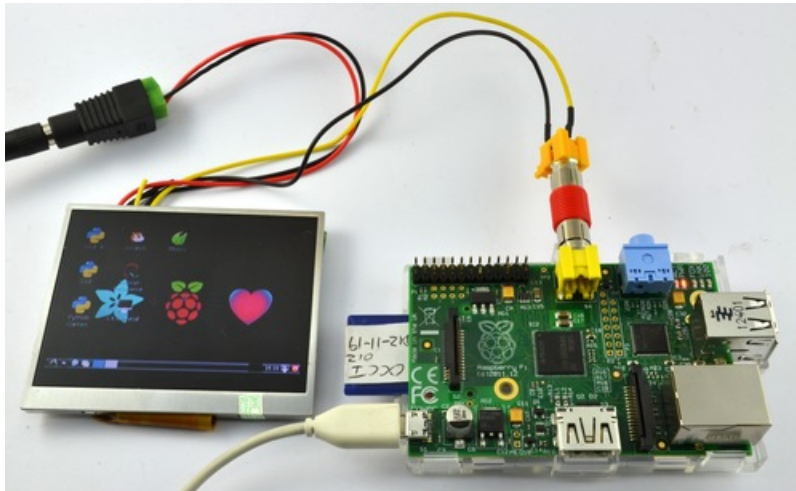


## Using a Mini PAL/NTSC Display with a Raspberry Pi

Created by Simon Monk



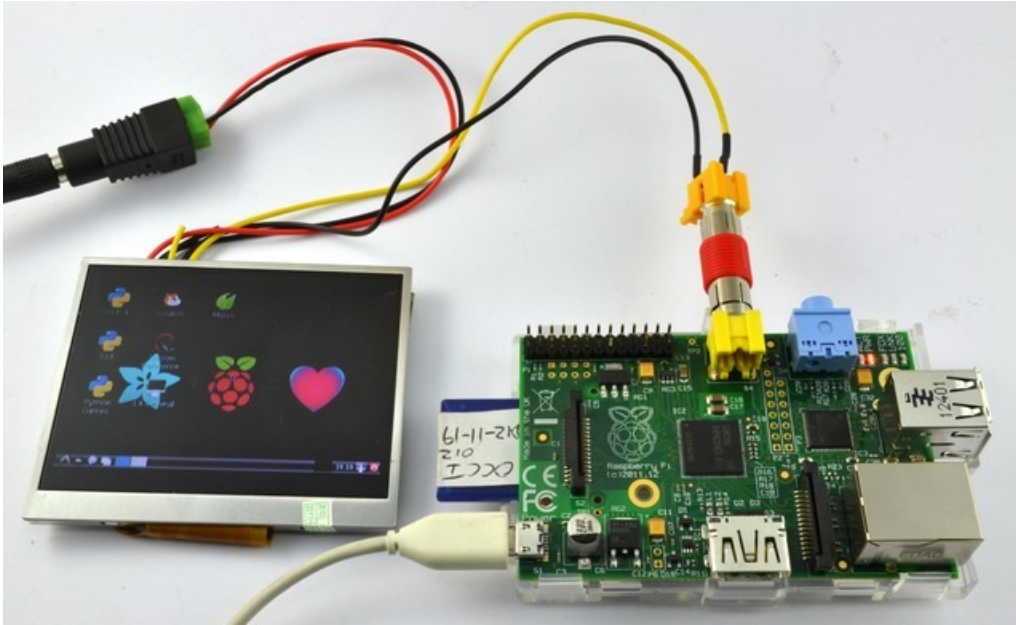
Last updated on 2018-08-22 03:33:45 PM UTC

## Guide Contents

Guide Contents	2
Overview	3
Parts	4
Connecting the Display	5
Configure and Test	6

## Overview

An often neglected feature of the Raspberry Pi is that it actually has two types of video output. The high quality, high resolution HDMI output and the yellow RCA jack that outputs composite video at a much lower quality. Miniature displays are readily available for this composite format, and are often used for things like CCTV security and pocket TVs.



In this tutorial, you will learn how to attach a miniature display to your Raspberry Pi and to adjust the Pi's settings to reduce the resolution sufficiently to read the screen.

## Parts

---

To use a mini PAL / NTSC display, you will need the following parts.

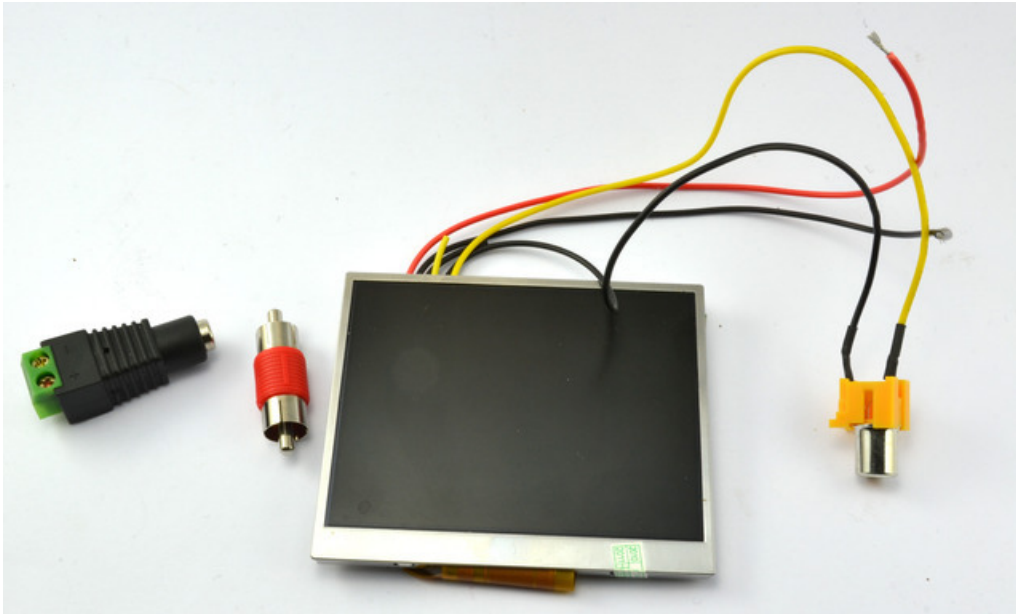
- A Raspberry Pi
- A PAL / NTSC Mini display, such as Adafruit products 910, 911, 913 or 947
- RCA Coupler Male to Male
- Female DC Power adaptor (unless your display has a DC barrel socket)
- 12V 1A power supply (unless included with the display)
- Raspberry Pi Console lead or SSH access to your Pi

## Connecting the Display

Even the smallest of these displays will use enough power to warrant its own power adapter, especially as they generally do not use the 5V that the raspberry Pi uses. Often, they are designed to operate at 12V but will be happy at a range of voltages.

Check when you buy the display, what voltage it operates from. For some displays, a power adapter will be included.

As an example, in this tutorial, we will use the [Adafruit 3.5 inch display \(http://adafru.it/913\)](http://adafru.it/913).



This display actually comes with two inputs (AV1 and AV2), I simply cut one of them off, to neaten up the leads. You do not have to do this, and you can use either lead when connecting to the Pi. Most only have one AV connection.

The unit comes in two parts, a control PCB and the display itself. They are quite delicate, especially the plastic ribbon that connects the two together, so handle with care.

I glued the control board to the back of the display, to make a single unit. If you decide to do this, just remember that if you need to separate them again, there is a good chance of breaking it.

The unit has a black and a red lead to supply the power. This can be between 6 and 12V. To make it possible to connect the bare leads to a power supply with a standard 2.1mm DC power plug, I used a [Female DC power adapter \(http://adafru.it/368\)](http://adafru.it/368). You can just pop the leads into the screw terminals, tighten them up and then use a regular DC power supply. Be very careful that the red lead goes to the terminal marked with a +.

The display I used has RCA sockets, as they are intended to be connected to a video source such as a camera or in this case the Raspberry Pi via a cable. As you are probably going to have the display right next to the Pi, you do not really need a cable, and an [RCA Male to Male adapter \(http://adafru.it/951\)](http://adafru.it/951) will allow you to connect it to the yellow socket on the Pi.

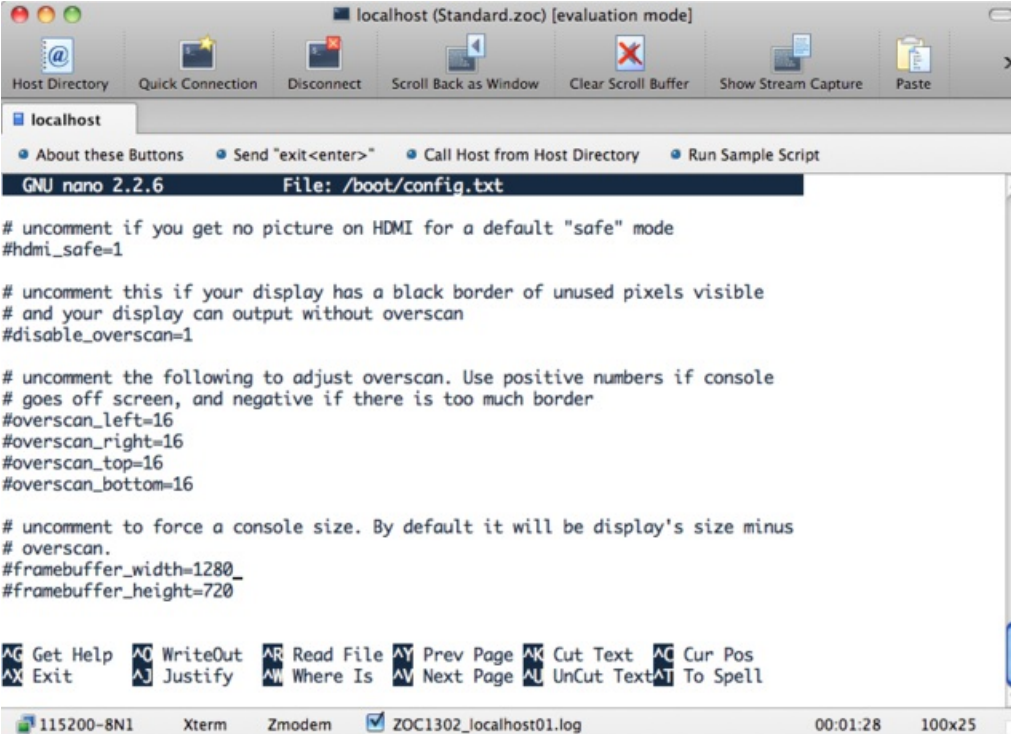
## Configure and Test

Turn on the power to the display and then power up your Pi. You should see the Pi start to boot up on the display. You will notice that the writing is pretty tiny and a little fuzzy.

We need to change the screen resolution so that we can see what is going on.

The native resolution of the display I used, is just 320 x 240 pixels.

To adjust the resolution, you need to edit the file `/boot/config.txt`



```
localhost (Standard.zoc) [evaluation mode]
GNU nano 2.2.6 File: /boot/config.txt
# uncomment if you get no picture on HDMI for a default "safe" mode
#hdmi_safe=1

# uncomment this if your display has a black border of unused pixels visible
# and your display can output without overscan
#disable_overscan=1

# uncomment the following to adjust overscan. Use positive numbers if console
# goes off screen, and negative if there is too much border
#overscan_left=16
#overscan_right=16
#overscan_top=16
#overscan_bottom=16

# uncomment to force a console size. By default it will be display's size minus
# overscan.
#framebuffer_width=1280_
#framebuffer_height=720_

AG Get Help  AJ WriteOut  AR Read File  AY Prev Page  AK Cut Text   AC Cur Pos
AX Exit      AJ Justify   AW Where Is  AV Next Page  AU UnCut Text AT To Spell

115200-8N1  Xterm      Zmodem      ZOC1302_localhost01.log  00:01:28  100x25
```

You can edit `config.txt` on the Pi itself by running `sudo nano /boot/config.txt` or you can safely shut down the Pi, remove the SD card and put it a computer, the `config.txt` file is on the 60MB partition that should appear when you insert the card

If you are trying to edit it directly on the Pi, this may present a bit of a problem, if the whole reason for changing the screen resolution is to be able to read the text. You can either attempt to edit the file using the mini display, or [connect remotely to your Raspberry Pi from another computer using SSH \(https://adafru.it/aWc\)](https://adafru.it/aWc) or, as I did here, [using a console cable \(https://adafru.it/aUA\)](https://adafru.it/aUA).

To edit the file, issue the command

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

Find the lines that say:

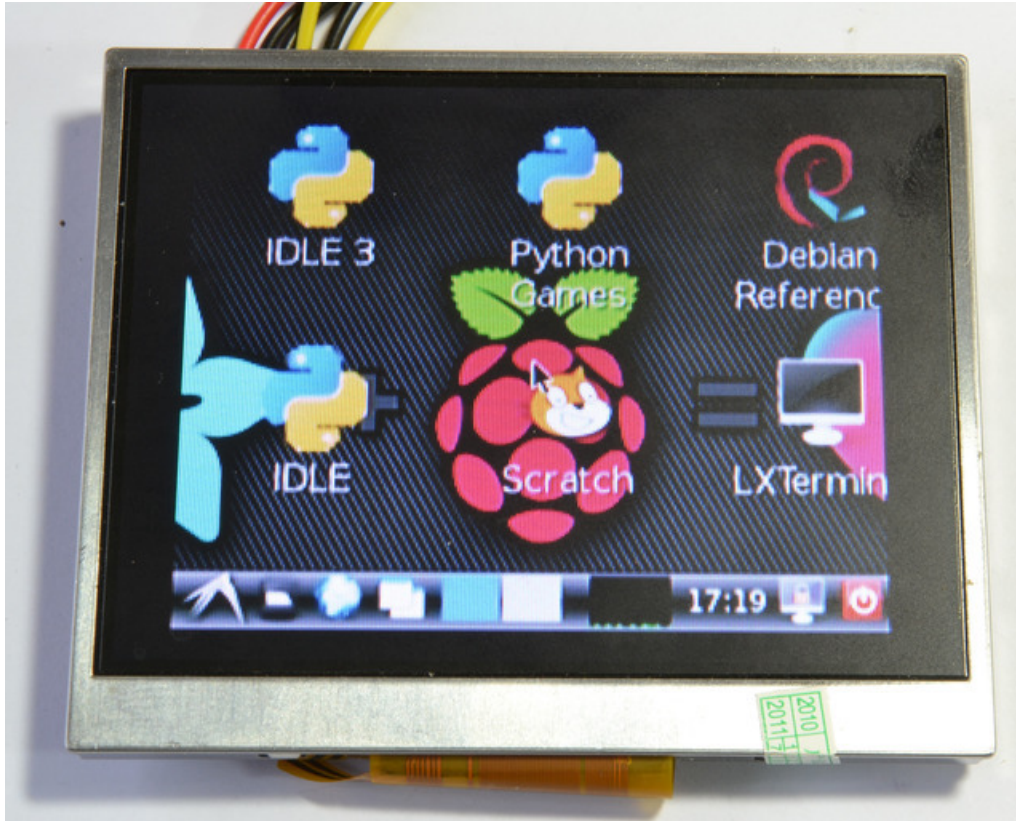
```
#framebuffer_width=1280
#framebuffer_height=720
```

You need to remove the # from the start of each of the two lines, to make the setting active and then change the two values, so that the two lines are like this:

```
framebuffer_width=320  
framebuffer_height=240
```

Save and exit the editor by pressing CTRL-x then Y.

Restart the Pi and you should find that all the text is bigger and clearer.



Now, you will have noticed that there is quite a wide border around the visible area of the screen. Since we have so little screen to play with, we really need to expand this as much as possible.

Fortunately, there is another set of settings that we can change to reduce this as much as possible.

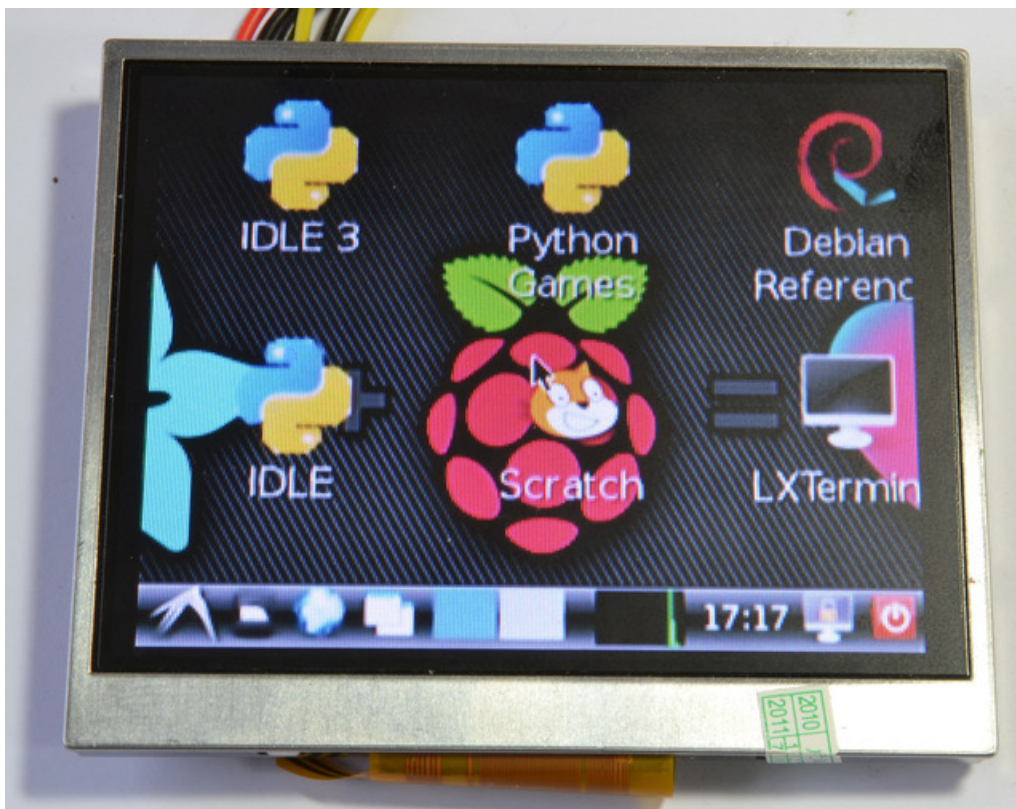
Edit /boot/config.txt again and this time, remove the # from the four lines that start with #overscan.

Change the numbers to be negative (makes the screen wider and taller). I found changing them to the figures shown below, used most of the screen vertically, however I could not get the screen any wider after about -25.

```
localhost (Standard.zoc) [evaluation mode]
Host Directory Quick Connection Disconnect Scroll Back as Window Clear Scroll Buffer Show Stream Capture Paste
localhost
About these Buttons Send "exit<enter>" Call Host from Host Directory Run Sample Script
GNU nano 2.2.6 File: /boot/config.txt
# uncomment if you get no picture on HDMI for a default "safe" mode
#hdmi_safe=1
# uncomment this if your display has a black border of unused pixels visible
# and your display can output without overscan
#disable_overscan=1
# uncomment the following to adjust overscan. Use positive numbers if console
# goes off screen, and negative if there is too much border
overscan_left=-26
overscan_right=-26
overscan_top=-16
overscan_bottom=-16
# uncomment to force a console size. By default it will be display's size minus
# overscan.
framebuffer_width=320
framebuffer_height=240
[Wrote 41 lines]
pi@raspberrypi:~$ _
115200-8N1 Xterm Zmodem ZOC1302_localhost01.log 00:13:00 100x25
```

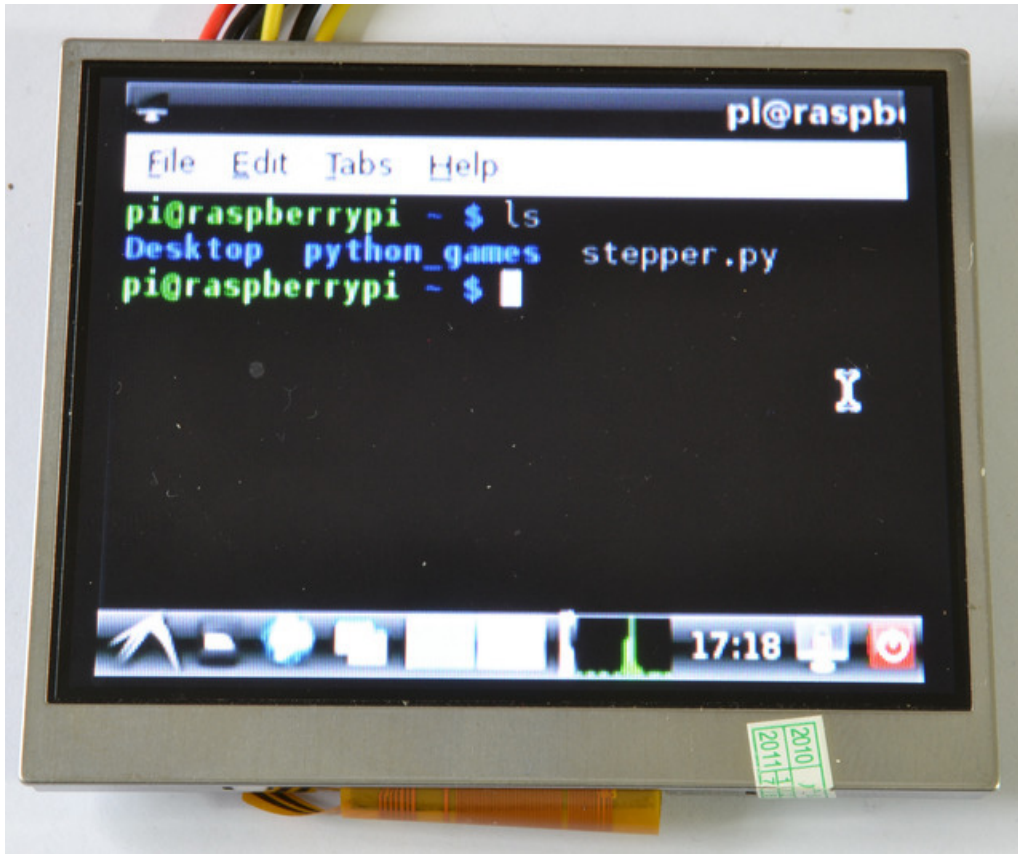
You will find that a bit of trial and error will be required to get the right settings for your display.

With the overscan adjusted, the resulting screen is now looking pretty good.



Starting a Terminal session, the text is now quite easy to read.





These changes to the settings will also affect (for the worse) the display if you connect up using the HDMI adaptor. To reverse the changes, just add a # character in front of each of the 6 lines that you changed earlier.