

CONTROL DE MATRIZ 8x8 CON ARDUINO

Juan Salvador Molina Martín

1.Introducción.

2.Desarrollo.

2.1.Lista de componentes.

2.2.PCB.

2.3.Librerías.

3.Código del sistema del ARDUINO.

1.Introducción.

Consiste en la simulación, programación, test y desarrollo de la PCB para el manejo de una matriz LED 8x8 con el circuito integrado 74HC595 que nos permite controlar las 8 columnas (o filas dependiendo del tipo de matriz y del uso que le demos) con solo 3 pines.

2.Desarrollo.

El esquemático del circuito sería.



2.1.Lista de componentes.



2.2.PCB.



2.3.Librerías.

El único componente del que no teníamos librería era la matriz de led. El profesor me facilitó una librería de la misma y se puede descargar del siguiente enlace: [matrix8x8.lbr](#)

Las otras librerías que hemos usado son 74xx-eu para el 74HC595, SparkFun-Boards para el Arduino y resistor para las resistencias.

3.Código del sistema del ARDUINO.

```
int pinLatch = 6; //Pin para el latch de los 74CH595

int pinDatos = 7; //Pin para Datos serie del 74CH595

int pinReloj = 5; //Pin para reloj del 74CH595

int letra = 0; //Variable para cada letra

int ciclo = 0; //Variable para los ciclos de cada letra en cada posicion

int desplaza = 0; //Variable para generar desplazamiento en las filas

void imprimeRojo();

//Definimos los numeros decimales que hacen falta para dibujar cada caracter

#define SP {8, 20, 34, 62, 65} //Espacio

#define EX {0, 125, 0, 0, 0} //Exclamacion !

#define A {31, 36, 68, 36, 31} // xxxxx000 31

#define B {127, 73, 73, 73, 54} // xxxxxxx0 127

#define C {62, 65, 65, 65, 34}

#define D {127, 65, 65, 34, 28}

#define E {127, 73, 73, 65, 65}

#define F {127, 72, 72, 72, 64}

#define G {62, 65, 65, 69, 38}

#define H {127, 8, 8, 8, 127}

#define I {0, 65, 127, 65, 0}

#define J {2, 1, 1, 1, 126}

#define K {127, 8, 20, 34, 65}

#define L {127, 1, 1, 1, 1}

#define M {127, 32, 16, 32, 127}

#define N {127, 32, 16, 8, 127}

#define O {62, 65, 65, 65, 62}

#define P {127, 72, 72, 72, 48}

#define Q {62, 65, 69, 66, 61}

#define R {127, 72, 76, 74, 49}
```

```
#define S {50, 73, 73, 73, 38}

#define T {64, 64, 127, 64, 64}

#define U {126, 1, 1, 1, 126}

#define V {124, 2, 1, 2, 124}

#define W {126, 1, 6, 1, 126}

#define X {99, 20, 8, 20, 99}

#define Y {96, 16, 15, 16, 96}

#define Z {67, 69, 73, 81, 97}

#define uno {0, 65, 127, 1, 0}

#define dos {33, 67, 69, 65, 49}

#define tres {34, 65, 73, 73, 54}

#define cuatro {24, 36, 69, 127, 5}

#define cinco {114, 82, 82, 76, 0}

#define seis {62, 73, 73, 73, 70}

#define siete {65, 66, 68, 72, 112}

#define ocho {54, 73, 73, 73, 54}

#define nueve {48, 73, 73, 73, 62}

#define cero {62, 69, 73, 81, 62}

//Escribimos la frase separando cada letra por comas

//En el primer numero lo adaptaremos la longitud de la frase (caracteres)

//byte frase[9][6]={uno,dos,tres,cuatro,cinco,seis,siete,ocho,nueve};

byte frase[4][6]={A, B, C, D};

//Almacenamos los pines de las filas que van conectadas a los cátodos

int gnd[13]={0,0,0,0,0,4,2,8,3,12,9,11,10};

//Configuramos la placa

void setup()

{ //Ponemos del pin 2 al 12 como salidas

for (int i=2;i<=12; i++)
```

```
{  
    pinMode(i, OUTPUT);  
}
```

//Ponemos a nivel alto todas las lineas de los cátodos de la matriz

//De esta forma inicialmente están apagados los LED

for (int g=2; g<=9; g++)

```
{  
    digitalWrite(g, HIGH);  
}
```

}

void loop()

{

for (desplaza = 9; desplaza>=0; desplaza--){

```
for (ciclo=0; ciclo<=35; ciclo++){  
    imprimeRojo();  
}
```

} //Una vez ha mostrado una letra, sumamos uno para que salga la siguiente

letra++;

//Cuando ha llegado al final de la frase, lo pone a cero para que vuelva a salir

//Si cambiamos la longitud de la frase, este valor hay que cambiarlo

if(letra == 4)

{

```
letra = 0;
```

```
}  
  
}  
  
//Funcion que imprime en color rojo  
  
void imprimeRojo(){  
  
    //Un contador del tamaño de las letras (5 lineas de ancho)  
  
    for (int z=0; z<=5; z++)  
  
        {  
            int col = z + desplaza; //Le  
            decimos en que linea empieza a dibujar  
  
            digitalWrite(gnd[col], LOW); //La  
            ponemos a cero  
  
            digitalWrite(pinLatch, LOW); //Le  
            decimos a los registros que empiecen a escuchar los datos  
  
            shiftOut(pinDatos, pinReloj, MSBFIRST, 0); //Le  
            decimos que en el ultimo registro no encienda nada  
  
            shiftOut(pinDatos, pinReloj, MSBFIRST, frase[letra][z]); //Le  
            decimos que imprima la línea z de la letra en el primer registro (rojo)  
  
            digitalWrite(pinLatch, HIGH); //Guarda y  
            graba las salidas en los registros al poner a 1 el latch  
  
            digitalWrite(gnd[col], HIGH); //Apagamos  
            esa fila poniendola en alto  
  
        }  
  
}
```