

Guía de instalación del Debian machikit ISO image en el Beaglebone Black - Windows

Esta guía tiene como propósito de explicar paso a paso como escribir un ISO image en la tarjeta SD para posteriormente usarla en el Beaglebone Black (BBB). El sistemas operativo usado para esta guía es Windows 10. La información oficial está ubicada en la página de internet del proyecto: <http://www.openvlc.org/openvlc.html>. Antes de empezar, asegúrese de tener una conexión de internet estable para poder descargar con normalidad los softwares para las siguientes instrucciones.

Instalacion paso a paso

1. Descargue el *Debian machinekit image*. Para descargar el *image* siga el siguiente link: <https://drive.google.com/file/d/0BwGT2J3dvAfNOEVibS1KQ2d5MGc/view>. Cuando la descarga este completa modifique el nombre del archivo. Cambiarlo de **.img.xz** a **.img**.
2. Formatear la tarjeta SD.
https://www.sdcard.org/downloads/formatter_4/

SD Card Formatter 4.0 for Windows and Mac

Download SD Card Formatter for Windows >

Released on January 30, 2013

Download SD Card Formatter for Mac >

Released on January 30, 2013

Article 8 Termination of License

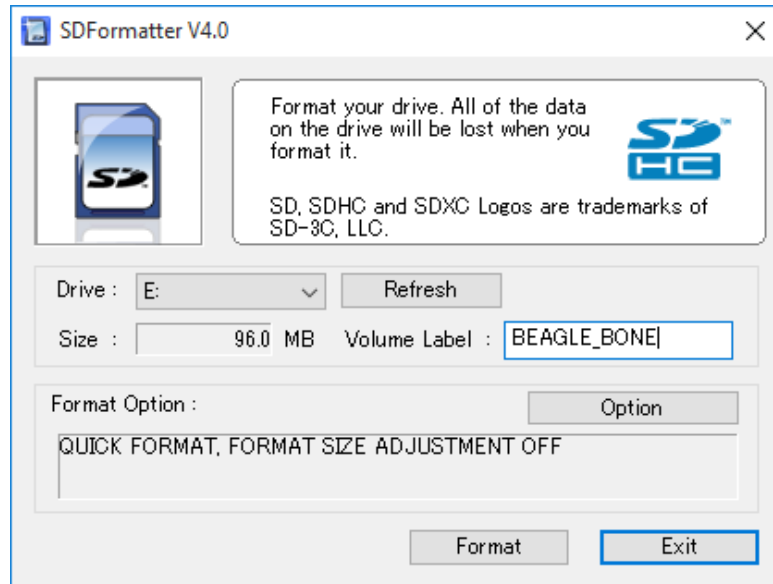
The rights granted to Licensee hereunder will be automatically terminated if Licensee contravenes of any of the terms and conditions of this Agreement. In the event, Licensee must destroy the Software and related documentation together with all the copies thereof at Licensee's own expense.

Back >

Decline >

Accept >


Cuando le dé clic para aceptar, debería comenzar a descargar. Extraiga los archivos del .zip folder y en este encontrara un archivo .exe. Córralo y la ventana de instalación aparecerá, a continuación, proceda a instalar la *SD Formatter*.



Cuando termine la instalacion, corra el programa y despues abra la ventana similar a esta. Solo de clic en *Format* y desues de unos segundos la tarjeta SD deberia estar lista.

3. Descargue 7 zip para extraer el ISO file.

<http://7-zip.org/download.html>



Home
7z Format
LZMA SDK

Download

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

Link	Type	Windows	Description
Download	.exe	32-bit x86	7-Zip for 32-bit Windows
Download	.exe	64-bit x64	7-Zip for 64-bit Windows x64 (Intel 64 or AMD64)

QtGui4.dll

bone-debian-8.6-lxqt-4gb-armhf-2016-11-06-4gb.img

debian-7.4-machinekit-armhf-2014-05-19-4gb.img

Extraer en debian-7.4-machinekit-armhf-2014-05-19-4gb.img\

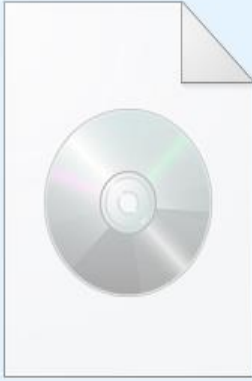
7-Zip

CRC SHA

9,91

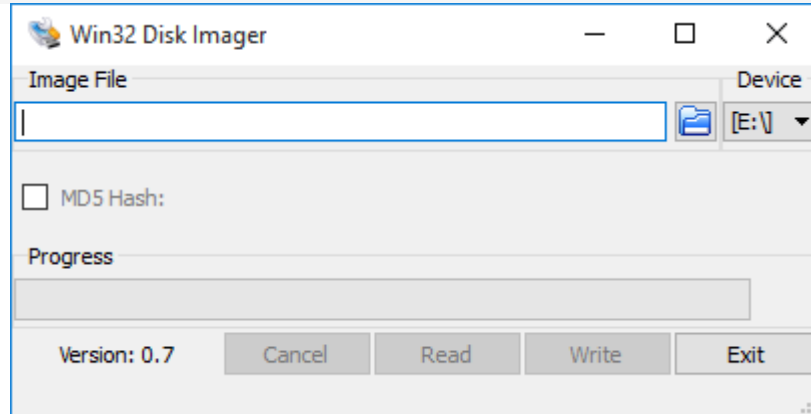
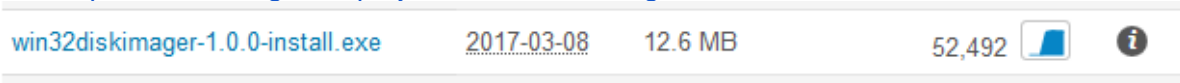
710,15

661,83

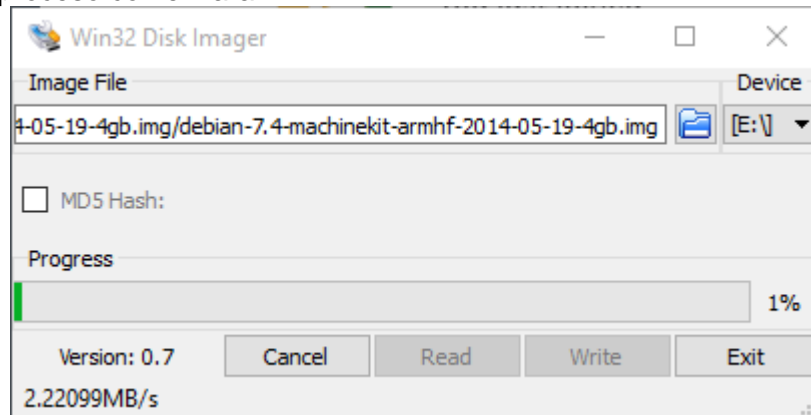


debian-7.4-machinekit-armhf-2014-05-19-4gb

4. Descargue *Win32 Disk Imager* para escribir el *image* a la tarjeta SD.
<https://sourceforge.net/projects/win32diskimager/files/Archive/>



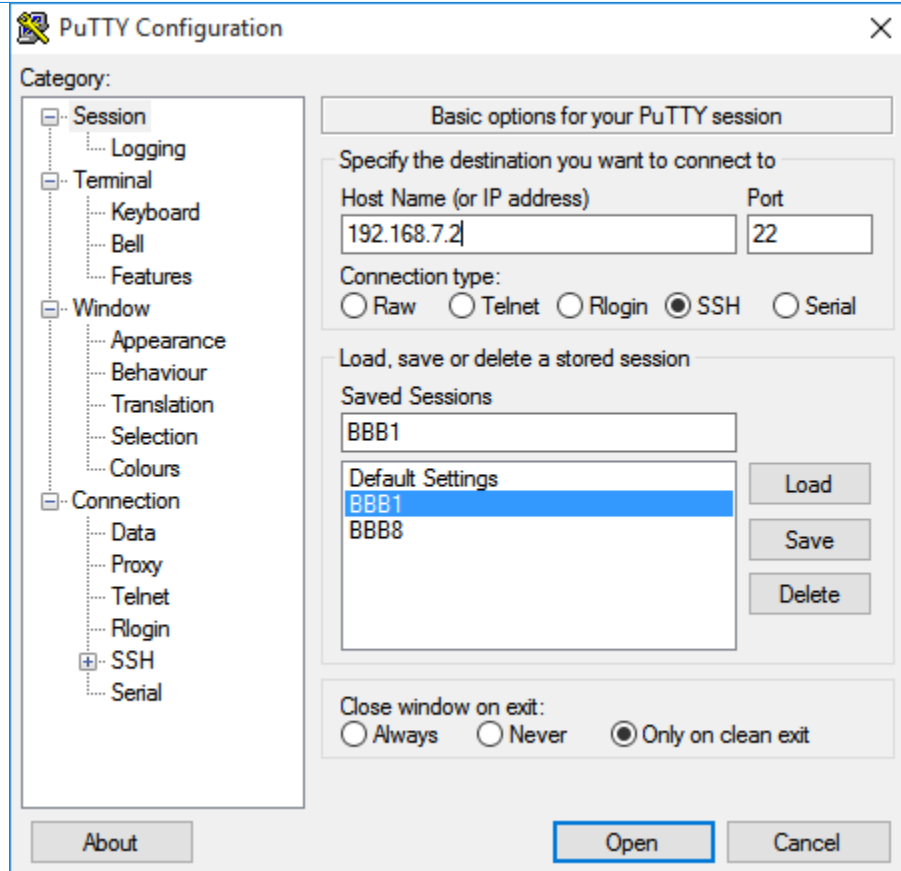
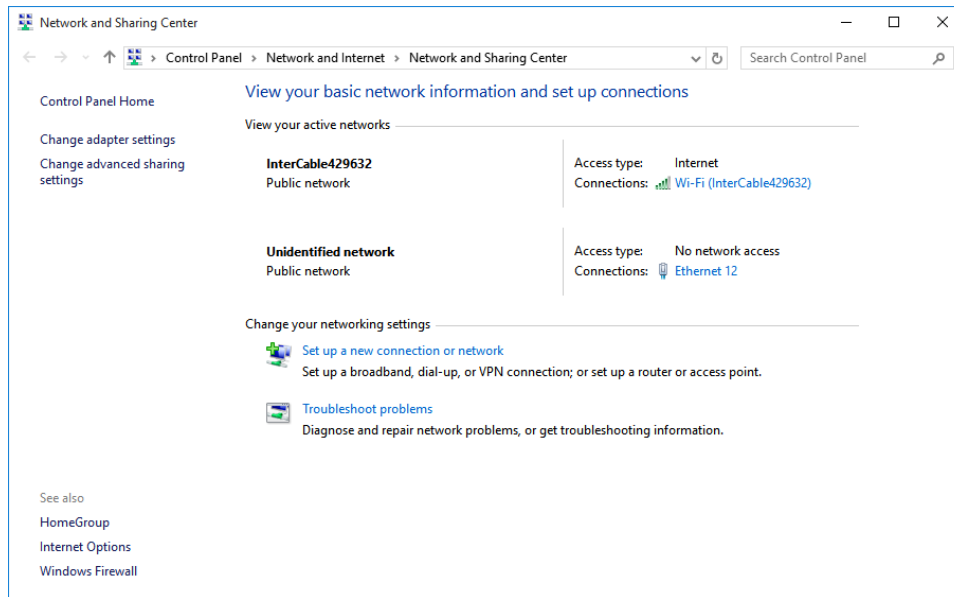
Seleccione la tarjeta SD en el lugar de *Device* y después haga clic en el folder azul y busque su archivo *.img*. Cuando lo seleccione, la opción de **Write** estará disponible. Después el proceso comenzará.

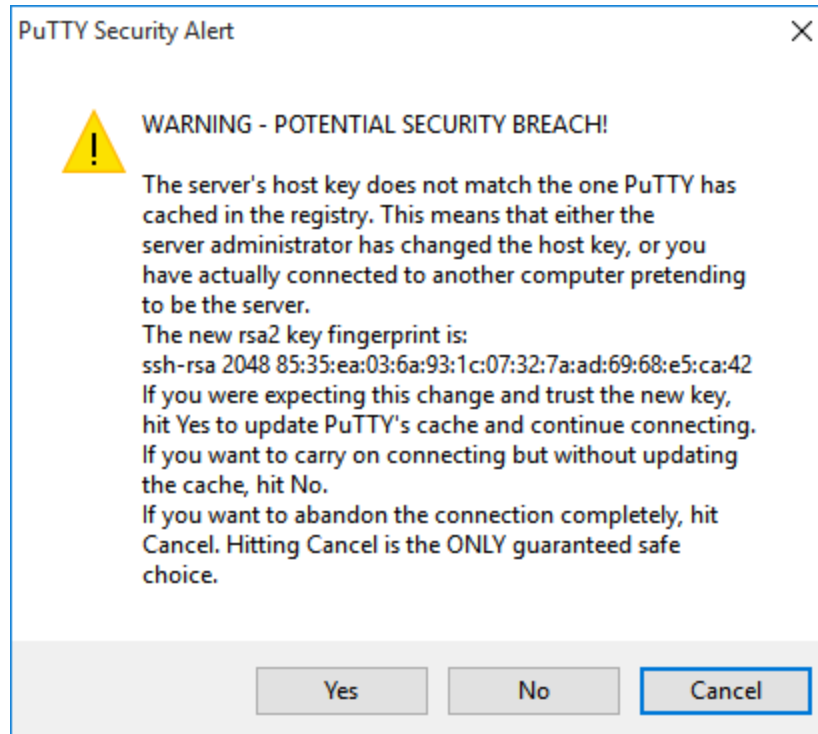


5. Instalar los drivers del BBB.
<https://learn.adafruit.com/ssh-to-beaglebone-black-over-usb/installing-drivers-windows>

6. Arrancar el OS en el BBB

Cuando el proceso de escribir *image* en la tarjeta SD esté terminado, conéctelo al BeagleBone Black y también conecte este por USB a la computadora. Para este paso usted deberá descargar *Putty* (<http://www.putty.org/>) para poder SSH de BBB. La dirección de ip indeterminada para el BBB es 192.168.7.2, entonces usted tendrá que escribirlo al momento de ingresar al software *Putty*. Abrir el **Network and Sharing Center** para revisar si el BBB está siendo detectado como un **Unidentified network**.

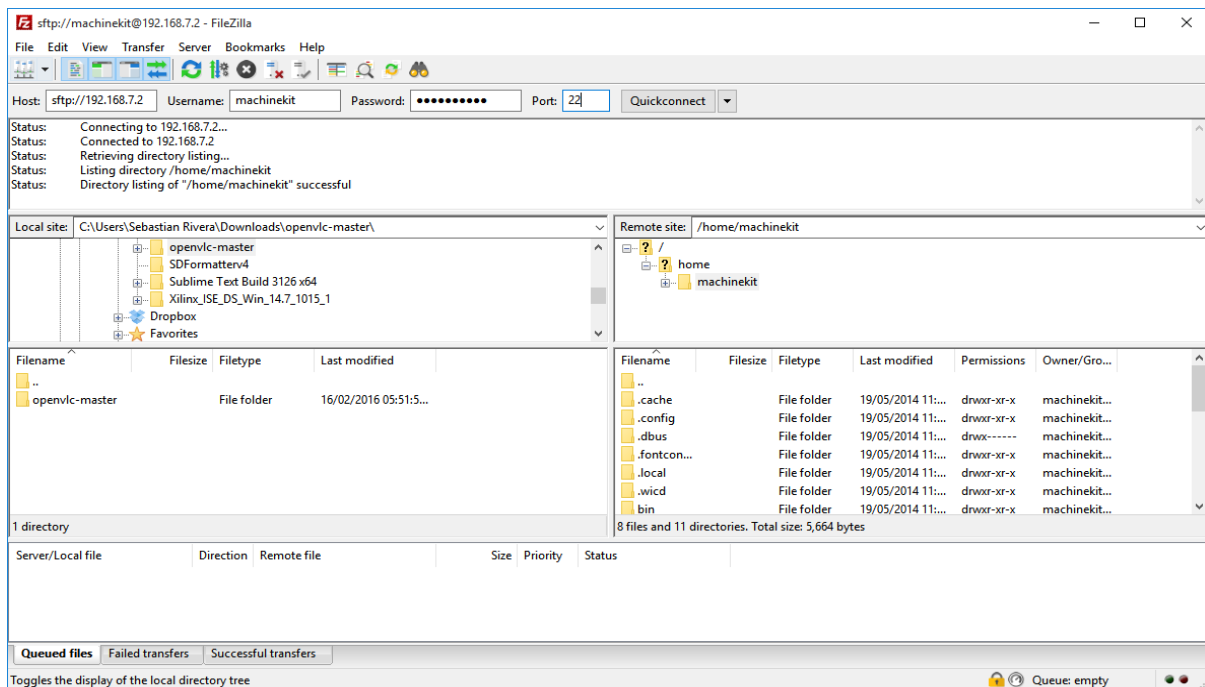




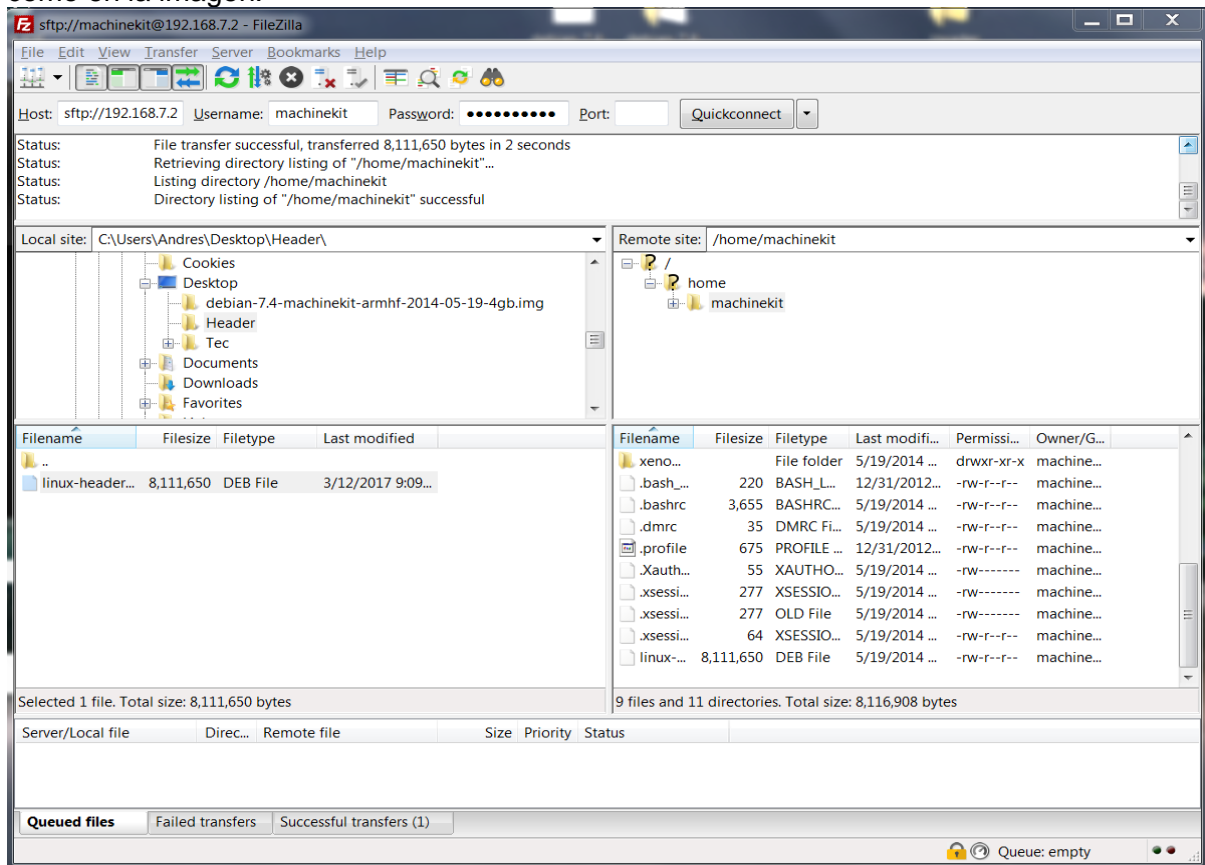
```
machinekit@beaglebone: ~  
login as: machinekit  
Debian GNU/Linux 7  
  
BeagleBoard.org BeagleBone Debian Image 2014-05-19  
  
Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian  
machinekit@192.168.7.2's password:  
Linux beaglebone 3.8.13xenomai-bone53 #2 Mon May 19 14:43:41 UTC 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. Descargar *FileZilla Client* para poder intercambiar archivos entre la computadora y el BBB.

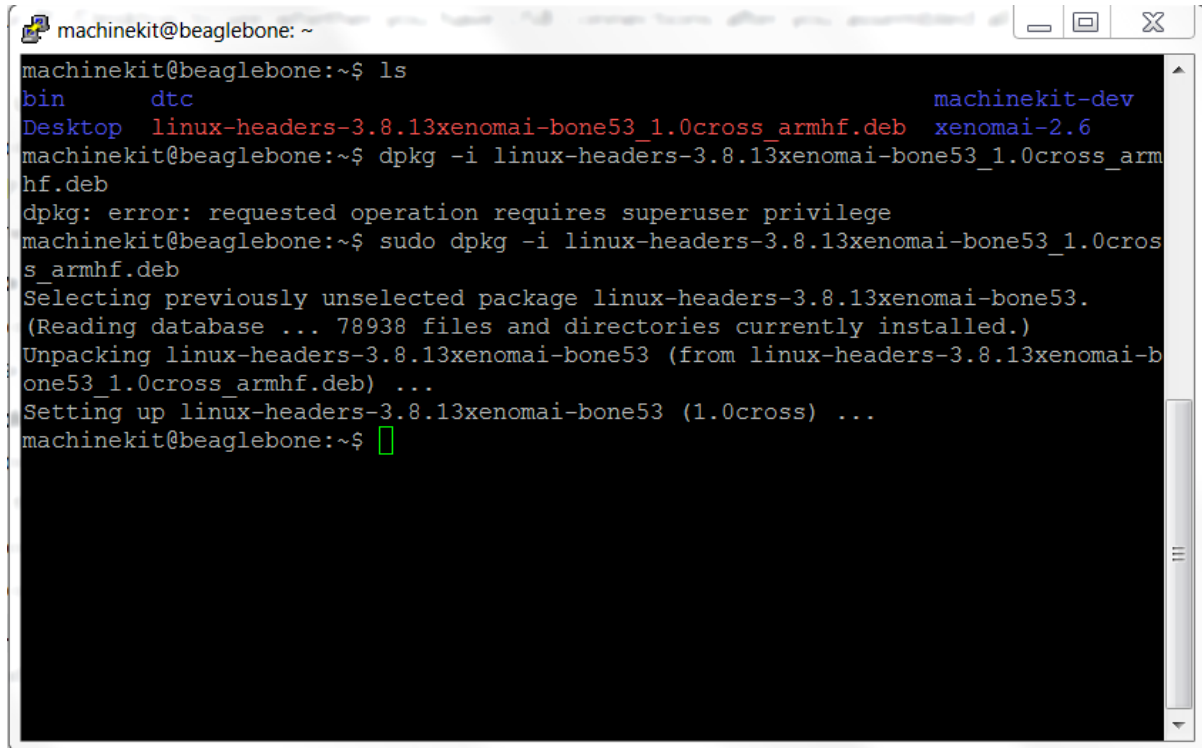
<https://filezilla-project.org/download.php>



8) Busque los *headers linux* en lugar de *local site*, de doble clic cuando encuentre el *header* y espere hasta se complete la copia. El *header* deberá aparecer en el recuadro de la derecha como en la imagen.

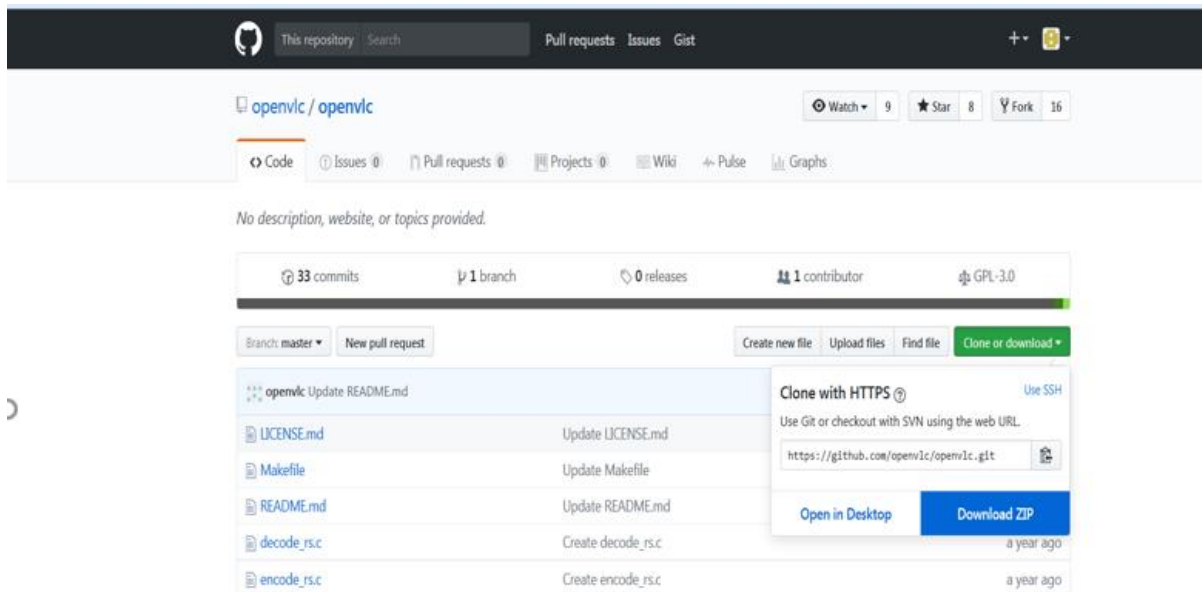


9) Vuelva a `machinekit@beaglebone`, y escriba lo siguiente: `sudo dpkg -i linux-headers-3.8.13xenomai-bone53_1.0cross_armhf.deb`. Una vez terminado, presione `enter` y espere hasta que `machinekit@beaglebone` muestre el mismo mensaje que en la siguiente imagen.



```
machinekit@beaglebone: ~  
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:~$
```

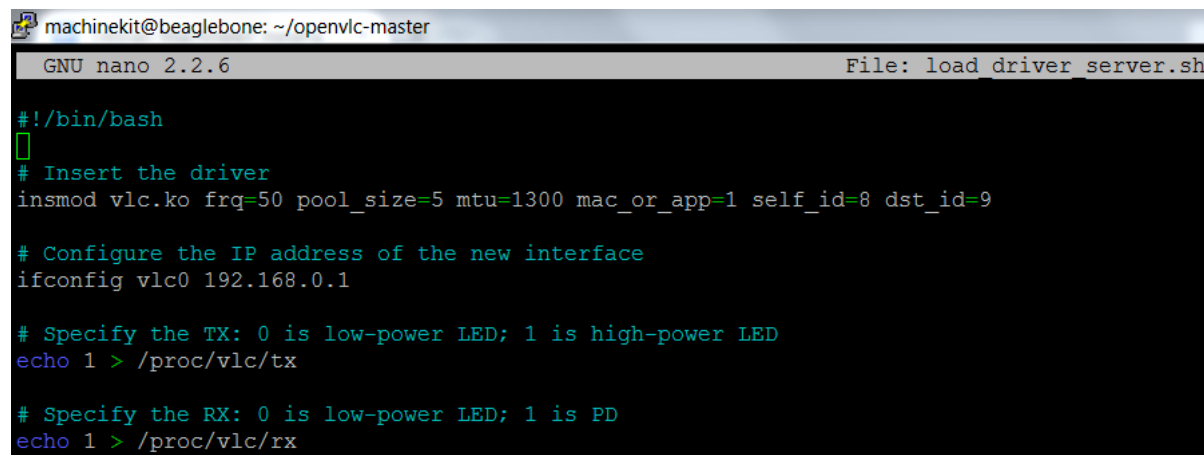
10) Siga el siguiente link: <https://github.com/openvlc/openvlc> y de clic derecho en el *clone or download* y busque la opción de *Download ZIP*.



11) Vuelva a Filezilla y encuentre *openvlc-master* y siga los mismos pasos como cuando transferimos los *headers* (8). Despues regresar al machinekit@beaglebone y dirigirse a *openvlc-master* file, una vez que se encuentre en ese lugar escribir lo siguiente: *make clean*; *make* esperar hasta que se despliegue el mensaje de la imagen de abajo.

```
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) Un archive *vlc.ko* sera genrado, ahora tenemos que enfocarnos en 2 archivos el *load_driver_server.sh* and the *load_driver_client.sh*. Escribir la instrucción: *nano load_driver_server.sh* un cuadro deberá aparecer como en la fotografía.



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

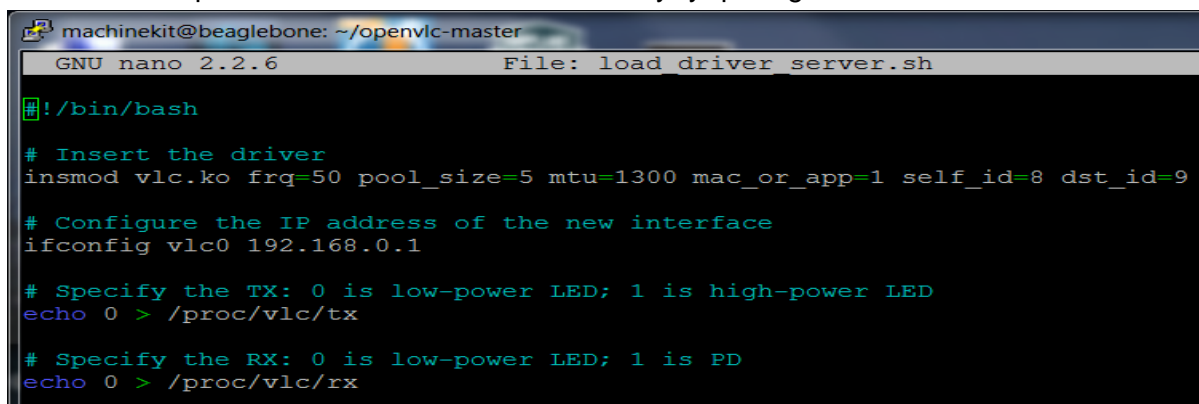
#!/bin/bash
# Insert the driver
insmod vlc.ko frq=50 pool_size=5 mtu=1300 mac_or_app=1 self_id=8 dst_id=9

# 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
```

Identifique la palabra *echo*, al lado de esta tiene el valor de 1, cambiarlo por el valor de 0. A continuación presionara las teclas “control” + “x” y “y” para guardar los cambios.



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

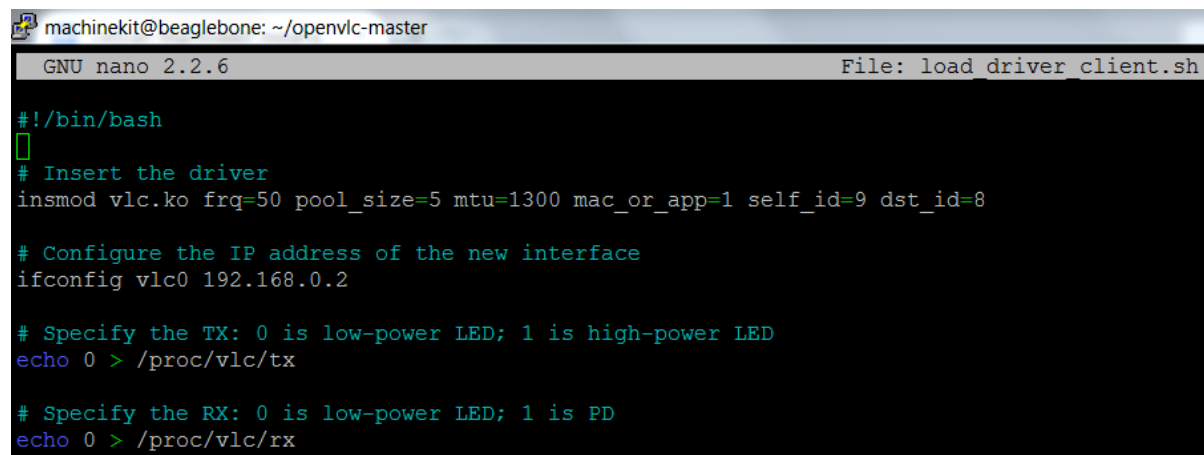
#!/bin/bash
# Insert the driver
insmod vlc.ko frq=50 pool_size=5 mtu=1300 mac_or_app=1 self_id=8 dst_id=9

# 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) Hacer los mismos pasos que en 12 para el *load_driver_client.sh*. El *machinekit@beaglebone* debería verse como la siguiente imagen.



```
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) Escribir la instrucción `ls -l`, usted vera que el `load_driver_server.sh` y el `load_driver_client.sh` no son ejecutables. Para poder hacer esto posible escribir la siguiente instrucción: `chmod u+x load_driver_server.sh` presione *enter* y a continuación escriba: `chmod u+x load_driver_client.sh`. Revisar si ambos `.sh` ahora son ejecutables.

```
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
```

15) Finalmente para verificar la correcta instalación se realizará una pequeña prueba con 2 BBB, La persona A deberá escribir la siguiente instrucción: `sudo ./load_driver_client.sh` y la persona B: `sudo ./load_driver_server.sh`. Después de eso la persona A deberá escribir la instrucción: `ping 192.168.0.1` y la persona B deberá escribir la instrucción: `ping 192.168.0.2` (Asegúrese de que los leds de cada BBB estén conectados). Apenas se introduzca la instrucción al `machinekit@beaglebone` el led comenzara a encender y apagarse. Colocar ambos BBB de frente y observar los valores que se despliegan en el `machinekit@beaglebone`. **(Para detener el programa presione 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
```