Dilpreet Bansi, ID: 218803213

EECS 1021 Minor Project

INTRODUCTION:
This minor project consists of a plant watering system, which is programmed to decide when the plant needs to be watered. After I set all the hardware up following the videos provided, I used the program IntelliJ which uses the language Java, to test the moisture sensor, water pump and the OLED screen. After testing, the values were obtained for the moisture of dry and water-saturated soil to ensure the watering system runs effectively and efficiently.

CONTEXT:
Various plants and soils require varying amounts of water to be healthy and to make sure minerals are moving to the correct location. Giving plants and/or soils large amounts of water can cause there to be limited oxygen supply, which can cause it to die down. This system is my take on creating an effective and efficient plant watering system which can be used to provide plants or soils a proper amount of water for it to grow to its fullest. The system uses an moisture sensor to detect the amount of moisture in the soil of a plant, then it prints the message on the OLED display, and then the water is pumped through the water pump and given to the plant if needed.

TECHNICAL REQUIREMENTS/SPECIFICATIONS:
System Requirements:

- Measure the moisture in the soil using the moisture sensor which is placed inside the soil
- The water pump should dispense the water to the plant if the moisture of the plant doesn't reach the given value for when the soil is wet.
- The OLED display should print out the value of the moisture measured and print out a message saying if the plant needs more watering or not.

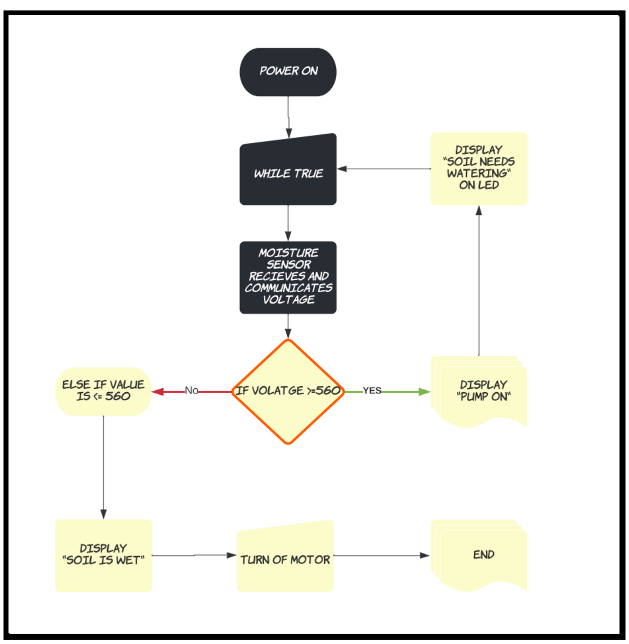

Figure 1: Flowchart of the things the system can do

COMPONENTS LIST:

- Grove Beginner Kit for Arduino (1)
- Moisture Sensor (1)
- Mosfet Switch (1)
- 9 Volt Battery (1)
- Water Pump (1)
- Tube to deliver the solution (1)
- USB cable (1)
- Pot (1 - To hold the soil and the plant)
- Plant and Soil
- Water Reservoir (1 - To hold the water and the water pump)

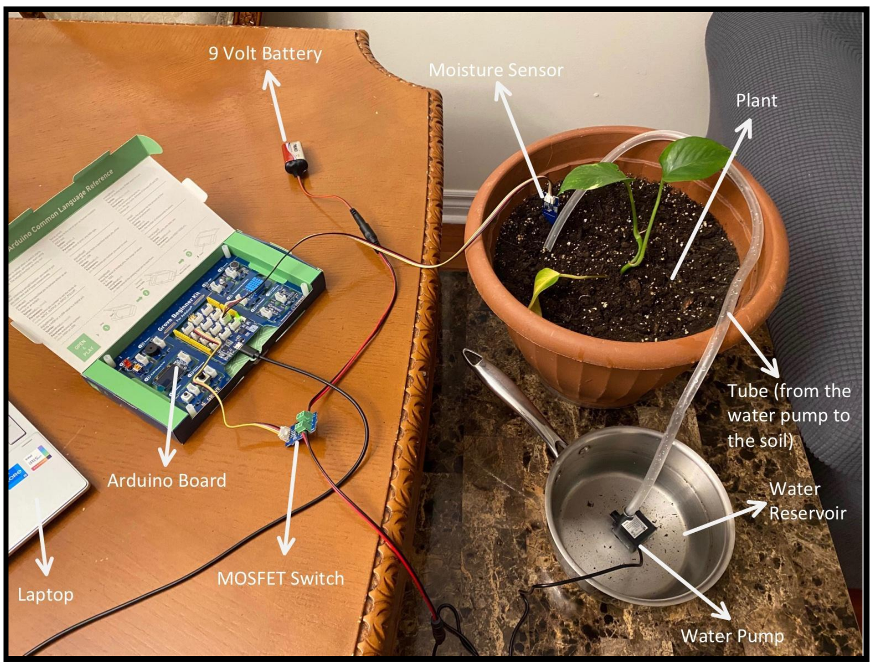

Figure 2 : Labeled diagram of my plant watering system

PROCEDURE:

Firstly, IntelliJ was used as the program of choice, in which both Java and Firmata4j were used to code. I started by initializing all the variables and objects needed to ensure that the system works effectively. Afterwards, I started to work on the Grove board by first getting the moisture sensor and placing that on the Arduino board. The moisture sensor was placed into the “A0” on the Arduino board. Next, the water pump and battery were connected to the mosfet switch to power and connect them together. The mosfet switch was placed into “D2” on the Arduino board. The water reservoir was filled and the water pump was placed inside the reservoir which was positioned slightly under the plant. Using Java and Firmata4j I programmed the system to successfully follow the tasks that it is required to do. The code allowed the plant watering system to be executed and the plant was watered until it was given the most appropriate amount of water.

TEST:
The method which I used to test my system was placing the moisture sensor in water and in air to see if the water is being pumped or not. If it was placed in water, the water pump wasn't supposed to dispense anything. Whereas. if it was placed in air the water pump would dispense the water through the tube.

LEARNING OUTCOMES:
Learning Outcome. 1. Demonstrate the ability to test and debug a given program and reason about its correctness. (GAI 2b)
I tested and debugged all the materials initially to test if they were operating fine. By doing this, I could tell where the problem was exactly later on if something went wrong, and as seen in the video my project functioned perfectly fine.
2. Given a problem specification and a suitable API, build an application that meets the given requirement. (GAI 4b) .I successfully built a working self-automated plant watering system using Firmata4J which meets the requirements of the project.
4. Build an event-driven application that controls sensors and actuators in order to connect events to physical actions. (GAI 4b) This project was event-driven as it depended on the voltage the moisture sensor read. If the moisture sensor read >=560, then the soil was watered but if it was <=560, then the pump stopped.
5. Program common applications from a variety of engineering disciplines using an object-oriented language and solve them on the computer. (GAI 4c). The project is similar to what most software and computer engineers do because of the coding aspect, and this application was created using an object oriented language on a computer.

CONCLUSION:
The plant watering system was successfully created and programmed. The programming ensured that the values measured using the moisture sensor are displayed accurately on the OLED display. Thus, I believe that working on this system was an amazing step as it allowed me to gain knowledge about different applications on the Arduino board using Java and Firmata4j.
The knowledge gained from this project will be beneficial to me moving on as it will allow me to gain confidence in executing difficult projects in the future.