

NOMBRE DE LA PRÁCTICA	Circuito de Semáforos			No.	UNIDAD 4
ASIGNATURA:	LENGUAJE INTERFAZ	CARRERA:	ISIC	PLAN:	ISIC-2010-204

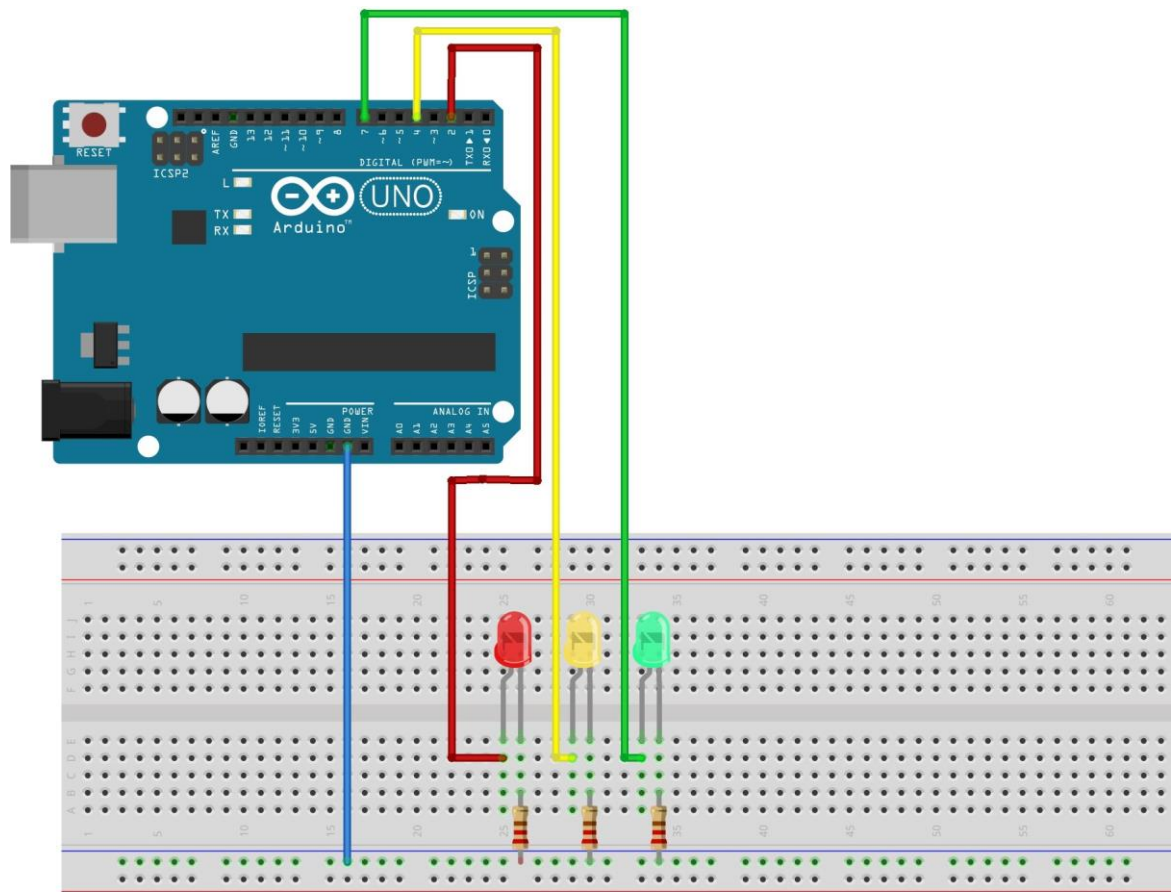
Nombre: Jesús Navarrete Martínez

Grupo: 3501

Objetivo: Realizar los circuitos y el programa para controlar un semáforo inicialmente y luego dos en forma sincronizada.

1. Realiza los circuitos y el programa para controlar un semáforo.

Circuito a realizar:



Código desarrollado:

```
sketch_mar05a Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

sketch_mar05a $

#define pinLED1 8 // Definimos pin 8
#define pinLED2 12 // Definimos pin 12
#define pinLED3 13 // Definimos pin 13

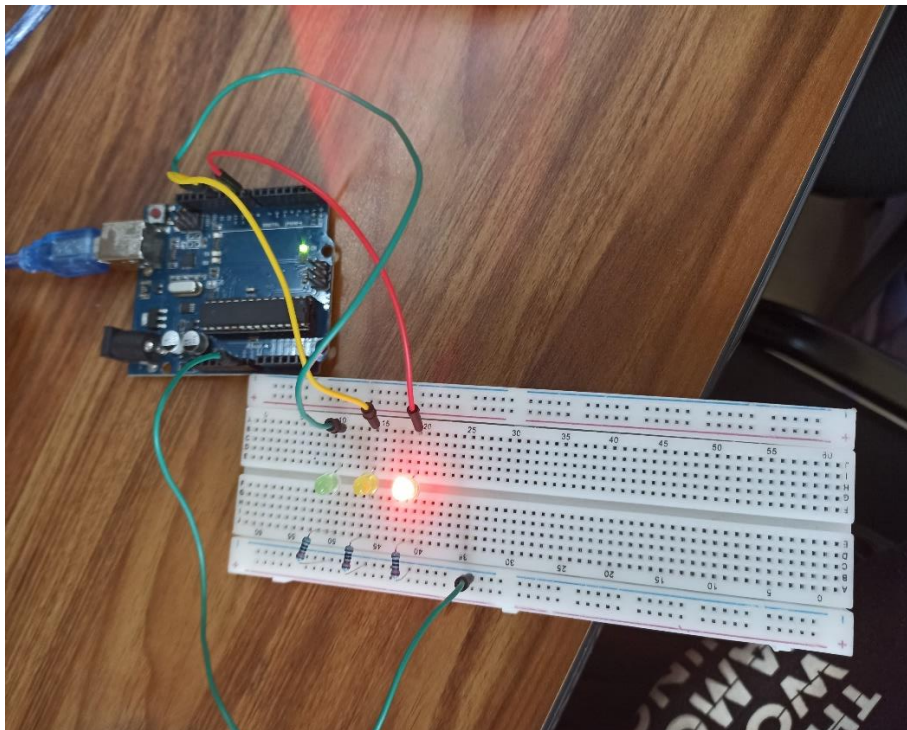
void setup() {
  pinMode(pinLED1, OUTPUT); // Declaramos LED1 como salida
  pinMode(pinLED2, OUTPUT); // Declaramos LED2 como salida
  pinMode(pinLED3, OUTPUT); // Declaramos LED3 como salida
}

void loop() {
  digitalWrite(pinLED1, HIGH); // Enciende LED1
  delay(500); // Espera 500ms
  digitalWrite(pinLED1, LOW); // Apaga LED1

  digitalWrite(pinLED2, HIGH); // Enciende LED2
  delay(500); // Espera 500ms
  digitalWrite(pinLED2, LOW); // Apaga LED2

  digitalWrite(pinLED3, HIGH); // Enciende LED3
  delay(500); // Espera 500ms
  digitalWrite(pinLED3, LOW); // Apaga LED3
}
```

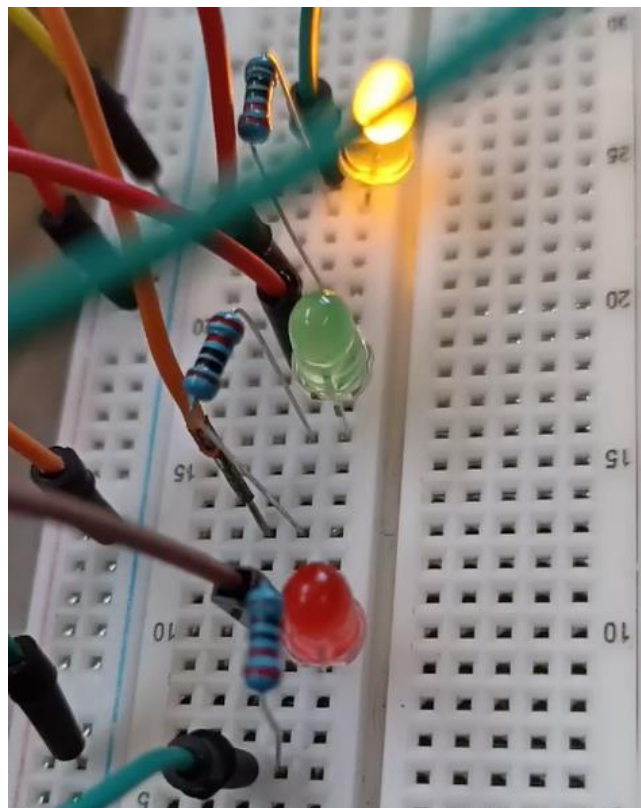
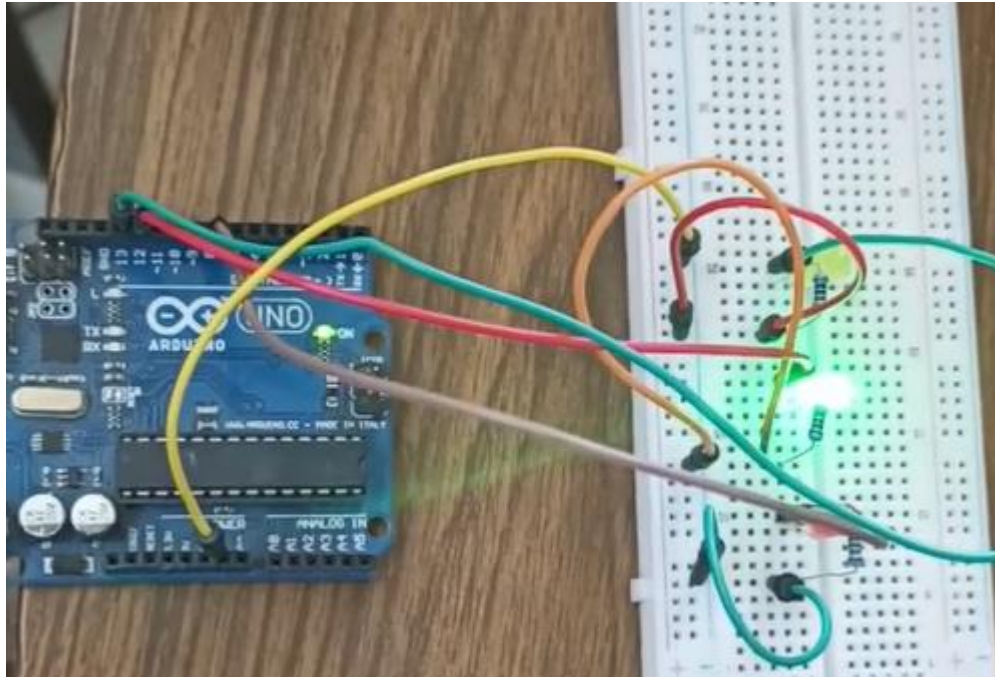
Demostración de su funcionamiento:



PRÁCTICA 11

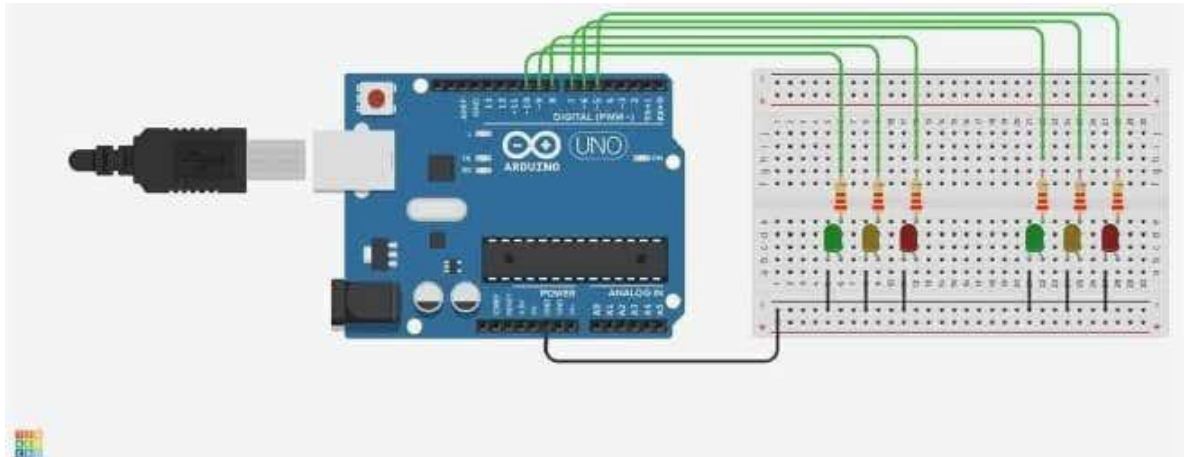
Ing. y Esp. Rodolfo Guadalupe Alcántara
Rosales

5



- Realiza los circuitos y el programa para controlar dos semáforos en forma sincronizada.

Circuito a realizar:



Código desarrollado:

```

sketch_mar05a Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

sketch_mar05a §

#define pinLED1 8
#define pinLED2 12
#define pinLED3 13
#define pinLED4 2
#define pinLED5 4
#define pinLED6 7

void setup() {
  pinMode(pinLED1, OUTPUT);
  pinMode(pinLED2, OUTPUT);
  pinMode(pinLED3, OUTPUT);
  pinMode(pinLED4, OUTPUT);
  pinMode(pinLED5, OUTPUT);
  pinMode(pinLED6, OUTPUT);
}

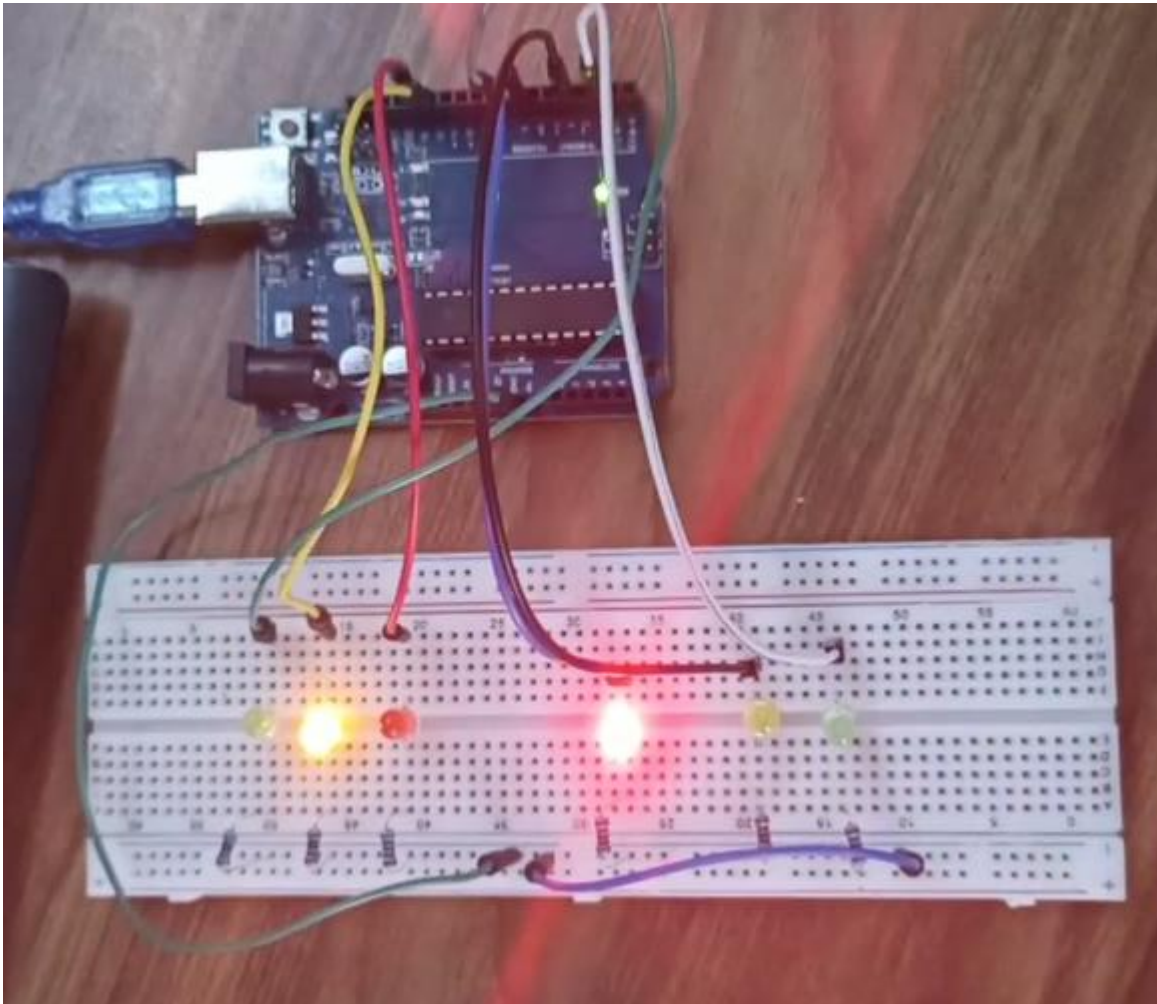
void loop() {
  digitalWrite(pinLED1, HIGH);
  digitalWrite(pinLED6, HIGH);
  delay(500);
  digitalWrite(pinLED1, LOW);

  digitalWrite(pinLED2, HIGH);
  digitalWrite(pinLED6, HIGH);
  delay(250);
  digitalWrite(pinLED2, LOW);
  digitalWrite(pinLED6, LOW);

  digitalWrite(pinLED3, HIGH);
  digitalWrite(pinLED5, HIGH);
  delay(250);
  digitalWrite(pinLED3, LOW);
  digitalWrite(pinLED5, LOW);
}

```


Demostración de su funcionamiento:



Conclusiones:

La práctica realizada sobre el diseño y control de semáforos, tanto de forma individual como sincronizada, representó un ejercicio integral para aplicar y consolidar conocimientos de electrónica, programación y control de sistemas. En la primera parte, trabajar con un semáforo individual permitió explorar los principios básicos de funcionamiento, como la secuencia lógica de los estados (rojo, amarillo y verde), el control de tiempos y la implementación de los circuitos necesarios para garantizar su correcto desempeño. Este enfoque inicial fue crucial para comprender la estructura elemental del sistema.

Posteriormente, extender el proyecto a la sincronización de dos semáforos introdujo nuevos desafíos que simulan escenarios del mundo real, como la necesidad de coordinar múltiples dispositivos que operan en conjunto para regular el tráfico en intersecciones. Este paso permitió trabajar en la gestión de sistemas concurrentes, desarrollando algoritmos que aseguren una sincronización precisa y eviten conflictos entre los semáforos, como la posibilidad de señales contradictorias. También fue importante considerar aspectos como la optimización de tiempos y la adaptación de la lógica para distintas condiciones del entorno.

Además, la práctica resaltó la importancia de la planificación previa y el diseño estructurado, tanto a nivel de hardware como de software, ya que cualquier error en la secuencia lógica o en el ensamblaje de los circuitos podría resultar en fallos significativos en el funcionamiento del sistema. A lo largo de la actividad, se enfatizó la necesidad de realizar pruebas exhaustivas para validar el correcto desempeño del programa y garantizar la fiabilidad del sistema.