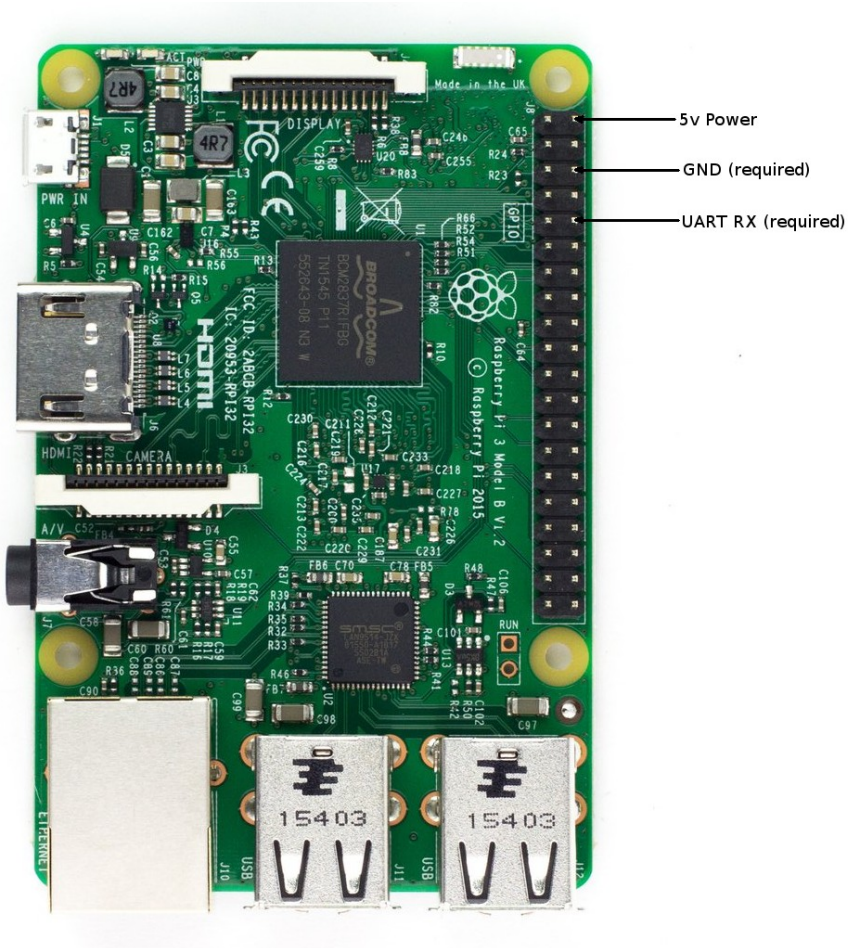
Connecting Picaxe to Cayenne using Raspberry Pi and MQTT.

Abstract:

Picaxe can be used as a cheap way to collect data from a variety of places, however viewing that data is not always the easiest task. This document with accompanying scripts explains how to connect a picaxe receiver to a Raspberry Pi and then use the provided scripts with minimal configuration to upload the data to Cayenne where it can be viewed online in real time. The collecting of the data is beyond the scope of this document, as is the transmission from collecting picaxe to receiver picaxe.

Setup:

The Picaxe needs to be connected to the Raspberry Pi's GND and UART RX pins. It could also be connected to the 5V power pin so that the picaxe doesn't need to rely on batteries. Connect picaxe to shown pins:



The picaxe must be programmed to send data to the UART RX pin using using the T2400 mode.

The data string must look like this:

```
Instrument X	Reading Y
```

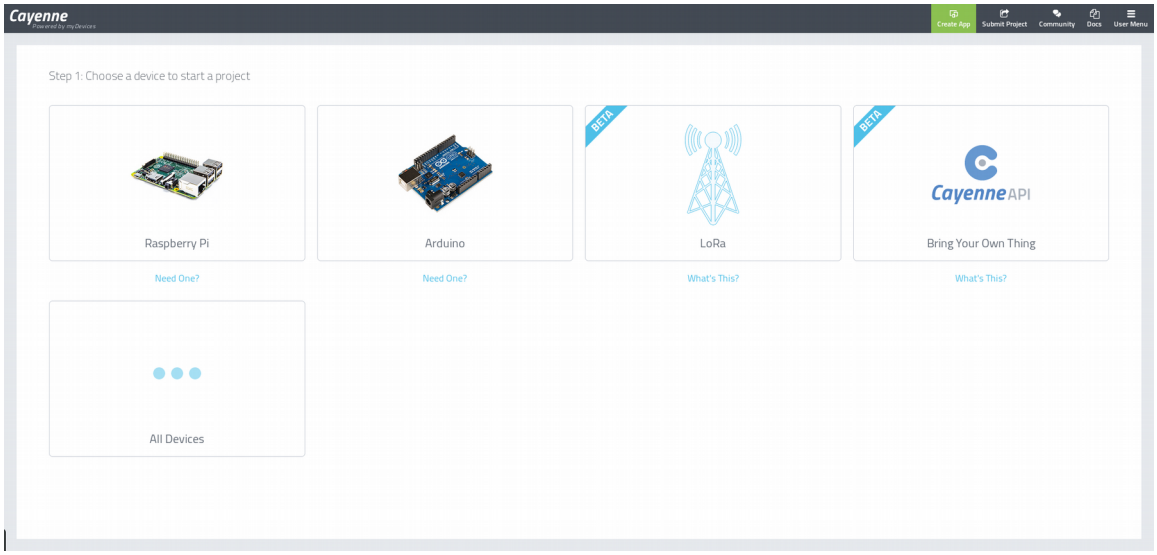
Where X is the Instrument id and Y is the reading it has detected. The long space is a tab in (\t, ASCII code 9) and is used as the delimiter.

Other debugging data may be including and will be ignored by the scripts provided, assuming they are separated but the delimiter.

See the online instructions for creating a bootable SD card with Raspberrian.

https://www.raspberrypi.org/downloads/raspbian/

Go to https://mydevices.com and signup (or login) Select *CayenneAPI*



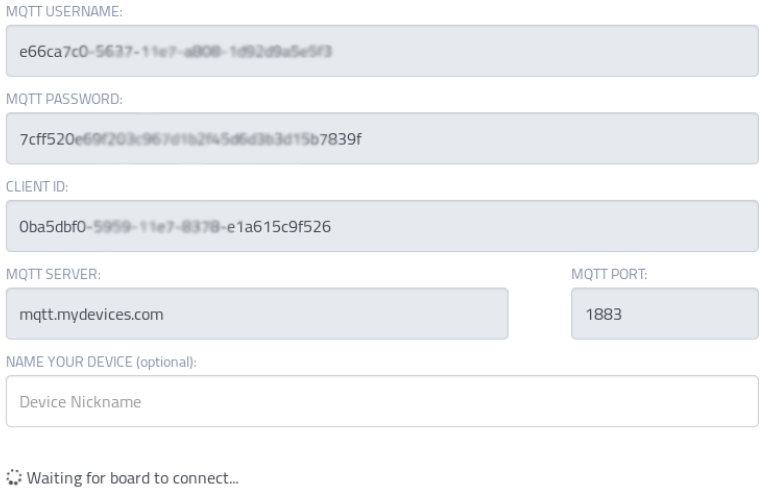
Boot the Raspberry Pi and select the terminal application from the menu. Type the follow command:

```
git clone https://github.com/green0range/Uploading_picaxe_data_with_cayenne.git
cd Uploading_picaxe_data_with_cayenne
sudo ./setup_helper.sh
```

It will ask if you want to setup the UART port. If you are using a Raspberry Pi 3, then type ‘y’ otherwise type ‘n’ (See page 3)

Next it will prompt you for your login information. This is the information you see after clicking on the Cayenne API above. (See right)

Note: It is not you email username and password used to login to Cayenne.



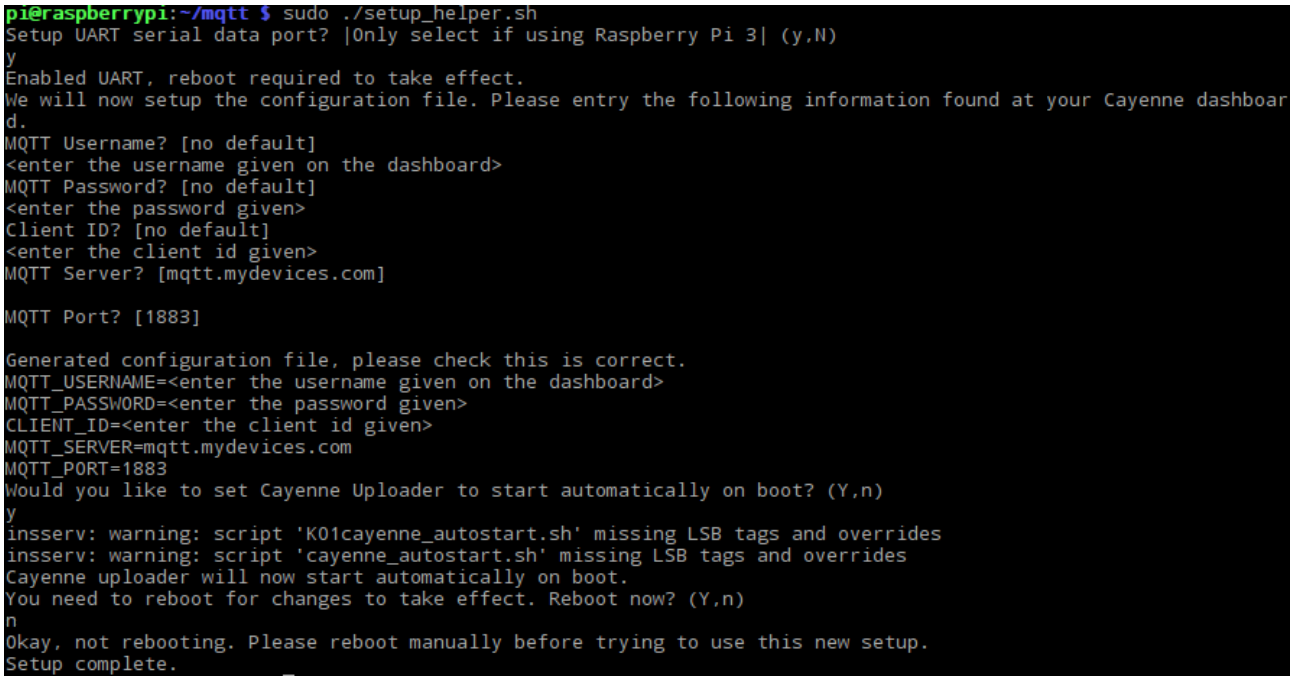
You do not need to enter the server and port as they are the same for everyone. Just press enter without typing anything to use the preset defaults.

Next it will download and install the cayenne MQTT library, this will happen automatically.

It will then ask you if you would like to have the uploader start automatically. Selecting ‘y’ is recommended.

If you setup the UART port then you will be asked to reboot. If you didn’t the script will start up the uploader and you can go to cayenne to check on it.

Here is what an example setup looks like:



Advanced Options:

To model the data according to some mathematical formula you can edit the interpret_raw_data function in bore_reader.py

The raw data is the *data* variable. Once completed return the corrected data to the *corrected_data* variable. Minimal python knowledge required.

Explanations:

The Raspberry Pi 3 requires addition setup to use the Serial / UART pins. This is because the previously main UART configuration was used to accommodate for built in BlueTooth, so a secondary mini UART system must be used. The fixes provided by the script set the clock used by the mini UART to a fixed rate and enable it. The modifications are made to /boot/config.txt and require a reboot

to take effect.

Port 1883 must be open for the Raspberry Pi to communicate with Cayenne. I have created a work around system but it doesn't work very well so it is best just to open port 1883.

The Raspberry Pi will also create a CSV inside it's working directory encase Cayenne stops working. This contains all the same data that is uploaded to Cayenne. I would be possible to sync this file via file syncing applications and view data that way bypassing Cayenne if needed.

References:

Milliways (2017) *How do I make serial work on the Raspberry Pi3 – Raspberry Pi Stack Exchange*, available:https://raspberrypi.stackexchange.com/questions/45570/how-do-i-make-serial-work-on-the-raspberry-pi3/45571#45571, accessed 25-06-2017

myDevicesIoT (2017) *Cayenne MQTT Python Library Readme*, available:https://github.com/myDevicesIoT/Cayenne-MQTT-Python/blob/master/README.rst accessed 25-06-2017

[Image of Raspberry Pi] Shopify, *Raspberry Pi 3* available:https://is.gd/DWb6gJ accessed 25-06-2017