Student Name: MDFAIAZ
Student ID: 219511500

https://youtu.be/OluwK4rl4bQ

EECS1021 Major Project: Watering Plant using Arduino Groveboard Java with Farmata

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.

Abstract:

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.

Objective of Project with Specifications:

Requirements according to project outline

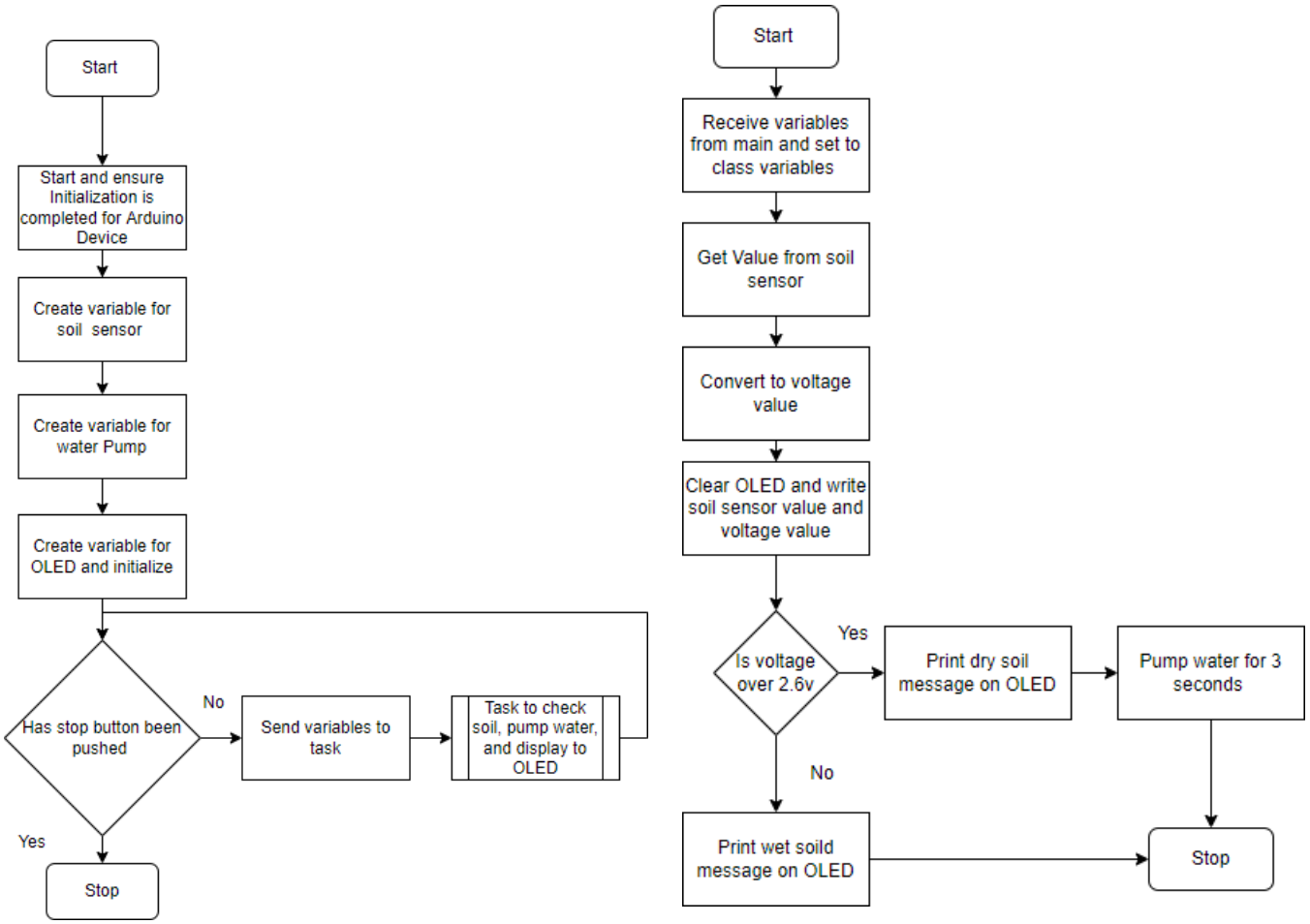
- using Arduino to power motor to water plant
- Using Java program and Arduino Grove board
- measure soil moisture and then stop depending on how wet
- Display information on OLED

Specifications

- Use time-based tasking and event driven functionalities

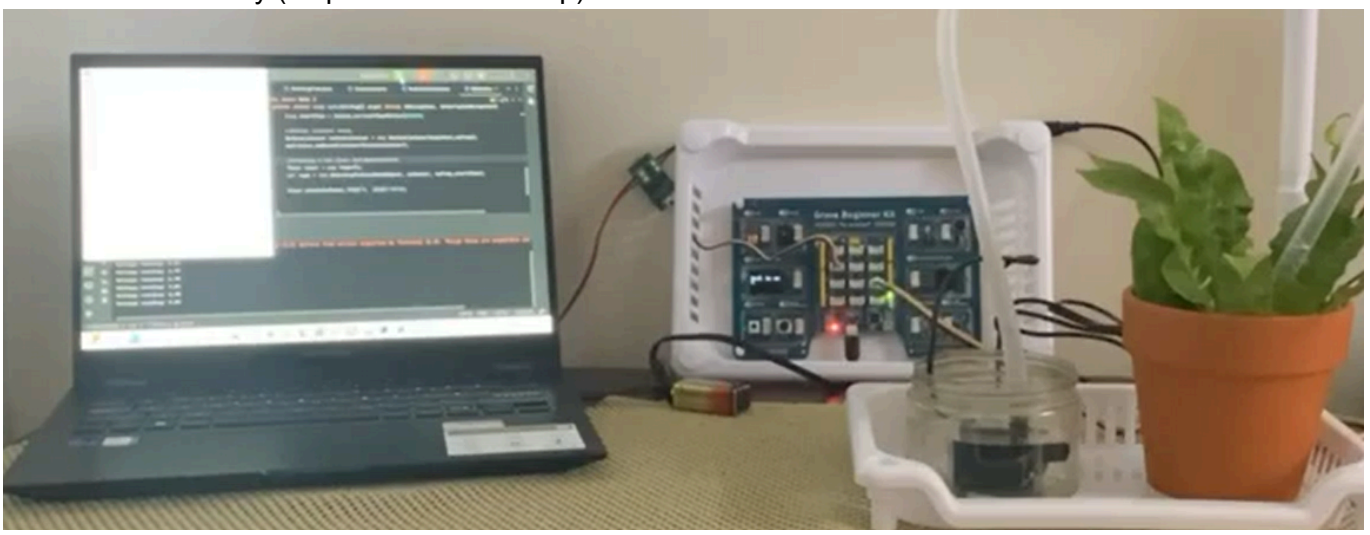
- Plant is watered based on soil moisture
- capable of running on its own over long periods of time

- Schematic of designing code for the Main Timer Task



Materials List:

- 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)



Set up & Procedure:

In order to create the project, I first created a list of requirements and specifications, and then planned the code to be based on the conditions specified by the moisture reading and the outline. By listing the requirements and specifications down, I used a flowchart as the groundwork for my program. I applied my knowledge on how to initialise an arduino board and create objects for the pins that needed to be used. Afterwards I used a timer task concept to make the program run at a set time interval, and passed all needed objects into the task. Within the task I created the constructor, and using “this” keyword to initialise the objects within the task. Then iterating loops to validate conditions using the if/else statements or the try/catch statements to run the loops to receive value by calling getValue(), placing a value by calling setValue(), then displaying values by using getCanvas() and thus this is how I checked the soil moisture, and regulated water pump accordingly, and displayed information on the OLED screen on the grove board.

Writing Test:

A technique to test if the system is working correctly is by creating a file that checks the values coming out of the main program and then feeding it to the functions defined the test class which includes all of the main components of the final system I created another function that checks the soil voltage and displays those information on the OLED screen.

Result:

I was successfully able to create the water pump that's able to pump water based on the soil moisture. This enabled me to allow my home plants to be watered autonomously.In my video you will see how I demonstrated my test functions handset parameters for that reason to correct faults. I tested the program using Intelij's built in debugger. For this Plant watering problem specification I used API's and built an application that meets the given requirement. My program was created solely in java and imported from the Firmata4j library. Implementing an event-driven application which controlled the sensors and actuators for watering; I also used timer tasking and if/else statements to control when the pump outputs water. I programmed common applications from a variety of engineering disciplines using an object-oriented language and solved them on the computer. 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.