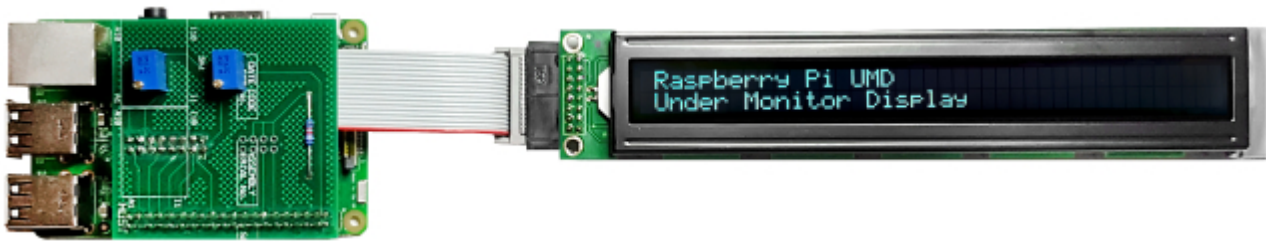


Under Monitor Displays



The Experience Center at Southampton has 8 x HD Monitors for PQ comparisons. To allow customers to identify and track what content is on each screen we need a solution that allows us to display clearly what is on each monitor. While we could use OSD, these can deteriorate on low bit rate tests. So we have decided to use some under monitor displays. These are 2 Line LCD screens that sit under the monitors and display messages identifying content such as 'MPEG2 SD' or 'Service 1' etc.

While these can be purchased, they can run in to sever thousand £ for each unit. We only require a simple low cost solution, for this we are using LCD displays that we already own (2 x 40 Char displays from RX units that we are scrapping). These displays will be driven from Raspberry Pis, then input for the diplay messages will be via a web interface to the primary Raspberry Pi.

As the LCD displays are two line, we will use four of these displays, one for each pair of monitors, the upper line for the upper monitor, lower line for the lower monitors. The first Raspberry Pi will run monitors 1 and 2, this Raspberry Pi will also be the master where the web interface to control the displays is running.



The other three Raspberry Pi's will run the remaining monitors, these will talk to the primary Raspberry Pi to see what messaging they need to display.

- [Setting up the Master UMD](#)
- [Setting up the Slave UMDs](#)

Slave Raspberry Pi(s)

The slave Raspberry Pi(s) download the latest 'my_data.txt' from the Master Raspberry Pi (once a second) so that each slave knows what to display. The 'my_data.txt' file is downloaded using a wget command.

The 'my_data.txt' file has 8 lines, each line is for a particular display, so for example, line 1 & 2 are for LCD1 and reference monitors 1 and 2. Lines 5 & 6 are for LCD and reference monitors 5 & 6.

```
Install Raspbian (noobs or Jessie, doesn't matter)
Update pi - sudo apt-get update && sudo apt-get upgrade -y
```

```
Install xrdp (this so we can access the R-pi using RDP if required)
Create Python folder (/home/pi/Python)
Copy files to python folder: (might have to change file/folder permissions to www-data)
    getData (downloads via wget the_data.txt file)
    UMDisplay0x.py (the code that writes to the display)
    my_data.txt (contains the 8 lines for the displays, this will be updated by getData)
    launchUMD.sh (autorun script)
Add line to crontab for autoboot
Add Python/logs folder
```

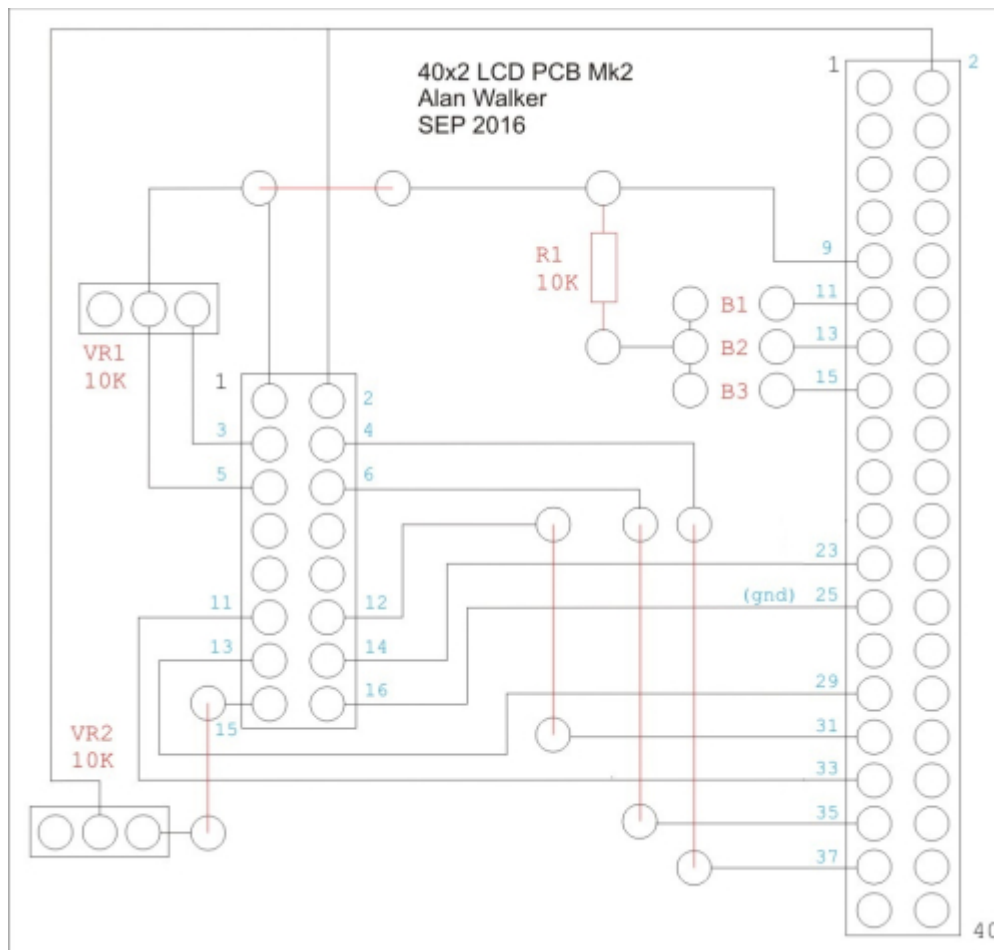
Wiring

The 2 Line is wired to the Raspberry Pi using the following pin out.

LCD	R-Pi	Desc	Notes
16W	40W		
01	06	GND	
02	02	+5V	
03	14	Contrast	Connect to GND via 10k Pot
04	37	RS	Register Select
05	09	GND	
06	35	En	Enable
07	nc	d0	
08	nc	d1	
09	nc	d2	
10	nc	d3	
11	33	d4	
12	31	d5	
13	29	d6	
14	23	d7	
15	04	Backlight	Connect to +5V via 10K Pot
16	25	GND	

NOTE The Pin out shows Pin 14 of the 40Way Pi header connected, but the diagram does not, this is because Pin 14 is GND, so any ground point can be used.

For this project I am getting some circuit boards made, with a layout similar to this:



This circuit board has a 40 pin header to fit to the Raspberry Pi via a ribbon cable, a 16 pin header to fit to the LCD via a ribbon cable. There are two Variable Resistors to adjust the LCD intensity and back-light. There is a 6 way header to allow the addition of three buttons, these will have the following function:

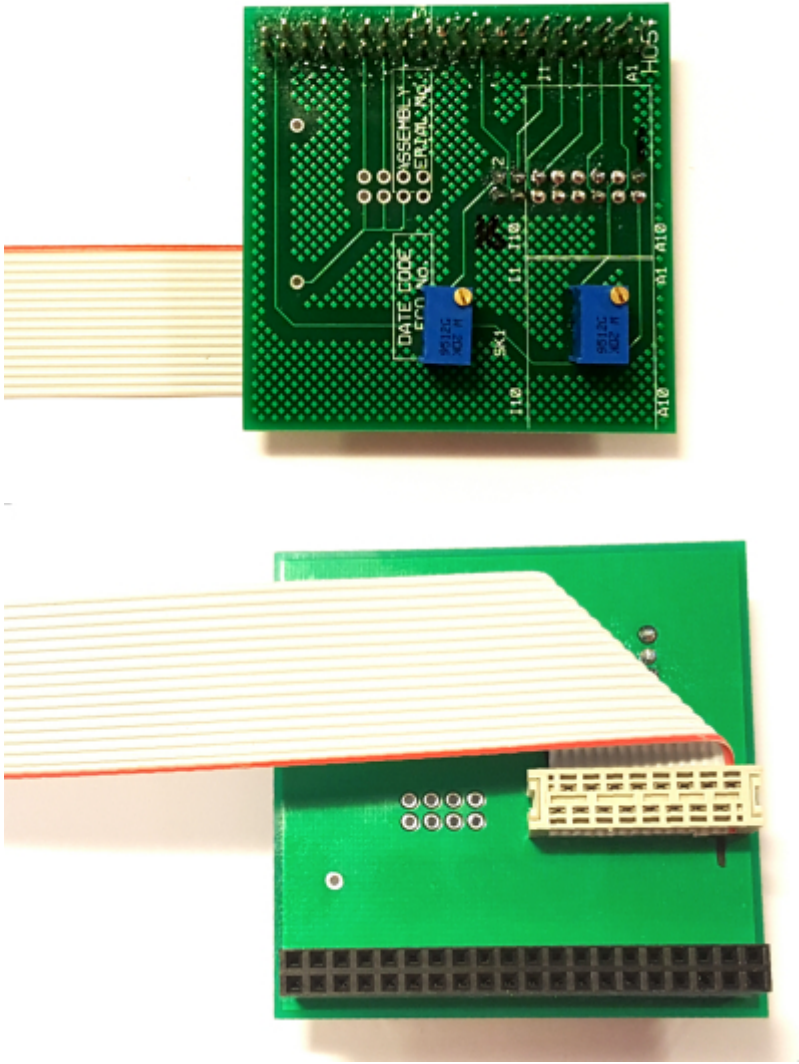
1. Display IP Address
2. Blank Line 1
3. Blank Line 2

As the IP addresses on the Raspberry Pi will be DHCP (so we can access them from the corporate network) we might need to check the IP addresses from time to time, as these units are headless, a button to display the IP Address on the LCD display will be useful.

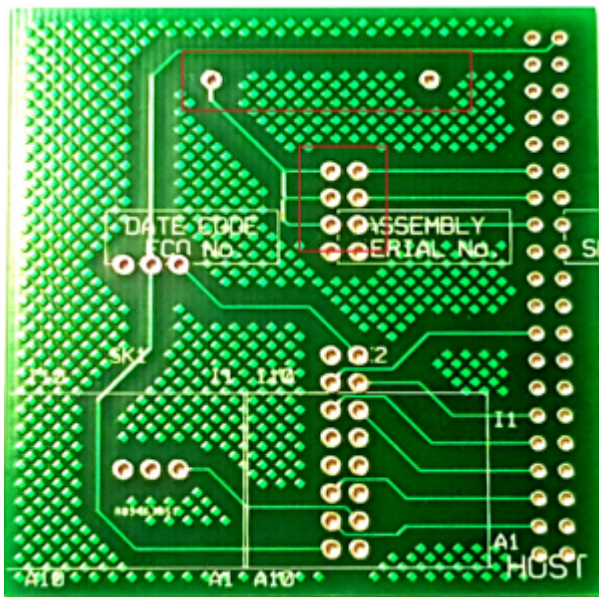
Buttons 2 and 3 are to blank the lines on the LCD, this is so that when doing a 'show and tell' with customers, we can keep them guessing before revealing the content types on the montiors.

LCD PCB

Here is the PCB, I had a slight issue with the tracks, and to solve this I had to mount the 16 pin header on the bottom of the PCB (the left and right hand pins of the 16W connector would have to be swapped if you wanted to mount the header on the top of the PCB)



There are an extra 6 pins for additional GPIO connectivity (they will all connect to ground via a 10KR resistor if fitted. While there are 8 pins, only the upper 6 are fitted.



The initial functionality for these will be for 3 buttons:

1. Hide Line 1
2. Hide Line 2

3. Show IP

Finished LCD PCB running on a Pi Zero.



From:

<https://wiki.alanwalker.uk/> - WalkerWiki - wiki.alanwalker.uk

Permanent link:

https://wiki.alanwalker.uk/doku.php?id=under_monitor_displays&rev=1482331783

Last update: 2023/03/09 22:35

