

Livrable F – Prototype 1 et rétroaction du client

Maryana Sfeir (300292176)

Grace Shamba-Tsha (300308784)

Aliou Traore (300331413)

Cédric Veilleux (300331293)

Nouria Nininahazwe (300292104)

GNG 1503-B00

Démonstrateurs: Mohamed Bougader et Haitam Zaiker

Le dimanche 5 mars 2023

Université d'Ottawa

Faculté de génie

Table des matières

1. Introduction	3
2. Rétroaction reçue du client.....	3
3. Développement du prototype	3
3.1. Objectif du prototype I :	3
3.2. Prototype de la boîte :	4
3.2.1. Test.....	5
3.3. Prototype du circuit	6
3.3.1. Test.....	6
3.4. Prototype du code	7
3.4.1. Test.....	8
4. Analyse de composante ou de système critique	8
5. Spécification cibles.....	8
6. Plan d'essai de prototypage.....	9
7. Wrike.....	10
8. Conclusion.....	10
9. Référence	10

1. Introduction

Le but de ce document est de développer notre premier prototype et d'établir un plan d'essai pour le prochain prototype. De plus, à partir de ce premier prototype nous obtiendrons de la rétroaction de nos clients pour améliorer notre prototype.

2. Rétroaction reçue du client

Le client trouve que nous ayons mis au point un concept de produit écologique en utilisant des piles rechargeables. Il est encourageant de voir une équipe qui prennent des mesures pour adopter des pratiques plus durables et plus écologiques.

L'idée d'utiliser une boîte à température contrôlée et d'avoir une batterie incluse dans la conception a été considérée comme une caractéristique intéressante. Cela permettrait une plus grande flexibilité en termes d'endroit où le produit peut être placé, sans avoir à se soucier de la proximité d'une prise de courant.

Pour ajouter, il a été suggéré d'utiliser un plastique résistant à la chaleur pour les clips, afin de garantir la résistance et durabilité du produit.

En ce qui concerne la communication avec l'application, le client n'a pas de préférence, mais il est important de prendre en compte les préférences de l'utilisateur et de s'assurer qu'il est tenu informé et mis à jour. Que ce soit par SMS ou par courriel, l'utilisateur aura la possibilité de choisir la méthode qu'il préfère.

3. Développement du prototype

3.1. Objectif du prototype I :

L'objectif du prototype I c'est de créer un modèle de la boîte afin de savoir le positionnement et l'espacement des composantes à l'intérieur de la boîte. Par la suite, nous voulons créer un prototype qui sera tester pour observer la résistance de la boîte en cas de chute, afin de savoir les modifications nécessaires afin de rendre notre boîte le plus résistant possible. De plus, nous développerons un prototype analytique d'un circuit avec les capteurs afin de recueillir les différentes données ou recueillir des données lorsqu'il y a un changement irrégulier de la température, l'humidité ou de la qualité de l'air dans différents zones ou emplacement d'une salle. Ce prototype devra être en mesure d'avoir des données les plus précis. Nous créerons un prototype analytique pour un code afin de programmer les circuits et les capteurs afin de pouvoir être capable d'alerter les utilisateurs d'un changement. Le code devra être en mesure d'avoir la précision dans les données.

3.2. Prototype de la boîte :



Figure 1 – L'extérieur du prototype 1 (avec caisse)



Figure 2 - L'intérieur du prototype 1

3.2.1. Test

Espacement à l'intérieur de la boîte et positionnement : Tester le positionnement de chaque composante (Les capteurs, batteries, fils) dans la boîte. Tester si la boîte à un bon espacement.

- Durée : 5 minutes
- Résultats :

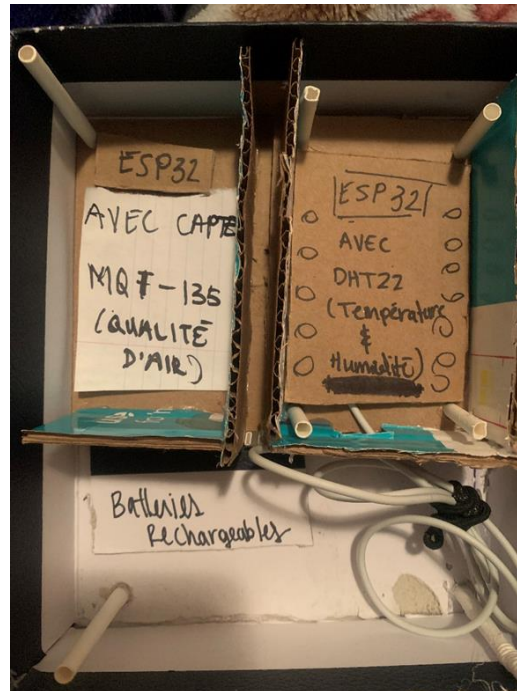


Figure 3 - Positionnement des composantes

Résistance :

- Durée : 30 minutes
- Résultats : Le prototype résiste lorsqu'il est jeté des escaliers.

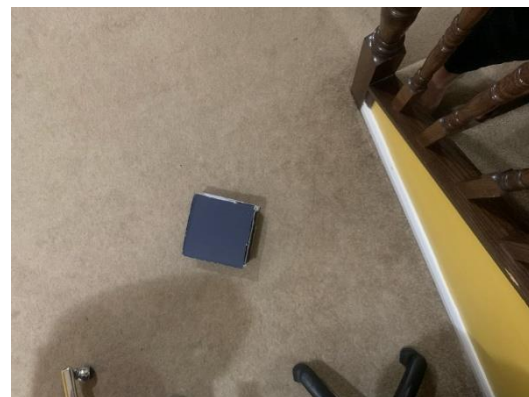


Figure 4 - Résistance du boîte

3.3. Prototype du circuit

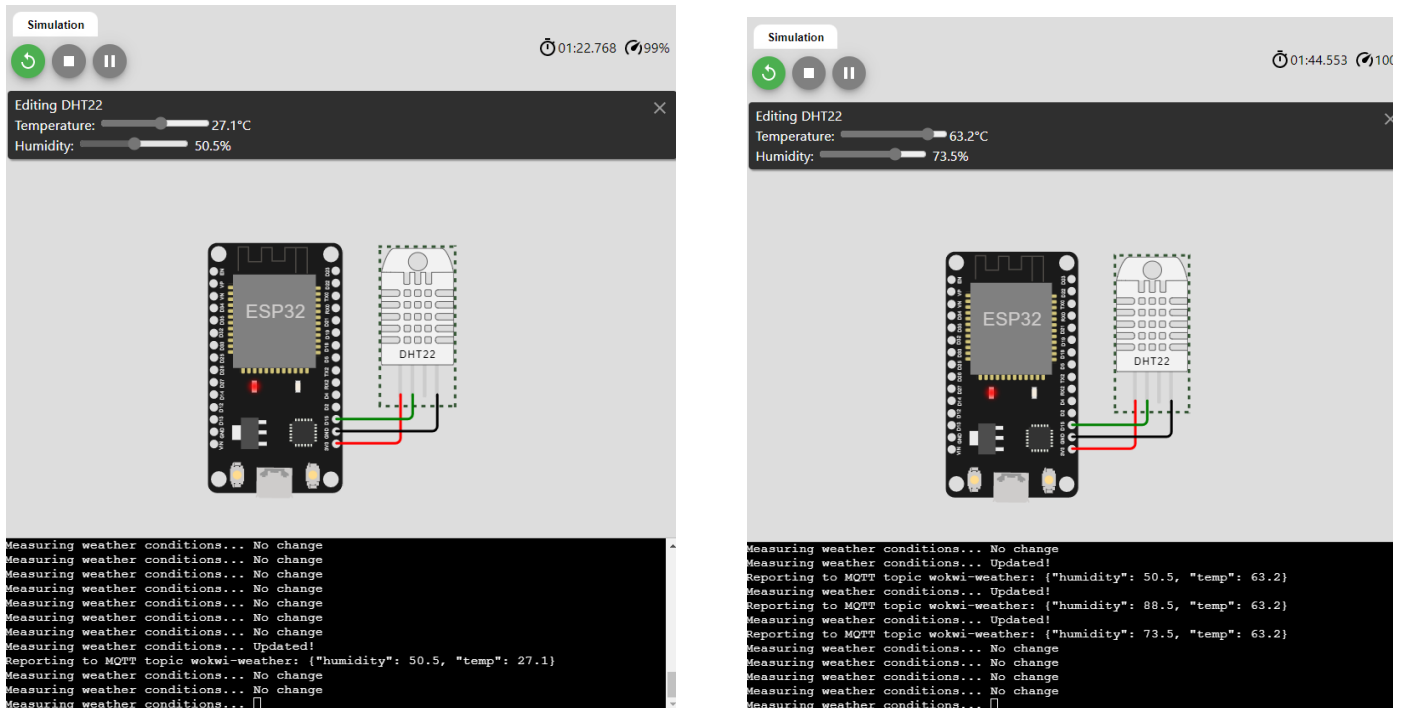


Figure 5 - Prototype du circuit (Température et Humidité DHT22)

(Interface DHT11 DHT22 w/ ESP8266 NodeMCU Using Web Server, 2023)

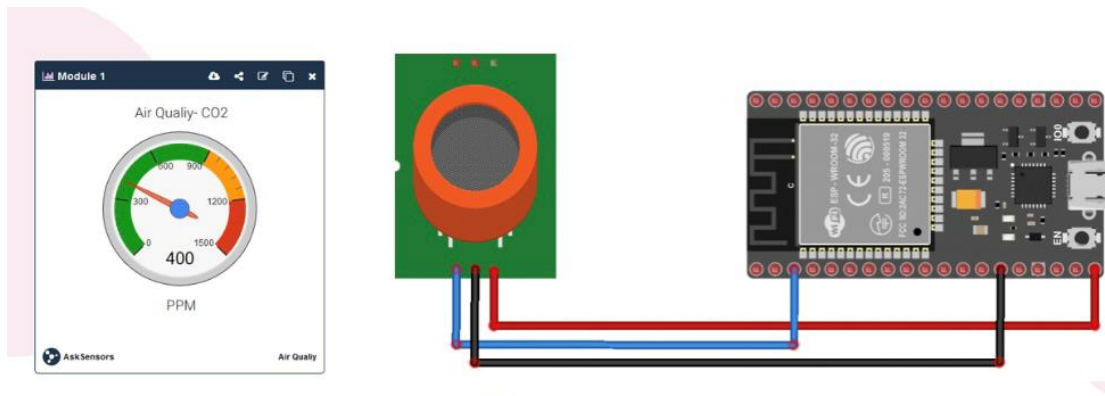


Figure 6 - Prototype du circuit (Qualité d'Air MQT-135)

(<https://blog.asksensors.com/air-quality-sensor-mq135-cloud-mqtt/>, 2023)

3.3.1. Test

Dans ce prototype de circuits nous avons tester le fonctionnement du circuit lorsque la température ou l'humidité augmente ou diminue dans une salle.

- Durée du test : 1 heure
- Résultats : Prend la température et l'humidité à différentes zones.

3.4 Prototype du code

```
19
20 import network
21 import time
22 from machine import Pin
23 import dht
24 import ujson
25 from umqtt.simple import MQTTClient
26
27 # MQTT Server Parameters
28 MQTT_CLIENT_ID = "micropython-weather-demo"
29 MQTT_BROKER = "broker.mqttdashboard.com"
30 MQTT_USER = ""
31 MQTT_PASSWORD = ""
32 MQTT_TOPIC = "wokwi-weather"
33
34 sensor = dht.DHT22(Pin(15))
35
36 print("Connecting to WiFi", end="")
37 sta_if = network.WLAN(network.STA_IF)
38 sta_if.active(True)
39 sta_if.connect('wokwi-GUEST', '')
40 while not sta_if.isconnected():
41     print(".", end="")
42     time.sleep(0.1)
43 print(" Connected!")
44
45 print("Connecting to MQTT server... ", end="")
46 client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER, user=MQTT_USER, password=MQTT_PASSWORD)
47 client.connect()
48
49 print("Connected!")
50
51 prev_weather = ""
52 while True:
53     print("Measuring weather conditions... ", end="")
54     sensor.measure()
55     message = ujson.dumps({
56         "temp": sensor.temperature(),
57         "humidity": sensor.humidity(),
58     })
59     if message != prev_weather:
60         print("Updated!")
61         print("Reporting to MQTT topic {}: {}".format(MQTT_TOPIC, message))
62         client.publish(MQTT_TOPIC, message)
63         prev_weather = message
64     else:
65         print("No change")
66     time.sleep(1)
67
```

Figure 7 - Prototype du code

```
//THONAIN MODULE CAPTEUR QUALITE AIR MQ-135 + ECRAN LCD
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
int measure;
void setup() {
  Serial.begin(9600);
  lcd.begin(16,2); // initialize the lcd
  lcd.backlight();
  lcd.print("TRONIK AVENTUR");
  delay(2000);
  lcd.setCursor(1,1);
  lcd.print("CAPTEUR MQ-135");
  pinMode(2, OUTPUT); // BUZZER
  pinMode(5, OUTPUT); // LED ROUGE
  pinMode(6, OUTPUT); // LED JAUNE
  pinMode(7, OUTPUT); // LED VERTE
  digitalWrite(2, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  delay(2000);
  lcd.clear();
}
void loop() {
  measure = analogRead(A0);
  Serial.print("valeur mesuree :");
  lcd.setCursor(1,1);
  lcd.print(measure);
  lcd.setCursor(0,0);
  if (measure < 80 ) {
    digitalWrite(7, HIGH);
    digitalWrite(6, LOW);
    digitalWrite(5, LOW);
    digitalWrite(2, LOW);
    lcd.print(" AIR CORRECT");
  }
  else
  {
    if (measure < 140 ) {
      digitalWrite(7, LOW);
      digitalWrite(6, HIGH);
      digitalWrite(5, LOW);
      digitalWrite(2, LOW);
      lcd.print(" AIR MALSAIN");
    }
    else
    {
      digitalWrite(7, LOW);
      digitalWrite(6, LOW);
      digitalWrite(5, HIGH);
      digitalWrite(2, HIGH);
      lcd.print("ALERTE POLLUTION");
    }
  }

  delay(1000);
  lcd.clear();
}
```

Figure 8 - Prototype du code (Qualité d’Air MQT-135)

3.4.1 Test

Dans le prototype nous testons le fonctionnement du code. Ce code peut être connecter avec et sans wifi. Le résultat de ce test affiche le changement de la température ou l'humidité. Les données sont mises à jour lorsqu'il y a un changement. Pour le prototype du code pour le capteur de la qualité d'air le capteur démontre les résultats tels que démontrer dans la figure 4.

- Durée du test : 2 heures
- Résultats :

```
Measuring weather conditions... No change
Measuring weather conditions... Updated!
Reporting to MQTT topic wokwi-weather: {"humidity": 50.5, "temp": 63.2}
Measuring weather conditions... Updated!
Reporting to MQTT topic wokwi-weather: {"humidity": 88.5, "temp": 63.2}
Measuring weather conditions... Updated!
Reporting to MQTT topic wokwi-weather: {"humidity": 73.5, "temp": 63.2}
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... [
```

Figure 9 - Changement de la température ou l'humidité

4. Analyse de composante ou de système critique

Les capteurs comme celui de la qualité d'air (MQ135 Air quality detector sensor) et celui de la température et humidité (DHT 11 Digital Temperature and Humidity) sont des composantes critiques de notre système. L'usure, les dommages et les interférences sur ces composantes peuvent nous donner des données inexactes, ce qui entraînera une mauvaise évaluation de la qualité d'air et la précision de la température et de l'humidité. Ceci pourrait causer des dommages aux serveurs.

Pour prévenir les défaillances sur le système, des opérations de contrôle, de calibrage et de test peuvent être faites régulièrement pour éviter les dommages physiques, l'usure et les erreurs de mesures, les dommages et l'usure. On pourrait également mettre des capteurs de secours pour éviter d'avoir une discontinuité dans les données.

5. Spécification cibles

- Nous voulons que notre produit reste dans notre budget de 400\$.
- Nous voulons que nos composantes et matériaux ait un impact environnemental le plus minime possible.
- Nous aimerions que la boîte soit faite en aluminium.
 - Nous voulons qu'elle puisse être visé (à partir de vis) (Une surface qui dépasse la base de la boîte).
 - Nous voulons que trois compartiments se retrouve à l'intérieur de cette boîte (Alimentation, circuit imprimé, fils et capteur).
 - Nous voulons que la boîte ait des trous sur les parois supérieur et inférieur pour qu'il y ait une circulation d'air.
 - Nous voulons que la batterie, le circuit imprimé, fils et capteur sois fixer dans la boîte à partir de support de fixation. Ceux-ci se retrouverons dans des boites fixer à partir de vis et ceux-ci auront un gros trou pour laisser une circulation d'air.
 - Le poids de la boîte, des composantes et des matériaux devront permettre à l'utilisateur de mettre la boîte partout dans la salle de serveur.

- Nous voulons que la boîte ait un volume de 35x25x15 cm.
- Nous voulons utiliser un circuit imprimé (ESP-32).
 - Nous voulons que ce circuit puisse faire fonctionner plusieurs capteurs au même moment. (Température, humidité, système de son)
 - Nous voulons que les informations de circuit imprimé soit envoyer à une application. (L'utilisateur aura accès à cette application et pourra observer les données)
 - L'utilisateur pourra lui-même décider lui-même à partir de quelle valeur de donnée l'application va leur envoyer une alarme.
 - Nous voulons que l'utilisateur puisse décider lui-même qu'elle type d'alarme qu'il veut être un courriel ou un SMS. Il pourra aussi décider de la langue du programme. (Français ou anglais)
 - Nous voulons que les données sur l'application soient démontrées à partir de chiffre.
 - Nous voulons que les données puissent être envoyer sans réseau si jamais le produit se retrouve dans un endroit qui n'en contient pas.
 - Nous voulons que les capteurs choisis aient une précision minimale de 2% de marge d'erreur. Ces composant ne devront pas interférer avec la prise de données.
 - Nous voulons que lorsque le produit fonctionne, qu'une lumière (LED) sois ouverte pour que les personnes qui travaille sur place pour savoir que ceux-ci fonctionnent. Une lumière rouge s'il y a un problème dans le circuit.
- Nous voulons que la source d'alimentation soit faite à partir d'une prise au mur lorsqu'il y a de l'électricité et lorsque la salle de serveur manque d'électricité, ceux-ci utiliseront des batteries rechargeables. De cette façon, les données seront toujours envoyées et la fiabilité du produit sera encore meilleure.

6. Plan d'essai de prototypage

Tableau 1 - Plan de test des prototypes

N° de Test	Objectif du Test (Pourquoi)	Description du Prototype Utilisé et de la Méthode de Test de Base (Quoi)	Description des Résultats à Documenter et Comment ces Résultats seront Utilisés (Comment)	Durée Estimée du Test et Date Prévues du Début du Test (Quand)
1	Tester le code	Analytique complet	L'affichage des températures, de l'humidité et de la qualité de l'air données. Installer esp 32 dans Arduino ide. Le logiciel en question est gratuit et avec ou sans wifi.	Durée : 5 jours Du 5 au 10 Mars
2	Tester l'application	Analytique complet	L'utilisation de IOT pour pouvoir nous présenter les données recueillies par les capteurs. Une	Durée : 1 Semaine

			marge d'erreur vraiment petite a 2%. Donnée présenter seront facile à lire pour toute utilisateur.	Du 11 au 18 Mars
3	Tester le circuit	Analytique ciblé	Sur Arduino, les matériaux seront les capteurs (température, humidité et qualité d'air), les fils et le esp 32	Durée : 1 semaine Du 6 au 11 Mars
4	Tester la source d'alimentation (batterie rechargeable et prise au mur)	Physique et Analytique complet	Pour l'environnement on pensait plus utiliser les batteries rechargeables et aussi en cas de panne de courant sinon retourner aux prises de mur. Voir le temp que la batterie tient avant d'avoir besoin d'une recharge.	Durée : 1 semaines Du 7 au 14 Mars

(Bouendeu, 2023)

7. Wrike

<https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=g9v6pglmGfYxZxNbeQeeei56KHfWvAY%7CIE2DSNZVHA2DELSTGIYA>

8. Conclusion

En conclusion, ce livrable a été utiliser pour démontrer notre premier prototype, ce que nous nous attendons de notre prototype et l'essai de ce prototype. Nous pouvons donc voir qu'elle type de capteurs nous voulons utiliser, quel type de boîte nous voulons utiliser, une avant-première du code à utiliser et toute nos spécifications cible envers ce produit.

9. Référence

Bouendeu, E. (2023, Mars 5). *Modèle de plan de test*. Récupéré sur GNG 1503 - Brightspace: <https://uottawa.brightspace.com/d2l/le/content/350158/viewContent/5156115/View>

<https://blog.asksensors.com/air-quality-sensor-mq135-cloud-mqtt/>. (2023, Mars 5). Récupéré sur The AskSensors Blog: <https://blog.asksensors.com/air-quality-sensor-mq135-cloud-mqtt/>

Interface DHT11 DHT22 w/ ESP8266 NodeMCU Using Web Server. (2023, Mars 5). Récupéré sur Last Minute Engineers: DHT11 DHT22 w/ ESP8266 NodeMCU : Display Values Using Web Server (lastminuteengineers.com)

Thonain. (2021, avril 30). TRONIK AVENTUR 298 - CAPTEUR QUALITE AIR MQ135 CO2 GAZ NH3.