

## Seattle Raspberry Jam

A Raspberry Pi Meetup for Beginners to Experts

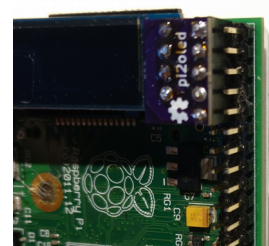


# Hardware Project #1: OLED Display

## Assembly

First, make sure the Raspberry Pi is not powered. Plug the circuit board into a Raspberry Pi on pins 1, 3, 5, 7 and 9 (see the picture to the right).

**WARNING! Incorrectly installing the display could damage both the display and the Raspberry Pi. Please have one of the organisers check that the display is plugged in correctly before applying power.**



## Software Setup

Before writing data to the display, several software changes need to be made on the Raspberry Pi. First, a device tree overlay and parameter need to be setup in `/boot/config.txt`. The overlay needed is the `ssd1306.dtbo` file (this file can be downloaded from <https://sites.google.com/site/mincepi/pi2oled/files>). This overlay sets up some parameters for the display, sets up I2C (used to communicate with the display) and creates a frame buffer for the display data. Use the following command to copy the overlay file to the `/boot/overlays` directory so that it can be used in `config.txt`.

**Tip:** The following commands require root permissions to run. `Sudo` runs a command with root permissions. Instead of typing `sudo` before every command, you can type `sudo su` to enter root mode before entering the commands that require root permissions. Type `exit` to return to regular permissions.

```
sudo cp ssd1306.dtbo /boot/overlays
```

Next, open `/boot/config.txt` in the `nano` text editor with the following command:

```
sudo nano /boot/config.txt
```

Add the following lines to the bottom of the file:

```
dtoverlay=ssd1306,inverted,sequential
dtparam=i2c_arm=on,i2c_baudrate=1000000
```

*Continued overleaf*

The first line enables the overlay we copied into `/boot/config.txt`, while the second line sets up I2C. To exit out of the editor, type Ctrl-X, then type y and finally press Enter. The software setup is now complete. Reboot the Raspberry Pi for the changes to take effect.

## Display Control

We need to run some commands before we can output text to the OLED screen. The following command sets the font we want to use for the display:

```
setfont Lat7-Terminus16
```

Now we are going to map the display's frame buffer to the terminal `tty8`. There are normally seven consoles numbered from `tty1` through `tty7` on a Raspberry Pi. The first six are used as command-line interfaces (CLIs) and the seventh is used for a graphical user interface (GUI). The following command maps frame buffer number 1 to console number 8 (`tty8`).

```
con2fbmap 8 1
```

Next, enter the following command to activate the `tty8` console.

```
chvt 8
```

Now we can display text on the OLED screen. The following command will print "Testing 1 2 3..." on the screen. The `echo` command prints text to a console. The `-n` flag removes the trailing newline character to prevent the display from scrolling and the `-e` flag executes escape sequences (such as `\ec`, which resets the screen's text, or `\n`, which prints a newline character). The `>` symbol pipes the output of the command into a file, which in this case is console `tty8`.

**Tip:** For more information on a command, type `man COMMAND`. This opens a page of information about the command and about how to use the command. Use the up and down arrow keys to move through the file and type q to exit. Most commands have a man pages.

```
echo -n -e "\ecTesting 1 2 3...\n" > /dev/tty8
```

The words "Testing 1 2 3..." should appear on the OLED screen. If they do not, notify one of the organisers. For a more interesting example, we can print the Raspberry Pi's IP address. The following command has the same configuration as the command above, but now with `'hostname -I'` instead of the "Testing 1 2 3..." text. The command `hostname -I` prints the IP address of the Raspberry Pi. Note: the symbols before and after `hostname -I` in the `echo` command are not single quotes but rather back quotes (the key for the back quote is found in the upper-left corner of the keyboard). The back quotes make `echo` execute the `hostname -I` command rather than just printing "hostname -I" to the screen.

```
echo -n -e "\ec`hostname -I`\n" > /dev/tty8
```

The IP address of the Raspberry Pi should now appear on the screen. Congratulations! You have completed the OLED screen tutorial.

**Bonus:** See if you can figure out how to get the display to show the Pi's processor speed and temperature.

**Hint:** the processor speed is in the file `/sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq` and the command `cat` prints the contents of a file. The command `vccencmd measure_temp` prints the processor temperature.