

Datos de los miembros del grupo

Fernando María del Castillo Belloso, fmdelcastillo

Luis Pérez González, luispgonzalez

Víctor Rodríguez Sánchez, victorrodriguezsanchez

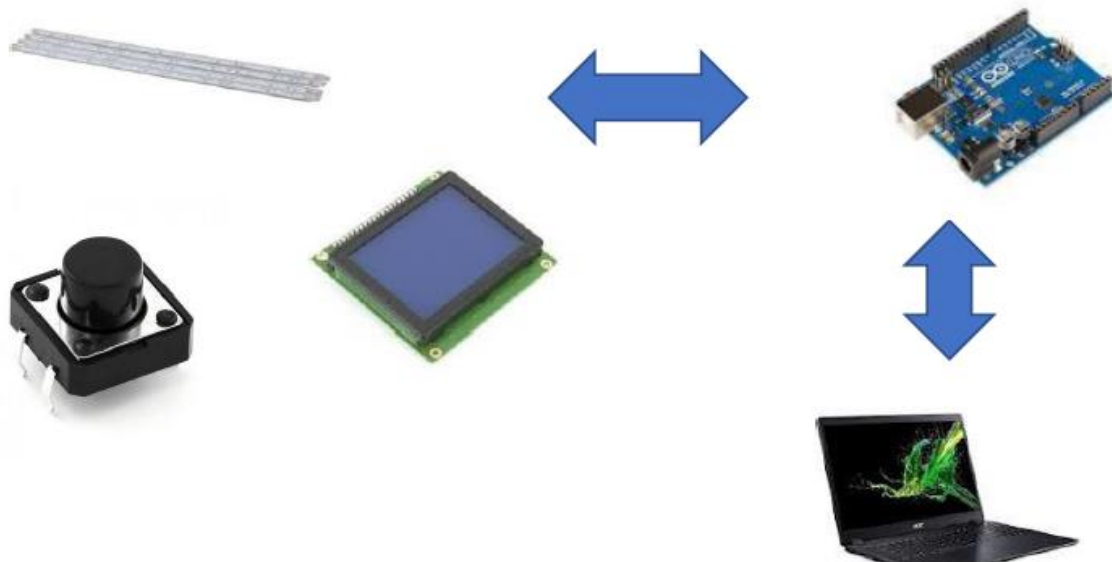
Título y resumen

El trabajo consiste en hacer una carrera con cintas led, habrá dos usuarios que serán los participantes, cada uno contara con un controlador con un pulsador que hará que avancen los leds, de tal modo que el que consiga encender antes el ultimo led de la su cinta, gana.

Requisitos funcionales

- 1º - Al iniciar la carrera, la pantalla mostrara una información para que ninguno de los participantes tenga ventaja sobre el otro.
- 2º - A su vez, en el inicio, en cada cinta LED se habrá borrado la última posición de la anterior carrera, y ambas estarán en la posición inicial.
- 3º - Cuando la partida comience, los usuarios podrán comenzar a utilizar sus mandos para que los LED de sus respectivas cintas avancen.

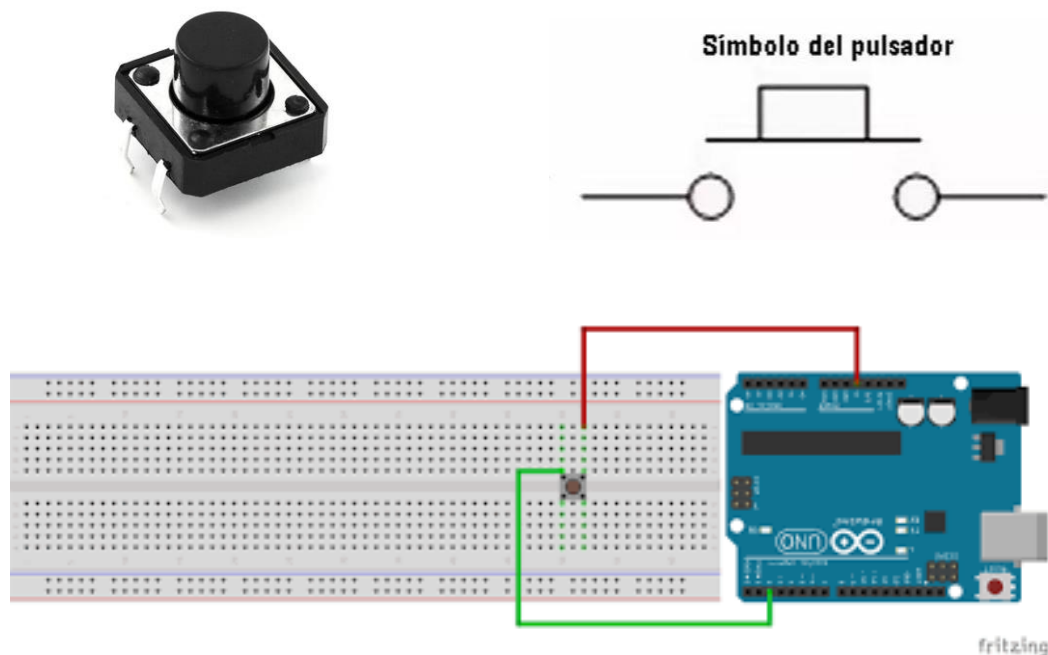
Hardware – Fundamentos técnicos



Pulsador

Un pulsador es un interruptor o switch cuya función es permitir o interrumpir el paso de la corriente eléctrica de manera momentánea a diferencia de un switch común, un pulsador solo realiza su trabajo mientras lo pegas presionado. Existen pulsadores NC (NC) y NA (NO), es decir, normalmente cerrados y normalmente abiertos.

Tiene 4 patillas, que son dos cables de cada lado, unidos en el interior de la caja de este pulsador.



```
const int inputPin = 2;

int value = 0;

void setup() {
  Serial.begin(9600);
  pinMode(inputPin, INPUT);
}

void loop(){
  value = digitalRead(inputPin); //lectura del pin

  //mandar mensaje a puerto serie en función del valor leído
  if (value == HIGH) { //Si el pulsador esta activado
    Serial.println("Encendido");
  }
  else { //El caso contrario, si el pulsador no esta activado
    Serial.println("Apagado");
  }
  delay(1000);
}
```

Cinta LED

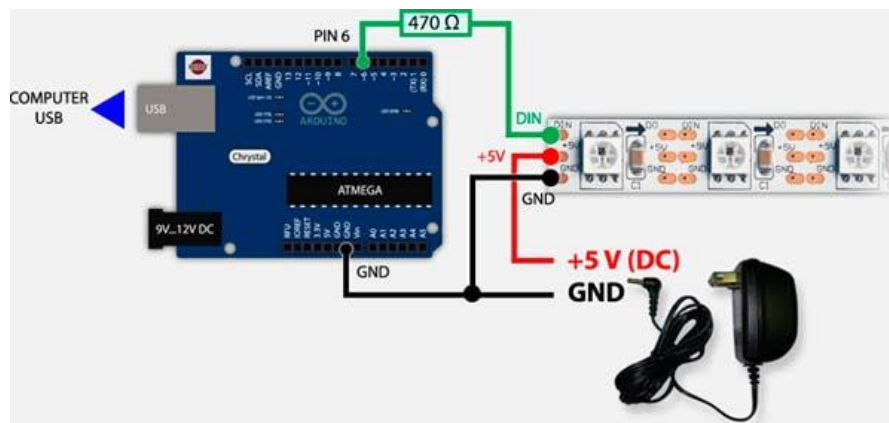
Todos los LED están conectados en cadena a un puerto del arduino u otro sistema que usemos. Cada led tiene una memoria de 3 bytes donde almacena su estado de los 3 led que está compuesto, este dato lo envía el arduino al primer led, este lo almacena, y una vez terminada su comunicación pasa a ser transmisor del siguiente dato al led siguiente.

Digamos que los datos de todos los leds se envían en serie uno detrás del otro y cada led almacena el que lleva su dirección y deja pasar el siguiente para el led correspondiente. Como esta transmisión se realiza a velocidades altas (unos 400Hz), no es perceptible por el ojo humano produciendo el efecto de simultaneidad de encendido de todos los LED.

Por supuesto la cantidad de leds no puede ser ilimitada, pero de esta manera se pueden controlar más de 1000 led a una frecuencia de refresco de 30 Hz que es suficiente para no ser preceptible por la retina del ojo humano.

La conexión con Arduino: La conexión de las tiras de led y arduino son muy sencillas ya que nos basta con la alimentación y un simple puerto de salida para enviar los datos a todos los LED interconectados entre sí.

Podremos observar en la tira de led 3 pines marcados con 5V y GND correspondientes a la alimentación y uno de datos marcado como Din.



```
1 // This is a demonstration on how to use an input device to trigger changes on your neo pixels.
2 // You should wire a momentary push button to connect from ground to a digital IO pin. When you
3 // press the button it will change to a new pixel animation. Note that you need to press the
4 // button once to start the first animation!
5
6 #include <Adafruit_NeoPixel.h>
7
8 #define BUTTON_PIN 2 // Digital IO pin connected to the button. This will be
9 // driven with a pull-up resistor so the switch should
10 // pull the pin to ground momentarily. On a high -> low
11 // transition the button press logic will execute.
12
13 #define PIXEL_PIN 6 // Digital IO pin connected to the NeoPixels.
14
15 #define PIXEL_COUNT 16
16
17 // Parameter 1 = number of pixels in strip, neopixel stick has 8
18 // Parameter 2 = pin number (most are valid)
19 // Parameter 3 = pixel type flags, add together as needed:
20 // NEO_RGB Pixels are wired for RGB bitstream
21 // NEO_GRB Pixels are wired for GRB bitstream, correct for neopixel stick
22 // NEO_KHZ400 400 KHz bitstream (e.g. FLORA pixels)
23 // NEO_KHZ800 800 KHz bitstream (e.g. High Density LED strip), correct for neopixel stick
24 Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, NEO_GRB + NEO_KHZ800);
```