

# DISEÑO DE SISTEMAS EMPOTRADOS

---

Memoria del Proyecto: **Simón Dice**

**Grupo 22**

Adrián Sanz Hervás

Baldomero Rodríguez Árbol

Nelson Gabriel Arze Martínez

Raúl de la Fuente Carrasco

# ÍNDICE

<b>INFORMACIÓN SOBRE EL PROYECTO</b>	<b>2</b>
<b>IMPLEMENTACIÓN</b>	<b>3</b>
SOFTWARE	3
HARDWARE	3
<b>PASOS DADOS</b>	<b>4</b>
<b>REPARTO DE TAREAS</b>	<b>5</b>
<b>COSTES</b>	<b>6</b>
<b>PROBLEMAS Y SOLUCIONES</b>	<b>7</b>
<b>CÓDIGO</b>	<b>8</b>
<b>CASOS DE USO</b>	<b>13</b>

# INFORMACIÓN SOBRE EL PROYECTO

El proyecto elegido por el grupo 22 es un Simón Dice. El juego se basa en una repetición de luces de distintos colores, el jugador deberá repetir la misma secuencia de colores en el mismo orden.

El proyecto tiene otras funciones implementadas para hacerlo más complejo, estas son:

- Dos tipos distintos de dificultades:
  - Número de colores: El jugador podrá elegir si hacer el juego con tres o cuatro colores diferentes.
  - Tiempo entre luces: La cantidad de tiempo que habrá entre luz y luz en cada secuencia del juego.
- Un timbre que sonará de forma distinta si se ha acertado o se ha fallado la secuencia de colores. Además el timbre sonará al principio de cada ronda.
- Un botón para reiniciar el juego en caso de que se quisiera hacer por algún motivo.
- Las secuencias de colores son infinitas hasta que el jugador falle una secuencia.

# IMPLEMENTACIÓN

## SOFTWARE

Con respecto a la implementación del juego de Simón Dice en Arduino, se hizo un selector de dificultad para elegir el modo de juego deseado y luego se hizo un bucle que añadía en cada iteración un número aleatorio (entre 1 y el número de colores) a un array de enteros, en el que cada número representa un color.

Luego se recorre el array encendiendo los leds correspondientes y finalmente se espera hasta que el usuario pulse los botones en el orden adecuado, comparando cada botón pulsado con cada elemento del array. Si el usuario pulsa todos los botones correctamente, se vuelve al inicio del bucle y el timbre suena con un sonido corto, pero si el usuario se equivoca se sale del bucle, suena el timbre más largo y agudo y se vuelve al estado del selector de dificultad.

El código completo se encuentra en el apartado de Código de la memoria.

## HARDWARE

Se conectaron cinco leds en paralelo (cuatro para el juego y un led morado adicional para indicar el comienzo de las rondas) con sus correspondientes resistencias de  $220\Omega$  a la protoboard y a los puertos 8, 9, 10, 11 y 12 de Arduino.

Se conectaron también cinco botones en paralelo con resistencias de  $10k\Omega$  a la protoboard y a los puertos 2, 3, 4, 5 y 6 de Arduino. En esa misma protoboard se conectó el timbre al puerto 13 de Arduino. Ambas protoboards se conectaron a GND en arduino en su polo negativo

Por último se conectó Arduino al ordenador para alimentarlo y subir el código, y se hizo una caja para poder jugar con los botones de forma cómoda.

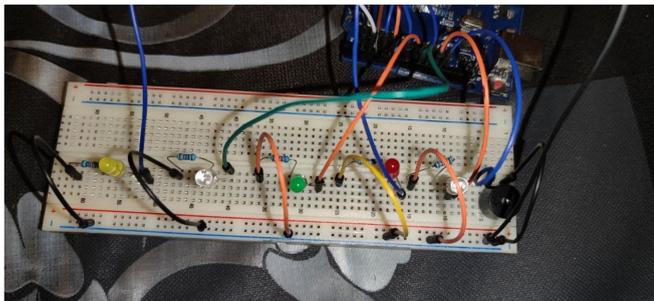
# PASOS DADOS

## Primeros pasos:

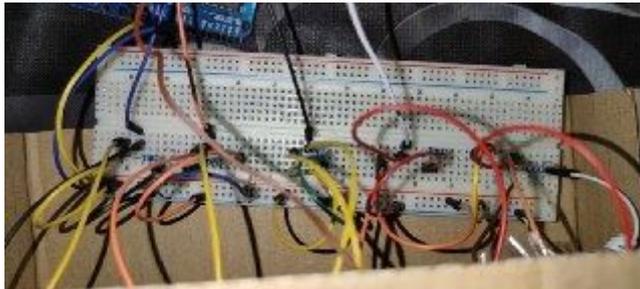
Lo primero que se hizo fue la planificación del proyecto, se pensaron las funcionalidad que se querían implementar, los posibles materiales a usar, hacer un pseudo código de la parte del software etc.

## Parte hardware:

- Preparación de materiales: leds, resistencias, cables, timbre etc.
- Una vez preparados los materiales, se dividió el trabajo en dos placas:
  - La placa destinada a los leds junto con el timbre.



- La placa destinada a los botones.



- Realizar las conexiones y montaje pertinente.
- Solucionar errores

## Parte software:

- Hacer el pseudocódigo.
- Preparar el código, que se mostrará en el apartado "CÓDIGO".
- Hacer las mejoras pertinentes (uso del timbre, dificultades etc).
- Solucionar errores.

## Parte de unión:

Durante el proceso de creación del software y hardware, se fueron probando distintas partes del código funcionales, con parte del hardware que se iba montando. Debido a esta continuidad no hubo que hacer cambios grandes ni en el código ni en el hardware, puesto que los cambios fueron graduales.

## Montaje de la caja:

Una vez que todo funcionaba correctamente, se montó la caja para poner los botones y preparar una mejor presentación del proyecto y grabar el video que se adjuntará en la entrega.

# REPARTO DE TAREAS

La distribución de las tareas se dividió en:

- Dos personas encargadas del software(Nelson y Baldomero).
- Dos personas encargadas del hardware(Adrián y Raúl).
- Fabricación de la caja conjunta entre todo el grupo.

Se eligió este método de división para agilizar el trabajo en el grupo y que cada parte, tanto hardware como software tuvieran una persona manejando el montaje/programación y un segundo asistiendo al manejador. La creación de la caja fue una idea unánime y al ser llevada a cabo después del resto de montaje, todo el grupo pudo ayudar en este apartado.

# COSTES

Lista de materiales comprados:

Descripción	Unidades	Precio
Caja de cartón y cola	1	5€
Botones pequeños	4	8€(2€/unidad)
Leds	5	2,50€ (50 céntimos/unidad)
Botones medianos	1	3,50€
<b>TOTAL</b>	-	19€

El resto de materiales que no se encuentran en esta lista, nos han sido facilitados por la universidad (como la placa de Arduino) o ya estaban en propiedad de alguno de los miembros del grupo.

# PROBLEMAS Y SOLUCIONES

Uno de los problemas más graves que ha sucedido en esta práctica, debido a las circunstancias adversas de nieve, uno de nuestros integrantes no pudo asistir a una de las sesiones de trabajo puesto que estaba atrapado en su oficina de trabajo y los servicios civiles no habían acudido todavía a rescatarlo.

Con relación al proyecto, aparte del gran trabajo que ha supuesto el montaje del hardware y el software, el mayor inconveniente llegó al juntar las diferentes partes puesto que por separado funcionaban pero juntas colapsaron.

Se tardó una gran cantidad de horas en encontrar los diferentes errores en ambas partes. Con respecto al hardware había 4 cables que estaban conectados a tierra, desde la entrada y la salida del botón, algo parecido nos sucedió con los Leds, solo que estos estaban conectados en el positivo de la protoplaca con lo cual el circuito no se cerraba correctamente, también estaban colocados muchos cables de manera redundante que únicamente enviaban corriente a los demás cables. Cambiando un poco la distribución del cableado, se consiguió no solo reducir el número de cables conectados sino que también el circuito cerraba y funcionaba correctamente.

Los fallos más graves que daba el software sucedían al detectar los fallos del juego y otro al pulsar los botones que enviaban señales aleatorias, las soluciones a estos problemas fueron una remodelación en las condiciones del bucle y registrar el botón al dejar de pulsar en vez de al pulsar como inicialmente se había programado.

# CÓDIGO

```
#include <stdlib.h>

//Inicialización del array de colores
int game[100];
int lon = 0;
bool playing = true;
bool pulsado = false;
int rand (void);

//Adjudicación de puertos a los botones
const int buttonRed = 6;
const int buttonGreen = 2;
const int buttonBlue = 3;
const int buttonYellow = 4;
const int buttonPurple = 5;

//Inicialización del estado de los botones
int botonTest = 0;

int buttonStateRed = 0;
int buttonStateGreen = 0;
int buttonStateBlue = 0;
int buttonStateYellow = 0;
int buttonStatePurple = 0;

//Adjudicación de puertos a los leds
const int lampRed = 8;
const int lampGreen = 9;
const int lampBlue = 10;
const int lampYellow = 11;
const int lampPurple = 12;

const int pinZumbador = 13;
const int frecuencia = 220;

//Electores de dificultad por defecto
bool chosen = false;
int wait = 500;
int numLamps = 4;

void setup() {
  Serial.begin(9600);
  Serial.write("start");
  pinMode(lampRed , OUTPUT);
  pinMode(lampGreen , OUTPUT);
```

```

pinMode(lampBlue , OUTPUT);
pinMode(lampYellow , OUTPUT);
pinMode(lampPurple , OUTPUT);
}

void loop() {
  playing = true;
  digitalWrite(lampPurple,HIGH);
  digitalWrite(lampRed,HIGH);
  digitalWrite(lampBlue,HIGH);
  chosen = false;
  while(!chosen){//Elector de dificultad
  botonTest = pressButton();
  switch(botonTest){
    case 1://ROJO
    digitalWrite(lampRed, HIGH);
    digitalWrite(lampGreen, LOW);
    numLamps = 4;
    break;
    case 2://VERDE
    digitalWrite(lampRed, LOW);
    digitalWrite(lampGreen, HIGH);
    numLamps = 3;
    break;
    case 3://AZUL
    digitalWrite(lampBlue, HIGH);
    digitalWrite(lampYellow, LOW);
    wait = 500;
    break;
    case 4://AMARILLO
    digitalWrite(lampBlue, LOW);
    digitalWrite(lampYellow, HIGH);
    wait = 200;
    break;
    case 5://LILA
    digitalWrite(lampRed, LOW);
    digitalWrite(lampGreen, LOW);
    digitalWrite(lampBlue, LOW);
    digitalWrite(lampYellow, LOW);
    digitalWrite(lampPurple, LOW);
    chosen = true;
    break;
    default:
    Serial.write("ERROR AL EMPEZAR");
  }
}
while(playing){
  Serial.write(" NUEVO CICLO ");
}
}

```

```

//Serial.write("lila on ");
tone(pinZumbador, frecuencia);
digitalWrite(lampPurple,HIGH);
delay(wait);
//Serial.write("lila off ");
digitalWrite(lampPurple,LOW);
noTone(pinZumbador);
delay(500);
game[lon] = rand() % numLamps + 1;
for(int i = 0; i < lon; i++){//RECORRE EL ARRAY DE COLORES ENCENDIENDO LEDS
  switch(game[i]){
case 1:
//Serial.write("rojo on ");
digitalWrite(lampRed,HIGH);
delay(wait);
//Serial.write("rojo off ");
digitalWrite(lampRed,LOW);
break;
case 2:
//Serial.write("verde on ");
digitalWrite(lampGreen,HIGH);
delay(wait);
//Serial.write("verde off ");
digitalWrite(lampGreen,LOW);
break;
case 3:
//Serial.write("azul on ");
digitalWrite(lampBlue,HIGH);
delay(wait);
//Serial.write("azul off ");
digitalWrite(lampBlue,LOW);
break;
case 4:
//Serial.write("amarillo on ");
digitalWrite(lampYellow,HIGH);
delay(wait);
//Serial.write("amarillo off ");
digitalWrite(lampYellow,LOW);
break;
default:
  Serial.write("ERROR AL ENCENDER LED ");
}
delay(wait);
}
for(int j = 0; j < lon; j++){
  if(playing==true){
    Serial.write("PREGUNTA");
    botonTest = pressButton();
  }
}

```

```

Serial.write("RESPONDIDO");
if(botonTest != game[j]){
  Serial.write("FALLO");
  tone(pinZumbador, 1000);
  delay(1000);
  noTone(pinZumbador);
  playing = false;
  lon = 0;
}
}
}
if(playing){
tone(pinZumbador, 500);
delay(300);
noTone(pinZumbador);
delay(500);
lon++;
}
}
}
bool down = false;
bool up = false;

int pressButton(){
int botonPulsado = 5;
down = false;
up = false;
do{
  //Serial.write(".");
  botonStateRed = digitalRead(buttonRed);
  botonStateGreen = digitalRead(buttonGreen);
  botonStateBlue = digitalRead(buttonBlue);
  botonStateYellow = digitalRead(buttonYellow);
  botonStatePurple = digitalRead(buttonPurple);
  if(botonStateGreen==HIGH){
    Serial.write("boton verde pulsado");
    botonPulsado = 2;
    down = true;
  }else if(botonStateRed==HIGH){
    Serial.write("boton rojo pulsado");
    botonPulsado = 1;
    down = true;
  }else if(botonStateBlue==HIGH){
    Serial.write("boton azul pulsado");
    botonPulsado = 3;
    down = true;
  }else if(botonStateYellow==HIGH){
    Serial.write("boton amarillo pulsado");
  }
}
}

```

```
    botonPulsado = 4;
    down = true;
}else if(buttonStatePurple==HIGH){
    Serial.write("boton lila pulsado");
    botonPulsado = 5;
    down = true;
}
}while(!down);
do{
    //Serial.write("-");
    botonStateRed = digitalRead(buttonRed);
    botonStateGreen = digitalRead(buttonGreen);
    botonStateBlue = digitalRead(buttonBlue);
    botonStateYellow = digitalRead(buttonYellow);
    botonStatePurple = digitalRead(buttonPurple);
    if(buttonStateRed==LOW && buttonStateGreen==LOW && buttonStateBlue==LOW &&
botonStateYellow==LOW && botonStatePurple==LOW){
        up = true;
    }
}while(!up);
//Serial.write("I");
return botonPulsado;
}
```

# CASOS DE USO

El proyecto diseñado funciona de la siguiente manera, al iniciarse el zumbador suena indicando que está operativo, en este estado estaría en el menú principal, donde podemos escoger dificultad, la dificultad se basa en dos variables, la cantidad de leds entre los que elegir por parte del programa para la generación del array de respuesta correcta, el cual empezará con un array de un solo elemento, aumentando con cada acierto en 1 más, la elección de esta parte de la dificultad es accesible de forma simple pulsando el primer pulsador para determinarla en 4 leds operativos o el segundo pulsador, dejándolo en 3 leds.

La segunda variable que determina la dificultad del juego es el tiempo de reacción, este configura el tiempo de separación entre leds a la hora de mostrar la combinación que debe ser acertada, esta varía entre 200 ms y 500 ms, es accesible mediante el tercer y cuarto pulsador, siendo el tercero el de 500ms y el cuarto el de 200ms, el único botón no mencionado, el primero, será el encargado de iniciar el juego.

En caso de no haber seleccionado dificultad la predeterminada será cuatro leds para elegir por el programa y 500 ms de retardo entre luces durante la serie.

Tras haber seleccionado o no la dificultad el zumbador emitirá un pitido indicando el inicio de la ronda, este se repetirá con el inicio de cada ronda al acertar, en caso de fallar la serie, el zumbador emitirá un pitido más agudo y largo que el de inicio de ronda, y volviendo al menú principal para volver a escoger dificultad y empezar el juego.