

PROJECT FLORA:

Auto-Gardener

UCF Senior Design II

April 26, 2021

Department of Electrical Engineering and Computer Science

University of Central Florida

Dr. Richie

Group 12

Wallace Borges Computer Engineer

Seana Falvey Electrical Engineer

Denver Lau Computer Engineer

Rafael Smith Computer Engineer

TABLE OF CONTENTS

| | |
|--|-----|
| Project Flora: _____ | i |
| TABLE OF CONTENTS _____ | ii |
| LIST OF FIGURES _____ | vii |
| LIST OF TABLES _____ | ix |
| LIST OF EQUATIONS _____ | x |
| 1 Executive Summary _____ | 1 |
| 2 Project Description _____ | 2 |
| 2.1 Motivation _____ | 2 |
| 2.1.1 Existing Projects and Products _____ | 2 |
| 2.1.2 Arduino Garden _____ | 3 |
| 2.1.3 FarmBot _____ | 3 |
| 2.2 Objectives _____ | 5 |
| 2.3 Requirements and Specifications _____ | 6 |
| 2.4 House of Quality _____ | 10 |
| Deliverables _____ | 12 |
| 3 Research Related to Project Description _____ | 13 |
| 3.1 Relevant Technologies _____ | 13 |
| 3.1.1 Digital Image Sensor _____ | 13 |
| 3.1.2 Electric Motor and Control _____ | 14 |
| 3.1.3 Wireless Networking _____ | 14 |
| 3.1.4 Image Recognition and Machine Learning _____ | 14 |

| | |
|--|----|
| 3.1.5 Position Sensors Theory of Operation | 15 |
| 3.1.6 Lithium Battery Cell | 16 |
| 3.1.7 Waterproofing Methods | 16 |
| 3.2 Strategic Components and Parts Selections | 18 |
| 3.2.1 Microcontrollers versus Microprocessors | 18 |
| 3.2.2 Camera Module | 22 |
| 3.2.3 Sensor Module | 24 |
| 3.3.4 Position Sensors | 34 |
| 3.3.5 Motor Module | 36 |
| 3.3.6 Power Module | 41 |
| 3.3.7 Communications Module | 45 |
| 3.3.7 Communications Module | 47 |
| 3.3 Printed Circuit Board | 49 |
| 3.3.1 Terminology | 49 |
| 3.3.2 Composition of a PCB | 50 |
| 3.3.3 Design Recommendations | 50 |
| 3.4 Possible Designs and Related Diagrams | 52 |
| 3.5 Parts Selection Overview | 53 |
| 4 Related Standards and Realistic Design Constraints | 55 |
| 4.1 Related Standards | 55 |
| 4.1.1 Battery Standards | 55 |
| 4.2 PCB Design Standard | 56 |
| 4.2.1 Electrical Equipment Standards | 57 |
| 4.2.2 Software Standard | 58 |

| | |
|--|-------------------------------------|
| 4.2.3 Communication Standard | 59 |
| 4.3 Realistic Design Constraints | 60 |
| 4.3.1 Economic and Time Constraints | 60 |
| 4.3.2 Environmental, Social, and Political constraints | 62 |
| 4.3.3 Ethical, Health, and Safety Constraints | 63 |
| 4.3.4 Manufacturability and Sustainability constraints | 64 |
| 4.3.5 Mechanical Design Constraint | 65 |
| 5 Hardware Design & Software Design | 67 |
| 5.1 Hardware Design Details | 67 |
| 5.1.1 Hardware Block Diagram | 69 |
| 5.1.2 Hardware Design Overview | 67 |
| 5.2 PCB | 69 |
| 5.2.1 Schematic | 70 |
| 5.2.1 Microprocessor | 71 |
| 5.2.2 Ultrasonic Sensors | 71 |
| 5.2.3 Limit Switches | 72 |
| 5.2.4 Moisture Sensor | 72 |
| 5.2.5 Wi-Fi Module | 74 |
| 5.4 Power Module | 75 |
| 5.5 Gantry and Frame Design | 76 |
| 5.5.1 Gantry | 76 |
| 5.5.2 Frame | 77 |
| 5.5.3 Weeding Tool | Error! Bookmark not defined. |
| 5.5.4 Stepper Motor and Gantry Plate Design | 79 |

| | |
|---|-------------------------------------|
| 5.5.5 Waterproof Electronic Enclosure and Water-Resistant Modifications | 79 |
| 5.5.6 The Garden | 81 |
| 5.6 Software Design Details | 82 |
| 5.6.1 Software Architectural Design | 82 |
| 5.7 Microcontroller Software Design | 83 |
| 5.7.1 Microcontroller Algorithms | 83 |
| 5.7 Web Application Design | 86 |
| 5.7.1 User Interface Design | 86 |
| 5.7.2 Database Design | 89 |
| 5.8 Machine Learning Design | 91 |
| 5.8.1 Related/Similar Software Projects | 92 |
| 5.8.2 Machine Learning Algorithms | Error! Bookmark not defined. |
| 6 Prototype Construction and Coding | 101 |
| 6.1 Prototype PCB | 103 |
| 6.2 Final Coding Plan | 105 |
| 7 Project Testing and Prototype Testing | 108 |
| 7.1 Hardware Testing Environment | 108 |
| 7.2 Hardware Specific Testing | 108 |
| 7.2.1 Hardware Testing Overview | 108 |
| 7.2.2 ATmega Testing | 109 |
| 7.2.3 Stepper Motors Testing | 111 |
| 7.2.4 Stepper Motor Resistance Test | 111 |
| 7.2.5 7..2.5 Ultrasonic Position Sensors | 112 |
| 7.2.6 Solenoid Testing | 114 |

| | |
|--|-------------------------------------|
| 7.2.7 Water Resistant Testing _____ | Error! Bookmark not defined. |
| 7.2.8 PCB Prototype Testing _____ | 116 |
| 7.3 Software Test Environment _____ | 116 |
| 7.4 Software Specific Testing _____ | 117 |
| 7.4.1 Microprocessor Testing Procedures: _____ | 117 |
| 7.4.2 Application Testing Procedures: _____ | 120 |
| 7.4.3 Database Testing Procedures: _____ | Error! Bookmark not defined. |
| 8 Project Operation _____ | 121 |
| 8.1 General Information _____ | 121 |
| 8.2 Using the Device _____ | 121 |
| 9 Administrative Content _____ | 123 |
| 9.1 Milestone Discussion _____ | 123 |
| 9.2 Budget and Finance Discussion _____ | 124 |
| 9.3 Project Roles _____ | 125 |
| 9.4 Looking Forward/Conclusion _____ | 126 |
| Appendix A Copyright Permissions _____ | i |
| Appendix B Data Sheets _____ | vi |
| Appendix C Software _____ | vi |
| Appendix D Works Cited _____ | vii |

LIST OF FIGURES

| | |
|---|-------------------------------------|
| FIGURE 1:FARMDUINO CONNECTIONS. REQUESTING REPRINT PERMISSION FROM FARMBOT..... | 4 |
| FIGURE 2: RASPBERRY PI CONNECTIONS OF FARMBOT. REQUESTING REPRINT PERMISSION FROM FARMBOT..... | 5 |
| FIGURE 3: DIAGRAM OVERVIEW OF A POTENTIAL DESIGN OF THE AUTO GARDENER..... | 12 |
| FIGURE 4: IN ORDER FROM TOP TO BOTTOM, GENERIC CAPACITIVE SENSOR, CATNIP ELECTRONICS SENSOR, AND ADAFRUIT CAPACITIVE SENSOR | 27 |
| FIGURE 5: COMPARING CAPACITIVE SOIL MOISTURE SENSORS AND SATURATION | 29 |
| FIGURE 6: EXAMPLE OF RESISTANCE VERSUS ILLUMINATION CURVE FOR A LIGHT DEPENDENT RESISTOR | 30 |
| FIGURE 7 POSSIBLE ATMEGA MICROCONTROLLER DESIGN AND CONNECTIONS | 52 |
| FIGURE 8: BOTTOM VIEW OF AUTO- GARDENER MODULE WITH CAMERA AND WEEDER DRILL. A POSSIBLE DESIGN. | 53 |
| FIGURE 9: HARDWARE BLOCK DIAGRAM..... | 67 |
| FIGURE 10: SCHEMATIC OF THE OVERALL HARDWARE..... | 70 |
| FIGURE 11: BLOCK DIAGRAM OF THE WATER SYSTEM | 73 |
| FIGURE 12:POWER CIRCUIT..... | ERROR! BOOKMARK NOT DEFINED. |
| FIGURE 13: TOP VIEW OF GANTRY, ALONG WITH RAIL CARRIAGES AND CAMERA MODULE CARRIAGE..... | 77 |
| FIGURE 14: A T-SLOT FRAME SLICED TO SHOW CONNECTION AND RAILS ON ALL FOUR SIDES OF THE FRAME. REPRINT PERMISSION REQUESTED BY LANGLE ALUMINUM | 77 |
| FIGURE 15: FRAME DESIGN WITH ALUMINUM T-SLOT TRUSSES. GANTRY IS SEEN MOUNTED ABOVE THE FRAME..... | 78 |
| FIGURE 16: THE WEEDING TOOL IS A GARDEN SPIRAL HOLE DRILL. REPRINT PERMISSION REQUESTED FROM ERGONICA. | ERROR! BOOKMARK NOT DEFINED. |
| FIGURE 17: WEED KILLER MOTOR HOLDER: A POSSIBLE DESIGN. REPRINT CONFIRMED FROM OPEN BUILDS [100]. | ERROR! BOOKMARK NOT DEFINED. |
| FIGURE 18: STEPPER MOTOR HOLDER PLATE DESIGN. REPRINT CONFIRMED FROM OPEN BUILDS [100] | 79 |
| FIGURE 19: COMPONENT HOUSING REPRESENTATION | 79 |
| FIGURE 20 OVERVIEW OF SOFTWARE ARCHITECTURAL DESIGN..... | 82 |
| FIGURE 21: BLOCK DIAGRAM OF SOFTWARE FOR WATER SYSTEM..... | 85 |
| FIGURE 22: BLOCK DIAGRAM OF FUNCTIONS OF CAMERA..... | 85 |
| FIGURE 23 PROTOTYPE OF THE MAIN PAGE IN AUTO GARDENER APPLICATION | 88 |

| | |
|---|-------------------------------------|
| FIGURE 24 ENTITY RELATIONSHIP DIAGRAM OF APPLICATION | 90 |
| FIGURE 25 COMMON CNN STRUCTURE FOR DETECTION OF PESTS..... | 93 |
| FIGURE 26 OPEN SOURCE DEEP LEARNING FRAMEWORKS. REQUESTED PERMISSION FOR PRINTING FROM MING SUN | 95 |
| FIGURE 27: CURRENT PURCHASES THAT WERE ACQUIRED SO FAR | ERROR! BOOKMARK NOT DEFINED. |
| FIGURE 28: PCB LAYOUT..... | 103 |
| FIGURE 29 PCB GERBER FILE | 104 |
| FIGURE 30 AUTO-GARDENER WATERING SOFTWARE FLOW CHART | 107 |
| FIGURE 31: PRELIMINARY TESTING SETUP..... | 110 |
| FIGURE 32:TESTING THE ATMEGA MICROCONTROLLER..... | 110 |
| FIGURE 33: TESTING THE ULTRASONIC POSITION SENSORS ACCURACY | 113 |

LIST OF TABLES

| | |
|---|-----|
| TABLE 1: MARKETING AND ENGINEERING TRADEOFF MATRIX..... | 7 |
| TABLE 2: SYSTEM REQUIREMENTS FOR THE AUTO-GARDENER | 8 |
| TABLE 3: HOUSE OF QUALITY | 11 |
| TABLE 4: KEY OF RELATIONSHIPS, DIRECTION OF IMPROVEMENT, AND CORRELATIONS | 12 |
| TABLE 5: COMPARISON BETWEEN ATMEGA'S [14]..... | 19 |
| TABLE 6: MICROPROCESSOR OPERATING LEVEL COMPARISONS..... | 21 |
| TABLE 7: MICROPROCESSOR ADDITIONAL COMPARISONS | 21 |
| TABLE 8: CAMERA MODULE COMPARISON | 23 |
| TABLE 9: A FURTHER LOOK INTO CAPACITIVE SOIL MOISTURE SENSORS | 29 |
| TABLE 10: PGM20 ELECTRONICS CHARACTERISTICS..... | 31 |
| TABLE 11: COMPARISONS OF MOISTURE SENSORS CHARACTERISTICS | 33 |
| TABLE 12: COMPARING LIGHT SENSOR CHARACTERISTICS..... | 33 |
| TABLE 13: COMPARISONS OF CLASSES OF POSITION SENSORS..... | 35 |
| TABLE 14: COMPARISONS OF CHARACTERISTICS OF LINEAR POSITION SENSORS..... | 35 |
| TABLE 15: COMPARING TYPES OF MOTORS [48]..... | 37 |
| TABLE 16: COMPARING TYPES OF DC TO STEPPER MOTORS | 38 |
| TABLE 17: COMPARISONS OF SELECTED STEPPER MOTOR DRIVERS | 40 |
| TABLE 18: COMPARING POWER SYSTEMS | 44 |
| TABLE 19: UNDERSTANDING THE SUPPLY VOLTAGE AND CURRENT NEEDED..... | 45 |
| TABLE 20: BLUETOOTH VS WI-FI COMPARISON CHART [65] | 46 |
| TABLE 21: WI-FI MODULES | 48 |
| TABLE 22: PARTS SELECTION OVERVIEW | 53 |
| TABLE 23: BILL OF MATERIALS | 101 |
| TABLE 24: ATMEGA328 PRELIMINARY TESTING RESULTS | 110 |
| TABLE 25: AIMED STEPPER MOTOR RESISTANCE READINGS AND RESULTS | 111 |

TABLE 26: TESTING THE ULTRASONIC POSITION SENSORS RESULTS 113

TABLE 27: OVERVIEW OF DATABASE QUERY TESTING **ERROR! BOOKMARK NOT DEFINED.**

TABLE 28: PROJECT MILESTONES 123

LIST OF EQUATIONS

EQUATION 1: DETERMINING X RANGE 16

EQUATION 2: GAMMA CHARACTERISTIC EQUATION 31

EQUATION 3: FORMULA FOR SOLVING FOR THERMAL CONNECTING TRACE WIDTH..... 52

EQUATION 4: FORMULA FOR DESIGN THERMAL RELIEF SIZE 52

1 EXECUTIVE SUMMARY

Climate change has been inevitable throughout these past years with the excessive use of carbon dioxide machines and power consumption. Our use of water is wasted in for production of goods in factories, long showers, watering cars, and namely watering lawns. Watering lawns to keep the grass luscious and green has been shown to wastewater. Floridians are depleting their aquifers for fresh water, and our consumption is faster than the time it takes for aquifers to naturally replenish water. The advantage of our smart garden is that it would consider the soil moisture in real time. It is often that lawns would set a watering schedule, yet it would not consider the time of the year in which it rains constantly. Lawns would be watered while it is raining. How much more clean water can we waste? Instead, this garden will only water when the moisture level is low, and it will use drip and spray irrigation. It will have multiple types of sensors. These techniques will drastically show a difference in the water bill.

Food is grown and shipped from across the world. It could cross thousands of miles using petroleum-based means of transportation. Avocados from Mexico, oranges from California, blueberries from Canada have become the norm. Growing your own food eliminates other petroleum-based products like plastics which come from shipping and the store. It may be more economical to produce foods across the country or internationally, but at the cost of the environment.

Our vision is to create an automated garden which eliminates the arduous work of gardening. The Investing in our project would be beneficial in the long run, as lawn mower companies are pricey. They can cost up to over \$200 per month. Most people do not want to work in the hot sun of Florida repeatedly in their free time. Setting up the garden and planting the young vegetables plants would be the only thing that customers would have to do. The Auto-Gardener would take care the rest of the work. The Auto-Gardener should be able to perform regular maintenance of the garden which includes navigating around the garden area and performing gardening tasks. It will be able to discern a vegetable seedling from weeds and adjust the watering levels according to the type of plant it reads. The gardening tasks may also include planting, applying garden compost, insecticide, and digging if the main components of the project are completed successfully. It will have several sensors including a moisture meter and pH reader. The device should be durable and weatherproof, it will display statistics of area completed, water consumption, and time of operation to a PC using wireless communication.

This document will report the design process. First it will delve into the motivation behind the project and goals. The document will report the specifications and requirements. Afterwards, the report will include research on related topics and how design decisions and part selections were made based on previous work. This document will then cover the constraints and standards of this project that limited the team's decisions. The next two chapter discusses the project's design via the hardware design and the software design. Each chapter goes into more detail and includes block diagrams of both types of designs. Next, the document explores prototype of the project. Any roadblocks that were come across, hardware testing, and software testing are in this chapter. Lastly the administrative content is discussed.

2 PROJECT DESCRIPTION

The motivation and objectives behind this senior design project are one of the first stipulations that are satisfied. FarmBot and Arduino Garden are described in detail in comparison to our own aspirations. The following sections fulfill those stipulations, giving the purpose behind our project and the objectives we want to achieve. Following them are the specifications and engineering and marketing requirements and the house of quality.

2.1 MOTIVATION

For those who cook often at home, it is nice to have herbs on hand, as fresh herbs from the market, such as cilantro, only lasts a few days. However, it is proven to be trickier than expected to grow vegetables and herbs for novice gardeners. It is too easy to over water plants or forget about them. Some people may not want to do the research to understand the multitude of factors that go into gardening such as time of year for planting, watering, soil types, fertilizing schedules, the amount of sunlight, the list goes on. Another issue is that some people may not have the spare time to garden but want to experience eating delicious home-grown food. Some people may not be able to handle the physical labor associated. This project's goal is to eliminate the problematic factors to make gardening novice friendly. In turn, people become empowered to take ownership of where their food comes from and make healthier decisions.

Another benefit to having a vegetable garden is that it is naturally a pesticide control. According to the National Academy of sciences, homeowners use 10 times more pesticides per acre than farmers [1]. Many of these pesticides are fossil fuel derived. These chemicals become runoff into natural water reserves, the ocean, or in the soil. Rather than an entire lawn of the same species, insects would not be able to reproduce in massive quantities because of the various types of plants available. The Auto-Gardener would know what plants were planted and take care of the plants accordingly. If insects were to be found, it could spray neem oil, a naturally insecticide and fungicide pressed from the fruits and seeds of neem, a tree from the Indian subcontinent. Only used in diluted doses, the oil is safe for edible plants.

According to the Environmental Protection Agency, “[on] average, Americans use an average of 30 percent of their water outdoors” [2]. By watering lawns and gardens in a more efficient manner, “Floridian residents could save 46 million gallons of water each day.” This is the equivalent of supplying water to every household in Tampa. Water-efficient irrigation can mitigate this.

A small, yet important, benefit to growing gardens instead of grass lawns is that many fruits and herbs naturally flower, which are beneficial to bees and butterflies.

2.1.1 Existing Projects and Products

There are two projects which are related to Project Flora – the Arduino Garden by a YouTuber and the FarmBot. The Arduino Garden is a simplistic version of what we intend to design. It consists

of simple inputs and outputs. Regardless of it being simple, it provided practical insight by a seasoned engineer. The FarmBot is an intelligent robotic gardener which surpasses our own intentions. It was created with similar aspirations of making gardening accessible and easy. It is a well-designed creation; however, it does not implement the artificial intelligence that the team anticipates utilizing.

2.1.2 Arduino Garden

There are multiple automated gardens posted online through blogs such as Instructables and YouTube similar to our project, but the breadth of these projects does not come nearly in depth or as broad as ours. The Arduino Garden by Practical Engineering has several sensors: a soil moisture sensor, soil temperature sensor, sunlight sensor, and a sensor which measures both air temperature and humidity. Respectively they are the VH300 Soil Moisture Sensor, the THERM200 Soil Temperature Sensor, a CdS Photoresistor, and a DHT22 Digital Temperature and Humidity Sensor. The Arduino Uno checks the water level once a day in the evening and switches the soaker hose or a set amount of time if it is below a certain threshold. Interestingly, they created their own sunlight sensor using a cheap photoresistor which changes the resistance relative to the intensity of light. The housing for the electronics looks like a wooden birdhouse with ventilation.

One of the benefits of this project is that author shares their post-experience as well as technical details about sensors. This is described in detail later in the report. Essentially, they recommend capacitive sensors over resistance sensors. The photoresistor “sunlight sensor” was utilized in conjunction with a voltage divider. Calculate the sensor’s resistance and take into account that the photoresistor’s relationship to illuminance is log to log. [3] If a resistor with a large resistance is applied to the voltage divider circuit, then the sensor will be sensitive to low lightings. It would not be able to tell the difference between a cloudy day and a sunny day. Another consideration by the author is to install the solenoid directly on the hose bib to reduce water hammer when the water is shut off abruptly. The main takeaways for this project are to not connect the garden to a water source, as it may flood the garden, taking notice to the types of sensors used, and that we do not have to overcomplicate the design.

2.1.3 FarmBot

FarmBot is an open-sourced project that exhibits what we are trying to achieve. Although, their products are much more costly, at \$1500 to \$4500. They have two main different designs, the Genesis and Express. They are open sourced so that people like us can take the project to a new scale or to make contributions to the project [4] [5]. Both designs have interchangeable heads which allows it to change the tools needed. They have a seeder, watering nozzle, weeder, soil sensor, and a camera. Having the head as a water nozzle allows it to water each individual plant according to its needs. Having the hardware and software design allows us to have a better idea of how we should implement our goals and specifications. What is useful is the CAD models are available for use. We may use parts of the designs to our advantage, as we do not have previous CAD experience.

The central electronics are the Farmduino and Raspberry Pi. The purpose of the Farmduino is to “provide power and control to the electrical components” [5]. It also controls the stepper drivers and motors. The Raspberry Pi is the microprocessor that “keeps track of the FarmBot’s plants, sequencies, regimens, events, and settings.” The Raspberry Pi also takes photos, monitors, and controls the push buttons and LEDs on the electronics box. The following images, Figure 2 and Figure 3, are taken from FarmBot. Since FarmBot is under the Public Domain, we can reuse the images. These schematics help us understand how our own design needs to have the correct connections/adapters, how to connect the motors, water vacuum pump, and how our own boards will communicate with each other. It will also help us have an idea of how to design our PCB working in conjunction with either an Arduino or Raspberry Pi.

Figure 1: Farmduino Connections. Requesting Reprint Permission from FarmBot.

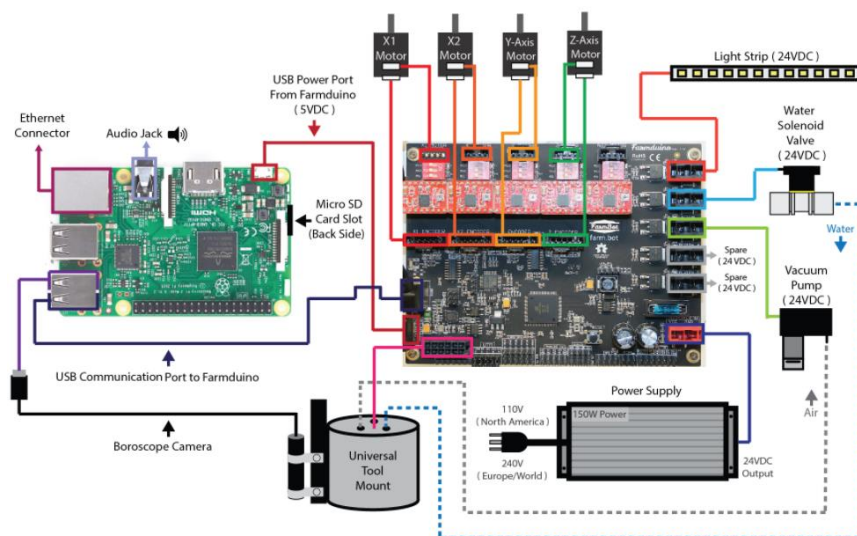
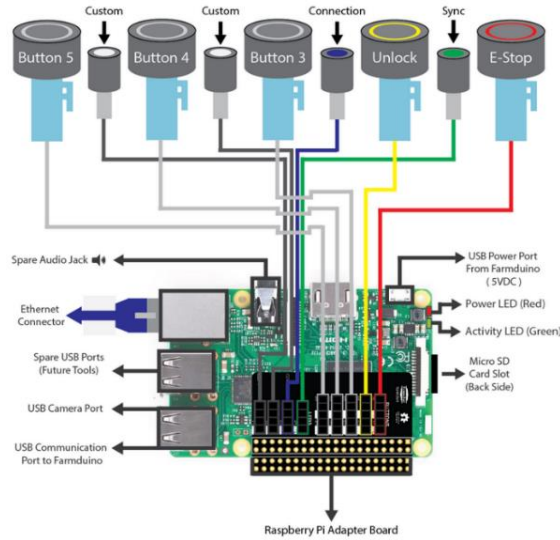


Figure 2: Raspberry Pi Connections of FarmBot. Requesting Reprint Permission from FarmBot.



Our design is enhancing the FarmBot in that we are sourcing cheaper materials. Although their design is exceptional, our design is much smaller, which makes it more practical for city areas which do not have access to lots of land. Our system could possibly be brought indoors or to patios for those who do not have access to backyards. Having our design be smaller and cheaper makes this design available to a larger demographic of people. In addition, we will be adding additional sensors, the soil moisture sensor and light sensor. We do not intend to supersede FarmBot's overall design as their design is more comprehensive and has clearly been developed over the course of multiple years. The overall goal is to have gardening be accessible.

2.2 OBJECTIVES

The main objective is to build and design an automated garden mechanism which performs regular maintenance of a garden that is both functional and technically sophisticated. Our project is divided into different component groups, such as hardware and software, each with a separate role working in conjunction to realize the comprehensive workings of our project. Consequently, each component group has distinct objectives in accordance to its mechanisms. These objectives are necessary to provide clarity in requirements and direction in design.

Hardware

The hardware will consist of the circuit board, the microprocessor, the various sensors, the camera, the Wi-Fi module, a laptop, and the power system.

Software

The software will consist of the different subsystems for the sensors and mechanical controls, including position and water pump control. It will transmit data offsite to a laptop. The offboard program will transmit back to the onboard location information of detected flora and provide desired or not desired plant feedback.

Control

The microprocessor will be the assistant of the system. The primary brain of the system will be a cloud-based AI.

Communications

Wireless communication from the garden to the laptop will be implemented. The form of communication will be through Wi-Fi. Part of the project will be connected to the internet to store and process the information by a laptop. The laptop will be connected to Google Colaboratory program, which may have the database of plant images and statistics.

Power Supply

The goal is to have low power consumption. A potential objective would be to have a solar panel power the system. Since it will be outside, there will be plenty of days of sunshine to charge the solar cells. Perhaps a backup power supply would be added if the solar panel is included.

Realistically, the goal of the project is to have the PCB be battery powered. This is to let the system be modular.

2.3 REQUIREMENTS AND SPECIFICATIONS

In order to build a device that is both functional and technically sophisticated, we need to have certain design and functionality goals we must reach. These goals are reflected as the requirements and specifications we wish to achieve. By satisfying them, we can verify that our project is both functional and technically sophisticated.

The project will have to meet the following specifications in order to ensure that the project is functional. The "system" is a broad term that will be used to describe the project in its entirety, including individual subsystems.

- The system should detect 70% of all weeds in an image.

- The system shall contain a gantry, power adapter, motors, and sensors.
- The system should be able to retrieve user data and correctly output the soil moisture and temperature.
- The system must be able to turn on the watering system with manual control and automatic control.
- The system should collect data in one centralized receiver and display the data.
- The camera is able to take pictures and store them in its standard format.
- The system will not turn on watering system with no water stored in the tank.
 - The system will be able to detect a minimum volume of water of 0.5L left in the tank.
- It will sense the position within 1-inch accuracy.
- The system should be able to display 256 pixels minimum.
- The system should be able to convert moisture levels to a digital readout of moisture within an accuracy of 15%.
- The gantry should be able to move with a precision of at least 1 inch.
- The cost for building the system should be no more than \$650.
- The system will operate from at most a 24V source and at least a 3.3V source.
- The Auto-Gardener should be able to operate in the temperature range of 40°F to 100°F
- The electrical components must be water resistant and be able to operate after it has encounters water.
- The system should not have dimensions larger than 6 ft long by 4ft tall by 4 ft wide.
- The team will make sure that none of the designs are infringing material or patents.

Table 1: Marketing and Engineering Tradeoff Matrix

| | | Ease of Use | Performance | Affordability | Durability | Maintenance | Target |
|------------|---|-------------|-------------|---------------|------------|-------------|---------------|
| | | + | + | - | + | - | |
| Efficiency | + | ↓ | ↑↑ | ↓ | ↑ | ↑↑ | >75% |
| Durability | + | ↑ | ↑↑ | ↓↓ | n/a | ↑↑ | 40°F - 100°F. |
| Precision | + | ↑↑ | ↑↑ | ↓ | ↓ | ↑ | >1 inch. |
| Dimensions | - | ↓ | ↓ | ↓ | n/a | ↑ | <6'x4'x4' |

| | | | | | | | |
|-------------|---|----|----|---|---|---|--------|
| Cost | - | ↑↑ | ↑↑ | ⇓ | ↑ | ↓ | <\$650 |
| Engineering | | | | | | | |
| Marketing | | | | | | | |

- ↑ = Positive correlation
- ⇓ = Strong negative correlation
- ↑↑ = Strong positive correlation
- + = Increases the requirements
- ↓ = Negative correlation
- = Decreases the requirements

The Engineering-Marketing Tradeoff Matrix shows that each requirement is linked with a negative or positive symbol. This means that increasing or decreasing the requirement correlates to how it will increase or decrease the product's suitability. Some entries do not have a correlation, as indicated by the n/a.

Table 2: System Requirements for the Auto-Gardener

| Marketing Requirements | Engineering Requirements | Justification |
|------------------------|---|--|
| 1,3,5 | 1. The system should not have dimensions larger than 6 ft long by 4ft tall by 4 ft wide. | Ensuring a smaller size will reduce price, cost of materials and voltage required to operate. This will also provide extra portability for ease of transfer. |
| 3,5 | 2. The system will operate from at most a 24V source and at least a 3.3V source. | Limiting the system to a voltage range will ensure lower energy costs. Parts that handle lower voltages are easier to replace. |
| 2 | 3. The system should be able to convert moisture levels to a digital readout of moisture within an accuracy of 15%. | Within an accuracy of 15% we can ensure the plants are not oversaturated with water. |
| 1,4,5 | 4. The electrical components must be water resistant and | Waterproofing and resisting will provide a feature of durability to enable the device to be easily operated in many more |

| | | |
|------|--|---|
| | be able to operate after it has encounters water. | conditions and prevent accidents maintaining the device. |
| 4 | 5. The Auto-Gardener should be able to operate in the temperature range of 40°F to 100°F. | This operating temperature range will provide the device durability to function outdoors with varying temperatures. |
| 1, 5 | 6. The system must be able to turn on the watering system with manual control and automatic. | The device will be capable of turning on the watering system easily with two different methods. This will make it easier to test and maintain since alternatives exist. |
| 2 | 7. The gantry should be able to move with a precision of at least 1 inch. | The device must be accurate to perform weeding and take photos of a selected plant. |
| 1 | 8. The system should collect data in one centralized receiver and display the data. | With on central receiver, the communication process is made simpler to implement and use. |
| 2 | 9. The system should detect 70% of all weeds in an image. | This is based on the variability between the data of weeds we train and the actual camera photos taken. |

Marketing Requirements

1. The device must be easy to use.
2. The device must be accurate.
3. The product must be affordable.
4. The device must be durable.
5. The device must be low maintenance.

The marketing requirements are short, as the team does not have an official sponsor. This project is self-funded and is a capstone project to our senior year. The marketing requirements were selected based off of basics of what potential customers would seek in a gardener. The engineering requirements were chosen based on the important functions, the operating environment, and simple specifications.

Engineering requirements 1 and 2 address the marketing requirement of how easy it is to use the Auto-Gardener. Engineering requirements 3, 7, and 9 address how accurate the system will be. Engineering requirements 1 and 2 address how affordable the gardener will

be. Engineering requirements 4 and 5 is correlated to the durability of the product. Engineering requirement 1, 2, 4, and 6 address how low maintenance the product will be. The justifications are explained in the farthest right column of Table 2.

2.4 HOUSE OF QUALITY

When in the product development process, a prominent tool used is the House of Quality matrix. This tool portrays the customer's wants and needs, translating them into a written plan that gets executed through technical requirements. Table 3 shows the legend for the parameters of the House of Quality, while table 4 shows the House of Quality for this project.

In Table 3, customer requirements present how the qualitative properties of our project can affect the desirability of our product to the consumer and how it could affect the functional requirements. The elements chosen are those that most affect the function of the project and determine how desirable it might be to a consumer. Portability being a function of both size and weight also includes possible features such as handle grips or wire tie-downs. Complexity is the inverse of how easily the device can be used, whether that be the physical device, or the software used to manage it and is especially important to non-technical consumers. Performance and efficiency properties demonstrate the trend in consumer products to perform not only better than previous generations but also to use less power as well.

The functional requirements listed in the table are demonstration of the quantitative properties needed for the project. These properties are taken from the engineer's perspective while the project is being worked on. Power usage is a measure of how many watts the device uses to accomplish its given tasks. How much water the device uses can be a measure of not only how efficient it is, but also how much area or size it can cover. Since the project will depend on robotic vision to identify desired plants from weeds, the resolution of the camera will have to meet a minimum requirement while also being adjusted to minimize costs. Motor precision and speed are important properties to keep track of, since the motor's precision will determine how large or small of an area can be tracked by the camera platform. The motor's speed will then determine how effective the camera platform is at scanning the entire planting bed. The device will also track properties of the planting area such as soil moisture and sunlight, higher quality parts would allow for better data but might increase the cost of the project beyond the budget. Processing power refers to the capacity of the onboard microcontroller to handle tasks, a more powerful microcontroller means more calculations could be handled but would also increase the size

and cost of the device. speed at which the onboard microcontroller can send and receive information becomes a priority, a better chip would increase speed but also increase the cost, size and power usage of the device. Floral recognition accuracy refers to the off-device image recognition software, this property is primarily constrained by the time it takes to train a machine learning algorithm and the dataset that can be secured to train it. Size can a property that directly impacts multiple others and is of chief concern during the project. The larger the physical properties are the stronger the motors must be, and the increased power usage required to power the motors, in addition to affecting the customer requirements.

Table 3: House of Quality

| Relative Weight | Customer Importance | Customer Requirements | Functional Requirements | | | | | | | | | |
|-----------------|---------------------|-----------------------|--------------------------|-------------|-------------------|-----------------|-----------------|-------------|------------------|------------------------|--------------------|------|
| | | | Direction of Improvement | | | | | | | | | |
| | | | □ | ▼ | □ | ▲ | □ | ▼ | ▲ | ▲ | ▲ | □ |
| | | | Power Usage | Water Usage | Camera Resolution | Motor Precision | Sensor Accuracy | Motor Speed | Processing Power | Transmit/Receive Speed | Floral ID Accuracy | Size |
| 5% | 2 | Cost | ↑↑ | ↑ | ↑↑ | ↑↑ | ↑↑ | ↑↑ | ↑ | ↑ | | ↑↑ |
| 20% | 9 | Portability | ↓ | ↓ | ↓ | ↑↑ | | | ↓ | | ↑↑ | ↓↓ |
| 5% | 2 | Maintenance | ↑ | ↑ | ↓ | ↓↓ | ↑ | ↓ | ↑ | ↓ | ↓↓ | ↑ |
| 11% | 5 | Complexity | ↑ | ↑ | | ↑↑ | ↑↑ | ↓ | ↑↑ | | ↑↑ | ↑↑ |
| 7% | 3 | Size | ↑↑ | ↑↑ | | ↓ | ↓ | ↓ | ↑ | | ↓ | |
| 2% | 1 | Weight | ↑↑ | | | | | ↓ | | | | |
| 9% | 4 | Appearance | | | | | | | | | | ↑ |
| 18% | 8 | Convenience | ↓ | ↓ | ↑↑ | ↑↑ | ↑↑ | ↑↑ | ↑↑ | ↑↑ | ↑↑ | ↓ |
| 2% | 1 | Performance | ↑ | | ↑ | ↑↑ | ↑ | ↑ | ↑↑ | ↑↑ | ↑↑ | ↓ |
| 20% | 9 | Efficiency | ↓ | ↑ | ↑ | ↑ | ↑↑ | ↓ | ↑↑ | ↑↑ | ↑ | ↑↑ |

| Relationships | | Correlations | | Direction of Improvement | |
|-----------------|----|--------------|---|--------------------------|---|
| Strong Positive | ⇧⇧ | None | | Maximize | ▲ |
| Positive | ↑ | Positive | + | Target | □ |
| Negative | ↓ | Negative | - | Minimize | ▼ |
| Strong Negative | ⇩⇩ | | | | |

Table 4: Key of Relationships, Direction of Improvement, and Correlations

Deliverables

1. The Auto-Gardener must be able to water the plants automatically without the need of a user.
2. The Auto-Gardener must be able to take pictures and send them to an external device.
3. The Auto-Gardener must be able to discern any foreign objects (weeds for example) from the vegetables planted.

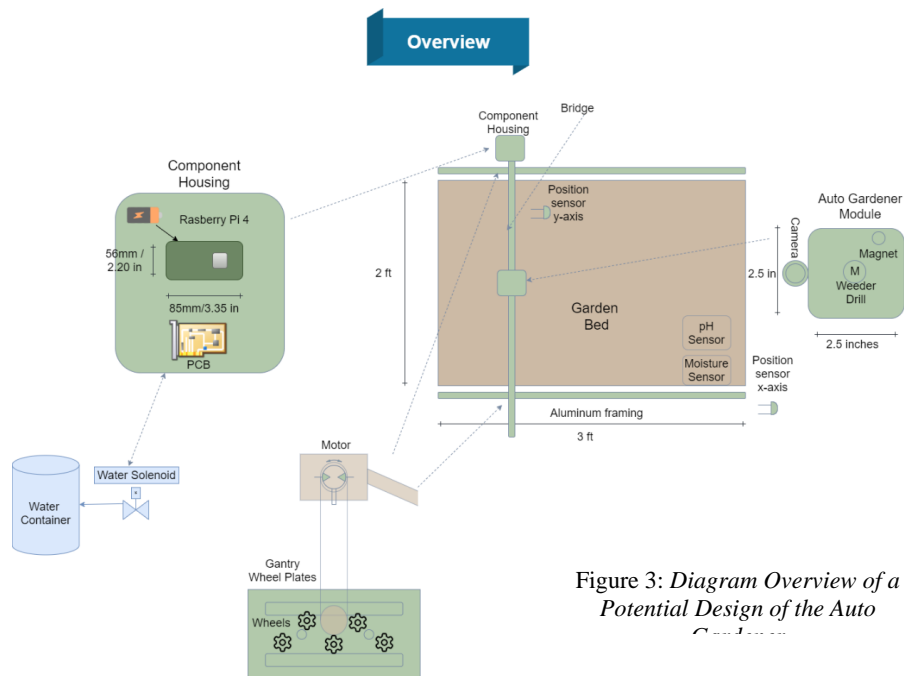


Figure 3: Diagram Overview of a Potential Design of the Auto Gardener

3 RESEARCH RELATED TO PROJECT DESCRIPTION

This chapter describes in detail of any projects and research related to Auto-Gardening. Chapter 3 initially examines any projects that are similar to Project Flora. Next, the chapter discusses any relevant technologies that will be implemented in the project. Relevant technologies section is mainly about the theory of operation of the technology implemented in the project. Technology investigation into specific components and parts are discussed thereafter. This section is very detailed as the team had numerous choices to decide from. How printed circuit boards are physically produced and designed are discussed next. This naturally progresses into section 3.6: Possible Designs and Related Diagrams after all the research has been made. The last section, 3.7, sums up the decisions made from the previous sections through diagrams and a table listing part selections.

3.1 RELEVANT TECHNOLOGIES

This section describes the theory of operation for the technology that may or may not be used in the design. How the digital image sensors, the electric motor and control, wireless networking, image recognition and machine learning, lithium battery cell operate are all described. Lastly, various waterproofing methods are discussed.

3.1.1 Digital Image Sensor

One of the key hardware components of our project is the ability to capture electromagnetic radiation and turn it into electrical signals. For our design we are mainly concerned about the part of the electromagnetic spectrum known as visible light. This light will strike a panel either a: Charged Coupled Device (CCD) or a CMOS image sensor. This panel is typically a combination of both a color mosaic filter and a brightness sensor. An image processor combines the signals from the filter and the sensor to generate an image that can be used by other programs and viewed on a screen to a user.

Digital cameras of varying levels of complexity can be found in:

- Security systems
- Motion sensors
- Day/night detectors
- Digital Photography
- Weather satellites

3.1.2 Electric Motor and Control

In order to have a single camera unit observe the entire planting area, that camera must have the ability to physically move from location to location without human intervention. For this project, an electric motor and accompanying H-bridge or full bridge will be used to convert electrical energy into mechanical energy. The motor operates using a property of electromagnetism known as Lorentz force law, which states that any current carrying conductor inside a magnetic field will experience a force. There are five main components of an electric motor: the rotor, stator, windings and commutator. The rotor is the moving part which turns the electrical energy into mechanical. The stator is the stationary housing around the rotor and contains windings or permanent magnets to generate a magnetic field. The commutator is the mechanism by which the direction of current within the rotor is changed. The motor is designed so a spinning rotor. The spinning rotor would then be connected to a gear and track system which would move the camera unit along a single axis.

The project will use an DC to AC converter which contains an H-bridge to alternate the current that powers the motors used for movement. The H-bridge allows the electronic control over the direction of current through a wire rather than the mechanical control like in a brushed motor. It does this via a combination of MOSFETs which are shown in figure # as open or closed switches, by controlling which gates are closed the direction the current travels through the motor can be determined.

3.1.3 Wireless Networking

Wireless networking is the use of high frequency radio waves to communicate between electronic devices, and the internet. The core of the network is a router, or hub, that manages several things: which frequency to communicate over either 2.4Ghz or 5Ghz, which communication protocol to use and how to manage traffic between multiple devices. The primary use for wireless communication in this project is to offload computationally heavy tasks, such as image recognition, to a more capable device and allow the user to access pertinent information from multiple locations.

3.1.4 Image Recognition and Machine Learning

Image recognition is the process by which a computer program takes a digital photo, or video frame, and by running it through several filtering techniques can guess what the photo contains. By filtering an image for a specific property, such as edge detection, more information can be generated that then can be used to help identify any object within.

The decision-making process is handled by a method of programming known as machine learning, in which the exact patterns or features of a photo that help determine object recognition are determined by the program itself that it learned via training. Within the program are several patterns which can be as small as 2 pixels by 2 pixels that look for a

match, the number of matches produces a score. This score is then weighted against other pattern filters, the combination of all gives the program as to the probability a detected object might be something it was trained on. The complexity of this can increase as the image might be tested through different filters that change it and then ran through the pattern matching layers.

Machine learning training is the process by which the program is given a set of known pictures so that it can calibrate its internal values to get the correct answers. Starting with a randomized set of weights for each of its filters and patterns the program will continually cycle through a given dataset, changing the weights each time depending on how often it was correct in determining which objects were within the picture.

3.1.5 Position Sensors Theory of Operation

3D printers know their home position through various combinations of switches, sensors, or through the code. This technology will be applied to our project so that the system will know the position of the camera.

Microswitch

Microswitches uses a physical a physical interaction to determine the position. It would be placed at the end of the x-axis and y-axis. A piece of metal is attached to the switch, and it senses presence or absence of a physical contact. It is essentially an on and off switch. “It is designed for repeatability, and extended life” [6].

Optical Sensor

Optical sensors are designed using either of two mechanisms. One works by transmitting a light. It receives that light at the other end. The sensor tracks one of the following for changes- intensity, wavelength, polarization, or phase. Another way optical sensors are designed is by reflecting the transmitted light onto an object. The light that is reflected towards the source is examined. These sensors may be more reliable [7].

Hall Effect Sensor

The Hall Effect states “that when a current carrying conductor is placed in a magnetic field, a voltage will be generated perpendicular to the direction of the field and the flow of current” [7]. When there is a magnetic field, there is a potential difference across the output terminals, called the Hall voltage. If the input current is constant, then the Hall voltage is directly proportional to the magnetic field. Hall effect sensors use the hall effect to their advantage. The moving part would be connected with a magnet. The sensor will have a hall element and a magnet. So, when the moving part, such as the camera in our case, will have a magnet. Camera magnet will be a function of the magnetic field of the Hall effect sensor and the Hall voltage. The potential difference outputted by the hall effect sensor

increases in proximity to the camera magnet and decreases when far away, allowing us to determine the position of the camera relative to the sensor.

Position “Sensor” by Code

There is a way to know the bounds of the frame by measuring the amperage of the motors when they stop. It steps the motor forward until it utilizes a lot of power until it knows the edge of the rail. Hence, it would know where it is based on the bounds of the machine. An example of equation used would be in the code would be:

$$\text{scale } X = \frac{x_{\max} - x_{\min}}{\text{input range}}$$

Equation 1: Determining x range

3.1.6 Lithium Battery Cell

Lithium batteries are either disposable or rechargeable. There are two kinds of rechargeable lithium batteries: lithium-ion and lithium-polymer. Batteries are fixed at a specific voltage due to its internal chemistry. The chemical energy is converted to electrical energy. A positive electrode is placed on one end and a negative electrode is placed on the other. In between the two is an electrolyte [8]. Usually the positive electrode is made from lithium iron phosphate, and the negative electrode is made from graphite. When the battery is discharging, the lithium ions move through the electrolyte, into the positive electrode. The electrons are flowing in the opposite directions of the ions. What makes a battery rechargeable is the reversible chemistry. The ions move in the opposite direction, to rejoin with the lithium-cobalt oxide.

For batteries, what needs to be considered is the low-voltage cutoff point. This is the threshold that needs to be addressed; if the battery discharges below the cutoff point, the battery will start to degrade. It will no longer operate as well [9]. We will most likely use a lithium battery to power the PCB or the development board as we are prototyping.

3.1.7 Waterproofing Methods

Project Flora aims to be water resistant. Essentially, the robotic garden can still function after short bouts of contact with water. Project Flora does not intend to be waterproof, which suggests that the robotic garden can function as normal after being completely submerged in water. Nor does the Auto-Gardener intend to be fully functional after a long thunderstorm. By the specifications outlined in section 2.3, it needs only to be contact with water to be functioning, not complete rain. The watering system will be accomplished by drip irrigation rather than sprinklers, so electronics should not be subjected to larger volumes of water. Although, a stretch goal is to have the robot be able to survive a typical

Florida rainstorm. Regardless, there are numerous ways of waterproofing methods, of course water-resistant methods in our case.

PCB Sprays

A PCB Varnish, such as Lakier PVB 16 PVB Varnish, can be applied as a spray directly on electronic boards. One of the benefits is that it is easy to apply. Once it hardens, it is transparent and creates a flexible, hard layer. You can still solder after the spray application. [10]

Clear Nail Varnish

This is the easiest method as it is common to find. It prevents both moisture and dust from getting on the board. A cheap method, it hardens relatively fast on the board. The circuit needs to be tested thoroughly since you cannot reverse this method. If the board gets hot, a layer of the nail coating will peel away. [10]

UV Curable Solder Mask

UV Curable Solder Mask protects the copper in the PCB. It is stable with fluctuations of temperature and has excellent adhesion. It is an inexpensive solution that comes in a syringe. Although to cure it, you need a UV flashlight. Curing is very fast, as it takes only roughly 30 seconds. The disadvantage with this waterproofing method is that it only works well on flat areas and you can only apply a thin layer. According to the blog, it works best for small mods and repair wires. [10]

Clear Silicone Adhesive

Clear silicone adhesive such as Kafuter K-705, can be used on electronics. It is non-conductive, is stable under varying temperatures and has decent adhesion. Although it is soft, you can use this to your advantage to pierce it with a multimeter probe. [10]

Heat Shrink

An additional way to waterproof wires is by heat shrink. There are various types of heat shrink, the standard heat shrink tubing made from Polyolefin [11]. A variation of Polyolefin is an adhesive lined polyolefin. The additional layer acts as a seal against moisture and other intrusions. Some heat shrinks have solder inside of it so two separate wires can be joined together. To choose the proper diameter, permit about 20 to 30 percentage diameter allowance for the heat shrink to join with the wire securely [11].

Waterproof Electronic Enclosure

Using a coating on the circuit boards as a waterproof precaution is only secondary. The first line of defense would be a waterproof housing which will store the PCB, microprocessor, and power module. A few things to consider when designing [12]:

- Take advantage of circles. This is because gaskets are typically O-rings and radial-type seals. O-rings and radial-type seals typically exert continuous, uniform force around the circumference.
- Avoid complex seals. Try not to have too many seals. Minimize the need for seals between 3 or more parts. It may be more difficult to have uniform gasket compression for more complex designs.
- Safeguard proper support when utilizing gaskets. The support needs to be on all sides, otherwise a leak may occur. The seal can fail when the gasket pulls out of the pinch zone, especially if there is external or internal pressure.
- Consider what will change in size (distort and shrink).
- RTV or liquid adhesive may be an alternative to O-rings and gaskets.
- Use a cable gland to secure the wires to the electronic enclosure. They can be used to seal the device and resist water from dripping into the container.

After researching the various waterproofing methods, we will most like go through waterproofing the hardware case by using gaskets and cable glands and use a PCB spray to waterproof the electronics. Any exposed wires will have heat shrinking and bundled together using cord protectors. To reiterate, the system will aim to be water resistant, not waterproofed, which indicates the system does not have to be submerged under water completely and keep working subsequently.

3.2 STRATEGIC COMPONENTS AND PARTS SELECTIONS

This section delves into technology investigation. This is the fundamental, important portion of the report. To determine what components will be implemented in the design, much research and comparison had to be done. These comparisons were grouped based on functionality and type. The first section is the most significant part of the design, choosing between microcontrollers and microprocessors. Next, the camera, sensors, position sensors, and power modules are discussed. The components are chosen based on how well it would work in regards to our what our needs are, performance, and costs.

3.2.1 Microcontrollers versus Microprocessors

The primary difference between microcontrollers and microprocessors is the inclusion of memory and other peripherals into the microcontroller chipset. Due to the integrated nature of microcontrollers, processor speed and memory size tends to become a limiting factor. Powerful microprocessor chips, like those manufactured by Broadcom, it becomes more difficult to buy the chip and are more often sold as a unit soldered onto a PCB.

3.2.1.1 Microcontrollers

Arduino Uno

Built around the ATmega328P, this microcontroller runs at 16MHz with 32KB of memory and 2KB of RAM. This hardware is limited also by the small number of I/O pins, 14, and the single UART pin which would limit the communication speeds of the Arduino. While the Arduino would be more than capable of controlling any motor, position control or moisture sensor functions, it would lack the memory and communication speeds necessary for operating a digital camera. The lack of communication options on the Arduino would also limit any network communications and the microprocessor included would not be powerful enough to run any image recognition algorithms. [13]

Table 5: Comparison Between ATmega's [14]

| Device | Flash | General Purpose I/O Pins | 16 bits resolution PWM Channels | ADC Channels |
|-------------------|--------|--------------------------|---------------------------------|--------------|
| ATmega328 | 32 KB | 44 | 6 | 10 |
| ATmega640 | 64 KB | 86 | 12 | 16 |
| ATmega1280 | 128 KB | 86 | 12 | 16 |
| ATmega2561 | 256 KB | 54 | 6 | 8 |

Table 5 shows that the ATmega family are different by their pin numbers and by their memory size. The table above summarizes the different qualities for each of the microprocessors. It becomes clear that the chips are named by their flash memory sizes. Not listed in the table are the sizes of the EEPROM and RAM since they all share the same sizes, except for the ATmega328 who holds a smaller size.

MSP 430

The Texas Instruments line of MSP430 microcontroller chips, specifically the MSP430G2553, features low cost and low power consumption in a small form factor. The G2553 model contains a 16MHz processor with 16KB of memory and 512B of RAM. The power consumption of the chip is the primary advantage, being able to run off a voltage of 3.6V while consuming 230 μ A in active mode. The MSP430 includes 24 I/O pins, however only one UART pin which, like the Arduino, limits the chip's ability to communicate large amounts of data to other devices. [15] While it is feasible to connect a camera to the MSP430 and transfer data using 16 of the GPIO pins, it still takes up to 38 seconds to transfer a single 320x240 image by serial connection. [16]

3.2.1.2 Microprocessors

Raspberry Pi 4

The Raspberry Pi is a single board computer that is comprised of a microprocessor, SDRAM, wireless card and a slot to insert a microSD card for memory. The microprocessor, a Broadcom BCM2711, is a 64-bit system on a chip. The microprocessor is a quad core running at 1.5GHz. Along with the included chips mentioned above, the Raspberry Pi also includes numerous options for I/O including: a 40 pin header, 2 micro-HDMI ports, a DSI display port, a CSI camera port, a 4-pole stereo audio port and several USB ports. The increased amount of components included in the Raspberry Pi does mean that its form factor is significantly larger than just microcontroller measuring in at 85mm by 56mm. [17]

Odroid-C4

Similar to the Raspberry Pi, the Odroid-C4 is a single board computer, built around the quad-core Cortex A55 cluster CPU. This computer also features a Mali-G31 GPU and passive heatsink for its components. The CPU and GPU for this board are combined into a single Amlogic S905X3 chip. For I/O options the board contains 5 USB ports, 47 general purpose pins, a 4 pin UART console connector, an RJ45 ethernet port, a HDMI 2.0 port and one micro SD slot. The board of the Odroid-C4 measures 85mm by 56mm, with the included heatsink creating 10mm of height. In order for this board to have Wi-Fi connectivity another addon board would need to be included, or a USB Wi-Fi connector plugged in. In addition to the SD card slot for memory, the Odroid-C4 also includes a eMMC module socket with support up to 64GB of memory [18]

LattePanda

The LattePanda is a single board computer that runs a windows 10 operating system. This microcomputer contains an Atom quad-core 1.84Ghz CPU, 2 to 4GB of DDR3 RAM and up to 64GB of onboard flash memory. Unique to the LattePanda is the inclusion of an Arduino co-processor, the ATmega32U4. The co-processor handles 20 of the GPIO pins for the main Atom microprocessor. For I/O the LattePanda has integrated Wi-Fi, Bluetooth, HDMI, ethernet, 32 GP pins and 3 USB ports. The board measures 88mm by 70mm, making it the largest of the three compared microprocessor boards. Since the LattePanda is built around an Intel Atom microprocessor, and uses a Windows 10 Operating System, a license key would have to be purchased in order to interface with the device and install programs on to it. [19]

3.2.1.3 Microprocessor Comparisons

The tables below include the microprocessor or microcontroller separate from the prefabricated board where possible.

Table 6: Microprocessor Operating Level Comparisons

| Microcontroller/ Microprocessor | Operating Voltage | Current | Flash Memory | Unit Price | % diff. from median price |
|------------------------------------|-------------------|------------------|-------------------|---------------|---------------------------|
| Arduino Uno | 5V | 0.2mA [20] | 32 KB | \$23.00 [20] | 0% |
| ATmega328P | 1.8 – 5.5V | 0.2mA [21] | SD Card | \$2.08 [21] | -91% |
| Raspberry Pi | 5.1V | 1.2A [22] | 16 KB | \$55.00 [23] | 139% |
| MSP430 G2553 | 1.8 – 3.6V | 230 μ A [24] | SD Card | \$1.84 [24] | -92% |
| Odroid-C4 | 5.5 – 15.5V | – N/A | 32 GB and SD Card | \$50.00 [18] | 117% |
| LattePanda | 5V | 1.6 A | 32 GB | \$89.00* [19] | 287% |

*Price does not include a Windows 10 Operating System Key

Table 7: Microprocessor additional comparisons

| Microcontroller/ Microprocessor | GPIO Pin Count | Operating Temp range | Additional ports | I/O | RAM |
|------------------------------------|----------------|----------------------|---|-----|------------|
| Arduino Uno | 14 | -40 to 85C | USB Type B | | 2 KB [20] |
| Raspberry Pi | 40 | 0 to 50 C | MIPI DSI display port, MIPI CSI camera port, stereo audio and composite video port | | 4 GB [17] |
| MSP430 G2553 | 24 | -40 to 85C | - | | 512 B [15] |
| Odroid-C4 | 40 | Issues above 20C | USB 3.0 host ports, USB host or Device mode, Peripheral Expansion Header, Debug Console | | 4 GB [18] |
| LattePanda | 32 | | Touch connector, USB 2.0, USB 3.0, HDMI, LAN | | 2 GB [19] |
| MSP430 F5529 | 63 | -40-85 C | - | | 8 KB |

3.2.1.4 Final Decision

After careful decision, the team has decided to move forward with the ATmega328 ease of use, and plenty of I/O pins, and simple processing.

3.2.2 Camera Module

According to the design of Auto-Gardener, the image sensor will primarily be used to detect weeds, plants, and foreign objects. Thus the inputs to the camera sensor will be in the designated “gardening area” and the view of that area that can be affected by varying amounts of sunlight, shade, rain, wind, etc. from a top-down standpoint. The sensor’s output are images that are transferred periodically, in batches or single images, via Wi-Fi.

Thus, the considerations for the camera will be weatherproofing, camera resolution, operating temperature, cost, operating voltage, and focal length. Since cameras already contain image sensors and have a more favorable way of connecting to the microcontroller via USB, for this project we will decide between an endoscope or borescope. As a reference, FarmBot uses 0.3 Megapixel borescope camera for weed detection [25]. These cameras are often used for hard to reach places and thus often come with light-emitting diodes (LEDs) and waterproofing. There are two types of image sensors inside the camera module, CCD and CMOS. For this project we plan to use CMOS because of lower power consumption and cheaper costs. We are not concerned with the frames per second or video feed since the camera module will optimally take each image between large intervals, i.e. every 3 seconds or possibly more because the robot must move and stop between each photo. Most machine learning algorithms will scale down their images to 256x256, so that will be the minimum required resolution of the sensor [26]. Ideally though, the resolutions of plant images taken for related projects used higher resolutions, as summarized in section 6.4.1. The operating temperature outside Orlando, not including any components producing heat around the sensor will be in the range of ~40-95 °F [27]. We will accommodate the case that the temperature exceeds further ranges by choosing a sensor with a temperature extra margin of around 10 degrees. Ideally the images captured should be in color, but we can also use the images translated to grayscale if they provide a lower loss rate when training and evaluating our model. We will discuss these considerations in further detail once we have chosen our camera.

For the selection of the camera, we have three primary types to consider: Image sensor, camera module, and borescope/endoscope. These vary in difficulty to implement from most to least respectively. For each of these types of cameras, we need the camera to communicate with the Wi-Fi module and microprocessor while being supplied with a power source. We will also contemplate a device that contains a Wi-Fi and Camera module in one. This would help alleviate costs in designing and price. The table below shows a comparison of these types by operating voltage, size, resolution, and price. Too high of a resolution may result in increasing the amount of necessary data to be transferred via Wi-Fi. The dimensions of the camera are much more important for this component than other components because the camera will be constantly moving on Auto-Gardener to take

pictures across the entire garden area. It must be small and light enough to not detriment this design.

Shown below is a table of camera modules selected for Auto-Gardener.

Table 8: Camera Module Comparison

| Camera Module | Operating Voltage | Dimensions | Resolution | Price | % diff. from median price |
|---|-------------------|----------------|------------------|---------------|---------------------------|
| OmniVision Tech. OV09724-A28A [28] | 1.5/2.8 V | 2.81x3.85 mm | 1280x720 | \$0.38 | -95% |
| OV5647 | | 25x24mm | 2592x1944 | \$3.75 | |
| Waveshare OV2640 Camera Board [29] | 3.3V | 35.7x23.9mm | 1622x1200 | \$12.99 | 63% |
| TTAKMLY Endoscope [30] | 5V (USB) | 7mm | 640x480p | \$14.99 | 88% |
| Ai-Thinker ESP32-CAM [31] | 5V | 40.5x27x4.5mm | 1622x1200 | \$7.99 | 0% |

OV5647

Arguably the highest quality camera of those researched with the largest camera resolution. The OV5647 contains just a camera module with a breakout board to connect to a device via its CSI cable. It is intended that this camera would be connected into a Raspberry Pi which would then do the processing necessary. Having a video resolution of 1440x1080 pixels this model should be more than capable of providing a smooth livestream of the garden area. The primary downsides are that it can only connect to a device via the CSI cable, and has no onboard processing of its own. For the final project design this camera, along with the Raspberry Pi were ultimately selected.

OV09724-A28A

For the image sensor, we will have to add connections on the PCB and create software to recognize and operate the sensor. The output images of this sensor would be 1280x720 bytes or 0.92 MB each image minimum, for colored images. So, we would need to connect the sensor to output images to an external memory source instead of onto our ATmega328P microprocessor [21]. The advantages of completing this are the image sensor is inexpensive and the design is more novel. Disadvantages of implementing this are more complexity in the design with having to research how to link an image sensor to a microprocessor as well as linking the sensor to an external memory source to store each image. The device also has a manufacturing lead time of 14 weeks, so it would result in a much shorter time to test and prototype [28].

Waveshare OV2640 Camera Board

Contemplating a prebuilt camera module board, the image sensor is already connected with a board and contains convenient labeled pins for use with the PCB. A significant advantage of using this is the board has built in existing libraries in C++ for usage to program the board [32]. This is useful because our microcontroller also uses the same programming language. Another positive is the board has an operating voltage of 3.3 V which is like the operating voltage of other components connected to the PCB, such as the Wi-Fi module and microprocessor. Therefore, this board will be selected to be used with Auto-Gardener. Although it is not the cheapest choice, it is not as complex to implement for our project. The disadvantages of using this device would be that the module is already designed, and the part in question is more expensive than just an image sensor.

T TAKMLY Endoscope

Borescopes and endoscopes are cameras usually used to probe difficult to reach locations [30]. The camera lens length and width are thus very small compared to our other selected camera modules and the device comes with a 16.4 ft cord [30]. This makes the type of camera compact and easy to fit into our project. The other advantages of using this endoscope are that it already meets IP67 waterproof standards and comes with six adjustable LEDs [30]. These features would allow a user to operate Auto-Gardener during the night and reduce the effort required to weatherproof the entire system. However, there is a major drawback to the endoscope. This is that it is only compatible with windows, android, and mac devices via USB connection [30]. We would have to directly contact the manufacturers for more documentation on this model, design a USB connection on the PCB, and possibly write our own library in order to program the device to operate with our microprocessor. This is the reason why the endoscope is not considered for this project.

Ai-Thinker ESP32-CAM

The fourth selected camera module to compare is a combination of both a Wi-Fi SoC (System on Chip) and camera. This module contains the same camera in the Waveshare module considered, the OV2640 [31]. The bundled discount would help save a bit on the budget. However, it is possible that the of handling both functions in one device would be complicated. For our project, we decided to separate these functions into two devices to make it simpler to manage and test. If something were to go wrong, the whole device would go bad and we would lose two functions instead of just one. In addition, we plan to have the least number of devices moving on the Auto-Gardener module, this would be just limited to the weeder tool and camera for the purposes of this project.

3.2.3 Sensor Module

This section describes the choices for the soil moisture sensor and the light sensor.

3.2.3.1 Soil Moisture Sensor

Soil moisture sensors are straightforward to use. [33] Typically, a potentiometer is built with the sensor. The sensor changes the output voltage when the soil water content changes. When the soil is wet, the output voltage decreases. When the soil is dry, the output voltage increases. Depending on how much water is in the soil, the output can be a digital signal of 0s and 1s. There is a predefined threshold built in. If the soil humidity passes that built in threshold, a LOW signal is sent out. If else, the output signal is HIGH. This signal is adjusted using the built-in potentiometer. Although, the output can be a signal between 0 and 1023, an analog signal. On the website, a user recommended to not have the moisture sensors powered up all the time. The probe will dissolve as a result of electrolysis. The solution is to use a relay to switch it on, wait, then take a reading.

The author of Practical Engineering, Grady Hillhouse, did not recommend moisture sensors that measure resistance or conductivity across soil as for one, the resistance is not a good quality as there are many factors between gardens. These factors may be minerals dissolved in the water and soil pH. Secondly, the contacts may erode quickly due to environmental factors. [3]

VH400

The VH400 is a professional option of an electronic soil moisture sensor [34]. It is very sensitive and environmentally rugged. Common soil moisture sensors use resistance or conductivity, but this soil moisture sensor does not use exposed metal to take measurements. Thus, it will never corrode or be recalibrated. Instead, the moisture sensor is made with ABS plastic and fiberglass. It uses transmission line techniques to measure the water moisture. Vegetronix advertises that it can measure the moisture between your hands and the sensor's blade. It works exceedingly fast in response to moisture changes. The VH400 can also be used as a water level sensor. The single pronged probe is 93.76 mm long, allowing it to be poked deep into the soil where the plants' roots are. The VH400 can be interfaced with an Arduino or any other common microcontroller. An additional advantage of the VH400 is that it does not have to be recalibrated, according to Vegetronix. The sole downside to this product is that it has a high price tag at \$40.

Vegetronix recommends pairing the VH400 sensor with a sensor relay board, which is also sold by the same company. The "relay boards accept input from a single sensor and control a solid state or mechanical relay" [34]. The relay board can be configured to be closed when it is wet, or vice versa. The threshold at this point is set by a variable resistor.

With the VH400 Vegetronix being about half a hundred dollars, including tax and shipping, it would take a significant part of the budget. We need to consider cheaper options. Albeit the capacitive and resistive type soil moisture meters must be considered in conjunction with a relay. The relay that will be used is the Songle Relay ISO9002. This is a protective

measure for the soil moisture meter. It uses 5V and can control up to 250V, more than enough for the soil moisture. Working fast, the relay operates at 10 msec maximum.

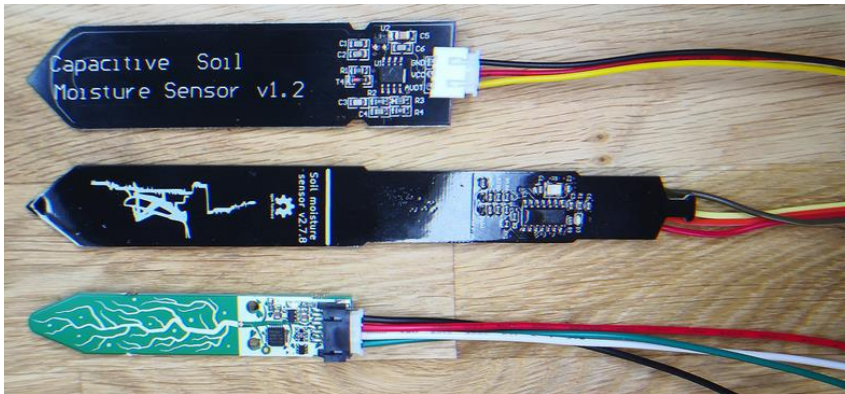
YL-69 Resistive

The YL-69 is a resistive based sensor with a LM393 comparator chip. It supports 3.3 V to 5V, making it ideal to use with Arduino. The YL-69 operates on a low current of less than 20mA [35]. However, resistance-based soil moisture sensors do not measure the moisture in the soil but measures the resistivity of the soil. The two-pronged probe is placed into the soil, and a current is passed through it. Afterwards, it reads the resistance of the soil between the prongs which are electrodes. The more moisture in the soil, the lower the resistance, and the more current which could pass through it. The lower the moisture, the higher the resistance or less water there is in the soil. The less the current can pass through.

The disadvantages of the YL-69 are that it only detects the soil moisture at a depth of 37 mm. For accurate results, you would want to measure the moisture near the plant's roots at a deeper depth. According to Taneja, it is not that accurate due to the nature of the probe [35]. It may have high sensitivity, nevertheless. High sensitivity does not mean that it has a high accuracy, as you can see in the figure below regarding the capacitive soil moisture sensors. The downside to this sensor is that it outputs a binary output, 0s and 1s. Digital output is not that accurate. On the other hand, the YL-69 moisture sensor can output an analog output, measured in ohms for resistance. A sensor shield can be applied to the circuit as an option. A sensor shield is not an active device, it merely helps the user to connect the cables.

Figure 4 gives a better idea of the lengths of each of the capacitive sensor: the SongHe capacitive sensor, the Catnip Electronics Sensor and the Adafruit STEMMA capacitive sensor, which will all be discussed in more detail afterwards. The generic capacitive sensor only has 3 connections, the GND, VCC, and analog output. The Adafruit STMMA has 4 connections, VIN, GND, SCL, and SDA, meaning it may need a pullup resistor between the SDA and SCL line. From the figure, it noticeable that the Catnip Electronics sensor is the longest.

Figure 4: In order from top to bottom, Generic Capacitive Sensor, Catnip Electronics Sensor, and Adafruit Capacitive Sensor



Requesting reprint from Flashgamer [36]

SongHe Capacitive

SongHe is a generic capacitive soil moisture sensor. It operates from 3 to 5 V, making it compatible with Arduino's. According to the author of Flashgamer, it is slow to react to changes in moisture [36]. Amazon advertises it is corrosion resistant [37]. Thus, the device could be considered rugged. The dimensions of this sensor are 5.51 inches by 3.27 inches by 0.59 inches. It is not as long as the VH400.

Adafruit STEMMA Capacitive

The Adafruit STEMMA soil sensor is a I2C capacitive moisture sensor [38], meaning it is a single pronged sensor, unlike the YL-69 sensor. The Adafruit capacitive sensor is a well-documented by how to use it in projects, such as a tweaking the firmware and showing what library to use. This sensor has a microchip built into it, the ATSAM10 chip. This gives a reading which ranges from 200 to 2000. The 200 indicates a very dry reading, while the 2000 indicates a very wet reading. This sensor also senses the temperature in Celsius. Adafruit asserts that it does not give a high precision reading of temperature. It is within the range of plus or minus two degrees Celsius. The Adafruit moisture sensor can be used with Arduino's or any other microcontroller with its I2C interface. At 3 inches long, this is the shortest of the capacitive sensors.

Catnip Electronics Capacitive

Catnip Electronics offers two versions of an IC soil moisture sensor- the rugged and non-rugged version [39]. The rugged version is merely the sensor coated in epoxy resin and a heat shrink for extra \$9. However, the light sensor becomes covered by the heat shrink, so

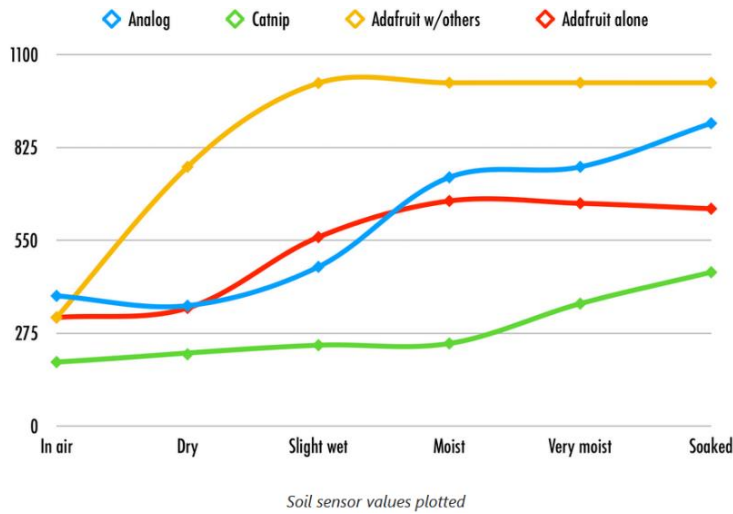
it will always report it being dark. The team can do the same thing by ourselves for a few pennies. The creator of this sensor recommends using a glue lined heat shrink for outdoor use. This soil moisture sensor has the following features [39]:

- Soil moisture sensor
- Light sensor
- Temperature sensor
- Reset chip
- IC address change
- Deep sleep mode

As with the rest of the soil moisture sensors, this can be used with an Arduino or most other microcontrollers. It operates at the standard 3.3V to 5 V. It consumes very little current. When it is idle, it uses 0.7 mA. When it is taking a measurement, it uses 7.8 mA. When it is polling at full speed, it uses 22.8 mA. This sensor has a wide range of temperature use, from 0°C to 85°C, making it ideal for outdoor use. At 5.87 inches long, this is the longest sensor of the capacitive-type sensors. Although, it is still not as long as the VH400. This sensor is also well documented, with example code, testing scenarios, and troubleshooting.

The figure below [36] shows the response of the three capacitive soil moisture sensors that were discussed above. The author waited to take the measurement over time, rather than immediately. The graph represents the average of two tests. A notable measurement of the Adafruit STEMMA sensor is that it reaches the maximum sensor value rather quickly. The figure shows that it reaches the maximum after the Adafruit STEMMA is slightly wet. When paired with other sensors, the Adafruit reaches the maximum even faster. This suggests that recalibration must be done if the Adafruit sensor is combined with other soil moisture sensors. The Catnip Electronics sensor is relatively stable. A concern is how the sensor does not change much from dry conditions to slightly wet conditions. The sensor may need to be calibrated to account for this. The generic capacitive soil moisture sensor, the line marked in red, shows to be the most sensitive sensor by itself of the three, when measuring between dry conditions and moist conditions. The Analog soil moisture is the most sensitive out of the four.

Figure 5: Comparing Capacitive Soil Moisture Sensors and Saturation



Requesting Reprint from Flashgamer [36]

Table 9 summarizes the additional capabilities of each sensor [36]. As shown, Catnip Electronics is a reasonable option. More than half the price of the VH400, Catnip Electronics sensor has the ability to measure temperature, light, and go into a sleep mode. This means that there may not be a need to have a relay to protect the sensor from dissolving due to electrolysis. The SongHe capacitive soil moisture sensor only measures the wetness of the soil, which explains the extremely cheap price. Adafruit STEMMA has two different measurements, the soil wetness as well as temperature. Judging between the 3 different capacitive sensors, Catnip Electronics is a clear winner.

Table 9: A further look into capacitive soil moisture sensors

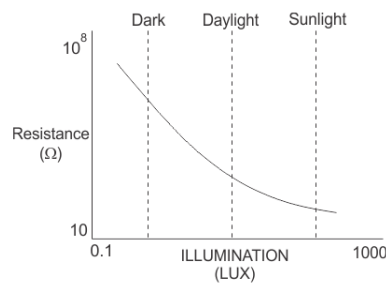
| | SongHe | Catnip | Adafruit STEMMA |
|--------------------------|--------|--------|-----------------|
| Capacitive Sensor | Yes | Yes | Yes |
| Temperature | No | Yes | Yes |
| Light Sensor | No | Yes | No |
| Deep Sleep | No | Yes | No |

| | | | |
|---------------------|---------|---------|--------|
| Probe Length | 5.51 in | 5.87 in | 3 in |
| Unit Price | \$1.66 | \$13.00 | \$7.50 |

3.2.3.2 Light Intensity Sensor

Despite the system getting real time data from the weather station, we want to implement a photoresistor, since the garden can be placed anywhere theoretically. The garden could be placed in the shaded area where there are trees or a building blocking the area. The garden bed would not receive as much sun under these circumstances. Having a photoresistor correlate its data to the moisture sensor as well as the weather data will allow the system to have more accurate results.

Figure 6.0.1: Example of Resistance versus Illumination curve for a light dependent resistor



Requesting Image Reprint from Electrical4U [40]

Photoresistors are also called light dependent resistor or LDR for short. They work on the principle of photoconductivity which is the event in which the conductivity of the material is boosted when light is absorbed by the material. On a photonic level, when photons fall onto the material, electrons in the valence band are excited to the conduction band [40]. Figure 6 shows the log to log (non-linear) correlation between the resistance and illumination in photoresistors. As it gets darker, the resistance is heightened. As it gets lighter, the resistance is lowered.

Photoresistors measure the light intensity through change in resistance. The resistance goes down to about $1k\Omega$ in light and $10k\Omega$ in darkness. According to Adafruit, “400nm (violet) and 600nm (orange) wavelengths, peaking at about 520nm (green)” [41].

Before going into discussion, some terms about photoresistors need to be defined.

- **Dark Resistance:** This is the resistance “measured at 10th seconds after closing 10 lux” [42].
- **Photo resistance:** This is the resistance measured at 10 lux with a standard light of 2854K-color temperature [42].
- **Gamma characteristic:** this is determined by the equation [42]

$$\gamma = \log(R_{10}/R_{100}) / \log(100/10) = \log(R_{10}/R_{100})$$
 where R10 = resistance at 10 lux and R100 = resistance at 100 lux

Equation 2: Gamma characteristic equation

PDV-P8001 (CdS Photoresistor)

The PDV-P8001 is a photoconductive photocell which can sense light at a range of 400 nm to 700 nm [43]. The dark resistance minimum is 0.2 MΩ, the rise response time is 55 msec and the decay response time is 20 msec.

PGM20 (CdS Photoresistor)

There are multiple types of photoresistors, PGM5, PGM12, and PGM20, all of which have several versions. All of them have a photo resistance between 2 and 300 kΩ [42]. We will only be looking at PGM20 for simplicity. The PGM20 has a higher dark resistance than the PGM5 model. The 20 indicates the photoresistor has a diameter of 20.5 mm. Table 10 shows a summary of the electronics characteristics of various models of the PGM12. All models have a maximum voltage of 500 VDC, a maximum power rating of 500 mW, a spectral peak of 560 nm, an ambient temperature of -30 °C to 70 °C and the same response time. The rise response time is 30 msec and the decay response time is 30-40 msec, depending on the model. This is not that different from the PDV-P8001 photoresistor.

| Model | Photo Resistance (10lx) (kΩ) | Dark Resistance (MΩ) min | γ min |
|----------------|------------------------------|--------------------------|-------|
| PGM2002 | 8~20 | 5.0 | 0.7 |
| PGM2003 | 18~50 | 10 | 0.8 |
| PGM2004 | 45~150 | 20 | 0.8 |
| PGM2005 | 140~300 | 20 | 0.8 |

Table 10: PGM20 Electronics Characteristics

From the table, the gamma characteristic is relatively the same between the different models. Of the series, the model PGM2004 will be the one we choose. It offers a higher dark resistance. It also offers a lower photo resistance relative to the PGM2005.

ICStation

Upon further research, the ICStation Light and Sound Sensor is merely a kit which incorporates a photosensitive resistor- the 5-10k. The ICstation DIY Light Sensor Module is overpriced on Amazon for \$14, when it is less than \$2 in parts. We omitted this choice quickly.

Catnip Electronics (as a light sensor)

According to the writer of Flashgamer, the light sensor in the Catnip Electronics Capacitive soil moisture sensor is too sensitive. It maxes out of its range before it reaches direct sunlight [36]. The sensor will notice if it is cloudy or not.

BH1750 Light Meter Module

A digital ambient light sensor, the BH1750 is an IC created for I2C bus interface [44]. This is a reasonable choice since it detects a wide range from 1 to 65535 lux at a high resolution. It does not need a high current to function. The typical voltage required to power it is 3.0V. Although, it can operate in a range of 2.4 V to 3.6 V. The peak wavelength it can detect is at 560 nm. Its accuracy measured by the ration of sensor out to actual lux is measured at 1.2 times.

3.2.3.3 Final Decision

Much like the previous sections, comparing different components is necessary to determine the optimal choice. Table 11 and Table 12 helps organize the information so that the team can decide. The most important deciding factors for the soil moisture are the price, the output signal (making sure it is not a digital output), and whether it is environmentally rugged. The most important deciding factors are the peak spectral output, the price, and the accuracy.

Much like the previous sections, comparing different components is necessary to determine the optimal choice. Table 11 and Table 12 helps organize the information so that the team can decide. The most important deciding factors for the soil moisture are the price, the output signal (making sure it is not a digital output), and whether it is environmentally rugged. The most important deciding factors are the peak spectral output, the price, and the accuracy.

Table 11: Comparisons of Moisture Sensors Characteristics

| Sensor | Op. Temp | Supply Voltage | Output Signal | Current Consumption | Env. Ruggedness | Unit Price | % diff. from median price | |
|-----------------------|--|----------------|---------------|--------------------------|-----------------|------------|---------------------------|------|
| Soil Moisture Sensors | VH400 (Vegetronix) | -45 to 85 °C | 3.5-20 V | % VWC | <13 mA | Yes | \$39.95 | 433% |
| | YL-69 (Resistive) | 10°C to 30 °C | 3.3-5 VDC | Value between 0 and 1023 | < 20 mA | No | \$1.30 | -83% |
| | SongHe Capacitive | - | 3.3-5.5 VDC | 0V to 3V | - | Yes | \$1.62 | -78% |
| | Catnip Electronics (Capacitive) | 0°C - 85°C | 3.3 - 5V | read 2 bytes | 0.7 mA | Yes | \$13.00 | 73% |

Table 12: Comparing Light Sensor Characteristics

| Sensor | Op. Temp | Vmax | Power Consumption | Resistance Range | Sensitivity Range | Spectral Peak | Unit Price | % diff. from median price | |
|---------------|------------------------------------|----------------|-------------------|------------------|--|---------------|------------|---------------------------|------|
| Light Sensors | PDV-P80 (CdS Photoresistor) | -30°C to +70°C | 150 VDC max | 100 mW/°C | 200KΩ (dark) to 10KΩ (10 lux brightness) | 400-700 nm | 520 nm | \$0.95 | -25% |
| | PGM2004 (CdS Photoresistor) | -30°C to +70°C | 250 VDC | 250 mW | 45~150 (KΩ) | | 560 nm | \$0.95 | -25% |
| | Catnip Electronics | 0°C to 85°C | - | 0.7mA | - | - | - | \$13 | 932% |
| | BH1750 | -40°C to +70°C | 4.5V | 260 mW | - | 1-65535 lux | 560 nm | \$1.26 | 0% |

After much consideration, we decided to choose the Catnip Electronics for the soil moisture sensor. The Catnip Electronics capacitive soil moisture sensor has a rugged version, does not saturate easily, and does not output binary values. The graph shows that it is not as sensitive as the other capacitive sensors, but this does not mean that it is accurate. The beauty of this sensor is that it does not reach a saturation point easily. Catnip Electronics does not require a library to use it. All it needs is a few lines of code. The measurements, looking at Figure 5 is consistent and linear.

For the light sensor, we chose the Catnip Electronics because we already chose that for the soil moisture sensor. We will test to see how accurate the light sensor is. We will also test to see if the light sensor still functions after waterproofing. If it is not accurate, the second choice is the BH1750. It is a cheap option, which can be interfaced with I2C. It has a high accuracy, and this is a better option to the photoresistors since it can adjust the measurement results. It also can measure in small increments.

3.3.4 Position Sensors

Position sensors come in a wide variety of classes, each using a different technology to determine linear and/or rotational position. Understanding what the requirements of a sensor are, as well as the design constraints of the overall project, are keys to selecting a proper sensor. Sensors come with different characteristics including size, resolution, repeatability, accuracy, mounting constraints, and environmental ruggedness [45]. Different classes of position sensors exist, each with their corresponding strengths and weaknesses.

Potentiometers

Potentiometers measure the voltage drop as a contact point slides along a resistive track. These types of sensors take advantage of the changes in voltage when that contact moves around the track, changing the resistance. They are low cost, simple, compact, and lightweight, however they are vulnerable to wear, foreign particles, and extreme temperatures. Theoretically, they have infinite resolution, but they are limited by the analog-to-digital converter interface.

Optical Encoders

Optical encoders can either be transmissive, reflective, or interferential. They all take advantage of light, encoding the patterns produced when light is detected, reflected, or refracted. The results are usually sinusoidal, which allow for repeatability. Transmissive encoders use optical scanning of a scale, which are linear or circular disks with very fine slits across it, that's been illuminated by an LED. The light from the LEDs must pass through the slits and make it to the other side of the scale where the sensors are. The intensity of the light sensed generates a voltage, which varies sinusoidally. Another sensor may be at a different phase of the scale, which culminates both readings to produce a response of the distance traveled. These are robust, accurate, highly repeatable, and cost effective sensors, however they are bulky and require a large amount of space for the actual scale. Reflective sensors on the other hand shine an LED onto a scale and detect the light reflected back, meaning the LED and the sensors are on the same side. Here, the angles, intensity, and displacement of the reflected light are used to calculate the position. This greatly reduces size of the sensor compared to the transmissive, however they sacrifice resolution and accuracy. Lastly, interferential encoders reflect lasers onto a diffraction grating and encodes the refraction. The different patterns produced give information about the distance. This is the most accurate of the three optical encoders, however, it is the most

expensive and requires a clean environment with no outside interference of light. This works best in an environment where no other light is present, minimizing contamination of another light source with the sensors.

Magnetic Encoders

Magnetic encoders use magnetic disks with multiple poles along the track. As the disk spins, the sensor keeps track of the magnetic flux occurring as the magnetic poles move across it. This type of sensor does not show absolute position but rather, sense change in position. Determining the amount of change that occurred is useful in applications where the displacement is more important than the absolute position itself. However, due to its nature, this sensor is susceptible to magnetic fields, temperature, and impact/shock resistance.

Table 13: Comparisons of Classes of Position Sensors

| Sensor | Cheap | Waterproof/Ruggedness | Data affected by temperature? | Accurate |
|--------------------------|-------|-----------------------|-------------------------------|----------------------|
| Microswitch | Yes | No | Yes | No (end points only) |
| Potentiometers | Yes | Yes | Yes | Yes |
| Optical Encoders | Yes | Yes | No | Yes |
| Magnetic Encoders | No | No | Yes | Yes |
| Coding Position | Yes | Yes | No | Yes |

A way to figure out what kind of linear position sensor to use is to look at the resolution, non-linearity, and ruggedness. Higher resolution is preferred, since it means that the output changes more accurately based on smaller steps. A lower non-linearity characteristic is ideal since it means the “difference between a straight line and the output” [46]. In terms of the Auto-Gardener, we would want a higher environmentally rugged sensor as it will face various weather conditions outdoors. The table below shows the comparison of different types of linear position sensors.

Table 14: Comparisons of Characteristics of Linear Position Sensors

| Position Sensor | Resolution | Non-linearity | Ruggedness |
|-------------------------|------------|---------------|------------|
| Magnetostriction | High | Low | High |
| LVDT | High | Medium | High |
| Inductive | Medium | Medium | High |

| | | | |
|----------------------|--------|--------|--------|
| Encoder | High | Low | Low |
| Ultrasonic | Low | High | Medium |
| Potentiometer | Medium | Medium | Medium |

For our final decision, we decided to use the microswitch due to their, ease of use, high ruggedness, and cheapness. Several can be used at each end point to allow for accurate results. The microswitch will switch on physically to let the system know the gantry has reached the end points. The ultrasonic sensor will be used as a water level sensor in the final design as getting accurate readings while the gantry is in motion proved to be insufficient. As a final precaution, the code will be adjusted to allow the system to know the position and speed accordingly.

3.3.5 Motor Module

There are four main types of motors – servos, stepper motors, brushed DC motors, and brushless DC motors. When selecting a motor, the voltage, current, torque, and RPM must be considered.

DC Brush Motors

DC motors are the simplest of the motors. They are cheap, simple to control, and have reasonable torque at slower speeds. The downsides of brush motors are that they do not have high torque, they heat up, and they are relatively poor at controlling over speed. (Insert citation)

Brushless DC Motors

Magnets are used in the rotor assembly of brushless DC motors. Brushless DC motors are more efficient and have a higher power density than brushed DC motors. With these kinds of motors, they can achieve higher speed and higher torque at the expense of having a higher current. They tend to have a longer life. The drawbacks are that they are difficult to control without a regulator, have low starting loads, and are expensive. [47]

Servo Motors

Servos have a fast response. They need accurate PWM signals. Since internally servos have a motor, potentiometer, and control circuit, a motor driver is not necessary. They are controlled via PWM (pulse width modulation) signals, which is sending electrical pulses over a control wire. Servos have a gear reduction, which has the opposite effect on torque. This means that the rotational speed is reduced by the gear ratio and some have a limited

angle of rotation. The benefit to servos is their fast response. Servos operate with either AC or DC. [47]

Stepper Motors

For precise positioning, stepper motors are the best choice. As with it corresponds to its name, stepper motors move in steps. They have a controller framework that assigns the situation through signal pulses sent to a driver, which deciphers them and sends corresponding voltage to the motor. Stepper motors are moderately easy to control; however, they draw maximum current constantly. The small distance that the motor moves prevents it from moving at top speeds. These small steps can be skipped at high loads. With this kind of motor, there is no limit on the angle and no feedback mechanism. [47]

Choosing one of these four would depend on the task that needs to be accomplished; brushless motors for high-speed racing; servos for robotic hand and tracking movement; and stepper motors for 3D printers and machine tools. For this project, we will most likely choose a DC motor to move along the x-, y- and possibly z-axis.

Linear Motors

Linear motors are one of the newer technologies Linear motors are like brushless motors which is unfolded into a straight line [48]. This motor does not require transferring rotary motion to linear motion. These components are removed from the inside. Inside of a linear motor is the motion module, linear bearing, and magnets.

Table 15: Comparing Types of Motors [48]

| Motor | Advantages | Disadvantages |
|---------------------|--|--|
| DC Brushed | <ul style="list-style-type: none"> • Simple • Good torque at low RPM • Cheap | <ul style="list-style-type: none"> • Wear out over time • Limited speed due to heating |
| DC Brushless | <ul style="list-style-type: none"> • Reliable • High speed • Efficient • No brushes therefore no maintenance | <ul style="list-style-type: none"> • Difficult to control without a controller • Needs to start at low starting loads |
| Servo | <ul style="list-style-type: none"> • Fast response | <ul style="list-style-type: none"> • Needs accurate PWM signals • Limited angle of rotation |
| Stepper | <ul style="list-style-type: none"> • Reliable • Great position accuracy • Can hold high torque | <ul style="list-style-type: none"> • Moves in small step distances which limits the max speed • Pulls maximum current continuously • Can skip steps at high loads |
| Linear | <ul style="list-style-type: none"> • Reliable • High speed | <ul style="list-style-type: none"> • Expensive • Needs custom controllers |

- No rotary to linear conversion necessary

Table 16: Comparing Types of DC to stepper Motors

| Motor | DC Voltage | Current | Torque | Max Speed, RPM | Size | Unit Price |
|-------------------------------------|------------|---------------------------|-----------|----------------|-------------------|------------|
| NEMA 17 with Rotary Encoders | 12-24 V | 0.7 A (Full load current) | 7.5 Kg*cm | 750 | 2.5"x1.75"x1.75" | \$61.71 |
| Geartisan DC Motor | 12 V | 1.1 A | 4.5 Kg*cm | 100 (no load) | 1.46" x 0.96" | \$13.99 |
| NEMA 17 Stepper | 2.8 V | 2 A | 59N*cm | n/a | 1.65"x1.65"x1.89" | \$13.99 |
| LMioEtool DC motor | 24 V | 0.32 A | 6 Kgf.cm | 100 (no load) | 4.75"x2.25"x2.75" | \$13.99 |
| LMioEtool DC motor | 12 V | 0.37 A | 6 Kgf.cm | 50 (no load) | 4" x 2.3" x 1.5" | \$13.99 |

Final Decision

Some servos have a limited angle of rotation. Servos are better for smaller projects which require a smaller torque. DC brushed motors are better car applications, where they need accurate positions. DC brushless motors can be powerful motors but can be expensive. Linear motors on the other hand can be a possible choice for the project. The motion is only straight on all axes. The issue with linear motors is how expensive they are. The final decision was the NEMA 17 stepper motor. The stepper motor provides the torque and precision needed. NEMA 17 motors are easy to find and relatively cheaply produced.

3.3.5.1 Motor Driver

Motor drivers are necessary since our microprocessor alone is not sufficient to operate the motors. This is because the supplied current from our microprocessor is not enough to output to our motors and we need to be careful not to exceed the maximum allowed current which would break our microprocessor. The absolute maximum output current for each

I/O pin is 40.0 mA in the ATmega328P p.308 [49]. Our NEMA 17 stepper motors require a 3.6 rated voltage and 1.5 A current which means that the microprocessor would need to supply 37.5 times more current on each pin to run motors without a motor driver [50]. Therefore, the motor drivers are pieces of hardware that allow the microprocessor to control a larger current/voltage load with a smaller signal [51]. Through these motor drivers, we will manage the direction and switch/on off each motor when necessary.

The table below shows a comparison of motor driver boards by load supply voltage, dimensions, peak current, and price. The drivers are sorted in ascending order by price. Load supply voltage is the output from the motor driver to the load, which is in our case the motors. The higher the range, the more room we can adjust the voltage in order to get better performance. This is because stepper motors can run at higher than their rated voltage, it is just that they may overheat without the driver regulated the current [52]. It is also important to choose a motor driver with sufficient current to run our motors, but not too much that it would go over the current the motors can handle [52]. Thus, it is not necessary we have the full 1.5 A continuous current for the motor driver. We will not need Auto-Gardener to perform its functions fast. Lastly the price per unit is considered, we will likely need at least three motor drivers, one for both Y-axis direction, one for X-axis direction, and one for the Z-axis weeder. More would be required if one driver cannot support two motors at the same time.

Table 17: Comparisons of Selected Stepper Motor Drivers

| Stepper Motor Driver | Load Supply Voltage | Dimensions | Peak Current | Continuous Current | Unit Price |
|--|---------------------|----------------|--------------|--------------------|------------|
| Aideepen DC Motor Driver [53] | 2-10V | 24.7x21x5mm | 2.5A | 1.5A | \$1.62 |
| BIQU A4988 Motor Driver [54] | 8-35 V | 20x15x11mm | 2A | 1.2A | \$1.90 |
| Pololu STSPIN220 Low-Voltage Stepper [54] | 7-45V | 15.24x20.32 mm | 1.3 A | 1.1A | \$5.95 |
| Qunqi L298N Motor Drive Controller [55] | 7-35 V | 43x43x27mm | 3 A | 2 A | \$6.69 |

Aideepen DC Motor Driver Module

This motor driver is the cheapest of the selected and comes in a bundle of five to reduce its price greatly [53]. However, the range of load supply voltage of the driver is much lower than the other drivers. This will limit the number of possible speeds we can have each motor. Overall, this motor driver is cheap and would meet all the minimum requirements, but we have decided on paying a bit more for the A4988 with a larger load supply voltage range would be more worth it for our project.

BIQU A4988 Motor Diver Module

This is the motor driver we will select for Auto-Gardener. Although it has a slightly higher price and worse specifications for peak and continuous current, the A4988 has a larger load supply voltage range. It also comes in a pack of five so we will have extra in case some error happens during testing. As the minimum requirements are met, we would not need to spend more of the budget to get additional functions for the driver.

Pololu STSPIN220 Low-Voltage Stepper

The STSPIN220 is another viable stepper motor driver for our project. In comparison to the two previously described drivers, it meets the minimum requirements. It also contains additional features including low-voltage operations as well as microsteps for 64, 128, and 256 resolutions [54]. This gives us additional control over the distance we want our motor controller travelled, but the additional cost proves unnecessary for our project.

Qunqi L298N Motor Drive Controller

The L298N motor driver has higher peak and continuous current than the other compared motor drivers, while being at a higher cost. It is also the largest in size and contains a built-in 5V power supply [55]. It may be advantageous that this device exceeds the minimum requirements to run the NEMA 17 stepper motors, but we may need to adjust the max current so that the driver would not damage our motors. This motor is not considered mostly for its relative higher cost.

3.3.6 Power Module

This section discusses the nuances of using a lithium-ion battery (both non-rechargeable and rechargeable), then the lithium polymer battery, then solar-panels, and lastly power adapters.

3.3.6.1 Lithium-Ion Battery (Non-rechargeable)

Lithium-ion batteries can easily be implemented into this project. All that is needed is a battery holder. Sometimes lower voltages will do the trick. We can double check to see if the circuit can run at lower voltages. 3 AA batteries will amount to 4.5V; 5V input may not be necessary.

3.3.6.2 Lithium Ion Rechargeable Battery

There are many kinds of batteries that can be used- alkaline, rechargeable NiMH AA's, and lithium polymer ion. What we will consider will be lithium ion. Two cell lithium polymer batteries have been tedious to use from experience. They need to be balance charged so that one cell does not have more power than the other. The battery should not be left alone by itself fully charged because of the chemistry. The cells degrade over time when it is discharged down to 0V and when it is not set to storage charge after every use. Since this will be a continuous on-off application of power use, a rechargeable lithium-ion battery is a better choice.

To track the how much power is left in a battery, a coulomb counter can be used. A coulomb counter is essentially a battery fuel gauge. Maxim Integrated has s smart battery monitor,

DS2438 [56]. It has 8 pins and is able to operate in the temperature $-40\text{ }^{\circ}\text{C}$ to $85\text{ }^{\circ}\text{C}$, which is roughly the range we are looking to operate the system in.

Another way to track power usage is by a Maxim battery monitor IC in combination with fuel gauging software. Temperature and discharge rate are considered. Maxim's fuel gauging equations are applied to any type of lithium-ion battery and a coulomb counter [57].

3.3.6.3 Lithium Polymer Rechargeable Battery

There are many things that can go wrong with Lithium Polymer (LiPo) rechargeable batteries. LiPo batteries with more than two cells need to be balanced charge. Each cell needs to have the same voltage. Another mistake that could arise is accidentally discharging the battery to 0V. This can lead to permanent cell degradation. The opposite can cause cell degradation, using a battery that is fully charged which sits around. LiPo batteries need to be set to storage charged, discharging the extra power, if it is not used immediately. Thus, even overcharging and undercharging can cause issues. You need to let LiPos be cooled down as well. They should always be stored in a cool, dry place. With all these issues, a battery management system is needed.

A design that can be implemented is using DW01X in combination with using the FS8205A. This setup is for preventing overcharge, overdischarge, and overcurrent detection and cutout. The DW01X is a one cell li-po battery protection IC. This means that more voltage is needed, thereby needing more than one cell, the DW01X should probably be attached to every cell. According to the datasheet, the DW01X is "designed to protect lithium-ion/polymer battery from damage or degrading the lifetime due to overcharge, discharge, and/or overcurrent" [58]. It is very accurate at detecting overcharge with in a 50mV detection. The FS8205A is a Dual N-channel Enhancement Mode MOSFET. It is essentially 2 switches [59]. One switch is for discharge control and the other switch is for charge control. The DW01 tells the two MOSFETs when to switch on or off. An additional design which can be incorporated is adding an IC which can balance charge cells.

The DW01X and FS8205A can also be used with Lithium-ion batteries.

3.3.6.4 Solar Panels

An idea would be to charge part of the entire system with solar power. The most feasible module that can be powered by the sun would be the microcontroller. The ATmega3288 only needs 5V to be supplied. Along with the solar panel, you would need a power management board and a rechargeable battery.

The solar panel will be connected to a circuit which is connected to a battery which will power the project. The rechargeable batteries can be any of the type described above. They

can be lithium-ion or lithium polymer. Essentially the solar panels will be connected to a similar battery management system as described in the previous section. The difference is that it needs to have a photovoltaic MPPT function. MPPT stands for maximum power point tracking. CN3722 is IC which has this function. This is design for single cell or multi-cell batteries [60]. This is also designed for both lithium-ion batteries and lithium polymer batteries. It has two modes, a constant current and a constant voltage mode. The CN3722 is rated to work at temperatures of -40°C to $+85^{\circ}\text{C}$. A feature of the CN3722 is the automatic sleep mode. The automatic sleep mode is activated when the input voltage is lower than the battery voltage.

The purpose of the MPPT is to adjust the match between the PV panel and the battery bank [61]. It converts the higher DC voltage to a lower DC voltage for the batteries. The controller reads the output of the panels, compares it to the voltage of the battery, and then calculates the best power output to charge the battery.

Many concerns arise from solar panels. They do not output a constant wattage. It changes with the intensity of light throughout the day as well as weather. Overcast weather would affect how much energy is absorbed. The placement of the panels are another concern. It needs to be placed in a sunny area away from buildings and trees and be placed at an optimal angle. For example, if the panel is rated at 12V, it will not charge the battery at a constant 12V. It can output below 12V or even higher than 12V, which is the purpose of having an IC which has an MPPT feature.

3.3.6.5 AC to DC Wall Adapter

An AC to DC wall adapter is only useful when the circuit is proven. The specifications of the adapter must match with the project. AC signals run throughout the house, but AC signals cannot be stored, whereas DC signals can. Constant voltages are essential for digital devices. Thus, to get DC signals, an AC to DC converter using a rectifier is needed.

An idea that was discussed amongst the team was to design the PCB as a full wave bridge rectifier for the AC to DC converter.

Another alternative is to have a power adapter. There are different ratings for power adapters, from 5V, 9V, 12V, and 24V. All of the power adapters are rated to convert the power outlet VAC to VDC. One model we would consider using is the HTY-1200500. This can convert a range from 100 to 240 VAC to 12VDC. This model was chosen since it is what the team had on hand.

3.3.6.5 Final Decision

Table 18 provides a brief overview of the pros and cons of each power supply. Having this format allows an ease of decision making.

Table 18: Comparing Power Systems

| | Advantages | Disadvantages |
|------------------------------|--|--|
| Lithium Ion Battery | <ul style="list-style-type: none"> • Cheap • Simple | <ul style="list-style-type: none"> • Explosive • Low maintenance • Protection required |
| LiPo Battery [62] | <ul style="list-style-type: none"> • Light weight • High capacity • Doesn't lose its charging capacity as fast as Li+ | <ul style="list-style-type: none"> • Cell degradation • Needs to be stored in a cool, dry place • Dangerous (can be explosive) • Less recharge life |
| Solar Panels | <ul style="list-style-type: none"> • Eco-friendly • Portable | <ul style="list-style-type: none"> • Expensive at higher voltages • May still need to purchase lithium battery for storage • Depends on the sun- weather may vary |
| AC to DC Wall Adapter | <ul style="list-style-type: none"> • Constant source of power | <ul style="list-style-type: none"> • System would not be portable • System would have to be near an outlet |

After careful consideration, we have decided to not use the lithium polymer batteries. They would need to have additional hardware to ensure balance charging as well as keeping it at a stable voltage. We have decided to use the AC to DC converter for any components needing more than 5V to be powered, namely the solenoid. The PCB will be powered using a lithium ion battery. We need to use two ways to power the project since there are essentially two different voltages needed as shown in Table 19. The AC to DC converter, or power adapter, is useful for high voltages. The model HTY-1200500 manufactured by Ledmo would add no cost to the project since it is what the team had. Any lower voltages can be powered using small AA batteries. This allows the system to be smaller and portable. This allows the design of the PCB to be portable as well. It can be placed on the z-axis arm or on the side of the gantry. Using solar panels will over complicate the project. We would need to implement a battery management system. There are too many parameters to consider for a battery management system. It could be an entire PCB schematic by itself.

We must ensure that the proper voltage and current are supplied to each component. Table 19 guarantees that we cover our bases.

Table 19: Understanding the Supply Voltage and Current Needed

| | Voltage | Current |
|---|-------------|---------------|
| NEMA 17 Stepper Motors | 2.8V | 2 mA |
| BIQU A4988 Motor Driver | 3.3V - 5V | 5-8 mA |
| Catnip Soil Moisture/ Light Sensor | 3.3V - 5V | 0.7mA |
| ATmega328p | 1.8 – 5.5V | 0.2mA |
| HC-SR04 Ultrasonic Sensor | 5V | 15mA [63] |
| OV2640 Camera | 1.7 – 3.3V | 6 – 40mA [64] |
| ESP 8266 | 2.5V - 3.6V | 80 mA |
| Solenoid | 6 – 12V | 160 – 320 mA |

3.3.7 Communications Module

In order to off-load processor intensive tasks such as image recognition and keep costs down this project will need a way to communicate with outside devices. Due to the possibly outside nature of the device wireless communication is preferred. For wireless communications, the project will use a Wi-Fi or Bluetooth SoC (System on chip) to allow for our microprocessor to connect to the internet. The main goal of this device is to transfer instructions between the microprocessor and an outside computer that will do the necessary computations for image recognition. Other instructions can be sent back and forth from the user to the microprocessor using this, and most of the details can be shown to the user on the web application. The factors to consider for the communications module is whether to use Bluetooth or Wi-Fi and then the appropriate comparisons between the selected SoC's will be discussed further.

Although the primary method the user will interact with Auto-Gardener is the web application and thus, we will be required to add Wi-Fi functionality, we will still discuss the tradeoffs of using Bluetooth communication for our project. Convenience of connectivity would be the main advantage for Bluetooth but is also a major weakness since it would pose a possible security vulnerability. Bluetooth devices can connect to each other through a key matching process, which would require less configuration and no password protected access compared to Wi-Fi [65]. This would make it easy for a user of Auto-Gardener to get started and working on a mobile application, but it would be possible for

anyone in the vicinity to connect to Auto-Gardener as well which raises ethical concerns. The other possible advantage of using Bluetooth is its lower power consumption [65]. Since, Bluetooth devices require less security and operate within a lower range their power consumption is lower relative to Wi-Fi devices [65]. In our project this is not as much of a concern since we are able to design the power supply and PCB to meet the higher power requirements. However, the lower range of Bluetooth is a major issue. Auto-Gardener would ideally operate outdoors and typically far from where the Wi-Fi provider would be located, so it is possible that the lower range of Bluetooth would be inconvenient for the user. Overall, a Wi-Fi device would be a much better fit for Auto-Gardener especially considering that we need higher data transfer speeds to transmit images for weed detection. Wi-Fi has much higher throughput values than Bluetooth, meaning the value of the actual data rate that a device can transfer is larger [65]. There are also significantly more applications of Wi-Fi using Internet. These comparisons are summarized in the table below.

Table 20 Bluetooth vs Wi-Fi Comparison Chart [65]

| Method | Max Data Transfer Rate | Range | Security | Power Consumption | Standard Org | Applications |
|-----------|------------------------|--------|--------------|-------------------|--------------|--------------|
| Bluetooth | 25Mbps* | 0-30m* | Key-Matching | Lower | SIG | Less |
| Wi-Fi | 250Mbps* | 100m* | WEP/WPA | Higher | IEEE | More |

*These values are taken in 2011, they may be inaccurate today but are shown to illustrate how the two methods compare to each other

In the table below is a comparison of three different Wi-Fi modules we will consider for Auto-Gardener. The first of which is a small Wi-Fi module that would solely be used to give Wi-Fi functionality to an external microprocessor. The latter two hardware are powerful enough to run the entire system of Auto-Gardener if we so choose to use them; both modules possess microcontrollers with sufficient memory and processing power. The features we will choose to compare between the selected modules are the microchips, operating voltage, size, CPU clock speed, SRAM, flash memory, and price. The features to consider the most for Auto-Gardener are the Wi-Fi chip, flash memory, and price. All chips selected will provide IEEE 802.11 b/g/n standard Wi-Fi functionality, but the more expensive ones give additional features such as Bluetooth and Hall sensors. The dimensions and operating voltage will be compared for difficulty of use in the PCB. CPU clock speed, SRAM, and flash memory are specifications that will deal with using the Wi-Fi module as a possible microcontroller. Their role will be described in brief with the paragraph following.

3.3.7 Communications Module

In this section, we will briefly discuss the value of SRAM, flash memory, and CPU clock speed in Wi-Fi modules. In the Wi-Fi SoC there is RAM memory, which consists of two types SRAM and DRAM. Both are useful for using the Wi-Fi SoC as a microcontroller to run other programs besides Wi-Fi functions. SRAM is memory used for storing global variables, libraries, and constants [66]. DRAM is used for local variables such as variables created during the use and call of functions, this type of memory is called the Stack [33]. It also holds the memory that is dynamically allocated for the program and functions which is Heap type memory [33]. DRAM is volatile memory and it requires refresh cycles to maintain the data that it is holding in each bit [67]. Both of these are necessary to run any complex programs, however SRAM is compared for these modules because it is much more expensive, faster, and less volatile [67]. Usually, using existing libraries will make the code for functions of Auto-Gardener standardized and easily readable, as well as simplify the maintenance and upscale process. Including more global variables will do this as well, so it is important to have sufficient SRAM on the Wi-Fi module. Flash memory is a type of memory external from the Wi-Fi chip. It can be used similarly to SRAM in that it is non-volatile and keeps data even when switched off [67]. Flash memory could be used to store any information that would be retained for an extended period in Auto-Gardener sort of like a USB drive or SD card, such as camera images or sensor info. However, we cannot use this as a suitable replacement for lack of SRAM because flash memory has a shorter lifespan due to physically destructive erase cycles [67].

Lastly, the CPU clock speed is simply the number of cycles a CPU can execute in a second. Instructions of a CPU each take a certain number of cycles, therefore a higher the clock speed results in a higher the number of instructions the CPU can execute. This is essential because the Wi-Fi functions depend on the CPU for transferring and receiving Wi-Fi packets. Higher CPU clock speeds will allow for higher data transfer rates in the Wi-Fi module and increasing this rate will permit an increase in value of the Wi-Fi module's throughput [68]. Thus, even if the chips are running at the same IEEE minimum Wi-Fi standard, we may expect a greater performance with a higher CPU clock speed, especially if we're using the chip to run other computation intensive tasks.

For each of the Wi-Fi SoC, they conform to IEEE 802.11 b/g/n Wi-Fi capability, provide several GPIO pins for connection to the microprocessor, and are offered at an ideal price range under \$25.00. In fact, besides the ESP8266 and ESP32 we were hard pressed to find any Wi-Fi module that contained a similar functional chip in this price range.

Table 21 Wi-Fi Modules

| Wi-Fi Module | Wi-Fi SoC | Operating Voltage | Size | CPU Clock Speed | SRAM | Flash Memory | Price |
|------------------------------------|------------|-------------------|----------------------|-----------------|------------|--------------|---------|
| SparkFun Wi-Fi Module | ESP 8266 | 3-3.6 V | 24.89x9.91 x11.94 mm | 80MHz | 32KB [69] | 4 MB | \$6.95 |
| Adafruit HUZZAH32 | ESP 32 | 3.5-6 V [70] | 35.7x23.9 mm [70] | 240MHz | 520KB | 4 MB [70] | \$13.50 |
| RedBearLab Wi-Fi board [71] | CC32 00 | 6.6-12V [72] | 73x54x15mm [71] | 80 MHz | 256KB [73] | 1 MB [71] | \$23.90 |
| Raspberry Pi 4 [74] | CYW 4345 5 | 5V | 56x85mm | 1.5GHz | 4GB | 32GB | \$55.00 |

Raspberry Pi 4

Included in the Raspberry Pi package is both an ethernet port and Wi-Fi chip. Being the most expensive of selected modules the Raspberry also has the most performance. Since the Raspberry is a complete computer, it is also significantly easier to program than the other modules, being able to run python programs natively on its operating system. Other modules considered for this project will require FTDI or SPI programmers in-order to flash their onboard memory with the necessary programming. Ultimately the performance requirements of broadcasting multiple live video streams and handling user communication over TCP/IP lead us to select the Raspberry Pi 4 for this project.

SparkFun ESP 8266 Wi-Fi Module

While primary advantages of this module are the size and price, the drawbacks of using this module lie in scalability and ease of use. Although this module has lots of software examples in the Arduino community, it would present a challenge to connect because the module has parts that are surface mounted. This makes the device difficult to prototype and repair [75]. Also, this device meets the minimum requirements of the project but does not provide features such as Bluetooth and has lower processing speeds. If Auto-Gardener were to include more complex operations or additional functions that needed higher data transfer rates, this device would not be adequate for our project.

Adafruit HUZZAH32

This device features an upgraded version of the ESP 8266 chip that the Spark Fun Wi-Fi Module has included. It has much higher SRAM, processing power, a dual core CPU, and provides additional Bluetooth functionality [72]. We did not choose this device because the extra features would be superfluous to our current project and possibly cause the project to exceed the allowed budget. Picking this device would allow possible additional functions for Auto-Gardener. One example would be simplification of a Wi-Fi connection setup. A phone app would be used to pair the Wi-Fi device and phone via Bluetooth and make it possible for the user to share the Wi-Fi SSID and password credentials conveniently. Another use would be the implementation of this device as the main microcontroller for the entire Auto-Gardener system. It would connect to all the sensors, motor drivers, and a possible SD card for extra memory storage. The disadvantage of using the Wi-Fi module as a microcontroller is that the programs would have to share resources while performing Wi-Fi and IP connections, thus we would have to design a software scheduling system to accommodate this [76].

RedBearLab CC3200WiFi board

The last Wi-Fi module described possesses a Wi-Fi SoC design by Texas Instruments [73]. It has an ARM Cortex CPU with similar CPU clock speeds to the SparkFun Wi-Fi module, except at a much higher price [73]. Therefore, the board will not be considered because the project would much rather use HUZAZH32 with Bluetooth functionality and stronger processing power. The board does have additional features though such as 256-Bit AES encryption and advanced-low power modes [73]. Overall, through the Wi-Fi modules researched the ESP32 and ESP8266 chips are much more affordable, they also are sold by a variety of brands for IoT (Internet of Things) projects. Alternatives to those two chips will require a chip of comparable prices.

3.3 PRINTED CIRCUIT BOARD

This section describes how printed circuit boards are made. This section first discusses the terminology, then the composition of the PCB, and lastly certain design recommendations.

A printed circuit board (PCB) is an electrical circuit that is contained in a mechanical structure. It has conductive features such as copper traces, pads, heat sinks, or conductive planes.

3.3.1 Terminology

- **SMD**- surface mound devices
- **TH**- through-hole components
- **Via** - an electrical connection between layers; a path that goes through one or more layers

- **Teardrops**- a teardrop shape at vias, contact pads, or traces. The purpose is to improve structural integrity when there is mechanical or thermal stress. [77]
- **Trace**- The network of wiring, insulation, and fuses. An adequate width is a requirement to ensure the required current amount without overheating [78]

3.3.2 Composition of a PCB

A printed circuit board (PCB) is an electrical circuit that is contained in a mechanical structure. It has conductive features such as copper traces, pads, heat sinks, or conductive planes. The board is made of insulated materials sandwiching conductive material. The conductive layer is made of copper foil. There are also double-sided PCBs which have the copper etched on both sides of the printed circuit board. The copper layer thickness depending on the requirements of the design for weight and current. Most PCBs have “1 ounce of copper per square foot etched on the side of the substrate” [79]. The overall board plated and covered with nonconductive material- a solder mask- and silk screen to map out electronic component locations.

3.3.3 Design Recommendations

According to David MARRAKCHI, an Engineer at Altium with over 15 years in EDA experience, the following five are the key guidelines to designing a PCB. [80]

1) Fine-Tuning

How components are placed on the PCB drastically affects the performance. How components are placed will also affect how well it meets the design requirements. Although there are many guidelines about general board layout, the following are specific guidelines to pay attention to.

Orientation: Similar components should be placed in the same direction. This ensures effective routing and efficiency during assembly process.

Placement: Try not to place components on the back side of the board (the solder side).

Organization: Position all surface mount devices (SMD) on the same side of the PCB. All through-hole components need to be positioned on the top side of the board, again to create efficiency during assembly process.

2) Power, Ground & Signal Traces Placement

The next important guideline is to route power, ground, and signal traces. Keeping this in mind will make sure your signals have a trouble-free path.

Orienting Power and Ground Planes: To prevent bending, place the power and ground planes internally centered and symmetric. Sometimes it will not be possible to have space for components on a two-layer board. To power integrated circuits (ICs), use a common rail for each supply and avoid daisy-chaining power supplies.

Routing Guidelines: Of course, connect the signal traces to match the schematic guidelines. The best practice is to “place traces as shortly and directly as possible between components”. Most importantly, if a horizontal trace route is inevitable, then route the traces vertically opposite on the side. Unless each layer is separated by a reference plane, your routing will alternate between horizontal traces and vertical traces.

Defining Net Widths: The recommended width is 0.010 inches wide for low current analog signals and low digital signals.

3) Keep Things Separate

It is common to have large voltage interfere with low voltage circuits. To regulate the interference, they recommended:

Separation: The power ground and control ground should be separate for every power supply stage.

Placement: If the ground is in the middle layer, place a small impedance path. The impedance path will reduce power interference and protects control signals. Keep the analog ground and digital ground separate.

Coupling: Attempt to have analog ground crossed solely by analog lines. This reduces capacitive coupling.

4) Combat Heating Issues

Problems can arise if heat dissipation is not considered. The circuit may start to perform worse or the board may become damaged. Heating issues could present a significant issue to the Auto-Gardener’s system as it must combat Florida’s heat. To help prevent heating issues consider the following:

Identify troublesome components. Determine which components will dissipate the most heat. To determine these components, look at the thermal resistance ratings in the datasheets. Follow the guidelines that are suggested to redirect the heat.

Add thermal reliefs. Since it can be complicated to maintain process temperatures, implement thermal reliefs on through-hole components which slows down the heat sinking rate through the component plates. The ground plane can act as a sizeable heat sink which transports heat throughout the board. So, exclude the thermal relief on a via that will allow

heat to conduct to the ground plane. Teardrops can also be added where traces connect with the contact pads to reduce thermal stress. Of course, instead of using equation 3 and 4, an online calculator may be used.

Designing thermal reliefs formulas which follow IPC-2222 Standards [81]:

$$Z = \frac{X * T}{Y}$$

Equation 3: Formula for solving for thermal connecting trace width

Where X = overall length of SMT pad, Y = quantity of thermal connecting traces, T= 60% of pad size, and Z = trace width

Equation 4: Formula for design thermal relief size

$$TR = 0.25 * X$$

Where TR = the distance from the thermal relief pad to the copper pour

Other ways to prevent heating issues to add heatsinks and cooling fans as well as keeping critical components away from high heat sources [80]. If a component dissipates a significant amount of heat, do not cluster the components together in one location. Distribute the components away from each other.

5) Check Your Layout

Verify that all the established specifications and constraints have been met. This will help you define gap widths, electrical requirements, trace widths, and other physical requirements. Some design software run checks which help detect design issues.

3.4 POSSIBLE DESIGNS AND RELATED DIAGRAMS

Figure 7 shows a possible design for the microcontroller. A motor driver, light sensor, water level sensor and a moisture sensor would be connected to the microcontroller. The two motors would be connected to the motor driver. The circuit is powered by 2 power adapters.

Figure 7 Possible ATmega microcontroller design and connections

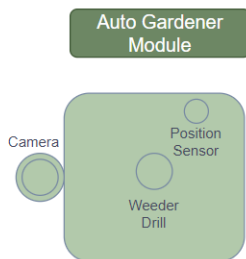
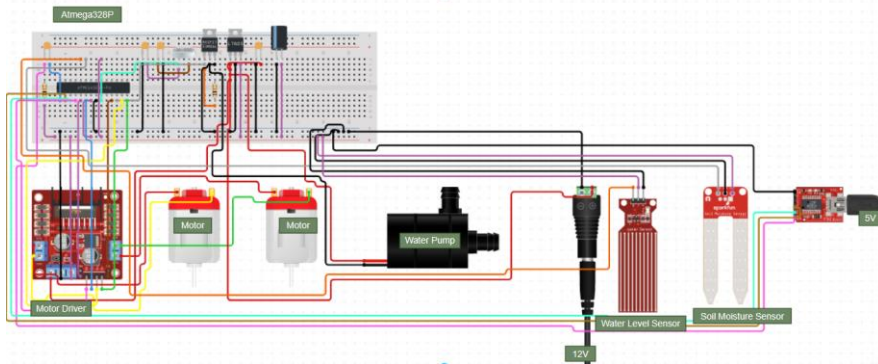


Figure 8: Bottom view of auto- gardener module with camera and weeder drill. A possible design.

3.5 Parts Selection Overview

Table 22 gives a parts selection overview for the project. Any smaller parts were omitted from this table. Any non-technical items associated with this project were excluded from this list as well. Details about how these decisions were made were discussed in Section 3.3.

Table 22: Parts Selection Overview

| Item | Part Number | Manufacturer | Cost per Part | Subtotal |
|-----------------------------------|-------------|--------------|---------------|----------|
| Microcontroller | ATmega328P | Atmel | \$2.97 | \$2.97 |
| Crystal Oscillators 16MHz | - | N/A | \$0.35 | \$3.49 |
| Stepper Motors | 17HS4401S | Usongshine | \$8.99 | \$35.96 |
| Ultrasonic Position Sensor | HC-SR04 | Smraza | \$1.90 | \$5.70 |

| | | | | |
|--------------------------------|----------------|--------------------|----------|-----------------|
| Solenoid | 634644748389 | Ximimark | \$4.50 | \$4.50 |
| Soil Moisture Sensor | - | Catnip Electronics | \$13.00 | \$13.00 |
| Motor Driver BIQU A4988 | BIQU A4988 | BIQU | \$1.90 | \$7.60 |
| Camera | OV5647 | Weewooday | \$3.75 | \$3.75 |
| Wi-Fi Module | Raspberry Pi 4 | - | \$55.00 | \$55.00 |
| Limit Switch | - | Xtension | * | * included cost |
| Gantry and Frame | ACRO 510 | Open Builds | \$294.49 | \$294.49 |

4 RELATED STANDARDS AND REALISTIC DESIGN CONSTRAINTS

With any product intended for commercial or beyond personal usage there exists regulatory bodies that ensure design standards are upheld. Since the project must also be a physical product and not simply a computer simulation, real-world limitations come into play. In this chapter the combination of regulations and real-world limitations are explored and the impact they have on the product are explained.

4.1 RELATED STANDARDS

Incorporating standards will undoubtedly improve this project's life cycle for numerous reasons. Some of these include simplifying the development of the product and its acceptability. This is due to enhancing the interoperability of different components within the system and communication to systems outside it. Standards also play a role in protecting the environment, health, and safety of those that could be affected by outlining criteria and requirements necessary to do so. Moreover, meeting their requirements will enhance the project's operability and maintenance process.

In this section, the standards created by standard developing organizations (SDO) such as the Institute of Electrical and Electronics Engineers (IEEE) and International Electrotechnical Commission (IEC) are considered for multiple parts of the project design. The details can be found from the IEEE standards website and other internet sources.

4.1.1 Battery Standards

IEEE 1547-2018 Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces

This standard deals with integrating and interconnecting distributed energy resources (DERs) in conjunction with the distribution system. Distributed energy resources are units of renewable energy used as power sources to locations such as in businesses or homes [82]. In the case of this project, solar cells will help power a portion the device's functions, so the details of this standard will prove useful.

[This standard deals with integrating and interconnecting distributed energy resources \(DERs\) in conjunction with the distribution system. Distributed energy resources are units of renewable energy used as power sources to locations such as in businesses or homes \[82\]. In the case of this project, solar cells will help power a portion the device's functions, so the details of this standard will prove useful.](#)

The standard contains eleven clauses in total, and the relevant details are summarized, from info provided by the U.S. Department of Energy [83], below:

The first portions of IEEE 1547 discuss the common technical terms required for understanding point of DER connection and point of common coupling as well as additional metrics for grid conditions. Further portions contain details about default and optional configurations, such as certain settings, that help meet the technical and performance requirements of the standard. It mentions 4 different modes of operation listed as:

- 1) Constant power factor control mode
- 2) Voltage reactive power control mode
- 3) Active power-reactive power control mode
- 4) Constant reactive power control mode

The modes differ in their capability of voltage regulation which is an important parameter in distributed energy resource connections. The DER is required to respond to abnormal conditions such as frequency ride-through and voltage ride through. It must also meet power quality performance metrics and interoperability as described in the standard. The requirements for interoperability include but are not limited to mandatory support for monitoring, control and information exchange, standardized information for nameplate, configuration, DER management, and certain communication protocols. Finally, the standard gives information regarding the testing and verification of the DER implementation to validate that it meets the requirements laid in the IEEE 1547.

4.2 PCB DESIGN STANDARD

IPC-2221 -Generic Standard on Printed Board Design

This standard is developed by the IPC (Institute of Printed Circuits) which covers generic requirements of printed board design. These guidelines are useful for the PCB design, which will support the electronic, mechanical, and electromechanical components of the Auto-Gardener. The standard gives a broad scope of information providing a combination technical terms and methods to meet the requirements.

Of the variety of topics and sub-topics, related ones as listed below and briefly described as referenced from the standard's text [84].

General Requirements: This gives an overview of necessary steps to create the design including an order of precedence, end-product requirements, design layout, parts list, evaluation, etc. It also describes boundary scan, in-circuit and functional testing for board assembly.

Materials: Different materials can be selected for the board according to structural strength, electrical properties, and environmental properties which are described in detail. It also mentions adhesives, laminate materials, plating, coatings, conductive materials, and pigmentation.

Mechanical/Physical properties: This describes options to the design such as bare board configuration, board type, board size, board geometrics, dimensions and tolerances, structural strength, bow and twist, component and feature locations, vibration design, etc.

Electrical properties: Electrical considerations about power distribution, circuit type, clearance, type of conductors, inductance, impedance controls, stripline properties etc.

Thermal Management: Methods of controlling heat and related terms to understand them are mentioned here. This includes conduction, radiation, heatsink assembly, thermal transfer and mapping, heat transfer techniques, and individual component heat dissipation.

Component and Assembly Issues: Considerations for these issues include component placement, accessibility, stress relief, flexible cable, solder terminals, fastening hardware, bus bar, etc.

Holes/Interconnections: In addition to listing of general requirements, spacing of adjacent holes, pattern variations, aspect ratio, annular ring requirements, hole location tolerances are stated.

Documentation: PCB design may require documentation detailing layout notes, accuracy and scale, deviation requirements, and more, in order to refer to and utilize after assembly.

Quality Assurance: This part gives a conformity assessment as well as multiple examples of tests to evaluate the plated hole, adhesion and surface solderability, interconnection resistance and continuity, and insulation resistance of the PCB design.

4.2.1 Electrical Equipment Standards

NEMA 250-2018 Enclosures for Electrical Equipment (1000 Volts Maximum)

This standard was made by the National Electrical Manufacturers Association. The standard has three primary goals. To begin with, the type of enclosure must be determined [85]. Each type has a rating associated with it according to its applications [85]. Secondly, a user can determine the expected features of an electrical enclosure [85]. The standard provides information preventing on construction, marking, and how to prevent corrosion for different types of enclosures [85]. Lastly, multiple conformance tests are described in its text [85]. These can verify if the enclosure can protect electrical equipment from outside hazards, such as rain, dust, dripping, lint, corrosion, submersion in water, etc. [85]. Auto-Gardener will use electrical equipment in an outdoor setting for a prolonged period, it is also not very portable so it can be exposed to many outside hazards as discussed. It is vital that we create an enclosure that conforms to these tests in order to prolong the operating time of our product.

4.2.2 Software Standard

ISO/IEC TR 19759:2015 SWEBOOK

The SWEBOOK (Software Engineering Body of Knowledge) was made possible with a combination of the ISO, IEC, and IEEE organizations and it is part of a list of publicly available standards to download from the ISO website for standardization purposes. Its goal is to provide a body of knowledge encompassing the limits of software engineering—that is, the specification of software engineering is compared to other disciplines, including its descriptions and features, and have that knowledge be consensually agreed upon worldwide [86]. It also includes a basis for certification, licensing, and educational material [86]. Software engineering is required throughout the Auto-Gardener, for construction of software that can perform its functions. Functions to perform movement, watering, detection, and communication are examples of this. It is also present in the testing, maintenance, and management of the project's operating life cycle. The information in SWEBOOK is divided into 15 knowledge areas (KA) [86]. For each KA, it fundamentally defines the topic, provides considerations, and gives tools or additional reference material.

Some relevant KA's are mentioned and described below in the guide to SWEBOOK [86]. Their ideas of fundamental definitions and descriptions are acquired from the text.

Software Requirements: This section covers the topics regarding formation of software requirements: the research, specification, analysis, legitimization and the managing of those requirements during a software product's entire life cycle. It is crucial in all projects, to avoid cases where software requirements are not properly handled and, in this case, they are "critically vulnerable". This includes Auto Gardener, which relies heavily on software to determine its next action. This project will cover all these aspects in the formation of software requirements because this is one of the initial steps in the practice of software engineering as well as referred upon in the entire process.

Software Design: This is a process continuing from software requirements that takes those requirements, analyzes them and provides a characterization describing the software architecture of the software. Software should be capable of being constructed from the description of these requirements. Thus, the specificity should be sufficient to discuss the top-level structure and organization (software architectural design/high level design) and the details for each component (software detailed design). The high-level design gives an architecture of the arrangement and role of components working together. It is the intermediate process linking requirements analysis and software construction because it defines the features of the tools needed to construct the software.

Software Construction: As implied, software construction involves the formation of working software via coding. The alteration process of coding which includes the areas of debugging, integration testing, unit testing, and verification is another aspect of software construction.

Software testing: Testing is an area where we would select some criteria and inputs to test a program that produces anticipated behavior. It is important to note that these criteria and inputs are taken from an infinite domain and the behavior can only be tested on a finite amount of test cases. What this tells with Auto Gardener, is that we must define an accurate test to meet all these specifications, such as defining what is acceptable behavior for our machine for the inputs that we provide. The project must sufficiently provide testing criteria to handle the most destructive problems, it is not possible to handle all cases. Testing can be done periodically in any case that a change would affect the current system.

Software Maintenance: The goal of this knowledge area is to provide support for software that preserves the integrity of it while being cost-effective. All activities associated with this goal are considered a part software maintenance. This process happens before and after delivery of the software. Pre-delivery activities for the Auto Gardener could include making software that will report current issues and planning for scalability so that future maintenance will go smoothly. Post-delivery activities could include modifying the software to accommodate other hardware/software, adding enhancements, correct problems, improving features such as security and efficiency, etc.

Software Engineering Professional Practice: An engineer that applies professional practice during the process software engineering does so, by conforming to a set of criteria from their applicable professional organization. These criteria can be the standards, generally accepted principles, code of ethics, professional conduct, and so on, as well as recognizing the impact of software. In addition to this, the engineer and us as taking part in this project, must comply with legal requirements and issues. The project must not conflict with any patents or copy rights and subject would be to liability if he/she chooses to sell a product that is either not appropriate or unsafe to use, it must be neither. Group Dynamics, psychology, and communication skills are also involved. As a group working in Auto-Gardener we must ensure that our outcomes are aligned, information is freely available, communication is clear, and we can work in cohesion in multiple environments. Thus, we must maintain attitudes and competency required in professional practice.

4.2.3 Communication Standard

IEEE 802.11-2016 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

This standard one part of a standard that has been several times revised, each time to adding enhancements such as data rate improvements. The original was published in 1997 [87].

The purpose of IEEE 802.11 is establishing wireless connectivity in a local area for stations that are either portable, fixed, or moving [87]. Auto-Gardener will use Wi-Fi which is synonymous to WLAN (Wireless Local Area Network) that this standard considers. All other devices conforming to this standard should be able to communicate to Auto-Gardener, and this is specifically important to relay camera and sensor information so that the computation can be handled elsewhere. Stated in the standard from IEEE [87], devices conforming to the standard will be compliant to these requirements:

- Allows use of device WLAN simultaneously with other IEEE 802.11 WLANs.
- Preserves data confidentiality and MAC management information during transfer
- Provides functions for dynamic frequency selection (DFS) and transmit power control (TPC) for band operations
- Provides quality-of-service (QoS) support for local area network (LAN) applications using MAC procedures. Video, voice, and audio applications are part of this.
- Capable of network discovery of devices and their selection. Provides information transfer from external networks and emergency services.
- Supports wireless multi-hop communication via MAC procedures
- Provides support for directional antenna patterns. This requires procedures concerned with MAC, beamforming, and PHY signaling.
- Provide for a significant number of services regarding device wireless network management. These may include but are not limited to, traffic filtering, channel usage, flexible multicast, timing measurement, and sleep modes.

4.3 REALISTIC DESIGN CONSTRAINTS

The following sections provide information on the multiple types of realistic constraints that we faced when designing this project. These constraints provided a framework for how we would develop our project. Constraints that we encountered possessed a variety of different characteristics, such as economic, environmental, ethical, and even manufacturability. Some constraints were based off of actual limitations that we faced as students, such as economic and time constraints, while others were associated with the end users' needs, like environmental and safety constraints. Ultimately, being put in a position where constraints formed an unavoidable reality to our project encouraged us to mold our design ideas into a model that was practical, yet innovative. By using our constraints as parameters and working within those parameters, it gave us an opportunity to focus on what we were able to work with and really expand our knowledge on it.

4.3.1 Economic and Time Constraints

Economic constraints impacted how we would fund our project and specifically, what parts we would end up picking for our design. Having an unlimited budget would allow us to

purchase any part we need and choose the latest technologies available. However, without a financial sponsorship from a company or from the school, the financial burden of the project was split between the four students designing it. Since we are all students with very limited personal budgets, this was a constraint we literally couldn't work around, guiding us to pick parts that were the least expensive and using test and manufacturing methods that were cheaper.

We agreed on a \$500 budget for this project, giving us our first economic constraint. Using this as our parameter, we created an estimated parts list that came out to about \$400. The bulk of this cost was for purchasing the microcontroller, the frame materials, the camera module, the multiple sensors, and the DC motors, which were the most important physical aspects needed for our type of project. The other parts, like pumps and gardening equipment, didn't need to be specific pieces, which gave us more flexibility in what parts we would choose. Along with this, picking parts that are cheaper will result in having to do more research to ensure the parts we pick work together, if we find in testing that they don't then we would have to spend even more resources to find a suitable replacement. Regarding research costs, our project intends to use artificial intelligence (AI) to detect if weeds or non-plant objects are present in the garden. In order to train our AI to determine what is considered a plant, we need to present it with datasets of plants. These datasets involve hundreds, if not thousands, of pictures that the AI can use to learn. The most practical way to obtain these datasets are to download them online. However, most datasets must be purchased in order to download them; depending on the quality and quantity of the dataset, the prices can be very expensive. This puts a financial constraint on the type of training we can deliver to our AI system. We could also create our own datasets, but the quality of the AI detection will be dramatically less than the potential it would have given the expensive, yet comprehensive datasets.

In our time of COVID-19 not every student lives on campus, so we will have to spend more on transportation to meet with group members over the course of the entire project. We may be given access to the labs in UCF but may not have chances to take full advantage of this as well as meeting with the professor to talk about parts selection in person.

Along with economic constraints, time constraints were also an unavoidable factor. We started this project in August 2020 as part of our Senior Design 1 class. Our given time constraints were to finish our final documentation by December 8, 2020 and have a final project and presentation ready by April 2021. The consequences of not following the time constraints meant that we would either be behind in our progress, or that we would fail the class and not be able to graduate. In order to facilitate a steady flow of progress throughout the research, design, manufacturing, implementation, and critiquing phases of our project, we decided to add our own personal deadlines. Using the constraints provided by the professors of our Senior Design 1 and Senior Design 2 classes as our framework, we prepared our own schedule of when research and documentation for different modules and components would be due, as well as when we would build the prototype, test and redesign, and when we would have the different sections of the report due. Having added our own time constraints made things more challenging for us because we could not spend a lot of

time on just one facet of design, like research or testing. Instead, it required us to allocate our time in a meaningful way that would allow for efficiency without sacrificing quality. Creating the final document before assembling the actual prototype created another time constraint, which meant we had four months to realize a final design on paper. However, this allowed us to have four months of assembly and testing.

Other huge, unavoidable time factors will be training the machine learning model (AI) for Auto-Gardener, as well as the students' inability to easily meet together and get things going. The process of training the AI can take varying amounts of time, lasting days or even weeks. After training it, we must then test the model; if the model has poor accuracy, then we must spend more time redesigning parts of the algorithm structure and train it all over again. This process will absorb copious amounts of time if we do not get it right on the first try. Now, during this current time, we as a nation are dealing with the pandemic of COVID-19. Our university decided to have most classes online to help prevent the transmission of the virus among students. However, this became an impediment to us since we don't have a dedicated time and place on campus to meet up, and each student lives about 20 minutes from each the other. Since our meetings occur on online platforms like Zoom and Discord, we are forced to allocate time to meet up for things like testing parts and processes, as well as physically putting modules together. We may be given access to the labs in UCF as a place to meet up, but we will not have the chance to take full advantage of them, nor the advantage of physically meeting with the professor to talk about important things like parts selection. Having this dilemma puts a constraint on the time we have to physically work on things with each other, as well as the time we have to wait for a response from our professors since they must schedule times to meet with other groups online. Also, over the course of the entire project, traveling to meet up with each other will require allocating money to pay for transportation, forming another financial constraint since gas and tolls are not free.

4.3.2 Environmental, Social, and Political constraints

The nature of our project introduced us to unique environmental constraints. For our project to be successful as a gardening assistant, the overall output had to provide a positive impact on the environment, specifically the microenvironment of the garden that it would be attending. We decided to use the environmental requirements of our end users' needs as our environmental constraints. Our project needed to sustain a biodiverse environment where multiple types of vegetables and herbs could grow, which involves manipulating and maintaining water levels, humidity levels, soil pH, and temperature of the garden, with customized care for each type of plant. This device would ultimately be a tool that aids in the preservation and fruitfulness of the garden that it manages, creating the perfect eco-friendly companion.

One problem that may arise involves the introduction of potentially foreign plant life to isolated microenvironments, causing unintended influences in the balance of the local ecology. The introduction of new plants creates the opportunity of a new source of food

for insects. If any infamous insects realize the new source of food and increase in numbers, they may pose a threat to other plants in the area, thus affecting the local bionetwork. For example, if a colony of locusts were to increase in size and become too large within a given ecosystem, local plant life would be at risk. Also, an increase in the number of locusts could affect the local food chain, increasing the number of predatory wasps, birds, and reptiles that eat locusts [88]. In order to prevent this, our device will need to properly handle an excess of insects populating in the garden and handle the prevention of insects eating the plants within the garden. By managing the prevention and/or elimination of pesky insects, our device will ensure sustainability for the garden, as well as sustainability of the local ecosystem that it's placed within.

Concerning social and political constraints, our project would support the ideals of the environmental movement, a social crusade aiming to protect nature and sustain the natural quality of the planet and the environment. In recent years, there has been a big push for companies to adopt eco-friendly measures. Taking care of the environment has gathered more popularity, along with growing one's own produce. Thus, with our device assisting users in growing and sustaining their own vegetable garden, they are also following the social movement towards personal and environmental health. One area of political concern that may arise is the local regulation of resources by municipalities, such as water. In some towns where the climate is a lot drier, water is rationed more scarcely. Our device would have to accommodate for the tight regulation of water by being efficient at monitoring and distributing the water it has access to. Another possible political constraint would be pushback from major produce companies. Large corporations in the produce market, like Monsanto, have a tight hold on garden seed control, which makes it intentionally harder for organic gardeners to grow their own vegetables and plants [89]. If this device were to get very popular and acquire enough influence, it would be on the radar of these large corporations, who would look for any opportunity for a lawsuit. Fortunately, our project does not operate in a large-scale area, nor is it expected to gain national attention since we are not investing in marketing of it. However, to prevent any future issues like these, this device would only be programmed to handle a small variety of plant types, thus making it efficient at the plants it does have experience with, while also staying non-threatening to larger corporations.

4.3.3 Ethical, Health, and Safety Constraints

Our project didn't really encounter ethical constraints, but it is still important to consider them. Ethical issues arise when people, property, or nature are negatively impacted by the actions of another. In that light, our device must not harm the health or livelihood of people, contaminate or pollute the location where it is set, and not cause harm to plants or animals that may encounter it. Since the nature of our project is to foster growth of plants for humans to consume while producing almost no waste, we will be fulfilling any ethical constraints.

As far as health and safety constraints, there are always issues that may arise. Our project must make sure that it does not negatively affect the health of its end users, as well as those who may be within its vicinity. Our project must therefore not be releasing any radiation in dangerous levels, it must not shock anyone that touches it, and it must keep sharp edges tucked away to prevent users from accidentally getting injured by them. Another health constraint would be making sure the vegetables/herbs themselves are not contaminated and that they are suitable for human consumption. Because contaminated produce could cause a serious health hazard, our project would need to focus on growing healthy produce by not using harsh chemicals and having a good monitoring system in place to detect any bad crops. It is possible that Auto-Gardener runs at night, so we must ensure that our project does not produce loud noises that could disturb the community.

In order to protect against harmful insects, different methods were considered. Among these methods were using lasers to stun insects that were considered harmful to the plants, a charged copper wire around the garden warding them away, and simply spraying natural chemicals on the plants and/or insects. For each of these options, there needs to be a measure of safety to make sure that they don't present a risk of harm to users. Lastly, the possibility of the camera being hacked is a security issue of the device. Because of that, a security constraint would be to make sure there is proper encryption and access management so that if the device connects to the internet, it can safely function without being taken over by an unauthorized outside party.

4.3.4 Manufacturability and Sustainability constraints

Manufacturability constraints give parameters to how our design can be manufactured. The goal of our device is to be something that can be manufactured easily. We are limited on where we can manufacture our design, as well as what materials we can use due to our limited budget. Manufacturing will consist mostly of 3D printing different chassis and assembling already-manufactured pieces. The chassis for the microcontroller, the different sensors, and the camera unit will all need to be designed and 3D printed to allow for efficient use and to keep them safe from the weather. The multiple tools that are garden tool will need to tend the plants will also need to be either stored somewhere or held by a chassis that allows easy access and switching of tools.

Our design will utilize moving parts, such as a camera and weed removal tool. For these units to move around, assembly of a rail system and frame are necessary that allow for easy movement without breaking down or being fragile. A framework will be assembled to allow the camera unit and gardening tools to operate within a 3D coordinate system, where different motors will move the units across each x-, y-, and z-axis. This rail system will be located above another frame holding the soil base where the plants of the garden will be planted. Because the pieces of these rails will be premanufactured, fabricating our design will require assembly. This framework needs to be able to withstand its own weight and not come undone after continued use in different directions.

Along with the physical connection of the frame, we will also need to connect these motors to power and allow space for electrical wires to run throughout the framework to power the different moving pieces. The introduction of power involves sustainability constraints. Our device needs to avoid electrical shocks, as well as be weatherproof. The nature of a gardening tool means that it would be interacting with water and dirt, two things that can easily harm an electronic device. Our device needs to be resistant to rain and splashes of water, as well as be dustproof to a certain extent to have functionality for the outdoors.

Sustainability constraints are important for creating a product that can sustain real-world physical forces. Things like vibrations, extreme temperature, the weather, and general wear and tear are all things that affect sustainability. The main constraint of our Auto-Gardener is that it must stand against different weather conditions. Rain and wind are the most common forces of weather that cause immediate damage. Being a gardening tool, our device will deal with water and plumbing systems. Because of that, it must be waterproof, handling the water within the plumbing system while also being resistant to water from the external environment. Most gardens are outdoors, so our design must take into account outdoor weather conditions. Strong winds can cause tremendous damage; thus, our device must be strong enough to endure harsh winds. Choosing materials that won't erode with water and that can withstand harsh winds allows our device to function properly outside.

Another constraint dealing with nature is being able to withstand various temperatures. Because our end users may live in either freezing or very hot and dry climates, our device must be able to withstand both ends of the temperature spectrum. The ability to operate outdoors and not overheat due to the sun is required of our project. Overheating is a common misfortune for many electronic devices because heat is not transferred out of the system appropriately. This constraint forces us to handle heat management, which can be accomplished through installation of heat shields and possibly fans aimed at major components most at risk to overheating. On the other end, resistance to snow is important if our user is to operate our device in a climate that is prone to freezing temperatures. Resistance to snow correlates to water resistance, which will already be fulfilled by our project through proper insulation of electronic components. Electronics typically perform well in colder temperatures. Regardless, we will pick components that can handle lower temperatures.

4.3.5 Mechanical Design Constraint

The biggest constraint we face is not having a mechanical engineer on our team, limiting our understanding of mechanical design. For example, the most crucial mechanical designs that we must produce are the Auto-Gardener module and the gantry wheel plates. The design of the gantry wheel plates is especially significant since inaccurate measurements can affect accuracy of movement, possibly leading to software calibration issues, as well as mechanical inefficiency with the entire design. These two designs will require precise measurements, such that we will have to use mechanical software to 3D print materials for them. This poses a problem as this software will be used multiple times to design

prototypes and the final product, yet none of us have worked with such a software before since our backgrounds are in computer and electrical engineering.

Although there is technology existing in which seeds are planted directly into the soil automatically with the help of an air vacuum pump, we have decided to omit this function from our design. Seeds are too delicate to grow outside. They first need to germinate in wet conditions. Once germinated, they survive better in a small pot indoors or in a greenhouse, where the environmental conditions can be controlled easily. In addition, we decided to exclude our initial idea of digging and planting vegetables into the soil via the gantry. The robot would need to be able to pick up a plant and dig the plant into the soil accurately. This design poses too many variables that can go awry.

As mentioned earlier, another major obstacle we will face is effectively water proofing and weather proofing our design. Since our device will primarily be operated outdoors, it's not an exaggeration to say there are many factors that we must be concerned about. These include varying operating temperatures, integrity of materials, resistance to corrosion, resistance to being submerged in water, invasion of foreign materials, etc. Our design must consider not just short-term protection of these hazards, but rather long-term since continued outdoor use can affect the operating accuracy of the machine. For example, excess dirt getting into the gantry frame would apply extra friction to the wheels driving the bridge on the y direction, causing movement to be inaccurate for that axis. Water could do the same, as well as having parts that are not uniform causing an imbalance in Auto-Gardener.

Overall, mechanical design limits the scalability of Auto-Gardener. As a machine that's goal is to perform automated tasks in a garden, many of these functions must be carefully done to avoid harming the delicate plants. Thus, Auto-Gardeners focus must shift toward other software and simple hardware parts to overcome this weakness.

5 HARDWARE DESIGN & SOFTWARE DESIGN

This chapter goes into detailed descriptions of both the hardware and software, starting from a high level of abstraction to more nuanced processes. Hardware design will cover the physical components, such as the PCB, gantry and power supply, as well as how such parts interact. Software Design will go in-depth into the programming algorithms that will be used to control the project.

5.1 HARDWARE DESIGN DETAILS

This section describes the methods we chose for the hardware design for our project. The different sections of our design correlate to the different roles the hardware modules will take. Each system essentially handles a different function; the interweaving of the different functions and how they physically interact are what make the choices for hardware design significant. To minimize the load on the motors, the Auto-Gardener Module holds the bare minimum components to capture images and monitor position. The rest of the components are outside the gantry or inside the component housing. The reasoning for component housing is to protect the components inside from weather damages as well as provide a space for which the components can be organized. The Auto-Gardener module will be able to move on three axis, the x-axis is along the bridge of the gantry, the y axis is along the long aluminum framing of the garden bed, and the z-axis which is up and down from the center of the bridge. Thus, we will need at least four motors for movement in the hardware design. Most components will be connected using appropriate wire types. Shown below is a block diagram of the hardware Auto-Gardener, in Figure 12.

5.1.2 Hardware Design Overview

The hardware can be split into three main components: The PCB which contains all the electronics, the power module which takes US standard mains voltage and converts it into usable DC for the PCB module, and the gantry which is the frame upon which all these components are built into.

The PCB can further be divided into four main components: the microcontroller, the sensors, the motor control, and communication. The microcontroller of the PCB acts as the main computing unit, taking in information provided by the sensors and communication modules then provided an output to either the motor control or back to the communications module. The sensors of the PCB consist of all ultrasonic sensors and the camera, providing necessary data to the microcontroller. The motor control module acts as an intermediary between the microcontroller and the stepper motors since the motors require a higher voltage and more amperage than what can be provided by the microcontroller. acts as an

Figure 9: Hardware Block Diagram

intermediator between the microcontroller and the stepper motors since the motors require a higher voltage and more amperage than what can be provided by the microcontroller. The communications module consists of a Wi-Fi chip, and is used to send sensor information or receive commands from a remote user.

Figure 10 shows a very general connection to how everything is connected. Since the project has been completed, the only difference is that the MicroSD memory and Wi-Fi

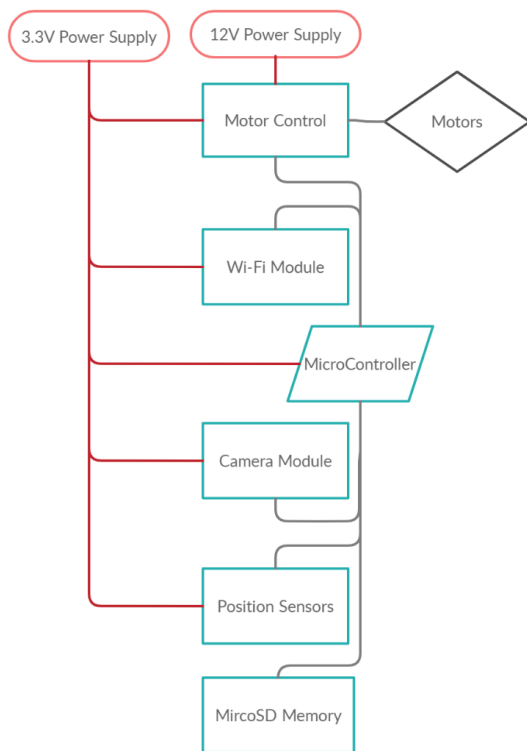


Figure 10: Hardware Overview

module is from the Raspberry Pi. Figure 11 shows a slightly more in-depth view of the inputs and outputs of the hardware.

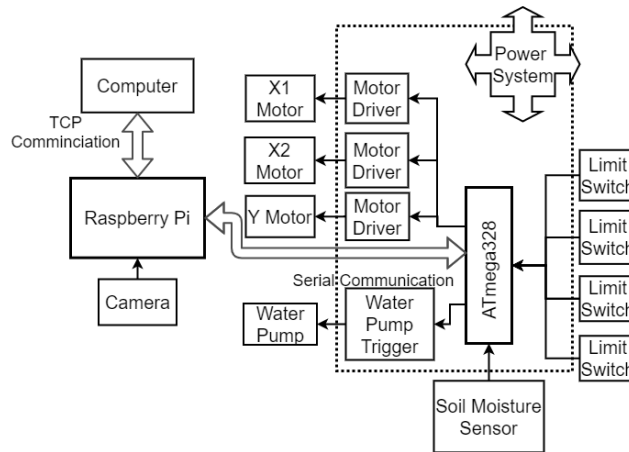


Figure 11. Hardware Block Diagram input and output connections

5.1.1 Hardware Block Diagram

The three main processors are the ATmega328, which handles the majority of the functions, the Raspberry Pi 4 Model B, and a computer hosting as a server. Fig. 1 shows the block diagram representation of the hardware interface in the overall design. The arrows indicate the flow model of the inputs and outputs. The Raspberry Pi and ATmega328 transmit data by serial communication. The Raspberry Pi and computer, which hosts the server and Web App, communicate via the Transmission Control Protocol. The ATmega328 within the PCB is the primary hub, controlling 10 of the components. The Raspberry Pi had a Wi-Fi chip and SD card that was utilized.

Figure 11 symbolically depicts how the hardware will function in this project. The red lines indicated the power lines between the power sources and the modules. Only two power supplies are needed, the 3.3V supplied from a DC to DC converter and a 12 V supply from a power adapter. The gray lines indicate the communication lines between the different modules. As shown in the figure, the processes appear to be linear from a broad point of view.

5.2 PCB

This section will discuss how the PCB will be designed. There will be two potential designs, one PCB to handle the various voltages required, and another design with the microcontroller managing the motor drivers, the position sensors, the moisture sensor, and the camera/Wi-Fi combo.

The microcontroller has an internal oscillator, but it still needs to be connected to an external crystal oscillator. This is more accurate, and you can program the clock signal. Other components needing a clock signal can also be connected.

5.2.1 Schematic

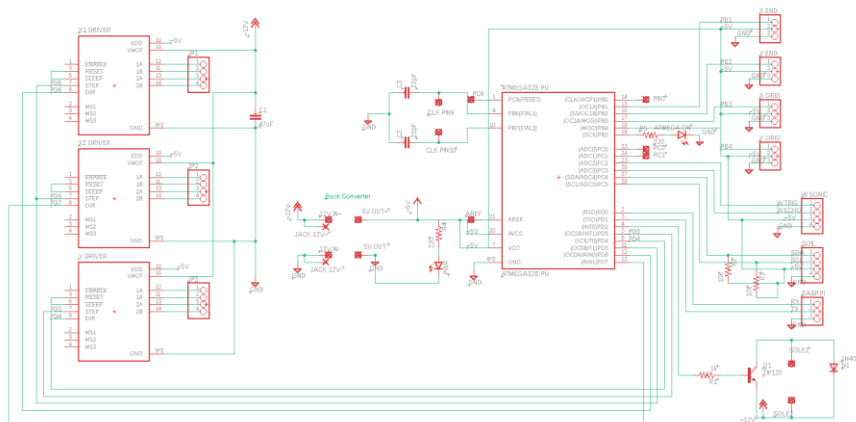


Figure 12: Schematic of the Overall Hardware

The PCB will have the ATmega as a microcontroller. Attached to that will be the Wi-Fi module and the motor drivers. At the last minute, we decided to use the ATmega1280, since we would not have enough pins open for the Wi-Fi chip, the camera, and the SD. With all-in-one combination, we may not need as many pins anymore and revert to the ATmega328p, which is what the paper discusses. For now, this is simply a rendition of a possible design.

Each of the components are attached to the different types of pins. The microcontroller has pins which have voltage or connected to ground, some are digital input/output, others are analog input/output, others have serial interface, and some are connected to the internal clock crystal. These different functions have been kept in mind in designed the connections.

Similarly, the solenoid has been represented by the teal box. The solenoid is controlled by a single pin. If the pin is on, the transistor allows the current to flow from the voltage source to power the solenoid.

The yellow box is representative of the soil moisture sensor. The soil moisture sensor is controlled by I2C, so it needs to be connected to the microcontroller through the SDA and

SCL pins. Two pull-up resistors are included. 10kohms was what was chosen, which is a common choice for ATmegas.

The motor drivers are shown along the side of the microcontroller. Each motor driver is connected to the microcontroller by two pins. One pin control the steps which is how far the motors will travel and one pin controllers the direction which tells it to turn counterclockwise or clockwise. The two x-axis motors are attached at the direction line and stepper motor line to ensure that they run in tandem.

The ultrasonic sensors have a very simple connection of two pins to the microcontroller. An LED circuit was added in order to indicate that the PCB is on. The upper right circuits are of the ESP-WROOM-32. This is the Wi-Fi, camera, and SD card combination. It is simply connected directly to the microcontroller.

5.2.1 Microprocessor

The microprocessor we will be using is the ATmega328P. It has 23 GPIO pins for our input/output purposes [49].The microprocessor has an internal RC oscillator running at 8.0 MHz, but we will be connecting it to an external 16 MHz oscillator for more accurate clock functions [49]. We will be using an Arduino board to program the micro controller.

5.2.2 Ultrasonic Sensors

Auto-Gardener will utilize one HC-SR04 Ultrasonic sensor for distance sensing to measure the water level of the water container. The Ultrasonic Sensors have these following features as taken from a retailer's website [90]:

- Min/Max Range Distance: 2cm - 4m
- Operating Current: 15mA
- Measuring Angle: 15°
- Operating Voltage: 5V DC
- Dimensions: 45x20x15 mm [91]

Each of the sensors will use four pins VCC, Echo, GND, and Trig. VCC and GND pins help power the sensor, while Echo and Trig are functions that are used for the sensors' operations [91]. For the process of using each sensor, first the microcontroller must send a signal via the Trig pin as an output pin to the sensor. This signal requires a duration of 10us for the sensor to detect it. Once this signal is detected, the sensor will send a sound eight pulses at 40k Hz and set the ECHO pin to high. After the signal is received, the ECHO pin will be at the low position so the software measuring this input pin can calculate the time elapsed between the initial pulses and return pulse [91]. Now, we have information on when the ECHO pin was last at the low position and last at the high position [92]. We can determine the actual distance from the reflected object with half of the elapsed duration between these times and the velocity of sound (340m/s) [91]. These sensors will be turned

on separately as to avoid interference with each other and the information of the calculated distance will be sent to the web application for the user to view.

5.2.3 Limit Switches

Our project will utilize four Xtension limit switches on the frame to tell the microprocessor that the Auto-Gardener module has reached the limits of the X axis and Y axis. Since we only have two sensors, it is ideal that we use it on the Y axis, which is the front and back of the garden, because the distance to traverse is the longest. Thus, the position on the Y axis would be the most uncertain. The limit switches will be placed at the edges of the gantry, when the Auto-Gardener reaches this position, the switched will be triggered. The limit switches will have an output voltage to one of the GPIO pins on our ATmega328P. This will turn the pin to high position. In software we will determine when this pin is turned to high, this will tell the microprocessor to stop supplying voltage to the appropriate motor so it will come to a halt.

The limit switch will need to be in series with a pull-up resistor in connection with the microcontroller. The Xtension limit switch by Copper Pour has no documentation or a data sheet that we can refer to, so we will refer to a similar micro switch, the Micro Switch ZM series by Honeywell. The pull-up resistor will be 10 k ohms initially [93]. If this value is too high or too low, the resistance will be adjusted. The limit switch only has 3 pins connections, one for input, one for normally close and normally open pin.

5.2.4 Moisture Sensor

Our project will be using the i2C soil moisture sensor by Catnip Electronics to measure soil moisture of our garden. This sensor uses a i2C protocol to communicate readings and provides other features like light and temperature sensing. The following technical data is provided by the retailer's website [39]:

- Supply voltage: 3.3 – 5V
- Current consumption: 0.7 – 1.1mA when idle, 7.8 – 14mA when taking measurement
- Operating temperature: 0°C - 85°C

The readings for the moisture sensor are in a linear fashion, meaning more moisture will produce a greater value of the reading. In free air at 5V supply the moisture sensors read around 290-310, according to the retailers [39]. This sensor is compatible with both Arduino and RaspberryPi controller supports. The pinout of the sensor is as follows: red – VCC, black – GND, blue/green – SDA, yellow – SCK. Also, because different soil types have varying saline content, calibration is possible to achieve absolute readings.

Source code and hardware files are all available for this sensor as it is registered as an open source hardware [94]. This allows us to program the sensor to fit specific needs if the

factory settings are not meeting our requirements and/or design constraints. Using the 6-pad header and any AVR programming hardware, we can program the sensor to fit our needs. The sensor's address can also be modified by writing a new one-byte address to register 1 and resetting the sensor, otherwise the default address set as 0x20 (hex).

Communication using the i2c protocol works by transferring data in “messages”. These messages are broken up into different frames of data, including a start condition, a stop condition, an address frame, a read/write bit, and an ACK/NACK bit. According to the website Circuit Basics, using i2c protocol has the advantage of using only 2 wires, supporting multiple masters/slaves, the confirmation of frame transfer using the ACK/NACK bit, and being a well-known and widely used protocol with lots of resources and support [95]. However, it has a slower data transfer rate compared to SPI, the data frame is limited to 8 bits, and requires hardware that, although is less complicated hardware than UARTs, is still more complicated than that needed to implement SPI.

Watering

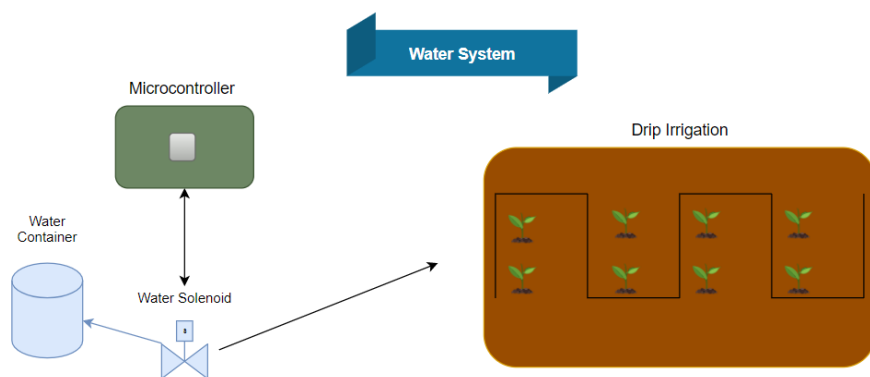


Figure 13: Block Diagram of the Water System

Figure 13 shows the potential block diagram design of the watering system. A microcontroller will determine when the water will be turned on. The soil moisture sensor reads that the soil is dry and stores this information. The system processes the information, comparing the preferred soil moisture levels to the plant. Most plants prefer to have their moisture levels cyclical, meaning they like to have periods of wetness and dryness. After it determines if the plants need to be watered, the water solenoid located at the bottom of the tank will open. Water will be stored inside of a tank, no larger than 20 gallons. Water will feed through the drip irrigation tubes in a snake like fashion. A timer will begin to determine how long the solenoid should be set as open. After the timer runs out, the solenoid will be set to close, and the cycle repeats itself.

5.2.5 Wi-Fi Module

In our finished prototype for this project, we used the Raspberry Pi 4 to handle our Wi-Fi and camera functionality. The minicomputer can connect to standard Wi-Fi, operate in normal outdoor temperatures for our test environment: Florida. The Raspberry Pi's hardware is more than capable of sending and receiving data at acceptable speeds simultaneously. In our prototype, the Wi-Fi communication of the Raspberry Pi using TCP/IP is handled by a dedicated thread, independent of the video live stream and the serial communication with the AT Mega. Continued testing is necessary to ensure the Wi-Fi component will not conflict with other devices.

The following technical specifications are taken from documentation from the developer of the Raspberry Pi 4 [23], and additional details from a technical guide to IEEE 802.11ac specifications [96] p.1-20.

The Raspberry Pi 4 contains a Broadcom BCM2711 64-bit SoC processor, with a max clock speed of 1.5GHz and 4 processor cores. This is required to process all the data sent from the user, the camera, and the microcontroller. For memory, the Pi contains 4GB SDRAM and up to 32GB of memory on an external SD card. The Pi also has 26 GPIO pins to interface with the microcontroller for communication via UART. The Pi has b/g/n IEEE 802.11 legacy compatibility. This means the chips WLAN operations are backwards compatible and can communicate with other WLAN devices such as 802.11b and 802.11n. The Wi-Fi chip supports 2.4 and 5.0 GHz receiver and transmitter frequency ranges of 284 to 2412 MHz and multiple channels for this frequency band. Higher frequency ranges ensure Wi-Fi devices do not conflict with each other because it can access different Wi-Fi providers on different channels, this means that Auto-Gardener Wi-Fi module will be suitable to be used near other Wi-Fi devices. These features ensure that the Wi-Fi chip is suitable for use in Auto-Gardener since it is interoperable with other systems.

Next, we will discuss issues with transfer speeds. The major concern for data transfer is sending images over WLAN, as they require contain the most data. The OV5647 outputs images as 8/10-bit RGB RAW data, being able to transfer 30 frames per second at 1080p resolution. Using lossless JPEG format, each frame would take 3.11MB to store. To send this image to the user within in an acceptable time of five seconds the throughput required of the Wi-Fi module would need to be at least 8.4 Mbps. The minimum requirement for a device conforming to IEEE 802.11n with a 20 MHz channel is 24 Mbps throughput [96] p.7. This requirement is well within the 114 Mbps specified by the technical documentation of the Raspberry Pi 4. This clearly allows Auto-Gardener to transfer its camera output, along with other sensor information without any issue. After confirming this, we can be sure that the Raspberry Pi 4 is suitable for our needs and no other external Wi-Fi module is necessary for this project until further requirements are needed. [17]

For connections to other hardware, the Raspberry Pi 4 will be connected to its included power supply, the OV5647 camera via CSI, and the microcontroller via the serial communication pins. The power supply will remain on as long as the microprocessor is on

because Auto-Gardener will always need to be connected to Wi-Fi to perform its functions. The microprocessor will be connected in order to send a signal to the camera when it needs to take a photo. Since these images are way too large to store in the microprocessor, the camera will connect directly to the SD card module to store each image. The transfer of images will be accomplished via a serial connection [97]. Finally, the data will be retrieved in a connection from the SD card module to the microprocessor.

5.2.6 Water Pump

The circuit of a TIP120 transistor, a 1N0001 diode, and 2.2k resistor shown in the top middle area of Fig. 2 controls the water pump. The transistor receives signals LOW or HIGH from the pin which the base is connected to. In this design shown in Fig. 2, the emitter had to be physically soldered to ground, as there was a small, yet significant error.

2.

5.4 POWER MODULE

To keep the power module simple, the Auto-Gardener will be powered by 3 sources, a 12-volt source and two 5-volt sources. The power module is simply a power adapter connected to step down voltage regulators.

2 power adapters are used in the project. The 12-volt source will be for components which need higher power. This includes the motors and the solenoid. The power will be coming from an adapter which converts the alternating current of around 120V to 12-volts. Another adapter to power the Raspberry Pi is included. It drops down the AC voltage down to 5 V DC.

This source will have a protection circuit. The protection circuit will have a non-inverting output. The non-inverting output for cases when the plug is inserted in the opposite direction. This protects the circuit from drawing power in the opposite direction.

The 12-volt source is high enough to be able to step down into the two values of voltages we need: 5 volts. The figure of the Power Circuit shows that from the DC source, a switch is applied to show whether to apply the 3.3V or 5V. The diodes ensure that the current is flowing in the right direction. The LM7085 is what steps down the voltages. In the future, these may be replaced by buck converters. With the LM7805, stepping down voltage over a longer period makes it stop working. It dissipates too much heat through the heat sinks, thus making it not able to work properly within its operating temperature range. It stops drawing current periodically and also draws too much power. Buck converters do not have this issue. The LEDs are place to visually show that current is flowing through each portion without having the need to bring out the digital multimeter for testing.

5.5 GANTRY AND FRAME DESIGN

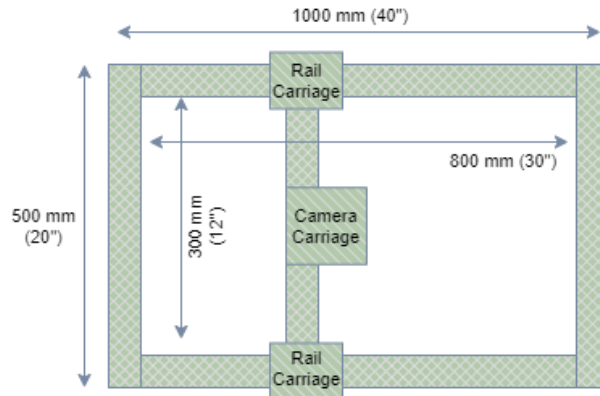
The gantry and frame design section describes how they will both be designed. As novice engineers, we will be using the design from other sources.

5.5.1 Gantry

Gantry design is important because it provides the support for any overhead structures. The movement of our Auto-Gardener requires an overhead space with support for movement in three degrees of freedom. For our gantry, we decided to have an overhead frame that would allow for both x- and y- movement along a cartesian coordinate system. An important part of gantry design is that it allows Auto-Gardener to be very accurate and precise with its functions. The system will have a consistent way of navigating around all areas of the garden in a controlled manner. Our gantry will be raised on side supports to span over the area where the garden will be located. If the gantry is not able to span the area of the garden, then functions of the device will be compromised. Being able to span all four corners of the gardening area is necessary for things such as pulling out weeds. Also, the gantry must be a rugged structure that will not tremble from light external forces. Because the gantry will hold the camera module, which will analyze data from the camera lens, it must provide a sturdy and reliable foundation. Not having a sturdy frame can harm the quality of the camera feed, therefore decreasing the effectiveness of the input data, ultimately harming the effectiveness of the AI's ability to detect plants and weeds.

The gantry will consist of four aluminum bars forming a rectangle with dimensions similar to the frame of the device. This will provide a solid structure similar to a truss that will support any moving systems or carriages attached to it. Two opposite sides of the rectangular frame will support rail carriages with wheels attached to them, allowing for parallel movement along those sides. Attached to the two rail carriages will be another frame perpendicular to the first two frames, creating an "I" shape. This design will provide for a sturdy, yet mobile platform for the camera module to be suspended on. The picture below shows how this gantry system will work. The rectangular structure will be made of aluminum t-slot trusses. These will provide strength, as well as a rail for the wheels of the carriages to travel through. Following this design allows for robustness in design, as well as the ability for the carriages to move about the frame.

Figure 14: Top view of gantry, along with rail carriages and camera module carriage



5.5.2 Frame

The frame is a very important part of the structural design of a robot. Often called a chassis, it is the framework which supports the body of the Auto-Gardener. The goal of the frame is to be lightweight, yet durable. We will be using a Cartesian style frame to achieve a 3-dimensional approach with rigid structures along the x-, y-, and z-axes. To achieve a lightweight structure, we will use aluminum frames to create the truss system. Using aluminum t-slot frames provides easy connection with other t-slot frames, as well as a providing a rail for wheels to pass along.



Figure 15: A t-slot frame sliced to show connection and rails on all four sides of the frame. Reprint permission requested by LangLe Aluminum

Commented [WB1]:

The frame sits on top of the garden base, which holds all the soil and the plants. The frame provides 4 metal trusses that extend upward from the base. Mounted on top of the trusses is the gantry, which has a height of about 16 inches, allowing room for the camera carriage

to move up and down, utilizing the weed tool that will be part of that carriage. The diagram below shows how the frame will be modeled using the aluminum t-slot trusses and the carriages. Note, the image does not show the garden base, which will be underneath and within the four legs of the frame.

Figure 16: Frame design with aluminum t-slot trusses. Gantry is seen mounted above the frame.

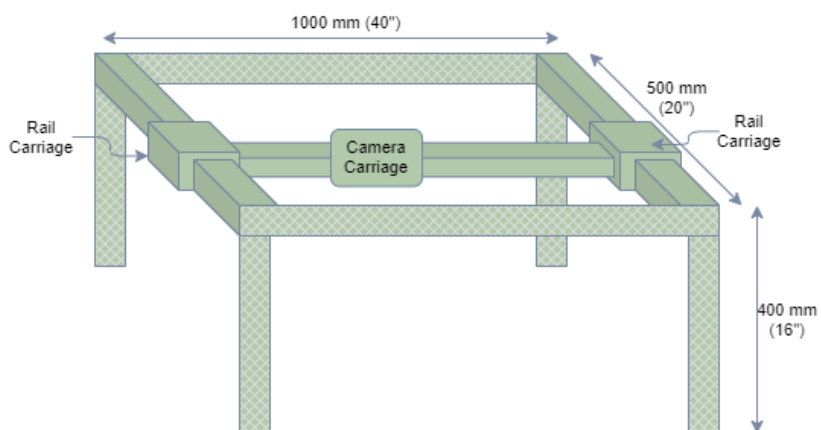


Figure 15 shows that there are rivets in the aluminum frame which will be used to hold the timing belt and wires. To install the timing belt, a small spacer will first be screwed in with a tee nut inside of these slots. The belt will be placed inside of the empty space. Then, the belt will be held down by an M3tee nut perpendicular to the belt. The wires will sit inside of these rivets and be covered by the same plastic slot covers used in 3-D printers. At the end points of the axes, the Xtension switch will be mounted. Only two Xtension switches will be used. With the help of the plastic slots, the design will appear seamless.

5.5.4 Stepper Motor and Gantry Plate Design

The stepper motor will be screwed onto the gantry plate for the x-axis. The plate will be made of an acrylic plastic with holes on various areas. The holes will be for the stepper motor shaft, the screws holes holding the stepper on to the plate, screw holes to hold the plate onto the aluminum slot, holes to place the shaft of the wheels. On the shaft of the stepper motor will hold the 14-tooth timing pulley. Attached to the toothed pulley will be

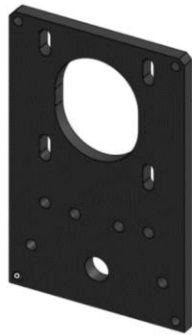


Figure 17: Stepper Motor Holder Plate Design. Reprint Confirmed from Open Builds [128]

a belt. The belt will be wrapped around the wheels. Hence, when the motors spin, the teeth will spin which will grab onto the belt, and the wheels will spin. This same design will be repeated on the other side of the x-axis except mirrored. For the y-axis, the gantry design will be repeated, but it will be rotated 90 degrees. Figure 20 shows the design of this plate that we will be using.

5.5.5 Waterproof Electronic Enclosure and Water-Resistant Modifications

This project is to be set outside, exposed to the elements. Any electrical component needs to be waterproofed. One of the main ways to waterproof the electrical components is to create a housing which will protect it from the rain and any wind which may pick up debris.

Figure 18: Component Housing Representation

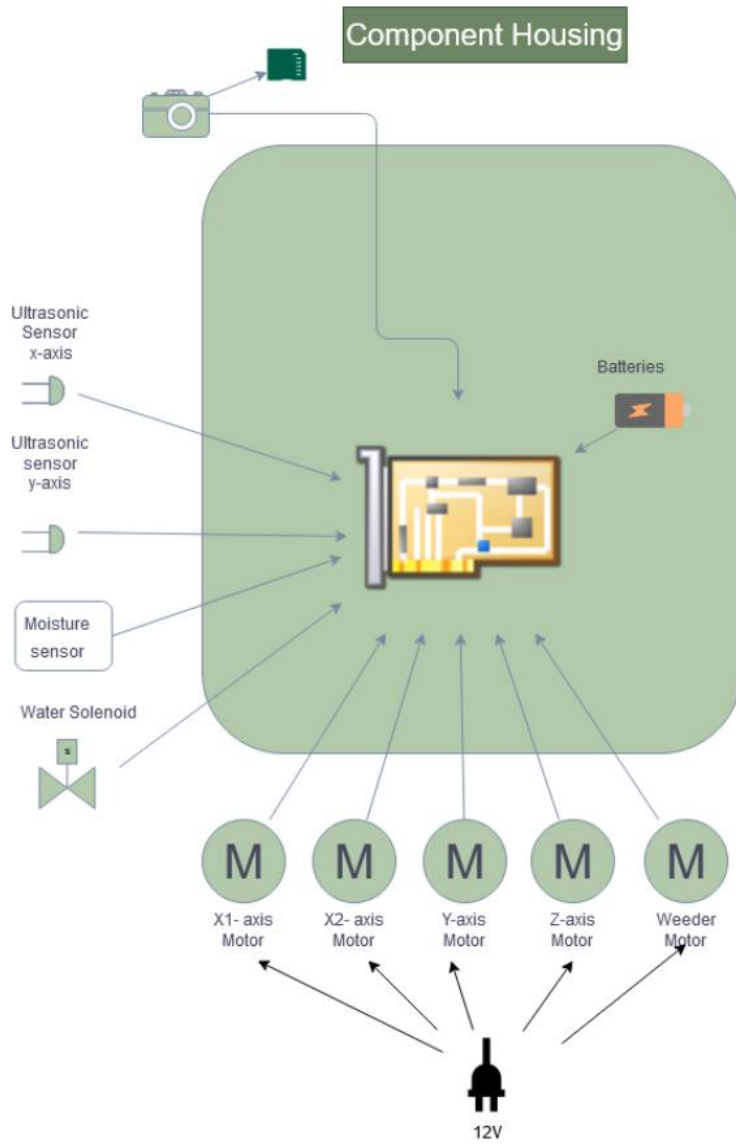


Figure 19 shows how the electronics will be combined, roughly. Figure 19 shows which of the components can fit inside of the housing. The motors need to be placed on the outside of the housing in order to make the gantry be able to move. All the sensors logically follow a similar reasoning as to why it is located outside of the electronic enclosure. The position sensors need to be placed at the end points of the x and y axis to accomplish what we are trying to achieve. The soil moisture sensor naturally should be placed in the garden bed near the plants. The water solenoid obviously should be placed on the water tank. All of these external connections will be modified to be rugged.

The bottom of the housing will incorporate cable glands and O-rings. The cable glands, located at the bottom of the enclosure where the motors are, will twist around the wires to ensure that water will not damage the circuitry. The housing may be developed using SolidWorks. This ensures that the holes that are created for bolts are exactly the size that they need to be. The radius for the wire exit points would also be the exact diameter at what gaskets are. An alternative approach would be to purchase a spill proof plastic container made for food. A constraint the team faces is not having any CAD experience. Holes would be drilled at the areas where wires would protrude. To ensure that no water will enter between the hole created and the cable glands, caulking can be used. Either design would incorporate a spray to waterproof the electronics. An extra precaution is to keep the electronic housing away from the water source. Although, it would still be subject to weather changes.

An optional design is to have angled slots along the sides to allow the parts to cool off. Florida is known to have hot weather. Any overheated parts will cause the electrical components to quit working.

Wherever exposed wires meet a new connection, glue lined heat shrink will be applied. Any wires running along the ground will be encapsulated by flexible corrugated tubing. The flexible corrugated tubing may not be waterproof, but they do protect the wires from dirt. The wires are bundled nicely together as well.

5.5.6 The Garden

It is important to document how the garden itself will be planted as well as how the water system will be physically designed.

Topsoil in backyards may not have enough nutrients to sustain fruits and vegetables; the soil may be too sandy given Florida's geography, especially on the coastal areas. Consequently, the garden soil for in-ground use like Miracle-Gro will be spread around. Nutrient rich soil should allow the vegetables to grow healthy. Seedlings will be planted with a space as little as 1 inch and as much as 12 inches depending on the plant. A list of vegetables and herbs the team will grow has already been generated but will not be listed here. Luckily, Florida has a warm climate which allows numerous plants, like cilantro, arugula, kale, and zucchini, to grow.

The garden will be irrigated using drip irrigation. As noted previously in the report, drip irrigation is the advisable direction. Sprinkler waste too much water from overspray and misdirection. Water pumps may cause overwatering. A drip line will be placed in a snake-like fashion around the garden. Figure 13 depicts the direction of the irrigation hose. The drip line may be clamped down into the ground. How long the valves will be open depends on the timer set in the software.

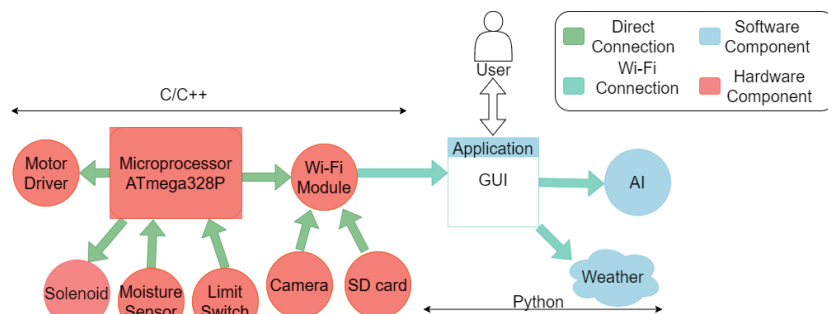
5.6 SOFTWARE DESIGN DETAILS

In this section, the software design details of Auto-Gardener will be discussed. This will include all considerations of the top level (architectural design) which answers how the components interact with each other as well as multiple sections and their sub-sections following detailing the functions of each component including which algorithms, programming languages, and potential concerns that we will be required to address.

5.6.1 Software Architectural Design

The software system of Auto-Gardener includes the microcontroller, which handles most of the functions. It connects directly to the water pump, motors, and sensors. The microcontroller connects to the internet via Wi-Fi and from there a web application is accessed that sends and receives information to Auto-Gardener. This web application will

Figure 19 Overview of Software Architectural Design



also have access to a cloud-based AI which will compute classification information back to the web application. Pictured below is a visual representation of this architecture, Figure 20. The user will only have to interact with the application to access all the software functions. Of course, the level of control for the user can vary between an administrator or

an Auto-Gardener customer and this architecture is scalable to larger projects with this design. [Fig.199 Overview of Software Architecture Desi](#)

5.7 MICROCONTROLLER SOFTWARE DESIGN

The software aboard the microcontroller will handle the majority of the functions performed by the Auto-Gardener. The main program loop will cycle through each of the functions: communicating with the web application for an update of parameters or to receive new commands, automated checking of soil moisture, whether to activate the water pump to water the planter, and finally movement of the camera bed to the next position to capture a new image to be sent on the next communication cycle.

5.7.1 Microcontroller Algorithms

These will be decided by the microcontroller, but settings can be altered using the web application. The programs on the ATmega with an Arduino bootloader. These algorithms will be performed in general by turning on the required devices need to perform actions, sending a command via an output pin to that device to trigger operations, and inputs reading in information from those devices. Generally, Wi-Fi communication will be on to receive the initial commands from the user. Then Auto-Gardener will perform either watering or planter patrol. Since these functions are complex, they can be further broken down to their individual algorithms, as discussed below.

Connecting to Wi-fi

On startup, the microcontroller loop will connect to a hardcoded Wi-Fi SSID and send a connectivity check to a hardcoded local IP address. The address will be that of the computer hosting the application GUI and image processing software. For this, the microcontroller will follow the IEEE communication standards mentioned above. At the start of each cycle the microcontroller will check its connectivity before attempting to send or receive information to prevent errors. If it is unable to verify its connection, it will attempt to save and record data onto its included memory drive.

Receive Other Information (Instructions to perform tasks)

During the communication function the microcontroller will first check to see if it has received any updated parameters regarding, soil moisture thresholds, camera movement patterns or timing of movement. It will then check if it has received any user ordered movement commands which if true will immediately trigger the movement function and then check for a user ordered camera capture commander, executing if needed. The function will then check to see if it has received any detected weed locations and add them to the weeding function queue.

Movement

The movement function of the program will be called by other program functions in order to accomplish a given task. Both user command and plant patrol functions would be able to either give the command to go to a specific position or move in a simple direction. Using the position tracking function to make sure the gantry does not attempt to go out of bounds, and thus off the rails possibly damaging the hardware, the movement function will set specific hardware pins on the ATmega to signal to the motor drivers to power the motors. For simple movement commands the time the motors will receive power is governed by one of the parameters that could be changed by the user. Otherwise, the movement function will run the motors for as long as needed to reach a specific position.

Position Tracking

Before and after any movement, the position tracking function will be called to check the inputs of the position management hardware. The hardware solution chosen for determining the camera platform's position will dictate exactly how the program will execute this function, but the return result will be the same: A X-Y position coordinate. With this coordinate data, other functions will be able to perform necessary calculations and this data can be sent through the communication protocol to update the application GUI.

The microcontroller will use a combination of limit switches and ultrasonic sensors to track positioning. There will be two limit switches on the frame, which will output a potential difference when the bridge has reached that sensor. The microcontroller will note that output and know to stop the motors. The ultrasonic sensors will allow Auto-Gardener to determine its approximate position relative to the back and side of the frame. The microcontroller will send a signal to trigger the ultrasonic sensor, the sensor will constantly emit a sound wave and output a voltage to the microcontroller when the sound wave is reflected, and the sensor receives it. Using the time elapsed between the trigger of the sensor and the return output along with a predetermined constant, the speed of sound, we can use software to calculate distance. The first ultrasonic sensor will determine distance from the back of the frame to the front and the second from one side of the frame to the other. This information will then be sent to the web application in the future.

Planter Patrol

How the camera platform moves across the planting bed, along with when and how often to take pictures will be governed by the microcontroller's planter patrol function. This function will make use of the movement, position tracking and photo capture functions in a specific way to ensure that not only is the entire planting area photographed at least once per given standard, but it is done in the most efficient possible way. In order to allow for flexibility in planting areas and different plants the user will be able to specify which pattern the platform will follow. This could include simple patterns such as scanning from left to right then moving from one end of the planting area to the opposite end or could be a complex movement of photographing specific coordinates first to check on the health of known plants then random photographs to check for weeds.

User Commands

From the application's GUI, the user will be able to send movement and camera capture commands to the microcontroller through the communication protocol. To ensure maximum usability, these commands will be executed first and can interrupt the planer patrol's routine. Since the user can send a variety of commands in differing order, a queue will be necessary to ensure the software feels responsive to the end user.

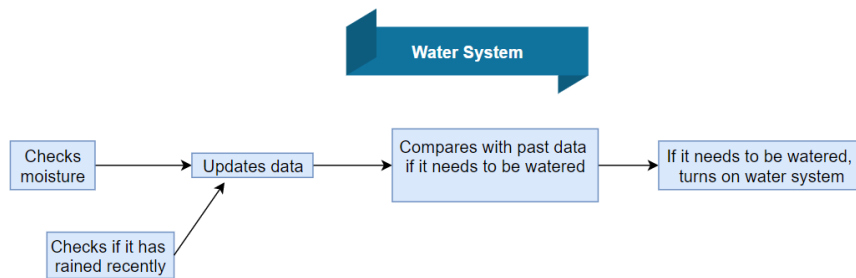
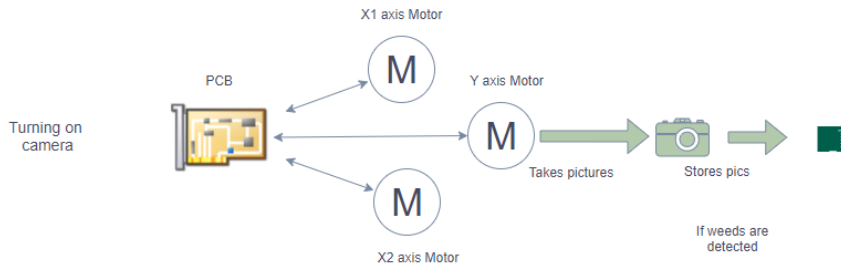


Figure 20: Block Diagram of Software for Water System

Watering

Managing the soil moisture level will be a two-step process within the main program loop; the first step would be to check the input from the soil moisture sensor, if the value falls below a threshold parameter then the second step would trigger. The second step would involve setting one of the GPIO pins on the main PCB to high, this pin would be connected to the valve or pump allowing water to flow out of the reservoir and into the planting area. A second parameter would govern how long the water should flow into the planting area once triggered. This function could then be triggered repeatably by the main loop in order to raise the soil moisture levels above the threshold value. A visual representation of the water system is shown below in Figure 20.

Figure 21: Block Diagram of Functions of Camera



Taking Photos

A function to trigger the camera module to capture an image would be triggered by either user command or by the planter patrol. The program would then make a call to the device driver of the camera to capture an image and write a file to the onboard memory of the device. This file name would be saved within an array such that other program functions can access a list of recently archived images needing to be sent.

Sending Information

At the ending of the main program loop, the microcontroller will communicate back to the host device. The function will first validate it is still connected to the network and the server is online. After validation, the archived photos from the planter patrol will be sent for image recognition processing on the server. The current soil moisture levels, gantry coordinate positions, and status of weed removal queue will then be sent. Once all relevant information has been sent, the microcontroller will send a confirmation packet to validate that all data was received. Once the microcontroller receives confirmation

5.7 WEB APPLICATION DESIGN

The web application will be the primary interface that the user will have to view details and adjust parameters of Auto-Gardener. It will be necessary to have a user-friendly interface so that the user can find and adjust the appropriate settings, such as water dispersion amount and weeding frequency. Information obtained from sensors and user preferences will be stored in the database for future use.

5.7.1 User Interface Design

The graphical user interface (GUI) will have control of all the primary functions in Auto-Gardener. We will use the Kivy framework to design the layout of the web application. Kivy is a cross-platform, python-based framework that specializes in handling media. It separates layout design and algorithm logic by using kv language to handle the layout design of the GUI and the python language to handle the logic. The web app will separate

main functionalities by different tabbed pages. Each tab will provide different controls and functions. At all times the connection status, battery level, operating status, and camera feed will be available to see from Auto gardener on a locked sidebar. The application will have five pages which consist of the main page, manage plants page, cloud AI page, view history page, and contact us page.

As discussed, a small, fixed sidebar will remain on the right of the website. It must maintain a constant supply of updates to display to the user from the microcontroller. The connection status will show either connected, connecting or disconnected from the Auto-Gardener. An option to view the last photo taken by the camera is given, this will change periodically, and this output can be paused. There will be hidden tabs which the user can expand and view live output details from the sensors connected to the microcontroller. Lastly the sidebar will display the current task of Auto-Gardener. The user has an option to stop this task unless the Auto-Gardener is in certain phases such as “shutting down” or “getting things ready,” in this case the buttons will be greyed out and unavailable to use.

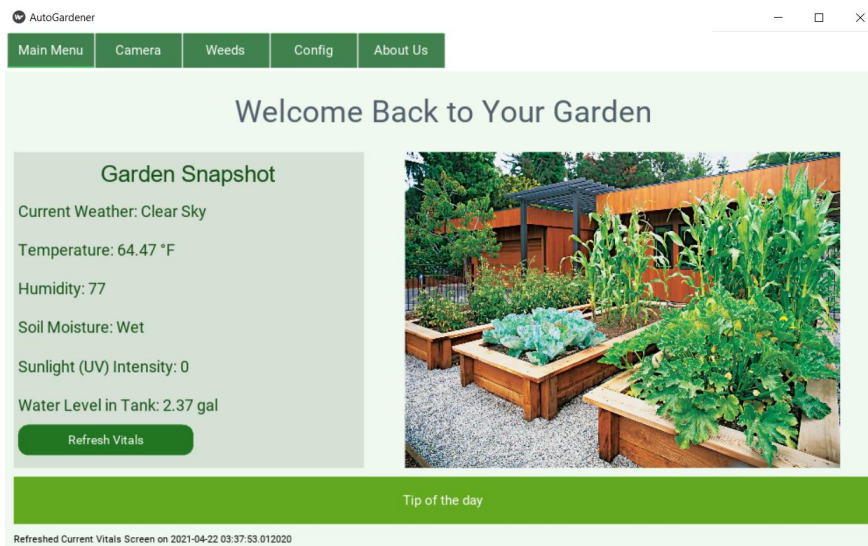
The main page is the front page of the web application. It first gives user information on the date and weather condition. This will help the user decide whether to water the plants that day. It will also provide the user a visual image of the location of the camera via a coordinated grid. The grid will give the option to display either a combination of photos of the top-down view of the garden area, or a user-friendly grid with icons to view individual plants. The grid will allow navigation with the cursor to view different coordinate points as well as a zoom function. A visual example of the main page is shown below in Figure 23 created with the use of Marvelapp.com prototype. After the grid is shown, the selected plant details will be provided for the user to view. The user can edit details such as the nickname for the plant and add appropriate comments to it. Any user input will be saved on the database, so when the page is refreshed or returned to the information will not be lost. The user can view a more detailed image of the plant by selecting the zoom button next to the image of the plant. Certain statistics of the plant will be recorded such as the date of first planting, plant age, and duration since last water. Some of these functions will have to be calculated when the user selects the plants.

The manage plants page consists of a drop-down list of plants identified in the garden. This will provide the same information as in the main page, but in a condensed manner and for many more plants at a time. This page’s functions will prove useful to the user when needing to manage multiple plant profiles or when necessary to remove/replace plants.

The cloud AI page will contain options to view and manage cloud AI settings. The primary changes will be in the super parameters of the AI which are chosen according to user preference. In the case that we are provided with more datasets to train on or noticeable

updates on effective weed detection algorithms appear the user may be able to update the AI training model within this page.

Figure 22 Main Page in Auto Gardener Application



The view history page contains the operational history of Auto-Gardener. Akin to the manage plants page, it will have a drop-down list of dates when operations and watering have been done. Different types of events will be denoted by different icons/colors and possess specific information. For example, the watering event can display the number of plants watered, percentage of plants water, give links to the manage plants page for each plant that was watered, display time the task was initialized, date of completion, and duration taken.

Contact us will provide the user information on the developers behind Auto-Gardener. It will give brief descriptions of the roles of each engineer, motivation behind the project, and methods of contacting the developer. An option to send information in case of a hardware error or software bug will be contained here. Other cases where credit is given for contribution to the project will be shown. The date of last software maintenance and website version will be displayed on the bottom, so the user can know if their product is up to date.

These are all the planned pages so far, future pages may include a frequently asked questions and troubleshooting page, database of plants to plant with common suggestions, links to other recommended pages, and possibly a documentation page to show our design.

5.7.2 Database Design

The database system is the primary way that Auto-Gardener will store information. The type of database this project will use is a relational database because our members have previous experience with this type. Database storage is preferred to other storage methods because it already has built in commands for data retrieval and access, and it is already interoperable with most website or app-based programming language. As such, even with a change of programming language for the website, the portion of effort attributed to data storage is transferable to other projects. It is also scalable and easy to modify through preexisting database handling software. This section will discuss the structure of the database, as well as the means that the website sends information to the database and receives information.

5.7.2.1 Database Structure

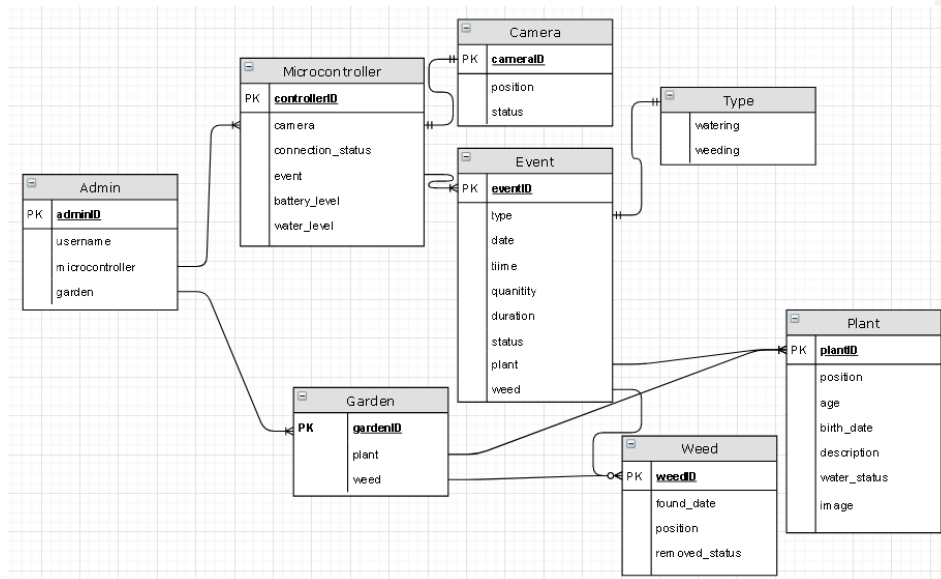
The application we will be using to generate the database code and host the database server is MySQL Workbench. The version is open source and free for commercial use. The method for constructing the database will be simply to create a model then add tables to the model for every entity necessary on the website. Each table will list multiple variables with their associated variable types such as Booleans, integers, strings, and dates. These variables may contain access restrictions such as read-only and have their values auto incremented by the database. After completing the model, code can be generated from the MySQL Workbench to host it on the SQL server. Once the database is hosted, the web application is free to send and receive information. An alternative option is using a python module called sqlalchemy, its advantage is not requiring a server to connect to the database. The following paragraphs will provide further details on the components of the database.

The relationships between each table and variable are vital to the structure of the database. For example, the admin user controls all the functions of the Auto-Gardener but to do this he or she requires a microcontroller to handle the functions and a garden to apply the functions on. In the same sense, when an admin user is created an associated micro controller and garden is auto created in database for future use. These relationships from the admin to the garden and from the admin to the microcontroller will be called mandatory one to one relationship. Another type of relationship is a one to many relationship, where participation may be optional. An example of this in Auto-Gardener will be the relationship between the garden and the plants. The garden may contain anywhere from zero to a finite countable amount of plants and a table would not be created automatically in this case, it would require the user to manually input data.

Variable design is a factor as well. Each variable data type will have associated limits as to prevent accidents in user inputs, reduce data storage, and maintain consistency throughout the website and code. Each table does not exist as one instance of itself, it can be pictured more of as a row in a spreadsheet, while the variables are the columns in each row. In this way, the database can have an infinite number rows, but only a single value per column for each row. A primary key is a tool useful for defining a variable that is unique to a certain

table type for that variable. For example, the Plant table has the variable “plantID” that is the primary key for that table. This plant can now be uniquely identified by this ID, and all other variables can be looked up through this database by searching using that ID in the set of Plant tables. Another function of variable design is non-null variables. These set of variables provide a limitation on the user, so that their values must be defined when the table is created. This helps prevent issues related to missing variable values. Other ways limitations can help prevent bugs are options non-negative value variables. Pictured below is an Entity Relationship diagram. This shows the tables generated in the database as well as their associated variables. The arrows connecting each table the describe the relationship between them, either one to one, one to many, or many to one. Their relationships can also be mandatory or optional. In the future, permissions to view or edit information can be controlled by adding a non-admin table for customers of Auto-Gardener. Additional types of events such as seeding, fertilizing, and pruning can be added. Security features such as passwords, security questions, and emails are other future considerations for the database.

Figure 23 Entity Relationship Diagram of Application



5.8 MACHINE LEARNING DESIGN

Auto-Gardener requires machine learning to make decisions about weeds. By the end of the machine learning training, the goal is that Auto-Gardener will be able to recognize weeds from photos taken during the operating process, with an acceptable amount of accuracy. As machine learning is a broad field that can solve a wide variety of problems, it is important to specify the problem as much as possible then find algorithms appropriate for it. Thus, the foremost problem is image recognition and Auto-Gardener uses Machine Learning to solve this problem. Machine learning, in terms of our project, can be described as a process of using labeled data and an algorithm to extract information from that data in order to train a model to make predictions on unseen and unlabeled data [98]. The model has the characteristics of improving its performance with more data, this is called the training process [98]. The Auto-Gardener will use this trained model to make predictions on weeds. More specifically, Auto-Gardener uses supervised machine learning and the type of problem addressed is commonly called a classification problem because it aims to classify the images between different on labels provided to the machine, the weeds or non-weeds in this case [98].

This project will be using a dataset of weeds that will be split into a combination of training and testing data. After confirming the accuracy of the tests, then we can test the model on actual images taken by the device's camera. To ensure consistent results between training

and tests, we must ensure that the images taken from the dataset are similar to photos Auto-Gardener takes during operation. Since the camera will be raised above the planting bed, the view the Auto-Gardener receives will be a top-down and consist mainly of the leaves of any plants growing. Since it is expected that this device will operate outside, considerations for lighting will play a key role. This can be mitigated in-part through programming, limiting when the camera can be active. Ensuring optimum light levels for each picture. Since dataset photos are taken in a controlled environment and from varying angles, we would only need to filter the dataset for top-down viewpoints.

5.8.1 Related/Similar Software Projects

In this section, we will discuss related previous or on-going projects that solve weed detection using machine learning. Many of the examples described here use a deeper field of machine learning called deep learning. The main difference between deep learning and machine learning, is that in machine learning the programmer fine tunes and adjusts the parameters for algorithms for better accuracy while in deep learning the machine makes these decisions on its own [99]. The consequence of having a “smarter” machine in deep learning is that it requires an extra layer of structure the “artificial neural network” and is therefore more complex with added layers, it also requires larger amounts of data to train, which might be much more than we have available [99]. Another slight issue would be that the training is self-contained much like a black box. The algorithms in deep learning are also separated into layers, so the machine may choose to alter these in any way it wants [99]. Therefore, we would not be able to explain or see how the model detects weeds after training, we just know it does and can only use accuracy as a measurement. Since we cannot explain how it works, we cannot test it versus edge cases that would trick the machine into detecting a plant as a weed. However, the possibly lies that we discover methods to generate additional images and methods of shortening training time so that it does not require as much time.

In a paper: A Review of Application of Deep Learning for Weeds and Crops Classification in Agriculture the authors study various applications of deep learning and compare their effectiveness and methodology. The paper also mentions three classification methods to solving weed detection problems, “Color and threshold-based approaches suffers loss in accuracy when the light is too high or too low, learning-based approaches promise more precision on the other hand [100].” As the source of our dataset might be unreliable, it might be best to follow their reasoning and go for a learning-based approach method for machine learning. Of the multiple methods of deep learning that the paper reviewed, the most successfully accurate deep learning approach used semantic segmentation on CNN to detect sugar beets with 10,000 images in their data set [100]. The reported accuracies varied from 90% to 94.74%, with them increasing in correlation to the number of images used [100]. This makes sense, as deep learning requires lots of data and time, it will be useful to study the usage of semantic segmentation to improve our accuracy in the future. The inputs to the sugar beets classification were 512×384-pixel images and training time was around 48 hours [100].

Yang and Sun study another application of deep learning to weeds in A Survey on Deep Learning in Crop Planting, they highlight multiple studies that yield high accuracy rates of over 94% using convolutional neural networks (CNN) [101]. Convolutional neural networks are one of three common deep learning architectures, the other two are deep belief networks (DBN) and recurrent neural networks (RNN) [101]. An example visual representation of how CNN's are commonly constructed for detection of pests is shown below in Figure 25. In the image shown, there are multiple types of layers in convolutional neural networks such as a convolution and pooling layer, these layers are stacked upon each other to create the entire structure. This study highlighted the effectiveness of CNN in image recognition, showing high accuracy rates in training. Thus, the use of CNN's in combination with deep learning should be a priority in the discussion of algorithms for Auto-Gardener.

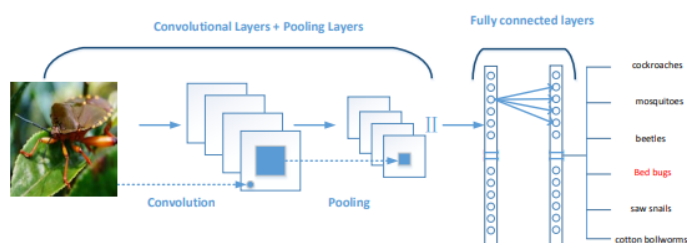


Figure 24 Common CNN Structure For Detection of Pests

In a paper titled Deep learning for image-based weed detection in turfgrass, the project tested weed detection on three different deep convolutional neural networks. Of these three two deep CNNs were designed for image classification and one for object classification, GoogLeNet and VGGNet were the former and DetectNet is the latter [102]. The journal authors “evaluated the feasibility of using DetectNet, GoogLeNet, and VGGNet for detection of annual bluegrass (*Poa annua* L.) or poa annua growing with various broadleaf weeds in dormant bermudagrass [102].” The results of the detection on these three types of weeds following will prove useful in the selection of our deep CNN structure, the authors used tested the deep CNNs and compared results with both detecting a single species and multiple species. Interesting details to note in comparison of our problem compared to turfgrass weed detection of multiple weeds is that:

- 1) Multiple species of weeds are tested at once simultaneously
- 2) Testing is made with images of other plants included in the same area(turfgrass)
- 3) Usage of labeled data-negative and positive labels for weeds and non-weeds

The process for comparing and training the three deep CNN to detect multiple weed species is described as follows from the paper [102]:

- 1) Images containing just a single species of weed were trained on the CNNs, they were split into two variants, the positive and negative labels. Each species had 6000 positive and 6000 negative images to compare to.
- 2) A portion of the image data set was reserved for validation and testing phase. Each contained 500 negative and 500 positive images for image detection and 50 positive/50 negative for object detection
- 3) Images are split into 9 pieces, each piece with size 640×360 , this is done using Irfan view software. This was used for image detection deep CNNs. The copies of those same images before the split were rescaled to 1280×720 for use in object detection
- 4) NVIDIA Deep Learning GPU Training System (DIGITS) was the software used for the deep learning task, the dataset was loaded there for each deep CNN
- 5) Super parameters and algorithms were selected, AdaDelta was chosen as the solver type and each deep CNN was trained with the same parameters
- 6) The model for each deep CNN was trained from the above steps, their results compared

After running the models, the team found that for all testing metrics: recall, precision, and F1score which is a mean of recall and precision, VGGNet outperformed GoogLeNet for both single and multiple weed detection [102]. Although training for object detection took longer, DetectNet outperformed both image detection deep CNNs VGGNet and GoogLeNet for all cases [102]. DetectNet did have a potential issue regarding detecting weeds on the edge of images with high weed density and that will probably present a challenge for Auto-Gardener as well [102].

| Framework | Core language | Institution | Features | Benefits | Cons |
|------------|-------------------------------------|------------------------|--|---|---|
| Theano | python | University of Montreal | computational graph, automatic differentiation | flexible, lots of libraries | slow compilation and runtime, non-distributed |
| TensorFlow | C++/Python | Google | computational graph, automatic differentiation | flexible, support visualization tools | slow, non-distributed |
| Caffe | C++/Python | BVLC | computer vision oriented | fast, architecture as a file | Support only CNN and MLP, hard to extend, non-distributed |
| Keras | Python | fchollet | High level neural network API | Easy to use, modularity, easy to extend | slow runtime, more memory occupancy |
| MXNet | R/Julia/C++/Scala/MATLAB/JavaScript | Amazon | computational graph, automatic differentiation | light weight, high, portability, easy to expand | Small community |

Figure 25 Open Source Deep Learning Frameworks. Requested permission for printing from Ming Sun

5.8.2 Machine Learning Algorithms

In this section, we will discuss multiple possible algorithms to train the weed detection model on. The purpose of choosing multiple algorithms is that they each will result in a trained model with varying amounts of accuracy. Thus, we aim to reduce the amount of error by either changing the parameters of an algorithm (super parameters) or by changing the algorithm itself.

From our research on plant-based weed detection methods in the previous section, the most successful way would be using a deep CNN with object detection to detect weeds in the Auto-Gardener area. This differs from our initial hypothesis that it would be an image recognition problem, but object detection is proven from one study to work better. Depending on the time frame we finish training this method, we may be able to try both types of algorithms to approach weed detection. Overall, in the implementation of our algorithm we should be wary of certain parameters. These are the types and sizes of filters, input size of images, learning rate decay algorithm, regularization, activation function, and the overall architecture. The reasoning for each parameter chosen will be discussed in further detail.

We will be using python as our programming language for machine learning. Python is supported by many users that have created functions useful for machine learning. The table below lists several open source frameworks that have modules and functions useful for the design of our machine learning algorithm [101].

Initially developed for evaluating matrix-based mathematical expressions, Theano is a python-based library and compiler, released in 2007. The oldest machine learning library discussed in this report, it can be used to analyze images although this was not the expressed goal during its development. This is possible due to being able to represent images as a matrix, or in computing terms: a 2D array. However, since Theano doesn't natively support image processing, the mathematical expressions needed to compare two images would need to be provided by the developer. At which point Theano is able to optimize those mathematical expressions and translate them into C++ for CPU processing or CUDA for GPU processing. In addition, Theano is able to take and optimize machine learning algorithms for better performance compared to implementation that involves those algorithm's default compiles. Overall, the usability of Theano for the needs of this project are limited as it would require the development of in-depth machine learning algorithms that would then be optimized by Theano. [103]

Developed by Google and released in 2017, TensorFlow is an adaptable multi-language, multi-platform machine learning library. This flexible architecture allows TensorFlow to run on most all operating systems including mobile devices and on varying amounts of hardware from single devices to multiple servers. From a developer standpoint TensorFlow is much easier in its implementation, in part due to Google's release of several training resources and due to the amount of sample projects available. However, TensorFlow's architecture is more difficult to use, meaning it can be more difficult to tweak specific settings. TensorFlow can also make use of other machine learning algorithms such as Keras which it is able to handle at a high-level of abstraction to make development easier. [104]

Acting as an interface to artificial neural networks, Keras released in 2015, was designed to enable fast experimentation. Keras has the advantage of acting as an interface for not only TensorFlow and other standard neural networks, but also the ability to support convolutional and recurrent neural networks. [105] Keras succeeds in bringing simple and concise functions, with little to no debugging needed on the neural network. Unfortunately, Keras is optimized for smaller datasets, which means that the networks generated may not be as accurate as others. This also limits the speed at which Keras can run as well, due to its extra features being slower than TensorFlow alone. [106]

5.8.3 Weed Detection Approach

Our primary task is to first identify the region in an image where a possible weed is located. Then the next step would be to classify it to be a weed or non-weed(crop). In this case there would be three different classes our AI would need to distinguish from, the weeds, crops, and the background. We will formulate a plan to find a dataset of at least weeds, then we

can work up to either distinguishing between that weed and our selected crop. Other ways we can do this is by detecting all plants in the garden, then classifying weeds based on location instead of trained characteristics using the AI. This would allow us to train the AI on only a weed dataset and minimize loss from determining from two different classes. The final detector achieved this by training an object detector on three classes, Garden Sage (*Salvia officinalis*), Red Lettuce (*Lactuca sativa*), and random weeds in the surrounding area. Certain species of plants are selected so that the detector can be trained on a variety of weed species while reducing the chances that it detects desired plants in the garden as weeds.

Due to our constraints of having no GPU to train the AI and limited budget to pay for processing power, we will also keep in mind to choose an architecture that works well with our dataset. Due to the availability of certain free GPUs such as Google Collab and Kaggle, our main constraint will be the dataset. In order to use an object detector, we would need to collect images from the internet and then manually annotate it using some kind of software. Next, we would have to save that in an annotation format that corresponds with the object detector system. In the next section, we propose a method that avoids this.

5.8.4 Dataset Collection

Dataset collection would be the largest constraint of weed detection besides computation power. In order to successfully test the AI specification, we would need to collect photos of a certain weed and then plant the weeds ourselves.

The ideal requirements for the dataset would be an accessible type of weed and plant we can grow/test and provided annotations of bounding boxes in either xml, csv, or text format. The number of images would be at least 100 and we can use data augmentation to overcome issues from a small dataset. From our research, *A survey of public datasets for computer vision tasks in agriculture* provides a list of 15 public datasets for weeding use [107]. Limiting the datasets down to ones with both over 100 images and annotated images at pixel level, we have 1) Grass-Broadleaf, 2) Synthetic Sugar beets, 3) Joint Stem Detection, 4) Open Plant Phenotype Database (OPPD) [107]. We also found one more useful dataset, *Dataset of annotated food crops and weed images for robotic computer vision control*.

Out of these datasets, we selected to consider between OPPD, synthetic sugar beets, and grass-broadleaf are most attractive due to their number of images. However, only OPPD has images of seedlings. Synthetic sugar beets only labels images semantically instead of by instance, and grass-broadleaf only provides masks of each plant. Thus, out of all publicly available datasets, none of them fit our project purposes. The decision was then to create a dataset that would match Auto-Gardener conditions, such as weeds and plants a certain distance above the ground and photos taken from top down.

For the dataset selection, a synthetic dataset was created due to lack of availability of large, annotated weed datasets for the detector to train on. 40 images of empty soil and ground in the surrounding areas were taken and resized to 2016x1512 resolution from phone cameras

at 16 inches in height. 10 pictures of garden sage, 10 pictures of red lettuce, and 99 images of weeds were taken using the same methods, but the resolution was maintained below 700 pixels. Each of the plant images were manually cropped out using GIMP to add transparency for image composition. This involved opening each photo Using python software, a random portion for the background was cropped out and a combination of 1-3 weeds, 0-2 of garden sage/red lettuce was composed onto the image. The portion size width and height were decided by the user input parameters. For the final version of the detector, the generated images were at 992x992px, but previous versions used 640x640px. Then additional image augmentation was performed randomly including rotation, flipping, brightness, and rescaling to the image to increase variability of the dataset. Each foreground of either red lettuce, weed, or garden sage was randomly rotated from 0-359 degrees and applied a random scale factor from 0.2-1.5x. Care was taken not to scale each foreground too because of increase in blurriness for upscaled images. For each plant foreground added, a RGB mask was generated onto another image. Using OpenCV, each different colored RGB mask was found by detecting the largest contour of that color. Then the program drew a bounding box around it, recording the classification, width, height, center x, center y and coordinates. These were written to an annotation file for each image. These annotations will be saved in darknet format, which takes four values describing a bounding box with ranges from zero to one. 3000 synthetic images were generated at 992x992px for training, validation, and testing in an 80-10-10 split.

5.8.5 Object Detection Models

In our object detection model selection, we chose between SSD(Single Shot Detector), Mask R-CNN, RetinaNet, and YOLO. All of these are well known architectures that have been used for multiple object detection problems [108]. They differ by a number of metrics, the relevant ones we will consider is frames per second, accuracy, and architecture. In general, two-stage detectors have high accuracy but are slower than one stage detectors for computation [109]. The accuracy is measured by comparing the performance to each dataset called mAP (mean average precision) on PASCAL/COCO datasets.

This section gives a overview of mAP, which is the standard metric that measures accuracy between different object detectors. To understand this, the metric to determine a correct prediction is necessary called IoU(Intersection over Union). IoU is defined as the quotient of the area of intersection over the area of union [110]. These involve the predicted bounding box by the detector and the ground truth box of the actual weed or plant. In this way, a higher IoU will have a more accurate prediction of location of a weed. So lower IoUs are used when the localization of weeds or plants is not as necessary. The detector will use an IoU of 50% for Auto-Gardener, meaning that a correct prediction has 50% or more of the predicted bounding box intersecting with the red lettuce, garden sage, or weed. Once IoU is determined, precision and recall can be calculated for different thresholds. The thresholds are the percentage confidence that a bounding box prediction is correct, which ranges from zero to one. Precision is calculated as the number of true positives divided by the sum of true positives and false positives [110]. In the detector case, given a labeled

bounding box, the percentage chance it is a true weed or plant is precision. Recall is calculated as the number of true positives divided by the sum of true positives and false negatives [110]. In terms of the detector, given a weed, the chance the detector draws a boundary box over it is recall. Maximizing both precision and recall at different thresholds leads to maximizing average precision. Average precision is equal to the area under the precision-recall curve [111]. This also comes out to be a number between zero and one, and the mean of the average precision across all classes that an object detector can detect is the mean average precision. To summarize, mean average precision is provided as a standardized metric to measure a detectors performance on maximizing recall and precision with a specified amount of localization.

Table 23 Comparison of Popular One Stage Detectors on MS COCO-2017 [108]

| Model | Size | FPS | AP_50 |
|-----------|------|------|--------|
| YOLOv4 | 512p | 31 | 64.90% |
| RetinaNet | 500p | 11.1 | 53.10% |
| SSD | 512p | 22 | 48.50% |

Mask R-CNN

For accuracy, Mask R-CNN does better as a two stage detector, but it performs much slower as a trade off. Mask R-CNN was implemented in the first version of the detector on Kaggle. We did not implement Mask R-CNN for this project because of difficulties implementing with the app. Also, the main repository is less user friendly as it does not provide libraries for outputting metric data, videos, or bounding box coordinates. The advantages of Mask R-CNN would be that in addition to drawing a bounding box on the plant, it would also segment the plant with a different colored mask.

One Stage Detectors

One-stage detectors are prioritized for this project because of their faster computation speed to be used in live testing. An advantage of using popular detectors is that they have pretrained weights on ImageNet, MS COCO, and PASCAL. These are large datasets with multiple classes, and the detectors have been trained on these with much larger computation power and time that we have resources to. For example, YOLOv4 was trained in 500,500 steps in the original paper to achieve its accuracy [108]. The detector in Auto-Gardener was trained on 6000 steps with 10 hours training time on the synthetic dataset, compared to the 834.17 hours it would have taken on the COCO dataset. In consideration of one-stage object detectors, Table 23 shows a comparison of three popular one stage detectors ordered by accuracy. At similar sizes, YOLO outperforms the other two in terms of accuracy and FPS. This secures the choice of YOLO for the implementation of the weed detector. In the following sections, we will describe how the other detectors work in brief.

All of them have pretrained weights on the COCO, which will greatly reduce both the required amount of time to train and number of images necessary. We can get a higher accuracy right off the bat because COCO dataset has a couple of objects it detects very similar to plants and knows some basic features to determine them.

Single Shot MultiBox Detector (SSD)

SSD was published in 2015 as a detector that surpassed the version of YOLO and two stage detectors in the RCNN family for its time [112]. SSD has a backbone CNN like other one stage detectors, such as VGG-16 for example. It uses a combination of multiple bounding boxes per region, small 3x3 convolutional filters for classification, and non max suppression on final detections [112]. For each location, SSD predicts an object by applying fixed sized bounding boxes on different feature maps and scales [112]. Although SSD proved to be a better detector in its inception, the accuracy and computation of YOLOv4 has surpassed this system, so it will not be implemented for Auto-Gardener [108].

RetinaNet

RetinaNet applied a novel concept to one stage detectors called focal loss [113]. It also uses a backbone called a featurized image pyramid in its architecture [109]. Focal loss solves an issue an object detectors “in which there is an extreme imbalance between foreground and background classes during training [113].” This imbalance is proven to interfere the cross entropy loss that a normal object detector is trained on, therefore minimizing this loss would optimize training and lead to increases in accuracy [113]. The featurized image pyramid uses layers of convolutional layers in a sequence of pyramid levels [109]. The connections between these layers are separated into two types of pathways, bottom-up pathways and top-down pathways [109]. The image starts at the bottom pyramid level, and the bottom-up pathway follows this sequence to the top of the pyramid for computation [109]. The top-down pathway goes in the opposite pathway, using upscaling and deconvolution to update the feature maps [109]. These layers of the pyramid are finally merged into a feature map and a prediction is performed after a 3x3 convolution is applied [109].

You Only Look Once Version 4 (YOLOv4)

YOLOv4 is composed of the major components, the backbone, neck, and head [108]. The image is first fed into the backbone which consists of a convolutional neural network (CNN) CSPDarknet53, a neck with SPP and PAN, and head YOLOv3 [108]. An input image is fed to the backbone, which then extracts multiple features of the image in feature maps as it passes through the network. The neck connects and mixes feature maps from backbone output and passes them to the YOLOv3 head for prediction [114]. The head draws the bounding boxes and labels as the final output, displaying the percent confidence and classification for each. These bounding boxes are generated from anchor boxes. Anchor boxes are bounding boxes with predefined sizes, YOLOv3 has three different sizes of these [115]. The detector will choose the anchor box with the most

suitable score or IoU as a bounding box [115]. The head also uses a total of three different scales to make predictions on an image, the sizes of these scales differ on the input size settings [115]. These allow for YOLO to increase its detection rate on smaller objects.

Another advantage of using YOLO is that it is a more popular framework that has been updated in the last year. It has up to 5 versions now, with each version improving on either accuracy or the speed of calculation [116]. Two of these include YOLOv5 or PP-YOLO as arguable upgrades. YOLOv5 implements YOLO on Pytorch, which greatly decreased the size of the weights by almost 90 percent and decreased computation time [116]. On a Tesla P100 GPU, YOLOv5 outperformed the frames per second of YOLOv4 by 2.8 times, from 50 to 150 [116]. PP-YOLO improves on the detector, with increases in both mAP and inference speed by changing the backbone from Darknet53 to a ResNet backbone and additional super parameters [117]. By using YOLO detectors, the project could use a similar dataset and not have to convert annotations to another format. These also all provide functions to detect on both images and videos, and periodic calculations for mAP based on number of steps.

6 PROTOTYPE CONSTRUCTION AND CODING

This chapter is briefly describes the bill of the materials, the prototype PCB, how we will build the project, and the final coding plan.

The table below shows the bill of materials. Most of the parts were purchased through Amazon and eBay. Some of the insignificant parts, such as jumper wires, resistors, and capacitors, were omitted from this table, however, were purchased. **Error! Reference source not found.** shows the items that were purchased and received so far. Most of the items have been ordered from the Table 24. The items that have not been purchased were the nontechnical items such as soil and plant seedlings. So far, the project is working within the projected budget, leaving room for any future unexpected costs.

Table 24: Bill of Materials

| Item | Part Number | Manufacturer | Cost per Part | Number in Design | Subtotal |
|----------------------------|-------------|--------------------|---------------|------------------|----------|
| Microcontroller | ATmega328P | Atmel | \$2.97 | 1 | \$2.97 |
| Crystal Oscillators | - | N/A | \$0.35 | 1 | \$3.49 |
| Stepper Motors | 17HS4401S | Ugoshine | \$8.99 | 5 | \$44.95 |
| Ultrasonic Position Sensor | 8541583025 | DORHEA | \$1.90 | 3 | \$5.70 |
| Water Pump | - | | \$10 | 1 | \$10 |
| Soil Moisture sensor | - | Catnip Electronics | \$13.00 | 1 | \$13.00 |

| | | | | | |
|---|------------|------------------|----------|---------|----------|
| Motor Driver BIQU A4988 | BIQU A4988 | BIQU | \$1.90 | 4 | \$7.60 |
| Camera | | Amazon | \$10 | 1 | \$10 |
| Raspberry Pi | | Rasberry Pi Shop | \$60 | | \$60 |
| Limit Switch | - | Xtension | \$5.50 | 2 extra | \$11 |
| Gantry and Frame | ACRO 510 | Open Builds | \$294.49 | 1 | \$294.49 |
| Water Tank | - | Home Depot | \$5.00 | 1 | \$5 |
| Z-axis aluminum rod | 47065T101 | McMaster | \$7.79 | 1 | \$7.79 |
| Water tubing | | Home Depot | \$3 | 1 | \$3 |
| Level Shifter | | Amazon | \$6 | 1 | \$6 |
| Aluminum Legs (1 in. x 96 in. Aluminum Angle with 1/16 in. Thick) | - | McMaster-Carr | free | free | free |
| Seedlings/plants | - | Lucas | \$5.00 | 4 | \$20.00 |
| Versions of PCB | | JLPCB | \$25 | 2 | \$50 |
| | | | | Total | \$500.33 |



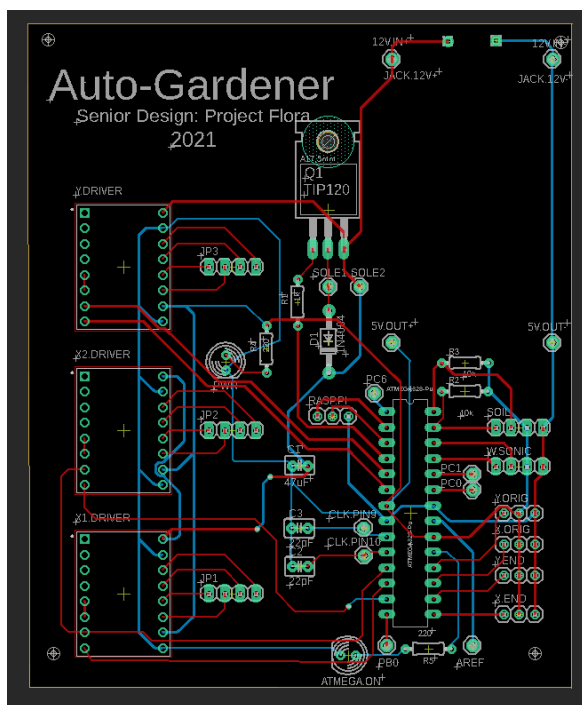
Figure 26: Current purchases that were acquired so far

The first physical aspect that will be built will be the frame and gantry. The frame and gantry will be built following the video instructions from Open Builds. An advantage of purchasing the Open Builds design is the lack of need for having a contractor or certain machines needed to cut and drill into metal. How it will be built was described in Chapter 5. The only difference between our frame design and their frame design are the legs which will prop up the design.

The ultrasonic sensors will be attached by holders which came with the package. It is shown below the ultrasonic sensors in Figure 17. A hole for the solenoid will be drilled into the side of the water tank. The soil moisture meter will be placed in the middle of the garden. Everything will be wired according to the schematics.

Figure 27: Final PCB Layout

6.1 PROTOTYPE PCB

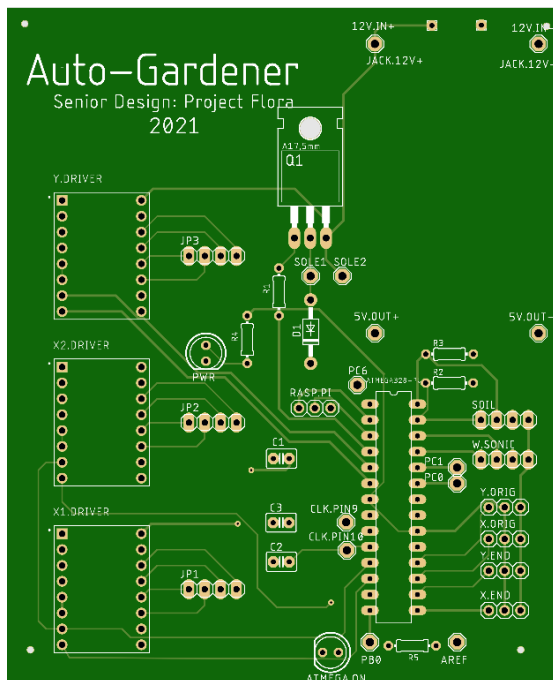


The Auto-Gardener will contain two PCBs. The PCBs will be manufactured by JLCPCB. JLCPCB has the option of preassembled components. We may take advantage of this for micro components such as the Wi-Fi chip. Otherwise, we will solder it on using flux. We will sweep solder the chip on and wick up the excess.

The PCBs will allow for all electrical connections for the listed devices as well as the future motor driver for the weeder tool. It will be powered by the power adapter connected to the power module. Figure 28 and 29 shows the final printed bread board design. The PCB includes the following:

- ATmega microcontroller
- Motor Drivers
- Connections for Raspberry Pi
- Connections for sensors
- Connections for water pump

Figure 28 Final PCB Gerber File

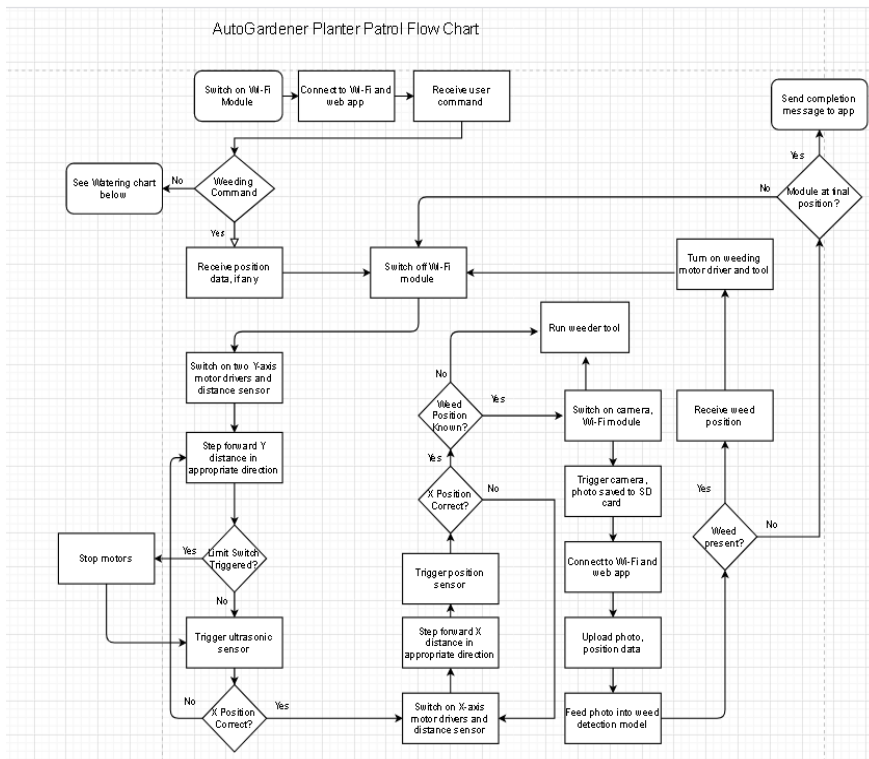


6.2 FINAL CODING PLAN

The Auto-Gardener will have two main complex functions: planter patrol and watering. The microprocessor will first wait for a command from the user to perform these functions. Then it will use a combination of software functions to perform either task. At a minimum Auto-Gardener will be required to connect to the web application using Wi-Fi to receive a user command. Planter patrol will involve the movement, position tracking, limit sensing, weed detection, database storage, camera, and SD card functions. Watering will involve a combination of water level, moisture sensor, and solenoid functions. These commands will also be tracked in the web application. Most of the data will be stored in the database for future use.

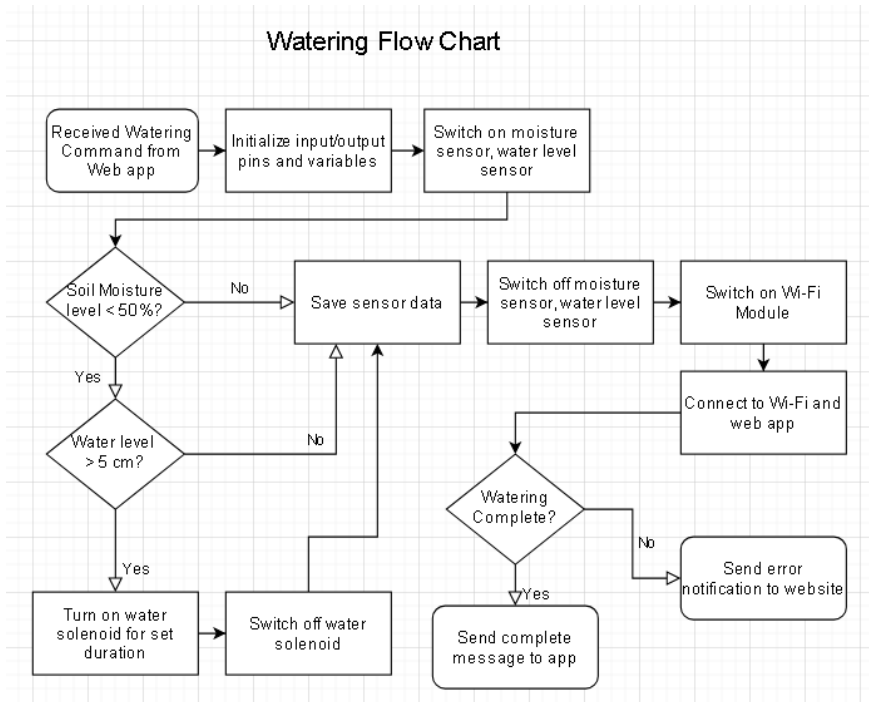
Planter patrol is the process of navigating the Auto-Gardener area and performing tasks. These tasks include taking photos, saving photos, sending the photos via Wi-Fi, and using the weeder. Before these tasks can be performed, the device must move to the proper X and Y position. It will do this by calculating the distance that the motor drivers will move given position info. The position sensors will ensure that the position is accurate, if not the motor drivers will have to be triggered again. Once the proper position has been reached the tasks will be completed. The microprocessor will take, save, and send those images to the web application and the weed detection process will begin. The device can either proceed to the next position and manage all the discovered weeds later or wait on the model to detect and send coordinates of the weeding position. After each movement is completed, Auto-Gardener will check whether the final position has been reached, if so, it will notify the web application with the proper details of the event. The flow chart below shows the flow chart of the Planter Patrol process in software.

Figure 30: Auto-Gardener Planter Patrol Flow Chart



The watering process is less complicated than planter patrol because it does not require movement of the Auto-Gardener module. After receiving the watering command, the system will be required to make two checks. If these conditions are not met, Auto-Gardener will skip the watering process. The device will save sensor data and send feedback to the user via web application regardless of completion. The conditions are the specified soil moisture level and water level. Both can be adjusted by the user. The sensors connected to the microprocessor will output moisture levels and calculations of water volume/water level back to be saved. After both conditions are met, the microprocessor will switch on the water solenoid for the specified amount of time. Once this duration has elapsed, it will send the completed message and sensor data to the web application. A flow chart below shows the watering software process of Auto-Gardener in detail.

Figure 29 Auto-Gardener Watering Software Flow Chart



7 PROJECT TESTING AND PROTOTYPE TESTING

Chapter 7 describes both the hardware and software testing. Each are inherently intertwined with each other, but we treated the testing separately as much as possible to ensure that the hardware was not an issue. Otherwise, the software may need to be adjusted. Any of the tests confirm for functionality and performance.

7.1 HARDWARE TESTING ENVIRONMENT

The hardware testing environment will be a mixture of indoors and outdoors. Indoor environmental testing refers to conducting tests in ‘laboratory’ conditions. Since senior design is online, we do not have easy access to actual laboratory conditions. Outdoor testing refers to conducting tests during ‘real life’ conditions for determining the ruggedness and how water resistant the design is. The hardware testing environment includes the following equipment:

- A DC Power Source Oscilloscope and Logic Analyzer: Analog Discovery 2 Pro Bundle Manufactured by Digilent, SKU: 240-123
- A Digital Oscilloscope, Waveform Generator, and Multimeter: Hantek 2D72 3in1

7.2 HARDWARE SPECIFIC TESTING

In this section, we define a specific technique to test a certain hardware component of Auto-Gardener, it will contain descriptions of the technique objective, testing tools, acceptance criteria and technique processes. We will discuss the primary and initial testing done for every component of Auto-Gardener. These general procedures will differ for components like wires and sensors. Each component should be tested before connecting with other components to ensure minimum error.

Some hardware components will not pose any issues during general hardware or specific hardware testing, but once their output is measured and sent to software there might be new issues present.

7.2.1 Hardware Testing Overview

Overall, the hardware testing will foresee power usage, mechanical testing, and hardware testing. This section is organized by the parts selection. The following describes a brief overview of how these parts will be tested to provide working verification.

Power Testing:

- We must ensure that every component is properly powered.

- We must ensure that every component is not getting too much or too little voltage or current, else the electronics will be shorted.
- We must ensure that the operation of the entire system simultaneously

Mechanical Testing:

- We must ensure that each part dimension conforms with its description.
- We must check for any loose parts.
- Secure those parts.
- We must ensure the timing belt is neither too tight nor too loose.
- We must ensure the motors can handle the load.

Hardware Testing:

- Multimeter tests:
 - Continuity test
 - Resistance test
 - Voltage test: AC or DC
- We must ensure the hardware can perform until normal conditions.

7.2.2 ATmega Testing

The preliminary testing procedure before the robotic garden is built should be as follows:

- 1) Install the bootloader of Arduino into ATmega328.
- 2) Write a program to test if an LED is blinking.
- 3) On a breadboard, create a circuit of the ATmega328, resistors, an LED, clock crystal, and capacitors.
- 4) Run the code to check each I/O pin. Write the results in Table 24.

| Pin | Does the LED Blink? Y/N | Pin | Does the LED Blink? Y/N |
|---------------|-------------------------|----------------|-------------------------|
| Digital pin 0 | Y | Digital pin 10 | Y |
| Digital pin 1 | Y | Digital pin 11 | Y |
| Digital pin 2 | Y | Digital pin 12 | Y |
| Digital pin 3 | Y | Digital pin 13 | Y |
| Digital pin 4 | Y | Analog pin 0 | Y |

| | | | |
|---------------|---|--------------|---|
| Digital pin 5 | Y | Analog pin 1 | Y |
| Digital pin 6 | Y | Analog pin 2 | Y |
| Digital pin 7 | Y | Analog pin 3 | Y |
| Digital pin 8 | Y | Analog pin 4 | Y |
| Digital pin 9 | Y | Analog pin 5 | Y |

Table 25: ATmega328 Preliminary Testing Results

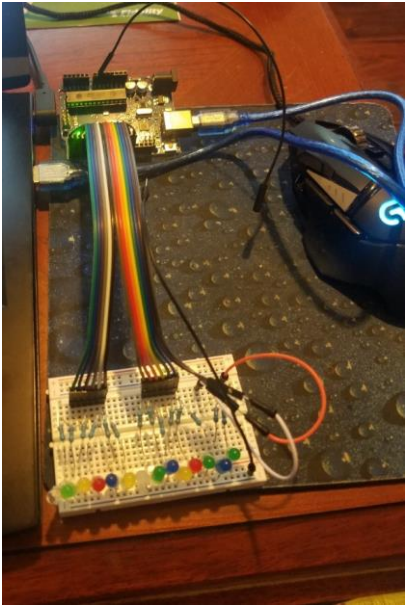


Figure 30: Preliminary Testing Setup

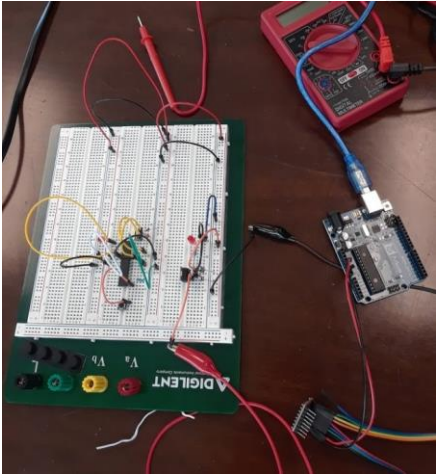


Figure 31: Testing the ATmega microcontroller

The difference between Figure 30 and Figure 31 is that Preliminary Testing Setup tests the individual pins of the ATmega. The figure on the right is testing the part of the schematic (see Chapter 5) that incorporates the external clock crystal which acts as a clock for the microcontroller and the step-down transistor. A reset button was added to this breadboard

circuit as well, but it is not required for our design. The circuit on the right was also testing the for installing the Arduino Bootloader.

For installing the bootloader, we first had to download the Arduino IDE and set up the circuit. The SCK, MISO, MOSI, and SSN (or the reset pin on the ATmega328p) must be matched. Next, we programmed the Arduino as an ISP, an in-system programming. There are built in examples of this in the Arduino IDE. Finally, we burned the bootloader.

7.2.3 Stepper Motors Testing

The next test ensures that both the ATmega and motor driver are working to signal the stepper motor to start working.

The testing procedure should be as follows:

- 1) Make sure ATmega, motor driver and stepper motor are properly connected.
- 2) Ensure signal pins on ATmega are active at the proper timing interval.
- 3) Ensure driver pins on motor driver are active at the proper timing interval.
- 4) Verify voltage output of the motor driver.
- 5) Ensure that stepper motor is active and in motion at the proper timing interval.

7.2.4 Stepper Motor Resistance Test

- 1) Unplug the stepper motor from the motor driver.
- 2) Measure the resistance of each coil in the stepper motor. Compare the readings to the specification sheet. The average of the readings needs to be less than 14% of the specified resistance rating.
 - a. A low reading may suggest the motor has a shorted winding.
 - b. A high reading may suggest the motor has an open winding.
 - c. Both a and b require the motor to be replaced [118]

Table 26: Aimed Stepper Motor Resistance Readings and Results

| | Aimed Readings | Coil A | Coil B | Average | Percent Difference |
|-------------------|----------------|--------------|--------------|---------------|--------------------|
| Spec Sheet | 3.3 Ω | | | | |
| Motor #1 | | 3.3 Ω | 3.4 Ω | 3.35 Ω | 1.50% |
| Motor #2 | | 3.3 Ω | 3.4 Ω | 3.35 Ω | 1.50% |
| Motor #3 | | 3.2 Ω | 3.3 Ω | 3.25 Ω | 1.53% |
| Motor #4 | | 3.8 Ω | 3.7 Ω | 3.75 Ω | 12.77% |

7.2.5 Stepper Motor Position Accuracy Test

We confirmed the engineering specification of moving to a precision of 0.5 inches. The stepper motors have a step angle. 200 steps is one revolution.

The testing procedure was as follows:

- 1) Set the gantry to the same location twice. The input coordinates are measured in steps by the stepper motor.
- 2) The position is marked and the difference between the expected and actual values are measured.

Table 27: Results for the Position Precision

| Input Coordinates | Difference |
|-------------------|--------------|
| (0,1000) | 0.12(0.30cm) |
| (1000,2000) | 0.16(0.41cm) |
| (2000,3000) | 0.14(0.35cm) |
| (3000,4000) | 0.14(0.35cm) |
| Avg | 0.14(0.35cm) |

7.2.6 Ultrasonic Precision Testing



Figure 32: Testing the ultrasonic position sensors accuracy

The preliminary testing procedure should be as follows: To test the ultrasonic position sensors, layout a ruler. Place the ultrasonic sensor at the end of the ruler.

- 1) Place a straight-edged object at the end of the ruler, starting at a distance of 30 cm. Record the program's output.
- 2) Repeat the readings at the distances as shown in Table 24.

Table 28: Testing the Ultrasonic Position Sensors Results

| Distance | Reading |
|--------------|---------|
| 30 cm | 31 cm |
| 25 cm | 25 cm |
| 15 cm | 14 cm |
| 10 cm | 11 cm |

| | |
|-------------|------|
| 5 cm | 6 cm |
| 2 cm | 3 cm |

Table 24 shows the results that we took for the position sensors. The readings show that the current program works well, being only off by a centimeter. Only one of the readings was accurate. This shows that we will need to recalibrate the code by a centimeter, record what is read and take the average, or take the readings and throw out the data that is not accurate.

After the frame and gantry are built, test how well the system records the location of the camera/weeder. The procedure should be as follows:

- 1) Create a 0.3 m by 0.8 m grid of 2 cm spacings. The ultrasonic sensors cannot read more than 3 meters, so this is more than enough space.
- 2) Move the gantry to the origin. Record what the ultrasonic sensors read at (0,0).
- 3) Move the gantry to the x, y coordinates of (2, 2) cm. Record what the ultrasonic sensors read. Move the gantry to the x, y coordinates of (4, 4) cm. Repeat in a linear fashion until the gantry has reached the coordinates (30, 30) cm.

7.2.7 Water Level Testing Accuracy

We confirmed the engineering specification ensuring that the water level detected is within 15% of the actual water level.

The testing plan is as follows:

- 1) Pour a measured amount of water into the tank. We measured a gallon of water.
- 2) Record the sensor readout.
- 3) Compare the actual measurements, using the lines on the side of the tank as a guide, to the detected amount.
- 4) Repeat the test.

Table 29: Water Level Testing Accuracy Results

| Test # | Water in Tank (gallons) | Detected Amount (gallons) | % Error |
|--------|-------------------------|---------------------------|---------|
| 1 | 1.1 | 1.020 | 7.27% |
| 2 | 2.1 | 2.038 | 2.95% |
| 3 | 3.1 | 3.058 | 1.35% |

Table 30: Measuring the water level accuracy



7.2.8 Interfacing Web App with the System

We confirmed the engineering specification of ensuring data is communicated by Wi-Fi within 5 seconds.

The testing plan is as follows:

- 1) Send a user movement command through the Web Application. Simultaneously start the timer.
- 2) Measure the time, by a phone timer, from hitting send to the first movement.

Table 31: System Response Time Results

| Test # | Actual Response Time |
|--------|----------------------|
| 1 | 1.66 |
| 2 | 1.19 |
| 3 | 1.77 |
| 4 | 2.35 |
| Avg | 1.74 |

7.2.9 PCB Prototype Testing

The PCB prototype testing will be done through both hardware and software testing. Prior to the PCB being built, the circuit should be built on a breadboard. This allows the team to adjust the software and smaller parts as needed. All the components will be connected to the breadboard. They will be tested to see if they turn on and off, and if they can continue working properly in the new circuit. After the breadboard circuit has been tested thoroughly, we will move onto ordering the printed circuit board.

We will do continuity testing for voltage and current, if possible, for the PCB. Each of the components can be tested by testing if it is obtaining the appropriate voltage. The voltages should be within a close range by 10% of the expected output. We can test for voltage at through-hole leads, solder pads and vias, in addition to pins.

We will test for accurate DC-DC conversion for the power module. These voltage readings should also be within ten percent of the expected output. Some components such as the ATmega chip and the soil moisture sensor has a low power mode. We will test to see if the low powered mode does indeed have low current ratings within 15% as provided onto the datasheet.

7.3 SOFTWARE TEST ENVIRONMENT

In preparation for working with multiple members we will be saving and uploading our work to GitHub. GitHub is a well-known platform for working cooperatively on software projects. There are many useful features working on this platform would provide, mainly to compartmentalize changes within our code, so it is easier to work on code on the same time, but also highlight potential conflicts. GitHub also has a feature to merge branches and clone code, all these functions will be useful for our software testing methods and a suitable software testing environment. Furthermore, after testing each individual component, the entire software system can be tested by using GitHub's merge branch functions to combine code so that the application can be tested in its entirety.

Most of the general software testing process steps will consist of cloning or saving a backup copy of previous software then testing the software with any modifications. Many programming IDE's have options to view device memory or pausing a program during its execution, these methods will be more optimal to finding any bugs from unexpected outputs. As it is easy to run software and detect bugs, most of the time and resources spent will be focused on fixing these bugs, therefore good software testing for Auto-Gardener will give clues to the origin of the bugs including the exact line of code where there may be an issue.

A large issue in preparation of fixing code and software maintenance is code readability and documentation. GitHub will help greatly in this, that the code each programmer writes and is separated until the software system is merged like previously stated. The process of

running software sometimes cannot be started until code from multiple group members is merged, but we can allow the author of the code to fix their own code before this and ensure that expected outputs are stated by that author.

Fixing errors found during a software tests is more of an ambiguous issue. We can remove software, add new software, or adjust variables in order to fix these issues, it all depends on the nature of the error. It may be necessary to fix these issues in hardware rather than software, such as inaccuracy with sensors or blurry input/output from the camera being displayed.

7.4 SOFTWARE SPECIFIC TESTING

Various procedures to test certain software functions will be described here. These designs will differ nature because of the medium of testing, the programming language will differ as well as the outputs of different functions. We will either be able to see these outputs using the application, internal database server, or visually through the actions of Auto-Gardener. As discussed in software specific testing, software is easy to test but difficult to fix so the primary goal of software specific testing if it finds a bug is to give clues on what the issue is.

7.4.1 Microprocessor Testing Procedures:

Each of the significant functions will be separated into different python scripts or functions and then tested based on their specific roles and purposed outputs.

Connecting to Wi-fi and Communicating Information

Testing the microcontroller connects to Wi-Fi is as simple as downloading software that can determine which devices are connected to the same Wi-Fi. We are more concerned that the device can maintain this connection throughout the entire operation process, so that information can be communicated smoothly. Thus, sending and receiving data must be confirmed on both the web application and the microprocessor. As such, we will have to confirm the information packets we send reach the server and are parsed correctly. The information sent back must also be accurate so the microprocessor can perform the specified commands. This data will also be significantly large as the images sent via file transfer protocol (FTP) or directly to the website are in sizes of about 2 MB each.

Movement

We can confirm position tracking by using a measurement tool to measure the actual distance that the motors have traveled, we can also confirm this using the ultrasonic sensors. The distance we track in software is not completely accurate because even when

we tell the microprocessor to stop supplying the motors, they would need time to stop their movement [51]. Braking the motors would be destructive toward the circuit as well, we will have to make calibrations in order to determine the appropriate distance the motors have moved in software [51]. This software will be present on both the motor driver to correctly operate the motors and on the web application for the user to test.

The objectives of testing would be to ensure that the program properly communicates to Auto-Gardener the command to move, the distance moved is accurate, and the distance moved is in the correct direction. The expected results would be the confirmed position in both actuality and software, accurate to a certain degree of error.

The testing procedure for ensuring the movement software meets these objectives is as follows:

- 1) Keep track of the current location of the motors on the X and Y axis
- 2) Select a distance and direction for the X or Y axis that the motors will move
- 3) Run the movement software that will connect to the microprocessor to execute these commands
- 4) Confirm the motors have moved in the correct direction and distance by measurement

After this running this testing, we will need to discover at which step the error, if there is any, has occurred. It could be in the connection to the microprocessor or the motor driver software. Once the appropriate fixes are complete, we can repeat the testing procedures to ensure compliance with the expected results.

Position Tracking

As discussed, position tracking in software will rely on both the accuracy of the movement software and the ultrasonic sensor software. We will have software on the ultrasonic sensor that calculates position, but its accuracy would be affected by outside temperature change. The other sensor we must confirm Auto-Gardener has reached the edge of the X axis is the limit switches. Once we have confirmed that the movement software can communicate the proper direction to the motors, we can use the movement testing process to confirm position tracking for our sensors.

Taking, Saving and Sending Photos

This process will take a combination of software from the camera module, microprocessor, and SD card module. For the camera module and SD card, we will have to confirm that the photos are sent and saved properly. Then we can send these photos in a connection between the SD card and the Wi-Fi module to the external computer.

The testing objectives would be to confirm that the complete photo taken from the camera at the proper time, is sent from the camera to the SD card and from the SD card to an external computer via Wi-Fi using commands from the microprocessor. The expected result would be that the camera photo is correctly transferred and saved onto the external computer and database.

The testing procedure for this process on the microprocessor software would be as follows:

- 1) Ensure that power supply is connected to every component, connect to Wi-Fi
- 2) Enable power to the camera and signal to take a photo at the desired location
- 3) The camera module sends this photo to the SD card module, ensure that the software allows proper time for the camera to completely transfer this information
- 4) Check if there is a new photo in the SD card
- 5) Send the newest photo on the SD card to the Wi-Fi module with the details of the desired destination

As this software will depend on correct communication with multiple components, we will have to ensure no data is lost between each step.

Planter Patrol

This microprocessor function will allow for the planned usage of all the previous functions, so this function does not need a detailed testing process to confirm success. We just need to confirm two issues. One is that the parameters entered for each function: the desired positions for movement and taking photos are valid. The second concern we need to verify is that the software executes this combination of functions in the right order. It must also complete the entire patrol process and not end prematurely.

Watering and Moisture Sensing

The watering software will require the usage of the water level sensor, moisture sensing, water solenoid, and Wi-Fi module. This command will specifically wait for communication from the user, then the microprocessor can begin to check if actions are necessary. After this function is done, the microcontroller will specifically need to time and send the user of this information. Unlike the planter patrol function, watering and moisture sensing functions do not require sending any types of images, so the software will send information instead. We will have to confirm that the information communicated from the sensor is also valid at each step.

The testing objectives of these functions would be the verification of the communication between the microprocessor and the water level sensor, moisture sensor, and water solenoid is successful along with the transfer of the results of this info via the Wi-Fi module. The expected results are that the microprocessor can send accurate data in every scenario describing these functions.

The testing procedure for this process on the microprocessor software would be as follows:

- 1) Ensure that power supply is connected to every component, connect to Wi-Fi
- 2) Receive the command from the web application to perform watering
- 3) Communicate to the moisture sensor and receive data on the moisture level via I2C. If there is not enough moisture then continue, otherwise skip to step 6
- 4) Receive water level data from the ultrasonic sensor, if there is enough water then continue, otherwise skip to step 6
- 5) Turn on the water solenoid to distribute water, the interval to leave it on may vary according to the water level
- 6) Send information on the water solenoid operating duration, water level, moisture sensor data, and possible errors that occurred via Wi-Fi

This testing process will have to be repeated multiple times to confirm that each error is sent to be displayed to the user. We will also have to specify and label each type of information in order to ease the parsing from the client side.

7.4.2 Application Testing Procedures:

The application will have many of its functionality separated on different pages, but some pages may share the same functions. Most of this testing will confirm that the proper message has been sent to Auto-Gardener, the database, the user, or the Cloud AI. If Auto-Gardener is successfully able to connect to the web application, it will upload some data to the server. The application will be able to determine the user that is communicating to it is the device by this data. Most of the functionality in the application must ensure the client-side user commands reaches the database server. The Cloud AI will provide its own metrics for accuracy, precision, recall etc. We will be required to designate a portion of the image dataset as testing data to see the actual accuracy of the AI in practice. This may require growing our own weeds and taking pictures using the Auto-Gardener module.

8 PROJECT OPERATION

This section discusses how to use the product. The project operation chapter describes how to correctly use the device and all of its functions. Section 8.1 briefly examines the features of the Auto-Gardener. Section 8.2 explains how to properly use the Auto-Gardener. In essence, section 8.2 is the “owner’s manual”.

8.1 GENERAL INFORMATION

Auto-Gardener is essentially a robotic gardener with the ability to:

1. Measure the wetness of soil
2. Automatically water the plants
3. Determine the presence of a weed

8.2 USING THE DEVICE

After proper assembly of the Auto-Gardener device, the user will need to power on the device and then connect to the web application to view, edit, and use the rest of the functions. Once on, Auto-Gardener is capable of watering independently of the web application, but the user can utilize more specialized commands by adjusting features on the application.

Auto-Gardener will be turned on and off with a physical switch on the device. If the device is not ready to perform any functions yet, it will notify the user on the app. After performing each function, Auto-Gardener will return to the designated start area for the device. The start area will be the backmost and leftmost of the garden by default. The user may be required to charge the battery to perform any of its functions.

For Auto-Gardener to connect to the preferred nearby Wi-Fi, the user will need to send these details to the microcontroller. Then the user can select the SSID and enter the Wi-Fi password. There will be visual indication once Auto-Gardener is connected to the Wi-Fi. After Auto-Gardener is connected, the user can verify this once more by connecting to the website and seeing the connection status. The user can then choose whether to perform each of the operations on the garden or check the device history.

Before starting the watering option, the app should notify user if the watering tank has enough water to start the drip feed via the water level sensor, if not the user will fill the tank to the sufficient amount. The next step will be to run the moisture sensor to check if there is enough water in the soil. If both conditions are satisfied, then the watering operation will be started. Auto-Gardener will be able to perform other functions while watering is happening. However, the user cannot turn off the device while this operation is occurring to avoid prolonged opening of the solenoid. After the watering process is done, this should

be noted in the app as well as amount of water consumed and water status. Some details of the watering operation are noted in the history page of the web application.

Auto-Gardener will automatically go on its patrol path and livestream via Wi-Fi. The starting area will either be the front or back of the garden and leftmost depending on the user preferences. The device will then go from left-to right, then from frontward to backward for each patrol path. At any time during the livestream, the user can capture an image on the application. Then this photo will be automatically processed to the plant detector. Once the detection is complete, the weeds tab will show the most recent predictions after clicking the refresh button. This will list the number of garden sage, red lettuce, and weeds detected. The percentage chance of each plant predicted will be shown above the bounding box. The user has the option to use a drop down menu to select undetected photos or view previously detected ones.

To perform maintenance, the user can notify the engineers on the contact us page of the website. The user can enter contact credentials such as email or phone number to be notified of a way to resolve the problem. This would possibly be a future software fix or instructions to modify the hardware. This may also contain a list of frequently asked questions that is supplied after common issues with the device after deployment.

9 ADMINISTRATIVE CONTENT

This section describes some methods that we used to organize and manage the project in a group setting. It contains the milestones, budget, project roles, and an overall summary of the project. In the milestones section, we discuss how we divided the project into lesser parts and provided an estimated duration for each part. It also discusses briefly challenges during completion of major milestones. The budget section will summarize the financial data on the project. Project roles provides explanations of the delegated roles for each group member. Finally, the overall summary lists some goals, problems, and accomplishments that the group encountered as well as the potential of the project.

9.1 MILESTONE DISCUSSION

The major milestones were separated by subsystems of the project. The dates and duration are listed in the table below. Other duties that were posted on Trello for the team members to keep track of. Trello was mainly used to assign tasks to each individual member, if necessary, and to set specific deadlines.

For our first milestone the completion of the frame design, this will be constantly tweaked throughout senior design one until the completion of the prototype. We had multiple considerations for the method that the gantry moves about in all three directions of a cartesian grid. In addition to this, introduction of new parts would make us debate about which location the components would be placed and connected. This would also easily be the single most expensive part of our budget, since the costs of the combination of aluminum framing, nuts, bolts, wheels, and other parts that hold the frame together are not pleasant. These must be stable and well-constructed to ensure the weeding operation is accurate.

Our next big milestone would be the part selection, once we have the frame and gantry designed, we would have a better idea of ideal parts. We would have to make sure that parts we have chosen are interoperable. This was especially difficult for the microprocessor as we must choose something comfortable enough that we can program the Wi-Fi and camera with. Also, more complex parts would require additional effort on PCB design as well. Once all parts are selected, we can start designing the PCB to account for connections to those devices.

In the construction of the prototype, we will have planned for most of the functions on the microcontroller except for the weeding tool and weeding detection. We will begin with constructing the frame then adding the motors, sensors, and wiring.

Table 32: Project Milestones

| Milestone # | Description | Duration | Dates |
|--|---|----------|-----------------------|
| Senior Design 1 | | | |
| Research, Documentation, & Design | | | |
| 1 | Frame Design | | 10/2 - 11/1 |
| 2 | PCB design | | 12/5 |
| 3 | Research agriculture | | 11/1 |
| 4 | X and Y axis motor assembly and control | | 10/2 - 11/15 |
| 5 | Weeder Motor assembly and control | | 10/2 - 11/15 |
| 6 | Water nozzle assembly and control | | 10/2 - 11/1 |
| 7 | Camera integration | | 11/15-12/1 |
| 8 | Ethernet integration | | 11/1-11/11 |
| 9 | Flora image recognition | | 1/15/2021 |
| 10 | Camera to image server communication | | 1/15/2021-2/1/2021 |
| 11 | Hardware housing, waterproof design | | 2/1/2020-3/1/2021 |
| Project Report | | | |
| 12 | Table of Contents | 4 weeks | 10/5 - 11/13 |
| 13 | First Draft (60 pages) | 4 weeks | 10/5 - 11/13 |
| 14 | Finalizing Paper (100 pages) | 2 weeks | 11/13 - 11/30 |
| 15 | Final Document Due Date | | 12/8 |
| Senior Design 2 | | | |
| 17 | Build Prototype | 4 weeks | 1/11/2021 - 2/11/2021 |
| 18 | Master control program | 2 weeks | 3/15 - 4/1 |
| 19 | Testing & Redesign | 6 weeks | 3/1 - 4/21 |
| 20 | Peer Presentation | TBA | April 2021 |
| 21 | Final Report | TBA | April 2021 |
| 22 | Final Presentation | TBA | April 2021 |

9.2 BUDGET AND FINANCE DISCUSSION

The project has aimed around the projected budget so far. Section 6 describes the Bill of Materials in Table 24 in greater detail. Since the team achieved their goal of spending less than \$650, then it leaves room for the team for any upgrades, additional hardware, or any unexpected costs. Unexpected costs could be replacing a device which stopped working.

Another unexpected cost could be redesigning the PCB again. Since our project is self-funded, we are seeking to keep our spending at a minimum while hitting our goals.

9.3 PROJECT ROLES

Despite the fact this is a team project in which everyone supports their peers, roles were assigned to ensure that the milestones are achieved. Each of the team members also had different experience and areas of interest in the project. In alphabetical order by last name, the following discusses the roles of everyone on the team.

Wallace Borges

Wallace is a Computer Engineering student with a background in robotics. Wallace is working on the gantry framework design with Denver, as this is one of the most important aspects of the project. This incorporates movement along the x- and y-axis for the camera and watering module. He is also in charge of developing the web app and making sure it can integrate all controllable features of the device, including user motor controls, managing captured pictures, utilizing AI weed detection functionality, and controlling the water valve.

Seana Falvey

Seana is the sole Electrical Engineering student as well as the project lead in the team. Hence, her primary focus is the hardware of the project. She is working on the PCB design, making the electronics water resistant, and dealing with correspondence between the components and microprocessor. Her experience in taking care of numerous plants offers practical insight to the automated garden. She will be ensuring that the output data will match the plants' needs. As project lead, she does administrative tasks such as making sure everyone meets deadlines, facilitating meetings, taking meeting notes, and delegating tasks.

Denver Lau

Denver is a Computer Engineering student with a background in web development, robot vision, and databases. An ambitious student, Denver likes to be involved in a wide breadth of tasks. Denver is working with Wallace on the gantry design. Since he is taking a class in artificial intelligence, he is writing the software needed to recognize flora images. He is also directing his attention to writing the graphical user interface.

Rafael Smith

Rafael is a Computer Engineering student. Rafael is the jack of all trades in the team. He works well with both hardware and software. Rafael, the go-to man for decisions across the board, is working primarily on the machine learning, the Wi-Fi module, and the camera to image server communication.

9.4 LOOKING FORWARD/CONCLUSION

The vision of the team was to create an automated garden which eliminates the arduous work of gardening. Our project Auto-Gardener was effectively designed to help with this problem. Just meeting the specifications that we designed proved to be challenging and we met significant hurdles along the way. This section will discuss some of the hurdles encountered in the project as well as a summary of our progress and improvements we could have made. Since the project is currently only at the stage before prototype production, this section will be unable to summarize those portions of the project until that date has been met. Thus, the complete specifications of the project have not been met so far, but we have planned a design and parts that will meet all the requirements of the prototype.

In the project thus far, we have successfully designed a system capable of performing watering and weed detecting functions. The completed prototype will be able to do the watering function and part of the weeding function. The most difficult hurdles lie in our constraints. Originally, we had planned for multiple functions such as seeding and fertilizing, but our design would have to be more complex mechanically. Overall, we had to make multiple simplifications to our design because we did not have a mechanical engineer in our team and were unfamiliar with this area. Other major constraints would be the budget, time, and complications due to COVID-19. We required more time to research and less ease of using lab equipment on campus. Besides constraints, our project required many more components connected than we expected. We had to account for more pins for our microprocessor and research different selections for every component. In this portion of the project, we believe we can meet the required specifications without any alteration.

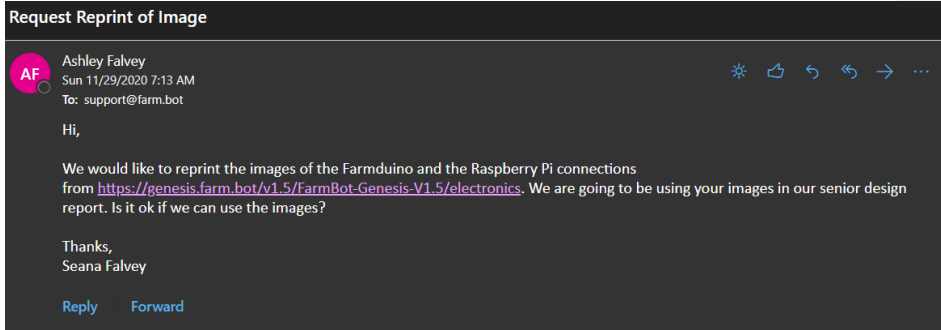
The design Auto-Gardener would be improved, with the knowledge of these problems beforehand. It could also be shifted in purpose to solve other problems. Our design only accounted for automation of tasks in a cost-effective and efficient manner. We did not consider appearance, quality, scalability, and other factors to make the project more marketable. In addition to this, the quality of our specifications would be enhanced with time. Advanced mechanical design could be added to account for more functions and ease of use. The field of machine learning is always improving as well as computation power, which would bring faster and more accurate weed detection.

Overall, our team successfully designed a device that can perform a portion of gardening tasks. The system reduces effort for a gardener to perform these tasks and is a few steps toward creating a fully automated garden. Only one user is necessary to manage the system and gardening area. Auto-Gardener can provide descriptive communication from its sensors on important statistics of plants and weeds in the garden. Commands are provided conveniently in the form of a web application and any of the activities of the device would be recorded to view at the user's leisure.

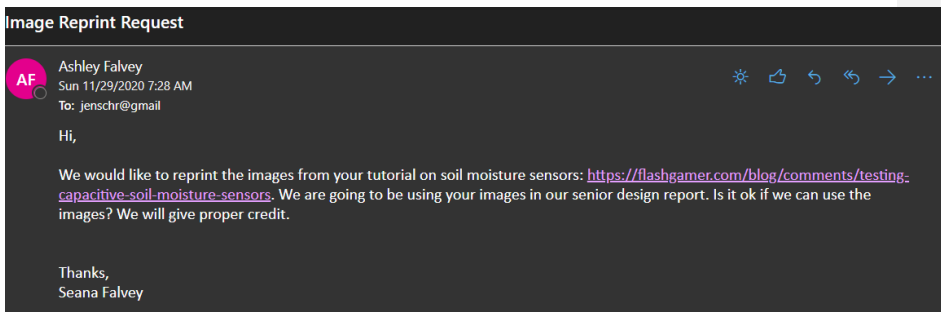
We have no doubt that with entirety of our project, the research, the design, and construction, there has been significant progress in the achievement of our goals

throughout. We are optimistic towards the future of the project and the potential of what it might become.

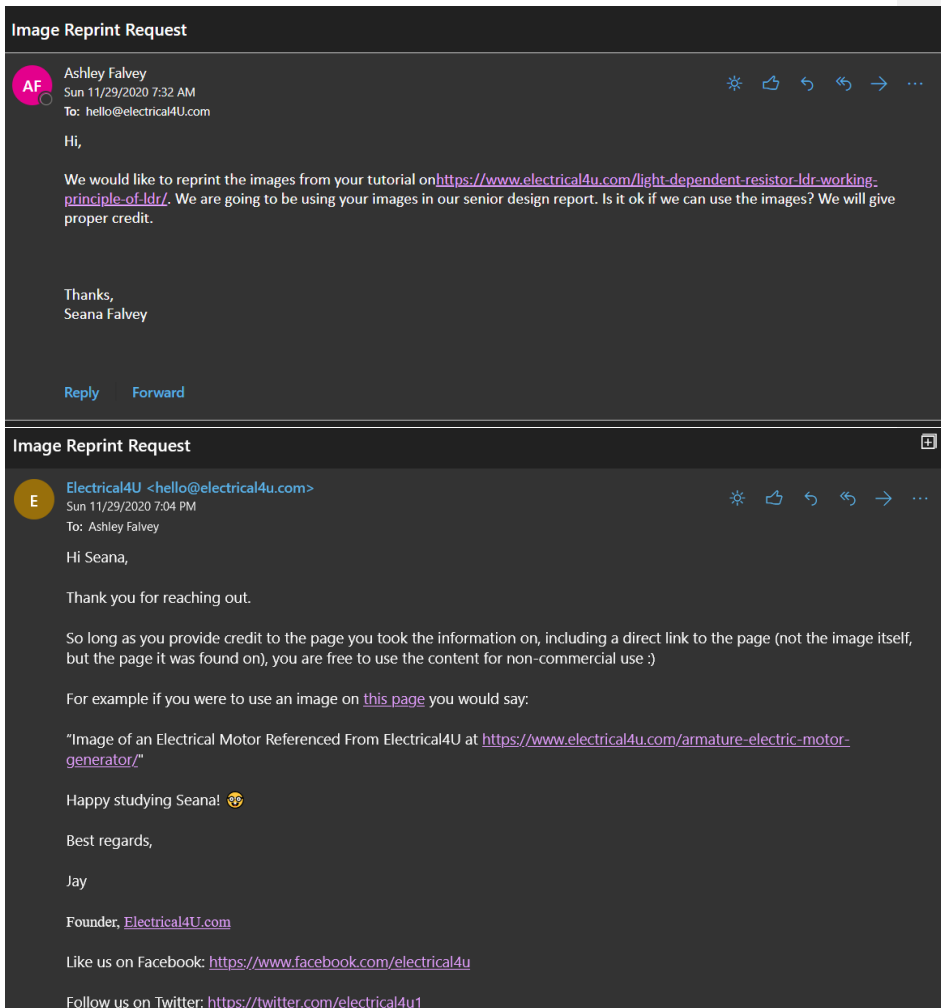
APPENDIX A COPYRIGHT PERMISSIONS



Farmbot Permission Request



Flashgamer reprint request



Electrical4U Request and Permission Confirmation

We're here to help.

Send your question below and we'll get back to you within the next 24 hours.

ashleysfalvey@knights.ucf.edu

Reprint Request

Hi,
I'm a student at the University of Central Florida. We would like to request to reprint your designs of the CAD models for the ACRO 510 in our senior design report. We will be providing proper citation that the works are by [OpenBuilds](#).

Thanks,
Seana Falvey

[SEND US AN EMAIL](#) CANCEL



We'll get back to you in the next 24 hours.

OpenBuilds Part Store <contact@openbuildspartstore.com>

Wed 12/2/2020 8:32 AM

To: Ashley Falvey

Hi Seana,

Thank you for contacting us. We are happy to help!

We appreciate you reaching out to us for permission to reprint the CAD files of our ACRO. You are more than welcome to reprint what you need for your Senior Project. We would love to also see your project once it is complete. You can tag us ([#openbuilds](#)) or send us at photo to contact@openbuildspartstore.com.

Good luck to you and thank you for your support!

Thank You,



OpenBuilds Team
Customer Support

OpenBuilds.com | OpenBuildsPartStore.com

Open Builds Request and Permission Confirmation



Lain (Marvel)

Nov 24, 2020, 10:38 GMT

Hey there,

Thanks for reaching out! I'm happy to help with this.

You can use your prototype however you like! Is there anything specific you needed help with for using the prototype in your senior design report?

Let me know if you have any questions!

Best wishes,

Lain

Senior Customer Support Manager

Marvel App Permission of Use



Christine von //SEIBERT/MEDIA GmbH <christine.bork@seibertmedia-gmbh.intercom-mail.com>

Mon 11/23/2020 3:42 PM
To: Denver Lau

Hi Denver,

there are no restrictions on the usage of any diagrams you generate with diagrams.net. The diagrams you create belong to you and we grant you a license to use any of our copyrighted icons in your diagrams and for making raster and vector versions of your diagrams and parts, using diagrams.net, for any purpose.

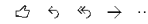
<https://desk.draw.io/support/solutions/articles/16000042494-usage-terms-for-diagrams-created-in-diagrams-net#:~:text=There%20are%20no%20restrictions%20on,.net%2C%20for%20any%20purpose>



Draw.io Permission of Use



孙明 <sunming@cau.edu.cn>
Mon 11/23/2020 10:28 PM
To: Denver Lau



No problem. It is my pleasure.

Ming Sun

-----原始邮件-----

发件人: "Denver Lau" <d.lau@Knights.ucf.edu>
发送时间: 2020-11-24 04:50:14 (星期二)
收件人: "sunming@cau.edu.cn" <sunming@cau.edu.cn>
抄送:
主题: A Survey on Deep Learning in Crop Planting Permission of Use

Hello,

My group and I currently are working on our senior design project at the University of Central Florida. Could you give us permission to use two images from your paper A Survey on Deep Learning in Crop Planting. These include figure 1 and table 1.

Thank you,
Denver Lau

Reply | Forward

A Review of Application of Deep Learning for Weeds and Crops Classification in Agriculture Permission of Use

RE: RE: Notification: Business Message from Mr. Wallace Borges



wallace.borges.35@gmail.com <wallace.borges.35@gmail.com>
5:10 AM



To: 外贸业务2

I am writing for a school project. I would like to request permission to use one of your photos in a design document. The photo is being used to show what type of aluminum frame is required for our future design.

The photo is in this url:

<https://anglealuminium.en.made-in-china.com/productimage/qpxsjjDYrHQ-2f1j00bMwtolCPfjhH/China-Aluminum-Extrusion-2020-2040-3030-4040-T-Slot-Linear-Motion-Open-Rail.html>

Please email me back on whether or not I have permission to use your photo in my report.

Thank you!

Requesting permission from LangLe Aluminum

APPENDIX B DATA SHEETS

A4988: <https://www.allegromicro.com/~media/Files/Datasheets/A4988-Datasheet.ashx>

ATmegas: <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf>

ESP8266: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

Limit Switch V-156-1C25: <https://datasheet.octopart.com/V-156-1C25-Omron-datasheet-23711649.pdf>

NEMA 17 stepper motor 17HS4401: <http://www.datasheetcafe.com/17hs4401-datasheet-stepper-motor/>

Ultrasonic Sensor HC-SR04: <https://datasheet4u.com/datasheet-pdf/ETC/HC-SR04/pdf.php?id=1380136>

APPENDIX C SOFTWARE

Autodesk Eagle (Schematic Design Software)

Circuito.io (Prototyping Raspberry Pi connections)

Discord (Meeting software)

Draw.io (Website for hardware/software design and diagrams)

GitHub (Group Software Development)

IrfanView (In consideration for resizing images)

Marvelapp.com (Prototyping for web application)

Microsoft Word (Documentation)

MySQL (Database and Database Server)

Nvidia Digits (In consideration for deep learning software)

Python (Application Backend, Microcontroller)

Trello (Milestone Organization)

Weather Website (To be denoted, to draw weather information on the application)

Zoom (Meeting software)

APPENDIX D WORKS CITED

- [1] L. Otto, "Landscaping with Native Plants," 2004. [Online]. Available: https://archive.epa.gov/greenacres/web/pdf/wo_2004b.pdf.
- [2] "Saving Water in Florida," 19 January 2017. [Online]. Available: https://19january2017snapshot.epa.gov/www3/watersense/docs/florida_state_fact_sheet_508.pdf.
- [3] G. Hillhouse, "GradyHillhouseGarduino," 30 August 2015. [Online]. Available: <https://github.com/gradyh/GradyHillhouseGarduino>.
- [4] "FarmBot Express," 2019. [Online]. Available: <https://express.farm.bot/v1.0/FarmBot-Express-v1.0/intro>.
- [5] "FarmBot Genesis," 2019. [Online]. Available: <https://genesis.farm.bot/v1.5/FarmBot-Genesis-V1.5/intro>.
- [6] "Basic Switches," [Online]. Available: <https://sensing.honeywell.com/switches/basics>.
- [7] P. Jain, "Position Sensor : Types of Position Sensor," 14 September 2012. [Online]. Available: https://www.engineersgarage.com/article_page/position-sensor-types-of-position-sensor/.

- [8] C. Woodford, "Lithium-ion batteries," Explain That Stuff, 23 November 2020. [Online]. Available: <https://www.explainthatstuff.com/how-lithium-ion-batteries-work.html>.
- [9] "Li-ion Battery," Virtual Amrita Laboratories Universalizing Education, 2013. [Online]. Available: <http://vlab.amrita.edu/?sub=77&brch=270&sim=1540&cnt=1>.
- [1] votlog, "How to Waterproof Your Electronics or PCBs," Instructables Circuits, 9 April 0] 2019. [Online]. Available: <https://www.instructables.com/How-to-Waterproof-Your-Electronics-or-PCBs/>. [Accessed 11 November 2020].
- [1] "All you need to Know about Heat Shrink Tubing and Sleeves," Components 101, 1] January 2020. [Online]. Available: <https://components101.com/articles/all-you-need-to-know-about-heat-shrink-tubing-and-sleeves>. [Accessed 27 November 2020].
- [1] J. Gelston, "Tips for Waterproofing an Electronic Enclosure," Aim Processing, 30 2] November 2016. [Online]. Available: <http://www.aimprocessing.com/blog/tips-for-waterproofing-an-electronic-enclosure>. [Accessed November 11 2020].
- [1] "Arduino Uno," Wikipedia, [Online]. Available: 3] https://en.wikipedia.org/wiki/Arduino_Uno.
- [1] Microchip, "ATmega640/V-1280/V-1281/V-2560/V-2561/V DATASHEET," 4] February 2014. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf>. [Accessed November 2020].
- [1] "MSP430G2553," Texas Instruments, [Online]. Available: 5] <https://www.ti.com/product/MSP430G2553#product-details##params>.
- [1] desaster, "OV7670-FIFO + MSP430 Launchpad," [Online]. Available: 6] <https://www.rpg.fi/desaster/blog/ov7670-fifo-msp430-launchpad.html>.

[1 "Raspberry Pi 4 Model B specifications - Raspberry Pi," Raspberry Pi Foundation, 7] [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/?resellerType=home&variant=raspberry-pi-4-model-b-4gb>.

[1 "odroid-c4 [ODROID Wiki]," Hardkernel, [Online]. Available: 8] <https://wiki.odroid.com/odroid-c4/odroid-c4>.

[1 "LattePanda 4G/64GB - LattePanda," LattePanda, [Online]. Available: 9] <http://www.lattepanda.com/products/3.html>.

[2 "Arduino Uno Rev3 | Arduino Official Store," Arduino, [Online]. Available: 0] <https://store.arduino.cc/usa/arduino-uno-rev3>.

[2 "ATmega328P 8-bit AVR Microcontrollers," Microchip, [Online]. Available: 1] <https://www.microchip.com/wwwproducts/en/ATmega328p>.

[2 "Power Supply - Raspberry Pi Documentation," Raspberry Pi Foundation, [Online]. 2] Available: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>.

[2 "Raspberry Pi 4 4GB," CanaKit, [Online]. Available: 3] <https://www.canakit.com/raspberry-pi-4-4gb.html?cid=usd&src=raspberrypi>.

[2 "MSP430G2553IN20 | Buy TI Parts | TI.com," Texas Instruments, [Online]. 4] Available: <https://www.ti.com/store/ti/en/p/product/?p=MSP430G2553IN20>.

[2 FarmBot, "FarmBot Genesis," 2019-a. [Online]. Available: 5] genesis.farm.bot/v1.5/FarmBot-Genesis-V1.5/tools/camera. [Accessed 2020].

[2 S. Gupta, "Deep Learning with High Resolution Images & Large Models," IBM, 12 6] December 2017. [Online]. Available: <https://www.linkedin.com/pulse/deep-learning-high-resolution-images-large-models-sumit-gupta/>. [Accessed November 2020].

- [2] "Orlando Temperatures: Averages by Month," Current Results, 2019. [Online].
7] Available: <https://www.currentresults.com/Weather/Florida/Places/orlando-temperatures-by-month-average.php>. [Accessed November 2020].
- [2] Arrow, "OV09724-A28A," Arrow Electronics, 2020. [Online]. Available:
8] <https://www.arrow.com/en/products/ov09724-a28a/omnivision-technologies>.
[Accessed 27 11 2020].
- [2] Waveshare, "OV2640 Camera Board," Waveshare, 2020. [Online]. Available:
9] <https://www.waveshare.com/product/ov2640-camera-board.htm>. [Accessed 27 11 2020].
- [3] T TAKMLY, "USB Endoscope for Otg Android Phone, Computer, 5.5 mm Borescope
0] Inspection Snake Camera Waterproof with Micro USB, Type C, 16.4FT Semi-rigid
Cord with 6 LED Lights, Compatible with Windows PC, MacBook," Amazon, 2020.
[Online]. Available: https://www.amazon.com/Seesi-Endoscope-Waterproof-Inspection-Semi-Rigid/dp/B07PBF6DX5/ref=sr_1_3?dchild=1&keywords=usb+borescope&qid=1606258166&sr=8-3. [Accessed 27 11 2020].
- [3] Waveshare, "ESP32-CAM, Camera Module Based on ESP32," Waveshare, 2020.
1] [Online]. Available: <https://www.waveshare.com/product/modules/cameras-audio-video/cameras/esp32-cam.htm>. [Accessed 27 11 2020].
- [3] Waveshare, "File:OV2640-Camera-Board-Code.7z," 4 8 2015. [Online]. Available:
2] <https://www.waveshare.com/wiki/File:OV2640-Camera-Board-Code.7z>. [Accessed 27 11 2020].
- [3] Rui, "Guide for Soil Moisture Sensor YL-69 or HL-69 with Arduino," Random Nerd
3] Tutorials, 2016. [Online]. Available: <https://randomnerdtutorials.com/guide-for-soil-moisture-sensor-yl-69-or-hl-69-with-the-arduino/>.
- [3] Vegetronix, "VH400 Low-Cost Soil Moisture Sensors," Vegetronix, [Online].
4] Available: <https://www.vegetronix.com/Products/VH400/>.

- [3 P. Taneja, "Soil Moisture Sensor MSE-I," 13 March 2016. [Online]. Available:
5] <https://www.slideshare.net/parveshtaneja9/soil-moisture-sensor-and-arduino>.
[Accessed November 2020].
- [3 J. C. Brynildsen, "Testing Capacitive soil moisture sensors," Flashgamer., 28 October
6] 2019. [Online]. Available: <https://flashgamer.com/blog/comments/testing-capacitive-soil-moisture-sensors>. [Accessed November 2020].
- [3 "Capacitive Soil Moisture Sensor Corrosion Resistant for Arduino Moisture Detection
7] Garden Watering DIY (Pack of 5PCS)," Amazon, [Online]. Available:
<https://www.amazon.com/Capacitive-Moisture-Corrosion-Resistant-Detection/dp/B07SYBSHGX>. [Accessed November 2020].
- [3 Adafruit, "Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor,"
8] Adafruit, [Online]. Available: <https://www.adafruit.com/product/4026>. [Accessed 26
November 2020].
- [3 Catnip electronics in Lithuania , "I2C Soil moisture sensor," tindie, [Online].
9] Available: <https://www.tindie.com/products/miceuz/i2c-soil-moisture-sensor/>.
[Accessed November 2020].
- [4 Electrical4U, "Light Dependent Resistor (LDR) or Photoresistor (What is it?),"
0] Electrical4U, 25 October 2020. [Online]. Available:
<https://www.electrical4u.com/light-dependent-resistor-ldr-working-principle-of-ldr/>.
- [4 L. Ada, "Photocells," Adafruit, 29 July 2012. [Online]. Available:
1] <https://learn.adafruit.com/photocells?view=all>. [Accessed 2020].
- [4 Token, "CDS Light-Dependent Photoresistors," 2010. [Online]. Available:
2] <http://yourduino.com/docs/Photoresistor-5516-datasheet.pdf>. [Accessed November
2020].
- [4 Advanced Photonix, Inc., "CdS Photoconductive Photocells: PDV-P8001," March 30
3] 2006. [Online]. Available: <https://datasheet.octopart.com/PDV-P8001-Advanced-Photonix-datasheet-127922.pdf>. [Accessed November 2020].

- [4 ROHM Semiconductor, "Digital 16bit Serial Output Type Ambient Light Sensor IC,"
4] April 2010. [Online]. Available:
https://www.velleman.eu/downloads/29/infosheets/bh1750fvi-e_datasheet.pdf.
[Accessed November 2020].
- [4 Celera Motion, "Position Sensors - Choosing the right sensor," Novanta, [Online].
5] Available: <https://www.celeramotion.com/zettlex/support/technical-papers/position-sensors/>. [Accessed 11 November 2020].
- [4 MTS Systems Corporation, "Magnetostrictive Linear-Position Sensors," 2006.
6] [Online]. Available:
https://www.controldesign.com/assets/wp_downloads/pdf/mts_sensors.pdf.
[Accessed 11 November 2020].
- [4 "Motor Buyers Guide," Jameco, [Online]. Available:
7] <https://www.jameco.com/Jameco/workshop/ProductNews/motor-buyers-guide.html>.
[Accessed 7 November 2020].
- [4 Caseytherobot, "Motors and Selecting the Right One," Sparkfun, [Online]. Available:
8] <https://learn.sparkfun.com/tutorials/motors-and-selecting-the-right-one/all#stepper-motors---simply-precise>. [Accessed November 2020].
- [4 Microchip Technology Inc., "ATmega48A/PA/88A/PA/168A/PA/328/P," Microchip
9] Technology Inc., 2020. [Online]. Available:
<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>. [Accessed 28 11 2020].
- [5 Singasong, "Usongshine Nema 17 Stepper Motor 42BYGH 1.8 Degree 1.5A 42 Motor
0] (17HS4401S) 42N.cm (60oz.in) 4-Lead with 1m Cable and Connector for DIY CNC
3D Printer (Pack of 3)," Amazon, 2020. [Online]. Available:
https://www.amazon.com/dp/B07KW6B3ZX/ref=twister_B089RBV837?_encoding=UTF8&th=1. [Accessed 27 11 2020].
- [5 Sam, "Motor Drivers vs. Motor Controllers," Core Electronics, 19 8 2019. [Online].
1] Available: <https://core-electronics.com.au/tutorials/motor-drivers-vs-motor-controllers.html>. [Accessed 28 11 2020].

- [5 Inventables, "Stepper Motors," Blogger, 2020. [Online]. Available:
2] <http://blog.inventables.com/p/stepper-motors.html>. [Accessed 28 11 2020].
- [5 AideepenAideepen, "Aideepen 5pcs 1.5A 2 Way DC Motor Driver Module Speed
3] Dual H-Bridge Replace Stepper L298N," Amazon, 2020. [Online]. Available:
<https://www.amazon.com/Aideepen-Driver-H-Bridge-Replace-Stepper/dp/B075S368Y2>. [Accessed 28 11 2020].
- [5 Pololu, "Stepper Motor Drivers," Pololu Corporation, 2020. [Online]. Available:
4] <https://www.pololu.com/category/120/stepper-motor-drivers>. [Accessed 28 11 2020].
- [5 Q. C, "Qunqi L298N Motor Drive Controller Board Module Dual H Bridge DC
5] Stepper For Arduino," Amazon, 2020. [Online]. Available:
<https://www.amazon.com/Qunqi-Controller-Module-Stepper-Arduino/dp/B014KMHSW6>. [Accessed 28 11 2020].
- [5 Maxim Integrated, "DS2438Smart Battery Monitor," Maxim Integrated, 2005.
6] [Online]. Available: <https://datasheets.maximintegrated.com/en/ds/DS2438.pdf>.
[Accessed November 2020].
- [5 Maxim Integrated, "Lithium-Ion Cell Fuel Gauging with Maxim Battery Monitor
7] ICs," Maxim Integrated, 29 March 2001. [Online]. Available:
<https://www.maximintegrated.com/en/design/technical-documents/app-notes/1/131.html>.
- [5 Fortune, "Datasheet DW01X One Cell Lithium-ion/Polymer Battery Protection IC,"
8] May 2014. [Online]. Available:
https://www.mpu51.com/mcucity/DATA_PDF/DW01x-DS-17_EN.pdf.
- [5 ShenZhen CanSheng Industry Development Co.,Ltd. , "FS8205A: Dual N-Channel
9] Enhancement Mode MOSFET," [Online]. Available:
<https://datasheet4u.com/datasheet-parts/FS8205A-datasheet.php?id=976676> .
[Accessed November 2020].

- [6 Shanghai Consonance Electronics Incorporated, "CN3722," Shanghai Consonance
0] Electronics Incorporated, 2020. [Online]. Available: <http://www.consonance-elec.com/seriesCN3722-E.html>. [Accessed November 2020].
- [6 Northern Arizona Wind & Sun, "What is Maximum Power Point Tracking (MPPT),"
1] Northern Arizona Wind & Sun, [Online]. Available: <https://www.solar-electric.com/learning-center/mppt-solar-charge-controllers.html/>. [Accessed November 2020].
- [6 RF Wireless World, "Advantages of LiPo Battery | Disadvantages of LiPo Battery,"
2] RF Wireless World, [Online]. Available: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-LiPo-Battery.html>.
- [6 M. P., *HC-SR04 Datasheet*, Electro Schematics.
3]
- [6 OmniVsino, *OV2640 Color CMOS UXGA (2.0 MegaPixel) CameraChip with
4] OmniPixel2 Technology*.
- [6 A. Bhatt, "Difference between Bluetooth and WiFi," WTW Media LLC, 15 5 2011.
5] [Online]. Available: https://www.engineersgarage.com/how_to/difference-between-bluetooth-and-wifi/. [Accessed 25 11 2020].
- [6 m. a. m. !. Memory, "Memory, memory, always memory !!!," WordPress, 13 1 2019.
6] [Online]. Available: <https://esp8266life.wordpress.com/2019/01/13/memory-memory-always-memory/>. [Accessed 26 11 2020].
- [6 R. Mitchell, "DRAM, SRAM, FLASH, and a New Form of NVRAM: What's the
7] Difference?," EETech Media, LLC, 16 2 2020. [Online]. Available: <https://www.allaboutcircuits.com/news/dram-sram-flash-and-a-new-form-of-nvram-whats-the-difference/#:~:text=Unlike%20DRAM%2C%20FLASH%20memory%20is,unusable%20as%20a%20CPU%20RAM..> [Accessed 26 11 2020].

- [6 A. Susiripala, "Load Testing an ESP8266," arunoda, 16 11 2017. [Online]. Available:
8] <https://arunoda.me/blog/load-testing-an-esp8266>. [Accessed 26 11 2020].
- [6 R. Mitchell, "ESP32 vs ESP8266 which is better," Electromaker, 5 5 2020. [Online].
9] Available: <https://www.electromaker.io/blog/article/esp32-vs-esp8266-which-is-better>. [Accessed 26 11 2020].
- [7 Adafruit, "Adafruit HUZZAH32 – ESP32 Breakout Board," Adafruit, 2020. [Online].
0] Available: <https://www.adafruit.com/product/4172#description>. [Accessed 25 11 2020].
- [7 Seeed, "RedBearLab CC3200 WiFi board," Seeed Technology Co.,Ltd., 2020.
1] [Online]. Available: <https://www.seeedstudio.com/RedBearLab-CC3200-WiFi-board-p-2313.html>. [Accessed 25 11 2020].
- [7 Eleduino Studio, "Eleduino RedBearLab CC3200 WiFi Board for Arduino," Amazon,
2] 2020. [Online]. Available: https://www.amazon.com/Eleduino-RedBearLab-CC3200-Board-Arduino/dp/B07MJ4JTVZ/ref=sr_1_2?dchild=1&keywords=RedBearLab&qid=1606328104566&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC3200. [Accessed 25 11 2020].
- [7 Texas Instruments, "CC3200 SimpleLink™ Wi-Fi® and Internet-of-Things Solution,
3] a Single-Chip Wireless MCU," 2 2015. [Online]. Available: https://www.ti.com/lit/ds/symlink/cc3200.pdf?ts=1606328104566&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC3200. [Accessed 25 11 2020].
- [7 Raspberry Pi Foundation, "Raspberry Pi 4 Tech Specs," Raspberry Pi, 2020. [Online].
4] Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/?resellerType=home>. [Accessed November 2020].
- [7 Y. Venugopal, "The Advantages and Disadvantages of Surface Mount Technology
5] (SMT)," SIERRA CIRCUITS INC, 21 8 2020. [Online]. Available: <https://www.protoexpress.com/blog/good-not-so-good-sides-surface-mount-technology/>. [Accessed 26 11 2020].

- [7] P. P, "A Beginner's Guide to the ESP8266," 3 8 2017. [Online]. Available:
6] <https://tttapa.github.io/ESP8266/Chap01%20-%20ESP8266.html>. [Accessed 25 11
2020].
- [7] P. Loughhead, "Removing Unused Pads and Adding Teardrops," Altium, 30 May
7] 2017. [Online]. Available: <https://www.altium.com/documentation/altium-designer/removing-unused-pads-and-adding-teardrops-ad?version=17.1>.
- [7] PCB gogo, "PCB Trace - Everything You Need To Know," PCB gogo, 18 July 2019.
8] [Online]. Available:
[https://www.pcbgogo.com/blog/PCB_Trace___Everything_You_Need_To_Know.ht
ml](https://www.pcbgogo.com/blog/PCB_Trace___Everything_You_Need_To_Know.html). [Accessed 9 November 2020].
- [7] Synergise PCB inc, "What is the Composition of a Printed Circuit Board?," Synergise
9] PCB inc, 2020. [Online]. Available: <https://www.altium.com/solution/what-is-a-pcb>.
- [8] D. Marrakchi, "Top 5 PCB Design Guidelines Every PCB Designer Needs to Know,"
0] Altium, 21 February 2017. [Online]. Available: [https://resources.altium.com/p/top-5-
pcb-design-guidelines-every-pcb-designer-needs-know](https://resources.altium.com/p/top-5-pcb-design-guidelines-every-pcb-designer-needs-know).
- [8] S. Ouyang, "How to Design Thermal Reliefs with Minimal Manufacturing Issues,"
1] Tempo Automation, 14 October 2016. [Online]. Available:
<https://medium.com/tempo-automation/thermal-reliefs-1a1576ea0e13>. [Accessed 9
November 2020].
- [8] "Distributed energy resources," Australian Renewable Energy Agency, 28 October
2] 2020. [Online]. Available: [https://arena.gov.au/renewable-energy/distributed-energy-
resources/](https://arena.gov.au/renewable-energy/distributed-energy-resources/). [Accessed November 2020].
- [8] Narang, "Clause-by-Clause Summary of Requirements in IEEE Standard 1547-2018,"
3] National Renewable Energy Laboratory, 2020. [Online]. Available:
<https://www.nrel.gov/grid/ieee-standard-1547/clause.html>.
- [8] I.-2. T. Group, "Generic Standard on Printed Board Design," IPC, Northbrook, IL,
4] 2003.

[8 National Electrical Manufacturers Association, "Enclosures for Electrical Equipment 5] (1000 Volts Maximum)," *NEMA 250-2014*, no. Rosslyn, VA , 2014.

[8 P. Bourque and R. Fairley, Guide to the Software Engineering Body of Knowledge, 6] IEEE Computer Society, 2014.

[8 The Institute of Electrical and Electronics Engineers, Inc, "Supplement to IEEE 7] Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY," Vols. IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012) , p. 3534, 2016.

[8 Wikipedia, "Desert locust," Wikipedia, 1 November 2020. [Online]. Available: 8] https://en.wikipedia.org/wiki/Desert_locust. [Accessed 13 November 2020].

[8 E. Vinje, "Seed Control and GMO Companies," Planet Natural Research Center, 2020. 9] [Online]. Available: <https://www.planetnatural.com/seed-control/>. [Accessed November 2020].

[9 SparkFun Electronics, "Ultrasonic Distance Sensor - HC-SR04," SparkFun 0] Electronics, 2020. [Online]. Available: <https://www.sparkfun.com/products/15569>. [Accessed 22 11 2020].

[9 Elec Freaks, "Ultrasonic Ranging Module HC - SR04," 2020. [Online]. Available: 1] <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>. [Accessed 22 11 2020].

[9 The Pi Hut, "HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi," The Pi Hut, 22 2] 10 2013. [Online]. Available: <https://thepihut.com/blogs/raspberry-pi-tutorials/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi>. [Accessed 28 11 2020].

[9 "Micro Switch or Snap-action Switch," Components 101, 2018. [Online]. Available: 3] <https://components101.com/switches/micro-switch-or-snap-action-switch>.

- [9 [Online]. Available: <https://github.com/Miceuz/i2c-moisture-4> sensor/blob/master/README.md.
- [9 [Online]. Available: <https://www.circuitbasics.com/basics-of-the-i2c-5> communication-protocol/#:~:text=I2C%20is%20a%20serial%20communication,always%20controlled%20by%20the%20master.
- [9 Cisco, "802.11ac: The Fifth Generation of Wi-Fi.," 11 2018. [Online]. Available: <https://www.cisco.com/c/dam/en/us/products/collateral/wireless/aironet-3600-series/white-paper-c11-713103.pdf>. [Accessed November 2020].
- [9 C. David, "SD Card Tutorial for Arduino, ESP8266 and ESP32," diyIoT, 2020. 7] [Online]. Available: <https://diyiot.com/sd-card-arduino-esp8266-esp32/>. [Accessed 28 11 2020].
- [9 M. Gupta, "ML | Types of Learning – Supervised Learning," Geeks for Geeks, 16 May 8] 2020. [Online]. Available: www.geeksforgeeks.org/ml-types-learning-supervised-learning. [Accessed November 2020].
- [9 B. Grossfeld, "Deep Learning vs Machine Learning," Zendesk, 12 October 2020. 9] [Online]. Available: www.zendesk.com/blog/machine-learning-and-deep-learning/. [Accessed 2020].
- [1 J. Iqbal, "A Review of Application of Deep Learning for Weeds and Crops 00 Classification in Agriculture.," in *2019 International Conference on Robotics and Automation in Industry (ICRAI)*, Rawalpindi, Pakistan, 2019.
- [1 X. Y. a. M. Sun, "A Survey on Deep Learning in Crop Planting," in *IOP Conference 01 Series: Materials Science and Engineering*, 2019.]
- [1 A. W. S. N. S. B. S. M. S. Yu J, "Deep learning for image-based weed detection in 02 turfgrass European Journal of agronomy," *European Journal of Agronomy*, no. DOI:] 10.1016/j.eja.2019.01.004, 2019.

[1 O. B. F. B. P. L. R. P. G. D. J. T. D. W.-F. Y. B. James Bergstra, "Theano: A CPU
03 and GPU Math Compiler in Python," *Python in Science Conference*, 2010.
]

[1 "Why TensorFlow," [Online]. Available: <https://www.tensorflow.org/about>.
04 [Accessed 2020].
]

[1 "Keras," [Online]. Available: <https://keras.io/>.
05
]

[1 J. Terra, "Keras vs TensorFlow vs Pytorch: Understanding the Most Popular Deep
06 Learning Frameworks," September 2020. [Online]. Available:
] <https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article>.

[1 Y. Lu and S. Young, "A survey of public datasets for computer vision tasks in
07 precision agriculture," *Computers and Electronics in Agriculture*, vol. 178, 2020.
]

[1 A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, "YOLOv4: Optimal Speed and
08 Accuracy of Object Detection," 23 4 2020. [Online]. Available:
] <https://arxiv.org/abs/2004.10934>. [Accessed 25 4 2021].

[1 L. Weng, "Object Detection Part 4: Fast Detection Models," 27 12 2018. [Online].
09 Available: [https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-](https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html)
] [4.html](https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html). [Accessed 26 4 2021].

[1 T. Shah, "Measuring Object Detection models — mAP — What is Mean Average
10 Precision?," *Towards Data Science*, 2018 15 1. [Online]. Available:
] [https://towardsdatascience.com/what-is-map-understanding-the-statistic-of-choice-](https://towardsdatascience.com/what-is-map-understanding-the-statistic-of-choice-for-comparing-object-detection-models-1ea4f67a9dbd)
[for-comparing-object-detection-models-1ea4f67a9dbd](https://towardsdatascience.com/what-is-map-understanding-the-statistic-of-choice-for-comparing-object-detection-models-1ea4f67a9dbd). [Accessed 15 4 2021].

[1 S. McCann, "Tag: average precision," Wordpress, 1 9 2011. [Online]. Available:
11 <https://sanchom.wordpress.com/tag/average-precision/>. [Accessed 15 4 2020].
]

[1 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD:
12 Single Shot MultiBox Detector," 8 12 2015. [Online]. Available:
] <https://arxiv.org/abs/1512.02325>. [Accessed 26 4 2021].

[1 T.-Y. Lin, P. Goyal, R. Girshick, . K. He and P. Dollár, "Focal Loss for Dense Object
13 Detection," 7 2 2018. [Online]. Available: <https://arxiv.org/abs/1708.02002>.
] [Accessed 27 4 2021].

[1 J. Solawetz, "Breaking Down YOLOv4," roboflow, 4 6 2020. [Online]. Available:
14 <https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/>. [Accessed 27 4 2021].
]

[1 A. Kathuria, "How to Implement a YOLO (v3) Object Detector from Scratch in
15 PyTorch: Part 1," KDnuggets, 5 2018. [Online]. Available:
] [https://www.kdnuggets.com/2018/05/implement-yolo-v3-object-detector-pytorch-
part-
1.html#:~:text=YOLO%20makes%20use%20of%20only,to%20downsample%20the
%20feature%20maps](https://www.kdnuggets.com/2018/05/implement-yolo-v3-object-detector-pytorch-part-1.html#:~:text=YOLO%20makes%20use%20of%20only,to%20downsample%20the%20feature%20maps). [Accessed 27 4 2021].

[1 J. Nelson and J. Solawetz, "YOLOv5 is Here: State-of-the-Art Object Detection at 140
16 FPS," roboflow, 10 6 2020. [Online]. Available: [https://blog.roboflow.com/yolov5-is-
\] here/](https://blog.roboflow.com/yolov5-is-here/). [Accessed 27 4 2021].

[1 C. Supeshala, "YOLO v4 or YOLO v5 or PP-YOLO?," towards data science, 22 8
17 2020. [Online]. Available: [https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-
\] yolo-dad8e40f7109](https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109). [Accessed 27 4 2021].

[1 "Stepper Motor Resistance Test," [Online]. Available: [http://www.melco-
18 service.com/docs/AMAYA_Tech_Manual_Web/Troubleshooting/Testing_Methods/
\] Stepper_Motor_Resistance_Test.htm](http://www.melco-18-service.com/docs/AMAYA_Tech_Manual_Web/Troubleshooting/Testing_Methods/Stepper_Motor_Resistance_Test.htm).

[1 IEEE, "IEEE Standard for Interconnection and Interoperability of Distributed Energy
19 Resources with Associated Electric Power Systems Interfaces," *IEEE Std 1547-2018*
] (*Revision of IEEE Std 1547-2003*), pp. 1-138, 2018.

[1 C. S. Corporation. [Online]. Available:
20 <https://www.cypress.com/file/358916/download>.
]

[1 Cypress Semiconductor Corp, "CYW43455 - WICED IEEE 802.11ac WiFi /
21 Bluetooth 4.1 Connectivity Solution for IoT Applications," vol. 2nd Ed, p. Cypress
] Semiconductor Corp, 2019.

[1 Raspberry Pi Foundation, "GPIO," Raspberry Pi Foundation, 2020. [Online].
22 Available: <https://www.raspberrypi.org/documentation/usage/gpio/>. [Accessed 22 11
] 2020].

[1 Expressif Systems, "ESP8266EX Datasheet," 2020. [Online]. Available:
23 [https://www.espressif.com/sites/default/files/documentation/0a-
\] esp8266ex_datasheet_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). [Accessed 28 11 2020].

[1 P. Forret, "Megapixel Calculator," Toolstud, 2020. [Online]. Available:
24 [https://toolstud.io/photo/megapixel.php?compare=video&calculate=compressed&wi
\] dth=1622&height=1200](https://toolstud.io/photo/megapixel.php?compare=video&calculate=compressed&width=1622&height=1200). [Accessed November 2020].

[1 [Online]. Available: [https://components101.com/wireless/esp8266-pinout-
26 configuration-features-datasheet](https://components101.com/wireless/esp8266-pinout-26-configuration-features-datasheet).
]

[1 C. Electronics, "12c Soil Moisture Sensor," Tindie, [Online]. Available:
27 <https://www.tindie.com/products/miceuz/i2c-soil-moisture-sensor/>.
]

[1 Open Builds, "Open Builds," [Online]. Available:
28 [https://cad.onshape.com/documents/da7bab15f184278cb94d3885/w/7e26a69187cd7
\] fec69587c82/e/5cd8318085c76c9166833301](https://cad.onshape.com/documents/da7bab15f184278cb94d3885/w/7e26a69187cd7fec69587c82/e/5cd8318085c76c9166833301).

[1 A. Bochkovskiy, . C.-Y. Wang and . H.-Y. M. Liao, "YOLOv4: Optimal Speed and
29 Accuracy of Object Detection," 23 4 2020. [Online]. Available:
] <https://arxiv.org/abs/2004.10934>. [Accessed 26 4 2021].