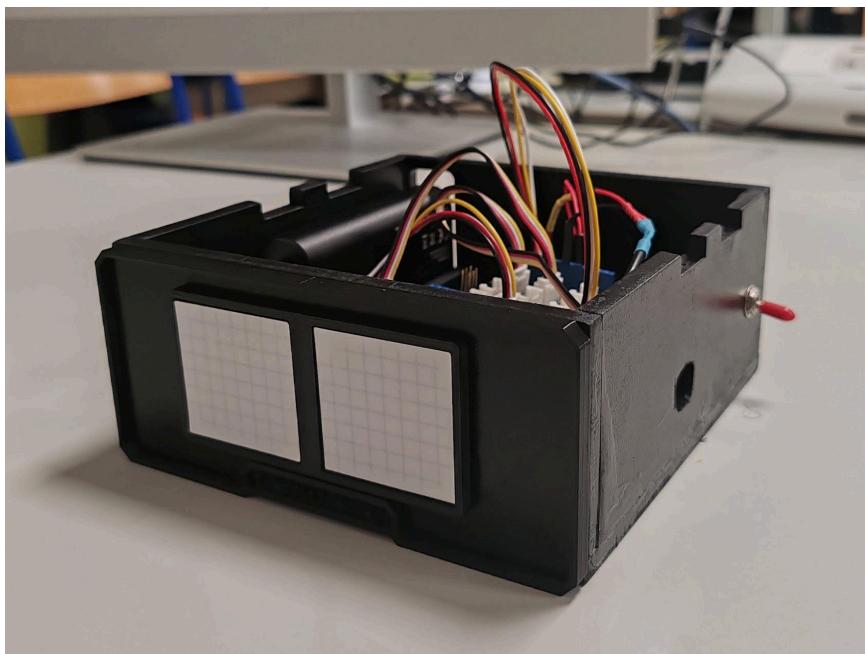


# Rapport de projet

## L1 - CMI

Session 2024

# Projet Compte Personnes



**Centre d'Enseignement et de Recherche en Informatique  
Campus Jean Henri Fabre  
339 Chemin des Meinajaries  
84911 Avignon Cedex 9**

**Auteurs :**

Mathis Hernandez

Amel Naak

**Encadrants :**

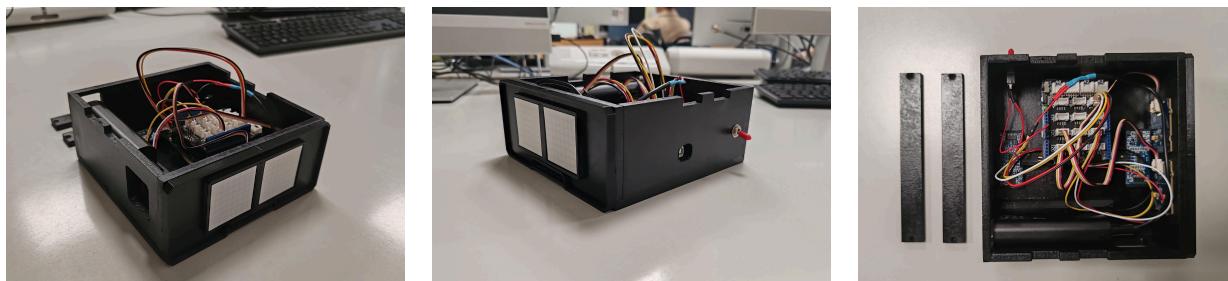
M. Gozlan

M. Silanus

# Résumé

L'objectif de ce projet est de réaliser un dispositif qui permet d'évaluer le nombre de personnes regroupées dans une salle en comptant les entrées et sorties, et de rendre cette information accessible en présentiel grâce à un compteur intégré et à distance depuis un site Web.

Nous sommes deux à avoir mené ce projet à bien et produit un système qui se positionne sur la partie supérieure d'un encadrement de porte et affiche le compteur sur un afficheur et sur un ordinateur grâce à une liaison filaire :



## Table des matières

<b>1 Introduction</b>	<b>3</b>
1.1 Contexte	3
1.2 Problématique	3
<b>2 Etat de l'art</b>	<b>3</b>
2.1 Solution 1 : Capteurs infrarouges	3
2.2 Solution 2 : Caméras 3D	3
2.3 Solution 3 : Tapis de comptage	3
<b>3 Solution développée</b>	<b>4</b>
<b>4 Organisation</b>	<b>4</b>
4.1 L'équipe de projet	4
4.2 Répartition des tâches	5
4.3 Environnement matériel/logiciel	5
4.4 Planification	6
<b>5 Réalisation</b>	<b>7</b>
5.1 Travail d'équipe	7
5.2 Partie de Mathis	7
Première tâche : Programmation	7
Deuxième tâche : Conception 3D	13
Troisième tâche : Assemblage et peinture	15
5.3 Partie d'Amel	17
5.4 Intégration	21
<b>6 Conclusion</b>	<b>22</b>

# Projet Compte Personnes

## 1 Introduction

### 1.1 Contexte

Ce projet a été réalisé dans le contexte du cours CMI 3, au sein du CMI Informatique de l'Université d'Avignon.

Il s'inscrit dans une démarche d'apprentissage de l'électronique embarquée, de la programmation sur microcontrôleur et de l'interconnexion avec une interface numérique.

### 1.2 Problématique

Comment concevoir un système compact et autonome permettant de détecter les entrées et sorties d'une salle, tout en affichant sur place le nombre de personnes qui y sont présentes et transmettre cette information via Bluetooth à une interface Web ?

## 2 Etat de l'art

Le fait de compter le nombre de personnes présentes dans un lieu permet de répondre à différents besoins, à différentes échelles :

- Pour un magasin : connaître les horaires d'affluence, les rayons les plus fréquentés
  - Pour un aéroport : adapter les mesures de sécurité
  - Dans les transports publics : ajuster la fréquence de passage selon l'utilisation réelle
  - Dans des bureaux/entreprises : Optimiser la consommation d'énergie
- ...

Pour cela, plusieurs solutions existent déjà :

### 2.1 Solution 1 : Capteurs infrarouges

Le comptage de passage par infrarouge utilise un faisceau infrarouge qui, lorsqu'il est interrompu par une personne, donne l'information d'un passage sur l'écran positionné sur le dispositif, qui affiche le nombre de personnes présentes dans la pièce.

### 2.2 Solution 2 : Caméras 3D

Une solution plus coûteuse et moderne est l'utilisation de caméras 3D qui utilisent une technologie de détection 3D pour analyser le mouvement et donc conclure si la personne entre ou sort. Les informations sont ensuite consultables sur l'ordinateur relié au dispositif ou depuis un navigateur Web.

### 2.3 Solution 3 : Tapis de comptage

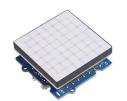
Le tapis de comptage peut être installé sous de la moquette ou du parquet flottant, utilise des capteurs de pression pour détecter les mouvements à la surface et utilise l'intelligence artificielle pour traiter les informations et conclure d'un passage. Les données sont ensuite communiquées par 4G, RJ45 ou Wi-Fi.

(Sources : [AES Protection](#) - [Sick AG](#) - [Tandem Security](#))

### 3 Solution développée

Nous avons alors décidé de partir sur un système alimenté par batterie qui se positionne sur la partie supérieure de l'encadrement d'une porte, détecte une entrée ou une sortie grâce à deux capteurs positionnés vers le bas, affiche le nombre de personnes présentes dans la salle sur un affichage sur place, et envoie les informations via Bluetooth vers un site Web.

Pour cela, nous utiliserons le matériel suivant :

Besoin	Matériel utilisé	
Détecter une entrée / sortie	2 capteurs à ultrasons HC-SR04	
Afficher le nombre de personnes présentes	2 matrices LED RGB Grove 8x8	
Traiter les informations	Carte Arduino Uno + Grove Base Shield	
Alimenter le système	Batterie 5200mAh sortie 5V	
Partager les informations via Bluetooth [non mis en place]	Module Bluetooth HC06	

### 4 Organisation

#### 4.1 L'équipe de projet

Notre équipe chargée de la réalisation de ce projet est constituée de deux membres : Amel Naak et moi-même, Mathis Hernandez.

Nous nous sommes également aidé de l'expertise de nos professeurs :

- M. Gozlan pour la partie matérielle (Arduino, capteurs, afficheurs, structure)
- M. Silanus pour la partie logicielle et la programmation

Ainsi que du regard extérieur d'autres étudiants, internes et externes au CMI, dont Logan Villard et Fouad Sbaghi avec qui nous avons pu échanger sur nos projets respectifs.

Travaillant chacun sur des parties distinctes du projet, nous avons utilisé l'application de messagerie instantanée WhatsApp, échangé par mail et créé un dépôt Github pour nous organiser en dehors des séances en présentiel et partager nos travaux.

Lien vers notre GitHub : <https://github.com/MathisHz/Projet-CMI3-Compte-Personnes>

## 4.2 Répartition des tâches

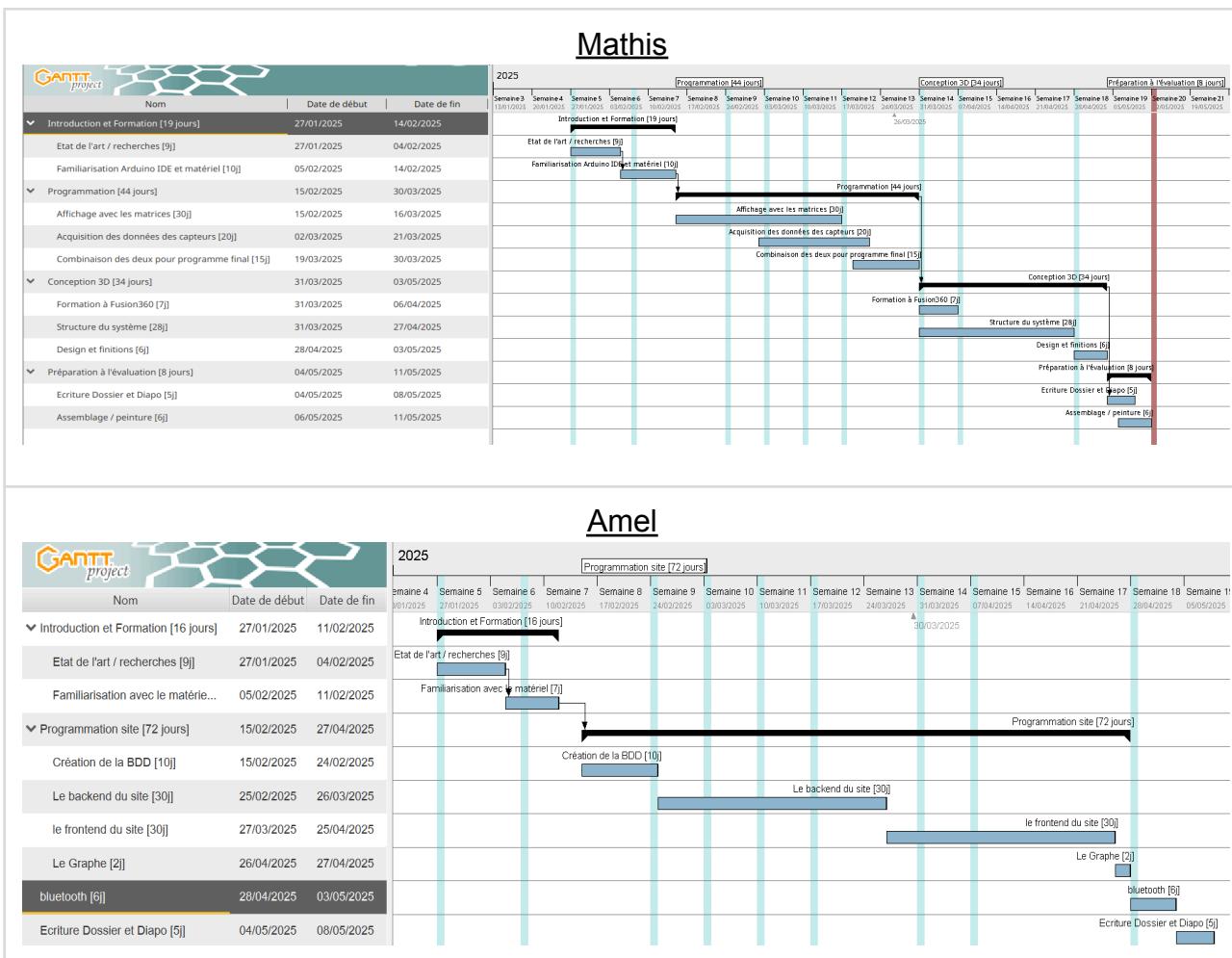
Responsable	Tâche	Besoins pour réaliser la tâche
Mathis	<u>Programmation</u>	- Formation Arduino IDE - Téléchargement des différentes bibliothèques
	<u>Conception 3D</u>	- Formation Fusion 360 - Modèles 3D des différents composants pour visualiser le rendu final
	<u>Assemblage et peinture</u>	- Bombe de peinture
	<u>Préparation du dossier et de la diapositive</u>	- Modèle de diapo de l'université
Amel	<u>Communication des informations en Bluetooth</u>	- Apprentissage fonctionnement du bluetooth et du framework arduino SoftwareSerial - module bluetooth HC-06
	<u>Stockage dans une base de données MySQL</u>	- Formation MySQL - Formation phpMyAdmin
	<u>Création du site Web</u>	- HTML - CSS - PHP
	<u>Réalisation d'un graphe</u>	- script PHP

## 4.3 Environnement matériel/logiciel

Domaine technique	Outil de développement utilisé
Plateforme matérielle (traitement embarqué des données acquises par les capteurs et affichage sur les matrices)	<u>Carte Arduino Uno</u> C'est une carte de développement adaptée aux débutants qui permet de connecter une variété d'appareils et qui est très bien documentée.
Programmation Arduino (C++)	<u>Arduino IDE</u> C'est l'environnement de développement officiel pour programmer les cartes Arduino. Il permet de compiler et téléverser directement le programme vers la carte et propose une base de bibliothèques.

Conception 3D	<u>Fusion 360</u> C'est un logiciel de conception 3D très complet et facile à prendre en main qui propose une licence gratuite pour les étudiants.
Affichage des données sur le site	- HTML et css pour l'interface du site - PHP pour récupérer et afficher les données stockées dans la base de données
Base de données	- MySQL (création de la BDD et des tabs) - PhpmyAdmin: pour visionner la base de données
Transmission des données en Bluetooth	- Transmettre les données de comptage (entrée/sortie) vers un script Python pour les stocker dans la base de données par un module Bluetooth HC-06.
Le graphe	PHP

## 4.4 Planification



## 5 Réalisation

### 5.1 Travail d'équipe

Nous avons mené nos travaux en séparant les tâches selon les compétences de chacun.

Nous avons pu communiquer à travers WhatsApp et GitHub pour nous accorder sur l'organisation et l'avancement de chacun, afin d'avancer à notre rythme tout en restant organisés.

C'est grâce à cela que nous avons abouti à un système fonctionnel qui détecte les entrées et sorties, les affiche en présentiel et sur un site Web.

Le système est alimenté par batterie et placé sur la partie supérieure d'un encadrement de porte au choix. Il suffit d'actionner l'interrupteur de mise en marche positionné sur le côté pour démarrer le système et le programme.

### 5.2 Partie de Mathis

#### Première tâche : Programmation

##### Étape 1 : Affichage avec les matrices

Pour commencer à prendre en main le matériel dont je dispose, je décide de faire un affichage simple sur une des deux matrices LED. Je branche donc la matrice à un port du bus I2C du Shield Grove de ma Arduino Uno.

J'utilise ensuite la [bibliothèque officielle de Seeed Studio](#) (qui produit ces matrices) que j'ai trouvé sur leur github, puis je la télécharge en .zip et l'installe manuellement en suivant [ce guide officiel](#). (il ne faut pas utiliser la bibliothèque qu'on peut trouver dans la partie "libraries" de l'Arduino IDE)

```
32 #include "grove_two_rgb_led_matrix.h"
33
34 GroveTwoRGBLedMatrixClass matrix;
35
36
37 void waitForMatrixReady() {
38     delay(1000);
39 }
40
41 void setup() {
42     Wire.begin();
43     SERIAL.begin(115200);
44     waitForMatrixReady();
45     uint16_t VID = 0;
46     VID = matrix.getDeviceVID();
47     if (VID != 0x2886) {
48         SERIAL.println("Can not detect led matrix!!!");
49         while (1);
50     }
51     SERIAL.println("Matrix init success!!!");
52
53     matrix.displayNumber(5000, 3000, true, 0x33);
54 }
```

Pour faire un test simple, j'utilise un des "examples" fournis : `rgb_display_num`

J'ai maintenant un affichage sur une matrice, je branche alors la deuxième et remarque que j'ai alors le même affichage sur les deux.  
Je dois donc répondre à la question : comment faire un affichage différent sur deux matrices ?

En faisant des recherches je trouve qu'il faut pour cela avoir des matrices qui ont deux adresses différentes. Je dois donc changer l'adresse d'une de mes matrices, et je dois faire cela au niveau du programme car il n'est pas possible de le faire au niveau matériel. En lisant la documentation et avec l'aide de mes professeurs qui me mettent à disposition le rapport de projet d'un groupe d'une année précédente, je trouve une fonction qui permet de changer l'adresse :

```
void changeDeviceBaseAddress(uint8_t newAddress);
```

Puis je trouve directement un constructeur qui permet de donner une adresse différente à un objet de la classe qu'on initialise :

```
GroveTwoRGBLedMatrixClass(uint8_t base, uint8_t screenNumber) {
```

Je crée donc deux objets pour mes deux matrices différentes, avec deux adresses différentes :

```
GroveTwoRGBLedMatrixClass matrix1(GROVE_TWO_RGB_LED_MATRIX_DEF_I2C_ADDR, 1);
GroveTwoRGBLedMatrixClass matrix2(GROVE_TWO_RGB_LED_MATRIX_DEF_I2C_ADDR + 1, 1);
```

Pour vérifier si les matrices sont bien branchées, l'exemple plus haut contient un test qui vérifie si le VID (Vendor IDentifier) est bien celui du fabricant.

Si oui, la matrice est bien détectée et tout va bien.

Sinon, un message d'erreur s'affiche.

J'ai alors implémenté cela dans mon programme pour mes deux matrices :

```
void setup() {
    Wire.begin();
    Serial.begin(9600);

    uint16_t VID1 = matrix1.getDeviceVID(); //VID de la matrice 1
    uint16_t VID2 = matrix2.getDeviceVID(); //VID de la matrice 2
    if (VID1 != 0x2886) {
        Serial.println("Can not detect led matrix 1 !!!");
    }
    delay(1000);
    if (VID2 != 0x2886) {
        Serial.println("Can not detect led matrix 2 !!!");
    }
    delay(1000);
    Serial.println("Matrix init success!!!"); //2 matrices bien branchées
}
```

Je peux donc maintenant définir une variable double "compteur", qui contiendra le nombre de personnes présentes dans la pièce, et qui sera affichée sur les matrices comme ceci :

```
// Affichage du compteur sur les matrices LED
matrix1.displayNumber(compteur / 10, 3000, true, 0xbdbed8);
matrix2.displayNumber(compteur % 10, 3000, true, 0xbdbed8);
```

### Obstacles rencontrés

Cette étape-là m'a posé problème car mes matrices refusaient de changer d'adresse et je ne pouvais donc pas avoir deux affichages différents.

Ce problème est d'autant plus étrange car lorsque nous avons, avec mes professeurs, essayé le même code, avec la même carte Arduino, mais en utilisant les matrices du projet d'un groupe d'une année précédente, un affichage différent sur les deux matrices était possible.

Le problème était donc matériel mais n'ayant pas trouvé de solution, j'ai continué la réalisation de cette partie du projet avec les matrices de ce groupe précédent. J'ai donc perdu beaucoup de temps à résoudre ce problème mais j'ai finalement pu continuer.

## Étape 2 : Détection par les capteurs

Pour utiliser les capteurs, j'installe la bibliothèque "HCSR04 ultrasonic sensor" de *gamegine*.

Pour savoir si le passage détecté est une entrée ou une sortie, on utilise deux capteurs, et on regarde lequel a été actionné en premier. Lorsqu'on utilise des capteurs à ultrasons, c'est une distance qui est captée. Puisque notre système sera placé au niveau du haut de l'encadrement d'une porte (un peu plus de 2m de haut), et que nous estimons qu'un humain fait plus d'1m40, j'ai donc fait en sorte que lorsqu'un capteur détecte une distance de moins de 60 cm (hauteur des capteurs - taille min. d'une personne), ce capteur est "actionné".

Mon programme devra donc vérifier en boucle si un capteur capte une distance <60 cm et s'il est le premier ou le deuxième dans une courte période de temps à être actionné.

Après avoir lu la documentation de la bibliothèque, je branche mes deux capteurs au Shield Grove de l'Arduino sur deux ports différents :

Capteur	Broche Trigger	Broche Echo
hc1	3	2
hc2	5	4

puis j'initialise les deux objets HCSR04 dans mon programme avec les broches correspondantes :

```
#include <HCSR04.h>

HCSR04 hc1(3, 2); // Capteur 1 (trig, echo)
HCSR04 hc2(5, 4); // Capteur 2 (trig, echo)
```

Avant de passer à la reconnaissance d'entrée ou de sortie, je décide de faire des essais pour apprendre à manipuler les fonctions de la bibliothèque.

La méthode dist() permettant de récupérer la distance captée par un objet de la classe, donc ici mes objets **hc1** et **hc2**.

Je fais alors un essai avec un court code qui affiche dans le moniteur série la distance renvoyée par la méthode pour chaque matrice :

```
void loop() {
    Serial.print("Distance capteur 1 : "); Serial.println(hc1.dist());
    Serial.print("Distance capteur 2 : "); Serial.println(hc2.dist());
    delay(200);
}
```

Cela affiche bien dans le moniteur série la distance captée par chaque capteur au millimètre près.

J'ai maintenant en main toutes les bases qu'il me faut pour faire mon programme final.

#### Obstacles rencontrés

J'ai à nouveau dans cette étape fait face à un problème technique :

Lors de mes premiers tests pour afficher dans le moniteur série les distances captées, celles-ci étaient constamment de 0, et le code n'était pas en faute car j'ai essayé plusieurs manières de faire que j'ai pu trouver durant mes recherches. J'ai donc conclu que le problème était à nouveau matériel mais en essayant plusieurs câbles et même d'autres capteurs rien n'allait.

C'est avec l'aide de M. Silanus que j'ai pu régler le problème : après m'avoir fourni une autre carte Arduino, mon programme de test fonctionnait et les bonnes distances s'affichaient.

J'ai à nouveau perdu beaucoup de temps à chercher au mauvais endroit la cause de mon problème, qui à nouveau résidait dans mon matériel de prêt.

### Étape 3 : Programme final : détection entrée/sortie et affichage sur matrices

Je veux pouvoir être en mesure de savoir à chaque itération de loop() dans quel état se trouve chaque capteur (a-t-il été actionné ou non). J'initialise alors deux booléens qui changeront de valeur au besoin :

```
bool hc1_detect = false;
bool hc2_detect = false;
```

Puis à chaque itération de loop(), j'initialise deux int qui prennent comme valeur la distance captée par chaque capteur :

```
int dist1 = hc1.dist();
int dist2 = hc2.dist();
```

Je peux maintenant commencer.

Je fais alors à chaque itération de loop() quatre tests :

Pseudo-code	Code
<ol style="list-style-type: none"> <li>Si <b>hc1</b> détecte une distance &lt;60cm et que <b>hc2</b> n'a pas encore été actionné -&gt; On actionne <b>hc1</b>.</li> <li>Si <b>hc2</b> détecte une distance &lt;60cm et que <b>hc1</b> n'a pas encore été actionné -&gt; On actionne <b>hc2</b>.</li> <li>Si <b>hc1</b> détecte une distance &lt;60cm et que <b>hc2</b> a déjà encore été actionné -&gt; On augmente le <b>compteur</b> -&gt; On remet <b>hc2</b> à son état par défaut.</li> <li>Si <b>hc2</b> détecte une distance &lt;60cm et que <b>hc1</b> a déjà encore été actionné -&gt; On diminue le <b>compteur</b> -&gt; On remet <b>hc1</b> à son état par défaut.</li> </ol>	<pre> if (dist1 &lt;= 60 &amp;&amp; !hc1_detect) {     hc1_detect = true; }  if (dist2 &lt;= 60 &amp;&amp; !hc2_detect) {     hc2_detect = true; }  if (hc2_detect &amp;&amp; dist1 &lt;= 60) {     if (compteur &gt; 0) compteur--; // Personne sortie     hc1_detect = false;     Serial.println("Personne sortie, compteur diminué !"); }  if (hc1_detect &amp;&amp; dist2 &lt;= 60) {     compteur++; // Personne entrée     hc2_detect = false;     Serial.println("Personne entrée, compteur augmenté !"); } </pre>

Et toujours à chaque itération, j'affiche le compteur sur les deux matrices.

**Problème :** Dans le principe, l'idée est bonne. En revanche, en pratique, lorsqu'un objet passe devant le premier capteur, **hc1\_detect** devient bien *true*, mais lorsqu'il passe devant le second, il redevient bien *false* mais **hc2\_detect** reste *true*.

Je conclus alors que le problème est un double-comptage :

**Solution :** J'ajoute deux contraintes, deux "verrous" :

- J'initialise un booléen **passage\_fini** en *false*, qui permettra de savoir si un passage vient tout juste de se terminer (s'il est *true*) et d'éviter le double comptage.
- Pour compléter cela, je dois le remettre à nouveau à *false* après un court délai.  
Pour apprendre comment implémenter cela dans mon code je fais des recherches et demande de l'aide à ChatGPT, l'agent conversationnel IA d'OpenAI.

J'ai donc implémenté ceci dans mon code :

Dans setup() :

```
unsigned long dernier_temps = 0;
const int delai_reset = 500; //500ms pour éviter un double comptage
```

une variable **unsigned long** est une variable entière (comme **int**) mais qui ne peut contenir que des valeurs  $\geq 0$  (**unsigned**) et plus grande qu'un simple **int** (**long**)

int	2 octets	valeurs de -32 768 à 32 767
unsigned long	4 octets	valeurs de 0 à 4 294 967 295

Dans loop() :  
(au début)

```
unsigned long temps_actuel = millis();
```

La fonction **millis()** donne le nombre de millisecondes écoulées depuis le démarrage de la carte.

Grâce à tout ça je peux construire un verrou qui empêchera le double-comptage :

1. Je regroupe mes quatres tests dans la contrainte que le programme ne vient pas tout juste de détecter un passage.

```
if (!passage_fini) {
    // Détection initiale par un capteur
    if (dist1 <= 60 && !hc1_detect) {
        hc1_detect = true;
    }
    if (dist2 <= 60 && !hc2_detect) {
        hc2_detect = true;
    }
    // Validation du passage (dans le bon ordre)
    if (hc1_detect && dist2 <= 60) {
        compteur++; // Personne entrée
        passage_fini = true;
        dernier_temps = temps_actuel;
        Serial.println("Personne entrée, compteur augmenté !");
    }
    if (hc2_detect && dist1 <= 60) {
        if (compteur > 0) compteur--;
        // Personne sortie
        passage_fini = true;
        dernier_temps = temps_actuel;
        Serial.println("Personne sortie, compteur diminué !");
    }
}
```

2. Je met en place une réinitialisation de l'état de mes capteurs (actionné -> neutre) ainsi que celui de **passage\_fini** peu de temps après un passage.

```
// Réinitialisation après un passage
if (passage_fini && (temps_actuel - dernier_temps > délai_reset)) {
    passage_fini = false;
    hc1_detect = false;
    hc2_detect = false;
    Serial.println("Passage terminé, réinitialisation.");
}
```

Ainsi, le programme atteint sa forme finale et lors de mes essais, le système fonctionne correctement. Le compteur de personnes fonctionne.

Puis quelques semaines passent et après avoir terminé l'assemblage de la structure du système et l'avoir installé dedans, je teste son bon fonctionnement à hauteur réelle avec l'aide de Logan Villard, et nous remarquons qu'il ne fonctionne pas comme il devrait : de temps en temps, alors que le compteur devrait s'incrémenter de 1 lors d'un passage, il le fait et puis décrémente aussitôt.

Après avoir analysé le code pour trouver l'erreur, nous concluons qu'il pourrait s'agir d'un problème au niveau du "reset" de **hc1\_detect** et **hc2\_detect**.

- Nous modifions alors le programme pour que le reset se fasse toutes les secondes plutôt qu'uniquement après un passage :

```
// Réinitialisation toutes les secondes
if (temps_actuel - dernier_temps > délai_reset) {
    passage_fini = false;
    hc1_detect = false;
    hc2_detect = false;
    dernier_temps = temps_actuel;
    Serial.println("Passage terminé, réinitialisation.");
}

delay(100); // Petite pause pour éviter une surcharge
```

Faire cela nous a permis de faire tourner le système comme prévu à hauteur humaine. 

## Deuxième tâche : Conception 3D

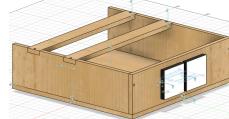
### Étape 1 : Conception de la structure

Pour débuter avec Fusion 360, j'ai regardé quelques vidéos sur Youtube, comme [celle-ci](#) de la chaîne "[La Nouvelle École - DIY](#)"

Ensuite, pour obtenir les modèles 3D des capteurs, de la carte Arduino et du module Bluetooth, je me suis tourné grâce aux conseils de M. Gozlan vers [GrabCAD](#), un environnement de collaboration gratuit avec une grande bibliothèque de modèles 3D créés et publiés par d'autres utilisateurs.

J'ai imaginé différents moyens d'accrocher le système en haut d'un encadrement de porte et la solution retenue a été un accrochage avec de l'adhésif double-face directement. La structure sera donc cubique, avec uniquement les afficheurs visibles de l'extérieur et les capteurs positionnés vers le bas, à travers quatre trous dans la base

J'ai néanmoins passé du temps à développer mes idées en 3D que je n'ai pas retenues :

1. Avant de partir sur le double-face, mon idée était de poser la structure sur une tringle à rideaux de douche -> pas pratique, idée oubliée	2. Première version avec double-face -> base de 20x20cm trop grande mais l'idée est là !	3. Pour réduire la profondeur, j'ai agrandi la largeur... mais un peu trop	4. Pour cette version j'ai réduit la base à 15x15cm, mais le problème ici était la batterie qui était trop grande, j'ai alors agrandi la base à 15x18cm et mis la carte au-dessus de la batterie	5. Le problème ici était le même que le précédent, j'ai alors laissé la base à 15x15cm mais j'ai mis les capteurs sur les côtés de la base, créant un look asymétrique plutôt dérangeant...
				

Je n'ai pas retenu ces versions mais elles m'ont permis néanmoins de m'exercer à la conception 3D, et de réaliser la version finale, en prenant en compte une batterie plus petite qui pourra être posée sur le côté, afin de garder une taille de 15x15cm.

Je commence alors la conception des différentes pièces en respectant les contraintes techniques ainsi que celles de M. Gozlan et les dimensions des appareils :

- Des trous de vis de 2 mm
- L'épaisseur des pièces est de 6 mm (profondeur du bois qui va être découpé au laser)
- Un trou sur la face gauche pour alimenter la batterie
- Un trou sur la face arrière pour voir l'indicateur de batterie
- Deux pièces pour retenir la batterie en place

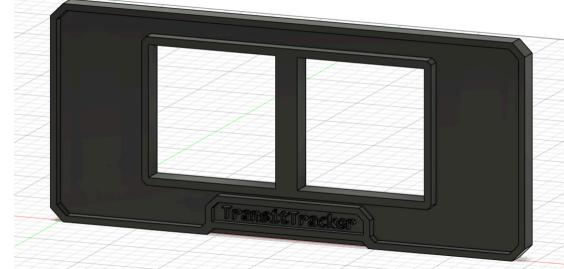
J'ai respecté ces contraintes pour construire les 8 pièces différentes de la structure, et les ai assemblées sur Fusion pour avoir un aperçu de la structure finale :



## **Étape 2 : Design et finitions**

Pour donner une identité à notre projet et en faire un réel objet technique, il a reçu le nom de **TransitTracker**.

Afin de rendre la face avant plus agréable à regarder, j'ai réalisé un design qui sera imprimé en 3D puis collé directement à la face avant de la structure, et qui lui permettra de porter fièrement son nouveau nom (qui sera plus visible après un coup de pinceau) :



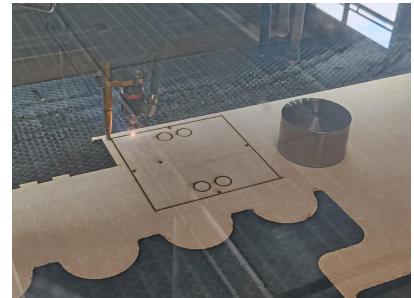
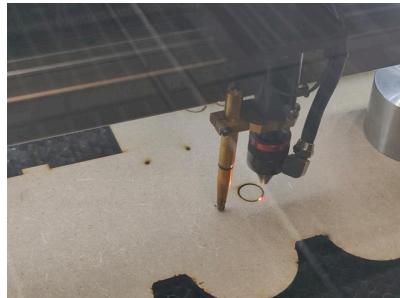
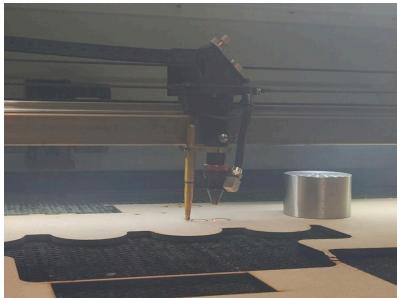
Un second défaut à corriger était les trous des capteurs au niveau de la base qui étaient trop grands (en voulant être sûr qu'il passent, j'ai pris trop de marge) J'ai alors fait une petite pièce qui viendrait se placer entre les capteurs et les trous pour combler l'espace vacant et en même temps réhausser les capteurs pour qu'ils ne dépassent pas de la structure :



Une fois toutes les pièces nécessaires à la réalisation de la structure entre les mains, je peux passer à la dernière partie : l'Assemblage.

### Troisième tâche : Assemblage et peinture

Les pièces en bois de la structure ont été découpées au laser avec une machine de l'IUT, qui est très près du CERI d'Avignon, dans des planches de bois MDF (Medium Density Fiberboard) ou "medium" :



La machine communiquait avec le logiciel LightBurn pour la découpe laser, dans lequel on dépose un fichier .dxf, généré en exportant l'esquisse 2D de la pièce à découper sur Fusion 360.

Les pièces ont ensuite été assemblées entre elles par vis et par colle. Le résultat était solide malgré quelques fragilités et fissures dûes à la finesse du panneau de medium. Il a fallu ensuite poncer les pièces sur la profondeur pour faire partir les marques de brûlures laissées par la découpe laser.



Pour soigner l'apparence, j'ai également passé un coup de peinture sur la structure (4 couches au total), avec une bombe de peinture noire satinée. Ci-dessous, deux couches ont été appliquées, et les pièces ne sont pas encore totalement sèches :



J'ai pu ensuite récupérer les 3 pièces que M. Gozlan m'a imprimées en 3D :

Supports / Réhausseurs pour capteurs

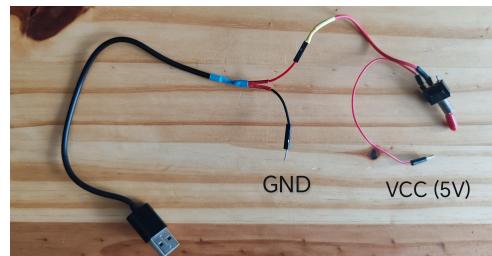


Design

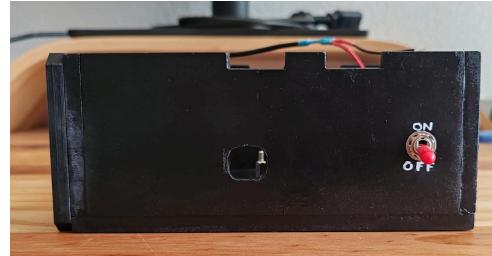


C'est après avoir conceptualisé en 3D puis assemblé la structure que je me suis dit qu'il serait bien d'ajouter un bouton Marche/Arrêt pour le système. C'est alors que nous nous sommes rendus compte avec M. Gozlan qu'il manquait deux ouvertures : une pour ce bouton, et une autre pour upload vers la carte Arduino (une ouverture pour le câble USB).

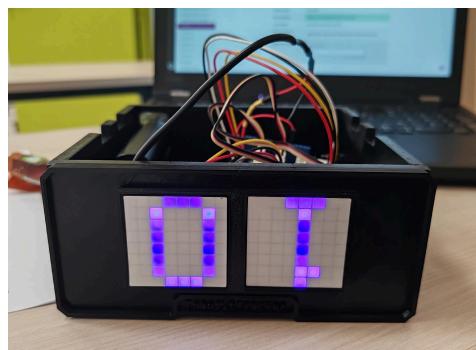
Nous avons donc soudé l'interrupteur au câble USB qui connectera la batterie à la carte :



Puis avons fait les trous pour le câble USB (qui bénéficiera d'une petite pièce 3D collée autour pour affiner le style) et pour le bouton M/A (auquel j'ajouterais au pinceau les inscriptions "ON" en haut et "OFF" en bas)



Ainsi, nous voilà en possession d'un système complet avec une structure prête à l'emploi une fois tous les appareils correctement installés :



(Lors de mes tests, j'ai malencontreusement installé les afficheurs à l'envers et j'étais pressé de prendre la photo. Errare humanum est. Néanmoins mon essai montre bien que mon système fonctionne)

## 5.3 Partie d'Amel

Cette partie du projet vise à assurer la transmission et l'affichage en temps réel des données collectées par le système de comptage. Grâce au module Bluetooth HC-06, les données sont envoyées sans fil depuis l'Arduino vers un script Python, qui les enregistre dans la base de données. Un site web développé en HTML, PHP et CSS, permet ensuite de visualiser ces données de manière claire.

### Etape 1 : Création de la base de données MySQL:

MySQL est un système de gestion de base de données relationnelle qui permet de stocker, organiser et gérer efficacement des données à l'aide de langage SQL.

La table "personnes": pour calculer le nombre de personnes dans la salles

	Éditer	Copier	Supprimer	id	salle_id	nb_entrée	nb_sortie	nombre_personnes	heure
<input type="checkbox"/>				1	1	10	3	7	2025-03-31 22:23:54
<input type="checkbox"/>				2	1	20	4	16	2025-03-31 22:24:55
<input type="checkbox"/>				3	1	10	4	6	2025-03-31 22:24:55

La table "salle": pour stocker les différentes salles

	Éditer	Copier	Supprimer	id	1	capacite	nom
<input type="checkbox"/>				1	50	Salle C024	
<input type="checkbox"/>				2	30	Salle S4 Nodes	
<input type="checkbox"/>				3	100	Amphi Blaise	

Voici le code permettant de créer ces deux tables:

```
1 CREATE TABLE salle ( id INT AUTO_INCREMENT PRIMARY KEY, capacite INT NOT NULL, nom VARCHAR(100) NOT NULL);

1 CREATE TABLE personnes (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     sale_id INT NOT NULL,
4     nb_entrée INT NOT NULL,
5     nb_sortie INT NOT NULL,
6     nombre_personnes INT NOT NULL,
7     heure VARCHAR(20) NOT NULL
8 );
```

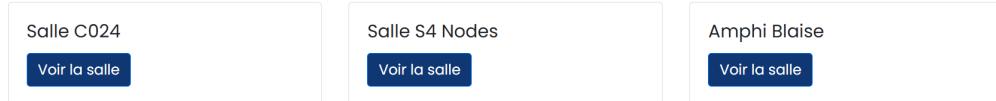
## **Etape 2 : Mise en page du site:**

Pour l'interface j'ai utilisé HTML et CSS pour structurer bien le site ainsi que le PHP pour récupérer les données de la base de données et les afficher.

Voilà l'interface du site:



Les salles :



En cliquant sur "voir la salle" vous trouvez: la capacité de la salle, le nombre de personnes présentes et le graphe de présence en fonction du temps :

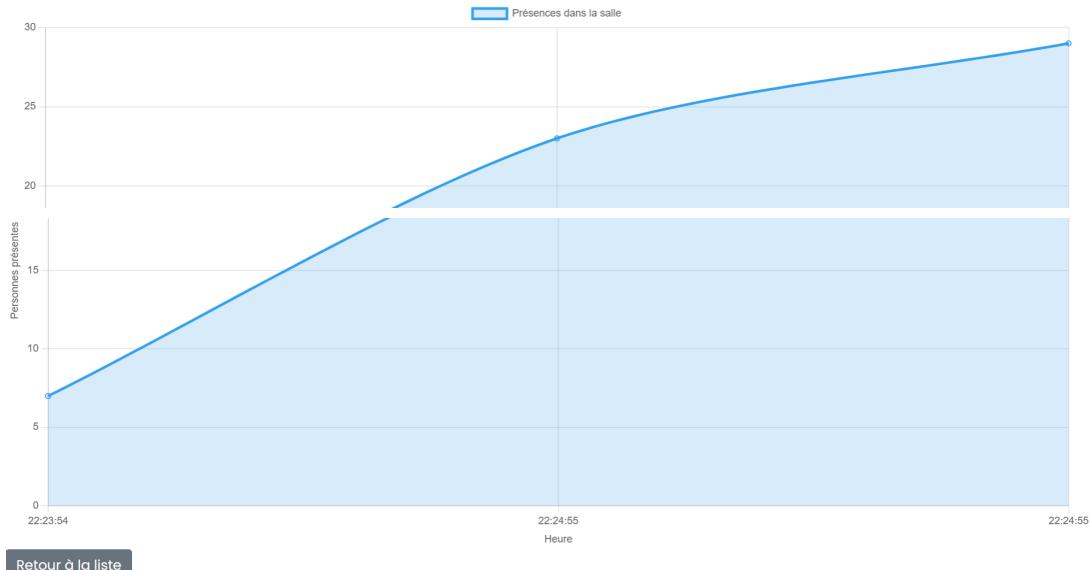


### Détails de la salle : Salle C024

Capacité : 50 personnes

Nombre de personnes présents : 29

Présence en temps réel



## **Difficultées rencontrées:**

Au départ, la barre de recherche ne fonctionnait que si on saisissait exactement le nom de la salle tel qu'il apparaît dans la base de données. Mais après quelques recherches, j'ai réussi à la rendre plus flexible, comme une recherche Google, en la rendant insensible à la casse et plus tolérante aux variations d'écriture.

### Etape 3 : génération du graphe :

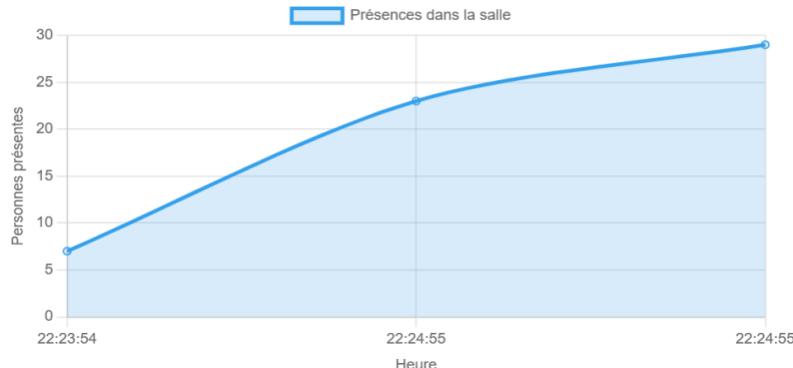
Pour le graphe j'ai rédigé un script Python qui le génère en récupérant le nombre de personnes de la base de données et le temps, puis le dessine en fonction du temps. On prend par exemple la salle C024:

## Détails de la salle : Salle C024

Capacité : 50 personnes

Nombre de personnes présents : 29

### Présence en temps réel



### Etape 4 : Mettre en place le Bluetooth

- Une fois le nombre de personnes dans la salle défini par le arduino, il faut envoyer les données en bluetooth. Pour cela j'utilise le frameWork arduino "SoftwareSerial". Voici la partie bluetooth isolée du reste du programme arduino.

```
#include <SoftwareSerial.h>

// Bluetooth HC-06 sur les pins 6 (RX), 7 (TX)
SoftwareSerial bluetooth(6, 7);

void setup() {
    bluetooth.begin(9600); // Initialisation du module Bluetooth
    bluetooth.println(0); // Envoi initial (par exemple, pour indiquer compteur à 0)
}

void loop() {
    // Exemple d'envoi d'une valeur via Bluetooth (remplacez `compteur` par la variable souhaitée)
    bluetooth.println(compteur);

    // Réception de commandes Bluetooth (optionnel, ici désactivé)
    /*
    if (bluetooth.available()) {
        char cmd = bluetooth.read();
        if (cmd == 'r' || cmd == 'R') {
            compteur = 0;
            bluetooth.println("Compteur réinitialisé à 0.");
        }
    }*/
    delay(50);
}
```

- Il faut ensuite réceptionner ces données avec un programme python, mais malheureusement je n'ai pas pu connecter le module du bluetooth HC-06 avec mon ordinateur. J'ai alors supposé une autre solution, qui est d'envoyer ces données par connexion filaire USB

- Avant d'exécuter le programme python, on doit installer les modules Python `mysql-connector-python` et `serial` par ces deux lignes de commandes:  
`amelx@DESKTOP-GPB2J9P:/var/www/html$ pip3 install mysql-connector-python`  
`amelx@DESKTOP-GPB2J9P:/var/www/html$ pip3 install pyserial`
- D'où changer le programme Arduino en utilisant le SoftwareSerial :

```
Serial.print("COMPTEUR:");
Serial.println(compteur); // Envoi au script Python
```

- Une étape essentielle dans le processus de transmission des données consiste à identifier correctement le nom du port série utilisé par l'Arduino. En effet, selon le système d'exploitation et la connexion utilisée, le nom du port peut varier. Il est donc nécessaire de repérer le bon port (comme `/dev/ttyACM0` sous Linux ou `COM6` sous Windows) afin d'établir une communication fiable entre l'Arduino et le script Python.

### Difficultées rencontrées:

- Le câble USB que j'ai utilisé au début était juste pour l'alimentation, il ne pouvait pas transférer des données.
- Trouver le vrai nom des ports séries.
- Le fait que je travaille sur un PC avec un système Windows mais sur un terminal Ubuntu

Pour conclure cette partie, il est important de noter que l'étape de transmission des données via l'USB de l'Arduino vers le script Python est encore en cours de réalisation. Bien que le programme Arduino soit fonctionnel et que la base de données ainsi que l'interface web soient opérationnelles, l'intégration complète entre le capteur et le site nécessite encore des ajustements, notamment au niveau de la reconnaissance du port série et de la communication entre les deux systèmes. Ce point sera finalisé prochainement afin d'assurer une synchronisation en temps réel des données.

## 5.4 Intégration

Le programme Arduino a dû être implémenté de la partie Bluetooth pour faire communiquer les données du compteur avec la machine sur laquelle est hébergée le site Web. Cette partie est visible plus haut (page 20) dans la partie d'Amel.

Ces lignes permettent d'établir la communication entre le module Bluetooth connecté à la carte Arduino et la machine hôte.

Cette partie-là n'a finalement pas pu être mise en place à cause de problèmes techniques qui ont rendu impossible la communication en Bluetooth.

La solution au problème de communication entre l'Arduino et le script Python a été trouvée en identifiant deux causes principales :

- Premièrement, le câble USB initialement utilisé était un simple câble d'alimentation, incapable de transférer des données. Pour permettre à l'Arduino de transmettre le compteur via le port série à un ordinateur (Ubuntu ou Windows), il est indispensable d'utiliser un **câble USB de type "data"**, conçu pour la transmission à la fois de l'alimentation et des données.
- Deuxièmement, même avec le bon câble, il est essentiel d'**identifier correctement le nom du port série** auquel l'Arduino est connecté. Sur Ubuntu, cela se fait généralement en exécutant la commande `ls /dev/ttyACM*` ou `ls /dev/ttyUSB*` juste après avoir connecté l'Arduino. Le nom exact du port (par exemple `/dev/ttyACM0`) doit ensuite être utilisé dans le script Python pour établir la connexion série.
- Bien que l'Arduino soit correctement connecté à un PC Windows sur le port COM6, Ubuntu ne détectait pas le port série correspondant (comme `/dev/ttyACM0`), ce qui empêchait la réception des données. Ensuite, il a fallu identifier le port série reconnu par Windows (COM6) et adapter le script Python pour l'exécuter directement sous Windows plutôt que dans l'environnement Ubuntu, en utilisant le port `COM6`. Cela a permis de récupérer correctement les données envoyées par l'Arduino. Il est donc essentiel, dans ce type de configuration, de s'assurer d'utiliser un câble USB compatible avec le transfert de données **et** d'identifier précisément le nom du port série pour établir la communication.
- En réunissant ces deux éléments (un câble adapté et l'identification correcte du port) la communication entre l'Arduino et le système a pu être rétablie avec succès.

## 6 Conclusion

Nous avons donc à l'issue de nos travaux produit un système fonctionnel qui permet le comptage des entrées et sorties dans la pièce où il est placé.

L'affichage du nombre de personnes présentes se fait en présentiel grâce aux afficheurs et en reliant le système à une machine, à défaut d'utiliser le Bluetooth pour une communication des informations à distance, puisque nous n'avons pas réussi à faire fonctionner le matériel comme nous le souhaitions.

Nous sommes contents du rendu final du projet mais il pourrait toujours bénéficier de nombreuses potentielles améliorations futures, comme :

- Un mode de transmission de données entre la carte et un site internet sans fil fonctionnel;
- Une mise en veille du système automatique grâce à une photorésistance LDR (Light Dependant Resistor), pour ne pas utiliser la batterie si la salle n'est pas allumée (donc pas occupée);
- Une application en plus du site Web;
- Un bouton pour réinitialiser le compteur sans avoir à éteindre/allumer le système entier;
- Analyse du taux de fréquentation d'une salle selon l'heure

Ce projet a constitué pour nous une nouvelle expérience et nous a permis d'apprendre de nouvelles compétences et en approfondir d'autres.

### Ressentis personnels

#### **Mathis**

J'ai pour ma part pu travailler pour la première fois en développement Arduino et je pense que tout ce que j'ai appris ici pourrait m'être utile pour des projets personnels et professionnels. J'ai beaucoup aimé pouvoir lier le réel au numérique avec les différents éléments techniques du projet et le programme que j'ai écrit.

J'ai également acquis le début d'une nouvelle compétence qui est la conception 3D, très pratique pour la réalisation de maquettes, pour visualiser une idée et bien sûr pour la concrétiser en découpe laser ou impression 3D. Je suis satisfait du rendu réel du projet même si certaines imperfections sont corrigibles, et je suis dans l'ensemble très content d'avoir pu mener à bien ce projet, de la programmation à l'assemblage final.

Je retiens donc de cette expérience l'importance de s'organiser en amont, les différentes compétences que j'ai acquises, ainsi que la nécessité d'avoir une équipe solide et réactive, sans quoi du temps précieux pourrait être perdu et un projet incomplet pourrait être rendu.

Je remercie mon binôme pour avoir créé un site Web fonctionnel et réussi à mettre en place une solution à la suite du problème technique rencontré avec le module Bluetooth que nous devions utiliser.

Je remercie également nos encadrants pour leur dévouement et leur soutien sans quoi notre projet n'aurait pas la forme finale obtenue, ainsi que Logan Villard qui m'a souvent aidé et conseillé dans plusieurs aspects du projet.

## **Amel**

De mon côté, j'ai découvert et approfondi de nombreuses compétences liées à la partie logicielle du projet. J'ai notamment appris à connecter un Arduino à un script Python via le port série, à exploiter des données en temps réel, et à les enregistrer dans une base de données MySQL. J'ai aussi conçu un site web dynamique (HTML et CSS) capable d'afficher ces données en direct grâce à PHP. Cette expérience m'a permis de mieux comprendre l'interconnexion entre le matériel et les services web, ainsi que l'importance de choisir les bons outils (comme un câble USB capable de transmettre les données, ou le bon port série).

Je suis fier d'avoir pu concrétiser cette passerelle entre le monde physique et le numérique, et je suis convaincu que ce savoir-faire me servira dans de futurs projets personnels et professionnels.

Je tiens également à remercier sincèrement mon binôme pour sa collaboration, son sérieux et sa réactivité tout au long du projet. Sa présence a été essentielle à l'avancement du travail. Un grand merci aussi aux enseignants pour leur accompagnement, leurs conseils techniques et leur disponibilité, qui nous ont permis d'aller au bout de cette réalisation avec confiance.