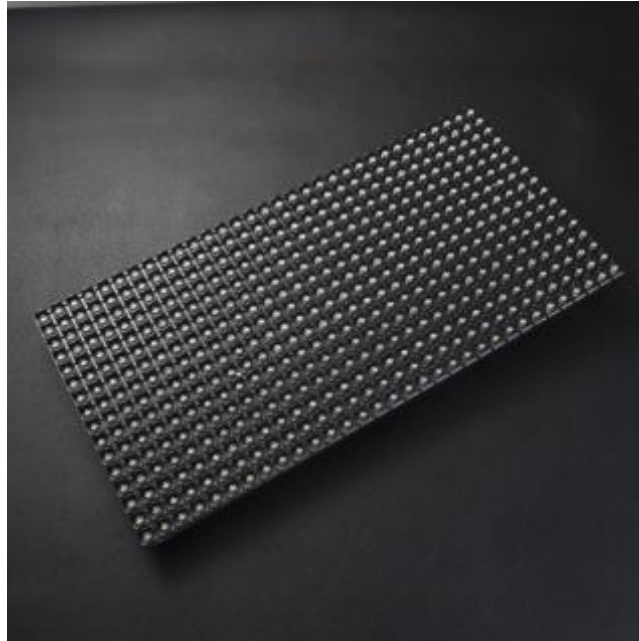# User Manual
of
# 32*16 LED Matrix Panel White(AD026)

This provides step by step approach to driving a Dot Matrix Display (DMD) Panel with an Arduino. The DMD is a 512 pixels single colour LED display arranged in 32x16 layout, a 16 pin (2x8) IDC connector is used to interface with Arduino.

  In order to drive the 32x16 Dot Matrix Display Panel from Arduino, the DMD library for Arduino is required.The Freetronics DMD library is able to write letter and text on the display board with limited function, variable string and text centering is not supported. Yeah! now you can use my code to write variable string on the display board and centering it horizontally.
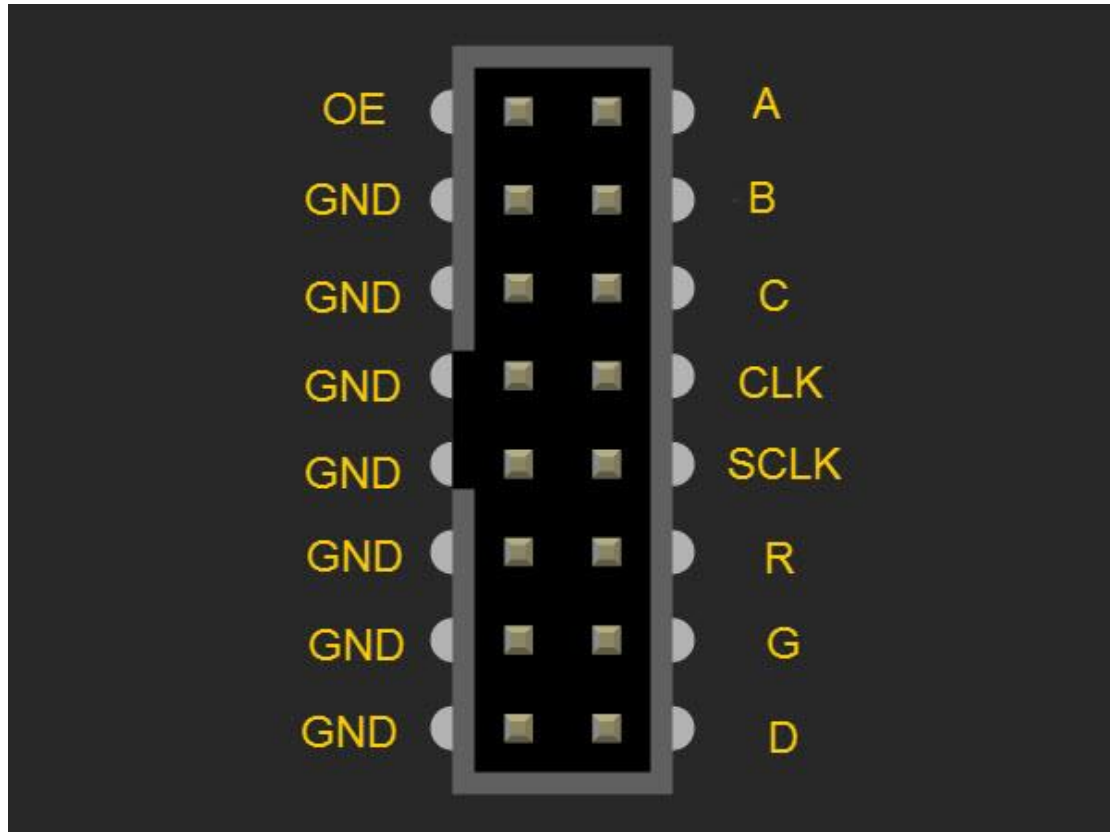
DMD Ribbon Connector Input   High brightness DMD 5V power input terminals   Output to more displays

HUB1 2   74HC245   74HC138   5V power   Data output

SN74HC04

Data input   74HC595   APM4953
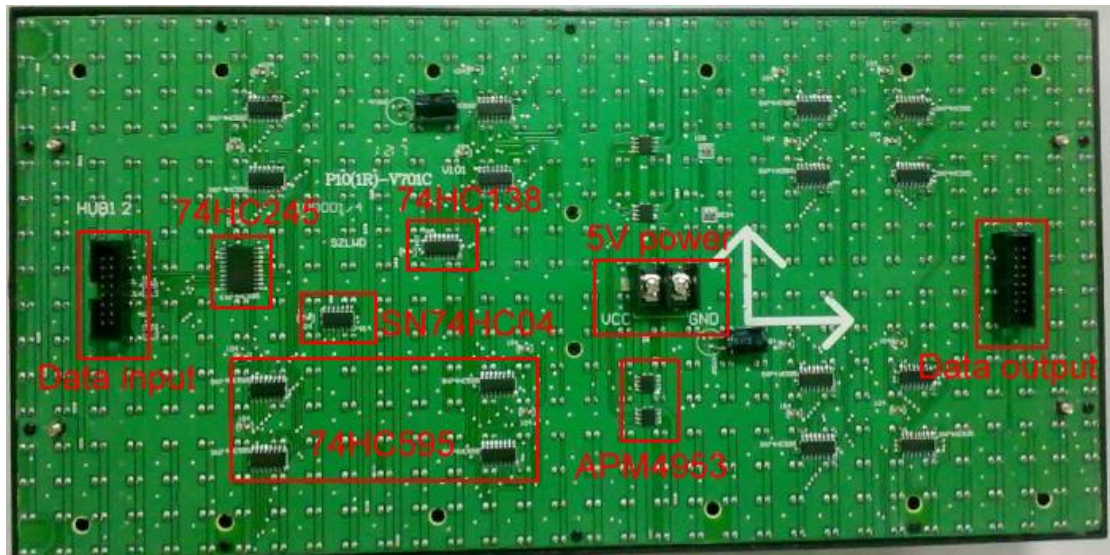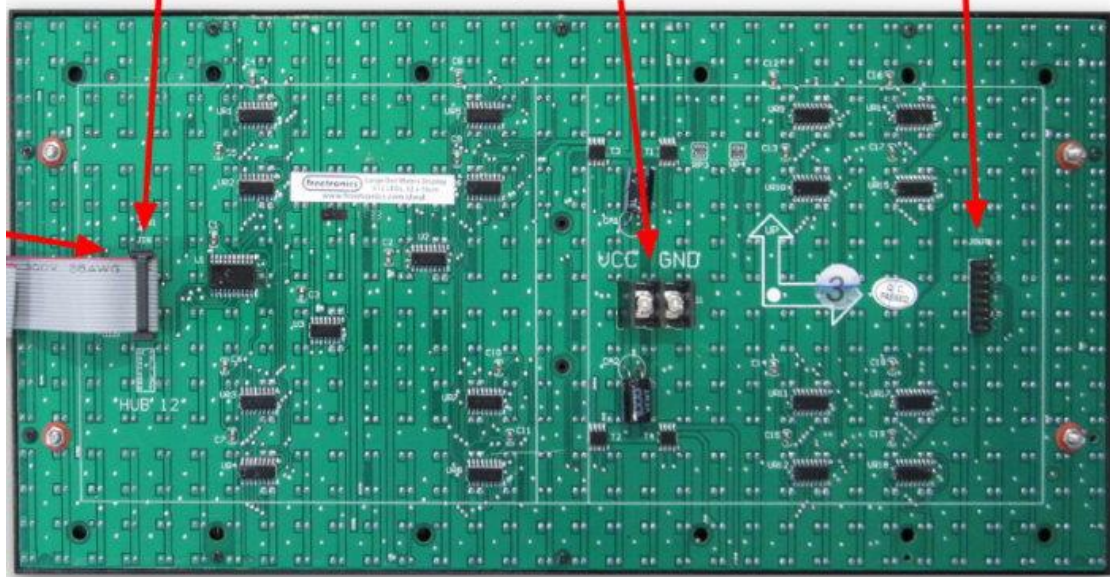
Pinout

OE - Enable / Disable ALL the LEDs on the DMD

A - pins select which 1/4 of the DMD is selected

B - pins select which 1/4 of the DMD is selected

C - pins select which 1/4 of the DMD is selected

D - pins select which 1/4 of the DMD is selected

CLK - Used to clock each pixel into the DMD shift registers

SCLK - Latches the current content of the DMD shift registers

R - The raw pixel data stream (NB: The 'R' stands for RED. You might notice an unused 'G' pin which is for Green on a HUB12 connector)

Library installation

The Freetronics DMD library must work together with TimerOne library, the TimerOne library is available here.

Install TimerOne library

- Download the TimerOne library, it should be TimerOne-r11.zip
- Create a TimerOne folder in the Arduino libraries folder, for my case, it is on \My Documents\Arduino\libraries\
- Extract all the files & folders from TimerOne-r11.zip to the TimerOne folder.

Install Freetronics DMD library

- Download the Freetronics DMD library, it should be DMD-master.ZIP
- Extract it to the Arduino libraries folder, eg. \My Documents\Arduino\libraries\
- Rename the folder from DMD-master to DMD

Upload sketch

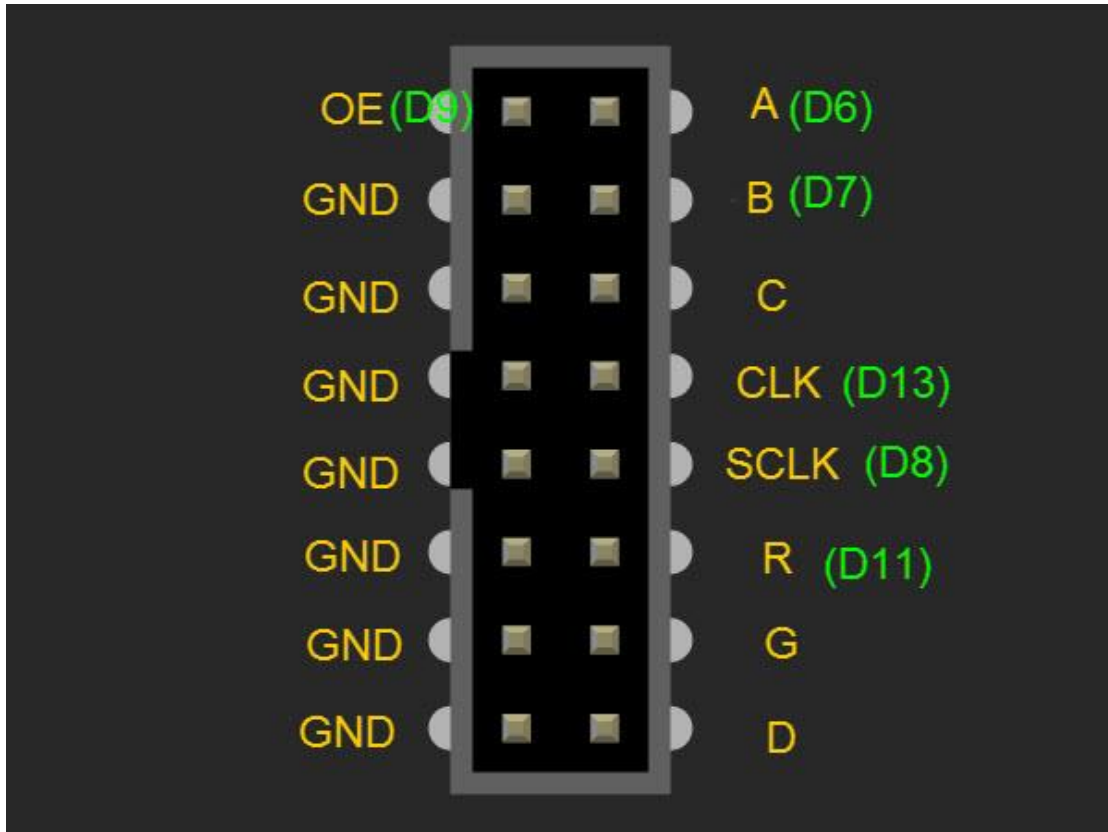Once libraries installation completed, upload the DMD demo sketch to the DIY Arduino

- Open Arduino IDE
- Select File>Examples>DMD>dmd_demo

Example

This example write variable text on the display board and centering the text horizontally.
Wire connection to Arduino UNO board as below:

OE   ========   D9
GND ========   GND
A   ========   D6
B   ========   D7
C   ========   No connection
CLK   ========   D13
SCLK =========   D8
R   ========   D11
G   ========   No connection
D   ========   No connection

```
#include "SPI.h"
#include "DMD.h"
#include "TimerOne.h"
#include "Arial_black_16.h"

/*-------------------------------------------------------------------
 Only 7 of the 16 pins on the DMD are actually used:
 GND - Hopefully obvious
 nOE - Enable / Disable ALL the LEDs on the DMD
 A - A&amp;B pins select which 1/4 of the DMD is selected
 B - A&amp;B pins select which 1/4 of the DMD is selected
 CLK - Used to clock each pixel into the DMD shift registers
 SCLK - Latches the current content of the DMD shift registers
 R - The raw pixel data stream (NB: The 'R' stands for RED. You might notice an unused 'G' pin which is for
 Green on a HUB12 connector)
 --------
 nOE(D9) |1        2| A(D6)
 Gnd     |3        4| B(D7)
 Gnd     |5        6| C
 Gnd     |6        8| CLK(D13)
```

```
  Gnd      |7     10| SCLK(D8)
  Gnd      |11    12| R(D11)
  Gnd      |13    14| G
  Gnd      |15    16| D
  ---------

  ----------------------------------------------------------------*/
//Fire up the DMD library as dmd
#define DISPLAYS_ACROSS 1
#define DISPLAYS_DOWN 1
DMD dmd(DISPLAYS_ACROSS, DISPLAYS_DOWN);

/*----------------------------------------------------------------
  Interrupt handler for Timer1 (TimerOne) driven DMD refresh scanning, this gets
   called at the period set in Timer1.initialize();
   ----------------------------------------------------------------*/
void ScanDMD()
{
    dmd.scanDisplayBySPI();
}

const byte PanelWidth = 32;
const byte MaxStringLength = 5;
char CharBuf[MaxStringLength + 1];

void setup() {
    //initialize TimerOne's interrupt/CPU usage used to scan and refresh the display
    Timer1.initialize( 5000 );              //period in microseconds to call ScanDMD. Anything longer than 5000
(5ms) and you can see flicker.
    Timer1.attachInterrupt( ScanDMD );     //attach the Timer1 interrupt to ScanDMD which goes to
dmd.scanDisplayBySPI()
    dmd.selectFont(Arial_Black_16);
}

void loop() {
    String myString;
    myString= "0"; center_theDisplay(myString); delay(1000);
    myString= "12"; center_theDisplay(myString); delay(1000);
    myString= "345"; center_theDisplay(myString); delay(1000);
    myString= "678."; center_theDisplay(myString); delay(1000);
}

void center_theDisplay(String input_Str) {
    byte charCount, total_charWidth, x_position;
    input_Str.toCharArray(CharBuf, MaxStringLength + 1); //string to char array
```

```
charCount=    input_Str.length();
if (charCount==0) exit;

total_charWidth= 0;
for (byte thisChar = 0; thisChar <charCount; thisChar++) {
    total_charWidth= total_charWidth + dmd.charWidth(CharBuf[thisChar]) +1; //add 1 pixel for space
}

total_charWidth= total_charWidth -1; //no space for last letter
x_position= (PanelWidth - total_charWidth) /2; //position(x) of first letter
dmd.clearScreen(true);

for (byte thisChar = 0; thisChar <charCount; thisChar++) {
    //dmd.drawChar(x, y,'@', GRAPHICS_NORMAL)
    dmd.drawChar( x_position,    1, CharBuf[thisChar], GRAPHICS_NORMAL );
    x_position= x_position + dmd.charWidth(CharBuf[thisChar]) + 1; //position for next letter
}
```
******Code End******

Reference Commended:

http://www.freetronics.com.au/products/dot-matrix-display-32x16-red#.WA3GJ1R97s0

https://learn.adafruit.com/32x16-32x32-rgb-led-matrix/new-wiring