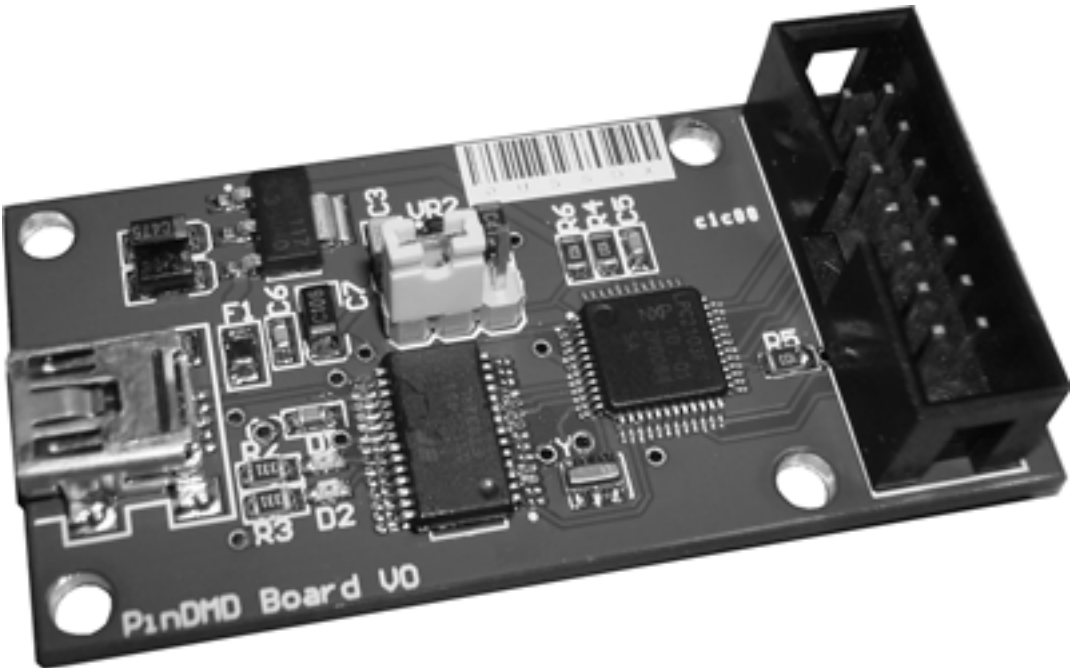


pinDMD Application Note AN001



INTRODUCTION

The pinDMD board will show up as a virtual com port device and there for can be interfaced with any application capable of writing to the computers com ports.

FTDI's D2XX drivers can also be used to talk directly with the pinDMD. This would be the preferred method as it has far greater control over the USB bus.

INTERFACING WITH THE PINDMD

The pinDMD expects a stream of 1048 bytes to display one 3 colour 128x32 frame. This process is repeated to create animations. The pinDMD can receive frames at around 30fps.

Frame Data Packet

Each frame packet is made up of 3 preamble bytes, 1 command byte and 1024 data bytes. The preamble bytes are used to keep the pinDMD in sync with the computer. The command byte is currently not used. The data bytes hold the image data.

Example Frame Packet

[0x81][0xC3][0xE7][0x0][DataByte x 1024]....

Data Structure

Each 128x32px three colour + black image is constructed using two 512 byte sub frames. One is the high frame and the other is the low frame. Each sub frame is a 128x32px single colour image. The idea is flipping between the two sub frames very quickly gives the illusion of three colours. (The pinDMD takes care of the frame refreshing)

When creating the two sub frames. All dim pixels are only placed onto the low frame. All medium pixels are only placed onto the high frame. And all fully lit pixels are placed on both frames.

Intensity	Low Frame	High Frame
Unlit		
Dim	X	
Medium		X
Full	X	X

The low frame 512 bytes are sent first followed by the high frame 512 bytes.

EXAMPLE CODE

This example creates a three colour 128x32pixel frame data packet byte array ready to be sent to the pinDMD board.

```
frame_buf[0] = 0x81;           // frame sync bytes
frame_buf[1] = 0xC3;
frame_buf[2] = 0xE7;
frame_buf[3] = 0x0;           // command byte
byteldx = 4;

// dmd height
for(y = 0; y < 32; y++) {
    // dmd width
    for(x = 0; x < 128; x+=8){
        bd0 = 0;
        bd1 = 0;
        for (v = 7; v >= 0; v--) {
            pixel = getPixelColour((7-v) + x, y);    // pixel colour at x,y of bitmap
            bd0 <<= 1;
            bd1 <<= 1;
            if(pixel==1)
                bd0 |= 1;
            if(pixel==2)
                bd1 |= 1;
            if(pixel==3){
                bd0 |= 1;
                bd1 |= 1;
            }
        }
        frame_buf[byteldx] = bd0;
        frame_buf[byteldx+512] = bd1;
        byteldx++;
    }
}
```