

GNG1503
Manuel d'Utilisateur du Projet de Conception

Soumis par:

Groupe D2

Abdelli, Mohamed Fadhel 300147804

Beaudoin, Nicolas 300160333

Clarke, Daniel 300132520

El Bitar, Rania

Mckay, Gabrielle 300055550

Le 10 décembre 2020

Université d'Ottawa

Résumé

Ce manuel donne les renseignements nécessaires pour l'installation, le maintien et l'utilisation du système de comptage d'occupants. Il est important de suivre les étapes de ce manuel pour une opération optimale du système et pour éviter des dommages.

La *section 2* du manuel liste tous les composants électroniques, ainsi que les composantes logiciels qui forment le système et explique brièvement leurs fonctionnalités. La quantité nécessaire de chaque composant peut être trouvée dans la *section 3.1.1*, et l'équipement nécessaire est situé dans la *section 3.1.2*.

Les instructions générales pour l'utilisation du système peuvent être trouvées dans la *section 3*, ceci inclut le code C++ adapté pour l'arduino. La *section 4* comprend l'information pour le maintien du système.

Finalement, des conseils pour des travaux futurs sur le système peuvent être trouvés dans la conclusion à la *section 5*.

Table des matières

Introduction	1
Comment le prototype est construit	2
Catégorie	3
LDM (Liste des Matériaux)	3
Liste d'équipements	3
Instructions	4
Figure 1 : Connexion planchette - arduino	4
Figure 2 : Connexion capteurs - LED	5
Figure 3 : Connexion LCD	6
Figure 4 : Connexion haut parleur	7
Comment utiliser le prototype	8
Comment maintenir le prototype	11
Conclusions et recommandations pour les travaux futurs	12

Liste de figures

Figure 1 : Connexion planchette - arduino	4
Figure 2 : Connexion capteurs - LED	5
Figure 3 : Connexion LCD	6
Figure 4 : Connexion haut parleur	7

Liste des acronymes

Acronyme	Définition
PIR	"Passive Infrared" Sensor
LCD	"Liquid Crystal Display"
LED/DEL	"Light emitting Diode" "Diode Electroluminescente"
I2C	"Inter-Intergrated Circuit" - en lien avec le fonctionnement de l'écran LCD

1 Introduction

Les établissements d'éducation se retrouvent dans l'obligation d'offrir certains services en présentiel, notamment les laboratoires, malgré que le reste du campus soit fermé en raison de la pandémie actuelle.

Néanmoins des mesures ont été mises en place, dont un nombre maximum de personnes permis par local. Cette dernière restriction nécessite de la supervision continue. C'est pourquoi notre groupe du cours du Génie de la conception entament une solution à ce problème.

Grâce aux senseurs infrarouge, notre système compte avec une plus grande efficacité dans n'importe quelle conditions. De plus, il est moins coûteux que la plupart des systèmes similaires sur le marché.

Au centre du système est le Arduino Uno, abordable, facile à utiliser et suffisamment puissant pour les besoins du client. Deux senseurs infrarouges permettent de savoir quand un étudiant rentre ou sort, puis, ce dernier affiche directement sur notre dernière composante, qu'est l'écran d'affichage le nombre d'occupants dans la salle et combien il reste avant d'atteindre la limite de cette salle. Ce sous-système fonctionne grâce à une planchette Arduino, qui lui fournit les données nécessaires pour l'affichage.

Ce manuel d'utilisation comporte le processus de fabrication du produit, son fonctionnement, son utilisation et son entretien.

2 Comment le prototype est construit

Pour ce qui est de la partie physique du produit, la pièce centrale est une carte arduino, qui sert à analyser l'information des senseurs et calculer le nombre de personnes. Relié à ce microcontrôleur est de toute évidence les senseurs senseurs, avec leur fil de signal sur une entrée digitale, l'écran, relié à des sorties digitales, et le haut-parleur, relié à une sortie analogique afin de contrôler la fréquence. Les senseurs sont conçues pour avoir un grand champ de vision, ce qui est désavantageux dans notre cas, donc des visières ont été installées de chaque côté. Ceux-ci envoient un signal lorsqu'ils reçoivent de la lumière infrarouge réfléchiée par un objet émettant de la chaleur, comme le corps humain. Ceci à l'avantage d'éliminer plusieurs faux comptes venant d'objets à température de la pièce comme une porte. L'écran LCD a été branché au arduino avec un adaptateur I2C ce qui permet d'avoir seulement 4 fils plutôt qu'une dizaine. Ceci simplifie grandement le circuit et le rend plus facile à reproduire. Cette méthode d'affichage permet clairement d'indiquer le nombre de personnes dans une salle, mais pour les personnes avec une déficience visuelle, un haut parleur qui sonne une alarme si le maximum est dépassé permet de communiquer l'information vitale. Le tout est relié à une source de courant par un fil usb A à usb B puis un adaptateur usb à 120V qui se branche dans le mur.

Le code qui se télécharge sur le arduino a été conçu d'abord sur Tinkercad sur le prototype virtuel 1. La première version du code est donc apparue avant toute version physique du prototype. Cette première version du code fonctionnait en enregistrant le temps auquel un senseur est déclenché avec la fonction millis() du arduino et en comparant lequel des deux temps des senseurs est le plus grand en boucle continue. Celui avec le temps le plus grand est celui déclenché en dernier, voulant donc dire que la personne a fini de ce côté de la porte. Ceci a toutefois mené à quelques problèmes dans l'exécution du code avec les senseurs puisqu'il fallait un délai de environ 1000 millisecondes avant que le code soit capable de détecter une nouvelle personne qui entre ou sort. La raison de cette erreur reste inconnue, mais la théorie actuelle est que les senseurs restent activés pendant un certain nombre de temps et il faut attendre qu'ils se remettent à zéro avant de pouvoir précisément mesurer les temps. C'est pourquoi au prototype 3, l'équipe de conception a complètement recréé le code en évitant d'utiliser la fonction millis() afin que les erreurs de compte ne se reproduisent pas à nouveau. Le nouveau code fonctionne en comparant l'état des senseurs dans une boucle. Il y a 4 états: aucun senseur activé, senseur 1

activé, senseur 2 activé, et les deux senseurs activés. Si l'état passe de 1 senseur activé à deux senseurs activés, dépendamment de quel senseur est activé en premier, il est possible de savoir la direction de la personne passant dans la porte. Ce nouveau code s'est avéré beaucoup plus rapide et efficace et a donc remplacé la version primitive. Le code a aussi vu l'ajout de quelques lignes simples qui permettent d'allumer le haut parleur si la condition est atteinte ou d'afficher sur l'écran le compte au fur et à mesure que les composantes fut ajoutées au prototype physique. Il a aussi été nécessaire d'adapter le code produit sur Tinkercad afin que le prototype physique puisse utiliser l'écran avec la bibliothèque <LiquidCrystal_I2C.h> plutôt que celle <LiquidCrystal.h> puisque le prototype virtuel et physique utilisait des écrans avec des interfaces différentes.

2.1 Catégorie

2.1.1 LDM (Liste des Matériaux)

Pour le produit :

- Écran LCD 20x4 WayInTop, avec adaptateur I2C pré-soudé
 - il est possible de souder les pièces ensemble, mais ceci est coûteux en temps, et peut demander plus de ressources que ne sont disponibles
- Arduino Uno et sa prise
- Mini planchette
- 2 x capteur PIR génériques
- 2 x résistances 220 Ω
- 2 x LED
- Haut parleur Gikfunk
- 11 x fils M-F
- 6 x fils M-M

Pour le boîtier :

- Boîtier conçu par imprimante 3D

2.1.2 Liste d'équipements

- Imprimante 3D pour impression du boîtier
- Ordinateur capable de faire opérer logiciel Arduino afin de le télécharger sur la carte, et de faire télécharger le fichier du boîtier à l'imprimante

- OU : être capable de construire ou modifier le fichier aux besoins de l'utilisateur

2.1.3 Instructions

1. Rassembler toutes les composantes et les diviser comme ci :
 - a. Planchette, Arduino Uno
 - b. Capteurs
 - c. fils M-F et M-M
 - d. Écran LCD et son adaptateur et haut parleur
2. Connecter la planchette à l'Arduino en connectant les fils 5V et GND selon la photo de référence suivante.

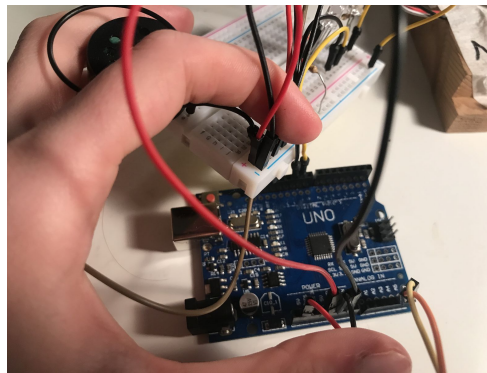


Figure 1 : Connexion planchette - arduino

3. Brancher les capteurs, les résistances 220 Ω , et les deux LEDs selon la photo de référence suivante.
 - a. Vous pouvez faire à ce moment un essai : brancher et télécharger le code final sur la carte Arduino. Quand les capteurs sont activés, les LEDs s'allumeront. La LED de droite correspond au capteur de droite, la LED de gauche au capteur de gauche.

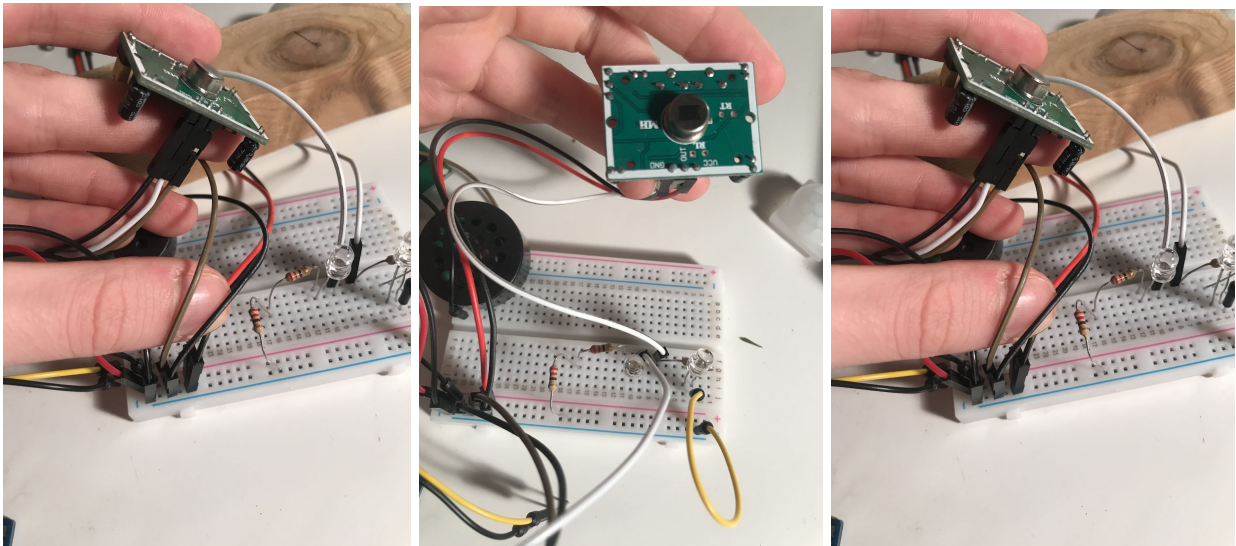
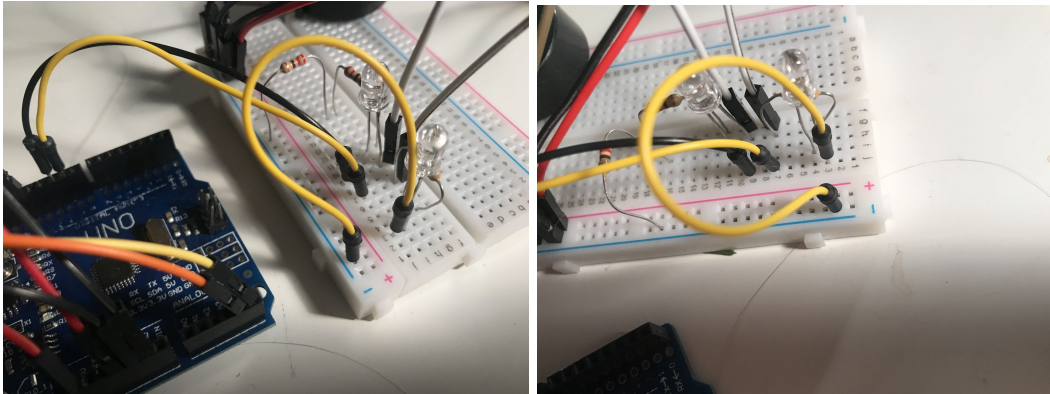


Figure 2 : Connexion capteurs - LED

4. Brancher l'écran LCD par intérim de son adaptateur tel la photo de référence suivante.
 - a. Vous pouvez faire à ce moment un essai : brancher et télécharger le code final sur la carte Arduino. Vous devriez voir apparaître le texte initial "Nombres de pers. ... Max : ..."

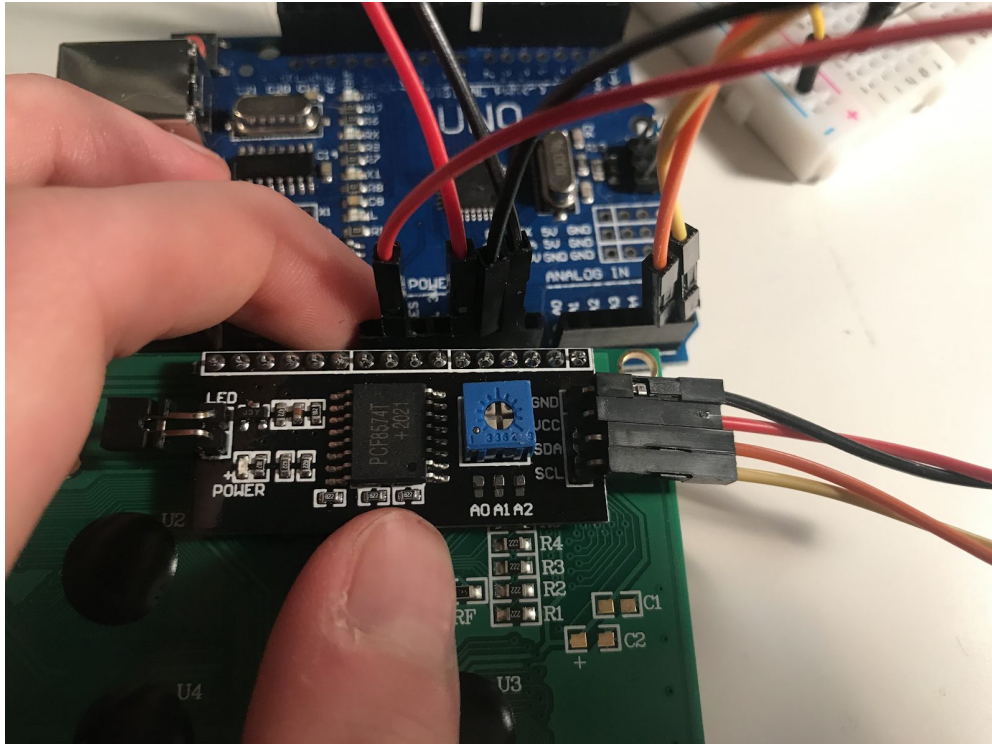


Figure 3 : Connexion LCD

5. Brancher le haut-parleur selon la photo de référence suivante.
 - a. Vous pouvez faire à ce moment un essai : brancher et télécharger le code final sur la carte Arduino. Vous pouvez maintenant soit passer devant le capteur jusqu'à avoir dépassé la limite imposée, soit modifier la limite dans le code Arduino, réinitialiser, et tester "l'alarme" du haut-parleur.

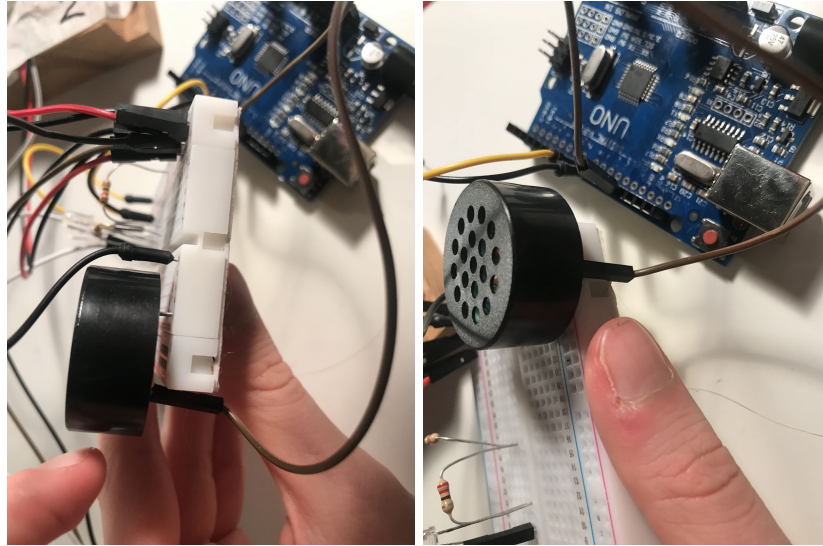


Figure 4 : Connexion haut parleur

3 Comment utiliser le prototype

Pour utiliser le prototype, il faut d'abord le fixer sur le bord du cadre de la porte ou sur un mur du corridor. Une hauteur de 100-130 cm est recommandée.

Par la suite, il faut ajuster le code pour la limite de la salle spécifique. Pour cela, il faut le une version récente du logiciel arduino disponible sur <https://www.arduino.cc/> installé sur un ordinateur portable (un ordinateur fixe est aussi possible, mais le produit doit donc être détaché du mur pour l'adapter). En branchant l'ordinateur à la carte arduino et en ouvrant le fichier code inclut avec le produit, modifiez la variable MAX=20 pour la valeur maximale de personnes désirée. Dans la copie du code ci-dessous, voyez l'endroit où se situe la variable MAX. Il est aussi possible de modifier le message affiché en modifiant les lignes lcd.print(), mais cela nécessite une familiarité avec l'écran. Télécharger le code sur la carte arduino pour finir la présentation. Une fois le système prêt, il suffit de brancher le Arduino à sa source de courant (pile ou prise de courant par l'entremise d'un adaptateur usb à 120V) et en marche.

```
//LCD setup
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);

#define STATE_1 1
#define STATE_2 2
#define STATE_3 3
#define STATE_4 4

int currentState=0;
int lastState=0;
int count=0;
boolean sensor_1_on;
boolean sensor_2_on;
int MAX=20; //Nombre maximal de personnes permis dans une salle

void setup()
{
```

```

Serial.begin(9600);

//LCD screen setup

lcd.init();
lcd.backlight();
lcd.print("Nombre de pers."); //message affiché au-dessus du nombre, limite de 18 caractères.
lcd.setCursor(10,1);
lcd.print("Max. "); // Message affiché devant le maximum, limité à environ 10 caractères.
lcd.print(MAX);

pinMode(9,INPUT); //sensor 2
pinMode(10,INPUT); //sensor 1
pinMode(13,OUTPUT); //buzzer speaker

}

void loop()
{
  sensor_1_on = digitalRead(10) == HIGH;
  sensor_2_on = digitalRead(9) == HIGH;

  //determine current state
  if (!sensor_1_on && !sensor_2_on){
    currentState=STATE_1;
  }

  if (sensor_1_on && !sensor_2_on){
    currentState=STATE_2;
  }

  if (sensor_1_on && sensor_2_on){
    currentState=STATE_3;
  }

  if (!sensor_1_on && sensor_2_on){
    currentState=STATE_4;
  }

  //check for transition
  if(lastState==currentState){
    //do nothing
  }
}

```

```

if (currentState==STATE_1 || lastState==STATE_1){
  //do nothing
}

if (currentState==STATE_3){
  //do nothing
}

if (lastState==STATE_3 && currentState==STATE_4){
  //entering
  count++;

  //Prints count on LCD and serial Monitor
  lcd.setCursor(0,1);
  lcd.print(" ");
  lcd.setCursor(0,1);
  lcd.print(count); //Imprime le nombre de personnes dans la salle
  Serial.println(count);
}

if (lastState==STATE_3 && currentState==STATE_2){
  //exiting
  count--;

  //Prints count on LCD and serial Monitor
  lcd.setCursor(0,1);
  lcd.print(" ");
  lcd.setCursor(0,1);
  lcd.print(count); //Imprime le nombre de personnes dans la salle
  Serial.println(count);
}

//Turns on alarm if MAX in exceeded
if (count>MAX){
  tone(13, 1000);
}

if (count<=MAX) {
  noTone(13);
}
lastState=currentState;
delay(100);
}

```

4 Comment maintenir le prototype

Ce produit nécessite très peu d'entretien. Si la source de courant est une pile, il faut périodiquement la recharger/remplacer. Sinon, il suffit de vérifier que le compte est à jour (au cas où une personne aurait volontairement activé le système sans entrer) et de nettoyer l'écran de poussière visible afin que l'affichage reste clair. Il faut aussi éviter de soumettre le système à des chocs mécaniques et thermiques importants ou à de l'eau dans son circuit puisque le boîtier et les composants ne sont pas conçus pour résister à ce type d'abus.

Les composants sont toutes conçues par différents fabricants, donc il est difficile de prédire leur fiabilité à long terme, mais l'équipe de conception du produit suppose que l'écran risque de faillir en premier puisqu'il doit fonctionner en tout temps et est généralement plus fragile que le arduino ou les senseurs.

5 Conclusions et recommandations pour les travaux futurs

Suite à la finalisation du projet, l'équipe de conception a retenue quelques leçons. La première est que les senseurs PIR sont très efficaces pour détecter la présence d'un animal dans des situations nécessitant peu de précision, mais pour tenir un compte précis, ils ont tendance à être imprévisibles. Deuxièmement, l'affichage sur un simple petit écran LCD n'est pas exactement idéal pour l'aspect visuel. Une application pour téléphone intelligent serait utile pour pouvoir afficher l'information plus clairement et régler les paramètres plus facilement. Finalement, l'organisation et la communication à distance pour concevoir un produit est loin d'être idéal, donc si un nouveau produit est à concevoir, il est grandement recommandé d'avoir au moins quelques rencontres en personnes avec le prototype.

APPENDICES

APPENDICE I: Fichiers de conception

Lien pour la page Makerrapo:

<https://makerepo.com/Nic/dispositif-de-comptage-de-personnes-quipe-d2-gng1503>

APPENDICE II: Autres Appendices

Code final du système Arduino:

```
//LCD setup

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);

#define STATE_1 1

#define STATE_2 2

#define STATE_3 3

#define STATE_4 4

int currentState=0;

int lastState=0;

int count=0;

boolean sensor_1_on;

boolean sensor_2_on;

int MAX=20;      //Maximum number of people allowed in a room

void setup()

{

    Serial.begin(9600);
```

```
//LCD screen setup

lcd.init();

lcd.backlight();

lcd.print("Nombre de pers.:");

lcd.setCursor(10,1);

lcd.print("Max. " );

lcd.print(MAX);

pinMode(9,INPUT); //sensor 2

pinMode(10,INPUT); //sensor 1

pinMode(13,OUTPUT); //buzzer speaker

}

void loop()

{

  sensor_1_on = digitalRead(10) == HIGH;

  sensor_2_on = digitalRead(9) == HIGH;

  //determine current state
```

```
if (!sensor_1_on && !sensor_2_on){
    currentState=STATE_1;
}

if (sensor_1_on && !sensor_2_on){
    currentState=STATE_2;
}

if (sensor_1_on && sensor_2_on){
    currentState=STATE_3;
}

if (!sensor_1_on && sensor_2_on){
    currentState=STATE_4;
}

//check for transition
if(lastState==currentState){
    //do nothing
}

if (currentState==STATE_1 || lastState==STATE_1){
    //do nothing
}
```

```
}
```

```
if (currentState==STATE_3){
```

```
    //do nothing
```

```
}
```

```
if (lastState==STATE_3 && currentState==STATE_4){
```

```
    //entering
```

```
    count++;
```

```
    //Prints count on LCD and serial Monitor
```

```
    lcd.setCursor(0,1);
```

```
    lcd.print("  ");
```

```
    lcd.setCursor(0,1);
```

```
    lcd.print(count);
```

```
    Serial.println(count);
```

```
}
```

```
if (lastState==STATE_3 && currentState==STATE_2){
```

```
    //exiting
```

```
    count--;
```

```
    //Prints count on LCD and serial Monitor
```

```
lcd.setCursor(0,1);  
lcd.print("  ");  
lcd.setCursor(0,1);  
lcd.print(count);  
Serial.println(count);  
}
```

```
//Turns on alarm if MAX in exceeded
```

```
if (count>MAX){  
    tone(13, 1000);  
}
```

```
if (count<=MAX) {  
    noTone(13);  
}
```

```
lastState=currentState;  
delay(100);
```

```
}
```