Projet NSI – TAJ'indoor-Alarme

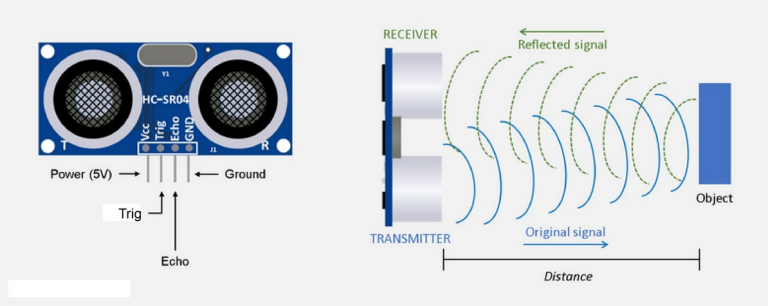
Auteurs

Arthur Balay

Thomas Chassanis Pons
Jérémy Brunel

Ce que je veux faire
Créer une alarme, pour sécurisé de manière simple une pièce

Description fonctionnelle
Ultrasonic sensor



Si une personne passe trop près de capteur ultrasonique déclenchera l'alarme.

On inclut la librairie du capteur pour l'utilisé avec grove.

```
long_ultrasonic = ultrasonic.MeasureInCentimeters();
Serial.println(long_ultrasonic);
```

On récupère la longueur que sur trouve l'objet et/ou la personne

Buzzer

Le buzzer représente l'alarme dans notre projet.

```
tone(buzzer, 100, 100);
```

La constante «buzzer » est le port où le buzzer est brancher, le premier « 100 » c'est la fréquence en KHz Du son et pour finir le temps de la note

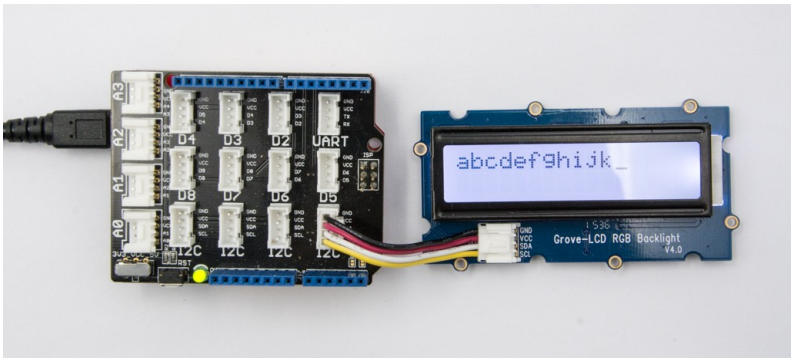
Bouton

Le bouton sert à allumer ou éteindre l'alarme.

```
state_button = digitalRead(buttonPin);
```

On regarde ici la position du bouton, si il est appuyé ou relâché

Écran Grove



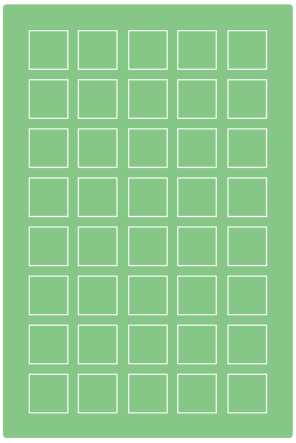
Chaque pixels de l'écran est allumable, ce qui permet de créer des forme.
Librairie :

```
#include "rgb_lcd.h"
```

```
lcd.setRGB(59, 199, 201);
lcd.print("Lancement de TAJ");
```

ici l'écran sera bleu-gris, et affichera
« Lancement de TAJ »

Pixel de l'écran



Exemple de code commenté

```
#include <TimerOne.h> // librérie pour créer une boucle en plus

#include <Wire.h> // dépendance de la librérie rgb_lcd.h
#include "rgb_lcd.h" // librérie pour faire fonctionner l'écran

#include "Ultrasonic.h" // librérie pour faire fonctionner le capteur ultrasonique

rgb_lcd lcd; // on défini la variable "lcd" sur la librérie rgb_lcd

Ultrasonic ultrasonic(2); // défini la où est branché le capteur ultrasonique

const int buzzer = 4; // défini la où est branché le buzzer
const int buttonPin = 5; // défini la où est branché le bouton
```

```
# Import de la librairie micro:bit

# on crée une image a afficher

# Boucle infinie
# si le bouton A à été préssé
# on éteint toutes les leds
# on attribue la valeur 1 à la variable num_ligne
# on boucle num_colonne de 0 à 4
# on éteint le pixel situé à (num_colonne, num_ligne)
# on allume ce même pixel à intensité 7
# on attent 0,1 seconde

# on boucle de 4 à 0
# on éteint le pixel situé à (num_colonne, num_ligne)
# on attend 0,1 seconde

# si le bouton B à été préssé
# on éteint toutes les leds
# on affiche l'image pendant 3 secondes et on l'efface
```

```
// Custom Character en byte :
byte valid[] = {
 B00000,
 B00001,
 B00010,
 B10100,
 B01000,
 B00000,
 B00000,
 B00000
};

byte not_valid[] = {
 B00000,
 B10001,
 B01010,
 B00100,
 B01010,
 B10001,
 B00000,
 B00000
};
////////////////////////////
void setup() {
 Timer1.initialize(100000); // on définie la
boucle "Timer1", qui s'actualise toutes les
100000 microseconds
 Timer1.attachInterrupt(bouton_boucle); //
on attache la boucle "Timer1" à la fonction
"bouton_boucle"

 Serial.begin(9600);
 lcd.begin(16, 3); // on définie l'écran sur
du 16x3 soit 48 pixels
 pinMode(buttonPin,INPUT); // buttonPin sur
INPUT
 pinMode(buzzer, OUTPUT); // buzzer sur
OUTPUT


 lcd.createChar(0, valid); // custom
Character 1
 lcd.createChar(1, not_valid); // custom
Character 2

 Serial.println("Lancement de TAJ !"); //
print dans la console

 // TAJ STARTING
 lcd.clear(); // clear l'écran
```

```
 lcd.setRGB(59, 199, 201); // set la couleur
de l'écran
 lcd.print("Lancement de TAJ"); // on
affiche des mots sur l'écran
 delay(1000); // on attent 1s
 lcd.clear(); // clear l'écran
 // charchement
 for(int i=0; i<16; i++){
  lcd.setCursor(i, 0);
  lcd.write(".");
  delay(75);
 }
 // fin charchement
 lcd.clear(); // clear l'écran
 lcd.setRGB(0, 255, 0); // set la couleur de
l'écran
 lcd.print("Alarme : "); // écrit sur
l'écran
 lcd.write((unsigned char)1); // et ajoute
le Character 1

 delay(1000); // on attent 1s
}

int long_ultrasonic; // défini var
long_ultrasonic

bool state_button; // défini var
state_button
bool button_press=false; // défini var
button_press

bool timer_1 = false; // défini var timer_1
bool alarme_interface = false; // défini var
alarme_interface
bool sound = false; // défini var sound

////////////////////////////////////////////
//////////////////////

void bouton_boucle(){
 state_button = digitalRead(buttonPin); //
récupére si le bouton est press

 if(state_button){
  button_press=true; // met button_press sur
true
 }

 if(button_press){
```

```
  if(state_button==0){

    button_press=false; // met button_press
sur false

    if(sound){ // condition si sound est sur
true on met sound sur false
     sound = false;
    }

    if(alarme_interface==false){ // condition
si alarme_interface est sur false on met
timer_1 sur true sinon sur false
     timer_1=true;
    }else{
     timer_1=false;
    }

    alarme_interface=not alarme_interface;
//alarme_interface est égale a son inverse
"si il est sur true on le met sur false"

   }
  }
}

////////////////////////////////////////////
//////////////////////

void loop() { // la boucle

 long_ultrasonic =
ultrasonic.MeasureInCentimeters(); // on
récupére la distance du capteur ultrasonique
 Serial.println(long_ultrasonic); // on la
print

 if(timer_1){
  long_ultrasonic=520; // pour corrigé un
beug quand la valeur et mise à 0
  int temps=10; // temps d'attente
  for(int i=0; i<10; i++){ // boucle for qui
nous fait attendre 10s avant d'activé
l'alarme et vérifie si elle est pas annulé

    if(timer_1==false){
     Serial.println("annulation");
     break;
    }
```

```
    lcd.clear();
    lcd.setCursor(8,0);
    lcd.print(temps);
    temps=temps-1;
    delay(1000);
   }

   timer_1=false;
  }

  if(alarme_interface){ // si
alarme_interface est sur true on affiche
l'acran en rouche avec le Character 2

   lcd.clear();
   lcd.setRGB(255, 0, 0); // rouge
   lcd.print("Alarme : "); // écrit sur
l'écran
   lcd.write((unsigned char)0); // Character
2

   if(long_ultrasonic<50 and sound==false)
{ // si long_ultrasonic est inférieur à 50
et que l'arme n'est pas entrain de sonné on
attend 10s et on fait sonné l'alarme
    int temps=10;
    for(int i=0; i<10; i++){
     if(alarme_interface==false){
      Serial.println("annulation");
      break;
     }
     lcd.clear();
     lcd.setCursor(8,0);
     lcd.print(temps);
     temps=temps-1;
     delay(1000);
    }
    if(alarme_interface){
     sound = true;
    }
   }
  }else{
   lcd.clear();
   lcd.setRGB(0, 255, 0);
   lcd.print("Alarme : ");
   lcd.write((unsigned char)1);
  }

  if(sound){ // si sound est égale à true
alors on fait sonné l'alarme
```

```
    tone(buzzer, 100, 100); // on fait sonné
le buzzer
    
    lcd.clear(); // clear l'écran
    lcd.setRGB(0, 0, 255); // couleur de
l'écran
    lcd.setCursor(4, 0); // on déplace le
curseur sur l'écran
    lcd.print("Attention"); // on écrit sur
l'écran
    
    delay(200); // on attent 0,2s
    
    tone(buzzer, 200); // on fait sonné le
buzzer
    
    lcd.clear();
    lcd.setRGB(255, 0, 0); // couleur de
l'écran
    lcd.setCursor(5, 0);
    lcd.print("Police"); // on écrit sur
l'écran
    lcd.setCursor(7, 1); // on déplace le
curseur sur l'écran
    lcd.print("!!"); // on écrit sur l'écran
    
    delay(200); // on attent 0,2s
    noTone(buzzer); // on éteint le buzzer
  }
  
  delay(100); // on attent 0,1s
}

////////////////////////////////////////////
//////////////////////
```