

MACHINE TO AUTOMATE SOIL HUMIDITY VARIATION DURING MULTIPLE, CO-  
OCCURRING, CLIMATIC-VARIABLE, PLANT-GROWTH EXPERIMENTS.

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE  
UNIVERSITY OF HAWAI'I AT MĀNOA IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE  
IN  
MECHANICAL ENGINEERING  
MAY 2016

By

Reika Arata

Thesis Committee:

A Zachary Trimble, Chairperson  
Peter Berkelman  
Scott Miller

Keywords: Phenotyping, Machine designing, Automation, Reliable

## **ACKNOWLEDGEMENTS**

I would like to express my deepest appreciation to my committee chair, Professor A Zachary Trimble for his constant guidance and encouragement during the term of my candidature. Without his valuable assistance, this work would not have been completed.

I am also indebted to the Department of Mechanical Engineering staff, faculty, and students at University of Hawai'i at Mānoa for their support and cooperation.

In addition, a thank you to Mr. Robert A. Rohde and Global Warming Art for permission to include a figure created by Mr. Rohde as part of my thesis/dissertation.

Most importantly, none of this would have been possible without the love and patience of my family in Japan. I would like to express my heart-felt gratitude to my family for their love, encouragement, and financial support throughout my life and my studies in the U.S.A. They made me live the most unique and carefree childhood that has made me who I am now.

Finally, I appreciate many friends who helped me manage through these difficult years. Their support and care helped me stay focused on my graduate study. I greatly value their friendship and I deeply appreciate their belief in me. I would like to thank David Horton, Jeffery Oshiro, and Laura Fitzpatrick for proofreading and supporting me.

Special thanks to all the people who along the way believed in me.

## ABSTRACT

By monitoring plant growth in an automated, controlled environment, the plant's response to changing climate conditions can be revealed to a greater degree of understanding and precision than current knowledge. Observing these responses with respect to changes in temperature, humidity, and amount of carbon dioxide and ozone can help us understand, predict, and mitigate factors related to climate change. The overall goal of this research is to introduce an Intelligent Plant System (IPS) that measures plant growth while automatically controlling the multiple climatic variables mentioned previously with stronger operational reliability, increased capability, reduced cost, reduced complexity, and reduced form factor when compared to similar devices on the market. In the course of this thesis, the weighing and watering system of an automated soil humidity machine is developed. The machine can study 100 plants simultaneously – 10 different humidity treatments with 10 replicates per treatment. Soil humidity is determined by mass measurement. Target weights for a plant-pot combination are created for 10 different weight groups by users. Each pot is maintained at the target mass, and thus target soil humidity, throughout each study by replenishing any lost mass with water. This thesis describes the theoretical design of the machine, in Solidworks, the manufacturing of the machine, the software development for the sensors and motors, and the results of measurement outputs. Ultimately, a machine is realized that automatically maintains a plant-pot's mass within a standard deviation of +/- 1 g.

## TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS.....	i
ABSTRACT.....	ii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
CHAPTER	
1. INTRODUCTION.....	1
1.1. Research Motivation.....	1
1.2. Overall Research Purpose and Vision.....	3
1.3. Scope of Research, Project Overview, and Summary of Contributions.....	6
1.4. Thesis Outline.....	7
2. BACKGROUND AND PRIOR ARTS.....	9
2.1. The Way to Measure Soil Humidity.....	9
2.2. Detailed Existing Machines.....	9
2.3. Detailed Prior Art on Manipulation and Watering.....	11
2.4. Reliability Theory.....	12
3. CONCEPT DESIGN.....	13
3.1. System Level Design Mechanical Design Process.....	13
3.1.1. Design Identification and Requirements.....	13
3.1.2. Design Features and Functionalities.....	15
3.2. Automated Soil Humidity Machine Design Process.....	17
3.2.1. Design Identification and Requirements.....	17
3.2.2. Design Features and Functionalities.....	18
3.2.2. Solid Modeling of the Machine.....	19
4. MECHANICAL SYSTEM.....	22
4.1. Mechanical Design Theory.....	22
4.1.1. Overview of Mechanical System.....	22
4.2. Structure Description.....	23
4.2.1. Hardware Identification and Architecture.....	23
4.2.1.1. Aluminum Angle Extrusion and Plastic Lumber..	24
4.2.1.2. 80/20 Aluminum Extrusion and Linear Bearings....	26
4.2.1.3. Styrofoam.....	27

4.3.	Positioning System (Y&Z positioning).....	28
4.3.1.	Hardware Identification and Architecture.....	29
4.3.1.1.	Stepper Motor.....	30
4.3.1.2.	Stepper Motor Driver.....	32
4.3.1.3.	Lead Screw.....	35
4.3.1.4.	Belt / Pulley.....	36
4.3.1.5.	Limit Switch.....	37
4.3.1.6.	Bearing and Vertical Linear Guide.....	38
4.3.2.	Carrier.....	39
4.4.	Water Delivery System.....	40
4.4.1.	Hardware Identification and Architecture.....	40
4.4.1.1.	Valves (solenoids) and Plumbing.....	41
4.4.1.2.	Relay.....	42
4.4.2.	Water Delivery Method.....	42
4.5.	Power Supply.....	43
5.	ELECTRICAL SYSTEM.....	45
5.1.	Electrical Design Theory.....	45
5.1.1.	Overview of Electrical System.....	45
5.2.	Hardware Identification and Architecture.....	48
5.2.1.	Microcontroller.....	48
5.2.2.	Load Cell.....	49
5.2.3.	ADC.....	50
5.2.4.	MicroSD Shield.....	52
5.3.	Power Supply.....	53
5.4.	Circuit Design.....	54
5.5.	Software Development.....	56
5.5.1.	Flow Chart.....	56
5.5.2.	Microcontroller – Arduino.....	59
5.5.2.1.	Library.....	59
5.5.2.2.	Arduino Mega Digital Pin Map.....	60
5.5.3.	Data Logging.....	60
6.	PROTOTYPE DEVELOPMENT AND CONSTRUCTION.....	62
6.1.	Solid Modeling Assembly.....	62
6.1.1.	Frame Assembly.....	62
6.1.2.	Carrier Assembly.....	63
6.1.3.	Prototype Solid Model .....	65
6.2.	Prototype Construction and Assembly .....	66
6.3.	Bill of Overall Materials.....	69
7.	RELIABILITY TESTING AND ANALYSIS.....	72
7.1.	Seed Selection.....	72

7.2.	Analysis.....	73
7.2.1.	Positioning System.....	73
7.2.2.	Cup Picking Up System.....	76
7.2.3.	Machine Reliability.....	78
7.3.	Repeatability Check Statistical Method.....	79
7.3.1.	Weighing System – Load Cell.....	80
7.3.2.	Watering System – Weight Related.....	87
7.3.3.	Measurements.....	89
7.4.	When Error Occurs.....	91
8.	CONCLUSION AND FUTURE WORK.....	93
8.1.	Conclusions.....	93
8.2.	Prototype Future Improvements.....	93
8.2.1.	PCB Development.....	94
8.2.2.	Mechanical Development.....	94
8.2.3.	Cost Improvement.....	95
8.3.	Plant Growing Pictures.....	96
8.3.1.	Current Situations.....	96
APPENDICES		
A.	Arduino Software.....	98
B.	Data Logging Example.....	106
BIBLIOGRAPHY.....		
		108

## LIST OF TABLES

	Page
Table 3.1. Summary of reasons for dissatisfaction with existing phenotyping machines. ....	14
Table 3.2. List of desired requirements of the prototype phenotyping machine..	15
Table 3.3. Desired features for the ideal prototype phenotyping machine. ....	17
Table 3.4. List of desired requirements of the automated soil humidity machine. ....	18
Table 3.5. Desired features for the automated soil humidity machine. ....	18
Table 4.1. Current and Micro-stepping settings of ST-M5045. ....	34
Table 4.2. Current and Micro-stepping settings of ST6600. ....	35
Table 4.3. Power budget of positioning and watering systems to select a power supply. ....	43
Table 5.1. Arduino digital pin map. ....	47
Table 5.2. Power budget of weighing system and other electronics. ....	53
Table 6.1. Purchased list. ....	70
Table 7.1. Experiment of Y-axis stepper motor steps lost. ....	75
Table 7.2. Mass measurement on 3 different locations on a load cell. For each sensor, 30 measurements are taken. Each averaged measurements are an average of 20 measurements. ....	81
Table 7.3. Explanation of Reading X-axis of Figure 7.8. ....	88

## LIST OF FIGURES

	Page
Figure 1.1. Emissions levels determine temperature rises. ....	3
Figure 1.2. Examples of available automated phenotyping machines. ....	6
Figure 3.1. System level overall block diagram. ....	16
Figure 3.2. Solid model of the original prototype automated soil humidity machine. ....	20
Figure 3.3. Brief automated soil humidity system overall flow chart. ....	21
Figure 4.1. Solid model of structural architecture. ....	24
Figure 4.2. Aluminum angle extrusion and plastic lumber installation. ....	25
Figure 4.3. Pictures of 1010, 1020, and a linear bearing from 80/20 Inc. ....	26
Figure 4.4. Styrofoam with LED lights attached. ....	27
Figure 4.5. Picture of the origin at each axis. ....	28
Figure 4.6. Solid model of positioning system. ....	29
Figure 4.7. Solid model of Z axis positioning system. ....	30
Figure 4.8. Y axis stepper motor from ebay, Z axis stepper motor from Sparkfun, and coupling hubs and a spider from McMaster-Carr. ....	31
Figure 4.9. Stepper motor driver, ST-M5045 for Y-axis stepper motor and ST6600 for Z-axis stepper motors. ....	33
Figure 4.10. Bearing and vertical linear guide installation. ....	39
Figure 4.11. Solid model of water delivery system. ....	41
Figure 4.12. Power connections for each components. ....	44
Figure 5.1. Arduino mega from Sparkfun. ....	49
Figure 5.2. Load cell from ebay. ....	50
Figure 5.3. ADC – Chip HX711. ....	51

Figure 5.4. MicroSD shield from Sparkfun. ....	52
Figure 5.5. Weighing and other power connections. ....	54
Figure 5.6. Schematics of all wire connections in Eagle file. ....	55
Figure 5.7. Detailed software flow chart. ....	58
Figure 6.1. Solid model of the frame – Wooden plate, poles, and the pulley system. ....	63
Figure 6.2. Solid model of the carrier – Square tube, lifting mechanism, cup holders, water tube fixers. ....	65
Figure 6.3. Solid model of prototype soil humidity machine. ....	66
Figure 6.4. Assembled prototype soil humidity machine. ....	68
Figure 6.5. Plants interacting with machine – Plants growing pictures taken at Day10. ....	69
Figure 7.1. Cup tipped over. ....	74
Figure 7.2. Plots of test 1 step readings. ....	76
Figure 7.3. 3D printed cup holder improvement. ....	77
Figure 7.4: Mass Measurement on a 3 different locations on a load cell; outside, midpoint, and inside. ....	82
Figure 7.5. Standard deviation of load sensor measurements when machine is not moving. ....	83
Figure 7.6. Standard deviation of load sensor measurements when machine moves Z axis only. ....	84
Figure 7.7. Standard deviation of load sensor with both Y and Z axis motions without watering system. Both data is taken before software is upgraded and #2 and #6 ADCs are replaced. Top plot shows that there are 2 bad sensors (#2 and #6), and bottom plot is zoomed in version after #2 and #6 data is removed. ....	86
Figure 7.8. After watered measurements and target mass from complete cycle. X axis indicates Station 1 is Sensor 1 through 8, Station 2 is Sensor 9 through 16, and so on. ....	88

Figure 7.9. Standard deviation from complete cycle. All sensors at any stations got standard deviation of less than 1 g. X axis indicates Sensor 1 through 8 are in Station 1, Sensor 9 through 16 are in Station 2 sensor 1 through 8, and so on. .... 89

Figure 7.10. Plots while operating. The cup slowly dries out until the mass falls below the threshold. Then the cup gets water so it increases its mass. This is the data from sensor 1 at station 1. X-axis indicates the number of cycle and Y-axis is the measured mass. .... 91

Figure 8.1. Plants growth – Day 21. .... 97

Figure 8.2. Plants growth – Cup#3.3: Day7 (Left), Day21 (Center), Day38 (Right). .... 97

# CHAPTER 1

## INTRODUCTION

### 1.1. Research Motivation

The Earth's climate and environment are constantly changing. With these changes, there arise many uncertainties in ecosystem behavior and plant growth that must be studied. In the last 150 years, industrial activities have increased atmospheric carbon dioxide levels from 280 parts per million to 401.85 parts per million (ESRL, 2015). There is a 90 percent probability that human-produced greenhouse gases such as carbon dioxide, methane, and nitrous oxide have caused much of the observed increase in Earth's temperature over the past 50 years (NASA, 2016). In fact, it is well known that Arctic ice is getting thinner and melting, and some islands and lands close to the sea level are sinking (ipcc, 2007).

Besides the fact that climate change brings human damages, it also affects plants and crops. As mentioned, continued greenhouse gas generated at or above current rates would cause further changes to other climate conditions, including increased temperature, rainfall, and reduction in tropospheric ozone, resulting in greater influx of solar radiation. Surely there is a possibility that shifting climate patterns may change the habitats of crops and other plants. Thus, understanding the behavior of plants in adaptation to particular environmental changes is critical and provides us with insight into the evolution and ecological needs of plant life (Alonso-Blanco, et al., 2009).

According to Morison and Lawlor (Morison & Lawlor, 1999), our knowledge of plant responses to interacting climate variables is limited because of technological issues. In fact, multi-variable studies are factorial in nature and the number of plants needed for each study increase exponentially as the number of variables increases linearly. Each variable, such as temperature, humidity, and gases (CO<sub>2</sub> and O<sub>3</sub>) contains 10 different treatments, which consists of 10 samples. For example, in a case of humidity, say 10 treatments are 10 g, 20 g, 30 g, and up to 100 g of water needs to be added to the original weight of the cup. One treatment family (i.e. 10 g) needs 10 plant samples, thus 100 plants are required in total. This is just one variable. A two variable interaction means that

the 100 plants from one variable are in one treatment of the other variable, and for 100 plants in 10 treatments shows 1,000 plants are required. Similarly, a three variable interaction requires an experiment with 10,000 plants, and a four-variable interaction experiment requires 100,000 plants. Controlling each of the variable treatments accurately and manipulating that many plants is humanly impossible. Thus utilizing automated machines to collect data and control variables might be the most effective solution for these experiments.

By contributing to climate change science with an automated machine, it increases the possibility to improve climate prediction models and resolve the plant's responses to the environment that is notably a major source of uncertainty in future climate and ecosystem projections. (Curtis and Wang 1998, Morison and Lawlor 1999, Cramer et al. 2001, Rustad et al. 2001, Tubiello et al. 2002, Coelho and Manfrino 2007, Oliver et al. 2009, Lobell and Gourdji 2012, Anav et al. 2013, Hou et al. 2013, Huntingford et al. 2013, Jones et al. 2013, Lu et al. 2013, Rosenzweig et al. 2013)

The Figure 1.1 shows an example of estimation of global average temperature rise under the Special Report on Emissions Scenarios emissions scenario relative to global average temperatures in 2000 (Rohde, 2009). Each color indicates different models taken from the Intergovernmental Panel on Climate Change – Data Distribution Centre (IPCC-DDC). We are uncertain how much global average temperature will actually rise, and this unknown temperature raise will be affecting on and damaging our environment if not preparing for the worst scenario (data taken from Center for Climate System Research & National Institute for Environmental Studies (CCSR / NIES)).

# Global Warming Projections

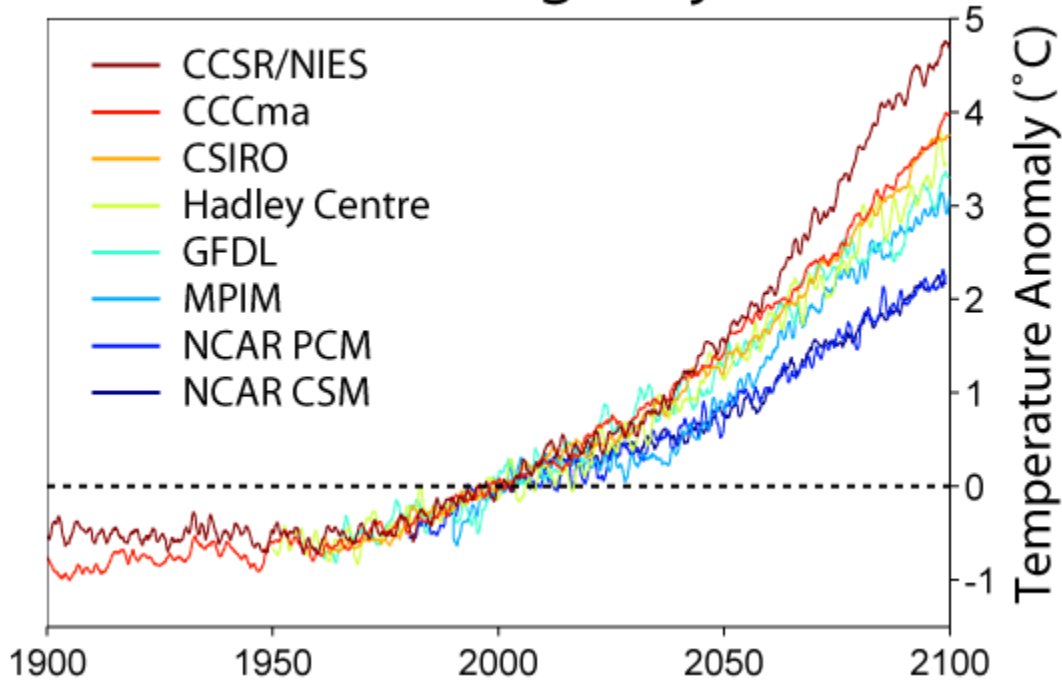


Figure 1.1. Emissions Levels Determine Temperature Rises

\*Image created by Robert A. Rohde / Global Warming Art.

[http://www.globalwarmingart.com/wiki/File:Global\\_Warming\\_Predictions\\_png](http://www.globalwarmingart.com/wiki/File:Global_Warming_Predictions_png)

## 1.2. Overall Research Purpose and Vision

In order to provide us useful ecological and evolutionary data to evaluate plants' growth various environments, all the requirements of the machine need to be considered. First, all the parameters that are causing climate change such as CO<sub>2</sub>, O<sub>3</sub>, and temperature need to be included. Also, the delivery of nutrients and water are controlled in a closed environment. Presumably increasing the number of phenotyping machines as well as its accuracy and throughput could provide human beings with some ideas of the future plants habitats and evolutions within a short time. Phenotype is how genetic and environmental influences come together to create an organism's physical appearance and behavior.

Indeed, there are existing phenotyping machines in the current market, shown in Figure 1-2. Phenoscope, WIWAM, and Phenopsis are examples of machines that are able to monitor large scale of growth responses to the environment for the purpose of plant research or breeding. The methodology of these machines are similar to what is described in this paper; manipulating a pot, measuring its weight, and watering correspondingly.

Despite the number of existing machines, research about habitats of plants have not progressed much. For example, Institut National de la Recherche Agronomique (INRA) Science and Impact owns four Phenoscopes to keep track of vegetable growths under various conditions. The Phenoscope consists of 3 m x 1 m aluminum table on a steel structure, which is capable of up to 735 plants, and which can perform various actions, such as weighing, watering, imaging, and moving. They also have two different stations for weighing/watering and imaging at the right and left corners of the machine, and each pots are taken to the station to be treated by the machine arm (Tisne, et al., 2013). In spite of their ability of tracking plant growths utilizing photographs and data analysis, Phenoscope and similarly Phenopsis are powered by an overly complex, and thus too expensive compressed air system. In addition, WIWAM and also Phenopsis utilize expensive closed-loop servo control, which provides sub-micron precision that is not necessary for manipulating plants. Their costs are extremely expensive. WIWAM, Phenoscope, and Phenopsis cost US\$109,000 (S.Dhondt, n.d.), US\$204,000 (Loudet, n.d.), and US\$109,000-134,000 (Desigaux, n.d.) respectively. Another system not shown in Figure 1.1 is PlantScreen which costs US\$195,000 (Hunt, n.d.). In all cases, the number of hundred plants ranges from 300 to 1,000 and only water content can be controlled independently. Temperature, CO<sub>2</sub>, and any other variables are controlled globally and thus the same for the entire room. Thus these machines are not qualified to evaluate the plant responses toward other conditions. Automatically monitoring plants with high precision is a big challenge, and surely these parameters greatly increase the cost of these systems.

These facts bring us a couple of problems. As mentioned, the experiment must be replicated 100 times more of existing machines for a three variable interaction, assuming these machines control 1000 plants at once, for example; water, temperature and CO<sub>2</sub> or

O<sub>3</sub> that each consist of 10 treatments. Considering this fact, one of the following options must be taken. The first choice is to use a single robot for multiple experiments. However, it leads to the time problem. If the duration of each environmental treatment experiment averages three months, the final results that one single machine will collect all three variable interaction data is after 300 months (3 months x 100). The second option is utilizing multiple machines at an increased cost. It is possible to have complete results in three months by investing money in 100 machines in 100 separate rooms. A single machine costs USD 100,000, and thus a total of 10 million dollars are required. Additionally, a space is needed that is capable of fitting 100 machines. For a Phenoscope, a space of at least 300 m x 100 m is required. Neither of these are easy choices, and it highlights the complications in conducting factorial climate change experiments.

This system provides a solution for these challenges. The overall system requirements of this machine are to develop an automated climate control system that provides the capability to create a completely stable environment with respect to CO<sub>2</sub>, O<sub>3</sub>, temperature, and humidity for a 100 plant sample size. This involves 10 replicates and 10 treatments, growing for a period of three months. In addition, the machine will actively gather plant growth data, and will be intended for a middle-school level end-user. The main objective is to develop a device with lower complexity, stronger operational reliability, increased capability, and relatively smaller form-factor than similar devices on the market while at a significantly lower cost. Operational reliability is defined as the probability that a given product will successfully perform a required function without failure, under specified environmental conditions, for a specified period of time (Wunnenberg & Frank, 1990). By considering utilizing a simple control system, such as open-loop stepper motors instead of a closed-loop servo control system with common components, those machines become inexpensive and the reliability increases (Instruments, 2008).



Figure 1.2. Examples of available automated phenotyping machines. Their nicknames are WIWAM (A), Phenoscope (B) and Phenopsis (C).

### 1.3. Scope of Research, Project Overview, and Summary of Contributions

The contribution of this study is the development, prototype construction, and evaluation of an automated climate machine without the gas and temperature systems, which control only the watering system. This prototype is called the automated soil humidity machine.

The requirements of the watering system are designed based on system level requirements and modified. The concept of the requirements are introduced in the prior art. The detailed functional requirements of the watering system are discussed in section 3.2.

Prior to mechanical design, an extensive review of background and prior work is performed. Following literature review, design constraints and design requirements are established and utilized to create a solid model of the automated soil humidity phenotyping machine using SolidWorks. With some assumptions, calculation and analysis of selective components such as motors, sensors and mechanical parts were performed before ordering parts. For example, the maximum mass of each cups are assumed about 500g to calculate maximum and center beam deflection. Final decision is made by comparing its dimensions, materials, and costs. The load cell is also examined and analyzed to determine the significance of the load position on the sensor. Another

example is stepper motor selection. The motor sizes are specified based on the moment of inertia, torque and speed, using the estimated system mass to find the most suitable motors.

Design drawings are drafted using Solidworks 3D modeling software. These designs are then prototyped with 3D printer systems or with the assistance of the University of Hawaii at Manoa machine shop. Picking up method is experimented with a simplified vertical motion technique and data was taken 100 times to test the design of a 3D printed cup holder. The base assembly is constructed of 80/20 aluminum extrusions attached on the wooden base. Atop the 80/20 extrusions is a carrier consisting of all the weighing and watering components. This carrier traverses the length of the extrusions on linear bearings and is driven by a stepper motor coupled to a pulley mechanism. Once the base is assembled and the motor is installed, the software is developed for testing. When the prototype is assembled and all electronics are placed and integrated, the software intended to drive all electronics and motors as well as calibrate the load cells is implemented. Several experiments are conducted to test and analyze the machine reliability and repeatability of the sensors.

#### 1.4. Thesis Outline

The overview of this thesis is organized as follows. Chapter 2 reviews background and prior arts related to existing machines and reliability and repeatability theory of sensors and stepper motors. Chapter 3 is a discussion of system level concept design and the automated soil humidity machine concept design. In this chapter, all design identification and requirements, and design features and functionalities are stated as well as a solid model of ideal machine is shown. In chapter 4 and chapter 5, mechanical system and electrical system are discussed respectively. Mechanical system consists of actions such as positioning system and watering system, and also frame design. All hardware is identified, and justifications of selection is mentioned. The weighing system is included in the electrical system with data logging system. Those hardware

identification and architecture, power budget, circuit design, and software development are also discussed in this section. After all components are stated, a solid model of prototype of automated soil humidity machine is introduced and also the construction and assembly process is discussed in chapter 6. The overall cost list is also covered in this chapter. The reliability testing and analysis of each subsystem such as positioning, weighing, and watering systems are performed in chapter 7. Lastly, chapter 8 closes this paper with conclusion and some future work.

## CHAPTER 2

### BACKGROUND AND PRIOR ARTS

This chapter provides background and prior arts information that are relevant to this research: methodology of soil humidity measurement, water manipulation of existing machines, humidity control in a greenhouse, and reliability theory.

#### 2.1. Methodology to Measure Soil Humidity

There are multiple ways to consider how to discover the soil humidity. The first consideration is to use a humidity sensor by simply inserting the sensor into the soil and measure the dryness. However, this method is not selected because of its disadvantages to this project. Due to the importance of accuracy, those inexpensive sensors don't qualify for this project. In addition, it is hard to measure humidity where seeds are planted. In other words, the sensor location needs to be directly underneath a watering system and plants. These inaccurate and random measurements cause a watering system not operate as it's requested.

Another method discovered is a weighing sensor. According to a book named "plant and soil", the water content of plants is easily determined by weighing a material (Turner, 1981). As an example, existing machines also utilize weighing method. Their technologies are discussed in the next section.

#### 2.2. Detailed Existing Machines

As mentioned in the previous chapter, there exist automated plant watering machines on the market today such as WIWAM, Phenoscope and Phenopsis (Figure 1.2).

The common mechanism of these machine is utilizing a weighing system to measure plants humidity.

One of the machine produced by WIWAM is the WIWAM xy (WIWAM, 2013). This machine treats pots one at a time by picking up a cup and locating it in a watering station. At the watering station, a cup rotates while watering for an optimal water distribution and water input is accurate to 0.1 mL. Another product, the WIWAM Conveyor, is a larger scale of phenotyping machine that also utilizes a rotating watering method (WIWAM, 2013). This machine treats multiple pots at once. The first row moves on the forwarding conveyor then each pots goes in a separated room to record its data. A watering system is set along the conveyor belt on the way to back to the pots original position. There are at least two watering stations in this machine, and their precision is up to 1 mL.

Another example is the Phenopsis. The major difference is that their cups are steady. The machine pushes up a cup to measure its weight and from the top, one single peristaltic pump pours the required amount of water into the cup. The water is a mixture of nutrients and water. Their method of adjusting a pot weight is to weight the cup before watering. That weight is compared to a present target mass and the weight of the cup is adjusted to match the preset value. Finally it is compared to the preset target mass and adjust a cup weight by adding water. Weight after the water is added is also recorded (Iepse, 2016).

The last existing machine is a Phenoscope. The Phenoscope rotates pots continuously, a typical cycle of 4 hours, on a closed-circuit track (Tisne, et al., 2013). Their weighing and watering station treats a cup with nutrient and water to adjust its weight until it reaches the target mass. A peristaltic pump is utilized in this machine. There are up to 735 plants can be treated with one machine and up to six cycles every 24 hours the robot moves each pot sequentially to all possible positions.

### 2.3. Detailed Prior Art on Manipulation and Watering

Another similar type of controlling environments is a green house. There are some methods utilized to control humidity in this case, such as pressure mist system, high pressure fogging technique, and a flood and bench irrigation system.

The mist systems are adding and cooling humidity by controlling a pressure. It forces water through nozzles that allows fine droplets to evaporate fairly quickly into the air (Dalpe, 2003). This system is dependent on the temperature system in the green house as well. The environment temperature needs to be high enough to evaporate the water, and to maintain its room temperature, the air needs to be cooled. Thus, this system is not applicable to introduce into the machine design as the temperature is also one of the factor to be controlled. Moreover, this mist system is suitable for controlling large amount of samples at once, but then to research a minimum number of plants for many different environments, it is not an appropriate idea. The fogging technique (Zhang & Shipp, 2002) also utilizes a similar method and thus not suitable idea.

The fogging system controls the greenhouse humidity and is capable of avoiding excessively low humidity conditions on sunny days. The fogging mechanism hangs about 3.5 m above plants, and also they utilize temperature sensors, relative humidity sensors, leaf wetness sensors, and infrared sensors to monitor the condition and control humidity.

The last method in the market is a flood and bench irrigation system (Solutions, 2009). The mechanism of this technique fills 1 to 2 inches of water in the area where plants are located. After 6 to 10 minutes, the filled sections are completely drained of water, and the remaining water returns to a reservoir for recirculating. This system could be suitable for the soil humidity machine since, in most plants, the roots are responsible for taking in water. However, controlling 10 different amount of water in one chamber is complex in many different ways such as dimension, cup size, complexity, and cost.

#### 2.4. Reliability Theory with Prior Art

As discussed earlier, the reliability of the system is significant for this project. There are two aspects of the system that need to be considered in terms of reliability: hardware and software. There are also human related factors that affect reliability (Rausand & Hayland, 2004). Depending on the customer and their educational level, which will be discussed in the next chapter, human reliability cannot be ignored. Thus, understanding the interactions between the hardware, software, and humans is essential to keeping the machine reliable. Reliability can be characterized by quality, availability, safety, security, and dependability (Rausand & Hayland, 2004). Because human are involved, it not only focuses on the performance of the machine, but also injury, damage, or loss of equipment that must be avoided. Altogether a machine is considered as a reliable machine.

## **CHAPTER 3**

### **CONCEPT DESIGN**

Before drafting concept models for a new phenotyping machine, it is important to define the functionalities of the machine and the features. To identify these characteristics, it is necessary to understand the needs of the experiment, and conduct background research on existing machines on the market. Also, it is necessary to evaluate how well these machines were meeting the needs of the customer themselves. Once enough information about the existing phenotyping machines and the users is gathered, a list of design features covering the requirements is described.

#### **3.1. System Level Design Mechanical Design Process**

First, the system level design requirements, which were established in a previous research, are discussed. Then based on the system level requirements, the automated soil humidity machine requirements will be considered in the next section 3.2.

##### **3.1.1. Design Identification and Requirements**

During the previous research, the target customer was determined to be high school and middle school students. To obtain complete data within a decent time period, (which requires 100,000 plants according to the factorial study with a four variable interaction experiment), schools such as high schools and middle schools were selected. The primary benefit is students will increase their knowledge of science, including engineering, botany, and biology, and their interest towards STEM fields might be expanded when they are young. The secondary benefit is that the efficiency of data

collection is enhanced for all scientists. At this time, all of us need to work together to gather data, and analyze what is going to happen in the future.

The next step is to understand the structural needs of the machine. Considering the existing machines, to manipulate 100,000 plants with a machine designed for 1,000 plants is unacceptable due to the large size requirements. As mentioned the existing machines require about 300 m x 100 m space for 100 chambers to satisfy the factorial study. If 1,000 chambers each containing 100 plants are designed, the required space is smaller. In addition, it is possible to use the 3<sup>rd</sup> dimension by stuck up a machine. The operational parameters that must be determined would be the user's level of education and training, as well as cost efficiency with regards to constructing each system. There would be 1,000 chambers in various locations that need to be controlled to collect data and be fixed in the event of a malfunction. This leads to the importance of reliability and simplicity. As described in the previous chapter, existing machines costs over US\$100,000 each, and that is hardly possible to purchase 1,000 of them. Thus in order to achieve a long term capacity goal, a machine must be designed such that multiple chambers can be manufactured at reasonable cost. Moreover, with the intended operator education level being high school or middle school, the machine cannot be complex or require high technical proficiency. Some of the reasons for dissatisfaction with existing phenotyping machines are summarized in Table 3.1.

Table 3.1. Summary of reasons for dissatisfaction with existing phenotyping machines.

Expensive
Lack of space
Lack of environmental conditions
Lack of factorial studies to take statistical data
Too complex
Not Portable

Using the summary of problems of existing machines listed in Table 3.1, several design requirements for the new automated climate control phenotyping machine can be identified in Table 3.2.

Table 3.2. List of desired requirements of the prototype phenotyping machine.

Cost efficient
Smaller form-factor
Easy to use
Control CO <sub>2</sub> , O <sub>3</sub> , temperature, humidity, luminosity
Reliable
Ease of manufacture
Ease of repair and maintenance

### 3.1.2. Design Features and Functionalities

First of all, based on the literature reviews, in order to understand plant adaptation, it is important to control factors other than humidity, and to evaluate growth data taken by imaging and weighing systems. Figure 3.1 shows what the ideal phenotyping machine should be and also indicates the purpose of this paper. Again, as seen in Figure 3.1, this paper only discusses humidity systems that is enclosed by the dotted line, but a full scope of this research behind it is to control all other conditions such as temperature and gas as well.

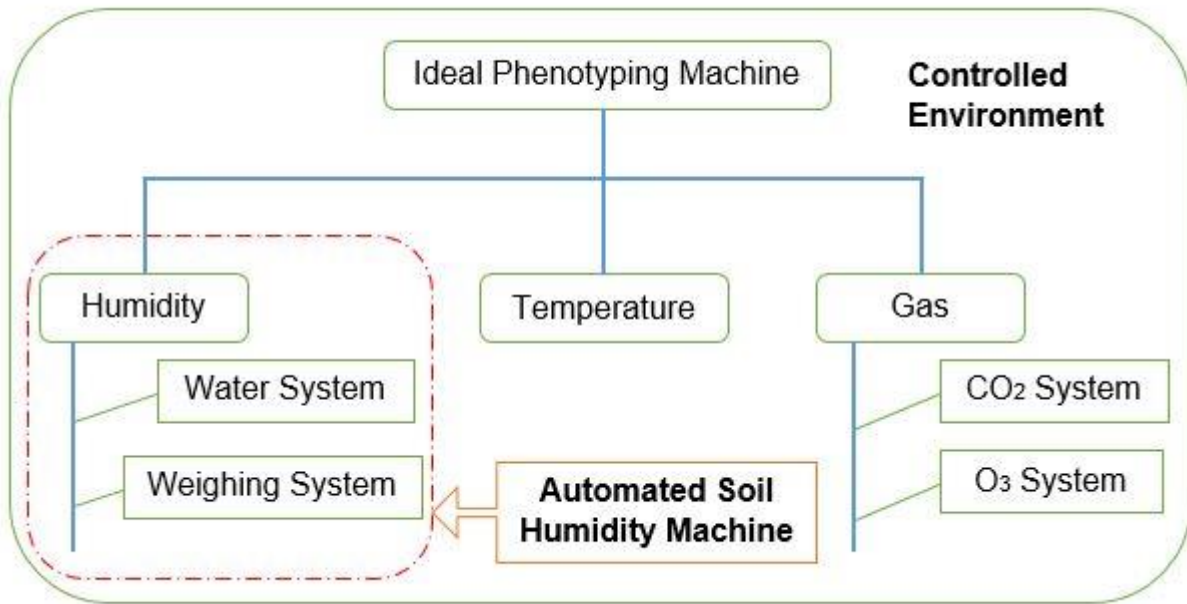


Figure 3.1. System Level Overall Block Diagram

As a part of identifying machine requirements, one of the important machine performance parameters is reliability. In comparison to existing phenotyping machines, the measurement precision of the designed machine shall be considered comparable if not superior and it should be repeatable. The level of precision exhibited by existing machines when measuring CO<sub>2</sub>, O<sub>3</sub>, temperature, and soil moisture are established as 1%, 1%, 0.1°C, and 1% respectively. A manipulation system is also designed with the desire of performing the physical weighing and watering operations, upon 100 plants once per hour. Accounting for the limited time of designing, constructing and analyzing, cost and performance are the greatest tradeoffs. The list of desired features that should be included in the design of an automated climate control phenotyping machine is summarized in Table 3.3.

Table 3.3. Desired features for the ideal prototype phenotyping machine.

<b>Desired Features</b>	<b>Description</b>
Humidity/ Moisture of cups	Precise as 0.1 g of weighing mass of cup
Chamber Temperature	Controlled and maintain the inside temperature within 0.1°C of error
CO <sub>2</sub> amount	Controlled and maintain the CO <sub>2</sub> amount inside the chamber as precise as 1%.
O <sub>3</sub> amount	Controlled and maintain the O <sub>3</sub> amount inside the chamber as precise as 1%.
Operation Hour	Once an hour
Number of plant	Growing 100 plants in one chamber (10 replicates & 10 treatments)

### 3.2. Automated Soil Humidity Machine Design Process

Once system level climate control machine design requirements are established, the automated soil humidity machine is ready to be considered.

#### 3.2.1. Design Identification and Requirements

Based on the given system level requirements shown above, since the machine is modeled to fit all the other systems in addition to the humidity system, ease of construction, consideration of space for the other systems, and portability with those systems in place must all be considered in order to design the machine itself. The control of the humidity and luminosity as well as measurement of plant weight must all be designed with reliability and repeatability as a priority. Moreover, certainly the cost efficiency is considered as well as the ease of maintenance. Controlling CO<sub>2</sub>, O<sub>3</sub>, and temperature is removed from the list. The summary of the automated soil humidity machine requirements is shown in Table 3.4.

Table 3.4. List of desired requirements of the automated soil humidity machine.

Cost efficient
Portable but at least 100 plants are required
Easy to use
Control humidity, luminosity
Reliable / Repeatable
Ease of manufacture
Ease of repair and maintenance
Measure weight for nutrients
Enough space left for other system (temperature & gasses)

### 3.2.2. Design Features and Functionalities

The desired features of the automated soil humidity machine is listed in the Table 3.5. One feature of cup picking up method is added and considerations of CO<sub>2</sub> and O<sub>3</sub> are subtracted from the previous table.

Table 3.5. Desired features for the automated soil humidity machine.

<b>Desired Features</b>	<b>Description</b>
Humidity/ Moisture of cups	Precise as 0.1 g
Weighing cups	Precise as 0.1 g of weighing cup
Cup picked up method	Cups are picked up by the machine from the bottom
Operation Hour	Once an hour
Number of plant	Growing 100 plants in one chamber

### 3.2.3. Solid Modeling of the Machine

Once the customers' requirements and features are determined, the solid model is created shown in Figure 3.2. In the requirements, the machine must be simple and cheap. With the literature review of utilizing a weighing sensor instead of a humidity sensor, the cup locations (either the machine picks up from the bottom or top) and the methodology of measuring masses (either 100 cups are at rest on each load cells or less number of sensors with cup movements) are focused. Moreover, a discussion of watering method is significant reflecting these ideas. Watering system should be either stationary or moveable, methodology of watering, or location etc. are also considered. By following the machine design objectives of simplicity and cost efficiency, one of the prototypes of the automated soil humidity machine came out as shown in Figure 3.2.

The chamber itself is 4 ft. by 8 ft. and can provide enough space for 102 cups as well as the control systems. The machine picks twelve cups at during operation unlike existing machines. The time period over which these actions take place is one cycle completion time. Utilizing twelve sensors at once increases the number of measurements per cycle completion time; therefore, reducing the amount of time that the machine needs to process all actions for all cups. The simplest and inexpensive method of measuring mass of cup is determined to be picked up from the bottom. In addition, the watering system, consisting of 12 solenoids, 12 relays, and 12 water tubes, is attached to a moving component, a carrier. Thus it is a quick operation to measure and water the cups, and also it is easy to construct.

To avoid water spillage on the electronics, all components such as z axis motor, motor drivers, sensors, and other electronics are located either in the square tube, under the square tube, or on the above the watering system. The water reservoir is placed at an elevated location to utilize hydrostatic pressure to induce water flow.

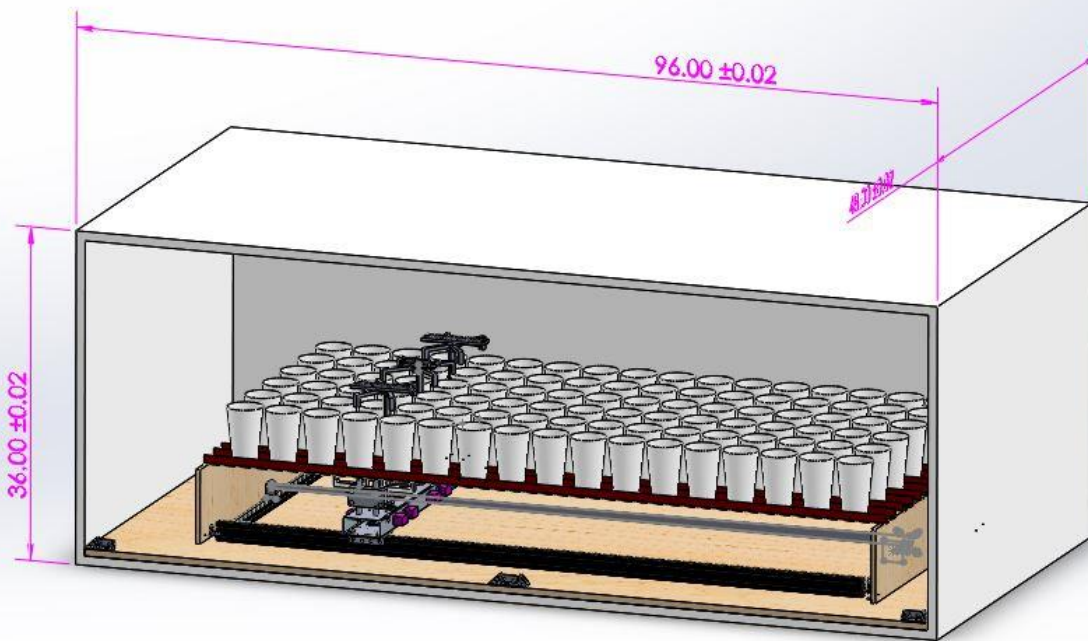


Figure 3.2. Solid Model of the Original Prototype Automated Soil Humidity Machine

Figure 3.3 briefly outlines the system flow chart for the automated soil humidity machine. When the machine is powered “on”, it will automatically go back to the home cycle at home position where the origins of each axis are located. The locations of the origins will be discussed in the following chapter, shown in Figure 4.5. In this positioning system, Y axis is considered as the direction of moving along the longitudinal axis. Z axis is moving up and down. One station is defined as a group of 12 cups that the machine can manipulate at once. The station number increases as the machine moves further from the origin. Figure 4.1 is a solid model of the machine explaining station numbers. Operation starts with the weighing system tare-ing all 12 weighing sensors, then moves to the positioning system to pick up cups. The machine examines the weight of each cup to determine if the cup requires water. If the cup is lighter than the initially defined target mass, the system goes to the watering system; otherwise, it will skip to the positioning system to move the machine downward. Once the machine goes into the weighing and watering system loop, the two systems alternate measuring and watering until the weight

reaches the target mass. Finally, after weighing and watering systems are completed, the machine moves to the next station. This sequence continues to the last station, and when the final process is done, the machine comes back to the home cycle of Y axis.

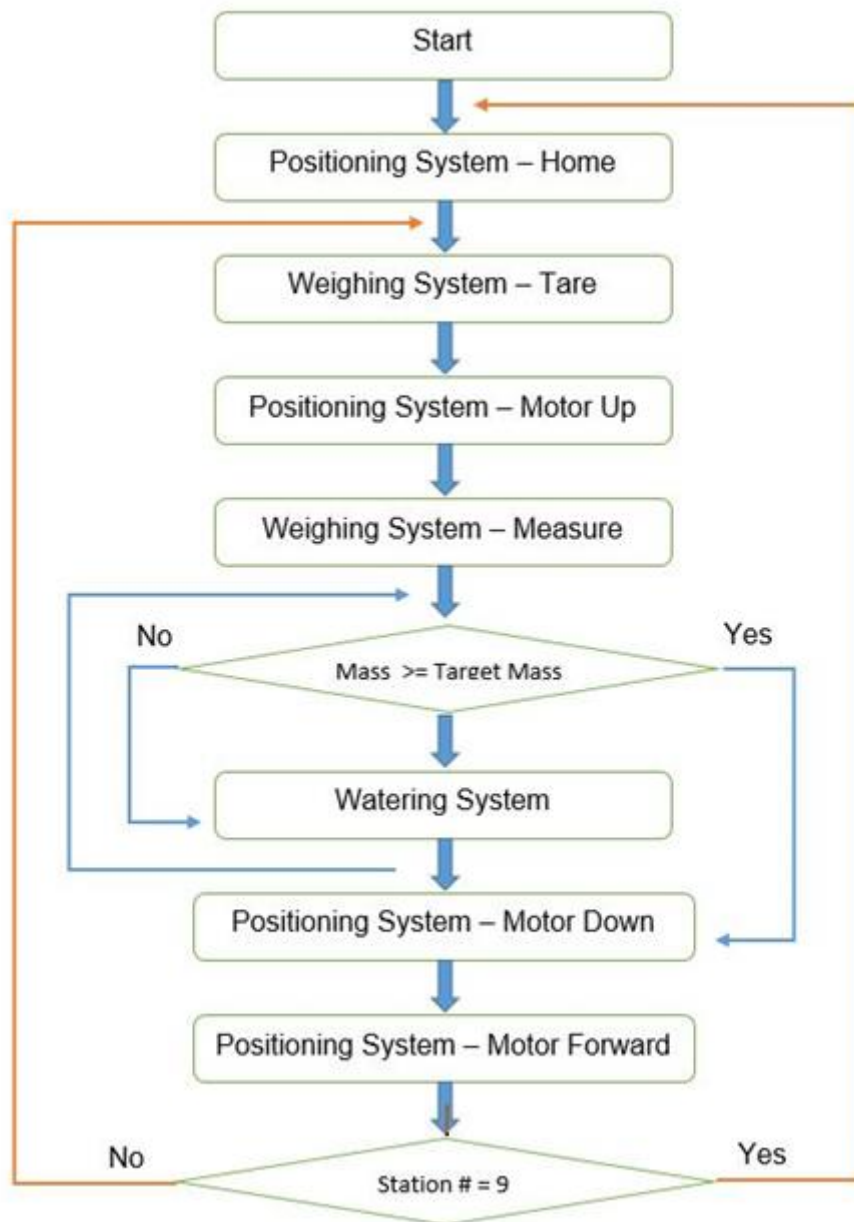


Figure 3.3. Automated Soil Humidity System Overall Flow Chart

## CHAPTER 4

### MECHANICAL SYSTEM

#### 4.1. Mechanical Design Theory

This chapter starts with discussions of a description of mechanical design theory that consists of an overview, structure description, positioning system, water delivery system, and power supply. At each section, all hardware used for this project will be identified followed by a diagram illustrating hardware architecture.

##### 4.1.1. Overview of Mechanical System

This automated soil humidity machine has been designed as a high performance, low-cost system that is capable of handling up to 102 individual cups, each containing some seeds. The overall automated soil humidity machine is composed of three primary components: a frame, a water reservoir and a carriage that conveys electrical components such as sensors and microcontrollers as well as mechanical components such as z-axis motors and water delivery system. Each station contains 12 cups and thus, a total of 9 stations are presented. The 9th station only consists of 6 cups instead of 12, see Figure 4.1. At each station, the load cells with attached cup holders measure the weight of cups and provides water delivery when needed. All cups sit on rails of aluminum angle extrusions elevated above the carrier which traverses the area beneath the cups, and lifts them from underneath Figure 4.2. The machine operation cycle is once an hour to collect data from all 102 cups.

The overall mechanical system consists of two functions, a positioning system and a water delivery system. Starting off at the station #1, (see Figure 4.1), the machine moves up to pick up 12 cups, then measures weights for watering, sequentially from sensor 1 through 12. The sensors numbered 1 through 6 are the side where z axis limit switches are installed. Similarly, the sensor numbered 7 and 12 are the other side, shown

in Figure 4.5. When all the cups are treated, the machine comes down to the original place, then moves to the next station. The machine utilizes two axis: moving forward/backward is on the Y axis and up/down is on the Z axis. The origin is set at pulley side for Y axis and bottom on Z axis (see Figure 4.5). After all stations are monitored, the machine moves all the way back to the origin and waits for another experiment.

## 4.2. Structure Description

The dimensions of the chamber are 4 ft. x 8 ft. x 2 ft. (1.22 m x 2.44 m x 0.61 m). To maintain the temperature inside the chamber and because of the lightweight design requirement, styrofoam walls are chosen. Discussing from the bottom of the design, 2.5 ft. x 6 ft. (0.762 m x 1.8288 m) 80/20 aluminum extrusions are utilized to construct the frame of the machine. The applicable dimension is selected based on the target number of cups and cup size. A major concern is portability, and to increase the portability factor, all components are built on a wooden board so that customers can easily remove the machine from the chamber and transport it without having to realign the frame. The longer sides of the base frame are 1020 extrusions and the shorter sides are 1010 extrusions, which are 1"x2" and 1"x1" respectively. A cup size is 16 fl.oz and it is a typical round shape plastic cup has a bottom diameter of 2.25 inches and a top diameter of 3.50 inches. All cups are located above the carrier, on the angle extrusion. Detailed descriptions of each components are discussed below.

### 4.2.1. Hardware Identification and Architecture

In this section, all hardware used for the structure will be identified. The overall solid model of structural hardware diagram summarizing hardware architecture is presented in Figure 4.1. As briefly discussed earlier, all components are constructed on

the wooden plate. On the top of it, 80/20 extrusions are installed as well as plastic panels at the both end of 80/20 extrusions. Angle extrusions for the cup are fixed atop the plastic panels.

Station numbers are designated starting with 1 on the pulley side of the machine and ordered sequentially to station 9 on the y axis stepper motor side. Each station consists of 12 cups. Now indicating the sensor numbers, by following Figure 4.1 and looking at station 1, the first set of 6 cups are sensor #7 through 12 from left to right when you look in the top view. The sensor #1 through 6 are in the second row from left to right (see Figure 4.5).

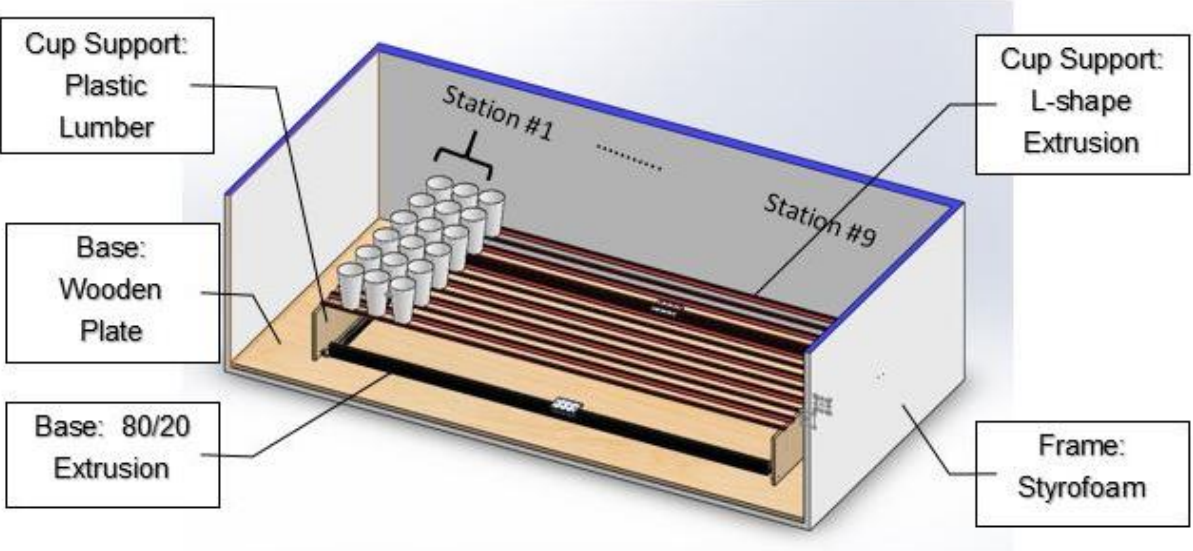


Figure 4.1. Solid Model of Structural Architecture

4.2.1.1. Aluminum Angle Extrusion and Plastic Lumber

Considering the cup locations, the discussion is whether cups should move around inside the chamber like what existing machines do or at rest at the certain place while the

machine moves. To have many cups treated at once and to save cycle time, the latter design is selected. The next question is where the cups are sitting. The cups can either be picked up from the top or below. Since the weight measurement is required and decision of the cup movement, the clever and simple solution is for the cups to be picked up from the bottom. To secure the cups from falling, angle extrusion is selected. Assuming each maximum weight of the cup is 500g. The elastic modulus of aluminum is used for the calculation and with the moment of inertia of angle extrusion, the maximum deflection is computed as 5.47 mm or 0.21 inch, which is acceptable amount for the design. All the aluminum angle extrusion is attached to plastic lumbars and secured at the both ends as shown in Figure 4.2. With deflection calculation and cost comparison of material, aluminum is chosen for its material and its length is 1" by 1".

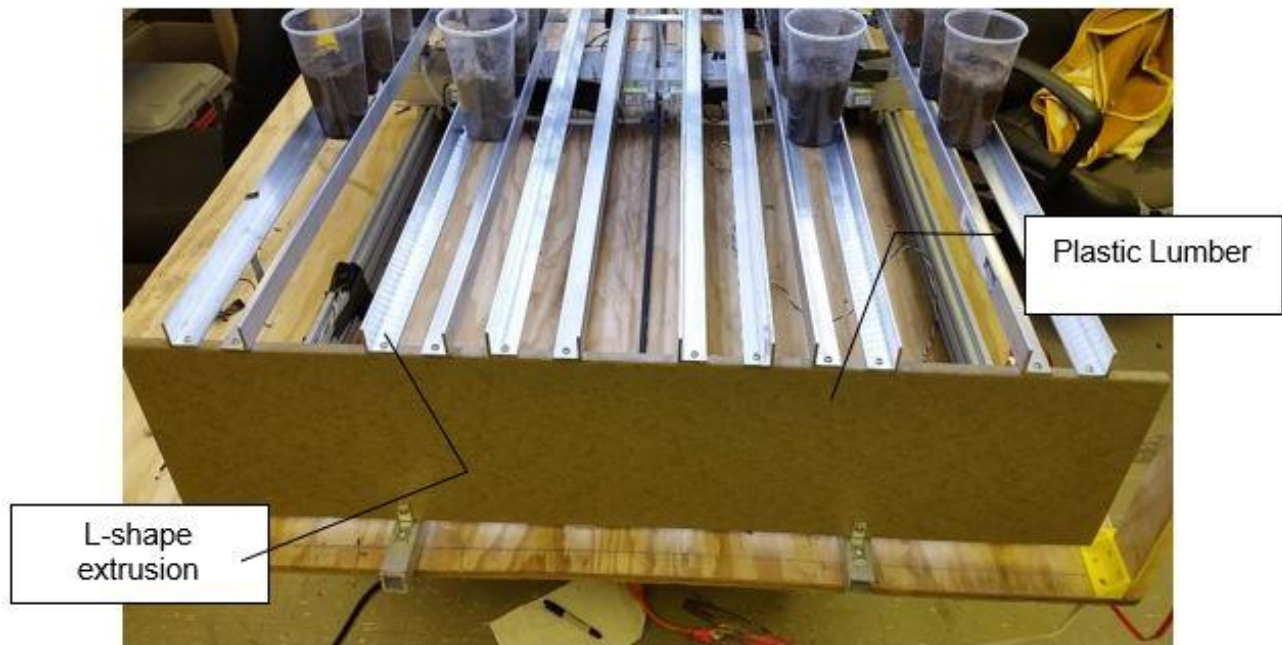


Figure 4.2. Aluminum Angle Extrusions and Plastic Lumbars Installation

#### 4.2.1.2. 80/20 Aluminum Extrusion and Linear Bearings

During the search for qualified linear rails that support smooth linear motions, the major concern was the cost and simplicity. In order to satisfy the requirements that the machine runs under cups, the minimum length of the longer side of base rails need to be 6 ft., which would be expensive depending on the parts in the market. To minimize cost and complexity of structure, an 80/20 aluminum extrusion was selected. In spite of the length, since an entire surface of the 80/20 can be touched to the base wooden plate, deflection could be negligible. To let the linear bearings move along the 80/20, the longer side of the 80/20 is utilized 1020, that is 1" by 2", shown in Figure 4.3. The bottom slots are used for the L-shaped brackets to stabilize the frame. 1010, that is 1" by 1" 80/20, is selected for the short side of the frame, where a motor and a pulley are located on the top, so that the belt height is equal to the carrier part to pull.

Linear bearings slide on the 80/20, but utilizing UHMW-PE polymer can reduce the friction. UHMW-PE polymer has a lower coefficient of friction than glass, which is 0.12 when static and 0.016 in kinetic motion, when utilized with an aluminum (Inc., 2015).



Figure 4.3. Pictures of 1010 (Left), 1020 (Center), and a linear bearing (Right) from 80/20 Inc.

#### 4.2.1.3. Styrofoam

Since it is not necessary to seal the chamber yet, all of the side walls are not purchased. However, this would be a part of the temperature system design, to maintain the temperature and to maintain the concentration of gases inside the chamber, insulation type styrofoam (expanded polystyrene) is selected. Because of the thickness that is 1", this styrofoam is quite flexible and thus integrated to the top of the chamber with LED lights, shown in Figure 4.4.

This could be a future consideration. Exact selection of the foam would be dependent on local cost and availability.



Figure 4.4. Styrofoam with LED lights attached.

### 4.3. Positioning System (YZ positioning)

This is an overview of the positioning system and is composed of two sections. The first section is a discussion of hardware identification and architecture and a summary of a carrier description is included in the following section.

Stepper motors are selected for their simplicity and low cost. There are two axis; Y (forward and backward) axis and Z (up and down) axis. By using only 2 axis as opposed to 3, the cost and complexity are further reduced.

The carrier, which is attached to the pulley and belt system, moves forward and backward pulled by a stepper motor. Midpoints of 1010 extrusions at opposite sides are selected to mount a pulley and belt system and Y axis stepper motor. A slot to create a tension is designed in a pulley mount. A rectangular aluminum tube is selected as the carrier, and two Z axis stepper motors are installed inside the rectangular tube. The origin of the Y axis is closer to station 1 (see Figure 4.1). Similarly, Z axis origin is at the bottom of the cup picking up mechanism where limiting switches are placed as a backup from losing steps, shown in Figure 4.5.

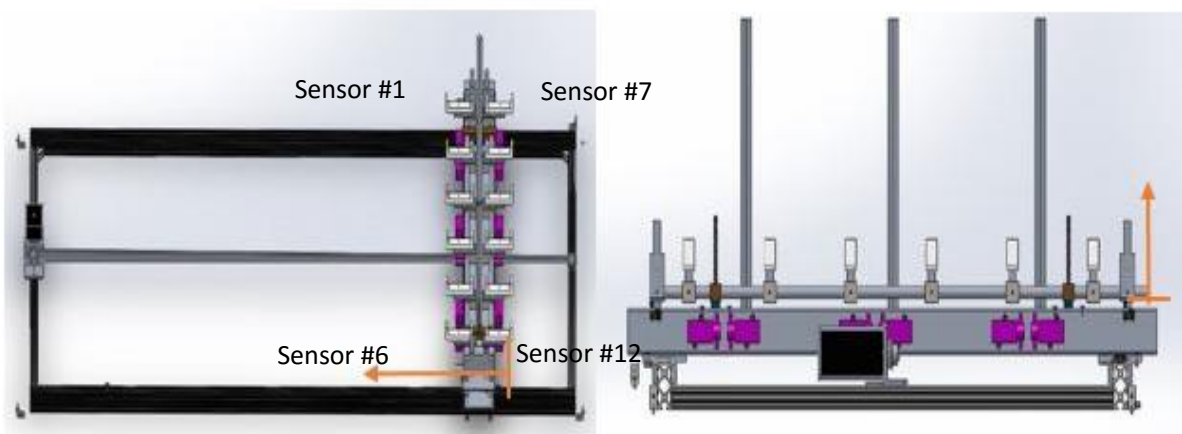


Figure 4.5. Picture of the origin at each axis. Machine is all the way to the pulley side when  $y = 0$  (Left) and all the way to the bottom when  $z = 0$  (Right)

#### 4.3.1. Hardware Identification and Architecture

In this section, all hardware utilized for the positioning system will be identified. All motions are generated from the stepper motor. Y axis applies a pulley-belt system (see Figure 4.6) and Z axis is operated by a linear actuator utilizing a lead screw, shown in Figure 4.7. To prevent from losing steps, 4 limit switches are installed; two in Y axis and two in Z axis. One of the Y axis limit switch is only for an emergency, when the machine doesn't go back to the Y axis origin but instead, it moves forward.

The components relating to this system such as a stepper motor, stepper motor driver, lead screw, belt/pulley, limit switch, and a structure of bearing and vertical linear guide are identified below.

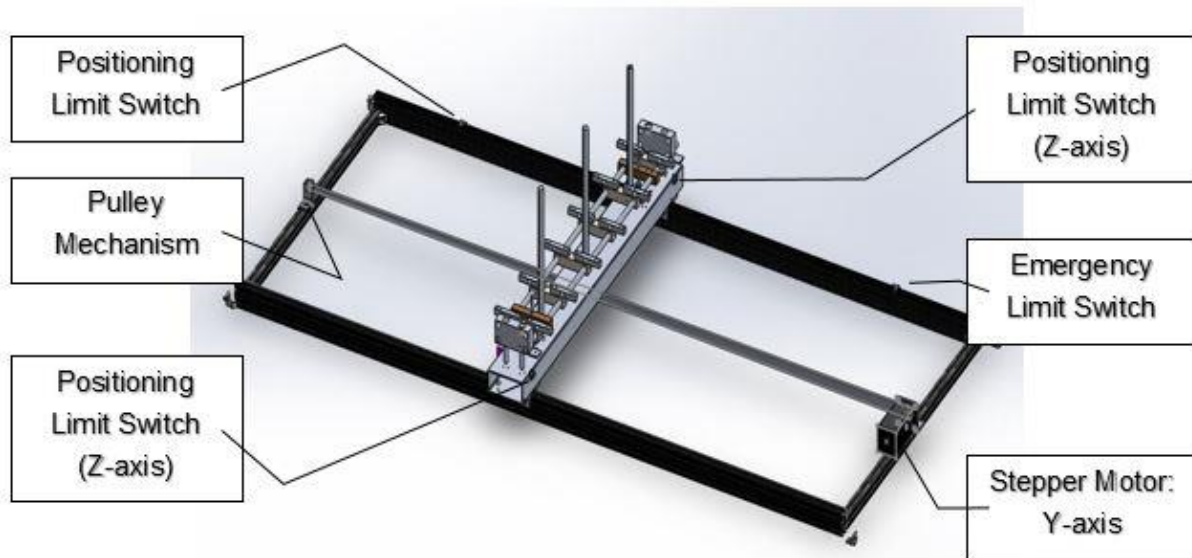


Figure 4.6. Solid Model of Positioning System

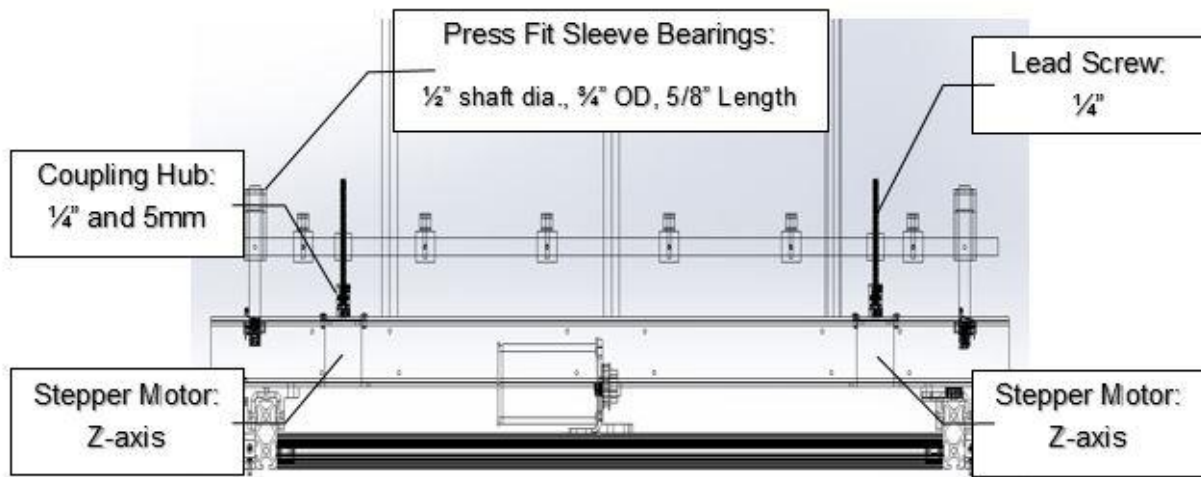


Figure 4.7. Solid Model of Z axis Positioning System

#### 4.3.1.1. Stepper Motor

Stepper motors are known for being simple and robust and capable of high mechanical reliability with a basic accuracy of +/- one step in an open-loop system. In addition, the operation of stepper motors and their associated drive circuits is effectively digital, permitting a relatively simple interface to a digital controller or to a computer (Crowder, 2006). Thus, stepper motors are reasonable selection for this operations.

Stepper motors are selected for both Y and Z directions, to drive the carrier to each station and pick up cups respectively. One motor is utilized for the Y axis and two motors for the Z axis. The stepper motors produce relatively high torque at low speed and the ability to move in discrete steps. Appropriate number of steps, speed and acceleration are commanded through software to make stepper motors ideal for this application. To improve its step accuracy, mechanical solutions using limit switches are introduced. Proper sizing and selection of a motor is ensures performance, reliability, and reduced cost of the equipment. In order to determine the correct motor specifications for the drive mechanism, estimated mass and friction coefficient are considered in order to calculate

the required load torques. Each axis utilizes a different driving method; y axis utilizes a belt and pulley drive, and the z axis utilizes a lead screw drive.

The Y axis motor is a NEMA 23 size and has a 425 oz.in holding torque with 200 steps/rev. It operates with a recommended voltage of 24-48V, and a rated current/phase of 4.2A. This stepper motor is two phases bipolar. The pulley diameter is set based on the motor specification and movement requirements of the machine.

For the Z axis motors, based on the required torque calculation and comparison of its cost, a 68 oz.in motor is selected. Acceleration torque and load torque of each candidates with a different torque and cost are computed to find its total torque. Multiplying the total torque by a safety factor of 3, the required torque is determined. The acceptable ratio of the torque (Torque motor / Torque required) is considered above 3.3. Although the cheapest stepper motor has a ratio of 3.39, the price wise, the 68 oz.in NEMA 17 and the lowest torque with the most inexpensive stepper motor have only USD 2.00 difference. Therefore 68 oz.in (400 steps/rev) motor is chosen (Figure 4.8). This is a 2 phase bipolar motor that has rated voltage of 3V and rated current of 1.7A/phase. The lead screw, that is going to be discussed later, is calculated to have a quarter inch diameter, however, the stepper motor shaft is 5 mm. Therefore, a quarter inch diameter of coupling hub and 5 mm diameter of coupling hub are required (Figure 4.8).

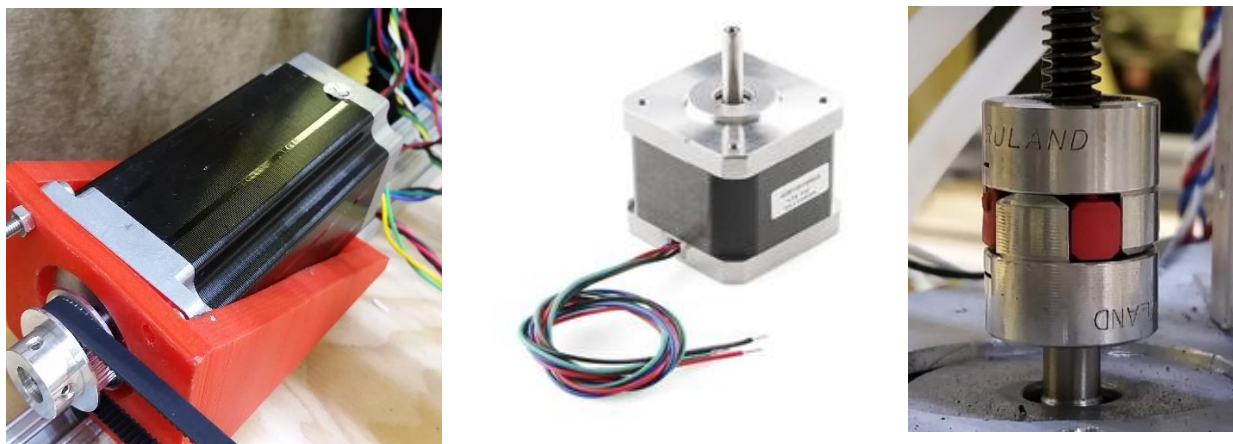


Figure 4.8. Y axis Stepper Motor (Left) from ebay, Z axis Stepper Motor (Center) from Sparkfun, and Coupling Hub and Spider from McMaster-Carr (Right)

#### 4.3.1.2. Stepper Motor Driver

Since the Y-axis stepper motor requires 4.2A rate current per phase and voltage of 24 to 48 volts, ST-M5045 is selected as a stepper motor driver. ST-M5045 handles output current from 1.0A to 4.5A and supply voltage from 20V DC to 50V DC, and thus it is suitable for a wide range of stepping motors from NEMA size 17 to 34. Although Z axis requires two stepper motors and so as stepper motor drivers, ST6600 is relatively inexpensive (USD 8.71 each). Z-axis stepper motor specifications are as stated earlier, rated voltage of 3V and current of 1.7A per phase, whereas ST6600 supply voltage is up to 50V DC and its output current is up to 5.0A.

It was decided that micro-stepping isn't really required for this application, so the lowest micro-step setting is selected for all stepper motor drivers. There are 8 switches on Y axis stepper motor driver and 6 switches for Z axis stepper motor drivers. A switch indicates 1 when it's "on" and 0 if "off". Z axis driver settings are 1 micro-stepping that is equal to one single step and 1.5 A, which leads to "001010". First three switches control number of micro-steps and the last three switches determine an electrical current. Y axis is set as "10011111", which creates 4.0 A and 400 pulse/rev or 2 micro-steps with full current. This driver controls a current with the first three switches and the fourth switch indicates a full current or a half current. The last four switches are for the micro-stepping. Other settings are listed below.

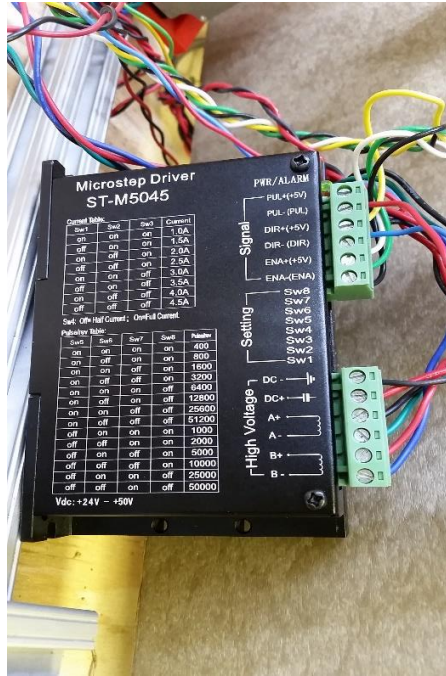


Figure 4.9. Stepper Motor Driver, ST-M5045 (Left) for Y-axis Stepper Motor and ST6600 (Right) for Z-axis Stepper Motors.

Table 4.1. Current and Micro-stepping settings of ST-M5045

Peak Current	Reference Current	SW1	SW2	SW3
1.0 A	0.71 A	1	1	1
1.5 A	1.07 A	0	1	1
2.0 A	1.43 A	1	0	1
2.5 A	1.79 A	0	0	1
3.0 A	2.14 A	1	1	0
3.5 A	2.50 A	0	1	0
4.0 A	2.86 A	1	0	0
4.5 A	3.21 A	0	0	0

Micro-steps	Pulse/rev.(for 1.8°motor)	SW5	SW6	SW7	SW8
2	400	1	1	1	1
4	800	1	0	1	1
8	1600	1	1	0	1
16	3200	1	0	0	1
32	6400	1	1	1	0
64	12800	1	0	1	0
128	25600	1	1	0	0
256	51200	1	0	0	0
5	1000	0	1	1	1
10	2000	0	0	1	1
25	5000	0	1	0	1
50	10000	0	0	0	1
125	25000	0	1	1	0
250	50000	0	0	1	0

\*SW4 is 0 = half current, 1 = full current

Table 4.2. Current and Micro-stepping settings of ST6600

Micro-step	SW1	SW2	SW3	Current [A]	SW4	SW5	SW6
OFF	0	0	0	0.5	0	0	0
1	0	0	1	1.0	1	0	0
1/2A	0	1	0	1.5	0	1	0
1/2B	0	1	1	2.0	1	1	0
1/4	1	0	0	2.5	0	0	1
1/8	1	0	1	3.0	1	0	1
1/16	1	1	0	3.5	0	1	1
OFF	1	1	1	4.0	1	1	1

#### 4.3.1.3. Lead Screw

Lead screws are selected using column strength calculation and critical speed. In design, a lead screw is fixed one end by the motor and the other end is free. In this case, the end fixity factor is 0.25 for column strength calculation. First selection of the length of a lead screw is 6 inches and 1/8 inch diameter. Column strength equation to compute the maximum load is given by

$$P_{cr} = \frac{14.03 \times 10^6 \times F_c \times d^4}{L^2} \quad (1)$$

where  $P_{cr}$  is the maximum load (lb.),  $F_c$  is the end fixity factor, which is 0.25 for one end fixed and one end free,  $d$  is the root diameter of screw (inch), which is 0.125 inch, and  $L$  is the distance between nut and load carrying bearing (inch), which is 6 inch.

From this, the calculation maximum load is computed as 23.8 lb. or 10.8 kg. This is too small for this experiment since mass of a cup is assumed about 0.5 kg and thus total of 6 kg is required just for cups. The mechanism utilizes two lead screws so it could be a half of it but to be safe, another calculation was made with 1/4 inch diameter lead screw. The result is 380.28 lb. or 172.63 kg which is an acceptable value.

The second consideration is critical speed in rpm. The critical speed, N, the root diameter of screw, d, and the distance between nut and load carrying bearing, L, are related by Equation 2.

$$N = \frac{C_s \times 4.76 \times 10^6 \times d}{L^2} \quad (2)$$

The constant number,  $C_s$ , is 0.36 for one end fixed and one end free. The 1/4 inch diameter lead screw is used for this calculation as well. Thus N is 11,900 rpm, which is enough for this experiment.

Therefore, 1/4" – 16 x 6" (L) acme threaded lead screw is purchased. In addition, since stepper motor shaft diameter is 5 mm and the lead screw selected is 1/4", two different diameter of coupling hubs with a spider are purchased to secure the screw.

#### 4.3.1.4. Belt/Pulley

According to the motor specification, the selected motor for Y axis has a 9.5 +/- 0.15 mm shaft diameter. Thus, as a pulley bore diameter (D), 10mm is selected. Both driven and driving pulley sizes utilize exactly the same pulleys. Holding torque of the motor is 3.0 N\*m. The selected pulley has an overall diameter of 30 mm. Torque load,  $T_L$ , using this pulley is calculated with the equation below.

$$T_L = \frac{FD}{2\eta_i} [Nm] \quad (3)$$

$$F = F_A + mg(\sin \theta + \mu \cos \theta) [N] \quad (4)$$

Assuming total mass of a carrier, m, is about 10kg. Since there is no external force,  $F_A$  is negligible and also there is no angle in this system,  $\theta = 0$  and thus,  $\sin \theta$  is also eliminated. A friction coefficient of sliding surface is assumed to be 0.05 (Motor, n.d.). Therefore, the force of moving direction is  $mg\mu$ , which is 4.905 N. By substituting the

force into the torque load equation, and using efficiency,  $\eta = 0.85$  and gear ratio,  $i = 1$ , the required torque load is computed as  $T_L = 0.08656$  [Nm]. Other specifications of the pulley is below.

- Tooth number: 40
- Tooth pitch: 2mm
- Bore diameter: 10mm
- Flange: double
- Tooth width: 6.5mm
- Setscrews: M4

Based on the pulley diameters, following formulas is used to calculate belt length:

$$Length = 2 * C + \frac{\pi(D_1 + D_2)}{2} \quad (5)$$

where C is center to center distance between shafts and  $D_1$  &  $D_2$  are pitch diameter of pulleys which are the same in this case. Thus the equation is simplified to:

$$Length = 2 * C + \pi * D. \quad (6)$$

A center to center distance, C, is 1.803 m and pitch diameter of pulleys, D, are 0.03 m. Thus the required length of a belt for this system is 3.701 m. From ebay, GT2 timing belt with 6 mm width and 2 mm pitch for 5 m is purchased.

#### 4.3.1.5. Limit Switch

As explained before, the limit switches serve an important role in this machine's reliability. Stepper motor loses about 4 steps every cycle. The experiment and analysis

of motor step loss is discussed in chapter 7.3. In order to account for the loss of steps, limit switches are installed. The result without limit switches are also discussed in chapter 7.

Micro roller lever arm limit switches are chosen for their availability. They are inexpensive, about 14 cents apiece. These utilize a power from Arduino through 470 ohm resistors and each of them is connected to Arduino pins.

In the beginning of this section, the locations of limit switches are briefly explained. 2 limit switches are placed at the Z-axis origin and on the Y-axis motor side face of the square tube at where vertical linear guides are. Both switches are controlled in the software to verify that the two motors are synchronized and stop when it's pressed. Normally open is used for all switches. Y-axis limit switches are set on the left side of 80/20 when the view of the machine is from a pulley side, see Figure 4.6. One is at Y-axis origin and the other one is at the other side for an emergency. In software program, when the machine hits any limit switches, the machine bounces back a little bit. Especially Y limit switch at origin is placed to stop the machine when it returns from the last station. Thus, to avoid the damage from the sudden shock due to the speed of machine and to make the machine smooth, after the machine bounces back, it slows down to move forward to touch the switch again to stop completely.

#### 4.3.1.6. Bearing and Vertical Linear Guide

In order to lift the cups, a mechanism needed to be designed for the task. Seen in Figure 4.10, on both axis two rods are inserted to an aluminum block after machined. Since there is vertical motion, bronze sleeve bearings are press fit to the aluminum bar to create motions smooth. The other axis has no movement and thus set screws are installed to fix rods. On those aluminum rods, 6 steel blocks are placed at regular intervals, and 2 bronze blocks, which is threaded with an acme tap for lead screws, are placed in between the first and third steel blocks from the both ends. All blocks are machined based on the solid design. Each steel blocks carries 2 load cells.

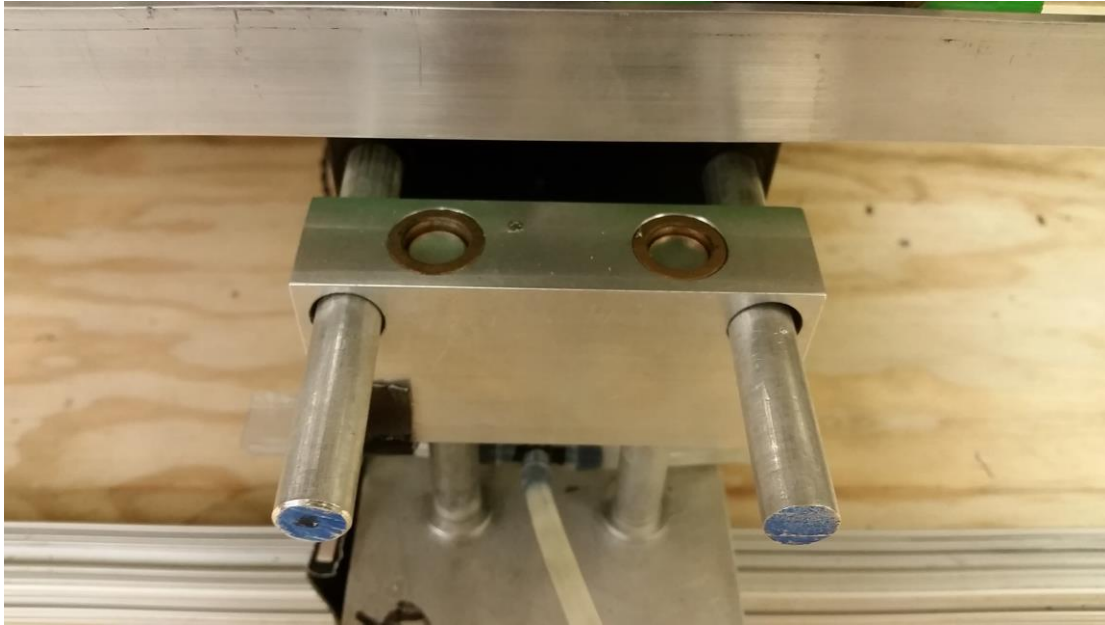


Figure 4.10. Bearing and Vertical Linear Guide Installation

#### 4.3.2. Carrier

The carrier is one of the major components in this machine. It is driven by the stepper motor and pulley system, and it carries the entire water delivery system, discussed next, as well as weighing system and the Z-axis positioning system. For the carrier, 1-3/4" x 4" x 48" aluminum 6063 rectangular extruded tube is chosen.

Along the side of the tube, 12 solenoids (6 on one face and the other half on the other face) are mounted. The bottom of the tube is machined, and cut square holes for the stepper motors, and two of stepper motors are installed, and secured inside the tube, shown in Figure 4.7. 12 relays and 2 stepper motor drivers are attached on the bottom of the square tube. Then to secure the water tubes, 3 of 1" x 1" square extrusions are mounted on the top and water tubes are zip tied along the extrusions.

Carrying all the components, both ends of the square tube are bolted down to the linear bearings. Timing belt is attached in the middle of the tube and tightened.

#### 4.4. Water Delivery System

This section 4.4 discusses water delivery system. The watering mechanism simply utilizes gravity. A bucket, filled up with water outside a chamber, is located on a high elevated place. While a solenoid is open, water comes out from a plastic tube and stops when the solenoid is closed. Four watering tubes are put together and tied up along an aluminum square tube. Three of the square tubes are screwed on the carrier at even intervals. The watering tubes are brought above each cups and attached to the 3D printed tube supports to stabilize them. Referring to 3D printer, all the cup holders, which are attached between each load cells and cups, are also designed as 3D printed materials.

##### 4.4.1. Hardware Identification and Architecture

In this section, all the hardware utilized for water delivery system is identified. The discussion starts with the valves and plumbing and the relay discussion follows. The water reservoir is briefly discussed in this section, with a more detailed discussion following in the next section 4.4.2. A solid model of the system is shown in Figure 4.11.

The mechanism is simple in operation. Solenoids receive power from Arduino through relays, and when the valve is open, water is delivered from water reservoir through water tube, utilizing pressure from gravity. Through some experiments, the water amount error is about +/- 1g. To secure water outputs, cross shape water tube mounts are designed and 3D printed, and all electronics are placed on the top of it.

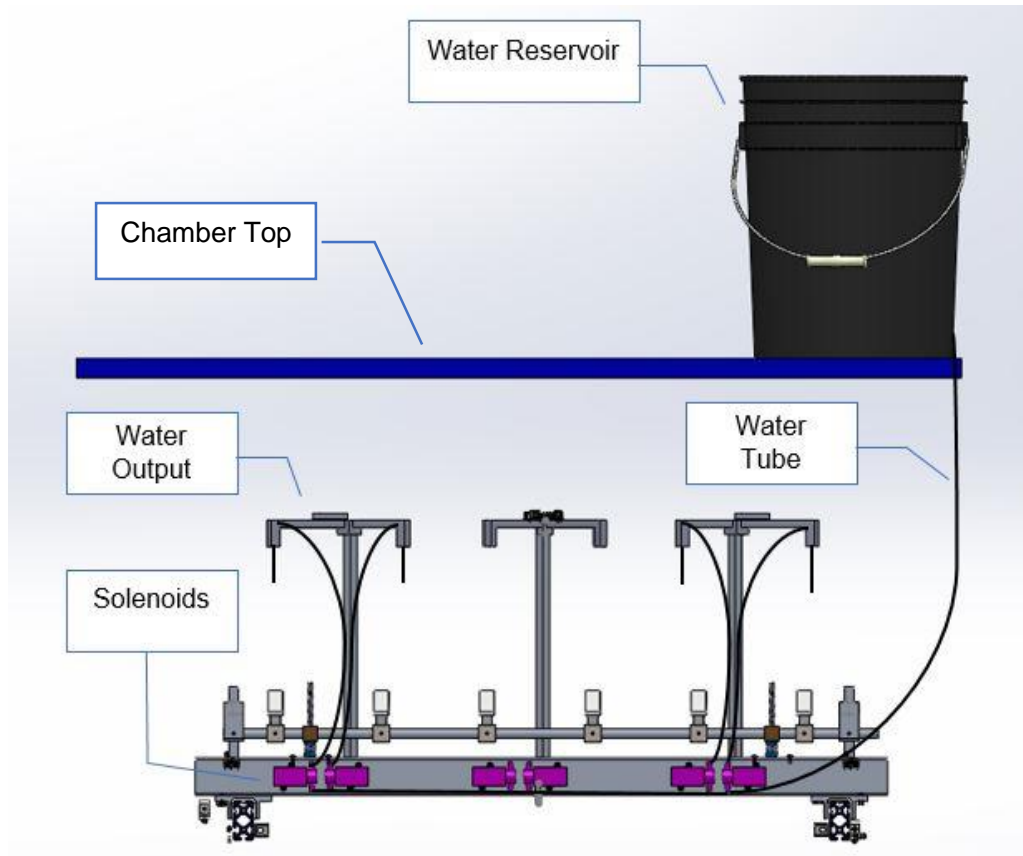


Figure 4.11. Solid Model of Water Delivery System

#### 4.4.1.1. Valves (solenoids) and Plumbing

12 Solenoid valves and 3.5 mm x 10 m length silicon tubing is purchased from TrossenRobotics. For the testing purpose, it is purchased from them; however, focusing on cost efficiency and installing 12 solenoids for USD 19.95 each needs to be reconsidered carefully before making a decision for the future work. Moreover, it is controllable by Arduino, just like other sensors.

This solenoid requires 12 V and 3 A current, and its inlet/outlet width is 6 mm. As following recommended method, the valve is controlled by applying power through a relay, purchased from the same company and will be discussed next. When power is not going

through, the solenoids is closed and water will not flow. Contrary when it's powered, the valve will open and water will flow freely.

#### 4.4.1.2. Relay

In order to operate the 12 V solenoids using the 5 V Arduino signal, the RobotGeek relay is selected as they recommended. These relays can handle up to 5 A at 250 V, thus this is the best fit for the solenoids that requires 3 A and 12 V. One side of the relay that control a switch is connected to 12 V power and a wire from a solenoid valve. The other side of a solenoid wire is connected to a power supply GND. The second part of relay is connected to Arduino. S is for an Arduino pin, V is for a 5 V power from Arduino, and G is an Arduino GND.

Although this relay is also reliable and relatively inexpensive by itself, it still might be a better decision if a transistor is chosen to reduce the number of wires and save some Arduino pins. This selection is considered in the future work.

#### 4.4.2. Water Delivery Method

As mentioned earlier, a water reservoir is located outside the chamber, on the top of the styrofoam in this project. This system relies only on pressure from gravity, and thus a reservoir needs to be placed high relative to the point of delivery. In the future, a dedicated fixture would help mount this reservoir as opposed to sitting it atop the chamber. It is reliably working so far as long as water is in and it is at high enough place.

#### 4.5. Power Supply

In this section, the power requirements for the positioning and watering system are discussed. By now, components that need some power from a power supply have been indicated. In Table 4.3, a summary of power budget is listed to determine a correct power supply that can handle all component's voltage and current requirements. Those which utilize Arduino 5 V is not listed in Table 4.3, and Arduino power requirement is discussed in the following chapter.

Based on a voltage requirement, 24 V and 12 V power supplies are used. 24 V is connected to all stepper motor drivers and 12 V is required for solenoids. Drawing of power connections for each components are described in Figure 4-12. Other parts such as limit switches and relays are powered by Arduino 5 V, where Arduino  $V_{in}$  is connected to a 5 V power supply (discussed in the next chapter).

Table 4.3. Power budget of positioning and watering systems to select a power supply

<b>Part</b>	<b>Qty</b>	<b>Rated Voltage (V)</b>	<b>Rated Current (A)</b>	<b>Power (W)</b>
Y axis Motor (23HS45-4204S)	1	24 ~ 48 V	4.2 A	100.8 ~ 201.6 W
Z axis Motor (ROB-10846)	2	3 V	1.7 A/Phase	10.2 W
Y axis Motor Driver (ST-M5045)	1	24 ~ 50 V	1 ~ 4.5 A	24 ~ 225 W
Z axis Motor Driver (ST6600)	2	~ 50 V	~ 5.0 A	~250 W
Solenoid	12	12 V	3 A	36 W

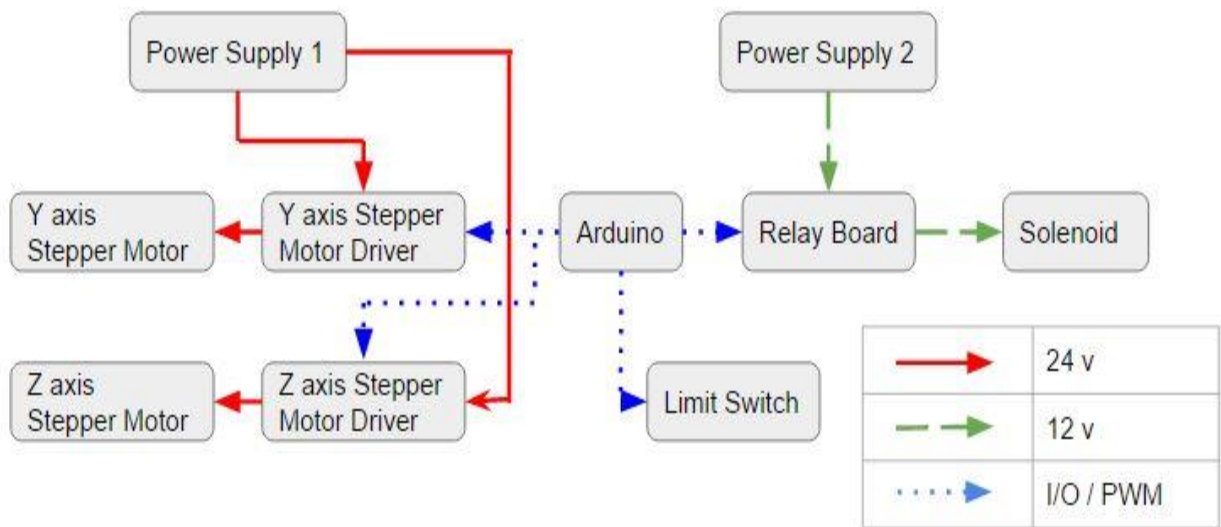


Figure 4.12. Power Connections for Each Components

## CHAPTER 5

### ELECTRICAL SYSTEM

#### 5.1. Electrical Design Theory

This chapter starts with discussions of a description of electrical design theory that consists of an overview of weighing system, data logging method, electronics location, and summarized software system architecture. Following the overview, hardware identification and architectures, power supply, circuit design, and software development are explored.

##### 5.1.1. Overview of Electrical System

Another system within the automated soil humidity machine is the weighing system. As found in literature reviews, weighing cups to determine the humidity is selected instead of utilizing a humidity sensor. Once again, the target weight accuracy is 0.1 g. By default, a load cell provides measurement information with an analog voltage. In order to increase its accuracy, an analog to digital converter (ADC) is added between the Arduino and the load cell. A selected ADC can be purchased from ebay and it allows the number of bits to increase to 24 or 32 bits from 8 bits. There is an Arduino library for this ADC, and it is discussed in the software development section.

For every measurement, the mass, timestamp, station number, and sensor number are stored in a MicroSD card through MicroSD shield. Currently, all the data in the MicroSD card is transferred from the machine to the computer once a week and the MicroSD card is cleared. Without the MicroSD card, the machine loses the ability to save any data. By introducing the SD card to save the data, a computer doesn't need to be connected all the time, and it will help if the computer suddenly shuts down and must be restarted as well as if memory space is exhausted. The chosen microSD shield is stackable on an Arduino Uno; however, due to the number of pins required for the entire

machine, an Arduino Mega is utilized, and thus determining the right pins for the Mega is required. Pins description is shown in Table 5.1.

An additional consideration for the electronics is their locations. Although the machine doesn't spill water, it is important to protect electronics from any physical accidents. Therefore, all electronics are placed either above the watering system or behind the carrier.

This is a brief overview of the software architecture that is discussed in detail later in this chapter. Machine operating order starts with the positioning system, proceeds through the home cycle, then tares for weighing system. Once tare-ing is done, the machine picks up a set of cups and begins measuring weight one by one. First, the load sensors measure the weight of the first cup in the series and decides if the cup requires humidity. If no water is needed, the second sensor starts operating. If the cup must be watered, the solenoid turns on and executes a watering cycle. While watering a cup, it keeps measuring its mass to find out when to stop. Then the next sensor activates and does the same thing. When all 12 sensors have performed their measurements, the machine activates the positioning system and moves to the next station. Once all stations and all cups have been treated, the machine returns to the home cycle and starts over again.

Table 5.1: Arduino Digital Pin Map

<b>Parts</b>	<b>Mega Pin #</b>	<b>Parts</b>	<b>Mega Pin #</b>	<b>Open Pin #</b>
Y Stepper (PUL - )	2	ADC 12 (DT)	24	0
Y Stepper (DIR - )	3	ADC 12 (SCK)	25	1
Relay 1	4	ADC 11 (DT)	28	19
Relay 2	5	ADC 11 (SCK)	29	26
Relay 7	6	ADC 6 (DT)	32	27
Relay 8	7	ADC 6 (SCK)	33	30
Relay 5	8	ADC 8 (DT)	34	31
Relay 6	9	ADC 8 (SCK)	35	38
Relay 11	10	ADC 5 (DT)	36	39
Relay 12	11	ADC 5 (SCK)	37	46
Limit Switch Z2	12	ADC 1 (DT)	40	47
Limit Switch Z1	13	ADC 1 (SCK)	41	49
Y Stepper (ENA - )	14	ADC 7 (DT)	42	/
Z2 Stepper (PUL - )	15	ADC 7 (SCK)	43	/
Z2 Stepper (DIR - )	16	ADC 2 (DT)	44	/
Z1 Stepper (PUL - )	17	ADC 2 (SCK)	45	/
Z1 Stepper (DIR - )	18	MicroSD Shield (Pin: 8)	48	/
MicroSD Shield (SDA: A4)	20	MicroSD Shield (Pin: 12)	50	/
MicroSD Shield (SCL: A5)	21	MicroSD Shield (Pin: 11)	51	/
Limit Switch Y2	22	MicroSD Shield (Pin: 13)	52	/
Limit Switch Y1	23	MicroSD Shield (Pin: N/A) *	53	/

\* Needs to be open. Nothing is connected on the Mega side but in Arduino code, pin 53 is set as output, (pinMode(53, OUTPUT))

## 5.2. Hardware Identification and Architecture

In this section, all hardware for an electrical system is identified. The core of the machine, microcontroller, load cell, and ADC from weighing system, and microSD shield for a data logging are discussed.

### 5.2.1. Microcontroller

This system configuration utilizes 8 sets of sensors (each set requires one pin for the relay and two pins for the ADC) which along with the motors, require a total of 42 digital IO pins on the Arduino Mega. Details are shown in Table 5.1. If the system had the full intended compliment of 12 sets of sensors, it would require 54 total pins to operate all sensors and electro-mechanicals. The Arduino Mega has 54 digital IO breakouts, however 2 of them (#0 and #1) are already being used for serial communications. Therefore, in order to support all the devices in the system, 2 Arduinos are used. This is more complicated than it needs to be, and for the future work, relays would be better to be replaced with transistors.

An alternative, the Raspberry pi microcomputer was brought up as another option. However, Raspberry Pi requires knowledge of Linux as well as programming such as Python, whereas the Arduino is simpler, harder to damage and has more learning resources and libraries. The Arduino is chosen, because of its simplicity.

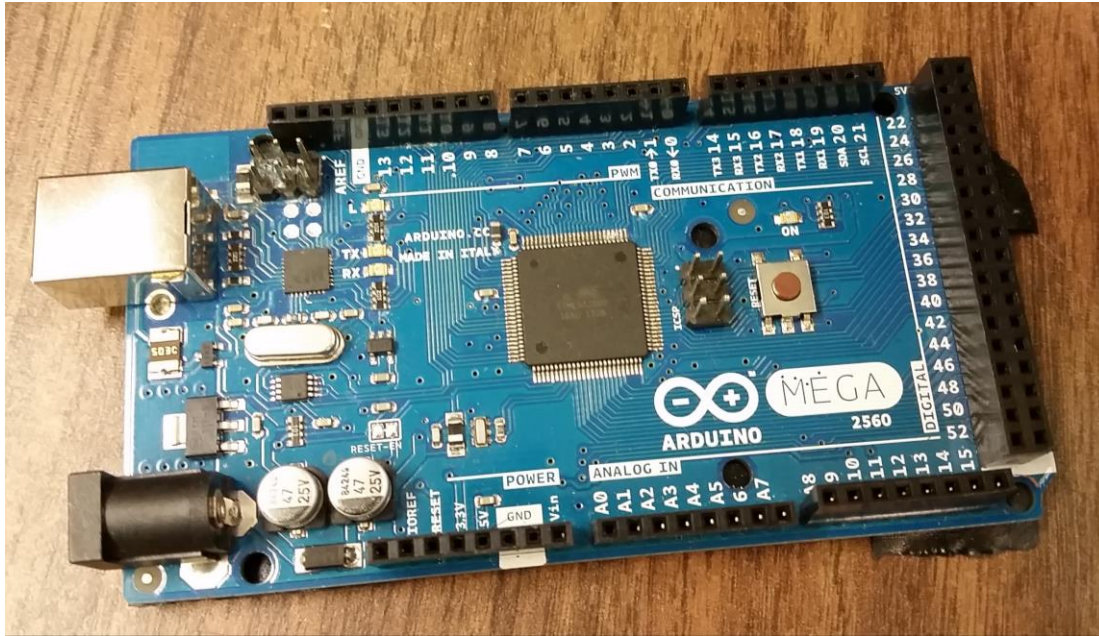


Figure 5.1. Arduino Mega from Sparkfun

### 5.2.2. Load Cell

Regarding the load cells, since a typical 4 wire cell is inexpensive and available, the design of the machine is considered efficient and as a result, twelve of cells are installed to save some operating time. A maximum cup weight is assumed to be 500 grams due to the size of the selected seeds (see Chapter 7) and thus 1 kg weighing sensor is selected to give a double margin. Although its accuracy is 0.03 % Full Scale (F.S), which means potential error of 0.3 g in its analog form, the design requirement of the weight accuracy is 0.1 g; therefore, introducing an ADC is required. According to Colm Slattery and Mariah Nie (Slattery & Nie, 2005), the ADC accuracy is required to be close to 20 bits to meet a higher resolution requirement. More about ADC and Arduino library functions are discussed in the later section. The load cell wires are too short for this operation, thus extension wires are soldered.

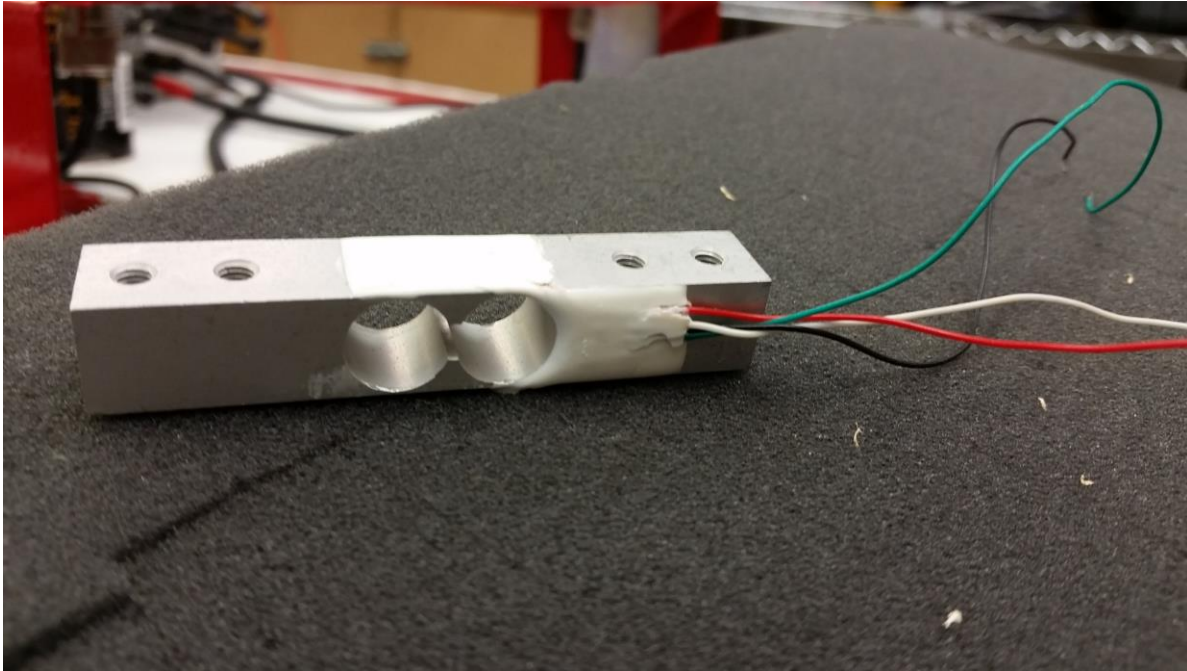


Figure 5.2. Load Cell from ebay

### 5.2.3. ADC

The 24 bit analog to digital converter (ADC – chip HX711) is used to get measurable data out from the load cell and the strain gauge. This item is introduced because the analog load cell output doesn't have the precision that is required for this project. By connecting load cell to this ADC, it can obtain 24 bits of readings that are ideal for high-precision measurements. According to its datasheet (Semiconductor, n.d.), this ADC has two analog input channels of which one can be programmed with a gain value of 128 or 64. Unfortunately the second channel has only a fixed amplification factor of 32 and it's too small for this project, so it still requires to have 12 ADCs, one ADC for each load cell, instead of two load cells with one ADC. There is also an Arduino library to easily control an ADC.

The four connecting wires that come out from the wheatstone bridge on the load cell (left four pins from the top in Figure 5.3) are below:

- Excitation+ (E+) → Red
- Excitation- (E-) → Black
- Amplifier- (A-) → Green
- Amplifier+ (A+) → White

The other side of ADC (right 4 pins in Figure 5.3) is connected to an Arduino.

- GND → Arduino GND
- DT → Arduino pin\*
- SCK → Arduino pin\*
- VCC → Arduino 5V power

\* Arduino digital pin numbers are indicated in Table 5.1.

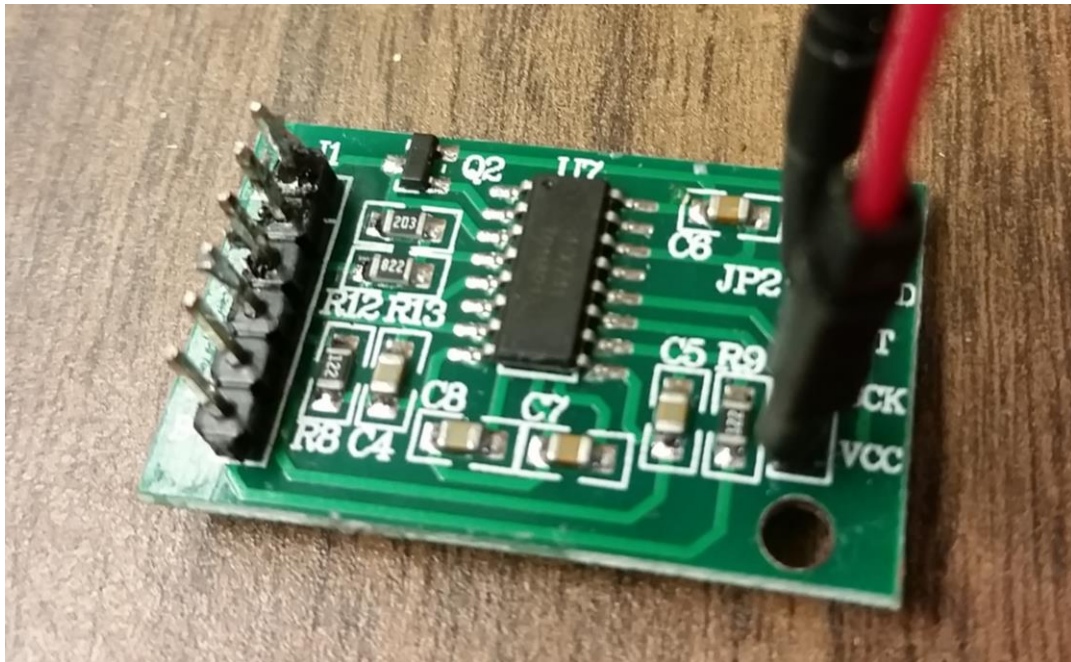


Figure 5.3. ADC – chip HX711 from ebay

#### 5.2.4. MicroSD Shield

A MicroSD shield is the last component integrated into the machine. The purpose of its use is to keep data logged and saved in an external device that has higher capacity and greater flexibility than the Arduino serial command.

This MicroSD shield from Sparkfun is stackable on the Arduino Uno. However, since we choose Arduino Mega, it's not stackable and needs a different connection configuration. Thus, important pins need to be changed. First, SDA and SCL lines on Uno (A4 and A5) need to be connected on Mega side of digital pin #20 and #21 respectively. SDA and SCL jumpers are by default connected to the A4 and A5 on the Arduino Uno for I<sup>2</sup>C communication. Then a non-PWM pin on the Mega is required so digital pin #48 is chosen. Communication with microSD cards is achieved over an SPI interface, which requires SPI pins (digital pin #11 - #13). The digital pins 11 through 13 are replaced with digital pins 50, 51, and 52 respectively on the Mega. What is found through this test is that digital pin #53 on the Mega needs to be open. In Arduino code, pin 53 is called but works without any wires connected. All corresponding pins are summarized in Table 5.1.

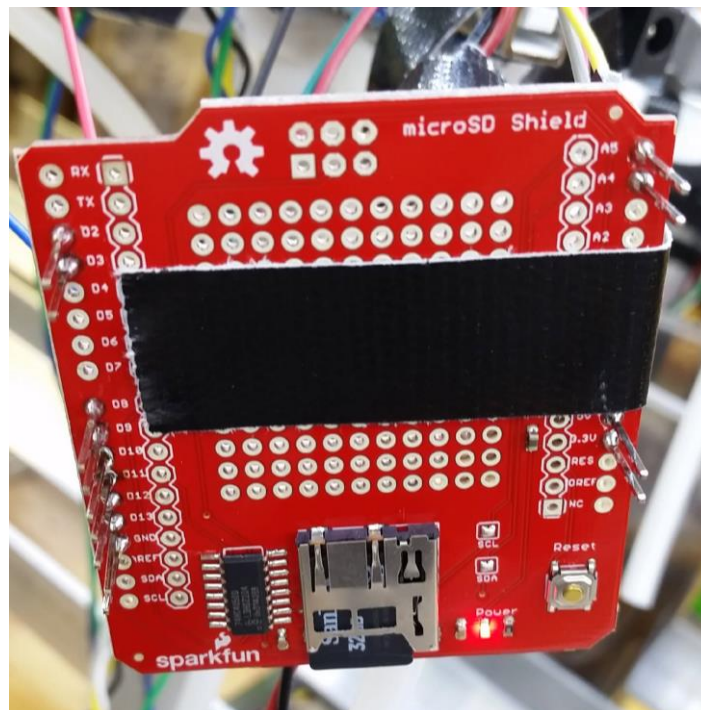


Figure 5.4. MicroSD Shield from Sparkfun

### 5.3. Power Supply

In this section, all power requirements for weighing system and data logging as well as microcontroller are discussed. In Table 5.2, a summary of power budget is listed to determine an appropriate power supply that can handle all the components' voltage and current requirements. Instead of using a computer power or a battery to power the Arduino, a 5 V power supply is introduced in the system and connected to Arduino  $V_{in}$ . The Arduino 5 V and GND are connected to the breadboards' rails to distribute powers to the ADCs, the relays, and the limit switches. The MicroSD shield utilizes Arduino 3.3 V power directly. Simplified power lines diagram to each weighing and data logging components is shown in Figure 5.5.

A summary of the total power system shows that this automated soil humidity machine requires 3 different voltages; 24 V for positioning system, 12 V for watering system, and 5 V for everything else. For the future work, it might be a consideration to utilize voltage regulators or a power supply that can handle three different voltages at once.

Table 5.2. Power budget of weighing system and other electronics

<b>Part</b>	<b>Qty</b>	<b>Voltage (V)</b>	<b>Current (A)</b>	<b>Power (W)</b>
Arduino Mega	1	7 ~ 12 V	20 ~ 50mA	140 ~ 600 mW
Load Cell	12	5 ~ 10 V	/	/
ADC	12	2.6 ~ 5.5 V	~ 1.5mA	3.9 ~ 8.25 mW

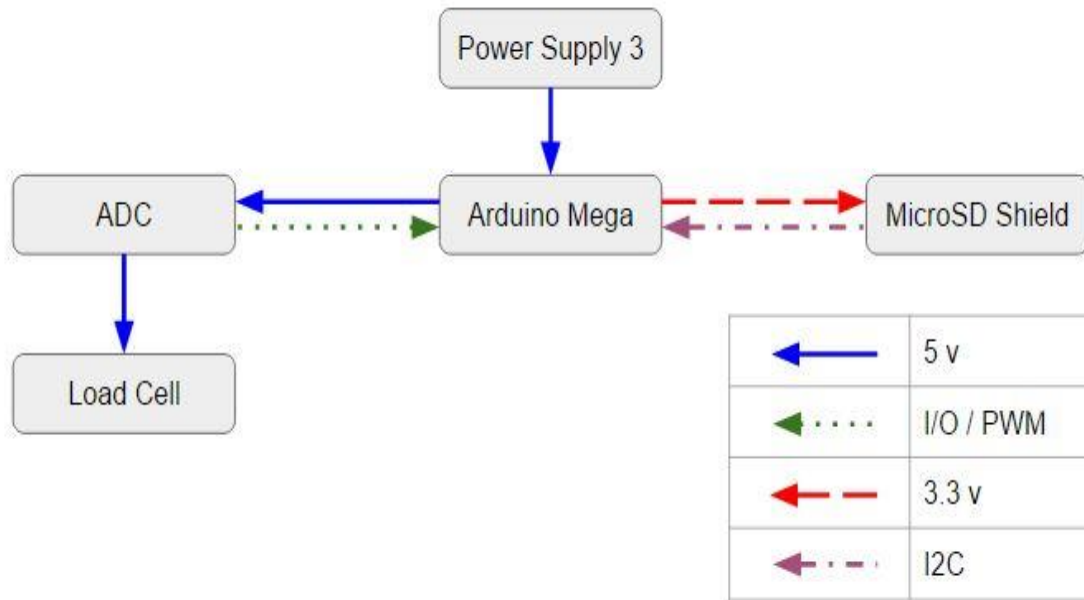


Figure 5.5. Weighing and other power connections

#### 5.4. Circuit Design

Simplified Eagle file schematic is shown in Figure 5.6. Instead of indicating each 12 sensors, it shows one of each components as an example and labels the pin numbers for Arduino inputs.

In the center of the schematic is an Arduino Mega, and that is connected to the top two systems: two at the top left are stepper motor drivers and on the right of motor drivers is an ADC. The left side of an Arduino Mega is a limit switch and on the right is a relay. The bottom part is a microSD shield. Eagle files of each components are downloaded from Sparkfun or RobotGeek and are combined or drawn directly.

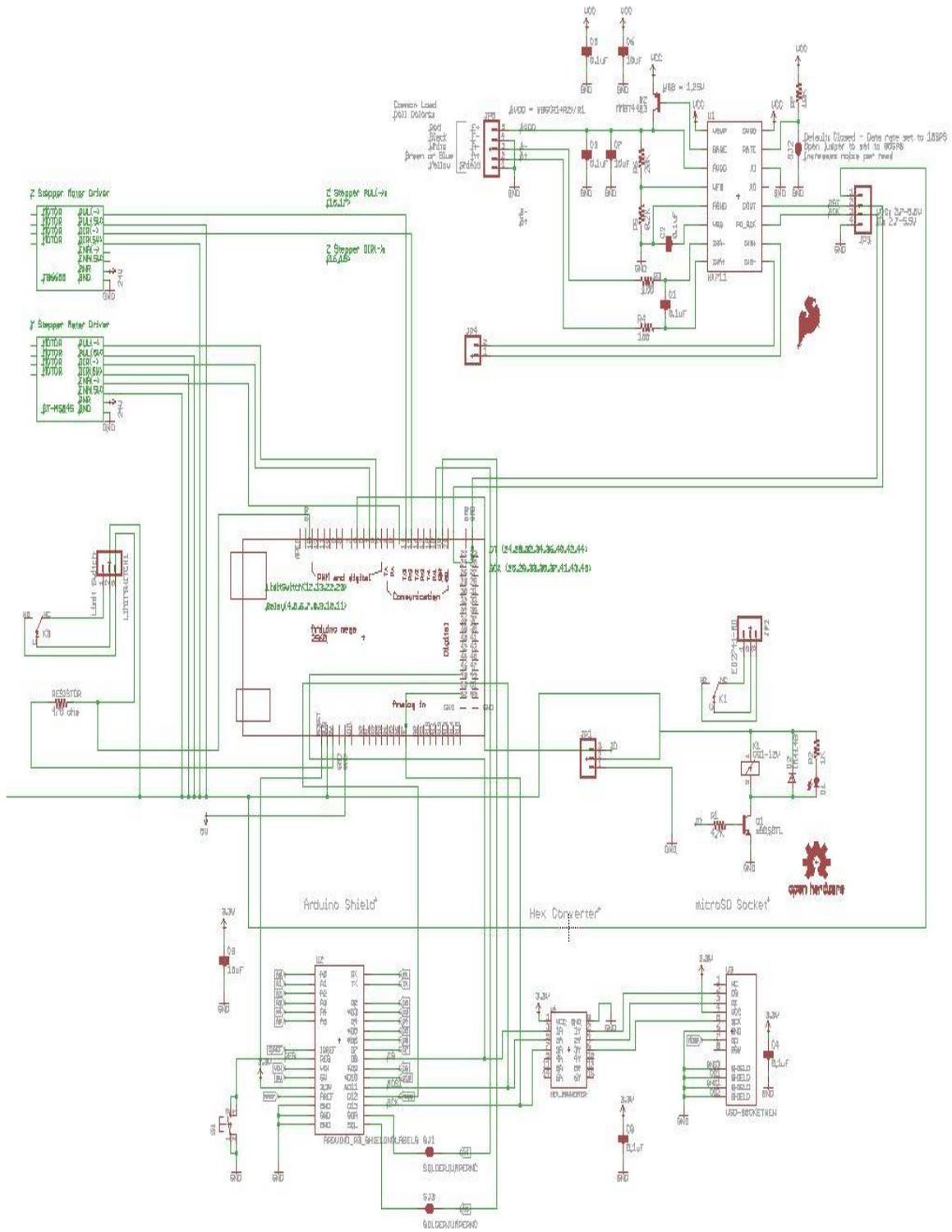


Figure 5.6. Schematics of all wire connections in Eagle file

## 5.5. Software Development

This section discusses the software development, including the programming method, introducing Arduino libraries, and data logging method as well as an Arduino Mega digital pin map. A brief flow-chart is shown in chapter 3, and an implemented detailed flow chart is discussed in this section. The final Arduino code is explained and included in Appendices A.

### 5.5.1. Flow chart

The detailed flow chart (shown in Figure 5.7) starts from after the load cells are tared and the motor moves up for picking up cups and ends after the operations and the motor moves down.

The first process is measuring the cup's mass, then saving the mass on the microSD card after creating a new text file during the first cycle. A random target mass map is created separately to test the machine reliability before the experiment and inserted into the program. A security mass is, on the other hand, calculated in the code based on the target mass and a margin. The relationship between the security mass, target mass, and margin in the code is created as Eq. 7,

$$Security = |Target - Margin|. \quad (7)$$

The first cycle has a margin of 67 g, because the random target mass is either +20 g, +40 g, +60 g, or 0 g heavier than the original weights, thus at least maximum of 60 g needs to be added in some cups. For all following cycles, a margin of 7 g is set since no cups are supposed to be filling another 20 ~ 60 g.

The machine comes to the first decision at the first rhombus (see Figure 5.7) to determine if the mass measured is lighter than a security mass and heavier than a target mass plus 7g. 7 g is added to prevent a cup from overflow, because if the weight is larger

than the target mass + 7 g, the watering system doesn't activate or stops. If the measured mass is in between the security mass and the target mass + 7g, then follows "No" arrow to the third rhombus. If the measured mass is either smaller than the security mass or heavier than the target mass + 7 g, then "Yes" to weight a cup again. The system saves the mass in the microSD card and replace the old mass with the new mass. The arrow follows to the second rhombus. This means that there is a probability of failure of measuring cups. At the second rhombus, when the new mass is heavier than the target mass + 7 g and the answer is "Yes", the cup is skipped and the following sensor starts weighing the next cup. If the new measurement is less than the target mass + 7 g and the answer is "No", it compares again if the mass is less than the security mass at the third rhombus. When this test is completed, the machine starts to treat the cup. At the third rhombus, if the mass is less than the security mass and the answer is "Yes", the machine assumes the cup is tipped over or an error situation has occurred, and skips the cup. If it passes the test and follows "No", the watering system turns on and keeps watering system activating until the mass reaches the target mass. During the process, the machine constantly in the loop, measures a new mass, saves it to the microSD card, and compare the measured mass with the target mass at the fourth rhombus. As soon as the measured mass reaches the target mass, the watering system shuts off and saves the final mass to the microSD card. Then the Z-axis motor returns to the Z-axis origin. By modifying the code with the redundant measurements, the machine become more reliable by not destroying plants with over-watering.

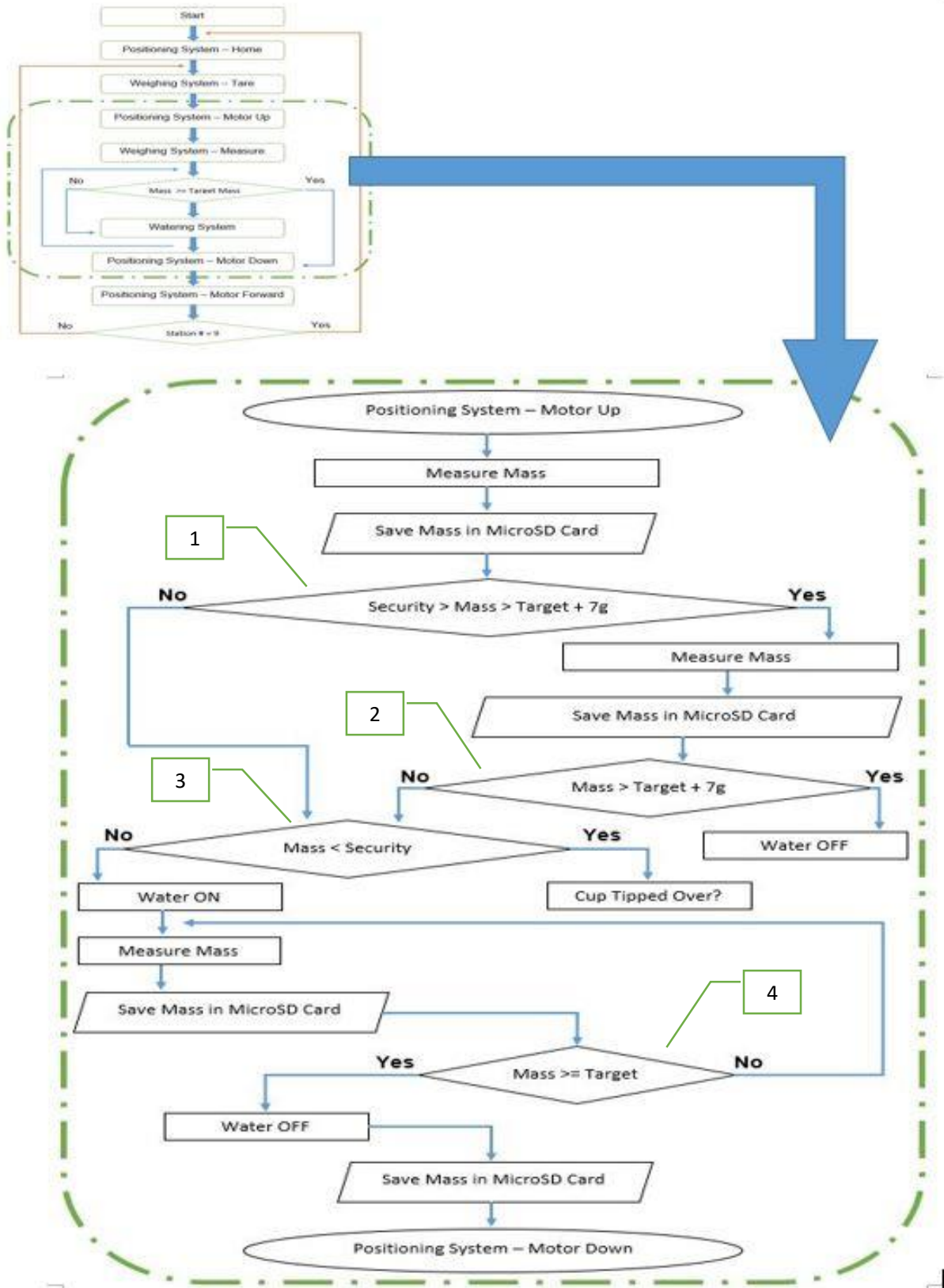


Figure 5.7. Detailed Software Flow Chart

## 5.5.2. Microcontroller – Arduino

The hardware of the microcontroller has already discussed in section 5.2. In this section, two discussions about the Arduino are introduced; libraries and Arduino Mega digital pin map.

### 5.5.2.1. Library

The libraries utilized in the code for the automated soil humidity machine are listed below:

- AccelStepper.h
- HX711.h
- SD.h
- SPI.h

SD.h and SPI.h are the libraries that's installed in the Arduino programming language and development environment by default. The other two libraries are downloaded from GitHub. AccelStepper.h library is to control stepper motors and HX711 is for the ADC. SD.h and SPI.h are used with microSD shield.

AccelStepper.h has useful functions that can control the speed, and the acceleration of each motor, and also all of the motor commands. In addition, there are disable and enable functions for a motor that allow it to power off while it's not operating. Especially the Y axis motor, if the motor is not disabled, under some circumstances the heat from the motor can damage the motor and decrease reliability. While experimenting without the disable function, Y axis motor gets extremely hot and stops operation all of the sudden.

HX711.h is also useful library for the weighing system. One of the functions allows a gain value to be set for the ADC, and allows the load cell calibration to be made

easier. Another helpful function is the tare function, which resets the scale to 0. Without a tare, every measurement is not repeatable over a number of cycles. This function is used in the code and called right before the machine picks up cups for measurements.

#### 5.5.2.2. Arduino Mega Digital Pin Map

Table 5.1 shows all Arduino Mega pins used and open pin numbers. The parts column shows what each pin is used for and its corresponding pin number. Total of 42 pins are used and 12 pins are still available. As mentioned earlier, pin #53 is left open but nothing should be attached to it because in the code, pin #53 is called as an output – pinMode (53, OUTPUT) and it's better to avoid not to make the Arduino confuse.

#### 5.5.3. Data Logging

The software side of detailed data logging is discussed in section 5.5.1. Every mass measurement is saved onto the microSD card right after it weighs a cup. For the purpose of analyzing and examining reliability, the machine is stopped and data is exported to the computer every week. Each line of data is composed of:

- Time stamp [sec],
- Station number,
- Sensor number, and
- Measured Mass [g]

Implementing the SD card storage increases the reliability of the data storage. Should the system lose power, the SD card will preserve the data as long as it is not in the middle of a data export. However, if the machine shuts down accidentally and

powers on again, it restarts the data indexing from the beginning; therefore, the time stamp becomes 0, station number is 1, and sensor number starts from 1. If no one is present while a power outage happens, it might be hard to determine this occurrence until a user reviews through the data.

## CHAPTER 6

### PROTOTYPE DEVELOPMENT AND CONSTRUCTION

The major topic of this chapter is a review of the assembly of the automated soil humidity machine described by solid modeling and actual construction. An overall budget is also discussed in this chapter. In section 6.1, all assembly are demonstrated by SolidWorks and are described how they should be constructed. Actual construction is discussed in the following sections with considerations and analysis of what needed to be changed from the solid model. The last section of this chapter presents the overall cost of the machine.

#### 6.1. Solid Modeling Assembly

The system was designed from the bottom up in solid modeling software, SolidWorks. This section will detail the system broken up into the frame and carrier subassemblies, followed by the completed prototype assembly. Each section discusses how the optimal machine should be built.

##### 6.1.1. Frame assembly

The solid model of this section is shown in Figure 6.1. On the top of a wooden panel base, 80/20 extrusions are placed and fixed by 1" x 1" inside corner brackets from 80/20 Inc. 80/20 linear bearings are simply slid onto these extrusions. Subsequently, outside poles are added to support a styrofoam ceiling with LED lights. Since the machine doesn't need to be sealed by the styrofoam, those panels are substituted by these poles to support the roof styrofoam.

After the pulley mount, that is originally an aluminum sheet, is machined based on the solid model design, and the Y-axis stepper motor mount is 3D printed. Y axis positioning motor and the pulley system are located on the both sides of 1" x 1" 80/20 extrusions. The Y axis stepper motor and pulley system are also secured with 80/20 fasteners. Additionally, the pulley mount is designed with a slot that provides the capability to increase and decrease the tension in the pulley system. The shoulder screw holding the pulley is secured to this slot and fixed by a pair of nuts. The pulley is pulled to tighten the belt before the nuts are tightened to hold it in place.

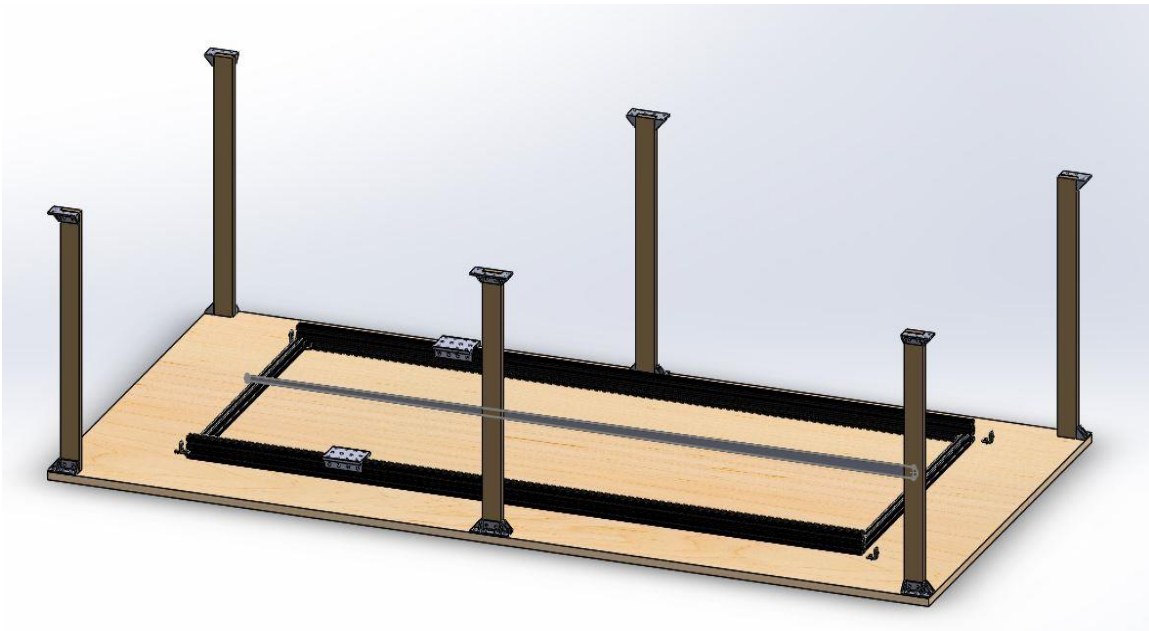


Figure 6.1. Solid Model of the Frame – Wooden Plate, Poles, the Pulley System

### 6.1.2. Carrier Assembly

The solid model of the carrier is shown in Figure 6.2. The carrier is a more complex element. First, an off-the-shelf section of rectangular extrusion is drilled with screw

holes. This includes the holes for the vertical linear guide and linear bearings, and square holes for mounting Z axis motors.

The cup lifting mechanism, consisting of vertical linear guides and actuators, is installed upon the carrier. The sub-assembly includes bronze linear guide bushings, round steel rod beams, and aluminum joining blocks. The drilling, reaming, and tapping of these parts was done in-house in the UH College of Engineering machine shop using the 3D solid models. The bottom of 1/2" round aluminum bar was threaded with a 1/4"-20 thread profile to be mounted vertically and used as the vertical guide rails. The rectangular tube facilitated assembly since most components could be mounted via holes drilled through the tubing wall. Bronze bearings are press fit with a tolerance of 0.001 inch into the aluminum blocks and slide fit on the vertical linear guides. Three square extrusion supports to which the water tubes are attached are also mounted on a carrier with L brackets.

Cup holders on the top of the load cells are designed to be 3D printed. Detailed design is in chapter 7, Figure 7.3 – 3D printed cup holder improvement. When the experiment is integrated and requires the watering system, the water tube fixers are designed and 3D printed.

When all machining is completed, the square tube is assembled with linear bearings and inserted to the 80/20 frame, and the belt is also attached and secured to the carrier.

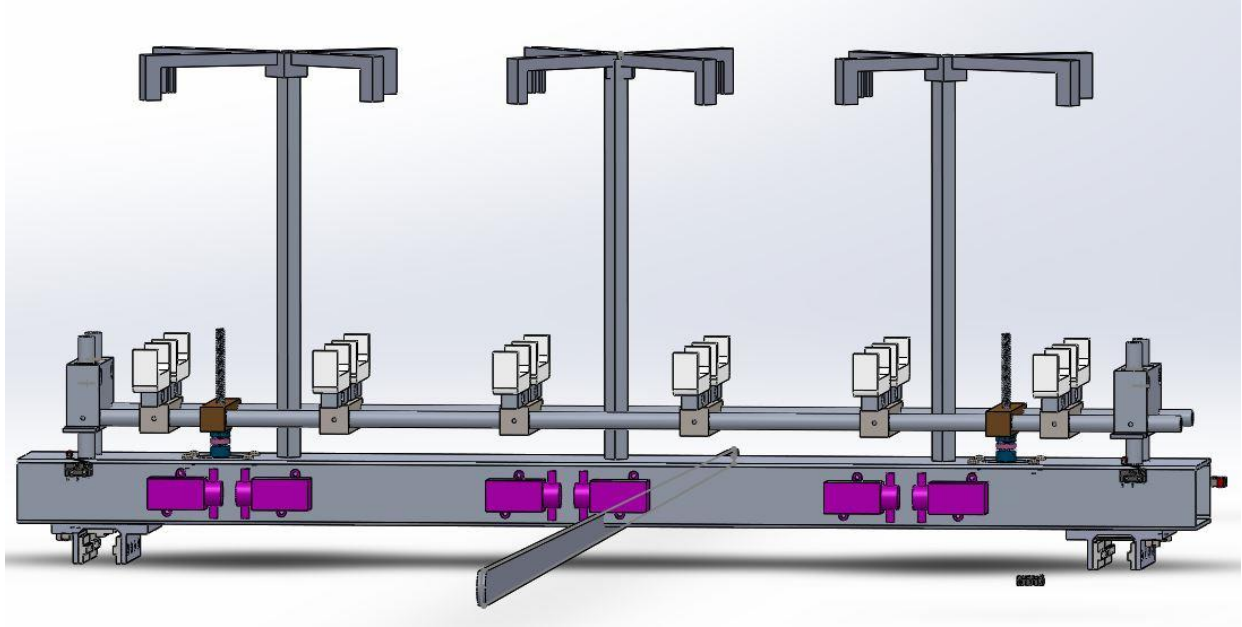


Figure 6.2. Solid Model of the Carrier – Square Tube, Lifting Mechanism (Vertical Linear Guides and Actuators), Cup Holders, Water Tube Fixers.

### 6.1.3. Prototype Solid Model

When frame and carrier assemblies are made, the final step is to integrate with cups and cup tables made by aluminum angle extrusions. Until the final experiment, the styrofoam roof is left removable and water reservoir is set on the top of the roof. Figure 6.3 shows a solid model of prototype soil humidity machine. 3D printed water tube fixers are attached after making sure three of water square tube extrusions are located properly, which means that nothing is touching cups. Importantly, cup table must be parallel to the 1" x 2" 80/20 extrusions.

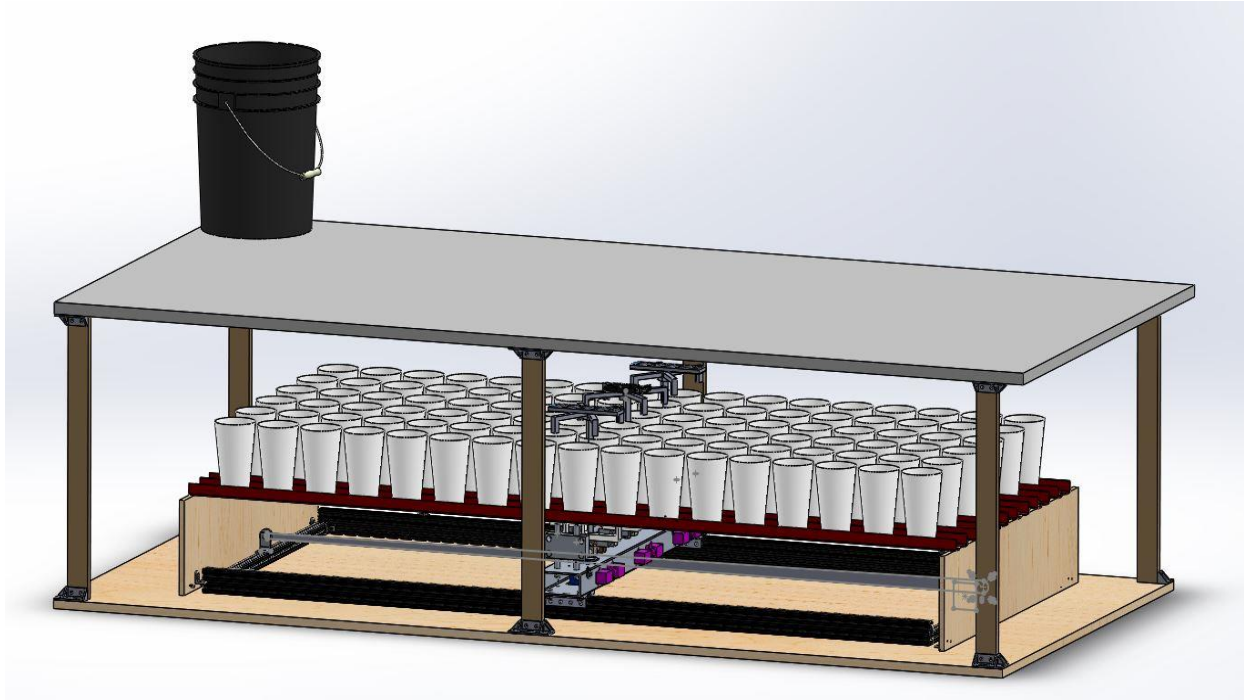


Figure 6.3. Solid Model of Prototype Soil Humidity Machine

## 6.2. Prototype Construction and Assembly

As demonstrated, 80/20 extrusions are located on the wooden plate and aligned so that the rails are perfectly parallel. The square tube and the pulley mount are machined and inserted to the frame before mounting the cup lifting assembly. Software for the Y-axis positioning system is created and tested. When the belt is tightened with sufficient tension, the motion is smooth.

Once all of the parts are machined, Z axis positioning system and the cup lifting system are assembled and examined. All electronics are also assembled and located on the carrier for testing. After individual tests and analysis are completed, the watering system needs to be tested. At this time, the watering square tube extrusions and the tube fixers as well as the cup table are built and assembled on a machine.

Changes from an original solid modeling are

- Outside Frame: Chamber → Wooden supports with a styrofoam roof
- Cup holder design: 90 deg. angle, center mounting hole, a nut as a spacer → 105 deg. angle, edge mounting hole, and 3D printed as a spacer (Figure 7.3)
- Electronics location: On the carrier → Top of the water tube fixers.
- Aluminum angle extrusions distance: A distance between angle extrusions where cups are sitting on bring closer to prevent cups tipping over

Explanations upon these changes are described in the next chapter. 3D printed parts are listed below:

- Cup holders
- Water system tube fixers
- Motor mount
- Frame supports

As mentioned, the electronics are now moved above the water tube fixers. The Arduino is mounted in the middle of the water tube fixer and each side fixer carries a breadboard. Thus the sensors are split in the middle, and half of the sensor sets are connected to one breadboard and the other half utilize the other breadboard. The relays are attached on the bottom of the carrier and wires route along the water square extrusion to the breadboard and the Arduino Mega. With some extension wires from load cells, these are soldered to each ADC which are placed outside of breadboard and along the water tube fixer arms. The MicroSD shield is located by the Arduino, and all wires from stepper motors and drivers are gathered into one wire bundle. The solenoids are installed on the surfaces of the carrier at where they are designed to be in the 3D model. Six solenoids are attached on one face of a carrier and the other 6 are on the other side. Constructed final prototype of a soil humidity machine is shown in Figure 6.4.

The example pictures of plants interacting with the machine is shown in Figure 6.5. The left picture is the cups with plant seeds treated water. The right up picture is a top view while measuring weight. It is easy to see that the water tube fixer secures the water

tubes above the cups. The right bottom picture shows the side view that cups are picked up for the measurements. These pictures are taken during the experiment at day 10.

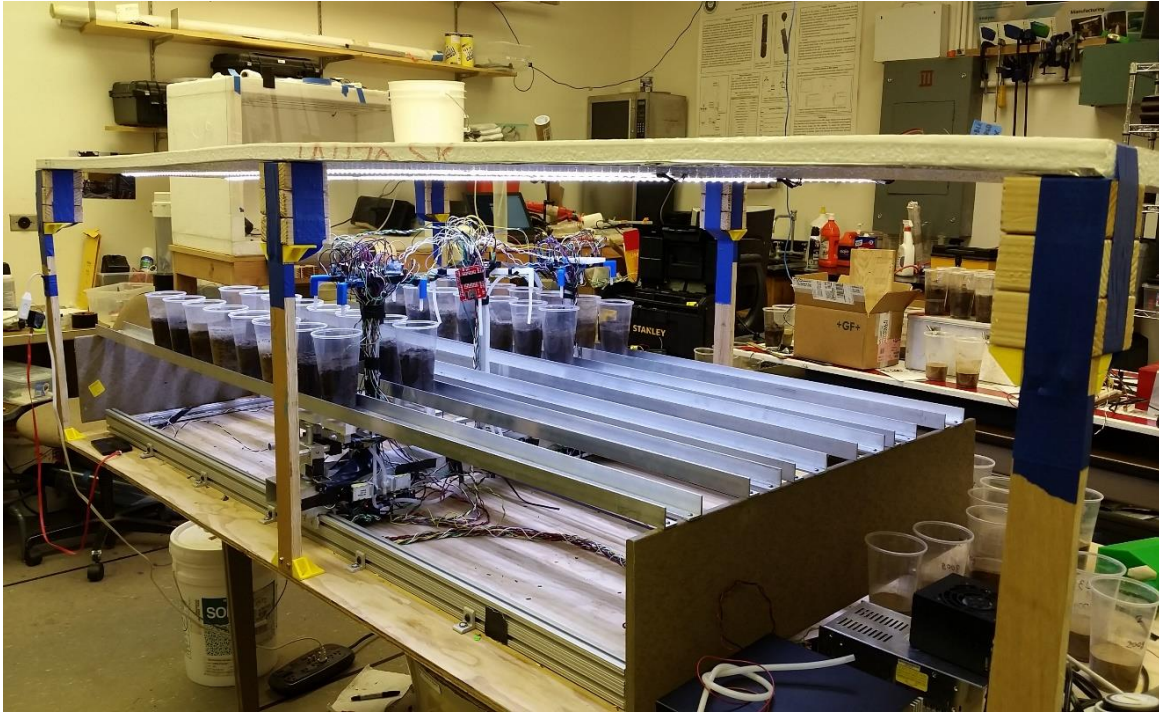


Figure 6.4. Assembled Prototype Soil Humidity Machine

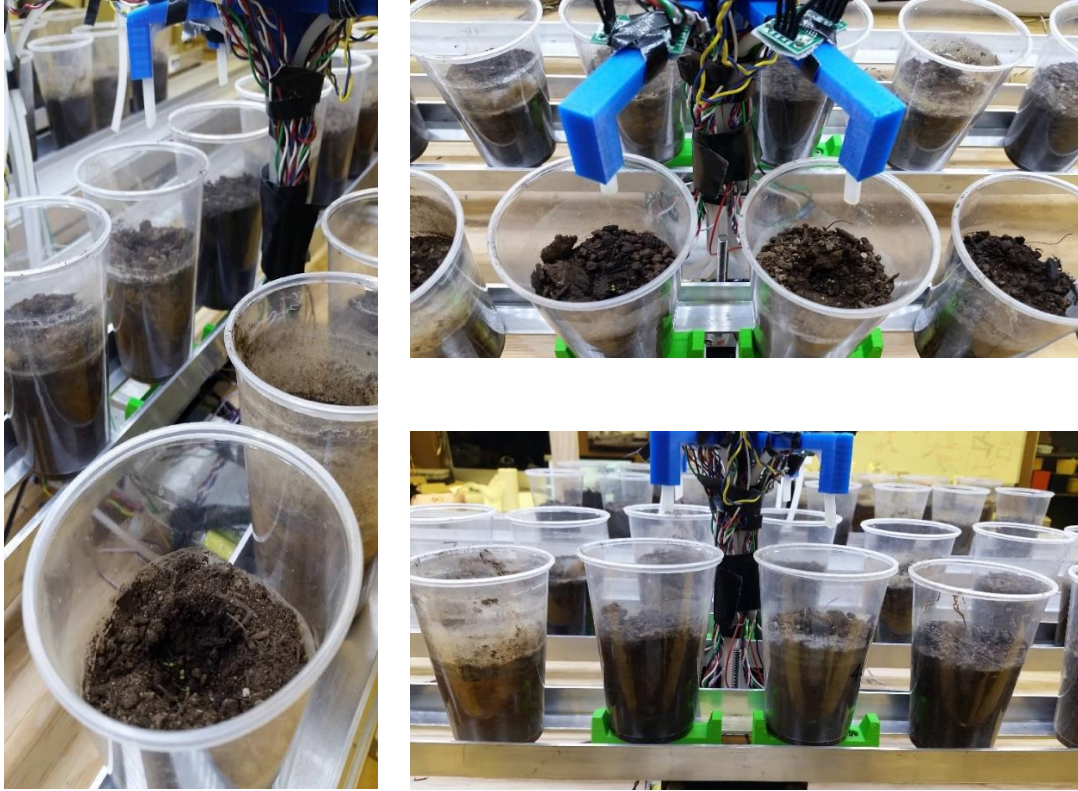


Figure 6.5. Plants Interacting with Machine – Plants Growing Pictures Taken at Day10

### 6.3. Bill of Overall Materials

Overall bill of materials is shown in Table 6.1. The total cost of this project is about \$1,415, whereas existing machines cost over \$100,000. There are some parts that are able to greatly reduce the cost, such as stepper motors, stepper motor drivers, relays, and solenoids. Details for the future work will be discussed in chapter 8 in the future work.

Table 6.1. Purchased List

Positioning System					
Parts	Dimension	Qty	Cost	Total Cost	Store
Stepper Motor (ROB-10846)	/	2	16.95	33.90	Sparkfun
Stepper Motor (23HS45-4204S)	/	1	29.90	29.90	ebay
Motor Driver (ST-M5045)	/	1	36.94	36.94	ebay
Motor Driver (ST6600)	/	2	8.71	17.42	ebay
Spider	3/4" OD	2	8.89	17.78	McMaster
Coupling Hub (1/4")	1/4" Dia. shaft, 1.1" (L)	2	17.54	35.08	McMaster
Coupling Hub (5mm)	5mm Dia. shaft, 1.1" (L)	2	17.54	35.08	McMaster
Lead Screw (1/4")	1/4"-16 Acme, 6"(L)	2	8.98	17.96	McMaster
Pulley	10mm Dia., 2mm pitch, 6.5mm width	2	4.60	9.20	ebay
RepRap GT2 Timing Belt	6mm Wide, 2mm pitch	1	4.89	4.89	ebay
<b>Subtotal:</b>				<b>238.15</b>	
Carrier					
Parts	Dimension	Qty	Cost	Total Cost	Store
Bronze Bar	3/4" (Th), 3/4" (W), 1/2' (L)	1	19.14	19.14	McMaster
Bronze Sleeve Bearing	1/2" Dia., 3/4" OD, 5/8" (L)	6	2.44	14.64	McMaster
Aluminum Rectangular Bar	1/8" x 2-1/2", 1/2' (L)	1	7.80	7.80	McMaster
Aluminum Rod - 3' long	1/2" Dia., 3' (L)	2	8.44	16.88	McMaster
Aluminum Rod - 2' long	1/2" Dia., 2' (L)	1	6.44	6.44	McMaster
Steel Rectangular Bar	7/8" (Th), 7/8" (W), 2' (L)	1	28.39	28.39	McMaster
Aluminum 6063 Square Tube	1/8" (Th), 1-3/4" x 4", 3' (L)	1	31.16	31.16	McMaster
<b>Subtotal:</b>				<b>124.45</b>	
Watering System & Electronics					

<b>Parts</b>	<b>Dimension</b>	<b>Qty</b>	<b>Cost</b>	<b>Total Cost</b>	<b>Store</b>
Solenoid Valve	/	12	19.95	239.40	TrossenRobotics
RobotGeek Relay	/	12	6.95	83.40	TrossenRobotics
Silicone Tubing	3.5mm x 10m (L)	1	19.95	19.95	TrossenRobotics
Arduino Mega	/	1	45.95	45.95	Sparkfun
ADC	/	12	0.99	11.88	ebay
Load Cell	/	12	3.99	47.88	ebay
MicroSD Shield	/	1	14.95	14.95	Sparkfun
<b>Subtotal:</b>				<b>463.41</b>	
<b>Frame</b>					
<b>Parts</b>	<b>Dimension</b>	<b>Qty</b>	<b>Cost</b>	<b>Total Cost</b>	<b>Store</b>
Recycled Plastic Lumber	1/2" (Th), 8" (W), 6' (L)	2	27.45	54.90	McMaster
Multipurpose Aluminum 90 Deg. Angle	1/8" (Th), 1" x 1" (Legs), 8' (L)	12	10.52	126.24	McMaster
80/20 - 1010 Extrusion	1" x 1", 30"(L)	2	8.85	17.70	ebay
80/20 - 1020 Extrusion	1" x 2", 72" (L)	2	30.03	60.06	ebay
Linear Bearing (#6726)	1.218"(H) x 2.937"(W) x 4"(L)	2	33.74	67.49	ebay
Styrofoam	4' x 8', 1"(W)	1	19.98	19.98	Lows
Base Wood	4' x 8', 1/2"(W)	1	39.95	39.95	Home Depot
<b>Subtotal:</b>				<b>386.32</b>	
<b>Miscellaneous</b>					
<b>Subtotal:</b>				<b>200.00</b>	
<b>Total :</b>				<b>1412.33</b>	<b>USD</b>

## CHAPTER 7

### RELIABILITY TESTING AND ANALYSIS

To achieve the reliability goal of the machine, testing data of the positioning system and the cup lifting system is collected and analyzed. Section 7.1 is the description of the seed selection. Section 7.2 is about the analysis of these systems and descriptions about design changes based on the results. Overall machine reliability is also evaluated in this section.

Section 7.3 is repeatability testing and analysis. Data is taken from stabilized load cell measurements, which involves no motion. Then mass measurement data is taken from tests with vertical motions alone and both horizontal and vertical motions, as well as complete cycles without the watering system. When three experiments succeed the repeatability tests, the watering system is incorporated to the cycle and the collected measurements. Each process with design changes and considerations are also included in this section. Through repeated experiments, the ADC and the load cells are found as sensitive sensors. Finally, the reading errors are discussed in the last section of chapter 7.

#### 7.1. Seed Selection

In order to study habitats of plants or crops, and understand precise evaluation of plants in adaptation to particular environments, seed selection is essential. A basic knowledge of how the environment affects to the plants growth or how plants evolve differently would be beneficial to this study.

One of the well-known plants used for scientific study is *Arabidopsis thaliana*. *Arabidopsis thaliana*, which has become a model plant for the study of natural variation (Mitchell-Olds & Schmitt, 2006), is one of the most popular species, which has many accessions and resources in functional genomics (Song & Mitchell-Olds, 2011). *Arabidopsis thaliana* has a fast life/growth cycle, which requires only air, water,

and light to complete its life cycle. Although Arabidopsis is not an economically important plant, they produce numerous self-progeny and grow easily in a greenhouse or indoor growth chamber. Selecting Arabidopsis is reasonable because it has several desirable traits and it has been the focus of intense genetic, biochemical, and physiological study. The existence of extensive scientific documentation on this species makes it suitable for controlled laboratory study (Foundation, 2013). Since Arabidopsis thaliana has diverse ecological conditions (Mitchell-Olds & Schmitt, 2006) (Bergelson, et al., 1998) (Le Corre, 2005) (Bomblies, et al., 2010) (Platt, et al., 2010), it is also a convenient species to lead to adaptive variation for many traits such as fitness traits that includes number and size of offspring, age at first reproduction, and reproductive lifespan and aging and many morphology (Koornneef, et al., 2004).

## 7.2. Analysis

In the section 7.2, analysis of the positioning system, the cup picking up system, and machine reliability are discussed. Also, based on failures that will be introduced in 7.2 subsections, a new design of components are created.

### 7.2.1. Positioning System

Before gathering reasonable data, there is a problem in the positioning system that required design changes. In the first design of the machine, limit switches were not introduced. Although a powerful motor is selected and machine's motion is slow, stepper motors in both axis lose steps. Because of step lost, the machine doesn't stop underneath the station if it's on the Y-axis and two Z-axis motors are tilt if one motor lose steps or the lifting mechanism doesn't reach cups. As a result, cups aren't picked up properly and thus cups are tipped over as seen in Figure 7.1. One tipped over cup causes the cups next to

it to fail like bowling pins. Worst scenario is when the Z-axis stepper motors lose steps and the machine doesn't clear the height from the cups. Or in another word, the picking up mechanism doesn't reach the Z-axis origin. This causes the machine to drag an entire row of cups to the next station. This problem requires a fix immediately to increase the machine reliability and repeatability.

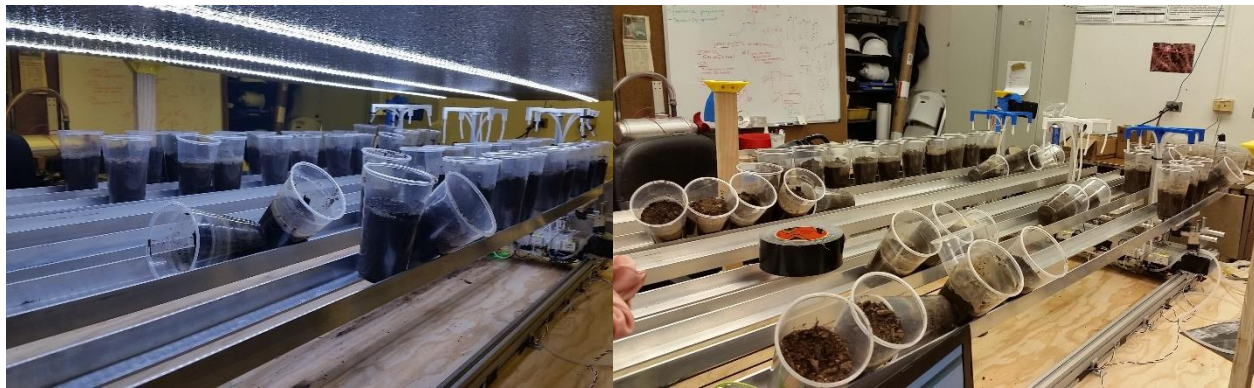


Figure 7.1. Cup tipped over

First solution is to have a documented occurrence of the machine losing steps. The setup of this experiment is without cups sitting on the cup table and the watering system. The machine runs a complete cycle, moves up and down, to the next station, and comes back to the Y-axis origin. A measuring tool, dial indicator, is fixed by the origin at about a height of the carrier. Every time when the machine returns to the origin, the dial indicator reads a measurement and the value is recorded. Up to 25 cycles are continuously tested in each experiment. Collected data is shown in Table 7.1, test 1. The number of steps between stations is 1,249 steps. The number of steps lost is calculated from "lost" divided by 0.1249, where the step lost is calculated from computed run 1 minus computed run 25.

Thus this experiment shows that the stepper motor loses about four steps every 25 cycles. Moreover, Figure 7.2 is a plot of test 1 in dots and its fitted linear regression model is also indicated in the plot. With this proof of the steps lost being significant and frequent, limit switches are introduced and installed on the machine.

Table 7.1. Experiment of Y-axis Stepper Motor Steps Lost

<b>Cycle</b>	<b>Test 1</b>	<b>Computed</b>	<b>Difference</b>
1	3.254	3.681	-0.427
2	3.594	3.702	-0.108
3	3.288	3.723	-0.435
4	3.688	3.743	-0.055
5	4.162	3.764	0.398
6	3.882	3.784	0.098
7	3.814	3.805	0.009
8	4.112	3.826	0.286
9	4.096	3.846	0.250
10	4.002	3.867	0.135
11	4.022	3.887	0.135
12	3.758	3.908	-0.150
13	4.072	3.929	0.143
14	4.446	3.949	0.497
15	4.026	3.970	0.056
16	4.092	3.990	0.102
17	3.754	4.011	-0.257
18	3.796	4.032	-0.236
19	4.020	4.052	-0.032
20	3.832	4.073	-0.241
21	3.936	4.093	-0.157
22	3.956	4.114	-0.158
23	4.164	4.135	0.029
24	4.222	4.155	0.067
25	4.218	4.176	0.042
		<b>Average</b>	0.000
		<b>StdDev</b>	0.231
<b>Lost</b>	-0.494		
<b>Steps lost</b>	<b>-3.958</b>		

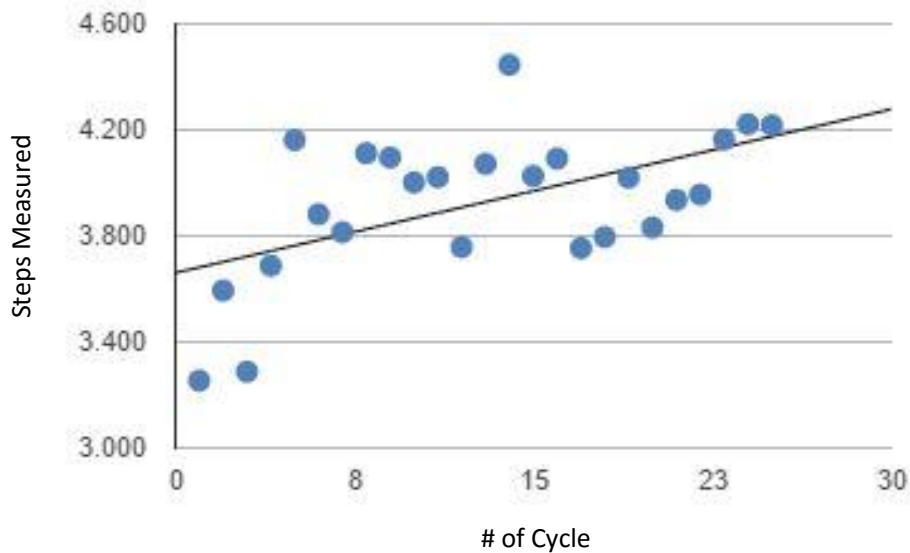


Figure 7.2. Plots of Test 1 Step Readings

### 7.2.2. Cup Picking up System

The second change is the design of the 3D printed cup holder. The initial role of these holders is to hold the cups into a secure position as they are being lifted. In addition to the step loss of motors, cups migrate little by little because of a vibration created by the positioning system. Major changes in the new design involve changing the angle of the cup holder edges. Previously it is designed with 90 degrees; however, when a cup shifts and the edge is right underneath picking up a cup, it causes the cup to tip over. To avoid this situation, another 15 degrees are added to the edges (see Figure 7.3). By doing this, it provides a cup an ability to slide down to a flat surface. A cup holder also gets shorter about 0.1 inch to not giving a cup a chance of migrating while it is picked up.

Another change on this design is the location of a screw hole. In an experiment, details are shown in section 7.3, it can be seen that the location of cup sitting on a load cell while measuring doesn't matter. Thus, bringing two cup holders closer, for example sensor 1

and sensor 7, sensor 3 and sensor 9, and sensor 6 and sensor 12 and so on, the total number of cups is increased. Therefore, the screw holes are now designed by the edge instead of in the center of the cup holders. In addition, an extruded step is created on the bottom of the cup holder. Before modification, the cup holder is designed with a flat surface. A nut as a spacer is utilized in between the load cell and the cup holder so that the strain can be correctly measured. By introducing a new design, the cost of nuts is eliminated and becomes simpler for installation. Previous design and modified design are shown in Figure 7.3.

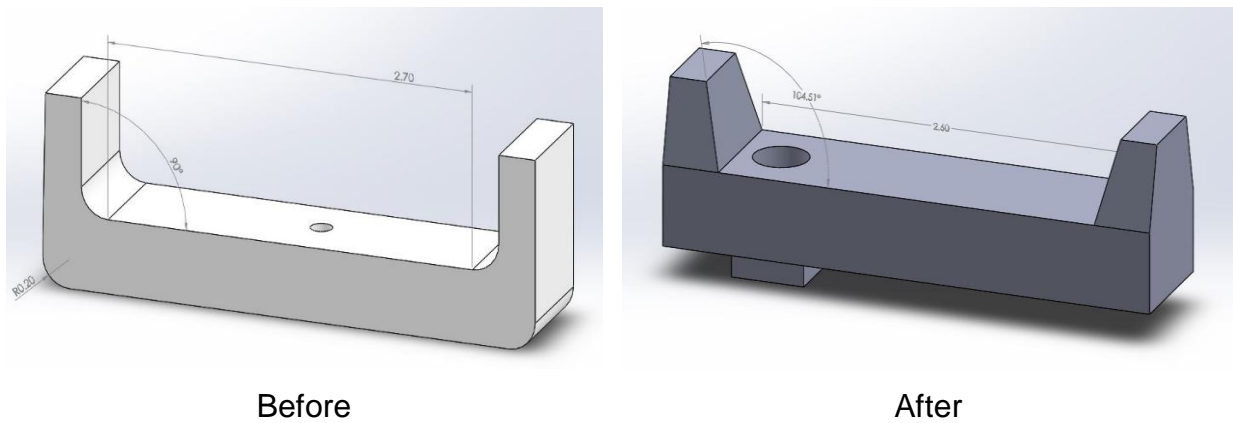


Figure 7.3. 3D Printed Cup Holder Improvement

Furthermore, in the case that the cups tip over, shown in Figure 7.1, there are two directions for cups to fail; one is tipped along an aluminum angle extrusion (parallel failure), and the other one is perpendicular to the extrusion. By installing limit switches on both axis and adjusting the design of cup holder, parallel failure is prevented.

A previous installation of a set of angle extrusions is 1.25" with approximately +/- 0.05" spaced from each other. The bottom cup diameter is 2.25" and the top cup diameter is 3.50". Since the angle extrusion dimension is 1" x 1", the cup has a wide margin, 3.25", to sit, and this means that when a cup is at all the way to the extrusion 90 degree corner,

the other edge of a cup slides into the space between, causing the cup to tip over perpendicularly. Especially in a middle of extrusions, the distance between angles tends to become gradually a wider distance like 1.45" since only both ends are fixed.

In order to alleviate this problem, the spacing between the extrusions is reduced closer to 1". This means that even when a cup is sitting at a corner of extrusion, there is 0.25" space where the other edge of the cup can sit. The extrusion distance change, of course, affect the design of the cup holder. It now becomes 0.8" wide. The diameter of the cup at the top of the extrusion must also be accounted for. If the extrusion spacing is too close at any point, the cup's vertical movement may be impeded and cause problems in the lifting sequence. In this case, the total distance where a cup can sit on is now 3" and the cup is free to move, allowing the measurements to be conducted without interruption or malfunction.

### 7.2.3. Machine Reliability

With some upgrades, the machine operates properly without the cup failures for testing. However, after two weeks of experimentation and numerous tests since the machine started moving, another issue is discovered. The pulley system starts making a squeaking noise. The plastic washer installed in between the shoulder screw and the pulley mount is broken. Although lubrication reduces the noise, clearly the washer needs replacement, and in order to increase the part's life, a design modification is required.

Two weeks' worth of data is imported to MATLAB and the number of cycles is counted. The first week from March 1<sup>st</sup> to 7<sup>th</sup>, about 7.1 days, is 1582 cycles in total and the following week from March 7<sup>th</sup> to March 14<sup>th</sup>, about 7.4 days, has 1671 cycles. Thus in two weeks, 3253 cycles are examined, that is about 232.36 cycles a day. The whole testing period is counted exactly two weeks since Y axis started moving, which conducts another 3253 cycles. An entire machine operation period is four weeks, or 6506 cycles, until the washer is broken. Although the target requirement is one cycle an hour, this machine has been running 232.36 cycles which is more than 24 cycles a day. If the

machine was completing 24 cycles a day as required, 6506 cycles would take about 271.08 days to complete, or approximately 9.04 months assuming 30 days in a month. This experiment shows that the machine lasts three times longer than the required three months experiment period if 8 sensors are utilized. Moreover, the plastic washer is the only machine failure so far, and thus it concludes that other mechanisms are reliable.

### 7.3. Repeatability Check Statistical Method

In this section, sensor repeatability is discussed. The weighing system is analyzed first, then the watering system is added and analyzed. As briefly introduced in the weighing system section, experiments for measuring weight on different locations on the cup holder are discussed. Also three experiments are explained here: measurements without any motions, measurements with a vertical movement, and then a whole cycle measurement without the watering system. Each set of results show standard deviations of measurements of 400 cycles. First two experiments are only utilizing 8 set of sensors at 1 station since no horizontal movement gets involved yet. Then the last experiment is with 5 stations and 8 set of sensors included. Only 8 sensors are utilized for this manipulation experiment due to the number of failed sensors.

The following subsection 7.3.2 includes an analysis of the watering system involved. A software upgrade and the target masses creation are introduced in the section.

Lastly, an analysis of measurements based on these experiments is discussed. This section has a discussion of average cycle time, load cell responses with data showing measurements while watering, and errors.

### 7.3.1. Weighing System – Load Cell

First introduced here is an experiment shows an evidence that the location of the load on the load cell doesn't effect on the mass measurements. Measurements are taken when the load is on outside, midpoint, and inside of the load cell. According to the concept of deflection, the measurements are subjected to the distance where a force is applied. The results are shown in Table 7.2 and Figure 7.4. The measurements are either averaged or not averaged and 12 sensors are utilized. Each ten mass measurements is averaged utilizing the function in the HX711.h library. Non-averaged measurement is a pure measurement directly coming from the first reading. Interestingly, in both cases, Table 7.2 shows that at any location, the mass is measured at a constant value, assuming an error of +/- 0.05 g. A corresponding plot is shown in Figure 7.4. As seen in both Table 7.2 and Figure 7.4, sensor #4 has dramatically changed. However this change is not corresponding to the distance. The averaged mass of the #4 sensor increases by the distance. On the other hand, the non-averaged values shows that the measurements are random and the measurement taken at 0 mm is not the smallest value. This concludes that the sensor #4 is not working properly and thus it is ignored from analysis. All the other sensors prove that the weighing location doesn't effect to the mass measurements. It is interesting to consider that this type of load cell acts like a Roberval balance that equals the mass to the calibrated mass regardless of where on the plates items are placed.

Table 7.2. Mass measurement on 3 different locations on a load cell.

Sensor #	Averaged Mass				Sensor #	Non Averaged Mass		
	0mm (Outside)	10mm (Midpoint)	20mm (Inside)			0mm (Outside)	10mm (Midpoint)	20mm (Inside)
1	257.0863	262.2333	262.3594		1	262.3140	262.2730	262.2457
2	173.7823	173.0260	173.2245		2	173.1987	173.2067	173.1927
3	205.4010	205.5133	206.0697		3	205.6750	205.6540	205.5987
4	128.7953	145.5563	182.1523		4	150.3480	154.6767	139.5537
5	152.8980	155.4000	155.4171		5	156.9290	156.9263	156.8123
6	200.8760	201.2440	201.1519		6	201.1353	201.1100	201.1523
7	163.9833	163.7557	163.7768		7	163.5283	163.5943	163.4603
8	172.7767	172.5390	172.5939		8	172.5923	172.5227	172.5183
9	150.5077	150.3787	150.3135		9	150.3760	150.3633	150.3767
10	170.3980	170.2200	171.2684		10	170.9377	170.9560	170.9133
11	137.5963	137.9317	137.8981		11	137.9193	137.9290	137.8780
12	171.4700	171.4483	171.5416		12	171.4183	171.4817	171.4400

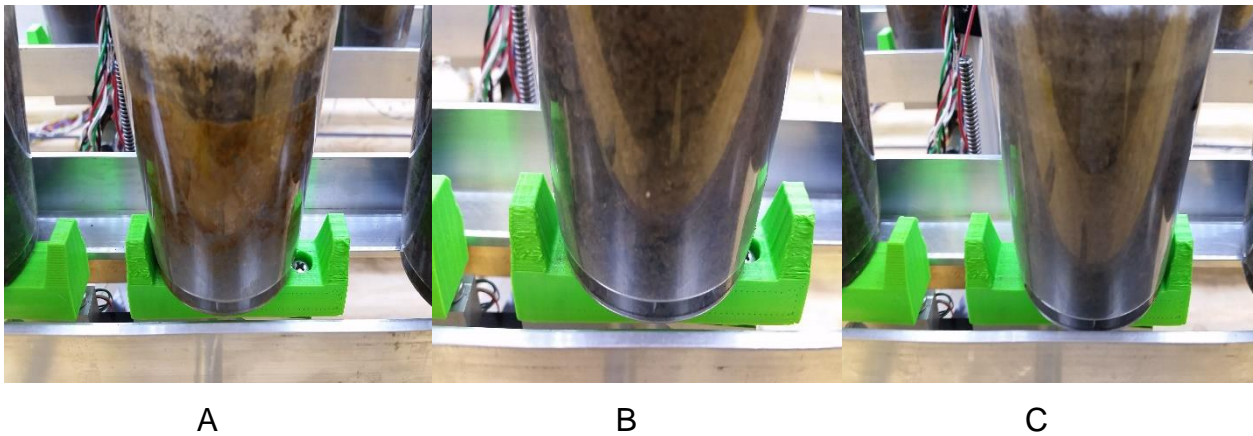
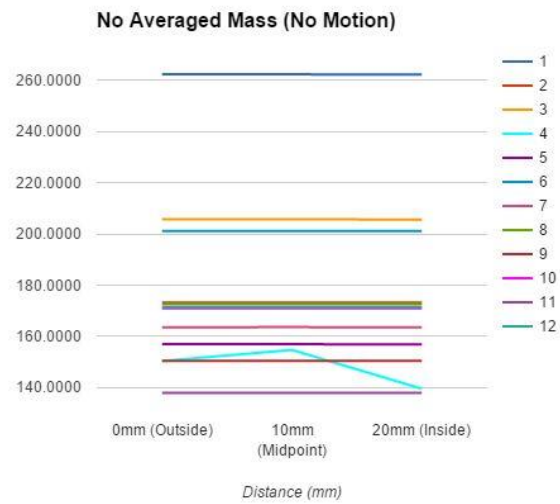
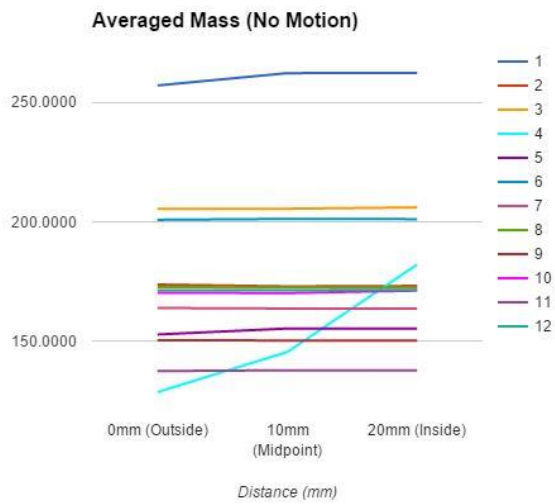


Figure 7.4. Plot on the top corresponds to the data in Table 7.2. Bottom pictures shows that the mass measurement on a 3 different locations on a load cell; outside (C), midpoint (B), and inside (A)

Knowing that the exact location of the load doesn't introduce problems to repeatability testing, 400 cycles of stabilized data is taken. These measurements start when the machine picks up the cups. A simplified programming code is created for this experiment. Thus the machine comes to the Y and Z origins, tares all sensors, and then picks up cups. The machine stays at the top and keeps weighing cups for 400 cycles.

At this time, only 8 sets of sensors are manipulated instead of 12 due to broken sensors. After multiple testing, either the ADCs or the load cells, at least four sets of

sensors are needed to be replaced. To equalize the location of the cups, four sensors in the middle of the machine (sensor # 3, 4, 9, and 10) are avoided to use for this experiment. Thus four broken sensors are removed from the machine and the other sensor numbers are indicated as sensor #1 through #8. All 400 measurements are imported to MATLAB and standard deviations of each sensors are calculated, shown in Figure 7.5. Because there is no motions involved and causing less noise, all sensors' standard deviations are computed less than 0.15 grams. Although its standard deviations are low, the sensor #8 is 0.1 g higher than other sensors, shown in Figure 7.5. This higher standard deviation could be gained from the programming or the noise from the connection of the sensor.

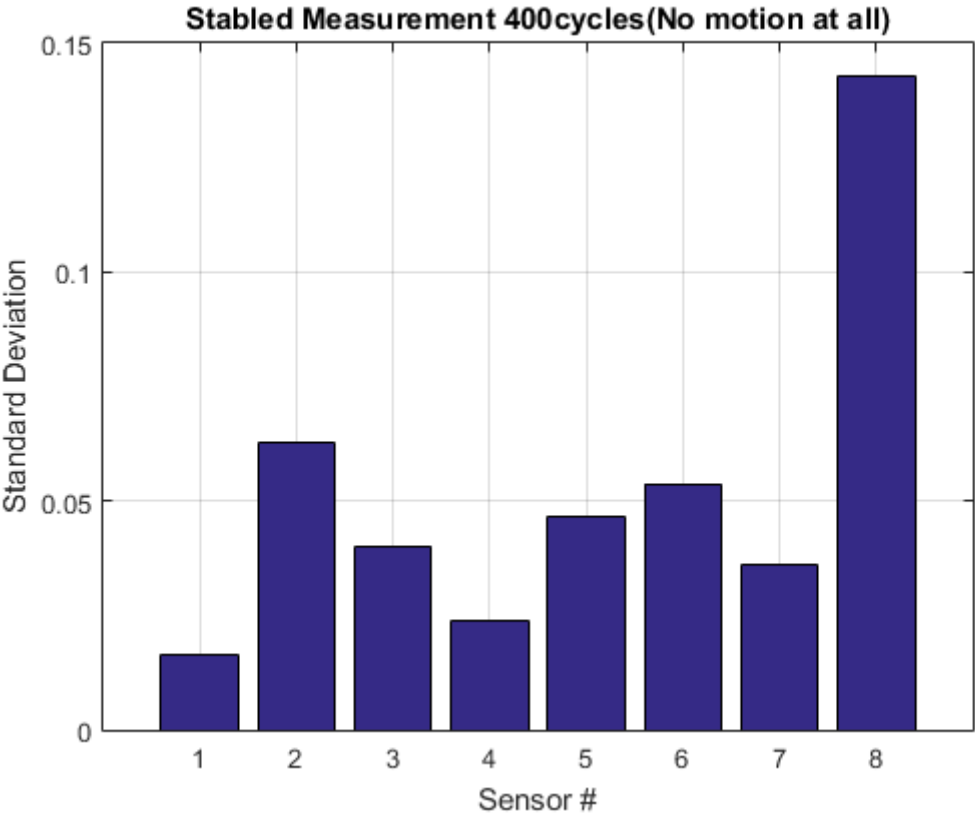


Figure 7.5. Standard Deviation of load sensor measurements when machine is not moving.

The plot below, Figure 7-6, shows standard deviations from vertical motions at station 1. During this experiment, similar to the previous method, the machine comes to the home cycle then position under station 1. After a tare process, the machine picks up cups and records its weight measurements. When all sensor measurements are recorded, the machine moves down and waits for a tare process again, then moves up and measures. There is no other stations are involved in this experiment. The results are shown in Figure 7-6. In spite of Z-axis movements, standard deviation of each sensors are below 0.1 g, including sensor #8. This can be seen by comparing Figure 7-5 and Figure 7-6, not only sensor #8 but also sensor #2 and #6 exhibit satisfactory standard deviations and measure below 0.05 g, whereas in Figure 7.5, the sensor #2 and #6 have standard deviation slightly larger than 0.05 g. It could be the tare system that improves these sensor. In any case, the plot indicates that all sensors are repeatable with a vertical motion with the tare system.

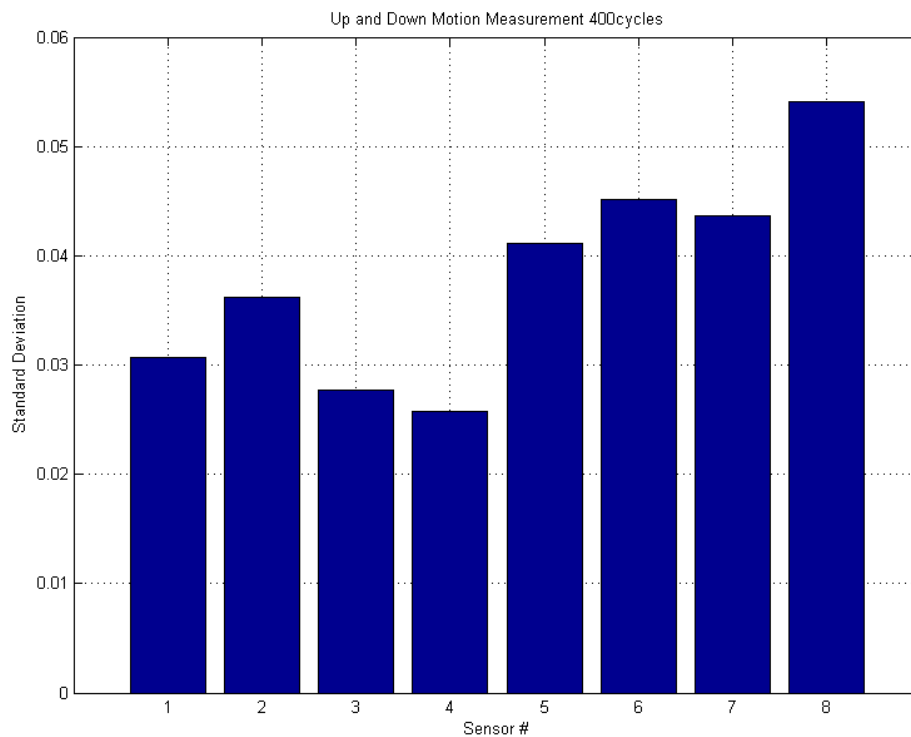


Figure 7.6. Standard Deviation of load sensor measurements when machine moves Z axis only.

Lastly, the Y-axis motion is applied. Watering system is disabled and the machine runs a complete 406 cycles. Unfortunately, as seen in Figure 7-7 top plot, sensor #2 and #6 had issues and those standard deviations get over 35 g. Based on the raw data, it clearly shows that #2 and #6 ADC sensors need to be checked and replaced if necessary. Raw data shows that these measurements are all over the place, including reading 0 g sometimes. Details of explanations about a sensor error is discussed in the next section.

Zoomed in data without sensor #2 and #6 is the bottom plot. On the plot (Figure 7.7), the standard deviation is on Y-axis and station number is on X-axis. The colors indicate sensor numbers. Only sensor #8 remained below 1 g and everything else gets exceeded 1 g at some point. To improve the results and to avoid water overflow when the machine is integrated with the watering system, some parts of software are improved to the final shape, described in chapter 5. Bad sensors (#2 and #6) are replaced, and also all sensors and wires are soldered and secured to prepare for the final experiment.

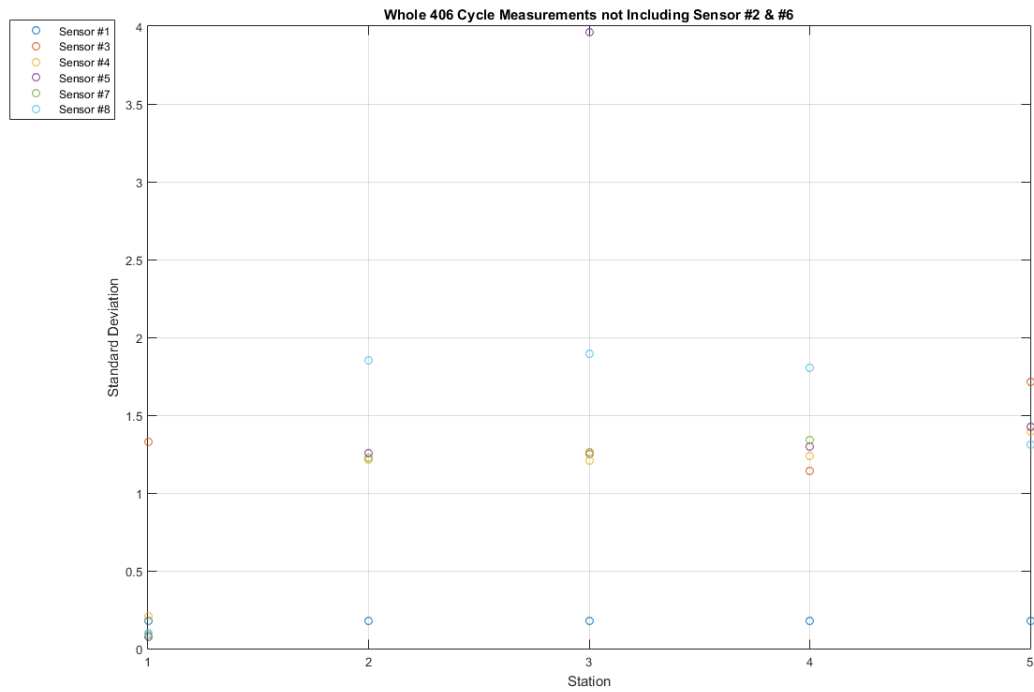
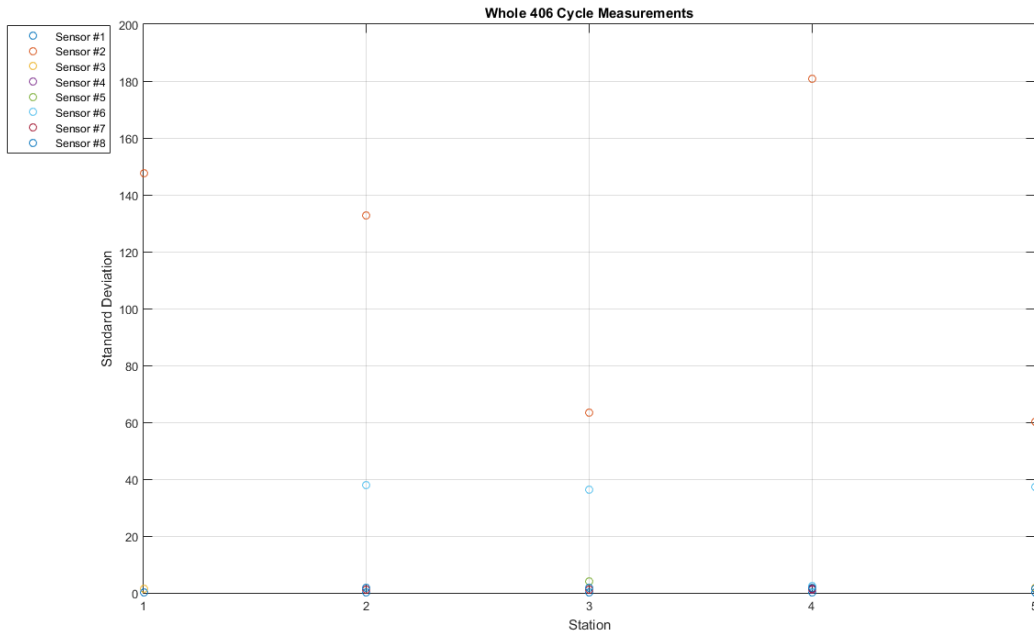


Figure 7.7. Top plot shows that there are 2 bad sensors (#2 and #6), and bottom plot is zoomed in version after #2 and #6 data is removed. Standard Deviation of the load sensor with both Y and Z axis motions without watering system. Both data is taken before software is upgraded and #2 and #6 ADCs are replaced.

### 7.3.2. Watering System – Weight Related

In addition to the weighing system, the watering system is applied after software is upgraded and bad sensors are replaced. New program double checks a cup weight when the first measurement failed. Moreover, by using a small number of margin (7 g), it avoids water overflow and to extinguish seeds by sensor failures.

Figure 7-8 is the plot from the data that includes all the systems and mechanisms, shown in Figure 5-7. Figure 7-8 shows the mass measured and the target masses. Original mass of each cups is measured and added to a random set of targeting masses, consisting of 0 g, 20 g, 40 g, and 60 g. The summation is now considered as the target masses and inserted into an Arduino code. The machine activates the watering system based on the target mass and measured weight. Red circles are values of the random set of targeting mass. Measured masses (blue stars) are computed in MATLAB by subtracting the original mass from the target mass to find an amount of water added. The yellow line indicates the difference errors. Y-axis of the plot is a mass in grams and X-axis is the sensor numbers. To read sensor numbers, since 8 load cells are manipulated, first sensor numbers #1 through #8 are at station #1, sensor numbers #9 through #16 means that sensor #1 through #8 respectively at station #2, and so on up to station #5 with sensor #33 through #40. The way to read X-axis of the plot below is summarized in the Table 7.3. This data is collected overnight, about 18 hours of measurements. All data in each stations and each sensors are averaged and plotted.

As seen in Figure 7-8, the measurements with watering system are reasonably accurate with the largest error of about 2.2 g. Similarly, standard deviations are also calculated and plotted utilizing MATLAB, shown in Figure 7-9. The overall repeatability of the machine achieved is less than 0.815 g.

Table 7.3. Explanation of Reading X-axis of the Figure 7.8.

Plot	Sensor	Station	Plot	Sensor	Station	Plot	Sensor	Station
1	1	1	9	1	2	17	1	3
2	2		10	2		18	2	
3	3		11	3		19	3	
4	4		12	4		20	4	
5	5		13	5		21	5	
6	6		14	6		22	6	
7	7		15	7		23	7	
8	8		16	8		24	8	
Plot	Sensor	Station	Plot	Sensor	Station			
25	1	4	33	1	5			
26	2		34	2				
27	3		35	3				
28	4		36	4				
29	5		37	5				
30	6		38	6				
31	7		39	7				
32	8		40	8				

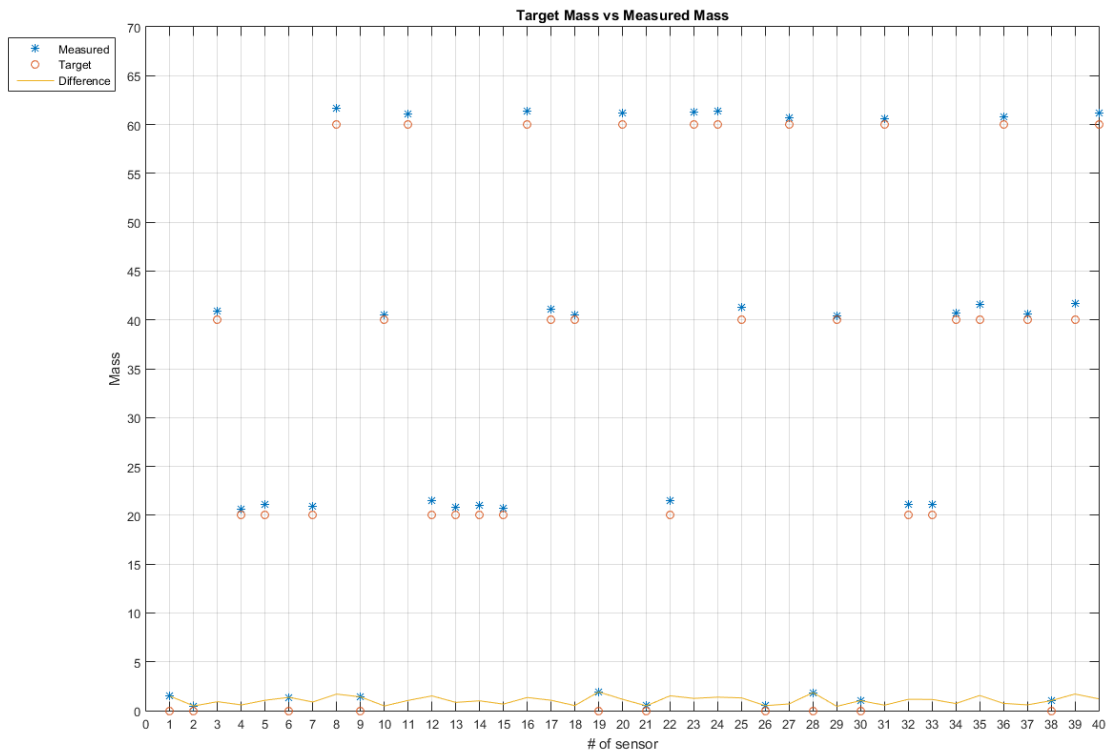


Figure 7.8. After Watered Measurements and Target Mass from Complete Cycle. X axis indicates Station 1 is Sensor 1 through 8, Station 2 is Sensor 9 through 16, and so on.

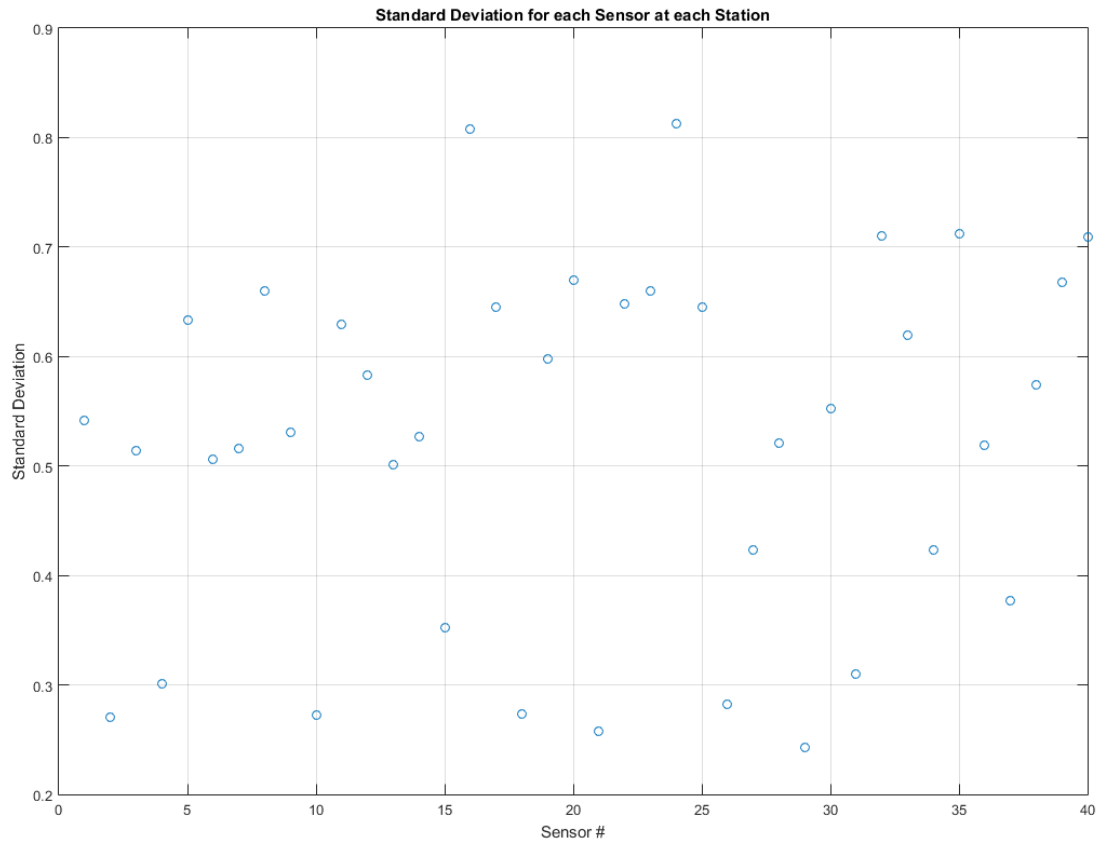


Figure 7.9. Standard Deviation from the Complete Cycle. All sensors at any station achieved standard deviation of less than 1 g. X axis indicates Sensor 1 through 8 are in Station 1, Sensor 9 through 16 are in Station 2 sensor 1 through 8, and so on.

### 7.3.3. Measurements

Given accurate and repeatable data as shown above, an average cycle time of 4.14 minutes is computed by utilizing MATLAB. With a basic calculation, each cup is treated for an average of about 6.21 seconds, and if 100 cups are placed, a total of 10.35 minutes is required to complete a cycle. Even if the equation is not linear, a cycle is only required to be complete once every hour, leaving a total of 49.65 min extra to complete each cycle.

Practical precision requirements are also considered. The original requirement is 0.1 g accuracy; however, the experimented machine only provides 1 g repeatability. The fact is based on the repeatability of sensors. First off, the selected load cell has a 0.03%F.S that is +/- 0.3 g accuracy by itself. In addition, from the experiment, solenoids also result in about +1 g errors every time they water. This is also seen in Figure 7.10. The second peak and the third peak have about 0.5 g errors. Furthermore, an additional error could be created by flawed wire connections producing noise. These facts increase the total error of the repeatability to approximately 1.5 g. From the result, assuming about three times more of 1.5 g as an acceptable precision, the system should be capable of about +/- 4.5 g of repeatability, whereas the designed soil humidity machine records 1 g.

The following plot, Figure 7-10, is a zoomed in version of the change of mass of sensor #1 at station #1 while watering. It shows that the mass slowly decreases through a cycle. When it decreases to below the target mass, watering system turns on and the mass increases backup to the target mass +/- 0.7 g. Despite some large errors, the wave shape of the measurements are constant. The upgraded double checking method is working correctly. All slopes in the Figure 7.10 are parallel each other and taking the third slope as an example, by assuming  $x_1 = 98$ ,  $x_2 = 180$ ,  $y_1 = 357.625$ , and  $y_2 = 356.4$ , the slope is computed as  $m = -\frac{49}{3280}$ .

First step to find a predicted cycles is using the slope  $m$ , finding a new linear equation with an initial mass at initial cycle. For example, to anticipate the number of cycles of the second peak in Figure 7.10, an initial mass is about 357.125 g at cycle # 45. This creates an equation,  $y = -\frac{49}{3280}x + 357.80$ . The minimum mass before the watering system turns on is about  $y = 356.4$  g according to the plot. Thus, it allows prediction of the number of cycles it takes for a cup to dry out, which is about at 94 cycles. It has some errors and readings might not be accurate in an example, it is easily estimated. Furthermore, knowing an average cycle period, that was 4.14 min in the previous experiment, it is possible to predict its estimated time period as well.

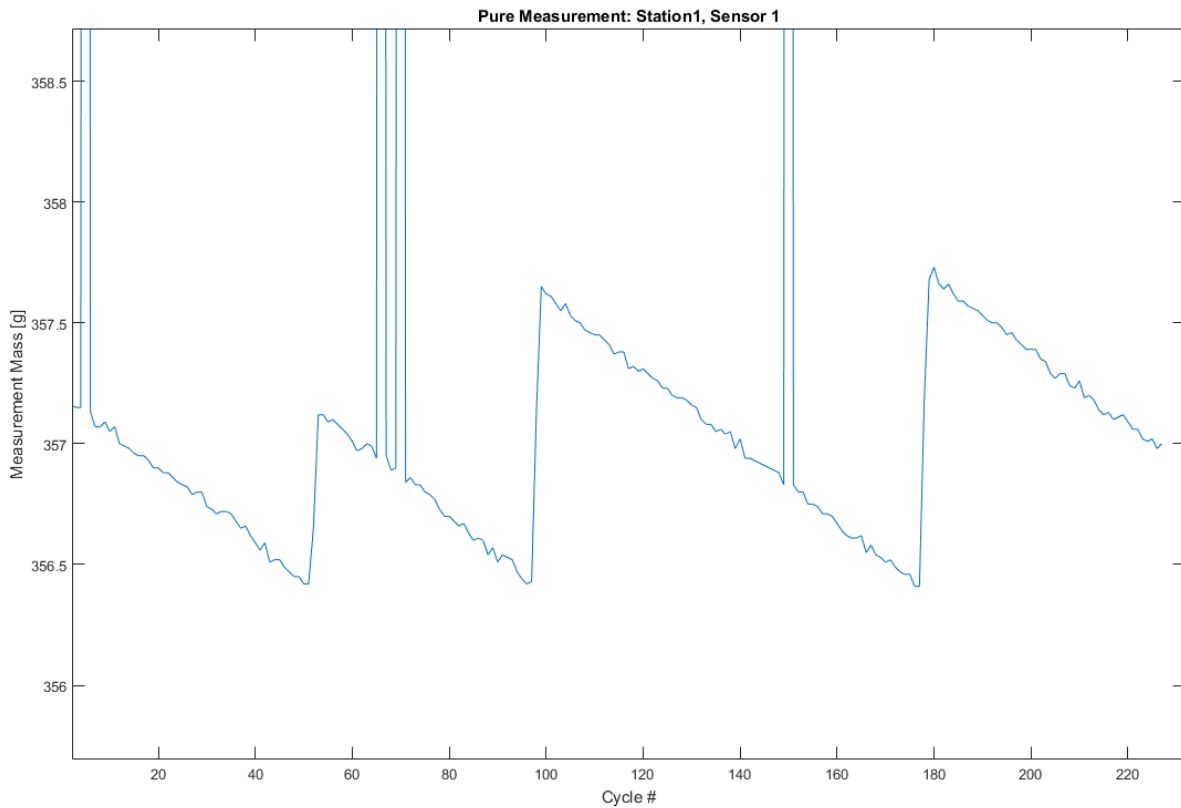


Figure 7.10. Plots while operating. The cup slowly dries out until the mass falls below the threshold. Then the cup gets water so it increases its mass. This is the data from sensor 1 at station 1. X-axis indicates the number of cycles and Y-axis is the measured mass. Spikes are measured from the noise.

#### 7.4. When Errors Occur

This section outlines the likelihood and probability of measurement failures and anomalies. The provided information below is only an assumption learned through this project.

When dealing with the selected ADC and the load cells, many electrical problems that happen are difficult to observe visually. On some occasions, the data would be registering consistently and accurately, then stop registering altogether suddenly. If the data shows a constant reading of 0g, it is an indicator that the ADC is either broken or not

receiving any signal from the load cell. On the other hand, if the measurements seem to be erratic and unstable, connection quality of the wiring and hardware might be the cause. The numerous wire extensions and soldered joints are susceptible to disconnections and poor connectivity. As a part of future works, these issues could be alleviated by utilizing connectors instead of soldered joints. Also, placing ADC and other electronics in locations that would eliminate the need for complex wiring and extension which could help avoid these situations. The last time when data is checked at Day 31, sensor #7 reads various weights during the one third of an entire cycles then becomes 0 g after all. The issue is found on the load cell wires. One of the load cell wire connections is torn although it is soldered. It is fixed; however, it is important to secure these connections from movements.

## CHAPTER 8

### CONCLUSION AND FUTURE WORK

#### 8.1. Conclusions

This thesis introduces an automated soil humidity machine design. The main objective is to develop a device with lower complexity, stronger operational reliability, increased capability, and relatively smaller size than similar devices on the market while at a significantly lower cost. Although a prototype climate control machine needs more development on this system, a soil humidity machine showed functional manipulation of humidity control. In addition, it is designed and constructed at a relatively low cost, a total of \$1,412. Despite this, it provides precise data with a standard deviation of +/- 1 g and is simple to build. Experiments show that this machine can last about 9 months while each experiment spans only 3 months. Moreover, this machine does not need to be isolated in a room like existing machines but it is mobile, and compact, allowing use of multiple systems in a given space. These features lend this system to be used in any situation where performing phenotyping and evaluation operations are needed.

#### 8.2. Prototype Future Improvements

In this section, future works are discussed. Three main improvements required are developing a printed circuit board (PCB), a machine design improvement, and cost considerations.

### 8.2.1. PCB Development

For one future work, PCB development should be considered. As discussed earlier, the wire connections lead to the most prominent reliability issues in this machine. By integrating the prototype boards and other electronics into one single PCB board, numerous wires will be eliminated, and shorter wires or printed wires create less noise from both wire resistances and connections. It would be also easier if the board is compact and can be packaged and in protected areas of the carrier. In addition, troubleshooting and serviceability can be made easier by using a PCB with integrated connectors that can interface with quick disconnect connectors integrated into the load cell or motor control systems.

### 8.2.2. Mechanical Development

Another criteria for improvement in mechanical design the location of the belt attachment on a carrier. This would be a small and the simple change. Relocating the timing belt away from the water tubes will reduce the likelihood of the parts interfering with each other. When in close proximity, the timing belt could cause wear on the water tube, causing leaks or damage. Separating these elements would eliminate this issue.

In addition, it was previously mentioned that a washer in a pulley system experienced problems associated with wear and friction. The rotation of a pulley causes an excessive wear between a nylon washer and the pulley mount. The nylon washer is used because of its low friction, but it is not durable enough to withstand constant cycling, especially without regular lubrication. This washer is effectively acting as a bearing surface between the pulley and the mount, which it is not designed to do. Although lubrication helps a little bit, a design of the mechanism should be reconsidered. Incorporating a more durable material such as brass or a thrust bearing in place of this washer would potentially prolong its service life beyond that required by the reliability standards.

### 8.2.3. Cost Improvement

The last improvement is a cost. Compared to the existing machines, this machine is 1/100th the cost; however, since the soil humidity machine is a part of a climate control machine, this only represents a portion of the possible cost. Practical parts to be considered are listed below:

- Stepper Motor
- Stepper Motor Driver
- Solenoid
- Relay

Based on the experiment and analysis, the stepper motor is more than capable of performing its job. The Y axis motor could be substituted with a less expensive one. The stepper motor drivers could also be replaced with streamlined, less expensive units that would be cheaper if procured in bulk. The best driver is one that can handle multiple axis, and control 3 motors utilizing just one driver. Because the micro-stepping functions are unnecessary, and the stepper motors can be downsized, using a less sophisticated driver is a very feasible, and cost effective solution. The next component fit for cost reduction is the solenoid. Using 12 solenoids at \$20 each is an excessive cost. The solenoids when purchased in bulk would be much less expensive, so this cost could likely be reduced purely by using alternative vendors. However, it might be possible to implement a water delivery system capable of using only a single solenoid to portion water out to multiple plants. Being able to replace the 12 solenoids with one solenoid would automatically reduce costs greatly. Last cost reduction item is the relay. The selected relay is simple to use and reliable, making it suitable for prototyping. However, once a custom PCB is introduced, it is easy to replace a set of relays with a transistor. A transistor would automatically reduce the number of wires hanging around and would be relatively cheaper than a relay.

### 8.3. Plant Growing Pictures

On March 1<sup>st</sup> 2016, Arabidopsis seeds are placed in each cups. Since then, the soil humidity machine has been controlling humidity for each cups automatically. After about a week or so, some seeds sprout and begin to grow. The last discussion is a brief explanation of current situation.

#### 8.3.1. Current Situations

Plant interaction with the machine is explained in this section. Currently the machine is operating constantly about almost a month. Once a week, the machine is turned off and data saved in microSD card is taken to the computer and saved for a future analysis. While the machine is powered off, pictures of all cups are taken manually and imported to the computer. When the data is all transferred, microSD card is cleared and placed back to the machine. Then machine is turned on again. Because the program is already compiled in the Arduino, it is now simply watching the machine to go back to the home cycle.

Pictures taken during the experiment is shown in Figure 8.1 and Figure 8.2. Figure 8.1 shows the three different cups on the experiment day 21 have plants growing. The following figure (Figure 8.2) shows the same plant growing on three different days. The first picture is taken at Day 7 when some of the seeds start germinating. The middle picture is at Day 21, two weeks after the first picture is taken. The last picture on the right is taken at Day 38, about two weeks after the second picture.



Figure 8.1. Plants growth – Day 21



Figure 8.2. Plants growth – Cup#3.3: Day7 (Left), Day21 (Center), Day38 (Right)

## APPENDIX A

### ARDUINO SOFTWARE

```

#include <AccelStepper.h>
#include "HX711.h" //for load cells
#include "SD.h"
#include "SPI.h"

AccelStepper stepper1y(1, 2, 3); //20,19
(Yellow_Pul, White_Dir)
AccelStepper stepper1z(1, 17, 18);
//18,17 (Blue, Green)
AccelStepper stepper2z(1, 15, 16);
//16,15 (Blue_Pul, Green_Dir)
#define LimitY1 23 //blue
#define LimitY2 22 //yellow
#define LimitZ1 13
#define LimitZ2 12

int x = 1;
int trig = 67;
int pz;
int i;
int Time;
const int m = 4; // Number of stations
const int n = 8; //Number of sensors
float Error[n][m]={};

float M[n][m] =
{{260.41,224.95,219.91,285.04},{220.47
,281.30,281.38,260.98},{260.41,223.29,
240.76,242.89},{260.47,280.94,279.68,2
40.86},{260.41,244.48,240.32,283.03},{
260.41,282.21,262.51,242.83},{220.41,2
20.41,260.75,261.55},{261.60,220.47,24
0.60,221.87}};

float D[n][m] = {};

const int chipSelect = 48;

/*Set-up the size parameters of the
systems*/

int ym = 6600; //Max y size
int yn = 1050; //Number of steps
between measurements

int zm = 5000;

/*Set-up speed and acceleration
parameters*/
int sr = 500; //Max speed during
operation
int ar = 500; //Acceleration in general
int sh = 300; //Max speed during homing

/*Set-up delays*/
int d1 = 3000; //Watering Delay

float measured;

//pins for loads
HX711 scale1(40, 41);
//scale(DT,SCK);
HX711 scale2(44, 45); // parameter
"gain" is omitted; the default value 128
is used by the library
HX711 scale5(36, 37);
HX711 scale6(32, 33); // parameter
"gain" is omitted; the default value 128
is used by the library
HX711 scale7(42, 43);
HX711 scale8(34, 35); // parameter
"gain" is omitted; the default value 128
is used by the library
HX711 scale11(28, 29);
HX711 scale12(24, 25); // parameter
"gain" is omitted; the default value 128
is used by the library

int sel1 = 4; //pin selenoid 1
int sel2 = 5; //pin selenoid 2
int sel5 = 8; //pin selenoid 3

```

```

int sel6 = 9; //pin selenoid 4
int sel7 = 6; //pin selenoid 5
int sel8 = 7; //pin selenoid 6
int sel11 = 10; //pin selenoid 7
int sel12 = 11; //pin selenoid 8

int station = 1; //Initial Setting
int max_ymove = m; // number of
stations
int movenum = 0; // This has been made
redandunt by constant variable m
int cycle = 1;

void setup() {
  Serial.begin(9600);
  while(!Serial)
  {
    ;
  }
  delay(500);
  stepper1y.setMaxSpeed(sr);
  stepper1y.setAcceleration(ar);
  stepper1z.setMaxSpeed(sr);
  stepper1z.setAcceleration(ar);
  stepper2z.setMaxSpeed(sr);
  stepper2z.setAcceleration(ar);
  pinMode(LimitY1, INPUT);
  pinMode(LimitY2, INPUT);
  pinMode(LimitZ1, INPUT);
  pinMode(LimitZ2, INPUT);
  //for selenoids
  pinMode(sel1, OUTPUT);
  pinMode(sel2, OUTPUT);
  pinMode(sel5, OUTPUT);
  pinMode(sel6, OUTPUT);
  pinMode(sel7, OUTPUT);
  pinMode(sel8, OUTPUT);
  pinMode(sel11, OUTPUT);
  pinMode(sel12, OUTPUT);

  // correction factor by cells
  scale1.set_scale(2056.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details

```

```

  scale2.set_scale(2001.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details
  scale5.set_scale(2059.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details
  scale6.set_scale(2075.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details
  scale7.set_scale(2227.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details
  scale8.set_scale(2122.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details
  scale11.set_scale(2075.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details
  scale12.set_scale(1977.f); // this
value is obtained by calibrating the scale
with known weights; see the README
for details

  scale1.set_gain(128);
  scale2.set_gain(128);
  scale5.set_gain(128);
  scale6.set_gain(128);
  scale7.set_gain(128);
  scale8.set_gain(128);
  scale11.set_gain(128);
  scale12.set_gain(128);

  homez();
  stepper1y.setEnablePin(14); // ENA
pin# = 14;
  stepper1y.enableOutputs();
  homey();

  Serial.println("Initializing SD Card");
  pinMode(53,OUTPUT);

```

```

pinMode(chipSelect,OUTPUT);
if(!SD.begin(chipSelect))
{
  Serial.println("Card failed, or not
present");
  return;
}
Serial.println("Card initialized");
}

void loop()
{
  Serial.println(cycle);
  Time = millis()/1000;
  tare();
  delay(500);
  pz=4400;
  stepper1z.moveTo(pz);
  stepper2z.moveTo(pz);
  while (stepper1z.currentPosition() != pz
|| stepper2z.currentPosition() != pz)
{
  stepper1z.run();
  stepper2z.run();
}

measurement();
delay(500);
Move();
station++;
if (station > max_ymove)
{
  for (int col=0;col<=(m-1);col++)
  {
    for (int row=0;row<=(n-1);row++)
    {
      if (Error[row][col] > 0.00)
      {
        Serial.print("Sensor#:
");Serial.print(row+1);Serial.print(" at
Station#: ");Serial.print(col+1);
        Serial.print(" has an Error about:
");Serial.println(Error[row][col]);
      }
    }
  }
}

station = 1;
}
cycle ++;
}

void tare()
{
  // Serial.println("Tare");
  scale1.tare();
  scale2.tare();
  scale5.tare();
  scale6.tare();
  scale7.tare();
  scale8.tare();
  scale11.tare();
  scale12.tare();
}

void measurement()
{
  // Serial.println("Measurement");
  for (i = 1;i<=n;i++)
  {
    delay(200);
    float thrs=M[i-1][station-1];
    if (cycle > m)
    {
      trig = 7;
    }
    float security = abs(thrs - trig); //42 =
100g 21.25 = 50g
    // Serial.print("Security:
");Serial.print(security);Serial.print("
Threshold: ");Serial.println(thrs);
    D[i-1][station-1] = load(i);
    String dataString = ""; //make a string
for assembling the data to log:
    dataString =
String(millis()/1000)+","+String(station)+
"+String(i)+","+String(D[i-1][station-1]);
    delay(500);
    SDcard(dataString);

//
Serial.print(station);Serial.print(",");Serial

```

```

.print(i);Serial.print(",");Serial.print(millis(
));Serial.print(",");
// Serial.print(D[i-1][station-
1]);Serial.println(",");
    delay(500);
    if (D[i-1][station-1] < security || D[i-
1][station-1] > thrs + 7)
    {
// Serial.println("Cup tipped over or
Measured more than +7g Larger than
Target Mass? Measure again!");
    D[i-1][station-1] = load(i);
    String dataString = ""; //make a
string for assembling the data to log:
    dataString =
String(millis()/1000)+","+String(station)+",
"+String(i)+","+String(D[i-1][station-1]);
    delay(500);
    SDcard(dataString);

```

```

//
Serial.print(station);Serial.print(",");Serial
.print(i);Serial.print(",");Serial.print(millis(
));Serial.print(",");
// Serial.print(D[i-1][station-
1]);Serial.println(",");
    delay(500);
    if (D[i-1][station-1] > thrs + 7)
    {
// Serial.println("More than 7g
Heavier than Target Mass --> No
Water!");
    Error[i-1][station-1] = D[i-1][station-
1] - thrs;
    }
    }

```

```

while (thrs > D[i-1][station-1])
//compare load to threshold and turn on
solenoid if load is below threshodl or
break if not
{
    if (D[i-1][station-1] < security)
    {
        Serial.println("Cup tipped over!");
        digitalWrite(sel1, LOW);
    }
}

```

```

digitalWrite(sel2, LOW);
digitalWrite(sel5, LOW);
digitalWrite(sel6, LOW);
digitalWrite(sel7, LOW);
digitalWrite(sel8, LOW);
digitalWrite(sel11, LOW);
digitalWrite(sel12, LOW);
Error[i-1][station-1] = D[i-
1][station-1] - thrs;
// Serial.print("Error:
");Serial.println(Error[i-1][station-1]);
    delay(300);
    break;
}
else
{
    Error[i-1][station-1] = 0;
}

```

```

Serial.print(station);Serial.print(",");Serial
.print(i);Serial.print(",");delay(500);
switch(i)
{
    case 1:
        digitalWrite(sel1, HIGH);
        D[i-1][station-1] =
abs(scale1.get_units(5)); //read Load
cell

```

```

Serial.print(millis());Serial.print(",");Serial
.println(D[i-1][station-1]); delay(500);
    if (thrs <= D[i-1][station-1])
    {
        digitalWrite(sel1, LOW);
        break;
    }
    break;
    case 2:
        digitalWrite(sel2, HIGH);
        D[i-1][station-1] =
abs(scale2.get_units(5)); //read Load
cell

```

```

Serial.print(millis());Serial.print(",");Serial
.println(D[i-1][station-1]); delay(500);

```



```

    String dataString = ""; //make a
string for assembling the data to log:
    dataString =
String(millis()/1000)+","+String(station)+"
","+String(i)+","+String(D[i-1][station-1]);
    delay(500);
    SDcard(dataString);
} //END While Loop
} // END For loop
}

```

```

void SDcard(String dataString)
{
    File dataFile =
SD.open("data.txt",FILE_WRITE);
    if(dataFile)
    {
        dataFile.println(dataString);
        dataFile.close();
        Serial.println(dataString);
    }
    else
    {
        Serial.println("error opening data.txt");
    }
}

```

```

float load(int i)
{
    switch (i)
    {
        case 1:
            D[i-1][station-1] =
abs(scale1.get_units(10));//measure
load
            return D[i-1][station-1];
        case 2:
            D[i-1][station-1] =
abs(scale2.get_units(10));//measure
load
            return D[i-1][station-1];
        case 3:
            D[i-1][station-1] =
abs(scale5.get_units(10));//measure
load
            return D[i-1][station-1];

```

```

        case 4:
            D[i-1][station-1] =
abs(scale6.get_units(10));//measure
load
            return D[i-1][station-1];
        case 5:
            D[i-1][station-1] =
abs(scale7.get_units(10));//measure
load
            return D[i-1][station-1];
        case 6:
            D[i-1][station-1] =
abs(scale8.get_units(10));//measure
load
            return D[i-1][station-1];
        case 7:
            D[i-1][station-1] =
abs(scale11.get_units(10));//measure
load
            return D[i-1][station-1];
        case 8:
            D[i-1][station-1] =
abs(scale12.get_units(10));//measure
load
            return D[i-1][station-1];
    }
}

```

```

void Move()
{
    stepper1y.enableOutputs();
    homez();

    if (movenum == max_ymove - 1)
    {
        stepper1y.moveTo(0);
        while (stepper1y.currentPosition() != 0
&& digitalRead(LimitY1) == 0 &&
digitalRead(LimitY2) == 0)
        {
            stepper1y.run();
        }
        homey();
        movenum = -1;
    }
    movenum++;
}

```

```

stepper1y.moveTo(-movenum * yn);
while (stepper1y.currentPosition() != -
movenum * yn && digitalRead(LimitY1)
== 0 && digitalRead(LimitY2) == 0)
{
  stepper1y.run();
}
if (digitalRead(LimitY1) == 1)
{
//  Serial.println("Hit Right Limit");
stepper1y.setCurrentPosition(100);
stepper1y.runToNewPosition(0);
movenum = 0;
delay(1000);
homey();
}
else if (digitalRead(LimitY2) == 1)
{
//  Serial.println("Hit Left Limit");
stepper1y.setCurrentPosition(-100);
stepper1y.runToNewPosition(0);
movenum = 0;
delay(1000);
homey();
}
stepper1y.disableOutputs();
}

void homey() {
//  Serial.println("Y Homing Cycle");
stepper1y.setMaxSpeed(sh);
stepper1y.moveTo(ym);
while (digitalRead(LimitY1) == 0)
{
  stepper1y.run();
}
stepper1y.stop();
stepper1y.setCurrentPosition(0);
stepper1y.setMaxSpeed(sr);
stepper1y.runToNewPosition(-30);
stepper1y.setMaxSpeed(sh / 10);
stepper1y.moveTo(5);
while (digitalRead(LimitY1) == 0) {
  stepper1y.run();
}
stepper1y.stop();

```

```

stepper1y.setCurrentPosition(100);
stepper1y.setMaxSpeed(sr);
stepper1y.runToNewPosition(0);
}

void homez()
{
//  Serial.println("Z Homing Cycle");
if (digitalRead(LimitZ1) == 1 ||
digitalRead(LimitZ2) == 1)
{
  stepper1z.setCurrentPosition(0);
  stepper2z.setCurrentPosition(0);
  pz = 400;
  runz(pz);
}
stepper1z.moveTo(-zm);
stepper2z.moveTo(-zm);
while (digitalRead(LimitZ1) == 0 ||
digitalRead(LimitZ2) == 0)
{
//Serial.println("Z switches are 0");
if (digitalRead(LimitZ1) == 0)
{
  stepper1z.run();
}
else
{
  stepper1z.stop();
  stepper1z.moveTo(-zm);
}
if (digitalRead(LimitZ2) == 0)
{
  stepper2z.run();
}
else
{
  stepper2z.stop();
  stepper2z.moveTo(-zm);
}
}
stepper1z.setCurrentPosition(0);
stepper2z.setCurrentPosition(0);
pz = 400;
runz(pz);

```

```
}  
while (stepper1z.currentPosition() != pz  
|| stepper2z.currentPosition() != pz)  
{  
  stepper1z.run();  
  stepper2z.run();  
}  
void runz(int pz)  
{  
  stepper1z.moveTo(pz);  
  stepper2z.moveTo(pz);  
}
```

## APPENDIX B

### Data Logging Example

Time [sec], Station#, Sensor#, Weight [g]

31,1,1,263.20	267,1,4,263.94	502,1,7,232.18
34,1,2,221.45	270,1,5,263.23	505,1,7,232.20
37,1,3,264.52	273,1,6,263.02	508,1,8,264.93
40,1,4,263.95	275,1,7,232.30	541,2,1,226.74
43,1,5,263.25	278,1,7,232.28	544,2,2,283.52
46,1,6,263.06	281,1,8,264.94	546,2,3,227.52
48,1,7,232.25	314,2,1,226.74	549,2,4,281.75
51,1,7,232.25	316,2,2,283.49	552,2,5,244.58
54,1,8,265.01	319,2,3,227.51	555,2,6,285.45
87,2,1,226.73	322,2,4,281.76	557,2,7,232.09
89,2,2,283.51	325,2,5,244.52	560,2,7,232.06
92,2,3,227.51	328,2,6,285.45	563,2,8,223.77
95,2,4,281.71	330,2,7,232.17	596,3,1,223.11
98,2,5,244.57	333,2,7,232.19	598,3,2,283.76
101,2,6,285.53	336,2,8,223.85	601,3,3,241.05
103,2,7,232.06	369,3,1,223.11	604,3,4,282.98
106,2,7,232.08	372,3,2,283.77	607,3,5,243.41
109,2,8,223.79	374,3,3,241.08	610,3,6,265.04
142,3,1,223.13	377,3,4,283.01	612,3,7,271.04
144,3,2,283.80	380,3,5,243.37	615,3,7,271.03
147,3,3,241.11	383,3,6,265.02	618,3,8,242.16
150,3,4,282.99	386,3,7,270.69	651,4,1,287.81
153,3,5,243.41	388,3,7,270.71	654,4,2,261.59
156,3,6,265.02	391,3,8,242.20	656,4,3,246.20
158,3,7,270.64	424,4,1,287.82	659,4,4,241.12
161,3,7,270.64	426,4,2,261.56	662,4,5,286.04
164,3,8,242.23	429,4,3,246.21	665,4,6,244.99
197,4,1,287.81	432,4,4,241.13	667,4,7,276.31
199,4,2,261.60	435,4,5,286.05	670,4,7,276.30
202,4,3,246.22	438,4,6,245.00	673,4,8,225.59
205,4,4,241.12	440,4,7,276.39	713,1,1,263.10
208,4,5,286.07	443,4,7,276.39	715,1,2,221.56
210,4,6,245.01	446,4,8,225.65	718,1,3,264.47
213,4,7,276.39	486,1,1,263.14	721,1,4,263.94
216,4,7,276.39	488,1,2,221.55	724,1,5,263.25
219,4,8,225.58	491,1,3,264.56	727,1,6,262.94
259,1,1,263.16	494,1,4,263.94	729,1,7,232.27
261,1,2,221.53	497,1,5,263.26	732,1,7,232.27
264,1,3,264.54	500,1,6,263.00	735,1,8,264.88

768,2,1,226.76	834,3,5,243.39	900,4,8,225.46
771,2,2,283.50	837,3,6,265.02	940,1,1,263.13
773,2,3,227.59	840,3,7,270.51	943,1,2,221.50
776,2,4,281.73	842,3,7,270.49	946,1,3,264.55
779,2,5,244.58	845,3,8,242.22	948,1,4,263.93
782,2,6,285.47	878,4,1,287.80	951,1,5,263.24
784,2,7,232.04	881,4,2,261.62	954,1,6,262.96
787,2,7,232.04	884,4,3,246.22	957,1,7,232.21
790,2,8,223.80	886,4,4,241.12	959,1,7,232.20
823,3,1,223.10	889,4,5,286.02	962,1,8,264.95
826,3,2,283.76	892,4,6,244.98	995,2,1,226.73
829,3,3,241.09	895,4,7,276.37	998,2,2,283.51
831,3,4,282.99	897,4,7,276.36	1001,2,3,227.49

## BIBLIOGRAPHY

---

Alonso-Blanco, C. et al., 2009. What has natural variation taught us about plant development, physiology, and adaptation?. *American Society of Plant Biologists*, 21(2009).

Assessment, N. C., 2014. *Future Climate Change*. [Online]

Available at: <http://nca2014.globalchange.gov/>

[Accessed 31 3 2016].

Bergelson, J., Stahl, E., Dudek, S. & Kreitman, M., 1998. Genetic variation within and among populations of *Arabidopsis thaliana*. *PubMed*, Issue 1998, pp. 1311-1323.

Bomblies, K. et al., 2010. Local-Scale Patterns of Genetic Variability, Outcrossing, and Spatial Structure in Natural Stands of *Arabidopsis thaliana*. *Plos Genetics*.

Crowder, R., 2006. *Electric Drives and Electromechanical System*. Great Britain: Elsevier.

Dalpe, S., 2003. *Alberta - Agriculture and Forestry*. [Online]

Available at: [http://www1.agric.gov.ab.ca/\\$department/deptdocs.nsf/all/opp2892](http://www1.agric.gov.ab.ca/$department/deptdocs.nsf/all/opp2892)

[Accessed 31 3 2016].

Desigaux, M., n.d. *Per.com*. [Online]

Available at: [per.com](http://per.com)

ESRL, 2015. *Earth System Research Laboratory - Global Monitoring Division*. [Online]

Available at: <http://www.esrl.noaa.gov/gmd/ccgg/trends/>

[Accessed 15 12 2015].

Foundation, N. S., 2013. *Arabidopsis: The Model Plant*. [Online]

Available at: <http://www.nsf.gov/pubs/2002/bio0202/model.htm>

[Accessed 10 2 2016].

Hunt, S., n.d. *Per.com*. [Online]

Available at: [per.com](http://per.com)

lepse, 2016. *The PHENOPSIS platform and experiments*. [Online]

Available at: <http://bioweb.supagro.inra.fr/phenopsis/InfoBDD.php>

[Accessed 31 3 2016].

Inc., 8., 2015. *80/20 Inc. The Industrial Erector Set*. [Online]

Available at: <https://8020.net/>

[Accessed 2 3 2016].

Instruments, N., 2008. *Redundant System Basic Concepts*. [Online]

Available at: <http://www.ni.com/white-paper/6874/en/>

[Accessed 10 2 2016].

ipcc, 2007. *Intergovernmental Panel on Climate Change*. [Online]

Available at: [http://www.ipcc.ch/publications\\_and\\_data/ar4/wg1/en/ch10s10-6-2.html](http://www.ipcc.ch/publications_and_data/ar4/wg1/en/ch10s10-6-2.html)

[Accessed 15 3 2016].

Koornneef, M., Alonso-Blanco, C. & Vreugdenhil, D., 2004. Naturally occurring genetic variation in *Arabidopsis thaliana*. *Plant*, Volume 55, pp. 141-172.

- Le Corre, V., 2005. Variation at two flowering time genes within and among populations of *Arabidopsis thaliana*: comparison with markers and traits.. *PubMed*, pp. 4181-4192.
- Loudet, O., n.d. *Per.com*. [Online]  
Available at: [per.com](http://per.com)
- Mitchell-Olds, T. & Schmitt, J., 2006. Genetic mechanisms and evolutionary significance of natural variation in *Arabidopsis*. *nature*, Issue 2006.
- Morison, J. I. & Lawlor, D. W., 1999. Interaction between increasing CO<sub>2</sub> concentration and temperature on plant growth. *Plant, Cell & Environment*, 22(6), pp. 689-682.
- Motor, O., n.d. *Basics of Motor Control*. [Online]  
Available at: <http://www.orientalmotor.com/technology/articles/motor-sizing-calculations.html>  
[Accessed 4 2015].
- NASA, 2016. *Global Climate Change*. [Online]  
Available at: <http://climate.nasa.gov/causes/>  
[Accessed 10 2 2016].
- Platt, A., Horton, M., Huang, Y. S. & Li, Y., 2010. The Scale of Population Structure in *Arabidopsis thaliana*. *Plos, Genetics*, Volume 6, pp. 1-8.
- Rausand, M. & Hayland, A., 2004. System Reliability Theory - Models, Statistical Methods, and Applications. In: *System Reliability Theory*. s.l.:Wiley-Interscience.
- Rohde, R. A., 2009. *Global Warming Predictions.png*. [Online]  
Available at: [http://www.globalwarmingart.com/wiki/File:Global\\_Warming\\_Predictions\\_png](http://www.globalwarmingart.com/wiki/File:Global_Warming_Predictions_png)  
[Accessed 6 4 2016].
- S.Dhondt, n.d. *Per.com*. [Online]  
Available at: [per.com](http://per.com)
- Semiconductor, 海., n.d. *24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales*, China: s.n.
- Slattery, C. & Nie, M., 2005. A Reference Design for High-Performance, Low-Cost Weigh Scales. *Analog Dialogue*, Volume 39.
- Solutions, D. T., 2009. *Greenhouse Water Management Systems*. [Online]  
Available at: [http://www.deltatsolutions.com/gh/gh\\_water.html](http://www.deltatsolutions.com/gh/gh_water.html)  
[Accessed 31 3 2016].
- Song, B.-H. & Mitchell-Olds, T., 2011. Evolutionary and Ecological Genomics of Non-Model Plants. *NIH Public Access*.
- Tisne, S. et al., 2013. Phenoscope: an automated large-scale phenotyping platform offering high spatial homogeneity. *The Plant Journal*, Volume 74, pp. 534-544.
- Turner, N. C., 1981. *Plant and Soil*. s.l.:Springer.
- WIWAM, 2013. *WIWAM Conveyor*. [Online]  
Available at: <http://wiwam.be/phenotyping-system/wiwam-conveyor/>  
[Accessed 31 3 2016].

WIWAM, 2013. *WIWAM xy*. [Online]

Available at: <http://wiwam.be/phenotyping-system/wiwam-xy/>

[Accessed 31 3 2016].

Wunnenberg, J. & Frank, P. M., 1990. Dynamic model based incipient fault detection concept for robotics. *IFAC 11th Triennial World Congr.*, pp. 61-66.

Zhang, Y. & Shipp, J. L., 2002. Manipulating Plant Moisture Conditions Using Greenhouse High-pressure Fogging. *HortTechnology*, 12(2), pp. 261-267.