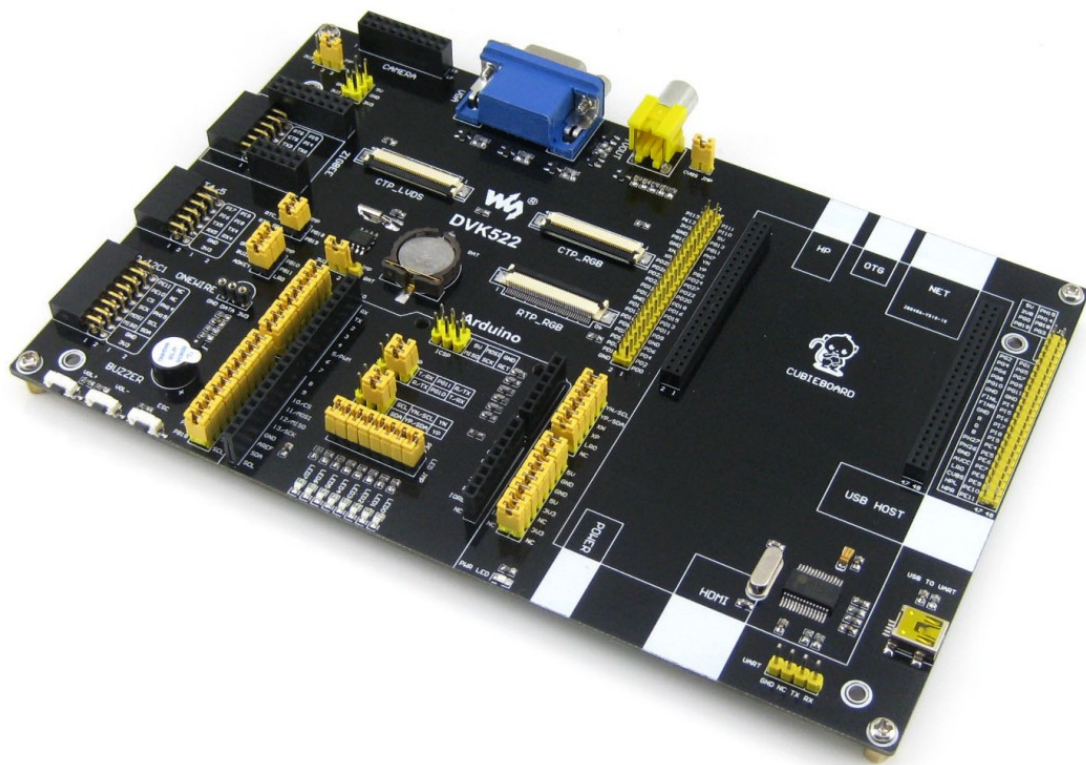


DVK522 Expansion Board

User Manual

2014.06.17 V1.1



Waveshare Electronics

www.wvshare.com

Copyright Statement

The ownership of this user manual is Shenzhen Waveshare Electronics Co., Ltd.
Any manner or form of modifying, distributing or copying any parts of this document without permission is forbidden, otherwise offender will have to suffer all the consequences.

Version update records

Version	Date	Description
V1.1	2014.06.17	Initial Release

Content

Copyright Statement	I
Version update records.....	I
Document formatting convention	1
1. Overview	1
2. What's on board	1
2.1. Core interface.....	1
2.2. Component.....	2
2.3. Jumper	3
3. Preparation	3
3.1 Firmware burning.....	3
3.2 USB to serial port driver installation.....	4
4. Starting mode.....	5
4.1. TF Card starting	5
4.2. Nand starting	5
5. Type of system	5
5.1 Ubuntu	5
5.2 Debian.....	5
6. Demos.....	5
6.1. Display Demo	6
6.2. LED Demo.....	7
6.3. Buzzer Demo.....	7
6.4. DS18B20 Demo	8
6.5. AD Keypad Demo.....	8
6.6. AT45DB Read/Write Demo.....	9
6.7. MAG3110 Demo.....	10
6.8. UART Interface Demo	11
6.9. RS485 Demo	11
6.10. GPS Demo	13
6.11. ZIGBEE Demo.....	13
6.12. RTC Demo	13
6.13. USB Camera Demo	14
6.14. WIFI Demo	17

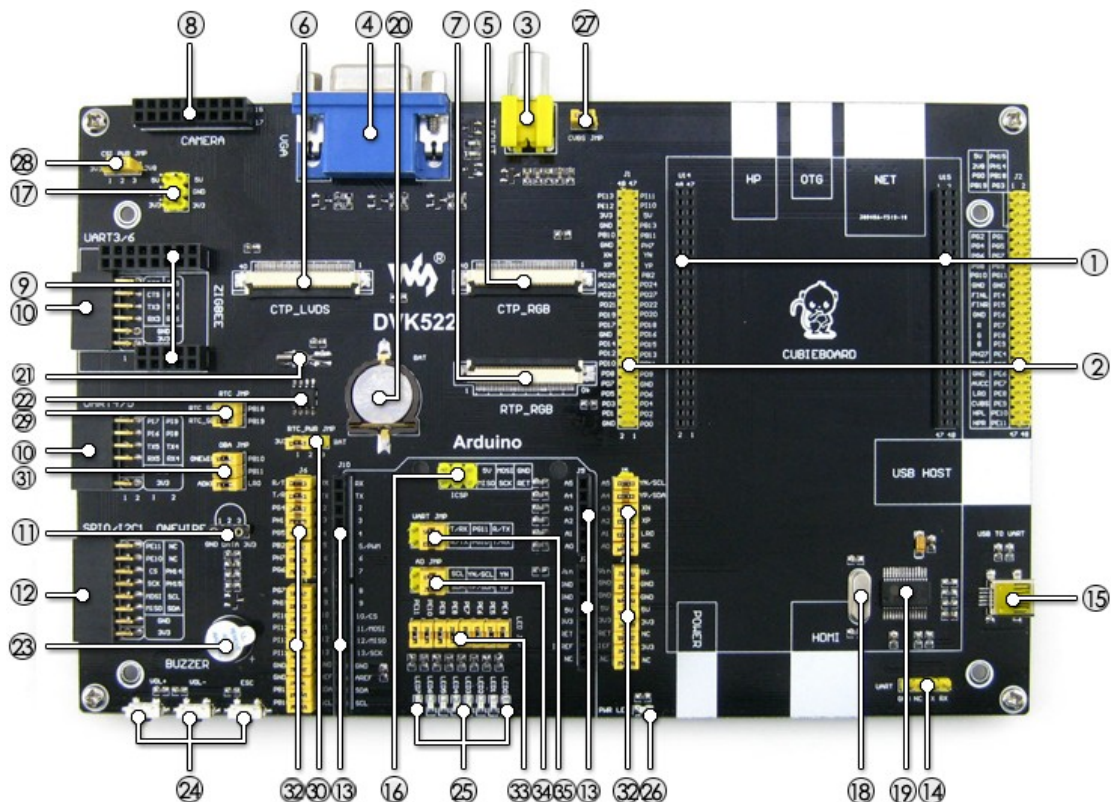
Document formatting convention

1. Commands on PC ubuntu terminal: preceding with '#', here root user privileges is requested.
Commands on the board terminal: preceding with '\$'
2. This Manual is for API demo on Ubuntu and Debian system.

1. Overview

DVK522 is an expansion board designed for Cubieboard1/2, integrates various components and interfaces for connecting external accessory boards, complete sample codes will also be offered, users can easily develop your own products with it.

2. What's on board



2.1. Core interface

- 1) CUBIEBOARD1/2 socket
for connecting CUBIEBOARD1/2

- 2) CUBIEBOARD1/2 expansion connector
For expanding IOs
- 3) TVOUT interface
video CVBS output, for connecting display devices like TV
- 4) VGA interface
for connecting VGA display Module
- 5) 7inch capacitive LCD RGB interface
for connecting 7inch capacitive LCD with RGB interface
- 6) 7inch capacitive LCD LVDS interface
for connecting 7inch capacitive LCD with LVDS interface
- 7) 7inch resistive LCD RGB interface
for connecting 7inch resistive LCD with RGB interface
- 8) CAMERA interface
for connecting OV7670 Camera Module
- 9) ZIGBEE connector
for connecting ZigBee modules like Core2530, supports working with ZB501
- 10) UART interface
easily connects to UART modules, such as RS485 Board (3.3V)、GPS module, etc.
- 11) ONEWIRE interface
easily connects to ONE-WIRE devices (TO-92 package), such as temperature sensor (DS18B20), etc.
- 12) SPI0/I2C1 interface
easily connects to SPI or I2C modules such as MAG3110 Board、AT45DBXX Dataflash, etc.
- 13) Arduino interface
for connecting compatible Arduino modules
- 14) UART interface (PL2303TA)
for connecting to UART0 interface on CUBIEBOARD1/2
- 15) USB interface
USB TO UART (PL2303TA)
- 16) ICSP interface
ICSP interface for Arduino
- 17) 5V/3.3V power input/output
usually used as power output, also common-grounding with other user board

2.2. Component

- 18) 12M crystal
for PL2303TA

- 19) PL2303TA
USB TO UART MCU
- 20) RTC backup battery
3.3V battery
- 21) 32.768KHZ crystal
for PCF8563
- 22) PCF8563
RTC board MCU
- 23) Buzzer
- 24) AD keys
3 common Android keys:
VOL+、VOL-、ESC
- 25) User LED
8 user LEDs
- 26) Power indicator

2.3. Jumper

- 27) TVOUT selection jumper
- 28) CAMERA power selection jumper
- 29) RTC selection jumper
- 30) RTC power selection jumper
- 31) ONEWIRE、buzzer and AD key selection jumper
- 32) Arduino Port selection jumper
easily controlling the Arduino connection
- 33) User LED selection jumper
- 34) Arduino AD selection jumper
short the right: while connecting Arduino ADC shield;
short the left: while connecting Arduino I2C shield.
- 35) Arduino UART selection jumper
short the right: while connecting Arduino UART shield
short the left: while connecting Arduino microcontroller board

3. Preparation

3.1 Firmware burning

Burning corresponding firmware: cbx-dvk-xxxx.img is needed when testing DVK522 ,
download it from :

<http://dl.cubieboard.org/parteners/waveshare/Image/>,
detail please refer to “Burning Firmware”.

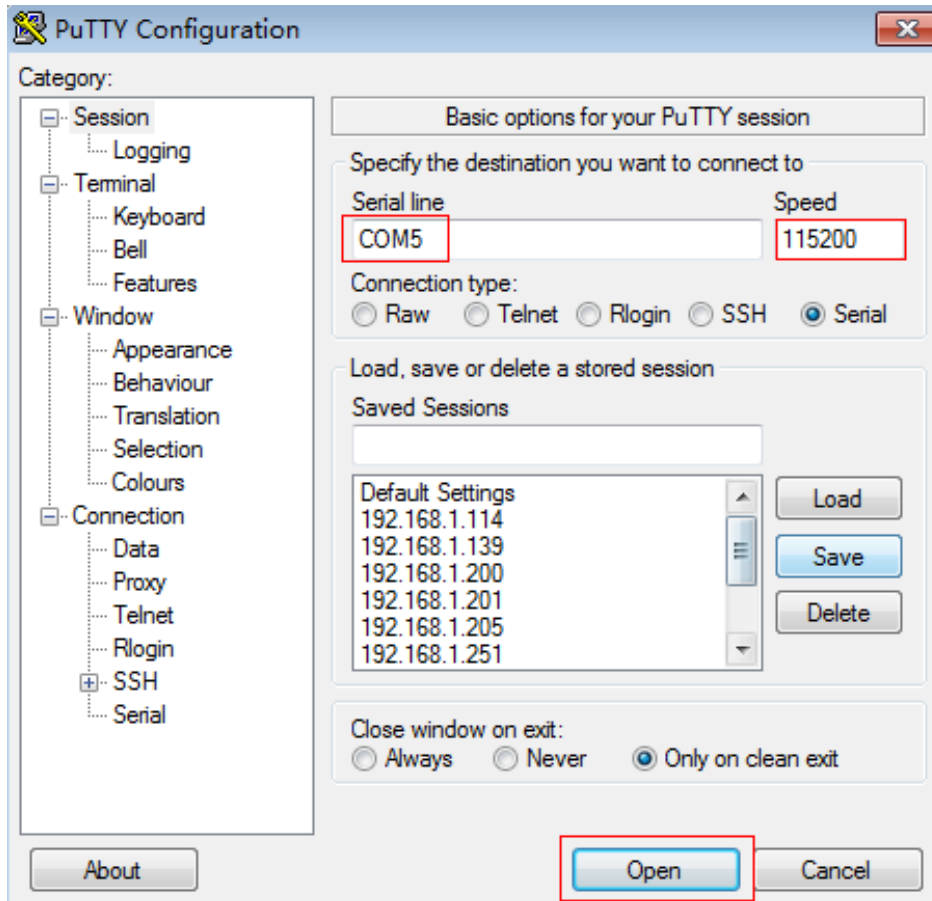
Note:

1.Note: Firmware with “sdcard” marking is TF card firmware; with “nand” marking is nand firmware.

2.Firmware version of v1.1 or above is suitable for testing DVK522, version v1.0 is compatible for old version DVK521.

3.2 USB to serial port driver installation

Connect UART connector of DVK522 to UART0 connector of cubieboard 1/2 via 4-pin or 2-pin wires, connect USB TO UART connector of DVK522 to PC via mini USB wire. Launch **PL2303_Prolific_DriverInstaller_v1.8.0.exe** to install the driver. After installed, launch the serial debugging assistant and check putty.exe, configure like below, then click “open”.



Note: Check your computer’s “Device Manager” to see what the COM is.

4. Starting mode

4.1. TF Card starting

Insert the burned firmware TF card to the DATA CARD slot of the Cubieboard1/2, connect to electricity for starting.

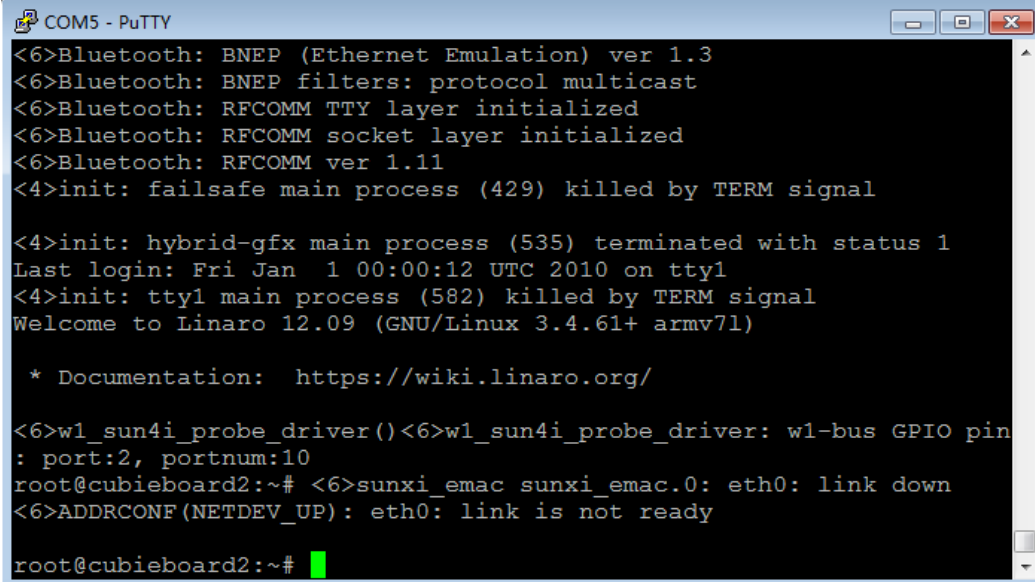
4.2. Nand starting

After burned firmware, connect to electricity for starting directly.

5. Type of system

5.1 Ubuntu

Input "enter" at the terminal to enter Bash Shell development environment with root user privileges.



```
COM5 - PuTTY
<6>Bluetooth: BNEP (Ethernet Emulation) ver 1.3
<6>Bluetooth: BNEP filters: protocol multicast
<6>Bluetooth: RFCOMM TTY layer initialized
<6>Bluetooth: RFCOMM socket layer initialized
<6>Bluetooth: RFCOMM ver 1.11
<4>init: failsafe main process (429) killed by TERM signal

<4>init: hybrid-gfx main process (535) terminated with status 1
Last login: Fri Jan  1 00:00:12 UTC 2010 on tty1
<4>init: tty1 main process (582) killed by TERM signal
Welcome to Linaro 12.09 (GNU/Linux 3.4.61+ armv7l)

* Documentation:  https://wiki.linaro.org/

<6>w1_sun4i_probe_driver()<6>w1_sun4i_probe_driver: w1-bus GPIO pin
: port:2, portnum:10
root@cubieboard2:~# <6>sunxi_emac sunxi_emac.0: eth0: link down
<6>ADDRCONF(NETDEV_UP): eth0: link is not ready

root@cubieboard2:~#
```

5.2 Debian

Reboot the system, input user name and password:

User:root

Password:123456

Then enter Bash Shell development environment with root user privileges.

Note: If it is not root user privilege, switch to root user !

6. Demos

Note:

1.Ensure the system firmware is provided by us, otherwise the test could not be operated!

2.Reboot the system, find all the testing sample under root of /home/waveshare_demo/API.

6.1. Display Demo

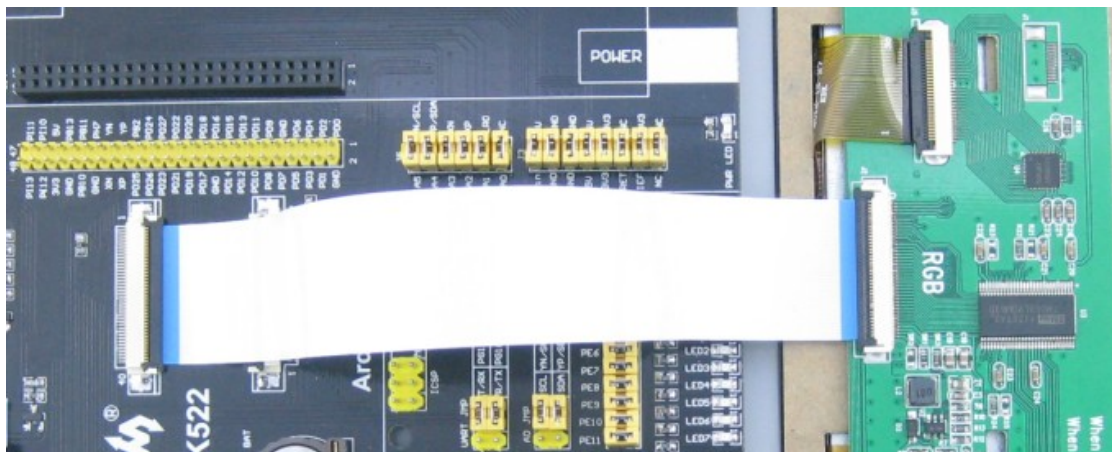
Different display demo must use corresponding firmware!

1) LCD demo

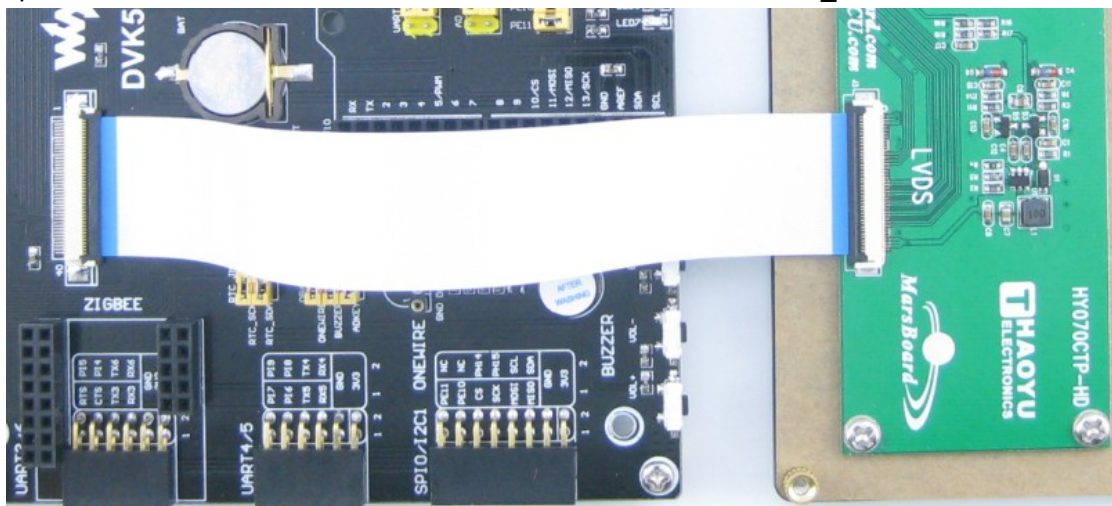
Connect connector of LCD to connector of DVK522 via 40pin wires like below:

LCD (connector)	DVK522 (connector)
RGB	CTP_RGB
LVDS	CTP_LVDS

a) Connect the RGB connector on the LCD to DVK522 via CTP_RGB connector :



b) Connect the LVDS connector on the LCD to DVK522 via CTP_LVDS connector :



Note:

Our standard 7" TFT LCD is capacitive touch screen, the controller is FT5X06, resolution is 1024x600, with RGB and LVDS connector.

It requests to make hardware configuration on LCD board(detail please check on

back side of the LCD board) when using different connector:

1) RGB connector: R24=10K, R25=NC, the 40PIN FFC connect to the RGB connector;

2) LVDS connector: R24=NC, R25=10K, the 40PIN FFC connect to the LVDS connector.

Default is RGB conector.

2) VGA demo

Connect the display LCD to the DVK522 onboard VGA connector via VGA wire.

3) HDMI demo

Connect the display to the cubieboard1/2 onboard HDMI connector via HDMI wire.

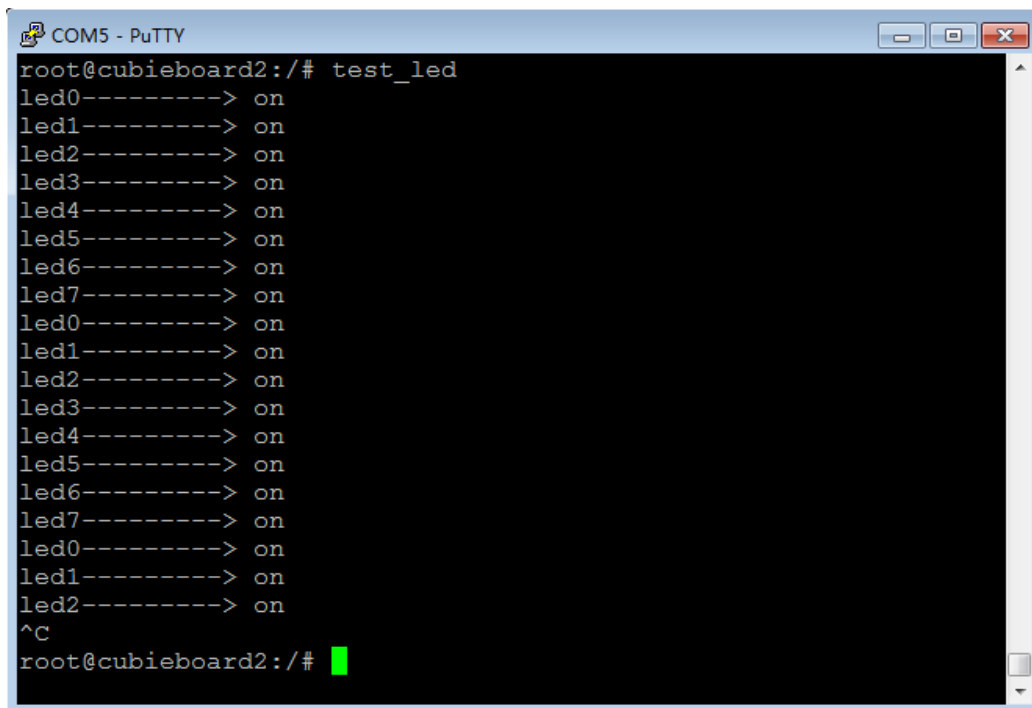
6.2. LED Demo

Short the onboard LED jumper (set by default, following are the same)

Enter the following command:

```
$ test_led
```

8 leds blinking:



```
COM5 - PuTTY
root@cubieboard2:/# test_led
led0-----> on
led1-----> on
led2-----> on
led3-----> on
led4-----> on
led5-----> on
led6-----> on
led7-----> on
led0-----> on
led1-----> on
led2-----> on
led3-----> on
led4-----> on
led5-----> on
led6-----> on
led7-----> on
led0-----> on
led1-----> on
led2-----> on
^C
root@cubieboard2:/# █
```

Press "Ctrl+C" to stop.

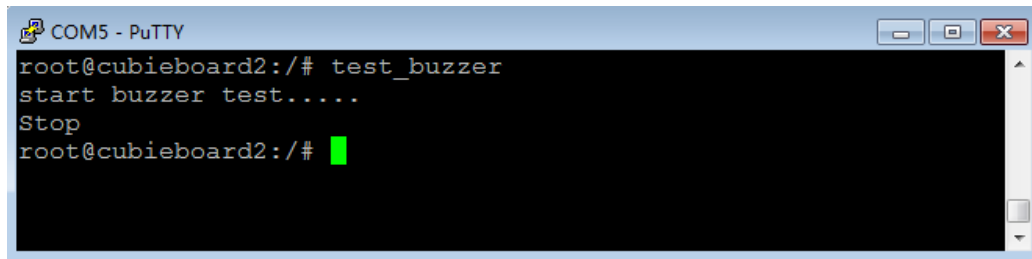
6.3. Buzzer Demo

Short the onboard BUZZER jumper

Enter the following command:

`$test_buzzer`

The buzzer will start to make sound.



```
COM5 - PuTTY
root@cubieboard2:/# test_buzzer
start buzzer test.....
Stop
root@cubieboard2:/# █
```

6.4. DS18B20 Demo

Insert the DS18B20 into the ONEWIRE socket, short the onboard ONEWIRE jumper

Enter the following command:

`$ls /sys/bus/w1/devices/`

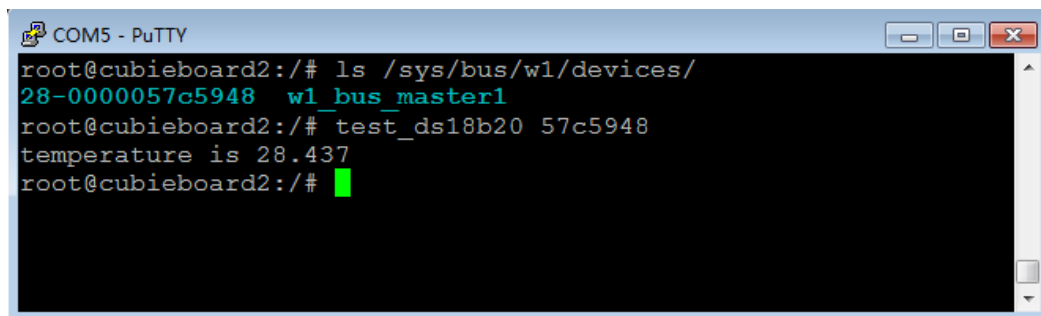
There's a folder with the name looks like "28-0000054a5bec " (the last 7 characters is an unique device ID of every different DS18B20, in this case, the device ID is 57c5948)

Enter the following command:

`$test_ds18b20 57c5948`

The current environment temperature will be printed on the console terminal.

As shown in the figure below:



```
COM5 - PuTTY
root@cubieboard2:/# ls /sys/bus/w1/devices/
28-0000057c5948 w1_bus_master1
root@cubieboard2:/# test_ds18b20 57c5948
temperature is 28.437
root@cubieboard2:/# █
```

6.5. AD Keypad Demo

Short the ADKEY jumper

Enter the following command:

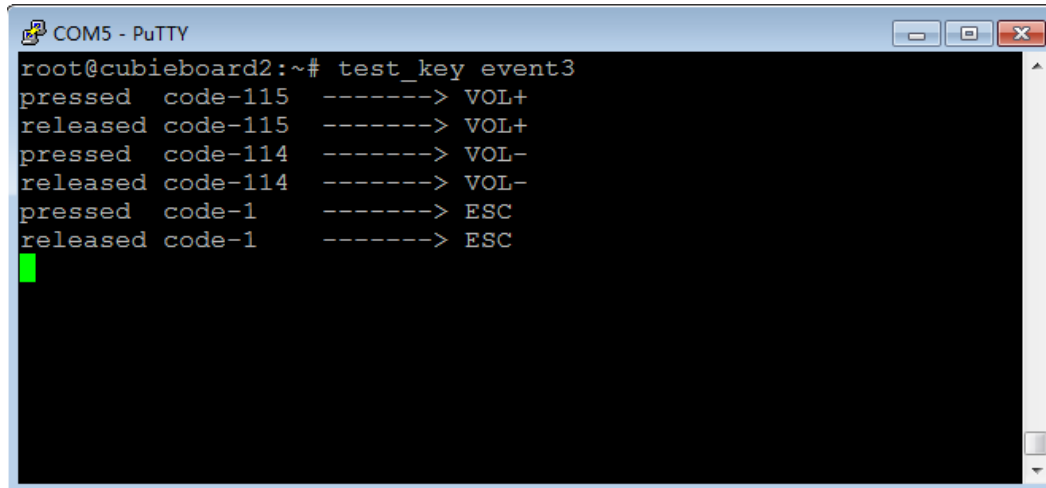
`$ test_key event3`

Note:

1.What the event* is depended on the actual situation, please check directory "/dev/input";

2. Please perform: `modprobe sun4i-keyboard` and put `sun4i-keyboard` into listing `/etc/modules` if the keypad driver is not `insmod`.

Press 3 keypads respectively, then related key value that were pressed or released will be displayed on the console terminal:



```
COM5 - PuTTY
root@cubieboard2:~# test_key event3
pressed  code-115  -----> VOL+
released code-115  -----> VOL+
pressed  code-114  -----> VOL-
released code-114  -----> VOL-
pressed  code-1    -----> ESC
released code-1    -----> ESC
█
```

Press “ Ctrl+C ” to stop.

6.6. AT45DB Read/Write Demo

Connect the AT45DBXX DataFlash Board to the onboard SPI0 interface

Enter the following command:

```
$ test_at45db
```

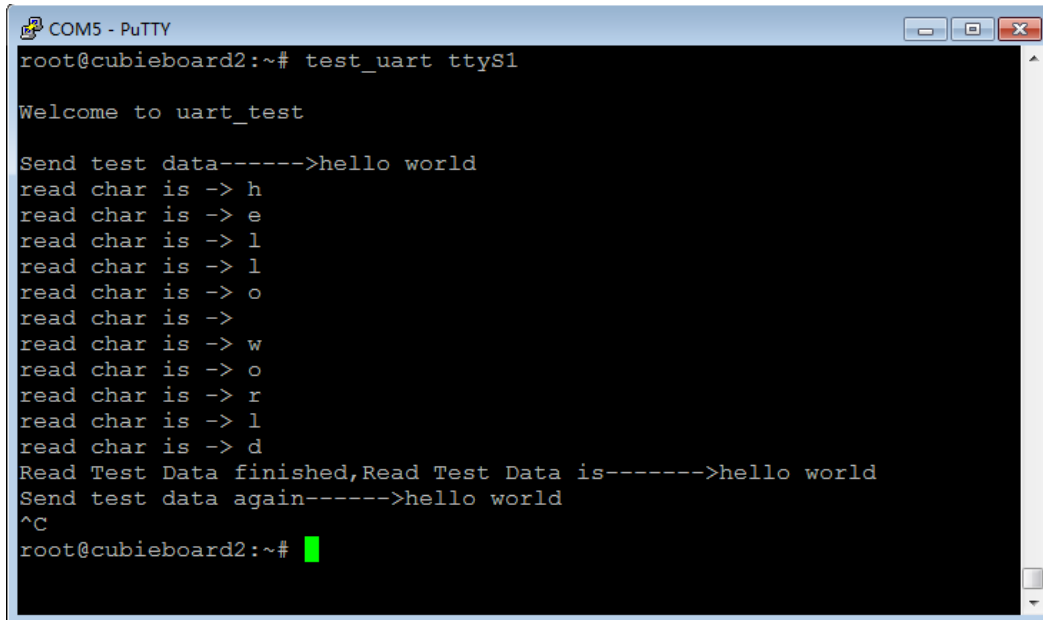
Data written and read will be displayed on the console terminal:

6.8. UART Interface Demo

Short RXD and TXD of the UART

Enter the following command:

```
$ test_uart ttyS1
```



```
COM5 - PuTTY
root@cubieboard2:~# test_uart ttyS1

Welcome to uart_test

Send test data----->hello world
read char is -> h
read char is -> e
read char is -> l
read char is -> l
read char is -> o
read char is ->
read char is -> w
read char is -> o
read char is -> r
read char is -> l
read char is -> d
Read Test Data finished,Read Test Data is----->hello world
Send test data again----->hello world
^C
root@cubieboard2:~#
```

If the serial port can receive and send automatically, it means the UART can work normally.

Note:

UART3 interface corresponds to ttyS1,
UART4 interface corresponds to ttyS2,

6.9. RS485 Demo

Connect the two RS485 Boards to the onboard UART3 and UART4 interface. Connect the A, B side of one RS485 module to the A, B side of the other RS485 module using connecting wires. Users can also use their own RS485 for testing.

The two RS485 devices can optionally regarded as a receiver or transmitter, the receiver should firstly be in receiving status, then the transmitter begin to send.

Open two console terminals, one enter the following command:

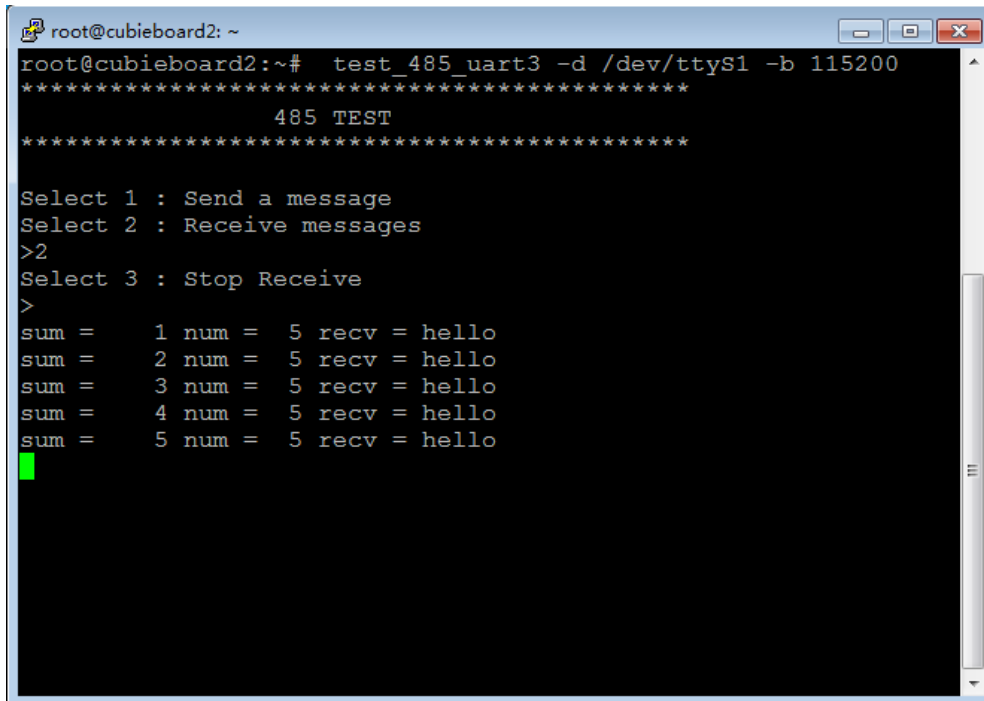
```
$ test_485_uart3 -d /dev/ttyS1 -b 115200
```

The other enter the following command:

```
$ test_485_uart4 -d /dev/ttyS2 -b 115200
```

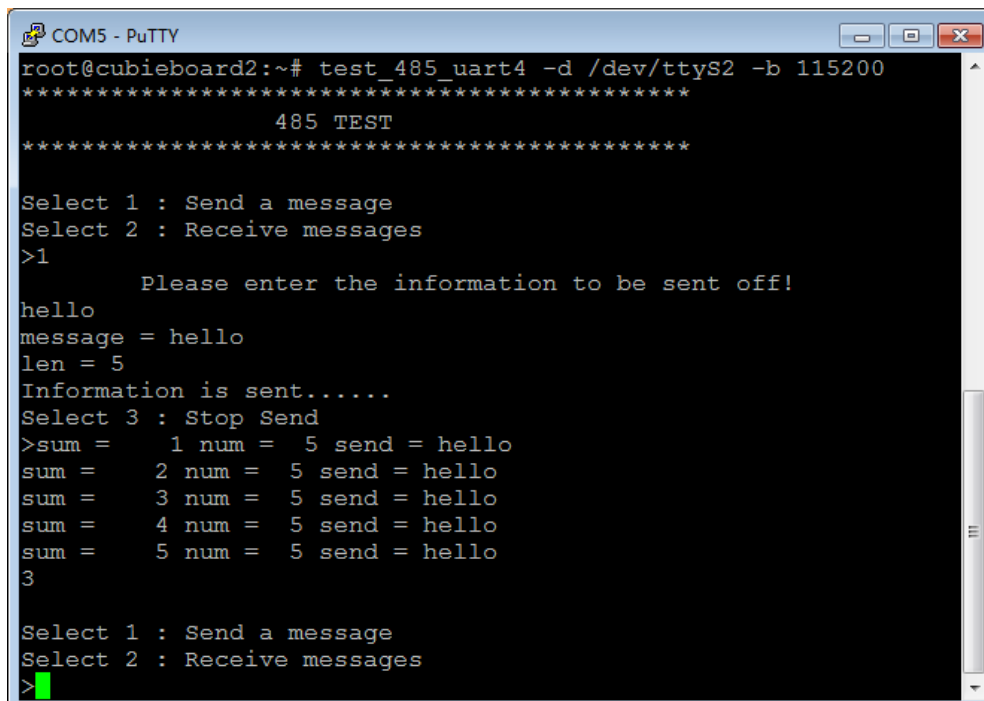
1) Select "2" for the receiver, before select "3" to stop receiving, the receiver remain

in the status of receiving ,
As shown in the figure below:



```
root@cubieboard2: ~  
root@cubieboard2:~# test_485_uart3 -d /dev/ttyS1 -b 115200  
*****  
485 TEST  
*****  
Select 1 : Send a message  
Select 2 : Receive messages  
>2  
Select 3 : Stop Receive  
>  
sum = 1 num = 5 recv = hello  
sum = 2 num = 5 recv = hello  
sum = 3 num = 5 recv = hello  
sum = 4 num = 5 recv = hello  
sum = 5 num = 5 recv = hello  
█
```

- 2) Select "1" for the receiver, enter information that you want to send, such as "hello", before select "3" to stop sending, the transmitter remain in the status of loop sending, keep sending data,
As shown in the figure below:



```
COM5 - PuTTY  
root@cubieboard2:~# test_485_uart4 -d /dev/ttyS2 -b 115200  
*****  
485 TEST  
*****  
Select 1 : Send a message  
Select 2 : Receive messages  
>1  
Please enter the information to be sent off!  
hello  
message = hello  
len = 5  
Information is sent.....  
Select 3 : Stop Send  
>sum = 1 num = 5 send = hello  
sum = 2 num = 5 send = hello  
sum = 3 num = 5 send = hello  
sum = 4 num = 5 send = hello  
sum = 5 num = 5 send = hello  
3  
Select 1 : Send a message  
Select 2 : Receive messages  
>█
```

Press Ctrl+C to stop.

6.10. GPS Demo

Connect the UART GPS module to the onboard UART3 interface,
Enter the following command:

Note:

1.The GPS module demo must be operated outdoor, otherwise it can not receive the satellite data.

2.The testing program is compatible with GPS module with baud rate 9600.

```
$ test_gps ttyS1
```

User can analyze corresponding data according to their needs. Press Ctrl+C to stop.

Detail for how to use the UART GPS module please refer to:

<http://www.wvshare.com/aspx/search.aspx?keywords=GPS>

6.11. ZIGBEE Demo

Interface of onboard ZIGBEE only compatible with configured Core2530 board.

When burning program for the core board or other corresponding configure, it may need support from ZB501. Detail development package please refer to:

<http://www.wvshare.com/product/ZB501.htm>

6.12. RTC Demo

Test onboard DVK522 RTC, enter corresponding demand on the console terminal:

Note:

This demo must be sure to use button battery, that is short RTC_PWR JMP to "2-3', BAT side.

1) Add device:

```
$ echo pcf8563 0x51 > /sys/class/i2c-adapter/i2c-1/new_device
```

Note:

Please execute: modprobe rtc-pcf8563 if RTC driver did not load and add rtc-pcf8563 to the list of /etc/modules

2) Read the system time

```
$ date
```

3) Configure the system time:

```
$ date 061717502014.23
```

4) Set the hardware of RTC time:

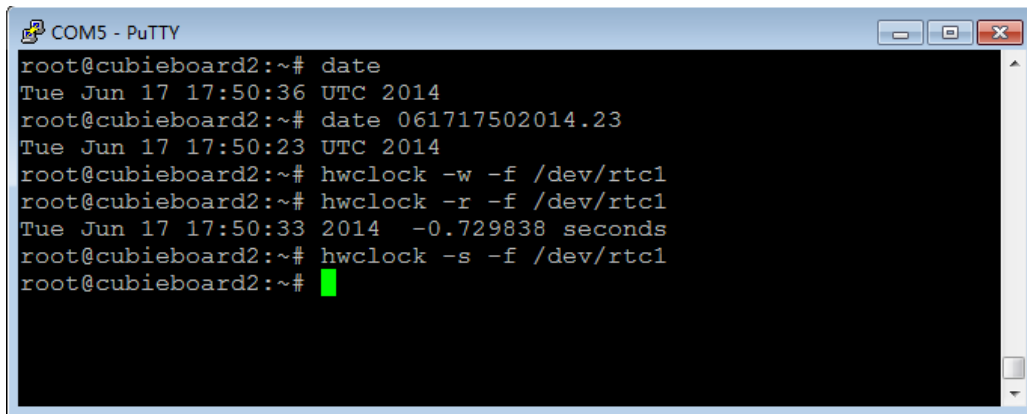
```
$ hwclock -w -f /dev/rtc1
```

5) Read time of RTC hardware:

```
$ hwclock -r -f /dev/rtc1
```

6) Time of RTC hardware synchronized to the system time:

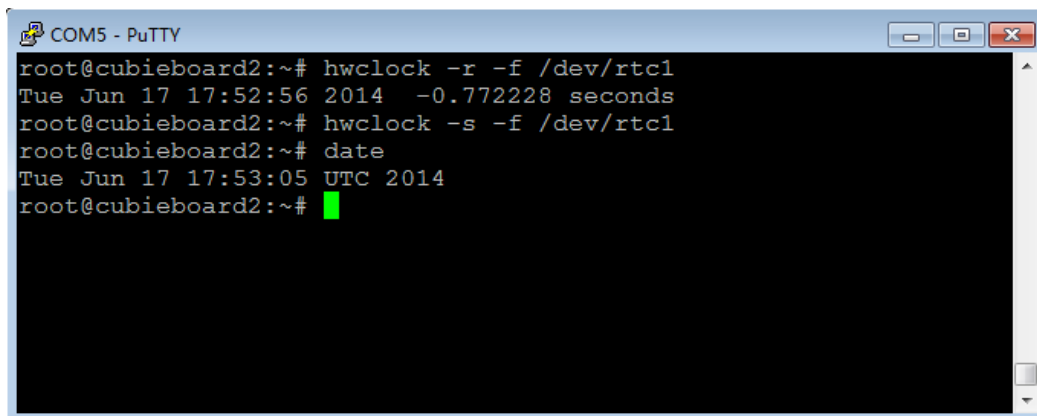
```
$ hwclock -s -f /dev/rtc1
```

```
COM5 - PuTTY
root@cubieboard2:~# date
Tue Jun 17 17:50:36 UTC 2014
root@cubieboard2:~# date 061717502014.23
Tue Jun 17 17:50:23 UTC 2014
root@cubieboard2:~# hwclock -w -f /dev/rtc1
root@cubieboard2:~# hwclock -r -f /dev/rtc1
Tue Jun 17 17:50:33 2014 -0.729838 seconds
root@cubieboard2:~# hwclock -s -f /dev/rtc1
root@cubieboard2:~#
```

- 7) Cut off the power and restart, read the time of RTC hardware and synchronized to the system time, enter the following command:

```
$ echo pcf8563 0x51 > /sys/class/i2c-adapter/i2c-1/new_device
$ hwclock -r -f /dev/rtc1
$ hwclock -s -f /dev/rtc1
$ date
```



```
COM5 - PuTTY
root@cubieboard2:~# hwclock -r -f /dev/rtc1
Tue Jun 17 17:52:56 2014 -0.772228 seconds
root@cubieboard2:~# hwclock -s -f /dev/rtc1
root@cubieboard2:~# date
Tue Jun 17 17:53:05 UTC 2014
root@cubieboard2:~#
```

Now, time of the software and hardware are synchronized, the RTC works normally.

6.13. USB Camera Demo

Connect our standard USB Camera to the onboard Cubieboard1/2 USB Host interface, connect the network cable(to ensure access to the Internet), enter corresponding command:

Note: Users can directly start from step 5) as the firmware we provided already finished step 1)- step 4).

- 1) Install corresponding tool

```
$ apt-get install libv4l-dev
$ apt-get install libjpeg8-dev
$ apt-get install subversion
$ apt-get install imagemagick
$ apt-get install make
$ apt-get install vim
```
- 2) Download mjpg-streamer source code

`$cd /`

`$svn co https://svn.code.sf.net/p/mjpg-streamer/code/ mjpg-streamer`

It will download automatically after executed the command, after finish download, the mjpg-streamer document will appeared in the current directory.

- 3) Modify the configuration file

`$vi mjpg-streamer/mjpg-streamer/plugins/input_uvc/input_uvc.c`

Modify:

`int width = 640, height = 480, fps = 5, format = V4L2_PIX_FMT_MJPEG, i;`

to:

`int width = 320, height = 240, fps = 5, format = V4L2_PIX_FMT_YUYV, i;`

Exit after save it.

- 4) Compile

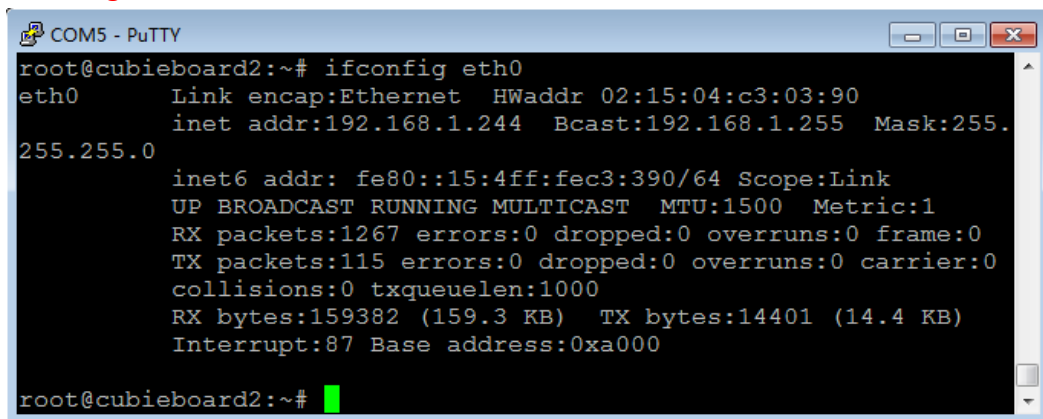
`$cd /mjpg-streamer/mjpg-streamer`

`$make clean`

`$make`

- 5) Check the assigned ip address:

`$ifconfig eth0`



```
COM5 - PuTTY
root@cubieboard2:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:15:04:c3:03:90
          inet addr:192.168.1.244  Bcast:192.168.1.255  Mask:255.
          255.255.0
          inet6 addr: fe80::15:4ff:fec3:390/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1267  errors:0  dropped:0  overruns:0  frame:0
          TX packets:115  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:159382 (159.3 KB)  TX bytes:14401 (14.4 KB)
          Interrupt:87 Base address:0xa000

root@cubieboard2:~#
```

The ip address is 192.168.1.244.

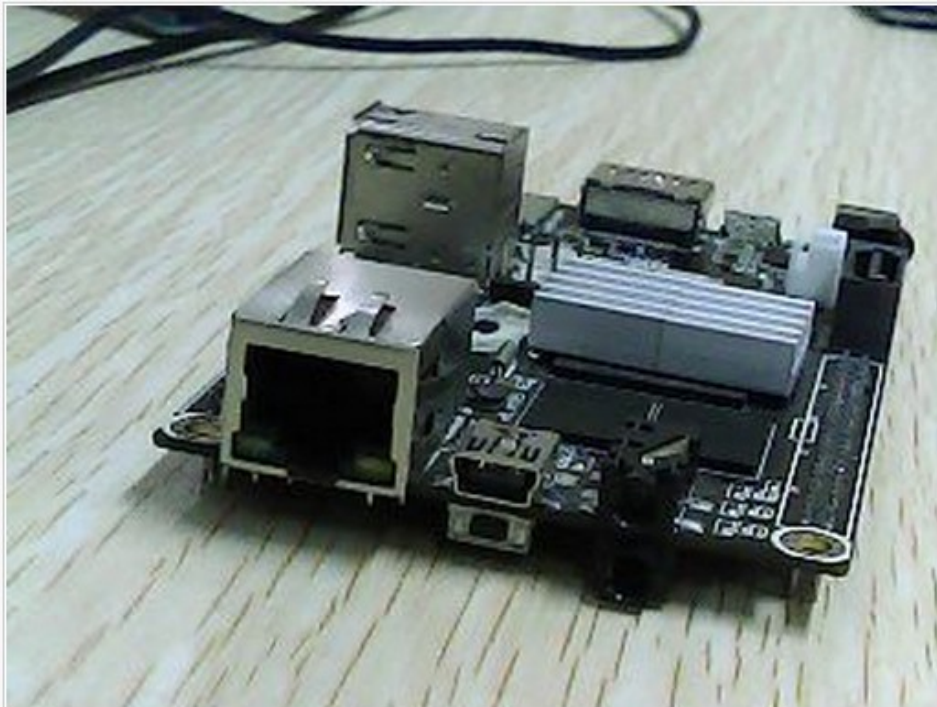
- 6) Check the assigned ip address:

`$cd /mjpg-streamer/mjpg-streamer`

`$./start.sh`

```
COM5 - PuTTY
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
```

- 7) Open the browser on a computer which is connected to the same subnet(or directly on the LCD display which is connected to DVK522), enter the following address: <http://192.168.1.244:8080/javascript.html>
You can see the captured video stream. Press Ctrl+C to stop.



© The **MJPEG-streamer** team | Design by **Andreas Viklund**

6.14. WIFI Demo

Insert our standard USB WIFI to the Cubieboard1/2 onboard USB Host connector, enter corresponding command on the console terminal:

- 1) Install corresponding tools

```
$apt-get install wifi-radar
```

```
$apt-get install linux-firmware
```

When finished, reboot the system.

- 2) After rebooted the system, turn off the Ethernet card and open WIFI card.

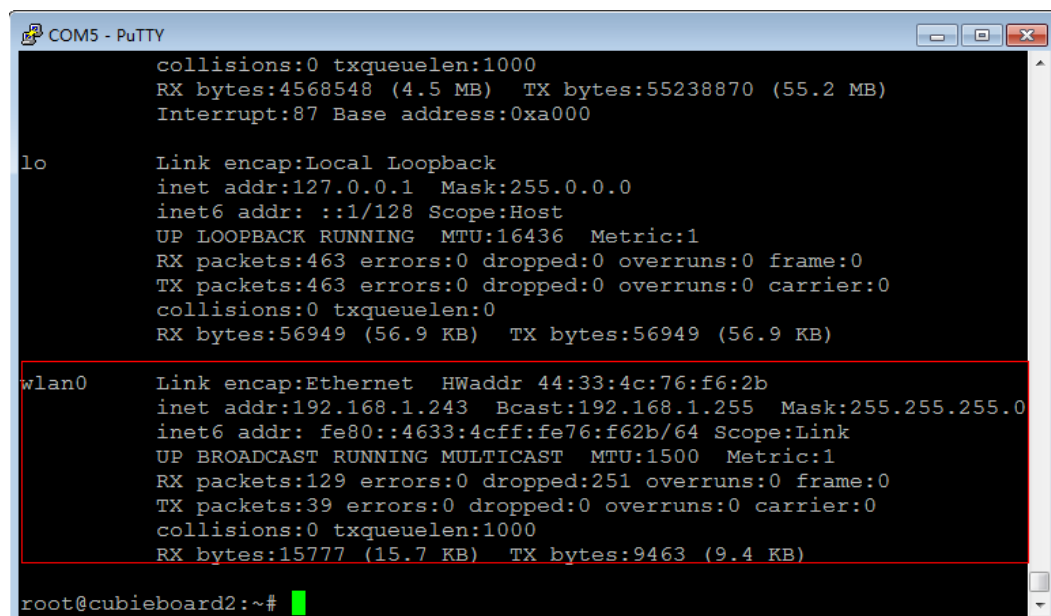
```
$ifconfig eth0 down
```

```
$ifconfig wlan0 up
```

Note: It may be "wlan1" in different system, according to the actual situation.

- 3) Check the network status:

```
$ifconfig
```



```
COM5 - PuTTY
collisions:0 txqueuelen:1000
RX bytes:4568548 (4.5 MB) TX bytes:55238870 (55.2 MB)
Interrupt:87 Base address:0xa000

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:16436 Metric:1
  RX packets:463 errors:0 dropped:0 overruns:0 frame:0
  TX packets:463 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:56949 (56.9 KB) TX bytes:56949 (56.9 KB)

wlan0
  Link encap:Ethernet HWaddr 44:33:4c:76:f6:2b
  inet addr:192.168.1.243 Bcast:192.168.1.255 Mask:255.255.255.0
  inet6 addr: fe80::4633:4cff:fe76:f62b/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:129 errors:0 dropped:251 overruns:0 frame:0
  TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:15777 (15.7 KB) TX bytes:9463 (9.4 KB)

root@cubieboard2:~#
```

- 4) Scan the wireless router:

```
$ iwlist wlan0 scan
```

to find available wireless network.

```
COM5 - PuTTY
ESSID:"HP-Print-1E-Officejet Pro 8600"
Protocol:IEEE 802.11bg
Mode:Master
Frequency:2.437 GHz (Channel 6)
Encryption key:on
Bit Rates:54 Mb/s
Extra:rsn_ie=30140100000fac040100000fac040100000fac020
000
IE: IEEE 802.11i/WPA2 Version 1
  Group Cipher : CCMP
  Pairwise Ciphers (1) : CCMP
  Authentication Suites (1) : PSK
Quality=101/100 Signal level=-74 dBm
Cell 06 - Address: 2C:B0:5D:25:33:71
ESSID:"NETGEAR17"
Protocol:IEEE 802.11bgn
Mode:Master
Frequency:2.447 GHz (Channel 8)
Encryption key:on
Bit Rates:144 Mb/s
Extra:rsn_ie=30140100000fac040100000fac040100000fac020
c00
IE: IEEE 802.11i/WPA2 Version 1
  Group Cipher : CCMP
  Pairwise Ciphers (1) : CCMP
  Authentication Suites (1) : PSK
IE: Unknown: DD310050F204104A000110104400010210470010F
B0548E7C1A08D76293FB64054D1CFAF103C0001031049000600372A000120
Quality=101/100 Signal level=-63 dBm
root@cubieboard2:~#
```

5) Configure wlan0:

`$vi /etc/network/interfaces`

Acquire the IP configuration dynamically as following:

```
#auto lo eth0
#iface lo inet loopback
#iface eth0 inet dhcp

auto wlan0
iface wlan0 inet dhcp
pre-up ip link set wlan0 up
pre-up iwconfig wlan0 essid waveshareNet
wpa-ssid waveshareNet
wpa-psk 123456
```

Acquire the IP configuration statically as following:

```
#auto lo eth0
#iface lo inet loopback
#iface eth0 inet dhcp

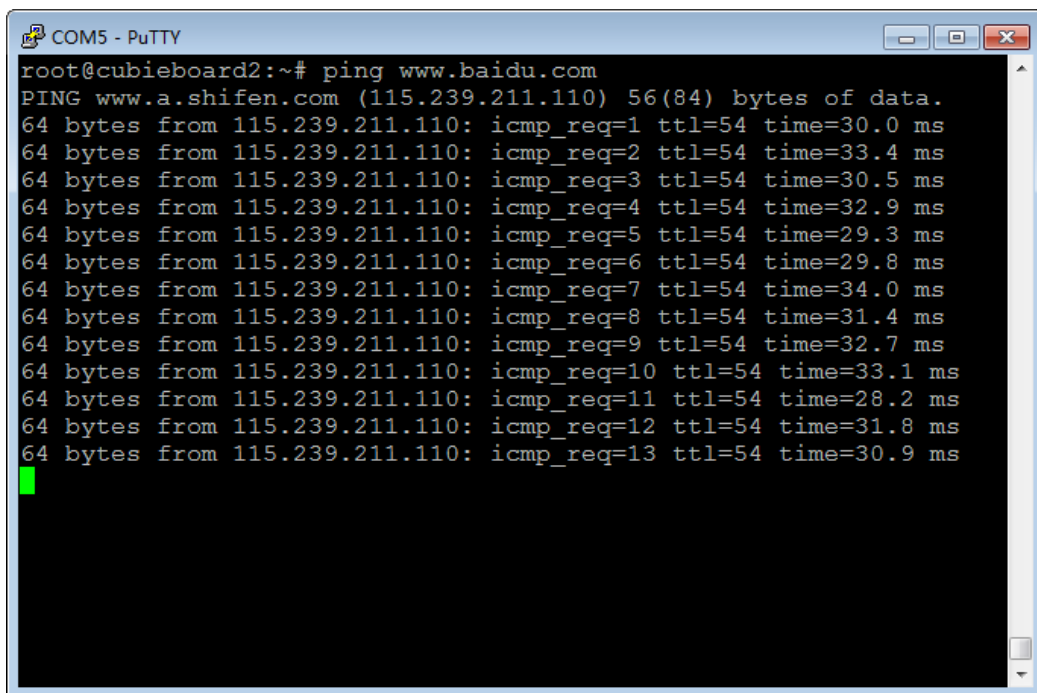
auto wlan0
iface wlan0 inet static
address 192.168.1.121
```

```
netmask 255.255.255.0
gateway 192.168.1.1
pre-up ip link set wlan0 up
pre-up iwconfig wlan0 essid waveshareNet
wpa-ssid waveshareNet
wpa-psk 123456
```

Exit after save it.

Note: Configure ESSID and PSK properly according to the scan result.

- 6) Restart the network:
\$ /etc/init.d/networking restart
- 7) Restart the network:
\$ ping www.baidu.com



```
COM5 - PuTTY
root@cubieboard2:~# ping www.baidu.com
PING www.a.shifen.com (115.239.211.110) 56(84) bytes of data.
64 bytes from 115.239.211.110: icmp_req=1 ttl=54 time=30.0 ms
64 bytes from 115.239.211.110: icmp_req=2 ttl=54 time=33.4 ms
64 bytes from 115.239.211.110: icmp_req=3 ttl=54 time=30.5 ms
64 bytes from 115.239.211.110: icmp_req=4 ttl=54 time=32.9 ms
64 bytes from 115.239.211.110: icmp_req=5 ttl=54 time=29.3 ms
64 bytes from 115.239.211.110: icmp_req=6 ttl=54 time=29.8 ms
64 bytes from 115.239.211.110: icmp_req=7 ttl=54 time=34.0 ms
64 bytes from 115.239.211.110: icmp_req=8 ttl=54 time=31.4 ms
64 bytes from 115.239.211.110: icmp_req=9 ttl=54 time=32.7 ms
64 bytes from 115.239.211.110: icmp_req=10 ttl=54 time=33.1 ms
64 bytes from 115.239.211.110: icmp_req=11 ttl=54 time=28.2 ms
64 bytes from 115.239.211.110: icmp_req=12 ttl=54 time=31.8 ms
64 bytes from 115.239.211.110: icmp_req=13 ttl=54 time=30.9 ms
```

Note:
Users can connect WIFI directly on graphical interface.