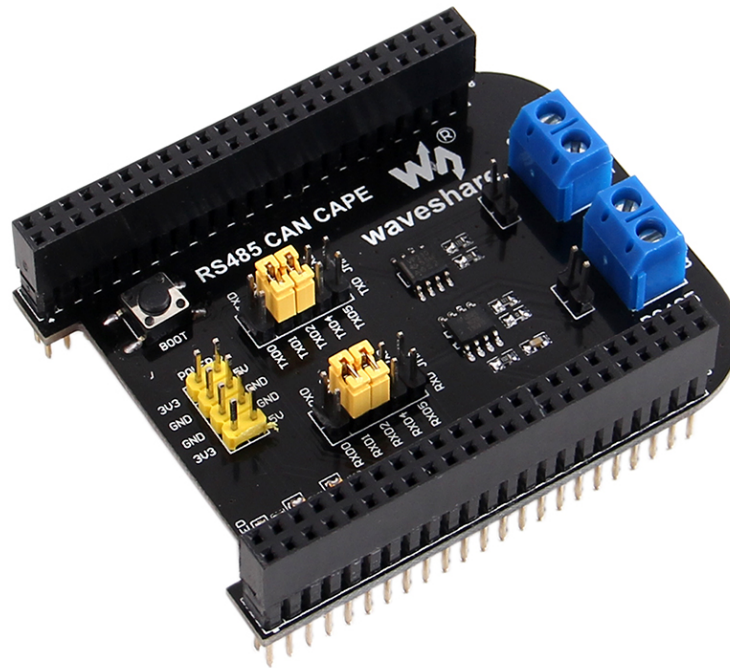


BEAGLEBONE RS485 CAN CAPE EXPANSION BOARD

User Manual



Getting Ready

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

- [Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.12-beaglebone-2013.05.24-LCD-cape-LCD4.3-v1.1.img.7z](#)

LCD CPAE(7inch) image for testing

- [Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.12-beaglebone-2013.05.24-LCD-cape-LCD7-v1.1.img.7z](#)

MISC CAPE and RS485/CAN CAPE image for testing

- [Angstrom-Cloud9-IDE-GNOME-eglibc-ipk-v2012.12-beaglebone-2013.05.24-MISC-cape-HDMI-v1.1.img.7z](#)

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!

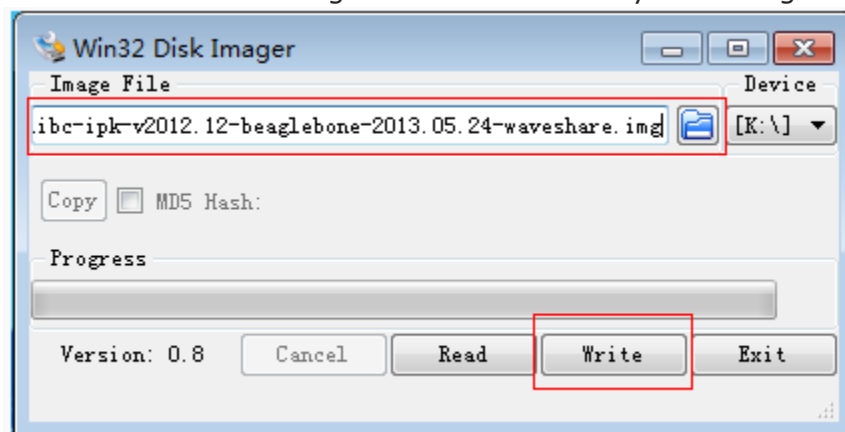
Use HPUSBDisk.exe to format the TF card



3) Writing the system image

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

Use Win32DiskImager.exe to write the system image



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:

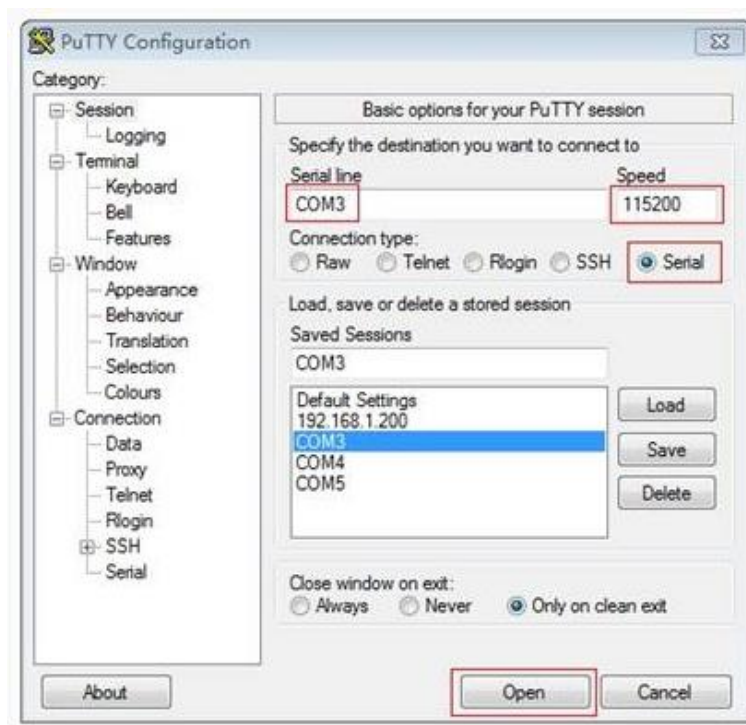
- Red: VCC
- Black: GND
- Green: TXD(connect to RXD)
- White: RXD(connect to TXD)

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.

PuTTY Settings



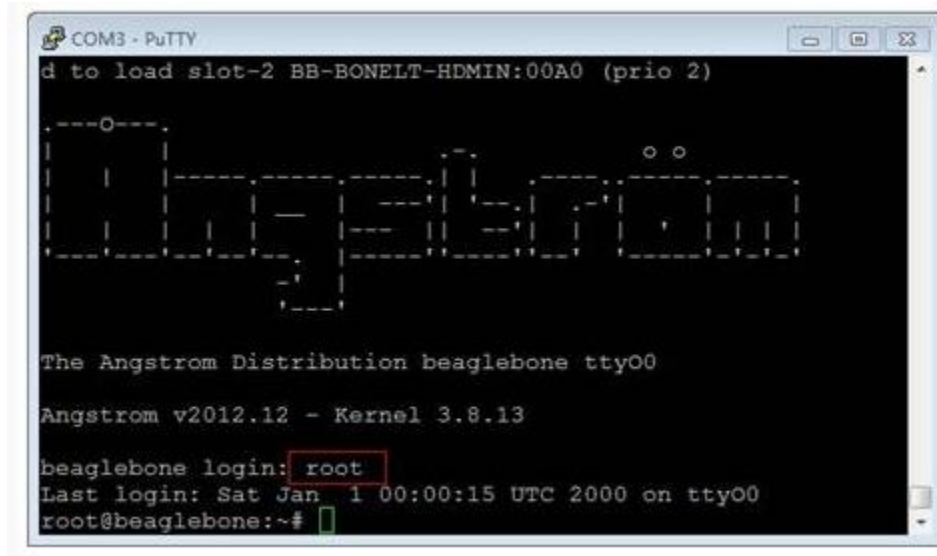
Note:

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

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.

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.

API Source Code

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

LCD CAPE

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

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.

Connect BB Black to the 4.3inch LCD.

Connect to the 4.3inch LCD



- When using 7inch LCD

Download LCD CPAE(7inch) image for testing.

Connect BB Black to the 7inch LCD.

Connect to the 7inch LCD



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

```
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.**

MISC CAPE

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

LED

Short the LED jumper, enter:

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

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

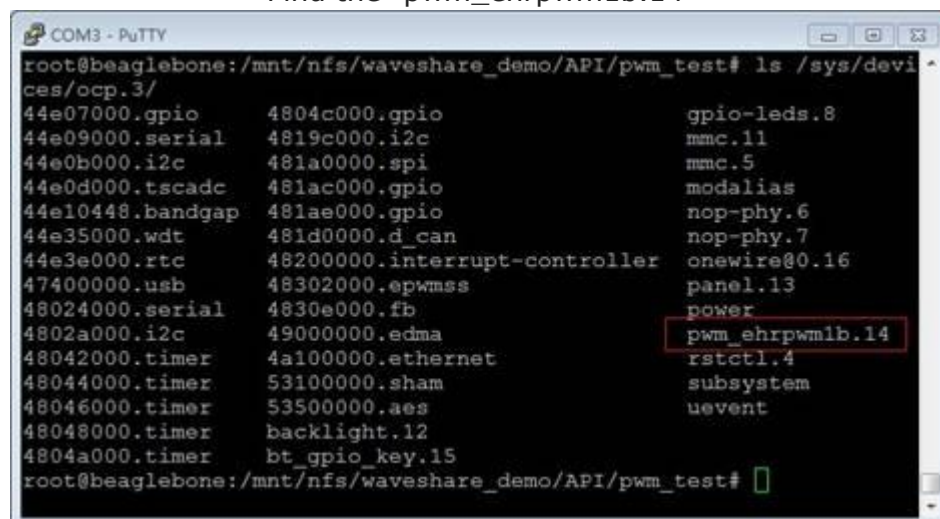
Buzzer

1) Short the buzzer jumper, enter:

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

You'll find the "pwm_ehrpwm1b.14" as shown:

Find the "pwm_ehrpwm1b.14"



```
COM3 - PuTTY
root@beaglebone:/mnt/nfs/waveshare_demo/API/pwm_test# ls /sys/devices/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#
```

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.

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.

Buttons

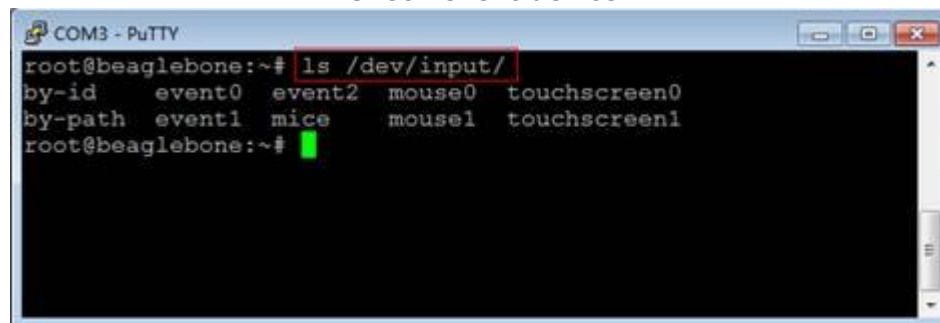
1) Execute

```
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
```

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 joystick/buttons, the LCD screen seems that it was touched at the same time, it's normal.

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
```

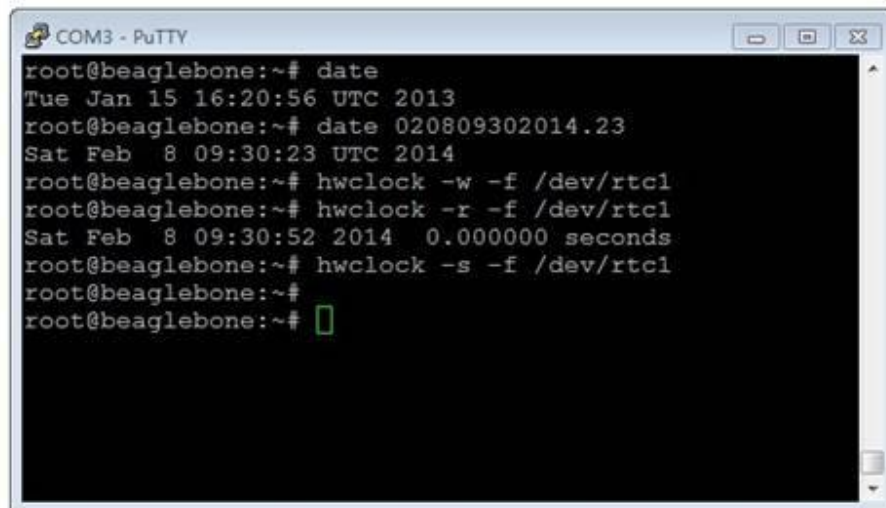
4) Show the hardware clock of RTC module:

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

5) Synchronize the hardware clock to system date time:

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

Synchronize the hardware clock to system date time



```
COM3 - PuTTY
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:~#
```

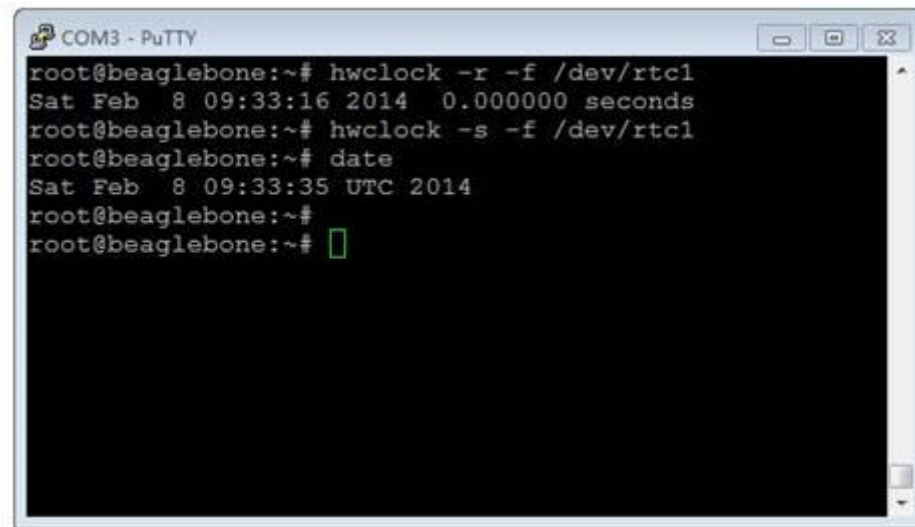
6) Power off and reboot, read the hardware clock of RTC module and synchronize to system date time:

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

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

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

Synchronize to system date time



```
COM3 - PuTTY
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:~#
```

Now the date time of software and hardware are synchronous.

RS485 CAN CAPE

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
```

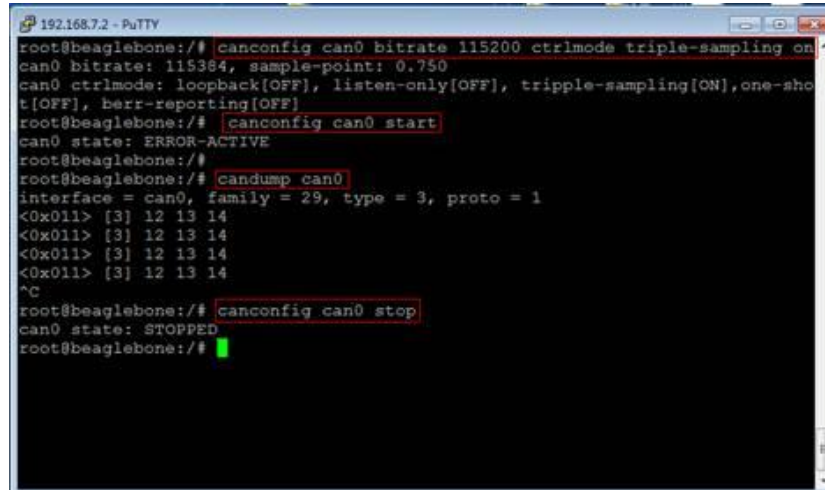
The two CAN devices act as receiver and sender separately. The receiver gets ready to receive data first, and then the sender starts sending data.

3) Receiver:

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

As shown in the picture:

CAN receiver status



```
192.168.7.2 - PuTTY
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-shot[OFF], berr-reporting[OFF]
root@beaglebone:~# canconfig can0 start
can0 state: ERROR-ACTIVE
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:~#
```

5) Stop the devices:

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

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:

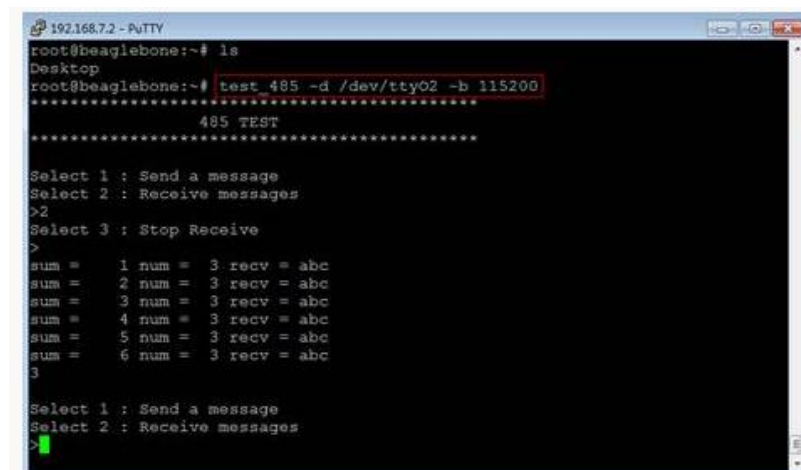
```
root@beaglebone:~# test_485 -d /dev/ttyO2 -b 115200
```

2) Receiver:

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

As shown in the picture:

RS485 receiver status



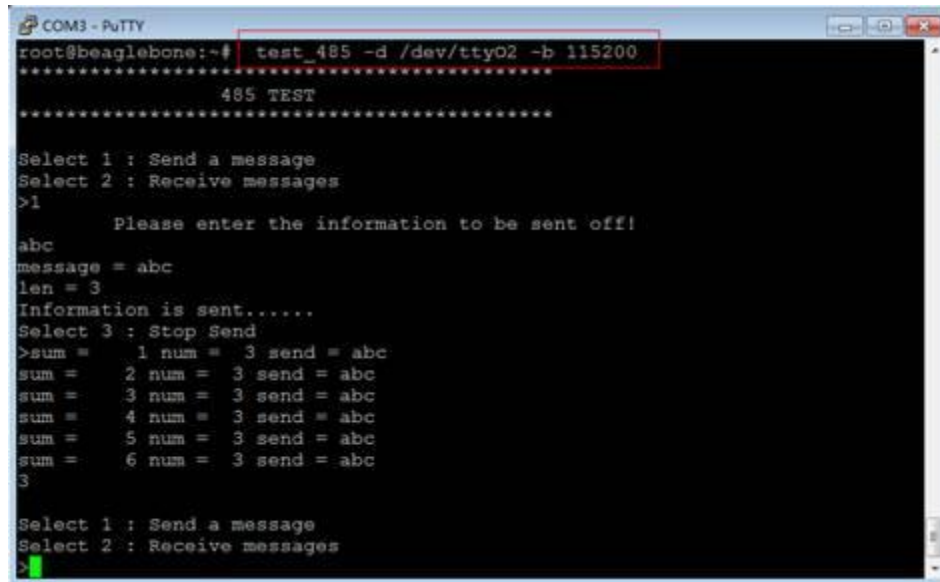
```
192.168.7.2 - PuTTY
root@beaglebone:~# ls
Desktop
root@beaglebone:~# test_485 -d /dev/ttyO2 -b 115200
*****
          485 TEST
*****
Select 1 : Send a message
Select 2 : Receive messages
>2
Select 3 : Stop Receive
>
sum = 1 num = 3 recv = abc
sum = 2 num = 3 recv = abc
sum = 3 num = 3 recv = abc
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
>
```

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:

RS485 sender status



```
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
>
```

4) Press Ctrl+C to exit.

Other Expansion Module

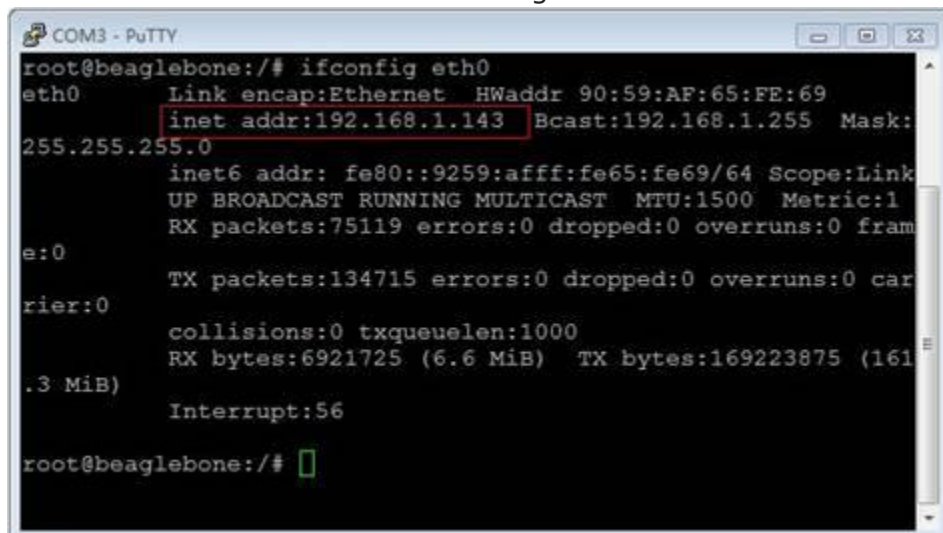
USB Camera

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

2) Check the assigned IP:

```
root@beaglebone:~# ifconfig eth0
```

Check the assigned IP



```
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 frame
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:/#
```

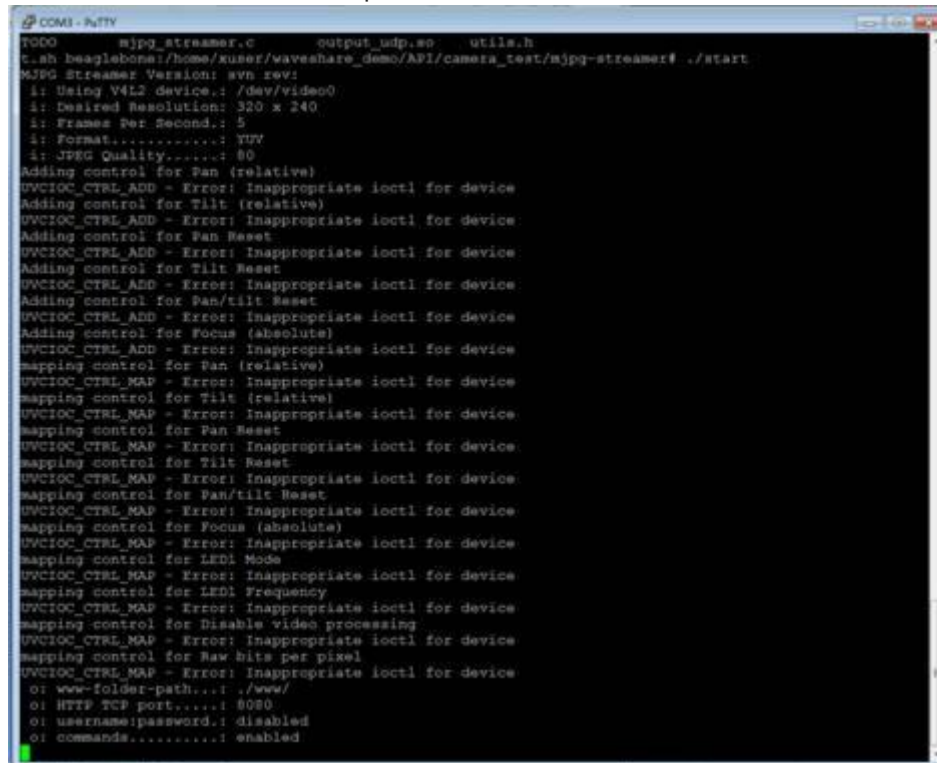
As shown in the picture, the IP is 192.168.1.143. Note this IP(it depends).

3) Start up the video stream server:

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

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

Start up the video stream server



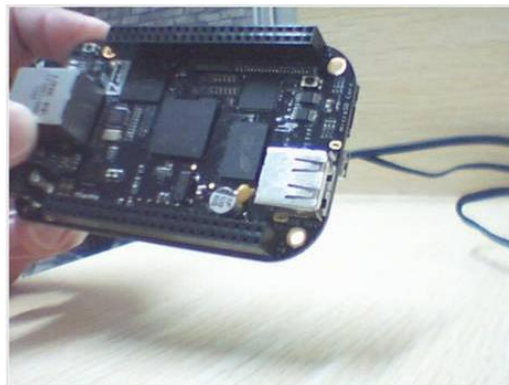
```
root@beaglebone:~# ./ start.sh
mjpg_streamer.c  output_udp.so  utils.h
c.sh beaglebone:/home/xuser/waveshare_demo/API/camera_test/mjpg-streamer# ./start
MJPEG Streamer Version: svn rev:
i: Using V4L2 Device: /dev/video0
i: Desired Resolution: 320 x 240
i: Frames Per Second: 5
i: Format: YUYV
i: JPEG 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
```

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)

Visit video stream server



5) Press Ctrl+C to exit.

USB WIFI

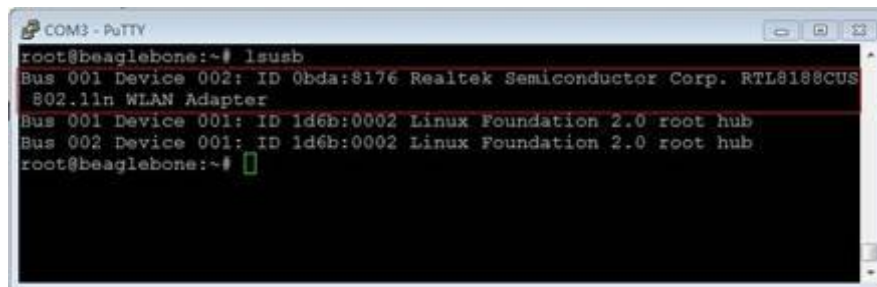
1) Config the wireless network:

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
```

Check the USB status

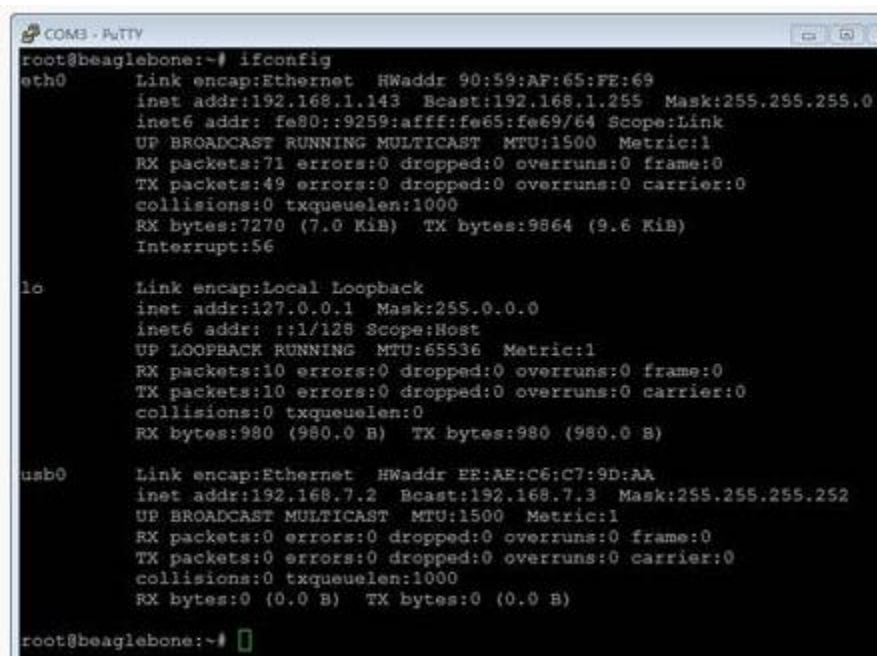


```
COM3 - PuTTY
root@beaglebone:~# lsusb
Bus 001 Device 002: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS
802.11n WLAN Adapter
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@beaglebone:~#
```

c) Check the network status:

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

Check the network status



```
COM3 - PuTTY
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:aff: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:~#
```


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

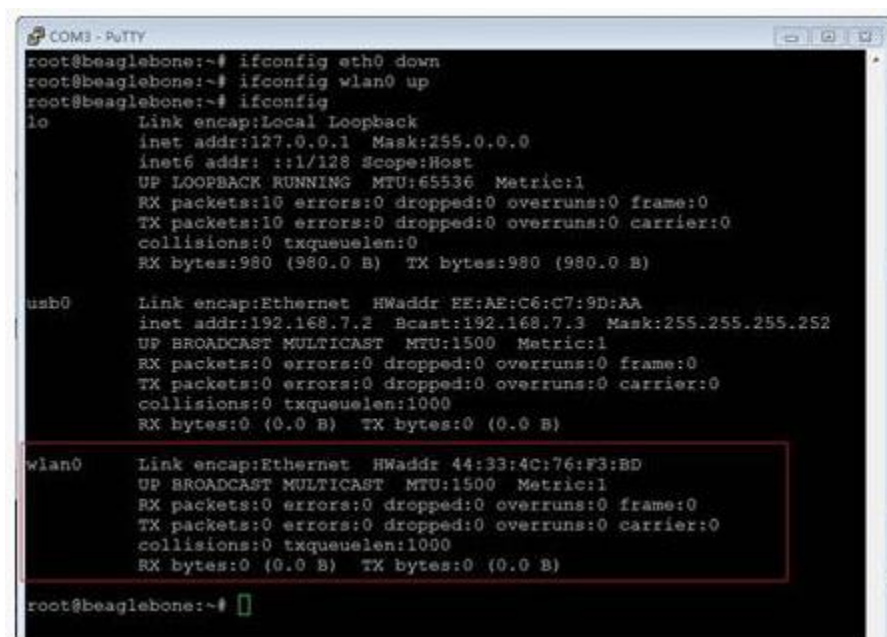
```
root@beaglebone:~# ifconfig eth0 down
```

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

e) Check the WIFI status:

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

Check the WIFI status



```
COM3 - PuTTY
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:~#
```

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
```

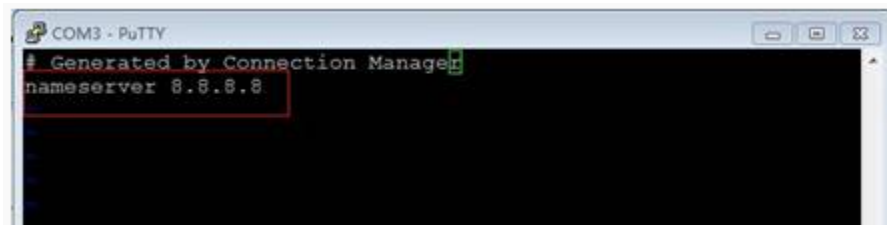
h) Config the DNS:

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

```
Edit
nameserver 127.0.0.1
```

As:
nameserver 8.8.8.8

Configure the DNS



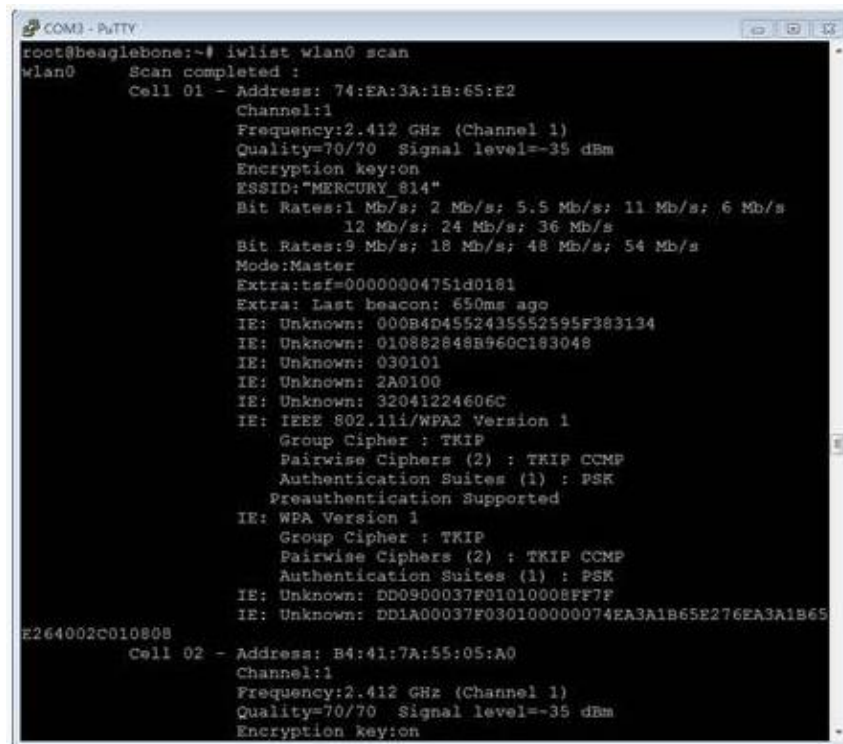
```
COM3 - PuTTY
# Generated by Connection Manager
nameserver 8.8.8.8
```

Save and exit.

i) Scan wireless router:

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

Scan wireless router



```
COM3 - PuTTY
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) : TRIP CCMP
      Authentication Suites (1) : PSK
      Preauthentication Supported
    IE: WPA Version 1
      Group Cipher : TKIP
      Pairwise Ciphers (2) : TRIP CCMP
      Authentication Suites (1) : PSK
    IE: Unknown: DD0900037F01010008FF7F
    IE: Unknown: DD1A00037F030100000074EA3A1B65E276EA3A1B65
#264002C010808
  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
```

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.

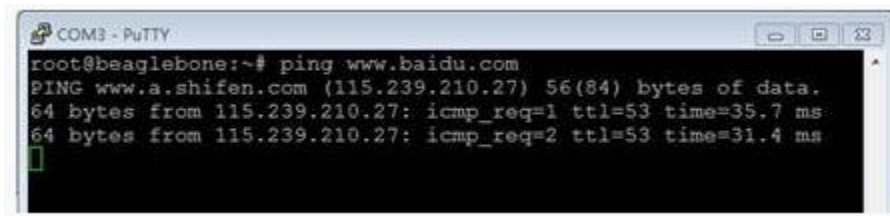
b) Connect manually:

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

c) Test the connection:

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

Test WIFI connection:

A screenshot of a terminal window titled 'COM3 - PuTTY'. The terminal shows the command 'root@beaglebone:~# ping www.baidu.com' and its output. The output indicates a successful ping to 'www.a.shifen.com (115.239.210.27)' with 56(84) bytes of data. Two ping requests are shown: the first with a time of 35.7 ms and the second with a time of 31.4 ms. A green cursor is visible at the bottom of the terminal.

```
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  
^
```