The Chronicle

of a ColdFusion Expatriate

January 6, 2020

Raspberry Pi on DisplayLink

So I'm sitting here on my computer—something I do quite often—and find myself staring at this Lilliput 7" USB display that I have here. I've had this thing for a while; I bought it to use for Slack at work originally, and it worked OK. 7" turns out to be just slightly too small for anything serious, but it worked alright for chat at least.

Anyway, I brought it home and hooked it up to my Windows PC and never found anything really great to use it for. Suddenly I had a brainstorm: what if it could work on a Raspberry Pi? Then I could put some useful home automation output on it and maybe position it somewhere else in the house.

I set about figuring out how to get the Raspberry Pi to send GUI output to this USB-powered DisplayLink monitor. I did, eventually, get it to work. But, I ran across so many conflicting, incorrect, and outdated articles that I am compelled to describe what I did to make it work.

This Lilliput display uses DisplayLink, which is a proprietary USB graphics chip with associated drivers and protocol. Windows has DisplayLink support built in, but other operating systems typically need some added software to make it go.

In Linux, the driver that you need is called `udlfb`, which is provided as a kernel module. This module should already be there on recent Raspbian builds; I didn't have to do anything to get it.

In Linux, the DisplayLink device appears to the system as a *framebuffer*, which is a hardware-independent API for accessing video memory. I could make some assumptions about how DisplayLink works based on this, but I won't because they'll probably be embarrassingly wrong. Suffice it to say, you just need a way for your Raspberry Pi system to output whatever you want to display to a framebuffer.

The `udlfb` driver talks to the DisplayLink screen via USB and exposes that interface through the Linux framebuffer device at `/dev/fb1`.

You can confirm that this is working by running `dmesg`. In that log output, you should find lines similar to these:

```
[    3.431954] usb 1-1.2: new high-speed USB device number 6 using dwc_otg
[    3.565400] usb 1-1.2: New USB device found, idVendor=17e9, idProduct=02a9, bcdDe
vice= 1.50
[    3.569491] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[    3.573677] usb 1-1.2: Product: USB Monitor
[    3.575785] usb 1-1.2: Manufacturer: DisplayLink
[    3.577844] usb 1-1.2: SerialNumber: 78530686
[    5.517598] usb 1-1.2: fb1 is DisplayLink USB device (800x480, 1504K framebuffer
memory)
```

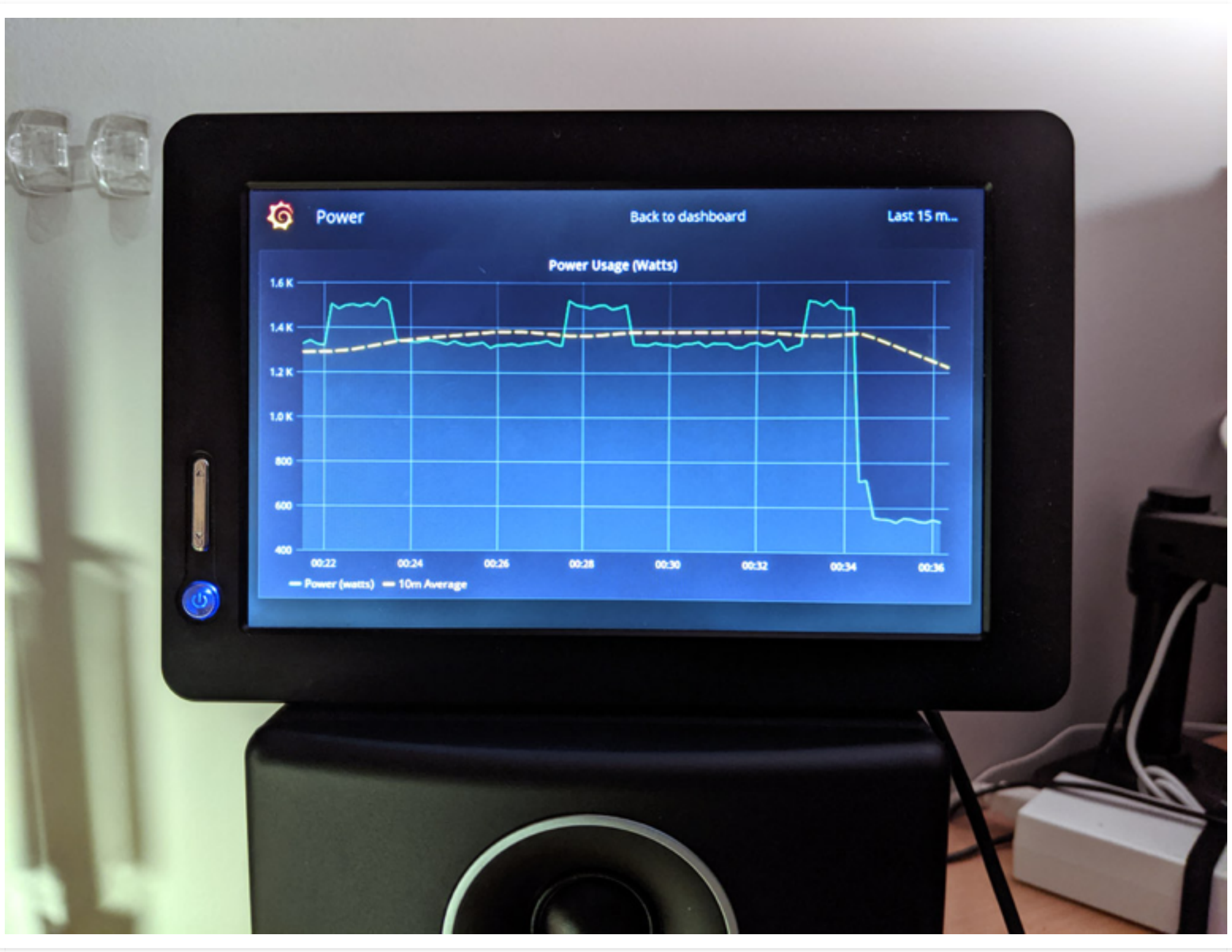
Then all you have to do is inform the X-Windows system that you have a monitor using that device and what driver it should use. To do that, you must edit (or create) the file `/etc/X11/xorg.conf`. I had to create this file, and this is its entire contents:

```
Section "ServerLayout"
        Identifier "Server Layout"
        Screen 0 "DisplayLinkScreen" 0 0
EndSection

Section "Files"
        ModulePath "/usr/lib/xorg/modules"
        ModulePath "/usr/local/lib/xorg/modules"
        ModulePath "/usr/local/lib/xorg/modules/drivers"
EndSection

Section "Device"
        Identifier "DisplayLinkDevice"
        Driver "fbdev"
        Option "fbdev" "/dev/fb1"
        #Option "ShadowFB" "off"
EndSection

Section "Monitor"
        Identifier "DisplayLinkMonitor"
EndSection

Section "Screen"
        Identifier "DisplayLinkScreen"
        Device "DisplayLinkDevice"
        Monitor "DisplayLinkMonitor"
        SubSection "Display"
                Depth 16
                Modes "800x480"
        EndSubSection
EndSection
```

The `Identifier` values are meaningless, it's only important that the references in the `Screen` section are correct. The important part is the `Driver` and `Option` keys in the `Device` section, where we tell X that we have an `fbdev` device and that it is located at `/dev/fb1`.

Once you have set all of this up, you should be able to reboot the Pi with only your DisplayLink monitor plugged in and see your GUI desktop environment appear!

Here is my Lilliput monitor proudly displaying a live power usage graph in Grafana in Firefox, fullscreen:



I'm sure you have questions. I may have answers… You can try in the comments below!

Posted by Aaron Bieber • Jan 6, 2020

« Introducing Hugo.el

My Surface Pro Journey »

Hi, I'm Aaron Bieber. Once, long ago, I wrote ColdFusion for a living. Now I lead software engineers in Boston writing a variety of non-ColdFusion languages.

Recent Posts

"Leadership BS" Is BS

How the 1990s Made Me an Introvert

Do You Need a CS Degree?

How Did I Start?

America Is in Love With Work

Comments

What do you think?

6 Responses

Upvote Funny Love Surprised Angry Sad

11 Comments | Aaron Bieber | Disqus' Privacy Policy | 1 Login

Favorite 1 | Tweet | Share | Sort by Best

Join the discussion…

LOG IN WITH | OR SIGN UP WITH DISQUS ? | Name

The Urban Spaceman · 2 years ago
Awesome. I'd been trying for a while to get this to work on my Pi4 and a Mymo screen. To note, I had to replace the "fb1" reference with "fb0" after doing a dmesg to identify my device was mapped to that ID.
1 · Reply · Share ›

Don Harper · 2 years ago
And it work! Thanks.
1 · Reply · Share ›

Azar Ali Zain · 2 months ago
This worked fantastic on a Raspberry Pi Zero W that I was using with a Lenovo ThinkVision USB powered monitor. The monitor draws power from the micro USB post and surprisingly working without a single glitch! I setup SSH and configured Bluetooth mouse/keyboard, so it all seems awesome!

- Power bank
- Rpi Zero W
- Lenovo ThinkVision USB Monitor
- Android app with bluetooth mouse connected to Rpi

Works flawlessly.
Reply · Share ›

Sean001 · 5 months ago
View — uploads.disquscdn.com

Hi Aaron, I am new to Raspberry/linux. I have a DisplayLink dock connected to my Pi400 usb 3.0 port. When i run "dmesg", I can see it detect DisplayLink dock but the "fb1" is not showing. Please see my attachment.
Any idea what is missing? Hope you can advise, thank you!
P/S: The DisplayLink dock is linking to a monitor with vga input.
Reply · Share ›

D Framstead · 8 months ago
I am trying to get a Raspberry PI 4 w/ 8GB to play an mp4 video file on a MiniRay DLP projector (DisplayLInk) with NO other display connected. I have installed a new Raspbian OS and followed your instructions above without success. I can get the Projector to work if I have another display connected to the HDMI_0 but if it is disconnected I get a splash screen <included image=""> .

Do you have any suggestions on how to correct this?

Thank you

View — uploads.disquscdn.com
Reply · Share ›

> **aaronbieber** Mod → D Framstead · 8 months ago
> Wish I could help, but I don't know what could be wrong!
> Reply · Share ›
>
> > **D Framstead** → aaronbieber · 8 months ago
> > Thank you for thinking about it.
> > Reply · Share ›

Brian Watson · a year ago
Thank you so much!!! I had been trying to get a DisplayLink monitor to work, and this tutorial worked perfectly!
Reply · Share ›

C Stanford · a year ago
I tried this on my rpi3 and my displaylink display (AOC E1659FWU) was detected upon plugin, but did not automatically create a /dev/fb device. I can see the detection in dmesg, and with a "modprobe udlfb" it told me "usbcore: registered new interface driver udlfb" in dmesg.
I'm using the latest kernel 5.4.51-v7+.
Any ideas of how to have it create the dev/fb1 device?
Thanks.
Reply · Share ›

> **C Stanford** → C Stanford · a year ago
> Turns out I have a usb 3.0 displaylink chipset which are completely incompatible with the existing usb 2.0 kernel drivers (udlfb and udl).
> Reply · Share ›
>
> > **Stephen Hazel** → C Stanford · a year ago
> > so you're pretty sure about that? i would KILL to be able to have my raspi4 be able to use my pluggable-ud3900 usb3 hub sigh :)
> > but it's a usb3 hub which means probably no dice, eh?
> >
> > How the heck did you determine this? just that /dev/fb1 (or whatever number) never showed up?
> >
> > any details would be most appreciated :)
> > And thanks for this !!
> > Reply · Share ›