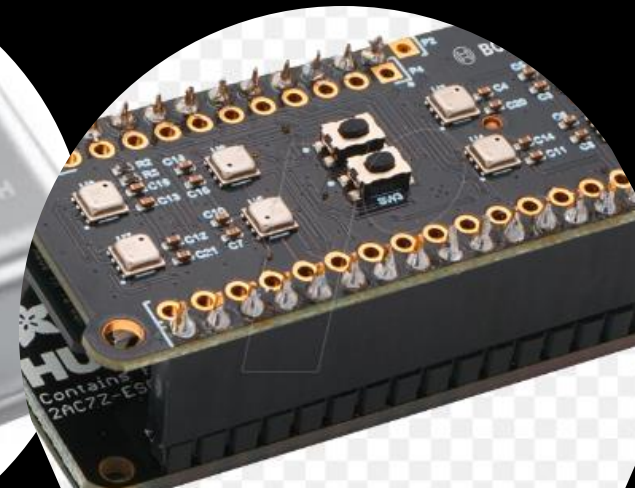
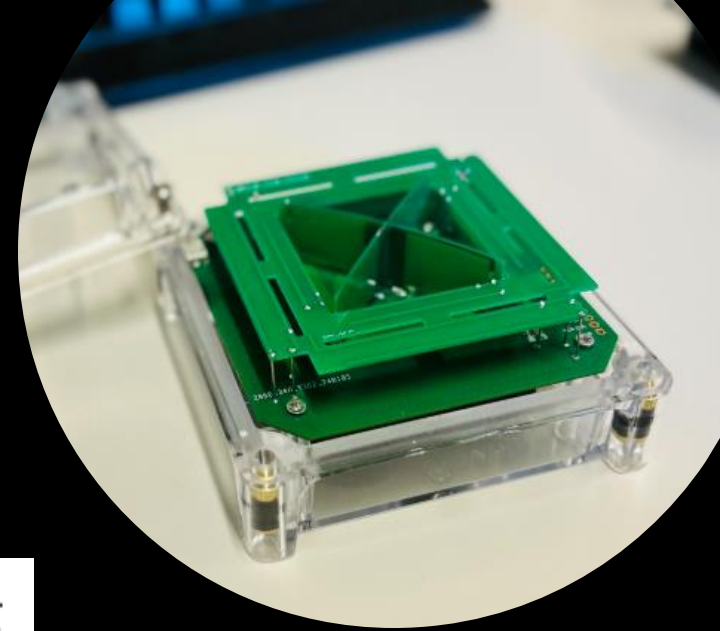


# Détecteur d'incendie



**Encadrant :** Fabien Ferrero

**Présenter par :**  
Hilman Bin Rokman  
Anissa Boumahdi

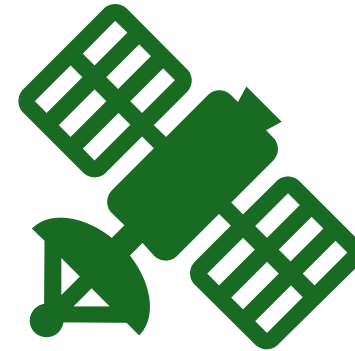
# La problématique

- Avoir des fausses alertes en présence de brouillade , et être dans l'obligation de remonter aux caméras surveillance .
- Comment concevoir et mettre en œuvre un système de détection d'incendie intelligent, basé sur l'utilisation de capteurs de température, de gaz et d'humidité, afin de garantir une détection fiable des incendies , tout en minimisant les fausses alertes et en s'adaptant aux variations des conditions d'environnements isoler.

# Objectif de notre projet :



**Surveillance des gaz dangereux** : Identifier les situations d'incendie ou de fuite de gaz grâce aux changements dans la composition des gaz.



**Facilité d'intégration** : Compatible avec de nombreux microcontrôleurs pour transmettre les données via la carte satellite EM2050.

# Détecteur de gaz (BME680) :

## Caractéristique principale :

**Capteur multi-gaz** : Détecte les gaz potentiellement dangereux tels monoxyde de carbone (CO), les oxydes d'azote (NOx)...

**Capteurs supplémentaires intégrés et leur Portée de mesure :**

température	humidité	Pression
-40°C à +85°C	0-100%	300-1100 hPa

**Consommation faible** : Mode optimisé pour les appareils à batterie.

**Interface** : Compatible pour une communication facile avec des microcontrôleurs .

# Les outils importants



- Programmer le microprocesseur
- Communication entre le satellite et les capteurs

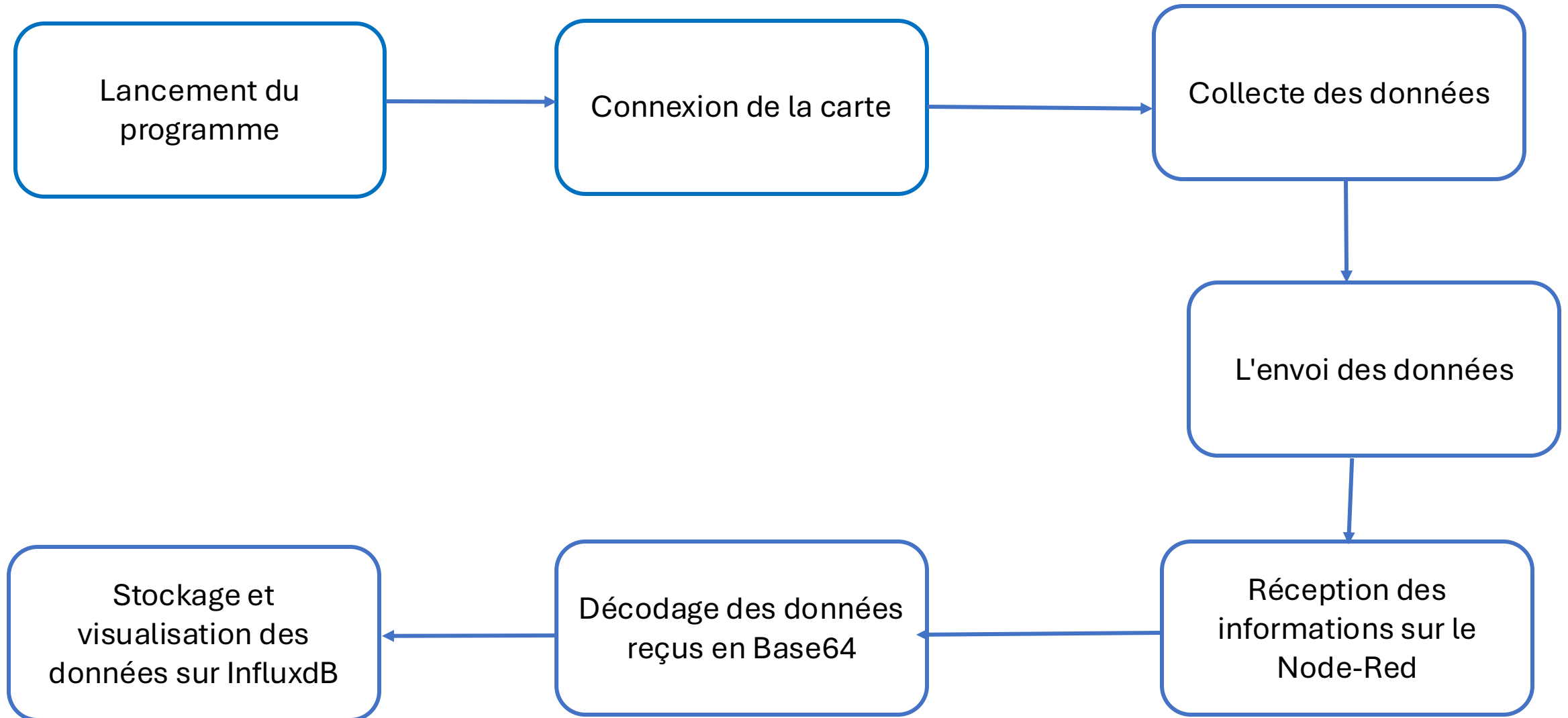


- Gérer les données envoyées par le satellite au serveur broker
- Stocker les données dans une base de données



- Sauvegarde et visualisation des données

# Processus :



```
if (!bme.begin(SENSORS_BME680_ADDRESS))
{
  USB_SERIAL.println(F("Could not find a valid BME680 sensor, check wiring!"));
  while (1)
  {
    ;
  }
}
```

```
unsigned long endTime = bme.beginReading();
if (endTime == 0)
{
  USB_SERIAL.println(F("Failed to begin reading :("));
  return;
}
delay(50);
if (!bme.endReading())
{
  USB_SERIAL.println(F("Failed to complete reading :("));
  return;
}
```

```
USB_SERIAL.print(F("Temperature = "));
USB_SERIAL.print(bme.temperature);
USB_SERIAL.println(F(" *C"));
USB_SERIAL.print(F("Pressure = "));
USB_SERIAL.print(bme.pressure / 100.0);
USB_SERIAL.println(F(" hPa"));
USB_SERIAL.print(F("Humidity = "));
USB_SERIAL.print(bme.humidity);
USB_SERIAL.println(F(" %"));

USB_SERIAL.print(F("Gas = "));
USB_SERIAL.print(bme.gas_resistance / 1000.0);
USB_SERIAL.println(F(" KOhms"));

delay(10000);
```

## Mise en place de la partie programmation

- L'initialisation du capteur BME680
- Lecture des données prélevées par le capteur
- Verification via le moniteur série

# Alerte d'incendie

- En air propre, la résistance du capteur est élevée, généralement supérieure à 100 k $\Omega$  voire 1 M $\Omega$  cela dépend de l'environnement.
- En présence de gaz polluants, cette résistance diminue progressivement pour atteindre quelques k $\Omega$  à quelques dizaines de k $\Omega$  selon la concentration.
- Lorsqu'elle descend en dessous du seuil critique de 50 k $\Omega$ , cela indique une dégradation significative de la qualité de l'air, envoyant ainsi l'alerte pour signaler un danger potentiel.

14/02/2025 10:59:41 noeud: feaefa47.e344d8

msg.payload : Object

▼ *object*

temperature: 25.96

pression: 1003.9

humidité: 31.04

gaz: 90.84

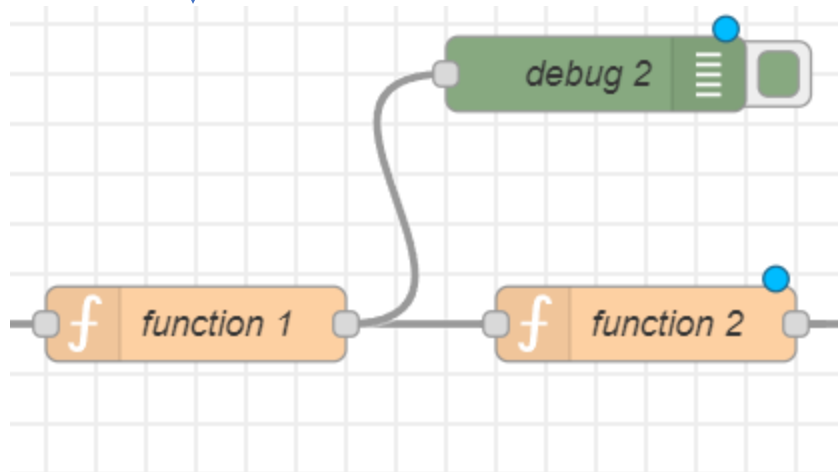
---



# Résultat d'envoi :

Une fois la carte connectée, le système envoie les informations désirer avec un écart de 10 seconds

Intégration d'une fonction directement sur le Node-Red pour le décodage des données reçu par la carte (base\_64) pour une visualisation directe.



```
msg.payload : Object
  ▶ { temperature: 12.6, pression: 1025.63,
    humidité: 58.06, gaz: 97.17 }

21/02/2025 09:44:55 noeud: debug 2
cma/echo/1179ec/uplink : msg.payload : string[26]
"12.61 1025.63 58.53 97.06 "

21/02/2025 09:44:55 noeud: feaefa47.e344d8
msg.payload : Object
  ▶ { temperature: 12.61, pression: 1025.63,
    humidité: 58.53, gaz: 97.06 }

21/02/2025 09:45:06 noeud: debug 2
cma/echo/1179ec/uplink : msg.payload : string[26]
"12.62 1025.62 58.70 97.50 "

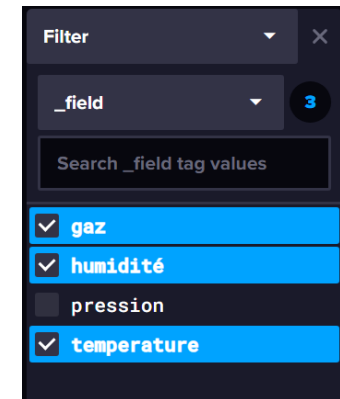
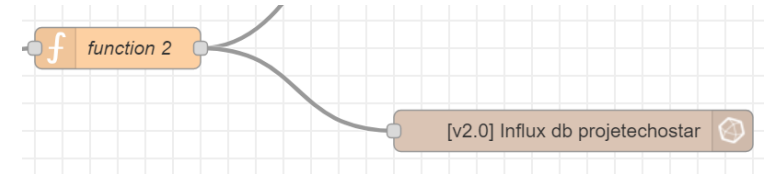
21/02/2025 09:45:06 noeud: feaefa47.e344d8
msg.payload : Object
  ▶ { temperature: 12.62, pression: 1025.62,
    humidité: 58.7, gaz: 97.5 }

21/02/2025 09:45:16 noeud: debug 2
cma/echo/1179ec/uplink : msg.payload : string[26]
"12.63 1025.62 58.36 97.12 "

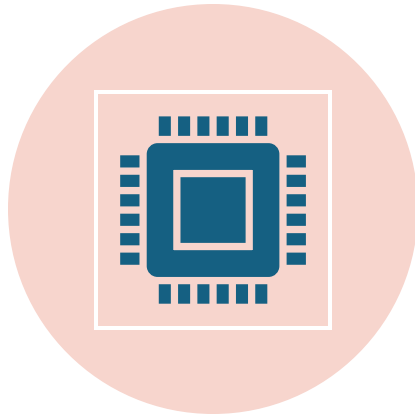
21/02/2025 09:45:16 noeud: feaefa47.e344d8
msg.payload : Object
  ▶ { temperature: 12.63, pression: 1025.62,
    humidité: 58.36, gaz: 97.12 }
```

# Enregistrement et visualisation des données

- Utilisation d'un noeud influxdB pour sauvegarder les données.
- Recupération des données en utilisant leurs clés (JSON).
- L'affichage en temps réel les informations sur la température, l'humidité et le gaz.



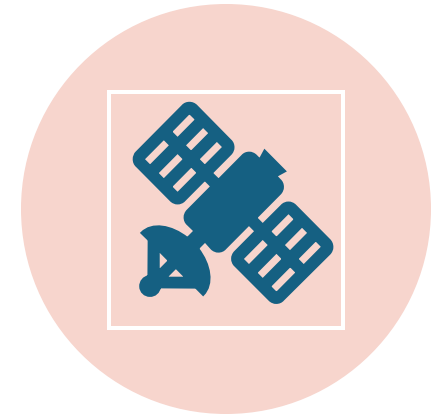
# Conclusion



AVOIR UNE CARTE AVEC UN CAPTEUR  
DE GAZ INTÉGRER POUR NE PLUS  
AVOIR LES FAUSSENT ALERTES.



CRÉER UN SYSTÈME PERMETTANT DE  
DÉTECTER LES INCENDIES DANS UN  
ENVIRONNEMENT ISOLÉ.



S'ASSURER UNE BONNE  
COMMUNICATION ENTRE LE  
SATELLITE ET LA CARTE.