

Setting up a Raspbian Raspberry Pi to run a video or other dedicated application that appears to take over the machine. Dale E. Parson, April 2019. Copyright 2019 by Dale E. Parson, using the Creative Commons Attribution 4.0 International (CC BY 4.0), which allows free distribution and reuse as long as you attribute me as the author.

Work done April 2019 to support a turnkey video display for my 4:30 minute art video *Flaming Beauty* accepted for the 2019 Art of the State exhibit at the State Museum in Harrisburg.

(<https://ethosting.s3.amazonaws.com/artofthestatepa/index.html>)

1. Attach Ethernet cable from Pi to LAN <-> Internet, USB keyboard & mouse cables, and HDMI cable to a monitor. Connect power cord to outlet strip; turn on outlet strip.
2. If the NOOBS OS-installation tool comes up by default (new Pi card), or if you hold down the shift key before OS boot to bring up NOOBS for reimaging, set terminal and language to U.S. Choose the Raspbian [Recommended] OS. Go through the install and update steps to get the most recent release of Raspbian. Note the LAN IP address in one of the configure panels that comes up. Running *sudo apt-get update* from a bash terminal should be enough if Raspbian is already installed. If the latter case, *sudo vi /etc/default/keyboard* and change *gb* to *us* if the '*|*' does not type correctly. I use *vi* instead of the *nano* editor because I can. ☺ Also, use *ifconfig* from a terminal to find the IP address. I stayed with the default */home/pi* account for my working area. It has super-user access.
3. After system is up-to-date and you reboot, go through the following menu path:

Rasperry (upper left of GUI) -> Preferences -> Raspberry Pi Configuration -> Interfaces and Enable SSH. This allows you to *ssh -l pi IPADDRESS* into the Pi via the LAN from a laptop/PC terminal window as long as the Pi is connected to the LAN via an Ethernet cable. It does not work for me if the Pi is attached to the LAN only via wireless. Remote login is essential if you happen to hang the main window without crashing the machine, which I did many times on my way here. You can edit incorrect files, and you can kill your processes and even *sudo shutdown now* in order to break a hung process that disables the keyboard, mouse, & main HDMI display when the hung process is not attached to a terminal. Your laptop can use wireless. You do have to prefix certain commands with *sudo* from a remote terminal in some cases where *sudo* is not necessary on the console; error messages will indicate the need.

4. I tried several classic Unix approaches to booting to a dedicated application (<https://raspberry-projects.com/pi/pi-operating-systems/raspbian/auto-running-programs-gui>), and most of them failed in annoying or even machine-stopping ways. Here is what worked:

Assuming that I am using the pi account, here is what I copied via scp from my laptop to directory /home/pi/:

FROM MY LAPTOP:

```
ku135515parson:scripts parson$ scp
../graphics/ParsonFlamingBeautyAOS2019.mov
pi@10.0.1.11:/home/pi/ParsonFlamingBeautyAOS2019.mov
pi@10.0.1.11's password:
ParsonFlamingBeautyAOS2019.mov          100% 445MB  7.7MB/s  00:57
ku135515parson:scripts parson$ scp FlamingBeauty
pi@10.0.1.11:/home/pi/FlamingBeauty
pi@10.0.1.11's password:
FlamingBeauty                          100% 885  163.7KB/s  00:00
ku135515parson:scripts parson$ scp FlamingTerminal
pi@10.0.1.11:/home/pi/FlamingTerminal
pi@10.0.1.11's password:
FlamingTerminal                        100% 52   20.8KB/s  00:00
```

ON THE PI:

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls -lrt
total 455764
(pre-installed directories...)
-rw-r--r-- 1 pi pi 466656565 Apr 17 17:51 ParsonFlamingBeautyAOS2019.mov
-rwxr-xr-x 1 pi pi   885 Apr 17 17:51 FlamingBeauty
-rwxr-xr-x 1 pi pi   52 Apr 17 17:52 FlamingTerminal
```

Use *chmod +x Flaming** if the above execute permissions are not on. Those files constitute my application, which plays the ParsonFlamingBeautyAOS2019.mov video file:

```
pi@raspberrypi:~ $ cat FlamingTerminal
#!/bin/bash
```

```
lxterminal -e /home/pi/FlamingBeauty
```

FlamingTerminal starts application **FlamingBeauty** via a terminal window/process. I have found using a terminal framework necessary in order for the 'q' key (which omxplayer interprets as quit) and Alt-F4 key combination (which Raspbian interprets as kill-this-application) to have an effect when a USB keyboard is attached for debugging. If my scripts start **FlamingBeauty** directly, there is no way to kill it from its terminal, necessitating a remote kill from my laptop via the LAN. Here is **FlamingBeauty**:

```
pi@raspberrypi:~ $ cat FlamingBeauty
#!/bin/bash
```

```
omxplayer --aspect-mode fill -o hdmi --no-osd --loop
/home/pi/ParsonFlamingBeautyAOS2019.mov # & # DO NOT BACKGROUND THE
FINAL COMMAND!
# USE THIS BACKGROUND AMPERSAND IF YOU ENABLE THE SLEEP/SHUTDOWN
BELOW
# sleep 300 ; sudo shutdown now # TODO ENABLE WITH > 300 IF YOU LEARN A
TIME.
```

```
# Note on the above sleep/shutdown nonsense:
# If I determine that it should always shut down some fixed time after
# booting this app, I must do four things:
# 1. Run omxplayer in the background by adding a & to that command.
# 2. Change 300 to the correct fixed number of seconds.
# 3. Uncomment the sleep, shutdown line above.
# 4. Reboot.
```

```
# I recommend having a keyboard and mouse attached for debugging,
# or a hard-wired Ethernet connection over which you can ssh,
# to debug any changes. Don't go < "sleep 300" or you won't
# have time to edit this file before the pi reboots!!!
```

As noted in the **FlamingBeauty** comments, the final commands need to run without '&' backgrounding. I experimented with auto-shutdown as seen above in case I want the box to turn off after some set number of hours after boot, presumably based on length of museum hours. I don't know whether I will use that feature, so it is disabled at present.

Finally, to hook **FlamingTerminal** into the GUI boot process, you need to create the following directories and files. One Pi that I used already had these directories; the other did not. Make sure to use *sudo* to give them root ownership. I don't know if that is necessary. I don't have time for that experiment. From pi's home directory:

```
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ ls -ald .config/
drwx----- 3 pi pi 4096 Apr  8 06:31 .config/
pi@raspberrypi:~ $ ls -al .config/
total 20
drwx----- 3 pi pi 4096 Apr  8 06:31 .
drwxr-xr-x 15 pi pi 4096 Apr 17 17:52 ..
drwx----- 3 pi pi 4096 Apr  8 06:31 lxpanel
-rw----- 1 pi pi 632 Apr  8 06:31 user-dirs.dirs
-rw-r--r-- 1 pi pi  5 Apr  8 06:31 user-dirs.locale
```

```
pi@raspberrypi:~ $ sudo mkdir .config/lxsession
pi@raspberrypi:~ $ ls -ld .config/lxsession
drwxr-xr-x 2 root root 4096 Apr 17 18:15 .config/lxsession
pi@raspberrypi:~ $ sudo mkdir .config/lxsession/LXDE-pi
pi@raspberrypi:~ $ ls -ld .config/lxsession/LXDE-pi
drwxr-xr-x 2 root root 4096 Apr 17 18:15 .config/lxsession/LXDE-pi
pi@raspberrypi:~ $ sudo vi .config/lxsession/LXDE-pi/autostart
```

Insert this line into autostart and save:

```
@/home/pi/FlamingTerminal
```

Next make autostart executable. I am not certain this is necessary, but it doesn't hurt:

```
pi@raspberrypi:~ $ sudo chmod +x .config/lxsession/LXDE-pi/autostart
pi@raspberrypi:~ $ ls -l .config/lxsession/LXDE-pi/autostart
-rwxr-xr-x 1 root root 26 Apr 17 18:17 .config/lxsession/LXDE-pi/autostart
```

Now you can reboot into the application via this command line, typed either remotely via *ssh* or from a terminal window on the console:

```
pi@raspberrypi:~ $ sudo shutdown -r now
Connection to 10.0.1.11 closed by remote host.
Connection to 10.0.1.11 closed.
```

Leave out the “-r” if you want to shut down without reboot.

These are the steps I needed to make a Raspberry Pi running Raspbian into a video viewer. Python, Java, and Processing have their own approaches to bundling applications that you can start from `.config/lxsession/LXDE-pi/autostart`.

You can disconnect the cables except for the HDMI and power cables after debugging is done. My plan is to have the museum turn off the video monitor, leaving the Raspberry Pi running during the installation. This application does not consume much power nor make the Pi hot. Make sure to install the heat sinks on the two biggest Pi integrated circuits. If it hangs, power cycling its outlet strip should get it going again. (“Have you tried turning it off and on?”) I have not had the need to do that after several days and many hours of continuous running, but the exhibit runs for over 3 months, so it has to keep running. I am hopeful. I will take a keyboard, mouse, and small LAN router along for setup and visits. Hopefully, there will be no extra trips to Harrisburg.