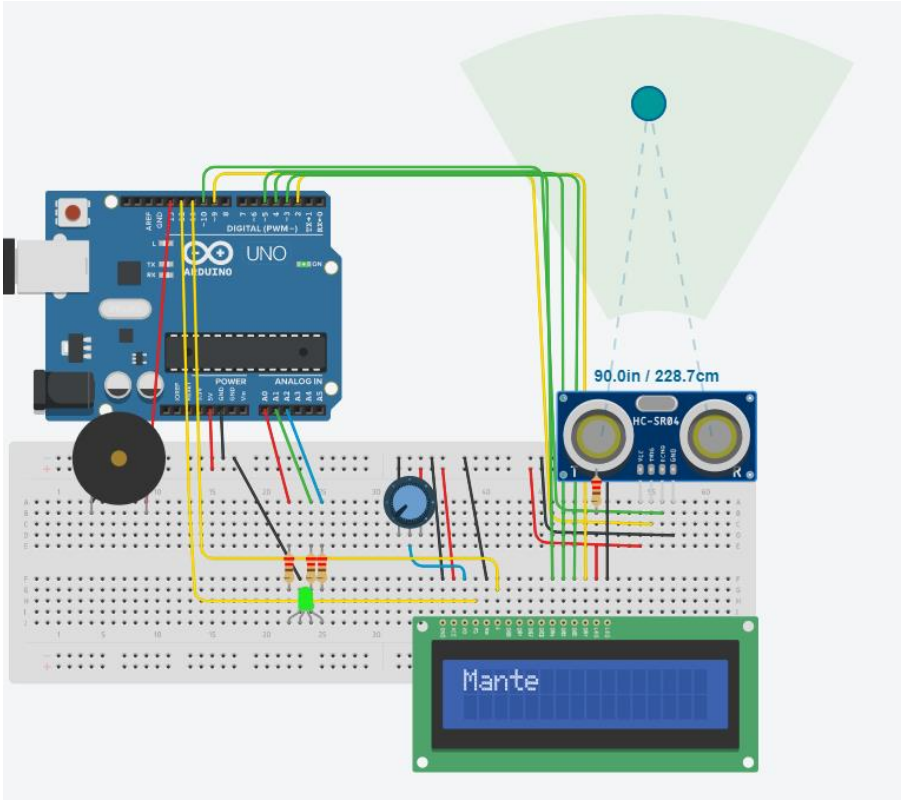
Universidad Nacional de Quilmes

Sensor de Distancia Social

Introducción a la Electrónica y Programación de Controladores con Arduino.



Alumno: Luis Mitre.

Profesor: José Luis Di Blase.

Objetivo:

El objetivo es crear un sensor de distancia social, impulsado por el contexto de pandemia donde se desarrolla este proyecto, con la placa Arduino que fuimos trabajando a lo largo del cuatrimestre, y algunos componentes más, en mi caso a través de Tinkercad emular el comportamiento del sensor que podría ser aplicado en un caso real para determinar la distancia entre 2 objetos o personas, y alertar sobre la proximidad de los mismos.

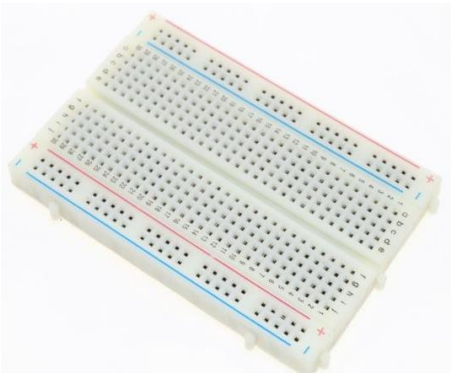
Descripción del Proyecto:

El sensor de Distancia Social mide a través del sensor la distancia que hay entre el, y otro objeto y/o persona, donde determina la distancia expresada en centímetros, y si la distancia es menor a los 100 cms el display imprime el mensaje “Aléjese, Por favor” junto con una luz led del color rojo, y un “pitido” o sirena a modo alerta. En el caso contrario, donde se respeta la distancia superior a los 100 cms, prende una luz verde, y el display imprime el mensaje “Mantenga Distancia” sin sonido alguno, ya que no es necesario emitir ningún tipo de alerta.

Componentes:

Placa Arduino x1

Breadboard x1

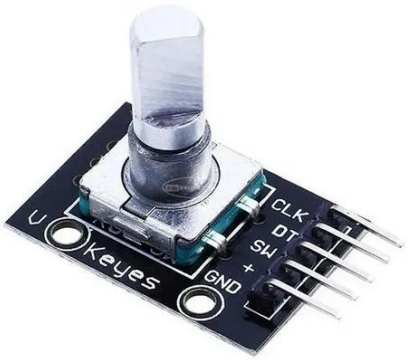


Sensor ultrasónico HC-SR04

Resistencia 220 ohms x4

LCD Display x1

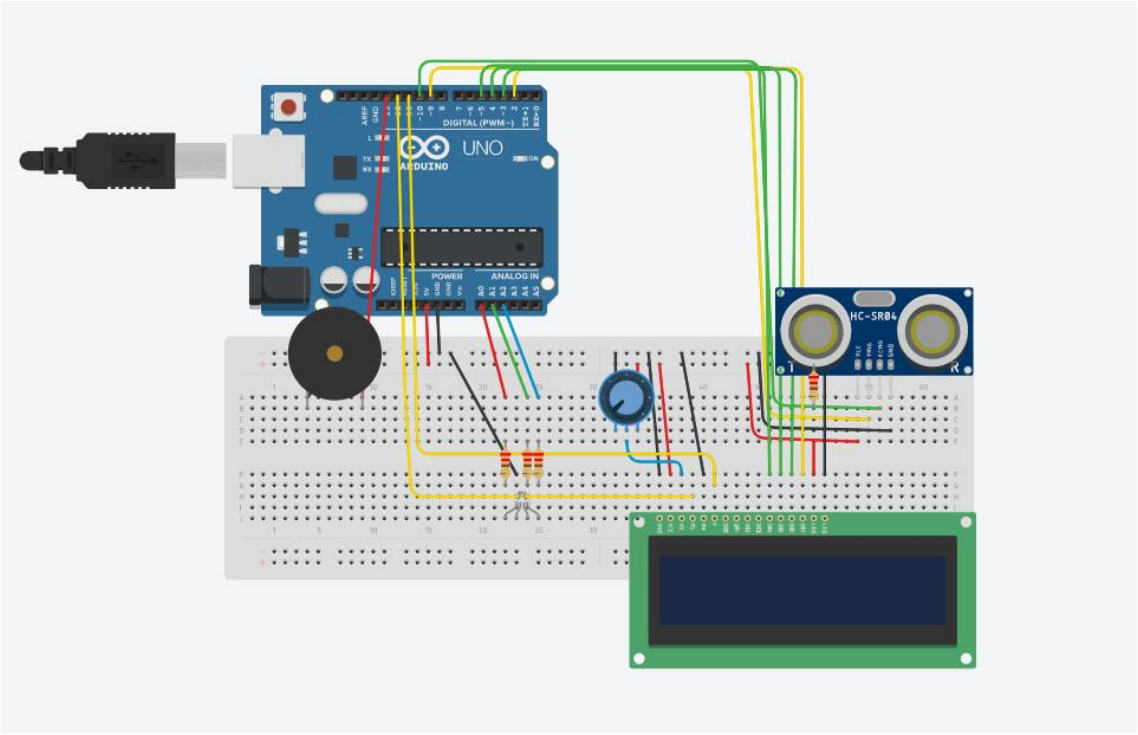
Potenciómetro x1



Buzzer x1

Muchos Cables.

Esquema:



Paso a Paso:

Inspirado en el proyecto: https://hackaday.com/2020/04/09/ultrasonic-sensor-helps-you-enforce-social-distancing/

Comencé con quizás lo más sencillo, pero lo que era lo más importante, ya que determinaba el comportamiento de todos los componentes, y por lo tanto el proyecto.

Primero monté el sensor HC-SR04 a mi Breadboard con la placa Arduino, si bien en el emulador Tinkercad la interacción nos permite ver la distancia en la cual ponemos el objetivo, en el código tuve que hacer la conversión para que exprese la distancia en centímetros, también agregué un serial para ir viendo la variación de la distancia.

Luego conecté la pantalla LCD para poder imprimir esa distancia que se iba calculando en el sensor, y acá es donde empecé a agregar más comportamiento al código, incorporé la lógica de si la distancia era inferior a los 100 centímetros, imprimía una alerta, y en el caso contrario imprimiría otro mensaje.

Seguí agregando una luz Led RGB con dos variaciones, prendería en color Rojo en el caso de la alerta de distancia, y en el caso contrario permanecería encendida de color verde, indicando que está todo bien. Finalizando agregué un buzzer o speaker, que emitiría un sonido o pitido en el caso de la alerta inferior a los 100 centímetros, no agregué ningún otro sonido en el otro caso, porque no tenía sentido.

Inconvenientes:

Quizás los inconvenientes que tuve fueron principalmente con el buzzer, ya que los ejemplos que encontraba, la mayoría utilizaba una librería “pitches.h” que yo en el emulador no podía importar, y aunque accedí al código no funcionaba como yo esperaba, y como de todas maneras necesitaba algo más básico ya que solo necesitaba un “pitido” para simular la alerta, encontré otro código que pude adaptar a mi necesidad.

Código Fuente:

```
//defines pins numbers
//include the library code:
#include <LiquidCrystal.h>

//initialize the library by associating any needed LCD interface pin
//with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
const int trigPin = 9;
const int echoPin = 10;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//defines variables
long duration;
int distance;
//led RGB
int red_light_pin= A0;
int green_light_pin = A1;
int blue_light_pin = A2;
// buzzer
int speakerPin = 13;
int length = 15; // el numero de las notas
char notes[] = "ccggaagffeeddc ";
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;

void setup() {
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
pinMode(trigPin, OUTPUT); //Sets the trigPin as an Output
pinMode(echoPin, INPUT); //Sets the echoPin as an Input
pinMode(red_light_pin, OUTPUT);
pinMode(green_light_pin, OUTPUT);
pinMode(blue_light_pin, OUTPUT);
pinMode(speakerPin, OUTPUT);
Serial.begin(9600); //Starts the serial communication
}
```

```
void loop() {
//Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
//Sets the trigPin on HIGH state for 10 micro seconds
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;
//Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);

  if(distance < int(100)){
    RGB_color(255, 0, 0); // Red
    lcd.print("Alejese, Por favor");
    playNote(notes[1], beats[1] * tempo);
    delay(500);
  }else{
    if(distance > int(100)){
      RGB_color(0, 0, 255); // Green
      lcd.print("Mantenga Distancia");
      delay(500);
    }
  }

  delay(500);
  lcd.clear();

}

void RGB_color(int red_light_value, int green_light_value, int blue_light_value){
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
  analogWrite(blue_light_pin, blue_light_value);
}
```

```

void playNote(char note, int duration) {
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
    // toca el tono correspondiente al nombre de la nota
  for (int i = 0; i < 8; i++) {
    if (names[i] == note) {
      playTone(tones[i], duration);
    }
  }
}

void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}
```

Links y Referencias:

Inspirado en : https://hackaday.com/2020/04/09/ultrasonic-sensor-helps-you-enforce-social-distancing/

Sensor HC-SR04: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

Lcd Display: https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay

RGB Led: https://create.arduino.cc/projecthub/muhammad-aqib/arduino-rgb-led-tutorial-fc003e

Código Buzzer: https://drive.google.com/file/d/0BwsJ8iCEj2GdbkdtSUpIUDhZTTg/view