



## 64x32 RGB LED Matrix - 4mm pitch SKU:DFR0460



### Introduction

City night is always beautiful. She is just like a charming girl, showing her beauty every time. The colorful light is jewelry, dotted on her dress.

This is a 64x32 RGB LED Matrix Panel, it has 2048 full-color RGB LEDs in all. Each LED can be independently addressed and controlled. It requires at least 13 digital GPIOs to control the LED matrix. So the UNO board won't be a good choice in this application, recommended Mega 2560, Raspberry Pi and the other kinds of microcontroller with large RAM and high speed.

The led matrix has 2 IDC connectors (DATA\_IN, DATA\_OUT) on the back, you can cascade multiple panels and make a huge screen together. BUT Arduino doesn't support this function, its speed is not enough to multiple panel.

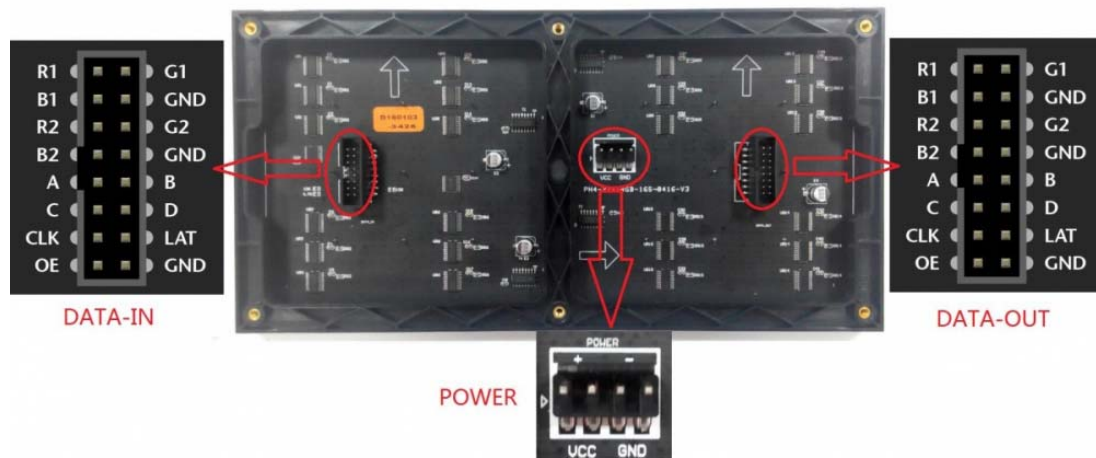
Whats more, It is a high brightness, long life, no pollution, pure color LED display module. It can be used both indoor and outdoor, safety and stability, the module can not be mutually extrusion deformation, also can be used normally in harsh environment .

## Specification

- Operating voltage : DC 5V
- Average power consumption : <math><500W/m^2</math>
- Maxim Power Consumption : <math><1000w/m^2</math>
- Pixel : 64\*32=2048
- Level of viewing Angle :  $\geq 160^\circ$
- Control mode : synchronous control
- Drive mode : 1/16 scanning
- Repetition frequency :  $\geq 60\text{Hz}$
- White Balance Brightness :  $\geq 1200\text{cd}/m^2$
- Refresh frequency :  $\geq 300\text{Hz}$
- Pixel pitch : 4mm
- Dimension : 125mm\*250mm
- Thickness : 11mm

## Board Overview

**Note** : The pin order of DATA-IN and ADTA-OUT is same, POWER SUPPLY 5V.



DATA-IN and DATA-OUT		
Label	Name	Function
1	DR1	High R data
2	DG1	High G data
3	DB1	High B data
4	GND	GND

5	DR2	Low R data
6	DG2	Low G data
7	DB2	Low B data
8	GND	GND
9	A	A line selection
10	B	B line selection
11	C	C line selection
12	D	D line selection
13	CLK	CLOCK
14	LAT	LATCH
15	OE	Output Enable
16	GND	GND
POWER		
Label	Name	Function
1	VCC	5V
2	VCC	5V
3	GND	GND
4	GND	GND

## Tutorial

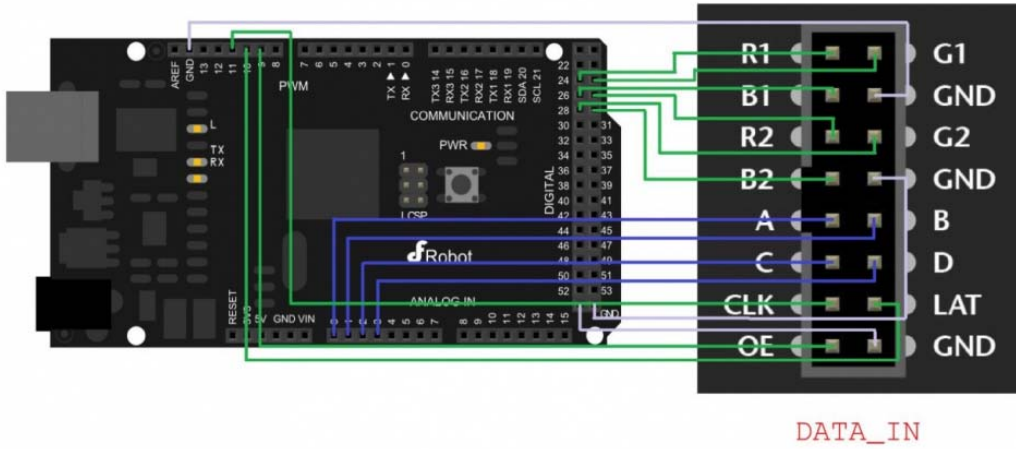
According to the pinout to connect, then upload the code to MEGA, you will be able to see a beautiful display effect.

## Requirements

- **Hardware**  
MEGA controller X1  
DFR0460 X1  
DuPont cables

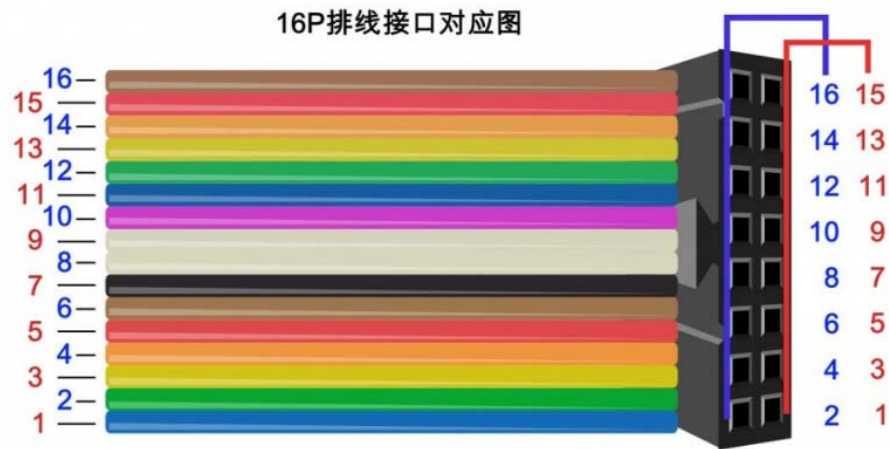
- **Software**  
 Arduino IDE [Click to Download Arduino IDE from Arduino®](https://www.arduino.cc/en/Main/Software)  
<https://www.arduino.cc/en/Main/Software>

### Connection Diagram



### DFR0460 Diagram

### 16P Interface Diagram



### DFR0460 Diagram

### Sample Code

Click to download the library [Adafruit-GFX-Library](https://github.com/adafruit/Adafruit-GFX-Library). [RGB-matrix-Panel](https://github.com/adafruit/Adafruit-GFX-Library). [How to install the library?](https://github.com/adafruit/Adafruit-GFX-Library) <https://github.com/adafruit/Adafruit-GFX-Library/archive/master.zip>

<https://github.com/adafruit/Adafruit-GFX-Library/archive/master.zip>  
<http://www.dfrobot.com.cn/community/forum.php?mod=viewthread&tid=1854&page=1&extra=#pid6955>

```

/*****
*
*NOTE THIS CAN ONLY BE USED ON A MEGA! NOT ENOUGH RAM ON UNO!
*
*****
* 64x32 RGB LED Matrix - 4mm pitch
* *****
* testshapes demo for RGBmatrixPanel library.
* Demonstrates the drawing abilities of the RGBmatrixPanel library.
* For 32x64 RGB LED matrix.

* @author lg.gang(lg.gang@qq.com)
* @version V1.0
* @date 2016-9-6

* GNU Lesser General Public License.
* See <http://www.gnu.org/licenses/> for details.
* All above must be included in any redistribution
* *****/
#include <Adafruit_GFX.h> // Core graphics library
#include <RGBmatrixPanel.h> // Hardware-specific library

#define CLK 11
#define OE 9
#define LAT 10

#define A A0
#define B A1
#define C A2
#define D A3

RGBmatrixPanel matrix(A, B, C, D, CLK, LAT, OE, false, 64);

```

```
void setup() {
    delay(1000);
    matrix.begin();
}

void loop() {

    // draw a pixel in solid white
    matrix.drawPixel(0, 0, matrix.Color333(7, 7, 7));
    delay(5000);

    // fix the screen with green
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.C
olor333(0, 7, 0));
    delay(5000);

    // fix the screen with white
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.C
olor333(7, 7, 7));
    delay(5000);

    // fix the screen with red
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.C
olor333(7, 0, 0));
    delay(5000);

    // fix the screen with blue
    matrix.fillRect(0, 0, matrix.width(), matrix.height(), matrix.C
olor333(0, 0, 7));
    delay(5000);
    // fill the screen with 'black'
    matrix.fillScreen(matrix.Color333(0, 0, 0));

    // draw a box in yellow
```

```
matrix.drawRect(0, 0, matrix.width(), matrix.height(), matrix.Color333(7, 7, 0));
delay(5000);

// draw a box in fuchsia
matrix.drawRect(5, 5, 53, 23, matrix.Color333(7, 0, 7));
delay(5000);

// draw a box in yellow
matrix.drawRect(10, 10, 43, 13, matrix.Color333(7, 1, 3));
delay(5000);

// draw an 'X' in red
matrix.drawLine(0, 0, matrix.width()-1, matrix.height()-1, matrix.Color333(7, 0, 0));
matrix.drawLine(matrix.width()-1, 0, 0, matrix.height()-1, matrix.Color333(7, 0, 0));
delay(5000);

// draw a blue circle
matrix.drawCircle(10, 10, 10, matrix.Color333(0, 0, 7));
delay(5000);

// fill a violet circle
matrix.fillCircle(40, 21, 10, matrix.Color333(7, 0, 7));
delay(5000);

// fill the screen with 'black'
matrix.fillScreen(matrix.Color333(0, 0, 0));

// draw some text!
matrix.setTextSize(1); // size 1 == 8 pixels high
matrix.setTextWrap(false); // Don't wrap at end of line - will do ourselves
matrix.setCursor(3, 0); // start at top left, with 3 pixel of spacing
```

```

uint8_t w = 0;
char *str = "Welcome ToDFROBOT";
for (w=0; w<10; w++) {
    matrix.setTextColor(Wheel(w));
    matrix.print(str[w]);
}

matrix.setCursor(13, 8);    // next line
for (w=10; w<17; w++) {
    matrix.setTextColor(Wheel(w));
    matrix.print(str[w]);
}
matrix.println();
matrix.setCursor(2, 16);
matrix.setTextColor(matrix.Color333(7,7,7));
matrix.println("I'm always");

// print each letter with a rainbow color
matrix.setCursor(3, 24);
matrix.setTextColor(matrix.Color333(7,0,0));
matrix.print('B');
matrix.setTextColor(matrix.Color333(7,4,0));
matrix.print('y');
matrix.setTextColor(matrix.Color333(7,7,0));
matrix.print(' ');
matrix.setTextColor(matrix.Color333(4,7,0));
matrix.print('U');
matrix.setTextColor(matrix.Color333(0,7,0));
matrix.print(' ');
matrix.setTextColor(matrix.Color333(0,7,7));
matrix.print("S");
matrix.setTextColor(matrix.Color333(0,4,7));
matrix.print('i');
matrix.setTextColor(matrix.Color333(0,0,7));

```



```

matrix.print('d');
matrix.setTextColor(matrix.Color333(4,0,7));
matrix.print("e");
matrix.setTextColor(matrix.Color333(7,0,4));
matrix.println("!");
delay(50000);

}

// Input a value 0 to 24 to get a color value.
// The colours are a transition r - g - b - back to r.
uint16_t Wheel(byte WheelPos) {
  if(WheelPos < 8) {
    return matrix.Color333(7 - WheelPos, WheelPos, 0);
  } else if(WheelPos < 16) {
    WheelPos -= 8;
    return matrix.Color333(0, 7-WheelPos, WheelPos);
  } else {
    WheelPos -= 16;
    return matrix.Color333(0, WheelPos, 7 - WheelPos);
  }
}
}

```

## Expected Results

The LED module will take turns display: a white point, full screen green, full screen white, full screen red, a yellow rectangle, a fuchsia rectangle, a yellow rectangle, a red X and a blue circle, filled with a purple circle, "Welcome ToDFROBOT I'm always By U Side!". .

For any questions, advice or cool ideas to share, please visit the [DFRobot Forum](#).