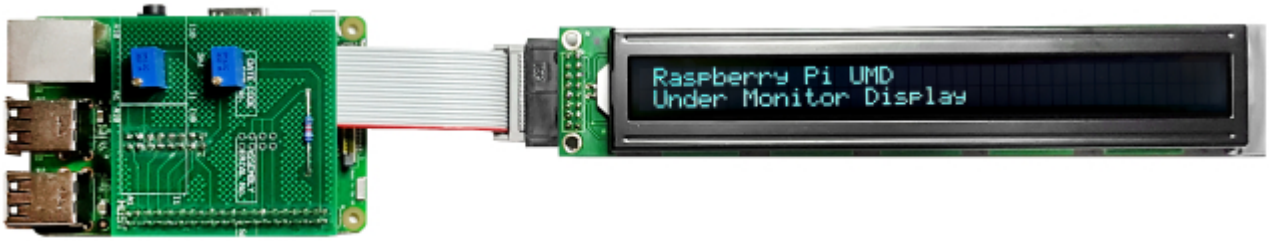
Setting up the Slave UMDs



Slave Raspberry Pi(s)

The slave Raspberry Pi(s) download the latest 'my_data.txt' from the Master Raspberry Pi (once a second) using a 'wget' command, this means that each slave knows what to display. The my_data.txt file is downloaded every second, so changes to the file via the web interface propagate through the UMDs very quickly.

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.

The setup of the slave UMDs is very simple compared to the Master. Setup Raspbian, copy some files and away you go.

Set Static IP Address

To set a static IP address, we need to login to the RPi via SSH and change the following file:

The default login is username: pi and password: raspberry (if this is an existing unit that is misbehaving, the login will be pi and Ericss0n)

```
/etc/dhcpcd.conf
```

Enter the following

```
sudo nano /etc/dhcpdc.conf
Add these lines to the end of the file (default is dhcp) using  your own IP Address details
eth0
    static ip_address=192.168.1.11/24
    static routers=192.168.1.1
    static domain_name_servers=192.168.1.4
```

Use Ctrl-X to exit and Y to save, now reboot the pi and connect to the new address

---

Raspi Config

From the terminal, run:

```
sudo raspi-config
```

You will see the following menu, there are several items we want to change here.

```
                    ┤ Raspberry Pi Software Configuration Tool (raspi-config)
┌──────────────────────────────┐
│
```

```
│
│     1 Expand Filesystem            Ensures that all of the SD card storage is available to the
│
│     2 Change User Password         Change password for the default user pi)
│
│     3 Boot Options                 Choose whether to boot into a desktop environment or the command
line │
│     4 Wait for Network at Boot     Choose whether to wait for network connection during boot
│
│     5 Internationalisation Options Set up language and regional settings to match your location
│
│     6 Enable Camera                Enable this Pi to work with the Raspberry Pi Camera
│
│     7 Add to Rastrack              Add this Pi to the online Raspberry Pi Map (Rastrack)
│
│     8 Overclock                    Configure overclocking for your Pi
│
│     9 Advanced Options             Configure advanced settings
│
│     0 About raspi-config           Information about this configuration tool
│
│
│
│
│                            <Select>                                <Finish>
│
│
│
└──────────────────────────────────────────────────────────────────────────────────────────────────
──┘
```

Select Option 2 (Change User Password) - Change the password to Ericss0n
Select Option 3 (Boot Options) - Change to B2 Console Autologin Text console, automatically logged in as 'pi' user
Select Option 9 (Advanced) then A2 (HostNane) - Change the Hostname to UMD00n (where n is the display number)
Select Finish when asked to reboot select Yes

Update RPI

Even if you downloaded the latest version of Raspbian, chances are there are some updates. To update the RPi use the following command line:

```
sudo apt-get update && sudo apt-get upgrade -y
```

This will most likely take a few minutes on a new install.

Create Folder, Copy Files, Set Permissions

To hold the UMD LCD files, create a folder in your home location called Python and give the following rights:

```
mkdir /home/pi/Python
sudo chmown www-data /home/pi/Python
sudo chmod 777 /home/pi/Python
```

We need to copy the files to the /home/pi/Python location, for the Slave there are four files:

- launchUMD.sh - A script to auto start the UMD at boot
- getData - This is the script that gets the updated my_data.txt file
- my_data.txt - This holds the LCD data text, and downloaded from the Master via wget commands

- UMDisplay02.py - The Python code that runs the LCD

These files need to be copied to the /home/pi/Python location

sudo chmod 777 /home/pi/Python/*

Edit getData File

The getData file contains the wget command that pulls the latest version of my_data.txt from the Master Pi. The my_data.txt contains the 8 lines of text for the four LCD displays (2 lines for each LCD).

Example my_data.txt

```
UPPER MONITOR (1) - H264 SD
LOWER MONITOR (2) - H264 HD
UPPER MONITOR (3) - HEVC SD
LOWER MONITOR (4) - HEVC HD
UPPER MONITOR (5) - UHD TV
LOWER MONITOR (6) - HDR UHD TV
UPPER MONITOR (7) - Sky Cinema HD
LOWER MONITOR (8) - Sky Sports UHD
```

Example getData

```
sudo wget -q http://192.168.1.11/my_data.txt >null -O my_data.txt
```

In the getData file we need to replace the IP Address with the IP Address of our Master UMD Pi.

To change the IP Address use:

```
sudo nano /home/pi/Python/getData
```

You could use HostNames, but I have found hostnames on Pi's pretty unreliable.

Edit launchUMD File

The launchUMD file contains a link to the UMDisplay0x.py file

```
#!/bin/sh
# launchUMD.sh
# navigate to home directory, then to this directory then launch script

cd /
cd home/pi/Python
sudo python UMDisplay01.py &
cd /
```

This file needs to be edited so that the line sudo python UMDisplay1.py & points to the correct UMD Device (so if this is your third UMD Device it needs to contain UMDisplay03

```
sudo nano launchUMD.sh
```

Edit UMDsiplay0x File

The UMDisplay0x file contains the code to write to the LCD. We do have to edit this file to tell it which of the two LCD lines to display from the my_data.txt file

```
# write line 1 and 2 to the LCD
lcd_string(" " + line1,LCD_LINE_1)
lcd_string(" " + line2,LCD_LINE_2)
```

Line1 refers to the line in the my_data.txt file. So if this was UMD3, then you would want lines 5 and 6. so this would be:

```
  lcd_string(" " + line5,LCD_LINE_1)
  lcd_string(" " + line6,LCD_LINE_2)
```

LCD_LINE_1 refers to the Line on the LCD (Line 1 is the top, line 2 is the bottom)

Edit the UMDisplay0x.py file by using:

```
  sudo nano /home/pi/Python/UMDisplay0x.py
```

Test LCD Code

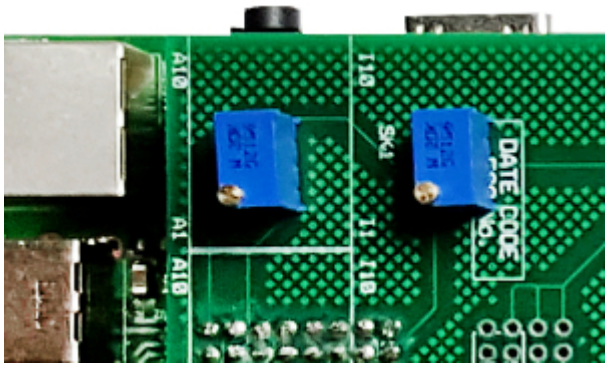
Assuming that you have connected the LCD to the RPi (via the ribbon cable and driver board) then you can test the code that runs the LCD, on the slave this file is the:

```
  UMDisplay0x.py (where x is 2, 3, 4 4 etc.)
```

This is a python script, you can run this script by entering at the command line:

```
  ./UMDisplay02.py
```

If you are lucky, you will see some text on the LCD, chances are however you will not. This is due to the Potentiometer settings on the driver board, you will need to change these to set the Backlight and LCD intensity.



Adjusting these two potentiometers will allow you to balance the text and backlight intensity to get the brightness level you require.

Making it Automatic

The last part is to automate the process of starting the UMD code. For this we need to add the launcher file to a system file on the RPi.

Edit the file /etc/rc.local to automate this process.

```
  sudo nano /etc/rc.local
  add this line at the end:
```

```
sudo ./home/pi/Python/launchUMD.sh
```

To test this, reboot and the LCD should start up automatically (the LCD won't clear on boot, but wait for the “Waiting for 5 Seconds for Network Start” message).

UMDsiplay0x.py File Differences

There are a couple of differences between the Python scripts on the Master and the slaves, here is an overview of those differences.

#import

On the #import section there is one difference

```
Master                                  Slave

#import                                 #import
import RPi.GPIO as GPIO                 import RPi.GPIO as GPIO
import time                             import time
import socket                           import socket
import fcntl                            import fcntl
import struct                           import struct
                                        import os
```

On the slave there is an extra library that is imported (the os library). The os library allows the running of command lines from within Python.

Program Start

At the Program Start section of the Python code there is an extra line.

```
## Get text file from master Pi UMD001 (runs the file getData in the Python folder)
os.system("/home/pi/Python/getData")
```

The os.system(“/home/pi/Python/getData”) calls the getData script. This is the script that pulls the latest my_data.txt file from the UMD Master. This command is the one that needs the extra #import library.

From:
https://wiki.alanwalker.uk/ - **WalkerWiki - wiki.alanwalker.uk**

Permanent link:
https://wiki.alanwalker.uk/doku.php?id=setting_up_the_slave_umds&rev=1482353971

Last update: **2023/03/09 22:35**