

Rapport du Projet Final

Cours : INF1643 Architecture des ordinateurs II

Session : Hiver 2020

Membres du groupe :

- Saredo Moussa Omar (MOUS27569504)
- Rodrigue Seka (SEKR26058506)
- Abdoulhafid Daher (OUMA10019005)
- Ruth Bernice Obonga (OBOR07609909)

Intitulé du sujet : Système d'alarme Arduino

Introduction

Nous voulons réaliser un système d'alarme en utilisant la carte Arduino. Comment voulons-nous que ce système réagisse ?

Ce système détectera le mouvement de tout élément étranger à sa proximité. Nous utiliserons donc un capteur de distance pour relever tout mouvement. Pour se faire, nous comptons utiliser le «Ultrasonic Sensor».

Aussi, ce système d'alarme va détecter toute infraction. Une fois que le système est armé, il est désormais actif pour trahir ou révéler toute transgression à sa sécurité. On voudrait donc incorporer dans notre système un capteur de son pour notifier les infractions.

Description de l'environnement de programmation

Nous utiliserons une carte Arduino pour la conception de ce système d'alarme. Il est donc tout naturel pour nous et plus adapté pour notre code d'utiliser l'environnement de programmation Arduino.

Description du projet

Notre projet consiste à créer un système d'alarme Arduino. Ce projet a déjà été réalisé par un internaute. Nous avons repris le projet et y avons ajouter quelques modifications. Voici en générale comment notre réalisation va fonctionner :

Tout d'abord, lorsque l'on appuie sur le bouton A, l'alarme s'active en 10 seconde. Ensuite, on utilise un capteur à ultrasons pour détecter des objets. Aussitôt que l'alarme détecte des objets, une sonnerie commence à émettre un son. Pour l'éteindre, il suffit de composer un mot de passe à 4 chiffres. Un mot de passe par défaut a été préalablement défini (1234)

mais il est possible de le changer; pour cela il faut appuyer sur le bouton B, qui nous permet d'accéder au menu ou il va d'abord falloir entrer le mot de passe actuel avant de continuer et enregistrer un nouveau mot de passe à 4 chiffres.

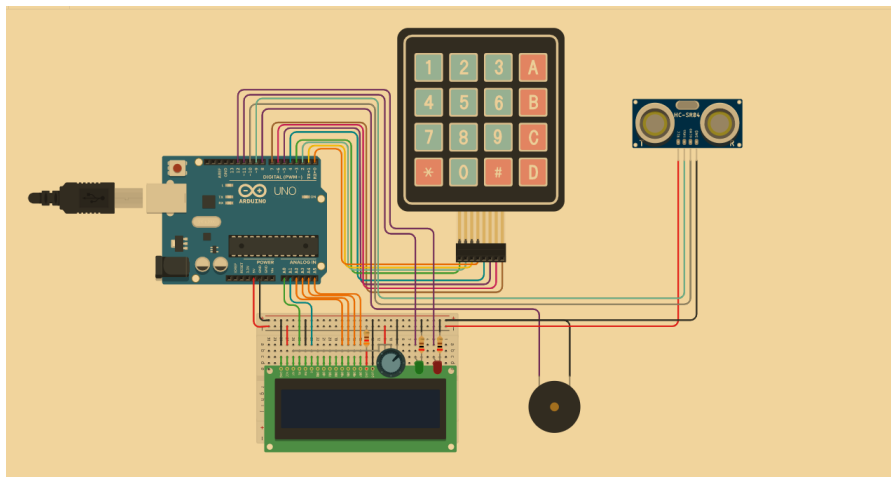
Une fois le mot de passe modifié, lors de la prochaine activation de l'alarme, celle-ci ne pourra être arrêtée qu'en entrant le nouveau mot de passe. Si un mauvais mot de passe est entré, un message d'erreur invitant à réessayer est alors affiché. Nous avons aussi des LED rouge et verte qui fonctionnent selon la situation. Quand l'alarme est active, c'est la LED verte qui clignote et lorsqu'elle se déclenche, c'est la LED rouge qui clignote.

Matériel nécessaire pour la conception

Voici un inventaire du matériel dont nous aurons besoin pour mettre en place le système d'alarme :

- Carte Arduino
- Keypad : clavier pour composer le mot de passe
- Un écran LCD 16×02
- Ultrasonic Sensor : détecteur de son
- Buzzer : pour la notification son
- Platine d'essai
- Fils de connections
- Source d'alimentation

Schema



1- Carte Arduino :

La carte Arduino est la principale composante de notre projet, elle sert à la programmation des divers événements de notre système d'alarme. Elle est branchée à la platine ainsi toutes les autres matériels

2- Keypad:

Un clavier est un ensemble de boutons disposés en bloc ou "pavé" qui portent des chiffres, des symboles ou des lettres alphabétiques.

3- Ultrasonic Sensor:

La tête du capteur émet une onde ultrasonore et reçoit l'onde réfléchiée par la cible. Les capteurs à ultrasons mesurent la distance de la cible en mesurant le temps s'écoulant entre l'émission et la réception du signal sonore.

4- Buzzer:

Un capteur piézoélectrique est un appareil qui utilise l'effet piézoélectrique pour mesurer les changements de pression, d'accélération, de température, de déformation ou de force en les convertissant en une charge électrique. Le préfixe piezo- vient du grec pour « appuyer » ou « presser ».

5- Fils de connections :

Support de connexion

6- L'écran LCD:

L'afficheur LCD est directement connecté à la carte Arduino. Il indique ce que la carte Arduino lui demande. Pour notre projet, elle affiche le message active ou changer le mot de passe par défaut du système l'alarme en tapant les touches l'alphanumérique.

7- LED rouge et vert :

Les LED rouges et vertes sont branchées de la même façon que le haut-parleur. Elles sont donc contrôlées par la carte Arduino. Celui-ci allume à chaque fois que le haut-parleur produit un son et s'éteint s'il n'y a plus de son.

Commenté [OB1]: De quelles composantes ici ? L'Arduino, les LED ??

8- Résistance :

La résistance permet de réguler le courant qui entre dans la LED, car celle-ci ne peut prendre le 5V que lui fournit la carte Arduino.

9- Platine d'essai :

La platine d'essai est la plaquette blanche servant à la création d'un circuit électrique. Elle est alimentée par la carte Arduino. Le LED rouge, vert et le petit

haut-parleur sont connectés à cette plaquette. C'est grâce à cette plaque que nous pouvons contrôler à quel moment la lumière et le haut-parleur sont en fonction.

Le code

```
#include <LiquidCrystal.h> // inclut la bibliothèque LiquidCrystal
#include <Keypad.h>
//definition des buzzer
#define buzzer 8
#define trigPin 9
#define echoPin 10
#define LedPin1 12
#define LedPin2 11
//definition de variables
longue durée;
int distance, initialDistance, currentDistance, i;
int screenOffMsg =0;
String password="0000";
String tempPassword;
boolean activated = false; // Etat de l'alarme
boolean isActivated;
boolean activateAlarm = false;
boolean alarmActivated = false;
boolean enteredPassword; // Etat du mot de passe saisi pour arrêter l'alarme
boolean passChangeMode = false;
boolean passChanged = false;
const byte ROWS = 4; // quatre rangées.
const byte COLS = 4; // quatre colonnes.
char keypressed; // caractère tapé
// définir les symboles sur les boutons des claviers
char keyMap[ROWS][COLS] = {
```

```

    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte rowPins[ROWS] = { 3, 2, 1, 0}; // Brochage des lignes du clavier
byte colPins[COLS] = { 4, 5, 6, 7}; // Brochage des colonnes du clavier
Keypad myKeypad = Keypad( makeKeymap(keyMap), rowPins, colPins, ROWS, COLS);
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5); // Crée un objet LC. Paramètres: (rs, enable, d4,
d5, d6, d7)

void setup() {
    lcd.begin(16,2);
    pinMode(buzzer, OUTPUT); // Définir le buzzer comme sortie
    pinMode(trigPin, OUTPUT); // Définit le trigPin comme une sortie
    pinMode(echoPin, INPUT); // Définit echoPin comme entrée
    pinMode(LedPin1, OUTPUT); //Définir LedPin ccomme sortie
    pinMode(LedPin2, OUTPUT); //Définir LedPin2 comme entrée
}

//configurations

void loop() {

/*activation de l'alarme et en utilisant une boucle while, un compt à rebours de 3secondes
avant l'alarme soit activé.

*/

    if (activateAlarm) {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("L'alarme sera");
        lcd.setCursor(0,1);
        lcd.print("activé dans ");
    }
}

```

```

int countdown = 3; // Compte à rebours de 3 secondes avant d'activer l'alarme
while (countdown != 0) {
    lcd.setCursor(13,1);
    lcd.print(countdown);
    countdown--;
    tone(buzzer, 700, 100);
    delay(1000);
}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Alarm Activated!");
initialDistance = getDistance();
activateAlarm = false;
alarmActivated = true;
}
// verification en permanence de la distance par le cpteur à ultrason.

```

```

if (alarmActivated == true){
    currentDistance = getDistance() + 10;
    if ( currentDistance < initialDistance) {
        tone(buzzer, 1000); // Envoi d'un signal sonore à 1 KHz
        digitalWrite(LedPin1,HIGH);
        digitalWrite(LedPin2,LOW);
        lcd.clear();
        enterPassword();
    }
}

```

```

// Impression de message indiquant que l'alarme est activé.
if (!alarmActivated) {
  if (screenOffMsg == 0 ){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("A - Activate");
    lcd.setCursor(0,1);
    lcd.print("B - Change Pass");
    screenOffMsg = 1;
  }
  keypressed = myKeypad.getKey();
  if (keypressed == 'A'){    //Si A est enfoncé, active l'alarme
    tone(buzzer, 1000, 200);
    activateAlarm = true;
    digitalWrite(LedPin1, LOW); /
    digitalWrite(LedPin2, HIGH);
  }
  //vérification de la conformité du mot de pass
  else if (keypressed == 'B') {
    lcd.clear();
    int i=1;
    tone(buzzer, 2000, 100);
    tempPassword = "";
    lcd.setCursor(0,0);
    lcd.print("Current Password");
    lcd.setCursor(0,1);
    lcd.print(">");
    passChangeMode = true;
  }
}

```

```

passChanged = true;
while(passChanged) {
keypressed = myKeypad.getKey();
if (keypressed != NO_KEY){
    if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
        keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
        keypressed == '8' || keypressed == '9') {
        tempPassword += keypressed;
        lcd.setCursor(i,1);
        lcd.print("*");
        i++;
        tone(buzzer, 2000, 100);
    }
}
if (i > 5 || keypressed == '#') {
    tempPassword = "";
    i=1;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Current Password");
    lcd.setCursor(0,1);
    lcd.print(">");
}
if ( keypressed == '*') {
    i=1;
    tone(buzzer, 2000, 100);
    if (password == tempPassword) {
        tempPassword="";
    }
}
}

```



```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Set New Password");
lcd.setCursor(0,1);
lcd.print(">");
while(passChangeMode) {
  keypressed = myKeypad.getKey();
  if (keypressed != NO_KEY){
    if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||
        keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
        keypressed == '8' || keypressed == '9') {
      tempPassword += keypressed;
      lcd.setCursor(i,1);
      lcd.print("*");
      i++;
      tone(buzzer, 2000, 100);
    }
  }
  if (i > 5 || keypressed == '#') {
    tempPassword = "";
    i=1;
    tone(buzzer, 2000, 100);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Définir un nouveau mot de pass");
    lcd.setCursor(0,1);
    lcd.print(">");
  }
}

```

```

        if ( keypressed == '*' ) {
            i=1;
            tone(buzzer, 2000, 100);
            password = tempPassword;
            passChangeMode = false;
            passChanged = false;
            screenOffMsg = 0;
        }
    }
}
}
}
}
}
}
}

void enterPassword() {
    int k=5;
    tempPassword = "";
    activated = true;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" *** ALARM *** ");
    lcd.setCursor(0,1);
    lcd.print("Pass>");
    while(activated) {
        keypressed = myKeypad.getKey();
        if (keypressed != NO_KEY){
            if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '3' ||

```

```

        keypressed == '4' || keypressed == '5' || keypressed == '6' || keypressed == '7' ||
        keypressed == '8' || keypressed == '9' ) {
    tempPassword += keypressed;
    lcd.setCursor(k,1);
    lcd.print("*");
    k++;
}
}
if (k > 9 || keypressed == '#') {
    tempPassword = "";
    k=5;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" *** ALARM *** ");
    lcd.setCursor(0,1);
    lcd.print("Pass>");
}
if ( keypressed == '*' ) {
    if ( tempPassword == password ) {
        activated = false;
        alarmActivated = false;
        noTone(buzzer);
        digitalWrite(LedPin1, LOW);
        digitalWrite(LedPin2, LOW);
        screenOffMsg = 0;
    }
    else if (tempPassword != password) {
        lcd.setCursor(0,1);

```

```

        lcd.print("Wrong! Try Again");
        delay(2000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" *** ALARM *** ");
        lcd.setCursor(0,1);
        lcd.print("Pass>");
    }
}
}

// Fonction personnalisée pour le capteur à ultrasons
long getDistance(){
    //int i=10;
    //while( i<=10 ) {
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Définit le trigPin sur l'état HIGH pendant 10 micro secondes
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    distance = duration*0.034/2;
    //sumDistance += distance;
    //}
}

```

```
//int averageDistance= sumDistance/10;  
return distance;  
}
```

L'assemblage

Notre projet a été conçu selon les notions vues en classe et des recherches effectuées. Nous avons pu l'assembler dans le site de simulation comme sur l'image. La photo plus haut montre les différents composants du projet et la façon avec laquelle ils sont assemblés. Et voici quelques captures d'écran du résultat obtenu après la mise en place.

Conclusion

Enfin, nous avons pu réaliser notre projet sur le site du simulateur de tinkercad. Malgré les changements survenus dernièrement à cause de la situation sanitaire actuelle, nous avons pu avec la vidéo vous montrer les différentes étapes de lors de l'utilisation et de la mise en place du projet. Le système d'alarme est fonctionnel. Ainsi donc se termine notre projet.

Documentation

- Nous nous sommes procuré le code d'un projet qui a été réalisé il y a quelques années par *Arduino Security and Alarm System Project*. Ce projet a été conçu par Dejan Nedelkovski.
Source : <https://howtomechatronics.com/projects>
- Manuel de laboratoire: programmation en langage C sur l'Arduino (lecture du cours)