BB Black Expansion Board User Manual

Copyright

Revision History

Revision	Date	Description
V1.0	2014.02.07	Initial revision
V2.0	2014.11.20	Major update
V3.0	2015.04.08	Add Debian IMG demo

Contents

1 Getting Ready

1.1 Writing the TF Card System Image

Follow the steps below to write the Angstrom into TF card:

1) Extract the system image

Extract the system image file .img.7z by using archiver software like 7z920.exe.

Note: please download the image file from:

LCD CPAE(4.3inch) image for testing

http://www.waveshare.com/wiki/index.php?title=LCD-CAPE-4.3inch-Image

LCD CPAE(7inch) image for testing

http://www.waveshare.com/wiki/index.php?title=LCD-CAPE-7inch-Image

MISC CAPE and RS485/CAN CAPE image for testing

http://www.waveshare.com/wiki/index.php?title=MISC_CAPE_Image

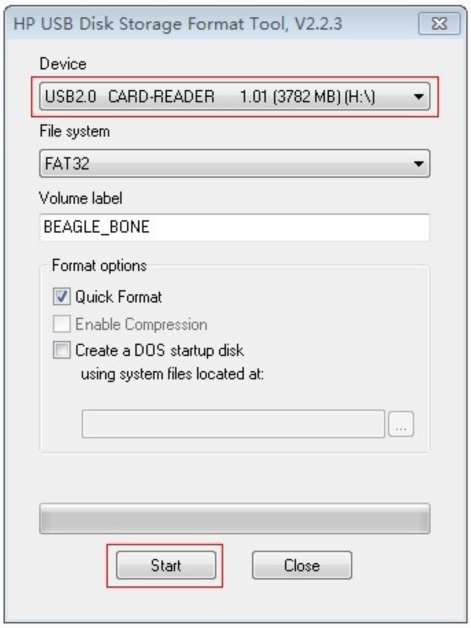
NOTE: MISC CAPE and RS485/CAN CAPE share an IMG file.

2) Format the TF card

Use **HPUSBDisk.exe** to format the TF card.

Choose the Device as your TF card, File system as FAT32. Then click Start.

Note: the TF card capacity should be 4GB or above!



HP USB Disk Storage Format Tool, V2.2.3

Device: USB2.0 CARD-READER 1.01 (3782 MB) (H:\)

File system: FAT32

Volume label: BEAGLE_BONE

Format options

☑ Quick Format

□ Enable Compression

□ Create a DOS startup disk using system files located at:

Figure 1: Use HPUSBDisk.exe to format the TF card

3) Writing the system image

   Launch Win32DiskImager.exe, select the extracted system image. Then click write.

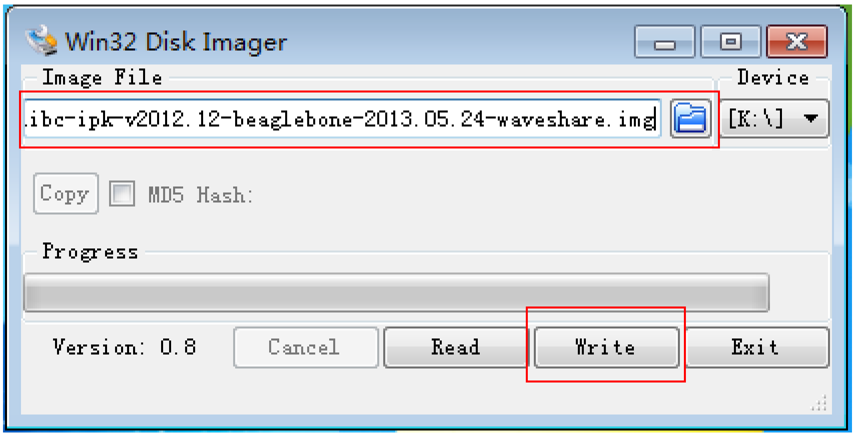
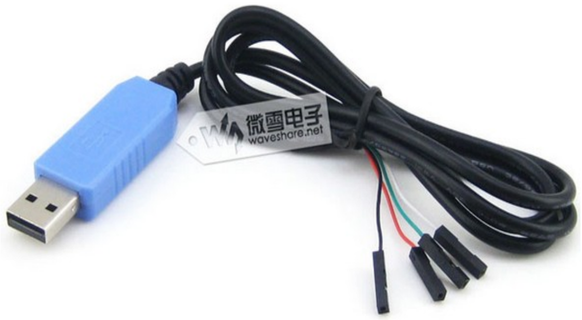


Figure 2: Use Win32DiskImager.exe to write the system image

1.2 Install USB to UART driver

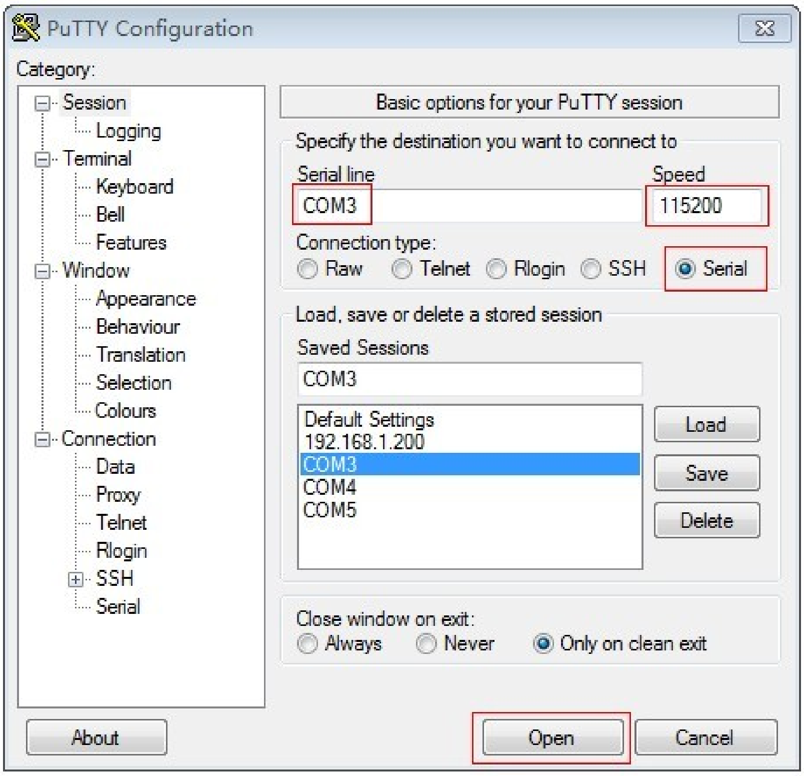
1) Connect the LCD CAPE with USB TO UART interface to the PC through a USB cable. **Note: You need to install the corresponding driver. Please refer to the respective manuals of your USB TO UART module.**



Interface definition:
Red: VCC
Black: GND
Green: TXD(connect to RXD)
White: RXD(connect to TXD)

Figure 3: USB to UART Cable

2) Open **PL2303_Prolific_DriverInstaller_v1.8.0.exe** and install the driver.
3) Launch putty.exe, configure as follows, then click Open.



Specify the destination you want to connect to

Serial line: COM3

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions: COM3

Default Settings
192.168.1.200
COM3
COM4
COM5

Close window on exit: ○ Always ○ Never ● Only on clean exit

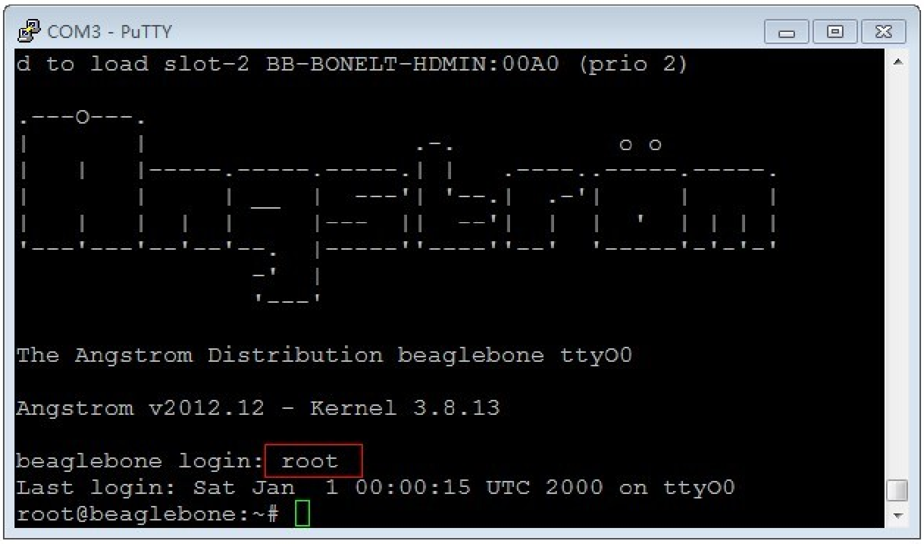
Figure 4: PuTTY Settings

Note:

- Serial line: check the PC "Device Manager" to confirm which COM port should be selected.
- Speed: 115200。
- Connection type: Serial。

1.3 Enter Bash Shell

1) Insert the TF card into the BeagleBone Black onboard slot, keep pressing down the BOOT button, power up the board, then release the BOOT button.
2) When the system startups, input "root" to enter Bash Shell environment, now the shell commands are available to use. All the commands follows are executed here.



```
d to load slot-2 BB-BONELT-HDMIN:00A0 (prio 2)

.---O---.
|       |                  .-.           o o
|   |   |-----.-----.-----.| |   .----..-----.-----.
|       |     | __  |  ---'| '--.|  .-'|     |     |
|   |   |  |  |     |---  ||  --'|  |  |  '  | | | |
'---'---'--'--'--.  |-----''----''--'  '-----'-'-'-'
                -'  |
                '---'

The Angstrom Distribution beaglebone ttyO0

Angstrom v2012.12 - Kernel 3.8.13

beaglebone login: root
Last login: Sat Jan  1 00:00:15 UTC 2000 on ttyO0
root@beaglebone:~#
```

Figure 5: Enter Bash Shell

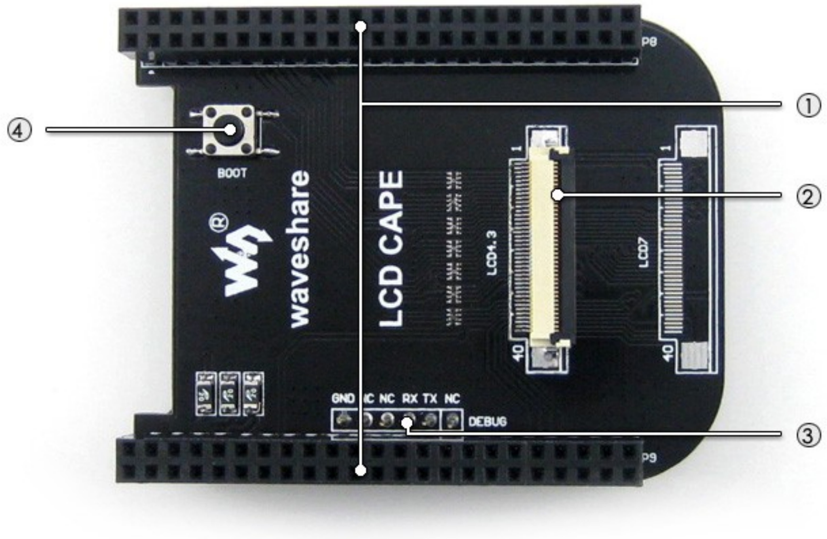
Note: please make sure that you are using the TF card image that we provide, and the system should boot from TF card (keep pressing down BOOT button, then power up), otherwise, the testing will fail.

1.4 API Source Code

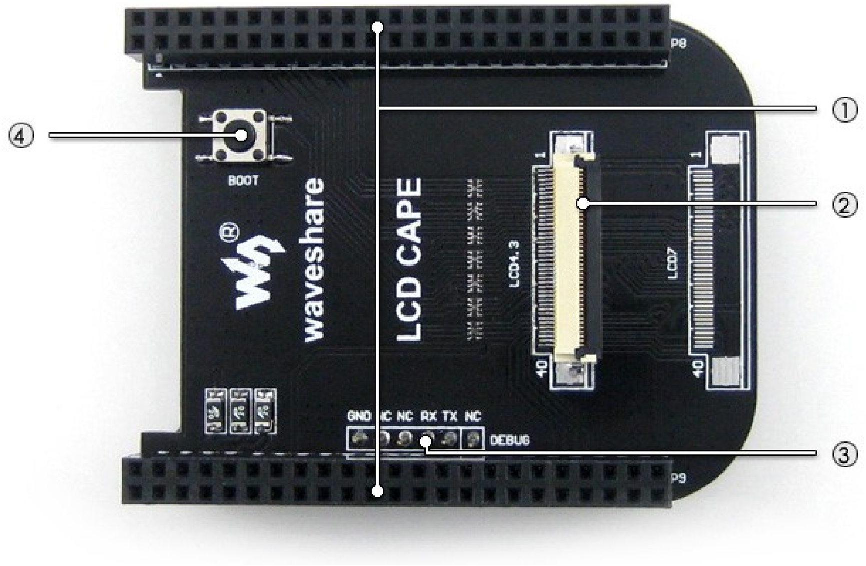
The API source code can be found on **/home/xuser/waveshare_demo/API**.

2 LCD CAPE

2.1 LCD Display Overview



1. **BB_BLACK connector :** for connecting BB Black
2. **4.3inch LCD interface :** for connecting 4.3inch resistive touchscreen LCD
3. **DEBUG interface :** BB Black debug interface, for connecting serial modules
4. **BOOT selection button:** Boot from TF card



1. **BB_BLACK connector : for connecting BB Black**
2. **7inch LCD interface : for connecting 7inch resistive touchscreen LCD**
3. **DEBUG interface : BB Black debug interface, for connecting serial modules**
4. **BOOT selection button : Boot from TF card**

2.2 LCD Display

1) Connect to a LCD

   Note: There are two models of LCD CAPE, LCD CAPE (4.3inch) and LCD CAPE (7inch), each one corresponds our 4.3inch or 7inch resistive touch screen respectively. Some batches of LCD CAPE provide both interfaces. If the CAPE connected to the LCD by a wrong interface, it may damage the LCD and the main board.

   - When using 4.3inch LCD
     Download **LCD CPAE(4.3inch) image for testing**, please refer to **1.1**.
     Connect BB Black to the 4.3inch LCD.

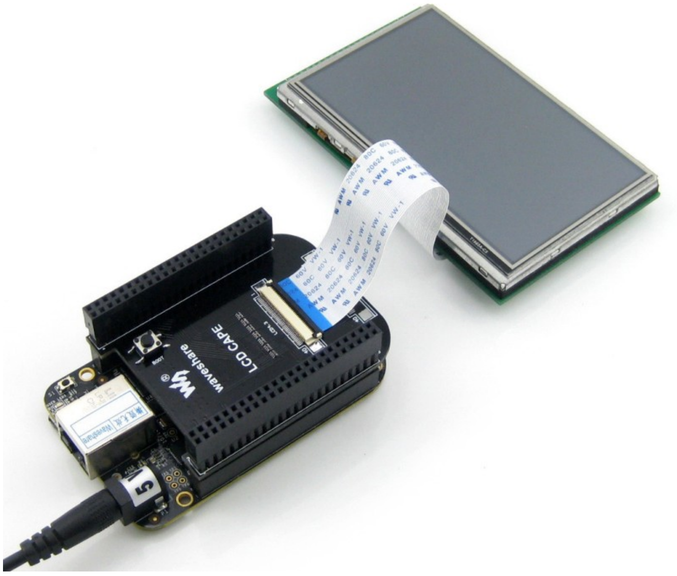


Figure 6: Connect to the 4.3inch LCD

   - When using 7inch LCD
     Download **LCD CPAE(7inch) image for testing**, please refer to **1.1**.
     Connect BB Black to the 7inch LCD.

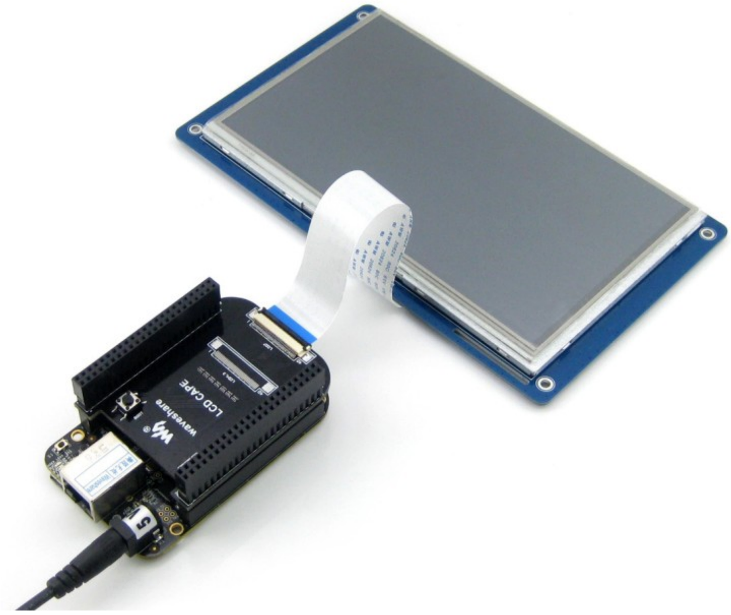


Figure 7: Connect to the 7inch LCD

2) The touchscreen should be calibrated when the display mode has been changed:

Angstrom:

```
root@beaglebone:~# rm -rf /etc/pointercal*
root@beaglebone:~# ts_calibrate
root@beaglebone:~# sync
```

Reboot the system.

Note: if the calibration failed, reboot and retry again.

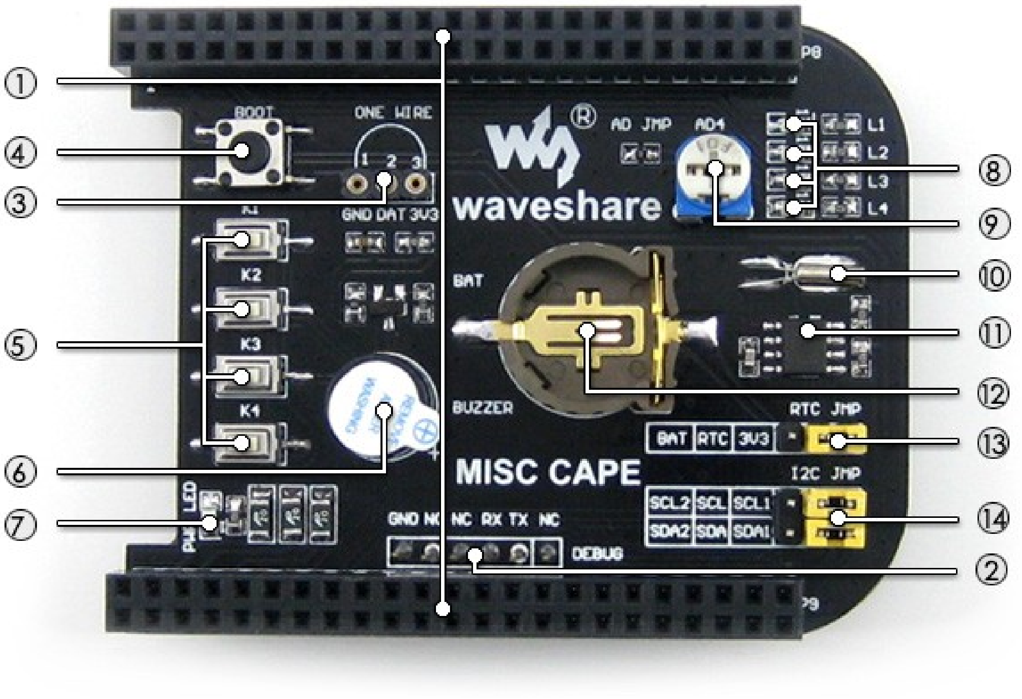
Debian:

```
root@beaglebone:~# rm -rf /etc/pointercal*
root@beaglebone:~# sync
```

Reboot the system, the calibration procedure will boot automatically, the touch function can be used properly after the calibration.

3 MISC CAPE

3.1 MISC CAPE Overview



1. **BB_BLACK connector :** for connecting BB Black
2. **DEBUG interface :** BB Black debug interface, for connecting serial modules
3. **ONE-WIRE interface:** easily connects to ONE-WIRE devices (TO-92 package), such as temperature sensor (DS18B20), electronic registration number (DS2401), etc.
4. **BOOT selection button :** Boot from TF card
5. **User button :** 4 buttons
6. **Buzzer**
7. **Power indicator**
8. **User LED :** 4 LEDs
9. **Potentiometer :** AD adjustable potentiometer
10. **32.768KHz crystal :** for RTC
11. **PCF8563 :** RTC
12. **RTC battery holder :** for 3.3V battery
13. **RTC power selection jumper**
14. **RTC I2C selection jumper:** select I2C1 OR I2C2

3.2 LED

Short the LED jumper, enter:

```
root@beaglebone:~# test_led
```

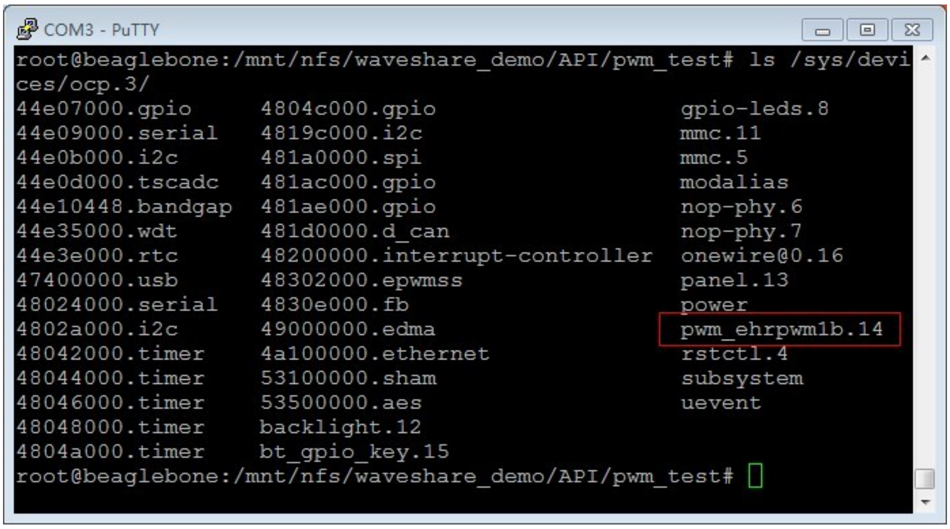
The 4 LEDs will light up one by one, press Ctrl+C to exit.

3.3 Buzzer

1) Short the buzzer jumper, enter:

   ```
   root@beaglebone:~# ls /sys/devices/ocp.3/
   ```

   You'll find the “pwm_ehrpwm1b.14” as shown:



```
root@beaglebone:/mnt/nfs/waveshare_demo/API/pwm_test# ls /sys/devi
ces/ocp.3/
44e07000.gpio      4804c000.gpio                  gpio-leds.8
44e09000.serial    4819c000.i2c                   mmc.11
44e0b000.i2c       481a0000.spi                   mmc.5
44e0d000.tscadc    481ac000.gpio                  modalias
44e10448.bandgap   481ae000.gpio                  nop-phy.6
44e35000.wdt       481d0000.d_can                 nop-phy.7
44e3e000.rtc       48200000.interrupt-controller  onewire@0.16
47400000.usb       48302000.epwmss                panel.13
48024000.serial    4830e000.fb                    power
4802a000.i2c       49000000.edma                  pwm_ehrpwm1b.14
48042000.timer     4a100000.ethernet              rstctl.4
48044000.timer     53100000.sham                  subsystem
48046000.timer     53500000.aes                   uevent
48048000.timer     backlight.12
4804a000.timer     bt_gpio_key.15
root@beaglebone:/mnt/nfs/waveshare_demo/API/pwm_test#
```

Figure 8: Find the “pwm_ehrpwm1b.14”

2) Because the pwm_ehrpwm1b.14's extension is .14. It is needed to add parameter 14 when buzzer tested.

   ```
   root@beaglebone:~# test_pwm 14
   ```

   The buzzer will make sounds in different frequency.

3.4 DS18B20

1) Insert the DS18B20 into the 1-WIRE socket, short the 1-WIRE jumper, enter:

   ```
   root@beaglebone:~#ls /sys/bus/w1/devices/
   ```

   You will find the "28-00000 57c5948" (the last 7 characters are unique for each DS18B20, it depends)

2) Execute:

   ```
   root@beaglebone:~#test_ds18b20   57c5948
   ```

   The terminal will print the current temperature.

3.5 Buttons

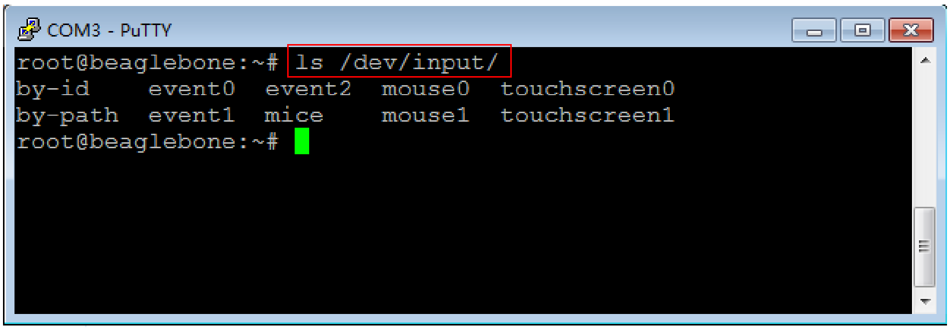
1) Execute

   root@beaglebone:~# test_key event2

   Note:

   Device file "event2" is based on user's device, Not necessarily "event2". you can execute the following command to view:

   root@beaglebone:~# ls /dev/input



```
root@beaglebone:~# ls /dev/input/
by-id     event0  event2  mouse0  touchscreen0
by-path   event1  mice    mouse1  touchscreen1
root@beaglebone:~#
```

Figure 9: Check event device

2) Press any of the onboard buttons, the terminal will show which one has been pressed, press Ctrl+C to exit.

   Note: if the LCD screen was connected while testing buttons, the LCD screen seems that it was touched at the same time, it's normal.

3.6 RTC

Confirm that the module is powered by onboard battery.

1) Show system date time:

   root@beaglebone:~# date

2) Set system date time:

   root@beaglebone:~# date 020809302014.23

3) Set the hardware clock of RTC module:

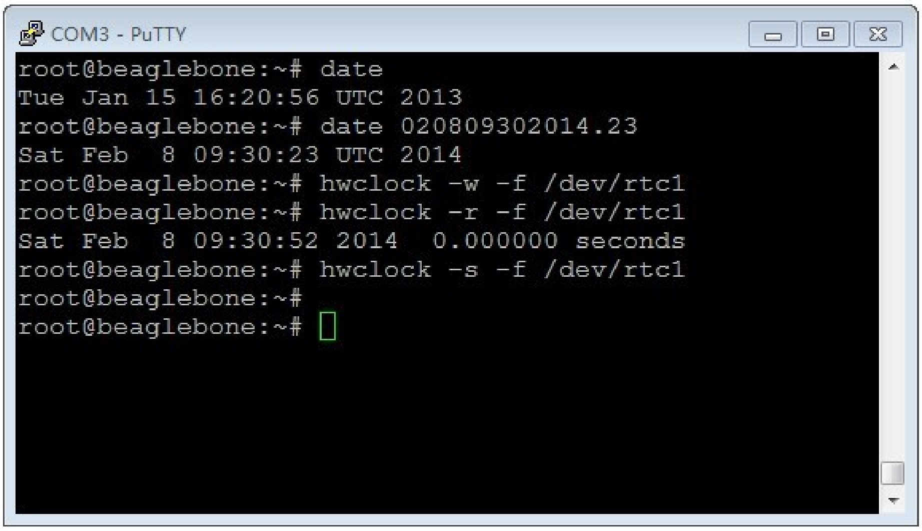
   root@beaglebone:~# hwclock –w –f /dev/rtc1

4) Show the hardware clock of RTC module:

   root@beaglebone:~# hwclock –r –f /dev/rtc1

5) Synchronize the hardware clock to system date time:

   root@beaglebone:~# hwclock –s –f /dev/rtc1

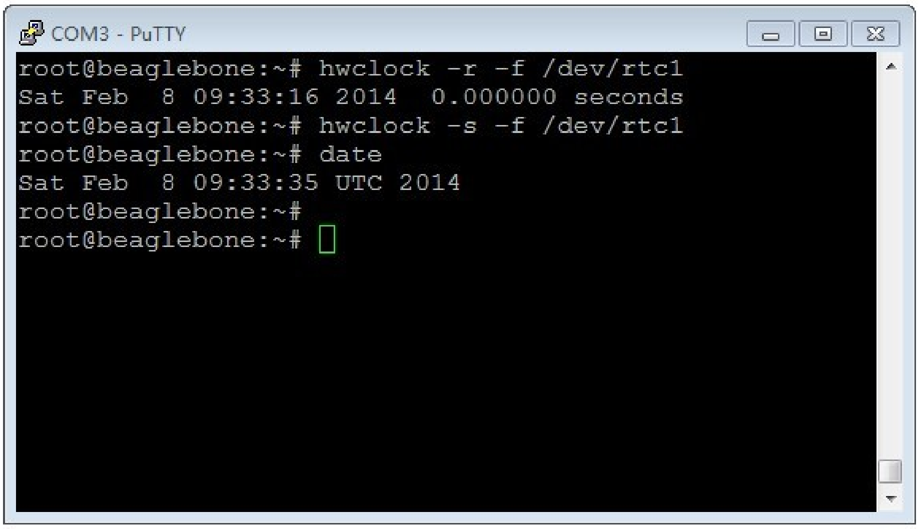


```
root@beaglebone:~# date
Tue Jan 15 16:20:56 UTC 2013
root@beaglebone:~# date 020809302014.23
Sat Feb  8 09:30:23 UTC 2014
root@beaglebone:~# hwclock -w -f /dev/rtc1
root@beaglebone:~# hwclock -r -f /dev/rtc1
Sat Feb  8 09:30:52 2014  0.000000 seconds
root@beaglebone:~# hwclock -s -f /dev/rtc1
root@beaglebone:~#
root@beaglebone:~# 
```

Figure 10: Synchronize the hardware clock to system date time

6) Power off and reboot, read the hardware clock of RTC module and synchronize to system date time:

```
root@beaglebone:~# hwclock –r –f /dev/rtc1
root@beaglebone:~# hwclock –s –f /dev/rtc1
root@beaglebone:~# date
```



```
root@beaglebone:~# hwclock -r -f /dev/rtc1
Sat Feb  8 09:33:16 2014  0.000000 seconds
root@beaglebone:~# hwclock -s -f /dev/rtc1
root@beaglebone:~# date
Sat Feb  8 09:33:35 UTC 2014
root@beaglebone:~#
root@beaglebone:~# 
```

Figure 11: Synchronize to system date time

Now the date time of software and hardware are synchronous.

4 RS485 CAN CAPE

4.1 CAN

Two BeagleBone Black and two RS485/CAN CAPEs are required for this testing. Set jumper to enable UART1(RXD1, TXD1). Connect two CAN Board to CAN1 interface separately, connect the CANH, CANL of one module to the CANH, CANL of another module via jumper wires.

1) Configure the baud rate:

```
root@beaglebone:~#canconfig can0 bitrate 115200 ctrlmode triple-sampling on
```

2) Enable the CAN device:

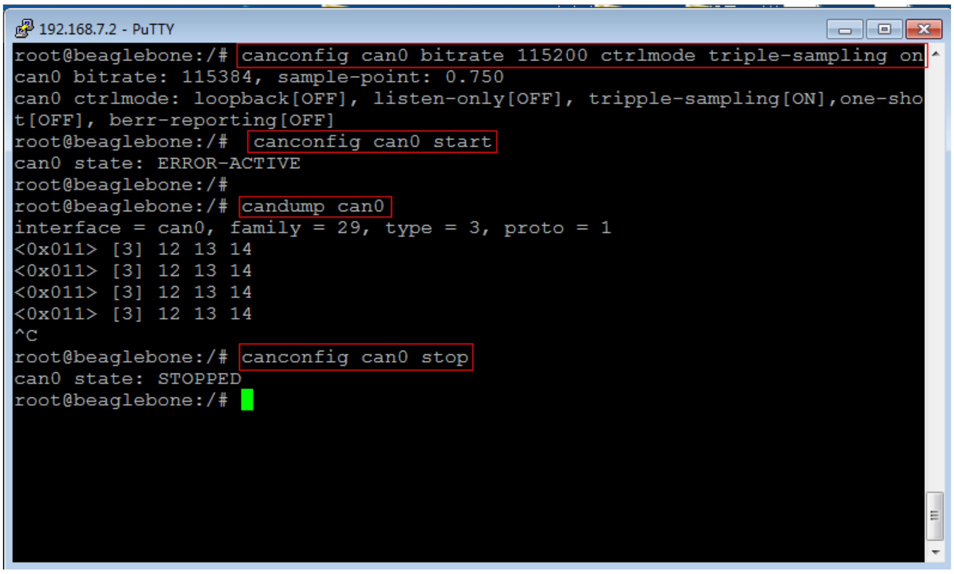
```
root@beaglebone:~# canconfig can0 start
```

The two CAN devices act as receiver and sender separately. The receiver gets ready to receive data first, and then the sender starts sending data.

3) Receiver:

```
root@beaglebone:~# candump can0
```

As shown in the picture:



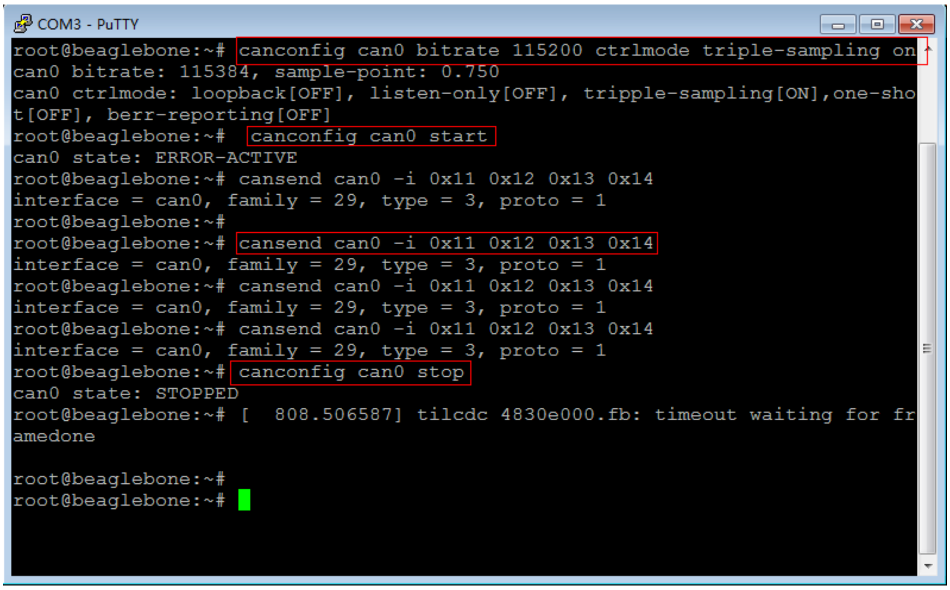
```
root@beaglebone:/# canconfig can0 bitrate 115200 ctrlmode triple-sampling on
can0 bitrate: 115384, sample-point: 0.750
can0 ctrlmode: loopback[OFF], listen-only[OFF], tripple-sampling[ON],one-sho
t[OFF], berr-reporting[OFF]
root@beaglebone:/#  canconfig can0 start
can0 state: ERROR-ACTIVE
root@beaglebone:/#
root@beaglebone:/# candump can0
interface = can0, family = 29, type = 3, proto = 1
<0x011> [3] 12 13 14
<0x011> [3] 12 13 14
<0x011> [3] 12 13 14
<0x011> [3] 12 13 14
^C
root@beaglebone:/# canconfig can0 stop
can0 state: STOPPED
root@beaglebone:/#
```

Figure 12: CAN receiver status

4) Sender:

```
root@beaglebone:~# cansend can0 -i 0x11 0x12 0x13 0x14
```

As shown in the picture:



```
root@beaglebone:~# canconfig can0 bitrate 115200 ctrlmode triple-sampling on
can0 bitrate: 115384, sample-point: 0.750
can0 ctrlmode: loopback[OFF], listen-only[OFF], tripple-sampling[ON],one-sho
t[OFF], berr-reporting[OFF]
root@beaglebone:~#  canconfig can0 start
can0 state: ERROR-ACTIVE
root@beaglebone:~# cansend can0 -i 0x11 0x12 0x13 0x14
interface = can0, family = 29, type = 3, proto = 1
root@beaglebone:~#
root@beaglebone:~# cansend can0 -i 0x11 0x12 0x13 0x14
interface = can0, family = 29, type = 3, proto = 1
root@beaglebone:~# cansend can0 -i 0x11 0x12 0x13 0x14
interface = can0, family = 29, type = 3, proto = 1
root@beaglebone:~# cansend can0 -i 0x11 0x12 0x13 0x14
interface = can0, family = 29, type = 3, proto = 1
root@beaglebone:~# canconfig can0 stop
can0 state: STOPPED
root@beaglebone:~# [  808.506587] tilcdc 4830e000.fb: timeout waiting for fr
amedone

root@beaglebone:~#
root@beaglebone:~#
```

Figure 13: CAN wender status

5) Stop the devices:

root@beaglebone:~# canconfig can0 stop

4.2 RS485

Two BeagleBone Black and two RS485/CAN CAPEs are required for this testing. Set jumper to enable UART2(RXD2, TXD2). Connect two RS485 Boards to UART2 interface separately. Connect the A, B of one module to the A, B of another module via jumper wires.

The two RS485 devices act as receiver and sender separately.

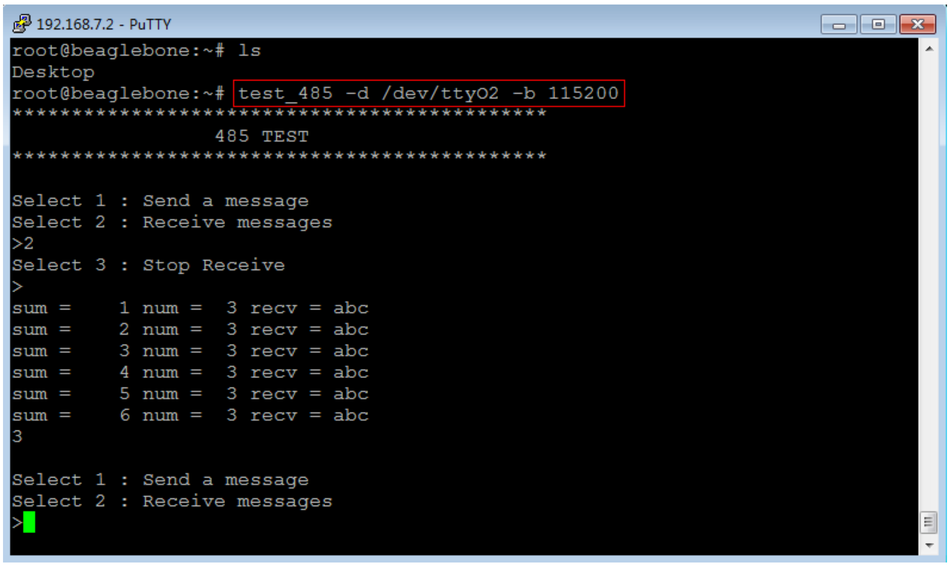
1) The receiver gets ready to receive data first, and then the sender starts sending data. Execute:

root@beaglebone:~# test_485 –d /dev/ttyO2 –b 115200

2) Receiver:

Select "2", it will keep receiving data until you select "3".

As shown in the picture:



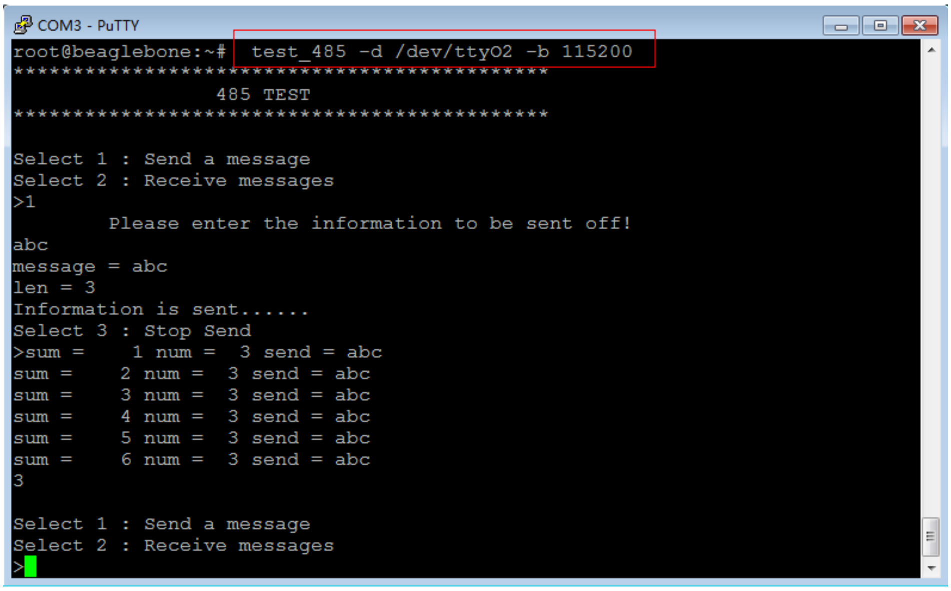
```
root@beaglebone:~# ls
Desktop
root@beaglebone:~# test_485 -d /dev/ttyO2 -b 115200
*********************************************
                 485 TEST
*********************************************

Select 1 : Send a message
Select 2 : Receive messages
>2
Select 3 : Stop Receive
>
sum =     1 num =  3 recv = abc
sum =     2 num =  3 recv = abc
sum =     3 num =  3 recv = abc
sum =     4 num =  3 recv = abc
sum =     5 num =  3 recv = abc
sum =     6 num =  3 recv = abc
3

Select 1 : Send a message
Select 2 : Receive messages
>
```

Figure 14: RS485 receiver status

3) Sender:

Select "1", and then enter the message, say, "abc", it loops sending the message until you select "3".

As shown in the picture:



```
root@beaglebone:~# test_485 -d /dev/ttyO2 -b 115200
*********************************************
                 485 TEST
*********************************************

Select 1 : Send a message
Select 2 : Receive messages
>1
        Please enter the information to be sent off!
abc
message = abc
len = 3
Information is sent......
Select 3 : Stop Send
>sum =    1 num =  3 send = abc
sum =    2 num =  3 send = abc
sum =    3 num =  3 send = abc
sum =    4 num =  3 send = abc
sum =    5 num =  3 send = abc
sum =    6 num =  3 send = abc
3

Select 1 : Send a message
Select 2 : Receive messages
>
```

Figure 15: RS485 sender status

4) Press Ctrl+C to exit.

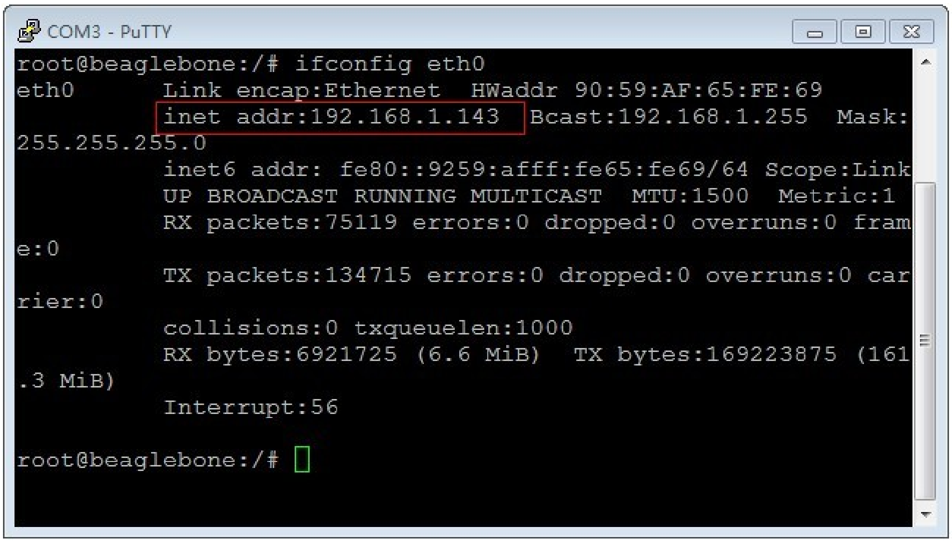
5 Other Expansion Module

5.1 USB Camera

1) Connect the USB Camera to the BeagleBone Black USB Host connector.

2) Check the assigned IP:

root@beaglebone:~# ifconfig eth0



```
root@beaglebone:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 90:59:AF:65:FE:69
          inet addr:192.168.1.143  Bcast:192.168.1.255  Mask:
255.255.255.0
          inet6 addr: fe80::9259:afff:fe65:fe69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:75119 errors:0 dropped:0 overruns:0 fram
e:0
          TX packets:134715 errors:0 dropped:0 overruns:0 car
rier:0
          collisions:0 txqueuelen:1000
          RX bytes:6921725 (6.6 MiB)  TX bytes:169223875 (161
.3 MiB)
          Interrupt:56

root@beaglebone:/# 
```

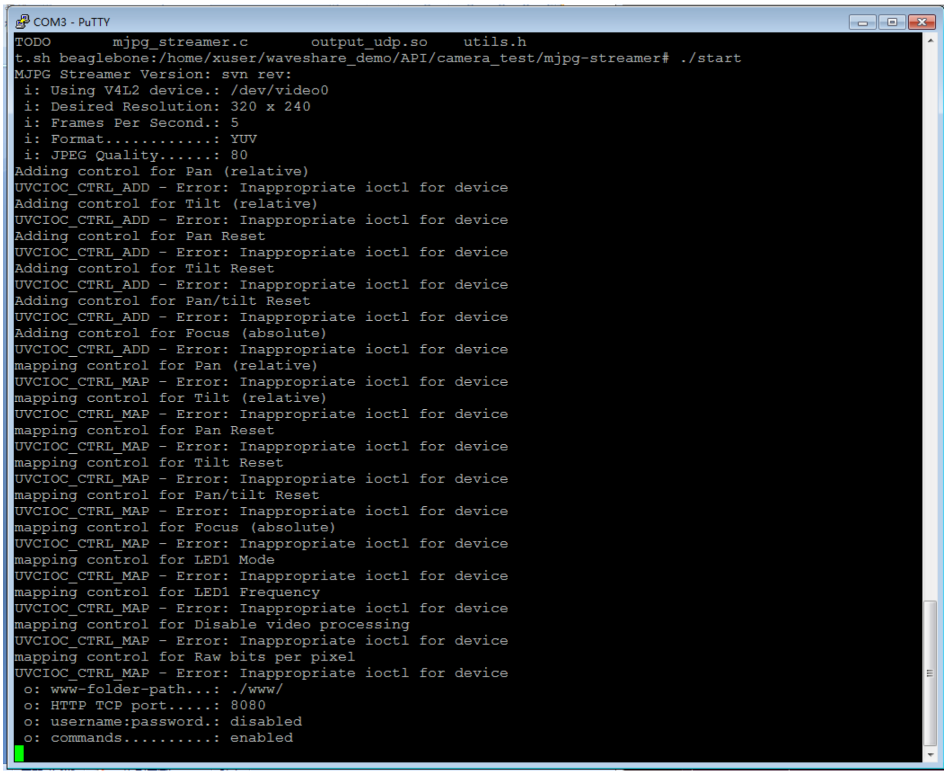
Figure 16: Check the assigned IP

As shown in the picture, the IP is 192.168.1.143. Note this IP(it depends).

3) Start up the video stream server:

root@beaglebone:~# cd/home/xuser/waveshare_demo/API/camera_test/mjpg-streamer

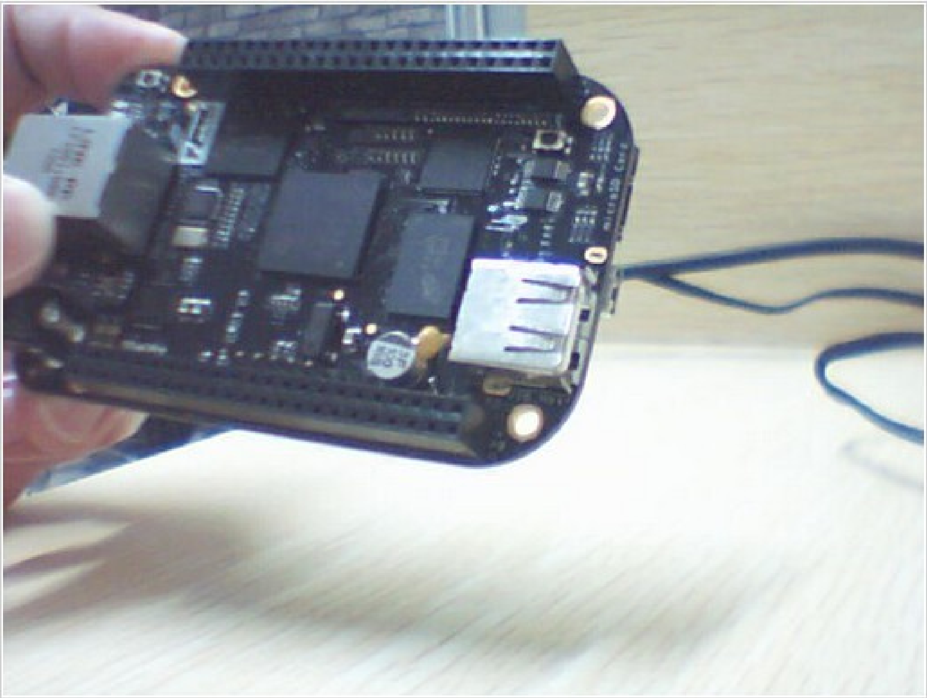
```
root@beaglebone:~# ./ start.sh
```



```
TODO       mjpg_streamer.c       output_udp.so    utils.h
t.sh beaglebone:/home/xuser/waveshare_demo/API/camera_test/mjpg-streamer# ./start
MJPG Streamer Version: svn rev:
 i: Using V4L2 device.: /dev/video0
 i: Desired Resolution: 320 x 240
 i: Frames Per Second.: 5
 i: Format............: YUV
 i: JPEG Quality......: 80
Adding control for Pan (relative)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Tilt (relative)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Pan Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Tilt Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Pan/tilt Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Focus (absolute)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
mapping control for Pan (relative)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Tilt (relative)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Pan Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Tilt Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Pan/tilt Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Focus (absolute)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for LED1 Mode
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for LED1 Frequency
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Disable video processing
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Raw bits per pixel
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
 o: www-folder-path...: ./www/
 o: HTTP TCP port.....: 8080
 o: username:password.: disabled
 o: commands..........: enabled
```

Figure 17: Start up the video stream server

4) Open browser on the PC in the same local network, visit the following address to view the video (default port number is 8080):
http://192.168.1.143:8080/javascript.html
(It depends on the IP you have noted)



© The MJPG-streamer team | Design by Andreas Viklund

Figure 18: Visit video stream server

5) Press Ctrl+C to exit.

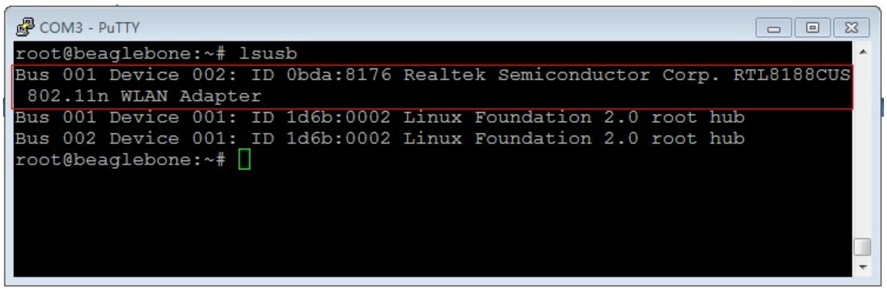
5.2 USB WIFI

1) Config the wireless network:

a) Power off, connect the USB WIFI module to the BeagleBone Black USB Host connector, power up again.

b) Check the USB status:

root@beaglebone:~# lsusb

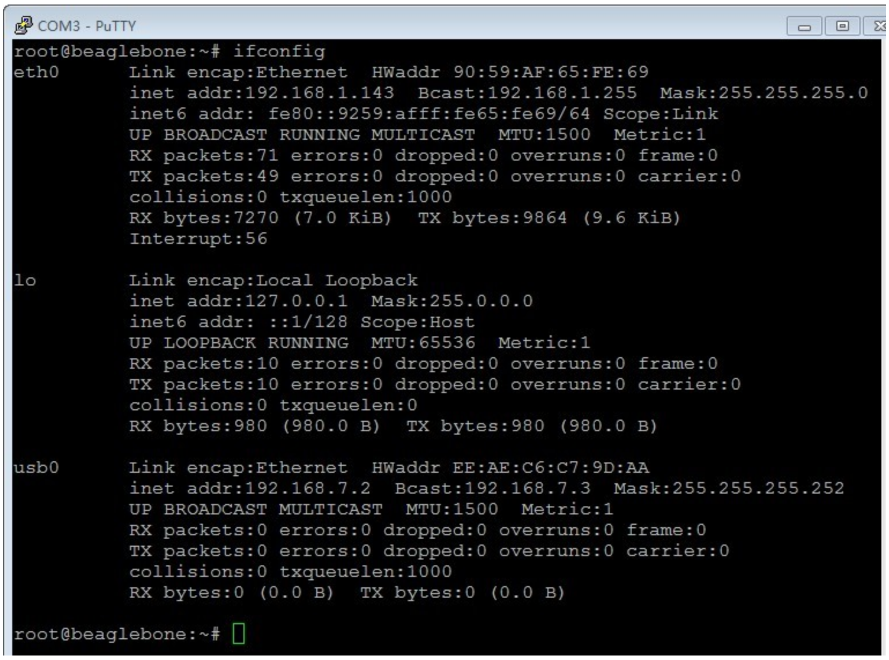


```
root@beaglebone:~# lsusb
Bus 001 Device 002: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS
 802.11n WLAN Adapter
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@beaglebone:~#
```

Figure 19: Check the USB status

c) Check the network status:

root@beaglebone:~# ifconfig



```
root@beaglebone:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 90:59:AF:65:FE:69
          inet addr:192.168.1.143  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::9259:afff:fe65:fe69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:71 errors:0 dropped:0 overruns:0 frame:0
          TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7270 (7.0 KiB)  TX bytes:9864 (9.6 KiB)
          Interrupt:56

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:980 (980.0 B)  TX bytes:980 (980.0 B)

usb0      Link encap:Ethernet  HWaddr EE:AE:C6:C7:9D:AA
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@beaglebone:~#
```

Figure 20: Check the network status

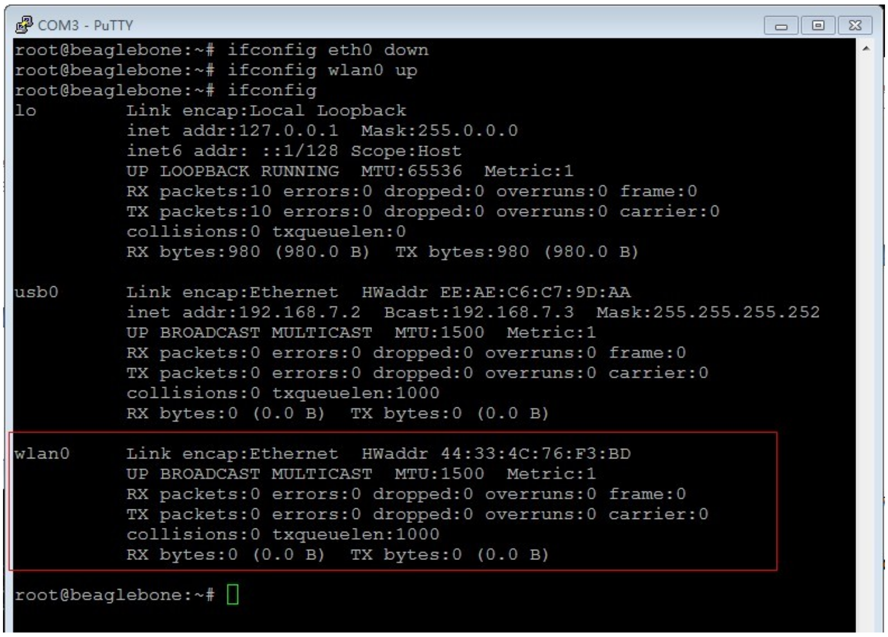
d) Shut down the Ethernet NIC, start up the WIFI NIC:

root@beaglebone:~# ifconfig eth0 down

root@beaglebone:~# ifconfig wlan0 up

e) Check the WIFI status:

root@beaglebone:~# ifconfig



```
root@beaglebone:~# ifconfig eth0 down
root@beaglebone:~# ifconfig wlan0 up
root@beaglebone:~# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:980 (980.0 B)  TX bytes:980 (980.0 B)

usb0      Link encap:Ethernet  HWaddr EE:AE:C6:C7:9D:AA
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr 44:33:4C:76:F3:BD
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@beaglebone:~#
```

Figure 21. Check the WIFI status

There's no AP connected, therefore, the wlan0 RX TX packets are both 0.

f) Config the wlan0 IP:

```
root@beaglebone:~# ifconfig wlan0 192.168.2.107
```

g) Config the gateway:

```
root@beaglebone:~# route add default gw 192.168.2.1
```

h) Config the DNS:

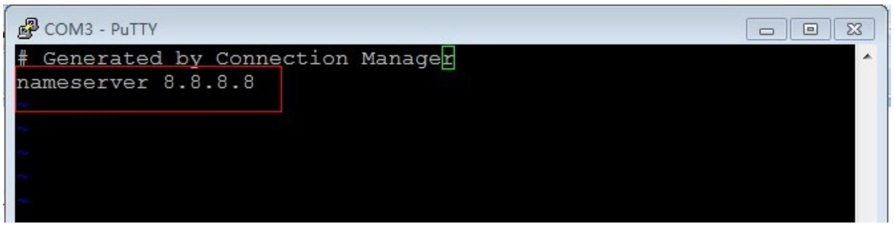
```
root@beaglebone:~# vi /etc/resolv.conf
```

Edit

```
nameserver 127.0.0.1
```

As:

```
nameserver 8.8.8.8
```



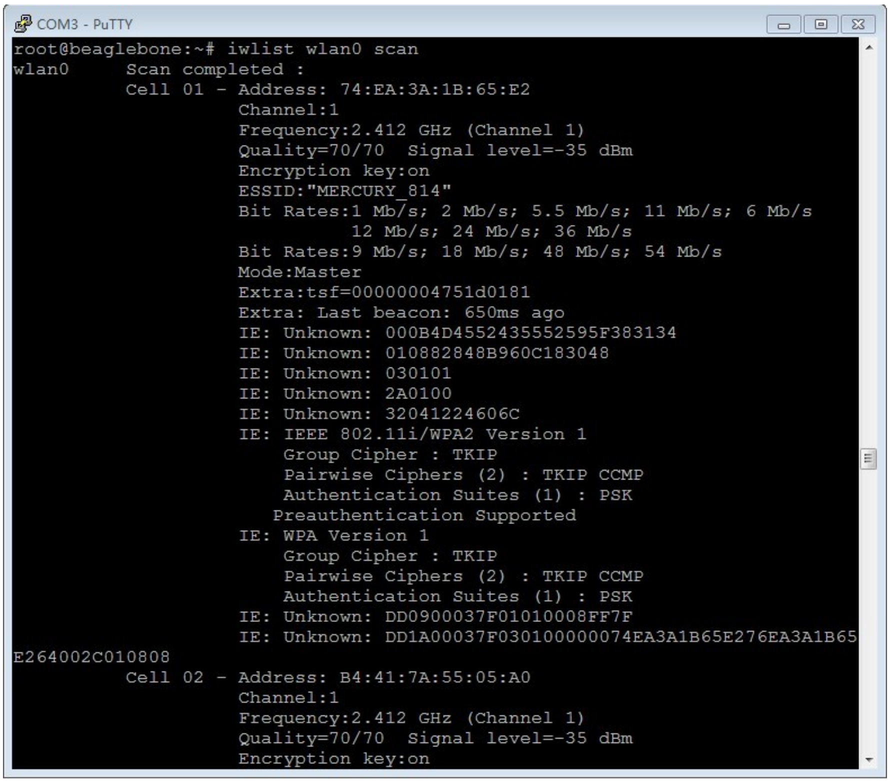
```
# Generated by Connection Manager
nameserver 8.8.8.8
~
~
~
~
~
```

Figure 22: Configure the DNS

Save and exit.

i) Scan wireless router:

```
root@beaglebone:~# iwlist wlan0 scan
```



```
root@beaglebone:~# iwlist wlan0 scan
wlan0     Scan completed :
          Cell 01 - Address: 74:EA:3A:1B:65:E2
                    Channel:1
                    Frequency:2.412 GHz (Channel 1)
                    Quality=70/70  Signal level=-35 dBm
                    Encryption key:on
                    ESSID:"MERCURY_814"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 6 Mb/s
                              12 Mb/s; 24 Mb/s; 36 Mb/s
                    Bit Rates:9 Mb/s; 18 Mb/s; 48 Mb/s; 54 Mb/s
                    Mode:Master
                    Extra:tsf=00000004751d0181
                    Extra: Last beacon: 650ms ago
                    IE: Unknown: 000B4D4552435552595F383134
                    IE: Unknown: 010882848B960C183048
                    IE: Unknown: 030101
                    IE: Unknown: 2A0100
                    IE: Unknown: 32041224606C
                    IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (2) : TKIP CCMP
                        Authentication Suites (1) : PSK
                       Preauthentication Supported
                    IE: WPA Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (2) : TKIP CCMP
                        Authentication Suites (1) : PSK
                    IE: Unknown: DD0900037F01010008FF7F
                    IE: Unknown: DD1A00037F030100000074EA3A1B65E276EA3A1B65
E264002C010808
          Cell 02 - Address: B4:41:7A:55:05:A0
                    Channel:1
                    Frequency:2.412 GHz (Channel 1)
                    Quality=70/70  Signal level=-35 dBm
                    Encryption key:on
```

Figure 23. Scan wireless router

2) Connect to the router:

a) Config the key file /etc/wpa_supplicant.conf:

```
root@beaglebone:~# vi /etc/wpa_supplicant.conf
```

Edit:

```
network={
          key_mgmt=NONE
}
```

As:

```
network={
          ssid="waveshare"
          psk="12345678"
}
```

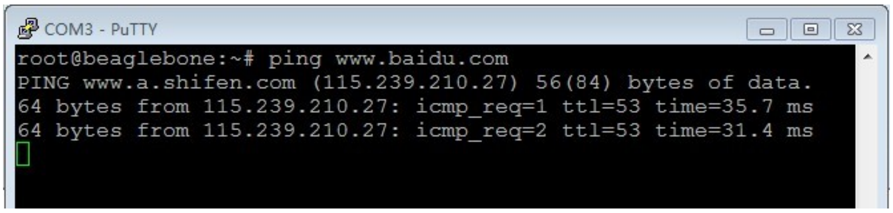
Save and exit.

b) Connect manually:

```
root@beaglebone:~# wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant.conf
```

c) Test the connection:

```
root@beaglebone:~# ping www.baidu.com
```



```
root@beaglebone:~# ping www.baidu.com
PING www.a.shifen.com (115.239.210.27) 56(84) bytes of data.
64 bytes from 115.239.210.27: icmp_req=1 ttl=53 time=35.7 ms
64 bytes from 115.239.210.27: icmp_req=2 ttl=53 time=31.4 ms
```

Figure 24. Test WIFI connection: