GNG1503

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

Système de surveillance environnementale pour un centre de données

Soumis par:

Stark Industries Inc. / FB22

Estelle Ngounou, 300269700

Inès N'Sanza, 300322445

Kelly-Ann Lessard, 300307532

Abigail Mulumba, 300301292

Ny Antsa Ratsimbarison, 300330531

Patient Ntita Nyembue, 300329948

2023-04-16

Université d'Ottawa

Table des matières

T	able d	es m	natièresii
L	iste de	e figu	uresError! Bookmark not defined.
L	iste de	e tabl	leauxiv
1	Int	trodu	action5
1	Ap	perçu	16
	1.1	Co	onventions
	1.2	Mi	ises en garde & avertissements
2	Po	our co	ommencer
	2.1	Co	onsidérations pour la configuration
	2.2	Co	onsidérations pour l'accès des utilisateurs
	2.3	Ac	ccéder/installation du système
	2.4 Organisation du système & navigation		
	2.5	Qu	uitter le système
3	Ut	iliseı	r le système
	3.1	Sys	stème électronique du SensorBox X
	3.1	1.1	DHT22
	3.1	1.2	MQ-135
	3.2	Sys	stème physique du SensorBox X
	3.3	Ap	oplication
	3.3	3.1	Code
	3.3	3.2	Interface
4	Dé	énanr	nage & assistance

	4.1	Mes	ssages ou comportements d'erreur	15
	4.2	Cor	nsidérations spéciales	15
	4.3	Ent	retien	15
	4.4	Ass	sistance	15
5	Do	cume	entation du produit	16
	5.1	Sol	ution finale	17
	5.1	.1	NDM (Nomenclature des Matériaux)	17
	5.1	.2	Liste d'équipements	18
	5.1	.3	Instructions	18
	5.2	Ess	ais & validation	19
6	Coı	nclus	sions et recommandations pour les travaux futurs	20
A	PPENI	DICI	ES	21
7	AP	PEN	IDICE I: Fichiers de conception	21

Liste de tableaux

Table 1. Documents référencés	21

1 Introduction

Dans le cadre du cours GNG1503, nous étions chargés de conceptualiser une boîte à capteur pour pouvoir monitorer les conditions environnementales nécessaires dans un centre de données. En effet, certaines conditions sont essentielles à surveiller dans un centre de données puisque si elles ne sont pas dans les marges désirés, l'équipement peut être endommagés ce qui ferait que les données puissent être affectés.

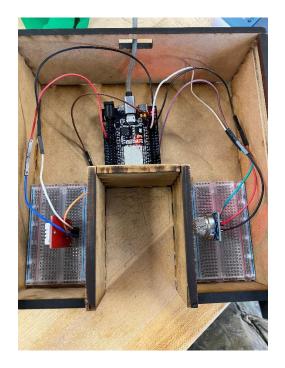
Après avoir fini tous nos prototypes et d'être venu à la solution finale, nous nous devons de faire un manuel d'utilisation et de produit (MUP) fournit les informations nécessaires aux ingénieurs pour utiliser efficacement le SensorBoxX, qui est notre prototype final, et pour la documentation du prototype.

1 Aperçu

Notre problème était de pouvoir créer un système complet et fonctionnel qui permettrait de surveiller les conditions environnementales désirées d'un centre de données. Les données importantes à capter était la température et l'humidité et la qualité d'air. Cela est assez important pour le bien-être de l'équipement dans le centre de données.

Notre produit est simple, efficace et facile à faire. Il arrive a limiter l'influence des capteurs entre eux. Il est fermé pour que rien ne rentre tout en ayant des fentes pour que l'environnement extérieur et intérieur soit homogène et les données soient fiables.





La boîte est faite en bois composite, on peut voir les deux capteurs et le microcontrôleur reliés par des fils.

1.1 Conventions

Dans ce manuel d'utilisateur, nous allons utiliser seulement un type de ponctuation pour pouvoir vous indiquer ce que vous en tant qu'utilisateur vous devez-faire. Nous allons utiliser «•» pour ce qu'il faut faire. Le reste est pour votre connaissance ou encore qui n'est pas forcément à faire.

1.2 Mises en garde & avertissements

Avant d'utiliser le prototype, on vous conseille de vraiment être familier avec le produit et savoir ce que chaque composante fait et les caractéristiques de celles-ci puisqu'il est très facile, comme on a une partie du système qui est électrique, de faire court-circuiter notre système.

2 Pour commencer

Ce système est utile à tout ceux qui aimerait monitorer certaines conditions environnementales. Voici comment utiliser notre produit :

- Poser la boîte sur une surface stable.
- Brancher le câble USB à une prise murale ou une batterie si voulu pour alimenter le système.
- Vérifier que les données soient envoyées à l'application.

2.1 Considérations pour la configuration

Pour la configuration du système, il est très important pour que les données soient transmises à l'application. Pour ce faire ne jamais oublier de changer dans le code le wifi lorsqu'on change d'environnement et l'implémenter au microcontrôleur.

```
char auth[] = "pHVnWC7TPB20BX1jCpaynj_kWgMN6Ik7";
char ssid[] = "45Mann";
char pass[] = "Rogers45Mann";
```

2.2 Considérations pour l'accès des utilisateurs

Ce système est utilisé pour surveiller la température, l'humidité et la qualité de l'air. Il vise alors les utilisateurs qui sont dans les domaines de la météorologie, ingénierie ou encore ceux qui ont des intérêts personnels.

2.3 Accéder/installation du système

• Avoir toutes les composantes

Écrire le code dans l'Arduino IDE et l'envoyer au microcontrôleur. (Ceci est notre code avec notre « Template». Bien évidement que vous en aurez un différent que le nôtre donc il est important que vous sachiez comment modifié les informations nécessaires.)

```
#include <MQ135.h>
 #include <BlynkSimpleEsp32.h>
 3 #include <WiFi.h>
 4 #include <WiFiClient.h>
    #include <WiFiServer.h>
    #include <WiFiUdp.h>
    #include <DHT.h>
 8 #include <DHT U.h>
 9
    #define BLYNK PRINT Serial
10
     #define BLYNK_TEMPLATE_ID "TMPL3-dVdAnr"
11
    #define BLYNK_TEMPLATE_NAME "ServicesPartagésCanada"
12
     #define BLYNK AUTH TOKEN "pHVnWC7TPB20BX1jCpaynj kWgMN6Ik7"
13
    char auth[] = "pHVnWC7TPB20BX1jCpaynj_kWgMN6Ik7";
15
     char ssid[] = "45Mann";
16
     char pass[] = "Rogers45Mann";
17
18
    #define DHTPIN 4
19
    #define DHTTYPE DHT22
20
     #define MQTYPE MQ135
21
     DHT dht(DHTPIN,DHTTYPE);
23
     BlynkTimer timer;
24
     void sendSensor()
26
27
28
       float h = dht.readHumidity();
      float t = dht.readTemperature();
29
       int mq135 = analogRead(34);
30
31
32
       if (isnan(h)||isnan(t)) {
33
         Serial.println("Failed to read from DHT sensor");
34
        return;
       if (isnan(mq135)){
35
         Serial.println("Failed to read from MQ sensor");
36
37
38
39
40
       Blynk.virtualWrite(V4, t);
41
       Blynk.virtualWrite(V5, h);
       Blynk.virtualWrite(V0, mq135);
42
43
44
45
     void setup() {
46
47
       Serial.begin(115200);
       delay(1000);
49
       Blynk.begin(auth, ssid,pass);
50
       dht.begin();
       timer.setInterval(5000L,sendSensor);
53
54
55
56
     void loop() {
57
       // put your main code here, to run repeatedly:
58
       Blynk.run();
60
       timer.run();
61
```

- Lié les composantes au microcontrôleur.
- Faire votre plateforme Blynk.
- Connecté le microcontrôleur à une batterie ou une prise.

2.4 Organisation du système & navigation

Notre système est essentiellement composé de 2 parties, la boîte et l'application. Pour la boîte tout est connecté par des encoches et de la colle pour les stabilisées. La boite a aussi un couvercle glissant qui permet d'avoir accès à l'intérieur facilement. Nous avons notre microcontrôleur et nos deux capteurs à l'intérieurs liée par des câbles.

Pour notre application, nous avons deux pages, celle de la température/humidité et celle de la qualité de l'air. Qui nous donne des graphiques et les données en temps réels.

2.5 Quitter le système

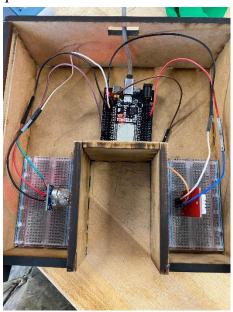
Pour quitter le système, il suffit seulement de débrancher le prototype et de ranger le câble dans la boîte et la mettre dans un endroit sécuritaire à température ambiante.

3 Utiliser le système

Les sous-sections suivantes fournissent des instructions détaillées, étape par étape, sur la façon d'utiliser les diverses fonctions ou caractéristiques du SensorBoxX.

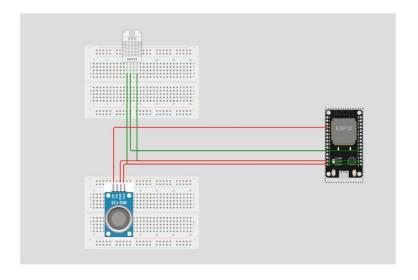
3.1 Système électronique du SensorBox X

Ce système est clé à notre produit final. Sans lui le fonctionnement de la boîte ne peut pas se produire. Nous avons deux capteurs, celui de température/humidité (DHT22) et celui de gaz (MQ-135) qui vont être relié à notre microcontrôleur (ESP32).



3.1.1 DHT22

Voici une représentation du circuit électrique. On voit que le DHT22 a 4 PIN mais on utilise que 3, le VCC, le DATA et le GND. Il faut connecter le VCC au 3V du Esp32, le DATA au D4 du Esp32 et le GND à un GND du Esp32.



3.1.2 MQ-135

Tout en vous fiant à la figure de la sous-section précédente, vous pouvez voir comment le MQ-135 est connecté au Esp32. Le capteur MQ-135 a 4 pin, le A0, le D0, le VCC, le GND. Comme vous le savez peut-être le MQ-135 est un capteur analogue alors il est important de connecter son pin A0 et non D0. Le A0 va donc être connecté au D34 du Esp32, le VCC au VIN du Esp32 et le GND au GND du Esp32.

3.2 Système physique du SensorBox X

Notre système physique consiste seulement de la boîte. Elle est faite en MDF, nous avons seulement utilisé une planche de 24x18x1/4 inches, pour toutes les parties. Cela nous a permis de faire cela.







3.3 Application

Notre application nous permet de transmettre les données récoltées par les capteurs et d'avoir accès à celle-ci en temps réel. Nous utilisons Blynk qui nous accès à des fonctions de base pour créer l'application.

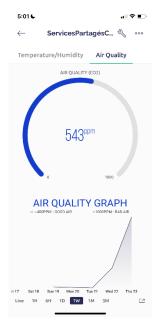
3.3.1 Code

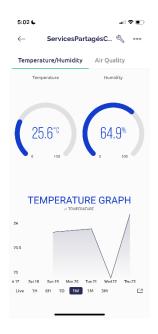
Voici le code que nous avons implémenter au Esp32 en utilisant la bibliothèque Blynk pour pouvoir envoyer les données à l'application.

```
#include <MQ135.h>
     #include <BlynkSimpleEsp32.h>
     #include <WiFi.h>
     #include <WiFiClient.h>
    #include <WiFiServer.h>
 6 #include <WiFiUdp.h>
7 #include <DHT.h>
 8 #include <DHT_U.h>
10 #define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPL3-dVdAnr"
     #define BLYNK_TEMPLATE_NAME "ServicesPartagésCanada"
     #define BLYNK_AUTH_TOKEN "pHVnWC7TPB20BX1jCpaynj_kWgMN6Ik7"
char auth[] = "pHVnWC7TPB20BX1jCpaynj_kWgMN6Ik7";
16 char ssid[] = "45Mann";
     char pass[] = "Rogers45Mann";
17
18
19
      #define DHTTYPE DHT22
21
     #define MQTYPE MQ135
     DHT dht(DHTPIN,DHTTYPE);
23
24
     BlynkTimer timer;
27
       float h = dht.readHumidity();
28
       float t = dht.readTemperature();
29
       int mq135 = analogRead(34);
31
       if (isnan(h)||isnan(t)) {
   Serial.println("Failed to read from DHT sensor");
33
       if (isnan(mq135)){
         Serial.println("Failed to read from MQ sensor");
       Blynk.virtualWrite(V4, t);
41
       Blynk.virtualWrite(V5, h);
       Blynk.virtualWrite(V0, mq135);
42
43
     void setup() {
48
       delay(1000);
       Blynk.begin(auth, ssid,pass);
50
51
       dht.begin();
       timer.setInterval(5000L,sendSensor);
     void loop() {
       // put your main code here, to run repeatedly:
       timer.run();
61
```

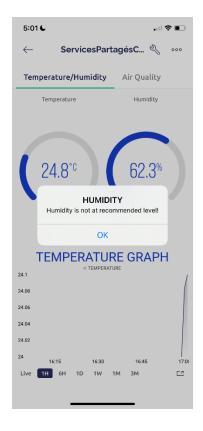
3.3.2 Interface

Ci-dessous vous pouvez observer l'interface blynk que nous avons choisi de faire pour notre application.





Nous avons aussi décidé de faire en sorte que lorsque les données ne sont pas dans les marges désirés une alerte est envoyée.





4 Dépannage & assistance

Les équipements de notre solution finale ont des rôles bien précis et c'est pour cela que l'utilisateur aurait besoin d'être bien conscient de tout ce qui pourrait affecter le bon fonctionnement du système.

4.1 Messages ou comportements d'erreur

Ce qui peut être une complication, serait les capteurs de gaz. Comme ils étaient importés, il n'y avait pas le manuel d'utilisation avec et l'erreur qu'on a fait est qu'on l'a utilisé directement au lieu de faire la période de préchauffage et que le capteur puisse s'adapter à l'air ambiante.

4.2 Considérations spéciales

Il serait assez bien d'acheter plusieurs capteurs de gaz, puisque ce sont les capteurs les plus fragiles.

4.3 Entretien

Comme ceci, est un système qui est censé roulé en permanence, il est important pour que les capteurs et le microcontrôleur ne s'endommage pas fréquemment, de laisser le système complet respirer parfois en le débranchant.

4.4 Assistance

- Logiciel Onshape: Toutes nos mesures serait disponibles dans nos livrables précédents ou vous pouvez contacter l'assistance sur le logiciel en question
- Logiciel Blynk: Notre code apparait dans nos livrables précédents ou si besoin d'assistance vous pouvez contacter l'assistance sur la page Web de Blynk.
- Dans le cas, où vous souhaitez en savoir un peu plus sur comment nous avons conceptualiser cette boîte, vous pouvez nous joindre à engou005@uottawa.ca.

5 Documentation du produit

Notre équipe devait passer par trois prototypes pour pouvoir faire le produit final. Par la suite, plusieurs tests ont été effectué pour pouvoir obtenir la meilleure solution dans les délais qu'on avait.

Tout, d'abord nous avons décidé d'utiliser Onshape pour concevoir notre boîte et avoir une idée de ce que notre boîte aurait l'air. Nous avions initialement décidé d'utiliser de l'ACM qui est un matériel que nous recommandons, cependant nous n'avions pas pu l'obtenir puisqu'il majoritairement vendu en quantité industrielle lorsqu'à bon prix. Nous avions un budget a respecté donc nous avons opté pour du MDF pour notre solution finale. En utilisant le MDF et ainsi la découpe Laser, Onshape nous a permis de faire le dessin qui a été envoyé à la machine via PDF.

On avait aussi pensé opter pour le Raspberry Pi, puis le Arduino Uno. On a rapidement vu que nous n'avions pas assez d'expérience avec le Raspberry Pi et le Arduino Uno nous donnait pas la facilité de transmettre les données à une application. C'est pour cela que nous avons opté pour le combo ESP32 et Blynk. Tout cela nous a donc donné une solution respectant le budget et respectant les besoins.

5.1 Solution finale

5.1.1 NDM (Nomenclature des Matériaux)

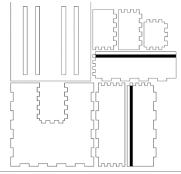
5.1.1 NDI	M (Nomenc	ature (ies Mai	<u>eriaux)</u>		
Nom de l'item	Description	Unité de mesure	nombre	Coût unitaire	Coût étendu	Lien
Jumper wires	100	mm	1	1.00 CAD	1.00 CAD	https://makerstore.ca/shop/ols/products/jumper-cables- per-10
Jumper wires	50	mm	1	1.00 CAD	1.00 CAD	https://makerstore.ca/shop/ols/products/jumper-cables- per-10
Capteur température et humidité	DHT22	unité	1	11.53 CAD	11.53 CAD	dht22
Capteur de gaz et fumée	MQ-135	unité	1	9.70 CAD	9.70 CAD	mq135
Esp32	Wifi module	unité	1	14.49 CAD	14.49 CAD	esp32
Breadboards	Connecter les capteurs à l'Arduino	unité	2	5 CAD	10 CAD	https://makerstore.ca/shop/ols/products/breadboard/v/B 15-HLF
Breakout board	Stabiliser le esp32	unité	1	3.58 CAD	3.58 CAD	https://phillipjfry.com/products/breakout-board-for-esp- wroom-32-cp2102-30-pin-module-with-usb-type-c
Feuille MDF	MDF (24x18 x1/4)	inch	1	4.00 CAD	4.00 CAD	https://www.piedmontplastics.com/products/acm-sheet
Onshape	Design la boîte	unité	1	0 CAD	0 CAD	
Carton	Maquette	unité	1	0 CAD	0 CAD	
Papier de construction	Maquette	unité	1	0 CAD	0 CAD	
Figma	Design l'interface de l'application	unité	1	0 CAD	0 CAD	
Blynk	Codez l'application	unité	1	0 CAD	0 CAD	
Arduino IDE	Codez pour les capteurs	unité	1	0 CAD	0 CAD	
Wifi.h	Bibliothèque implémenté dans Arduino					
BlynkSimpleEs p32	Bibliothèque implémenté dans Arduino					
Dht.h	Bibliothèque implémenté dans Arduino					
Coût total (sans taxe ou livraison) Coût total du produit(avec taxe et livraison)				55.30 CAD		
				98.67 CAD		

5.1.2 Liste d'équipements

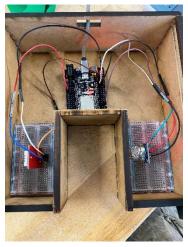
- Machine à coupe Laser
- Lime

5.1.3 Instructions

- Faire les dessins de chaque partie de la boîte dans un logiciel, nous avons choisi Onshape et Inkshape.
- Enregistré le fichier sous format pdf et l'envoyer dans la machine à coupe Laser.



- Mettre le matériel choisi, dans notre cas le MDF, dans la machine et configurer pour qu'il puisse être coupé et engravé.
- Ne pas hésiter à faire plusieurs couches de coupe si la coupe n'a pas traversé tout le matériel.
- Assemblez les parties ensemble, il se peut qu'elle ne soit pas très stable, on peut donc utiliser de la colle forte pour le stabiliser.
- Mettre les breadboards dans les deux compartiments de devant et les coller grâce à leur adhésif.
- Lié les capteurs sur les breadboards au microcontrôleur dans la partie principale de la boite.



5.2 Essais & validation

Plusieurs tests ont été effectué pour voir si nous étions capables d'obtenir des valeurs fiables. Nous avons donc essayé de récolter à l'aide de notre boîte les données du capteur de température et d'humidité et de le comparé au thermomètre de la pièce. En faisant cela nous avons pu voir que les données était fiable à l'entier prêt, ce qui était la précision qu'on recherchait. Pour celle du gaz, nous avons décidé qu'un de nos coéquipiers allait souffler devant le capteur de gaz pour voir si les données prises par le capteur allaient changer. Nous avons pu observer ce résultat ce qui nous a satisfait.

6 Conclusions et recommandations pour les travaux futurs

Pour conclure, durant ce projet, nous avons appris à nous familiariser à la conception 3D, la découpe laser et le codage. Nous avons aussi appris comment être débrouillard et développé notre pensée conceptuelle.

Si nous avions, plus de temps, nous aurions aimé ajouter plus de capteur, comme celui de poussière, de mouvement, de son et etc. Cela nous aurait permis de faire un projet totalement complet qui surpasserait les besoins.

APPENDICES

7 APPENDICE I: Fichiers de conception

Table 1. Documents référencés

Nom du document	Emplacement du document et/ou	Date d'émission	
Nom au document	URL	Date a chrission	
Identification des	https://makerepo.com/	22 janvier 2023	
besoins			
Critère de	https://makerepo.com/	29 janvier 2023	
conception			
Conceptualisation	https://makerepo.com/	5 février 2023	
Plan et cout du	https://makerepo.com/	12 février 2023	
projet			
Prototype 1	https://makerepo.com/	19 février 2023	
Prototype 2	https://makerepo.com/	5 mars 2023	
Prototype 3	https://makerepo.com/	12 mars 2023	
Journée de	https://makerepo.com/	29 mars 2023	
conception			
Présentation finale	https://makerepo.com/	3 avril 2023	