Greenhouse Temp System

PROJECT DOCUMENTATION

Benjamin King
SOUTHERN INSTITUTE OF TECHNOLOGY | IT715 PROJECT

Table of Contents

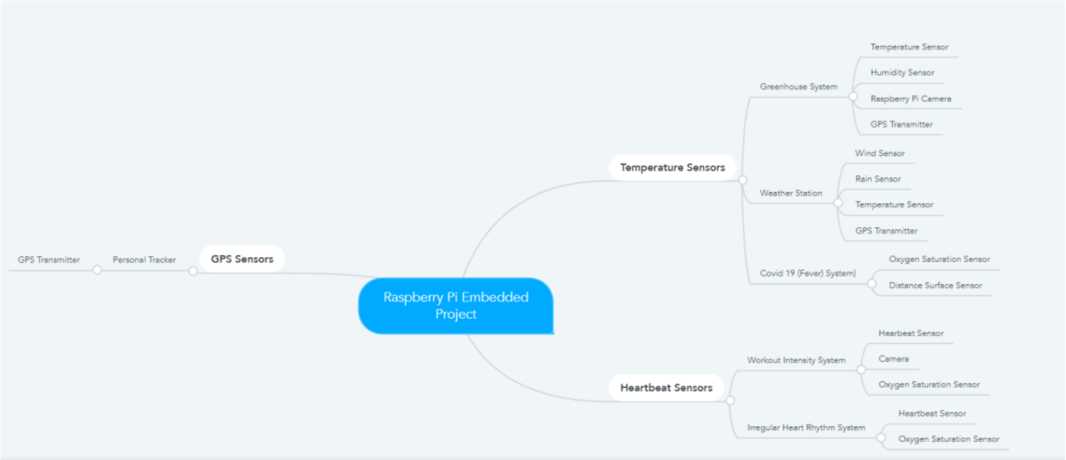
Introduction

This document contains the Planning and Testing methods of the Greenhouse Temp system designed and created by Benjamin King.

Planning

Project Scope Planning

Originally when the project scope was given, there was a large range of options to choose from. As part of the planning stage, I created a Mind map, thinking of the different core sensors I could implement in my system and them building on from them.



(can be viewed at https://mm.tt/1686209466?t=ChktonBC8C)

Thinking about the practical uses of a Raspberry Pi System, I decided on the concept of a Greenhouse Temp System utilising a Temperature and Humidity Sensor.

This Greenhouse Temp System would monitor the current temperature and humidity of the Greenhouse, as well as whether any water was detected in the sensor. This information would be collected periodically, in order to allow the owner to monitor the Greenhouse.

Adding to this idea, I decided that the system should allow the user to easily view the state of the plant(s) its monitoring, such as using a Raspberry Pi Camera to take photos, with an additional Water Soil Sensor allowing the user to see whether the plants need watered.

From this I decided that the different sensors which would make up this embedded system would be a :

- 1x Temperature/Humidity Sensor (DHT11) – this would send readings of the current temperature and humidity in the Greenhouse
- 1x Soil Moisture Sensor – this would send readings of whether water is detected in the soil of the plant(s)
- 3x LED (RGB) – this would allow the end user to communicate to the plants by lighting up using the end-device
- Raspberry Pi 5MP Camera – this would take photos of the plants throughout the days

Due to some of the previous stuff I had, such as the Raspberry Pi 5MP Camera from a previous project, I decided to include this into the project, enhancing the user experience.

Sensor	Sensor Type
Temperature/Humidity Sensor (DHT11)	Digital (processed through a library)
Soil Moisture Sensor	Digital or Analogue (in this project used through the Digital Interface)

Raspberry Pi Camera 5MP	Official Raspberry Pi Camera which utilises the Camera Port on the Raspberry Pi
LEDs (RGB)	Digital Output used through a 330 ohmz resistor bank

Another key feature I wanted to implement was a GPS Transmitter, allowing the user to see the current location of the Greenhouse Temp Kit on a Map. This would provide more benefits to the user in situations where they had more than 1 Greenhouse, allowing them to click on the map and see the latest readings.

To implement this, feature a NEO6MV2 GPS Transmitter was obtained and connected to the Raspberry Pi. Unfortunately, the only available driver I could obtain has not been updated for the Raspberry Pi 4, so this feature unfortunately was not implemented. In the future once the driver is updated, I plan to incorporate this feature in the Pi.

End-User Planning

Originally the end-user user interface was planned to be an Android Mobile Application which allowed the user to view the sensor details (and photo), with a button allowing the data to be refreshed and another button allowing the user to say hi to the plants (light the LEDs up remotely). This communication would be performed through a Flask Web Server running on the Raspberry Pi. However due to time constraints and other commitments this was scaled back to a simple HTML Website allowing the user to view this information and perform these functions.

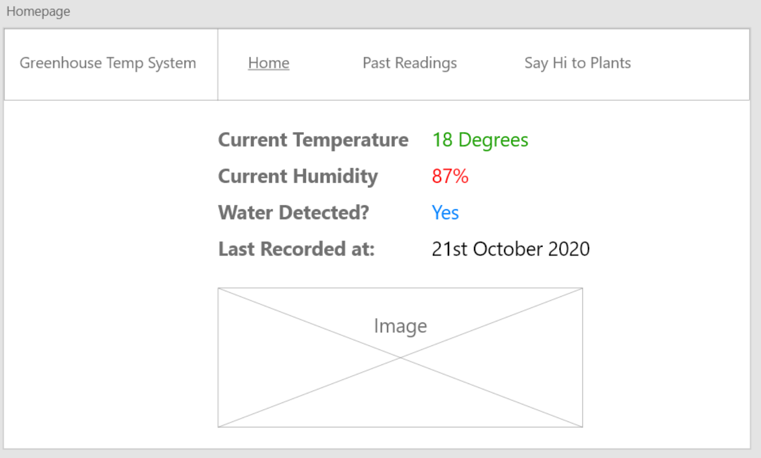
The website will be cross-compatible and able to support a large range of different screen resolutions and devices, improving accessibility of the system where as a mobile app would limit the reach of this application to Android Devices (such this would be where it would be deployed).

A simple 3-page website was decided on, with the homepage consisting of the latest sensor readings and the latest photo taken. An additional HTML Page would show previous readings retrieved from a MySQL Database, while the third page would be the communication from the end-user to the system consisting of the LEDs lighting up.

A consistent theme between the webpages would exist, with a navigation bar at the top of each webpage and a subtle grey background. The latest recordings would stand out on the homepage by being different colours.

Wireframes of proposed website

Below are some draft wireframes of the proposed website showing the sensor readings. These wireframes show the draft look of the website.

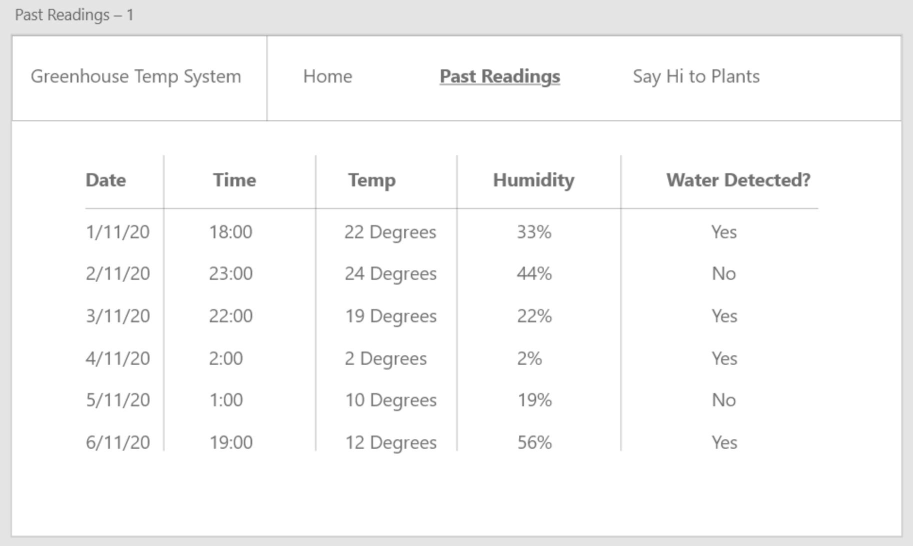


Current Temperature 18 Degrees

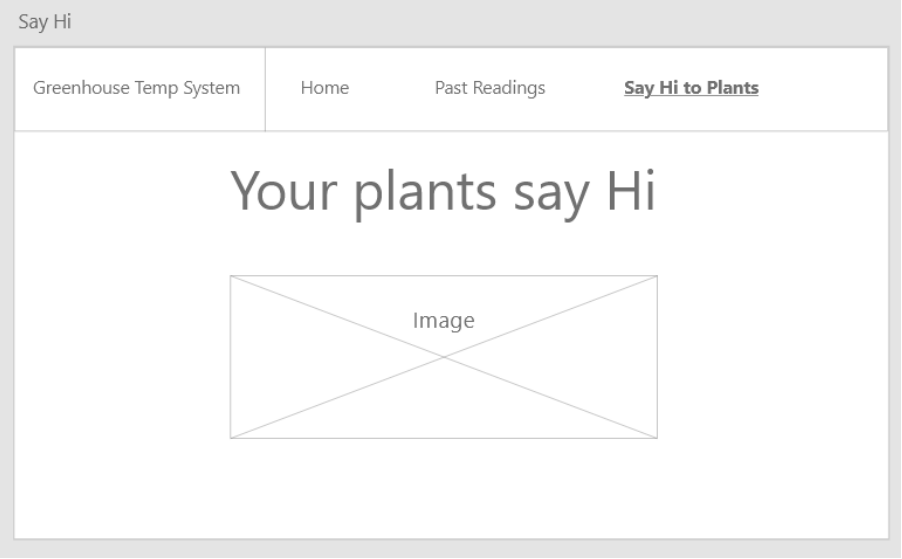
Current Humidity 87%

Water Detected? Yes

Last Recorded at: 21st October 2020



Date	Time	Temp	Humidity	Water Detected?
1/11/20	18:00	22 Degrees	33%	Yes
2/11/20	23:00	24 Degrees	44%	No
3/11/20	22:00	19 Degrees	22%	Yes
4/11/20	2:00	2 Degrees	2%	Yes
5/11/20	1:00	10 Degrees	19%	No
6/11/20	19:00	12 Degrees	56%	Yes



Your plants say Hi

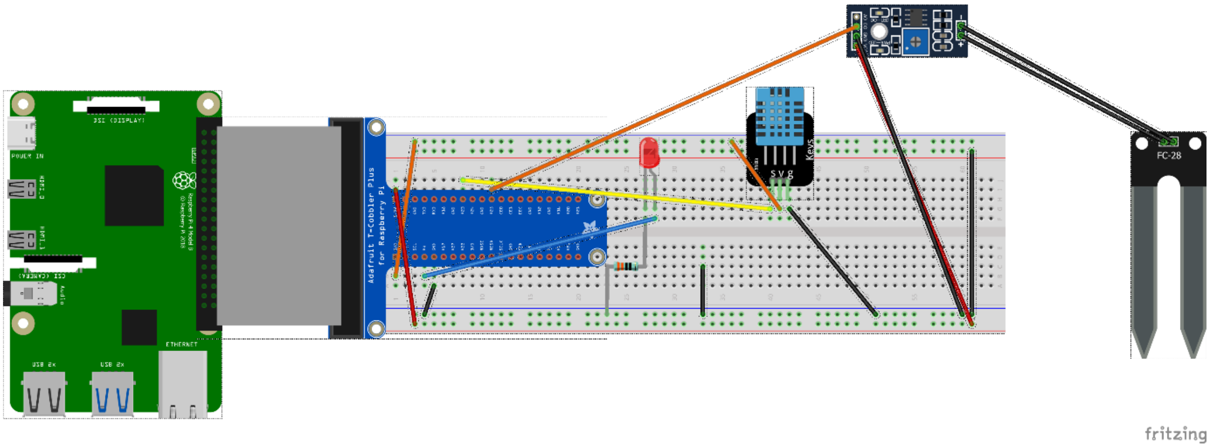
Testing Methods and Data

Testing was rigorously done with the system. The core sensors were all tested to ensure that they outputted live, real results.

Sensor	Testing
Temperature and Humidity Sensor (DHT11)	**Problem** - Initially I was unable to get a reading from the sensor due to a faulty sensor. When the python script ran, the library was unable to detect the sensor **Solution** - Replaced the sensor with another DHT11 Sensor and it could successfully be read. **Problem** - Another Problem which was encountered was that the DHT11 sensor would produce a Checksum error occasionally when the temperature and humidity sensor was read. **Solution** - This error was occurring due to how there was not a long enough time period between different the sensor readings. Originally the sensors were recording every 2 minutes, which the sensor did not like. After changing this to every 30 minutes, this issue was resolved.
Soil Moisture Sensor	**Problem** - The sensor could not detect any water, despite when placed in a cup of water **Solution** - The attached connection was accidentally connected to the Analogue output, meaning that a digital signal was not being obtained. Once the correct output was connected, the correct signal was being received by the Raspberry Pi. - The sensitivity of the sensor needed to be adjusted so it was more sensitive and could detect whether there was water when the sensor was placed. Once the sensitivity had been increased, it correctly detected when water was detected
LEDs	**Problem** - The LEDs would not light up when the user navigated to the flicker page on the Web Server **Solution** - There were two issues which contributed to this issue. One of these was that the wrong Pin Output was configured, meaning that the digital High Signal was going out the wrong pin and not to the LEDs. The other issue was that one of the LEDs was back to front, with the cathode and anode of the LED installed in the wrong direction.
Inserting into the Database	**Problem** - When the data was trying to insert into the database, there was the issue that the data was unable to successfully be inserted into the table. **Solution** - After several different tests it was found that the table had been designed incorrectly, with the date data type of the table being incorrectly setup as an Int variable. Due to how the value that was being inserted was a string, this was resulting in an insertion error. Once changed in the database to a VARCHAR, this worked successfully.

Wiring Diagram

The below diagram shows the wiring of the sensors connected to the Raspberry Pi, minus the Camera (which plugs into the Camera Port on the Raspberry Pi).



References

During the creation of the wiring Diagram I used the following resources for the additional sensors in the Fritzing Diagram:

- Fritzing. (2017). Retrieved November 12, 2020, from https://fritzing.org/projects/soil-moisture-sensor-and-serial/

- Humidity and Temperature Sensor DHT11. (2019, January 12). Retrieved November 12, 2020, from https://forum.fritzing.org/t/humidity-and-temperature-sensor-dht11/6307

- Raspberry pi 4 model B - fritzing forum. (2020, April 23). Retrieved November 12, 2020, from https://forum.fritzing.org/t/raspberry-pi-4-model-b/8622

- Raspberry pi 4 model B - fritzing forum. (2020, April 23). Retrieved November 12, 2020, from https://forum.fritzing.org/t/raspberry-pi-4-model-b/8622