EECS 1021 Minor Project Report

Arthur Sabadini Nascimento
Student Number 220007175

April 8, 2023

Introduction

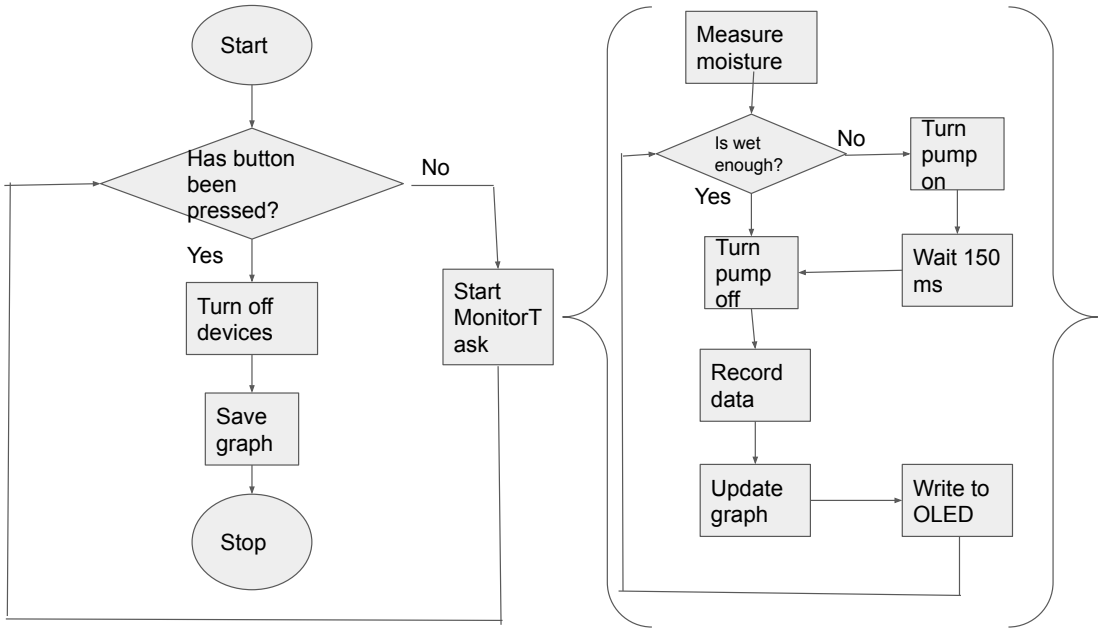
This is a plant watering system, it was developed using Java and the APIs firmata4j and also StdDraw from the standart library from Princeton. It checks weather a plant has enough water, if not the system will water it.

Context

This project creates a plant watering system, utilizing the listed APIs for Java. It is relevant not only because of its main use (water plants), but through working on the project it develops programming skills and computational thinking. The main goal of the project I think is to further develop the listed characteristics, and also to show that it is possible to make something useful, even as a beginner.

Technical Requirements

Flowchart



APIs used

The APIs that were used for this project were firmata4j and StdDraw from Princeton standart CS library. Firmata4j was used to do the communication with the arduino board, reading and writing digital/analog pin values, reading sensors outputs and communicating with the pump to deliver water to the plant.

StdDraw was used to graph the data collected in form of Sample x Voltage. It was used to graph and as a method of visualization of the evolution of the system with time.

Events

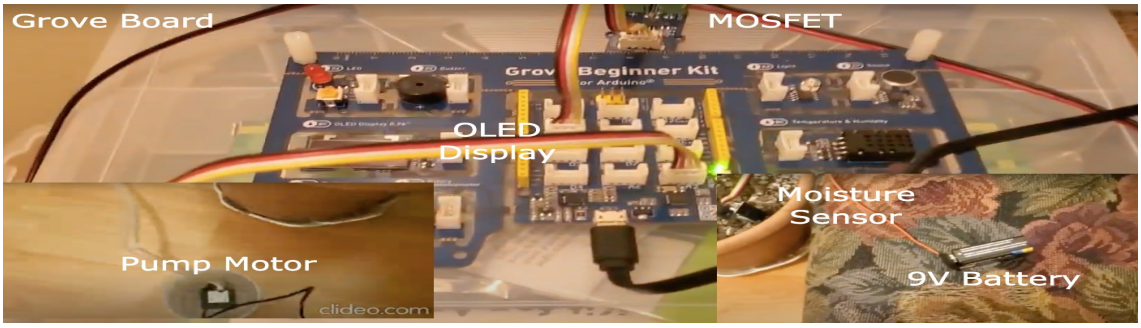
To be able to deal with events, I wrote a TimerTask in Java to check for the moisture of the plant soil every 150 ms, with a delay of 8 ms. This way I would be able to read the moisture reliably, if the moisture level was to high, the program would turn the pump on to water the plant, otherwise the pump would be turned off. Whener this task was executed, we would print the moisture reading in the OLED monitor.

I also wrote a PinEventListener for the button, since I wanted to be able to shut the system off at will, so then whenever the button was pressed, the system would turn off.

HashMap usage

I used a HashMap to store the data collected, (Sample number, Moisture level). And then I used this structured to graph it using the StdDraw library.

Components Used



- Grove Arduino Board
- Pump motor (to pump water to the plant)
- 9V Battery (to power the pump)
- Capacity Moisture Sensor (to measure soil moisture)
- MOSFET (to control pump motor)
- OLED Display 0.96” (to display moisture level readings)

Procedure

A lot of things I based on the experience that I when working on the minor project for EECS 1011, but this time I had to change somethings.

This time, I wanted to create more abstractions; mostly for organization sake. I created and ArduinoUno class, as well as one class for each component, and a TimerTask class to monitor the whole system.

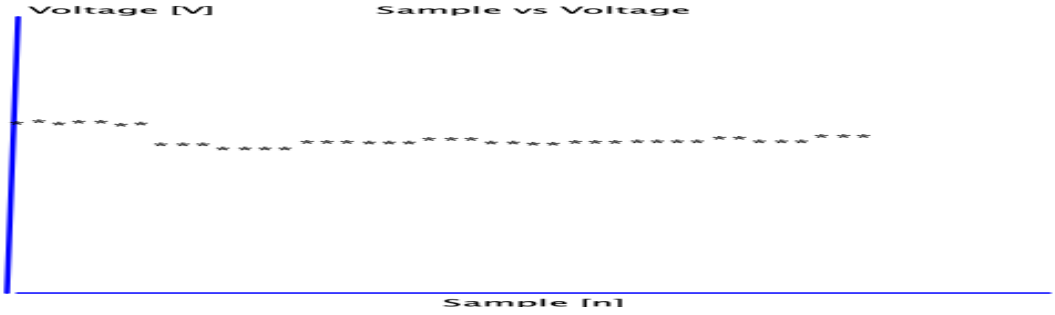
When I started working on the project, I had an idea what I had to do, but I had to organise everything and fix some bugs.

I strated by creating some abstrations, that was my code would be easier to read and better organized, I debugged my code multiple times, to figure out exactly where the bugs were.

Testing

To make sure that my system worked, I used the StdDraw library to graph my data, and I tested to see if the pump would activate when the moisture level was low, and turn off otherwise.

When I encountered problems, I would check my wiring, and also debug by code for possible bugs, or cases that I had not though about.



Learning Outcomes

- CLO 1: I did have to debug my code multiple times, to make sure that the pumping of the water was working properly, as well as other bugs that I encoutered; such as timing issues with the TimerTask class.
- CLO 2: In the project I used firmata4j and StdDraw from Princeton CS library, to construct a time-based application that waters a plant if the soild is to dry.
- CLO 3: I used a HashMap to collect the data, and graphed it using the StdDraw API.
- CLO 4: My application waters the plant whenever the soil is too dry or not wet enough. It measures the soil moisture through a TimerTask object, and can be shut down on the press of the button.
- CLO 5: I used Java + firmata4j + StdDraw to develop an event driven application that waters a plant if the soil is not wet enough. It could be used to monitor the well being of other plants in a massive scale, to perhaps prevent climate change.

Conclusion

In conclusion, I used Java, and APIs such as firmata4j and StdDraw from Princeton CS library to develop an event driven application that waters a plant if the soil moisture if not wet enough. This was done using the components listed above, to be able to measure the moisture level, decide weather it's high or not, if it is we used the MOSFET to turn the pump on and water the plant. The demonstration video of the project can be found here. (https://youtu.be/9GGUeIOOtuc).