Informe

Parcial 1 - Informática II

Miguel H. Martin Matiz
Emiliano Lince Díaz

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

Índice

1. Introducción

Partiendo de una situación problema propuesta, una empresa quiere implementar estratategias que le permitan promover sus productos y llamar la atención de sus potenciales clientes, se diseña una solución visual del tipo panel de leds o aviso publicitario que permita a la empresa mostrar algún mensaje publicitario.
Para esto, se realiza un montaje de una matríz de leds de 8x8 controlados por un Arduino UNO y un integrado registro de desplazamiento de 8 bits (conversor serial a paralelo).

2. Contenido

2.1. Analisis y solución del sistema de LEDs 8x8

Creación de un sistema de leds de 8x8 para implementarlo en anuncios publicitarios, el sistema esta conformado por 64 leds, 64 resistencias, un Arduino uno y 8 integrados 74HC595. Para que el sistema de leds 8x8 sea efectivo debemos crear un programa el cual reciba patrones (números, letras u otros caracteres), de los que la matriz de leds 8x8 debe imprimir dichos patrones, para ello vamos a usar modificadores y punteros, como arrays y operaciones con arrays.

Para la creación de dicho sistema comenzamos con la organización de la matriz 8x8 con varios leds, luego conectaremos todos los leds con la tierra en serie y conectando a cada uno una resistencia de 560 ohmios, también vamos conectando los 8 integrados 74HC595 a un Arduino uno en 3 pines digitales (en nuestro caso usamos los pines 5,4 y 2) y conectamos la fuente y la tierra a los 8 integrados, en nuestro caso conectamos los integrados en serie para ahorrar mas pines digitales y solo usar los 3 hablados anteriormente, podemos apoyarnos de placas de prueba para las conexiones para que así quede mas organizado el sistema. Luego de hacer todas las respectivas conexiones vamos al algoritmo y comenzamos a programarlo, Nosotros iniciamos creando una función la cual la llamamos comprobación la cual vamos a utilizar para ver si funcionan todos los leds correctamente, después creamos otra función que llamaremos publik la cual, por medio de arreglos, vamos a hacer que al ingresar un patrón por la consola y la cantidad de tiempo que dura cada uno, en el matriz de leds muestre dicho patrón con su respectivo tiempo de duración.

2.2. Evolución del Sistema 8x8

Muestra de como fué la evolución del algoritmo:

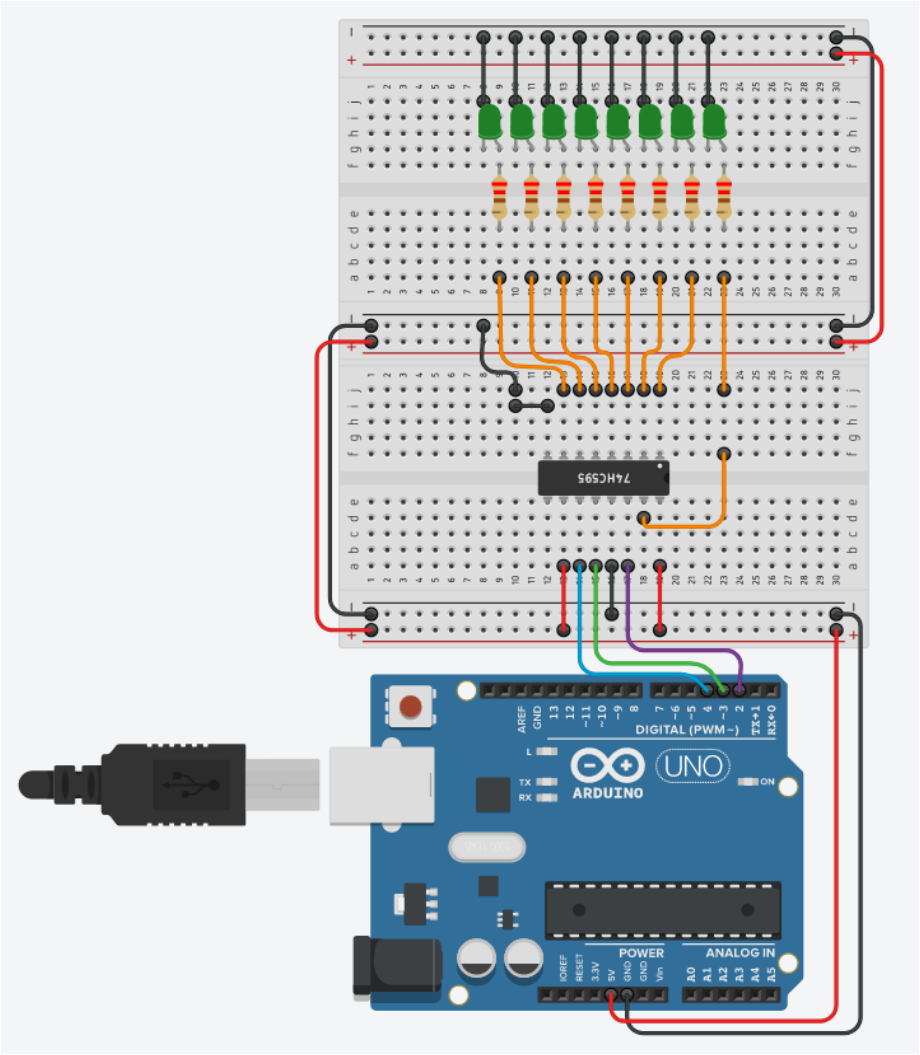

Figura 1: Montaje inicial del circuito.

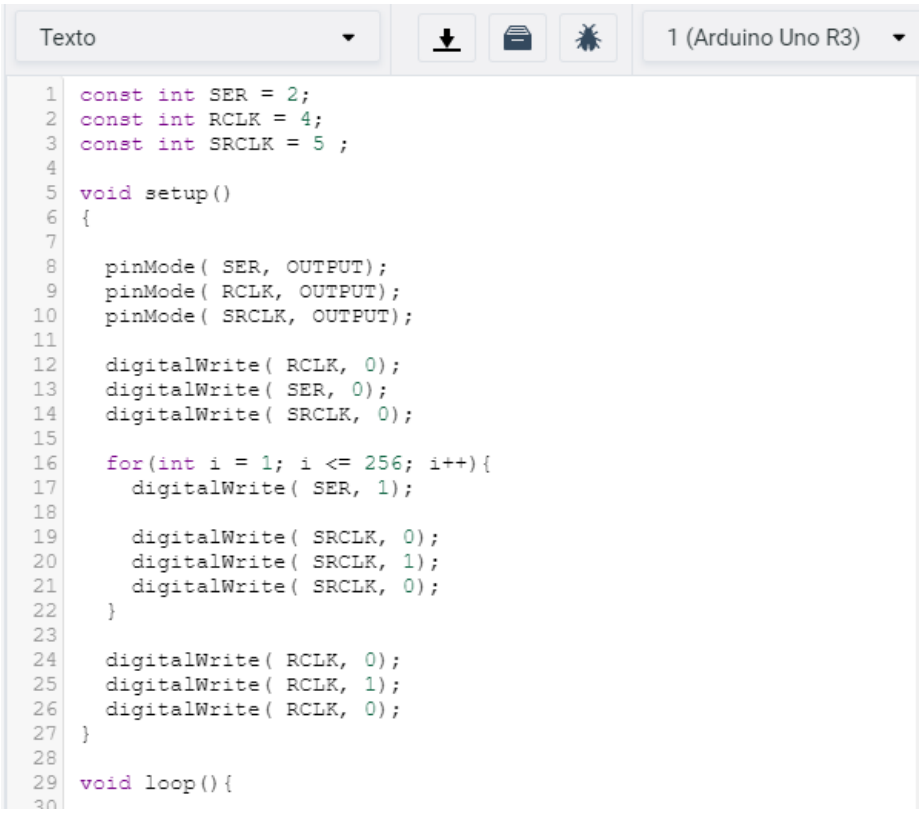

Figura 2: Algoritmo del montaje inicial del circuito.

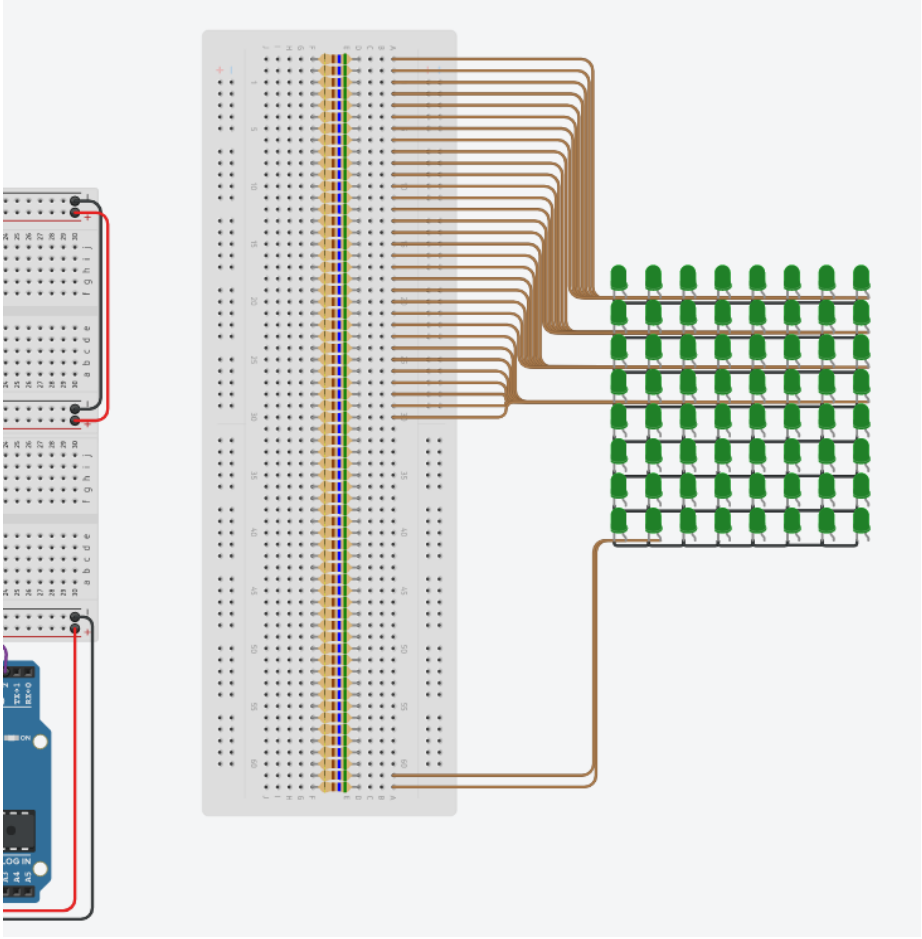

Figura 3: Montaje de las cuadricula 8x8 de LEDs.

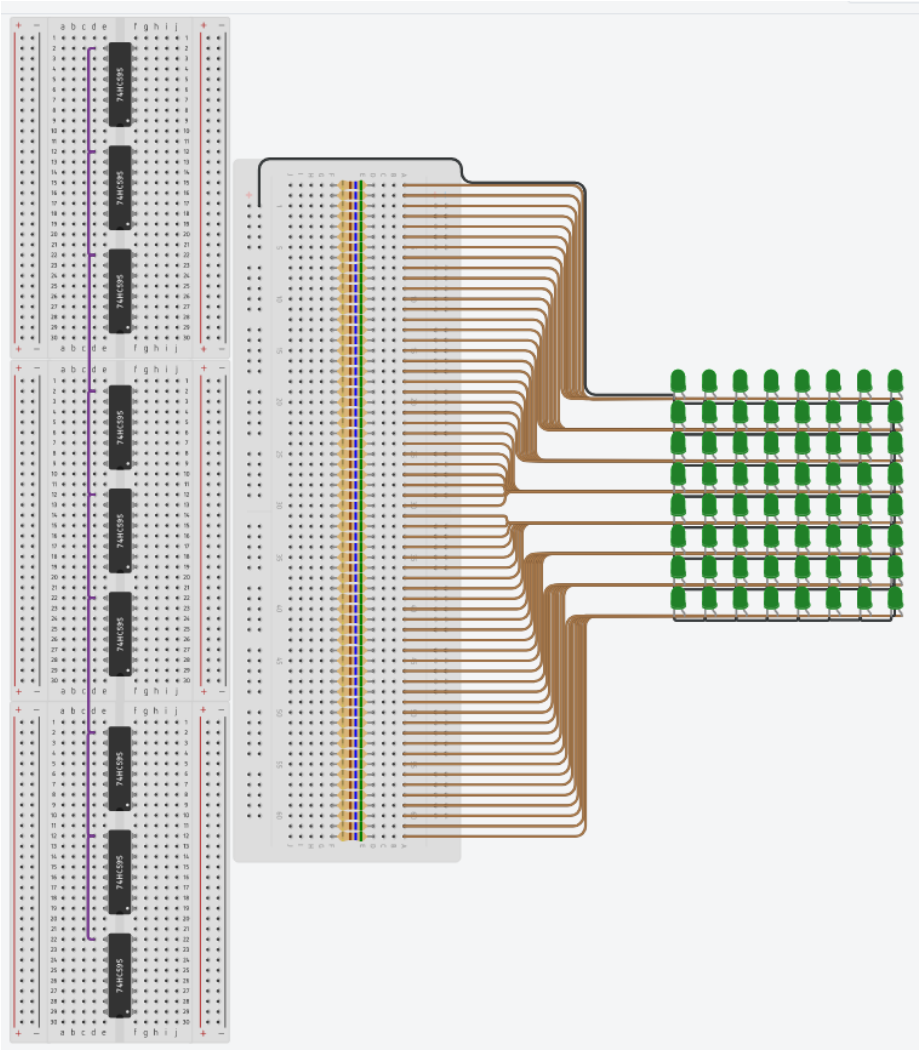

Figura 4: Montaje de los 8 integrados al sistema.

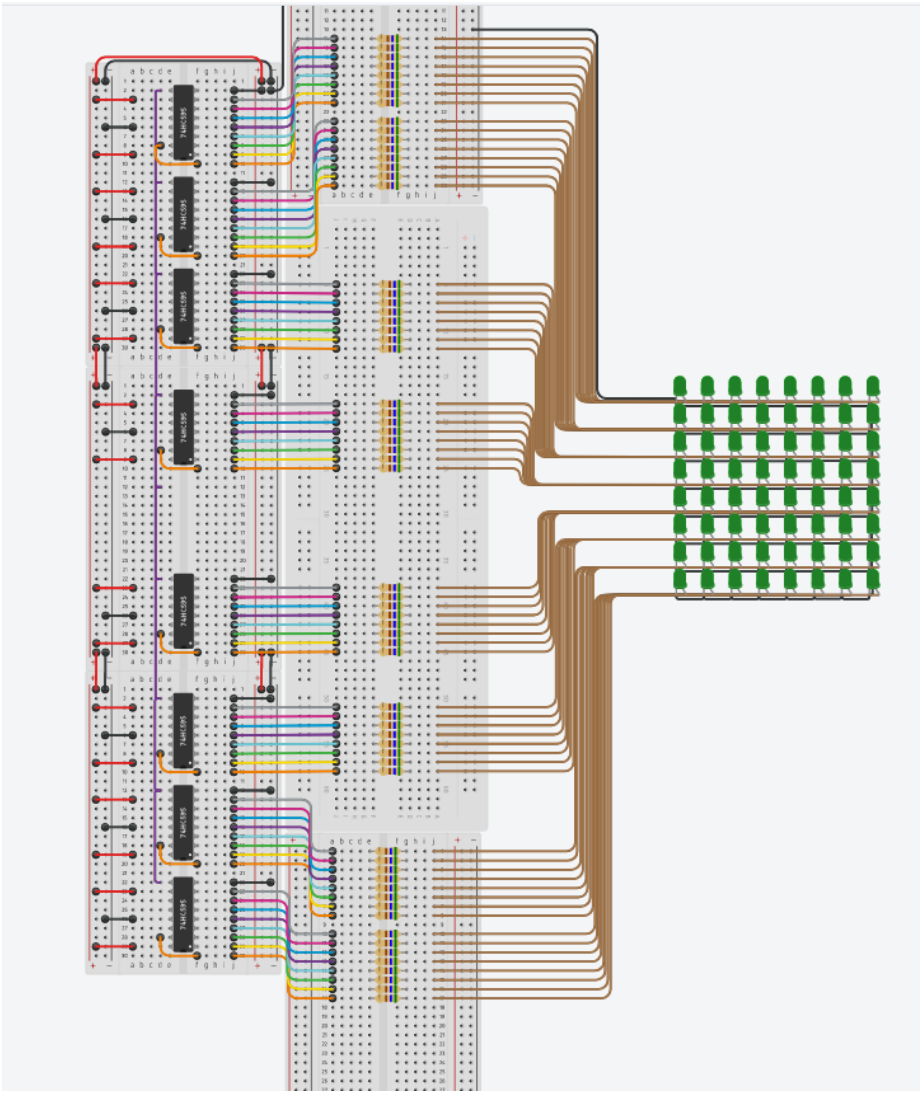

Figura 5: Montaje del sistema con los 8 Integrados 64 Resistencias y el sistema 8x8 de LEDs (todos coneectados).

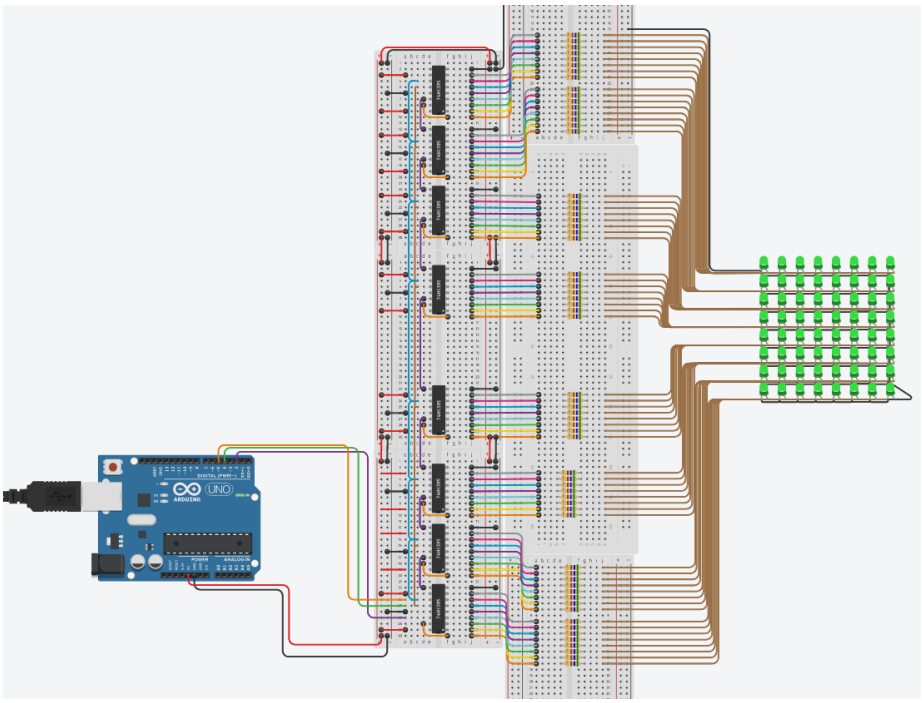

Figura 6: Montaje del circuito completo con el sistema 8x8 conectado y funcionando.

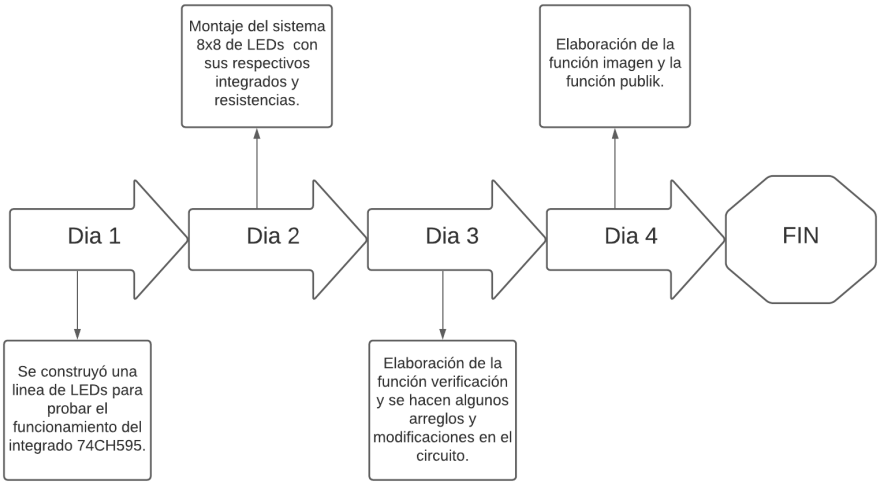

Figura 7: Diagrama evolución del sistema 8x8 de LEDs

2.3. Evolución del Algoritmo

Imagenes de la evolución de los Algoritmos implementados:

```
const int SER = 2;
const int RCLK = 4;
const int SRCLK = 5 ;

//unsign int puertos[3] = {2, 4, 5}

void setup()
{
  // {Paso 1} Configuración de puertos digitales como salida
  pinMode( SER, OUTPUT);
  pinMode( RCLK, OUTPUT);
  pinMode( SRCLK, OUTPUT);

  digitalWrite( RCLK, 0);
  digitalWrite( SER, 0);
  digitalWrite( SRCLK, 0);

  // Envío de datos al CI(registro) 74HC595
  // {Paso 2} Llevar el dato al puerto serial del 74HC595 -> 0/1
  digitalWrite( SER, 1);

  // {Paso 3} Activar el reloj de la 1ra etapa de los registros SRCLK
  // (los registros de entrada)

  digitalWrite( SRCLK, 0);
  digitalWrite( SRCLK, 1);
  digitalWrite( SRCLK, 0);

  digitalWrite( SER, 0);

  digitalWrite( SRCLK, 0);
  digitalWrite( SRCLK, 1);
  digitalWrite( SRCLK, 0);

  digitalWrite( SER, 1);

  digitalWrite( SRCLK, 0);
  digitalWrite( SRCLK, 1);
  digitalWrite( SRCLK, 0);


  // {Paso 3} Activar el reloj de la 2Da etapa de los registros RCLK
  // (los registros de salida)

  digitalWrite( RCLK, 0);
  digitalWrite( RCLK, 1);
  digitalWrite( RCLK, 0);

}

void loop(){

}
```

Figura 8: Primer codigo implementado para prueba de LEDs

```
int pinData  = 2;
int pinLatch = 3;
int pinClock = 4;
char input;
#define TIEMPO 200

void ledWrite(int RLed, int GLed, int BLed, int YLed, int KLed, int MLed, int HLed, int WLed){
   shiftOut(pinData, pinClock, LSBFIRST, BLed);
   shiftOut(pinData, pinClock, LSBFIRST, GLed);
   shiftOut(pinData, pinClock, LSBFIRST, RLed);
   shiftOut(pinData, pinClock, LSBFIRST, YLed);
   shiftOut(pinData, pinClock, LSBFIRST, KLed);
   shiftOut(pinData, pinClock, LSBFIRST, MLed);
   shiftOut(pinData, pinClock, LSBFIRST, HLed);
   shiftOut(pinData, pinClock, LSBFIRST, WLed);
   digitalWrite(pinLatch, HIGH);
   digitalWrite(pinLatch, LOW);
}

void setup(){
   pinMode(pinData, OUTPUT);
   pinMode(pinLatch, OUTPUT);
   pinMode(pinClock, OUTPUT);
   Serial.begin(9600);
}

void loop(){
   ledWrite(128,128,128,128,0,128,128,0);
   ledWrite(64,64,64,64,0,64,64,64);
   ledWrite(32,32,32,32,0,0,32,32);
   ledWrite(16,16,0,16,0,0,16,16);
   ledWrite(8,8,0,8,0,0,8,8);
   ledWrite(4,4,0,4,4,4,4,4);
   ledWrite(2,2,0,2,2,2,2,2);
   ledWrite(1,1,0,0,1,1,1,0);
}

void verificacion(){
   ledWrite(128,128,128,128,128,128,128,128);
   ledWrite(64,64,64,64,64,64,64,64);
   ledWrite(32,32,32,32,32,32,32,32);
   ledWrite(16,16,16,16,16,16,16,16);
   ledWrite(8,8,8,8,8,8,8,8);
   ledWrite(4,4,4,4,4,4,4,4);
   ledWrite(2,2,2,2,2,2,2,2);
   ledWrite(1,1,1,1,1,1,1,1);
}

void imagen(){
  if (Serial.available()>0){
    input=Serial.read();

    Serial.println(input);
  }
}
```

Figura 9: Segundo codigo implementado para prueba de LEDs

2.4. Algoritmo final

include ¡iostream¿
using namespace std;
//const int MAXROW = 8; //const int MAXCOLUMN = 8;
define MAXROW (unsigned char)8 define MAXCOLUMN (unsigned char)8
const int SER = 2; const int RCLK = 4; const int SRCLK = 5 ;
unsigned int digitalSignal[3] = 0, 1, 0; unsigned int puertos[3] = SER, RCLK, SRCLK;
int (*matrizLeds)[MAXCOLUMN] = new int[8][8];
int patron[MAXROW][MAXCOLUMN];
//int charA[]=65,126,255,195,255,255,195,195,195; //A //int charI[]=73,255,255,24,24,24,24,255,255; //I //int charE[]=69,255, 255, 192, 254, 254, 192, 255, 255; //E //int charJ[]=74,3, 3, 3, 3, 3, 195, 255, 126; // J
int characters[][9] = 65,126,255,195,255,255,195,195,195, // A 73,255,255,24,24,24,24,255,255, // I 69,255, 255, 192, 254, 254, 192, 255, 255, // E 74,3, 3, 3, 3, 3, 195, 255, 126 // J ;
// ********** FUNCTIONS PROTOTYPE **********
int verificacion(int [][MAXCOLUMN]); int imagen(int [][MAXCOLUMN],int [][MAXCOLUMN]);
//int decimalToBinary(int [], int (*p)[MAXCOLUMN]); int decimalToBinary(int []);
void dataSER(int); void signalSRCLK(); void signalRCLK();
// **********
int main() //int b[] = 8,41; //int arreglo[][3] = *b,7,54,35,15; //int patron[MAXROW][MAXCOLUMN]; //int (*ptrPatron)[MAXCOLUMN] = patron;
/* // Caracter a ASCCI a binario char caracter; int binaryChar[8]=; int i = 0; while(1) cout « endl; cout " Ingrese caracter: "; cin " caracter; cout " Caracter «¡caracter ' .en ASCII: «¡int(caracter) endl; caracter = int(caracter); do caracter = caracter / 2; binaryChar[i] = caracter i++; while (caracter / 2 ¿0); for (int j = 0; j ¡8; j++) cout binaryChar[j]; */
»"'// Inicialización de los puertos digitales // for(unsigned int i = 0 ; i ¡3 ; i++) // pinMode(puertos[i] , OUTPUT); // digitalWrite(puertos[i] , 0); // char caracter;
»"'cout Caracter: "; cin caracter;
»"'//cout .Aquí: «¡**(arreglo+1) endl;
»"'//cout .Aquí«¡*(*(characters+1)+3) ; cout "Decimal: «¡**(characters+2) ;
»"'for(int i = 0; i ¡4 ; i++) // if(int(caracter) == **(characters + i)) // decimalToBinary(*(characters+i), ptrPatron); // if(caracter == **(characters+i)) //Rif(caracter == characters[i][0]) //out "Valor: «¡characters[i][0]; //cout "Bit: «¡characters[i+1][0]; //cout "Bit: «¡charA[i+1]; cout characters[i]; //decimalToBinary(characters[i], ptrPatron);
»"'// Envía como parámetro el arreglo que esté en la posición actual decimalToBinary(characters[i]); /* for(int i = 0; i ¡1 ; i++) if(int(caracter)

```
== **(characters + i) ) //decimalToBinary( *(characters+i), ptrPatron ); //-
decimalToBinary( *(characters+i)); //decimalToBinary( characters[i][0] );  //
if( caracter == charA[0] ) // //cout  "Valor: «¡characters[i][0]; // //cout  "Bit:
«¡characters[i+1][0]; // //cout  "Decimal: «¡charA[i+1]; // //cout  characters[i];
// //decimalToBinary( characters[i], ptrPatron ); // decimalToBinary( charA);
//  */
    »"'//int matrizLeds[MAXROW][MAXCOLUMN]; //int (*ptrMatrizLeds)[8][8];
    »"'//ptrMatrizLeds = matrizLeds;
    »"'// verificacion(matrizLeds); // imagen(ptrPatron, matrizLeds); imagen(patron,
matrizLeds);
    »"'return 0;
    »"'// ************************************************** // ***************
FUNCTIONS DEFINITION *************** // **************************************************
    »"'//int patron[MAXROW][MAXCOLUMN]; //int charA[]=65,126,255,195,255,255,195,195,195;
    »"'int eraser(int matriz[][8])
    »"'// Rellenado de toda la matríz con valores igual a 1 for(int row = 0; row
¡MAXROW ; row++) for(int column = 0; column ¡MAXCOLUMN ; column++)
matriz[row][column] = 0; cout   matriz[row][column]; //dataSER(matriz[row][column]);
signalSRCLK(); signalRCLK();  //cout  endl;
    »"'//dataSER(matriz[]);
    »"'return 0;
    »"'//int decimalToBinary( int arrayPatron[], int (*patron)[8] ) int decimal-
ToBinary( int arrayPatron[] )
    »"'int bit, decimal=0;
    »"'//cout  "in function: «¡arrayPatron[0]; //cout  "in Function: «¡arrayPatron[1];
for(int j = 0; j ¡8; j++) bool biteDivide = true; int k=8; //for(int k = 7;
k ¿= 0; k–) //division = arrayPatron[j+1] / 2; decimal = arrayPatron[j+1];
//cout  "in Function: «¡decimal; //bit = arrayPatron[j+1] //bit = decimal whi-
le(biteDivide) k–; //division = *(arrayPatron + (j+1) ) / 2; bit = decimal
    »"'//division = arrayPatron[j+1] / 2; //division = arrayPatron[m] / 2; //-
cout   *«¡division;  //cout   *«¡bit;  //bit = *(arrayPatron + (j+1) ) //bit =
arrayPatron[j+1]
    »"'//*(*(patron+j)+k) = bit; patron[j][k] = bit; //cout  «¡bit; //m++;
    »"'if( decimal / 2 == 0 ) biteDivide = false;
    »"'decimal = decimal / 2;  //cout  "K: «¡k  endl; for(int m=k-1; m ¿0 ; m– )
patron[j][m] = 0; // *(*(patron+j)+k) = 0;  //cout  «¡bit; // //END for k cout
endl;
    »"'return 0;
    »"'int verificacion(int matriz[][8])
    »"'// Rellenado de toda la matríz con valores igual a 1 for(int row = 0; row
¡MAXROW ; row++) for(int column = 0; column ¡MAXCOLUMN ; column++)
matriz[row][column] = 1; cout   matriz[row][column]; //dataSER(matriz[row][column]);
signalSRCLK(); signalRCLK();  //cout  endl;
    »"'//dataSER(matriz[]);
    »"'return 0;
```

»”’//**********
int imagen(int patron[][8], int matriz[][8]) /*for(int row = 0; row ¡MAXROW ; row++) dataSER(charA[row]); signalSRCLK(); signalRCLK(); */
»”’for(int row = 0; row ¡MAXROW ; row++) for(int column = 0; column ¡MAXCOLUMN ; column++) matriz[row][column] = patron[row][column]; cout matriz[row][column]; dataSER(matriz[row][column]); signalSRCLK(); signalRCLK(); cout endl; return 0;
»”’//**********
void dataSER(int bit)
»”’// Señal de dato SERIAL //digitalWrite(SER, bit);
»”’//**********
void signalSRCLK()
»”’// Reloj del registro de ENTRADA for(int bit=0; bit ¡3; bit++) // digitalWrite(SRCLK, digitalSignal[bit]);
»”’//**********
void signalRCLK()
»”’// Reloj del registro de SALIDA for(int bit=0; bit ¡3; bit++) // digitalWrite(RCLK, digitalSignal[bit]);