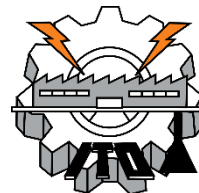




TECNOLÓGICO
NACIONAL DE MÉXICO



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA

MATERIA:

TÓPICOS AVANZADOS DE PROGRAMACIÓN

CLAVE DE MATERIA:

SCD1027

CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

“PROYECTO FINAL”.

PRESENTAN:

LÓPEZ ROJAS JOSE LUIS
VÁZQUEZ MEDINA ELEONAI
JOSE SEBASTIAN JAFET

NOMBRE DEL CATEDRÁTICO:

AQUINO PEREZ ANTONIO DE JESÚS

GRUPO: 4SB

HORARIO: 17:00-18:00

OAXACA DE JUÁREZ, OAXACA A 10 DE DICIEMBRE DEL 2024



Avenida Ing. Víctor Bravo Ahuja No. 125 Esquina Calzada Tecnológico, C.P. 68030



Contenido

INTRODUCCIÓN	3
COMPONENTES ELECTRÓNICOS:	4
MATERIALES:.....	4
PROCESO.....	5
FUNCIONAMIENTO.	7
CÓDIGO.....	8
CONCLUSIÓN	11
BIBLIOGRAFÍA.....	12

INTRODUCCIÓN

En este proyecto, nos propusimos desarrollar una maqueta funcional que simula un sistema de semáforos con detección de infracciones mediante componentes electrónicos controlados por una placa de Arduino UNO. El objetivo principal fue crear un modelo interactivo que combine programación y electrónica para demostrar cómo se puede aplicar la tecnología en escenarios cotidianos como el control de tráfico vehicular. Con un enfoque práctico, utilizamos materiales accesibles y técnicas sencillas, haciendo que este proyecto sea no solo educativo, sino también una experiencia divertida y significativa.

COMPONENTES ELECTRÓNICOS:

- 3 leds rojo.
- 3 leds verde.
- 3 leds amarillo.
- Protoboard.
- Arduino UNO.
- Sensor PIR.
- Jumpers (hembra-macho, hembra-hembra, macho-macho).
- 3 resistencias de 220Ω .

MATERIALES:

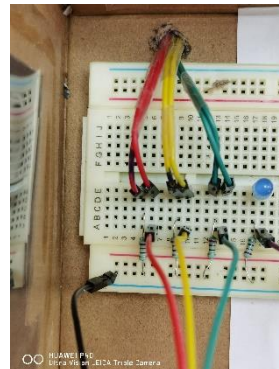
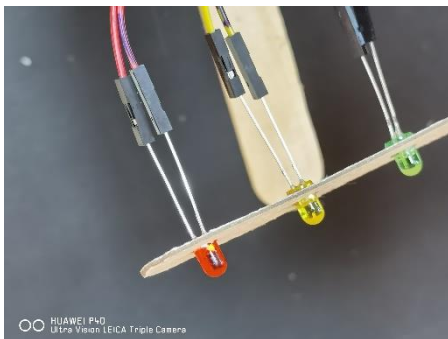
- Cartulina negra.
- $\frac{1}{2}$ de Papel cascarón.
- Silicón líquido.
- Lengüetero.
- Tijeras.
- Hojas blancas.
- Pintura acrílica roja, gris, amarilla y blanca.
- Regla.
- Carro a control remoto.
- Cautín.
- Lápiz.
- Caja de cartón pequeña.

PROCESO.

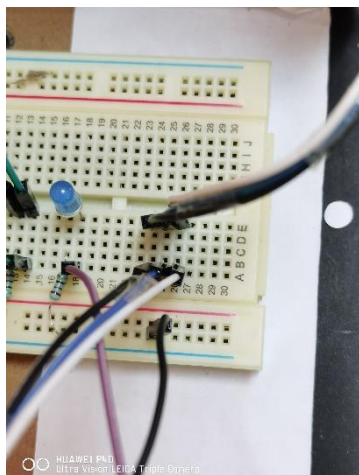
Para empezar a hacer la maqueta, pegamos la cartulina negra sobre el papel cascarrón y cortamos las partes sobrantes; esto para simular el asfalto de las carreteras. Trazamos líneas para simular los cruces peatonales y las líneas de carril; los cruces peatonales se colorean con pintura amarilla y las líneas de carril con pintura blanca. Forramos con hoja blanca la cajita de nuestro carro a control remoto para posteriormente pintarla con los colores gris, amarillo y rojo, simulando una tienda.

Para el circuito eléctrico, se utilizaron los 3 leds de colores rojo, amarillo y verde para simular el semáforo, están conectadas a las salidas 13, 12 y 11 de la placa del Arduino UNO por medio de la protoboard.

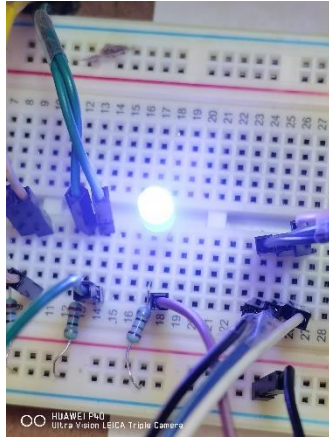
A cada led se le conecta una resistencia de 220Ω en la salida a GND.



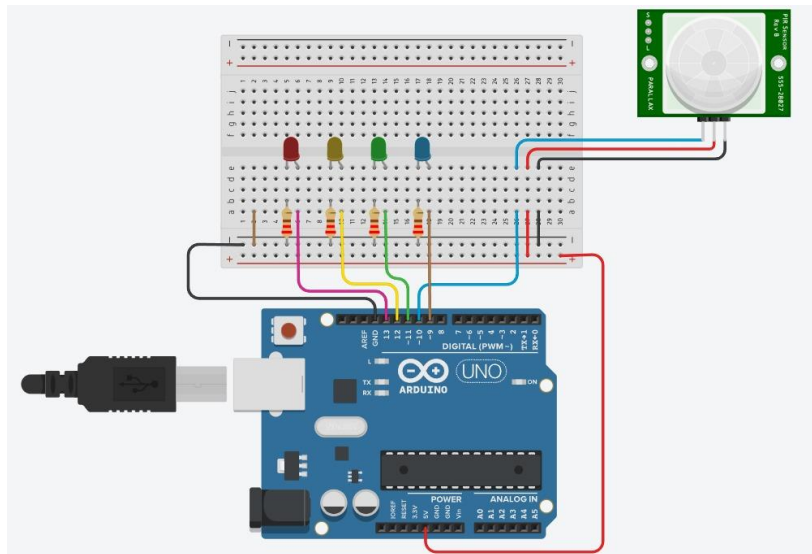
El sensor PIR está conectado a la salida 10 de la placa del Arduino UNO por medio de la protoboard y tendrá como función detectar el movimiento de los vehículos cuando la luz del semáforo indique rojo.



Se conecta un led de advertencia de color azul en la salida 9 de la placa del Arduino UNO, que se enciende cuando el sensor PIR detecta movimiento.



Finalmente se codifica la placa del Arduino UNO para controlar todos los componentes anteriores.

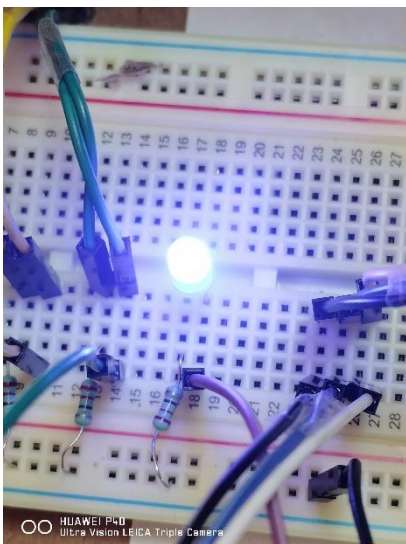


FUNCIONAMIENTO.

El semáforo funcionará como uno habitual, ejecutando un ciclo en el código del Arduino en la secuencia tradicional, el cual estará 10 segundos en color rojo, 9 segundos en color verde, 4 segundos en color verde parpadeando, 2 segundos en color amarillo y finalmente encender el rojo para iniciar el ciclo de nuevo.



El sensor PIR tendrá la función de detectar el cambio de radiación infrarroja que producen los objetos al pasar cuando el led rojo del semáforo este encendido. Al detectar un movimiento, este lanzará una alerta encendiendo un led azul, indicando que se ha cometido una infracción.



CÓDIGO.

// Pines para los LEDs

int LED_ROJO = 13;

int LED_AMARILLO = 12;

int LED_VERDE = 11;

int LED_IR = 9;

// Pin para el sensor IR

const int SENSOR_IR = 10;

void setup() {

 // Configurar pines como salida

 pinMode(LED_ROJO, OUTPUT);

 pinMode(LED_AMARILLO, OUTPUT);

 pinMode(LED_VERDE, OUTPUT);

 pinMode(LED_IR, OUTPUT);

 // Configurar pin del sensor IR como entrada

 pinMode(SENSOR_IR, INPUT);

 // Iniciar comunicación serie

 Serial.begin(9600);

}

void loop() {

 // Secuencia del semáforo

 digitalWrite(LED_ROJO, HIGH); // Enciende el LED rojo

 unsigned long inicio = millis();


```

while (millis() - inicio < 10000) {
    verificarVehiculo();      // Medir continuamente
    delay(200);               // Verificación cada 200 ms
}

digitalWrite(LED_ROJO, LOW); // Apaga el LED rojo

digitalWrite(LED_VERDE, HIGH); // Enciende el LED verde
delay(9000);                  // Espera 5 segundos
for(int i = 0; i <= 3; i = i + 1){
    digitalWrite(LED_VERDE, LOW);
    delay(400);
    digitalWrite(LED_VERDE, HIGH);
    delay(400);
}
digitalWrite(LED_VERDE, LOW); // Apaga el LED verde

digitalWrite(LED_AMARILLO, HIGH); // Enciende el LED amarillo
delay(2000);                  // Espera 2 segundos
digitalWrite(LED_AMARILLO, LOW); // Apaga el LED amarillo

}

// Función para verificar la presencia de un vehículo
void verificarVehiculo() {
    // Leer el estado del sensor IR
    int estadoSensor = digitalRead(SENSOR_IR);
    // Si el sensor detecta un objeto (auto)
    if (estadoSensor == HIGH) {

```

```
Serial.println("Infraccion detectada");  
// Aquí podrías agregar una alarma o encender un buzzer  
advertencia();  
} else {  
    Serial.println("No se detecta vehículo.");  
}  
}
```

```
void advertencia(){  
    for(int i = 0; i <= 5; i = i + 1){  
        digitalWrite(LED_IR, HIGH);  
        delay(150);  
        digitalWrite(LED_IR, LOW);  
        delay(150);  
    }  
}
```

CONCLUSIÓN

Este proyecto no solo nos permitió poner en práctica los conocimientos adquiridos en programación y electrónica, sino que también nos retó a ser creativos y a trabajar en equipo para superar los desafíos. El resultado final, una maqueta funcional que simula un semáforo con detección de infracciones, refleja nuestro esfuerzo y compromiso. Mas allá de cumplir con los objetivos académicos, este proyecto nos dejó valiosas lecciones sobre la importancia de la tecnología en la vida diaria y como esta puede ser utilizada para mejorar procesos y crear soluciones innovadoras.

BIBLIOGRAFÍA

1. Banzi, M., & Shiloh, M. (2020). Getting Started with Arduino: The Open Source Electronics Prototyping Platform (4th ed.). Maker Media.
2. Monk, S. (2021). Programming Arduino: Getting Started with Sketches (3rd ed.). McGraw-Hill Education.
3. Massimo, B. (2011). Arduino: A Quick-Start Guide (2nd ed.). Pragmatic Bookshelf.