$ AUD ▾   **Your cart is empty (/cart)**
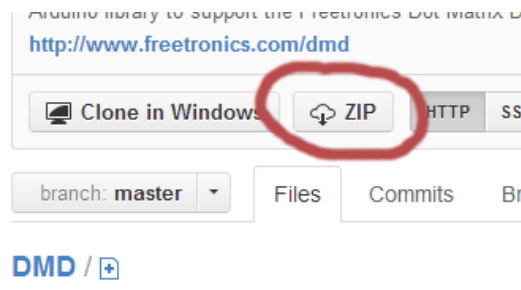
Using Your Freetronics DMD

 This guide continues from the *Quickstart Guide* that is included with your DMD, and also available for download from here
(http://cdn.shopify.com/s/files/1/0045/8932/files/DMD_Getting_Started.pdf?100647).

8. Install Libraries

We're using the latest versions of the Arduino IDE which is currently v1.0.5 or 1.5.2. Please note that the DMD is not compatible with the Arduino Due board.

Next, download the "TimerOne" library from https://code.google.com/p/arduino-
timerone/downloads/list (https://code.google.com/p/arduino-timerone/downloads/list).
Extract the folder from the zip file, rename it to "TimerOne" and then copy it into your …
\arduino-1.0.x\libraries folder.

Finally, download the DMD library from https://github.com/freetronics/DMD
(https://github.com/freetronics/DMD). The best way is to click on the zip icon, for example:

Once downloaded, extract and rename the folder to DMD - and once again copy it to the Arduino libraries folder as you did for TimerOne. Finally, close all Arduino IDE windows if any are open, then restart the IDE.

Pre-requisite code to enable your DMD

No matter what you're doing, *the following code needs to be inserted into your sketch to use a DMD:*

Featured Products

EtherTen (100% Arduino compatible with onboard Ethernet) (/products/etherten)

(/products/etherten)

Bluetooth Shield (/products/btsh-bluetooth-shield)

(/products/btsh-bluetooth-shield)

GPS Module (/products/gps)

(/products/gps)

Zero Insertion Force (ZIF) Breakout for 28-Pin AVR (/products/zero-insertion-force-zif-breakout-for-28-pin-avr)

(/products/zero-insertion-force-zif-
breakout-for-28-pin-avr)

StepDuino Stepper Motor Controller (/products/stepduino-stepper-motor-controller)

(/products/stepduino-stepper-
motor-controller)

```
  #include "SPI.h"
  #include "DMD.h"
  #include "TimerOne.h"
  #include "Arial_black_16.h"<arial_black_16.h>
  // you can remove the fonts if unused
  #include "SystemFont5x7.h"
  #define DISPLAYS_ACROSS 1
  #define DISPLAYS_DOWN 1
  /* change these values if you have more than one DMD connected */
  DMD dmd(DISPLAYS_ACROSS,DISPLAYS_DOWN);

  void ScanDMD()
  {
    dmd.scanDisplayBySPI();
  }

  void setup()
  {
     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.clearScreen( true );
  /* true is normal (all pixels off), false is negative (all pixels on)
  */
  }
```

Displaying Text on your DMD

Now we'll run through the functions required to display text, in both static and scrolling modes on your DMDs. Remember that the DMD has sixteen rows of thirty-two LEDs - and whenever referencing a position on the DMD please note that the possible values for the x-coordinate is 0~31 and the y-coordinate is 0~15 - with 0,0 being the top-left of the DMD and 31,15 being the bottom-right.

To display text, clear the DMD and so on - choose from the following functions as described below.

To turn off all the pixels, use:

```
  dmd.clearScreen( true);
```

or to turn them all on, use

```
  dmd.clearScreen( false );
```

Note that when all the pixels are on, the current draw for each DMD is at the maximum. When displaying text, you first need to select a font. Currently there are two fonts available - a small one which is selected with:

```
  dmd.selectFont( SystemFont5x7 );
```

and a larger one which is selected with:

```
  dmd.selectFont( Arial_Black_16 );
```

Either of those functions are used before displaying text. There are functions to display a single character, or a line of text. For a character, use:

```
  dmd.drawChar(  x,  y, '@', GRAPHICS_NORMAL );
```

where *x* and *y* are the coordinates for the top-left pixel of the character to be displayed; and @ is the character to be displayed. For example, using the following:

```
  dmd.selectFont(SystemFont5x7);
  dmd.drawChar(  5,  5, 'X', GRAPHICS_NORMAL );
```
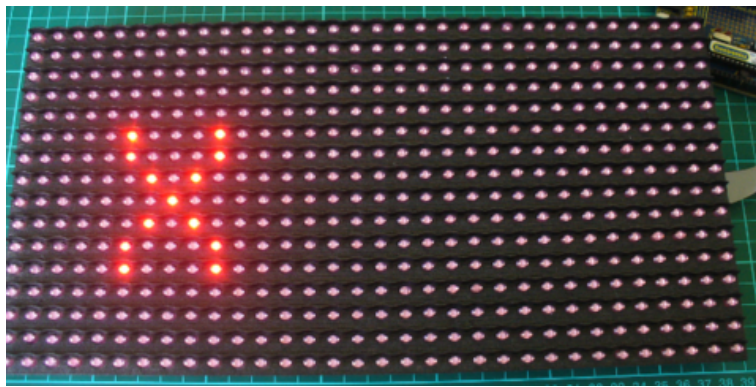
results with the following:

Note that the examples displayed in this guide are not using external power. When doing so the brightness is to strong for the camera!

To display a line of text, use the following:

```
dmd.drawString( x,y, "text,", z, GRAPHICS_NORMAL );
```

where *x* and *y* are the coordinates for the top-left pixel of the first character in the line of text being displayed, *text* is the text to display, and *z* is the length of the text to display in characters. For example,

```
dmd.selectFont(SystemFont5x7);
dmd.drawString( 0,0, "Hello,", 5, GRAPHICS_NORMAL );
dmd.drawString( 2,9, "world,", 5, GRAPHICS_NORMAL );
```

results with:
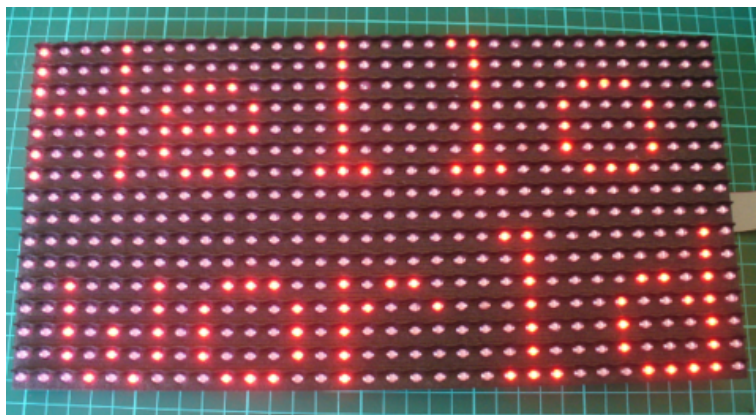
Although static text displays can be useful, scrolling text across the DMD is even better. And if you have more than one DMD - the text will scroll from right to left across the displays. To do so, have your sketch first assemble the text to display into a String variable (http://arduino.cc/en/Reference/StringObject). The maximum length is 255 characters. Or you can just have a fixed value, for example:

```
String textToScroll = "Hello, this is a really long length of text";
```

You will also need the following function in your sketch:

```
void drawText(String dispString)
{
  dmd.clearScreen( true );
  dmd.selectFont( Arial_Black_16 );
  char newString[256];
  int sLength = dispString.length();
  dispString.toCharArray( newString, sLength+1 );
  dmd.drawMarquee(newString,sLength,( 32*DISPLAYS_ACROSS )-1 , 0 );
  long start=millis();
  long timer=start;
  long timer2=start;
  boolean ret=false;
  while(!ret){
    if ( ( timer+20 ) < millis() ) {
      ret=dmd.stepMarquee( -1 , 0 );
      timer=millis();
    }
  }
}
```

Then to scroll the text, simply call the function as required, for example:

```
drawText(textToScroll);
```

The following is the total sketch for a basic text-scrolling application using the function from above:

```
#include "SPI.h"
#include "DMD.h"
#include "TimerOne.h"
#include "Arial_black_16.h"
/* you can remove the fonts if unused */
#define DISPLAYS_ACROSS 1
#define DISPLAYS_DOWN 1
DMD dmd( DISPLAYS_ACROSS , DISPLAYS_DOWN );

void ScanDMD()
{
   dmd.scanDisplayBySPI();
}

void setup()
{
    Timer1.initialize( 5000 );
    Timer1.attachInterrupt( ScanDMD );
    dmd.clearScreen( true );
}

String textToScroll="Hello, this will be displayed on the DMD";

void drawText( String dispString )
{
   dmd.clearScreen( true );
   dmd.selectFont( Arial_Black_16 );
   char newString[256];
   int sLength = dispString.length();
   dispString.toCharArray( newString, sLength+1 );
   dmd.drawMarquee( newString , sLength , ( 32*DISPLAYS_ACROSS )-1
,0);
   long start=millis();
   long timer=start;
   long timer2=start;
   boolean ret=false;
   while( !ret ){
     if ( ( timer+20 ) < millis() ) {
       ret=dmd.stepMarquee( -1 , 0 );
       timer=millis();
     }
   }
}

void loop()
{
   drawText(textToScroll);
}
```
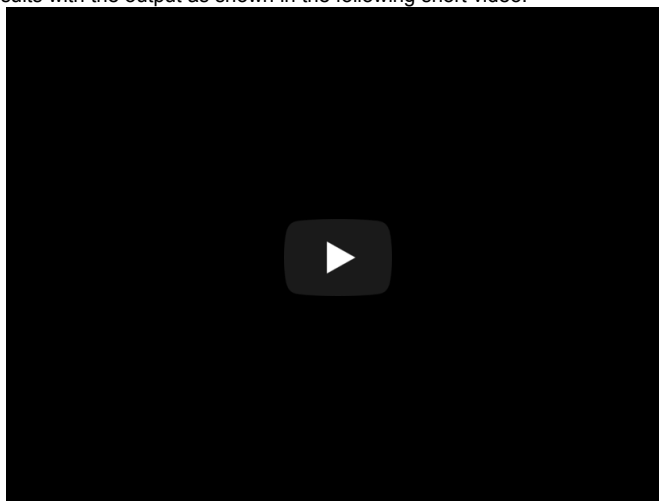
Which results with the output as shown in the following short video:

Displaying graphics on your DMD
Using the functions described below and a little imagination you can create all sorts of images and data visualisations. To turn an individual pixel on or off, use:

```
dmd.writePixel( x, y, GRAPHICS_NORMAL, z );
```

where x and y are the coordinates of the pixel, and z turns the pixel on (with 1) or off (with 0). To draw a circle with the centre at x, y and a radius of r, use:

```
dmd.drawCircle( x,  y,  r, GRAPHICS_NORMAL );
```

To draw a line (straight or otherwise) from location $x1, y1$ to $x2, y2$ use:

```
dmd.drawLine(  x1, y1, x2, y2, GRAPHICS_NORMAL );
```

To draw a rectangle with the top-left at *x1, y1* and the bottom-right at *x2, y2* use:

```
dmd.drawBox(x1, y1, x2, y2, GRAPHICS_NORMAL );
```

and you can draw a filled rectangle with:

```
dmd.drawFilledBox(x1, y1, x2, y2, GRAPHICS_NORMAL );
```

Moving forward
You can see these functions demonstrated using the sketch

```
dmd_demo.ino
```

provided with the DMD library. Furthermore check out the Freetronics DMD forum (http://forum.freetronics.com/viewforum.php?f=26) where you can find Q&A, demonstrations of user's projects, discussion and troubleshooting for beginners.

699

Privacy Policy (/pages/privacy-policy)

RoHS (/pages/rohs)

Terms & Conditions (/pages/terms-conditions)

Distributor Login (/account)

PAYMENT METHODS WE ACCEPT