Práctica nº 3: Efecto parpadeo ("El coche fantástico")

Esta práctica es una ampliación del programa de led parpadeante.

Ahora vamos a tratar de encender y apagar 6 leds secuencialmente imitando a las luces delanteras del "Coche fantástico".

Los leds deberán encenderse uno a uno de derecha a izquierda, y luego de izquierda a derecha, sucesivamente. Se ha de cumplir con dos condiciones:

1. Siempre tiene que haber al menos un led encendido, para evitar parpadeos molestos. Por ello antes de apagar un led habrá que encender el que le sigue en la secuencia.

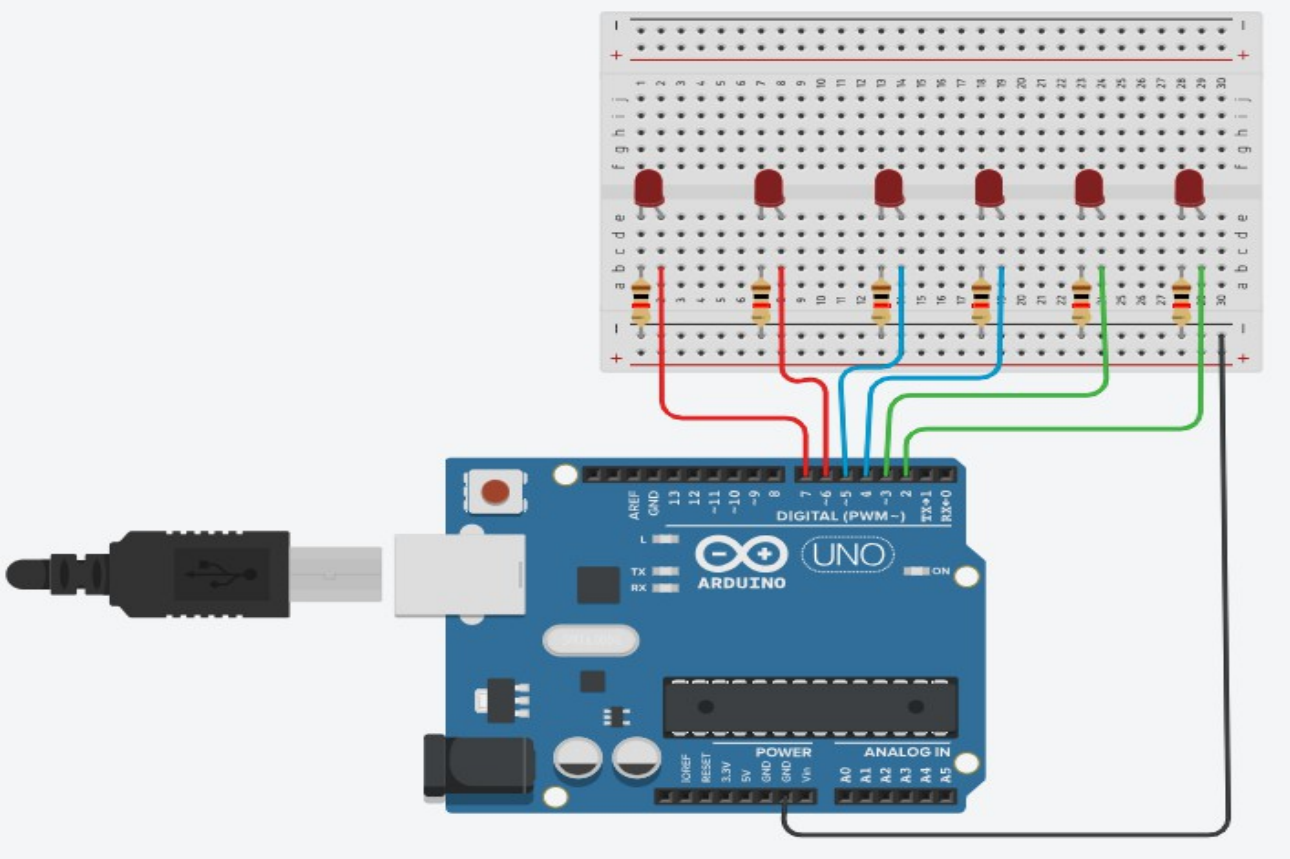
2. Cada led tiene que estar encendido el mismo tiempo que los demás y consecutivamente.

MATERIALES

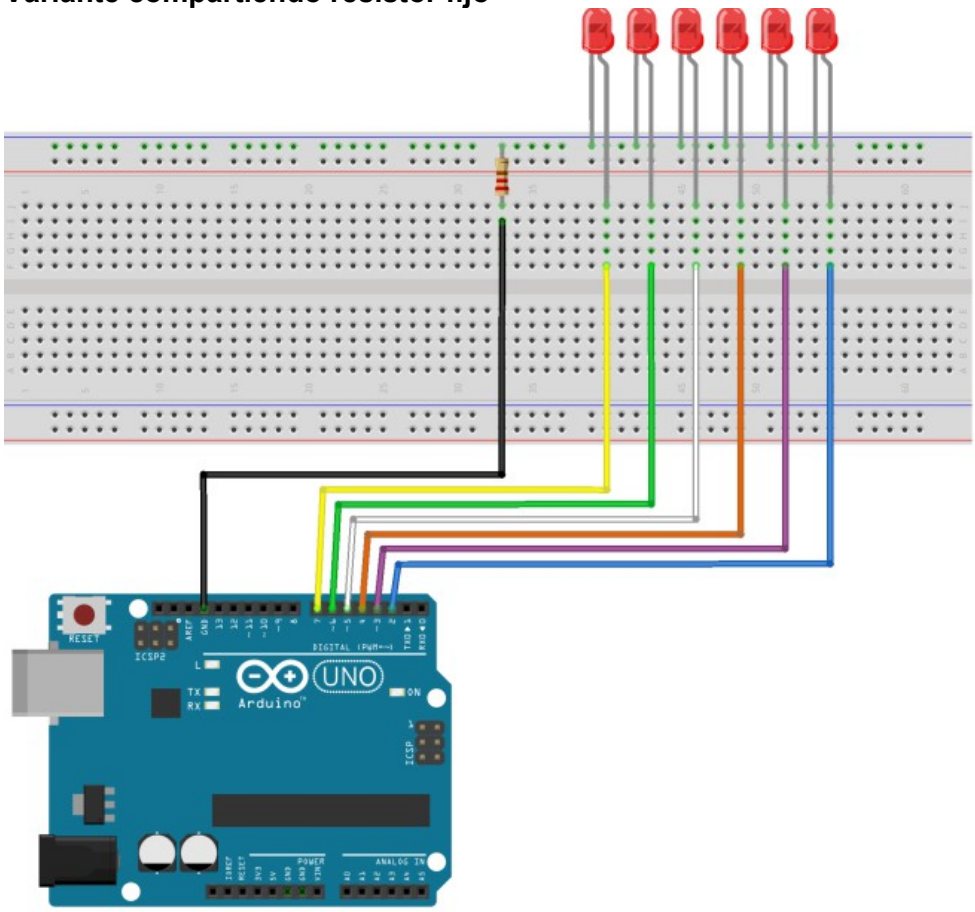
- Arduino UNO.
- Cable USB tipo A-B.
- 6 LED de 3mm
- 6 Resistencias de 220Ω.
- 1 placa Protoboard.
- Cables de conexión.

MONTAJE

Para este montaje utilizaremos 6 LEDs, conectados en pines consecutivos (del 2 al 7, por ejemplo).



- **Variante compartiendo resistor fijo**



PROGRAMACIÓN

Aquí tienes cuatro versiones diferentes del programa. Comprueba las diferencias en el código y en el funcionamiento.

1.- Coche fantástico (Versión 1.Definiendo variables)

```
int pin2 = 2;
int pin3 = 3;
int pin4 = 4;
int pin5 = 5;
int pin6 = 6;
int pin7 = 7;
int espera = 70; // El tiempo que se esperará

void setup() {
  pinMode(pin2, OUTPUT); // Configuración de los PIN como salida
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(pin5, OUTPUT);
  pinMode(pin6, OUTPUT);
  pinMode(pin7, OUTPUT);
}

void loop() {
  digitalWrite(pin2, HIGH); // Enciende y apaga secuencialmente los LEDs
  delay(espera);
  digitalWrite(pin2, LOW);
  delay(espera);
  digitalWrite(pin3, HIGH);
  delay(espera);
  digitalWrite(pin3, LOW);
  delay(espera);
  digitalWrite(pin4, HIGH);
  delay(espera);
  digitalWrite(pin4, LOW);
  delay(espera);
  digitalWrite(pin5, HIGH);
  delay(espera);
  digitalWrite(pin5, LOW);
  delay(espera);
  digitalWrite(pin6, HIGH);
  delay(espera);
  digitalWrite(pin6, LOW);
  delay(espera);
  digitalWrite(pin7, HIGH);
  delay(espera);
  digitalWrite(pin7, LOW);
  delay(espera);
  digitalWrite(pin6, HIGH);
  delay(espera);
  digitalWrite(pin6, LOW);
  delay(espera);
  digitalWrite(pin5, HIGH);
  delay(espera);
  digitalWrite(pin5, LOW);
  delay(espera);
  digitalWrite(pin4, HIGH);
  delay(espera);
  digitalWrite(pin4, LOW);
  delay(espera);
  digitalWrite(pin3, HIGH);
  delay(espera);
```

```
  digitalWrite(pin3, LOW);
  delay(espera);
}
```

2.- Coche fantástico. (Versión 2.Uso de Arrays [] para simplificar las definiciones iniciales)

```
/* El coche fantástico versión 2, utilizando Array para definir los pines

    < less than, es lo mismo que el signo <
    > greater-than, es lo mismo que el signo >
    ≤ stands for the less-than or equals sign ( <= )
    ≥ stands for the greater-than or equals sign ( >= )

*/

int pinArray[] = {2, 3, 4, 5, 6, 7};
int count = 0;
int timer = 15;
//timer marca como queremos que vaya de rápido la ráfaga de encendido-apagado de los
LEDS

void setup() {
  for (count = 0; count < 8; count++) {
    pinMode(pinArray[count], OUTPUT);
  }
}

void loop() {
  for (count = 0; count <8 ; count++) {
    digitalWrite(pinArray[count], HIGH);
    delay(timer);
    digitalWrite(pinArray[count + 1], HIGH);
    delay(timer);
    digitalWrite(pinArray[count], LOW);
    delay(timer * 2);
  }
  for (count = 7; count > 1; count--) {
    digitalWrite(pinArray[count], HIGH);
    delay(timer);
    digitalWrite(pinArray[count - 1], HIGH);
    delay(timer);
    digitalWrite(pinArray[count], LOW);
    delay(timer * 2);
  }
}
```

3.- Coche fantástico usando bucle FOR. (Versión 3)

```
void setup() {
  // initialize the digital pins as an output
    int i;
    for (i=2;i<8;++i) {         // Se repite desde que i=2 hasta 7, con incrementos de 1
      pinMode( i, OUTPUT) ;
  }
}

void loop() {
    int i;
    for (i=2;i<8;++i) {
       digitalWrite( i , HIGH) ;
           delay (500) ;
         digitalWrite( i , LOW);
           delay (500) ;
                }
}
```

Aqui ha aparecido una nueva estructura: **for (**incio**;** condición**;** incremento**) {** … **}**
La estructura for repite un número de veces las instrucciones que estén contenidas entre llaves, y la lógica que sigue es la siguiente:

Emplea una variable (en este caso i) que se inicia (asignándole un valor inicial, en este ejemplo i=7).

Dicha variable va incrementándose cada vez que se repite el for. El
incremento puede expresarse así:
i=i+5 //el valor de i se incrementa en 5
i+=5 //el valor de i se incrementa en 5 (es otra forma)
i=i+1 //el valor de i se incrementa en 1
i+=1 //el valor de i se incrementa en 1 (es otra forma)
i++ //el valor de i se incrementa en 1 (solo para incremento +1)
i=i-1 //el valor de i disminuye en 1
i-=1 //el valor de i disminuye en 1 (es otra forma)
i-- //el valor de i disminuye en 1 (solo para incremento -1)
i=i*3 //el valor de i se multiplica por 3
i*=3 //el valor de i se multiplica por 3 (es otra forma)
i=i/2 //el valor de i se divide entre 2
i/=2 //el valor de i se divide entre 2 (es otra forma)

El bucle for se repetirá siempre y cuando se siga cumpliendo la condición.
for(i=2; i<=8; ++ i) { **…** } hará un primer ciclo con el valor i=2. Después
de hacer todo lo que está entre las llaves, incrementara en 1 el valor de i. .Es i (ahora 3) menor que 8?
Como la respuesta es sí, volverá a hacer el bloque.
Cuando termine el ciclo con valor i=8, la i se incrementara en 1 (valdrá 9). En ese momento ya no cumplirá la condición y el programa se saldrá del bucle for.
Otra observación es que, como la variable i solo se va a utilizar dentro del for (y en ninguna otra parte del programa) puedo declararla en ese mismo momento:
For (int i=2; i<=11; ++ i) { **…** }

- **Ejercicio propuestos**

— Comprueba que cada uno de los montajes funcionan correctamente y que el código realiza lo especificado.

— Ahora, añade 3 leds al montaje y realiza modificaciones en cada una de las versiones del programa, para que funcione correctamente y se iluminen los 8 leds siguiendo la misma secuencia.