

# **UNIVERSIDAD PONTIFICIA BOLIVARIANA**

## FACULTAD DE INGENIERÍA ELECTRÓNICA

Seccional de Montería

## Examen numero 1

“semáforo con Arduino 1 tinkercad*”*

##### *Joseph Rodríguez Cardales*

000549061

## PROYECTO INTEGRADOR

##### DOCENTE DEL PROGRAMA

Francisco Ricardo Barreiro pinto

05 / sep / 2024

REFORZADO PARA MEJORAR SU USO EN VIVIENDAS DE INTERÉS SOCIAL EN TUXTLA GUTIÉRREZ, CHIAPAS”

### T E S I S

## QUE PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERIA

(CONSTRUCCIÓN)

PRESENTA

Arq. Beatriz Adriana Vázquez Cruz

##### DIRECTOR DE TESIS

*DR. FRANCISCO ALBERTO ALONSO FARRERA*

TUXTLA GUTIÉRREZ, CHIAPAS ABRIL DE 2010.

TUXTLA GUTIÉRREZ, CHIAPAS ABRIL DE 2010.

Tabla de Contenido

1. Introducción

- Planteamiento de la Problemática

- Objetivos del Proyecto

2. Descripción del Proyecto

- Herramientas y Componentes

- Lógica de Funcionamiento

3. Desarrollo del Proyecto

- Diseño del Circuito en Tinkercad

- Programación en Arduino

4. Simulación y Verificación

- Ejecución de la Simulación

- Ajustes y Verificación

5. Documentación

- Explicación de la Lógica de Programación

- Capturas de Pantalla y Enlace de Tinkercad

- Integrantes del Grupo(no aplica)

6. Conclusiones

Introducción:

las intersecciones con múltiples vías la coordinación del tráfico es esencial para prevenir accidente y mejorar la eficiencia del flujo vehicular. Un sistema de semáforo bien diseñado es clave para asegurar que los vehículos se detengas y avancen de manera ordena y segura. En este proyecto utilizaremos la plataforma de desarrollo Arduino 1 y la herramienta de simulación tinkercad para diseñar y programar un sistema de semáforo que controle una intersección de tres vías.

planteamiento de la problemática:

las intersecciones con tres vías presentan un desafío significativo en cuanto a la gestión del trafico ya que es necesario coordinar las señales de trafico de manera de que solo una vía este en movimiento mientras la otras dos se encuentran detenidas esto es crucial para evitar colisiones y garantizar un flujo de trafico seguro. La ´problemática principal radica en crear un sistema automatizado que controle las señales de trafico de manera eficiente y sin errores, utilizando Arduino 1 como base en el sistema.

objetivos del proyecto

el objetivo principal de este proyecto es desarrollar un sistema de semáforo controlado por Arduino para una intersección de tres vías.

objetivos específicos

1.utilizar la herramienta de tinkercad para simular un circuito electrónico necesario para el semáforo.

2.programar la tarjeta de desarrollo para controlar las luces de trafico de manera adecuada y que permita un cambio de luces ordenado y seguro para las tres vías.

3.verificar el funcionamiento del sistema y ajustar cosas que sean necesarias.

desarrollo del proyecto

diseño del circuito em timkercad

1. espacio adecuado con todos los componentes los cuales pueden entrar

# 9 leds: tres para cada vía (rojo, naranja, verde)

# resistencias: resistencias adecuadas para cada led en este caso utilizamos unos 220 ohmios

# cables conexiones y la fuente que en este caso es el Arduino

conexiones del circuito

1. los pines digitales utilizados fueron como son tres semáforos los que se utilizan para las vías pues los pines para el primer semáforo son (2,3,4) segundo semáforo (5,6,7) y tercer semáforo (8,9,10)
2. en el siguiente paso conecte las resistencias en serie con los leds para limitar la corriente
3. cables tenemos que tierra es (GND) a las patas negativas de los leds
4. para el 4 paso me asegure de tener todas las conexiones buenas y que todo este en su lugar

programación de Arduino

los pasos son:

1. definir los pines los cuales ya definimos.
2. configurar el modo: de los pines como salida del código.
3. programa la secuencia de luces: cada vía debe tener un siclo de verde, naranja y rojo debemos de asegurarnos de que los ciclos de las otras vías este en rojo cuando una vía este en verde.
4. ajustar los tiempos: para cada estado de luz para que el semáforo funciones de manera realista.
5. subir el código: después subes el código y lo pruebas a ver si todo esta en correcto estado y ya por último tenemos que iniciar la simulación y ver que corregir o si ya está perfecto todo.

documentación

Entregar un documento que incluya:

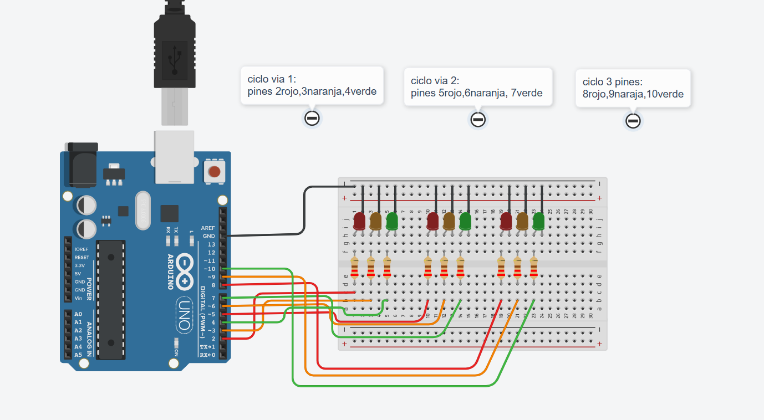
- El enlace al circuito en Tinkercad para que el instructor pueda realizar la simulación:

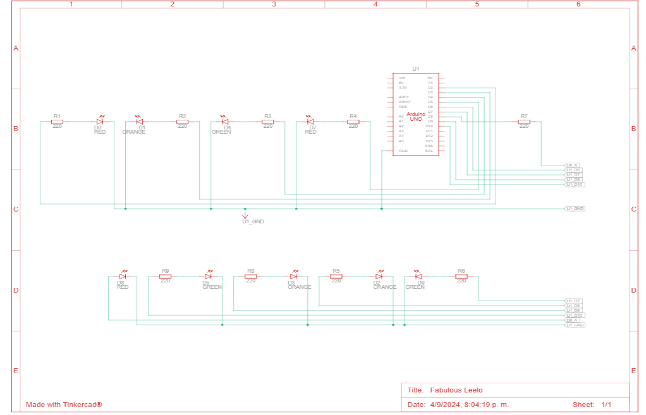
https://www.tinkercad.com/things/00u1QDSwtc6-semaforo-3-vias-arduino-1-

Fotos:

ESQUEMATICO

CIRCUITO





COMPONENTES



- Una explicación detallada de la programación

En la función setup:

1. Se configuran todos los pines definidos como salidas (OUTPUT), ya que se usarán para encender y apagar LEDs.
2. Se llama a la función apagarLeds para asegurar que todos los LEDs estén apagados cuando el programa comienza.

void setup() {

// Configurar pines como salida

pinMode(red1, OUTPUT);

pinMode(orange1, OUTPUT);

pinMode(green1, OUTPUT);

pinMode(red2, OUTPUT);

pinMode(orange2, OUTPUT);

pinMode(green2, OUTPUT);

pinMode(red3, OUTPUT);

pinMode(orange3, OUTPUT);

pinMode(green3, OUTPUT);

// Apagar todos los LEDs al inicio

apagarLeds();

}

2. void apagarLeds() { // Apagar todos los LEDs digitalWrite(red1, LOW); digitalWrite(orange1, LOW); digitalWrite(green1, LOW); digitalWrite(red2, LOW); digitalWrite(orange2, LOW); digitalWrite(green2, LOW); digitalWrite(red3, LOW); digitalWrite(orange3, LOW); digitalWrite(green3, LOW);

1. void apagarLeds() { // Apagar todos los LEDs digitalWrite(red1, LOW); digitalWrite(orange1, LOW); digitalWrite(green1, LOW); digitalWrite(red2, LOW); digitalWrite(orange2, LOW); digitalWrite(green2, LOW); digitalWrite(red3, LOW); digitalWrite(orange3, LOW); digitalWrite(green3, LOW);

Aquí se definen los pines digitales del Arduino que estarán conectados a los Leds de los semáforos. Cada pin controla un color específico de un semáforo:

const int red1 = 2;

const int orange1 = 3;

const int green1 = 4;

const int red2 = 5;

const int orange2 = 6;

const int green2 = 7;

const int red3 = 8;

const int orange3 = 9;

const int green3 = 10;

3. void apagarLeds() { // Apagar todos los LEDs digitalWrite(red1, LOW); digitalWrite(orange1, LOW); digitalWrite(green1, LOW); digitalWrite(red2, LOW); digitalWrite(orange2, LOW); digitalWrite(green2, LOW); digitalWrite(red3, LOW); digitalWrite(orange3, LOW); digitalWrite(green3, LOW);

En la función loop, el código ejecuta un ciclo para cada vía del semáforo:

1. Vía 1: Llama a semaforoVia con el semáforo verde de la primera vía, los semáforos rojos de las vías 2 y 3, y el semáforo naranja de la vía 1.
2. Vía 2: Llama a semaforoVia con el semáforo verde de la segunda vía, los semáforos rojos de las vías 1 y 3, y el semáforo naranja de la vía 2.
3. Vía 3: Llama a semaforoVia con el semáforo verde de la tercera vía, los semáforos rojos de las vías 1 y 2, y el semáforo naranja de la vía 3.

void loop() { // Ciclo para vía 1 semaforoVia(green1, red1, red2, red3, orange1, 15000, 1000);

// Ciclo para vía 2 semaforoVia(green2, red2, red1, red3, orange2, 15000, 1000);

// Ciclo para vía 3 semaforoVia(green3, red3, red1, red2, orange3, 15000, 1000); }

5. void apagarLeds() { // Apagar todos los LEDs digitalWrite(red1, LOW); digitalWrite(orange1, LOW); digitalWrite(green1, LOW); digitalWrite(red2, LOW); digitalWrite(orange2, LOW); digitalWrite(green2, LOW); digitalWrite(red3, LOW); digitalWrite(orange3, LOW); digitalWrite(green3, LOW);

4. void apagarLeds() { // Apagar todos los LEDs digitalWrite(red1, LOW); digitalWrite(orange1, LOW); digitalWrite(green1, LOW); digitalWrite(red2, LOW); digitalWrite(orange2, LOW); digitalWrite(green2, LOW); digitalWrite(red3, LOW); digitalWrite(orange3, LOW); digitalWrite(green3, LOW);

La función apagarLeds se encarga de apagar todos los LEDs, asegurando que no queden encendidos cuando se inicie un nuevo ciclo de semáforo.

void apagarLeds() {

// Apagar todos los LEDs

digitalWrite(red1, LOW);

digitalWrite(orange1, LOW);

digitalWrite(green1, LOW);

digitalWrite(red2, LOW);

digitalWrite(orange2, LOW);

digitalWrite(green2, LOW);

digitalWrite(red3, LOW);

digitalWrite(orange3, LOW);

digitalWrite(green3, LOW);

La función semaforoVia controla el comportamiento de un semáforo específico:

1. Apagar todos los LEDs: Llama a apagarLeds para asegurarse de que todos los LEDs estén apagados antes de comenzar el ciclo.
2. Encender rojos en los otros semáforos: Enciende los LEDs rojos en los semáforos de las otras dos vías.
3. Encender verde en la vía actual: Enciende el LED verde en la vía actual y espera el tiempo especificado (15 segundos en este caso).
4. Cambiar a naranja: Apaga el LED verde y enciende el LED naranja en la misma vía. Espera 1 segundo.
5. Cambiar a rojo: Apaga el LED naranja y enciende el LED rojo en la misma vía.

void semaforoVia(int verde, int rojoActual, int rojoOtro1, int rojoOtro2, int naranja, int tiempoVerde, int tiempoNaranja) {

// Apagar todos los LEDs al inicio del ciclo apagarLeds();

// Encender rojo para los otros dos semáforos digitalWrite(rojoOtro1, HIGH); digitalWrite(rojoOtro2, HIGH);

// Encender verde para la vía actual digitalWrite(verde, HIGH); delay(tiempoVerde);

// Mantener el verde por 15 segundos

// Apagar verde y encender naranja para la vía actual digitalWrite(verde, LOW); digitalWrite(naranja, HIGH); delay(tiempoNaranja);

// Mantener el naranja por 1 segundo

// Apagar naranja y encender rojo para la vía actual digitalWrite(naranja, LOW); digitalWrite(rojoActual, HIGH); }

RESUMEN

Para concluir pues este fue todo mi proceso para llevar a cabo este proyecto con Arduino q y con el simulador de tinkercad vemos que todo funciona a la perfección y pues con esta implementación podemos ver que no habrá accidentes todo será seguro y las escalas que tiene son correctas con los ciclos también muchas gracias por este proceso de la fabricación del semáforo para tres vías.

Link de repuesto: https://www.tinkercad.com/things/00u1QDSwtc6-semaforo-3-vias-arduino-1-