

PDIH - UGR

Práctica 3 - Ángel Gómez Ferrer

Índice

Práctica 3 - Ángel Gómez Ferrer	1
Índice	1
Entorno	1
Ejercicio 1	1
Ejercicio 2	3
Ejercicio Opcional 1	4

Nota:

Hay un video en el repositorio para cada ejercicio en el que se comprueba el funcionamiento

Entorno

Ya que en mi caso no dispongo de ningún arduino real, entre las opciones vistas en el seminario he elegido las siguiente para la realización de la práctica:

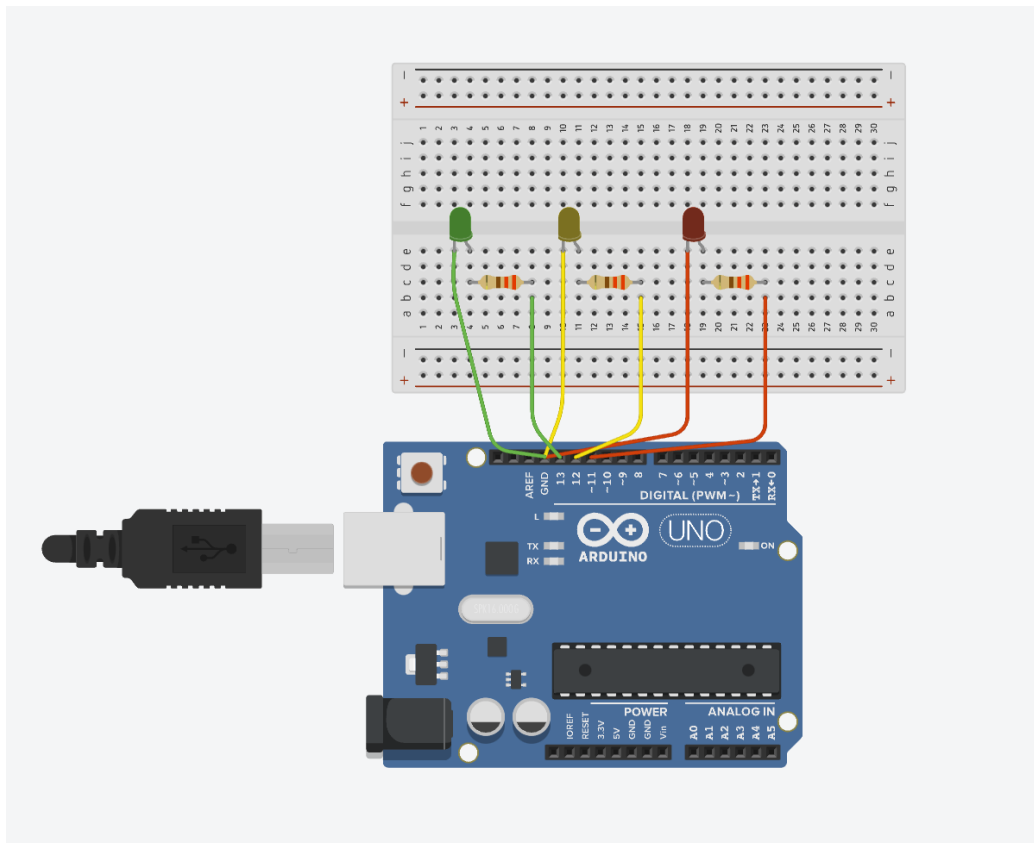
<https://www.tinkercad.com>

Ejercicio 1

Materiales:

- 3 Resistencias de $220\ \Omega$
- 3 LEDs(rojo, verde y amarillo)
- 1 Arduino Uno R3

Para este ejercicio simplemente he unido los 3 leds con una de las patas a tierra del arduino y a la otra una resistencia (en total se han usado 3 de $220\ \Omega$) para que el led no se queme y a su vez la resistencias a los pines (verde) 13, (amarillo) 12 y (rojo) 11.

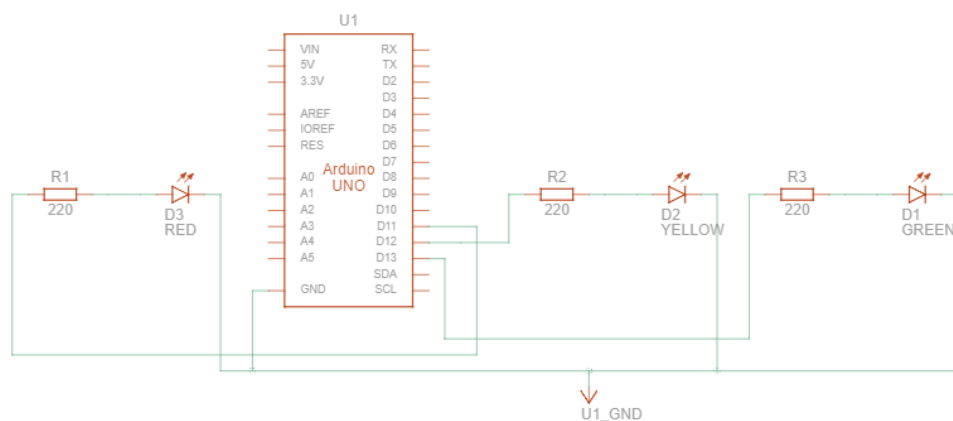


Tras montar el arduino con los dispositivos mencionados esto ha sido el código utilizado:

```
1 // C++ code
2 //
3 void setup()
4 {
5     pinMode(13, OUTPUT);
6     pinMode(12, OUTPUT);
7     pinMode(11, OUTPUT);
8 }
9
10 void loop()
11 {
12     digitalWrite(13, HIGH);
13     digitalWrite(12, LOW);
14     digitalWrite(11, LOW);
15     delay(1500); // Wait for 1500 millisecond(s)
16     digitalWrite(13, LOW);
17     digitalWrite(12, HIGH);
18     digitalWrite(11, LOW);
19     delay(1500); // Wait for 1500 millisecond(s)
20     digitalWrite(13, LOW);
21     digitalWrite(12, LOW);
22     digitalWrite(11, HIGH);
23     delay(1500); // Wait for 1500 millisecond(s)
24 }
```

Donde primero inicializamos los pines como salidas y aplicamos las salidas para encender el led correspondiente dependiendo de su pin, también aplicamos un delay entre cada acción para conseguir el resultado deseado.

Vista esquemática:

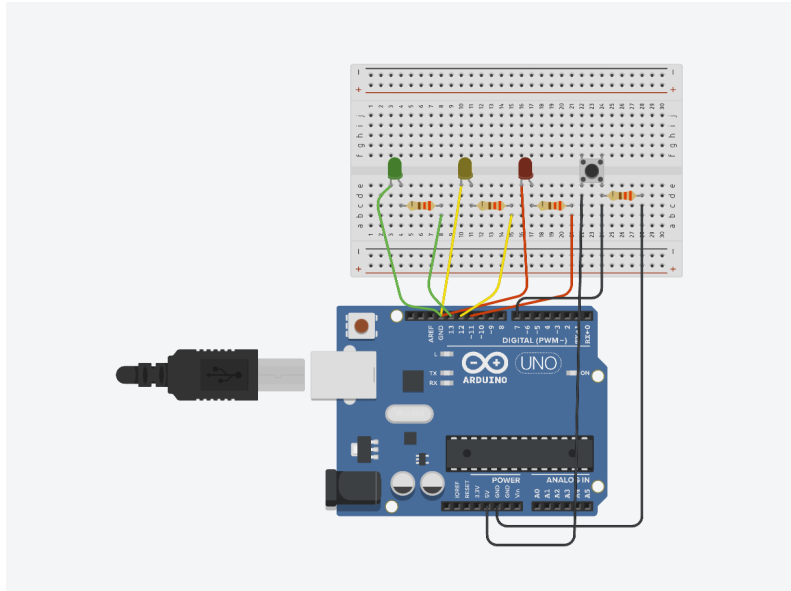


Ejercicio 2

Hemos partido del esquema del ejercicio anterior.

Materiales:

- 4 Resistencias de 220 Ω
- 3 LEDs(rojo, verde y amarillo)
- 1 Arduino Uno R3
- 1 interruptor



Los cambios que se han hecho son añadir el pulsador el cual lo hemos conectado a la entrada digital 7 y a la de corriente de 5V además de a tierra.

El código es el siguiente:

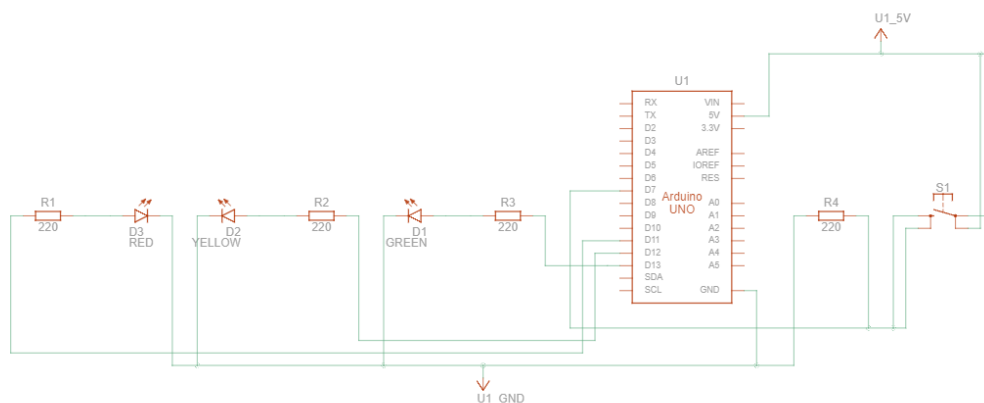
```

1  // C++ code
2  //
3  int state;
4  void setup()
5  {
6      pinMode(7, INPUT);
7      pinMode(13, OUTPUT);
8      pinMode(12, OUTPUT);
9      pinMode(11, OUTPUT);
10 }
11
12 void loop()
13 {
14     state = digitalRead(7);
15
16     if (state == HIGH) {
17         digitalWrite(13, LOW);
18         digitalWrite(12, LOW);
19         digitalWrite(11, HIGH);
20     } else {
21         delay(1500); // Wait for 1500 millisecond(s)
22         digitalWrite(13, HIGH);
23         digitalWrite(12, LOW);
24         digitalWrite(11, LOW);
25         delay(1500); // Wait for 1500 millisecond(s)
26         digitalWrite(13, LOW);
27         digitalWrite(12, HIGH);
28         digitalWrite(11, LOW);
29     }
30 }
31
32 }

```

Simplemente inicializamos con input la entrada 7, es decir la del interruptor, y en cada iteración del bucle lo comprobamos el estado del interruptor, para conseguir de esta forma el comportamiento deseado.

Vista esquemática:

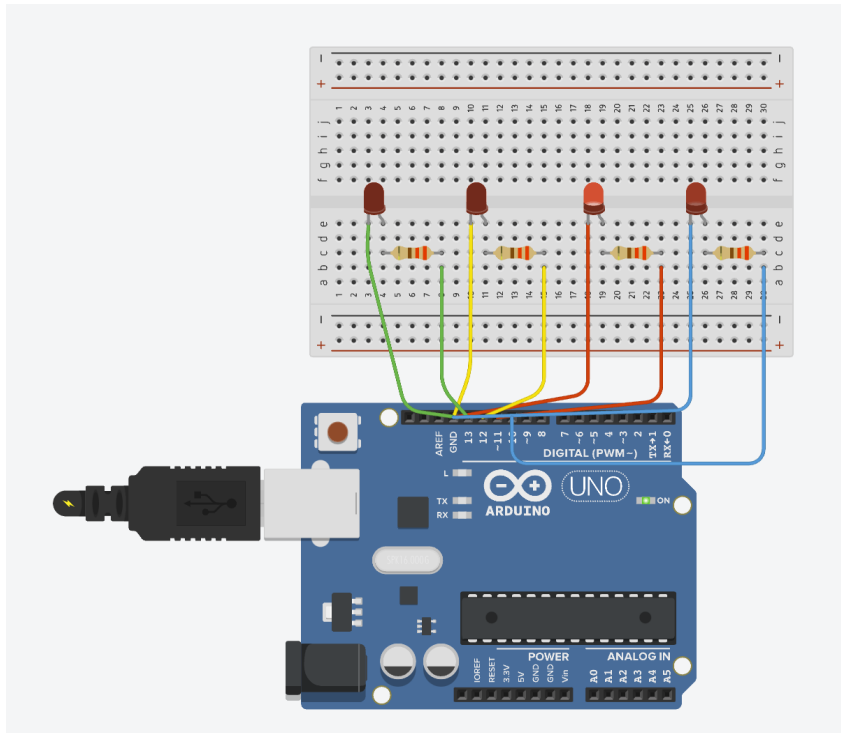


Ejercicio Opcional 1

Hemos partido del esquema del primer ejercicio y añadido un led más con su correspondiente resistencia.

Materiales:

- 4 Resistencias de 220 Ω
- 4 LEDs (rojos)
- 1 Arduino Uno R3



Para el código mostrado abajo simplemente inicializamos como output el pin número 10 en el que está el nuevo led y realizamos primero el recorrido normal cada 100 ms encendiendo el siguiente led en la secuencia y tras esto añadimos que haga el mismo recorrido a la inversa solamente hemos tenido que añadir dos estados de los LEDs más ya que no queremos que cada LED final e inicial repita su estado de encendido.

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(13, OUTPUT);
6   pinMode(12, OUTPUT);
7   pinMode(11, OUTPUT);
8   pinMode(10, OUTPUT);
9 }
10
11 void loop()
12 {
13   digitalWrite(13, HIGH);
14   digitalWrite(12, LOW);
15   digitalWrite(11, LOW);
16   digitalWrite(10, LOW);
17   delay(150);
18   digitalWrite(13, LOW);
19   digitalWrite(12, HIGH);
20   digitalWrite(11, LOW);
21   digitalWrite(10, LOW);
22   delay(150);
23   digitalWrite(13, LOW);
24   digitalWrite(12, LOW);
25   digitalWrite(11, HIGH);
26   digitalWrite(10, LOW);
27   delay(150);
28   digitalWrite(13, LOW);
29   digitalWrite(12, LOW);
30   digitalWrite(11, LOW);
31   digitalWrite(10, HIGH);
32   delay(150); // Recorrido inverso
33   digitalWrite(13, LOW);
34   digitalWrite(12, LOW);
35   digitalWrite(11, HIGH);
36   digitalWrite(10, LOW);
37   delay(150);
38   digitalWrite(13, LOW);
39   digitalWrite(12, HIGH);
40   digitalWrite(11, LOW);
41   digitalWrite(10, LOW);
42   delay(150);
43 }
```

Vista esquemática:

