Configuring Mycroft on Raspberry Pi 3

Lumbini Parnas

February 14, 2018

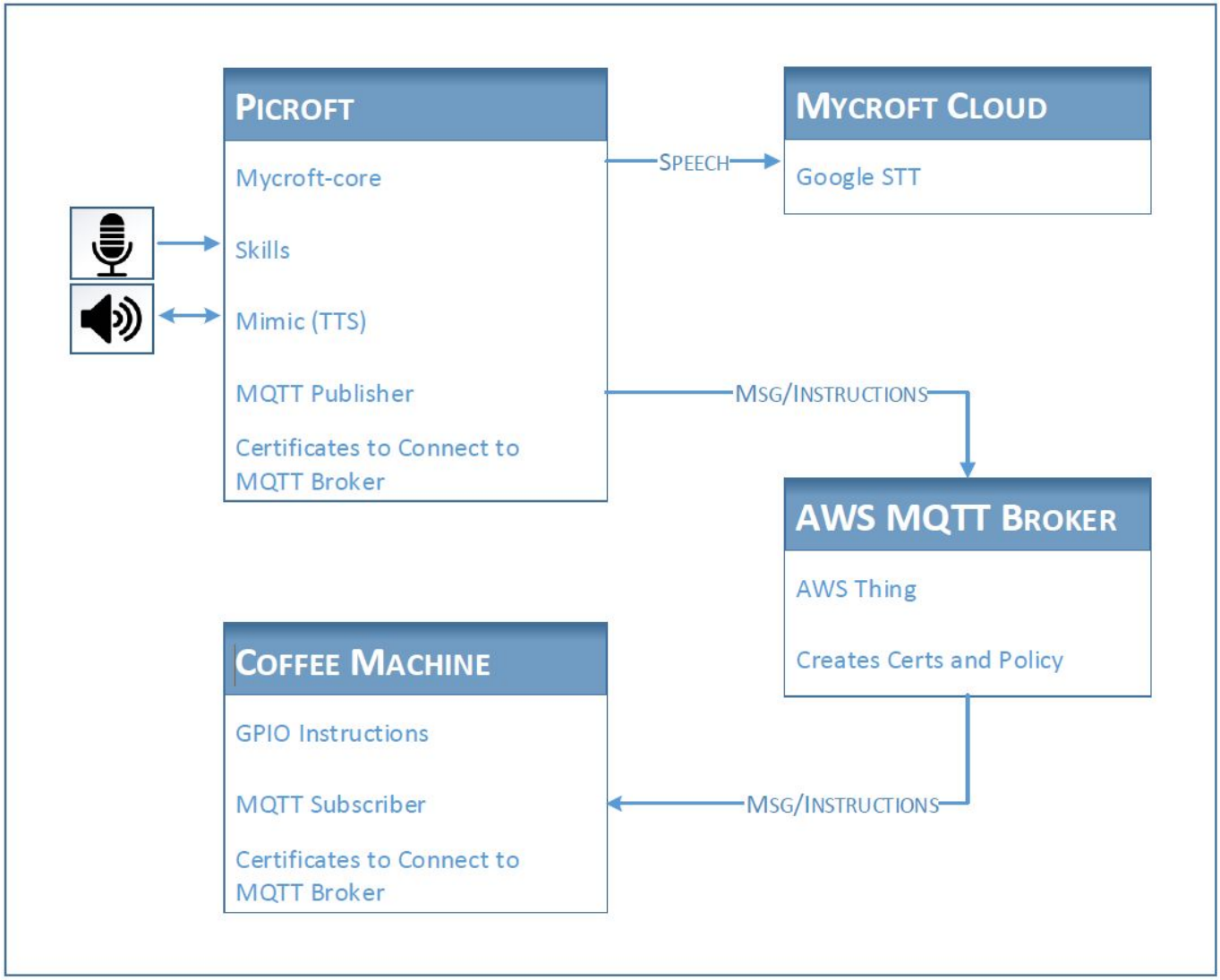
Contents

1 Introduction

Mycroft is a software suite that uses natural language processing and machine learning to make up an open source voice assistant. The software can be run on supported devices like the Raspberry Pi 3. As the software grows, it should be able to support a wide range of platforms and devices - called enclosures. The following project uses an enclosure of Mycroft called Picroft, designed to run on the Raspberry Pi 3 (or newer). Picroft is used to voice control a pseudo-coffee machine with basic commands to develop a Mycroft based IoT system. The speech-to-text and text-to-speech processes take place in the Mycroft cloud which by default uses Google STT and mimic. MQTT is used to establish a connection between Mycroft and the IoT for which Amazon AWS acts as the MQTT broker. The following documentation describes the process of configuring Mycroft on the Raspberry Pi 3 and establish a working model for the IoT.
Additional information about Mycroft can be found here: https://mycroft.ai/documentation/

2 Basic Architecture

Figure 1: Architecture



3 Components

Some of the required components for this project are:

- Hardware Pi
  - Raspberry Pi 3
  - Micro-usb power adapter
  - Micro SD card (8GB or higher)
  - An analog Speaker that can be plugged into the 3.5mm audio jack on the RPi 3
  - USB Microphone
  - Monitor (Optional)
  - Keyboard (Optional)
- Hardware Pseudo-coffeemachine
  - Breadboard
  - 4 LEDs
  - 4 Resistors
  - Wires
- Software
  - AWS Account
  - AWS thing
  - Etcher (or equivalent to burn SD card)
  - Download Picroft
  - Python 2,7
  - paho-mqtt

4 Software Setup

4.1 Picroft Setup

Please refer to https://mycroft.ai/documentation/picroft/ for any updates in the setup process. The Raspberry Pi can either be ssh-ed into or a monitor and keyboard can be connected to access the terminal for any debugging purposes.

1. **Download the disk image** from here: http://bit.ly/2o5S3v9

2. **Burn disk image to micro SD card** using Etcher: https://etcher.io/
   Download Etcher and install it. Connect SD card reader with SD card in it to your computer. Open etcher and select the disk image you downloaded in step 1. Select the SD card and hit 'Flash!' to begin writing the data to SD card. It might take a few minutes.

3. Refer to the Hardware Setup section to first build the coffee mahcine before continuing the software setup.

4. **Booting up Picroft.** Once the SD card is done, insert it into the Raspberry Pi. Plug in the microphone and speakers/headphones. If you are using a monitor and keyboard, plug them in too. Finally, plug the power adapter and switch the power on.

5. **Connecting Picroft to the Internet.** **[Wifi]** On your computer or a mobile device, connect to the Wifi SSID *MYCROFT* using the password *12345678*. Once connected to the SSID, go to the web page http://start.mycroft.ai. A list of available WiFi networks will be presented. Select the WiFi network that you wish to connect the Picroft to, and enter the WiFi password. Picroft will attempt to connect to the WiFi network.

6. **Pairing Picroft** Once connected to the Internet, Mycroft will speak a registration code. Go to https://home.mycroft.ai/ and under devices, register your device using the registration code. Once paired, basic skills can be used on the Pi.

7. **Connecting via SSH** SSH access ti Picroft is enabled by default. Ensure you know the IP address of the Picroft device on the network. The IP address of the device can be found by asking Mycroft,"Hey Mycroft, What is your IP address?" to which you will get a reply with the wlan IP address. Open a terminal on Linux or PuTTy on windows and type *ssh pi@IP_ADDRESS*. The default password is *mycroft*. Once successfully connected you can access everything on the Raspberry Pi.

4.2 AWS Thing Setup

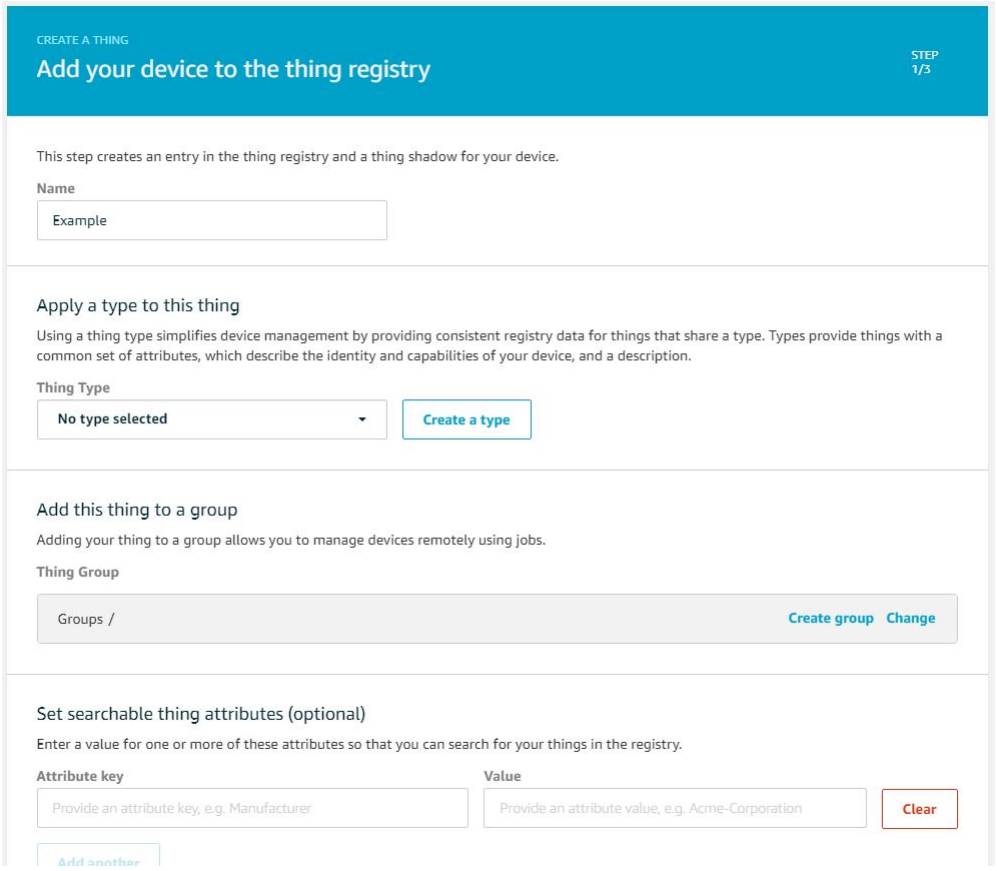
1. Make an AWS account and login if you do not already have one.

2. Navigate to AWS IoT page

3. Hit "Create a Thing" and follow the steps to create a shadow of the coffee machine on AWS. You just need to provide the name of the Thing. Figure 2

4. The next step asks to create certificates, click on the recommended link, "One-click Certificate creation". Figure 3

Figure 2: Create Thing



5. Download 4 certificates which you will be using to connect the IoT to the AWS MQTT broker. Figure 4

6. The next step is to create a policy for your Thing. Follow the step sin the image below. Figure 5

7. Once the policy is created, navigate to “secure” and click on “certificates” on the menu on the left. Figure 6

8. The next step is attaching the certificates to the policy. Click on the 3 dots on the certificate you created with your Thing and select ”Attach a Policy”. This will bring up the list of policies you created, attach the one you created for the Thing. After this, click on the same menu and hit ”Activate”. Figure 7

9. Make sure to download all the certificates for the code set up.

Figure 3: Create Certificates

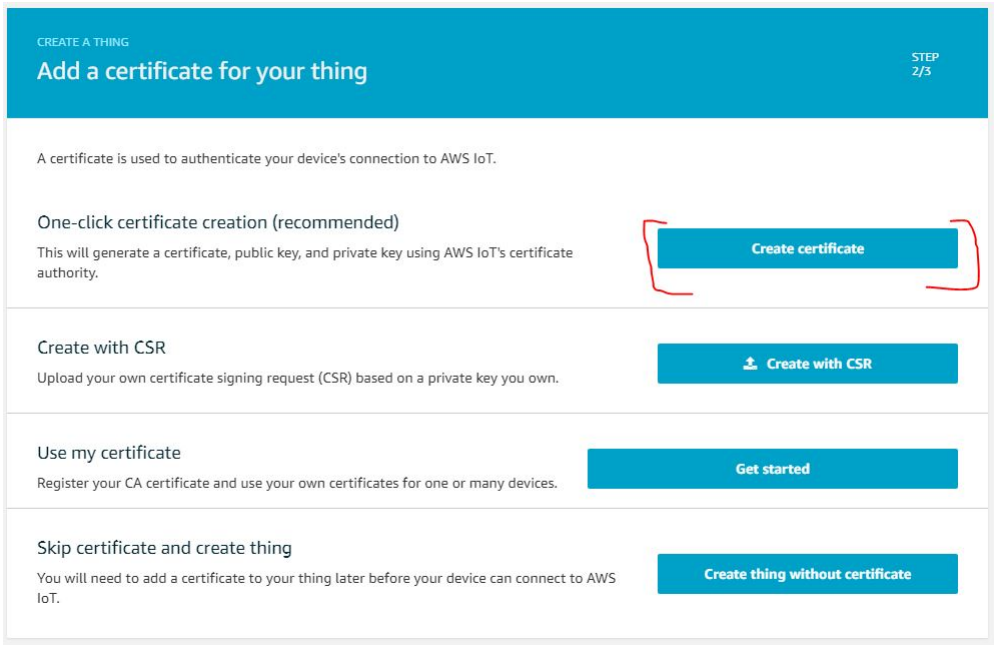


Figure 4: Download Certificates

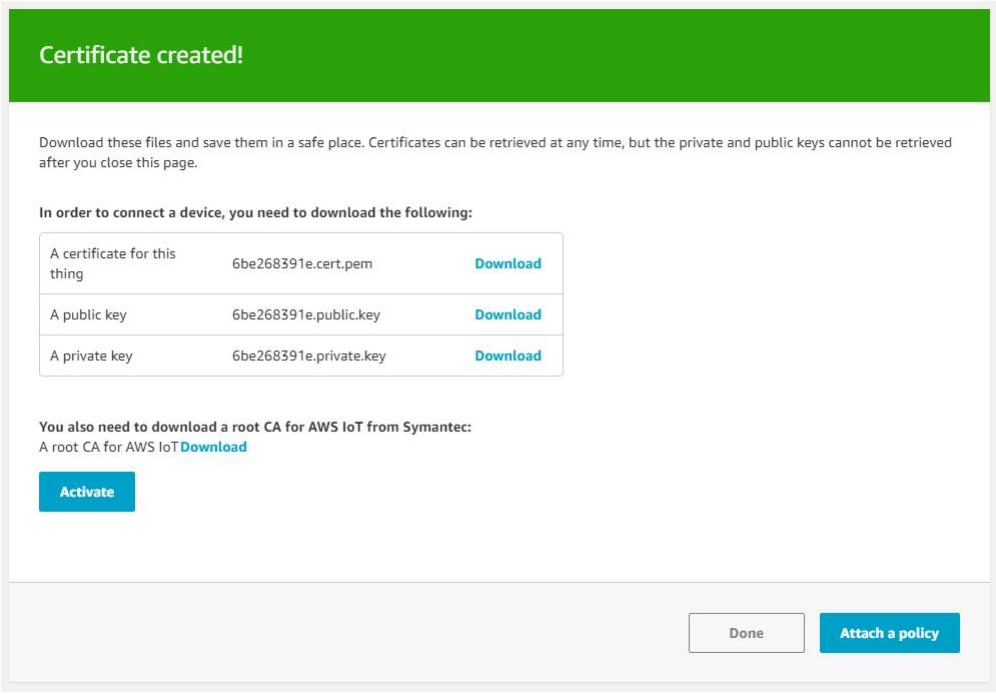


Figure 5: Create Policy

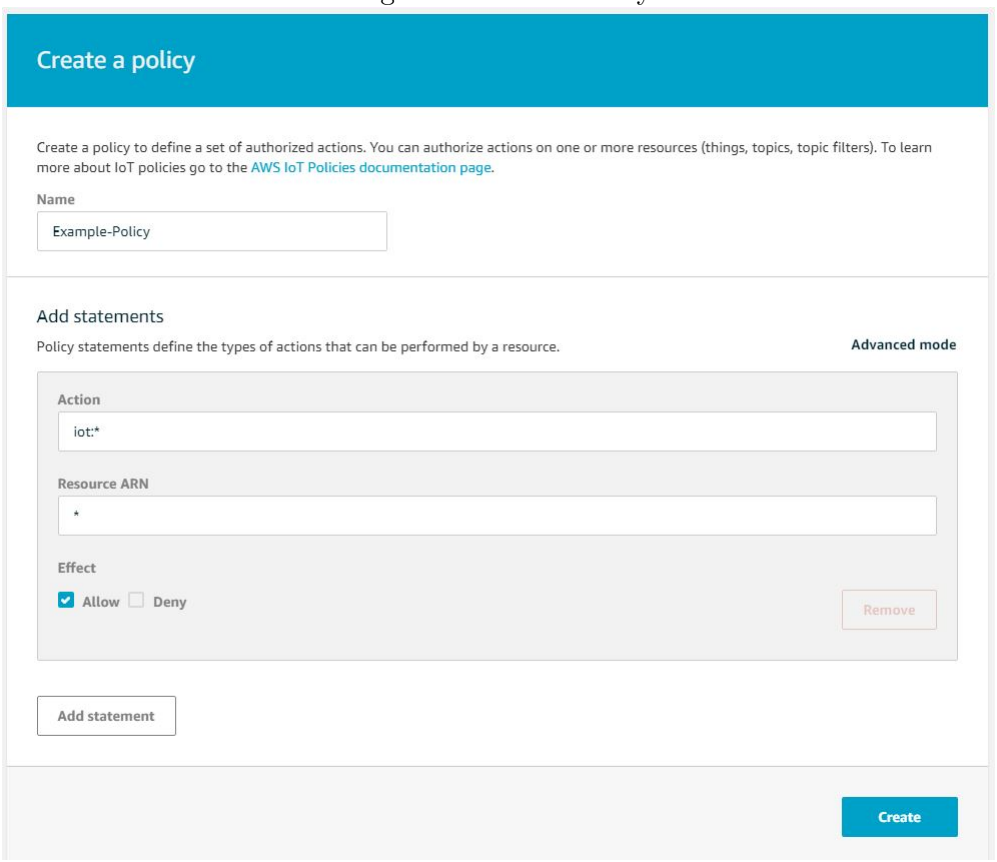
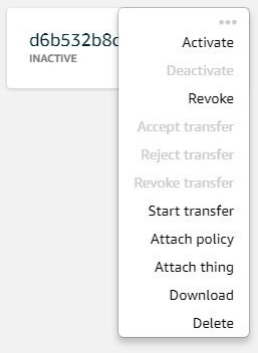


Figure 6: Menu

Figure 7: Attach Policy



4.3 Code and Certificates setup

A few changes will be made to the code in order for it to work with the certificates and the Thing you created. Please note that we will have 3 main parts of the software system, one on the AWS with consists of the Thing, two is the skill portion that runs on the Raspberry Pi through mycroft-core and third is the IoT part of the code that also runs on the Raspberry Pi but only controls the GPIO pins depending on what is received on the MQTT topic. Basically, you issue voice commands to mycroft, which activates skills - the coffee machine skill writes the required commands onto the MQTT topic which is then transferred to the IoT via AWS. The messages are pick up by the IoT using the MQTT subscriber on the IoT. The software set up needed to achieve this will be listed below.

- **Replacing the certificates** The code for this project will be found on the github repository with the link: *sudo git clone https://github.com/Lumbini/PicroftCoffee.git*. However, this repository has old certificates that will not work with the new Thing created. You can clone this project into your directory and change the certificates in all locations. Make sure to edit the *__init__.py* and *mqttSubscriber.py* files to match the information of the thing your created on AWS along with the correct certificates and paths.

- **Importing the Coffee Machine Skill.** This step will happen on the Raspberry Pi. The terminal on the Pi can be accessed either by connecting a monitor and keyboard or by SSH-ing into the device as shown in the previous step. You will need to transfer the skill folder along with the correct certificates into */opt/mycroft/skills*.

- **Editing permissions of skill** Once you transfer the skill directory, you will need to give the user write and execute permissions to the project. This can be done using, *sudo chmod u+wx PicroftCoffee*.

- **Running the coffee machine code** Open a new terminal on the Pi. This can be done using *ctrl+alt+F1..F4* any of the first few F buttons. cd into the PicroftCoffee folder and into the Run on Pi folder. Run the mqtt subscriber in the new terminal using *python mqttSubscriber.py*. Make sure the coffee machine hardware set up is ready before this step.

- Once you have the MQTT connection and Mycroft successfully running you will be able to issue sample tasks. Refer to "Running the Project" section for information on getting started.

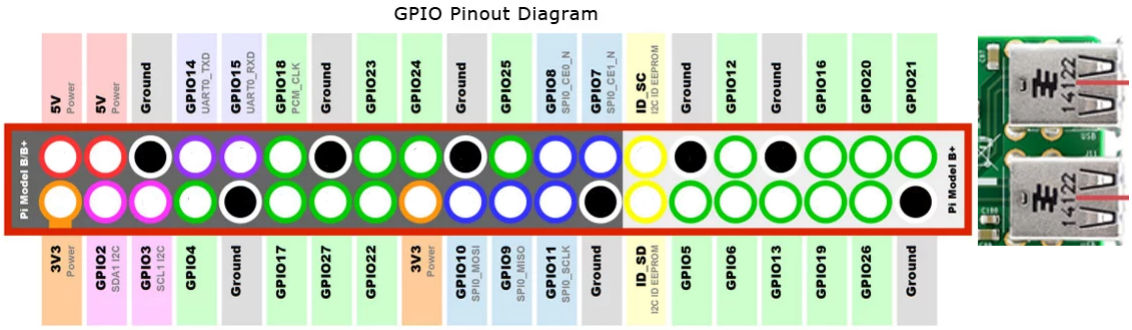
5 Hardware Setup

The Psuedo-coffeemachine used in this project is a series of LEDs triggered by different GPIO pins on the Raspberry Pi based on the command received on the MQTT topic. The following steps help to build the circuit. Please complete this step before turning on the Raspberry Pi.
You will need be building a simple circuit where the LED and resistor will be connected in series to the GPIO pins. Please follow the following circuit diagram to continue.

- LED 1 connected to PIN 21
- LED 2 connected to PIN 20
- LED 3 connected to PIN 16
- LED 4 connected to PIN 19
- Any of the ground pins

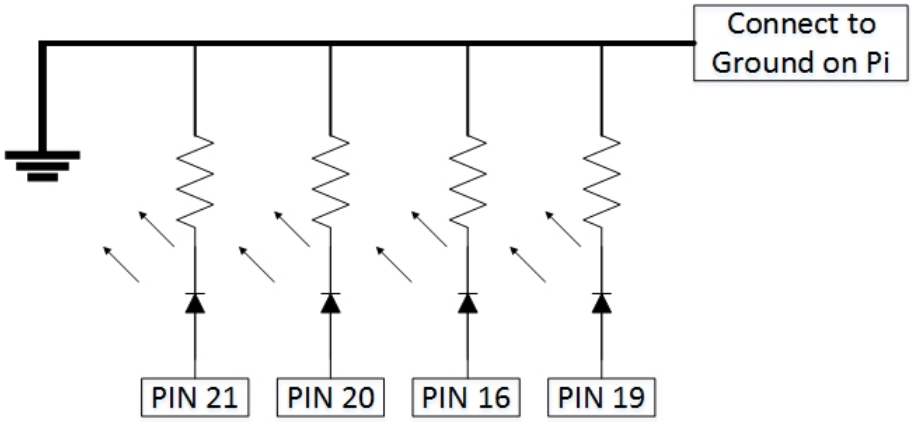
5.1 Raspberry Pi 3 pinout

Figure 8: Raspberry Pi 3 pinout



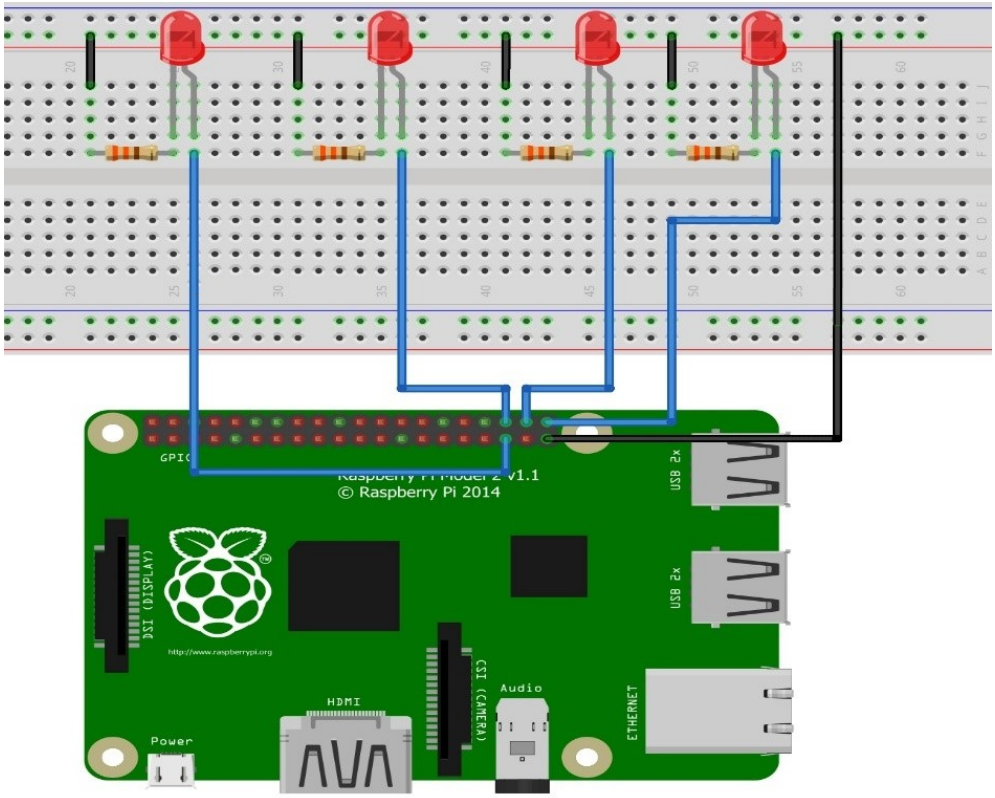
5.2 Circuit Diagram

Figure 9: Circuit Diagram for the LEDs



5.3 Sample setup

Figure 10: Example of circuit



6 Running the Project

- Turn on the raspberry pi
- Make sure speaker and microphone are on
- SSH into the Raspberry Pi
- In one terminal you should have the Mycroft cli client running, if not you can start it using *mycroft-cli-client*. Other useful commands can be found in the README inside the root folder on the Raspberry Pi device.
- Open a second terminal on the Pi and run the mqtt subscriber. *python mqttSubscriber.py*
- Issue the "Hey Mycroft" wake command.
- Example of a conversation:
  - Hey Mycroft, Switch on the Coffee Machine
  - **Response:** *The Coffee Machine is On*
  - Hey Mycroft, Brew me a cappuccino.
  - **Response:** *The Coffee Machine will brew you a cappuccino.*
  - Hey Mycroft, cancel the cappuccino.
  - **Response:** *The Coffee Machine will cancel your cappuccino.*
  - Hey Mycroft, shutdown the Coffee Machine.
  - **Response:** *Goodbye.*

7 Known Issues

- Need to say "Hey Mycroft" before every command. This can be solved by making mycroft interactive using the *get_response()* function that is available in the API.
- "Turn on" and "Turn off" do not work whereas, "Switch On", "Start", "Shutdown" and "Switch off" do work for machine control. This is probably due to a duplicate utterance with a different skill. Need to be debugged.
- Need to be very explicit with the instructions and make the intents provided in the skill code-base
- Mycroft is at a level of maturity where developers and hardware hobbyists will be able to use it effectively. However, it is not yet ready for mainstream adoption. (Feb 2018)

8 Useful tools

- Mycroft API: http://mycroft-core.readthedocs.io/en/stable/source/mycroft.html
- Mycroft Documentation: https://mycroft.ai/documentation/

- Subscribe to Mycroft's slack channel on: chat.mycroft.ai Here's is where most developers answer questions in real-time. If not developers Mycroft users and fellow enthusiasts will be available to help with debugging issues.
- Setting static IP on raspberry pi: https://www.modmypi.com/blog/how-to-give-your-raspberry-pi-a-static-ip-address-update