Running OpenGL-based Games & Emulators on Adafruit PiTFT Displays

Created by Phillip Burgess

https://learn.adafruit.com/running-opengl-based-games-and-emulators-on-adafruit-pitft-displays

Last updated on 2023-08-29 02:46:16 PM EDT

Table of Contents

Overview

The Ideal: Adafruit's PiTFT displays are razor sharp. Whereas small composite screens on the Raspberry Pi usually require some video scaling (resulting in blurriness), PiTFT uses the GPIO header, digitally controlled pixel-by-pixel for a rock steady image. Though not a lot of pixels, it works great for retro gaming (and the display neatly stacks above the board, no side protuberances for video cables).

The Downside: this GPIO link entirely bypasses the Pi's video hardware, including the graphics accelerator. Many games and emulators depend on the GPU for performance gains. So the PiTFT has traditionally been limited to just a subset of specially-compiled emulators that can work and run well enough without the GPU.

The Solution: our latest PiTFT drivers, along with a tool called fbcp (framebuffer copy), careful system configuration, and (optionally) the more potent Raspberry Pi 2 board open the doors to many more gaming options. Existing emulator packages (such as RetroPie, with dozens of high-performance emulators and ports) — previously off-limits to the PiTFT — can run quite effectively now!

The Plan

You'll need:

- Most any model of Raspberry Pi computer (Model A, B, A+, B+, Pi 2 or Zero).
- A 320x240 pixel PiTFT display (2.8" resistive, 2.8" capacitive, 2.2" HAT). For gaming we won't be using the touchscreen features, but still need to distinguish among the various models. The 3.5" PiTFT (480x320) can work but is NOT reco mmended for this project — more pixels means slower refresh.

- A 4GB or larger microSD card (or a full-size SD card for “classic” Model A or B).
- An HDMI monitor and USB keyboard are used temporarily during installation and setup.

Emulators require game ROM files. These are not included. Native, non-emulated ports of Doom, Duke Nukem 3D, Quake and Quake 3 are available (explained on next page) that don’t require additional ROM files.

Steps will include:

1. Download and setup RetroPie using the temporary HDMI monitor.
2. Download and setup additional Adafruit software “on top of” RetroPie.
3. Configuration to redirect game output to the PiTFT screen; HDMI screen is then no longer needed.

Though we focus on RetroPie, some of these steps should be applicable to other software.

This is not a guide for first-timers. Some prior familiarity with Raspberry Pi (SD card prep, network configuration, etc.) is assumed.

Current Cupcade or PiGRRL users...

If you’d like to try this, we suggest using a separate SD card...don’t upset your working Cupcade installation. Set your original card aside for safekeeping! This guide is still experimental. If it all works out well, we’d like to make this our “official” method for gaming with the PiTFT, but not yet...it might not work for everyone, configuration may be troublesome, or the performance might not feel as snappy on some systems.

Any MAME ROMs you’re currently using with Cupcade or PiGRRL might not work with RetroPie — they’re based on different versions of MAME, and the ROM file format they used changed along the way. You may need to convert or acquire new ones.

RetroPie Setup

Let’s begin our adventure on the Retropie downloads page (). Fetch the version appropriate to your Raspberry Pi board type — there are separate SD card images optimized for the Raspberry Pi “1” (Models A, B, A+, B+ and Zero), and another for the Raspberry Pi 2 or 3.

If you've used RetroPie in the past, check if a newer release is available...it's updated frequently! This guide requires version 3.6 or newer.

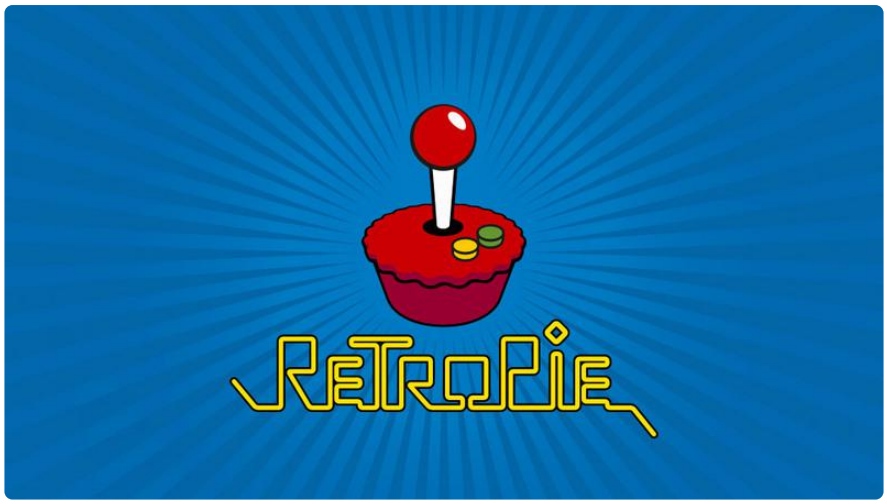
While that downloads, you can get started with formatting a 4GB or larger microSD card (or full-size SD if using a "classic" Model A or B).

After downloading and uncompressing the RetroPie image, you can write it to the SD card as you would any other operating system. It's explained in this guide ().

Use the latest RetroPie image (3.6 or newer) optimized for your board type (Raspberry Pi 1 or 2/3).

Initial Setup...

Insert the SD card in the Raspberry Pi. Plug in an HDMI monitor, USB keyboard, then connect power.



On first boot, you should see some Linux console messages scroll by, followed by a RetroPie splash screen. It will report that it's resizing the SD card partition, then performs a reboot cycle. The second time around, you'll get the RetroPie splash screen, followed by the EmulationStation splash screen, and then it will prompt you to configure an input device.

You do not need to configure inputs yet...instead, press F4 to exit EmulationStation and get a Linux prompt. Then we'll do some basic system configuration...

```
sudo raspi-config
```

The following raspi-config options are required:

- Under System Options:
    - Audio: force 3.5mm headphone jack (since HDMI will be disconnected later). In earlier releases, this is under Advanced Options.
- Under Display Options:
    - Underscan: disable (earlier releases: Overscan in Advanced Options)
- Under Interface Options:
    - SPI: enable

If using an older version of RetroPie, also select Expand Filesystem (recent releases do this automatically on first boot), enable the Device Tree under Advanced Options, and when enabling SPI above, answer “yes” when prompted whether to enable the kernel module.

These steps are optional but recommended:

- Change Password (since everyone knows the default).
- Under Internationalization Options, select Change Locale, Change Timezone and Change Keyboard Layout to your liking. If keys aren't producing the expected characters, this is why.
- Under Advanced Options, change Hostname if desired (default is “retropie”) and enable SSH (for remote administration).

DO NOT select:

- Overclock. We can enable this later, but there are some important details that need explaining with the PiTFT first! Start out at the default speed.
- Memory Split. In the past when using the PiTFT you'd want this as small as possible (16 MB). But now that we're actively using the GPU, we want some GPU memory, so don't touch this!

When you’re done, tab to “finish” and reboot when prompted.

Configure Input

After rebooting, the system will once again ask you to configure an input device. Let's set it up for keyboard input for now (we can change this later). Hold down any key on the keyboard for a few seconds until it starts asking for keys to assign to certain controls.

There are a HUGE number of inputs (shoulder buttons, dual analog sticks, etc.)...but for any you don't anticipate using you can just hold a key down for a few seconds and it will skip ahead to the next option. As an 80's kid, I spend most of my time in MAME, so I configure EmulationStation's input to be fairly minimal (but your emulator of preference might be different):

Button	Key on Keyboard
Up/Down/Left/Right	Arrow keys
Start	1
Select	5
A	Left Control
B	Left Alt or Command

Remember — once configured, EmulationStation is navigated using these keys only. So, for example, the Return key won't necessarily select an item, even though the same key may have the expected effect within some emulators.

You can set set up a more extensive combination of inputs for different emulators later...no need to get serious with the gameplay yet, just confirm that things launch as expected. Most emulators will exit and return to the EmulationStation menu with the Escape key.

Later, you can reconfigure it for a gamepad...you'll need to access the EmulationStation settings menu with a keyboard still attached. Press the key you'd configured as "Start," then select "Configure Input," then proceed through each control as you did before with the keyboard, but using the gamepad now.

The control settings you select here don't always carry through to some emulators. Others may have their own settings that overlap the keys you've chosen, causing mayhem. See the "Configuring Individual Emulators" section for tips on finding each emulator's individual config file, so you can fine-tune these aspects.

Networking

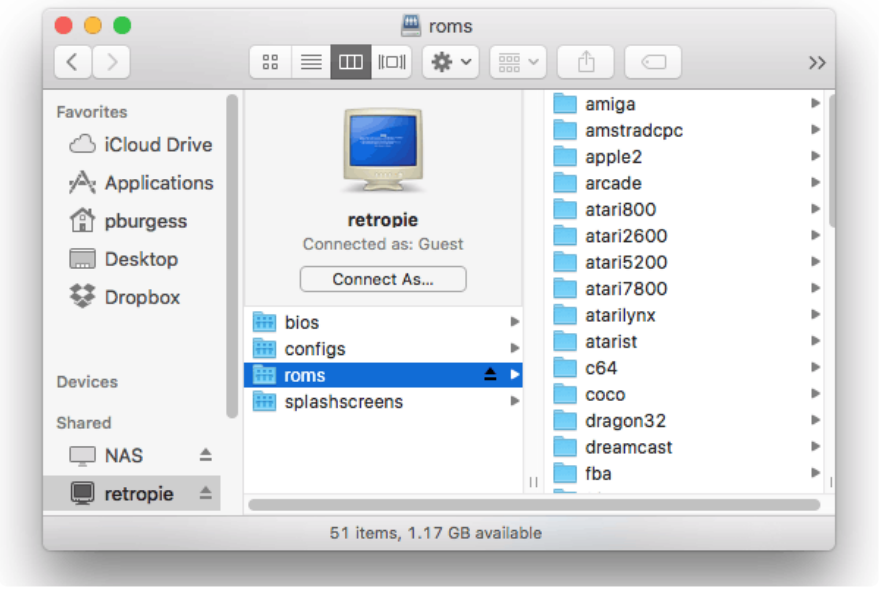
If you don't plan to use networking, skip ahead to the next section below: Install ROMs. You'll need to use a USB flash drive and move files manually from the command-line.

Loading ROM files is most easily done on a wired Ethernet network. Recent RetroPie releases already have file sharing enabled; the system should appear on your network as "retropie.local" (unless you gave the system a different hostname during the initial setup).

If you plan to use wireless networking...the RetroPie menu includes an option for configuring WiFi, but I couldn't get this to connect. If you have trouble with it, press F4 to exit EmulationStation for a command-line prompt and set up WiFi the old-fashioned way, by editing /etc/wpa_supplicant/wpa_supplicant.conf. Here's a guide for basic network setup on Pi.
()

Once networking is all set up (it may require another reboot), you should be able to access the RetroPie system remotely...as a network share for transferring files, and (if you enabled SSH in raspi-config) for remote login to finish certain command-line tasks later.

You can always exit EmulationStation with the F4 key for a command line prompt, but remote login has the benefit that you can copy-and-paste some of the commands we'll be using.



Networking must be configured and operable before proceeding.

Install ROMs

Emulators require ROM files. If you don't want to do this step, that's okay...a few self-contained games can be installed in the "Ports" section of EmulationStation, such as Doom and Quake (explained in next section below).

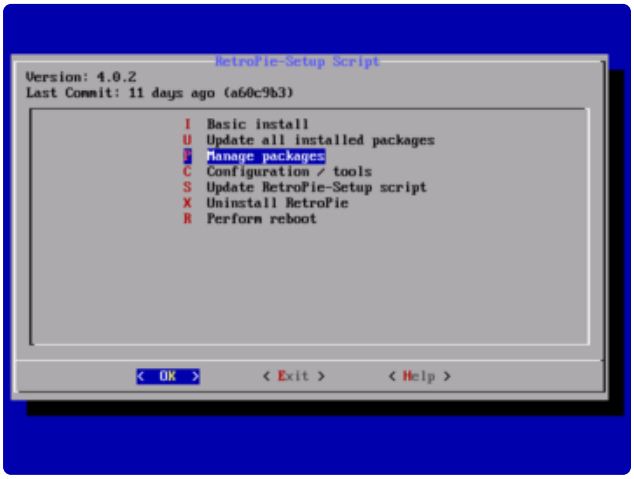
With networking enabled, it's possible to copy the ROM files across the network — the retropie system should be visible as an SMB or AFP share. Alternately, you can fill a USB stick with ROM files, mount this on the Raspberry Pi and move them to the appropriate locations.

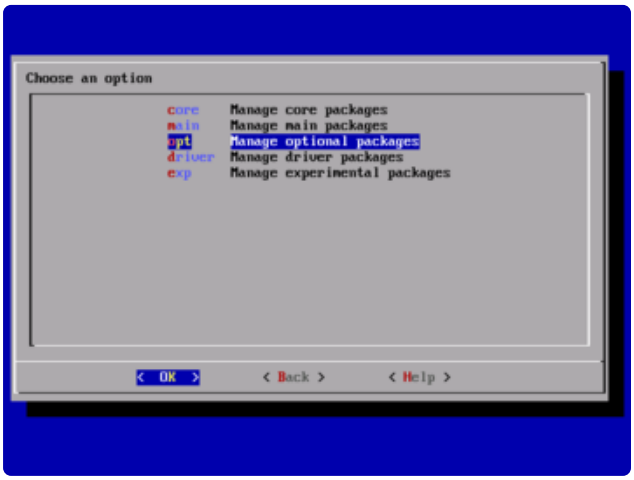
ROM files are installed in specific subdirectories of /home/pi/RetroPie/roms — one for each different emulator package. For example, NES roms go in the "nes" directory.

Some systems are supported by multiple emulators (for example, there are different MAME implementations, each with its own directory). Each emulator has its own quirks — some may render or perform better than others, or some might not work with certain ROMs. Hunting down these quirks and determining your preference for one emulator over another is a process you'll have to work through yourself...there are just too many emulators and far too many ROMs for us to know them all. There are entire sites and forums devoted to the various emulators, and you may need to Google around and do your own research on the best selections for different games. The RetroPie web site () and EmulationStation web site () are great places to start to start, with forums, FAQs and wikis.

Additional Emulators and Ports

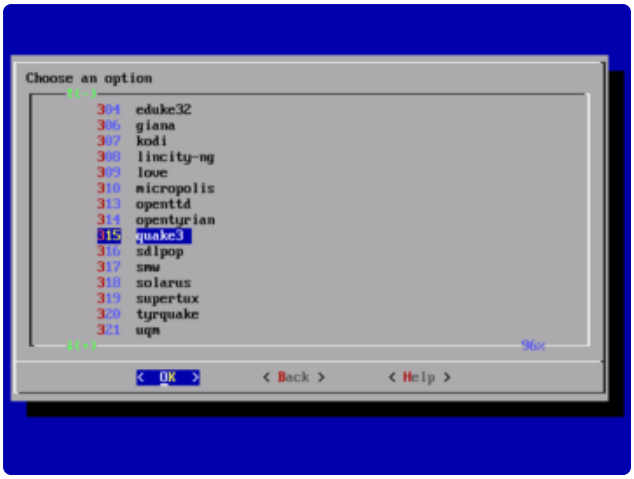
Version 4 (and later) of RetroPie does not include games like Doom and Quake by default. If you're patient and reasonably tech savvy, you can add these on after the fact...

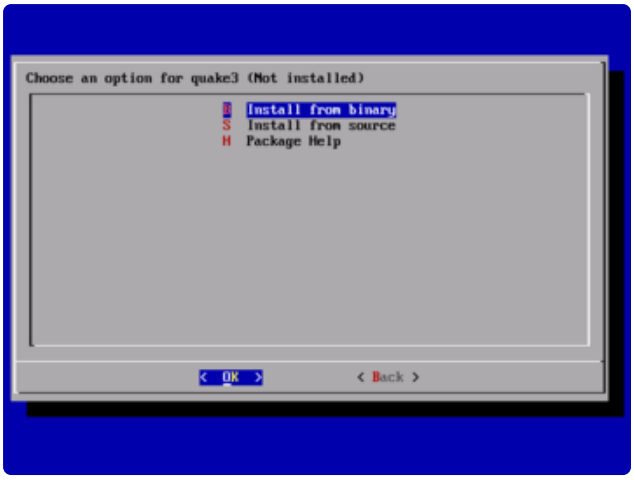




From the RetroPie menu, select "RetroPie Setup." This runs a text-based menu from which options can be selected (use the arrows and enter key here rather than the "A" button).

Select "Manage packages," then "Manage optional packages." You'll be greeted with a list of additional emulators and native (non-emulated) Raspberry Pi ports of some games. Use the "Install from binary" option to install these.





Test It!

An emulated system won't show up in the EmulationStation menu until corresponding ROMs are installed...but it only scans for ROMs at startup. It's easiest just to reboot...or you can exit to the command line (F4) and type "emulationstation" to restart the menu.

If you haven't already configured the keyboard or a gamepad for input, do that now. Hold down a key or button and proceed through each button when prompted.

Do not continue until you're able to successfully launch one or more games, displayed on the HDMI monitor.

PiTFT Setup

This sequence requires RetroPie (3.6 or later) or Raspbian Jessie or Jessie Lite (2016-03-18 or later). “Native” PiTFT support like this is a recent addition...older Raspbian releases (Wheezy, etc.) are not compatible. The 3.5" PiTFT is supported as of RetroPie 4.1 or Raspbian 2016-11-25.

With the emulator(s) working, now we’ll re-route the graphics to the PiTFT. This is the cool part...as far as the Raspberry Pi is concerned, it thinks there’s still an HDMI display attached. This is a departure from how the Cupcade system worked, where the PiTFT was the only display.

If you don’t already have the PiTFT assembled and connected, do that now. Shut down the Raspberry Pi (in EmulationStation, select “Shutdown System” from the “Quit” menu, or type “shutdown” on the command line, wait for the “halted” message and then unplug), solder the socket to the PiTFT board, plug it into the GPIO header (making sure all the pins are aligned) then re-connect power. Don't install any software yet, just solder up the pins if necessary!

When first powered up, the display will be blank white. This is normal.

If you’ve enabled SSH, these next steps are most easily done with a remote login... commands can simply be copied and pasted from the web browser into a terminal window. If not, that’s okay...press F4 to exit EmulationStation and type these commands very carefully.

We’ve written a script to install all the PiTFT-related software and perform some related system configuration. From the command line:

```
cd
curl https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/
master/pitft-fbcp.sh &gt;pitft-fbcp.sh
sudo bash pitft-fbcp.sh
```

When run, this presents a list of Adafruit projects that use the PiTFT display, plus an option for manual configuration.

If using a resistive-screen PiTFT (2.4, 2.8 or 3.2 inches), in most cases you can select the PiGRRL 2 option, this setup will do what you need. At worst, the screen might be rotated 180 degrees, in which case try it again with the “manual” option.

For the manual option: select your PiTFT screen type from the first list. For the second (HDMI rotation), you'll almost always want option #1 (0 degrees rotation), unless you plan to use your display in a vertical "portrait" orientation. For the third selection, it depends which way you plan to orient the display when in use: PiGRRL 2 expects option #4 (270 degrees), or you might want it flipped the other way with option #2 (90 degrees).

```
This script enables basic PiTFT display
support for portable gaming, etc.  Does
not cover X11, touchscreen or buttons
(see adafruit-pitft-helper for those).
HDMI output is set to PiTFT resolution,
not all monitors support this, PiTFT
may be only display after reboot.
Run time ~5 minutes. Reboot required.

CONTINUE? [y/N] y

Select project:
1. PiGRRL 2
2. Configure options manually

SELECT 1-2: 1

Device: pitft28-resistive
HDMI framebuffer rotate: 0
TFT MADCTL rotate: 270

CONTINUE? [y/N] y
```

Before continuing, the current selections will be displayed. When you proceed, the script automatically does the following:

- Updates the package index files used by apt-get
- Installs cmake, needed by the next step
- Downloads and installs fbcp, which mirrors HDMI output to the PiTFT display
- Configures miscellaneous system files to use the PiTFT display

```
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Configuring done
-- Generating done
-- Build files have been written to: /tmp/rpi-fbcp-master/build
Scanning dependencies of target fbcp
[100%] Building C object CMakeFiles/fbcp.dir/main.c.o
Linking C executable fbcp
[100%] Built target fbcp
Configuring PiTFT...
Done.

Settings take effect on next boot.

REBOOT NOW? [y/N]
```

On completion, you'll be prompted to reboot the system. Unless you have other system configuration to perform, do this.

On startup, after a few seconds' delay, you should see the RetroPie splash screen on the PiTFT, followed by the rest of the boot process and then EmulationStation. Go ahead and try launching something!

If an HDMI monitor is still attached, and if it supports 320x240 resolution, you should see the same content on both the monitor and PiTFT. Not all HDMI monitors can display this resolution — you might see "no signal" on the monitor after reboot. That's okay...with everything set up right, it'll all be routed to the PiTFT after a few seconds' boot time.

Once the system is working satisfactoraly, you can disconnect the HDMI monitor. Everything's now done through the PiTFT (or remote login via SSH).

Restoring Original Behavior

If you want to return to using regular HDMI output without the PiTFT screen, there are a couple of options:

1. Do a fresh RetroPie install on the card — this is easiest if you haven't customized a lot of settings or installed lots of emulators and games
2. ...OR... disable the fbcp software and restore HDMI video settings.

To disable fbcp, edit the file /etc/rc.local as root, e.g.

sudo nano /etc/rc.local

and comment out (insert a '#') or delete this line:

```
/usr/local/bin/fbcp &
```

Then edit /boot/config.txt, a la:

sudo nano /boot/config.txt

and comment out or delete these lines:

```
dtoverlay=pitft28-resistive,rotate=270,speed=80000000,fps=60
display_rotate=0
hdmi_cvt=320 240 60 1 0 0 0
```

That will restore the default HDMI resolution on next reboot. If you need different video settings, run raspi-config and explore the Display Options.

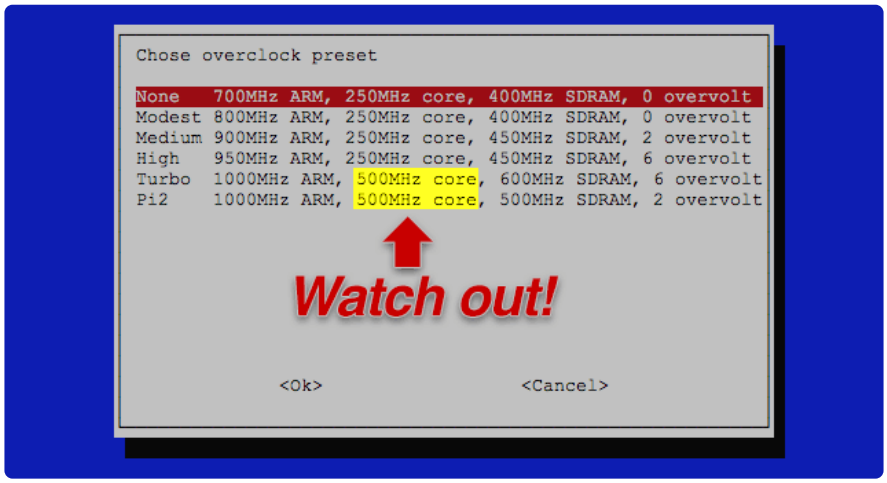
Tuning & Tweaking

Overclocking

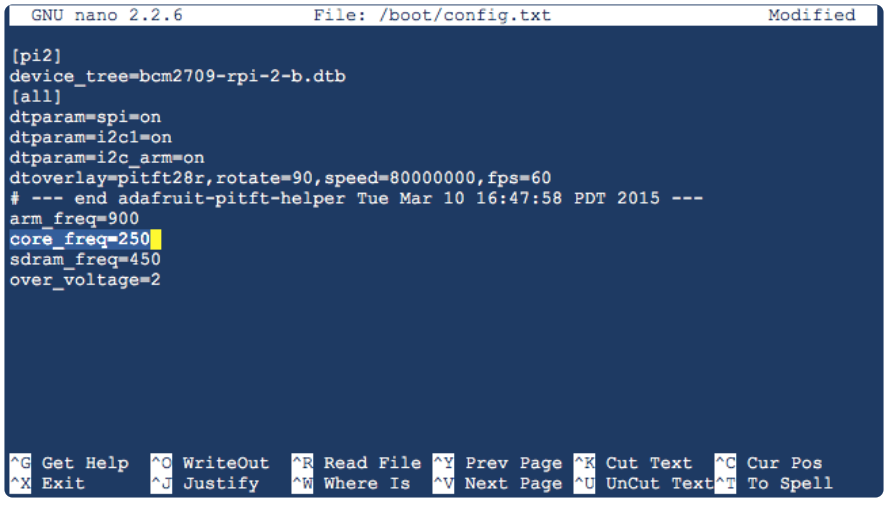
Because the Raspberry Pi is more powerful than many vintage systems, most emulators run at an acceptable speed. A few (such as SNES) aren't quite up to snuff. We can narrow this gap by overclocking the Pi, but there's just one little gotcha to watch out for...

fbcp starts to misbehave if the core frequency exceeds 300 MHz.

Some of the higher overclock settings that raspi-config offers far exceed this...



Therefore, if you enable these faster settings, don't immediately reboot...exit raspi-config to the command line and then edit the file /boot/config.txt to dial back the core frequency to an acceptable range (300 max):



After saving the changes, then reboot.

Moderate overclocking is usually trouble-free, but if you really push it, some systems won't boot. Don't panic! With an SD card reader, the /boot partition will mount normally on most computers, and you can edit the config.txt file there, dialing back the speed until you find a stable setting.

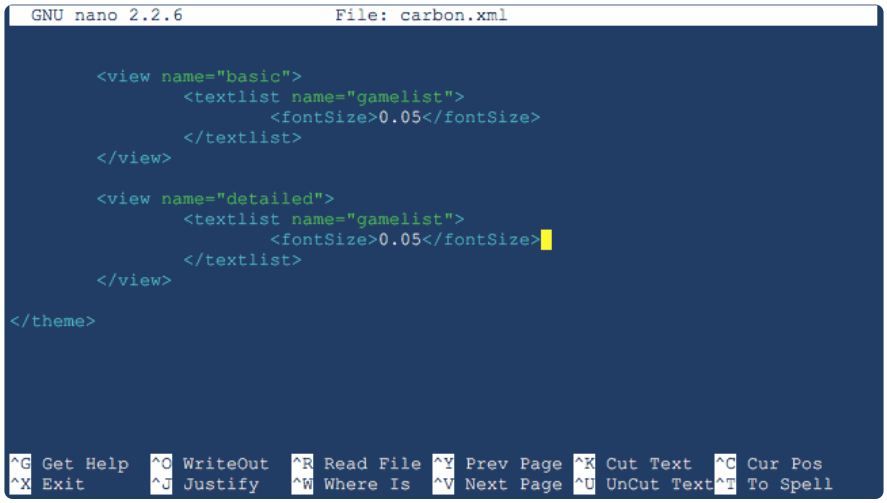
Bigger Text

The EmulationStation UI looks nice on a high-resolution monitor, but on the PiTFT the game lists can be hard to read. We can adjust the font size to help...

```
sudo nano /etc/emulationstation/themes/carbon/carbon.xml
```

(In earlier versions of EmulationStation, the file was called simple/simple.xml, and might change again in the future...you may need to poke around the "themes" directory to find the default settings file.)

Scroll down to the "gamelist" section and change any occurrence of the value of font Size to 0.05.



After saving changes, you can either reboot, or (with USB keyboard attached) press F4 to exit followed by "emulationstation" to restart.

Configuring Individual Emulators

Settings for each emulator are stored in /opt/retropie/configs

Some are configured by editing text files...for example, /opt/retropie/configs/nes/retroarch.cfg

Others provide an interface in the emulator itself...for example, in MAME press the TAB key for a configuration menu that can be navigated with the keyboard.

Configuration for all the emulators is way beyond the scope of a single guide...but each emulator has its own web site and documentation, sometimes forums as well. A little Googling will reveal all the secrets. The RetroPie () and EmulationStation () web sites both have a ton of helpful information, including forums, FAQs and documentation.

Unfortunately the EmulationStation controls you previously configured are not automatically transferred to the emulators...each one will need work. This might be addressed in future versions of EmulationStation.

Adjusting GPU Memory

This tweak is mostly for the Model A or A+ and the very earliest Model B boards. With only 256 MB RAM available, one must split this carefully between CPU (which needs RAM for running emulators) and GPU (needing RAM for graphics). It's a non-issue for later boards with ample RAM.

Edit the file /boot/config.txt

Search for any existing entries for gpu_mem_256, gpu_mem_512, or gpu_mem_1024. You may comment out or delete these lines.

Add a new line, or edit any existing entry for gpu_mem, setting its value to 44 megabytes:

```
gpu_mem=44
```

Save the file and reboot.

Though the Pi can boot with as little at 16 megabytes of GPU memory, 44 MB is the bare minimum required for the EmulationStation menu to work.

With this modification, some parts of the EmulationStation UI might be more visually plain than before...solid colors instead of images, etc. This is normal. In exchange, most games will perform noticably better on these systems with limited RAM.

If you have a Model B, B+ or Pi 2 or 3, this adjustment isn't necessary and won't provide any benefit.

Adding Controls

Half the fun of these PiTFT displays is making tiny portable projects. Having a giant USB keyboard attached runs counter to that. One more piece of software called retro game lets you connect buttons directly to the Raspberry Pi's GPIO header and simulates key presses. We use this in a lot of our gaming projects!

Retrogame is already covered extensively in its own guide:

Retro Gaming with Raspberry Pi: Adding Controls: Hardware ()

Retro Gaming with Raspberry Pi: Installing Retrogame ()

Retro Gaming with Raspberry Pi: Configuring Retrogame ()

Keep in mind you'll want to steer clear of the GPIO pins used by the PiTFT display — they're indicated in that guide.

Rescaling

The native resolution of the PiTFT screen works quite well for retro gaming — quite a few 1980s CRT-based arcade and home systems worked at or around this same resolution.

For certain projects though...emulating later systems, home computers with higher-resolution displays, or if you just want to run GUI-type X11 Linux applications on the PiTFT...there just aren't enough pixels. Application windows and other elements may extend off the right or bottom edges of the screen.

The fbcp program (installed by our script on the PiTFT Setup page) has the ability to scale higher-resolution graphics down to the PiTFT's limited size, with image interpolation so elements like text (unless extremely small) are often still quite legible at the reduced size. fbcp does all this automatically, we just need to change the HDMI output resolution, and on next boot we have a larger graphical canvas (the actual PiTFT resolution itself does not change).

Edit the file /boot/config.txt — use the "sudo" command for root access:

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

Look for this line, most likely near the bottom:

```
hdmi_cvt=320 240 60 1 0 0 0
```

(On the 3.5" PiTFT, the first two numbers will be 480 and 320.)

Try doubling the values to 640 and 480 (or 960 and 640 for the 3.5" PiTFT):

```
hdmi_cvt=640 480 60 1 0 0 0
```

Save changes to the file, then exit and reboot. When the system comes up, you'll see everything is smaller, but antialiased and still generally legible.

You can use even higher resolution settings if needed (up to the Pi's full HD resolution), but this 2X scale is optimal. The bilinear image interpolation algorithm may start to drop details if the HDMI resolution is more than 2X the PiTFT resolution. An exact 2X scale also maintains the same 4:3 aspect ratio as the PiTFT (3:2 for 3.5"), so the image isn't "squashed" horizontally.

Troubleshooting RetroPie and retrogame

This page documents the most common pitfalls encountered when making retro gaming projects, offering some solutions and where to turn for further help.

There are some things we can fix and some we can't...we're not involved in RetroPie's development, for example...but we'll try to point you in the right direction.

Most of the following troubleshooting steps will require a USB keyboard attached. Some require a network connection.

General Troubleshooting

There's a lightning bolt icon in the upper right part of the screen!

- That icon means the Raspberry Pi isn't receiving adequate power; most likely, you need a higher current power supply. Very occasionally this is caused by a flimsy USB cable with thin wires...try swapping out for something heftier.

retrogame Related Troubleshooting

retrogame is our software that converts GPIO button actions into keyboard events. Th ese are the problems we're best equipped to fix.

Some common things to check for any retrogame installation:

- Confirm button/joystick wires go to the correct pins and to ground. A multimeter with a continuity beep function is helpful for testing.
- The retrogame configuration file (/boot/retrogame.cfg) uses Broadcom pin numbers...these are not sequential along the GPIO header pins. This site has a nice reference chart () (use the "BCM" numbers).
- Earlier versions of retrogame didn't use a configuration file...you had to edit and compile the source code. That's just horrible. If you're running an early version like that, this would be a good time to upgrade. See the Installing Retrogame () page.

Some of my buttons/controls aren't working!

- Check connections and configuration file pin numbers as explained above. (If some controls are responding, that's an encouraging start...it at least means the code is running.)
- The key codes generated by retrogame might not be assigned to EmulationStation's keyboard inputs; one or the other will need to be changed. Either edit /boot/retrogame.cfg, or, from the EmulationStation main screen, press Start to access the main menu, then select "Configure Input" and proceed through each of the controls.

NONE of my buttons/controls are working!

- Confirm that retrogame is actually running...either exit to the command line (F4) or log in using ssh, then use “ps -ef | grep retrogame” to check. If you used our installer script or one of our ready-made SD card images, it should be started automatically on boot (added to /etc/rc.local).
- Confirm that the file “/etc/udev/rules.d/10-retrogame.rules” exists. Our installer script creates this file, but if you installed retrogame manually or from source, it may have been overlooked.

My controls only work if there’s also a USB keyboard plugged in!

Confirm that the file “/etc/udev/rules.d/10-retrogame.rules” exists. Our installer script creates this file, but if you installed retrogame manually or from source, it may have been overlooked.

Retrogame doesn’t work with my optical buttons!

Unfortunately, yes. retrogame only works with “passive” switches between a GPIO pin and ground (logic low=pressed). It won’t work with switches that have the opposite logic level (high=pressed).

I ran Adafruit’s retrogame installer script and rebooted, and now the keyboard and network are unresponsive!

This can happen if you’re running an early Raspberry Pi (Model A or B) with the 26-pin GPIO header and select the “Six buttons + joystick” option in the retrogame installer. That particular configuration is set up for our Arcade Pack and newer (40 pin) Raspberry Pi boards. Some of the pin numbers referenced don’t exist on the older 26-pin header and lead to trouble.

If this happens to you, not to worry. Power off the Pi and insert the SD card in a reader on a PC or Mac. Look for a file called retrogame.cfg...edit this file and change the pin numbers to match your specific controller wiring.

RetroPie Related Troubleshooting

RetroPie provides the actual emulation software and a nice user interface. As this is third-party code, the “depth” of problems we can troubleshoot is more limited, but here are some of the common issues we’ve seen and how to resolve them...

My controls work in the EmulationStation UI, but not in one or more specific emulators!

This can happen with certain emulators (usually older or more esoteric ones) that don't use the libretro library. Among other things, libretro allows the global controller configuration to be used everywhere. There are a couple of workarounds that might help, but no guarantees...

- You can rummage around in /opt/retropie/configs and look for a configuration file specific to the problem emulator, then edit its keyboard layout to match your controls. The format of this file, if one even exists, is likely specific to that one emulator, so you'll need to do some research (Google search, etc.), it's not something we can help out with.
- You can try hunting for an alternate emulator based on libretro, if there's one available (see "Installing RetroPie Packages" below).

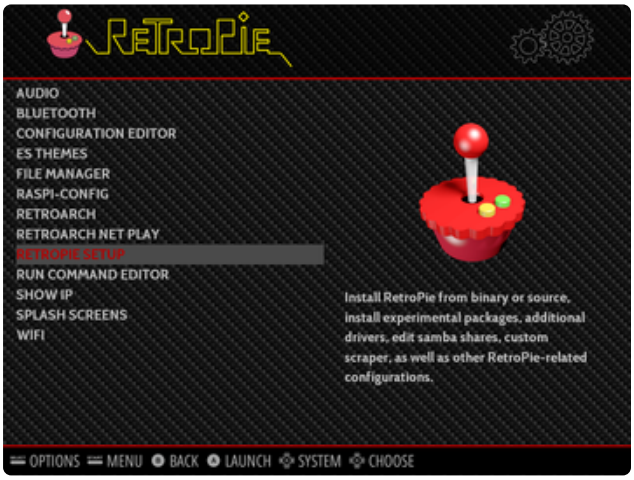
The ROM files I have worked in a different emulator before, but aren't working in RetroPie!

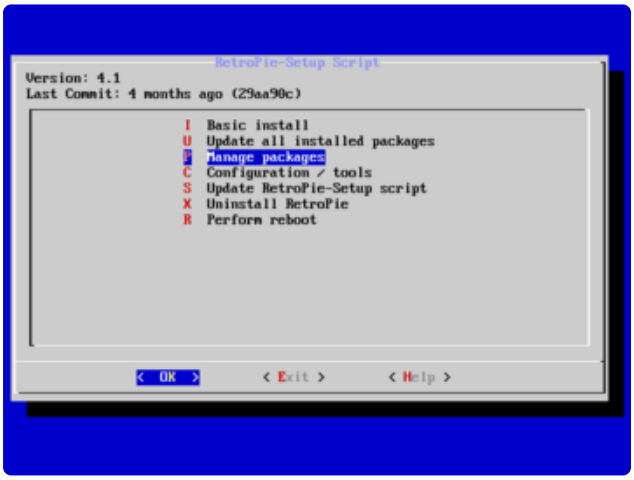
This can happen if the ROM file format changes between versions of an emulator, or if two emulators for the same system use different formats.

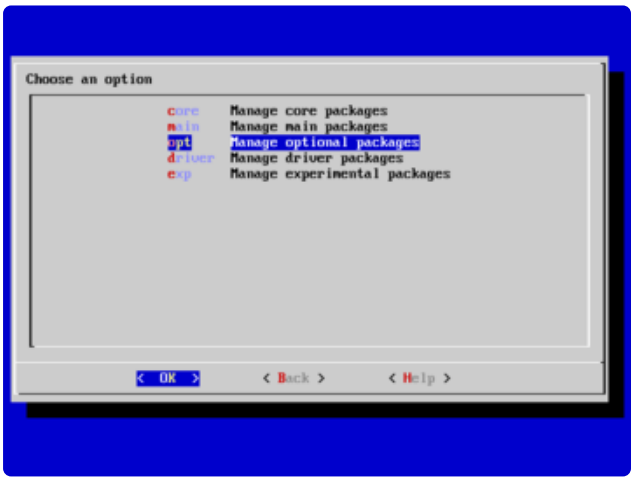
- Do some research (Google search, etc.) to see if this is the case. It's possible there may be utilities to convert among different ROM formats.
- Try installing an alternate emulator, if there's one available (see "Installing RetroPie Packages" below).

A few emulators may require a "BIOS file" in order to function, but it's not included with the software for legal reasons. This is something you'll have to research and track down.

Installing RetroPie Packages







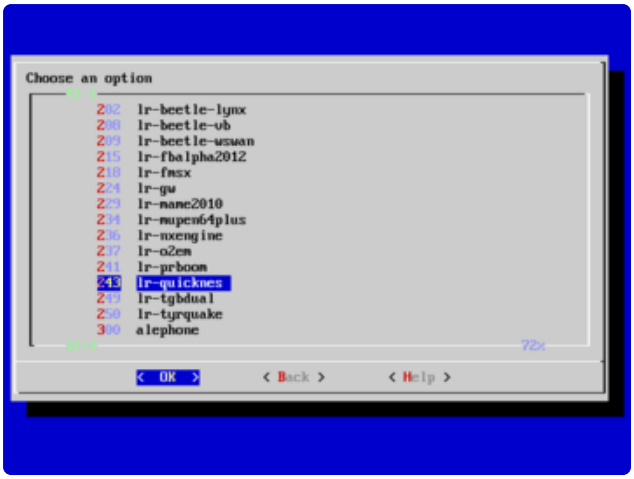
To add support for a system not present in RetroPie by default, or to add an alternate emulator program for an existing system, select “RetroPie Setup” from the RetroPie menu. This brings up a text-menu-based interface and will require a USB keyboard to navigate.

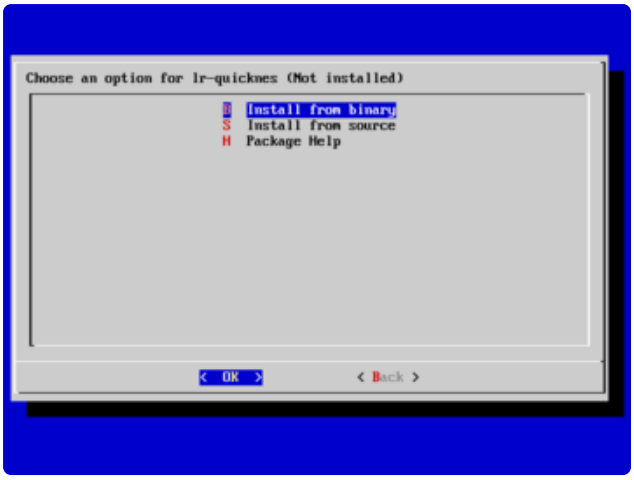
Select “Manage packages” and then one of the core, main, optional or experimental selections...you’ll probably want to navigate through each of them to see what’s available, keeping in mind that each successive category might be a little rougher around the edges.

Your best bet are packages whose names begin with “lr-”. This means they’re built using libretro and the control inputs should already work with what you have! Other packages may require their own manual controller setup, which can be a real nuisance.

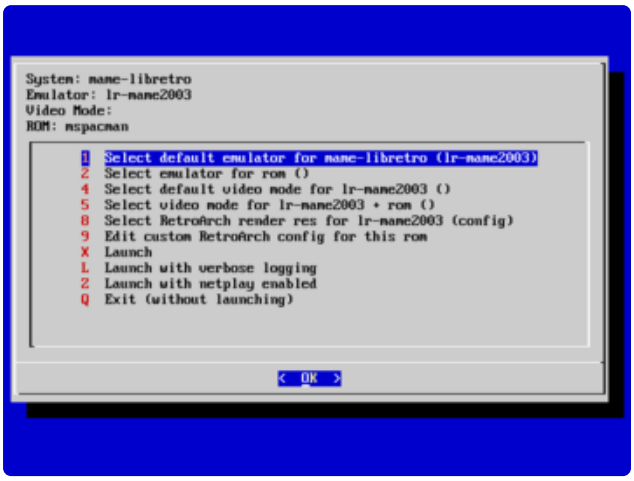
When asked, select “Install from binary.” The source option takes much longer and won’t provide any benefit for the average user...only attempt that if you know you need absolutely bleeding-edge code.

It’s totally valid to install multiple emulator packages that handle the same type of system. Each might have different performance or compatibility benefits, so it’s worth testing your options. See “Accessing Alternate Emulators” below.


Choose an option
202 lr-beetle-lynx
208 lr-beetle-vb
209 lr-beetle-wswan
215 lr-fbalpha2012
218 lr-fmsx
224 lr-gw
229 lr-mame2010
234 lr-mupen64plus
236 lr-nxengine
237 lr-o2em
241 lr-prboom
243 lr-quicknes
249 lr-tgbdual
250 lr-tyrquake
300 alephone
72x
< OK >
< Back >
< Help >


Choose an option for lr-quicknes (Not installed)
B Install from binary
S Install from source
H Package Help
< OK >
< Back >

Accessing Alternate Emulators



When you launch a game, you'll briefly see this nondescript launching message. You have a couple of seconds to hit a button or key...

The first option in this configuration menu lets you select a different emulator package for a given system type...or even on an individual ROM-by-ROM basis, if different games benefit from different emulation software.

Test it out with. If you don't like the results, next time you launch that game you can access the configuration menu again and restore the original selection.

More RetroPie Help

For issues not covered above, the best sources for RetroPie-specific help are the official documentation and forum on the RetroPie web site:

RetroPie Documentation ()

RetroPie Forum ()

Userspace Tools

Sometimes the PiTFT device tree and related kernel package don't work across different OS releases, so we've experimented with an alternate approach that doesn't rely on a custom kernel — it instead works in "user space." So far it's worked well regardless of the OS version being used!

There are tradeoffs. The code is still in a rough state with many features yet to be implemented, and also the performance is slightly less than the kernel approach. It's typically adequate though, even for game emulation (RetroPie, etc.), so give it a try if you've had trouble with the "classic" approach.

This currently requires a bit of Linux-y knowledge, editing files and such...

Download, Test and Install

PiTFT displays use SPI to communicate, so make sure that's enabled using the raspi-config utility:

```
sudo raspi-config
```

Menu options move around from time to time...at the time of this writing, SPI is under "Interfacing Options."

Then retrieve the software using wget...

```
wget https://github.com/adafruit/Adafruit_Userspace_PiTFT/archive/master.zip
unzip master.zip
```

And then a quick test...

```
cd Adafruit_Userspace_PiTFT-master
sudo ./tftcp
```

The PiTFT should mirror the contents of the Raspberry Pi's HDMI output at this point. Text and everything will be microscopic, but we're just checking that the program runs. If not, confirm that the file /dev/spidev0.0 exists — this should happen when SPI is enabled. Double-check raspi-config and it never hurts to reboot.

Does it run? Good. Press control+c to kill the program, and we'll set it up to run automatically on boot.

First, copy the tftcp executable to /usr/local/bin:

```
sudo cp tftcp /usr/local/bin
```

Then edit the file /etc/rc.local as root (you can substitute your editor of preference for nano):

```
sudo nano /etc/rc.local
```

Just above the final “exit 0” line, insert the following line:

```
/usr/local/bin/tftcp &amp;
```

The screen looks best if the HDMI resolution exactly matches the PiTFT resolution, so the final step is to configure the system for 320x240 video:

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

Append the following lines to the bottom of the file:

```
disable_overscan=1
hdmi_force_hotplug=1
hdmi_group=2
hdmi_mode=87
hdmi_cvt=320 240 60 1 0 0 0
```

OPTIONAL: you can also use “640 480” in place of “320 240” above. This is exactly twice the PiTFT native resolution, and the tftcp utility will perform a smooth 2:1 filtering of the image. Any larger though and the image isn’t as sharp (and text becomes tiny, like when we first tested it).

Now reboot and the PiTFT should activate toward the end of the boot process.

Resistive Touchscreen Support

This is even more experimental than the tftcp utility...it only works with the resistive screen, and there’s no calibration support yet, but if you’d like to try it out...

First there’s some prerequisite software to install:

```
sudo apt-get update
sudo apt-get install python-pip python-smbus python-dev
sudo pip install evdev
```

“cd” to the same directory where the software was downloaded earlier, and try it out...

```
cd Adafruit_Userspace_PiTFT-master
sudo python touchmouse.py
```

Whether you’re in X11 or in text console mode (e.g. Raspbian Lite), the cursor should move in response to touch, which is emulating a mouse.

If that seems OK, press control+c to stop it and we'll use the same steps to make it auto-run on boot:

```
sudo cp touchmouse.py /usr/local/bin
sudo nano /etc/rc.local
```

Insert this line just above the "exit 0" at the end of the file:

```
/usr/bin/python /usr/local/bin/touchmouse.py &amp;
```

reboot and both PiTFT and touch should be active now.

Extreme Performance

There's yet another option besides the kernel modules or our userspace code: fbcp-ili9341 (), developed by Jukka Jylänki, takes extreme measures to maximize TFT frame rates and minimize latency. It's the best choice for fast "twitch" gaming.

Some things to be aware of before trying this approach:

- Any PiTFT-related kernel modules you may have previously installed need to be disabled, and SPI needs to be disabled as well. It's probably easiest to begin with a fresh install of RetroPie or Raspbian!
- Getting the very best performance requires lots of tweaking and trial-and-error! If you just want to get games up and running on a PiTFT, other options are simpler and may perform well enough. Remember, non-optimal doesn't necessarily mean pessimal.
- We're not involved in the development of this third-party code. Any issues...like if something stops working with a new OS release...will need to be brought up with the developer.
- Touch is not supported.

Setting Up

Complete instructions are provided on the project's Github page. () Here's an abbreviated walkthrough for getting things up and running quickly:

Prior Pi experience is required. These directions assume you already have the Pi booting and on the network, and with ssh enabled (makes it easier to copy-and-paste the commands that follow). You'll want some games on there for testing, and have configured the controls if using RetroPie.

To begin, let's set the HDMI resolution to match the PiTFT. This makes games more "pixel perfect" on the small display.

Edit the file /boot/config.txt (this requires a "sudo" command) and add the following lines:

```
hdmi_group=2
hdmi_mode=87
hdmi_cvt=320 240 60 1 0 0 0
hdmi_force_hotplug=1
```

For the larger 480x320 PiTFT, replace the values 320 and 240 on the hdmi_cvt line with 480 and 320.

Some newer games might not look good at these coarse resolutions, or menus may appear large and unusable. What you can do is set the HDMI resolution to exactly twice the PiTFT resolution and the software will provide nice 2x2 area sampling...so that's "640 480" for a 320x240 PiTFT or "960 640" for 480x320.

Reboot the system after making this change. If an HDMI monitor is connected, it's possible that it won't sync to these unusually low settings. That's okay, we'll do the rest through ssh...

If playing vector games (e.g. Battlezone, Vectrex games, etc.), things look vastly better if antialiasing is enabled. Edit the file /opt/retropie/configs/all/retroarch.cfg and change this line:

```
video_smooth = "false"
```

to:

```
video_smooth = "true"
```

Download, Build, Test

If using a Raspberry Pi 3 or a late-model (v1.2) Pi 2 along with any of the 320x240 pixel PiTFT displays, enter the following commands:

```
git clone https://github.com/juj/fbcp-ili9341.git
cd fbcp-ili9341
mkdir build
cd build
cmake -DARMV8A=ON -DADAFRUIT_ILI9341_PITFT=ON -DSPI_BUS_CLOCK_DIVISOR=6 -
DSTATISTICS=0 ..
make -j
```

For any Pi 1 or Pi Zero variant, replace:

```
-DARMV8A=ON
```

with:

```
-DARMV6Z=ON
```

For early-model Pi 2 boards (pre-1.2), use:

```
-DARMV7A=ON
```

If using a 480x320 pixel PiTFT, replace the options:

```
-DADAFRUIT_ILI9341_PITFT=ON -DSPI_BUS_CLOCK_DIVISOR=6
```

with:

```
-DADAFRUIT_HX8357D_PITFT=ON -DSPI_BUS_CLOCK_DIVISOR=8
```

Don't forget the two periods at the end of the "cmake" line! If you get a "CMake Error:" message, that's probably the reason.

You can then test the program with:

```
sudo ./fbcp-ili9341
```

This should mirror the current contents of the HDMI display to the PiTFT. Press Control+C to exit, and we'll continue with more setup...

Fine Tuning

The above commands should get you up and running with decent performance in most cases. If you're seeing any graphic "glitches" on the display, or if you really want to fine-tune settings to their fullest, the project's README () explains every detail.

If working to improve the frame rate, you'll want statistics enabled. These are overlaid on game graphics, so you'll want to work everything out and then disable statistics afterward.

To enable statistics, on the cmake line, replace:

```
-DSTATISTICS=0
```

with:

```
-DSTATISTICS=1
```

(Or 2 if you want a frame rate graph as well.)

After making changes with cmake, rebuild the software and retest:

```
cmake [options] ..
make -j
sudo ./fbcp-ili9341
```

Test out your favorite games using various settings (with guidance from the README) to narrow in on the best performance, then build once more with statistics disabled.

Run on Startup

When everything looks good, let's set up the system to run this automatically...

Edit the file /etc/rc.local (this requires a "sudo" command) and insert a line just above the final "exit 0":

```
/home/pi/fbcp-ili9341/build/fbcp-ili9341
```

(Change the path if the software is situated elsewhere.)

Reboot and the software should run automatically. It may take up to a minute before it starts and shows anything on the PiTFT. This thread () in the software's repository talks about ways to get it started quicker at boot-time with a little extra work.