



**Ecole Nationale d'Ingénieur de Sousse
Université de Sousse
Année universitaire 2022– 2023**



Département Electronique Industrielle

Rapport du projet semestriel

Système de surveillance de la pollution de l'air

Réalisé par

Ammari Rafik

Bourkhis Iheb

Boukadida khairieddine

Hawala Mohamed Ali

Filière

1er Année Génie Electronique Industrielle

Encadré par

Mr. Bouraoui Ouni

Remerciements

Au terme de ce travail, nous tenons à présenter nos vifs remerciements et tous nos respects à toutes les personnes qui ont contribué, de près ou de loin, à son déroulement, et qui nous ont encouragées, soutenues et confortées tout au long de la réalisation de ce rapport.

On souhaite renouveler nos remerciements à notre encadrant, Mr Bouraoui Ouni, pour la confiance. Qu'elle nous a témoignée du début jusqu'à la fin du travail, pour sa disponibilité à comprendre et à communiquer. Sincèrement, c'est grâce à elle on a pu développer beaucoup mieux notre problématique.

On ne peut conclure ces remerciements sans exprimer toute notre reconnaissance aux membres du jury pour avoir accepté d'évaluer notre travail.

Table de matières

Introduction générale.....	6
Chapitre 1.....	8
Cahier de charges	8
I-Introduction	8
II-Problématique	8
III-Description du cahier de charges	8
1-Schéma synoptique	8
Chapitre 2.....	12
Etude de la partie matérielle et logicielle (Hardware et software).....	12
I-Introduction	12
II-Le hardware	12
1-Carte ESP8266 Nodemcu :	12
2-Capteur de gaz MQ2 :	13
3-Module GPS NEO 6M :	15
4-Module carte mémoire (Micro Sd Card Adapter) :	15
5-Fils de connexions :	16
6-Une plaque à essai :	16
III-Le software :	17
1-Isis Proteus 8 Pro :	17
2-Arduino IDE :	17
3-Firebase :	18
4-HTML :	19
4-CSS :	19
4-JAVASCRIPT :	20
4-REACT.JS:	21
5-JSX :	22
5-API DE GOOGLE MAPS :	22
VI-Conclusion	23
Chapitre 3	25
Réalisation du projet.....	25
I-Introduction	25
II-Phase de conception électronique	25
III-Phase de codage	28
1-Programmation via Arduino :	28

2-stockage et visualisation de données via firebase.....	26
VI-Conclusion	29
Conclusion générale	30

Listes de figures

Figure 1. 1 : schéma synoptique	9
Figure 2.1 : la nodemcu.....	13
Figure 2. 2 : capteur de gaz MQ2.....	14
Figure 2. 3 : courbe des gaz de mq2.....	14
Figure 2. 4 : GPS NEO 6M	15
Figure 2. 5 : module carte sd	16
Figure 2. 6 : Fils de connexions	16
Figure 2. 7 : plaque à essai.....	17
Figure 2. 8 : Isis Proteus	17
Figure 2. 9 : Arduino IDE.....	18
Figure 2. 10 : Firebase.....	19
Figure 2. 11 : HTML IDE	19
Figure 2. 12 : CSS IDE	20
Figure 2. 13 : JAVASCRIPT IDE	21
Figure 2. 14 : REACT JS	22
Figure 2. 15 : JSX	22
Figure 2. 16 : Google maps	23
Figure 3. 1 : conception de la carte	26
Figure 3. 2 : Pcb de la carte	27
Figure 3. 3 : Montage 3D	27
Figure 3. 4 : réalisation de la carte.....	28
Figure 3. 5 : test sous plaque à essai.....	28
Figure 3. 6 : test finale suer la carte.....	29

Introduction générale

Le présent rapport de projet semestriel a pour objectif de présenter la conception et la réalisation d'une carte électronique capable d'envoyer des données de localisation GPS Neo 6M et de capteur de gaz MQ2 à une base de données Firebase. Cette base de données est ensuite utilisée pour afficher les données en temps réel sur une page web dynamique.

Le projet a été réalisé dans le cadre d'un semestre universitaire, En mobilisant les connaissances théoriques acquises lors du premier semestre sur la conception électronique, ainsi qu'en travaillant sur le développement web et la gestion de bases de donnée

Le but était de concevoir une carte électronique capable de collecter et d'envoyer des données à une base de données en temps réel, tout en fournissant une interface utilisateur conviviale pour visualiser les données collectées.

Dans les sections suivantes, nous allons décrire les différentes étapes du projet, depuis la conception de la carte électronique jusqu'à l'implémentation de la base de données Firebase et de l'interface utilisateur web. Nous discuterons également des résultats obtenus et des limitations du système, ainsi que des perspectives d'avenir pour son développement.

Chapitre 1 :

Cahier de charges

Chapitre 1 :

Cahier de charges

I-Introduction

Dans ce chapitre nous allons présenter le cahier de charges, ce cahier de charges décrit les besoins des individus pour la conception d'une carte électronique qui collecte des données de localisation GPS et de capteur de gaz en temps réel pour identifier les zones polluées. Les entreprises peuvent utiliser cette technologie pour surveiller les niveaux de pollution de l'air dans les zones environnantes et prendre des mesures pour limiter les émissions nocives. Les personnes souhaitant acheter ou louer un terrain peuvent également bénéficier de ces informations pour prendre des décisions éclairées quant à leur investissement. Le cahier des charges définit les spécifications techniques de la carte électronique et l'interface utilisateur web pour visualiser les données collectées. Ainsi on va s'intéresser aux différents diagrammes et tableaux permettant d'expliquer le fonctionnement du système

II-Problématique

Comment concevoir une carte électronique capable de collecter des données de localisation GPS et de capteur de gaz en temps réel pour identifier les zones polluées, tout en répondant aux exigences de précision et de sensibilité nécessaires pour la surveillance de la qualité de l'air et de l'environnement, et en proposant une interface utilisateur conviviale pour visualiser les données collectées sur une page web dynamique ?

III-Description du cahier de charges

1- Schéma synoptique :

Le schéma synoptique suivant va nous permettre de mieux comprendre le fonctionnement global du système étudié :

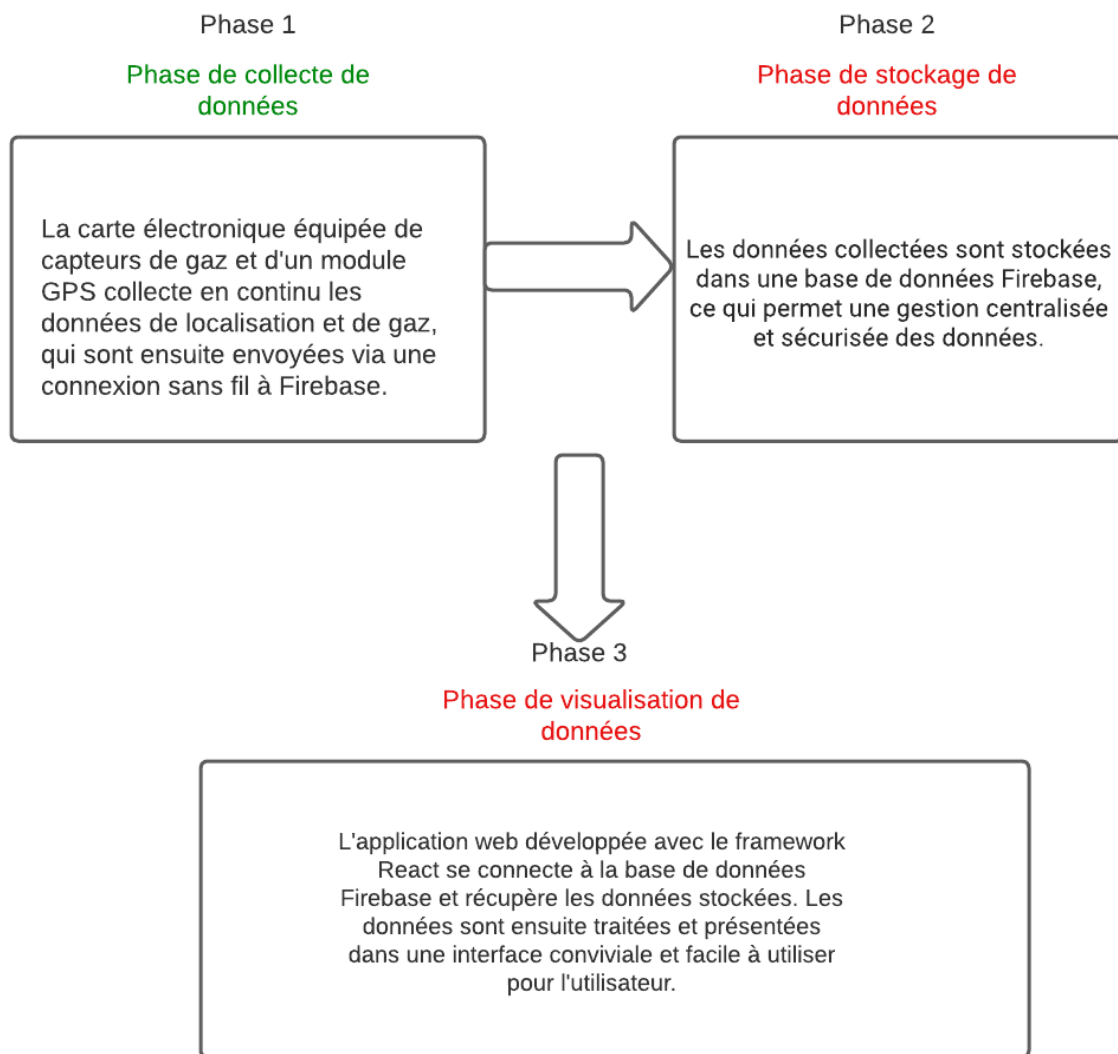


Figure 1. 1 : schéma synoptique

IV- Conclusion :

En conclusion, le présent cahier des charges a permis de définir les besoins des individus pour le projet de conception d'une carte électronique permettant de collecter des données de localisation GPS et de capteur de gaz en temps réel pour identifier les zones polluées. La qualité de l'air et de l'environnement étant une préoccupation croissante pour les entreprises et les particuliers, la conception de cette carte électronique peut contribuer à améliorer la surveillance de la qualité de l'air et à informer les décisions d'investissement en matière de propriétés immobilières.

La prochaine étape du projet consiste à concevoir et à développer la carte électronique en

fonction des spécifications techniques définies dans le cahier des charges, tout en garantissant que l'interface utilisateur web soit bien conçue pour répondre aux besoins des utilisateurs.

Chapitre 2 :
Etude de la partie
Matérielle et logicielle
(Hardware et Software)

Chapitre 2 :

Etude de la partie matérielle et logicielle (Hardware et software)

I-Introduction :

Dans le deuxième chapitre nous allons présenter le hardware et le software utilisé durant ce projet.

II-Le hardware :

1-Carte ESP8266 Nodemcu :

Le NodeMCU est une carte de développement basée sur le module ESP8266, un microcontrôleur Wi-Fi développé par Espressif Systems. Cette carte est conçue pour être facile à utiliser avec une interface USB intégrée pour la programmation et la communication série.

La carte NodeMCU dispose d'un processeur ESP8266 cadencé à 80 MHz, d'une mémoire flash de 4 Mo, de 4 Mo de mémoire vive (RAM) et prend en charge la connectivité Wi-Fi. Elle est équipée de broches d'E/S (Entrée/Sortie) compatibles avec le standard Arduino, ce qui la rend compatible avec une grande variété de capteurs et d'autres modules électroniques. La carte NodeMCU est souvent utilisée dans des projets de prototypage rapide pour des applications IoT (Internet des objets) ou des projets connectés à Internet. Elle peut être programmée avec l'IDE Arduino ou avec des langages de programmation tels que Lua et MicroPython, offrant ainsi une grande flexibilité dans le choix de l'environnement de développement.

Ainsi la carte utilisée est représenté ci-dessous :

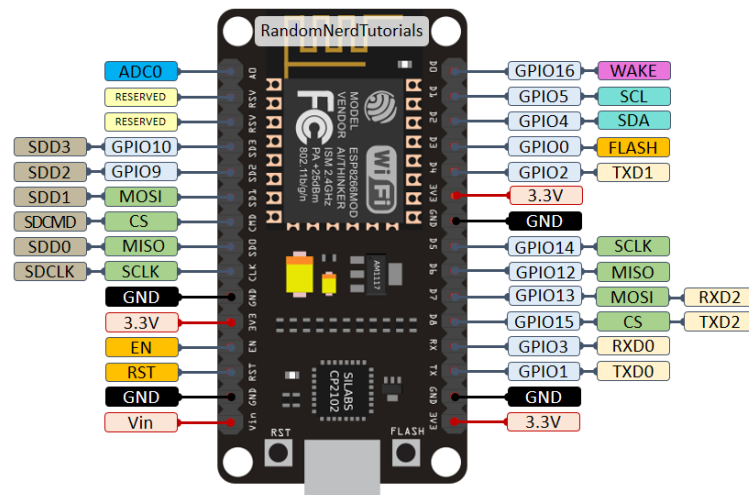


Figure 2.1 : la nodemcu

2-Capteur de gaz MQ2 :

Le capteur MQ-2 est un détecteur de gaz utilisé pour mesurer la présence de gaz inflammables, de gaz liquéfiés, de fumée et de gaz toxiques dans l'air. Il est souvent utilisé dans des projets d'Internet des objets (IoT) pour surveiller la qualité de l'air dans les bâtiments, les maisons, les usines ou les espaces publics.

Le capteur MQ-2 est basé sur la technologie de détection de gaz par résistance. Il est composé d'un élément chauffant et d'un élément de détection de gaz. L'élément chauffant est utilisé pour chauffer le capteur à une température spécifique, tandis que l'élément de détection de gaz mesure les changements de résistance causés par la présence de gaz.

Le capteur MQ-2 peut détecter plusieurs types de gaz inflammables et toxiques, tels que le méthane, l'éthane, le propane, l'alcool, l'hydrogène, le monoxyde de carbone, l'hydrogène sulfuré, etc. Il fournit une sortie analogique proportionnelle à la concentration de gaz dans l'air. Plus la concentration de gaz est élevée, plus la sortie analogique est élevée.

Ainsi le MQ2 est donné par la figure suivante :

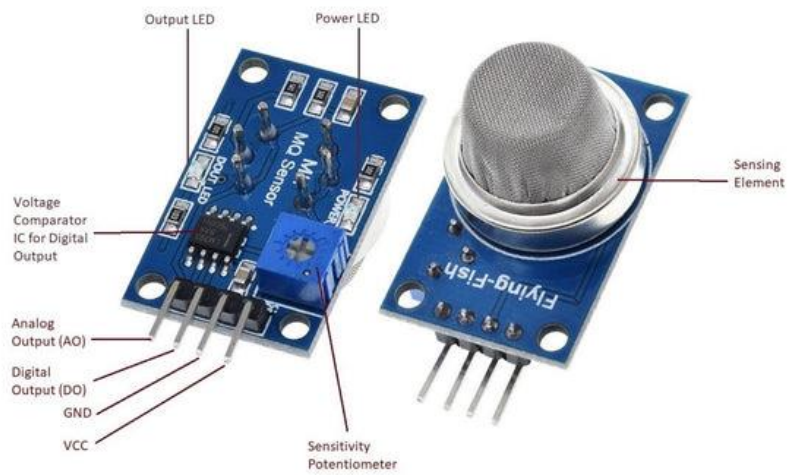


Figure 2. 2 : capteur de gaz MQ2

Les 7 types de gaz pouvant détecté par ce capteur sont représentés par la figure suivante :

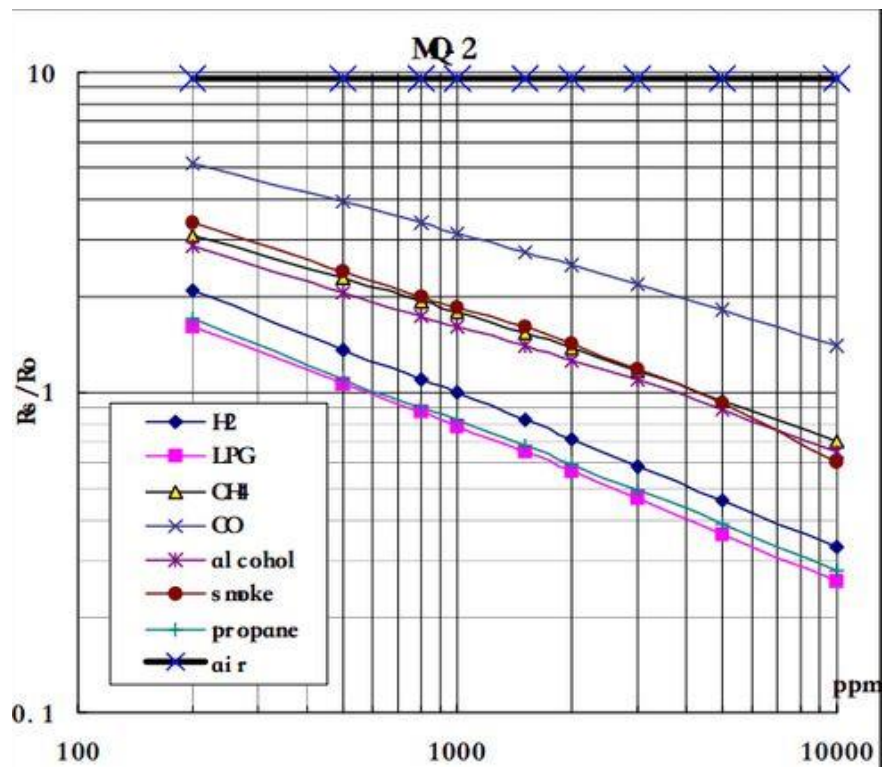


Figure 2. 3 : courbe des gaz de mq2

N.B :

1 mg/L = 1 partie par million (ppm) pour les solutions aqueuses diluées. Par exemple, une concentration de chlore de 1,8 mg/L équivaut à 1,8 ppm de chlore.

3-Module GPS NEO 6M :

Le GPS NEO-6M est un module de positionnement GPS compact et abordable conçu pour être intégré dans des projets électroniques. Il est largement utilisé dans les applications de navigation et de surveillance de véhicules, les drones, les projets IoT et les systèmes de géolocalisation en temps réel.

Le GPS NEO-6M est basé sur la technologie de positionnement par satellite et peut recevoir des signaux de satellites pour déterminer la position géographique, la vitesse et l'heure. Il utilise une puce de traitement GPS de haute précision pour assurer une acquisition rapide du signal GPS et un positionnement précis.

Le module NEO-6M est capable de recevoir des signaux de GPS, GLONASS, SBAS et QZSS, ce qui le rend compatible avec une large gamme de systèmes de navigation par satellite. Il dispose également d'une antenne céramique intégrée qui permet une réception optimale des signaux GPS.



Figure 2. 4 : GPS NEO 6M

4-Module carte mémoire (Micro Sd Card Adapter) :

Le module de carte mémoire microSD se connecte au NodeMCU via l'interface SPI (Serial Peripheral Interface), qui permet une communication série synchrone entre le microcontrôleur et le module de carte mémoire. Les pins nécessaires à la connexion de la carte mémoire sont disponibles sur le NodeMCU, permettant ainsi de réaliser une connexion facile.

Une fois que la carte mémoire est insérée dans l'emplacement prévu à cet effet sur le

NodeMCU, il est possible de stocker et de lire des fichiers à partir de celle-ci en utilisant des fonctions spécifiques dans le code du projet. Cette fonctionnalité peut être utilisée pour stocker des données de capteurs, des fichiers de configuration, des images et des vidéos, et peut être utile pour une grande variété d'applications IoT.

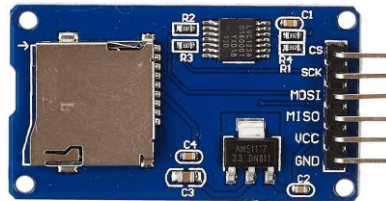


Figure 2. 5 : module carte sd

5-Fils de connexions :

Nous avons utilisé des fils de connexions male/femelle et male/male



Figure 2. 6 : Fils de connexions

6-Une plaque à essai :

Avant de créer la carte PCB, nous devons d'abord effectuer des tests. Pour cela, il est nécessaire d'utiliser une plaque d'essai, également appelée breadboard ou protoboard. Cette plaque est composée d'orifices qui sont électriquement connectés entre eux de manière interne. Elle permet l'insertion d'éléments électroniques et de fils pour le prototypage et le montage de circuits électroniques.

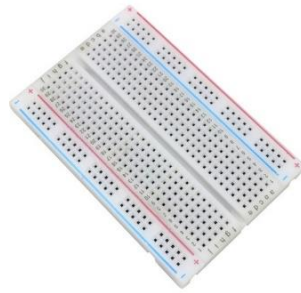


Figure 2. 7 : plaque à essai

III-Le software :

1-Isis Proteus 8 Pro :

Dans le cadre de ce projet, ISIS Proteus pourrait être utilisé pour concevoir le schéma électronique de la carte électronique de surveillance de la pollution de l'air et simuler son fonctionnement avant la fabrication. Les concepteurs pourraient également utiliser ISIS Proteus pour vérifier que la carte électronique respecte les spécifications techniques définies dans le cahier des charges.

ISIS Proteus offre également la possibilité de créer des modèles de simulation pour les capteurs de gaz et les modules GPS, ce qui permettrait aux concepteurs de simuler le fonctionnement de ces composants avant l'intégration dans la carte électronique.

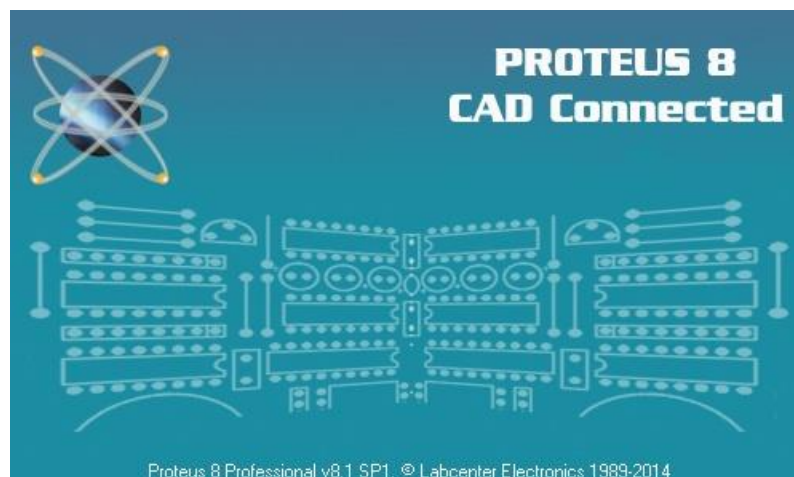


Figure 2. 8 : Isis Proteus

2-Arduino IDE :

Dans le cadre de ce projet, Arduino pourrait être utilisé pour programmer la carte électronique

de surveillance de la pollution de l'air, en lui donnant la capacité de collecter des données de localisation GPS et de capteur de gaz en temps réel. Les entrées/sorties numériques et analogiques de la carte Arduino pourraient être utilisées pour interagir avec les capteurs de gaz et les modules GPS, en collectant les données et en les transmettant à l'ordinateur central pour l'analyse.

D'autre part Arduino offre la possibilité de collecter des données de localisation GPS et de capteur de gaz en temps réel, et de les transmettre vers une plateforme de stockage en ligne telle que Firebase pour une analyse ultérieure. Cela permet de surveiller la pollution de l'air de manière efficace et de prendre des mesures pour limiter les émissions nocives.



Figure 2. 9 : Arduino IDE

3-Firebase :

Firebase est une plateforme de développement d'applications mobiles et web proposée par Google. Elle propose différents services, notamment de stockage de données en temps réel, d'authentification, de messagerie, d'hébergement et d'analyse. Dans le cadre de ce projet, Firebase a été utilisé comme une base de données en temps réel pour stocker les données collectées par la carte électronique de surveillance de la pollution de l'air. Les données ont ensuite été récupérées par une application web développée avec Firebase, permettant une visualisation en temps réel des données de pollution de l'air collectées par la carte électronique. Firebase a donc joué un rôle clé dans la mise en œuvre de ce projet, en permettant de stocker, récupérer et afficher les données collectées de manière claire et compréhensible.



Figure 2. 10 : Firebase

4-HTML :

L'HyperText Markup Language, HTML, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. Il permet, entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu.

L'HTML permet à un créateur de sites Web de gérer la manière dont le contenu de ses pages Web va s'afficher sur un écran, via le navigateur. Il repose sur un système de balises permettant de titrer, sous-titrer, mettre en gras, etc., du texte et d'introduire des éléments interactifs comme des images, des liens, des vidéos...



Figure 2. 11 : HTML IDE

4-CSS :

Le CSS pour Cascading Style Sheets, est un langage informatique utilisé sur Internet pour la mise en forme de fichiers et de pages HTML. On le traduit en français par feuilles de style en cascade.

le CSS se présente comme une alternative à la mise en forme via des balises, notamment HTML. Un peu plus complexe à maîtriser, il permet un gain de temps considérable dans la mise en forme d'une page web par rapport à ces balises. Grâce au CSS, vous pouvez en effet appliquer des règles de mise en forme (titrage, alignement, polices, couleurs, bordures, etc.) à plusieurs documents simultanément.

Sur le plan de la conception d'une page Web, le CSS permet par ailleurs de séparer la présentation d'une page HTML et sa structure.

L'HTML et le CSS sont complémentaires.



Figure 2. 12 : CSS IDE

4-JAVASCRIPT :

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript sont au cœur des langages utilisés par les développeurs web. On le retrouve principalement dans les pages Internet. Il permet, entre autres, d'introduire sur une page web ou HTML des petites animations ou des effets. Le langage JavaScript est principalement employé pour améliorer l'ergonomie d'un site Internet et/ou d'une interface applicative utilisateur. Il sert également à intégrer des effets esthétiques, toutefois rarement indispensables. Son principal intérêt réside dans son mode de fonctionnement : le langage JavaScript offre en effet la possibilité d'exécuter un code sans être contraint de recharger une page web. En cela, il joue un rôle dans l'amélioration de la rapidité de chargement des pages, un critère d'ergonomie et de référencement de plus en plus importante.

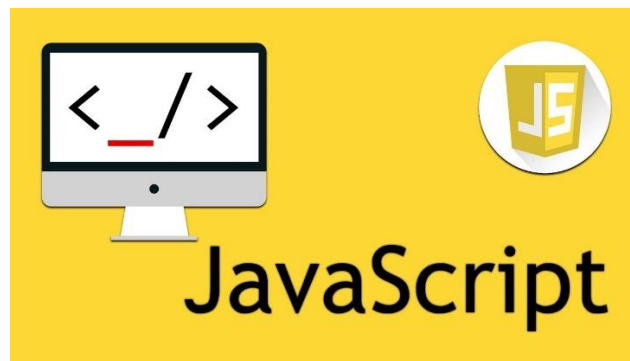


Figure 2. 13 : JAVASCRIPT IDE

4-REACT.JS:

React est une bibliothèque JavaScript utilisée pour construire des composants d'interface utilisateur réutilisables.

React JS est une technologie développée pour faciliter la création d'interface d'une application, elle permet notamment de créer des sites internet sous forme de Single Page Application (des sites internet qui sont constitués d'une seule page), en créant un DOM virtuel.

***Les caractéristiques de react sont :**

JSX :

JSX est une extension de la syntaxe JavaScript. Il n'est pas nécessaire d'utiliser JSX dans le développement React, mais cela est recommandé.

Notion des Composants :

React repose sur les composants. Vous devez penser à tout comme à un composant. Cela vous aidera à maintenir le code lorsque vous travaillez sur des projets à plus grande échelle.

Flux de données unidirectionnel et Flux :

React implémente un flux de données unidirectionnel qui facilite le raisonnement de votre application. Flux est un pattern qui permet de garder vos données unidirectionnelles.

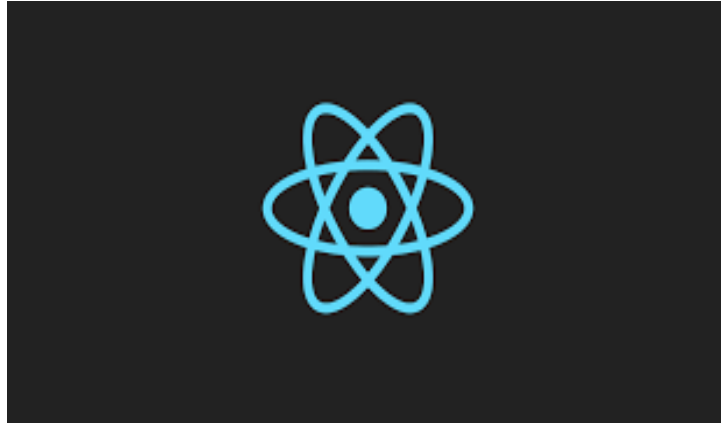


Figure 2. 14 : REACT JS

5-JSX :

JSX est une extension syntaxique de JavaScript. Nous recommandons de l'utiliser avec React afin de décrire à quoi devrait ressembler l'interface utilisateur (UI). ce langage est plus lisible pour les développeurs car il permet de convertir le HTML en Javascript.

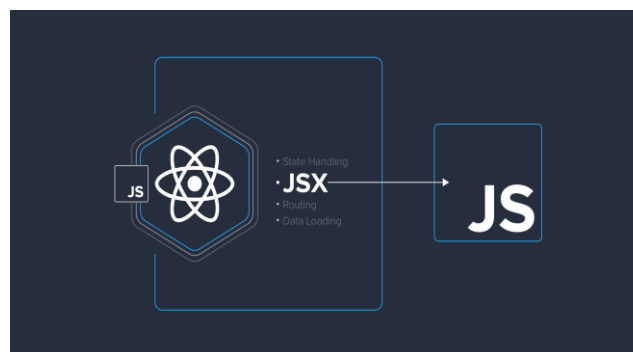


Figure 2. 15 : JSX

5-API DE GOOGLE MAPS :

On a ainsi utilisé l'API de google map pour afficher la carte de monde dans notre site web.



Figure 2. 16 : Google maps

VI-Conclusion :

Dans ce chapitre, nous avons présenté l'étude de la partie matérielle et logicielle (hardware et software) pour la conception d'une carte électronique de surveillance de la pollution de l'air.

Nous avons discuté des spécifications techniques requises pour les capteurs de gaz et les modules GPS, ainsi que des outils nécessaires pour la collecte et la transmission des données, tels que l'Arduino et Firebase.

Nous avons également examiné l'interface utilisateur web pour visualiser les données collectées, qui sera développée à l'aide de technologies telles que REACT, et JavaScript.

Dans le prochain chapitre, nous allons passer à la réalisation concrète du projet, en détaillant les étapes de conception, de développement et de mise en œuvre de la carte électronique de surveillance de la pollution de l'air. Nous allons discuter des défis rencontrés et des solutions apportées, ainsi que des tests et des résultats obtenus.

Chapitre 3 :

Réalisation du projet

Chapitre 3 :

Réalisation du projet

I-Introduction :

Le dernier chapitre de ce rapport de projet décrit en détail la réalisation de la carte électronique de surveillance de la pollution de l'air. Nous présentons les différentes étapes de conception, de développement et de mise en œuvre du projet, en incluant les défis rencontrés et les solutions apportées. Nous expliquons également comment Firebase a été utilisé pour stocker les données collectées par la carte électronique et comment un site web a été créé pour afficher les données de manière claire et compréhensible, avec des fonctionnalités telles que la visualisation en temps réel des données de localisation GPS et de capteur de gaz.

II-Phase de conception électronique :

La conception électronique à travers ISIS Proteus a permis de créer un schématique complet du système de surveillance de la pollution de l'air, qui a été converti en une version 3D pour une meilleure visualisation. Ce schématique a été utilisé pour la réalisation de la carte électronique PCB, qui est la pièce centrale de ce projet.

Le schéma suivant donne le schématique réalisé :

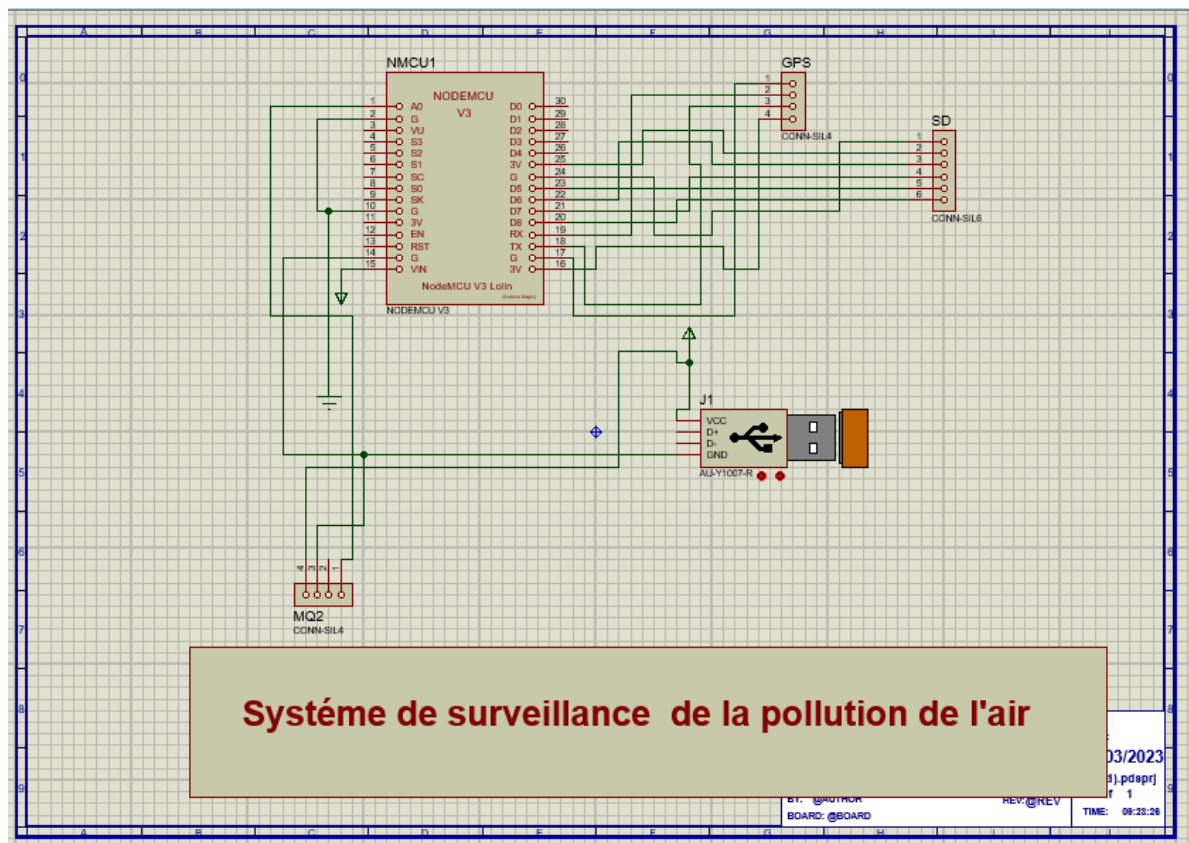


Figure 3. 1 : conception de la carte

Le montage PCB a été réalisé avec soin en suivant les spécifications techniques du cahier des charges, afin d'assurer un fonctionnement optimal du système :

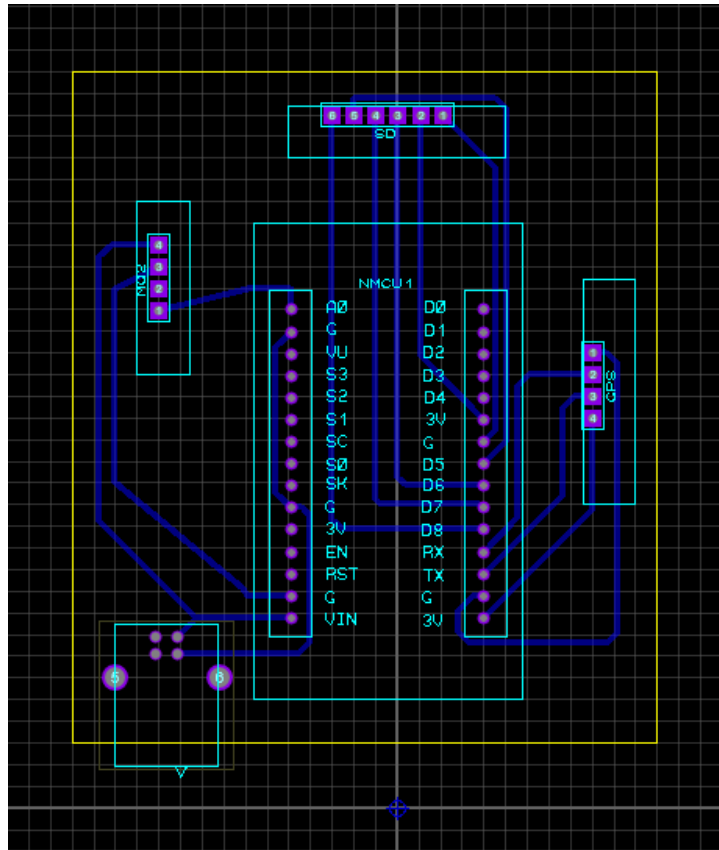


Figure 3. 2 : Pcb de la carte

Ainsi ceci est le montage 3D :

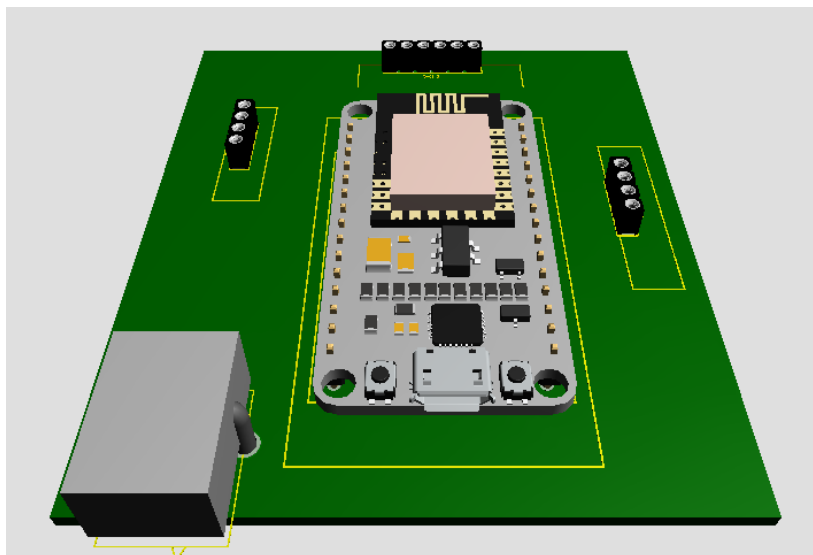


Figure 3. 3 : Montage 3D

Finalement la figure ci-dessous illustre la réalisation réelle de la carte électronique :

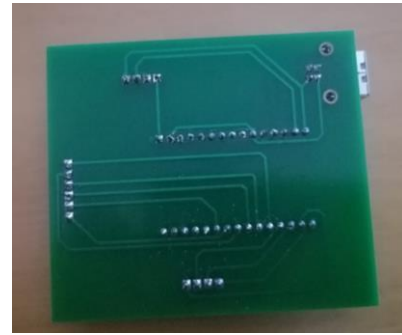
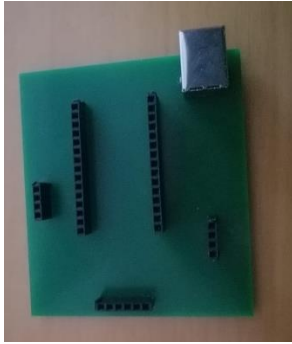


Figure 3. 4 : réalisation de la carte

III-Phase de codage :

1-Programmation via Arduino :

Pour mener à bien ce projet, nous avons d'abord préparé le code Arduino qui permet de stocker les données dans la carte mémoire, en utilisant le module correspondant. Cette étape a été réalisée en premier lieu pour tester le système, en considérant la carte mémoire comme une base de données provisoire. Voilà la figure du montage sur la plaque à essai avant de réaliser la carte PCB finale :

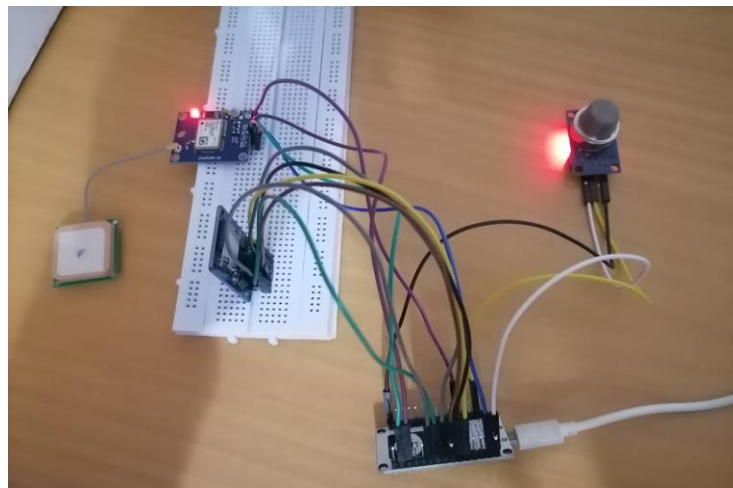


Figure 3. 5 : test sous plaque à essai

Le code permettant de stocker les données dans la carte mémoire est illustré par la figure de l'annexe 1 (**Code 1 dans l'annexe**).

2-stockage et visualisation de données via firebase :

Ensuite, nous avons programmé le NodeMCU pour que la carte envoie les données à Firebase en temps réel si une connexion Wi-Fi est disponible. Nous avons également utilisé la carte

mémoire comme solution de secours dans le cas où il n'y a pas de connexion internet. Dans ce cas, la carte mémoire est utilisée comme base de données.

Le code 2 correspondant est présenté dans l'annexe.

Enfin, la réalisation physique de la carte électronique avec tous ses composants est présentée ci-dessous dans la figure correspondante.

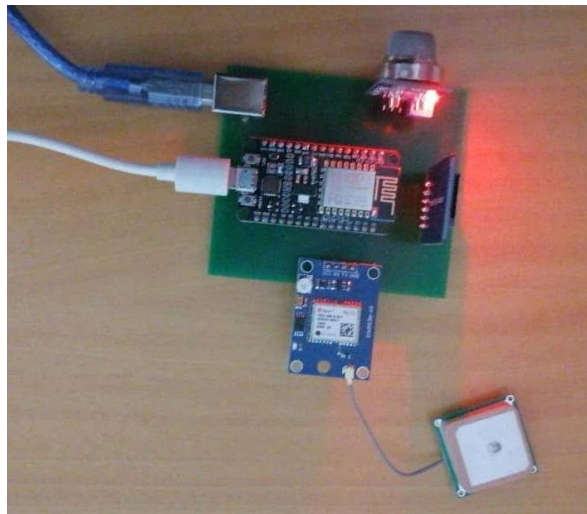


Figure 3. 6 : test finale sur la carte

VI-Conclusion :

En somme, ce chapitre a détaillé la réalisation pratique du projet de surveillance de la pollution de l'air, en présentant les étapes de conception, de développement et de mise en œuvre de la carte électronique. Les difficultés ont été surmontées grâce à des solutions efficaces, aboutissant à un produit fonctionnel respectant les spécifications du cahier des charges. Les tests ont confirmé l'efficacité du système pour collecter et visualiser en temps réel les données de pollution de l'air, stockées et consultables via Firebase

Conclusion générale

En conclusion, ce rapport de projet semestriel a présenté la conception et la réalisation d'un système de surveillance de la pollution de l'air, avec l'utilisation de différentes technologies et outils tels que la carte électronique Arduino, les capteurs de gaz, le module GPS, ISIS Proteus et Firebase. Les différents chapitres de ce rapport ont détaillé les différentes étapes de la conception, de la mise en œuvre et des tests du système, en mettant en évidence les défis rencontrés et les solutions trouvées.

Ce projet a permis de mettre en pratique les connaissances acquises dans le cadre du programme de formation en ingénierie, en particulier dans le domaine de l'électronique et de la programmation. Il a également permis de développer des compétences importantes telles que la gestion de projet, la résolution de problèmes, et la collaboration en équipe.

Le système de surveillance de la pollution de l'air développé dans le cadre de ce projet est un outil précieux pour aider à surveiller la qualité de l'air dans les zones urbaines. Il peut être utilisé pour collecter des données en temps réel et aider à prendre des décisions en matière de santé publique.

Enfin, ce projet a ouvert la voie à de nombreuses possibilités d'amélioration et d'expansion futures, telles que l'ajout de nouveaux capteurs, l'amélioration de la précision de la localisation GPS, l'ajout d'alertes en temps réel pour les utilisateurs, etc. Ce projet est donc un point de départ solide pour une future recherche et développement dans le domaine de la surveillance de la pollution de l'air.

Annexes

Code 1

```
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <SD.h>
#include <SPI.h>
#include <MQ2.h>

#define GPS_TX D1 // Pin TX du module GPS Neo 6M connectée à la pin D1 du NodeMCU ESP8266
#define GPS_RX D2 // Pin RX du module GPS Neo 6M connectée à la pin D2 du NodeMCU ESP8266
#define CS_PIN D8 // Pin CS de la carte mémoire MicroSD connectée à la pin D8 du NodeMCU ESP8266
int pinAout = A0; // pin A0 of NodeMcu is connected to pin A0 of MQ-2
float LPG, Co, Smoke; // Variable values of lpg, CO-gas, and Smoke
SoftwareSerial gpsSerial(GPS_TX, GPS_RX);
TinyGPSPlus gps;
File dataFile;

MQ2 mq2(pinAout);
void setup() {
  Serial.begin(9600);
  mq2.begin();
  pinMode(pinAout, INPUT); //Declare pin A0 as input
  gpsSerial.begin(9600);
  pinMode(CS_PIN, OUTPUT);
  if (!SD.begin(CS_PIN)) {
    Serial.println("Erreur carte mémoire MicroSD");
    return;
  }
  dataFile = SD.open("data.txt", FILE_WRITE);
  if (!dataFile || dataFile.isDirectory()) {
    Serial.println("Erreur ouverture fichier");
    return;
  }
  dataFile.println(" Heure      Date      Latitude  Longitude  LPG    CO    Smoke ");
}
void loop () {
  while (gpsSerial.available() > 0) {
    if (gps.encode(gpsSerial.read())) {
      if (gps.location.isValid()) {
        float lat = gps.location.lat();
        float lon = gps.location.lng();
        String date = String(gps.date.day()) + "/" + String(gps.date.month()) + "/" + String(gps.date.year());
        String heure = String(gps.time.hour()) + ":" + String(gps.time.minute()) + ":" + String(gps.time.second());
        float* values= mq2.read(true); //if set to "false" will not appear on serial monitor
        LPG = mq2.readLPG();
        Co = mq2.readCO();
        Smoke = mq2.readSmoke();
        dataFile.println(heure + "|" + date + "|" + String(lat, 6) + "|" + String(lon, 6) + "|" + String(LPG) + " |" + String(Co) + "|" + String(Smoke));
        dataFile.flush(); // Ajout de l'instruction flush pour forcer l'écriture des données
        Serial.println("Données enregistrées !");
        break; // Sortie de la boucle while car les données ont été enregistrées
      }
    }
  }
}
```


Code 2

```
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <MQ2.h>

// WiFi credentials
const char* WIFI_SSID = "ooredoo_C22830";
const char* WIFI_PASSWORD = "33NCCRHP93RE";

// Firebase credentials
#define FIREBASE_HOST "projet1-5fa0c-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "lrrxvq3A3YzQfNuDpzWQy8u7J0AejwJM94XCdsVP"

#define GPS_TX D1 // Pin TX du module GPS Neo 6M connectée à la pin D1 du NodeMCU ESP8266
#define GPS_RX D2 // Pin RX du module GPS Neo 6M connectée à la pin D2 du NodeMCU ESP8266

// Initialisez les pins de communication GPS
SoftwareSerial gpsSerial(GPS_TX, GPS_RX);

// Initialisez l'objet TinyGPS++
TinyGPSPlus gps;
FirebaseData firebaseData;
int pinAout = A0; // pin A0 of NodeMcu is connected to pin A0 of MQ-2
float LPG, Co, Smoke; // Variable values of lpg, CO-gas, and Smoke

MQ2 mq2(pinAout);

MQ2 mq2(pinAout);
void setup() {
  Serial.begin(9600);
  mq2.begin();
  pinMode(pinAout, INPUT); //Declare pin A0 as input
  gpsSerial.begin(9600);

  // Connexion au réseau WiFi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  Serial.print("Connecting to WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.print("Connected to WiFi ");
  Serial.println(WIFI_SSID);

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  Serial.println("Connexion à Firebase établie !");
}

void loop() {
  // Lire les données GPS
  while (gpsSerial.available() > 0) {
```

```

void loop() {
  // Lire les données GPS
  while (gpsSerial.available() > 0) {
    if (gps.encode(gpsSerial.read())) { if (gps.location.isValid()) {
      float lat = gps.location.lat();
      float lon = gps.location.lng();
      String date = String(gps.date.day()) + "/" + String(gps.date.month()) + "/" + String(gps.date.year());
      String heure = String(gps.time.hour()) + ":" + String(gps.time.minute()) + ":" + String(gps.time.second());
      float* values= mq2.read(true); //if set to "false" will not appear on serial monitor
      LPG = mq2.readLPG();
      Co = mq2.readCO();
      Smoke = mq2.readSmoke();

      // Envoyer les données GPS à Firebase

      Firebase.setString(firebaseData, "latutude ",lat);
      Firebase.setString(firebaseData, "longetutide ",lon);
      Firebase.setString(firebaseData, " date ",date );
      Firebase.setString(firebaseData, "heure ",heure);
      Firebase.setFloat(firebaseData,"LPG", LPG);
      Firebase.setFloat(firebaseData,"CO", Co);
      Firebase.setFloat(firebaseData,"Smoke", Smoke);
    } else {
      Serial.println("Error sending data to Firebase!");
    }
  }

  // Attente de 5 secondes
  delay(5000);
}
}

```

