



# Rapport De Projet Internet des Objets

## Captain Planet

Heytem BOUMAZA

Rémi PHYU THANT THAR

M1 Informatique 2019-2020

---

***Table des matières***

---

<b>I</b>	<b>Introduction</b>	<b>4</b>
<b>II</b>	<b>Problématique</b>	<b>5</b>
<b>III</b>	<b>Etat de l'art</b>	<b>7</b>
III.1	Temtop . . . . .	8
III.1.1	Produit . . . . .	8
III.1.2	Producteur . . . . .	8
III.1.3	Analyse . . . . .	8
III.1.4	Produits associés . . . . .	10
III.2	Netatmo . . . . .	11
III.2.1	Produit . . . . .	11
III.2.2	Producteur . . . . .	11
III.2.3	Analyse . . . . .	11
III.2.4	Produit associé . . . . .	13
III.3	Flow/Flow 2 . . . . .	14
III.3.1	Produit . . . . .	14
III.3.2	Producteur . . . . .	14
III.3.3	Analyse . . . . .	14
III.4	Etat du marché/ chiffre d'affaire . . . . .	16
<b>IV</b>	<b>Solution proposée</b>	<b>17</b>
<b>V</b>	<b>Etude Fonctionnelle</b>	<b>18</b>
V.1	La conception . . . . .	18
V.2	Diagramme de Use Case . . . . .	19
V.3	Composants physiques . . . . .	20
V.3.1	Branchemet . . . . .	20
V.3.2	Composants . . . . .	21

V.4 Analyse de données . . . . .	28
<b>VI Application smartphone</b>	<b>30</b>

## **Section I**

---

### **Introduction**

---

Ce jour d'aujourd'hui, nous connaissons une très grande dépendance de la productivité industrielle pour la société et de l'économie mondiale, d'où la montée alarmante des infrastructures et des installations industrielles auprès des habitations de la population. En effet, cette croissance rapide des activités industrielles créent des problèmes environnementaux comme pollution (air, eau, bruit) ainsi que des problèmes planétaires, qui n'affecte pas seulement que le climat global mais aussi la santé et le bien-être de la population. De ce fait, la nécessité de systèmes de surveillance adaptables, efficaces, bon marché et intelligents dans nos lieux d'habitation, même à l'intérieur de nos résidences, devient importante de nos jours.

Lorsque on parle de polluants, nous pensons certainement souvent à ceux qui se trouvent dans l'environnement à l'extérieur, que ce soit dans l'air, le sol ou les cours d'eau. Cependant, La qualité de l'air à l'intérieur de notre résidence, est tout aussi vitale pour notre santé et bien-être au quotidien, et les polluants trouvés à l'intérieur devraient nous préoccuper autant, sinon plus.

Dans le contexte de la combinaison de nombreux défis de l'informatique, des communications sans fil et de l'électronique, la technologie basée sur l'internet des objets (Internet of Things) est un domaine de recherche émergent, avec le principe étant l'interconnexion entre l'Internet et des objets, des lieux et des environnements physiques. L'IOT recouvre aujourd'hui nombreux domaines d'applications comme la domotique, la gestion des usines, ou bien la détection environnementale, connectant des millions d'objets et de personnes.

Dans ce rapport, nous présentons le problème exact auquel nous sommes confrontés et les solutions existantes ainsi que la proposition de notre propre solution sous la forme d'un système de surveillance de la qualité de l'air basé sur la technologie IOT.

## ***Section II***

---

### ***Problématique***

---

La pollution de l'air est l'un des problèmes environnementaux majeurs que nous connaissons aujourd'hui. En effet, 92 % de la population mondiale respire un air trop pollué par les particules fines en milieu rural et urbain selon les rapport de l'OMS. De ce fait, une attention croissante est centrée sur la résolution de ce problème. La pollution de l'air dans les grandes villes engendre un risque important pour la santé de la population. Elle peut réduire l'espérance de vie jusqu'à 22 mois et c'est pourquoi les communautés profitent des technologies IOT pour améliorer le contrôle des émissions environnementales.



Pollution de l'air à New Delhi, Inde

L'objectif est d'atténuer les risques menaçant la santé et de sensibiliser aux effets de l'exposition à la pollution atmosphérique. Dans le cas de notre projet, il s'agit de surveiller la qualité de l'air à l'intérieur de notre résidence. Bien que les solutions IoT ne remplaceront pas initialement les stations de surveillance fixes, la facilité de leur déploiement peut apporter des avantages importants. On pourra alors utiliser un mini-station de météo à déployer chez soi, détectant aussi la qualité de l'air de l'intérieur de la résidence. Etant un objet connecté, on pourra alors consulter les résultats détectés à distance via une application mobile, qui traite les données captées et transmises par la station. Cela permettra aux foyers dans les grandes villes de surveiller les conditions météologique intérieures, ainsi que rester au dessus des niveaux dangereux de qualité de l'air, tout en réalisant leurs propres mesures météologique. Lorsqu'on parle de la qualité de l'air, on doit prendre en compte plusieurs éléments :

- CO<sub>2</sub> (dioxyde de carbone) : polluant associé aux activités utilisant le gaz naturel comme les voitures, usines, avion etc.
- PM1/PM2.5/PM10 (particulate matter) les matières en suspension dans l'atmosphère en phase liquide et solide (aérosols), représentant un mélange complexe de substances chimiques organiques et inorganiques et que l'on regroupe sous le terme général de "particules en suspension", pouvant nuire à notre santé. Ils sont classés en fonction de leur "diamètre aérodynamique" (ex. PM10 : particules avec diamètre inférieur à 10 µm).



## ***Section III***

---

### ***Etat de l'art***

---

En veillant sur les produits déjà conçus et semblables à notre projet, on constate qu'il existe des stations et des capteurs personnels sophistiqués et complètes en fonctionnalités déjà bien installées dans le marché. En effet, on peut retrouver couramment de stations météologique pour usage personnel dans le marché actuel, fiables et à un prix relativement convenable. Parmi elles, il existe aussi celles de type 'intelligente' et connectée, recouvrant toutes les fonctionnalités (et plus) qu'on veux avoir avec notre station, dont on peut s'en servir de modèle. On étudiera ici quelques produits disponibles associés à notre projet.

### III.1 Temtop

(Déetecte PM2,5 HCHO,TVOC, AQI)

#### III.1.1 Produit

**Temtop**



Temtop M10 DéTECTEUR de qualité de l'air

Prix : 91,99 €

Poids : 198 g

Dimensions : 81 \* 81 \* 30 mm

Capacité de la batterie : 2200 mAh.

Durée de vie de la batterie : 6 heures

Capteur Temtop

#### III.1.2 Producteur

Temtop est une marque de la firme américaine Elitech Technolgy, Inc, fondé en 1996. Elle est principalement dédiée à la R& D et fabrication de produits pour l'industrie de la congélation et de la réfrigération et fourniture de solutions de surveillance à distance de congelation. Elle travaille avec l'industrie d'alimentation, de stockage phamaceutique, et la surveillance de température.

#### III.1.3 Analyse



Le détecteur de qualité de l'air Temtop est un simple moniteur de qualité de l'air, avec comme fonctionnalité de moniteur de qualité de l'air : Lecture des Niveaux de PM2,5 HCHO, TVOC, AQI. Muni d'un écran LCD, l'utilisateur peut alors changer l'affichage entre les niveaux de PM2,5 HCHO, TVOC, et AQI avec un bouton du détecteur. Son usage est uniquement à l'intérieur d'un domicile et doit être placé à l'exterieur durant plus de six heures avant de

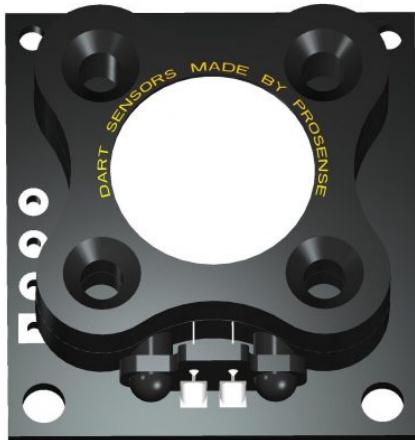
l'allumer pour pouvoir l'utiliser. Sa portabilité est limitée par sa capacité de la batterie batterie au lithium intégrée de 2200 mAh qui ne peut fonctionner que jusqu'à 6 heures. Il est donc recommandé pour un usage ménager de laisser connecter à une prise.

La plage de mesure est le suivant :

- (PM2,5) : 0 999  $\mu\text{g} / \text{m}^3$   
Résolution (PM2.5) : 0.1 $\mu\text{g} / \text{m}^3$
- (TVOC) : 0 5 mg /  $\text{m}^3$   
Résolution (TVOC) : 0.01mg /  $\text{m}^3$
- (HCHO) : 0 2 mg /  $\text{m}^3$   
Résolution (HCHO) : 0,01 mg /  $\text{m}^3$

L'exposition à longue durée du détecteur à un environnement ayant HCHO au dessus de 0.5mg/m<sup>3</sup> et PM2.5 au dessus 500mg/m<sup>3</sup>. Le détecteur doit être recalibré manuellement après trois mois d'utilisation. Les capteurs utilisés par le détecteur sont :

- Dart Electrochemical Sensor (pour HCHO)



Dart Sensors WZ-S formaldehyde module

Technical Specification

MODEL	WZ-S
Detection Principle	Micro fuel cell
Detectable Gas	HCHO
Detection Range	0-2ppm
Overload	10ppm
Input Voltage	5-7V
Warm up time	<3min
Response Time (T90)	<40S
Recovery Time (T10)	<60S
Resolution	0.001ppm
Operating temperature range	-20°C~50°C
Operating Humidity Range	10%~90%RH (non-condense)
Storage Condition	0~20°C
Lifetime	5 years in air
Warranty Period	12 months
Weight	4g

- Temtop Laser Particle Sensor (pour PM2.5)



Temtop utilise leur propre capteur laser pour détecter PM2.5. D'après le mode de fonctionnement de leur produit Temtop PM-900M Laser Particle Sensor :

Le capteur est basé sur la technologie d'analyse de la concentration des particules de la théorie de la diffusion de Mie. Une source de lumière laser éclaire une particule lorsqu'elle est tirée à travers la chambre de détection. Les particules se diffusent lors du passage à travers le faisceau laser. Le circuit de traitement de l'information convertit le signal d'intensité lumineuse de la lumière diffusée via le capteur photoélectrique en un signal d'impulsion de tension, qui est préamplifié et converti par A/D en signal numérique. Après traitement du bruit et analyse optimale de l'algorithme, le nombre de particules avec la taille de particules équivalente par unité de volume peut être obtenu, puis la concentration des particules est calculée.

### **III.1.4 Produits associés**

Il existe aussi dans le marché american (non livré en France) une version du détecteur connecté par Temtop, pour le prix de \$159.99. Le produit est identique au détecteur normal, avec l'ajout de l'application mobile à télécharger, permettant de consulter les données captées par le détecteur, ainsi que les prévisions météologiques sur l'application.

## III.2 Netatmo

(Capteur de température, humidité, CO2, et son)

### III.2.1 Produit



Station Météo Netatmo

Netatmo NWS01-US Aluminium station météo numérique

Prix : 165,19 €

Dimension (module intérieur) : 45\*45\*155 mm

Dimension (module extérieur) : 45\*45\*105 mm

Fréquence de capture : toutes les 5 minutes

Batterie (module intérieur) : adaptateur USB

Batterie (module extérieur) : 2 piles AAA (durée de vie de deux ans)

### III.2.2 Producteur

Netatmo est une entreprise française spécialisée dans les objets connectés. Elle est fondée en 2011, et a lancé 14 produits et accessoires dans la maison intelligente.

### III.2.3 Analyse

Netatmo propose avec ce produit deux stations météos en forme cylindrique connect, l'un pour l'exterieur (plus petit) et l'autre pour l'interieur (plus grand) de la maison. La difference entre les deux réside dans leur plage de mesure et le manque de sonomètre pour celle d'extérieure. Elle est néanmoins "water-proof" et est utilisés pour les prévisions météologiques de son application.

#### Station 'Interieur'

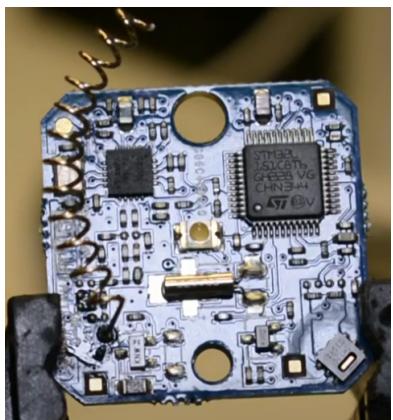
- Température : 0°C à 50°C / 32°F à 112°F; précision :  $\pm 0.3^\circ\text{C}$  /  $\pm 0.54^\circ\text{F}$
- Hygrométrie : 0 à 100%; précision :  $\pm 3\%$
- Capteur de CO2 : 0 à 5000 ppm; précision :  $\pm 50$  ppm (de 0 à 1000 ppm) ou  $\pm 5\%$  (de 1000 à 5000 ppm)
- Sonomètre : 35 dB à 120 dB

#### Station 'Exterieur'

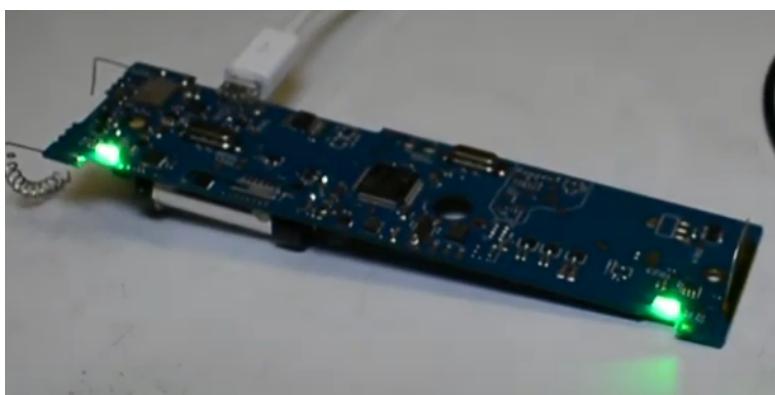
- Température : - 40°C à 65°C / - 40°F à 150°F; précision :  $\pm 0.3^\circ\text{C}$  /  $\pm 0.54^\circ\text{F}$
- Hygrométrie : 0 à 100%; précision :  $\pm 3\%$
- Baromètre : 260 à 1160 mbar / 7.7 à 37.2 inHg; précision :  $\pm 1$  mbar /  $\pm 0.03$  inHg

Les mesures capturées sont transmis sans fil à l'application smartphone, en utilisant le wi-fi (Compatible avec le Wi-Fi 802.11 b/g/n (2.4GHz) et Sécurités supportées : Ouvert/WEP/WPA/WPA2-personnalisé (TKIP et AES)). L'application affiche toutes les mesures capturées en temps réel des deux stations, et la prévisions météologiques locale des 7 jours suivants. Etant connecté, les données sont toutefois envoyées au serveur de Netatmo, pour qu'elles soient étudiées (et revendues évidemment).

Contrairement à Temtop, les valeurs ne sont pas affichés sur les stations. Elles possèdent tout simplement une lampe LED pour montrer l'état de la station.



La station extérieure est équipée de microcontrôleur STM32, où se trouve le chip de température et humidité, avec une surface exposée à l'extérieur, pour pouvoir capturer des mesures. Il y a aussi une antenne pour pouvoir transmettre des données, et une lampe LED indicatrice de bon fonctionnement des piles.



La station intérieure utilise une board plus longue pour contenir tous ses capteurs. Le microprocesseur STM32F103-Arm est toujours présent, avec le capteur de température et d'humidité (identique à celui de l'extérieur), ainsi que le capteur avec un mini-microparleur pour détecter les sons, deux antennes : pour le wifi et bluetooth, et un chip mémoire pour stockage des mesures.



Sur le dos du board, on retrouve un grand capteur pour CO2.

### III.2.4 Produit associé



Netatmo Capteur de qualité de l'air intérieur intelligent

Prix : 99,99 €

Dimension : 45\*45\*155 mm

Batterie : adaptateur USB

Fréquence de capture : toutes les 5 minutes

#### Détecteur Qualité de l'air

Il existe aussi un autre produit connecté au design identique au couleur rose de Netatmo, qui est 'centré' cette fois, la qualité de l'air. Ayant les mêmes fonctionnalités que la station de météo avec une application fournie, il mesure uniquement pour l'intérieur de la maison la température, laqualité de l'air (CO2 uniquement), l'humidité et le niveau sonore, pour le prix de 99,99 €

### **III.3 Flow/Flow 2**

(Déetecte (Particulate Matter) PM1, PM2.5, PM10, NO2(Nitrogen Dioxyde), COVs(composés organiques volatils))

#### **III.3.1 Produit**



détecteur Flow 2

Flow 2 - Capteur Personnel de Pollution de l'Air

Prix : 159,00 €

Dimension : 125\*40\*35 mm

Poids : 70g

Batterie : adaptateur USB

Durée de vie de la batterie : (24 à 72h selon mode d'utilisation)

#### **III.3.2 Producteur**

Flow et Flow 2 sont conçus par Plume Lab, un start-up français. Il se spécialise dans la surveillance de la qualité de l'air, avec leur application permettant de consulter l'exposition à la pollution en direct, et Flow, leur capteur innovant.

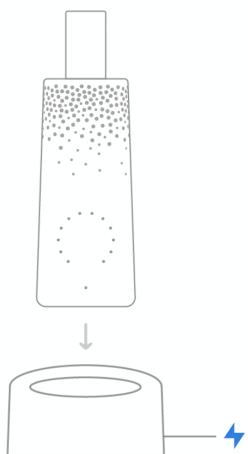
#### **III.3.3 Analyse**

Flow est un détecteur de qualité de l'air portable, accompagné de son application portable. C'est un produit particulièrement innovant et spécial par rapport aux produits précédents, du fait de sa petite taille et sa portabilité, axé donc sur une utilisation personnelle et pas uniquement pour la maison. Pesant uniquement 70g, l'utilisateur peut en effet le porter avec soi, en l'attachant par la sangle à un objet comme une porte-clé. Son point fort est donc sa mobilité par rapport aux autres moniteurs.



Sa petite taille implique aussi la miniaturisation des capteurs se trouvant à l'interieur. Les petits trou sur le détecteur, accompagné d'un mini-ventilateur permet l'accès de l'air à l'intérieur de celui-ci. Un capteur laser de particule se met à capturer les particules entrants en traduisant les rayons déviés en courant électrique qui peut être mesuré. Le capteur de NO<sub>2</sub> et COV détruit toute molécule NO<sub>2</sub> et COV, et la variation de l'énergie pour stabiliser la température de sa membrane lors de la destruction est constamment mesurée. Plume Lab réclame que la corrélation observée entre Flow 2 et les capteurs statiques de référence pour les polluants que Flow mesure est de 90 à 95%.

Les données stockées dans la clé USB du détecteur sont envoyés à l'application du portable via une connexion Bluetooth Low Energy (BLE). Avec le GPS de l'utilisateur, à la connexion 3G/4G de l'utilisateur, les données sont traitées et envoyées à un serveur qui fournit une carte d'activité de pollution en temps réel consultable via l'application.



Flow peut être chargé avec une station de charge USB, et est capable d'une durée de 24 à 72 heures d'autonomie, bien au dessus de sa compétition. Un mode 'sommeil' qui met en pause les captures est disponible pour économiser la batterie. La différence de Flow et Flow 2 est dans le design de détecteur avec la couleur noir pour Flow 2, blanc pour Flow, et avec Flow 2, les données et leurs localisations peuvent être envoyées par mail à l'utilisateur.

### **III.4 Etat du marché/ chiffre d'affaire**

Dans le marché international, on peut aisement constater que dans le domaine de maison intelligente et détecteur de qualité de l'air connecté, les produits français ont la cote. Netatmo et Plumb labs sont cités parmi les meilleurs 'smart' détecteurs de qualité de l'air maison disponible dans le marché (3dinsider.com, techhive.com) et des notes globalement positives sur leur page de vente sur Amazon. Cependant, en analysant les ventes, ils ont du mal à coexister dans le marché face à la concurrence américaine et chinois. Ici, en comparant les ventes uniquement sur amazon pour les trois produits dans la section 'Meilleurs vente en Moniteur de qualité de l'air Maison' :

- Temtop M10 Air Quality Monitor : #10
- Netatmo Smart Indoor Air Quality Monitor : #62
- Flow 2 - The Personal Air Quality Sensor by Plume Labs : #97

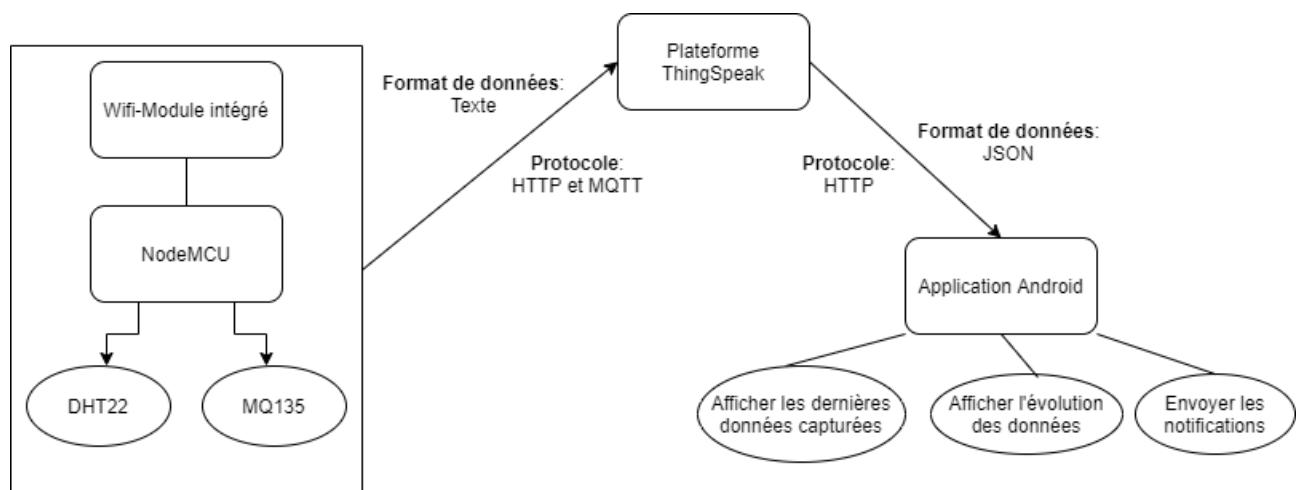
Le marché de moniteur de qualité de l'air étant relativement petit, on peut supposer les ventes faibles de Flow est dû à son produit trop neuf de la part d'un petit start-up français, et son utilité limitée (qui ne fait que détecter) pour son prix relativement élevé par rapport au détecteurs professionnels. Le produit de Netatmo est citée pour son bon design cylindrique, mais est aussi très critiqué avec des problèmes de connectivité récurrents avec l'application, et le fait qu'il détecte uniquement le CO<sub>2</sub> et non les autres particules aussi nocifs pour la santé. Enfin, on peut constater en générale que les objets connectés ne sont pas encore bien implémentés dans ce domaine avec les détecteurs 'classiques' très présents dans le haut du classement, mise à part les produits connectés de la firme Awair.

## **Section IV**

---

### **Solution proposée**

---



Nous proposons une station légère qui utilise les capteurs mq135 et dht22 qui sont connectés à la carte de développement, qui envoie les données à la plate-forme Thingspeak, la station est accompagnée d'une application Android.

## **Section V**

---

### ***Etude Fonctionnelle***

---

#### **V.1 La conception**

Dans l'objectif de pouvoir surveiller la pollution ainsi que la météo, notre projet se portera sur la conception d'une station météo portable et de détecteur de pollution. L'avantage important de notre station météo/détecteur de pollution par rapport au marché résidera sur sa petite taille et sa portabilité, ce qui différencie des stations fixes ou celles nécessitant une installation difficile sur place. Un utilisateur pourra alors poser notre station dans un endroit choisi à tout moment, dans le but de détecter la température et l'humidité grâce au capteur DHT-22, la qualité de l'air de la zone auquel se trouve la station grâce aux capteurs MQ-135, MQ-7, ainsi que la pollution sonore grâce au capteur sonore de notre station.

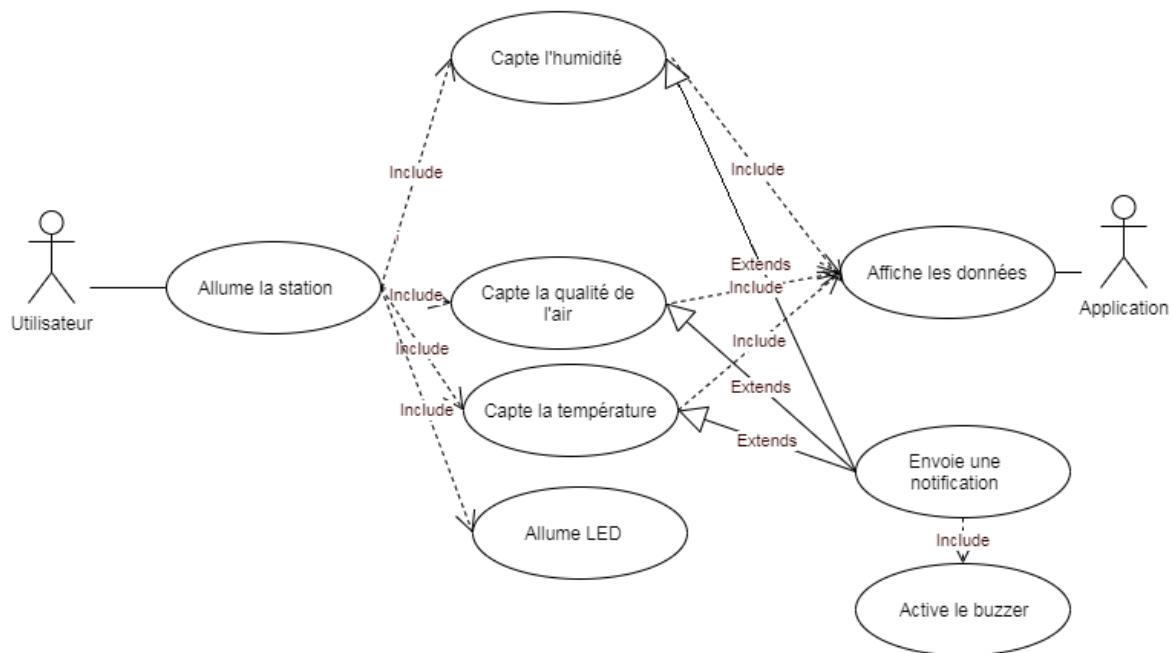
Dans un premier temps, notre station pourra signaler à l'utilisateur lors qu'un seuil au niveau de la température ou la pollution a été atteint, soit via une alarme sur place grâce à un buzzer se trouvant dans la station, soit à distance avec une notification via SMS à l'utilisateur. La station sera munie aussi de 3 LEDs pour distinguer les différents niveaux de pollution : vert si le niveau de pollution est faible, jaune ou orange si le niveau est modéré et rouge si c'est dangereux.

Dans un deuxième temps, l'utilisateur pourra ensuite surveiller et étudier en temps réel la météo et la pollution de la zone dans laquelle se trouve sa station, sur un ordinateur via une application. Cette application lui permettra en effet de consulter en temps réel les résultats des capteurs, ainsi que d'analyser l'évolution de la température et de la pollution dans la journée, dans le mois etc.

L'un des problèmes importants de ce projet sur lesquels nous devons résoudre par la suite est de décider comment la station doit être fabriquée. En effet, comme les capteurs doivent

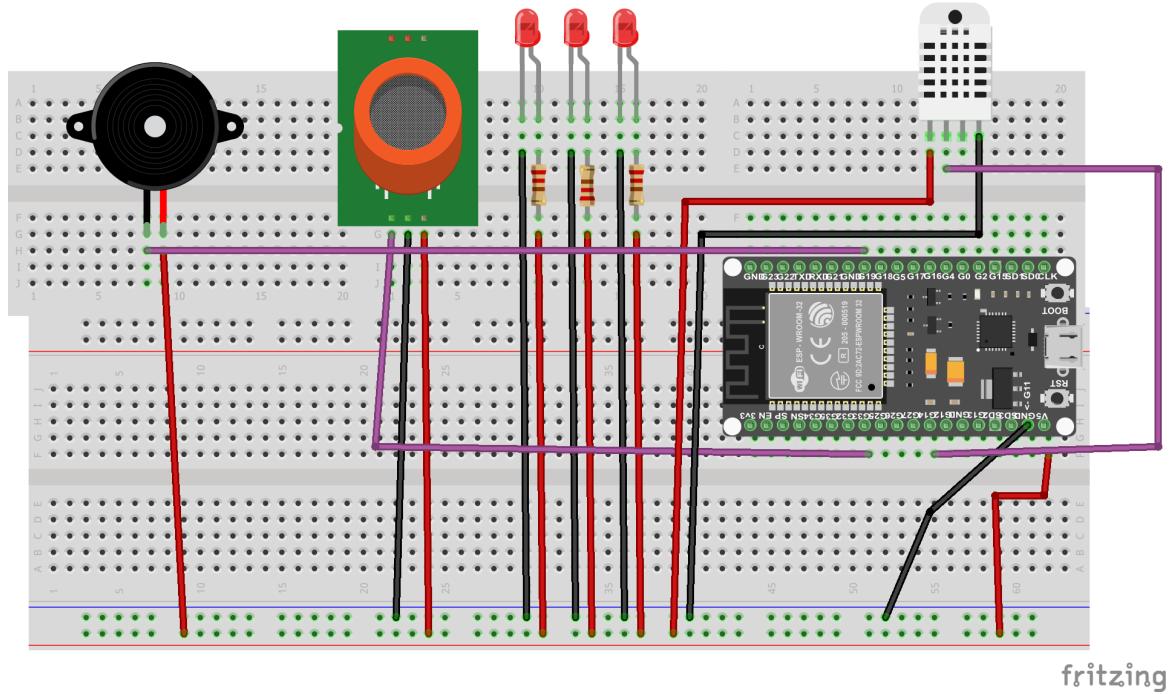
être exposés à l'air, nous devons penser à un moyen de les contenir de manière compacte et résistante à l'environnement de l'exterieur (le froid, la chaleur etc.) pour ne pas abîmer les capteurs. De ce fait, lorsque nous ne parviendrons pas à résoudre ce problème, notre station sera conçue pour être utilisée uniquement à l'intérieur d'une maison ou d'un bâtiment.

## V.2 Diagramme de Use Case

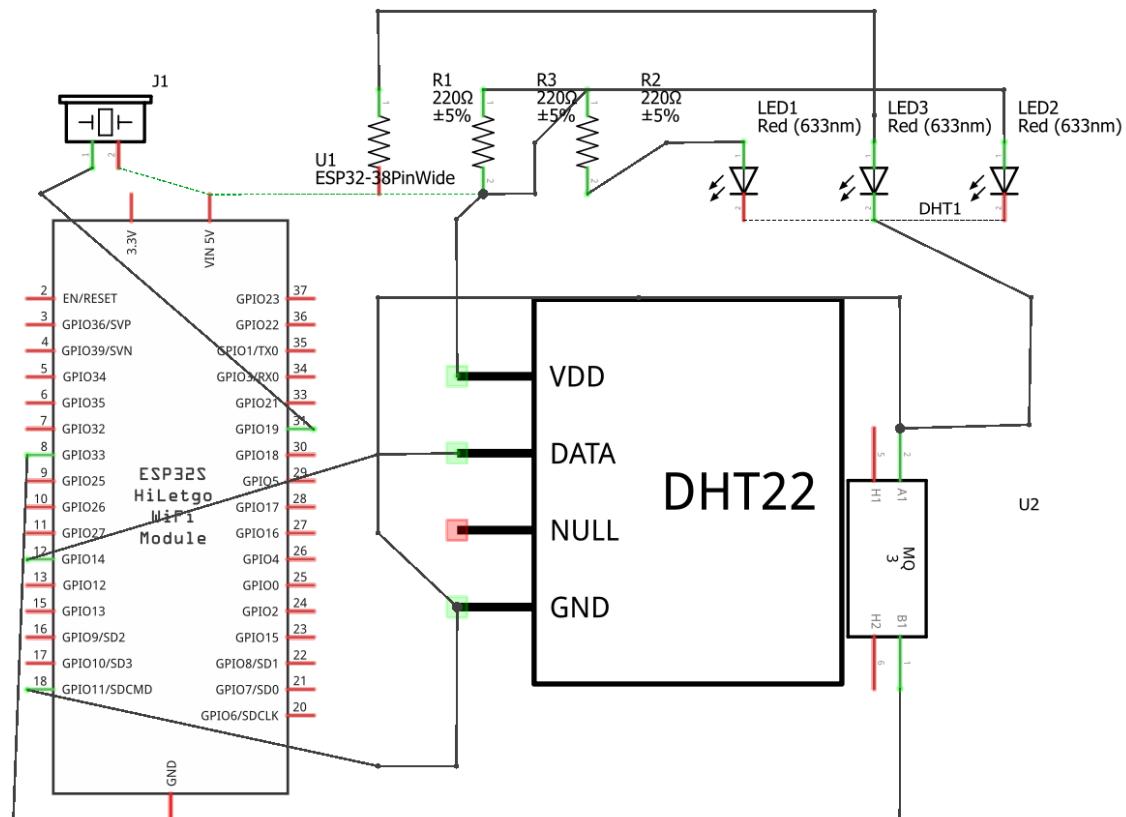


## V.3 Composants physiques

### V.3.1 Branchement



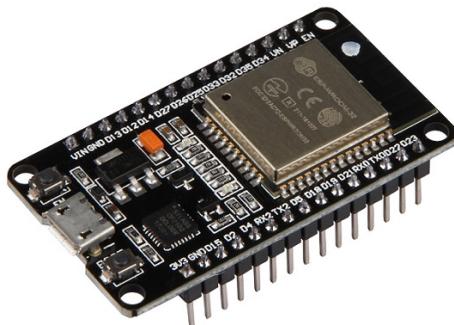
fritzing



fritzing

### V.3.2 Composants

#### 1. NodeMCU



#### Aperçu

La carte de développement utilisée dans ce projet est le nodeMCU32, avec un module wifi intégré, il est parfait pour nos besoins.

#### Connexion avec la plateforme ThingSpeak

Sous la plate-forme de ThingSpeak, nous pouvons créer des canaux qui peuvent recevoir et stocker des données à partir de différents appareils, dans notre cas, nous avons créé un canal composé de 3 champs (qualité de l'air, température, humidité). Ci-dessous vous pouvez voir le code utilisé pour établir la connexion :

```
#include "Arduino.h"
#include <WiFi.h>
#include <ThingSpeak.h>

...
...

//Parametres de connexion avec ThingSpeak
const char * API_KEY = "H8FPZCNB49AOMNIT";
unsigned long CHANNEL = 1022022 ;
const char *ssid = "HONOR 20 Lite";
const char *pass = "HonorableThief";
const char* server = "api.thingspeak.com";
WiFiClient client;

...
```

---

```
void setup() {
    ...
    ...
    WiFi.mode(WIFI_STA);
    ThingSpeak.begin(client);
}

...
if (WiFi.status() != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    while (WiFi.status() != WL_CONNECTED) {
        WiFi.begin(ssid, pass);
        Serial.print(".");
        delay(5000);
    }
    Serial.println("\nConnected.");
}
```

---

Pour nous connecter à la plateforme, nous utilisons deux bibliothèques : la bibliothèque wifi et la bibliothèque Thingspeak.

Nous utilisons l'identifiant unique du canal et la clé secrète de l'api ainsi que l'url du serveur, le nom et le mot de passe du réseau wifi.

### Envoi de données

les données sont envoyées au canal sous forme de 3 valeurs distinctes, voici le code :

---

```
int airRes = ThingSpeak.writeField(CHANNEL, 1, a, API_KEY);
delay(20000);
int tempRes = ThingSpeak.writeField(CHANNEL, 2, t, API_KEY);
delay(20000);
int humRes = ThingSpeak.writeField(CHANNEL, 3, h, API_KEY);
delay(20000);
// Write to the ThingSpeak channel
//check for errors
if (tempRes == 200) {
    Serial.println("temperature updated");
} else {
    Serial.println("Problem updating temperature field.
```

```

        HTTP error code " + String(tempRes));
    }

    if (humRes == 200) {
        Serial.println("humidity updated");
    } else {
        Serial.println("Problem updating humidity field.

        HTTP error code " + String(humRes));
    }

    if (airRes == 200){
        Serial.println("air quality updated");
    }else {
        Serial.println("Problem updating air quality

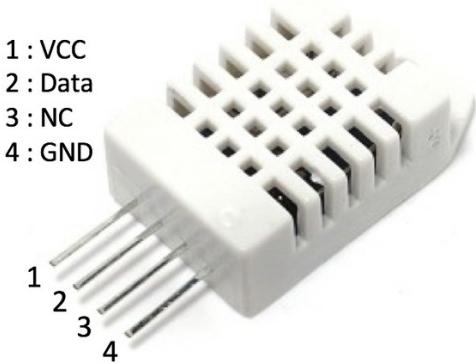
        field. HTTP error code " + String(airRes));
    }
    delay(10000);

```

---

Comme vous pouvez le voir ci-dessus, chaque valeur est écrite dans son propre champ et envoyée toutes les 20 secondes car c'est la limite imposée par la plate-forme, après ça nous écoutons le code de réponse pour voir si la valeur a été envoyée avec succès ou s'il y a eu un erreur.

## 2. DHT22



### Aperçu

Les capteurs DHT sont constitués de deux parties, un capteur d'humidité capacitif et

une thermistance, il y a aussi une puce très basique à l'intérieur qui fait une conversion analogique-numérique et crache un signal numérique avec la température et l'humidité.

### **Capture de température**

Grâce à la bibliothèque dht, la lecture des données du capteur est facile. Voici le code :

---

```
#include <DHT.h>
#include "Arduino.h"
...
#define DHTPIN 14
#define DHTTYPE DHT22 // DHT 11
DHT dht = DHT(DHTPIN, DHTTYPE);
...
// Read the temperature as Celsius:
float t = dht.readTemperature();
// Read the temperature as Fahrenheit:
float f = dht.readTemperature(true);
```

---

Nous définissons la pin analogique dht et la version dht du capteur dans notre cas c'est DHT22.

Nous appelons la méthode readTemperature() pour lire la température en deux unités : celcius et farenheit.

### **Capture d'humidité**

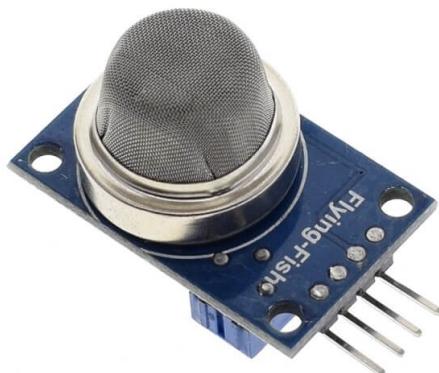
Pour lire l'humidité, nous appelons la méthode readHumidity(), voici le code :

---

```
// Read the humidity in %:
float h = dht.readHumidity();
```

---

## 3. MQ135



## Aperçu

Le MQ-135 est un capteur de gaz utilisé pour mesurer la concentration de gaz combustibles. Il a une conductivité plus faible dans l'air pur tandis que sa conductivité augmente avec la présence de gaz combustibles dans l'air. Le capteur est très sensible aux gaz comme l'ammoniac, le sulfure et la vapeur de benzène ,NH3, NOx, alcohol, smoke, CO2, etc.

## Calibration et capture de qualité de l'air

Par rapport à d'autres capteurs de qualité de l'air et de gaz sur le marché qui coûtent très cher, le MQ135 est très bon marché en comparaison et peut détecter la concentration de nombreux gaz nocifs différents dans l'air, cependant, tous ces avantages ont un coût, le capteur a besoin une période de préchauffage d'au moins 24 heures avant la première utilisation et il a une portée relativement courte (environ 10 mètres max) mais ce n'est pas un problème car dans notre projet, il est destiné à une utilisation en intérieur, en plus de cela, la valeur brute capturée par la sortie analogique du capteur n'est pas celle dont nous avons besoin, elle doit être convertie en PPM (parts per million), cela se fait à l'aide d'une formule mathématique. Voici les deux méthodes utilisées pour trouver la cancentration de gaz en PPM :

```
float calculateR0(){  
    float sensor_volt; //Define variable for sensor voltage
```

```

float RS_air; //Define variable for sensor resistance
float R0; //Define variable for R0
float sensorValue=0.0; //Define variable for analog readings
Serial.print("Sensor Reading = ");
Serial.println(analogRead(D33));

for(int x = 0 ; x < 500 ; x++) //Start for loop
{
    sensorValue = sensorValue + analogRead(D33); //Add analog values of sensor 500 times
}
sensorValue = sensorValue/500.0; //Take average of readings
Serial.print("Average = ");
Serial.println(sensorValue);
sensor_volt = sensorValue*(5.0/1023.0); //Convert average to voltage
RS_air = ((5.0*10.0)/sensor_volt)-10.0; //Calculate RS in fresh air
R0 = RS_air/3.7; //Calculate R0

Serial.print("R0 = "); //Display "R0"
Serial.println(R0); //Display value of R0
return R0 ;
}

float readPPMData(){
    float sensor_volt; //Define variable for sensor voltage
    float RS_gas; //Define variable for sensor resistance
    float ratio; //Define variable for ratio
    int sensorValue = analogRead(D33); //Read analog values of sensor
    Serial.print("SENSOR RAW VALUE = ");
    Serial.println(sensorValue);
    sensor_volt = sensorValue*(5.0/1023.0); //Convert analog values to voltage
    Serial.print("Sensor value in volts = ");
    Serial.println(sensor_volt);
    RS_gas = ((5.0*10.0)/sensor_volt)-10.0; //Get value of RS in a gas
    Serial.print("Rs value = ");
    Serial.println(RS_gas);

    ratio = RS_gas/calculateR0(); // Get ratio RS_gas/RS_air

    Serial.print("Ratio = ");
    Serial.println(ratio);
    float ppm_log = (log10(ratio)-b)/m; //Get ppm value in linear scale according to the the
}

```

---

```
float ppm = pow(10, ppm_log); //Convert ppm value to log scale
if(ppm < 100){
    greenFlash();
}else{
    if(ppm>100 && ppm<200){
        yellowFlash();
    }else if(ppm>200){
        redFlash();
    }
}
```

---

Après une longue recherche sur internet, et grâce à cette source [1], les deux méthodes donnent la valeur PPM appropriée.

Dans la deuxième méthode et en fonction de la valeur PPM, l'une des trois LED (verte, jaune, rouge) s'allumera.

**Comment le niveau de danger est déterminé ?** Le niveau de danger est déterminé en fonction des valeurs d'humidité, de température et de concentration des gaz nocifs.

L'indice de qualité de l'air que nous avons utilisé est le suivant :

Range(PPM)	Status
0-50	Good
51-100	Moderate
101-150	Unhealthy for sensitive people
151-200	Unhealthy
201-300	Very Unhealthy
301-500	Hazardous

Le taux d'humidité relative optimal pour le confort et pour éviter les effets sur la santé se situe entre 35 et 60 pourcent, toute valeur inférieure ou supérieure à cet intervalle a un effet négatif sur la santé.

En ce qui concerne les niveaux de température, nous avons considéré qu'une valeur supérieure à 32 ou inférieure à 10 peut être considérée comme inconfortable et malsaine.

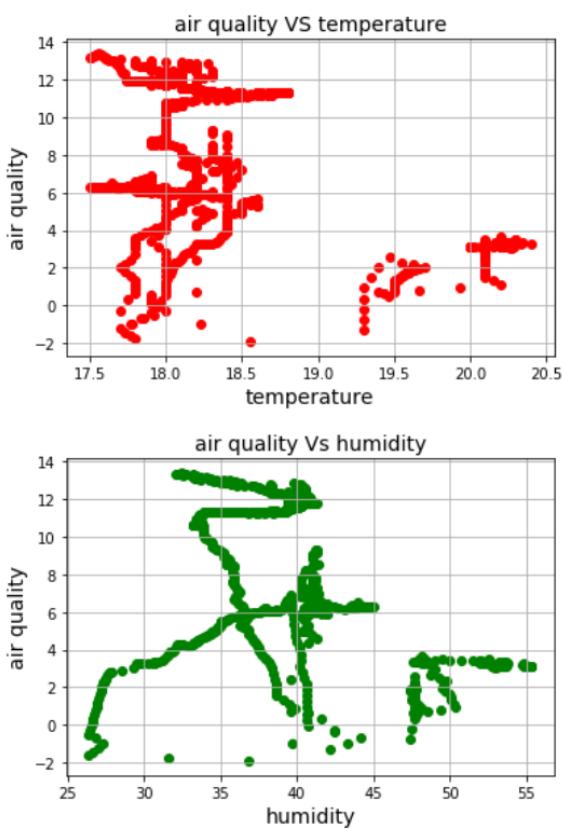
Si l'une de ces 3 valeurs dépasse leurs intervalles recommandés, une notification est envoyée via l'application.

## V.4 Analyse de données

Il y a eu de nombreuses études sur la relation entre la température, l'humidité et la qualité de l'air, et beaucoup d'entre elles ont constaté qu'il existe une forte corrélation entre la température et la pollution de l'air en général. Les inversions de température affectent la pollution de l'air car elles modifient la dynamique du mouvement de l'air. L'air chaud monte dans l'atmosphère car il est moins dense et donc plus flottant que l'air plus froid au-dessus de lui, cet effet d'étouffement emprisonne les polluants atmosphériques et permet d'augmenter leurs concentrations.

**application d'une régression multivariée sur les données capturées :**

En vérifiant s'il y avait une relation linéaire entre la variable dépendante (qualité de l'air) et les variables indépendantes (humidité et température), nous obtenons les résultats suivants :



Les 2 courbes indiquent qu'il n'y a pas de relation linéaire, entre les variables indépendantes et dépendantes, même lorsque nous appliquons le modèle multivarié, nous obtenons un message qui dit qu'il existe une forte multicolinéarité entre les variables, ce qui signifie qu'il existe d'autres facteurs qui peuvent affecter la qualité de l'air et pas seulement la température et l'humidité.

Cela peut être retracé au fait que nous avons utilisé un échantillon relativement petit de données.

## ***Section VI***

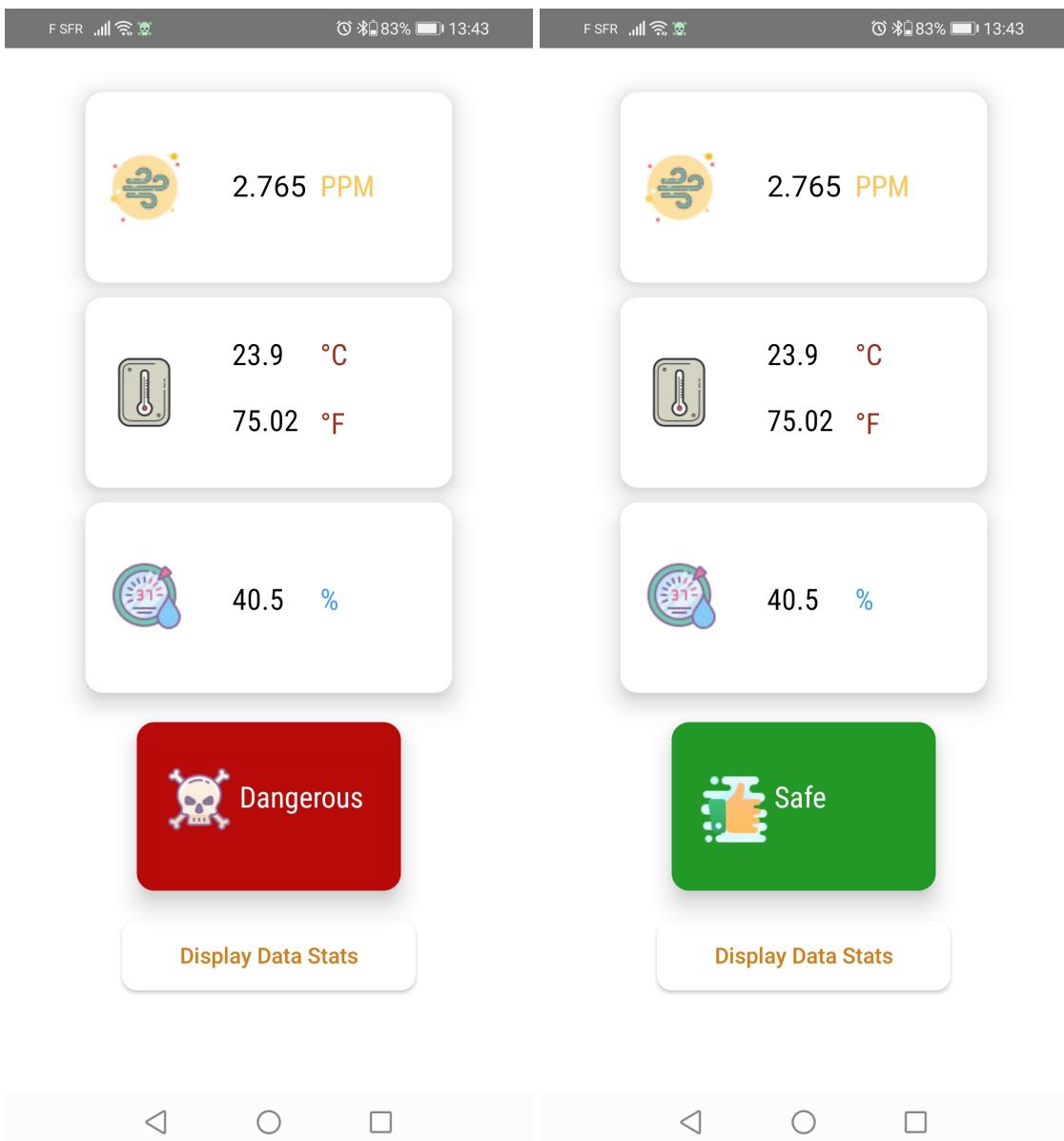
---

### ***Application smartphone***

---



Ecran de lancement de l'application.



L'application appelée «capitain planet» utilise la bibliothèque 'volley' pour récupérer des données en format JSON de la plateforme ThingSpeak sous le protocole http, puis il affiche les valeurs sur la page principale de l'application, comme vous pouvez le voir ci-dessus.

Voici le code de la méthode qui récupère la valeur de température et met à jour l'interface :

```
public void getTemperatureRecord(){
    String channel_temperature_url = "https://api.thingspeak.com/channels/1022022/
        fields/2.json";
    RequestQueue mQueue = Volley.newRequestQueue(MainActivity.this);
    StringRequest mRequest = new StringRequest(Request.Method.GET,
        channel_temperature_url, new Response.Listener<String>() {

```

```

@Override
public void onResponse(String response) {
    try {
        JSONObject object=new JSONObject(response);
        JSONArray array=object.getJSONArray("feeds");
        Log.d("feeds", ""+array);
        for(int i=array.length()-1;i>=0;i--) {
            JSONObject object1=array.getJSONObject(i);
            if(!object1.getString("field2").equals("null")){
                float f = Float.parseFloat(object1.getString("field2"));
                temp_progress_bar.setVisibility(View.INVISIBLE);
                temperature_text.setText(new DecimalFormat("###.###").format(f));
                c_indicator.setVisibility(View.VISIBLE);
                f = Float.parseFloat(fromCelciusToFeh
                    (Float.parseFloat(object1.getString("field2"))));
                temperature_text_f.setText(new DecimalFormat("###.###").format(f));
                f_indicator.setVisibility(View.VISIBLE);
                //Toast.makeText(getApplicationContext(),""+
                if(f<10 || f>32){
                    setDangerousLevel();
                }
                break ;
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
},
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Log.d("error", "onErrorResponse: ");
    }
});

mQueue.add(mRequest);
mQueue.start();

}

```

**Explication** Nous créons une requête à la plateforme ThingSpeak en fournissant l'url du champ spécifique que nous voulons récupérer, ici c'est la température (field2), nous recevons en conséquence un objet json contenant les valeurs enregistrées dans la plateforme, nous choisissons le dernier, après cela nous le convertir en fehrenhait parce que nous voulons afficher les deux unités sur l'interface.

Nous utilisons cette structure de code lors de la récupération des valeurs d'humidité et de qualité de l'air.

Les demandes de récupération sont envoyées toutes les 3 secondes à l'aide d'un Timer :

---

```
public void usingCountDownTimer() {  
    countDownTimer = new CountDownTimer(Long.MAX_VALUE, 3000) {  
        public void onTick(long millisUntilFinished) {  
            getAirQualityRecord();  
            getHumidityRecord();  
            getTemperatureRecord();  
        }  
  
        public void onFinish() {  
            start();  
        }  
    }.start();  
}
```

---



Il y a le bouton "Display Data Stats" sur la page principale, qui envoie à la page des évolutions des données. Sur cette page, l'utilisateur peut consulter l'évolution des trois données sous forme de graphe, ainsi que la localisation de la station.

L'application est capable de notifier l'utilisateur.



L'application continue de fonctionner en arrière-plan et en utilisant la fonctionnalité WorkManager, nous pouvons récupérer des données et envoyer des notifications, le tout en arrière-plan.

**Explication** L'API WorkManager facilite la planification de tâches asynchrones reportables qui devraient s'exécuter même si l'application se ferme ou que l'appareil redémarre, tout ce que nous devons faire est de définir une classe personnalisée qui étend le travailleur, puis d'effectuer les opérations à l'intérieur d'une méthode spécifique qui s'appelle 'doWork' :

```
public class NotificationsWorker extends Worker {  
    private String CHANNEL_ID = "555";
```

```
public NotificationsWorker(  
    @NonNull Context context,  
    @NonNull WorkerParameters params) {  
    super(context, params);  
    createNotificationChannel();  
  
    //prepareNotifications();  
}  
  
public String fromCelciusToFeh(float c){  
    return ""+((c*9)/5)+32;  
}  
//string request version  
public void getAirQualityRecord(){  
....  
}  
public void getTemperatureRecord(){  
....  
}  
public void getHumidityRecord(){  
....  
}  
  
private void createNotificationChannel() {  
....  
}  
  
public void prepareNotifications(String text,int typeOfDanger){  
    Intent intent = new Intent(getApplicationContext(),MainActivity.class);  
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);  
    PendingIntent pendingIntent = PendingIntent.getActivity(getApplicationContext(),  
    0, intent, 0);  
    NotificationCompat.Builder builder = new NotificationCompat.Builder  
(getApplicationContext(), CHANNEL_ID);  
    NotificationManagerCompat notificationManager = NotificationManagerCompat.from  
(getApplicationContext());  
    switch (typeOfDanger){  
        case 0:  
            //air quality  
            builder.setSmallIcon(R.drawable.green_death)  
    }  
}
```

```

.setContentTitle("Air Quality Alert")
.setContentText("air quality level at :" +text)
.setStyle(new NotificationCompat.BigTextStyle()
        .bigText("air quality level at :" +text))
.setPriority(NotificationCompat.PRIORITY_DEFAULT)
.setContentIntent(pendingIntent).setAutoCancel(true);

notificationManager.notify(0, builder.build());
break;
case 1:
//temperature
    builder.setSmallIcon(R.drawable.green_death)
    .setContentTitle("Temperature Alert")
    .setContentText("temperature level at: " +text)
    ..setStyle(new NotificationCompat.BigTextStyle()
        .bigText("temperature level at: " +text))
    .setPriority(NotificationCompat.PRIORITY_DEFAULT).setContentIntent
    (pendingIntent).setAutoCancel(true);

notificationManager.notify(1, builder.build());
break;
case 2:
//humidity
    builder.setSmallIcon(R.drawable.green_death)
    .setContentTitle("Humidity Alert")
    .setContentText("humidity level at: " +text)
    ..setStyle(new NotificationCompat.BigTextStyle()
        .bigText("humidity level at: " +text))
    .setPriority(NotificationCompat.PRIORITY_DEFAULT).setContentIntent
    (pendingIntent).setAutoCancel(true);

notificationManager.notify(2, builder.build());
break;
}

@Override
public Result doWork() {

```

```
getAirQualityRecord();
getHumidityRecord();
getTemperatureRecord();
return Result.success();
}
}
```

---

Les notifications sont créées à l'aide du NotificationsManager et envoyées à un canal spécifique géré par le système Android. Le travail en arrière-plan est effectué toutes les 10 à 15 minutes, c'est la limite imposée par le système Android pour des raisons d'optimisation.

---

## ***Bibliographie***

---

- [1] <https://jayconsystems.com/blog/understanding-a-gas-sensor>
- [2] <https://www.irceline.be/fr/documentation/faq/quappelle-t-on-particules-fines>