

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Organización Computacional, Sección A
Ing. Juan Carlos Maeda Juárez
Auxiliar: Javier Alejandro Gutiérrez de León
Vacaciones Del Primer Semestre 2024



PROYECTO FINAL

Lector de Braille

No	Nombre	Carné	CUI
1.	Josseline Griselda Montecinos Hernández	202201534	3010008160101
2.	Evelyn Maricely Balcarcel Rivera	202200110	3149328281501
3.	Pablo Andrés Rodríguez Lima	202201947	3014214630101
4.	Carlos Manuel Lima Y Lima	202201524	3009269950101

Guatemala, 27 de junio del 2024.

ÍNDICE

OBJETIVOS	4
Objetivo General	4
Objetivos Específicos	4
Descripción Del Problema	5
Lógica Del Sistema	6
Funciones Booleanas Y Mapas De Karnaugh	7
Diagramas De Estado	8
Diagramas Del Diseño Del Circuito	9
Código Comentado	11
Equipo Utilizado	28
Diagramas Con Explicación	29
Presupuesto	30
CONCLUSIONES	31
RECOMENDACIONES	32
ANEXOS	33

INTRODUCCIÓN

La educación inclusiva busca garantizar que todas las personas, independientemente de sus capacidades, tengan acceso a recursos educativos de calidad. Entre los retos más significativos se encuentra el desarrollo de tecnologías que apoyen a personas con discapacidades visuales. Este proyecto tiene como objetivo diseñar y construir un prototipo de lector de Braille, una herramienta que no solo ayuda a la verificación de la legibilidad de textos en Braille, sino que también abre nuevas posibilidades para la educación de personas ciegas.

El desarrollo de este lector de Braille se basa en la aplicación de múltiples áreas de conocimiento adquiridas en el curso de Organización Computacional. Estas incluyen el de contadores, la implementación de transmisiones seriales, el control de motores paso a paso, y la gestión de memorias RAM. La integración de estos componentes en un sistema funcional demuestra la capacidad de aplicar teoría a la práctica, resolviendo problemas reales mediante soluciones tecnológicas.

Además de la relevancia técnica, este proyecto tiene un fuerte componente social. Al mejorar la capacidad de leer y verificar textos en Braille, se facilita el acceso a la educación para personas con discapacidad visual. Esto no solo contribuye a su desarrollo académico, sino también a su integración social y autonomía personal.

El lector de Braille utilizará sensores de color para verificar el posicionamiento correcto de las tarjetas y motores paso a paso para mover los sensores a lo largo de los ejes X e Y. Se implementará una memoria RAM dinámica para almacenar temporalmente las letras antes de su procesamiento por una computadora. Esta combinación de mecánica, electrónica y software reflejará un esfuerzo colaborativo y multidisciplinario, dirigido a crear una herramienta que responde a necesidades específicas y urgentes.

OBJETIVOS

Objetivo General

1. Poner en práctica todos los conocimientos adquiridos en el laboratorio.

Objetivos Específicos

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

Descripción Del Problema

La Asociación de Prociegos y Sordos de Guatemala ha creado numerosos libros educativos en Braille que son utilizados por estudiantes desde primaria hasta niveles diversificados. Sin embargo, con el paso del tiempo, muchas de las páginas de estos libros se han deteriorado, dificultando la lectura y comprensión de los textos por parte de los usuarios. La principal preocupación es que las letras en Braille se vuelven ilegibles, lo cual representa un obstáculo significativo para la educación de las personas ciegas.

Para abordar este problema, se nos ha encomendado desarrollar un prototipo que permita verificar la legibilidad de las letras en Braille. Este prototipo deberá ser capaz de identificar y validar la posición y existencia de los puntos que componen cada letra en una tarjeta individual. La correcta identificación de las letras permitirá evaluar el estado de los textos y planificar la creación o restauración de materiales educativos en Braille.

El sistema debe utilizar sensores de color para verificar el posicionamiento de las tarjetas y motores paso a paso (stepper) para mover estos sensores a lo largo de los ejes X e Y. Además, se requiere la implementación de una memoria RAM dinámica que funcione como un heap para almacenar temporalmente hasta 8 letras antes de ser transmitidas a una computadora para su procesamiento. Este proyecto no solo pondrá a prueba nuestros conocimientos técnicos, sino que también contribuirá a la inclusión y accesibilidad educativa de personas con discapacidad visual.

Lógica Del Sistema

Interfaz Gráfica del Usuario

Esta interfaz le permite al usuario visualizar la palabra escaneada a través de la estructura física y poder reiniciar el sistema para una nueva la lectura.

Comunicación con el Arduino

El software establece una comunicación serial con el Arduino Virtual, que a su vez se comunica con el Arduino Físico.

Tras cada Lectura en todas las posiciones de la tarjeta se agrega en un arreglo del código del Arduino la coordenada con un 1 si se encuentra pintada y un 0 si no lo está, de manera que al finalizar la lectura comparar con todo el abecedario cual es la letra que corresponde.

Memoria RAM

El arduino recibe los datos y los almacena de manera temporal en una memoria RAM de 7X8 Almacena el código binario del ASCII de la letra que se le dio lectura.

Recorrido de Lectura

Con los 2 motores stepper se coordina el movimiento para pasar el sensor de color central por las coordenadas de core XY del siguiente orden (0,0) (1,0) (0,1) (1,1) (0,2) (1,2)

Funciones Booleanas Y Mapas De Karnaugh

EJE X

Funciones		Booleanas	
$Q(t)$		$Q(t+1)$	
A	B	A	B
0	0	1	0
0	1	x	x
1	0	x	x
1	1	0	1
$f_A = \bar{A}\bar{B}$		$f_B = AB$	

EJE Y

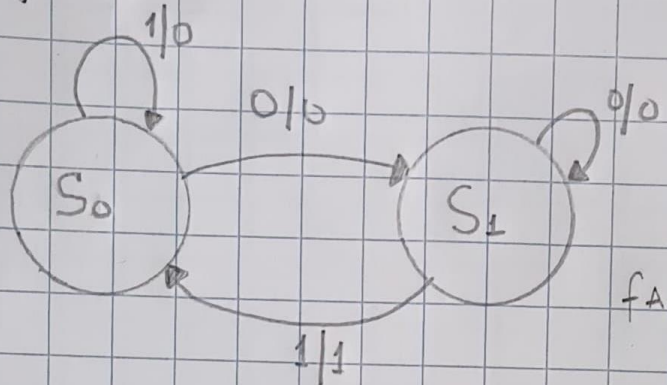
Entradas			Salidas		
A	B	C	X	Y	Z
0	0	0	0	1	0
0	0	1	x	x	x
0	1	0	1	0	0
0	1	1	x	x	x
1	0	0	x	x	x
1	0	1	0	0	1
1	1	0	x	x	x
1	1	1	x	x	x

Mapas de Karnaugh y Funciones Booleanas					
Función X:					
$\begin{matrix} bc \\ A \end{matrix}$	00	01	11	10	
0	0	x	x	1	$f(x) = B$
1	x	0	x	x	
Función Y:					
$\begin{matrix} bc \\ A \end{matrix}$	00	01	11	10	
0	1	x	x	0	$f(y) = \bar{A}\bar{B}$
1	x	0	x	x	
Función Z:					
$\begin{matrix} bc \\ A \end{matrix}$	00	01	11	10	
0	0	x	x	0	$f(z) = A$
1	x	1	x	x	

Diagramas De Estado

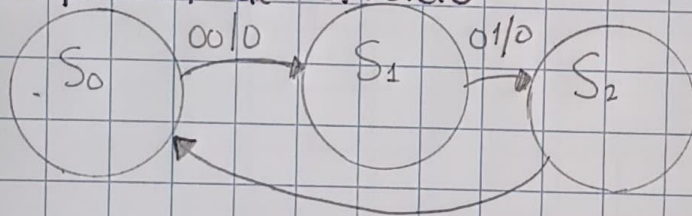
Contador eje "x"

Diagrama de estado

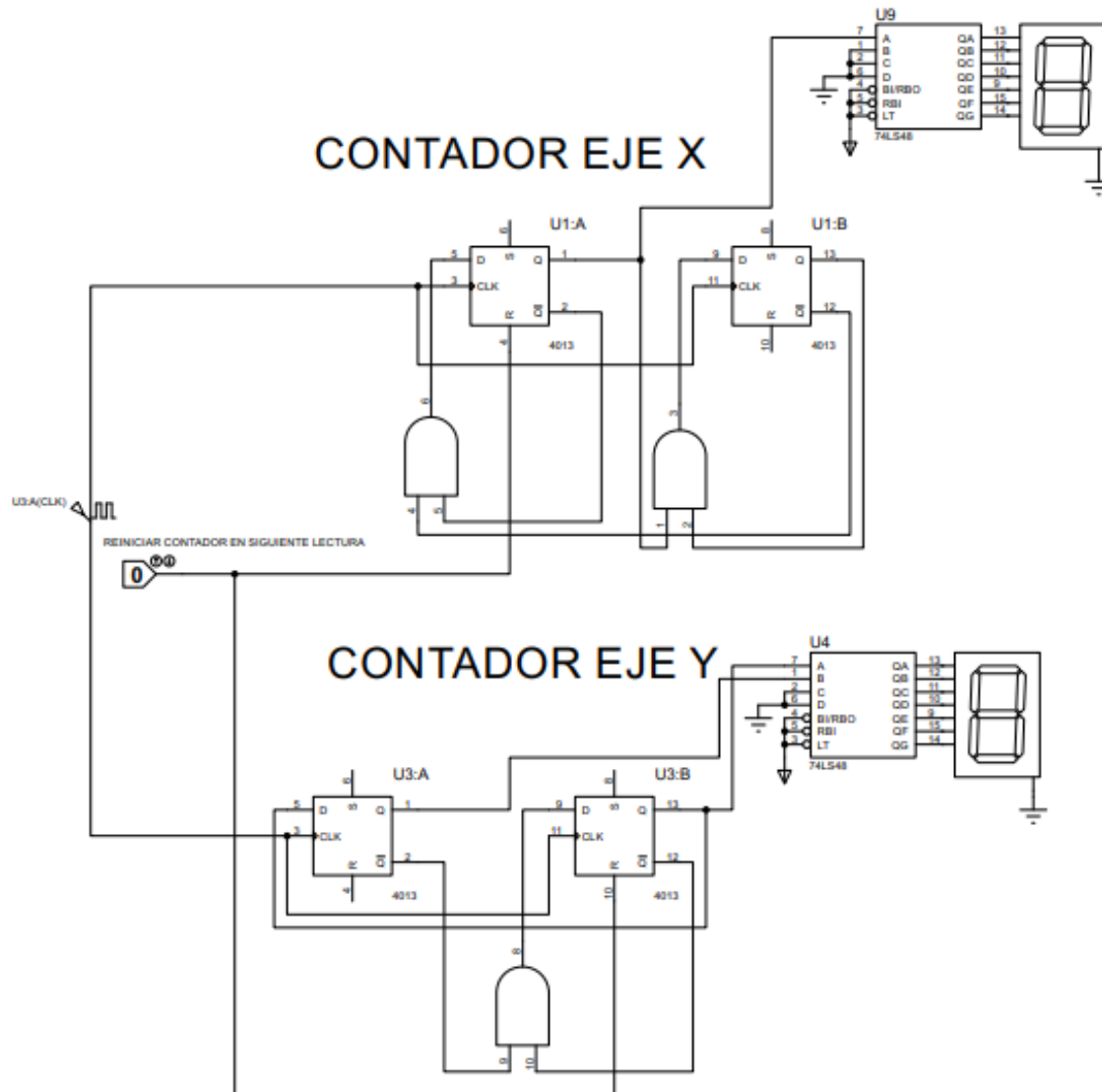
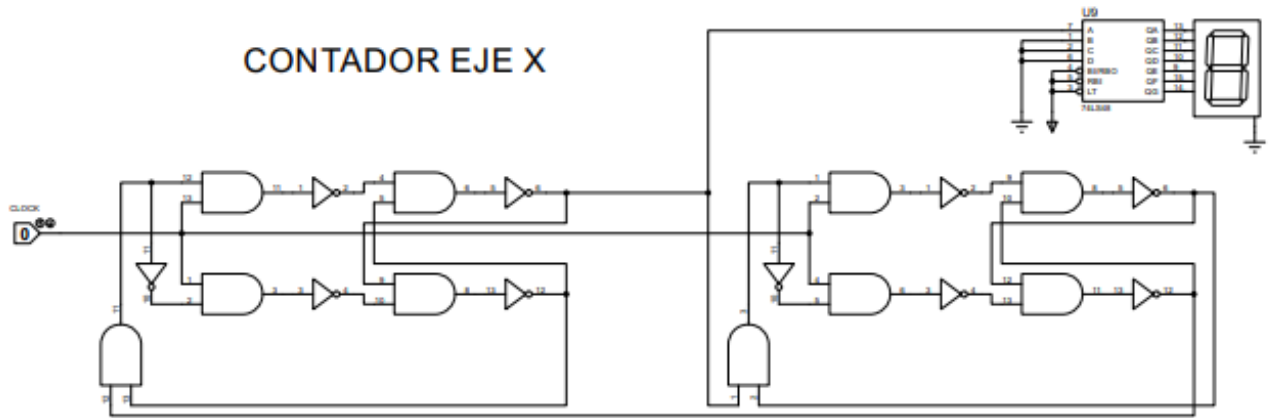


Contador eje "y"

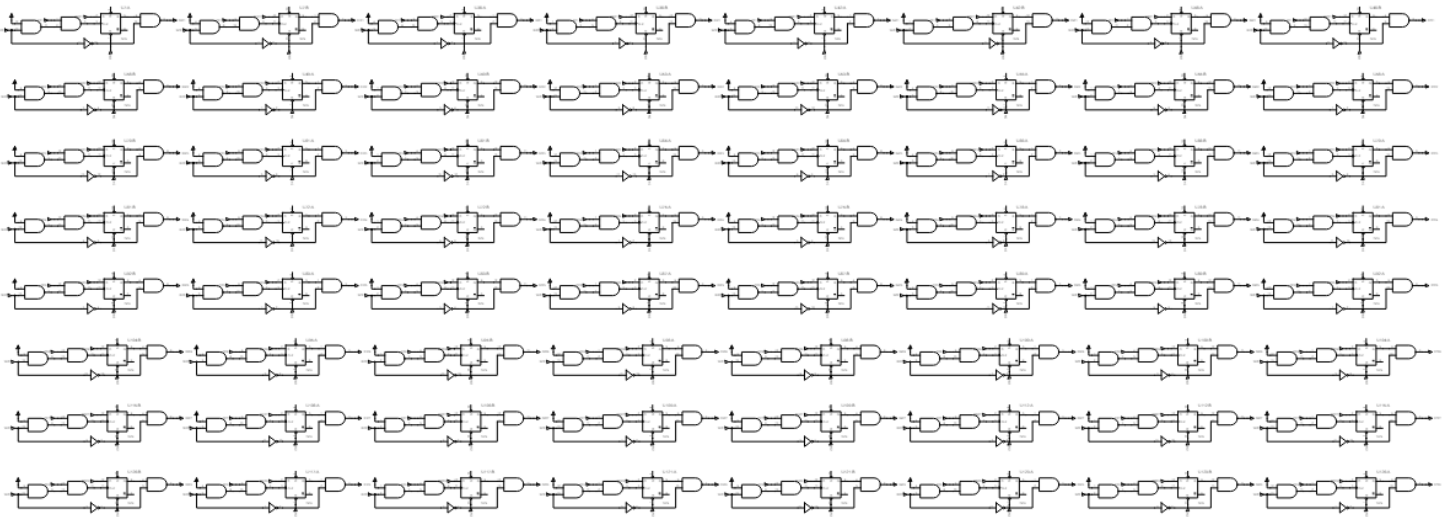
Diagrama de estado



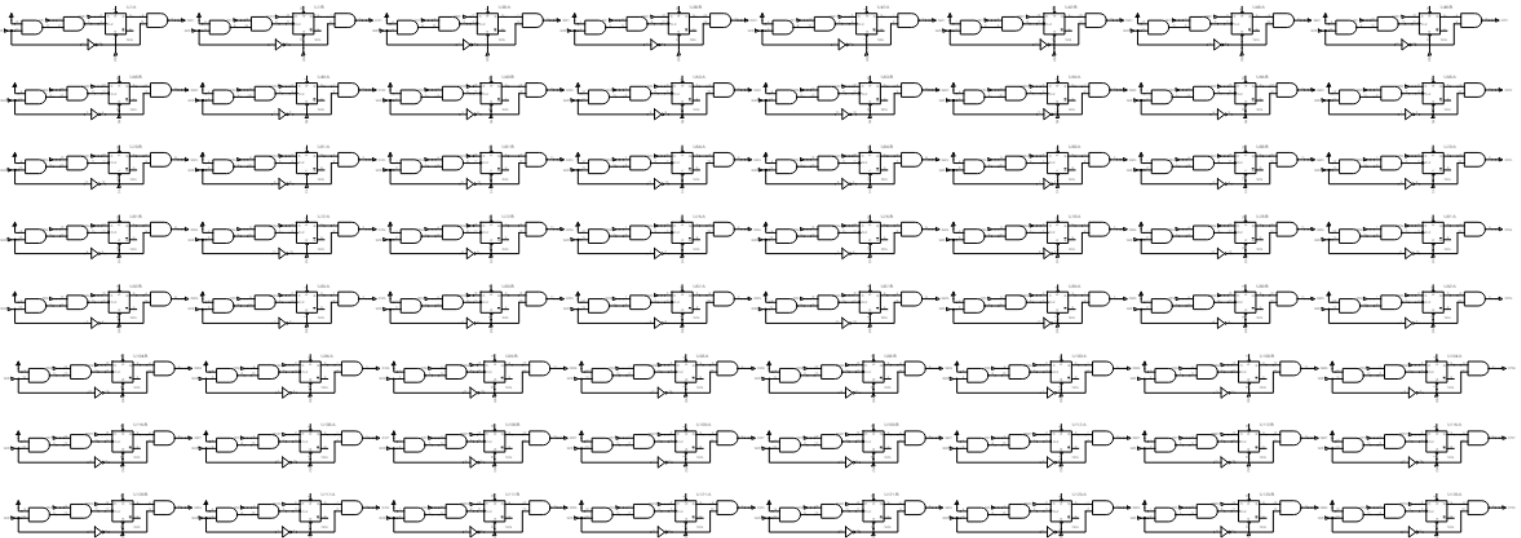
Diagramas Del Diseño Del Circuito



RAM 7 X 8



RAM 7 X 8



Código Comentado

ARDUINO SIMULADO

```
int contador = 0;

void setup() {
  Serial.begin(9600);
  // PINES NUMERO DECIMAL A BINARIO
  for (int pin = 7; pin <= 13; pin++) {
    pinMode(pin, OUTPUT);
  }
  // PINES POSICION DE MEMORIA
  for (int pin = 4; pin <= 6; pin++) {
    pinMode(pin, OUTPUT);
  }
  // PIN PULSO DE RELOJ
  pinMode(3, OUTPUT);

  // PIN LED DE CONFIRMACION
  pinMode(2, OUTPUT);

  // PINES DE ENTRADA
  for (int pin = 22; pin <= 28; pin++) {
    pinMode(pin, INPUT);
  }

  // PIN LED DE CONFIRMACION
  pinMode(29, OUTPUT);
}
```

```

void loop() {
  if (Serial.available() > 0) {
    digitalWrite(29, HIGH);
    char command = Serial.read();
    if (command == 'R') {
      for (int pos = 0; pos < 8; pos++) {
        // Configura los pines de posición de memoria
        for (int i = 0; i < 3; i++) {
          int pin = 6 - i;
          if (pos & (1 << i)) {
            digitalWrite(pin, HIGH);
          } else {
            digitalWrite(pin, LOW);
          }
        }
      }
      delay(1000);

      // Lee el valor binario de los pines de entrada
      int valorBinario = 0;
      for (int i = 0; i < 7; i++) {
        int pin = 22 + i;
        valorBinario |= (digitalRead(pin) << (6 - i)); // El pin 22 es el más significativo
      }

      // Envía el valor como carácter ASCII
      Serial.println(valorBinario);
      delay(1000);
    }
  }
}

```

```

    digitalWrite(29, LOW);
}
} else {
    digitalWrite(2, HIGH);
    // POSICION DE MEMORIA
    for (int i = 0; i < 3; i++) {
        int pin = 6 - i;
        if (contador & (1 << i)) {
            digitalWrite(pin, HIGH);
        } else {
            digitalWrite(pin, LOW);
        }
    }
    contador = (contador + 1) % 8;
    delay(1000);
    // LECTURA DE NUMERO
    String numero = Serial.readStringUntil('\n');
    int valor = numero.toInt();
    delay(1000);
    // TRADUCCION EN BINARIO PARA LA RAM
    for (int i = 0; i < 7; i++) {
        int pin = 13 - i;
        if (valor & (1 << i)) {
            digitalWrite(pin, HIGH);
        } else {
            digitalWrite(pin, LOW);
        }
    }
}

```

```

delay(1000);
// ESCRITURA DE LECTURA
digitalWrite(3, HIGH);
delay(1000);
digitalWrite(3, LOW);

// REINICIO DE PINES
delay(1000);
for (int pin = 7; pin <= 13; pin++) {
    digitalWrite(pin, LOW);
}
delay(1000);
for (int pin = 4; pin <= 6; pin++) {
    digitalWrite(pin, LOW);
}
delay(1000);
digitalWrite(2, LOW);
}
}
}

```

ARDUINO FÍSICO

```
#include <Stepper.h>
```

```
//VUELTA
```

```
int vuelta = 2048;
```

```
//LECTURA BRAILLE
```

```

const int puntosBraille[6] = {1, 4, 2, 5, 3, 6};
const int numPuntos = 6;
int lecturaPuntos[numPuntos];

//PINES MOTORES
Stepper motorX(2048, 13, 11, 12, 10);
Stepper motorY(2048, 9, 7, 8, 6);

//SENSOR CENTRAL
const int S0_C = 49;
const int S1_C = 50;
const int S2_C = 51;
const int S3_C = 52;
const int OUT_C = 53;

int rojo_central;
int verde_central;
int azul_central;

void setup() {
  Serial.begin(9600);

  // Inicializar el arreglo de lectura de puntos en braille
  for (int i = 0; i < numPuntos; ++i) {
    lecturaPuntos[i] = 0;
  }
}

```

```

// Configuración de la velocidad de los motores
motorX.setSpeed(10);
motorY.setSpeed(10);

// Configuración del pin de entrada
pinMode(5, INPUT);

//PIN SENSOR CENTRAL
pinMode(S0_C, OUTPUT);
pinMode(S1_C, OUTPUT);
pinMode(S2_C, OUTPUT);
pinMode(S3_C, OUTPUT);
pinMode(OUT_C, INPUT);
digitalWrite(S0_C, HIGH);
digitalWrite(S1_C, HIGH);
}

void loop() {

    int estado = digitalRead(5);

    if (estado == HIGH) {
        Serial.println("INICIA ESCANEEO");

        delay(2000);

        rojo_central = get_rojo_central();
        delay(100);
    }
}

```



```
verde_central = get_verde_central();  
delay(100);  
azul_central = get_azul_central();  
delay(100);
```

```
Serial.print("Lectura 1 - Rojo: ");  
Serial.print(rojo_central);  
Serial.print(", Verde: ");  
Serial.print(verde_central);  
Serial.print(", Azul: ");  
Serial.println(azul_central);
```

```
if (rojo_central < 10 && verde_central < 10 && azul_central < 10) {  
    lecturaPuntos[0] = 0;  
} else if (rojo_central > 15 && verde_central > 20 && azul_central > 20) {  
    lecturaPuntos[0] = 1;  
}  
delay(2000);
```

```
//+Y  
for (int i = 0; i < vuelta/2; i++) {  
    motorX.step(-1);  
    motorY.step(1);  
}
```

```
delay(2000);
```

```
rojo_central = get_rojo_central();
```

```
delay(100);
verde_central = get_verde_central();
delay(100);
azul_central = get_azul_central();
delay(100);
```

```
Serial.print("Lectura 2 - Rojo: ");
Serial.print(rojo_central);
Serial.print(", Verde: ");
Serial.print(verde_central);
Serial.print(", Azul: ");
Serial.println(azul_central);
```

```
if (rojo_central < 10 && verde_central < 10 && azul_central < 10) {
    lecturaPuntos[1] = 0;
} else if (rojo_central > 15 && verde_central > 20 && azul_central > 20) {
    lecturaPuntos[1] = 1;
}
delay(2000);
```

```
//-Y
for (int i = 0; i < vuelta/2; i++) {
    motorX.step(1);
    motorY.step(-1);
}
delay(2000);
//-X
for (int i = 0; i < vuelta/2; i++) {
```

```

    motorX.step(-1);
    motorY.step(-1);
}

delay(2000);

rojo_central = get_rojo_central();
delay(100);
verde_central = get_verde_central();
delay(100);
azul_central = get_azul_central();
delay(100);

Serial.print("Lectura 3 - Rojo: ");
Serial.print(rojo_central);
Serial.print(", Verde: ");
Serial.print(verde_central);
Serial.print(", Azul: ");
Serial.println(azul_central);

if (rojo_central < 10 && verde_central < 10 && azul_central < 10) {
    lecturaPuntos[2] = 0;
} else if (rojo_central > 15 && verde_central > 20 && azul_central > 20) {
    lecturaPuntos[2] = 1;
}
delay(2000);

//+Y

```

```

for (int i = 0; i < vuelta/2; i++) {
    motorX.step(-1);
    motorY.step(1);
}

delay(2000);
rojo_central = get_rojo_central();
delay(100);
verde_central = get_verde_central();
delay(100);
azul_central = get_azul_central();
delay(100);

Serial.print("Lectura 4 - Rojo: ");
Serial.print(rojo_central);
Serial.print(", Verde: ");
Serial.print(verde_central);
Serial.print(", Azul: ");
Serial.println(azul_central);
if (rojo_central < 10 && verde_central < 10 && azul_central < 10) {
    lecturaPuntos[3] = 0;
} else if (rojo_central > 15 && verde_central > 20 && azul_central > 20) {
    lecturaPuntos[3] = 1;
}
delay(2000);

//-Y
for (int i = 0; i < vuelta/2; i++) {

```

```

    motorX.step(1);
    motorY.step(-1);
}
delay(2000);
//-X
for (int i = 0; i < vuelta/2; i++) {
    motorX.step(-1);
    motorY.step(-1);
}

delay(2000);
rojo_central = get_rojo_central();
delay(100);
verde_central = get_verde_central();
delay(100);
azul_central = get_azul_central();
delay(100);

Serial.print("Lectura 5 - Rojo: ");
Serial.print(rojo_central);
Serial.print(", Verde: ");
Serial.print(verde_central);
Serial.print(", Azul: ");
Serial.println(azul_central);
if (rojo_central < 10 && verde_central < 10 && azul_central < 10) {
    lecturaPuntos[4] = 0;
} else if (rojo_central > 15 && verde_central > 20 && azul_central > 20) {
    lecturaPuntos[4] = 1;
}

```

```

}
delay(2000);

//+Y
for (int i = 0; i < vuelta/2; i++) {
    motorX.step(-1);
    motorY.step(1);
}

delay(2000);
rojo_central = get_rojo_central();
delay(100);
verde_central = get_verde_central();
delay(100);
azul_central = get_azul_central();
delay(100);

Serial.print("Lectura 6 - Rojo: ");
Serial.print(rojo_central);
Serial.print(", Verde: ");
Serial.print(verde_central);
Serial.print(", Azul: ");
Serial.println(azul_central);
if (rojo_central < 10 && verde_central < 10 && azul_central < 10) {
    lecturaPuntos[5] = 0;
} else if (rojo_central > 15 && verde_central > 20 && azul_central > 20) {
    lecturaPuntos[5] = 1;
}

```

```
delay(2000);
```

```
//-Y
```

```
for (int i = 0; i < vuelta/2; i++) {
```

```
    motorX.step(1);
```

```
    motorY.step(-1);
```

```
}
```

```
delay(2000);
```

```
//+X
```

```
for (int i = 0; i < vuelta; i++) {
```

```
    motorX.step(1);
```

```
    motorY.step(1);
```

```
}
```

```
delay(2000);
```

```
int letra = determinarLetra(lecturaPuntos);
```

```
Serial.println("-----");
```

```
Serial.println(letra);
```

```
Serial.println("&&&&&&&&&&");
```

```
}
```

```
}
```

```
//SENSOR CENTRAL
```

```
int get_rojo_central() {
```

```
    digitalWrite(S2_C, LOW);
```

```
    digitalWrite(S3_C, LOW);
```

```
    return pulseIn(OUT_C, digitalRead(OUT_C) == HIGH ? LOW : HIGH);
```

```
}
```

```

int get_verde_central() {
    digitalWrite(S2_C, HIGH);
    digitalWrite(S3_C, HIGH);
    return pulseIn(OUT_C, digitalRead(OUT_C) == HIGH ? LOW : HIGH);
}

```

```

int get_azul_central() {
    digitalWrite(S2_C, LOW);
    digitalWrite(S3_C, HIGH);
    return pulseIn(OUT_C, digitalRead(OUT_C) == HIGH ? LOW : HIGH);
}

```

```

int determinarLetra(int puntos[]) {
    if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 0 && puntos[3] == 0 && puntos[4] ==
0 && puntos[5] == 0) {
        return 97; // 'a' en ASCII
    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 0 && puntos[3] == 1 && puntos[4]
== 0 && puntos[5] == 0) {
        return 98; // 'b' en ASCII
    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 0 && puntos[3] == 0 && puntos[4]
== 0 && puntos[5] == 0) {
        return 99; // 'c' en ASCII
    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 0 && puntos[3] == 1 && puntos[4]
== 0 && puntos[5] == 0) {
        return 100; // 'd' en ASCII
    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 0 && puntos[3] == 0 && puntos[4]
== 1 && puntos[5] == 0) {
        return 101; // 'e' en ASCII
    }
}

```



```

    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 0 && puntos[3] == 0 && puntos[4]
== 1 && puntos[5] == 0) {
        return 102; // 'f' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 0 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 0) {
        return 103; // 'g' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 0 && puntos[3] == 0 && puntos[4]
== 1 && puntos[5] == 1) {
        return 104; // 'h' en ASCII

    } else if (puntos[0] == 0 && puntos[1] == 1 && puntos[2] == 0 && puntos[3] == 1 && puntos[4]
== 0 && puntos[5] == 0) {
        return 105; // 'i' en ASCII

    } else if (puntos[0] == 0 && puntos[1] == 1 && puntos[2] == 0 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 0) {
        return 106; // 'j' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 0 && puntos[4]
== 0 && puntos[5] == 0) {
        return 107; // 'k' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 1 && puntos[3] == 0 && puntos[4]
== 0 && puntos[5] == 0) {
        return 108; // 'l' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 0 && puntos[5] == 0) {
        return 109; // 'm' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 0) {
        return 110; // 'n' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 0 && puntos[4]
== 1 && puntos[5] == 1) {
        return 111; // 'o' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 0 && puntos[5] == 0) {

```

```

    return 112; // 'p' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 0) {

        return 113; // 'q' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 1) {

        return 114; // 'r' en ASCII

    } else if (puntos[0] == 0 && puntos[1] == 1 && puntos[2] == 1 && puntos[3] == 0 && puntos[4]
== 0 && puntos[5] == 0) {

        return 115; // 's' en ASCII

    } else if (puntos[0] == 0 && puntos[1] == 1 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 0 && puntos[5] == 0) {

        return 116; // 't' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 0 && puntos[4]
== 0 && puntos[5] == 1) {

        return 117; // 'u' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 1 && puntos[3] == 0 && puntos[4]
== 1 && puntos[5] == 0) {

        return 118; // 'v' en ASCII

    } else if (puntos[0] == 0 && puntos[1] == 1 && puntos[2] == 0 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 0) {

        return 119; // 'w' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 0 && puntos[5] == 1) {

        return 120; // 'x' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 1) {

        return 121; // 'y' en ASCII

    } else if (puntos[0] == 1 && puntos[1] == 0 && puntos[2] == 0 && puntos[3] == 1 && puntos[4]
== 1 && puntos[5] == 1) {

        return 122; // 'z' en ASCII

```

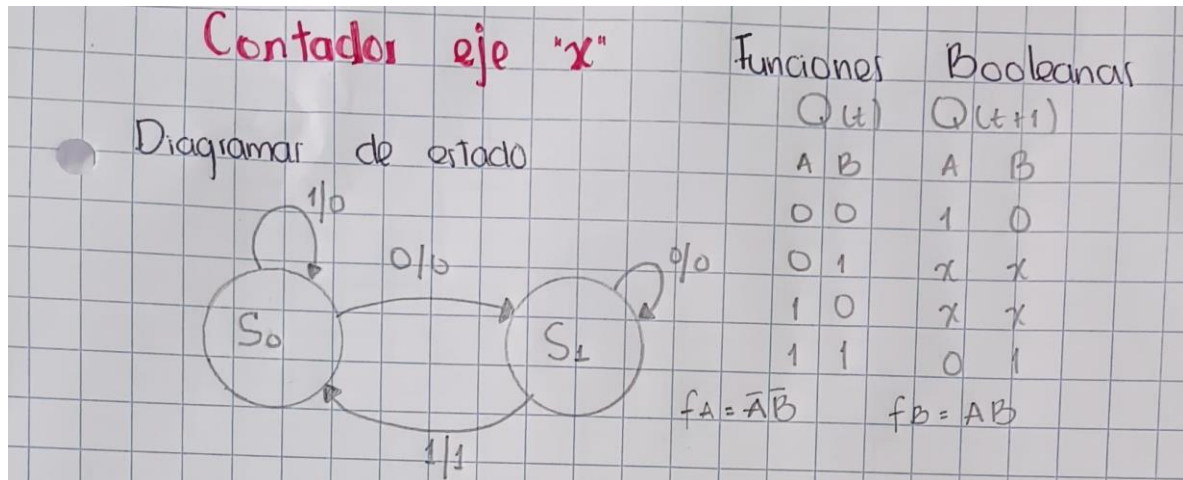
```
    } else if (puntos[0] == 1 && puntos[1] == 1 && puntos[2] == 1 && puntos[3] == 1 && puntos[4]  
== 1 && puntos[5] == 1) {  
        return 999999999;  
    } else if (puntos[0] == 0 && puntos[1] == 0 && puntos[2] == 0 && puntos[3] == 0 && puntos[4]  
== 0 && puntos[5] == 0) {  
        return 85858585;  
    }  
}
```

Equipo Utilizado

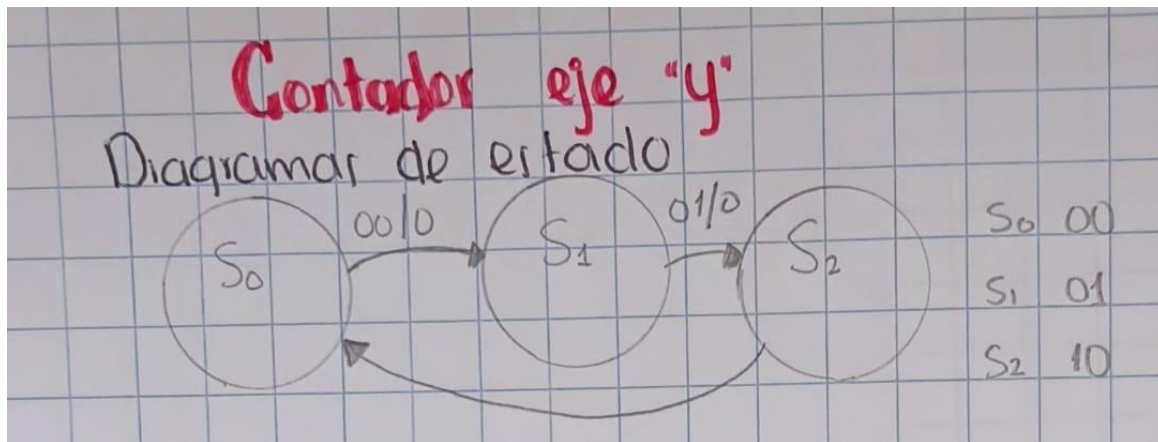
Equipo
Protoboard
Pulsador
Cable para protoboard
Display 7 Segmentos
Compuertas AND
Compuertas NOT
Decoder (74LS48)
Resistencias
Sensores de Color
Jumpers HxH
Microcontrolador (Arduino)
Motores Stepper
Driver para Motores
Correa dentada
Multímetro

Diagramas Con Explicación

Contador del Eje X con la secuencia de 0-1 como coordenadas necesarias



Contador del Eje X con la secuencia de 0-2 como coordenadas necesarias



Presupuesto

- Gastos

Componente	Cantidad	Precio Unitario	Total
Metro de Alambre para protoboard	30	Q 2.5	Q 75
Pulsador	1	Q 1.25	Q 1.25
Display 7 Segmentos Cátodo Común	2	Q 6	Q 12
Componte 74LS04 (NOT)	5	Q 5	Q 25
Componte 74LS08 (AND)	5	Q 5	Q 25
Componte 74LS48 (Decoder)	2	Q 15	Q.30
Motor Stepper	2	Q. 29	Q.58
Driver para Motor	2	Q. 18	Q.36
Sensor de Color	5	Q. 79	Q.395
Engranaje Para motor	2	Q.30	Q.60
Material Estructura	-	-	Q.150
Metro Correa Dentada	2	Q 19.00	Q 38.00

Total: Q 905.25

- Aportación Económica

Nombre	Carné	Aportación Económica
Josseline Montecinos	202201534	Q. 226.50
Pablo Rodríguez	202201947	Q. 226.50
Evelyn Balcarcel	202200110	Q. 226.50
Carlos Lima	202201524	Q. 226.50
	Total:	Q 906.00

CONCLUSIONES

El desarrollo del lector de Braille utilizando una estructura de máquina CNC o impresora 3D con un diseño Core XY ha sido un proyecto exitoso y enriquecedor, que ha permitido aplicar una variedad de conocimientos y habilidades adquiridos a lo largo del curso. La implementación de sensores de color y motores paso a paso ha demostrado ser fundamental para lograr la precisión necesaria en la lectura de las tarjetas de Braille. Estos componentes permitieron verificar el posicionamiento correcto de las tarjetas y detectar con exactitud los puntos que componen cada letra en Braille.

La utilización de una estructura de máquina CNC o impresora 3D proporcionó una base robusta y versátil para el proyecto. Esta estructura facilitó la implementación del diseño Core XY, permitiendo un movimiento suave y preciso del sensor de color a través de las tarjetas. La capacidad de controlar los movimientos de los ejes X e Y con motores stepper demostró ser esencial para el éxito del proyecto, proporcionando la precisión y el control necesarios para la lectura efectiva de las letras en Braille.

El proyecto también incluyó la implementación de una memoria RAM dinámica para almacenar temporalmente hasta ocho letras antes de ser transmitidas a una computadora para su procesamiento. La gestión de la memoria y la comunicación serial entre el Arduino físico y otro Arduino en Proteus fueron cruciales para la correcta transferencia y almacenamiento de datos. La simulación en Proteus permitió diseñar y probar el circuito de manera eficiente antes de su implementación física, asegurando que todos los componentes funcionaran correctamente en conjunto.

RECOMENDACIONES

- Es recomendable explorar la posibilidad de utilizar diferentes tipos de sensores o mejorar la calibración de los sensores de color actuales para aumentar la precisión en la detección de los puntos en las tarjetas de Braille. La implementación de sensores con mayor resolución o sensibilidad podría mejorar la exactitud y reducir los errores en la lectura.
- En lugar de utilizar los cuatro sensores de color en las esquinas para verificar el posicionamiento de las tarjetas, se podría considerar el uso de un sistema de posicionamiento más eficiente. Esto podría incluir el uso de un solo sensor de alta precisión combinado con algoritmos de detección de bordes y posicionamiento, lo que simplificaría el diseño y reduciría costos.
- Aumentar la capacidad de la memoria RAM dinámica para almacenar más de ocho letras podría ser beneficioso. Esto permitiría procesar bloques de texto más grandes antes de la transmisión a la computadora, mejorando la eficiencia del sistema.

ANEXOS

