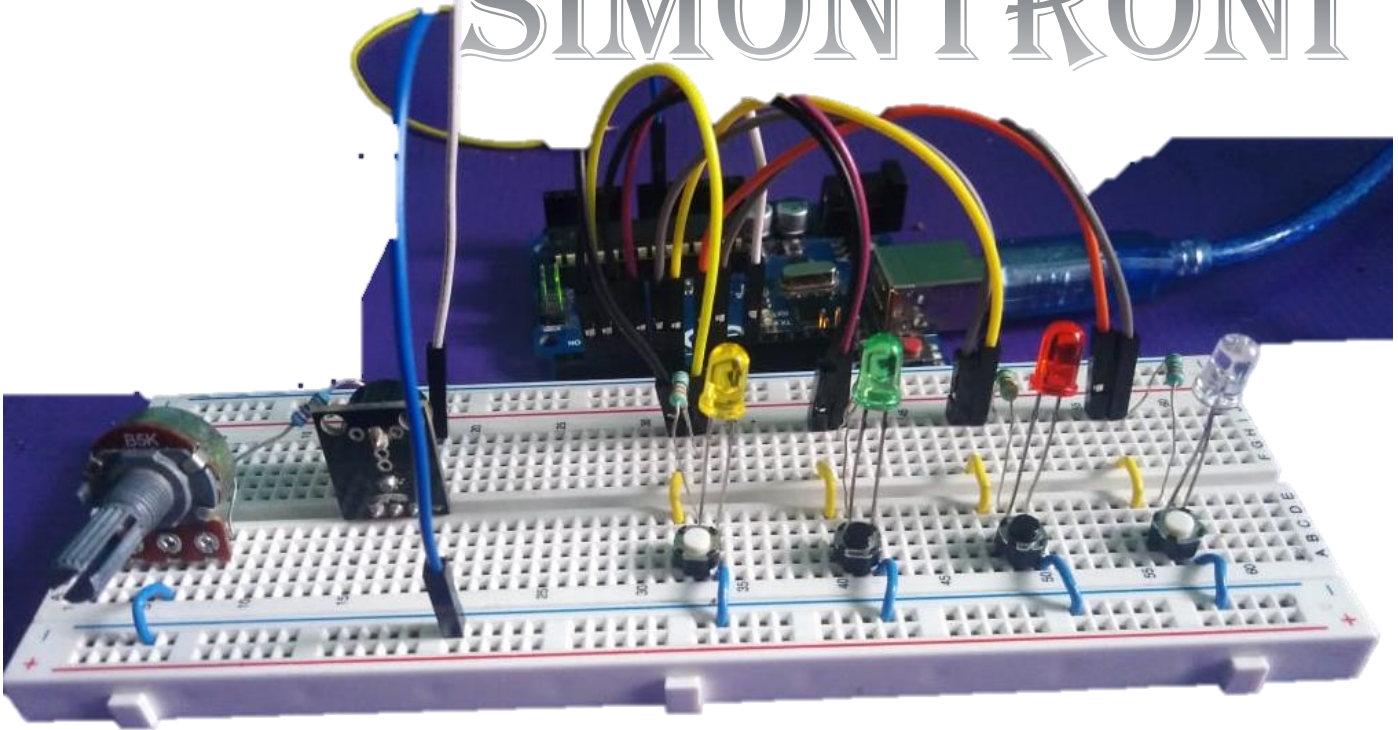


SIMONTRONI



Autor: Hernández Beltrán David Roni

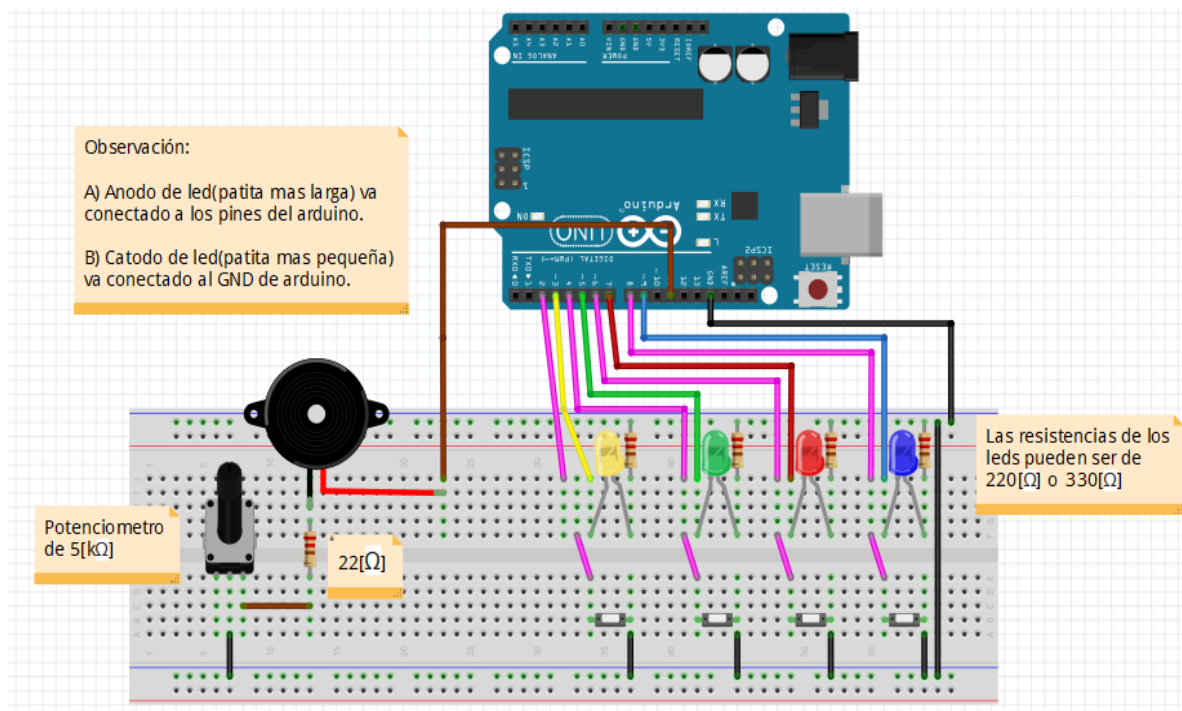
ELECTRONICA DEL PROYECTO:

El funcionamiento del juego es muy sencillo, se prende una secuencia de leds y nosotros la debemos de repetir pulsando los botones respectivos, a medida que avanzamos, se prendera 1 led más, y así sucesivamente hasta terminar el juego o perder el juego.

Las resistencias de los leds son para limitar la corriente que pasa a través de ellos y así evitar dañar la placa Arduino o los leds por un flujo grande de corriente.

El potenciómetro lo utilice para limitar la corriente que le llega al zumbador, de tal manera que si lo movemos alteramos la resistencia de este, lo cual ocasiona una alteración en la corriente, lo que significa una alteración en la potencia que recibe el zumbador, lo cual significa una variación en el volumen del juego.

La resistencia que une al potenciómetro con el zumbador sirve como medida de protección en caso de que el jugador decida subirle todo el volumen al juego (tener un potenciómetro con un resistencia de $0\ [\Omega]$) y así evitar que la corriente que fluya por el zumbador nunca sea muy grande pues no importando como gire el potenciómetro el jugador, la resistencia mínima siempre serán $22[\Omega]$.



Todos los botones están conectados de un pin de Arduino al GND de Arduino, debido a que active la resistencia interna de los pines que van conectados a los botones, declarando esos pines como pines: INPUT_PULLUP

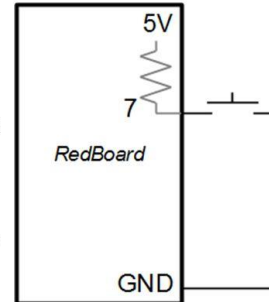
Arduino Digital Input



Pins Configured as INPUT_PULLUP

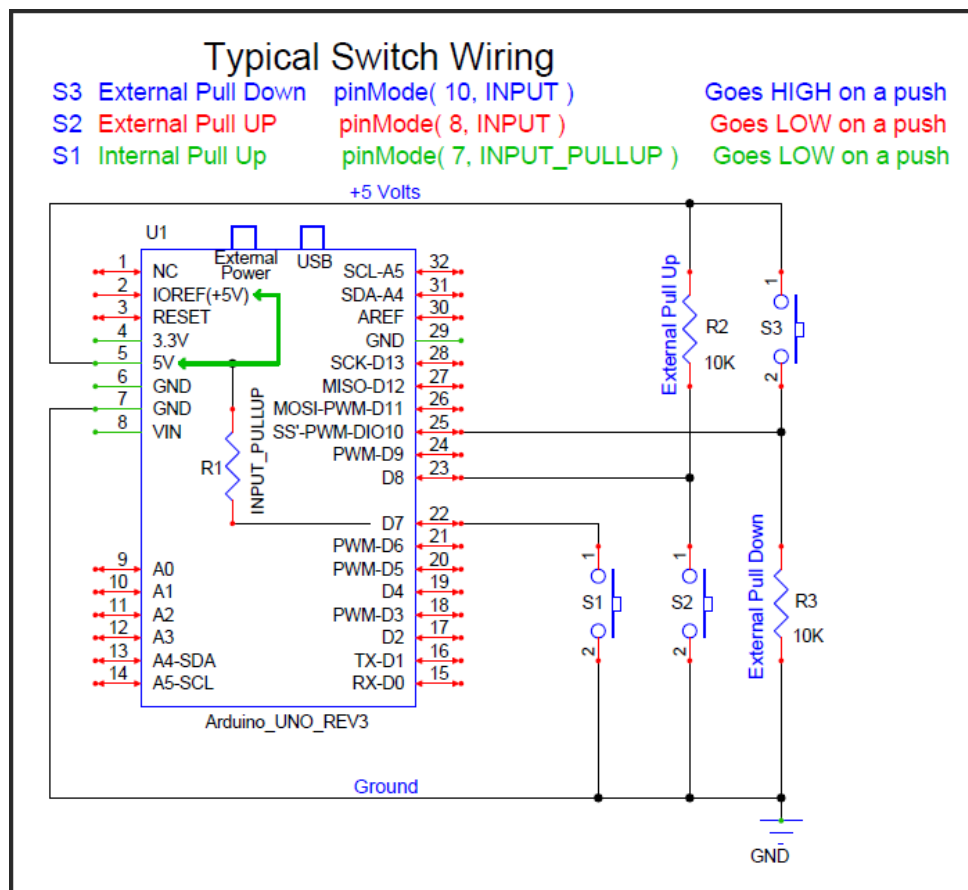
The Atmega chip on the Arduino has internal pull-up resistors (resistors that connect to power internally) that you can access. If you prefer to use these instead of external pull-down resistors, you can use the **INPUT_PULLUP** argument in `pinMode()`.

<http://arduino.cc/en/Reference/Constants>



UNION
COLLEGE
FOUNDED 1862

De esa manera me ahorro el trabajo extra de colocar resistencias externas para los botones:

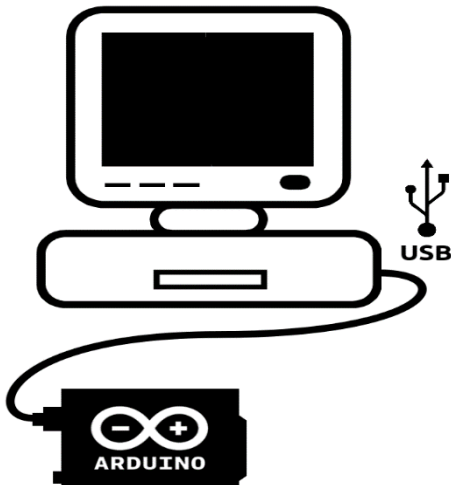


Las imágenes fueron obtenidas de las siguientes fuentes de información:

<https://slideplayer.com/slide/4469076/>

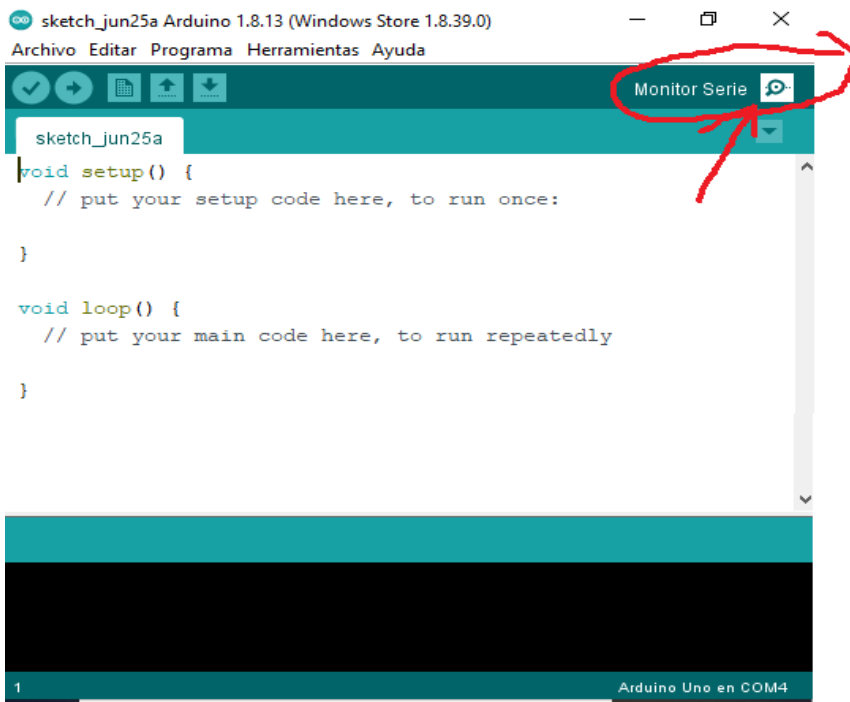
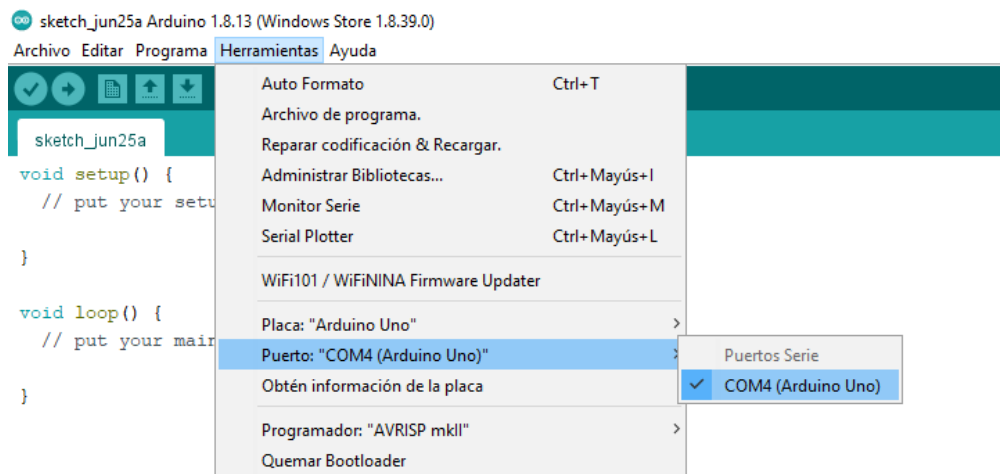
<https://forum.arduino.cc/index.php?topic=575915.0>

INSTRUCCIONES DEL JUEGO:

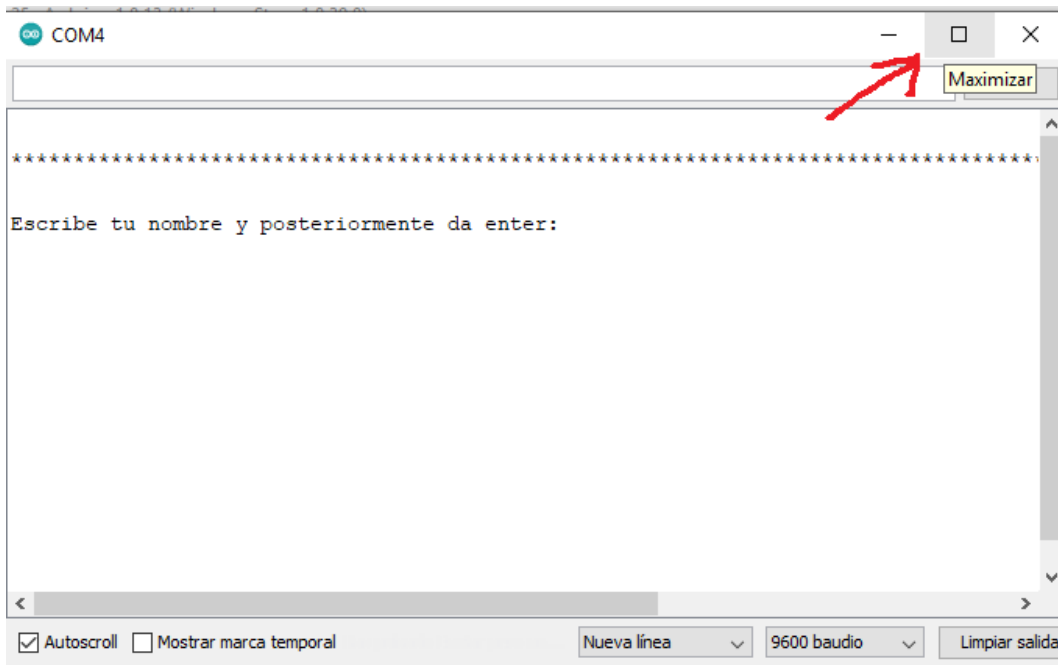


PASO 1: Para poder jugar con **simonTroni** debemos conectar el Arduino a la computadora, pues el juego se comunica vía serial.

PASO 2: Debemos abrir el IDE de Arduino, y cerciorar que esta seleccionado el puerto del Arduino uno que conectamos:

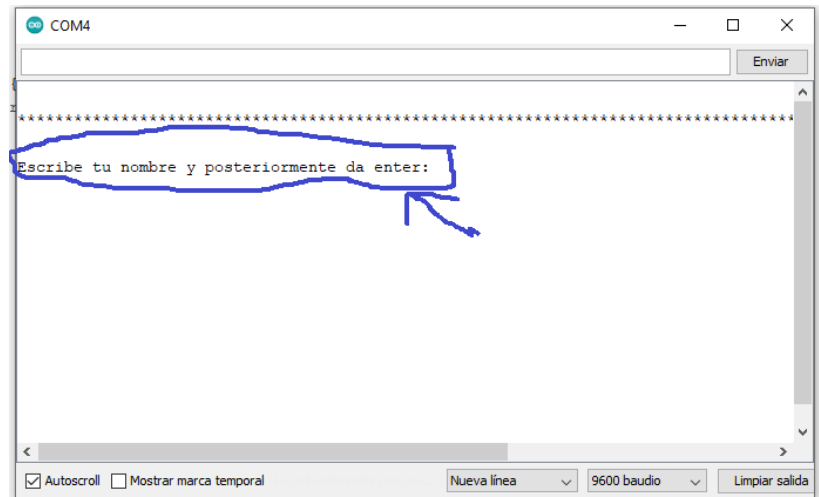


PASO 3: Necesitamos abrir el monitor serie del IDE de Arduino, así que damos clic izquierdo sobre la lupa:

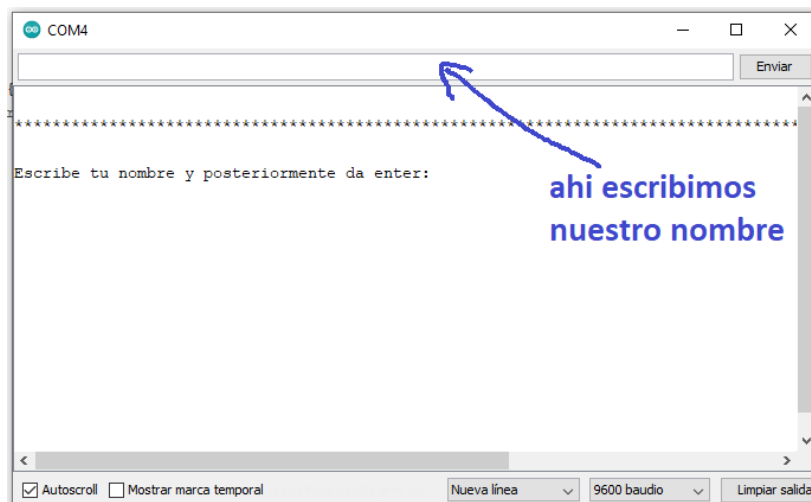


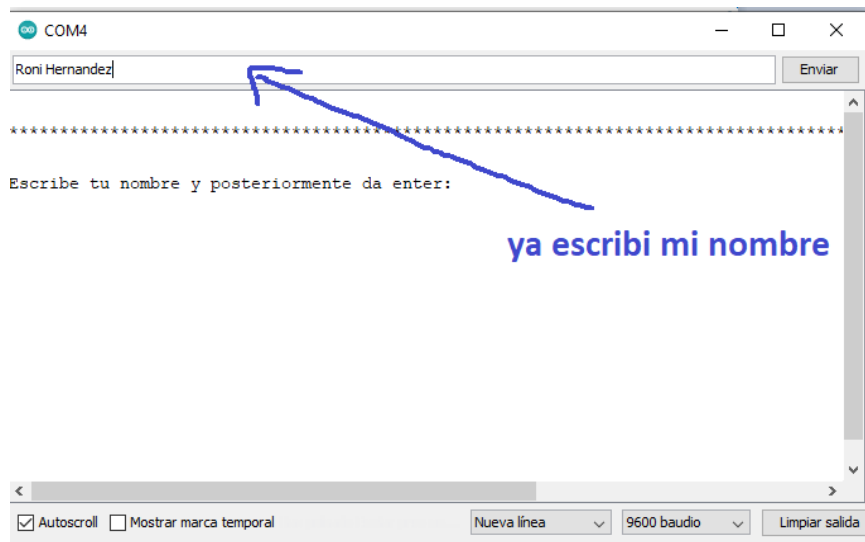
PASO 4: Antes de hacer cualquier cosa, lo primero que debemos hacer es maximizar la pantalla, así que proseguiremos a darle clic izquierdo sobre el icono de maximizar:

PASO 5: Podemos observar que nos pide que escribamos nuestro nombre y posteriormente tecleamos ENTER, pues eso haremos...

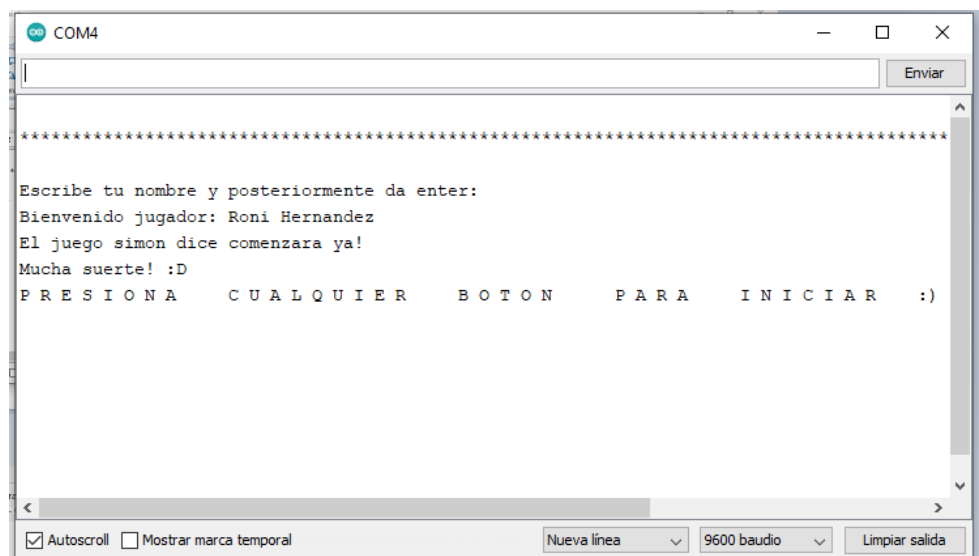


PASO 6: Seguir las siguientes instrucciones.

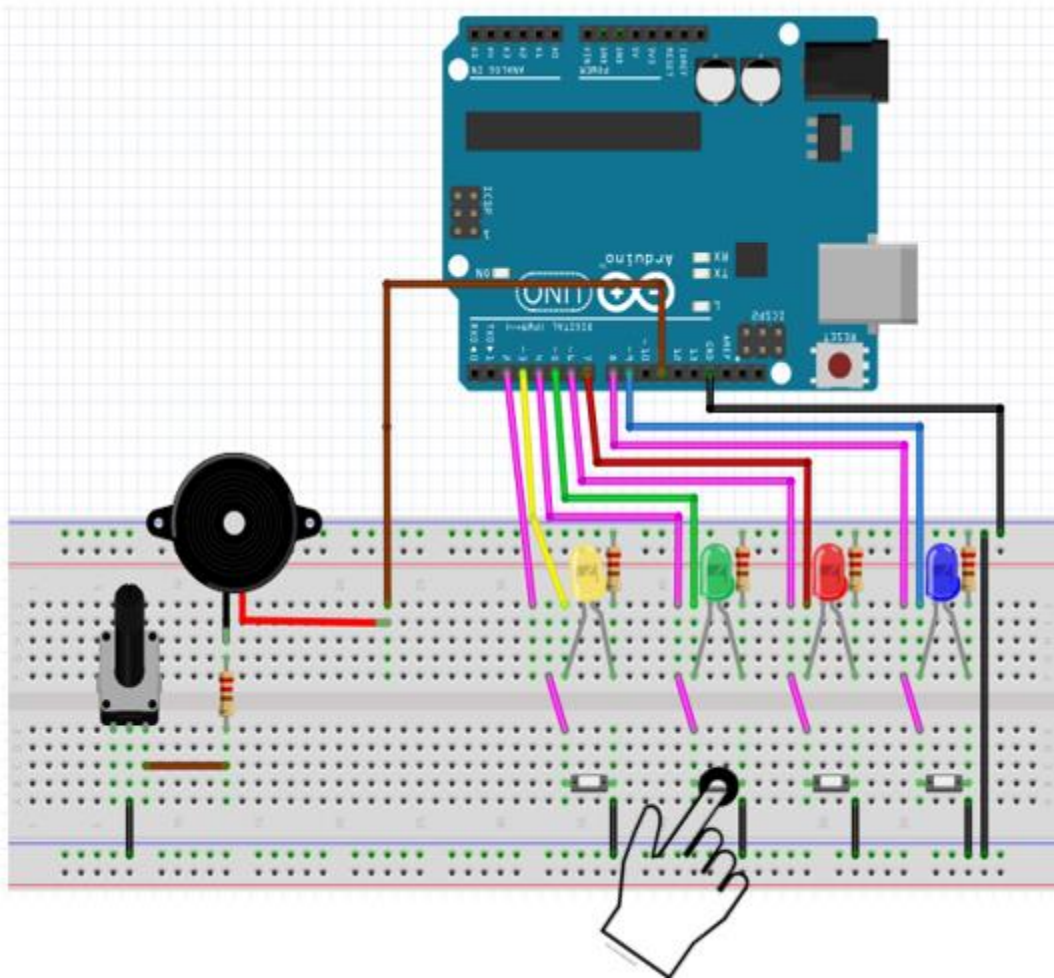




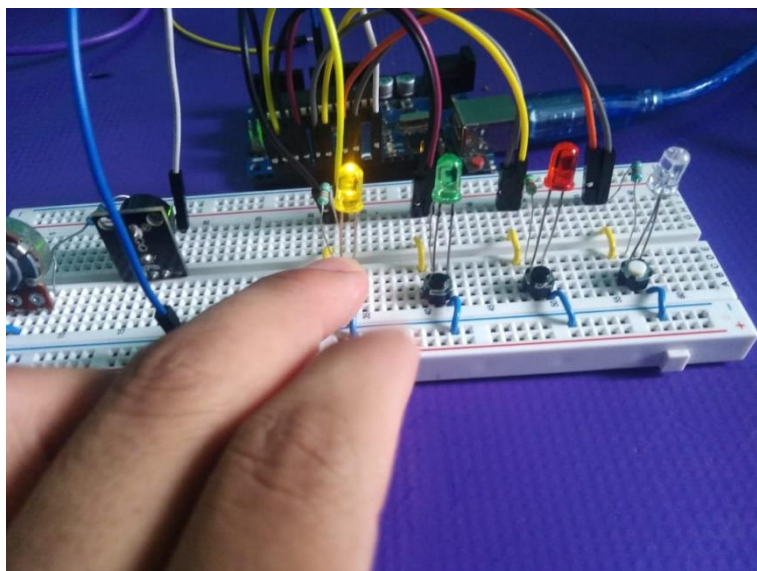
PASO 7: Observar que una vez realizado lo anterior, nos aparecerá el siguiente mensaje, diciéndonos que presionemos cualquier botón del juego para poder iniciar el juego

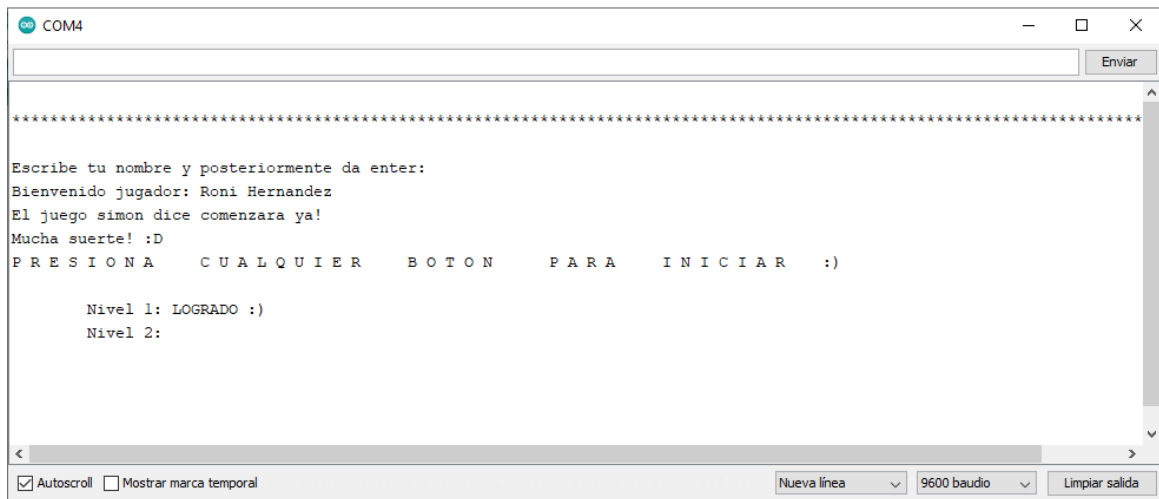


PASO 8: Presionamos cualquier botón del juego para poder iniciar el juego:



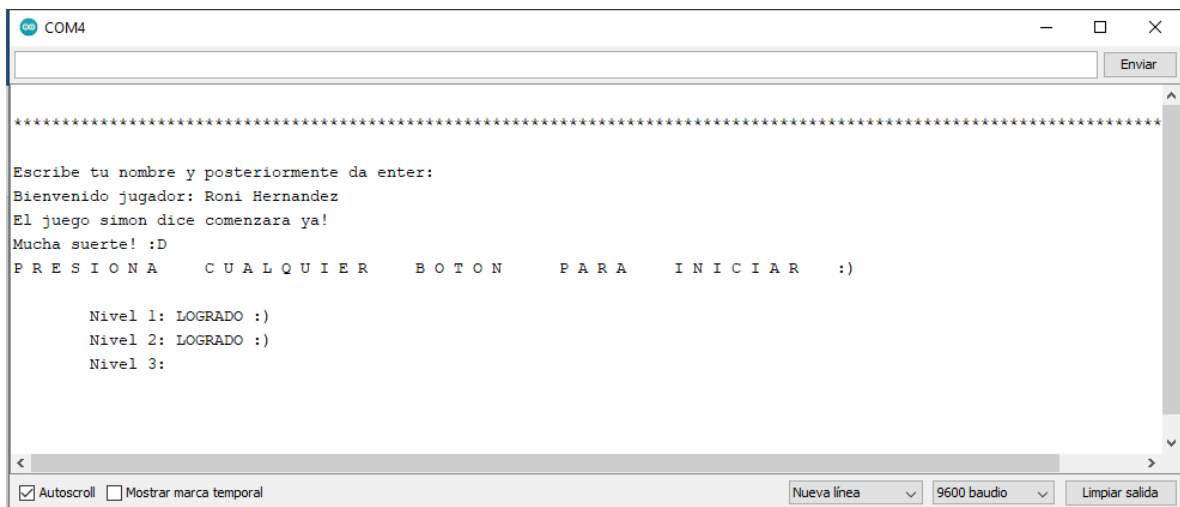
PASO 9: Finalmente el juego comenzara preniendo un led al mismo tiempo que la bocina da un pitido. Nosotros deberemos presionar el botón del led que se prendió.





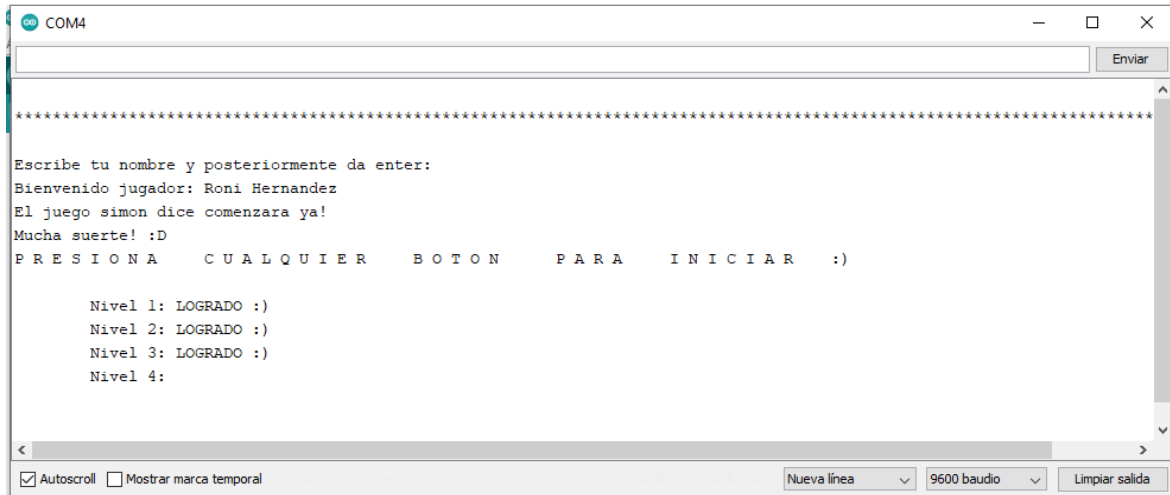
```
*****  
Escribe tu nombre y posteriormente da enter:  
Bienvenido jugador: Roni Hernandez  
El juego simon dice comenzara ya!  
Mucha suerte! :D  
P R E S I O N A      C U A L Q U I E R      B O T O N      P A R A      I N I C I A R      :)  
  
    Nivel 1: LOGRADO :)  
    Nivel 2:  
  
<----->  
☒ Autoscroll ☐ Mostrar marca temporal  Nueva línea  9600 baudio  Limpiar salida
```

PASO 10: Si pulsamos el botón correcto del led que se prendió, pasaremos al siguiente nivel del juego, el cual consistirá en presionar los botones en el orden respectivo de los 2 leds que se prenderán de forma cronológica.



```
*****  
Escribe tu nombre y posteriormente da enter:  
Bienvenido jugador: Roni Hernandez  
El juego simon dice comenzara ya!  
Mucha suerte! :D  
P R E S I O N A      C U A L Q U I E R      B O T O N      P A R A      I N I C I A R      :)  
  
    Nivel 1: LOGRADO :)  
    Nivel 2: LOGRADO :)  
    Nivel 3:  
  
<----->  
☒ Autoscroll ☐ Mostrar marca temporal  Nueva línea  9600 baudio  Limpiar salida
```

PASO 11: Si pulsamos los botones correctos en el orden correcto, pasaremos al siguiente nivel del juego, el cual consistirá en presionar los botones en el orden respectivo, pero ahora de los 3 leds que se prenderán de forma cronológica.

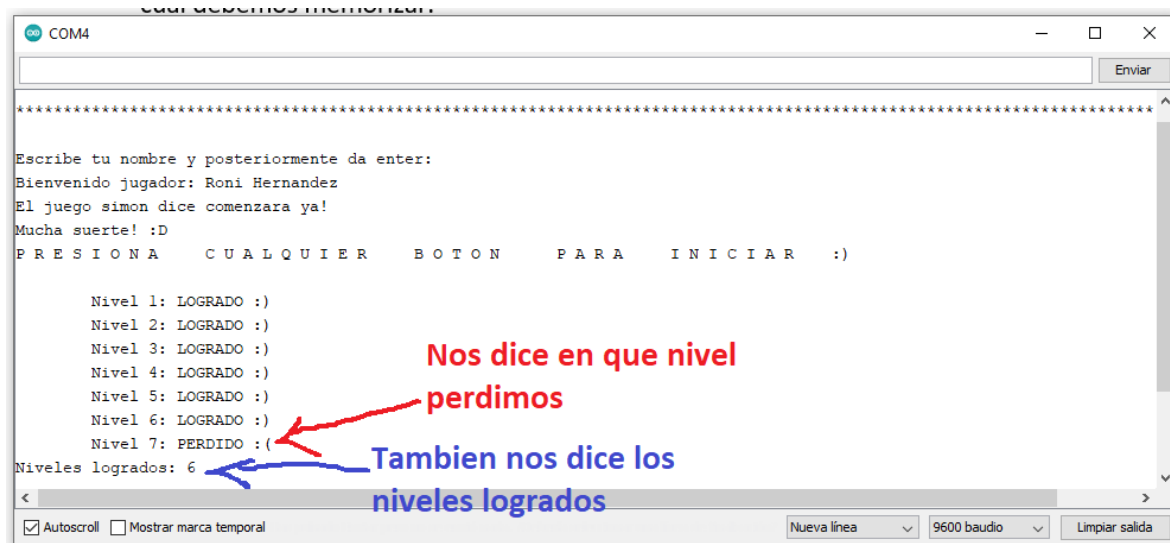


```
*****
Escribe tu nombre y posteriormente da enter:
Bienvenido jugador: Roni Hernandez
El juego simon dice comenzara ya!
Mucha suerte! :D
P R E S I O N A   C U A L Q U I E R   B O T O N   P A R A   I N I C I A R   :)

Nivel 1: LOGRADO :)
Nivel 2: LOGRADO :)
Nivel 3: LOGRADO :)
Nivel 4:
```

PASO 12: Y así seguirá el juego a medida que pasemos al siguiente nivel aumentará un led más el cual debemos memorizar.

PASO 13: Si perdemos el juego nos avisará que perdimos y nos dirá nuestra puntuación lograda, y si queremos volver a jugar debemos seguir los mismos pasos a partir del paso número 5.



```
*****
Escribe tu nombre y posteriormente da enter:
Bienvenido jugador: Roni Hernandez
El juego simon dice comenzara ya!
Mucha suerte! :D
P R E S I O N A   C U A L Q U I E R   B O T O N   P A R A   I N I C I A R   :)

Nivel 1: LOGRADO :)
Nivel 2: LOGRADO :)
Nivel 3: LOGRADO :)
Nivel 4: LOGRADO :)
Nivel 5: LOGRADO :)
Nivel 6: LOGRADO :)
Nivel 7: PERDIDO :(
Niveles logrados: 6
```

Nos dice en que nivel perdimos

Tambien nos dice los niveles logrados

PASO 14: Si pasamos todos los niveles del juego, el juego nos avisara, y nos tocara una pieza musical como recompensa de que hemos ganado, si queremos volver a jugar debemos seguir los mismos pasos a partir del paso número 5.

```
*****
Escribe tu nombre y posteriormente da enter:
Bienvenido jugador: Roni Hernandez
El juego simon dice comenzara ya!
Mucha suerte! :D
P R E S I O N A   C U A L Q U I E R   B O T O N   P A R A   I N I C I A R   :)

Nivel 1: LOGRADO :)
Nivel 2: LOGRADO :)
Nivel 3: LOGRADO :)
Nivel 4: LOGRADO :)
Nivel 5: LOGRADO :)
Nivel 6: LOGRADO :)
Nivel 7: LOGRADO :)
Nivel 8: LOGRADO :)
Nivel 9: LOGRADO :)
Nivel 10: LOGRADO :)
Nivel 11: LOGRADO :)
Nivel 12: LOGRADO :)
Nivel 13: LOGRADO :)
Nivel 14: LOGRADO :)
Nivel 15: LOGRADO :)
Nivel 16: LOGRADO :)
FELICIDADES HAS TERMINADO EL JUEGO,
POR FAVOR DISFRUTA ESTA BONITA CANCION
<
☒ Autoscroll ☐ Mostrar marca temporal
Nueva línea 9600 baudio Limpiar salida
```

Nos dice que hemos terminado finalmente todo el juego

PASO 15: Si deseamos bajar el volumen del juego basta con girar el potenciómetro para ajustar el volumen del juego

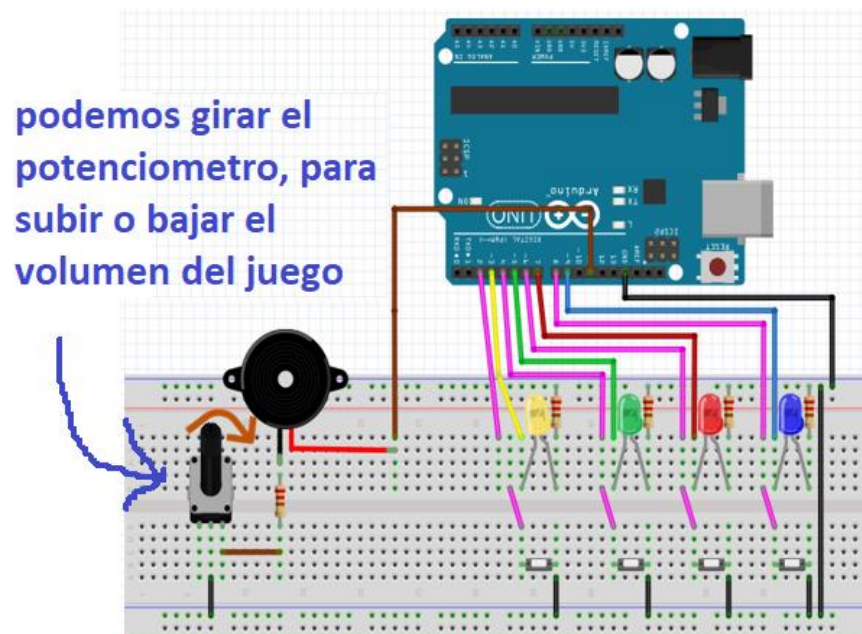


DIAGRAMA DEL PROYECTO:

Observación:

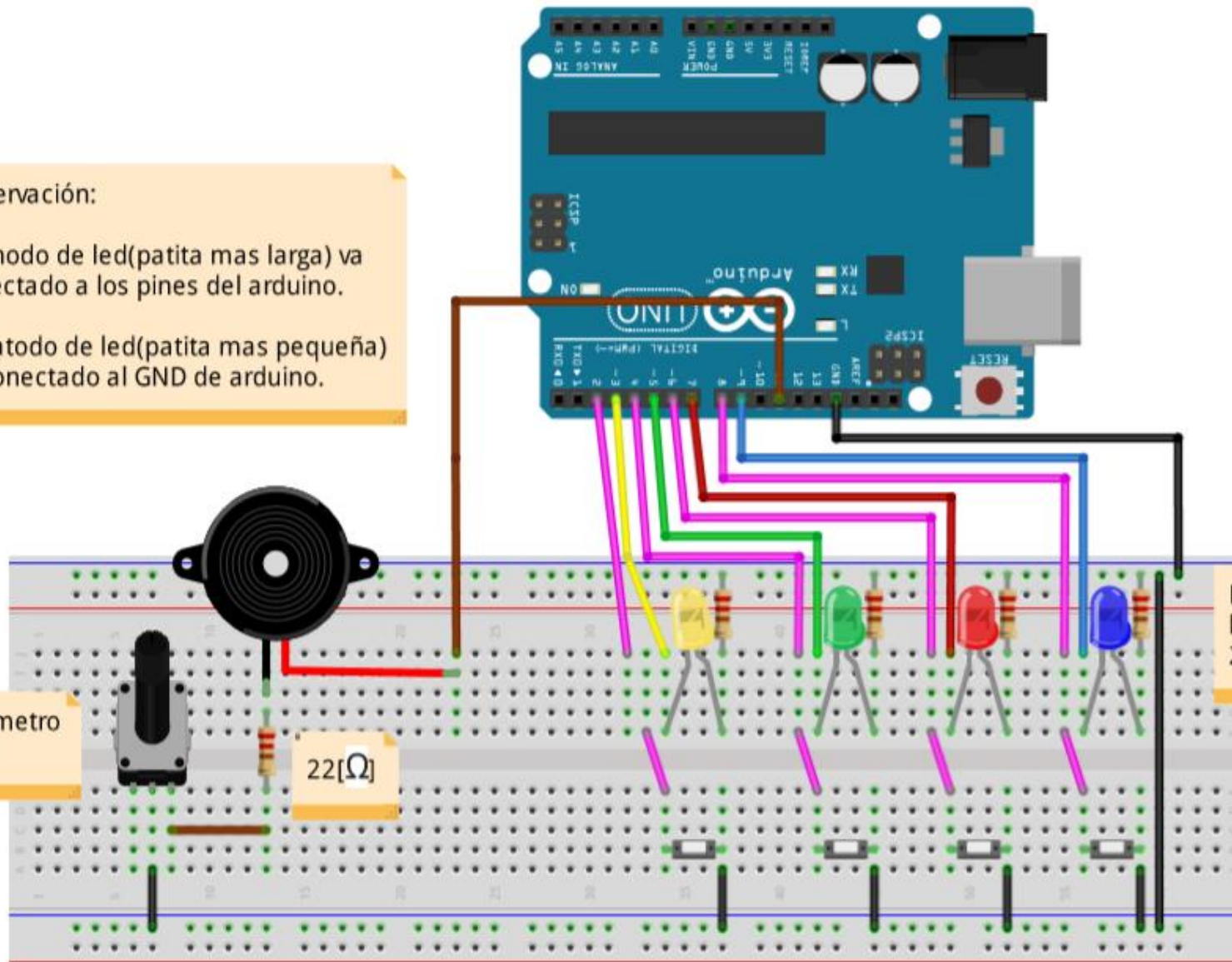
A) Anodo de led(patita mas larga) va conectado a los pines del arduino.

B) Catodo de led(patita mas pequeña) va conectado al GND de arduino.

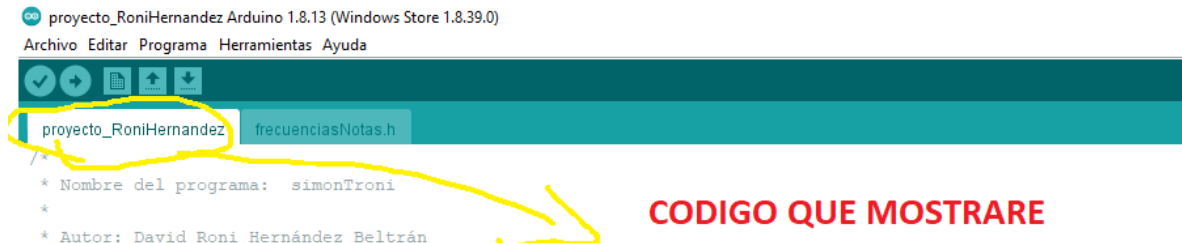
Potenciometro
de 5[k Ω]

22[Ω]

Las resistencias de los
leds pueden ser de
220[Ω] o 330[Ω]



CODIGO DE ARDUINO:



```
/*
 * Nombre del programa:  simonTroni
 *
 * Autor: David Roni Hernández Beltrán
 *
 * Descripción: Este proyecto se basa en el clasico
 * juego de memoria en el cual se te muestra una
 * secuencia y tu tiene que tocarla, a medida que
 * avanzas en el juego tienes que memorizar una nueva
 * posicion.
 */

/*
 * Nombre del programa: simonTroni
 *
 * Autor: David Roni Hernández Beltrán
 *
 * Descripción: Este proyecto se basa en el clasico
 * juego de memoria en el cual se te muestra una
 * secuencia y tu tiene que tocarla, a medida que
 * avanzas en el juego tienes que memorizar una nueva
 * posicion.
 */

//Lubreria que contiene definidas varias frecuencias de notas musicales
//de distinta octavas:
#include "frecuenciasNotas.h"
//definiendo los pines de cada boton...
#define BOTON_1 2
#define BOTON_2 4
#define BOTON_3 6
#define BOTON_4 8
//definiendo los pines de cada led...
#define LED_1 3
#define LED_2 5
#define LED_3 7
#define LED_4 9
#define NO_BOTONES 4
//array que almacenara el numero de cada pin que sera boton...
byte G_arrayBotones[NO_BOTONES]={BOTON_1,BOTON_2,BOTON_3,BOTON_4};
//array que almacenara el numero de cada pin que sera led...
byte G_arrayLeds[NO_BOTONES]={LED_1,LED_2,LED_3,LED_4};
```

```

/////////////////////////////////////////////////////////////////
//  DATOS DE LA MELODIA: YESTERDAY
/////////////////////////////////////////////////////////////////
//numero de notas de la melodía yesterday...
#define TAM_YESTERDAY 31
// Notas de la melodía de yesterday:
int G_notas_yesterday[TAM_YESTERDAY] = {
    G4,F4,F4,PAUSA, A4,B4,CS5,D5,E5,F5,E5,D5,D5,PAUSA, D5,D5,C5,AS4,A4,G4,AS4,A4,A4,
    G4,F4,A4,G4,D4,F4,A4,A4 };
// Duración de las notas:
// Estandar que utilice en los tiempos de las figuras musicales: 0.5=corchea 1=negra 2=blanca
float G_duracionNotas_yesterday[TAM_YESTERDAY] = {
    0.5,0.5,3,1, 0.5,0.5,0.5,0.5,0.5,0.5,0.5,3,1, 0.5,0.5,0.5,0.5,0.5,1,0.5,1, 1,1,0.5,1,1,1,0.5,2.5 };
//pin al cual se conectara el buzzer para emitir sonidos...
const int PIN_BOCINA=11;
/////////////////////////////////////////////////////////////////
// F U N C I O N E S   Q U E   C R E E :
/////////////////////////////////////////////////////////////////
//declaracion de las funciones que cree,para modular de mejor
//manera el programa...
void mensajeGanador();
void mensajePerdedor();
void mensajeBienvenida();
char botonPresionan(byte* arrayBotones,byte noBotones,bool sonInputPullUp);
void tocarTono(int frecuencia,int duracion );
void maquinaEscribirTeclea(String elTexto,long tiempoEntreLetra);

void setup() {
    //configurando pines de botones como pines INPUT_PULLUP
    int c;
    for(c=0;c<NO_BOTONES;c++){
        pinMode(G_arrayBotones[c],INPUT_PULLUP);
    }
    //configurando pines de leds como pines OUPUT(SALIDA)
    for(c=0;c<NO_BOTONES;c++){
        pinMode(G_arrayLeds[c],OUTPUT);
    }
    //apagando todos los leds antes de empezar...
    for(c=0;c<NO_BOTONES;c++){
        digitalWrite(G_arrayLeds[c],LOW);
    }
    //velocidad de comunicacion serial...
    Serial.begin(9600);
}

```

```

void loop() {
  mensajeBienvenida();
  //mientras no presionen un boton no los dejare iniciar el juego
  //y por lo tanto no podran salir de este ciclo while anidado
  while( botonPresionan(G_arrayBotones,4,true) == 0 ){
  }
  //void simonDice_byRoni(String secuenciaJuego,byte* arrayLeds, byte* arrayBotones,byte noBotones )
  int noNiveles=simonDice_byRoni("1314141243213432",G_arrayLeds,G_arrayBotones,NO_BOTONES);
  if(noNiveles<=0){//significa que perdio el juego...
    mensajePerdedor();
  }else{//significa que gano el juego....
    mensajeGanador();
  }
}
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// P R O G R A M A C I O N   D E L   J U E G O :
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/* int simonDice_byRoni (String secuenciaJuego,byte* arrayLeds, byte* arrayBotones,byte noBotones )
*
* Crea un juego de simon clasico el cual seguira la secuencia del parametro de tipo
* string cuyo nombre es 'secuenciaJuego', tambien recibe un array de los pines de botones
* y de leds debido a que la funcion necesita saber que botones controlaran a los leds
* y que leds son los que prendera para el juego
*
* La funcion retorna el puntaje obtenido por el jugador como numero negativo en caso
* de que el jugador no haya conseguido ganar...
*
* La funcion retorna el puntaje obtenido por el jugador como numero positivo en caso
* de que el jugador haya terminado todo el juego completo...
*
*
*/
int simonDice_byRoni(String secuenciaJuego,byte* arrayLeds, byte* arrayBotones,byte noBotones ){
  int noNiveles=secuenciaJuego.length();
  int nivelesLogrados=-1;
  for(int nivel=0;nivel<noNiveles;nivel++){
    if(nivel>0){
      Serial.println("LOGRADO :) ");
    }
    //MOSTRANDO SECUENCIA DE LEDS....
    Serial.print("\tNivel ");Serial.print(nivel+1);Serial.print(": ");
    for(int ledPrender=0;ledPrender<nivel+1;ledPrender++){
      //prendemos foco
      //codigo ascii 49 = 1 ( Número uno )
      digitalWrite(arrayLeds[ byte(secuenciaJuego[ledPrender])-49 ] ,HIGH);
      tocarTono(500,400);
      digitalWrite(arrayLeds[ byte(secuenciaJuego[ledPrender])-49 ] ,LOW);
    }
    //PREGUNTANDO SECUENCIA DE LEDS...
    byte botonEsPresionado=0;
    for(int ledPrender=0;ledPrender<nivel+1;ledPrender++){
      //HASTA QUE SE PRECIONE UN BOTON...

```



```

    botonEsPresionado=0;
    while(botonEsPresionado==0){
        botonEsPresionado=botonPresionan(arrayBotones,4,true);
    }
    digitalWrite(arrayLeds[botonEsPresionado-1],HIGH);
    //¿el boton pulsado fue el incorrecto?....
    if(botonEsPresionado != ( byte(secuenciaJuego[ledPrender])-48 ) ){
        nivelesLogrados=nivel;
        nivel=noNiveles;
        digitalWrite(arrayLeds[botonEsPresionado-1],LOW);
        Serial.println("PERDIDO :(");
        break;
    }
    tocarTono(500,400);
    digitalWrite(arrayLeds[botonEsPresionado-1],LOW);
}
delay(1000);
}
if(nivelesLogrados==1){//el usuario logro terminar el juego satisfactoriamente
    Serial.print("LOGRADO :");
    return noNiveles;
}else{//el usuario no consiguio terminar el juego satisfactoriamente
    Serial.print("Niveles logrados: ");
    Serial.println(nivelesLogrados);
    return nivelesLogrados*-1;//regresando el numero de niveles logrados con numeros negativos
    //para que sepan que no paso el juego
}
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//  O T R A S  F U N C I O N E S  :
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/*
 * void tocarTono(int frecuencia,int duracion )
 *
 * Esa funcion sirve para tocar un frecuencia durante un tiempo por el pin que esta conectado a un buzer...
 *
 */
void tocarTono(int frecuencia,int duracion ){
    tone(PIN_BOCINA, frecuencia, duracion);
    delay(duracion);
    noTone(PIN_BOCINA);
    delay(100);
}
/*
 * Retornara el numero de boton que haya sido presionado, en caso de no haber detectado
 * ningun boton presionado retornara cero
 *
 */
char botonPresionan(byte* arrayBotones,byte noBotones,bool sonInputPullUp){
    bool unBotonPresionado=!sonInputPullUp;
    char botonEsPresionado=-1; //valor default el cual permanecera
        //en caso de no detectarse ningun boton
    for(int c=0;c<4;c++){
        if( digitalRead(arrayBotones[c])== unBotonPresionado ){
            botonEsPresionado=c;
            break;
        }
    }
    botonEsPresionado++;
    return botonEsPresionado;
}

/*
 * void maquinaEscribirTeclea(String elTexto,long tiempoEntreLetra)
 * Esta funcion escribira como una maquina de escribir a traves del puerto serial
 * el texto que se le pone, y entre cada caracter de este, esperara 'timeEntreLetra'
 * milisegundos...
 *
 */
void maquinaEscribirTeclea(String elTexto,long tiempoEntreLetra){
    for(int c=0;c<elTexto.length();c++){
        Serial.print( elTexto[c] );
        delay(tiempoEntreLetra);
    }
}

```

```
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
// M E N S A J E S           D E L           J U E G O  
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
/*  
 * void mensajeBienvenida()  
 * Preguntara el nombre del jugador, y despues dara las instrucciones basicas del juego...  
 *  
 */  
void mensajeBienvenida(){  
    Serial.setTimeout(100000); //100 segundos de espera  
  
Serial.print("\n*****  
*****+\\n");  
    Serial.println("\\nEscribe tu nombre y posteriormente da enter: ");  
    String nombre=Serial.readStringUntil('\\n');  
    maquinaEscribirTeclea("Bienvenido jugador: "+nombre+"\\nEl juego simon dice comenzara ya! \\nMucha  
suerte! :D \\nPRESIONA CUALQUIER BOTON PARA INICIAR :)\\n\\n",50);  
}  
  
/*  
 * void mensajePerdedor()  
 * Encendera los leds al mismo tiempo y luego los apagara, y entre cada ciclo que hace eso, emitira un  
pitido  
 * y de esa manera le hago saber al jugador que no ha podido pasar mi juego...  
 */  
  
void mensajePerdedor(){  
    int c,repetir;  
    for(repetir=0;repetir<8;repetir++){  
        //configurando pines como pines OUPUT(SALIDA)  
        for(c=0;c<NO_BOTONES;c++){  
            digitalWrite(G_arrayLeds[c],HIGH);  
        }  
        tocarTono(250,150);  
        //configurando pines como pines OUPUT(SALIDA)  
        for(c=0;c<NO_BOTONES;c++){  
            digitalWrite(G_arrayLeds[c],LOW);  
        }  
        delay(250);  
    }  
}
```

```

/*
 * void mensajeGanador()
 * Encendera los leds en forma de escalerita acendiendo y bajando varias veces y cada vez emitiendo un
pitido
 * despues escribira el mensaje de victoria y finalmente tocara una melodía con leds prendiendo
aleatoriamente
 * simulando que tocan cada nota, y de esa manera festejo al ganador de mi juego...
 *
 */
void mensajeGanador(){
//semilla atraves del ruido del pin analogico 0...
randomSeed( analogRead(A0) );
byte ledPrender=0;
int c, repetir;
for(repetir=0; repetir<10; repetir++){
//configurando pines como pines OUPUT(SALIDA)
for(c=0; c<NO_BOTONES; c++){
digitalWrite(G_arrayLeds[c], HIGH);
delay(50);
digitalWrite(G_arrayLeds[c], LOW);
}
tocarTono(500, 150);
//configurando pines como pines OUPUT(SALIDA)
for(c=NO_BOTONES-2; c>=0; c--){
digitalWrite(G_arrayLeds[c], HIGH);
delay(50);
digitalWrite(G_arrayLeds[c], LOW);
}
tocarTono(500, 150);
}
maquinaEscribirTeclea("\nFELICIDADES HAS TERMINADO EL JUEGO,\nPOR FAVOR DISFRUTA ESTA BONITA
CANCION\n", 50);
for(repetir=0; repetir<2; repetir++){
for (int noNotaTocar = 0; noNotaTocar < TAM_YESTERDAY; noNotaTocar++) {
int duracionDeEsaNota = 600*G_duracionNotas_yesterday[noNotaTocar];
ledPrender=random(NO_BOTONES);
digitalWrite(G_arrayLeds[ledPrender], HIGH);
tone(PIN_BOCINA, G_notas_yesterday[noNotaTocar], duracionDeEsaNota);
delay(duracionDeEsaNota);
noTone(PIN_BOCINA);
digitalWrite(G_arrayLeds[ledPrender], LOW );
}
}
}
}

```



```
/*
 * alturas.h: constantes con las frecuencias
 * correspondientes a las alturas de notas musicales
 * (parte entera)
 */
#define PAUSA 0

/*
 * alturas.h: constantes con las frecuencias
 * correspondientes a las alturas de notas musicales
 * (parte entera)
 */
#define PAUSA 0
#define B0 31
#define C1 33
#define CS1 35
#define D1 37
#define DS1 39
#define E1 41
#define F1 44
#define FS1 46
#define G1 49
#define GS1 52
#define A1 55
#define AS1 58
#define B1 62
#define C2 65
#define CS2 69
#define D2 73
#define DS2 78
#define E2 82
#define F2 87
#define FS2 93
#define G2 98
#define GS2 104
#define A2 110
#define AS2 117
#define B2 123
#define C3 131
#define CS3 139
#define D3 147
#define DS3 156
#define E3 165
#define F3 175
#define FS3 185
#define G3 196
#define GS3 208
#define A3 220
#define AS3 233
```

Parte de código que
mostrare

```
#define B3 247
#define C4 262
#define CS4 277
#define D4 294
#define DS4 311
#define E4 330
#define F4 349
#define FS4 370
#define G4 392
#define GS4 415
#define A4 440
#define AS4 466
#define B4 494
#define C5 523
#define CS5 554
#define D5 587
#define DS5 622
#define E5 659
#define F5 698
#define FS5 740
#define G5 784
#define GS5 831
#define A5 880
#define AS5 932
#define B5 988
#define C6 1047
#define CS6 1109
#define D6 1175
#define DS6 1245
#define E6 1319
#define F6 1397
#define FS6 1480
#define G6 1568
#define GS6 1661
#define A6 1760
#define AS6 1865
#define B6 1976
#define C7 2093
#define CS7 2217
#define D7 2349
#define DS7 2489
#define E7 2637
#define F7 2794
#define FS7 2960
#define G7 3136
#define GS7 3322
#define A7 3520
#define AS7 3729
#define B7 3951
#define C8 4186
#define CS8 4435
#define D8 4699
#define DS8 4978
```