Aquaponics Monitoring

Calder West, Hunter Yarbrough, William Mendoza, Payton Smith

UNIVERSITY OF ARKANSAS
College of Engineering

Computer Science & Computer Engineering

Introduction

The objective of our project is to develop an efficient monitoring system that makes maintaining an aquaponics system simple for the average home gardener. This system does the time-consuming work of measuring water levels, plant moisture, pH levels, water temperature, and light levels in an aquaponic system and then automate it, sending the data back to the user and notifying them if anything is abnormal in their system. The data will be readily available on the user's smartphone so they can review it whenever they want and stay on top of their aquaponics system(s) wherever they are.

Hardware Design

Data from the aquaponics system will be measured by five sensors. We are using a breadboard to better organize each sensor's wiring—apart from the temperature sensor, which has its own separate board. These wires are connected to an Arduino board's 3.3V and ground pins. Three sensors are placed in the lower tank with the fish. These include a pH sensor to measure the acidity of the water, a temperature sensor submerged in the water to measure its temperature, and a water level sensor to ensure that the tank does not overflow. The last two sensors are placed in the upper portion of the system to collect data on the plants: a light level sensor that determines if the plants are getting the correct amount of light, and a hygrometer that detects whether the plants are receiving water.

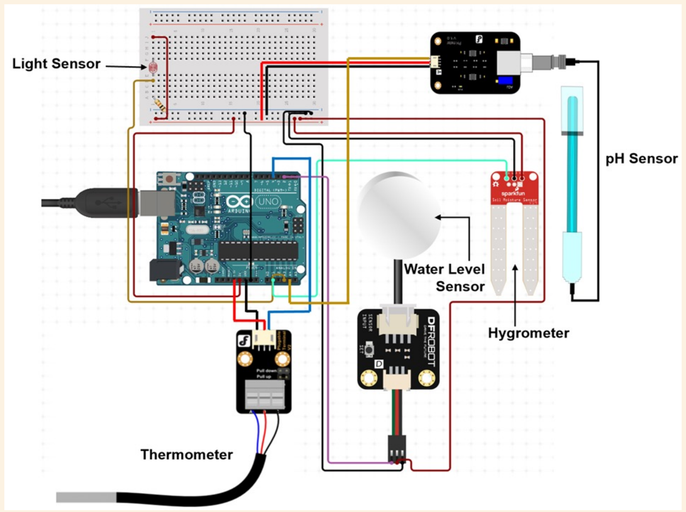


Figure 1: Schematic of Arduino and sensor system with breadboard

Software Design

The Arduino board takes the data collected by each of the sensors and sends it using Wi-Fi to the database to be stored. The application collects data from the database when it observes a change and verifies whether the measurement collected is within its user-specified range. Approximately every 10 seconds, the data within the application is refreshed.

Database

Our server is hosted by Google Firebase, which receives the measurements from the Arduino and stores them as JSON objects. There are two paths: one that stores current collected data, and one that stores past recorded (historical) data.

Application

The frontend of our mobile application is written in Swift. We chose this for its simplicity, as it allows us to offer a clean, organized, and data driven view for the user. A predetermined Arduino Id number connects the user's app to the database location that stores their Arduino's data. The app displays the data and allows the user to configure a range they want the sensor data to be in.

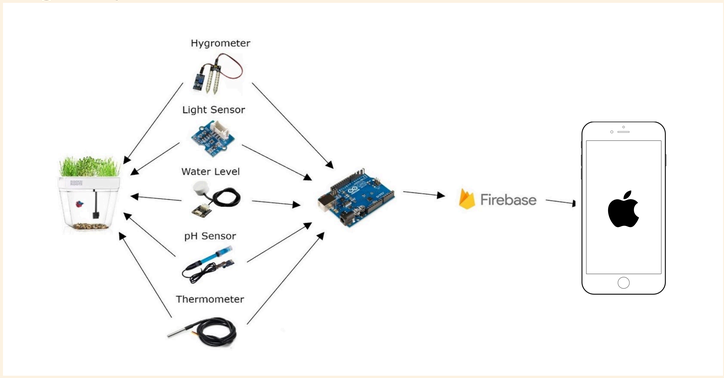


Figure 2: High level architecture displaying how the application and system are linked to one another

Challenges/Roadblocks

In our initial stages, we had issues with faulty temperature sensors. We ended up purchasing 3 different sensors until we decided on one that came with its own separate board; it was only then that we were able to obtain expected measurements. Additionally, given the time constraint, we decided that writing the frontend of the application in Xcode was simpler and quicker to learn/adapt to.

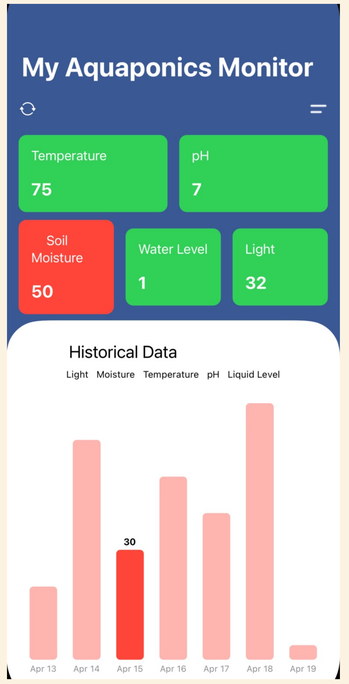
Results

As for the outcome of our project, our application allows users to do the following:

- Sign Up or Create an Account
  Users will provide a username, password, and Arduino ID
- View their measured data
- Set a range of minimum and maximum values which the application will use to determine whether a measurement is out of range and the color of the tile on screen
- View historical data for a time frame of up to a week

Our hardware components allow users to read:

- Temperature
- pH
- Light level
- Water level
- Moisture



Conclusion

Although we faced challenges like faulty temperature sensors and unexpected results with our original language--Android Studio--we were able to overcome them. By:
(1) Choosing to change the language that our application was written in
(2) Testing out different temperature sensors
This allowed us to produce a functional app that meets the goals set at the beginning of the project – to monitor aquaponics data and indicate to the user when measurements are out of range.