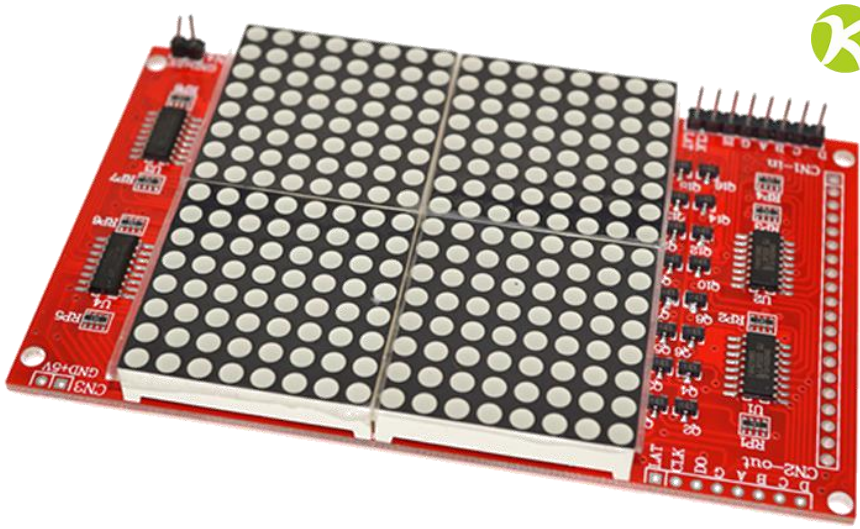
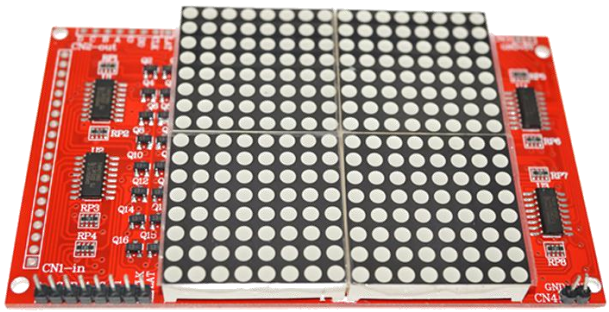
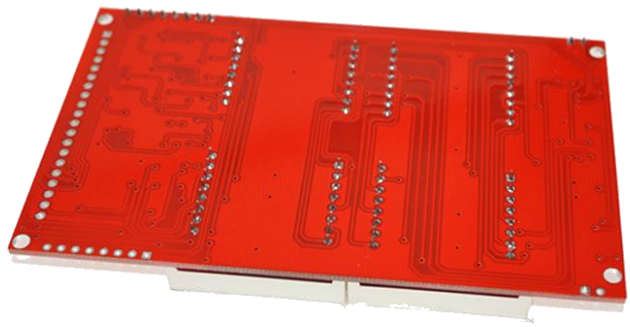
OKY3525-1: Controlador Para Matriz LED 16x16



Descripción:

Modulo de control para matriz LED color rojo, cátodo común. Utiliza los C.I. 74HC138 para controlar filas y 74HC595 para las columnas, decodifica y controla 4 matrices LED 8x8, puede combinarse para crear pantallas más grandes, compatible con una amplia gama de microcontroladores.





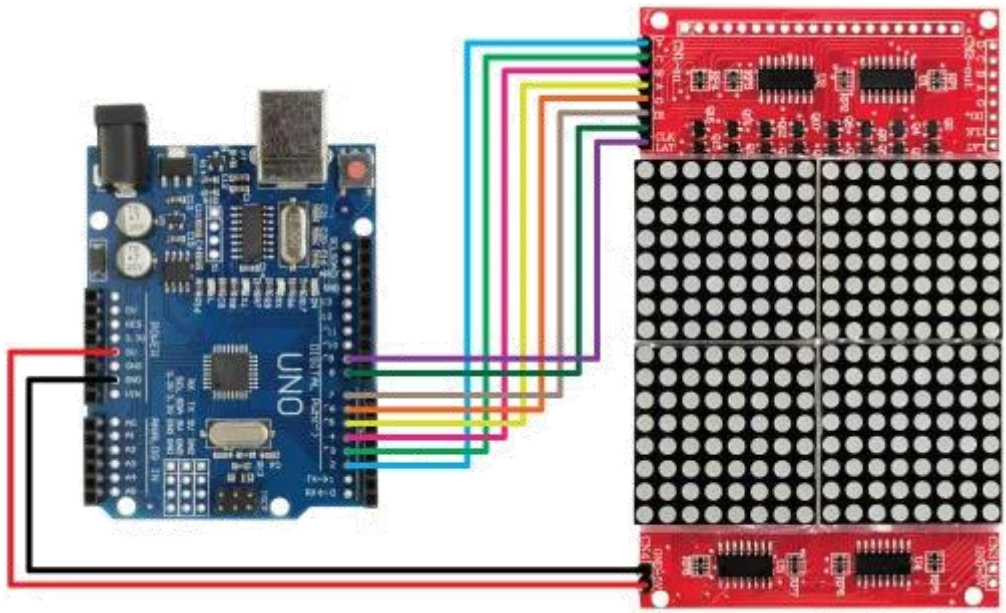
Características:

- Circuito de control: 74HC595 Y 74HC138
- Matriz LED 8x8 color rojo
- Compatible con Arduino, Raspberry Pi, Microcontroladores PIC, etc.
- Soporta múltiples módulos en cascada
- Dimensiones: aprox. 113mm x 64mm x 12mm

Especificaciones:

Voltaje típico	**5V**
Voltaje de trabajo	**4.7~5V**

Diagrama de conexión:



Código ejemplo:

```
**********************************************************************

#include <Arduino.h>

#define LEDARRAY_D 2
#define LEDARRAY_C 3
#define LEDARRAY_B 4
#define LEDARRAY_A 5
#define LEDARRAY_G 6
#define LEDARRAY_DI 7
#define LEDARRAY_CLK 8
#define LEDARRAY_LAT 9
#define led 13
#define Num_Word 1



unsigned char Display_Buffer[2];
unsigned char Display_Swap_Buffer[Num_Word][32]={0};
bool Shift_Bit = 0;
bool Flag_Shift = 0;
unsigned char Timer0_Count = 0;
unsigned char temp = 0x80;
unsigned char Shift_Count = 0;
const unsigned char Word[1][32] =

{
0xFF,0x86,0xF6,0xF6,0x86,0xBF,0xBC,0xBD,0x85,0xF5,0xF4,0xF7,0xF7,0xF7,0xD7,0xEC,
0xFF,0x07,0xF7,0xF7,0x07,0xBF,0x03,0xBB,0xBB,0xBB,0x03,0xBF,0xB7,0xBB,0x81,0x3B,
};

void setup()

{

    pinMode(LEDARRAY_D, OUTPUT);
    pinMode(LEDARRAY_C, OUTPUT);
    pinMode(LEDARRAY_B, OUTPUT);
    pinMode(LEDARRAY_A, OUTPUT);
    pinMode(LEDARRAY_G, OUTPUT);
    pinMode(LEDARRAY_DI, OUTPUT);
    pinMode(LEDARRAY_CLK, OUTPUT);
    pinMode(LEDARRAY_LAT, OUTPUT);



    Clear_Display();
}

    void loop()
{
    unsigned int i;
```

```
	for(i = 0 ; i < 30; i ++)
	{
		Display(Display_Swap_Buffer);
			}

	Calc_Shift();
	Shift_Count++;
	if(Shift_Count == 32)

	{
	Shift_Count = 0;
	}
}

//*************************************************************
//*************************************************************

void Clear_Display()

{
	unsigned char i,j;
	for(j = 0 ; j < Num_Word; j++)

	{
		for(i = 0 ; i < 32 ;i++)
		{
			Display_Swap_Buffer[j][i] = 0xff;
		}
	}
}

//*************************************************************
//*************************************************************

void Calc_Shift()

{
	unsigned char i;
	for(i = 0;i < 16;i++)

		{
			if((Display_Swap_Buffer[0][16+i]&0x80) == 0)
			{
			Display_Swap_Buffer[0][i] = (Display_Swap_Buffer[0][i] << 1)&0xfe;
			}

			else
			{
				Display_Swap_Buffer[0][i] = (Display_Swap_Buffer[0][i] << 1)|0x01;
			}

			if(Shift_Count < 8)
			{
```

```
                Shift_Bit = Word[0][i]&temp;
            }

                    else if(Shift_Count < 16)
            {
                Shift_Bit = Word[0][16+i]&temp;
            }

            else
                {
                    Shift_Bit = 1;
                }

            if( Shift_Bit == 0)
                {
                    Display_Swap_Buffer[0][16+i] = (Display_Swap_Buffer[0][16+i] << 1)&0xfe;
                }

            else
                {
                    Shift_Bit =1;
                    Display_Swap_Buffer[0][16+i] = (Display_Swap_Buffer[0][16+i] << 1)|0x01;
                }
            }

            temp = (temp>>1)&0x7f;
            if(temp == 0x00)

        {
                temp = 0x80;
        }
    }

//************************************************************
//************************************************************

void Display(unsigned char dat[][32])

{
    unsigned char i;
    for( i = 0 ; i < 16 ; i++ )
    {
        digitalWrite(LEDARRAY_G, HIGH);

        Display_Buffer[0] = dat[0][i];
        Display_Buffer[1] = dat[0][i+16];
        Send(Display_Buffer[1]);
        Send(Display_Buffer[0]);
        digitalWrite(LEDARRAY_LAT, HIGH);
        delayMicroseconds(1);
        digitalWrite(LEDARRAY_LAT, LOW);
        delayMicroseconds(1);
        Scan_Line(i);
        digitalWrite(LEDARRAY_G, LOW);
```

```
            delayMicroseconds(300);;
        }
}

//*************************************************
//*************************************************

void Scan_Line( unsigned char m)
{
      switch(m)
      {
            case 0:
digitalWrite(LEDARRAY_D,LOW);digitalWrite(LEDARRAY_C,LOW);digitalWrite(LEDARRAY_B,LOW);digitalWrite(LEDARRAY_A,     LOW);
break;

            case 1:
digitalWrite(LEDARRAY_D, LOW);digitalWrite(LEDARRAY_C, LOW);digitalWrite(LEDARRAY_B, LOW);digitalWrite(LEDARRAY_A, HIGH);
break;

            case 2:
digitalWrite(LEDARRAY_D, LOW);digitalWrite(LEDARRAY_C, LOW);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, LOW);
break;

            case 3:
digitalWrite(LEDARRAY_D,LOW);digitalWrite(LEDARRAY_C, LOW);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, HIGH);
break;

            case 4:
digitalWrite(LEDARRAY_D, LOW);digitalWrite(LEDARRAY_C, HIGH);digitalWrite(LEDARRAY_B, LOW);digitalWrite(LEDARRAY_A, LOW);
break;

            case 5:
digitalWrite(LEDARRAY_D,LOW);digitalWrite(LEDARRAY_C, HIGH);digitalWrite(LEDARRAY_B, LOW);digitalWrite(LEDARRAY_A, HIGH);
break;

            case 6:
digitalWrite(LEDARRAY_D,LOW);digitalWrite(LEDARRAY_C, HIGH);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, LOW);
break;

            case 7:
digitalWrite(LEDARRAY_D,LOW);digitalWrite(LEDARRAY_C, HIGH);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, HIGH);
break;

            case 8:
digitalWrite(LEDARRAY_D, HIGH);digitalWrite(LEDARRAY_C, LOW);digitalWrite(LEDARRAY_B, LOW);digitalWrite(LEDARRAY_A, LOW);
break;

            case 9:
digitalWrite(LEDARRAY_D,HIGH);digitalWrite(LEDARRAY_C, LOW);digitalWrite(LEDARRAY_B, LOW);digitalWrite(LEDARRAY_A, HIGH);
break;

            case 10:
digitalWrite(LEDARRAY_D,HIGH);digitalWrite(LEDARRAY_C, LOW);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, LOW);
break;
```

```
            case 11:
digitalWrite(LEDARRAY_D,HIGH);digitalWrite(LEDARRAY_C, LOW);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, HIGH);
break;

            case 12:
digitalWrite(LEDARRAY_D,HIGH);digitalWrite(LEDARRAY_C, HIGH);digitalWrite(LEDARRAY_B, LOW);digitalWrite(LEDARRAY_A, LOW);
break;

            case 13:
digitalWrite(LEDARRAY_D,HIGH);digitalWrite(LEDARRAY_C, HIGH);digitalWrite(LEDARRAY_B, LOW);digitalWrite(LEDARRAY_A, HIGH);
break;

            case 14:
digitalWrite(LEDARRAY_D,HIGH);digitalWrite(LEDARRAY_C, HIGH);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, LOW);
break;

            case 15:
digitalWrite(LEDARRAY_D,HIGH);digitalWrite(LEDARRAY_C,HIGH);digitalWrite(LEDARRAY_B, HIGH);digitalWrite(LEDARRAY_A, HIGH);
break;

            default : break;
      }
}

//*************************************************
//*************************************************



void Send( unsigned char dat)

{
      unsigned char i;
      digitalWrite(LEDARRAY_CLK, LOW);
      delayMicroseconds(1);;
      digitalWrite(LEDARRAY_LAT, LOW);
      delayMicroseconds(1);;

      for( i = 0 ; i < 8 ; i++ )
      {
            if( dat&0x01 )
            {
                  digitalWrite(LEDARRAY_DI, HIGH);
            }
            else
            {
                  digitalWrite(LEDARRAY_DI, LOW);
            }
```

```
digitalWrite(LEDARRAY_CLK, HIGH);
delayMicroseconds(1);;
digitalWrite(LEDARRAY_CLK, LOW);
delayMicroseconds(1);;
dat >>= 1;
```

AG Electrónica S.A.P.I DE C.V
República del Salvador N° 20 Segundo Piso
Teléfono: 5130 - 7210

ACOTACIÓN: N/A	http://www.agelectronica.com/	ESCALA: N/A	REALIZO: DGG REV: VJSR
TOLERANCIA: N/A	Controlador Para Matriz LED 16x16		
TOLERANCIA: N/A	Fecha: 20/05/2019	OKY3525-1	

AG Electrónica ¿Qué vamos a innovar hoy?