

Semáforo Digital

Laboratorio de Sistemas Electrónicos Digitales
GRUPO H

Elena Del Río Galera
Alicia Gordo Azabal
Andrea López Recio
Yolanda Lillo Mata

ÁREA DE TECNOLOGÍA ELECTRÓNICA-URJC

- LINK DIRECTO A SIMULACION EN TINKERCAD:

<https://www.tinkercad.com/things/5v3Xjto4KJ6-semaforostarwars/editel?sharecode=CHR8Pf0TVKPzueDwi9vzmP1apC9dkSStHrPnp62tO1s=>

- Sección 2-Semáforo: Tabla de Verdad que represente el funcionamiento de tu código y puertas lógicas asociadas.

E	B	S1	S2	S3	S4	S5	SIGNIFICADO
0	X	0	0	0	0	0	El circuito no tiene corriente. (Aunque la entrada sea 0/1 la salida siempre va a ser 0).
1	0	0	0	1	0	0	ESTADO 1 → Circuito con corriente. Botón sin pulsar. El semáforo está en verde y los vehículos pueden circular.
1	1	1	0	0	X	0	ESTADO 2 → Circuito con corriente. Botón pulsado. El semáforo está fijo en azul ("peatón espere") pero intermitente (0/1) en amarillo ("indica reducir la velocidad a los vehículos").
1	1	1	0	0	1	0	ESTADO 3 → Circuito con corriente. Botón pulsado. El semáforo está fijo en azul ("peatón espere") y fijo también en amarillo ("los vehículos deben detenerse porque está a punto de ponerse en rojo").
1	1	0	1	0	0	1	ESTADO 4 → Circuito con corriente. Botón pulsado. El semáforo está en rojo ("vehículos detenidos") y buzzer sonando ("suena una melodía que indica que los peatones pueden cruzar").

Significado de las letras:

E= Enable
 B = Botón (Entrada)
 S1 = Led azul
 S2 = Led rojo
 S3= Led verde
 S4= Led amarillo
 S5 = Buzzer

No se puede realizar el circuito de puertas lógicas en nuestro semáforo, ya que sólo tenemos una entrada (botón). Si hubiésemos hecho un semáforo diferente con respecto al que nos ha dado el profesor podríamos haber hecho el circuito ya que habría más entradas (botón, peatones y vehículos).

- **Sección 3-Semáforo : El código de tu fichero de Arduino copiado dentro del mismo documento.**

```
#define c3 7634 //Definimos la frecuencia de cada nota musical.
#define d3 6803
#define e3 6061
#define f3 5714
#define g3 5102
#define a3 4545
#define b3 4049
#define c4 3816 // 261 Hz
#define d4 3401 // 294 Hz
#define e4 3030 // 329 Hz
#define f4 2865 // 349 Hz
#define g4 2551 // 392 Hz
#define a4 2272 // 440 Hz
#define a4s 2146
#define b4 2028 // 493 Hz
#define c5 1912 // 523 Hz
#define d5 1706
#define d5s 1608
#define e5 1517 // 659 Hz
#define f5 1433 // 698 Hz
#define g5 1276
#define a5 1136
#define a5s 1073
#define b5 1012
#define c6 955
#define R 0

// Marcha Imperial
int melody1[] = { a4, R, a4, R, a4, R, f4, R, c5, R, a4, R, f4, R, c5, R, a4, R, e5, R, e5,
R, e5, R, f5, R, c5, R, g5, R, f5, R, c5, R, a4, R};
int beats1[] = { 50, 20, 50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 80, 50, 20,
50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 40};

// Melodía Star Wars Clásica
int melody2[] = { f4, f4, f4, a4s, f5, d5s, d5, c5, a5s, f5, d5s, d5, c5, a5s, f5, d5s, d5,
d5s, c5};
int beats2[] = { 21, 21, 21, 128, 128, 21, 21, 21, 128, 64, 21, 21, 21, 128, 64, 21,
21, 21, 128};
```

```

int button =2;           //Definimos todos los valores necesarios, leds, boton...
int speakerpin=3;
int ledrojo=5;
int ledamarillo =6;
int ledverde=7;
int ledazul=8;
int val=0;           //Esta variable la usamos para leer el valor del boton.
int MAX_COUNT = sizeof(melody1) / 2; //Variables para la melodía.
long tempo = 10000;
int pause = 1000;
int rest_count = 50;
int toneM = 0;
int beat = 0;
long duration = 0;
int potVal = 0;
int speakerOut = 3; //Buzzer, altavoz.

```

```

void setup()
{
  Serial.begin(9600);           //Abre el puerto serie, velocidad 9600 bps.
  pinMode(button, INPUT);       //Definimos el botón como única entrada.
  pinMode(ledrojo, OUTPUT);     //Definimos los leds y el buzzer como salidas.
  pinMode(ledamarillo, OUTPUT);
  pinMode(ledverde, OUTPUT);
  pinMode(ledazul, OUTPUT);
  pinMode(speakerOut, OUTPUT);
}

```

```

void playTone() {           //Procedimiento para hacer la base de la melodía

```

```

  long elapsed_time = 0;
  if (toneM > 0) {
    while (elapsed_time < duration) {
      digitalWrite(speakerOut,HIGH); //Encender el buzzer
      delayMicroseconds(toneM / 2);
      digitalWrite(speakerOut, LOW); //Apagar el buzzer
      delayMicroseconds(toneM / 2);
      elapsed_time += (toneM);
    }
  } else {
    for (int j = 0; j < rest_count; j++) {
      delayMicroseconds(duration);
    }
  }
}

```

```

void suenamelodia() { //Procedimiento que define la melodía, los tiempos...
// Melodia 1

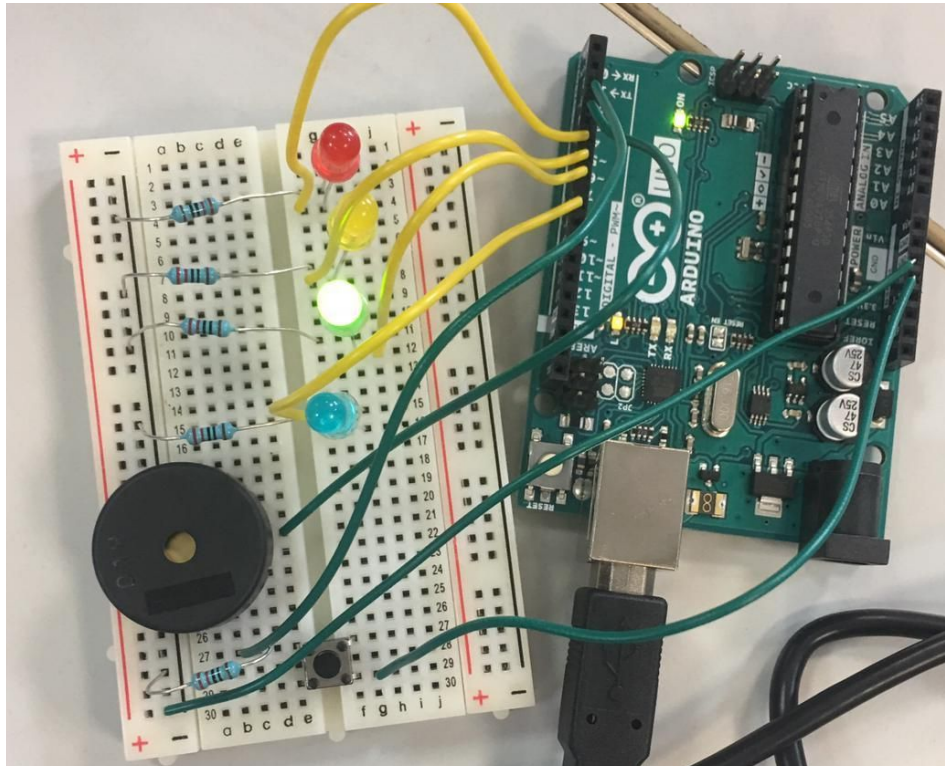
for (int i=0; i<MAX_COUNT; i++) {
    toneM = melody1[i];
    beat = beats1[i];
    duration = beat * tempo;
    playTone();
    delayMicroseconds(pause);
}

// Melodía 2
MAX_COUNT = sizeof(melody2) / 2;
for (int i = 0; i < MAX_COUNT; i++) {
    toneM = melody2[i];
    beat = beats2[i];
    duration = beat * tempo;
    playTone();
    delayMicroseconds(pause);
}
}

void loop()
{
    digitalWrite(ledverde, HIGH); //Led verde encendido
    val= digitalRead(button); //Leemos el valor del boton
    if (val==HIGH){ //Si el botón está encendido
        delay(2000);
        digitalWrite(ledverde, LOW); //Apagamos el led verde
        digitalWrite(ledazul, HIGH); //Encendemos el led azul
        for (int i= 0; i < 11; i++) {
            digitalWrite(ledamarillo, HIGH); //Durante la duración del bucle for
            delay(1000); //El led amarillo en intermitencia
            digitalWrite(ledamarillo, LOW);
            delay(1000);
        }
        digitalWrite(ledamarillo, HIGH); //Encendemos el led amarillo fijo
        delay(4000);
        digitalWrite(ledamarillo, LOW); //Apagamos el led amarillo
        digitalWrite(ledazul, LOW); //Apagamos el led azul
        digitalWrite(ledrojo, HIGH); //Encendemos el led rojo
        suenamelodia(); //Llamamos al procedimiento que empieza la melodía
        digitalWrite(ledrojo, LOW); //Apagamos el led rojo
    }
}

```

- Añadimos una imagen de nuestro circuito en placa, aunque mostramos ya el funcionamiento a la profesora en clase.



- Añadimos también la imagen de la simulación en Tinkercard:

