IFB102 – Mini Project

Braydan Newman

n11272031

Project Objectives

The Goal of the project was to automate the prosses of watering my plants while also making the system smart enough to not over water and to let me know metrics about the plant and the system, to better inform me on what I should do to take care of the pant.

To make this project how I wanted it have to tick a couple points that really would make it something useful:

- Check weather data
- Check soil moister
- Check Water level in tank
- Have it controllable and viewable without needed to ssh or log into the pi or servers
- Needs to water the plants
- Needs to use the data that has be gathers and make a choice on if the plant should be watered

Review and discussion of Technologies Used

There were many different technologies used for this project

MQTT

The connection between the raspberry pi and the home assistant Server is done using MQTT as its easy to use and works well with home assistant. Mqtt uses a Publish and subscribe method to communicate between devices and is designed for devices with low connectivity, this works really well for my application as signal strength on the balcony is very week. Something like a constant socket connection wouldn't work as its higher bandwidth and need for a constant connection would make this unreliable and unstable.

The basic premiss of a publish subscribe model is that each device can subscribe to a topic and this will allow this devise to hear all data being transmitted with that as its topic. Whereas publishing is the opposite and you publish data with a topic to then be picked up by devices that have subscribed. All the connections are controlled by the Broker and in the project the Home Assistant server is working as the Broker.

Other options I could have used was have a sockets connection to the pi, but this would be unreliable as discussed before. Another option was run an API end point of both the home assistant server and the Raspberry pi and uses normal HTTPS as the connection method between them, this could have worked but would have been very involved and would take a lot more time to set up. This method would be much more secure but for my purpose's security wasn't a big problem as this is alone a local network with no access from the outside.

FTP

The raspberry pi uses the File Transfer Protocall to send log files to a server running in the cloud, The log files are stored here as good practise measure as data should never be stored in one device and all in one location

FTP was the Protocall of choice for this application as this Protocall is designed for the transfer of files from one server to another.

The Raspberry Pi also has the FTP running on it as this was to upload all the Python scripts used to build the rest of the application

An alternative to this could have been the Secure Copy Protocol as this allows the copy of file between 2 remote hosts. This option would have worked just as well as FTP but due to the well documented and easy to use FTP python library FTB was chosen in favour of SCP

API

The Raspberry pi also fetches data from an Api serving weather forecast data, this data is used to determine whether to water the plants or hold off and wait for rain.

An API or Application Programming Interface is used by software to interface with other software, in this case you send a request to a server endpoint and it sends back weather forecast data.

An alternative to using a weather forecast API is to measure Weather data in real time with sensors connected to the raspberry pi, this approach was considered but due to the increased cost was disregarded, however this will be revisited as locally sourced data that you have ultimate control over is normally better.

Programming Language

The language I chose was python and all the scripts for the MQTT, FTP and Api connection were all done with python, this was chosen as that is the language that I'm most familiar with and most experienced with.

Any other programming language would have worked and have been just as effective, but due to speed not being a concern and pythons' great number of libraries it worked really well for this project

Electrical components

The Electrical components used in the project are a relay, a moister sensor, and a magnet switch.

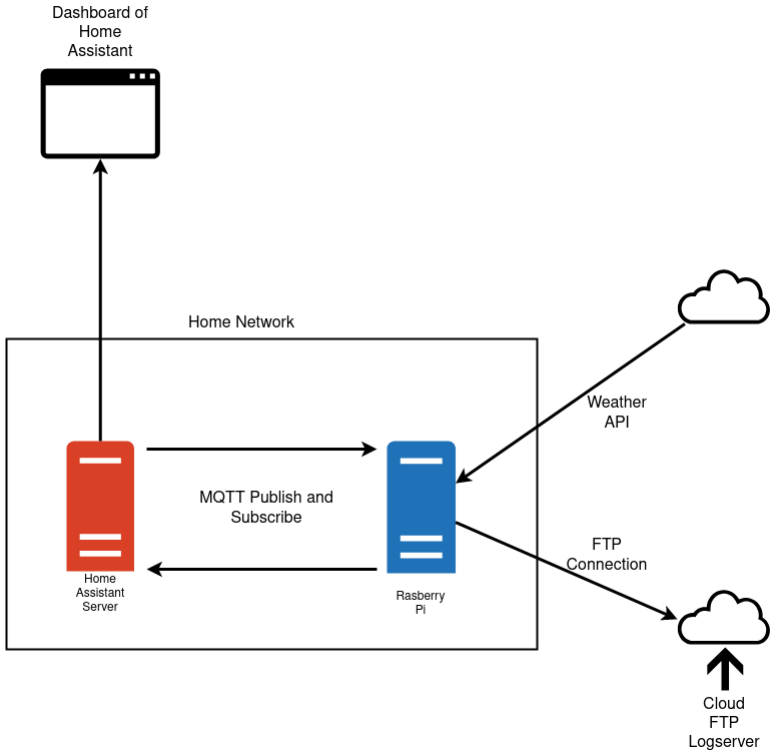
The relay is used to switch on and off the valve which requires a higher voltage and current then the raspberry pi can give it thus through the relay it connected to an external power supply, an alternative that could have been used was a MOSFET, this would allow me to switch the current on and off to the vale at the required voltage and current, and it has 0 moving parts making it more resistant to wear and tear. The reason a mosfet wasn't used is that my skills in electronics aren't as well developed and due to time constants connecting a prebuilt relay circuit was easier and still got the job done.

The moister sensor is used to test the level of moister in the soil and this sensor gives an analog reading which the raspberry pi cannot read. The raspberry pi can only work with digital signals and needs a special chip to convert the analog to a digital signal. In my care the sensor was connected to the pi directly anyway as if the signal was high enough the raspberry pi would interpret that as digital 1. Not the idle solution but one that works and is time efficient.

The magnet switch sensor is used to tell when the water level in the water tank is getting low, this gives a digital reading and work fine for my purpose, any other switch would have worked but this was the easy and effective.

Design and implementation

Below in a diagram Showing the different connections that were used in the project



The Troubles faced during this project where limited to only a couple of notable times, one of these was getting home assistant running on an old laptop that I had laying around, I initially tried to run the home assistant OS but due to unknown reasons the laptop would not boot with this os installed and had to more to different approach on install Ubuntu server and Building home assistant on top of this. But after this home assistant worked perfectly with no trouble. The other issue was with cron and was finicky and took some trial and error to run reliably. No other issues were faced except for time management but that was due to me.

The next step for this project is to clean up the code, encloser and mounting solution for the various components. After this is would like to be able to access the dashboard from outside my local network and from my domain name and this are relatively easy and just requires time. This would help make the project more useful and user friendly.

Over View of code

One python script runs continuously listening for any publishes on any of the subscribed topics this then runs separate scripts depending on the message, if the message is water, it will run a script which handles the watering time. Another section is a script that every 5 seconds retrieves the current value from the sensors and publishes the data on certain topic to be picked up by home assistant. Another script is run by a cron job to check weather and the sensors and determines if the pants should be

watered this then runs the water script depending on if the script determines. This scrip will also run the ftp scrip which send all the logs to the ftp server.

References

Assistant, H. (n.d.). *Installation*. Home Assistant. Retrieved June 12, 2022, from https://www.home-assistant.io/installation/

Raspberry Pi Documentation. (n.d.). Raspberry Pi. Retrieved June 12, 2022, from

https://www.raspberrypi.com/documentation/