Final Project Report
CSCI 43300
Smart Garden

12/08/2020

Riya Shrestha
riyshres@iu.edu
Student ID: 2000466049
Zoe Yang
zoeyang@iu.edu
Student ID: 2000092679

1. **Introduction**

In this project, we will be developing a smart garden that will record the surrounding temperature, light level, and soil moisture of the plant. We plan to build an automated system that waters the plant using a water pump when the soil is too dry, switches on the light when it is too dark and turns on the fan to bring in fresh air when the temperature gets too high. The soil moisture sensor will measure the amount of water in the soil and maintain an ideal condition for the plants. We will be using Raspberry Pi to receive data from different sensors and control different actuators. The messages will be displayed on the web page. When the soil moisture sensor detects the moisture level to be very low, it will signal the Raspberry Pi. Raspberry Pi then turns on the water pump to water the plants until a sufficient soil moisture level is achieved. For the automated light, when the photoresistor detects a low level of light present, it will turn the LED light on that will act like the sun to allow photosynthesis to occur. We will be using fans to bring in fresh air regularly whenever the temperature is above the set threshold. We will be using a web page to view real-time data of environmental conditions and also able to control the system manually.

Hardware platform used:

- Raspberry Pi 4
- Laptop
- For Automated Watering:
    - Soil Moisture Sensor (YL-69)
    - Water Pump and Relay
    - External Power Source (5V)
- For Auto Light:
    - Photoresistor and LED
- For Air Conditioning:
    - Temperature sensor (DS18B20) and Fan
    - Transistor NPN 2N2222

Software platforms used:

- Raspberry Pi OS
- macOS Big Sur
- Python programming language
- Google chrome browser

We used the breadboard, different resistors, wires to build the circuit in this project.

2. **Project design**

Auto watering:

We use YL-69 to sense the humidity of the soil by immersing it in the flowerpot. If the sensor returns that the soil is dry (showing ‘Water me please!’ on the webpage), then the water pump will automatically draw water from the container and water the

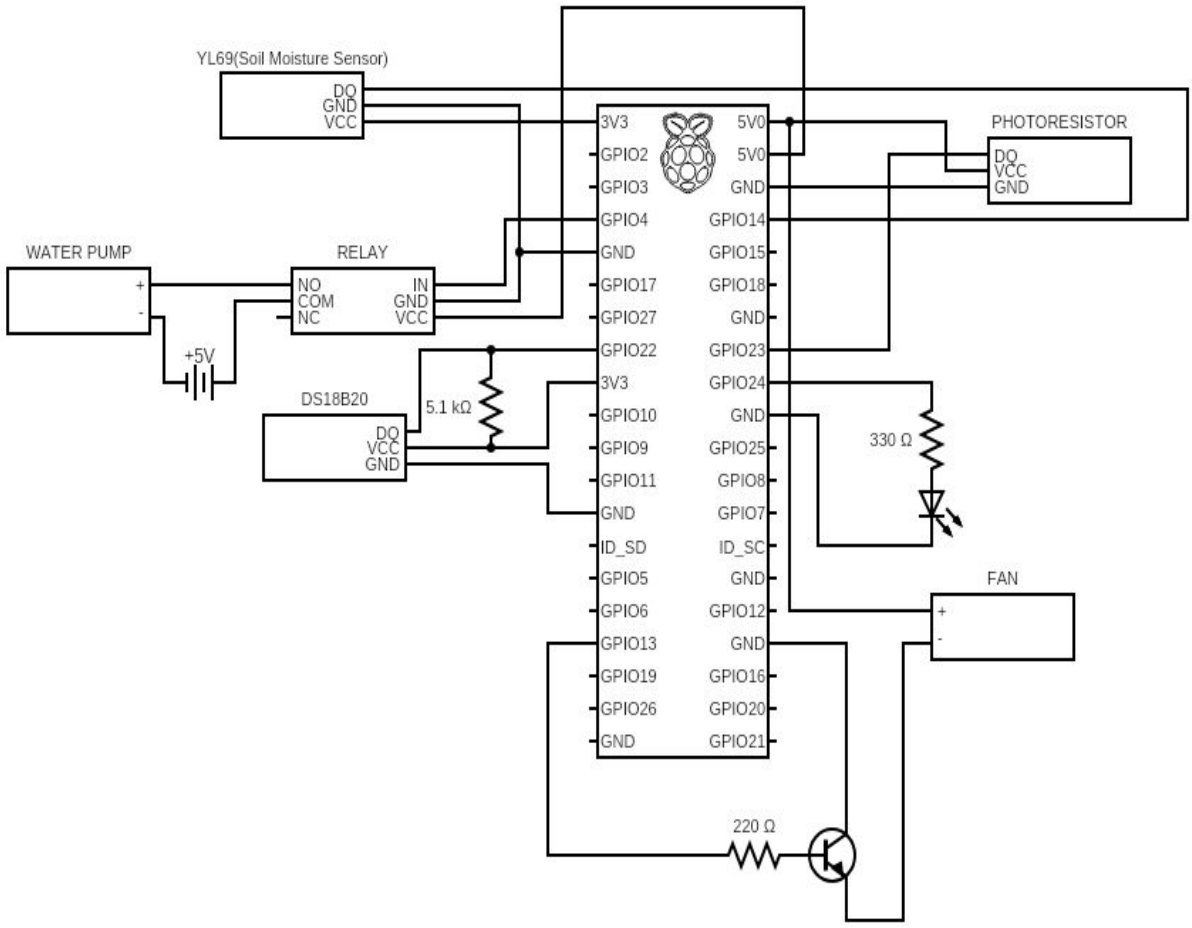
plant. Each time watering lasts for a few seconds and pauses. It will keep watering until soil status returns 'I'm a happy plant!'

Auto Light:
We use the photoresistor (LDR) to check the brightness of the surrounding environment. When we click the "Turn ON Auto Light" button on the web page, the photoresistor measures the brightness. When the sensor measures the low brightness, it returns the value of zero and the LED is turned on. Similarly, when the sensor measures the high brightness, it returns the value of 1 and the LED is turned off. Clicking on the "Turn OFF Auto Light" button turns the LED light and the photoresistor sensor stops until it is turned back on.

Air conditioning:
We use the temperature sensor (DS18B20) to measure the temperature of the surrounding environment. We also use the NPN transistor 2N2222 to control our fan. When we click the "Turn ON Auto Fan" button on the web page, the sensor measures the temperature. When the sensor measures the temperature above the threshold, it turns the fan ON for the air conditioning. Similarly, when the sensor measures the temperature below the threshold, it turns the fan OFF. Clicking on the "Turn OFF Auto Fan" button turns the fan off and the temperature sensor stops, until it is turned back on.

Server and Building Webpage:
We used the python web framework Flask to turn the Raspberry Pi into a dynamic web server, using which we ran our web page on a browser by port forwarding. The web page has three different parts for Water, Light, and Fan.

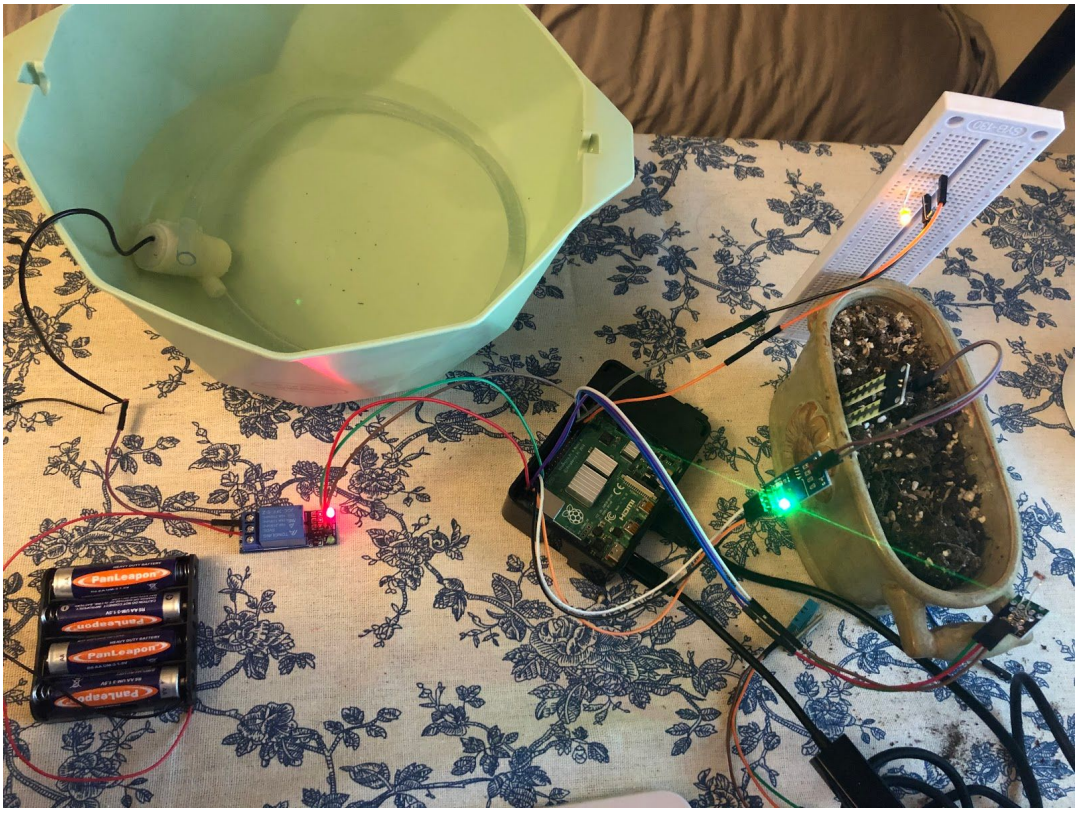
This is the circuit diagram for our entire project.

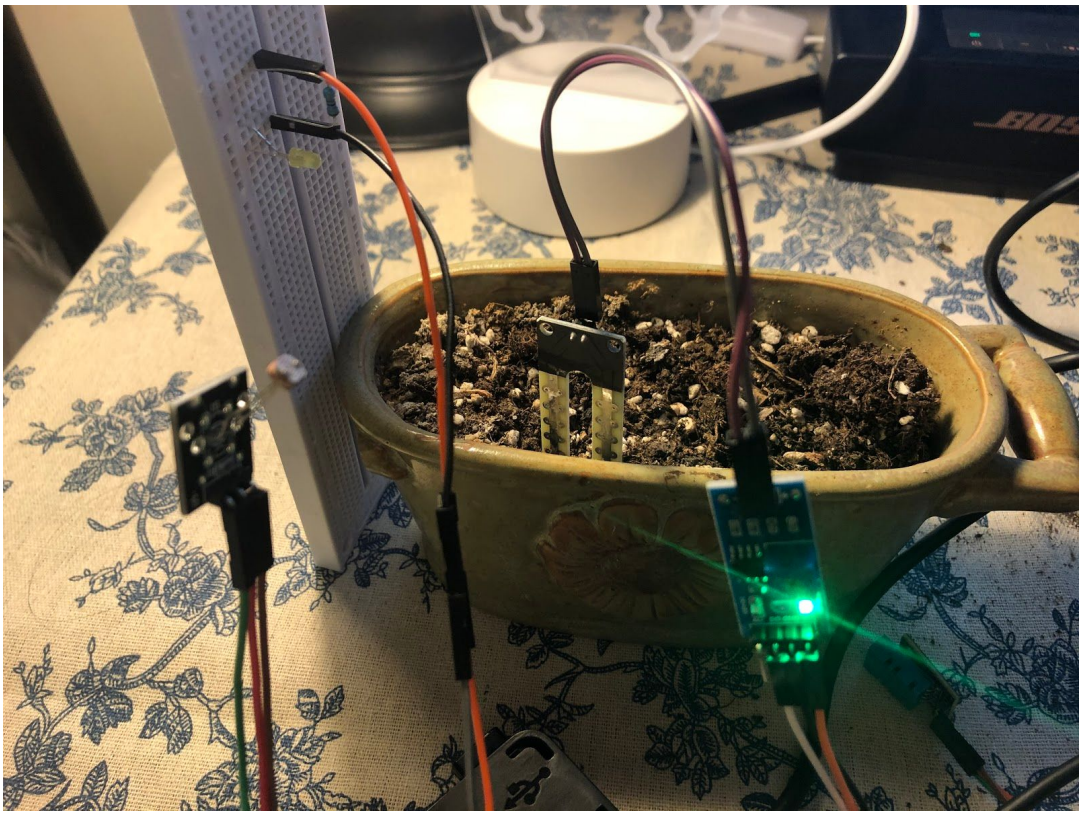


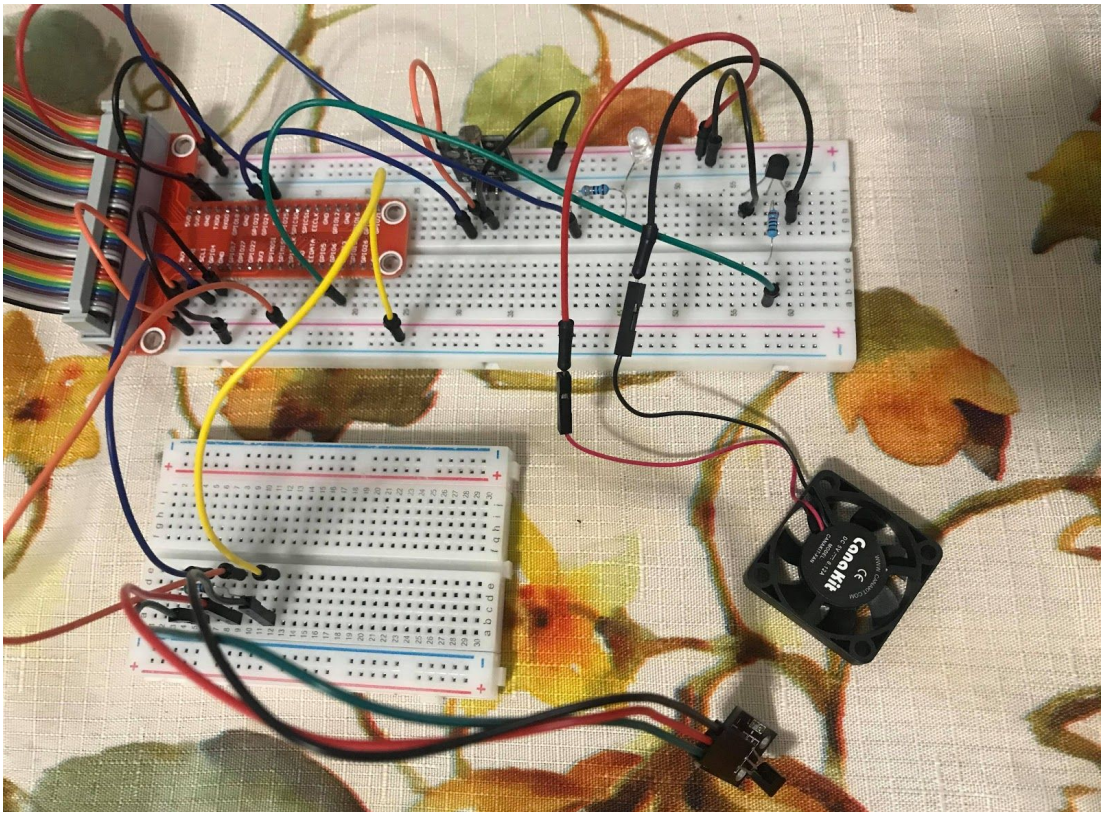
3. Implementation

We used the Python programming language for our project and HTML to build our web page. We used some libraries like Flask, datetime, psutil, and time for this project. Our project works correctly for each part as defined in the project assignment. The “Water” part lets you enable and disable the auto watering system. The “Light” part lets you enable and disable the auto light system. The “Fan” part lets you enable and disable the auto air conditioning system.

Some snapshots of the project:







CSCI 433 Final Project: Smart Garden

Zoe Yang & Riya Shrestha

WATER

Turn ON Auto Watering | Turn OFF Auto Watering | Water Once

Check Soil Status | Check Time Last Watered

LIGHT

Turn ON Auto Light | Turn OFF Auto Light

FAN

Turn ON Auto Fan | Turn OFF Auto Fan

Auto Fan Off

4. **Experiences**

   We had some difficulty managing time as we both live in a different time zone. We struggled most with learning how to run a web page and redirect into its different functions. We started with building smaller parts of the project, so we had problems in combining every part of the project into one. We learned how to use different sensors, port forwarding, and controlling our pi through a web page.

5. **Video demo**

   Demo for Water:
   https://drive.google.com/file/d/1dHKHuCpX5qLU5Ic2bgaCpW5m1m9pR8YS/view?usp=sharing

   Demo for Light:
   https://drive.google.com/file/d/1qeS68ZXsFhVvgNv0kX8x8Nt4F7h9N2aH/view?usp=sharing

   Demo for Fan:
   https://drive.google.com/file/d/1QUaOBLbdJlkKfRKY74q_95rxMyZy8CoZ/view?usp=sharing

6. **References**

- https://www.hackster.io/mokxf16/smart-garden-raspberry-pi-arduino-65c7b7
- https://www.instructables.com/Raspberry-Pi-Powered-IOT-Garden/
- https://www.hackster.io/mtechkiran/smart-home-gardening-system-using-raspberry-pi-1570a7
- cyber-omelette.com/2017/09/automated-plant-watering.html