Animación en Matriz LED 8x8

Universidad Autónoma De Aguascalientes

Ingeniería Electrónica

Tercer Semestre

Orlando Contreras[1], Emilio Peregrina[2], Ángel Granados[3]
348390[1], 230220[2], 263610[3]

***Abstract*—El presente documento tiene como objetivo explicar de manera clara y concisa el funcionamiento, la forma de mostrar imágenes en una matriz LED 8x8 así como, la manera de secuenciarlas entre éstas, centrándose exclusivamente en los aspectos técnicos y de funcionamiento. Enfocándose en cómo estos dispositivos forman parte de una maquina de estados, son secuenciales y controlados mediante hardware y software especializado como VHDL.**

***Index Terms*—FPGA, VHDL, Matriz de LED´s 8x8, Secuencial, Máquina de Estados.**

I. PLANEACIÓN

1. Delimitación y planteamiento

Al principio se pensó en ser un equipo con el doble de integrantes, pero eso llevaba de la mano un proyecto muchísimo más complejo, y debido a los tiempos, situaciones personales de los integrantes de ese primer equipo y el conflicto de intereses, se decidió dividir el equipo en dos, cada uno con su proyecto propio.

Cuando se llegó a un acuerdo acerca de los integrantes y objetivos, se organizó una junta con los integrantes del equipo después de clases y se llevó a cabo una lluvia de ideas. Aunque dentro de ésta hubo muchas ideas interesantes, cómo desafiantes, se consideró una animación en la matriz de LED´s como la más conveniente contemplando los tiempos, así como los conocimientos que se poseían al momento.

También cabe destacar que, resultó mucho más atractiva esta idea de la animación gracias a que ya se había elaborado y razonado el cómo ilustrar una imagen en la matriz de LED´s con hexadecimal y binario.

II. MARCO TEÓRICO

1.1 Propósito e importancia

Frecuentemente en los proyectos de electrónica suele surgir la necesidad de comunicar información básica al usuario y es aquí donde entra en escena la matriz de LED´s. Este sencillo componente de 8x8 resulta muy útil a la hora de querer mostrar alguna indicación simple, facilitando las interacciones entre el usuario y el circuito.

Ahora bien, cabe destacar que, aunque cada uno de los 64 LED´s de la matriz pueda ser controlado a gusto del usuario, la matriz se encuentra bastante limitada por su tamaño en cuanto a lo que puede mostrar en ella.

1.2 Funcionamiento y especificaciones

1.2.1 MAX7219

La matriz está conectada a una placa con un chip MAX7219, lo que permite trabajar rápidamente con matrices led y utilizando solo 4 cables para la comunicación SPI. Estos módulos se pueden conectar en cascada utilizando un solo bus SPI y mostrar textos o gráficos más grandes.

Este chip es lo que permite controlar cada uno de los LED´s independientemente uno del otro, también incluye un decoder BCD y posee una memoria RAM estática interna de 8x8.

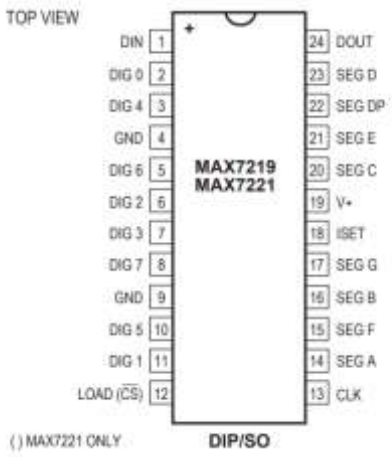


Figura 1. Vista superior del chip MAX7219

1.2.2 Diagrama de tiempos

El diagrama de tiempos ilustra cómo deben sincronizarse las señales para garantizar una comunicación adecuada entre el microprocesador y el controlador. Del diagrama se pueden concluir tres señales involucradas:

CLK: Señal de reloj que sincroniza la transferencia de datos.

DIN: Entrada de datos seriales.

LOAD/CS: Indica el final de un paquete de datos y los almacena en registros internos.

Así mismo, se puede observar en el diagrama que se destacan ciertos tiempos que resultan ser de suma relevancia para la matriz:

tCP (Clock Period): Período de la señal de reloj, mínimo de 100 ns.

tCH y tCL: Duración de los niveles alto y bajo del reloj (50 ns cada uno).

tDS y tDH: Tiempo de configuración y retención de datos en DIN (25 ns y 0 ns, respectivamente).

tCSS y tCSH: Tiempo mínimo de activación y desactivación de LOAD/CS respecto a CLK.

El diagrama asegura que los datos enviados en paquetes de 16 bits se procesen correctamente. El bit más significativo (D15) se envía primero, y LOAD/CS debe

activarse al final del paquete para que los datos se transfieran al registro correspondiente.

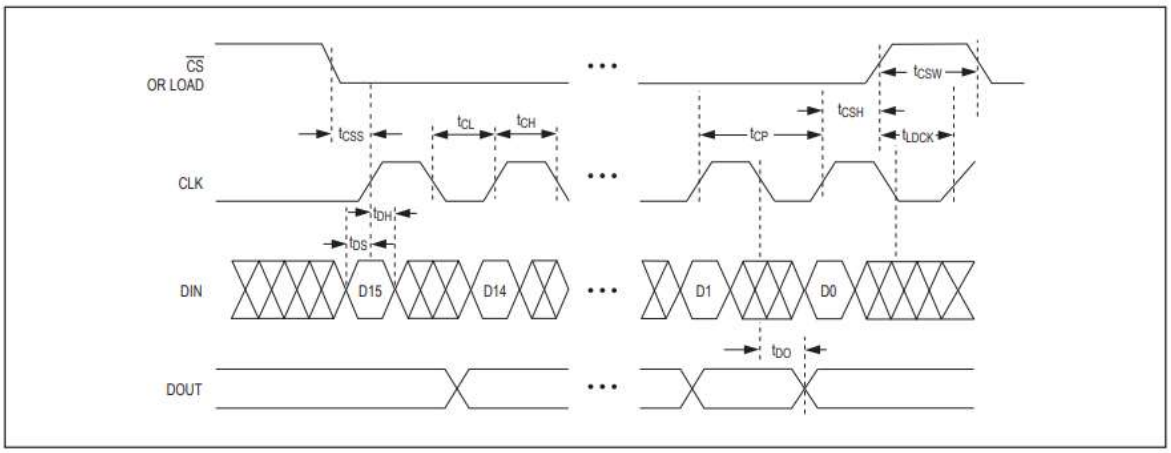


Figura 2. Diagrama de tiempos

1.2.3 Tabla 1: Formato de Datos Seriales

Esta tabla aclara cómo es que deben de estructurarse los 16 bits que se envían al dispositivo para configurar o actualizar datos.

D15-D12: Bits no utilizados, pueden ser cualquier valor.

D11-D8: Dirección del registro al que se quiere acceder.

D7-D0: Datos que se escriben en el registro seleccionado.

Table 1. Serial-Data Format (16 Bits)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB			DATA				LSB

Figura 3. Tabla de datos seriales

1.2.4 Tabla 2: Mapa de direcciones de registros

Se enlistan las direcciones hexadecimales de los registros disponibles en el controlador como:

0xX0 (No-Op): Se usa para mantener los datos en cascadas sin cambiar estados.

0xX1-0xX8 (Digit 0-7): Controlan los estados de cada dígito (qué segmentos están encendidos).

0xX9 (Decode Mode): Configura si se usa o no decodificación para los dígitos.

0xXA (Intensity): Ajusta el brillo del display.

0xXB (Scan Limit): Determina cuántos dígitos se escanean.

0xXC (Shutdown): Activa o desactiva el display.

0xXF (Display Test): Enciende todos los segmentos para pruebas.

Table 2. Register Address Map

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xXF

Figura 4. Tabla de registros

1.2.5 Tabla 3: Registro de apagado

Define si el dispositivo está encendido o apagado.

D7-D0: Un valor de 0x0 pone el controlador en modo apagado, y 0x1 lo pone en operación normal.

Table 3. Shutdown Register Format (Address (Hex) = 0xXC)

MODE	ADDRESS CODE (HEX)	REGISTER DATA							
		D7	D6	D5	D4	D3	D2	D1	D0
Shutdown Mode	0xXC	X	X	X	X	X	X	X	X
Normal Operation	0xXC	X	X	X	X	X	X	X	1

Figura 5. Tabla de ON/OFF

1.2.6 Tabla 4: Registro de modo de decodificación

Configura si se usa decodificación BCD (Code-B) o control directo de segmentos.

D7-D0: Cada bit corresponde a un dígito. Un 1 activa la decodificación para el dígito, y un 0 permite control directo. Ejemplo:

- 0x00: Sin decodificación en ningún dígito.
- 0xFF: Decodificación en todos los dígitos.

Table 4. Decode-Mode Register Examples (Address (Hex) = 0xX9)

DECODE MODE	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
No decode for digits 7–0	0	0	0	0	0	0	0	0	0x00
Code B decode for digit 0 No decode for digits 7–1	0	0	0	0	0	0	0	1	0x01
Code B decode for digits 3–0 No decode for digits 7–4	0	0	0	0	1	1	1	1	0x0F
Code B decode for digits 7–0	1	1	1	1	1	1	1	1	0xFF

Figura 6. Tabla de modo de decodificación

1.2.7 Tabla 7: Registro de intensidad

Configura el brillo del display mediante un modulador PWM.

D7-D4: No se usan.

D3-D0: Controlan el ciclo de trabajo del PWM, ajustando el brillo en 16 niveles desde 0x0 (mínimo) hasta 0xF (máximo).

Table 7. Intensity Register Format (Address (Hex) = 0xXA)

DUTY CYCLE		D7	D6	D5	D4	D3	D2	D1	D0	HEX CODE
MAX7219	MAX7221									
1/32 (min on)	1/16 (min on)	X	X	X	X	0	0	0	0	0xX0
3/32	2/16	X	X	X	X	0	0	0	1	0xX1
5/32	3/16	X	X	X	X	0	0	1	0	0xX2
7/32	4/16	X	X	X	X	0	0	1	1	0xX3
9/32	5/16	X	X	X	X	0	1	0	0	0xX4
11/32	6/16	X	X	X	X	0	1	0	1	0xX5
13/32	7/16	X	X	X	X	0	1	1	0	0xX6
15/32	8/16	X	X	X	X	0	1	1	1	0xX7
17/32	9/16	X	X	X	X	1	0	0	0	0xX8
19/32	10/16	X	X	X	X	1	0	0	1	0xX9
21/32	11/16	X	X	X	X	1	0	1	0	0xXA
23/32	12/16	X	X	X	X	1	0	1	1	0xXB
25/32	13/16	X	X	X	X	1	1	0	0	0xXC
27/32	14/16	X	X	X	X	1	1	0	1	0xXD
29/32	15/16	X	X	X	X	1	1	1	0	0xXE
31/32	15/16 (max on)	X	X	X	X	1	1	1	1	0xXF

Figura 7. Tabla de registro de intensidad

1.2.8 Tabla 8: Registro de límite de escaneo

Determina cuántos dígitos (de 1 a 8) están activos en el display.

D7-D3: No se usan.

D2-D0: Número de dígitos activos. Ejemplo:

- 0x0: Muestra solo el dígito 0.
- 0x7: Muestra todos los dígitos (0-7).

Table 8. Scan-Limit Register Format (Address (Hex) = 0xXB)

SCAN LIMIT	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
Display digit 0 only*	X	X	X	X	X	0	0	0	0xX0
Display digits 0 & 1*	X	X	X	X	X	0	0	1	0xX1
Display digits 0 1 2*	X	X	X	X	X	0	1	0	0xX2
Display digits 0 1 2 3	X	X	X	X	X	0	1	1	0xX3
Display digits 0 1 2 3 4	X	X	X	X	X	1	0	0	0xX4
Display digits 0 1 2 3 4 5	X	X	X	X	X	1	0	1	0xX5
Display digits 0 1 2 3 4 5 6	X	X	X	X	X	1	1	0	0xX6
Display digits 0 1 2 3 4 5 6 7	X	X	X	X	X	1	1	1	0xX7

Figura 8. Tabla de limite de escaneo

1.2.9 Tabla 9: Corriente máxima por segmento para 1, 2 o 3 dígitos

Establece la corriente máxima permitida por segmento dependiendo del número de dígitos activos.

Número de dígitos activos: 1, 2 o 3.

Corriente máxima recomendada: 10mA, 20mA o 30mA, respectivamente.

Table 9. Maximum Segment Current for 1-, 2-, or 3-Digit Displays

NUMBER OF DIGITS DISPLAYED	MAXIMUM SEGMENT CURRENT (mA)
1	10
2	20
3	30

Figura 8. Tabla de limite de corriente

III. IMPLEMENTACIÓN Y DISEÑO

Ahora, se mostrarán los resultados que se obtuvieron en el desarrollo de este proyecto, implementando los conocimientos obtenidos en el transcurso de la materia y al mismo tiempo contemplando las especificaciones e instrucciones de la datasheet de la matriz 8x8.

3.1 Código VHDL

```
---------- Code ----------
------ Orlando Reyes ------
--------- Auf Das ---------
-------- LedMatrix --------
-------- 05/12/2024 --------
------- Main Library -------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

--------- Pin/out ---------
entity LedMatrix is
   port
      (
      CLK,SW  : in std_logic;
      CS, CLK2, DIN   : out std_logic
      );
end LedMatrix;

architecture juve3dstudio of LedMatrix is
   -- Main Machine --
   signal actual, siguiente : std_logic_vector (6 downto 0) :=  "0000000" ;
   -- Program Counter --
   signal PC,PC_H : std_logic_vector (4 downto 0) := "00000";
   signal EN_PC : std_logic ;
   -- 1 Second Counter --
   signal c, cplus : std_logic_vector(24 downto 0) := (others => '0'); -- 25 Bits
   signal d : std_logic_vector (1 downto 0) := "00";

   -- DIN --
   signal Qsig, Qs, BUSS: std_logic_vector (15 downto 0):=
"0000000000000000";

   -- Data --
   signal ADDRESS : std_logic_vector (3 downto 0):= "0000";
```

```
    signal DATA : std_logic_vector (7 downto 0) := "00000000";
    signal TRANS : STD_LOGIC_VECTOR (2 downto 0) := "000"; -- 5
    signal DATA1, DATA2 ,DATA3, DATA4, DATA5, DATA6, DATA7, DATA8 :
std_logic_vector (7 downto 0) := "00000000";
    constant m : integer := 1;
    constant tim : integer := 33554431;

begin
    ----- 1 Second Counter -----
    -- Memoria --
    c <= cplus when clk'event and clk='1';
    -- Logica de estado Siguiente --
    cplus <= c+'1' when (c <= tim) else (others => '0');
    -- Logica de Salida --
    TRANS <= TRANS + '1' when c = tim and clk'event and clk = '1';
    --TRANS <= TRANS + '1' when SW'event and SW = '0';

    ----- Main Machine -----
    -- Memoria --
    actual <= siguiente when CLK'event and CLK = '1';
    -- Logica de estado Siguiente --
    siguiente <= actual + '1' when actual < 67  else (others => '0') ;
    -- Logica de Salida --
    CS<= '0' when actual>=0 and actual< 66 else '1' ;

    CLK2 <=  actual(1);


    --------- Data ---------
    -- 0x00,0x24,0x00,0x42,0x42,0x3C,0x00,0x00,
    -- 0x00,0x24,0x00,0x00,0x7E,0x00,0x00,0x00,
    -- 0x07,0x27,0x01,0x00,0x3C,0x42,0x42,0x00,
    -- 0x3F,0x3F,0x0F,0x07,0x3F,0x41,0x41,0x00,

    -- 0xFF,0xFF,0xFF,0x7F,0x3F,0x5F,0x4F,0x07,
    -- 0x90,0x12,0xBC,0x7C,0x7D,0x38,0x31,0x84,
    -- 0x00,0xA5,0x42,0xA5,0x00,0x3C,0x0C,0x00,

    DATA1 <=
       x"00" when TRANS < 3 and TRANS >0 else
       x"07" when TRANS = 3 else
       x"3F" when TRANS = 4 else
       x"FF" when TRANS = 5 else
       x"90" when TRANS = 6 else
       x"00" when TRANS = 7 else
```

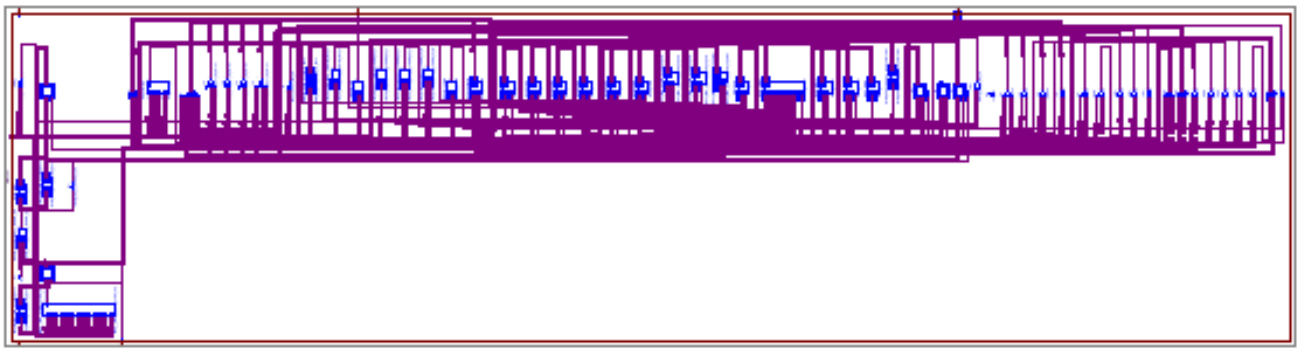
```
    x"00";

  DATA2 <=
    x"24" when TRANS < 3 and TRANS >0 else
    x"27" when TRANS = 3 else
    x"3F" when TRANS = 4 else
    x"FF" when TRANS = 5 else
    x"12" when TRANS = 6 else
    x"A5" when TRANS = 7 else
    x"00";

  DATA3 <=
    x"00" when TRANS < 3 and TRANS >0 else
    x"01" when TRANS = 3 else
    x"0F" when TRANS = 4 else
    x"FF" when TRANS = 5 else
    x"BC" when TRANS = 6 else
    x"42" when TRANS = 7 else
    x"00";

  DATA4 <=
    x"42" when TRANS = 1 else
    x"00" when TRANS = 2 or TRANS = 3  else
    x"07" when TRANS = 4 else
    x"7F" when TRANS = 5 else
    x"7C" when TRANS = 6 else
    x"A5" when TRANS = 7 else
    x"00";

  DATA5 <=
    x"42" when TRANS = 1 else
    x"7E" when TRANS = 2 else
    x"3C" when TRANS = 3 else
    x"3F" when TRANS = 4  else
    x"7D" when TRANS = 6 else
    x"00" when TRANS = 7 else
    x"00";

  DATA6 <=
    x"3C" when TRANS = 1 else
    x"00" when TRANS = 2 else
    x"42" when TRANS = 3 else
    x"41" when TRANS = 4 else
    x"5F" when TRANS = 5 else
    x"38" when TRANS = 6 else
```

```
    x"3C" when TRANS = 7 else
    x"00";

DATA7 <=
    x"00" when TRANS < 3 and TRANS >0 else
    x"42" when TRANS = 3 else
    x"41" when TRANS = 4 else
    x"4F" when TRANS = 5 else
    x"31" when TRANS = 6 else
    x"0C" when TRANS = 7 else
    x"00";

DATA8 <=
    x"00" when TRANS < 5 else
    x"07" when TRANS = 5 else
    x"84" when TRANS = 6 else
    x"00" when TRANS = 7 else
    x"00";


------ Program Counter ------
ADDRESS <=
    x"C" when PC = 0 else
    x"A" when PC = 1 else
    x"B" when PC = 2 else
    unsigned(PC(3 downto 0)) - 2;

PC <= PC_H  when EN_PC = '1' and clk'event and clk = '1';
PC_H <= PC + '1' when  PC < 10 else "00000";

EN_PC <= '1' when (actual = 65) else '0'; -- numero maximo de estados

DATA <=
    x"01" when PC = 0 else
    x"00" when PC = 1 else
    x"07" when PC = 2 else
    DATA1 when PC = 3 else
    DATA2 when PC = 4 else
    DATA3 when PC = 5 else
    DATA4 when PC = 6 else
    DATA5 when PC = 7 else
    DATA6 when PC = 8 else
    DATA7 when PC = 9 else
    DATA8 when PC = 10 else
    x"00";
```

```
   BUSS <= "0000" & ADDRESS (3 downto 0) & DATA (7 downto 0) ;


   -------- Shift Register --------

   -- Estado siguiente
   Qsig <=
      BUSS when actual <  4 else -- Load Mode
      Qs(14 downto 0) & '0';
   -- Registro
   Qs <= Qsig when Clk'event and CLK = '1' and actual(1 downto 0) = "01";
   -- Salida
   DIN <= Qs(15);

end juve3dstudio;

-------- Testbench --------
------ Orlando Reyes ------
--------- Auf Das ---------
--------- LedMatrix ---------
-------- 5/12/2024 --------
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity tb is
end tb;

architecture sim of tb is

component LedMatrix
   port
   (
      CLK : in std_logic;
      CS, CLK2, DIN   : out std_logic
   );
end component;

signal S_Clk : std_logic := '0';

signal S_CS : std_logic := '0';
signal S_CLK2 : std_logic := '0';
signal S_DIN : std_logic := '0';
```

3.2 Simulación

3.3 Esquemático del circuito



IV. CONCLUSIONES PERSONALES

- Orlando Contreras:

El pensar en la electrónica y los circuitos de esta manera te da otra perspectiva completamente diferente cuando te estás enfrentando a un problema, ya que, si bien tenemos la lógica de programación, la lógica de circuitos se debe de abarcar de manera distinta ya que no puedes resolver estos problemas solamente con un tipo de lógica.

- Ángel Granados

El desarrollo de este proyecto es el culmen de lo aprendido en dos semestres dedicados a VHDL. Sin duda el lenguaje de descripción de hardware es importante, pero lo es más la lógica y el pensamiento detrás del diseño de los circuitos que armamos, aunque es una animación medianamente simple, detrás de esos pocos frames de animación se encuentran horas de trabajo, conocimientos reforzados e incluso algunos nuevos adquiridos. Considero que ese es el verdadero logro que nos llevamos.

-Emilio Peregrina

La forma en la que VHDL y la descripción de circuitos te obliga a razonar y pensar es algo que muchas veces resulta incómodo, pues ya teniendo una forma de pensar en la resolución de problemas con la programación, sin duda salir de la zona de comfort resulta algo desafiante pero no por eso menos interesante.

Creo que este proyecto me ha ayudado a entender conceptos que no me atrevía a abarcar por que resultaban intimidantes o incluso abrumadores, así mismo a no tener miedo a abrir nuevas puertas y oportunidades para aprender algo nuevo.

V. REFERENCIAS

[1] Kit Matriz LED 8x8 MAX7219. (s. f.). Naylamp Mechatronics - Perú. https://naylampmechatronics.com/led/265-kit-matriz-led-8x8-max7219.html

[2] General Description. (s/f). *MAX7219/MAX7221 serially interfaced, 8-digit LED display drivers*. Analog.com. Recuperado el 16 de diciembre de 2024, de https://www.analog.com/media/en/technical-documentation/data-sheets/max7219-max7221.pdf

[3] *Matriz 8x8 Max7219 Led*. (s/f). UNIT Electronics. Recuperado el 16 de diciembre de 2024, de https://uelectronics.com/producto/matriz-8x8-max7219-led/?srsltid=AfmBOopOCPo0fBbjLeEzAz4wZM6-Gom1xHsARDHYDSt7mc9FP-afncst

[4] General Description. (s/f-a). *8 by 8 dot matrix LED displays with Cascadable Serial driver B32CDM8 B48CDM8 B64CDM8*. Farnell.com. Recuperado el 16 de diciembre de 2024, de https://www.farnell.com/datasheets/29075.pdf