Dylan Cook

Senior Project

James Donahue

September 13th, 2018

Hass.io Installation

Once I had rounded out my idea for smart home automation I found that Hass.io had the best implementation of my idea. I started out wanting to create my own software that houses smart home devices but I have since changed my approach. I will now be developing an applet or extension for Hass.io that will utilize several smart home devices. A few ideas I have are an applet that uses location on your phone to communicate to your NEST and Hue lights. When you get within a certain distance of your house it will turn on lights for the rooms you will be entering and will set the temperature to preferred levels. This idea will help save costs on gas and electric bills by turning off air conditioning/heaters and lights when you are not within a distance to use them. It also creates the convenience of creating a user profile for specific configuration of temperature and lighting schemes.

My second idea for implementation of Hass.io applets is to have a users Hue lights flash with selected notifications. For example, if you set a timer and the timer runs out, the Hue light will flash to indicate it is completed. You can designate different colors for different notifications, i.e. blue for a facebook message, green for a text, red for an alarm. This will be very useful for people who are busy and do not always have their phone on them. It is also useful if you are listening to music or focusing on work. Instead of audio cues or looking at your phone for the notification you can identify what is happening by the lighting in the room.

To develop both of these applets I will need two different softwares. I will be using Hass.io and IFTTT. These two programs work very well together and give the developer a lot of room for configuration. “Home Assistant is a Python program, in simple words. It can be run by various operating systems and provide the ability to track, control and automate your devices. When people talk about Home Assistant they usually refer to a standalone installation method. Hass.io is a combination of Home Assistant and tools which allows one to run it easily on a Raspberry Pi and other platforms without setting up an operating system first. Hass.io is an all-in one-solution and has a management user interface that can be used from the Home Assistant frontend. This interface is not present in a standalone setup of Home Assistant. It is important to be aware that add-ons are only available in Hass.io, due to the way Hass.io is installed.”

IFTTT is pronounced like “gift” without the “g” and stands for “If this then that.” IFTTT allows users to create, store, and implement services, often called applets, for their devices. It pairs extremely well with Hass.io and allows you to make your devices communicate with each other through lines of code. There is a lot of different services and applets on IFTTT but I believe mine are unique in comparison to other similar ones.

I have a bit of history using IFTTT and fooled around with the GUI to get comfortable with using it. I have not had the time to develop anything yet but I am getting more comfortable using the software. I have also used other people's completed applets on IFTTT to make my devices communicate with each other. Next came the installation of Hass.io which proved itself to be much more challenging than assumed. When on my private network at home during a visit I was able to install Hass.io with no issues and even connected an Amazon Alexa and Nest that were both recognized by the Raspberry Pi. When I came back to school I re-implemented the install because you have to change the internet settings to the network you're on. This is where I ran into my issue.

William Jessups WiFi is very restrictive because they have so many devices on their network that they must protect. Unfortunately these restrictions infringe on what I need to install for Hass.io to work. When I first tried I noticed that it stalled the installation once it got to network settings in the boot process. I pulled the SD card and manually input the school’s IPv6 network information and re-ran the install. This did not prove to be fruitful as it stalled in the same place. From there I attempted to use my wireless hotspot on my phone but quickly realised I would have to run a hotspot everytime I needed to work on my project and that is not a cost I want to be responsible for. From there I attempted to plug ethernet into the Raspberry Pi during installation, again the same issue arose. Lastly I attempted to get an ethernet switch so that I could have both my Raspberry Pi and my desktop on the same network because once Hass.io runs you access it through a local IP on your desktop. I ran the install and found my Pi’s IP on Fing but ran into the same issue.

My solution for this is to talk to the network administrator in IT on campus. I went to speak with him because we work together but in God’s perfect timing the network admin went on vacation for a week. He has returned and I will be having a conversation with him this week about the possibility of allowing certain things through the network or if he has a suggestion for another solution. In the meantime I will be playing around with IFTTT more, I just won't be able to test my code because I will not connect to Hass.io until I talk with the Network Administrator