EXAMEN TP

Système de détection des obstacles avec Arduino

BlidaouiAida

BelMabroukBacem

MPSE 2

GROUPE A

But de ce projet:

Dans ce projet on va réaliser un système de détection des obstacles avec Arduino. Ce système utilise principalement un **capteur à ultrasons, buzzer et une LED**.

Lorsque le capteur à ultrason détecte un obstacle à une distance < 3cm, la carte Arduino donne l'ordre au buzzer de sonner et la LED rouge de s'allumer.

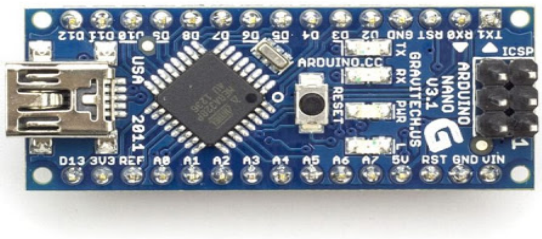
Composants nécessaires :

- **prise mini-USB**

Fonctionnement

Le câble USB connecter l'Arduino avec l'ordinateur. Cela permet à la fois de téléverser (charger un nouveau programme) et d'alimenter l'Arduino.

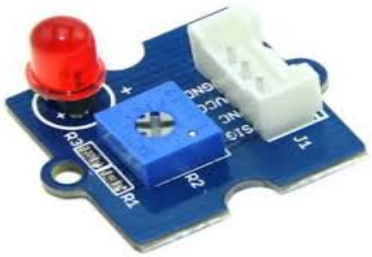
Carte Arduino NANO



Fonctionnement

L'Arduino Nano intègre toutes les fonctionnalités électroniques qui permettent de réaliser des travaux de programmation sans difficulté, mais aussi d'utiliser un microcontrôleur intégré. Pour cela, il est juste nécessaire de relier la carte à un PC à l'aide d'un câble USB.

- **1 LED rouge**

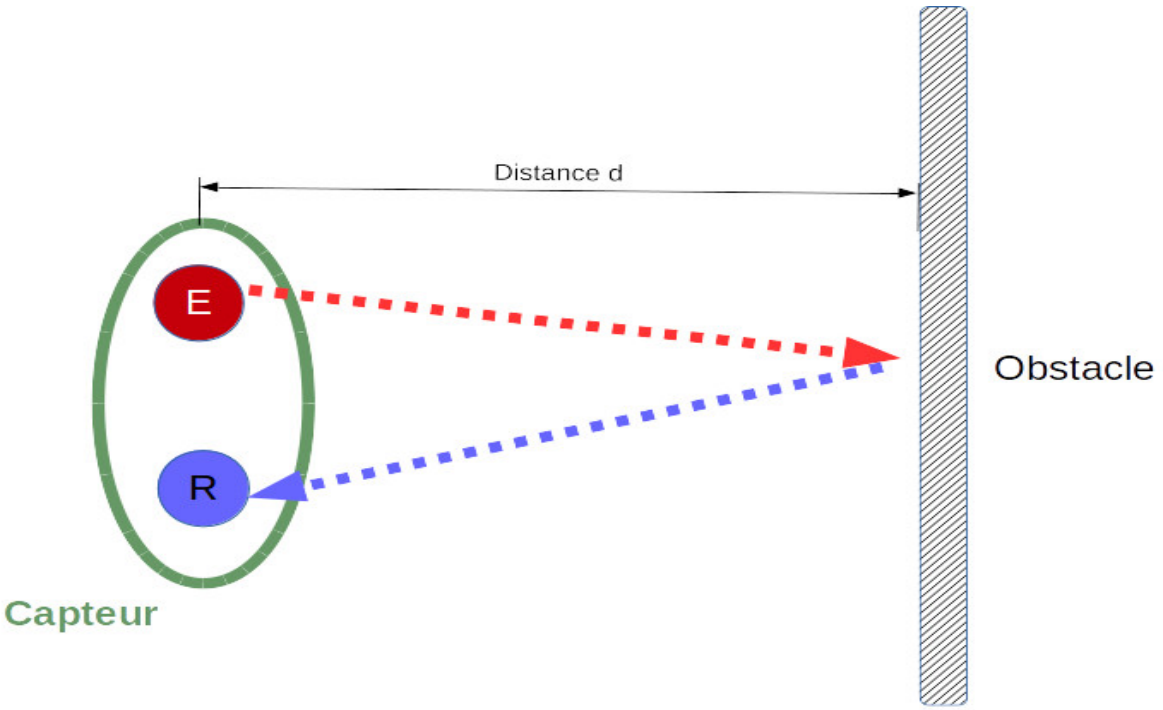


- **Un détecteur à ultrason sonore**

Fonctionnement

Le détecteur HC-SR04 utilise les ultrasons pour déterminer la distance à laquelle se trouve un objet. Peu importe l'intensité de la lumière, la température ou le type de matière, le capteur pourra facilement détecter s'il y a un obstacle devant lui. Tout de fois, il peut être contraint sur certains types de couleurs tel que le noir (contraste), ou encore sur la matière comme le textile. Son champ de vision est de 90° environ selon l'environnement. Si une impulsion de plus de 10µS et détecter, alors le

capteur envoie une série de 8 impulsions à ultrason de 40kHZ et attends le réfléchissement du signal. Ensuite, en ayant en tête la vitesse du son, il effectue un rapide calcul pour déterminer la distance.



$$t = \frac{2d}{V}$$

- t : Temps entre émission et réception
- d : distance entre source et obstacle
- V : Vitesse de déplacement des ultrasons dans l'air

- **fils de connexion**

Fonctionnement : **Connecter les composants entre eux**

- **Un buzzer**

Fonctionnement

Ce composant électromagnétique ou piezzo électrique qui transforme l'énergie électrique en vibration, qui peut recevoir une tension continue

Montage :

➢ **Pour réaliser le montage, on peut connecter**

- **Pour LED :**

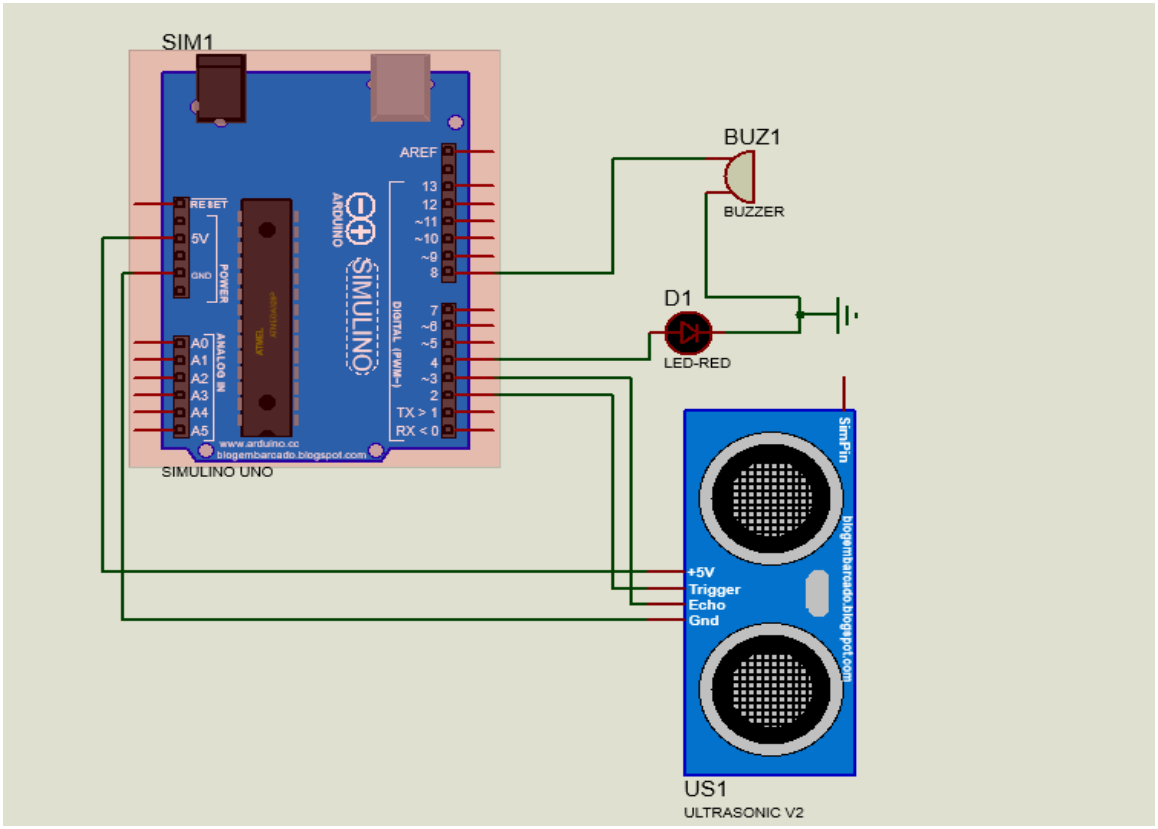
  - **La borne (+) de la LED rouge à la PIN 4 de l'Arduino**
  - **La borne (-) de la LED rouge à la PIN GND de l'Arduino**

- **Pour le capteur ultrason sonore:**

  - **la broche TRIG du capteur à ultrason à la PIN2 de l'Arduino**
  - **la broche Echo du capteur à ultrason à la PIN3 de l'Arduino.**
  - **la broche VCC du capteur à ultrason à la PIN +5V de l'Arduino**
  - **la broche GND du capteur à ultrason à la PIN GND de l'Arduino**

- **Pour buzzer :**

  - **Les bornes (+) de la buzzer à la PIN8 de l'Arduino**
  - **les bornes (-) de la buzzer à la PIN GND de l'Arduino**



Code : (avec commentaire)

test | pitches.h

```
//ultrason---
int pinTrig = 2;
int pinEcho = 3;
long temps;
float distance;
//------
//buzzeur---
#include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};
//------
const int led = 4;
// ============================
void setup() {  // put your setup code here, to run once:
    Serial.begin(9600);
   pinMode (led , OUTPUT);
  pinMode(pinTrig, OUTPUT);
  pinMode(pinEcho, INPUT);
  digitalWrite(pinTrig, LOW);
}
```

```
void loop() {// put your main code here, to run repeatedly:
//affichage du distance
  digitalWrite(pinTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTrig, LOW);
  temps = pulseIn(pinEcho, HIGH); //On lit la durée d'état haut sur la broche "Echo"
  if (temps > 25000) { //si la durée est supérieure à 25ms, l'onde est perdue
    Serial.println("Echec de la mesure");
  }
  else {
    temps = temps/2; // On divise cette durée par deux pour n'avoir qu'un trajet
    distance = (temps*340)/10000.0; //On calcul la distance avec la formule d=v*t
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
  }
  delay(2000);
//----buzzeur et LED
if ( distance < 3){
    // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {

    // to calculate the note duration, take one second divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);

    // to calculate the note duration, take one second divided by the note type.
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);

    // to distinguish the notes, set a minimum time between them.
    // the note's duration + 30% seems to work well:
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    // stop the tone playing:
    noTone(8);
  }
  digitalWrite (led , HIGH); delay(1000);
  }else{digitalWrite (led , LOW);delay(1000);}
}
```

Compilation terminée.

Le croquis utilise 5920 octets (18%) de l'espace de stockage de programmes. Le maximu