

# Parcial 1

Informe

**Jhonny Alejandro Ortiz Osorio**

**C.C: 1001015092**

**Juan José Florez Argáez**

**C.C: 1001765286**

Departamento de Ingeniería Electrónica y

Telecomunicaciones

Universidad de Antioquia

Medellín

Marzo de 2021

# Índice

<b>1. Sección introductoria</b>	<b>2</b>
<b>2. Sección de contenido</b>	<b>2</b>
2.1. Análisis del problema . . . . .	2
2.2. Esquema de trabajo . . . . .	2
2.3. Pruebas de desarrollo . . . . .	2
2.4. Problemas encontrados . . . . .	2
2.5. Evolución del algoritmo . . . . .	2
2.6. consideraciones . . . . .	3
2.7. Algoritmo implementado . . . . .	3
<b>3. Inclusión de imágenes</b>	<b>4</b>

## 1. Sección introductoria

Pondremos a prueba nuestras destrezas en análisis de problemas, manejo del lenguaje C++ y la plataforma arduino a través de un gran reto que integra todos los conocimientos enseñados hasta la fecha, el cual tendrá unos requisitos que harán aun mas interesante el desafío.

## 2. Sección de contenido

Esta sección es para agregar toda la información correspondiente con código, citas, etc.

### 2.1. Análisis del problema

Principalmente debemos entender el funcionamiento del integrado 74HC595. Luego el diseño y planificación de todas las conexiones necesarias para lograr un circuito estable, eficiente y por ultimo el diseño y desarrollo del software que controlará nuestra matriz de leds.

### 2.2. Esquema de trabajo

- Estudiar el integrado 74HC595.
- Diseño del circuito.
- Diseño inicial (Idea) del software.
- Diseño de las funciones necesarias.
- 

### 2.3. Pruebas de desarrollo

### 2.4. Problemas encontrados

### 2.5. Evolución del algoritmo

- Función verificación

```
int SER = 2;
int RCLK = 3;
int SRCLK = 4;

void Verificacion()
{
    short int matriz_LED[8][8] = { {1,0,0,0,0,0,0,0}, {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0}, {0,0,0,0,0,0,0,0}, {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0}, {0,0,0,0,0,0,0,0}, {0,0,0,0,0,0,0,0} },
```

```

{0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0}};
for(int f = 0; f < 8; f++){
    for(int c = 0; c < 8; c++){
        digitalWrite(SER, matriz_LED[f][c]);
        digitalWrite(SRCLK, 0);
        digitalWrite(SRCLK, 1);
        digitalWrite(SRCLK, 0);
        digitalWrite(RCLK, 0);
        digitalWrite(RCLK, 1);
        digitalWrite(RCLK, 0);
        delay(50);
    }
}
for(int f = 0; f < 8; f++){
    for(int c = 0; c < 8; c++){
        matriz_LED[f][c]=1;
    }
}
for(int f = 0; f < 8; f++){
    for(int c = 0; c < 8; c++){
        digitalWrite(SER, matriz_LED[f][c]);
        digitalWrite(SRCLK, 0);
        digitalWrite(SRCLK, 1);
        digitalWrite(SRCLK, 0);
        digitalWrite(RCLK, 0);
        digitalWrite(RCLK, 1);
        digitalWrite(RCLK, 0);
    }
}
}
}

```

- Función Imagen.

## 2.6. consideraciones

- Limitar la cantidad de secuencia de patrones ingresados por el usuario.

## 2.7. Algoritmo implementado

A continuación, se presenta el código 2.7, que nos permite incluir en el informe partes de programa que requieran una explicación adicional.

```

// Programa desarrollado , compilado y ejecutado en https://www.onlinegdb.com
#include <iostream>

/*

```

```

    * Esto es un comentario de varias lineas
    */

// Comentario de una sola linea

#define N 10

using namespace std;

int main()
{
    for( int i = 0 ; i < N ; i++ ){
        if( !(i % 2) )
            cout << "El valor de i es -> " << i << endl;
    }

    return 0;
}

//Resultado programa

/*
El valor de i es -> 0
El valor de i es -> 2
El valor de i es -> 4
El valor de i es -> 6
El valor de i es -> 8
*/

```

En la sección 3, se presentará como añadir ilustraciones al texto.

### 3. Inclusión de imágenes

Básicamente utilizaremos solo 4 componentes para esta actividad.

1. En la Figura (1), se presenta el integrado 74HC595.
2. En la Figura (2), se presenta el Arduino UNO R3.
3. En la Figura (3), se presenta el LED.



Figura 1: integrado 74HC595

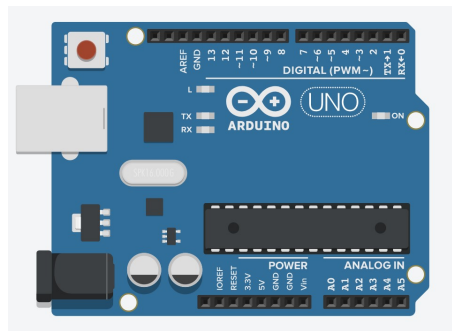


Figura 2: ARDUINO UNO R3

4. En la Figura (4), se presenta la resistencia.
5. En la Figura (5), se presenta la conexión de los objetos a la potencia y la tierra.

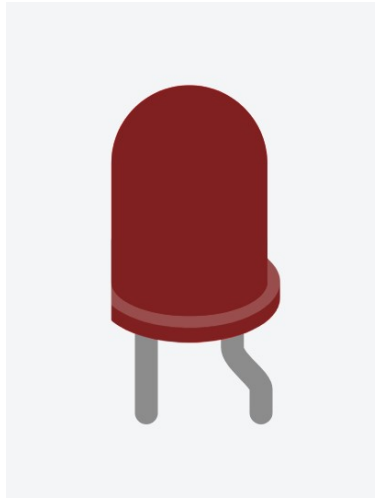


Figura 3: LED

En la Figura (6), representa la conexión de los pines de Arduino a la entrada y salida de reloj de los integrados.

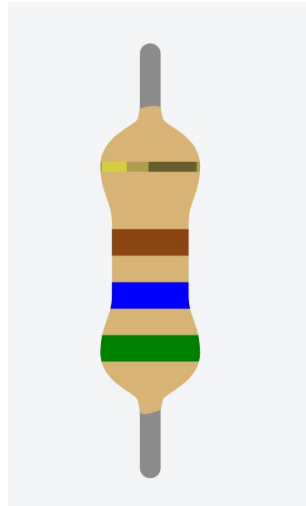


Figura 4: Resistencia

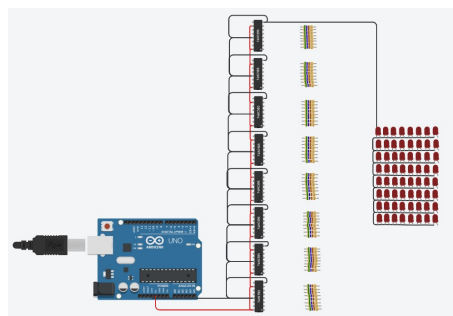


Figura 5: Conexión de los objetos a la potencia y la tierra

En la Figura (7), representa la conexión de los pines de Arduino a la entrada y salida de datos de los integrados.



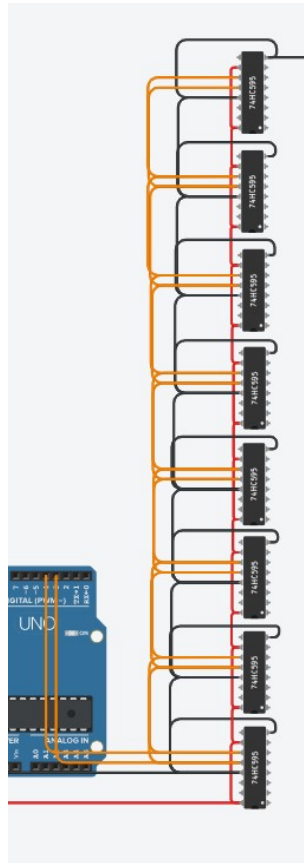


Figura 6: Conexion de los pines de reloj de los integrados

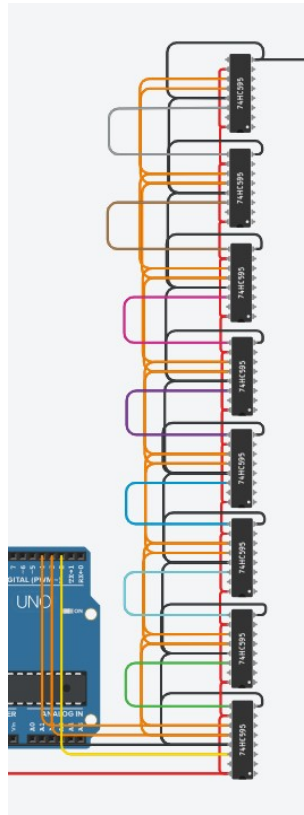


Figura 7: Conexion de los pines de reloj de los integrados