Manpreet Dhanjal
219623339
2023-04-10

EECS 1021 Minor Project Report

Introduction: The main objective of this project is to design or build an automated system that can water a plant depending on how damp/moist the soil is, this has to be done with the minimal need of a human. Furthermore, this project requires one to build an Arduino setup that uses sensors, to measure the dampness of the soil, and then further water the plant when needed. Overall the objective is to automate plant watering in an efficient way so that later on we as humans have more time to do other tasks that need tending, rather than having to tend to a plant.

Context: In this Minor Project, an Aundino is connected to ones computer and takes commands from a code. The Arduino will collect moisture data from that plant and depending upon the reading that comes from the moisture sensor, the code will evaluate if the plant needs watering or not. The data that is collected from the sensor will be displayed on a graph that will maintain the record of the plant's moisture history. A project like this is important as the plant can be properly hydrated and maintained when it is required. The setup makes sure that the plant receives the right amount of water at the right time, as the soil will always be damp all times but may require additional watering once the soil starts to dry out. Along with this, it will leave one less thing one has to attend to, which can thus save them time and leave them to do other things they want to do.

Technical Requirments / Specifications

List of things the system should do:

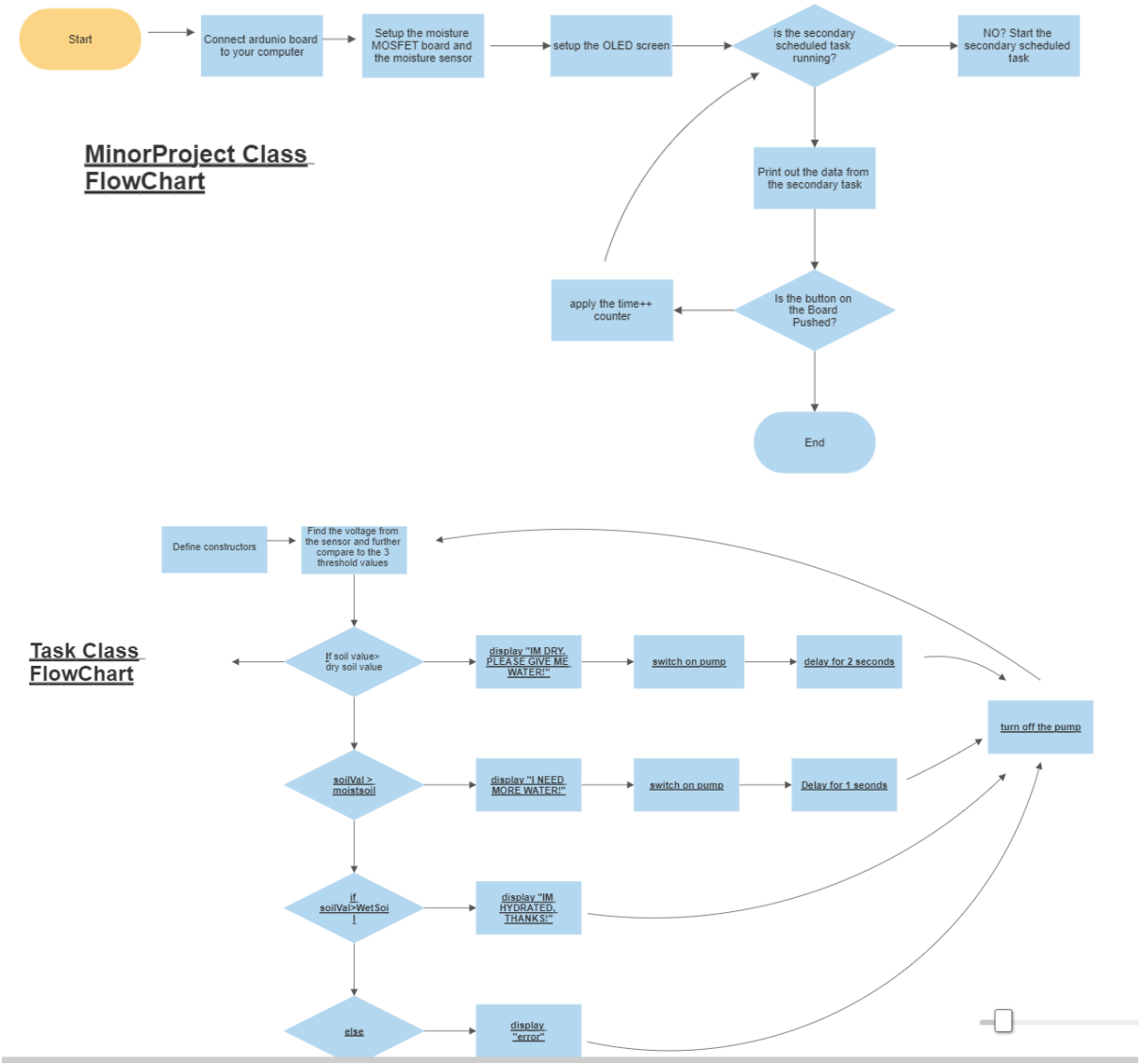
- Water the plant depending on the moisture content
- Store the data values from the moisture sensor in a HashMap, and further use the data to display it on a graph
- Read moisture levels from the plant
- Present data on OLED screen

In order for me to deal with these events, I created a class called “task” which further extends the TimerTask subclass. Furthermore, in the class, i mainly initialized the 3 main moisture level values that the sensor will read for wet values, moist values and dry values and Ii did this by the “mySensor.getValue” function. I then further used a series of if and else if statements to run through the conditionals of the moisture sensor readings. The conditionals consisted of testing to see if the moisture level fell under a certain threshold and then displaying the moisture status on the oled screen, from there i also implemented a try-catch to turn off and on the pump when it was required.

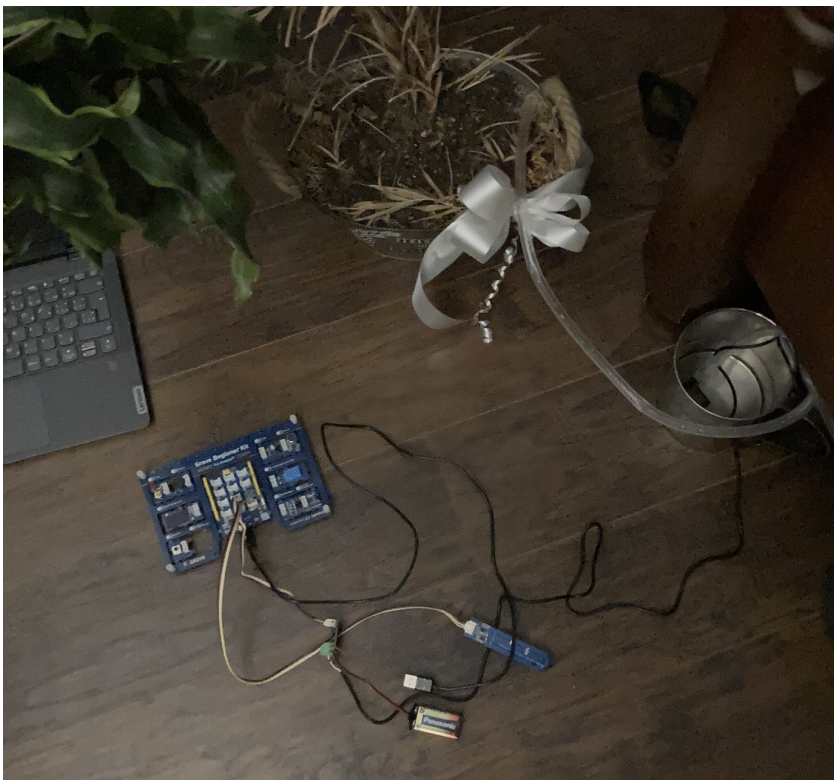
To gather data from the moisture sensor before plotting i useda harshmap. To start off, i initizaized the hashmap then added the sensor data to the hashmap. I then used a for-each statement to run the points on my graph.

Manpreet Dhanjal
219623339
2023-04-10

Flowchart:



Picture of Setup:



Components List:

- Grove Ardunio Kit
- MOSFET Board
- Water Pump
- 9V Battery
- Soil Moisture Sensor
- OLED Screen Display
- Plant
- Water Container

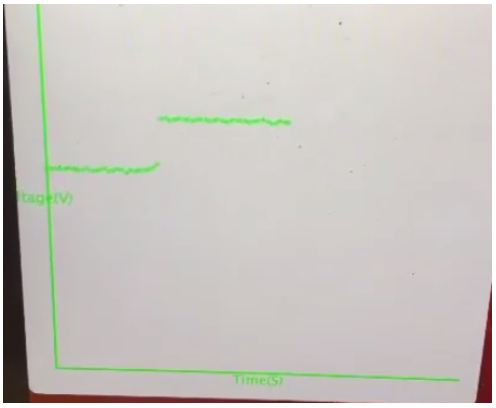
Manpreet Dhanjal
219623339
2023-04-10

Procedure

To Start off this Minor Project I followed the video posted my professor James Andrew Smith called “Wiring up the MOSFET, Pump & Soil Moisture SensorSeeed Studio Grove Beginner Kit for Arduino”, this was the same video that i used from last semester build my setup however i followed it again to ensure a proper connection was made. From there I used the code at the end of the video in the Ardunio IDE to run and test to see if my moisture sensor and pump was working. From there I started building the code, I had an idea of how to build the code as I was told it can be build using the knowledge from the previous labs. So to kickstart my code I used the previous knowledge of how to initialize variables, and I setup variables for my pump, my mosfet board and finally my port that the ardunio was connected to. From there i started the mosfet board that was connected to the pump, the Moisture Sensor, and the OLED Screen. After this I was having some trouble as I didn’t know where to proceed to next in my code, my friend recommended the I watch the “Arduino + Firmata & Java + StdDraw Graphing” video that Professor Smith posted, and from here I was able to kick start the next part of my code where i setup the graph portion of my Minor Project. After creating the Graph portion I moved on, and created a new class called “task” where i stored a series of if-else statements for the values that the moisture sensor was giving me. I didnt know what threshold to put for the values so i did a test run until i found a comfortable range where the pump would turn on when dry and turn off when wet. For the storage of my values from the moisture sensor, I used a hashmap. My first plan was to use an arraylist as that is what came up from the previous labs we did but implementing the array list into my code was difficult for me, so my friend recommended to use a hashmap and that it would be rather simple to use. After adding the hashmap I attempted at running my code and found that the code worked great however, when the graph was plotting the points, it was plotting a scatter plot rather than something in a linear fashion that would go up and down. To solve this i tried many things but what seemed to work was changing the pins, from what i had previously to a different set of pins. Once I made this change the code was running flawlessly and didnt have many major issues. From there I properly formatted the code as some parts werent properly in line and added some reader notes. In the end, the main thing i that changed in my code was the transition from using an array list to a hashmap, and i can say that in my opinion using a hashmap made it easier to code this final project.

Test: For me to test and check if the system is working correctly, I first checked for each individual component. As mentioned earlier I used the code at the end of the video, test to see if my moisture sensor and pump was working. Once thoes two factors are out of the way, I repeated the code from the previous Lab G which involved the OLED Screen, upon running the lab everything came out fine and the 3 main components of the lab were dealt with and situated correctly. After that I tested the overall code, for which it came to be the graph not working correctly, the axis popped up along with the graph labels however the data being plotted wasnt correct, to solve this I connected the moisture sensor to a different pin and tried running it again in hopes that it would and after I confirmed my new graph with my friend it, everything came out to be running fine and the graph was code was plotting the correct values.

Graph:



Manpreet Dhanjal
219623339
2023-04-10

Learning Outcomes:

CLO 1 - To meet this CLO, I mainly just broke down my code by reading it through and trying to test each part with what its suppose to correlate to. Occasionally i would get some errors that IntellJ would put up that would require me to backtrack my steps and kind of reverse engineering my code to see where the problem was coming from. Along with this, I used the small red light blug that comes up on the left side of the code when you highlight a problematic area in the code. Along with this I would also refer back to previous techniques that i've been taught in grade 12 when needed, such as writing code on paper when in the need to debug by yourself.

CLO 2 - Firmata4j and StdLib are appropriate for this minor project as these two allow the the code thats being written to very easily interface with the ardunio uno, and overall from the computer to the ardunio. I used StdLib when creating the graph, graph axis, and, graph labels to display the data that was being read by the moisture sensor. The use of frimata4j came into play when i needed to display the status of the plant and values of the pump on the small OLED screen along with that i also used it when taking the soil moisture values and determine whether the pump needed to be turned on or not.

CLO 3 - For this CLO, I used a hashmap. Before the point on the graph was displayed it was passed through and stored in the hashmap. From there it was passed onto the graph and then plotted.

CLO 4 - What I did for this requirement was, I made a separate class called "task" and in this class, I added a state machine. The state machine would mainly just take a look at the current value that the moisture sensor was reading then match it to the values that are present in the conditional that i gave. That conditional being, if the sensor value is under/over a certain number switch off/on the water pump. After that it would print out the text on the screen of the plants state and would then switch the water pump off/on.

CLO 5 - Throughout the process of writing this code I started out strong and decided to use my previous knowledge from the labs to do this code, as time when on i noticed that errors were coming up in my code and that it was getting harder to track out where the problem was coming from. So to do fix this issues from what i initially started with i pretty much just backtracked my code to the start and I broke it down, moreover making it easier for the user to read and along with that for the coder(me) to solve upcoming issues that may popup. In terms of object oriented programming concepts that were applied in this code, I pretty much made classes that would represent and execute cerain parts of this project individually, these inclued the OLED screen, Moisture, and, Pump. Inside the classes I added many variables that took ahold of the operations of the classes, and overall just controlled the way the devices worked.

Conclusion: To wrap it all up, the minor project was a successful project that was objective was to try to automate plant watering, by the use of an ardunio, moisture sensor, and water pump. By completing this minor project we have proven that life can become easier for us by removing stuff that involve time being set aside for things that need tending, doing so can provide more comfort in life and enhance the overall effectiveness in us. Along with this provide a great way for a plant to be watered and maintained.