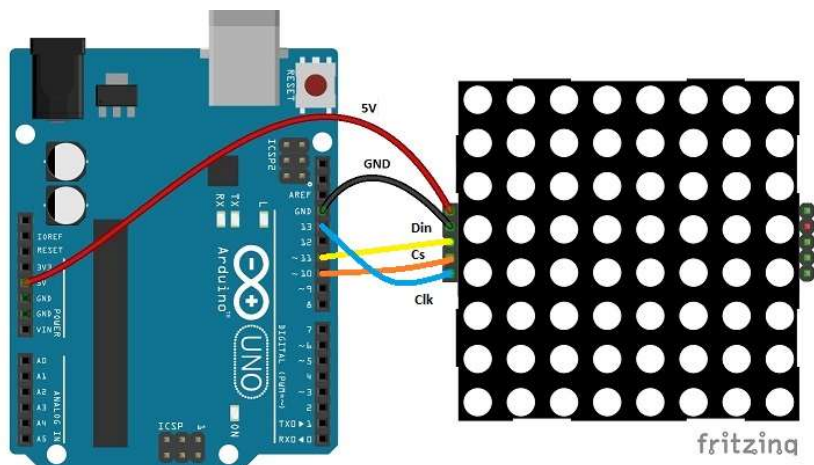
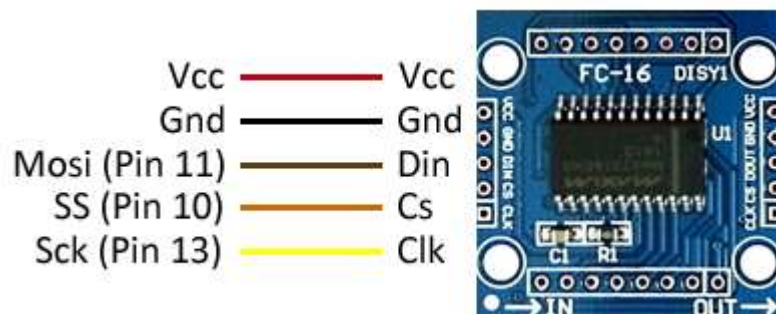


Matriz de led 8x8x4 (FC-16)



El pinout para una placa Arduino será:



Para utilizar la matriz, lo más práctico es a través de librerías existentes para MAX7219.

Una de las más completas es la MD_MAX72XX-master

Ver apéndice de "MD_MAX72XX-master Instalación".

Dicha librería se implementa a través de objetos y clases, con lo cual es necesario conocerla, y para ello pueden recurrir al apéndice "MD_MAX72XX-master Referencia de Clase".

Veamos parte de un ejemplo, para ver como se implementa dicha librería en Arduino.

```
#include <MD_MAX72xx.h>
#include <SPI.h>

// Turn on debug statements to the serial output
#define DEBUG 0

#if DEBUG
#define PRINT(s, x)    { Serial.print(F(s)); Serial.print(x); }
#define PRINTS(x)      Serial.print(F(x))
#define PRINTD(x)      Serial.println(x, DEC)
#else
#define PRINT(s, x)
#define PRINTS(x)
#define PRINTD(x)
#endif

// Define the number of devices we have in the chain and the hardware interface
// NOTE: These pin numbers will probably not work with your hardware and may
// need to be adapted
#define MAX_DEVICES    4

#define CLK_PIN         13  // or SCK
#define DATA_PIN       11  // or MOSI
#define CS_PIN          10  // or SS

// SPI hardware interface
MD_MAX72XX mx = MD_MAX72XX(CS_PIN, MAX_DEVICES);

// Arbitrary pins
// MD_MAX72XX mx = MD_MAX72XX(DATA_PIN, CLK_PIN, CS_PIN, MAX_DEVICES);

// We always wait a bit between updates of the display
#define DELAYTIME 100 // in milliseconds

void use_setPoint()
// Demonstrates the use of setPoint and
// show where the zero point is in the display
{
    mx.clear();

    if (MAX_DEVICES > 1)
        mx.setChar((2*COL_SIZE)-1, '0');

    for (uint8_t i=0; i<ROW_SIZE; i++)
    {
        mx.setPoint(i, i, true);
        mx.setPoint(0, i, true);
    }
}
```

Si lo cambiamos a 1,
permitirá hacer DEBUG,
e imprimir en el monitor
serial, parámetros

Nuestro kit FC16 posee
4 Dispositivos
encadenados

Definición del pinout

Constructor, que crea el objeto mx, a
ser manejado con las funciones
miembros

Una vez creado el objeto mx, se
invocan los métodos con mx.metodo,
Ej: mx.clear, mx.setchar, mx.setPoint

```

        mx.setPoint(i, 0, true);
        delay(DELAYTIME);
    }

    delay(DELAYTIME*3);
}

void lines()
// Demonstrate the use of drawLine().
// fan out lines from each corner for up to 4 device blocks
{
    const uint8_t stepSize = 3;
    const uint8_t maxDev = (MAX_DEVICES > 4 ? 4 : MAX_DEVICES);

    mx.clear();
    for (uint16_t c=0; c<(maxDev*COL_SIZE)-1; c+=stepSize)
    {
        mx.drawLine(0, 0, ROW_SIZE-1, c, true);
        delay(DELAYTIME);
    }

    mx.clear();
    for (uint16_t c=0; c<(maxDev*COL_SIZE)-1; c+=stepSize)
    {
        mx.drawLine(ROW_SIZE-1, 0, 0, c, true);
        delay(DELAYTIME);
    }

    mx.clear();
    for (uint16_t c=0; c<(maxDev*COL_SIZE)-1; c+=stepSize)
    {
        mx.drawLine(ROW_SIZE-1, (MAX_DEVICES*COL_SIZE)-1, 0, (MAX_DEVICES*COL_SIZE)-1-c, true);
        delay(DELAYTIME);
    }

    mx.clear();
    for (uint16_t c=0; c<(maxDev*COL_SIZE)-1; c+=stepSize)
    {
        mx.drawLine(0, (MAX_DEVICES*COL_SIZE)-1, ROW_SIZE-1, (MAX_DEVICES*COL_SIZE)-1-c, true);
        delay(DELAYTIME);
    }
}

void scrollText(char *p)
{
    uint8_t      charWidth;
    uint8_t      cBuf[8];          // this should be ok for all built-in fonts

    mx.clear();

    while (*p != '\0')
    {
        charWidth = mx.getChar(*p++, sizeof(cBuf)/sizeof(cBuf[0]), cBuf);

        for (uint8_t i=0; i<charWidth + 1; i++) // allow space between characters
        {
            mx.transform(MD_MAX72XX::TSL);
            if (i < charWidth)
                mx.setColumn(0, cBuf[i]);
            delay(DELAYTIME);
        }
    }
}

```

La librería posee métodos más potentes que setPoint() (prender el led), como por ejemplo drawLine(), drawRectangle(). Será necesario investigar.

Podemos ver que el scroll de un texto, lo hace el método transform, en este caso, ::TSL (transform shift left), hacia izquierda.

```

void intensity()
// Demonstrates the control of display intensity (brightness) across
// the full range.
{
    uint8_t row;

    mx.clear();

    // Grow and get brighter
    row = 0;
    for (int8_t i=0; i<=MAX_INTENSITY; i++)
    {
        mx.control(MD_MAX72XX::INTENSITY, i);
        if (i%2 == 0)
            mx.setRow(row++, 0xff);
        delay(DELAYTIME*3);
    }

    mx.control(MD_MAX72XX::INTENSITY, 8);
}

```

O que la modificación de la intensidad del display se hace a través del método control().

```

void setup()
{
    mx.begin();

    #if DEBUG
        Serial.begin(57600);
    #endif
}

```

mx.begin inicializa el objeto

```

void loop()
{
    use_setPoint ();
    lines();
    scrollText("Control  ");
    intensity();
}

```

En el loop(), uso los métodos

Para información detallada de la clase:

https://majicdesigns.github.io/MD_MAX72XX/class_m_d__m_a_x72_x_x.html