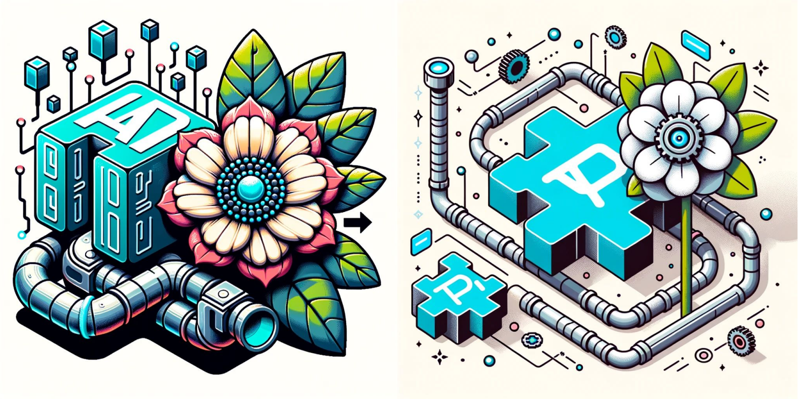
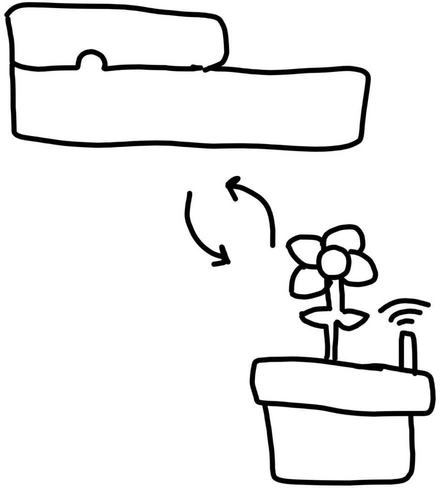
Hydroponic Scheduling

Michael Bishai, Marco Costa





Motivation

Read: https://blogs.chapman.edu/engineering/2023/04/21/computer-science-student-broadens-sustainable-futures-through-self-manufactured-hydroponics-farm/

- Questions:
  - How do we improve the hydroponic farm?
  - What systems need to be put in place to make sure everything runs smoothly?
  - What hardware and software requirements must be implemented?

Logic Demo

Hardware

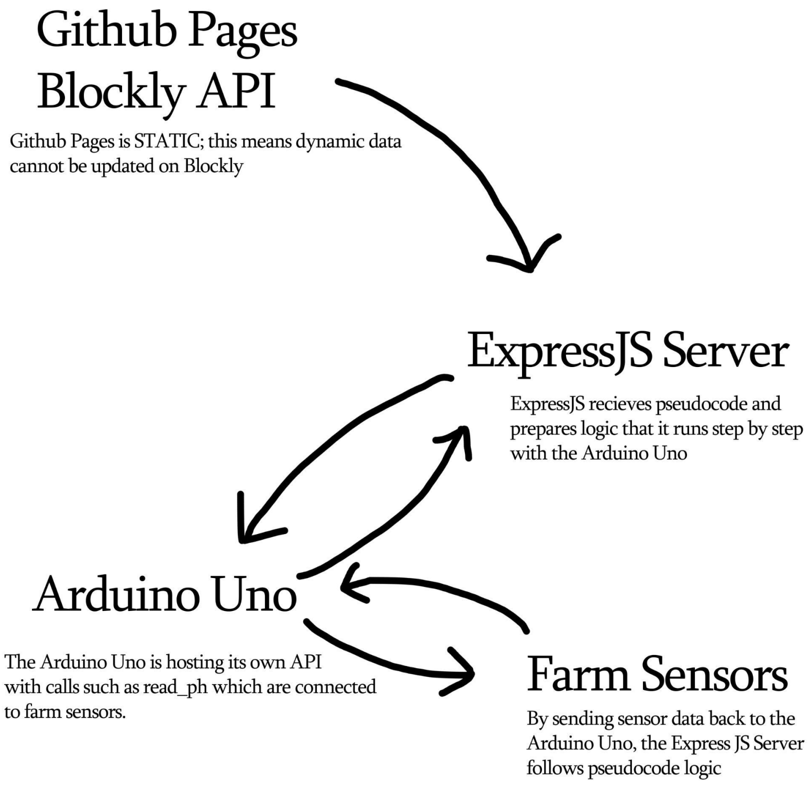
- 1x Arduino Uno R4 WiFi (API Host)
- 1x Hosted Server (Middleman)
- 1x Plant Farm
    - 1x PH sensor
    - 1x DHT11 sensor
    - 1x Temperature sensor
    - 1x Relay
    - 1x Pump

Software

- 1x Arduino Hosted API
    - host/read_ph
    - host/read_temp
- 1x ExpressJS Server
    - Logic interpreter and program counter
- 1x Blockly
    - User end schedule creation

Project Difficulty

- This course project is intended to focus on the generation of code to an existing executable language with a prebuilt interpreter
- We built our own interpreter and logic
- Chapman University WiFi blocks microcontrollers accessing the internet as well as blocks ports for network traffic.
    - Plant demo has to be hosted at Michael’s apartment



Github Pages
Blockly API

Github Pages is STATIC; this means dynamic data
cannot be updated on Blockly

ExpressJS Server

ExpressJS recieves pseudocode and
prepares logic that it runs step by step
with the Arduino Uno

Arduino Uno

The Arduino Uno is hosting its own API
with calls such as read_ph which are connected
to farm sensors.

Farm Sensors

By sending sensor data back to the
Arduino Uno, the Express JS Server
follows pseudocode logic

Demo:

Blockly Demo:
https://cdnmonitor.github.io/hydroponic-scheduling/design-blocks/

ExpressJS Demo:

Arduino Demo:

Plans For the rest of the semester

- Improving Blockly Website Design
- Adding new farm functionalities to blockly
    - For every new puzzle piece, a new Arduino API command has to be hard coded, and an ExpressJS logic handling must be implemented.
    - Everything has to be implemented 3 DIFFERENT WAYS for one feature, in pseudo code, JS, and C.
- Adding lower complexity Arduino Commands
    - DigitalRead() DigitalWrite() AnalogRead() AnalogWrite() are all commands we want to add to Blockly so the system can be used not just in a hydroponic setting; however these commands need catches and exceptions due to risk of ruining the Arduino.
- Farm Improvements
    - Improving the farm will make for a better demonstration in the future

Contributions

Michael Bishai

- Arduino Wiring
- Arduino Code
- Express JS Code

Marco Costa

- Blockly Design
- Pseudocode Language for Express JS Server