EECS1021A - Object Oriented Programming
Automated Plant Watering System
Omar Al-Chikha

Introduction

In this project, an automated plant watering system is designed and developed through a Java application interfaced with an Arduino-compatible device. The system can monitor soil moisture content to the extent that it can auto-activate a water pump when it is too dry. An OLED screen displays real-time data about moisture levels and pump status. This demonstrates good competence in hardware integration into software for the realisation of responsive real-world applications.

Context

This minor project showcases an automated system plant watering system that is crucial for consistent and effective plant maintenance inside a building. What: A plant watering system automatically waters a plant according to real-time data regarding the quantity of moisture so that it is well maintained without manual labour. Why: Watering plants could often become a source of inconvenience for busy people, so it is better if plant maintenance could become automated. The potential of applications for this system goes beyond household use and can also incorporate greenhouses, agricultural automation, and smart garden systems. Exceeding just the basic expectations, this application will lead you into the next phase.

Technical Requirements/ Specifications

- **System Functions:** Constantly monitors soil moisture and turns on the water pump once the moisture drops below a certain level and off when found sufficient.
- **Hardware:** Arduino-compatible board, soil moisture sensor, water pump, OLED display (SSD1306).
- **Software:** A Java application with Firmata4j to communicate with the hardware.
- **User Interface:** The OLED display will visualise the moisture detected, along with displaying the status of the system.
- **Operation:** The system operates continuously in monitoring the humidity every second and managing water pumping in a dynamic manner based on sensor information.

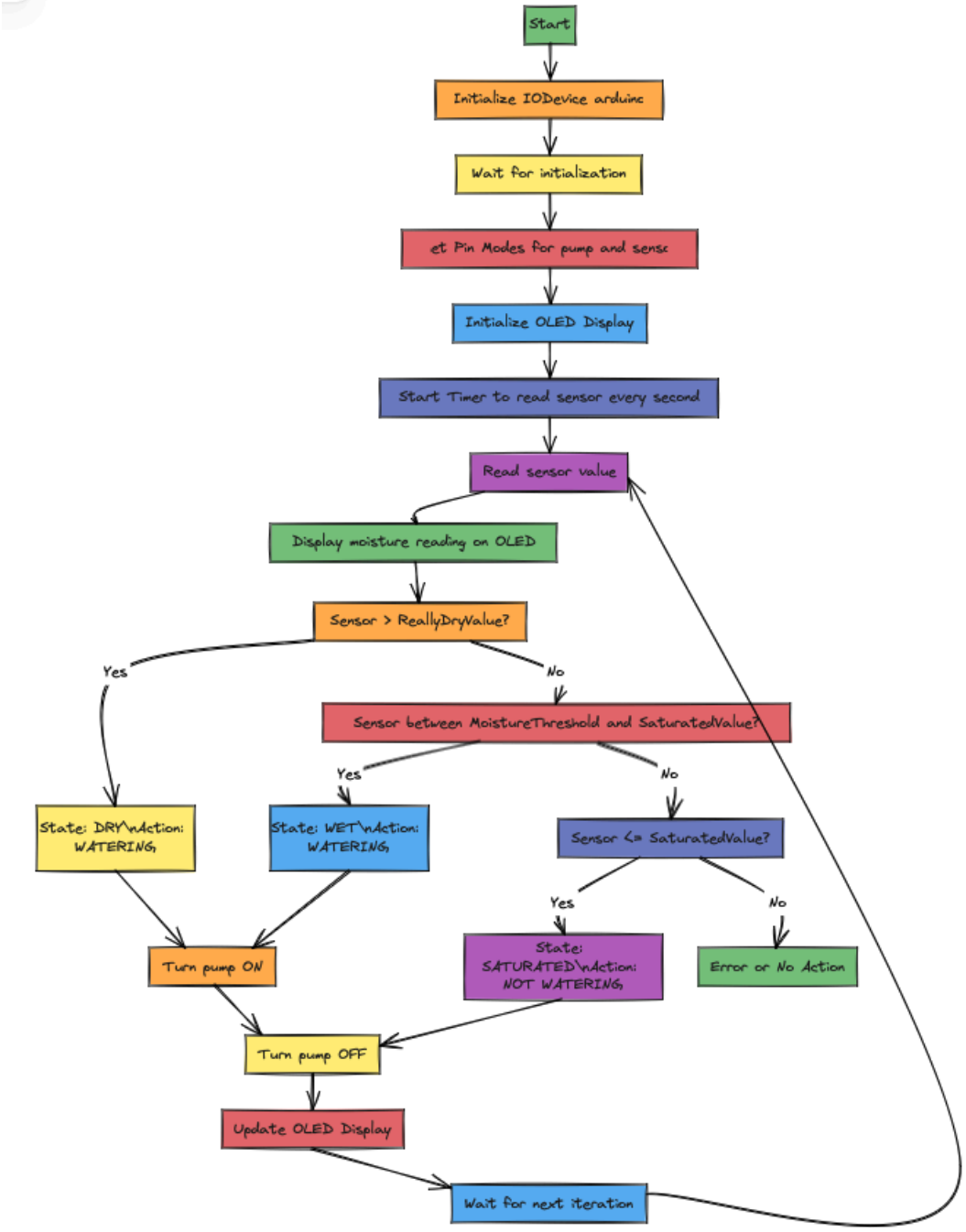

Figure 2: Flowchart for Code, created using blockandarrows.com

Components List

- **Arduino-Compatible Board:** It will be the central controller that interfaces with the sensor, pump, and display
- **Soil Moisture Sensor:** Provides real-time data on the percentage moisture content in the soil.
- **Water Pump:** Waters the plant when there is a drop in moisture levels.
- **OLED Display (SSD1306):** View the moisture readings and system status, making the whole process of monitoring quiet.
- **Wiring and Powering:** Ensure that there are solid, reliable connections and that all components are powered up.
- **Software Tools:** Java programming language; Firmata4j library for interaction with Arduino.

Figure 2: My Plant Watering System

Procedure

1. **System Setup:** I set up all the hardware components by connecting a soil moisture sensor, a water pump, to the Arduino-compatible board. Note: The OLED Display is built into the arduino board
2. **Programming:** Created Java code to handle sensor input, pump control, and to update the OLED in real-time. Utilised the Firmata4j library for interacting with the Arduino board from the Java application.
3. **Integration Testing:** All the tests were done to ensure that the sensor readings properly initiated the water pump and the system status was reflected correctly on the display.
4. **Iteration and Optimization:** The moisture threshold values were tuned, and the intervals of timing were set into place to make the best of all possible systems. At each iteration, several tests confirmed the improvement.
5. **Final Validation:** Ran the whole system over a long period for stabilisation and to establish performance under a range of moisture conditions.

Test

The set of scenarios consisted of simulating different soil moisture conditions:

- **Dry Soil Test:** I maintained the soil moisture as low as possible in order for the water pump to be turned on. Sensor readings were then watched to ensure the ON condition of the pump.
- **Moist Soil Test:** We used soils that were already in a moist condition to check whether the pump had stopped and that soil moistness had reached the anticipated set point.
- **Saturated Soil Test:** The system was tested under high moisture conditions to ensure the pump shut off in good time to avoid overwatering.
- **Long-Term Test:** The test running was left on for a long time, in which the plant underwent many cycles of wet-dry to prove consistent performance
- **Note:** OLED gave real-time feedback and the pump responded to the sensor data very accurately, where all technical requirements could be confirmed.

Learning Outcomes

1. Testing and Debugging (GAI 2b): The sensor readings and activation of the pump were all working correctly through having the program tested and debugged properly. Some of the many debugging approaches used are calibration of the sensors, code iteration, which eventually led to a possible solution.
2. Building an Application (GAI 4b):An application in Java was constructed, applying the API of Firmata4j, according to all specifications given—real-time data monitoring from sensors, control of hardware, and presentation of data. A solution is made effectively and durably to professional standards.
3. Event-Driven Application (GAI 4b): The project follows a clear event-driven model whereby soil moisture levels act as an event determining whether the water pump and

display are activated. This design is essentially connecting sensor input to physical actions and, therefore, an indication of advanced understanding of the event-driven programming concept.

4. Object-Oriented Programming (GAI 4c): Designed an object-oriented Java application which integrates hardware and software in order to solve a real-world problem. This project showed the flexibility of object-oriented principles when coming up with modular code that is both maintainable and has applications across all engineering disciplines.

Conclusion

Hardware and software integration in this project result in a successful automated plant watering system. Designed with perfection through testing and iteration, the system ensures proper monitoring of soil moisture levels and control of the water pump to maintain proper plant growth conditions. This project fulfils not only requirements for coursework but also serves as a basis for continued, improved work on automated systems and smart agriculture.