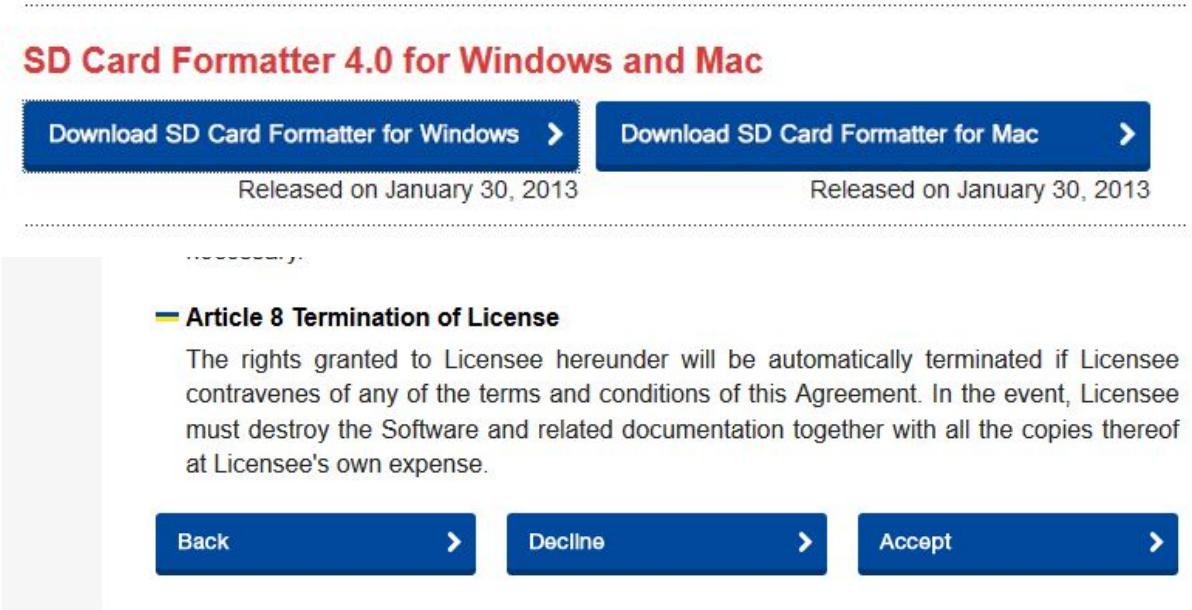
Installation Guide for Debian machinekit ISO image on the Beaglebone Black - Windows

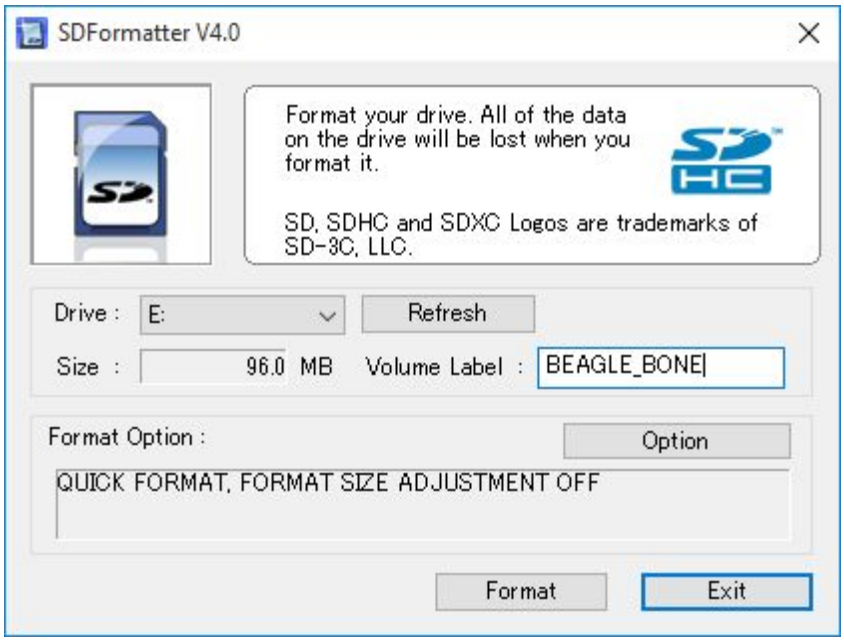
This guide intends to explain step-by-step how to write an ISO image to a SD card to use it on a Beaglebone Black board (BBB). The Operating System used to carry out this task is Windows 10. The official information is posted on the web page of the project: http://www.openvlc.org/openvlc.html. Before you start, make sure you have a stable Internet connection because you will need to download some software to complete the steps.

Step-by-step Installation

1) *Download the Debian machinekit image*
   To download the image follow this link:
   https://drive.google.com/file/d/0BwGT2J3dvAfNOEVibS1KQ2d5MGc/view.
   When the download is complete modify the name of the file. Change it from **.img.xz** to **.img**.
2) *Format the SD card*
   https://www.sdcard.org/downloads/formatter_4/

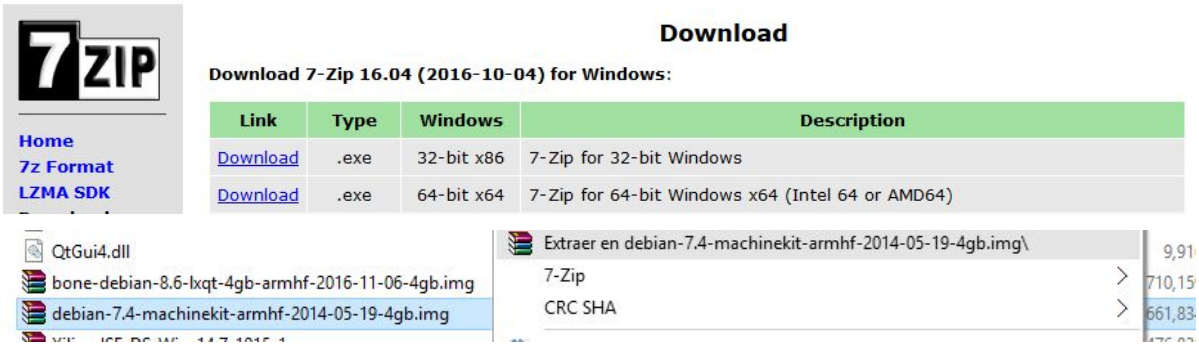


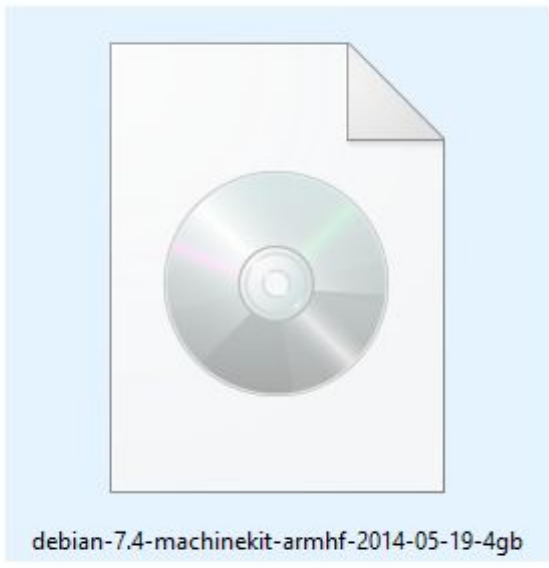
When you click accept the download should start. Extract the files from the .zip folder and you will find the .exe file. Run it and the installation window will prompt in order to install SD Card Formatter.



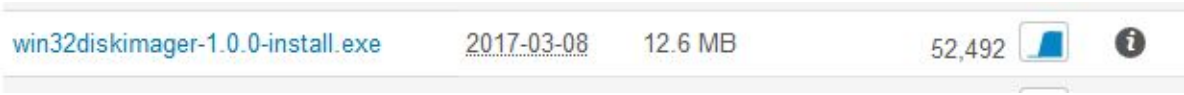
When you finish the installation, run the program and you will open a window similar to this. Just click format and after a few seconds your SD card should be ready.

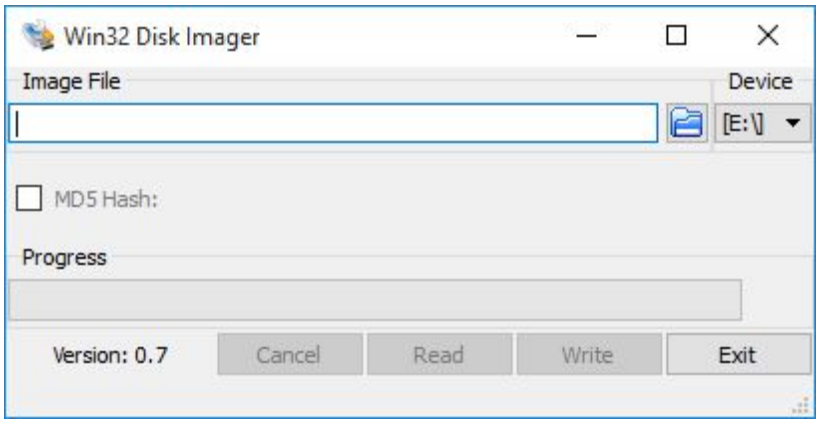
3) *Download 7 zip to extract the ISO file*
http://7-zip.org/download.html



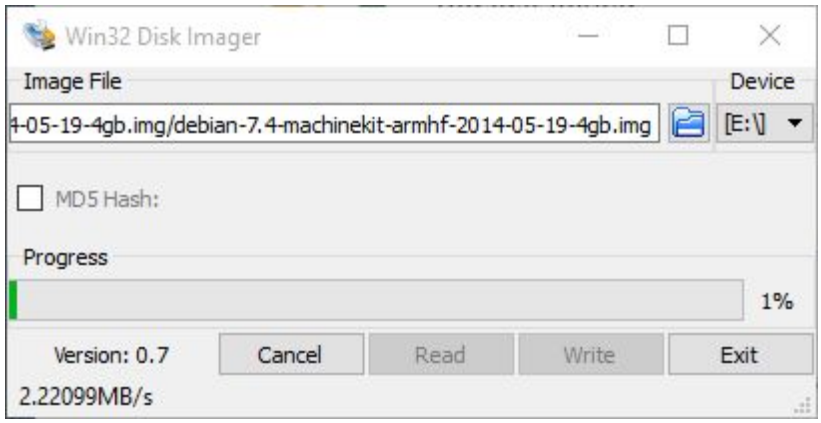


4) *Download Win32 Disk Imager to write the image into the SD card*
https://sourceforge.net/projects/win32diskimager/files/Archive/





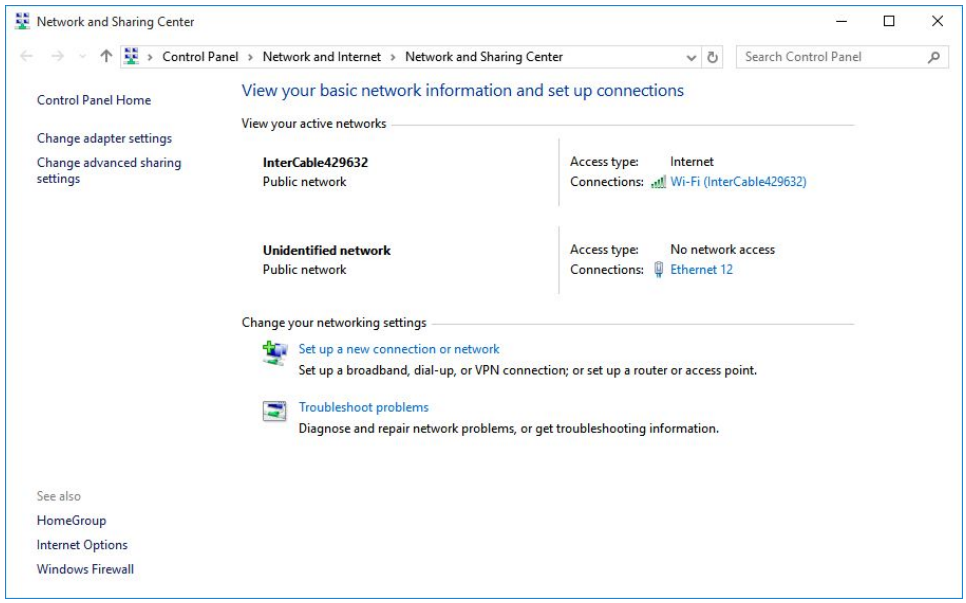
Select the SD card in the field for the Device and then click the blue folder to the left of the name of the device to search for your .img file. When you select it, the **Write** option will be enabled. Then the process will start.

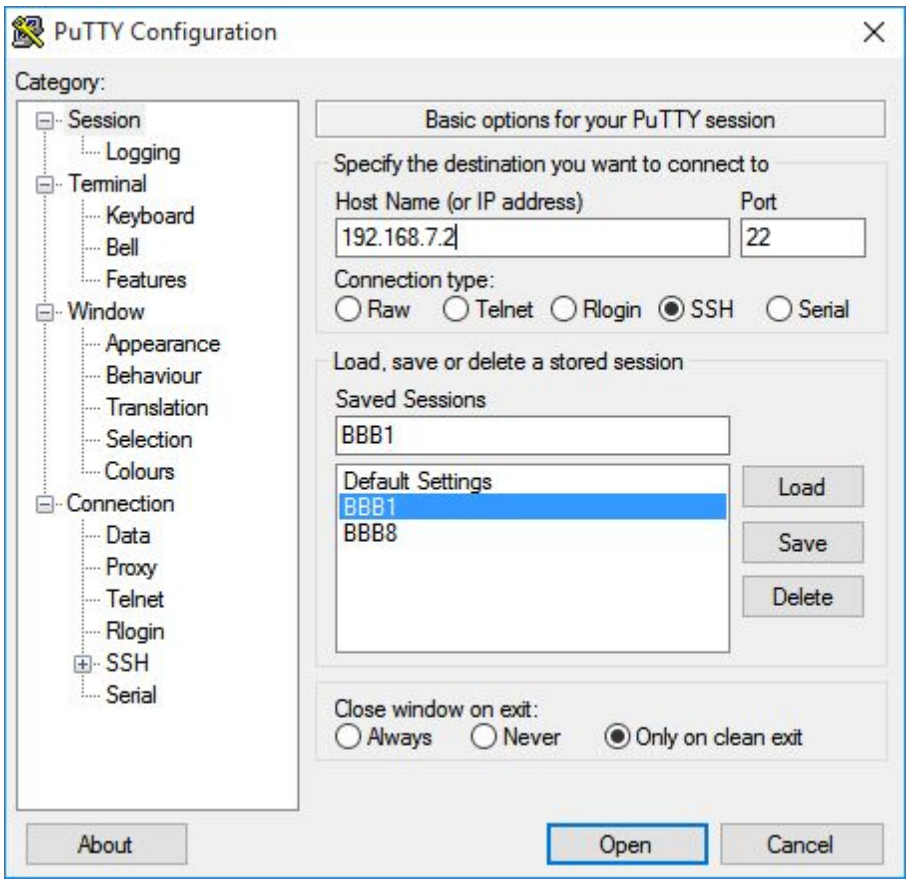


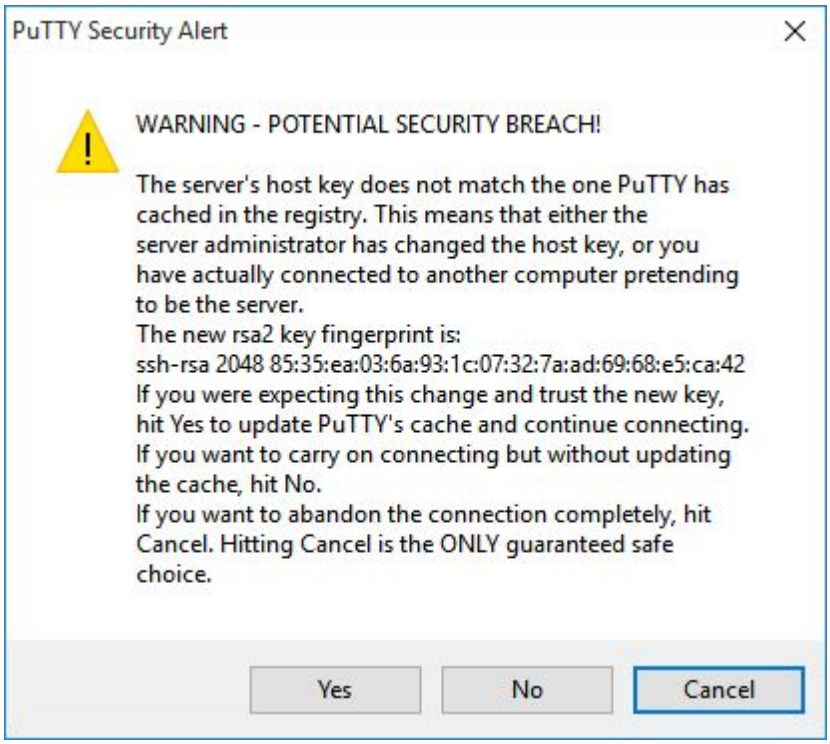
5) *Install the drivers for the BBB*

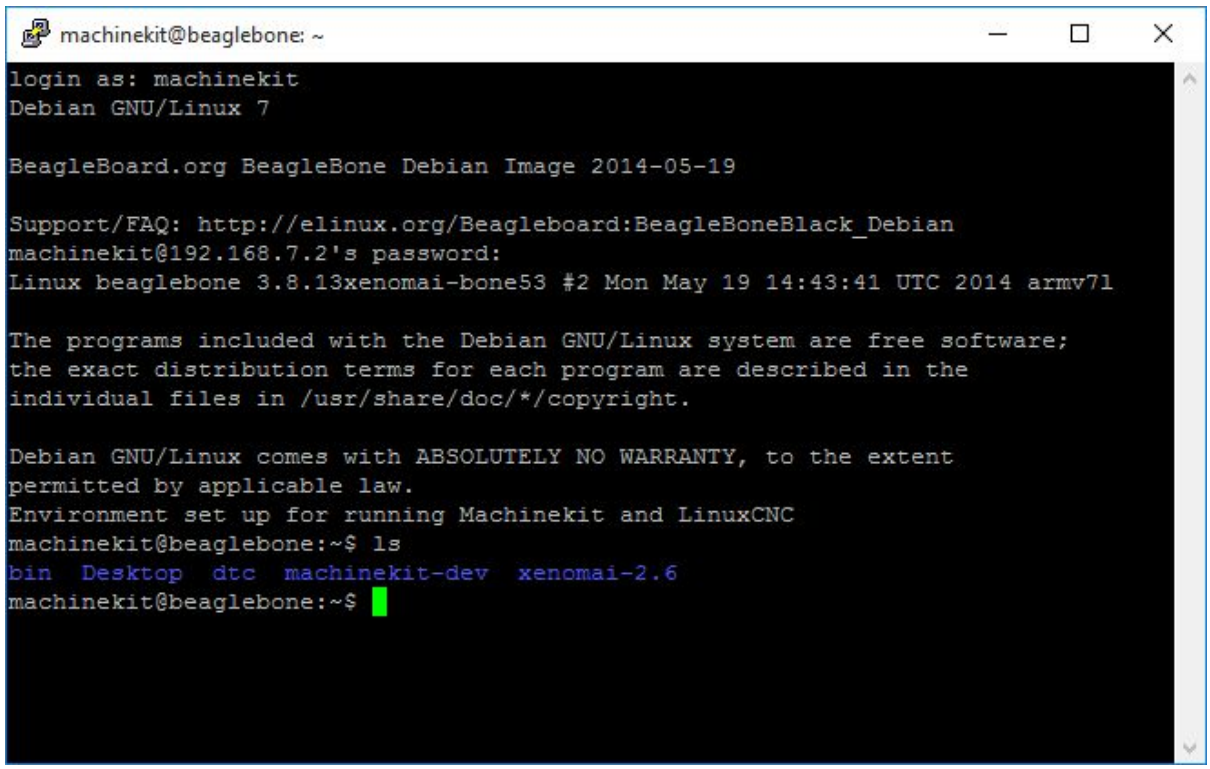
https://learn.adafruit.com/ssh-to-beaglebone-black-over-usb/installing-drivers-windows

6) *Boot up the OS on the BBB*
When the process of writing the image into the SD card is finished, plug it in into the Beaglebone Black and connect it via USB to your computer. For this step you will need to download Putty (http://www.putty.org/) in order to SSH the BBB. The default IP address for the BBB is 192.168.7.2, so you will need to type it when you open Putty. Open the **Network and Sharing Center** to check out if the BBB is being detected as a **Unidentified network**.

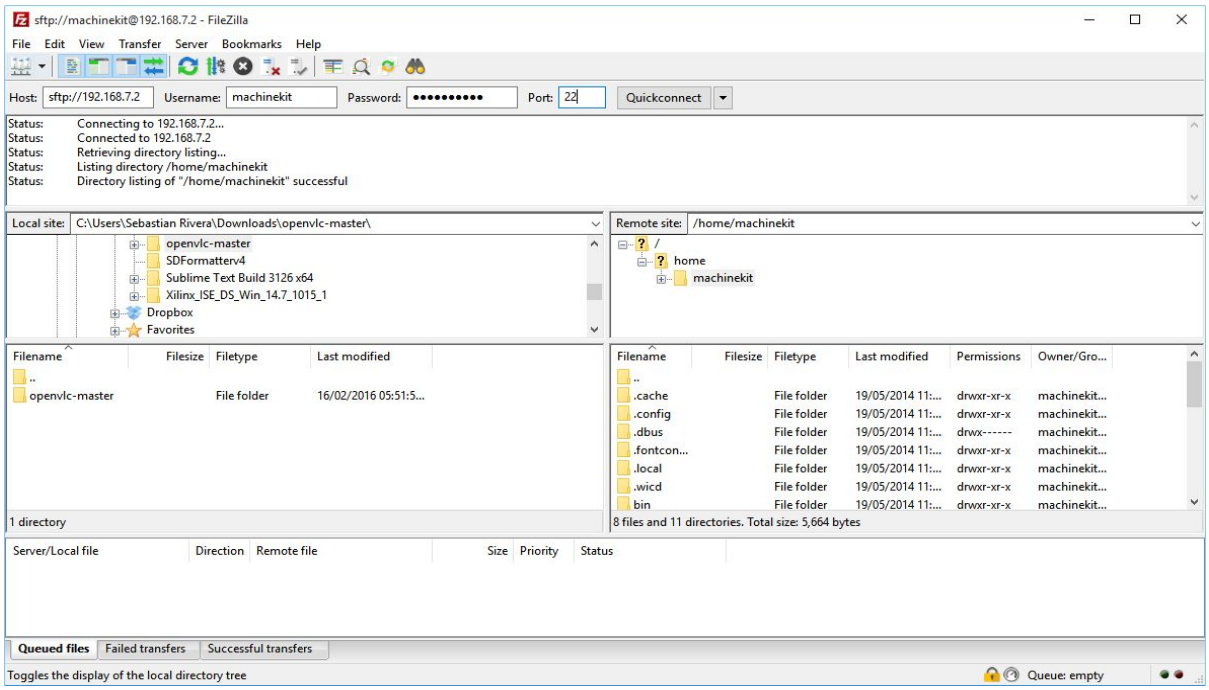

Network and Sharing Center
Control Panel › Network and Internet › Network and Sharing Center
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
View your basic network information and set up connections
View your active networks
InterCable429632
Public network
Access type: Internet
Connections: Wi-Fi (InterCable429632)
Unidentified network
Public network
Access type: No network access
Connections: Ethernet 12
Change your networking settings
Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.
Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.
See also
HomeGroup
Internet Options
Windows Firewall


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
192.168.7.2
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
BBB1
Default Settings
BBB1
BBB8
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel

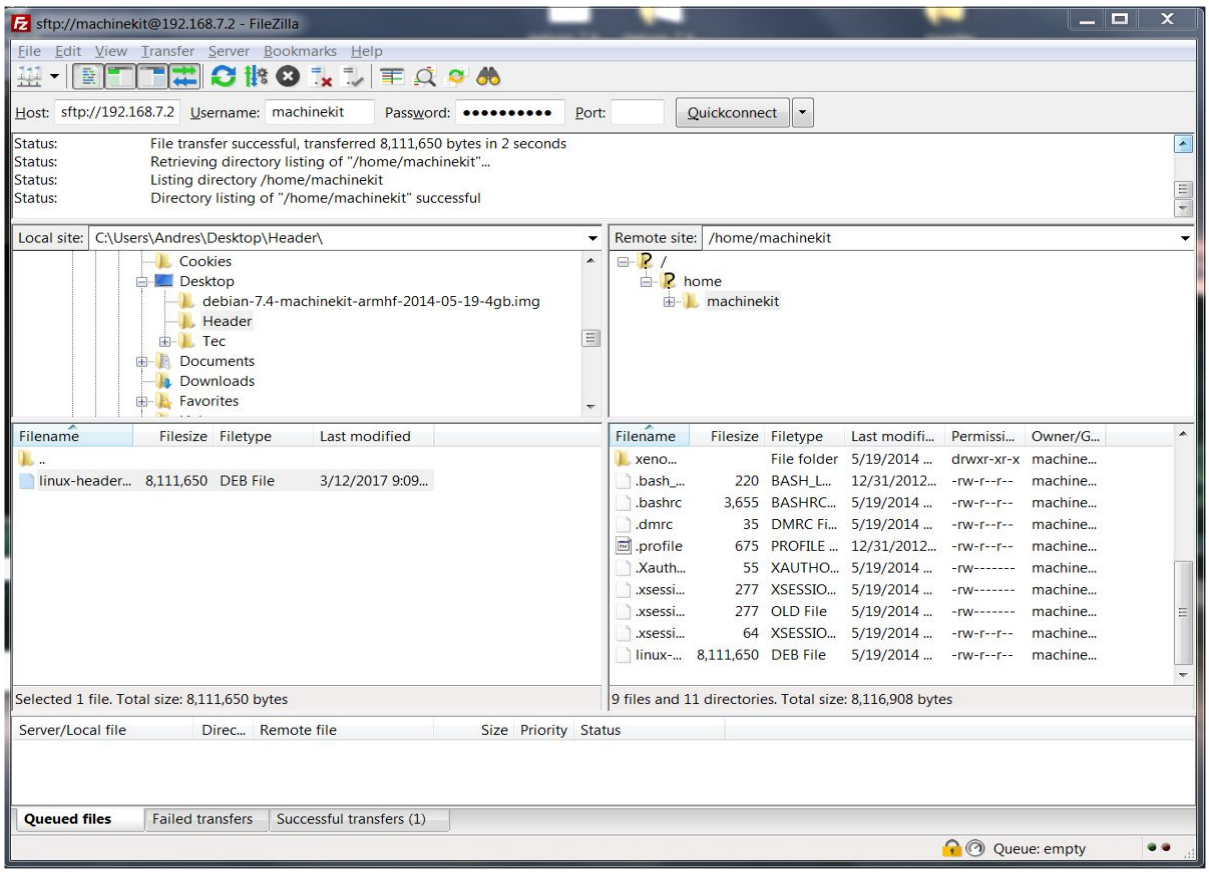




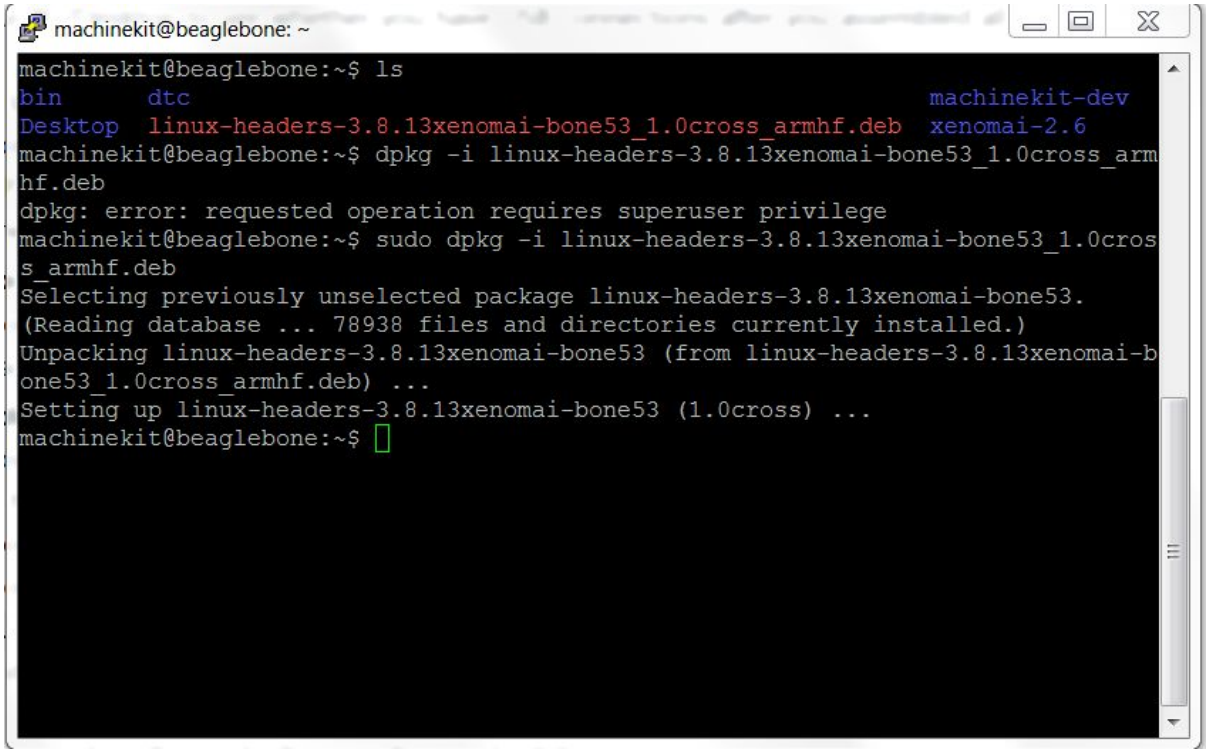
7) *Download FileZilla Client to exchange files between your computer and the BBB*
https://filezilla-project.org/download.php



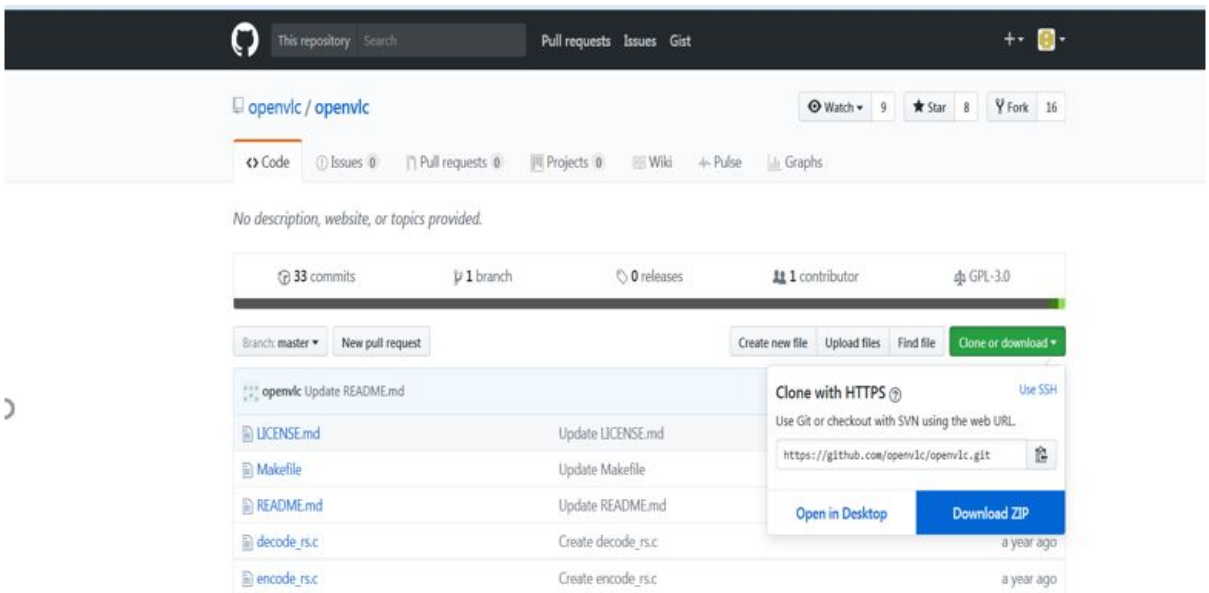
8) Search the linux headers in the *local site,* double click when you find the header and wait until the copy is done. The header must appear in the right box as in the following image.



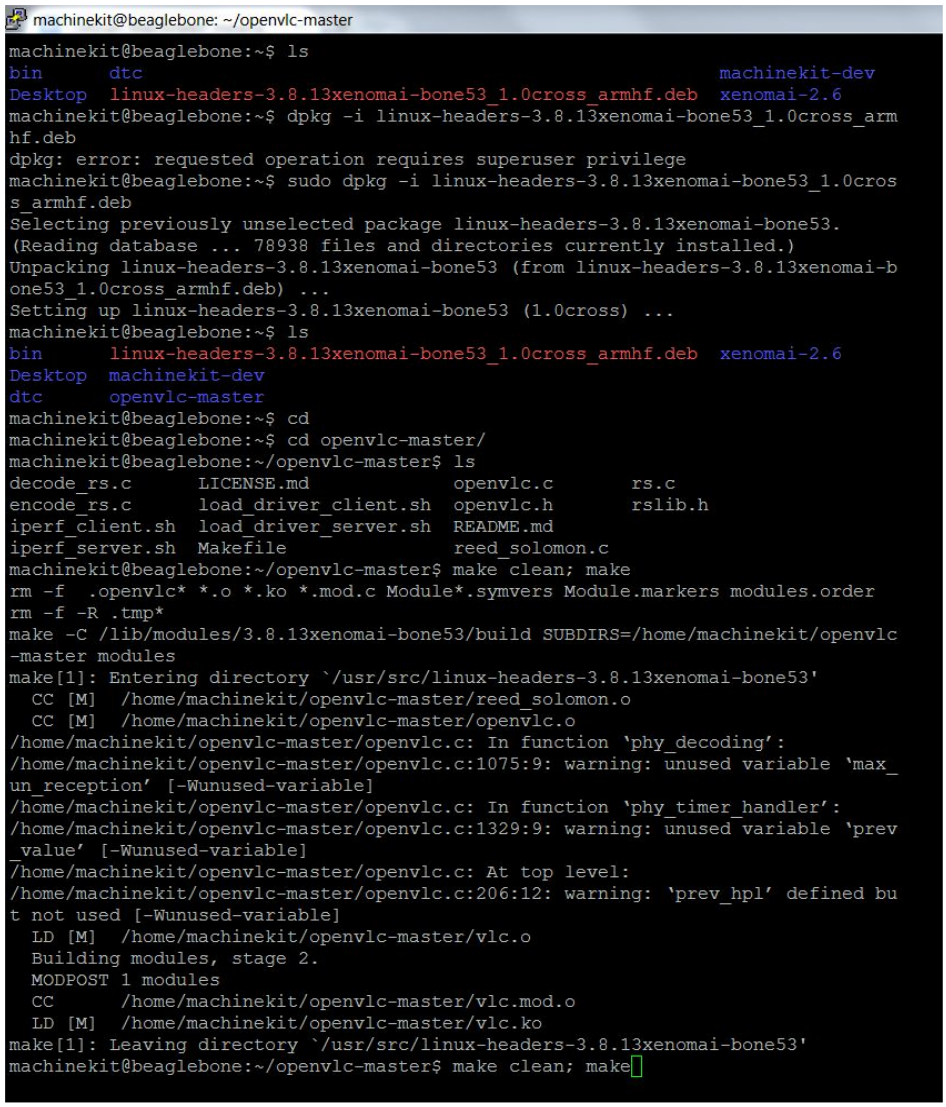
9) Go back to the machinekit@beaglebone, and write the following: *sudo dpkg -i linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb. Once you finish, press enter and wait until the machinekit@beaglebone display the same message as in the following image.*



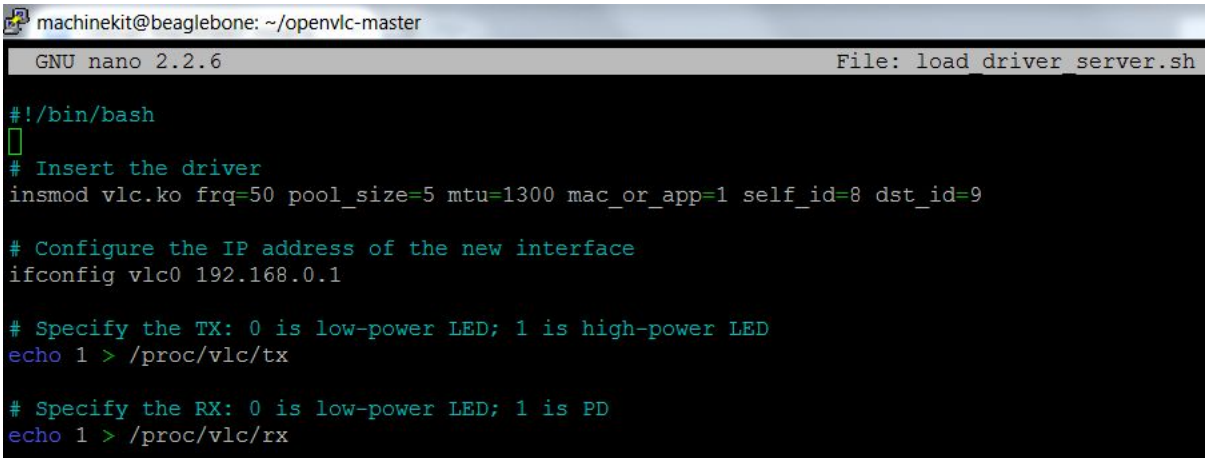
10) Go to the following link https://github.com/openvlc/openvlc and right click on the *clone or download* and search for the option of *Download ZIP.*



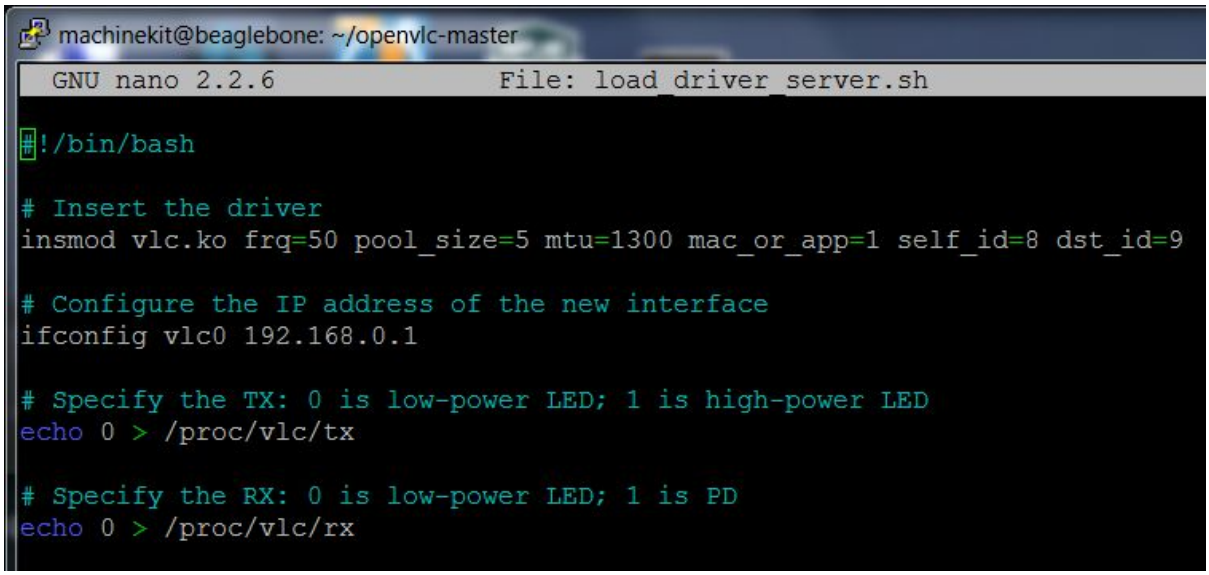
11) Go back to Filezilla and find the *openvlc-master* and do the same steps as when you transfer the headers (8). After that go back to the machinekit@beaglebone and go to the *openvlc-master* file, once you are there write: *make clean; make* wait until machinekit@beaglebone displays the same message as the following image.



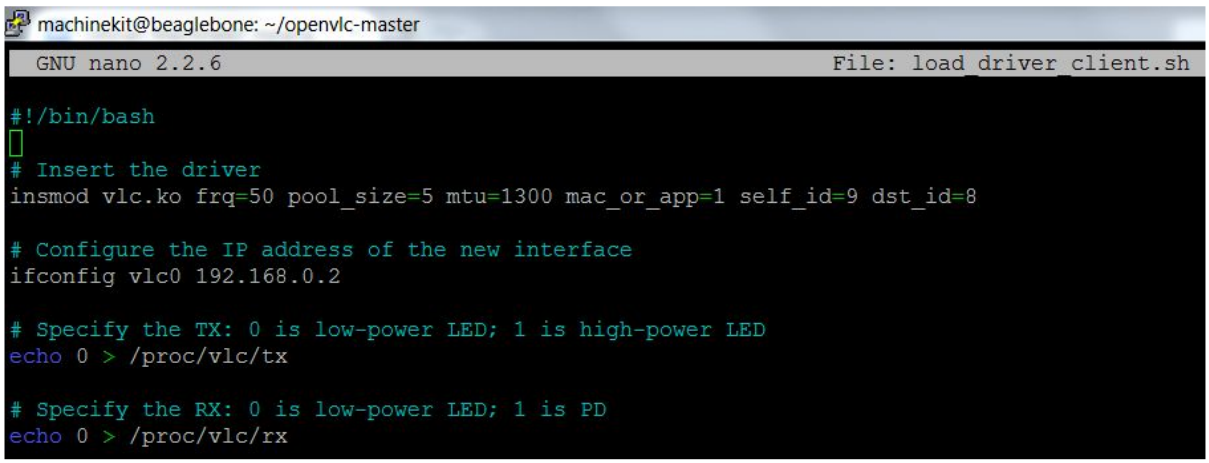
12) A vlc.ko file must be generated, now we have to focus on 2 files the *load_driver_server.sh* and the *load_driver_client.sh.* Write the instruction: nano load_driver_server.sh, a chart must appear like the following image.



Notice that next to the word echo is the number 1, change both echo number 1 into 0. Then press *control* + *x* and *y* to save the changes. The result must look like the following image.



13) Do the same steps as in 12 for *load_driver_client.sh.* The machinekit@beaglebone must look like the following image.



14) Write the instruction *ls -l* , you will see that the *load_driver_server.sh* and the *load_driver_client.sh* cannot be executable. In order to make this possible write the following instruction: *chmod u+x load_driver_server.sh* press enter and then *chmod u+x load_driver_client.sh.* Check now if both .sh are executable.

```
machinekit@beaglebone:~/openvlc-master$ ls -l
total 1060
-rw-r--r-- 1 machinekit machinekit   6959 May 19 16:33 decode_rs.c
-rw-r--r-- 1 machinekit machinekit   1330 May 19 16:33 encode_rs.c
-rw-r--r-- 1 machinekit machinekit     67 May 19 16:33 iperf_client.sh
-rw-r--r-- 1 machinekit machinekit     52 May 19 16:33 iperf_server.sh
-rw-r--r-- 1 machinekit machinekit  35147 May 19 16:33 LICENSE.md
-rw-r--r-- 1 machinekit machinekit    332 May 19 16:57 load_driver_client.sh
-rw-r--r-- 1 machinekit machinekit    332 May 19 16:56 load_driver_server.sh
-rw-r--r-- 1 machinekit machinekit   1068 May 19 16:33 Makefile
-rw-r--r-- 1 machinekit machinekit     46 May 19 16:35 modules.order
-rw-r--r-- 1 machinekit machinekit    539 May 19 16:35 Module.symvers
-rw-r--r-- 1 machinekit machinekit  60630 May 19 16:33 openvlc.c
-rw-r--r-- 1 machinekit machinekit   3288 May 19 16:33 openvlc.h
-rw-r--r-- 1 machinekit machinekit 237504 May 19 16:35 openvlc.o
-rw-r--r-- 1 machinekit machinekit    384 May 19 16:33 README.md
-rw-r--r-- 1 machinekit machinekit  12068 May 19 16:33 reed_solomon.c
-rw-r--r-- 1 machinekit machinekit  50226 May 19 16:35 reed_solomon.o
-rw-r--r-- 1 machinekit machinekit   4299 May 19 16:33 rs.c
-rw-r--r-- 1 machinekit machinekit   3069 May 19 16:33 rslib.h
-rw-r--r-- 1 machinekit machinekit 303857 May 19 16:35 vlc.ko
-rw-r--r-- 1 machinekit machinekit   2521 May 19 16:35 vlc.mod.c
-rw-r--r-- 1 machinekit machinekit  21004 May 19 16:35 vlc.mod.o
-rw-r--r-- 1 machinekit machinekit 285437 May 19 16:35 vlc.o
```

```
-rw-r--r-- 1 machinekit machinekit   6959 May 19 16:33 decode_rs.c
-rw-r--r-- 1 machinekit machinekit   1330 May 19 16:33 encode_rs.c
-rw-r--r-- 1 machinekit machinekit     67 May 19 16:33 iperf_client.sh
-rw-r--r-- 1 machinekit machinekit     52 May 19 16:33 iperf_server.sh
-rw-r--r-- 1 machinekit machinekit  35147 May 19 16:33 LICENSE.md
-rwxr--r-- 1 machinekit machinekit    332 May 19 16:57 load_driver_client.sh
-rwxr--r-- 1 machinekit machinekit    332 May 19 16:56 load_driver_server.sh
-rw-r--r-- 1 machinekit machinekit   1068 May 19 16:33 Makefile
-rw-r--r-- 1 machinekit machinekit     46 May 19 16:35 modules.order
-rw-r--r-- 1 machinekit machinekit    539 May 19 16:35 Module.symvers
-rw-r--r-- 1 machinekit machinekit  60630 May 19 16:33 openvlc.c
-rw-r--r-- 1 machinekit machinekit   3288 May 19 16:33 openvlc.h
-rw-r--r-- 1 machinekit machinekit 237504 May 19 16:35 openvlc.o
-rw-r--r-- 1 machinekit machinekit    384 May 19 16:33 README.md
-rw-r--r-- 1 machinekit machinekit  12068 May 19 16:33 reed_solomon.c
-rw-r--r-- 1 machinekit machinekit  50226 May 19 16:35 reed_solomon.o
-rw-r--r-- 1 machinekit machinekit   4299 May 19 16:33 rs.c
-rw-r--r-- 1 machinekit machinekit   3069 May 19 16:33 rslib.h
-rw-r--r-- 1 machinekit machinekit 303857 May 19 16:35 vlc.ko
-rw-r--r-- 1 machinekit machinekit   2521 May 19 16:35 vlc.mod.c
-rw-r--r-- 1 machinekit machinekit  21004 May 19 16:35 vlc.mod.o
-rw-r--r-- 1 machinekit machinekit 285437 May 19 16:35 vlc.o
```

14) Finally we have to verify if the installation was successful. We will develop a little test with 2 beagle bones, person A must write the instruction: *sudo ./load_driver_client.sh* and person B must write the instruction: *sudo ./load_driver_server.sh.* After that person A must write the instruction: *ping 192.168.0.1* and person B must write the instruction *ping 192.168.0.2. (*Make sure that the led is already place in the BBB). As soon as the instruction is introduced to the machinekit@beaglebone the led will start to turn on and off, stay both beagle bones close and in the machinekit@beaglebone it will display this message. (To stop the program press control + c).

```
machinekit@beaglebone:~/openvlc-master$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_req=10 ttl=64 time=115 ms
64 bytes from 192.168.0.1: icmp_req=11 ttl=64 time=115 ms
64 bytes from 192.168.0.1: icmp_req=12 ttl=64 time=106 ms
64 bytes from 192.168.0.1: icmp_req=13 ttl=64 time=95.7 ms
64 bytes from 192.168.0.1: icmp_req=14 ttl=64 time=115 ms
64 bytes from 192.168.0.1: icmp_req=15 ttl=64 time=109 ms
64 bytes from 192.168.0.1: icmp_req=16 ttl=64 time=206 ms
64 bytes from 192.168.0.1: icmp_req=18 ttl=64 time=99.6 ms
64 bytes from 192.168.0.1: icmp_req=19 ttl=64 time=107 ms
64 bytes from 192.168.0.1: icmp_req=21 ttl=64 time=216 ms
64 bytes from 192.168.0.1: icmp_req=22 ttl=64 time=111 ms
64 bytes from 192.168.0.1: icmp_req=23 ttl=64 time=115 ms
64 bytes from 192.168.0.1: icmp_req=24 ttl=64 time=103 ms
64 bytes from 192.168.0.1: icmp_req=25 ttl=64 time=109 ms
64 bytes from 192.168.0.1: icmp_req=48 ttl=64 time=125 ms
64 bytes from 192.168.0.1: icmp_req=49 ttl=64 time=115 ms
64 bytes from 192.168.0.1: icmp_req=50 ttl=64 time=99.1 ms
64 bytes from 192.168.0.1: icmp_req=51 ttl=64 time=606 ms
64 bytes from 192.168.0.1: icmp_req=52 ttl=64 time=492 ms
64 bytes from 192.168.0.1: icmp_req=53 ttl=64 time=99.6 ms
64 bytes from 192.168.0.1: icmp_req=54 ttl=64 time=212 ms
64 bytes from 192.168.0.1: icmp_req=55 ttl=64 time=119 ms
64 bytes from 192.168.0.1: icmp_req=56 ttl=64 time=212 ms
64 bytes from 192.168.0.1: icmp_req=58 ttl=64 time=714 ms
64 bytes from 192.168.0.1: icmp_req=61 ttl=64 time=111 ms
64 bytes from 192.168.0.1: icmp_req=62 ttl=64 time=120 ms
64 bytes from 192.168.0.1: icmp_req=63 ttl=64 time=322 ms
64 bytes from 192.168.0.1: icmp_req=65 ttl=64 time=115 ms
64 bytes from 192.168.0.1: icmp_req=66 ttl=64 time=107 ms
```