EECS1021 Main Project

Due: April 4, 9 p.m.

1 Overview

This assignment is drawn from James Smith's section of EECS1021.

In this assignment you will grow a plant with the help of an automated plant watering system that combines a program that resides on your Grove board (e.g Arduino + Firmata) and a program that resides on your personal computer (Java + Firmata4j).

Automated plant watering is a fantastic example of an engineering project that combines sensors, electro-mechanical and software components that have real-world applications in a broad range of disciplines. Whether you're a civil engineering student interested in transportation systems or a space engineering student with a vision for developing a micro satellite, the components, system-integration and computational thinking skills developed in this main project will be directly transferable. Basically, you'll be making a plant watering state machine, written in Java that communicates with external hardware.

2 Learning Outcomes

Your main project will be evaluated in the context of the following five course learning outcomes:

1. Demonstrate the ability to test and debug a given program and reason about its correctness.
2. Given a problem specification and a suitable API, build an application that meets a given requirement.
3. Use ready-made collections to solve problems involving aggregations of typed data. We will use Java ArrayLists or HashMaps for this purpose.
4. Build an event-driven application that controls sensors and actuators in order to connect events to physical actions.
5. Program common applications from a variety of engineering disciplines using an object-oriented language and solve them on the computer.

3 Equipment

You will need:

- An "EECS 1021 Course Supplies Kit". This is found in the Engineering section YorkU Bookstore (in person or at https://tinyurl.com/2pc8dmdf). If you already have the kit from EECS 1011 then you don't need to buy a new kit. The kit includes the pump and tubing. You'll need to supply a plant (pot, dirt, as well as seeds or seedling or mature plant) and a water reservoir to hold the water used to water the plant. A multimeter, like that found at the Bookstore, may prove valuable, too. Depending on your personal situation, you can use an equivalent electronic system using other Arduino components, but the sourcing of such parts is your responsibility.
- A plant. What kind of plant should you use? One that is easy to grow indoors is best. A potted herb (e.g. basil plant) or potted flower from the grocery store of a florist often will work well. When in doubt, ask a florist: "what kind of plant will grow well indoors in indirect sunlight and need watering once or twice a week?". Avoid cacti or succulents as they are sensitive to over-watering. In your project demonstrations you will need to show that the soil can go from dry to wet so that your soil moisture sensor values change as water is applied. If you find that your soil is too wet to demonstrate the change, simulate soil conditions by applying a wet towel to the soil moisture sensor and then exposing it to air.

4 Specification

You will complete a Java based implementation of a plant watering system, a report, as well as a video summary of your work.

Your implementation must:

- Be a Java program that can control an Arduino-compatible device running Firmata and that this Arduino-compatible device can sense soil moisture in a plant of the student's choosing. The Arduino-compatible device must be able to receive commands from the Java program for which a water pump can apply water to the plant if the soil is too dry. Once the soil is sufficiently wet, the pump is to stop providing water. *The system should be capable of running over multiple days so that multiple wet-dry soil transitions can be achieved.*
- Display data related to soil moisture and pump operation on an Arduino-compatible display like the OLED found on the Grove Beginner Kit for Arduino.
- Utilize a TimerTask or event- driven programming techniques. We will cover both of these concepts in class.

Note that this main project is a continuation of the original project from the Fall 2021 edition of EECS 1011 and we assume that you are familiar with it in that context. If this is not the case, please reach out to the instructors during class or office hours to have a discussion about it.

Your report must be three to four pages in 12 point font and include the following headings and content:

- INTRODUCTION: A short description (three to five sentences)
- CONTEXT: Describe "what" and "why". What does this project do and why is creating an automated watering system important?
- TECHNICAL REQUIREMENTS / SPECIFICATIONS: List of things that the system should do and **include a flowchart here**. You can be more general and less formal ("requirements") or more specific and formal ("specifications"). Provide an overview of your API. Write 2-4 sentences to describe how you handle events. Explain how you use either an ArrayList or HashMap to gather data from the sensors prior to plotting. These all relate to course learning objectives.
- COMPONENTS LIST: Detail the physical items in your system. Write a bulleted list. Provide descriptions to clarify details. A photo of the system is appropriate here.
- PROCEDURE: Describe the process that you used in creating your project. Explain how your project changed between when you started working on it and when you finished it.
- TEST: Detail the design of your unit tests. Explain any other method you used to validate your system's functioning. Include a graph of sensor values vs. time in this section.
- Does your project address the learning outcomes described in this document? Provide one bullet point that details how your project related to each outcome [CLO 1, 2, 3, 4 and 5].
- CONCLUSION: Wrap up in a few sentences. Was the project a success? Would you like to do more with it, and if so, what?

Your video must:

- Be between 2 and 2.5 minutes long.
- Include a demonstration of the water pump turning on and off in reaction to soil moisture.
- Address each of the five course learning outcomes, describing how your project relates to each.

Evaluation Details

The following components of your submission will be evaluated for a mark. The code will be worth 10% of the main project mark; the video and report will each be worth 45%.

- **Java code is be clear, correct and well commented**; we must be able to generate a Javadoc API using your code and your code must be intelligible and well formatted.
- **Report is well written**; sentences must be complete and grammatica and the structure of the report must conform to the specification.
- **Report includes a flowchart**; this must accurately use symbols and reflect your implementation.
- **Report contains meaningful graphs that accurately reflect different states**; these must be clearly labelled and should indicate thresholds that define state changes.
- **Report contains illustrations or photos**; these must be clearly describe the components of the final system.
- **Report reflects on CLOs**; it must include a meaningful reflection on your learning and associated learning challenges, with mention of the course learning objectives.
- **Video contains a demonstration**; this must illustrate your system capturing states and responding to state changes.
- **Video reflects on CLOs**; much like the report, the video should discuss what you have learned as a part of the implementation process.

What you must submit

- **Your code in a .zip file**; name this 'code.zip'. Do NOT include '.idea', '.class' or '.iml' files; onlye submit '.java' files in your submission.
- **Your report as a .doc or .pdf**; name this 'report.pdf' or 'report.doc'.
- **Your video**; you can submit this as an mp4 to eClass. Alternately, submit a document called 'video.pdf', 'video.doc' or 'video.txt' that contains a LINK to a video that is accessible to the teaching team via the internet. If the video is NOT ACCESSIBLE YOU WILL NOT RECEIVE CREDIT FOR THIS PORTION OF THE WORK.

Have fun and GOOD LUCK!!