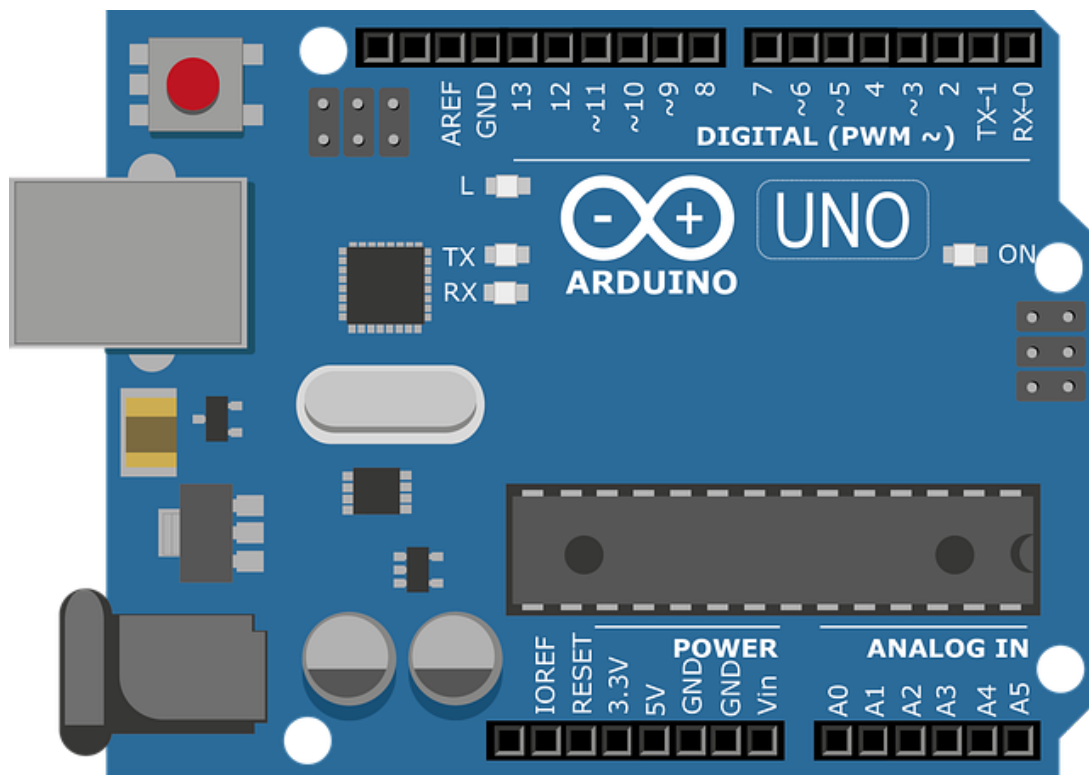


PRÁCTICAS CON ARDUINO



Práctica 13: MATRIZ DE LEDS

MAX7219

Grupo de Trabajo

Mónica Cañón Pascual

IES Pino Rueda

Departamento de Tecnología

Abril 2018



Finalidad de la práctica

1. En esta práctica vamos a programar una matriz de leds 8x8. La matriz que vamos a usar es del tipo FC-16 DIY Kit Module, cuya orientación es:

Module Orientation

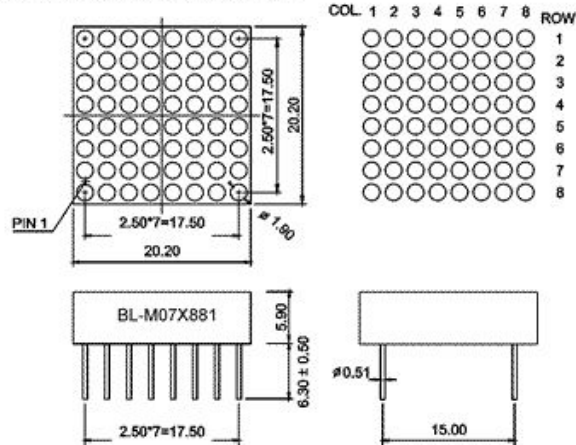
| | DP | A | B | C | D | E | F | G | |
|-----------|----|-------|---|---|---|---|---|---|-------------|
| | + | ----- | + | | | | | | |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | D0 |
| CLK <--- | | | | | | | | 1 | D1 <--- CLK |
| CS <--- | | | | | | | | 2 | D2 <--- CS |
| DOUT <--- | | | | | | | | 3 | D3 <--- DIN |
| GND ---- | 0 | | | | | | | 4 | D4 ---- GND |
| VCC ---- | 0 | 0 | | | | | | 5 | D5 ---- VCC |
| | 0 | 0 | 0 | | | | | 6 | D6 |
| | 0 | 0 | 0 | 0 | | | | 7 | D7 |
| | + | ----- | + | | | | | | |

2. Vamos a aprender a mandar texto o imágenes .

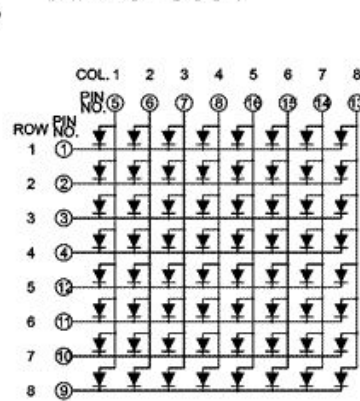
Información

1. Una matriz de leds de 8 x 8 es un cuadrado donde hay 64 leds, colocado en 8 filas y 8 columnas. En el siguiente ejercicio la matriz es monocolor, pero hay matrices cuyos leds son del tipo RGB, con lo que se aumentan las posibilidades de diseño.
2. La matriz de los 64 leds, los hay comercialmente de cátodo común o de ánodo común.

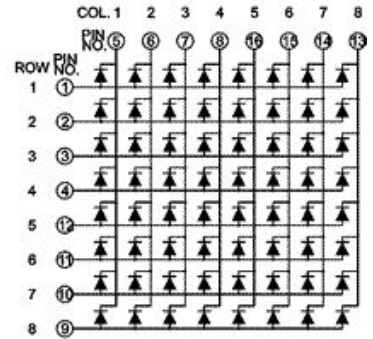
BL-M07X881 Series



BL-M07C881



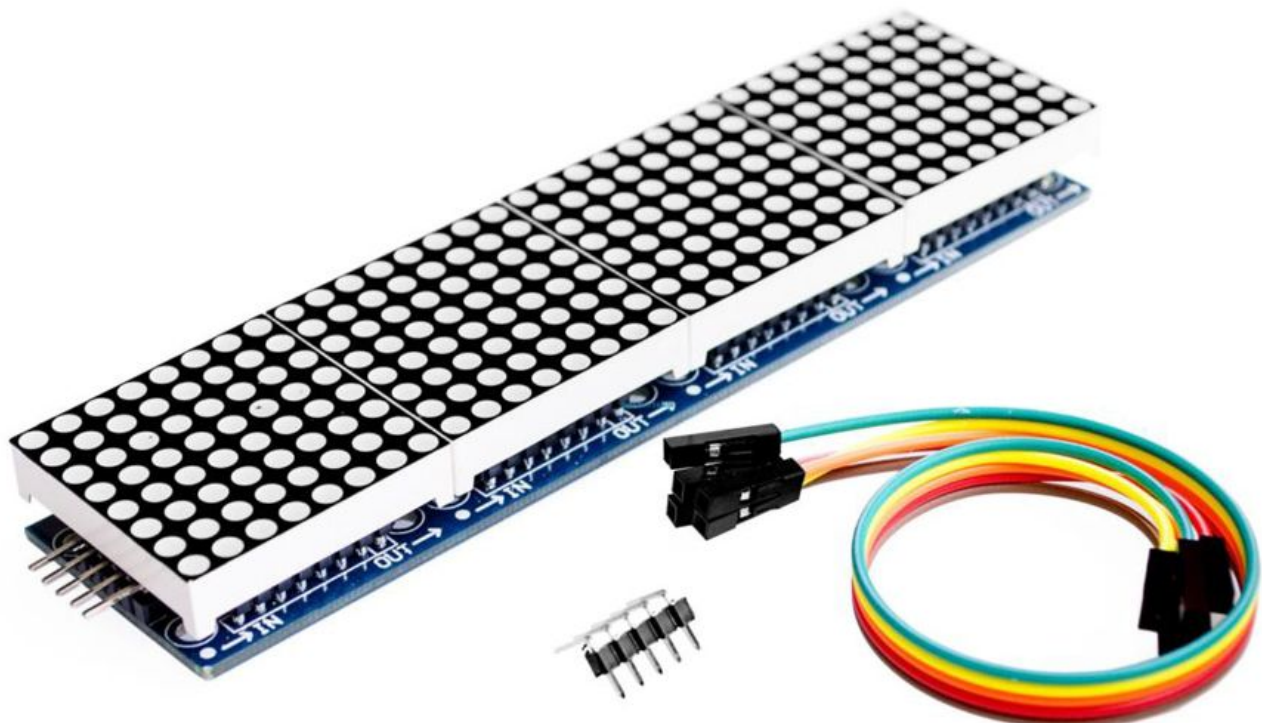
BL-M07D881



Para facilitar la conexiones de la matriz se usará la matriz con el chip de registro de desplazamiento integrado.

Hardware necesario

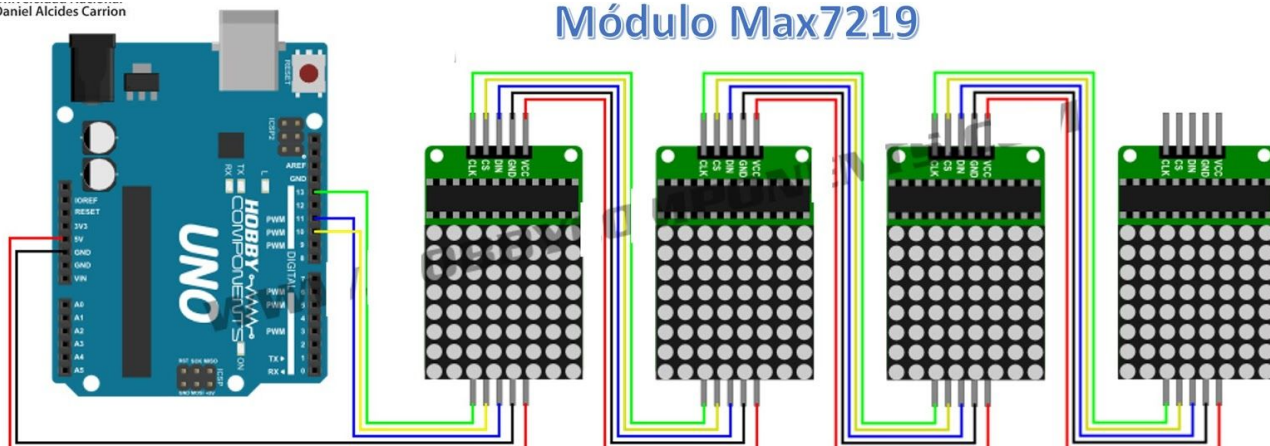
I. Esquema de conexiones





Daniel Alcides Carrion

Módulo Max7219



(Precio por Aliexpress: 4,01€ comprado el 1 noviembre de 2017)

- Nombre de la marca: LANDZO
- Número y nombre de modelo: **MAX7219 Dot Matrix Module** (Cátodo común)

- Operating Voltage: 5V
- Dimensions: 12.8 x 12.8 x 1.3 cm
- 5 PINES de conexiones:
 - VCC: 5V
 - GND: GND
 - DIN: P2.2
 - CS: P2.1
 - CLK: P2.0

Programación

IDE arduino

```
/*Copyright (C) 2017 Iván Rodríguez Méndez*/

// Tutorial 15 - Matriz LED 8x8
#include "LedControlMS.h"

LedControl lc=LedControl(12,11,10,1); // Los numeros se refieren
a que pin de ARDUINO tienes en cada uno de los terminales
/* 12 para el DIN, 11 para el CLK, 10 para el CS y el 1 se
refiere a la asignacion de la matriz*/
void setup()
{
    // El numero que colocamos como argumento de la funcion se
refiere a la direccion del decodificador
    lc.shutdown(0,false);
    lc.setIntensity(0,8); // La valores estan entre 1 y 15
    lc.clearDisplay(0);
}
void loop()
{
    for (int row=0; row<8; row++)
    {
        for (int col=0; col<8; col++)
        {
            lc.setLed(0,col,row,true); //
            delay(25);
        }
    }
}
```

```

}
for (int row=0; row<8; row++)
{
    for (int col=0; col<8; col++)
    {
        lc.setLed(0,col,row,false); //
        delay(25);
    }
}
delay(500);
lc.writeString(0,"Piensa 3D");
delay(1000);
}

```

```

1  /*Copyright (C) 2017 Iván Rodríguez Méndez*/
2  //Tutorial 16: Simbolos en matriz de LEDS
3  #include "LedControlMS.h"
4
5  LedControl lc=LedControl(12,11,10,1); // Los numeros se refieren
6  a que pin de ARDUINO tienes en cada uno de los terminales
7  /* 12 para el DIN, 11 para el CLK, 10 para el CS y el 1 se
8  refiere a la asignacion de la matriz*/
9
10 //Corazón pequeño
11 byte Corazon_datos[] = {
12 B00000000,
13 B01100110,
14 B11111111,
15 B11111111,
16 B01111110,
17 B00111100,
18 B00011000,
19 B00000000};
20
21 byte Cara_datos[] =
22 {B00111100,
23 B01000010,
24 B10100101,
25 B10000001,
26 B10100101,

```

```
27 B10011001,
28 B01000010,
29 B00111100};
30
31 byte Mensaje_datos[] =
32 {B00000000,
33 B00000100,
34 B00000000,
35 B01010100,
36 B01110100,
37 B01010100,
38 B00000000,
39 B00000000};
40
41 void setup()
42 {
43     // El numero que colocamos como argumento de la funcion se
44     refiere a la direccion del decodificador
45     lc.shutdown(0,false);
46     lc.setIntensity(0,8); // La valores estan entre 1 y 15
47     lc.clearDisplay(0);
48
49 }
50 void loop()
51 {
52     Representar(Mensaje_datos,5000);
53     trans();
54     delay(500);
55     Representar(Corazon_datos,5000);
56     Representar(Cara_datos,3000);
57     lc.clearDisplay(0);
58 }
59
60 // Funcion para representar una transicion animada
61
62 void trans(){ //Funcion de transicion para llenar y vaciar la
63 pantalla de puntos
64     for (int row=0; row<8; row++)
65     {
66         for (int col=0; col<8; col++)
67         {
68             lc.setLed(0,col,row,true); //
69             delay(25);
70         }
71     }
```



```
72   for (int row=0; row<8; row++)
73   {
74       for (int col=0; col<8; col++)
75       {
76           lc.setLed(0,col,row,false); //
77           delay(25);
78       }
79   }
80 }
81
82 // Definimos una funcion para representar las figuras
83
84 void Representar(byte *Datos,int retardo) //Funcion para la
85 representacion de bytes de datos para una matriz de 8x8
86 {
87     for (int i = 0; i < 8; i++)
88     {
89         lc.setColumn(0,i,Datos[7-i]);
90     }
91     delay(retardo);
92 }
```

Actividades y propuestas de mejora

Librerias :

https://github.com/tidus747/Tutoriales_programacion_basica_ARDUINO