



**Antonio Carlos Merlano Ricardo**  
**1010152199**

**Informática 2**

**Matrix 8x8 led**

**UDEA**  
**25/09/2023**

**Ingenieria telecomunicaciones**

## 1. Analisis del problema

Para enfrentar el problema planteado en cual debemos crear una matriz de leds en los cuales debemos encenderlos y apagarlos para representar patrones ingresados por el usuario proponemos el uso de 8 circuitos integrados 74HC595 los cuales conectaremos haciendo un puente entre los pines del mismo tipo y así llevarlos al los pines digitales del arduino.

Tenemos un circuito montado en la plataforma Tinkercard, donde estamos implementado por medio de un Arduino Uno R3, una protoboard, ocho circuitos integrados, 64 cables amarillos y demás cables para realizar la comunicación de datos entre el arduino, la placa de pruebas, los chips y los leds. Mediante los pines del Arduino, SER=5, reloj-des=3, reloj-salida=4 y mediante la variable opciones =-1, este último es una variable para guardar el número que pide el Serial.available() y guardarlo con Serial.parseInt(), para llevar a cabo una presentación de lo que el proyecto puede mostrarle al usuario. Nuestro proyecto, se compone de las funciones void: setup, loop, inicialización, apagar, patrón, ingreso, grupo8, orden, guardador, verificación, imagen, publik.

La función loop() desprende la información del menú por consola con tres opciones; la primera corresponde presionando la tecla 1 para invocar la función verificación( ) y si al presionar la tecla uno sale el mensaje “todos los leds función correctamente , por favor presione cualquier tecla alfanumérica para volver al menú principal”, por lo tanto se puede proseguir con el funcionamiento del proyecto, este menú por consola también ofrece la opción 2, la cual sirve para llamar la función imagen() y la tercera opción con la tecla 3 ofrecerá al ejecutar la funcion publik(). En caso contrario, cuando no se presionó la tecla indicada en el menú propuesto de la consola se imprimirá también por consola “el dato ingresado no se contempla en el menú” lo quiere decir que se ingresó un valor diferente a 1,2 o 3, esta captura de datos se hace por medio del dato creado opciones=Serial.parseInt();.

Inmediatamente después de la función loop() viene la función inicializacion() la cual al ejecutarse inicializara los “relojes” reloj-des y reloj-salida, después de esta función esta la función apagar() que se encargara de apagar los leds del led 1 al led 64 mediante un ciclo for y con la ayuda de la instrucción digitalWrite (SER,O), para escribir el valor lógico digital en pin de salida de nuestro Arduino Uno R3, dentro de este ciclo también se llamara la mencionada función inicialización().

La función patron() tiene como parámetros int guardado[][] e int num-arre. Esta función ofrece al usuario la posibilidad de que escoja el número de leds que se van a prender por fila, y también ingresar el número que representan los leds prendidos. Esta función guardara en la variable num el número en las posiciones en que se va encender el led, siendo las posiciones desde 1 hasta ocho. Por ejemplo, si se ingresa 1356, estará pidiendo encender los leds 1 3 5 86. La función ingreso() mediante su ejecución guardará en una arreglo los números d'ígitos excluyendo el cero y el nueve. Esto para para mostrar los leds encendidos en esas posiciones excluyendo los mencionados, es decir, entre uno y ocho. La siguiente función a tener en cuenta es grupo8() se encarga de recorrer las filas

de la matriz de leds y pide una cadena de enteros, dentro de esta función que llama la función `inicializacion()` para encender o apagar los leds según el carácter requerido por el usuario. La función `orden()` permite guardar en el arreglo `fin[]` los "1" las posiciones que ingreso el usuario por la consola. Mientras que la función `guardador()` va a guardar los arreglos de arreglos, que posteriormente guardara los valores de encendido y apagado de los led. La función `verificación()` prende todos los leds esto es solamente para verificar que cada uno de los leds enciendan o funcionen correctamente. La función `imagen()` juega un papel muy importante en la ejecución del circuito y del código, ya que ésta permite que el usuario interactúe con la consola; en una primera opción podrá el usuario ingresar los números de las posiciones en que quiera que se encienda los leds, de segunda opción podrá ingresar las letras que formaran el encendido de los leds, mostrando detalladamente los números que debe presionar para que se encienda los leds y forme una letra según el número que le haya sugerido el programa por la consola. La función `publik()` permitirá al usuario ingresar el número de patrones que el usuario desea mostrar, también permitirá al usuario ingresar los milisegundos entre cada visualización de patrones, el usuario también deberá ingresar por consola el tiempo de visualización entre cada patrón. Dentro de esta función hay dos for anidados, el primero consta de tres for el cual el primera recorrerá el número de patrones y los guardara; el segundo for recorrerá las filas de la matriz para guardar las secuencias, mientras que el tercer for es para recorrer los leds que no se van a encender. El ultimo for anidado de esta función contiene cuatro for que ayudaran a llamar la función `grupo8()` cuyo parámetro es el arreglo 2x2 secuencia; este último for anidado tiene un delay cuyo argumento es la variable tiempo, que el usuario habrá ingresado por consola en esta misma función.

Por ultimo tenemos la función `secuenciadepatron()` que fue implementada aparte para poder implementar los patrones del los led mostrara los 4 patrones requeridos donde se encargara de hacer que estos aparezcan y desaparezcan cada lo patrones va acompañados de unos otras dos funciones que son que se encargan de apagar y encender led en cada cierto tiempo.

## 2. problemas en el desarrollo

- Lograr la conexión de los 8 circuitos integrados 74HC595 al arduino para controlar los leds.
- Conectar la matriz al circuito integrado 74HC595 de tal forma que podamos controlar el encendido de todos los leds.
- Entender cómo se manipulan los leds por medio del circuito integrado 74HC595 y el registro de desplazamiento.
- Buscar las funciones más óptimas en c++ la escritura y estructura del código.
- Abuso de la memoria disponible por la plataforma de Tindercard, lo cual llevaba a una lentitud exagerada del programa.

- Falta de conocimiento de las funciones básicas de Arduino como la captura de datos y manipulación de punteros, por falta de conocimiento procesos tardíos de ejecución del código.
- Plataforma Tindercard no respondía,

### **definió en el desarrollo del algoritmo.**

Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo().

Proceso Configuración. Definir serial, pines. Llenar Digitalmente los reloj. Fin proceso.

Proceso cíclico. Imprimir Menu. Sugerir presión de teclas al usuario. Seleccionar opción para el funcionamiento de los leds. Seleccionar la imagen del led. Llamar la función publik. Fin proceso.

Proceso inicialización . Iniciar los relojes en 0 y 1. Fin proceso.

Proceso apagado. De 0 a 64 apagar cada led. Inicializar . Fin Proceso.

Proceso patron. Definir num, aux, tamaño, cifra. Pedir letra. Capturar letra. Calcular los dígitos del número ingresado. Declarar arreglo. Ingresar número anterior al arreglo. Fin proceso.

Función grupo8. Recorrer de 0 a 8 . Escribir valores lógicos digitales en el pin ser. Llamar a la función inicialización. Fin proceso.

Funcion orden. Declarar arreglo int fin. Para i = hasta tamano-1 incrementado de a 1. a es espacios en la posicion i-1. fin[a]=1. fin para. fin proceso.

Funcion guardador. Para i=1 hasta 64, aumentar de a 1 . Escribir digital en el puerto serial 1 para encender. Inicializar. Fin proceso.

Funcion imagen. Imprimir información del menu. Seleccionar opción con cambiar o switch. Caso 1:. Para i=0 hasta hasta 7 . Llamar a la función patron y la función grupo8. Fin ciclo for. break.

función publik. declarar variable tiempo y num-patrones. Imprimir el mensaje para que el usuario ingrese la. cantidad de patrones . Capturar dato con parseint. Mientras numj0. Ingresar el tiempo de visualizacion. Tiempo=seria.parseint. Fin mientras.

Declarar secuencia como arreglo de 2x2 y de 8 elementos Para i=0 hasta j=num-patrones, i++. Patronnn[8][8]. Llamar imagen(patronn). Para e=0 hasta 8,e++. Para a=0 hasta aj8,a++. Hacer que el arreglo seuencia en la posicon. [(e\*1)-1][a]. sea igual a patronn[e-1][a]. Fin ultimo for. Fin penúltimo for. Fin primer for.

Para q=0 hasta qi2,q++. Para m=1 hasta mj=num-patrones;m++. Para i=0 hasta ij8,ii++. Para a=0 hasta aj8,a++. Llamar a la función grupo8(secuencia[a\*m]);. Fin cuarto for. Delay-¿según el el valor que le usuario haya ingresado. Fin tercer for. Fin segundo for. Fin primer for.

### 3. Algoritmo implementado.

```
//pines
const int SER =5;
const int reloj_des=3;
const int reloj_salida=4;
// Menu
int opciones;

// Patrones
int patron1[][8] = {
    {0, 0, 0, 1, 1, 0, 0, 0},
    {0, 0, 1, 1, 1, 1, 0, 0},
    {0, 1, 1, 1, 1, 1, 1, 0},
    {1, 1, 1, 1, 1, 1, 1, 1},
    {1, 1, 1, 1, 1, 1, 1, 1},
    {0, 1, 1, 1, 1, 1, 1, 0},
    {0, 0, 1, 1, 1, 1, 0, 0},
    {0, 0, 0, 1, 1, 0, 0, 0},
};

int patron2[][8] = {
    {1, 0, 0, 0, 0, 0, 0, 1},
    {0, 1, 0, 0, 0, 0, 1, 0},
    {0, 0, 1, 0, 0, 1, 0, 0},
    {0, 0, 0, 1, 1, 0, 0, 0},
    {0, 0, 0, 1, 1, 0, 0, 0},
    {0, 0, 1, 0, 0, 1, 0, 0},
    {0, 1, 0, 0, 0, 1, 0, 0},
    {0, 1, 0, 0, 0, 0, 1, 0},
    {1, 0, 0, 0, 0, 0, 0, 1},
};

int patron3[][8] = {
    {1, 1, 0, 1, 1, 0, 1, 1},
    {1, 1, 0, 1, 1, 0, 1, 1},
    {0, 1, 1, 0, 1, 1, 0, 1},
    {0, 1, 1, 0, 1, 1, 0, 1},
    {1, 1, 0, 1, 1, 0, 0, 1},
    {1, 1, 0, 1, 1, 0, 0, 1},
    {0, 1, 1, 0, 1, 1, 0, 1},
    {0, 1, 1, 0, 1, 1, 0, 1},
};

int patron4[][8] = {
    {1, 1, 1, 1, 0, 0, 0, 0},
    {0, 1, 1, 1, 1, 0, 0, 0},
```

```

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

void setup()
{

  Serial.begin(9600);
  pinMode(SER,OUTPUT);
  pinMode(reloj_des,OUTPUT);
  pinMode(reloj_salida,OUTPUT);
  for(int i=3;i==5;i++)
  {digitalWrite(i,0); }
  //Serial.print("Bienvendio al menu");

}

void loop()
{
  //esto para que no pase de este campo sin el dato a ingresar
  Serial.print(F("bienvenido al menu!!!\n"));
  Serial.print(F(" presiona 1. para invocar la funcion verificacion\n presiona 2. para invocar la funcion imagen\n presiona 3. para
la funcion publik\n presiona 4. para ver la secuencia de Patrones\n \n \n"));
  while(Serial.available()==0){}
  opciones=Serial.parseInt();
  witch(opciones)
  {
    case 0:
      break;
    case 1:
      verificacion();
      Serial.println(F("todos los leds funcion correctamente \n porfavor presione cualquier l para volver al menu principal \n \n
\n"));
      while(Serial.available()==0){}
      apagado();
      break;
      ase 2:
      int patronn[8][8];
      imagen(patronn);
      break;
    case 3:
      publik();
      break;
    case 4:
      secuenciaPatrones();
      break;
    default:

```

```

        Serial.println(F("el dato ingresado no se contempla en el menu \n \n \n"));
    }

}

// funciones auxiliares

void inicializacion()
{
    // funcion para inicializar los relojes
    digitalWrite(reloj_des,0);
    digitalWrite(reloj_des,1);
    digitalWrite(reloj_des,0);

    digitalWrite(reloj_salida,0);
    digitalWrite(reloj_salida,1);
    digitalWrite(reloj_salida,0);
}

void apagado()
{
    for(byte i=0 ; i<64 ; i++)
    {
        digitalWrite(SER,0);
        inicializacion();
    }
}

void grupo8(int cadena[])
{
    //recorrer las filas dejando datos, pide una cadena de enteros
    for(short int e=0;e<8;e++)
    {
        digitalWrite(SER,*(cadena+e));
        inicializacion();
    }
}

void patron(int guardado[][8], short int num_arre)
{
    long int num, aux;
    short int tamano=0, cifra;
    Serial.print(F("ingrese el numero que representan los leds de la fila: "));
    Serial.print(num_arre);
    Serial.print(F("\n"));
    while(Serial.available()==0){}
    num=Serial.parseInt();
    Serial.print(num);
    Serial.print(F("\n"));
    while(num < 0)
    {
        Serial.print(F("ingrese el numero que representan los leds prendidos \n"));
        while(Serial.available()==0){}
        num=Serial.parseInt();
    }
}

```

```

        aux= num;
        while(num>0)
    {
        cifra= num % 10;
        if(cifra !=9 and cifra !=0)
        {
            tamano++;
        }
        num=num/10;
    }
    int espacios[tamano],ingresor[8]={0,0,0,0,0,0,0,0};

        ingreso(espacios,aux,cifra);
        orden(espacios,tamano,ingresor);
        guardador(guardado,ingresor,num_arre);
    }

void ingreso(int espacios[],long int aux,short int cifra )
{
    byte e=0;
    while(aux>0)
    {
        cifra = aux % 10;
        if (cifra !=9 and cifra !=0)
        {
            espacios[e]=cifra;
            e++;
        }
        aux= aux / 10;
    }
}

void orden(int espacios[],short int tamano,int fin[])
{
    int a;
    for(short int i=0; i<tamano ;i++){
        a=espacios[i]-1;
        fin[a]=1;
    }
}

void guardador(int arreglodarreglos[][8],int ingreso[],short int num_arre)
{
    for(byte i= 0 ; i <8;i++)
    {
        arreglodarreglos[num_arre][i]=*(ingreso+i);
    }
}

void verificacion()
{//funcion que prende todos los leds para verificar que funcionan

```



```

for(int i=1;i<=64;i++)
{
    digitalWrite(SER,1);
    inicializacion();
}
}

```

```

void imagen(int patronn[][8])
{
    Serial.print(F("Menu de la funcion imagen\n"));
    Serial.print(F(" presiona 1. para ingresar tu propio patron\n \n \n"));
    while(Serial.available()==0){}
    opciones=Serial.parseInt();

    switch(opciones)
    {
        case 1:
            Serial.print(F("ingrese el numero que representan los leds prendidos \n "));
            Serial.print(F("Ejemplo:si usted ingresa 1468, encenderan los led en posicion 1 4 6 8 \n"));

            for(int i=0;i<8;i++)
            {
                patron(patronn,i);
                grupo8(patronn[i]);
            }

            break;
    }
}

```

```

void publik()
{
    int tiempo=0;
    short int num_patrones=0;
    Serial.println(F("Ingrese la cantidad de patrones que desea: "));
    while(Serial.available()==0){}
    num_patrones = Serial.parseInt();

    while (num_patrones < 0)
    {
        Serial.println(F("debes ingresar el numero de patrones que deseas mostrar"));
        Serial.println(F("Ingrese la cantidad de patrones que desea: "));
        while(Serial.available()==0){}
        num_patrones = Serial.parseInt();
    }

    Serial.println(F("Ingrese los milisegundos entre cada visualizacion entre patrones : "));
    while(Serial.available()==0){}
    tiempo = Serial.read();
}

```

```

while (tiempo < 0)
{
    Serial.println(F("Ingrese el tiempo de visualizacion entre cada patron :"));
    while(Serial.available()==0){}
    tiempo = Serial.parseInt();
}
int secuencia[8*num_patrones][8];
for(short int i=0; i <= num_patrones; i++)
{ int patronn[8][8];
    imagen(patronn);

    for(short int e=1; e<=8; e++)
    {
        for(short int a=0; a<8; a++)
        {
            secuencia[(e*i)-1][a]= patronn[e-1][a];
        }
    }
}

for(short int q=0; q<2; q++)
{
    for(short int m=1; m <= num_patrones; m++)
    {
        for(short int i=0; i<8; i++)
        {
            for(short int a=0; a<8; a++)
            {
                grupo8(secuencia[(a*m)]);
            }
            delay(tiempo);
        }
    }
}

void secuenciaPatrones() {
    int tiempoEntrePatrones =3000;//podemos ajustar el tiempo entre patrones en milisegundos

    for (int patron = 0; patron < 4; patron++) { // Cambia 4 por el número total de patrones que tengas
        switch (patron) {
            case 0:
                mostrarPatron(patron1);
                break;
            case 1:
                mostrarPatron(patron2);
                break;
            case 2:
                mostrarPatron(patron3);
                break;
            case 3:
                mostrarPatron(patron4);
                break;
        }
    }
}

```

```

    }

    delay(tiempoEntrePatrones);
    apagarMatriz(); // Apagar la matriz antes de mostrar el siguiente patrón
    delay(tiempoEntrePatrones);
}
}

void mostrarPatron(int patron[][8]) { // Cambio de byte a int
    for (int fila = 0; fila < 8; fila++) {
        grupo8(patron[fila]);
        delay(0); // Ajusta el tiempo de visualización de cada fila según tu preferencia
    }
}

void apagarMatriz() {
    int matrizApagada[8][8]; // Cambio de byte a int
    for (int fila = 0; fila < 8; fila++) {
        for (int columna = 0; columna < 8; columna++) {
            matrizApagada[fila][columna] = 0;
        }
    }
    for (int i = 0; i < 8; i++) {
        grupo8(matrizApagada[i]);
    }
}

```

## **Evolución del algoritmo y consideraciones a tener en cuenta en la implementación.**

El algoritmo implementado fue cambiado Varias veces dado que al hacer pruebas de escritorio cada vez éstas nos sugerían cambios como en los tipos de datos como punteros y la eliminación de funciones que, aunque reducían el tiempo de ejecución del código y eliminación de líneas de código también, se hacían tardías ya que se llegó a un punto donde habían más tuve un código que netraba en bucle o se trababa en cierto momentos y el código editor de Tindercard no permitía la edición.

También cree en el repositorio un análisis breve como fui implementando el diseño de la solución.