Student Name: Jonah Ottini
Studetn ID: 218945030

INTRODUCTION

This is my report for the EECS1021 Minor Project. The objective of this project is to grow a plant with an automated system using the arduino grove board, and program on java. The program should be automated, and be able to keep the plant sufficiently watered for a long period of time.

CONTEXT

The project is meant on a surface level to be able to keep a plant watered over a long period of time. Although it may seem simple it can be a groundwork for multiple other uses. The basis of the project is a program that can monitor a value over time, and do something when the value passes a certain threshold. This basis can be modified to serve another purpose, like monitoring the temperature of a room, or the cleanliness of water in a tank. The project I created may simply be for watering a plant, but with a few small modifications it can serve an entirely different purpose.

TECHNICAL REQUIREMENTS / SPECIFICATIONS

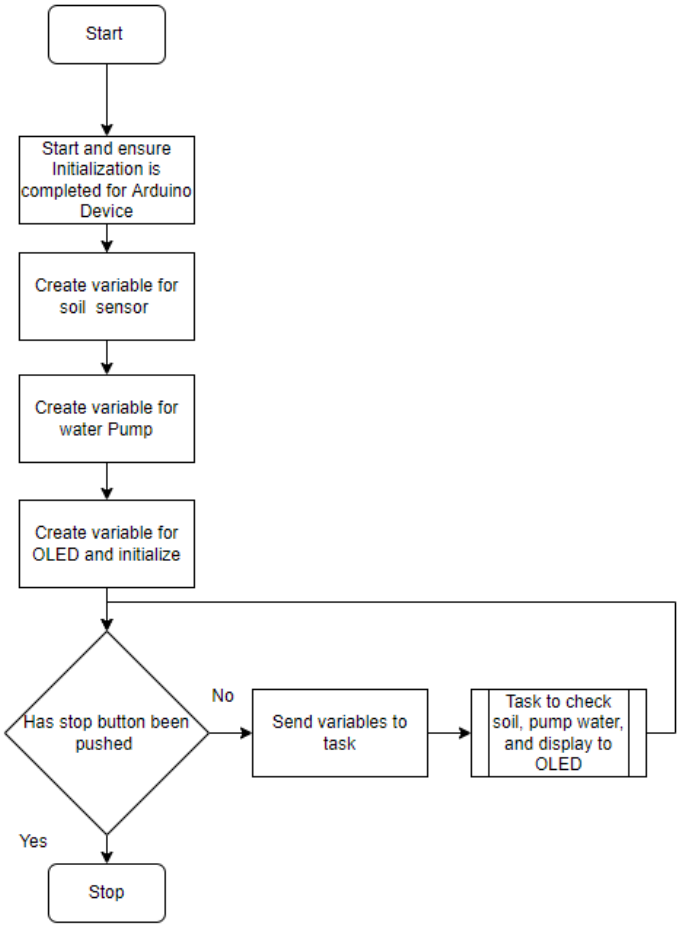
Requirements

- Is able to water plant
- Uses a Java program and Arduino board
- Can sense soil moisture
- Display information on OLED

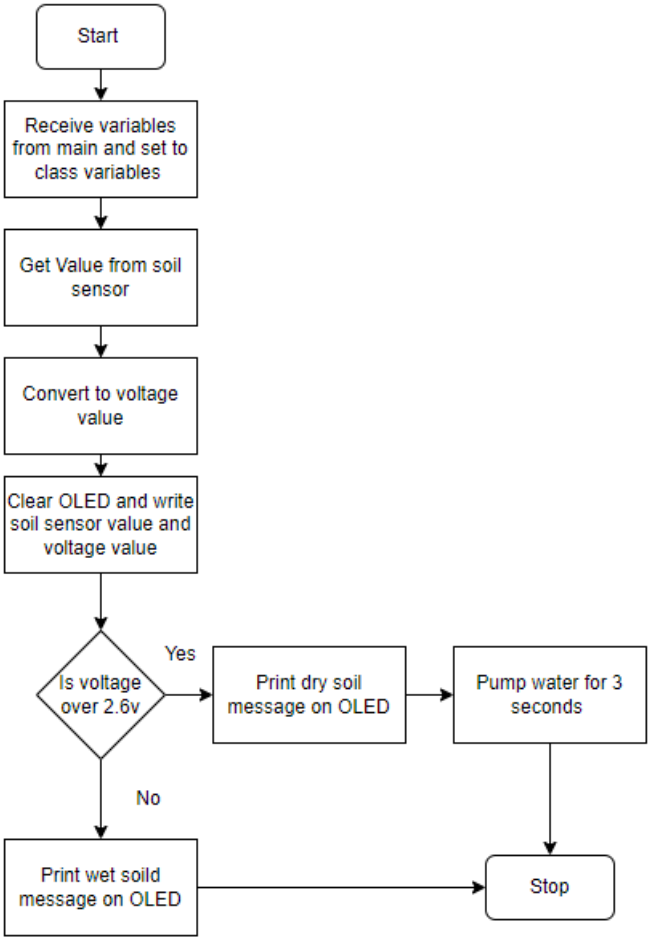
Specifications

- Uses time-based tasking and event driven functionalities
- Plant is watered based on soil moisture
- Is capable of running on its own over long periods of time

Main



Timer Task



COMPONENTS LIST: [as you built the device]

- House Plant + Pot and Soil
- Cup of Water
- Arduino Grove Board
- OLED (on Arduino Grove Board)
- Moisture Sensor
- Water pump (Goes inside Cup of Water)
- 9V battery (To power Water Pump)

PROCEDURE

In order to create my finished project, I first had to create a list of requirements and specifications, and then plan out what the code should look like. I listed the requirements and specifications down, and used a flowchart as the groundwork for my program. When I eventually started to create my program I looked back on the lessons and labs done in class to use the concepts in my own program. In the classes and labs we learned how to initialise an arduino board and create objects for the pins that needed to be used. Next I used a timer task, another concept learned in class, to make the program run at a set time interval, and passed all needed objects into the task. Within the task I created I first used the constructor, and the “this” function to initialise the objects within the task. I then used the objects within the task and applied concepts learned in class (if/else statements, try/catch statements, getValue(), setValue(), getCanvas(), drawString(), Thread.sleep() etc.) to check the soil moisture, water accordingly, and display any important information on the OLED.

TEST

A very simple and easy way to test if the system is working correctly is by creating a very similar program, that includes all of the main components of the final system, except for actually pumping the water. I created a second task that is able to check the soil voltage and display all of the information on the OLED, without pumping water, and risking spilling any while testing. In my main program a can simply uncomment the specific task I want to use, as they are both identical except for the water pumping function. This allows me to quickly switch between the tasks and testing is effortless.

LEARNING OUTCOMES

My project addresses each of the four learning outcomes described in this document.

1. Demonstrate the ability to test and debug a given program and reason about its correctness. (GAI 2b)

- Program was tested thoroughly using java's built in debugger, and the test program I created.

2. Given a problem specification and a suitable API, build an application that meets the given requirement. (GAI 4b)

- This program was created in java using the Firmata4j library

3. Build an event-driven application that controls sensors and actuators in order to connect events to physical actions. (GAI 4b)

- Uses timer tasking and if/else statements to control when the pump outputs water.

4. Program common applications from a variety of engineering disciplines using an object-oriented language and solve them on the computer. (GAI 4c)

- This problem was solved effectively and clearly, using java an object-oriented language.

CONCLUSION

In conclusion, using many of the concepts and techniques taught through the online lessons and labs, gave me all of the tools needed to create and test a self automated plant watering system, using an arduino grove board, java with the firmata4j library.