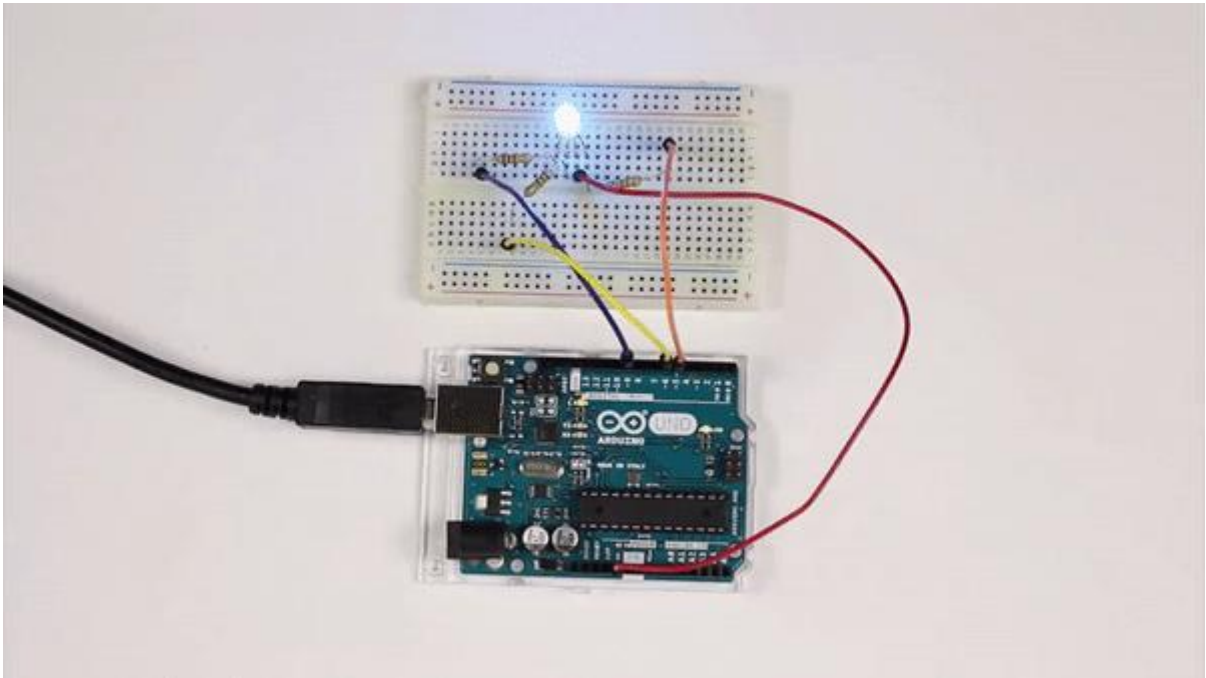


Arco Iris. Led RGB



En esta práctica vamos a visualizar los colores rojos, verde y azul y una mezcla de ellos gracias al uso de pines PWM de la placa de Arduino.

1. [Materiales](#)
2. [Diagrama de flujo](#)
3. [Esquema eléctrico](#)
4. [Programación en Arduino](#)
5. [Video resumen](#)

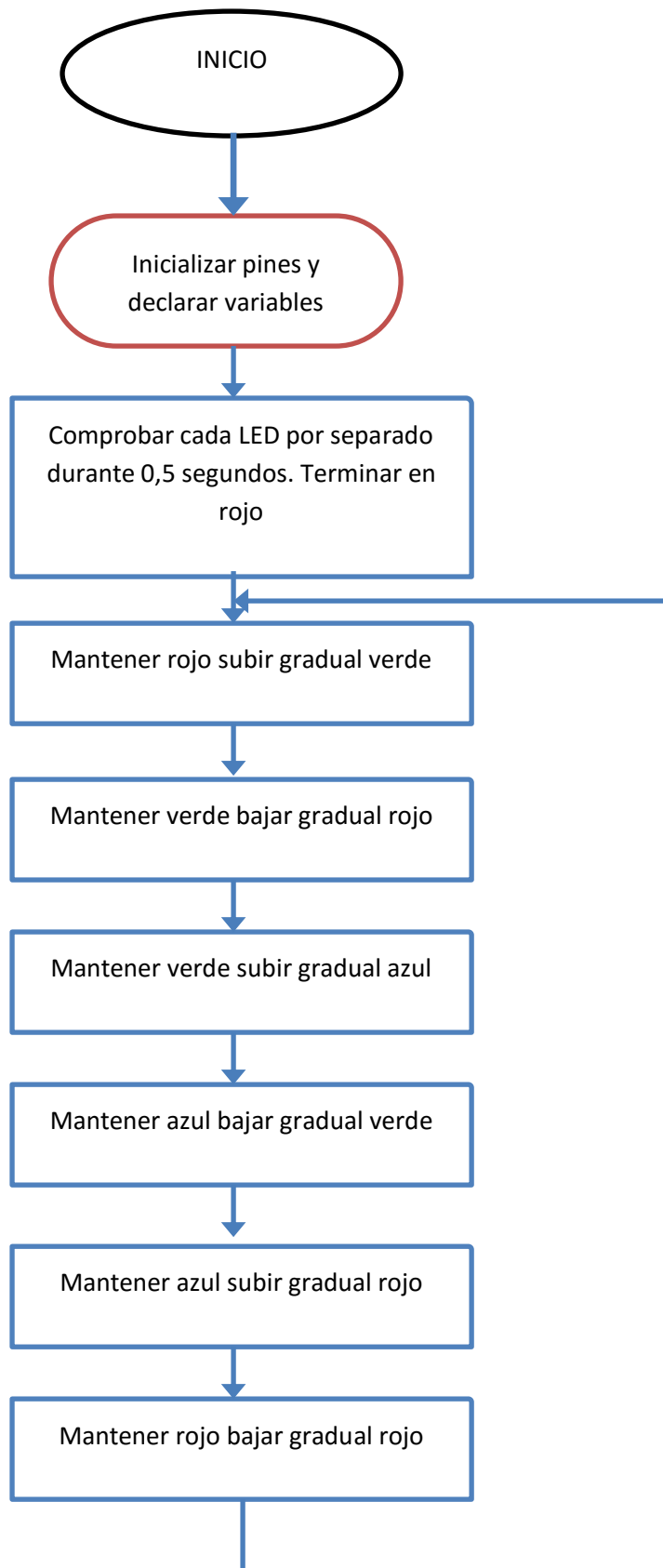
Materiales

Para llevar a cabo la práctica, vamos a necesitar los siguientes materiales:

- 1 Placa de Arduino UNO
- 1 Protoboard
- 4 latiguillos
- 1 Diodo Led RGB (cátodo común)
- 3 Resistencias

Diagrama de flujo

Antes de ponernos a desarrollar la práctica, conviene realizar un diagrama de flujo para entender la lógica de la programación.



Esquema eléctrico

Cuando conectamos un componente electrónico a una placa de Arduino, se ejerce sobre él una diferencia de potencial de 5V en ámbos extremos, es decir, si conectamos el Led a una placa de Arduino, estará recibiendo una tensión de 5V, que es la tensión operativa que ejerce Arduino en sus pines.

Diodo Led RGB (cátodo común)	
Polarizado	Sí
Diámetro	5mm
Intensidad de Corriente	20mA
Tensión en Led (rojo)	2,1V
Tensión en Led (verde)	3,3V
Tensión en Led (azul)	3,3V

Como se puede observar en la tabla anterior, el Led RGB (ánodo común) que estamos utilizando para la práctica admiten una tensión máxima de 2,1V para la patilla del Led rojo y 3,3V para las patillas del Led verde y azul.

Para evitar que se puedan dañar tendremos que colocarle una resistencia al circuito. Para ello, vamos a calcular el valor de la resistencia siguiendo la *Ley de Ohm* y teniendo en cuenta los datos de la tabla anterior.

La tensión en la patilla del Led rojo tiene que ser 2,1V. Si el pin de Arduino da 5V, la tensión que debe circular por la resistencia es $5V - 2,1V = 2,9V$. De igual manera, la tensión en las patillas del Led verde y azul tiene que ser 3,3V. Si el pin de Arduino da 5V, la tensión que debe circular por la resistencia en ambas patillas es $5V - 3,3V = 1,7V$.

Por otro lado, la intensidad que circula por el Led es de 20mA. A diferencia de la tensión, la intensidad se mantiene constante durante el circuito.

LED ROJO

$$V = 2,9V$$

$$I = 20mA$$

$$V = I \times R ; R = V / I$$

$$R = 2,9V / 0,02A = 145\Omega$$

--

LED VDERDE Y AZUL

$$V = 1,7V$$

$$I = 20mA$$

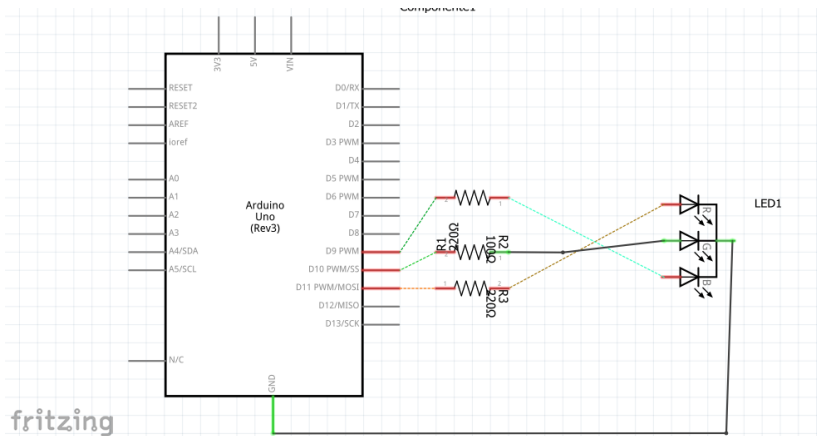
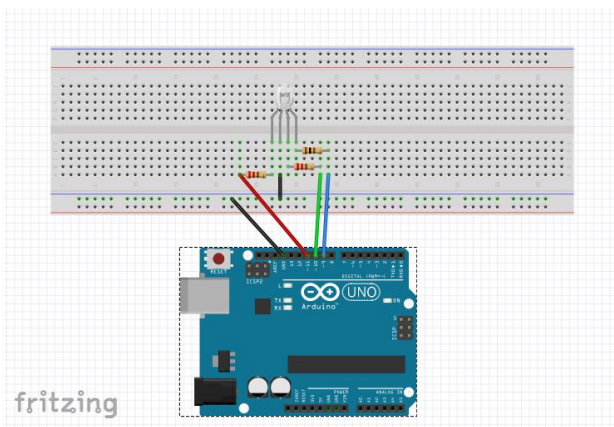
$$V = I \times R ; R = V / I$$

$$R = 1,7V / 0,02A = 85\Omega$$

Redondeamos los resultados obtenidos a un valor de resistencia por encima de su resistencia ideal, obtenemos una resistencia de 220Ω y 2 resistencias de 100Ω .

Una característica de los Diodos RGB de ánodo común, es que al alimentar el mismo sobre el ánodo, el led quedará apagado alimentando los pines R, G y B.

El siguiente paso será conectar los diferentes componentes sobre la placa de prototipado siguiendo el esquema eléctrico.



Programación en Arduino

```
//Declaración de los pines:  
int LEDr=11; //Color rojo pin 11  
int LEDg=10; //Color azul pin 10
```

```
int LEDb=9; //Color verde pin 9
```

```
//Declaración de las variables:
```

```
int rojo=0;
```

```
int verde=0;
```

```
int azul=0;
```

```
void setup() {
```

```
  //Iniciación del LED
```

```
  //Comprobación de cada color por separado
```

```
  setColor(255,0,0);
```

```
  delay(500);
```

```
  setColor(0,255,0);
```

```
  delay(500);
```

```
  setColor(0,0,255);
```

```
  delay(500);
```

```
  //Comprobación todos los colores
```

```
  setColor(255,255,255);
```

```
  delay(500);
```

```
  setColor(0,0,0);
```

```
  delay(1000);
```

```
  setColor(255,255,255);
```

```
  delay(500);
```

```
  //Dejamos el color rojo para empezar el loop
```

```
  setColor(255,0,0);
```

```
  delay(100);
```

```
}
```

```
void loop() {
```

```
  //Se mantiene el rojo al máximo mientras sube el verde
```

```
  for (int verde=0;verde<=255;verde++){
```

```
    setColor(255,verde,0);
```

```
    delay(20);
```

```
  }
```

```
  //Se mantiene el verde al máximo mientras baja el rojo
```

```
  for (int rojo=255;rojo>=0;rojo--){
```

```
    setColor(rojo,255,0);
```

```
    delay(20);
```

```
  }
```

```
  //Se mantiene el verde al máximo mientras sube el azul
```

```
  for (int azul=0;azul<=255;azul++){
```

```
    setColor(0,255,azul);
```

```
    delay(20);
```

```
  }
```

```
  //Se mantiene el azul al máximo mientras baja el verde
```

```
  for (int verde=255;verde>=0;verde--){
```

```
    setColor(0,verde,255);
```

```
    delay(20);
```

```
  }
```

```
  //Se mantiene el azul al máximo mientras sube el rojo
```

```
  for (int rojo=0;rojo<=255;rojo++){
```

```
    setColor(rojo,0,255);
```

```
    delay(20);
```

```
  }
```

```
  //Se mantiene el rojo al máximo mientras baja el azul
```

```
  for (int azul=255;azul>=0;azul--){
```

```
    setColor(255,0,azul);
```

```
    if(azul==0){
```

```
      delay(100);
```

```
    }
```

```
    else{  
        delay(20);  
    }  
}  
}
```

```
void setColor(int rojo,int verde,int azul)  
{  
    analogWrite(LEDr,rojo);  
    analogWrite(LEDg,verde);  
    analogWrite(LEDb,azul);  
}
```

Video resumen