

Parcial 1

Juan Pablo Cárdenas Sanabria
Mario Andres Leal Galvis
John Edison Chamorro Coral

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

Índice

1. Análisis del problema y consideraciones para la alternativa de solución propuesta.	2
2. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo.	2
3. Algoritmo implementado.	3
4. Problemas de desarrollo que se presentaron.	9
4.1. Problema 1:	9
4.2. Problema 2:	9
4.3. Problema 3:	10
4.4. Problema 4:	10
4.5. Problema 5:	10
4.6. Problema 6:	10
5. Evolución del algoritmo y consideraciones a tener en cuenta en la implementación.	10
6. Inclusión de imágenes	11

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

La problemática que tenemos es poder desplegar patrones en una matriz 8x8 leds, para poder llegar a una solución mediante el controlador de arduino y sumado a la limitante de salidas digitales se utiliza el integrado 74HC595, lo cual nos permite controlar una cantidad de 8 led por integrado, resultando así en la utilización de 8 integrados 74HC595 para así dar el total de 64 led, lo que corresponde a la matriz de leds 8x8.

Adicionalmente centramos el problema en cómo enviar la cadena de bit para reenterar cada patrón, la solución que planteamos en el principio fue tomar el primer bit de una cadena de caracteres formadas por ceros y unos y ubicarlo en la última posición de la matriz, esto se daría por medio de que el usuario ingrese el patrón que desea ver por medio de bits ya sea escribiéndolo uno por uno, o ingresando unos patrones ya predeterminados.

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

Como tarea fundamental se pacta el de cumplir con los requerimientos puestos en el parcial.

- En el desarrollo del programa se toma como tarea prioritaria implementar el uso de funciones, para así automatizar el Código de forma que se ahorren líneas de código y así de esa forma evitar problemas de uso de memoria.
- Se trabaja en la construcción del circuito usando la plataforma Tinkercad de forma que se consiga un orden con los componentes y crear una parte física y visual para el programa.
- Se buscan formas de optimizar los pines digitales del arduino de una forma que se cumpla con los requerimientos dados en las indicaciones y de una forma que resulta sencilla y cómoda de trabajar.
- Se investiga acerca del funcionamiento del circuito integrado 74HC595 para así poder trabajar con el de una forma correcta y adecuada.
- Se forma el código que regirá el funcionamiento y la parte operativa de los datos que el usuario ingrese.


```

    Serial.begin(9600);

}
//-----
void loop()
{
    //caracter2(cinco);

    while(state==0)
    {
        state=1;
        Serial.println("QueIn desea realizar :\n");
        Serial.println("1) verificar funioncionamiento de los leds \n2) Ingresar pat

    while(state==1)
    {
        if(Serial.available()>0)
        {
            char opc;

            while(Serial.available()>0)
            {
                opc= Serial.read();

            }
            if(opc=='1')
            {

                state=2;

            }
            else if(opc=='2')
            {
                state=3;

            }
            else if(opc=='3')
            {
                state=4;

            }
            else {state=0;}
        }
    }
    while(state==2)
    {

```

```

        Serial.println("\nVerificacion de los leds\n");
        verificacion(ver);
        state=0;
    }
    while(state==3)
    {
        Serial.println("\nIngrese el patron a observar segun el manual: \n");
        state=5;
        String bits;
        bool ban=false;
        while(state==5)
        {

            ban=recepcion(bits);
            if(bits.length()>=64)
            {
                state=3;
            }
        }

        str2car2(bits);

        state=0;
    }
    while(state==4)
    {
        Serial.println("\nIngrese los caracteres que desea observar\n");
        state=6;
        bool datafull=false;
        String secuencia="";
        String tiempoD="";

        while(state==6)
        {

            datafull=recepcion(secuencia);
            if(datafull==true)
            {
                state=4;
            }
            delay(500);
        }
        Serial.println("\nIngrese el tiempo (milesegundos) entre cada patron\n");
        state=7;
        datafull=false;
        while(state==7)

```

```

        {
            datafull=recepcion(tiempoD);
            if (datafull==true)
            {
                state=4;
            }
            delay(500);
        }

        publik(sequencia , tiempoD);
        state=0;
    }
}

//-----
void str2car2(String vector)
{
    int vecInt[64];
    for(int i=0;i<64;i++)
    {
        vecInt[i]=(vector[i]-48);
    }
    caracter2(vecInt);
}

//-----
void publik(String sequencia ,String tiempo)
{
    int t=tiempo.toInt();
    char nombre;
    for(int i=0;i<sequencia.length();i++)
    {

        Serial.println(sequencia[i]);
        nombre=sequencia[i];
        findVector(nombre);
        delay(t);
    }
}

//-----
void caracter2(int car1[])
{
    Serial.println("Desplegando caracter en matriz\n");
    int j=0;

```

```

    for (int i=0;i<64;i++)
    {
        digitalWrite (pinSerial ,*( car1+i ));
        SCLK();j++;
        if (j==8){j=0;RCLK();}
    }
}
//-----
bool recepcion (String &secuencia)
{
    bool bandera=false;
    if (Serial.available()>0)
    {
        while (Serial.available()>0)
        {
            secuencia+= (char) Serial.read();
        }

        bandera=true;
    }
    return bandera;
}

//-----
void verificacion (int ver []){
    caracter2 (ver);
}
void findVector (char nombre)
{
    int *ptr;
    if (nombre=='a' || nombre=='A')
    {
        ptr=&A[0];
    }
    else if (nombre=='b' || nombre=='B')
    {
        ptr=&B[0];
    }
    else if (nombre=='c' || nombre=='C')
    {
        ptr=&C[0];
    }
    else if (nombre=='d' || nombre=='D')

```



```

    {
        ptr=&D[0];
    }
    else if (nombre=='e' || nombre=='E')
    {
        ptr=&E[0];
    }
    else if (nombre=='f' || nombre=='F')
    {
        ptr=&F[0];
    }
    else if (nombre=='g' || nombre=='G')
    {
        ptr=&G[0];
    }
    else if (nombre=='h' || nombre=='H')
    {
        ptr=&H[0];
    }
    else if (nombre=='i' || nombre=='I')
    {
        ptr=&I[0];
    }
}

    caracter2(ptr);
}
//-----
//FUNCION PARA ENVIAR DATOS A SCLK
void SCLK(){

    delay(20);
    digitalWrite(pSclk, LOW);
    delay(20);
    digitalWrite(pSclk, HIGH);
}
//FUNCION PARA ENVIAR DATOS A RCLK
void RCLK(){

    digitalWrite(prclk,LOW);
    delay(10);
    digitalWrite(prclk,HIGH);
    delay(10);
    digitalWrite(prclk,LOW);
}

```

4. Problemas de desarrollo que se presentaron.

A medida del desarrollo del algoritmo se presentaron una serie de problemas que requerían su pronta solución para el correcto funcionamiento del programa, los siguientes son algunos de ellos.

4.1. Problema 1:

En la consola de Arduino , dentro del loop, se genera un ciclo de impresión sobre las opciones que tiene el usuario para la manipulación del programa, además al ingresar al switch-case este se sale automáticamente de dicha estructura sin antes haber terminado la ejecución que allí se propone.

Solución:

Dentro de las posibles soluciones, está trabajar con estados que son ideales para este tipo de microprocesadores y permiten pausar el proceso hasta que se complete la acción requerida. Se realizan en total 8 estados iniciando desde el estado cero en el cual se despliega la información general del usuario, es decir las tres opciones que tiene el programa a realizar, posteriormente se ingresa al estado 1 donde se espera que el usuario ingrese la opción deseada por consola y así determinar a cuál de los otros 3 estados dirigirse para realizar la petición requerida. Los estados 5, 6 ,7 y 8 estados se emplean para la toma de datos.

En el estado 2 se realiza la verificación de todos los leds En el estado 3 se realiza la petición y despliegue del patrón deseado por el usuario En el estado 4 se realiza el despliegue de los caracteres por defecto en el tiempo que se desean ver.

En el estado 5 , 6 , 7 y 8 se toman los datos del patrón cualquiera ingresado por el usuario, Fue necesario emplearlo para que se puedan guardar los datos sin salirse del ciclo antes de tiempo. Dentro de dichos estados fue necesario uso de una bandera que me determine si se completó los datos ingresados y además un delay para darle tiempo al sistema que guarde todo lo ingresado.

Punto a tener en cuenta, en el estado 4 debido a que se necesita saber que patrones por defecto requiere el usuario, se creó una función la cual determina el carácter y así emplear un puntero al vector que representa dicho carácter para desplegarlo en la matriz.

4.2. Problema 2:

En el numeral donde se solicitaba que el usuario ingresara el patrón que deseaba observar en la matriz, se propuso usar una variable tipo string para que guarde todos los datos ingresados por el usuario. El problema surge a la hora de desplegar en pantalla pues la función que realiza dicha acción solo emplea variables tipo int.

Solución:

Se propone realizar una función la cual se encarga de tomar dato a dato de la variable string e ir almacenando en un vector tipo int para posteriormente desplegarlo en la función caracter 2.

4.3. Problema 3:

Dentro de la función publik se necesita de alguna manera asociar los vectores representativos de cada carácter con los patrones que el usuario desea observar.

Solucion:

Se creó una función que por medio de un condicional busca el vector respectivo y lo grafica en la matriz.

4.4. Problema 4:

Dentro de la función publik se necesita de alguna manera asociar los vectores representativos de cada carácter con los patrones que el usuario desea observar.

Solucion:

Se creó una función que por medio de un condicional busca el vector respectivo y lo grafica en la matriz.

4.5. Problema 5:

Se presenta una falla con el circuito integrado 74HC595 en donde el dispositivo se rompía debido a que la corriente a través del pasador Q5 es 56.7mA, mientras que el máximo es 25.0mA. La corriente total a través de los pines de corriente es 137mA, mientras que el máximo es 50.0mA.” **Solucion:**

Se creó una función que por medio de un condicional busca el vector respectivo y lo grafica en la matriz.

4.6. Problema 6:

Se presenta una sobrecorriente en el chip de la tercera fila de leds, debido a un cambio abrupto de voltajes por requerimientos de despliegue de bits.”

Solucion:

Se decido reestructurar el codgio de forma que no se sobrecargara el mismo.

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

Como primera consideración, se debe tener paciencia con el tiempo de ejecución del programa ya que debido a la optimización de la página Tinkercad el procesamiento de la información se hace de una forma lenta y para anda optima lo cual no está para nada relacionado con la forma en que se hizo el programa ni con su implementación.

Al inicio consideramos trabajar el codigo con algoritmos a fuerza bruta implementando funciones, pero luego al ver que tinkercad no recibia tantas lineas de codigo decidimos comenzar a optimizarlo metiendole matrices con puntero que las recorran, se implementó sin ningún condicional con lo que género, que se

imprimieran en pantalla constantemente y a su vez no se tomaran las variables según lo deseado, por lo que se implementó trabajar con `while()` lo que permitía estar en un estado de pausa mientras se cumpliera las condiciones requeridas. Para poder cumplir los requerimientos se implementaron 7 estados. En el estado cero que abarcaba todos los demás estados se implementó la impresión en pantalla, pero luego el código no funcionaba correctamente por lo cual optamos por dejar con arreglos y punteros sin recorrer tantas líneas de código y ayudaba la optimización del código.

6. Inclusión de imágenes

En la Figura (1), se muestra el circuito usado para el parcial.

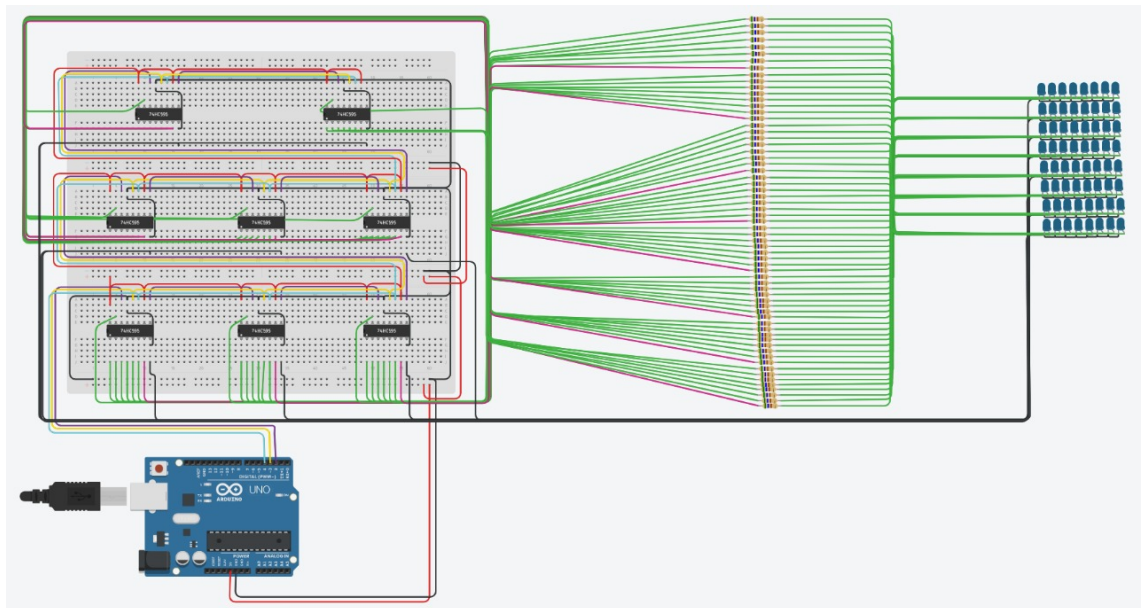


Figura 1: Circuito Tinkercad