ENSC 305W/440W Grading Rubric for Design Specification

Criteria	Details	Marks
Introduction/Background	Introduces basic purpose of the project.	**/05%**
Content	Document explains the design specifications with proper justification for the design approach chosen. Includes descriptions of the physics (or chemistry, biology, geology, meteorology, etc.) underlying the choices.	**/20%**
Technical Correctness	Ideas presented represent valid design specifications that will be met. Specifications are presented using tables, graphs, and figures where possible (rather than over-reliance upon text). Equations and graphs are used to back up/illustrate the science.	**/20%**
Process Details	Specification distinguishes between design details for present project version and later stages of project (i.e., proof-of-concept, prototype, and production versions). Numbering of design specs matches up with numbering for functional specs.	**/15%**
Test Plan	Provides a functional test plan for the present project version. (Note that project success will be measured against this test plan.)	**/10%**
Conclusion/References	Summarizes functionality. Includes references for information from other sources.	**/05%**
Presentation/Organization	Document looks like a professional specification. Ideas follow in a logical manner.	**/05%**
Format Issues	Includes letter of transmittal, title page, abstract, table of contents, list of figures and tables, glossary, and references. Pages are numbered, figures and tables are introduced, headings are numbered, etc. References and citations are properly formatted.	**/10%**
Correctness/Style	Correct spelling, grammar, and punctuation. Style is clear concise, and coherent. Uses passive voice judiciously.	**/10%**
Comments		

E - Plant Innovation

School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

November 14, 2013
Mr. Lakshman One
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Design Specification for the SmartPlant™ design

Dear Mr. Lakshman One,

We are writing with regards to our design specification for SmartPlant™, which describes our prospective project for Engineering Science 440. The objective of our design is to inspire people to grow their own fruits and vegetables. The SmartPlant™ is a smart flower pot, which has an automated watering system and shading system to protect the plant from dehydration and sun damage. The product also comes with a SmartPlant™ application, which helps users to gain some knowledge about gardening.

This document outlines all of the necessary information and specifications for our proposed SmartPlant™. The purpose of this document is to provide an in-depth look into the design process required to meet the basic functional specifications for a proof-of-concept model. We will also occasionally be discussing future design considerations that will be required for a finished, marketable product; however these iterations will not be implemented in this stage of development.

E-Plant Innovation is composed of four talented and motivated team members whose knowledge and skill set is superlative. These include, David Hsu, Jae Sung Park, Mandan Vahabzadeh and Yang Zhang who are all fourth-year engineering students majoring in electronics and systems engineering.If you have any questions or concerns regarding our functional specification, please do not hesitate to contact me anytime at mvahabza@sfu.ca.

Sincerely,

Mandan Vahabzadeh

Mandan Vahabzadeh
Chief Executive Officer (CEO)
E-Plant Innovation

E - Plant Innovation

Team Members:

David Hsu
Jae Sung Park
Mandan Vahabzadeh
Yang Zhang

Date: November 14, 2013

Executive Summary

The home gardening has been getting more popular in recent years. Growing fruits and vegetables seems overwhelming to most of us, but all you need is soil, a little time and a SmartPlant™ device. “Backyard gardening can inspire you to take an interest in the origins of your food and make better choices about what you put on your plate. When you grow your own food, you savor it more because of the effort it took to get to the table.” says Dr. Helen Delichatsios, an internist at Harvard-affiliated Massachusetts General Hospital [1].

This document will outline the design specification for E-Plant Innovation and provide justification for certain design consideration. Also design improvements for future final product of the E-Plant innovation will be discussed in this report. The design specifications in this document are only for proof-of-concept model. Therefore, we will only embody the functional requirements labelled A or B in the functional specification.

The SmartPlant™ device consist of three major parts: watering system, shading system and SmartPlant™ android application. The watering system has moisture sensors which detect the moist level in the soil and send the data to a microcontroller. Then, the microcontroller will activate the water pump which water will be transferred from a water tank to the flowerpot. The shading system has photo resistor sensors which sense the sunlight. When sunlight level exceeds an upper threshold level, sunroof will be activated. Moreover, the SmartPlant™ android application has plant encyclopedia which provides information such as water requirement, temperature and indoor or outdoor planting. This application can send data to the microcontroller by Bluetooth in order to adjust the watering and shading system.

Furthermore, at E-Plant Innovation we attempt for complete and easy gardening solutions. Thus our design will undergo a series of testing to ensure compliance with functionally accurate results and ease of use.

In conclusion, we aim to develop an affordable and simple design to motivate everyone to grow their own plants. This document will describe and justify the design choices for the watering system, shading system and SmartPlant™ android application.

Table of Contents

Executive Summary 3
List of Figures 6
List of Tables 6
Glossary 7
1.0 Introduction 8
1.1 Scope 9
1.2 Intended Audience 9
2.0 System Specification 10
3.0 Watering System 12
3.1 Hardware 12
3.1.1 Power Supply 12
3.1.2 Moisture Sensor 12
3.1.3 Water Pump 14
3.1.4 Water level detector 15
3.2 Signal Processing 16
3.3 Mechanical Design 18
3.3.1 Controller Packaging 18
3.4 Safety and Environmental Protection Consideration 20
4.0 Shade System 21
4.1 Mechanical Design 21
4.2 Electrical Design 24
4.3 Safety Consideration 25
5.0 Mobile App User Interface 25
5.1 Mobile App Development 25
5.2 Mobile App Interface Design 27
6.0 Power Supply 31
7.0 System Test Plan 32
7.1 Typical usage scenario 33

8.0 Conclusion 34

9.0 References 35

List of Figures

Figure 1: SmartPlant™ Model 8
Figure 2: SmartPlant™ Sub-systems 10
Figure 3: YL-69 and Potentiometer (Blue) 13
Figure 4: Schematic Diagram for the Moisture Detection Sensor Module 13
Figure 5: Sensor-Microcontroller Wiring 14
Figure 6: Water Pump-Relay-Microcontroller Wiring 14
Figure 7: High Sensitivity Water Sensor -Arduino Compatible 15
Figure 8: Block Diagram of Watering System 16
Figure 9: Flow Chart of Microcontroller 17
Figure 10: Interior component Layout of controller packaging 19
Figure 11: Shading System Setup 21
Figure 12: Tubular Motor 22
Figure 13: SN754410 H-Bridge Circuit 23
Figure 14: Arduino Light Sensor on Breadboard 24
Figure 15: MIT App Inventor Designer Interface 26
Figure 16: MIT App Inventor Blocks Editor Interface 27
Figure 17: Android App Screen Transition when user selects one of the Models 28
Figure 18: Flow Chart of Profile Selection by Android App 29
Figure 19: Android App Screen Transition when User Selects Encyclopedia 30
Figure 20: Series Connection: Voltages Add, Capacity is Constant 31
Figure 21: Rechargeable Lithium Battery 31

List of Tables

Table 1: Size specifications for interior components of controller package 19
Table 2: Datasheet of Tubular Motor 22
Table 3: Inputs and Outputs of SN754410 23

Glossary

CEO	Chief Executive Officer
TM	Trademark
PC	Personal Computer
Hz	Hertz
DC	Direct Current
AC	Alternating Current
PCB	Printed Circuit Board
UNO	Universal Network Object
A/D	Analog to Digital
LED	Light Emitting Diode
TTL	Transistor-Transistor Logic
LDR	Light Dependent Resistor
MIT	Massachusetts Institute of Technology

1.0 Introduction

At E-Plant Innovation we aim to develop a device that will help and encourage everyone to have their own garden. The SmartPlant™ device have photoresistor sensor, motor and shade in order to protect the plant from excessive sunlight. Also this device waters the plant automatically by having moisture sensor, water pump and water tank. These two systems will be controlled by microcontroller and SmartPlant™ Android application. The Android application will help the user to search for specific plants, which will give instruction and information on how to grow the plant. Also it will send the plant type to microcontroller in order to adjust the watering and shading system. The design of our SmartPlant™ has been broken down into three modules, the watering system, the shading system, and the SmartPlant™ Android application. For each module, a design was chosen and this document will explain our design and provide brief explanations for each part. Figure 1 shows the design for SmartPlant™ device.

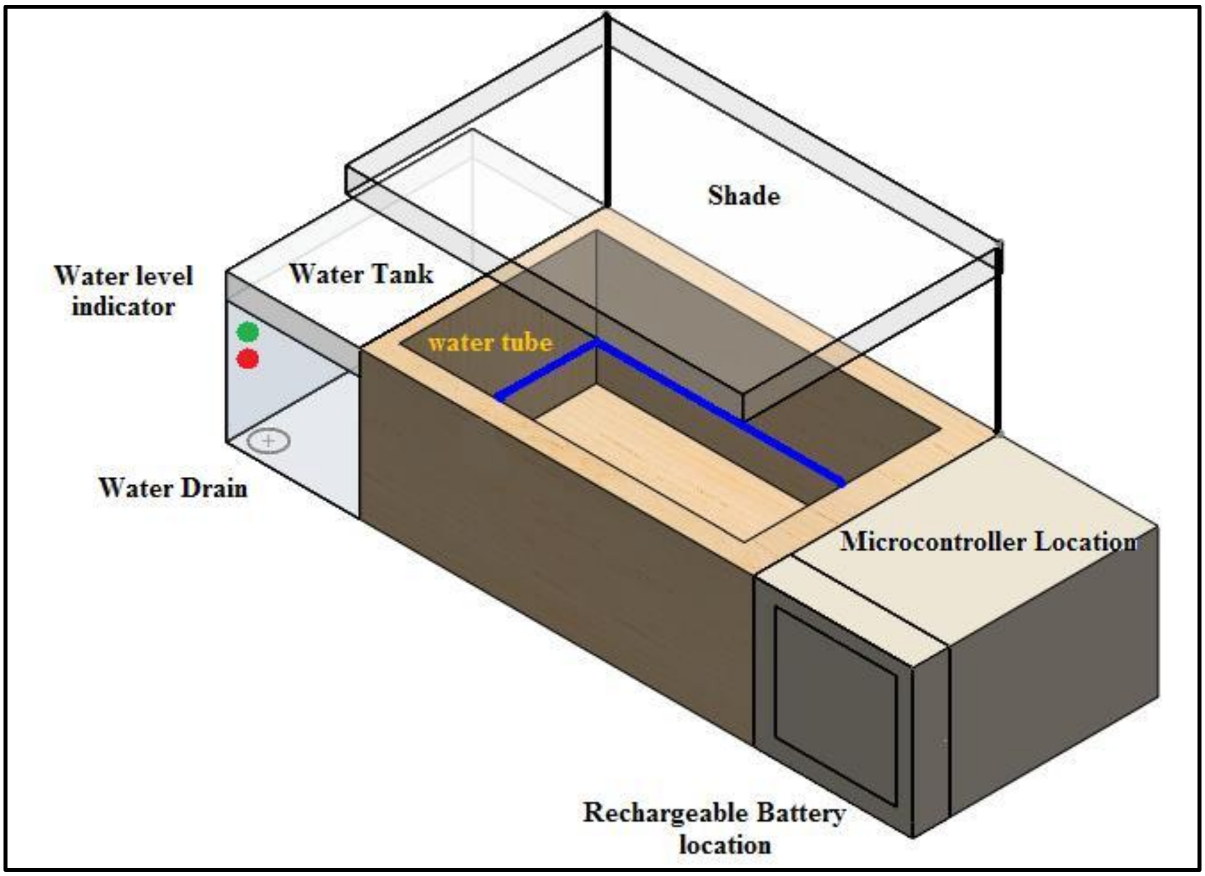

Figure 1: SmartPlant™ Model

1.1 Scope

This document will expand on our reasoning behind our design choices and will lay out the hardware and software design for SmartPlant™. The design specification discusses all the requirements for a proof of concept system which are labelled 'A' and 'B' in the functional specification. Moreover, series of test plans for each module are provided to evaluate the function of our product.

1.2 Intended Audience

The design specification is written as a guideline for implementation process of SmartPlant™, which will be used by all members of E-Plant Innovation. The CEO, Mandan Vahabzadeh, will use this document as a measure of compliance and progress of our project development. Design engineers of E-Plant Innovation will refer to design specification to monitor the progress and aim to satisfy all of the requirements of the system during implementation phase. Lastly, during testing, this document will operate as a template against which the final product is evaluated.

2.0 System Specification

The SmartPlant™ will automatically adjust the watering and shading system, based on the user choice of planting. The product design consists of three main sub-systems: the watering system, the sunlight protection system and the SmartPlant™ Android app user interface, as it shows on the figure 2.

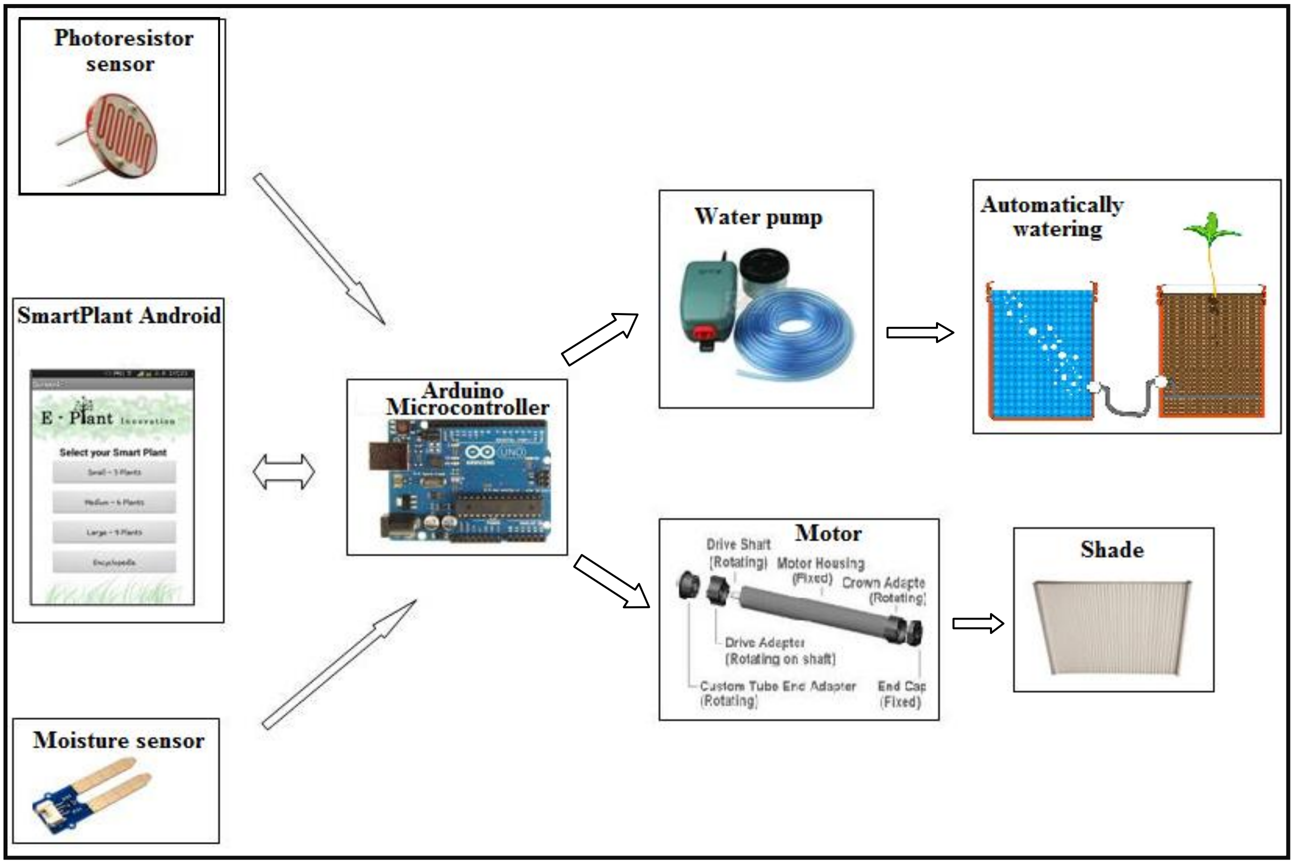

Figure 2: SmartPlant™ Sub-systems

The watering system contains moisture sensor, microcontroller, water tank and water level indicator in order to water the plant automatically. The soil moisture sensor has two probes which can be installed vertically or horizontally. The horizontal orientation is the preferred method, because soil can be easily packed around the probes which will give more accurate measurement of soil moisture. "The electrical capacitance of a capacitor that uses the soil as a dielectric depends on the soil water content"[2]. The moisture sensor senses the moisture level of the soil and sends data to the microcontroller. The way how it works is, when the moisture content increases, the soil resistivity decreases. The resistivity of a dry soil works in the other way around. If the low moisture level is sensed, the microcontroller activates the water pump, which transfers the water from the water tank to the flowerpot. Also the water tank has water level indicator. If the water level is low, the red LED light will be activated to notify the user that water tank need more water.

Some plants need full sun which is 6-8 hours of direct sunlight daily, but prefer afternoon shade. While others can thrive in shade but can handle direct sunlight as long as it is not the hot afternoon sun. The sunlight protection system is consisted of photoresistor sensor, DC tabular motor, microcontroller and shade. Depends on the plant the shading system will be activated by sensing the sunlight amount. The photoresistor sensor detects light and converts light energy to an electric signal output which more light allows more current to flow. The output signal will be sent to Arduino microcontroller. The microcontroller has pre-set thresholds which get compared with the output signals from photoresistor sensor and if the received data is higher than the pre-set values, the microcontroller will activate the shading system.

The SmartPlant™ Android application has three major parts: SmartPlant™ size, plant encyclopedia, notifications and Bluetooth communication. User can select certain plant that user wants to grow or can combine up to three plants which SmartPlant™ app has recommended. Once any of the decision is made from the mobile Android application, it sends data to the microcontroller to select the corresponding profile. In our prototype SmartPlant™, there will be three profiles programmed and saved in the microcontroller. One of the profiles will be assigned for terminating any activated watering or shading system and the other two profiles will initialize the actions for two groups of plants that with different watering and shading requirements. The encyclopedia part has comprehensive database which can be searched using common plant names or botanical name. Our plant encyclopedia provides information about cultivation for each plant, temperature, soil and water requirements and other important information. The SmartPlant™ application also notifies user weekly about checking the water level in water tank as well as recharging the battery.

3.0 Watering System

The watering system is used to water plants whenever their soil level drops below the pre-defined value. Furthermore, it will alert user to refill the water tank when the water level falls below a threshold level. For wiring and setting up the watering system, we will refer to the information provided by Liseman's Garduino project posted on webpage [3]. In this section, the watering system design will be discussed and distributed into various parts: Hardware, signal processing, mechanical design and safety and environmental protection consideration.

3.1 Hardware

The hardware used in E-Plant Innovation's watering system is consisted of the water tank, the soil moisture sensor, the water level sensor and the controller package. The controller packaging includes a battery supply, a PCB with relay, the Arduino UNO microcontroller and the small water pump. In order to meet the requirements of a portable watering system, the controller packaging must be as compact as possible. This section of the document describes the watering system's sub-components and their basic functionalities.

3.1.1 Power Supply

The Arduino Uno microcontroller has quite low power requirement hence it is possible to only use AA batteries as a supply. Specifically, the Arduino board is recommended to operate on 7-12 V of direct current (DC) due to its instability at lower voltage supply levels [4]. This can be achieved by placing a 9 V Alkaline Battery. Furthermore, the relay requires a 5 V power source which can be achieved by the 5 V output pin on the Arduino board; when the pin is at HIGH state, it provides 5V voltage as a source. Finally, the water pump needs a 120 V of alternating current (AC) power supply hence it could be supplied by North American standard power source.

3.1.2 Moisture Sensor

The moisture senor has two probes and uses them to measure soil moisture in the soil by telling how well an electrical current is passed between the two probes. The probes (electrodes) will directly contact with the soil. The more moisture in the soil gives the better conductivity or lowers the electrical resistance [5]. The moisture sensor with LM393 comparator chip, model YL-69 is used and it comes with a digital potentiometer (blue) as shown in Figure 3. The sensor is consisted of 4 pins: a digital output (0 and 1), an analog output, a VCC, and a ground [6]. The sensor can be powered up by connecting the VCC pin to the 5V pin on the microcontroller.

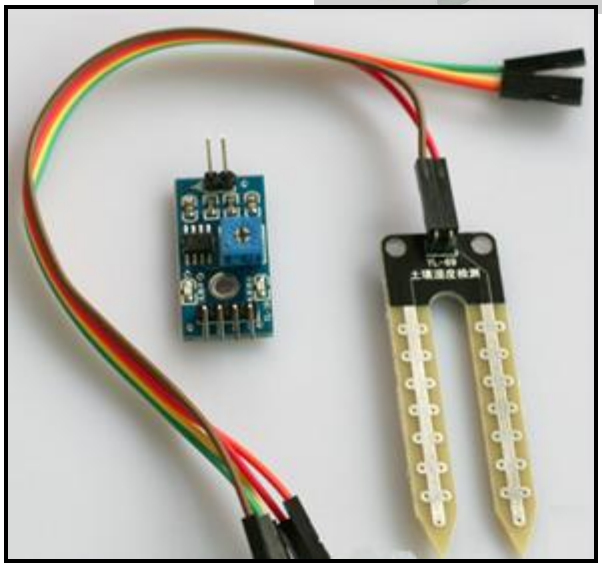

Figure 3: YL-69 and Potentiometer (Blue) [6]

Figure 4 below, shows the schematic for the moisture detection sensor module. Analog output is the voltage measurement between two probes of the sensor. For the digital output, it gets set as HIGH when the measured value is greater than the threshold set by the potentiometer and it gets LOW when the measured value is lower than the threshold. However, due to the fact that the digital output is not accurate [6], the analog output will be used as the analog input to the microcontroller directly connected through jumper wires.

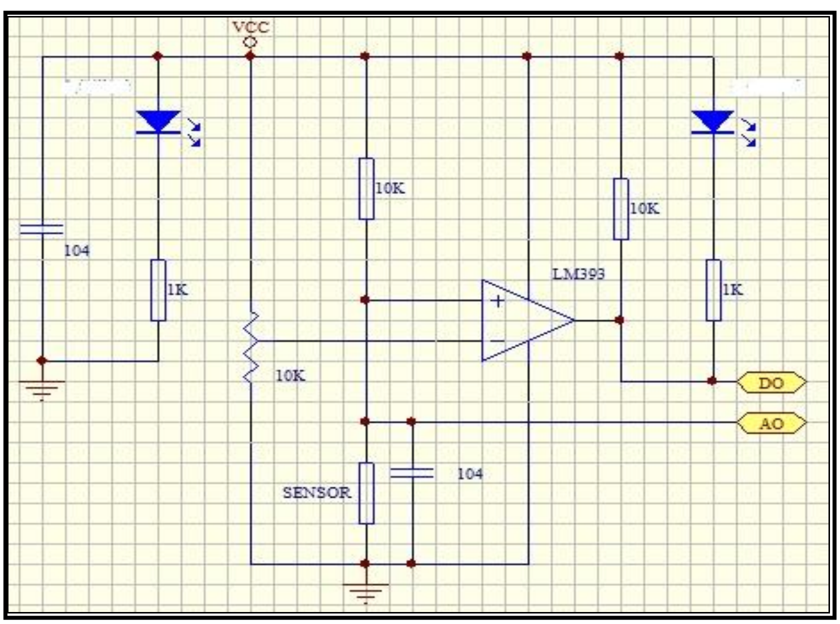

Figure 4: Schematic Diagram for the Moisture Detection Sensor Module [7]

The wiring of the sensor with the microcontroller can be easily set-up. Since the Atmega328P-PU microcontroller used for the Arduino Uno contains an onboard 10-bit 6-channel analog-to-digital (A/D) converter, the analog input pin of Arduino can read analog signals being sent from the sensor and return binary integers from 0 to 1023 [8]. The sensor-microcontroller wiring is shown in Figure 5.

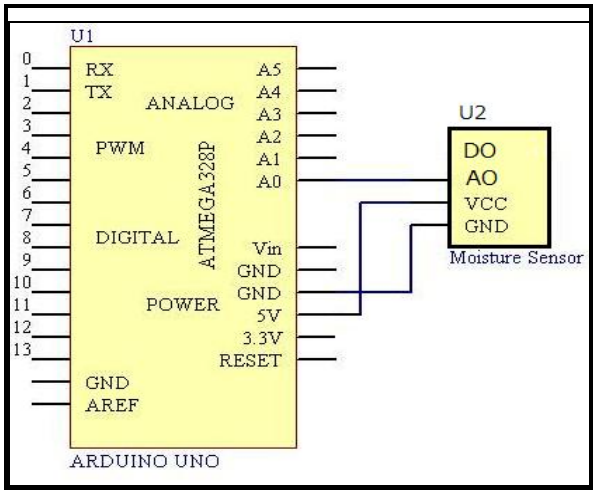

Figure 5: Sensor-Microcontroller Wiring

3.1.3 Water Pump

For watering operation, Aqua Lifer AW20 vacuum pump from Tom Aquarium will be used. This pump operates at voltage of 120V alternating voltage. The pump is designed to pump water up to 30" at a flow rate of 3 1/2 gallons per hour [9] which will be capable enough for simple watering purpose. However, the water pump cannot be activated by microcontroller directly. The pump should be turned on/off with a help of relay. The figure 6 below shows the diagram how the water pump will be connected to microcontroller.

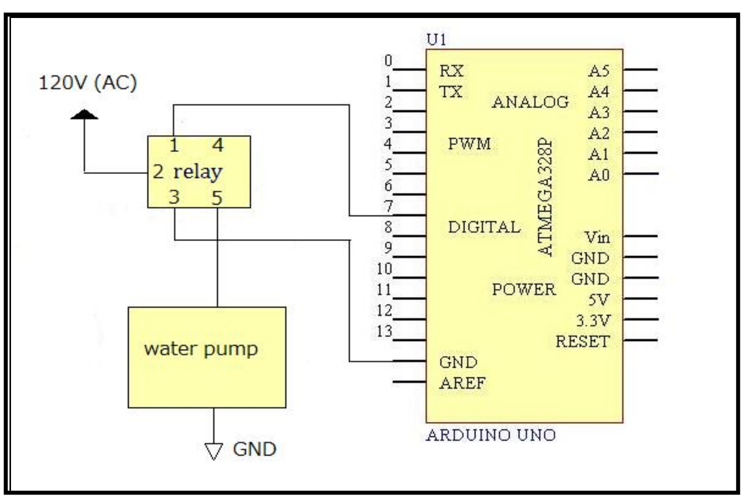

Figure 6: Water Pump-Relay-Microcontroller Wiring

Relay is a switch that is electrically controlled. When the water pump needs to be turned on, the microcontroller will output a 5 V signal (logic HIGH signal), which is needed to activate the relay. When the relay is activated, a current will flow from the external 120V AC power source to the water pump through the relay. Conversely, a logic LOW signal from the microcontroller will turn the pump off.

3.1.4 Water level detector

For water detection, a high-sensitivity water level sensor that is compatible with Arduino UNO is used and it is shown in figure 7 [10]. The sensor has a series of exposed traces and the traces should be completely immersed in water when the reservoir is full. The sensor consists of 3 pins: “S” stands for analog output, “+” stands for voltage supply and “-” stands for ground. This sensor operates at 5 V and outputs voltage signal between 0 and 4.2 V. Output analog value can directly be used in the program function, then to achieve the function of water level alert. The LED is connected to Arduino UNO and is used to prompt the user to refill the water tank. When the water level falls below pre-set threshold value, microcontroller will send a HIGH signal to turn on a LED indicator light.

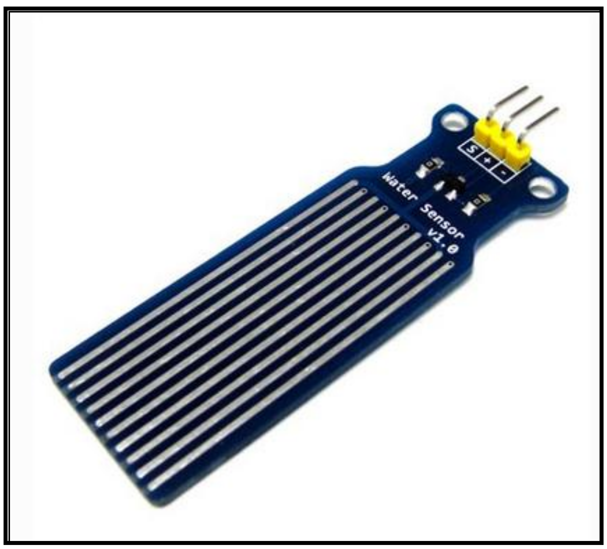

Figure 7: High Sensitivity Water Sensor - Arduino Compatible [10]

3.2 Signal Processing

As Figure 8 shows, the watering system has three inputs that include moisture sensor which will be inserted into flowerpot, the water level sensor which will be mounted on water tank and the mobile application. Moisture sensor read the moisture of soil in flowerpot and water level sensor read the water level in water tank. Moreover, user can set up the threshold values to turn water pump on/off.

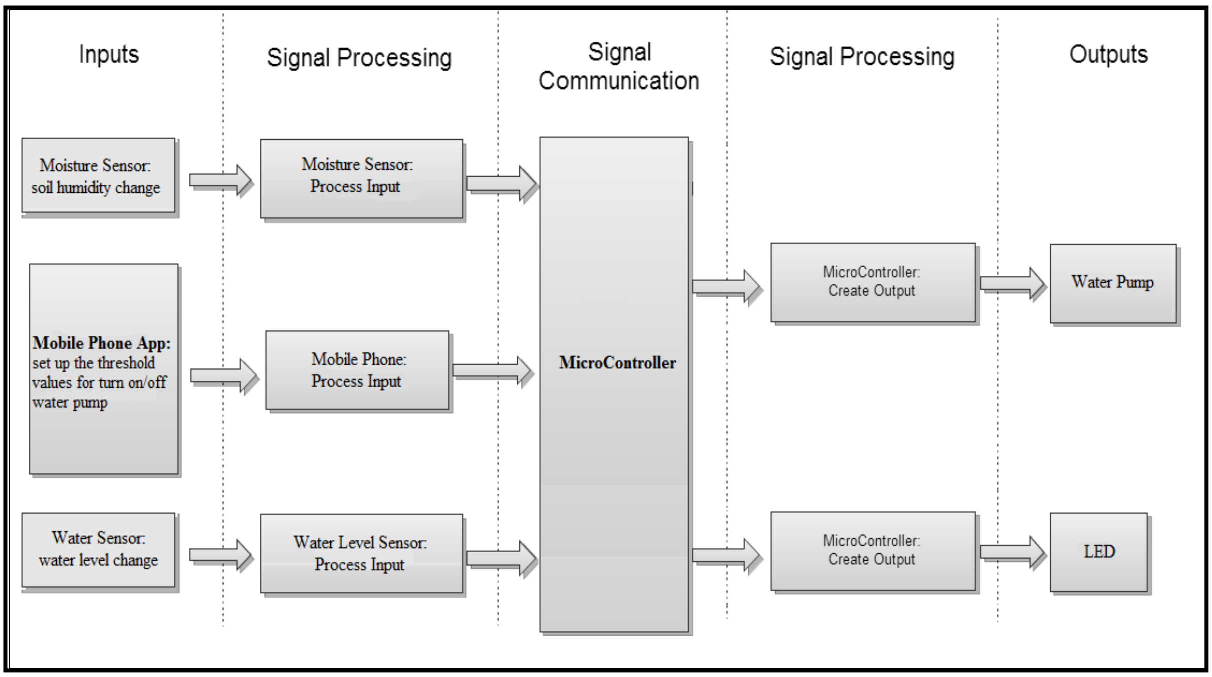

Figure 8: Block Diagram of Watering System

In signal processing stage, the moisture sensor and water level sensor create the analog inputs transmitting to microcontroller. The mobile phone sends the data through Bluetooth module to setup the threshold values at the microcontroller.

In the watering system, the microcontroller has five tasks:

1. Obtaining the data from Android app and select the profiles to set lower and higher threshold.
2. Receiving the moisture values from moisture sensor and comparing them with threshold values.
3. Generating a digital output to turn water pump on/off.

4. Getting the water level values from water level sensor and comparing them with preset values.
5. Generating a digital output to turn LED indicator on/off.

The flow chart of microcontroller is shown below:

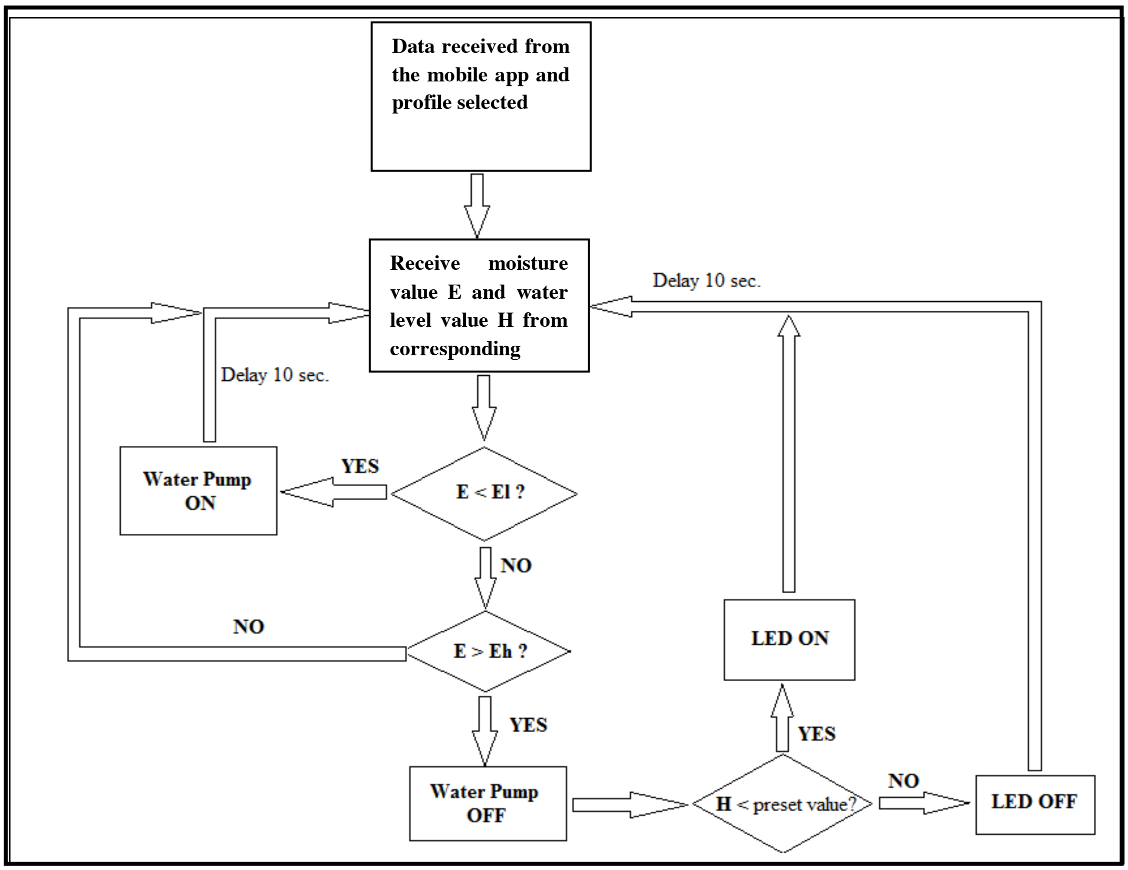

Figure 9: Flow Chart of Microcontroller

After a certain profile is determined at the microcontroller, it will get the moisture value of soil in flowerpot as described in the beginning stage of the loop in the flow chart. Firstly, it will compare the current moisture value E with the lower threshold El. If E < El, the water pump will be turned on to water the plant. After a delay of 10 seconds, the E value must be increased, the microcontroller will read the moisture value and compares it with El again.

When E > El, the E will be then compared with upper threshold. If E > Eh, the water pump will be turn off. If El < E < Eh, nothing happens and it will get back to start point to update the moisture value.

When turning the water pump off, it will test the water level of water container. If the water level value measurement is less than the pre-set value, it means the water tank is nearly empty. The LED indicator will be turned on to alert user to refill water tank. If the water level value is higher than the threshold, the LED will be remained off. Whatever status of the LED, the microcontroller will continue to work. After 10 seconds, it will back to start point.

3.3 Mechanical Design

This section provides the overview of the mechanical design of the watering system. While designing the watering system, the following requirements are considered:

1. For the effective watering, the water pipe with lots of holes should be installed inside and along the sidewall of flowerpot.
2. There is a water drain on the bottom of water tank for draining and a cover on the top for refilling water.
3. There is a LED for indicating water level status for letting user know for water tank refill water.
4. There should be a controller package in a waterproof case beside the flower pot.
5. Controller package, water tank and flowerpot should be designed as uni-body.

The dimension of the flowerpot is 750 mm*200 mm*130 mm. In consideration of exterior appearance, the water tank dimension will be 300 mm*200 mm*130 mm.

3.3.1 Controller Packaging

The controller packaging of watering system is vital to fulfilling its waterproof and insulation requirement. Many electrical components such as Arduino chip, water pump, relay, 9V power supply and bread board must be contained within the controller package. For physical requirements, the packaging case must be firm enough, yet lightweight, in order to withstand harsh weather condition and environment. Upon these requirements, an ABS plastic was chosen for the material of the case.

A fundamental requirement of controller packaging is portability. This is achieved by designing the case that just large enough to contain the necessary electrical components. Table 1 below shows the size requirements for the interior components taken into consideration of layout of the case.

Specification	Water Pump	Arduino Uno	Relay	9V Power Supply	PCB
Shape:	Rectangle	Rectangle	Rectangle	Rectangle	Rectangle
Quantity:	1	1	1	1	1
Dimension:	Length: 100mm Width: 60mm	Length: 75mm Width: 55mm	Length:20mm Width: 15mm	Length:60mm Width: 30 mm	Length:165mm Width: 55mm

Table 1: Size specifications for interior components of controller package

According to the data shown in Table 1, we can construct a layout of electrical components in the case. Figure 10 shows the detail of controller packaging's internal plant layout. Base on the information, we determined the dimension of case to have approximately 230 mm length and 120 mm height.

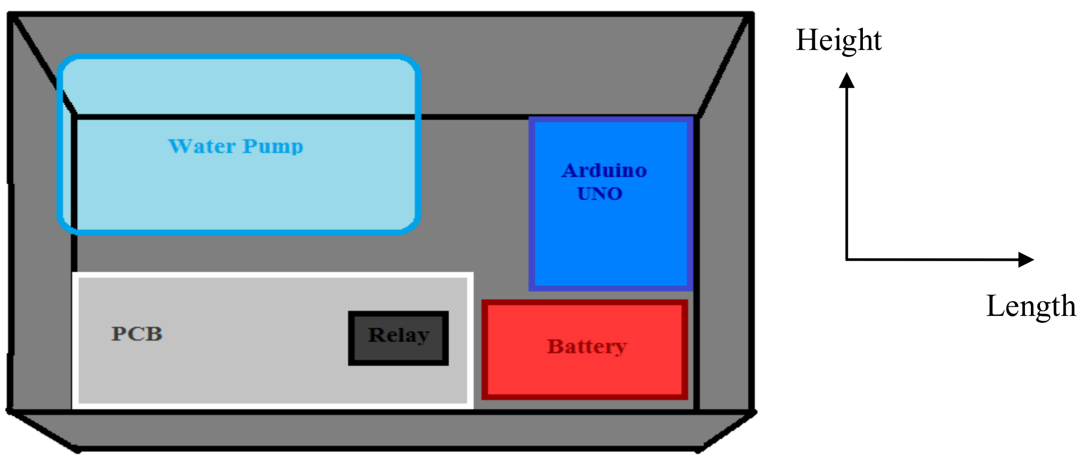

Figure 10: Interior component Layout of controller packaging

Furthermore, the next significant design consideration is the tall of package case. In these components, water pump have a maximum height. Therefore, the height of the case should be at least 60 mm.

3.4 Safety and Environmental Protection Consideration

The working voltage of electrical components such as moisture sensor, water level sensor and LED indicator are under 5V, so parts can be considered as hazardous-free. However, the working voltage of water pump is 120V alternating current. Thus, insulating material, the ABS plastic is chosen for packaging case to avoid possible electrical hazard. In addition, the system has uni-body structure including watering components, so the case must be waterproof. The ABS plastic satisfies both insulation and waterproof characteristics.

In order to accommodate the operating condition of moisture sensor, the watering system must be capable of operating in the environment of 10 to 40°C with 0 to 98% non-condensing humidity.

4.0 Shade System

In the following sections, details into the hardware and software design will be discussed, which includes the mechanical, electrical, and software design, as well as any concerns regarding safety and environmental protection.

The Sunlight Protection System will consist of light intensity sensor(s) that communicate with the microcontroller via wired connection. Based on the data taken by the photo resistor, the electric shade will be commanded to either stay stationary, or to move to block sunlight.

4.1 Mechanical Design

This particular section will provide insights and overview into the mechanical component of the SmartPlant™ Shade system. For this portion of the product, the hardware components consist of DC motor, lightweight wires, springs, and a lightweight fabric designed to block sunlight.

The majority of the Shade System will be mounted directly above the plant system, as depicted in Figure 11. First of all, there will be four to six supporting columns, depending on the stress that the system will exert on the DC motor. That is to say, the initial design will be done with four supporting columns, but if we find that the stress is too great, our fall-back plan is to change to a six column method, in order to separate the load to two parts instead of one.

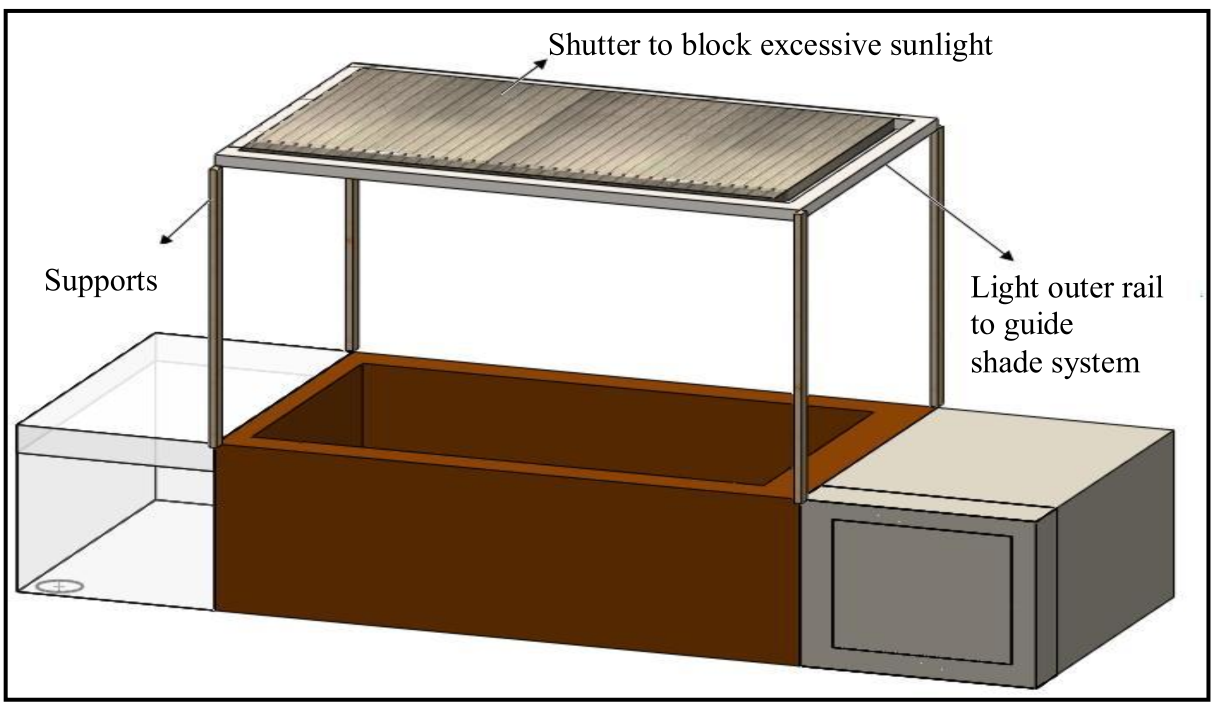

Figure 11: Shading System Setup

Next, the one part of the shade will be attached to a DC motor via wires, and the motor will work to pull the shade towards the other side (the motor side). This command is given by the microcontroller, which relays any information via wired transmissions. The following table is the spec of the tubular motor:

Stall Current	3000 mA
Speed	28 rpm
Operating Voltage	12V DC
Operating Current	1500 mA
Measured Lift Capacity	5.5 kg lift force
Maximum Rotation	Unlimited
Overall Length	280 mm
Length in Tube	267 mm
Drive Diameter	35 mm
Standby Current	0 mA
Torque	1.5 N*m
Weight	272 g

Table 2: Datasheet of Tubular Motor

These figures are within acceptable parameters to our project. The rotational speed is 28 rpm, so this is approximately half a rotation per second. The speed of the rotation, and therefore the rate at which the shade will be pulled across the track won't be too fast. According to the datasheet, the measured lift capacity is 5.5 kg, which should theoretically be powerful enough to pull the shade. This will hold to be true as long as the shade is not too heavy, and the friction be kept to a minimum by proper design methods.

Theoretically, the 28 cm motor length should be sufficient in covering the span side of the flower pot. This is especially important, as the ideal situation is to have the shade cover up as much area as possible in the flower area. Figure 12 explicitly shows the internal composition of the tubular motor, which gives simple insights into the internal workings of this particular gadget.

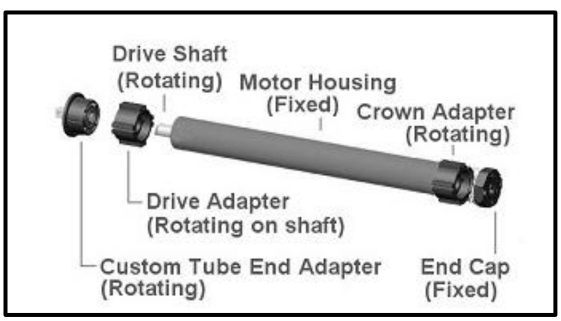

Figure 12: Tubular Motor [11]

For the prototype, H-Bridges are used to control the motors. The H-Bridges require very low power consumption at the input, and provides as much as 1A for each joint when it's connected to an external power source. The H-Bridges are also called SN754410 Quad Half High Current H-Bridge, which makes use of TTL (transistor-transistor logic) 5V logic levels for bidirectional drive. The following is the connection of the motor driver:

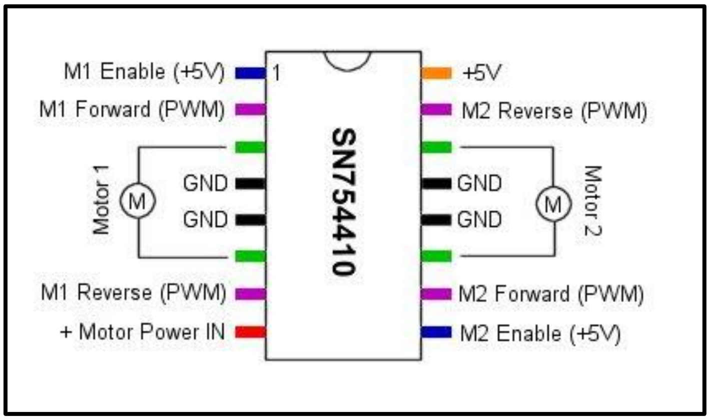

Figure 13: SN754410 H-Bridge Circuit [12]

With the proper data inputs, each pair of drivers of H-Bridge can form a bridge reversible drive suitable for motor applications as seen in table 3:

Inputs			Output
Motor1 Forward	Motor1 Reverse	EN	Motor 1 output (Both Green lines)
H	L	H	5V (Forward)
L	H	H	-5V (Backward)
X	X	L	Z (0V)
H/L	H/L	H	Z (0V)

Table 3: Inputs and Outputs of SN754410

4.2 Electrical Design

One of the main components of the design process is the Photo-resistor, or more formally, the light dependent resistor (LDR). The LDR is generally a simple light sensor that will alter its resistance based on the intensity light level. Figure 14 shows the wiring that is required to be done in order for the photo resistor to communicate with Arduino.

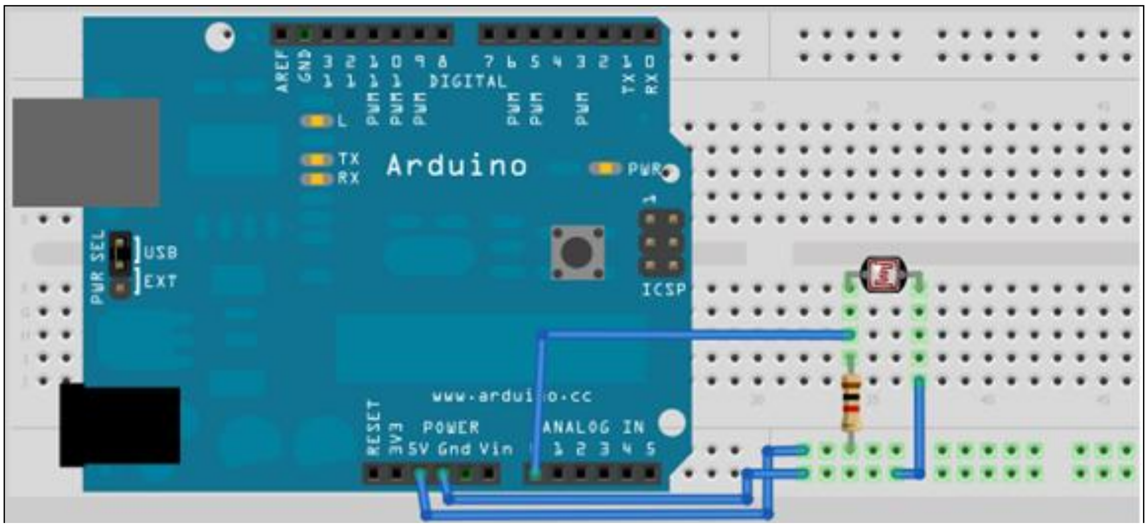

Figure 14: Arduino Light Sensor on Breadboard [13]

As for the electrical design, there are several things that are taken into consideration:

The sunlight intensity sensor will measure the concentration of light at a particular part of the plant pot, and will continuously feed this data to the Arduino microcontroller. The Arduino microcontroller will determine if the threshold for light intensity has been reached or not. If so, this will activate the shade system to move into position in order to block sunlight. If not, it will go back to state 1. The system will be able to send the light intensity data to a mobile device via Bluetooth. This data will be shown as an easy-to-understand range, from mild to strong sunlight levels.

As stated in the previous list, the shade system has two main components: sensor and motor/shade. These two communicates with each other via wired transmissions, and conversely, the microcontroller can also communicate with a mobile device wirelessly. This will give the user more options, and also gives more information on the status of the flower pot.

Asides from these connections, the individual electrical connections are also crucial. For the parts, there is the Arduino, light sensor, breadboard, and circuit components. There are also the jumper wires and individual wirings that are needed to connect each parts to each other.

4.3 Safety Consideration

There are not a lot of safety concerns, except for the safety with motors. The rpm of the motor will need to be taken into serious consideration, as the rate should not be so fast as to cause possible injury to the user. Another thing to consider is the electrical safety of the motor. Certain safeguards could be kept in place to prevent injury by malfunction. However, as there is no definite motor in possession, it is inconceivably difficult to predict the types of safety to take into account.

As for the environmental consideration, there will always be a risk of using a battery as a power source. Over time, alkaline battery might run the risk of leakage due to prolonged usage. This will cause detrimental impacts on the environment, as well as to the plants itself.

Therefore, there is always a concern for safety when it comes to any electronic and mechanical system, and it is the utmost priority to find these possible flaws, and to minimize the risks as much as possible.

5.0 Mobile App User Interface

The Android application will be used as the user interface for SmartPlant™. It is the most familiar user interface in these days and it will be a critical factor for the success of SmartPlant™ in the market. The user interface using smartphone applications was chosen due to the fact that smartphone became core electric device in people's daily lives. On August 2013, Google Canada has reported that the smartphone penetration has reached 56 % of Canadian population [14]. In addition, the Android mobile platform has far more hardware than its major competitor, the iOS. According to a recent statistic result, the Android market share has risen to 79.5 % of worldwide market during the second quarter of year 2013 which directly tells us the reason why the Android platform should be chosen for our SmartPlant™ [15].

One of the most up-to-date Android devices, Samsung Galaxy Note 3 with most recent Android OS version 4.3 is used to test the prototype SmartPlant™ Android app. This is done because, by testing the app on the most recent OS and the latest device, the developer can ensure the compatibility of the apps with other recent and future Android devices and subsequent OS updates.

5.1 Mobile App Development

For the Android app development the MIT App Inventor will be used. This open-source web-based application is chosen over other app development software due to its user-friendly features. The MIT App Inventor does not require any high level software language backgrounds and even the complicated lines of code are not needed. It introduces simple and intuitive graphical user interface that allows users to be familiar with app developments. The App Inventor is basically composed of two parts, Designer and Blocks Editor. Figure 15 shows the layout of the App Inventor Designer when designing the home screen of SmartPlant™ Android app. The strength of the App Inventor comes from the Designer. The user can design the apps with straightforward drag-and-drops to place components (images, buttons, and etc.) wherever it is desired.

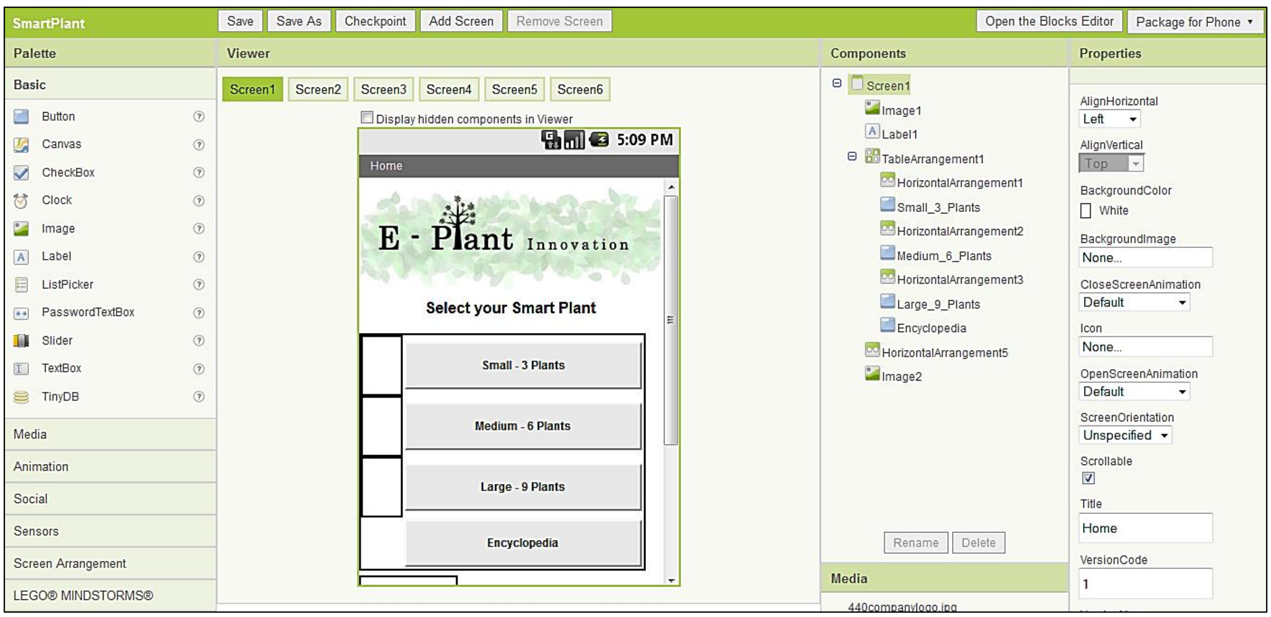

Figure 15: MIT App Inventor Designer Interface

Figure 16 shows the App Inventor Blocks Editor. Here again, the programming is done with block drag-and-drops and the behaviors are specified for the components previously defined in the Designer. For example, blocks in Figure 16 represent the behaviors of opening the other screens named “Screen2” or “Screen3” when the buttons “Small_3_Plants” or “Encyclopedia” gets clicked respectively.

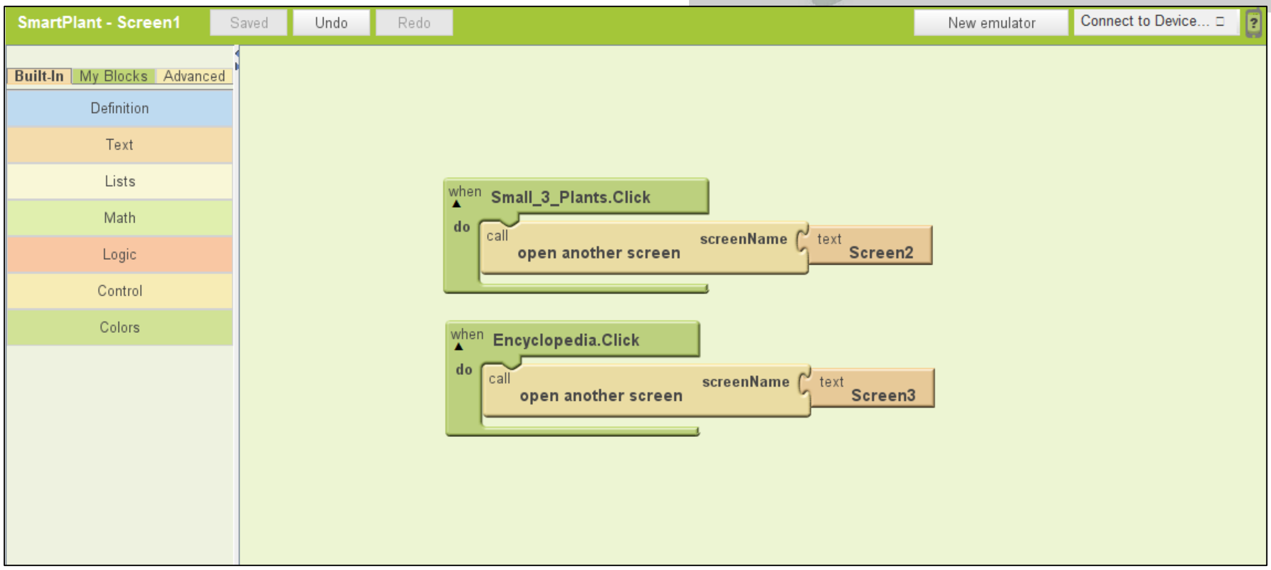

Figure 16: MIT App Inventor Blocks Editor Interface

5.2 Mobile App Interface Design

The future marketable product of SmartPlant™ will be available in three different models: Small, Medium and Large respectively having single, double and triple shading/watering setup. The prototype of the SmartPlant™ will be model Small which has a real capacity up to three plants so the Android app will be developed targeting this model.

When the Android app is turned on, it displays the home screen where user can either select the model he/she owns or read the encyclopedia. Figure 17 shows the Android app interface transition when the user chooses model Small and decides to plant cucumber. The red arrows are processed when the user clicks a button in the red-dotted circles and blue arrows when the user clicks a Back button in the blue-dotted circles. Based on online plant guide websites [16] and [17], plants with same light intensity and watering requirements are registered and labeled as a group. Notice from second screen, along with cucumber, the eggplant and okra are also labeled in the same button. This indicates that the other two plants can also be raised with cucumbers as they have same light intensity and watering requirements. After user confirms the plants what he/she wants to plant, a simple light/watering requirements are displayed for user's interest and the screen also shows the alert message for user to turn on the Bluetooth. When the Bluetooth connection is set between the mobile device (Android app) and the microcontroller of the system, by clicking the "Connect to SmartPlant™" button, the app sends data to the microcontroller to select corresponding "profile". Among the two final screens with "Success" or "Bluetooth communication error" messages, one will show up if the data is successfully transferred to the microcontroller and the other will be displayed in opposite case. Notice the screen with error message is omitted in the figure.

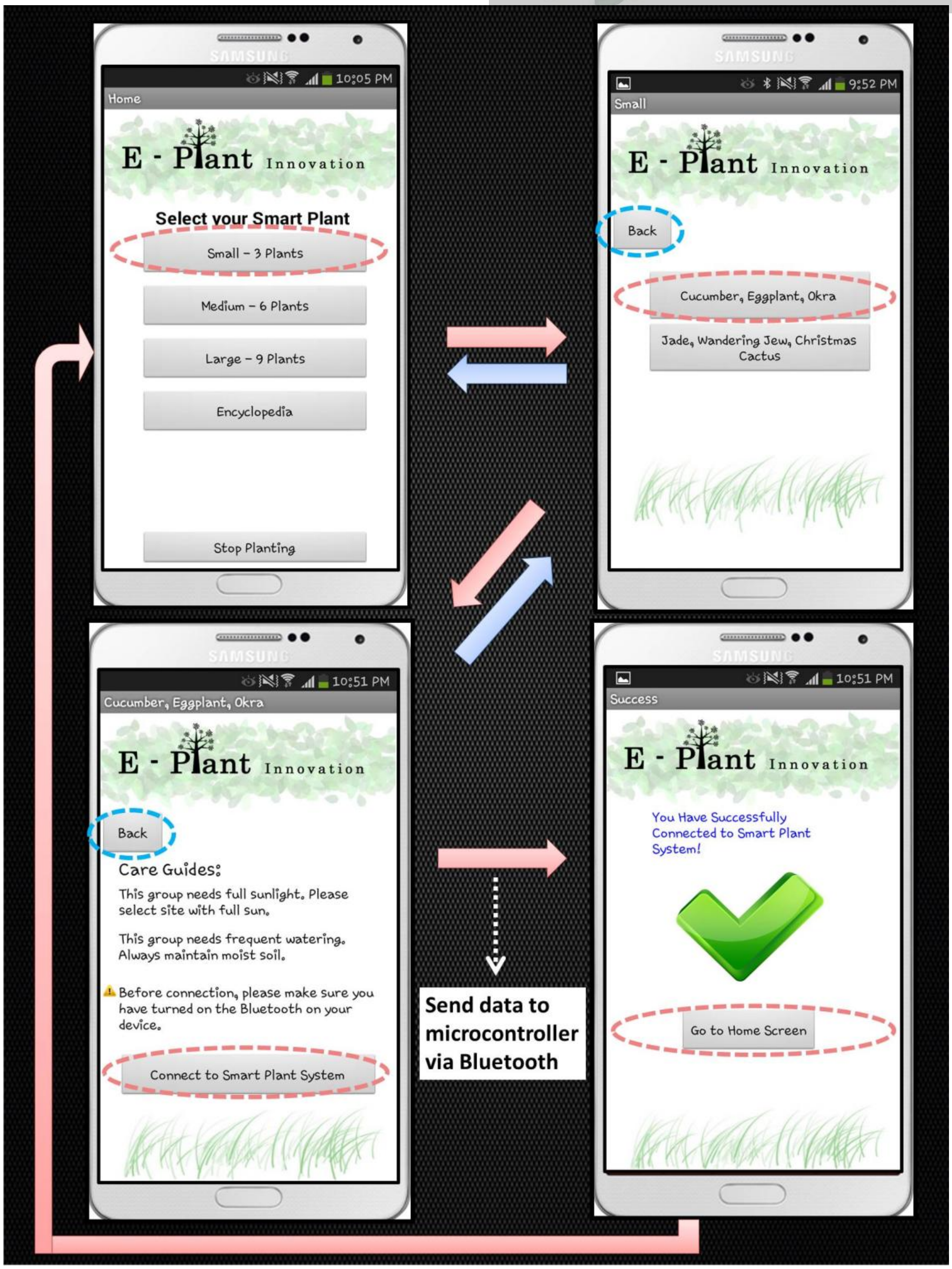

Figure 17: Android App Screen Transition when user selects one of the Models

The flow chart of Android app Bluetooth communication with Arduino microcontroller is shown in Figure 18. The Android App will be basically used to select "profile" at the microcontroller. From the second screen of Figure 17, the group including cucumber will be set as a profile1 while the group with jade will be set as a profile2. The Arduino IDE will be programmed to have three major conditions separated by “profiles”. A profile0 is a system termination profile which is processed when the user clicks a button “Stop Planting”. Thus at this profile the microcontroller gets deactivated. The profile1 is processed at the microcontroller when the user chooses to plant the cucumber group from the Android app. In this profile, water pump and electric shade start to work. The water pump and electric shade will be controlled based on the comparison between the sensors’ measurements and the profile’s own moisture and light intensity threshold values. On the other hand, if the user chooses the jade group to plant, the profile2 is processed and the rest works in the same way as profile1.

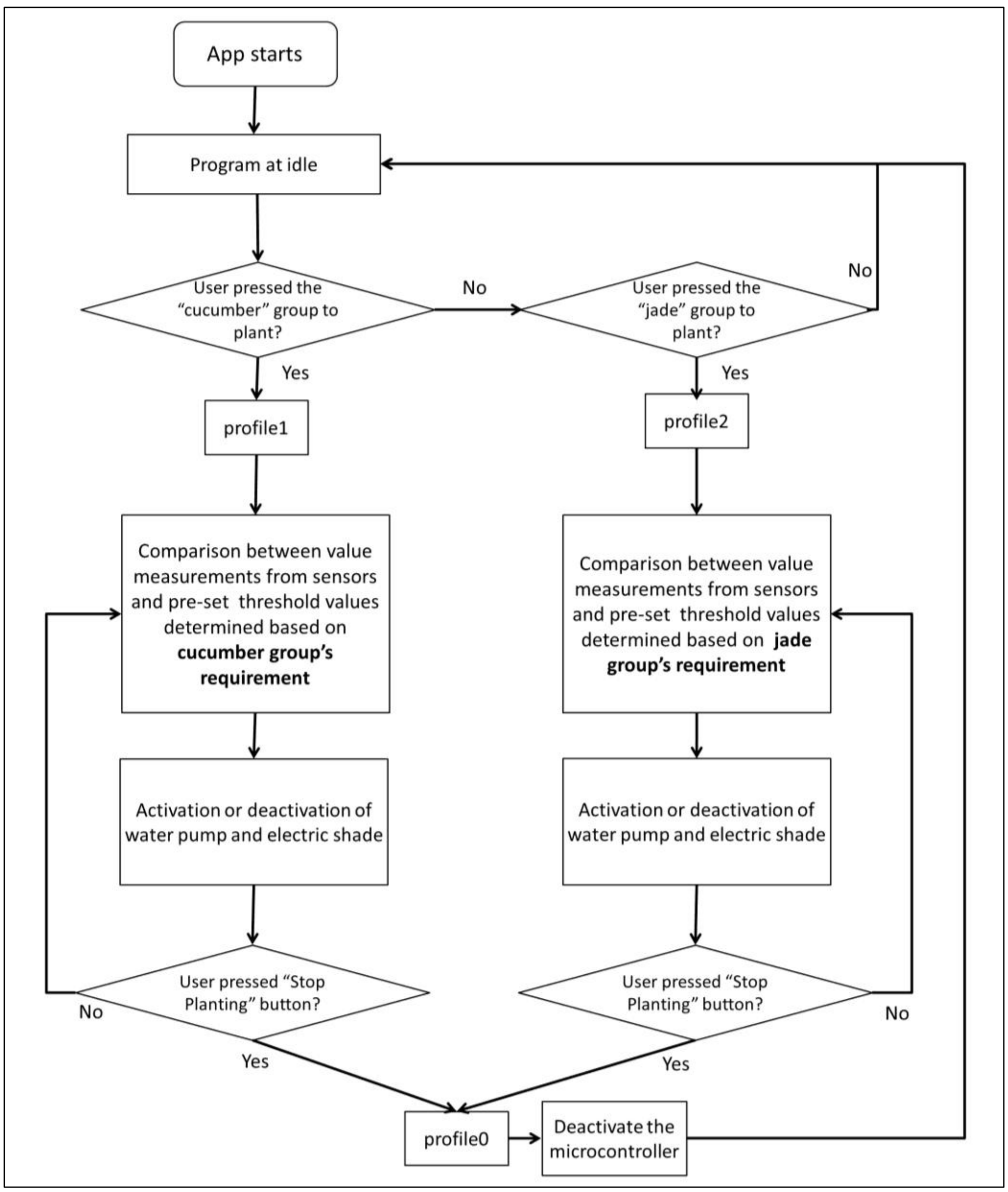

Figure 18: Flow Chart of Profile Selection by Android App

Figure 19 shows Android app screen transition for the case when the user chooses to view encyclopedia and learns about cucumber.

Figure 19: Android App Screen Transition when User Selects Encyclopedia

6.0 Power Supply

The SmartPlant™ has four components which need power supply: Ardunio microcontroller (12V DC), Water pump (120V AC), shading motor (12V DC) and LED water level sensor. It is possible to use AA batteries as power supply by having two 6V battery packs in series which each pack contain 4 AA batteries. Figure 20 shows that connecting batteries in series add the voltage of the two batteries but keep the same amperage rating. For example these two 12V batteries joined in series which produce 24V but still have a constant capacity.

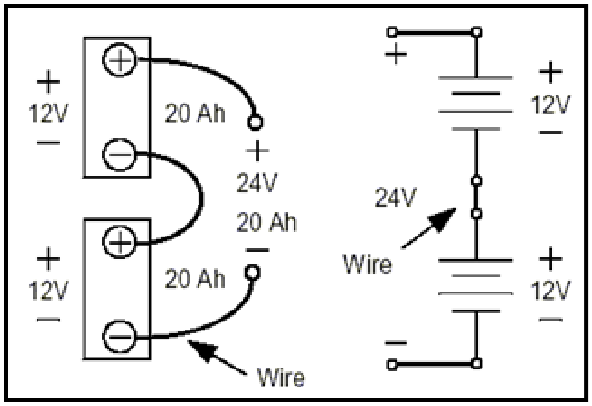

Figure 20: Series Connection: Voltages Add, Capacity is Constant [18]

Also instead of battery we can use 12V rechargeable lithium battery which is easily rechargeable with for the user and it will last longer than AA batteries. “The internal battery is designed with protection function to bear with overload, overvoltage and short-circuit. When the battery is not used properly, the internal circuitry will automatically cut off power supply, resulting in no voltage output, then to charge it again to resume the power supply” [19]. Figure 21 shows the rechargeable Li-ion battery.

Figure 21: Rechargeable Lithium Battery [20]

7.0 System Test Plan

Plants change very fast depending on a variety of factors such as temperature, location, season, and weather. Several tests need to be performed to ensure the product is both durable and reliable. To achieve this, we need to do separate testing for individual tasks and at the end combine all of them together.

The first section of testing is the watering system. The purpose of this test is to see if the watering system is responding to a different level of soil moisture. We can test the sensitivity of moisture sensor by placing the sensor into dry soil and measure the resistance value and then insert it in wet soil and measure it again. Software testing is an important part of this step, which is required to get data from the sensor and transfer it to Arduino microcontroller. The microcontroller should respond to data by activating or deactivating the water pump. We can achieve this by writing a simple program to make sure that the sensor and microcontroller are communicating such as turning on the LED light. The one side of a resistor goes to Arduino pin 5V and the other side to moisture sensor which is connected to pin A0. Also the moisture sensor is connected to pin GND. The LED needs to be connected to Arduino as well. Insert the probes into the soil, if plant needs water the LED will light up [21].

The second section is testing the electric shade which will be open when there is excessive sunlight. There are two test plans for shading system: hardware and software. The mechanical part is the testing the motor to accurately open and close the shade. Also we will need to test our design to make sure opening the shade will not damage the plant. The programming part is to ensure that the photoresistor sensor is operating and sending data to the microcontroller so the motor will be started. A photoresistor sensor and 1K resistor are hooked up in series. The Arduino pin +5V goes to the photoresistor sensor and pin GND goes to the 1K resistor. Light makes a photoresistor sensor have a low resistance so by blinding the sensor the LED should go off and more light will turn on the LED [22].

The Android application is another section of the test plan. The android application has to communicate with the microcontroller. Therefore, we will write a simple program to send different numbers to Arduino which lead to turning on and off the light. Later we will develop the program to send the plant type to Arduino.

Software bugs will almost always exist in any software module so software testing will be performed for any correctness, performance and reliability testing. We will see if the microcontroller correctly and successfully responds to data. This will be performed by adding components one by one to the circuit, when the previous bugs have been eliminated. By debugging the programming we can ensure that the code is running correctly without any miscalculation.

Finally, we have to combine these three sections together to ensure that all of them work simultaneously. The battery life would impact the system effectiveness. Therefore, in this section we have to test the power supply to see how much battery we need to activate watering system and open/close the shade.

7.1 Typical usage scenario

From SmartPlant™ application user chooses what type of flower pot have; depend on the flower pot size the application suggest to plant specific set of vegetables and fruits. Depends on what type of plant user is planting the watering and shading system will adjust automatically. The application will notify user weekly to refill the water tank or recharge the battery

Unit test 1: Moisture sensor

User input: User inserts the sensors vertically inside the soil.

Conditions: User turns on the power and chooses the plant type from SmartPlant™ application in order to communicate with micro-controller.

Expected observations: If soil is dry the water pump will be activated until the moisture level reaches to the pre-set moisture value.

Unit test 2: water tank level indicator

User input: User fills up the water tank.

Conditions: The water tank has two LED: red and green.

Expected observations: If the water level in water tank is full or half the green light is on. When the water level reaches 1/3 of the water tank the red light should turn on.

Unit test 3: Shading system

User input: User adjusts the height of the shade and turn on the power.

Conditions: The sensor will send the signals to analog input of the microcontroller and the microcontroller will compare the input with its pre-defined thresholds. If the input is out of appropriate range, the microcontroller will turn on or off the switch to fold or unfold the electric shade.

Expected observations: The motor should respond to microcontroller and rotate forward and backward in order to close and open the shade.

Unit test 4: SmartPlant™ Android application

User input: User has to download the SmartPlant™ application and select a certain plant that user wants to grow. Also should turn on the cell phone Bluetooth.

Conditions: Once the plant decision is made, the cellphone application will send the moisture and light intensity thresholds to the corresponding microcontrollers. Thereby microcontrollers control the water pump and electric shade.

Expected observations: By selecting different plant type the shading and watering system will change accordingly.

8.0 Conclusion

The design specification provides a detailed description and specification of the SmartPlant™ system. This document will be used as a guideline for adherence and compliance to the original function specifications and design requirements. The document also discusses the test plan which will significantly increase the reliability and functionality of the SmartPlant™ device.

9.0 References

[1] H. Godman. "Backyard Gardening", Internet: http://www.health.harvard.edu/blog/backyard-gardening-grow-your-own-food-improve-your-health-201206294984, June 29, 2012, [Nov. 3, 2013].

[2] R. Muñoz-Carpena "Field Devices for Monitoring Soil Water Content", Internet: https://www.bae.ncsu.edu/topic/go_irrigation/docs/Field-devices-monitoring-.pdf, [Nov. 3, 2013].

[3] Lisemen, "Garduino: garding+Aduino" Internet: http://www.instructables.com/id/Garduino-Gardening-Arduino/, Dec. 15, 2008, [Oct. 28, 2013]

[4] Arduino Uno, Internet: http://arduino.cc/en/Main/arduinoBoardUno, [Oct. 28, 2013]

[5] Brent Q. Mecham "A practical guide to using soil moisture sensors to control landscape irrigation" Internet: https://watergreat.com/reference/Practical_overview_2010.pdf, [Oct. 28, 2013]

[6] "Soil Hygrometer Detection Module Soil Moisture Sensor For Arduino Smart car" Internet: http://www.ebay.co.uk/itm/Soil-Hygrometer-Detection-Module-Soil-Moisture-Sensor-For-arduino-Smartcar/230947908317?hash=item35c5916add&pt=UK_BOI_ Electrical_ Components_Supplies_ET, [Oct. 28, 2013]

[7] YL-69 Moisture Sensor Schematic, Internet: http://m2.img.dxcdn.com/CDDriver/sku.200142.jpg, April 1, 2013, [Oct. 28, 2013]

[8] Arduino Uno, Internet: http://arduino.cc/en/Reference/AnalogPins, [Oct. 28, 2013]

[9] "Tom Aquatics Aqua-Lifter Dosing Pump Information" Internet: http://www.marinedepot.com/Tom_Aquatics_Aqua_Lifter_Dosing_Pump_Accessorie_for_Aquarium_Overflow_Boxes-Tom_Aquatics-OE1137-FIOFOA-vi.html#Manuals, [Oct. 28, 2013]

[10] Water Level Sensor Picture, Internet: http://www.emartee.com/product/42240/High%20Sensitivity%20Water%20Sensor%20%20Arduino%20Compatible, [Oct. 28, 2013]

[11] DC Tubular Motor, Internet: http://rollertrol.com/store/en/12v-tubular/14-projector-screen-motor.html, [Oct. 28, 2013]

[12] H-Bridge Motor Driver 1A - SN754410, Internet: http://www.hobbytronics.co.uk/h-bridge-driver-sn754410, Nov. 15, 1995, [Nov. 2, 2013]

[13] “Arduino Light Sensor Circuit Tutorial” Internet: http://browse-tutorials.com/tutorial/arduino-light-sensor-circuit-tutorial, Dec. 10, 2011, [Nov. 2, 2013]

[14] CWTA (Canadian Wireless Telecommunications Association). “Google Canada: Smartphone Penetration Reaches 56% of Canadians” Internet: http://cwta.ca/blog/2013/08/02/google-canada-smartphone-penetration-reaches-56-of-canadians/, Aug. 2, 2013, [Nov. 2, 2013].

[15] D. Reisinger. “Android nans record 80 percent market share in Q2” Internet: http://news.cnet.com/8301-1035_3-57596548-94/android-nabs-record-80-percent-market-share-in-q2/, Aug. 1, 2013, [Nov. 2, 2013].

[16] Almanac. “Information About Plants: Vegetables, Herbs, & Fruit Guides” Internet: http://www.almanac.com/content/information-about-plants-vegetables-herbs-fruit-guides, [Nov. 2, 2013]

[17] Bonnieplants. “How To Grow” Internet: http://bonnieplants.com/, [Nov. 2, 2013]

[18] “Connecting Batteries & Chargers” Internet: http://batterytender.com/includes/languages/english/resources/Connecting_Batteries_and_Chargers_in_Series_and_Parallel.pdf ,Apr. 9, 2002, [Nov. 5,2013]

[19] “Battery rechargeable for golf carts 12v 15000mah” Internet: http://www.alphapowerbank.com/sell-battery_rechargeable_for_golf_carts_12v_15000mah-34392.html, [Nov. 9, 2013]

[20] “Portable DC 12V 4500mAh Rechargeable Li-ion Battery”, Internet: http://www.amazon.com/Portable-4500mAh-Rechargeable-Battery-Wireless/dp/B008UKKXRC, [Nov. 12, 2013]

[21] P. Marian “Simple Soil Moisture Sensor” Internet: http://www.electroschematics.com/6519/, [Nov. 7, 2013]

[22] “Arduino Photoresistor Lighting Control”, Internet: http://www.practicalmaker.com/arduino/arduino-photoresistor-lighting-control, [Nov 9, 2013]