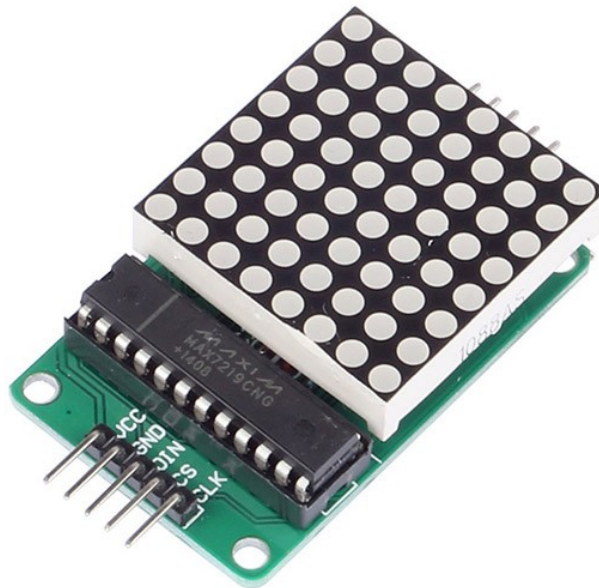


MAX7219 Red LED Dot Matrix Display Module (ME042)



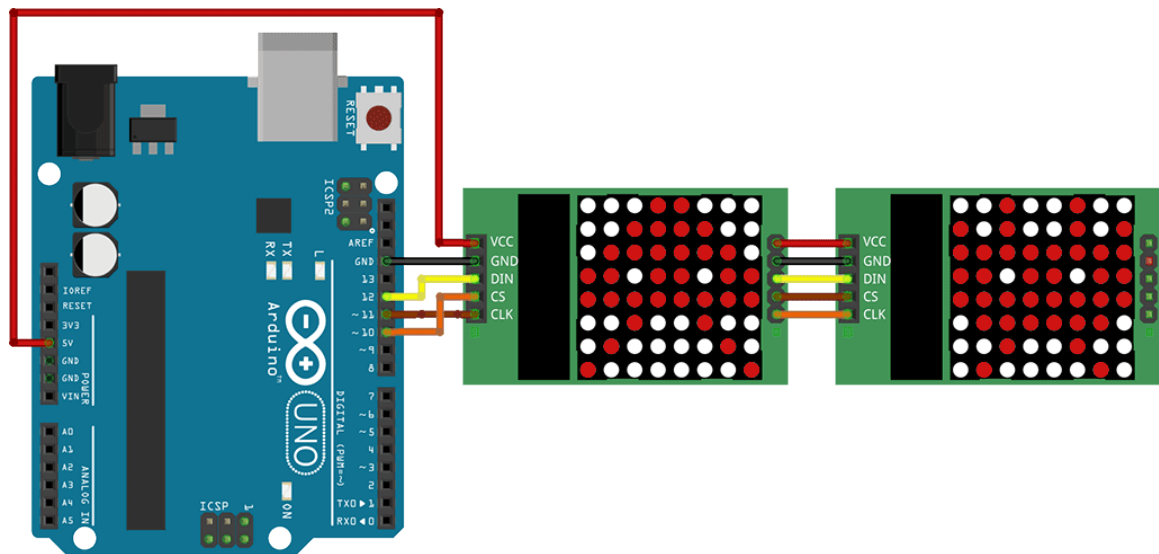
Specification

- Size:50mmx32mmx15mm/1.95"x1.25"x0.58"(inch) (approx)
- Module operating voltage: 5V
- A single module can drive an 8 * 8 common cathode lattice
- With four screws hole, aperture 3mm, by using our M3 Tongzhu fixed.
- Module with input and output interfaces, supports multiple modules cascade.

How to use a MAX7219 LED Dot Matrix with an Arduino

Using a LED Dot Matrix in your next project can be a way to incorporate some cool little animations. By using these 8X8 matrix modules you can create your own. Since these modules use the MAX7219 LED driver chip, we will be able to turn on and off the 64 LEDs of each module, using only 3 pins on our Arduino. In this tutorial we will connect 2 of them together to display different animations on each one.

Wire connection:



fritzing

As you can see, even if we are using 2 Modules, the number of pins needed, does not increase.

VCC and Ground are connected to the Arduino.

Pin 12 is connected to DIN, Pin 11 is connected to CLK and Pin 10 is connected to CS.

Example

Our Sketch will make use of the "LedControl" Library to communicate with the MAX7219 modules.

Library download: <https://github.com/wayoda/LedControl/releases>

Download and extract it to your Library folder, then restart your IDE software.

The code below assumes that we have 2 Dot Matrix connected together in cascade, but you can modify it easily if you have more or less of them connected.

We select the module we want to write to by using a number based on where it is in the chain, starting from 0 for the first one, 1 for the second, and so on...

*****Code Begin*****

```
#include "LedControl.h"
```

```
LedControl lc=LedControl(12,11,10,2); // Pins: DIN,CLK,CS, # of Display  
connected
```

```
unsigned long delayTime=200; // Delay between Frames
```

```
// Put values in arrays
```

```
byte invader1a[] =
```

```
{
```

```
    B00011000, // First frame of invader #1
```

```
    B00111100,
```

```
    B01111110,
```

```
    B11011011,
```

```
    B11111111,
```

```
    B00100100,
```

```

        B01011010,
        B10100101
    };

    byte invader1b[] =
    {
        B00011000, // Second frame of invader #1
        B00111100,
        B01111110,
        B11011011,
        B11111111,
        B00100100,
        B01011010,
        B01000010
    };

    byte invader2a[] =
    {
        B00100100, // First frame of invader #2
        B00100100,
        B01111110,
        B11011011,
        B11111111,
        B11111111,
        B10100101,
        B00100100
    };

    byte invader2b[] =
    {
        B00100100, // Second frame of invader #2
        B10100101,
        B11111111,
        B11011011,
        B11111111,
        B01111110,
        B00100100,
        B01000010
    };

    void setup()

```

```
{
    lc.shutdown(0,false); // Wake up displays
    lc.shutdown(1,false);
    lc.setIntensity(0,5); // Set intensity levels
    lc.setIntensity(1,5);
    lc.clearDisplay(0); // Clear Displays
    lc.clearDisplay(1);
}
```

// Take values in Arrays and Display them

```
void sinvader1a()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(0,i,invader1a[i]);
    }
}
```

```
void sinvader1b()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(0,i,invader1b[i]);
    }
}
```

```
void sinvader2a()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(1,i,invader2a[i]);
    }
}
```

```
void sinvader2b()
{
    for (int i = 0; i < 8; i++)
    {
        lc.setRow(1,i,invader2b[i]);
    }
}
```

```
}

void loop()
{
  // Put #1 frame on both Display
  sinvader1a();
  delay(delayTime);
  sinvader2a();
  delay(delayTime);

  // Put #2 frame on both Display
  sinvader1b();
  delay(delayTime);
  sinvader2b();
  delay(delayTime);

}
```

*****Code End*****