Device: PLT-1001

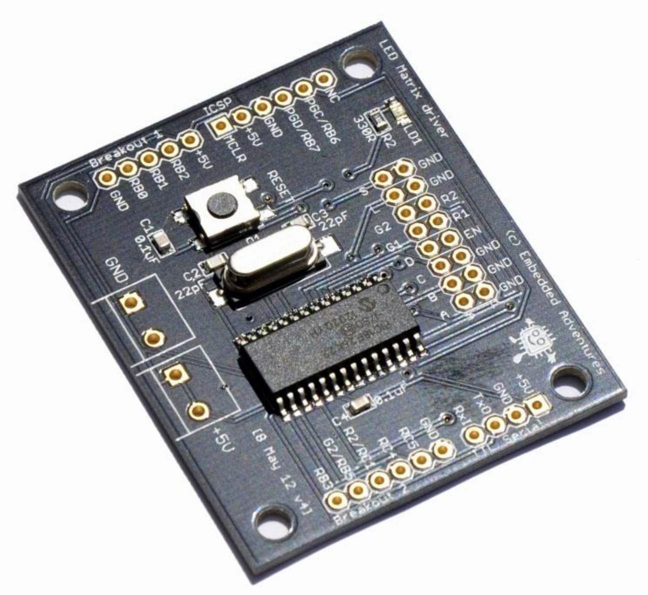
This document Version: 3.0

For hardware Version: 4

For firmware Version: 2.10

Date: April 2014

Description: LED Matrix Display Driver board



Contents

Introduction

The PLT-1001v4 is a driver board designed to drive dynamically-scanned LED matrix displays (LED display panels) such as the LDP-3208S, LDP-8008, LDP-6416 and LDP-6432 from Embedded Adventures. Other panels that have similar pin-outs will work as well. Other panels that have different pin-outs but work the same way, will work with a little jumper wiring.

You don't need to know anything about the microcontroller on board to use the driver board however – it takes simple commands over the TTL serial port to enable rich graphical displays on common LED matrix display panels. Of course, if you want to, you can completely hack the firmware to change it in any way you like, but this is not necessary for normal operation.

Features

The PLT-1001 is designed to make it easy to interface to LED matrix displays. These displays require constant updating which consumes microcontroller CPU. This driver board allows graphical and text functions to be displayed through simple serial commands.

Hackability

The PLT-1001 is 100% hackable.

At Embedded Adventures, we believe you have the most fun when you have the most control over your hardware. For the PLT-1001 we provide a datasheet, complete schematic and complete source code. After that, it's all up to you. We'd love to hear about the projects you're using it for – send us information and photos to myproject@embeddedadventures.com

Construction

The PLT-1001v4 can be back-pack connected to LDP-8008 displays by soldering an 8x2 0.1" female connector on the underside of the board (so the pins are soldered on top). The holes on the PLT-1001v4 are designed to match the holes on the LDP-8008.

It can also be back-pack connected to LDP-6416 or LDP-6432, or alternatively a male connector can be soldered to the top side of the board and the supplied IDC cable can be used to connect to the display.

For power, the PLT-1001 requires 5V and ground connections. Solder the terminal connectors on the board for easy screw connections, or simply power it through the 5V pin on the serial header.

Connections

The PLT-1001 has five connection ports. The breakout ports will be used for future expansion. The ICSP port is used for reprogramming the driver board using a PicKit or similar, but is not required for normal use.

TTL SERIAL	TTL level serial port
ICSP	Programming port for PIC, RB6, RB7 connections
Breakout 1	Breakout pins GND, RB0, RB1, RB2, +5V
Breakout 2	RB3, RB5, RC1, RC4, RC5, GND
LEDPANEL	16 pin connection to LED display panel

Power

The PLT-1001 provides a terminal block for power input for connection to your LED matrix display(s). The PLT-1001 is designed to run at 5v. Note that running large quantities of LEDs can result in a voltage drop across the supply, which may affect the reliability of the microcontroller. See the datasheet for your LED matrix display.

Setup

The PLT-1001 comes with firmware for the LDP-8008 board. For other boards you will need to update the firmware.

Updating the firmware is easy!

- Download the appropriate .hex file from the Embedded Adventures product page that matches your board.
- Download the screamer bootloader application from the Tutorials | Downloads section of the web site also – it's called Screamer.
- Connect a TTL serial port to your PC, like a USB to Serial PLT-1003. Connect TX from your TTL serial port to RX-I on the PLT-1001, and RX on your serial port to TX-O on the PLT-1001, and GND to both devices. The PLT-1001 will need to be powered with 5V if it can't get this from your serial port device.
- In the Screamer app, set the COM port setting to your COM port
- Set the Baud rate to 115,200
- Press Open and choose the .hex file appropriate to your LED matrix display.
- Press Download, and then reset the PLT-1001 by pressing the reset button.
- The new firmware will then be downloaded to the PLT-1001.

Depending on the timing of some serial ports, you may need to hold the reset button down, click the download button, then release your finger from the reset button. You should see the progress bar fill as the new firmware is downloaded.

Operation

The driver board is controlled by the TTL Serial port. Connect a device to TxO and RxI – keeping in mind that you will need to connect transmit on your device to RxI on the PLT-1001, and receive on your device to TxO on the PLT-1001.

Set your serial port to 115,200bps and then you're ready to issue some commands.

Commands

The driver is double buffered. There is a "draw" buffer, which is what these commands act on, and a "display" buffer which unsurprisingly is what is actually used to drive the display.

This means that you can set up exactly what you need on the screen using multiple commands and then transfer the entire completed picture to the actual display in one hit for very smooth updates.

The driver board comes with a collection of graphical and text commands. Each command starts with a text word and ends with <ENTER> (ASCII character 13 – which is \r if you are using a microcontroller to send commands).

Once you have sent all commands you need, send a **paint** command. Effectively the paint command copies the draw buffer to the display buffer.

The exceptions to this are the **scroll** command, which doesn't require **paint**, since it updates the buffer automatically as the text moves across the display, and the **test** and **title** commands.

Parameters to the commands can be separated by commas or spaces.

Most LED matrix displays, at least the ones from Embedded Adventures, have two pixel colours available - red and green, giving you black (colour 0), red (colour 1), green (colour 2), yellow/orange (colour 3).

clear

clear

Clear the display (all black pixels)

pixel

pixel colour x y

Set the pixel at position x, y to the specified colour. On LED matrix displays, x = 0 on the left most column, and y = 0 on the top row.

line

line colour x1 y1 x2 y2

Draws a line between x1, y1 and x2, y2 in the specified colour.

rect

rect colour x y width height

Draws a rectangle with left, bottom corner x,y that is width wide and height high.

circle

circle colour x_centre y_centre radius

Draws a circle centred at x_centre, y_centre in the specified radius and colour.

circlef

circlef colour x_centre y_centre radius

Draws a filled circle centred at x_centre, y_centre in the specified radius and colour.

circle2

circle2 colour x_centre y_centre radius

Draws a "squared" circle, centred at x_centre, y_centre – like an ordinary circle but with an extra pixel inserted at the top and sides so that the circle completely fills a square space.

font

font font_num

Sets the current font to font_num. The fonts are as follows:

1 – 5 pixel high (narrow)

2 – 5 pixel high (normal)

3 – 7 pixel high (normal)

4 – 8 pixel high (normal)

5 – 16 pixel high (double width)

6 – 16 pixel high (triple width)

7 – Bignum font – great for clocks and temperature displays. This font only has 0-9, :, C, F and the degree symbol (use the ‵ character for degree).

text

text colour x y "text"

Displays text in the current font where the bottom left corner of the text is positioned at x,y.

textv

textv colour x y "text"

Displays text in the current font where the bottom left corner of the text is positioned at x,y – and the text is printed vertically from that point.

This enables you to make different shaped displays – eg put three LDP-6432 displays together vertically, rotate the physical displays 90 degrees, and you have yourself a 96x64 pixel display which you can print text horizontally on using the textv command.

scroll

scroll colour x y width "text"

Displays text that scrolls through the rectangle made by x,y to width and the font height. The driver will send **<done>** when the scroll is complete. You can interrupt an existing scroll by simply issuing another **scroll** command. Scroll can handle strings of up to 170 characters.

scrollspeed

The default speed that scrolling moves at is a stately 150ms between changes. To alter this, pass the set the new speed using **scrollspeed**. This can even be done while scrolling.

paint

paint

Copies the draw buffer to the display buffer so previous commands are actually displayed.

title

The **title** command is a great way of testing your display. Using whatever resolution is available, it will draw a boarder around the outside, and show you the version of the firmware among other things. This enables you to see which display is which and see which way around the displays are orientated – the text will always be in the top-most or left-most display.

displays

Most firmware can control multiple displays. The LDP-6432 variety comes in two flavours – horizontal (ie the smaller sides next to each other) or vertical (the longer sides next to each other).

To set the display, pass it the parameter of the number of displays you have. It will default to 1 on boot, so if you have 2, simply issue:

```
displays 2
```

After setting the number of displays, you can then issue a **title** command to check which display is which.

enableactive

The Enable line on LED matrix displays can be a different polarity on different displays. All the displays you use in one daisy chain must be the same type.

For a display that is active low, issue the command:

```
enableactive 0
```

For a display that is active high, issue the command:

```
enableactive 1
```

It won't harm your display either way, but if you're getting nothing displayed at all, or it is quite feint, try the opposite polarity.

Test commands

test test_num

A series of test commands are available to confirm that everything is working, this is especially handy when working with a new board or when developing drivers for a new board.

Test	Result
0	Top left pixel
1	Top right pixel
2	Bottom right pixel
3	Bottom left pixel
4	Full screen, colour 1
5	Full screen, colour 2
6	Full screen, colour 3

7	Circle, full height, centred
8	Rectangle 1 pixel in from the edge, colour 1
9	Rectangle 2 pixel in from the edge, colour 1
10	Rectangle 3 pixel in from the edge, colour 1
11	Lines around the edge
12	Static text
13	8 pixel high bitmap
14	16 pixel high bitmap (if display is big enough)
15	32 pixel high bitmap (if display is big enough)

Tips and tricks

I'm sending commands but nothing is displayed!

Check the power connections. Be careful when hooking up the board since reversing the connections will damage LED matrix displays.

Make sure you send a paint command after you send commands. Try the **title** command. It's a great first step to see that everything is working.

Check that your panel has the same pin-out as the PLT-1001 – it's called "HUB08", a semi-standard for LED panels - if not, see the section on using the PLT-1001 with other panels.

The firmware will print out its version over the serial port when the PLT-1001 is reset, so check that the right firmware for the display is loaded.

And if your display is either blank or you can see pixels only faintly, then you probably need to try the **enableactive** command to set the ENable line correctly.

I'm sending commands and there's crazy stuff being displayed!

The firmware is designed to be small and fast – so it often avoids parameter checks.

Most likely if you're seeing strange things displayed, one of your parameters is out of spec.

For example, if you are using an LDP-8008, remember that the vertical pixels (y dimension) are numbered 0 through 7. Setting a pixel at x=0, y=8 will set a random area of memory rather than what you expect. Of course this is amplified when using commands that use large areas of the screen. Check your parameters to make sure that you are within the spec of the screen you are using. Set a pixel at x=0, y=7 instead.

Something else to be conscious of is that the commands that take x, y parameters specify the bottom left corner of the function, and the x=0, y=0 pixel is in the top left hand corner of the display. So to display a full screen red rectangle on an LDP-8008, you would issue the commands:

rect 1 0 7 80 8

paint

(where 1 = red, x=0, y=7 (bottom left corner), 80 pixels wide, 8 pixels high.

Red displays okay but green is very faint and orange/yellow is more red than orange/yellow.

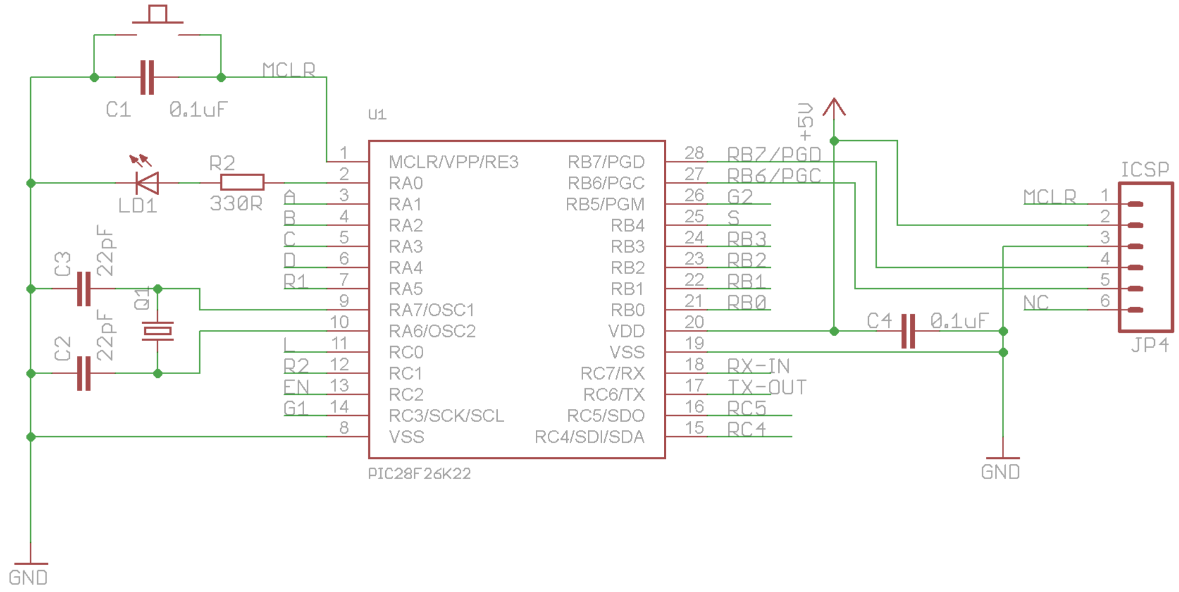
You probably haven't powered the displays with their 5V and GND connections and your display is leaching current from the serial connection.

Other panels

If your LED panel has a different pin-out to the standard set, it is a simple matter of connecting the correct pins to the LED Panel connection on the PLT-1001. In this case, you may find it easier to use a male connector on the top of the board so female jumpers can be used.

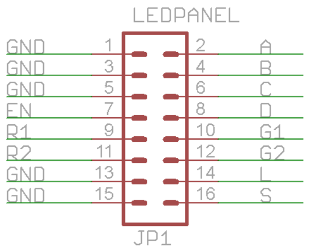
Schematic

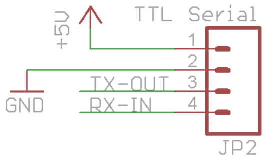
General circuit



The microcontroller circuitry is fairly standard. A reset button is provided, which pulls MCLR to ground. C1 prevents debounce noise. C2, C3 and crystal Q3 allow the PIC to run at 16mips. LED1 is available by pulling RA0 high; it is used by the 18f26K22_16Mhz_64Mhz_PLL version of Boostbloader to indicate the bootloader is alive and well. C4 is a power supply bypass for the PIC and the ICSP connection provides a way of reprogramming the PIC without a bootloader (or to flash a new bootloader).

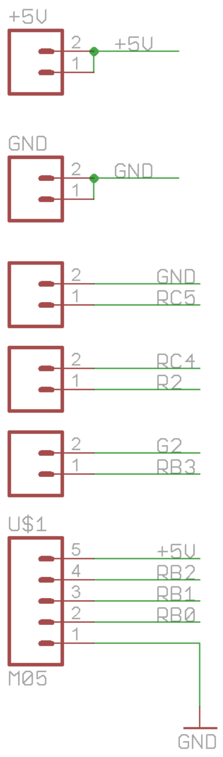
Connections





The LED panel connection follows the HUB08 standard connection for LED matrix display panels.

The TTL serial connection allows for control of the PLT-1001.

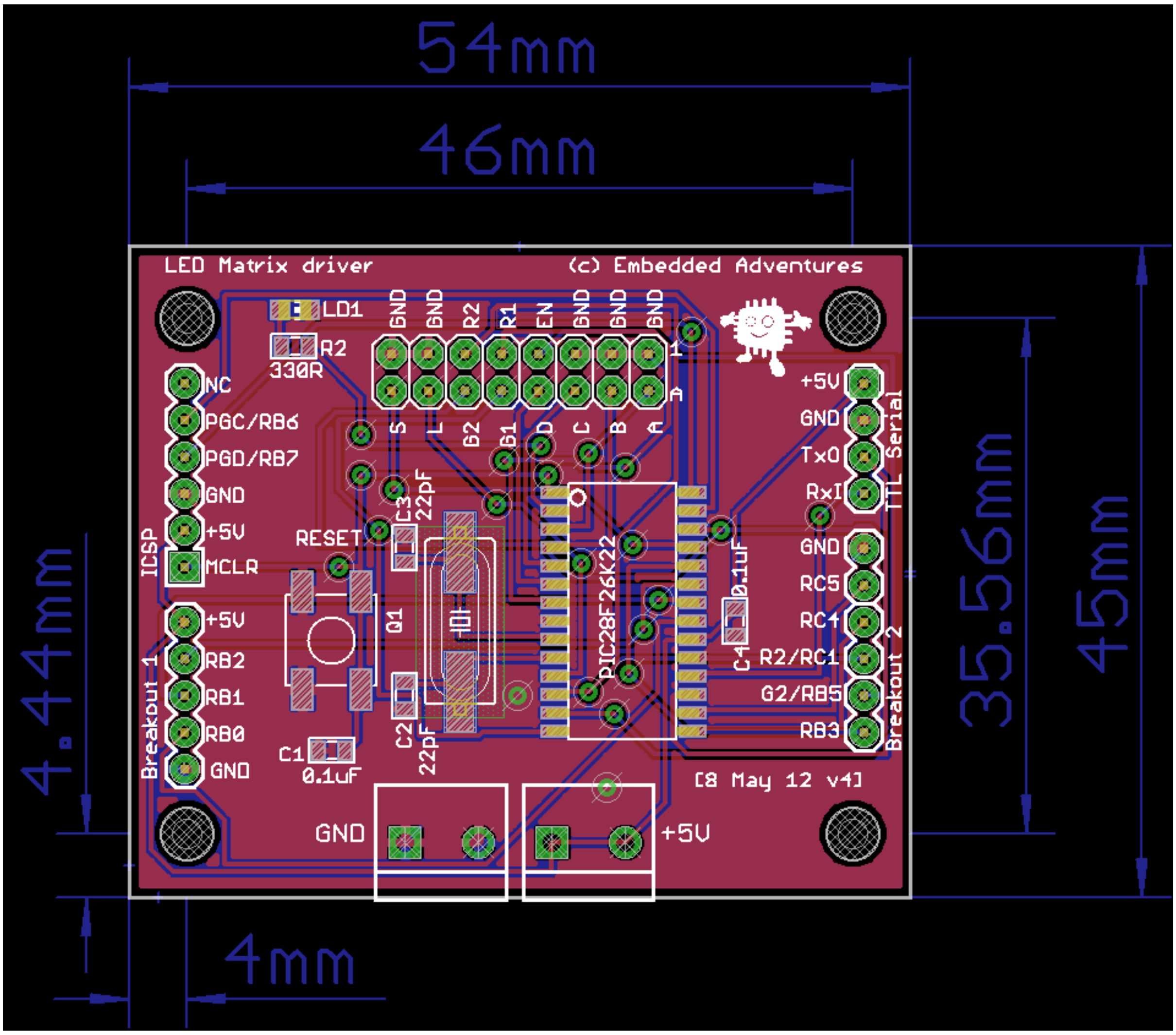


The other connections break out remaining pins on the microcontroller for future expansion.

A previous version of the PLT-1001 board contained a real time clock and temperature sensor, however to reduce the size of the board to allow the backpack configuration (and reduce the cost of the board!), these were removed from this version. However other modules can be connected to the PLT-1001 using these breakout connections.

In future updates to the firmware we'll enable other modules to be connected to the PLT-1001v4.

PCB



What next?

We really hope you enjoy using this display driver and come up with some fantastic projects. Please tell us about it – email us at myproject@embeddedadventures.com. And don't forget that we'll be making updates to the firmware so that you can interface with real time clock modules and more!

Versions

Version	Date	Comments
Version 1.0	9 June 2010	Initial version for board v1.2
Version 2.0	16 June 2012	Updated version for the completely revamped v4
Version 3.0	7 April 2014	Updated version for new 2.10 firmware