Biola’s Smart Garden

Joshua Williams

B.S. Engineering Physics, CPE Department

ABSTRACT

Biola’s Organic Garden contains a fish pond and plants that have very specific monitoring and watering needs. No existing solution is able to meet the requirements of watering specific plants at specified intervals while interfacing with sensors to regulate the fish pond. Previously, all watering and monitoring were performed manually. A programmed Raspberry Pi is the perfect solution to monitor and maintain the Organic Garden. Raspberry Pi’s are cost-efficient, support inputs and outputs, and are easily programmed. In this research, the Raspberry Pi was programmed to control valves in the garden’s sprinkler system, allowing individual plant beds to be watered at specific intervals. The system also monitors the fish pond by interfacing with a temperature sensor, a water level sensor, and a dissolved oxygen level sensor. System sensor values were tested against separate instruments to ensure accuracy. The system is headless, meaning it can be connected to over the internet by any device with the appropriate software and login, allowing remote system monitoring and maintenance. All activities performed by the system are logged and can be viewed online, while important alerts are notified over the Internet. A voice-controlled device is programmed to connect to the Raspberry Pi, allowing garden status updates to be accessed by asking, “how is my garden doing?”

OBJECTIVES

Biola’s Organic Garden contains a tilapia pond connected to four gravel beds which grow a range of fruits and vegetables. The gravel beds drain into the pond. The system developed needed to accomplish a variety of regulatory and monitoring tasks for both the gravel beds and the fish pond.

Figure 1. Image of the Organic Garden’s tilapia pond and gravel beds.

A. *Watering of Gravel Beds*

The four gravel beds within the garden must be watered for 15 minutes every hour from the hours of 7:00 AM until 7:00 PM. Previously, a 24-hour outlet timer was used to water all four beds simultaneously for the first 15 minutes of every hour. However, the fish pond maintains the most stable level of nutrients when each gravel bed is filled separately. The system was designed to begin filling a new bed every 15 minutes, filling all four beds individually over the course of each hour.

B. *Maintenance of Pond Dissolved Oxygen Level*

Tilapia require a specific level of dissolved oxygen in the water they live in to survive, typically above 6.0 mg/L. There was no previous method for monitoring dissolved oxygen. The system developed needed to monitor the dissolved oxygen and take steps to maintain an appropriate level.

C. *Maintenance of Pond Water Level*

The water level of the pond frequently drops due to evaporation. The water level was formerly maintained manually. The system developed required autonomous maintenance of the pond’s water level.

D. *Monitoring of Pond Water Temperature*

Tilapia require a specific water temperature to prosper, ideally above 65° F. No method previously existed for monitoring water temperature. The system developed necessitated monitoring of temperature and notification when the temperature drops below 65° F.

GENERAL PROCEDURES

The Raspberry Pi was selected to accomplish the desired objectives. Raspberry Pis are very inexpensive microcontrollers which have substantial processing power coupled with a large number of general purpose input output (GPIO) ports. A new Raspberry Pi was bought and a fresh install of Raspbian was loaded onto its SD card. All programming was done in Python 3 using the default IDLE Python editor.

A. *Watering of Gravel Beds*

The valves connected to the watering system in the garden require AC power, so an AC power adapter powers the valves. To control the valves, a 5V relay board is connected to the Raspberry Pi. Each input on the relay board was assigned an appropriate GPIO pin, as shown in Figure 2. When the GPIO pin for the respective valve is activated, the relay closes and the valve opens. The main Python script was programmed to store the value of which valve is activated. When the script is run again, the code increments to the next valve. The default value for each GPIO output is high, which is changed to low to activate the relay and allow power to the valve.

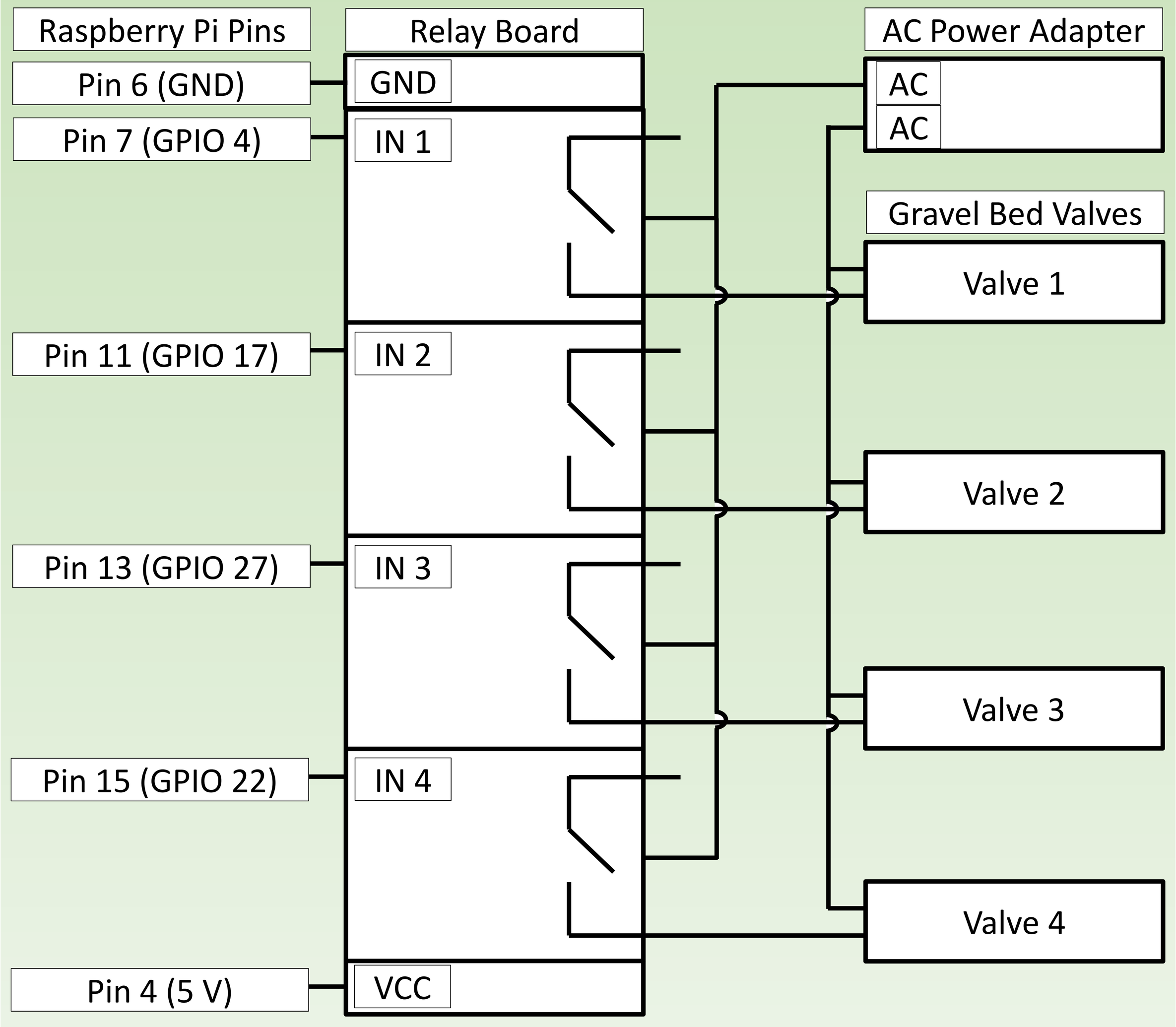

Figure 2. Schematic of gravel bed watering circuit.

B. *Maintenance of Dissolved Oxygen Level*

The Atlas Scientific Dissolved Oxygen Probe was selected to connect to the Raspberry Pi and measure the dissolved oxygen in the tilapia pond. The desired dissolved oxygen level is above 6.00 mg/L. The sensor was connected as shown in Figure 3 and the Python script was adjusted to measure the dissolved oxygen level every 15 minutes. When the level is measured to be too low, GPIO pin 16 is set to low, triggering the relay that is connected to valve 5. Valve 5 controls the water in a “sprinkler system” pipe directly above the surface of the tilapia pond. When water flows through the pipe, water sprinkles down and breaks the surface of the pond, adding dissolved oxygen to the water. When the dissolved oxygen reaches an appropriate level, the relay is opened and the valve is closed.

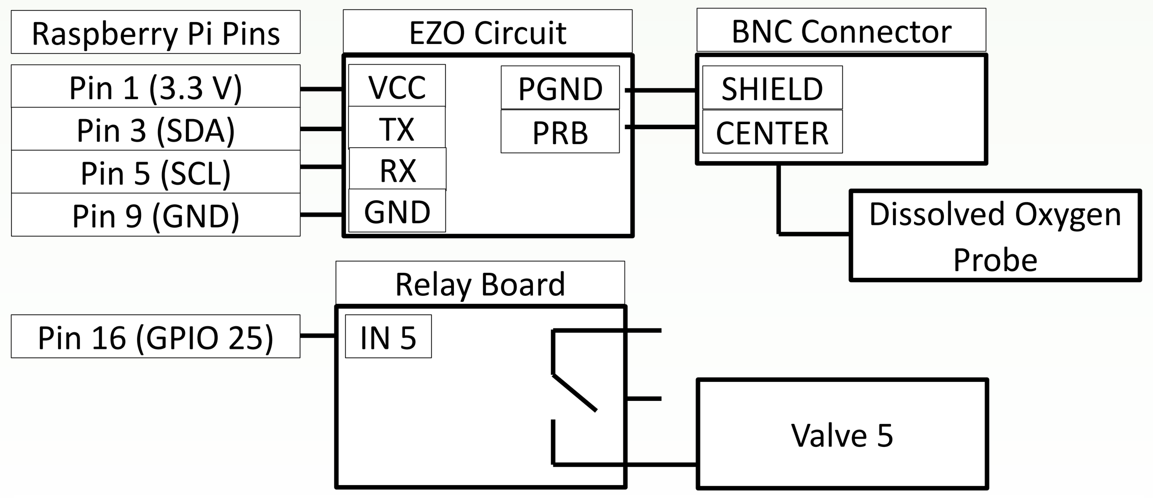

Figure 3. Schematic of dissolved oxygen maintenance circuit.

GENERAL PROCEDURES (cont.)

C. *Maintenance of Pond Water Level*

The water level of the pond frequently decreases due to evaporation and must be adjusted. Every 15 minutes, the water level is measured by an ultrasonic sensor. The distance from the sensor to the surface of the water is stored by the main Python script, allowing the water depth to be extrapolated. The schematic of the sensor is shown in Figure 4. If the water level is too low, pin 22 is changed to low, causing relay 6 to close and power valve 6. Valve 6 is connected to a water line which fills the tilapia pond. When the sensor measures that the water level has reached an appropriate level, the relay switches off and the valve closes.

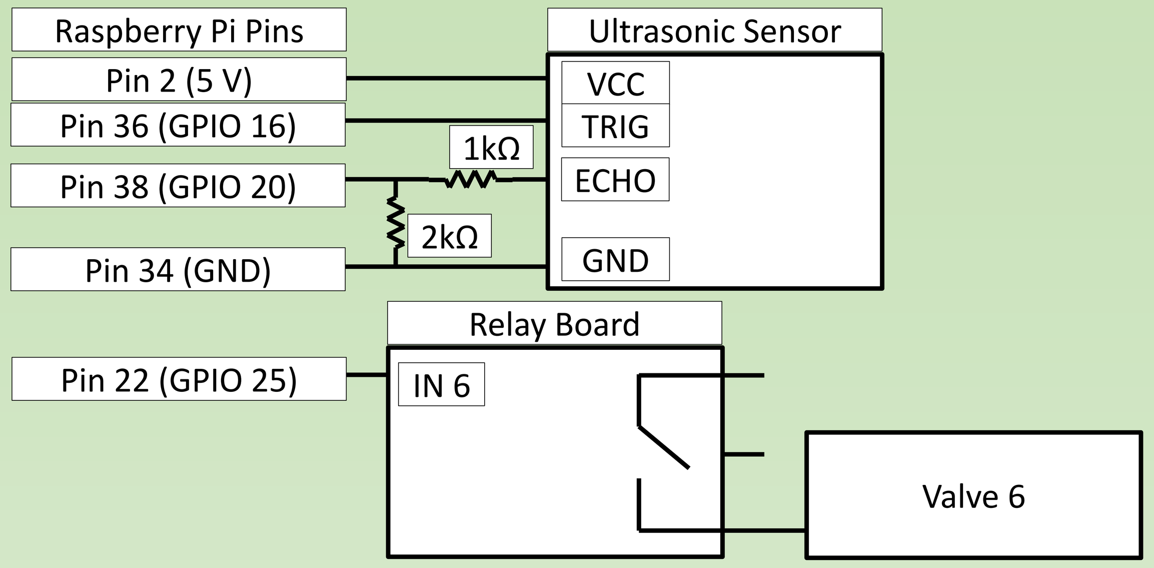

Figure 4. Schematic of water level maintenance circuit.

D. *Monitoring of Pond Water Temperature*

The tilapia pond water must maintain a temperature above 65° F. To monitor the temperature, a waterproof thermometer sensor was connected to the Raspberry Pi, as shown in Figure 5. The main Python script was modified to record the temperature every 15 minutes. When the temperature falls below 65° F, an alert is sent out. Only one temperature alert can be sent per day.

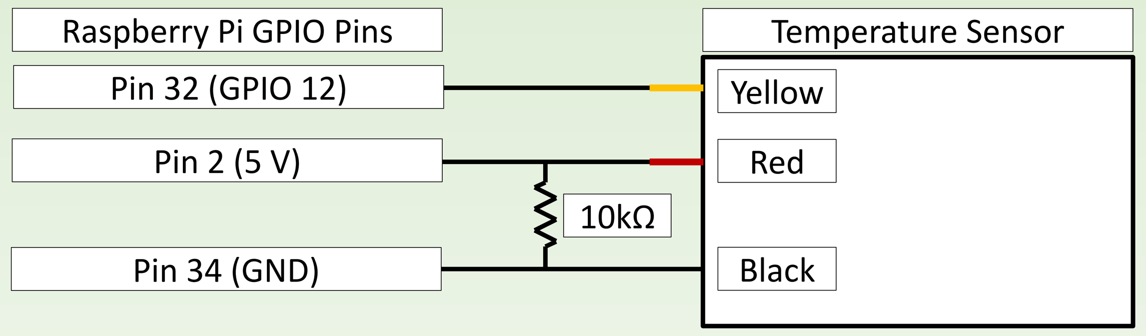

Figure 5. Schematic of temperature monitoring circuit.

E. *Notification of Important System Events*

The system administrator must be notified whenever the system discovers an issue and takes steps to correct it. To notify the administrator, the Python script was connected to a Twitter account. Tweets are sent out when the pond is being filled, when the water’s dissolved oxygen level is low, and when the temperature of the water is low.

F. *System Access and Garden Status Updates*

Since the system is located in a remote location, steps needed to be taken to ensure remote access to the system. The Raspberry Pi was configured to run headless. Connecting to the Raspberry Pi is identical to connecting the system to a monitor, with the connected device providing the inputs. Furthermore, an Amazon Echo Dot was configured to communicate with the Raspberry Pi. The status of the garden can be accessed by asking the Echo Dot, “Alexa, how is my garden doing?” The Echo Dot will then read back the most recent sensor values along with which relays are currently open. Amazon Echo Dot compatibility is still being tested and has not achieved full functionality.

G. *System Case*

The system and components need to be protected from the environment. A 3D printed case was produced to encase the Raspberry Pi, relays, and other electrical components. The case is placed in the barn in the Organic Garden and is connected to the tilapia pond through sprinkler wire run through conduit. A 3D printed case was also developed to protect the ultrasonic sensor, which is located directly above the pond. Both cases were printed using PLA.

RESULTS

Each system feature was tested individually to verify functionality. After passing individual tests, each feature was added to the main Python script and further tests were performed to ensure whole-system functionality. All features were successfully implemented and tested, barring comprehensive Echo Dot interfacing. Regardless, the system is ready for deployment and is currently awaiting necessary infrastructure within the garden to be installed.

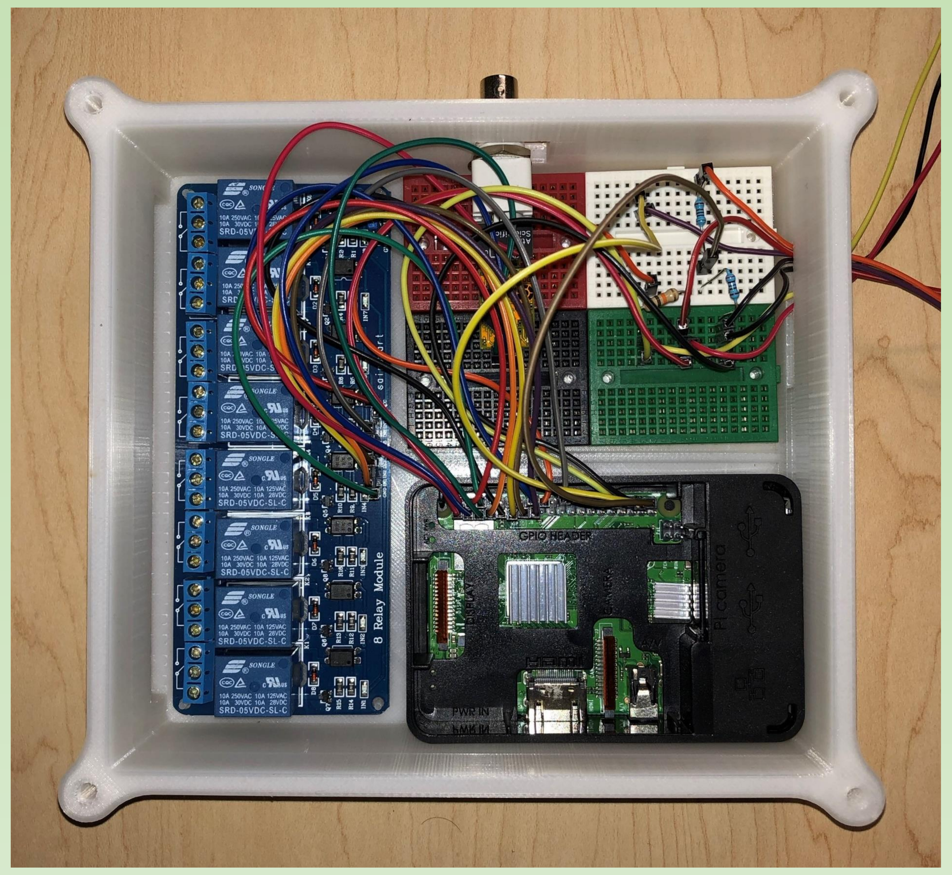

Figure 6. Image of the completed system within its 3D printed case.

DISCUSSION

This research broke new ground into combining technology with gardening. Technological advances have made the Raspberry Pi as strong as a basic computer while staying in a price range that is easily affordable. The utilization of a Raspberry Pi microcontroller to maintain a healthy garden and tilapia pond demonstrates the wide range of applications for modern devices. Furthermore, the system developed in this research has even greater capability and can be built upon as the Organic Garden expands and grows. This research also demonstrated the interfacing of a Raspberry Pi with an Amazon Echo Dot. These two fairly recent devices are not designed to interface together, but this research showed that it can be accomplished. In the future, interfacing between a Raspberry Pi and an Amazon device will likely be expanded as users realize the potential of systems that utilize both devices.

REFERENCES

All pictures and figures in this report were produced by me, Joshua Williams. This report details the methods used to accomplish the desired objectives of this research project. Although this report did not directly draw information from sources other than myself, all daily activities and resources used to accomplish this research project are detailed within the laboratory notebook that was maintained to provide proper engineering documentation of the system. All code for the project can be found at github.com/jwills15/garden

ACKNOWLEDGEMENTS

Dr. Jason Tresser, Associate Professor of Biological Sciences and supervisor of Biola’s Organic Garden.

Dr. Albert Yee, Associate Professor of Engineering and CPE Summer Engineering Internship Coordinator.

Professor Stanley Ng, Professor of Circuits I and Circuits II.

Nathan Coddington (B.S. Chemistry) and Jimmy Su (B.S. Engineering Physics) for designing the 3D printed cases needed for the system.

Dr. Behzad Varamini and the Stewart Science Honors Program.