MrFox

IoT project with a Raspberry Pi 3B+

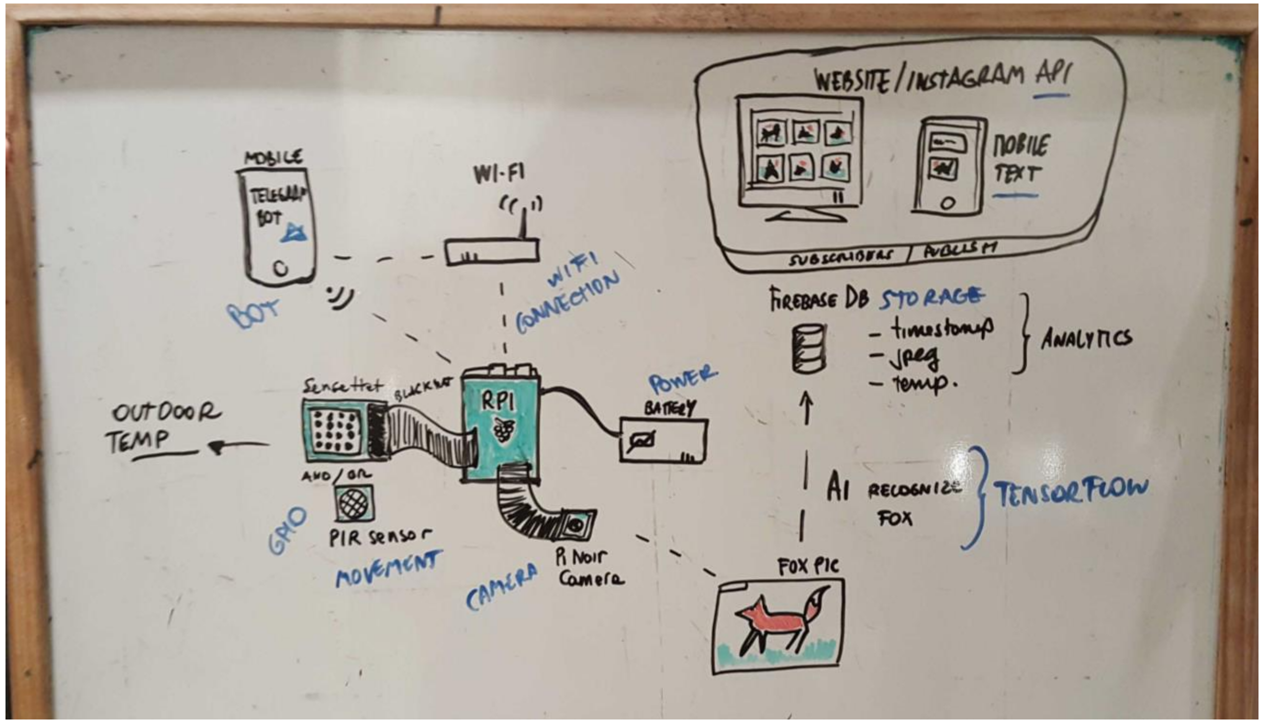
Computer Systems & Networks

Assignment 2 – Networking using Connected Devices

Federica Fiorenza

ID 20091413

The original proposal



The original idea was an outdoor camera to detect the presence of a fox in the garden using a **PIR movement sensor** and a **Pi noir camera**.

The captured images would be store in **Firebase DB** and then **published on a Website** (ideally also on Instagram). Analytics will also be published on the website and available to subscribers.

Once a fox would be detected, a **notification** is sent by email to the subscribers (or via phone) The object recognition would have been carried by **Tensorflow Coco SSD model**.

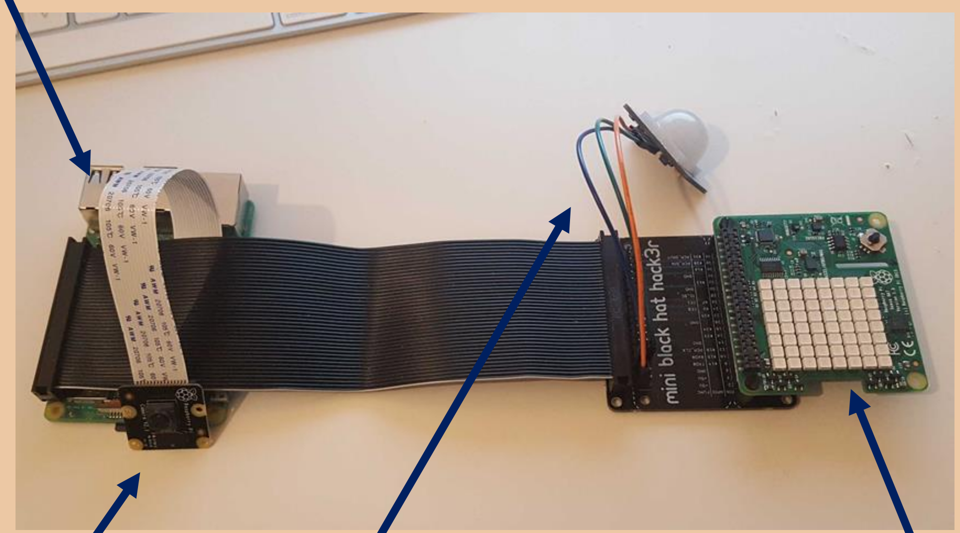
The **SenseHat** in the meantime will share the temperature, humidity and pressure via **ThingSpeak API/MTLAB visualization** on a website.

Project Graphics & Benchmarks

Raspberry Pi 3B+

Connected to the home WiFi router using 802.11n protocol (Private IP 192.168.0.25)
Network band 2.4 GHz

Access to the Terminal remotely using PuTTY using machine 192.168.0.24 (MAC - AC-67-5D-A6-1B-2F)



Pi Noir camera

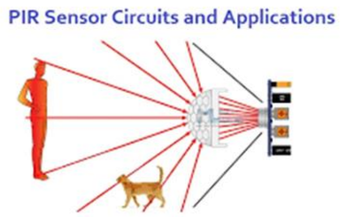
The Pi Noir Camera, designed to work also in low light conditions, gets triggered only when the PIR sensor detects movement
The images are stored to Firebase Database

PIR Motion sensor

The PIR sensor – model Parallax Inc 555-28027 (connected to the mini black hat hack3r via female to female jumper wires: GND (blue wire to GND), VCC (orange wire to 5V+) and OUT (green wire to Pin 4) will detect movement (output 1) or will be off (output 0)

SenseHat

The SenseHat is connected on GPIO 40 pins (on a mini black hat hack3r) and will be activated by a Python script who will send the data to ThingSpeak. The same data are published on a website and can be use via API to get the current values of temperature, humidity and pressure



When the sensor gets triggered, it will activate the Pi Noir camera which will take a photo in JPEG format, which will be stored into Firebase DB
(Firebase SDK credentials are specified in the Python script)

SenseHat will send Temperature, Humidity and Pressure values to ThingSpeak, which will deploy analytics to aggregator.
Using MTLAB visualizations for publishing the information to a website

ThingSpeak

The website is hosted on 000webhost.com server. The projects page will render the values taken from ThingSpeak API, the photo from Firebase Storage DB. All data displayed are live. Languages: HTML, CSS, JS
https://mrfox26445.000webhostapp.com/project.html

mailtrap by railsware

When image is taken, an email notification is sent indicating the time, the image and the current temperature taken from an API get request to the ThingSpeak endpoint.
For this project I've been configured a SMTP server (smtp.mailtrap.io), smtp user, smtp password and used port 587

Firebase Realtime Database ⇄ TensorFlow

When image is taken, this will be stored in the Firebase DB.
To make sure we only get notified when a fox (dog) is detected, a Python script will enable TensorFlow image recognition which will analyse the image on the Coco SSD model.
TensorFlow can be trained to recognize a fox, in our case we use the object "dog" for test purposes.

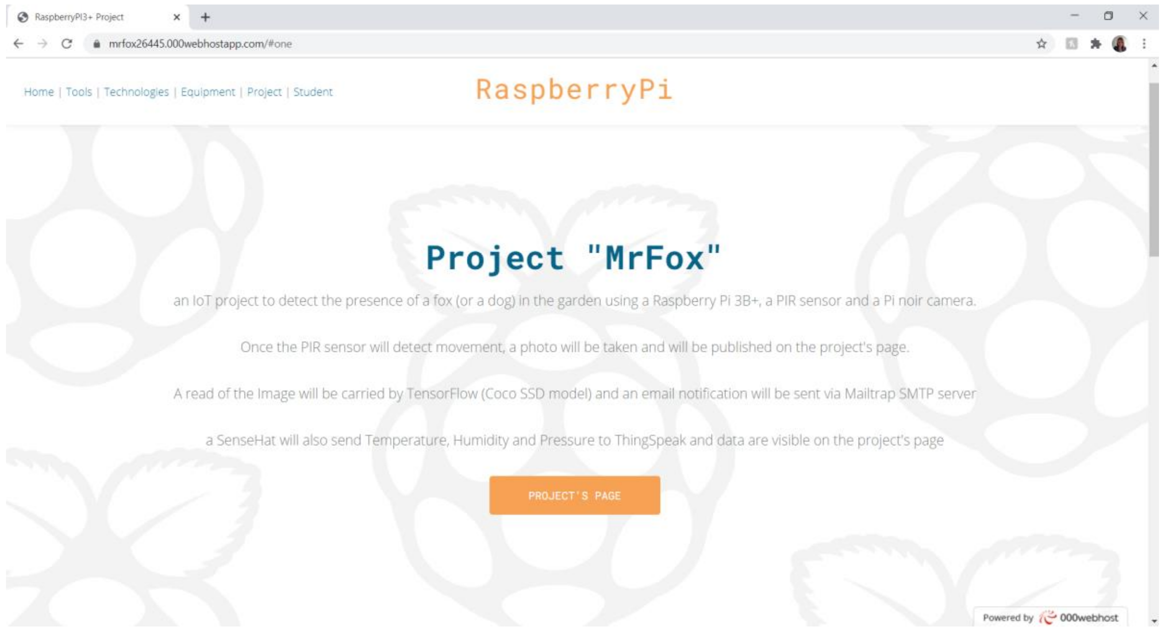
Website

For this project I've opted for a free web hosting such 000webhostapp as Glitch only allowed to make the project private after a recurring payment subscription.

URL main website:
https://mrfox26445.000webhostapp.com/

In the project.html page you can find:

- Data in relation to temperature, humidity and pressure the values are read from ThingSpeak (coming from SenseHat)
- Photo with timestamp taken when the motion sensor has detected an object.


RaspberryPI3+ Project
mrfox26445.000webhostapp.com/#one
Home | Tools | Technologies | Equipment | Project | Student
RaspberryPi
PROJECT'S PAGE
Powered by 000webhost

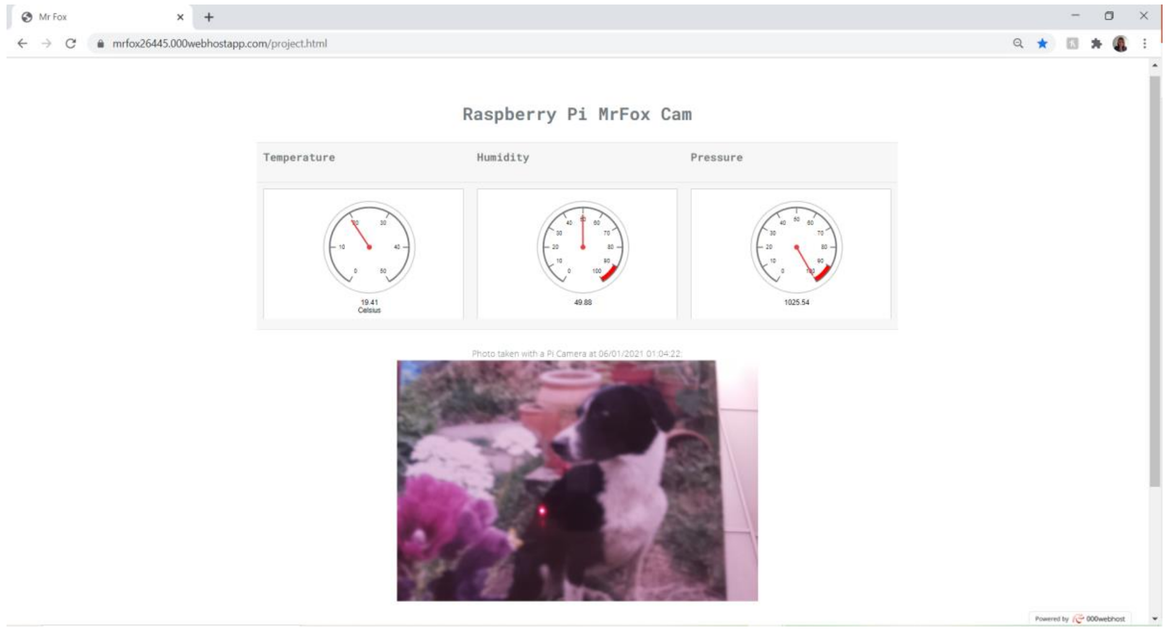
Project "MrFox"

an IoT project to detect the presence of a fox (or a dog) in the garden using a Raspberry Pi 3B+, a PIR sensor and a Pi noir camera.

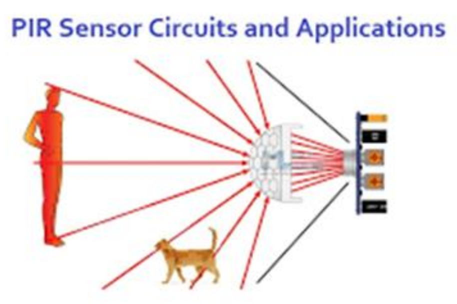
Once the PIR sensor will detect movement, a photo will be taken and will be published on the project's page.

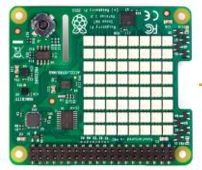
A read of the Image will be carried by TensorFlow (Coco SSD model) and an email notification will be sent via Mailtrap SMTP server

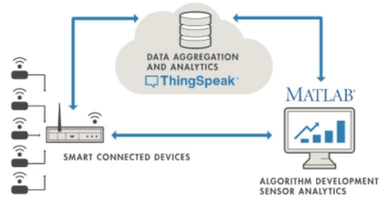
a SenseHat will also send Temperature, Humidity and Pressure to ThingSpeak and data are visible on the project's page



Project without TensorFlow – Phase 1



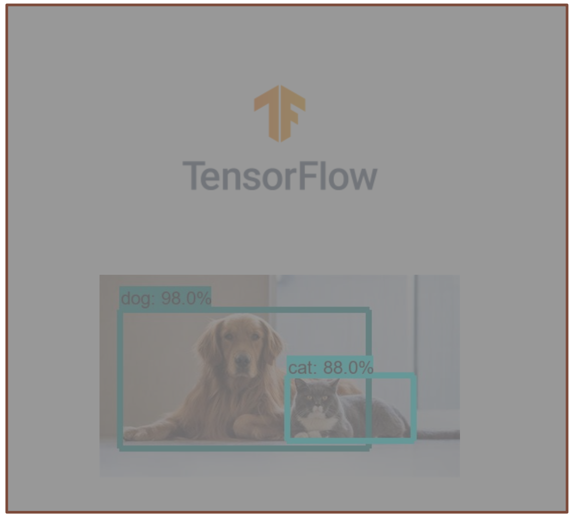



DATA AGGREGATION AND ANALYTICS
ThingSpeak
MATLAB
SMART CONNECTED DEVICES
ALGORITHM DEVELOPMENT SENSOR ANALYTICS

API

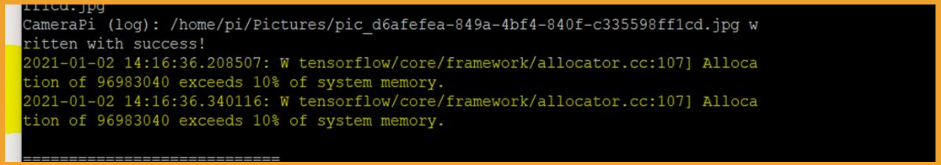
Firebase Realtime Database

API



mailtrap by railsware

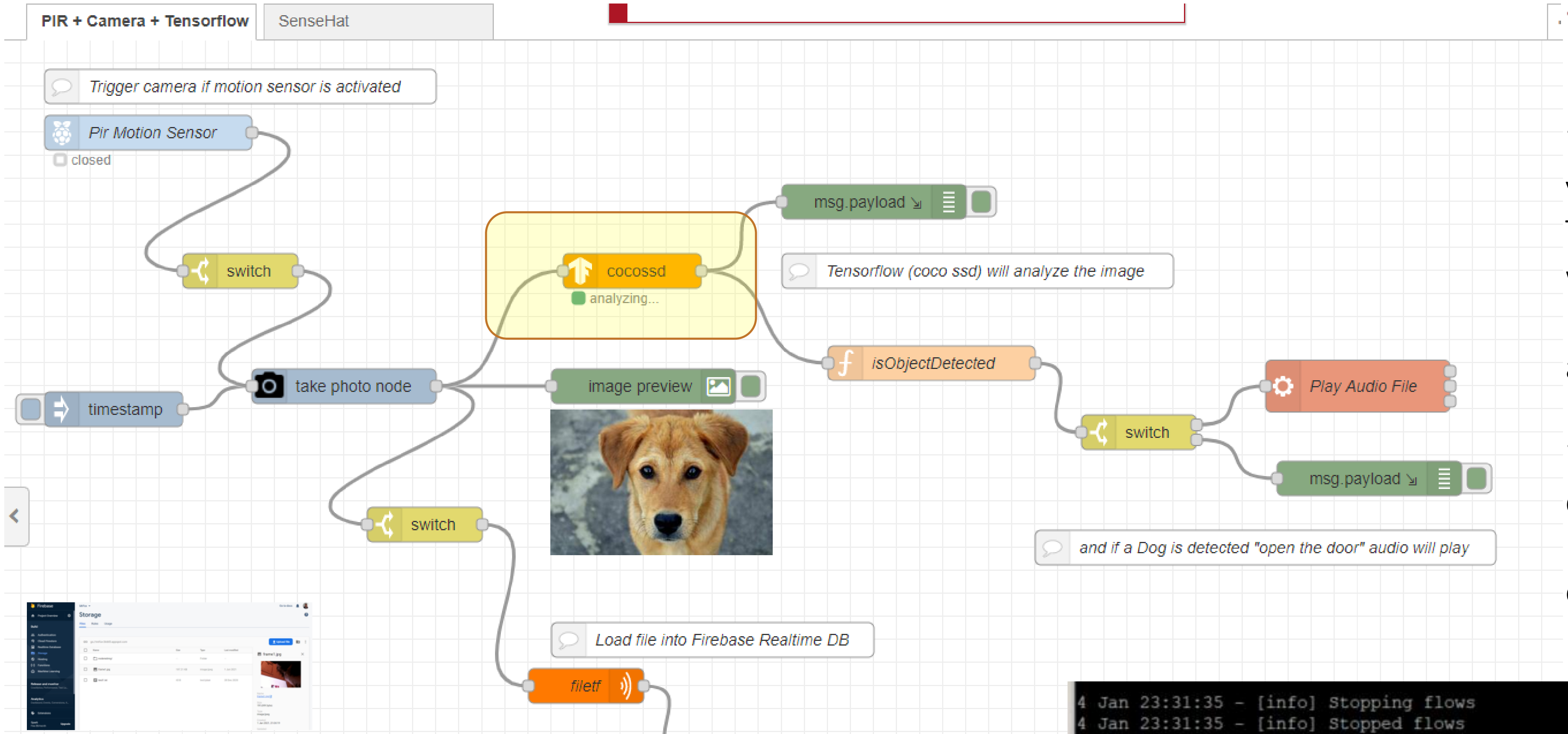
After being able resolving a major issue during the installation of TensorFlow (https://raspberrypi.stackexchange.com/questions/107483/error-installing-tensorflow-cannot-find-libhdfs-so) and managing to install version 1.14 by updating pip version, unfortunately due to a major memory issue with TensorFlow, which was already running locally with the lightest of the models (Coco), I needed to find a new way to introduce this technology, in phase 2 of this project I've opted for node-red



```
ff1cd.jpg
CameraPi (log): /home/pi/Pictures/pic_d6afefea-849a-4bf4-840f-c335598ff1cd.jpg w
ritten with success!
2021-01-02 14:16:36.208507: W tensorflow/core/framework/allocator.cc:107] Alloca
tion of 96983040 exceeds 10% of system memory.
2021-01-02 14:16:36.340116: W tensorflow/core/framework/allocator.cc:107] Alloca
tion of 96983040 exceeds 10% of system memory.
===========================
```

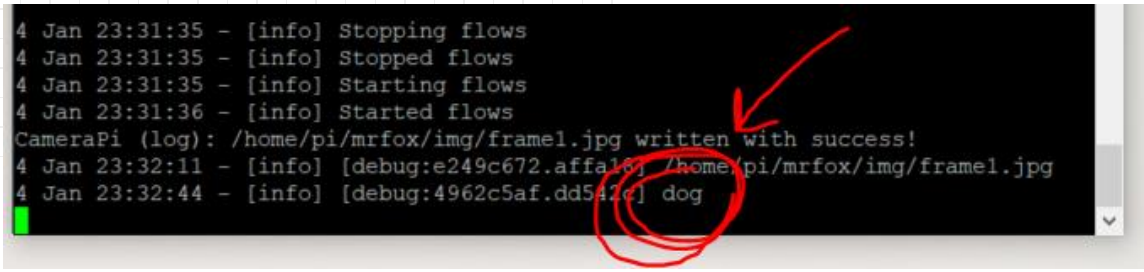
Project – Phase 2 with Node-Red

Node-RED


PIR + Camera + Tensorflow
SenseHat
Trigger camera if motion sensor is activated
Pir Motion Sensor
closed
switch
timestamp
take photo node
image preview
cocossd
analyzing...
msg.payload
Tensorflow (coco ssd) will analyze the image
isObjectDetected
Play Audio File
switch
msg.payload
and if a Dog is detected "open the door" audio will play
switch
Load file into Firebase Realtime DB
filetf
msg.payload
Storage

Finally with node-red it was possible to view the result from Tensorflow (coco ssd), which was able to recognize an image of a dog.
In the JS script, if the image detected is of a dog is true, there will be an audio file at the end of the flow.

Note: The Firebase module in use in node-red console contains a bug, which prevented to integrate initially the module correctly. It has been resolved by changing the project id of the JSON credentials with the correct database name. The owner of the module has been notified via Slack.



```
4 Jan 23:31:35 - [info] Stopping flows
4 Jan 23:31:35 - [info] Stopped flows
4 Jan 23:31:35 - [info] Starting flows
4 Jan 23:31:36 - [info] Started flows
CameraPi (log): /home/pi/mrfox/img/frame1.jpg written with success!
4 Jan 23:32:11 - [info] [debug:e249c672.affa16] /home/pi/mrfox/img/frame1.jpg
4 Jan 23:32:44 - [info] [debug:4962c5af.dd542c] dog
```

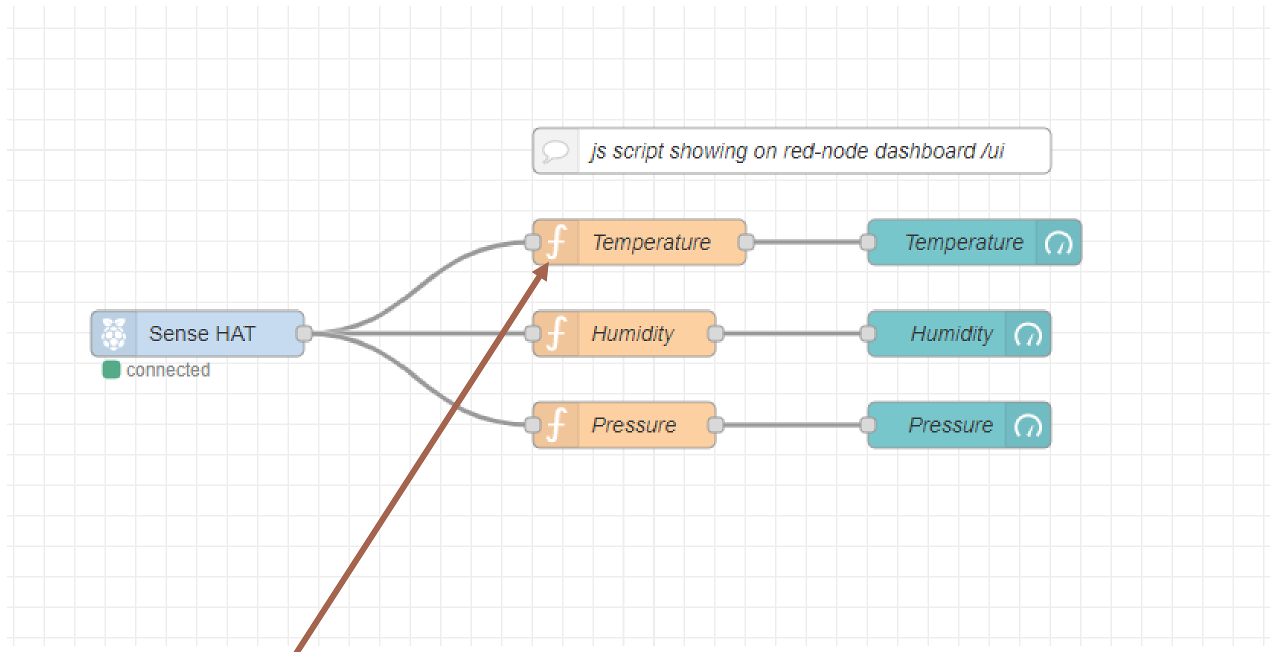
Additional features /discoveries
Learning Opportunities

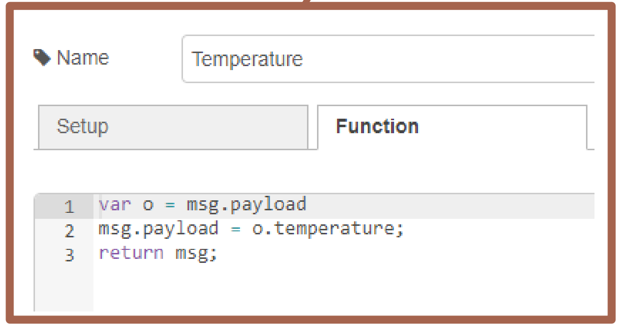
SenseHat

node-red UI

Functionality achieved with node-red UI

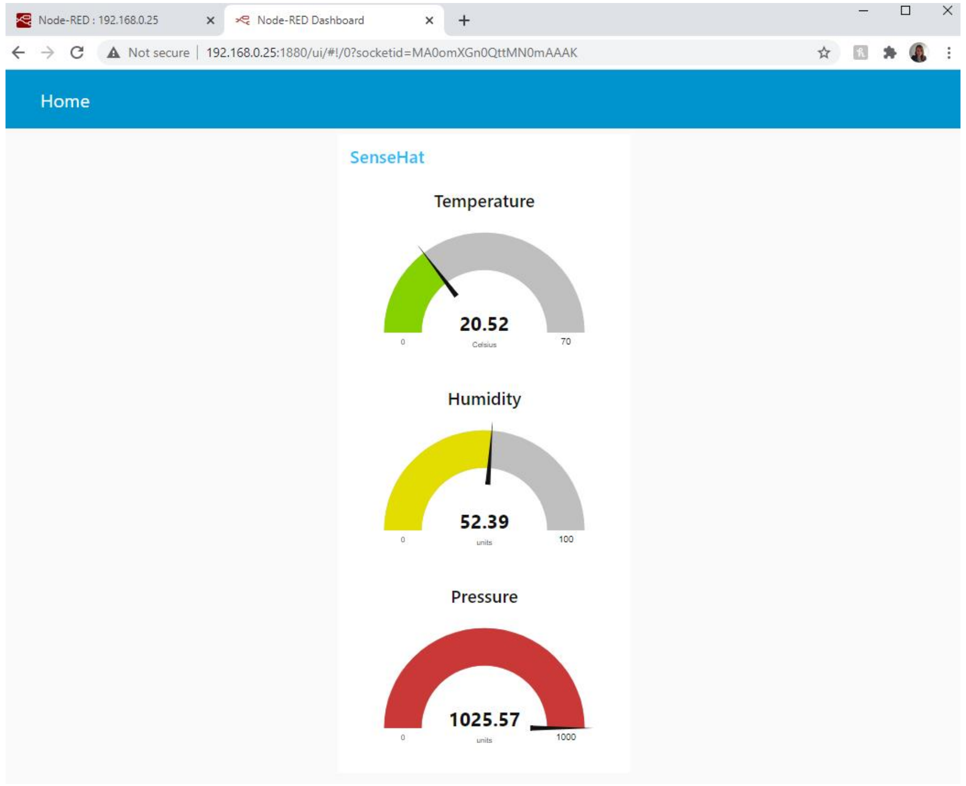
Data are published from the SenseHat directly to the website



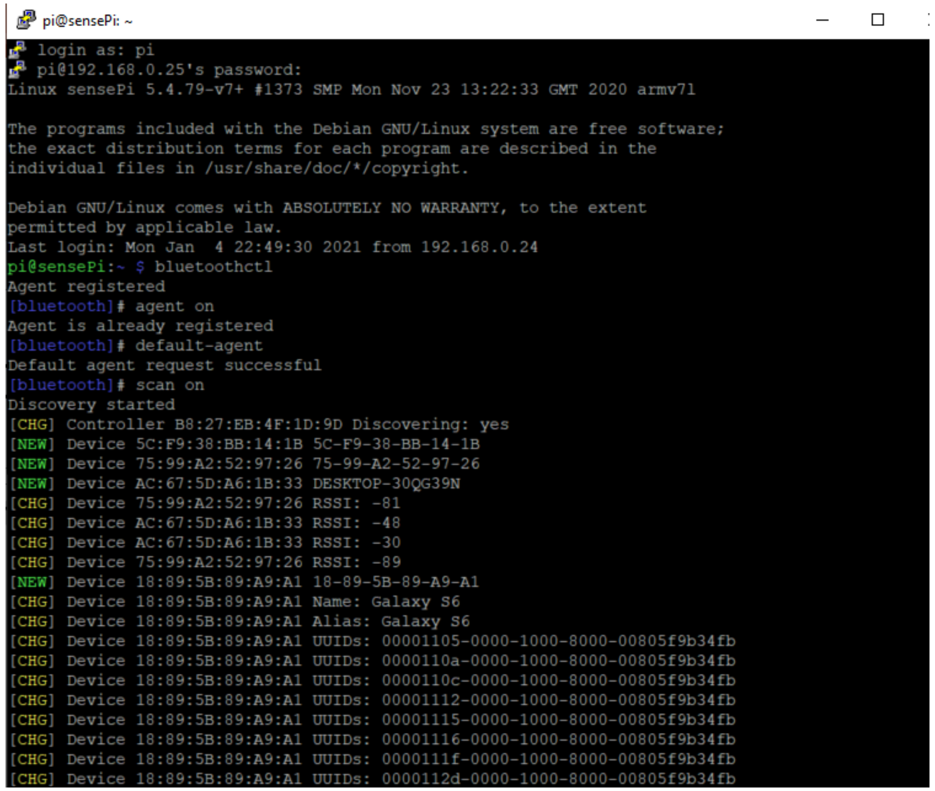


```
var o = msg.payload
msg.payload = o.temperature;
return msg;
```

JS function example



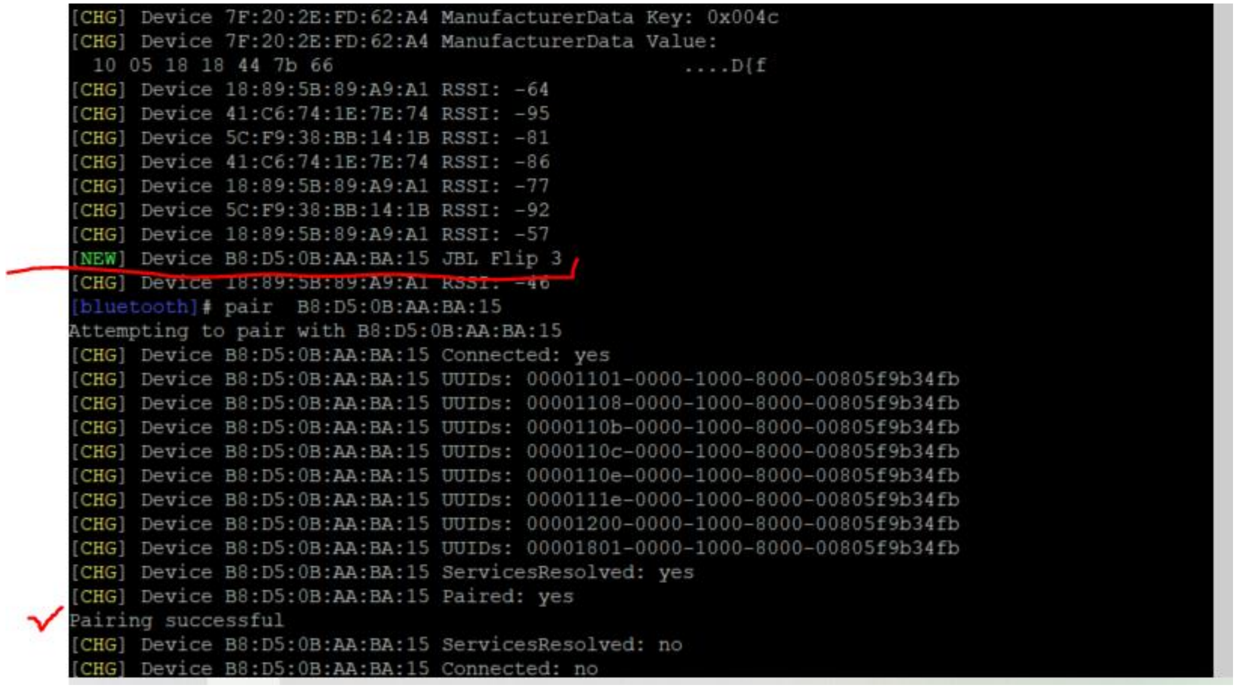
Pairing headless RPi with JBL BT speaker using mac address discovery



```
login as: pi
pi@192.168.0.25's password:
Linux sensePi 5.4.79-v7+ #1373 SMP Mon Nov 23 13:22:33 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jan  4 22:49:30 2021 from 192.168.0.24
pi@sensePi:~ $ bluetoothctl
Agent registered
[bluetooth]# agent on
Agent is already registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:4F:1D:9D Discovering: yes
[NEW] Device 5C:F9:38:BB:14:1B 5C-F9-38-BB-14-1B
[NEW] Device 75:99:A2:52:97:26 75-99-A2-52-97-26
[NEW] Device AC:67:5D:A6:1B:33 DESKTOP-30QG39N
[CHG] Device 75:99:A2:52:97:26 RSSI: -81
[CHG] Device AC:67:5D:A6:1B:33 RSSI: -48
[CHG] Device AC:67:5D:A6:1B:33 RSSI: -30
[CHG] Device 75:99:A2:52:97:26 RSSI: -89
[NEW] Device 18:89:5B:89:A9:A1 18-89-5B-89-A9-A1
[CHG] Device 18:89:5B:89:A9:A1 Name: Galaxy S6
[CHG] Device 18:89:5B:89:A9:A1 Alias: Galaxy S6
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 18:89:5B:89:A9:A1 UUIDs: 0000112d-0000-1000-8000-00805f9b34fb
```



```
[CHG] Device 7F:20:2E:FD:62:A4 ManufacturerData Key: 0x004c
[CHG] Device 7F:20:2E:FD:62:A4 ManufacturerData Value:
  10 05 18 18 44 7b 66                             ....D{f
[CHG] Device 18:89:5B:89:A9:A1 RSSI: -64
[CHG] Device 41:C6:74:1E:7E:74 RSSI: -95
[CHG] Device 5C:F9:38:BB:14:1B RSSI: -81
[CHG] Device 41:C6:74:1E:7E:74 RSSI: -86
[CHG] Device 18:89:5B:89:A9:A1 RSSI: -77
[CHG] Device 5C:F9:38:BB:14:1B RSSI: -92
[CHG] Device 18:89:5B:89:A9:A1 RSSI: -57
[NEW] Device B8:D5:0B:AA:BA:15 JBL Flip 3
[CHG] Device 18:89:5B:89:A9:A1 RSSI: -46
[bluetooth]# pair  B8:D5:0B:AA:BA:15
Attempting to pair with B8:D5:0B:AA:BA:15
[CHG] Device B8:D5:0B:AA:BA:15 Connected: yes
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 00001101-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 00001108-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device B8:D5:0B:AA:BA:15 ServicesResolved: yes
[CHG] Device B8:D5:0B:AA:BA:15 Paired: yes
Pairing successful
[CHG] Device B8:D5:0B:AA:BA:15 ServicesResolved: no
[CHG] Device B8:D5:0B:AA:BA:15 Connected: no
```

https://www.cnet.com/how-to/how-to-setup-bluetooth-on-a-raspberry-pi-3/