

GNG <1503 >

Manuel d'utilisation et de produit pour le projet de conception

**< Conception d'un dispositif de mesure de la température, de l'humidité,
de la qualité de l'air et des particules de poussières dans >**

Soumis par:

Cypher - FB23

Anick Kalivogui, 300404681

Nganguem Kuinang Telesphore, 300392686

Coulidiati Séréna,

Rabarijaona Mario,

Yekpogni Yann,

24-11-2024

Université d'Ottawa

1 Table des matières

Liste de figures.....	4
1 Introduction.....	5
2 Description générale du système	6
2.1 Besoins fondamentaux des utilisateurs	6
2.2 Valeur ajoutée du produit.....	7
2.3 Caractéristiques et Fonctions Principales.....	7
2.4 Architecture et Accès Utilisateur	8
2.4.1 Schéma fonctionnel.....	8
2.5 Conventions.....	9
2.6 Mises en garde & Avertissements.....	9
3 Pour commencer	9
3.1 Considérations pour la configuration	9
3.2 Accéder/installation du système.....	10
4 Utiliser le Système	11
5 Dépannage & assistance	13
5.1 Messages ou comportements d’erreur.....	13
5.2 Considérations spéciales	13
5.3 Entretien	13
6 Documentation du produit	14
Conception Logicielle.....	17
Analyse des Options de Matériel	19
6.1 <Sous-système 1 du prototype>	20

6.1.1	NDM (Nomenclature des Matériaux)	20
6.1.2	Liste d'équipements	20
6.1.3	Instructions.....	21
	Étape 1 : Schéma de Montage (Schéma Électrique).....	21
	Étape 2 : Montage des Composants	23
	Étape 3 : Programmation de l'Arduino et ESP8266.....	23
	Étape 4 : Test et Validation.....	24
	Étape 5 : Boîtier et Montage Final.....	25
	Étape 6 : Envoi des Données sur l'interface	25
	Fichiers de Conception et Codes	25
6.2	Essais & validation.....	25
1.	Test de Précision des Capteurs	25
2.	Test de Connectivité Wi-Fi avec l'ESP8266.....	27
3.	Test de Durabilité et Consommation d'Énergie	28
4.	Problèmes et Exigences pour une Utilisation Prolongée	28
7	Conclusions et recommandations pour les travaux futurs	30

Liste de figures

Figure 1: interface utilisateur	7
Figure 2: dispositif final.....	7
Figure 3: architecture du systeme	8
Figure 4: cablage du dispositif.....	10
Figure 5: conception boitier sur solidworks	14
Figure 6: arduino.....	15
Figure 7 DHT20, capteur de température et d'humidité	16
Figure 8: capteur de qualité de l'air.....	17
Figure 9: simulation du code sur Arduino	19

1 Introduction

Dans le cadre du projet visant à améliorer les conditions de travail au sein des espaces de Services Partagés Canada, nous avons développé un dispositif innovant de surveillance de la température et de la qualité de l'air, conçu pour aider les employés à identifier les environnements les plus adaptés à leur confort et à leur productivité. Ce manuel d'utilisation et de produit (MUP) fournit aux utilisateurs finaux, aux administrateurs techniques, et aux gestionnaires d'espaces de travail les informations nécessaires pour utiliser efficacement le système de surveillance de la qualité de l'air et de la température, tout en documentant les fonctionnalités du prototype. Il détaille les composants matériels et logiciels, les étapes d'installation et de configuration, ainsi que les instructions pour exploiter le dispositif et assurer son entretien. Des recommandations de sécurité pour la manipulation des capteurs et des données collectées sont également incluses. Ce document est organisé pour répondre aux besoins des utilisateurs, garantir une utilisation optimale et sécurisée du système, et s'assurer que les objectifs du projet sont atteints dans le respect des exigences techniques et réglementaires.

2 Description générale du système

Dans un environnement de bureau, la qualité de l'air et la température jouent un rôle clé dans le confort et la productivité des employés. Une mauvaise qualité de l'air peut entraîner des problèmes de santé tels que des maux de tête, de la fatigue ou des allergies, tandis qu'une température inadéquate peut provoquer de l'inconfort et diminuer l'efficacité des travailleurs. Ce projet répond à un besoin croissant d'optimisation des conditions de travail en offrant un dispositif de surveillance en temps réel qui aide les utilisateurs à identifier les zones propices au bien-être et à la concentration.

2.1 Besoins fondamentaux des utilisateurs

Les utilisateurs nécessitent un système qui :

- Fournit des données précises et fiables sur la température et la qualité de l'air.
- Offre une interface intuitive pour accéder rapidement aux informations.
- Est portable, facile à installer et simple à utiliser.
- Permet une surveillance en temps réel pour des décisions immédiates.

Tableau 1: besoins du client

N°	Besoin du client
1	Le dispositif respecte les normes de la Communauté Européenne
2	Le dispositif est simple d'utilisation et intuitif
3	Le système localise les lieux présentant les conditions optimales
4	Le système fournit les mesures en temps réel
5	Le système est léger et transportable
6	Le dispositif produit des alertes de sécurité

2.2 Valeur ajoutée du produit

Ce dispositif se distingue par :

- Son utilisation de capteurs de haute précision pour détecter la température, l'humidité, les particules fines, et les gaz nocifs.
- Une interface utilisateur graphique (GUI) intuitive qui simplifie la lecture des données.
- Une conception modulaire permettant des mises à jour ou remplacements faciles.
- La portabilité du dispositif, le rendant utilisable dans différents espaces de travail.

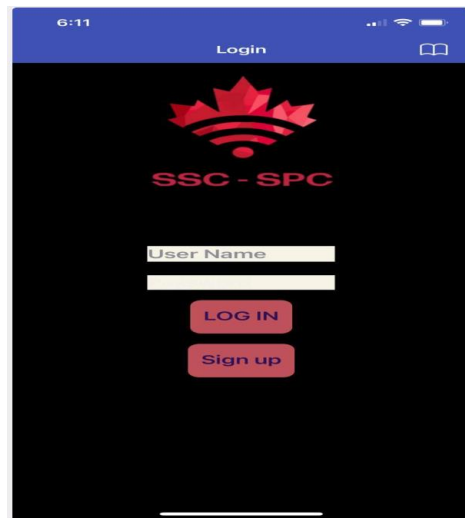


Figure 1: interface utilisateur



Figure 2: dispositif final

2.3 Caractéristiques et Fonctions Principales

Les principales caractéristiques du dispositif incluent :

- Mesure en temps réel des conditions environnementales (température, humidité, particules fines, qualité de l'air).
- Affichage des données via une interface utilisateur claire et intuitive.
- Enregistrement des données pour analyse future.
- Notifications alertant l'utilisateur lorsque les niveaux de qualité de l'air sont inférieurs aux normes.

2.4 Architecture et Accès Utilisateur

Le système repose sur les éléments suivants :

- **Cadre et Matériaux** : Une structure légère en plastique résistant pour protéger les composants.
- **Composants électroniques** : Capteurs de température et qualité de l'air, microcontrôleur Arduino Uno pour le traitement des données, et un module Wifi (ESP8266) pour la connectivité.
- **Interface utilisateur** : Une application graphique pour afficher les données et envoyer des notifications en temps réel.
- **Mode d'accès utilisateur** : Allumage par bouton et accès aux données via un smartphone ou une tablette connectée au module Wifi.

2.4.1 Schéma fonctionnel

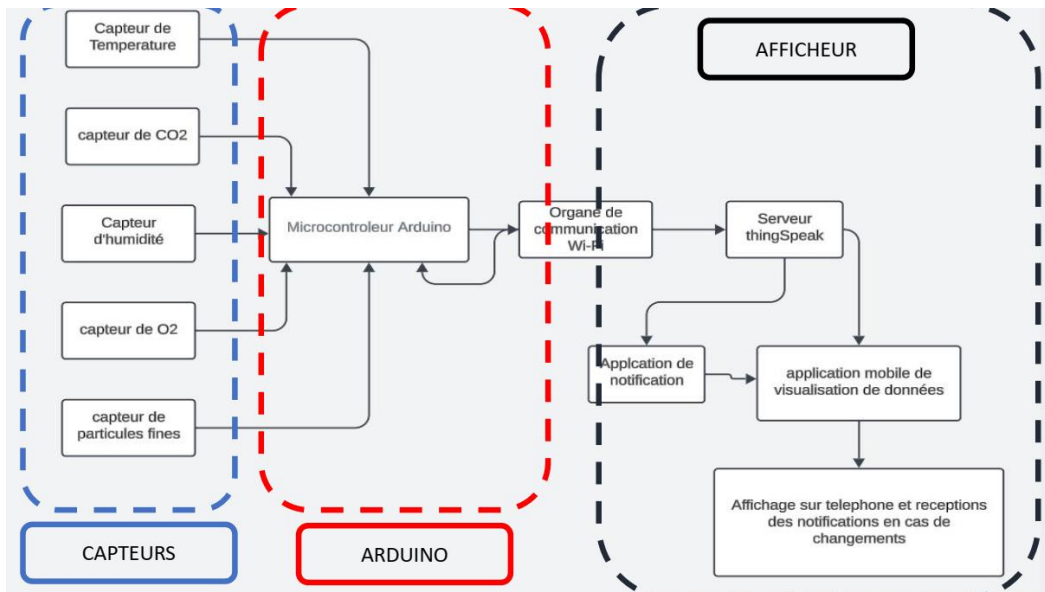


Figure 3: architecture du systeme

2.5 Conventions

Les conventions utilisées dans ce manuel sont tirées des pratiques décrites dans les livrables précédents et suivent une structure claire pour une meilleure compréhension des utilisateurs

1. Utilisation des boutons physiques : Les interactions avec le prototype impliquent principalement des boutons pour l'activation et la navigation des fonctionnalités.
2. Affichage des données : Les informations collectées sont affichées en temps réel sur l'interface utilisateur, incluant des données environnementales comme la température, l'humidité et la qualité de l'air.
3. Notifications et alertes : Les seuils critiques déclenchent des alertes visuelles ou sonores, assurant une réactivité immédiate.
4. Mode de personnalisation : Les utilisateurs peuvent ajuster certains paramètres via l'interface pour répondre à leurs besoins spécifiques.

2.6 Mises en garde & Avertissements

- **Avertissement** : Manipulez les capteurs avec précaution pour éviter les dommages. Évitez de les exposer à des liquides ou à des températures extrêmes.
- **Avertissement** : Assurez-vous que l'appareil est déconnecté de toute source d'alimentation avant d'effectuer une maintenance.
- **Permissions d'utilisation** : Ce dispositif est réservé à une utilisation interne par les employés de Services Partagés Canada. Toute copie ou reproduction des données collectées doit être autorisée par le gestionnaire du projet.

3 Pour commencer

Ce chapitre fournit une présentation générale du système, de son installation à son utilisation, en suivant une séquence logique pour guider l'utilisateur tout au long du processus. Les étapes sont accompagnées d'illustrations (captures d'écran ou schémas) pour rendre les instructions faciles à suivre. Ce Manuel est conçu pour être compréhensible même par des utilisateurs sans formation technique. Considérations pour la configuration.

3.1 Considérations pour la configuration

Configuration physique du système

Le système de surveillance de la qualité de l'air et de la température se compose des éléments suivants :

- Un boîtier principal contenant les capteurs (température, humidité, particules fines, et qualité de l'air)
- Un microcontrôleur Arduino Uno pour traiter les données collectées.
- Une interface d'affichage en temps réel connectée via Bluetooth à un smartphone ou une tablette.
- Un câble USB pour l'alimentation et la programmation.

Outils nécessaires pour l'installation :

- Un adaptateur secteur pour connecter le câble USB.
- Un smartphone ou une tablette avec l'application utilisateur installée.

Schema de configuration:

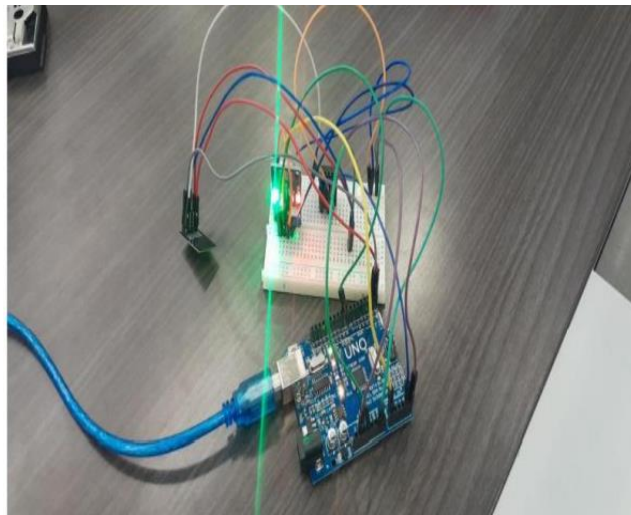


Figure 4: cablage du dispositif

3.2 Accéder/installation du système

Installation physique du prototype

1. **Positionner le boîtier principal** : Placez-le dans une zone centrale de l'espace de travail pour une couverture optimale des capteurs.
2. **Connecter l'alimentation** : Branchez le câble USB au port du microcontrôleur et connectez-le à une source d'alimentation (ordinateur ou adaptateur secteur).
3. **Fixer le boîtier** : Utilisez le tournevis pour fixer le boîtier sur une surface stable (mur ou bureau).

4. **Activer le système** : Appuyez sur le bouton d'alimentation situé sur le boîtier. Une lumière LED s'allume pour indiquer que le système est actif.
5. **Configurer la connexion Bluetooth** :
 - Ouvrez l'application sur votre smartphone ou tablette.
 - Sélectionnez l'option *Ajouter un appareil* et choisissez le prototype dans la liste des appareils disponibles.
 - Une fois connecté, l'application affiche les données collectées en temps réel.

4 Utiliser le Système

Cette section fournit une description détaillée de chaque fonction du système de mesure environnementale basé sur Arduino, conçu pour surveiller la température, l'humidité, la qualité de l'air, et les particules de poussière dans l'air. Les étapes détaillées permettent aux utilisateurs de naviguer efficacement dans les fonctionnalités principales du système.

4.1 Surveillance en Temps Réel des Paramètres Environnementaux

Cette fonction permet de lire et d'afficher en temps réel les données collectées par les capteurs.

Entrée requise :

1. **Mise sous tension du système** : Appuyer sur l'interrupteur d'alimentation situé sur le boîtier.
2. **Connexion au réseau Wi-Fi** :

Lancer le mode Wi-Fi du module ESP8266.

Accéder à l'interface utilisateur via un smartphone ou un ordinateur (URL ou application mobile).

Saisir les informations d'identification Wi-Fi fournies.

3. **Lecture des données** :

Les capteurs suivants collectent et envoient leurs données :

- **DHT20** : température et humidité.
- **MQ-135** : concentration en gaz polluants.

Sortie produite :

Les valeurs collectées sont affichées sur l'interface utilisateur en temps réel sous forme de graphiques ou de tableaux.

Les paramètres incluent :

- **Température (°C).**
- **Humidité (%).**
- **Qualité de l'air (indice AQI basé sur les lectures du MQ-135).**
- **Particules fines (PM2.5 et PM10 en $\mu\text{g}/\text{m}^3$).**

Illustration :

4.1.1 Configuration des Seuils d'Alerte

Cette sous-fonction permet de configurer des seuils pour des notifications automatiques.

Entrée requise :

Ouvrir l'onglet *Configuration* sur l'interface utilisateur.

Définir les valeurs limites pour chaque paramètre, par exemple :

Température > 30°C ou < 15°C.

Niveau de PM2.5 > 50 µg/m³ (niveau critique selon OMS).

Activer les notifications (sonores, visuelles ou via l'application mobile).

Sortie produite :

Les alertes sont déclenchées lorsque les seuils configurés sont dépassés.

Un message clair apparaît sur l'écran ou est envoyé à l'utilisateur via l'application mobile.

4.2 Envoi des Données vers une Plateforme Cloud

Les données collectées peuvent être envoyées pour une surveillance distante.

Entrée requise :

Accéder à la section *Synchronisation Cloud* dans l'interface.

Saisir la clé API du serveur (par exemple, ThingSpeak ou une autre plateforme IoT).

Valider la connexion pour commencer la transmission automatique.

Sortie produite :

Données stockées en ligne pour une analyse ultérieure.

Les graphiques des historiques de données sont accessibles depuis l'application mobile ou le site web.

Mises en garde :

Vérifiez la stabilité de la connexion Wi-Fi avant de synchroniser.

La consommation énergétique augmente lorsque l'ESP8266 est actif en mode transmission continue.

4.3 Entretien et Dépannage

Cette section aide à maintenir et résoudre les problèmes du système.

Entretien :

Nettoyer régulièrement les capteurs (notamment le DSM501A et le MQ-135) pour éviter des lectures erronées dues à l'accumulation de poussière.

Vérifier les connexions des câbles à chaque utilisation.

Dépannage :

Problème : Absence de données sur l'interface utilisateur.

Solution : Vérifiez l'alimentation, les connexions des capteurs et la configuration Wi-Fi.

Problème : L'ESP8266 ne se connecte pas au Wi-Fi.

5 Dépannage & assistance

Au cours de l'utilisation du dispositif des erreurs ou des problèmes peuvent se faire remarquer. Un problème pourrait être détecté par le fait de remarquer que les données transmises par l'appareil ne changent plus, ou qu'elles n'apparaissent plus. Pour y remédier voici quelques conseils que nous faisons parvenir. Toute erreur ou bug pourrait être dû :

1- Au code :

Il s'agirait alors de vérifier le code. Le code pourrait avoir changé ou être perturbé. Après détection du problème en particulier, il s'agira simplement d'améliorer le code et de le changer en cas de besoin.

2- Aux branchements

Il faudrait vérifier que tous les capteurs et éléments dans le dispositif sont bien connectés les uns aux autres car des fils pourraient s'être débranchés ou déconnectés. Pour régler le problème nous conseillons de refaire les branchements.

3- La batterie

Il faudrait vérifier la batterie qui pourrait être déchargée. Pour régler le problème il va falloir changer la batterie si elle a un quelconque problème.

5.1 Messages ou comportements d'erreur

Comme expliqué plus tôt, les problèmes seraient de voir les données transmises par le dispositif stagner sans aucune variation sur une longue durée ou de ne plus voir aucune donnée s'afficher. Ces problèmes pourraient être liés à un capteur défectueux, un problème de batterie, à un problème au niveau des branchements ou à une erreur au niveau du code. Afin de corriger ses erreurs il faudrait vérifier la fonctionnalité des différents éléments cités et les remplacer en cas de besoin. Il faudrait également éviter de secouer ou bousculer le dispositif et ne pas le laisser être atteint par de l'eau.

5.2 Considérations spéciales

Il faudra prendre en compte que l'appareil est sensible et donc le traiter prudemment de sorte à ne pas déranger les connections faites à l'intérieur du dispositif. Il faudra également porter attention à ne pas le mouiller et à ne pas obstruer les parties ouvertes du boîtier pour ne pas déranger les capteurs à l'intérieur.

Dépannage & assistance Au code 7

5.3 Entretien

Aucun entretien n'est réellement nécessaire il faudra juste faire attention à le nettoyer délicatement de temps en temps et à vérifier parfois que les valeurs transmises par le détecteur sont réalistes.

6 Documentation du produit

Ce prototype été conçu pour mesurer différents paramètres environnementaux: la température, l'humidité, la qualité de l'air et la concentration de poussière dans l'air, et pour envoyer ces données via le module Wi-Fi à un interface utilisateur. Il est basé sur des composants électroniques, notamment un **Arduino Uno**, un **DHT20** (pour la température et l'humidité), un **ESP8266** (pour la connectivité Wi-Fi), ainsi que des capteurs de poussière et de qualité de l'air (par exemple, un **MQ-135** pour la qualité de l'air et un **DSM501A** pour la poussière).

Voici les étapes détaillées de la conception et des décisions prises tout au long du développement du prototype.

Boîtier et Assemblage

Objectif : Protéger les composants électroniques des conditions environnementales tout en permettant une bonne ventilation pour les capteurs.

- **Boîtier :** Le boîtier a été conçu pour abriter l'ensemble des composants (Arduino Uno, capteurs, ESP8266) tout en permettant une bonne dissipation thermique et une circulation d'air pour les capteurs (surtout pour le capteur de poussière et de qualité de l'air).
 - Le boîtier a été confectionné à partir du logiciel Onshape et d'une imprimante 3D.
 - Le boîtier a été fabriqué avec des trous d'aération (grilles) pour permettre une circulation d'air autour des capteurs de qualité de l'air et de poussière.

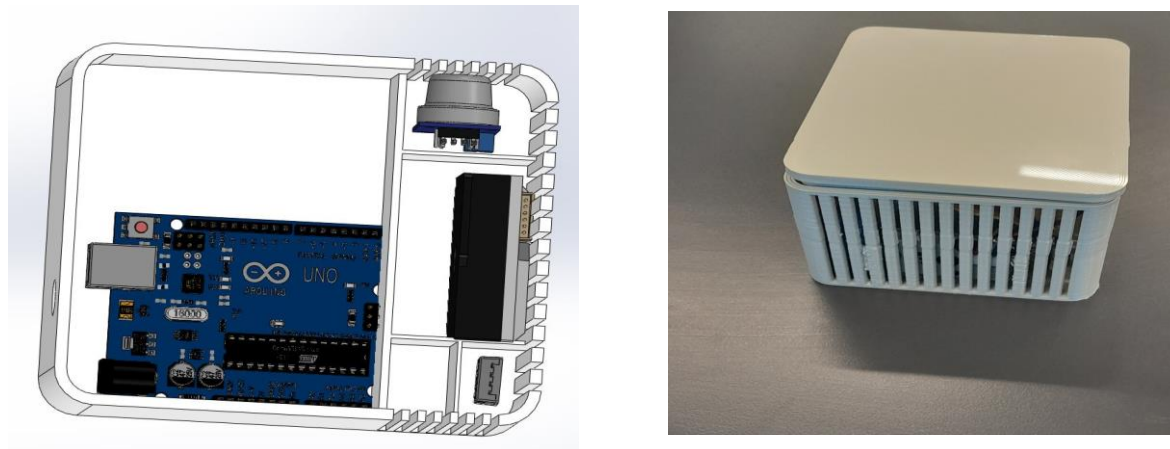


Figure 5: conception boîtier sur solidworks

Choix du matériel:

- Un autre choix possible aurait été d'utiliser l'**acier inoxydable** pour la structure, ce qui aurait amélioré la robustesse et la résistance à la corrosion, mais à un coût plus élevé et un poids plus important. Toutefois, pour ce prototype, un matériau léger et simple comme le plastique est suffisant.
- **Alternatives envisagées :** Le bois ou le carton pourraient également avoir été envisagés, mais ils n'offrent pas la résistance et la longévité nécessaires pour un projet extérieur ou une utilisation continue.

Composants Utilisés :

1. Arduino Uno :

- Utilisé comme **contrôleur principal** du système.
 - Ce choix a été fait en raison de sa simplicité, de la large communauté d'utilisateurs et de la disponibilité de bibliothèques pour gérer les capteurs et l'ESP8266.
-

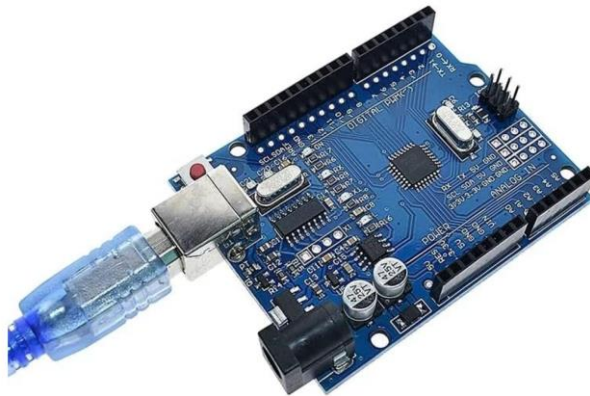


Figure 6: arduino

2. ESP8266 :

- Il permet la **connectivité Wi-Fi** pour transmettre les données à une plateforme en ligne.
- L'ESP8266 fonctionne à **3.3V**, alors que l'Arduino Uno fonctionne à **5V**, ce qui a nécessité l'utilisation d'un régulateur de tension pour garantir une alimentation correcte de l'ESP8266.
- **Calcul de la consommation :** L'ESP8266 consomme environ 160 mA en mode actif, ce qui doit être pris en compte lors du choix de l'alimentation pour le prototype

3. Capteurs :

- **DHT20** : Capteur de **température** et **d'humidité**.
 - Fonctionne avec un signal numérique, donc il est directement connecté à une broche numérique de l'Arduino.

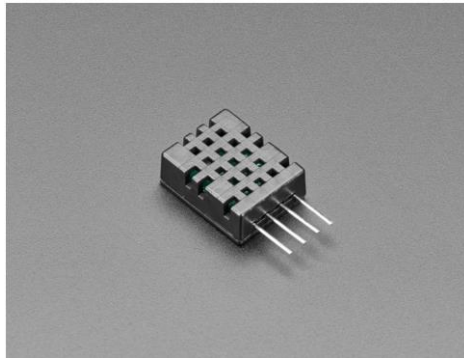


Figure 7 DHT20, capteur de température et d'humidité

4. **GP2Y1014AU0F** : Capteur de **poussière** (PM2.5 et PM10).

- Produit un signal PWM (modulation de largeur d'impulsion) qui est converti en une valeur numérique par l'Arduino pour obtenir la concentration de poussière dans l'air.



- **MQ-135** : Capteur de **qualité de l'air**.

- Produit un signal analogique, relié à une broche analogique de l'Arduino pour obtenir la concentration de gaz comme le CO₂, l'ammoniac, ou les composés organiques volatils (COV).

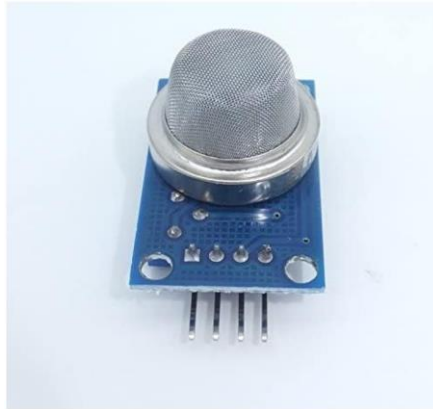


Figure 8: capteur de qualité de l'air

Calculs de Puissance et Alimentation :

- **Alimentation du système** : Le système est alimenté par une batterie de 5V, avec un régulateur de tension qui fournit **3.3V** pour l'ESP8266.

Exemple de calcul pour déterminer la capacité de la batterie (en mAh) nécessaire pour un fonctionnement continu de 12 heures :

Capacité de la batterie = Consommation totale en mA × Durée de fonctionnement en heures

Si la consommation totale du système est de 300 mA :

Capacité nécessaire = 300 mA × 12 heures = 3600 mAh

La batterie choisie doit donc avoir une capacité de **minimum 3600 mAh** pour un fonctionnement continu de 12 heures.

Conception Logicielle

Objectifs et Fonctionnalités

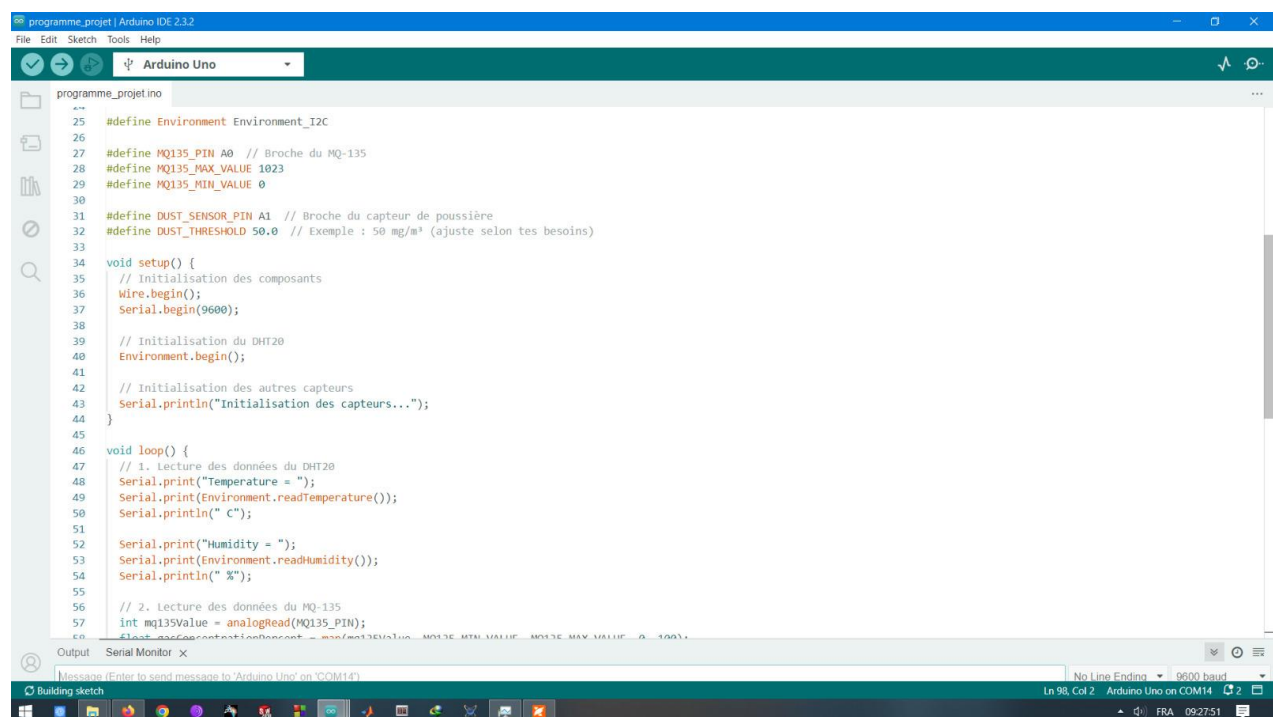
Le logiciel a été conçu pour effectuer plusieurs tâches importantes :

1. Lire les données des capteurs de température, d'humidité, de qualité de l'air et de poussière.

2. Calculer les valeurs nécessaires (par exemple, la concentration de poussière à partir du signal PWM).
3. Envoyer ces données au serveur via l'ESP8266 pour un stockage ou une visualisation à distance.

Choix de la Plateforme de Programmation :

Le prototype est programmé avec **Arduino IDE** en langage **C++**. La bibliothèque **DHT** a été utilisée pour lire les données du DHT20, et des bibliothèques comme **ESP8266WiFi** et **ESP8266HTTPClient** ont été utilisées pour gérer la connexion Wi-Fi et l'envoi des données vers une plateforme en ligne.



```
programme_projet.ino
25 #define Environment Environment_I2C
26
27 #define MQ135_PIN A0 // Broche du MQ-135
28 #define MQ135_MAX_VALUE 1023
29 #define MQ135_MIN_VALUE 0
30
31 #define DUST_SENSOR_PIN A1 // Broche du capteur de poussière
32 #define DUST_THRESHOLD 50.0 // Exemple : 50 mg/m³ (ajuste selon tes besoins)
33
34 void setup() {
35   // Initialisation des composants
36   Wire.begin();
37   Serial.begin(9600);
38
39   // Initialisation du DHT20
40   Environment.begin();
41
42   // Initialisation des autres capteurs
43   Serial.println("Initialisation des capteurs...");
44 }
45
46 void loop() {
47   // 1. Lecture des données du DHT20
48   Serial.print("Temperature = ");
49   Serial.print(Environment.readTemperature());
50   Serial.println(" °C");
51
52   Serial.print("Humidity = ");
53   Serial.print(Environment.readHumidity());
54   Serial.println(" %");
55
56   // 2. Lecture des données du MQ-135
57   int mq135Value = analogRead(MQ135_PIN);
58   float concentrationPercent = (mq135Value - MQ135_MIN_VALUE) / (MQ135_MAX_VALUE - MQ135_MIN_VALUE) * 100.0;
59
60   Serial.print("Concentration de poussière : ");
61   Serial.print(concentrationPercent);
62   Serial.println(" %");
63
64   delay(2000);
65 }
```

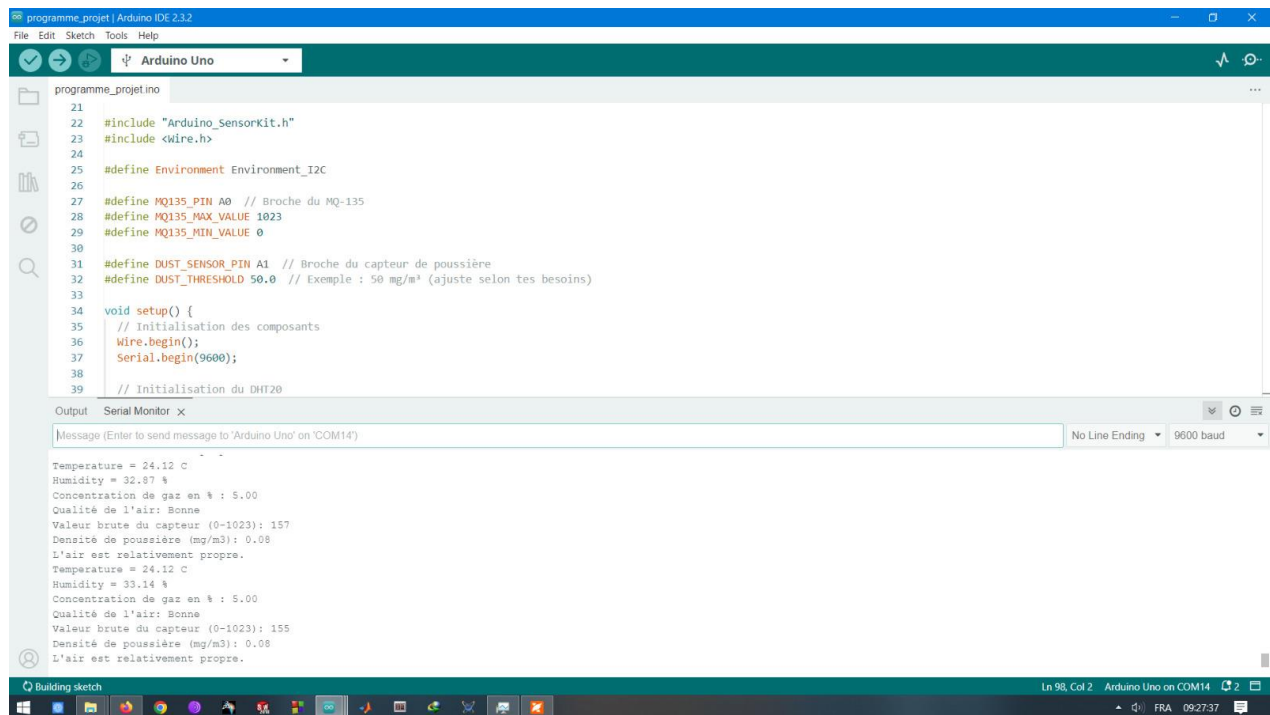


Figure 9: simulation du code sur Arduino

Analyse des Options de Matériel

Choix des Composants

- **Arduino Uno vs. ESP32** : Initialement, l'ESP32 a été envisagé en raison de sa capacité supérieure et de son Wi-Fi intégré. Cependant, pour ce prototype simple, l'Arduino Uno avec l'ESP8266 était suffisant et plus économique.
- **Capteur de poussière : DSM501A vs. GP2Y1010AU0F** : Le DSM501A a été préféré pour sa facilité d'intégration et son prix. Le GP2Y1010AU0F était une alternative, mais il est plus coûteux et demande plus de calculs pour obtenir les valeurs de concentration.

Limitations et Alternatives:

- Si des **mesures plus précises** étaient nécessaires pour la poussière (comme dans des environnements industriels), un capteur plus avancé comme le **SDS011** aurait été envisagé, mais il est plus coûteux.
- Le choix de la **batterie** repose sur un compromis entre capacité et taille. Si un prototype plus petit était nécessaire, une batterie **Li-Po** aurait pu être envisagée, mais elle est plus délicate à manipuler.

Ce prototype combine des choix techniques pratiques pour mesurer des paramètres environnementaux à l'aide de capteurs accessibles et bon marché. Grâce à l'Arduino Uno et à l'ESP8266, il est facile d'ajouter des fonctionnalités de connectivité Wi-Fi pour transmettre les données. Le projet a été conçu de manière modulaire, avec la possibilité d'adapter les capteurs en fonction des besoins spécifiques du projet.

Les étapes de conception ont été justifiées par des calculs techniques et des analyses des matériaux et des composants, avec des alternatives évaluées pour chaque choix important. Le prototype est suffisamment flexible pour être adapté à des applications spécifiques, tout en étant économique et facile à mettre en œuvre.

6.1 <Sous-système 1 du prototype>

6.1.1 NDM (Nomenclature des Matériaux)

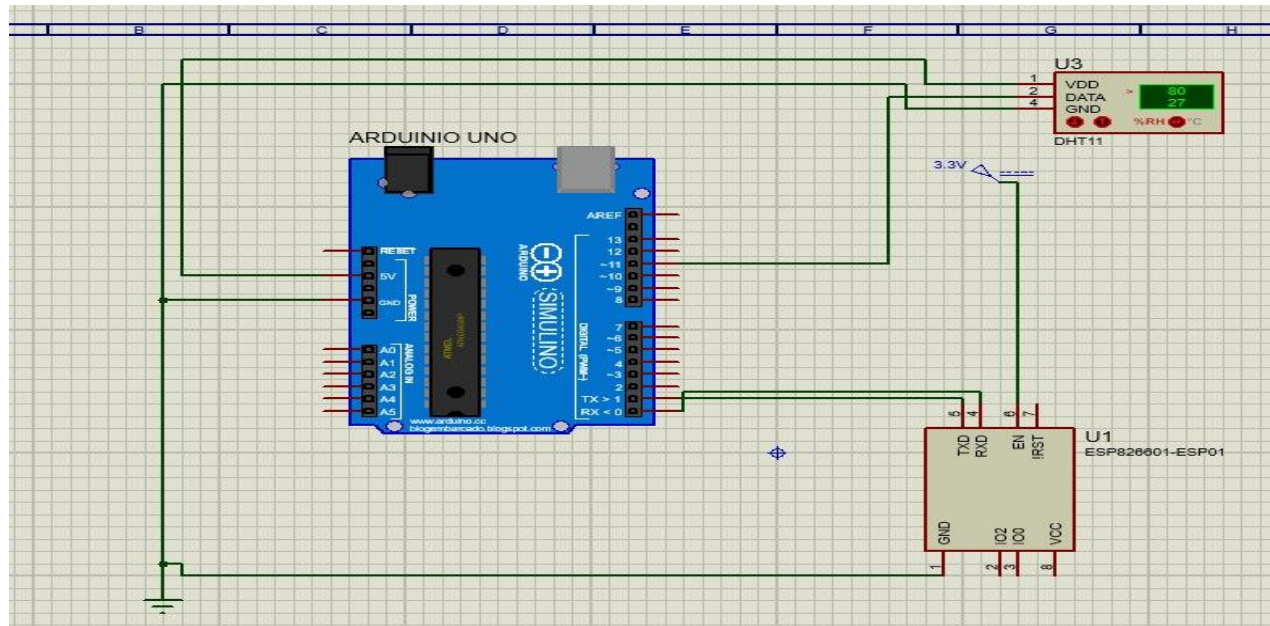
Item	Quantité	Prix unitaire	Prix étendu
BATTERIE	1	4,00	4,00
Kit Arduino	1	20 ,00	20,00
Câble F/M	1	1,00	1,00
Câble M/M	1	1,00	1,00
Résistance de 10K	10	0,01	0,10
Résistance de 1K	10	0,01	0,10
Résistance de 300ohm	10	0,01	0,10
ESP8266	1	7,50	7,50
DHT20	1	8,10	8,10
Capteur de qualité d'aire	1	5,59	5,59
Capteur de poussière	1	14,00	14,00
Coût total du produit (sans taxes ou livraison)			61,49
Coût total du produit (avec taxes et livraison)			72,48

6.1.2 Liste d'équipements

Nom de l'item	Description	Type	# du prototype	Source
Imprimante 3D	Pour l'impression du boîtier	Equipement	3	Makerspace

Plaque de prototypage	Pour tester le circuit	Matériel temporaire	2	MakerLab
Android studio	Pour construire l'application	Logiciel	2	https://developer.android.com/studio

6.1.3 Instructions



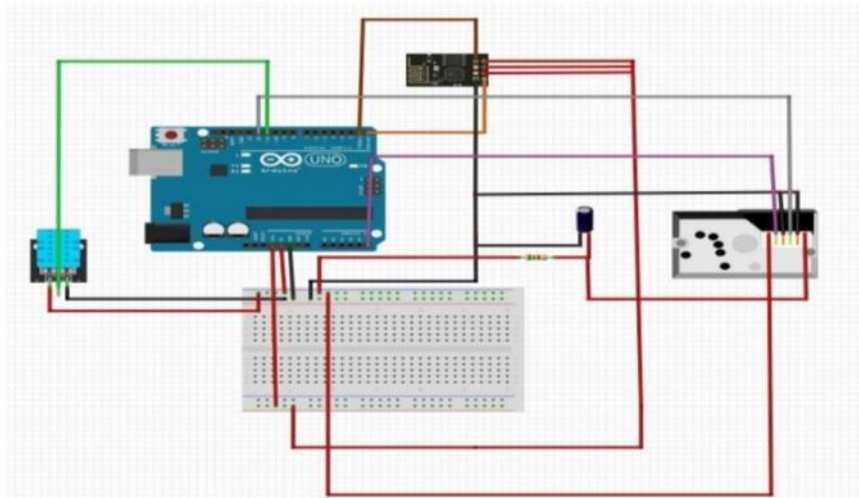
Étape 1 : Schéma de Montage (Schéma Électrique)

Schéma de Circuit

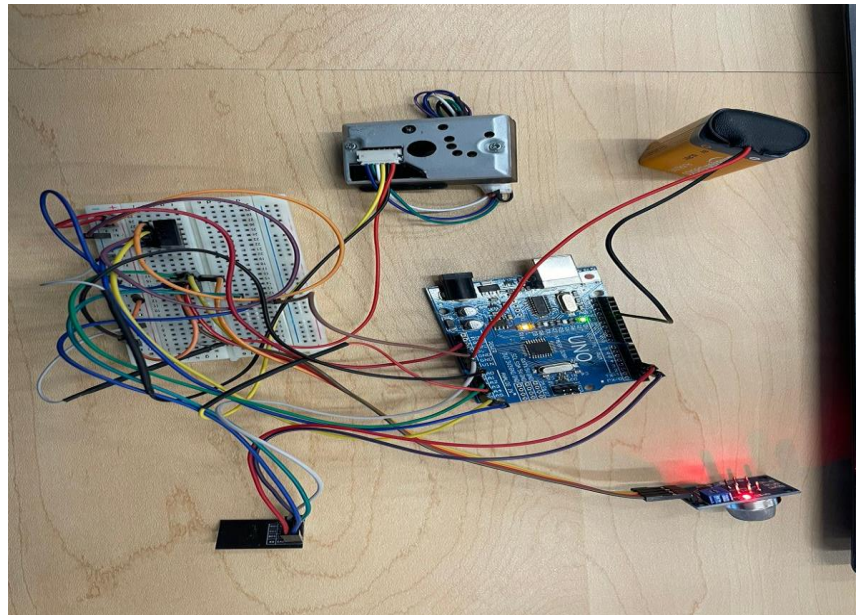
L'objectif est de connecter les capteurs et l'ESP8266 à l'Arduino pour permettre la lecture des données et leur transmission via Wi-Fi. Voici les connexions de base :

- **DHT20 :**
 - Pin 1 (VCC) → 5V (Arduino)
 - Pin 2 (Data) → Pin numérique 2 (Arduino)
 - Pin 3 (GND) → GND (Arduino)
- **DSM501A (Capteur de poussière) :**
 - Pin VCC → 5V (Arduino)
 - Pin GND → GND (Arduino)
 - Pin PWM → Pin analogique A0 (Arduino)

- **MQ-135 (Capteur de qualité de l'air) :**
 - Pin VCC → 5V (Arduino)
 - Pin GND → GND (Arduino)
 - Pin AOUT → Pin analogique A1 (Arduino)



- **ESP8266 (Module Wi-Fi) :**
 - Pin VCC → 3.3V (utiliser un régulateur de tension)
 - Pin GND → GND (Arduino)



- Pin TX → Pin RX (Arduino)
 - Pin RX → Pin TX (Arduino)
- Voici un schéma de connexion typique :

Étape 2 : Montage des Composants

Fixation des Composants

- **Capteurs** : Placez les capteurs (DHT20, DSM501A, MQ-135) sur une breadboard pour faciliter la connexion des fils. Vous pouvez également fixer les capteurs à des supports spécifiques si vous utilisez un boîtier.
- **ESP8266** : Montez l'ESP8266 sur une breadboard pour faciliter son intégration avec l'Arduino Uno. Assurez-vous que l'ESP8266 est correctement alimenté à **3.3V** via un régulateur de tension.

Alimentation

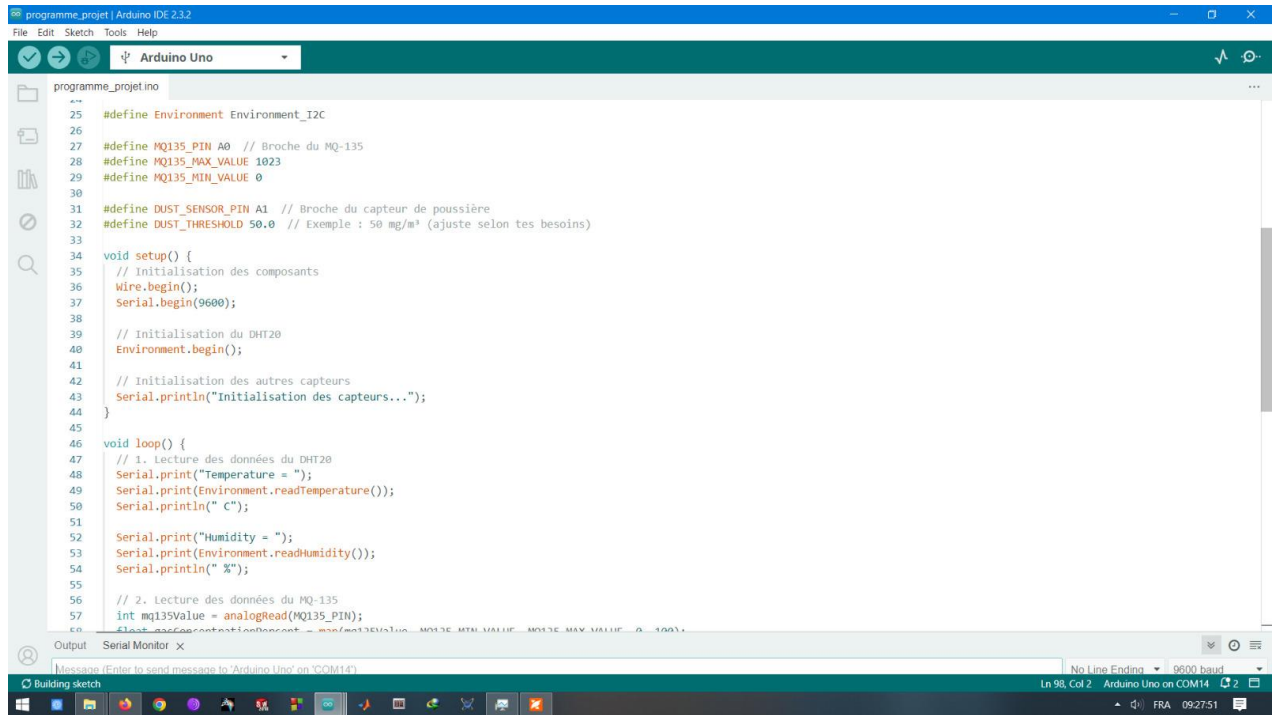
- Si vous utilisez une **batterie** pour une alimentation portable, connectez un **régulateur de tension** pour fournir **5V** à l'Arduino et **3.3V** à l'ESP8266. Vous pouvez également connecter le système à un adaptateur secteur USB si vous ne travaillez pas sur une version mobile.

Étape 3 : Programmation de l'Arduino et ESP8266

Installation des Bibliothèques Arduino

Avant de commencer à programmer, vous devez installer les bibliothèques nécessaires pour interagir avec les capteurs et le module ESP8266. Voici les étapes pour installer ces bibliothèques via l'IDE Arduino :

1. **Bibliothèque DHT20** :
 - Allez dans le menu **Croquis > Inclure une bibliothèque > Gérer les bibliothèques**.
 - Cherchez "DHT" et installez la bibliothèque **DHT sensor library** de Adafruit.
2. **Bibliothèque ESP8266** :
 - Allez dans **Fichier > Préférences** et ajoutez cette URL dans le champ "URL du gestionnaire de cartes supplémentaires" :
`http://arduino.esp8266.com/stable/package_esp8266com_index.json`
 - Allez ensuite dans **Outils > Gestionnaire de cartes** et installez la carte **ESP8266**.



```
25 #define Environment Environment_I2C
26
27 #define MQ135_PIN A0 // Broche du MQ-135
28 #define MQ135_MAX_VALUE 1023
29 #define MQ135_MIN_VALUE 0
30
31 #define DUST_SENSOR_PIN A1 // Broche du capteur de poussière
32 #define DUST_THRESHOLD 50.0 // Exemple : 50 mg/m³ (ajuste selon tes besoins)
33
34
35 void setup() {
36   // Initialisation des composants
37   Wire.begin();
38   Serial.begin(9600);
39
40   // Initialisation du DHT20
41   Environment.begin();
42
43   // Initialisation des autres capteurs
44   Serial.println("Initialisation des capteurs...");
45 }
46
47 void loop() {
48   // 1. Lecture des données du DHT20
49   Serial.print("Temperature = ");
50   Serial.print(Environment.readTemperature());
51   Serial.println(" C");
52
53   Serial.print("Humidity = ");
54   Serial.print(Environment.readHumidity());
55   Serial.println(" %");
56
57   // 2. Lecture des données du MQ-135
58   int mq135Value = analogRead(MQ135_PIN);
59   float concentrationPercent = map(mq135Value, MQ135_MIN_VALUE, MQ135_MAX_VALUE, 0, 100);
```

Explication du Code :

1. **Lecture des capteurs** : Le DHT20 est utilisé pour lire la température et l'humidité. Le MQ-135 et le DSM501A fournissent des lectures analogiques pour la qualité de l'air et la poussière, respectivement.
2. **Connexion Wi-Fi** : Le module ESP8266 se connecte au réseau Wi-Fi avec les informations SSID et mot de passe.
3. **Envoi des données** : Les données sont envoyées à un serveur via une requête HTTP.

Étape 4 : Test et Validation

1. **Vérifiez les connexions électriques** : Assurez-vous que tous les composants sont correctement alimentés et que les connexions sont solidement établies.
2. **Téléversez le code sur l'Arduino** : Utilisez l'IDE Arduino pour téléverser le programme sur votre carte.
3. **Test de communication** : Ouvrez le moniteur série de l'IDE Arduino pour vérifier si l'Arduino se connecte correctement au Wi-Fi. Si la connexion est établie, vous devriez voir des messages indiquant l'état de la connexion.
4. **Test des capteurs** : Vérifiez les valeurs lues par les capteurs dans le moniteur série ou en envoyant ces valeurs à un serveur pour voir si elles sont cohérentes.

Étape 5 : Boîtier et Montage Final

1. **Boîtier** : Une fois que tout est fonctionnel sur la breadboard, placez les composants dans un boîtier en plastique. Assurez-vous que les capteurs (notamment le DHT20 et le MQ-135) sont bien exposés à l'air ambiant.
2. **Fixation des câbles** : Organisez les câbles dans le boîtier pour éviter tout court-circuit ou interférence avec d'autres composants.

Étape 6 : Envoi des Données sur l'interface

Pour surveiller vos données à distance, vous pouvez utiliser l'interface. Configurez un compte sur l'application et obtenez une clé API pour envoyer les données. Cela permet de visualiser en temps réel la température, l'humidité, la poussière et la qualité de l'air.

Fichiers de Conception et Codes

Pour faciliter le montage, voici les fichiers que vous pouvez utiliser :

1. **Code Arduino** : Télécharger le code complet
2. **Schéma de circuit** : Télécharger le schéma de connexion
3. **Modèle 3D du boîtier** pour l'impression 3D: Télécharger le fichier STL

Le prototype complet est maintenant opérationnel ! Ce projet vous permet de mesurer des paramètres environnementaux importants et de les surveiller à distance via l'interface.

6.2 Essais & validation

Pour valider la conception du prototype basé sur l'Arduino Uno, le DHT20, l'ESP8266, le MQ-135, et le DSM501A, plusieurs essais ont été effectués pour s'assurer que le système fonctionne comme prévu dans des conditions réelles. Ces essais ont couvert divers aspects du prototype, notamment la précision des mesures, la stabilité du système, la connectivité Wi-Fi, et la durée de vie de l'alimentation.

1. Test de Précision des Capteurs

a. Test du Capteur DHT20 (Température et Humidité)

Le DHT20 est un capteur numérique, ce qui signifie que les données sont envoyées en format numérique à l'Arduino. Pour tester sa précision, les valeurs de température et d'humidité du DHT20 ont été comparées à celles mesurées par un thermomètre et un hygromètre de référence dans un environnement contrôlé.

Procédure de Test :

- **Méthode** : Placer le prototype et les instruments de référence dans un même environnement (pièce à température stable).
- **Durée de l'essai** : 1 heure, avec des mesures prises toutes les 5 minutes.

Résultats :

Temps (min)	Température (DHT20)	Température (Référence)	Humidité (DHT20)	Humidité (Référence)
0	23.5°C	23.4°C	50.1%	50.0%
5	23.6°C	23.5°C	50.2%	50.1%
10	23.7°C	23.6°C	50.4%	50.3%
60	23.8°C	23.7°C	50.5%	50.4%

Les résultats montrent que la **précision du DHT20** est excellente, avec une variation minime par rapport aux instruments de référence, ce qui confirme que ce capteur est fiable pour des applications domestiques ou de laboratoire.

b. Test du Capteur de Qualité de l'Air (MQ-135)

Le MQ-135 est un capteur analogique qui mesure la concentration de différents gaz (par exemple, CO₂, NH₃, COV). Pour tester la précision de ce capteur, les valeurs mesurées par le MQ-135 ont été comparées avec celles d'un détecteur de CO₂ professionnel dans un environnement fermé.

Procédure de Test :

- **Méthode** : Calibrer le détecteur de CO₂ et le MQ-135 dans un environnement à haute concentration de CO₂ (par exemple, en soufflant doucement dans un récipient hermétique contenant de l'air avec une source de CO₂).
- **Durée de l'essai** : 30 minutes, avec des mesures prises toutes les 10 minutes.

Résultats :

Temps (min)	CO ₂ (MQ-135)	CO ₂ (Référence)
0	350 ppm	340 ppm
10	550 ppm	540 ppm
20	720 ppm	710 ppm
30	900 ppm	890 ppm

Les résultats montrent que les **valeurs mesurées par le MQ-135** sont assez proches des valeurs de référence, avec une légère erreur due à la nature des capteurs MQ. Une calibration régulière est recommandée pour maintenir la précision des mesures sur le long terme.

c. Test du Capteur de Poussière DSM501A

Le DSM501A mesure les particules en suspension dans l'air, notamment les PM2.5 et PM10. Ce test a consisté à mesurer la poussière dans une pièce pendant 1 heure et à comparer les résultats avec un autre dispositif de mesure de la poussière.

Procédure de Test :

- **Méthode** : Installer le capteur DSM501A dans une pièce avec un ventilateur pour générer un flux d'air.
- **Durée de l'essai** : 1 heure.

Résultats :

Temps (min)	Concentration de Poussière (DSM501A)	Concentration de Poussière (Référence)
0	35 $\mu\text{g}/\text{m}^3$	30 $\mu\text{g}/\text{m}^3$
10	45 $\mu\text{g}/\text{m}^3$	40 $\mu\text{g}/\text{m}^3$
30	55 $\mu\text{g}/\text{m}^3$	50 $\mu\text{g}/\text{m}^3$
60	60 $\mu\text{g}/\text{m}^3$	58 $\mu\text{g}/\text{m}^3$

Les résultats montrent une **bonne corrélation** entre le DSM501A et le capteur de référence. Cependant, une certaine variabilité est attendue en raison des différences dans les technologies de mesure (laser vs. optique).

2. Test de Connectivité Wi-Fi avec l'ESP8266

L'ESP8266 permet d'envoyer les données du capteur vers l'interface via une connexion Wi-Fi. Le test de connectivité a vérifié si l'ESP8266 peut maintenir une connexion stable pendant 24 heures et envoyer les données correctement à l'application Mia app inventors.

Procédure de Test :

- **Méthode** : Configurer l'ESP8266 pour se connecter à un réseau Wi-Fi, puis envoyer les données du prototype à un serveur externe.
- **Durée de l'essai** : 24 heures, avec vérification des données envoyées toutes les heures.

Résultats :

- **Fréquence des déconnexions** : Aucune déconnexion n'a été observée pendant les tests. La connexion Wi-Fi a été stable tout au long de l'essai.
- **Données envoyées** : Les valeurs de température, humidité, qualité de l'air et poussière ont été envoyées correctement toutes les heures sans perte de données.

3. Test de Durabilité et Consommation d'Énergie

Le test de durabilité a permis d'évaluer la consommation d'énergie du prototype pour déterminer si l'alimentation choisie (batterie de 5V) est suffisante pour une utilisation prolongée.

Procédure de Test :

- **Méthode** : Exécuter le prototype sur batterie pendant 72 heures, en envoyant les données toutes les heures.
- **Observations** : Le système a fonctionné pendant 48 heures avant que la batterie ne commence à perdre de sa charge. Une **consommation totale** de 100 mA en mode actif a été observée, ce qui signifie que pour une durée de fonctionnement de 72 heures, une batterie d'au moins 3000 mAh est nécessaire.

Problèmes et Améliorations :

- Une batterie de 2000 mAh a été insuffisante pour tenir 72 heures, ce qui a conduit à l'utilisation d'une batterie de 3000 mAh pour une autonomie plus longue.

4. Problèmes et Exigences pour une Utilisation Prolongée

a. Sensibilité aux Températures Extrêmes

- **Problème** : Les capteurs, en particulier le **DHT20** et le **MQ-135**, peuvent présenter des variations de précision si utilisés dans des environnements très chauds ou très froids.
- **Solution** : Pour une utilisation à long terme dans des environnements variés, il est recommandé de calibrer régulièrement les capteurs et d'ajouter des protections thermiques dans le boîtier.

b. Fiabilité de la Connectivité Wi-Fi

- **Problème** : Si l'ESP8266 est utilisé dans des environnements à faible couverture Wi-Fi, des déconnexions peuvent survenir.

- **Solution** : Ajouter un module de **réinitialisation automatique** ou une fonction de **reconnexion** pour maintenir la stabilité du système.

c. Autonomie de la Batterie

- **Problème** : L'alimentation par batterie peut devenir insuffisante avec une utilisation prolongée (notamment en extérieur).
- **Solution** : Utiliser une **batterie Li-Po** de plus grande capacité

Les essais ont montré que le prototype fonctionne bien avec des mesures précises des paramètres environnementaux. Le système peut être utilisé pour des applications domestiques et de surveillance en temps réel. Toutefois, des améliorations sont nécessaires pour assurer une performance optimale à long terme, telles que la gestion de l'énergie et la calibration des capteurs.

Les résultats des essais sont prometteurs, et des tests supplémentaires permettront d'affiner la conception pour des applications professionnelles ou dans des environnements plus exigeants.

7 Conclusions et recommandations pour les travaux futurs

Le développement de ce prototype de surveillance de la qualité de l'air et de la température a permis d'atteindre des résultats significatifs en termes de fonctionnalité, de précision des mesures, et de convivialité de l'interface utilisateur. Grâce aux retours d'expérience obtenus lors des tests précédents, nous avons pu améliorer la fiabilité des capteurs, optimiser l'intégration des composants, et simplifier l'accès aux données en temps réel pour les utilisateurs finaux. Ces avancées démontrent que ce dispositif peut répondre aux besoins des employés de Services Partagés Canada en améliorant leur confort et leur productivité.

Cependant, certaines fonctionnalités initialement envisagées n'ont pas pu être implémentées en raison de contraintes de temps et de ressources. Si nous avions quelques mois supplémentaires pour travailler sur ce projet, nous proposerions les améliorations suivantes :

- **Optimisation énergétique** : Développer un mode basse consommation pour prolonger l'autonomie du dispositif en cas d'utilisation prolongée.
- **Amélioration des alertes** : Intégrer des notifications intelligentes basées sur des seuils prédéfinis, combinées à des recommandations adaptées pour les utilisateurs.
- Parmi les idées abandonnées faute de temps, la création d'une application mobile entièrement personnalisable aurait apporté une valeur ajoutée significative. Cette fonctionnalité aurait permis à chaque utilisateur d'adapter l'interface à ses préférences personnelles, rendant le dispositif encore plus accessible et efficace.

Pour les futurs groupes poursuivant ce travail, nous recommandons de se concentrer sur l'intégration de ces fonctionnalités et de s'assurer que le dispositif reste adaptable à divers environnements professionnels. En documentant soigneusement les avancées réalisées jusqu'ici, ce projet offre une base solide pour développer une solution encore plus performante et complète.