4G module via USB

Introduction

BPI-4G module extension board with 1 Non-standard micro USB port,you can connect banana pi board with this port ,use 4G function development . also support PIN define ,and connect to BPI-R2. user also can choose 4G module onboard and use the universal 4G module in the market.

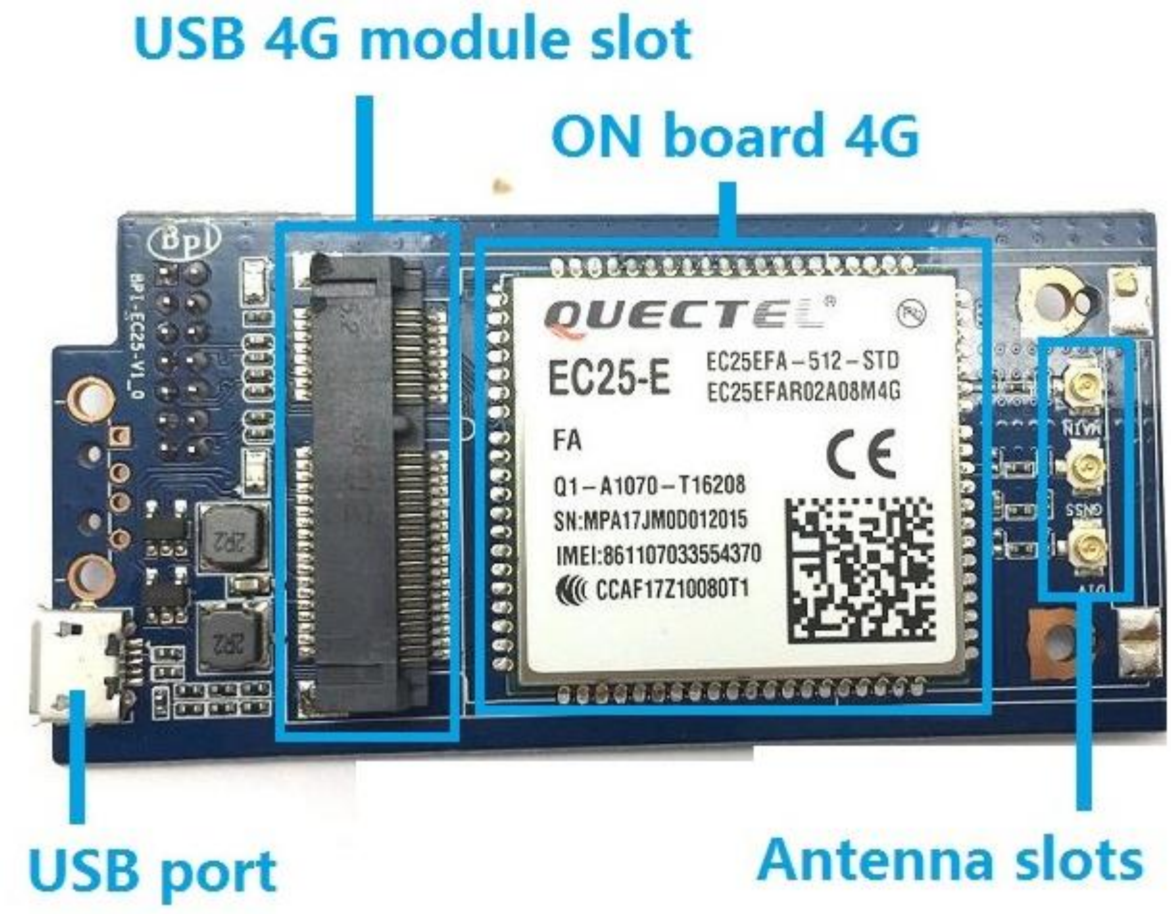
Using 4G module with BananaPi

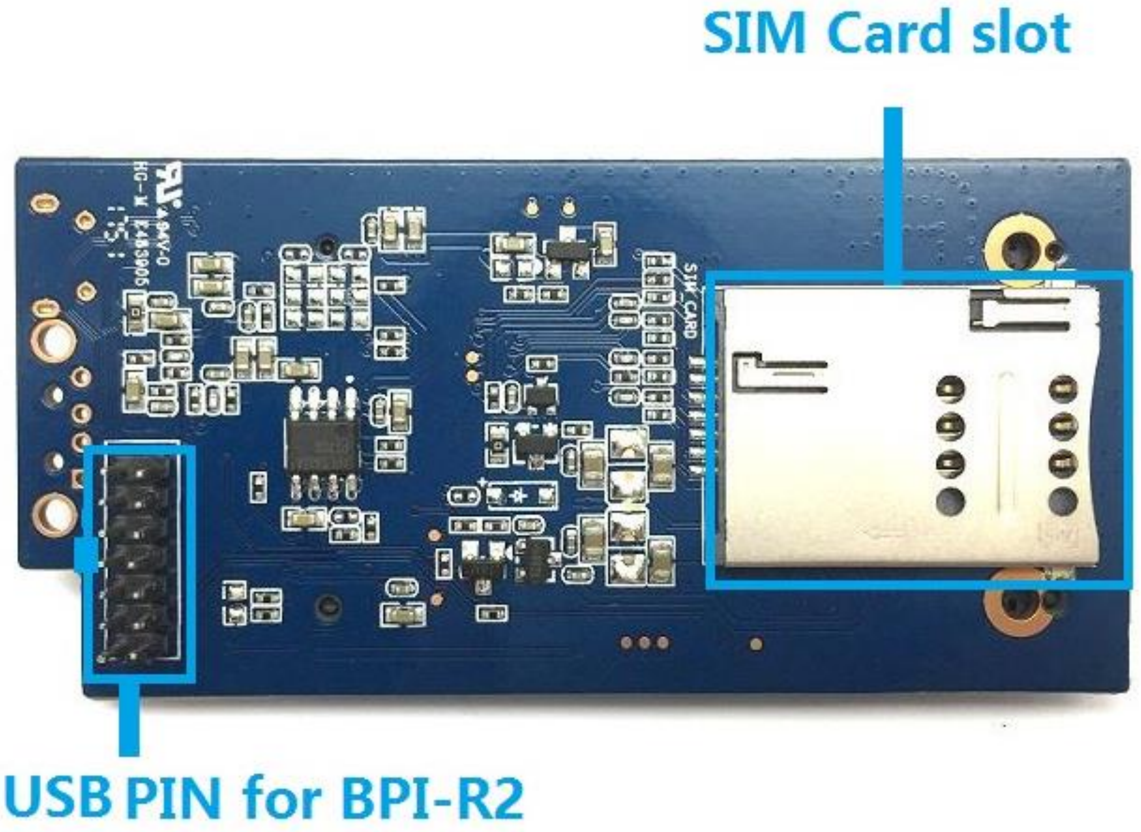
- Using 4G module with BananaPi

Hardware

hardware interface

BPI-4G module (4G on board)



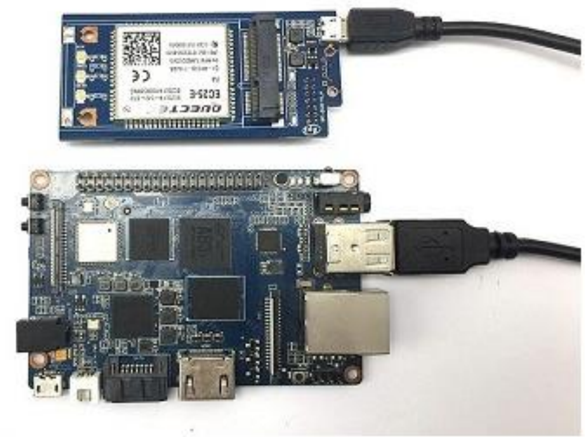


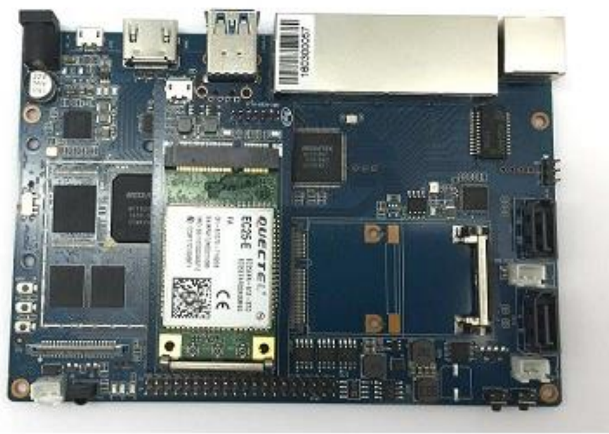
Hardware spec

BPI-4G module (4G on board)

How to use 4G

4G board connect banana pi SBC with USB line ,and connect BPI-R2 with PIN define

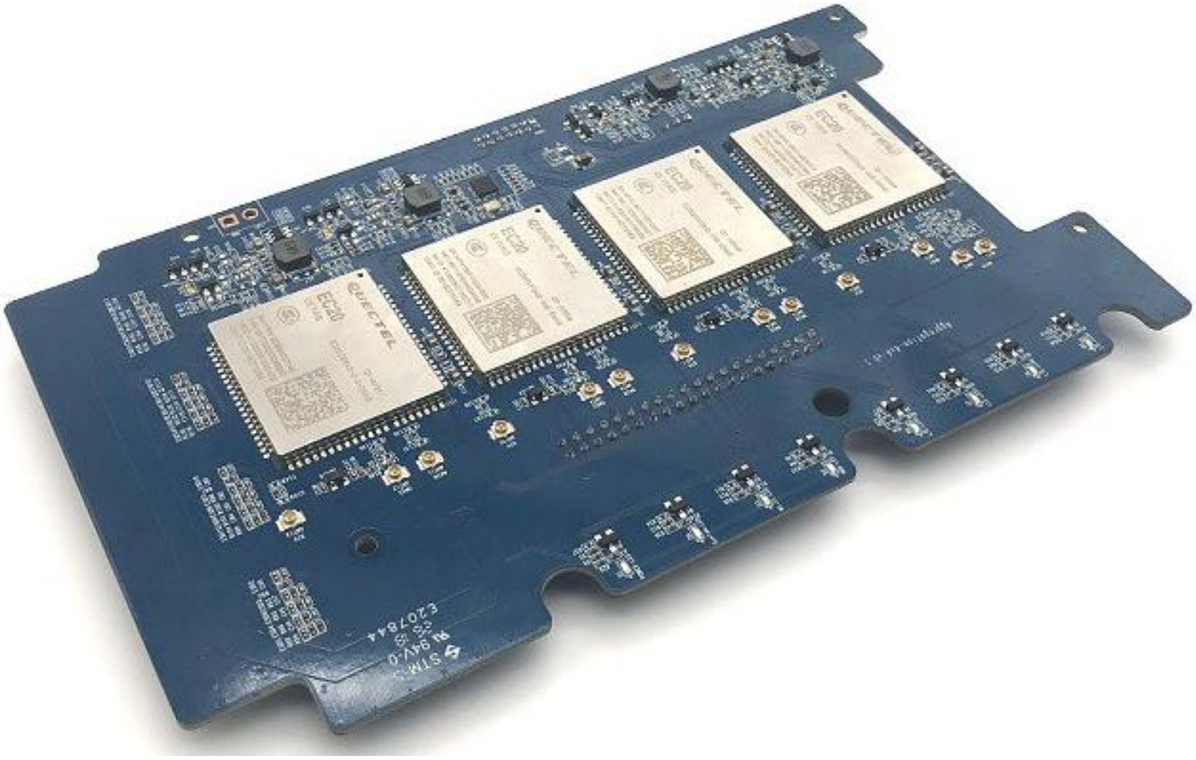




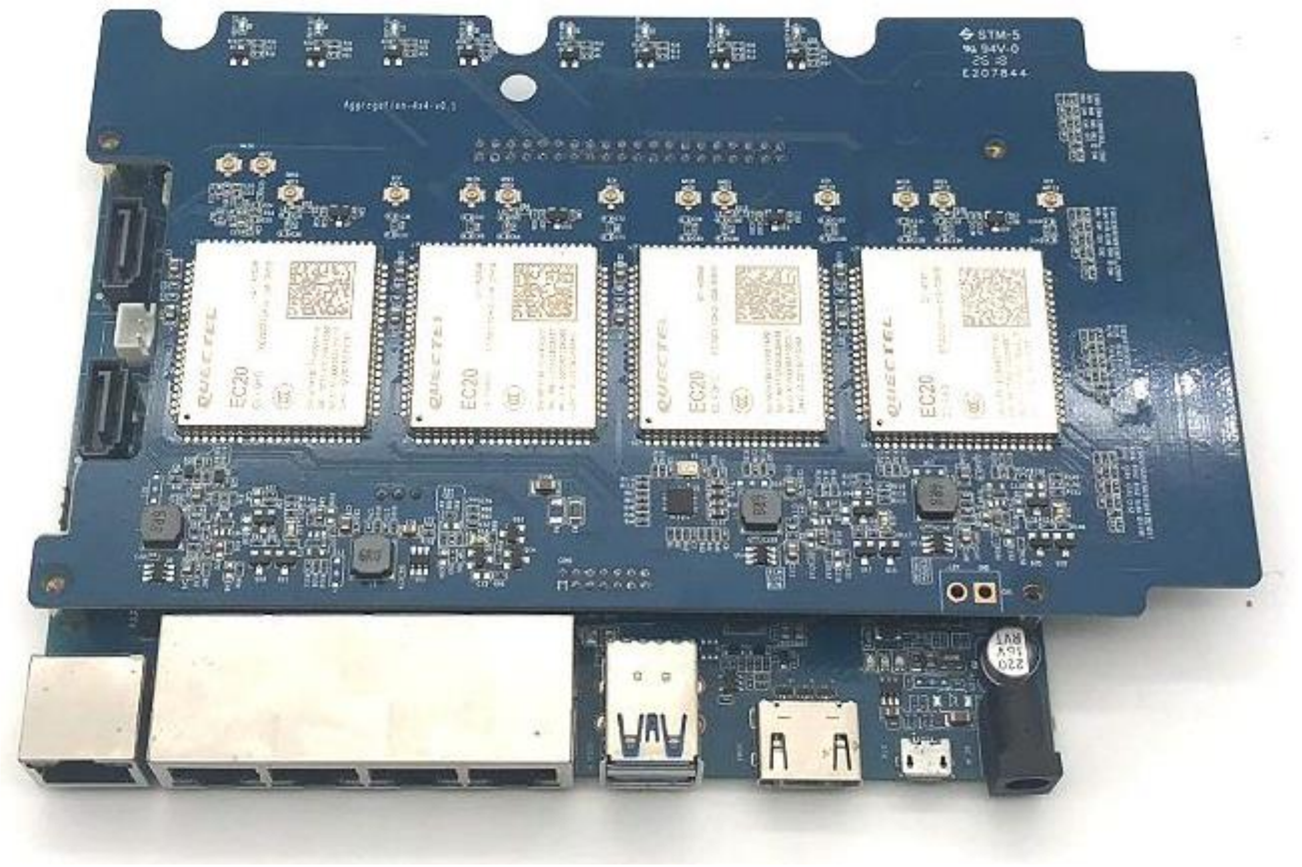
BPI 4.0:customization

if you have any 4G customization , we can support you.

4 4G module board via USB interface.

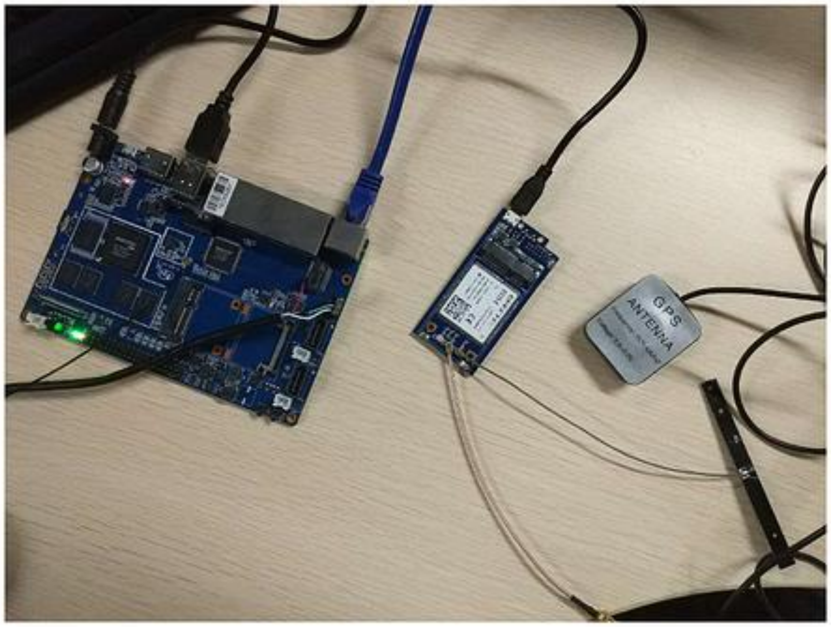


use this module on BPI-R2

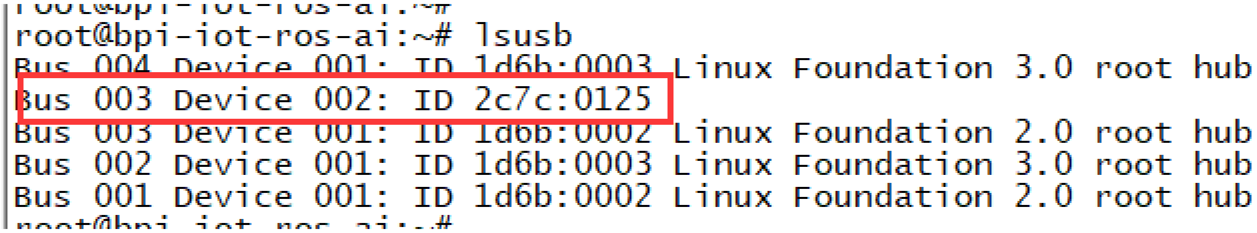


Get startting

- forum pthread for discuss : http://forum.banana-pi.org/t/how-to-use-bpi-4g-lte-module-with-bpi-r2/5223/4

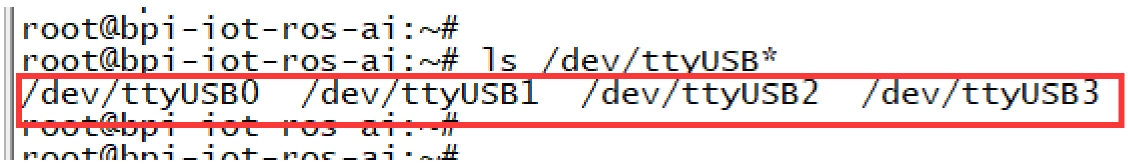


1.Firstly, use R2 connects LTE module by USB or 14pin Headers. then execute command "lsusb" to have a look if R2 have drove the module. The red box is the id of 4G module.



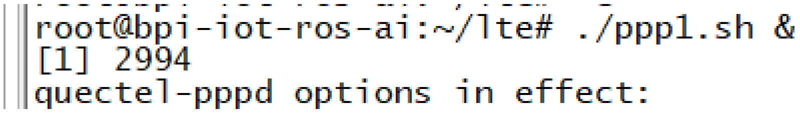
```
root@bpi-iot-ros-ai:~#
root@bpi-iot-ros-ai:~# lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 2c7c:0125
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@bpi-iot-ros-ai:~#
```

After seeing this, execute command "ls /dev/ttyUSB*" to make sure the 4G module have been recognised as USB devs.



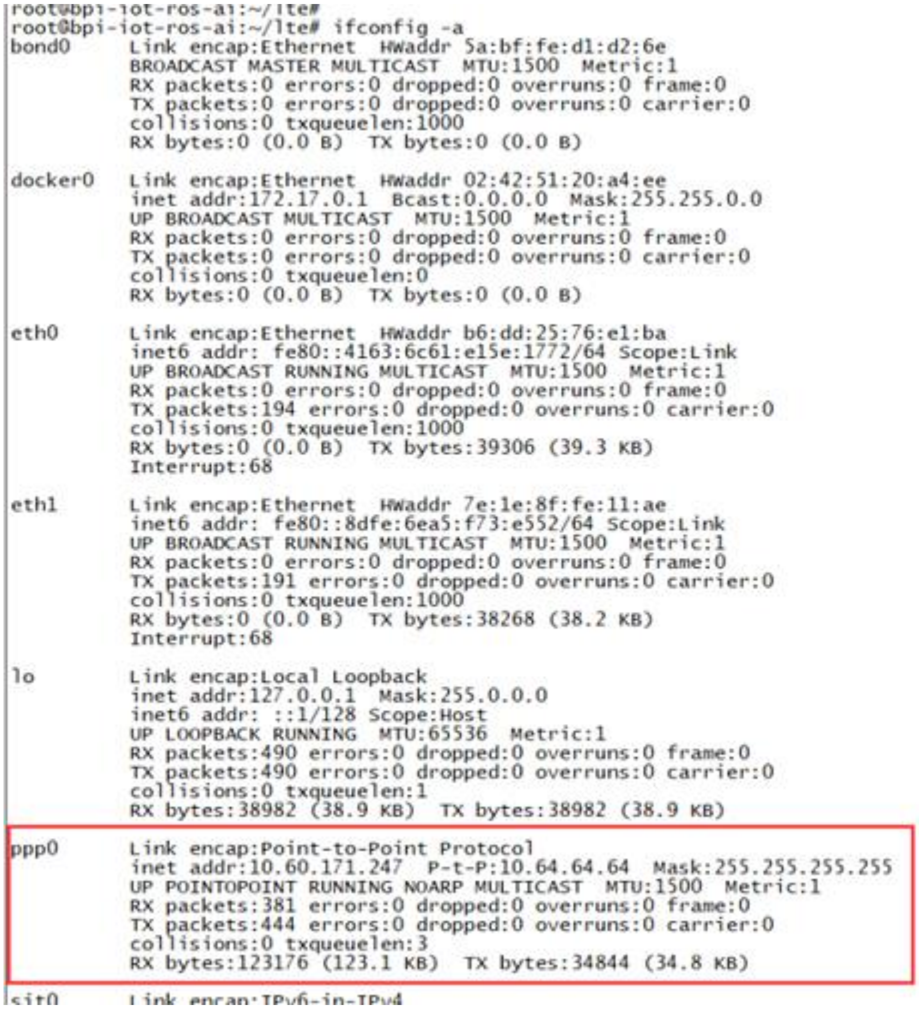
```
root@bpi-iot-ros-ai:~#
root@bpi-iot-ros-ai:~# ls /dev/ttyUSB*
/dev/ttyUSB0  /dev/ttyUSB1  /dev/ttyUSB2  /dev/ttyUSB3
root@bpi-iot-ros-ai:~#
root@bpi-iot-ros-ai:~#
```

2. Execute the "./ppp1.sh &"



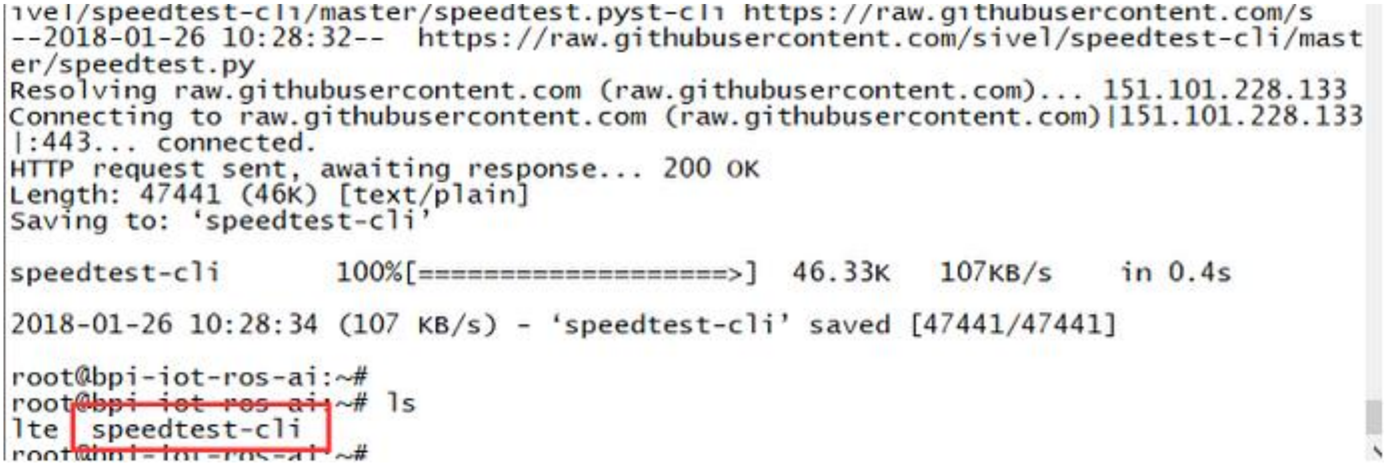
```
root@bpi-iot-ros-ai:~/lte# ./ppp1.sh &
[1] 2994
quectel-pppd options in effect:
```

then execute command “ifconfig -a”,   you will see ppp0



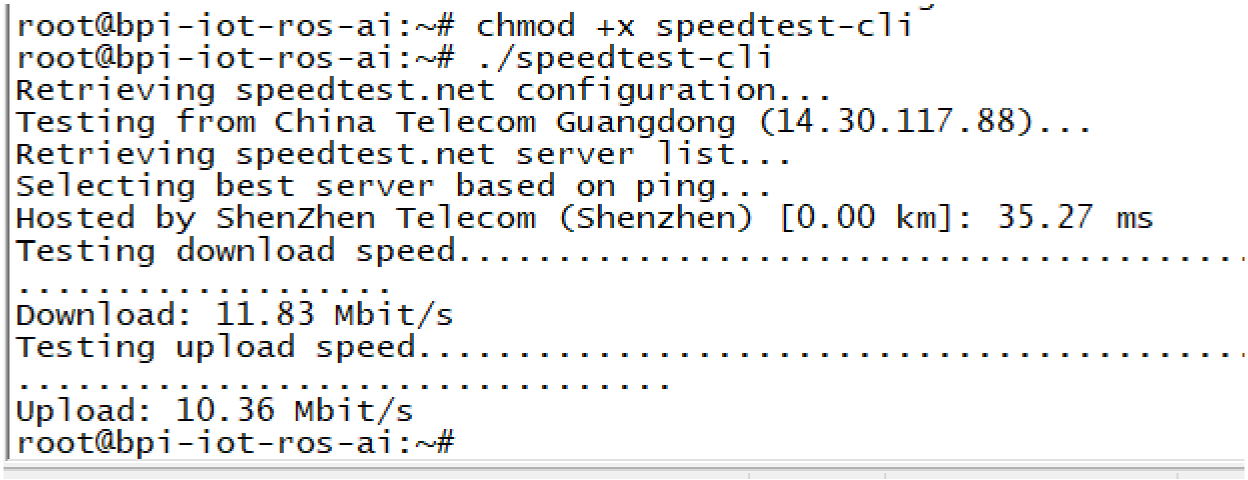
```
root@bpi-iot-ros-ai:~/lte#
root@bpi-iot-ros-ai:~/lte# ifconfig -a
bond0     Link encap:Ethernet  HWaddr 5a:bf:fe:d1:d2:6e
          BROADCAST MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

docker0   Link encap:Ethernet  HWaddr 02:42:51:20:a4:ee
          inet addr:172.17.0.1  Bcast:0.0.0.0  Mask:255.255.0.0
          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:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0      Link encap:Ethernet  HWaddr b6:dd:25:76:e1:ba
          inet6 addr: fe80::4163:6c61:e15e:1772/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:194 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:39306 (39.3 KB)
          Interrupt:68

eth1      Link encap:Ethernet  HWaddr 7e:1e:8f:fe:11:ae
          inet6 addr: fe80::8dfe:6ea5:f73:e552/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:191 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:38268 (38.2 KB)
          Interrupt:68

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:490 errors:0 dropped:0 overruns:0 frame:0
          TX packets:490 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:38982 (38.9 KB)  TX bytes:38982 (38.9 KB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.60.171.247  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:381 errors:0 dropped:0 overruns:0 frame:0
          TX packets:444 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:123176 (123.1 KB)  TX bytes:34844 (34.8 KB)

sit0      Link encap:IPv6-in-IPv4
```

3. Execute command ”wget -O speedtest-cli https://raw.githubusercontent.com/sivel/speedtest-cli/master/speedtest.py” to download speed test tool.



```
ivel/speedtest-cli/master/speedtest.pyst-cli https://raw.githubusercontent.com/s
--2018-01-26 10:28:32--  https://raw.githubusercontent.com/sivel/speedtest-cli/mast
er/speedtest.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.228.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.228.133
|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 47441 (46K) [text/plain]
Saving to: ‘speedtest-cli’

speedtest-cli       100%[===================>]  46.33K   107KB/s    in 0.4s

2018-01-26 10:28:34 (107 KB/s) - ‘speedtest-cli’ saved [47441/47441]

root@bpi-iot-ros-ai:~#
root@bpi-iot-ros-ai:~# ls
lte  speedtest-cli
root@bpi-iot-ros-ai:~#
```

4. Execute command "chmod +x speedtest-cli", then execute "./speedtest-cli"



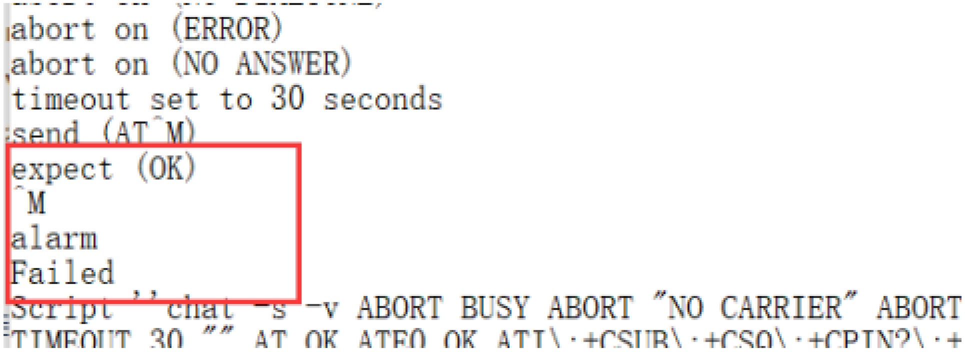
```
root@bpi-iot-ros-ai:~# chmod +x speedtest-cli
root@bpi-iot-ros-ai:~# ./speedtest-cli
Retrieving speedtest.net configuration...
Testing from China Telecom Guangdong (14.30.117.88)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by ShenZhen Telecom (Shenzhen) [0.00 km]: 35.27 ms
Testing download speed.........................................
....................
Download: 11.83 Mbit/s
Testing upload speed...........................................
.................................
Upload: 10.36 Mbit/s
root@bpi-iot-ros-ai:~#
```

ppp1_success_log.txt ： http://forum.banana-pi.org/uploads/default/original/2X/e/eb7d46a0971c25f6337116d60e76590ce298b676.txt

FAQ

- (1) You could get ppp1.sh from here : https://pan.baidu.com/s/1yu0nTNQKPNPzprVPjhOPUA

- (2) Q: What should I do, if you meet this "expect (OK) ^M alarm Failed "



```
abort on (ERROR)
abort on (NO ANSWER)
timeout set to 30 seconds
send (AT^M)
expect (OK)
^M
alarm
Failed
Script '' chat -s -v ABORT BUSY ABORT ″NO CARRIER″ ABORT
TIMEOUT 30 ″″ AT OK ATE0 OK ATI\:+CSUB\:+CSQ\:+CPIN?\:+
```

A:The root cause is the Modem.service has used the module usbserial, so the 4G module can't use usbserial; we can solve the problem by using command "systemctl disable ModemManager.service " or "systemctl stop ModemManager.service "

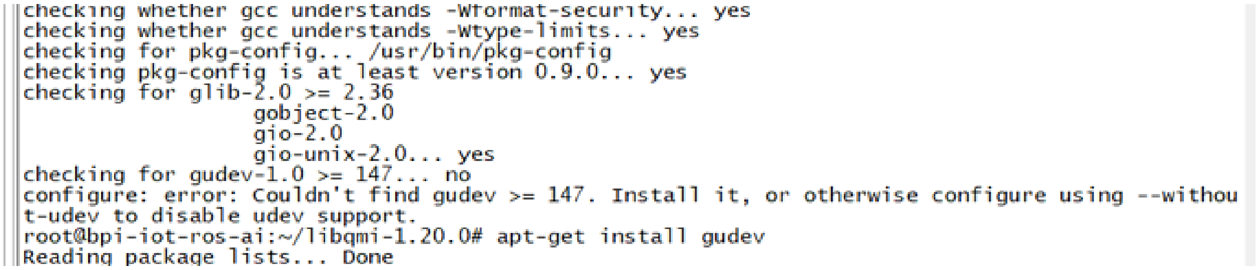
- (3) Q：How to test the communication between SIM card and 4G module？

A:Use minicom, ”minicom -D /dev/ttyUSB2”, and then use AT commands to test, such as ”at”, ”at+csq”, “at+ccid”, if you get response without error, that means the communication is OK.

- (4) Q: How to install libqmi?

A: Download from here: http://www.linuxfromscratch.org/blfs/view/cvs/general/libqmi.html 1 ; then make and make install.

If you meet this:



```
checking whether gcc understands -Wformat-security... yes
checking whether gcc understands -Wtype-limits... yes
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for glib-2.0 >= 2.36
                  gobject-2.0
                  gio-2.0
                  gio-unix-2.0... yes
checking for gudev-1.0 >= 147... no
configure: error: Couldn't find gudev >= 147. Install it, or otherwise configure using --withou
t-udev to disable udev support.
root@bpi-iot-ros-ai:~/libqmi-1.20.0# apt-get install gudev
Reading package lists... Done
```

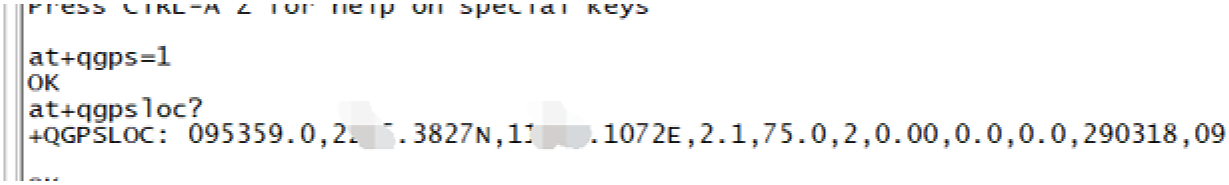
you need to install libgudev-1.0-dev, then make and make install again.

- (5) Q：How to use qmi-wwan？

A：There is a very easy way to use, I've put quectel-CM in home path of root, just execute "./quectel-CM &", then execute "dhclient wwan0", now you have finished the configurations.

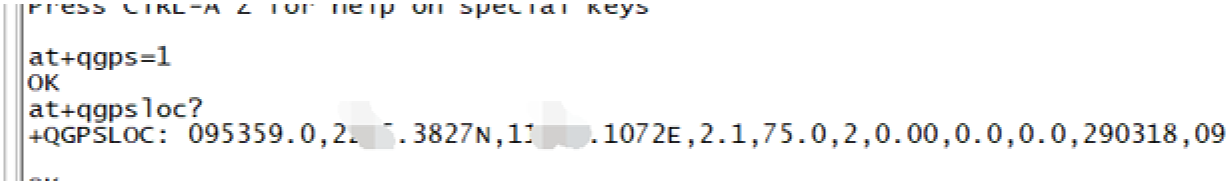
- (6) Q：How to use GPS？

A：BPI-4G module supports GPS and BeiDou, execute "minicom -D /dev/ttyUSB2", then input AT commands like this：



```
Press CTRL-A Z for help on special keys

at+qgps=1
OK
at+qgpsloc?
+QGPSLOC: 095359.0,2[redacted].3827N,1[redacted].1072E,2.1,75.0,2,0.00,0.0,0.0,290318,09

OK
```

if you want to use BeiDou, just input AT commands like this：



```
Press CTRL-A Z for help on special keys

at+qgps=1
OK
at+qgpsloc?
+QGPSLOC: 095359.0,2[redacted].3827N,1[redacted].1072E,2.1,75.0,2,0.00,0.0,0.0,290318,09

OK
```

- (7) Here are kernel configurations and driver patch for kernel 4.14.

config : http://forum.banana-pi.org/uploads/default/original/2X/5/54a89518b88857f06fd44757d3413e700338b269
patch : http://forum.banana-pi.org/uploads/default/original/2X/5/5739bb02c8dc5817ed8ba6ec466b4bdb70cf37fa

- (8) Support areas:

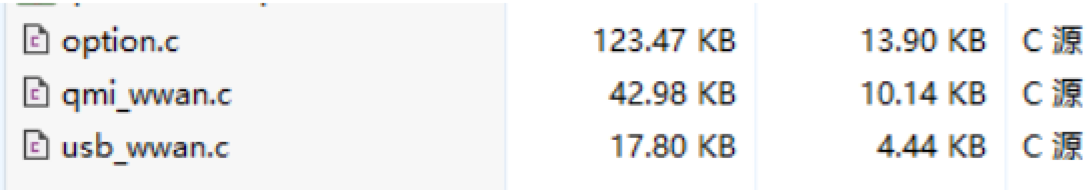
EC25-E support: EMEA/Thailand/India/South Korea
EC25-AU support: Australia
EC25-V/-A support：North America

- (9) Openwrt Supported:

A. Clone the source code from this link : BPI-R2 OpenWrt(LEDE) Souce code : 2018-04-11
B. Add qmi-wwan driver to this project :

Execute "./scripts/feeds/ install minicom" to install minicom to openwrt project
Execute "./scripts/feeds/ install python" to install python to openwrt project
Execute "make menuconfig"
Add minicom "Utilities —> Terminal —> minicom"
Add python "Languages —> Python"

Add driver “Network —> wwan …Genetic OpenWrt 3G/4G proto handler”
Add "Kernel modules —> USB Support —>
kmod-usb-core
kmod-usb-net —> kmod-usb-net-qmi-wwan
kmod-usb-ohci
kmod-usb-serial —> kmod-usb-serial-option , kmod-usb-serial-wwan
kmod-usb-uhci
kmod-usb2-*
Use these source files to replace your openwrt project source files (“build_dir/target-arm_cortex-a7+neon-vfpv4_musl_eabi/linux-mediatek_32/linux-4.9.44/drivers/usb(or net)/…”) :



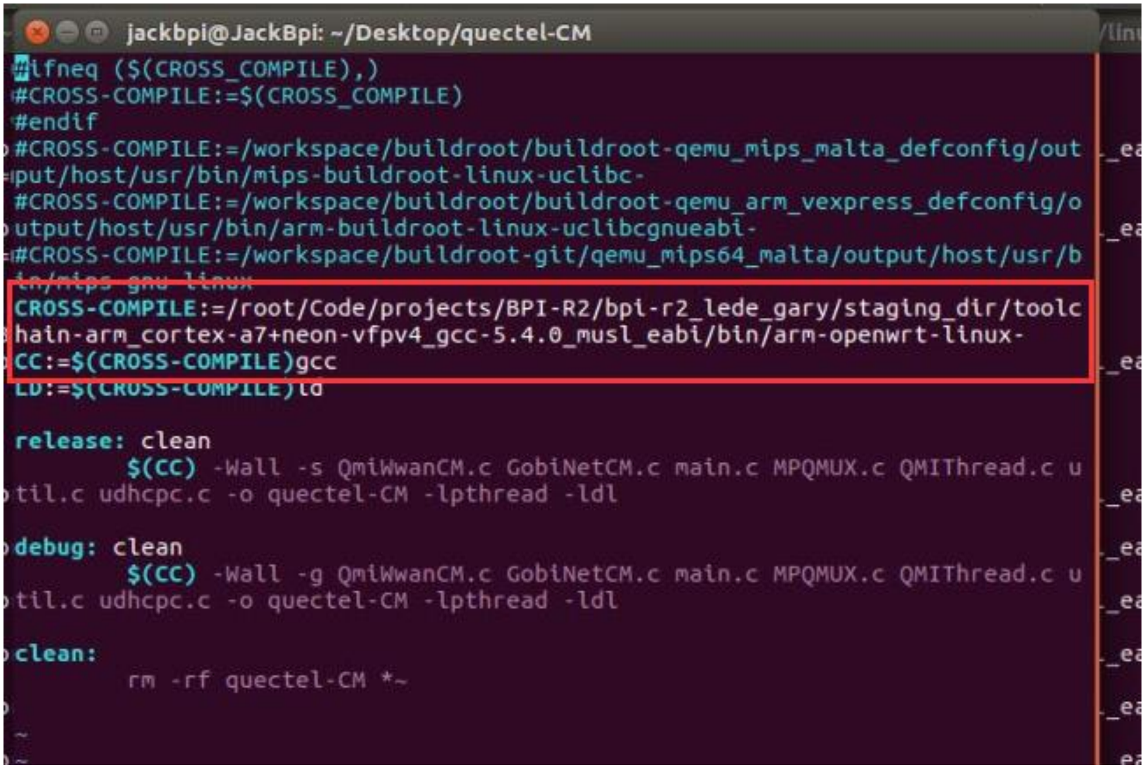
option.c	123.47 KB	13.90 KB	C 源
qmi_wwan.c	42.98 KB	10.14 KB	C 源
usb_wwan.c	17.80 KB	4.44 KB	C 源

C. Image & Stuff Link :

Baidu Drive : https://pan.baidu.com/s/1e6bvsznfFW3PuaONz0OR1Q
Google Drive : https://drive.google.com/file/d/1Zjj0TiZOFGlmAGmRmErqnwdSosUrLUPZ/view?usp=sharing

D. Cross-Compile quectel-CM

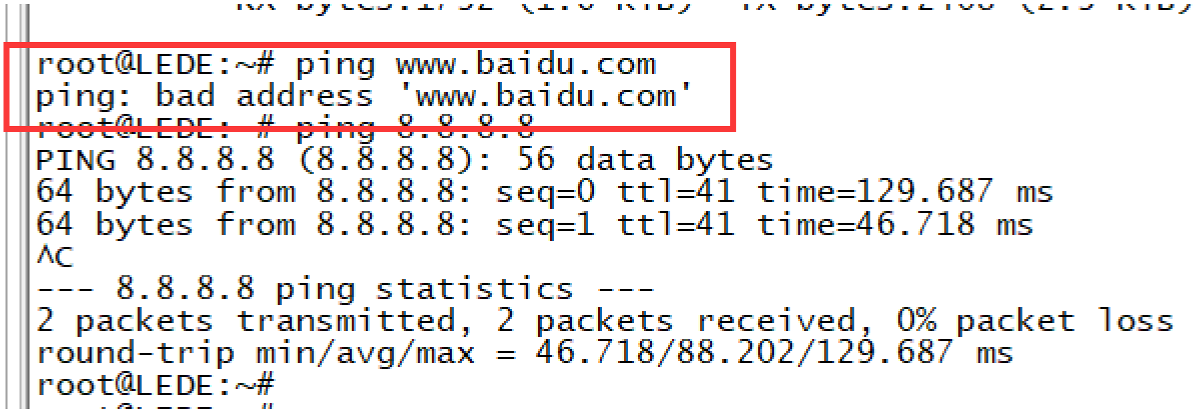
Copy “quectel-CM” to Linux host PC， and change “Makefile” under the “quectel-CM”, change tools like this：The tools are under openwrt project， you can find them according to the following picture.



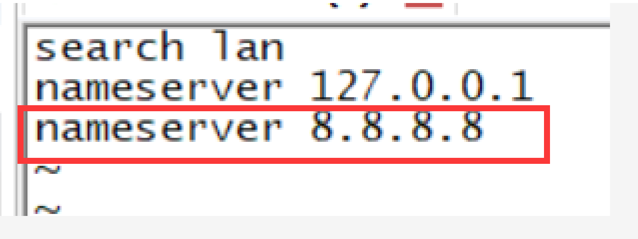
jackbpi@JackBpi: ~/Desktop/quectel-CM

```
#ifneq ($(CROSS_COMPILE),)
#CROSS-COMPILE:=$(CROSS_COMPILE)
#endif
#CROSS-COMPILE:=/workspace/buildroot/buildroot-qemu_mips_malta_defconfig/out
put/host/usr/bin/mips-buildroot-linux-uclibc-
#CROSS-COMPILE:=/workspace/buildroot/buildroot-qemu_arm_vexpress_defconfig/o
utput/host/usr/bin/arm-buildroot-linux-uclibcgnueabi-
#CROSS-COMPILE:=/workspace/buildroot-git/qemu_mips64_malta/output/host/usr/b
in/mips-gnu-linux
CROSS-COMPILE:=/root/Code/projects/BPI-R2/bpi-r2_lede_gary/staging_dir/toolc
hain-arm_cortex-a7+neon-vfpv4_gcc-5.4.0_musl_eabi/bin/arm-openwrt-linux-
CC:=$(CROSS-COMPILE)gcc
LD:=$(CROSS-COMPILE)ld

release: clean
        $(CC) -Wall -s QmiWwanCM.c GobiNetCM.c main.c MPQMUX.c QMIThread.c u
til.c udhcpc.c -o quectel-CM -lpthread -ldl

debug: clean
        $(CC) -Wall -g QmiWwanCM.c GobiNetCM.c main.c MPQMUX.c QMIThread.c u
til.c udhcpc.c -o quectel-CM -lpthread -ldl

clean:
        rm -rf quectel-CM *~

~
```

If ping Domain name fail， add “nameserver 8.8.8.8” to /etc/resolv.conf



```
root@LEDE:~# ping www.baidu.com
ping: bad address 'www.baidu.com'
root@LEDE:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=41 time=129.687 ms
64 bytes from 8.8.8.8: seq=1 ttl=41 time=46.718 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 46.718/88.202/129.687 ms
root@LEDE:~#
```



```
search lan
nameserver 127.0.0.1
nameserver 8.8.8.8
~
~
```

- (10) If you use mPCIE interface 4G module to connect R2 by 14 pin header, use these cmds to enable 4G module, after you did these steps, USB0 port will be disable.

cd /sys/class/gpio/

echo 266 > export 【# 266(gpio_offset) = 232(base) + 34(GPIO_34)】

cd gpio266/

echo out > direction

echo 1 > value