top=1.5cm, bottom=1.0cm, left=1cm, right=1cm

# PARCIAL 1

Informática II

**Juan Esteban Garcia Durango**
**Jessica Valentina Gaviria Samboni**

Despartamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Abril de 2021

# Índice

# 1. Sección introductoria

El siguiente trabajo contiene el proceso realizado en pro de el desarrollo del desafio planteado por la empresa Informa2 S.A.S..

# 2. Sección de contenido

## 2.1. Análisis del problema y consideraciones para la alternativa de solución propuesta.

El primer planteamiento que nos trazamos respecto al problema propuesto, imprimir en una matriz de leds 8*8 un mensaje ingresado por el usuario, fue el cómo reducir la cantidad de puertos digitales a emplear,y esto tuvo una solucion viable en el hecho de poder conectar los integrados 74HC595 en serie, ya que de esta forma se emplean los mismos 3 puertos del arduino, y lo que se hace es que los bits ingresados se van desplazando, hasta llenar el registro.

Otros de los planeamientos previos fueron:
**1.**Decidir si los patrones serían predeterminados o diseñados por el usuario. Bajo la aparición de esta duda, decidimos darle la facilidad y oportunidad al usuario de que si decidía mostrar una letra en la matriz de leds, simplemente seria ingresar la letra mayúscula que quisiera.Pero de igual forma se le permitiría generar el mismo el patrón o patrones a su gusto.De lo anterior se generaron 2 alternativas:
**1.1** Manejo de los bits para poder representar la letra mayúscula predeterminada: Cada letra tendría su propio arreglo, que a la vez almacenaria subarreglos, en los cuales cada subarreglo contendria la informacion de los bits para cada columna de leds.
**1.2** Manejo de los bits para representar el patrón deseado por el usuario: Para este item lo que hicimos fué crear una arreglo bidimensional de [8][8],el cual tendría en todas sus entradas el valor cero inicialmente, pero que se vería modificado a 1, en caso de que el usuario quisiese emplear ese led en su patrón y por consiguiente encenderlo (estado 1 o HIGH).

**2.Análisis del problema y consideraciones finales.**
En vista del desbordamiento de la memoria disponible en tinkercad, no pudimos hacer la implementacion de los patrones de las letras predeterminadas,

decidimos dejar unicamente la opcion en la cual el usuario genera su patrón libremente.Y el hecho de usar memoria dinamica tuvo un conflicto para usar los arreglos de los patrones predeterminados. Adicional a esto, al implenmeter el new para un arreglo, el tinkercad arrojaba este error: 'arreglo was not declared'.

## 2.2. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo.

En la Figura (**??**), se presenta el esquema de las tareas.


Figura 1: Logo de C++

## 2.3. Problemas de desarrollo que presentó.

Unos de los problemas que presentaron fueron:

1.En un primer momento no teníamos idea de cómo reducir los puertos digitales a usar, ya que originalmente tomamos la decision de interconectar el SRCLK y el RCLK de los 8 integrados que usamos, para asi solamente emplear un puerto, ya que el SRCLK y el RCLK funcionan bajo la misma idea de activacion por flanco. Pero aún asi nos seguían faltando pines, ya que pretendíamos conectar la entrada serial de cada integrado a un pin(8 en total). En vista de lo anterior y gracias a una investigación ardua, encontramos la foma de reducir los puertos a emplear.
2.Ocupamos más espacio de lo admitido por el footprint del arduino, lo cual imposibilitó las pruebas del código.
3.Se nos presentaron varios inconvenientes con el desarrollo del circuito, ya que se nos borró el montaje del circuito.
4.Error intermitente de los datos guardados en el heap, lo cual imposibilita la correcta lectura y ubicacion del patron en los leds.

## 2.4. Algoritmo implementado.

```
#include <SoftwareSerial.h>

int main(){
//Aqui se realiza la declaracion de algunas variables empleadas posterio
short int eleccion=0;
short int Letra, retraso=0,cantidad;
short int estilo;
int** patron;

//puertos seriales
//ARDUINO ------>74HC595
//  2     -->  Ser    Registro de entrada
//  4     --> rclk    Reloj desplazamiento
//  5     -->srclk    Reloj salida
const int Ser=2;
const int rclk=4;
```

```
const int srclk=5;

//Prototipo de las funciones
void ingresoDatos( int** ,int ,int ,int);
void Verificacion(int , int, int);
int** imagen(int ,int ,int);
void publik(int ,int ,int ,int ,int );

//Configuraci n de los pines digitales Ser(2),rclk(4) y srclk(5)
pinMode(Ser, OUTPUT);
pinMode(rclk, OUTPUT);
pinMode(srclk, OUTPUT);

//Estado LOW (0) de los pines
digitalWrite(Ser,0);
digitalWrite(rclk,0);
digitalWrite(srclk,0);

init( );
  //SET UP

  //Inicializacion puerto Serial
  Serial.begin(9600);

  //MEN PARA EL USUARIO:
  Serial.println("1.Verificar que todos los leds funcionan ");
  Serial.println("2.Mostrar  nico  patr n de leds");
  Serial.println("3.Mostrar una secuencia de patrones");
  Serial.println("4.Mostrar el manual de instrucciones para ingresar un

  while(Serial.available()==0);
  Serial.println("Ingrese la opcion que desee:");
  eleccion=Serial.parseInt();
  Serial.print(eleccion);
  Serial.println();

//////////////////
//PRIMERA ELECCION, la verificacion del estado de los 64 leds.
```

```
if (eleccion==1){
    Verificacion(Ser,srclk,rclk);
                    }
/////////////////
//SEGUNDA ELECCION, mostrar el unico patron deseado por el usuario
else if(eleccion==2){

    /*Al doble puntero patron, se le asigna el resultado de la funcion
    imagen, la cual permite llevar el registro de los leds que el
    usuario quiere encender para formar el patron, y retorna una un
    doble puntero que apunta a un arreglo que contiene la matriz
    modificada con los unos y ceros*/
        patron=imagen(Ser,srclk,rclk);

    //Se llama la funcion -ingresoDatos-,la cuasl recibe un doble punter
    //y al acceder a cada uno de los elementos de la matriz modificada,
    //manda esta se al al puerto ser para que si el dato es 1 encienda
    //HIGH, de lo contrario manda la se al LOW(0)
    ingresoDatos(patron,Ser,srclk,rclk);
                    }
/////////////////
//TERCERA ELECCION, Secuencia de patrones

else if(eleccion==3){
  Serial.println("Ingrese el retraso que desee:");
  while(Serial.available()==0);
  retraso=Serial.parseInt();
  Serial.print(retraso);

  Serial.println("Ingrese cantidad de patrones:");
  while(Serial.available()==0);
  cantidad=Serial.parseInt();
  Serial.print(cantidad);

  /*Llamado a la funcion publik la cual guarda en un arreglo doble punte
  la direccion de memoria del arreglo modificado por el usuario, y esto
  repite la cantidad de patrones deseados por el usuario*/
```

```
  publik(retraso,cantidad,Ser, srclk,rclk );


   }

//////////////////
//CUARTA ELECCION, Visualizacion del menu donde se explica al
//usuario los pasos a seguir
else if(eleccion==4){
    Serial.println("Para ingresar un patron de leds se hace de la siguer
        Serial.println("1.el programa le pedira que escoja del 1 al 8, l
        Serial.println("la cantidad de columnas de la matriz de leds");
        Serial.println("            |->0");
        Serial.println("            |  0");
        Serial.println("columna ===|  0");
        Serial.println("            |  0");
        Serial.println("            |->0");
        Serial.println("2.Una vez escogido la columna, el paso siguente
    Serial.println("entre el 1 y el 8, el cual significa el elemento de
    Serial.println("es decir, si por ejemplo escoges el elemento 2, se e
    Serial.println("Y se ver  representado como un led encendido en esa
    Serial.println("              0");
        Serial.println("elemento 2 ->1");
        Serial.println("              0");
        Serial.println("              0");
        Serial.println("              0");
        Serial.println("              0");
        Serial.println("              0");
        Serial.println("              0");
        Serial.println("Nota 1: este proceso se repetir  indefinidament
    Serial.println("Cuando el programa le pregunte : si desea terminar i
    Serial.println("Si usted ya ha terminado de registrar su patr n de
    Serial.println("Nota 2:En caso de equivocarse al dibujar el patr n

}

while(1);
```

```
return 0;
 }

//Funcion que recibe como parametros de entrada: un doble puntero(apunta
//a la direccion del arreglo)y los puertos Ser, scrlk y rclk
void ingresoDatos(int** acom, int Ser, int srclk,int rclk){
        for(short int k=0;k<8;k++){
        for(short int bits=0;bits<8;bits++){

        digitalWrite(Ser,*(*(acom+k)+bits));
        /*Aqui se accede por medio de un puntero doble
         a los datos contenidos en el arreglo:
         1-->HIGH   o   0-->LOW y se le asigna ese estado
         al puerto Ser */

        digitalWrite(srclk,0);
        digitalWrite(srclk,1);//Activacion por flanco srclk
        digitalWrite(srclk,0);

        digitalWrite(rclk,0);
        digitalWrite(rclk,1);//Activacion por flanko rclk
        digitalWrite(rclk,0);
        }
    }
}


//Funcion tipo void que recibe como parametros de entrada:
//los puertos Ser,srclky rclk
void Verificacion(int Ser, int srclk,int rclk){

  for(int cont=0;cont<65;cont++){
                                  //En este ciclo for se le asigna al
        digitalWrite(Ser,1);      //puerto Ser el estado HIGH 64 veces
                                  //que son el total de leds

        digitalWrite(srclk,0);
```

```
        digitalWrite(srclk,1);  //Activacion por flanco
        digitalWrite(srclk,0);

        digitalWrite(rclk,0);
        digitalWrite(rclk,1);   //Activacion por flanco
        digitalWrite(rclk,0);
        }
  delay(1000);



  for(int cont=0;cont<65;cont++){
     digitalWrite(Ser,0);//--->     //En este ciclo se le asigna
     digitalWrite(srclk,0);         //64 veces el estado LOW al
     digitalWrite(srclk,1);         //puerto Ser, para apagar los leds
     digitalWrite(srclk,0);

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

}



//Esta funcion, tipo doble puntero recibe como parametros de entrada:
// los puertos Ser,srclk y rclk
int** imagen(int Ser,int srclk,int rclk){
    int Numero=0,m=0,t=0;
    int **pt = new int*[8];

    for(short int i=0;i<8;i++){     //Creacion de un arreglo[8][8] en la
       pt[i] = new int[8];     //dinamica
    }

    for (short int m=0;m<8;m++){        //Asignacion del valor cero a ca
        for( short int t=0;t<8;t++){    //de los datos del arreglo
```

```
            pt[m][t]=0;
        }
    }

    while(Numero!=2){
        for(short int t=0;t< 8;t++){
            for(short int elem=0;elem<8;elem++){
                Serial.print(pt[elem][t]);
            }
        Serial.println();
        }
    Serial.print("escoja la columna que desea modificar: ");
    while(Serial.available()==0);
    t=Serial.parseInt();
    Serial.println(t);
    Serial.print("en un rango de 1 a 8 escoja la el caracter que quiere
        while(Serial.available()==0);
    m=Serial.parseInt();           //En este punto se le pide al usuario
    Serial.println(m); //el numero de la columna y el
caracter de 1 a 8
        //que representa el led de esa columna elegido

      pt[t-1][m-1]=1;  //------>al dato ubicado en la fila y columna selec
    //por el usuario se le asigna el valor 1 (HIGH) .

    Serial.print("si desea terminar ingrese 2, si desea continuar ingres
    while(Serial.available()==0);
    Numero=Serial.parseInt();
    Serial.println(Numero);
        }
   delete []pt;
return (pt);  //------>se retorna la direccion del arreglo modificado
}


//Funcion tipo void la cual recibe como parametros de entrada
//el delay del usuario, la cantidad de patrones y los puertos(Ser,scrlk
void publik(int retraso,int cantidad,int Ser,int srclk,int rclk ){
```

```
  int** Almacen[8]={}; //Declaracion arreglo doble puntero vac o
  int** patrones=0;
  for(short int cont=0;cont<cantidad;cont++){
       patrones=imagen(Ser,srclk,rclk);
    Almacen[cont]=patrones;//Asignacion del retorno de la funcion imager
  }                                  //al arreglo doble puntero Almacen
                                    //este proceso se realiza la cantidad
                                   // de patrones queridos por el usuario

  for(short int cant=0;cant<=4;cant++){
    for(short int cant1=0;cant1<cantidad;cant1++){
  ingresoDatos( Almacen[cant1],Ser,srclk,rclk); //Llamado a la funcion
    delay(retraso*1000);}              //ingresoDatos la cual va rrecorrie
  }                           //el arreglo Almacen y pasando esos datos de
  delete [] Almacen;
}                           //1's y 0's al puerto Ser
```

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

En primera instancia implementamos un algoritmo solucion, en el cual tratamos de incluir arreglos bidimensionales predeterminados, los cuales luego serian leidos, y cada dato representaria en estado de cada led.Esto no fue posible debido a varios problemas, como:implementacion de arreglo tridimensional el cual contendria como subarreglos los arreglos predeterminados para cada patron.

Finalmente decidimos dejar al usuario ingresar su propio patron libremente, tomando en cuenta las instrucciones dadas en el manual del algoritmo. En este codigo final, usamos la memoria dinamica para almacenar los arreglos que contendrian los patrones diseñados por el usuario.Ademas de esto, hicimos uso de punteros para acceder a los datos de los arreglos y de igual forma empleamos arreglos tipo doble puntero.

Consideraciones:
1.Es primordial tener todos los parametros claros antes de iniciar la codificacion.

2.Hacer un analisis profundo de todas las posibles soluciones a emplear y hacer un contraste de sus pro y sus contra detalladamente, para elegir la opcion mas viable.
3.Tener conceptos teoricos claros, para asi hacer uso adecuado de todas las herramientas.

# Referencias