Auto Plant Watering with Arduino and Java - Main Project

Ben Petlach
220938080
March 28, 2024
EECS 1021 - Z

Introduction

The Automated Plant Watering System is designed to maintain optimal soil moisture levels by integrating an Arduino-compatible microcontroller with Java-based control software. The system continuously monitors soil moisture and activates a water pump when necessary. This project provides an excellent example of an embedded system that utilizes sensors, actuators, and software to achieve automation in plant care.

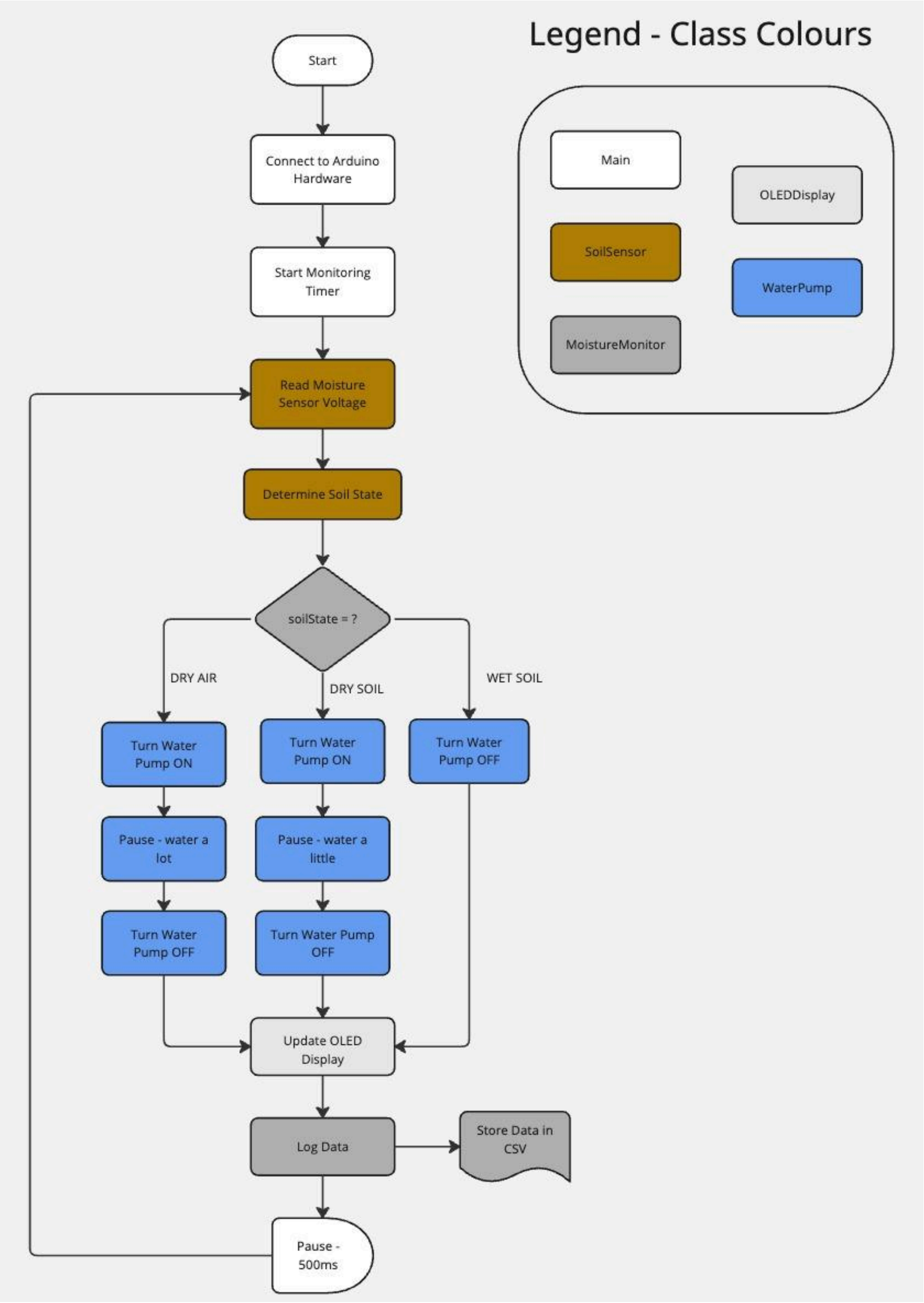
Context

With increasing urbanization, indoor gardening and plant maintenance have become more popular. However, maintaining a consistent watering schedule can be challenging. This project automates the process, ensuring that plants receive adequate hydration without human intervention. Automated watering systems are crucial for efficient agriculture, smart gardening, and even space exploration, where environmental conditions must be carefully managed.

Technical Requirements

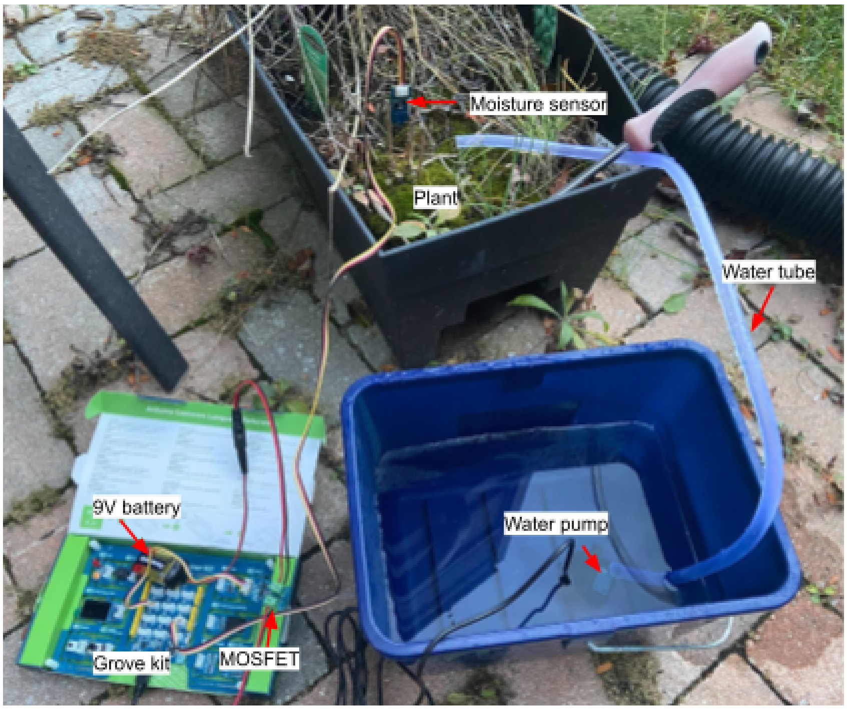
- Continuously monitor soil moisture using a sensor.
- Trigger a water pump for varying durations when the soil moisture falls between certain levels.
- Display real-time moisture levels and system status on an OLED screen.
- Log moisture readings using a Java ArrayList for later analysis.
- Operate autonomously, checking moisture levels at regular intervals using TimerTask.

Event handling is achieved using a TimerTask in Java, which periodically reads sensor data and controls the pump accordingly. The OLED display provides real-time feedback on the plant’s condition.



Components List

- Arduino Grove board
- Soil moisture sensor
- 9V battery
- MOSFET
- Water pump
- Water tubing and reservoir



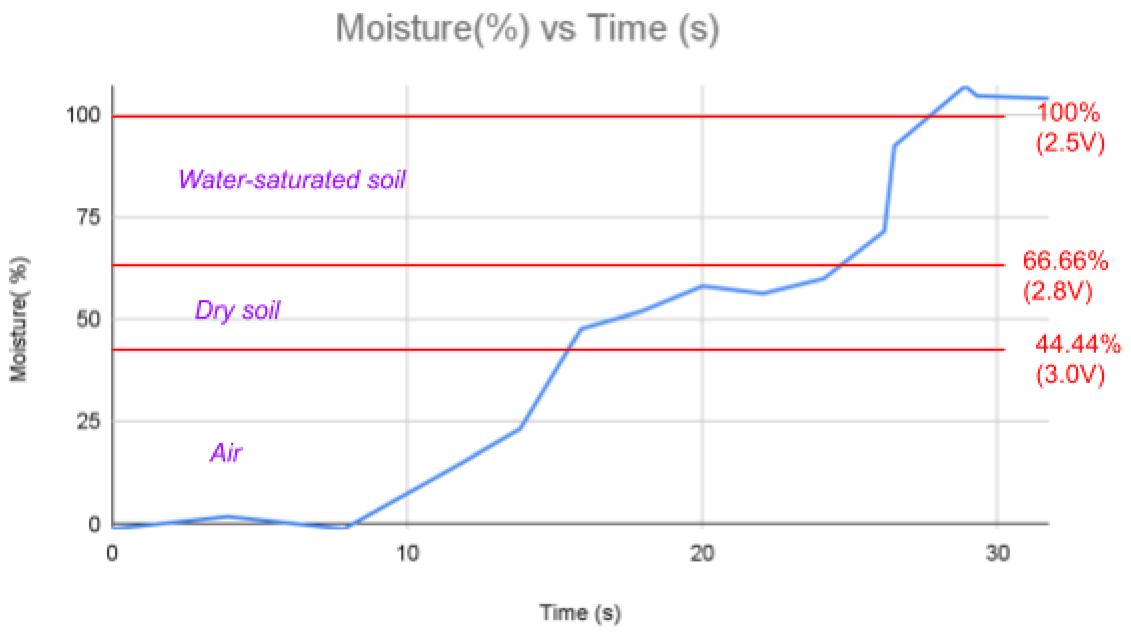
Procedures

1. Developed and planned control flow of the project.
2. Initialized the Grove board with Firmata.
3. Developed Java classes for sensor readings, pump controls, and display integration.
4. Implemented event-driven programming using TimerTask for periodic monitoring.
5. Added OLED support to display soil moisture data and system status.
6. Tested the system by allowing the soil to dry and observing automatic watering.
7. Refined the logic to ensure more accurate soil-watering parameters.

Testing

Unit testing involved verifying:

- Correct soil moisture readings under dry and wet conditions.
- Proper activation and deactivation of the water pump.
- Accurate logging of data.



Learning Outcomes

- **CLO 1**: Debugging Java-Firmata integration to ensure correct hardware communication.
- **CLO 2**: Developed an event-driven application with an intuitive API for controlling the system.
- **CLO 3**: Used Java ArrayLists to store and analyze moisture readings over time.
- **CLO 4**: Integrated sensors and actuators to respond dynamically to environmental changes.
- **CLO 5**: Created a practical, object-oriented Java application with real-world engineering applications.

Conclusion

This project successfully automated the watering process, demonstrating the effectiveness of integrating hardware with software. Future enhancements could include remote monitoring via a web interface or integrating additional sensors for more precise environmental control. This project has provided hands-on experience in embedded systems, sensor integration, and Java programming.