

# 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/](https://www.sdcard.org/downloads/formatter_4/)

## SD Card Formatter 4.0 for Windows and Mac

Download SD Card Formatter for Windows >

Download SD Card Formatter for Mac >

Released on January 30, 2013

Released on January 30, 2013

### Article 8 Termination of License

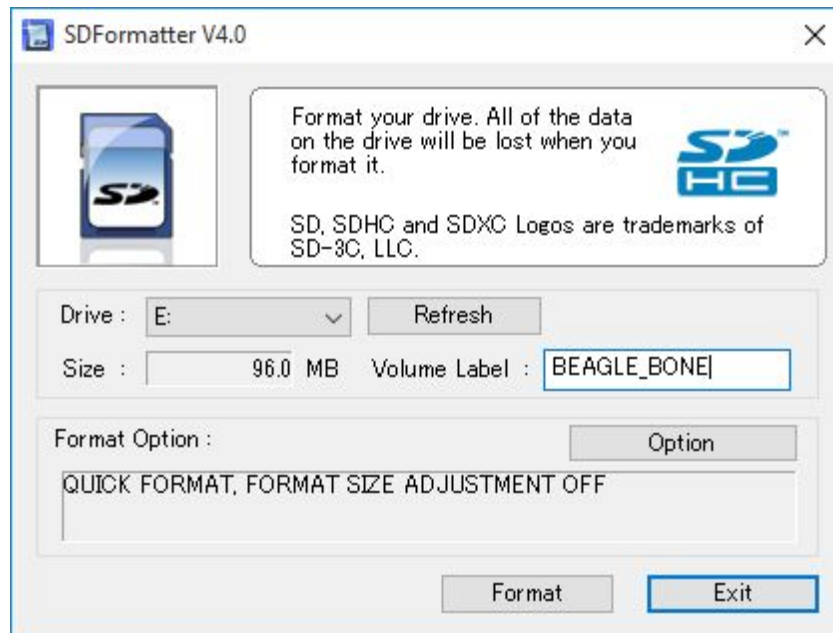
The rights granted to Licensee hereunder will be automatically terminated if Licensee contravenes of any of the terms and conditions of this Agreement. In the event, Licensee must destroy the Software and related documentation together with all the copies thereof at Licensee's own expense.

Back >

Decline >

Accept >

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>

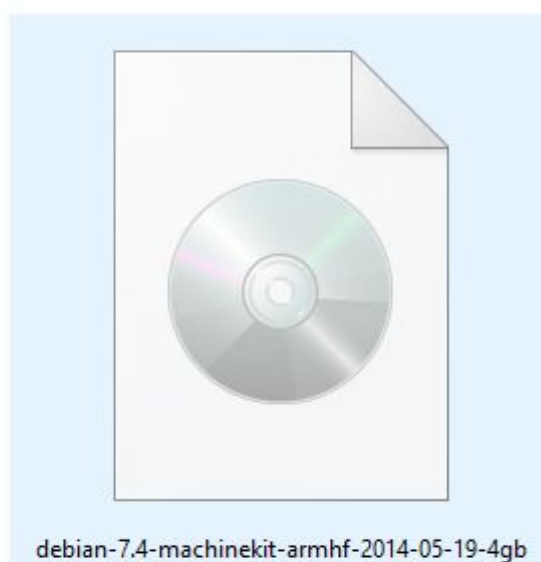


## Download

Download 7-Zip 16.04 (2016-10-04) for Windows:

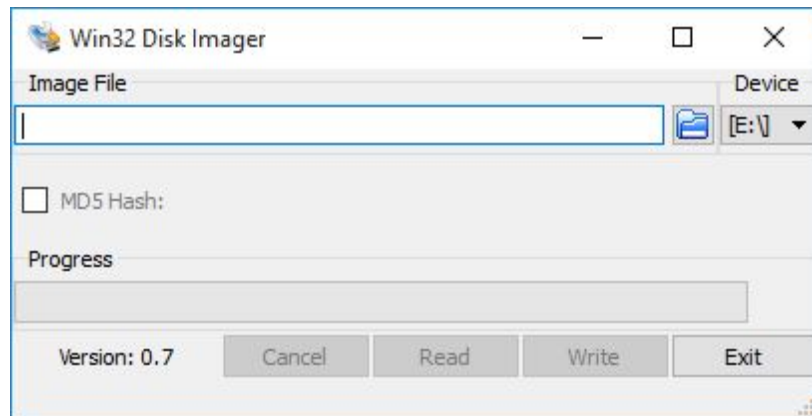
Link	Type	Windows	Description
<a href="#">Download</a>	.exe	32-bit x86	7-Zip for 32-bit Windows
<a href="#">Download</a>	.exe	64-bit x64	7-Zip for 64-bit Windows x64 (Intel 64 or AMD64)

QtGui4.dll	Extraer en debian-7.4-machinekit-armhf-2014-05-19-4gb.img\	9,91
bone-debian-8.6-lxqt-4gb-armhf-2016-11-06-4gb.img	7-Zip	> 710,15
debian-7.4-machinekit-armhf-2014-05-19-4gb.img	CRC SHA	> 661,83

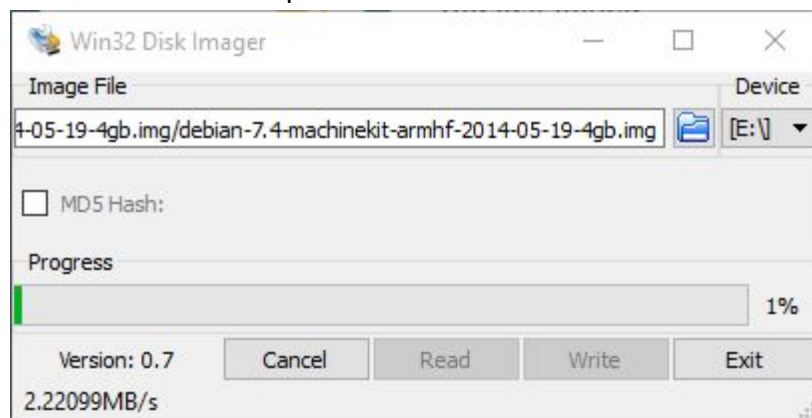


4) Download Win32 Disk Imager to write the image into the SD card

<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.

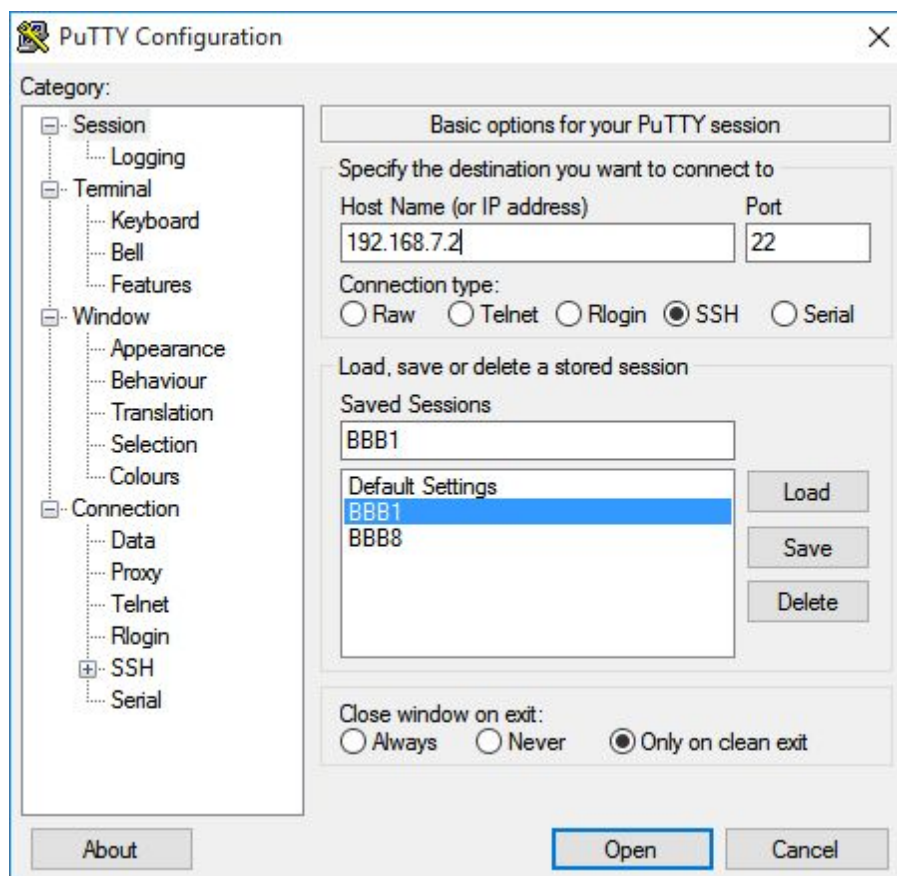
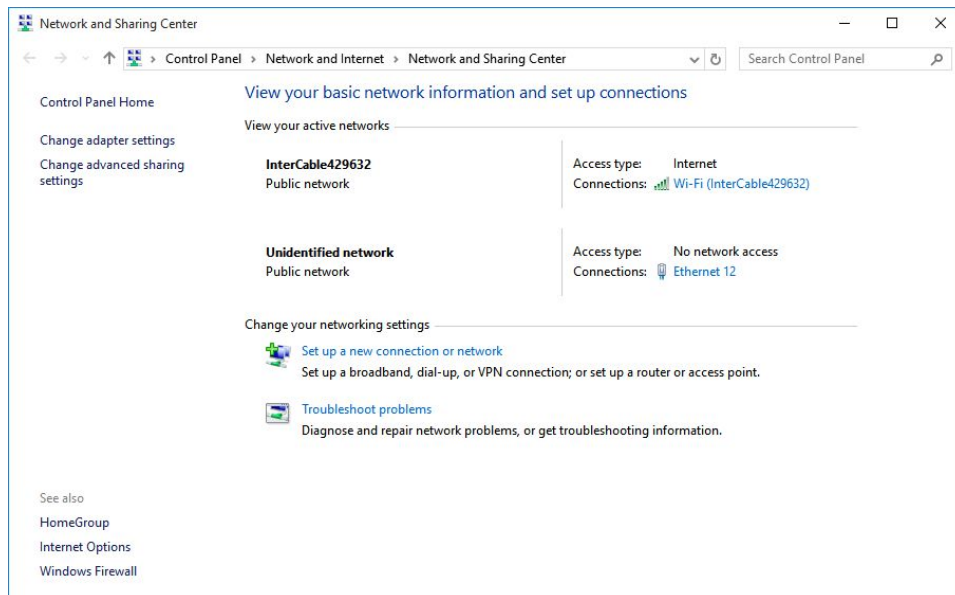


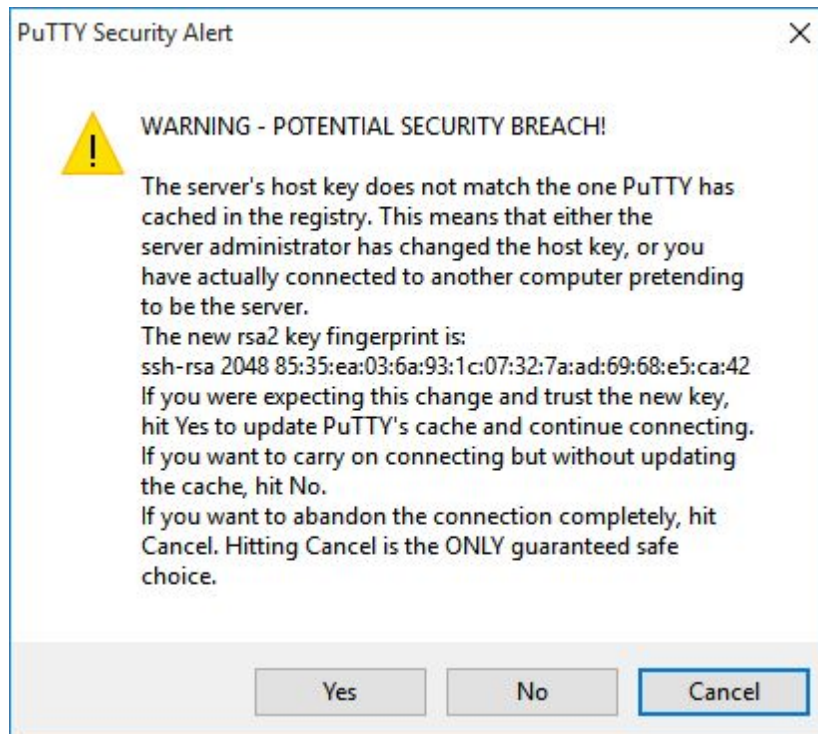
5) Install the drivers for the BBB

<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**.

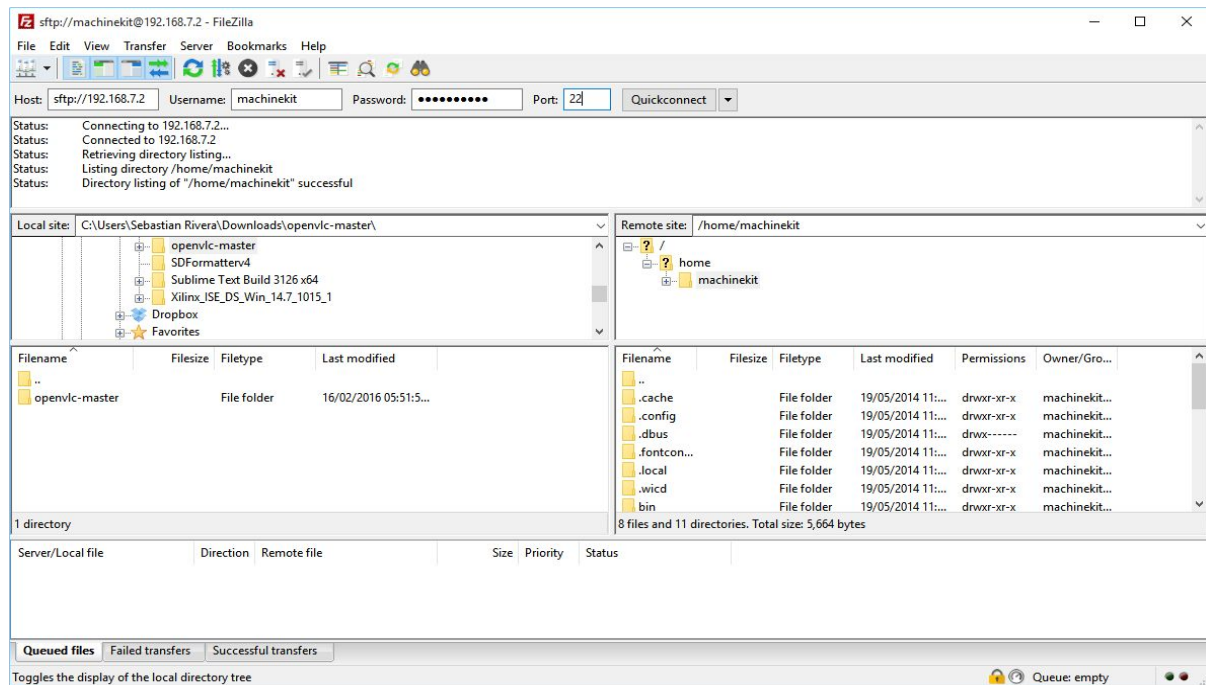




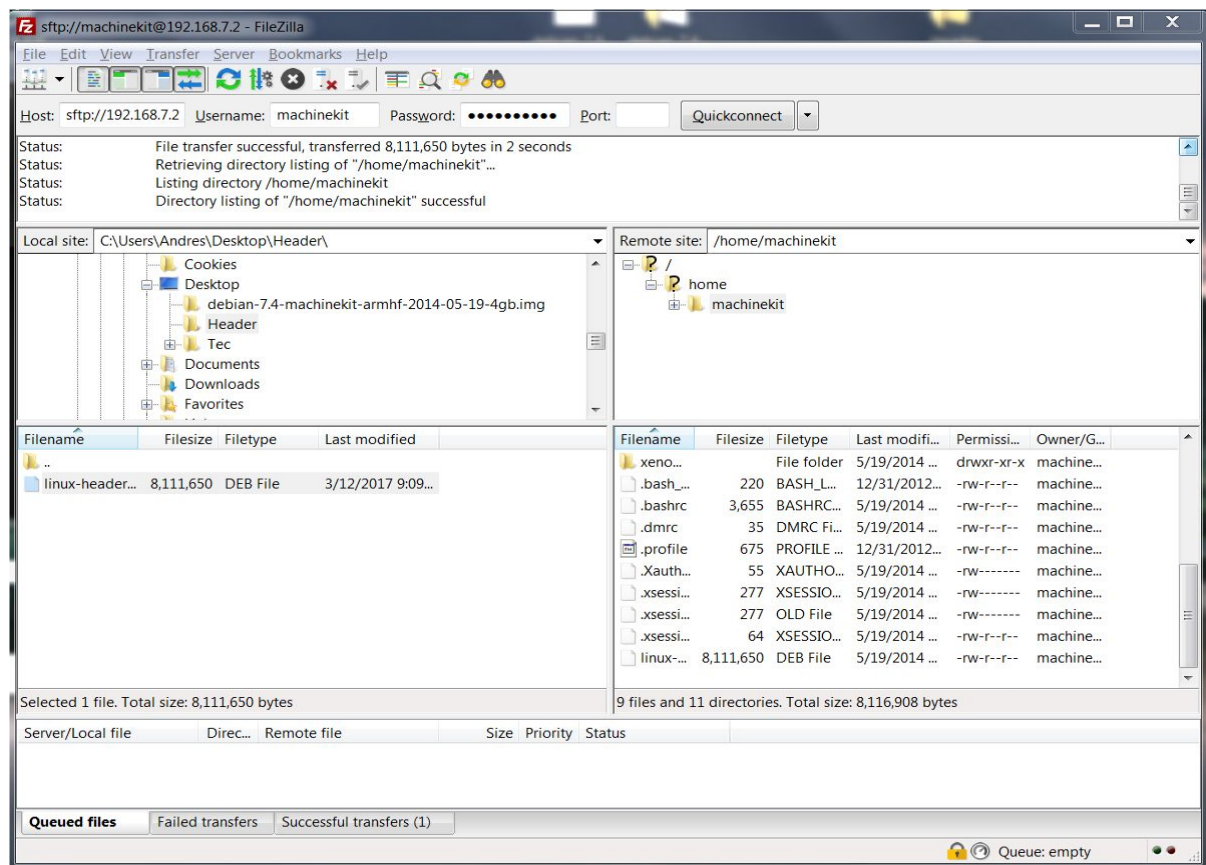
```
machinekit@beaglebone: ~  
login as: machinekit  
Debian GNU/Linux 7  
  
BeagleBoard.org BeagleBone Debian Image 2014-05-19  
  
Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian  
machinekit@192.168.7.2's password:  
Linux beaglebone 3.8.13xenomai-bone53 #2 Mon May 19 14:43:41 UTC 2014 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Environment set up for running Machinekit and LinuxCNC  
machinekit@beaglebone:~$ ls  
bin Desktop dtc machinekit-dev xenomai-2.6  
machinekit@beaglebone:~$
```

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

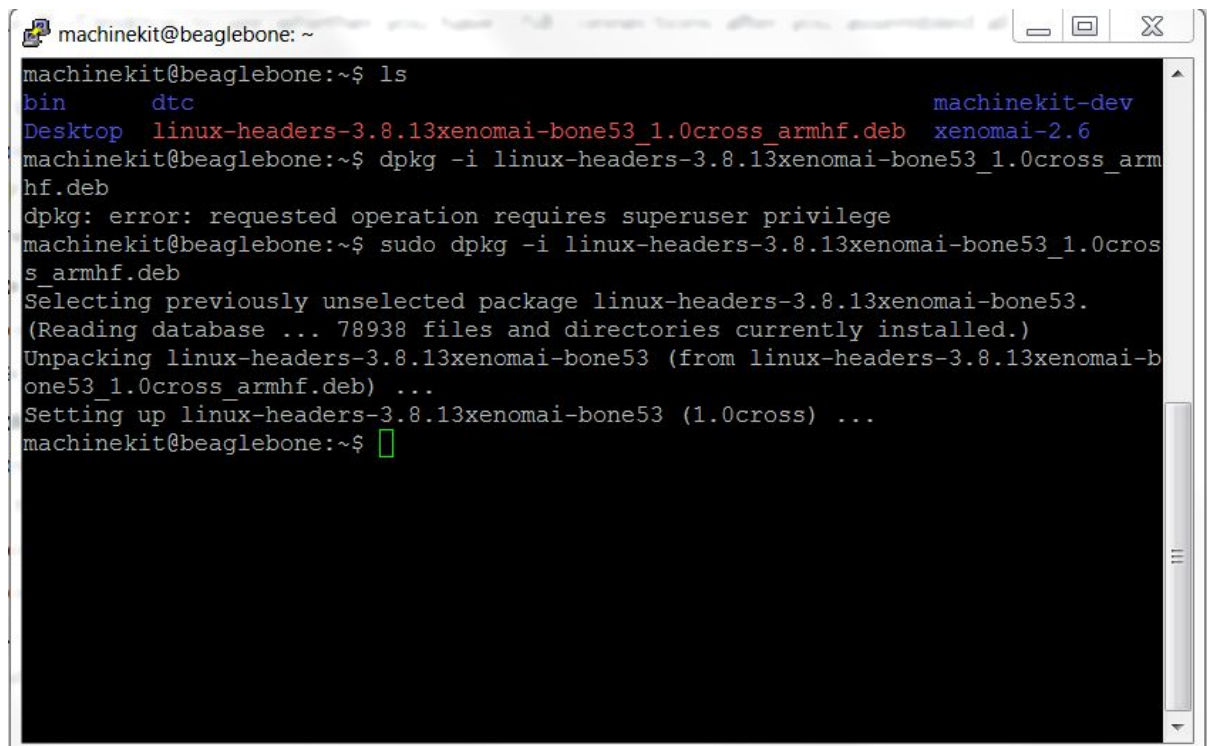




8) Search the linux headers in the *local site*, double click when you find the header and wait until the copy is done. The header must appear in the right box as in the following image.

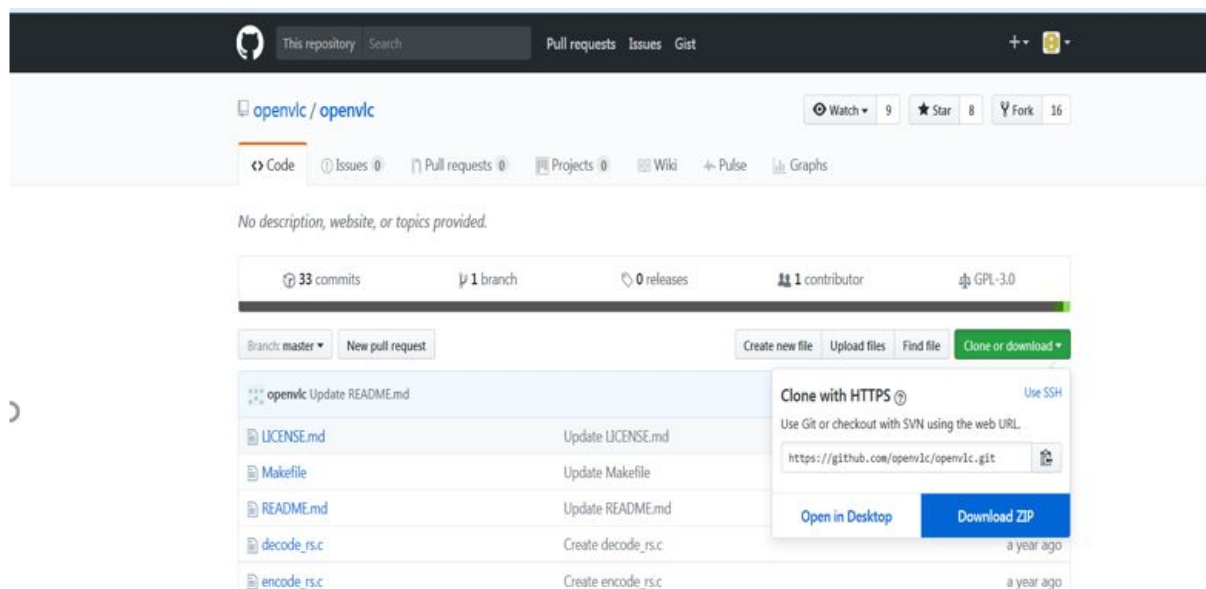


9) Go back to the machinekit@beaglebone, and write the following: `sudo dpkg -i linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb`. Once you finish, press enter and wait until the machinekit@beaglebone display the same message as in the following image.



```
machinekit@beaglebone: ~  
machinekit@beaglebone:~$ ls  
bin      dtc      machinekit-dev  xenomai-2.6  
Desktop  linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb  
machinekit@beaglebone:~$ dpkg -i linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb  
dpkg: error: requested operation requires superuser privilege  
machinekit@beaglebone:~$ sudo dpkg -i linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb  
Selecting previously unselected package linux-headers-3.8.13xenomai-bone53.  
(Reading database ... 78938 files and directories currently installed.)  
Unpacking linux-headers-3.8.13xenomai-bone53 (from linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb) ...  
Setting up linux-headers-3.8.13xenomai-bone53 (1.0cross) ...  
machinekit@beaglebone:~$
```

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

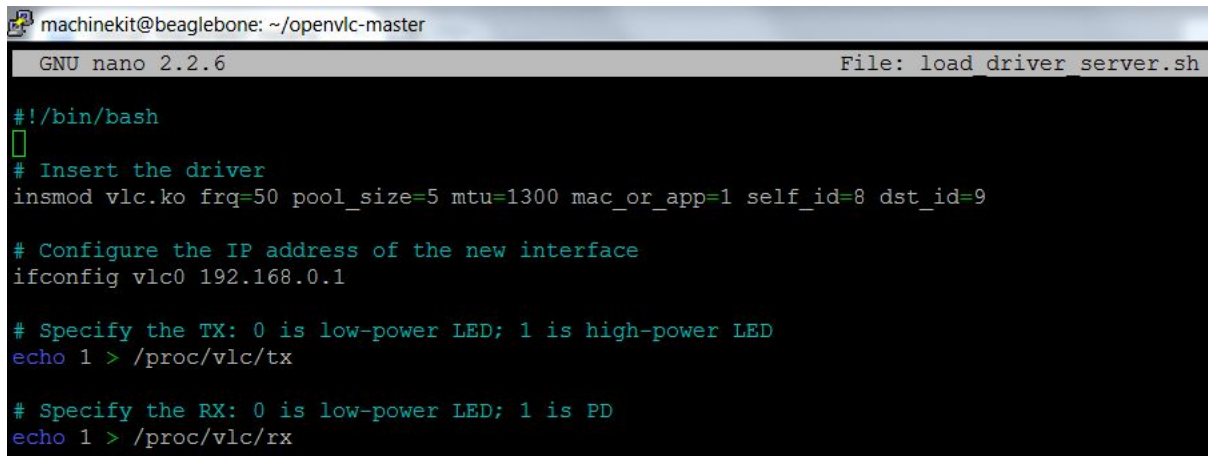


11) Go back to Filezilla and find the *openvlc-master* and do the same steps as when you transfer the headers (8). After that go back to the *machinekit@beaglebone* and go to the *openvlc-master* file, once you are there write: *make clean; make* wait until *machinekit@beaglebone* displays the same message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

```
machinekit@beaglebone: ~/openvlc-master
machinekit@beaglebone:~$ ls
bin      dtc                                     machinekit-dev
Desktop  linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb  xenomai-2.6
machinekit@beaglebone:~$ dpkg -i linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb
dpkg: error: requested operation requires superuser privilege
machinekit@beaglebone:~$ sudo dpkg -i linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb
Selecting previously unselected package linux-headers-3.8.13xenomai-bone53.
(Reading database ... 78938 files and directories currently installed.)
Unpacking linux-headers-3.8.13xenomai-bone53 (from linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb) ...
Setting up linux-headers-3.8.13xenomai-bone53 (1.0cross) ...
machinekit@beaglebone:~$ ls
bin      linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb  xenomai-2.6
Desktop  machinekit-dev
dtc      openvlc-master
machinekit@beaglebone:~$ cd
machinekit@beaglebone:~$ cd openvlc-master/
machinekit@beaglebone:~/openvlc-master$ ls
decode_rs.c  LICENSE.md  openvlc.c  rs.c
encode_rs.c  load_driver_client.sh  openvlc.h  rslib.h
iperf_client.sh  load_driver_server.sh  README.md
iperf_server.sh  Makefile  reed_solomon.c
machinekit@beaglebone:~/openvlc-master$ make clean; make
rm -f .openvlc* *.o *.ko *.mod.c Module*.symvers Module.markers modules.order
rm -f -R .tmp*
make -C /lib/modules/3.8.13xenomai-bone53/build SUBDIRS=/home/machinekit/openvlc-master modules
make[1]: Entering directory `/usr/src/linux-headers-3.8.13xenomai-bone53'
  CC [M]  /home/machinekit/openvlc-master/reed_solomon.o
  CC [M]  /home/machinekit/openvlc-master/openvlc.o
/home/machinekit/openvlc-master/openvlc.c: In function 'phy_decoding':
/home/machinekit/openvlc-master/openvlc.c:1075:9: warning: unused variable 'max_un_reception' [-Wunused-variable]
/home/machinekit/openvlc-master/openvlc.c: In function 'phy_timer_handler':
/home/machinekit/openvlc-master/openvlc.c:1329:9: warning: unused variable 'prev_value' [-Wunused-variable]
/home/machinekit/openvlc-master/openvlc.c: At top level:
/home/machinekit/openvlc-master/openvlc.c:206:12: warning: 'prev_hpl' defined but not used [-Wunused-variable]
  LD [M]  /home/machinekit/openvlc-master/vlc.o
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/machinekit/openvlc-master/vlc.mod.o
  LD [M]  /home/machinekit/openvlc-master/vlc.ko
make[1]: Leaving directory `/usr/src/linux-headers-3.8.13xenomai-bone53'
machinekit@beaglebone:~/openvlc-master$ make clean; make
```



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.



```
machinekit@beaglebone: ~/openvlc-master
GNU nano 2.2.6 File: load_driver_server.sh

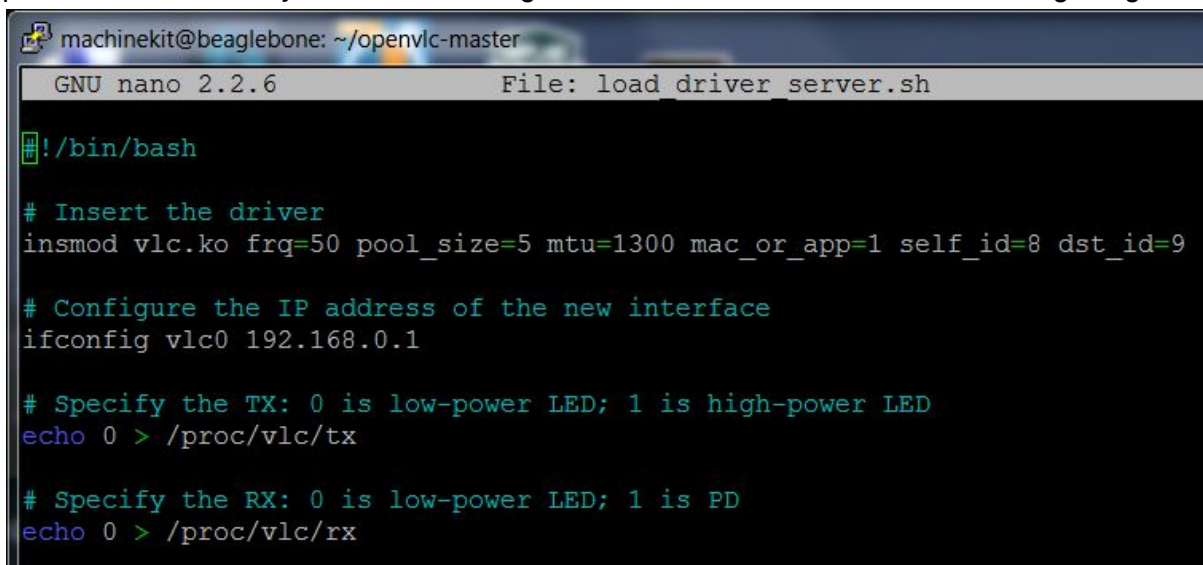
#!/bin/bash
# Insert the driver
insmod vlc.ko frq=50 pool_size=5 mtu=1300 mac_or_app=1 self_id=8 dst_id=9

# Configure the IP address of the new interface
ifconfig vlc0 192.168.0.1

# Specify the TX: 0 is low-power LED; 1 is high-power LED
echo 1 > /proc/vlc/tx

# Specify the RX: 0 is low-power LED; 1 is PD
echo 1 > /proc/vlc/rx
```

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.



```
machinekit@beaglebone: ~/openvlc-master
GNU nano 2.2.6 File: load_driver_server.sh

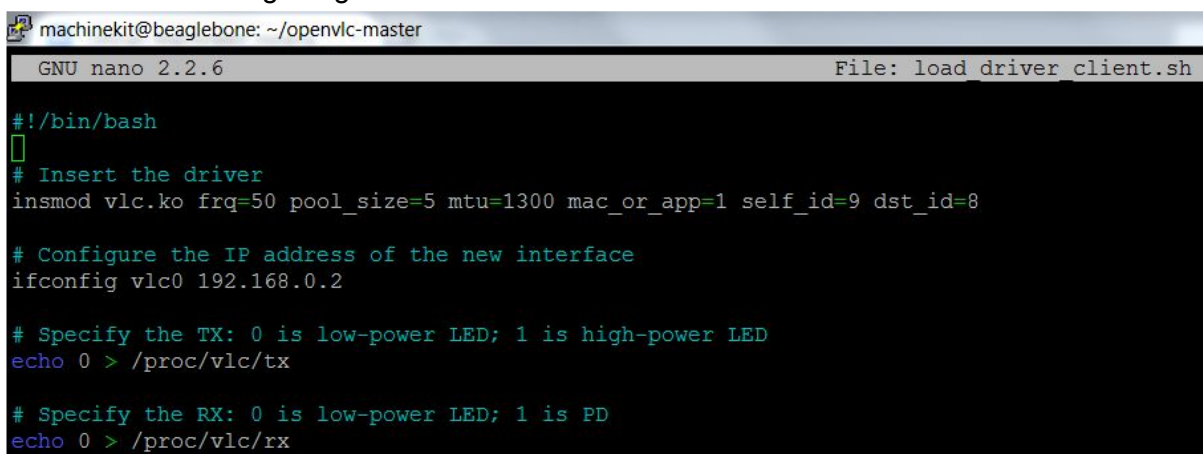
#!/bin/bash
# Insert the driver
insmod vlc.ko frq=50 pool_size=5 mtu=1300 mac_or_app=1 self_id=8 dst_id=9

# Configure the IP address of the new interface
ifconfig vlc0 192.168.0.1

# Specify the TX: 0 is low-power LED; 1 is high-power LED
echo 0 > /proc/vlc/tx

# Specify the RX: 0 is low-power LED; 1 is PD
echo 0 > /proc/vlc/rx
```

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



```
machinekit@beaglebone: ~/openvlc-master
GNU nano 2.2.6 File: load_driver_client.sh

#!/bin/bash
# Insert the driver
insmod vlc.ko frq=50 pool_size=5 mtu=1300 mac_or_app=1 self_id=9 dst_id=8

# Configure the IP address of the new interface
ifconfig vlc0 192.168.0.2

# Specify the TX: 0 is low-power LED; 1 is high-power LED
echo 0 > /proc/vlc/tx

# Specify the RX: 0 is low-power LED; 1 is PD
echo 0 > /proc/vlc/rx
```

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:~/openvnc-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
```