

GNG1503

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

Weather Live Box

Soumis par:

Toe Beans - FB 33

Jérémie Galipeau 300324127

Tyler Sauve 300295582

Andrea Khater 300292634

Khloé Thibus 300319336

Kevin Ghneim 300327077

Exaucé Kamanga 300321343

16 avril 2023

Université d'Ottawa

Table des matières

Table des matières

Liste de figures

Liste de tableaux

Liste d'acronymes et glossaire

1 Introduction

2 Aperçu

2.1 Conventions

2.2 Mises en garde et avertissements

3 Pour commencer

3.1 Considérations pour la configuration

3.2 Considérations pour l'accès des utilisateurs

3.3 Accéder au système

3.4 Organisation du système & navigation

3.5 Quitter le système

4 Utiliser le système

4.1 Fonction/Caractéristique donnée

5 Dépannage & assistance

5.1 Messages ou comportements d'erreur

5.2 Considérations spéciales

5.3 Entretien

5.4 Assistance

6 Documentation du produit

6.1 Design

6.1.1 NDM (nomenclature des matériaux)

6.1.2 Liste d'équipements

6.1.3 Instructions

6.2 Circuit

6.2.1 NDM (nomenclature des matériaux)

6.2.2 Liste d'équipements

6.2.3 Instructions

6.3 Codage

6.3.1 NDM (nomenclature des matériaux)

6.3.2 Liste d'équipements

6.3.3 Instructions

6.4 Essais & validation

7 Conclusions et recommandations pour les travaux futurs

8 Bibliographie

APPENDICES

9 APPENDICE I: Fichiers de conception

Liste de figures

Aperçu

- Figure 1: photo 1 du prototype final
- Figure 2: photo 2 du prototype final
- Figure 3: schéma de la boîte du produit final
- Figure 4: schéma du circuit du produit final
- Figure 5: schéma de l'application du produit final

Pour commencer

- Figure 6: photo du prototype final
- Figure 7: capture d'écran du modèle
- Figure 8: capture d'écran du devant du modèle
- Figure 9: capture d'écran du derrière du modèle
- Figure 10: capture d'écran du côté du modèle
- Figure 11: capture d'écran du dessus du modèle
- Figure 12: photo des murs de côtés
- Figure 13: photo des murs de devant et derrière
- Figure 14: photo du couvercle et du fond
- Figure 15: photo du circuit complet
- Figure 16: code
- Figure 17: options du WiFi
- Figure 18: lecture des données
- Figure 19: page Info
- Figure 20: page Metadata
- Figure 21: page Datastream
- Figure 22: page Events
- Figure 23: page Automations
- Figure 24: page Web Dashboard

Documentation du produit

- Figure 25: Capture d'écran du menu du logiciel Blender
- Figure 26: Capture d'écran de l'importation de fichiers STL
- Figure 27: Capture d'écran de la boîte
- Figure 28: Capture d'écran du couvercle
- Figure 29: Capture d'écran de l'exportation de fichier STL
- Figure 30: Capture d'écran d'un fichier STL
- Figure 31: Capture d'écran du menu du logiciel UltiMaker Cura
- Figure 32: Capture d'écran du menu du logiciel UltiMaker Cura
- Figure 33: Capture d'écran de l'importation d'un fichier
- Figure 34: Capture d'écran des paramètres d'impression
- Figure 35: Capture d'écran du bouton découpage
- Figure 36: Capture d'écran d'une pièce découpée
- Figure 37: Images du micro contrôleur Arduino MKR WiFi 1010, ensemble breadboard et du capteur Adafruit BME 680
- Figure 38: Schéma du circuit
- Figure 39: Photos du circuit
- Figure 40: Capture d'écran du téléchargement d'Arduino IDE
- Figure 41: Capture d'écran du menu du logiciel Blynk
- Figure 42: Capture d'écran du menu de template 1
- Figure 43: Capture d'écran de la page Info

- Figure 44: Capture d'écran de la page Metadata
- Figure 45: Capture d'écran de la page Datastreams
- Figure 46: Capture d'écran de la page Event
- Figure 47: Capture d'écran de la page Automations
- Figure 48: Capture d'écran de la page Web Dashboard
- Figure 49: Capture d'écran du code 1
- Figure 50: Capture d'écran du code 2
- Figure 51: Capture d'écran du code 3
- Figure 52: Capture d'écran du code 4
- Figure 53: Photo de l'intérieur de la boîte finale
- Figure 54: Captures d'écran des données capturées sur Arduino IDE
- Figure 55: Captures d'écran des données capturées sur Arduino IDE avec le ventilateur
- Figure 56: Capture d'écran du code 5
- Figure 57: Capture d'écran du code 6
- Figure 58: Capture d'écran du code 7
- Figure 59: Capture d'écran du code 8

Liste de tableaux

Table 1: Acronymes

Table 2: Glossaire

Table 3: Connection entre le micro contrôleur et le capteur

Table 4: NDM de la boîte

Table 5: Liste d'équipement pour la boîte

Table 6: Fonction des pins du capteur BME 680

Table 7: NDM du circuit

Table 8. Liste d'équipement pour le circuit

Table 9: NDM du logiciel

Table 10: Liste d'équipement pour le codage

Table . Documents référencés

Liste d'acronymes et glossaire

Table 1. Acronymes

| Acronyme | Définition |
|----------|------------------------------------|
| NDM | Nomenclature des matériaux |
| MUP | manuel d'utilisation et de produit |

| | |
|-----|--|
| PLA | Acide polylactique: plastique utilisé pour l'impression 3D. |
| RPM | Revolution per minute: tour par minute |
| STL | Stéréolithographie: format de fichier utilisé pour l'impression 3D |
| WLB | Weather Live Box: nom du projet |

Table 2. Glossaire

| Terme | Définition |
|------------------|---|
| Blynk | Application permettant de visualiser les données captées. |
| Breadboard | Planche de pain: planche permettant de fabriquer et tester des circuits électriques. |
| Découper | Le découpage en impression 3D est le processus par lequel un modèle est découpé en petite couche une par dessus l'autre pour empiler le matériau. |
| Ethernet | Connection internet avec un câble. |
| Hot spot | Connection internet via WiFi à proximité. |
| imprimante 3D | Machine permettant d'imprimer des modèles numériques en objets tangibles. |
| librairie | Préréglage pour la programmation d'un code disponible sur Internet ou sur Arduino IDE. |
| Micro contrôleur | Carte mère d'un circuit permettant de d'appliquer le code. |

| | |
|----------------|--|
| Micro USB | Type de cable. |
| Open source | Disponible gratuitement à tous et modifiable |
| Serial Monitor | Sur Arduino IDE, fenêtre permettant de voir les données captées. |

1 Introduction

Ce manuel d'utilisation et de produit (MUP) fournit les informations nécessaires aux utilisateurs pour utiliser efficacement le Weather Live Box(WLB) et pour la documentation du prototype.

Ce projet consiste en un système de surveillance de l'environnement des centres de données permettant de recevoir en temps réel les variations des conditions environnementales par l'entremise d'une application. Nous supposons que tout doit être contenu dans une boîte ainsi que les données transférées par Wifi vers une application ou site Web.

Ce document contient un aperçu du projet, comment le commencer et l'installation du système, l'utilisation du système et sa documentation, le dépannage et l'assistance, la documentation du produit ainsi que des recommandations. Ce document est destiné à tous ceux qui voudraient refaire ce produit basé sur le concept élaboré par l'équipe et servir de guide pour réaliser cette tâche.

Tout le matériel du projet a été fabriqué de toute pièce par l'équipe ou est open source sur Internet. Nous donnons permission de copier, modifier et améliorer ce travail, tant que nous sommes reconnus et que nous sommes crédités comme les créateurs originels de ce produit.

2 Aperçu

Les clients ont de l'équipement dans des centres de données, mais ils ne sont pas sur place pour assurer la maintenance de l'équipement. Ils recherchent donc une solution qui leur permettrait de pouvoir surveiller le centre de données. Dans le passé, il est advenu qu'une

situation d'urgence est arrivée et que l'équipement a été endommagé, mais les clients n'avaient aucune façon de le savoir.

Les clients ont demandé que la solution soit facile d'utilisation, qu'elle ne soit pas plus grosse qu'une boîte de soulier, que la solution capte la température, la qualité de l'air et l'humidité et transfère les données vers un logiciel pouvant les rendre lisibles en temps réel.

Notre produit est simple de fabrication et d'utilisation. En attachement dans ce projet, vous trouverez le modèle numérique de la boîte pour la créer. La boîte est imprimée en 3D en PLA, un matériau recyclable et naturel. Pour faire fonctionner le système, il ne suffit que de le brancher dans un port USB. Une fois en marche, les données sont transmises vers l'application Blynk, où elles peuvent être visualisées en temps réel dans des graphiques et des figures. La boîte a été conçue en gardant en tête que le produit peut être modifié au besoin des utilisateurs. Elle comporte assez d'espace pour ajouter des éléments au circuit. Le code utilisé est **Open Source**, ce qui veut dire qu'il est accessible à tous et qu'il y a beaucoup d'informations en ligne pour l'améliorer.



Figure 1: photo 1 du prototype final

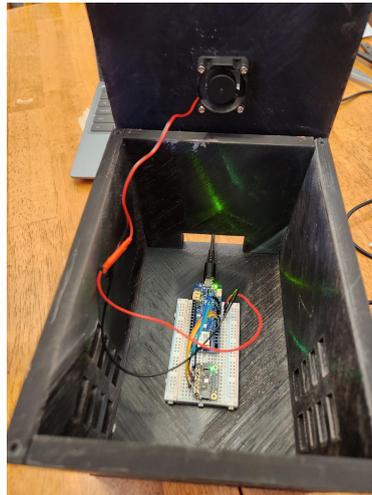


Figure 2: photo 2 du prototype final

Notre boîte était conçue dans le but de récolter des données pour prévenir les catastrophes dans le centre des données. Grâce à tous nos systèmes par exemple: le capteur, les codes, la carte d'arduino et l'application nous permettent de régler le problème de nos clients tout en surveillant leur centre autant de loin que sur place.

Notre produit est fait en PLA grâce à l'imprimante 3D. Chaque côté prenait environ 10 heures pour être imprimé. Comme vous pouvez voir, la boîte est faite de plusieurs trous. Pour le côté de derrière, il y a un trou qui permet au câble d'alimentation de passer ce qui fait en sorte que notre circuit fonctionne tout en captant les données. Ensuite, pour les côtés de devant et ceux de gauche et de droite, il y a plein de petits trous pour aérer le circuit et le capteur au cas d'un surchauffement. Ces trous permettent aussi au capteur de capter les données requises à l'extérieur de la boîte et non à l'intérieur. Pour finir, le couvercle à un trou pour mettre le ventilateur qui est face vers le haut pour que la précision du BME 680 ne soit

pas affectée. De plus, il y a aussi quatre petits trous pour les vis. Celles-ci permettent de visser ce qui rend l'accessibilité plus facile au circuit et un couvercle plus solide. Maintenant rentrons plus dans le détail du circuit. Le carte arduino est faite de wifi ce qui fait en sorte que les données peuvent être transmises à notre application blynk. De plus, comme nous l'avons installé, notre application peut envoyer des notifications si la température est trop élevée ou si l'humidité est trop intense pour prévenir les catastrophes qui peuvent arriver aux équipements. Ensuite, notre capteur BME 680 peut capter le gaz, la température, l'humidité et la pression. Donc, elle a plusieurs modes. Ce petit capteur est connecté à la carte arduino pour transmettre les informations recueillies. Pour finir, la carte arduino a besoin d'un câble **micro USB** pour le brancher à l'ordinateur et le mettre en fonction.

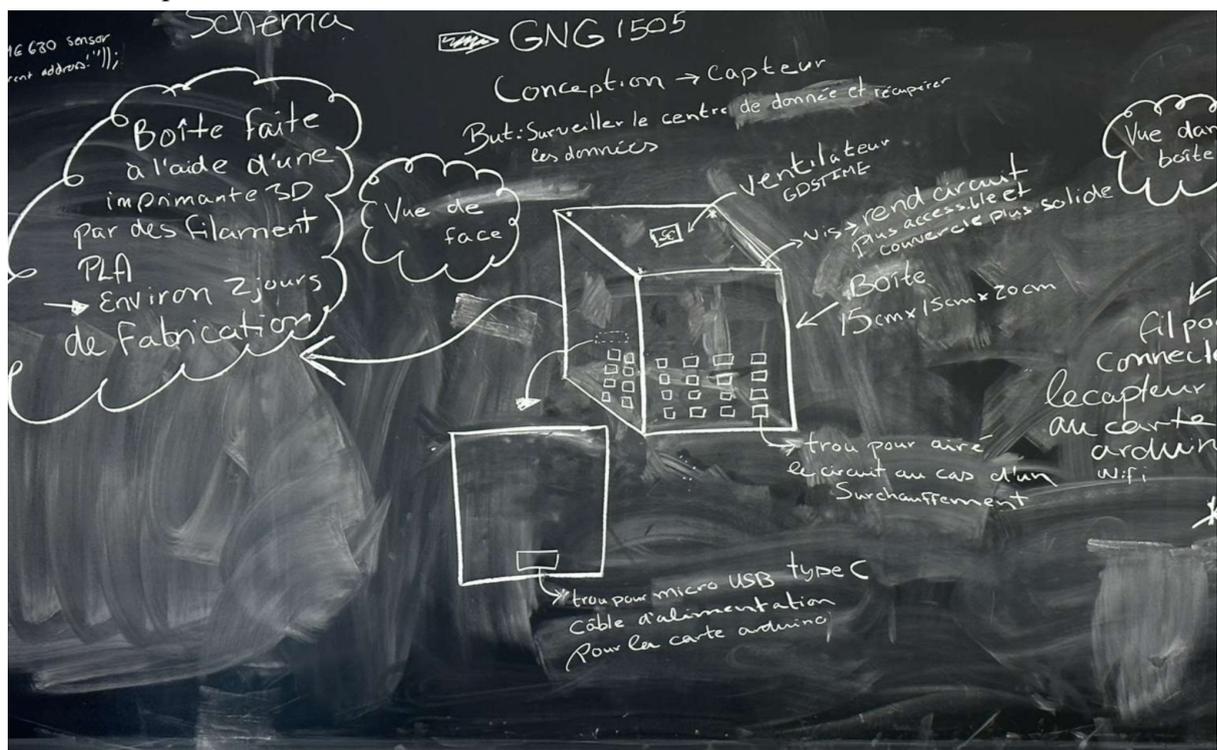


Figure 3: Schéma de la boîte du produit final

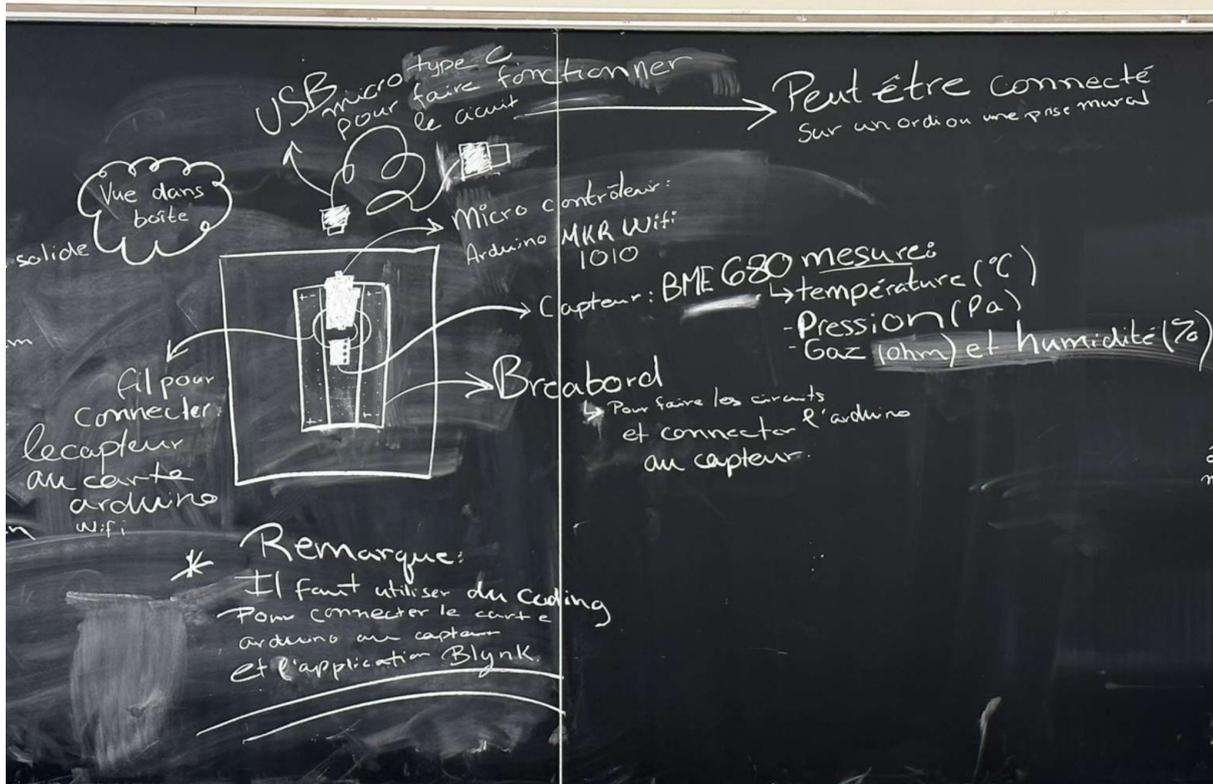


Figure 4: Schéma du circuit du produit final

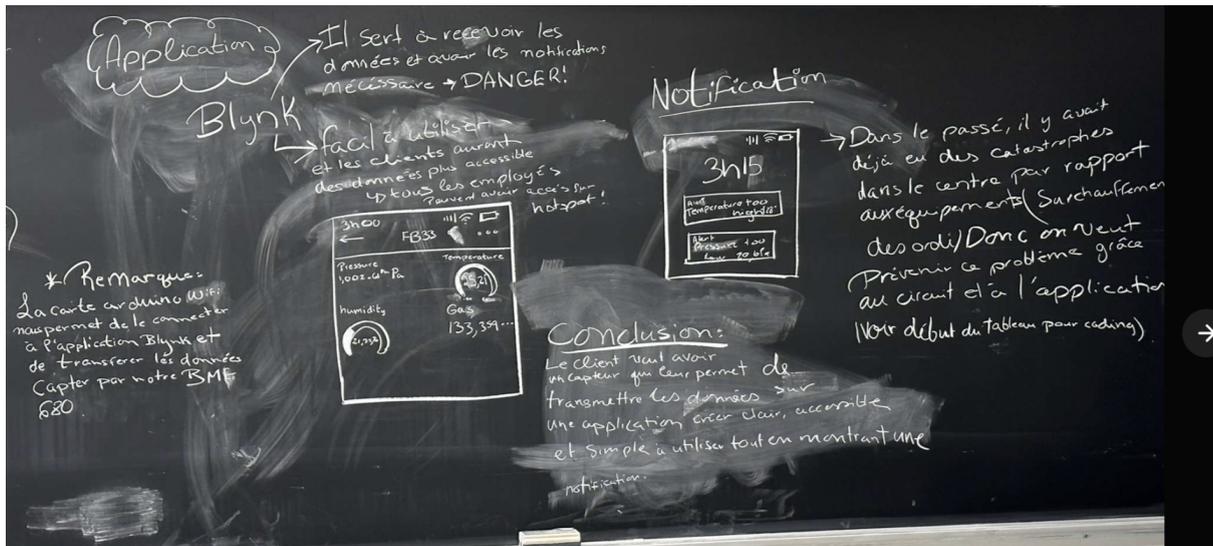


Figure 5: Schéma de l'application du produit final

2.1 Conventions

- Les termes techniques sont en gras et sont définis dans le glossaire.
- Les actions à effectuer par l'utilisateur sont indiquées par un verbe d'action suivi de l'instruction.
- Les boutons et les commandes du logiciel sont écrits en majuscules.
- Les messages d'erreur sont écrits en gras et sont précédés de "Erreur :".
- Les avertissements importants sont écrits en gras et en italique

2.2 Mises en garde et avertissements

AVERTISSEMENT : Le non-respect des consignes de sécurité peut entraîner des blessures graves ou des dommages matériels.

- Risque électrique : Le capteur de température est alimenté par une source électrique. Pour éviter tout risque de choc électrique, assurez-vous de débrancher le capteur avant toute manipulation ou maintenance.
- Risque de brûlure : Le capteur peut devenir chaud pendant l'utilisation. Manipulez-le avec précaution pour éviter les brûlures.
- Précautions : Portez des gants de protection en cas de manipulation du capteur lorsqu'il est chaud. Ne touchez pas les fils électriques nus.
- Garantie : Ce capteur est garanti uniquement pour une utilisation prévue. Toute utilisation autre que celle prévue peut annuler la garantie.

3 Pour commencer

Pour commencer, notre système comprend deux types de composantes. La première, le prototype physique et le deuxième le prototype logiciel. Dans les prochaines étapes, le manuel va vous expliquer étape par étape comment assembler la boîte, comment bien connecter le capteur et le ventilateur aux microcontrôleurs, le code et tout ce que vous avez besoin de savoir pour avoir une application fonctionnelle.



Figure 6: Photo du prototype final

3.1 Considérations pour la configuration

Prototype physiques

1 - La boîte

Le but d'avoir une boîte est de garder toutes les composantes qui sont dedans en toute sécurité. Pour nous, nous avons décidé d'utiliser une **imprimante 3D** pour que la boîte soit durable et efficace. De plus, ceci ajoute un niveau de fiabilité a notre boîte en cas de défaillance. Voir 3.3 pour la préparation des morceaux et 3.4 pour l'assemblage.

2 - Le circuit

Le circuit permet de lire les données du capteur et ensuite l'envoyer à notre application. Notre application fonctionnant par wifi donc il est important d'avoir un microcontrôleur qui fonctionne par wifi. Voir 3.3 pour la préparation des morceaux et 3.4 pour l'assemblage.

Prototype logiciel

1 - L'application

L'application qui est la plus simple et efficace à utiliser et naviguer est l'application blynk. Cette application va capturer toutes les données du capteur et va être sauvegardée dans le Blynk.cloud. Ensuite, l'application peut être télécharger sur IOS ou Android et si le client envoyé le hotspot à ces membre de son organisation, tous les utilisateur auront les données actuels en ce même temps.

3.2 Considérations pour l'accès des utilisateurs

Quand vous allez créer votre application blynk, chacun des projets contient un adresse spécifique, le nom du template et le nom du hotspot. Donc les différents utilisateurs et/ou groupes d'utilisateurs qui pourraient utiliser le produit sont tous les membres qui auront accès au hotspot du template/projet. Les restrictions imposées à l'accessibilité ou à l'utilisation de

quelqu'un peuvent être modifiées. Par exemple, quelqu'un qui vient d'accéder au projet va seulement pouvoir voir les données actuelles sous forme de graphique et étiquette. Ceci peut être modifié par la personne qui a créé le projet.

3.3 Accéder/installation du système

1 - La boîte

Pour la boîte, tandis que nous allons le 3D print, vous avez besoin d'une imprimante 3D ainsi que du filament PLA. De plus pour créer un projet que tu peux 3D print, il faut utiliser un logiciel comme onshape ou blender.

2 - Le circuit

Pour le circuit, vous avez besoin d'une couple de morceaux

- **Microcontrôleur** : Arduino MKR 1010 WiFi
- Capteur de temperature, gas, qualité d'aire, humidité
- **Breadboard**
- Fils électriques

Voir 3.4 pour l'installation des morceaux

3 - L'application

Nous allons utiliser l'application **Blynk**. Les projets Blynk peuvent être modifiés sur le blynk.cloud pour la base du projet et sur le téléphone.

3.4 Organisation du système & navigation

1 - La Boite

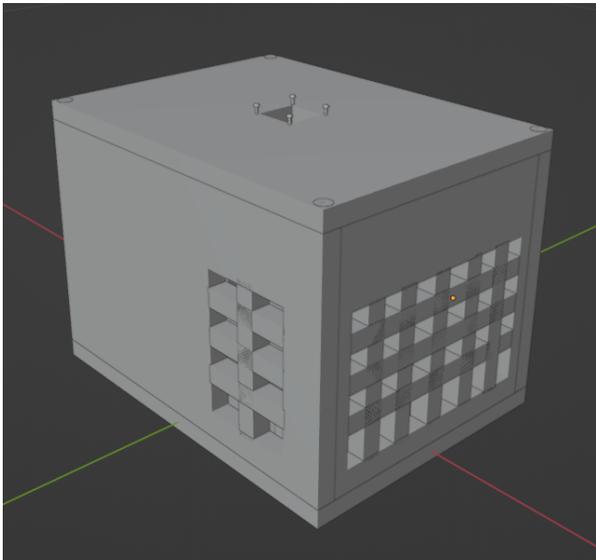


Figure 7: Capture d'écran du modèle

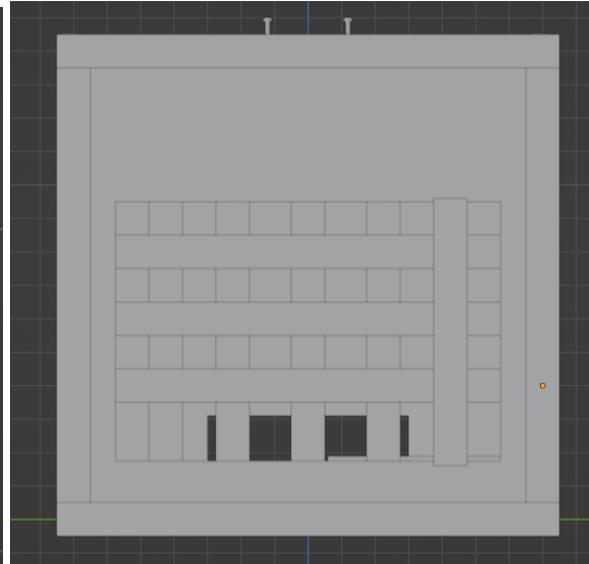


Figure 8: Capture d'écran du devant du modèle

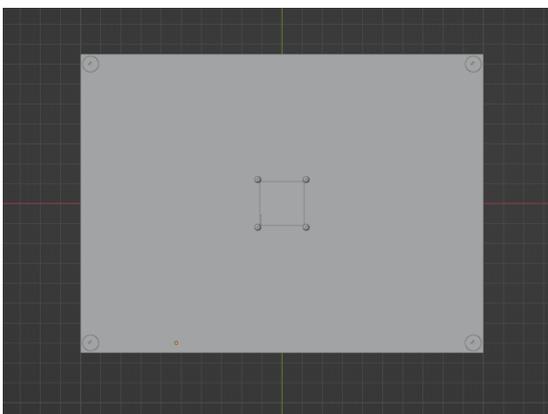
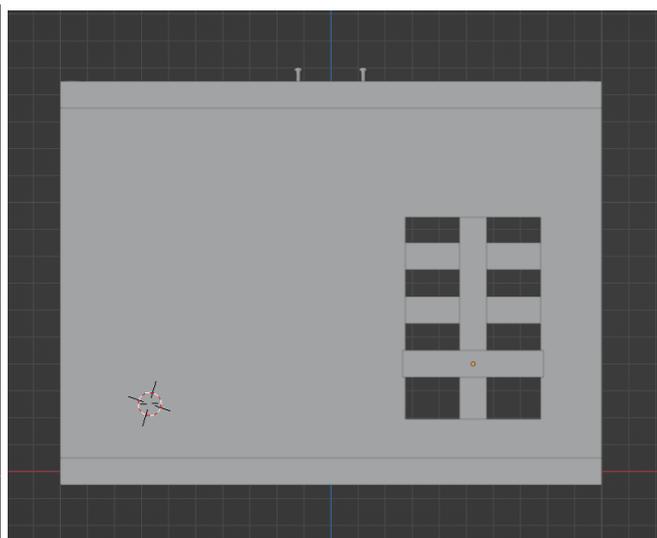
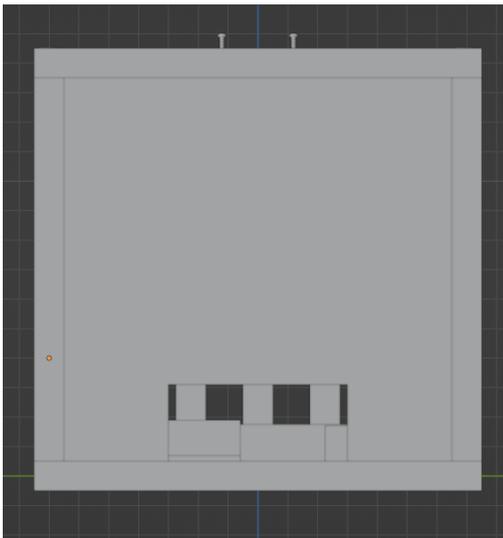


Figure 9, 10 et 11: Capture d'écran du derrière, coté et dessus du modèle

Les clients se questionnaient sur le design puisque les trous changeaient la fonction de la boîte. Ils veulent une boîte fermée avec de l'aération et le design du prototype 1 était plus ou moins une boîte ouverte. Nous avons donc fermé les murs et laissé des trous pour que les capteurs puissent bien fonctionner, surtout le capteur de poussière.

Un autre aspect que les clients étaient que les tiges dans les trous étaient peut-être trop petites et donc la structure pourrait briser facilement. Avec le changement de design, ce problème a disparu et la boîte est également beaucoup plus solide.

la boîte imprimée en 3D vas ressembler comme ceci:



Figure 12: murs des côtés



Figure 13: Murs du devant et derrière

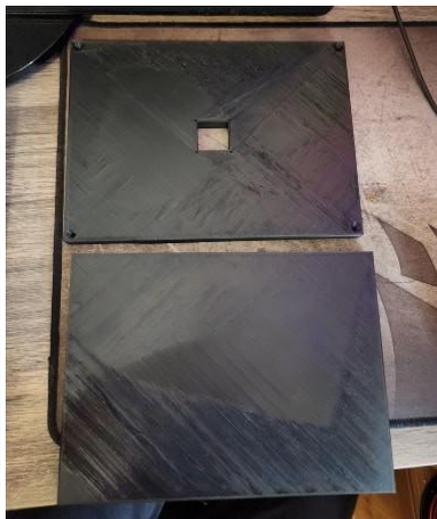


Figure 14: fond et couvercle

Comme on peut le constater, les ouvertures sont assez grandes pour accommoder le circuit ainsi qu'être solide. Chaque pièce a une épaisseur de 1cm et entre chaque trou également est une distance d'environ 1cm, ce qui rend la structure très solide. Pour tenir toutes les pièces ensemble, les murs seront collés et le couvercle sont vissé dans les murs.

2 - le circuit

Pour le circuit, ceci était assez simple. Nous avons 4 connexions à faire entre notre microcontrôleur (arduino MKR 1010 wifi) avec notre capteur (Adafruit BME 680). Les connexions sont les suivantes.

Table 3 : Connection entre le micro contrôleur et le capteur

| Arduino MKR 1010 wifi | Adafruit BME 680 |
|-----------------------|------------------|
| 5V | vin |
| GND | GND |
| 12 (SCL) | SCK |
| 11(SDA) | SDI |

Voici ce que le poursuit du circuit doit ressembler

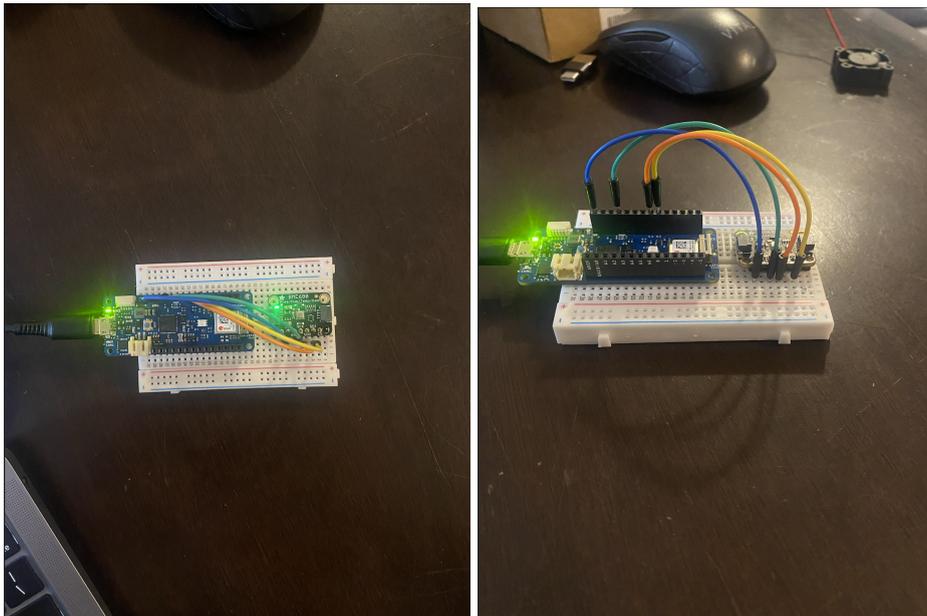


Figure 15: Circuit complet

3 - L'application

L'application était déjà préparée et installée. Celle-ci est appelée Blink qui est une application facile à utiliser. Cette application va permettre au client de voir les données sur leur appareil électronique 24/7. Ainsi, les données vont toutes être capturées et stockées dans le blynk.cloud. Pour que ce test soit fonctionnel, nous avons besoin du code, un tableau de bord complet avec toutes les composantes qu'on veut capturer et le **hotspot** pour avoir les

données sur le téléphone. Donc en premier, voici le code:

```
#define BLYNK_TEMPLATE_ID "TMPL4YzQ1FrP"
#define BLYNK_TEMPLATE_NAME "FB33"
#define BLYNK_AUTH_TOKEN "P7pYbn3vtAofqlkf6Y9XxvRzbSNrpW5Z"
#define BLYNK_PRINT Serial
#include <WiFiNINA.h>
#include <BlynkSimpleWiFiNINA.h>

char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "CBY-Makerspace"; // type your wifi name
char pass[] = "W2uOttawa!"; // type your wifi password

BlynkTimer timer;
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <SPI.h>
#include <Adafruit_BME680.h>

#define BME_SCK (13)
#define BME_MISO (12)
#define BME_MOSI (11)
#define BME_CS (10)

Adafruit_BME680 bme; // I2C
//Adafruit_BMP680 bmp(BMP_CS); // hardware SPI
//Adafruit_BMP680 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);

void setup() {
  Blynk.begin(auth, ssid, pass);
  timer.setInterval(100L, sendSensor);
  Serial.begin(9600);
```

```

while ( !Serial ) delay(100); // wait for native usb
Serial.println(F("BME680test"));
unsigned status;
//status = bmp.begin(BMP280_ADDRESS_ALT, BMP280_CHIPID);
status = bme.begin(0x77);
if (!status) {
    Serial.println(F("Could not find a valid BMP680 sensor, check wiring or "
        "try a different address!"));
    while (1) delay(10);
}
}

void sendSensor()
{
    float temperature = bme.readTemperature();
    float pressure = bme.readPressure()/100.0F; //convert Pa to hPa
    float gas = bme.gas_resistance ;
    float humidity = bme.readHumidity();

    //print sensor data to serial monitor
    Serial.print(F("Temperature = "));
    Serial.print(temperature);
    Serial.println(" *C");

    Serial.print(F("Humidity = "));
    Serial.print(humidity);
    Serial.println(" %");

    Serial.print(F("Pressure = "));
    Serial.print(pressure);
    Serial.println(" Pa");

    Serial.print("Gas = ");
    Serial.print(gas);

```

```

Serial.println(" Ohms");
Serial.println();

Blynk.virtualWrite(V0, temperature);
Blynk.virtualWrite(V1, humidity);
Blynk.virtualWrite(V2, pressure);
Blynk.virtualWrite(V3, gas); //The "1019.66" is the pressure(hPa) at sea
level in day in your region
delay(2000);
}
void loop()
{
Blynk.run();
timer.run();
}

```

Figure 16: Code

Pour que le code soit fonctionnel et transmet nos données, nous avons besoin de quelques éléments que nous devons spécifier. Le premier élément à spécifier est notre ID de modèle, nom de périphérique et jeton d'authentification. Ceci permet de transférer les données à notre application blynk

```

#define BLYNK_TEMPLATE_ID "TMPL4YzQ1FrP"
#define BLYNK_TEMPLATE_NAME "FB33"
#define BLYNK_AUTH_TOKEN "P7pYbn3vtAofqlkf6Y9XxvRzbSNrpW5Z"

```

Ensuite nous avons besoin d'entrer notre nom de wifi et le mot de passe pour que nos données soient transférées par wifi. De plus, notre microcontrôleur (arduino MKR 1010 wifi) fonctionne par wifi donc nous avons besoin du wifi pour qu'il fonctionne.

```

char ssid[] = "CBY-Makerspace"; // type your wifi name
char pass[] = "W2uOttawa!"; // type your wifi password

```

Figure 17: option du WiFi

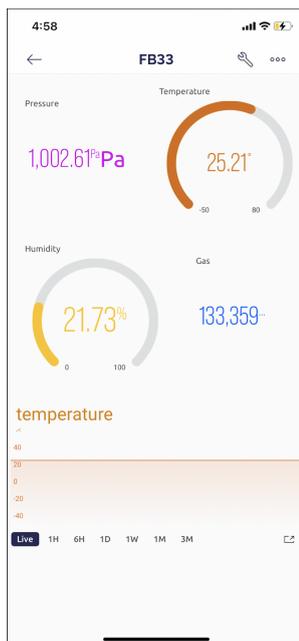
Avec le code, les données sont envoyées à notre centre de donnée (blynk.cloud)



Figure 18: Lecture des données

On peut voir que toutes les données capture dans le arduino IDE sont transférées à notre centre de données blynk. Enfin, la dernière partie de ce test est de confirmer que les données sont transférées à notre application. Les images suivantes prouvent que les données sont toutes envoyées à l'application. (l'application peut être téléchargée par tous les membres auxquels le client leur donne accès (avec le hotspot)).

(application blynk)



Pour l'application , il y a 3 composantes a ce qu'il soit fonctionnelle.

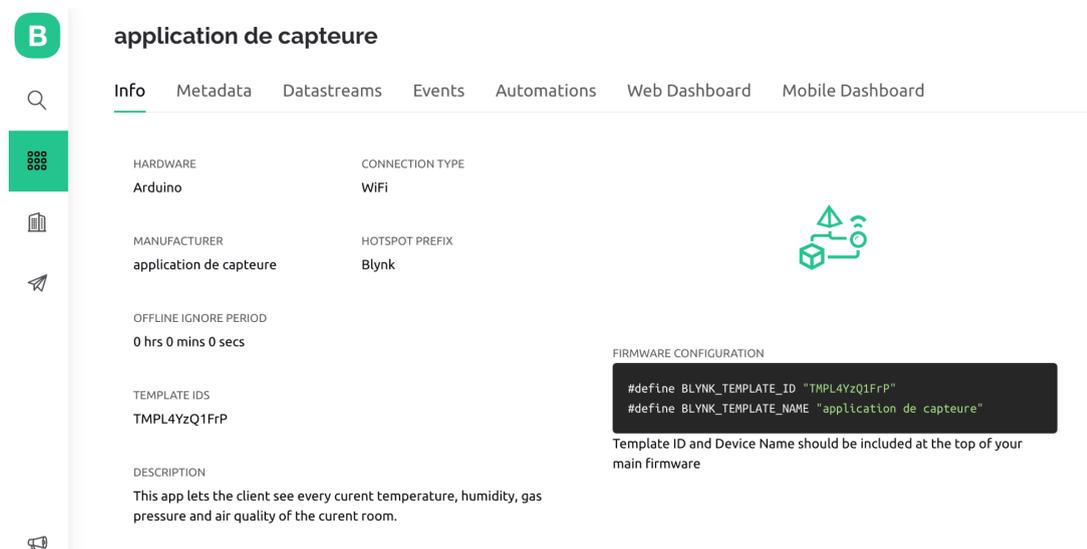
1. Blynk.Console : application Web où vous pouvez configurer, connecter, superviser vos appareils, analyser les données des capteurs, mettre à jour le micrologiciel OTA et gérer la manière dont les autres utilisateurs et organisations accèdent à leurs appareils.
2. Blynk.Apps : applications mobiles pour iOS et Android où vous pouvez créer une interface utilisateur pour vos appareils sans codage et la partager avec d'autres utilisateurs.

3. Blynk.Cloud : serveur qui envoie en toute sécurité des données entre vos appareils et applications.

1. Centre de donnée

Nous avons choisi Blynk comme application car nous croyons qu'il était très simple et efficace. Cette application va afficher toutes les données nécessaires d'après les demandes du client ainsi qu'il peuvent customiser certains aspects comme la marge des données à leur préférence. D'abord, la première composante que nous avons créée est le Blynk.Console. Ce que ceci fait est qu'avec les captures et le code, les données sont envoyées à cette place et ensuite, il serait dispersé à tous les membres qui ont l'application (essentiellement le centre de données). Donc voici tout ce qui appartient à Blynk.Console pour qu'il soit fonctionnelle.

1.1



The screenshot shows the Blynk console interface for an application named "application de capteur". The page is divided into several sections:

- Hardware:** Arduino
- Connection Type:** WiFi
- Manufacturer:** application de capteur
- Hotspot Prefix:** Blynk
- Offline Ignore Period:** 0 hrs 0 mins 0 secs
- Template ID:** TMPL4YzQ1FrP
- Description:** This app lets the client see every current temperature, humidity, gas pressure and air quality of the current room.
- Firmware Configuration:** A code block showing the template ID and name:

```
#define BLYNK_TEMPLATE_ID "TMPL4YzQ1FrP"
#define BLYNK_TEMPLATE_NAME "application de capteur"
```

Figure 19: page Info

Ceci est la base de l'application. Nous avons d'abord inscrit notre carte arduino que nous avons choisie (Arduino MKR WiFi 1010) et ensuite indiqué notre connexion type - Wifi (comment que nos données seront transférées). De plus, il y a un bout de code qui serait utile pour nous dans la prochaine partie.

1.2

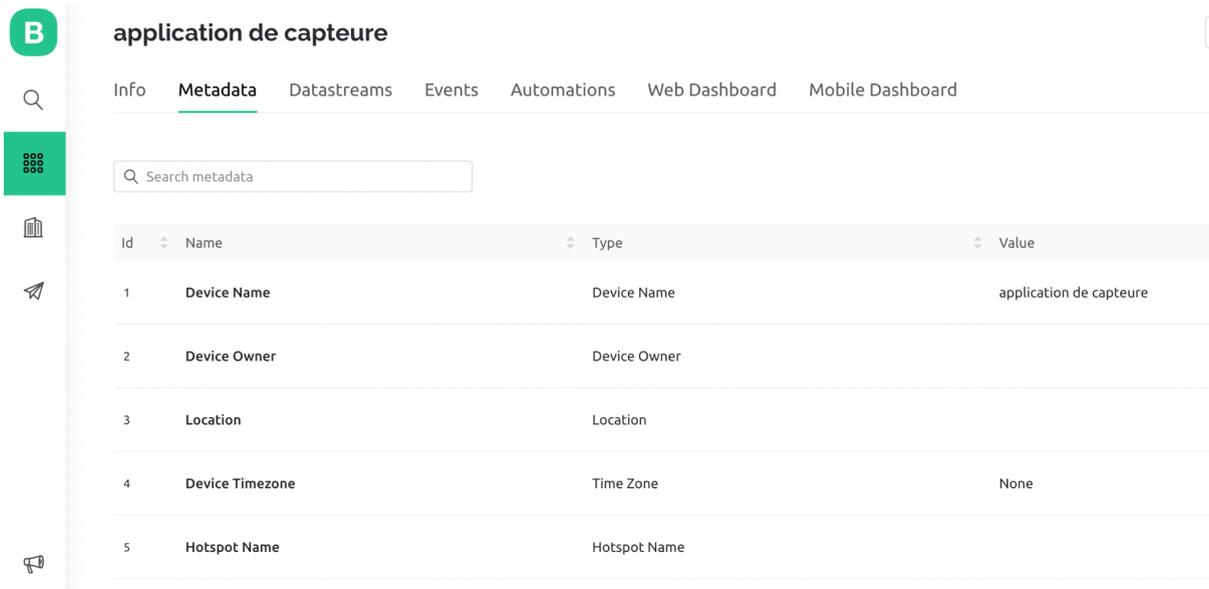


Figure 20: page Metadata

Le metadata est essentiel pour le client. Cette partie n'a pas de contribution a les donner mais permet à l'utilisation de mettre son nom, location, fuseau horaire et un hotspot name pour que d'autre personne puisse s'inscrire également.

1.3

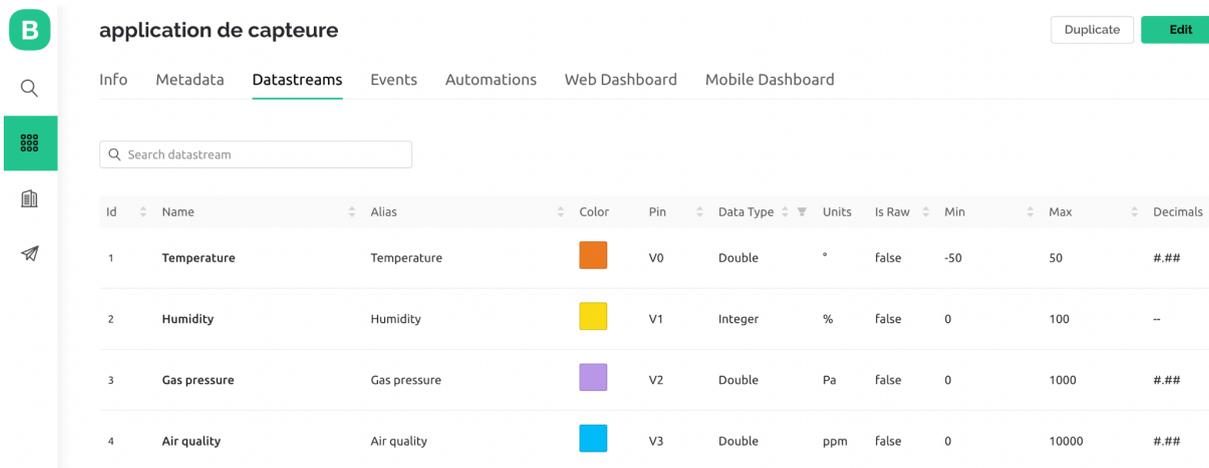


Figure 21: page Datastream

Le Datastream est toutes les données que nous allons capturer. Nous avons d'abord inscrit chacun de leur pin (peut changer avec notre arduino - il va juste falloir le changer ici pour qui puisse afficher la donnée). Ensuite nous avons mis le data type (ex. double affiche 2 chiffres après la virgule). Les unités de mesure sont les unités que chaque donnée occupe. Enfin, nous avons mis une marge de base pour chaque donnée qui peut être changée par l'utilisateur.

(par exemple, la température est entre -50 à 50 en ce moment mais le client pour le changer a 0 et 40)

1.4

The screenshot shows the 'Events' page of the 'application de capteur'. The interface includes a search bar for events and a table with the following data:

| Id | Name | Code | Color | Type | Description |
|----|--------------|---------|-------|---------|-------------|
| 1 | Online | online | Green | Online | |
| 2 | Offline | offline | Red | Offline | |
| 3 | OTA Update | sys_ota | Green | Info | |
| 4 | Debug | sys_dbg | Green | Info | |
| 5 | Notification | notify | Green | Info | |

Figure 22: page Events

Pour les événements, une des demandes du client était de pouvoir recevoir des notifications lorsqu'une donnée dépasse leur marge. à id 5, ceci va leur envoyer une notification. Il y a d'autres événements d'extra par exemple, si leur appareil est en ligne ou pas, quand il y a une mise à jour etc.

1.5

The screenshot shows the 'Automations' page of the 'application de capteur'. The interface includes a search bar for datastreams and a table with the following data:

| Name | Pin | Data Type | Type Of Automation | Condition | Action |
|--------------|-----|-----------|--------------------|--------------------------|--------------------------|
| Temperature | V0 | Double | Sensor | <input type="checkbox"/> | <input type="checkbox"/> |
| Humidity | V1 | Integer | Switch | <input type="checkbox"/> | <input type="checkbox"/> |
| Gas pressure | V2 | Double | Sensor | <input type="checkbox"/> | <input type="checkbox"/> |
| Air quality | V3 | Double | Sensor | <input type="checkbox"/> | <input type="checkbox"/> |

Figure 23: page Automations

Cette partie est similaire à la partie 1.3. Cette fois, nous avons inscrit le type d'automatisation pour chaque capture.

1.6

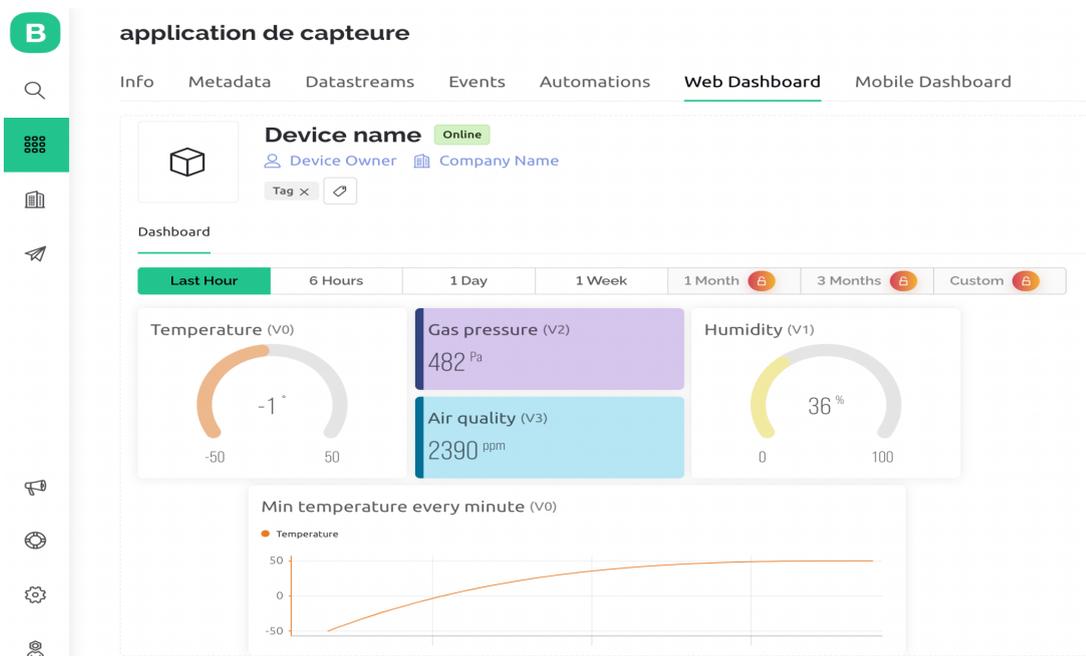


Figure 24: page Web Dashboard

Ceci est le centre de données où toutes les données recueillies de la salle seront affichées. Pour le design, nous avons été très simples mais avec une esthétique assez agréable à la vue (des couleurs et pas tout le même style d'étiquette). Au haut de la page, le *device name*, *device owner* et *company name* vas déjà être indiqué car il a été déclaré à la partie 1.2. De plus, cette application va afficher les résultats de la dernière heure, 6 heures, journée et semaine. Enfin, il faut mentionner que nous avons pas encore reçus tous nos capteurs, arduino etc donc c'est donnée ne sont pas précise mais, ceci est ce que notre centre de donnée final vas être.

3.5 Quitter le système

Prototype physique

Pour ranger correctement le système, suivez les étapes ci-dessous :

- Éteignez le capteur de température en débranchant l'alimentation ou en appuyant sur l'interrupteur.
- Débranchez tous les câbles connectés au capteur de température.

- Rangez le capteur de température dans un endroit sec et sûr, loin de la poussière et des débris.

Il est important de ranger correctement le système pour éviter tout dommage ou dysfonctionnement.

Prototype logiciel

Pour quitter le système et éteindre le capteur de température, suivez les étapes ci-dessous :

- Ouvrez l'application Blynk sur votre appareil mobile.
- Appuyez sur le bouton "Déconnexion" en haut de l'écran.
- Attendez que la connexion soit interrompue.
- Éteignez le capteur de température en débranchant l'alimentation ou en appuyant sur l'interrupteur.

Il est important d'éteindre correctement le système pour éviter tout dommage ou dysfonctionnement. Ne débranchez pas l'alimentation du capteur de température sans l'avoir éteint préalablement via l'application Blynk.

4 Utiliser le système

4.1 Fonction/Caractéristique donnée

1. Connexion au système

Pour utiliser le système, vous devez d'abord vous connecter à l'application Blynk. Suivez les étapes ci-dessous pour vous connecter :

- Téléchargez l'application Blynk sur votre smartphone ou votre tablette.
- Créez un nouveau compte ou connectez-vous à un compte existant.
- Appuyez sur le bouton "+" pour ajouter un nouveau projet.
- Choisissez le type de connexion Wi-Fi ou **Ethernet**.

- Entrez les informations de connexion Wi-Fi ou Ethernet de votre système.
- Appuyez sur le bouton "Play" pour commencer la connexion.

2. Contrôle de la température

Le système vous permet de surveiller et de contrôler la température en temps réel. Suivez les étapes ci-dessous pour utiliser cette fonctionnalité :

- Ouvrez le projet dans l'application Blynk.
- Appuyez sur le widget "Température" pour afficher la température actuelle.
- Appuyez sur le widget "Réglage de la température" pour régler la température de consigne.
- Utilisez le curseur pour régler la température de consigne.
- Appuyez sur le bouton "OK" pour confirmer la température de consigne.

3. Surveillance de l'état du système

Le système vous permet de surveiller l'état du système en temps réel. Suivez les étapes ci-dessous pour utiliser cette fonctionnalité :

- Ouvrez le projet dans l'application Blynk.
- Appuyez sur le widget "État du système" pour afficher l'état actuel du système.
- Si l'état est "en marche", le système est opérationnel.
- Si l'état est "arrêté", le système n'est pas opérationnel.

4. Gestion des alarmes

Le système vous permet de configurer des alarmes pour vous avertir lorsque la température dépasse une certaine limite. Suivez les étapes ci-dessous pour utiliser cette fonctionnalité :

- Ouvrez le projet dans l'application Blynk.
- Appuyez sur le widget "Réglage d'alarme" pour configurer une alarme.
- Utilisez les curseurs pour définir la température maximale et minimale pour l'alarme.
- Appuyez sur le bouton "OK" pour confirmer la configuration de l'alarme.

5. Arrêt du système

Lorsque vous avez fini d'utiliser le système, vous devez l'arrêter correctement pour éviter toute corruption de données. Suivez les étapes ci-dessous pour arrêter le système :

- Appuyez sur le bouton "Stop" dans l'application Blynk pour arrêter la connexion au système.
- Débranchez le système de l'alimentation électrique.
- Rangez le système dans un endroit sûr et sec.

5 Dépannage & assistance

1. Vérifiez la connexion physique de votre système
 - Assurez-vous que tous les câbles sont correctement branchés et qu'il n'y a pas de connexions lâches.
 - Vérifiez également que le capteur est correctement installé et qu'il est en contact avec l'environnement que vous souhaitez surveiller.
2. Vérifiez les configurations de votre système
 - Assurez-vous que toutes les configurations de votre système sont correctement définies et mises à jour. Cela peut inclure les configurations de l'application Blynk et les paramètres de votre microcontrôleur Arduino MKR WiFi 1010.
 - Assurez-vous également que votre système est connecté à Internet et que vous avez une connexion stable.
3. Vérifiez les erreurs dans l'application Blynk
 - Ouvrez l'application Blynk et vérifiez si des erreurs sont signalées. Si oui, identifiez-les et essayez de les résoudre en suivant les instructions données par l'application.
4. Vérifiez les erreurs dans le microcontrôleur Arduino MKR WiFi 1010
 - Connectez-vous à votre microcontrôleur Arduino MKR WiFi 1010 et vérifiez si des erreurs sont signalées. Si oui, identifiez-les et essayez de les résoudre en suivant les instructions données par le microcontrôleur.
5. Vérifiez les erreurs du capteur BME680
 - Si vous pensez que les erreurs sont liées au capteur BME680, vérifiez la documentation du fabricant pour voir les types d'erreurs possibles et comment les résoudre.
6. Réinitialisez le système
 - Si aucune des étapes précédentes n'a résolu le problème, vous pouvez essayer de réinitialiser complètement votre système. Cela peut impliquer de réinitialiser les configurations de l'application Blynk, les paramètres du microcontrôleur Arduino MKR WiFi 1010 et les configurations du capteur BME680.
7. Demandez de l'aide
 - Si vous ne pouvez pas résoudre le problème par vous-même, demandez de l'aide à un professionnel de l'informatique, un technicien qualifié ou à l'équipe de conception.

5.1 Messages ou comportements d'erreur

1. "Données de capteur manquantes" : Ce message peut s'afficher si le capteur BME680 ne communique pas les données de température, d'humidité ou de pression à l'application Blynk. Les causes possibles peuvent être un mauvais câblage, une défaillance du capteur ou du microcontrôleur, ou une interruption de la connexion Internet. Les actions correctives possibles sont de vérifier les connexions, de recalibrer les capteurs, de mettre à jour le micrologiciel, de vérifier la connexion Internet et de redémarrer le système.
2. "Données de capteur inexactes" : Ce message peut s'afficher si les données de température, d'humidité ou de pression semblent inexactes ou dépassent les limites prévues. Les causes possibles peuvent être une calibration incorrecte des capteurs, une température ambiante trop élevée ou trop basse, une humidité excessive ou une pression anormale. Les actions correctives possibles sont de recalibrer les capteurs, de régler la température ambiante, d'ajuster l'humidité ou la pression ou de remplacer les capteurs défectueux.
3. "Erreur de communication" : Ce message peut s'afficher si l'application Blynk ne parvient pas à communiquer avec le microcontrôleur ou le capteur BME680. Les causes possibles peuvent être une connexion Internet instable, une configuration incorrecte de l'application, un microcontrôleur défectueux ou un capteur endommagé. Les actions correctives possibles sont de vérifier la connexion Internet, de réinstaller l'application Blynk, de mettre à jour le micrologiciel ou de remplacer les composants défectueux.
4. "Microcontrôleur endommagé" : Ce message peut s'afficher si le microcontrôleur est endommagé ou en panne. Les causes possibles peuvent être une surchauffe, une surtension, une défaillance du circuit ou une mauvaise utilisation. Les actions correctives possibles sont de remplacer le microcontrôleur ou de contacter le support technique du fabricant pour obtenir une assistance supplémentaire.
5. Pièces susceptibles de se casser : Les pièces susceptibles de se casser dans ce système peuvent être les câbles de connexion, les broches du capteur ou du microcontrôleur, ou le capteur BME680 lui-même. Les causes possibles peuvent être une mauvaise manipulation, une utilisation abusive, une usure normale ou une défaillance du composant. Les actions correctives possibles sont de remplacer les pièces endommagées ou de contacter le support technique du fabricant pour obtenir une assistance supplémentaire.

5.2 Considérations spéciales

1. Si le capteur BME680 ne fonctionne pas correctement, il peut être nécessaire de le remplacer. Cependant, il est important de noter que le capteur BME680 est un composant délicat qui doit être manipulé avec précaution pour éviter toute rupture ou endommagement.

2. Si le microcontrôleur est endommagé, il peut être nécessaire de le remplacer. Toutefois, il est important de noter que le microcontrôleur est un composant délicat qui doit être manipulé avec précaution pour éviter toute rupture ou endommagement.
3. Si les messages d'erreur continuent de s'afficher malgré les actions correctives, il peut être nécessaire de contacter le support technique du fabricant pour obtenir une assistance supplémentaire.
4. Il est important de vérifier régulièrement les connexions et les mises à jour du micrologiciel pour s'assurer que le système fonctionne correctement.
5. Il est important de garder à l'esprit que les données collectées par le capteur BME680 peuvent être sensibles à certaines conditions environnementales. Par conséquent, il peut être nécessaire de prendre des mesures supplémentaires pour assurer des conditions environnementales stables et cohérentes.
6. Il est important de se rappeler que l'application Blynk est une application tierce et que des problèmes de compatibilité peuvent survenir avec certains microcontrôleurs ou capteurs. Il est donc important de vérifier la compatibilité des composants avant de les utiliser ensemble.

5.3 Entretien

1. Vérification régulière des connexions : Il est important de vérifier régulièrement les connexions du système pour s'assurer qu'elles sont bien connectées et qu'il n'y a pas de corrosion ou d'autres problèmes. Les connexions doivent être inspectées pour détecter les signes d'usure et de fatigue, et doivent être resserrées si nécessaire.
2. Nettoyage régulier : Il est important de nettoyer régulièrement les composants du système pour éviter l'accumulation de poussière, de saleté ou d'autres contaminants. Un nettoyage régulier peut aider à maintenir la précision des mesures de température, d'humidité et de pression.
3. Calibration régulière : Il est important de calibrer régulièrement le capteur BME680 pour maintenir l'exactitude des mesures. La calibration doit être effectuée conformément aux instructions du fabricant.
4. Surveillance des mises à jour du micrologiciel : Il est important de surveiller régulièrement les mises à jour du micrologiciel pour s'assurer que le système fonctionne avec la dernière version. Les mises à jour du micrologiciel peuvent inclure des correctifs de bugs, des améliorations de performance et de nouvelles fonctionnalités.
5. Remplacement des composants défectueux : Si un composant du système est défectueux, il doit être remplacé immédiatement pour éviter toute défaillance du système.
6. Environnement de stockage : Il est important de stocker le système dans un environnement sec et propre. Les températures extrêmes et l'humidité peuvent affecter le fonctionnement du système.

5.4 Assistance

Si l'utilisateur rencontre des problèmes avec le système de surveillance de la température, de l'humidité et de la pression équipé d'un capteur BME680 qui communique ses données à une application Blynk, il peut contacter le support technique du fabricant du capteur BME680 ou de l'application Blynk pour obtenir de l'aide.

Voici les coordonnées de contact pour le support technique du fabricant :

Nom du fabricant : Bosch Sensortec Adresse email : support@sensortec.bosch.com

Voici les coordonnées de contact pour le support technique de l'application Blynk :

Nom du fournisseur : Blynk Inc. Adresse email : support@blynk.cc

Lorsque l'utilisateur signale un problème avec le système, il doit fournir une description détaillée du problème et de ses symptômes. Il est recommandé de fournir également des captures d'écran si possible pour aider à identifier le problème.

En cas d'incident de sécurité, l'utilisateur doit immédiatement contacter le fabricant du capteur BME680 ou de l'application Blynk et suivre les instructions fournies pour signaler l'incident et prendre les mesures nécessaires pour protéger le système.

Il est important de noter que pour une assistance rapide et efficace, il est recommandé à l'utilisateur de conserver une copie des informations de contact et des instructions de dépannage à portée de main en tout temps.

6 Documentation du produit

6.1 Design

Pour le design de la boîte, il n'y avait pas de contraintes des clients autre tout contenir dans celle-ci et ne pas être plus grosse qu'une boîte de soulier. Le reste était libre à l'équipe de décider. Nous avons donc élaboré un premier concept. Les clients nous ont informés qu'il était trop ouvert et nous avons donc apporté les modifications nécessaires pour obtenir le produit final.

Nous avons choisi l'impression 3D et le PLA parce que c'est un matériau accessible, facile à travailler et pas trop cher. Nous étions conscients que l'impression 3D n'est pas la méthode la plus rapide et avons prévu de changer à la découpe laser si un problème majeur était survenu. Étant donné que les pièces ont pris beaucoup de temps avant d'arriver, nous avons eu assez de temps pour tout imprimer. Si vous étiez pressé dans le

temps, nous recommandons d'envisager l'alternative tel que la découpe laser ou autre méthode au choix.

Pour la grandeur de la boîte, nous nous sommes basés sur les dimensions du micro contrôleur Arduino MKR WiFi 1010 et du capteur de poussière que nous allons intégrer. L'arduino mesure 6 cm par 2.5 cm. Le capteur de poussière était environ 4 cm par 3 cm. Il fallait donc environ 10cm de long par 5 cm de large au minimum. Pour avoir une boîte pouvant accommoder pour des modifications et éviter que les différents capteurs s'influencent les uns les autres, la boîte mesure 15 cm par 20 cm par 15 cm, considérant ces mesures comme les mesures extérieures. Chaque épaisseur est de 1 cm, ce qui permet une bonne solidité. Le remplissage des pièces lors de l'impression était à 10% de infill, ce qui est amplement suffisant pour avoir des morceaux légers, mais solides également.

6.1.1 NDM (Nomenclature des Matériaux)

Table 4: NDM de la boîte

| Composante | Coût unitaire | Quantité | Coût étendue | Lien |
|----------------|--------------------|----------|--------------|---|
| Filament PLA | \$26,99 | 1 | 26,99\$ | https://www.amazon.ca/ERYON-E-Filament-1-75mm-Printing-Printer/dp/B07ZPT32M8/ref=sr_1_5?keywords=pla%2Bblack&qid=1676858036&sr=8-5&th=1 |
| Colle | \$11,54 | 1 | 11,31\$ | amazon.ca/gp/product/B07W47FCSR/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1 |
| Vis | environ 5\$/paquet | 1 | 3,99\$ | https://www.rona.ca/fr/produit/vis-a-tete-plate-precision-no-8-1-2-po-empreinte-carree-paquet-de-10-zinguee-820-660-63868214 |
| Blender | Gratuit | | Gratuit | https://www.blender.org/download/ |
| Ultimaker Cura | Gratuit | | Gratuit | https://ultimaker.com/fr/software/ultimaker-cura |
| Pince | 23,99\$ | 1 | 23,99\$ | https://www.amazon.ca/CRAFTSMAN-Linesman-Pliers-8-Inch-CMHT81648/dp/B07RCNHV61/ref=sr_1_6?crd=V3NCISM29WHY&keywords=pliers&qid=1681250075&srefix=pince%2Caps%2C94&sr=8-6&th=1 |

| | | | | |
|------------|--------|---|-----------------|---|
| Tourne vis | 7,79\$ | 1 | 7,79\$ | https://www.amazon.ca/Amazon-Brand-Multi-Bit-Racheting-Screwdriver/dp/B09G7J1X2N/ref=sr_1_8?crid=YUDUBBCFQUW2&keywords=screwdriver&qid=1681250335&srefix=tournevis%2Caps%2C100&sr=8-8 |
| Coût total | | | 74,07\$ + taxes | |

6.1.2 Liste d'équipements

Table 5. Liste d'équipement pour la boîte

| |
|---|
| Équipement |
| Ordinateur |
| Imprimante 3D avec une plateforme d'impression d'au moins 200mm x 200mm |
| Serres en C |

6.1.3 Instructions

Étape 1. Télécharger le logiciel Blender (ou autre logiciel de modélisation 3D).
<https://www.blender.org/download/>

Étape 2. Télécharger le logiciel UltiMaker Cura.
<https://ultimaker.com/fr/software/ultimaker-cura>

Étape 3. Ouvrir Blender.

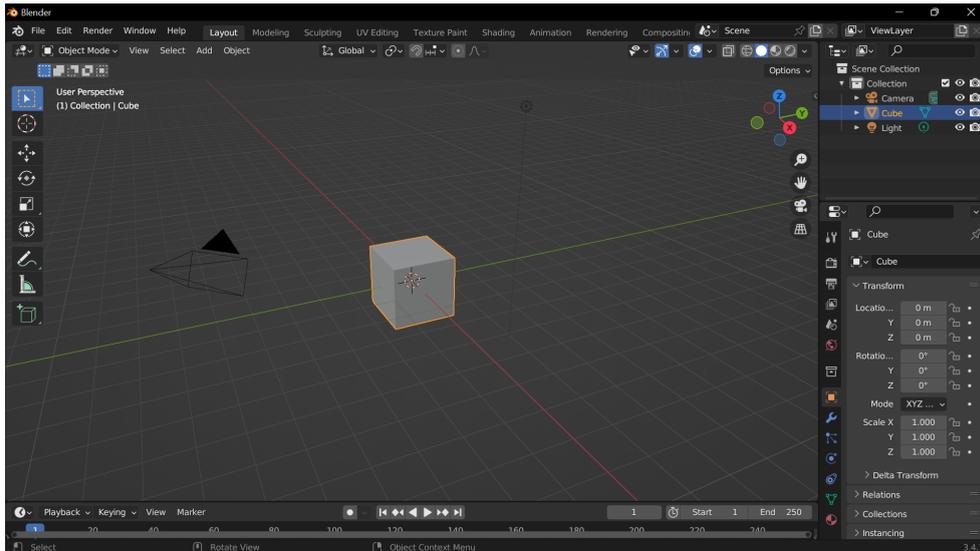


Figure 25: Capture d'écran du menu du logiciel Blender

Étape 4. Importer le projet dans Blender en allant sur File > Import > Stl (.stl)

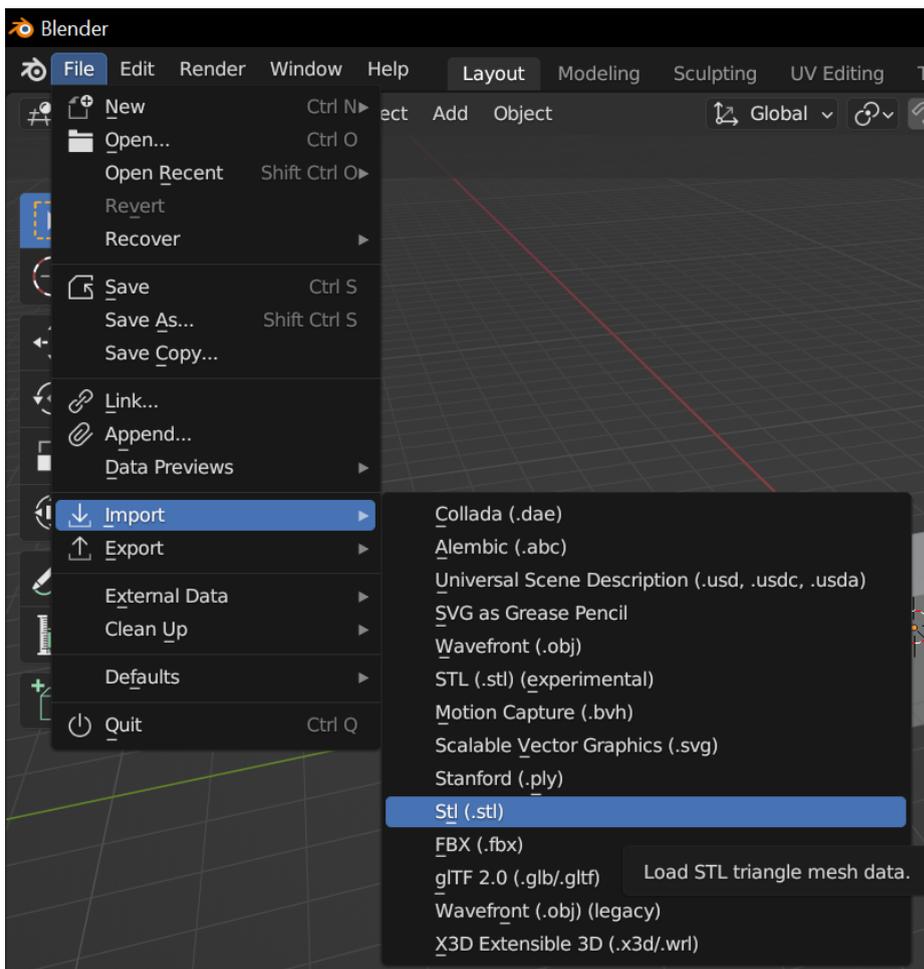


Figure 26: Capture d'écran de l'importation de fichiers STL

Étape 5. Vous devriez vous trouver devant le modèle de la boîte. Sur la droite, chaque pièce est séparée dans une collection différente de façon à les retrouver facilement. Vous pouvez apporter les modifications que vous désirez.

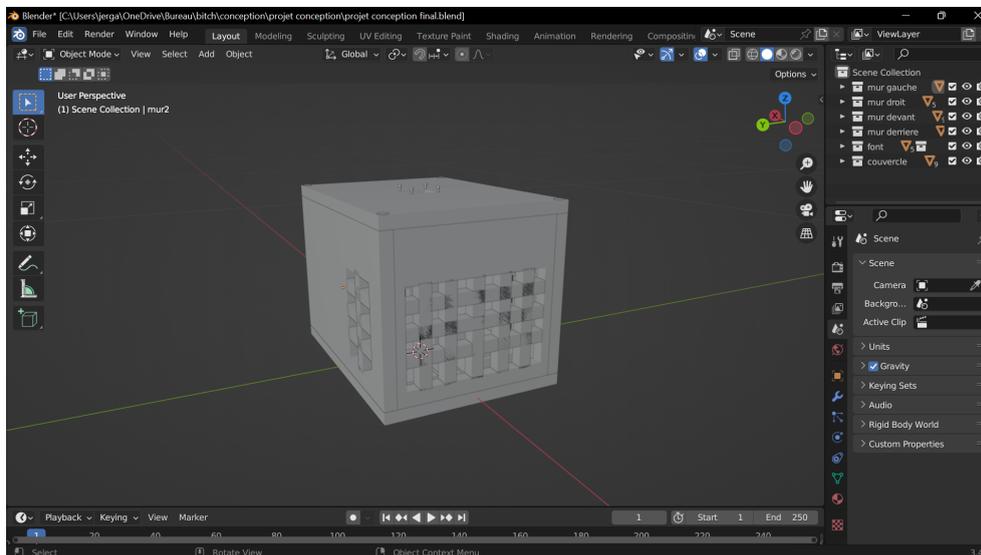


Figure 27: Capture d'écran de la boîte

Étape 6. Pour exporter les pièces, copier le morceau désiré dans un nouveau projet pour l'isoler et exporter en allant sur file > Export > Stl (.stl). Cette étape vous permettra d'avoir un fichier que le logiciel Cura pourra lire. Voici un exemple avec le couvercle:

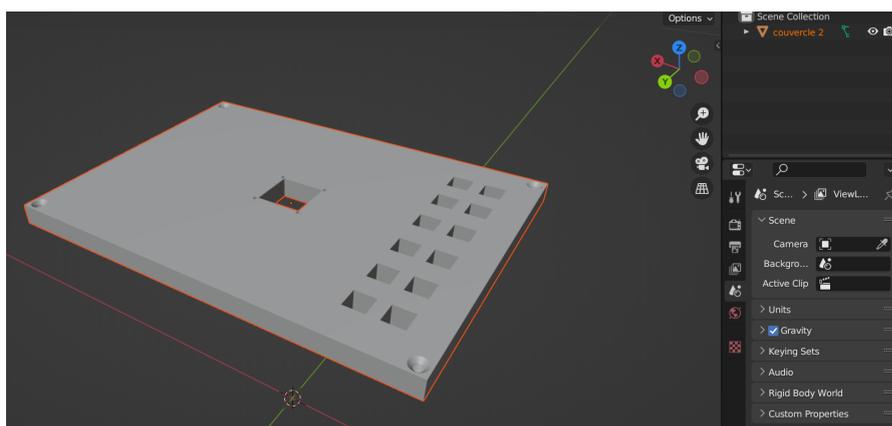


Figure 28: Capture d'écran du couvercle

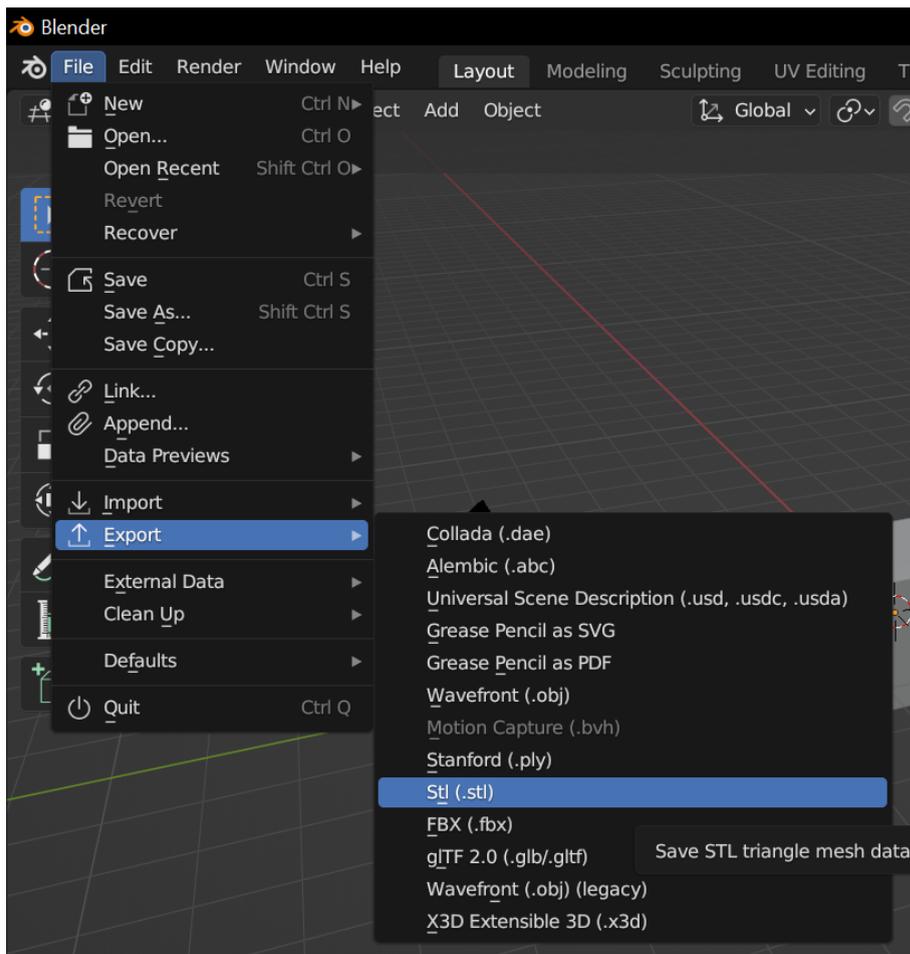


Figure 29: Capture d'écran de l'exportation de fichier STL

Étape 7. Ouvrir la pièce sur le logiciel UltiMaker Cura.

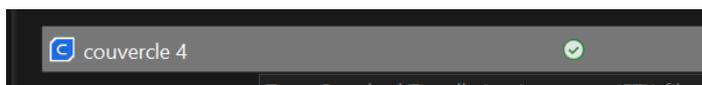


Figure 30: Capture d'écran d'un fichier STL

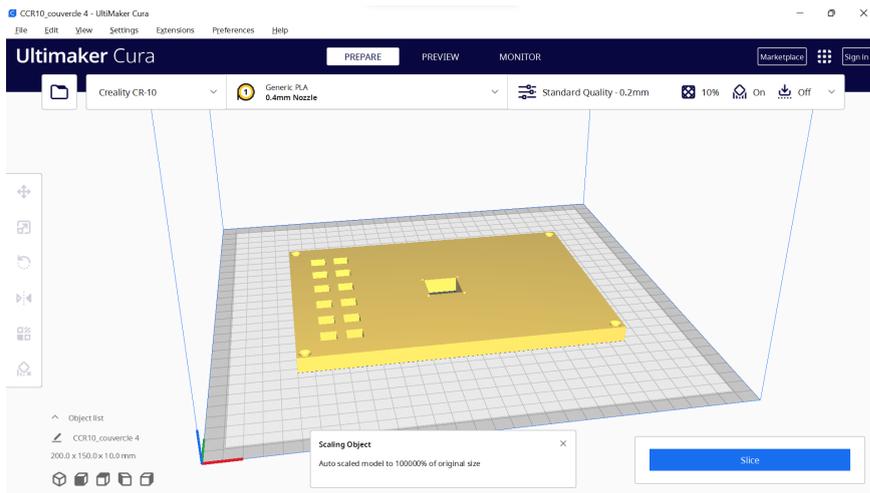


Figure 31: Capture d'écran du menu du logiciel UltiMaker Cura

Étape 8. Si vous ouvrez le logiciel pour la première fois, vous devrez configurer votre plateforme d'impression selon l'imprimante 3D que vous utiliserez. En haut de l'écran à gauche, allez sur Printer > Add printer > Add a non-networked printer > trouvez votre imprimante selon le modèle et manufacturier.

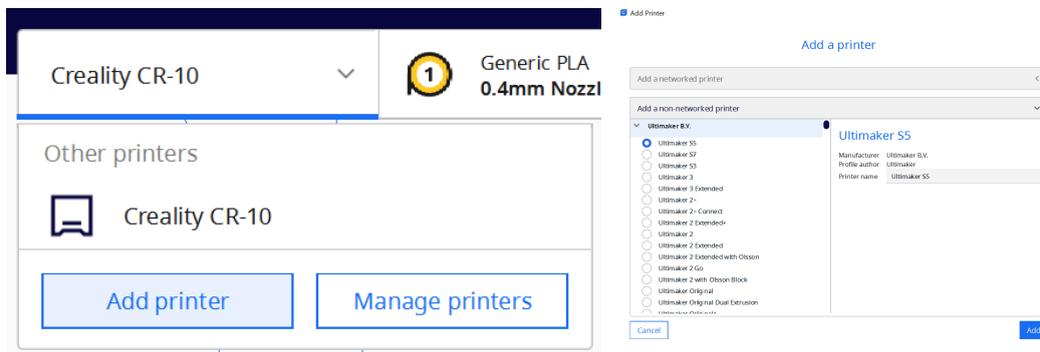


Figure 32: Capture d'écran du menu du logiciel UltiMaker Cura

Étape 9. Vous devriez être maintenant prêt à **découper** le modèle. En cliquant sur le fichier à gauche, importer les pièces.



Figure 33: Capture d'écran de l'importation d'un fichier

Étape 10. Ouvrir le menu de contrôle en haut à gauche et ajuster les options au besoin.

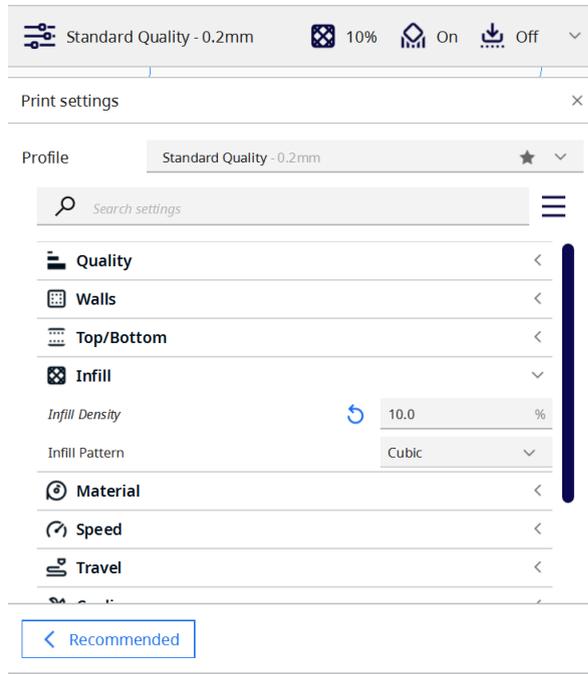


Figure 34: Capture d'écran des paramètres d'impression

Étape 11. En bas à droite, cliquer sur Slice pour découper le modèle.



Figure 35: Capture d'écran du bouton découpage

Étape 12. Si le découpage est convenable, sauvegarder le .gcode sur la carte SD de votre imprimante. Sinon, refaire les étapes 8 et 9. Sur l'option Preview en haut de l'écran, vous pouvez voir toutes les actions de l'imprimante. En bas à droite, vous pouvez voir le temps d'impression et la quantité de filament utilisée.

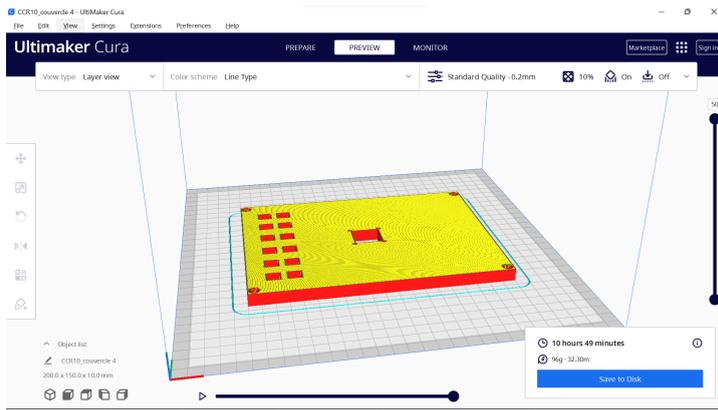


Figure 36: Capture d'écran d'une pièce découpée

Étape 13. Insérer la carte SD dans l'imprimante et démarrer l'impression.

Étape 14. Répéter pour chaque pièce.

Étape 15. Si vous le désirez, vous pouvez peindre les pièces comme vous le souhaitez. Cette étape est optionnelle.

Étape 16. Une fois toutes les pièces imprimées, coller les pièces formant les murs au fond de la boîte. Il est recommandé d'utiliser les serres en C pour garder les pièces en place et s'assurer que les murs sont droits.

Étape 17. Visser les quatre vis aux coins des murs pour fermer la boîte.

6.2 Circuit

Le circuit est très simple. Après avoir fait des recherches en ligne, nous avons découvert que certains capteurs n'ont pas les mêmes nomenclatures pour les pins. Nous avons donc dû trouver une conversion pour pouvoir brancher le capteur.

Au départ, nous allions avoir le capteur de température Adafruit BME 680 et un capteur de poussière. Ce dernier a cependant pris trop de temps à arriver entre nos mains et nous avons choisi de ne pas l'inclure et de se concentrer sur le BME 680 pour avoir un produit fonctionnel. Cela étant dit, comme mentionné précédemment, la boîte est assez grande pour incorporer d'autres capteurs, le capteur de poussière y compris. Les détails pour ce capteur ne seront pas documentés puisqu'il n'a pas été utilisé dans ce projet.

Également, l'option d'une carte non WiFi reste toujours possible. Il faut toutefois ajouter un capteur pouvant relier la carte par WiFi vers le système Blynk. Pour notre équipe, il était plus simple et facile d'utiliser une carte WiFi, comme la Arduino MKR WiFi 1010.

Table 6. Fonction des pins du capteurs

| | | | |
|-----|-------------------|-----|-------------------------------|
| VCC | Powers the sensor | SDA | SDA pin for I2C communication |
| | | | SDI (MOSI) pin for SPI |

| | | | |
|-----|--|-----|---------------------------------------|
| | | | communication |
| GND | Common GND | SDO | SDO (MISO) pin for SPI |
| SCL | SCL pin I2C communication SCK pin for SPI communication | CS | Chip select pin for SPI communication |

6.2.1 NDM (Nomenclature des Matériaux)

Table 7: NDM du circuit

| Composante | Unité de mesure | Quantité | Coût étendue | Lien |
|---|-----------------|----------|------------------|---|
| Capteur de température, humidité, pression et gaz | °C, pa, % | 1 | \$18.95 | https://www.adafruit.com/product/3660?hidden=yes&main_page=product_info&part_id=3660&clid=Cj0KCQiAxbefBhDfARIsAL4XLRqpyoaU59CZXB39OPLDSKLSJvJ57MAlmoyRS8i4MD9X75ZXBZqi9RUaAh-YEALw_wcB |
| Breadboard kit (extra(fils électrique)) | N/A | 1 | \$20.99 | https://www.amazon.ca/gp/product/B0BMFXPSVG/ref=ox_sc_act_title_3?smid=A3JRR1O4XCME47&pssc=1 |
| Carte Arduino MKR WiFi 1010 | N/A | 1 | \$47.39 | https://store.arduino.cc/products/arduino-mkr-wifi-1010#looxReviews |
| Câble d'alimentation | m | 1 | \$6.90 | https://store-usa.arduino.cc/products/usb-cable-type-a-male-to-micro-type-b-male |
| Ventilateur | V | 1 | \$13,99 | https://www.amazon.ca/GDSTIME-Dupont-Connector-Brushless-Cooling/dp/B07SWXLZHS/ref=sr_1_3?crd=KE5SON7Q9JI0&keywords=small%2Bfan%2Bfor%2Barduino&qid=1676829401&s=industrial&srefix=small%2Bfan%2Bfor%2Barduino%2Cindustrial%2C69&sr=1-3&th=1 |
| Coût total du sous-systèmes | | | 108,22\$ + taxes | |

6.2.2 Liste d'équipements

Table 8. Liste d'équipement pour le circuit

| |
|--------------------------|
| Équipement |
| Ordinateur |
| Ensemble de fer à souder |

6.2.3 Instructions

Étape 1. Acquérir les matériaux nécessaire (Arduino MKR Wifi 1010, ensemble breadboard et fils, capteur Adafruit BME 680)

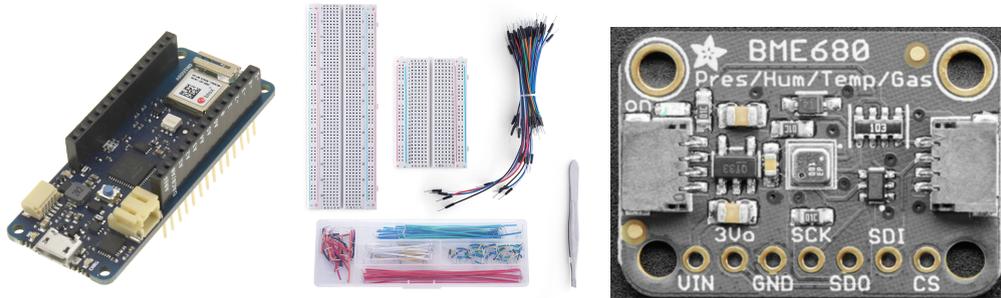


Figure 37: Images du micro contrôleur Arduino MKR WiFi 1010, ensemble breadboard et du capteur Adafruit BME 680

Étape 2. Élaborer le circuit suivant:

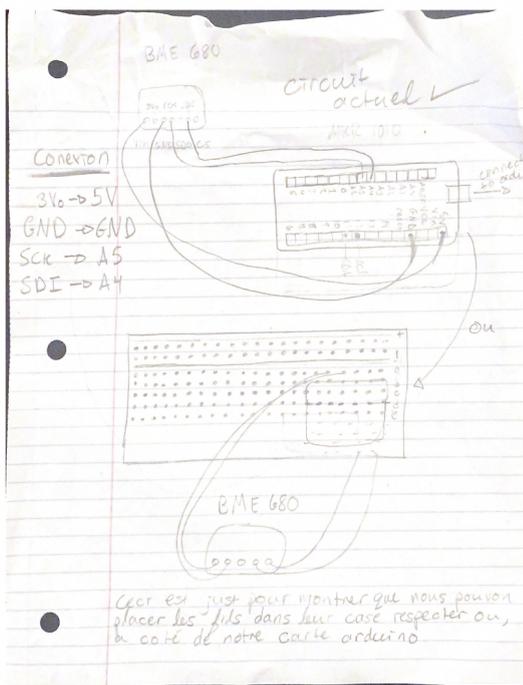


Figure Figure 38: Schéma du circuit

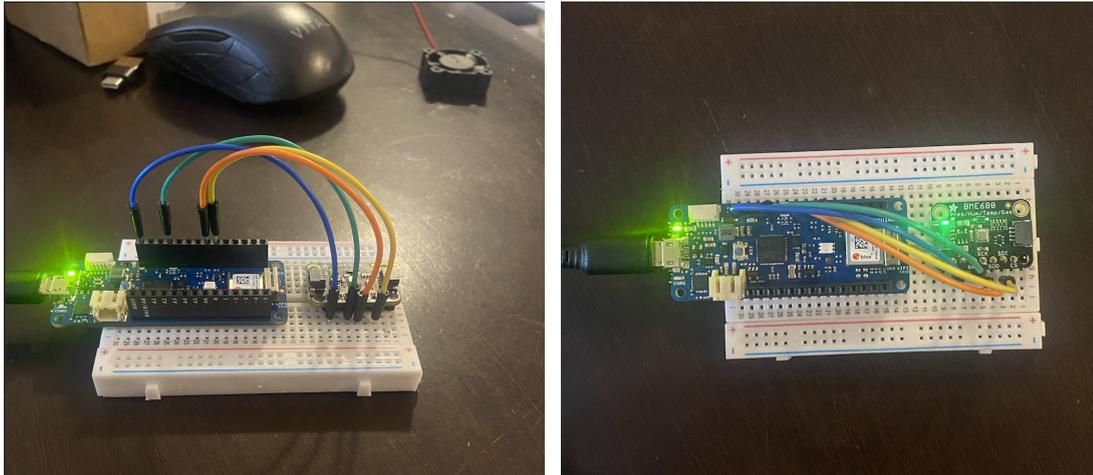


Figure 39: Photos du circuit

Étape 3. Souder les fils aux capteurs BME 680.

Étape 3. Ajouter le ventilateur . Le fil noir connecte au ground (GND) et le fil rouge au 5V.

Étape 4. Brancher le micro contrôleur MKR WiFi 1010 avec le fil micro USB à un port USB sur un ordinateur. Si la lumière sur le micro contrôleur et le capteur BME 680 allument, le circuit est bien branché et fonctionne.

6.3 Codage

Nous avons choisi l'application Blynk pour afficher les données puisqu'elle est très facile à installer, à configurer et à utiliser. Il est également possible de donner accès à plusieurs membres d'une équipe pour que tous y aient accès.

Le code a été bâti à partir de la **librairie** Adafruit B680 test et nous avons ajouté tous les éléments manquants pour que le capteur et l'application fonctionnent ensemble. Toutes les ressources utilisées sont accessibles en ligne puisque Arduino est une plateforme open source disponible à tous. Le code peut être modifié et amélioré au besoin des utilisateurs.

6.3.1 NDM (Nomenclature des Matériaux)

Table 9: NDM du logiciel

| Composante | Quantité | Coût étendue | Lien |
|-----------------------------|----------|---|---|
| Carte Arduino MKR WiFi 1010 | 1 | Déjà calculés au sous système précédent | https://store.arduino.cc/products/arduino-mkr-wifi-1010#looxReviews |
| Câble d'alimentation | 1 | | https://store-usa.arduino.cc/products/usb-cable-type-a-male-to-micro-type-b-male |
| Arduino IDE | | Gratuit | https://www.arduino.cc/en/software |

| | | | |
|--------------------------------|--|----------|---|
| Librairie WiFiNINA | | Gratuite | Trouver directement sur Arduino IDE dans la section Library |
| Librairie BlynkSimpleWiFi NINA | | Gratuite | Trouver directement sur Arduino IDE dans la section Library |
| Librairie Wire | | Gratuite | https://github.com/esp8266/Arduino/blob/master/libraries/Wire/Wire.h |
| Librairie Adafruit_sensor | | Gratuite | https://github.com/adafruit/Adafruit_Sensor |
| Librairie SPI | | Gratuite | https://github.com/arduino/ArduinoCore-avr/blob/master/libraries/SPI/src/SPI.h |
| Librairie Adafruit_BME680 | | Gratuite | Trouver directement sur Arduino IDE dans la section Library |
| Coût total du sous-systèmes | | Gratuit | |

6.3.2 Liste d'équipements

Table 10: Liste d'équipement pour le codage

| Équipement |
|------------|
| Ordinateur |

6.3.3 Instructions

Étape 1. Télécharger la version appropriée d'Arduino IDE

Downloads

Arduino IDE 2.0.4

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE
The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** Intel, 10.14: "Mojave" or newer, 64 bits
- macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Figure 40: Capture d'écran du téléchargement d'Arduino IDE

Étape 2. Ouvrir Arduino IDE et télécharger les bibliothèques mentionnées dans le tableau précédent.

Étape 3. Télécharger l'application Blynk sur un téléphone intelligent et/ou ouvrir un compte sur le site web: <https://blynk.io/>

Étape 4. Créer un nouveau projet sur Blynk en cliquant sur New Template

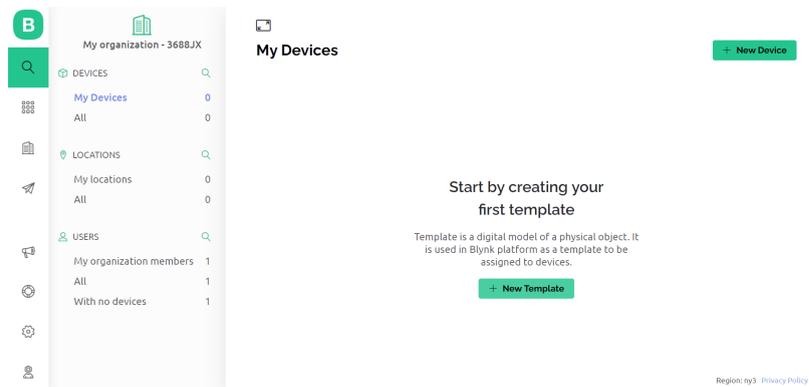


Figure 41: Capture d'écran du menu du logiciel Blynk

Étape 5. Choisir le nom, le micro contrôleur utilisé et le type de connexion pour ce projet. Le nom est important pour le code. Une fois le template créé, vous pourrez continuer vers finalisation de la plateforme et le code.

Create New Template

NAME
Name
Field is required

HARDWARE Other CONNECTION TYPE WiFi

DESCRIPTION
This is my template
19 / 128

Cancel Done

Figure 42: Capture d'écran du menu de template 1

Étape 6. Ouvrir le template. Vous devriez vous trouver devant un écran similaire au suivant. Garder en tête la partie de code qui se trouve en bas à droite.

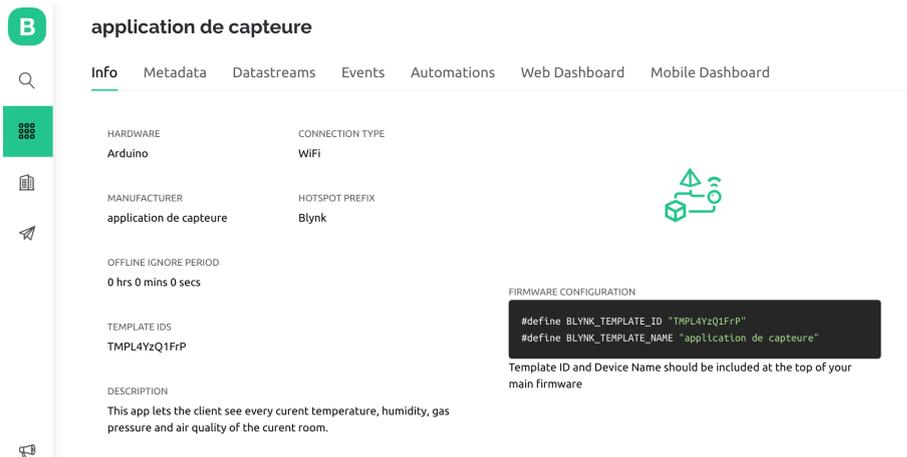


Figure 43: Capture d'écran de la page Info

Étape 7. Cliquer sur la page Metadata. Ici, vous pourrez configurer les éléments spécifiques à votre projet.

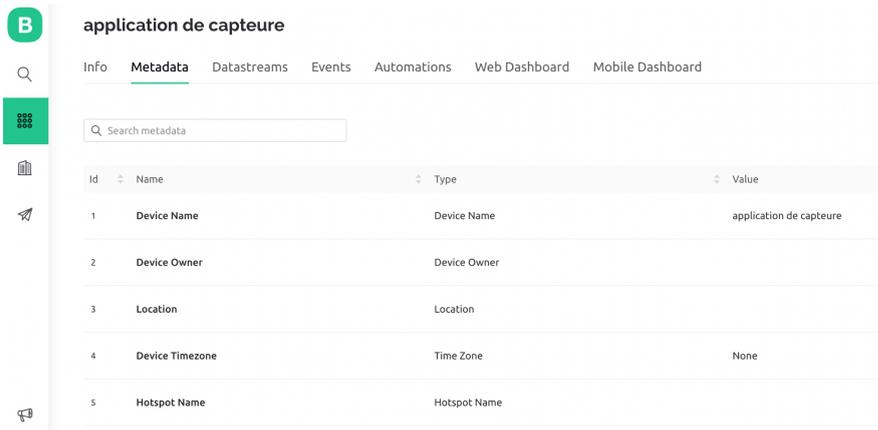


Figure 44: Capture d'écran de la page Metadata

Étape 8. Cliquer sur la page Datastreams. Ici, vous pourrez ajouter les éléments que vous voulez observer et toutes les variables concernant ceux-ci.

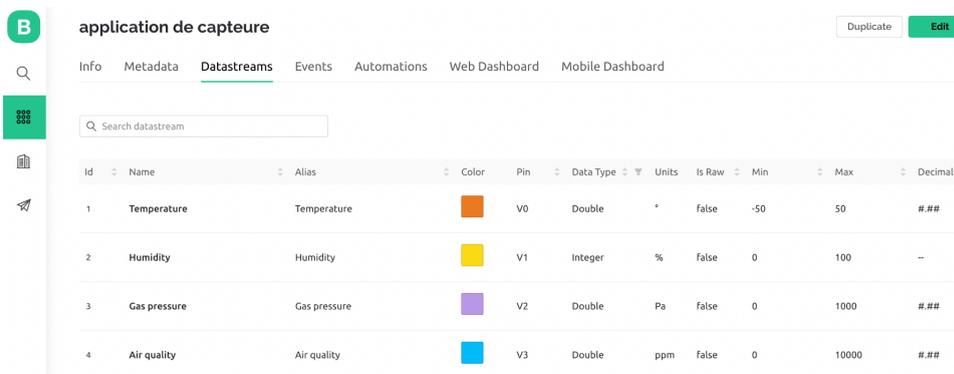


Figure 45: Capture d'écran de la page Datastreams

Étape 9. Cliquer sur la page Event. Ici, vous pourrez configurer les éléments d’alerte ou d’avertissement pour le système.

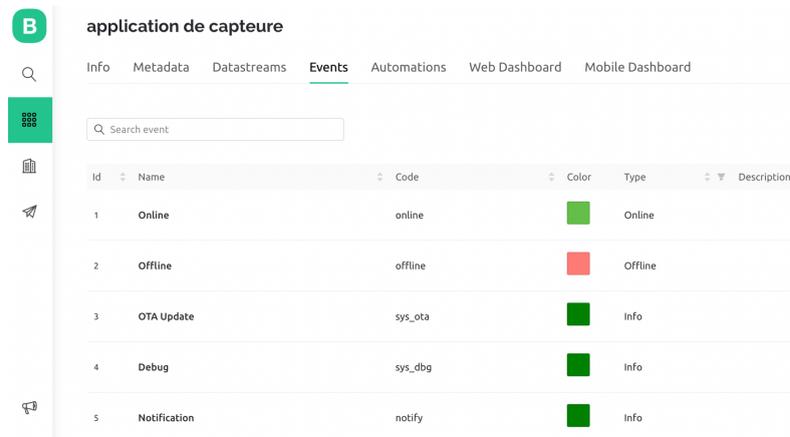


Figure 46: Capture d’écran de la page Event

Étape 10. Cliquer sur la page Automations. Ici, vous pourrez identifier ce que chaque variable représente.

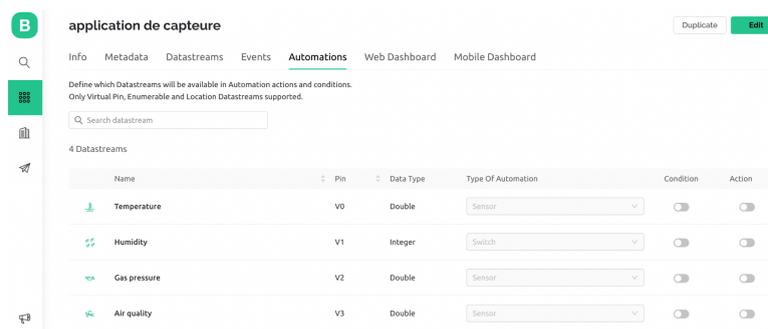


Figure 47: Capture d’écran de la page Automations

Étape 11. Cliquer sur Web Dashboard. Ici, vous pourrez configurer le tableau de bord en incorporant les tables, graphiques et diagrammes représentant les données collectées.

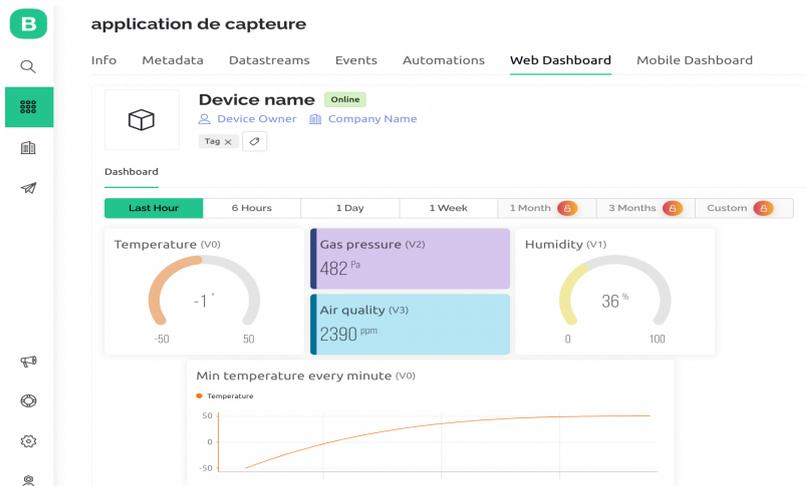


Figure 48: Capture d'écran de la page Web Dashboard

Étape 12. Ouvrir Arduino IDE et ouvrir le code. Vous trouverez le code utilisé sur notre page MakerRepo sous le document nommé URFPiezo2_FR.ino

<https://makerepo.com/Jerg08/1625.toe-beans>

Étape 13. Changer les informations du code aux informations de votre projet. Revenir sur la page Info pour trouver les deux premières lignes du code et remplacer les. Inscrire le nom et le mot de passe du WiFi que le système utilisera.

```
#define BLYNK_TEMPLATE_ID "TMPL4YzQ1FrP"
#define BLYNK_TEMPLATE_NAME "FB33"
#define BLYNK_AUTH_TOKEN "P7pYbn3vtAofq1kf6Y9XxvRzbSNrpW5Z"
#define BLYNK_PRINT Serial
#include <WiFinINA.h>
#include <BlynkSimpleWiFinINA.h>

char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "guOttawa"; // type your wifi name
char pass[] = ""; // type your wifi password
```

Figure 49: Capture d'écran du code 1

Étape 14. Au besoin, modifier et ajouter les pins qui seront utilisés sur le micro contrôleur.

```
#define BME_SCK (13)
#define BME_MISO (12)
#define BME_MOSI (11)
#define BME_CS (10)
```

Figure 50: Capture d'écran du code 2

Étape 15. Identifier les pins selon ceux inscrites sur Blynk.

```
Blynk.virtualWrite(V0, temperature);
Blynk.virtualWrite(V1, humidity);
Blynk.virtualWrite(V2, pressure);
Blynk.virtualWrite(V3, gas);
```

Figure 51: Capture d'écran du code 3

Étape 16. Implémenter le code terminé sur le micro contrôleur.

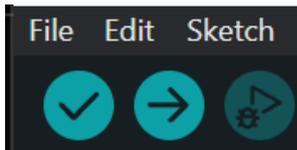


Figure 52: Capture d'écran du code 4

Étape 17. Retourner sur Blynk. Si tout fonctionne, les données devraient s'afficher sur la page Web Board et Mobile Board.

Étape 18. Placer le circuit dans la boîte et la refermer.

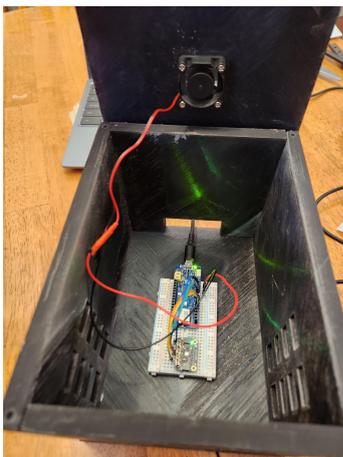


Figure 53: Photo de l'intérieur de la boîte finale

6.4 Essais & validation

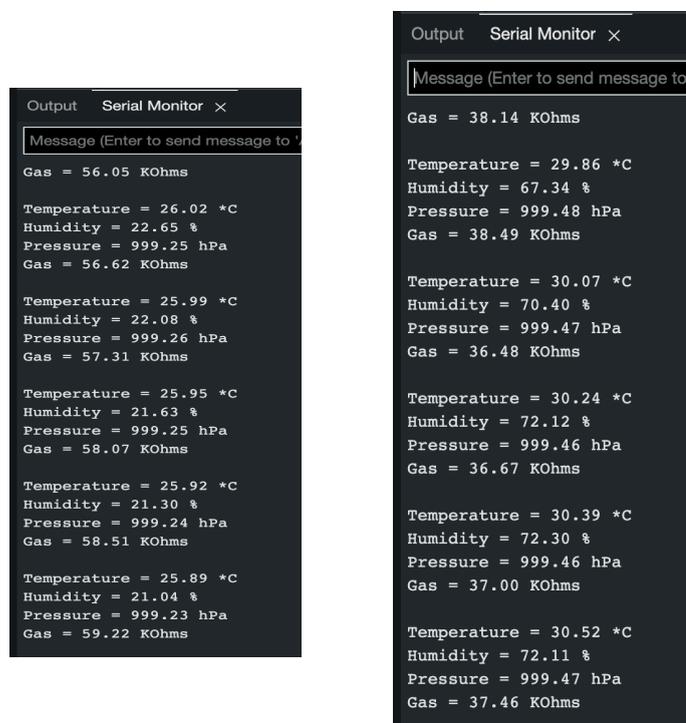
6.4.1 Design

Pour tester la résistance de la boîte, nous avons appliqué un poids sur les différents morceaux pour vérifier que l'épaisseur était bonne. Nous avons également tenté de plier et tordre les morceaux. Ceux-ci n'ont subi aucun dommage ou signe de défaillance.

6.4.2 Circuit

Une fois que le circuit a été assemblé selon nos plans, nous avons branché le système dans un ordinateur ayant le code ouvert et avons implémenté ce code sur la carte. Avec un code fonctionnel, il était possible de lire les données captées par le BME 680. Cependant, le capteur se déconnectait puisque les fils étaient trop petits et le capteur ne restait pas en place. Nous avons donc soudé les fils au capteur et le circuit fonctionnait bien après cette étape.

Pour voir si le système fonctionnait, nous avons mis un doigt sur le capteur. Nous avons donc vu que la température et l'humidité changeaient en fonction du temps que l'on laissait le doigt sur le capteur. Voici les données avant et après avoir mis un doigt sur le capteur:



The image shows two screenshots of the Arduino IDE Serial Monitor. The left screenshot shows the initial sensor readings, and the right screenshot shows the readings after a finger is placed on the sensor, causing a significant increase in temperature and humidity.

| Parameter | Initial Value | Value with Finger |
|-------------|---------------|-------------------|
| Gas | 56.05 KOhms | 38.14 KOhms |
| Temperature | 26.02 °C | 29.86 °C |
| Humidity | 22.65 % | 67.34 % |
| Pressure | 999.25 hPa | 999.48 hPa |
| Gas | 56.62 KOhms | 38.49 KOhms |
| Temperature | 25.99 °C | 30.07 °C |
| Humidity | 22.08 % | 70.40 % |
| Pressure | 999.26 hPa | 999.47 hPa |
| Gas | 57.31 KOhms | 36.48 KOhms |
| Temperature | 25.95 °C | 30.24 °C |
| Humidity | 21.63 % | 72.12 % |
| Pressure | 999.25 hPa | 999.46 hPa |
| Gas | 58.07 KOhms | 36.67 KOhms |
| Temperature | 25.92 °C | 30.39 °C |
| Humidity | 21.30 % | 72.30 % |
| Pressure | 999.24 hPa | 999.46 hPa |
| Gas | 58.51 KOhms | 37.00 KOhms |
| Temperature | 25.89 °C | 30.52 °C |
| Humidity | 21.04 % | 72.11 % |
| Pressure | 999.23 hPa | 999.47 hPa |
| Gas | 59.22 KOhms | 37.46 KOhms |

Figure 54: Captures d'écran des données capturées sur Arduino IDE

Ensuite, nous avons comparé les données affichées dans Arduino sur le **Serial Monitor** et les avons comparées à d'autres instruments, comme un thermostat. Les données étaient identiques, ce qui confirme que le circuit et le capteur fonctionnaient bien.

Pour continuer, nous avons testé le ventilateur pour s'assurer qu'il n'influence pas les données récoltées par le capteur. Le ventilateur fonctionne idéalement avec 12V, hors le circuit fonctionne à 5V. Cela a donc fait en sorte que le RPM du ventilateur était plus bas et n'influence pas les données, mais permet tout de même de faire de la circulation dans la boîte. Voici les données sans et avec le ventilateur:

| | |
|---|---|
| Temperature = 25.55 *C Humidity = 19.51 % Pressure = 1000.08 hPa Gas = 89.83 KOhms | Temperature = 25.93 *C Humidity = 22.29 % Pressure = 1000.16 hPa Gas = 82.57 KOhms |
| Temperature = 25.55 *C Humidity = 19.45 % Pressure = 1000.08 hPa Gas = 90.22 KOhms | Temperature = 25.95 *C Humidity = 22.29 % Pressure = 1000.16 hPa Gas = 82.65 KOhms |
| Temperature = 25.54 *C Humidity = 19.40 % Pressure = 1000.08 hPa Gas = 90.42 KOhms | Temperature = 25.95 *C Humidity = 22.37 % Pressure = 1000.15 hPa Gas = 81.92 KOhms |
| Temperature = 25.54 *C Humidity = 19.36 % Pressure = 1000.07 hPa Gas = 90.42 KOhms | Temperature = 25.97 *C Humidity = 22.54 % Pressure = 1000.15 hPa Gas = 81.92 KOhms |
| Temperature = 25.54 *C Humidity = 19.32 % Pressure = 1000.08 hPa Gas = 90.91 KOhms | Temperature = 25.97 *C Humidity = 22.45 % Pressure = 1000.15 hPa Gas = 83.23 KOhms |

Figure 55: Captures d'écran des données capturées sur Arduino IDE avec le ventilateur

Nous avons également testé différents modes d'alimentation. Le câble micro usb vers un port USB d'ordinateur fonctionne toujours bien. Nous avons essayé avec un chargeur portatif et une prise murale avec un bloc à prise USB, mais ces deux options ne fonctionnaient pas pour des raisons que nous n'avons pu identifier.

6.4.3 Codage

Nous avons d'abord utilisé la librairie Adafruit BME 680 test pour avoir la base du code que nous allons utiliser.

```
This is a library for the BME680 gas, humidity, temperature & pressure sensor

Designed specifically to work with the Adafruit BME680 Breakout
----> http://www.adafruit.com/products/3660

These sensors use I2C or SPI to communicate, 2 or 4 pins are required
to interface.

Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing products
from Adafruit!

Written by Limor Fried & Kevin Townsend for Adafruit Industries.
BSD license, all text above must be included in any redistribution
*****/

#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BME680.h"
```

Figure 56: Capture d'écran du code 5

```
#define BME_SCK 13
#define BME_MISO 12
#define BME_MOSI 11
#define BME_CS 10

#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_BME680 bme; // I2C
//Adafruit_BME680 bme(BME_CS); // hardware SPI
//Adafruit_BME680 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK);

void setup() {
  Serial.begin(9600);
  while (!Serial);
  Serial.println(F("BME680 test"));

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

Figure 57: Capture d'écran du code 6

```

// Set up oversampling and filter initialization
bme.setTemperatureOversampling(BME680_OS_8X);
bme.setHumidityOversampling(BME680_OS_2X);
bme.setPressureOversampling(BME680_OS_4X);
bme.setIIRFilterSize(BME680_FILTER_SIZE_3);
bme.setGasHeater(320, 150); // 320°C for 150 ms
}

void loop() {
  if (! bme.performReading()) {
    Serial.println("Failed to perform reading :(");
    return;
  }
  Serial.print("Temperature = ");
  Serial.print(bme.temperature);
  Serial.println(" °C");

  Serial.print("Pressure = ");
  Serial.print(bme.pressure / 100.0);
  Serial.println(" hPa");
}

```

Figure 58: Capture d'écran du code 7

```

Serial.print("Humidity = ");
Serial.print(bme.humidity);
Serial.println(" %");

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

Serial.print("Approx. Altitude = ");
Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
Serial.println(" m");

Serial.println();
delay(2000);
}

```

Figure 59: Capture d'écran du code 8

À partir de ce code, nous avons ajouté les composantes liant le système à Blynk, les pins du micro contrôleur et du capteur BME 680. Le processus était majoritairement essai-erreur, puisqu'aucun d'entre nous dans l'équipe n'avait d'expérience en programmation. Avec de la recherche et les forums d'Arduino, nous avons pu arriver au code final fonctionnel. Une fois que les données s'affichent bel et bien sur Blynk, nous avons arrêté de changer le code.

7 Conclusions et recommandations pour les travaux futurs

Au cours de ce projet, nous avons appris l'importance de la conception de circuits, de la programmation et de la sélection de composants pour développer un système de capteurs précis et fiables. Nous avons également appris comment mesurer et interpréter les données de température, ainsi que comment les transmettre à des dispositifs externes pour une utilisation ultérieure.

Pour améliorer notre travail futur, nous recommandons l'utilisation d'un capteur de température plus sophistiqué pour obtenir des mesures encore plus précises. Nous pourrions également envisager d'utiliser des algorithmes de traitement de signal plus avancés pour améliorer la précision de nos mesures. Une autre piste productive serait d'intégrer notre capteur de température dans un système plus complexe, comme un système de surveillance de la température de la maison ou un système de contrôle de la climatisation.

Si nous avons quelques mois de plus pour travailler sur ce projet, nous pourrions envisager d'ajouter des fonctionnalités supplémentaires telles que la connectivité sans fil pour la transmission de données à distance, ou la possibilité de stocker les données de température dans une base de données pour une analyse ultérieure. Nous aurions également pu effectuer plus de tests pour optimiser le fonctionnement du capteur et valider la précision des données.

8 Bibliographie

- Bosch Sensortec. (2021). BME680 Datasheet. <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme680-ds001.pdf>
- Blynk. (2021). Blynk Documentation. <https://docs.blynk.io/>
- National Institute of Standards and Technology (NIST). (2019). Guide to Industrial Control Systems (ICS) Security. <https://www.nist.gov/publications/guide-industrial-control-systems-ics-security>
- International Electrotechnical Commission (IEC). (2018). IEC 62443-4-1:2018. Security for industrial automation and control systems - Part 4-1: Secure product development lifecycle requirements.
- TechTarget. (n.d.). Troubleshooting. <https://searchnetworking.techtarget.com/definition/troubleshooting>

APPENDICES

9 APPENDICE I: Fichiers de conception

Table . Documents référencés

| Nom du document | Emplacement du document et/ou URL | Date d'émission |
|---|---|------------------------|
| Lien MakerRepo | https://makerepo.com/Jerg08/1625.toe-beans | 30 mars 2023 |
| projet conception STL.stl | https://makerepo.com/Jerg08/1625.toe-beans | 30 mars 2023 |
| Code: UFRPiezo2_FR.ino | https://makerepo.com/Jerg08/1625.toe-beans | 13 avril 2023 |
| Manuel d'utilisation et de produit_FB33.pdf | https://makerepo.com/Jerg08/1625.toe-beans | 16 avril 2023 |