

# **Livrable G: Prototype II**

12-11-2020

GNG1503, Section D00

No. d'équipe : D5

## **Membres de l'équipe:**

- 1) Armand Guigma
- 2) Mohamed Amine Benamara
- 3) Linda Uwase
- 4) Shaun Montebon

## **Table des matières**

<b>Introduction</b>	<b>3</b>
<b>Rencontre #3</b>	<b>3</b>
<b>Plan de prototypage</b>	<b>4</b>
<b>Prototype II</b>	<b>6</b>
<b>Analyse des résultats</b>	<b>9</b>
<b>Fonctionnement</b>	<b>9</b>
<b>Analyse</b>	<b>11</b>
<b>Conclusion</b>	<b>12</b>

## I. Introduction

Ce livrable porte sur la rétroaction du Prototype 1 ainsi que la description et l'analyse de notre Prototype 2. D'une part, il s'agit maintenant de développer le code non simplifié de détection et transmission. D'une autre part, une amélioration doit être portée au prototype I selon la rétroaction de la cliente. Dès lors, l'objectif est de créer les sous-systèmes partiels et enfin, de les rassembler pour finaliser le produit.

## II. Rencontre #3

Le but de la rencontre 3 avec le client est de présenter notre prototype à notre cliente pour obtenir ses commentaires ou sa rétroaction afin de voir les aspects de notre conception que nous devons améliorer ou modifier pour satisfaire notre client. Voici un résumé de cette rencontre :

Après la présentation orale, il a été mentionné qu'il vaut mieux un système qui n'est pas alimenté par des batteries. En effet, l'Université n'a pas les ressources nécessaires à long terme, pour maintenir ou changer des piles pour chaque dispositif installé. Dans le cas d'une batterie, il faut réfléchir à un système de notification en cas de batterie faible. Ensuite, il faut également prendre en compte que s'il y a 2 portes que nos systèmes doivent être inter-reliés. Après, il faut prévoir une alternative aux capteurs wi-fi en cas de pannes de courants ou un problème avec le wi-fi de l'établissement. Enfin, il est important de prendre en compte le sens de l'ouverture des portes, et donc revoir notre placement de base.

### III. Plan de prototypage

Dans cette partie, on va simuler une partie de notre concept, et cette partie consiste à générer un code qui sert à détecter le mouvement d'une personne en sortant ou entrant la salle et afficher le résultat dans un afficheur.

#### A. Objectifs typiques

<i>N° de Test</i>	<i>Objectif du Test</i> (Pourquoi?)	<i>Description du Prototype Utilisé et de la Méthode de Test de Base</i> (Quoi?)	<i>Durée Estimée du Test et Date Prévues du Début du Test</i> (Quand?)
1	Vérifier si le code fonctionne ou s'il y a une erreur de codage ou syntaxe.	Compilation de code dans un compilateur en ligne ou avec un logiciel comme le code blocks.	45 minutes 02/11/2020
2	Vérifier si le code pour détection fonctionne si deux personnes entrent presque au même temps.	Essayer de simuler cette action à l'aide de programme 'Tinkercad'.	30 minutes 03/11/2020
3	Tester la réponse de système lors de détection d'une entrée.	Sur Tinkercad, on fait une simulation de manière que si une personne entre l'afficheur incrémente.	20 minutes 04/11/2020
4	Tester la réponse de système lors de détection d'une sortie.	Sur Tinkercad, on fait une simulation de manière que si une personne entre l'afficheur décrémente.	20 minutes 04/11/2020
5	Vérifier que tous les câbles qui connectent la plaque arduino avec le breadboard sont considéré comme entrée logique.	Lier tous les câbles aux ports numériques de la carte arduino.	30 minutes 05/10/2020

## B. Critères d'arrêt

- Le code détecte l'entrée et la sortie de personne qui passe par le champ de détection.
- Le code n'a aucune faute de syntaxe de manière qu'il soit approuvé par le compilateur.
- Le code affiche le nombre exact de personnes qui sont dans la salle.

## C. Fidélité acceptable

- La fidélité acceptée est moyenne.

## IV. Prototype II

Le code pour la détection et l'affichage est adapté à partir d'un tutoriel sur Youtube dont le lien est le suivant [https://www.youtube.com/watch?v=Q--g8YpXI\\_w](https://www.youtube.com/watch?v=Q--g8YpXI_w). Voici des captures d'écran du code :

```
bon_code_avec_lcd
// include the lcd library code:
#include <LiquidCrystal.h>

// INITIALIZE THE LCD LIBRARY WITH THE NUMBERS OF THE PINS:
LiquidCrystal lcd(0, 1, 2, 3, 4, 5);

//INITIALIZE SENSORS PINS;
int in = 6;
int out = 7;

//INITIALIZE LED PINS:
int led001Pin = 8;
int led01Pin = 9;
int led1Pin = 10;
int led2Pin = 13;

//INITIALIZE COUNTER :
int count = 0;

void setup(){
  //LCD SETTINGS:
  lcd.begin(16,2);
  lcd.print("Visitor Counter");
  delay(2000);

  //DECLARE SENSORS AS INPUT:
  pinMode(in , INPUT);
  pinMode(out, INPUT);

  //DECLARE LED AS OUTPUT:
  pinMode(led001Pin, OUTPUT);
}
```

Figure 1 : Capture d'écran 1 du code.

```

//LED LIGHT OFF/ON:
if(count <= 0){
    lcd.clear();
    digitalWrite(led001Pin, LOW);
    digitalWrite(led01Pin , LOW);
    digitalWrite(led1Pin  , LOW);
    lcd.clear();
    lcd.print("Nobody In Room");
    lcd.setCursor(0,1);
    lcd.print("Light Is Off");
    delay(1000);
}
if(count >= 1 && count <= 5){
    digitalWrite(led001Pin, HIGH);
    digitalWrite(led01Pin , HIGH);
    digitalWrite(led1Pin  , HIGH);
    digitalWrite(led2Pin  , LOW);
}
if(count > 5 && count <= 10){
    digitalWrite(led001Pin, HIGH);
    digitalWrite(led01Pin , HIGH);
    digitalWrite(led1Pin  , HIGH);
    digitalWrite(led2Pin  , HIGH);
}
}
}

```

---

Figure 2: Capture d'écran 2 du code.

```

pinMode(out, INPUT);

//DECLARE LED AS OUTPUT:
pinMode(led001Pin, OUTPUT);
pinMode(led01Pin , OUTPUT);
pinMode(led1Pin  , OUTPUT);
pinMode(led2Pin  , OUTPUT);
}
void loop(){
//SENSORS READING:
if(digitalRead(in)){
    count++;
}
if(digitalRead(out)){
    count--;
}

//LCD DISPLAY :
lcd.clear();
lcd.print("Person In Room:");
lcd.setCursor(0,1);
lcd.print(count);
delay (1000);

//LED LIGHT OFF/ON:
if(count <= 0){
    lcd.clear();
    digitalWrite(led001Pin, LOW);
    digitalWrite(led01Pin , LOW);
    digitalWrite(led1Pin  , LOW);
    lcd.clear();
}
}
}

```

---

Figure 3: Capture d'écran 3 du code.

Le code fonctionne selon le principe suivant. Les capteurs sont définis “in” pour l’entrée et “out “ pour la sortie. Lorsqu’une personne franchit, le premier capteur, et ensuite le second, cela est compté comme une entrée et inversement pour la sortie. L’écran LCD prend en compte l’incréméntation et la décrémentation en fonction des sorties et des entrées. Cependant, vu que nous n’avons pas encore le système physique, nous avons simplifié le code pour l’adapter à la plateforme Tinkercad pour effectuer nos essais. L’écran d’affichage affiche le nombre de personnes actuellement présentes dans la salle. Lorsque la salle est pleine, une LED de couleur rouge s’allume indiquant aux usagers de ne pas entrer. Dans le système final, cela sera remplacé par un système de haut-parleur. Dans le code, nous avons aussi définis des limites de 5 et 10 personnes. Le but est de nous assurer que les commandes désirées lorsque la salle est pleine, sont respectées. Enfin, lorsque la salle est vide, l’écran affiche “nobody in room”.



## V. Analyse des résultats

### a. Fonctionnement

Voici les captures d'écrans du sous-système de détection et comptage.

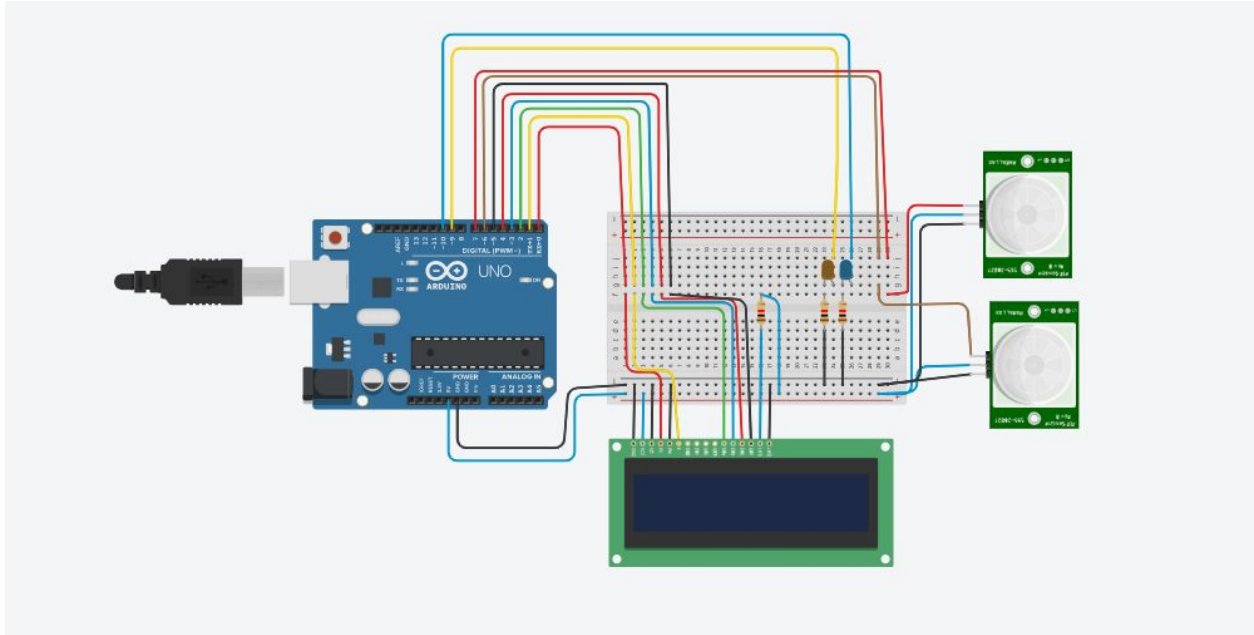


Figure 4: Capture d'écran 1 de la simulation.

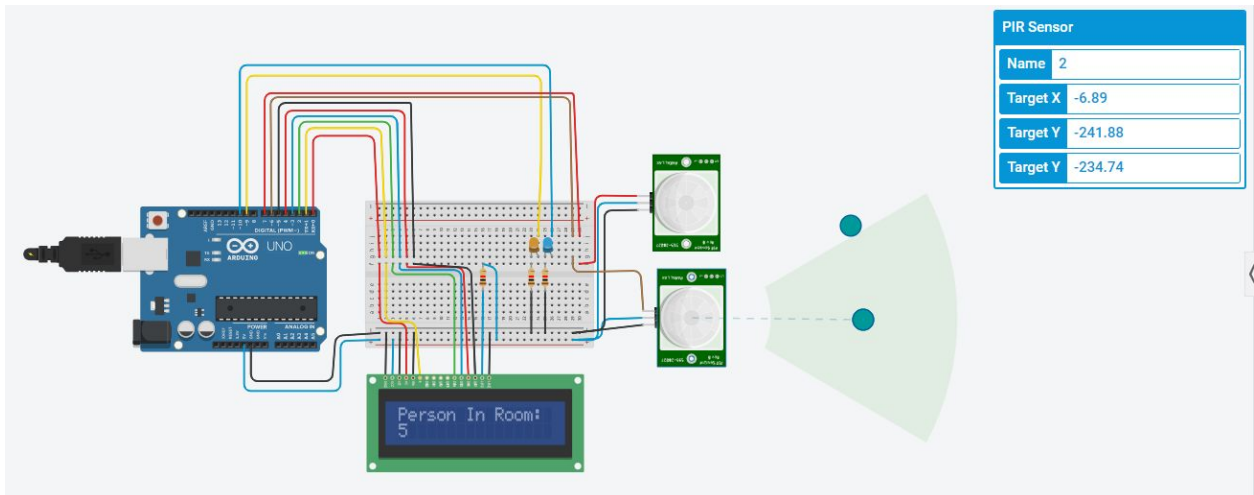


Figure 5: Capture d'écran 2 de la simulation.

Une fois que le capteur en haut détectera un obstacle qui brise le faisceau infrarouge, un signal est envoyé à compter une entrée (+1) dans la salle.

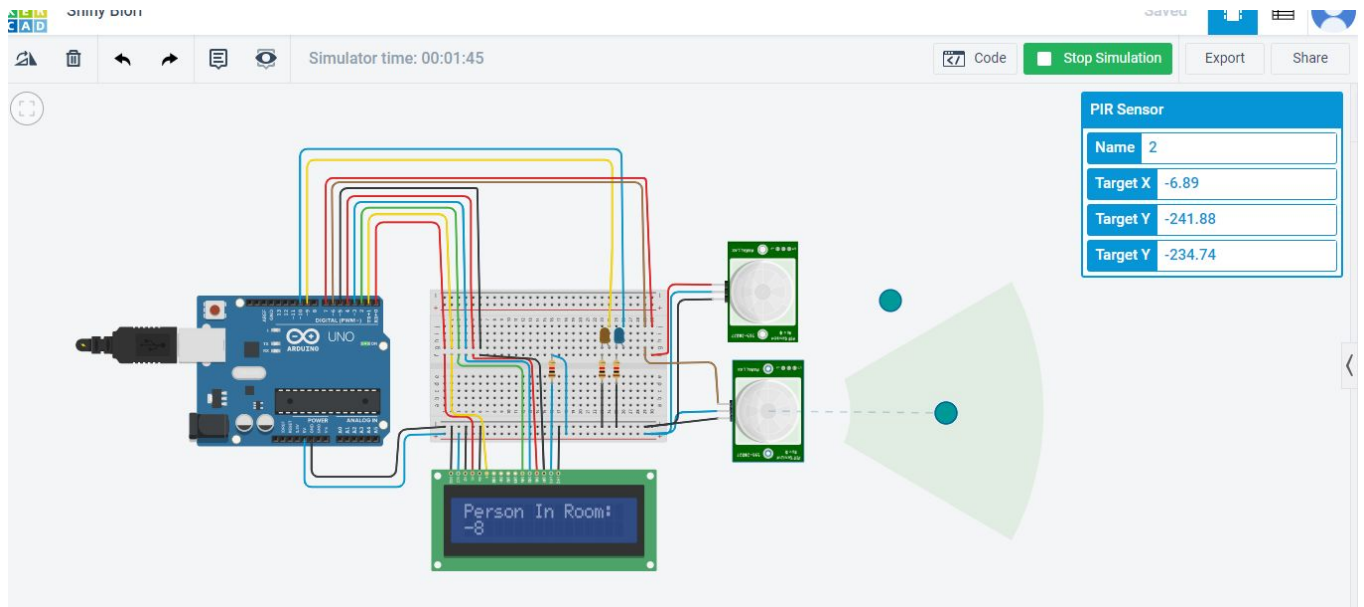


Figure 6: Capture d'écran 3 de la simulation.

L'inverse est vrai lors de la sortie. Une fois que le capteur de sortie (celui en bas) détectera un obstacle, un signal est envoyé pour soustraire une personne (-1) de la salle.

Le compte total effectué par le système est montré à l'afficheur.

## b. Analyse

Notre deuxième prototype se concentre sur l'implémentation du code de détection et affichage ainsi que l'amélioration du système. Nous avons utilisé le programme gratuit Tinkercad pour réaliser nos buts concernant l'assemblage du système de comptage/affichage et le code de ce système. Grâce à cela, nous avons pu nous rapprocher des besoins du client.

Pour le sous-système de détection et comptage, nous avons choisi une configuration avec un compteur bi-directionnel. Ce choix nous permet facilement d'avoir l'interconnectivité entre nos capteurs. Cela signifie que même s'il devait y avoir des obstacles tels qu'une porte ou un panneau, tout ce dont il aurait besoin serait un changement simple dans le code puisque les capteurs à chaque côté des portes sont interreliés par le microcontrôleur.

D'après nos choix nous avons pu remplir les besoins de visibilité et autonomie du système, tout en gardant un coût abordable. Néanmoins, notre cliente a remarqué que l'alimentation par pile n'est pas idéale à long terme. En conséquence, nous avons choisi d'alimenter le système par des fils. Grâce au choix d'un seul microcontrôleur, l'assemblage entier du système pourrait être facilement réorganisé au besoin. Dans le cas d'une réorganisation, le seul facteur limitant serait la longueur des câbles car ils sont de 22AWG.

## VI. Conclusion

Pour conclure, on a achevé une étape importante qui montre notre progression dans le projet, nous avons appris de notre dernière rencontre avec le client que nous devons bien gérer le temps pour notre prochaine présentation. Finalement, il reste la rétroaction du client sur le prototype II. Elle déterminera donc si nous sommes sur le bon chemin.