A Plant Habitation System for Life Sciences Research

University of Central Florida
National Aeronautics and Space Administration
Florida Space Grant Consortium

College of Engineering and Computer Science

Department of Electrical & Computer Engineering
Dr. Lei Wei
Dr. Samuel Richie

Department of Computer Science
Dr. Mark Heinrich
Dr. Rick Leinecker

Project Information

Standard Small Satellite research platform for Life Sciences research – Plants
Preliminary Design Review

Group Members

Nicolas El Tenn
Computer Engineer

Raquel Guzman
Computer Engineer

Matthew Philpott
Electrical Engineer

Shivani Kumar
Computer Scientist

Noah Heikes
Computer Scientist

Table of Contents

0. Abbreviations

AI	Artificial Intelligence
APH	Advanced Plant Habitat
ARM	Advanced Reduced Instruction Set Computer Machines
BNC	Bayonet Neill–Concelman
CO2	Carbon Dioxide
CSV	Comma Separated Values
DB4S	Database Browser for SQLite
GB	Gigabyte
GHz	Gigahertz
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HPC	High Performance Computing
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
ISS	International Space Station
LED	Light-emitting Diode
LEO	Low Earth Orbit
LPDDR	Low-Power Double Data Rate
MCU	Microcontroller Unit
MSL-1	Microgravity Science Laboratory
NASA	National Aeronautics and Space Administration
O2	Dioxygen
PGBA	Plant Generic Bioprocessing Apparatus
pH	Potential of Hydrogen
U	Unit
UCF	University of Central Florida
USB	Universal Serial Bus
VPN	Virtual Private Network
RGB	Red Green Blue

1. Executive Summary

University of Central Florida Senior Design PlantPod is working to develop a plant habitation system with a human-machine interface for Life Sciences research to produce high-yield plants and reduce crew time operation.

The plant habitation system will be composed of off-the-shelf components as well as components that will be designed and tested in-house. The reason behind this design selection is to abide by the constraints set forth by both UCF and NASA which are quite similar in nature. The constraints are discussed extensively in this document in the Constraints section. From the hardware perspective, the frame is highly likely to be constructed in-house so it abides by the volume and weight constraints. The team is aiming for a 6U CubeSat volume to house the plant(s) as well as the array of sensors, power supply, MCU, wiring, display system, water irrigation system, air filtration system, lightning system, temperature system, etc. The power supply module will be within the limitation of the spacecraft allowed wattage and tested to that threshold. The weight requirement will also be heavily considered when designing this system as it limits the selection of components based on their weight. Moreover, the components will also be selected based on their thermal limitations and heat emissions which would affect not only the system, but the plant(s) on board.

From the software perspective, for the frontend, a GUI will be developed and designed to exhibit the plants health and growth. The GUI will gather information and data from the plant and store it in a database for researchers and scientists to look at. The software will get data input from the sensors and in return it will give out control output. The data gathered will be from sensors that monitor the internal conditions of the plant habitation system as well as sensors that monitor the plant's soil conditions. This sensor data includes humidity, temperature, pH, CO2 levels, pressure, water levels, and light levels. As a result, reporting back data on plant status, providing insight on growth and yield. The ability to grow supplemental food through plant crops would allow humans to explore deep space with long-term missions. Additionally, growing food in space is crucial since it provides a solution to rid the constant costly supply of packaged food from Earth that is currently the standard for a permanent presence in space.

The GUI specifically will be composed of 3 main functions. The three functions are a Login page, a Sensor Dashboard, and a Growth Timeline. The login page will allow users, such as researchers and scientists, to create accounts or login into their

pre-existing accounts. Here, they can manage their account and create different notes. They can also edit and delete these notes. They can input whatever they need into these notes such as updates on the plant's growth, health and environment. The notes section will also allow the user to view associated data and notes in the Growth Timeline. The Growth timeline is composed of two functions: Compare Data and Viewing Reports/History. In Compare Data, the user will be able to compare the growth and health of the plant with the history of the plant to see the progress. Here they can check to see if certain changes to the plant habitation system are assisting the plant or harming it. They will also be able to see the history of the plants and see the different notes others have put in for it. The user is also able to see the history of the notes in the View Reports/History tab. The user is able to check out the reports of the health of the plant. The last section is the Sensor Dashboard. Here the user can see the Current Sensor Data of the plant that is currently in the plant habitation system. All the sensor information from temperature to light levels will be displayed here in real time.

Since this project is also being used for NASA's Kennedy Space Center Senior Design Program, there are hopes that by showing proof of concept, scientists are able to study and provide solutions to NASA's aerospace problems and projects. NASA has been funding senior design projects for a couple years and they claim that the Senior Design studies have piloted at Kennedy, with excellent results [1]. Therefore, this project will enable humanity to explore the solar system and the surrounding solar systems without the dependency on Earth-supplied packaged food.

2. Project Overview

2.1 Project Description

The team is working to develop a plant habitation system with a human-machine interface for Life Sciences research to produce high-yield plants and reduce crew time operation. The data gathered will be from sensors that monitor the internal conditions of the plant habitation system as well as sensors that monitor the plant's soil conditions provided none of the component parts exceed the budget and timeframe thresholds allotted. These sensors will then be reporting back data on plant status, which provide insight on growth and yield. A camera and a ruler inside of the habitation system will also be providing updates on the growth and condition of the plant. Using Python, a UI will be made which will contain all the data from the sensors. The data that will be recorded are the RGB lights the plant is growing in, how much water, what are the levels of CO2 and O2, and the growth status of the plant. The data will also gather information on the soil such as pH level, humidity, and the temperature. Using this data, the GUI will be able to come up with an ideal scenario in which the plant is likely to thrive. This scenario will be used by future astronauts for research on which conditions plants thrive the most in. This research will help humanity understand how to experience smoother long-duration missions and how plants can flourish in space.

2.2 Statements of Motivation

2.2.1 Nicolas El Tenn

My previous experiences with NASA have given me insight into what it takes to research and develop space-grade technology. I have had first person experience at Kennedy Space Center where they study plant habitation systems amongst other things. There, I visited Veggie where I was exposed to their plant habitation system design. So, the team aims to meet all the requirements for the design while abiding by the sponsor's (NASA) solicitation. My academic background is in Computer Engineering, so I hope to contribute to this project from both the hardware and software side. I have experience with electronics such as embedded systems and sensors as well as programming such as data structures and algorithms.

My interest in this project stems from my fascination with space at an early age. I believe in a future where humanity becomes interplanetary, expanding the scope and scale of humankind’s imprint on the Universe. For that, the technology that will allow us to live elsewhere besides Earth must be developed. However, being human propels us to take a part of Earth with us ensures the connection to the mother planet is maintained during deeping space travel. This organic touch on space exploration allows us to live for prolonged periods of time in space without the dependency on Earth-supplied packaged food. As a result, developing a plant habitation system for microgravity is crucial to the collective future in space. This vision also aligns well with that of NASA as humankind prepares to explore from the Moon to Mars.

2.2.2 Shivani Kumar

I’ve been in love with space from a very young age. Many people around me are afraid to talk, explore, or even learn about space. Although that is understandable since the majority of it is unknown, the curiosity in me only wants to learn more about it. Space has always been an area that interests me, but I have never been able to work on a project that combines my computer science skills with my space knowledge. Now that I have been presented with this opportunity to work on an incredible project with team members that have the same interest in space as me, I’ve been looking forward to implementing my knowledge and learning more from them as well.

Space is the next big thing. In a couple million years, the Sun is going to consume Earth and the human race will die off. The only way to prevent this is to start exploring space and the planets that lay within the galaxy. Humanity has been given the freedom to pursue this plan and I think it is very important that this opportunity is seized. As exploration expands our knowledge , it will aid in the discovery of furthering evolution and starting a new life in a new place. If I can help out humanity even in the slightest way by being able to expand the knowledge of growing food in space, mankind will change entirely and space will become humankind's new home. As humans explore deeper into space, constantly packaging food for astronauts will not be ideal nor will it keep the astronauts healthy. Not only will humanity evolve with this technology, but so will humanity's future in space.

2.2.3 Noah Heikes

Space has always been the next big step in humanity's future. Whether in the next hundred years or the next ten, humankind's path forward as a species will always be off of this planet. But the technology level needed to truly begin colonising space is yet to be achieved. Research on the effects space has on humans, how to create a sustainable atmosphere, how gravity works, and more still needs to be accomplished before humanity can move forward.

With that in mind, the first step towards colonising other planets and exploring space is a stable food source. Without food (and air and water, but that's a problem for another project), humans can't survive anywhere at all. I want to have a hand in building the technology that will help humans to survive outside of Earth's atmosphere. If this project is able to help further the goal of making long term space travel possible, then I want to contribute as much as I can.

2.2.4 Matthew Philpott

Growing up on the Space Coast, the view of a rocket launch was just two steps from my front door. Because this reason space has always felt accessible. With that along with sci fi movies and tv shows dreams were inspired to make space accessible to the average man. The access to space will allow for many things including the reduction in cost for average goods because humans will have access to different planetary bodies and natural resources. As well as allow for the human race to become more than a type one civilization.

I am interested in Plant-Pod because when I imagine a future, I see a civilization that is not limited by the gravity of earth. Space is this next frontier, with that comes similar challenges the unknown will bring. From the beginning of humanity plants and humans have lived in coexistence. This means millions of years ago humans needed plants to survive, and will need to continue this symbiosis in order to make space as well as bodies in space a permanent habitable place for humans. This is why Plant-Pods studies can teach us how to determine plants yield, and hopefully maximize it. This goes along with my long-term goal in life is to explain to my family or friends how I had a lasting effect on meeting my expectations for that type of future.

2.2.5 Raquel Guzman

I personally wanted to be involved in this project because of my interest in space. I believe that space is the next frontier and efforts to continue exploration should be supported. Right now astronauts mainly eat prepackaged meals and have constant resupply missions. As humanity continues to explore and travel deeper into space, constant resupply missions will become impossible and the need to bring plants to keep astronauts healthy is vital. The plant pod would be helping these efforts and allow for humans to travel and survive for extended periods of time in deep space.

For the past three years I have been working for UCF's Center for Microgravity research and have worked on microgravity experiments. While there I have worked on ground-based experiments, which never travel on flights but give great insights on how certain projects will interact in microgravity. Others were flight-based experiments that needed space grade materials, and traveled on parabolic flights. Regardless of experiment type, each one has required intensive investigation and research to properly develop and design, as well as testing plans to ensure the hardware and software components worked properly. I hope to apply the knowledge I have gained from working there throughout senior design, but ultimately contribute to humanity's efforts in space exploration.

2.3 Project Objective

2.3.1 Goals

The following Project Life Cycle will be abided by as provided by NASA Systems Engineering Handbook to best efforts [72]:

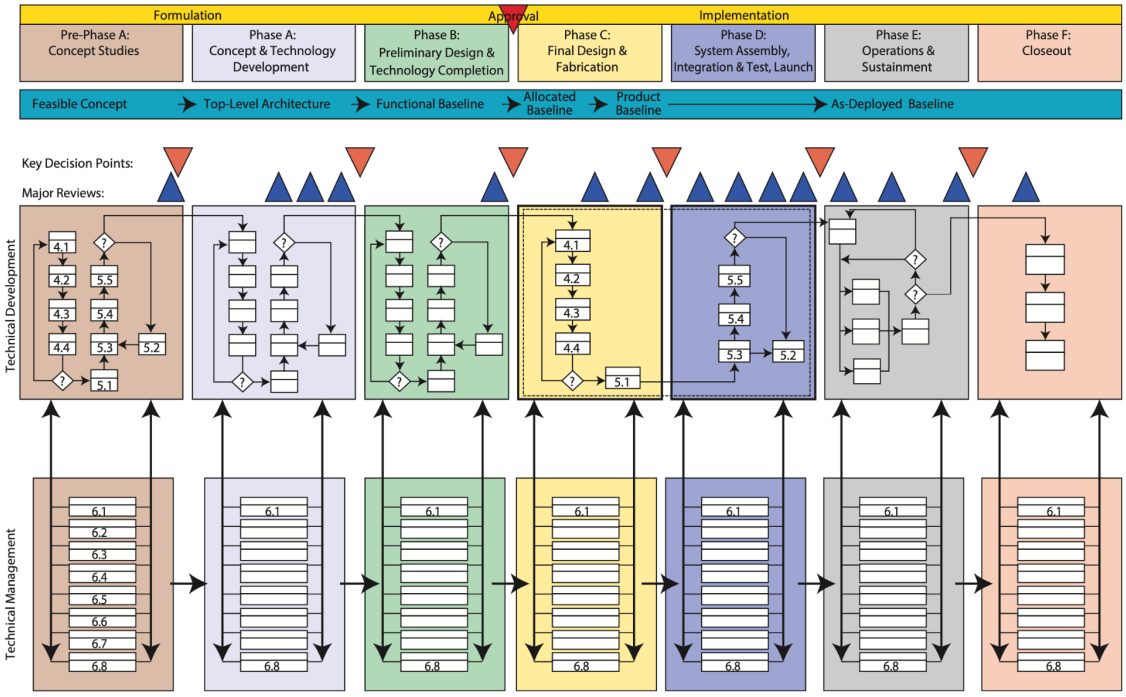
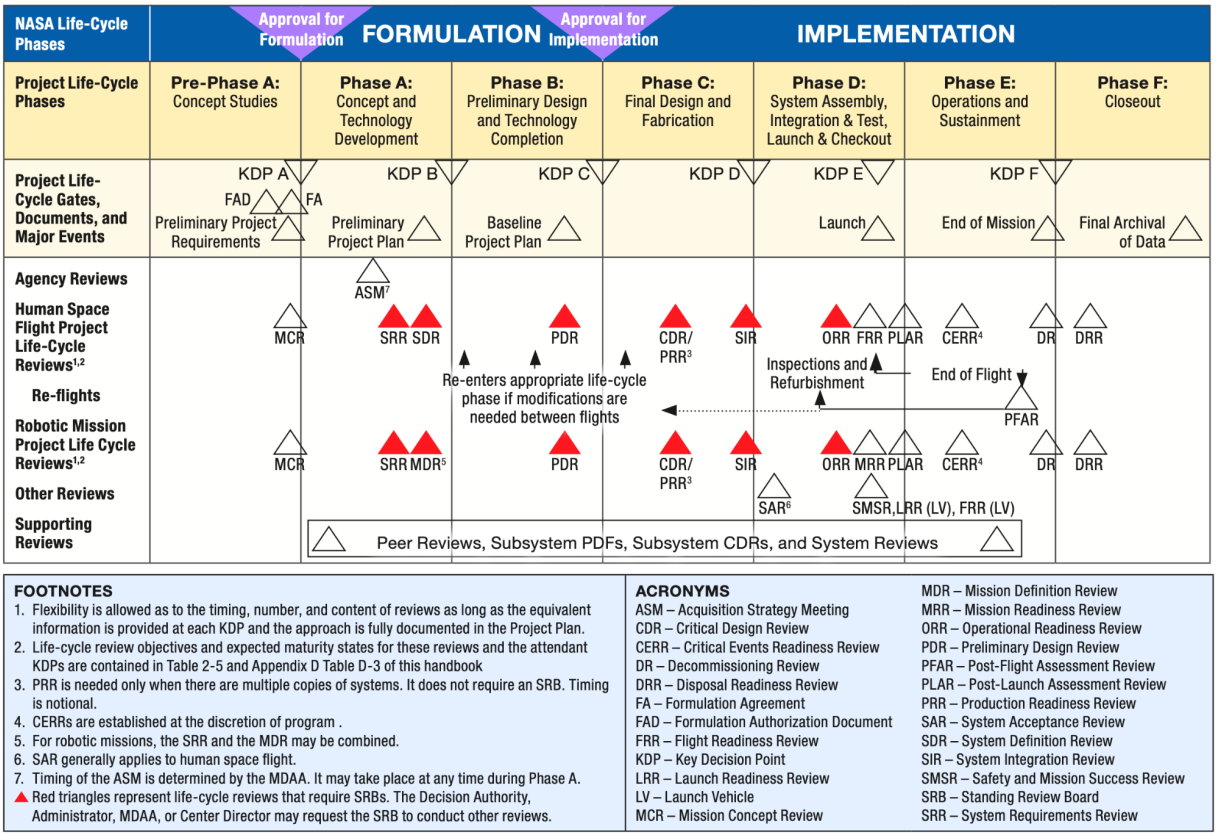


Figure 1: NASA Project Life Cycle

This project is to continue contributing to agency goals as well agency objectives. Moreover this project is to be confined within the budget and time constraints. Furthermore, the above Project Life Cycle is to assist in establishing a cost efficient program that adheres with both Agency and mission directorate goals as well as objectives. [72]



FOOTNOTES

1. Flexibility is allowed as to the timing, number, and content of reviews as long as the equivalent information is provided at each KDP and the approach is fully documented in the Project Plan.
2. Life-cycle review objectives and expected maturity states for these reviews and the attendant KDPs are contained in Table 2-5 and Appendix D Table D-3 of this handbook
3. PRR is needed only when there are multiple copies of systems. It does not require an SRB. Timing is notional.
4. CERRs are established at the discretion of program .
5. For robotic missions, the SRR and the MDR may be combined.
6. SAR generally applies to human space flight.
7. Timing of the ASM is determined by the MDAA. It may take place at any time during Phase A.

▲ Red triangles represent life-cycle reviews that require SRBs. The Decision Authority, Administrator, MDAA, or Center Director may request the SRB to conduct other reviews.

ACRONYMS

ASM – Acquisition Strategy Meeting
CDR – Critical Design Review
CERR – Critical Events Readiness Review
DR – Decommissioning Review
DRR – Disposal Readiness Review
FA – Formulation Agreement
FAD – Formulation Authorization Document
FRR – Flight Readiness Review
KDP – Key Decision Point
LRR – Launch Readiness Review
LV – Launch Vehicle
MCR – Mission Concept Review
MDR – Mission Definition Review
MRR – Mission Readiness Review
ORR – Operational Readiness Review
PDR – Preliminary Design Review
PFAR – Post-Flight Assessment Review
PLAR – Post-Launch Assessment Review
PRR – Production Readiness Review
SAR – System Acceptance Review
SDR – System Definition Review
SIR – System Integration Review
SMSR – Safety and Mission Success Review
SRB – Standing Review Board
SRR – System Requirements Review

Figure 2: NASA Project Life Cycle

Upon completion, the plant habitation system will be capable of:

- The ability to grow supplemental food through plant crops would allow humans to explore deep space with long-term missions.
- Growing food will also provide a solution to rid the constant costly supply of packaged food from Earth that is currently the standard for a permanent presence in space.
- It will prove that humans will be able to grow crops which will maintain the astronauts vitality and health.
- The system will have a GUI diagram written in Python, which will display the sensor information of the plant so that it has the ideal conditions to grow plants.
- The sensors will take in data from O2 levels, CO2 levels, water levels, soil health, and growth of plants to compare.
- The camera will take pictures of the plant in order to compare and view the health of the plant.

- The GUI diagram will display all information of the plant and how to maintain the health of it, using this the ideal situation in which the plant will thrive will be found.

The success of the plant habitation system is fully dependent upon the completion of these goals. If a goal is missed, the plant habitation system will face difficulties maintaining plant growth rate, maintaining the plant health, and supplying accurate and sufficient data about the plant conditions. As discussed in later sections, the successful implementation of each individual subsystem will also prove to be critical to the system's functionality.

2.3.2 Stretch Goals

Once the above goals are met, the system will aim to:

- Use Computer Vision to better the data and automate how it looks
- Use synthetic data to boost automated image-based plant phenotyping
    - Compare the number of leaves of plants from the dataset to the plant for better health and growth
    - Use 10,000 top down images of Arabidopsis plants
    - Use dataset provided by Leaf Segmentation Challenge of the Computer Vision Problems in Plant Phenotyping

It would be ideal for the plant habitation system to use computer vision, pre existing leaf segmentation datasets, and the Arabidopsis images. They could be used to determine the best course of action for a plant's health and growth. Computer vision could be used to process the images provided by the cameras and help count the number of leaves in a given plant. Using the processed data and other datasets, it could then be determined if a plant requires more light or water for optimal plant yield. Ultimately, each of these components are not necessary for the main functionality of the system. These goals have a lower priority and will only be completed if there is enough time.

2.4 Broader Impacts

2.4.1 Planetary Scale

Since the rise of agriculture 10,000-12,000 years ago, humanity has seen nothing but astronomical progress. Before agriculture, humans were hunter gatherers. Humans mostly lived in groups of 50 people or so, and wandered around looking for food all day long. This not only required a large portion of one's day, but also one's energy. [20]

Humans back then had their mind set on survival all day long to the point where they had little to no time for anything else. As agriculture rose, so did free time. That meant more time to make tools, houses, irrigation systems, etc. Humanity then began to settle. [20]

As people started reaping what they sowed, their productivity increased. Cultivation areas grew bigger, and so did settlements. Cities started forming and crop trade started to take place. The world was starting to form. After that, industrialization took the globe by storm resulting in a global shift of lifestyle. [20]

Humans then moved on to develop systems that would grow life in outer space. Potatoes were the first plants to grow in space in 1995. NASA and Wisconsin-Madison studied plant life along with optimization for high yield in controlled environments. The research was largely focused on underlining the environmental impacts on the plant's growth. As a result, the research would then yield a series of data points that would have influence on the final output. [21]

Moving forward, NASA would continue to push efforts to grow plants in space. In June of 1997, the Space Shuttle Columbia's STS-94 mission would have the PGBA on board ready to be deployed on the ISS. That served as a testbed on the ISS, but would also give humanity insight on what to focus on when cultivating plants commercially here on Earth. [22]

2.4.2 Interplanetary Scale

Plants and humans have a long history of interconnection where humans have a need for plants to ensure survival. Plant-based research sheds light on this and digs deeper to help understand this complex relationship. Studying plants in microgravity could give more insight on how humans can work and live in space for long periods of time. The team is working to develop a plant habitation system with a human-machine interface for Life Sciences research to produce high-yield plants and reduce crew time operation. By using interchangeable research modules that work in Low Earth Orbit, a Lunar Outpost, Lunar Orbit, on Earth, and on the Moon, the advancements and discoveries made through this development can then be implemented in all locations. As a result, benefiting humankind entirely.

This project aligns well with NASA Taxonomy TX06: Human Health, Life Support, and Habitation Systems. Furthermore, this project specifically falls under TX06.3.5: Food Production, Processing, and Preservation. This section of the Taxonomy focuses on the following example technologies directly extracted from the Taxonomy [71]:

- *Bioregenerative food system*
- *Vegetable production system*
- *Packaged food mass reduction*
- *Vegetable cleaning and safety verification • Stabilized foods*
- *Low oxygen permeability barrier films*
- *Plants habitat*

The team hopes to contribute to the following technology focus areas with this project, as meeting the requirements is a top priority for the team.

2.5 Constraints

The plant habitation system must be in the range of a standard CubeSat. Accordingly, there are restraints to the project's volume and weight. Additionally, the constraints set forth by the College of Engineering and Computer Science as well as NASA and FSGC.

Constraint	**Description**
Volume	The dimensions of a standard CubeSat are 10 cm x 10 cm x 10 cm per unit with units ranging between 1U, 2U, 3U, or 6U.
Weight	Within the CubeSat range, the system must weigh less than 1.33 kg or 3 lbs per U.
Time	The system will be in space for a few months. Sensor data, water levels, and photos will need to be measured and recorded frequently to accommodate the plant's limited time in orbit.
Budget	This project has received funding from the Florida Space Grant Consortium, however the team would hope to minimize costs when possible. As of right now, the team has estimated costs around $1600.
Testing	The testing environment that will be used to test the hardware will be the Senior Design labs located at Harris Engineering Center, University of Central Florida, Orlando FL. The equipment that will be used is provided by University of Central Florida. The equipment includes, but not limited to, a function generator, an oscilloscope, a multimeter, a logic analyzer, etc.

Table 1: Constraints

2.6 Requirements Specifications

Based on the objectives, stretch goals, impacts, and constraints listed above, the system will need to fulfill software and hardware related engineering requirements.

To maintain accordance with NASA Requirements, the following documentation provided in the Systems Engineering Handbook will be followed to best efforts [72]:

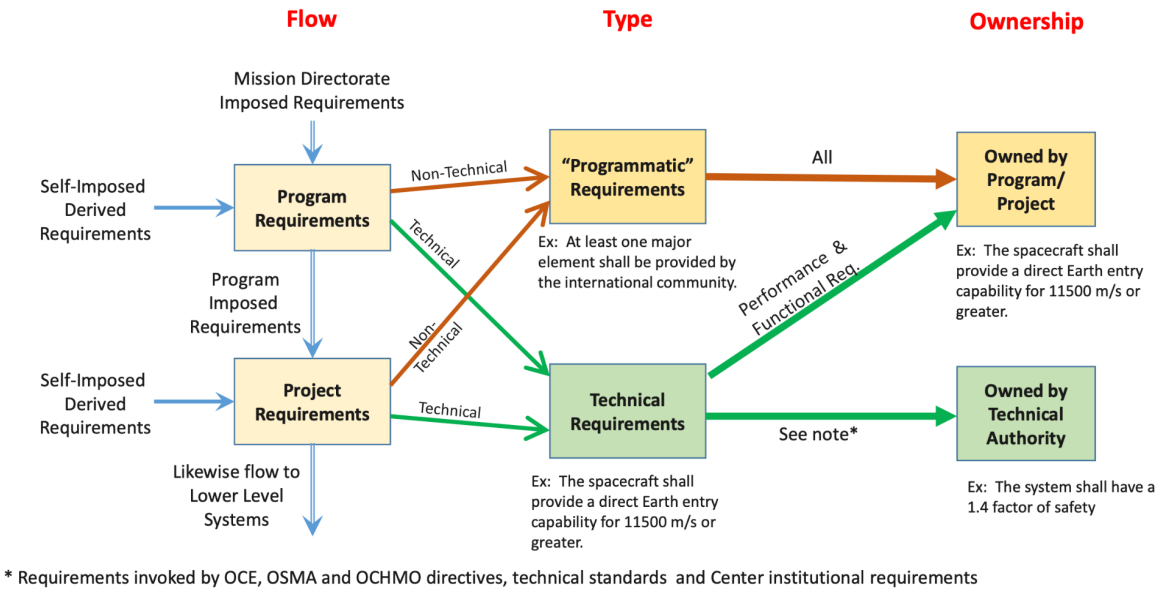


Figure 3: Type, Flow, and Ownership of Requirements

2.6.1 Software Requirements

The following list is used to define software requirements:

- Software will be written in Python.
- The user interface will display sensor data, light settings, and images.
- The GUI will allow users to log in to see their past notes, documents, and photographs from different dates.
- The GUI will allow users to compare data and photographs from sensors from different dates.
- The GUI will allow users to edit, delete, and change any data that they have written.

- The GUI will be checked at least twice a day (once in the morning and once at night).
- The camera will take 3 pictures daily and upload them to the GUI.
- Pictures will be compared every 2 weeks.
- Optimal plant settings will be adjusted every 2 weeks based on gathered data (mainly light settings, and water level).
- Settings are also manually adjustable within the GUI.
- The GUI will notify the user if the plant is ready for harvest, or if other components in the system need assistance.
- Display technical habitation system environment information when requested:
    - Temperature sensor data
    - Humidity sensor data
    - CO2 sensor data
    - Pressure sensor data
    - pH sensor data
    - Water level
    - LED light level
- Display plant-specific information when requested:
    - Real-time plant health monitoring
    - Expected size growth
    - Plant classification
- Display planning tools:
    - Task procedures
    - Daily planning
    - Irrigation procedures
    - Plant harvesting procedures
- Document processing functions:
    - View documents
    - Edit documents
    - Take notes

2.6.2 Hardware Requirements

The following list is used to define hardware requirements for the system:

- Multiple sensors to measure CO2 levels at least 3 times an hour.
    - 3.25V - 5.5V power
    - ±3% accuracy
- Multiple sensors to measure soil moisture and humidity 2-3 times every hour.
    - 3.3V - 5V power
- Individual component to measure remaining water levels. Sends signal to microcontroller if water levels are low.
    - 12V
- Camera and ruler to determine plant height
    - 720p - 1080p resolution
    - 30fps - 60fps frame rate
    - 3.3V - 5V power
- Water-resistant components and waterproofed electrical connections
- Individual sensor to measure the system's temperature. Measured every hour.
    - 2.4V - 5.5V power
- Sensor to measure pH level of plant's water source once every hour.
    - 2.4V - 5.5V power
- System-wide reliable connection
    - Jetson Nano will connect sensors to software
    - PCP and Jetson Nano will regulate power to the system

2.6.3 Overall Requirements

The below figure portrays The Process of Logical Decomposition extracted from NASA Systems Engineering Handbook [72]:

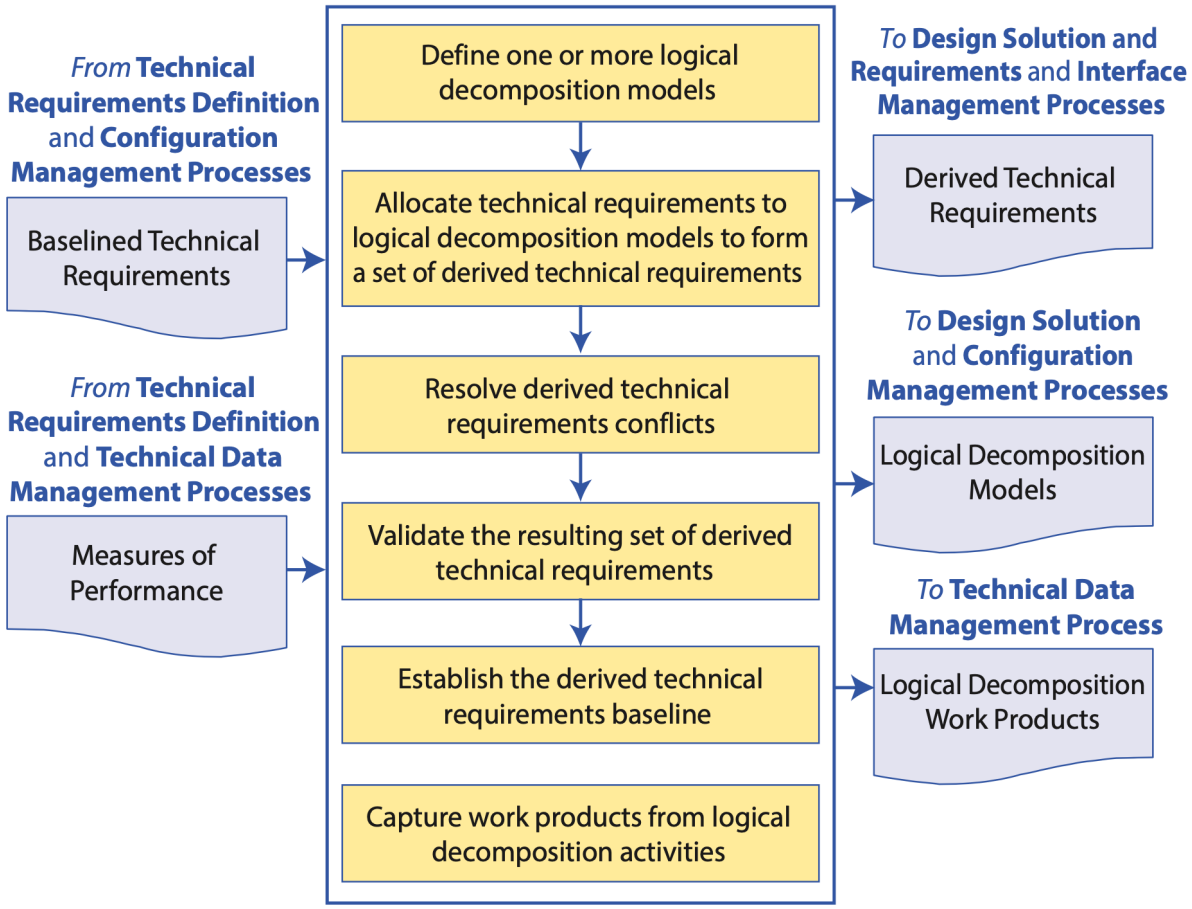


Figure 4: The Process of Logical Decomposition

Through this process, the requirements are to include the following:

- The plant habitation system will have sensors, a microcontroller, a display module to aid in system-wide reliable connection.
- To gather soil data, sensors will be used for the pH levels, humidity, and temperature inside the CubeSat.
- GUI will be connected to the habitation system using a Jetson Nano and Arduino.
- The microcontroller being used is the Arduino Mega 2560.
- The Arduino is in charge of dealing analog and digital connections with the sensor utilizing and has its own libraries for each sensor.

2.7 Division of Labor

The following list is used to define the division of labor between group members including their roles and responsibilities for the project. Note that some of the listed responsibilities overlap due to the extensive nature of their requirements.

2.7.1 Shivani Kumar

- Create and deploy a home screen.
- Design and create a Login/Sign Up page to enable data access
- Allow users to manage their account and personal information
- Create and maintain documents, notes, and files for research
- View automated schedules for sensor data and allow for modifications
- Integrating computer vision

2.7.2 Noah Heikes

- Design and create a User Interface for comparison page
- Integrate a database for sensor data storage
- Backend logic for comparing data
- Create and deploy a User Interface for the sensor page
- Integrate sensors page with comparison page
- Integrating computer vision

2.7.3 Nicolas El Tenn

- Design the overall architecture of the cyber-physical system
- Integrate irrigation system to power and computing modules
- Dataset cleaning and feature learning/engineering
- Training and testing neural network

2.7.4 Raquel Guzman

- Help integrate sensors with jetson nano
- Help with habitation system design and fabrication
- Dataset cleaning and feature engineering
- Training and testing machine learning models
- Testing sensor accuracy and verifying data

2.7.5 Matthew Philpott

- Design and manufacture a power management system with a backup battery
- Design and fabricate lighting solution for plants
- Integrate sensors with Jetson Nano
- Help with habitation system design and fabrication
- Integrate user interface with sensors and microcontroller
- Testing sensor accuracy and verifying data

Everyone was responsible for finding relevant resources related to the development and implementation of the plant habitation system as well as creating designs for the various subsystems included.

2.8 House of Quality

Below is the house of quality that includes analysis of the customer requirements (volume, weight, time, budget, testing) and engineering requirements (frame dimensions, sensor accuracy, sensor operating voltage, data upload time, cost).

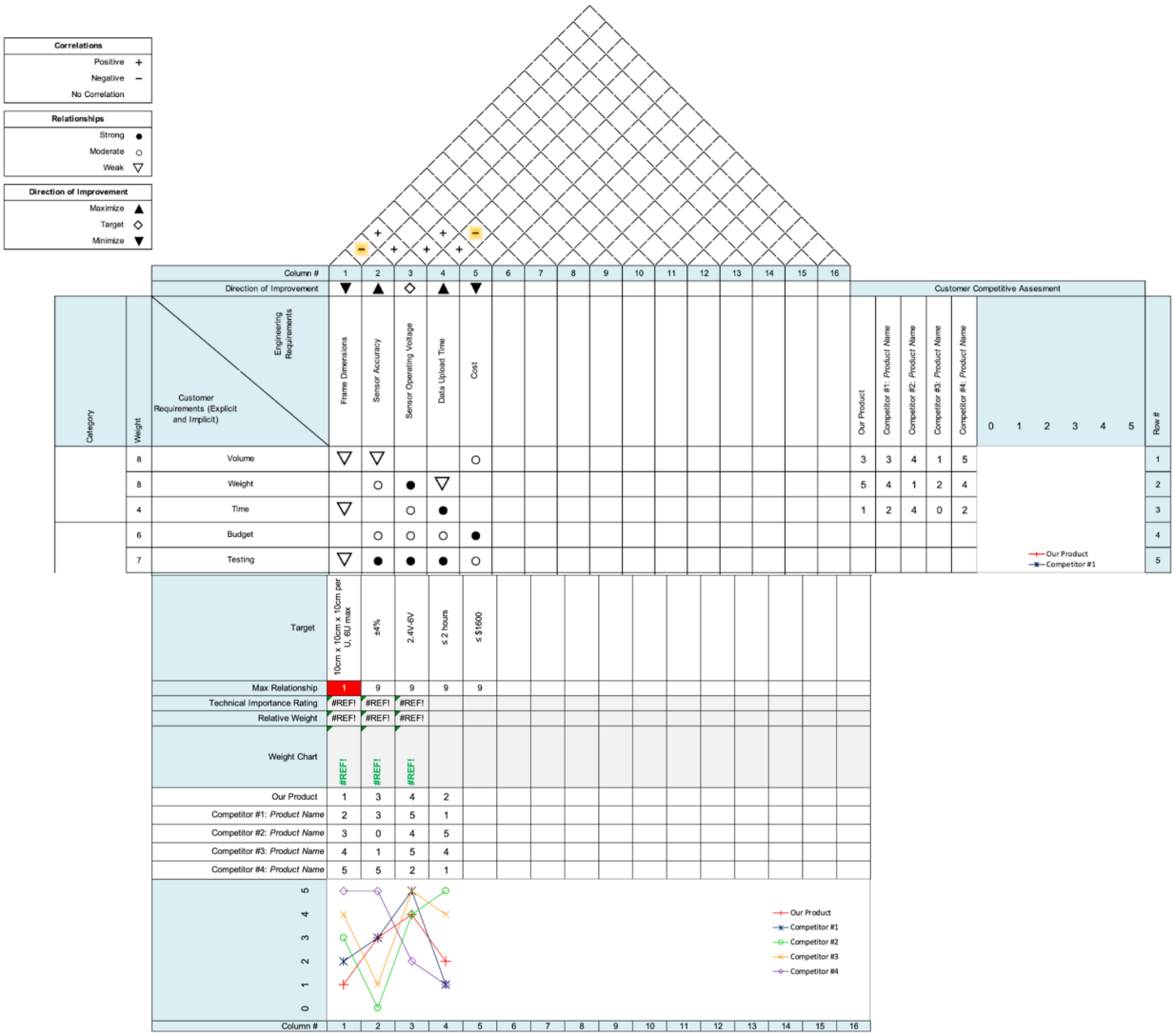


Figure 5: House of Quality

3. Research related to Project Definition

3.1 Existing Similar Projects and Products

This section discusses existing projects and products related to plant habitation systems. It begins with an exploration of previous NASA projects as well as previous university projects. Afterward, products and components related to the design of the system are compared. The computer for the system is also selected.

To maintain accordance with NASA Requirements, the following documentation provided in the Systems Engineering Handbook will be followed to best efforts when approaching similar projects and products [72]:

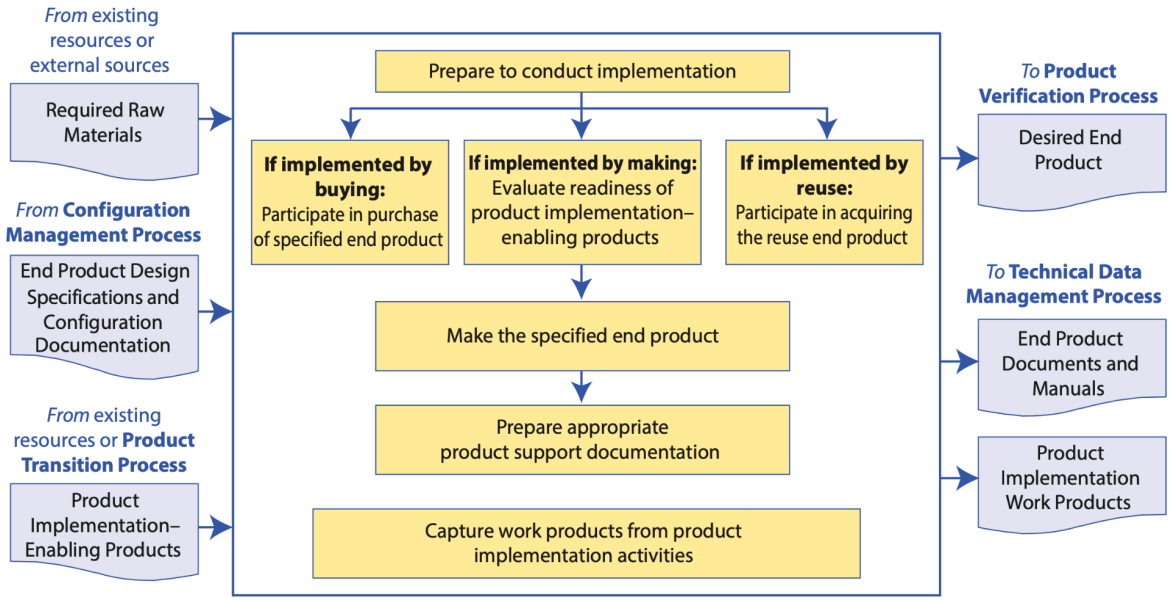


Figure 6: NASA Requirements

3.1.1 PGBA

Plant Generic Bioprocessing Apparatus: A Plant Growth Facility for Space Flight Biotechnology Research, developed by BioServe Space Technologies, University of Colorado at Boulder, and NASA reached the ISS in 1996 [23]. PGBA returned data

on lighting, thermals, humidity, and plant nutrition [23]. It focused on studying the plants' CO2 usage and O2 production, irrigation system, etc.

3.1.2 Growth Monitor Pi

The University of Missouri and Arizona in a recent paper published in *Applications in Plant Sciences*, describe a growth chamber called the Growth Monitor pi. They define the Growth Monitor pi as an open monitoring system for plant science that provides affordability and a wide range of functions. The Growth Monitor pi can sense fluctuations in growth conditions such as changes in temperature, humidity, and light. It is also capable of sending gathered data to the cloud, capturing images of plants, and generating alerts to inform plant scientists on unacceptable plant conditions [18].

In terms of parts, the Growth Monitor pi uses a Raspberry PI Model 3B+ connected to temperature, humidity, light sensors, and a camera module. The Raspberry Pi is installed with a Raspbian distribution, and all scripts were written in Python so they could be run automatically with Unix tools[18, 19]. This project highlights the importance of choosing appropriate sensors and cameras for the plant habitation system. It also highlights how the growth conditions of a plant can be determined using temperature, humidity, light sensor data and images. More importantly, it reveals the need for independence in the system. This would entail the own integrated irrigation system and growth chamber for plants. In general, the system will need to work independently from pre-existing plant monitoring systems, growth chambers, and be self contained.

3.1.3 Veggie

The Veggie, or Vegetable Production System, is a deployable plant growth unit that produces salad-type crops to provide the crew with a safe source of fresh food [40]. It provides nutrients, light, temperature control, and carbon dioxide to help the plant grow and thrive. A portion of the crops are harvested and consumed by the crew members. The Veggie was built by ORBITEC in Madison, Wisconsin and earlier versions were tested at KSC [41]. The seeds are placed into plant pillows and with the combination of light and water, germination begins. Overall, almost 15 different types of plants have been grown from the veggie and more than 100 have been tested on Earth. The system that is being built will be very similar in terms of having the crops harvested and consumed in the future. There will be light, temperature,

and carbon dioxide control to aid the plant in growth. It will provide nutrients and fresh produce for astronauts in space.

3.1.4 APH

NASA and ORBITEC of Madison, Wisconsin developed an Advanced Plant Habitation to be used on the International Space Station. Unlike the Veggie, the APH is a more sophisticated growth chamber. The system is managed by KSC located in Florida. It is made to require minimal crew involvement so that they do not have to worry about how the system is doing. The system uses LED lights and has 180 sensors that all report back to the team at Kennedy [39]. The system is supposed to last up to 135-days for research purposes and at least one year of operation without any maintenance. The design of the system is to send real-time information to the Kennedy team as well as automatically water the plant and detect the flow [39].

NASA's plant habitation system is very similar to the plant habitation that is being built. Although there are not 180 sensors, the system will have water sensors, temperature detection, oxygen level detection, and moisture level detection. The system will automatically water the plant and upload sensor data to the GUI. It will also capture photographs of the plant and upload those as well. This way, users will be able to compare plant data and see which environment the plant thrives in. The GUI can be accessed from anywhere so it requires minimal crew involvement as well.

3.2 Relevant Technologies

3.2.1 Python Framework

One of the reasons why Python was chosen as the programming language is because of the numerous GUI frameworks that are available. There are many Platform-Specific frameworks, Cross-Platform frameworks as well as Cross-Browser frameworks. The three big GUI toolkits that can be used with Python are TkInter, wxPython, and PyQt [2]. All three of these toolkits are Cross-platform frameworks, with PyQt being mobile compatible.

Although TkInter is the most commonly known Python GUI framework and relatively easy to work with, the biggest concern for this framework is that it does not have a modern and shiny interface. Although functional and cross-platform friendly, it was built with a very outdated look to it that people might stay away from [3]. WxPython uses actual widgets on the native platform which makes applications look native to the operating system, but it tends to reduce the value of a really usable UI experience.

Therefore, PyQt5 is the one that was chosen to go for this project. It is a combination of the Qt C++ framework and the language Python. A programmer that uses PyQt has the advantages of both; the power of Qt, and the simplicity of Python. Unlike tkInter, PyQt comes with a vast set of widgets while tkInter has a thin number of widgets. It is able to implement 440 classes and 6,000 settings and methods such as SQL databases, XML parser, and multiple GUI widget packs [4]. Besides being a GUI toolkit, PyQt also features abstractions of network sockets and thread, which is why PyQt is the best framework to use for this project.

3.2.2 Database

A database is an organized collection of data usually controlled by a DBMS and stored in a computer system [10]. Data is usually stored in tables, composed of rows and columns. This makes it easier to sort, edit, update, and organize the data. SQL is a programming language that's used in many databases to query, manipulate, and define data [10]. There are many new databases that have evolved to make it easier to collect, store, manage, and utilize data. SQLite is an embedded SQL database engine that reads and writes directly to ordinary disk files since it does not have a separate server process. A complete SQL database with triggers,

views, multiple tables, and indices is all contained in a single disk file [11]. It was decided to use SQLite over any other database because SQLite is file based making it reliable and portable. The lite in SQLite means lightweight for the setup, database administration, and other resources. Since the project doesn't require high-security features nor are numerous amounts of data being stored, SQLite was the best to use. SQLite database files are used as containers to transfer content between systems. It is built into many mobile phones since it is stable, cross-platform, and backwards compatible. It is the most used database engine in the world. Since the code is open source and free to use, similar to any other projects, there are bugs in the code; however, SQLite is very open and has bug lists for anyone to view. Some common companies that use SQLite in certain applications are: Apple, Google, Adobe, Facebook, and Skype for many different reasons [12].

3.2.3 Arduino Microcontroller

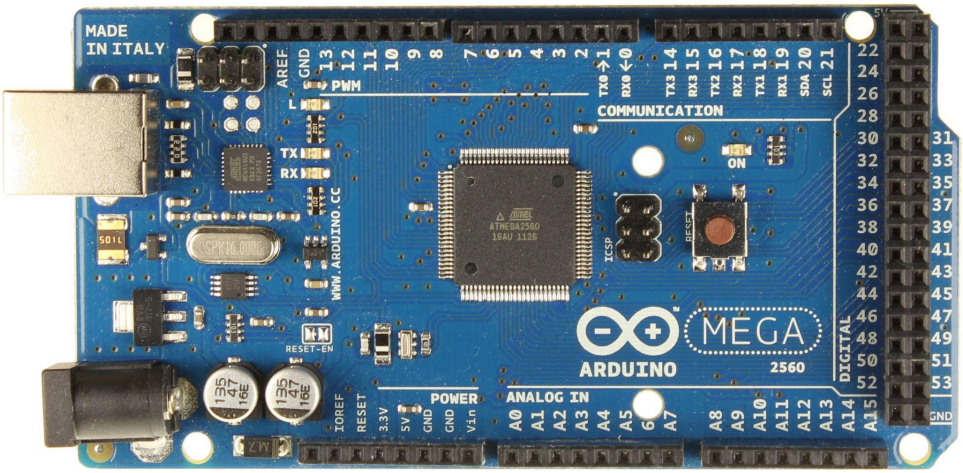


Figure 7: Arduino Mega 2560 Rev3

Python was also chosen because it is native to the microcontroller that is being used. The controller that was decided to use for this project is the Arduino Mega 2560 board, as shown above in Figure 2. An Arduino is a microcontroller board with I/O pins that connect to a computer with a USB cable [5]. It allows easy connection to different components of an experimental system and gains control and monitor abilities. Although the board can be programmed with the Arduino Software (IDE), which is heavy on Java and Python, it was decided to go with Python since it will allow easy extension. Many devices that are usually complex are easier to control with the help of Python since the integration of other programming interfaces

become simpler. One of the frameworks that assist in the simple integration is the Instrumentino framework, which is an open-source GUI framework that is used for controlling Arduino instruments [6]. It allows operation sequences to automatically run without user intervention and it will automatically save usage logs and data on the computer. With the help of Instrumentino, all of the implementation for the GUI is done on the computer, and the programming for the Arduino is replaced with a file in Python.

3.2.4 NVIDIA Jetson Nano

The two computers that seemed to best fit the system were the Jetson Nano and the Raspberry Pi 4. A Raspberry Pi has the processing power to run a GUI, pin inputs for getting sensor data, and integrated storage so an external drive isn't necessary. However, the Jetson Nano is also a powerful computer that has those features plus embedded applications and AI IoT. Therefore, a table was developed to compare the specs between both as shown below and used to help pick which one to use.

	Jetson Nano	**Raspberry Pi 4**
GPU	128-core Maxwell	Broadcom BCM2711
CPU	Quad-Core ARM Cortex A57 64-bit @ 1.43 GHz	Quad-core Cortex-A72 64-bit (ARM v8) @ 1.5 GHz
Memory	4 GB LPDDR4 25.6 GB/s	4 GB LPDDR4-3200
Storage	microSD	microSD
USB	4x USB 3.0, USB 2.0 Micro-B	2x USB 3.0, 2x USB 2.0
Other	40-pin GPIO	40-pin GPIO

Table 2: Nano and Pi Comparison Table

It was decided to go with a Jetson Nano since it'll be incorporating computer vision and the nano has a full blown GPU. Since Nvidia focuses on AI and machine learning, the Jetson Nano was an important part for lots of startups and

enthusiasts. It has the performance and capabilities needed to run modern AI workloads, giving the fastest and easiest way to add advanced AI into products [9]. Since it is high-powered compared to other computers, it is aimed to be used for high end machine learning tasks. Most of Nano runs on full Ubuntu Linux, making it very flexible and having a nice GUI. Since it can run Python code, using it for cameras, sensors, and machine vision is the best approach. Below is a figure displaying the jetson nano kit that will be used for the project.

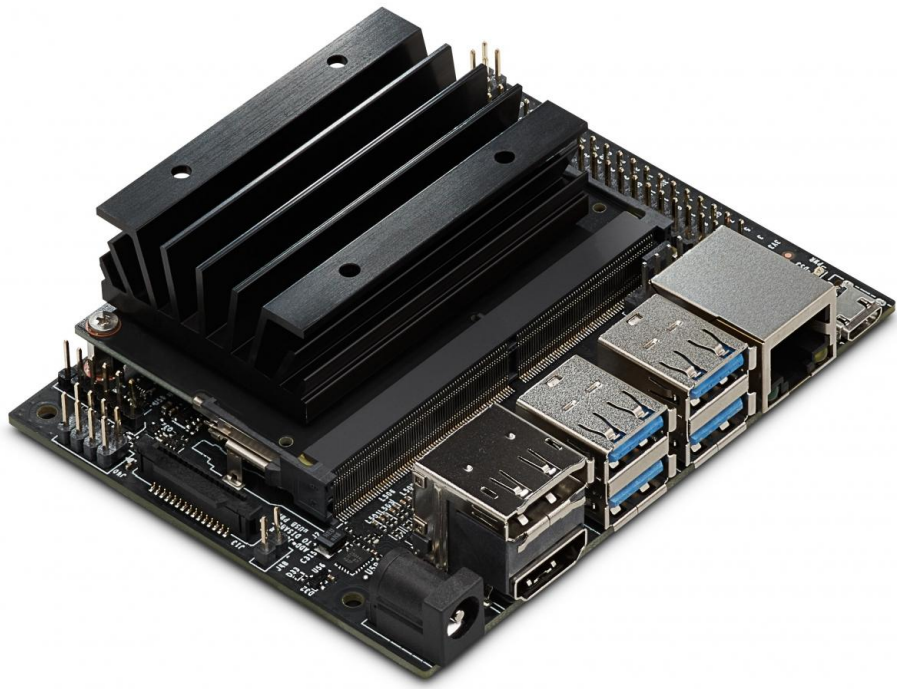


Figure 8: Jetson Nano Developer Kit

3.3 Strategic Components and Part Selections

3.3.1 12V Battery Comparisons

When designing a standalone plant habitation unit it must take into consideration that the system must function under its own power without an external supply. A battery is critical to most space related missions due to the orbit of most satellites wont allow for continuous supply of power. In the case of a satellite in LEO it spends 59 percent of its orbit in direct sunlight, this is approximately 53 minutes in Sunlight and 37 minutes in the Earth's shadow. This is similar to the orbit of the international space station. As well as in the case of any sudden outage from the station's onboard power the plant habitation unit is being configured with a 12AH battery to supply power for maximum sustained load for at least 3 hours. Below is a visible figure of the batter that will be used, as well as a table under it for comparison.

Figure 9: Standard Sealed Lead Acid Battery showing F2 terminals

Item	Description	Size	Price	Weight
Altronix AL400ULXR 12V	Sealed Lead Acid (SLA) battery	7.2 AH	$18.99	4.50 Lbs
Genuine FiOS 12V	Sealed Lead Acid (SLA) Battery	8AH	$44.99	5.72 pounds
SLA12-12F2 Duracell Ultra	Sealed Lead Acid (AGM) battery	12 AH	$49.99	7.7583 lbs

Table 3: Comparison of Batteries

3.3.1.1 Altronix AL400ULXR

A battery example is the 12V 7Ah Battery Replacement for Altronix AL400ULXR. The Altronix is commonly available at many local retailers and drastically cheaper, with a maintenance free design. This is a spill proof battery and allows for high discharge rates over a large operating temperature range. But only comes with a one year warranty. Another concern is the amperage is 7.2 AH and the system design might call for a longer discharge time or capacity.

3.3.1.2 Duracell Ultra AGM SLA Battery

Another example is the Duracell Ultra 12V 12AH AGM SLA Battery with F2 Terminals. The duracell ultra is also a lead acid battery that is sold by several retailers. The battery design is also maintenance free and sealed, and available in multiple terminal types. The battery is also backed by a one year limited warranty. The size of this battery is more advantageous for longer discharge times, but does come with a weight cost.

3.3.2 CO2 sensor

CO2 monitoring was important because of the ability to measure the plants ability to filter the air of carbon dioxide. When dealing with plants the atmospheric content of carbon dioxide is important to track. In sealed plant habitation systems that have already been implemented oxygen scrubbers have been implemented to keep the plants from over saturating their atmosphere to allow for plant growth to continue. This will be necessary in future versions when oxygen is scrubbed at

predetermined levels or allowed to blend to its surrounding atmosphere inside the spacecraft.

Figure 10: Size reference of COZIR-LP-5000 ultra-low power CO2 sensor

3.3.2.1 Gravity Analog CO2 Gas Sensor

The Gravity Analog CO2 Gas Sensor (MG-811 Sensor) was selected as a sensor for determining the CO2 content near the plant. This sensor provides a large range of data collection from 0〜10000 ppm, with a tolerance of ±100ppm@400ppm. This is an analog signal output that provides fast response time and in a slim form factor. This sensor is more optional and native to working with the arduino microcontroller of choice. One drawback to this sensor is the life expectancy is only rated at one year. In future revisions, an average life expectancy of less than a year is not a viable space grade solution. As of now sourcing is available and needs to be ordered before part becomes unavailable again.

3.3.2.2 COZIR-LP-5000

The COZIR-LP-5000 ultra-low power CO2 sensor was a candidate because of its features and availability. These features include a 30ppm typical accuracy. Along with its digital communication method. This sensor has a built-in backup battery option for continuous communication in case of an outage. Meaning the supply voltage can be zero and the microcontroller will still be able to read incoming data. The applications for a sensor of thys type is monitoring of indoor air quality. Because the system might only need to call on the sensor the measurements can be streamed or output on request.

3.3.3 Soil Moisture Sensor Module

In keeping with the goal of a semi-autonomous plant habitation system a monitor to view and maintain the soil moisture content is necessary. This will automatically set a minimum and maximum levels when watering the plants. This method will provide a way to reduce water wasted. This is an important parameter to keep track of because the amount of water available in a space station is limited. And in many cases water is recycled to maximize usage.

Figure 11: Analog Capacitive Moisture sensor

3.3.3.1 KeeYees Sensor

The KeeYees 5 Pcs High Sensitivity Soil Moisture Sensor Modules will monitor the soil moisture content. When dealing with an autonomous watering system there needs to be a way to monitor the soil moisture levels. Plants can be highly sensitive to the amount of water in the soil. This sensor uses a capacitive probe to test moisture content of soil. The signal sent to the microcontroller is intercepted by a control board that needs to be adjusted manually using a potentiometer. The operational voltage is 3.3V to 5V.

3.3.3.2 Adafruit STEMMA Soil Sensor

The Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor was chosen because documentation on some of the cheaper brands of soil moisture or soil humidity sensors were difficult to find. As well as Adafruit being a well known brand, providing extra security that the data that is calibrated from the factory is accurate. But also it is stated that the sensor will need to be recalibrated if the resistivity measurement goes up due to oxidation. This can be done using code and does not require a manual adjustment on the sensor. These readings range from about 200 (very dry) to 2000 (very wet).

3.3.4 Barometric Pressure and Temperature Sensor

Because plants have evolved with a normalized pressure relative to earth's natural fluxuations. This means that it is important to monitor and when a seal system is implemented in the future a method of allowing the pressure to normalize. This is similar to the effect temperature can have on plants. Because the lights and other elements are exposing the plant to heat. When the plant isn't sealed the heat radiated by internal components will bleed off. But in the future when the habitation system is sealed, the temperature will need to be regulated by an internal radiator to release the heat away from the plant.

Item	Description	Price	Temperature range	Accuracy	Pressure Range	Accuracy
MPL115A2	Temperature /Pressure	$7.95	-40°C to +105°C	±1°C	50 to 115 kPa	±1 kPa
BMP180	Temperature /Pressure	$9.95	-40 to +85°C	±1°C	300 to 110 kPa	±1 kPa
BME280	Temperature / Pressure/ Humidity	$14.95	-40 to +85°C	±1°C	300 to 110 kPa	±1 kPa

Table 4: Sensor Comparison Table

3.3.4.1 BMP180 Pressure & Sensor

The BMP180 Barometric Pressure & Temperature Sensor chip is a product of Bosch, this is a modern chip rendition for a digital pressure and temperature. The pressure sensor can measure an altitude from 9000m to -500m above sea level. The chip can also measure temperature with its specifications -40°C to 85°C with ±1.0°C accuracy. This sensor comes pre-calibrated and requires no time for set up.

3.3.4.2 MPL115A2 Pressure & Sensor

A MPL115A2 - I2C Pressure/Temperature Sensor is used to keep track of the temperature and air pressure.This sensor comes precalibrated from the factory

and has no set up time. The operating range of this sensor is a slightly larger temperature range of -40°C to 105°C. This along with the increased accuracy and cost is the reason for selecting this sensor over the BMP180. There is also a documented use with a raspberry pi as well as jetson nano, making integration easier.

3.3.5 Camera Module

A camera will allow for active and remote monitoring of plants' health visually. Because the cameras can be fixed in position they can document a plant's growth over time. As well as monitor plants, cameras can be used to accurately count the number or leaves on the growing plant. Along with counting plant's leaves, cameras and the machine learning model can be trained to help quantify fruits and other consumables.

This method could be used in the future when a fully autonomous system is implemented and needed when determining optimum size and age for harvesting. Two camera modules will be used to monitor both plants that are in the habitation system. This is the method that was chosen because the data set found was readily available including top down images to train the model for counting leaves. The table below displays the three cameras that seemed the best fit for the project. It includes the IMX219-160 Camera, the Arducam Raspberry Pi Camera v2, and the Arducam Raspberry Pi Camera v1.

Item	Description	Price	Resolution	Minimum Focal Distance
IMX219-160 Camera	Camera	$22.99	3280 × 2464	3.15mm
Arducam Raspberry Pi Official Camera V2	Camera	$47.99	3280 x 2464	3mm
Arducam for Raspberry Pi Camera v1	Camera	$15.10	2592 x 1944	1 m

Table 5: Camera Comparison Table

3.3.5.1 IMX219-160 Camera

The IMX219-160 Camera, 160° FOV (Applicable for Jetson Nano) has a wide angle camera that allows for close focal distances, this is going to be necessary when dealing with the distance the camera is located in reference to the plants. The wider angle will allow the capture of all of the plant through its growth. The features do not include an auto focus script, so it will have to be manually adjusted if needed.

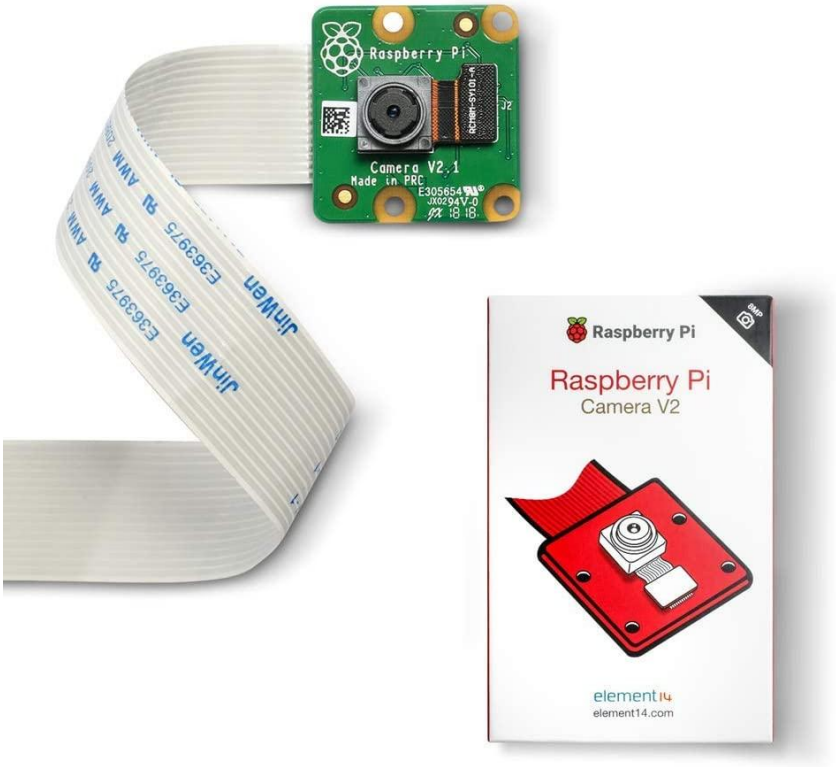


Figure 12: Raspberry Pi Official Camera V2

3.3.5.2 Arducam Raspberry Pi Camera

The Arducam Raspberry Pi Official Camera Module V2, shown above in Figure 6, was selected to allow monitoring as well as an autofocus feature native to enhance image quality. This camera is an open source code to implement software auto-focus function for Jetson Nano. Accessory lenses, similar to a fisheye lens can be implemented to capture a wider FOV. Because this camera is going to be used in a confined space the minimum focus distance of 3 mm is important to note. The

ribbon cable that comes with the V2 is 15 cm cable. If a longer cable is needed to reach from the Jetson mount location to the top of the plant, extension cables can be purchased separately. The camera will be set up and connected to the power module outside of the plant environment.

3.3.6 Potential Hydrogen(pH) sensor

Figure 13: Analog pH Sensor

3.3.6.1 Analog pH Sensor

The Gravity Analog pH Sensor/Meter Kit V2 is made by DFRobot and is designed almost identically to the other manufactures. Compared to the other Atlas branded pH kit, this comes with a 2 point celebration method, with the two included solutions. The detection range is approximately 0 to 14 with an accuracy of 0.1 at 25°C. This sensor probe specifications for temperature ranges from 5 to 60°C. But one negative is the probe is not rated for complete submersion in water or soil. And only has a life expectancy of half a year, depending on frequency of use.

3.3.6.2 Atlas Analog pH Kit

The Atlas Scientific Gravity Analog pH Kit is the preferred sensor because it is designed as a probe that will allow for a submersible probe that can measure the pH of the water and in soil that is critical for proper plant growth. The kit comes with a 3 point celebration solution. Portable pH testers have a life expectancy of 4 to 6 months depending on battery life and usage. With an integrated solution with Atlas it will allow for an average The Atlas pH allows for permanent placement into soil, this will also allow for further information on plant health. Keeping the pH in check will allow for the best chances for the plant to grow.

3.3.7 Touchscreen Display

Touch screen displays are important for implementing a Graphic User Interface (GUI) so that if it is needed to look at something on the actual system, any user is able to. With that interface. A pixelated detail is needed to visualize graphs and software compatibility to display information. Information that is planned on being displayed to the usar about plants health can be taken from the sensors and shown in a graphical fashion. As well as information that can be inputted from the user, such as any notes or documentation.

3.3.7.1 Waveshare Resolution Monitor

Figure 14: Waveshare 7inch HDMI LCD (H) (with case) 1024x600 Resolution Monitor

This display uses a capacitive touch screen that is capable of 5 point touching. The touch screen uses a USB to communicate the touch commands to the microcontroller. There are extra controls for brightness and contrast built into the OSD menu. The display resolution is as stated above,1024x600.

3.3.7.2 Adafruit Display Backpack

Given that a GUI is needed to be implemented, the Adafruit HDMI 7" 800x480 Display Backpack - With Touchscreen was selected since it is small and simple, making it perfect to fit onto the system. This display is powered through USB or with the default HDMI backlit current. It can take unencrypted videos and output raw 24-bit color pixel data [42]. It has been tested successfully on Mac, Windows, and Linux computers, therefore, it is capable of being used by any user.

3.3.8 Peristaltic Pump

There are a few options for choosing an irrigation pumping method. A gravity feed system can be used that opens and closes a valve allowing water to flow, or a water pumping method. The simplest solution would be to have a gravity feed system, but in microgravity there is little to no precision to keeping water flossing in one direction. This means that a pump must be used. The various pumps include a cyntrical pump and a Peristaltic pump. This will pump water whenever it is time to into the soil where the plants will be.

3.3.8.1 Gikfun DC Mini Water Pump

One of the options is the Gikfun DC 3V 5V Micro Submersible Mini Water Pump. This is a low current drawing pump that works with the common voltages used by arduino and various sensor components. The system does not come with a life expectancy or recommended for continuous use. The main issue would be keeping the pump submerged in a micro gravity situation. This would mean that the pump could fail unexpectedly as well as have to be reprimed to function properly.

3.3.8.2 Gikfun Dosing Head

Another is the Gikfun 12V DC Dosing Pump Peristaltic Dosing Head, which allows use to draw both air and water with its pumping head design. This pump does draw more current then each of the smaller submersible pumps, but will function better allowing for better liquid dispensing then the submersible pump.

3.3.9 Lighting solutions for plants

As with many greenhouses and indoor plants, supplemental lighting may be necessary to keep the plant healthy. Because the system is intended for space and will not have access to a window or external light source besides that available to the crew and operators. These external lights are not designed for the wavelengths that are beneficial for plants as well as in a location that provides proper coverage.

3.3.9.1 CALIDAKA LED Plant Grow Light Strips

The CALIDAKA LED Plant Grow Light Strips provide both heat and light to the plants promoting photosynthesis in the plants. The features of this light strip include the flexible and IP66 waterproof design. This means that in the limited space capacity of the habitation system the light strip can be molded or bent to the need by which the final product will be needed. The waterproof design is necessary to prevent corosine and help with longevity of the component in a humid environment.

3.3.9.1 Homevenus 4-Heads Full Spectrum Clamp LED Grow Lights

The Homevenus 4-Heads Full Spectrum Clamp LED Grow Lights is a fixed length grow light that helps accelerate plant growth as well as blossom and fruiting of plants. This light was designed with the red and blue spectrum of wavelengths that are essential for plants. This product has a waterproof design and adjustable light settings. This can allow for efficient scheduling of the plant's light needs throughout the day. This product does not come in a strip and flexible form factor and sizing will have to be taken into consideration. The warranty and 50000-hours life time and offering low energy usage is also a benefit.

3.3.10 Final Sensor list

This experiment gives a flexibility to choose between various manufactures and specifications. Because for some of the sensors the major requirements for each part selected is concerned with reliability and relative accuracy. These parts are useful in tracking the plant's heal, but the overall goal of the experiment is to determine plant yield. This yield is critical, and the more critical parts include the camera's ability to communicate and provide detailed image data for the image

processing. Therefore the list below provides a final sensor and parts list for electronic and sensing data.

Item	Amount Needed	Cost	Total
Jetson Nano	1	$129	$129
Arduino Mega 2560	1	$45	$45
Gravity Analog CO2 Gas Sensor (MG-811 Sensor)	4	$50	$200
Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor	4	$7.50	$30
MPL115A2 - I2C Pressure/Temperature Sensor	4	$8	$32
Arducam Raspberry Pi Official Camera Module V2	2	$55	$110
Atlas Scientific Gravity Analog pH Kit	4	$66.50	$266
Homevenus 4-Heads Full Spectrum Clamp LED Grow Lights	1	$29.99	$30
Adafruit HDMI 7" 800x480 Display Backpack - With Touchscreen	1	$90	$90
Gikfun 12V DC Dosing Pump Peristaltic Dosing Head	1	$30	$30
Total			$962

Table 6: Final Parts List and Cost

3.4 Possible Architectures and Related Diagrams

3.4.1 PGBA Design

PGBA is composed of mainly 2 parts, a shell body and a plant growth chamber that is attached to the front panel along with the LCD touch screen and switches. This design allows for the containment structure to be installed months in advance for flight. The plant growth chamber along with the front panel can then be deployed a few days before launch [23]. Figure 3 displays an overview of PGBA that outlines the general composition of the system. This is very helpful for explaining what each component will be.

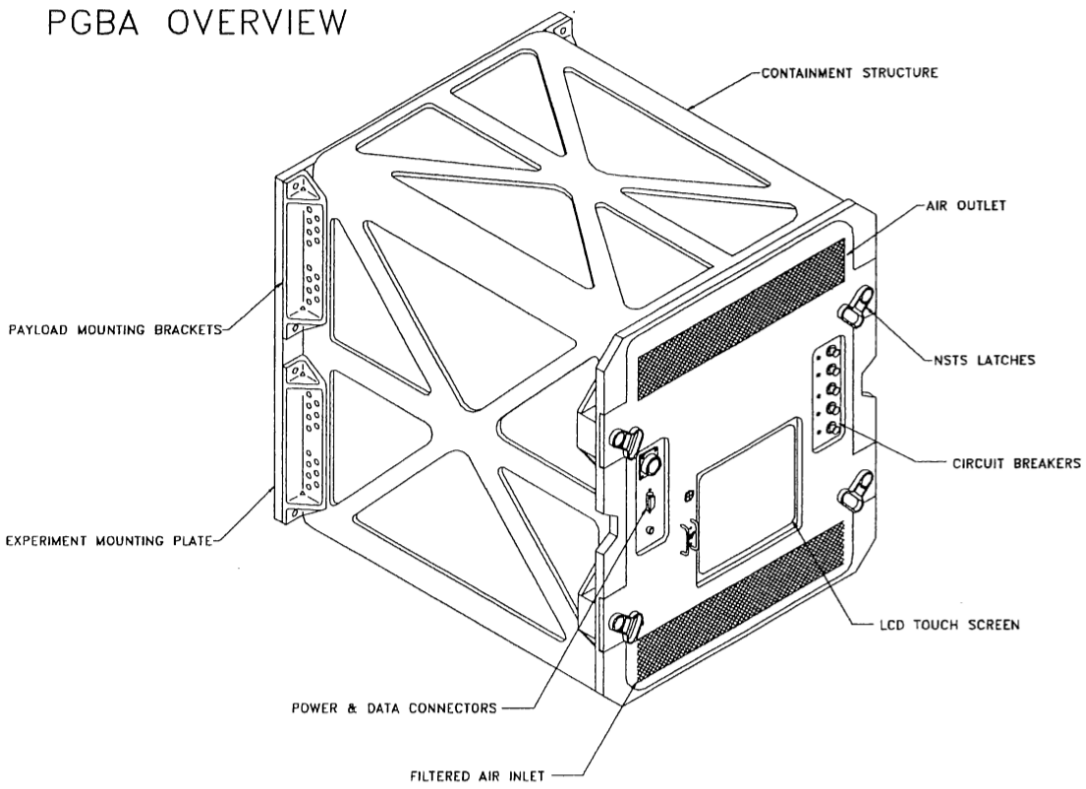


Figure 15: An overview of PGBA

3.4.3 Growth Monitor Pi Design

Environmentally controlled facilities are essential for experimental research. Not only does it help with maintenance but it also is beneficial to improve reproducibility. Therefore, a Growth Monitor pi is developed to monitor growth chamber conditions. The Growth Monitor Pi consists of three main parts: a camera, sensors, and a Raspberry Pi [10]. Below is a diagram showing the connections between the Raspberry Pi and its peripherals. As it is depicted, the pins will be available for any output or input for the system. The camera will be placed at the bottom for photographs. The pins will be connected to temperature, humidity, and light sensors.

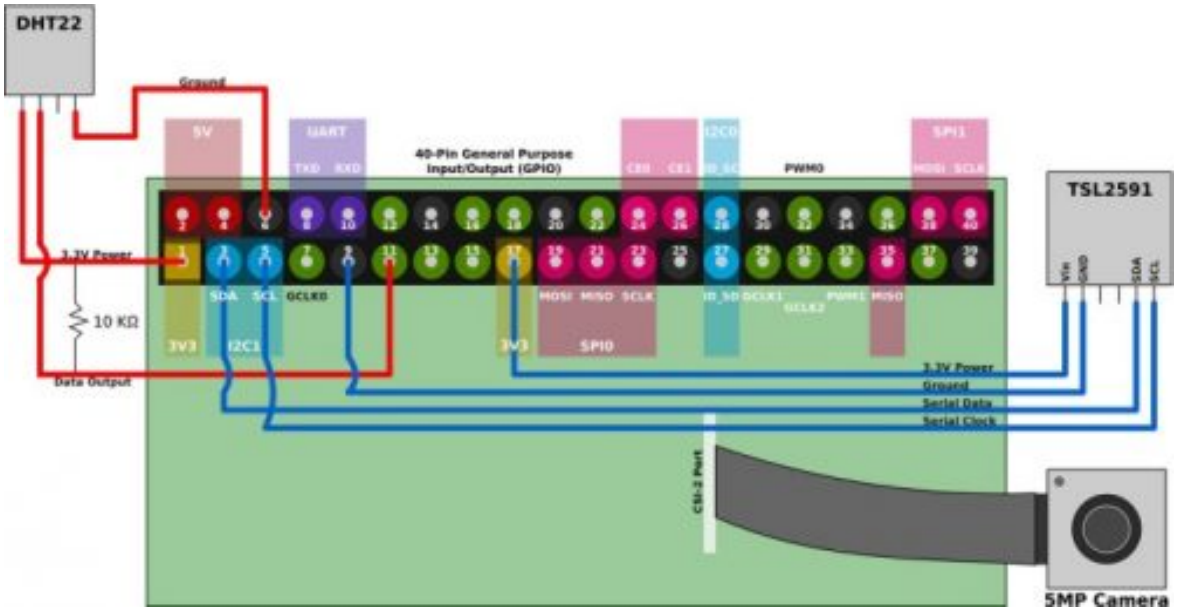


Figure 16: Growth Monitor Pi Diagram

3.4.3 Veggie Design

The design features a dynamic system with a variable height and mechanical arms to accommodate. However further schematics, parts, and other specifications were challenging to find. [40]

As a result, the following has been assumed about the system:
The system is to have a passive irrigation system and the lighting is to be connected to the top layer assembly cap. In addition, the water reservoir being on the bottom ensures the splitting of organic matter and electronic matter from the top and bottom assembly units. This design allows for ease of access to electrical components for maintenance as well as ease of access to the plants without connection to any electronics.

The below diagram depicts the design of NASA Veggie on a high level:

Figure 17: NASA Veggie

This design has been successfully tested on the International Space Station. The following iteration on the design was the Advanced Plant Habitat or APH. [40]

3.4.4 APH Design

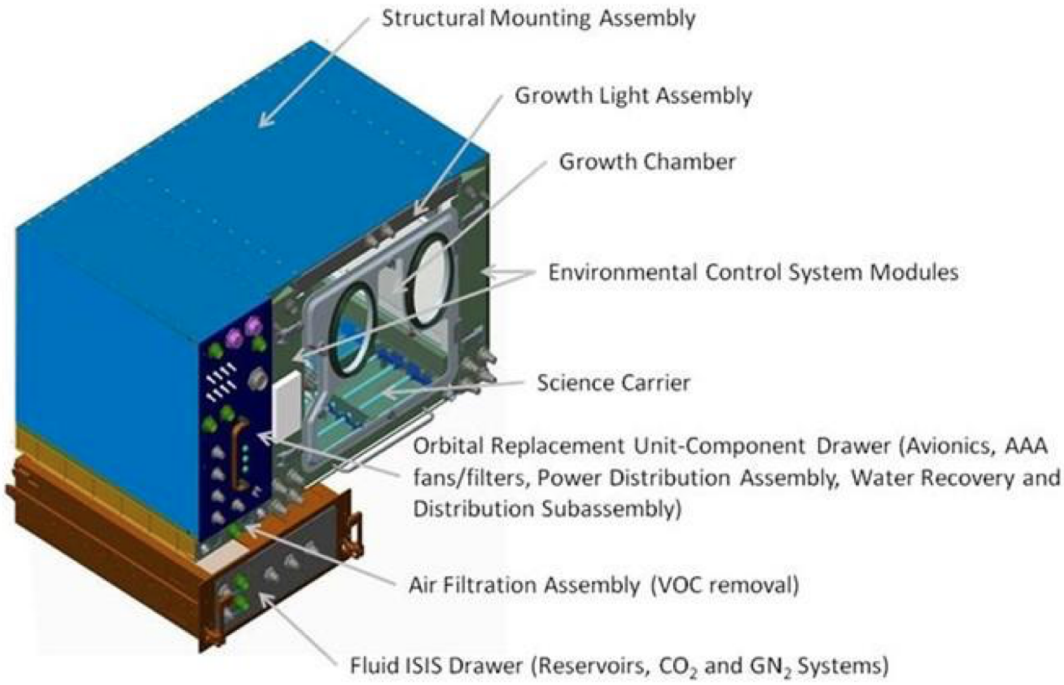


Figure 18: APH Design Diagram

NASA Advanced Plant Habitat was designed in conjunction with Orbital Technologies Corporation to be the largest plant growth chamber on the ISS. APH iterates on all of the areas covered by PGBA in addition to some extra ones. It optimizes functionality while abiding by the constraints: weight, space, energy, crew time. It also has all the necessary components for plant growth while ensuring high yield. APH also features modular subsystems available for replacement on the ISS.

3.5 Parts Selection Summary

All Parts reliability and compatibility were taken into account since the project requires two microcontrollers. First, the Arduino Mega for sensor data and the sensor libraries. Second, the Jetson Nano runs a version of linux to display information to the GIU and handle camera data. Given the plant habitation system is being designed with a mindset that it could go to space. And with that part failure is less than optimal when the nearest amazon distribution center is a multimillion dollar rocket launch away. This is why several parts and sensors have redundancy. Space is an unforgiving environment and sensitivity to bit errors is another consideration, the levels of radiation are all things that would normally be taken into account. Because the payload and sensor data is not mission critical, meaning that if a failure occurs the system will still function as normal after. The redundancy and various data transfer techniques can allow the GUI to do comparisons of data and can also help with eliminating these types of errors. Because the system has to be self-contained and the design for a plant habitation system is intended for microgravity, more planning must be taken to ensure reliability. Parts should work in a microgravity scenario and not cause unintended errors. Because space grade hardware is usually a niche market, prices for a single part and building materials would put this project over budget. This means that commercial and educational equipment will be used as a substitute for practical purposes.

Along with that redundancy is key in keeping the system working for its entire life cycle. The data gathered will be from sensors that monitor the internal conditions of the plant habitation system as well as sensors that monitor the plant's soil conditions. This sensor data includes humidity, temperature, pH, CO2 levels, pressure, soil moisture levels, and light levels. All this data will be stored and integrated into a Graphical user interface, that way operators can infer about a plants health overtime. Cameras will also be implemented to provide autonomous monitoring and quantitative analysis on plant leaf production. This type of analysis can be further expanded on to provide the user with accurate crop yields and other analytics in the future.

4. Related Standards and Realistic Design Constraints

4.1 Standards

The following subsections discuss standards that the plant habitation system must abide by and their impact on the design of the system.

4.1.1 IP Ratings

The International Electrotechnical Commission (IEC) is a global non-profit organization responsible for developing standards related to quality infrastructure and trade in electronic goods. They are also responsible for developing ingress protection (IP) ratings, which define the protection measures a system has against solids and liquids [13]. Each rating is identified using the prefix IP and two unique numbers. The first number represents its protection against solids and the second represents its protection against liquids. A system rated as IP67 would be completely dustproof and be protected against temporary immersion.

In terms of solids, the plant habitation will be accessed and used by individuals through a touch screen and GUI. However, internal electronics will need to be protected from dirt and dust. Given that the system will have running water and water will be dispensed for each plant, the electronics will need to be protected against water splashing from any angle. Although it is unlikely that the irrigation system would be dispensing water around the clock, it is better for electronics to always be protected even when the irrigation system is not running. Sensors will not need to be submerged either. Based on these factors, the system should be rated at IP54. The table shown below is a summarized version of the IEC's IP ratings [13].

IP rating	Protection against solid objects	Protection against liquids
0	None	None
1	Greater than 50 mm	Vertically falling water drops
2	Greater than 12.5 mm	Water dripping at 15°
3	Greater than 2.5 mm	Spraying water
4	Greater than 1.0 mm	Splashing water
5	Dust protected	Normal water jets
6	Dust-tight	Powerful water jets
7	N/A	Temporary immersion in water
8	N/A	Continuous immersion in water
9	N/A	High pressure and temperature water jets

Table 7: IP Rating

4.1.2 Coding Standards

4.1.2.1 Readability

Readability is important in code since it would be easier on the eyes to see what's happening. In the example below, it can be a little difficult to quickly make out what is happening in the code and one might get confused or they might have to spend extra time trying to keep up with the math that is happening. Their eyes have to work harder to figure out what operation is being done.

```
y=2
i=i-2/8-y
ex=(x+2+y)-i+4-y
b=(i+2)*(y+2)
```

Figure 19: Incorrect example of Readability

If spaces were added to the code and readability was prioritized, the code would be neater and easier for someone to read. Simple and easy to read code not only helps the coder when they need to look back on something but it also helps the reader keep up with what is being written.

```
y = 2
i = i - 2/8 - y
ex = (x + 2 + y) - i + 4 - y
b = (i + 2) * (y + 2)
```

Figure 20: Correct example of Readability

4.1.2.2 Continuation Character

Another example of easy readability is spreading the line of code into a couple lines instead of trying to fit everything on one line. In Python, the backslash (\) is a continuation character and when it is placed at the end of a line, it interprets that

the next line is a continuation. This is really helpful when someone is opening, reading, and writing to new files and their path is very long. The example below shows the correct way to use the line continuation instead of trying to fit everything onto one line.

```
# Wrong:
with open('/here/is/a/path/to/some/file/you/want/to/read') as f1, open('/here/is/a/path/to/some/file/b

# Correct:
with open('/here/is/a/path/to/some/file/you/want/to/read') as f1, /
open('/here/is/a/path/to/some/file/being/written', 'w') as f2:
f2.write(f1.read())
```

Figure 21: Example of line continuation

In the wrong example, the line is so long that it goes off the page, making it difficult for readers to see what is happening to the second path. It is unpleasant for readers and other coders that are trying to read the code and see what is happening. The correct example uses the line continuation to fix the line and split it into a couple lines. This is easier to read and help understand what the code is doing.

4.1.2.3 Comments

In Python, comments are indicated by a pound (#) sign. Making sure the comments for a function are clear and precise is very important. It should be a priority to keep the comments up-to-date whenever the code changes. In the example below, it is easy to distinguish what the function is doing even without comments.

```
def division(first, second):
    value = first/second
    return value
```

Figure 22: Proper example without comments

However, for an example such as the one below, it is difficult for someone to understand exactly what is happening. If this was a project that is being developed collaboratively, then someone might get confused and interpret it differently.

```
graph = bonobo.Graph(
    bonob.No('file.csv'),
    bonobo.Yes(
        lambda **that: that['point'] != 'CEO'
    ),
    this,
    bonobo.No('thank you'),
)
```

Figure 23: Incorrect example without comments

Therefore, there should be comments included for functions that explains what is happening so others can stay up to date with what is going on. Instead of having to read through each line of code to figure out what is happening, they are able to read the comments quickly.

```
# import libraries
import requests

# initiate dictionary
data = {}

# getting the response from the API endpoint
response = requests.get("http://api")

# return the information to our data dictionary
data = response.json()
```

Figure 24: Correct example with comments

When comments are included, they are only benefiting the reader and the writer instead of hurting them. Even for a simple project like Figure 18, it is easy to read and understand what is happening because of the comments.

4.1.2.4 Exceptions

Not only do exceptions separate code making it easier to read, review, and correct errors but it also speeds up runtime as well. Maintaining code will be painful for anyone if there are no exceptions to catch subtle bugs in the code. If an exception handler at least logs or even prints the error, the user will be aware that an error has occurred and maybe even where that error occurred. Sometimes exceptions are necessary when cleaning up code. Below is a simple example of catching an exception.

```
try:
    num = collection[key]
except KeyError:
    return no_key(key)
else:
    return value(num)
```

Figure 25: Example of exceptions

4.1.3 USB Standards

The USB Implementers Forum (USB-IF) is a non-profit organization created by the group of companies that developed the universal serial bus specification (USB). The seven founding companies included: Compaq, Digital, IBM, Intel, Microsoft, NEC, and Nortel [14]. It is currently maintained by the following companies: Apple, HP, Intel, Microsoft, Renesas Electronics, STMicroelectronics, and Texas Instruments [15]. The USB is an industry standard that defines the cables, connectors, and protocols for connections between computers, peripherals, and other computers [16]. In total, there have been four generations of USB.

4.1.3.1 USB 1.x

The original standard, 1.0 USB, was created in January 1996. It specified two signaling rates: low speed (1.5 Mbit/s) and full speed(12 Mbit/s). It had many limitations, and few of these USB devices made it to the market. In August 1998, the 1.1 revision was released and widely adopted by the industry. Both 1.0 and 1.1 only specified designs for standard type A or B [16].

4.1.3.2 USB 2.0

In April of 2000, the second generation was released. USB 2.0 added a higher maximum signaling rate of 480 Mbit/s and introduced different size connectors: Micro-A. Micro-B, Mini-A, Mini-B [16].

4.1.3.3 USB 3.x

USB 3.0 was released in November of 2008. It introduced a SuperSpeed mode, which provided an increased maximum signaling rate of 5.0 Gbit/s. The standard also defined that USB 3.0 be backward compatible with USB 2.0. USB 3.1 was released in July 2013 and has two versions. The first kept USB 3.0's SuperSpeed node and is called USB 3.1 Gen 1. The second version defines a new SuperSpeed+ mode and is called USB 3.1 Gen 2 and doubles the maximum signaling rate to 10.0 Gbit/s [16, 17]. In 2017, the specification for USB 3.2 was released. It introduced dual-lane capabilities to SuperSpeed+ and allowed for maximum signaling rates of 20 Gbit/s [17].

4.1.3.4 USB 4.0

USB 4.0 is the most recent specification and was released in August 2019. It supports a higher signaling rate of 40 Gbit/s. This iteration is compatible with Thunderbolt 3, and backward compatible with second and third-generation USB connectors [17]. Below is a table that displays the different USB Types and their specifications.

Standard	Year Introduced	Connector Changes	Maximum Data Transfer Speed	Connection Length
USB 1.0	1996	USB-A USB-B	12 Mbps	3 m
USB 1.1	1998	USB-A USB-B	12 Mbps	3 m
USB 2.0	2000	USB Micro A USB Micro B USB Mini A USB Mini B	480 Mbps	5 m
USB 3.0	2008	USB Type C	5 Gbps	3 m
USB 3.1 Gen 1 & 2	2013	USB Type C	10 Gbps	3 m
USB 3.2	2017	USB Type C	20 Gbps	3 m
USB4	2019	USB Type C	40 Gbps	0.8 m

Table 8: USB Types Specifications

4.1.3.5 Implications on Design

Taking all standards into consideration, the GUI's internal electronics are to be fully protected against any dust, dirt, or water splash. All other electronics, such as sensors, cameras, and power modules, will be made sure to stay protected. For the coding standards, it will be made sure that all code has proper commenting for the computer scientists now and other collaborators in the future. Readability will be a priority since each line of code might include a lot of links to photographs as well as database information. For USB functions, since the Arduino takes a USB 2.0, that will be the one that is used for the system as well. Although it might not be as fast as some of the other options, it will be the best fit for the system.

4.1.4 Sensor Standards and Accuracy Metrics

The implementation of sensors has multiple benefits. In the case of a plant habitation system, they allow for predictive and preventative maintenance of a plant. Monitoring if a plant is getting too much water through moisture sensors. Monitoring if a plant is ill if the soil is too acidic through a pH sensor. Each sensor enhances the data collection process by taking accurate, reliable, and continuous measurements. These sensors must also adhere to industry and organizational standards to ensure quality data collection.

4.1.4.1 Temperature Sensor

Temperature sensors are one of the most common in the industry and are used in a wide variety of applications. Surprisingly, there are no organizations that standardize the temperature range in sensors, so temperature ranges will depend on the item the user is sensing. Generally, most commercial products use temperature probes with a temperature range of -20 to 85°F. This allows the sensor to safely operate without damaging its components. Different temperature sensors offer different advantages as well. Most commercial products implement temperature probes but environmental monitoring systems and aerospace applications might use NTC Thermistors since they can handle higher temperatures and a wider range [24]. A simple temperature probe is to be used for this project. Temperatures will likely average around 62.5°F, since most flowering plants grow between 55°F and 70°F.

4.1.4.2 Soil Moisture Sensor

The plant habitation system will need to conserve as much water as possible while simultaneously providing sufficient water to the plants. To accomplish this goal, the team can implement soil moisture sensors to monitor the irrigation. In industry, there are two main types of soil moisture sensors. These include sensors that measure volumetric water content or sensors that measure soil tension. Each offers its own advantages and disadvantages [25]. Volumetric water content sensors have quick measurement times, with very accurate readings. They are generally expensive with most costing over $200 per sensor.

On the other hand, soil water tension sensors have good accuracy in most soil types. They are inexpensive at less than $50 for the more basic models but have

slower response times. Both types require calibration before information can be recorded, but some can come pre calibrated. Depending on the quality of the sensor, it may need to be recalibrated manually or through code. Due to the constraints imposed by NASA's funding, the project would be unable to work with volumetric water content sensors as this would exceed the team's budget of $1600 so it would need to work with a basic soil water tension sensor to minimize costs.

4.1.4.3 pH Sensor

The pH sensor will be used in tandem with the soil moisture sensors to monitor the health of the plant and decide proper irrigation levels. pH measurements can be used to gauge the acidity or alkalinity of a plant's soil. The National Institute of Standards and Technology (NIST) specifies a fourteen point pH scale which has become the industry standard for measuring an item's acidity and alkalinity. Measurements below seven are considered acidic, while measurements above seven are considered basic [28]. The NIST does not define types of pH sensors but there are four main types used in industry: combination pH sensors, differential pH sensors, laboratory pH electrodes, and process pH probes. The most common type is a combination pH sensor, which finds the pH using two electrodes. The other types of sensors are typically used in laboratories or industrial applications [26]. For the purposes of this system, the group will use a combination sensor to measure the overall health of the plant.

4.1.4.4 CO2 Sensor

CO2 sensors are commonly used to detect the amount of CO2 in an environment. The most common sensors abide by the nondispersive infrared (NDIR) standard. NDIR sensors work through an infrared source, a sample chamber, light filter, and infrared detector. CO2 gasses are then diffused into the light tube and the electronics measure the absorption of the characteristic of light [31]. Typical sensors can greatly vary in price, so the design team picked a simple analog CO2 sensor to measure the plants emissions.

4.1.5 Design impact of relevant standards

The main impact of design will be the sensor and probe placement. The main four sensors that will be placed into the plant habitation system are: temperature, soil moisture, pH, and CO2 sensors. These will be displayed at the back of the system

facing the plants in the middle of the system. The probes for the pH and soil moisture will be placed directly in the soil. The connections for each of the sensors will be placed on the appropriate pins to aid in exporting the data to the GUI. The sensors will also be aligned with the power module for power and the soil moisture sensor will connect to the irrigation system to allow a certain amount of water flow at a certain time. All of the sensors have to be connected to the correct ports in order to make sure they are picking up the correct data to be transferred. All connections will need to be protected to ensure the sensors are reading accurately.

4.2 Realistic Design Constraints

4.2.1 Economic and Time constraints

The plant habitation system will need to abide by NASA's budget of $1600, and be completed by April 2022. Although the cost of parts will be covered under this budget, if any component breaks or malfunctions the group will be required to cover the costs. The team must submit updates and reports throughout the year in the form of design reviews to ensure that deadlines are being met.

4.2.2 Environmental, Social, and Political constraints

4.2.2.1 Political/Legal Issues

Laws from the International Space Law as well as the Outer Space Treaty that could relate to the project. This list is not exhaustive.

- Article VI of the Outer Space Treaty states that space activities of nongovernmental entities shall require authorization and continuing supervision by the appropriate state [7].
- Article II of the Registration Convention states that each state that launches or procures the launching of a space object must establish a national registry of objects launched into outer space, inform the UN Secretary-General of its establishment, and include in such registry its space objects [7].
- Article IX of the Outer Space Treaty states that any experiment that might cause potentially harmful interference shall undertake appropriate international consultations before proceeding [7].

This project does not interfere with these articles since the project will meet all the requirements mandatory for the Outer Space Treaty, the Registration Convention, and the International Telecommunication Union. There will be regular supervision of the plant habitation system in space and to ensure that it is meeting all the requirements listed for the National Registry of Object. The system will also be included in the registry of space objects to ensure that there is proper implementation for planetary protection as listed below in the Ethical Issues Section.

4.2.2.2 Environmental Issues

An environmental issue that the plant habitation system faces is when and if the system plans to be returned to Earth, there could be chances of it bringing back dangerous samples of outer space into the environment. This could lead to major changes in the environment. If the system happens to come across a chemical that is not found on Earth, bringing it back to Earth could cause chemical reactions that the Earth is not used to. This could impact the future of Earth. However, this can be prevented by incorporating planetary protection to make sure the system is completely protected and contamination is limited or completely destroyed.

4.2.2.3 Social Issues

A social issue that can arise from this plant habitation system is automated farming. Since the plant will be growing in space from simply a computer feeding it light and water, it eases the labor and time-intensive process of agriculture. Eventually, this will improve the quantity and quality of the agriculture produce. It will also increase the food production level. However, this can impact farmers that are working hard to make money and provide for their families. They will no longer have a job and will have to look at other ways to financially support their families. This can create a lot of issues for people that would rather not allow automated farming to become normal.

4.2.3 Ethical, Health, and Safety constraints

4.2.3.1 Privacy Issues

As of right now, there are no privacy issues associated with this project. No collection of anyone's information (name, address, phone number, etc) is being done in conjunction with the lack of spying into anyone's personal lives. There is no intention of leveraging anyone's personal information for monetary gain or sharing it with third parties; therefore, there seems to be no privacy issues with this project.

4.2.3.2 Ethical Issues

Article IX of the Outer Space Treaty brings up an ethical issue. The issue brings awareness to the risk of contamination by alien forces that may be destructive. Since this project aims to design a man-made habitation system that may carry

Earth's microbes to outer space, the possibility of it being harmful to an existing habitable ecosystem does exist. At the same time, if a habitation system is planning to be returned to Earth, it may bring dangerous samples into the terrestrial habitat. However, the incorporation of planetary protection from the earliest stages of development can ensure proper implementation. Furthermore, this can prevent harmful contamination before the habitation systems leave Earth and also when they are returning back to Earth.

4.2.3.3 Health Issues

A health issue can be similar to the environment issue. If the habitation system brings back a chemical or a sample that can cause contamination on Earth, it could potentially impact the health of human beings on planet Earth. This chemical could affect how human's body evolves or even eliminate all human life on Earth. It can also affect plants that grow on Earth as well as food for humans. It can create dangerous effects that might not be visible until decades later. This can be prevented by quarantining the system until everything is clear.

4.2.3.4 Safety Issues

The system and all its components should pose no risks to its users or cause any bodily harm. All electrical connections will be enclosed to not add any risks to the environment.

4.2.4 Manufacturability and Sustainability constraints

Because a complete system from top to bottom is going to be built, some items are manufacturable. This includes the ability to take a component or its raw materials and create a finished product. Because a lot of the components are going to be purchased it is not necessarily a concern. But in terms of creating a frame out of a plexiglass structure, a subtractive method will be used, where it will start with a large piece and be cut to the size necessary. Additionally the class can be fused back together using an epoxy or held with a metal channeled frame. The issues this will bring forward is one of sustainability and this can be solved with maximizing used material and eliminating waste. In some mass production cases the shavings and excess material can be recycled and reused.

5. Hardware and Software and Design Details

5.1 Initial Design Architectures, Related Diagrams

As seen in the figure below, the first sketch of the initial diagram provided enough insight for the team to have a rough vision for the minimum viable product. The mass and volume constraints were taken into consideration as well as the array of sensors required for data collection. In addition an input method was also explored and a layout of the user interface was included.

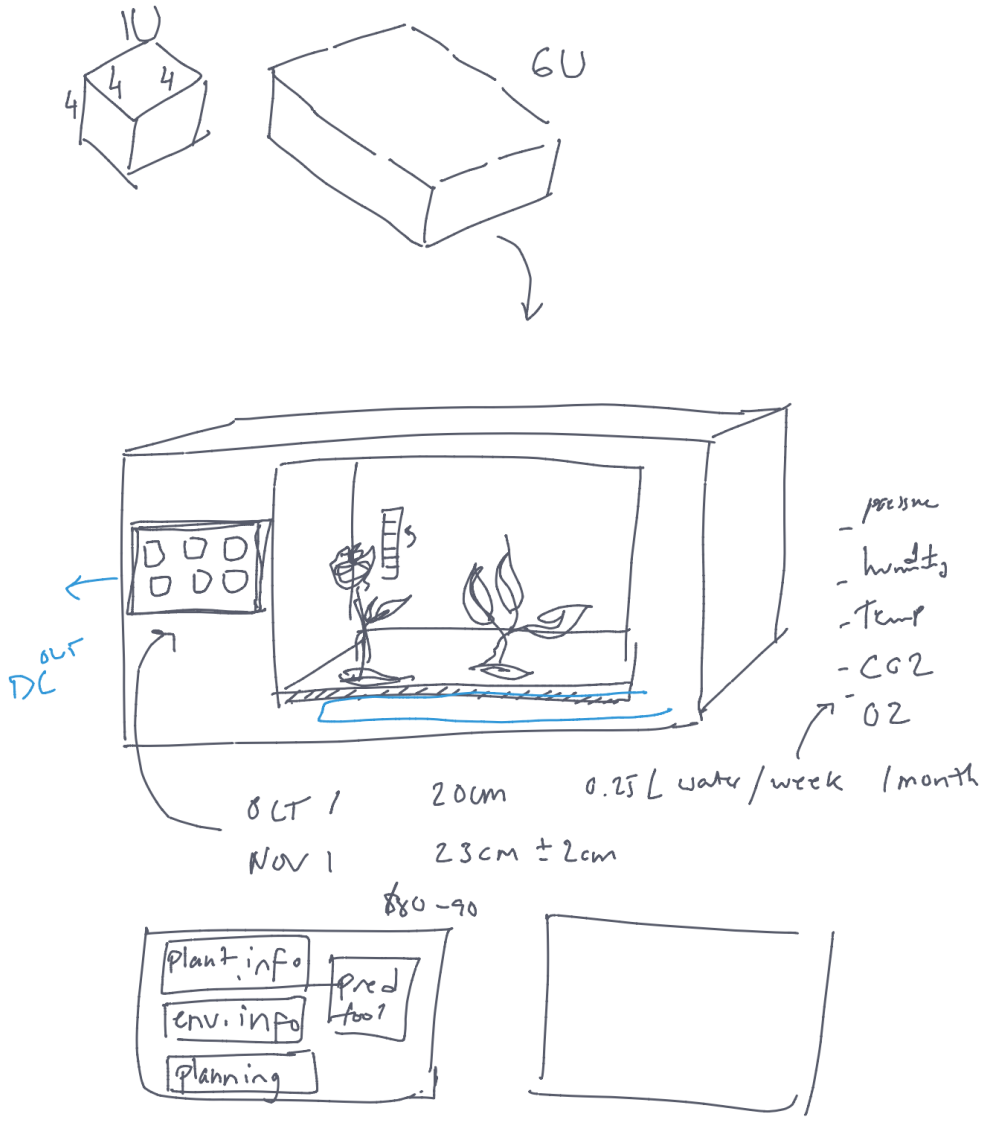


Figure 26: First sketch of the hardware and software design

The first sketch served as a basis to build upon. So, the second sketch was a refined diagram depicting in detail the components that would need to exist in the platform as shown below. It gives a top to bottom view of the window to show where in relation all the sensors will be. The plants will be in the middle connected to the water supply. The water supply will be to the right of the window as well as the power supply. There will be two cameras in the system to show the growth of the plants.

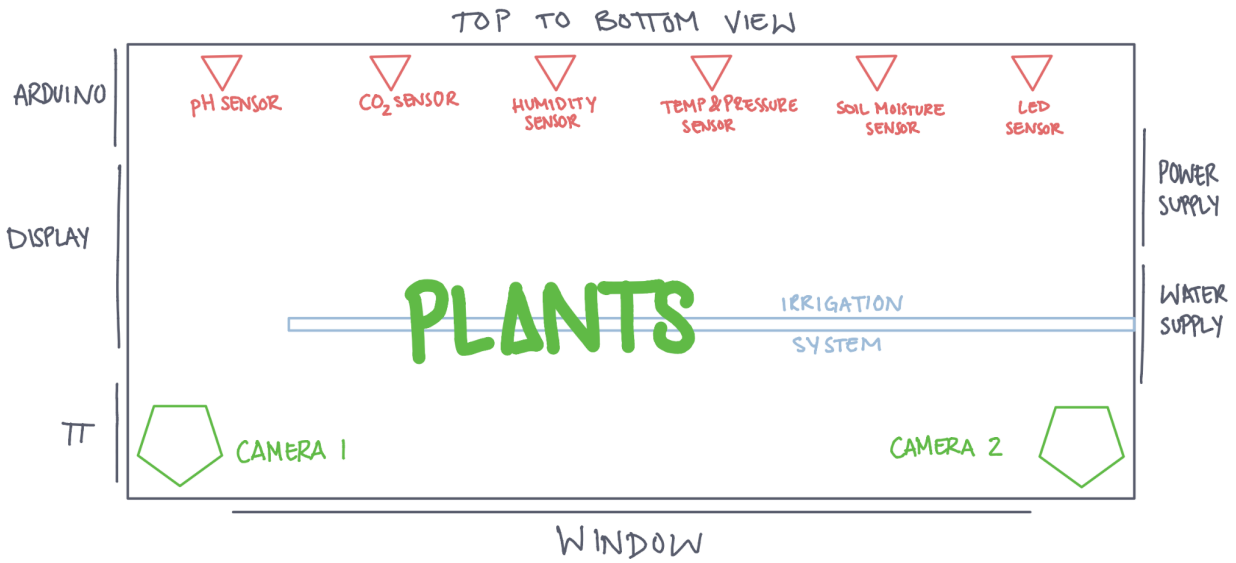


Figure 27: Hardware-focused secondary design

This diagram was further iterated on to achieve more elaborate modules. However, the principles of this design were maintained and the focus remains the same. The team is dedicated to deliver the most minimal model that would achieve all the requirements, reducing crew time for maintenance, reducing costs, development time, and increasing the likelihood of repeatability for further missions.

5.1.1 Software Diagram

The diagram shown in the figure (next page) describes how the user will login and be introduced by a homescreen that includes at least four main functions. The user can then select one of the functions. The functions include, but not limited to, viewing the sensor data, viewing the timeline of plant growth, viewing plant details, and viewing daily tasks.

- The first function, viewing sensor data, introduces a screen that includes graphs of sensory data over time, from pH levels, CO2 levels, etc.
-
- The second function, viewing the timeline of plant growth, portrays the plant growth over time in terms of size to date, yield to date, expected size in the near future, expected yield in the near future, etc.
-
- The third function, viewing plant details, includes real-time data on plant health, plant type, plant division, class, order, family, genus, species, etc.
-
- The fourth function, viewing daily tasks, will display watering procedures, lighting requirements, and plant harvesting details for the day. All of these functions will be available in the GUI of the plant habitation system.

This GUI will allow users to interact with the machine easily without having to go to the system itself and check how the plant is doing. It will portray and maintain the plant's health and allow users to easily access information and data on the plant. The GUI will be based on the database that will input and store all the data from the different sensors inside the habitation system.

This database will be based on the SQLite database since the database is small compared to other databases. There will be several tools included in the GUI that will allow the users to compare the data from the database, to better understand what conditions the plant best grows in. A draft of how the software design will be developed and used is shown below in Figure 28.

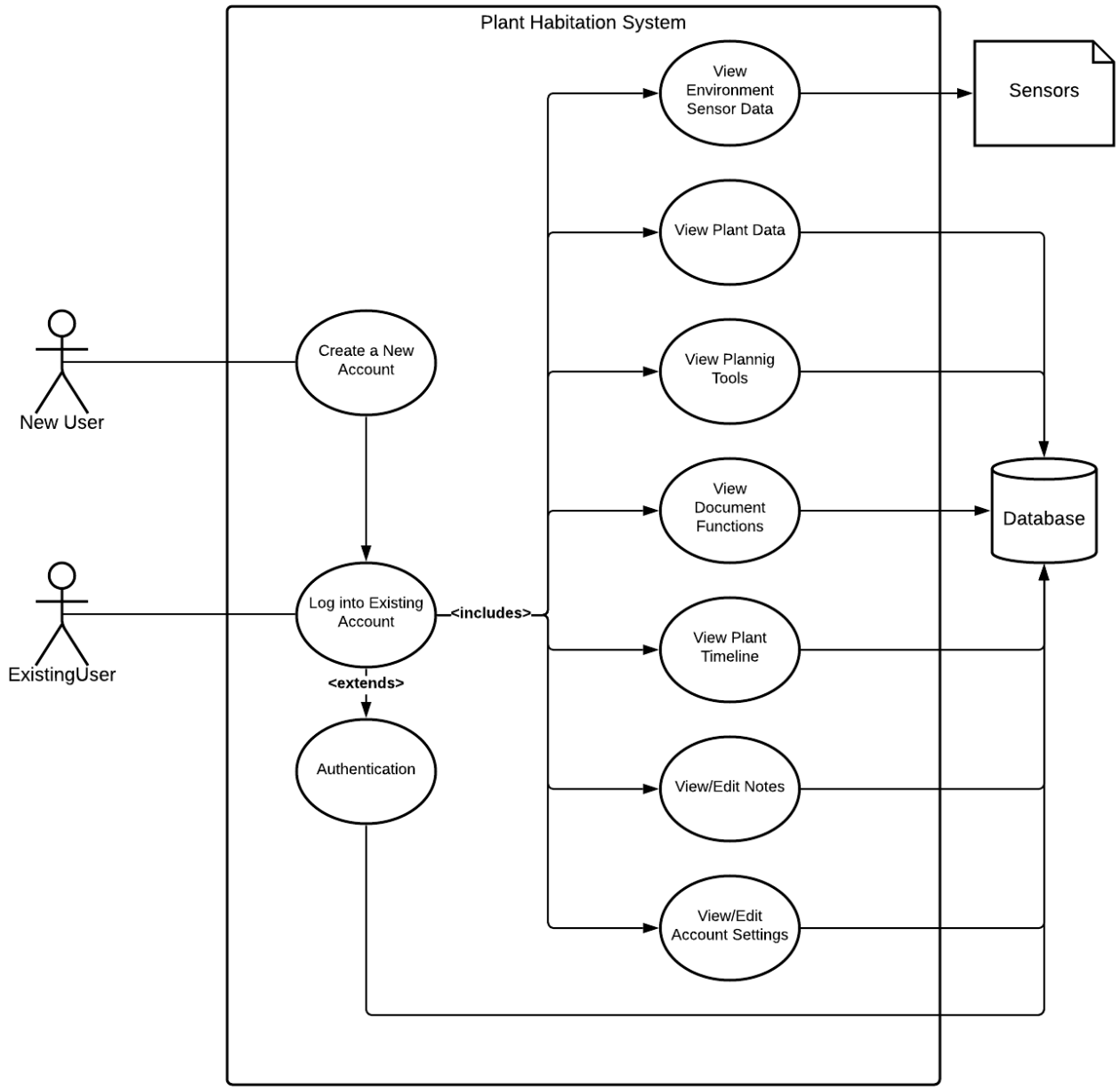


Figure 28: Use Case Diagram

5.1.2 Hardware Diagram

The diagram in figure below displays the connections between the GUI and hardware components. It also gives a general layout of the system. On the left, user input will be entered on the display module and sent as a command through the GUI. Data is then transmitted between the plant habitat and the GUI. Note that a majority of components such as the sensors, camera, and lighting lie in the plant habitat which are each controlled and regulated by the microcontroller.

Some sensors will measure environmental factors such as temperature and CO2 levels, others will be used to monitor plant health such as soil moisture and pH. The water supply module will include a water reservoir and an irrigation system to distribute water. The lighting module will consist of LEDs to maintain appropriate light levels. The air filtration and temperature modules will regulate the CO2 levels and temperature of the habitat. On the far right is the power module which distributes power to the entire system and is also regulated by the microcontroller.

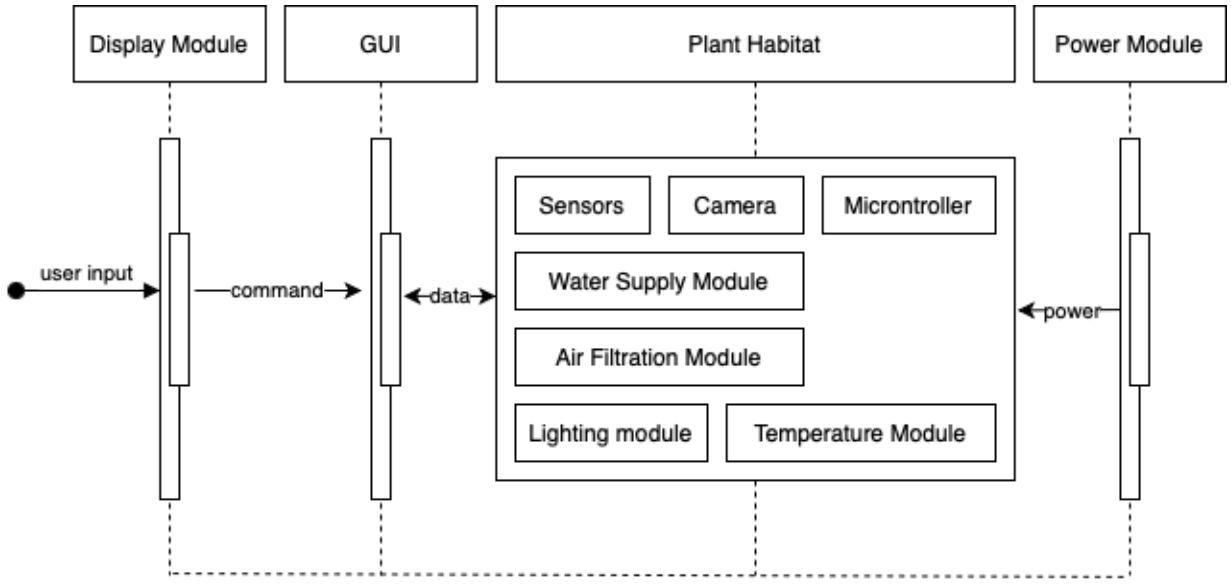


Figure 29: System Design Diagram

The plant habitation system will accommodate space for a viewing window on one side panel of the enclosure so researchers can view plants in their current state. Connections between components must be protected from water, dust, and soil. The sensors must also be laid out so that the probes do not interfere with one another. The following four sections describe the layout of each subsystem.

5.2 First Subsystem: Encloser

The system will be enclosed in a 6U CubeSat according to the NASA requirements [1]. All the components will be inside the enclosure except the power module, water pump, water tank, and the display module.

5.3 Second Subsystem: Power and Irrigation

5.3.1 Power Module

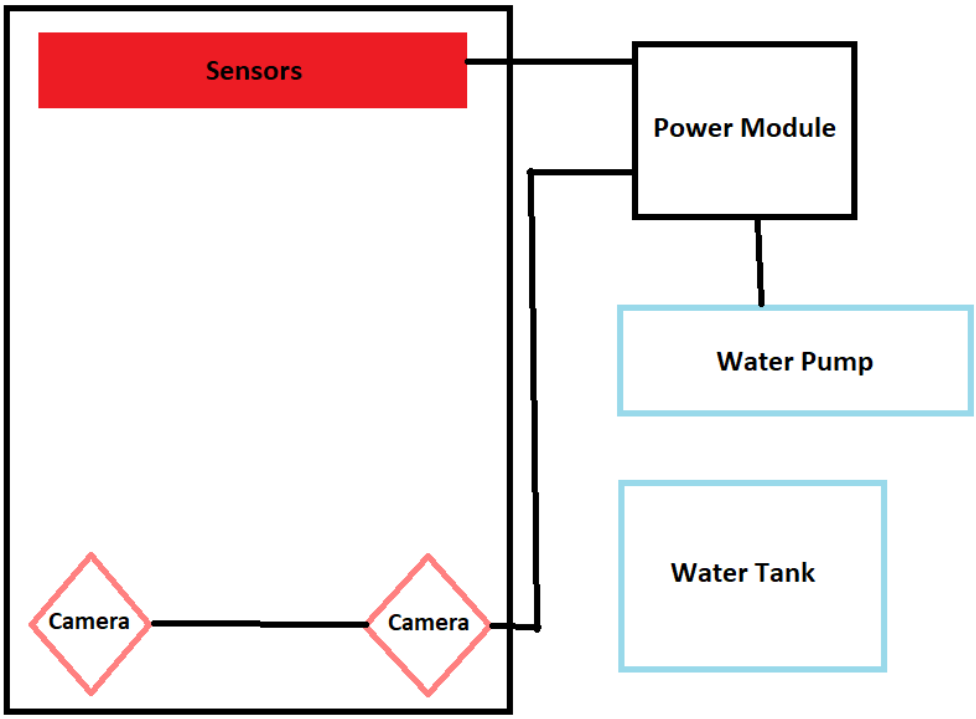


Figure 30: Power Module Connectivity

The power module will be supplying power to various sensors and components. In the figure above, the power module is displayed in the top right. It is connected to the sensor array to the left of it, as well as the cameras and the water pump right below it. Because of the demand on the Arduino and the Jetson Nano to power every sensor, analyzing the maximum current output through their integrated 3.3V and 5V pins is crucial. The maximum current output from the Arduino Mega 2560 from it is one 3.3V pin and four 5V pins, are able to provide a current up to 50 mA.

The Jetson Nano also allows for various outputs including the 50mA on the 3.3V rail and 500mA on the 5V pins. Therefore there will be different volts of power going to sensors depending on how much they will be needing and determining this need is necessary for designing a power module to provide the various voltages to each of the sensors. The three different amounts that will be distributed are: 3.3, 5, and 12. The power supply will be connected to a battery that will have a switch. This switch will control the supply or if a backup battery is needed to be used. The power module will go on the outside of the plant environment, right above the water pump and irrigation system. The power module will be aligned with the sensors but also connected to the two cameras and lights inside the plant habitation system.

5.3.2 Irrigation System

On the left, there is the box that contains the plants, sensors, soil, and cameras. This box will be the plant environment. It will also have the viewing window on it which is the display module. The display module will display the GUI that is being designed to show the data and information of the sensors. This box is connected to the water pump which pumps water from the water tank located on the outside of the enclosure. The power module, which is connected to the water pump, the cameras, and the sensors, is displayed above the water pump. Inside the box, the plants sit on the bed of soil. The water pump will pump water into the soil during the times that water is supposed to be dispensed. The soil will also be connected to the soil moisture sensor which will depict when water is needed. The irrigation system must also provide sufficient space for other environmental probes and sensors. The cameras will be taking photographs of the plants and the sensors will transfer data to the GUI.

Figure 31 below shows the top down view of the plant habitation system that is being built. The figure also includes a key that provides more contextual insight on the design. More details on the specific components selected for this design can be found in the parts section of this document.

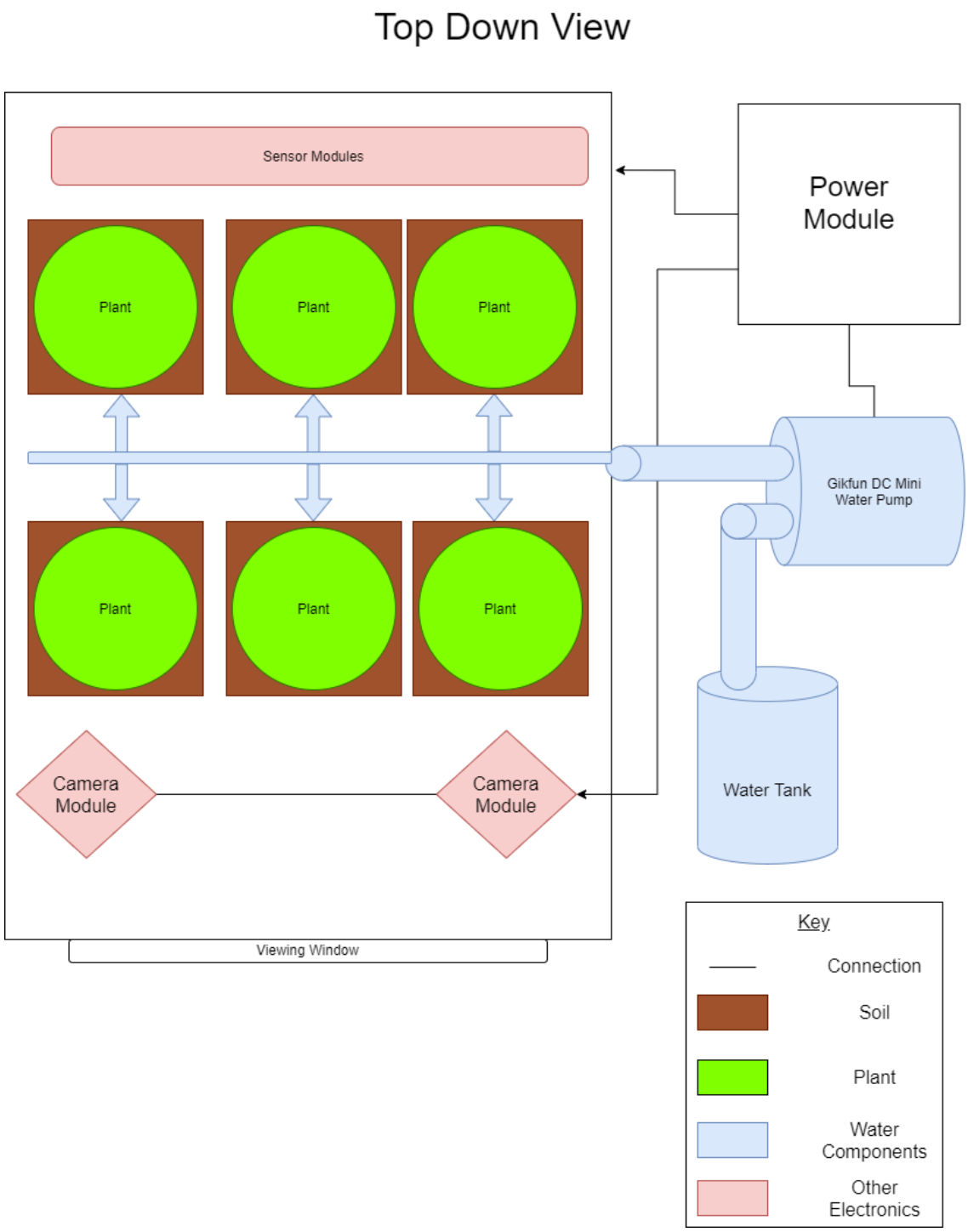


Figure 31: Irrigation System

5.4 Third Subsystem: Sensors and Cameras

5.4.1 Sensor Array

The figure below shows a side view of the sensors inside the plant habitation system. A color-coded key has been added to the left of the diagram to help others depict the difference between the components of the figure.

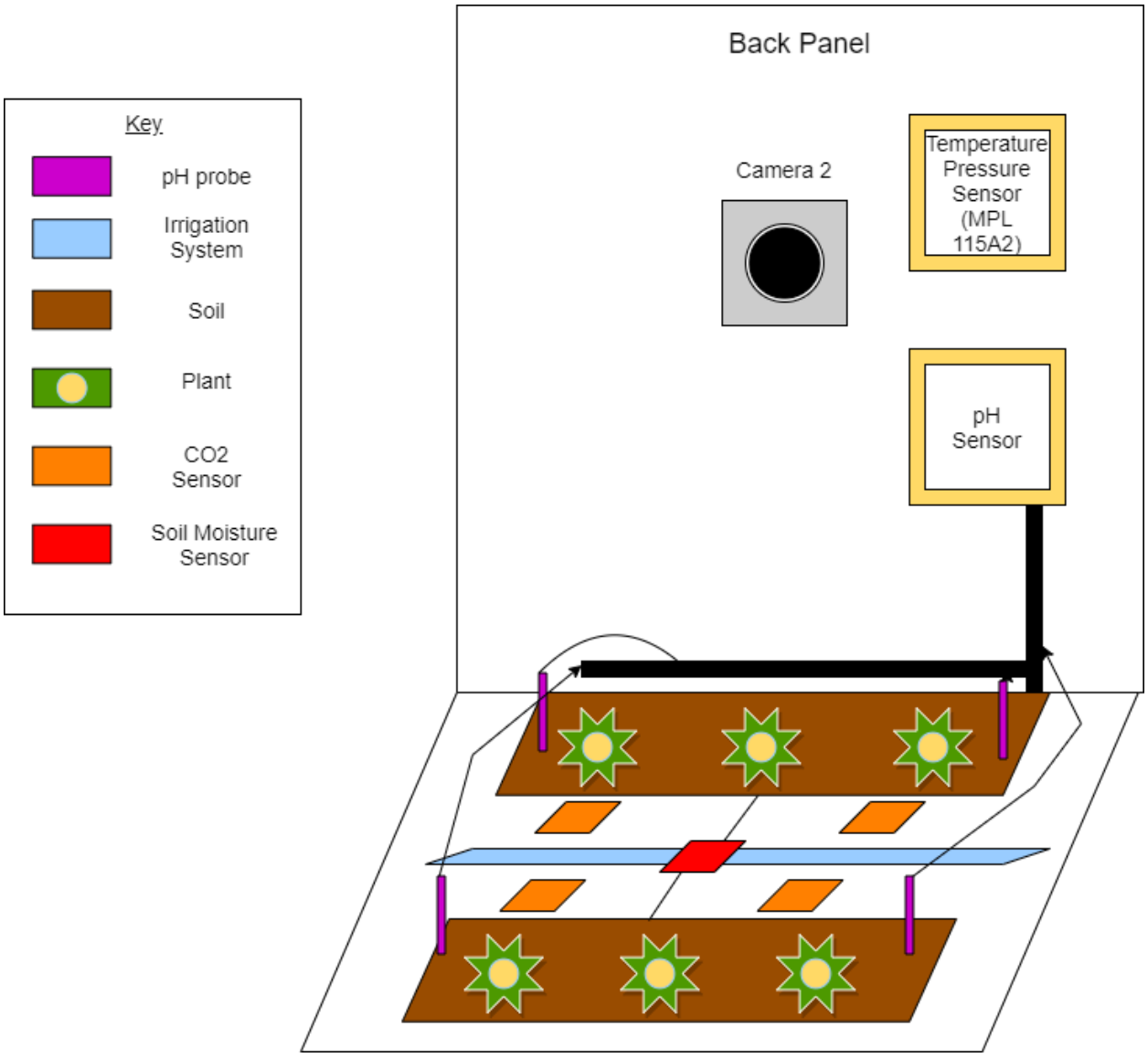


Figure 32: Sensor Array Diagram

On the back panel, there is the pH sensor and barometric temperature & pressure sensor. There is also the camera 2, which will show how tall the plants are growing. The pH sensor is connected to the pH probes which will gain insight from the soil of the plants. The soil moisture sensor, CO2 sensors, and pH probes are located near the soil and plants on the bottom panel. Although not visible in the diagram, LED light sensors will be located on the top panel. The camera system goes into more details on the components located on this panel. The soil moisture sensor will be connected to the soil of the plants to get a sense when water levels are low. The LED light sensor will be connected to the lights that will be at the top of the system. All sensors will be connected to the power supply which will be to the right of it outside the plant environment.

5.4.2 Camera System

Initial designs of the plant habitation system had both cameras on the same panel to provide users with a closer view of plants on each side, but this quickly changed due to the needs of our computer vision tasks. Ultimately it was decided to place one camera on top and one located on a side of the enclosure to assist in proper plant leaf segmentation. To allow for easy connections to the power module and capture the best view of every plant in the habitat, the first camera was centered on the top panel. The second camera was centered on the back panel.

In addition to the camera 1, on the top there will be four columns of LED lights that will provide light to the plants, aiding in growth. Below the top panel is the back panel of the system. This shows the camera in relation to the ruler and sensors. This way, there are photographs from the top, which will show the amount of leaves on the plant, and the back, which will show the length of the plant. In the top right corner, the view of Camera 1 is shown. Since this is from the top, users are able to see the sensors, ruler, camera 2, all plants and the amount of leaves that are on it, as well as the water irrigation system. At the bottom right, is the view from camera 2. This shows how tall each different plant is growing. In the middle is the view of the entire system as well as the power module and irrigation system.

The figure below displays the final design for the camera system inside the plant habitation system as well as the perspective for each camera. Note that the display module will remain on the left hand side of the enclosure.

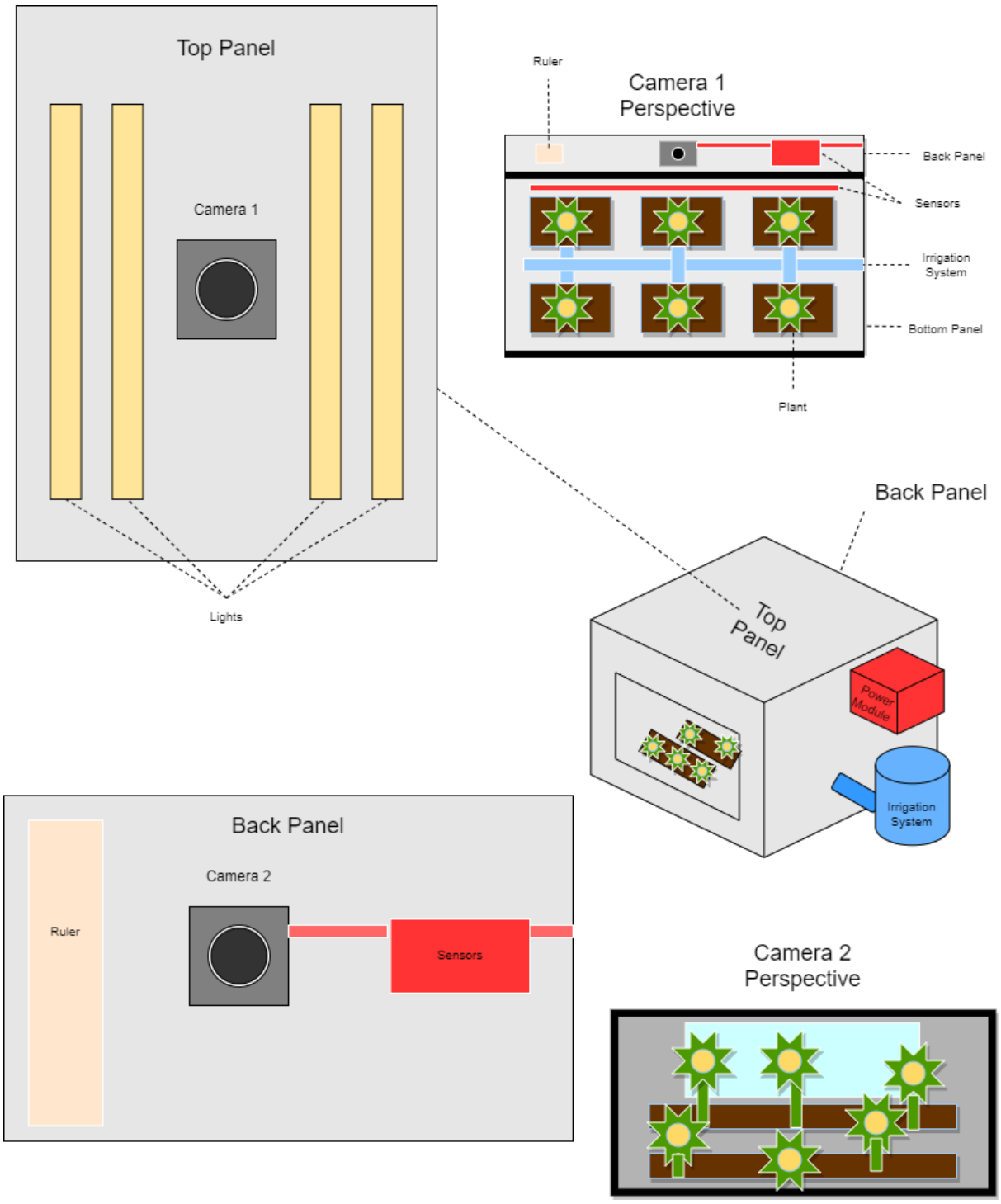


Figure 33: Camera System in Plant Habitat System

5.5 Fourth Subsystem: Computation and Interface

5.5.1 Computational Units

The two computation units for the plant habitation are the Jetson Nano and the Arduino Mega. The Jetson will be receiving the sensor data and sending it to the GUI that will display that data. The Jetson will also process data from the dataset to predict plant health. The Arduino will be the software for the sensors. This data has to be precise since users will be using it to change the environment conditions. Both of these units are critical to make sure the data being displayed is sufficient.

5.5.1.1 Jetson Nano

The Jetson Nano serves as the core computational unit for the system driving both the GUI as well as the stream of data from the sensors and camera units. The computer is capable of supporting an operating system due its dedicated NVIDIA graphics card. The graphics card helps push the computation in parallel, optimizing performance. It will also serve as the computational node for the robot vision processes of the project.

5.5.1.2 Arduino Mega

The Arduino Mega uses its open source integrated development environment software. This allows for the various hardware manufactures of sensors to provide their own libraries for each of the sensors that were selected. This will allow for a seamless integration of the various sensors into one output stream. The microcontroller is necessary for processing the analog singles that are received by the various sensors. This is a feature that is not available on the Jetson nano and along with the sensor libraries provided will make for easier computation and data intake into the database.

5.5.2 Display Module

A display module will be displayed on the plant environment to show the GUI. It will display the sensor data, photographs, and allow users to be able to log in and see their notes. This display module will be mounted on and have a touch screen so users will be able to check data by using their finger. The module will have power through USB connected to the power module, which will also be right outside the plant environment. This way, the screen of the module doesn't have to worry about getting dust or dirt on it. Being able to see the graphic information of the sensors and plants on the screen will be beneficial for anyone that wants to look at it when they are visiting the habitation system.

Note that the display module will also contain buttons and button prompts on the screen to allow for easy communication between the parts of the system. End users will not need to manually enter a command to access specific data, but instead press the correct set of prompts. The display module will provide complete control of the different modules and subsystems. The figure below shows the display module in relation to the system:

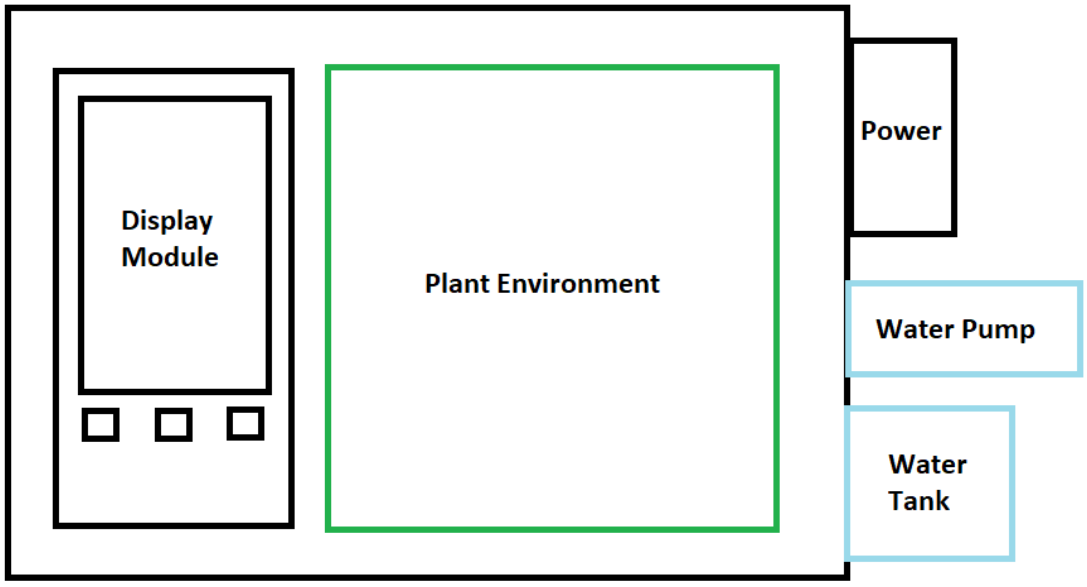


Figure 34: Display Module on System

5.6 Software Design

5.6.1 Graphical User Interface

A GUI is a Graphical User Interface that allows for easier interaction between the user and a machine [8]. The interface usually has common user interface icons such as buttons and text that the user can interface with instead of having to use a command-based communication. Before GUIs, everything had to be done through a command line. Developing a GUI is important because it allows simplicity for users, it is visually appealing, and it also allows users with no computer knowledge access to different components of a computer. Having a GUI for this project will be helpful for scientists and astronauts who need to access the interface to check on the growth and health of the plant. Even if they have no programming experience or knowledge, they can easily perform simple tasks in just a matter of a few clicks.

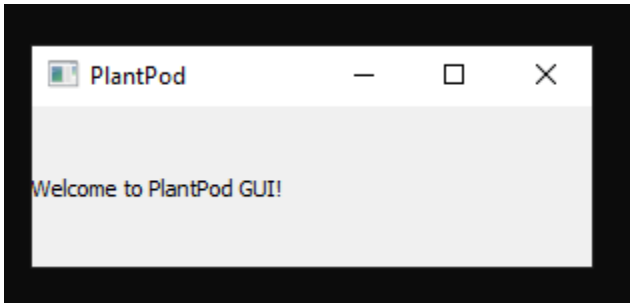


Figure 35: Example of Simple Startup GUI

A couple of the imports that will be used inside the GUI are: import sys, import Qt from PyQt5.QtCore, and QApplication, QLabel, QWidget, and QMainWindow from PyQt5.QtWidgets. Sys is system-specific parameters and functions; it will provide access to variables used by the interpreter and to functions that interact with the interpreter [61]. In the figure below taken from the GUI code, sys is being used in the main thread of sending the information to Python. Then exit is used to exit from python.

```
14
15  if __name__ == '__main__':
16      app = QApplication(sys.argv)
17      window = Login()
18      window.show()
19      sys.exit(app.exec_())
```

Figure 36: Import Sys Example

The rest of the imports (QLabel, QWidget, etc) are being used to open a GUI window, setting the title as PlantPod and labeling the window to “Welcome to PlantPod GUI!”. This will be the window of the GUI that users will be seeing and using when they first open up the application.

```
class Login(QMainWindow):
    def __init__(self, parent=None):
        super().__init__(parent)
        self.setGeometry(100, 100, 280, 80)
        self.move(60, 15)
        self.setWindowTitle('PlantPod')
        self.setCentralWidget(QLabel("Welcome to PlantPod GUI!"))
```

Figure 37: PyQt5.QtWidgets Example

5.6.2 Account System

The account system will be a simple database table that holds the username, password, and user id. The GUI will check against this table on login, and grant the user access to the user id’s notes. The notes themselves will be visible in the account management page, or on the plant data page. The goal of the account system itself will not be security. The password is more of a formality than an actual security measure, as GUI will be attached to the growing box and the data will not be encrypted anyways. The point of the system is to separate notes from other users and make them easier to read.

5.6.3 Data Management

The plant data will be stored in a database table, and will have a record of all of the sensor's data attached to the time they were taken at. The table will be designed as such:

id	date	sensor data 1	sensor data 2	...	link to image file	

Table 9: Database Table Sample

The GUI will have a page dedicated to viewing and managing the plant data. It will include the ability to view a single entry, view multiple entries side by side, edit an entry in case of the data being wrong, and the ability to look at trends.

The goal of the GUI is to give the user a streamlined interface to look at the data collected and make inferences. There will be several tools that allow the user to compare the data against itself, to better understand what conditions the plant best grows in.

5.6.4 Computer Vision

This section will cover the setup and goals for the portion of the project that focuses on computer vision. The overarching goal is to identify and segment plant leaves in order to predict plant health. As a result, high yield analysis can then be performed.

5.6.4.1 Cisco AnyConnect Virtual Private Network

Cisco Anyconnect is a VPN that allows the user to securely connect to a remote network from a local network. This is achieved through the VPN basic functions that include: authentication, authorization, and creation of a secure path in the network. In addition, VPN encrypts data inside the IP data packet. [56]

Below is a figure that portrays the VPN Tunnel: the packet is carried between the two routers where only IP addresses of the routers are visible.

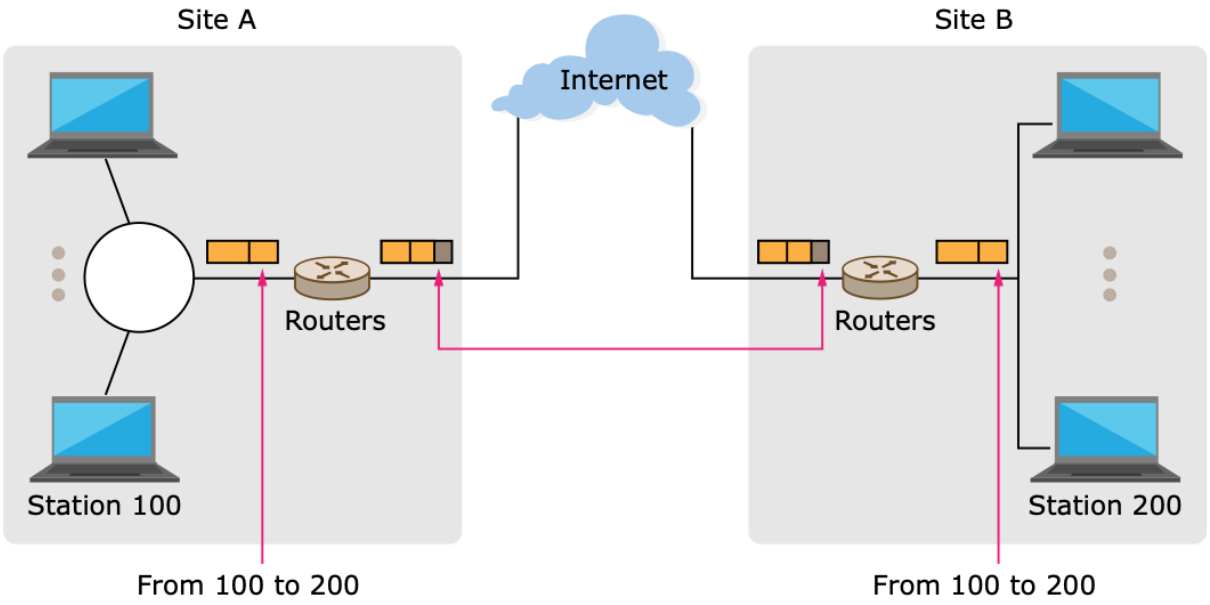


Figure 38: VPN Tunnel

By following the guide provided by UCF IT Support Center, installing Cisco AnyConnect provides access to UCF resources remotely on any of the following operating systems (Mac OS, Linux, and Windows). After installation, ensuring a secure connection can be achieved through “secure.vpn.ucf.edu” then logging in with an active UCF ID and password. [57]

The interface for connecting through Cisco Anyconnect is depicted below:

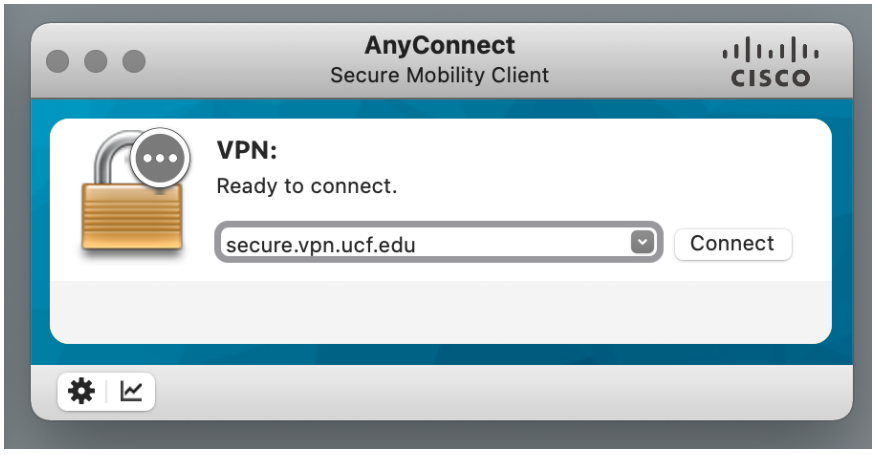


Figure 39: Cisco Anyconnect prompt

5.6.4.2 Newton HPC at the UCF Advanced Research Computing Center

Contacting the Advanced Research Computing Center Institute for Simulation and Training and requesting access to a GPU cluster is the following step. After approval, the generation of public/private key pairs are to be done through the following command on linux terminal: “ssh-keygen”. That will establish the handshake with the host containing the cluster at UCF. [58]

Below is a snapshot of the successful access to Newton HPC at the UCF Advanced Research Computing Center.

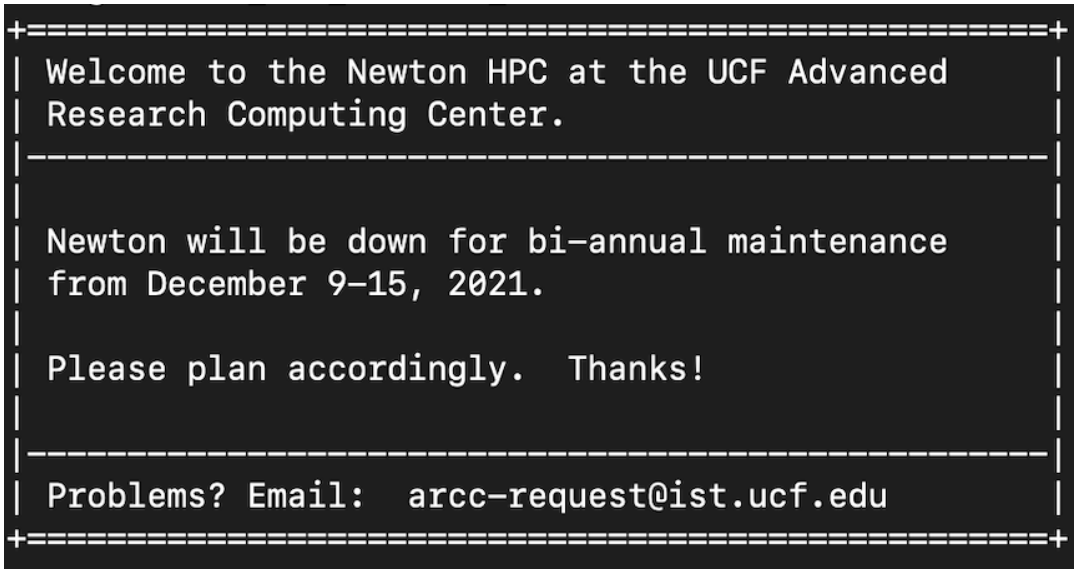


Figure 40: Successful connection to UCF Newton

The high performance computing resources provided by UCF allow the machine learning models to be trained and tested. The Newton cluster is also a GPU accelerated cluster with 40 NVidia Tesla V100 GPUs which will assist with training time and robot vision tasks. [58]

5.6.4.3 GitHub Repository

GitHub is a cloud-based Git Repository that allows developers to store, manage, track, and control changes to their own repositories. Setting up a GitHub Repository provides the team members with version control over code iterations with tools

from the open source Git community. This repository can also hold code for other portions of the project such as GUI. Since two main software components of this project are the GUI and the ML/Robot Vision detection, two different GitHub Repositories have been made for organizational purposes. There is also a Github repository for the database that is going to be used for the GUI. This database will hold information sent by the sensors in the plant habitation system.

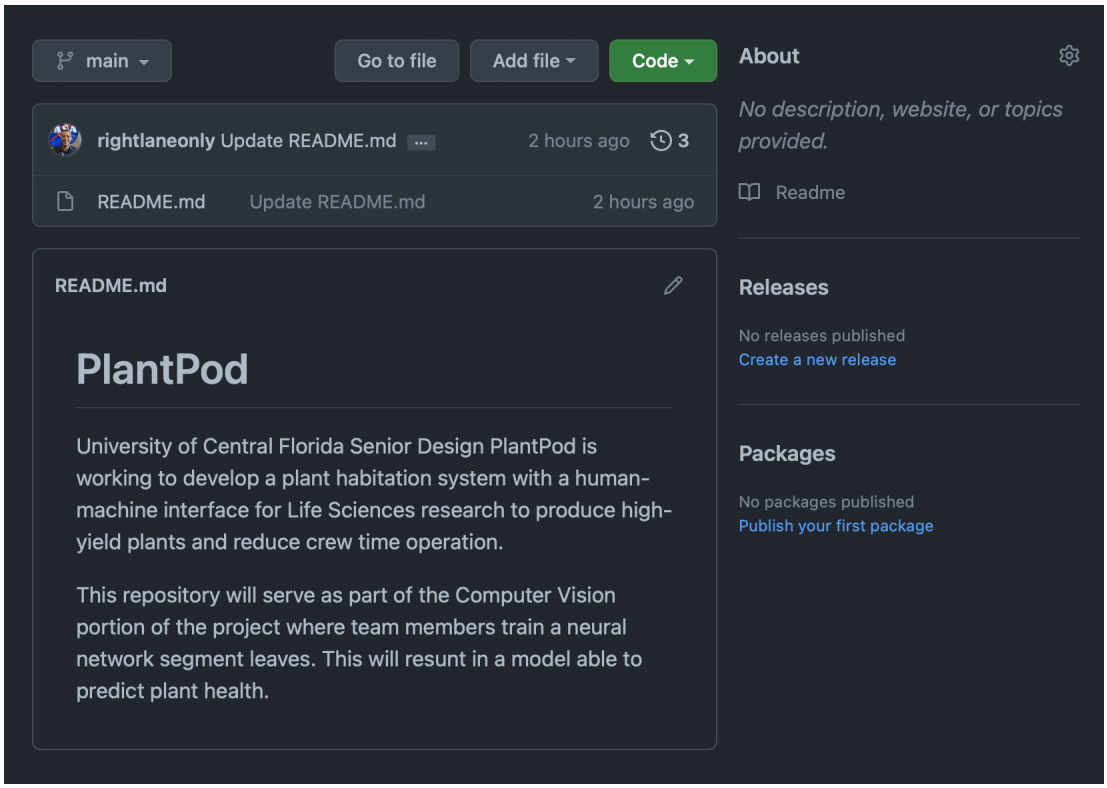


Figure 41: ML GitHub Repository

The above GitHub Repository is for the Machine learning and Robot Vision that will be added into the project during the Spring Semester of 2022. This will be used to study the leaves of the plant(s) in order to see how the plant's health and growth is doing.

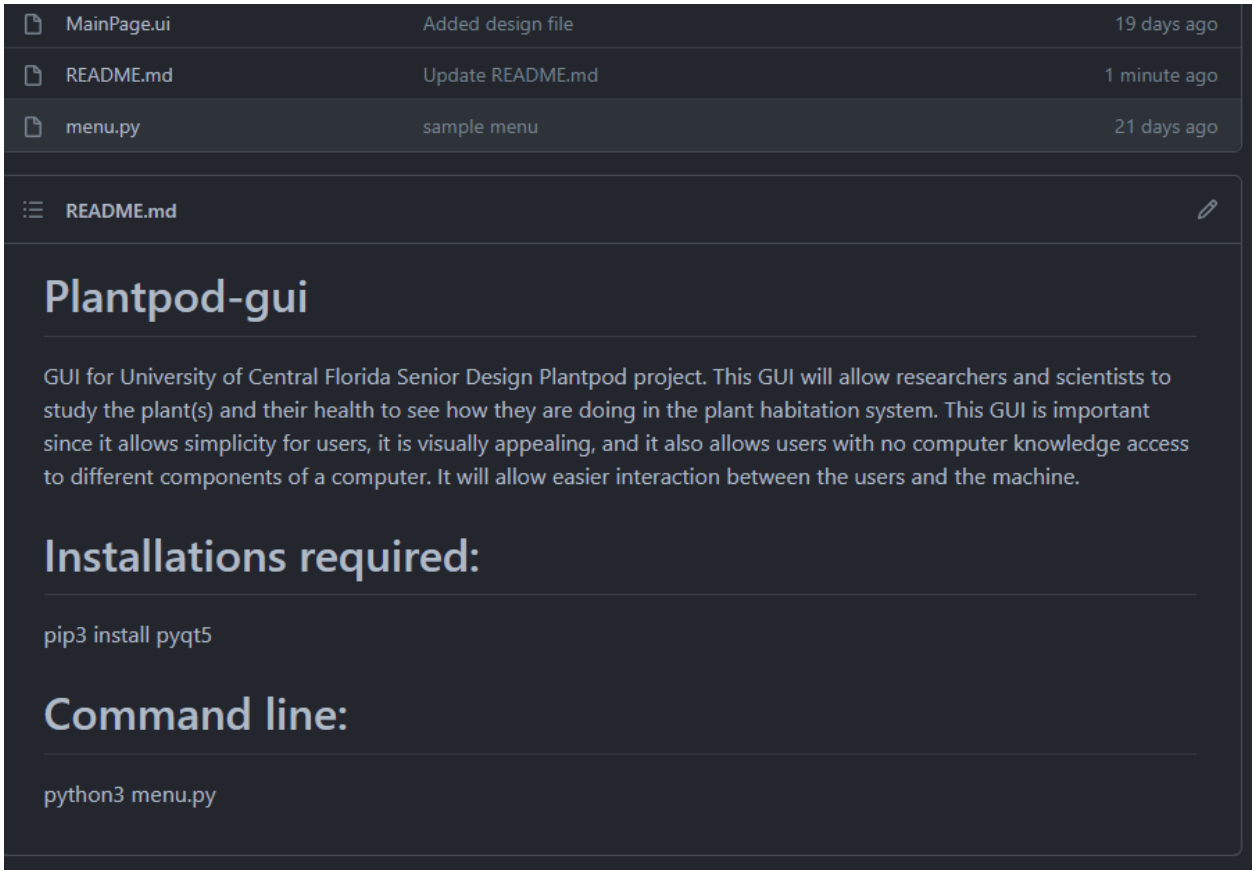


Figure 42: GUI Github Repository

Above is the GUI Github Repository, which will allow users to interact with the plant habitation system without having to have human interactions. This will allow researchers to see the plant's growth and compare it with other data and information that will be provided from the sensors inside the plant habitation system. This will be based on the database that is included in its own repository as shown below.

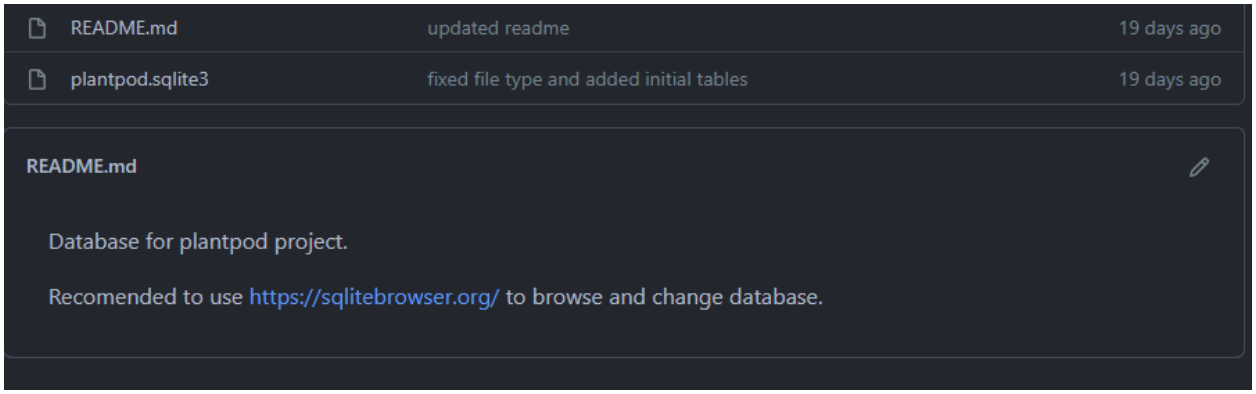


Figure 43: Database GitHub Repository

Above is the GUI Database Github Repository. This will be storing all the data and information that is taken from the sensors in the plant habitation system. This database will be used to showcase all the user's information that was recorded over the past couple entries. This database will also be used in the comparison data that users will be allowed to see in the GUI. In order to browse and change the information in the database, the DB Browser for SQLite (DB4S) will be used. The DB4S is an open source tool that allows users to create, design, and edit database files that are compatible with SQLite [62].

5.6.4.3 Machine Learning Process

The process in developing the code to train the model will be the following:

0. Loading the import statements required for the code
1. Loading the data from the provided CSV files
2. Checking for missing values within the training data and, if required, describing and implementing an approach to handle those missing values
3. Checking for outliers within the training data and, if required, describing and implementing an approach to handle those outliers
4. Describing any data transformations or feature engineering that are required and providing an explanation as to why each is being done
5. Building and training a model on the training data and evaluating its performance on a set of validation data
6. Generating a distribution of validation scores, as well as summary statistics for this distribution

5.7 Summary of Design

The extracts below are from the NASA Systems Engineering Handbook outlining the processes needed to abide by for this project. [72]

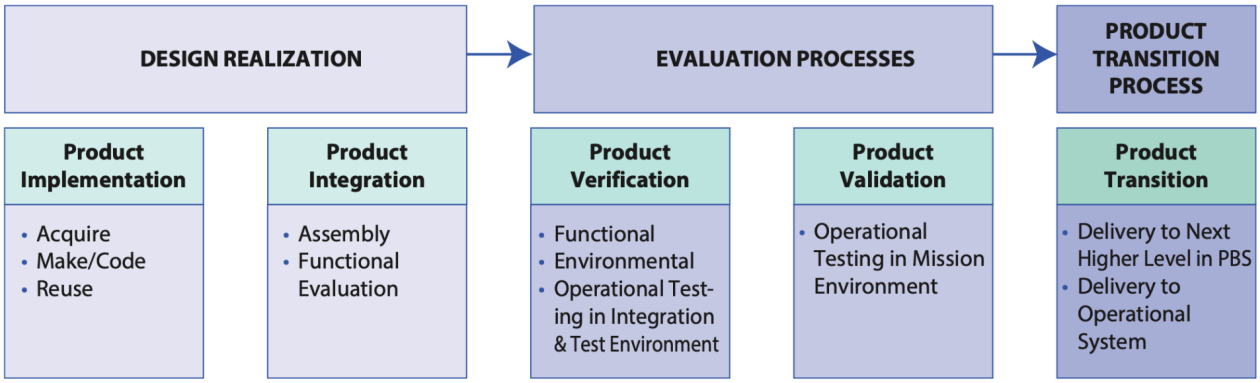


Figure 44: Summary of Design from NASA

The product implementation stage involved the planning and design of each subsystem. This included identifying specifications and requirements to create a successful system, analysis of previous projects, determining the necessary components in order to meet the requirements for the plant habitation system, developing mockups for the hardware and software, coding the GUI, creating repositories for organizing our code, and starting to identify the steps required to move on to the next stage: product integration.

Unfortunately, the funding from FSGC has not been fully processed so the product integration stage is on hold. Once the funding is available the selected sensors, cameras, microcontroller, etc. will be purchased and the first prototype of the system can be assembled. After the parts arrive, some sensors may need to be recalibrated and adjusted to account for the specific qualities of the plants. Hopefully none of the parts come damaged or broken, since this would delay the product integration even more. In the meantime, the GUI is being coded, designs for the enclosure are being refined, and the datasets to be used with the machine learning models are being analyzed. The remaining stages: product verification, product validation, and product transition will begin during the spring semester. Once the enclosure is created and the components are set in place, the evaluation process can finally begin. The details of the system's testing environment and equipment are detailed in section 7.

6. Project Prototype Construction and Coding

6.1 Integrated Schematics

The plant habitation system is going to need to supply power to various sensors and components. Each sensor is designed to to its own specifications for input power varying from 12 volts, 5 volts and 3.3 volts. Each sensor will be given power through a power management PCB. The board is designed to take a 12 volt DC input from a power rectifier that uses US standard 120v two prong plug. This rectifier can be designed to be plugged directly into the side of the PCB using a 2.1mm jack or a female adapter to wire-block terminal for ease of flexibility in small confined spaces. The power PCB will also need a terminal connection to convert the 12 volt battery supply to the various DC voltage outputs. The battery will need a switch to control when the power supply is connected or when the back up battery is being used. This switchover circuit can be designed using an automatic circuit. Or simply using a diode to prevent the backfeeding of the battery and overcharging. Another useful feature that needs to be implemented is an external or integrated trickle charger for the battery. That way the battery remains at a consistent level even during stages of none use to eliminate parasitic loss.

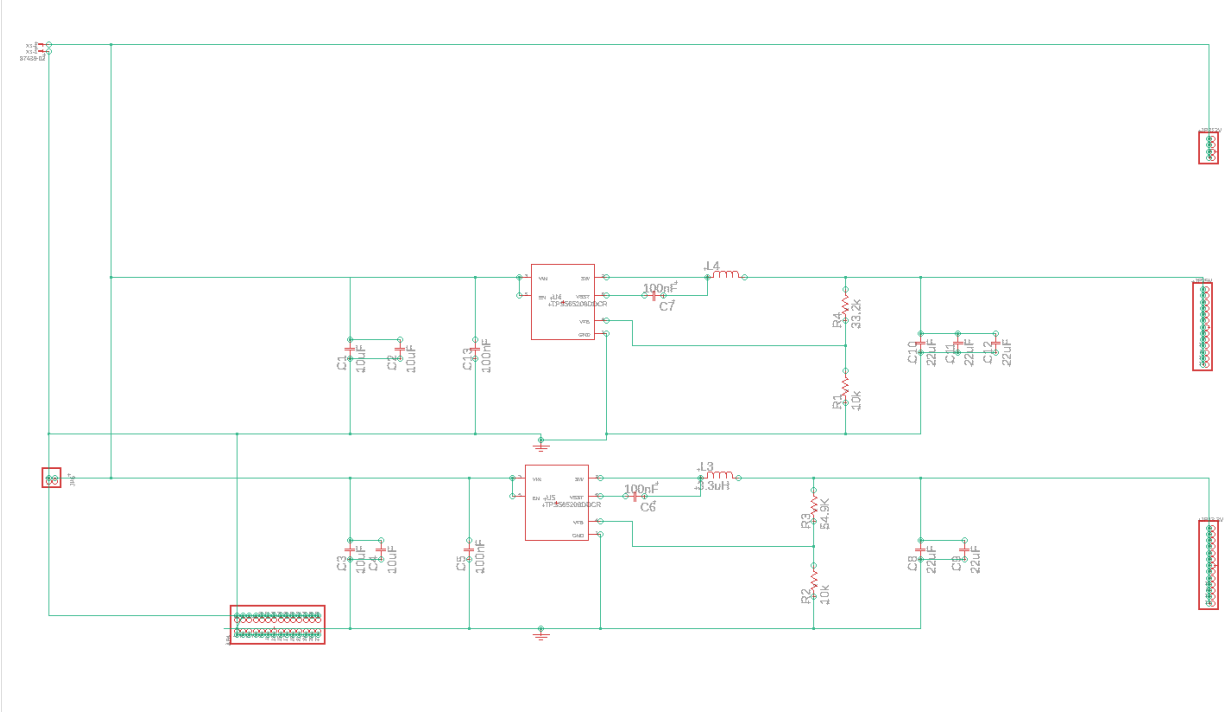


Figure 45: Power PCB Schematic

The schematic above only includes the pin headers and inputs for bothe the 12 volt regulator and a 12 volt battery.

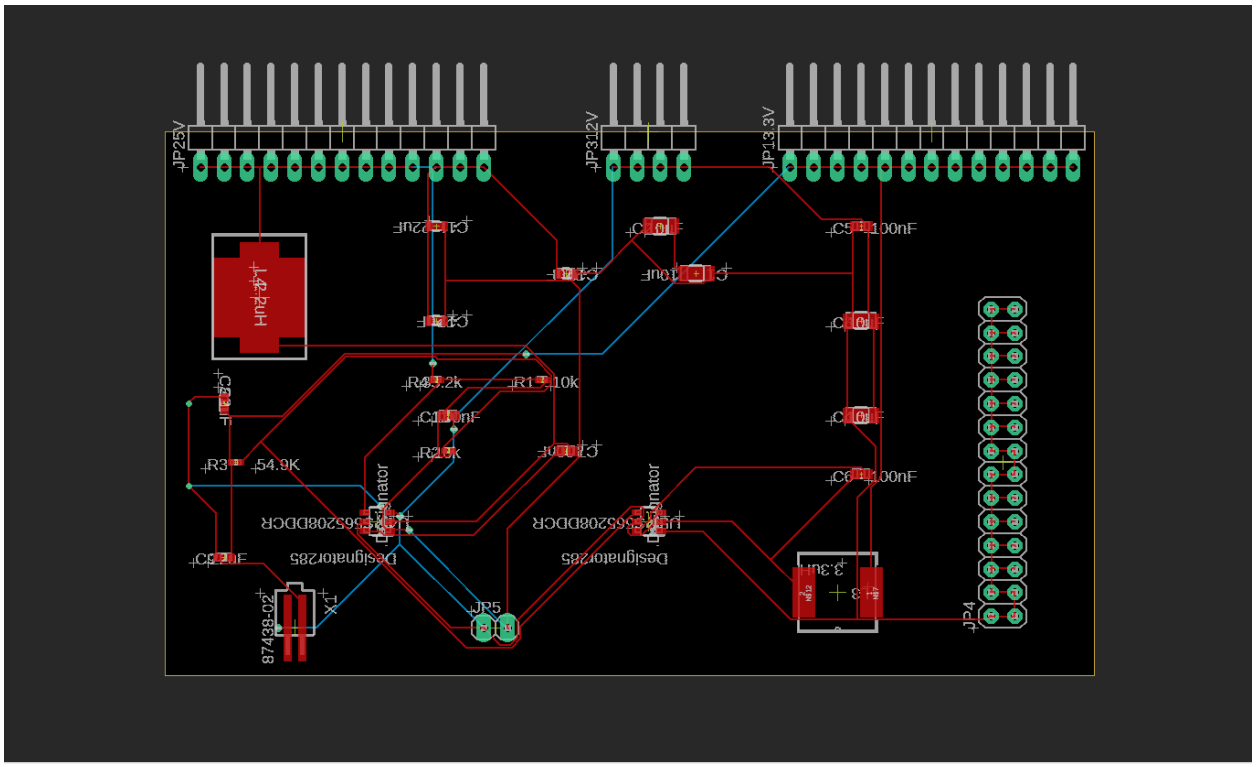


Figure 46: Power PCB Board Layout

The board layout was set up so the pin headers are located on a single side perpendicular to the board. The Grounding header is located to the right and is vertical in orientation. This might be necessary to change if the space retrains become an issue in prototyping. The board might benefit from two external USB headers allowing power to be delivered to both the Arduino Microcontroller and the Jetson Nano.

6.2 PCB Vendor and Assembly

When choosing a PCB vendor it is important to take in mind the manufacturing time and quality of the products. Well known manufactures have very quick turn around times and easy to use websites that interface with the users and allow the users to upload and have thousands of parts in stock so there will be no waiting. One option to consider is if the user wants to have it assembled at the manufacturer or do it

themselves. In the case of a project like this, a single board will not take long to solder components onto and is a good learning experience for group members.

6.3 Final Coding Plan

6.3.1 Main Page

The blueprint of the main page, accessible via the display module is shown below with description below them:

Figure 47: Main Page Blueprint

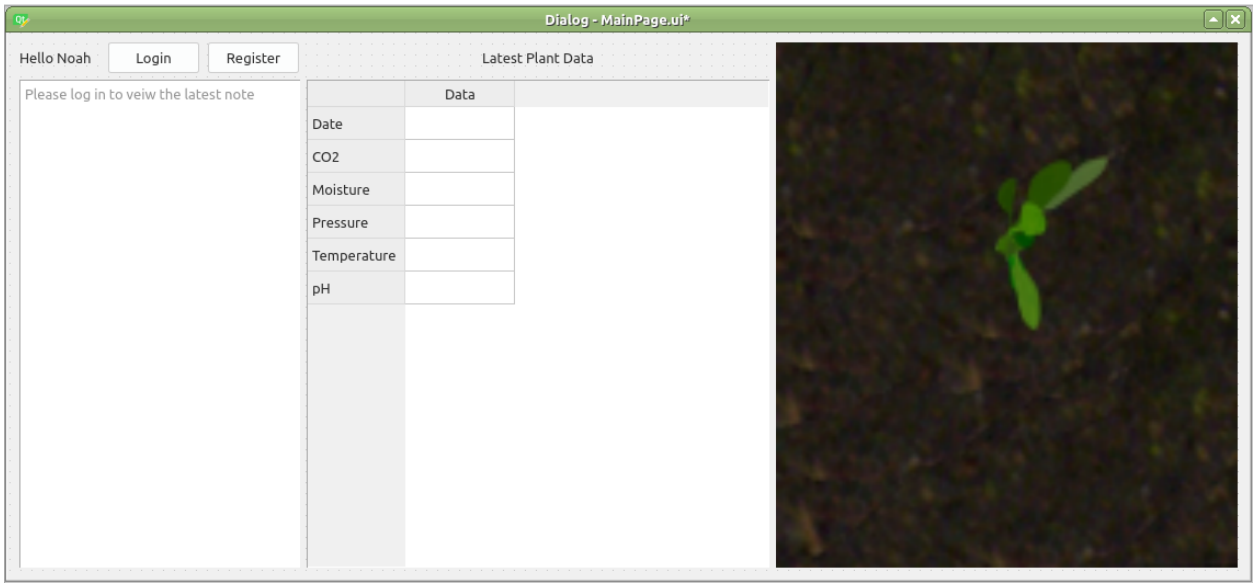


Figure 48: Main Page Figma Design

Above is a Figma design of how the Home Screen will look for the GUI. The first figure depicts what each box is going to be displaying, while the second one gives a view of how the GUI will actually look. It includes the username of the person that is logging into the portal. If they need to register for a new account, they can do that as well. Once logged in, they will be able to view their latest user note, which will be what was edited and saved right before logging out of the account. There is also a section in the middle where the sensor data will be displayed for the sensors in the plant habitation system. The right side is the plant image of the latest plant that was taken from the plant habitation system. Since this is the home page of the GUI, it will be displayed to anyone that is trying to access the database and any of the sensor data or information. They will have to login in order to be able to see any of the information in the GUI.

6.3.2 User Page

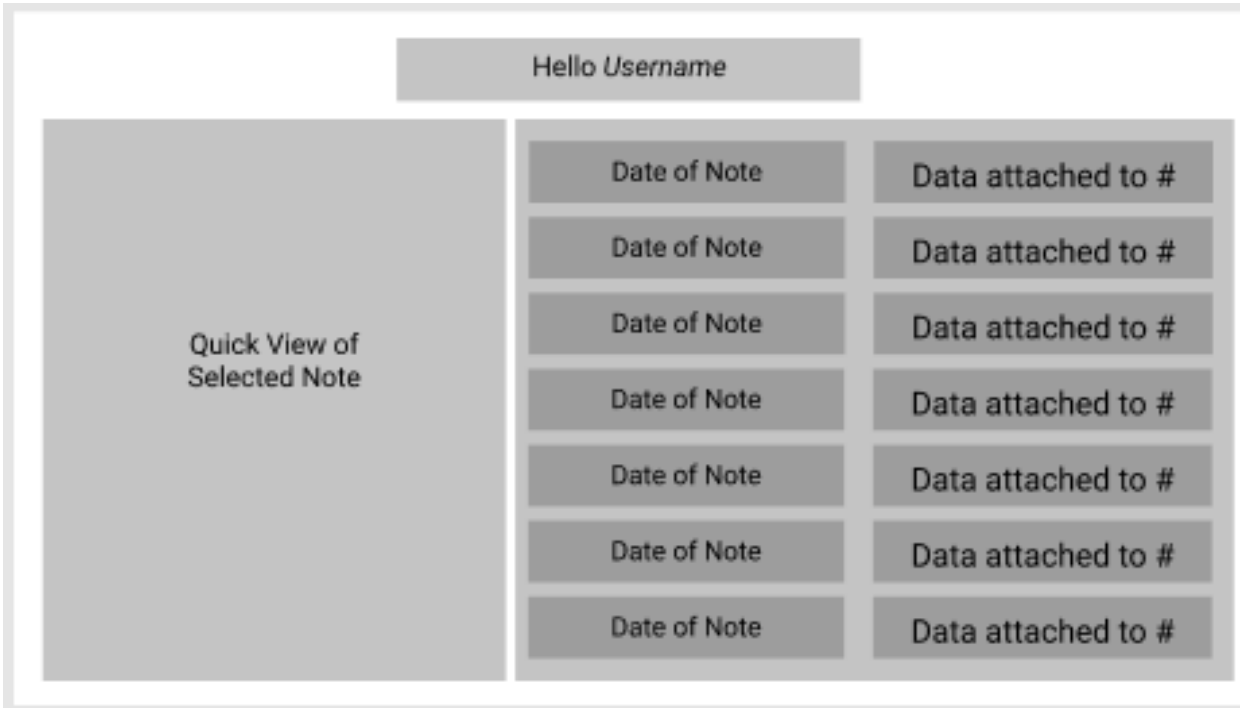


Figure 49: User Page Figma Design and Prototype

Figure 50: User Page Figma Design and Prototype

The User page will consist of the specific user's account and notes. This is where the user can view any past notes that they have made. They will be able to differentiate between the notes based on the date of the note. Right beside the date, is the attachments that they can add to their notes such as pictures, graphs, etc. When they select a note, they will be able to see a quick view of it or double clicking on it will allow them to further look into the entire notes document.

6.3.3 Data Page

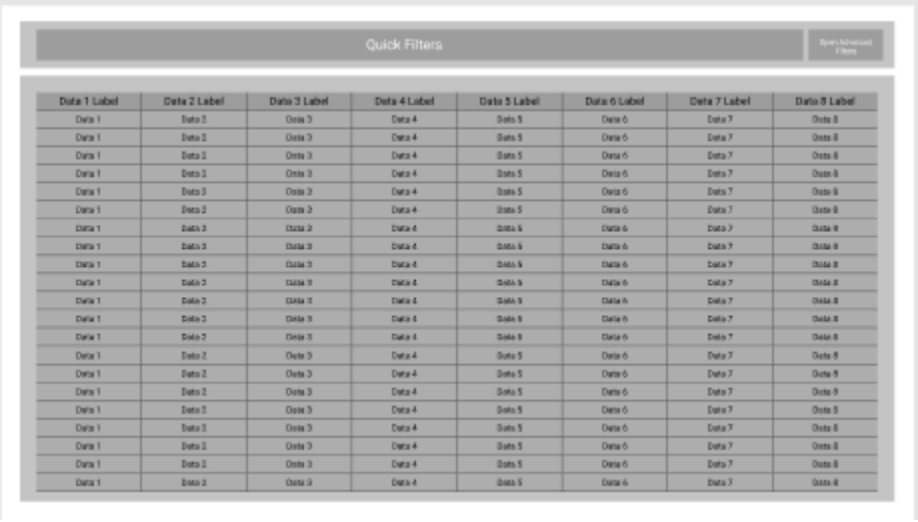


Figure 51: Data Page Figma Design

The data page shows all the data gathered about the plant(s) that are in the system. This will include all the sensor data, the dates, the photographs, and other links. This will contain all the information of the sensors that is gathered from the plant habitation system.

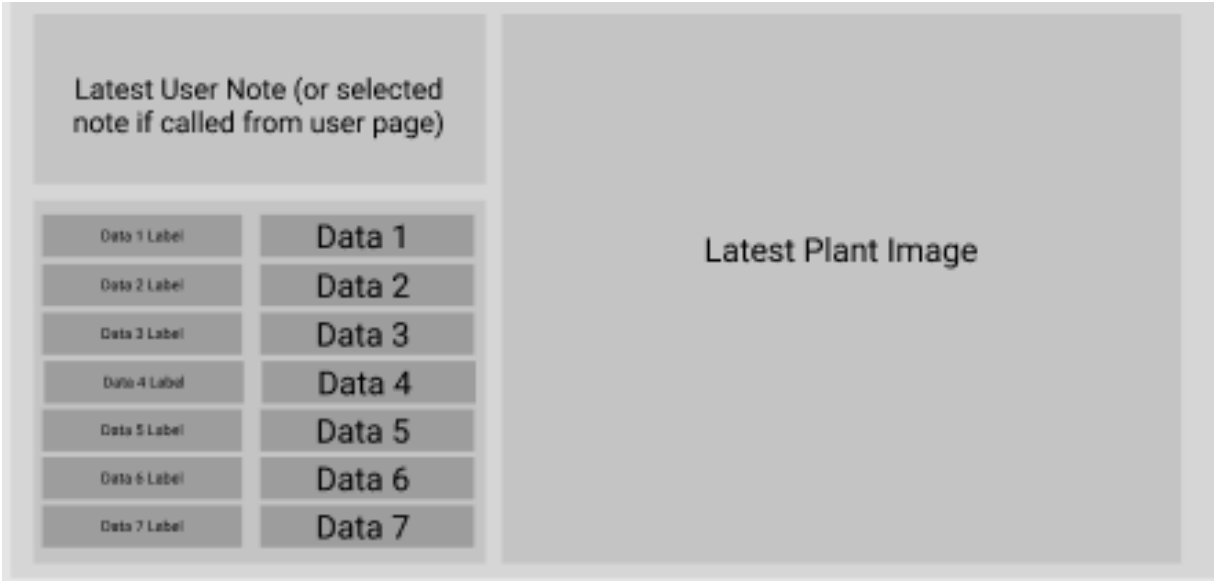


Figure 52: Specific Data Page Figma Design

When a row is selected to view, it shows the same information that users can look at when they click on their user page. This information is available for anyone to view that can log into the system. It doesn't have to only be their information or notes that they have put into the system. It will portray the notes of the user, the data, the description, as well as an image of the plant. This can be helpful when the user needs to go back to look at multiple data.

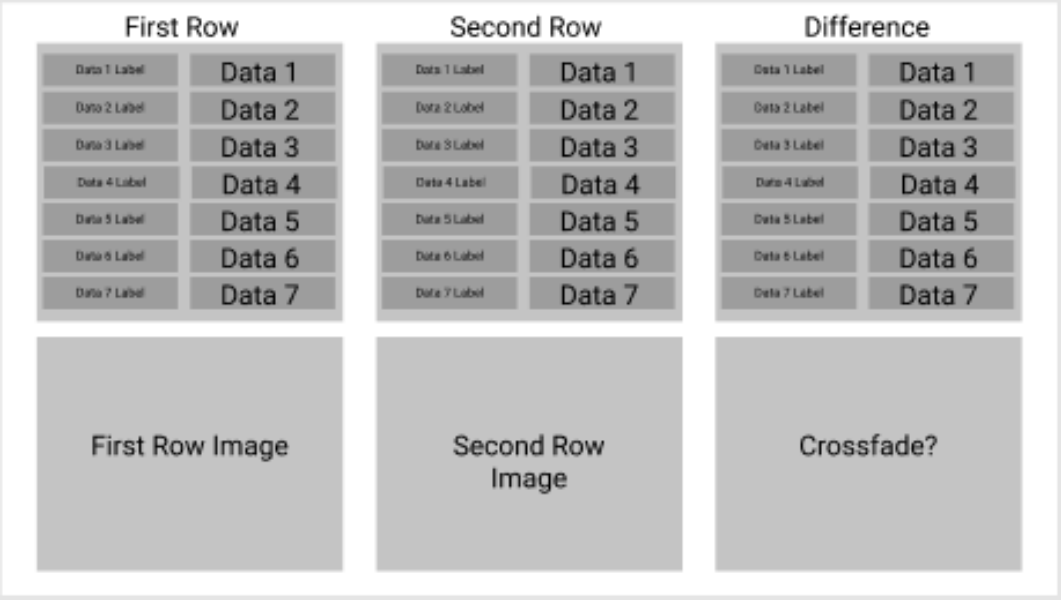


Figure 53: Comparison Data Page Figma Design

When two rows are selected from the data page, both are displayed alongside their data and images. There is a third column that shows the difference between the data, the description, and a crossfader of the image. This will allow users to see the difference between the data and if the plant is doing well or if changes need to happen so the plant can do better. It can show if the plant was doing better in a different light setting or if water levels need to change to allow the plant to grow better. This will be very crucial for researchers since it will allow them to find the perfect environment for the plant to grow in.

6.3.4 Database Mockups

The figure below shows a mock up of the database and the relationship between the tables. The database will include the user's information, all the notes that are in the GUI as well as its dates, and the sensor data that is taken in from the sensors in the plant habitation system. The database itself will be using SQLite3. PK stands for Primary Key, and all other rows use standard SQL notation.

- Datatypes
    - INTEGER - Stores any signed integer
    - REAL - Stores any floating point number
    - TEXT - Stores any string

- Column Options
    - NOT NULL - does not allow null values to be inserted
    - AUTOINCREMENT - automatically sets the value to be the previous row's value + 1
    - UNIQUE - does not allow a value to be inserted if that value already exists in the column

Below is an example that shows the entity relationship diagram including the column options data types, and the primary keys. This model organizes the data points that are being related to others in a way that is easier to understand.

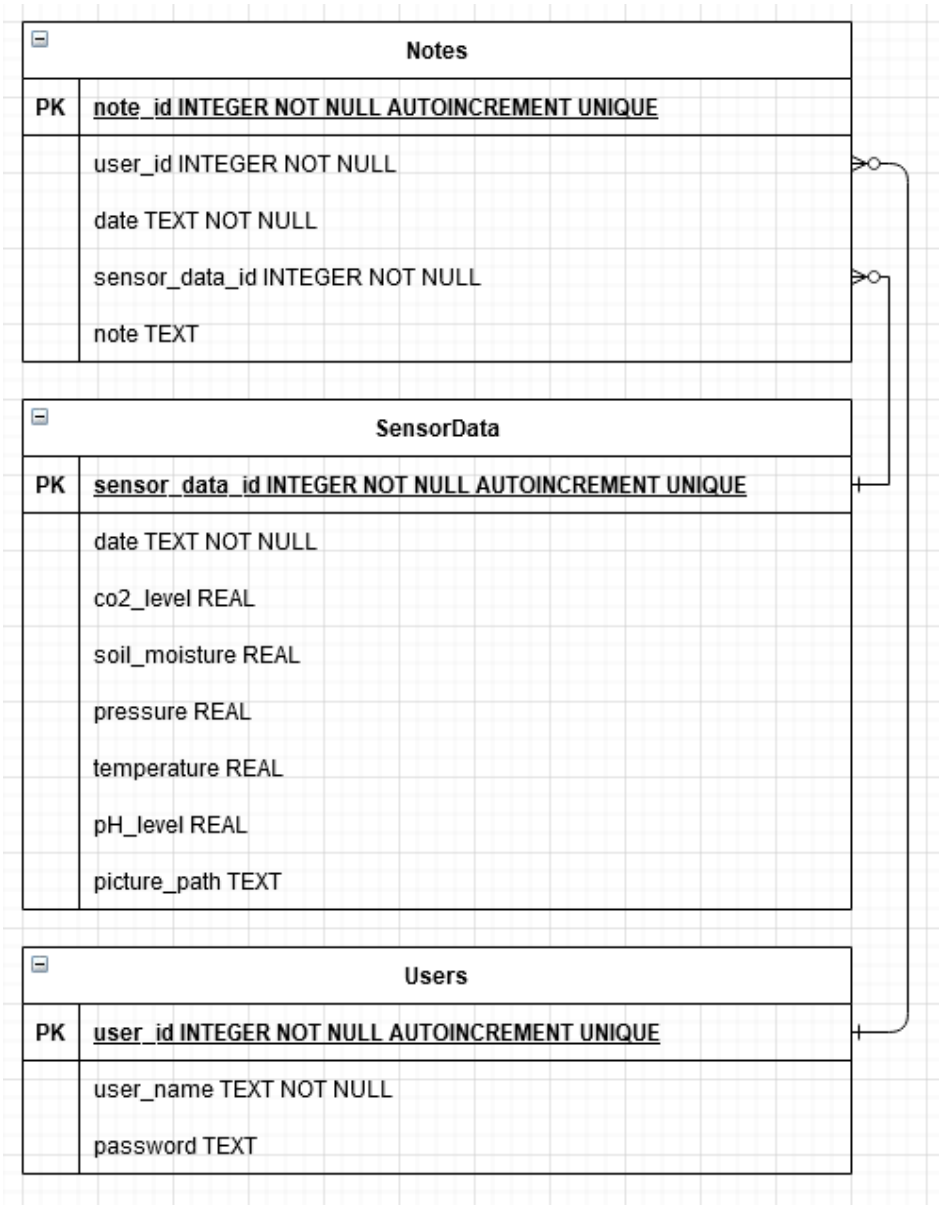


Figure 54: Entity Relationship Diagram (SQLite3)

Below are a few more examples of the actual database that will be used for the GUI. These examples include the users database, the notes database, as well as the sensor data database.

	id	user_name	password
	Filter	Filter	Filter
1	1	exampleUsername	examplePassword
2	2	exampleUsername2	examplePassword2

Figure 55: Example Users

In the figure above, an example of the user has been listed as to how it will look in the database. This will include their id number in the first column, followed by their username that they used to sign up as well as the password after it. This is how all the users that have accounts with the plant habitation system and the GUI will be listed in the database. If a user is not listed in the database, they will not be able to log into the GUI and will have to create/sign up for a new account. Although it is a simple database, the amount of users will be expanding endlessly since the plant habitation system will continue to grow plants in space.

	id	user_id	sensor_data_id	note	date
	Filter	Filter	Filter	Filter	Filter
1	1	1	3	Lorem ipsum dolor sit amet, consectet...	2021-11-17 11:54:32.325
2	2	2	3	Sed ut perspiciatis unde omnis iste ...	2021-14-17 10:32:11.722

Figure 56: Example Notes

In the figure above, an example of the notes database is shown. The notes database will be very crucial to the project since every user will be able to add as many notes, documents, and photographs to the GUI. The database will include all of the user's personal notes, public notes, photographs, and comparison notes when they compare two different dates. This will include the note's id number, the user's id number, the sensor data's id number, then the notes and the date that the notes have been added as well. All of the columns will have the option to filter the

data in case a specific user needs to be looked up or if a user's notes for a specific date needs to be found. The user_id will be the same as the id shown in the previous figures.

	id	date	co2_level	soil_moisture	pressure	temperature	pH_level	picture_path
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	2021-11-12 09:54:32.325	0.53	0.2	33.5	72.34	0.6	~/Documents/ExampleFolder/plantImage2021-11-12 09:54:32.png
2	2	2021-11-13 09:54:32.947	0.51	0.25	33.4	71.7	0.61	~/Documents/ExampleFolder/plantImage2021-11-13 09:54:32.png
3	3	2021-11-14 09:54:32.947	.52	.23	33.4	72.1	.6	~/Documents/ExampleFolder/plantImage2021-11-14 09:54:32.png

Figure 57: Example Sensor Data

In the figure above there is an example of the sensor data listed. The sensor data will include the sensor data's id number, the date of the data, the data, as well as a link or path to the image taken of the plant for that data. The data includes but is not limited to: CO2 levels, soil moisture level, pressure, temperature, and pH levels inside the plant habitation system. The sensor data's id number will be the same as the sensor_data_id shown in the figure before in the previous figure.

7. Project Prototype Testing Plan

The testing plan for this project will be in accordance with NASA System Engineering processes and standards to best efforts. The below figure depicts the methodology for the various testing phases.

7.1 Hardware Test Environment

7.1.1 Senior Design Lab

The University of Central Florida's Department of Electrical and Computer Engineering has provided a testing environment in Room 456, at Engineering 1. This lab provides a workshop space with other instrumentation, equipment, and software that include: [44]

- Tektronix Oscilloscopes
- Tektronix Dual Arbitrary Function Generators
- Tektronix DMM 4050 Digital Multimeters
- Keithley 2230-30-1 Triple-Channel Power Supplies
- Dell Precision 3420 Computers
- SMD Rework Station
- Soldering and Desoldering Stations
- Digital Microscope Inspection Station

7.1.2 TI Innovation Lab

The University of Central Florida and Texas Instruments provide a lab that is equipped with a Universal Laser Cutter that can cut through an array of materials including: wood and acrylic. The lab also features 8 3D printers that print in PLA, ABS, as well as Carbon Fiber [45]. These printers use software that is easy to use and efficient for this project and they also allow for printing in different materials, which will come in handy for some of the parts in the plant habitation system. This is also beneficial since lots of parts are to be in-house made since ordering parts would take up a lot of time and money. Having the lab aids in reducing production costs as well as speeds up manufacturing.

7.1.2.1 Stratasys Fortus 450mc

Fused Deposition Modeling allows for rapid prototyping of parts that assist in the iteration on design over a shorter period of time. It delivers accurate performance that allows users to transform supply chains, speed up manufacturing, and reduce multiple production costs [65]. FDM also comes at a lower cost than traditional subtractive manufacturing over even ordering parts off the shelf at times. It also helps cut on time since processes for ordering parts could be tedious and require third party vendors as well as shipping. Designing, testing, and fabricating parts in-house provides more control over the product. [68]

Figure 58: Stratasys Fortus 450mc

7.1.2.2 Stratasys J55

The Stratasys J55 is a 3D printer that will be used for custom design of hardware mechanical parts to reduce the cost of the design. There are three main components of the printer: design, import, and print. It's able to accept designs using native CAD files or 3MF files formats and using the software GrabCAD Print, it's able to print just about any design that is sent to it [63]. Having a 3D printer that has the option to print in different materials will be very beneficial as the team iterates on the design. The software is GrabCAD Print, which is a cloud-based solution for 3D printing. [64]

Below is a figure portraying the J55 taken at the TI Innovation Lab:

Figure 59: Stratasys J55

7.1.2.3 Delta Makers

The Delta Markers are also 3D printers that are able to print objects up to 22 inches tall. A special function that the delta markers have is the ability for webcam support, which will help for video monitoring if the team is not at the lab watching the printing job. The software that these printers use is the Simplify3D professional-quality printing software. The Delta Makers also use PLA as the type of material. [67]

Below is a figure portraying the Delta Makers taken at the TI Innovation Lab:

Figure 60: Two Delta Makers

7.1.2.4 ILS12.150D

The laser cutter takes an array of materials documented in the figure below above the laser cutter on a white board. The frame of the product will be cut using this laser cutter which provides the fastest way of iteration over design to ensure integration is executed accurately and with the least amount of error. It supports a bed of 48’ x 24” x 12” that meets the size requirements for our product. It also supports the materials needed for this project. [70]

Below is a figure portraying the CO2 Laser Cutter taken at the TI Innovation Lab:

Figure 61: ILS12.150D

7.1.2.5 Tektronix MDO3034

The Tektronix MDO3034 is a Mixed Domain Oscilloscope that would allow for the testing of the voltage and current to be executed accurately and safely. The MDO3034 contains up to 16 logic channels, a 50 MHz arbitrary/function generator, protocol analysis and a spectrum analyzer of up to 3GHz [68]. This machine will allow for the components on board the product to fit within the power constraints as well as maintain safe operations across components.

Below is a figure portraying the Mixed Domain Oscilloscope taken at the TI Innovation Lab:

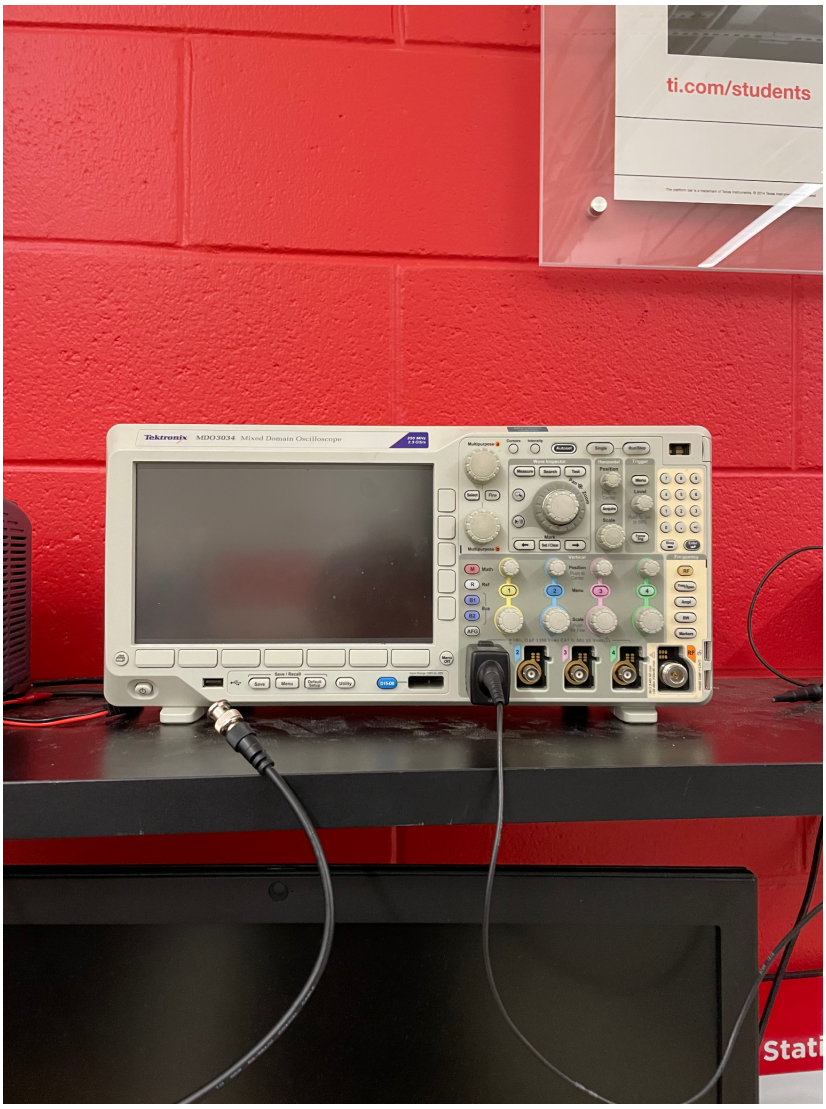


Figure 62: Tektronix MDO3034

7.1.2.6 Tektronix AFG3022C

This function generator allows for the testing of the custom made components for the project that will be developed in house. It provides 2 channels, 25 MHz analog bandwidth, 14 bits of vertical resolution, 25 Mhz of output frequency and 20 Vp-p into 150 ohm loads. These specifications allow for the design to be tested to both NASA and UCF standards. [69]

Below is a figure portraying the Function Generator taken at the TI Innovation Lab:

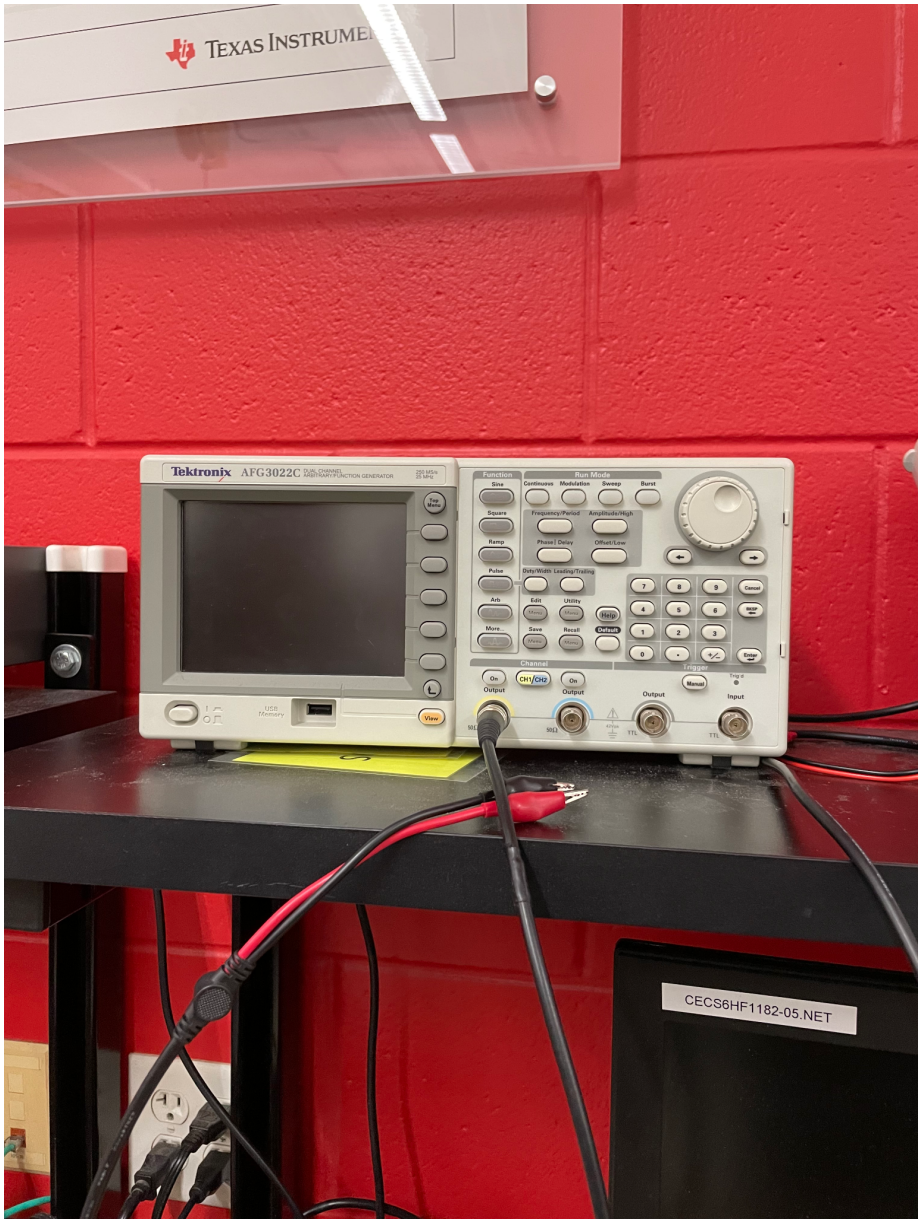


Figure 63: Tektronix AFG3022C

7.2 Hardware Specific Testing

The University of Central Florida has provided testing equipment. This testing equipment will be a key component in testing the design ensuring it meets UCF and NASA standards.

7.2.1 Breadboard

A breadboard provides the ability to test and build circuits rapidly before designing and soldering. The holes in the breadboard provide access to a lower layer that can transmit electrical current, so components such as capacitors and resistors can be connected in series or in parallel.

The following conventions will be used when building and testing on the breadboard:

- Side-line power supply connections for chips instead of direct power supply
- Ground connects to use black wire and red wire for other
- Jumper wires are to be kept on the board flat decreasing clutter
- Jumper wires are to be routed around chips instead of over so changing chips is more efficient
- Component legs are to be trimmed ensuring a tight fit on the breadboard

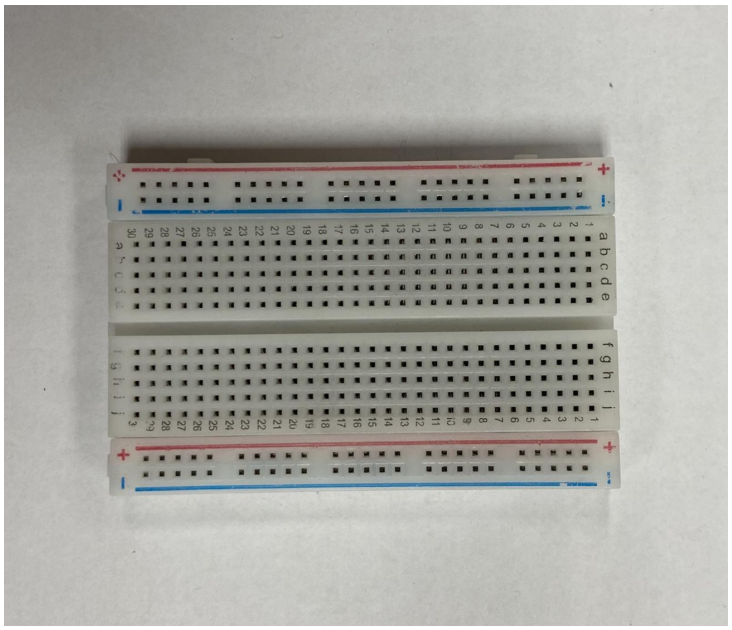


Figure 64: Breadboard

7.2.2 Analog Discovery 2

This multi-function component, that features analog and digital inputs and outputs, contains an array of testing and measurement instruments that include: [46]

- Oscilloscope
- Waveform Generator
- Logic Analyzer
- Protocol Analyzer
- Spectrum Analyzer
- Power Supplies

This list is directly extracted from Digilent Reference without modification since it provides accurate and detailed insight on the capability of Analog Discovery 2 to be set up to work as any of these traditional instruments [46].

- *"Two-channel oscilloscope (1MΩ, ±25V, differential, 14-bit, 100MS/s, 30MHz+ bandwidth - with the Analog Discovery BNC Adapter Board)*
- *Two-channel arbitrary function generator (±5V, 14-bit, 100MS/s, 12MHz+ bandwidth - with the Analog Discovery BNC Adapter Board)*
- *Stereo audio amplifier to drive external headphones or speakers with replicated AWG signals*
- *16-channel digital logic analyzer (3.3V CMOS, 100MS/s)1) 2)*
- *16-channel pattern generator (3.3V CMOS, 100MS/s)3) 4)*
- *16-channel virtual digital I/O including buttons, switches, and LEDs – perfect for logic training applications 5) 6)*
- *Two input/output digital trigger signals for linking multiple instruments (3.3V CMOS)7)*
- *Two programmable power supplies (0…+5V , 0…-5V). The maximum available output current and power depend on the Analog Discovery 2 powering choice:*
- *500mW total when powered through USB*
- *2.1W max for each supply when powered by an auxiliary supply*
- *Two-channel voltmeter (AC, DC, ±25V)*
- *Network analyzer – Bode, Nyquist, Nichols transfer diagrams of a circuit. Range: 1Hz to 10MHz*
- *Spectrum Analyzer – power spectrum and spectral measurements (noise floor, SFDR, SNR, THD, etc.)*
- *Digital Bus Analyzers (SPI, I^2C, UART, Parallel, CAN)"*

Figure 65: Oscilloscope, Logic Analyzer and Variable Power Supply

7.2.3 P6100 BNC Clip Probes Cable

The BNC clip probes connect to the oscilloscope, shown in the previous figure, to a circuit for measuring high-frequency signals. The set of probes shown below includes two BNC cables with hook type oscilloscope probes. The cables are rated at 6 MHz at 1X attenuation and 100 MHz at 10x attenuation[50].

The figure below shows the P6100 BNC clip probes provided in the Analog Discovery kit:

Figure 66: P6100 BNC Clip Probes Cable

7.2.4 BNC to Alligator Clip Probe

This probe serves to connect from the Analog Discovery 2's waveform generator channels with the BNC Adapter. This probe provides a secure connection due to its alligator-style clip at one end for the transmission of signals between the components.

The figure below shows the probe:

Figure 67: BNC to Alligator Clip Probe

7.2.5 Oscilloscope Probe Cables

These probe cables provide a direct and secure connection from the oscilloscope to the circuit under test with minimum load on the circuit. The probe holds signal fidelity for accurate measurements. [52]

Below is a figure depicting the probes:

Figure 68: Oscilloscope Probe Cables

7.2.6 BNC Adapter

The BNC adapter works with the BNC probes shown in the figures above to accurately measure higher voltage signals than what an oscilloscope can usually measure. The probes change the input impedance seen by the oscilloscope in the discovery kit and attenuate the signal by a predetermined factor. The BNC adapter provided attenuates the signal by a factor of ten [51].

Below is a figure of the adapter:

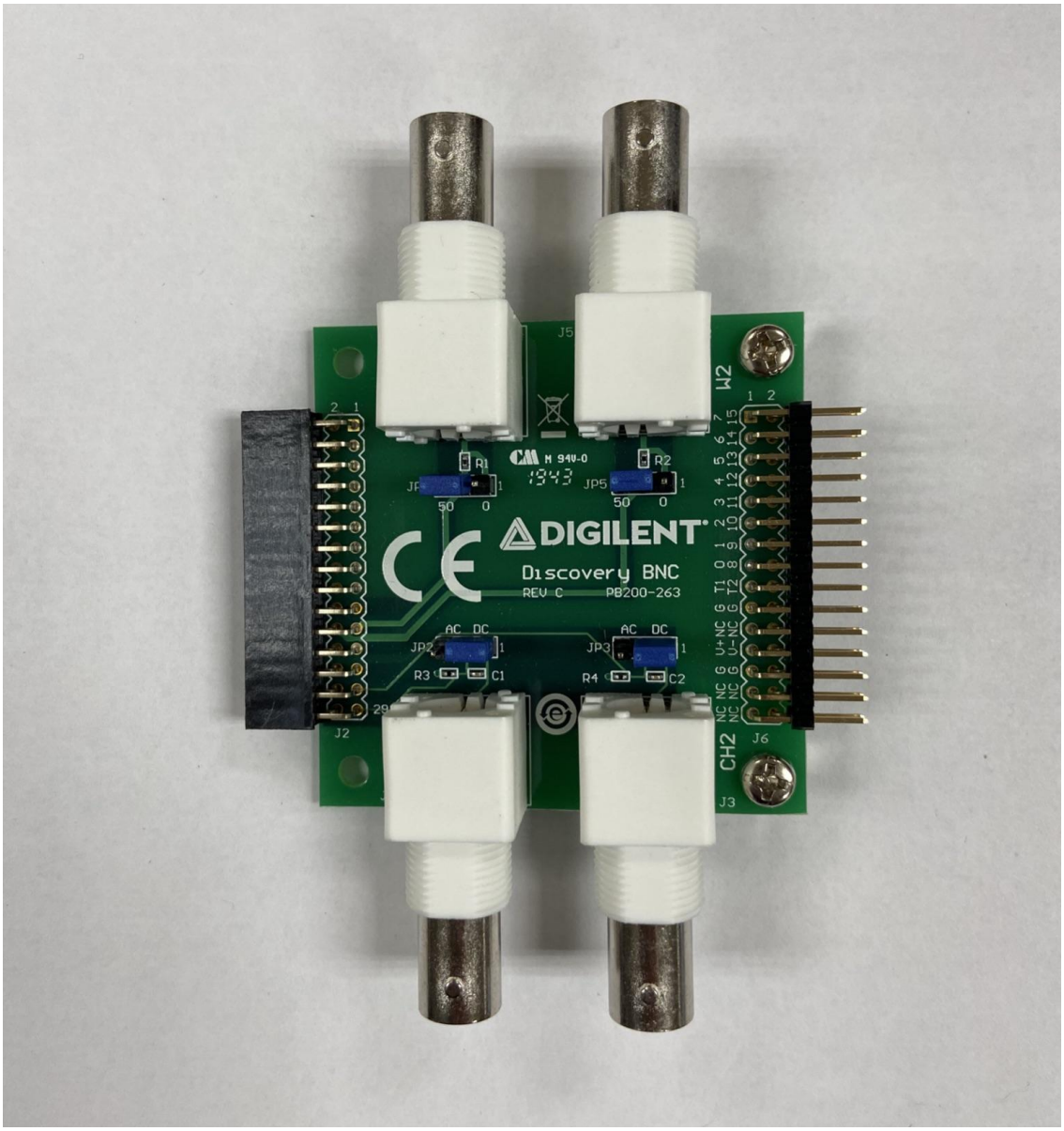


Figure 69: BNC Adapter

7.2.7 Flywires

Signal cables provide a straightforward way to connect the Analog Discovery 2 to the breadboard with color coded wires. The cable features VIO wires in red, GND wires in black and other colored wired for various use cases. [48]

Below is a figure displaying the wires:

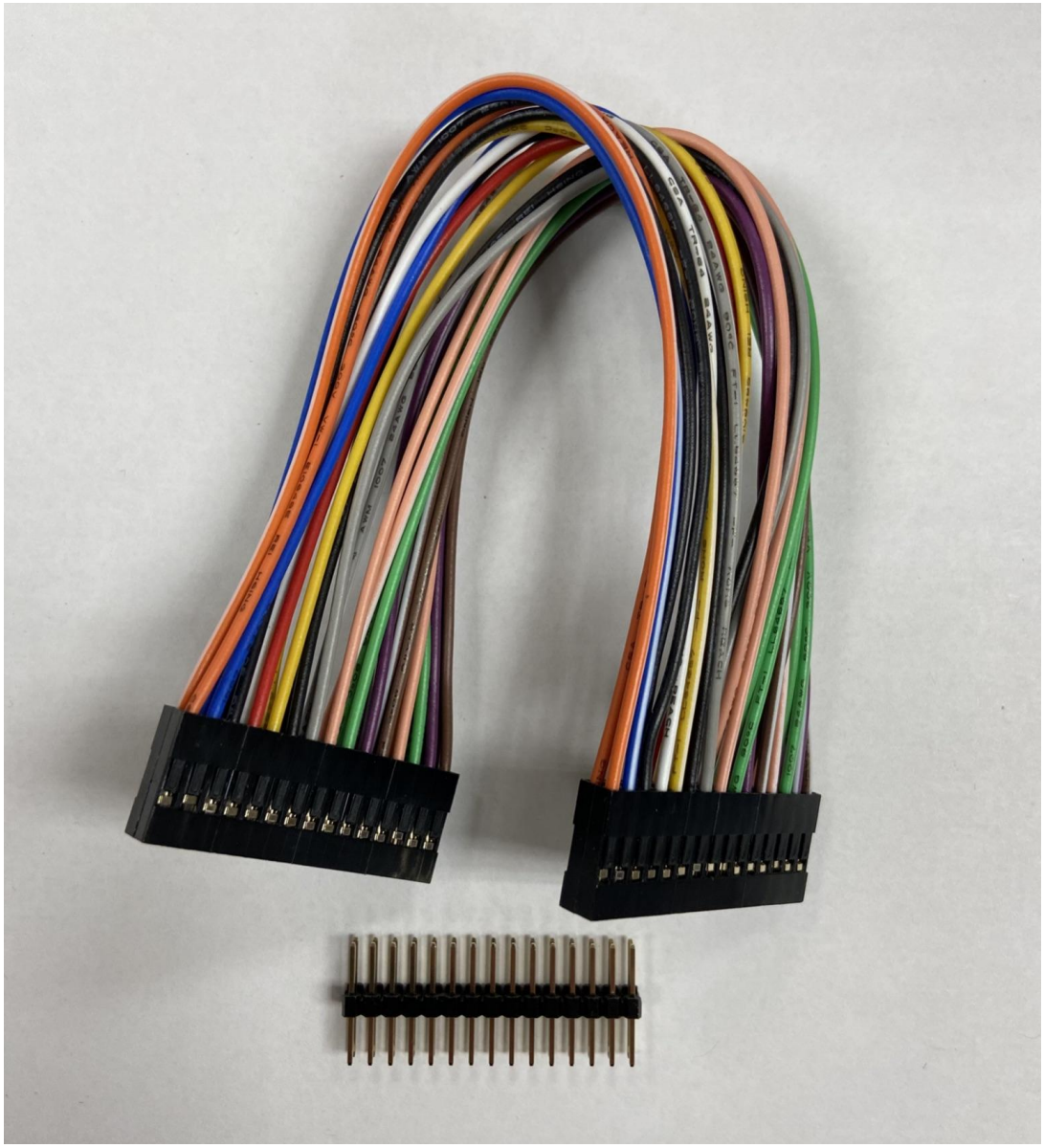


Figure 70: 2x15 Flywires, Signal Cable Assembly for the Analog Discovery

7.2.8 Mini Grabber Test Clips

A secure connection is needed when probing small scale circuits, the mini grabber test clips provide that security. They have two small pincers which allows them to grab components and wires up to 1.27mm in diameter. [49]

The figure below displays the clips:

Figure 71: Mini Grabber Test Clips

7.2.9 Breadboard Breakout

When using a breadboard, connecting all of the inputs and outputs on the analog discovery to the breadboard can be done through the breadboard breakout. This allows for a more efficient transition between circuits without disrupting the circuit. [53]

The below figure shows a photo if the breadboard breakout:

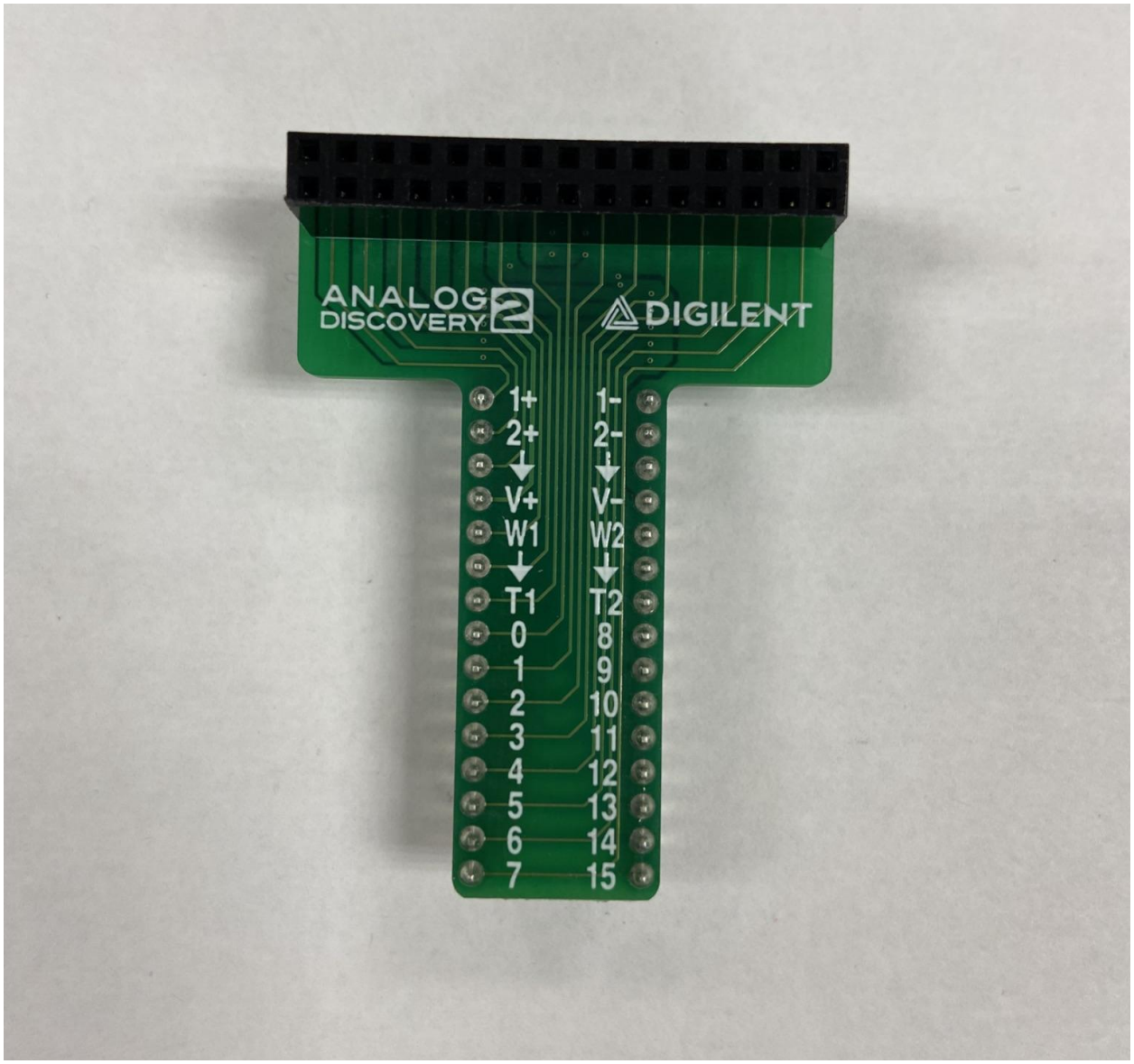


Figure 72: Breadboard Breakout

7.2.A Analog Discovery Impedance Analyzer

The analyzer was designed with automatically adjusting reference resistors and relays to help the user of the analyzer tools in WaveForms [59]. This analyzer is used to analyze inductive elements as well as capacitive. It's able to automatically select the most appropriate configuration for any attachment. It allows for auto-scaling and will select reference resistances automatically.

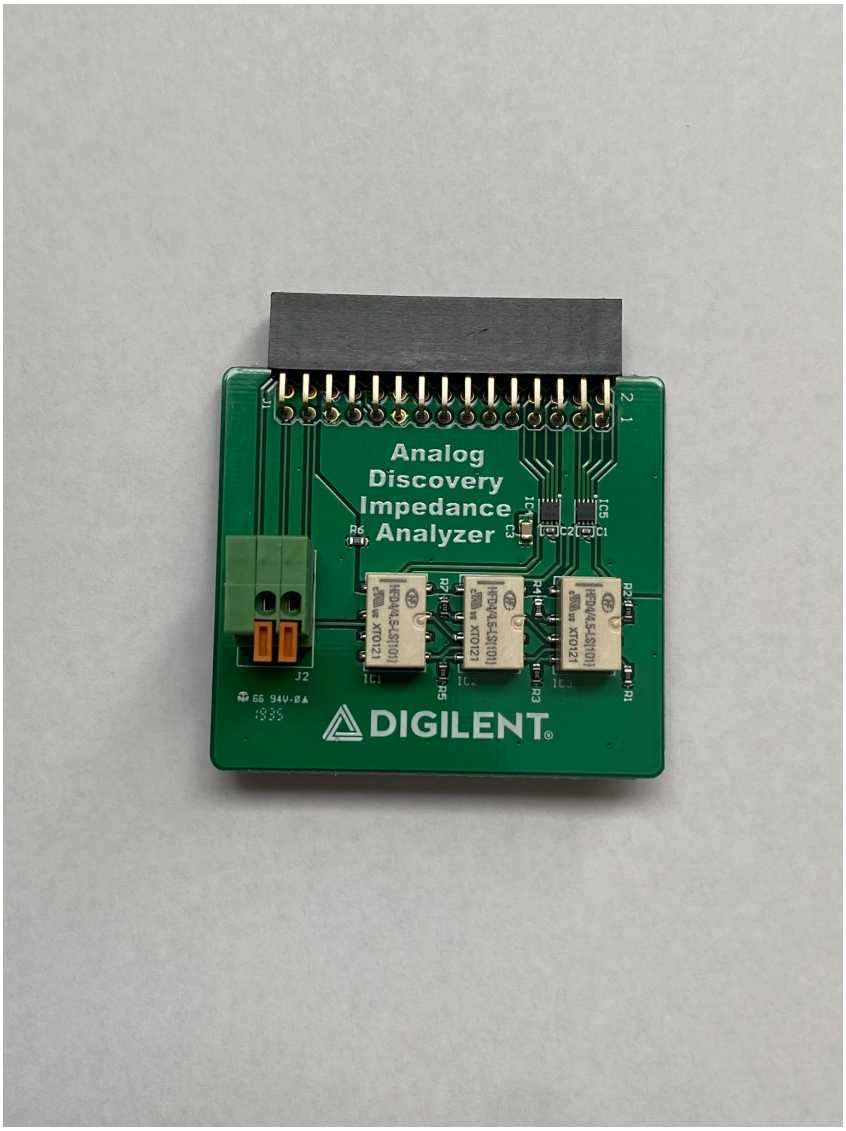


Figure 73: Analog Discovery Impedance Analyzer

7.2.B Breadboard Adapter

The Breadboard Adapter is used to conveniently switch between testing multiple projects without having to unplug each wired connection [61]. It provides a female header so any male can connect to it, and the components and wires can be soldered directly to the surface of the prototype. This will allow a more secure set up.

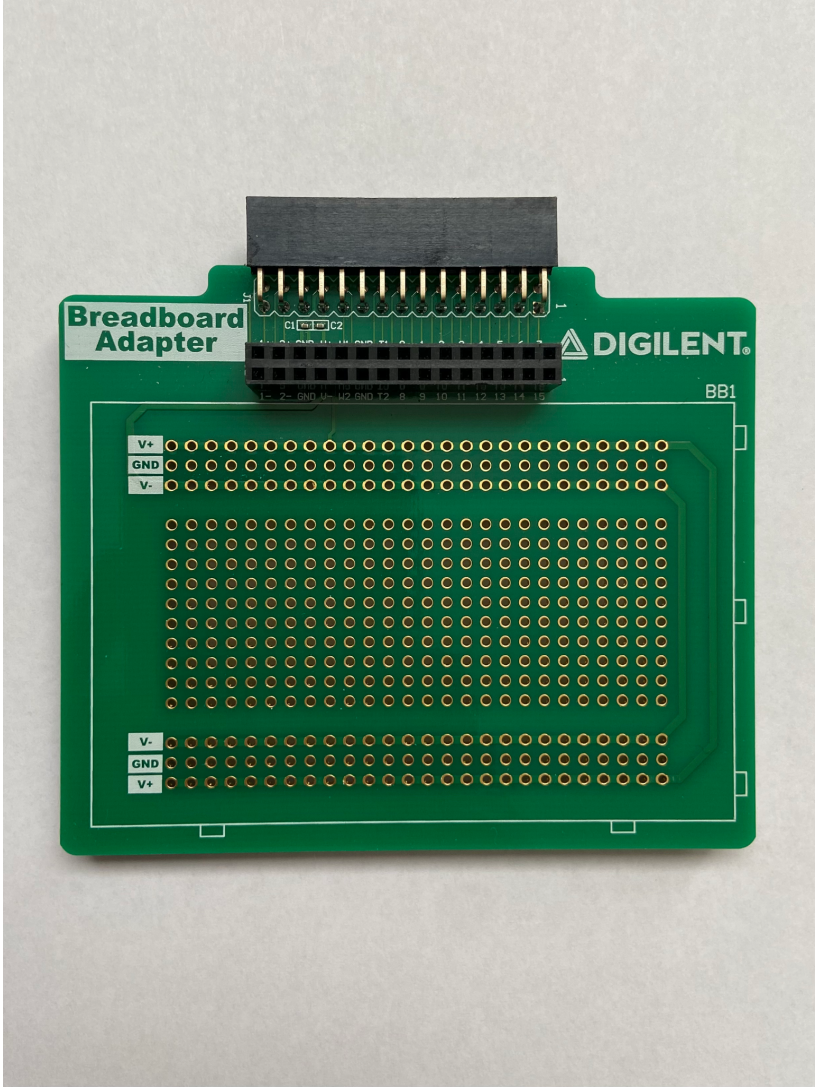


Figure 74: Breadboard Adapter

7.3 Software Test Environment

7.3.1 Machine Learning Test Set

Splitting the dataset into two would give a test set and a training set. They can be any percentage of the original data set, however, usually a higher percentage (>50%) is allocated for the training set to ensure a well-trained model. Creating a test set is important because the model can then be trained on data it has never seen before. It can also evaluate the model's performance from the training set to the test set by using error rates. If the model is going great on the training set but not on the test set then the model is likely overfitted with the training set. It is also possible but highly unlikely that the model is underfitting. This can be easily fixed by increasing the training set size.

The figure below shows the splitting of data into a training set, a testing set and a validation set. The key to creating a useful machine learning model is finding the best balance between the amount of data used for training and testing. This methodology will be abided by when training and testing the machine learning model. [55]

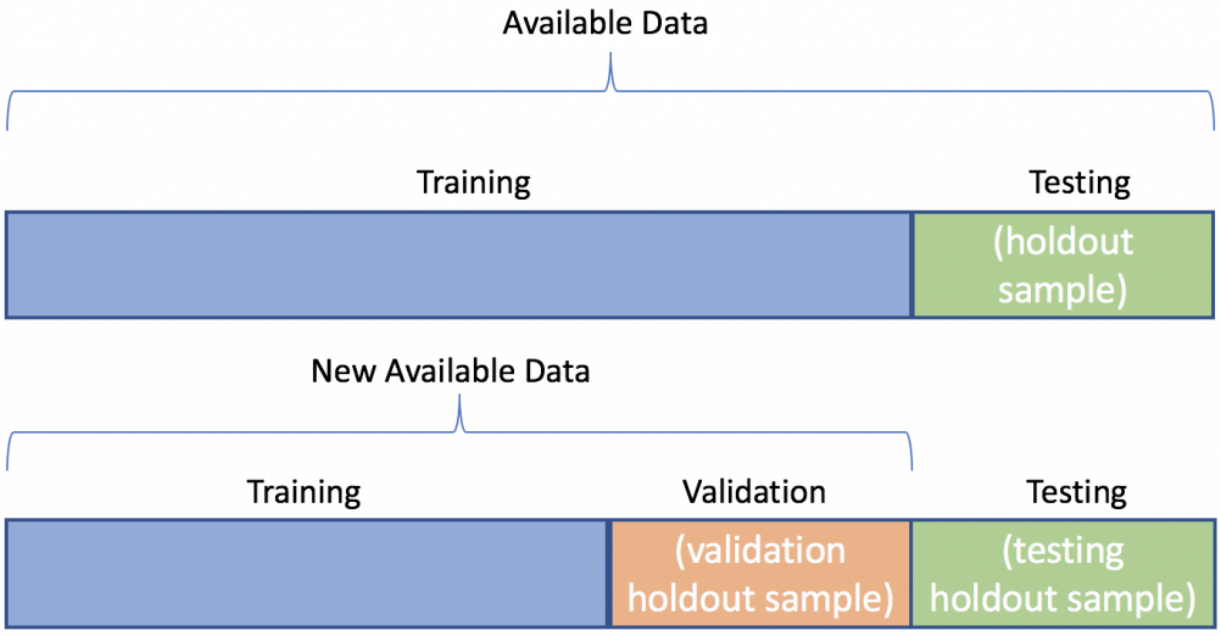


Figure 75: Splitting dataset into sets

7.3.2 Machine Learning Confusion Matrix

A Confusion Matrix is a performance measure for classification and can be outlined as follows:

Actual/ Predicted Class	Positive	Negative
Positive	True Positive	False Negative
Negative	False Positive	True Negative

Table 10: Confusion Matrix

From the Confusion Matrix, the following equations can be obtained for evaluating the model:

- Accuracy (all correct / all) = TP + TN / TP + TN + FP + FN
- Misclassification (all incorrect / all) = FP + FN / TP + TN + FP + FN
- Precision (true positives / predicted positives) = TP / TP + FP
- Sensitivity aka Recall (true positives / all actual positives) = TP / TP + FN
- Specificity (true negatives / all actual negatives) =TN / TN + FP

Finally, it can calculate the F1 Score that helps obtain precision based on mean: F1 = 2*(precision*recall) / (precision + recall) = TP / (TP + ½ (FP + FN).

7.3.3 Loss Function

A loss function helps adjust a model's weights so that they fit the training set without overfitting. The loss function is important because without it the model is highly unlikely to have accurate predictions. The loss function stems from the Euclidean distance described as follows:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Figure 76: Loss Function

where d is the distance, and p and q are the points being measured.

The most popular measures for regression models are RMSE short for Root Mean Squared Error, and MSE short for Mean Squared error, and they are both similar. The equations for performance measures of are the following:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Figure 77: RMSE Equation

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$$

Figure 78: MSE Equation

and where n is the number of data points in the dataset, ŷ are the labels or output and y are the feature vectors. The above figures for the RMSE and MSE equations show the actual representation of the equations.

7.3.4 Gradient Descent

Gradient descent is an optimization algorithm that will be used with the goal of finding a local minimum of a function. The method it uses to do so goes against the direction of the gradient, otherwise it would locate the local maximum instead of becoming gradient ascent. However, Gradient Descent looks for the point on the function with the greatest slope, that's where it then marks as a local minimum. The different types of gradient descents all work very similarly by selecting a data point, passing it through the model, computing the gradient, then updating the weights and again.

Below is an example of gradient descent that depicts the local minimum in multidimensional space with the desired behavior in orange: [54]

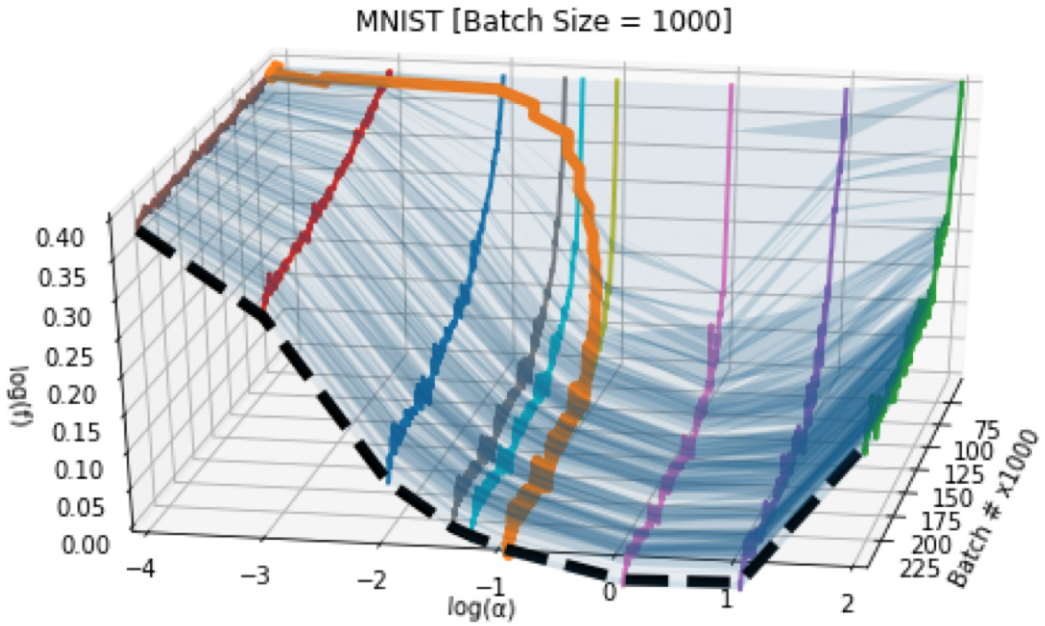


Figure 79: Gradient Descent, Hyper Optimization Surface

7.3.4.1 Batch Gradient Descent

The whole dataset's training data is used at once for every step of the descent. At the end of a step, the parameters of the model are updated using the mean gradient of all instances of the training data. This works well with simple functions and datasets that are not too large. As the number of steps increases, the loss continues to decrease but at a decreasing rate. Unfortunately, with more complex loss functions and larger datasets this approach starts becoming a computational issue.

7.3.4.2 Stochastic & Mini-Batch Gradient Descent

When a dataset is large, then stochastic or mini-batch gradient descent can be used instead. Stochastic gradient descent takes one instance of the data at each step. The gradient of this instance is then used to update the weights of the model. With one instance being considered at a time, the cost will never reach its minimum but converges faster than batch gradient descent with large datasets. Mini-batch gradient descent uses portions of the dataset for every step instead of the whole dataset. The mean gradient of the mini-batch can then be calculated and the weights are updated accordingly.

7.4 Software Specific Development & Testing

7.4.1 ARCC Environment Setup

7.4.1.1 SLURM Components

SLURM the scheduling manager for clusters that monitors resources and jobs. Slurm is composed of nodes where each server node contains a slurmd daemon that provides a fault-tolerant system. [43]

At the top, the controller daemons are shown, which slurmctld manages and monitors resources. The slurmdbd is a database that can be used to record accounting information for an array of clusters in one database. REST API can help communicate with SLURM with an optional slurmrestd daemon.

On the user end, this series of commands is directly extracted from Livermore Computing Center for High Performance Computing that provides an accurate and detailed list: [47]

- *sacct: display accounting data for all jobs and job steps in the Slurm database*
- *sacctmgr: display and modify Slurm account information*
- *salloc: request an interactive job allocation*
- *sattach: attach to a running job step*
- *sbatch: submit a batch script to Slurm*
- *scancel: cancel a job or job step or signal a running job or job step*
- *scontrol: display (and modify when permitted) the status of Slurm entities. Entities include: jobs, job steps, nodes, partitions, reservations, etc.*
- *sdiag: display scheduling statistics and timing parameters*
- *sinfo: display node partition (queue) summary information*
- *sprio: display the factors that comprise a job's scheduling priority*
- *squeue: display the jobs in the scheduling queues, one job per line*
- *sreport: generate canned reports from job accounting data and machine utilization statistics*
- *srun: launch one or more tasks of an application across requested resources*
- *sshare: display the shares and usage for each charge account and user*
- *sstat: display process statistics of a running job step*
- *sview: a graphical tool for displaying jobs, partitions, reservations, and Blue Gene blocks*

Below is a depiction of the different slurm components and their connections, however, the below configuration can be changed depending on the infrastructure:

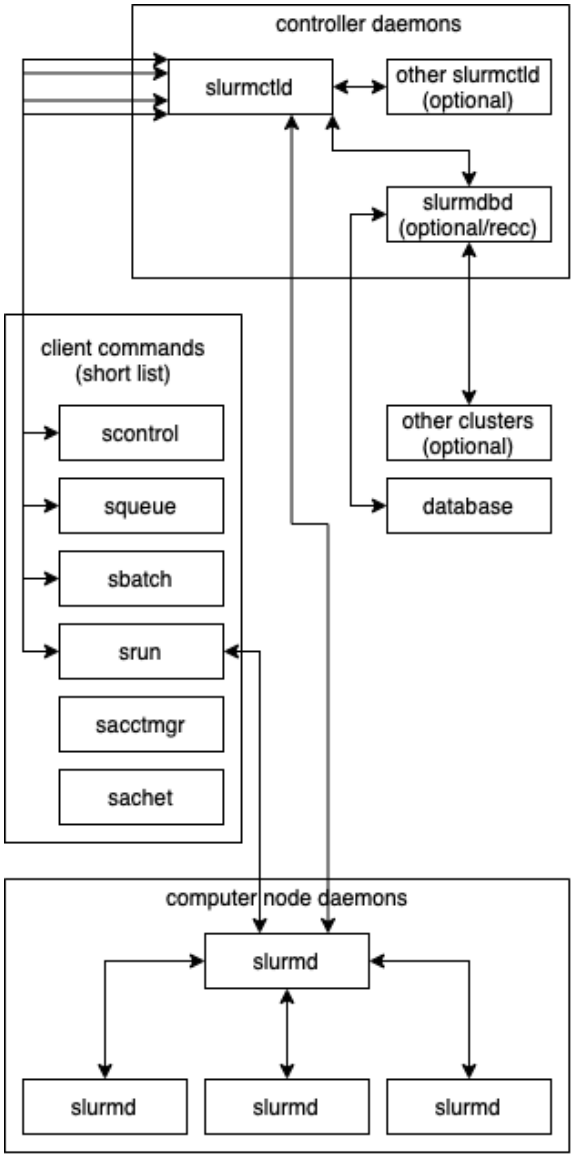


Figure 80: SLURM Components

7.4.1.2 SLURM Entities

Nodes and partitions govern the entities that SLURM daemons or slurmd manage. Those partitions sort nodes logically based on jobs, resource demand, and the steps needed to accomplish a certain job. SLURM draws out conclusions as to what jobs can be performed in parallel and which can be done in series. [43]

Below is a figure that portrays the SLURM entities in relation with partitions, and jobs. Note that this configuration is only a demonstration of the architecture and can be modified to suit every project need.

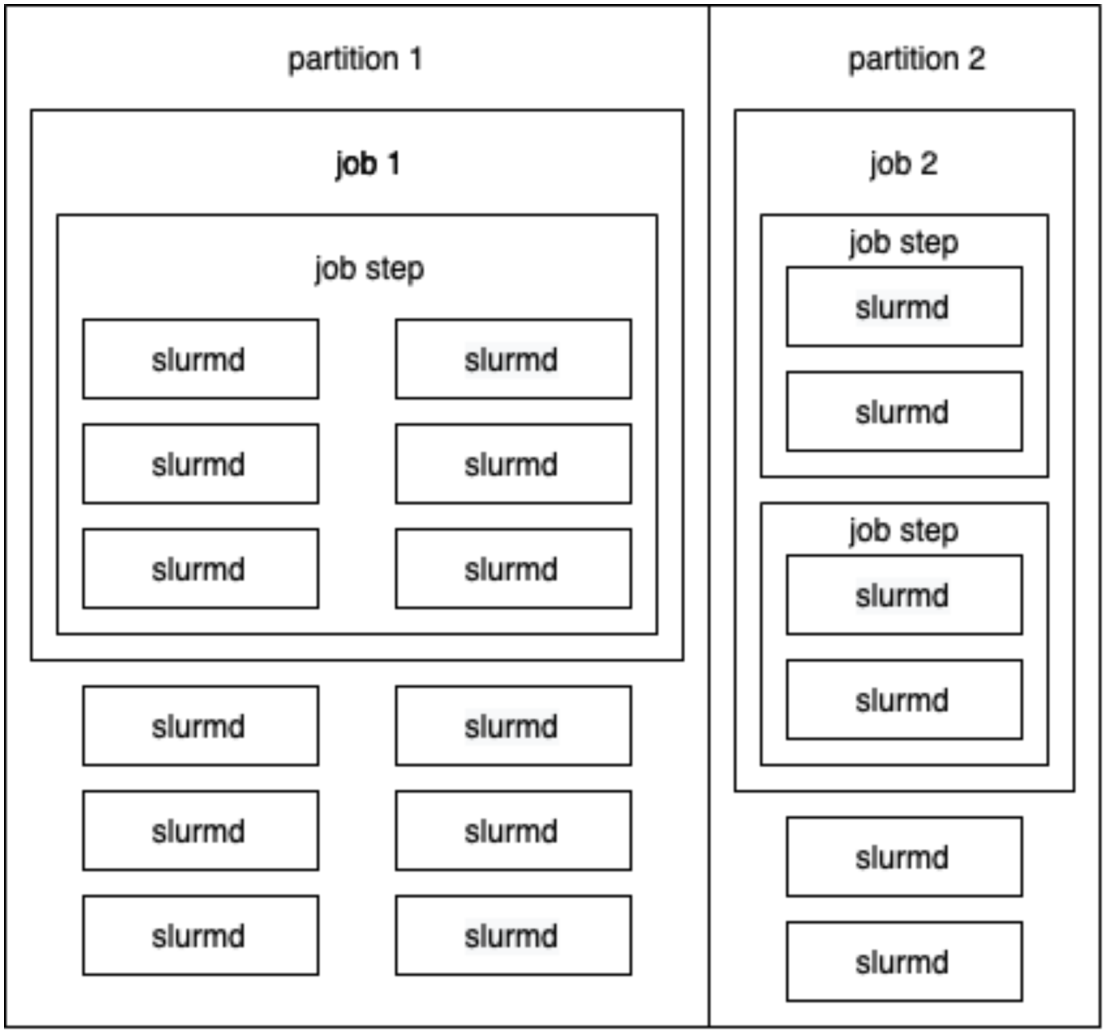


Figure 81: SLURM Entities

8. Administrative Content

8.1 Milestone Discussion

The following section breaks down the milestone requirements in order to complete the project. The milestones have been divided into two sections: Senior Design I and Senior Design II. Senior Design I focused on documentation, research, and purchasing components. Senior Design II will mainly focus on system development, testing, and implementation. Note that the milestones for Senior Design II have been projected, but more will be added and timeframes will be adjusted after Senior Design II begins.

The milestones in this section also follow the documentation provided in NASA Systems Engineering Handbook to best efforts [72]:

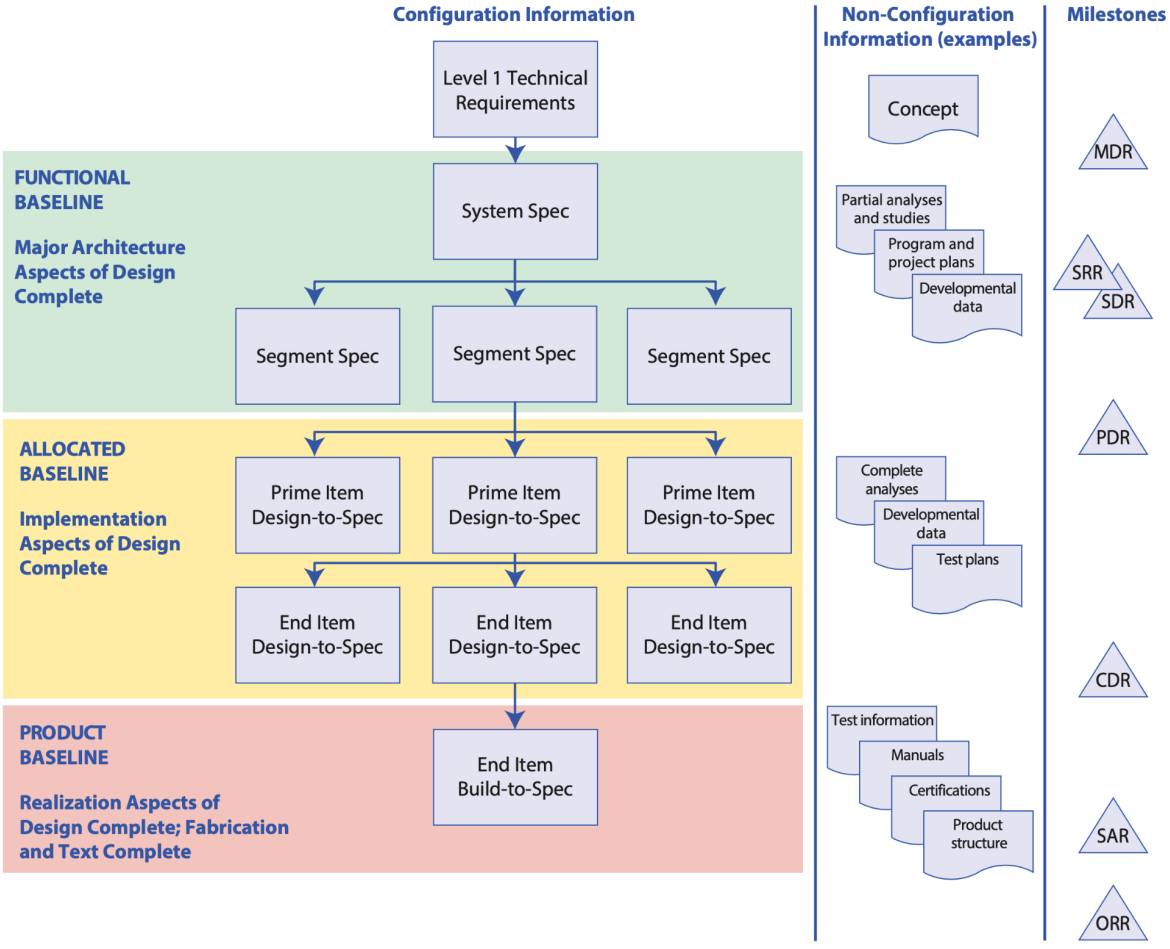


Figure 82: Evolution of Technical Baseline

8.1.1 Senior Design I

Below are the milestones for Senior Design I. By the end of the fall semester all milestones will be completed:

- August 27: Senior design project idea assignment (completed)
- September 10: Boot camp assignment (completed)
- September 16: Project Selection (completed)
- September 17: Divide and Conquer (completed)
- September 22: Bootcamp Team Formation (completed)
- September 23 - September 27: Setting up group meetings and establishing general project understanding (completed)
- September 28: TA Meeting 1 (completed)
- September 28 - October 9: Hardware/software implementation (completed)
- September 30: Initial Design Proposal Due (completed)
- October 1: Divide and Conquer V2 (completed)
- October 1 - October 7: Reworking project proposal, determining timeline, determining research goals and basic resource gathering (completed)
- October 8: NASA Proposal 2 Due (completed)
- October 10 - October 31: Research Python Library for GUI, determine which technologies to use, bridging sensor data collection and GUI (completed)
- October 22: SD Presentation w/ Professor (completed)
- October 23 - October 25: Rework Assignment 4 if needed (completed)
- October 25: Assignment 4 Design Document Due (completed)
- October 27: Quiz 1 Due (completed)
- October 29: Quizzes A-G (completed)
- November 1 - November 29: Start building prototype GUI (in progress)
- November 5: 60 page Draft Documentation (completed)

- November 8 - November 15: TA Meeting 2 (completed)
- November 8 - November 15: Assignment 5 Due (completed)
- November 16 - December 3: Final Design proofreading (completed)
- November 19: 100 page Documentation (completed)
- November 25 - November 28: Testing of prototype (completed)
- November 29: Quiz 2 Due (completed)
- December 6: Final Paper Due (completed)
- December 7: Final Document Due

8.1.2 Senior Design II

Below are the milestones for Senior Design II. General milestones such as testing and implementation will remain, however, some dates are expected to change as more milestones are added to the list:

- January 10 - January 14: Rework GUI, finalize prototype from Winter Break
- January 15 - 17: Mock CSVs for data
- January 17 - January 24: Implement data viewing
- January 25 - January 30: Manipulation into GUI with Mock CSV
- February 2 - February 22: Computer Vision implementation
- February 23 - March 16: Rework code for actual sensors
- March 5 - March 13: Spring Break
- March 17 - April 7: Implement sensor control
- April 11 - April 22: Project alpha testing
- April 8 - April 25: Final Design Project Due
- April 15 - April 27: Final Presentation Review
- TBD: Final Presentation
- May 5: Graduation

8.2 Budget and Finance

Item	Amount	Cost	Total
Jetson Nano	1	$100	$100
MCU	2	$45	$90
Battery	2	$10	$20
CO2 Sensor Module	4	$52	$108
Soil Moisture Sensor	1	$30	$30
Pressure/Temperature Sensor	4	$8	$32
Camera	2	$40	$80
pH Sensor	1	$32	$32
Frame (testing)	3	$50	$150
Frame (prototype)	1	$250	$250
MUX	1	$27	$27
Power PCB/Components	1	$30	$30
LEDs	20	$1	$20
LED Boards	4	$10	$40
Monitor w/ Touch Screen	1	$90	$90
Irrigation System	1	$90	$90
Testing Equipment	3	$77	$231
Safety	3	$60	$180
Total Costs			$1600

Table 11: Budget

The team is still investigating the best parts for the water irrigation system and looking into different materials for the system's frame. The group has been awarded $1600 from the FSGC which will cover all current costs. Any additional costs will be covered by the group if the total were to exceed the allotted budget.

8.3 Conclusion

Throughout this semester, the University of Central Florida Senior Design PlantPod team has worked diligently on this project, making sure to stay on track with the deadlines that were set for the project. The overall goal of the project is to create a plant habitation system with a human-machine interface that is developed to reduce crew time operation. This project is also being used for NASA's Kennedy Space Center Senior Design Program in hopes to show proof of concept of being able to grow and maintain plants in space. Using this type of technology, humanity will be able to explore the solar system and surrounding solar systems without having to depend on Earth-supplied food. This will not only aid humanity to expand but also astronauts to have fresh food, which will benefit their overall health and psychological health.

In the beginning, it was difficult to incorporate different ideas that everyone had into the project such as the cs side; however, after multiple discussions, creating a GUI became the focus. The GUI itself is designed to showcase the plant's health and growth, while also gathering data and storing it into a database for researchers to view. The data is gathered from various sensors inside the plant habitation system, which is a 6U CubeSat volume that houses the plant(s). This plant habitation system is composed of different components that will be either bought or designed and tested in-house.

In addition, it was also challenging to decide on the parts needed for this project that fit within the budget constraints, testing constraints, and time. One of the main obstacles was being able to order the parts in time since a lot of the parts were out of stock. On the hardware side, the assembly and integration had to be iterated over the original design schematics that the team members came up with in order to reach a design that is both functional and minimal.

On the computer vision side, a dataset has been collected and server environments were set up as well as a github repository that would enable easier and more efficient version control. The computer vision will be studied by how the plant does in different conditions. Using the leaves of the plant(s), a model will be generated to depict the right environment in which the plant(s) thrive in. This will be difficult to implement; however, it will become the focus of the project once the plant habitation is successfully built. The machine learning then will be implemented together with the GUI and the database for the GUI to find the perfect conditions for the plant. We hope to achieve images that compare to the figures below:

Figure 83 is an extract from the dataset depicting the segmented leaves [33]:

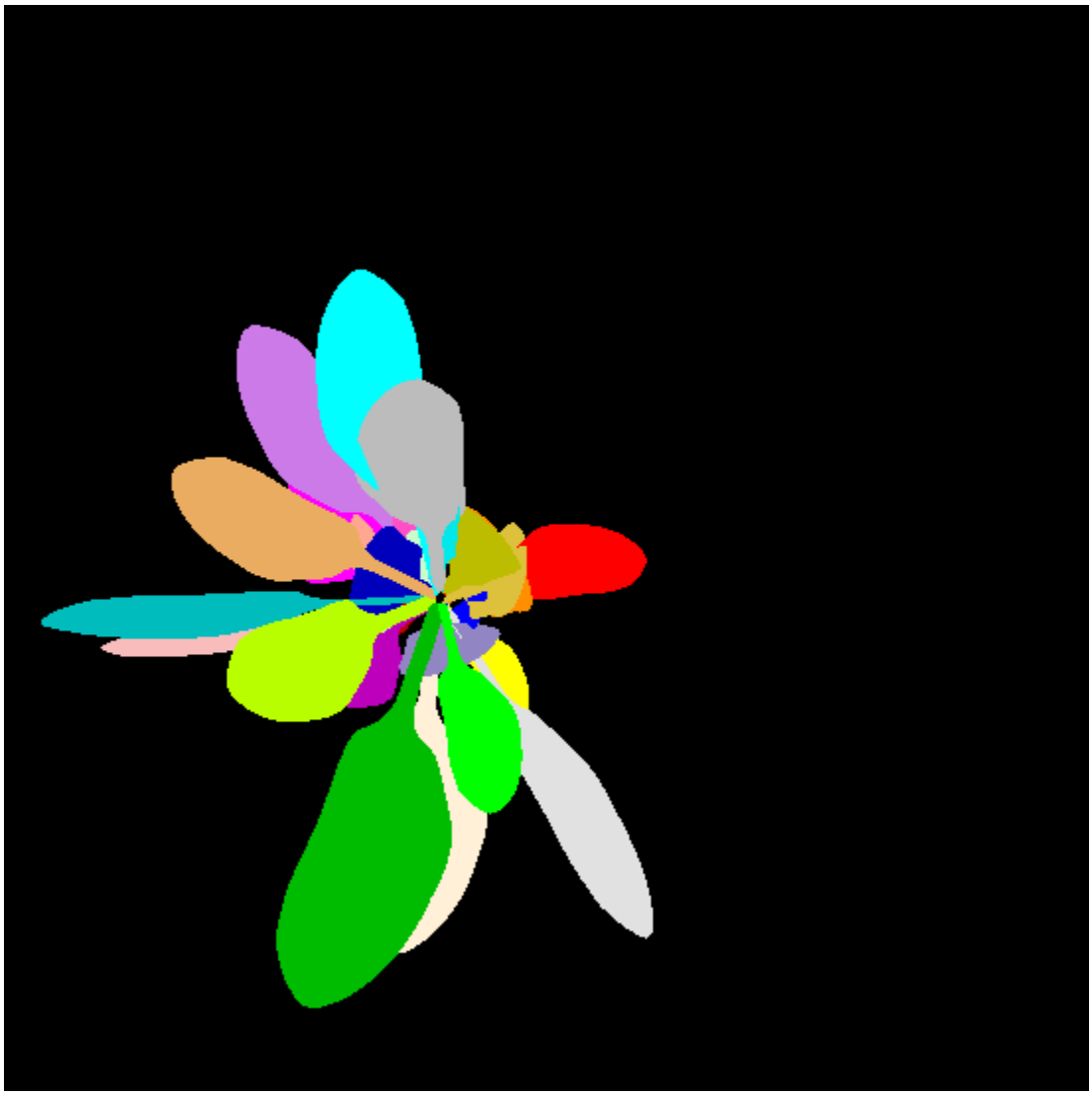


Figure 83: plant00012_label.png

Since technology is advancing faster every year, this project will be worked on for multiple years to come and hopefully improved to the point where plants will be able to grow on their own without any human interference. The PlantPod team hopes that this research will contribute to the technical advancement of the astro botanical field to provide new alternatives for sustainable farming for future generations. The team also hopes that soon enough, humanity will learn how to live in space and expand to different parts of the galaxy.

9. Appendices

9.1 Appendix A - Copyright Permissions

9.1.2 Synthetic Arabidopsis Dataset Copyright Permissions

The metadata and files (if any) are available to the public. They are composed of a series of RGB images and labeled images that segment the leaves. [33]

9.2 Appendix B - Datasheets

9.2.1 NVIDIA Jetson Nano Datasheet

The datasheet for NVIDIA Jetson Nano can be found in Appendix C, URL: [32].

9.2.2 Arduino Datasheet: Schematic

The full size schematic for the Arduino can be found in Appendix C, URL: [34].

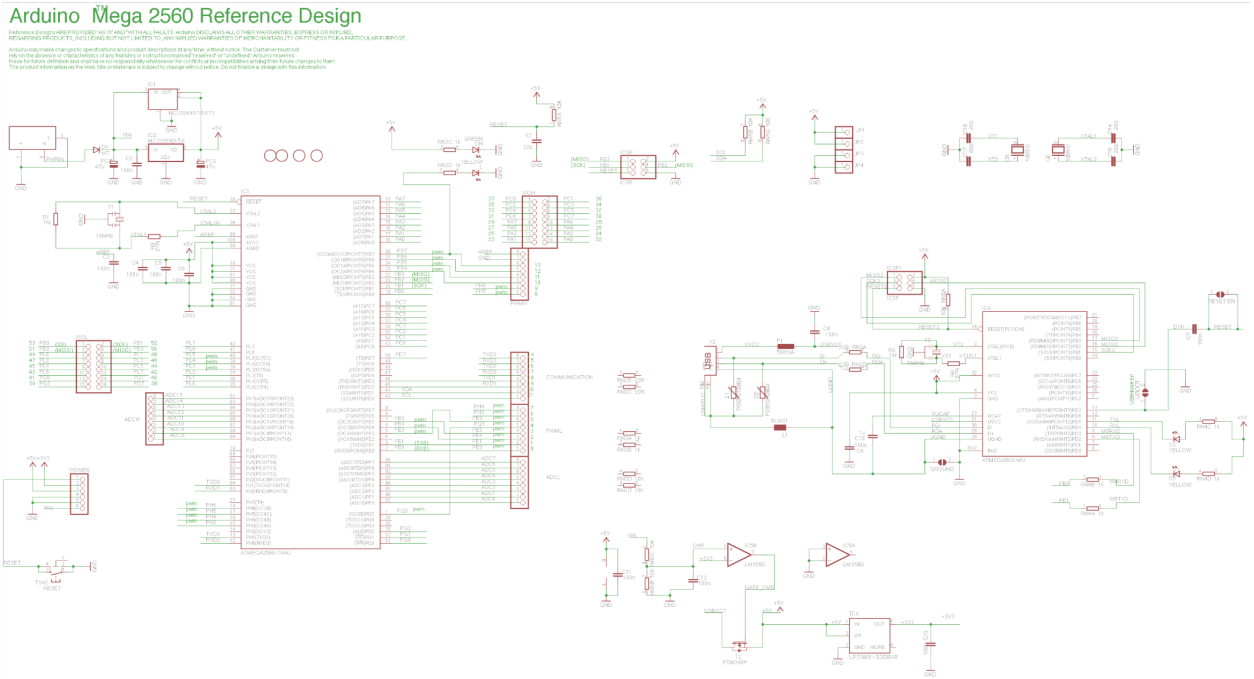


Figure 85: Arduino Mega 2560 Reference Design

9.2.3 Arduino Reference Design

The Arduino Reference Design Zip file can be found in Appendix C, URL: [35].

9.2.4 Atmel ATmega Datasheet

Since the Arduino Mega 2560 is based on the ATmega 4560, the datasheet for that can be found in Appendix C, URL: [36].

9.3 Appendix C - References

[1] “KSC Senior Design Projects.” *Florida Space Grant Consortium*, https://floridaspacegrant.org/program/ksc-senior-design-projects/.

[2] “Page.” *GuiProgramming - Python Wiki*, https://wiki.python.org/moin/GuiProgramming.

[3] “Introduction.” *TkDocs Tutorial - Introduction*, https://tkdocs.com/tutorial/intro.html.

[4] “Learn Python Pyqt.” *Learn Python PyQt | Learn Python PyQt*, https://pythonpyqt.com/.

[5] “Arduino Mega 2560 REV3.” *Arduino Online Shop*, https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us.

[6] *Arduino Playground - Python*, https://playground.arduino.cc/Interfacing/Python/.

[7] Morozova, Elina, and Yaroslav Vasyanin. “International Space Law and Satellite Telecommunications.” *Oxford Research Encyclopedia of Planetary Science*, 23 Dec. 2019, https://oxfordre.com/planetaryscience/view/10.1093/acrefore/9780190647926.001.0001/acrefore-9780190647926-e-75#:~:text=Satellite%20Telecommunications%20and%20the%20Provisions%20of%20International%20Space,Under%20International%20Space%20Law.%20...%20More%20items...%20.

[8] “What Is Gui?: How It Works?: Need & Uses with Examples & Advantages.” *EDUCBA*, 30 Mar. 2021, https://www.educba.com/what-is-gui/.

[9] “Jetson Nano.” *NVIDIA Developer*, 29 Mar. 2021, https://developer.nvidia.com/embedded/jetson-nano.

[10] "What Is a Database?" *Oracle*, https://www.oracle.com/database/what-is-database/.

[11] *About Sqlite*, https://www.sqlite.org/about.html.

[12] *Well-Known Users of SQLite*, https://www.sqlite.org/famous.html.

[13] "IP Ratings." *IEC*, https://www.iec.ch/ip-ratings.

[14] "USB Implementers Forum." *Wikipedia*, Wikimedia Foundation, 1 Oct. 2021, https://en.wikipedia.org/wiki/USB_Implementers_Forum.

[15] "About USB-If." *USB*, https://www.usb.org/about.

[16] "USB: Port Types and Speeds Compared: Tripp Lite." *Tripp Lite Website*, https://www.tripplite.com/products/usb-connectivity-types-standards.

[17] "USB." *Wikipedia*, Wikimedia Foundation, 2 Dec. 2021, https://en.wikipedia.org/wiki/USB.

[18] "Growth Monitor PI: An Open Monitoring System for Plant Science." *Raspberry Pi*, 18 Sept. 2021, https://www.raspberrypi.com/news/growth-monitor-pi-an-open-monitoring-system-for-plant-science/.

[19] Grindstaff, Brandin, et al. "Affordable Remote Monitoring of Plant Growth in Facilities Using Raspberry Pi Computers." *Applications in Plant Sciences*, John Wiley and Sons Inc., 12 Aug. 2019, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6711351/.

[20] *The Origin of Agriculture - Austin Community College District*. https://www.austincc.edu/sziser/Biol%201409/1409lecnotes/LNExamIII/The%20Origin%20of%20Agriculture.pdf.

[21] Tibbitts, T. W., et al. "Growth of Potatoes for CELSS - NASA Technical Reports Server (NTRS)." *NASA*, NASA, https://ntrs.nasa.gov/citations/19950005536.

[22] Dunbar, Brian. “KSC Fact Sheet ‘STS-94/Columbia - Microgravity Science Laboratory-1 (MSL-1) Fact Sheet.’” *NASA*, NASA, https://www.nasa.gov/missions/shuttle/68-97.html.

[23] Hoehn, Alex, et al. “Plant Generic Bioprocessing Apparatus: A Plant Growth Facility For Space Flight BioTechnology Research.” *1996ESASP.390...61H Page 61*, http://adsabs.harvard.edu/full/1996ESASP.390...61H.

[24] *Temperature Sensor Comparison*, https://www.variohm.com/news-media/technical-blog-archive/temperature-sensor-comparison.

[25] Sharma, Vasudha. “Soil Moisture Sensors for Irrigation Scheduling.” *UMN Extension*, https://extension.umn.edu/irrigation/soil-moisture-sensors-irrigation-scheduling#pros%2C-cons-and-costs-of-soil-water-tension-sensors-1751861.

[26] “Ph Sensors for Water and Wastewater Monitoring.” *Sensorex*, 24 Feb. 2020, https://sensorex.com/ph-sensors-3/#process-ph-tab.

[27] *Chapter Contents - Nasa.gov*. https://www.nasa.gov/sites/default/files/atoms/files/9.soa_comm_2021_0.pdf

[28] Regina.montgomery@nist.gov. “NIST Ph Standard Reference Material Supports One of Manufacturing's Most Measured Properties.” *NIST*, 2 Nov. 2021, https://www.nist.gov/news-events/news/2021/10/nist-ph-standard-reference-material-supports-one-manufacturings-most.

[29] Dunbar, Brian. “The Importance of Plants in Space.” *NASA*, NASA, https://www.nasa.gov/audience/forstudents/postsecondary/features/F_Importance_of_Plants_in_Space.html.

[30] *A New Plant Habitat Facility for the ISS*. https://ttu-ir.tdl.org/bitstream/handle/2346/67664/ICES_2016_320.pdf?sequence=1.

[31] "Carbon Dioxide Sensor." *Wikipedia*, Wikimedia Foundation, 3 Oct. 2021, https://en.wikipedia.org/wiki/Carbon_dioxide_sensor.

[32] "Jetson Download Center." *NVIDIA Developer*, 30 Nov. 2021, https://developer.nvidia.com/embedded/downloads.

[33] "Synthetic Arabidopsis Dataset." *CSIRO Data Access Portal*, https://data.csiro.au/collection/csiro:34323v4.

[34] *Arduino mega2560 Schematic*. https://www.arduino.cc/en/uploads/Main/arduino-mega2560-schematic.pdf.

[35] "Arduino Mega 2560 Datasheet - Robotshop." *Vdocument.in*, https://vdocument.in/arduino-mega-2560-datasheet-robotshop.html.

[36] *Atmel ATMEGA640/V-1280/V-1281/V-2560/V-2561/V*. https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf.

[37] *Space Technology - Nasa.gov*. https://www.nasa.gov/sites/default/files/atoms/files/fs-hpsc_fact_sheet.pdf.

[38] "1. Datatypes in SQLite." *Datatypes In SQLite*, https://www.sqlite.org/datatype3.html.

[39] *NASA's Strategic Human Capital Implementation Plan*. https://www.nasa.gov/sites/default/files/atoms/files/o53003551.pdf.

[40] *Veggie Fact Sheet - NASA*. https://www.nasa.gov/sites/default/files/atoms/files/veggie_fact_sheet_508.pdf.

[41] *NASA*. https://www.nasa.gov/sites/default/files/atoms/files/nnj12jd12t_mod_46.pdf.

[42] Industries, Adafruit. “HDMI 7’ 800X480 Display Backpack - with Touchscreen.” *Adafruit Industries Blog RSS*, https://www.adafruit.com/product/2407.

[43] “Slurm Workload Manager.” *Slurm Workload Manager - Overview*, https://slurm.schedmd.com/overview.html.

[44] “Senior Design Laboratory.” *Senior Design Laboratory – Department of ECE*, https://www.ece.ucf.edu/academic-laboratories/senior-design-laboratory/.

[45] “Texas Instruments Innovation Lab.” *TI Innovation Lab – Texas Instruments Innovation Lab*, https://www.cecs.ucf.edu/innovationlab/#capabilities.

[46] K, Sam. “Analog Discovery 2 Reference Manual (Updates under Construction).” *Analog Discovery 2 Reference Manual (UPDATES UNDER CONSTRUCTION) - Digilent Reference*, https://digilent.com/reference/test-and-measurement/analog-discovery-2/reference-manual.

[47] “Slurm Commands.” *Slurm Commands | High Performance Computing*, https://hpc.llnl.gov/banks-jobs/running-jobs/slurm-commands.

[48] Customer, Digilent. “2X15 Flywires: Signal Cable Assembly for the Analog Discovery.” *Digilent*, https://digilent.com/shop/2x15-flywires-signal-cable-assembly-for-the-analog-discovery/.

[49] “Mini Grabber Test Clips with Leads.” *Digilent*, https://digilent.com/shop/mini-grabber-test-clips-with-leads/.

[50] Charles, et al. “BNC Oscilloscope X1/X10 Probes (Pair).” *Digilent*, https://digilent.com/shop/bnc-oscilloscope-x1-x10-probes-pair/.

[51] Colvin, James. “Getting Started with the BNC Adapter.” *Getting Started with the BNC Adapter - Digilent Reference*, https://digilent.com/reference/test-and-measurement/bnc-adapter-board/getting-started-guide.

[52] Martha. “BNC to Alligator Clip Probes.” *BNC to Alligator Clip Probes - Digilent Reference*, https://digilent.com/reference/test-and-measurement/bnc-alligator-clips/start.

[53] Schneider, Lawrance, et al. “Breadboard Breakout for Analog Discovery.” *Digilent*, https://digilent.com/shop/breadboard-breakout-for-analog-discovery/.

[54] *Abstract Arxiv:1909.13371v1 [Cs.LG] 29 Sep 2019*. https://arxiv.org/pdf/1909.13371.pdf.

[55] *Train/Test Split and Cross Validation Training Test Dataset Partitioning and Cross Validation for the Python Tutorial*, https://www.fatalerrors.org/a/0dhy2Dw.html.

[56] *1.264 Lecture 37 - MIT Opencourseware*. https://ocw.mit.edu/courses/civil-and-environmental-engineering/1-264j-database-internet-and-systems-integration-technologies-fall-2013/lecture-notes-exercises/MIT1_264JF13_lect_37.pdf.

[57] *Knowledge Base - UCF IT Support Center*, https://ucf.service-now.com/ucfit?id=kb_article&sys_id=ff89f4764f45e200be64f0318110c763.

[58] *Home*, https://arcc.ist.ucf.edu/.

[59] “Impedance Analyzer for Analog Discovery.” *DigiKey*, https://www.digikey.com/en/product-highlight/d/digilent/impedance-analyzer-for-analog-discovery.

[60] “Breadboard Adapter for Analog Discovery.” *Digilent*, https://digilent.com/shop/breadboard-adapter-for-analog-discovery/.

[61] “Sys - System-Specific Parameters and Functions.” *Sys - System-Specific Parameters and Functions - Python 3.10.0 Documentation*, https://docs.python.org/3/library/sys.html.

[62] “DB Browser for Sqlite.” *DB Browser for SQLite*, 25 July 2021, https://sqlitebrowser.org/.

[63] “Stratasys J55.” *Javelin 3D Solutions*, 4 Oct. 2021, https://www.javelin-tech.com/3d/stratasys-3d-printer/stratasys-j55/.

[64] “Grabcad Print.” *Javelin 3D Solutions*, 28 Apr. 2021, https://www.javelin-tech.com/3d/grabcad-print/.

[65] “Fortus 450MC 3D Printer for High Performance.” *Stratasys*, https://www.stratasys.com/3d-printers/fortus-450mc.

[66] Maker, Delta. “DeltaMaker 2xt: Large Format 3D Printer with 22 Inch Tall Build Volume.” *DeltaMaker 3D Printers*, https://www.deltamaker.com/products/deltamaker-2xt-large-format-3d-printer-with-22-inch-tall-build-volume.

[67] Madaraka Mwema, Frederick, and Esther Akinlabi. “Basics of Fused Deposition Modelling (FDM) - Researchgate.” *ResearchGate*, https://www.researchgate.net/publication/341745627_Basics_of_Fused_Deposition_Modelling_FDM.

[68] *MDO3000 Oscilloscope User Manual - Tektronix*. https://download.tek.com/manual/MDO3000-Oscilloscope-User-Manual.pdf.

[69] *AFG3000 Quick Start User Manual - Tektronix*. https://download.tek.com/manual/AFG3000-Quick-Start-User-Manual_0.pdf.

[70] "Ils12.150D Platform." *ULS*, https://www.ulsinc.com/products/platforms/ils12150d.

[71] Bryan, William. "2020 NASA Technology Taxonomy." *NASA*, NASA, 20 Aug. 2019, https://www.nasa.gov/offices/oct/taxonomy/index.html.

[72] NASA. (n.d.). *NASA Systems Engineering Handbook - NASA Technical Reports Server (NTRS)*. NASA. Retrieved December 7, 2021, from https://ntrs.nasa.gov/citations/20170001761.