



FocusLCDs.com
LCDs MADE SIMPLE®

Ph. 480-503-4295 | NOPP@FocusLCD.com

TFT | CHARACTER | UWVD | FSC | SEGMENT | CUSTOM | REPLACEMENT

Application Note FAN4202

Configuring Screen Resolution

This application note is part 1 of our 2 part TFT LCD screen series. Part 1 discusses configuring the screen resolution of a TFT with a Raspberry Pi. Part 2 discusses calibrating the touch screen of a TFT. [Click here to read part 2.](#)

Configuring Screen Resolution with Raspberry Pi

This application note discusses how to setup a TFT touch screen display with Raspberry Pi. The display chosen for this project is a 7-inch LCD with a capacitive touch feature. The display has 1024x600 pixels of resolution and when calibrated for the Raspberry Pi can display 16.7M colors. With a few changes to the Raspberry Pi configuration files the dimensions of this display can be matched for perfect resolution.

The display in this project is easy to interface with the Raspberry Pi because it has an LVDS to HDMI backpack adapter connected. This makes it simple to connect an HDMI cable between the two devices and begin your project. In a few steps any display can be configured for optimal resolution. Additionally, this display has a capacitive touch screen which can be calibrated through Raspberry Pi for accuracy. These features combined with the large size of the display make it a great choice for a variety of Raspberry Pi projects.

The display in this project is [E70RA-HW520-C](#). Here are the details of the display:

Features

- 7", 165.0x100.0mm
- 1024(RGB)x600 pixels
- Controller: [EK73215/EK79001](#)
- 6/8-bit LVDS Interface
- HDMI connection
- Capacitive Touch Screen
- I2C CTP Interface
- CTP Driver: [GT9271](#)

For more information about this display, check out the [datasheet](#).

What You'll Need

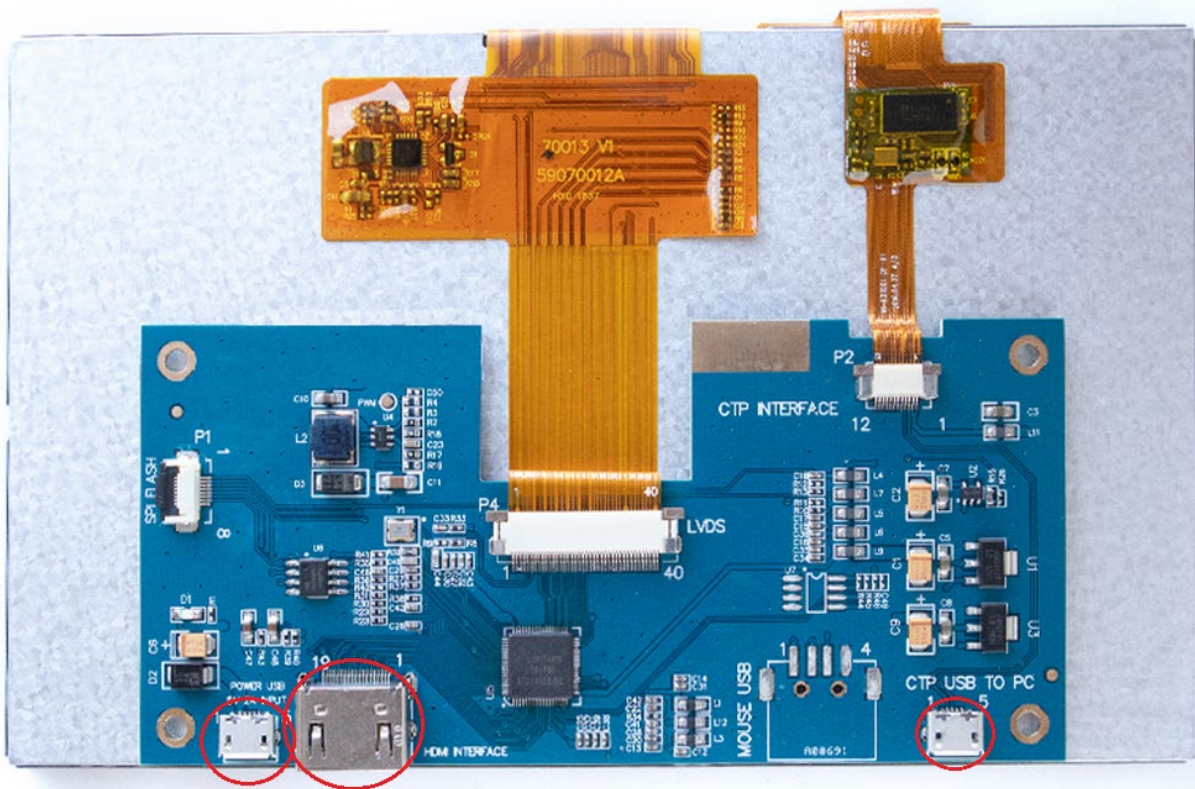
A list of the materials used in this project need are detailed below.

QTY	Description	Note
1	Raspberry Pi 3 with USB Power adapter	Raspberry Pi
1	7-inch TFT Display	Focus LCDs
1	HDMI cord	
2	Micro-USB cable	
1	Wireless Mouse and Keyboard	
1	Ethernet cable	
1	32MB microSD card with NOOBS installed	NOOBS

Connecting to the Raspberry Pi

Connecting the TFT to the Raspberry Pi is simple and only requires a few steps. The display has a HDMI connection backpack interfaced with the device so there is no need for soldering or individual pin connections. If you have used Raspberry Pi before, you will notice that the connection to the TFT is standard and straightforward.

Before powering the Raspberry Pi you should make all the necessary connections to the screen. Plug in the HDMI. Connect the CTP port with a micro-USB cable. Power the screen with micro-USB cable connected to a USB port on the Raspberry Pi or a power adapter that provides 5V. You only need three connections to the HDMI backpack module on the TFT.



You will need a mouse and keyboard to setup the Raspberry Pi before the touch screen is enabled. You will also need a micro SD card with Noobs loaded onto it before powering. [Installing NOOBS](#). Lastly you will need an ethernet cable to setup internet to download or update the software.

Once you power the Raspberry Pi, Noobs will give you some options of what software platform you want to install. For this project I chose Raspbian (recommended option). [Installing Raspbian](#). After installation you will likely notice that the graphics are distorted or that the width and height of the screen are not the desired dimensions. This is because Raspberry Pi's auto-calibrate feature doesn't recognize this display. So, to get the best picture, we will need to calibrate it manually.

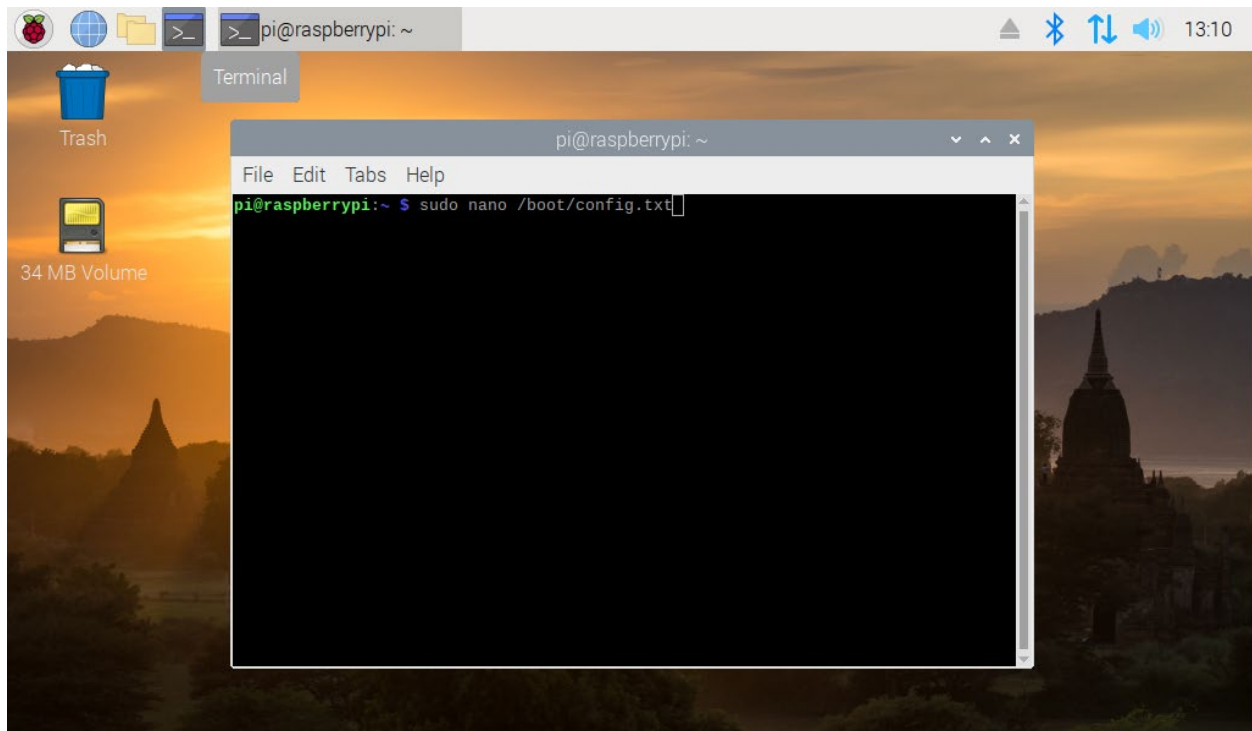
Calibrating the Resolution

There are a couple ways to fix the resolution.

If it's not terrible, and you can still see what you're doing on the display, you can do this through Terminal. We want to edit the "config.txt" which loads from the boot file. The other option is to remove the SD card and load the file on the computer or setup SSH and edit the file remotely.

First open Terminal and execute the following command to open the config.txt file:

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



The file will open in a text editor called nano. We need to add the specifics of our display to the file by adding the following commands to the file:

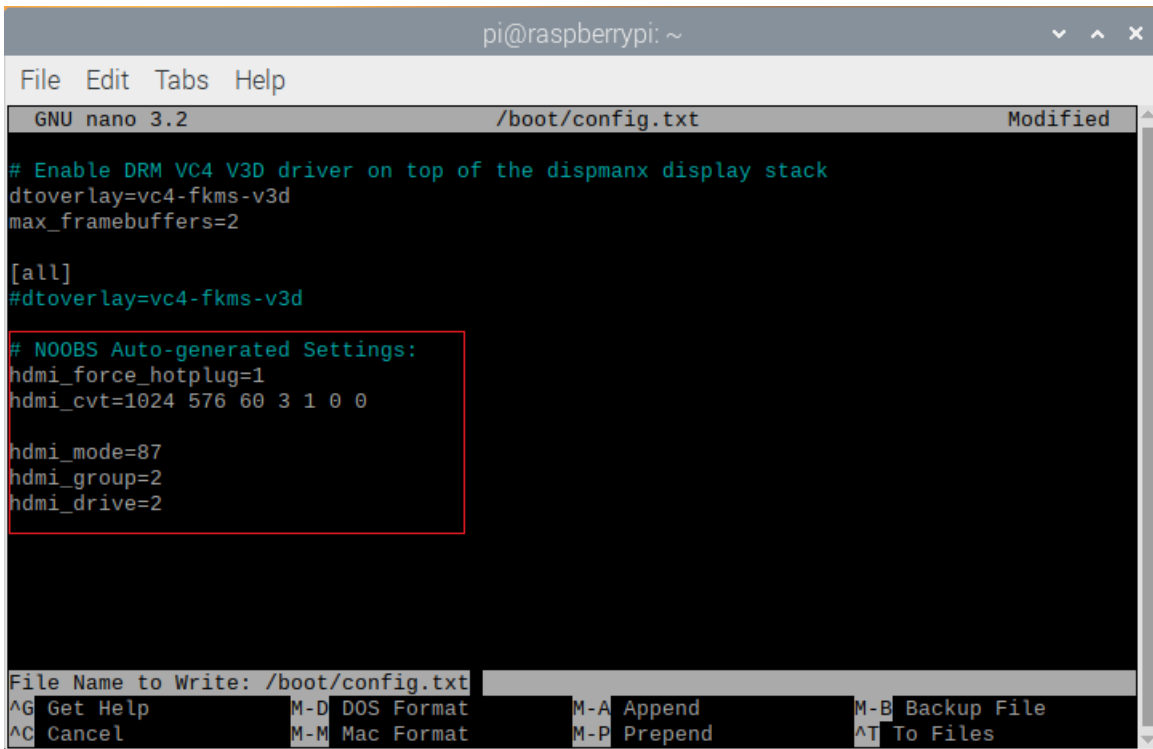
```
hdmi_force_hotplug=1
hdmi_cvt=1024 576 60 3 1 0 0
hdmi_mode=87
hdmi_group=2
hdmi_drive=2
```

The `hdmi_cvt` command creates a custom CVT mode. The input is defined:

```
hdmi_cvt=<width><height><framerate><aspect ratio><margins><interlace><rb>
```

The last three fields of `hdmi_cvt` can be omitted, and they will refer to the default values. The other commands declare the use of HDMI and create a new mode for the `hdmi_cvt` specifications.

For detailed information on these commands and others that function in the `config.txt` file refer to the Raspberry Pi documentation [website](#).



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /boot/config.txt Modified
# Enable DRM VC4 V3D driver on top of the dispmanx display stack
dtoverlay=vc4-fkms-v3d
max_framebuffers=2

[all]
#dtoverlay=vc4-fkms-v3d

# N00BS Auto-generated Settings:
hdmi_force_hotplug=1
hdmi_cvt=1024 576 60 3 1 0 0

hdmi_mode=87
hdmi_group=2
hdmi_drive=2

File Name to Write: /boot/config.txt
^G Get Help      M-D DOS Format   M-A Append      M-B Backup File
^C Cancel        M-M Mac Format   M-P Prepend     ^T To Files
```

Save these changes and reboot the system.

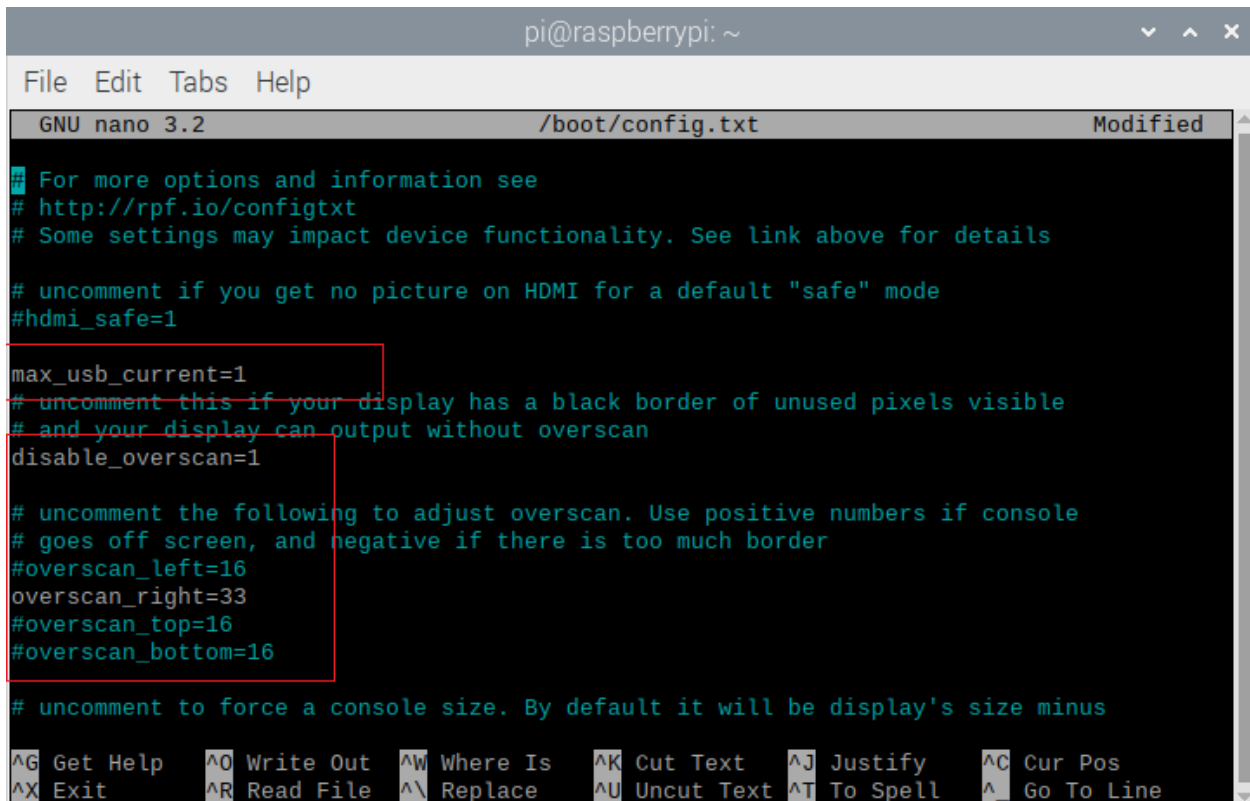
CTRL O (Write Out) → CTRL X (Exit)

```
sudo reboot
```

The resolution is now defined for our display once rebooted.

If the screen is still not perfectly configured for the device, a few other tweaks can be made to the config.txt file. Adjust the overscan parameters to add or remove pixels on the edges. For example, adding 33 pixels to the “overscan_right” parameter corrects overflow of the image on the right side of the screen. This takes a few trial and error attempts to perfect.

```
disable_overscan=1          #erases default overscan values
overscan_right=33
```



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /boot/config.txt Modified
# For more options and information see
# http://rpf.io/configtxt
# Some settings may impact device functionality. See link above for details
# uncomment if you get no picture on HDMI for a default "safe" mode
#hdmi_safe=1
max_usb_current=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=33
#overscan_top=16
#overscan_bottom=16
# uncomment to force a console size. By default it will be display's size minus
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^ Go To Line
```

If you want to power the display through the Raspberry Pi instead of an external power source, you can add the following command:

```
max_usb_current=1
```

This sets the output current at 1A from the USB ports on the Raspberry Pi.

Troubleshooting

If the display is still not configured correctly there are a few things you may want to check in the config.txt file.

- 1.) `hdmi_safe` is OFF.

Verify that `hdmi_safe` is not being used. `hdmi_safe=1` should be commented in the file.

→ `# hdmi_safe=1`

When this command is used it will take precedence over the resolution changes made and revert to the preset HDMI safe mode values and resolution.

- 2.) Display only functions in `hdmi_safe` mode?

If you notice that turning safe mode off results in a blank screen you could have a bad HDMI connection. First try another HDMI cable. If this does not fix the problem, then we can increase the signal strength of the HDMI by editing the `config_hdmi_boost` command in the config.txt file. The standard setting in the configuration file is set as:

→ `config_hdmi_boost=4`

This default value is 2 and can be increased up to 11 if necessary. Try adding this command at increased increments until the issue is solved. You will need to edit the config.txt file on your computer or remotely for this step.

- 3.) Consult Raspberry Pi's configuration documentation for further information on changes that can be made to the config.txt file. ([website](#))

Summary

With only a few changes to the Raspberry Pi's default configuration file the TFT can become compatible with the Raspberry Pi. This display is a great choice for interfacing with the Raspberry Pi because it has an HDMI connection type which makes for any easy setup. After the resolution and is set to a desired value we can now optimize the touch feature of the display. [See here for more information on how to further calibrate this display.](#)

DISCLAIMER

Buyers and others who are developing systems that incorporate FocusLCDs products (collectively, "Designers") understand and agree that Designers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Designers have full and exclusive responsibility to assure the safety of Designers' applications and compliance of their applications (and of all FocusLCDs products used in or for Designers' applications) with all applicable regulations, laws and other applicable requirements.

Designer represents that, with respect to their applications, Designer has all the necessary expertise to create and implement safeguards that:

- (1) anticipate dangerous consequences of failures
- (2) monitor failures and their consequences, and
- (3) lessen the likelihood of failures that might cause harm and take appropriate actions.

Designer agrees that prior to using or distributing any applications that include FocusLCDs products, Designer will thoroughly test such applications and the functionality of such FocusLCDs products as used in such applications.