

Matriz de 8x32 con Arduino.

Las matrices compuestas por leds son empleadas comúnmente para iluminaciones de letreros grandes o de espectaculares, esto permite una mejor visualización. La matriz está constituida por leds que se conectan a una fila y a una columna, el ánodo va conectado a una fila y el cátodo va conectado a una salida. La matriz de 8x 32 es un sistema integrado de serie de Entrada/Salida, es de cátodo común y compatible con la mayoría de los sistemas de microcontroladores, incluyendo Arduino, Raspberry Pi, Pic, Arm, AVR. Entre otros.

En el mercado existen diferentes tipos de dispositivos que son capaces de multiplexar, desfasar y variar el encendido y apagado de leds. En la figura n° 1 se puede observar un ejemplo de cómo se lleva a cabo una de estas funciones.



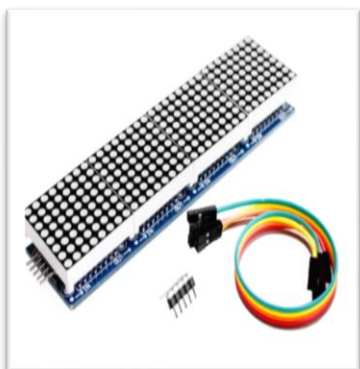
Figura n° 1: Matriz de leds mostrando un reloj digital.

Como se mencionó anteriormente existen circuitos que se encargan de hacer diferentes procesos para el encendido de la matriz led, su implementación en distintos equipos electrónicos varía dependiendo de las necesidades del proyecto; uno de tales circuitos es el max7219, el cual, multiplexa y envía las señales de forma que se puedan programar las entradas para una salida correcta, por lo que se puede implementar con la tarjeta de desarrollo Arduino. Más adelante se podrá observar un ejemplo de cómo hacer el funcionamiento una matriz de 8x32 con Max7219.

Material Utilizado.

Electrónico

Matriz 8x8 con Max7219.



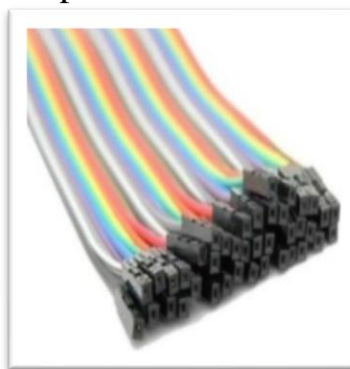
SKU: MO0073

Arduino uno RV3.



SKU: AR0016

Dupont M/H de 20 cm.



SKU: CS0006

Diagrama de Conexión.

El integrado MAX7219 permite controlar una matriz LED de 8x8, matrices más pequeñas, visualizador de 7 segmentos y 8 dígitos o bien una conexión de matrices como la de 8x32; la cual es la conexión en cascada de matrices de led de 8x8.

Posee los pines o conectores VCC, GND, DIN (datos de entrada), CS (Chip select) que nos sirve para seleccionar el numero de matriz a usar y CLK (Clock - reloj). En la figura n° 2 se puede observar las conexiones la matriz de 8x32.

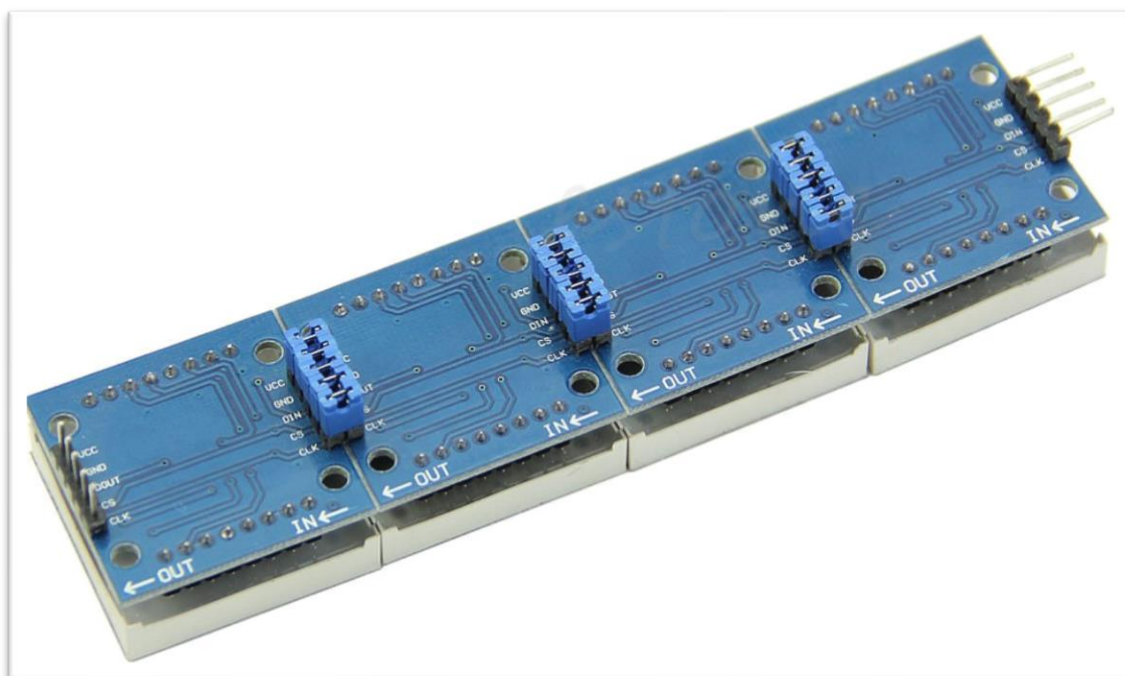


Figura n° 2: Matriz 8x32 con Max7219.

Para probar la matriz, se conecta al Arduino uno como se muestra en la figura n° 3 y se describe a continuación:

Alimentación del Circuito.

- Conecte el pin 5V del Arduino UNO al pin Vcc de la matriz.
- Conecte el pin GND del Arduino UNO al pin GND de la matriz.

Conexión de la Matriz de 8x32.

- Conecte el pin digital 12 del Arduino UNO al pin DIN de la matriz.
- Conecte el pin digital 11 del Arduino UNO al pin CLK de la matriz.
- Conecte el pin digital 10 del Arduino UNO al pin CS de la matriz.

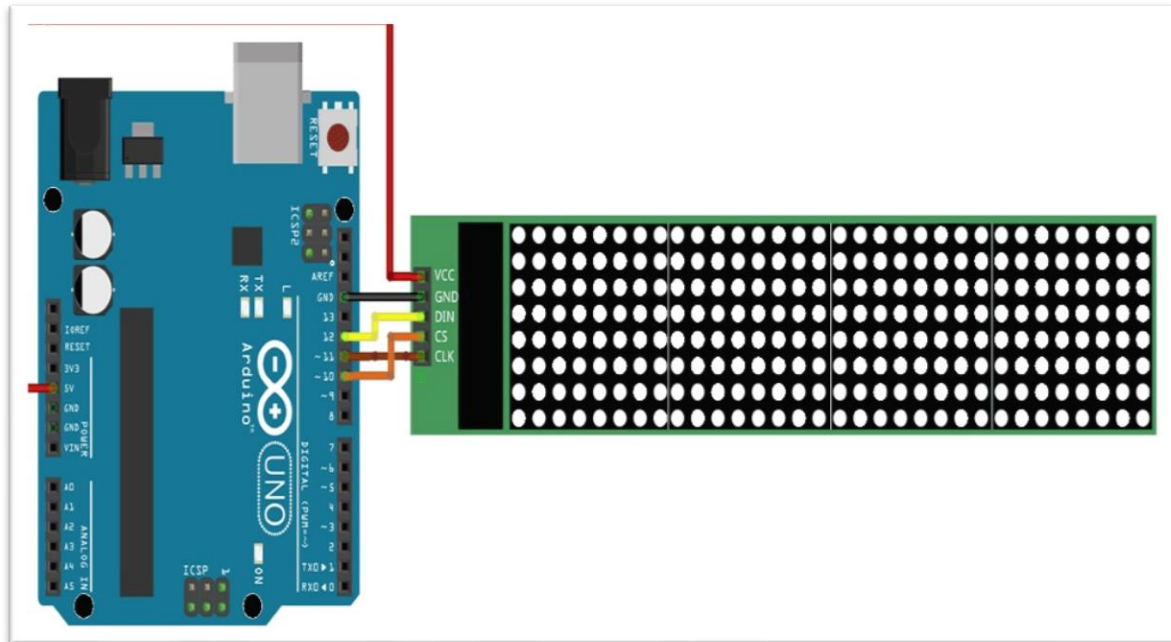


Figura nº 3: Diagrama de Conexión la matriz de 8x32.

Código Usado.

Para la programación de la matriz hay dos formas sencillas de hacerlo. La primera consta de declarar todos los caracteres posibles (A-Z y 0-9) como valores binarios, la segunda consta de dibujar los patrones o dibujos que se necesitan representar en la matriz o avisos de manera sencilla. Para ambos casos se necesitará una librería especial, la cual se puede descargar en uno de los links al final; además de contar con el IDE de arduino.

```
#include "LedControl.h"

//Declaramos los pines: DIN, CLK, CS y el
nº de displays conectados en serie.

LedControl lc=LedControl(12,11,10,4);
unsigned long delayTime=500;          //Pausa entre simbolos

byte Heart1[] = { // Letra H
    B00000000,
    B01000010,
    B01000010,
    B01111110,
    B01111110,
    B01000010,
    B01000010,
    B00000000
};

byte Heart2[] = { // Simbolo -
    B00000000,
    B00000000,
    B00000000,
    B01111110,
    B01111110,
    B00000000,
    B00000000,
    B00000000
};

byte Heart3[] = { // Letra A
    B00000000,
    B01111110,
    B01000010,
    B01000010,
    B01111110,
    B01000010,
    B01000010,
    B00000000
};

byte Heart4[] = { // Letra V
    B00000000,
    B01000010,
    B01000010,
    B01000010,
    B01000010,
    B00100100,
    B00011000,
    B00000000
};

byte Heart5[] = { // Letra R
    B00000000,
    B01111110,
    B01000010,
    B01111110,
    B01001000,
    B01000100,
    B01000010,
    B00000000
};
```

```
byte Heart6[] = { // Espacio
    B00000000,
    B00000000,
    B00000000,
    B00000000,
    B00000000,
    B00000000,
    B00000000,
    B00000000
};

//Sub para transformar array #1 en un patron para la matriz.
void Heart1GO()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(3,i,Heart1[i]);
        lc.setRow(2,i,Heart2[i]);
        lc.setRow(1,i,Heart3[i]);
        lc.setRow(0,i,Heart4[i]);
        lc.setRow(4,i,Heart5[i]);
        lc.setRow(4,i,Heart6[i]);
    }
}

void Heart2GO()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(3,i,Heart2[i]);
        lc.setRow(2,i,Heart3[i]);
        lc.setRow(1,i,Heart4[i]);
        lc.setRow(0,i,Heart5[i]);
        lc.setRow(4,i,Heart6[i]);
        lc.setRow(4,i,Heart1[i]);
    }
}

void Heart3GO()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(3,i,Heart3[i]);
        lc.setRow(2,i,Heart4[i]);
        lc.setRow(1,i,Heart5[i]);
        lc.setRow(0,i,Heart6[i]);
        lc.setRow(4,i,Heart1[i]);
        lc.setRow(4,i,Heart2[i]);
    }
}

void Heart4GO()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(3,i,Heart4[i]);
        lc.setRow(2,i,Heart5[i]);
        lc.setRow(1,i,Heart6[i]);
        lc.setRow(0,i,Heart1[i]);
        lc.setRow(4,i,Heart2[i]);
        lc.setRow(4,i,Heart3[i]);
    }
}
```

```

void Heart5GO()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(3,i,Heart5[i]);
    lc.setRow(2,i,Heart6[i]);
    lc.setRow(1,i,Heart1[i]);
    lc.setRow(0,i,Heart2[i]);
    lc.setRow(4,i,Heart3[i]);
    lc.setRow(4,i,Heart4[i]);
  }
}

```

```

Heart5GO();    //Mostramos el patrón #5
delay(delayTime); //Pequeña pausa

```

```

Heart6GO();    //Mostramos el patrón #6
delay(delayTime); //Pequeña pausa
}

```

```

void Heart6GO()
{
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(3,i,Heart6[i]);
    lc.setRow(2,i,Heart1[i]);
    lc.setRow(1,i,Heart2[i]);
    lc.setRow(0,i,Heart3[i]);
    lc.setRow(4,i,Heart4[i]);
    lc.setRow(4,i,Heart5[i]);
  }
}

```

```

void setup()          //Esta sub se ejecutará 1 sola vez al arrancar el Arduino.

```

```

{
  lc.shutdown(3,false); //Iniciamos la matriz led #4
  lc.shutdown(2,false); //Iniciamos la matriz led #3
  lc.shutdown(1,false); //Iniciamos la matriz led #2
  lc.shutdown(0,false); //Iniciamos la matriz led #1

  lc.setIntensity(3,5); //Intensidad de los led en la matriz #4
  lc.setIntensity(2,5); //Intensidad de los led en la matriz #3
  lc.setIntensity(1,5); //Intensidad de los led en la matriz #2
  lc.setIntensity(0,5); //Intensidad de los led en la matriz #1

```

```

  lc.clearDisplay(3); //Apagamos todos los led de la matriz #4
  lc.clearDisplay(2); //Apagamos todos los led de la matriz #3
  lc.clearDisplay(1); //Apagamos todos los led de la matriz #2
  lc.clearDisplay(0); //Apagamos todos los led de la matriz #1
}

```

```

void loop()          //Esta sub se ejecutara en bucle una y otra vez mientras el Arduino este alimentado.

```

```

{
  Heart1GO();    //Mostramos el patrón #1
  delay(delayTime); //Pequeña pausa

```

```

  Heart2GO();    //Mostramos el patrón #2
  delay(delayTime); //Pequeña pausa

```

```

  Heart3GO();    //Mostramos el patrón #3
  delay(delayTime); //Pequeña pausa

```

```

  Heart4GO();    //Mostramos el patrón #4
  delay(delayTime); //Pequeña pausa

```


Imágenes de Funcionamiento.

En las figuras nº 4 y 5 se puede observar el funcionamiento la matriz de 8x32 conectado al Arduino, mostrando un letrero hecho por símbolos.

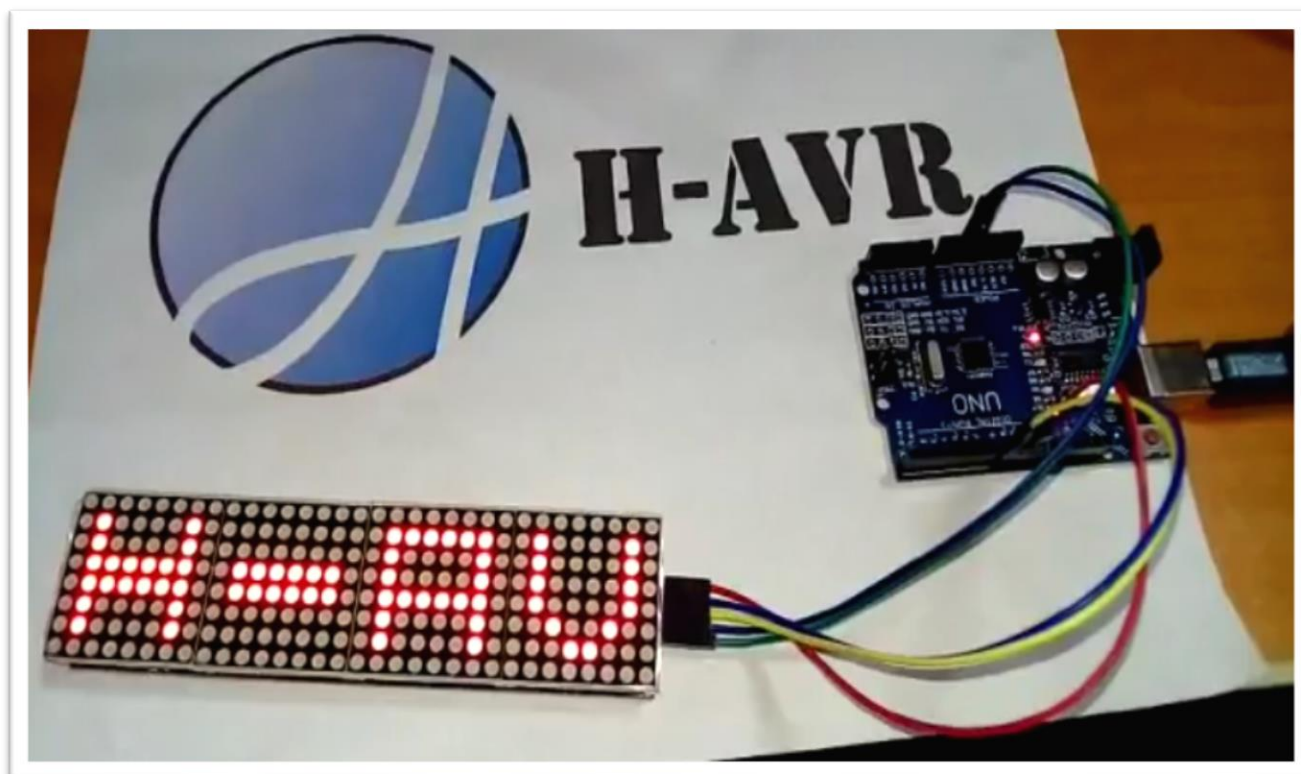


Figura nº 4: Funcionamiento la matriz 8x32.

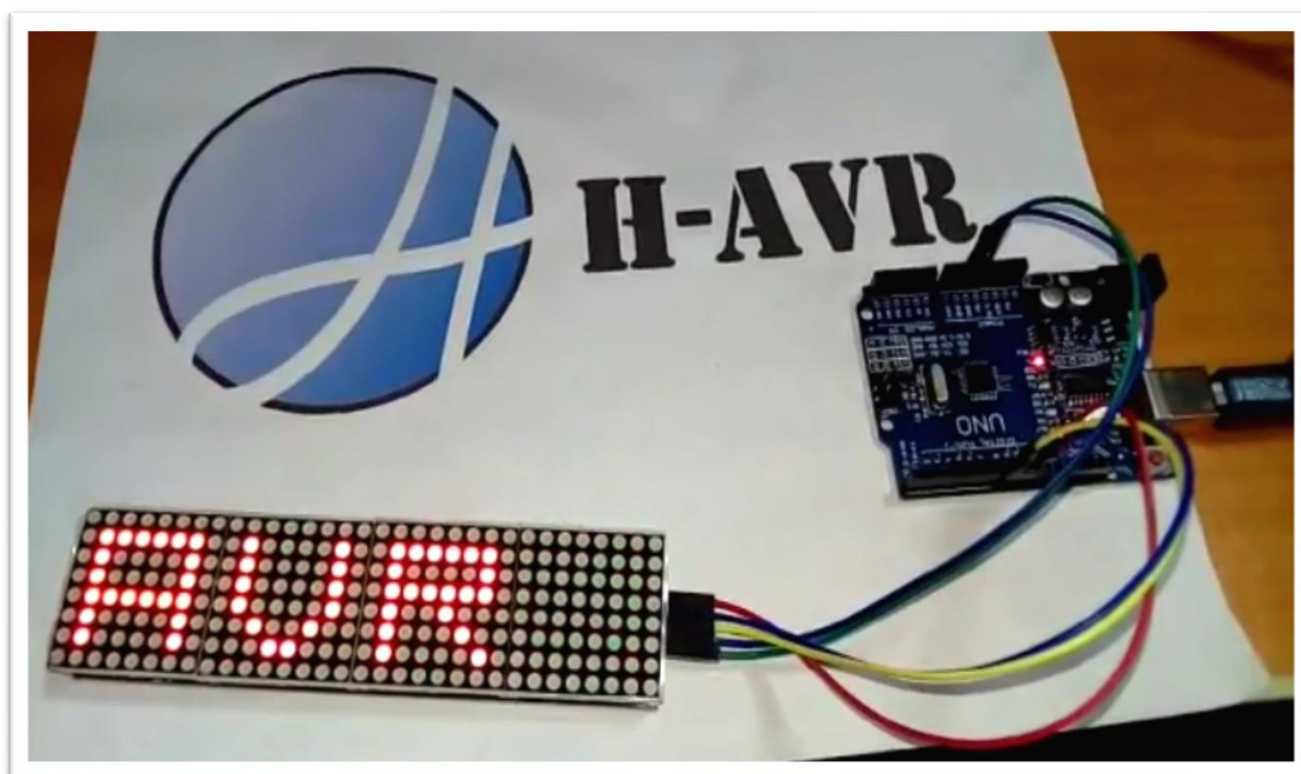


Figura nº 5: Funcionamiento la matriz 8x32 (2).

Conclusiones.

Al término de esta pequeña práctica demostrativa de como conectar una matriz de 8x32 a un Arduino uno, se puede aclarar algunas cosas que son importantes tomar en cuenta; una de ellas es el tiempo de visualización; es decir, si el tiempo es muy rápido no se alcanzara a leer bien el mensaje, por el contrario si es muy lento será tedioso.

Del mismo modo otro punto de consideración es el valor binario del símbolo o símbolos a utilizar para ello utilizamos una herramienta online hecha por minitronica.com; donde se dibuja el patrón o símbolo y nos da el valor binario y hexadecimal del dibujo. Sin dejar de mencionar que la librería sigue siendo parte importante de nuestro programa.

Al final, en comparación por ejemplo con una LCD, la visualización a grandes distancias es mucho mejor con la matriz de leds y más fácil en cuestión de manejo de símbolos y animación sencilla.

Contacto.

- <http://www.h-avr.mx/>

Video del Funcionamiento.

- <https://www.youtube.com/watch?v=B11G2SZk-q8>

Librería.

- <https://github.com/wayoda/LedControl>

Herramienta Online.

- <https://www.riyas.org/2013/12/online-led-matrix-font-generator-with.html>

Donde comprar:

