Project Management and Design of a Modularized Internet of Things (IoT) Prototyping System

by

Tyler J. Inberg

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Electrical and Computer Engineering
(Honors Scholar)

Honors Baccalaureate of Science in Innovation Management
(Honors Scholar)

Presented May 22, 2018

Commencement June 2019

AN ABSTRACT OF THE THESIS OF

Tyler J. Inberg for the degree of Honors Baccalaureate of Science in Electrical and Computer Engineering and Honors Baccalaureate of Science in Innovation Management presented on May 22, 2018. Title: Project Management and Design of a Modularized Internet of Things (IoT) Prototyping System.

Abstract approved: ____

Chet Udell

Project Loom is a system of sensors, actuators, processors, and wireless connectivity chips that the user can pick and choose from in order to create an Internet of Things application that fits their desired needs. The modularity of the system and ease of use allows for the user to create their application without high costs or programming and soldering experience. This thesis begins with a description of the overall project and the testing validation that was used on the system, and then delves into the project management that went into the process from conception to the final system as well as details the design specifications, validation and integration for the I^2C sensors and enclosure that were used for this project. The end of the document focuses on the case studies that the project has been used for this year as well as potential future applications that could be explored. The information contained in this document should provide the reader with insight into the inner workings of the I^2C sensor block and enclosure as well as the project management process used so that they can recreate and even build upon the author's portion of Project Loom.

Key Words: Engineering, Project Management, Project Development, Internet of Things, IoT

Corresponding E-mail Address: inbergt@oregonstate.edu

©Copyright by Tyler J. Inberg

May 22, 2018

All Rights Reserved

Project Management and Design of a Modularized Internet of Things (IoT) Prototyping System

by

Tyler J. Inberg

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Electrical and Computer Engineering
(Honors Scholar)

Honors Baccalaureate of Science in Innovation Management
(Honors Scholar)

Presented May 22, 2018

Commencement June 2019

Honors Baccalaureate of Science in Electrical and Computer Engineering and Honors Baccalaureate of Science in Innovation Management project of Tyler J. Inberg presented on May 22, 2018.

APPROVED:

Chet Udell, Mentor, representing Biological and Ecological Engineering

---

Stephen Good, Committee Member, representing Biological and Ecological Engineering

---

Kirsten Winters, Committee Member, representing Electrical Engineering and Computer Science

---

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, University Honors College. My signature below authorizes release of my project to any reader upon request.

Tyler J. Inberg, Author

Table of Contents

I. INTRODUCTION

With the rise of research and development in the field of the Internet of Things (IoT), the industry is poised to increase connected IoT devices fivefold [1]. Companies all over the world are getting ready to support this influx with the implementation of 5G, the cost reduction in microcontrollers and sensor technology, and the open source code that is being created and expanded upon all over the internet; however, if the means to create these IoT applications isn't easily accessible by the masses, then the adoption and growth of this technology will be unnecessarily slow and the barriers of entry will restrict expansion of this technology to those with extensive engineering and software design experience.

This thesis follows the fulfillment of an Electrical and Computer Engineering Senior Capstone Project titled Project Loom designed by three senior capstone students, Kenny Noble, Dongjun Lee, and Tyler Inberg, and delves into the blocks that were specifically designed by the author. The document has been created to convey information about the top-level system, interfaces, product research and related engineering decisions, project management, the project timeline that was followed, integration of I^2C sensors, design and creation of the enclosure, and the design and validation of each block. By the end of document, the audience should have an understanding of what Project Loom achieves, the three sections that were created by the author in order to accomplish that vision, the resources that were referenced during the design process, and potential future applications of the system.

II. PROJECT OVERVIEW

Project Loom was developed by a team of Oregon State University ECE capstone students in conjunction with a team of OSU CS capstone students. The main objective of Project Loom was to create a modular IoT learning and development platform to allow anyone, even those without a strong technical background, to quickly and easily create IoT projects. The platform consists of a suite of modular IoT sensors and actuators that can be configured with little to no soldering or programming. IoT sensors and actuator "bricks" connect to each other in an intuitive manner so that users with little technical experience can work with them. Project Loom also utilizes a graphical user interface to decrease the barrier of entry for users to create their own IoT projects. It was important to ensure that the platform is highly accessible to allow a wide variety of users from diverse backgrounds and experience levels to work with it.

One of the main requirements of Project Loom was retrieving data from a variety of sensors. Users are able to use the sensors over a variety of distances and in both open and closed network configurations. To allow the IoT sensors to be used in such a wide variety of contexts, they support multiple types of wireless communication and wireless protocols. By ensuring that the product works with a variety of wireless communication processes, in this case LoRa, nRF, and WiFi, users can manipulate the system in order to utilize the wireless capabilities that best fit their needs. This will be accomplished by programming the system to support data transfer through the Adafruit Feather M0 WiFi board, the Adafruit Feather M0/32u4 with RFM95 LoRa Radio board, and the Adafruit Feather M0/32u4 with nRF24L01+ board.

A. System Requirements

At the beginning of the project, the team worked with the Stakeholder to produce a set of 10 system requirements that had to pass at the end of the school year in order for the project to pass. These system requirements were split up into four main areas: wireless requirements, peripheral requirements (sensors and actuators), assembly requirements, and hardware requirements. Below is a list of all the requirements that the team produced with the requirements that are addressed in the author's block bolded.

Wireless Requirements:

1. The system will be configurable to transmit data through WiFi at rates of 500 Kbps over short distances up to 30 meters.
2. The system will either create its own WiFi access point to transmit data or will connect to preexisting WiFi access points to transmit data.
3. The system will be configurable to transmit data through Nordic nRF24L01+ (nRF) at transmission rates of 250 Kbps over mid-range distances up to 100 meters using either the Adafruit 32u4 or Adafruit M0 processors.
4. The system will be configurable to transmit data through LoRa over distances up to 0.25mi in dense biomass (forest) to between 2 and 26 kilometers open-air "line-of-sight" using either the Adafruit 32u4 or Adafruit M0 processors.

Peripheral Requirements:

5. **The system will measure light (10 lux - 25,000 lux with 5% accuracy or 10 lux whichever is greater), weight (0-11.24 lbf with 5% accuracy or .5lbf whichever is greater), distance (10mm-200mm with 5% accuracy or 2mm whichever is greater), and humidity (50-100rh with 5% accuracy or 10rh whichever is greater) using either the Adafruit 32u4 or Adafruit M0 processors.**
6. **The system will use (when configured) a gyroscope (-2 to 2 rad/s) and an accelerometer (-10 to 10 m/s^2) sensor to measure 3-axis positioning and acceleration using either the Adafruit 32u4 or Adafruit M0 to process the data**
7. The system will use either the Adafruit 32u4 or Adafruit M0 processor to process the code which will position a servo between 0 and 170 degrees.
8. The system will measure apparent dielectric permittivity (1-80) and water temperature (0-37 C) using the sensors to measure the data at the minimum and maximum cases and using either the Adafruit 32u4 or Adafruit M0 to process the data.
9. **The system collects data from 8 I^2C devices with the same I^2C address.**

Assembly Requirements:

10. **The hardware shields will connect together without soldering connections for ease of swapability; 9/10 people can connect and disconnect at least 4 shields by hand, collect data with one shield, and state current measurement conditions in a 10 minute period.**

B. Testing System Requirements

Each system requirement required a testing procedure that would either pass or not pass depending upon if all the conditions were met. This way, at the end of the project, there was a measurable way to test if all requirements had been completed or not. **Appendix A** lists all of the system level testing procedures that were created and completed by the author.

The first test that was conducted was the Light, Pressure, Temperature, and Humidity Test and was designed to confirm that Requirement 5 as listed above had been completely fulfilled at the end of the project. The test consisted of checking the low and high range of the lux, pressure, temperature, and humidity sensor and ensuring that the readings were within 5% of the value being recorded by the provided measuring devices outside the system. A video documenting the entire test can also be accessed at https://tinyurl.com/i2c-system-test.

The second test that was conducted was the Accelerometer and Gyroscope Test and was designed to confirm that Requirement 6 as listed above had been completely fulfilled by the end of the project. The test consisted of connecting the FXOS/FXAS breakout sensor to the system, measuring the accelerometer and gyroscopic data being produced by the sensor, and comparing the results with the values that were expected. If the values matched the expected range that was set forth in the requirement, then the test passed and the requirement was considered completed. A video of the test can be accessed at https://tinyurl.com/accel-gyro-system-test.

The third test that was conducted was the I^2C Multiplexer Test and was designed to confirm that Requirement 9 as listed above had been completely fulfilled by the end of the project. The test consisted of connecting eight sensors simultaneously with the same address, in this case 8 lux sensors were used, and identifying if all eight sensors were collecting measurements and sending the data to the receiver. The test passed if the receiver was receiving data from all eight sensors. A video of the test can be accessed at https://tinyurl.com/i2c-multiplexer-system-test.

The fourth test that was conducted was the Assembly Test and was designed to confirm that Requirement 10 as listed above had been completely fulfilled by the end of the project. The test consisted of having ten participants connect and disconnect four different shields to the provided processor and then connect the processor to the overall system and start collecting data. The test passed if at least 9 out of 10 of the participants were able to successfully connect and disconnect the shields as well as collect data from the provided sensor within 10 minutes. A video of the test can be accessed at https://tinyurl.com/assembly-system-test.

C. Block Diagram and Overview of System Architecture

The final black box design of Project Loom consisted of six input interfaces and two output interfaces (see **Figure 2.1**). The input interfaces that correlate to the author's blocks are the **outside_enclosure_usrin, outside_enclosure_envin,** and the **outside_i2csensor_envin. Figure 2.2** delves into the project design one level deeper and shows the entire top level diagram. The three blocks that will be specifically addressed in this paper are the Project Management Block, the Enclosure Block, and the I^2C Sensors Block. All other blocks and interfaces were designed and built by the other team members, Kenny Noble and Dongjun Lee, and will not be further discussed after this section. **Appendix B** shows all the interfaces along with their type and the specifics that each interface needed to accomplish. The interfaces that will be delved into in this paper are the I^2C Sensor interfaces (**outside_i2csensor_envin, psupply_i2csensors_dcpwr, i2csensor_processing_data**) and the Enclosure interfaces (**outside_enclosure_envin, outside_enclosure_userin**).

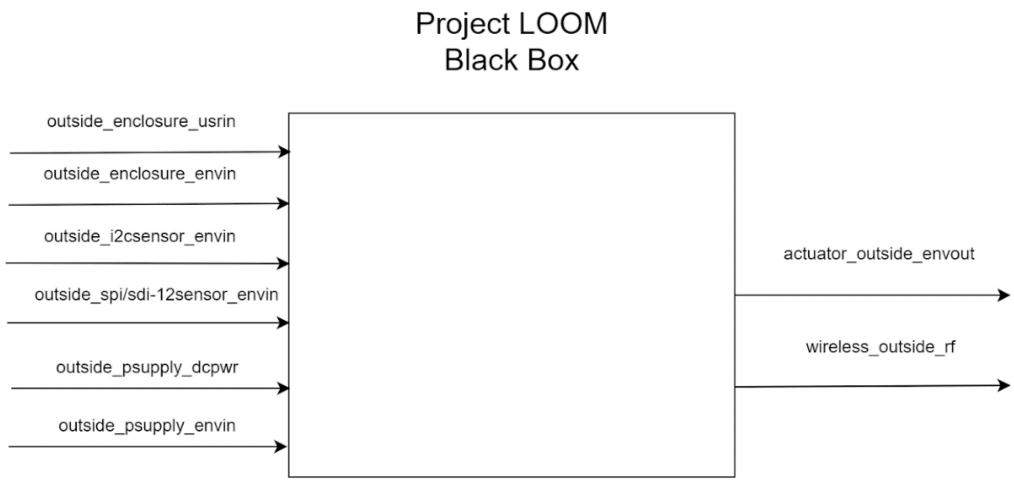


Figure 2.1: Project Loom Black Box Diagram

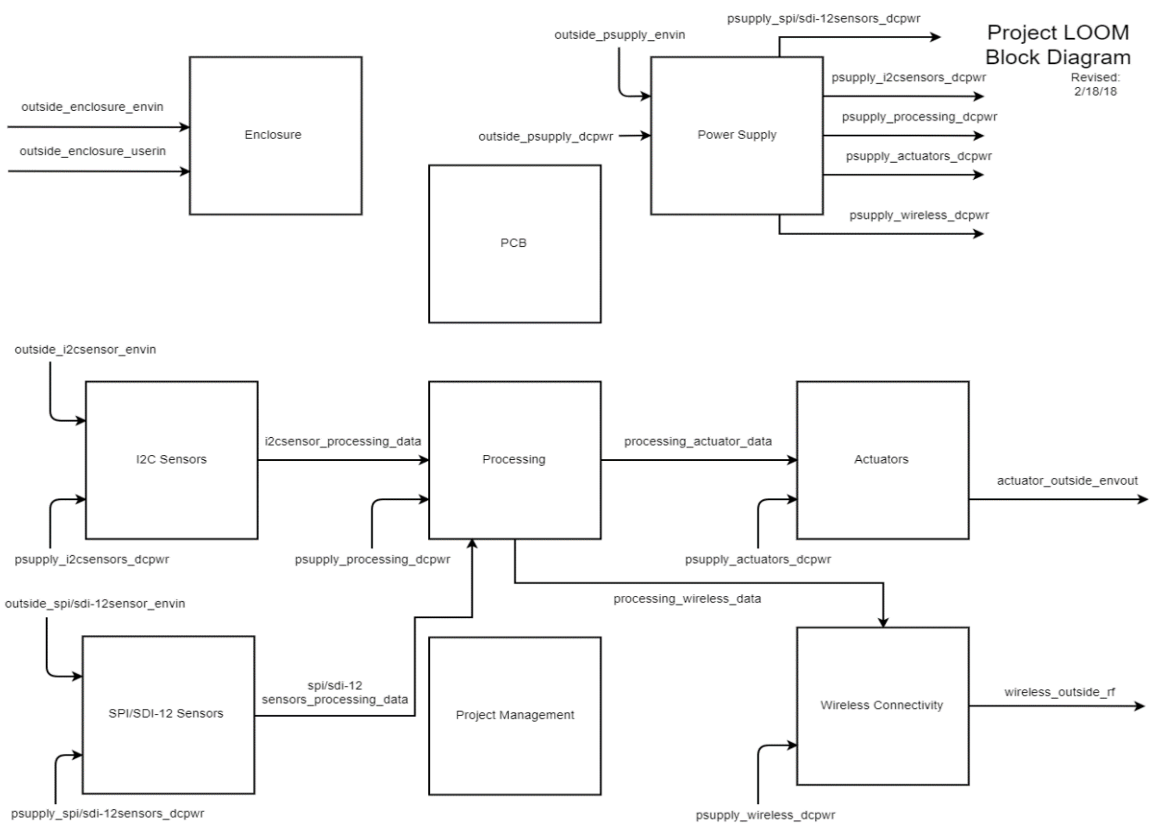


Figure 2.2: Project Loom Top Level Diagram

III. PROJECT MANAGEMENT

A. *Project Management Overview*

In the ECE capstone project, the project manager has a variety of responsibilities that need to be completed in order to keep the project moving forward and help in pushing the team towards the end product. These responsibilities include attending approximately eight project manager meeting throughout the year, keeping the project timeline up to date, setting weekly meetings for the group and taking note of what happened, reflecting on the team and project success as well as the challenges faced, summarizing the project's progress and task success before every Technical Manager meeting, and keeping the team's rules and responsibilities document up to date. All of these responsibilities play a crucial role in keeping the project on track and help to facilitate progress throughout the term. With all of these tasks being completed, it is much easier for the project manager to see the whole picture and to see where potential challenges may crop up before they happen. Throughout the project, a few challenges came up with the three most prominent ones being the multi-disciplinary nature of the project, the large scope of what the project was trying to achieve, and the steep learning curve that was required to be brought up to speed on what had already been achieved in the project. The project required a lot of collaboration between the CS and ECE capstone teams in order to ensure that the all parts of the project would be able to be seamlessly integrated together and the large scope provided by the project ensured that both teams always had an extensive workload. On top of that, due to Project Loom being a project that the stakeholder had already made some progress on and one that had strict deadlines as to when the system needed to be ready, the team was quickly thrown into the middle of the project and worked to make quick progress on being brought up to speed on all that had already been accomplished. That being said, with collaboration between both the ECE and CS capstone team in the form of weekly meetings and discussions on Basecamp as well as using the weekly meetings to ensure that both teams and the stakeholder had the same expectations of what needed to be accomplished, the team was able to work through the aforementioned challenges and gain a few key-takeaways including:

- The importance of collaboration with people of many disciplines
- The importance of making sure that all involved parties have the same expectations for the end product.

B. *Project Narrative*

When the team first began the project, the initial step was to be brought up to speed on the progress that had already been made on the project. Project Loom was started a year back with a grant to explore the rise in the field of the Internet of Things (IoT) so there had already been some progress made on the conceptualization and initial code of the Project Loom system. After researching the code that had been created and learning about the Adafruit feather ecosystem that was being used to complete the project, the next step was the formal documentation of what the system was going to accomplish and what it would consist of. The documentation provided a few challenges as it was quickly realized that the expectations of the stakeholder and the expectations for the capstone class weren't completely aligned. This lead to some confusion when it came to documenting requirements and adjusting the system to ensure that the project would address all areas that the capstone required including a team made PCB and an enclosure. Once the expectations were clarified, it became much simpler to work on documentation as everyone was on the same page. Using Google Drive as a collaborative document repository and Slack for all the team's communication needs, all the essential documents were created and edited throughout the term to meet the needs of the class and the stakeholder. Google Drive became an incredible resource for the team to work together simultaneously and help each other to create a professional document that would be sent out to the stakeholder and posted on the project's website.

The next step after documentation was to begin creating the system itself. With each team member assigned specific blocks, a block for each team member was selected to begin on pertaining to the initial needs of the stakeholder. Since the stakeholder was looking to have a faculty demonstration of the system during Winter term, the team focused on the sensor and actuator integration in order to illustrate to the participants a variety of real world applications that the system could be used for once the system was complete. With the focus on sensors and actuators, each team member was able to help out the others by showing examples of their work and discussing solutions to problems that the team faced. By the end of the first term, a nearly complete project specification document was presented and the first three blocks, I^2C sensors, SPI/SDI-12 sensors, and Actuators, had been completed and presented to the stakeholder and professors.

The second term began with a review of the three blocks that had been completed and then a discussion on what was needed for the remaining blocks. PCB design of the I^2C shield had already begun and a completed schematic was presented during ISO Week 3. Processor integration had been moving along steadily until a problem came up with the Arduino Wire libraries that were being used which led to a task to find out the underlying cause of the problem. Initial research into the Power Supply Block had been conducted and a solar powered shield was ordered to begin implementing a solar powered solution to the power supply. Finally, tasks were assigned to begin looking into enclosure designs as well as ways to integrate LoRa and nRF wireless capabilities into the system.

C. *Timeline*

The timeline was initially created through smartsheet, a free online project management tool, and then later exported to Microsoft Office Project when the free license ran out for the online tool. The first step to making the timeline was utilizing the base requirements and the blocks to section off when each part of the project would be completed. With each block addressing it's specific requirements, the blocks tasks were then assigned to a period of time over which they was expected to be completed. At the beginning of the term, a lot of these task durations were estimated and as the team began to complete tasks and gained a feel of how long specific tasks would take, the timeline was updated to reflect the new dates as well as what the new expectations were for future tasks. On top of that, as new tasks were introduced into the project, they were also added into the timeline for easier tracking. By having task tracking capabilities in Basecamp, keeping the project timeline up to date has been a pretty easy task and teammates can see the real time progress of tasks in Basecamp. The timeline itself has served as a more professional illustration of the tasks that are being completed and how they relate to other tasks in the project. It provides a big picture view of how all the moving pieces will eventually fit together which the task management section of Basecamp doesn't do as effectively. It also allows the stakeholder, technical manager, and professors to see the overall progress of the project and what areas the team is ahead on as well as what areas they're behind on.

Figures 3.1, 3.2, and **3.3** each show the timeline of one term of the project from Fall 2017 to Spring 2018. During the Fall term (**Figure 3.1)**, the OSU Project Loom Team completed all the initial research and software setup, detailed the required engineering requirements for the system, researched the basic design and components that were desired, and begun work on understanding and integrating the WiFi capabilities of the chips. Once the initial research and design had been completed, all of Winter term (**Figure 3.2)** was focused on creating a working prototype that could be demonstrated to a variety of IoT faculty members who were interested in potential applications of the system. Weeks were spent researching and integrating a variety of sensors, integrating LoRa and nRF capabilities, creating an enclosure for the design, and building a power supply that could power the system whether it was deployed out in the field or connected to an outlet. Then, once the main functionality of the system was integrated and working effectively, time was spent working on modularizing the code for ease of use by the user. Once a working prototype was built and shown to faculty, the team than worked on two specific applications requested by IoT faculty members who wanted Decagon versions of the systems deployed out in their labs. The end of Winter Term and beginning of Spring Term (**Figure 3.3)** focused on building the correct configuration of the system for each Professor's use and then deploying the systems out in their labs. Finally, the rest of Spring Term was focused on preparing for the Engineering Expo and leaving resources for future students to pick up the project in later years.

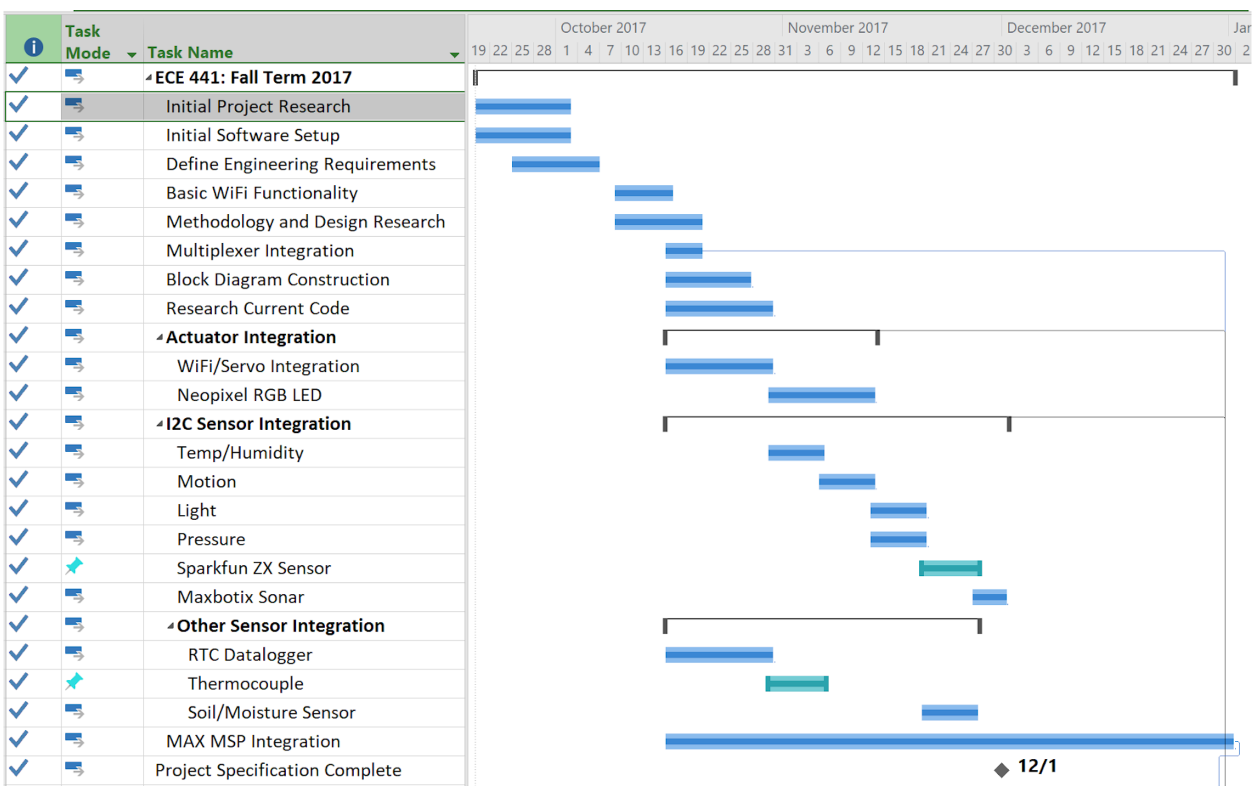


Figure 3.1: Project Timeline ECE 441

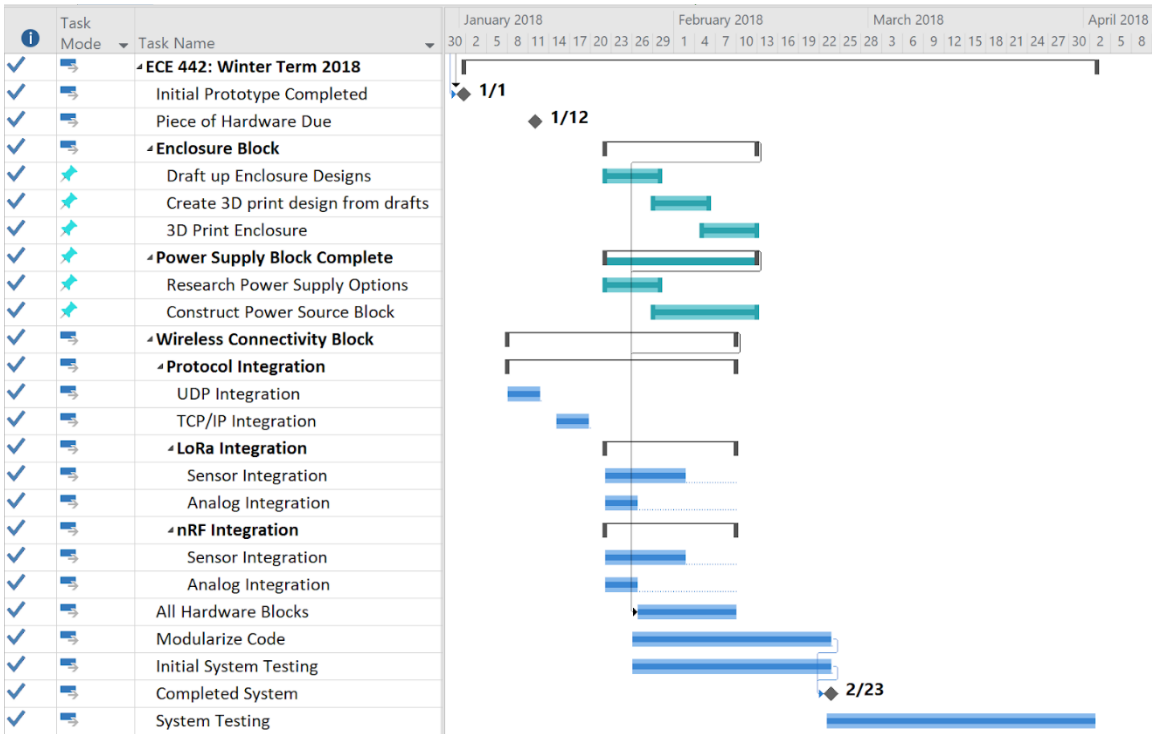


Figure 3.2: Project Timeline ECE 442

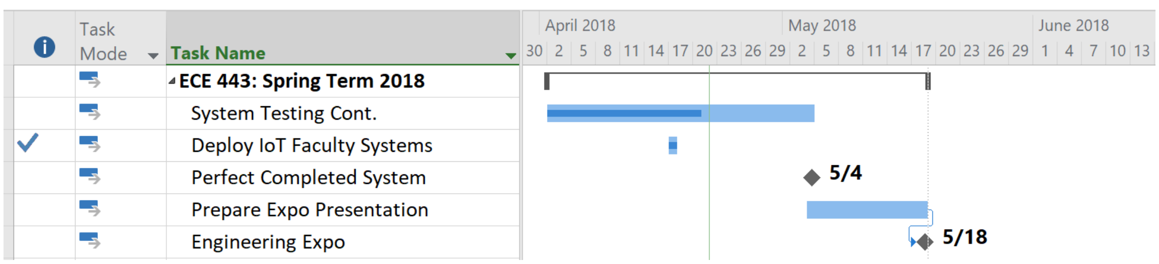


Figure 3.3: Project Timeline ECE 443

D. *Rules and Responsibilities*

Appendix C contains the Project Group/Stakeholder/Instructor Responsibilities document which is a list of responsibilities agreed upon by the group, the stakeholder, the managers, and the instructors. It details the responsibilities that the group had to the stakeholder, the instructor and manager had to the group, and the group had with the rest of the group. A list of general group responsibility questions was answered in order to help layout the expectations that everyone on the team had of one another in order to reduce opportunities for miscommunication and misunderstanding of expectations during the year. All group members agreed to the responsibilities that were laid out and signed to ensure that expectations were uniform across the team.

Our team worked really well to stay on track with all of our tasks and maintained constant communication with each other. I felt as if all my expectations of my team and my teammates were met throughout the term and there was never a time I felt like somebody wasn't doing their fair share or was pushing too much work on others. As far as professionalism in our assignments went, we were able to grow throughout the term by utilizing feedback that we got from other classmates and TA's in order to improve the documentation that we had been working on.

The biggest thing that we could have done as a team was to make sure that our expectations of each other were clearly defined and that we were constantly helping each other to reach everyone's expectations and goals. By keeping in contact and reviewing each other's progress weekly, we could have kept up to date on all the progress that each team member had made and gave feedback to help improve or maintain the level of work that was being created by the team. Finally, by having our stakeholder review our documents more regularly, we could have utilized him to help increase our professionalism in the documents that we were creating. That being said, the progress that we did make and the work that we achieved far surpassed what we initially thought our team would have been capable of at the beginning of the year.

E. Communication Protocols

Communication was a topic that the team had very early on in the process to help keep everyone on the same page as to how information would be spread to all the appropriate parties. Eventually the team settled on five different forms of communication in order to meet all the different needs that were present including instant communication resources, a centralized document repository, real time editing capabilities, professional means to communicate with the stakeholder, version control of code, and task management. **Table 3.1** list out the five platforms that were chosen (Slack, Basecamp, Google Drive, Email, and GitHub) as well as the pros and cons that came with using each one.

Table 3.1: Communication Protocols Pros & Cons

Communication Protocol	**Pros**	**Cons**
Slack	Great for instant communication with teammates Free to use Can easily make more channels if necessary	Not great for document repository needs Doesn't have collaborative document editing Has no task management capabilities
Basecamp	Include document repository, task management, message board, and central communication area to ask questions Links to Google Drive in order to easily access team documents	Only provides one communication channel Doesn't have collaborative document editing
Google Drive	Provides a central space to hold all documents related to the project	Doesn't provide any communication applications

	All documents can be edited collaboratively	other than the document chat feature Doesn’t have any task management capabilities
Email	Great for more professional communication such as updating a stakeholder or providing updates to other involved parties	Longer response time to inquiries Doesn’t have any task management capabilities Not great for document repository needs
GitHub	Provides version control for all the code that was worked on Provides a central space to house all the code related to the project Each person had their own workspace (branch) that they could work on	Doesn’t provide any real time communication capabilities Doesn’t have a task management application Not great for anything other than version control and a central place to hold code

Communication Protocol Reflection

When the project was first started, Slack was the primary form of communication because it was being used by the capstone class and it was really easy to get fast responses from teammates once the Slack app was installed on everyone’s phones. Once the documentation was started however, the team realized that Slack wasn’t going to be the best application to hold the documents in and collaborate on so Google Drive was introduced into the mix in order to provide a document repository where each teammate could edit the document simultaneously and communicate through the in document chat feature that was provided. Once the design and construction of the project was started, Basecamp was introduced in order to provide a central application where the team, the stakeholder, and outside researchers that were involved in the project could all come together and post announcements, track their specific tasks, and see how the rest of the team was progressing. This also allowed the team to easily set up meetings with other members of the larger project and ask questions as needed. Next, email was introduced as a more professional means of communication that allowed for the team to keep the stakeholder updated bi-weekly with all the progress that had been made over the prior two weeks. Finally, the team needed a place to store all the code that was being created as well as the ability to work on code simultaneously. By utilizing GitHub, the team was able to store all the code that was being worked on in one place and provided version control capabilities for all the different files that were being worked on. This allowed the team to work on the code simultaneously and create our own versions that would then come together and be compiled on the main branch when we were done. Being able to work on code simultaneously from home was immensely helpful, especially when it came to working with the CS team because it allowed for both teams to easily access any changes that had been made to the code and start building off the new additions.

All of these platforms were of immense help to the team and allowed for the team to stay on track and keep each other accountable for their specific tasks. While establishing what communication platforms to use took a few weeks due to everyone figuring out what platforms worked best and what applications to use for different tasks, by about halfway through the fall term, the team was fully up to speed on how to use everything and what each platform would be used for. Other than that, there were very few challenges with the communication platforms and the team worked on continuing the effective use of all four of these in order to effectively communicate and keep the project moving forward throughout the year.

F. *Challenges and Significance*

One of the main challenges that the team faced was ensuring that information flowed between the CS and ECE capstone teams involved with Project Loom. With its conception coming from the recently growing Internet of Things field, the project had a huge emphasis and need in both the Computer Science and Electrical Engineering fields which lead to both a CS and ECE capstone team being assigned to the project. Where the challenges came in was with both capstone teams being involved in different classes and having different expectations for this project. The teams had to work together outside of their respective capstone classes in order to ensure that each team was making progress on their respective tasks and that at the end of the year, both team’s efforts would be able to seamlessly come together into the final product. The way that our team worked through this was by having a project meeting with at least one representative from each team as well as the stakeholder and any outside researchers that were available in order to bring all aspects of the project together and give updates on what progress had been made. This gave us the opportunity to see what other areas of the project were being worked on and how our blocks may integrate with their work further down the line. It also allowed us the opportunity to ask questions and get clarifications from other teams and the stakeholder as needed.

Another challenge that the team faced was the fact that the scope of the project had already been created and work had been started on this project before the team had been brought on board. There were a few requirements like the need for the system to be completely built using Arduino code that conflicted with the requirements of the capstone class. This caused a few minor challenges in the beginning when creating documentation because the team had to be brought up to speed on the progress that had already been made, turn that into documentation, all while constructing the system itself because the stakeholder needed the team to make progress on the system for a faculty demonstration early Winter Term. For a few weeks, this meant balancing a heavy workload of documentation and construction for all the team members. With a variety of discussions with the stakeholder and professors, the team was able to work through these challenges and complete all expectations for both the stakeholder and the class. In the discussions, all parties involved discussed what their top priorities were for the project and what expectations were not negotiable. This lead to documentation being created that expressly stated what the system would accomplish and what requirements had to be met in order for all parties to see the project as complete. With a uniform set of expectations that everyone could refer back to, the team was able to focus on the tasks that were listed as a top priority and only move on to the lower priority tasks if time permitted.

Through the challenges that were faced, there were two key-takeaways that had we known at the beginning, may have helped us avoid the challenges that we came across. Knowing what we know now, the team would have strongly stressed the importance of collaboration with people of many disciplines and the importance of making sure that all involved parties have the same expectations for the end product at the beginning of this process. By having a variety of people working in different disciplines, the team was able to see different views for solving the same problems and could ask questions and get help from others who had more experience in their specific field. So often people and teams stick to what they know and what their specialties are but by broadening our horizons and seeing how other teams attack a problem and see a system, we're able to better create a system that meets the needs of every team involved. By making sure that all parties involved have the same expectations through effective communication strategies and weekly debriefs, it helps to solidify what the main goals of the project are and know that different people aren't expecting different things. It also helps to create a more cohesive unit when all the expectations are laid out on the table and everyone agrees on one set of expectations to move forward on. Through the challenges that were faced and the lessons that were learned, our team was able to grow in a variety of areas including the creation of technical documents, different communication strategies, collaboration between disciplines, and setting clear expectations between all those involved. Each challenge required that we develop the aforementioned strategies in order to work past them and often forced us to take a step back and figure out how the challenge had occurred. Through the growth that came from these challenges, each team member will be able to utilize these strategies in nearly all aspects of life from the university setting to the workplace.

IV. PROJECT DEVELOPMENT

A. I^{2}C SENSORS

1) Introduction

The I^{2}C sensor block contains all the I^{2}C sensors that collect data for the Project Loom system and works with the processor block to process the data into a usable format for the user of the system. This section provides an overview of the block's interfaces, the verification and design of the block, and finally a Bill of Materials with all the components that were used within the block. The purpose of the section is to create an understanding for any college student of what the sensor block accomplishes, how it was designed, and the tests it needs to pass in order to be successful.

2) Block Overview

The I^{2}C block combines a variety of sensors including the TSL2591, the FXOS8700, the FXAS21002, the SHT31-D, the Sparkfun ZX sensor, and a force sensitive resistor, and as seen in **Figure 4.1**, the block has two inputs (**outside_i2csensor_envin,**

psupply_i2csensor_dcpwr) and one output (**i2csensor_processing_data)**. Each of the sensors measures a different input from the **outside_i2csensor_envin** input with the TSL2591 measuring light, the FXOS8700 measuring location, the FXAS21002 measuring gyroscopic movement, the SHT31-D measuring temperature and humidity, the Sparkfun ZX sensor measuring distance, and the force sensitive resistor measuring pressure. Once the data is collected, it is sent through the **i2csensor_processing_data** interface where the data is processed and formatted using the I^2C protocol for the user to utilize. The block itself is powered through the **psupply_i2csensors_dcpower** with the voltage ranging from a 2.7 V minimum and a 3.6 V maximum. A TCA9548A multiplexer is used to read from all the sensors simultaneously and it takes a nominal current of .1 uA while in standby and around 6 uA while operating. The peak current that the system can handle is 100mA at which case the multiplexer breaks down. A listing of the interface properties can be seen in **Table 4.1**.

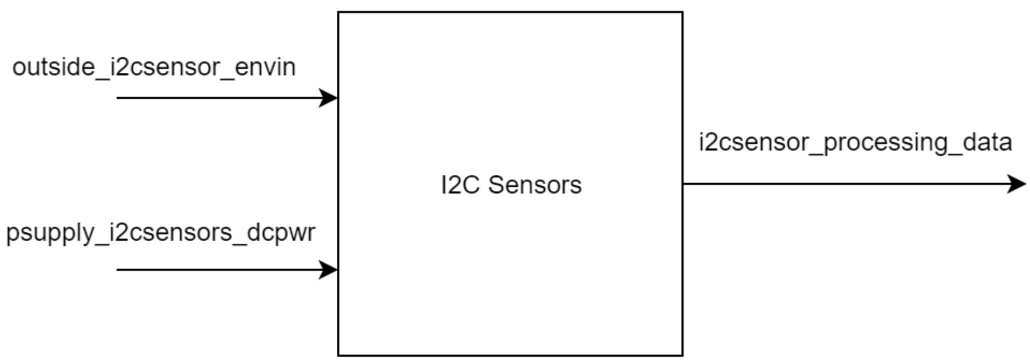


Figure 4.1: Black Box Diagram of I^2C Sensors Block

Table 4.1: I^2C Sensor Interfaces and Properties

Interface	**Properties**
outside_i2csensor_envin	Humidity: 50-100% Light: 10-25,000 Lux Distance: 1-20 cm Pressure: 0-50 N Accelerometer: -10-10 m/s^2 Temperature: 0-30 C
I2csensor_processing_data	Messages: All sensors will display the measurements with numeric values corresponding to the ranges set forth in the envin interface. Protocol: I2C Other: 3.3 V Logic Level
psupply_i2csensors_dcpwr	Inominal: 0.1 uA Standby Mode 6 uA Operating Mode

	Ipeak: 100 mA Vmax: 3.6 V Vmin: 2.7 V

3) Verification

With the interfaces listed above, a verification process is required in order to ensure that the block is working properly and fits the needs of all other blocks that it is connected to. Once these tests have all passed, the block is then ready to be fully integrated into the overall system. All tests can be found in **Appendix D** at the end of this document.

The first test that was conducted was the Sensor Measurement Test which was used to verify all of the **outside_i2csensor_envin** properties. The test consisted of attaching all five of the sensors: the SHT-31D, the TSL2591, the Sparkfun ZX sensor, the Force Sensitive Resistor, and the FXOS/FXAS breakout, and then test that all sensors were hitting the ranges as listed in the properties detailed above. The test passed if all the sensors were able to measure both their respective minimum and maximum values using both the 32u4 and M0 processors. If the test passed, the **outside_i2csensor_envin** interface was considered verified. A video of the testing process can be seen at https://tinyurl.com/i2c-sensor-test.

The second test that was conducted was the Processing Test which was used to verify all of the **i2csensor_processing_data** properties. The test consisted of connecting all the sensors to the system and then measuring the voltage that each sensor output while collecting data. If all the sensors were running on a 3.3 V logic level, outputting no more than 3.3 V, and all sensors were connected to the multiplexer through a SDA and SCL pin, then the test passed and the properties of the **i2csensor_processing_data** were considered verified. A video of the testing process can be seen at https://tinyurl.com/i2c-processing-test.

The final test that was conducted for the I^2C Block was the Power Test which was used to validate the **psupply_i2csensors_dcpwr** interface. The test consisted of powering the system via a power source and measuring the current provided by the multiplexer. In order to pass the test, the system had to be able to run in the case that the minimum and maximum voltages as listed in the interface properties above were supplied. If the system could ran properly during both cases, then the test passed and the properties of the **psupply_i2csensors_dcpwr** interface were considered verified. A video of the testing process can be seen at https://tinyurl.com/i2c-power-test.

4) Design

The schematic as seen in **Figure 4.2** shows how the system is wired together and all the sensors that are involved in the block. The block itself is powered by the microcontroller from the processing block and the data comes in and out through the I^2C protocol utilized by the SDA and SCL pins. The TCA9548A multiplexer acts as the break out for all the sensors with each sensor accessing one of eight ports that the multiplexer is operating. The interfaces are validated even further in **Table 4.2** in order to further clarify why each property was chosen and the impact that that property has. **Figure 4.3** shows a pinout of the TCA9548 which illustrates how the multiplexer connects to the clock (SCL) and data (SDA) pins of each sensor as well as the connections needed to connect to the microcontroller in order to pass along the measurements from the sensors (SCL & SDA) and to power the multiplexer (Vin & GND). The address pins (A0, A1, & A2) are also included in order to change the address of the multiplexer using solder jumpers. This is an added functionality of the multiplexer used to avoid any address conflicts with sensors that would cause the system to not work properly.

i. Wiring Diagram

In the case of the I^{2}C sensor block, the SDA and SCL connections are the **i2csensor_processing_data** interface whereas the microcontroller supplies the power for the **psupply_i2csensors_dcpwr** interface through the 3V and GND pins. The **outside_i2csensor_envin** interface is seen through the forces outside the system acting upon the sensors.

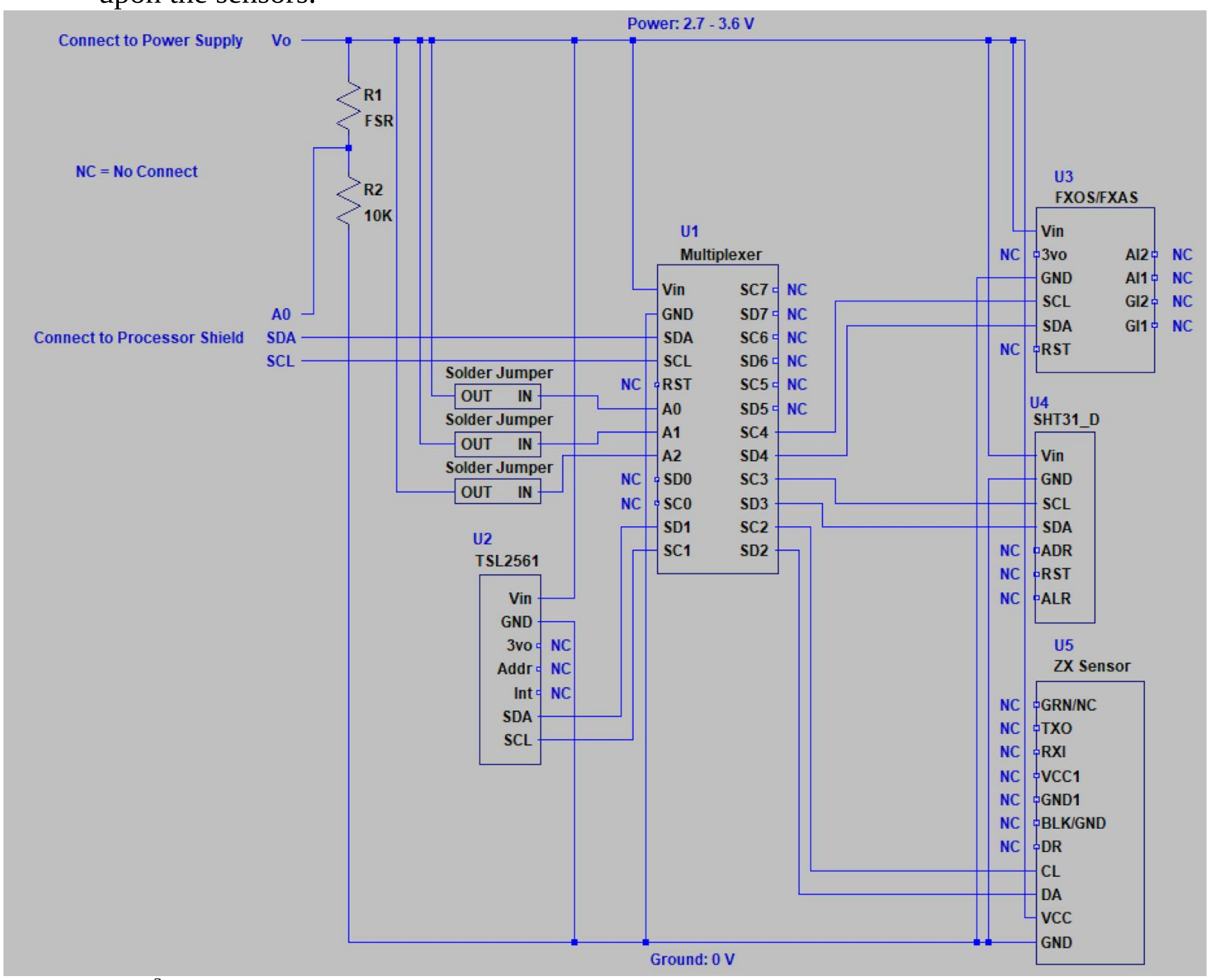


Figure 4.2: I^{2}C Sensor Block Schematic

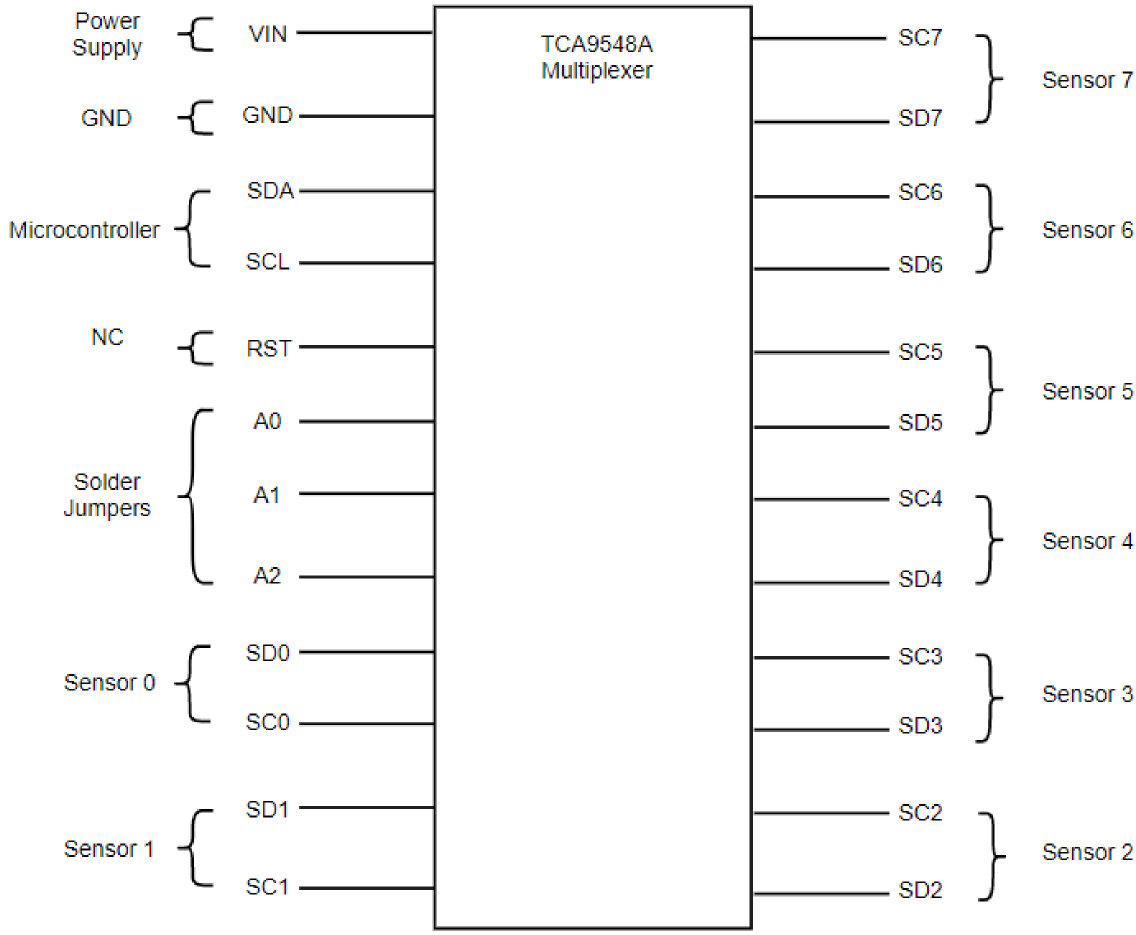


Figure 4.3: TCA9548A Multiplexer Pinout

ii. Design Validation

For this block, a variety of sensors were chosen from Adafruit and Sparkfun which fit the needs of the desired measurements that the system needed to collect. While each sensor could have been custom built, the team decided that the time it would take to design, build, and then implement the sensors would take much more time than the team had for this project so for those reasons the off the shelf solutions were chosen instead.

Table 4.2 lists out all the properties along with the validation for them and every property in this table has passed expectations and is ready to be implemented into the full system.

Table 4.2: I^2C Interface Validation

Property	**Validation**
outside_i2csensor_envin	
Humidity: 50-100%	The SHT31-D sensor is able to measure a range of 0-100% humidity (Page 2, Humidity Sensor Performance [2]); however, the current applications of the system and the difficulty in creating a 0% humid environment, the range has been raised to 50-100% which meets the overall requirements of the system. Any ability to measure values larger than that range will exceed expectations for the system.
Light: 10-25,000 Lux	The TSL2591 sensor is able to measure 188uL to 88,000L (TSL2591 Product Page [3]); however, due to the difficulty in measuring such a high and low light output, the range has been reduced to 10-25,000 Lux which fit the needs of the system. Any ability to measure values larger than that range will exceed expectations for the system.
Distance: 1-20 cm	The ZX sensor is able to measure up to 25 cm (Sparkfun ZX Sensor Documentation [4]) so for the needs of this system, a distance of 1-20 cm will meet the requirement; whereas the full 25 cm will exceed expectations.
Pressure: 0-50 N	The force sensitive resistor can measure a range of 0-100 N (Adafruit FSR Documentation [5]) but the system only requires 0-50 N so that's the range that needs to be met. Any ability to measure values larger than that range will exceed expectations for the system.
Accelerometer: -10-10 m/s^2	The FXOS sensor can measure a range of -4-4 gs (Page 8, Accelerometer mechanical characteristics [6]) which converts to about -40-40 m/s^2 but the range has been reduced to -10-10 for the needs of the system and ease of testability. Any ability to measure values larger than that range will exceed expectations for the system.

Temperature: 0-30 C	The SHT31-D sensor is able to measure a range of -40-125 C (Page 4, Temperature Sensor Performance [2]); however, the current applications of the system can work with a much smaller range of 0-30 degrees Celsius. Any ability to measure values outside that range will exceed expectations for the system.
I2csensor_processing_data	
Messages: All sensors will display the measurements with numeric values corresponding to the ranges set forth in the envin interface.	With the validation shown in the above properties, the purpose of this property is to ensure that the data that is coming from the sensors is in a viewable format for the user to see. The values displayed must meet at minimum the ranges set forth in the above properties.
Protocol: I2C	All of the sensors other than the force sensitive resistor require I2C protocol to transmit data and are built around the integral utilization of the SDA and SCL pins for data transfer and clock functionality.
Other: 3.3 V Logic Level	All of the sensors utilize 3.3 V logic levels opposed to 5 V logic due to the low power needs and applications; therefor, 3.3 V logic is what will be utilized when sending and receiving data from sensors.
psupply_i2csensors_dcpwr	
Inominal: 0.1 uA Standby Mode 6 uA Operating Mode	The TCA9548A multiplexer draws .1 uA while in standby and 6 uA while in operating mode (Page 5, Electrical Characteristics [7]). The voltage supplied to the block can't restrict this current otherwise the block will fail.
Ipeak: 100 mA	The TCA9548A multiplexer can have a peak supply current of 100 mA (Page 4, Absolute Maximum Ratings [7]) otherwise the multiplexer will fail causing the block to fail.
Vmax: 3.6 V	According to the TSL2591 datasheet, the TSL2591 has the lowest max voltage of the sensors with a value of 3.6 V max (Page 5, Recommended Operating Conditions [3]) that can be supplied thus

	the max voltage for the system was chosen to be 3.6 V since all sensors can operate at that max.
Vmin: 2.7 V	According to the TSL2591 datasheet, the TSL2591 has the highest min voltage of the sensors with a value of 2.7 V min (Page 5, Recommended Operating Conditions [3]) that can be supplied thus the min voltage for the system was chosen to be 2.7 V since all sensors can operate at that value.

5) Bill of Materials

The Bill of Materials provided in **Appendix E** lists all the parts that were used in order to implement the functionality of the block as described in the sections above. In the Bill of Materials, the part name, reference designator, value (resistor only), manufacturer, part number, distributor, price, and additional documentation are all provided in order to easily get ahold of all the parts that are required to create the I^2C block. The overall cost of the block came out to be $78.85 which included all the sensors, the multiplexer, and the wires needed to connect the parts together. What is not provided in this Bill of Materials is the processors or power supply needed to run the code and power the block. Details on those blocks and their corresponding Bill of Materials can be found at https://tinyurl.com/ECELoomProjectSpec.

B. ENCLOSURE

1) Introduction

The purpose of the enclosure block is to keep the system safe from harmful environmental elements as well as to contain all the systems pieces in a singular unit. This section provides an overview of the environmental inputs the enclosure needs to withstand, as well as the verification, design, and Bill of Materials that go along with the block. The purpose of the section is to create an understanding for college student of what the enclosure block accomplishes, how it was designed, and the tests it needs to pass in order to be successful.

2) Block Overview

The enclosure was built using 3D modeling software to model the design and then 3D printing to print the enclosure itself. This block ensures that the sensor blocks, processing block, and actuator block are all protected in a way that they can still function while protecting them from harmful environmental impacts that are coming in through the **outside_enclosure_envin** interface**.** The **outside_enclosure_userin** was added to illustrate the fact that the system was configurable and allow for the user to swap out parts as desired. **Figure 4.4** shows the black box diagram for this block and further down **Table 4.3** shows all the properties associated with the enclosure.

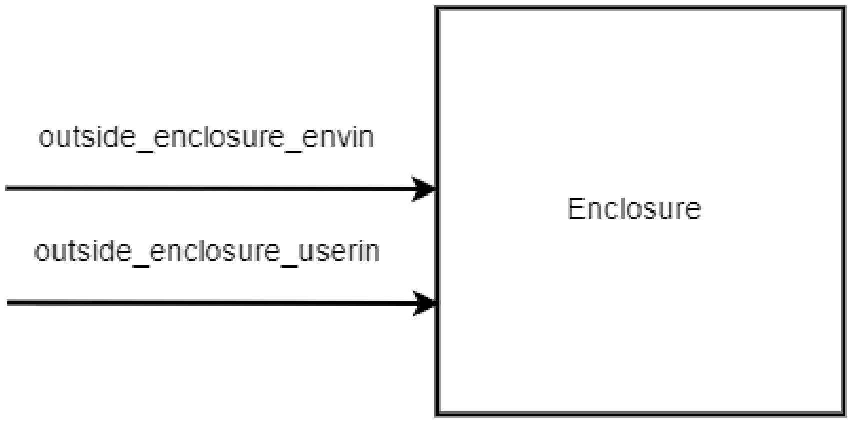


Figure 4.4: Block Box Diagram of Enclosure Block

Table 4.3: Enclosure Interfaces and Properties

Interface	**Properties**
outside_enclosure_userin	Other: Allows for the user to configure the system during use due to the modularity of the system Type: User can load code onto the processor, switch out processors, sensors, actuators, or other shields, and can open and close the enclosure to access all parts that are being swapped out.
outside_enclosure_envin	Temperature: 100 degrees F Light: 10-25,000 Lux Water: 1 inch of water from the base

3) Verification

With the interface listed above, a verification process is required in order to ensure that the block is working properly and fits the needs of all other blocks that it is protecting. Once the test has passed, the block is then ready to be fully integrated into the overall system.

The one test conducted for the Enclosure Block was the Outside Environment Test which was used to verify all of the **outside_enclosure_envin** properties. The test consisted of submitting the Enclosure to environmental factors in four special cases. The Enclosure was heated up to 100 degrees F, placed in a dark environment, placed in a bright environment, and set in an inch of water in order to test the effects of temperature, light, and water on the Enclosure. The test passed if the output of the system other than the measurements from the respective sensors was not affected by the environmental factors. A video of the testing process can be found at https://tinyurl.com/enclosure-environment-test and the detailed testing process can be found in **Appendix F**.

4) *Design*

i. Mechanical Drawings

All mechanical drawings for the enclosure were created in Autodesk Fusion 360. **Figure 4.5** shows the 3D view of the enclosure with both the lid and the main body. The rest of the measurements and data sheets are listed in **Appendices G-L** for reference **Appendix G** shows a variety of views of the main body of the enclosure. These show the outer dimensions as well as the dimensions of the connector holes and the battery holder. **Appendix H** shows the lid of the enclosure as well as where the solar panel will rest and it's screws reside. **Appendix I** contains the top down dimensions of the processors that will be used with the system and they will reside next to the battery compartment. **Appendix J** shows the top down view of the battery charging circuit which will reside above the battery compartment as well as the solar panel which is screwed down to the top of the lid. **Appendix K** shows the dimensions of the battery which lays in the battery holder. **Appendix L** contains the dimensions of the I^2C printed circuit board (PCB) which will connect to the processor via a set of headers. **Figure 4.6** and **Figure 4.7** show the finalized printed enclosure that was produced for the Project Loom system. In **Figure 4.7,** the solar charging circuit and the processor can be seen in their respective positions along with the battery which is connected to the solar charging circuit. **Figure 4.8** shows the completed and waterproofed enclosure with six I^2C sensors connected to the system through the waterproof connectors.

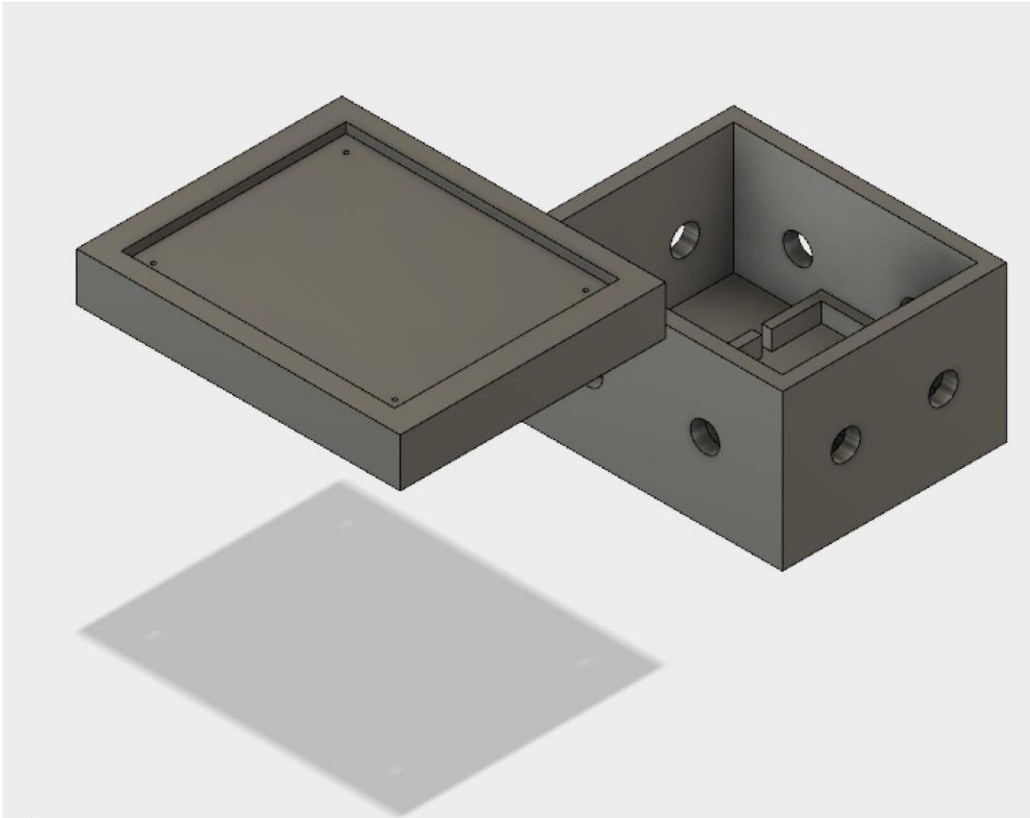


Figure 4.5: 3D View of Enclosure

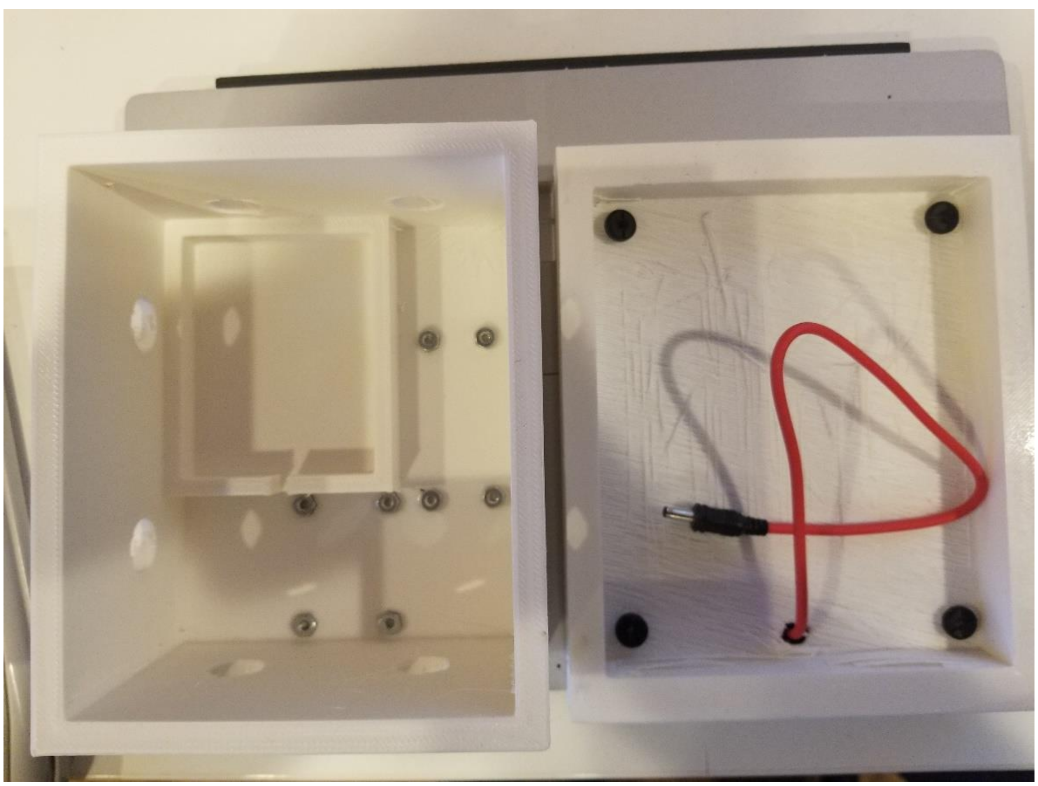


Figure 4.6: 3D Printed Enclosure

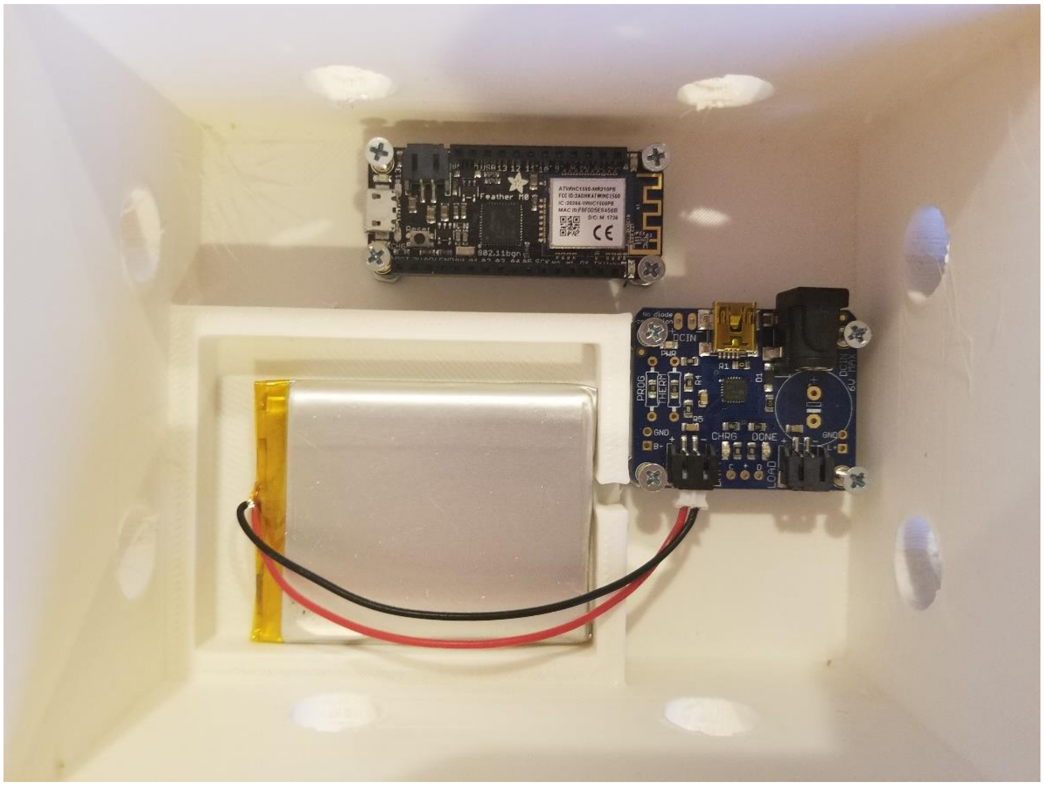


Figure 4.7: 3D Printed Enclosure with PCB

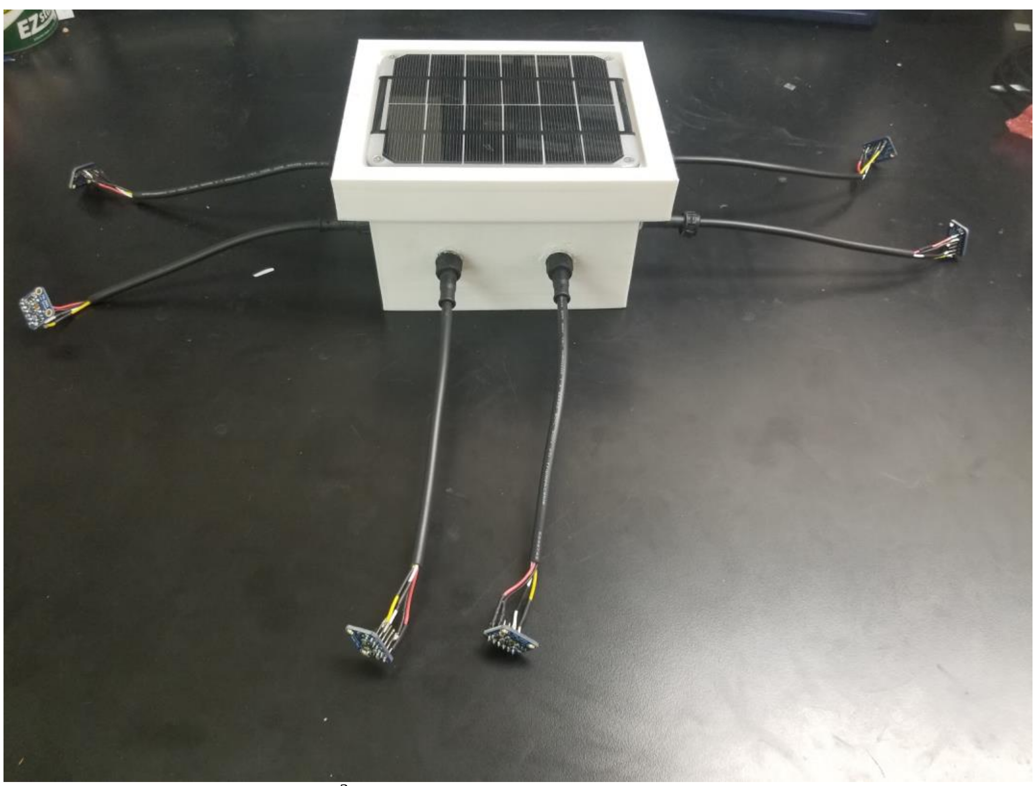


Figure 4.8: Enclosure with I^2C Sensors Attached

ii. Design Validation

The main requirement for the Enclosure Block is that it is able to contain all blocks that are connected to the system. The enclosure will need to contain the processor from the processing block, the I^2C printed circuit board, the power supply, and connectors to any external sensors and actuators. In the enclosure, the I^2C printed circuit board will be stacked on top of the processor via their headers and attached to the enclosure via a #3 screw and nut combo connected to the bottom of the enclosure. The battery will lay in the battery compartment on the enclosure shown in **Appendix G** and the battery charging board will be connected to the enclosure via #3 screw and nut combo**.**

Another requirement is that the enclosure is able to be opened and closed in order to connect different processors and shields that are integrated into the Loom system. For example, the user may want to swap out an I^2C PCB for a servo shield or a 32u4 processor for an M0 processor and needs to be able to do that by opening the enclosure. As seen by **Figure 4.5**, the enclosure will come in two pieces, a lid and a body, so that the lid can be taken off and the system can be accessed. **Appendix H** shows that the lids is larger than the enclosure body in order to prevent rain from entering from the top of the enclosure.

Finally, the enclosure needs to survive 100 F degrees temperature, 25,000 lux, and being immersed in at least 1 inch of water from the base of the enclosure. Plastic material was used on the enclosure for its resistance to water, ability to not alter under bright light conditions, and its melting point which is upwards of 200 C degrees depending upon the plastic used [14]. The connectors being used to connect sensors from the outside of the enclosure to the inside are Adafruit's waterproof polarized 4-wire cables [15] which will keep the connections between the system and the sensors safe from water. A silicone conformal coating [16] was also used on the sensors to ensure that they can survive water damage outside of the enclosure. For additional security, a few layers of FlexSeal [17] were sprayed on the enclosure which filled any pores in the plastic with a liquid rubber seal. Finally, #3 nuts were glued to the bottom of the enclosure and used with #3 screws to secure the printed circuit boards to the enclosure.

5) *Bill of Materials*

The Bill of Materials provided in **Appendix M** lists all the parts that were used in order to implement the functionality of the block as described in the sections above. In the Bill of Materials, the part name, manufacturer, part number, distributor, price, and additional documentation are all provided in order to easily get ahold of all the parts that are required to create the Enclosure block. The overall cost of the block came out to be $76.52 which included the plastic and printing of the enclosure, the silicone conformal coating, the #3 screw and nut combo that connects the printed circuit boards to the enclosure, the screws for the solar panel to connect to the enclosure, and the waterproof connectors. What is not provided in this Bill of Materials is the power supply needed to power the system or the solar panel to charge the battery. Details on those blocks and their corresponding Bill of Materials can be found at https://tinyurl.com/ECELoomProjectSpec.

V. CASE STUDIES & FUTURE APPLICATIONS

A. HC 407 – Data Driven Enchanted Objects

The first application of Project Loom was introduced to a group of Honors College students in a class lead by Professor Udell, titled "Data Driven Enchanted Objects". In the class, students had the opportunity to utilize the initial variation of the Project Loom system that consisted of the M0 processor, an MPU6050 gyroscope and accelerometer, a neopixel, a temperature sensor, and a pressure sensor, in order to "enchant" some sort of everyday object. **Figure 5.1** shows the course description and details how the project focuses on hands-on projects to explore the connection between magical objects we've heard of in stories and the real world applications that are now being created that mimic them.

HC 407 **Data Driven Enchanted Objects**

CRN: 38919 Section 010 SEM T 1600 - 1750 2 HC Credit(s)

Instructor(s): Chet Udell

Arthur C Clarke famously wrote, "Any sufficiently advanced technology is indistinguishable from magic." How have our ideas of enchanted objects inspired new technology over time? How has advancing technology transformed our notions of magic? You will explore these ideas through experiential hands-on projects using plug and play wireless sensors to build your very own enchanted objects that interact with the seemingly magical digital world around us.
From Harry Potter to Hunger Games, run, watches detect when their bearer has heart trouble, and you can click your heels three times (to send an emergency call to your phone) to get out of a meeting or bad date. While technologies and the words we use to describe them may evolve, our desire to acquire objects that augment our capacities to gain knowledge, communicate, protect, and create have remained largely consistent throughout recorded history and across cultural barriers. Enchanted objects that facilitate these wishes are extant in our folklore, mythologies, epic poems, religious texts and can be found in much of our earliest recorded literature. We'll supplement and inform our project experiences through reading and video excerpts you select to investigate a variety of magical objects and their real-world counterparts throughout history. **Graded: P/N. Satisfies: HC Colloquia**

Figure 5.1: Data Driven Enchanted Objects Course Info

One of the projects that was created was a smart pen (**Figure 5.2**) that could be used to draw or write on a computer. By applying pressure to the pressure sensor the pen could be used to erase lines. A button on the system allowed for another way to erase by erasing the entire drawing when pressed. By applying less pressure to the pressure sensor, the pen could be used to shade by drawing in lighter versions of gray depending upon the amount of pressure applied. The MPU6050 accelerometer/gyroscope was then used to plot where the pen was writing. By detecting the movement of the pen, the MPU6050 would use that information to know where to draw on the computer. Finally, a distance sensor was and by moving it towards or away from the screen, the pen could be used to change the size of the pen strokes being used on the computer.

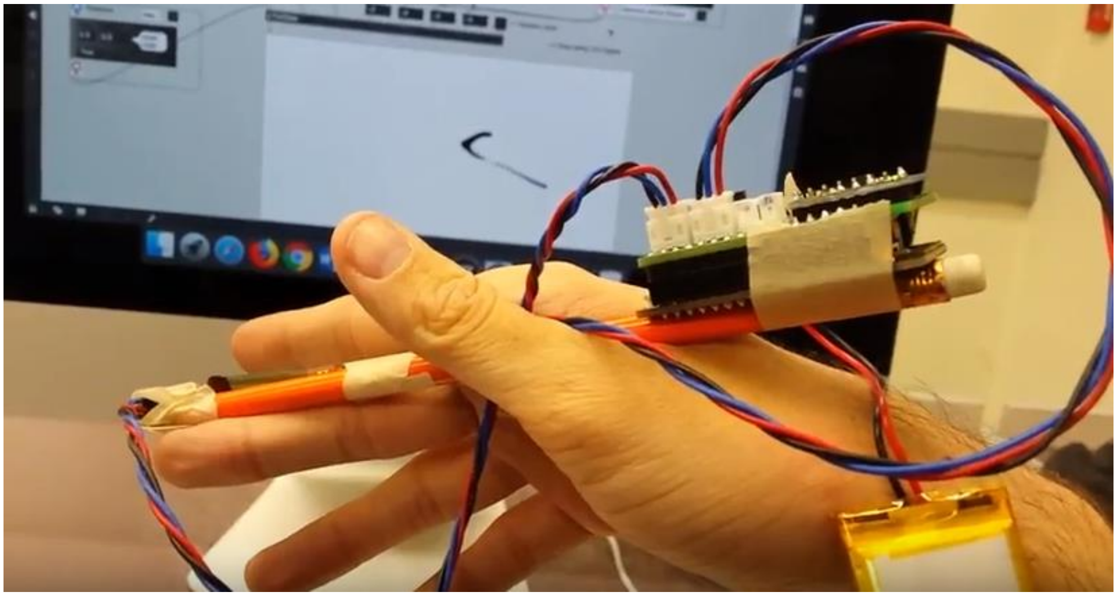


Figure 5.2: Smart Pen Utilizing Project Loom

Another project created by a student utilized a Benny the Beaver (**Figure 5.3**) foam figure and a variation of the Project Loom system to play music and control videos via a variety of sensors. A photoresistor was used to control the volume of the music being played. By shining light on the sensor, music would be played by Benny and fade out as the light was moved away. A pressure sensor was utilized to have Benny play a clapping sound anytime the figure was squeezed. Finally, a MPU6050 was used to control a video. By tipping Benny down, the video starts, by rotating him approximately 45 degrees, the video played at half speed, and then by rotating him 90 degrees, the video played at double speed. Finally, by pointing Benny straight up, the video would stop.

Figure 5.3 Benny the Beaver Figure Utilizing Project Loom

Other projects for the class included a LED glove that turned on a neopixel in the glove when pressure was applied to a pressure sensor, a Walkman that could be controlled via a MPU6050 accelerometer/gyroscope, and a box that emitted a variety of sounds depending upon which sensor was activated (i.e. pressure sensor for purr, push button for roar, distance sensor for growl). All of these projects were created by students who had no prior experience with the Project Loom system and many students had very little to no programming or soldering experience prior to the class. As an initial usability test of the system, Data Driven Enchanted Objects illustrated that Project Loom's goal of creating a system of sensors and actuators that required very little experience to create IoT applications was viable and helped solidify the foundation that would later be built upon to create an even more robust system with much wider variety of applications.

B. Spring Term 2018 – LB Farm Rollout

During Winter Term 2018, the team had two professors who were interested in the system that was being designed and wanted to test pilot some of the systems for their Spring Term 2018 classes. Both applications consisted of using the systems to test out soil moisture and electrical conductivity and required that the data be sent from the field systems to a "hub" that could collect all the data and send the data to a Google Spreadsheet that could be accessed from anywhere that was connected to the internet.

The first application was looking specifically at the soil moisture levels of putting greens vs regular grass soil and had two groups of students who would be collecting soil moisture levels using different techniques. The first group would be travelling to the field multiple times throughout the term to measure the moisture levels by hand whereas the second group would be collecting the data on the online spreadsheet using the Project Loom system. The second application was looking more broadly at collecting a variety of soil data including soil moisture levels, temperature, and electric conductivity. These systems wouldn't be placed on the putting greens but rather in a soil bed with one sensor being six inches deep and the other sensor being twelve inches deep. This allowed for students to analyze the data that was coming in over time and see how the varying depths differed. After discussing the needs with the professors, it was decided that four field systems (2 per professor) would be constructed with two hubs collecting data (1 per professor) and a Bill of Materials (**Figure 5.4**) was put together with all the materials that were needed to construct all the systems.

4 Field Systems & 2 Hubs						
Decagon Sensors		Currently Have	4	-	Sensors being used to collect the soil moisture data in the field systems	N/A
Decagon Shields		Currently Have	4	-	Shields that will be used to connet the sensors to the processor and RTC	N/A
RTC Shields	ACJU	$13.95	5	$69.75	Used for timing transmisions and other time related tasks on the system	Adafruit
LoRa/M0 Shields	ACJU	$34.95	7	$244.65	Used to process the sensor data and transmit/receive the data being sent	Adafruit
2000 mAh Lithium Battery	ACJU	$12.50	5	$62.50	Used to power the field systems that are deployed	Adafruit
3.5mm OD/1.1mm ID DC jack panel mount	ACJU	$1.49	5	$7.45	Used to connect the solar panel to the enclosure of the field systems	Tinkersphere
Long Range Antennas	ACJU	$18.22	3	$54.66	Used to receive data from the field systems	Amazon
Ethernet Breakout Board	ACJU	$19.95	3	$59.85	Used to connect the LoRa Hub to the internet	Adafruit
900MHz Antenna kit (Small for Transmitters)	ACJU	$12.75	7	$89.25	LOOM Eval components	Adafruit
uFL SMT connectors	ACJU	$0.75	7	$5.25	LOOM Eval components	Adafruit
Enclosure		Made in Lab	7	-	Used to encase the field systems and the hub in order to protect the system	Made in Lab
4 Solar Panel Additions (Optional)						
Solar Panel		$29	5	$145	Solar panel used to chage the battery	Adafruit
Solar Lithium Ion Polymer Charger		$17.50	5	$87.50	Board used to charge the battery using the solar panel	Adafruit
Male DC Power Adapter		$2	5	$10	Adapter used to connect the solar panel to the charger board	Adafruit

Figure 5.4 Spring Term Rollout Bill of Materials

After further discussions with the professors, it was decided that both applications could be conducted at Lewis-Brown Horticulture Farm and that multiple sensors could be utilized with one field system. This led to the team deciding on one hub that could collect data for both applications and a single field system for the application that was testing varying depths. This narrowed the construction down to three field systems and one hub to collect data. All the systems were deployed at Lewis-Brown Horticulture Farm on April 17th, 2018 and the current measurements being recorded by the system can be seen at https://tinyurl.com/lbfarm-results. **Figures 5.5**, **5.6**, and **5.7** show the installation of the sensors out at LB Farm.

Figure 5.5: Hub Antenna

Figure 5.6 Line Running From System to Sensors

Figure 5.7: Line Running Out to Putting Green

C. *Potential Future Applications*

With the two applications shown above, it's clear that Project Loom has a lot of potential when it comes to redefining the education sector. According to a Business Insider article titled "How IoT in education is changing the way we learn", "73% of all U.S. teenagers had access to a smartphone…nearly 100% of U.S. public schools have internet access" [18] meaning that schools are primed to begin implementing internet of things applications if they're providing a benefit and are easy to use. One of the biggest barriers of entry for the IoT industry used to be the cost of technology and the experience needed to build and program the technology; however, with systems like Project Loom, those barriers to entry are decreased by an easy to use set of components that don't require that knowledge or large capital. This opens up applications such as having K-12 students work on hands-on STEM projects that can be used in a variety of settings. They can build a suite of sensors for the school that provide a low cost security system, or implement sensors and actuators on their pens to create a more efficient note taking system, or enchant a diorama that brings a history scene to life. The possibilities are endless and the opportunities to peak students interest in the STEM fields with low cost, easy to learn systems in an educational environment would likely give rise to a new generation of innovators and entrepreneurs. Producing applications in the field of technology and the Internet of Things has often been limited to those who have spent years learning how to build and program hardware but with a system like Project Loom, these fields are opened to a whole new audience of people ranging from the smartest minds in the world to the young children currently in our education system.

While Project Loom provides a lot of opportunities for K-12 and higher education students, the ease of use also opens up the possibility of integrating technology into a wide range of fields such as music and the arts that don't often spend a lot of time focusing on technology applications. Imagine a painting that has a variety of Sparkfun ZX distance sensors embedded in them that alter the painting as people pass in front of the sensors. Or think of a sculpture that embeds SHT31-D temperature/humidity sensors as well as TSL2561 sensors and adjusts its color depending upon the lighting and temperature of the environment around it. Or imagine a musical instrument that has a FXOS & FXAS sensor embedded so that it can map out the movement of the instrument and control a variety of lighting effects depending upon that movement. Those are just a few examples of seemingly endless possibilities that could be produced with the applications presented by Project Loom. The system wasn't made to produce one specific application but rather to provide as much customization and potential applications while still keeping the use of the system simple enough for somebody who had little to no experience with IoT or hardware and software skills. While the applications listed in this section cover a range of potential uses, the future applications of Project Loom are in the hands of whoever decides to use it.

VI. CONCLUSION

With the field of IoT growing rapidly in the next 5-10 years, the opportunities to start applying technology that takes advantage of this industry are incredibly promising. Project Loom started as a mere concept that aimed at addressing the main barriers of entry that held people back from creating potential IoT applications and has ended with a viable foundation to help people start creating their own applications. Over the process of a year, eight sensors (temperature/humidity, light, soil moisture, water temperature, accelerometer/magnetometer, gyroscope, distance, and pressure), three actuators (relay, servo, motor, and neopixel), three wireless protocols (Wi-Fi, nRF, LoRa), and two processors (32u4 AVR, M0 SAMD) have all been integrated into the Project Loom framework and are now supported with a variety of potential applications. The conclusion of this thesis resulted in six of those sensors being integrated, a waterproof enclosure to house the system out in the field, and the project management that kept the team on track and facilitated the tasks that were accomplished throughout the year. With the foundation that has been created, Project Loom has been left in a state where new sensors and actuators can be integrated for use and minimal technical experience is necessary to use the applications that are currently being supported. Recommended next steps for Project Loom include the following:

- Gather information on the ease of use of the system by working with a variety of groups (high school students, artists, researchers, etc) to create applications using the current system
- Create a How To Guide with details on how to create recommended applications as well as suggested future applications that could be created
- Continue to integrate new sensors and actuators into the system for use

In terms of Project Management, a lot was learned about what worked well and what didn't work well for the team that was being led. Both the ECE and CS team were incredibly self-lead and the ECE team did not require a lot of oversight when it came to keeping the team on track. Each person completed their tasks on time and to the degree of quality that was expected of them so the project management of the team mostly focused on ensuring that everyone knew what the expectations were and when the deadlines for assignments and tasks were. While many leaders believe that a set of traits or behaviors is what is needed to best lead every group, the team's experience during the capstone project over the last year shows that what's really important is situational leadership also known as the ability to adapt one's leadership style to the situation at hand. While some teams needed a very hands on approach, the ECE Project Loom team didn't and thus the project management style of the author was adapted to fit the needs of the team.

In terms of I^2C sensor integration and Enclosure design, very little knowledge was had prior to the start of the project on I^2C sensors or on 3D modeling software; however, throughout the year, time was put into developing knowledge and experience in these areas. For the I^2C sensors, time was spent at the beginning of the year looking over the datasheets for the I^2C sensors and looking through provided example code to see what types of functions were supported by each sensor. Then research was done into how the sensors interacted with the processor (through the SDA and SCL pins) as well as how multiple sensors of the same address could be utilized simultaneously (through a multiplexer). Finally, circuits were designed first on a breadboard and then after testing and validation of the sensors had been completed the design was integrated into a PCB. For the Enclosure, time was spent learning how to use Fusion 360, an online 3D modeling tool provided by Autodesk, in order to gain the skills necessary to create a 3D printed enclosure. Once the design was created, time was spent learning how the 3D printer worked and printing the enclosure from the Fusion 360 design file. The last step involved looking into waterproofing techniques and implementing the techniques to ensure that water damage would not occur while the system was out in the field. All of these tasks were necessary to learn in order to complete the requirements set forth by the project and the conclusion of the project resulted in a better understanding of how these processes work as well as the time needed to research a new sensor or learn a new process from scratch.

With the conclusion of the initial Project Loom system, the groundwork has been lain for future teams and researchers to carry the progress forward and create an even more robust and inspired modularized Internet of Things prototyping system. On top of that, with the skills learned in project management, I^2C integration, and enclosure design, the application of these skills will without a doubt be utilized in future electrical engineering projects and the foundation created during this year will help open new opportunities that require these skills in order to be an effective engineer and project manager.

VII. REFERENCES

[1] Forbes.com. (2018). 2017 Roundup Of Internet Of Things Forecasts. [online] Available at: https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/ [Accessed 23 Apr. 2018].

[2] "Datasheet SHT3x-DIS", *Cdn-shop.adafruit.com*, 2017. [Online]. Available: https://cdn-shop.adafruit.com/product-files/2857/Sensirion_Humidity_SHT3x_Datasheet_digital-767294.pdf. [Accessed: 03- Dec- 2017].

[3] "TSL2591 Datasheet - Apr. 2013 - ams163.5", *Cdn-shop.adafruit.com*, 2017. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/TSL25911_Datasheet_EN_v1.pdf. [Accessed: 03- Dec- 2017].

[4] S. 5V, "ZX Distance and Gesture Sensor Hookup Guide - learn.sparkfun.com", *Learn.sparkfun.com*, 2017. [Online]. Available: https://learn.sparkfun.com/tutorials/zx-distance-and-gesture-sensor-hookup-guide. [Accessed: 03- Dec- 2017].

[5] "Overview | Force Sensitive Resistor (FSR) | Adafruit Learning System", *Learn.adafruit.com*, 2017. [Online]. Available: https://learn.adafruit.com/force-sensitive-resistor-fsr/overview. [Accessed: 03- Dec- 2017].

[6] "FXOS8700CQ", *Cdn-learn.adafruit.com*, 2017. [Online]. Available: https://cdn-learn.adafruit.com/assets/assets/000/043/458/original/FXOS8700CQ.pdf?1499125614. [Accessed: 03- Dec- 2017].

[7] "TCA9548A Low-Voltage 8-Channel I 2C Switch With Reset", *Cdn-shop.adafruit.com*, 2017. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/tca9548a.pdf. [Accessed: 03- Dec- 2017].

[8] Learn.adafruit.com. (2018). *Downloads | Adafruit Feather 32u4 Basic Proto | Adafruit Learning System*. [online] Available at: https://learn.adafruit.com/adafruit-feather-32u4-basic-proto/downloads [Accessed 8 Feb. 2018].

[9] Learn.adafruit.com. (2018). *Downloads | Adafruit Feather M0 Basic Proto | Adafruit Learning System*. [online] Available at: https://learn.adafruit.com/adafruit-feather-m0-basic-proto/downloads [Accessed 8 Feb. 2018].

[10] Learn.adafruit.com. (2018). *Using the Charger | USB, DC & Solar Lipoly Charger | Adafruit Learning System*. [online] Available at: https://learn.adafruit.com/usb-dc-and-solar-lipoly-charger/using-the-charger?view=all#downloads [Accessed 8 Feb. 2018].

[11] Cdn-shop.adafruit.com. (2018). *LiPo Battery Datasheet*. [online] Available at: https://cdn-shop.adafruit.com/datasheets/785060-2500mAh_specification_sheet.pdf [Accessed 8 Feb. 2018].

[12] GitHub. (2018). *VoltaicEngineering/Solar-Panel-Drawings*. [online] Available at: https://github.com/VoltaicEngineering/Solar-Panel-Drawings/blob/master/Voltaic%20Systems%202W%206V%20112x136mm%20DRAWING%20CURRENT%202017%207%2020.pdf [Accessed 8 Feb. 2018].

[13] Cdn-shop.adafruit.com. (2018). *Headers Datasheet*. [online] Available at: https://cdn-shop.adafruit.com/datasheets/00548.pdf [Accessed 8 Feb. 2018].

[14] Plastikcity.co.uk. (2018). *Material Melt and Mould Temperature Chart | PlastikCity*. [online] Available at: https://www.plastikcity.co.uk/useful-stuff/material-melt-mould-temperatures [Accessed 8 Feb. 2018].

[15] Industries, A. (2018). *Waterproof Polarized 4-Wire Cable Set*. [online] Adafruit.com. Available at: https://www.adafruit.com/product/744 [Accessed 8 Feb. 2018].

[16] Circuitspecialists.com. (2018). *Silicone Conformal Coating*. [online] Available at: https://www.circuitspecialists.com/422b-340b.html [Accessed 8 Feb. 2018].

[17] SEAL®, F. (2018). *Flex Seal®, in Original, Brite, and Clear. The easy way to coat, seal, protect, and stop leaks fast!*. [online] Official Site - Flex Seal® Family Of Products. Available at: https://www.flexsealproducts.com/product/flex-seal/ [Accessed 21 May 2018].

[18] Meola, A. (2018). How IoT in education is changing the way we learn. [online] Business Insider. Available at: http://www.businessinsider.com/internet-of-things-education-2016-9 [Accessed 13 May 2018].

VIII. APPENDIX

Appendix A: System Level Tests

Light, Pressure, Temperature, and Humidity Test

a) Take the Adafruit M0 board with LoRa capabilities and attach the I2C multiplexer shield to it. Connect the light sensor, humidity/temperature sensor, distance sensor, and pressure sensor to the I2C shield.
b) Download the provided program, disconnect the USB, and connect the M0 board to the LiPo battery.
c) Connect the LoRa receiver, download the provided program, and open the Serial Monitor
d) Verify that data packages with an attribute designation of "Lux" are being displayed on the provided computer. Cover the light sensor and verify that the output is now less than 10 lux. Shine the provided flashlight on the light sensor and verify that the output is between 23,750 and 26,250 lux.
e) Place your hand at the minimum distance (10mm), verify that data packages with an attribute designation of "Distance" are being received. Compare the output distance reading with distance being measured by the provided ruler. Verify that the distance reading from the sensor is within 5% of the measured distance. Repeat with a distance of 200mm.
f) Verify that data packages with an attribute designation of "pressure" are being received. Without touching the sensor, verify that the output pressure is less than .5lbf. Now, place the provided weight on the sensor and verify that the pressure is within 5% of 11.24 lbf.
g) Verify that data packages with an attribute designation of "humidity" are being received. Compare the humidity reading from the sensor with a humidity reading from the provided humidity monitor. Verify that the humidity reading from the sensor is within 5% of the reading on the humidity monitor. Repeat this step in a bathroom after a hot shower has been run for 10 minutes and verify the reading is within 5% of 100 rh.
h) Repeat the previous steps but with the Adafruit 32u4 instead of the Adafruit M0.

PASS: The resulting measurements of each sensor must meet, at minimum, the min and max values of the desired range as set forth in the testing process.

Link to Video: https://tinyurl.com/i2c-system-test

Accelerometer and Gyroscope Test

a) Download the provided code to the Adafruit M0 board.
b) Take the Adafruit M0 board with LoRa capabilities and attach the I2C shield to it with no other extraneous peripherals.
c) Connect the FXOS/FXAS sensor to the enclosure via a waterproof connector.
d) Connect the LiPo Battery to the Adafruit M0 board.

e) Rotate the sensor 180 degrees and confirm that the readings on the provided display are greater than 2 rad/s.
f) Rotate the sensor 180 degrees in the opposite direction and confirm the readings are less than -2 rad/s.
g) Position the top of the sensor straight up, apply a slight force upwards force, and confirm the readings are greater than 10 m/s^2
h) Position the bottom of the sensor straight up, apply a slight force upwards force, and confirm the readings are less than -10 m/s^2
i) Repeat the previous steps but with the Adafruit 32u4 instead of the Adafruit M0.

PASS: The results that are received on the provided display at the end of the test must display, at minimum, the min and max cases of the ranges listed in the requirement.

Link to Video: https://tinyurl.com/accel-gyro-system-test

Assembly Test

a) Hand a participant four different shields, an nRF shield, a thermocouple protoshield, a decagon shield, and a MPU6050 shield along with the M0 processor.
b) Start a 10 minute timer.
c) Have the participant connect the Adafruit M0 to each shield and then disconnect the shields.
d) Have the participant connect the Adafruit M0 to the MPU6050 shield
e) Have the participant connect the LiPo battery in the system to the Adafruit M0
f) Have the participant close the lid of the system and view the results on the provided computer to see that measurements are being transmitted from the Adafruit M0.
g) Check to make sure the task was done in less than 10 minutes.
h) Repeat this process with 9 more participants.
i) Verify that at least 9 out of 10 of the participants are successfully able to complete this procedure within the allotted amount of time.

PASS: 9 out of 10 of the participants are successfully able to complete this procedure within the allotted amount of time.

Link to Video: https://tinyurl.com/assembly-system-test

I^2C Multiplexer Test

a) Take the Adafruit M0 board with LoRa capabilities and attach the I2C multiplexer shield to it.
b) Connect eight light sensors to the enclosure via the waterproof connectors..
c) Connect the LiPo battery to the Adafruit M0 shield..
d) Verify that data packages with an attribute designation of “Lux” are being received by the LoRa receiver connected to the provided computer.

e) Verify that the enclosure is receiving the data packages from all eight Lux sensors by confirming that eight measurements are displayed to the screen during each measurement round
f) Repeat the previous steps but with the Adafruit 32u4 instead of the Adafruit M0.

PASS: The provided computer will display measurements from all eight lux sensors while they're connected to the enclosure simultaneously.

Link to Video: https://tinyurl.com/i2c-multiplexer-system-test

Appendix B: System Level Interface Definitions

Interface Name	Interface Type	Specifics
outside_psupply_envin	Environmental Input	○ Sunlight ○ Other: Plug In/out Solar panel
outside_psupply_dcpwr	DC Power	○ Ipeak: 2.5 A ○ Vmax: 5.4 V ○ Vmin: 4.9 V
outside_enclosure_usrin	User Input	○ Other: Allows for the user to configure the system during use due to the modularity of the system ○ Type: User can load code onto the processor, switch out processors,sensors, actuators, or other shields, and can open and close the enclosure to access all parts that are being swapped out.
outside_enclosure_envin	Environmental Input	○ Light: 10-25,000 Lux ○ Temperature (Absolute): 100 F Max ○ Water: 1 inch of water from the base
outside_i2csensor_envin	Environmental Input	○ Humidity: 50-100% ○ Light: 10-25,000 Lux ○ Other: Distance: 1-20 cm ○ Other: Pressure: 0-50 N ○ Other: Accelerometer: -10-10 m/s^2 ○ Temperature (Absolute): 0-30 Celsius
psupply_i2csensors_dcpwr	DC Power	○ Inominal: 0.1 uA Standby Mode 6 uA Operating Mode ○ Ipeak: 100 mA ○ Vmax: 3.6 V ○ Vmin: 2.7 V
psupply_processing_dcpwr	DC Power	○ Vmin: 4.40 V ○ Vmax: 5.25 V ○ Inominal: 10 mA (+/-25%) for basic Feather M0 and Feather 32u4 protos, 100 mA (+/-20%) for Feather M0 with active WiFi chip, 13mA (+/-20%) for Feather 32u4 and Feather M0 with active LoRa chips ○ Ipeak: 300 mA
psupply_actuators_dcpwr	DC Power	○ Inominal: 15 mA (+/- 10mA) during standby mode ○ Ipeak: 1 A during positioning

		○ Vmax: 5.4V ○ Vmin: 4.9 V
psupply_wireless_dcpwr	DC Power	○ Inominal: 15 mA (+/- 5 mA) when transmitting/receiving (nRF) ○ Ipeak: 100 mA (nRF) ○ Vmax: 3.6 V (nRF) ○ Vmin: 3.0 V (nRF)
psupply_spi/sdi-12sensor_dcpwr	DC Power	○ Inominal: 3 mA (+/- 10%) during readings for the Decagon soil moisture sensor. 1.2 mA (+/- 0.3) during readings for the thermocouple. ○ Ipeak: 30 mA for the Decagon soil moisture sensor. 2 mA for the Adafruit universal thermocouple. ○ Vmax: 3.6 V ○ Vmin: 3.0 V
i2csensor_processing_data	Data	○ Messages: All sensors will display the measurements with numeric values corresponding to the ranges set forth in the envin interface. ○ Other: 3.3 V logic level ○ Protocol: I2C
outside_spi/sdi-12sensor_envin	Environmental Input	○ Other: The soil moisture sensor will work in an apparent dielectric permittivity between 1 and 80. ○ Temperature (Absolute): 0-37 degrees Celsius
processing_wireless_data	Data	○ Messages: Messages: Data sent as floating point and integer values. ○ Messages: Arduino code
processing_actuator_data	Data	○ Messages: Instructs actuator how to perform ○ Other: 3.3V Logic Level
spi/sdi-12 sensors_processing_data	Data	○ Messages: The universal thermocouple sends temperature readings from K-type thermocouples in the form of a 19-bit signed integer, which represents the temperature in Celsius multiplied by 2^7. Temperature readings are within ±6 degrees Celsius of the actual value. ○ Messages: The soil moisture sensor reading for apparent dielectric permittivity ranges from 1 to 80 with an accuracy of ±2 from 1 to 40 and an accuracy of ±20% from 40 to 80. ○ Messages: The universal thermocouple reports raw voltage in the form of a 19-bit signed integer from which voltage can be calculated

		according to the following transfer function: Code = GAIN * 1.6 * 2^17 *VIN Where Code is the signed integer sent by the thermocouple, GAIN is either 8 or 32 depending on the VMODE selected, and VIN is the raw voltage reading. Raw voltage readings are within ±2 mV of the actual value. o Messages: The soil moisture sensor reading for temperature ranges from 0-37 degrees C with an accuracy of ±2 degrees C. o Messages: The soil moisture sensor sends readings to the processor in the form of an ASCII string containing 4 numbers delimited by “+” symbols. The first number is the SDI-12 address of the sensor, the second value is the apparent dielectric permittivity, the third value is the reported temperature, and the final value is the electrical conductivity reported by the sensor. o Protocol: Thermocouple utilizes SPI. o Protocol: The Soil Moisture sensor utilizes SDI-12.
wireless_outside_rf	Data	o Protocol: WiFi uses UDP protocol to transmit data. o Protocol: The WiFi board will create its own access point to transmit data. o Protocol: The WiFi board will run in client mode if provided an SSID and password for an existing access point. o Protocol: Data is packaged into OSC bundles. o Protocol: Data values sent in OSC packages are addressed by project, shield, sensor, and data type according to the following format: "/LOOM/[shield][Instance #]/[sensor]/[data type]". o Protocol: OSC bundles sent via LoRa or nRF are encoded as space delimited c strings.
actuator_outside_envout	Environmental Output	o Other: Servo positioning from 0 degree to 170 degrees with an accuracy of ±10 degree o Other: Two Servos positioning individually.

Appendix C: Project Group/Stakeholder/Instructor Responsibilities Document

Group to Stakeholder Responsibilities

- Attend tri-weekly (or as specified by the stakeholder) group meetings & scheduled meetings with stakeholder
- Deliver best effort in executing the project with the stakeholder
- Work effectively as a member of the team and take lead as needed
- Complete project according to standards set by stakeholder – be proactive if changes need to be made including rapid and proficient communication with mentor, team members and instructor
- Provide all documentation to stakeholder for review and feedback

Instructor/Manager to Group Responsibilities

The course instructors and teaching assistants will...

- respect the groups time by assigning only important work
- clearly evaluate each group members submitted work
- Act in the best interest of each student and the group will strive to ensure a positive and realistic environment

Students in the course will...

- respect that assignments in the course are designed to be beneficial to all students in the course
- act in the best interest of each project and group
- will strive to ensure a positive and realistic environment

Group to Group Responsibilities

- Work effectively as a member of the team and take lead as needed
- Communicate with team members on a regular basis
- Complete project according to standards set by stakeholder – be proactive if changes need to be made including rapid and proficient communication with mentor, team members and instructor
- Provide all documentation to the stakeholder for review and feedback

Group Responsibilities Questions

1. How frequently will we meet during the term and for how long? What will we expect for being on time?

We will meet 1-2 times per week during the term, for durations of an hour or more, as fit. Being on time means either we will show up within 5-10 minutes of the agreed upon time or give notice of where we are at within that time frame.

2. How will we communicate with each other and how fast will we respond? What is the backup plan?

We will communicate through Slack. Responses will be prompt, within a day tops. Backup communication will be email.

3. What quality of work is acceptable for work that will receive a group score?

As best quality as we can achieve. If anyone feels that another's work is below the acceptability that is expected, they will make sure to let the team know right away and the team will work to bring the work up to an acceptable standard.

4. What will our group consider to be ‘behind in their work’? Is it different for different types of assignments?

‘Behind in our work’ varies with the goals we set for ourselves. It’s tough to know how long different assignments will take before actually setting out to complete them. You’re “behind” when you haven’t put in the time to try to reach your goals. For hard deadlines as set by the stakeholder or the course instructor, any work that is submitted past those points and will be considered late or behind. If that’s the case, the team will work together to bring the work timeframe back up to the standards expected by the team.

5. How will we handle individuals getting behind in their work? How will substandard group work be handled?

They’ll get themselves back on track. If not, we’ll provide any assistance we can but in the end, it’s on each individual to fulfill their requirements and keep the team on track as a whole.

6. How will we decide to make changes to the system level or other technical changes that affects more than one member of the team??

We will decide to make changes as a group. Any system level change needs to be approved by the group as a whole. For any technical change that affects others, anyone affected needs to be notified immediately.

7. How will we make decisions that require everyone to accept the decision? What is the backup plan?

We talk it over then make a decision once we’ve come to an agreement. If someone does not agree some parts of our group’s decision with reasonable evidence, the others will do best to reflect their opinion and revise the plan.

I agree to fulfill the above responsibilities according to my project role.

Project Title: Team 4: Project Loom

Group Members:

Tyler Inberg	Date: 10/27/2017	Kenny Noble	Date: 10/27/2017
Dongjun Lee	Date: 10/27/2017		

Appendix D: I^2C Interface Verification Tests

Sensor Measurement Test

For this test, the block will have five different sensors connected to a multiplexer which will undergo a variety of inputs through the **outside_i2csensor_envin** interface.

1. Connect the provided microcontroller to the block and load the test program onto the microcontroller.
2. Open up the Serial Monitor in Arduino by going to Tools and clicking on “Serial Monitor”
3. Humidity/Temperature Sensor
   a) Take note of the initial measurement for humidity on the Serial Monitor and compare with the results of the provided humidity sensor.
   b) Place the system in a bathroom after a hot shower has been run for 10 minutes and compare the results on the Serial Monitor with the results of the provided humidity sensor.
   c) Use the provided bag of ice to lower the temperature of the sensor to below 0 C and compare the results on the Serial Monitor with the results of the provided temperature sensor.
   d) Use a hair dryer to raise the temperature to above 30 C and compare the results on the Serial Monitor with the results of the provided temperature sensor.
4. Lux Sensor
   a) Cover the sensor entirely so it is getting nearly no light across it and take note of the lux measurement displayed on the Serial Monitor
   b) Shine a flashlight from the provided cell phone and take note of the lux measurement displayed on the Serial Monitor
5. Distance Sensor
   a) Take note of the initial measurements when no object is in front of the ZX sensor on the Serial Monitor
   b) Place your hand at the minimum distance (10mm) and compare the output distance reading on the Serial Monitor with distance being measured by the provided ruler.
   c) Repeat with a distance of 200mm.
6. Pressure Sensor
   a) Take note of the initial measurements when no pressure is applied.
   b) Place the provided weight on the sensor and verify that the output on the Serial Monitor is above 50 N.
7. Accelerometer Sensor
   a) Take note of the initial measurements with the sensor directly above the testing area and apply a small amount of upwards force to the sensor.
   b) Repeat step a) but with the bottom of the sensor facing upwards and take note of the value being displayed on the Serial Monitor.
   c) Repeat steps a) and b) with the other two axis of the sensor.

PASS: The test passes if all sensors are able to measure and display the minimum and maximum values as displayed in the interface properties.

Link to Video: https://tinyurl.com/i2c-sensor-test

Processing Test

For this test, the protocol and logic level of the **i2csensor_processing_data** interface will be evaluated.

1. Connect the provided microcontroller to the block and load the test program onto the microcontroller.
2. Connect a voltmeter between the SDA pin of the lux sensor and GND and take note of the voltage that is being supplied.
3. Repeat step 2 with all other sensors.
4. Look over the system and ensure that all sensors other than the pressure sensor (not I^2C supported) are connected via an SDA and SCL pin to the multiplexer.

PASS: The test passes if the voltage being supplied by the sensors does not exceed 3.3 V and all sensors other than the pressure sensor are connected to the multiplexer via a SDA and SCL pin.

Link to Video: https://tinyurl.com/i2c-processing-test

Power Test

For this test, the block will be connected to a power supply and the **psupply_i2csensors_dcpwr** interface must be able to supply the appropriate current at both the 2.7 V minimum and 3.6 V maximum.

1. Connect the I^2C sensor block to power and supply the block with 2.7 volts.
2. Connect an inline current meter between the power supply and the Vin pin on the TCA9548A multiplexer.
3. Connect the provided microcontroller to the block.
4. Load the test program onto the microcontroller and take note of the current being drawn while 10 measurement rounds are being taken (approximately 15-20 seconds)
5. Set the power supply to 3.6 volts and repeat steps 3 and 4.

PASS: The test passes if the peak current is never over 100mA during the entirety of the test.

Link to Video: https://tinyurl.com/i2c-power-test

Appendix E: I^2C Sensor Block Bill of Materials

Part	Reference Designator	Value	Manufacturer	Manufacturer's Part Number	Where to Buy	Price	Resources/Documentation
Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout	U1	N/A	Texas Instruments	TCA9548A	Adafruit	$6.95	Adafruit TCA9548A Documentation
Adafruit TSL2591 High Dynamic Range Digital Light Sensor	U2	N/A	AMS	TSL2591	Adafruit	$6.95	Adafruit TSL2591 Documentation
Adafruit Sensiron SHT31-D Temperature & Humidity Sensor Breakout	U4	N/A	Sensirion	SHT31-DIS	Adafruit	$13.95	Adafruit SHT31-D Documentation
Adafruit Precision NXP 9-DOF Breakout Board - FXOS8700 +FXAS21002	U3	N/A	NXP	FXOS8700 + FXAS21002	Adafruit	$14.95	Adafruit FXOS/FXAS Documentation
Sparkfun ZX Distance and Gesture Sensor	U5	N/A	XYZ Interactive	SEN-13162	Sparkfun	$24.95	Sparkfun ZX Documentation
Adafruit Force Sensitive Resistor	R1	Variable	Interlink Electronics	30-81794	Adafruit	$7.00	Adafruit FSR Documentation
10 K Resistor	R2	10K	Yageo	CFR-25JB-52-10K	Digi-Key	$0.10	Digi-Key 10K Resistor
Premium Male/Male Jumper Wires - 20 x 3" (75mm)	N/A	N/A	Adafruit	1956	Adafruit	$1.95 x2 : $3.90	Adafruit Male/Male Wire Purchase
Total Cost:						**$78.75**	

Appendix F: Enclosure Interface Verification Tests

Outside Environment Test

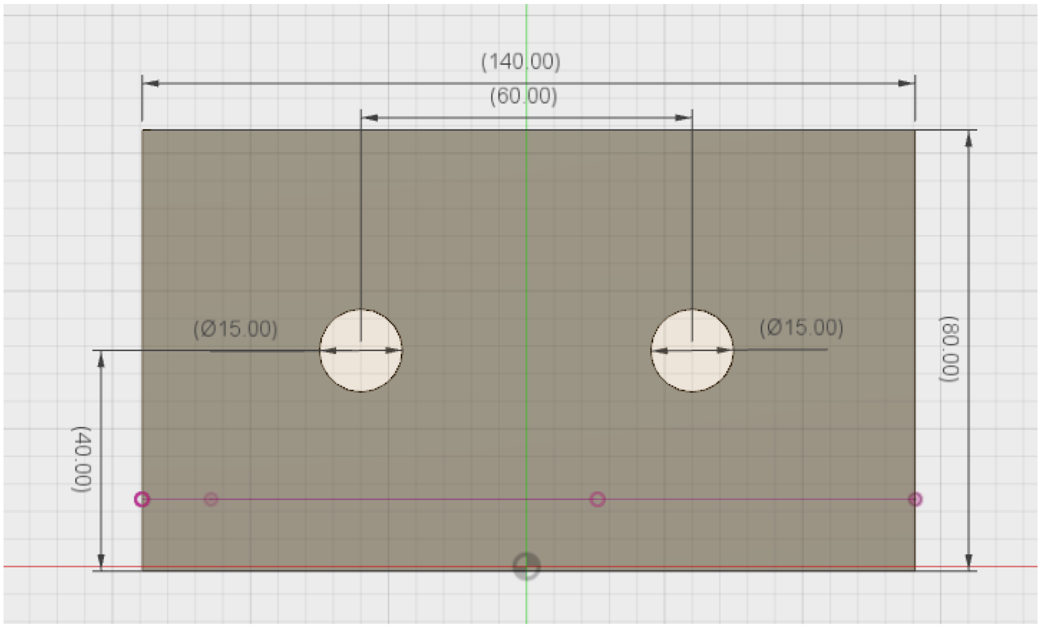
For this test, the block will contain all the physical parts of the system and will undergo a variety of inputs through the **outside_enclosure_envin** interface.

1. Secure all the physical components to the enclosure.
2. Run the test program on the microcontroller and close the enclosure.
3. Complete the Sensor Measurement Test (pg 14) while exposing the enclosure to the following inputs:
    a) Heat the enclosure up to 100 degrees F and note any fluctuations in data.
    b) Cover the enclosure so no light is hitting the enclosure and note any fluctuations in data.
    c) Shine a flashlight from the provided cell phone on the enclosure and note any fluctuations in data.
    d) Set the enclosure in inch deep water and note any fluctuations in data

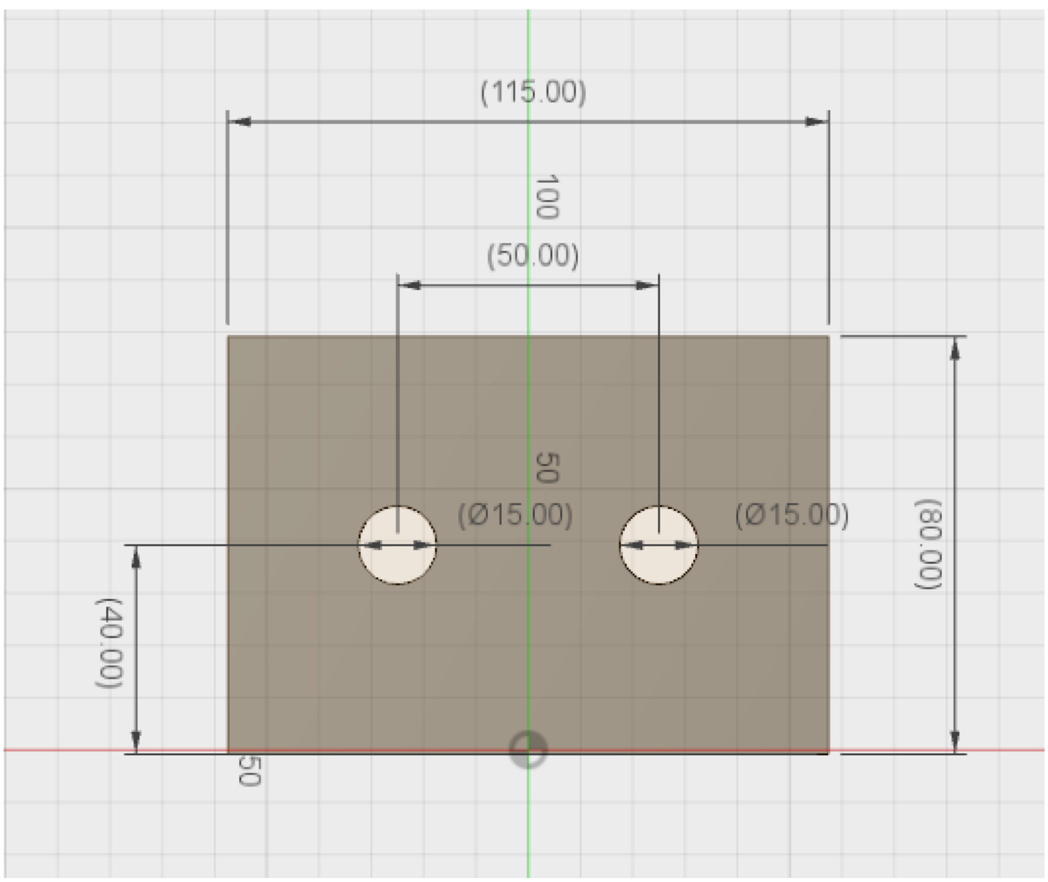
PASS: The test passes if the output of the system is not affected (other than the respective sensor i.e. lux increasing when light is shined on the box, temp reading increases when temperature increases, etc.) by the environmental elements tested.

Link to Video: https://tinyurl.com/enclosure-environment-test

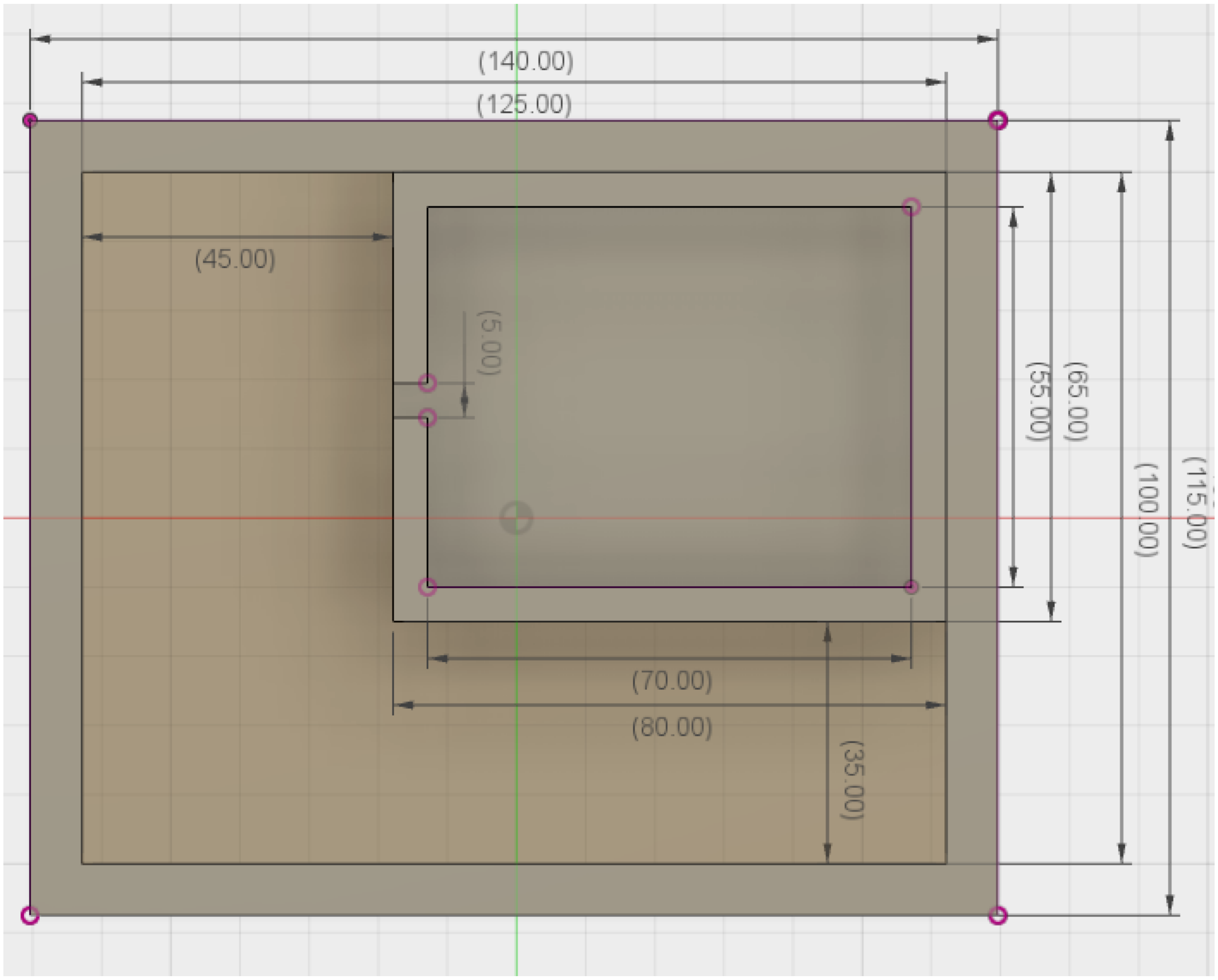
Appendix G: Main Body Measurements of the Enclosure



Front View of Main Enclosure (Units in mm)

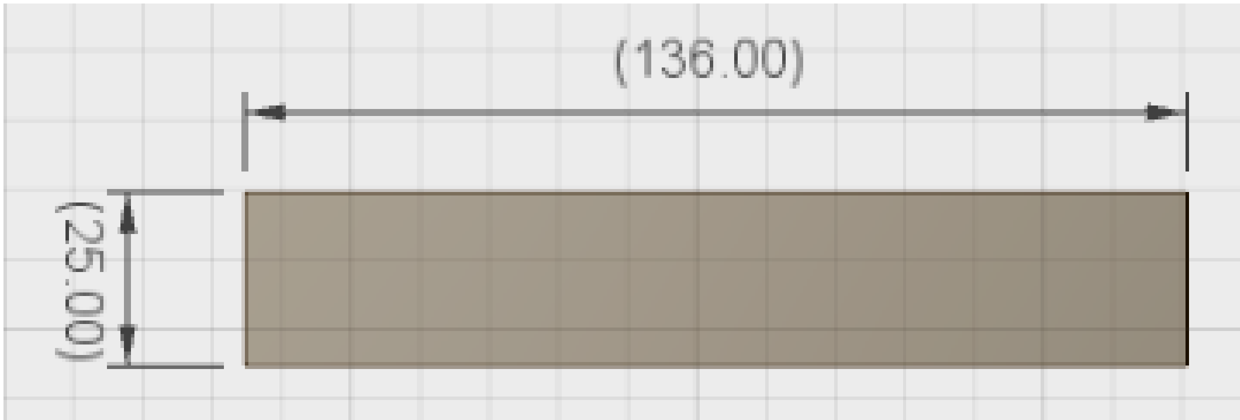


Side View of Main Enclosure (Units in mm)

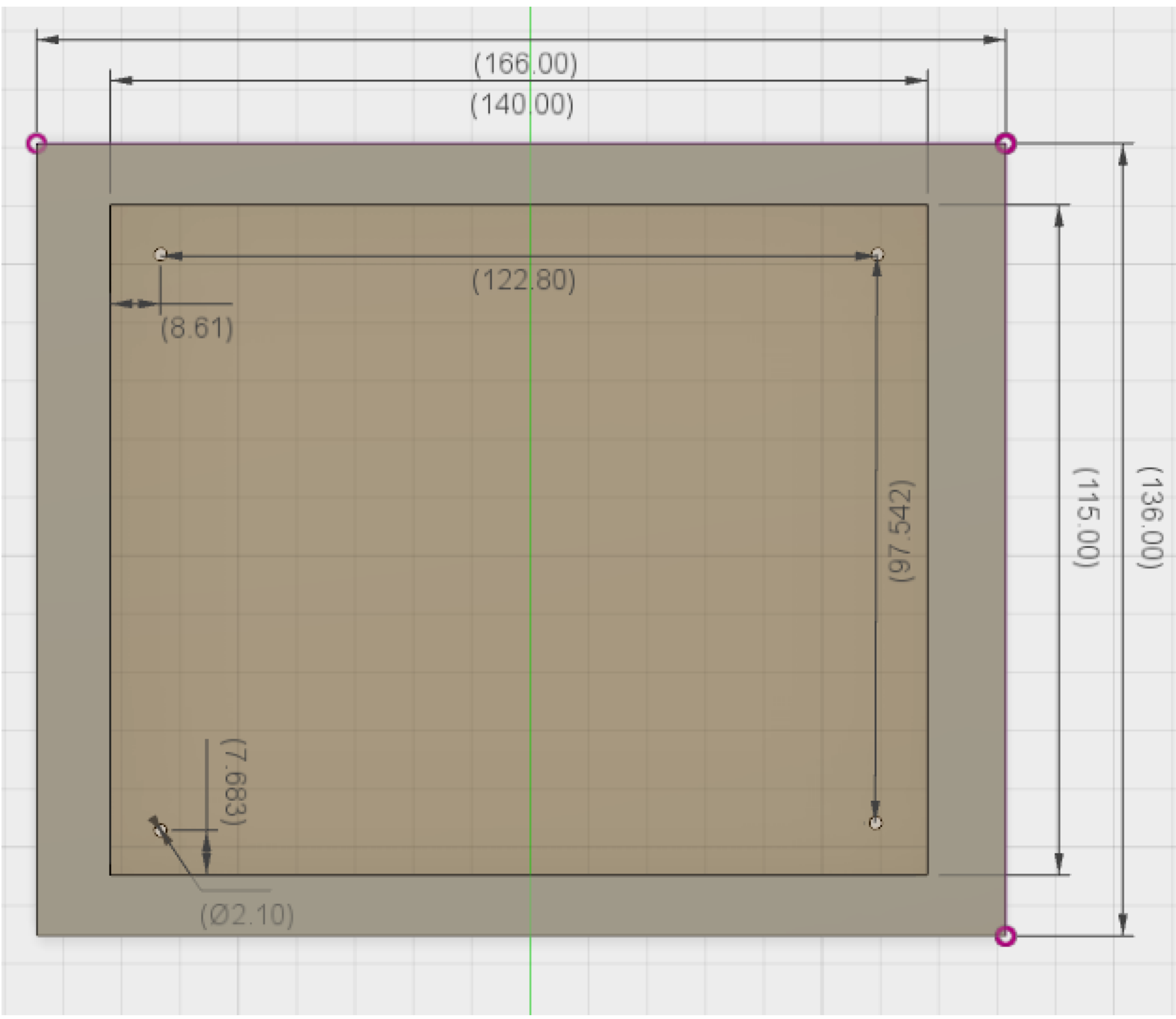


Top View of Main Enclosure (Units in mm)

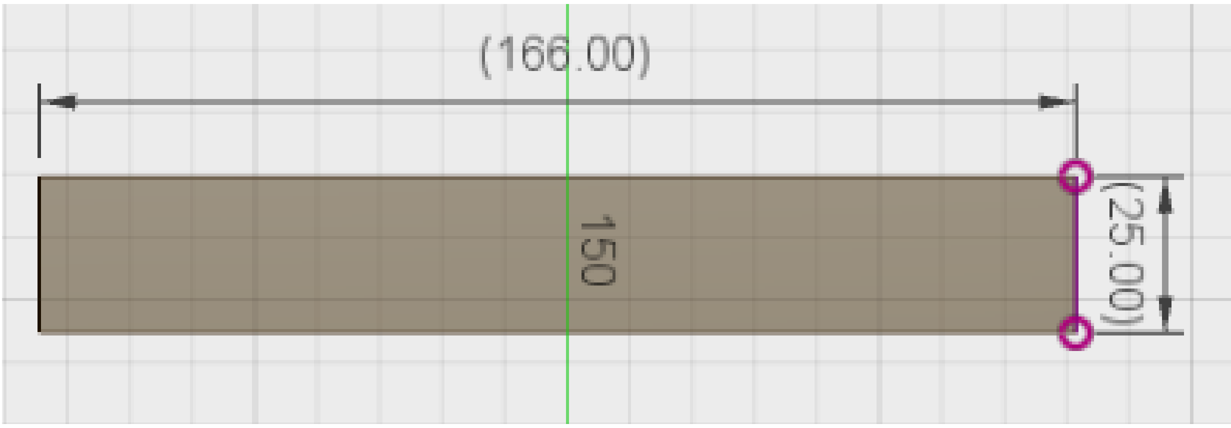
Appendix H: Lid Measurements of the Enclosure



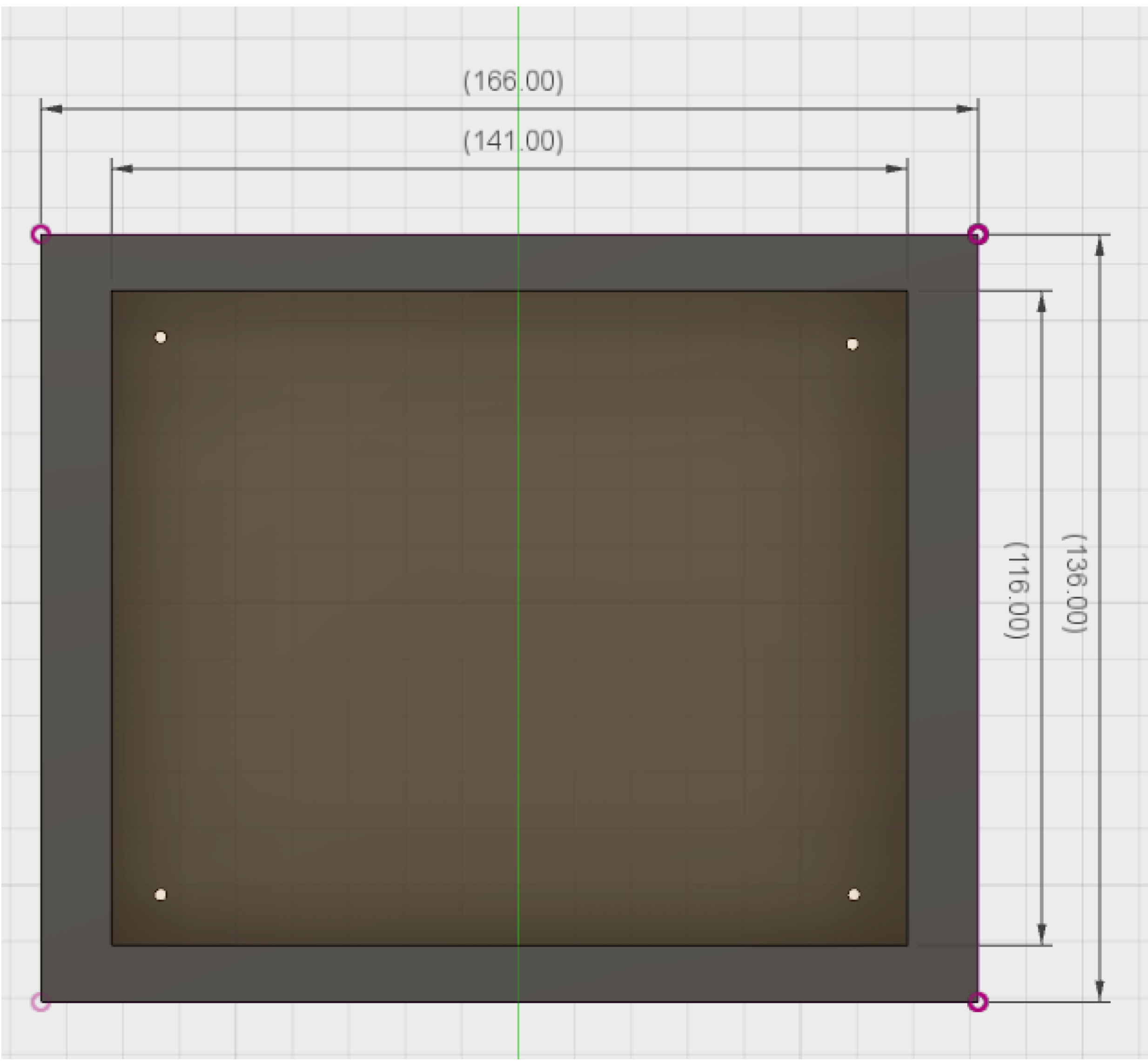
Side View of Lid (Units in mm)



Top View of Lid (Units in mm)

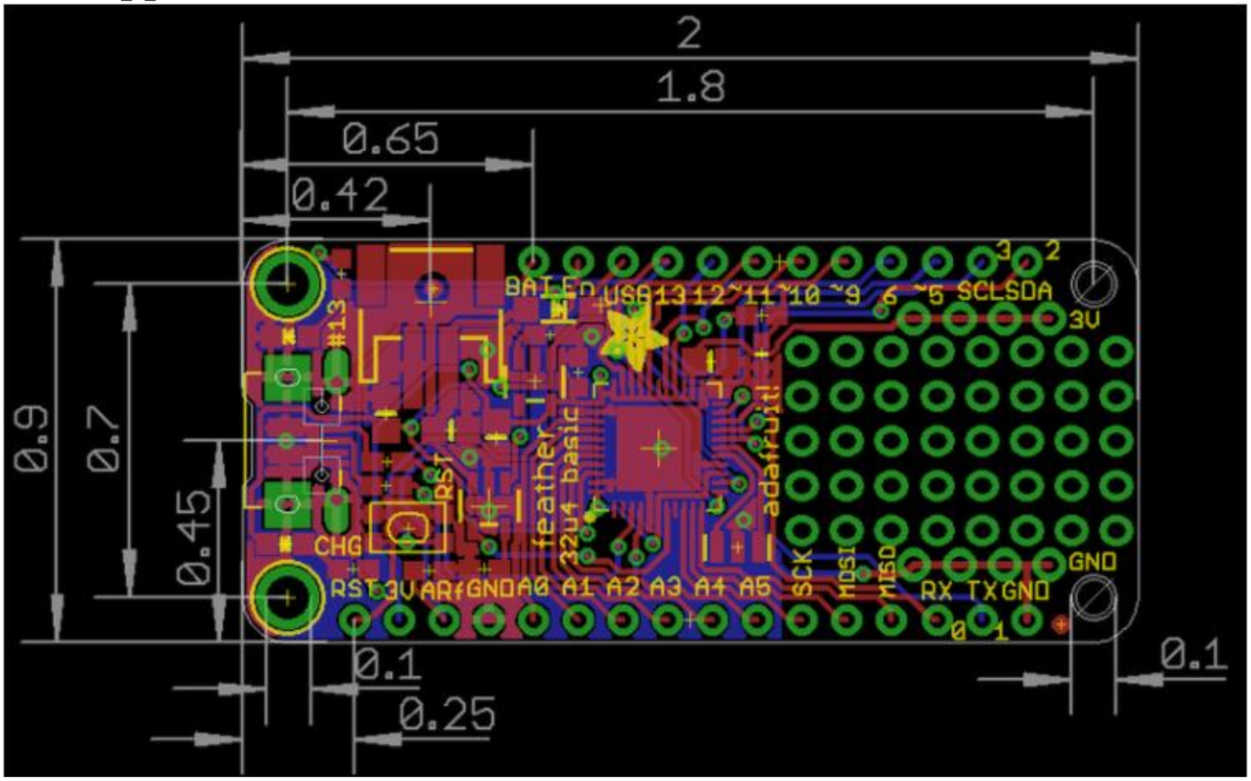


Front View of Lid (Units in mm)

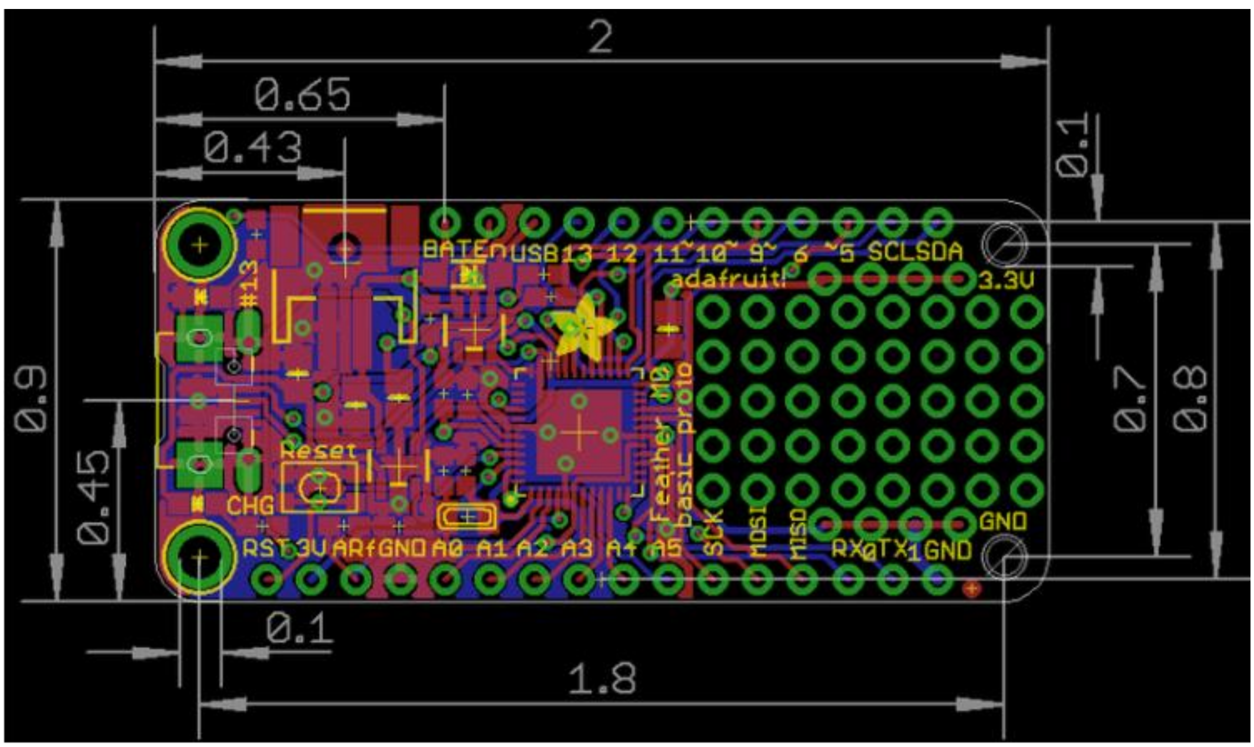


Bottom View of Lid (Units in mm)

Appendix I: Processor Measurements Inside the Enclosure

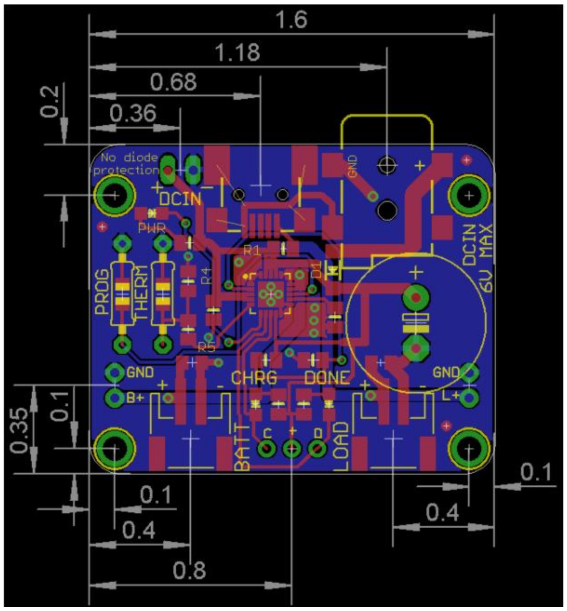


32u4 Processor Dimensions (Units in inches) [8]



M0 Processor Dimensions (Units in inches) [9]

Appendix J: Solar Charging Circuit & Solar Panel Dimensions



LiPo Charger Dimensions (Units in inches) [10]

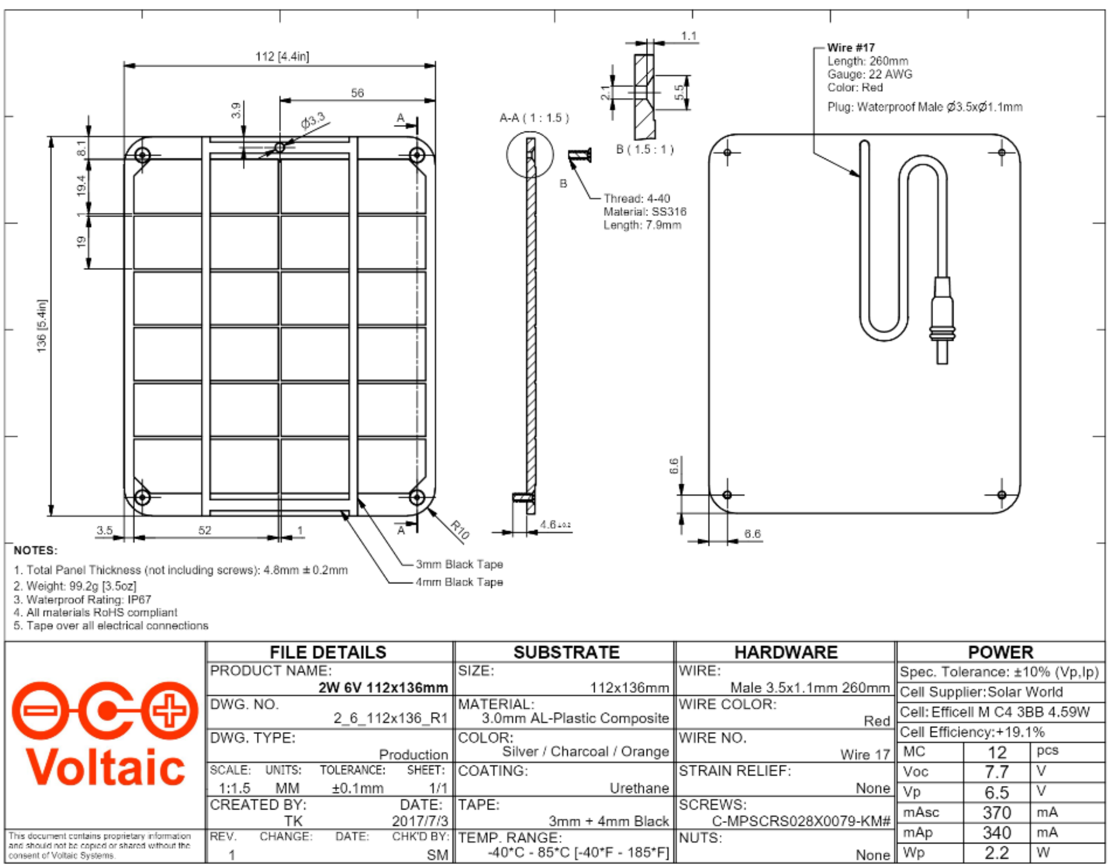
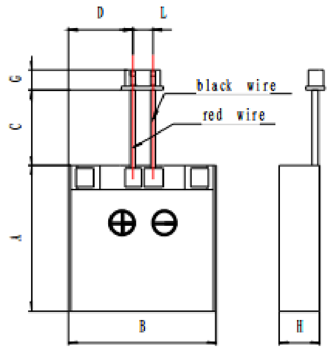
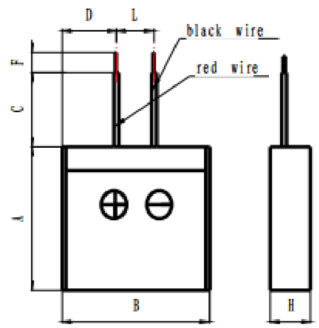


Figure 3.17: Solar Panel Technical Drawing (Units in mm) [11]

Appendix K: Battery Dimensions



☐Cell adding PCM, wire



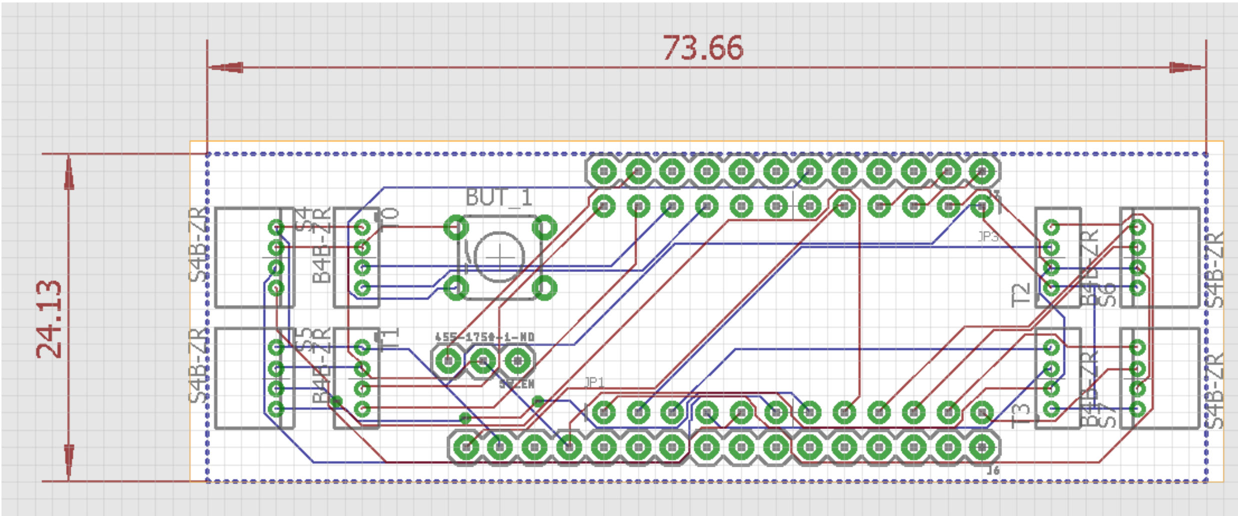
☐ Cell adding wire

Parameter: √

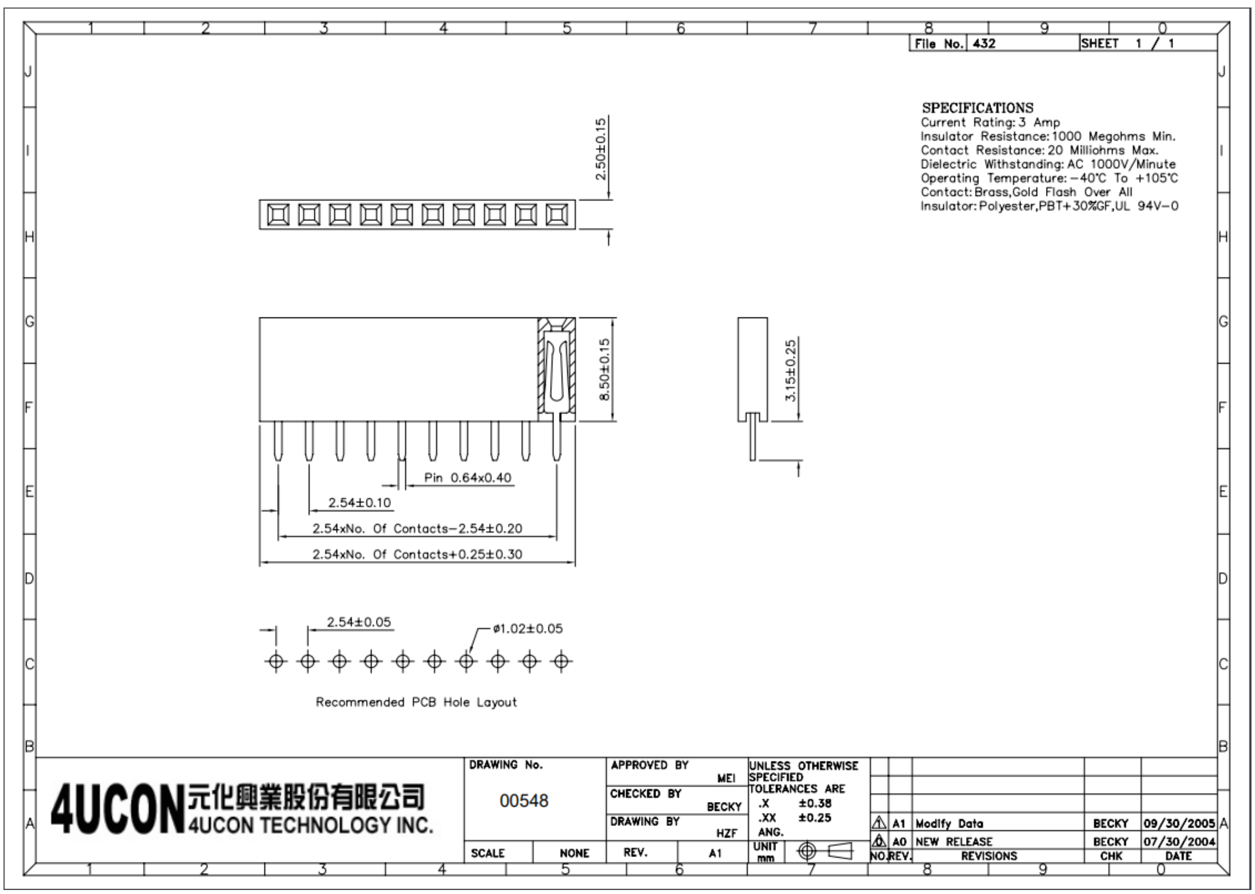
Sign	Item	Max（mm）	Remak	Sign	Item	Max（mm）	Remak
A	Length	60.2		L	Space between Tabs		
B	Width	50.5		E	PP membrane Length	2.0	
H	Thickness	7.9		b	Tab Width	3	
	Wire Standard	/			Plug Standard	/	

LiPo Battery Dimensions (Units in mm) [Page 10, Product Specification [12]]

Appendix L: I^{2}C PCB & Header Dimensions



PCB Board Dimensions (Units in mm)



Header Dimensions (Units in mm) [13]

Appendix M: Enclosure Bill of Materials

Part	**Manufacturer**	**Manufacturer's Part Number**	**Where to Buy**	**Price**	**Resources/Documentation**
Plastic	OSU	N/A	OPEnS Lab 3D Printing	$0.10 per gram x 196: $19.60	OPEnS Lab 3D Printing
Silicone Conformal Coating	MG Chemical	422B-340G	Circuit Specialist	$19.09	Material Safety Datasheet
FlexSeal	Flex Seal Products	N/A	Flex Seal Products	$12.99	FlexSeal Product Description
Waterproof Polarized 4-Wire Cable Set	Adafruit	Adafruit 744	Adafruit	$2.50 x 8: $20	Adafruit Cable Documentation
16-pin Female Header Set	Adafruit	Adafruit 2886	Adafruit	$0.95 x 4: $3.8	Adafruit Header Documentation
#3-48 x 1/2" Machine Screws	Bolt Depot	Bolt Depot 21181	Bolt Depot	$0.08 x 8: $0.64	Bolt Depot Product Details
#3-48 Nuts	Bolt Depot	Bolt Depot 11911	Bolt Depot	$0.05 x 8: $0.40	Bolt Depot Product Details
Thread 4-40 Screws (Comes with Solar Panel)	Voltaic	C-MPSCRS028X0079-KM#	Adafruit	$0.00 x 4 : $0	Adafruit Solar Panel Documentation
Total Cost:				**$76.52**	