

Group 12: "The Internet Explorers"

Logan Whitmire (lwhit@vt.edu)

Alex Tejada (alexart@vt.edu)

Neil Singh (neils@vt.edu)

Austin Lamicella (lustin@vt.edu)

ECE 4564, Assignment 3

1. Objectives

The purpose of this assignment was to communicate with an openHAB home automation system in order to control and get the status of IOT connected devices. Two RPi's are involved in this assignment, a Client RPi, and an openHAB RPi.

The Client RPi receives various commands from the terminal and sent those commands to the openHAB RPi. These commands are specific API requests that are able to get and put data on the openHAB demo site using the RESTful interface.

The openHAB RPi received requests from either the Client RPi, or cURL commands from a user laptop. These commands were then sent to the openHAB demo house in order to access various items on the openHAB bus. The openHAB RPi also controlled a GPIO LED. This LED was configured to the openHAB demo house, and could be turned on and off using specific commands.

We approached this assignment by first configuring the openHAB demo house, and expanding from there. Understanding openHAB was the most important part of the assignment. Once we had enough knowledge about openHAB, we then created the demo house, which ran on the openHAB RPi. From there, we worked to configure the GPIO LED. Once GPIO was working, we then sent cURL commands to the demo house. Finally, we implemented the Client RPi.

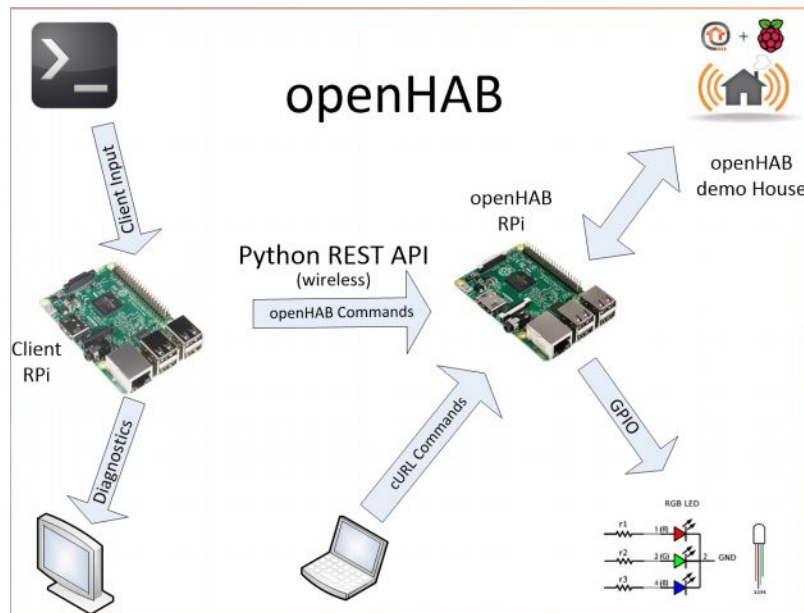
To help with our integration of the openHAB RPi we used:

<https://mcuoneclipse.com/2015/12/23/installing-openhab-home-automation-on-raspberry-pi/>

To help with our integration of the Client RPi, we used:

<https://github.com/openhab/openhab/wiki/Samples-REST#python>

System Diagram:



cURL commands:

Get state of LED:

```
curl http://192.168.1.144:8080/rest/items/RaspiLED/state
```

Change state of LED:

```
curl --header "Content-Type: text/plain" --request POST --data "ON"
http://192.168.1.144:8080/rest/items/RaspiLED
```

Client RPi example commands:

```
python3 client.py -r POST -i RaspiLED -ip 192.168.1.144 -p 8080 -d ON
python3 client.py -r GET -i RaspiLED -ip 192.168.1.144 -p 8080
python3 client.py -r PUT -i Light_FF_Bath_Ceiling -ip 192.168.1.144 -p 8080 -d OFF
python3 client.py -r GET -i Light_FF_Bath_Ceiling -ip 192.168.1.144 -p 8080
```

2. Responsibilities

Logan worked to get the openHAB server up and running. He also aided in the configuration of the openHAB demo site.

Austin developed the correct cURL commands necessary in order to communicate with the openHAB demo site. He also helped with the Client RPi.

Neil worked to get the openHAB server up and running, and worked to initiate GPIO connection to the openHAB RPi. In addition to this, he also configured the Client RPi to receive command input.

Alex was there for emotional support.

3. Conclusions

Overall the project went very smoothly. Perhaps the biggest challenge was learning how to correctly implement openHAB via raspberry Pi. Once the demo site was set up, it was a matter of finding the necessary commands and syntax needed in order to access items on the openHAB bus.

This project was very useful in seeing how a smart home can be created. The biggest takeaway from the project was using the RESTful interface to send requests for data, and understanding a website's API's. Knowledge of this can be applied throughout many aspects of modern networking, and will be useful in future projects.