

# **Examen 1**

Primer parcial - Informatica II

**JOSE MIGUEL GOMEZ MONSALVE  
DANIELA ROSA VILLADIEGO  
PADILLA.**

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

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Análisis del problema</b>	<b>3</b>
2.1. Solución propuesta . . . . .	3
2.2. Componentes Utilizados . . . . .	3
2.3. Esquema de trabajo . . . . .	4
<b>3. Algoritmo Implementado</b>	<b>5</b>
<b>4. Problemas Presentados</b>	<b>12</b>
<b>5. Evolución del algoritmo</b>	<b>13</b>

## **1. Introducción**

El siguiente trabajo tiene como función implementar el uso de matrices de leds para imprimir un mensaje en la fachada de un negocio.

## 2. Análisis del problema

A través de este trabajo se pondrá a prueba la capacidad de solución de problemas, donde se trabajará con componentes de Arduino y se integrará con programación en C++.

Para realizar el trabajo requerido se tendrá una matriz de led's, las cuales van conectadas entre sí y a su vez a la board, donde se hará uso de 2 integrados 74hc595, esto con el fin de poder controlar los 64 leds conectados al arduino por los pines 2, 3 y 4.

### 2.1. Solución propuesta

Se plantea una solución en 3 ciclos. **Planeación, Ejecución, Verificación**

#### **Paso 1 (Planeación):**

Se analiza el problema propuesto y se deducen los componentes a utilizar, adicional se busca el datasheet del integrado 74HC595 para ir bosquejando un circuito.

#### **Paso 2 (Ejecución):**

En esta fase ya se empezó a integrar los componentes al proyecto, lo primero que se hizo fue realizar una conexión de leds de 8 filas por 8 columnas, situando uno debajo de otro, esto con el fin de poder formar una matriz de leds para luego imprimir caracteres en ella. Se hicieron las conexiones pertinentes y se implementaron 2 integrados. Luego de tener los componentes establecidos se procedió a programar el Arduino. En esta etapa de programación se configuraron las funciones que solicitaba el documento, las cuales son **verificar** y **public**, adicional a estas se crearon otras funciones para dar funcionamiento al programa.

#### **Paso 3 (Verificación):**

Ya por último se ejecutó el programa verificando que todo estuviera en orden y haciendo las llamadas pruebas de escritorio, con el fin de garantizar el funcionamiento óptimo del programa, también se hizo una verificación de componentes conectados.

### 2.2. Componentes Utilizados

- Arduino Uno R3
- Placa de pruebas pequeña
- 2 Componentes 74HC595 (Registro de desplazamiento 8 bits)
- 64 Led (RED)
- Cable de conexión (cobre)
- 8 resistencias de 560ohm.

- 1 Placa de pruebas

### 2.3. Esquema de trabajo

Se divide el trabajo para una mejor optimización del tiempo y se asignan unas tareas a realizar por cada día

- Día 1: Se empiezan a realizar conexiones de arduino, buscando una semejanza al trabajo realizado por:**Jose Miguel Gomez**
- Día 2: Se conectan todos los 64 led's, unidos en forma de matriz de 8 filas por 8 columnas por:**Jose Miguel Gomez**
- Día 3: Se culmina la conexión del arduino con todos sus componentes y se implementa el código para la función verificar por:**Jose Miguel Gomez**
- Día 4: Se programan las funciones imagen y publik por:**Daniela Rosa Villadiego**
- Día 5: Se hacen mejoras en el código y sus funciones, con el fin de corregir algunos errores por:**Daniela Rosa Villadiego**
- Día 6: Se siguen corrigiendo partes del código con el fin de garantizar su funcionamiento por:**Daniela Rosa Villadiego, Jose Miguel Gomez**

### 3. Algoritmo Implementado

Se implemento un algoritmo computacional, este fue desarrollado en C++ y compuesto por tres partes fundamentales; **Input, Proceso y Output**

**Input:**El ingreso de los datos que el algoritmo necesita para operar.

**Proceso:**la operación lógica que el algoritmo realizará con lo datos recibidos del input

**Output:**Los resultados obtenidos del proceso sobre el input, una vez terminada la ejecución del algoritmo

```
#define DATA 2 //Definicion de pin de datos
#define SHIFT 3 //Definicion de pin de cambio de registro
#define STORE 4 //Definicion de pin de almacenamiento

// PROGRAMA

// Declaracion de funciones y variables.
int potiValue;
const int fila[8] = {2,7,19,5,13,18,12,16}; // Vector para las filas
const int columna[8] = {6,11,10,3,17,4,8,9}; //Vector para las columnas
void imagen(char, int);
void encenderLed(int);
void verificacion();
int opcion = 0;
int contadorTiempo=0;
int VelAnimacion=0;
int Numpatron;
char patrones []={};
char patron;
char times []={};
int time;

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

int bitA [] = {0,0,0,0,0,0,0,0};
```

```

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

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

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

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

void setup(){
    pinMode(DATA, OUTPUT);

```

```

pinMode(SHIFT, OUTPUT);
pinMode(STORE, OUTPUT);

    for(int i=0; i<8; i++){ //inicializar los pines como salidas con los vec

pinMode(columna[i], OUTPUT);
pinMode(fila[i], OUTPUT);
Serial.begin(9600);

digitalWrite(columna[i], HIGH);    // matriz catodo comun, as que ponemos

    }

Serial.println(" _MANUAL_DE_USO.");
Serial.println(" El _presente _manual _esta _organizado _de _la _siguiente _manera:");
Serial.println(" _");
Serial.println(" 1. _Funcion _de _verificacion");
Serial.println(" 2. _Funcion _imagen");
Serial.println(" 3. _Funcion _publik.");
Serial.println("_____");
Serial.println(" 1. _Funcion _de _verificacion");
Serial.println(" En _esta _funcion _usted _podra _comprobar _que _todos _los _leds _de _");
Serial.println("_____");
Serial.println(" Presiona _1 _para _continuar _");
    while(Serial.available()==0){}
        Serial.println(" _");
Serial.println(" 2. _Funcion _imagen.");
Serial.println(" Esta _funcion _le _permitira _mostrar _un _patron _en _la _matriz _de _");
Serial.println("_____");
Serial.println(" 3. _Funcion _publik");
Serial.println(" Con _esta _funcion _se _le _permitira _mostrar _una _secuencia _de _ca");
Serial.println(" _");

//memoria dinamica

char *pentrada= new char[11];
char valor[11]={ 'B', 'I', 'E', 'N', 'V', 'E', 'N', 'I', 'D', 'O', '.' };

for(int k=0; k<11; k++){
    *(pentrada+k)= valor[k];
}

```



```

    for (int k=0; k<11;k++){
        Serial.print(*(pentrada+k));
    }

    delete [] pentrada;

Serial.println("Cual de las opciones del sistema desea ejecutar?");
Serial.println("");
Serial.println(" (1). Funcion de verificacion");
Serial.println(" (2). Funcion imagen");
Serial.println(" (3). Funcion publik");


    verificacion();

} // cierre del setup


void loop()
{

    if(Serial.available()>0){
        opcion = Serial.read();

        if(opcion>48 && opcion<=51){

            if(opcion == 49){
                verificacion();
            }

            else if(opcion == 50){

                Serial.print(" Ingrese un solo caracter:");
                //Serial.println(patron);
                while(!Serial.available()){ }
                if(Serial.available()){
                    patron = (char)Serial.read();
                }
            }
        }
    }
}

```

```

        imagen(patron);
    }

}

else if(opcion == 51){
    Serial.print("Ingrese el numero de patrones: ");
    Numpatron = Serial.read();
    for(int i=0; i< Numpatron; i++){
        Serial.print("Ingrese patron: ");
        patron= Serial.read();
        patrones[i]= patron;
        Serial.print("Ingrese el tiempo de duracion para ese patron: ");
        time= Serial.read();
        times[i]= time;
        //publik(patrones, times, Numpatron );
    }

}

else{
    Serial.println("Opciones");
    Serial.println("(1). Funcion de verificacion (2). Funcion imagen (3). Funcion");
    opcion = Serial.read();
}

}

}

} // cierre del Loop

```

*// PROGRAMANDO LOS LED'S*

```

void encenderLed(int arreglo[8]){

    for (int i=0; i<8; i++){

        //Serial.println((int)pow(2,7+i)+1);
        shiftOut(DATA, SHIFT, LSBFIRST, ~arreglo[i]);
    }
}

```

```

        shiftOut(DATA, SHIFT, LSBFIRST, 128 >> i);
        unstore();
        store();
    }
}

void imagen(int a){ //funcion para mostrar un patron ingresado por el usuario

    if(a==65){
        for(int i =0; i<8; i++){
            for(int j =0; j<8; j++){
                if(A[i][j]==1){
                    bitA[i] += (int)pow(2,j);
                }
            }
        }
        encenderLed(bitA);
    }

}

/*void publik(char a[], char b[], int num){ // recibe patrones[], times[] y Num

    contadorTiempo++;

    for(int k=0; k< num; k++){

        while (contadorTiempo < b[k]){
            char Encender= a[k];
            encenderLed(Encender);

        }

        contadorTiempo=0;

    }

}*/

```

```

void verificacion ()
{
    shiftOut (DATA, SHIFT, MSBFIRST, 0);
    shiftOut (DATA, SHIFT, MSBFIRST, 255);
    unstore ();
    store ();
    shiftOut (DATA, SHIFT, MSBFIRST, 0);
    unstore ();
    store ();
}

```

```

void store ()
{
    digitalWrite (STORE, LOW);
    delayMicroseconds (100000);
}

```

```

void unstore ()
{
    digitalWrite (STORE, HIGH);
    delayMicroseconds (100000);
}

```

## 4. Problemas Presentados

Al momento de montar el circuito y hacer la verificación de que en la matriz prendían todos sus led's correctamente, los integrados explotaban pues al principio se pensó que no era necesario conectar resistencias.

Al intentar compartir el circuito en tinkercad con los otros dos compañeros para seguir editandolo todos juntos ¿Qué pasó? que al parecer el tinkercad no habido guardado los cambios y al momento de ponerlo en dominio público para compartir el circuito, borró todas las conexiones y con ello el código, mostrando solo los componentes, por lo que tocó empezar de cero nuevamente.

## 5. Evolución del algoritmo

```
#define DATA 2 //Definicion de pin de datos
#define SHIFT 3 //Definicion de pin de cambio de registro
#define STORE 4 //Definicion de pin de almacenamiento

// PROGRAMA

// Declaracion de funciones y variables.
    int potiValue;
    const int fila[8] = {2,7,19,5,13,18,12,16}; // Vector para las filas
    const int column[8] = {6,11,10,3,17,4,8,9}; //Vector para las columnas
    void imagen(char, int);
    void encenderLed(int);
    void verificacion();
    int opcion = 0;
    int contadorTiempo=0;
    int VelAnimacion=0;
    char LetrasNumeros[] = {'A','B','C','D','E','F','G','H','I','J','K','L','M','N'};
int Numpatron;
    char patrones []={};
    char patron;
    char times []={};
    int time;
int A[8][8]= {
    {1,1,0,0,0,0,1,1},
    {1,0,0,0,0,0,0,1},
    {0,0,0,0,0,0,0,0},
    {0,0,0,1,1,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0},
    {0,0,0,1,1,0,0,0},
    {0,0,0,1,1,0,0,0},
};

int bitA [] = {0,0,0,0,0,0,0,0};

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

```

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

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

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

int U[8][8]= { {1,0,0,1,1,0,0,1},
                                {1,0,0,1,1,0,0,1},
                                {1,0,0,1,1,0,0,1},
                                {1,0,0,1,1,0,0,1},
                                {1,0,0,1,1,0,0,1},
                                {1,0,0,1,1,0,0,1},
                                {1,0,0,0,0,0,0,1},
                                {1,0,0,0,0,0,0,1},
                                };
//t LetrasMatriz[5][8][8];
//trasMatriz[0]=A;

void setup(){
    pinMode(DATA, OUTPUT);
    pinMode(SHIFT, OUTPUT);
    pinMode(STORE, OUTPUT);

```

```

        for(int i=0; i<8; i++){ //inicializar los pines como salidas con los vec

pinMode(columna[i], OUTPUT);
pinMode(fila[i], OUTPUT);
Serial.begin(9600);

digitalWrite(columna[i], HIGH);    // matriz catodo comun, as que ponemos

    }
Serial.println(" _MANUAL_DE_USO.");
Serial.println(" _bla_bla_bla.");
Serial.println(" Opciones_del_sistema.");
Serial.println(" (1). _Funcion_de_verificacion_(2). _Funcion_imagen_(3). _Funcion_pu

//cion = Serial.read();

    verificacion();

} // cierre del setup

void loop()
{

    if(Serial.available()>0){
        opcion = Serial.read();
        //rificacion();
        //rial.println(opcion);
        //rificacion();

        if(opcion>48 && opcion<=51){

            if(opcion == 49){
                verificacion();
            }

            else if(opcion == 50){
                /*for(int i =0; i<8; i++){
                    for(int j =0; j<8; j++){
                        if(A[i][j]==1){
                            bitA[i] += pow(2,j);

```



```

        Serial.println(pow(2,j));
        Serial.println(bitA[i]);
    }
}
Serial.println(bitA[i]);
}*/
for(int k=0; k<8; k++){
    bitA[k]=(int)pow(2,k);
}
encenderLed(bitA);
//Serial.print(" Ingrese un solo caracter: ");
//patron = Serial.read();
//imagen(patron);

}

else if(opcion == 51){
    Serial.print(" Ingrese el numero de patrones: ");
    Numpatron = Serial.read();
    for(int i=0; i< Numpatron; i++){
        Serial.print(" Ingrese patron: ");
        patron= Serial.read();
        patrones[i]= patron;
        Serial.print(" Ingrese el tiempo de duracion para ese patron: ");
        time= Serial.read();
        times[i]= time;
        //publik(patrones, times, Numpatron );
    }

}

else{
    Serial.println(" Opciones");
    Serial.println(" (1). Funcion de verificacion (2). Funcion imagen (3). Funcion");
    opcion = Serial.read();
}

}

}

} // cierre del Loop

// PROGRAMANDO LOS LED'S

```

```

void encenderLed(int arreglo[8]){
    for (int i=0; i<8; i++){
        Serial.println((int)pow(2,7+i)+1);
        shiftOut(DATA, SHIFT, LSBFIRST, ~arreglo[i]);
        shiftOut(DATA, SHIFT, LSBFIRST, 128 >> i);
    }
    store();
}

```

```

void imagen(int a){ //funcion para mostrar un patron ingresado por el usuario

```

```

    for(int i =0; i<8; i++){
        for(int j =0; j<8; j++){
            if(A[i][j]==1){
                bitA[i] += pow(2,j);
            }
        }
    }

    if(a==65){
        encenderLed(bitA);
    }

}

```

```

/*void publik(char a[], char b[], int num){ // recibe patrones[], times[] y Num

```

```

    contadorTiempo++;

```

```

    for(int k=0; k< num; k++){

```

```

        while (contadorTiempo < b[k]){
            char Encender= a[k];
            encenderLed(Encender);

```

```

        }

```

```

    contadorTiempo=0;

```

```

}

```

```
}*/
```

```
void verificacion ()  
{  
    shiftOut (DATA, SHIFT, MSBFIRST, 0);  
    shiftOut (DATA, SHIFT, MSBFIRST, 255);  
    store ();  
    shiftOut (DATA, SHIFT, MSBFIRST, 0);  
    store ();  
}
```

```
void store ()  
{  
    digitalWrite (STORE, HIGH);  
    delayMicroseconds (100);  
    digitalWrite (STORE, LOW);  
    delayMicroseconds (10000);  
}
```