E-Ink IoT Labels

Project Documentation

6 May 2022

Carlos Gonzalez
Benjamin Jones
Alexander Mote
Emanuel Murillo

Oregon State University
ECE44x Senior Design
Fall 2021 – Spring 2022

Table of Contents

1. Overview

1.1. Executive Summary

The E-Ink Label is an IoT device that wirelessly updates an e-ink display via WiFi. The label allows users to display information to an audience using a wireless, low-power display. The system will be built using a custom designed PCB that encompasses all the components. Information on the display is updated through a website that the user enters on a personal computer. The project is still in the early stages of development. Extensive research on the main focus areas of the project is still ongoing such as modularity, low-power consumption, efficient wireless connectivity, and solar-power capabilities.

1.2. Team Communication Protocols and Standards

Team Member	Role	E-Mail Address
Carlos Gonzalez	Web Developer	gonzalc5@oregonstate.edu
Benjamin Jones	Hardware Engineer	jonesben@oregonstate.edu
Alexander Mote	Software Engineer	motea@oregonstate.edu
Emanuel Murillo	Power Systems Engineer	murilloe@oregonstate.edu

Communication Type	Primary Method	Secondary Method
Team Meetings	In Person at OSU Mondays, 10AM	Zoom Meeting Mondays, 10AM
Inter-Meeting Communication	Discord Group Chat	Group Text Message
Urgent Messages	Group Text Message	E-Mail
Progress Updates	Trello Board	Discord Group Chat
Contact with Instructors	Canvas Message (Other team members included)	E-Mail (Other team members included)
Contact with Other Teams	Canvas Discussion Boards	E-Mail

- Decisions on where to hold each team meeting will be determined at the end of the previous meeting by majority vote.
- The team meetings can be expected to take at least 30 minutes and up to 60. Absences should be conveyed to the rest of the team with at least 24 hours notice.

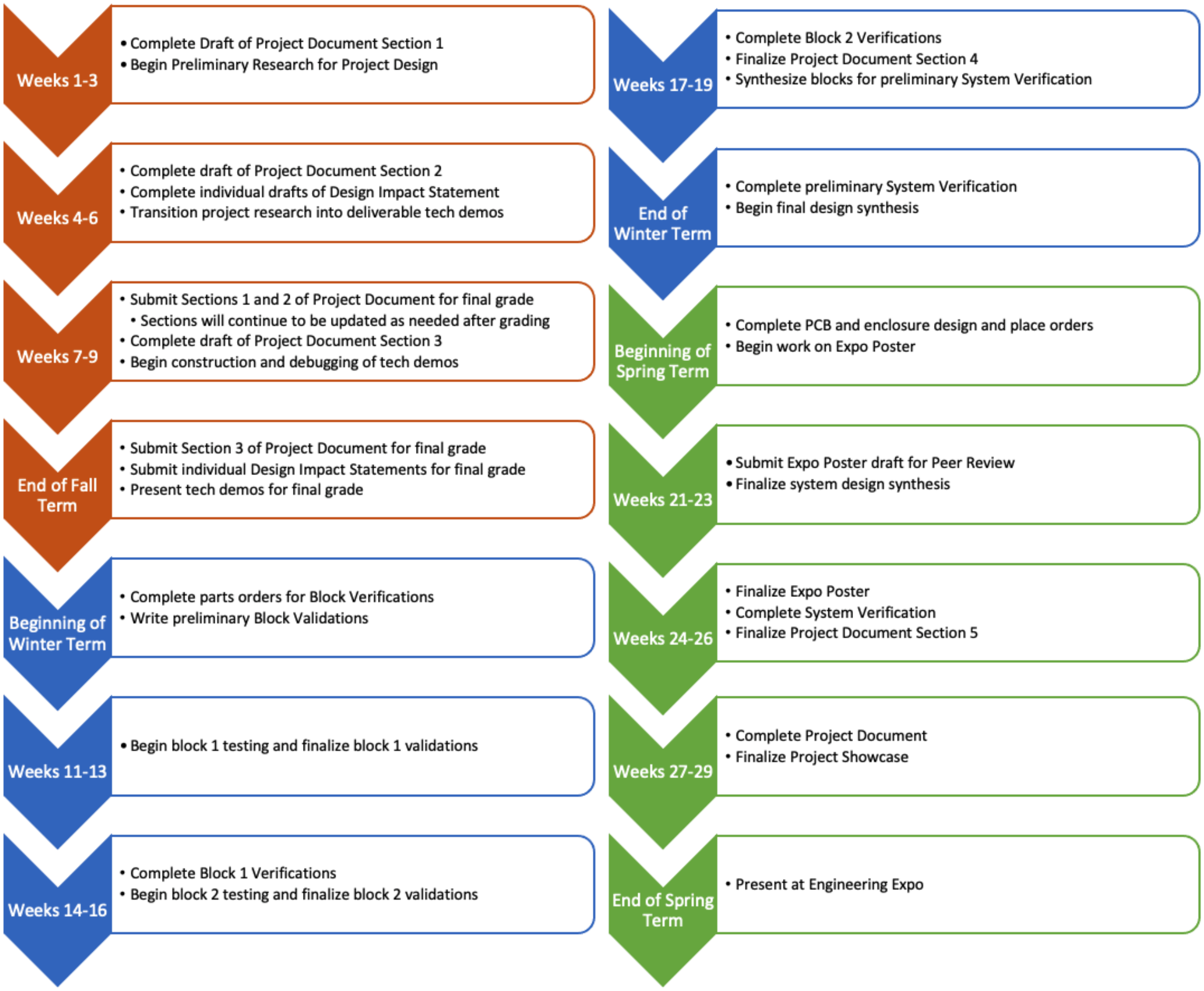
- Individual direct messages between group members are discouraged unless absolutely necessary, in the interest of facilitating a cohesive group understanding.
- Decisions involving the entire team will be made by majority vote. Other decisions will be made on an individual level. If a single team member's decisions impact other members of the group, these impacts should be brought to the attention of that team member, with the understanding that those decisions are still theirs to make.
- Conflicts or other discrepancies in team decisions should first be discussed as a group. If possible, they will be resolved by majority vote. If these discussions are unsuccessful in resolving the conflict, a team member can choose to contact an instructor (ideally including the entire team in the point of contact as shown above) or reach out to the OMBUDS office.

1.3. Gap Analysis

This project aims to improve upon the design of pre-existing products in multiple ways. These improvements are: increased battery life, lower power consumption, support for multiple devices, and solar power. We aim to cut down on the need for new signs as these could be a useful way to fill their function and be able to change the labels with ease.

One potential end user of this device is schools or businesses who want to have room number labels that can also display who is currently using the room. These will be able to update in real time and account for different events occurring in a given room. As buildings can be remodeled or staff can change rooms overtime, this removes the need for new room signs to be created. A clearer labeling of rooms reduces confusion for new students or when a new term starts for schools. Similarly for businesses, labeling of conference rooms and their durations will give more transparency of how the business is operating and will also allow customers or partners to navigate their building easier.

1.4. Timeline



Weeks 1-3
- Complete Draft of Project Document Section 1
- Begin Preliminary Research for Project Design

Weeks 4-6
- Complete draft of Project Document Section 2
- Complete individual drafts of Design Impact Statement
- Transition project research into deliverable tech demos

Weeks 7-9
- Submit Sections 1 and 2 of Project Document for final grade
  - Sections will continue to be updated as needed after grading
- Complete draft of Project Document Section 3
- Begin construction and debugging of tech demos

End of Fall Term
- Submit Section 3 of Project Document for final grade
- Submit individual Design Impact Statements for final grade
- Present tech demos for final grade

Beginning of Winter Term
- Complete parts orders for Block Verifications
- Write preliminary Block Validations

Weeks 11-13
- Begin block 1 testing and finalize block 1 validations

Weeks 14-16
- Complete Block 1 Verifications
- Begin block 2 testing and finalize block 2 validations

Weeks 17-19
- Complete Block 2 Verifications
- Finalize Project Document Section 4
- Synthesize blocks for preliminary System Verification

End of Winter Term
- Complete preliminary System Verification
- Begin final design synthesis

Beginning of Spring Term
- Complete PCB and enclosure design and place orders
- Begin work on Expo Poster

Weeks 21-23
- Submit Expo Poster draft for Peer Review
- Finalize system design synthesis

Weeks 24-26
- Finalize Expo Poster
- Complete System Verification
- Finalize Project Document Section 5

Weeks 27-29
- Complete Project Document
- Finalize Project Showcase

End of Spring Term
- Present at Engineering Expo

1.5. References and File Links

1.5.1. References

1.5.2. File Links

IEEE Citation Guidelines.

1.6. Revision Table

Date	Editor(s)	Section(s) Edited	Description

30 Apr 2021	A. Mote	1.4	Completed timeline for Winter and Spring terms.
3 Dec 2021	A. Mote	1.4	Added graphical element to timeline as recommended by instructors.
19 Nov 2021	A. Mote	1.2	Added team member roles.
12 Nov 2021	A. Mote	1.2 1.4	Added team member contact information. Transferred timeline to text format for easier readability.
12 Nov 2021	E. Murillo	1.1, 1.3	Added to sections 1.1 ‘Executive Summary’ and 1.3 ‘Gap analysis’
29 Oct 2021	E. Murillo	1.3	Expanded on potential end-user functionality
27 Oct 2021	A. Mote	1.2	Re-formatted content as a table and bulleted list; added conflict resolution and decision making
22 Oct 2021	C. Gonzalez	1.1	Added Section 1.1 ‘Executive Summary’
22 Oct 2021	A. Mote	1.4	Added a high-level overview timeline for Fall term to 1.4 ‘Timeline’
20 Oct 2021	B. Jones	1.3	Added Section 1.3 ‘Gap Analysis’
19 Oct 2021	A. Mote	1.2 1.5.1 1.6	Added Section 1.2 ‘Team Communication Protocols and Standards’ Added IEEE Citation Guidelines to 1.5.1 ‘References’ Added 1.6 ‘Revision Table’

2. Requirements, Impacts, and Risks

2.1. Requirements

2.1.1. Display & Readability

- Project Partner Requirement (PPR): The label must use an E-Ink Display, and the label must be easy to read.
- Engineering Requirement (ER): The completed circuit design must interface with and utilize an E-Ink Display. Any text shown on the E-Ink Display must be identifiable (i.e. legible) from a distance of at least 5 feet by at least nine out of ten testers.
- Verification:
    1. The completed circuit design will consecutively send and display twenty different elements of data on the E-Ink Display.
    2. At least nine out of ten testers randomly selected from OSU students will be able to correctly read these displays from a distance of at least 5 feet.

2.1.2. Internet of Things

- PPR: The label must be an IoT Device.
- ER: The completed circuit design must be able to connect to and read data from the Internet.
- Verification:
    1. The completed circuit design will connect to the Internet.
    2. Requirements will be verified when the completed circuit design consecutively retrieves twenty different elements of data from an online database and displays them on the E-Ink Display.

2.1.3. Power Efficiency

- PPR: The label must consume little power and be operable for long periods of time between charging.
- ER: The completed circuit design must have a low-power idle state. The system must be able to run for a week's worth of updating the label without requiring charge.
- Verification:
    1. The completed circuit design will draw no more than 200 mW of power when not updating the E-Ink Display (in idle mode).
    2. The completed circuit design will have its battery charge level measured over a 1/2 hour period of average expected use.
    3. Requirement will be verified when the battery charge level is maintained at a constant value over a two hour period of average expected use.

2.1.4. Rechargeability

- PPR: The label must be rechargeable.
- ER: The completed circuit design must be able to recharge the battery that is powering the system.
- Verification:
  4. An external power source must be able to connect to our system via usb or solar panel.
  5. The completed circuit design will convert power received into 4.2V and no more than 175mA.
  6. Once the battery is fully charged, charging will be paused to protect the battery.

2.1.5. Updating the Display

- PPR: The label must update quickly.
- ER: The E-Ink Display updating process will take no more than ten seconds.
- Verification:
  1. The E-Ink Display will be timed on both startup and updating; five different updates will be sent. Timing will start when the screen begins to change, and end when the change is complete.
  2. The startup and all updates must be completed and displayed on the E-Ink Label in less than ten seconds.

2.1.6. Speed of Setup

- PPR: The label must have a quick and easy setup process.
- ER: 9 out of 10 end users will be able to set up the E-Ink Display without need for any programming or electrical knowledge.
- Verification:
  1. Test users will be shown how to set up the E-Ink Display by connecting it to the OSU Access network. This process will be demonstrated by a project team member.
  2. Users will then be asked to set up the system themselves with no input from the project team member. Set up is defined as getting the display connected to a database and displaying an element from that database.
  3. Requirements will be verified if nine out of ten users are able to set up the display in five minutes or less.

2.1.7. Ease of Use

- PPR: The label must be easy to use.
- ER: 9 out of 10 end users will be able to update the E-Ink Display without need for any programming or electrical knowledge.
- Verification:

1. Test users will be shown how to modify the E-Ink database by a member of the project team.
2. Testers will then be asked to modify the display database themselves with no input from the project team member (changing a course number, the time a class will start/end, etc.)
3. Requirement will be verified when nine out of ten users are able to update the database in sixty seconds or less.

2.1.8. Enclosure

- PPR: The label must be safe to use.
- ER: The completed design must be fully enclosed to prevent tampering and damage from bumping or contact sustained during expected use.
- Verification:
  1. The completed circuit design will be fully enclosed, with the exception of the display screen.
  2. The combined enclosure and circuit design profile will be no larger than 6.5” x 5.5” x 2”.
  3. The enclosure will be able to withstand light physical contact, such as bumping, knocking, tapping, and pressing, without any loss of functionality.

2.2. Design Impact Statement

2.2.1.Introduction

It is easy, when designing and constructing an engineering project, for an engineer to focus solely on the end product, and disregard the impact that product may have on the wider world at every stage of development. However, even small projects like this one are built within a larger system, and it is important to understand and assess these issues when working on a project of any size. This section seeks to explain, understand, and address the impact of our E-Ink IoT Label design on the social, environmental, economic, and public welfare sectors.

2.2.2. Public Health, Safety, and Welfare Impacts

As with any device that connects to the Internet and displays information to an end user, there are a handful of public safety issues that must be considered when designing these IoT labels. If the interface to update these labels can be accessed by nefarious actors, the content displayed on the labels could be changed to cause confusion, harm, or other negative public consequences. This has been known to happen in the past with IoT devices, even ones created for commercial use; Tatum Hunter wrote in November of 2021 about a host of software issues that made smart-home devices liable to hacking with minimal effort [1].

As with any IoT device, there are numerous possible security issues that arise as a natural extension of the device’s connection to the Internet [2]. Attempting to address and solve all of these problems is untenable in a project of this scale; as such, this design will focus on a few key security features that are most relevant to the project’s Internet use. One is to ensure that users have a secure login to access the interface; because our project is designed for use in OSU campus buildings, we will be using MyOSU to meet this goal. Another is to keep a log of which users have been using the system, and which modifications have been made; this may not be implementable without the help of OSU's IT department, but it is possible to achieve. The last, and most important to keep in mind during the design process, is to use secure transfer protocols to exchange information between the server and the labels. This is achievable by use of HTTPS protocol.

The use of lithium batteries in our design also requires addressing in a public health context. Unfortunately, the majority of lithium batteries use rare earth metals like cobalt that are mined from areas which use abusive work conditions and child labor [5]. The lack of accountability inherent in globalization and corporatization have made it difficult to track the ethical sourcing of these materials; however, some battery manufacturers have begun to phase cobalt out of their design process as a result of these problematic sources [6]. Our project will make a concerted effort to source batteries without cobalt, so that this project does not support the unethical labor practices used in mining these materials.

2.2.3. Cultural and Social Impacts

The sociocultural impacts of a device like this are difficult to notice at first glance; however, upon closer inspection, the inherent pros and cons of this technology in a larger social light begin to come to the surface. Oregon State University has a large population of students for whom American English is not their first language; unfortunately, OSU does not track the primary language of their students in their demographics, so an exact number or percentage of students who meet this qualifier is uncertain. What is certain, however, is that ESL literature places an important emphasis on making translingual identities feel valued and welcomed in the multilingual college learning environment [3]. These labels could allow for a more holistic embrace of the translingual community of OSU by displaying information in a rotation of multiple languages. In order to do this, our labels will need to support Unicode text formatting, which our design currently intends to do.

In a more general sense, this label also has the opportunity to cultivate a better sense of belonging in new OSU students who do not know their way around campus. Studies in the transition to college life show that the first two weeks of the initial term are the most crucial to a student getting their feet under them and acclimating to the systems and services available to them at their university; unfortunately, this is also the time where students are either most apprehensive to seek help, or are unaware of what they need to do in order to succeed [4]. Because of this discrepancy, most students do not seek out the help they need until it is too late, causing their success within the university to suffer. Not only do these labels allow for additional class information to be displayed, removing one thing from students' minds as they navigate the campus, but they could also be used to display other information regarding services available to students, and their locations or contact information. Seeing something like this on a place that is known to attract the attention of new students, such as a room label, may be more successful in informing students about these services and how to reach out to obtain them.

Another issue that is often overlooked when designing and implementing any new technology is the inclusion of accommodations for individuals with disabilities. For example, an E-Ink display with changing text would ideally have rewritable Braille text to go along with it, and while "E-Ink adjacent" systems for displaying rewritable Braille text do exist [12], they are not produced in high enough quantity to be economically sound for this project. Accessibility concerns will be discussed with the OSU office of Disability Access Services in order to find which issues can be accounted for in our design, and which are outside the scope of the current project.

2.2.4. Environmental Impacts

This section of the assessment is arguably the most relevant to our project; after all, why does it matter how little power a digital display consumes, when a non-electronic display consumes zero power and contributes very little to overall waste generation? This is one of the most important impacts we must address and mitigate in our overall design, driving down power consumption and waste generation enough that the benefits of installing such a display can outshine the environmental downsides.

The largest environmental impact our design faces comes from the use of a lithium ion battery. These batteries often use rare earth metals such as cobalt in their design and manufacturing, which have a problematic relationship with the environment to say the least [7]. The recycling processes for these elements are still developing, and expensive physical and chemical processes are required to separate the rare earth materials from the electrical components used in these batteries. Additionally, many of these batteries don't even get the chance to be recycled, as they are simply thrown away and added to landfills or other dumps, leading to the need for more equipment to be mined. These materials also break down further in their landfills, leading to them potentially contaminating the surrounding environment [8].

While it is not realistic to place the entire weight of the lithium battery and rare earth metal mining industry on this project, we will still recognize our place in that system by sourcing long-lasting batteries that, ideally, do not contain any rare earth metals. We will also include a section in the user guide describing how to safely replace and recycle these batteries. In an effort to decrease the power consumption of the labels, we will also be including a solar panel that will use renewable energy to charge the batteries of the device. Ideally, this will lead to the device never needing a wired charge; however, due to the use of the devices indoors, it is possible that they will occasionally require a wired charge, albeit less often than if no solar panel were present. Of course, the manufacturing of solar panels has its own similar set of material mining and waste disposal issues [9]; therefore, the device will include no more solar panels than are absolutely necessary to charge the batteries, and our user guide will include similar instructions for replacement and proper disposal of the solar panels.

2.2.5. Economic Impacts

The financial side of this design is a difficult thing to balance. On one hand, we want to include quality parts that will last and not require replacement for as long as is feasible, and we do not want to undervalue the efforts of our design time, nor do we want to be underpaid for the physical labor of things like assembly. On the other hand, we do not want the final product to be so expensive that it is prohibitive for individuals or institutions to purchase any labels they may need. Currently, the closest thing to our design that is available on the market costs US$34.95 without a battery, requiring a constant wired connection to function [10]; our design will attempt to match this market price, but it may be difficult to do so unless orders are made in bulk, since

ordering parts in small quantities often leads to a significant increase in price. These are qualifications we will need to give to any prospective buyers.
Going hand-in-hand with the other concerns we have presented in this assessment are the financial and work-hour cost of evaluating and solving each of the issues that arise. Developing and testing secure connectivity methods will require an initial upfront investment of time and energy that may need to be spent on other aspects of the project. Using components that are long lasting and contain ethically sourced materials will no doubt be more expensive than using cheap components without considering their longevity or material impact. These constraints are neatly wrapped up by the concept of the Project Management Triangle, where a design's scope, cost, and time must be balanced in order to ensure a quality product. This concept, which has been used for over 60 years, is met in modern times with almost universal criticism [11] for oversimplifying the development process and its expected outcomes. All of this further highlights the complexity of balancing project expectations with budget and scope.

2.2.6. Conclusion

All in all, this assessment serves as a testament to how genuine, deep thought about a design and its many moving parts can uncover some surprising truths and issues regarding its holistic impact. Even a project as small in scope and application as this can be tied to larger sociocultural, ecological, and public safety concerns that should be identified and addressed early in the design process. Having each member of the project team perform this process allows for unique perspectives and solutions to be added to the conversation that may not have arisen in a singular group effort. Through comparing these assessments, a best path forward in the development process can be determined early on, and the scope of the project can be decided with the greatest possible impact consideration in mind.

2.3. Risks

ID#	Description	Category	Prob.	I.	Performance Indicator	Responsible Party	Action Plan
01	Vendor Delay	Timeline	50%	H	Part lead time	Emanuel	Retain
02	Material Price increase	Cost	10%	L	Price Change	Ben	Reduce
03	WiFi connectivity issues	Technical	15%	M	E-Ink Displays failing to connect to database	Alex	Reduce
04	User Input Accessibility	Technical	10%	L	Unintuitive or Frustrating UI	Carlos	Reduce
05	COVID-19 Resurgence	Public Health	15%	M	University announcement of remote transition.	N/A	Transfer

06	Decrease in group size or member replacement	Team	1%	L	Minus a group member or new member added to team	N/A	Reduce
07	Device components lost or broken	Technical	10%	M	Component missing	Ben/ Emanuel	Retain
08	Security attack on website	Technical	15%	M	Device updated with unrecognizable information	Carlos/Alex	Avoid

2.4. References and File Links

2.4.1. References

[1] T. Hunter, "Buggy Software in Off-Brand Smart Home Devices Is a Hacker's Playground," The Washington Post, Nov. 18, 2021. https://www.washingtonpost.com/technology/2021/11/18/smart-home-security/ (accessed Nov. 29, 2021).
[2] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: Perspectives and Challenges," Wireless Networks, vol. 20, no. 8, pp. 2481–2501, Jun. 2014, doi: 10.1007/s11276-014-0761-7.
[3] R. Jain, "Global Englishes, Translinguistic Identities, and Translingual Practices in a Community College ESL Classroom: a Practitioner Researcher Reports," TESOL Journal, vol. 5, no. 3, pp. 490–522, Aug. 2014, doi: 10.1002/tesj.155.
[4] E. E. Morales, "Navigating New Worlds: a Real-Time Look at How Successful and Non-successful First- Generation College Students Negotiate Their First Semesters," International Journal of Higher Education, vol. 1, no. 1, May 2012, doi: 10.5430/ijhe.v1n1p90.
[5] T. Cheshire, "Cobalt Mining for Phones: How You Could Be Holding a Product of Child Labour," Sky News. Feb. 28, 2017, [Online]. Available: https://www.youtube.com/watch?v=AA6BnitgdP8.
[6] E. Airhart, "Alternatives to Cobalt, the Blood Diamond of Batteries," Wired, Jun. 07, 2018. https://www.wired.com/story/alternatives-to-cobalt-the-blood-diamond-of-batteries/.
[7] N. Haque, A. Hughes, S. Lim, and C. Vernon, "Rare Earth Elements: Overview of Mining, Mineralogy, Uses, Sustainability and Environmental Impact," Resources, vol. 3, no. 4, pp. 614–635, Oct. 2014, doi: 10.3390/resources3040614.
[8] A. Nigro, G. Sappa, and M. Barbieri, "Boron Isotopes and Rare Earth Elements in the Groundwater of a Landfill Site," Journal of Geochemical Exploration, vol. 190, pp. 200–206, Jul. 2018, doi: 10.1016/j.gexplo.2018.02.019.
[9] Ç. Gönen and E. Kaplanoğlu, "Environmental and Economic Evaluation of Solar Panel Wastes Recycling," Waste Management & Research: the Journal for a Sustainable Circular Economy, vol. 37, no. 4, pp. 412–418, Feb. 2019, doi: 10.1177/0734242x19826331.

[10] Adafruit Industries, “Adafruit MagTag - 2.9" Grayscale E-Ink WiFi Display,” www.adafruit.com. https://www.adafruit.com/product/4800 (accessed Nov. 05, 2021).
[11] R. Atkinson, “Project management: cost, Time and quality, Two Best Guesses and a phenomenon, Its Time to Accept Other Success Criteria,” International Journal of Project Management, vol. 17, no. 6, pp. 337–342, Dec. 1999, doi: 10.1016/s0263-7863(98)00069-6.
[12] K. Ren, S. Liu, M. Lin, Y. Wang, and Q. M. Zhang, “A Compact Electroactive Polymer Actuator Suitable for Refreshable Braille Display,” Sensors and Actuators A: Physical, vol. 143, no. 2, pp. 335–342, May 2008, doi: 10.1016/j.sna.2007.10.083.

2.4.2. File Links

E-Ink Display Datasheet.
ESP32 Datasheet.
P121 Datasheet.

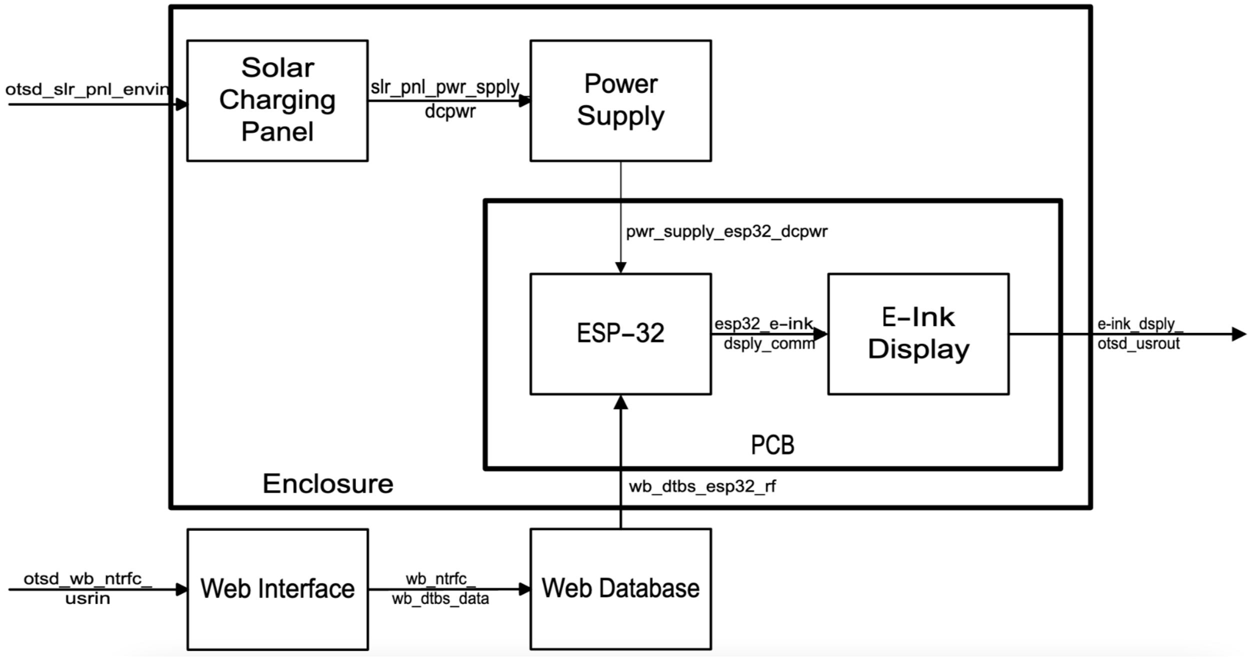
2.5. Revision Table

Date	Editor(s)	Section(s) Edited	Description
6 May 2022	A. Mote	2.2	Added final Design Impact Assessment.
6 Mar 2022	B Jones	2.1	Edited maximum enclosure dimensions
3 Dec 2021	A. Mote	2.1 2.3	Added Accessibility requirement. Adjusted risks based on feedback provided by instructors.
22 Nov 2021	A. Mote	2.1	Adjusted requirement verification format to match instructor guidelines. Updated requirements to reflect improvements suggested by instructor.
19 Nov 2021	A. Mote	2.1	Adjusted requirements according to instructor recommendations.
12 Nov 2021	C. Gonzalez	2.3	Added 4 additional risks (05-08)
12 Nov 2021	A. Mote	2.1 2.4 2.3	Updated requirement formatting according to new instructor guidelines. Added datasheets for relevant components to references. Adjusted risks to better express external factors, rather than internal issues.
12 Nov 2021	B. Jones	2.3	Improved risk table
29 Oct 2021	E. Murillo	2.3	Added to risk table
29 Oct 2021	A. Mote	2.1	Added a few baseline requirements.

29 Oct 2021	B. Jones	2.3	Began filling out risk table
27 Oct 2021	A. Mote	2.2	Added brief explanation of blank section for Fall term.

3.Top-Level Architecture

3.1. Block Diagram



3.2. Block Descriptions

Name	Description
E-Ink Display Champion: Alex Mote	The E-Ink display is the primary reason for this project's existence; it is a rewritable digital display that consumes very little power, ensuring sustainability and longevity. This display will interface with the ESP32 chip to show what the user inputs, and to know when to change the display to show any updates or other modifications made by the user.
ESP32 Champion: Benjamin Jones	The Esp32 block reads data from the web database over wifi, and then updates the E-Ink display using the data that was fetched.

Solar Cell Champion: Carlos Gonzalez	The block champion for the solar panel is Carlos Gonzalez. The solar panel will be our main method of charging the battery contained in the system, in an effort to avoid the need for wired charging. The intended target location for the overall system will be on campus room doors. Thus, a method for wireless charging will be more practical for this purpose. The low-power idle nature of the ESP32 chip and E-Ink technology should allow the circuit ample time to use this panel to recharge between the moments of power consumption where the screen is updated. The expected size of the solar panel will be similar to that of a calculator's solar panel. This should be sufficient to recharge the Li-polymer battery in reasonable time.
Power Supply Champion: Emanuel Murillo	The power supply will be a DC-DC buck converter. It will take power from a rechargeable 3.6V button battery. The power supply will provide an output voltage to the ESP32 chip and also incorporate a solar panel to recharge our button battery.
Web Database Champion: Alex Mote	The web database is used to store user inputs online in a way that can be accessed by the ESP-32 chip, in order to obtain the information that will be shown on the E-Ink Display. This database will be written to by the user interface, with SQL code being used to modify the database according to the user's inputs. It will be read wirelessly by the ESP-32 using the CircuitPython code flashed onto the chip.
Enclosure Champion: Benjamin Jones	The Enclosure holds all of the other blocks, while allowing the display to be read, and the solar panel to get light. It also needs to be designed in such a way that it can be hung on a wall.
Web Interface Champion: Carlos Gonzalez	The web interface is where the user directly interacts with the digital display information. In order to update the information displayed, the user must enter the webpage and submit the information to be sent to the display. The webpage will allow the user to interact with a textbox, upload an image, and preview the display before submitting. The information submitted will be sent to the web database to be stored and used by the ESP-32 chip.
PCB Champion: Emanuel Murillo	The PCB will connect all individual circuit elements and ensure that they can all interface with one another successfully. A custom-designed PCB will be used to accommodate our project's unique components and engineering requirements.

3.3. Interface Definitions

Name	Properties
otsd_wb_ntrfc_usrin	● **Type:** Images
otsd_wb_ntrfc_usrin	● **Type:** Text
e-ink_dsply_otsd_usrout	● **Other:** 4-Color Grayscale ● **Type:** Text, Images ● **Usability:** Readable by 9/10 users from ≥5 ft
esp32_e-ink_dsply_comm	● **Messages:** Display Functions (Clear screen, display text, etc.) ● **Protocol:** SPI ● **Vnominal:** 3.3V
pwr_spply_esp32_dcpwr	● **Inominal:** 500mA ● **Ipeak:** 1.5A ● **Vmax:** 3.5V ● **Vmin:** 3.1V

3.4. References and File Links

3.4.1. References

3.4.2. File Links

3.5. Revision Table

Please add revisions to the top of the table, so that changes appear in reverse chronological order.

Date	Editor(s)	Section(s) Edited	Description
30 Apr 2021	A. Mote	3.2, 3.3	Added final versions of Block Descriptions and Interface Properties after block verifications.
3 Dec 2021	B. Jones	3.3	Updated some interface Properties
3 Dec 2021	A. Mote	3.2	Changed to table format as requested by instructors.
19 Nov 2021	B. Jones	3.2 3.3	Added Block descriptions Added Interface definitions
19 Nov 2021	E Murillo	3.2	Added Block descriptions
19 Nov 2021	C. Gonzalez	3.2	Added descriptions of block I am championing
19 Nov 2021	A. Mote	2.1 2.2 2.3	Began first draft of block diagram. Added descriptions of blocks I plan on championing. Added interfaces and properties for blocks I plan on championing.

4. Block Validations

4.1 E-Ink Display

4.1.1 Description

The E-Ink display is the primary reason for this project's existence; it is a rewritable digital display that consumes very little power, ensuring sustainability and longevity. This display will interface with the ESP32 chip to show what the user inputs, and to know when to change the display to show any updates or other modifications made by the user.

4.1.2 Design

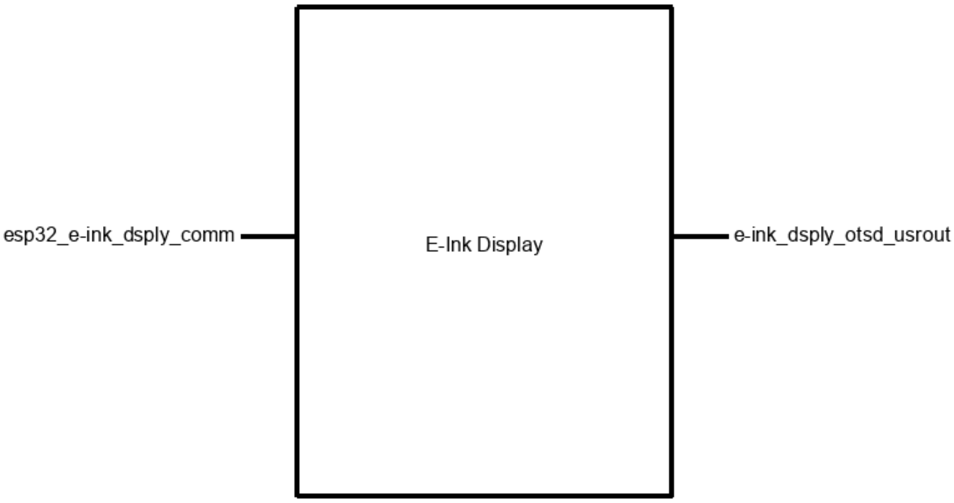


Figure 1: Block Diagram

Pseudocode:

```
global pixel_data;
clear_area(x1, x2, y1, y2):
    // set all pixel values in given
    x-y coordinate range to zero
update_area(x1, x2, y1, y2):
    // if there are pixel values in pixel_data:
    // send pixel values to screen buffer
    // once all pixel values are sent, update screen
display_text(position, font_size, string):
    // start at given position
    for letter in string:
        // store pixel values to display letter on screen
    (these values will be held until update is called)
    // if string is too long, break loop
    update_area();
display_image(position, image_file):
    // start at given position
    // if image is too large, return error
    // store bitmap data to display image on screen
    (these values will be held until update is called)
    update_area();
```

4.1.3 General Validation

The E-Ink Display must be updated by code sent from the ESP32. This code uses a SPI connection, and the display uses a buffer to store pixel data until an update command is sent [1]. These displays also support partial screen updates, which allow us to write programs that only change portions of the screen at a time, reducing overall update time. Displaying both text and images allows our displays to be more aesthetically appealing.

4.1.4 Interface Validation

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**

e-ink_dsply_otsd_usrout : Output

Other: 4-Color Grayscale	This will allow us to add more depth and texture to the images and text we display, which will improve readability.	This is a feature of the E-Ink Display we will be using for this project. [1]
Type: Text, Images	This is a requirement of our design based on objectives set in our project document.	The E-Ink display is able to show both text and images.
Usability: Readable by 9/10 users from ≥10 ft	This is a requirement of our design based on objectives set in our project document.	The ability to control font size, as well as the overall size of the display, will allow us to maximize the ability of the average user to read what is on the display.

esp32_e-ink_dsply_comm : Input

Messages: Display Functions (Clear screen, display text, etc.)	The software must be able to take text strings and images, and convert them to pixel representations to be shown on the E-Ink display.	The code we will use for this project will have functions that display a given string of text based on a specified position and size.
Protocol: SPI	The display we have chosen for this project uses SPI to send and receive data [1].	The code we will use for this project will be designed with a SPI connection in mind.
Vnominal: 3.3V	This is the operating voltage of the ESP32.	Our power system is designed to supply this voltage to the ESP32 and the E-Ink Display.

4.1.5 Verification Process

- First, the display will be shown to operate when supplied with the given nominal voltage.
- The code that converts our text and images to pixel values will then be shown and walked through.
- The ability to display both a text string and an image using our software will be shown next. The SPI protocol will be demonstrated both within the code and with the hardware connection to the ESP32.
- The text will then be updated to show a new value.
- Finally, the results of a “vision test” given to several individuals will be shown. This will demonstrate the display’s readability.

4.1.6 References and Links

[1] Waveshare E-Ink Display Datasheet.

4.1.7 Revision Table

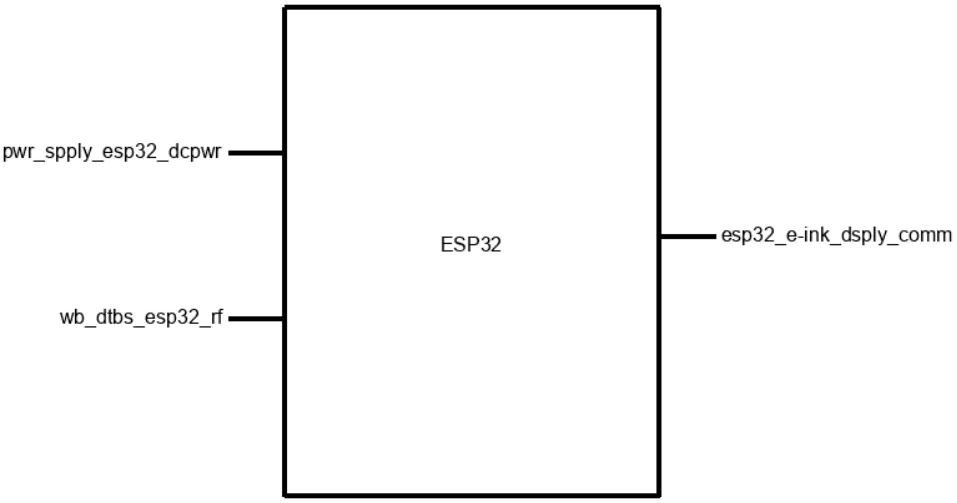
Date	**Editor(s)**	**Section(s) Edited**	**Description**
18 Feb 2022	A. Mote	2, 4	Added Block Diagram and refined interface properties.
4 Feb 2022	A. Mote	All	Wrote drafts of block design, interface details, and verification plan.

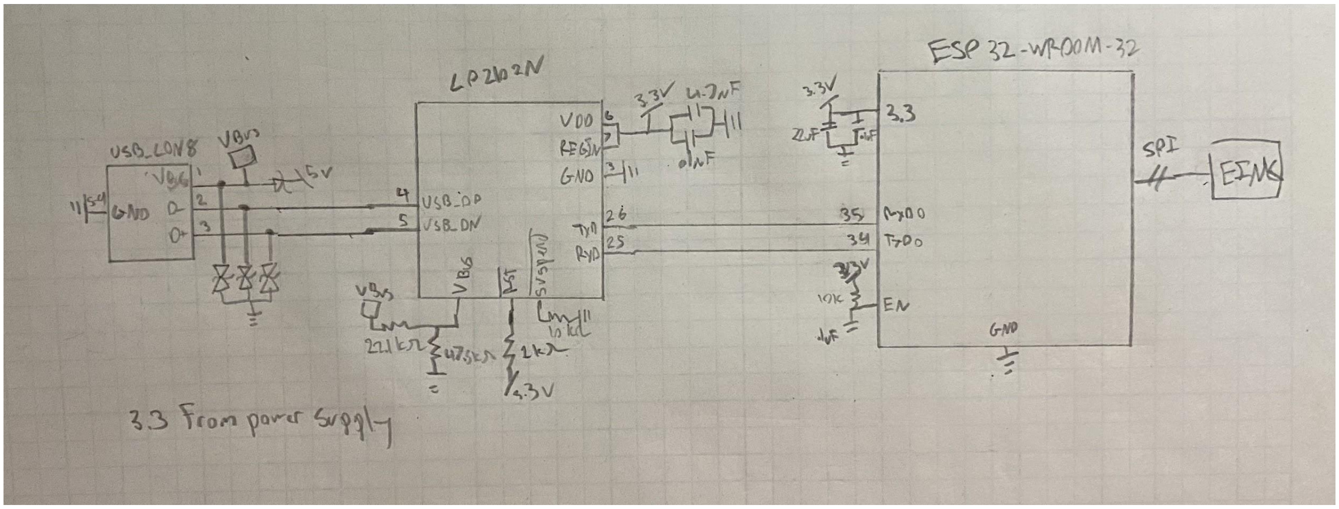
4.2 ESP 32

4.2.1 Description

The Esp32 block reads data from the web database over wifi, and then updates the e ink display using the data that was fetched.

4.2.2 Design





The schematic is based off of the devkit schematic provided by espressif[1]. It contains a micro usb plug, a usb to uart adapter chip, and the ESP32-WROOM-32E chip. This particular ESP32 model has a built-in wifi antenna and on board memory. The usb chip recommended by espressif is the CP2109 from silicon labs. The 3.3V VDD is provided from the power supply block. Data is pulled from the data based over Wi-Fi, using the onboard antenna. Data is then sent to the display over a SPI interface.

4.2.3 General Validation

The design of this block was chosen because it is a relatively simple and straightforward design, based on how the manufacturer recommends that the ESP32 chip be used.

4.2.4 Interface Validation

esp32_e-ink_dsply_comm : Output

Messages: Pixel Values (Location, 0-255 Greyscale Value, etc.)	Data that is sent to the display	
Protocol: SPI	SPI is the interface that the E Ink display uses	The ESP32 is capable of utilizing a SPI interface
Vnominal: 2.6V	The ESP32 has a high output voltage of 0.8*VDD, .8*3.3	VDD of 3.3V means output of 2.64 V

pwr_spply_esp32_dcpwr : Input

Inominal: 500mA	Minimum current to supply the chip with power, as found in the data sheet	
Ipeak: 1.5A	Above the minimum input current	
Vmax:3.5 V	Below absolute max power for the chip	Max power based on datasheet is 3.6V
Vmin: 3.1 V	Just above the minimum voltage required to power the chip	Minimum power based on datasheet is 3.0V

wb_dtbs_esp32_rf : Input

Messages: Text Strings (Room Number, Course Number, Class Name, etc.)	Data received from database	
Protocol: Wi-Fi	Allows ESP32 to access the online database that holds the data to be displayed.	ESP32 is a wifi capable chip

Data Rate: 1 Mbps	Device will receive data at a rate of at least 1 Mbps, Primarily transferring small amounts of text, infrequently, so high transfer rates not needed	1 Mbps is the lowest data rate specified in the datasheet

4.2.5 Verification Process

Wb_dtbs_esp32_rf and esp32_e-ink_dsply_comm

1. Setup test database
2. Run code that gets test data and then displays it
3. Verify that the correct data is displayed

Pwr_spply_esp32_dcpwr

1. Perform test above at 3.1V to verify minimum voltage
2. Perform test above at 3.5V to verify maximum voltage

4.2.6 References and Links

[1] "ESP32-devkitc," *Resource | Espressif Systems*. [Online]. Available: https://www.espressif.com/en/products/devkits/esp32-devkitc/resources. [Accessed: 07-Jan-2022].

[2] https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

[3] https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf

4.2.7 Revision Table

Ben Jones	1/21/22	Updated Verification Plan
Ben Jones	1/21/22	Updated Interface Verification
Ben Jones	1/7/22	Added General Validation
Ben Jones	1/7/22	Added References
Ben Jones	1/7/22	Added Verification plan

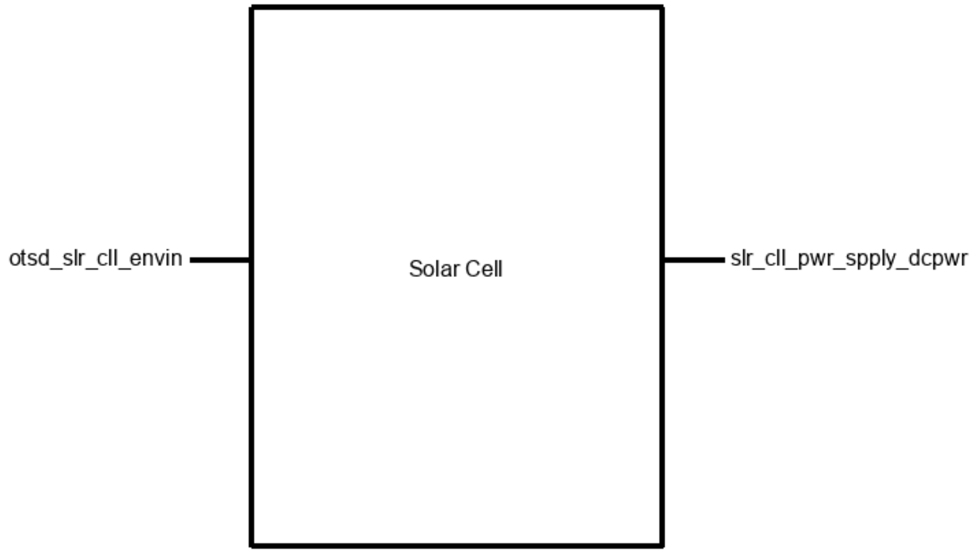
Ben Jones	1/7/22	Added Interface Validation
Ben Jones	1/7/22	Added block diagram, schematic, and schematic definition

4.3 Solar Cell

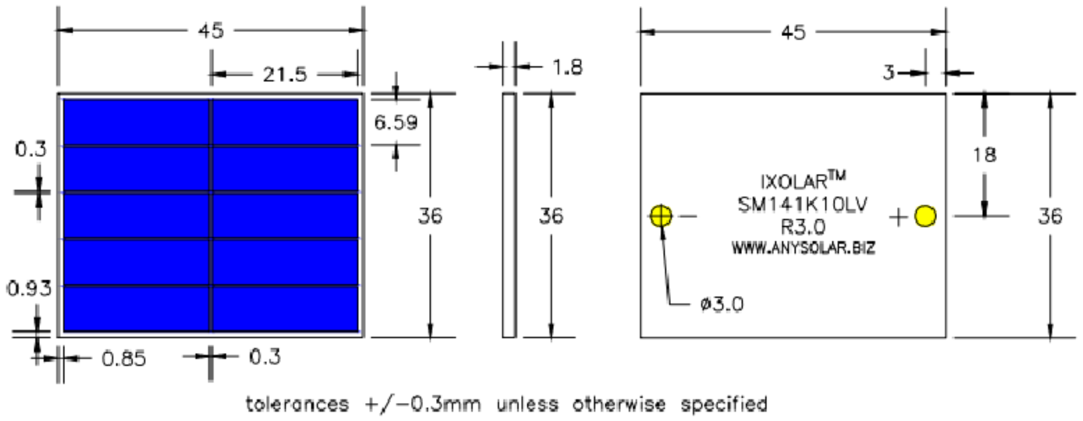
4.3.1 Description

The block champion for the solar panel is Carlos Gonzalez. The solar panel will be our main method of charging the battery contained in the system, in an effort to avoid the need for wired charging. The intended target location for the overall system will be on campus room doors. Thus, a method for wireless charging will be more practical for this purpose. The low-power idle nature of the ESP32 chip and E-Ink technology should allow the circuit ample time to use this panel to recharge between the moments of power consumption where the screen is updated. The expected size of the solar panel will be similar to that of a calculator’s solar panel. This should be sufficient to recharge the Li-polymer battery in reasonable time.

4.3.2 Design



Package front-side and back-side view. (dimensions in millimeters)



Front-side View details **Back-side View details**

4.3.3 General Validation

This block fulfills the role of a charging source for the battery in the IoT device. This device is intended to be attached on a flat surface like a door or wall near a campus room. Thus, it is more practical to wirelessly charge the battery with solar cells. A small array of solar cells will approximately be 2” x 1” which is sufficiently small to fit on an enclosure. This should result in a cost of under $10 usd which is low-cost for this system. An alternative solution for the block is a USB connection to charge the battery. However, this will not allow for wireless charging and may not fit the needs of the project partner.

4.3.4 Interface Validation

Otsd_slr_cll_envin: input

Interface Property	Why is this interface this value	Why do you know that your design details for this block above meet or exceed each property?
Light: Natural Sunlight 80-100 Lumens	This value was decided based on the average lumens produced on a sunny day.	According to the solar cell datasheet, the measure values were calculated using the sun as the light source sun (= 1000 W/m²)

Light: Artificial Light 15-30 Lumens	This value was decided based on the typical brightness of a classroom.	Based on our calculation, this value is sufficient in creating the expected charging voltage for the battery.
Temperature: 20-25°C	This value was based on the expected operating temperature.	According to the solar cell datasheet, the measure values were calculated using an environmental temperature of 25°C

Slr_cll_pwr_spply_dcpwr: output

Interface Property	Why is this interface this value	Why do you know that your design details for this block above meet or exceed each property?
Inominal: 55.1 mA	In this example, this value was chosen based on the needs of the overall system.	Based on our calculations, this value should be sufficient in charging the Li-Po battery in the system.
Ipeak: 58.6 mA	In this example, this value was chosen based on the operating conditions from the datasheet.	According to the solar cell datasheet, this value should be the highest expected current under ideal operating conditions.
Vpeak: 6.91 V	In this example, this value was chosen based on the operating conditions from the datasheet.	According to the solar cell datasheet, this value should be the highest expected voltage under ideal operating conditions.
Vnominal: 5.58 V	In this example, this value was chosen based on the needs of the overall system.	Based on our calculations, this value should be sufficient in charging the Li-Po battery in the system.

4.3.5 Verification Process

1. If the room has ceiling lights, may skip to step 4.
2. Set a light source, **Otsd_slr_cll_envin,** (i.e. LED bulb) of 45-90 lumens at a distance of 4-5 feet from the solar cell.
3. Direct the light source at the solar cell.
4. Place a lumen meter directly in front of the solar cell to ensure 15-30 lumens is being received. If not, return to step 2.
5. Measure the voltage and current (**slr_cll_pwr_spply_dcpwr**) at the output terminals of the solar cell.
6. Compare the results to the expected values.

4.3.6 References and Links

Solar cell datasheet,
https://waf-e.dubudisk.com/anysolar.dubuplus.com/techsupport@anysolar.biz/O18Ae0F/DubuDisk/www/Gen3/SM141K10LV%20DATA%20SHEET%20202007.pdf

Li-Po battery datasheet,
https://cdn-shop.adafruit.com/datasheets/LiIon2000mAh37V.pdf

4.3.7 Revision Table

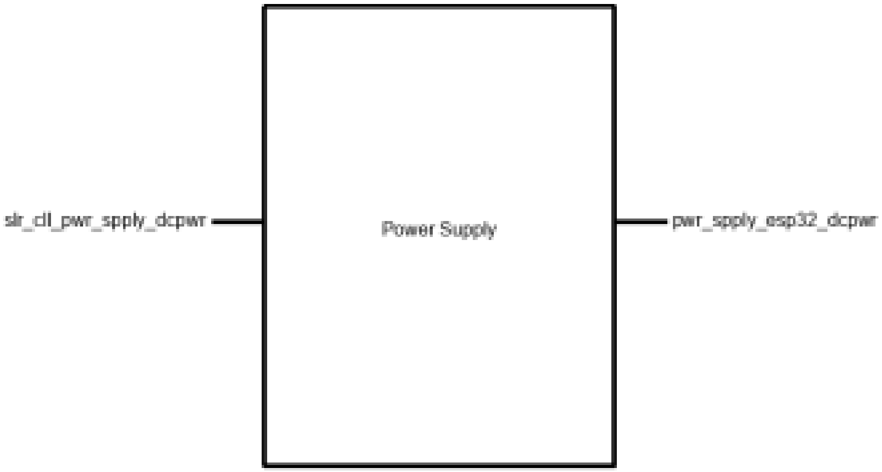
Date	Note	Recommender
2/18	Specified light source distance in verification plan	Jordan Porter
2/18	Included more details in the verification plan	Carlos Gonzalez
2/18	Changed lumen value to be range instead of an exact number.	Graham Donaldson
2/18	Expanded design section	Leif Miller
2/18	Simplified section values	Leif Miller
2/3/22	Modified Interface Property Values	Carlos Gonzalez
2/1/22	Modified Number of Interfaces	Carlos Gonzalez
1/31/22	Document Created	Carlos Gonzalez

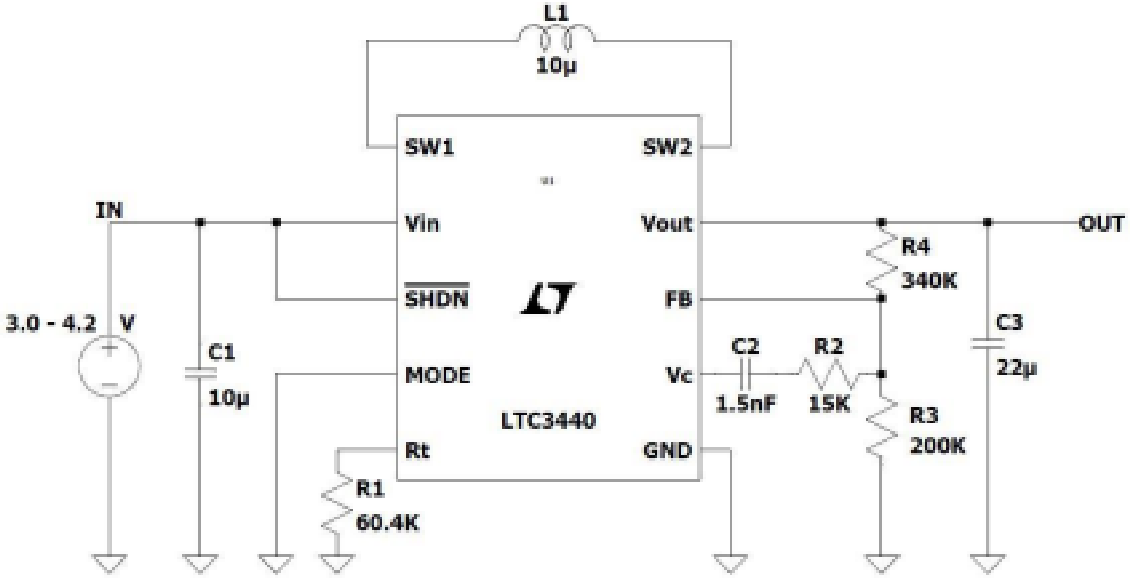
4.4 Power Supply

4.4.1 Description

The power supply is an essential DC to DC buck/boost converter. It will supply the ESP32 module and the E-Ink display with a constant voltage and current that each device requires. This allows the system to work at a consistent rate. The output voltage and current are determined by the max ratings for both devices. Output voltage will remain constant at 3.3V, whereas the output current will fluctuate, depending on the demands of the devices on a specific task. The power supply will be powered by a rechargeable 3.7V Lipo battery.

4.4.2 Block Design





4.4.3 Block General Validation

This block will require a DC-to-DC converter to ensure constant output voltage that the ESP32 module and E-Ink display requires to operate. The power from the battery will vary over time as the battery loses power. This varying of power isn't ideal to keep both of our devices in operating conditions. The power supply will help with this issue, stepping down our voltage when needed and stepping up when needed. It is a crucial part of the system and will require a fine understanding of power electronics to accomplish. The power supply is merely one component that will be mounted on our PCB design that will help regulate our power.

4.4.4 Block Interface Validation

pwr_spply_dcpwr: Input

Inominal: 240mA	This nominal current was chosen based on the expected current needs of the system.	The datasheet sets the standard discharge at constant current of 0.2CA. With a capacity of 1.2A, we get 240mA.
Ipeak: 1A	This peak current was chosen based off the max current discharge of the battery and max current input of the power supply.	The datasheet sets the max continuous discharge current at 1CA, and with a capacity of 1.2A we get 1.2A and the max input current of the LTC3440 is 1A. The battery can supply more than the max current of the LTC3440, so the peak current will be limited by the power supply at 1A.
Vmax: 4.2V	This max voltage was chosen based on a fully charged battery	A fully charged 3.7V Lipo battery is rated at 4.2V. This will be the max voltage our power supply will have to handle.
Vmin: 3.0V	This minimum voltage was chosen based on the voltage the battery experiences at the end of discharge	The 3.7V Lipo battery is rated at 3.0V at the end of its discharge cycle.

pwr_spply_dcpwr: Output

Inominal: 35mA	This nominal current was chosen based on the expected current needs for the ESP32 chip and the E-Ink display during low level work.	The ESP32 low-level sink current is rated at 28mA and the E-Ink display during standby mode is 5uA.
Ipeak: 600mA	This peak current was chosen based on the required max currents of each device.	The ESP32 chip is rated to operate at 500mA. The E-Ink display is rated to require a max of 10mA when updating an image.
Vmax: 3.6V	This max voltage was based on the max operating voltage for the ESP32 chip.	The ESP32 module is rated to operate at a max voltage of 3.6V and will not expect to surpass this value. The E-Ink display is rated to operate from 3V-5.5V, so limiting max voltage to 3.6 is needed for the ESP32 module.
Vmin: 3.0V	This min voltage was based on the min operating voltage of the ESP32 chip and E-Ink display.	The ESP32 module and E-Ink display are both rated to operate at a minimum voltage of 3.0V and will not expect to fall below this value.

4.4.5 Block Testing Process

1. Connect battery to positive and negative terminals for power supply
2. Measure input voltage and current from the battery with DMM/power supply. 3.
Check voltage and current matches the min and max requirements for battery supply. 4.
Measure output voltage and current with electronic load.
5. Check voltage and current matches the min and max requirements output load.

4.4.6 References and File Links

"Adjustable and fixed low drop positive voltage regulator." [Online]. Available: https://www.st.com/resource/en/datasheet/ld1117.pdf. [Accessed: 20-Jan-2022].

"ESP32C3MINI1 ESP32C3MINI1U - Espressif." [Online]. Available:

http://www.espressif.com/sites/default/files/documentation/esp32-c3-mini
1_datasheet_en.pdf. [Accessed: 20-Jan-2022].

“Shenzhen PKCELL Battery Co., Ltd Li ... - Adafruit Industries.” [Online]. Available: https://cdn shop.adafruit.com/datasheets/LiIon2000mAh37V.pdf. [Accessed: 04-Feb-2022].

4.4.7 Revision Table

Emanuel Murillo	Updated section 4.1.1, 4.1.2, and 4.1.4	2/5/22
Emanuel Murillo	Updated section 4.1.4	2/3/22
Emanuel Murillo	Each section was added to	1/21/22

4.5 Web Database

4.5.1 Description

The web database is used to store user inputs online in a way that can be accessed by the ESP-32 chip, in order to obtain the information that will be shown on the E-Ink Display. This database will be written to by the user interface, with SQL code being used to modify the database according to the user’s inputs. It will be read wirelessly by the ESP-32 using the CircuitPython code flashed onto the chip.

4.5.2 Design

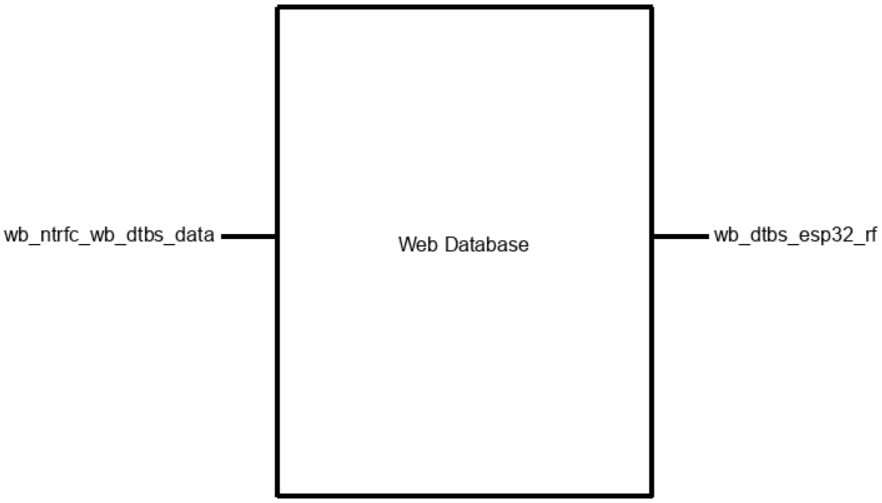


Figure 1: Block diagram.

Pseudocode:

```
time, room, crn, name = collect_user_input();
database = open(“database.csv”);
if(time not in database.times):
    database.add_row(time);
database.add_vals(time, room, crn, name);
database.close();
```

4.5.3 General Validation

This design was chosen because the ESP32 is most easily coded using CircuitPython scripting; since this hardware interfacing is the most difficult part of the coding process, this somewhat bodged together database structure was made to facilitate this interfacing, rather than making the database simpler and the hardware interfacing more difficult.

4.5.4 Interface Validation

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
wb_dtbs_esp32_rf : Output		
Datarate: 1 Mbps	This is the minimum Wi-Fi transfer speed of the ESP32.	The OSU servers, which will store the database, easily exceed this value; therefore, the value is dictated by the ESP32.

Messages: Text Strings (Room Number, Course Number, Class Name, etc.)	These text strings will be read in by the CircuitPython program and displayed on the E-Ink Display.	Transferring text strings from the database to the ESP32 is handled by the code demonstrated above.
Protocol: FTP	This is the standard for file transfer over TCP/IP connections.	Connection and transfer of database values to the ESP32 is handled by the code shown above.

wb_ntrfc_wb_dtbs_data : Input

Datarate: Time to Update Database: 10 seconds	This value is set high to ensure that the database has time to update over HTTP connections. It may be lowered in the future.	Transfer of information over HTTP typically takes on the order of milliseconds, so this allows for a large margin of error and even multiple transmission attempts.
Messages: Database Values	The database needs to be able to process and use any given user input appropriately.	The function that collects user input will process these values appropriately.
Protocol: FTP	This is the standard for file transfer over TCP/IP connections.	Connection and transfer of data between web pages is handled by the code demonstrated above.

4.5.5 Verification Process

1. A test interface will be used to add, modify, and remove values in the database.
2. These changes will be demonstrated on a developed board (the Adafruit Magtag) and show how they are displayed on the E-Ink screen.

4.5.6 References and Links

No references or links were used in this section.

4.5.7 Revision Table

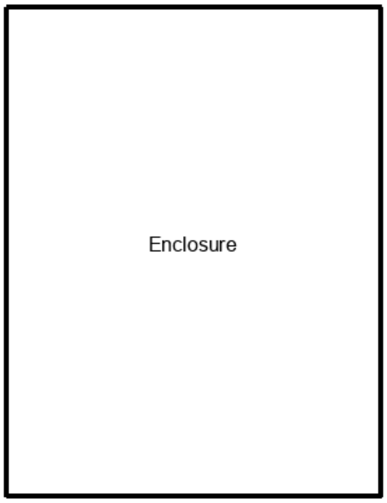
Date	**Editor(s)**	**Section(s) Edited**	**Description**
21 Jan 2022	A. Mote	2, 4, 5	Added pseudocode, interface properties, and modified verification plan.
8 Jan 2022	A. Mote	All	Wrote drafts of block design, interface details, and verification plan.

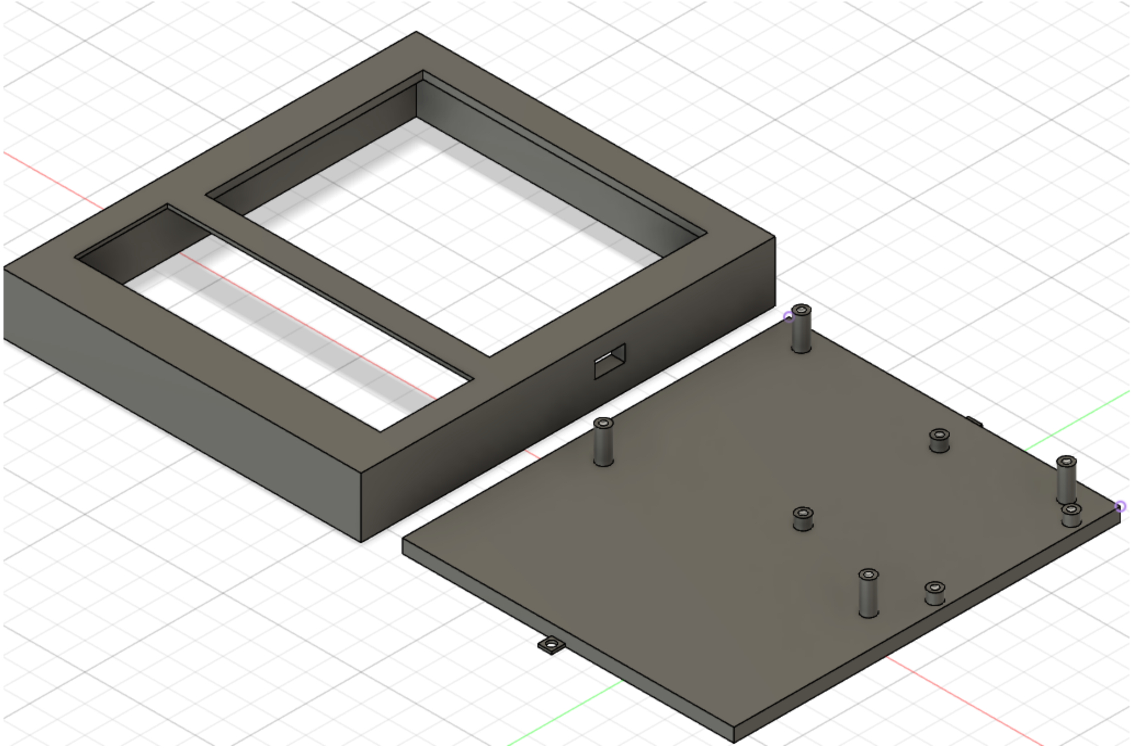
4.6 Enclosure

4.6.1 Description

The Enclosure holds all of the other blocks, while allowing the display to be read, and the solar panel to get light. It also needs to be designed in such a way that it can be hung on a wall.

4.6.2 Design





The enclosure will be 3D printed using PLA plastic. It will be sized based on the dimensions of the display which is approximately 3.5 inches by 3 inches. The different components will be connected to a flat back plate with a cover that contains holes for the display and solar cell.

4.6.3 General Validation

The main purpose of this block is to house the electrical components of our project. By using PLA plastic to 3D print this block, all of the requirements will be met. PLA is fine to use at the operating temperatures of all the components, and is sturdy enough to survive a small fall. Windows can be added to allow the display to be seen ,and for the solar panel to receive light.

4.6.4 Interface Validation

The Enclosure doesn't have any interfaces with other blocks.

4.6.5 Verification Process

1. Show that the different blocks fit in the enclosure
2. Show that the mounting hardware is the correct size

4.6.6 References and Links

4.6.7 Revision Table

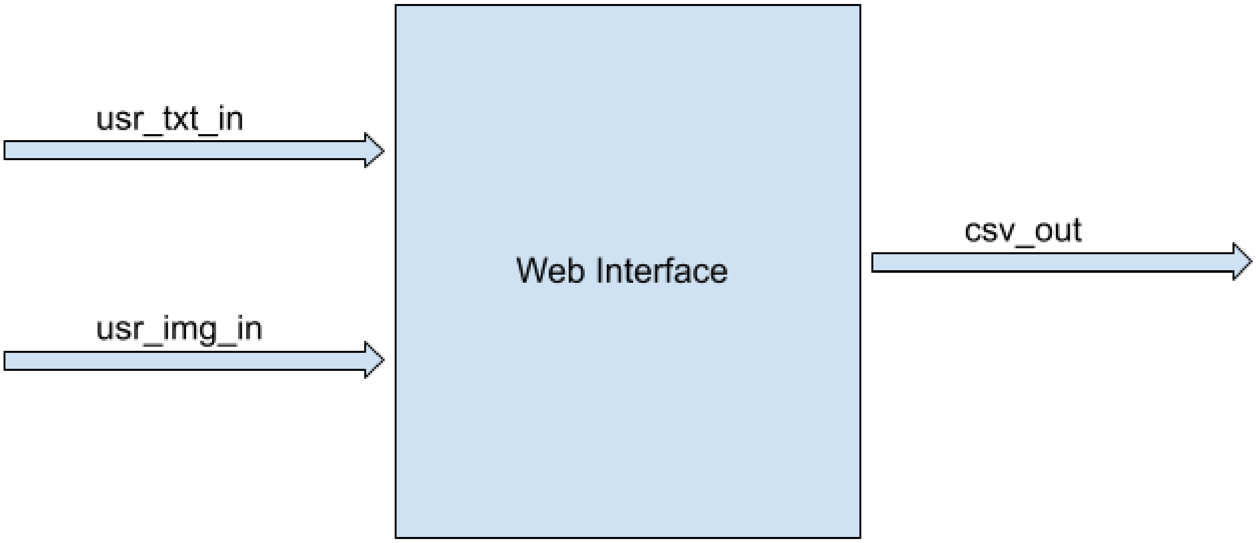
2/18/22	Ben Jones	Updated Design
2/18/22	Ben Jones	Updated General validation
2/18/22	Ben Jones	Updated Interface validation
2/4/22	Ben Jones	

4.7 Web Interface

4.7.1 Description

The champion for this block is Carlos Gonzalez. This block serves as an interface that allows users to connect with the E-Ink Display. The interface will be hosted on a website and can be accessed via public domain name. This block is necessary for updating the information displayed on the screen.

4.7.2 Design



4.7.3 General Validation

This design meets the criteria of the system because it connects the user with the E-Ink display in a simple manner. The user submits text and/or images to be displayed on the display. The display retrieves the user's information from an online database. For this reason, the web interface outputs a csv file containing the submitted information from the user that will be sent to the online database.

4.7.4 Interface Validation

usr_txt_in: input

text	Text is an essential	Text will allow users to write

	component in relaying information on a digital display.	messages or data to the display.

usr_img_in: input

.jpg, .png	Images are an essential component in relaying information on a digital display	Images allow users to display important visualizations.

csv_out: output

.csv	csv files are useful for storing data such that it can parsed efficiently.	csv files contain pixel values that the display will need to show the user's submitted information.

4.7.5 Verification Process

1. Be able to enter text into a textbox on the web page.
2. Be able to upload images.
3. Produce a csv file containing the text and images.

4.7.6 References and Links

No references or links were used in this section.

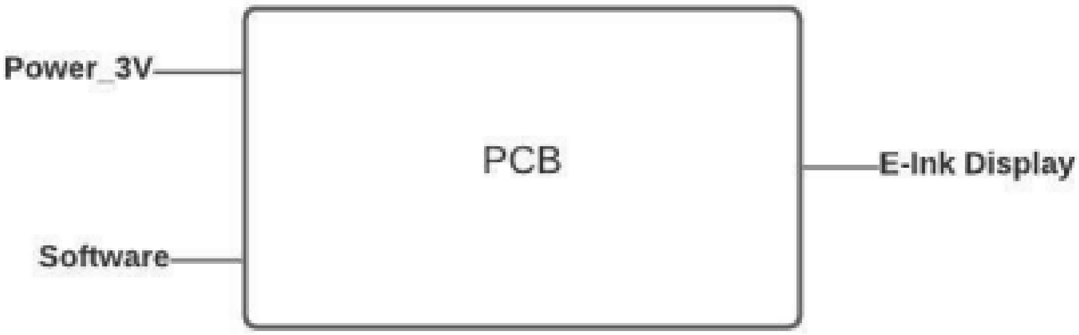
4.7.7 Revision Table

4.8 PCB

4.8.1 Description

The PCB will connect all individual circuit elements and ensure that they can all interface with one another successfully. A custom-designed PCB will be used to accommodate our project's unique components and engineering requirements.

4.8.2 Block Design



4.1.3 Block General Validation

The PCB will be the most complicated piece of hardware in our design. This block will be responsible for connecting all individual blocks and allows them to communicate with one another. The main components the PCB will contain is the wi-fi chip, esp32 chip, power supply, and E-Ink Display. There will be a halt on most manufacturing in China in the next couple weeks, finishing this design is important to avoid this added time delay. It will take the longest to order and receive compared to any of the other parts in our design. As we want to ensure that our design is working correctly and the best design we can implement, this might be our most expensive part as well since we will pay for express shipping if needed. We will try and finish before the week to 2- week halt on manufacturing but won't order if it's not done before then.

4.8.4 Block Interface Validation

Test points will be added to the design in order to demonstrate the validations of other blocks of the system such as the power supply, esp32, e-ink display, ect..

4.8.5 Block Testing Process

1. System must be operational once receiving power from battery
2. Display must show information
3. Display must be able to be updated from personal computer

4.8.6 References and File Links

4.8.7 Revision Table

Emanuel Murillo	Section 4.1.3 added to	1/21/22
Emanuel Murillo	Each section was added to	1/7/22

4.9 Revision Table

Date	Editor(s)	Section(s) Edited	Description
30 Apr 2022	A. Mote	4.1, 4.5	Added information from Block Validation documents
3/6	Carlos Gonzalez	4	Created Section 4 and subsection headers

5. System Verification Evidence

5.1 Universal Constraints

5.1.1 The system may not include a breadboard

Our system was designed and built on a custom PCB.

5.1.2 The final system must contain both of the following: a student designed PCB and a custom Android/PC/Cloud application

Our final system was built on a custom PCB and the online webpage was also a custom design by the team.

5.1.3 If an enclosure is present, the contents must be ruggedly enclosed/mounted as evaluated by the course instructor

The internal hardware is protected inside a custom enclosure. The PCB is mounted to the enclosure using screws to maintain stability and durability of the internal hardware.

5.1.4 If present, all wire connections to PCBs and going through an enclosure (entering or leaving) must use connectors

The system will include a USB port that will be soldered onto the PCB. A USB wire will be able to connect to the port. There are no other additional wires entering or leaving the enclosure.

5.1.5 All power supplies in the system must be at least 65% efficient

The power supply design utilized in our design utilizes a fixed frequency setting and a burst mode setting while under light loads. The efficiency at 3.3V output at a fixed frequency of 300kHz is 96% when drawing 50-60mA. Under burst mode, the power supply will experience an efficiency of 75%-79%, which our system will primarily operate at. This meets our 65% efficiency for our power supply.

5.1.6 The system may be no more than 50% built from purchased modules

There are 8 total blocks that finalize the system. The PCB, enclosure, web interface, and the web database are all custom designs by the team. Therefore, our system is no more than 50% built from purchased modules.

5.2 Display and Readability

5.2.1 Requirement

The completed circuit design must interface with and utilize an E-Ink Display. Any text shown on the E-Ink Display must be identifiable (i.e. legible) from a distance of at least 5 feet by at least nine out of ten testers.

5.2.2 Testing Process

1. The completed circuit design will consecutively send and display twenty different elements of data on the E-Ink Display.
2. At least nine out of ten testers will be able to correctly read these displays from a distance of at least 5 feet.

5.2.3 Testing Evidence

All usability test results can be found in the Usability Test Spreadsheet [1]. Test success was measured to be above 90 percent.

5.3 Internet of Things

5.3.1 Requirement

The completed circuit design must be able to connect to and read data from the Internet.

5.3.2 Testing Process

1. The completed circuit design will connect to the Internet.
2. Requirements will be verified when the completed circuit design consecutively retrieves twenty different elements of data from an online database and displays them on the E-Ink Display.

5.3.3 Testing Evidence

Evidence of the system displaying elements of an online database can be seen in this video [2].

5.4 Power Efficiency

5.4.1 Requirement

The completed circuit design must have a low-power idle state. The system must be able to run for a week's worth of updating the label without requiring charge.

5.4.2 Testing Process

1. The completed circuit design will draw no more than 200 mW of power when not updating the E-Ink Display (in idle mode).
2. The completed circuit design will have its battery charge level measured over a 1/2 hour period of average expected use.
3. Requirements will be verified when the battery charge level is maintained at a constant value over a two hour period of average expected use..

5.4.3 Testing Evidence

Using a digital power meter will be able to measure the amount of power consumed to our system. Being able to measure the amount of power consumed during idle state and when the system is being updated. Planning a schedule of updates within a ½ hour period, we can extrapolate this data to meet our one week requirements.

5.5 Rechargeability

5.5.1 Requirement

The completed circuit design must be able to recharge the battery that is powering the system.

5.5.2 Testing Process

1. An external power source must be able to connect to our system via usb or solar panel.
2. The completed circuit design will convert power received into 4.2V and no more than 175mA.
3. Once the battery is fully charged, charging will be paused to protect the battery

5.5.3 Testing Evidence

Solar cell charging requires additional testing and development to meet this requirement; currently, the system does not support solar cell charging, but can be charged via USB cable. We were able to implement a deep sleep functionality in the software, and the ESP32 datasheet shows that the microcontroller consumes less than 200 microAmps of current in this mode; however, we were unable to produce a measurement from our final PCB that confirms this value. Further work is required to complete this requirement.

With our new solar cell, we are able to produce the required voltage required to charge the battery. A power supply is required to produce a stable voltage and current for the charging. Solar-charging ic’s are available and will be utilized to accomplish this. USB charging was designed and tested but not incorporated into our current PCB. A new PCB was designed and ordered that included this feature but was unable to arrive prior to system verification. Latest design of our PCB will be able to properly demonstrate the power requirements for power and portability.

5.6 Updating the Display

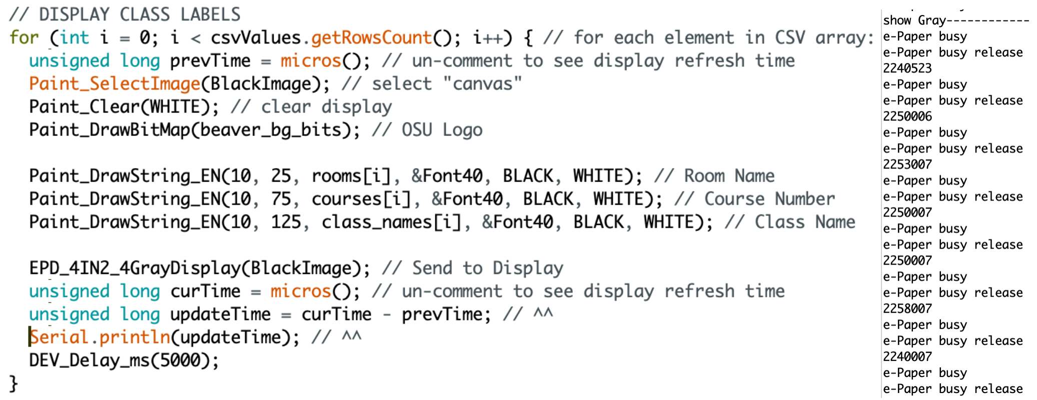
5.6.1 Requirement

The E-Ink Display updating process will take no more than five seconds.

5.6.2 Testing Process

1. The E-Ink Display will be timed on both startup and updating; five different updates will be sent. Timing will start when the screen begins to change, and end when the change is complete.
2. The startup and all updates must be completed and displayed on the E-Ink Label in less than five seconds.

5.6.3 Testing Evidence



```
// DISPLAY CLASS LABELS
for (int i = 0; i < csvValues.getRowsCount(); i++) { // for each element in CSV array:
  unsigned long prevTime = micros(); // un-comment to see display refresh time
  Paint_SelectImage(BlackImage); // select "canvas"
  Paint_Clear(WHITE); // clear display
  Paint_DrawBitMap(beaver_bg_bits); // OSU Logo

  Paint_DrawString_EN(10, 25, rooms[i], &Font40, BLACK, WHITE); // Room Name
  Paint_DrawString_EN(10, 75, courses[i], &Font40, BLACK, WHITE); // Course Number
  Paint_DrawString_EN(10, 125, class_names[i], &Font40, BLACK, WHITE); // Class Name

  EPD_4IN2_4GrayDisplay(BlackImage); // Send to Display
  unsigned long curTime = micros(); // un-comment to see display refresh time
  unsigned long updateTime = curTime - prevTime; // ^^
  Serial.println(updateTime); // ^^
  DEV_Delay_ms(5000);
}
```

```
show Gray------------
e-Paper busy
e-Paper busy release
2240523
e-Paper busy
e-Paper busy release
2250006
e-Paper busy
e-Paper busy release
2253007
e-Paper busy
e-Paper busy release
2250007
e-Paper busy
e-Paper busy release
2250007
e-Paper busy
e-Paper busy release
2258007
e-Paper busy
e-Paper busy release
2240007
e-Paper busy
e-Paper busy release
```

Figure XX: Screenshots of timekeeping code and serial output from ESP32. Numbers in serial output represent time needed for each screen update, in microseconds. Average update time is 2.25 seconds.

5.7 Speed of Setup

5.7.1 Requirement

9 out of 10 end users will be able to set up the E-Ink Display without need for any programming or electrical knowledge.

5.7.2 Testing Process

1. Test users will be shown how to set up the E-Ink Display by connecting it to the OSU Access network. This process will be demonstrated by a project team member.
2. Users will then be asked to set up the system themselves with no input from the project team member. Set up is defined as getting the display connected to a database and displaying an element from that database.
3. Requirements will be verified if nine out of ten users are able to set up the display in five minutes or less.

5.7.3 Testing Evidence

All usability test results can be found in the Usability Test Spreadsheet [1]. Test success was measured to be above 90 percent.

5.8 Ease of Use

5.8.1 Requirement

9 out of 10 end users will be able to update the E-Ink Database without need for any programming or electrical knowledge.

5.8.2 Testing Process

1. Test users will be shown how to modify the E-Ink database by a member of the project team.
2. Testers will then be asked to modify the display database themselves with no input from the project team member (changing a course number, the time a class will start/end, etc.)
3. Requirements will be verified when nine out of ten users are able to update the database in sixty seconds or less.

5.8.3 Testing Evidence

All usability test results can be found in the Usability Test Spreadsheet [1]. Test success was measured to be above 90 percent.

5.9 Enclosure

5.9.1 Requirement

The completed design must be fully enclosed to prevent tampering and damage from bumping or contact sustained during expected use.

5.9.2 Testing Process

1. The completed circuit design will be fully enclosed, with the exception of the display screen.
2. The combined enclosure and circuit design profile will be no larger than 6.5” x 5.5” x 2”.
3. The enclosure will be able to withstand light physical contact, such as bumping, knocking, tapping, and pressing, without any loss of functionality.

5.9.3 Testing Evidence

Evidence was presented in a live demonstration.

5.10 References and File Links

5.10.1. References

“LTC3440 - Analog Devices.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/3440fd.pdf. [Accessed: 04-Mar-2022].

5.10.2. File Links

[1] Usability Test Spreadsheet.

[2] Internet of Things Testing Evidence Video.

5.11 Revision Table

Date	Editor(s)	Section(s) Edited	Description
30 Apr 2022	A. Mote	5.2, 5.6, 5.7	Added testing evidence for aforementioned sections.
3/6	Emanuel Murillo	5.1.5, 5.10	Filled in section 5.1.5 and 5.10
3/6	Carlos Gonzalez	5.1 - 5.11	Created Section 5 and partially filled in required subsections

6. Project Closing

6.1. Future Recommendations

6.1.1. Technical Recommendations

The area of this project that requires the most technical modification at this point is solar charging. We selected solar cells designed for indoor use, which were optimized to draw power from low-light conditions. However, these cells did not seem to draw enough current to send charge to the battery used in our design. A different battery with a lower current threshold may be able to support these solar cells; otherwise, a new solar cell may be required.

In order for this design to scale to multiple rooms and real-time updating, the demonstration code will also need to be updated. Currently, the code simply iterates through all elements of the database, arranged by time. The scalable version of this code would likely need a room number assigned to each label, which it would then use to select only the elements of the database that contain that room number. Additionally, in order to maximize battery life, the ESP32 would need to go into sleep mode after each update, for the length of the class/event in the room, reactivating to update only when the next event is about to occur.

6.1.2. Global Impact Recommendations

One of the impacts we noticed during testing is that the display has some accessibility issues. In addition to being an English, text-only display, which were elements we had already accepted as limitations of the design's scale, we noticed that individuals with conditions such as dyslexia had issues reading the screen. This was something we had attempted to account for by spacing out the letters in each word, so that the individual letters were more distinct; however, a font designed to further distinguish each letter may help improve this accessibility concern.

6.1.3. Teamwork Recommendations

One of the biggest pitfalls we experienced this term was finding a reliable way to track project progress. We originally attempted to use Trello for this; however, our group was largely unfamiliar with this program, and shied away from using it in favor of other, more familiar forms of communication. This caused some issues in both collaboration and understanding, and hurt the overall quality of our project in the long run. Taking the time to learn a project management system would have allowed us to better keep track of progress and objectives, mitigating these issues.

6.2. Project Artifact Summary

System Microcontroller Code
User Interface Code
Enclosure Diagram
PCB Files

6.3. Presentation Materials

Expo Poster
Project Showcase