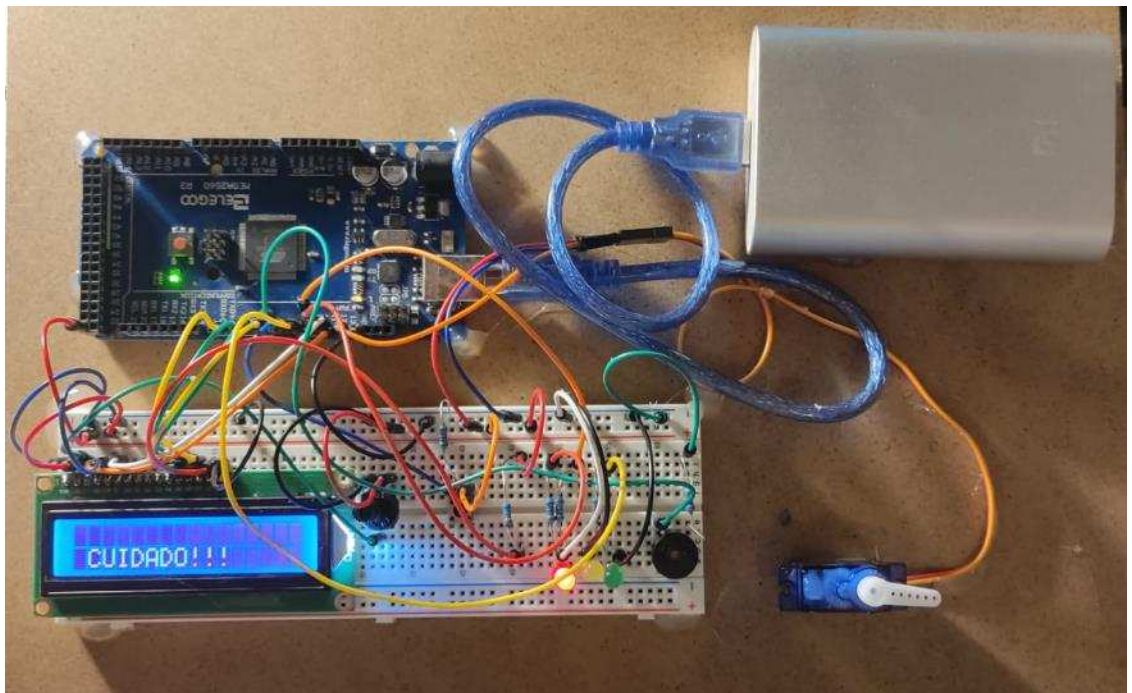


“Semáforo con barrera y pulsador”



Alumno: Pablo Conde Fernández

Centro Educativo: I.E.S Otero Pedrayo Ourense

1. INTRODUCCIÓN

¡Bienvenido a este proyecto de Arduino! En este proyecto utilizaremos una placa Arduino Mega, un display LCD de 16x2, un buzzer pasivo, un servomotor SG90 y un botón para crear un sistema interactivo. El objetivo del proyecto es demostrar cómo se pueden utilizar diferentes componentes y sensores en conjunto para crear un sistema automatizado.

El display LCD se utilizará para mostrar información y mensajes, mientras que el buzzer pasivo emitirá sonidos para indicar eventos importantes. El servomotor SG90 se usará para controlar un movimiento mecánico de barrera y el botón se utilizará para activar la petición del peatón.

A medida que avancemos en el proyecto, aprenderemos cómo programar el Arduino para controlar cada componente y cómo hacer que trabajen juntos para crear un sistema funcional y útil. ¡Empecemos!

2. MATERIAL Y MÉTODOS

Para crear un proyecto de semáforo con Arduino, puedes seguir estos pasos y utilizar los siguientes materiales y métodos:

Materiales:

- 1 placa de Arduino Mega
- 1 Protoboard
- 1 LED rojo
- 1 LED amarillo
- 1 LED verde
- Display LCD 16x2 LCD1602
- Buzzer pasivo KY-006
- Servomotor SG90
- 3 resistencias de 220 ohmios
- 2 resistencias de 200 ohmios
- 1 resistencias de 10KΩ
- cables para protoboard
- 1 potenciómetro
- 1 cable USB para conectar Arduino a la computadora
- 1 fuente de alimentación

Métodos:

- Conecta el LED rojo al pin 10 de la placa Arduino, el LED amarillo al pin 9 y el LED verde al pin 8.

- Conecta las resistencias de 220 ohmios a cada LED, conectando el otro extremo de la resistencia al polo negativo de la fuente de alimentación.
- Conecta el polo negativo de la fuente de alimentación al pin GND de Arduino y el polo positivo al pin de 5V.
- Conecta el cable USB a la placa de Arduino y al puerto USB de la computadora.
- Abre el software de Arduino IDE en la computadora y crea un sketch:

```

// C++ code
//
#include <Servo.h>

#include <LiquidCrystal.h>
int rs = 12;
int m = 11;
int d4 = 5;
int d5 = 4;
int d6 = 3;
int d7 = 2;

LiquidCrystal lcd(rs, m, d4, d5, d6, d7);

const int ledRojo = 10;
const int ledAmarillo = 9;
const int ledVerde = 8;

const int Pulsador = 7;

Servo servo_13;

//Darth Vader - Marcha Imperial

const int c = 261;
const int d = 294;
const int e = 329;
const int f = 349;

const int g = 391;
const int aS = 415;
const int a = 440;
const int aS2 = 465;
const int b = 485;
const int cH = 523;
const int cSH = 554;
const int d4 = 587;
const int dSH = 622;
const int eH = 659;
const int fH = 696;
const int fSH = 740;
const int gH = 784;
const int aSH = 830;
const int aH = 880;

const int buzzerPin = 6;
const int ledPin1 = 9;
const int ledPin2 = 1;

int counter = 0;

void setup()
{
  lcd.begin(16,2);
  pinMode(ledRojo, OUTPUT);
  pinMode(ledAmarillo, OUTPUT);
  pinMode(ledVerde, OUTPUT);
  pinMode(Pulsador, INPUT);
  servo_13.attach(13, 500, 2500);
  pinMode(buzzerPin, OUTPUT);
}

void loop()
{
  if(digitalRead(Pulsador) == HIGH)
  {
    digitalWrite(ledRojo, LOW);
    for (int i=0; i<5; i++)
    {
      lcd.setCursor(1,2); // coordenadas de inicio de escritura
      lcd.print("¡ATENCIÓN!!");
      digitalWrite(ledAmarillo, HIGH);
      delay(250);
      digitalWrite(ledAmarillo, LOW);
      delay(250);
    }
  }
}

servo_13.write(90);
digitalWrite(ledVerde, HIGH);
lcd.setCursor(0,0) // coordenadas de inicio de escritura
lcd.print("¡PULSE PULSADOR.");
lcd.setCursor(1,2) // coordenadas de inicio de escritura
lcd.print("¡¡JAJAJ!!");

{
  firstSection();
  secondSection();
}

beep(f, 250);
beep(aS, 500);
beep(f, 500);
beep(a, 125);
beep(cH, 500);
beep(a, 375);
beep(aH, 125);
beep(cH, 650);
delay(500);

secondSection();

beep(f, 250);
beep(aS, 500);
beep(f, 375);
beep(aH, 125);
beep(a, 500);
beep(f, 375);
beep(aH, 125);
beep(a, 650);
delay(100);
}
delay(1000);
lcd.clear();

servo_13.write(0);
digitalWrite(ledRojo, HIGH);
digitalWrite(ledAmarillo, LOW);
digitalWrite(ledVerde, LOW);
delay(1000);

}

}

{
  digitalWrite(ledRojo, HIGH);
  servo_13.write(0);
  lcd.setCursor(1,2) // coordenadas de inicio de escritura
  lcd.print("¡¡CUIDADO!!!");
}

void beep(int note, int duration)
{
  digitalWrite(buzzerPin, LOW);
  digitalWrite(buzzerPin, HIGH);
  delay(duration);
  digitalWrite(buzzerPin, LOW);
}

for(counter % 2 == 0)
{
  digitalWrite(ledPin1, HIGH);
  delay(duration);
  digitalWrite(ledPin1, LOW);
  digitalWrite(ledPin2, HIGH);
  delay(duration);
  digitalWrite(ledPin2, LOW);
}

noTone(buzzerPin);
delay(50);
counter++;

}

void firstSection()
{
  beep(a, 500);
  beep(a, 500);
  beep(f, 350);
  beep(cH, 175);
  beep(f, 125);
  beep(f, 125);
  beep(f, 250);
  delay(325);
}

void secondSection()
{
  beep(aS, 250);
  beep(aS, 500);
  beep(dH, 325);
  beep(cSH, 175);
  beep(cH, 125);
  beep(b, 125);
  beep(cH, 250);
  delay(250);
}

```

Imagen 1: código de funcionamiento

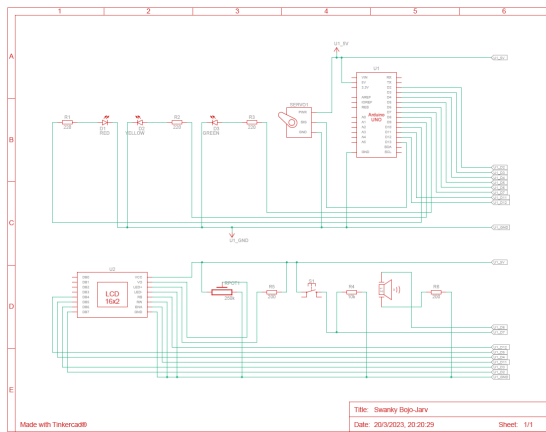


Imagen 2: esquema de conexiones

3. RESULTADOS

El semáforo de tráfico construido con Arduino y LED funcionó correctamente. El LED rojo se encendía, seguido del LED amarillo y el LED verde.

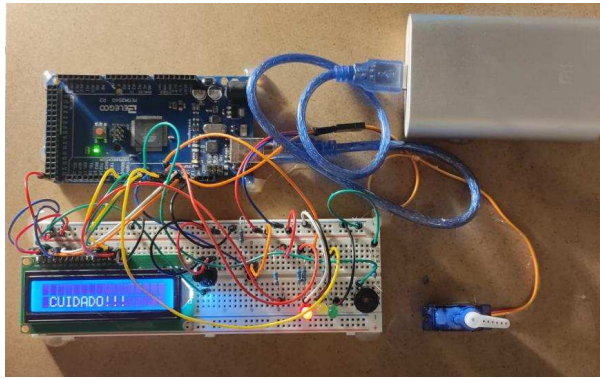


Imagen 3: montaje

Pasos de actuación:

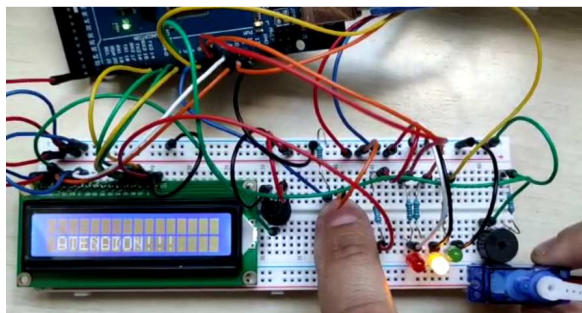


Imagen 4: paso 1

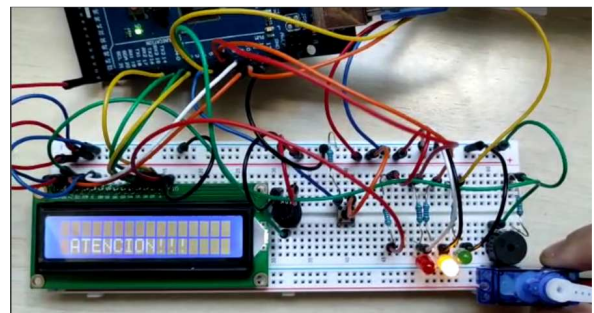


Imagen 5: paso 2

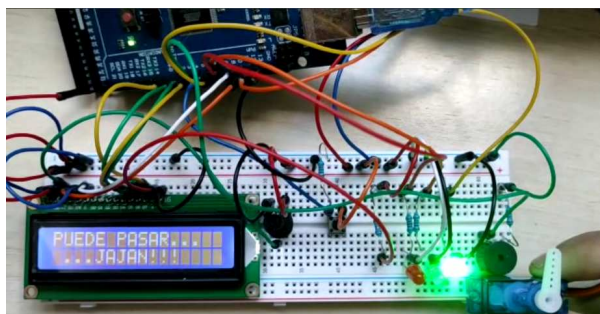


Imagen 6: paso 3

4. CONCLUSIONES

El proyecto demostró que es posible construir un semáforo de tráfico funcional con Arduino y LED. Además, se aprendió cómo programar los pines de salida de Arduino y cómo conectar componentes electrónicos básicos. Este proyecto puede ser utilizado como base para proyectos más avanzados con Arduino y LED.

5. BIBLIOGRAFIA

- Programa TinkerCad para el desarrollo previo del proyecto
- “Melodía de Star Wars con Buzzer y Arduino” de Rubén Gonzalez