

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Organización Computacional
Ing. Juan Carlos Maeda Juárez
Aux: Javier Gutierrez



Proyecto LECTOR DE BRAILE

NOMBRES:	CARNET:	% PARTICIPACIÓN
Andrés Alejandro Agosto Mendez	202113580	25%
Engel Emilio Coc Raxjal	202200314	25%
Carmen María Marroquín Llamas	202001132	25%
Lizz Andrea Morelia Castellanos Salazar	20170897	25%

INTRODUCCIÓN

El presente tiene como objetivo la creación de un prototipo de lector de Braille. Este proyecto busca abordar un problema específico de la comunidad de personas ciegas en Guatemala. El prototipo desarrollado se centrará en la verificación del texto en tarjetas individuales, utilizando tecnologías y conocimientos adquiridos durante el curso.

En el desarrollo del proyecto se implementó de varios componentes como flip-flops, las diferentes compuertas lógicas, transmisión serial, motores paso a paso (stepper), y sensores de color. Además, se aplicarán conocimientos avanzados de memoria RAM y lógica secuencial para asegurar un funcionamiento eficiente del sistema.

OBJETIVOS

General

Aplicar los conocimientos teóricos aprendidos en clase magistral y laboratorio para la construcción de circuitos combinacionales.

Específicos

1. Implementar una transmisión serial a través de los puertos de una PC.
2. Desarrollar una estructura mecánica funcional.
3. Aplicar e implementar conocimientos de registros.
4. Implementar software para el control de puertos.
5. Aplicar e implementar conocimientos de memorias.
6. Aprender conocimientos de lógica secuencial y simplificación de estados.
7. Aprender el funcionamiento y uso de motores paso a paso (Stepper).
8. Integrar circuitos eléctricos y producir movimientos mecánicos.
9. Aprender la aplicación de una memoria RAM y aplicación de Arduino.
10. Uso y manejo de distintos tipos de Flip Flops para contadores.

MARCO TEÓRICO

Memorias y Registros

Las memorias y los registros son componentes críticos en cualquier sistema digital. En este proyecto, se utilizará una memoria RAM para almacenar temporalmente hasta 8 letras en formato binario. La memoria RAM se gestionará utilizando el "heap", una estructura de datos dinámica que permite la reserva y liberación de memoria durante la ejecución del programa. Esta memoria dinámica es esencial para el manejo eficiente de los datos recibidos y procesados por el sistema.

Heap

El heap es una estructura de datos que permite la gestión dinámica de la memoria. A diferencia de la pila de ejecución, que maneja variables locales y se ajusta automáticamente, el heap requiere una gestión explícita, permitiendo la reserva y liberación de memoria según sea necesario. En el contexto de este proyecto, el heap se utilizará para almacenar los datos de las letras leídas por los sensores en formato ASCII binario.

Lógica Secuencial

La lógica secuencial es una parte fundamental del diseño digital, que implica la utilización de flip-flops, contadores, y máquinas de estado finitas para crear circuitos que dependen no solo de las entradas actuales, sino también del historial de entradas pasadas. En este proyecto, la lógica secuencial se utilizará para crear contadores que controlen los pulsos de reloj generados por el Arduino cada vez que se mueve el eje a una nueva posición.

Contadores y Flip-Flops

Los contadores son dispositivos que almacenan y cuentan pulsos de reloj, mientras que los flip-flops son elementos básicos de almacenamiento que pueden mantener un estado binario. Estos componentes son esenciales para la creación de sistemas que requieren un seguimiento preciso de eventos secuenciales, como el movimiento de los ejes X e Y en el lector de Braille.

Transmisión Serial

La transmisión serial es una técnica de comunicación de datos en la que los bits se envían uno tras otro a través de un único canal. Este método es eficiente para la comunicación entre dispositivos, especialmente en sistemas embebidos como el Arduino. En el proyecto, se utilizará la transmisión serial para enviar y recibir datos entre el Arduino físico y otro Arduino dentro del software de simulación Proteus.

Arduino y Comunicación Serial

Arduino es una plataforma de hardware libre basada en una sencilla placa con un microcontrolador y un entorno de desarrollo integrado (IDE). La comunicación serial en Arduino se facilita mediante bibliotecas integradas que permiten el envío y recepción de datos a través de puertos seriales, haciendo posible la interacción entre el hardware y el software de simulación.

Motores Paso a Paso (Stepper)

Los motores paso a paso son motores que se mueven en pasos discretos, permitiendo un control preciso del movimiento y la posición. Estos motores son ideales para aplicaciones donde se requiere un posicionamiento exacto, como en el desplazamiento de sensores en los ejes X e Y del lector de Braille.

Funcionamiento de los Motores Stepper

Los motores paso a paso funcionan mediante la activación secuencial de sus bobinas, lo que genera un movimiento incremental del eje. Este tipo de motor permite un control detallado del ángulo de rotación, lo que es crucial para el posicionamiento exacto de los sensores que verifican las letras en las tarjetas.

Sensores de Color

Los sensores de color son dispositivos que detectan la presencia de colores específicos en una superficie. En el proyecto, se utilizarán sensores de color para verificar el correcto posicionamiento de las tarjetas y para identificar las letras en las tarjetas. Los sensores de color operan mediante la detección de la luz reflejada desde la superficie de la tarjeta, permitiendo la identificación de los colores predefinidos.

Implementación de Sensores de Color

La implementación de sensores de color en el proyecto requiere la calibración y programación adecuada para asegurar que los sensores puedan distinguir correctamente los puntos de posicionamiento y las letras. Estos sensores se integrarán con los motores stepper y el sistema de control del Arduino para proporcionar datos precisos sobre la posición y el contenido de las tarjetas.

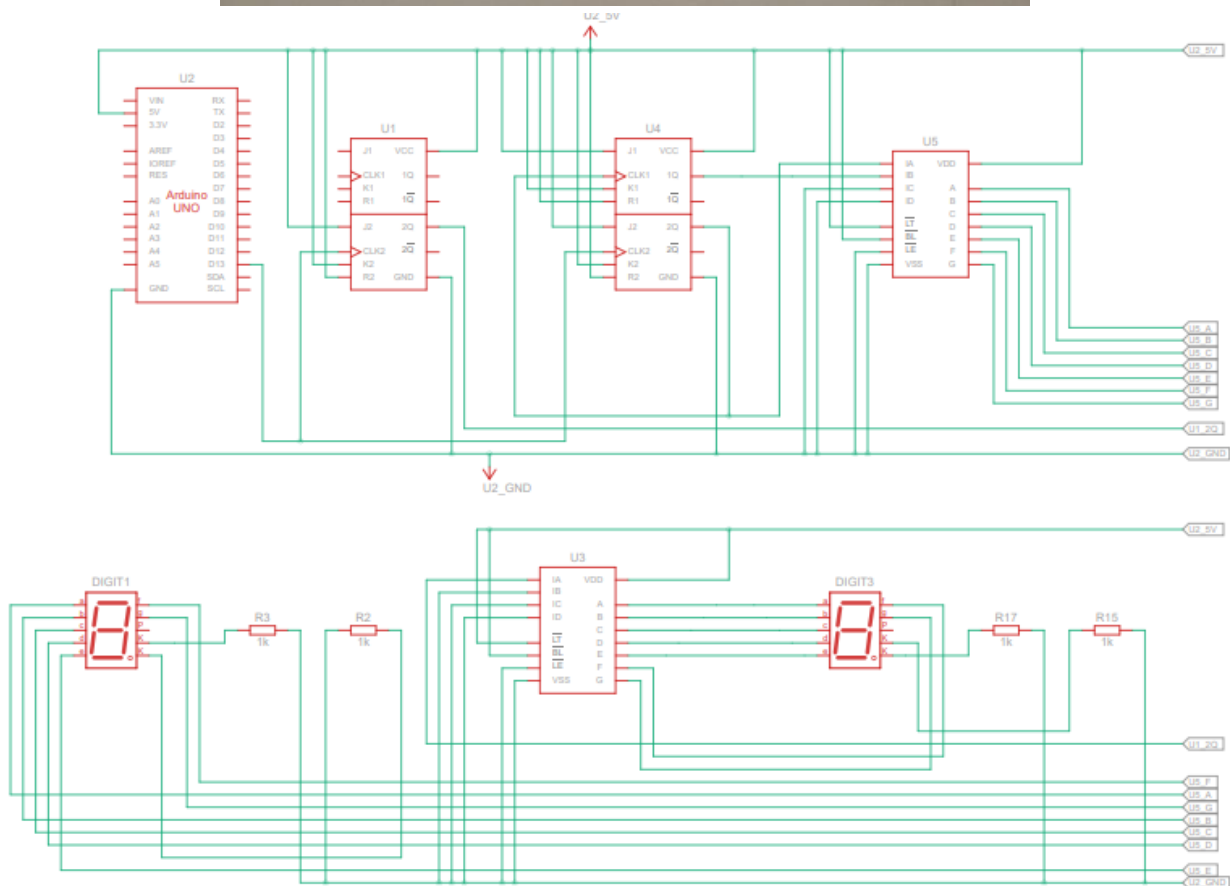
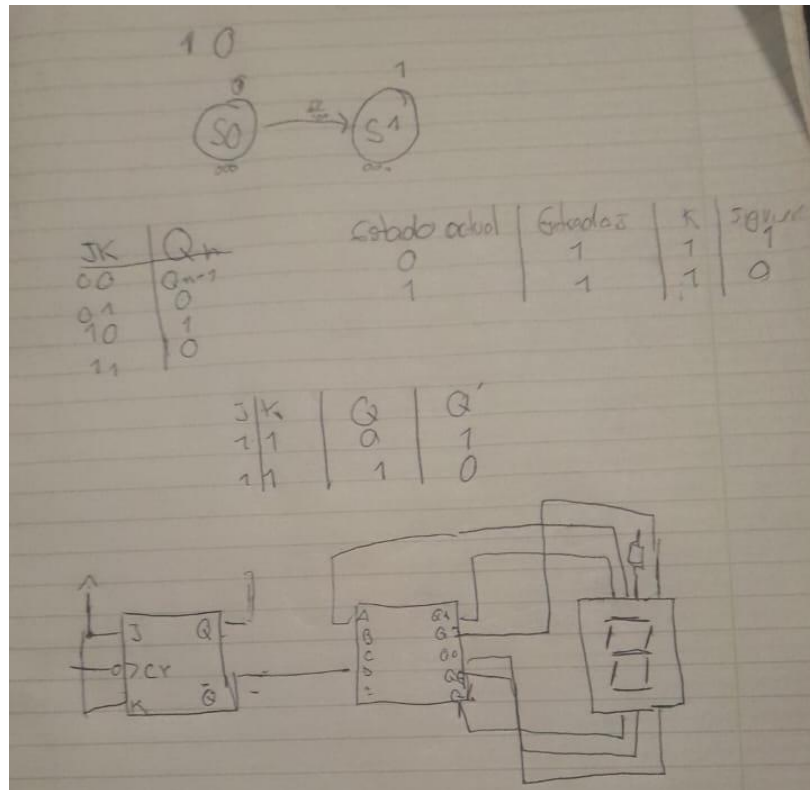
Diseño y Simulación de Circuitos

El diseño y la simulación de circuitos son pasos cruciales en el desarrollo de sistemas electrónicos complejos. En este proyecto, se utilizará el software Proteus para simular el circuito combinacional y secuencial necesario para el funcionamiento del lector de Braille.

Proteus

Proteus es una suite de software utilizada principalmente para la simulación de circuitos electrónicos. Permite a los ingenieros diseñar y probar sus circuitos en un entorno virtual antes de implementarlos físicamente. En este proyecto, Proteus se utilizará para diseñar y simular la memoria RAM, los contadores, y otros componentes del sistema.

CÓDIGO Y TABLA DE CONTADOR



Stepper:

```
// Conexión arduino con driver
```

```
// Motor y
```

```
const int IN1 = 13;
```

```
const int IN2 = 12;
```

```
const int IN3 = 11;
```

```
const int IN4 = 10;
```

```
// Motor x
```

```
const int MIN1 = 9;
```

```
const int MIN2 = 8;
```

```
const int MIN3 = 7;
```

```
const int MIN4 = 6;
```

```
int Sensor1 = 0;
```

```
int Sensor2 = 0;
```

```
int Sensor3 = 0;
```

```
int Sensor4 = 0;
```

```
int NumeroDePasos = 512;
```

```
int LetraBraille[3][2] = // Coordenada x,y
```

```
{
```

```
  {0,0},
```

```
  {0,0},
```

```
  {0,0},
```

```
};
```

```
int demora =3; // demora entre pasos, no debe ser menor a 10 ms.
```



```
// paso completo con maximo torque
```

```
int PasoCompleto[4][4] = // matriz (array bidimensional) con la secuencia de pasos
```

```
{
  { 1, 1, 0, 0 },
  { 0, 1, 1, 0 },
  { 0, 0, 1, 1 },
  { 1, 0, 0, 1 }
};
```

```
// Medio Paso
```

```
int pasolnv[4][4] = // matriz (array bidimensional) con la secuencia de pasos, pero en orden
inverso
```

```
{
  { 1, 0, 0, 1 },
  { 0, 0, 1, 1 },
  { 0, 1, 1, 0 },
  { 1, 1, 0, 0 }
};
```

```
void setup() {
```

```
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
```

```
  pinMode(MIN1, OUTPUT);
  pinMode(MIN2, OUTPUT);
  pinMode(MIN3, OUTPUT);
  pinMode(MIN4, OUTPUT);
```

```
}
```

```
void loop() {
```

```

// Motor y
//Mover(NumeroDePasos,PasoCompleto, "++",IN1, IN2,IN3,IN4); //++
//Mover(NumeroDePasos,PasoCompleto, "--",IN1, IN2,IN3,IN4); // --

// Motor x
// Mover(NumeroDePasos,PasoCompleto, "++",MIN1, MIN2,MIN3,MIN4); // ++
//Mover(NumeroDePasos,PasoCompleto, "--",MIN1, MIN2,MIN3,MIN4); // --
//Detener();

//LecturaBraille();

if ( Sensor1 == 1 && Sensor2 == 1 && Sensor3==1 && Sensor4==1){

}

}

void Mover(int NumeroDePasos, int TipoPaso[4][4], String Direccion, int Bobina1, int
Bobina2, int Bobina3, int Bobina4){
    // Mueve el motor hacia adelante
    for (int i = 0; i < NumeroDePasos; i++) {

        for (int j = 0; j < 4; j++){
            if (Direccion=="--"){
                digitalWrite(Bobina1, TipoPaso[j][0]);
                digitalWrite(Bobina2, TipoPaso[j][1]);
                digitalWrite(Bobina3, TipoPaso[j][2]);
                digitalWrite(Bobina4, TipoPaso[j][3]);
                delay(demora);
            }else if (Direccion=="++){
                digitalWrite(Bobina1, TipoPaso[j][3]);
                digitalWrite(Bobina2, TipoPaso[j][2]);
                digitalWrite(Bobina3, TipoPaso[j][1]);
                digitalWrite(Bobina4, TipoPaso[j][0]);
                delay(demora);
            }
        }
    }
}

```

```

    }
}
delay(500);
}

```

```

void Detener(){
    digitalWrite(IN1, LOW); // detiene
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
}

```

```

void ReconocerColor(int x, int y){
    String color="Blanco";

    if(true){ // verificación para el blanco
        color = "Blanco";
    }else if (false){ // verificación para el negro
        color = "Negro";
    }
}

```

```

if(color == "Negro"){
    LetraBraile[x][y] = 1;
}else if (color = "Blanco"){
    LetraBraile[x][y] = 0;
}

}

```

```

void Alinear(){
}

```

```

void ReconocerLetra(){

```

```

    // Mandamos las casillas

```

```
}
```

```
void LecturaBraille(){
```

```
    // Alinando el cabezal a la primer coordenada
```

```
    Mover(NumeroDePasos,PasoCompleto, "++",IN1, IN2,IN3,IN4); // Eje y
```

```
    Mover(NumeroDePasos,PasoCompleto, "--",MIN1, MIN2,MIN3,MIN4); // Eje x
```

```
    // Comienza la lectura
```

```
    ReconocerColor(0,0); // Reconocemos la coordenada 0,0
```

```
    Mover(NumeroDePasos,PasoCompleto, "--",IN1, IN2,IN3,IN4); // Eje y
```

```
    ReconocerColor(0,1); // Reconocemos la coordenada 0,1
```

```
    Mover(NumeroDePasos,PasoCompleto, "--",IN1, IN2,IN3,IN4); // Eje y
```

```
    ReconocerColor(0,2); // Reconocemos la coordenada 0,2
```

```
    Mover(NumeroDePasos,PasoCompleto, "++",MIN1, MIN2,MIN3,MIN4); // Eje x
```

```
    ReconocerColor(1,2); // Reconocemos la coordenada 1,2
```

```
    Mover(NumeroDePasos,PasoCompleto, "++",IN1, IN2,IN3,IN4); // Eje y
```

```
    ReconocerColor(1,1); // Reconocemos la coordenada 1,1
```

```
    Mover(NumeroDePasos,PasoCompleto, "++",IN1, IN2,IN3,IN4); // Eje y
```

```
    ReconocerColor(1,0); // Reconocemos la coordenada 1,0
```

```
    ReconocerLetra();
```

```
}
```

paso completo

PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

Sensor:

```
const int s0 = 30;
const int s1 = 31;
const int s2 = 32;
const int s3 = 33;
const int out = 34;
```

```
int rojo = 0;
int verde = 0;
int azul = 0;
```

```
void setup(){
  Serial.begin(9600);
  pinMode(s0,OUTPUT);
  pinMode(s1,OUTPUT);
  pinMode(s2,OUTPUT);
  pinMode(s3,OUTPUT);
  pinMode(out,INPUT);
  digitalWrite(s0,HIGH);
  digitalWrite(s1,HIGH);
}
```

```
void loop(){
  color();
  Serial.print(" ");
  Serial.print(rojo, DEC);
  Serial.print(" ");
  Serial.print(verde, DEC);
  Serial.print(" ");
  Serial.print(azul, DEC);






  if (rojo < 10 && verde < 10 && azul < 10) {
    Serial.println(" Negro");
  }
  else if (rojo < azul && verde > azul && rojo < 35)
  {
    Serial.println(" Rojo");
  }
  else if (azul < rojo && azul < verde && verde < rojo)
  {
    Serial.println(" Azul");
  }




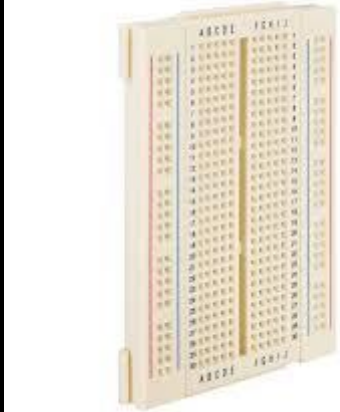

  else if (rojo > verde && azul > verde )
  {
    Serial.println(" Verde");
  }
}
```

```
else{
  Serial.println(" ");
}
delay(900);
}


void color()
{
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  rojo = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s3, HIGH);
  azul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s2, HIGH);
  verde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}
}
```

Equipo utilizado

Equipo	Imagen
Motor stepper	
Fuente de Alimentación	
deslizante 4 rodamientos	
Driveres	
Cables	

<div data-bbox="279 224 715 387" data-label="Text"> <p>Barra deslizante para rodientos</p> </div>	
<div data-bbox="399 510 592 725" data-label="Text"> <p>Componentes flip flop jk y decodificador</p> </div>	
<div data-bbox="448 741 542 779" data-label="Text"> <p>Polea</p> </div>	
<div data-bbox="399 1041 592 1115" data-label="Text"> <p>PROTOBOARD</p> </div>	
<div data-bbox="399 1496 496 1534" data-label="Text"> <p>Llanta</p> </div>	

Paleta de madera	
Faja dentada	
resistencias	

Sensor de color	
-----------------	--

Presupuesto

Andres,Lizz,Engel,Carmen			
Componentes	Precio	Cantidad	Subtotal
Arduino	350	1	350
cable	3	2	6
RESISTENCIA	0.50	4	2
Rodos de 4	85	2	170
Rodos	35	4	140
FUENTE 5V	30	1	30
Faja dentada	20	2	40
motores	2	45	90
Sensor de color	45	4	180
faja	20	1	20
llanta	25	1	25
silicon	20	4	80
barra	50	2	100
drivers	60	2	120
paletas	1	8	8
TOTAL			1361

Conclusiones

En esta práctica pudimos observar varios elementos que son muy importantes en la electrónica digital, algunas de las conclusiones que llegamos en esta fase son las siguientes:

1. **Aplicación Práctica de Teoría:** La implementación de proyectos prácticos como la transmisión serial, el desarrollo de estructuras mecánicas, el control de puertos y la integración de circuitos eléctricos con movimientos mecánicos ha permitido aplicar directamente los conocimientos teóricos adquiridos en clase.
2. **Desarrollo de Habilidades Técnicas:** Has tenido la oportunidad de desarrollar habilidades técnicas en áreas como la electrónica, la programación, la mecánica y el uso de herramientas como Arduino y motores paso a paso. Esto fortalece tu capacidad para resolver problemas complejos en proyectos interdisciplinarios.
3. **Aprendizaje Profundo:** La experiencia en la aplicación de registros, memorias, lógica secuencial y flip-flops para contadores ha proporcionado un entendimiento más profundo de cómo funcionan estos componentes dentro de sistemas más grandes y complejos.
4. **Interrelación de Disciplinas:** La integración de conocimientos de diferentes disciplinas como la electrónica y la mecánica muestra cómo estas áreas se complementan para crear soluciones tecnológicas completas y funcionales.
5. **Fundamentales en contadores y registros:** Son componentes fundamentales en la construcción de contadores y registros, ampliamente utilizados en la electrónica para diversas aplicaciones.
6. **Diseño asistido por computadora:** Las herramientas de diseño asistido por computadora son cruciales para implementar y verificar circuitos lo que ayuda a reducir errores y acelerar el diseño.

Recomendaciones

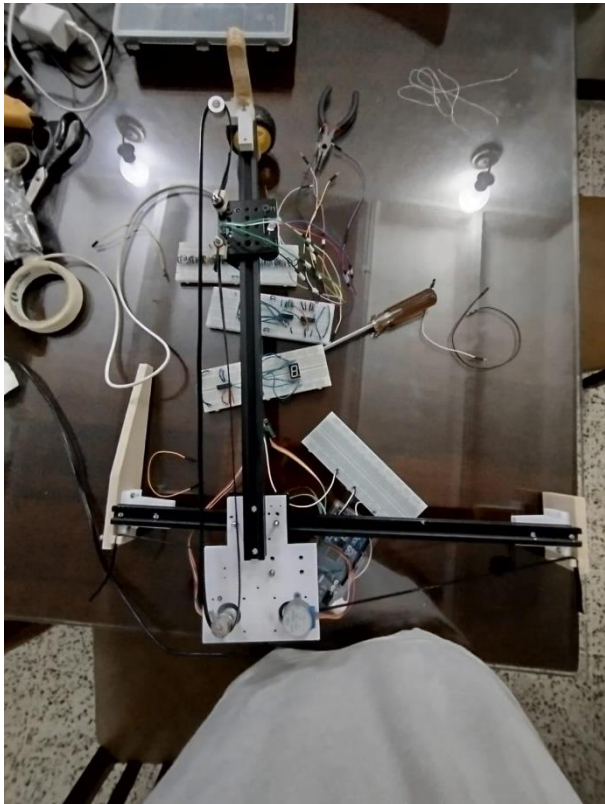
Esquemático eléctrico: Proporciona un diagrama claro del circuito eléctrico que incluya cómo se conectan Arduino, los contadores y otros componentes relevantes.

Diagrama de bloques: Si es necesario, usa un diagrama de bloques para mostrar la interacción funcional entre los diferentes módulos del sistema

Esquemático eléctrico: Proporciona un diagrama claro del circuito eléctrico que incluya cómo se conectan Arduino, los contadores y otros componentes relevantes.

Diagrama de bloques: Si es necesario, usa un diagrama de bloques para mostrar la interacción funcional entre los diferentes módulos del sistema

ANEXOS



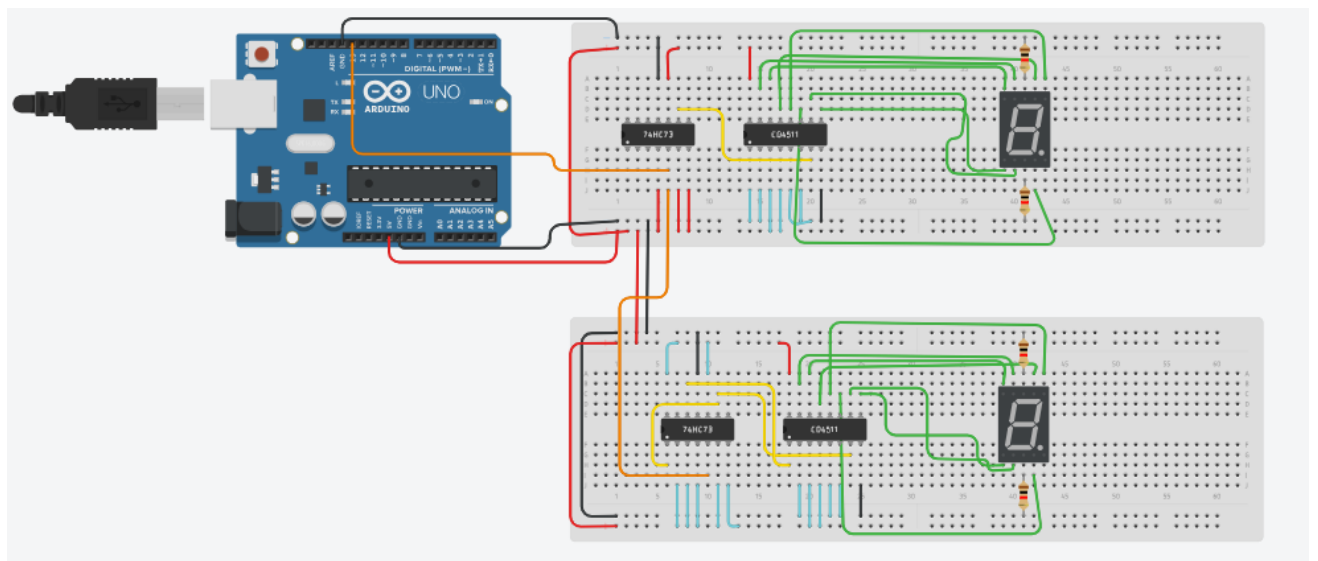


Figura de colores

A •	B •	C ••	D ••	E •	F ••	G ••	H ••	I •	J ••	K •	L ••	M ••
N ••	O ••	P ••	Q ••	R ••	S ••	T ••	U ••	V ••	W ••	X ••	Y ••	Z ••