**Carpeta de Campo**

**Grupo 3:**Joaquín Baz, David Llacua, Lucas Solis e Ignacio Rosales.

**Proyecto de arduino:**Simon says

**Carpeta de:** Joaquín Baz



El arduino físico se trabaja los días viernes y la carpeta de campo se trabaja en cualquier momento si es en casa o en el colegio.

**Fecha de arranque del proyecto y carpeta:** 25/10/2024

El proyecto que vamos a realizar va a ser Simon says “Simón dice”, lo cual tenemos que hacer la réplica de juego de luces en el arduino.

En este trabajo decidimos darle lo más fácil a 3 personas que no están tan concentrados con el código de arduino solo Nacho y Joaquín saben hacer código tanto físico. En la primera clase el que armó el arduino fue Joaquín y Llacua ya que solís y nacho se encargaron del código, llegamos a muy poco del proyecto en esa misma clase faltó luciano cadima. Hasta después que decidimos hacer cambio de roles.

En este proyecto hicimos el simon says clásico no llegamos a meterle niveles de dificultad solo lo hicimos por rondas. Más abajo voy a estar explicando nuestras primeras pruebas como modificamos el código y cómo decidimos llegar a otra modificación del diseño del simon says.

En esta carpeta se van a encontrar fotos de las primeras pruebas con leds RGB ya que al pasar del tiempo fuimos cambiando muchas cosas, tanto en código como diseño. Más abajo se encuentra el código real del proyecto, que días faltó el profe, que fechas estuvimos siguiendo y que días fui a la mañana para continuarlo y que

**Comienzo:**

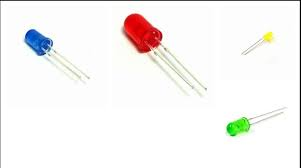
Arrancamos armando el código en el tinkercad y luego lo pasamos al programa de la computadora y a la vez armamos el arduino físico. Cambiamos un par de materiales y no usamos los mismos que en el tinkercad reemplazamos los botones que llevan la resistencia por otros distintos que no la llevan. También cambiamos el led común a el led RGB.

Los materiales que usamos en el arduino para el simon dice son: Cable USB que va conectado de la PC al arduino, Protoboard (placa de pruebas), 4 leds (RGB), Buzzer y 4 Pulsadores.

Vamos a usar el led de cada color en vez del RGB los leds que vamos a utilizar son color Rojo, Azul, Verde y amarillo todavia no sabemos si vamos a remplazar el led amarillo por otro color. El primer código lo usamos de prueba para ver cómo funcionan los leds y cómo funcionan los botones y el buzzer.

**Led comun:**

El LED es un componente digital de salida, es decir, su estado depende de la información que le entrega la tarjeta Arduino por medio de uno de sus pines. En este ejemplo conectaremos el pin de señal de nuestro LED al pin digital.



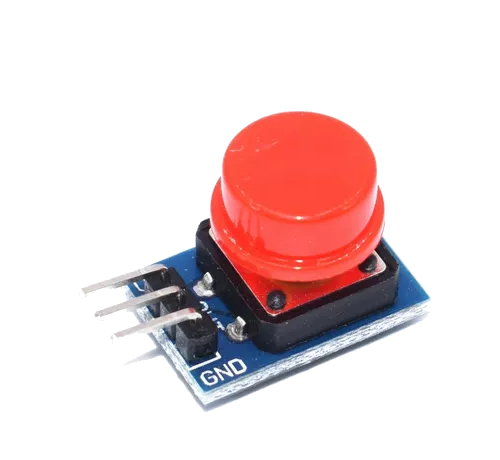
**Buzzer:**

Un zumbador (en inglés buzzer) es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono (generalmente agudo). Sirve como mecanismo de señalización o aviso y se utiliza en múltiples sistemas, como en automóviles o en electrodomésticos, incluidos los despertadores.



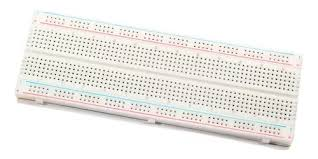
**Botón:**

Ahora hablamos de pulsador arduino, un botón que permite interrumpir o enviar una señal electrónica. Con el interruptor se pueden crear proyectos para muchas aplicaciones. El uso de este tipo de pulsadores es muy común cuando se trata de proyectos con arduino. No debemos confundir pulsador con interruptor.



**Protoboard:**

Una placa de pruebas (protoboard) es un tablero compuesto por orificios que se encuentran conectados eléctricamente entre sí de manera interna. En ella se pueden insertar elementos electrónicos y cables para el armado y prototipado de circuitos electrónicos.



**Arduino:**

El Arduino es utilizado como un microcontrolador, cuando tiene un programa descargado desde un ordenador y funciona de forma independiente de éste, y controla y alimenta determinados dispositivos y toma decisiones de acuerdo al programa descargado e interactúa con el mundo físico gracias a sensores.

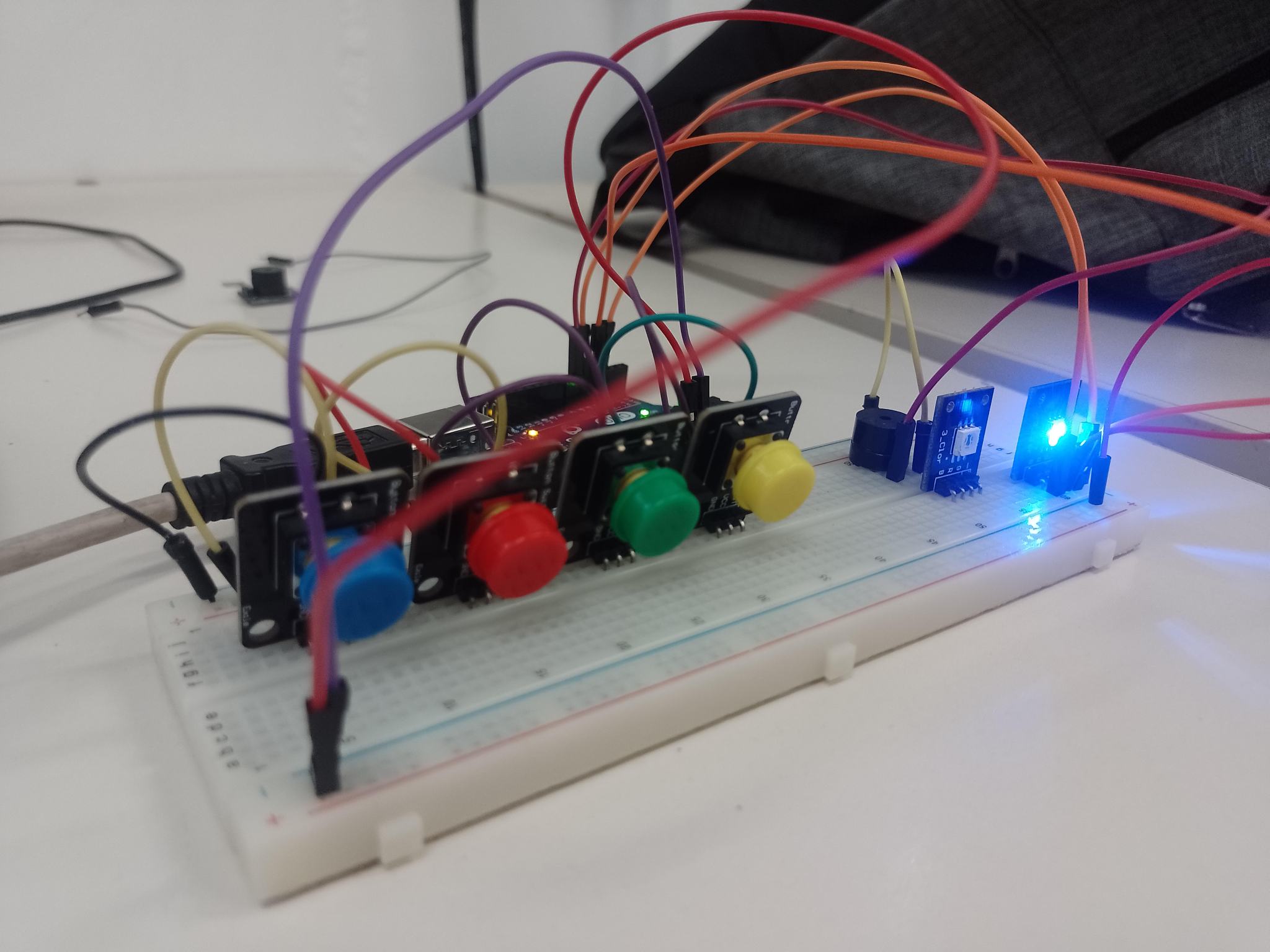


**Primeros procesos:**

Primero prendemos el primer led RGB que encendió verde y azul luego tenemos que probar los otros leds para que los primeros niveles sean el simon dice clásico.

Lo que tenemos que corregir es el buzzer que cuando lo encendemos no produce el sonido enciende la luz rgb azul y verde pero por el momento tenemos que corregir el buzzer.

La idea es que cuando hace el zumbido el buzzer y parpadean los led (Todos) ahí es cuando comienza el juego. Todos los leds menos uno tienen que estar apagados.

En el código del arduino encendimos el led rgb pero no de un solo color (como tiene que ser). Encendieron de 2 colores distintos como primeras pruebas esta bien solo que hay que corregir el buzzer al principio hicimos un mini simón dice de 2 colores y bastante bien al principio con los zumbidos del buzzer incluido.

Esa es toda la clase del viernes que fue el arranque del proyecto.

**La proxima clase sera el 1/11/2024**

**Clase 1/11/2024**

El profe faltó

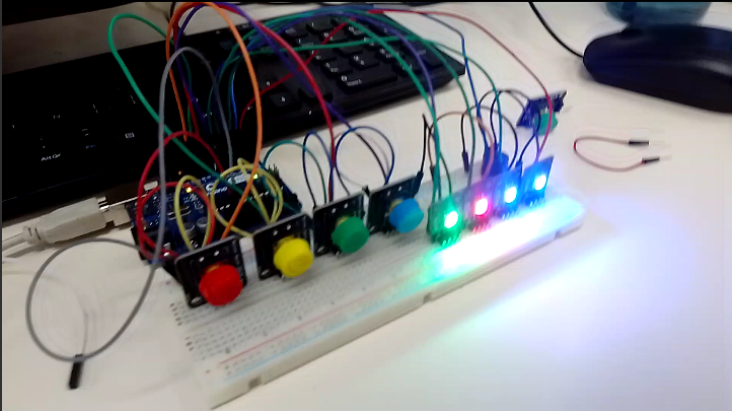
En esta clase el profe no vino ya que el jueves **31/10** en la hora de base de datos avanzamos un poco en lo físico. Corregimos el buzzer y conectamos los leds que nos quedaban en ese mismo tiempo me ayudó luciano cadima por el momento estamos en el primer nivel. En el grupo decidimos hacer cambio de tareas Nacho y Joaquín se van a encargar de la parte de código y Lucas, Luciano y David se van a encargar en lo físico.

Por el momento configuramos los zumbidos del buzzer y el delay cuando los leds encienden comienza del juego la secuencia de luces va para el lado de import random de forma aleatoria.

Por el momento así estamos en el físico de arduino debido a la última clase que faltó consorti estamos atrasados en nivel físico del arduino se nos complicaba para seguirlo en los días de la mañana. Es complicado seguir la carpeta de campo ya que no podemos seguir el físico de simón dice y los pocos minutos que tuvimos en la hora de base de datos no me ayudó mucho solo me dejo corregir algunas cosas del arduino y su componentes.

**Clase 8/11/24:**

En esta clase como mencione cambiamos los roles de cada uno Nacho y Joaquín se encargan del código, Lucas, Lucho y Llacua se encargan de lo físico pudimos ya encender todos los leds de cada color solo que el color que nos falta es el color amarillo.



En ese mismo momento teníamos que modificar el código para cambiar de color el ultimo led que esta de color azul pero con el tiempo en el que estuvieron armando el arduino se complicó ya que tenían problemas para conectar los leds.

**Clase 15/11/24:**

En esta clase faltó el profe.

**Clase 22/11/24:**

En esta clase estuvimos muy flojos con el tiempo y con lo físico del proyecto ya que hicimos otra modificación de materiales y usamos otros botones que van con resistencia y no nos fue muy bien ya que con el tiempo nos la jugamos y tuvimos errores en el código.

Pero en la próxima clase vamos a seguirlo como en la primera vez que lo configuramos, también tuvimos dificultades con los de la noche ya que tuvieron que mover nuestro pines y cables, y eso dificulta nuestro armado. No vamos a volver a modificar el armado ya que nos salió mal y perdimos toda una clase.

**Clase 29/11/24:**

En esta clase modificamos la mitad del código decidimos cambiar a un led de colores y un error que tuvimos con los botones, cuando abrimos el monitor en serie aparecen los botones que se presionan solos, decidimos cambiar el diseño tinkercad. No avanzamos mucho solo hicimos pruebas para saber que nos convenía no tenía a mi compañero nacho para arreglar el código ya que modificamos la mitad e hicimos pruebas pero nada nos sirvió, fue tiempo perdido y la única forma de seguirlo es en mi casa y modificando tanto como diseño tanto como en código.

No fue buena idea usar un solo led es más sencillo tener los botones en los 4 leds un botón que quede para cada uno pensamos que íbamos a ahorrar tiempo al ir probando diferentes opciones.

**Dia 1/12/24:**

Dia domingo decidí rearmar todo el simon dice a mi manera usando un video de ejemplo de youtube, pude configurar el monitor en serie para que diga en qué momento se perdió el juego y en el momento que gane la ronda.

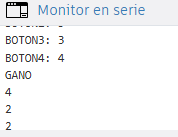
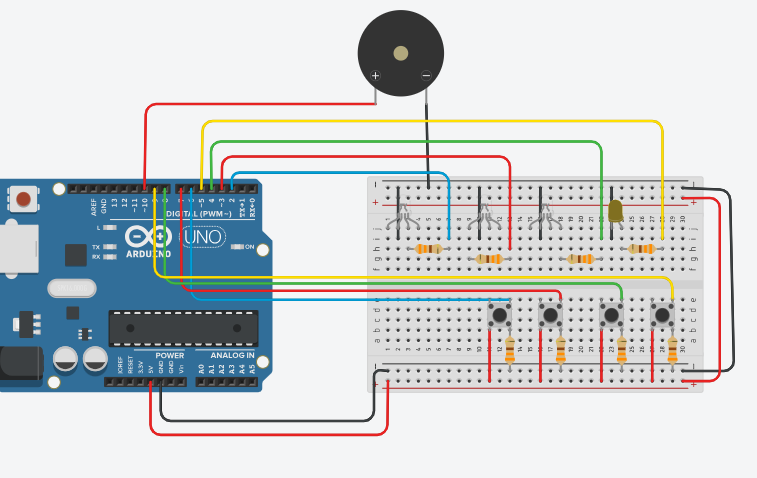
El 4to led es el problema está en color rojo y tenemos que corregir el código fue algo parecido a las primeras fotos que saqué en la página 7 no vale la pena usar un solo led eso mismo lo estuvimos probando el día viernes 29/11. Pero no sirvió y en este circuito nuevo que diseñe funciona mejor y queda guardada la secuencia del color al apretar el botón.

El error es el 4to led queda nomas solucionar el color y una pequeña modificación en el código y queda el proyecto hecho con el primer nivel, si el usuario se equivoca la secuencia del primer led cambia puede cambiar totalmente el orden si jugamos la primera y perdemos se reinicia solo y cambia el orden de encendido de led no es siempre el mismo.

El principio de código antes de poner el void setup.

Arriba que está las primeras filas de lo aleatorio y de los botones que voy a utilizar 4 botones y 4 leds ahora a continuación paso el codigo del void set up estamos haciendo pruebas si elegimos los leds de un color o uno que sea RGB y otro de un color solo como por ejemplo uso uno RGB para el rojo, verde y azul y uno aparte que sea del color amarillo.

Cuando comienza el juego es donde se encienden 4 colores puede ser de 1ed u otro cuando termina el momento donde los leds se apagan ahí es donde la persona entra a jugar en el caso de apretar los botones tiene que hacerlo uno a la vez y despacio por que o sino no se guarda la jugada correctamente. La secuencia de leds es aleatoria al momento de presionar los botones cuando el usuario ya presiono los leds de forma correctamente se encienden todos y mientras que el buzzer suena que pasamos a la siguiente ronda.



**Días 1/2/3/4 de diciembre:**

Esos días estuve modificando el código y armando el arduino físico por el momento llegamos al proyecto base no se sabe si llegamos a meterle algo extra a nuestro proyecto por el momento seguimos así para el que quiera probar el simon says vamos a conseguir un led de color amarillo alfinal el arduino nos quedaria así si me alcanza tiempo vemos si metemos un led o 2 leds demas.

Más abajo voy a poner el código donde están los leds ubicados y sus pines, también voy a estar explicando las funciones de mi código. Este código es el que voy a usar para la presentación ya que el otro no nos servía de mucho. En el código ponemos en donde están ubicados los leds y los botones ya después en el void loop ahí es donde se configura el apagado de los botones. El buzzer también se encarga de hacer los zumbidos cuando ganamos la ronda y en el momento que apretamos los leds y también el zumbido cuando perdemos.

**Dia 5/12/24:**

En este dia consorti me ayudo a solucionar el error de los botones

| Acá en este apartado ya tenemos los leds, botones y el buzzer ubicado ahora paso a explicar como funciona lo otro del código de void loop. En el void loop tenemos guardados los leds en high y low.  **Generación de números aleatorios:**  El primer bloque condicional verifica si las variables Aleatorio1, Aleatorio2, Aleatorio3, y Aleatorio4 son todas iguales a 0. Si es cierto, se ejecutará el código dentro de este bloque.  Si la condición es verdadera, se generan números aleatorios entre 2 y 5 para cada una de las variables Aleatorio1, Aleatorio2, Aleatorio3 y Aleatorio 4. Para esto se usa la función random, que genera un número aleatorio especificado. La condición comprueba que todas las variables estén en 0. Si alguna de ellas no es 0, el código dentro de este if no se ejecutará.  A medida que vamos jugando hay una dificultad en 2 leds ya que se encienden en poco tiempo y hay que memorizar rápido. En las primeras 2 secuencias de luces el nivel es intermedio ya después en las últimas 2 secuencias de luces el nivel de delay está modificado para dificultar el juego. |
| --- |

El delay en aleatorio 1 y 2 está configurado el delay en 500 tanto cuando prende y tanto cuando se prende y apaga. El delay en aleatorio 3 y 4 el delay en prendido está configurado en 100 y en apagado en 50.

Aca tenemos la primera secuencia de luz configurado en dificultad normal también en aleatorio 2 está configurado a la dificultad normal esta igual que en aleatorio 2 y después pasó a explicar como esta configurado el delay en aleatorio 3 y 4.Acá cambia en dificultad difícil en las últimas 2 secuencia de luz el delay sigue en 500 tanto prendido y apagado. Después paso el código de los botones como está configurado el delay cada botón está en 500 de delay es un tiempo justo a la hora de jugar ya que antes cuando estaba en 1000 todo el juego se hacía lento y cuando se pasaba de ronda se hacía muy largo hicimos muchas modificaciones con tema de tiempo a la hora de jugar con las luces botones y buzzer. El monitor en serie sigue igual nos avisa cuando ganamos y a la hora de cuando perdemos más abajo en al otra pagina voy a pasar y explicar los botones y el zumbido del buzzer. Esa configuración vamos a estar usando para los demás botones sea botón 2, 3 y 4. Después usamos serial print en los botones que tenemos en la protoboard ya que hicimos lo mismo con los leds y el buzzer.

Explicación paso a paso del código.

| int ALEATORIO1 = 0;  int ALEATORIO2 = 0;  int ALEATORIO3 = 0;  int ALEATORIO4 = 0;  int BOTON2 = 0;  int BOTON3 = 0;  int BOTON4 = 0;  int BOTON1 = 0; |
| --- |

Aca definimos los leds y botones que vamos a usar son 4 leds y 4 botones cambiamos pines en el orden del arduino y voy a explicar que utilizamos en el void setup buzzer, leds y botones. Después queda explicar la funciones de lo que utilice para guardar las jugadas y cómo se comporta los botones ya que tuve errores a la hora de ver en el monitor en serie ya que se presionaban en automático

**INT:** Sirve para guardar números enteros, es decir, cualquier número positivo o negativo pero sin decimales.

**OUTPUT:** Una salida digital es un dispositivo que permite cambiar su voltaje a uno de estos dos valores

**INPUT:** Es cada pin definido para leer el estado de entrada.

**pinMode:** Configura el pin especificado para que se comporte como entrada o salida.

**Serial.begin:** Abre un puerto serie y especifica la velocidad de transmisión.

**randomSeed:** Si es importante que una secuencia de valores generada por random() difiera, en ejecuciones.

**Delay:** Controla el tiempo en milisegundos para buzzer, leds, botones, etc.

| void setup() {  Serial.begin(9600);  pinMode(2, OUTPUT);  pinMode(3, OUTPUT);  pinMode(4, OUTPUT);  pinMode(5, OUTPUT);  pinMode(13, OUTPUT);  pinMode(9, INPUT);  pinMode(8, INPUT);  pinMode(11, INPUT);  pinMode(12, INPUT);  randomSeed(analogRead(A0)); } |
| --- |

Tenemos los leds conectados al OUTPUT después los botones INPUT y abajo después ponemos random seed analogread para luego conectar el led dependiendo del aleatorio de forma aleatoria . Después en el void loop tenemos toda la escritura del buzzer con los zumbidos a la hora de ganar o perder el juego.

| void loop() {  if ((ALEATORIO1 == 0 && ALEATORIO2 == 0) && (ALEATORIO3 == 0 && ALEATORIO4 == 0)) {  ALEATORIO1 = random(2, 5 + 1);  ALEATORIO2 = random(2, 5 + 1);  ALEATORIO3 = random(2, 5 + 1);  ALEATORIO4 = random(2, 5 + 1);  Serial.println(ALEATORIO1);  Serial.println(ALEATORIO2);  Serial.println(ALEATORIO3);  Serial.println(ALEATORIO4);  delay(1000); |
| --- |

Los leds a la hora de prender las luces están de delay a 1000 un tiempo bien memorizable a la hora de jugar. Tenemos los zumbidos a la hora de presionar el botón y que se guarde la jugada.

| // APAGAMOS TODOS LOS LEDS  digitalWrite(2, LOW);  digitalWrite(3, LOW);  digitalWrite(4, LOW);  digitalWrite(5, LOW);  delay(500); |
| --- |

Los leds se apagan un delay de 500 y toca presionar los botones para luego pasar de ronda más abajo paso a explicar los aleatorios que están en HIGH. Y en donde tiene los zumbidos a la hora de hacer la jugada.

| if (ALEATORIO2 == 2) {  digitalWrite(2, HIGH);  tone(13, 294, 500);  }  if (ALEATORIO2 == 3) {  digitalWrite(3, HIGH);  tone(13, 294, 500);  }  if (ALEATORIO2 == 4) {  digitalWrite(4, HIGH);  tone(13, 165, 500);  }  if (ALEATORIO2 == 5) {  digitalWrite(5, HIGH);  tone(13, 92, 500);  } |
| --- |

Después el código sigue igual al definir los aleatorios, pasamos a definir los botones:

| Serial.print(digitalRead(8)); Serial.print(" "); Serial.print(digitalRead(9)); Serial.print(" "); Serial.print(digitalRead(11)); Serial.print(" "); Serial.println(digitalRead(12));    if (digitalRead(9) == LOW && BOTON1 == 0) {  tone(13, 92, 500);  BOTON1 = 5;  digitalWrite(5, HIGH);  delay(200);  digitalWrite(5, LOW);  delay(500);  }else if (digitalRead(8) == LOW && BOTON1 == 0) {  tone(13, 165, 500);  BOTON1 = 4;  digitalWrite(4, HIGH);  delay(200);  digitalWrite(4, LOW);   delay(500); |
| --- |

Los botones siguen en la misma función con los demás hasta llegar a definir high y low con los leds.

| Serial.println("GANO");  tone(13, 1000, 1000);  digitalWrite(5, HIGH);  digitalWrite(4, HIGH);  digitalWrite(3, HIGH);  digitalWrite(2, HIGH);  delay(100); |
| --- |

Después de ganar la ronda pasamos a la otra y se resetea el juego hasta el momento de perder.

| digitalWrite(5, HIGH);  digitalWrite(4, HIGH);  digitalWrite(3, HIGH);  digitalWrite(2, HIGH);  noTone(13);  delay(1000);  digitalWrite(5, LOW);  digitalWrite(4, LOW);  digitalWrite(3, LOW);  digitalWrite(2, LOW);  delay(500);  } |
| --- |