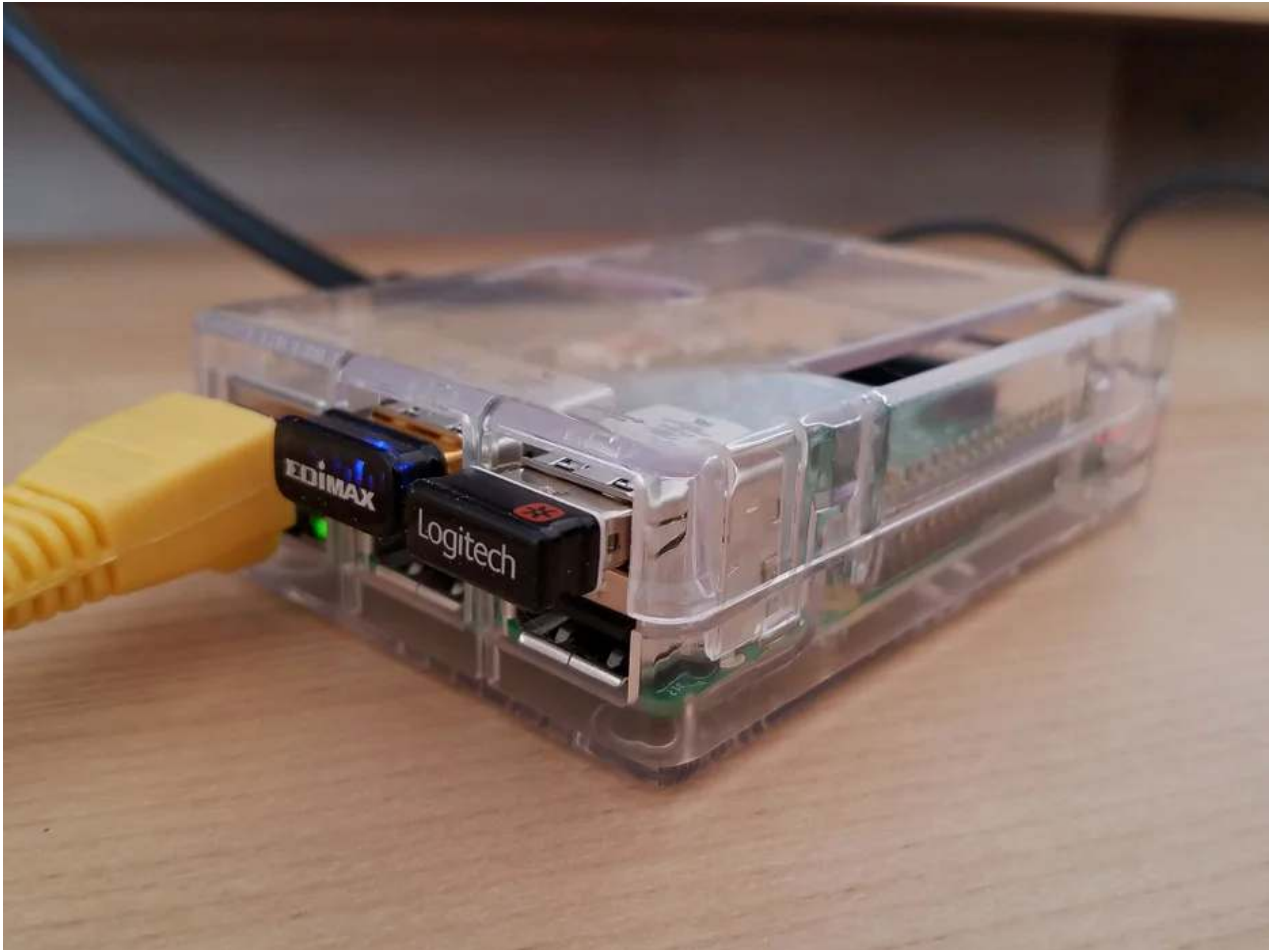
AUTODESK
Instructables

Share WiFi With Ethernet Port on a Raspberry Pi

By WaltW in CircuitsRaspberry Pi

Introduction: Share WiFi With Ethernet Port on a Raspberry Pi

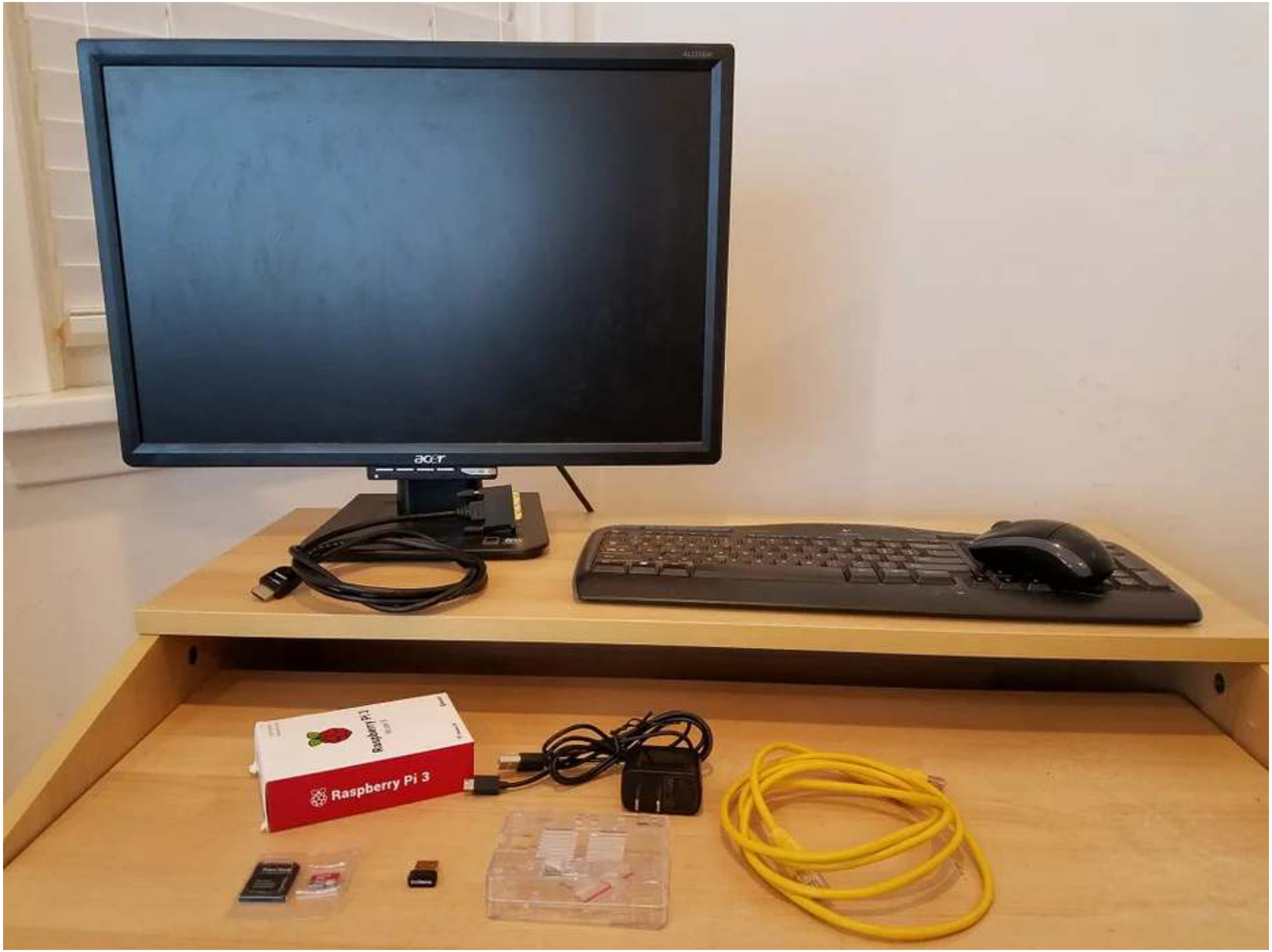

EDiMAX
Logitech

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.

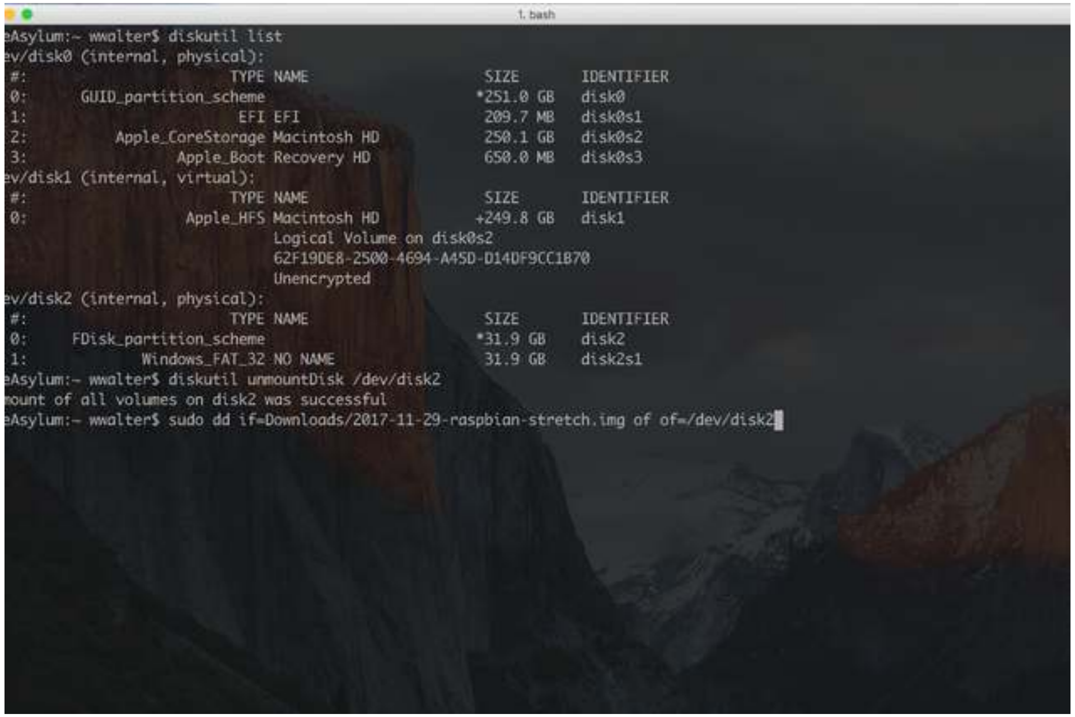
This easy DIY solution will give you the capability of adding a wifi connection to your older devices without breaking the bank by purchasing wireless print adapter.

Step 1: Items You'll Need


acer
Raspberry Pi 3
Raspberry Pi 3

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

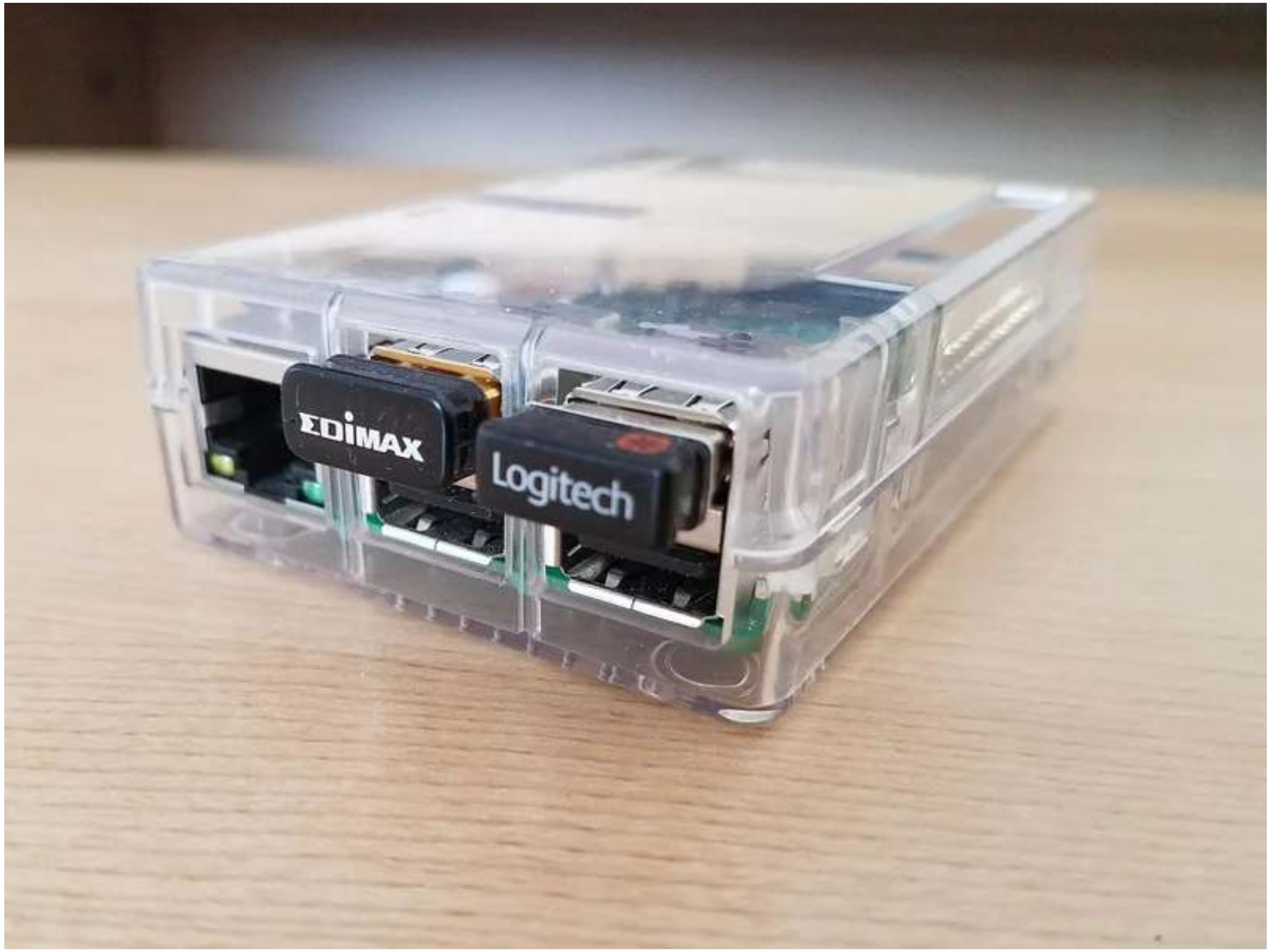


```
eAsylum:~ wwalter$ diskutil list
ev/disk0 (internal, physical):
 #:                       TYPE NAME                    SIZE       IDENTIFIER
 0:      GUID_partition_scheme                        *251.0 GB   disk0
 1:                        EFI EFI                     209.7 MB   disk0s1
 2:          Apple_CoreStorage Macintosh HD            250.1 GB   disk0s2
 3:                 Apple_Boot Recovery HD             650.0 MB   disk0s3
ev/disk1 (internal, virtual):
 #:                       TYPE NAME                    SIZE       IDENTIFIER
 0:                  Apple_HFS Macintosh HD           +249.8 GB   disk1
                               Logical Volume on disk0s2
                               62F19DE8-2500-4694-A45D-D14DF9CC1870
                               Unencrypted
ev/disk2 (internal, physical):
 #:                       TYPE NAME                    SIZE       IDENTIFIER
 0:     FDisk_partition_scheme                        *31.9 GB    disk2
 1:             Windows_FAT_32 NO NAME                 31.9 GB    disk2s1
eAsylum:~ wwalter$ diskutil unmountDisk /dev/disk2
mount of all volumes on disk2 was successful
eAsylum:~ wwalter$ sudo dd if=Downloads/2017-11-29-raspbian-stretch.img of of=/dev/disk2
```

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 Raspian operating system image onto the SD card by using these instructions:

- WIndows instructions
- Mac OSX instructions
- Linux instructions

Step 3: Assemble the Remaining Components

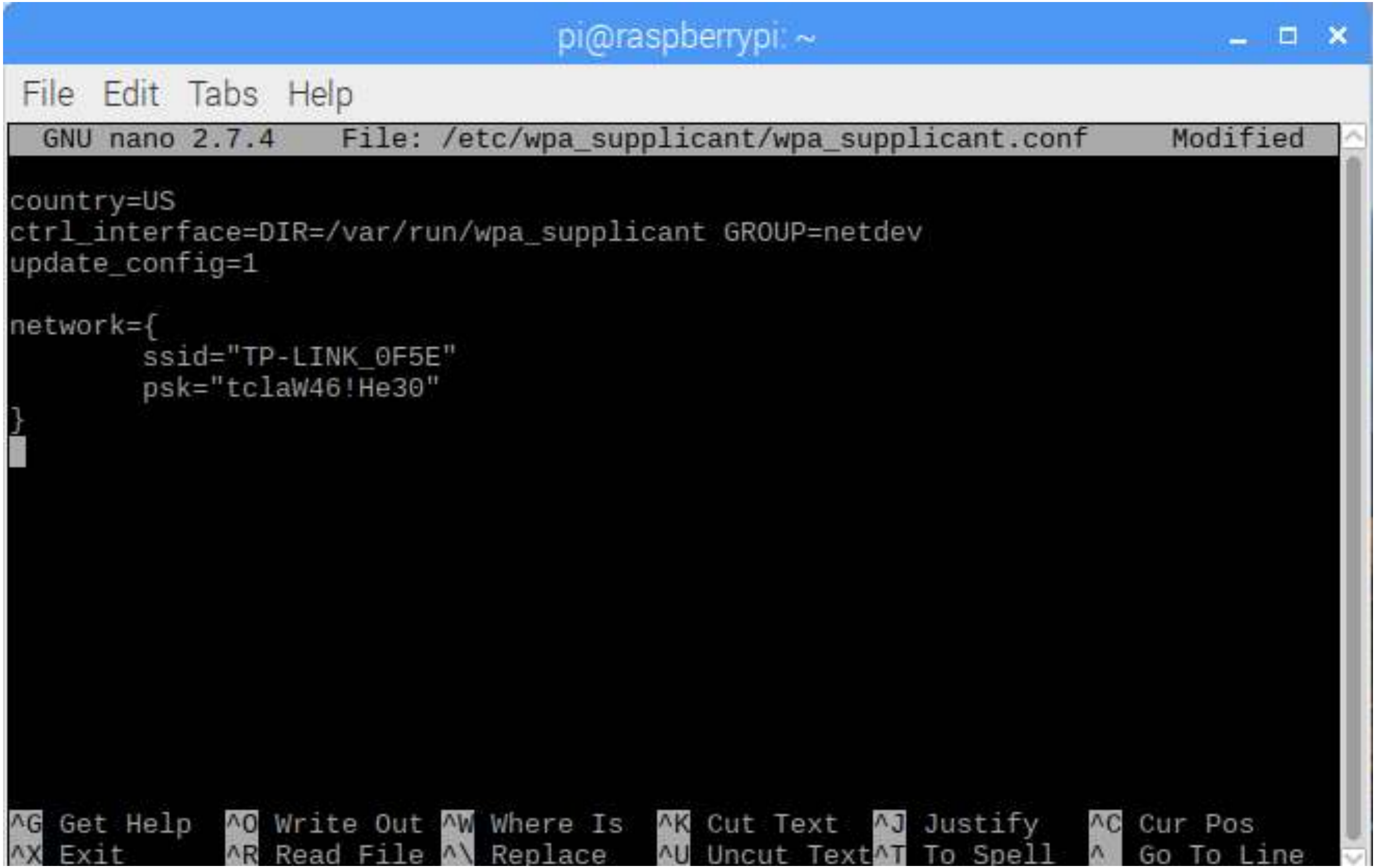


This will take some time to copy the image to your SD card. Assemble the rest of the RPi while you are waiting.

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



```
  GNU nano 2.7.4    File: /etc/wpa_supplicant/wpa_supplicant.conf     Modified

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

network={
        ssid="TP-LINK_0F5E"
        psk="tclaW46!He30"
}


^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.

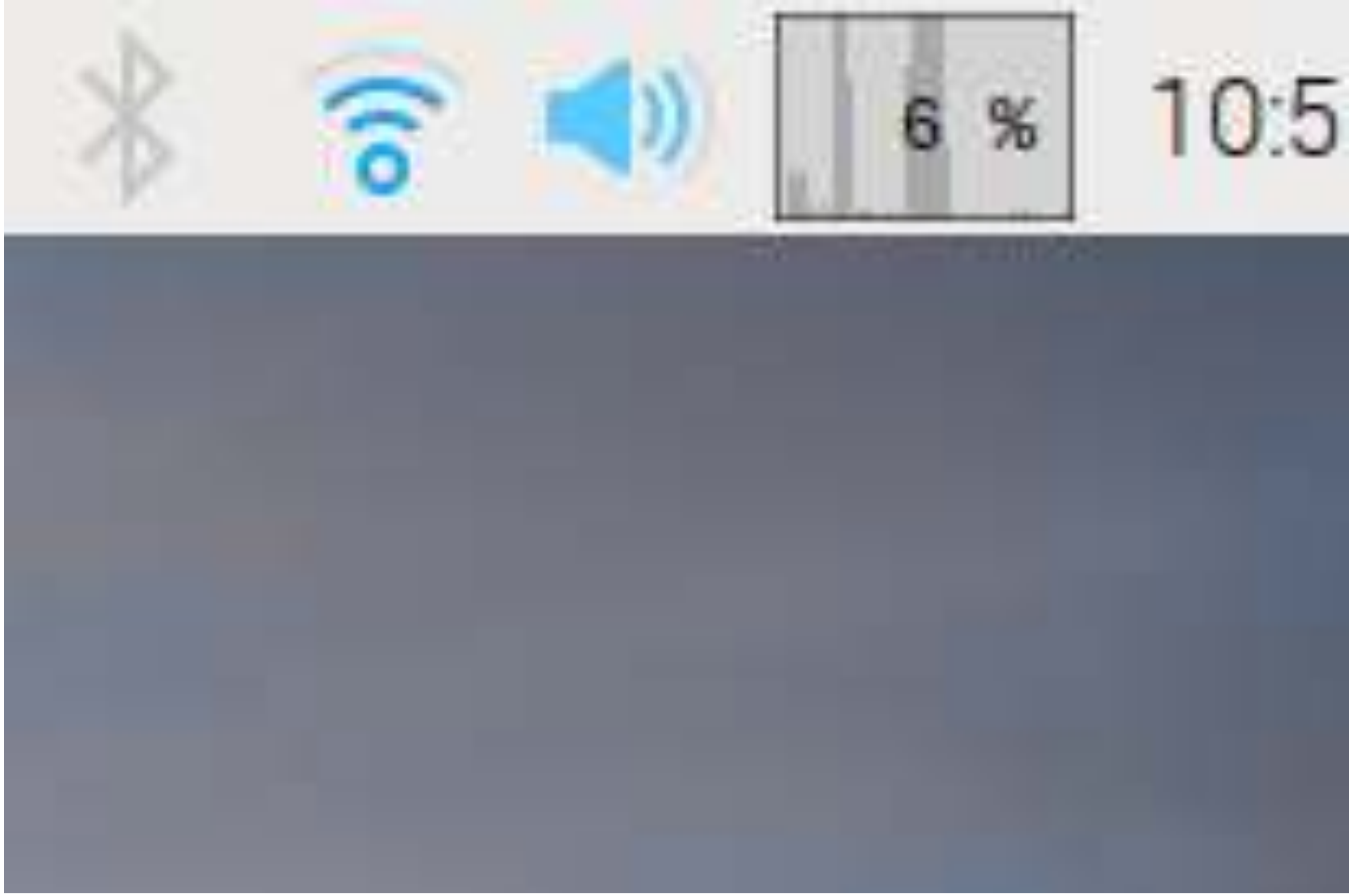
Bring the WIfi connection up by typing:

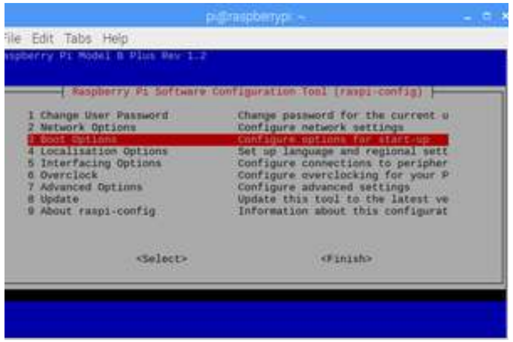
```
ifup
```

or by rebooting the Raspberry PI with:

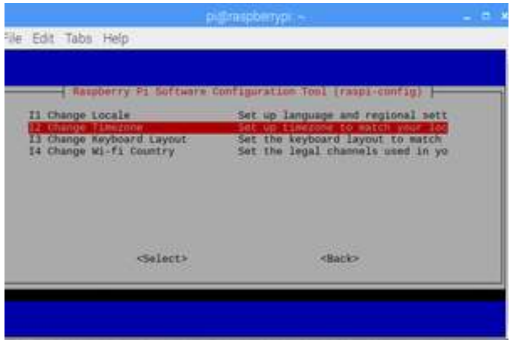
```
sudo reboot
```

Step 5: Configure Other Options With Raspi-config

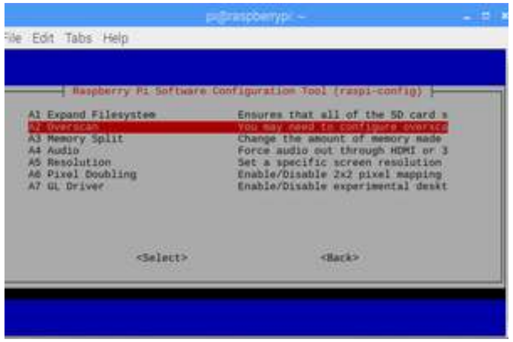




```
spberry Pi Model B Plus Rev 1.2

─────┤ Raspberry Pi Software Configuration Tool (raspi-config) ├─────

  1 Change User Password        Change password for the current u
  2 Network Options             Configure network settings
  3 Boot Options                Configure options for start-up
  4 Localisation Options        Set up language and regional sett
  5 Interfacing Options         Configure connections to peripher
  6 Overclock                   Configure overclocking for your P
  7 Advanced Options            Configure advanced settings
  8 Update                      Update this tool to the latest ve
  9 About raspi-config          Information about this configurat

             <Select>                  <Finish>
```



```
─────┤ Raspberry Pi Software Configuration Tool (raspi-config) ├─────

  I1 Change Locale              Set up language and regional sett
  I2 Change Timezone            Set up timezone to match your loc
  I3 Change Keyboard Layout     Set the keyboard layout to match
  I4 Change Wi-fi Country       Set the legal channels used in yo

             <Select>                  <Back>
```



```
─────┤ Raspberry Pi Software Configuration Tool (raspi-config) ├─────

  A1 Expand Filesystem          Ensures that all of the SD card s
  A2 Overscan                   You may need to configure oversca
  A3 Memory Split               Change the amount of memory made
  A4 Audio                      Force audio out through HDMI or 3
  A5 Resolution                 Set a specific screen resolution
  A6 Pixel Doubling             Enable/Disable 2x2 pixel mapping
  A7 GL Driver                  Enable/Disable experimental deskt

             <Select>                  <Back>
```

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 termnial window type:

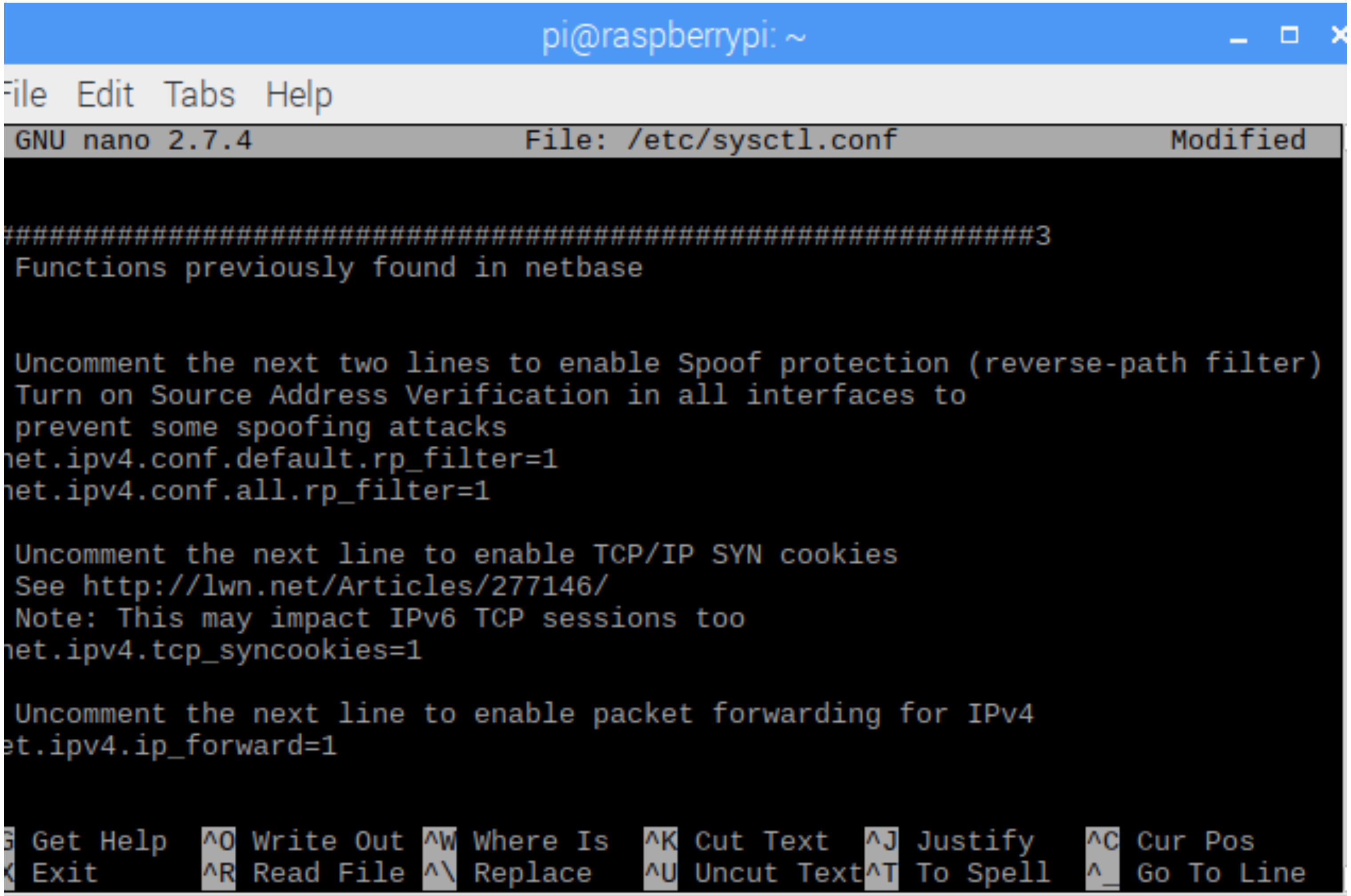
sudo raspi-config

This will bring up the raspi-config interface an 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:

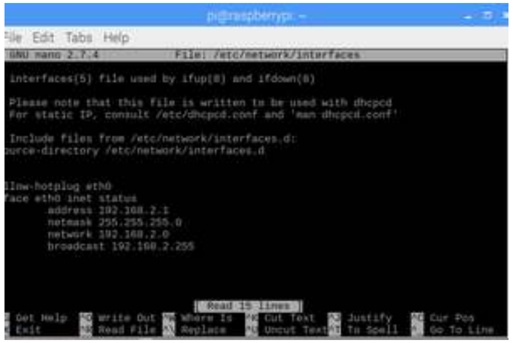
1. Change the default password. Don't leave devices on your network vulnerable by leaving the default passwords on your RPi for the pi and root users.
2. Set your location setting. This will give you the right keyboard settings, time settings and location for installing other software from the nearest 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 HD 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.

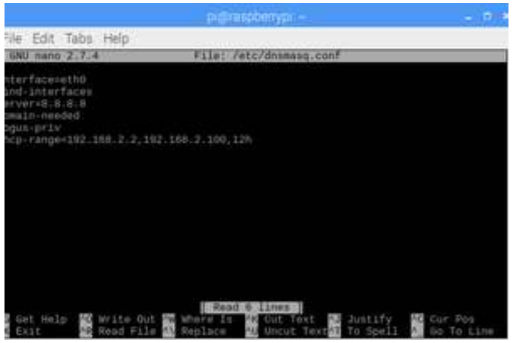
Step 6: Configure the Network Bridge From Wifi to Ethernet



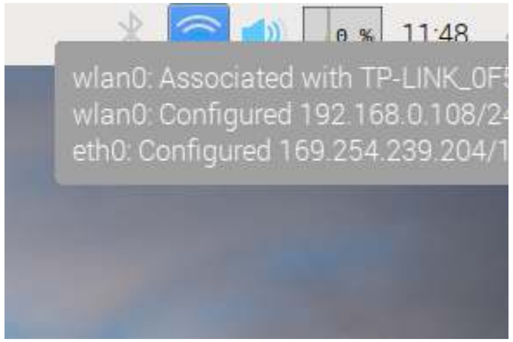
```
 GNU nano 2.7.4              File: /etc/sysctl.conf                     Modified


##############################################################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


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
```



```
 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 dhcpcd
 For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

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


llow-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



                          [ Read 15 lines ]
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
```



```
 GNU nano 2.7.4              File: /etc/dnsmasq.conf

nterface=eth0
ind-interfaces
erver=8.8.8.8
omain-needed
ogus-priv
hcp-range=192.168.2.2,192.168.2.100,12h



                          [ Read 6 lines ]
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
```



wlan0: Associated with TP-LINK_0F
wlan0: Configured 192.168.0.108/2
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 DCHP 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;

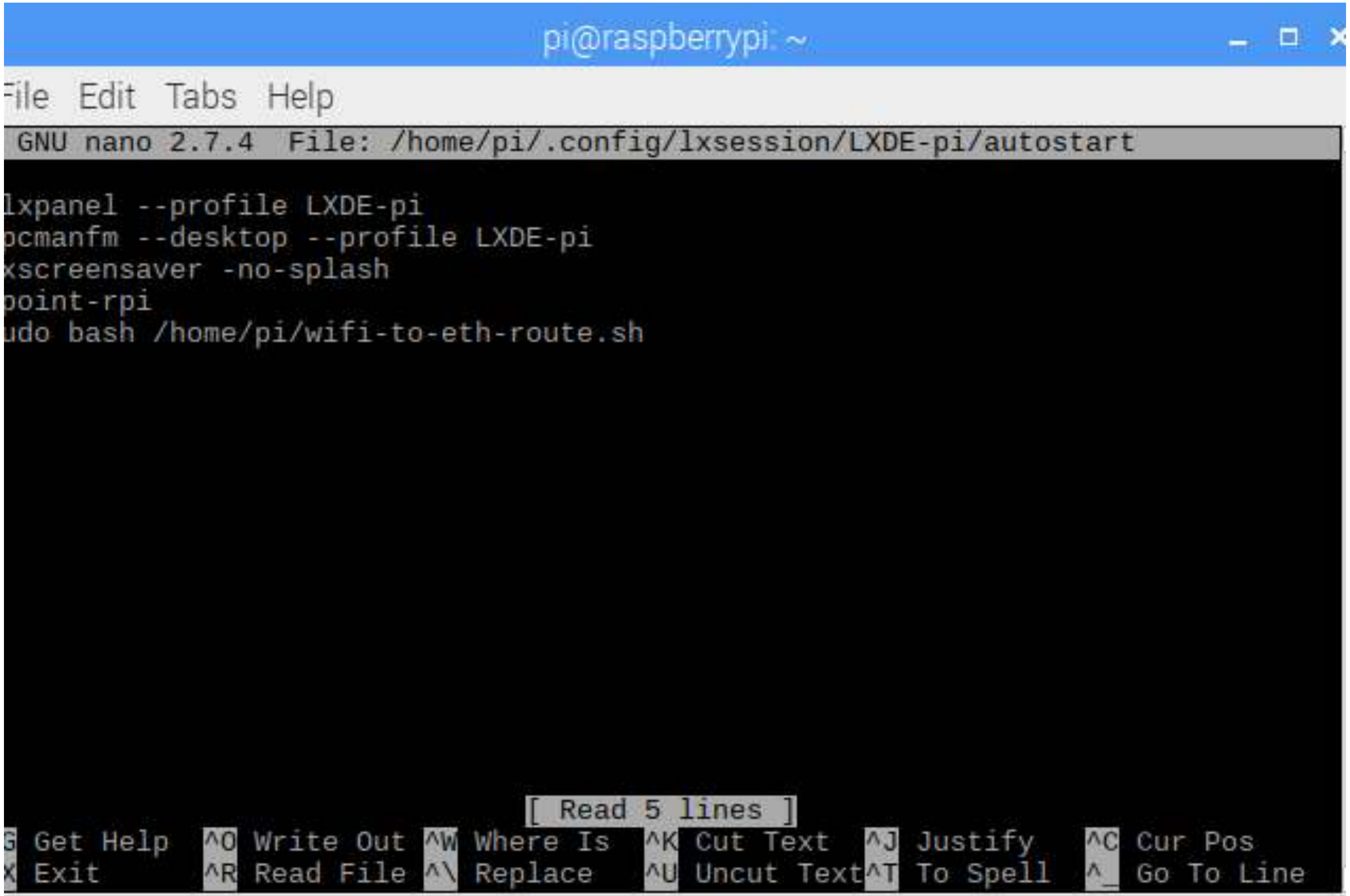
```
interface=eth0
bind-interfaces
server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=192.168.2.2,192.168.2.100,12h
```

Then run this command to start your dnsmasq services:

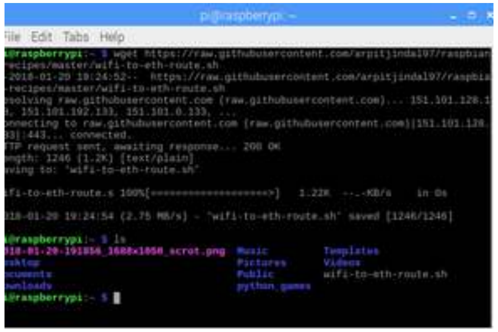
```
sudo systemctl start dnsmasq
```

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.

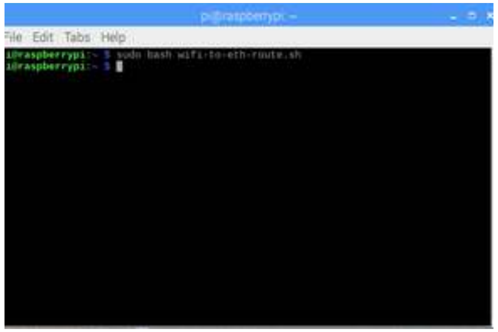
Step 7: Automate All Those Commands With a Script



```
 GNU nano 2.7.4  File: /home/pi/.config/lxsession/LXDE-pi/autostart

lxpanel --profile LXDE-pi
pcmanfm --desktop --profile LXDE-pi
xscreensaver -no-splash
point-rpi
udo bash /home/pi/wifi-to-eth-route.sh




                          [ Read 5 lines ]
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
```



```
i@raspberrypi:~ $ wget https://raw.githubusercontent.com/arpitjindal97/raspbian
recipes/master/wifi-to-eth-route.sh
-2018-01-20 19:24:52--  https://raw.githubusercontent.com/arpitjindal97/raspbia
-recipes/master/wifi-to-eth-route.sh
esolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.128.1
3, 151.101.192.133, 151.101.0.133, ...
onnecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.128.
33|:443... connected.
TTP request sent, awaiting response... 200 OK
ength: 1246 (1.2K) [text/plain]
aving to: 'wifi-to-eth-route.sh'

ifi-to-eth-route.s 100%[===================>]   1.22K  --.-KB/s    in 0s

018-01-20 19:24:54 (2.75 MB/s) - 'wifi-to-eth-route.sh' saved [1246/1246]

i@raspberrypi:~ $ ls
018-01-20-191856_1680x1050_scrot.png  Music         Templates
esktop                                Pictures      Videos
ocuments                              Public        wifi-to-eth-route.sh
ownloads                              python_games
i@raspberrypi:~ $ 
```



```
i@raspberrypi:~ $ sudo bash wifi-to-eth-route.sh
i@raspberrypi:~ $ 
```

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
```