Power Quality Monitor and Sub Meter System

ECE 445 Design Document - Spring 2024

Team #30
Nicole Viz
(nviz2@illinois.edu)
Soham Manjrekar
(sohammm2@illinois.edu)
Roshan Mahesh
(roshanm2@illinois.edu)

TA: Surya Vasanth
Professor: Jonathon Schuh

February 22, 2024

Abstract

This document provides details, diagrams, and schematics for the design and planned implementation of our project, which is a power quality monitor and sub meter system. Beyond our design and implementation, we will provide information on the parts we will use, the total cost, the requirements for each subsystem along with how we will verify it, a weekly schedule of when we plan parts of the project to be completed, and an analysis of the safety and ethics of our project.

Table of Contents

1 Introduction

In this section, we explain the problem that we are attempting to solve, and we highlight its relevance in the field of power electronics today. We also illustrate our solution to the problem, along with a visual aid and our high-level requirements so that others can understand our planning and design process.

1.1 Problem

In the rapidly evolving field of power electronics and energy technologies, maintaining consistent and high-quality power distribution and energy usage is critical for residential and commercial buildings. Using submeters can create energy savings, lower operating costs, increase building efficiency and reliability, and improve occupant comfort. However, devices today can be cost-inefficient, complex to operate and to read, and they may lack real-time insights. These shortcomings lead to difficulty in meeting recent sustainability efforts, and as such, an innovative solution is needed.

For example, a common residential electrical submetering product is from Byram Labs. It's a single phase submeter that costs $420, doesn't provide prior electrical usage data, doesn't provide a way to perform data analysis, and doesn't monitor power quality or notify owners of poor power quality for the following metrics: large voltage changes/irregularities, power outages, harmonic disturbances. This can cause many problems, especially for large commercial applications such as hospitals. In Japan, there was a study done on 13 pieces of medical equipment and found that voltage dips caused 7 devices to stop for about 0.5 seconds, which caused some of the 7 devices to start an automatic reboot. Without having the electrical submeter also measure for power quality, medical equipment could get damaged or patients could suffer from ineffective medical care.

1.2 Solution

For our project, we'd like to design and construct an improved device that monitors power quality and acts as a submeter to its loads – a device that is cost-effective, has high-fidelity data acquisition, and operates with an intuitive user interface LCD screen. Our project will solve the problems listed above by combining a power quality monitor along with a submeter in a cost-effective manner that stores real-time data and loads the data to a database that can be accessed through a website. The website would also allow users to be alerted of power quality issues, provide access to prior data, and showcase voltage and power waveforms.

Over the course of the build process, we will produce a prototype that analyzes voltage, current, and power from an outlet and examines the power usage of a connected load. To do this, the device will connect into an outlet, step down voltage and current to be sensed proportionally with

ICs, and send measurements through the microcontroller to the database. The control unit of the device will be powered by a converter circuit with a linear regulator, and in backup cases, with a battery subsystem. We will encase all circuitry and hardware within a box that has an LCD display on the front to display data for the load (which during building and testing will likely be in the form of a power resistor). Upon final testing and troubleshooting, we aim to demonstrate the capabilities of this product by using a power outlet at the ECEB and connecting a phone charger as the load.

1.3 Visual Aid

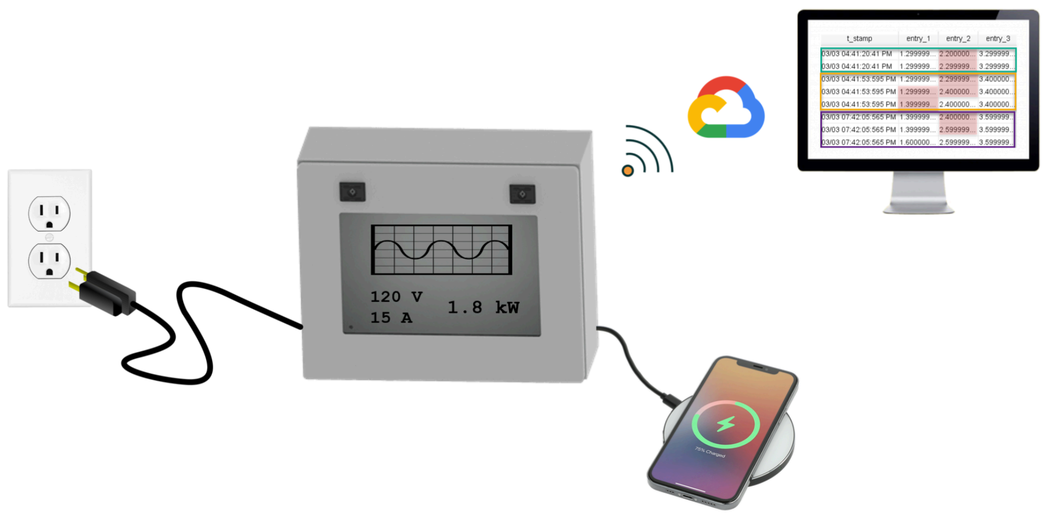

Figure 1: Visual aid of our entire system

1.4 High-Level Requirements

For our project to be considered successful, there are various high-level criteria that must be met. Our device should be able to perform the following tasks:

I. Sample a single-phase input for its voltage and current (with a high level of accuracy, within 10%, which can be checked with a wattmeter).

II. Use a microcontroller to process the voltage and current samples and calculate apparent and real power. Store data points onto an SD card (once every second) and to a cloud database (once every 10 seconds). Instantaneous waveforms will also be displayed on a screen.

III. Notify the user (in a timely fashion, within 5 seconds) of any disturbances in measurements outside of a set tolerance (5%) and of any failures.

More detailed subsystem requirements can be found in Section 2.3. As a stretch goal, we hope to implement measurement and notification of total harmonic distortion (THD).

2 Design

2.1 Block Diagram

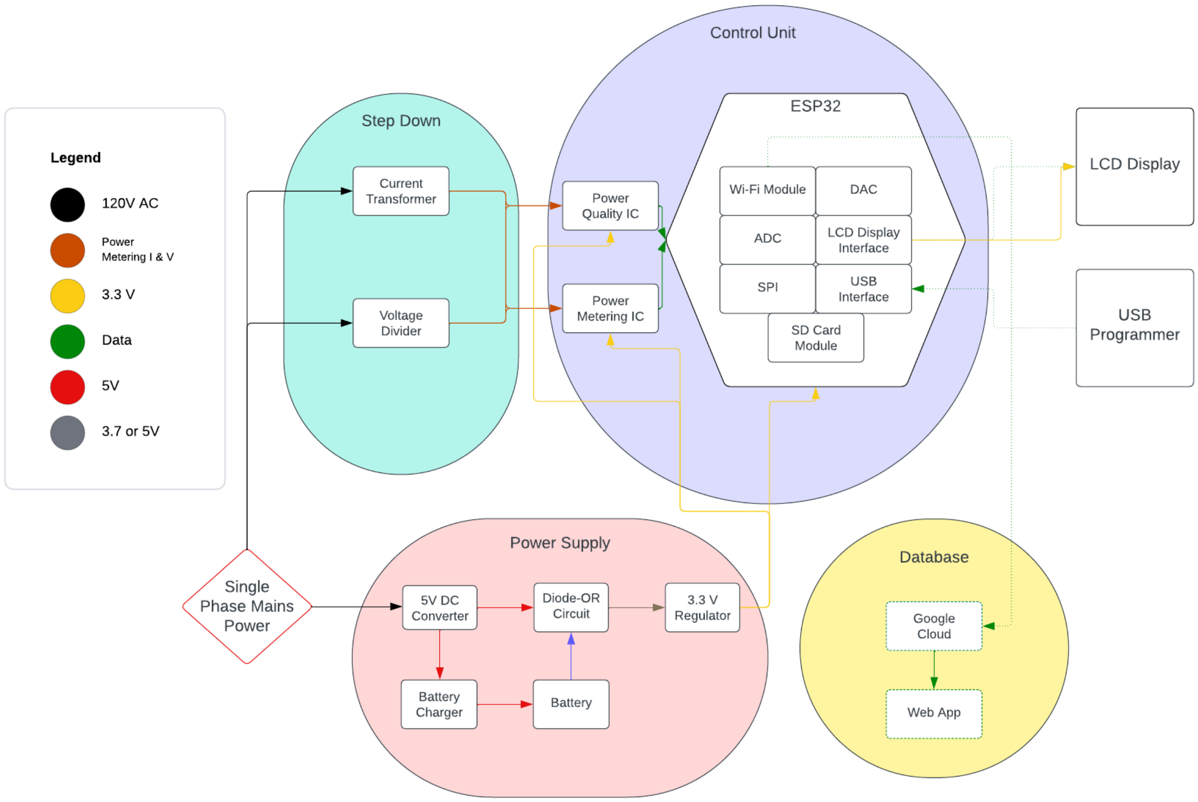

Figure 2: Block diagram

Our design is divided into 4 subsystems: Step Down, Control Unit, Power Supply, and Database. These blocks are implemented with a mixture of hardware and software, with the hardware being physically housed in a finished container to look like a professional product rather than a rapid prototype.

2.2 Physical Design

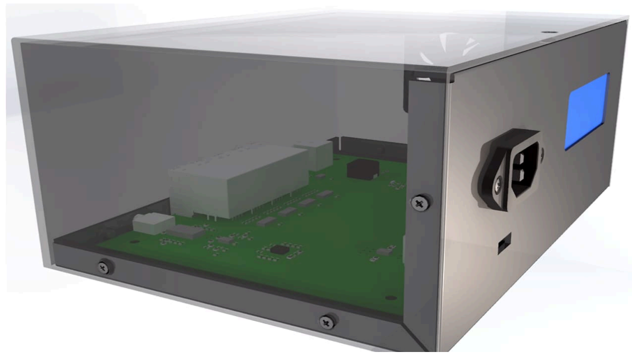

Figure 3: Enclosure Mock Up

Our design will use a metal and plexiglass enclosure, with two outlets (similar to Figure 1) and an LCD display. The plexiglass portion will reveal the PCB and all the electronics so that a user may observe the box during operation and can physically verify the board's operation. The LCD will display pertinent information, and the outlets will be for connecting the wall mains and loads.

2.3 Power Supply Subsystem

2.3.1 Overview

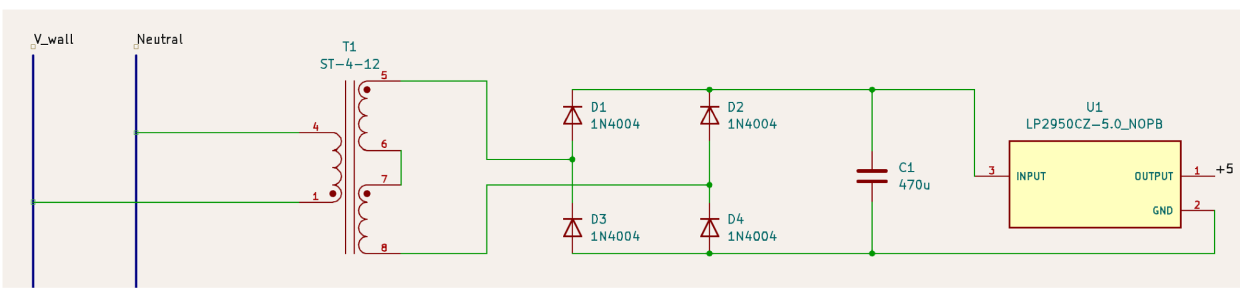

Figure 4: 5 V DC converter

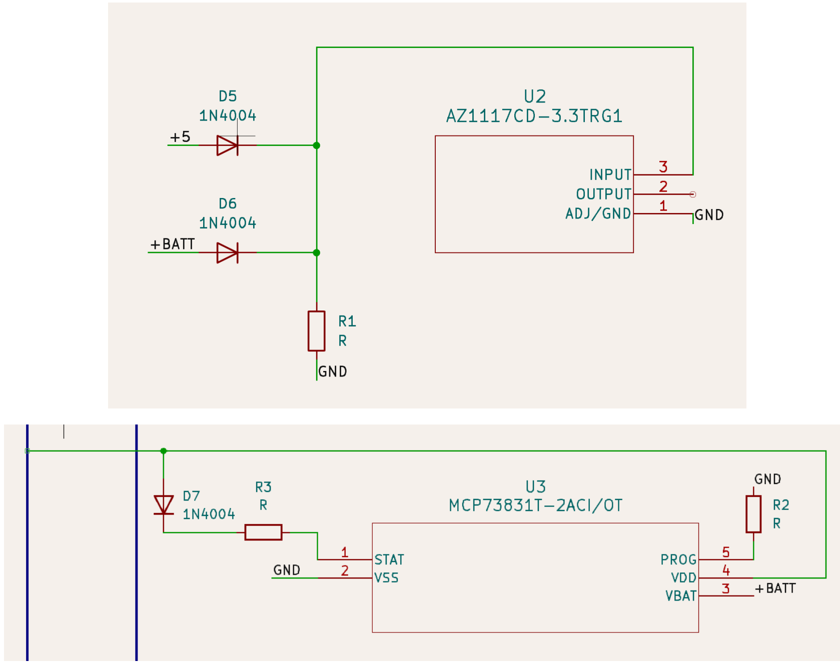

Figure 5: OR circuit, 3.3 V regulator, battery charger circuit

The Power Supply Subsystem takes the AC voltage from a wall outlet which is converted to 5V DC and used in two ways. Firstly, the 5V is regulated to 3.3 V and used to power the control unit. Concurrently, the 5V is sent to a battery charging circuit to charge our LiFePo4 battery. In the event of a blackout where the wall no longer powers the AC-DC converter, the battery shall take over powering the Control Unit via a diode OR circuit. The power supply is constructed with a power transformer (ST-4-12) that steps voltage down from 120V RMS to 12V RMS. The 12V is then rectified with a full wave bridge rectifier, then filtered, and passed through to a 5V linear regulator. The 5V DC is then used to power a battery charging IC to charge an 18650 LiFePo4 Battery rated for a nominal voltage of 3.7V. We have acquired two charging ICs (MCP73831T-2ACI/OT and TP5410). These ICs capabilities are similar, they both have protection built in and allow for pass through voltage when the battery is charging. After the charger and 5V converter there is a Diode-OR circuit that allows power to flow to a 3.3V regulator from the wall mains or the battery if there's a blackout.

2.3.2 Interfaces

- Wall Outlet
    - The Power Supply subsystem is connected directly to the wall as an input and converts it down to 3.3V with intermediate steps at 12V, 5V and 3.7V.
- Control Unit
    - The Power supply delivers 3.3V regulated to the control unit to power the ESP32 microcontroller, and the two measurement ICs.

2.3.3 Requirements/Verification

Requirements	Verification
Switch between wall powered and self-powered modes in the event of a blackout without loss of operation	● Simulate a blackout by removing wall power and validate that 3.3V is still being delivered to some load at the output of the supply
In self-powered mode system should run for >10 hours	● Fully charge battery and run system powering an analog bit adder counter circuit. Verify that the battery operates for more than 10 hours.
Charging circuit protections: charging battery to 80%, at rated current maximum.	● Utilize fuel gauge IC (MAX17048G+T10) to monitor whether charging circuit protections are operating effectively. ● Optionally integrate fuel gauge IC onto PCB to display battery %age.

Table 1: Requirements and Verification page for Power Supply Subsystem

2.3.4 Design Decisions

This section has the most uncertainty with regard to design, we have acquired multiple parts that perform the same function for example two charging ICs, and two linear regulators (AZ1117CD-3.3TRG1, LP2950CZ-5.0/NOPB). Our design is based solely on the performance of the system, and whatever will give us our 3.3V output from the subsystem in conjunction with smooth battery takeover during a blackout. As a result, we have decided to spend more time prototyping our power supply circuit multiple ways. Furthermore, this type of circuit has been done countless times in countless times which allows us to be much less particular about getting the exact components we need, unlike the control unit wherein we were very particular about our components.

2.4 Step Down Subsystem

2.4.1 Overview

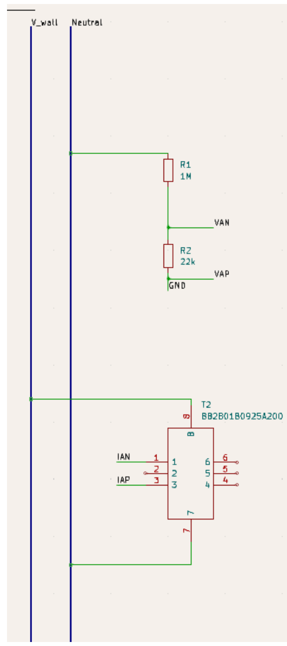

Figure 6: Step Down Block Diagram

The Step Down Subsystem handles converting the voltage and current from the wall down to a value that the metering and monitoring ICs (ADE953 and ADE9430 respectively) can measure. Specifically, there is a Voltage Divider circuit to step the voltage down to <3.4V, and a Current Transformer (B82801B0925A200) to step the current down to <75mA, this was in anticipation that there is a current limit, however the only limit for the IC's is a voltage limit, we may still use this to get an isolated current input to the measurement ICs and use the onboard PGA to get what the current draw is actually. The diagram for this subsystem is shown in the Control Unit schematic (2.5.1) as they directly feed into the measurement ICs.

2.4.2 Interfaces

- Wall Outlet
  - Direct input 120V RMS from the outlet to the subsystem to be stepped down
- Control Unit Subsystem
  - Directly to VAN-VAP and IAN-IAP
  - Should deliver less than 3.4V.

2.4.3 Requirements/Verification

Requirements	Verification

Step Down Voltage from 120V RMS to <3.4V since the ICs cannot handle a voltage higher than that	• Send an AC waveform smaller than the wall voltage through the divider with some load across the resistor and observe that it steps down to 3.4V ±2%
Step Down Current from <15A to ~<75mA	• Send an AC waveform smaller than the wall voltage through the transformer with max load of 15A across the secondary and observe that the current steps down to 75mA ±5%

Table 2: Requirements and Verifications table for Step Down Subsystem

2.4.4 Design Decisions

This decision came with the goal of ease of flexibility. At the time of designing, we were unfamiliar with the measurement ICs and wanted to remain as flexible as possible. We weren't worried so much about sophisticated solutions, we simply wanted a solution that we could replace if needed later down the road in case other complications arose.

2.5 Control Unit Subsystem

2.5.1 Overview

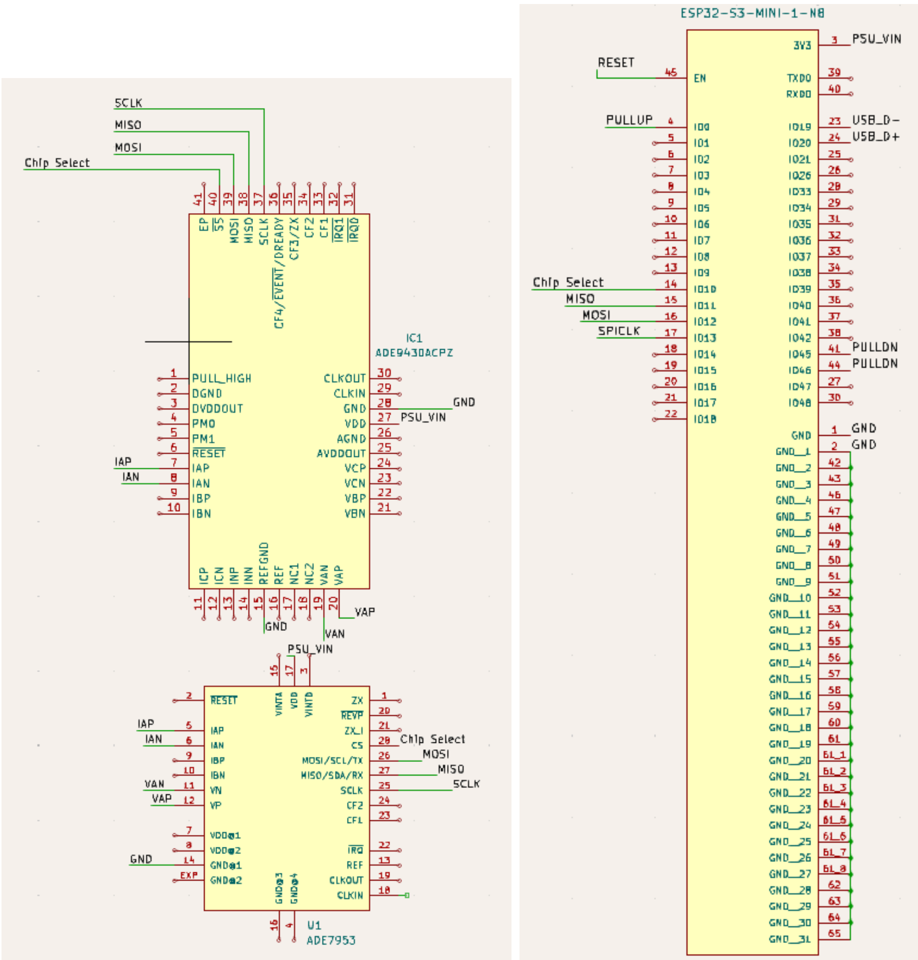

Figure 7: Control Unit Block Diagram

The Control Unit subsystem is constructed of the two measurement ICs (ADE953 and ADE9430) and the ESP32 Microcontroller (ESP32-S3-MINI-1-N8). The ICs are where the voltage and current signals are converted into digital signals. Stepped down voltages are delivered to both measurement ICs which then pass through a PGA and an ADC followed by a digital block where all pertinent DSP calculations are made. Data then will be passed through to the Microcontroller via SPI protocols to be outputted to a database. We will leverage the SPI interface on both the measurement ICs using the MISO pins as an output to stream data out from the ICs, with SCLK and MOSI and CS (Chip Select) being inputs. On the ESP32 these are all GPIO pins specifically pins 10-13, we will use these to generate a clk signal (SCLK), stream data from the ESP32 to the ICs (MOSI), stream data from the ICs (MISO), and enable/disable communication between the two sections of our subsystem. Once on the ESP32, the data will be stored locally and displayed on an LCD screen powered by the ESP32, with data sent via the I2C communication protocol. The data will also be sent via Wi-Fi and HTTP connection to the

Database subsystem (Section 2.6) for processing and delivery to a web app. The ESP32 itself will be powered from the Power Supply subsystem (Section 2.3) at 3.3V depicted as PSU_VIN in Figure 6.

2.5.2 Interfaces

- Step Down Subsystem
  - Voltage and Current Signals are taken in from the Step Down subsystem
- SPI
  - Data is passed through via SPI from the ICs to the ESP32
- Power Supply Subsystem
  - The 3.3V that powers the ESP 32 and measurement ICs comes from the Power Supply subsystem
- LCD Display
  - The power and data signals for the external LCD Display come directly from the main 3.3V pin on the ESP 32.
- Database Subsystem
  - Data is passed to the Database subsystem over Wi-Fi and HTTP connection

2.5.3 Requirements/Verification

Requirements	Verification
Original Voltages and Currents (pre Step Down) should be sent to the ESP32 from the measurement ICs and be accurate to ±5%	● Send a small AC waveform directly to the ICs and record the data directly to On-Chip Memory as a calibration test. ● Give 3.3V to VDD and Vin for the ICs and ESP32 respectively and use the wall as a source testing the Step down and Control unit subsystem at the same time, under no load and low load condition
System is not bottlenecked by recording speed	● Let system run for 1 minute and verify the number of data points collected, exceeds the number of data points being sent per minute by

Table 3: Requirements and Verification table for Control Unit Subsystem

2.5.4 Design Decisions

There are many ways to measure current and voltage from a source and to a load, in our case, we chose a more expensive and complicated system. We chose this to understand the potential capabilities of newer IC technology, we hope to extract much more data from these two measurement ICs than traditional sensors or circuits. Additionally, our project scope did not have a budget constraint, as a result we want to develop a device that meets industry level Submetering and Monitoring requirements and we believe these ICs will allow us to do just that all while keeping our final device packaged in a sophisticated manner.

2.6 Database Subsystem

2.6.1 Overview

The database subsystem is a full-stack web application that displays graph data, specifically the voltage and current of the power outlet and the load. The web app also notifies the user when there are disturbances or failures with the power outlet voltage or current in real-time so that a building manager or electrician can fix the problem as early as possible. The backend is written in JavaScript using Node.js and Express.js to create a server and send HTTP requests to and from the frontend and the database. The frontend is written in JavaScript as well by using React.js to create an interactive web application. To create real-time graphs, we will use websockets and the Chart.js library, which allows us to read and plot data. The frontend design, which is created using Figma, is pictured below. This is just a visual representation, not exactly what the end version will look like.

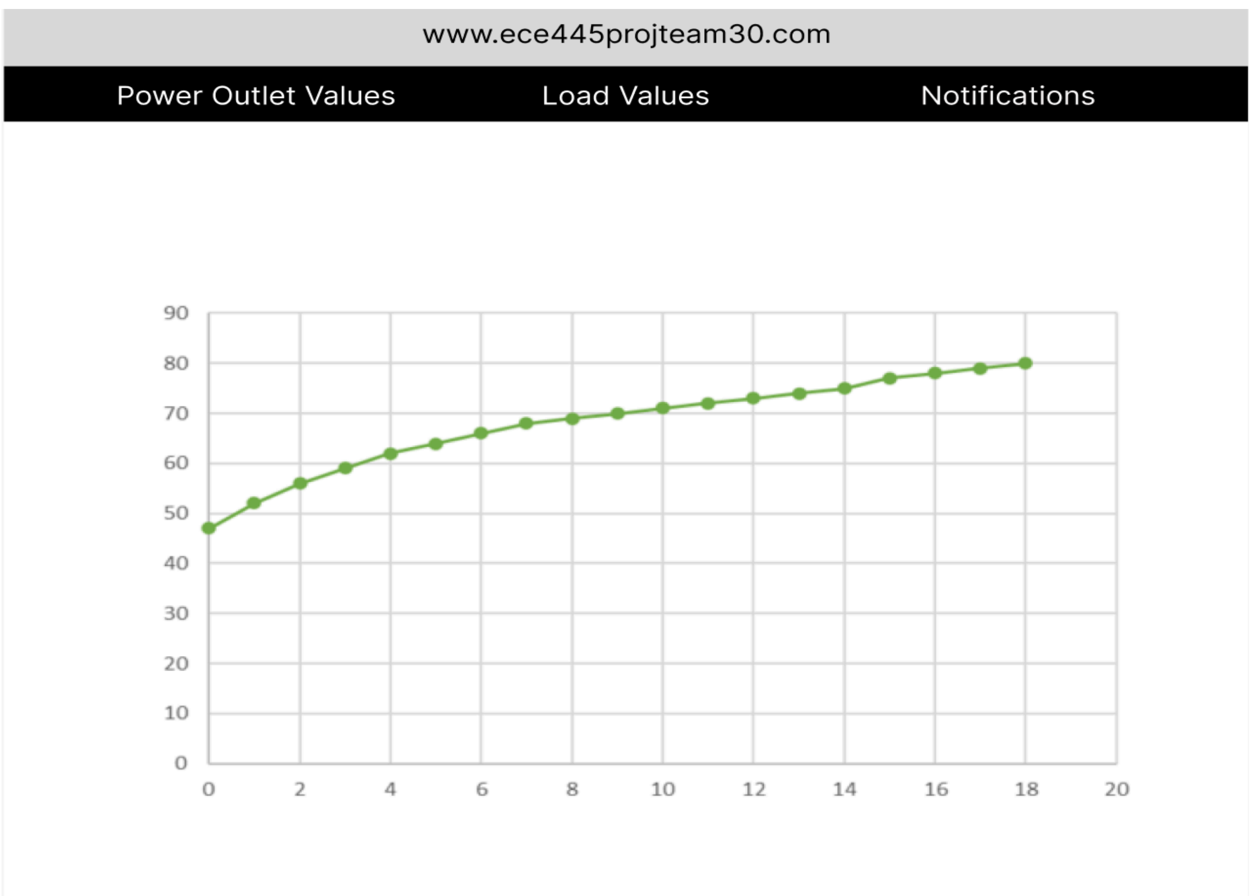

Figure 8: Front-end mock-up for web app

2.6.2 Interfaces

- Control unit ESP32 data
    - Gets data from ESP32 through an established HTTP connection that sends "POST" requests. These requests have SQL "INSERT INTO" statements which insert data into the Google Cloud database.
- Database subsystem
    - The user interacts with the frontend. The frontend sends HTTP "GET" requests to the backend, which gets the data from the database. This data is then analyzed and graphed on the frontend.

2.6.3 Requirements/Verification

This system must satisfy the following requirements:

1. The database must store new data once every 10 seconds from the ESP32.
2. The system must show a notification on the frontend when there's any disturbances, such as voltage or current fluctuations outside of a set tolerance by 5%, or power failures. This notification must appear within 5 seconds.
3. The frontend must have a graph that shows voltage and current data that is refreshed at least once per minute.

We have a set of verification procedures that are explained in detail in the table below.

Requirements	Verification
The database must store new data once every 10 seconds from the ESP32.	• Set up ESP32 Wi-Fi connection and establish HTTP connection with Google Cloud database. Set the ESP32 at least 50 meters away from the laptop running the database. • Write firmware code in ESP32 to send fake time (Unix epoch time), voltage, and current data once every 10 seconds for 5 minutes. This comes out to a total of 30 sets of data being sent. We would send fake data at first just for testing purposes to verify that we can satisfy this requirement. • Use "SELECT *" statement in Google Cloud terminal to see all the data. For success, we would need at least 29/30 data sets to be correctly present in the database.

The system must show a notification on the frontend when there's any disturbances, such as voltage or current fluctuations outside of a set tolerance by 5%, or power failures. This notification must appear within 5 seconds.	● Write firmware code in ESP32 to send fake time (Unix epoch time), voltage, and current data with the voltage set at 5% above a set tolerance, then the voltage set at 5% below a set tolerance, then repeated for the current. We are using fake data to simulate data for verification purposes. ● We will then repeat the prior step but set the voltage to 0V to simulate a power failure. ● After sending the data from the ESP32, we will then start a stopwatch to count the time it takes for the notification to show up. ● For each of the 5 instances, we must see a notification pop up on the frontend of the web application within 5 seconds of sending the data from the ESP32.
The frontend must have a graph that shows voltage and current data that is refreshed at least once per minute.	● Write firmware code in ESP32 to send fake time (Unix epoch time), voltage, and current data that is the exact same for 55 seconds. We are using fake data to simulate data for verification purposes. ● Then, we will change the data that is sent from the ESP32 for the next 55 seconds. ● For every 1 minute, we must see the graph changing to reflect the correct data.

Table 4: Requirements and Verification table for Database Subsystem

2.6.4 Design Decisions

The technologies (React.js, Node.js, Chart.js, Google Cloud database) for the ESP32, its communication with the backend, and the web application were chosen based on the skillset of the group members. One of our group members has experience with most of the aforementioned technologies because he's a Computer Engineering student that has taken classes such as Database Systems and has done software engineering internships. Everyone else in the group has had some level of coding experience through classes such as ECE 220 and other projects.

Therefore, everyone in our group should be able to code and debug portions of the firmware and software by using prior knowledge, online documentation, and problem solving skills.

We also chose Google Cloud to host our MySQL database because we know that we can configure the settings to have the lowest storage type and capacity such that it's enough for this project and is lower in cost compared to Azure and AWS.

2.7 Tolerance Analysis

For our design, we believe the most challenging portion will be the battery charging circuit. We expect there to be two portions of this circuit: the first is the actual battery charger, and the second will be a boost circuit to ensure that our battery voltage is at 5 V, so that after our voltage regulator when we are using battery power to operate our system, it operates as expected. Working from a Circuit Digest Post[1] we observe the circuit schematic shown in Figure 2.

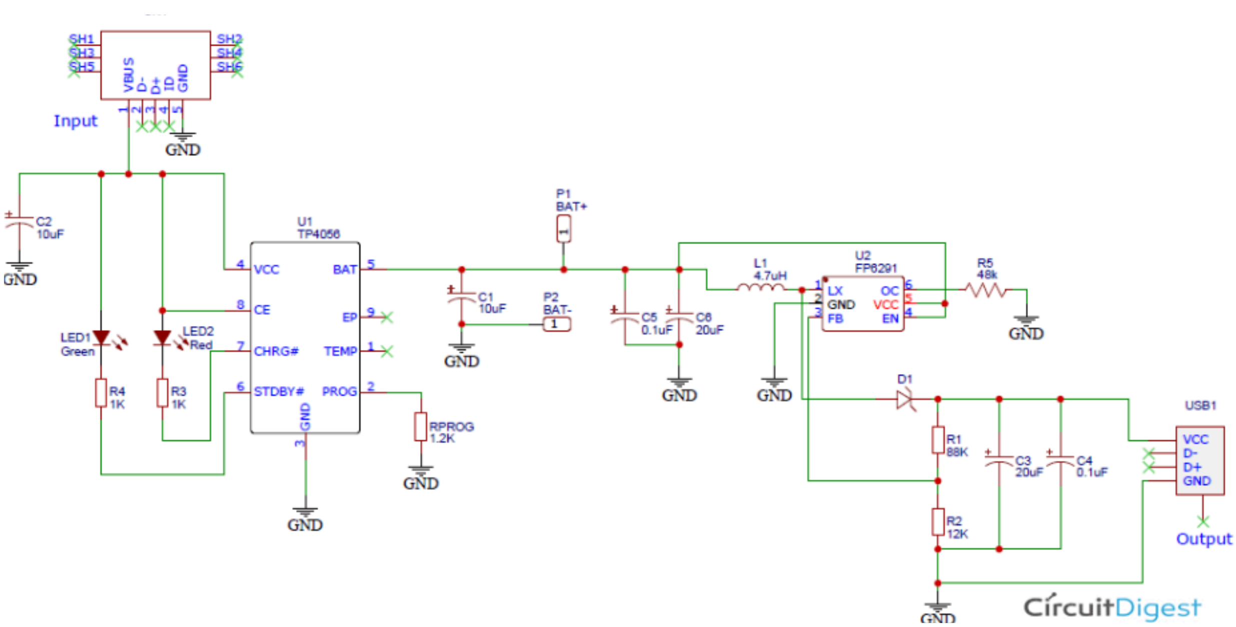

Figure 9: Battery Charge and Boost Circuit

However, instead of a TP4056 as our charging IC, we will be using a TI BQ25638 as our IC since it is intended for LiFePO batteries, which is the chemistry we intend to use. We will also change the boost IC to a MT3608 rather than a FP6291 Boost Converter IC since the MT3608 is rated for higher input voltages and higher output currents based on their datasheets[2,3]. This gives us more flexibility to push more current when the ESP 32 draws more power. Essentially the problem we want to confirm we are solving is that our system can run at full load when being powered just by a battery. Based on this[4] forum post, each GPIO pin is rated for 40 mA, ideally they draw 20 mA for efficiency. When all pins are being used and all modules on an ESP32 are running (Wi-Fi, Bluetooth, etc) it is estimated that there is a total load of 250 mA. Another source says that the chip could draw more than 800 mA of current when doing transmissions over bluetooth and Wi-Fi. We could reduce this number of active pins and thus the total power draw by using a multiplexer to decide which pins to power rather than having them all be powered, we could also code pragmatically and not allow different types of transmissions at the same time. However, for the sake of the tolerance analysis we will use the 800 mA value and double it to 1.6 A just to get a sense of the true worst case power draw.

$$\text{Given: } V_{ESP} = 3.3V\,,\ I_{max} = 1.6A,\ P_{max} \sim 5.28W$$

We need to check what the max load our boost converter can handle. Our MT3608 IC calculates V_{out} by the following ($V_{ref} = 0.6$ V)

$$V_{out} = V_{ref} \times (1 + \frac{R_1}{R_2})$$

If we want our V_{out} to be 5 V then R1/R2 will be 7.33 Ω. This is a feasible ratio and can be achieved. Also assuming a 18650 LiFePO4 battery rated for 3.7 V. We can also calculate the duty cycle necessary to boost the voltage to 5 V. This calculated value must be less than 0.9 (D_{max})

$$V_{out} = \frac{V_{in}}{1-D} \ => \ 5 = \frac{3.7}{1-D} \ => \ D = 1 - 0.74 = 0.26 < D_{max}$$

Back to the total current draw. If we take our assumed battery and look online[5] we see that the minimum current draw of this size battery is 5 A. Looking at the datasheet we see that the minimum efficiency of the converter at this voltage is around 80%. Meaning that 80% of the power consumed is delivered at worst.

$$P_{in} = 3.7V * 5A = 18.5\,W \ =>$$

$$P_{delivered} = 18.5(0.8) = 14.8W \ => \ I_{out} = \frac{P_{delivered}}{V_{out}}$$

$$= 2.96A$$

We have confirmed that we are capable of delivering the necessary power to the load. Now we can use this to understand how long our battery will last. Again referencing the battery inventory website we see that the battery capacity is 3400 mAh. If we assume an average current draw of 300 mA, meaning that all our pins are powered plus some arbitrary amount of current to account for larger draw due to transmissions.

$$3400/300 = 11.3\,hrs$$

We can expect our system to operate on battery power alone for up to 11.3 hours. We are not enthusiastic about this number but we feel that if we add more cells in parallel and add a protection IC perhaps we could elongate the time for which our system stays on.

Another system tolerance we could have considered would've been a timing analysis for all the components in the control unit and ensuring that all signals sent will be received properly. We felt this would be unnecessary since all data is passed through SPI which is standardized, and any functions within IC's that need to be clocked can be synchronized to the clock generated on the ESP32 Chip, and since the generated clock is generated by the digital device we can ensure that the rate at which data is sent is not faster than the rate at which the microcontroller can process it as the only digital device amongst analog devices.

3 Cost & Schedule

3.1 Cost Analysis

3.1.1 Parts/Materials

Component	Part #	Price (per Unit)	Units
ESP32 Microcontroller	ESP32-S3-MINI-1-N8	$3.10	1
Step Down XFMR	ST-4-12	$11.67	1
Current Sense XFMR	B82801B0925A200	$2.94	1
Battery Charging IC	MCP73831T-2ACI/OT	$0.76	2
Battery Charging IC	TP5410	$0.23	5
Voltage Regulator	AZ1117CD-3.3TRG1	-	1
Voltage Regulator	LP2950CZ-5.0/NOPB	$1.28	1
Power Quality IC	ADE9430	$22.40	1
Submetering IC	ADE7953	$5.19	1
Li-Polymer Batteries 18650	EVE 26V 18650 2550mAh 7.5A - Button Top Battery	$5.00	2
		Total	59.253

Figure 10: Cost Analysis

3.1.2 Estimated Hours of Development

An average UIUC EE and CE graduate has a starting salary of $87,769 and $109,176 respectively[7]. For the sake of simplicity, and since we have two electrical engineers and one computer engineer in our group, we will take the mean annual salary of our group to be:

$$(2/3)(\$87,769) + (1/3)(\$109,176) = \$94,904.67$$

Considering a 40-hour work week for 48 weeks of the year, this is equivalent to a wage of $49.43/hour. We estimate that we have worked / will work on this project for roughly 10 hours every week, for a total of 10 weeks in the semester. Therefore we can estimate the total cost of labor for this project as follows:

$$\$49.43/hour \times 10\ hours/week \times 10\ weeks \times 3\ members \times 2.5\ overhead\ factor = \$37,072.50$$

3.1.3 External Materials and Resources

- Machine Shop:
  In addition to the parts listed in Figure 9, we plan to assemble our finished project in an enclosure with the necessary input and output connections and an LCD screen. We anticipate being able to purchase an existing enclosure (waterproof and suitably electrically rated) and screen and simply have the machine shop put everything together. After perusing various offerings from Amazon, Home Depot, and other retailers, we feel a reasonable estimate for the enclosure is roughly \$50 and \$30, respectively. After speaking to Machine Shop personnel, we were informed that our project should be doable within a couple business days, so to be safe, we will estimate on the high side and calculate Machine Shop hours needed to be 4 hours a day for a whole week, or 20 hours total. Considering an hourly rate for the shop to be \$60 (assuming comparability to the Physics Machine Shop[8] since we are unable to find figures for the ECE shop), this would come out to:

$$\$50 + \$30 + (\$60/hour)(20\ hours) = \$1,280$$

- Senior Design Lab Resources:
  In addition to the parts and resources we have listed above, we intend to use the parts and equipment available to us at no cost in the senior design lab. This includes (but is not limited to) soldering equipment, oscilloscopes, multimeters, and commonplace resistors and capacitors as needed.

3.1.4 Total Estimated Cost

All things considered, the total estimated cost for this project is shown in Figure 10.

	Cost
Parts	\$59.25
Labor	\$37,072.50
Machine Shop	\$1,280.00
TOTAL	\$38,411.75

Figure 11: Total project costs

3.2 Schedule

WEEK OF	TASK	TEAM MEMBER(S)
2/26/2024	Prepare for design review	Everyone
	Create firmware to connect ESP32 to Wi-Fi and set up Google Cloud database	Roshan
	Start PCB design + parts should begin arriving	Everyone
3/4/2024	Simulate hardware subsystems and finalize PCB design	Soham, Nicole
	Determine layout, dimensions, and all other necessary specifications for Machine Shop order (due Friday)	Soham, Nicole
	Create firmware to connect ESP32 to Google Cloud using HTTP connection	Roshan
	Aim for first round PCBway order	Everyone
3/11/2024	Spring Break	
3/18/2024	Create the basic frontend design without the graphs for the web app	Roshan
	Breadboard power supply and control unit	Nicole, Soham
	Second Deadline for PCB finalization	Nicole, Soham
3/25/2024	Get notifications working on web app	Roshan
	Continue Breadboarding and debugging on breadboard	Nicole, Soham
4/1/2024	Get graphs working for power outlet input	Roshan
	Start Soldering onto PCB	Nicole, Soham
4/8/2024	Get graphs working for load	Roshan
	Finish soldering, test and debug subsystems separately and together	Nicole, Soham
4/15/2024	Mock demo	Everyone
	Complete any necessary debugging	Everyone

4/22/2024	Final demo	Everyone
	Mock presentation	Everyone
4/29/2024	Final presentation	Everyone
	Final report (due Wednesday)	Everyone

Table 5: Weekly project schedule delineated by group member

4 Ethics & Safety

4.1 Ethics

In terms of ethics, we do foresee a few ethical issues that could arise during the development of our project as well as the misuse of our prototype. For example, during the development of our project, we could have poor teamwork, alienate group members, or stop participating in the project. Additionally, we could reference the designs of other submeters or power quality monitors on the market. In terms of the misuse of our prototype, the data that is collected is private data for the users and must be protected. There are people that could try to access other people's data, which is a violation of laws and regulations.

Our group is committed to maintaining the highest ethical standards throughout the course of this project by following the IEEE Code of Ethics, such as upholding the highest standards of integrity, treating everyone equally and with respect, and making sure that the Code of Ethics is followed by the entire group. We will avoid these ethical breaches and follow the aforementioned Code of Ethics through the following:

1. We've set up a group chat through iMessage where we have consistently communicated our schedules, when we can work, and what we plan to work on. This has allowed us to split the workload well and finish our work on time while ensuring high quality. We have also set up a Google Drive folder for all our ECE 445 documents so that we can stay organized and stay on the same page.
2. We will hold each other accountable to make sure all of the work we do is of high quality and is original. If questions or uncertainties arise, we will make sure to contact and consult with experts in the field such as Surya (our TA), Professors, Machine Shop Technicians, or our mentor Mr. Jack Blevins.
3. Since this project isn't planned to be released to the public, we don't plan on addressing data privacy since having secure encryption could take months to create. However, if we were to sell the product, we would have all data be encrypted and password protected so that there is data privacy. For now, one of our stretch goals will be to incorporate password protection for the project.

4.2 Safety

In terms of the safety regulations for this project, we have identified the following risks: lithium-ion batteries, as well as the voltage and current transformers since we will step down the voltage and current. To combat these issues, we have talked to Jason and Surya, both being TAs in ECE 445, to identify a workaround for the lithium-ion batteries. They suggested using LiFePO batteries since the chemistry in the battery is less likely to cause a fire or other safety hazards. Additionally, we will mark the device as a potential safety hazard that shouldn't be used by a minor without an adult present. Along with the safety hazard, we will have a safety manual for the final demo that we will adhere to. Lastly, we will use adequate lab safety equipment such as safety goggles during the construction of the project to ensure that we are safe. On top of safety goggles, we will also follow OSHA safety standards such as regulation 1910.137, which outlines how to use proper electrical protective equipment. For example, if we are dealing with high voltage or high current, we will look to use rubber gloves to protect ourselves, which adheres to regulation 1910.137.

References

"1910.137 Electrical Protective Equipment Standards." *Occupational Safety and Health Administration*, 11 Apr. 2014, https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.137.

"18650 Batteries - High Quality Rechargeable Lithium-Ion Batteries." *18650BatteryStore.Com*, https://www.18650batterystore.com/collections/18650-batteries?pf_rs2_discharge_current_amps=4.90%3A4.90. Accessed 8 Feb. 2024.

"1MHz Step-Up PWM Converter with 2A Switch Current." *Perfect Microelectronics Co., Ltd. FP6291*, https://file2.dzsc.com/product/17/08/05/1125865_100842357.pdf. Accessed 8 Feb. 2024.

Communications, Grainger Engineering Office of Marketing and. "Machine Shop." Physics.illinois.edu, physics.illinois.edu/research/facilities/machine-shop#:~:text=Prices%20Professional%20products%20%2460.00%2Fhour%20plus%20cost%20of%20materials. Accessed 22 Feb. 2024.

Grainger Engineering Office of Marketing and Communications. "Salary Averages." Ece.illinois.edu, ece.illinois.edu/admissions/why-ece/salary-averages. Accessed 22 Feb. 2024.

"High Efficiency 1.2MHz 2A Step Up Converter." *AEROSEMI*, https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf. Accessed 8 Feb. 2024.

"How to Build a 18650 Lithium Battery Charger and Booster Module." *Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects*, https://circuitdigest.com/comment/31902#comment-31902. Accessed 8 Feb. 2024.

JKElectronics. "Esp32 Max Current Output - General Electronics - Arduino Forum." *Arduino Forum*, Arduino Forum, 6 Nov. 2022, https://forum.arduino.cc/t/esp32-max-current-output/1050205.

Staff, Editorial. "Insight Into ESP32 Sleep Modes & Their Power Consumption." *Last Minute Engineers*, Last Minute Engineers, 23 Dec. 2018, https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/#:~:text=ESP32%20Active%20Mode,-Normal%20mode%20is&text=Since%20everything%20is%20always%20active,and%20Bluetooth%20are%20used%20simultaneously.