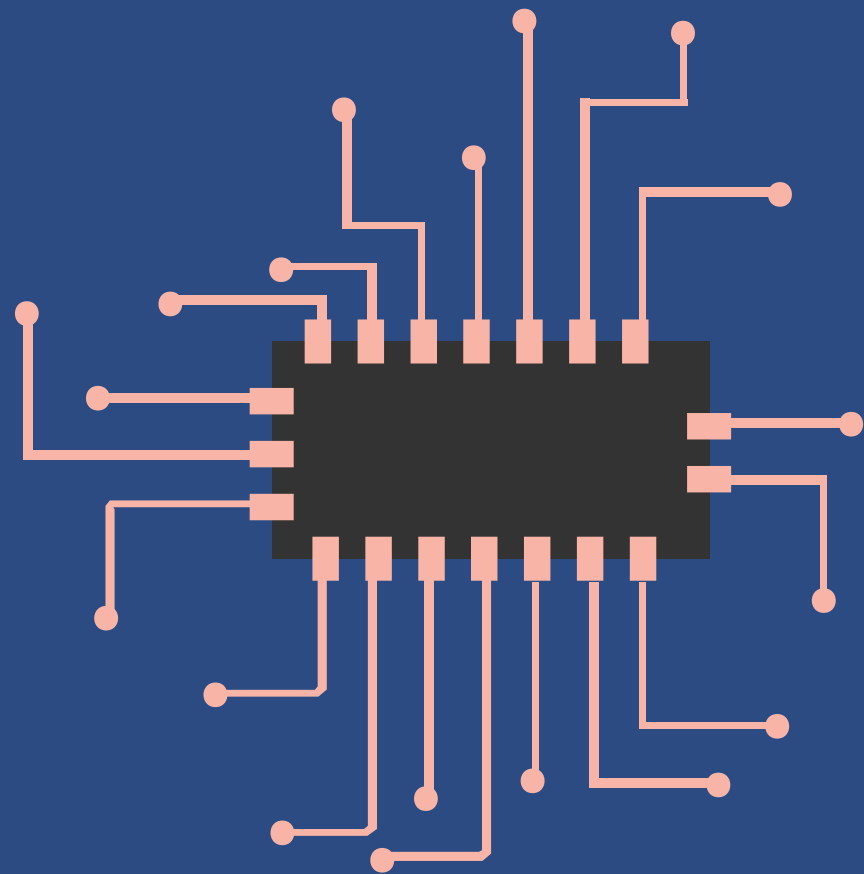


Proyecto Arduino

Daniel Molina Marín



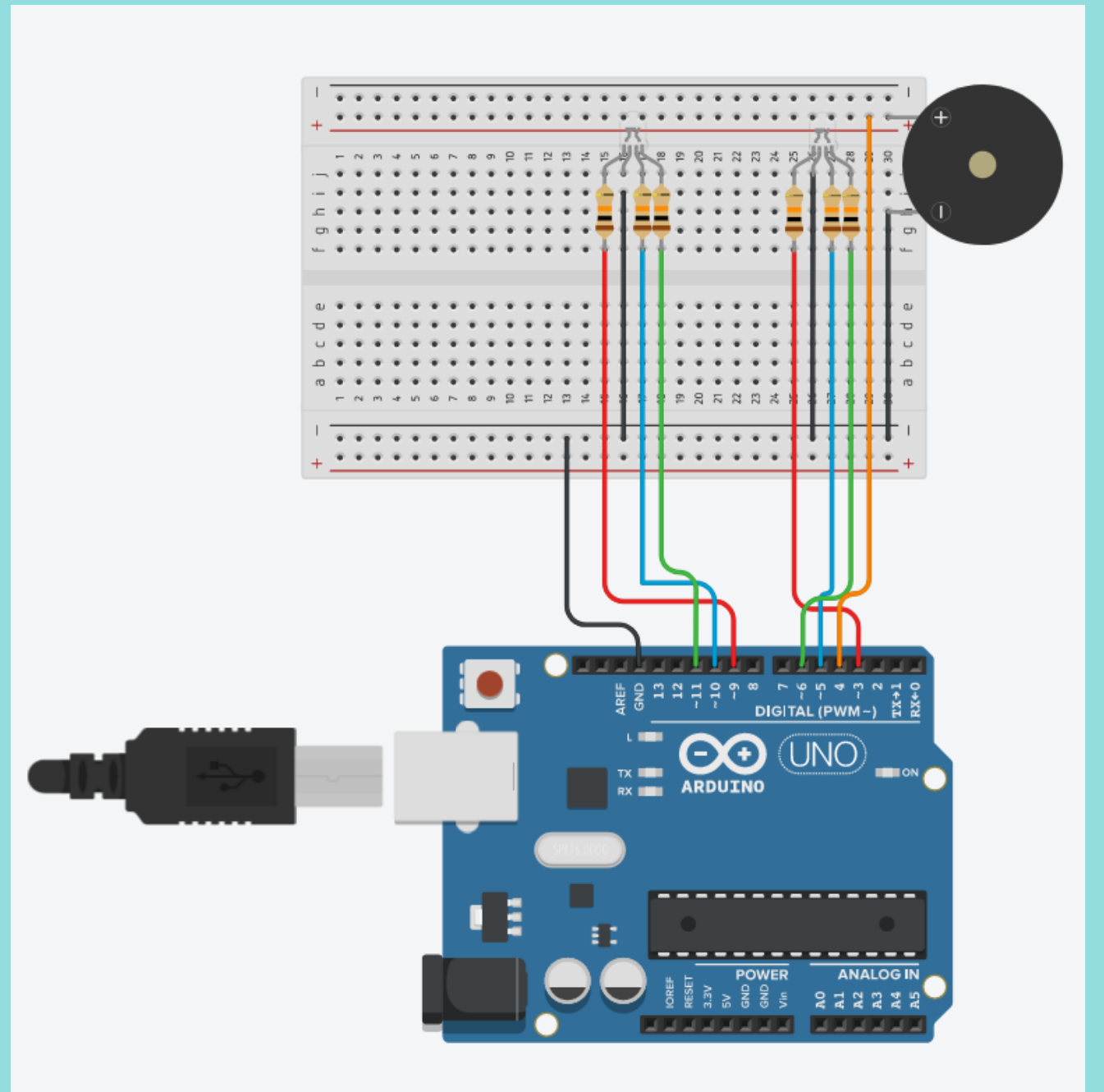
Índice

APARTADOS DEL PROYECTO

- Explicación
- Complicaciones
- Soluciones
- Presupuesto

Explicación

El proyecto trata de una lámpara con un zumbador integrado que emite un sonido cada 2 segundos.



Código

- Designación de variables
- Inicialización
- Bucle

Designación de variables

```
/*Proyecto final arduino*/  
#include <RGBMood.h> //incluimos la librería para que la RGB haga el efecto  
int redPin1 = 9;      //definimos la Led roja de la RGB1 en el pin 9  
int greenPin1 = 10;   //definimos la Led verde de la RGB1 en el pin 10  
int bluePin1 = 11;    //definimos la Led azul de la RGB1 en el pin 11  
int redPin2 = 6;      //definimos la Led roja de la RGB1 en el pin 6  
int greenPin2 = 5;    //definimos la Led verde de la RGB1 en el pin 5  
int bluePin2 = 3;     //definimos la Led azul de la RGB1 en el pin 3  
int buzzer =4;  
  
RGBMood mood1(redPin1, greenPin1, bluePin1); //declaramos el led1  
RGBMood mood2(redPin2, greenPin2, bluePin2); //declaramos el led2
```

Inicialización

```
void setup()
{
  /*Establecemos la led RGB 1 */
  mood1.setMode(RGBMood::RANDOM_HUE_MODE); // Establecemos el modo de cambio de color aleatorio
  mood1.setHoldingTime(2000); // Configuramos el tiempo de retención del color en milisegundos (2 segundos)
  mood1.setFadingSteps(150); // Configuramos el número de pasos de atenuación para una transición suave
  mood1.setFadingSpeed(50); // Configuramos la velocidad de atenuación para controlar la rapidez de la transición
  mood1.setHSB(random(359), 255, 255); // Establecemos los valores iniciales de HSB con un tono aleatorio, saturación y brillo máximos
  /*Establecemos la led RGB 2*/
  mood2.setMode(RGBMood::RANDOM_HUE_MODE); // Establecemos el modo de cambio de color aleatorio de la led RGB 2
  mood2.setHoldingTime(2000); // Configuramos el tiempo de retención del color en milisegundos (2 segundos) de la led RGB 2
  mood2.setFadingSteps(150); // Configuramos el número de pasos de atenuación para una transición suave de la led RGB 2
  mood2.setFadingSpeed(50); // Configuramos la velocidad de atenuación para controlar la rapidez de la transición de la led RGB 2
  mood2.setHSB(random(359), 255, 255); // Establecemos los valores iniciales de HSB con un tono aleatorio, saturación y brillo máximos de la led RGB 2
  |  pinMode(buzzer, OUTPUT);
}
```

Bucle

```
void loop()
{
  while (true)
  {
    for (int i = 0; i < 1000; i++) // Ejecutar 1000 veces el efecto de las tiras de LED
    {
      mood1.tick(); // Llamamos a la función tick de mood1 para avanzar en el efecto de iluminación de la led RGB 1
      mood2.tick(); // Llamamos a la función tick de mood2 para avanzar en el efecto de iluminación de la led RGB 2
      delay(10); // Pequeña pausa entre cada iteración del bucle secundario
    }
    tone(buzzer, 500); // Activamos el zumbador con una frecuencia de 500 Hz
    delay(2000); // Mantenemos el zumbador activado durante 2 segundos
    noTone(buzzer); // Apagamos el zumbador
  }
}
```

Problemas en el código

El principal problema que había era con el zumbador

```
void loop() {  
  digitalWrite(redPin1, LOW);  
  digitalWrite(greenPin1, LOW);  
  digitalWrite(bluePin1, LOW);  
  mood2.tick();  
  delay(5000);  
  digitalWrite(buzzerPin, HIGH);  
  delay(500);  
  digitalWrite(buzzerPin, LOW);  
  mood1.setHSB(random(359), 255, 255);  
  delay(5000);  
}
```


Soluciones

Encontré la solución de la siguiente forma

```
void loop() {
```

```
    while (true) {
```

```
        for (int i = 0; i < 1000; i++) {
```

```
            mood1.tick();
```

```
            mood2.tick();
```

```
            delay(10)
```

```
        }
```

```
        tone(buzzer, 500);
```

```
        delay(2000);
```

```
        noTone(buzzer);
```

```
    }
```

```
}
```

PRL

En mi proyecto se pueden presentar los siguientes riesgos :

- Riesgos por origen eléctrico

posibilidad baja, severidad ligeramente dañino = Riesgo Trivial

- Riesgos por incendio

Posibiliadad, baja, severidad ligeramente dañini= Riesgo Trivial

