors, and removable storage device reader and then another PCB for the power supply, power management. The other option is that there will be one PCB where all the previously mentioned components will be integrated together.

Power: To power the PHEE DAQ, we will design a PCB that will enable battery charging and distribute the power from that battery to the microcontroller and sensors that need power. This will require voltage level regulation, load management, and adjusting the charging rate of the batteries.

GUI: The GUI will display data received from the MCU onto a PC or laptop. This data will be in the form of waveforms or a summary of the data so the user can develop a general understanding of the data that has been gathered over a period of time. The GUI will also allow the user to configure settings to change how the data is sampled.

3.4. Modes of Operations

The default mode of operation for the PHEE DAQ will require that the DAQ is turned on by the user and placed in a suitable location relative to the explosive charge. The DAQ will run automatically without further interaction from the user until the user decides to remove the storage device.

The second mode of operation is a stretch goal for this project. This mode would include a GUI to allow the user to: (1) configure the necessary number of channels for optimum data sampling and (2) select different channels to display a small sample of the data (e.g., a waveform or summary of data such as maximum, minimum, and average data points).

3.5. Users

The target users of our system are researchers and engineers from government agencies, industry, and academic institutions with focuses on explosives testing and defense technology. Because the target users of our system have all the following knowledge and training, we expect our target users to have very few problems with setting up and operating our system given a user manual:

- Users of our system should have previous knowledge of how DAQs work, but our system will be simple to set up and activate with minimal technical training.
- If the GUI is implemented, the user is expected to have knowledge of high shock and explosive events so that they can adjust the DAQ's settings accordingly.

3.6. Support

A user manual will be created to specify how to operate the PHEE DAQ. It will include how to set up the DAQ and its sensors, gather data from the storage device, and interact with the GUI. Any code written for the microcontroller will be commented and README files will be included in the repository to further explain the purpose of the code.

4. Scenario(s)

4.1. #1 Sensing Explosive Events for Defense Mechanisms

The portability of the high energy DAQ system enables its use in a wide variety of applications that require the sensing of events with extreme decibel levels, vibrations, and overpressure. This includes its use as a part of military defense systems that require the sensing and detection of explosive events. In addition to the DAQ system's portability, its rapid sampling rate of real-time data makes it useful for systems that require immediate action to be taken based on the event that is sensed. More specifically, it can be used as a part of a larger system that will deploy defensive mechanisms to protect equipment or personnel from the onset damage of an explosion in the event that one had been detected.

#2 Characterization of Explosive Event (Stretch Goal)

The high energy DAQ system's number and variety of data inputs enable it to not only detect the occurrence of an explosive event, but also characterize such an event based on the estimated size of the charge. The system will be able to compare the gathered data and differentiate explosive events on the order of a few grams to tens of pounds. This data can also potentially be used to analyze the different metrics to aid in explosive weapons research. The acoustics data gathered by the system can also be valuable in research related to any type of extreme decibel-level event such as ballistics or natural disasters.

5. Analysis

5.1. Summary of Proposed Improvements

The list of improvements that the high energy DAQ system offers includes the following:

- The system will be portable and self-powered allowing it to be used in any environment without danger to the user in the context of explosive event detection.
- The sensing devices will be multi-directional allowing it detect events of interest in a wide area and without line-of-sight of the event occurring.
- With on-board memory the DAQ system will be able to store its decision regarding a detected event of interest and the time associated with it, giving the user the ability to access the recorded data at any time.
- The DAQ system will feature highly durable temperature and shock resistant sensing devices that can withstand explosions directly adjacent to them.
- The wide variety of data gathered by the system along with the on-board data and signal processing offers modularity for use in many other applications.
- As a part of the stretch goals for this design, the DAQ system will be able to not only make decisions on identifying events of interest but display data metrics on the event of interest in real-time.

5.2. Disadvantages and Limitations

The limitations associated with the high energy DAQ system includes the following:

- The design must include sensing devices that are separate from the processing unit since the processing unit is vulnerable to damage, limiting the compactness of the system.
- The sensing devices being separated from the processing unit also requires the use of long cables which are vulnerable to damage and contaminating the data with noise.
- The system's accuracy in sensing and identifying events of interest is reliant on certain explosive metric thresholds derived from previously recorded data of explosions. This may leave the system vulnerable to mis-identifying explosive events.
- To ensure data is acquired accurately, a mount will be designed to hold the sensors steady and position them correctly relative to the explosive event. As a result, the system is vulnerable to bad readings if the sensors are not mounted properly.

5.3. Alternatives

Several alternatives to the high energy DAQ system include the following:

- Use security cameras with computer vision to decide if an explosive event has occurred.
- Thermal imaging could be used to detect significant changes in temperature to identify explosions within a certain environment.
- A break trigger could be placed adjacent to an explosive charge so that an explosive event can be detected when the electrical signal through the trigger is interrupted.

5.4. Impact

There are several safety, ethical, and national security concerns associated with use of the high energy DAQ system:

- Its suitability to military defense systems designates it a technology that would be in the best interest of national security to prevent other nations or foreign threats to gain access to.
- The DAQ system gathers data that is useful for research in explosives and ballistics testing. Because of the dangerous nature of this type of research there are safety concerns that must be considered when using this system.
- Users must be properly trained on the optimal safety conditions when using this system for sensing explosives or any type of high energy event.
- The damage caused by explosive testing can be harmful to the environment if not properly contained.
- Because the DAQ system will utilize portable batteries there is a risk for batteries to explode and become a hazard to the environment

Portable High Energy Experiment DAQ

Ethan Barnes
Sang Hoon Chung
John Sabra

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 3
3 December 2023

Functional System Requirements
FOR
Portable High Energy Experiment DAQ

Prepared by:
Group <34>

_____
Author Date

Approved by:

_____
Project Leader Date

_____
John Lusher, P.E. Date

_____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	02/22/2023	Sang Hoon Chung		Revision 1
2	04/29/2023	Ethan Barnes		Revision 2
3	12/03/2023	Ethan Barnes		Final Report

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and Scope

The project aims to create a device that detects explosives within 100 feet. This device uses a pressure sensor to create a process for receiving real-time data of the detected explosion. This explosive detection device, or the Portable High Energy Experiment Data Acquisition System (PHEE DAQ), recognizes that an explosion has occurred and allows the user to protect equipment and personnel within the given range. Because this device is expected to gather data continuously, the design will be self-powered with a rechargeable battery. For accommodating real-time data acquisition, the microcontroller will capture the rise of the event with sampling at fast rates (2 kHz or higher). The microcontroller's output will identify whether an explosion has occurred and will be stored in onboard storage.

This document shows the technical requirements for the sensors and items used in the project and the systems being developed. It presents the products used in the flow diagram shown in CONOPS and specifies more detailed requirements.

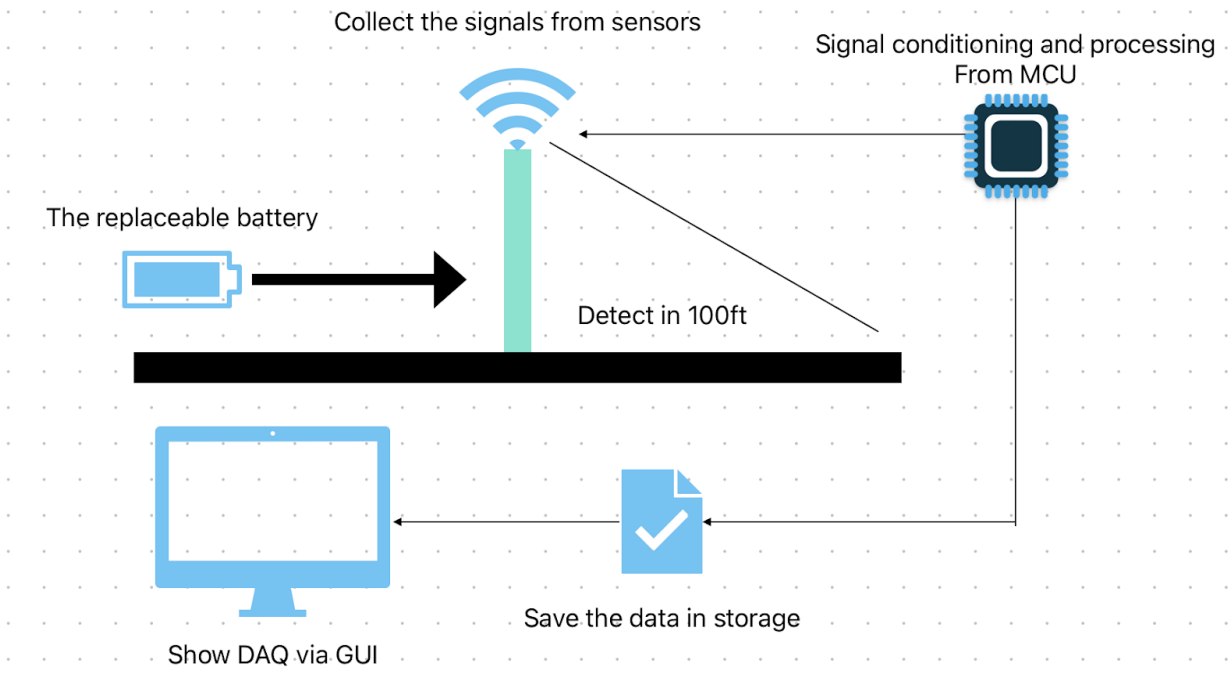


Figure 1. Project Conceptual Image

The following definitions differentiate between requirements and other statements.

Shall:	This is the only verb used for the binding requirements.
Should/May:	These verbs are used for stating non-mandatory goals.
Will:	This verb is used for stating facts or declaration of purpose.

1.2. Responsibility and Change Authority

This project is divided into three main items: a microcontroller that makes decisions based on the sensors' input, a power supply on PCB with chargeable batteries, and a PCB to integrate the microcontroller with signal processing. The team leader, Ethan Barnes, will verify all project requirements are met. These requirements can only be changed with the approval of the team leader and Professor Stavros Kalafatis.

Subsystem	Responsibility
Power Create power supply PCB capable of recharging the batteries	John Sabra
Microcontroller Create main PCB which integrates I/O with the microcontroller Create software to condition and process input signals	Ethan Barnes
Application (GUI) Create the application to analyze the result from the microcontroller	Sang Hoon Chung

Table 1: Responsibility and Change Authority

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
a1310510	Oct 2009	Experimental response of an optical sensor used to determine the moment of blast by sensing the flash of the explosion
0957-4174	Nov 2021	DeepShip: An underwater acoustic benchmark dataset and a separable convolution based autoencoder for classification
382-392	May 13, 2020	Research on piezoelectric pressure sensor for shock wave load measurement
s22176582	Aug 29 2022	Measurement and Analysis of Shock Wave Pressure in Moving Charge and Stationary Charge Explosions
s20010132	Dec 24, 2019	Predicting Blast-Induced Ground Vibration in Open-Pit Mines Using Vibration Sensors and Support Vector Regression-Based Optimization Algorithms
DS10693	Rev 10, 22-Jan-2021	STM32F446xC/E Datasheet
DB3871	Rev 6, 26-Nov-2021	STM32CubeIDE Data brief

Table 2: Applicable Documents

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
10.3389	Feb 6, 2018	On the Accurate Determination of Shock Wave Time-Pressure Profile in the Experimental Models of Blast-Induced Neurotrauma
10.1016	Jan 29, 2018	Locating fish bomb blasts in real-time using a networked acoustic system
PMC7163342	Mar 16, 2020	Acute C4 Ingestion and Toxicity: Presentation and Management
20030571	Jan 2005	Mining Publication: Detonation Wave Propagation in Underground Mine Entries

Table 3: Reference Documents

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

This section defines the minimum requirements that the development item(s) must meet. The requirements and constraints that apply to performance, design, interoperability, reliability, etc., of the system, are covered.

3.1. System Definition

The PHEE DAQ system will be used to determine if a recorded event in an environment was an explosive event or not. The system will use a pressure sensor to measure change in pressure. The analog data captured by these sensors will be filtered and conditioned using a low-pass filter and an op-amp. These analog signals will then be sent to a microcontroller where they will be converted to digital signals using an ADC. The data from ADC will be used to classify a measured event as either explosive or not explosive. The classification and the approximate time of the classified event will be saved onto an SD card which can be removed from the system so that the user can view its contents.

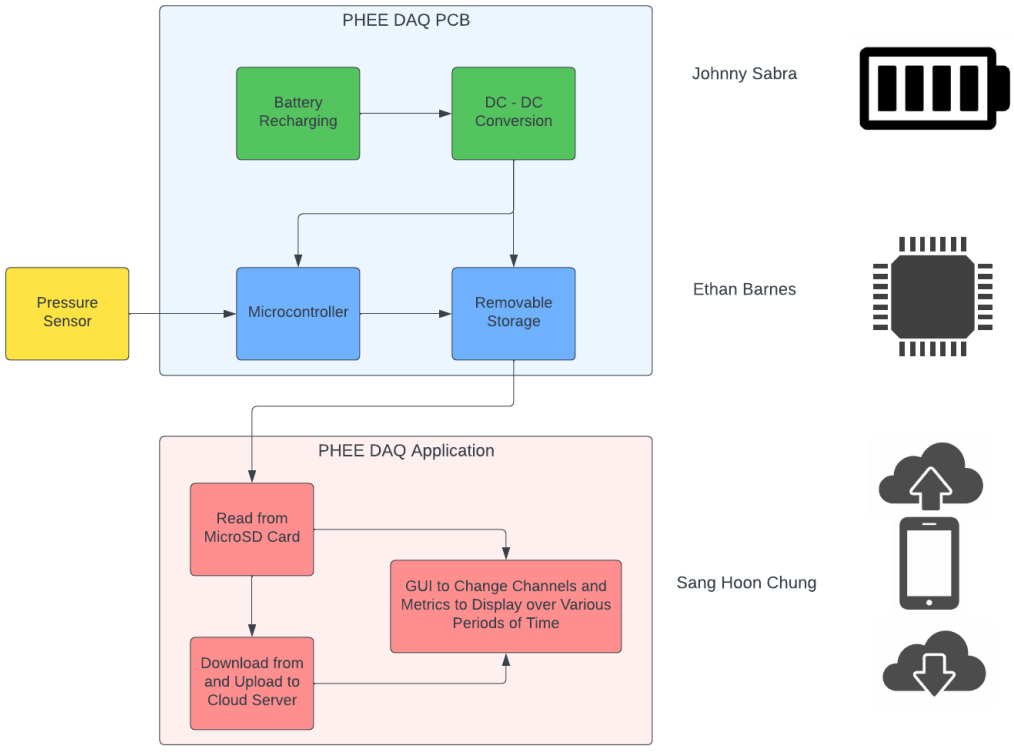


Figure 2. Block Diagram of PHEE DAQ System

The PHEE DAQ system will be divided into three main subsystems: power, microcontroller, and application.

The power subsystem will be designed to power the microcontroller, SD card reader, and any other component that will need a stepped-down supply voltage. It will also be designed

to recharge the batteries used as the supply. This will be implemented on a PCB that will be integrated to the main PCB and the sensors.

The microcontroller subsystem will consist of a microcontroller, sensor inputs, and an SD card reader integrated on a PCB. The microcontroller will be programmed to read the sensor input, write its output to an SD card and run and extract useful information from the digital signals read by the microcontroller. An algorithm we will create will then use the information gathered from the signals to determine if those signals indicate the occurrence of an explosive event or not. These results will be saved onto an SD card along with the approximate time at which the event occurred.

The proposed application is designed to read data from a micro SD card and display it in an interactive graph showing pressure versus time, with features to adjust the time frame for detailed analysis. It will include computational tools for analyzing metrics like average and peak pressure and advanced options for performing Maximum Fast Fourier Transforms (FFT) amplitude on the data. Users will have the flexibility to save their data locally or upload it to cloud storage, ensuring both convenience and data security. The application's intuitive GUI will cater to a range of users, from novices to experts, and will be optimized for performance across various platforms, ensuring a seamless and efficient user experience.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. Data Storage Requirement

The PHEE DAQ system will save results to a removable storage device.

Rationale: This is a requirement specified by our sponsor.

3.2.1.2. Classification Accuracy

The PHEE DAQ system shall meet a threshold objective of 90% accuracy.

3.2.1.3. Classification Speed

The PHEE DAQ system should generate the result in real-time.

Rationale: This is a requirement specified by our sponsor.

3.2.1.4. Battery Operating Time

The PHEE DAQ system should operate for 10 hours with 2000 mAh battery and a load current of around 140 mA. At a 3.3 V output voltage supply, it will have a run-time of 10 hours at a 200 mA load current.

Rationale: The system should be able to be on standby, therefore a specified battery operating time would allow users to predict when the batteries must be charged.

3.2.2. Physical Characteristics

The pressure sensor can withstand the force of an explosion.

3.2.2.1. Mass

The mass of the PHEE DAQ system shall be less than or equal to 1 kilogram.

Rationale: This is a requirement specified by our sponsor to consider portability.

3.2.2.2. Volume Envelope

The volume envelope of The PHEE DAQ system shall be less than or equal to 15 cm in height, 15 cm in width, and 15 cm in length.

Rationale: This is a requirement specified by our sponsor to consider portability.

3.2.3. Electrical Characteristics

3.2.3.1. Sensor Inputs

The PHEE DAQ system will take inputs from a pressure sensor.

Rationale: This is a requirement specified by our sponsor.

3.2.3.1.1 Power Consumption

With an estimated operating time of 120 hours, the maximum peak power of the system shall not exceed 11.1 watts.

3.2.3.1.2 Input Voltage Level

The input voltage level for the microcontroller from the power supply shall be between 1.7 VDC and 3.6 VDC.
The input voltage level for the microcontroller from the signals shall be between -0.3 VDC and 5.5 VDC.

Rationale: STM32F446RET6 specification, STM32F446xC/E Datasheet.

3.2.3.2. Outputs

3.2.3.2.1 Data Output

The PHEE DAQ system shall output the results of the event classification.

Rationale: This is a requirement specified by our sponsor.

3.2.3.2.2 GUI Metrics Display

The GUI of our application is designed for clarity and efficiency, displaying a concise sample of the data in two formats: a waveform for detailed, time-based analysis, and a summary view highlighting key statistics like average and peak values. This dual-view approach allows users to quickly grasp the essential aspects of the data, catering to both in-depth examination and at-a-glance understanding.

Rationale: This is a stretch goal specified by our sponsor.

3.2.3.3. Connectors

The PHEE DAQ shall use BNC connectors compatible with the sensors provided by the sponsor.

Rationale: The sensors provided by our sponsor use BNC connectors.

3.2.3.4. Wiring

The PHEE DAQ system shall follow the guidelines outlined in Figure 3 of STM32F446xC/E Datasheet.

Rationale: Conform to STM32 standard.

3.2.4. Environmental Requirements

The PHEE DAQ system shall be designed to withstand and operate in environments specified in the following section.

Rationale: This is a requirement specified by our sponsor due to constraints of the system in which the PHEE DAQ is integrating.

3.2.4.1. Ideal Environment

The PHEE DAQ system shall operate in an ideal environment (no explosive event, no extreme weather, no existing fires, normal atmospheric pressure, normal temperature).

3.2.4.2. Environment with Explosive Event

The PHEE DAQ system shall withstand an explosive event with extreme overpressure and heat.

3.2.4.2.1. Pressure

The PHEE DAQ sensors shall be able to withstand and work properly in an environment with high overpressure events.

Rationale: The overpressure generated by explosives of 5 lbs can reach up to 150 kPa.

3.2.4.2.2. Thermal

The PHEE DAQ sensors shall be able to withstand and work properly in an environment with high temperature.

Rationale: This range includes the environment of the place and the maximum temperature when the explosive happens. When the 5 lbs of TNT explodes, the temperature reaches 5000 °F within 100ft.

3.2.5. Software Specification

The PHEE DAQ system shall process the signals and give rapid results for customers who use this system.

3.2.5.1. Microcontroller

The PHEE DAQ system shall have a microcontroller that is adequate for the requirements in this section.

3.2.5.1.1. Programming Language

The software for the PHEE DAQ system shall be written in C or C++.

Rationale: STM32CubeIDE specification, page 2 of STM*STM32CubeIDE Data brief*.

3.2.5.1.1. RAM

The PHEE DAQ system shall have at least 40 kB of RAM to run the software.

Rationale: This is a requirement based on the volume of data that we are getting from a sensor and the size of the software we will develop.

3.2.5.2. Signal Processing and Conditioning

The PHEE DAQ system gets the signals from the sensors. The signals will be filtered by an operational amplifier and sampled by an analog to digital converter.

3.2.5.2.1. Specific Filter Design

The pressure sensor will need a filter that is designed depending on its expected output.

3.2.5.2.2. Sampling Rate

The PHEE DAQ system shall sample signals at 2 kHz or higher.

Rationale: This is a requirement specified by our sponsor.

4. Support Requirements

The Portable High Energy Experiment Data Acquisition System is equipped with all the parts required to successfully collect and process a signal. After storing the data that is conditioned and processed, GUI will display data received from the MCU. The system to collect the signal consists of (1) a pressure sensor, (2) a microcontroller, (3) removable storage, (4) a battery charging circuit, (5) a replaceable battery, and (6) an application that displays a GUI.

Appendix A: Acronyms and Abbreviations

SD	Secure Digital
SDIO	Secure Digital Input Output
GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
kHz	Kilohertz (1,000 Hz)
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
mA	Milliamp
mAh	Milliamp Hour
W	Watt
PCB	Printed Circuit Board
RMS	Root Mean Square
TBD	To Be Determined
USB	Universal Serial Bus
DAQ	Data Acquisition
DAC	Digital-to-Analog Converter
MCU	Microcontroller
RAM	Random Access Memory
kB	Kilobyte (about 1,000 bytes)
FFT	Fourier Fast Transform

Appendix B: Definition of Terms

Microcontroller: A small computer on a single integrated circuit chip that contains a processor, memory, and input/output peripherals.

Sampling: The process of measuring or recording a continuous analog signal at discrete time intervals.

Data acquisition: The process of measuring or acquiring data from physical or digital sources, such as sensors, instruments, or computer systems.

SD card: A type of removable flash memory card used for storing digital data.

Real-time: Refers to the immediate processing and response to input data, allowing systems to analyze and react to information as soon as it is received. (within milliseconds).

Portable High Energy Experiment DAQ

Ethan Barnes
Sang Hoon Chung
John Sabra

INTERFACE CONTROL DOCUMENT

REVISION – 3
3 December 2023

Interface Control Document
FOR
Portable High Energy Experiment DAQ

Prepared by:
Group <34>

_____
Author Date

Approved by:

_____
Project Leader Date

_____
John Lusher II, P.E. Date

_____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	02/22/2023	Ethan Barns		Draft Release
2	04/29/2023	Ethan Barnes		Revision 2
3	12/03/2023	Ethan Barnes		Final Report

Table of Contents

List of Tables

List of Figures

1. Overview

This document provides detailed interfaces for the three subsystems of this project. The ICD will include physical descriptions of the various elements and each sensors' electrical interface, power, and placement. The document will also explain how the subsystems will interface to achieve the goals and requirements mentioned in the FSR and ConOps documents.

2. References and Definitions

2.1. References

Refer to sections 2.1 and 2.2 of the Functional System Requirements document.

2.2. Definitions

mA	Milliamp
mW	Milliwatt
kHz	Kilohertz (1,000 Hz)
MHz	Megahertz (1,000,000 Hz)
kB	kilobyte (1,000 bytes)
MB	Megabyte (1,000, 000 bytes)
psi	Pounds per Square Inch
kPa	Kilopascal
dB	decibel
g (g-force)	Gravitational force equivalent
GUI	Graphical User Interface
PCB	Printed Circuit Board
SDIO	Secure Digital Input Output
VDC	Voltage Direct Current

3. Physical Interface

3.1. Weight

3.1.1. Main Control Unit

Component	Weight	Number of Items	Total Weight
Microcontroller	0.34 grams	1	0.34
PCB	~100 grams	1	~100 grams
Micro-SD card reader	2.46 grams	1	2.46

Table 1: Main PCB Weight

3.1.2. Sensor Unit

Component	Weight	Number of Items	Total Weight
Pressure Sensor	4.5 grams	1	4.5 grams
Current Source	~2200 grams	1	~2200 grams

Table 2: Sensor Unit Weight

3.1.3. Power Supply

Component	Weight	Number of Items	Total Weight
Batteries	80 g	1	80 grams
PCB	~20 grams	1	~20 grams

Table 3: Power Supply Weight

3.1.4. PHEE DAQ

Component	Weight	Number of Items	Total Weight
Pressure Sensor	4.5 grams	1	4.5 grams
Current Source	~2200 grams	1	~2200 grams
PCB	~120 grams	1	~120 grams

Table 4: PHEE DAQ Weight

3.2. Dimensions

3.2.1. Main Control Unit

Component	Length	Width	Height
Microcontroller	10 mm	10 mm	2 mm
PCB	88.9 mm	63.5 mm	N/A
Micro-SD card reader	15mm	11mm	2 mm

Table 5: Main Control Unit Dimensions

3.2.2. Sensor Unit

Component	Length	Width	Height
Pressure Sensor	52.6 mm	7.1 mm	N/A
Current Source	177.8 mm	38.1 mm	139.7 mm

Table 6: Sensor Unit Dimensions

3.2.3. Power Supply

Component	Length	Width	Height
Batteries	18 mm	34 mm	50 mm
PCB	71.6 mm	25.4 mm	2.5 mm

Table 7: Power Supply Dimensions

3.2.4. PHEE DAQ

Component	Length	Width	Height
Pressure Sensor	52.6 mm	7.1 mm	N/A
Current Source	177.8 mm	38.1 mm	139.7 mm
PCB	101.6 mm	70.4 mm	N/A

Table 8: PHEE DAQ Dimensions

3.3. Mounting Locations

The PHEE DAQ will be placed in a normal environment with no special circumstances (explosive event, extreme weather, existing fires, abnormal atmospheric pressure) within a range of 100 ft. It is assumed that explosives may be present in this area.

4. Thermal Interface

Because the PHEE DAQ system is portable and required to use batteries as its main source of power, it may require a heatsink or some type of cooling system to mitigate the effects of heat buildup. Additionally, since it is possible for it to operate continuously for 120 hours, heat buildup may be so intense that the system may require air circulation.

5. Electrical Interface

5.1. Microcontroller Specifications

Model	STM32F446RET6
SRAM	128 kB
Flash Memory	512 GB
Processor Frequency	180 MHz
ADC Channels	24
ADC Resolution	12 bits

Table 9: Microcontroller Specifications

5.2. Primary Input Power

The PHEE DAQ system will be powered by a 3.7 V rechargeable lithium ion battery. The battery will have a capacity of 3000 mAh and, with an expected working current of 25 mA, it is expected to last up to 120 hours. The power supply will have a maximum charging voltage of 4.2 V and estimated cycle life of 300 cycles where cycles represent the amount of times the battery can be charged and discharged until it reaches 80% of its capacity.

5.3. Signal Interfaces

5.3.1. Sensor

Sensor	Output Voltage	Voltage Supply
Pressure Sensor	0 V - 2.2 V	16 V

Table 10: Signal Interfaces

5.4. User Control Interface

The user control interface, designed as a Graphical User Interface(GUI), will be accessible via a computer application, offering a user-friendly and intuitive platform for interaction. This GUI is tailored to display and analyze data collected by the PHEE DAQ system, providing tools for detailed visualization and in-depth analysis of the captured information, enhancing the user's ability to make informed decisions based on the data.

6. Communications / Device Interface Protocols

6.1. SD Card Reader Interface

The MCU will transmit data to the SD card reader using 1-bit SDIO. SDIO is a communication protocol for SD cards and supports up to 4 bits of data written synchronously as well as SD card detection.

Portable High Energy Experiment DAQ

Ethan Barnes
Sang Hoon Chung
John Sabra

SUBSYSTEM REPORTS

REVISION – 1
29 April 2023

SUBSYSTEMS REPORT
FOR
Portable High Energy Experiment DAQ

TEAM 34

APPROVED BY:

_____
Project Leader Date

_____
Prof. Kalafatis Date

_____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	02/29/2023	Ethan Barnes		Revision 1
2	12/03/2023	Ethan Barnes		Final Report

Table of Contents

List of Tables

List of Figures

References

[1] L. Bai, Y. Zhao, and X. Huang, “A CNN Accelerator on FPGA Using Depthwise Separable Convolution,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 10, pp. 1415–1419, Oct. 2018, doi: https://doi.org/10.1109/tcsii.2018.2865896.

[2] “ELM - FatFs Generic FAT File System Module,” irtos.sourceforge.net. https://irtos.sourceforge.net/FAT32_ChaN/doc/00index_e.html (accessed Dec. 04, 2023).

[3] Wikipedia Contributors, “Sound pressure,” *Wikipedia*, Mar. 14, 2019. https://en.wikipedia.org/wiki/Sound_pressure

[4] “Overpressure,” Wikipedia, https://en.wikipedia.org/wiki/Overpressure (accessed Dec. 4, 2023).

1. Introduction

The Portable High Energy Experiment Data Acquisition System (PHEE DAQ) consists of three main components: a pressure sensor to detect shockwave overpressure, a PCB which includes a MCU and DC power supply, and a user-friendly application in order to conveniently inspect pressure data gathered by the DAQ. The PHEE DAQ gathers real-time data using a pressure sensor that measures the overpressure from the initial shockwave of a high-energy event. It will subsequently analyze the data gathered by the sensor within a processor to identify the occurrence of an explosive event. The PHEE DAQ will be able to detect an explosion within 100ft by utilizing a pressure sensor. Since this device is expected to gather data continuously, the design will be self-powered with a rechargeable battery. To accommodate real time data acquisition, the microcontroller will capture the rise of the event with sampling at fast rates (2 kHz or higher). The output of the microcontroller will identify whether an explosion has occurred and will be stored onto onboard storage. The application processes the data collected from the MCU to provide systematic data interaction and interpretation. After the pressure data is stored onto the DAQ's onboard storage, It will be opened to be viewed conveniently within an application that uses interactive data visualization via waveform plots and time range inspection. Additionally, the android application can upload the pressure data to a centralized cloud database to keep records of data gathered by the system.

2. Power Supply Subsystem

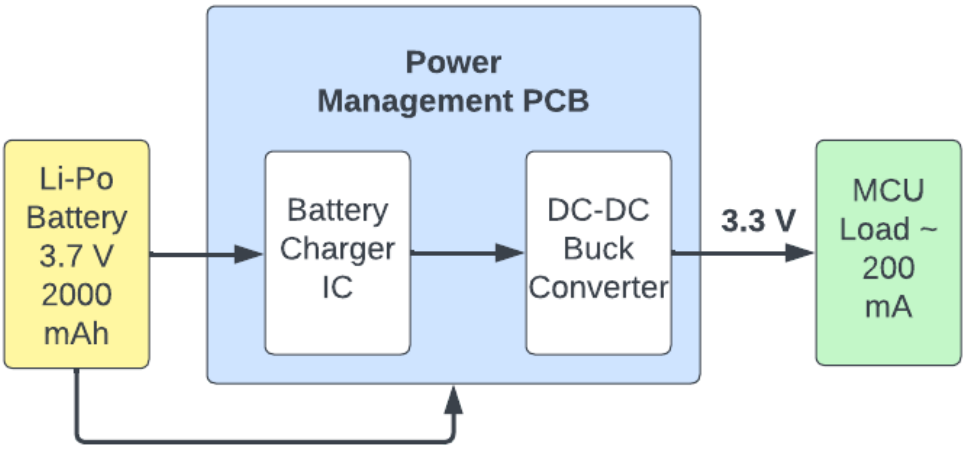
2.1. Subsystem Overview

The power supply subsystem is designed to provide power to the microcontroller unit and onboard storage while satisfying the portability requirements of the PHEE DAQ. The power supply utilizes a lithium-polymer battery as its main source. The main components of the power supply include a battery-charging circuit with battery voltage monitoring capabilities and a DC-DC buck converter to generate the required operating voltage of the MCU. The ability to monitor the battery voltage and adjust the output voltage supplied to the battery allows the battery to be charged in a safe manner. The power subsystem also features charging for the lithium-polymer battery from a set DC input to improve the subsystem's usability and usefulness for the given working environment. Along with battery-charging, the power supply subsystem also features a sleep-mode which increases the PHEE DAQs longevity, making it more viable for data capturing for long periods of time. This subsystem was tested to validate its ability to supply power to the PHEE DAQ's components in a consistent and stable manner.

2.2. Subsystem Details

The following block diagram outlines the main components of the power supply subsystem.

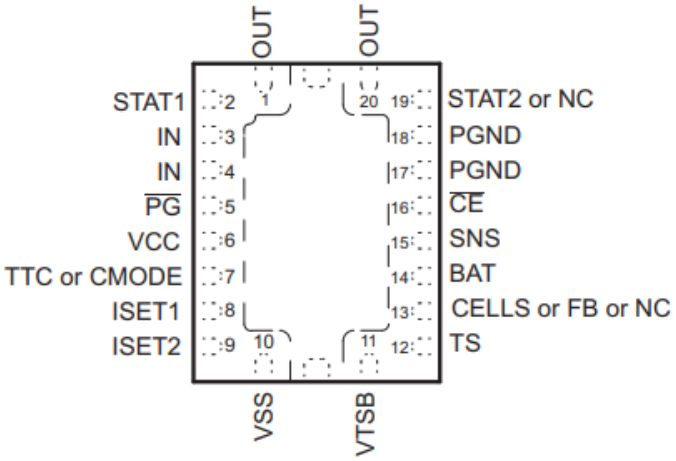
Figure 1: Functional Block Diagram of the Power Supply Subsystem



Because the nature of the working environment requires that the PHEE DAQ is powered for long periods of time it was required to use a high capacity power source. The battery chosen to satisfy this requirement was a 3.7 volt 2000 milliamp-hour lithium-polymer battery which would provide the PHEE DAQ with about a 10-hour run-time at a load current of 200 mA. Additionally, the battery voltage falls well within the input range (2.5V - 5V) on the buck converter used in this design. As shown in figure 1, the power path IC takes a set DC input to charge the battery at a programmable current rate that is set by an external voltage divider circuit. For this design the battery charger expects a set constant voltage at its feedback pin (FB, shown in figure 2, to set the constant current that it applies to the terminal

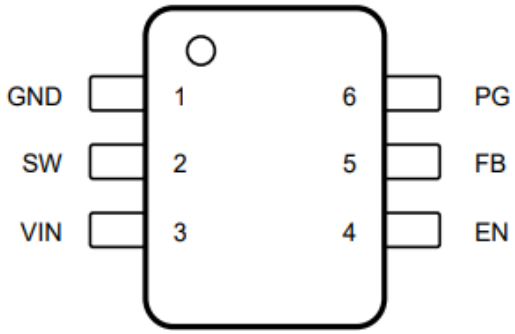
of the battery to ensure safe constant current-constant voltage charging. Therefore, the voltage divider circuit halves the battery voltage to ensure the constant voltage rate is set well within safe limits for the battery. Additionally, the battery charger IC has the capability of monitoring the temperature of the battery to ensure that it is within a safe range to begin charging. It accomplishes this by comparing the voltage applied at the TS pin shown in figure 2 to a voltage range that determines the safe operating temperature depending on the battery used. For this design, a constant voltage can be applied to this pin to bypass the feature and allow the IC to think the battery is within the same operating temperature range at all times.

Figure 2: Pin Diagram of the Battery Charger IC



The power path IC selected for this application can take up to 20 volts for its charging input and its input current limit was set to 100 mA for validation. The buck converter selected for this application was chosen based on price and was set to output 3.3 volts by using a voltage divider at its output. The output voltage that the buck converter can supply was determined by a voltage divider connected to the feedback pin (FB), shown in figure 3, where the IC would compare the voltage applied FB and adjust the duty cycle of the PWM accordingly to achieve the desired output voltage of 3.3 V. This feature enables the buck converter to have a wide input range from 2.5 to 5.5 V, giving the design flexibility in the choice of batteries used.

Figure 3: Pin Diagram of the Buck Converter



2.3. Subsystem Validation

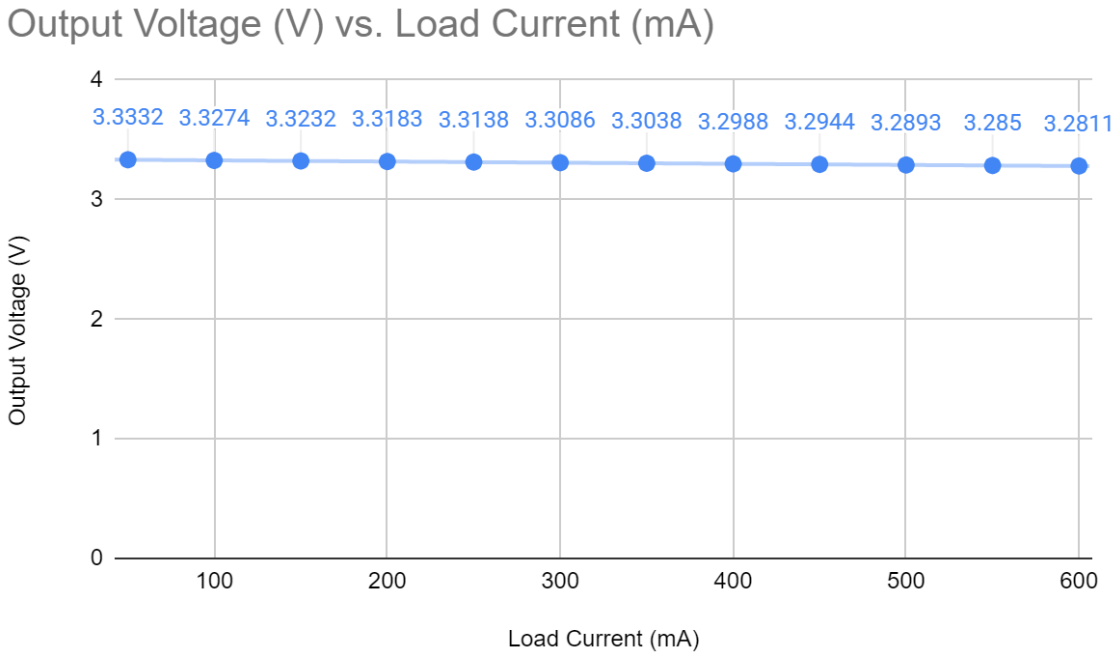
To ensure that the power supply subsystem would perform as it was designed in powering the main component of the PHEE DAQ it was validated under different loads defined by the expected current draws of the microcontroller unit and onboard storage. Specifically, both the microcontroller, which for our application the STM32 was selected, and the micro SD card had an operating voltage of 1.2 to 3.6 volts with a recommended operating voltage of 3.3 volts and current draw listed in Table 1.

Table 1: Current Draw of PHEE DAQ Components

Component	Current Draw
STM32 Microcontroller	Average 100 mA
ICP® Pressure Sensor Model 113B28	0 mA (powered by signal conditioner)
ICP® Microphone System Model 378C10	0 mA (powered by signal conditioner)
Accelerometer Model 3263AT	0 mA (powered by signal conditioner)
MicroSD Card SanDisk SDSDQM-004G	Average 40 mA

The procedure for validating if the 3.3 voltage output of the power supply system was capable of powering the STM32 and the MicroSD card included using an electronic load on the output of the DC-DC buck converter to draw set currents. This procedure validated that the recommended operating voltage of the PHEE DAQ's components could be maintained at all necessary load currents. The results of this procedure are presented in figure 4 where it can be observed that the voltage level measured at the output of the buck converter was within the required operating range at an average for all load currents. It can be observed that the output voltage is regulated to well within the operating voltage range of the MCU (1.6 - 3.6 V) even at load currents that are much higher than the current draw of the MCU.

Figure 4: Power Supply Load Regulation

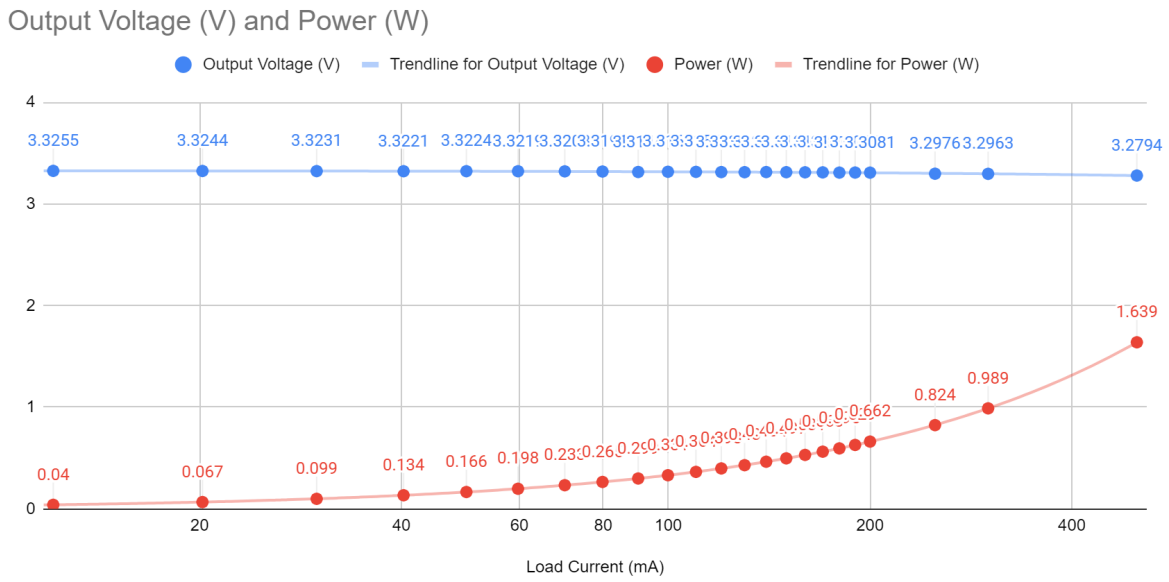


In addition to its ability to maintain a voltage within the required range for all load currents, the power draw of the power supply subsystem was measured for different load currents. The result of these measurements are displayed in figure 5 where the power consumption of the power supply is plotted versus the load current applied to it. As the load current increased, the power consumption of the power supply increased with it, while the regulated output voltage remained constant at around 3.3 V only seeing significant decreases around the 400 mA load current range. The summary of the power supplies output voltage, power draw, and voltage ripple is summarized in table 2. While the pressure sensor was gathering pressure data and the MCU wrote the data onto the SD card, the measured operating voltage of the MCU was 3.3183 V.

Table 2: Power Supply Validation

Average Output Voltage	**3.3064 V**
Output Voltage Under Load	**3.3183 V**
MCU Measured Nominal Voltage	**3.31784V**
Average Power	**1.071666667 W**
Avg Voltage Ripple	**0.01716674684 V under load**

Figure 5: Power Consumption

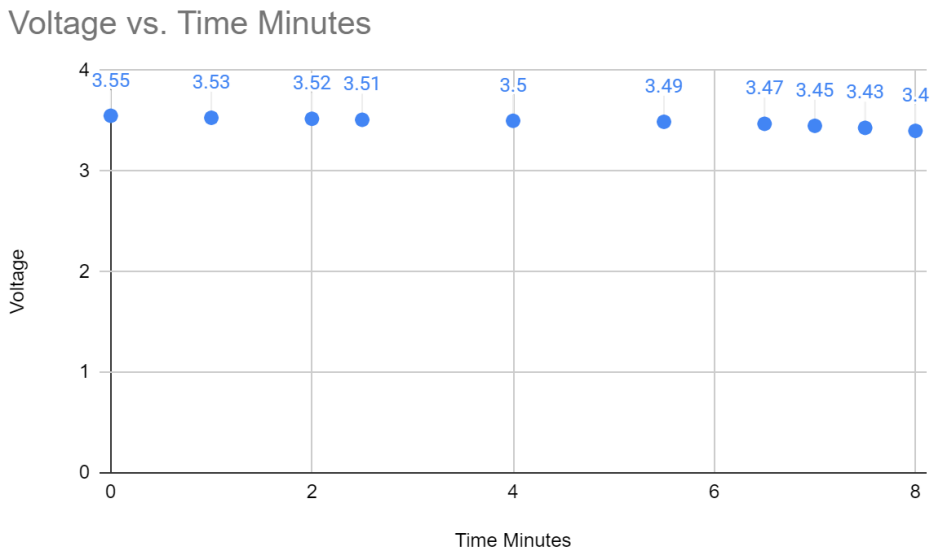


A 2 Ah battery at 3.3 volts yields an hourly power consumption of 6.6 Wh. With a runtime of 10 hours there is a power consumption of 66 watts for each battery life cycle.

The battery charging capabilities of the power supply was validated using a single-cell battery charger IC. With a measured starting voltage of 3.78 volts and set constant current of 100 mA, a battery voltage of 3.88 volts was measured after 15 mins of charging. This yields a charging rate of 0.4 V/hr.

To characterize the behavior of the 3.7V lithium-ion battery during discharge a battery discharger was used to measure the voltage of the battery when a 1.5 A constant current discharge rate was applied to it. Figure 6 shows the decrease of the battery voltage versus time. This testing was done in order to determine the optimal discharge rate for the battery being used in this power supply design.

Figure 6: Lipo Battery Discharge With Battery Discharger



2.4. Subsystem Conclusion

The power supply subsystem is designed to provide portable power to the main components of the PHEE DAQ. It utilizes a lithium polymer battery as its main power source and regulates the batteries voltage to the required 3.3 volts required for the microcontroller unit and onboard storage. The power supply subsystem was validated to be able to maintain a voltage within the required range for all load currents. Additionally, its battery-charging capability was validated by measuring the increase in voltage over time when its constant current setting was set to 100mA. The constant current and constant voltage limits were set by applying constant voltage to the feedback pin via voltage dividers that set the charging rates within safe operating range of the battery.

3. Microcontroller Subsystem

3.1. Subsystem Overview

The microcontroller (MCU) subsystem consists of two main components: (1) a program to collect data from an analog-to-digital converter (ADC), determine whether that data indicates that an explosion occurred, and save that data to an SD card and (2) a printed circuit board (PCB) that consists of sensor I/O, the MCU chip itself, and an SD card reader.

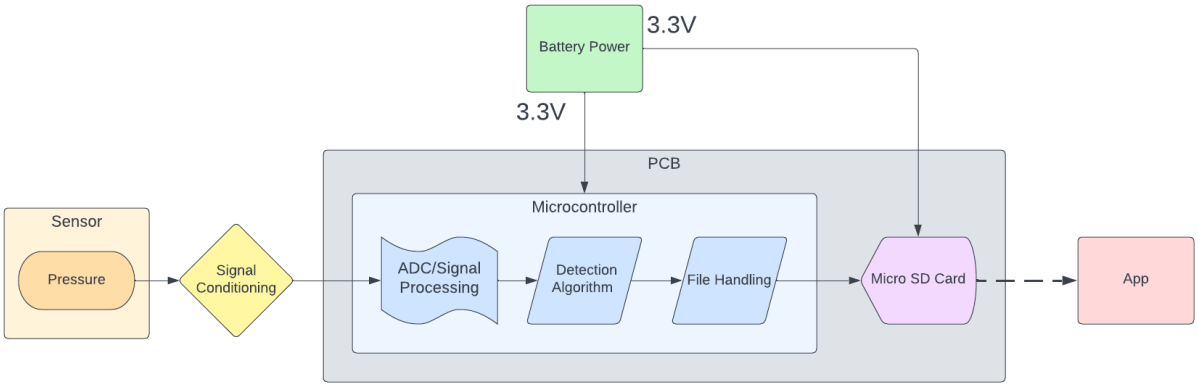


Figure 7: PHEE DAQ Block Diagram

3.2. Subsystem Details

3.2.1. GPIO and STM32 Setup

The setup process for the STM32 MCU consisted of allocating pins for sensor input, SDIO interface, internal clocks, and debugging (see figure below). The built-in ADC was used to read from four sensor inputs to 4 channels of the ADC (Ch.1 Sound, Ch.2 Pressure, Ch.3 Acceleration x-axis, and Ch.4 Acceleration y-axis) with a resolution of 12 bits. The clock for the ADC is taken from the main clock of the MCU (180 MHz) and divided down to 22.5 MHz.

For the remainder of this report, only Ch.2 Pressure is taken into account. This is because our team decided, with the approval of our sponsor, that overpressure alone is highly indicative of an explosive event. Audio and acceleration data would be useful when determining if an explosive event occurred because explosions are loud and the overpressure from the explosion physically interacts with the environment which will cause an accelerometer to experience acceleration. However, there are other natural and unnatural events that are extremely loud (lighting/thunder, heavy machinery, etc.) and could cause a high acceleration on an object (extreme winds, kicking the accelerometer, etc.) where this wouldn't be possible as easily with pressure. We are looking for a massive change in pressure from the pressure sensor (up to 150 kPa) which is rare to experience outside of an explosion. Also, the waveform of a blast wave from overpressure is easily modeled off of the Friedlander waveform, which shows a quick and extreme change in pressure followed by decreasing pressure until negative pressure is experienced,

which then returns back to the normal pressure that was originally experienced outside of the blast wave. The first change in pressure caused by the leading shock front can be detected easily, and this alone can be used to detect an explosive event.

SDIO was used to write to the SD card with 9 pins to interface with the MCU. There are only 5 pins being used in the current design iteration: VS, GND, CLK, CMD, and D0 (voltage supply, ground, clock, command, data0) where VS and GND are not dependent on the MCU. The MCU currently supports SDIO 1-bit read/write.

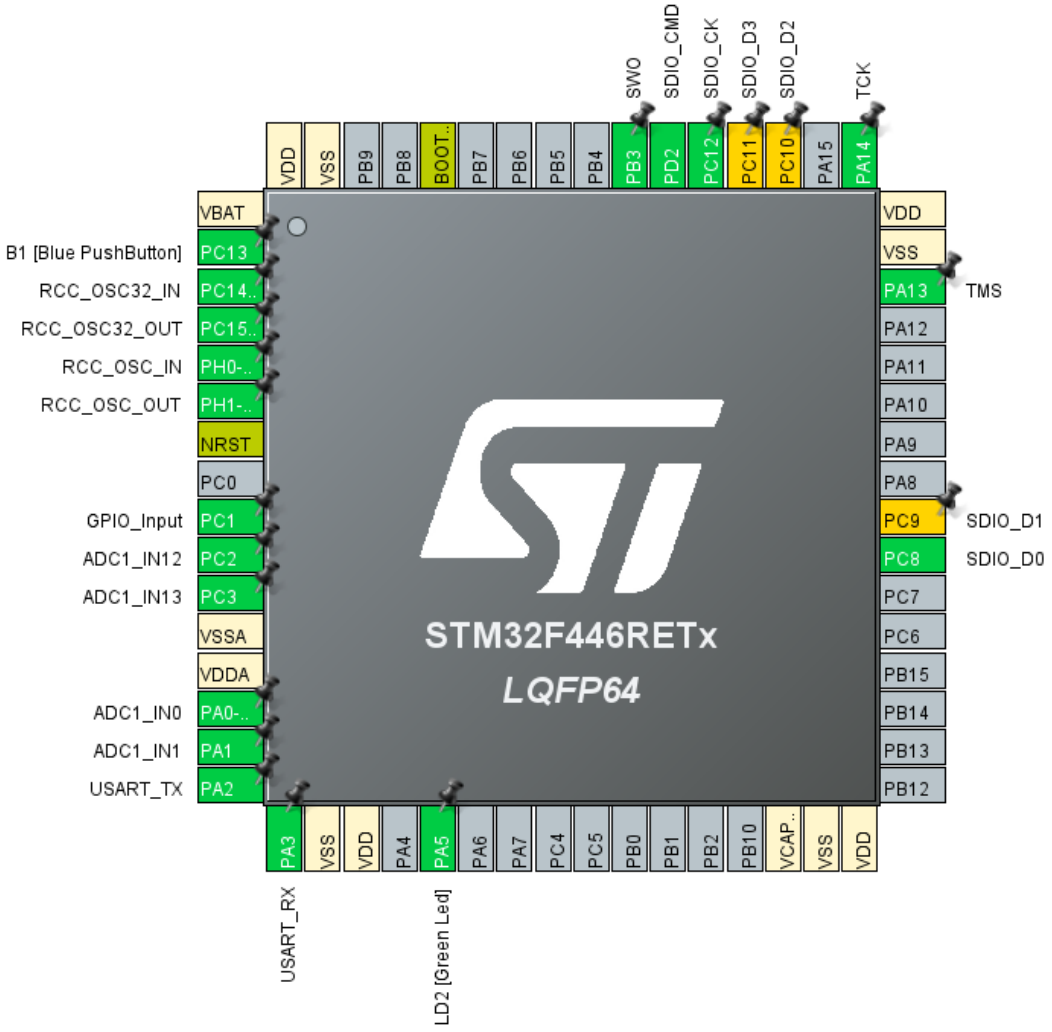


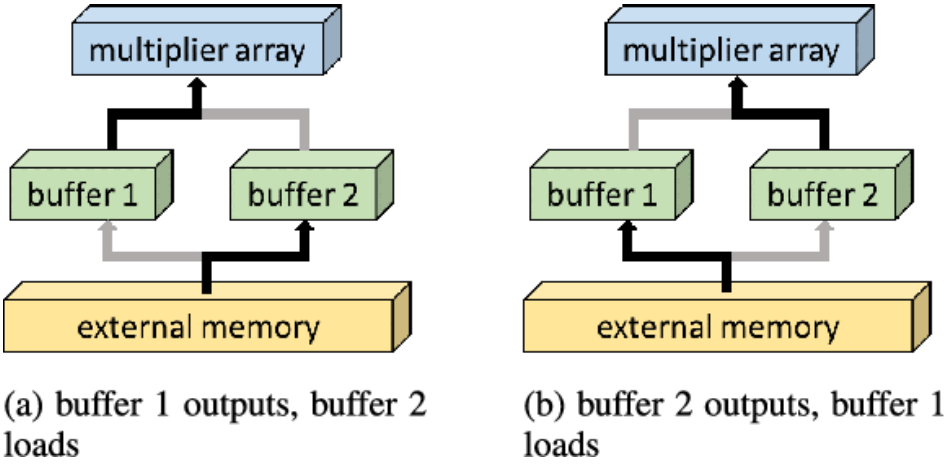
Figure 8: STM32 Pinout

3.2.2. ADC “Ping Pong” Buffer with DMA

The ADC is implemented using direct memory access (DMA) and a “ping pong” buffer. DMA allows the ADC to transfer the data sampled directly to memory without passing through the CPU, greatly reducing the workload of the CPU. This is then paired with a ping pong buffer which holds 400 samples at a time, where every 4

samples is one “complete” sample. A “complete” sample means one full sample with a reading from each of the 4 sensors. The same sensor is read every 4 channels since each of the 4 channels corresponds with a different sensor, so after 4 channels are read from, the cycle repeats. The buffer immediately starts filling up when the HAL_ADC_Start_DMA() function is called, and when it is half full, a flag is set which causes the processData() function to be called. This function will then take every sample in the buffer and append them as a formatted string to a buffer that will eventually be written to the SD card. As soon as this function is called, the second half of the ADC buffer will begin filling up in the background. Once the second half of the buffer is filled, the processData() function will already have finished executing for the previous half of the buffer so that it can be called again for the second half, where the first half of the buffer will begin to be overwritten and the cycle restarts This allows a nonstop collection of data as well as efficient use of CPU cycles. The figure below is a visual representation of the ping pong buffer process.

The new design for this buffer is essentially the same, except audio and acceleration data are no longer being captured. The buffer is still filled with data reading from the pins where the data is expected to be read from, but in this iteration of the design, we did not include an audio sensor or accelerometer. Now, the buffer fills with this data (although useless) as well as the pressure sensor data, but the audio and acceleration data are not used. Only the data from the pressure sensor is taken from the DMA buffer and written to the SD writing buffer than will eventually be saved to the SD card.



(a) buffer 1 outputs, buffer 2 loads

(b) buffer 2 outputs, buffer 1 loads

Figure 9: Ping Pong Buffer [1]

3.2.3. Explosive Event Detection

The process of detecting an explosion begins with identifying the values for SPL, pressure, and acceleration which indicate that an explosion has occurred. We found these values to be around 140 dB for SPL, 150 kPa for pressure, and 100 g for acceleration. When the ADC reads a value from a sensor, it converts the raw reading into a value in the units specified for each metric, and it compares that value with the value of the previous reading from the same sensor. The difference between these values (current sample - previous sample) is calculated and then compared to the

constant threshold values defined. If the difference (delta) is greater than the set threshold, then a bit will be set to 1 for that specific complete sample to indicate that an explosive event was detected. Because it is expected that no other phenomena will occur (as specified by the sponsor; an ideal environment with readings of 0 from sensors by default), this method of detecting explosive events is sufficient.

The new design for event detection does not consider audio or acceleration data, so whether an explosive event occurred or not is determined solely by the change in pressure between two pressure sensor samples. Our new design uses a lower threshold for pressure to detect an explosive event (10 kPa) because glass breaks and doors can be blown off of their hinges at an overpressure of around this level [4].

3.2.4. File Handling

In order to communicate with the SD card, File allocation table File system (FatFs) middleware which was built-in to STM32 was used. Functionality from FatFs included mounting/unmounting the SD card, reading/writing to and from the SD card, and other functions for file and directory access.

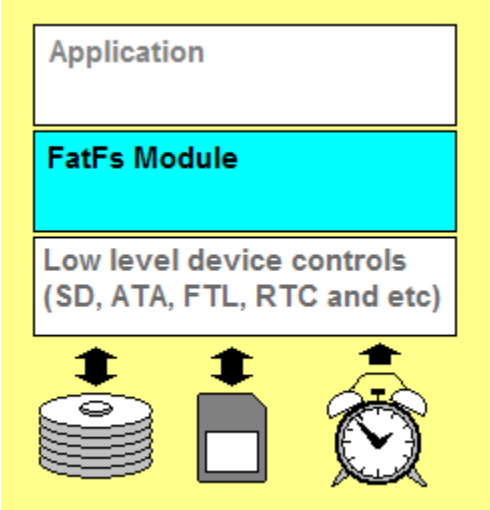


Figure 10: FatFs Module [2]

FatFs was used to implement a system of logging and saving data using a buffer file. Whenever the ADC buffer is filled half way as previously mentioned, the data collected is appended to another to the SD buffer (char array). Currently, the function responsible for this process appends 50 complete samples to the char array every half-cycle of the ADC buffer (400 samples total, 200 samples per half-cycle, 50 “complete” samples per half-cycle). This process is set to repeat 200 times, so the buffer is full when it reaches 10,000 complete samples. Each set of samples from the same ADC buffer half-cycle are all labeled with the same value to represent that specific “cluster” from which they originated. The entire buffer is then written to a buffer file on the SD card called “ADC_DATA.CSV”. Each sample from the same complete SD buffer is also labeled with a value to represent that specific “batch” from which they originated. Each sample currently contains the following information:

batch, cluster, explosionDetected, current_audio, current_pressure, current_acc, delta_audio

The explosionDetected variable can either be 0 or 1, so when any of the 10,000 samples in a batch have their explosionDetected bit set to 1, that entire buffer file is saved under a new name titled “BATCH_n.CSV”, where n is the batch number. A new buffer file with the same name as the previous buffer file is then created, and the process restarts for the next batch of samples. When the program is done taking in samples (set number of samples, set time, or user interrupt), then the final buffer file is closed and the SD card is unmounted.

The new output format to the SD card is as follows:
batch,cluster,explosionDetected,current_pressure,delta_pressure

Only pressure data is needed, so this is the only data that is saved out to the SD card, which will then be loaded into the Application subsystem that will display pressure data and indicate to the user if an explosive event occurred.

3.3. Subsystem Validation

3.3.1. Sampling Rate

The sampling rate using SDIO 1-bit is around 3.75 ksps which is higher than the minimum sampling rate requested by our sponsor.

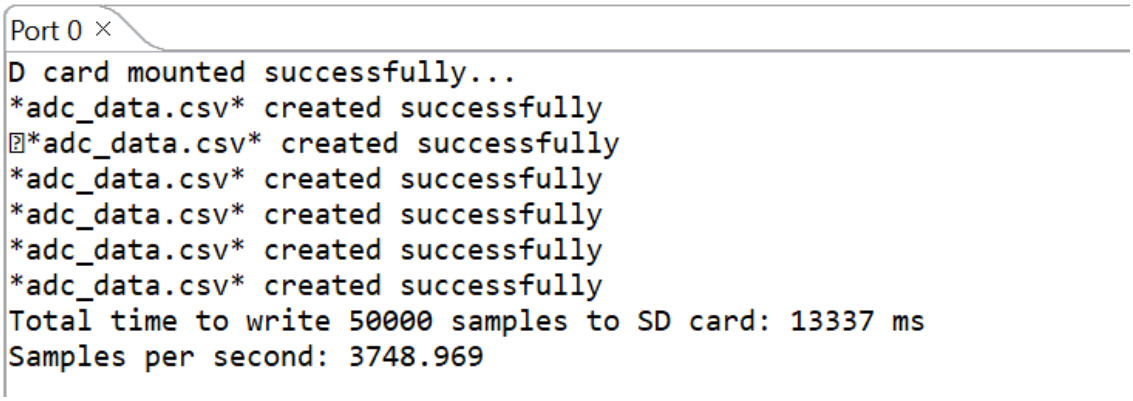


Figure 11: Sampling Rate and File Handling Time

3.3.2. Output Format

The current output in the buffer files when they are saved follow the format shown below. These files are read by the Application subsystem so that the information they contain can be displayed to the user. Since we are not considering negative pressure because it is not needed in our application, any change in pressure that is negative is unaccounted for and marked by a 0 change in pressure.

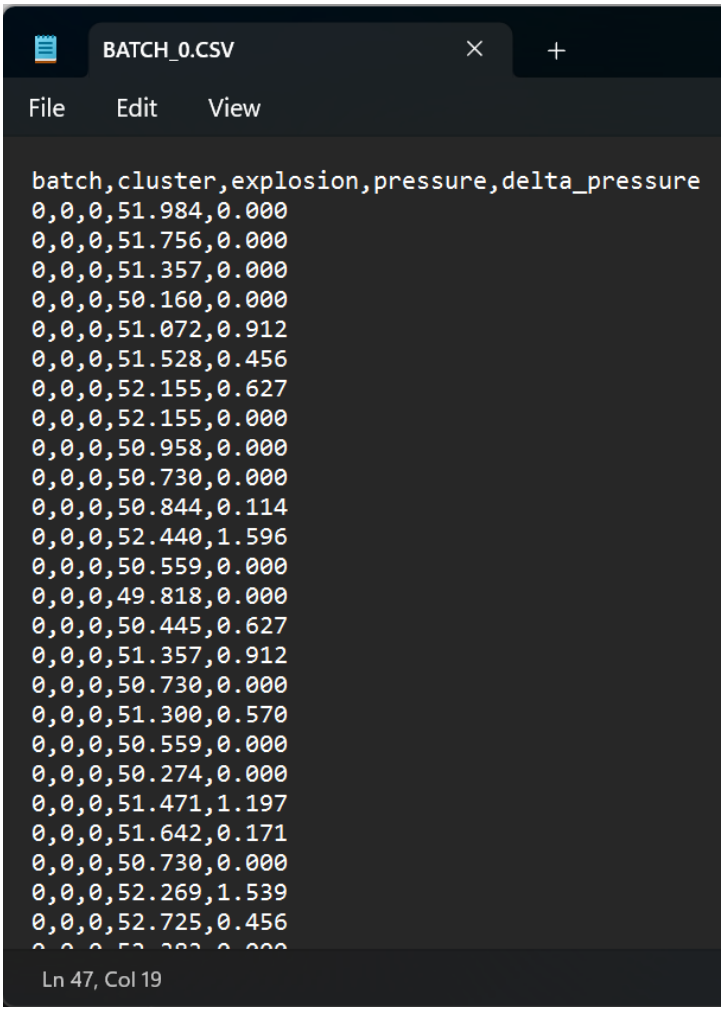


Figure 12: Sample Format

3.3.3. Sensor Values

As shown in the images below, the raw voltage read by the STM32 can be accurately converted into the units of each of the sensors. When a voltage is read by the STM32, it is a multiple of some internal reference voltage. In the case of the STM32F446RE, that is 1.2 volts. In order to convert this reading into an actual voltage, the reading must be divided by 1.2 (or multiplied by 0.833). Once this conversion is done, the MCU will know the voltage being read by each sensor. The next step is to convert the voltage to the correct metric for each sensor using the reference sensitivity specified by the manufacturer. Each voltage reading from each sensor is divided by the corresponding reference sensitivity to get the appropriate units. The only exception is the microphone, which did not have a reference sensitivity from the manufacturer and is assumed to be 1. This value is also then converted from pressure in Pa to sound pressure level (SPL) in dB using the following equation, where p is the sound pressure reading and p_o is the threshold of hearing (0.00002 Pa):

$$L_p = 20 \log_{10} \left(\frac{p}{p_0} \right)$$

Figure 13: SPL Conversion Formula [3]

Sensor	Reference Sensitivity
Microphone	1.00 mV/Pa
Pressure Sensor	14.62 mV/kPa
Accelerometer (x-axis)	50.00 mV/g
Accelerometer (y-axis)	48.97 mV/g

Table 2: Sensor Reference Sensitivities

The first two graphs and the first statistics metrics table below show that the MCU can properly convert input voltage (validated with 1 V input) to the correct units for the microphone and pressure sensor.

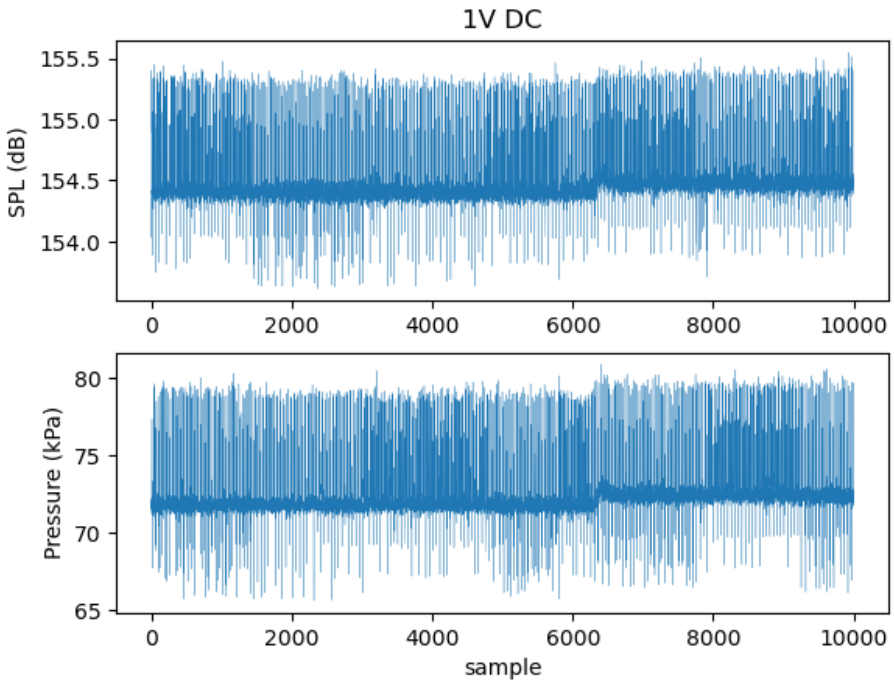


Figure 14: SPL and Pressure Plots

	batch	time	explosion	audio	pressure
count	10000.0	100000.000000	100000.0	100000.000000	100000.000000
mean	0.0	99.500000	0.0	154.504054	72.601720
std	0.0	57.734594	0.0	0.241541	1.934529
min	0.0	0.000000	0.0	153.610000	65.606000
25%	0.0	49.750000	0.0	154.431000	71.990000
50%	0.0	99.500000	0.0	154.465000	72.275000
75%	0.0	149.250000	0.0	154.492000	72.503000
max	0.0	199.000000	0.0	155.563000	81.281000

Figure 15: SPL and Pressure Averages

The next graph and the next statistics metrics table below show that the MCU can properly convert input voltage (validated with 1 V input) to the correct units for the accelerometer.

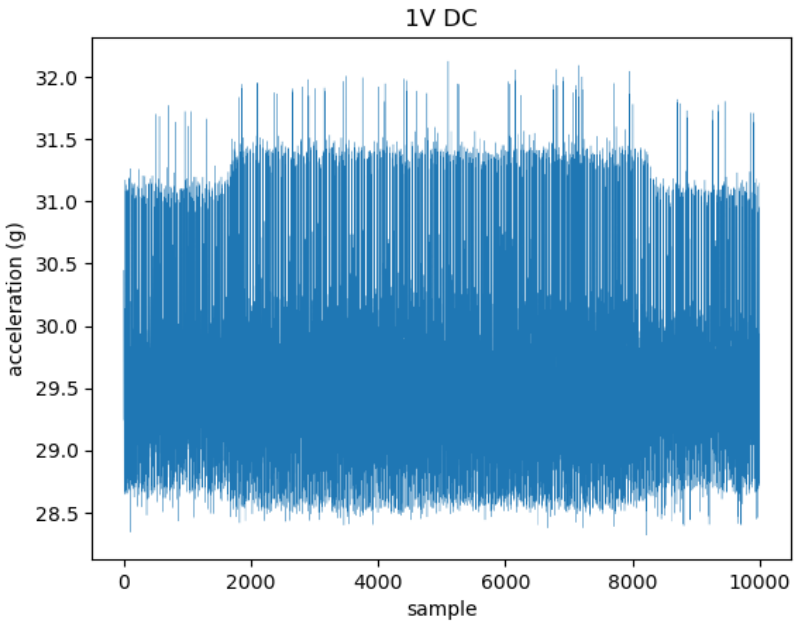


Figure 16: Pressure Plot

	batch	time	explosion	audio	pressure	acceleration
count	10000.0	100000.000000	100000.0	100000.000000	100000.000000	100000.000000
mean	0.0	99.500000	0.0	154.580854	66.856854	29.758409
std	0.0	57.734594	0.0	0.143614	1.267134	0.686937
min	0.0	0.000000	0.0	153.994000	62.244000	28.293000
25%	0.0	49.750000	0.0	154.520000	66.005000	29.427000
50%	0.0	99.500000	0.0	154.601000	66.746000	29.772000
75%	0.0	149.250000	0.0	154.675000	67.658000	29.928000
max	0.0	199.000000	0.0	155.015000	71.477000	32.126000

Figure 17: Pressure Average

3.3.4. Sensor Output

The images below show that each sensor was able to output readings when powered by the signal conditioner. There is some noise for each output, but when air is blown into the microphone and pressure sensor, the corresponding output appears. When the accelerometer is moved, the corresponding output also appears.

Figure 18: Microphone Output

Figure 19: Pressure Sensor Output

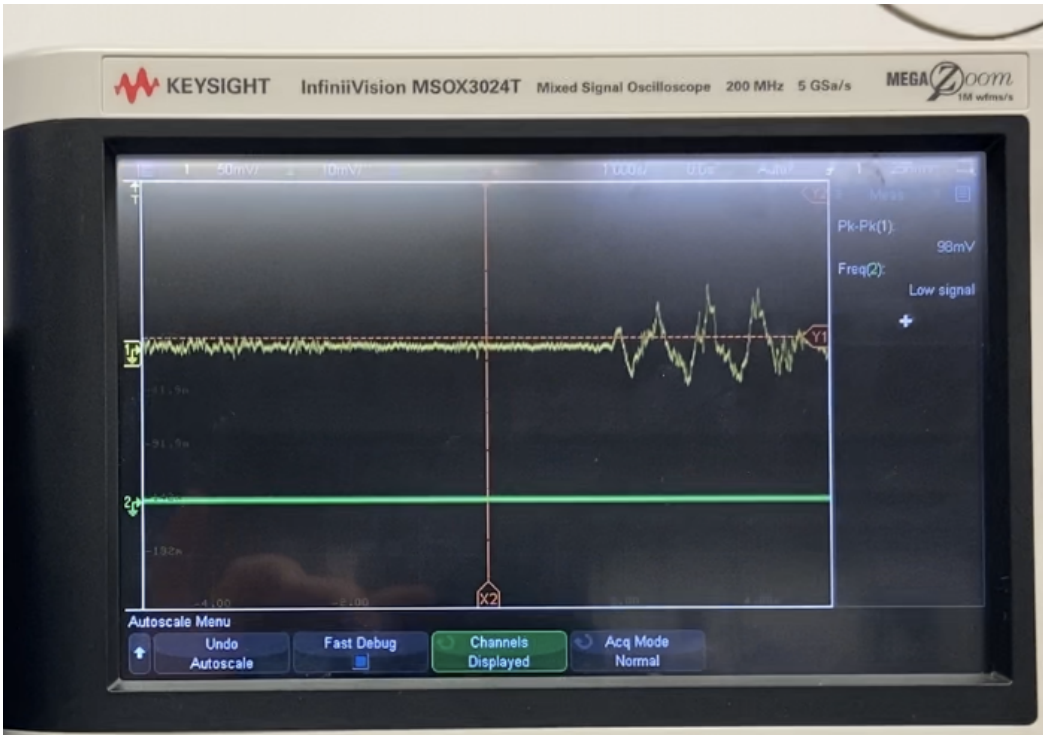


Figure 20: Accelerometer (z-axis) Output

3.4. Subsystem Conclusion

Overall the validation is all complete except for a few minor things that will be implemented during the next design iteration when the subsystem becomes more robust and needs to integrate with the rest of the subsystems. The main piece of validation missing is that the sensors cannot actually be read by the STM32 at this time due to the output voltage being too high.

4. Android Application Subsystem

4.1. Subsystem Overview

The main purpose of the Application subsystem is to develop a GUI to check the result of the stored data (DAQ) for PHEE DAQ. To build this App, we used Python with the PYQT framework. When we run the App after connecting the SD card to a PC, it informs us of the detailed information of the data (Max, Min, Average, Max FFT amplitude, etc..), and if an explosion occurred. A time range can also be set to view data from a specific starting time to a specific ending time. If we want to check the signal that occurred at a specific time, enter the appropriate time period and press the OK button to execute. We can save and upload data to the SD card and the Google Drive by connecting to the Google Cloud server.

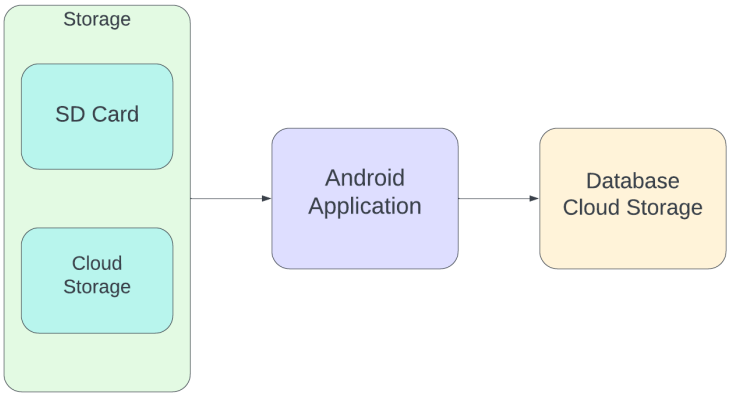


Figure 21: Functional Block Diagram of the Application Subsystem

4.2. Subsystem Details

4.2.1. Program Language

We built this app using Python with the PYQT framework. This powerful framework for Python is a good option for beginners and experts alike, as it follows industry standards and has features that help speed up the application development process. The reason why our team chose this framework is that PYQT, a framework for Python, is open-source and multi-platform to develop applications with complex features, friendly user interface, and multi-touch properties, all from an intuitive tool; we aim for efficient designs that quickly prototype, have reusable code, and are easy to deploy.

One of its biggest potentials is its high support for different input devices and protocols and its ability to develop native applications that can be ported to different operating systems. This will undoubtedly help us save time more efficiently and make it easy to access and use Google Drive app for the Google Cloud server.

4.2.2. Cloud server system

4.2.2.1. Choosing the server

We set the primary goal of storing data using the Google Cloud Platform. Amazon Web service or Microsoft Azure can also be used, but we decided to use Google Cloud Platform because of the advantages of storing data and ease of use. The final reason we use the server is to store and store data and call the saved data back into the app to analyze it.

4.2.2.2. The procedure to connect with app

When opening a CSV file in the application, the 'open CSV' button is connected to the file explorer to find the file. If we download the Google Drive app to our PC in an online environment, we can connect Google Drive to the file explorer. If we set the environment like that, we can open files in Google Drive through the app or in external storage through the app and then save them after setting the time range (you can save them even if you do not set the time range).

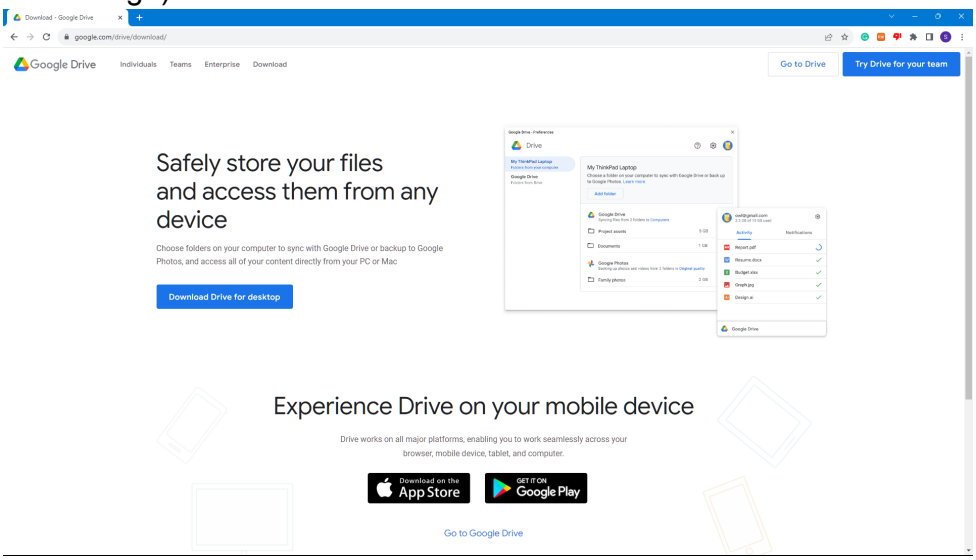


Figure 22: Screenshot for downloading the Google Drive app.

4.2.3. Application Detail

This app was created to view the blast information received by the sensor. The app displays the signal's shape when we select the file we want to see. The information we can know through this application is the Max, Min, and Average of the data points, the Max FFT amplitude of the signal, the explosion's presence, the explosion's duration, the signal's shape and the signal's frequency. Also, if we enter a specific time zone, the signal of that time zone is displayed, and signal information of that time zone is provided.

4.2.3.1. Max, Min, and Average of the Signal

The maximum point of a blast signal can be used to determine the magnitude of the explosion. This information can help assess the potential damage that may have occurred or in designing structures that can withstand such an explosion.

The minimum point of a blast signal can be used to determine the ground motion that was generated by the explosion. This information helps assess the potential for ground movement-induced damage to nearby structures.

The average point of a blast signal can be used to determine the attenuation or the rate at which the signal decreases in strength as it travels away from the explosion. This information helps assess the potential for damage to structures located at different distances from the explosion.

Overall, analyzing the max, min, and average points of a blasting signal can provide valuable information for assessing the potential damage caused by an explosion, designing structures that can withstand such an explosion, and understanding the characteristics of the surrounding environment.

4.2.3.2. The Frequency of the signal

Analyzing the frequency value of a blast wave is vital for understanding the explosion's energy content, assessing potential structural damage, and designing effective safety measures. High-frequency components indicate a more powerful explosion, allowing for the identification of the explosive material used and the evaluation of its impact on nearby structures, particularly their resonant frequencies. This information is also instrumental in forensic analysis and developing early warning systems, making frequency analysis a key tool in preventive safety measures and post-explosion investigations.

4.2.3.3. The Shape of the signal

The shape of a blast wave signal is crucial in analyzing the characteristics and effects of an explosion. A typical blast wave has a distinctive profile: an initial sharp rise to a peak pressure (the shock front), a gradual decrease, and a negative pressure phase (suction). This profile reveals the intensity and duration of the explosion, providing insights into the explosive's strength and the amount of energy released. Understanding the shape of the blast wave aids in predicting the extent of damage it can cause to structures and human safety. It also assists in forensic analysis to determine the nature of the explosive and in designing protective measures against such blasts. The sharp peak and subsequent trough of the blast wave's shape are key in assessing its potential impact and devising appropriate responses.

The signal's shape in the app is expressed as Friedlander, the most representative example of the blastwave described above, Wave, which represents a small signal, and none, which represents no signal.

4.2.3.4. The Maximum FFT(Fast Fourier Transform) Amplitude

The analysis of blast waves, mainly through the lens of the maximum FFT (Fast Fourier Transform) amplitude, is essential for several key reasons. Firstly, the FFT process converts time-domain data from blast waves into their frequency components. This transformation reveals the most dominant frequency within the blast wave, indicated by the maximum FFT amplitude. Understanding the blast's characteristics hinges on this dominant frequency. It tells us a lot about the explosion's energy and its intensity. When we see a higher amplitude at a certain frequency, it means much of the blast wave's energy was packed into that frequency.

Furthermore, the nature of the blast source can often be inferred from these frequency signatures. Different types of explosives and detonation mechanisms produce

distinct frequency patterns, and by analyzing the maximum FFT amplitude, experts can gain insights into the type of explosive material used and the nature of the explosion itself. This information is crucial not only for forensic analysis but also for improving safety measures against future incidents. Additionally, in scenarios involving multiple explosions, comparing the maximum FFT amplitudes provides a basis for understanding each blast's relative intensity and characteristics. This comparative analysis is invaluable in forensic investigations and in developing more effective response strategies and mitigation techniques. In summary, the maximum FFT amplitude is a key indicator in blast wave analysis, providing essential insights into the explosion's characteristics, energy distribution, and potential impacts, which are critical for effective management and preventive measures.

4.2.3.5. The Explosion’s Presence and Duration

Analyzing the presence and duration of a blast signal can provide important information about the characteristics of the explosion and its potential effects on nearby structures and people.

Analyzing the presence and duration of a blast signal can provide important information about the characteristics of the explosion and its potential effects on nearby structures and people. This information can be used to assess potential hazards, design structures that can withstand the blast, and comply with regulatory requirements.

In the Explosion column, the presence or absence of an explosion is indicated by 0 and 1. 0 means that the explosion was not detected, and 1 means that the explosion was detected. If 1 is repeated more than ten times, the app displays the explosion result as YES. Sections that are repeated ten or more times may appear more than once. In that case, the time is calculated based on the section with the most repeated 1s. For example, if a section where 1 is repeated 12 times and where 1 is repeated 28 times exists simultaneously in the CSV file, the app calculates the section where 1 is repeated 28 times and indicates ‘Explosion time.’

4.2.3.6. The Time Range Adjustment

Adjusting the time range of a blast signal can be helpful for several reasons. The time range adjustment involves changing the start and end time of the signal that is being analyzed. Here are some reasons why the time range adjustment is used in analyzing the blast signal:

Removing unwanted data: A blast signal may sometimes contain unwanted data, such as noise or pre-event data. Adjusting the time range allows us to remove this unwanted data and focus only on the portion of the signal that is of interest.

Aligning signals: When analyzing multiple blast signals, aligning them to be compared directly may be necessary. Adjusting the time range of each signal allows us to align them based on specific events, such as the time at which the explosion occurred.

Increasing resolution: By adjusting the time range of a signal, we can increase the resolution of our analysis. We can focus on smaller signal portions and analyze them in more detail.

Extracting specific information: Adjusting the time range of a signal allows us to extract specific information from the signal, such as the peak amplitude or frequency content of a particular section.

Overall, adjusting the time range of a blast signal can provide important benefits in removing unwanted data, aligning signals, increasing resolution, and extracting specific information. This is why it is a common technique used in analyzing blast signals.

4.2.4. Application Activation Screenshot

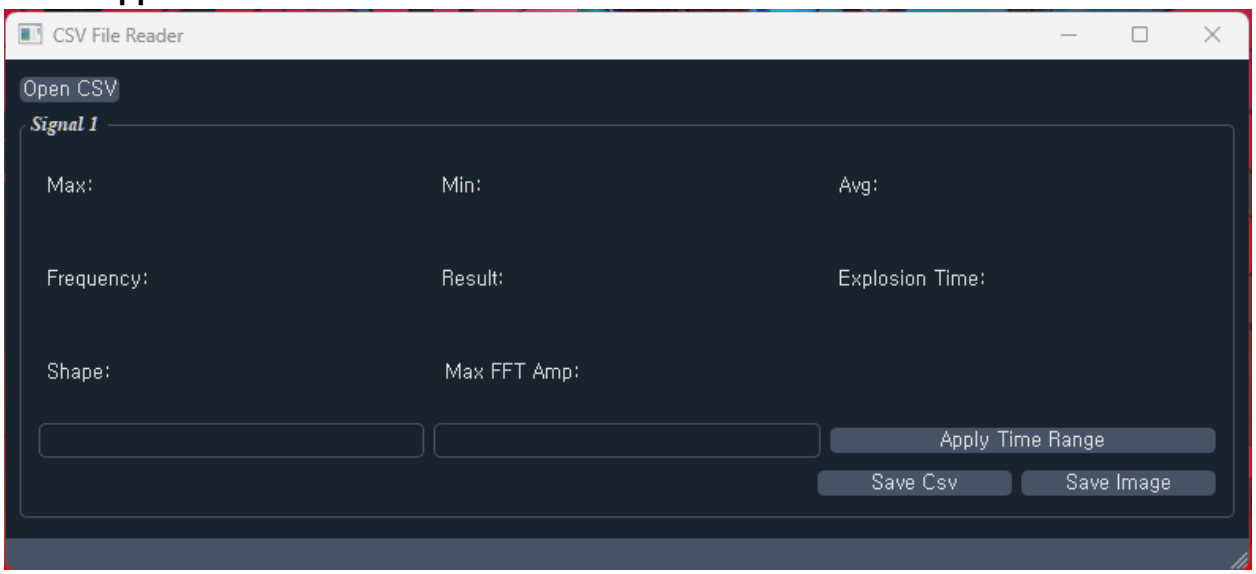


Figure 23: Default window of the application.

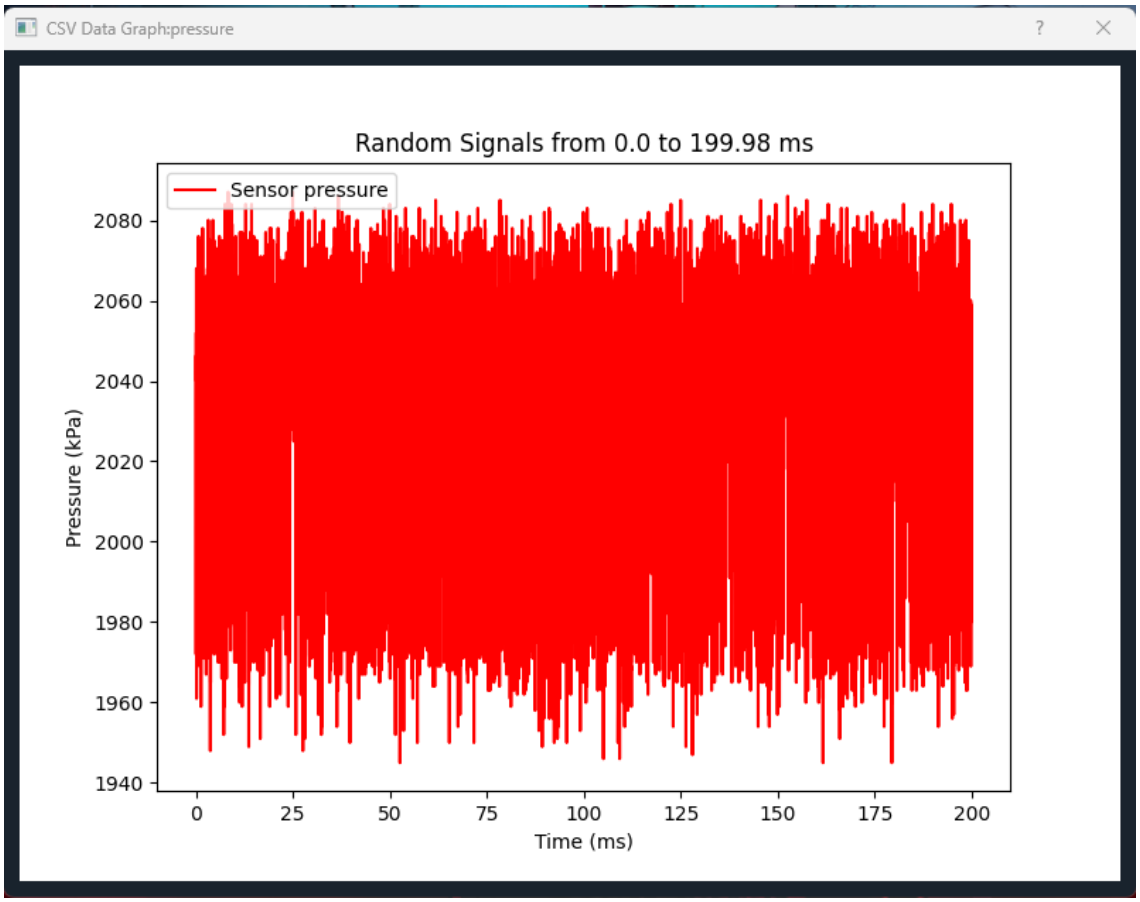


Figure 24: The plot shows the wave from the CSV file

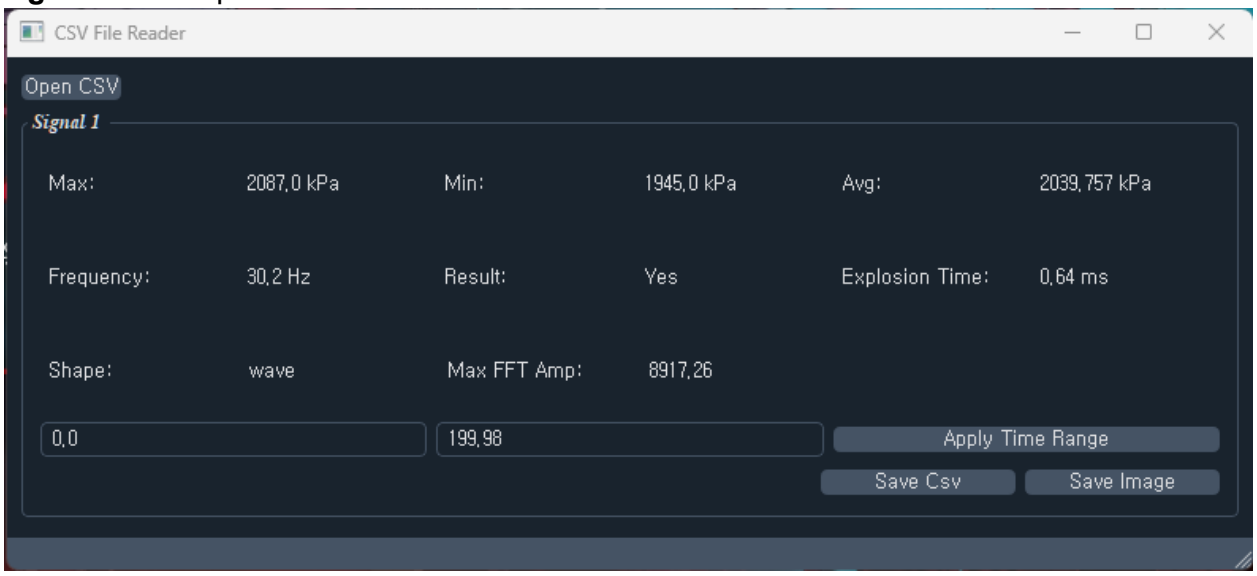


Figure 25: The application shows analysis data from the CSV file

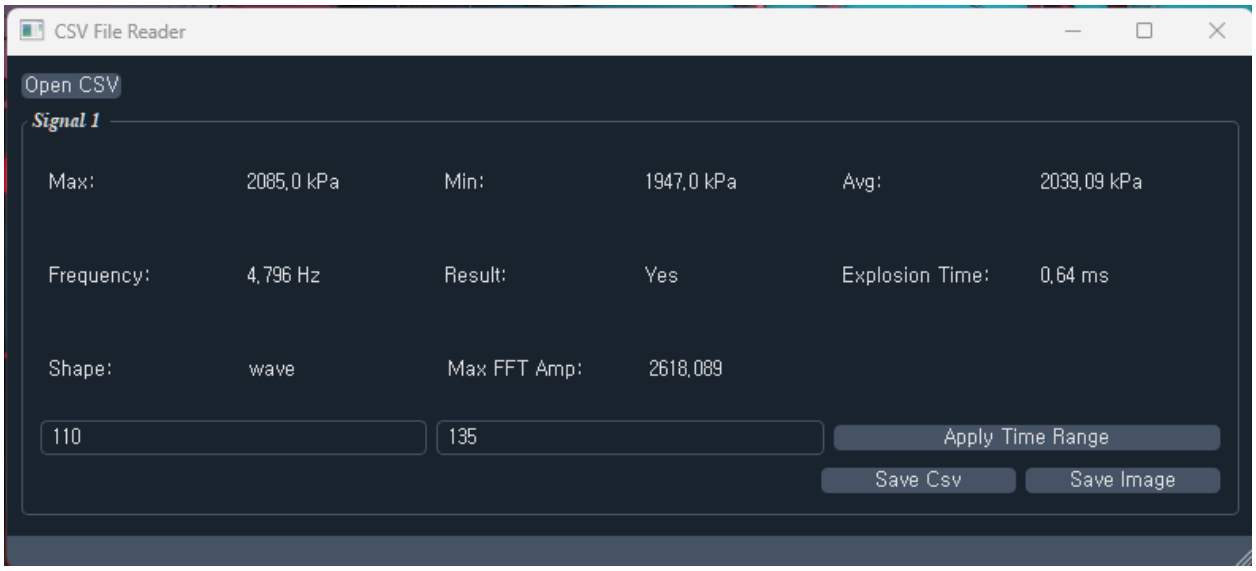


Figure 26: The application shows analysis data from the CSV file after adjusting the time range.

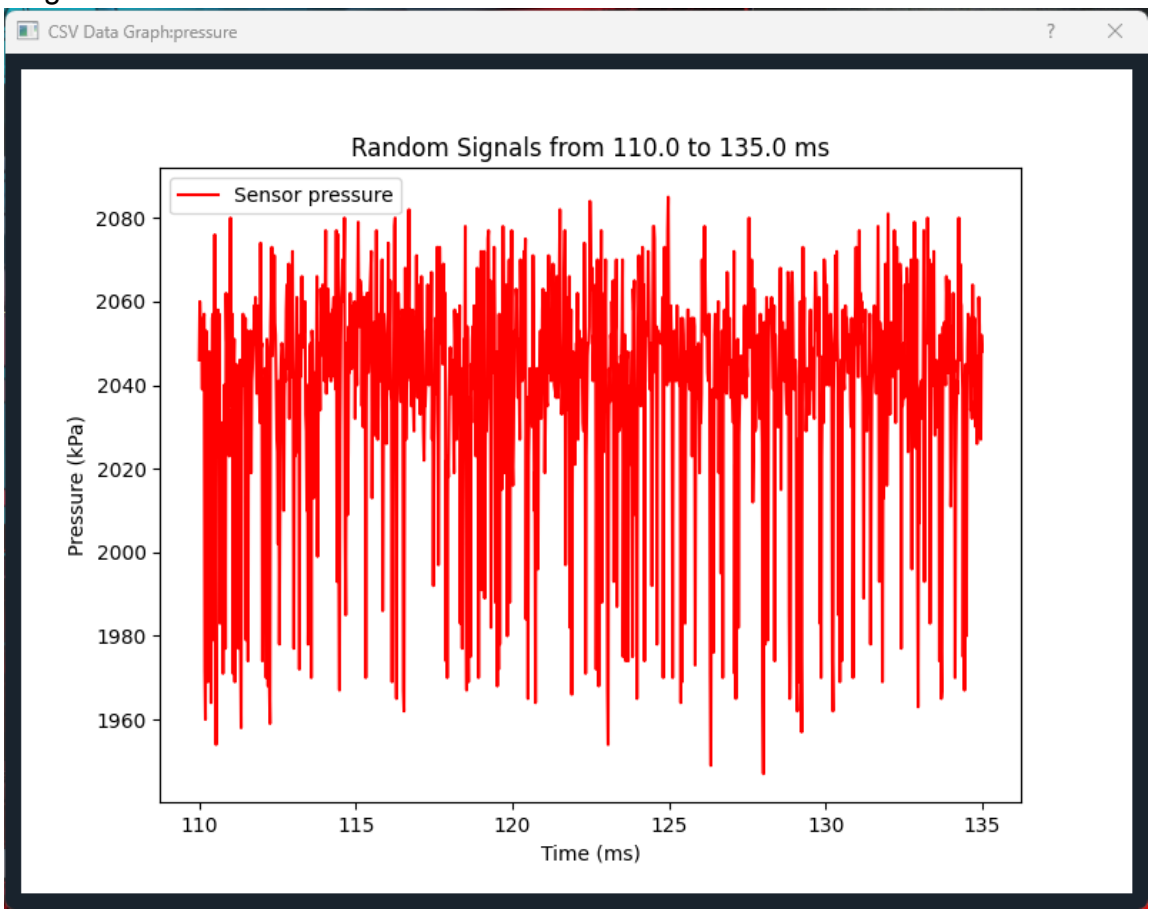


Figure 27: The plot shows the wave from the CSV file after adjusting the time range.

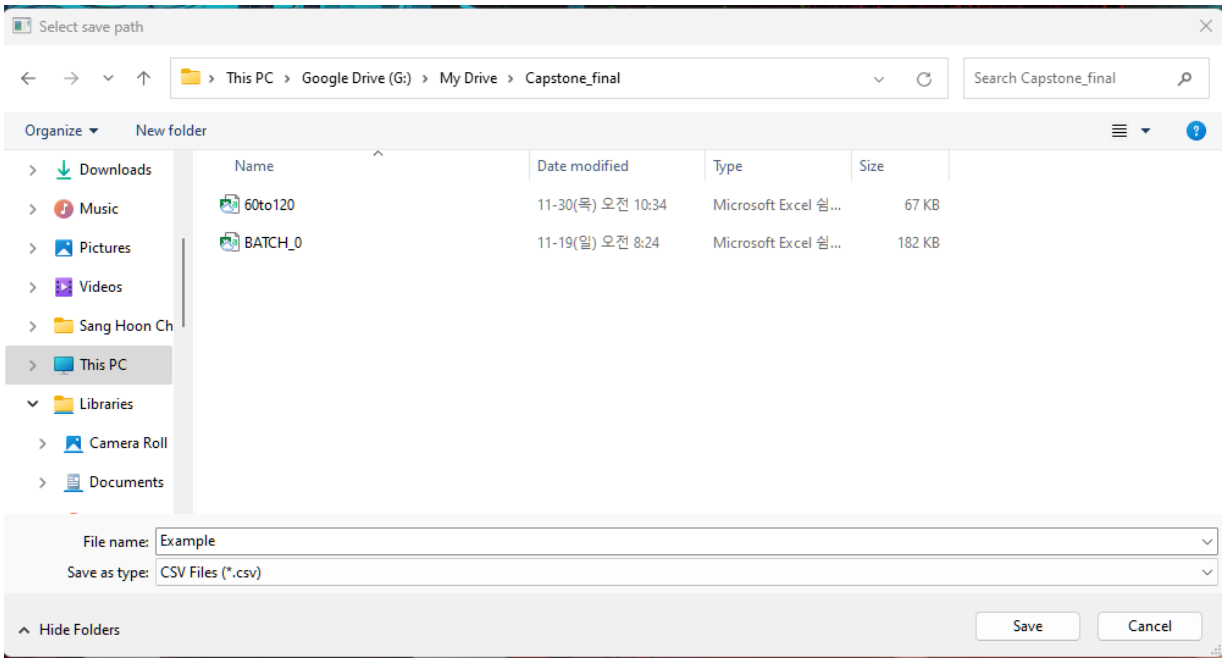


Figure 28: After Click 'Save CSV' button, file explorer directly find the Google Drive and try to save the file.

4.2.5. Subsystem Validation

4.2.5.1. Open inappropriate CSV file

From the output Format(3.3.2), the header of the column name should be Batch, Cluster, Explosion, Pressure, and delta_pressure. If any of these are missing, an error window will pop up and you will have to select them again.

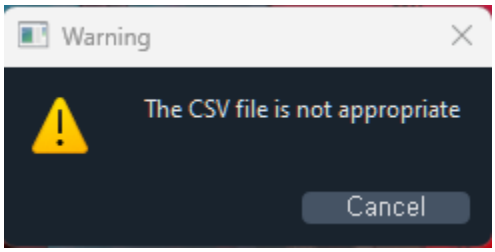


Figure 29: Error window when we open wrong CSV file.

4.2.5.2. Type inappropriate time range

If the time range we entered is inappropriate, the app should display a window indicating that we have entered an incorrect time range.

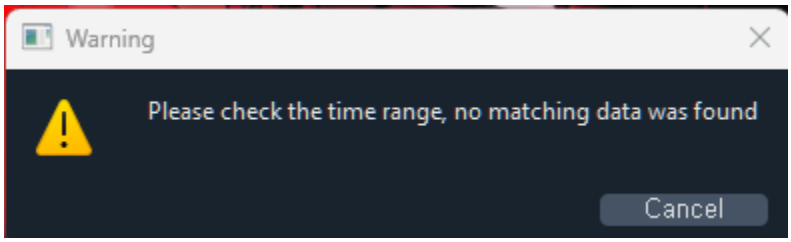


Figure 30: Error window when we type wrong time range.

4.3. Subsystem Validation

Upon launching the application, it successfully connects with Google Cloud Server, ensuring seamless data synchronization and storage. A prerequisite validation step involves downloading and logging into the Google Drive app with a designated account, which we have confirmed works as intended. Following access authentication, we verified the application's capability to read and retrieve files saved on the micro SD card and the Google Cloud server accurately.

The application adeptly reads CSV file data in the core functionality test, converting it into comprehensive plots. We rigorously confirmed that the application correctly displays essential data points, including maximum, minimum, and average values. Additionally, the application effectively computes and presents more complex metrics like the maximum FFT amplitude of the signal. Critical features like identifying the presence and duration of an explosion and delineating the signal's shape and frequency were also successfully validated.

Another key functionality, the ability to input and process specific time ranges, was thoroughly tested. The application demonstrated its proficiency in generating corresponding plots for any given time interval, showcasing its versatility in data analysis. Lastly, we confirmed the smooth operation of the data-saving feature, where users can effortlessly save their data in a desired directory on local and cloud storage. This comprehensive validation process ensures the application is robust, user-friendly, and meets all the outlined functional requirements.

4.4. Subsystem Conclusion

In summary, the application showed commendable performance in basic tasks such as connecting to Google Cloud Server, reading data from micro SD cards and cloud storage, and accurately displaying this data. By reading data, the app can display many metrics based on the data and a plot to increase intuition about the data. The app can save data to a cloud server or external storage.

Portable High Energy Experiment DAQ

Ethan Barnes
Sang Hoon Chung
John Sabra

INTEGRATED SYSTEM REPORT

REVISION – 1
3 December 2023

Integrated Systems Report
for
Portable High Energy Experiment DAQ

Team 34

Approved by:

_____
Project Leader Date

_____
Prof. Kalafatis Date

_____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	12/03/2023	Ethan Barnes		Final Report

Table of Contents

List of Tables

List of Figures

References

[1] V. Muthukrishnan, “Cutoff Frequency: What is it? Equation & How To Find it | Electrical4U,” *https://www.electrical4u.com/*, Nov. 19, 2022. https://www.electrical4u.com/cutoff-frequency/

[2] Here

[3] Here

1. Introduction

The Portable High Energy Experiment Data Acquisition System (PHEE DAQ) gathers real time data using a pressure sensor. It will subsequently analyze the data gathered by the sensor within a processor to identify the occurrence of an explosive event. The PHEE DAQ will be able to detect an explosion within 100ft by utilizing the overpressure data from the initial shockwave of a high energy event. Since this device is expected to gather data continuously, the design will be self-powered with a chargeable battery. To accommodate real time data acquisition, the microcontroller will capture the rise of the event with sampling at fast rates (2 kHz or higher). The output of the microcontroller will identify whether an explosion has occurred and will be stored onto onboard storage. After the pressure data is stored onto the DAQ's onboard storage, It will be opened to be viewed conveniently within an application that uses interactive data visualization via waveform plots and time range inspection. Additionally, the android application can upload the pressure data to a centralized cloud database to keep records of data gathered by the system.

2. Portable High Energy Experiment (PHEE) DAQ

2.1. System Overview

The Portable High Energy Experiment Data Acquisition System (PHEE DAQ) is a high speed sensing system that uses an explosion-grade pressure sensor to identify high energy events within an area. The PHEE DAQ's function exists mainly in the area of defense and military applications, especially in environments where equipment and personnel must be protected. The PHEE DAQ consists of three main components: a pressure sensor to detect shockwave overpressure, a PCB which includes an MCU and DC power supply, and a user-friendly application to conveniently inspect pressure data gathered by the DAQ as shown in figure 1.The Application processes data collected on the MCU to provide systematic data interaction and interpretation. It enables the free movement of data through smooth communication between cloud and external storage. It interprets the data contained in it to provide necessary data when an explosion occurs and provides a plot simultaneously, intuitively presenting the phenomenon as a plot.

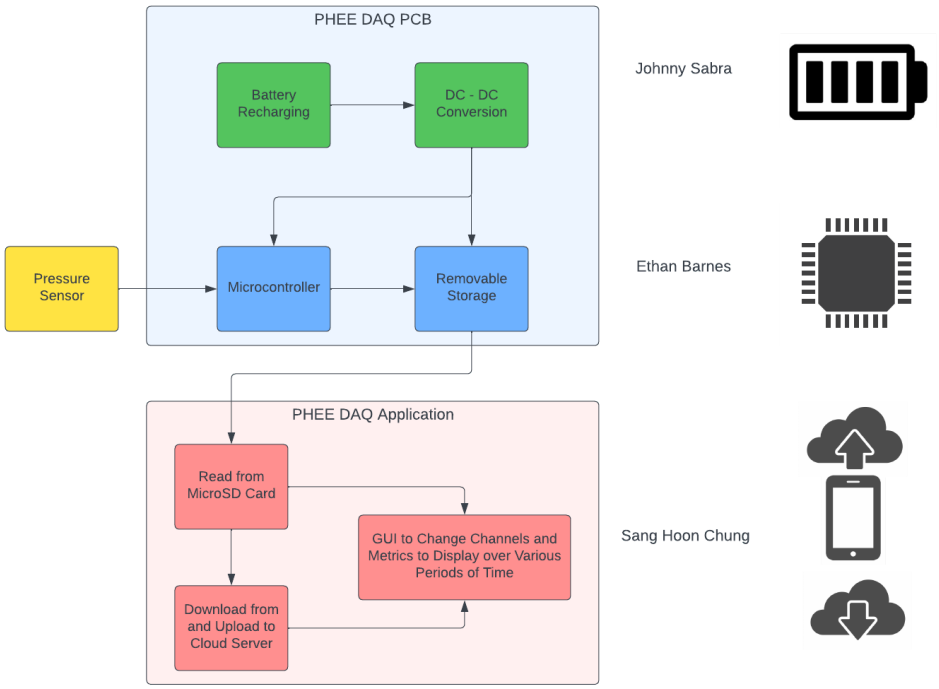


Figure 1: Functional Block Diagram of the PHEE DAQ

Figure 2 displays the layout of the main PCB which features two of the three subsystems of the PHEE DAQ. The pressure sensor is connected to the PCB via a BNC connector where the signal is measured as a voltage where every kilo pascal measured corresponds to a 14 mV voltage signal. For the design purposes of the DAQ, the maximum voltage expected to be read from the pressure is around 2.19 V which corresponds to 150 kPa overpressure. It was determined that 150 kPa is the highest expected overpressure to be captured from an explosion of the magnitude we are considering. Next, the analog pressure data is passed

through a low pass filter to filter high-frequency noise, then passed through an op-amp to limit the voltage output of the signal to be no greater than VDD (3.3 V) so as to mitigate potential damage done to the MCU. The ADC on the MCU collects the data from the pressure sensor and gathers the data into a data buffer. The data from this buffer is analyzed to determine difference in pressure between two samples and then determine based on this difference if an explosive event occurred. This decision along with the actual pressure reading and difference in pressure between a sample and the sample before it are written to a CSV file on a microSD card. The microSD card can then be removed from the device when data is done being captured and can be inserted into a PC or laptop where the Application subsystem will read the data in the CSV files on the micoSD card. From here, the user can use the application to plot the pressure data, compute various metrics on the data, or perform an FFT. The user can also change the time range of the data they want to view to show a plot with more useful information.

Figure 2 also features the DAQ's power supply which utilizes a DC-DC buck converter to supply the required 3.3 V operating voltage of the MCU and SD card. The power supply is powered by a 3.7V lithium polymer battery that can be charged via the battery charger IC which takes a DC input from 5 to 16V.

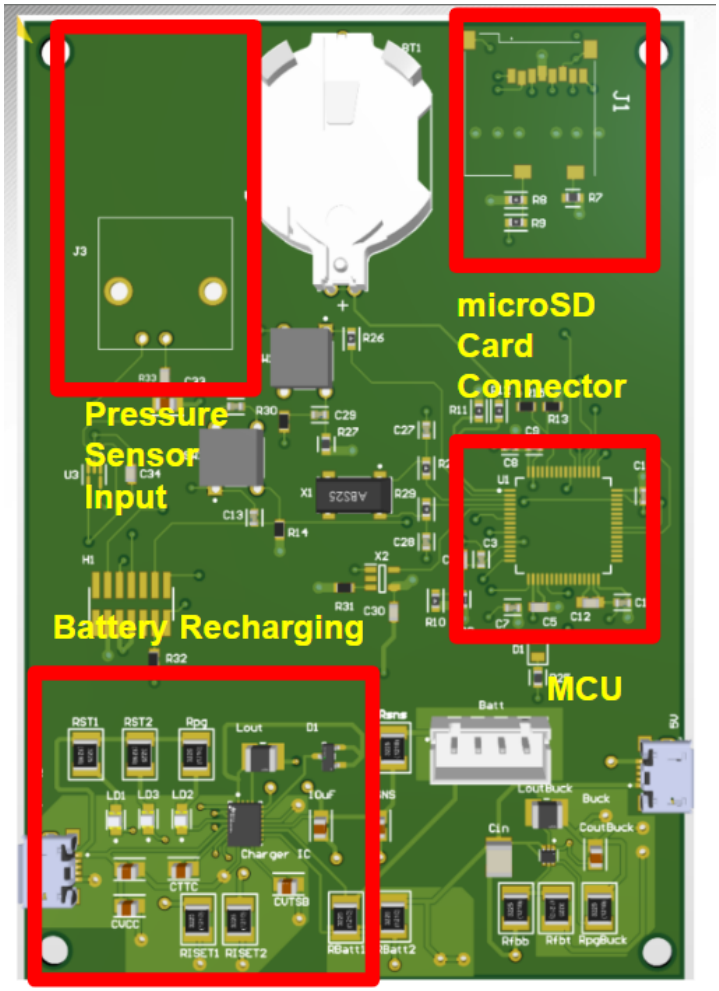


Figure 2: Integrated PCB Layput

Figure 3 is the first screen that appears when the application starts to run. When trying to click ‘Open CSV’ in the upper left corner, the file explorer opens and the user can select and open the CSV file that they want to read. Below the button above, eight useful metrics are displayed when analyzing blast waves. When the user opens a CSV file, the app automatically calculates it. There are three buttons below the Metrics section. If the user clicks the ‘Apply Time Range’ button after they put the appropriate time range into the text field, you can view the data for the entered time range. Also, if the user wants to save the newly extracted data, the user can use the two buttons below. When the user clicks the ‘Save CSV’ button, the user can save the specific time-ranged data as the CSV file. The ‘Save image’ button has a similar function to the ‘Save CSV’ button but it saves a specific time period plot as an image.

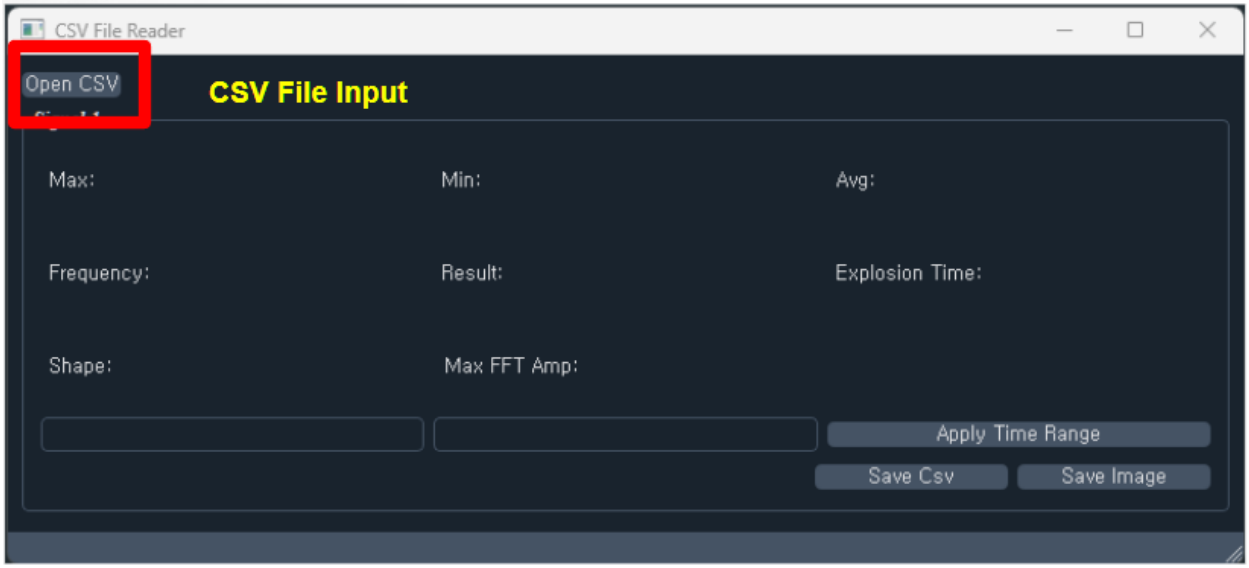


Figure 3: Default screen of the application

2.2. System Details

The PHEE DAQ is powered by a 3.7V rechargeable lithium-polymer battery which falls within the wide input range of the buck converter IC (2.5 -5.5 V). The voltage of the battery is regulated to a required 3.3 V via a voltage divider at the feedback pin of the buck converter. The buck converter uses the constant voltage at the feedback pin as a reference, adjusting the duty cycle of the pulse-width-modulation to achieve the desired 3.3V output. The DAQ PCB consists of 4-layers: the top layer containing the component footprints, a power layer connected to the output of the buck converter, a ground layer, and the bottom layer. This design provided easier routing between the components and a convenient way to access 3.3V or ground anywhere on the board. The power supply also supports DC battery charging via a battery charger IC where a constant voltage at its feedback pin (FB) sets the constant current that it applies to the terminal of the battery to ensure safe constant-current-constant-voltage charging. The voltage divider circuit halves the battery voltage to ensure the constant voltage rate is set well within safe limits for the battery.

Figure 4 captures the entire PHEE DAQ system where the pressure sensor, powered by an external current source, is placed in an area for gathering pressure data while connected to the main PCB via a BNC connector.

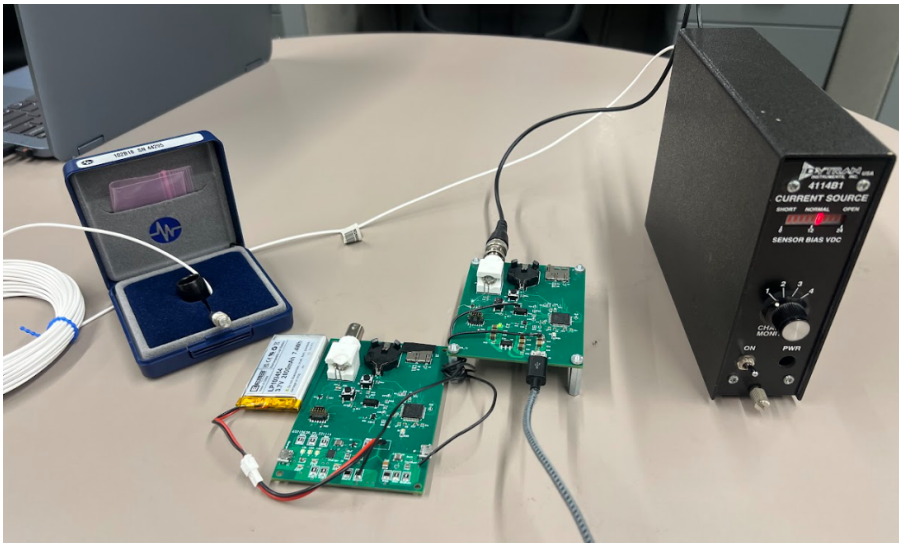


Figure 4: PHEE DAQ System

To flash the MCU so that the program can be run, the user must connect an ST-LINK/V3 using the 14-pin header on the PCB (H1 in Figure 5) while the board is powered on and open open STM32CubeIDE with the main file that the user desires. The user must then select the option to flash the MCU, and once the console confirms the STM32 connection and debugger connection, the MCU will be flashed with the program.

When the board is powered on and flashed, the MCU can begin the data collection process after the reset button is pressed (SW2 in Figure 5). This button will be pressed when the SD card is empty and reinserted into the SD header to begin a new experiment. When the code

begins running, the ADC on the MCU will read data from the pressure sensor. The data from the pressure sensor, however, is first conditioned.

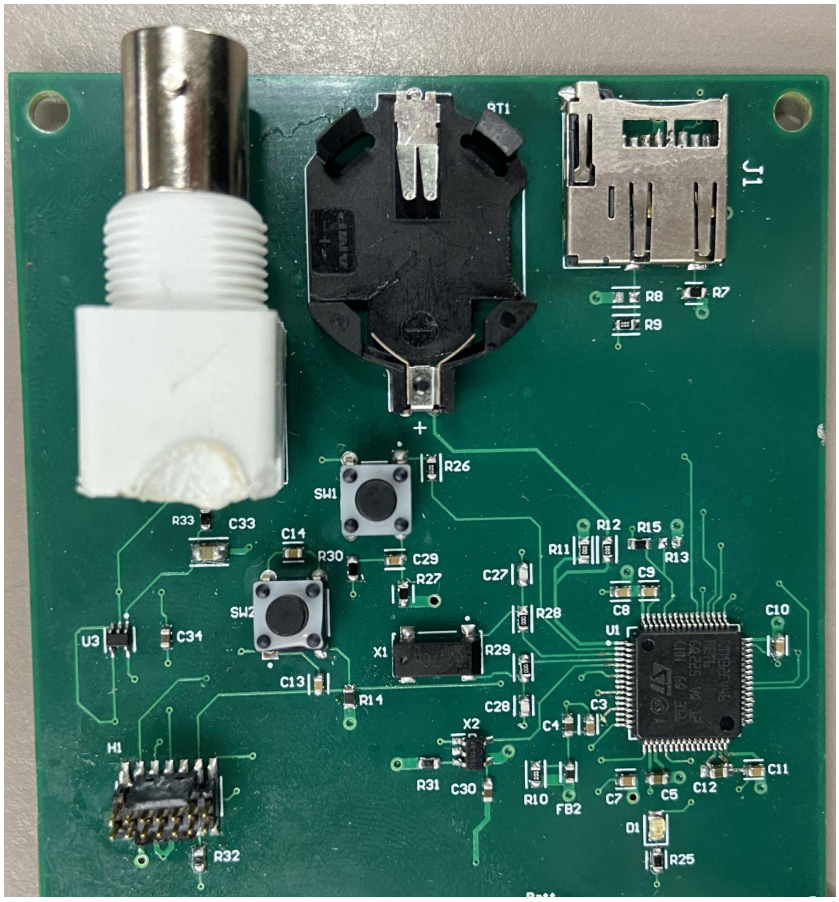


Figure 5: Microcontroller PCB

A small conditioning circuit at the output of the pressure sensor and the input of the ADC uses a low-pass filter and an op amp to reduce noise and limit the voltage input to the ADC pin. The low pass filter is designed to only pass signals with a frequency content below the designated cutoff frequency, which in this case is 1.007 GHz, the expected sampling frequency of the ADC. The cutoff frequency is calculated using the equation below, where R is resistance and C is capacitance:

$$f_c = \frac{1}{2\pi RC}$$

Figure 6: Cutoff Frequency Equation [1]

The filtered data is then passed into an op amp which is just a noninverting op amp without gain, also known as a voltage follower. This op amp is used to limit the max voltage input for the ADC. Op amps saturate at their rail voltages (positive and negative inputs), so we powered the positive supply rail with 3.3V to ensure that if the voltage output of the pressure sensor is above 3.3V, the op amp will saturate and only output a maximum of 3.3V.

Once the signal is conditioned, the 12-bit ADC reads the data and outputs a raw value in binary from 0 to 2^{12}. This value is converted to voltage and then kPa through a conversion provided by the manufacturer of the pressure sensor. The data is then analyzed using the methods described in the Subsystem Report, and the data is saved to the microSD card on-board using SDIO for communication.

Data collection ends when the user presses the stop program button (SW1 in Figure 5). This causes the program to finish writing the CSV file of data to the SD card that it is currently filling up and then enter an infinite idle loop where nothing occurs. The DAQ will not collect data again until the SD card is removed, completely cleared, inserted back into the DAQ, and the reset button is pressed.

Shown in figure 3, there are eight metrics for analyzing the blastwave. The maximum point helps determine the explosion’s magnitude, indicating potential damage, while the minimum point reflects the ground motion generated, which is crucial for assessing nearby structures’ safety. The average point is used to gauge the signal’s attenuation informing us about the blast’s impact over varying distances. Additionally, analyzing the frequency value through FFT (Fast Fourier Transform) is vital in identifying the explosive material used, its energy content, and potential structural damage. This analysis also plays a significant role in forensic investigations and in developing early warning systems. The shape of the blast wave signal, with its characteristic sharp rise and gradual decrease, provides insights into the explosion's intensity and energy. This understanding is critical in predicting damage to structures and human safety and designing protective measures. The application also intelligently analyzes the presence and duration of the blast, aiding in hazard assessment and regulatory compliance.

The application’s ability to adjust the time range of the signal enhances the focus on relevant data, increases analysis resolution, and extracts specific information, which is crucial for thorough understanding and comparison of blast events.

2.3. System Validation

The PHEE DAQ was validated by tapping on the sensing end of the pressure sensor in an attempt to simulate the sharp, spiked overshock pressure waves expected from a high energy explosive event. During the execution of the detection algorithm by the MCU the power supplies output voltage was measured at 3.317 V which is within the operating range of the MCU. Furthermore, the PHEE DAQ's power usage for increasing current loads was measured where the output voltage remained at the required 3.3 V as shown in figure 6.

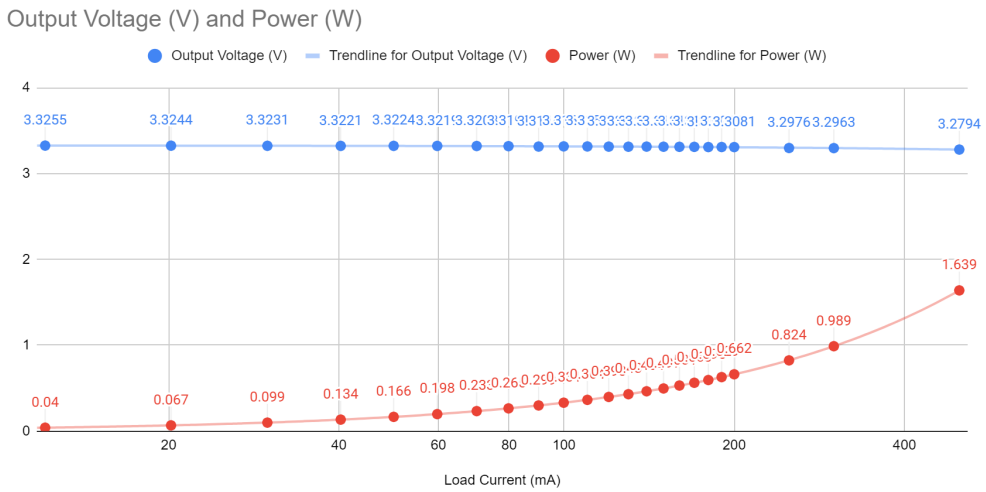


Figure 7: DAQ Power Supply Load Regulation and Power Usage

Table 1 summarizes the power supply validation showing the average voltage under load, power consumption, and voltage ripple.

Average Output Voltage	**3.306 V**
Output Voltage Under Load	**3.318 V**
MCU Measured Nominal Voltage	**3.3178 V**
Average Power	**1.072 W**
Avg Voltage Ripple	**0.017 V under load**

Table 1: Power Supply Validation

To validate the integrated PCB, the MCU subsystem of the PCB needed to be validated. The table below shows the components on the PCB and how we validated their functionality:

Component	Functionality	Validation Test
Reset Program Button	Start program up from beginning	Before this button is pressed, the SD card is empty After this button is pressed, the SD card is full with as much data as was collected until the stop program button is pressed
Stop Program Button	Finish saving current CSV file of pressure data and enter infinite loop	Pressing the stop button on the PCB causes the data collection process to stop When the SD card is removed from the SD header, the data collected does not surpass the time at which the button was pressed
SD Card	Data is written to the SD card in the format specified in the program run on the MCU	When the SD card is removed from the device and inserted into a computer, the data appears in the format specified by the program with the expected data from the pressure sensor
Pressure Sensor	Signals from the pressure sensor are read by the MCU's ADC	The data from the pressure sensor can be seen in the CSV files as expected based on readings from an oscilloscope

Table 2: Integrated PCB Functionality

The console output below shows the improved sampling rate of our integrated system. Because obsolete data is no longer written to the SD card, the sampling rate can increase to around 3.75 ksps which is faster than the minimum sampling rate requested by our sponsor. This means that on average, around 3,750 pressure samples are being written to the SD card every second. This is calculated by dividing the total number of samples written to the SD card when the program finishes running by the total runtime of the ADC collecting data.

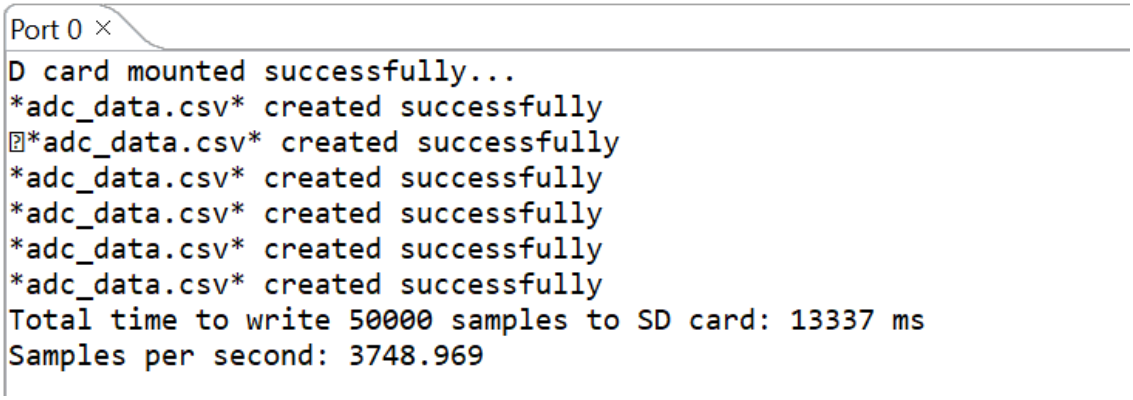


Figure 8: ADC Sampling Rate

A sample output file of the device is shown below. The file captures the batch (nth file of 10,000 samples), the cluster (used to estimate time), explosion (bit set to 1 if explosion occurs), pressure (reading from pressure sensor), and delta_pressure (change in pressure from previous sample to current sample). The pressure data shown in the file is gathered from the pressure sensor attached to the integrated PCB. The pressure being read indicates a pressure of around 50 kPa which matches the expected voltage output of the pressure sensor at 0.877 V after the signal is conditioned.

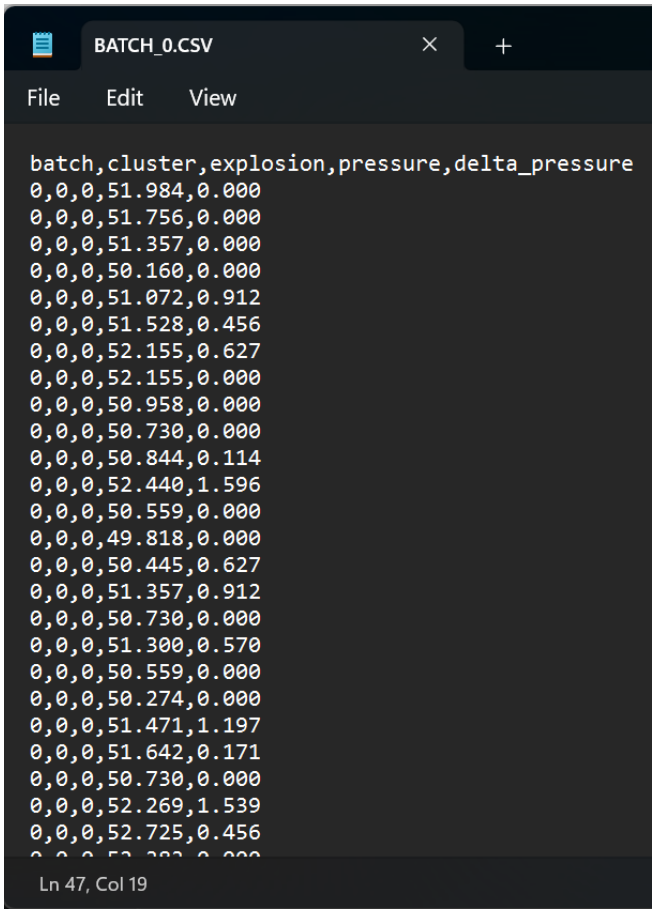


Figure 9: Sample File Output

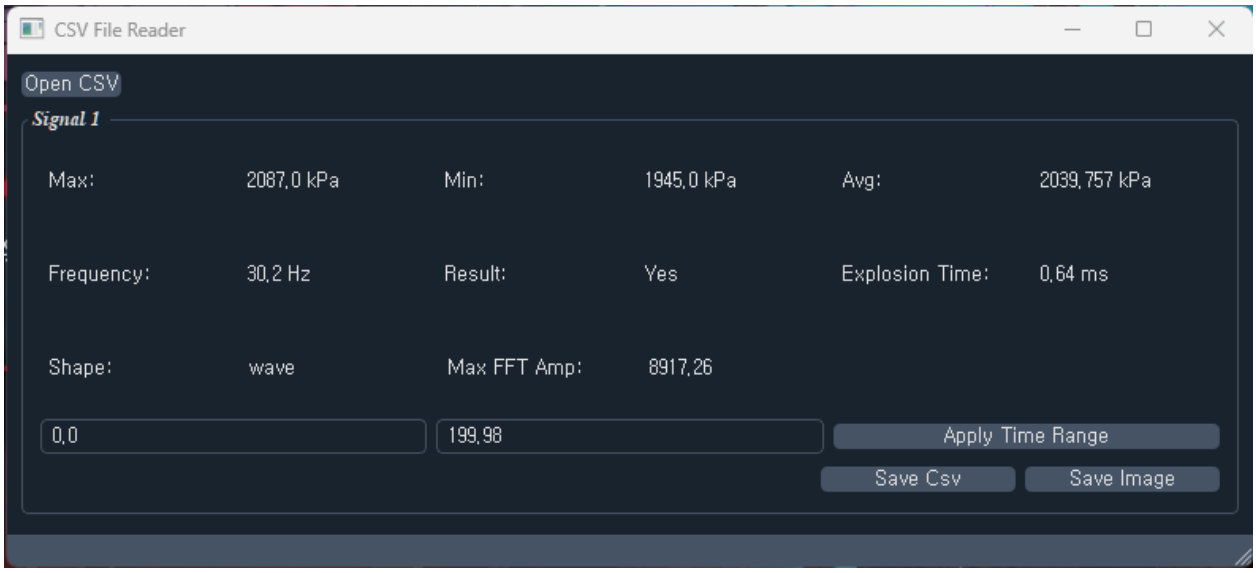


Figure 10: Application Analysis Tools

As depicted in Figure 7, upon opening a CSV data file in the application, it is evident that all eight metrics are automatically calculated. Verification results indicate that these metrics are accurately represented, as there was a precise match between the values displayed in the application and the verified values. It was also observed that the metric values appropriately changed when adjusted to a specific time zone(in this case, set up 110 ms to 135 ms), with these adjustments maintaining accuracy. These modifications are clearly illustrated in Figure 8.

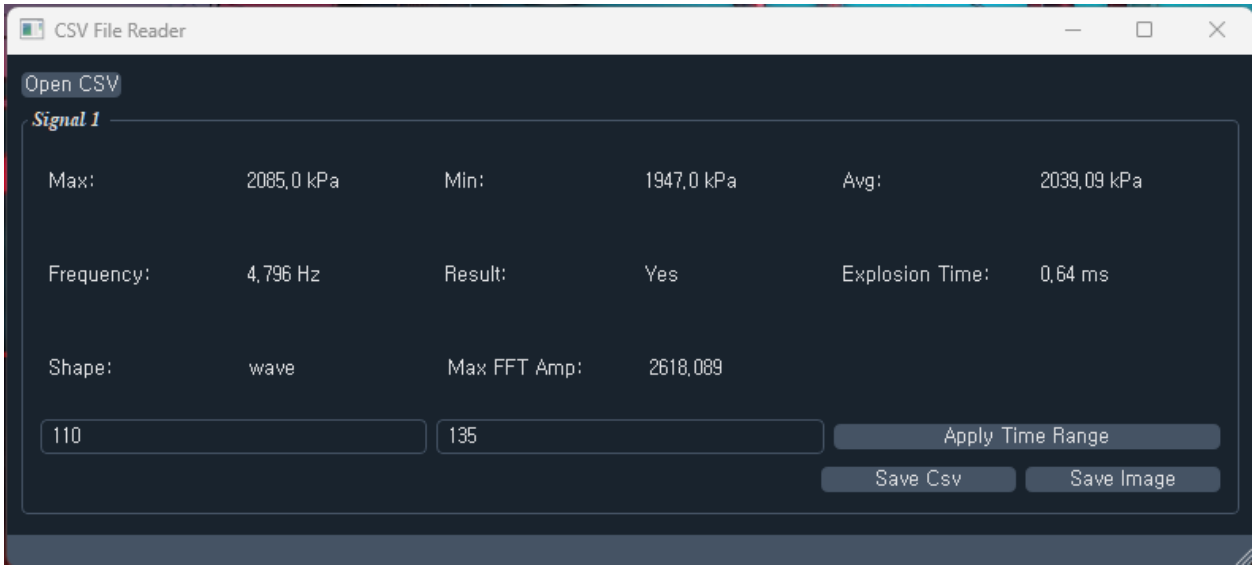


Figure 11: Application Window After Adjusting the Time Range

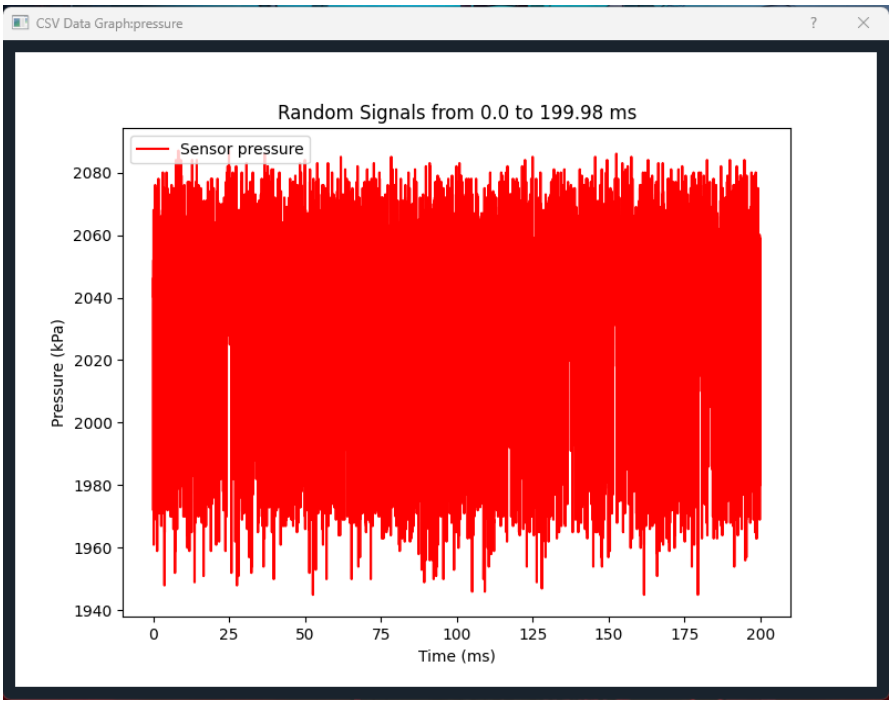


Figure 12: Waveform Derived from CSV File

Further testing was conducted to assess the accuracy of plot displays when opening CSV data files. This involved verifying the correct representation of titles and the x and y axes on the plots. A comparison between the plots generated in the application and those created in Excel showed identical results, as in Figure 9. Additionally, when the data was adjusted to a specific time zone, the application accurately generated and displayed plots corresponding to that time zone, as in Figure 10.

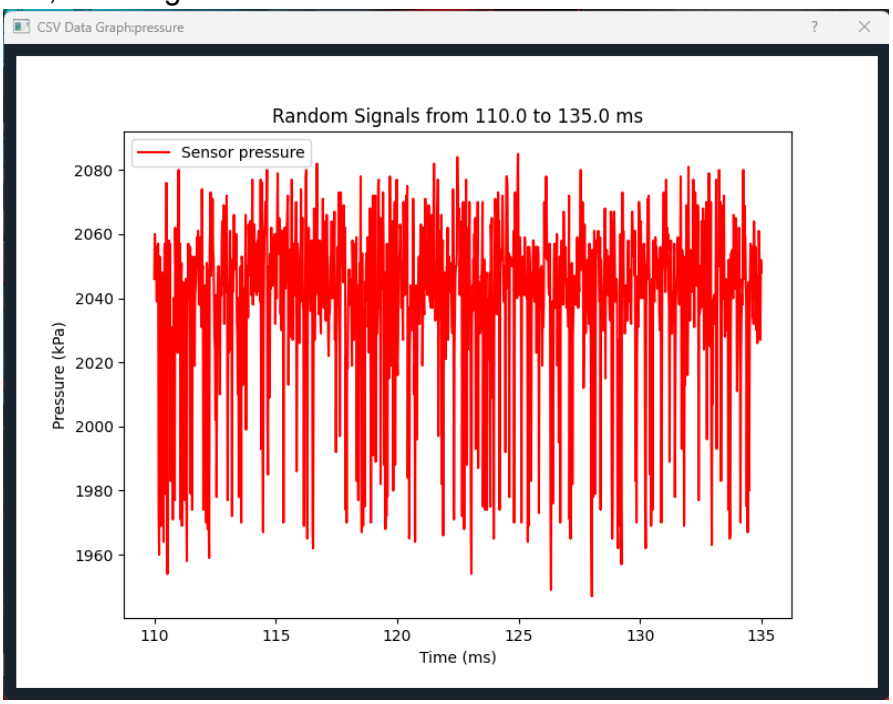


Figure 13:.Pressure Wave Plot After Adjusting Time Range

Through these tests, the functionality of the 'apply time range' button, which facilitates adjustments to specific time intervals, was validated. The efficacy of the 'Save CSV' and 'Save Image' buttons was also tested. Upon activation, File Explorer was prompted to open, allowing for successful saving of CSV data and images (in .jpeg format) to Google Drive. This confirmed the smooth operation of these features.

2.4. System Conclusion

The Portable High Energy Experiment Data Acquisition System (PHEE DAQ) detects high energy explosive events within sensitive areas by measuring overpressure caused by the initial shockwave of such events. Comprised of a pressure sensor to detect shockwave overpressure, a PCB which includes an MCU and DC power supply, and a user-friendly application to conveniently inspect pressure data, the DAQ is a portable system that can be used mainly in the area of defense and military applications where equipment and personnel must be protected.