

GNG1503

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

CLEAR DATA

Soumis par:

FB12

Rania Karami, 300282637

Bamps Bampelenga, 300289279

Ben Bamba, 300303463

Nicolas Nuccitelli, 300291637

16 avril 2023

Université d'Ottawa

Table des matières

1. Introduction	4
2. Aperçu.....	4
2.1 Conventions.....	6
3. Pour commencer.....	6
3.1 Considérations pour la configuration.....	10
3.2 Considérations pour l'accès des utilisateurs	10
3.3 Accéder/installation du système.....	10
3.4 Organisation du système & navigation	10
3.5 Quitter le système	11
4. Utiliser le système.....	11
5. Dépannage & assistance	11
5.1 Messages ou comportements d'erreur.....	12
5.2 Considérations spéciales	12
5.3 Entretien.....	12
5.4 Assistance	12
6. Documentation du produit	12
6.1 Développement de l'application	13
6.1.1 NDM (Nomenclature des Matériaux).....	13
6.1.2 Liste d'équipements	13
6.1.3 Instructions.....	13
6.2 Développement du boîtier et du circuit.....	14
6.2.1 NDM (Nomenclature des Matériaux).....	14
6.2.2 Liste d'équipements	14
6.2.3 Instructions.....	14
6.3 Finalisation de l'application/circuit et du boîtier	15
6.3.1 NDM (Nomenclature des Matériaux).....	15
6.3.2 Liste d'équipements	15
6.3.3 Instructions.....	16
6.4 Essais & validation	16
7. Conclusions et recommandations pour les travaux futurs.....	17
8. APPENDICE I: Fichiers de conception.....	17

Liste de figures

Figure 1 Concept final	5
Figure 2 Circuit du concept final	5
Figure 3 Étape pour connecter l'Arduino à l'ordinateur	6
Figure 4 Étape pour ouvrir un sketch dans Arduino IDE	6
Figure 5 Étape du codage	8
Figure 6 Sketch dans Arduino	9
Figure 7 Application Thingview.....	9
Figure 8 Graphiques sur l'application Thingview	10

Liste de tableaux

Tableau 1. Acronymes	3
Tableau 2 Table. Besoins de l'utilisateur.....	4
Tableau 3. Nomenclature des matériaux du prototype 1	13
Tableau 4 Liste d'équipements du prototype 1.....	13
Tableau 5 Nomenclature des matériaux du prototype 2	14
Tableau 6 Liste d'équipement du prototype 2	14
Tableau 7 Nomenclature des matériaux du prototype 3	15
Tableau 8 Liste d'équipements du prototype 3.....	15

Liste d'acronymes

Tableau 1. Acronymes

Acronyme	Définition
MUP	Manuel d'utilisateurs et de produit
CD	Clear Data
NDM	Nomenclature des Matériaux
3D	Trois dimensions

1. Introduction

Ce manuel d'utilisation et de produit (MUP) fournit les informations nécessaires aux futurs étudiants de l'université d'Ottawa et tous autres personnes pour utiliser efficacement le concept Clear Data (CD) et pour la documentation du prototype. Dans ce manuel d'utilisateur et de produit, on retrouve un concept qui capte la température, l'humidité, les gaz dangereux et la qualité d'air qui est contrôlable par l'application Thingview. Ensuite, l'assistance, la documentation et la conclusion vont être démontré dans le manuel. L'objectif du manuel est de permettre aux prochains utilisateurs, clients, futurs étudiant ou ingénieur de faire des améliorations sur le prototype pour le réparer ou pour le perfectionner. Le public visé est le client, des ingénieurs ou des futurs étudiants. Ce manuel devra être utilisé seulement à des fins de recherches ou académique et non commerciales.

2. Aperçu

Le but est de concevoir un concept pouvant capter la température et l'humidité d'un espace donné dans le cas de "Recherche en infrastructures et innovation" du groupe Activation Numérique, un centre de données.

Le tableau ci-dessous démontre les besoins fondamentaux de l'utilisateur

Tableau 2 Besoins de l'utilisateur

N°	Besoins
1	La taille du boîtier n'est pas plus grande qu'une boîte à soulier, préférablement comme une taille de boîte à mouchoir.
2	Être capable de modifier et déplacer les pièces/capteurs dans le boîtier.
3	Le budget est de 200\$ (un coût abordable).
6	Le produit démontre les variations de température et d'humidité à l'aide d'un graphique
7	L'appareil/L'application est disponible à tout moment.
8	Les données sont accessibles dans l'affichage et l'étendue est configurable
9	Une fonction d'alerte mineure lors d'un petit changement qui ajoute une autre alerte si le changement augmente et il y a plus qu'une personne qui reçoit des alertes.
10	L'utilisateur est capable par lui-même de paramétrer les limites de température à ne pas dépasser et de savoir dans quel zone l'humidité a augmenté ainsi que la température.
11	Le pourcentage d'erreurs est entre 2 à 3 degrés.

Ce qui nous différencie des autres équipes sont nos différentes composantes méticuleusement conçues, le fait que nous priorisons le produit que notre client désire et aussi le fait que la

matrice décisionnelle a été crucial quant au choix du meilleur prototype prouvant ainsi que notre prototype est de bonne qualité.

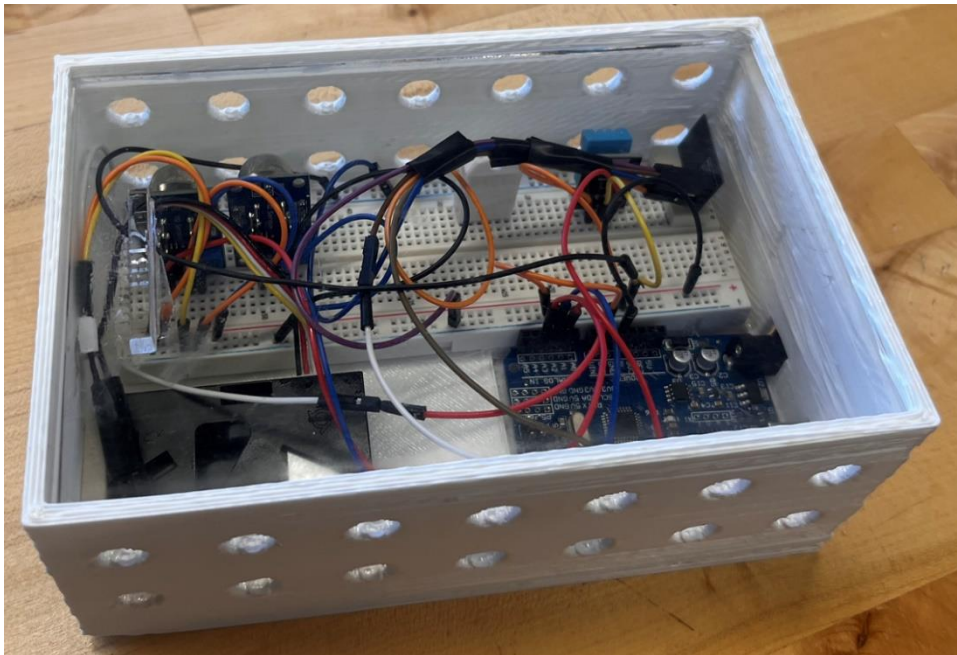


Figure 1 Concept final

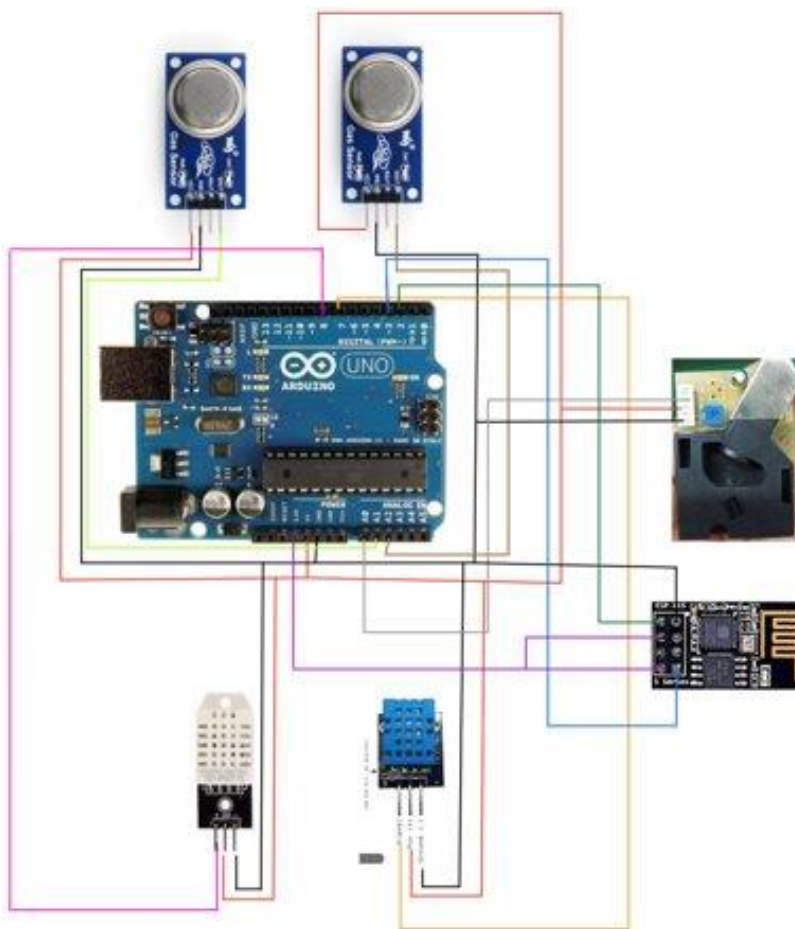


Figure 2 Circuit du concept final

Le produit est capable de détecter la température et l'humidité (capteurs DHT11 et DHT22), les gaz dangereux (capteur MQ-135) et la qualité d'air (capteur DSM501A PM2.5). Il y a un Arduino UNO R3 qui permet de programmer les capteurs et un module Wi-Fi ESP8266 ESP-01S qui permet de connecter les capteurs et l'application ensemble. Nous sommes capables de voir les données sous forme de graphique à l'aide de l'application Thingview sur un appareil.

2.1 Conventions

Aucune convention n'a été utilisée dans ce manuel d'utilisation et de produit.

2.2 Mises en garde & avertissements

- 1) Ne pas utiliser ce produit pour des raisons commerciales, mais seulement l'utiliser pour des raisons académiques ou pour des recherches.
- 2) Pour les trous de ventilations autour de la boîte, il faut éviter de faire des trous trop petits puisque lors de l'impression 3D, ils risquent de se refermer.

3. Pour commencer

1. Connecter le fil avec l'Arduino et un ordinateur à prise USB



Figure 3 Étape pour connecter l'Arduino à l'ordinateur

2. Ouvrir un nouveau sketch Arduino IDE pour le code



Figure 4 Étape pour ouvrir un sketch dans Arduino IDE

3. Taper ce code suivant en téléchargeant les bibliothèques nécessaires

```
#include <SoftwareSerial.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#define RX 2
#define TX 3
#define DHTPIN 8 // Analog Pin sensor is connected to
#define DHTTYPE DHT11
DHT dht = DHT(DHTPIN, DHTTYPE);
String AP = "CBY-Makerspace"; // AP NAME
String PASS = "W2uOttawa!"; // AP PASSWORD
String API = "UVRQGZPBFUYLVCS9"; // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
int valSensor = 1;
int air;
int gaz1;
int gaz2;
int ledPower = 7;
int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;
float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;

SoftwareSerial esp8266(RX, TX);

void setup() {
  Serial.println("initial air value:" + air);
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT", 5, "OK");
  sendCommand("AT+CWMODE=1", 5, "OK");
  sendCommand("AT+CWJAP=\"" + AP + "\",\"" + PASS + "\",20,\"OK\");");
  dht.begin();
  pinMode(ledPower, OUTPUT);
}

pinMode(ledPower, OUTPUT);
}

void loop() {

  String getData = "GET /update?api_key="+ API
  +"&field1="+getTemperatureValue()+"&field2="+getHumidityValue()+"&field3="+getAirQualityValue()+"&field4=
  +caleb();
  sendCommand("AT+CIPMUX=1", 5, "OK");
  sendCommand("AT+CIPSTART=0, \"TCP\", \"" + HOST + "\", " + PORT, 15, "OK");
  sendCommand("AT+CIPSEND=0, " + String(getData.length() + 4), 4, ">");
  esp8266.println(getData); delay(1500); countTrueCommand++;
  sendCommand("AT+CIPCLOSE=0", 5, "OK");
}

String getTemperatureValue(){
  Serial.print("initial air value:"); //remove
  Serial.println(air);
  float t = dht.readTemperature();
  Serial.print("Temperature: ");
  Serial.print(t);
  int temp = dht.readTemperature();
  Serial.println(temp);
  delay(50);
  return String(t);
}

String getHumidityValue(){
  float h = dht.readHumidity();
  Serial.print("Humidity: ");
  Serial.print(h);
  int humidity = dht.readHumidity();
  Serial.println(humidity);
  delay(50);
  return String(humidity);
}
```

```

        return String(humidity);
    }

String getAirQualityValue() {
    Serial.print("Air quality:");
    air= analogRead(A0);
    Serial.println(air);
    int airFinal = (air*100)/1023;
    delay(50);
    return String(airFinal);
}

String caleb() {
    gaz1 = analogRead(A1);
    gaz2 = analogRead(A2);
    int gazfinal= (gaz1+gaz2)/2;
    delay(50);
    Serial.print("Qualité d'air 2: ");
    Serial.print(gazfinal);
    return String(gazfinal);
}

void sendCommand(String command, int maxTime, char readReply[] ) {
    Serial.print(countTrueCommand);
    Serial.print(". at command => ");
    Serial.print(command);
    Serial.print(" ");
    while(countTimeCommand < (maxTime*1))
    {
        esp8266.println(command);//at+cipsend
        if(esp8266.find(readReply)//ok
        {
            found = true;
            break;
        }
    }
    countTimeCommand++;
}

delay(50);
Serial.print("Qualité d'air 2: ");
Serial.print(gazfinal);
return String(gazfinal);
}

void sendCommand(String command, int maxTime, char readReply[] ) {
    Serial.print(countTrueCommand);
    Serial.print(". at command => ");
    Serial.print(command);
    Serial.print(" ");
    while(countTimeCommand < (maxTime*1))
    {
        esp8266.println(command);//at+cipsend
        if(esp8266.find(readReply)//ok
        {
            found = true;
            break;
        }

        countTimeCommand++;
    }

    if(found == true)
    {
        Serial.println("OYI");
        countTrueCommand++;
        countTimeCommand = 0;
    }

    if(found == false)
    {
        Serial.println("Fail");
        countTrueCommand = 0;
        countTimeCommand = 0;
    }

    found = false;
}

```

Figure 5 Étape du codage

4. Téléverser le code dans le Arduino

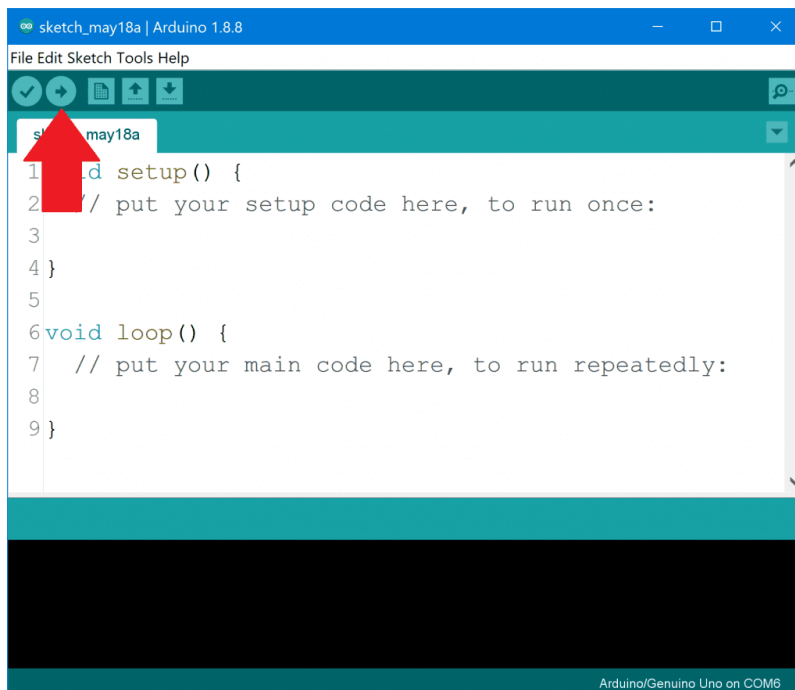


Figure 6 Sketch dans Arduino

5. Ouvrir l'application Thingview et connectez-vous



Figure 7 Application Thingview

6. Une fois connecter, vous serez capable de voir les graphiques



Figure 8 Graphiques sur l'application Thingview

3.1 Considérations pour la configuration

L'utilisateur a besoin d'un ordinateur à prise USB, d'un fil USB qui est compatible à un Arduino. Aussi, l'utilisateur a besoin d'installer l'application Thingview pour pouvoir accéder aux graphiques des données que captent les capteurs, aussi installer Arduino IDE pour ainsi programmer le tout.

3.2 Considérations pour l'accès des utilisateurs

Il n'y a aucune considération pour l'accès des utilisateurs puisque toute est accessible à leur disposition.

3.3 Accéder/installation du système

Pour démarrer l'application Arduino IDE, il faut d'abord commencer par faire l'installation sur le site web suivant : <https://www.arduino.cc/en/software> et choisir la version qui correspond à l'ordinateur portable (Mac ou Windows). Une fois installer, il faut ouvrir l'application et commencer un sketch. Une fois le sketch créer, l'utilisateur peut enfin commencer à taper son code.

3.4 Organisation du système & navigation

S'agissant d'un produit physique en se divisant en trois sous-systèmes notamment :

La boîte :

Nécessaire à recevoir nos composants, celle-ci fut conçue à l'aide de l'imprimante 3D nous évitant ainsi les besoins de l'usage de vis en ayant utilisé la découpe laser ou autres mais aussi une vitre en Acrylique permettant un regard sans soucis de nos composants internes sans forcément avoir recours au déballage autrement dit, sans avoir à ouvrir la boîte

Circuit électrique :

Au centre de ce circuit, nous possédons un Arduino UNO jouant le rôle d'un microcontrôleur et nous avons plusieurs autres composants qui sont :

DSM501Apm2.5 : un capteur de particules qui prends une portion de l'air et l'enferme dans une chambre noire pour ainsi compter les particules en suspension dans l'air, il envoie ainsi un chiffre entre 1 et 1023 qui représente l'intervalle de pollution de l'air. Tout en considérant que 1023, l'air est irrespirable et 1 d'une pureté sans pareil. Celui-ci est connecté à la prise Analog A0

MQ-135 : qui est un capteur de gaz dangereux qui détecte s'il y a dans l'air un des gaz nuisibles à la vie humaine et de la même façon que notre DSM501Apm2.5, celui-ci envoie un chiffre entre 1 et 1023. Nous avons ainsi décidé de les coupler en parallèle avec un deuxième MQ-135 pour avoir la meilleure approximation de la qualité d'air que nous puissions avoir. Ceux-ci sont connectés aux prises analog A0 et A1.

DHT11 et DHT22 qui sont tous les deux des capteurs d'humidité et de température qu'on a également couplés en parallèle. Celui-ci nous donne des valeurs de température précises notamment grâce à Read Temperature ou Read Humidity grâce notamment à leur bibliothèque commune DHT.h. Elles sont connectées à des prises digitales comme le graphique nous le démontre.

ESP8266 qui est un module Wifi permettant d'envoyer les données reçues par notre Arduino à ThingSpeak qui sera ensuite déléguée à l'application équivalente Thingview.

L'application Thingview:

Celle-ci reçoit les données de notre Arduino envoyé grâce aux ESP8266 à ThingSpeak et l'imprime sous forme de graphique.

3.5 Quitter le système

Pour quitter le système, on doit d'abord sauvegarder le code en appuyant sur File et ensuite Save Project dans un dossier assigné. Une fois cela complété vous pouvez fermer l'application et débrancher le fil qui est connecté à l'ordinateur portable et à l'Arduino.

4. Utiliser le système

Pour utiliser ce système, il suffit de l'alimenter électriquement et se munir de l'application Thingview pour ainsi recevoir tous les détails de températures, l'humidité, les gaz dangereux ainsi que la qualité de l'air nécessaire.

5. Dépannage & assistance

Quelques situations peuvent se présenter où l'utilisateur aura besoin du manuel d'utilisateur pour les situations suivantes :

- une panne d'électricité influant sur la source d'alimentation, dans ce cas il est possible de brancher notre dispositif à une batterie externe (Powerbank) afin de combler ce déficit.
- Un surchauffage entraînant un court-circuit, il serait judicieux de nous communiquer l'information afin d'y remédier.

5.1 Messages ou comportements d'erreur

Lors de la mauvaise lecture de température, humidité ou qualité de l'air, le code nous envoie une erreur de la forme "Mauvaise lecture de température ou autres", pour corriger ce genre d'erreur, il faut vérifier les branchements nécessaires de tous les composants.

Pour des fautes de mauvaises connectivités en affichant un message "Connexion : FAIL", il faut tout d'abord vérifier les branchements du capteur ESP8266 et ensuite vérifier l'API Keys dans le code qui correspond à la clé de notre Channel et ensuite refaire la compilation de notre code.

En cas de mauvaise lecture dû à la perte d'un capteur ou d'un autre, l'application thingview cessera d'envoyer des messages continues de ce capteur. Il faudra ainsi revenir dans l'IDE Arduino pour connaître le capteur défectueux.

5.2 Considérations spéciales

Souvent le capteur ESP8266 peut envoyer des messages de Fail dépendamment de la boîte dans laquelle, elle se trouve, il est donc bon de vérifier tout cela. La meilleure façon de faire est de vérifier son circuit en extérieur avant de la mettre dans une boîte et ainsi en cas d'erreur connaître précisément sa cause qui est donc la boîte qui ne permet pas au capteur d'envoyer des données.

5.3 Entretien

De temps en temps, nous pensons, toutes les semaines, il est bon d'empoussiérer les composants internes car ceux-ci peuvent être soumis à beaucoup de poussière qui à la longue pourrait altérer les données de nos capteurs respectif en particulier nos DHT11 et 22 qui possèdent plusieurs trous pouvant notamment recueillir de la poussière sans beaucoup d'entretien.

5.4 Assistance

Pour obtenir de l'assistance, vous pouvez contacter Rania Karami à l'adresse courriel rkara042@uottawa.ca .Pour qu'on puisse vous aider, il faut une capture d'écran du problème et une description de l'erreur. Aucune garantie que l'erreur sera réglée, mais de l'assistance sera offerte.

6. Documentation du produit

Le prototype est divisé en trois sous-systèmes particuliers : le développement de l'application, le développement du boîtier et du circuit.

6.1 Développement de l'application

6.1.1 NDM (Nomenclature des Matériaux)

Tableau 3. Nomenclature des matériaux du prototype 1

Nom de l'Item	Description	Quantité	Coût unitaire	Liens
Capteur MQ-135	Capteur de détection de gaz dangereux	3	10\$	Capteur de gaz dangereux
Capteur DSM501A PM2.5	Capteur de détection de poussières de la qualité d'air	1	17\$	Capteur de poussière et qualité d'air
Arduino UNO R3	Arduino UNO R3 utiliser pour programmer le tout	1	17\$	Arduino
ESP8266 ESP-01S	Module émetteur récepteur série Wifi sans fil compatible avec Arduino IDE	5	25\$	ESP8266 ESP-01S
Application ThingView	Application mobile qui permet de démontrer les données en forme de graphique	1	2.75\$	Application ThingView
Coût total				71.75\$

6.1.2 Liste d'équipements

Tableau 4 Liste d'équipements du prototype 1

Nom de l'item	Description	Type	# du prototypage	Source
Thingview	Pour construire l'application	Logiciel	2 et 3	Application Thingview
Arduino IDE	Pour coder l'Arduino	Logiciel	2 et 3	Arduino IDE

6.1.3 Instructions

Étape 1 : Téléversé le code dans le MakerRepo

Étape 2 : Copier le code du fichier “.txt”

Étape 3 : Coller le code dans le compilateur Arduino

Étape 4 : Envoyer le code au Arduino

6.2 Développement du boîtier et du circuit

6.2.1 NDM (Nomenclature des Matériaux)

Tableau 5 Nomenclature des matériaux du prototype 2

Nom de l'Item	Description	Quantité	Coût unitaire	Liens
ELEGOO UNO super kit	Une boîte contenant un capteur de température et humidité (DHT11 et DHT22) ainsi que des fils, écran LCD, une plaque à souder, lumière et résistance	1	58\$	ELEGOO UNO super kit
Capteur MQ-135	Capteur de détection de gaz dangereux	3	10\$	Capteur de gaz dangereux
Capteur DSM501A PM2.5	Capteur de détection de poussières de la qualité d'air	1	17\$	Capteur de poussière et qualité d'air
Arduino UNO R3	Arduino UNO R3 utiliser pour programmer le tout	1	17\$	Arduino
Plexiglass	Un plastique transparent	12 X 24	8\$	Plexiglass
ESP8266 ESP-01S	Module émetteur récepteur série Wifi sans fil compatible avec Arduino IDE	5	25\$	ESP8266 ESP-01S
Application ThingView	Application mobile qui permet de démontrer les données en forme de graphique	1	2.75\$	Application ThingView
Coût total du concept				137.75\$

6.2.2 Liste d'équipements

Tableau 6 Liste d'équipement du prototype 2

Nom de l'item	Description	Type	# du prototypage	Source
Thingview	Pour construire l'application	Logiciel	2 et 3	Application Thingview
Arduino IDE	Pour coder l'Arduino	Logiciel	2 et 3	Arduino IDE

6.2.3 Instructions

- Pour le circuit :

Étape 1 : Téléverser le diagramme du circuit retrouvé sur MakerRepo

Étape 2 : Suivre les connexions dans le schéma

- Pour le boîtier :

Étape 1 : Téléverser le fichier stl dans le MakerRepo

Étape 2 : Envoyer le fichier à l'imprimante 3D

Étape 3 : Imprimer le Modèle

6.3 Finalisation de l'application/circuit et du boîtier

6.3.1 NDM (Nomenclature des Matériaux)

Tableau 7 Nomenclature des matériaux du prototype 3

Nom de l'Item	Description	Quantité	Coût unitaire	Liens
ABS Filament	Matériel utilisé pour imprimer la boîte en 3D	1.75 m	60\$	ABS Filament
ELEGOO UNO super kit	Une boîte contenant un capteur de température et humidité (DHT11 et DHT22) ainsi que des fils, écran LCD, une plaque à souder, lumière et résistance	1	58\$	ELEGOO UNO super kit
Capteur MQ-135	Capteur de détection de gaz dangereux	3	10\$	Capteur de gaz dangereux
Capteur DSM501A PM2.5	Capteur de détection de poussières de la qualité d'air	1	17\$	Capteur de poussière et qualité d'air
Arduino UNO R3	Arduino UNO R3 utiliser pour programmer le tout	1	17\$	Arduino
Plexiglass	Un plastique transparent	12 X 24	8\$	Plexiglass
ESP8266 ESP-01S	Module émetteur récepteur série Wifi sans fil compatible avec Arduino IDE	5	25\$	ESP8266 ESP-01S
Application ThingView	Application mobile qui permet de démontrer les données en forme de graphique	1	2.75\$	Application ThingView
Coût total du concept				197.75\$

6.3.2 Liste d'équipements

Tableau 8 Liste d'équipements du prototype 3

Nom de l'item	Description	Type	# du prototypage	Source
Thingview	Pour construire l'application	Logiciel	2 et 3	Application Thingview
Plaque de prototypage	Pour tester le circuit	Matériel temporaire	2	MakerLab
Soudage	Pour souder les fils à l'intérieur du boîtier (Arduino)	Équipement	2	MakerLab
Imprimante 3D	Pour imprimer le boîtier	Équipement	1 et 2	MakerLab
Arduino IDE	Pour coder l'Arduino	Logiciel	2 et 3	Arduino IDE

6.3.3 Instructions

Finaliser le boîtier et vérifier que le concept est complet.

6.4 Essais & validation

Pour le prototype 1, nous avons fait des tests pour l'application. Au début nous comptions utiliser Thinkable, donc nous avons essayé de vérifier la confidentialité en créant l'interface utilisateur et essayer de la connecter à la data base de google. Aussi de vérifier la connectivité de notre prototype d'application avec le Arduino en essayant d'envoyer les données Arduino vers la data base Thinkable. Tous ses tests durant le prototype 1 nous avons réalisé que Thinkable n'était pas très facile d'utilisation et ne marchait pas très bien. Nous avons réussi à établir une interface utilisateur qui correspondait à nos prévisions, mais avons été confronté à certains problèmes dont le fait de connecter notre application à une data base pour recevoir les informations de notre Arduino, donc nous avons changé pour Thingview qui est une application payante, mais beaucoup mieux performante et facile d'utilisation.

Pour le prototype 2, nous avons vérifié si les trous pour la ventilation sont de taille appropriés (au cas qu'ils se referment lors de l'impression).

Pour le prototype 3, nous avons imprimé le boîtier, installé le couvercle de la boîte, placé les capteurs et fils d'une manière à que tout soit clair et propre et compléter le codage pour connecter les capteurs et les faire fonctionner. Pour le couvercle, nous avons opté pour le choix d'un couvercle en pexiglass afin d'être en mesure de voir l'intérieur du boîtier. Cependant, l'ouverture et la fermeture du couvercle se fait en glissant et non à l'aide d'une charnière. Il ne nous a donc pas été possible de faire une ouverture avec charnière car il nous a été dur de faire les trous avec le matériel abs. Suite à cela, nous avons décidé de faire le choix de l'ajout d'un glissoire qui consista à faire des rayures dans la boîte afin d'y mettre la glissoire. Ce dernier fût le meilleur choix car il facilite le mouvement. Enfin, les dimensions de la boîte ont été changées en raison d'une grande taille qui ne nous était pas de grande utilité. Ses dimensions sont à présent 11x21x6.

7. Conclusions et recommandations pour les travaux futurs

En conclusion, notre équipe a réussi à réaliser un produit intéressant pouvant capter la température, l'humidité, les gaz dangereux ainsi que la qualité de l'air notamment grâce à des procédés vu en classe comme la pensée conceptuelle.

Nous n'allons point éluder le fait que les multiples prototypes aux nombres de trois, nous ont permis d'aboutir à notre produit actuel. Les rétroactions de notre client, nous ont été également bénéfique pour notre réalisation.

Nous sommes fiers de notre travail et nous pensons être aller au bout de notre pensée en y mettant sept capteurs vu la taille de boitier que nous avons choisi. Nous sommes sans ignorer que ce projet nous a profondément aidé à nous améliorer de façon technique mais aussi en tant que personne.

APPENDICES

8. APPENDICE I: Fichiers de conception

Tous les fichiers de conceptions se retrouvent dans MakerRepo. Vous pouvez y accéder via le lien suivant : <https://makerepo.com/rkara042/1525.gng1503fb03clear-data> .