Plant Monitoring and Maintenance System

Sergio Gutierrez
dept. of Electrical and Computer Engineering
Cyber - Physical Systems
ECE453

***Abstract*—Hydroponic propagation is the process of propagating or sprouting and growing plants suspended in water. The Plant Monitoring and Maintenance System sees to control the environment around a hydroponically grown plant to assure proper health and development.**

I. Introduction

When suspended in water, there are various factors that should be considered when growing sprouts or seeds hydroponically. This system will contain various sensors and actuator to correct these factors so as to promote proper growth. This is done with components such as a temperature and humidity sensor as well as a DC Motor controlled fan. This will be done using a raspberry pi to interface between all the sensors and actuators to work autonomously to do its job.

A. Technical Details

The system will use a Raspberry Pi 3 to connect and program the system. The wiringPi library will also be used to help streamline development of certain actions.

II. System

The Plant Monitoring and Maintenance System will contain two containers: a reservoir and a container for the plant. The system would react when certain sensor thresh-holds are triggered within these containers and its surroundings. Below are a list of components which will be used

- Water Level Sensor
- Water Pump
- Temperature/ Humidity Sensor
- 4 Terminal Relay
- Water Pump
- DC Motor
- PCF8591

The sensors and actuators can be broken into 2 main sections: water level monitoring and temperature/humidity control.

A. Water Level Monitoring

Both containers, the reservoir and the container holding the plant will have water sensors. When the water level drops below a certain set threshold, it would prompt the water pump to start pumping water for a certain small amount of time too fill the container holding the plant. Similarly, the reservoir would also have a water level sensor. When the water decreased past a certain point, a buzzer would activate in intervals to alert the user it needs to be refilled.

B. Temperature/Humidity Control

The container holding the plant will have additional sensors apart form the water level senor. A temperature and humidity monitor and light sensitive LED would be taking constant readings of its surrounding. When the temperature and humidity of the surrounding of he plant passed a certain threshold, the DC Motor, fitted with a fan blade attachment, would begin to spin for a given amount of time to try and lower the temperature of the plant and provide adequate air flow. The light sensor LED would work similarly, after a certain period of being exposed to strong light, the fan would again turn on.

C. Sensors and Actuators

1) DC Motor and Water Pump: Most sensors present in the system were controllable by the Raspberry Pi GPIO pins; however, the DC Motor and Water Pump could not. The GPIO pins are meant to control and communicate with actuators and sensors meaning a different method had to be introduced to control these two modules. For this system I used a relay module. As opposed to use a motor driver which could control the speed of the DC Motor, for this system, it was only required to be on or off. Each relay has 3 terminals: NC, NO, and COM/CO which follow the logic depicted in figure 1. The connections for the pump and dc motor are:

- DC Motor: COM: Motor Input B, NC: 5V
- Pump: NO: Pump VCC, COM: 5V

The connections for the pump and dc motor can also be seen in figure 3 (the circuit diagram). Both modules work with different relay operating modes. For the motor, we can see Input B is in the COM terminal of the relay. This is because once the motor is connected to power and ground, supplying Input B with power through the relay will turn the dc motor in the desired direction only when a signal is output to the relay. The pump differs in that there is no control wire, there is only power and ground. This is why it uses a different relay operating mode.

2) Water Level Sensor: To detect changes in water level, we used a water level sensor by SongHe. The 10 parallel conductors on the front act together as a variable resistor which changes its value depending on the water level.

3) DHT11: For temperature we used sensor DHT11. The system calls the sensor every 10 seconds to check temperature of environment around the container. The sensor also reads humidity; however, it not used in the current state of the system.

In		Out	
Coil	CO	NC	NO
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

In			Out
Coil	NC	NO	CO
0	0	x	0
0	1	x	1
1	x	0	0
1	x	1	1

Fig. 1. Relay Logic

D. Models

Figure 2 shows the general model created during the initial development of the system.

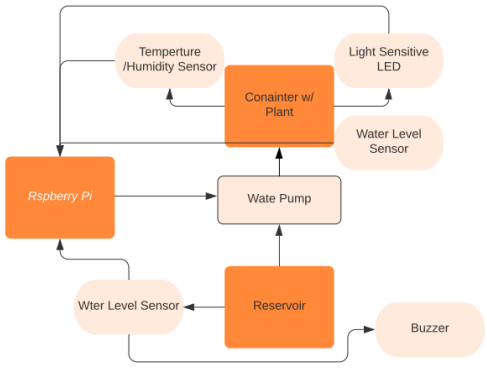

Fig. 2. System Model

Figure 3 is the finite state machine representation of the system. Below is a description of the FSM:

- Input: Water Level, Temperature →pure
- Output: DC Motor, Water Pump →pure

After initializing, the system will read the water level and the temperature of the environment. After reading these values, if they fall below or above the desired set values, the system will then change the pin mode of the respective module on. Then the module will run for a certain amount of time. Once the system has to read data, there will be a delay so as to not flood the system with readings.

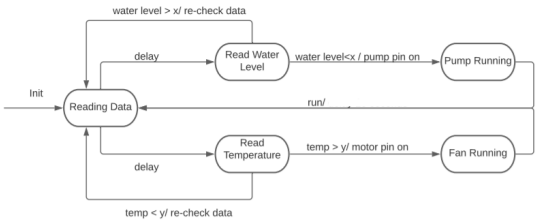

Fig. 3. FSM

III. Progress

A. Mid Year Progress

The water level sensor with the buzzer works to alert the user of decreased water level of the plant container. The water sensor would not give proper readings at first but after wiring it with module PCF8591, I was able to write code to work properly. The temperature and humidity sensor are also working properly and provide a constant stream of data readings which can be seen in figure 4.

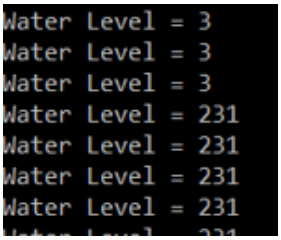

Fig. 4. Water Level Sensor - Mid Year

B. Current Progress

In its current state of development, the system works as initially planned. When the water of the plant container drops below a certain amount, the pump will run for a small amount of time and wait ten seconds and check again to ensure the water level is above the desired value. When the system detects a high user-set temperature, the fan will run for 10 seconds then delay for 10 seconds to allow system to recheck temperature. As can be seen in figure 5, the system now outputs water level as well as current temperature. The bottom two outputs in the terminal show the pump running and the water level changing, showing the functionality of the water pump.

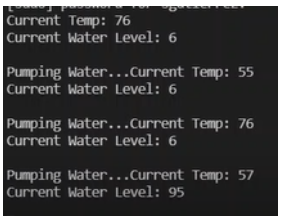

Fig. 5. Current Progress

C. Circuit Diagram

IV. Timeline

Figure 7 shows a gantt chart with estimates of when each component in the system was completed starting from the week of 18th of October. Realistically, the chart contains over estimations of how much time each component will take. This was done to account for possible errors or events that could hinder development.

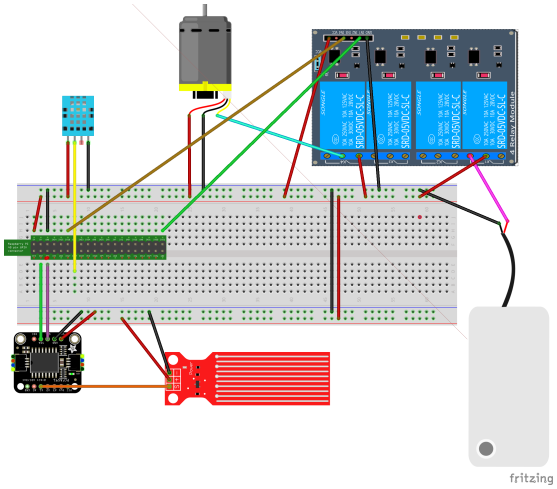

Fig. 6. Fritzing Circuit Diagram

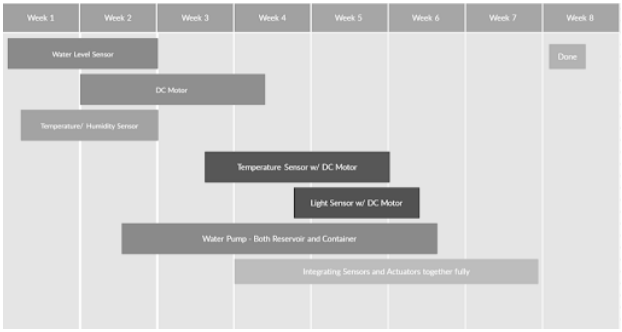

Fig. 7. Gantt Chart Timeline

V. Current Issues

When initializing, the system will occasionally read high temperatures; however, it rectifies itself after checking the temperature one more time. This does make the DC Fan run.

VI. Further Development

In its current development stage there is room for improvement.

1) DC Motor: The DC Motor will run for 10 seconds and then stop allowing the system to check the current temperature of the system. While the temperature remains high, the dc motor will continue to run. While this aligns with the purpose of the system, having the DC motor potentially constantly running during a hot temperature could be detrimental to the system.

2) Water Level Sensor: As can be seen in 6, the water level sensor is connected to an ADC module. The reference used to run a module like it used this module. Further development to remove that module from the system would make the system potentially more streamlined.

3) Different Temperatures: While the system detects changes in temperature over a certain amount, implementing a functionality to detect and notify if the temperature falls below a certain temperature could also be useful for the system.

4) Buzzer: For the mid-year progress update, the buzzer included in the preliminary general model was included in the system. In its final stage of development, the buzzer was removed. When the water level of the reservoir decreased below a certain value, it would sound to alert the user the reservoir needed to be refilled. In term of its efficacy, the water level of the reservoir did not decrease enough to validate its inclusion in the system.

VII. Conclusion

Reviewing the work of the past semester on the Plant Monitoring and Maintenance System, this was a positive experience. Implementing the various sensors and actuators helped me develop my skills in reference to cyber-physical systems and embedded systems. Revisiting the initial goal of the system, the Plant Monitoring System and Maintenance System achieved the goal of ensuring temperature and water level control.

References

[1] 020 New Ultimate Starter Kit for Raspberry Pi - Adeept Learn, https://www.adeept.com/learn/detail-47.html.
[2] deept - High Quality Arduino amp; Raspberry Pi Kits DIY IOT ... https://www.adeept.com/video/static1/itemsfile/449146Tutorial.pdf.
[3] ECE 553/453: Cyber-Physical Systems, https://www.albany.edu/faculty/dsaha/
[4] ser, Super. “Relay Logic.” MERCIA Relay Computer, https://www.relaiscomputer.nl/index.php/elements.