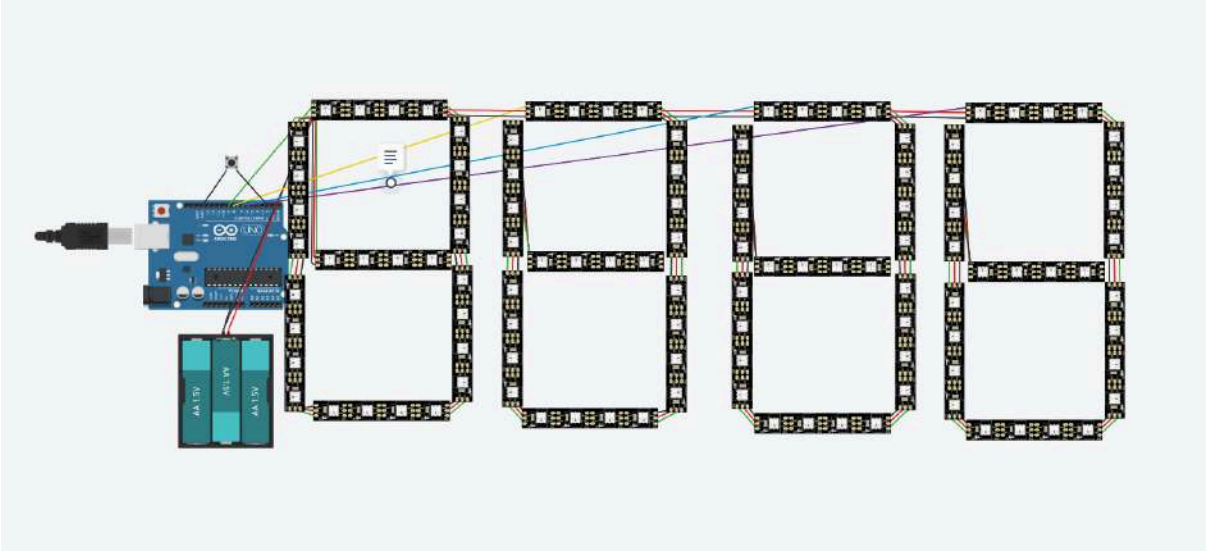
Proyecto Cronómetro de Carrera con Arduino y LEDs

Profesor: Juan Andrés Morales Cordovilla



Material necesario:

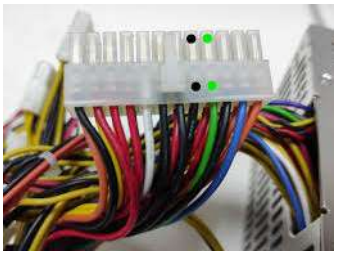
- Fuente de alimentación vieja de un PC con salida molex hembra 5 Volt y 18 Amperios (es algo típico). Interruptor.

- Conector Molex macho para alimentar los leds (por seguridad este es el macho):

- Cables gordos de corriente, 5 metros de luces leds Neopixel, Arduino Uno con USB largo (de impresora), conectores L para esquinas de píxeles, pestañas, cartón de cajas grandes (ej. de una TV), soldadura de estaño.

Ejecución y planos:

- Conectar el interruptor al verde y negro para puentear la fuente y encenderla con el interruptor.



- Crear línea de alimentación 5 Voltios (con su tierra) para los LEDs y conectar las señales a pines 6,7,8 y 9 de Arduino. Importante conectar tierra de alimentación con GND de Arduino. Para evitar interferencias es mejor alimentar Arduino por USB a un portátil.

- Poner el siguiente código en Arduino:

```
#include <Adafruit_NeoPixel.h>

#define NUM_LEDS 75
//#define PixSegm 4
#define PixSegm 11
#define NUM_TIRAS 4


// Declaración de un array de tiras Neopixel
Adafruit_NeoPixel strips[NUM_TIRAS] = {
 Adafruit_NeoPixel(NUM_LEDS, 6, NEO_GRB + NEO_KHZ800),
 Adafruit_NeoPixel(NUM_LEDS, 7, NEO_GRB + NEO_KHZ800),
 Adafruit_NeoPixel(NUM_LEDS, 8, NEO_GRB + NEO_KHZ800),
 Adafruit_NeoPixel(NUM_LEDS, 9, NEO_GRB + NEO_KHZ800)
};

//const int Pix0Segm[7] = {0, 4, 8, 12, 16, 20, 24};
const int Pix0Segm[7] = {0, 12, 23, 34, 44, 55, 65};
```

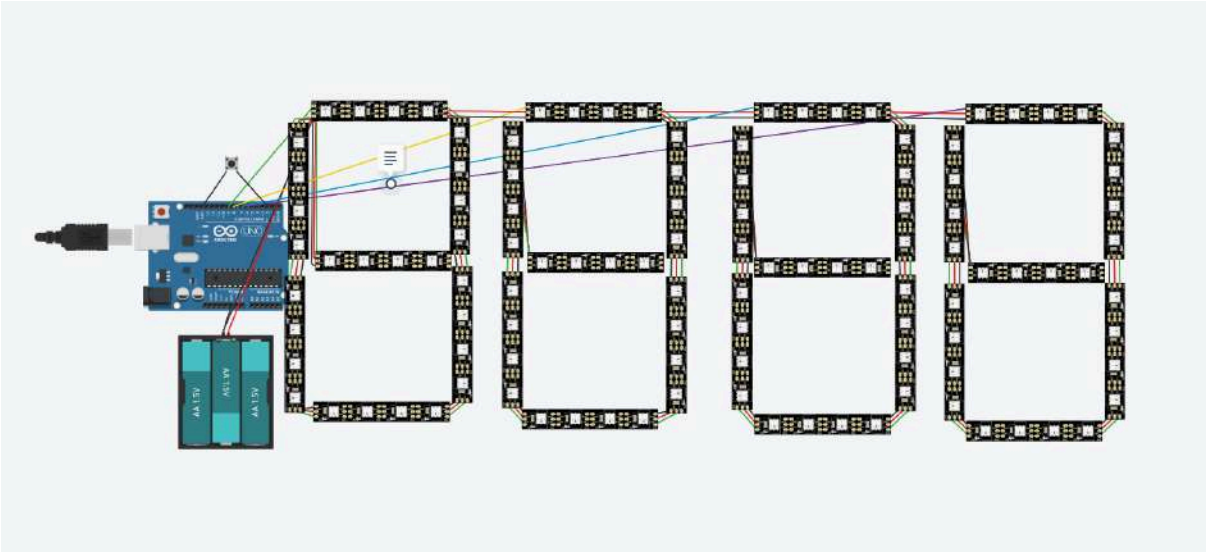
```
//uint32_t red;
int d1iz, d1de, d2iz, d2de, minutos, segundos;
unsigned long segArd, inicio;


const byte numeros[11][7] = {
 {1, 1, 1, 1, 1, 1, 0}, // 0
 {0, 1, 1, 0, 0, 0, 0}, // 1
 {1, 1, 0, 1, 1, 0, 1}, // 2
 {1, 1, 1, 1, 0, 0, 1}, // 3
 {0, 1, 1, 0, 0, 1, 1}, // 4
 {1, 0, 1, 1, 0, 1, 1}, // 5
 {1, 0, 1, 1, 1, 1, 1}, // 6
 {1, 1, 1, 0, 0, 0, 0}, // 7
 {1, 1, 1, 1, 1, 1, 1}, // 8
 {1, 1, 1, 1, 0, 1, 1},  // 9
 {0, 0, 0, 1, 0, 0, 0}  // letra de inicio
};

//Para chequear segmentos
/*const byte numeros[11][7] = {
 {1, 0, 0, 0, 0, 0, 0}, // 0
 {0, 1, 0, 0, 0, 0, 0}, // 1
 {0, 0, 1, 0, 0, 0, 0}, // 2
 {0, 0, 0, 1, 0, 0, 0}, // 3
 {0, 0, 0, 0, 1, 0, 0}, // 4
 {0, 0, 0, 0, 0, 1, 0}, // 5
 {0, 0, 0, 0, 0, 0, 1}, // 6
 {1, 0, 0, 0, 0, 0, 0}, // 7
 {0, 1, 0, 0, 0, 0, 0}, // 8
 {0, 0, 1, 0, 0, 0, 0},  // 9
 {1, 1, 1, 1, 1, 1, 0}  // letra de inicio
};*/




void setup() {
  inicio = millis();

 for (int i = 0; i < NUM_TIRAS; i++) {
   strips[i].begin();
   strips[i].show();  // Apagar LEDs
 }



  Serial.begin(9600);
 }

void loop() {
    segArd = (millis() - inicio) / 1000;

   minutos = (segArd % 3600) / 60;
   segundos = segArd % 60;

   d1iz = segundos / 10;
```

```
   d1de = segundos % 10;

   d2iz = minutos / 10;
   d2de = minutos % 10;

   if(segundos==0 ) {
     mostrarNumero(0, 0);  delay(100);
     mostrarNumero(0, 1);  delay(100);
     mostrarNumero(0, 2);  delay(100);
     mostrarNumero(0, 3); delay(100);
    }



   mostrarNumero(d1de, 0); delay(100);
   mostrarNumero(d1iz, 1); delay(100);
   mostrarNumero(d2de, 2); delay(100);
   mostrarNumero(d2iz, 3); delay(100);

   //Para chequear segmentos
   /*mostrarNumero(d1de, 0); delay(50);
   mostrarNumero(d1de, 1); delay(50);
   mostrarNumero(d1de, 2); delay(50);
   mostrarNumero(d1de, 3); delay(50); */

   Serial.print(d2iz);  Serial.print(d2de);   Serial.print(d1iz);  Serial.println(d1de);

 }


// Mostrar un número en una tira específica
void mostrarNumero(int num, int tiraIndex) {
 int segm, i;
 uint32_t red = strips[tiraIndex].Color(255, 0, 0);
  strips[tiraIndex].clear();
 for (segm = 0; segm < 7; segm++) {
   if (numeros[num][segm] == 1) {
     //strips[tiraIndex].fill(red, Pix0Segm[segm], PixSegm);
     for (i = 0; i <PixSegm; i++) strips[tiraIndex].setPixelColor(Pix0Segm[segm]+i, red);

   }
 }
 strips[tiraIndex].show();
}
```

Resultado final: ver la foto del día de la carrera

Otros detalles y mejoras:

Tinkercad: simulación, es un poco lenta.
Hardware: con un botón para resetear, iniciar y pausar:



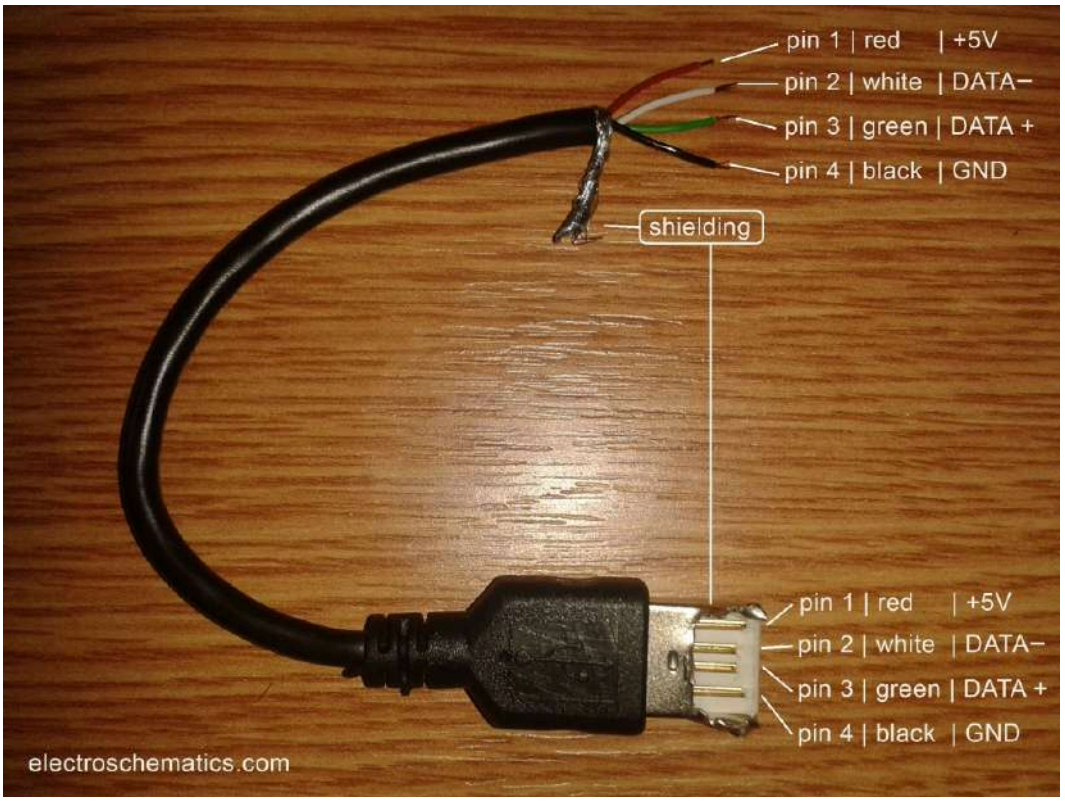
Interrupciones y estados: para reseteo, comienzo y pausa con pulsador conectado al Pin2-GND (típico de interrupciones en Arduino). Abajo software (solo para el profesor y avanzados). Tiene el inconveniente de que los pulsos externos electromagnéticos le hace saltar de estado (a pausa o inicio).

```
#include <Adafruit_NeoPixel.h>

#define NUM_LEDS 75
//#define PixSegm 4
#define PixSegm 11
#define NUM_TIRAS 4

// Declaración de un array de tiras Neopixel
Adafruit_NeoPixel strips[NUM_TIRAS] = {
 Adafruit_NeoPixel(NUM_LEDS, 6, NEO_GRB + NEO_KHZ800),
 Adafruit_NeoPixel(NUM_LEDS, 7, NEO_GRB + NEO_KHZ800),
 Adafruit_NeoPixel(NUM_LEDS, 8, NEO_GRB + NEO_KHZ800),
 Adafruit_NeoPixel(NUM_LEDS, 9, NEO_GRB + NEO_KHZ800)
};

//const int Pix0Segm[7] = {0, 4, 8, 12, 16, 20, 24};
const int Pix0Segm[7] = {0, 11, 23, 35, 45, 55, 65};


uint32_t red, off;
volatile int estadoCronometro;
volatile unsigned long ultimaInterrupcion = 0;  // Para evitar rebotes

const byte numeros[11][7] = {
 {1, 1, 1, 1, 1, 1, 0}, // 0
 {0, 1, 1, 0, 0, 0, 0}, // 1
 {1, 1, 0, 1, 1, 0, 1}, // 2
 {1, 1, 1, 1, 0, 0, 1}, // 3
 {0, 1, 1, 0, 0, 1, 1}, // 4
 {1, 0, 1, 1, 0, 1, 1}, // 5
 {1, 0, 1, 1, 1, 1, 1}, // 6
 {1, 1, 1, 0, 0, 0, 0}, // 7
 {1, 1, 1, 1, 1, 1, 1}, // 8
 {1, 1, 1, 1, 0, 1, 1},  // 9
 {0, 0, 0, 1, 0, 0, 0}  // letra de inicio
};

// Función para detener cronómetro por interrupción
void cambiarEstado() {
```

```
 unsigned long tiempoActual = millis();  // Obtenemos el tiempo actual
 if (tiempoActual - ultimaInterrupcion > 700) {  // Si han pasado más de 200 ms desde la última
interrupción, evitamos rebotes
   estadoCronometro = (estadoCronometro + 1) % 4;  // Cambia cíclicamente entre 0, 1, 2 y 3
   ultimaInterrupcion = tiempoActual;  // Actualizamos el tiempo de la última interrupción
 }
}

void setup() {
 for (int i = 0; i < NUM_TIRAS; i++) {
   strips[i].begin();
   strips[i].show();  // Apagar LEDs
 }

 red = strips[0].Color(255, 0, 0);
 off = strips[0].Color(0, 0, 0);
 estadoCronometro=0;
 pinMode(2, INPUT_PULLUP);  //INPUT_PULLUP activa resistencia interna para evitar flotación.
 attachInterrupt(digitalPinToInterrupt(2), cambiarEstado, FALLING);  //0: for pin2. 1: for pin3
 Serial.begin(9600);
 }

void loop() {
if (estadoCronometro==0) {   //Pausa o inicio
   mostrarNumero(10, 0);  // Mostrar en primera tira
   mostrarNumero(10, 1);  // Mostrar en segunda tira
   mostrarNumero(10, 2);  // Mostrar en segunda tira
   mostrarNumero(10, 3);  // Mostrar en segunda tira
   delay(500);
   estadoCronometro=1;
   delay(500);
 }else if (estadoCronometro==1){
   delay(200); //error aceptado al pulsar
 }else if (estadoCronometro==2){
   cronometraWhile();
 }else{
   //pausa

 }

 delay(500);
 Serial.println(estadoCronometro);
 }

void cronometraWhile() {
 int diz, dde, minutos, segundos;
 unsigned long inicio, segArd;
  inicio = millis();
 while(estadoCronometro==2){
   segArd = (millis() - inicio) / 1000;
   //horas = segArd / 3600;
   minutos = (segArd % 3600) / 60;
   segundos = segArd % 60;

   diz = segundos / 10;
   dde = segundos % 10;


   mostrarNumero(dde, 0);  // Mostrar en primera tira
   mostrarNumero(dde, 1);  // Mostrar en segunda tira
   mostrarNumero(dde, 2);  // Mostrar en segunda tira
   mostrarNumero(dde, 3);  // Mostrar en segunda tira
   delay(500);
 }
}

// Mostrar un número en una tira específica
void mostrarNumero(int num, int tiraIndex) {
```

```
  strips[tiraIndex].clear();
  for (int segm = 0; segm < 7; segm++) {
    if (numeros[num][segm] == 1) {
      strips[tiraIndex].fill(red, Pix0Segm[segm], PixSegm);
    }
  }
  strips[tiraIndex].show();
}
```

USB pin out: Para alimentar Arduino añadiendo a la fuente un USB hembra (aunque da un poco de inestabilidad)



Media aritmética: para calcular el nº de leds por panel, suma total dividido entre nº de sumandos (74+76+73+75)/4=74.5
Ru: 74
Ne: 76
Ra: 73
Ad: 75
—----------
Total: 298

Potencia: P=VI, si los leds son 90W y el voltaje típico del USB es 5V cuantos amperios van a circular: I=90/5=18A, luego necesitamos una fuente potente.

Soldaduras de precisión: soldar primero una gotita al cable, enfriar y luego soldar el cable al conector (no mover y esperar a que enfríe).

Alero y protecciones: añadir un saliente encima de los números de leds para dar sombra y que se vea mejor. Añadir protecciones de cartulina negra o azul sobre los cables y leds para que luzca mejor.