

Livrable G – Prototype 2 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 12 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 2 :.....	3
3.2. Prototype de l'application.....	3
3.2.1. Test.....	3
3.3. Prototype de source alimentation	4
3.3.1. Test.....	4
Le code pour la batterie.....	5
4. Analyse de composante ou de système critique	6
5. Onshape :	7
5.1. Lien.....	7
6. Spécification cibles.....	8
7. Plan d'essai de prototype	8
8. Wrike.....	9
9. Conclusion.....	9
10. Références	9

1. Introduction

L'objectif de ce livrable est de poursuivre le développement du deuxième prototype en prenant en compte les commentaires et les rétroactions des clients et d'élaborer un plan d'essai pour le troisième prototype. Elle vise principalement à améliorer le produit afin qu'il réponde de manière optimale aux besoins et aux attentes des clients, tout en garantissant son efficacité et son efficacité.

2. Rétroaction reçue du client

Le client a indiqué que le concept général de la boîte lui plaisait et qu'il trouvait le design visuellement attrayant. Toutefois, il n'a pas eu le temps de fournir des commentaires spécifiques sur les caractéristiques ou les fonctionnalités de la boîte. Malgré cela, le client a clairement indiqué qu'il était satisfait du résultat global.

Il nous a également recommandé d'ajouter des trous de ventilation à la boîte en raison de la chaleur générée par les composants, afin d'améliorer la circulation d'air et de prévenir la surchauffe qui pourrait endommager les composants.

3. Développement du prototype

3.1. Objectif du prototype 2 :

L'objectif du prototype 2 c'est de créer un modèle du prototype numérique pour voir si elle fonctionne. Par la suite, nous voulons créer un prototype qui sera tester pour voir l'application et les données qu'elle nous offre. De plus, nous développerons un prototype analytique de l'application pour afficher la température, l'humidité et la qualité d'air prise. Enfin celui de notre source d'alimentation.

3.2. Prototype de l'application

3.2.1. Test

Nous avons défini nos variables dans Arduino IOT cloud et, à l'aide du code, nous avons programmé les seuils pour déclencher les alertes. En outre, nous ajouterons un diagramme en temps réel qui enregistrera la température et l'humidité, afin que les travailleurs puissent être tenus informés et alertés en cas de besoin.

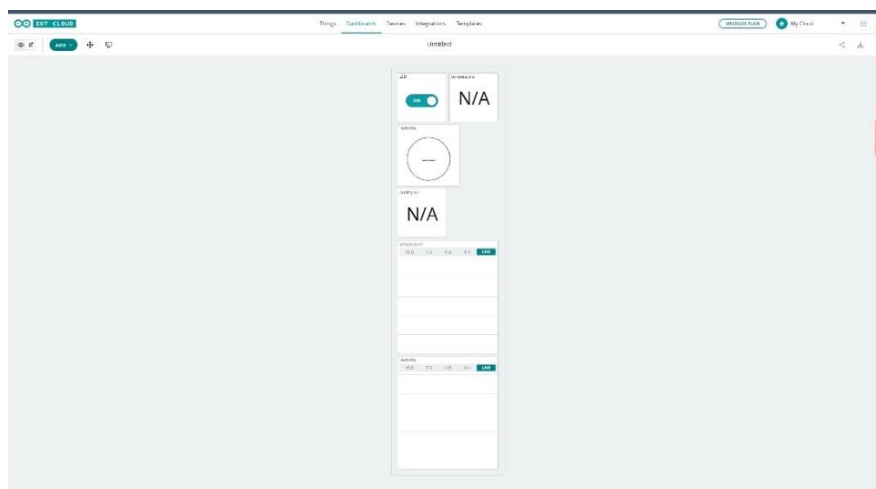


Figure 1 - Affichage sur l'application

Application: Installer [Arduino IOT cloud remote](#) sur votre mobile (Android ou Apple)

Lien: [Arduino Cloud IoT Cheat Sheet | Arduino Documentation | Arduino Documentation](#)

([Arduino Cloud IOT Cheat Sheet, 2023](#))

3.3. Prototype de source alimentation

3.3.1. Test

Pour la source d'alimentation, nous voulons que la source d'alimentation soit faite à partir des batteries rechargeables. De cette façon, les données seront toujours envoyées et la fiabilité du produit sera encore meilleure. Les données seront présentées sous forme de pourcentage de charge de la batterie et de tension, et deux graphiques afficheront les données courantes pour aider les travailleurs à avoir une idée de la situation avant les alertes.

Lien: <https://create.arduino.cc/iot/things> (Sign in to Arduino, 2023)

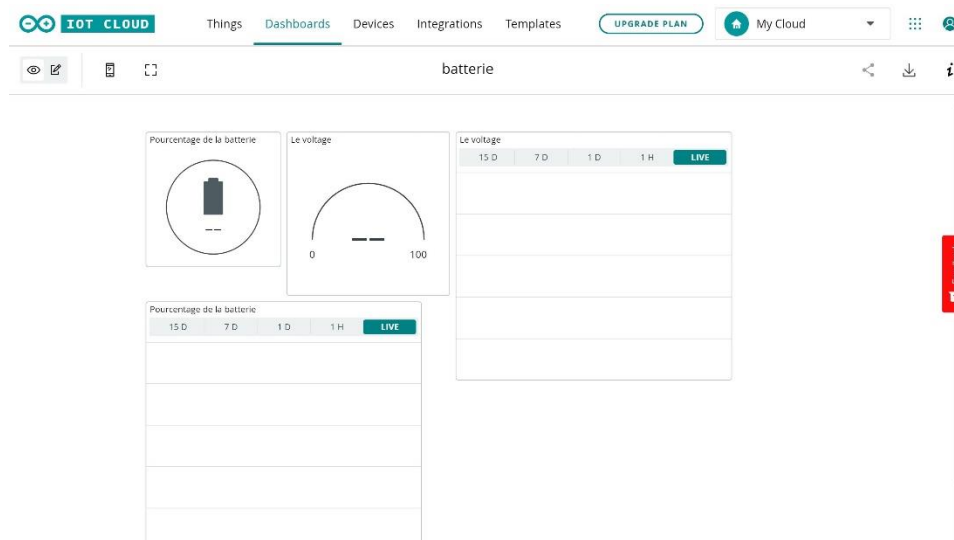


Figure 2 - Affichage sur l'application

Le code pour la batterie

```
/*
Sketch generated by the Arduino IoT Cloud Thing "Battery Monitoring"
https://create.arduino.cc/cloud/things/b7067d44-2084-45ca-bb06-424a1af2eb74
Arduino IoT Cloud Variables description
The following variables are automatically generated and updated when changes are made to the Thing
float voltage;
int bat_percentage;
Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this sketch.
*/
#include "thingProperties.h"
int analogInPin = A0; // Analog input pin
int sensorValue;
float calibration = 0.36; // Check Battery voltage using multimeter & add/subtract the value
void setup() {
// Initialize serial and wait for port to open:
Serial.begin(9600);
// This delay gives the chance to wait for a Serial Monitor without blocking if none is found
delay(1500);
// Defined in thingProperties.h
initProperties();
// Connect to Arduino IoT Cloud
ArduinoCloud.begin(ArduinoIoTPreferredConnection);
/*
The following function allows you to obtain more information
related to the state of network and IoT Cloud connection and errors
the higher number the more granular information you'll get.
The default is 0 (only errors).
Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();
}
void loop() {
ArduinoCloud.update();
sensorValue = analogRead(analogInPin);
voltage = (((sensorValue * 3.3) / 1024) * 2 + calibration); //multiply by two as voltage divider network is 100K &
100K Resistor
bat_percentage = mapfloat(voltage, 2.8, 4.2, 0, 100); //2.8V as Battery Cut off Voltage & 4.2V as Maximum Voltage
if (bat_percentage >= 100)
{
bat_percentage = 100;
}
if (bat_percentage <= 0)
{
bat_percentage = 1;
}
Serial.print("Analog Value = ");
Serial.print(sensorValue);
Serial.print("\t Output Voltage = ");
Serial.print(voltage);
Serial.print("\t Battery Percentage = ");
Serial.println(bat_percentage);
delay(1000);
}
float mapfloat(float x, float in_min, float in_max, float out_min, float out_max)
{
return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
}
```

4. Analyse de composante ou de système critique

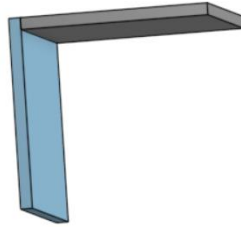


Figure 3 - Composante pour que le dessus de chaque compartiment puisse être visée

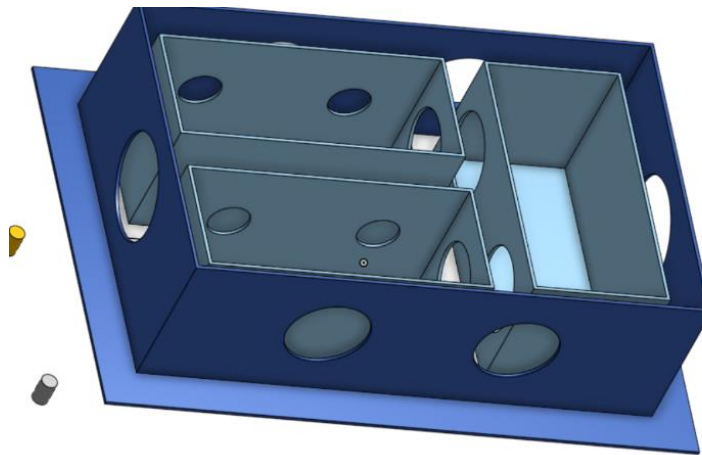


Figure 4 - Vue de l'intérieur (capsules ouvertes)

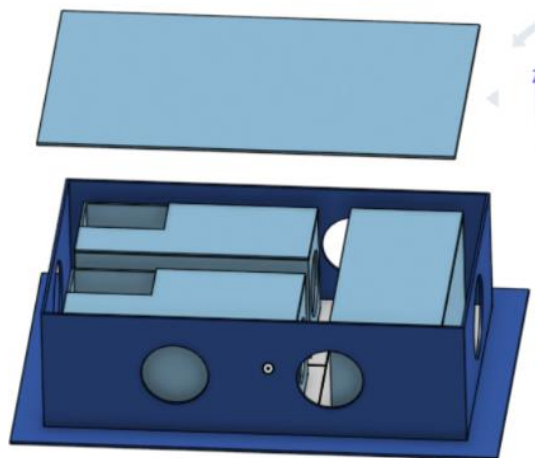


Figure 5 - Vue de l'intérieur (capsule fermée)

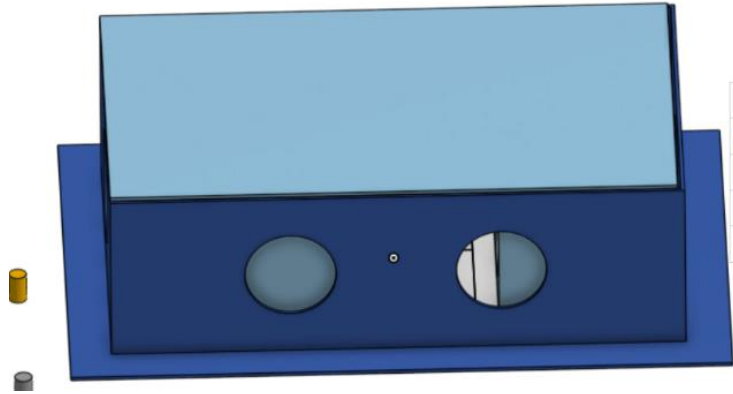


Figure 6 - Vue de l'extérieur (fermée)



Figure 7 - Les vis utilisées

5. Onshape :

- Je vous recommande de consulter le Part Studio 1 pour mieux comprendre le fonctionnement des vis pour ce produit.
- Pour la composante qui nous permet de rattacher les dessus, je n'en ai mis qu'une. Cependant pour ce produit, nous en aurons besoins de plus.

5.1. Lien

<https://cad.onshape.com/documents/4dfdabf216bcc6ae009f4b2c/w/19f42aff770531857d246639/e/3cdabb5170e82632a6c87121> (FB24, 2023)

6. Spécification cibles

Pour notre produit nous voulons avoir un produit qui reste dans le budget de 400 \$. Les matériaux utilisés ainsi que les toutes les composantes (ex. batteries rechargeables) de la boîte soit écologique. Notre boîte sera fabriquée d'un matériel d'aluminium qui sera résistant. D'après le besoin de nos clients, nous avons réévaluer les dimensions de la boîte à une dimension de 25x15x8 cm, ce qui fera que la boîte ne sera pas plus grosse qu'une boîte de soulier. D'après la rétroaction de nos clients notre produit finale aura des trous dans et à l'extérieur de la boîte afin d'assurer la ventilation de l'air dans la boîte. Il aura trois compartiments à l'intérieur de la boîte tel qu'une pour le capteur de température et humidité (DHT-11) avec un circuit de ESP-32, un compartiment pour le capteur de la qualité de l'air et un compartiment pour les batteries rechargeables et les fils attachées ensemble. De plus, les composantes de la boîte seront fixées par des vis afin de que les composantes de la boîte ne tombent pas en cas de chute ou ne soit pas déclasser de la boîte. Pour l'application, nous ciblons avoir une application qui montre les données qui seront en toute temps mise à jour afin que l'utilisateurs soit avisé des dernières données à chaque moment.

7. Plan d'essai de prototype

Tableau 1 - Plan d'essai de prototype

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 poids	Prototype physique, modèle réel du système	Assemblage de toutes les composantes dans la boîte pour déterminer le poids total du système	Durée : 1 jour Début : 14 mars
2	Tester les batteries	Prototype physique, installation du système d'alimentation	Mise en marche du système avec les batteries pour déterminer le temps entre les recharges	Durée : 1 semaine Début : 15 mars
3	Tester l'accrochage	Prototype physique, accrocher la boîte sur un mur	Accrochage de la boîte sur un mur à l'aide de vis pour déterminer la meilleure manière de l'accrocher	Durée : 2 jours Début : 14 mars

4	Tester l'application	Prototype analytique, utilisation d'un IOT	Un IOT sera utilisé pour recevoir les données des capteurs et les afficher sous forme de chiffres ou diagramme	Durée : 1 semaine Début : 13 mars
---	----------------------	--	--	--------------------------------------

8. Wrike

<https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=yWXy0MOnzvc3tllTn0VjqOrdbpUS2uGO%7CIE2DSNZVHA2DELSTGIYA>

9. Conclusion

Enfin, la poursuite du développement du deuxième prototype et l'élaboration d'un plan d'essai pour le troisième prototype sont des étapes importantes pour améliorer le produit et répondre aux besoins des clients d'une manière efficace et efficiente.

10. Références

Arduino Cloud IOT Cheat Sheet. (2023, Mars 12). Récupéré sur Docs: <https://docs.arduino.cc/arduino-cloud/getting-started/technical-reference>

FB24. (2023, Mars 12). *Client 2.* Récupéré sur On Shape: <https://cad.onshape.com/documents/4dfdabf216bcc6ae009f4b2c/w/19f42aff770531857d246639/e/3cdabb5170e82632a6c87121>

Sign in to Arduino. (2023, Mars 12). Récupéré sur Docs.