

## 1 Introduction

This application note describes:

- How to set the 5G environment on Layerscape platforms.
- How to connect 5G modules to Layerscape platforms.
- How to install the software required for 5G modules and 5G test on OK1028A-C v1.1 and OK1046A-C2 v1.1 boards. The boards come from a third-part company, Forlinx, and they can be bought at <https://www.forlinx.com/>.
- The test process and the reference result of the test.

### 1.1 Layerscape platform

- OK1028A-C v1.1
- OK1046A-C2 v1.1

### 1.2 5G modules

- Gosuncn 5G module GM800
- Fibocom 5G module FM150-AE-01
- Quectel 5G module RM500Q-CN, RM500U-CN

### 1.3 Benchmark

- Latency
- Downlink rate
- Uplink rate

## 2 Setting hardware environment

### 2.1 OK1046A-C2 v1.1 setup

OK1046A contains one M.2 interface and one SIM card socket, which can be used to test 5G module directly. The board is as shown in [Figure 1](#).

### Contents

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
1.1	Layerscape platform.....	1
1.2	5G modules.....	1
1.3	Benchmark.....	1
<b>2</b>	<b>Setting hardware environment.....</b>	<b>1</b>
2.1	OK1046A-C2 v1.1 setup.....	1
2.2	OK1028A-C v1.1 setup.....	3
<b>3</b>	<b>Setting software environment.....</b>	<b>4</b>
3.1	Enabling Kernel option for 5G.....	5
3.2	Installing 5G module software.....	6
3.3	Installing test tools.....	7
3.4	Setting date.....	7
3.5	Installing DHCP client tool.....	7
<b>4</b>	<b>Test process.....</b>	<b>8</b>
4.1	Preparing 5G module for USB mode.....	8
4.2	Preparing 5G module for PCIe mode.....	9
4.3	Testing 5G module.....	14
<b>5</b>	<b>Revision history.....</b>	<b>14</b>



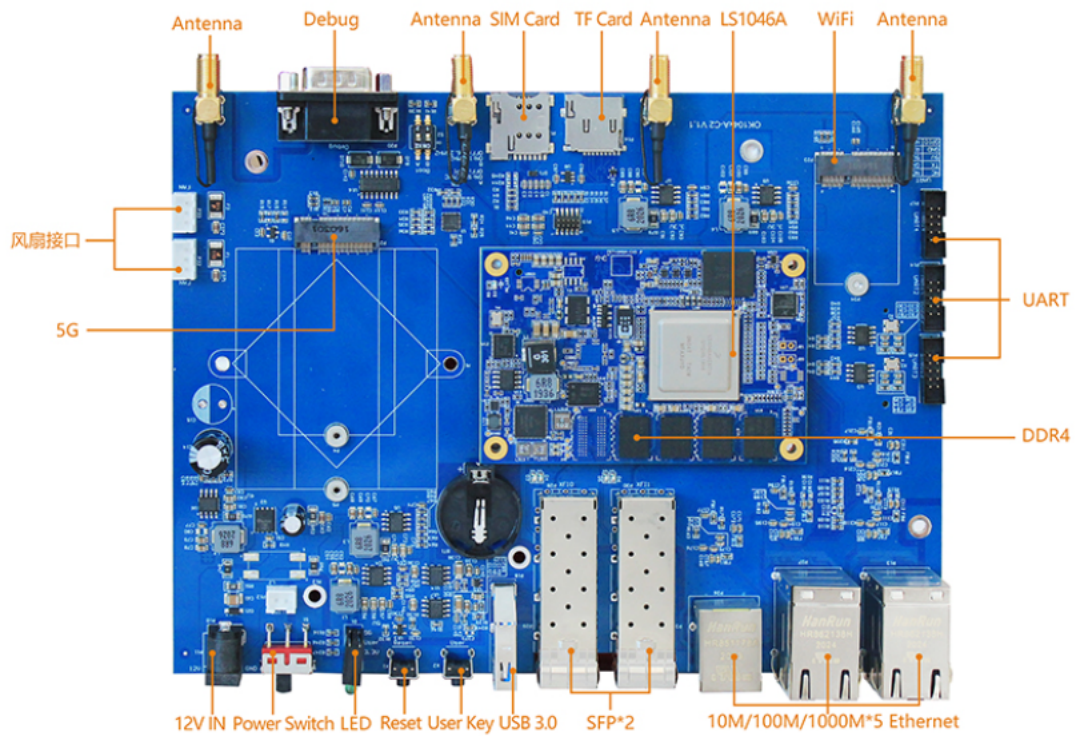


Figure 1. OK1046A board

**NOTE**

Only ok1046a can be used for PCIe mode test.

### 2.1.1 Setting SIM card

Insert the SIM card to the SIM card socket, as shown in [Figure 2](#).

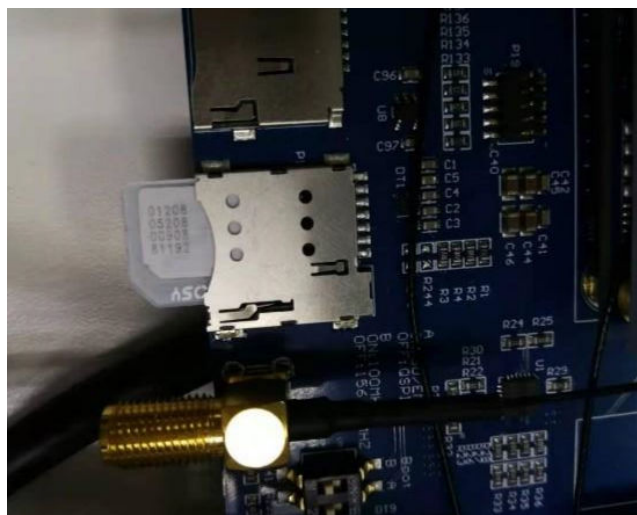


Figure 2. Inserting SIM card

### 2.1.2 Setting 5G module

Insert a 5G module to M.2 interface and connect the antenna array to the 5G module, as shown in [Figure 3](#).

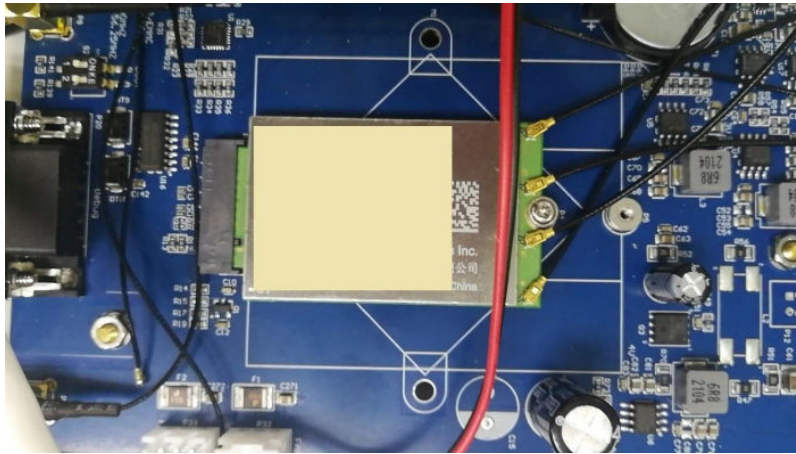


Figure 3. Inserting a 5G module

## 2.2 OK1028A-C v1.1 setup

OK1028A contains one M.2 interface and two SIM card sockets, which can be used to test 5G and 4G module directly. The board is as shown in [Figure 4](#).

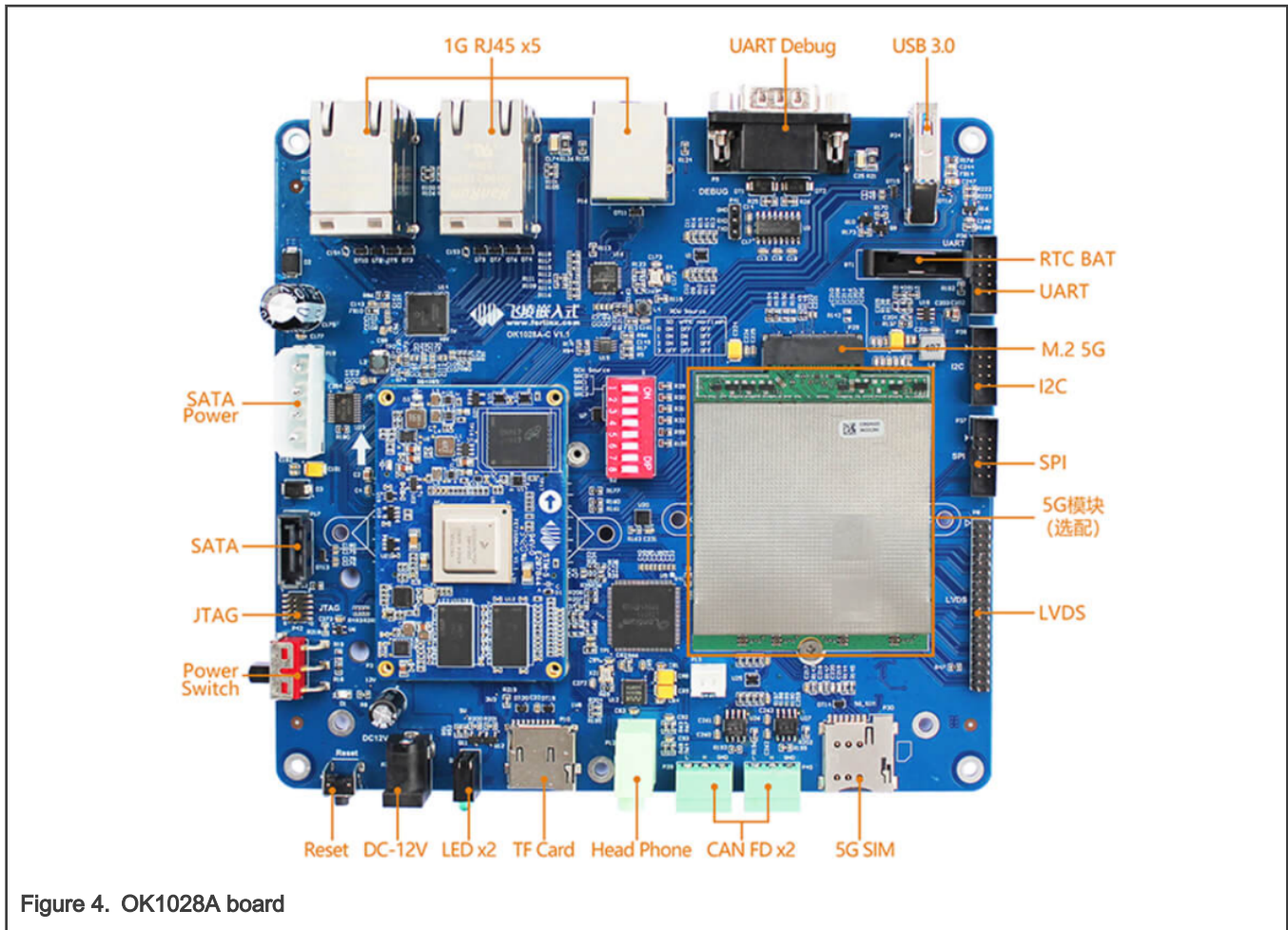


Figure 4. OK1028A board

### 2.2.1 Setting SIM card

Insert the SIM card to the 5G SIM card socket.

**NOTE**

There are two SIM card sockets on the OK1028A board and the top one is used for 5G module. Insert the SIM card to the socket on the **TOP**. For details, refer to [seg 2.1.1](#).

### 2.2.2 Setting 5G module

Insert the Fibocom, Gosuncn, or Quectel 5G module to the M.2 interface and connect the antenna array. For details, refer to [seg 2.2](#).

### 2.2.3 Setting SWITCH

Set the SWITCH correctly to enable the 5G module.

Set the SWITCH to **01101100**.

**NOTE**

If you have any doubts about the board settings, refer to the board documents provided by the board vendor.

## 3 Setting software environment

## 3.1 Enabling Kernel option for 5G

The kernel options enabled below vary with versions of Linux kernel, but the option names are same.

### 3.1.1 Enabling USB Network Framework

Enable kernel option `CONFIG_USB_USBNET` at:

```
Symbol: USB_USBNET [=y]
Type : tristate
Prompt: Multi-purpose USB Networking Framework
Location:
  -> Device Drivers
    -> Network device support (NETDEVICES [=y])
      -> USB Network Adapters (USB_NET_DRIVERS [=y])
Defined at drivers/net/usb/kconfig: 128
Depends on: NETDEVICES [=y] && USB_NET_DRIVERS [=y]
Selects: MII [=y]
```

### 3.1.2 Enabling CDC Ethernet support

Enable kernel option `CONFIG_USB_NET_CDCETHER` at:

```
Symbol: USB_NET_CDCETHER [=y]
Type: tristate
Prompt: CDC Ethernet support (smart devices such as cable modems)
Location:
  -> Device Drivers
    -> Network device support (NETDEVICES [=y])
      -> USB Network Adapters (USB_NET_DRIVERS [=y])
        -> Multi-purpose USB Networking Framework (USB_USBNET [=y])
Defined at drivers/net/usb/kconfig: 210
Depends on: NETDEVICES [=y] && USB_NET_DRIVERS [=y] && USB_USBNET [=y]
```

### 3.1.3 Enabling USB driver for GSM and CDMA modems

Enable kernel option `CONFIG_USB_SERIAL_OPTION` at:

```
Symbol: USB_SERIAL_OPTION [=y]
Type : tristate
Prompt: USB driver for GSM and CDMA modems
Location:
  -> Device Drivers
    -> USB support (USB_SUPPORT [=y])
      -> USB Serial Converter support (USB_SERIAL [=y])
Defined at drivers/usb/serial/kconfig: 558
Depends on: USB_SUPPORT [=y] && USB [=y] && USB_SERIAL [=y]
Selects: USB_SERIAL_WWAN [=y]
```

All the steps above have been done in this `config` file. The file is provided separately.

The `config` file aims to Linux kernel v5.4.47. If you are using the same kernel version, use it directly. Otherwise, perform the steps on your own.

```
$ make ./defconfig
```

## 3.2 Installing 5G module software

5G module can be used on either USB or PCIe mode, or on both through the M.2 interface.

The software needed is different for either mode.

- Quectel 5G module supports both mode in one module.
- Fibocom 5G module supports either mode.
- Gosuncn 5G module only supports USB mode.

### 3.2.1 Installing software for USB mode

#### 3.2.2 Enabling Gosuncn 5G module in Kernel

If Gosuncn 5G module is used, apply the patch provided separately.

```
$ git apply ./0001-gosuncn-5G-modem.patch
```

#### 3.2.3 Enabling Fibocom 5G module in Kernel

If Fibocom 5G module is used, apply the patch provided separately.

```
$ git apply ./0002-add-Fibocom-5G-module-support.patch
```

#### 3.2.4 Enabling Quectel 5G module in Kernel

If Fibocom 5G module is used, apply the patch provided separately.

```
// for qualcomm platform
$ git apply ./0001-Add-quectel-5G-module-support.patch

// for unisoc platform
$ git apply ./0001-Add-quectel-5G-module-support.patch
$ git apply ./ 0002-Add-quectel-5G-module-support-for-unisoc-platform.patch
```

#### 3.2.5 Installing software for PCIe mode

PCIe mode kernel software is compiled as kernel module.

#### 3.2.6 Compiling Fibocom 5G module driver

- Get the 5G module for PCIe mode software from the vendor, including kernel driver and dial software.
- Compile the kernel driver as the module:

```
$ tar xvf fibocom_pcie_mhi_driver_linux_v1.0.3.tar.gz
$ cd fibo_mhi
$ make KDIR=kernel/source/dir CROSS_COMPILE=cross/compile/toolchain/dir
```

After the compiling completes, the `fibo_mhi.ko` kernel module is created.

- Get the dial program from the vendor.

- Compile the dial program (This step can be done on embedded board, such as `ok1046a`):

```
$ unzip fibocom_dial_Linux_Tool_V2.0.7.zip
$ cd fibocom_dial_Linux_Tool_V2.0.7/src
$ make
```

- After the compiling completes, the `fibocom-dial` is generated.

### 3.2.7 Compiling Quectel 5G module driver

- Get the 5G module for PCIe mode software from the vendor, including kernel driver and dial software.
- Compile the kernel driver as the module:

```
// for qualcomm platform
$ unzip Quectel_Linux_PCIE_MHI_Driver_V1.3.0.17.zip
$ cd pcie_mhi
$ make KDIR=kernel/source/dir CROSS_COMPILE=cross/compile/toolchain/dir

// for unisoc platform (currently PCIe mode is not available on v5.4 or above kernel)
$ unzip Quectel_Linux&Android_SPRD_PCIE_Driver_V1.1.1.zip
$ cd pcie_mhi
$ make KDIR=kernel/source/dir CROSS_COMPILE=cross/compile/toolchain/dir
```

After the compiling completes, the `pcie_mhi.ko` kernel module is created.

- Get the dial program from the vendor.
- Compile the dial program (This step can be done on embedded board, such as, `ok1046a`):

```
$ unzip Quectel_QConnectManager_Linux_V1.6.0.26.zip
$ cd quectel-CM
$ make
```

- After compiling completes, **quectel-CM** was generated.

### 3.3 Installing test tools

The [speedtest.net](https://www.speedtest.net) website is used for latency and download/upload speed test. The website also provides a Command Line Interface tool. To install the tool, enter the commands as below:

```
$ curl -s https://install.speedtest.net/app/cli/install.deb.sh | sudo bash
$ sudo apt-get install speedtest
```

### 3.4 Setting date

Check whether the system date is correct. If not, set the date correctly with the commands as below:

```
$ date -s "20210713 11:00"
```

### 3.5 Installing DHCP client tool

The DHCP client tool is used to assign IP address to specific interface automatically. To install the DHCP client tool, enter the commands as below:

```
$ sudo apt update
$ sudo apt install -y udhcpd
```

## 4 Test process

The 5G module tests for USB mode and PCIe mode are different.

### 4.1 Preparing 5G module for USB mode

#### 4.1.1 Preparing 5G module

Before setting the 5G module, enter the AT mode to check the 5G module status, such as, SIM card or signal quality. To enter the AT mode, use the command as below:

```
root@localhost:~# minicom -D /dev/ttyUSB1
```

Now, the AT command can be entered.

#### 4.1.2 Making 5G module ready for test

#### 4.1.3 Checking Gosuncn 5G module

To check and set Gosuncn 5G module, enter the commands as below:

```
AT+ZPAS?  
+ZPAS: "LTE", "CS_PS", "TDD"  
  
OK  
AT+CPIN?  
+CPIN: READY  
  
OK  
AT+ZSWITCH?  
+ZSWITCH: e  
OK  
AT+ZECMCALL=1  
+ZECMCALL: CONNECT  
  
OK  
  
+ZECMCALL: IPV4 Connected  
AT+ZECMCALL?  
+ZECMCALL: IPV4,10.193.58.236,10.193.58.237,221.179.155.193,221.179.155.209  
  
OK  
at+zcdrun=f  
Exit download mode result(0:FAIL 1:SUCCESS):1  
OK
```

#### 4.1.4 Checking Fibocom 5G module

To check and set Fibocom 5G module, enter the commands as below:

```
AT+CPIN?  
+CPIN: READY  
  
OK  
AT+COPS?  
+COPS:0,0,"china mobile", 11  
AT+CGDCONT?
```



```
+CGDCONT: 1,"IPV4V6","CTLTE","0.0.0.0",0,0,0,0
OK
```

### 4.1.5 Checking Quectel 5G module

To check and set Quectel 5G module, enter the commands as below:

```
# minicom -D /dev/ttyUSB2
at+qcfg="data_interface"
+QCFG: "data_interface",0,0

AT+CPIN?
+CPIN: READY

OK
AT+COPS?
+COPS:0,0,"china mobile", 11
AT+CGDCONT?
+CGDCONT: 1,"IPV4V6","CTLTE","0.0.0.0",0,0,0,0
OK
```

## 4.2 Preparing 5G module for PCIe mode

Fibocom 5G module only supports one mode for each module. To test the PCIe module, use the 5G module for PCIe mode.

Quectel 5G module supports both USB3.0 and PCIe mode on one module. Use the AT command to switch modes.

### 4.2.1 Checking Fibocom 5G module

To install PCIe driver and enable PCIe device, enter the commands as below:

```
# insmod fibo_mhi.ko
[ 38.825220] fibo_mhi: loading out-of-tree module taints kernel.
[ 38.835509] mhi_init Fibocom_Linux_PCIE_MHI_Driver_V1.0.3

# echo 1 > /sys/bus/pci/rescan
[ 51.020842] pci 0000:01:00.0: [17cb:0306] type 00 class 0xff0000
[ 51.026998] pci 0000:01:00.0: reg 0x10: [mem 0x00000000-0x00000fff 64bit]
[ 51.033836] pci 0000:01:00.0: reg 0x18: [mem 0x00000000-0x00000fff 64bit]
[ 51.040998] pci 0000:01:00.0: PME# supported from D0 D3hot D3cold
[ 51.047279] pci 0000:01:00.0: 4.000 Gb/s available PCIe bandwidth, limited by 5 GT/s x1 link at 0000:00:00.0 (capable of 31.506 Gb/s with 16 GT/s x2 link)
[ 51.061153] OF: /soc/pcie@3400000: no msi-map translation for id 0x100 on (null)
[ 51.079827] pcieport 0000:00:00.0: BAR 14: assigned [mem 0x4040100000-0x40401fffff]
[ 51.087518] pci 0000:01:00.0: BAR 0: assigned [mem 0x4040100000-0x4040100fff 64bit]
[ 51.095220] pci 0000:01:00.0: BAR 2: assigned [mem 0x4040101000-0x4040101fff 64bit]
[ 51.103000] mhi_pci_probe pci_dev->name = 0000:01:00.0, domain=0, bus=1, slot=0, vendor=17CB, device=0306
[ 51.112588] [I][mhi_alloc_controller] 1112 size = 760
[ 51.117664] mhi 0000:01:00.0: BAR 0: assigned [mem 0x4040100000-0x4040100fff 64bit]
[ 51.125374] mhi 0000:01:00.0: enabling device (0000 -> 0002)
[ 51.303020] [I][mhi_netdev_enable_iface] Prepare the channels for transfer
[ 51.327971] [I][mhi_netdev_enable_iface] Exited.
```

To check the status, use the AT command as below.

```
# minicom -D /dev/mhi_DUN

AT+CPIN?
```

```

+CPIN: READY

OK
AT+COFS?
+COFS:0,0,"china mobile", 11

at+cgdcont?
+CGDCONT: 1,"IP","ctlte","10.40.102.180",0,0,0,0,,,,,,,,,"",,0
+CGDCONT:
2,"IPV4V6","ims","0.0.0.0,36.9.129.0.39.160.23.94.22.158.29.177.106.225.197.202",0,0,0,0,,,,,,,,,"",,
,0
+CGDCONT: 4,"IPV4V6","CMWAP","0.0.0.0,0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,0
+CGDCONT: 5,"IPV4V6","SOS","0.0.0.0,0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,1,,,,,,,,,"",,0

OK
at+gtusbmode?
+GTUSBMODE: 33

OK
at+cgdcont?
+CGDCONT: 1,"IP","ctnet","10.8.77.155",0,0,0,0,,,,,,,,,"",,0
+CGDCONT:
2,"IPV4V6","ims","0.0.0.0,36.9.129.0.38.137.30.122.22.158.40.132.157.186.225.136",0,0,0,0,,,,,,,,,"",,
,,0
+CGDCONT: 4,"IPV4V6","CMWAP","0.0.0.0,0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,0
+CGDCONT: 5,"IPV4V6","SOS","0.0.0.0,0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,1,,,,,,,,,"",,0

OK
at+gtrndis?
+GTRNDIS: 1,1,"10.8.77.155","221.179.155.193","221.179.155.209" // this is nameserver

OK

```

To exit the AT mode and start to dial, enter the commands as below:

```

# ifconfig pcie_mhi0
pcie_mhi0: flags=4240<POINTOPOINT,NOARP,MULTICAST> mtu 1500
    ether 00:00:00:00:00:00 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

# ./fibocom_dial_Linux_Tool_V2.0.7/src/fibocom-dial -s ctlte -d /dev/mhi_QMI0 &
[1] 1756
[08-24_14:29:29:160] ioctl(0x89f3, qmap_settings) failed: Operation not supported, rc=-1
[08-24_14:29:29:160] qmap_mode = 1, muxid = 0x81, qmap_netcard = pcie_mhi0
[08-24_14:29:29:160] ioctl(0x89f3, qmap_settings) failed: Operation not supported, rc=-1
[08-24_14:29:29:160] qmap_mode = 1, muxid = 0x81, qmap_netcard = pcie_mhi0
[08-24_14:29:29:165] cdc_wdm_fd = 9
[08-24_14:29:29:173] QmiWwanGetClientID: QMIType = 1 clientid 15
[08-24_14:29:29:173] Get clientWDS = 15
[08-24_14:29:29:177] QmiWwanGetClientID: QMIType = 2 clientid 1
[08-24_14:29:29:177] Get clientDMS = 1
[08-24_14:29:29:180] QmiWwanGetClientID: QMIType = 3 clientid 2
[08-24_14:29:29:180] Get clientNAS = 2
[08-24_14:29:29:181] QmiWwanGetClientID: QMIType = 11 clientid 2
[08-24_14:29:29:181] Get clientUIM = 2
[08-24_14:29:29:183] QmiWwanGetClientID: QMIType = 26 clientid 1
[08-24_14:29:29:183] Get clientWDA = 1

```

```

[08-24_14:29:29:185] requestBaseBandVersion 89602.1000.00.04.07.24
[08-24_14:29:29:186] qmap_settings.rx_urb_size = 32768
[08-24_14:29:29:190] dev: /dev/mhi_DUN
sendbuffer:AT+GTDUALSIMEN?

root@localhost:~# [08-24_14:29:32:190] read 25
+GTDUALSIMEN: 0

OK

[08-24_14:29:32:190] read 0/2/3

[08-24_14:29:32:190] sim_select = 0

[08-24_14:29:32:194] curr_ints_sim1
[08-24_14:29:32:194] curr_ints->CardState is 1
[08-24_14:29:32:194] curr_ints->NumApp is 1
[08-24_14:29:32:194] AppType = 2

[08-24_14:29:32:194] requestGetSIMStatus SIMStatus: SIM_READY
[08-24_14:29:32:197[ 1114.052463] [I][mhi_netdev_open] Opened net dev interface
] requestGetICCID DeviceICCID: 898600A00121F0036282
[08-24_14:29:32:199] requestGetIMSI DeviceIMSI: 460077010050709
[08-24_14:29:32:199] requestSetProfile[1] ctlte///0
[08-24_14:29:32:212] requestGetProfile[1] ctlte///0
[08-24_14:29:32:214] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: NR5G
[08-24_14:29:32:217] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[08-24_14:29:32:234] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: NR5G
[08-24_14:29:32:678] requestSetupDataCall WdsConnectionIPv4Handle: 0x531151b0
#
# ifconfig pcie_mhi0
pcie_mhi0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.32.193.100 netmask 255.255.255.248 destination 10.32.193.100
    inet6 fe80::b88b:51ff:feba:69ee prefixlen 64 scopeid 0x20<link>
    ether ba:8b:51:ba:69:ee txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 176 (176.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 1736 (1.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

To add the name server, enter the commands as below:

```

# echo "nameserver 221.179.155.193" > /etc/resolv.conf
# echo "nameserver 221.179.155.209" >> /etc/resolv.conf

# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=108 time=342 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=100 ms

```

## 4.2.2 Checking Quectel 5G module

To install PCIe driver and enable PCIe device, enter the commands as below:

```

# insmod pcie_mhi.ko
[ 63.801999] pcie_mhi: loading out-of-tree module taints kernel.
[ 63.811520] mhi_init Quectel_Linux_PCIE_MHI_Driver_V1.3.0.17

# echo 1 > /sys/bus/pci/rescan

```

```
[ 72.632271] pci 0000:01:00.0: [17cb:0306] type 00 class 0xff0000
[ 72.638472] pci 0000:01:00.0: reg 0x10: [mem 0x00000000-0x00000fff 64bit]
[ 72.645299] pci 0000:01:00.0: reg 0x18: [mem 0x00000000-0x00000fff 64bit]
[ 72.652453] pci 0000:01:00.0: PME# supported from D0 D3hot D3cold
[ 72.658732] pci 0000:01:00.0: 4.000 Gb/s available PCIe bandwidth, limited by 5 GT/s x1 link at
0000:00:00.0 (capable of 31.506 Gb/s with 16 GT/s x2 link)
[ 72.672597] OF: /soc/pcie@3400000: no msi-map translation for id 0x100 on (null)
[ 72.691295] pcieport 0000:00:00.0: BAR 14: assigned [mem 0x4040100000-0x40401fffff 64bit]
[ 72.698965] pci 0000:01:00.0: BAR 0: assigned [mem 0x4040100000-0x4040100fff 64bit]
[ 72.706660] pci 0000:01:00.0: BAR 2: assigned [mem 0x4040101000-0x4040101fff 64bit]
[ 72.714451] mhi_pci_probe pci_dev->name = 0000:01:00.0, domain=0, bus=1, slot=0,
vendor=17CB, device=0306
[ 72.724177] mhi_q 0000:01:00.0: BAR 0: assigned [mem 0x4040100000-0x4040100fff 64bit]
[ 72.732079] mhi_q 0000:01:00.0: enabling device (0000 -> 0002)
root@localhost:~# [ 75.891198] [I][mhi_netdev_enable_iface] Prepare the channels for transfer
[ 75.916322] [I][mhi_netdev_enable_iface] Exited.
[ 75.922674] rmnet_vnd_register_device(rmnet_mhi0.1)=0#
```

To check whether the module is in the PCIe mode, enter the commands as below:

```
// only for qualcomm platform
# minicom -D /dev/ttyUSB2

at+qcfg="data_interface"
+QCFG: "data_interface",1,1

OK

at+cpin?
+CPIN: READY

OK

at+cops?
+COFS: 0,0,"CHINA MOBILE",11

OK

at+cgdcont?
+CGDCONT: 1,"IPV4V6","CTLTE","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,0
+CGDCONT: 2,"IPV4V6","ims","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,0
+CGDCONT: 3,"IPV4V6","CMNET","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,0
+CGDCONT: 4,"IPV4V6","CMWAP","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,0
+CGDCONT: 5,"IPV4V6","SOS","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,1,,,,,,,,,"",,0

OK

# ifconfig pcie_mhi0
pcie_mhi0: flags=4240<POINTOPOINT,NOARP,MULTICAST> mtu 1500
    ether 00:00:00:00:00:00 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

To start to dial, enter the commands as below:

```
# ./quectel-CM/quectel-CM &
[1] 1052
[08-24_11:34:33:025] Quectel_QConnectManager_Linux_V1.6.0.26
[08-24_11:34:33:026] Find /sys/bus/usb/devices/2-1 idVendor=0x2c7c idProduct=0x800,
bus=0x002, dev=0x002
[08-24_11:34:33:026] network interface '' or qmidev '' is not exist
```

```

[08-24_11:34:33:026] netcard driver = pcie_mhi, driver version = V1.3.0.17
[08-24_11:34:33:026] qmap_mode = 1, qmap_version = 9, qmap_size = 16384, muxid = 0x81, qmap_netcard
= rmnet_mhi0.1
[08-24_11:34:33:027] Modem works in QMI mode
[08-24_11:34:33:050] cdc_wdm_fd = 7
root@localhost:~# [08-24_11:34:34:051] QmiWwanInit message timeout
[08-24_11:34:35:066] Get clientWDS = 14
[08-24_11:34:35:071] Get clientDMS = 1
[08-24_11:34:35:076] Get clientNAS = 4
[08-24_11:34:35:079] Get clientUIM = 1
[08-24_11:34:35:082] Get clientWDA = 1
[08-24_11:34:35:086] requestBaseBandVersion RM500QCNAAR12A06M4G
[08-24_11:34:35:088] qmap_settings.rx_urb_size = 16384
[08-24_11:34:35:088] qmap_settings.ul_data_aggregation_max_datagrams = 11
[08-24_11:34:35:088] qmap_settings.ul_data_aggregation_max_size = 8192
[08-24_11:34:35:088] qmap_settings.dl_minimum_padding = 0
[08-24_11:34:35:102] requestGetSIMStatus SIMStatus: SIM_READY
[08-24_11:34:35:105] requestGetProfile[1] CTLTE///0
[08-24_11:34:35:109] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: 5G_SA
[08-24_11:34:35:112] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[08-24_11:34:35:112] ifconfig rmnet_mhi0 down
[08-24_11:34:35:115] ifconfig rmnet_mhi0.1 0.0.0.0
SIOCSIFFLAGS: Network is down
[08-24_11:34:35:119] ifconfig rmnet_mhi0.1 down
[08-24_11:34:35:657] requestSetupDataCall WdsConnectionIPv4Handle: 0x7703b7c0
[ 61.329878] net rmnet_mhi0: link_state 0x0 -> 0x1
[08-24_11:34:35:684] ifconfig rmnet_mhi0 up
[ 61.338696] [I][mhi_netdev_open] Opened net dev interface
[08-24_11:34:35:694] ifconfig rmnet_mhi0.1 up
[08-24_11:34:35:700] busybox udhcpc -f -n -q -t 5 -i rmnet_mhi0.1
udhcpc: started, v1.27.2
udhcpc: sending discover
udhcpc: sending select for 10.40.2.157
udhcpc: lease of 10.40.2.157 obtained, lease time 7200

```

To check whether the dialing is successful, enter the commands as below:

```

# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 50 bytes 2966 (2.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 50 bytes 2966 (2.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

rmnet_mhi0: flags=193<UP,RUNNING,NOARP> mtu 1500
    inet6 fe80::46d0:8e1e:f6cf:f1ca prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
    RX packets 2 bytes 632 (632.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 800 (800.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

rmnet_mhi0.1: flags=193<UP,RUNNING,NOARP> mtu 1500
    inet 10.56.68.137 netmask 255.255.255.252
    inet6 fe80::50:f4ff:fe00:0 prefixlen 64 scopeid 0x20<link>
    ether 02:50:f4:00:00:00 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 612 (612.0 B)

```

```

RX errors 0  dropped 0  overruns 0  frame 0
TX packets 4  bytes 824 (824.0 B)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

# cat /etc/resolv.conf

nameserver 221.179.155.193
nameserver 221.179.155.209

# ping www.baidu.com
PING www.a.shifen.com (39.156.66.18) 56(84) bytes of data.
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=1 ttl=52 time=43.4 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=2 ttl=52 time=33.9 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=3 ttl=52 time=83.9 ms

```

## 4.3 Testing 5G module

### 4.3.1 Checking tool usage

Now the 5G module is ready for test. The `speedtest` tool is used for test. Before starting the test, to check the usage of this tool, enter the command as below:

```
root@localhost:~# ./speedtest --help
```

### 4.3.2 Listing available test servers

List all the available test server so that the best server can be selected as the test server. Usually, the nearest server is the best server.

```
root@localhost:~# ./speedtest -L
```

### 4.3.3 Starting the test

To use the server in 石家庄 (id = 41912) as the test server, enter the commands as below:

```

root@localhost:~# ./speedtest -s 41912

Speedtest by Ookla

Server: China Mobile Hebei Co., Ltd - 石家庄 (id = 41912)
ISP: China Mobile Guangdong
Latency: 19.33 ms (6.33 ms jitter)
Download: 275.22 Mbps (data used: 324.2 MB)
Upload: 40.99 Mbps (data used: 53.2 MB)
Packet Loss: Not available.
Result URL: https://www.speedtest.net/result/c/fab26147-7c5e-4a8b-939e-9517d25f0764

```

## 5 Revision history

Rev.	Date	Description
0	15 November 2021	Initial release

## How To Reach Us

### Home Page:

[nxp.com](http://nxp.com)

### Web Support:

[nxp.com/support](http://nxp.com/support)

**Limited warranty and liability**— Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

**Right to make changes** - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Security**— Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetic, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2021.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 15 November 2021

Document identifier: AN13455

