**Final Project**

**Garage Door Opener**

Alexander Reinhardt

Internet Of Things

[reinhara@iu.edu](mailto:reinhara@iu.edu)

2000099839

12/12/2021

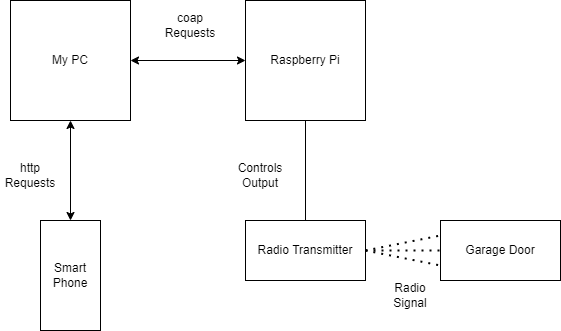
For my final project, my goal was to create a garage door opener that could be accessed from my phone. This would be accomplished by setting up a radio transmitter with my raspberry pi and creating an app that could interface with my raspberry pi. Through many pitfalls, I was able to find a solution that works well.

How it works:

A lot of what my project is, is the network I’ve created. First, a user will access the webapp I’ve created on their phone. From here they can send either a close or open request to an http server running on my desktop. This server then sends a coap request to my raspberry pi getting information from the various resources on the server and setting the state of the garage door if requested. If requested, the raspberry pi will activate a relay that will allow a current through the radio transmitter sending a signal to the garage door to either open or close. The server also keeps track of other miscellaneous data that the user might find useful, such as time since last activation or average uses per day.

The Network:

Below is the network map for my project.

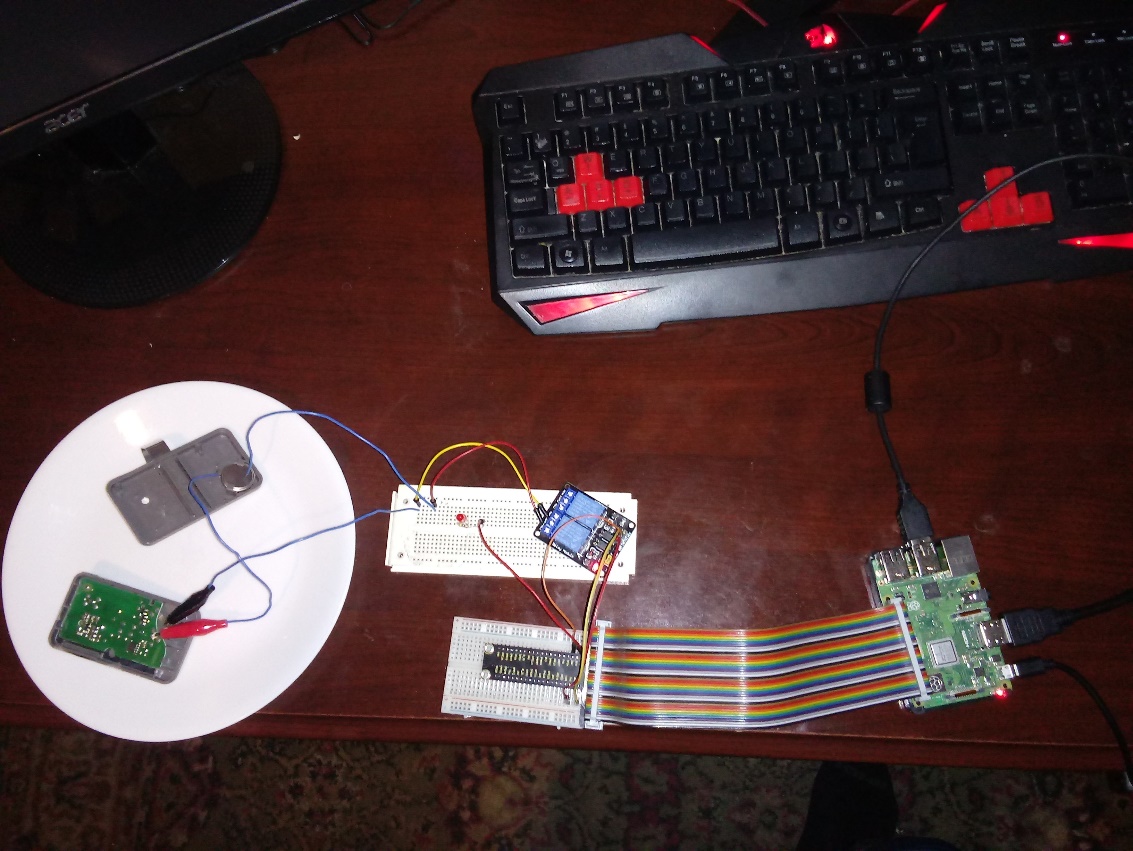
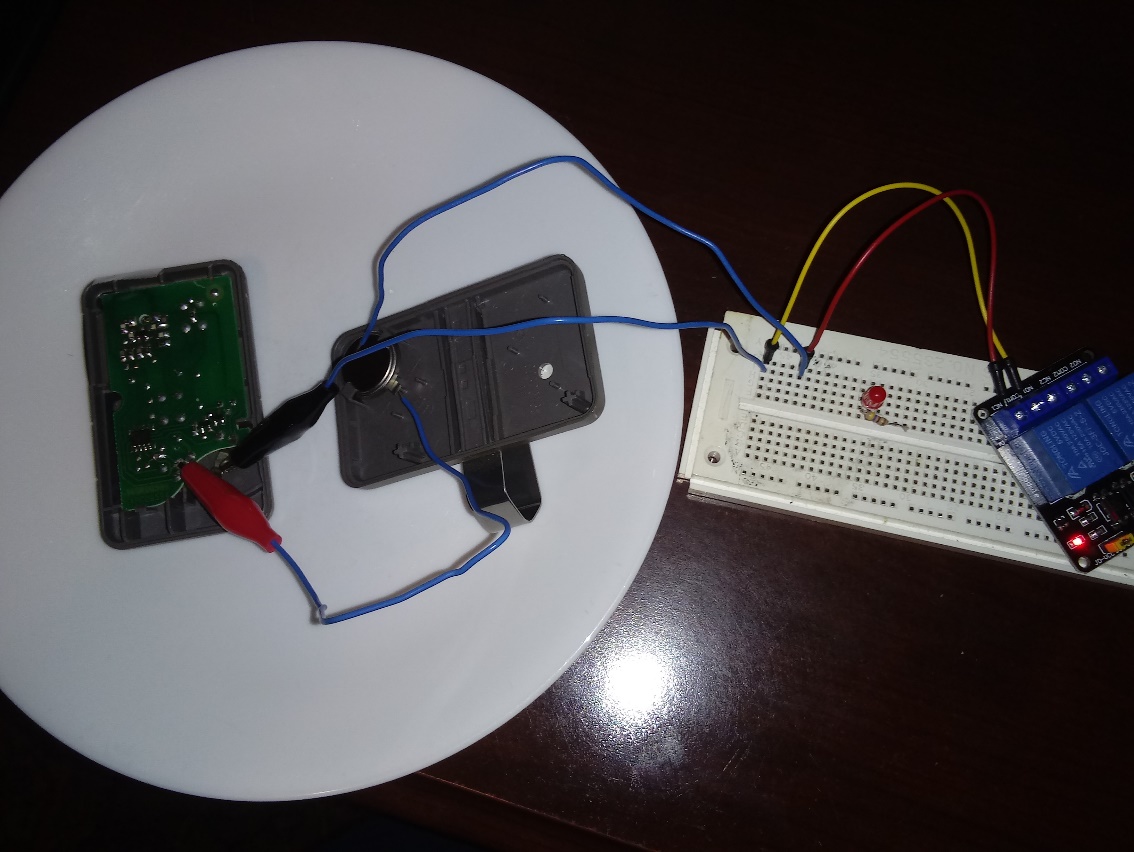


From this image, the network design can clearly be seen. I run the app on my smartphone which connects to the server on my pc. The phone will send http requests to the server and the server will send coap requests to the raspberry pi to collect data or the operate the garage door. The raspberry pi controls the radio transmitter which sends a radio signal to the garage door to either open or close it.

Circuit design:



This image clearly displays how the pi, relay, and transmitter are hooked up.

Here are some images of the circuit. 

What I used:

The node I used for this project was a raspberry pi 3 B+. This was connected to a simple double relay that controlled the current through the radio transmitter. The radio transmitter is a repurposed garage door opener remote. Since the remote activates by button press, I simply compress the button and delegate control of the transmitter to the relay. The transmitter will emit a 315 MGhz radio signal with a specific code that the garage door picks up. I also created an http server for this project for use with my web app using Python’s Putty library.

The App:

The app is a simple interface that lets a user either close or open the door. The app will also display the state of the door, the time since last action, and the average uses per day.

Conclusion/Pitfalls:

There were many problems I had when creating this project. In my initial design, I had the raspberry pi sending out the radio signal itself. This proved to be unusable as the raspberry pi can only output an up to 250 MGhz radio signal while the garage door requires 315 MGhz. There was also the problem of the garage door requiring a special code on this radio channel for security reasons. I then found the remote that was programmed to work with the garage. The first issue I had to overcome was to make sure the current of the device wasn’t too high; if it was, it could have destroyed my pi. I checked and it has a resistance of 22 k ohms approximately. Since it used a 3V battery, I calculated that the current would be about .13 mA, which is perfectly safe for the pi. I initially tried to hook the remote up directly to the pi, but this didn’t work as the volage from the pi is 3.3V while the voltage used by the remote was 3v. This caused me to include the relay, which would simply allow a current through when the pi told it to. This was my final solution.

On the network side of things, I also had problems. My first challenge was to create an app that could run on my smartphone, but that could communicate with the raspberry pi server. It would be difficult to run python on my phone (the language my pi coap server was using), so I came up with the idea of making a second server running on my computer that could be connected to using any web browser, essentially creating a web app. I used putty to create a python web server that would communicate with the raspberry pi. This solution worked perfectly.