

AUTODESK
Instructables

Share WiFi With Ethernet Port on a Raspberry Pi

By [WaltW](#) in [CircuitsRaspberry Pi](#)



Introduction: Share WiFi With Ethernet Port on a Raspberry Pi



Do you have an old laser printer or scanner that still works great but isn't wifi compatible? Or maybe you want to connect an external hard drive as a backup device on your network and you've run out of ethernet ports on your home router. This instructable will help you create a bridge from the wifi connection to the ethernet port on a Raspberry Pi.

I needed a way to connect an older Xerox copier/printer that has a built in network adapter and network software but wasn't wifi compatible. This printer was in an older building and the printer was in a location that wasn't close to an ethernet punch down and couldn't be moved. With a few parts that I already had around my house I was able to put together a solution that solved my needs.

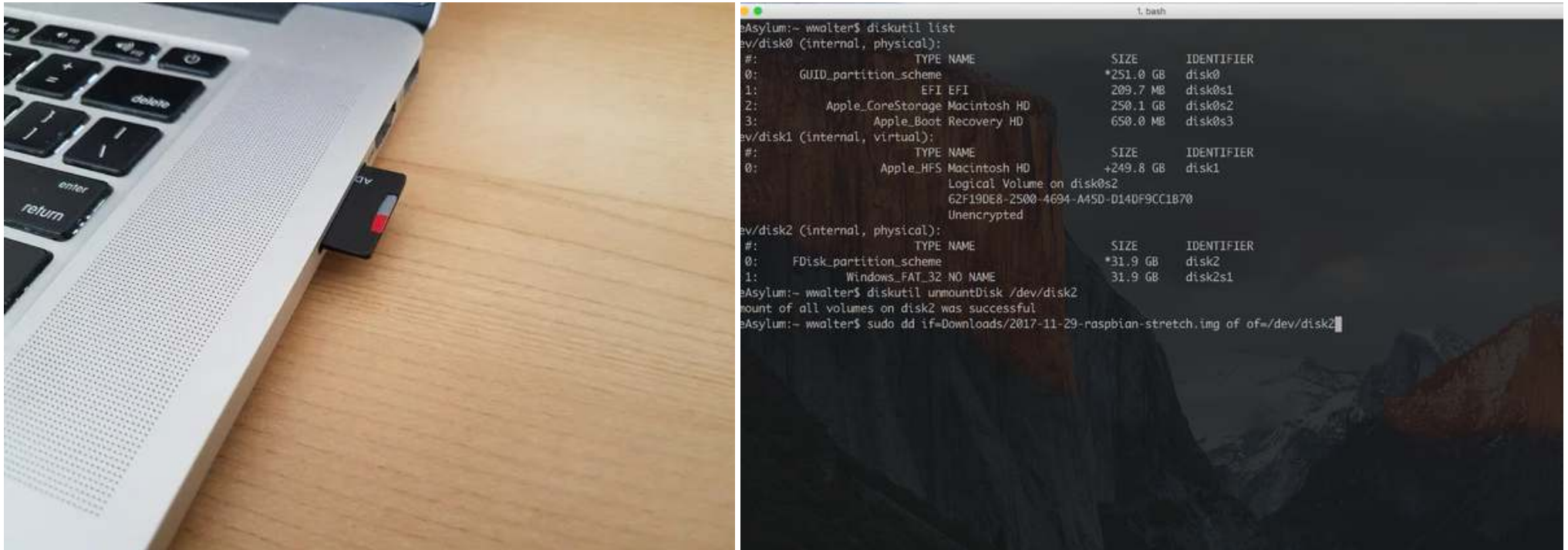
This easy DIY solution will give you the capability of adding a wifi connection to your older devices without breaking the bank by purchasing wireless print adapter.

Step 1: Items You'll Need



-
1. [Raspberry Pi](#) (any model will do, but you'll see faster results with a model 3).
 2. [Power adapter](#) for your Pi.
 3. [SD Card](#) to install the operating system on (you could use an 8GB card up to whatever size you want. I usually go with a 32GB card in case I want to add more options to the RPi).
 4. [Wifi adapter](#)
 5. [Ethernet cable](#)
 6. [HDML cable](#) (I have an old DVI monitor so I use this [HDMI to DVI cable](#)).
 7. [Keyboard and mouse](#)
 8. [Card reader](#) or computer with built in card reader.
 9. [A case for the Pi](#) (optional)

Step 2: Install Raspbian Operating System



Download the latest version of Raspbian (Stretch as of this writing) operating system from [here](#). Insert the SD card in the adapter that ships with it and put the adapter with the SD card into your card reader. Copy the Raspbian operating system image onto the SD card by using these instructions:

- [Windows instructions](#)
- [Mac OSX instructions](#)
- [Linux instructions](#)

Step 3: Assemble the Remaining Components



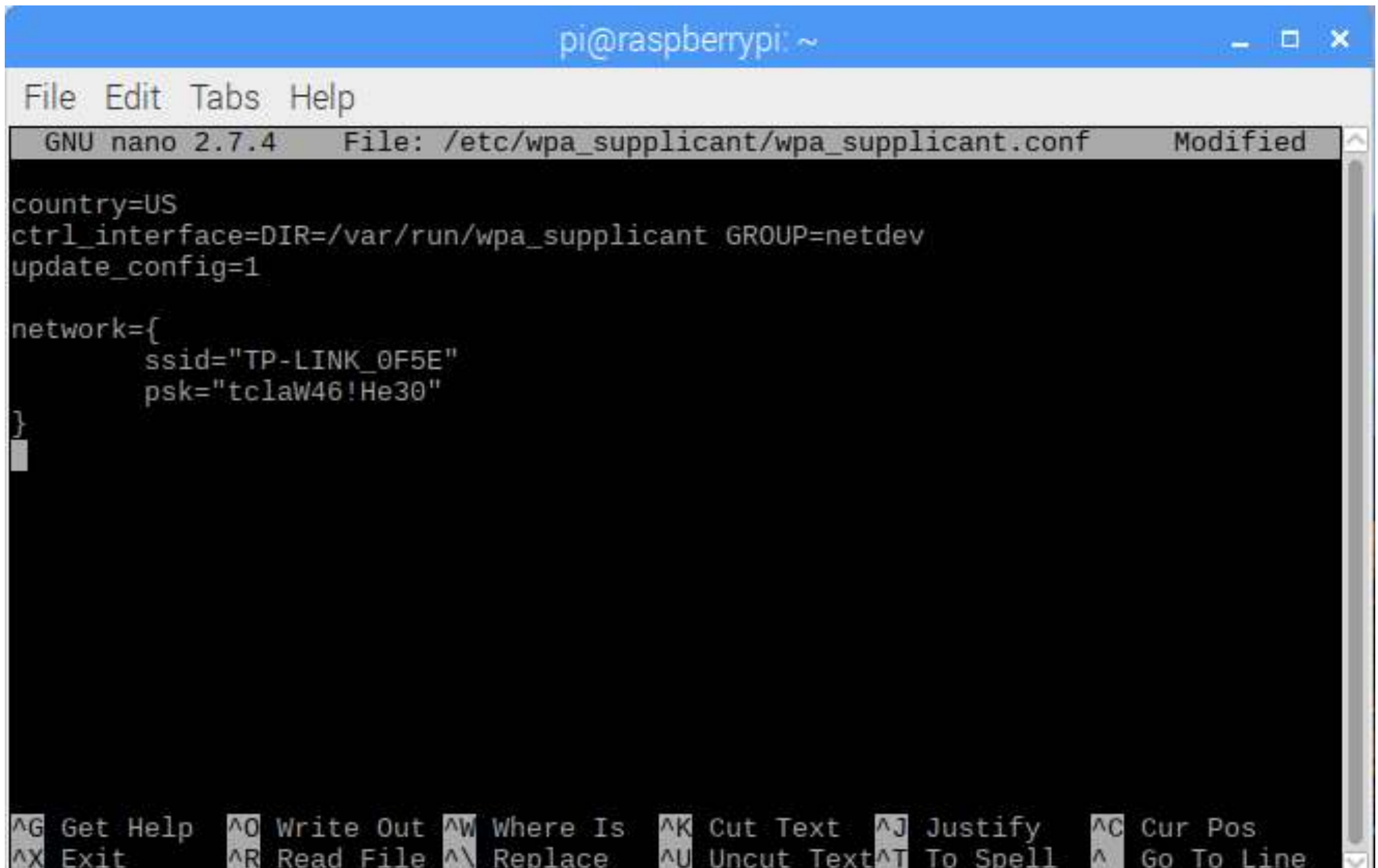
This will take some time to copy the image to your SD card. Assemble the rest of the RPi while you are waiting.

Insert the wifi adapter in one of the USB ports. Insert the keyboard and mouse dongle into one of the other USB ports. Connect a monitor to the Raspberry Pi with the HDMI cable.

When the Raspbian image has completed installing on the SD card, remove the SD card from the adapter and insert it into the SD card slot on

the underside of the Raspberry Pi. Then insert the power adapter into the micro USB port and power up the Raspberry Pi.

Step 4: Set Up WiFi Connection



The image shows a terminal window titled "pi@raspberrypi: ~" with a menu bar (File, Edit, Tabs, Help). The terminal is running GNU nano 2.7.4, editing the file "/etc/wpa_supplicant/wpa_supplicant.conf". The file content is as follows:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="TP-LINK_0F5E"
    psk="tclaw46!He30"
}
```

The cursor is at the end of the closing brace of the network block. The bottom of the terminal displays a list of nano editor shortcuts:

^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

Once the Raspberry Pi has finished booting up set up your wifi connection on the Raspberry PI by opening a terminal window and edit the `wpa_supplicant.conf` file by using the following command:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Change the country to your two letter country code.

Add your Wifi access point SSID and password at the bottom of the file:

```
network={  
  ssid="Your Wifi SSID"  
  psk="yourWifiPassword"  
}
```

One thing to note: The default keyboard setting is to use a GB configuration. For those of us in the US it places some of the special characters in different places, mainly the `@` and the `"` symbols are switched.

Save the file and exit nano.

Bring the Wifi connection up by typing:

```
ifup
```

or by rebooting the Raspberry PI with:

```
sudo reboot
```

Step 5: Configure Other Options With Raspi-config



When your Raspberry Pi has successfully connected to your Wifi you should see the Wifi symbol in the menu bar in the upper right corner of the screen for the Raspberry Pi.

You can now configure other options for your Pi. From a terminal window type:

```
sudo raspi-config
```

This will bring up the raspi-config interface and allow you to configure other options your Raspberry PI. You don't have to do this but there are a couple of things that you should do:

1. Change the default password. Don't leave devices on your network vulnerable by leaving the default passwords on your RPi for the pi and root users.
2. Set your location setting. This will give you the right keyboard settings, time settings and location for installing other software from the nearest repos. If you get the annoying PERL warnings about the locale cannot be set you can resolve it by using [these instructions](#).
3. Expand the file system to use the entire SD card. This will give you access to the entire storage space on the SD card.

Feel free to look at the other options that are available to you through this interface. You can do other things like overclocking your CPU, set up ssh and ftp connections, and change your boot settings to boot to a command line or the desktop.

Step 6: Configure the Network Bridge From Wifi to Ethernet

```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/sysctl.conf Modified

#####3
Functions previously found in netbase

Uncomment the next two lines to enable Spoof protection (reverse-path filter)
Turn on Source Address Verification in all interfaces to
prevent some spoofing attacks
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1

Uncomment the next line to enable TCP/IP SYN cookies
See http://lwn.net/Articles/277146/
Note: This may impact IPv6 TCP sessions too
net.ipv4.tcp_syncookies=1

Uncomment the next line to enable packet forwarding for IPv4
et.ipv4.ip_forward=1

Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
  
```

```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/network/interfaces

Interfaces(5) file used by ifup(8) and ifdown(8)

Please note that this file is written to be used with dhcpd.
For static IP, consult /etc/dhcpd.conf and 'man dhcpd.conf'.

Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

line-hotplug eth0
face eth0 inet static
address 192.168.2.1
netmask 255.255.255.0
network 192.168.2.0
broadcast 192.168.2.255

Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
  
```

```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/dnsmasq.conf

interfaceeth0
bind-interfaces
server=8.8.8.8
main-needed
log-dns
log-priv
dhcp-range=192.168.2.2,192.168.2.100,12h

Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
  
```

wlan0: Associated with TP-LINK_OF
 wlan0: Configured 192.168.0.108/24
 eth0: Configured 169.254.239.204/1

To do this we're going to use **dnsmasq** to set up the RPi to be a DHCP server and set up some custom DNS settings. This will allow the device connected to the RPi through ethernet to get an IP address from the RPi and also to for the RPi to pass DNS queries.

We'll also configure some **iptables** settings to make a NAT between the ethernet adapter and the Wifi connection.

First, install **dnsmasq**:


```
sudo apt-get install dnsmasq
```

Set your ethernet adapter to a static IP address

This will serve as a gateway for the device that you want to connect to the RPi ethernet port. Most Wifi routers use what's called a Private Network and set the IP range to something similar to:

```
192.168.1.1
```

For the ethernet adapter on your RPi you'll want to set that to an address that won't interfere with the routers ability to assign addresses, so we'll increment the subnet of the PRi to be:

```
192.168.2.1
```

Along with that you'll need to set up the the netmask to:

```
255.255.255.0
```

As well as the DHCP settings to broadcast what IP address are available:

```
network 192.168.2.0  
broadcast 192.168.2.255
```

Use iptables to configure a NAT setting to share the Wifi connection with the ethernet port

NAT stands for Network Address Translation. This allows a single IP address to server as a router on a network. So in this case the ethernet adapter on the RPi will serve as the router for whatever device you attach to it. The NAT settings will route the ethernet requests through the Wifi connection.

There are several commands to run here:

```
sudo iptables -F  
sudo iptables -t nat -F  
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE  
sudo iptables -A FORWARD -i wlan0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT  
sudo iptables -A FORWARD -i eth0 -o wlan0 -j ACCEPT
```

Configure the dnsmasq settings

The first thing to do is to turn on IP forwarding. This is done by putting a single number 1 in the `/proc/sys/net/ipv4/ip_forward` file:

```
sudo nano /proc/sys/net/ipv4/ip_forward
```

Put a 1 on the first line and then exit and save. *Hint: you also may have to edit `/etc/sysctl.conf` and uncomment this line:*

```
net.ipv4.ip_forward=1
```

Next set up ip routing:

```
sudo ip route del 0/0 dev eth0 &> /dev/null
a=`route | awk "/${wlan}/" '{print $5+1;exit}'`
sudo route add -net default gw 192.168.2.1 netmask 0.0.0.0 dev eth0 metric $a
```

The last thing to do is edit your `/etc/dnsmasq.conf` file and include these settings;

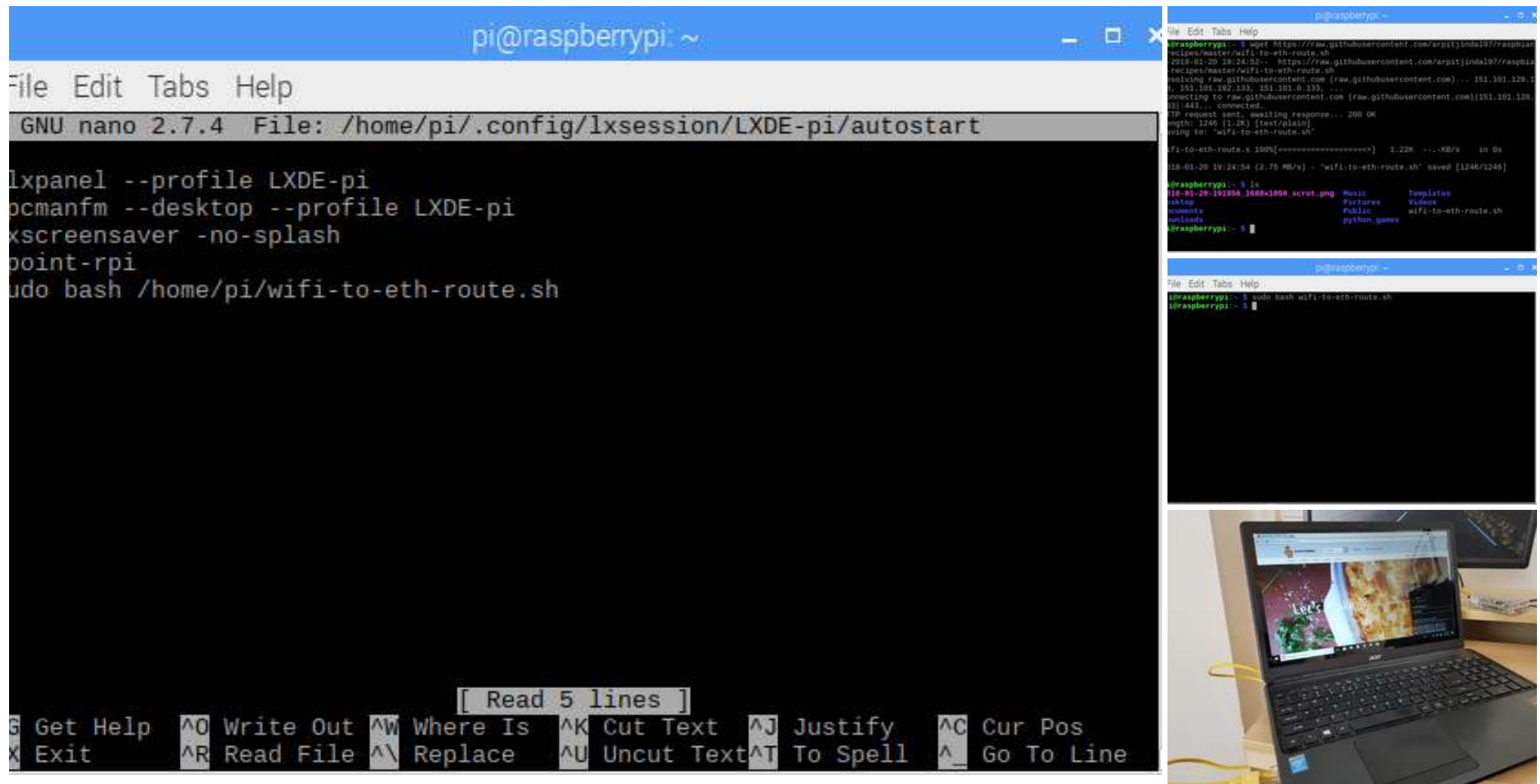
```
interface=eth0
bind-interfaces
server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=192.168.2.2,192.168.2.100,12h
```

Then run this command to start your dnsmasq services:

```
sudo systemctl start dnsmasq
```

Now plug a CAT5 network cable into the device you want to include on the network and put the other end of the cable into the ethernet port on the RPi and you should be good to go! When we set up the ethernet interface we made it hot pluggable, so you should see the ethernet interface come up when you plug the device into the RPi.

Step 7: Automate All Those Commands With a Script



This was a lot of work to get the network bridge up and running. You'll probably want this to run automatically every time your RPi boots up, so to do that we're going to need a script to run all of these commands for us. Luckily [Arpit Agarwal](#) has already created a script and is available for [download here](#).

Don't worry about typing all those commands above and run this command from your home directory to download the script file:

<https://raw.githubusercontent.com/arpitjindal97/raspbian-recipes/master/wifi-to-eth-route.sh>

To get this file to run every time you boot your RPi you'll need to add a directive to your session autostart file:

```
nano /home/pi/.config/lxsession/LXDE-pi/autostart
```

and add this to the bottom of the file:

```
sudo bash /home/pi/wifi-to-eth-route.sh
```

Then just reboot the RPi and the script does all the work for you. You can also run this set up any time you want by running this command from a terminal:

```
sudo bash /home/pi/wifi-to-eth-route.sh
```