

RAPPORT DE PROJET IOT



Ea Victorine, DAILLIE Corentin, NOMENYO Perla, VINCENT Kolin

Efrei Paris, ISCE promotion 2019

SOMMAIRE

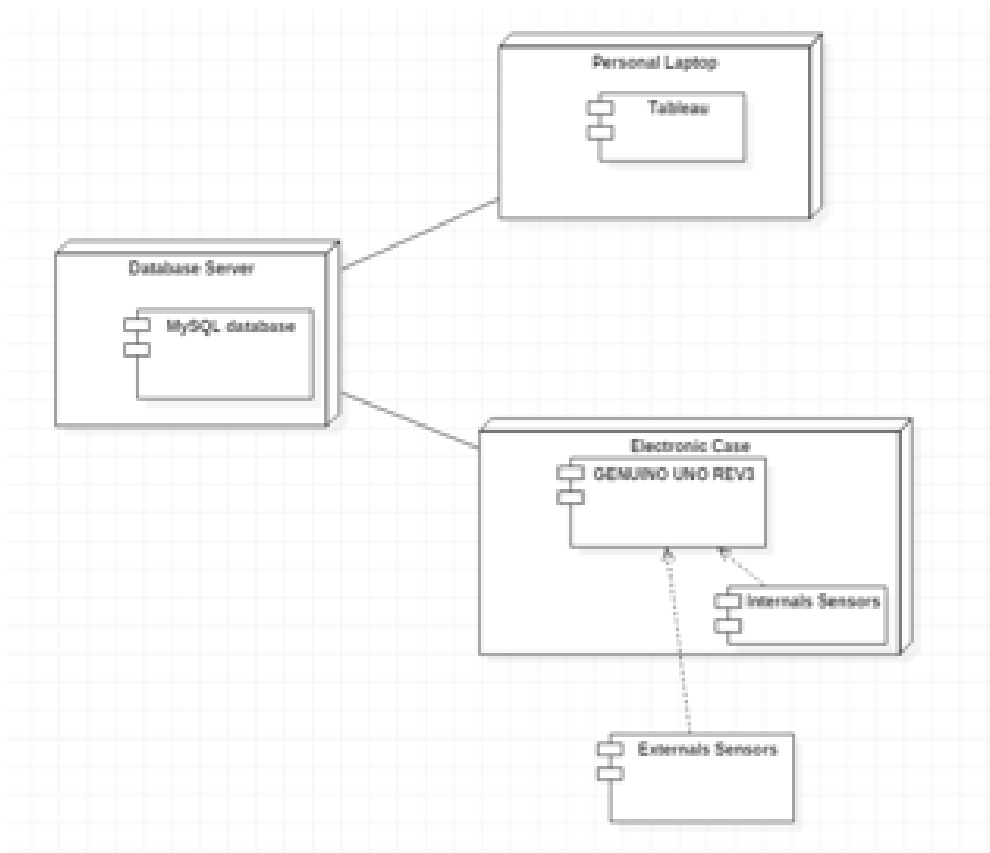
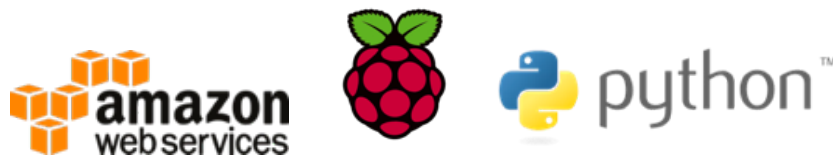
I. COMPOSANTS UTILISÉS	3
Température intérieure et humidité	3
Température extérieure	3
Mouvement	4
Raspberry Pi	5
Autres composants	5
II. RÉALISATION DU PROJET	6
Branchements et mise en marche des capteurs	6
Structure de la base de données	7
Affichage des données sur console et .txt	8
Transmission des données récoltées vers la base de données (AWS)	10
III. CONTRIBUTION PERSONNELLES	11

INTRODUCTION

En ce moment en M1 nous sommes en train de réaliser un grand projet de 6 mois à 8 personnes. Notre projet consiste à réaliser un système basé sur un ensemble de capteurs permettant d'effectuer diverses mesures (température, luminosité, mouvement, qualité de l'air...). Nous pourrions récolter les informations obtenues et les traiter afin de pouvoir d'un côté déterminer les éventuelles causes de perte/gaspillage énergétique et d'un autre côté sensibiliser les occupants des locaux à leur impact environnemental.

Le but de ce projet est de permettre à différents organismes de pouvoir récupérer des données, pour en faire une étude statistique afin d'évaluer leur consommation d'énergie.

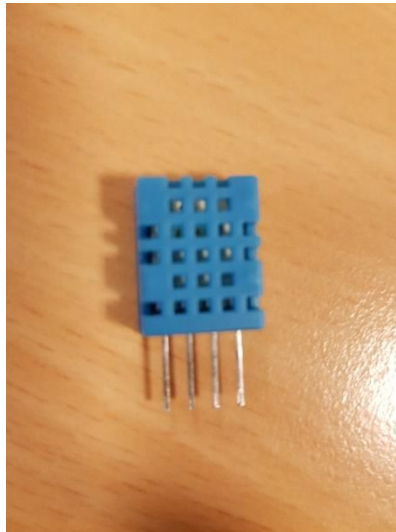
Technologies utilisées : Raspberry PI, Capteurs, Tableau, Python, AWS, DynamoDB (NoSQL), Tableau.



Nous avons décidé de baser notre projet d'IOT sur un projet sur lequel nous travaillions déjà auparavant. Le but de ce projet est de relier différents capteurs à une raspberry PI afin de pouvoir quantifier en de manière temporelle puis financière le gaspillage de ressources énergétique des occupants d'une pièce.

I. COMPOSANTS UTILISÉS

Température intérieure et humidité



- SODIAL(R) Capteur d'humidité et température numérique Moudle sonde pour le CVC Arduino 4 broches.
- Temps de réponse: <5s.
- Alimentation: DC 3.5-5.5V
- Faible consommation d'énergie.
- DHT11 Senseur

Température extérieure



- Sonde en utilisant DS18B20 puce d'origine du capteur de température étanche
- tuyau en acier inoxydable de haute qualité anti-rouille emballage étanche à l'eau
- Boîtier en acier inoxydable (6 * 50mm), longueur de câble 100cm
- Alimentation 3,0V ~ 5,5V
- 9 à 12 résolution réglable
- Large plage de température de -55 ~ + 125
- Taille: Env. 100cm x 0.5cm x 0.5cm / 39.37x0.2x0.2inch
- Matière: Acier inoxydable

Mouvement



- Type de produit: HC - SR501 Module de capteur de corps module PIR motion humaine mini module détecteur IR infrarouge
- Gamme de tension de fonctionnement: DC 4.8-20V
- Courant de repos: <50uA
- Niveau de sortie: élevé 3.3 V / Bas 0V
- Déclencheur: L ne peut pas être déclenché en répétition / H peut être déclencheur répété (déclencheur répété par défaut)
- Temps de retard: 0.5-200S (réglable)
- Temps de blocage: 2.5S (par défaut)
- Dimensions de la carte: 32mmX24mm
- Capteur d'angle: angle de cône <100 °
- Opération Temp: -15- + 70 degrés
- Capteur de taille de lentille: Diamètre: 23mm (par défaut)

Raspberry PI 3 : modèle B



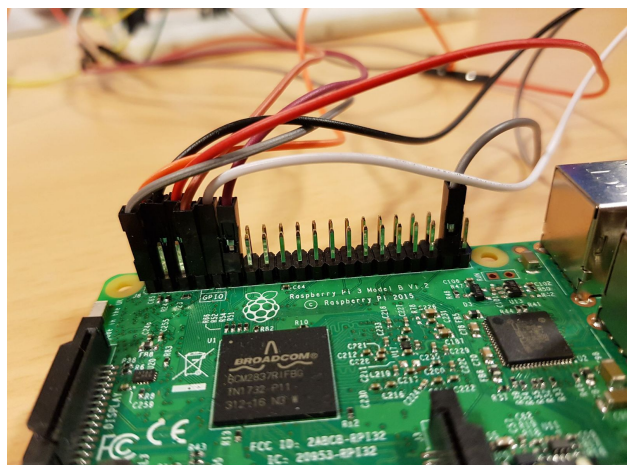
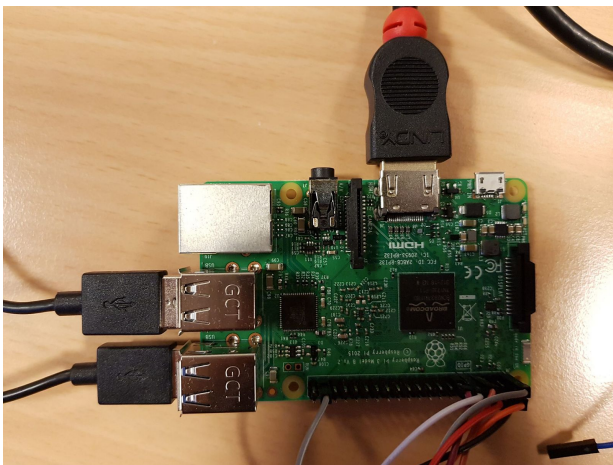
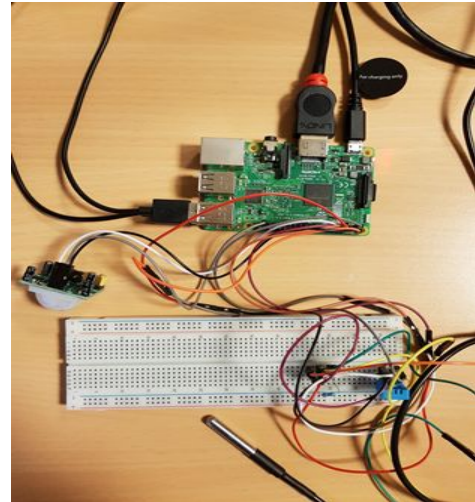
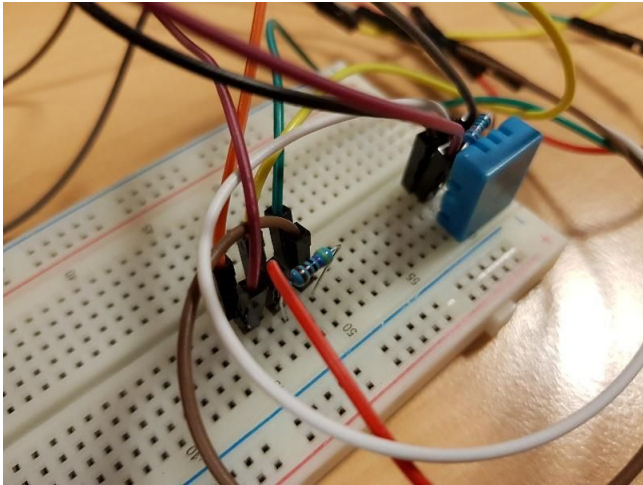
Autres composants :

Fils (Jumper Wires), câbles HDMI, clavier, souris, résistance, moyenne breadboard, câble d'alimentation, carte micro-SD.



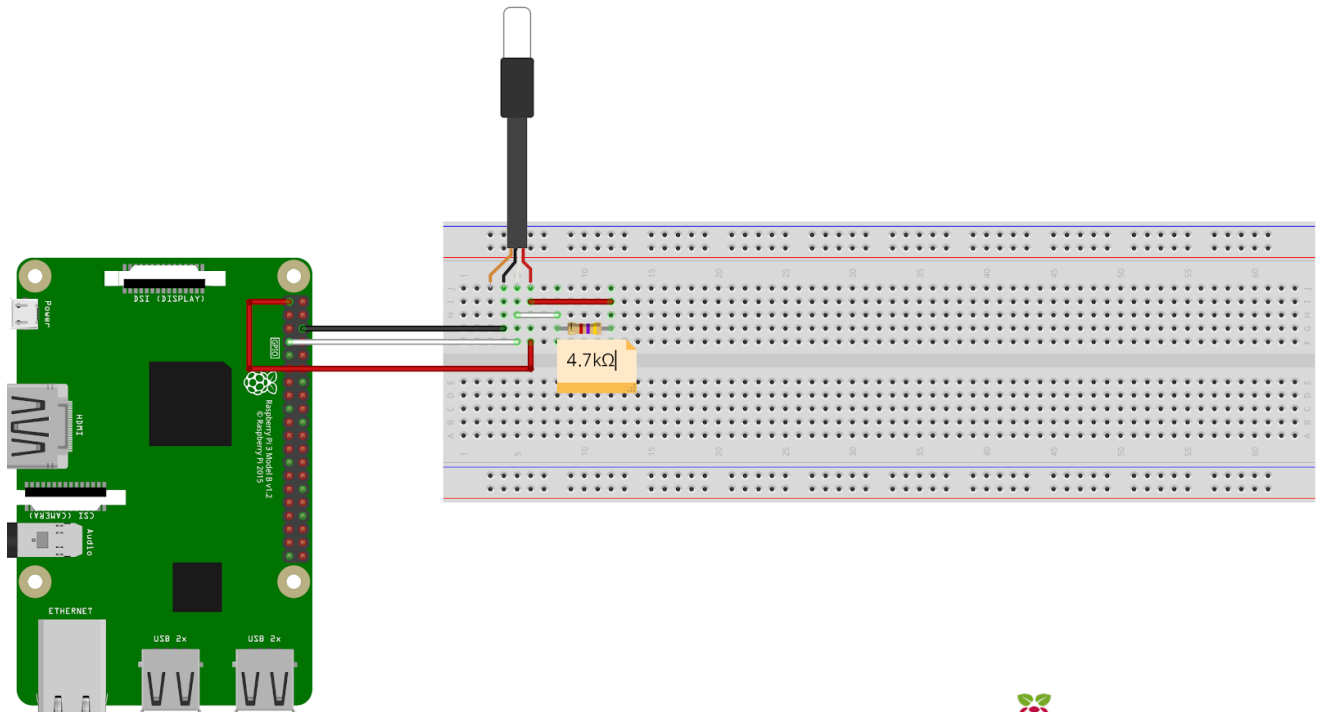
II. RÉALISATION DU PROJET

Branchements et mise en marche des capteurs



Pour la réalisation du projet nous avons suivi des tutoriels sur internet qui nous ont expliqué l'emplacement des branchements avec des schémas mais aussi les scripts de la commande, de bash et de python fournis afin de mettre en marche les différents capteurs.

Voici un exemple de schémas pour le DS18B20 température extérieure :



 Espace Raspberry Francais

Voici notre bash .sh :

```
dependencies.sh
1 #Setup dependencies
2
3
4 #DHT sensor humidity and temperature
5 sudo apt-get update
6 sudo apt-get install python-dev
7 sudo apt-get install python-rpi.gpio
8
9 git clone https://github.com/adafruit/Adafruit_Python_DHT.git
10 cd Adafruit_Python_DHT
11 sudo apt-get update
12 sudo apt-get install build-essential python-dev python-openssl
13 sudo python setup.py install
14
15 #TSL2561 Luminosity Sensor
16 cd ..
17 git clone https://github.com/bivab/smbus-cffi.git
18 cd smbus-cffi
19 python setup.py install
20
```


Quelques sites sur lesquels nous nous sommes basés :

<http://mycoolpizza.blogspot.fr/2013/05/raspberry-pi-utiliser-un-capteur-de.html>

<https://raspberrypi.developpez.com/cours-tutoriels/capteur/mag-pi-utiliser-port-gpio/parte-2-detection-temperature-1-wire/>

<http://espace-raspberry-francais.fr/Composants/Sonde-de-temperature-DS18B20-sur-Raspberry-Francais/>

<http://nagashur.com/blog/2015/08/30/ds18b20-raspberry-pi-mesurer-la-temperature-avec-une-sonde-numerique/>

Configuration du raspberry PI

Comme cela a été dit précédemment, le Raspberry Pi est un nano-ordinateur. Par conséquent, comme tout ordinateur, pour pouvoir l'utiliser, il est nécessaire d'avoir :

- une carte micro-SD est nécessaire, car c'est elle qui va contenir la partie logicielle et Système d'Exploitation
- un écran ;
- un clavier ;
- une souris ;
- un câble réseau ou une connexion WiFi pour Internet ;
- une source d'alimentation.

Étapes de branchement de vos périphériques avec votre Raspberry Pi

1. Insérer une carte SD dans le connecteur Raspberry Pi dédié
2. Brancher le câble ethernet à la box du fournisseur d'accès à internet (FAI)
3. Brancher le clavier et la souris aux ports USB
4. Brancher le câble HDMI avec l'écran
5. Brancher l'alimentation

Votre Raspberry Pi devrait alors démarrer automatiquement.

La configuration du raspberry PI a également été faite à l'aide de tutoriels dont voici les sources :

<http://www.raspberrypi-france.fr/premiere-utilisation-raspberry-pi/>

<https://alexandre-laurent.developpez.com/articles/hardware/raspberry-pi/mise-en-route/>

<http://www.freney.net/web/?Premiers-pas-avec-un-Raspberry-Pi-3-decouverte-et-installation>

Structure de la base de données

Compte tenu de la diversité et de la disparité des données que nous devons prendre en compte, nous avons décidé d'utiliser une base de données *DynamoDB* qui permet de gérer des données non relationnelles (NoSQL)...

Affichage des données sur console et .txt

On branche le raspberry PI au PC. Puis on ouvre le code python pour les 3 capteurs dans un terminal (Température intérieure & humidité, Température extérieure et mouvement).

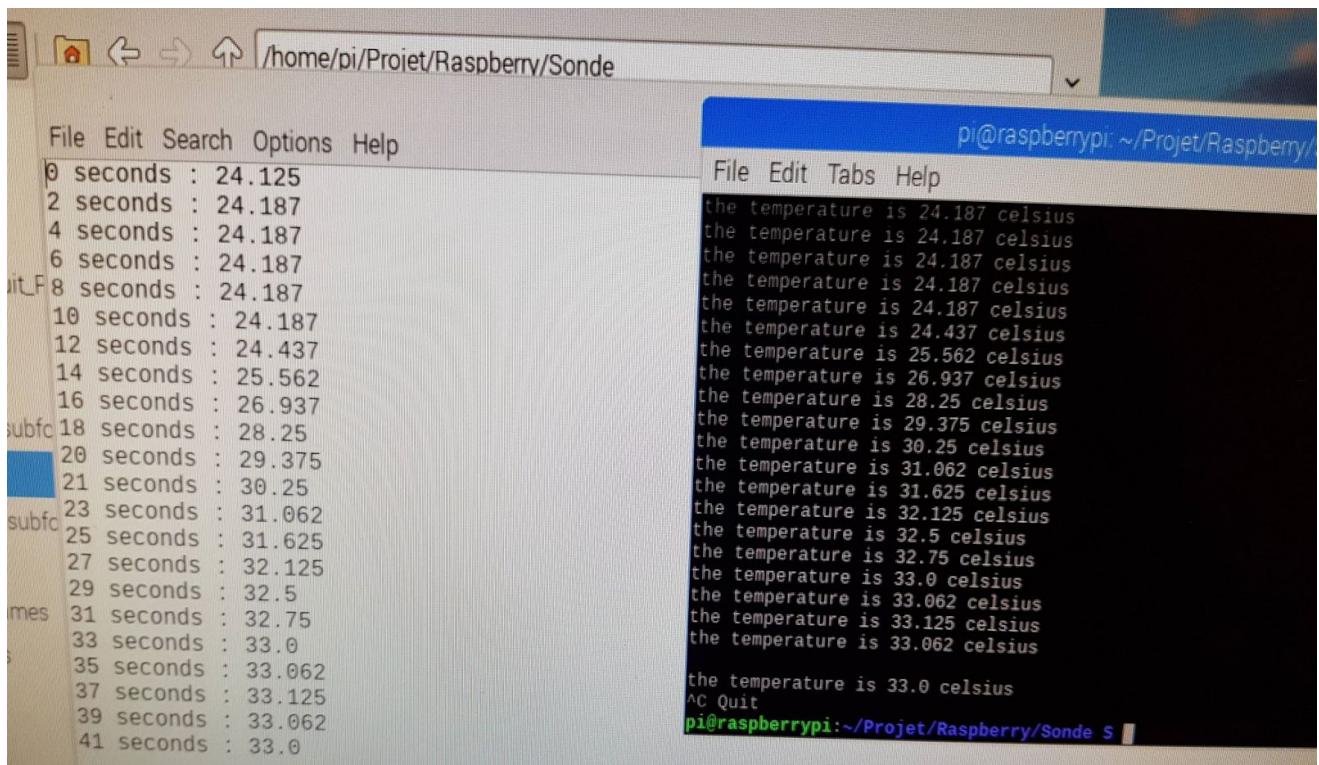
Les données seront directement récupérées dans un fichier .txt mais aussi connectées directement à une base de données via un serveur cloud AWS.

Température intérieur et humidité : >> sudo python temphumo_run.py 11 4

The screenshot shows a Raspberry Pi terminal window with the command `sudo python temphumo_run.py 11 4` executed. The terminal output displays a list of temperature and humidity readings over time. A file explorer window is also open, showing the contents of the `temphumiint.txt` file, which contains the same data as the terminal output.

Time (seconds)	Temp	Humidity
1	23.0*	26.0%
12	24.0*	24.0%
22	24.0*	23.0%
33	24.0*	23.0%
44	24.0*	23.0%
54	24.0*	24.0%
65	24.0*	24.0%
75	24.0*	24.0%
86	24.0*	23.0%
96	24.0*	23.0%
107	24.0*	22.0%
117	24.0*	22.0%
128	26.0*	95.0%
138	28.0*	95.0%
149	27.0*	95.0%
159	26.0*	95.0%
170	26.0*	95.0%

Température extérieure : >> sudo python3 sonde_run.py



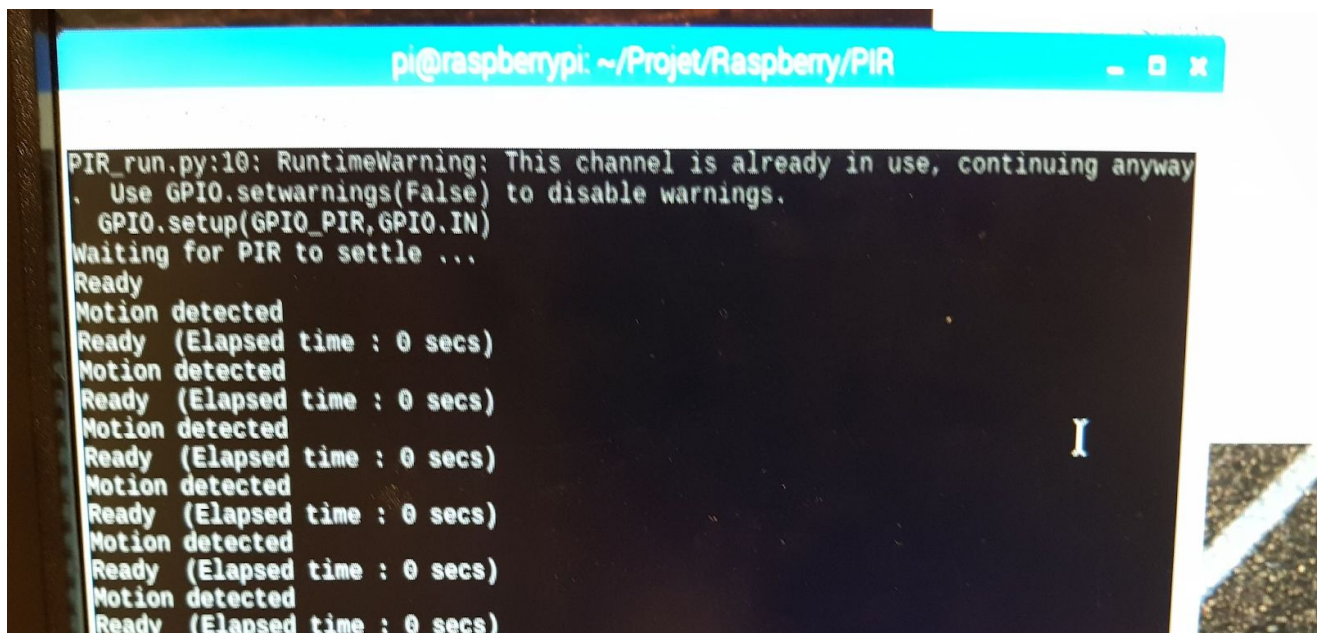
```

File Edit Search Options Help
0 seconds : 24.125
2 seconds : 24.187
4 seconds : 24.187
6 seconds : 24.187
8 seconds : 24.187
10 seconds : 24.187
12 seconds : 24.437
14 seconds : 25.562
16 seconds : 26.937
18 seconds : 28.25
20 seconds : 29.375
21 seconds : 30.25
23 seconds : 31.062
25 seconds : 31.625
27 seconds : 32.125
29 seconds : 32.5
31 seconds : 32.75
33 seconds : 33.0
35 seconds : 33.062
37 seconds : 33.125
39 seconds : 33.062
41 seconds : 33.0

pi@raspberrypi: ~/Projet/Raspberry/
File Edit Tabs Help
the temperature is 24.187 celsius
the temperature is 24.187 celsius
the temperature is 24.187 celsius
the temperature is 24.187 celsius
the temperature is 24.187 celsius
the temperature is 24.437 celsius
the temperature is 25.562 celsius
the temperature is 26.937 celsius
the temperature is 28.25 celsius
the temperature is 29.375 celsius
the temperature is 30.25 celsius
the temperature is 31.062 celsius
the temperature is 31.625 celsius
the temperature is 32.125 celsius
the temperature is 32.5 celsius
the temperature is 32.75 celsius
the temperature is 33.0 celsius
the temperature is 33.062 celsius
the temperature is 33.125 celsius
the temperature is 33.062 celsius
the temperature is 33.0 celsius
AC Quit
pi@raspberrypi: ~/Projet/Raspberry/Sonde S

```

Mouvement : >> sudo python PIR_run.py



```

pi@raspberrypi: ~/Projet/Raspberry/PIR
PIR_run.py:10: RuntimeWarning: This channel is already in use, continuing anyway
Use GPIO.setwarnings(False) to disable warnings.
GPIO.setup(GPIO_PIR,GPIO.IN)
Waiting for PIR to settle ...
Ready
Motion detected
Ready (Elapsed time : 0 secs)
Motion detected
Ready (Elapsed time : 0 secs)
Motion detected
Ready (Elapsed time : 0 secs)
Motion detected
Ready (Elapsed time : 0 secs)
Motion detected
Ready (Elapsed time : 0 secs)
Motion detected
Ready (Elapsed time : 0 secs)

```


Transmission des données récoltées vers la base de données AWS

DataPA8 [Close](#)

Overview **Items** Metrics Alarms Capacity Indexes Global Tables Backups [More](#) ▾

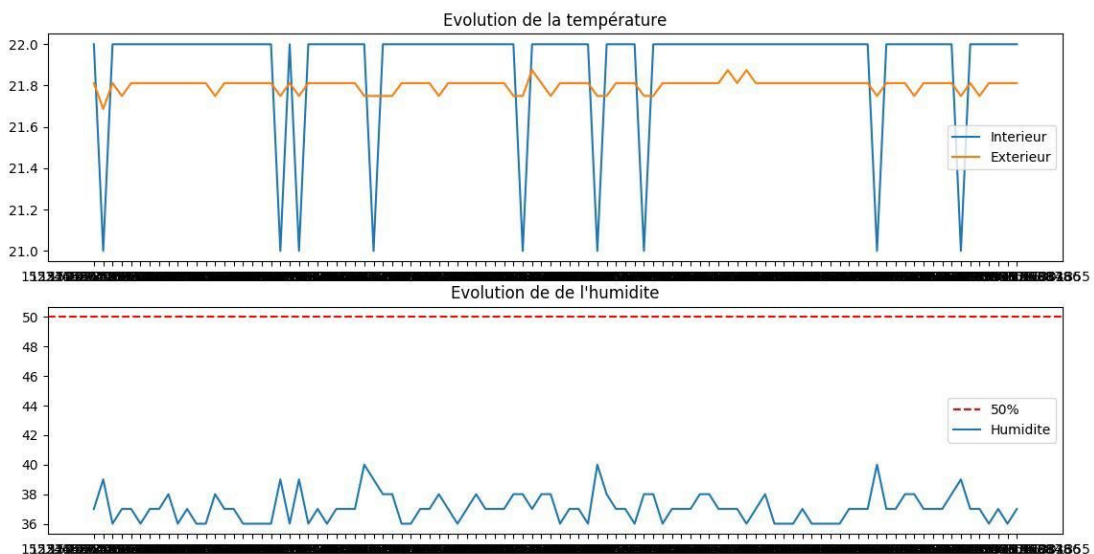
[Create item](#) [Actions](#) ▾

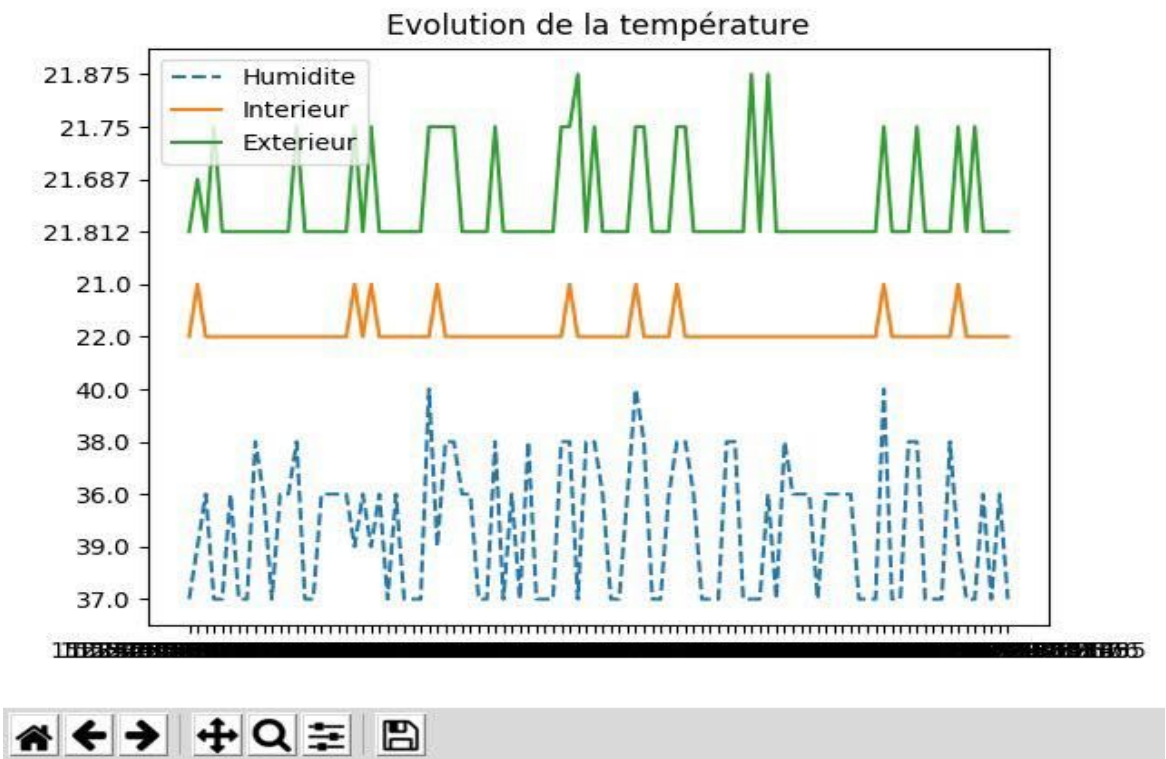
Scan: [Table] DataPA8: id, timestampMS [Viewing 1 to 100 items](#) >

Scan [Table] DataPA8: id, timestampMS [Add filter](#) [Start search](#)

<input type="checkbox"/>	id	timestampMS	info
<input type="checkbox"/>	1	1523530287....	{ "extern" : { "S" : "21.312" }, "humidity" : { "S" : "49.0" }, "intern" : { "S" : "21....
<input type="checkbox"/>	1	1523530294....	{ "extern" : { "S" : "21.312" }, "humidity" : { "S" : "50.0" }, "intern" : { "S" : "21....
<input type="checkbox"/>	1	1523530312....	{ "extern" : { "S" : "21.312" }, "humidity" : { "S" : "50.0" }, "intern" : { "S" : "22....
<input type="checkbox"/>	1	1523530318....	{ "extern" : { "S" : "21.312" }, "humidity" : { "S" : "50.0" }, "intern" : { "S" : "22....
<input type="checkbox"/>	1	1523530328....	{ "extern" : { "S" : "21.312" }, "humidity" : { "S" : "50.0" }, "intern" : { "S" : "22....

Voici quelques schémas illustrant les données récoltées sur les capteurs via AWS :





III. CONTRIBUTIONS PERSONNELLES

EA Victorine	Achat des composants, branchement et mise en marche des capteurs de température (code), transmission des données vers aws
DAILLIE Corentin	Structure de la base de données, transmission des données vers aws
NOMENYO Perla	Branchement et mise en marche du capteur de mouvement, configuration du Raspberry, transmission des données vers aws
VINCENT Kolin	Branchement et mise en marche de l'ensemble des capteurs (code), configuration du Raspberry, transmission des données vers aws