How to connect Tinkerforge sensors to Cayenne IOT dashboard via MQTT on PI

I had a Raspberry PI Model B since some time and wanted to monitor the environment of a small rack.

As I am not an electronic guy nor a programmer, I wanted an easy way to connect sensors without soldering.

Found the Tinkerforge concept interesting and build on it.

Then I saw the Cayenne IOT dashboard and tried to combine the two via MQTT protocol.

Thanks to the help of both Tinkerforge and Cayenne forums, here is how I did it.

I detailed this doc as far as I could, so that someone else with as little knowledge as me could replicate it.

The text in blue is the text you type at the pi prompt or some hyperlink to a download or doc.

Hope it can help you.

List of parts:

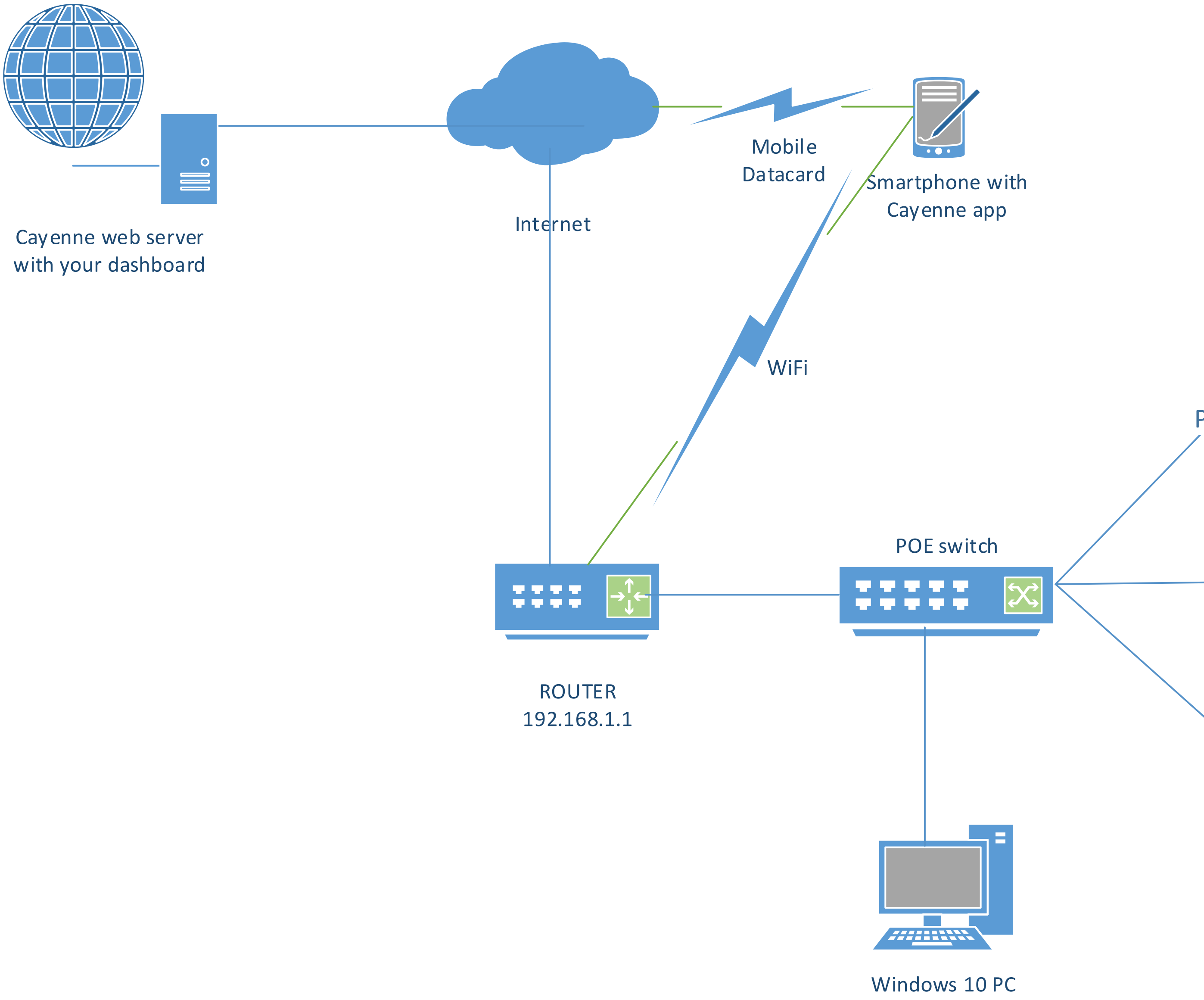
1x raspberry PI model B

1x switch with POE ports with Ethernet cables

Tinkerforge:

- 1x Master Brick
- 1x Ethernet Master Extension
- 1x Temperature Bricklet
- 1x Humidity Bricklet
- 1x Server Room Monitoring Case
- 3x Bricklet Cable Black 15cm
- 1x Bricklet Cable Black 50cm
- 6x Mounting Kit 12mm

Diagram



Step 1 installing and configuring Jessie on PI:

Install raspbian Jessie lite from https://www.raspberrypi.org/downloads/raspbian/

See https://www.raspberrypi.org/documentation/installation/installing-images/windows.md how to do it.

Also install putty (http://www.putty.org/) on your pc

After loaded the raspbian image on the SD card, start the pi

IP is normally the first free dhcp lease from your router or via terminal command “ ifconfig “ if you have a screen connected to your PI

Easiest way is to have an usb keyboard and an hdmi screen connected to your PI.

Default login is pi and password is raspberry

configure the config file:

`sudo raspi-config` or use the PI menu, preference, Rasperberry Configuration on the screen

Change the start config:

-1expand file system : to use all the sd card memory

-2change password to pi (easier to type) or something else

-4 international option: # I live in Belgium so set the stings for your country

local : keep EN GB as default otherwise you will find some Russian texts

timezone: europe, brussels #put your timezone

keyboard: logitech cordless desktop, other, Belgian, first belgian, the default, no compose key,yes. #put your keyboard settings

-8 advanced settings

A2 hostname : pi (or something else)

A3 memory split :32 or lower (no screen once configured, so all memory to system)

A4 SSH enable

finish and reboot

Configure static IP instead of DHCP:

Use putty or screen to login with Pi and password you changed just above.

At prompt `sudo nano /etc/dhcpcd.conf`

At end of file add the fixed ip #replace 192.168.1.20 with your IP

```
# A hook script is provided to lookup the hostname if not set by the DHCP
# server, but it should not be run by default.
nohook lookup-hostname

interface eth0
static ip_address=192.168.1.20/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Ctrl x to exit, y(es) to save modified buffer and click “enter” to finish

After reboot open putty with ip 192.168.1.20 with pi and raspberry or your password if you changed it

Update/upgrade packages:

```
sudo apt-get update
sudo apt-get upgrade
```

Step2 connect the Tinkerforge brick and bricklets

In my case I have 3 bricklets (temperature, humidity and barometer) connected to a master brick with master Ethernet with POE extension

(see info https://www.tinkerforge.com/en/doc/) . Connect the bricklets to the master brick with the appropriate cables. Stack the POE Ethernet master extension on top of the master brick and connect your network cable to a POE port on your switch.

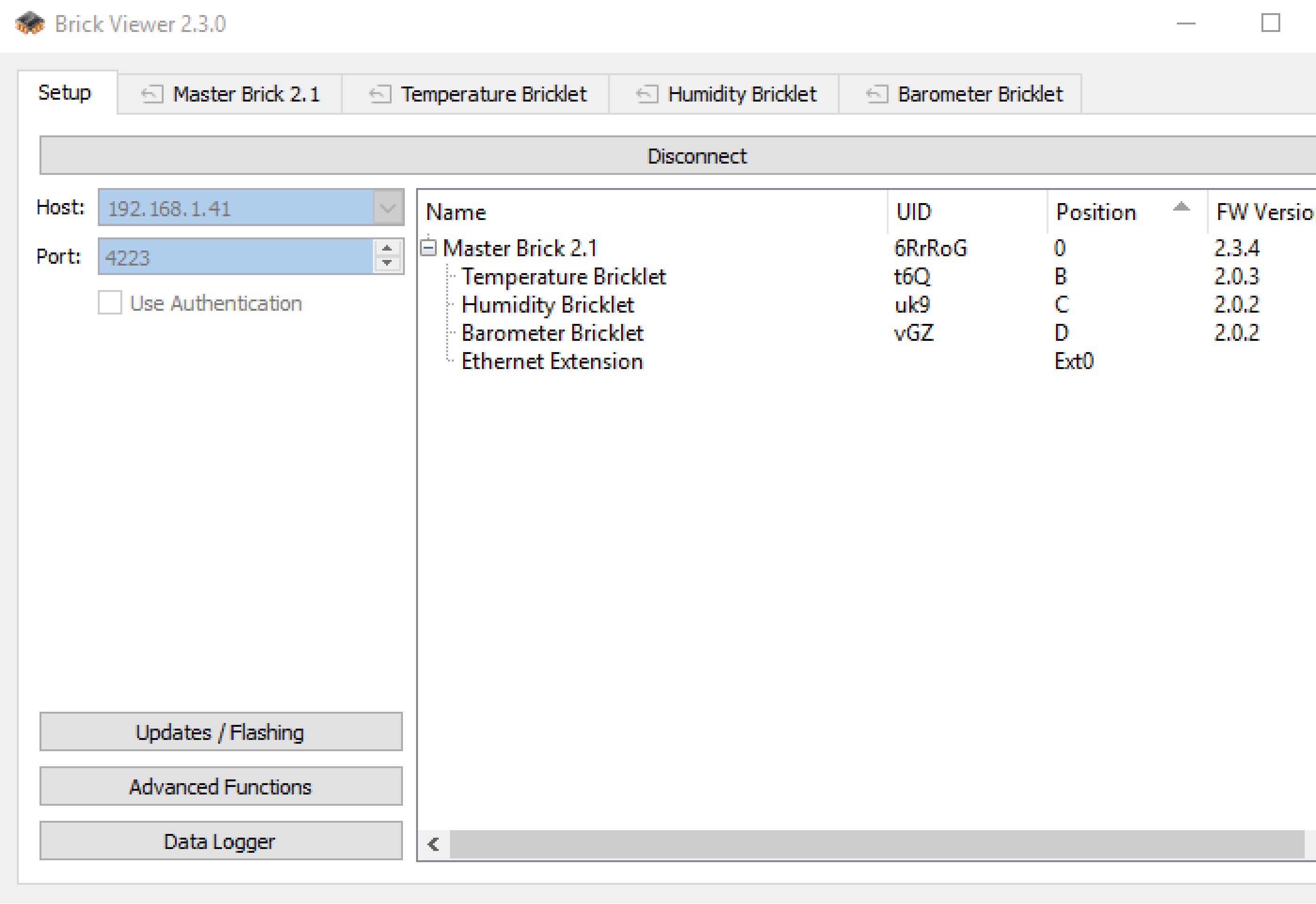
Configure the IP address see https://www.tinkerforge.com/en/doc/Hardware/Master_Extensions/Ethernet_Extension.html#ethernet-configuration

For the first connect, you will have to find IP address of the brick the DHCP lease in your router.

Change connection from DHCP to Static. Put your IP (my case 192.168.1.41), subnet mask, gateway and save Ethernet configuration.

Install the brick deamon https://www.tinkerforge.com/en/doc/Software/Brickd.html and the brick viewer https://www.tinkerforge.com/en/doc/Software/Brickv.html on your pc.

Open the brick viewer and connect to the host, in my case 192.168.1.41 and port 4223 (default)



Host: 192.168.1.41

Port: 4223

□ Use Authentication

Name	UID	Position	FW Versio
Master Brick 2.1	6RrRoG	0	2.3.4
Temperature Bricklet	t6Q	B	2.0.3
Humidity Bricklet	uk9	C	2.0.2
Barometer Bricklet	vGZ	D	2.0.2
Ethernet Extension		Ext0	

Here you can see the UID of the 3 bricklets (case sensitive !!!)

Temperature = t6Q

Humidity = uk9

Barometer = vGZ

Take note of those values, you will need them later.

Step 3 install the Tinkerforge MQTT proxy

Thanks to “rwblinn” from the Tinkerforge forum

Open putty with ip 192.168.1.20 with pi and raspberry or your password if you changed it

```
sudo apt-get install python3-pip
sudo apt-get install python-pip
sudo pip install tinkerforge paho-mqtt
sudo apt-get install mosquitto mosquitto-clients python-mosquitto
sudo apt-get install libusb-1.0-0 libudev0 pm-utils
mkdir /home/pi/tinkerforge
```

```
cd /home/pi/tinkerforge
wget http://download.tinkerforge.com/tools/brickd/linux/brickd_linux_latest_armhf.deb
sudo dpkg -i brickd_linux_latest_armhf.deb
rm brickd_linux_latest_armhf.deb      #delete the file brickd_linux_latest_armhf.deb:
wget https://raw.githubusercontent.com/Tinkerforge/brick-mqtt-proxy/master/brick-mqtt-proxy.py
```

This last command is to manually start tinkerforge MQTT proxy.

```
python brick-mqtt-proxy.py --brickd-host 192.168.1.41 --brickd-port 4223 --broker-host localhost --broker-port 1883 --update-interval 5&
```

info about this command:

- my brick IP is 192.168.1.41, replace it with your brick address
- the interval 5 is the number of seconds between 2 mqtt messages, replace 5 with your interval, but do not forget the & at the end. Otherwise you could have problems.
- You can replace localhost with ip rapsberry broker/server
- Type sudo python brick-mqtt-proxy.py –help for more info

This last command has to be rerun in the /home/pi/tinkerforge directory of your PI after every reboot. To run automatically after every reboot, add this line with your values at the end of the existing crontab file with:

```
sudo crontab -e
```

```
@reboot python /home/pi/tinkerforge/brick-mqtt-proxy.py --brickd-host 192.168.1.41 --brickd-port 4223 --broker-host localhost --broker-port 1883 --update-interval 5&
```

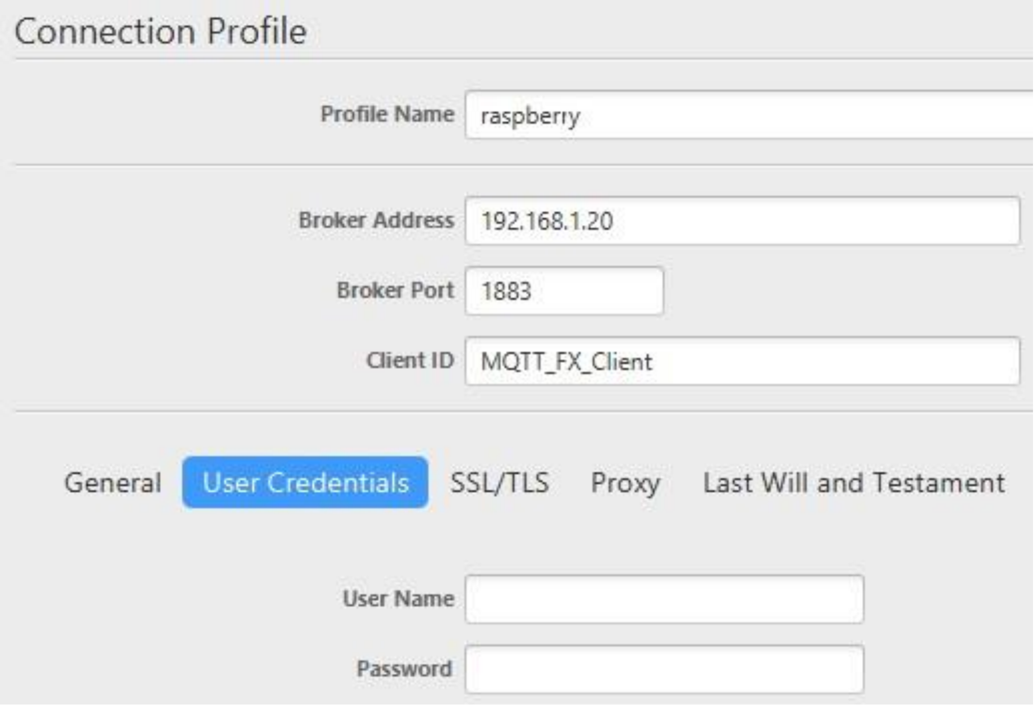
Save the file with Ctrl x , y for yes and enter to confirm.

Check if broker is running:

You can check if your broker is running by for instance installing mqttfx (http://mqttfx.jfx4ee.org/index.php/download) on your pc.

Connect to your raspberry. Give a name to the profile, broker IP of the pi and port 1883 (default value used in the command line above).

Leave the rest as is



Connection Profile

Profile Name: raspberry

Broker Address: 192.168.1.20

Broker Port: 1883

Client ID: MQTT_FX_Client

User Name:

Password:

Then subscribe to your sensor. In my case to the temperature, barometer and humidity.

You can find how to subscribe to a sensor on this page https://www.tinkerforge.com/en/doc/Software/Brick_MQTT_Proxy.html , in the Topic and payload structure paragraphe:

The proxy publishes retained messages about value and configuration changes of supported devices on topics with the following pattern:

```
tinkerforge/<prefix>/<uid>/<suffix>
```

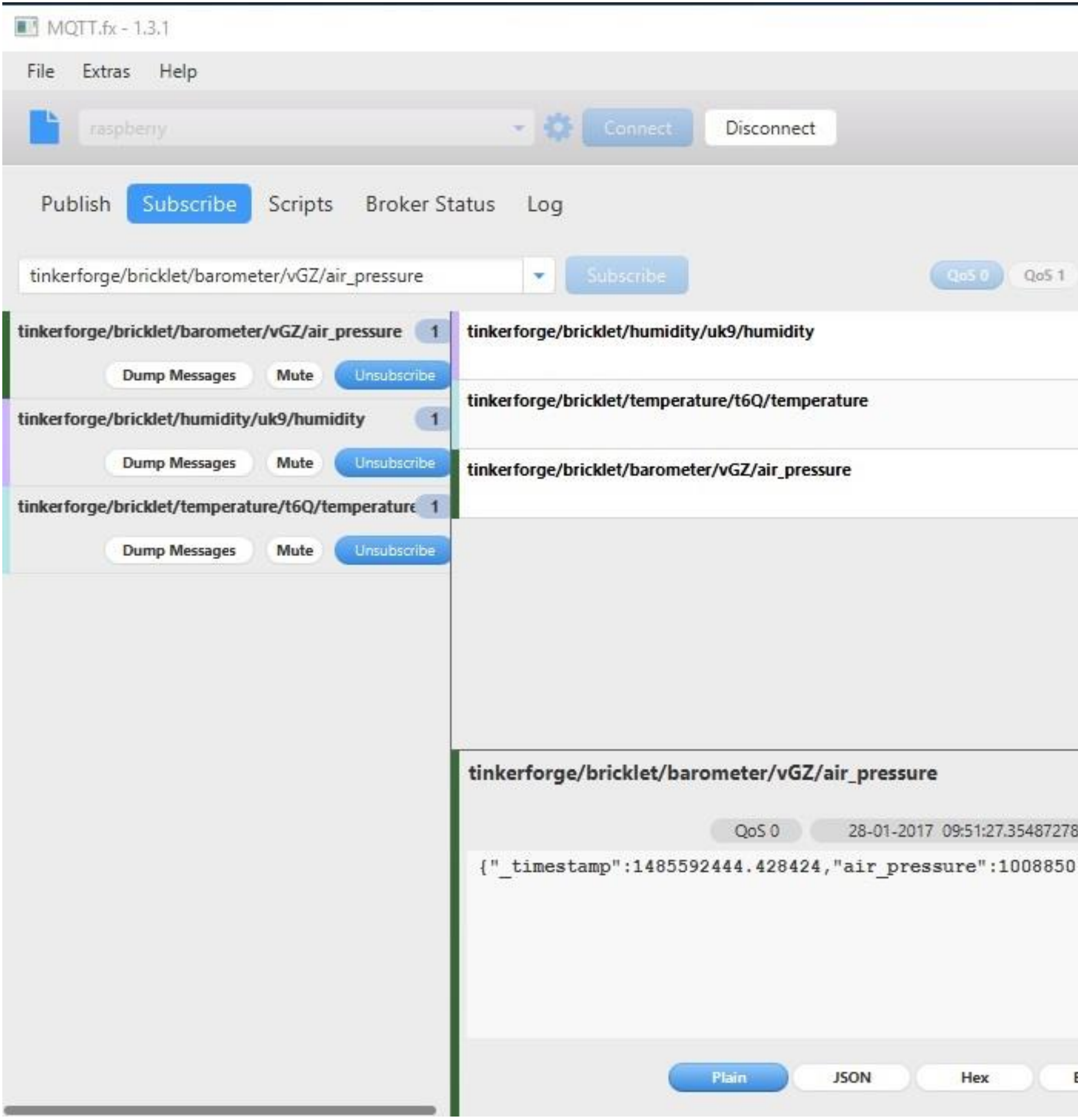
For example, for a Temperature Bricklet with UID `XYZ` the temperature value is published on:

```
tinkerforge/bricklet/temperature/XYZ/temperature
```

You can find the prefix and the suffix of your bricklet on the same page in the supported device topic paragraph and his corresponding UID at the end of the step 2 paragraph of this document.

Subscribe to your bricklet(s) with the tinkerforge/bricklet/your blicklet prefix/his UID/one of the suffix

You can see the result in the right down corner


MQTT.fx - 1.3.1
File
Extras
Help
raspberry
Connect
Disconnect
Publish
Subscribe
Scripts
Broker Status
Log
tinkerforge/bricklet/barometer/vGZ/air_pressure
Subscribe
QoS 0
QoS 1
tinkerforge/bricklet/barometer/vGZ/air_pressure 1
Dump Messages
Mute
Unsubscribe
tinkerforge/bricklet/humidity/uk9/humidity 1
Dump Messages
Mute
Unsubscribe
tinkerforge/bricklet/temperature/t6Q/temperature 1
Dump Messages
Mute
Unsubscribe
tinkerforge/bricklet/humidity/uk9/humidity
tinkerforge/bricklet/temperature/t6Q/temperature
tinkerforge/bricklet/barometer/vGZ/air_pressure
tinkerforge/bricklet/barometer/vGZ/air_pressure
QoS 0
28-01-2017 09:51:27.35487278
Plain
JSON
Hex

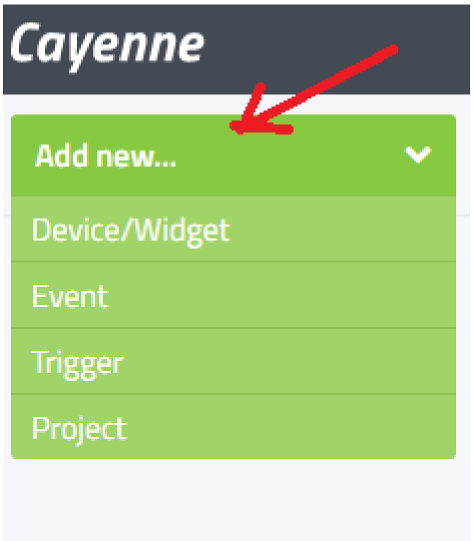
```
{"_timestamp":1485592444.428424,"air_pressure":1008850
```

Step 4 connecting to the CAYENNE IOT dashboard

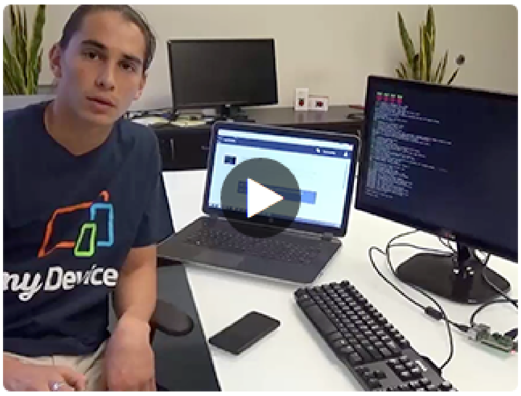
With a lot of thanks to “Adam S” of the Cayenne forum for his help.

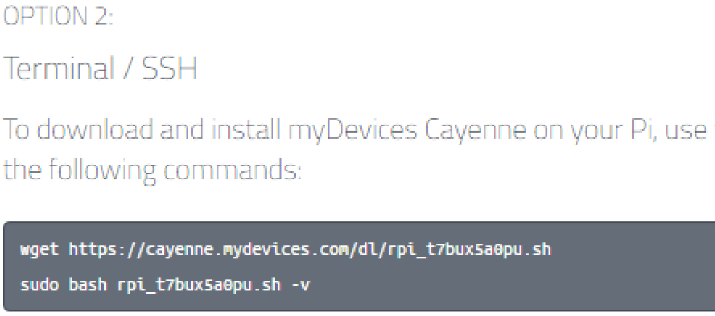
First of all GET STARTED or SIGN IN on the Cayenne IOT page
https://mydevices.com/

Once on the blank dashboard, add your raspberry pi with Add new, device/widget and chose Raspberry



You will receive two option to connect your Raspberry. I choose the second as I already have a SSH connection to my PI.




OPTION 2:

Terminal / SSH

To download and install myDevices Cayenne on your Pi, use the following commands:

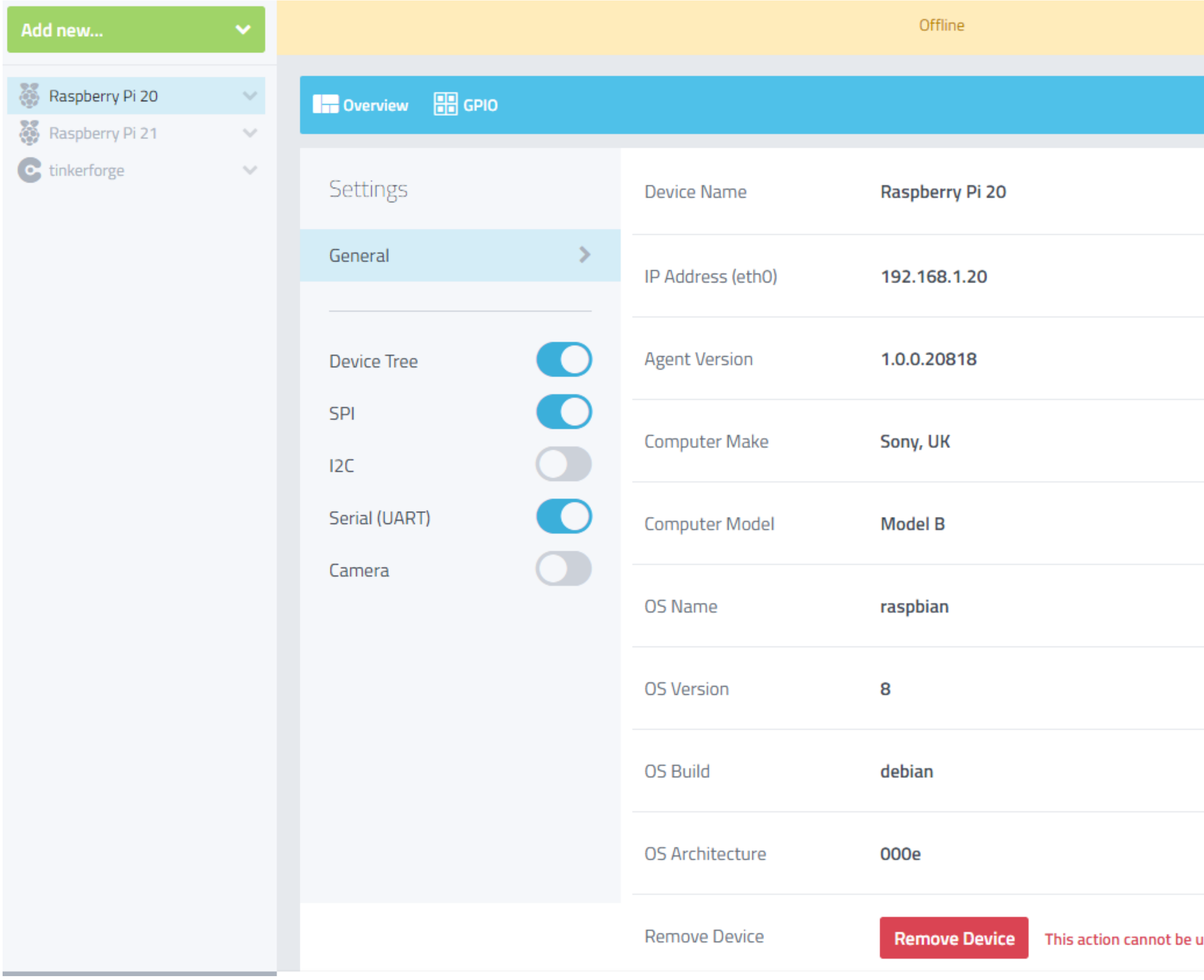
```
wget https://cayenne.mydevices.com/dl/rpi_t7bux5a0pu.sh
sudo bash rpi_t7bux5a0pu.sh -v
```

Copy and paste the two commands on your putty terminal.

REMARQUE: If you format your PI for any raison, you have to delete the formatted PI on your dashboard and redo the procedure.

The two commands line change with every new Pi.

You can edit some of your Raspberry Pi for ease of use by clicking on the setting “wheel” in the right corner

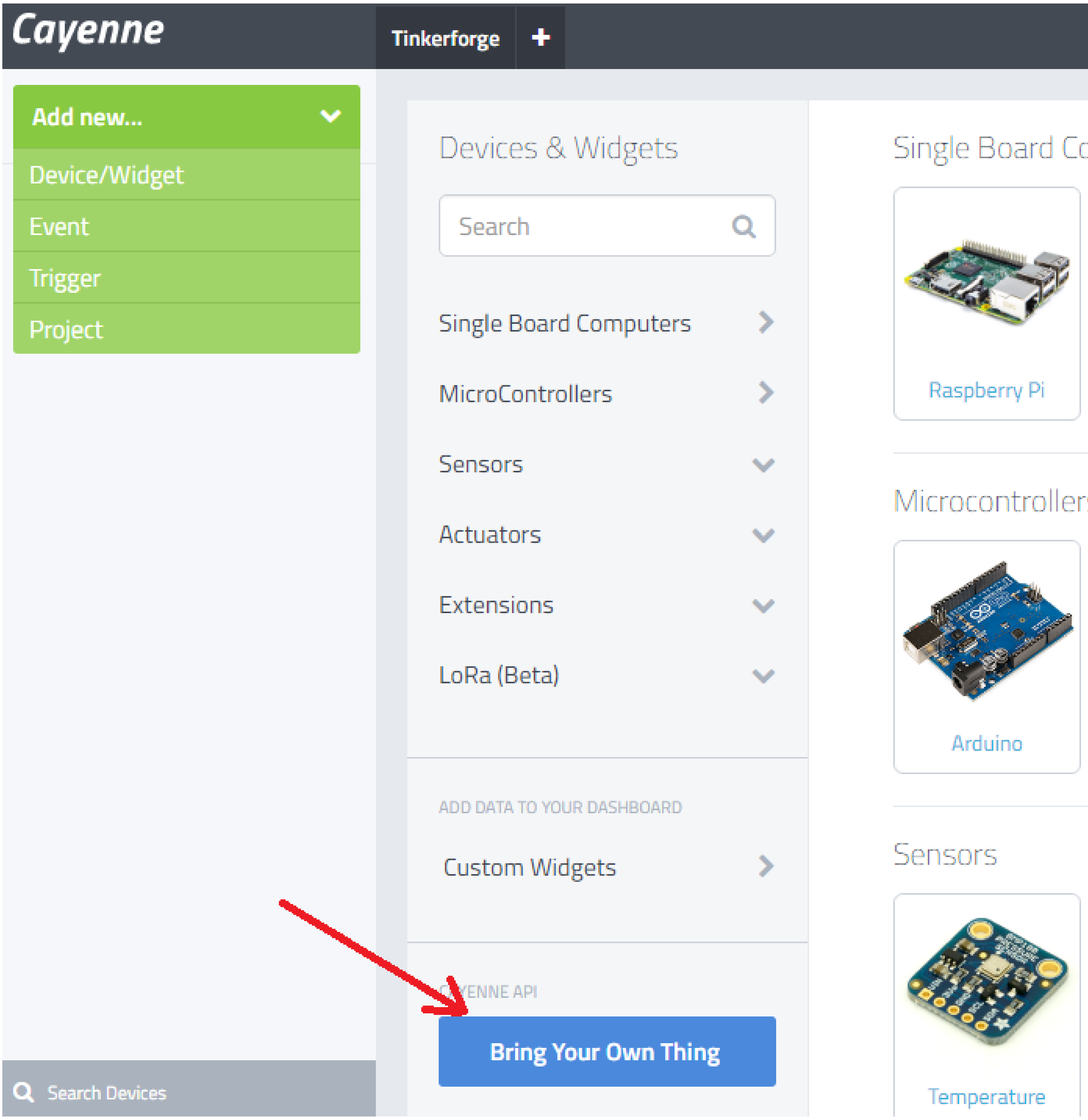


Device Tree	●
SPI	●
I2C	○
Serial (UART)	●
Camera	○

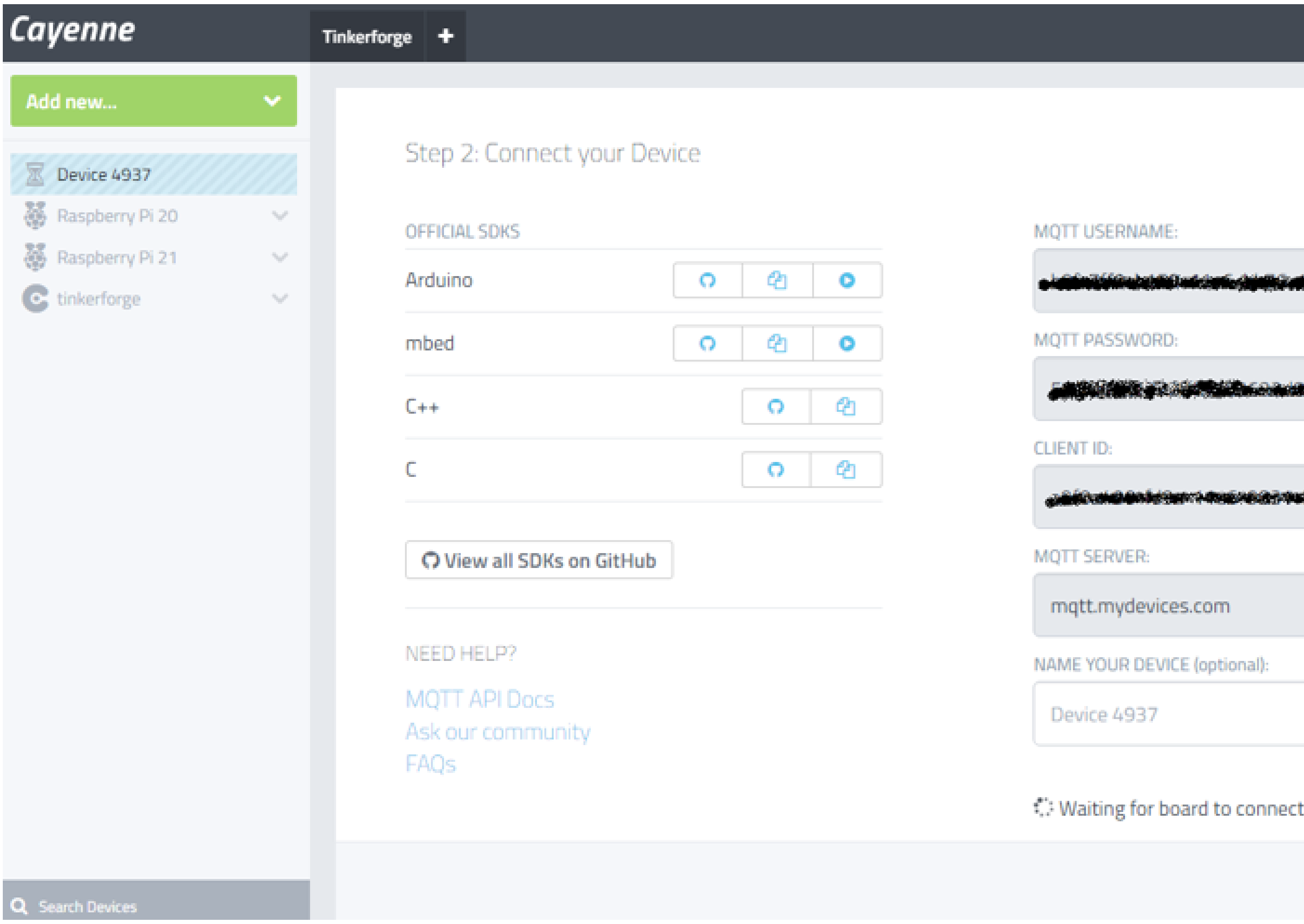
Device Name	Raspberry Pi 20
IP Address (eth0)	192.168.1.20
Agent Version	1.0.0.20818
Computer Make	Sony, UK
Computer Model	Model B
OS Name	raspbian
OS Version	8
OS Build	debian
OS Architecture	000e
Remove Device	Remove Device This action cannot be u

This is also the place you can remove an old Raspberry with the Remove Device button.

Now you have to connect the MQTT sensor to the dashboard. For this you will need to add a new device/widget a “Bring Your Own Thing”



You receive a new page:



MQTT USERNAME: [redacted]

MQTT PASSWORD: [redacted]

CLIENT ID: [redacted]

MQTT SERVER: mqtt.mydevices.com

NAME YOUR DEVICE (optional): Device 4937

With a very long unique MQTT username shorted to XXX, MQTT password shorted to YYY and Client ID shorted to ZZZ for the rest of this document.

You will need the reel values in the cayenne-mqtt.py script later.

You can also edit the Name of your Device to a suitable name; I use Tinkerforge

REMARK: if you do not see this page after clicking on the Bring Your Own Device but a continuous turning wheel, use another browser.

I had some problems with IE11 and no problems with Chrome.

Python script to publish MQTT:

Now add a python script on your Pi to publish the MQTT values to the cayenne dashboard.

Here is the code that Adam S (many thanks) from the Cayenne forum made for my arrangement

I use notepad++ to edit the file on my pc.

```
import paho.mqtt.client as mqtt
import time
import sys
import subprocess
import json

temp = None
humidity = None
airpress = None

def on_message_temp(mosq, obj, msg):
    global temp
    temp = json.loads(msg.payload)

def on_message_humidity(mosq, obj, msg):
    global humidity
    humidity = json.loads(msg.payload)

def on_message_airpress(mosq, obj, msg):
    global airpress
    airpress = json.loads(msg.payload)

Cayenne = mqtt.Client(client_id="ZZZ")    #replace your clientid value within the double quotes
Cayenne.username_pw_set("XXX", password="YYY")  #replace your username and password values within the double quotes

Cayenne.connect("mqtt.mydevices.com", port=1883, keepalive=60)

localmqtt = mqtt.Client()
localmqtt.message_callback_add("tinkerforge/bricklet/temperature/t6Q/temperature", on_message_temp) #do not include leading slash on topic
localmqtt.message_callback_add("tinkerforge/bricklet/humidity/uk9/humidity", on_message_humidity) #do not include leading slash on topic
localmqtt.message_callback_add("tinkerforge/bricklet/barometer/vGZ/air_pressure", on_message_airpress) #do not include leading slash on topic
localmqtt.connect("localhost", port=1883, keepalive=60)
localmqtt.subscribe([("/tinkerforge/bricklet/temperature/t6Q/temperature",0),("/tinkerforge/bricklet/humidity/uk9/humidity",0),("/tinkerforge/bricklet/barometer/vGZ/air_pressu
re",0)])
localmqtt.loop_start()

# for for topic is "v1/username/things/clientid/data/1" #replace username and clientid here and increment last digit for every sensor
topic_temp = "v1/XXX/things/ZZZ/data/1"  #replace username and clientid with your values here
topic_humidity = "v1/XXX/things/ZZZ/data/2" #replace username and clientid with your values here
topic_airpress = "v1/XXX/things/ZZZ/data/3" #replace username and clientid with your values here
#topic_other sensor="v1/XXX/things/ZZZ/data/4" #replace username and clientid with your values here



while True:
    try:
        if temp is not None:
            cayennetemp = "temp,c=" + str(temp['temperature']/100)
            Cayenne.publish(topic_temp, payload=cayennetemp , retain=True)

        if humidity is not None:
            cayennehumidity = "rel_hum,p=" + str(humidity['humidity']/10)
            Cayenne.publish(topic_humidity, payload=cayennehumidity , retain=True)

        if airpress is not None:
            cayenneairpress = "press,bar=" + str(airpress['air_pressure']/1000)
            Cayenne.publish(topic_airpress, payload=cayenneairpress , retain=True)

# for other sensors you can find the values for aaa, bbb and sensor name at the last pages of the MQTT Manually publishing / Subscripbing documentation
          #if sensor is not None:
                       #cayennesensor = "aaa,bbb=" + str(sensor['sensor')
              #Cayenne.publish(topic_sensor payload=cayennesensor , retain=True)



        time.sleep(5)
    except (EOFError, SystemExit, KeyboardInterrupt):
        localmqtt.loop_stop()
```

```
      Cayenne.disconnect()
      localmqtt.disconnect()
      sys.exit()
```

Make the necessary changes by replacing XXX, YYY, ZZZ with the reel values you have on your “connect your device” cayenne page.

Adapt the code to other or extra sensors if needed.

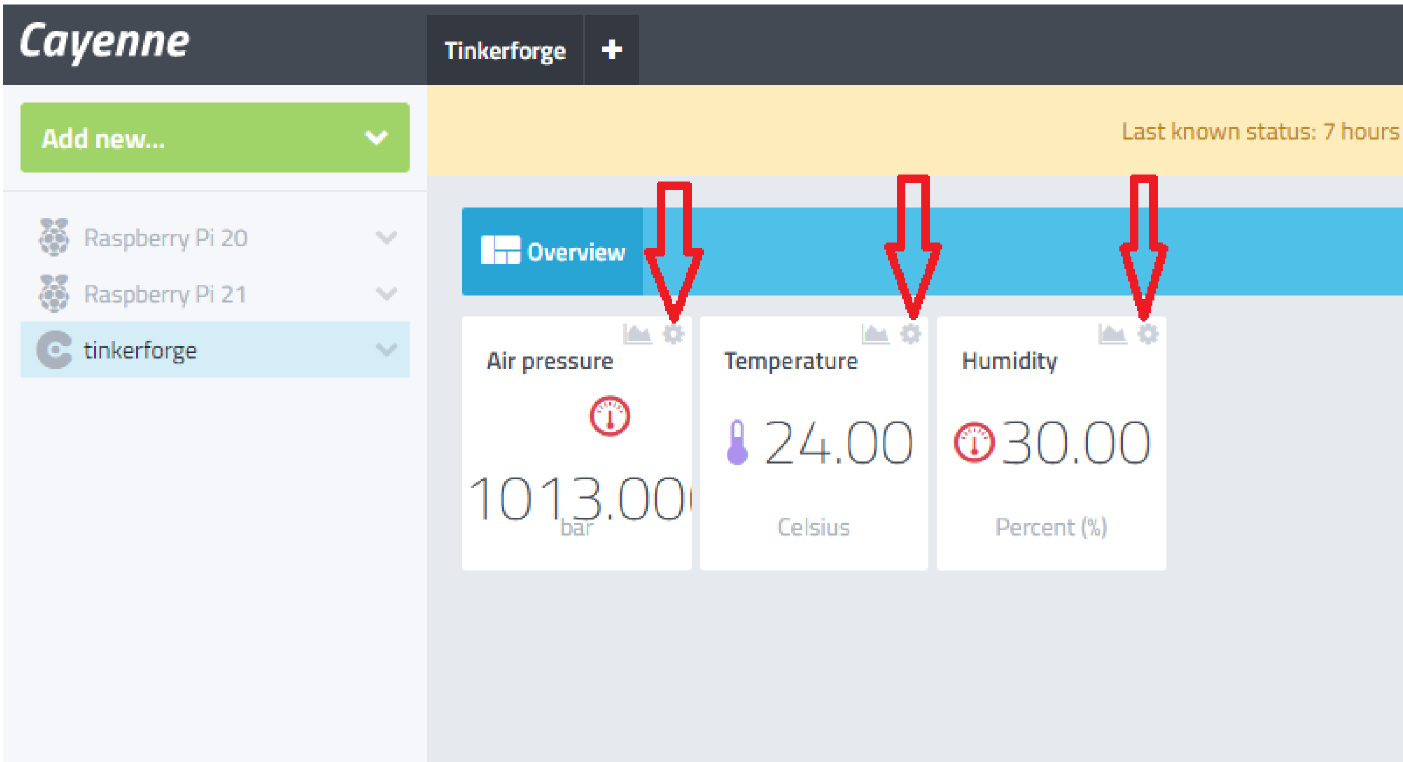
Save the file as cayenne-mqtt.py.

I used WinSCP (https://winscp.net/eng/index.php) to copy the file from my pc to the tinkerforge directory on my Raspberry PI.

Via putty terminal to the raspberry, go to the tinkerforge directory and run the cayenne-mqtt.py

```
cd /home/pi/tinkerforge
python cayenne-mqtt.py
```

Go to you cayenne dashboard and you should see, in my case, 3 new widgets temperature, humidity and airpressure.



You can customize the page with the “setting wheels”

If you have problems with the new widgets, hit ctrl+f5, try logging out and back in, or delete and re-add the widget.

REMARK: you have to retype the

```
cd /home/pi/tinkerforge
python cayenne-mqtt.py
```

commands every time you reboot your PI.

To have it start automatically at every reboot you have to modify the crontab file.

Hope this document can help you.

If you have problems please contact the Tinkerforge or Cayenne IOT forums who helped me realize this project.

Without them I could not have done it alone.

Success.