MeiG 5G&LTE Module
Linux Driver Application Note

Controlled Version Number：V1.1

Release Date：2021/5/20

Add：32/F,Building B,Shenzhen International Innovation Center,
No.1006 Shennan Avenue,Futian District,Shenzhen,China
Tel：+86-755-83218588 Fax：+86-755-83219788
E-mail：ir@meigsmart.com Web：www.meigsmart.com

Important Notice

Copyright Notice

Statement of No Guarantee

MeiG Smart Technology Co., Ltd makes no representations or guarantees, either express or implied, for any contents in this document, and assumes no responsibility for the merchantability and fitness for a particular purpose or any indirect, extraordinary or consequential losses.

Confidentiality Claim

The information contained in this document (including any annexes) is confidential. The recipient understands that the document obtained by him is confidential and shall not be used for any purpose other than the stated purpose, and he shall not disclose this document to any third party.

Disclaimer

The company assumes no responsibility for property damage or personal injury caused by customers' improper operation. Customers are requested to develop corresponding products according to the technical specifications and reference designs in the manual. Before the disclaimer, the company has the right to change the contents of this manual according to the needs of technological development, and the version is subject to change without further notice.

Revision History

Revision	Date	Description
V1.0	2020-08-13	Created, Initial Version
V1.1	2021-05-20	Added the instruction of PCIe dialing Added the instruction of GobiNet+ AT IPv6 dialing

Content

1 Introduction

1.1 Purpose of this Document

This document mainly introduces the adaptation ofMeiG modules based on Linux systems. With this document, user can easily design with SLM750 module into the device, and provide data, voice, SMS and other telecommunication services with the device.

1.2 Safety Instructions

By following the safety principles, user can ensure personal safety and help protect the product and work environment from potential damages:

	Driving safety first! When you are driving, do not use a handheld mobile terminal unless it has a hands-free function. Please stop and call again!
	Please turn off the mobile terminal device before boarding. The wireless function of the mobile terminal must not be turned on on the airplane to prevent interference to the communication system of the airplane. Ignoring this prompt may lead to flight safety and even violate the law.
	In hospitals or health care facilities, pay attention to whether there are restrictions on the use of mobile terminal equipment. RF interference can cause medical equipment to malfunction, so mobile terminal equipment may need to be turned off.
	The mobile terminal device cannot be effectively connected under any circumstances, and there is no call charge on the mobile device or the SIM is invalid. When you encounter the above situations in an emergency, please remember to make an emergency call and ensure that your device is turned on and in an area with sufficient signal strength.
	Your mobile terminal device receives and transmits radio frequency signals when it is turned on. Radio frequency interference occurs when it is close to TVs, radio computers or other electronic equipment.
	Keep mobile devices away from flammable gases. When you are close to a gas station, oil depot, chemical plant or explosion site, please turn off the mobile terminal device. There are potential safety hazards in operating electronic equipment in any potentially explosive place.

2 MeiG Module Information

2.1 Vendor ID and Product ID

Table 1 MeiG Module VID & PID

Module	Vendor ID	Product ID	SIM Hot plug
SLM750	0x05C6	0xF601	N
SLM790	0x2DEE	0x4D20	N
SLM868	0x05C6	0xF601	N
SRM815	0x2DEE	0x4D22	Y
SRM815(ECM)	0x2DEE	0x4D23	Y

2.2 Module Virtual Port

SRM750/SLM868/SRM815 modules can generate 6 virtual ports as shown in Table 2.

Table 2 SLM750/SLM868/SRM815 Virtual Ports

Port Number	Functions
0	Diag port—Diagnostic port, capturing debug log
1	Modem port—PPP dialing
2	AT Port—AT Command port, communication with MCU
3	NMEA Port—GNSS Port
4	adb Port
5	RMNET Port—used for send qmi

SLM790module **NCM version** has 5 virtual ports listed in Table 3.

Table 3 SLM790 Module NCM version Virtual Ports

Port Number	Functions
0	Network Port，used for NCM dialing
1	AT Port, used for AT command communication
2	3G Application Port，for debug use
3	ApplicationPort，for capturing system log
4	Modem Port, for PPP dialing

SLM790 module **ECM version** also has 5 ports, shown in Table 4.

Table 4 SLM790 Module ECM Version Ports

Port Number	Functions
0	Application Port，capturing system log
1	AT Port, used for AT command communication
2	3G Application Port，for debug use
3	Modem Port, for PPP dialing
4	ECM Network Port

Note:
The above port sequence is only consistent with the generated /dev/ttyUSBX sequence, not necessarily the same as the actual port number.

2.3 MeiG Module Supported Dialing Methods

Table 5 Dialing Method

Module	PPP	ECM	NCM	Gobinet	Qmi_wwan	MBIM	RNDIS	GobiNet (Multiplex)
SLM750	Y	Y	N	Y	Y	N	Y	Y
SLM790	Y	Y	Y	N	N	N	Y	N
SLM868	Y	Y	N	Y	Y	Y	Y	Y
SRM815	Y	Y	N	Y	Y	Y	Y	Y

3 List of Linux Adpated Files

Table 6 List of Linux Adapted Files

File Name	Description
ppp_script_for_linux.tar.gz	PPP dialing script
udhcpc_script.tar.gz	Udhcpc script, can be used on platforms that do not have scripts by default

4 USB to UART Dirver Adaptation

The module USB port is multiplexed, so user needs to use the option driver to separate multiple serial ports for use.

4.1 Kernel Configuration

Add the following items in the kernel configuration file

```
CONFIG_USB_SERIAL_GENERIC=y
CONFIG_USB_SERIAL_OPTION=y
CONFIG_USB_SERIAL_QT2=y
```

Note:
The kernel can be compiled on the PC after make menuconfig, and then add the above configuration to the .config file.

4.2 Modify Option Driver

Add module information in the option driver, the kernel modification method of version 4.x and above:

```
--- a/drivers/usb/serial/option.c
+++ b/drivers/usb/serial/option.c
@@ -85,6 +85,13 @@ static intoption_probe(structusb_serial *serial,
 #define HUAWEI_PRODUCT_K3765                0x1465
 #define HUAWEI_PRODUCT_K4605                0x14C6
 #define HUAWEI_PRODUCT_E173S6               0x1C07
+/*[MEIG-zhaopf-2019-11-04]add for meig modem supported {*/
+#define MEIG_VENDOR_ID                      0x2DEE
+#define MEIG_PRODUCT_SRM815                  0x4D22
+#define MEIG_PRODUCT_SRM815_ECM              0x4D23
+#define MEIG_PRODUCT_SLM790                  0x4D20
+#define MEIG_QCM_VENDOR_ID                   0x05C6
+#define MEIG_QCM_PRODUCT_SLM750_SRM815_SLM868   0xF601
+/*[MEIG-zhaopf-2019-11-04]add for meig modem supported }*/
+
 #define QUANTA_VENDOR_ID                    0x0408
 #define QUANTA_PRODUCT_Q101                  0xEA02
@@ -564,6 +571,12 @@ static intoption_probe(structusb_serial *serial,

 static conststructusb_device_idoption_ids[] = {
+/*[MEIG-zhaopf-2019-11-04]add for meig modem supported {*/
+{ USB_DEVICE(MEIG_VENDOR_ID,  MEIG_PRODUCT_SRM815),
+.driver_info = RSVD(4) | RSVD(5) },
+{ USB_DEVICE(MEIG_VENDOR_ID,  MEIG_PRODUCT_SRM815_ECM),
+.driver_info = RSVD(4) | RSVD(5) },
+{ USB_DEVICE(MEIG_QCM_VENDOR_ID,  MEIG_QCM_PRODUCT_SLM750_SRM815_SLM868),
+.driver_info = RSVD(4) | RSVD(5) },
+{ USB_DEVICE(MEIG_VENDOR_ID,  MEIG_PRODUCT_SLM790),
+.driver_info = RSVD(0) | RSVD(5) | RSVD(6) | RSVD(7) },
+/*[MEIG-zhaopf-2019-11-04]add for meig modem supported }*/
       { USB_DEVICE(OPTION_VENDOR_ID, OPTION_PRODUCT_COLT) },
```

Kernel modification methods for versions below 4.x：

```
--- a/kernel/drivers/usb/serial/option.c
+++ b/kernel/drivers/usb/serial/option.c
@@ -86,12 +86,11 @@ static void option_instat_callback(structurb *urb);
 #define HUAWEI_PRODUCT_K4605               0x14C6
 #define HUAWEI_PRODUCT_E173S6              0x1C07
 /*[MEIG-zhaopf-2019-11-04]add for meig modem supported {*/
+#define MEIG_QCM_VENDOR_ID                 0x05C6
+#define MEIG_VENDOR_ID                     0x2DEE
+#define MEIG_PRODUCT_SLM790                 0x4D20
+#define MEIG_PRODUCT_SRM815                 0x4D22
+#define MEIG_PRODUCT_SRM815_ECM0x4D23
+#define MEIG_QCM_PRODUCT_SRM815_SLM750_SLM868   0xF601
 /*[MEIG-zhaopf-2019-11-04]add for meig modem supported }*/


@@ -701,13 +700,23 @@ static conststructoption_blacklist_info yuga_clm920_nc5_blacklist
= {
        .reserved = BIT(1) | BIT(4),
 };

+/*[MEIG-zhaopf-2019-11-04]add for meig modem supported {*/
+static conststructoption_blacklist_info meig_slm790_blacklist = {
+       .reserved = BIT(0) | BIT(5)| BIT(6)| BIT(7),
+};
+static conststructoption_blacklist_info meig_slm790_ecm_blacklist = {
+       .reserved = BIT(4) | BIT(5)| BIT(6)| BIT(7),
+};

+static conststructoption_blacklist_info meig_srm815_slm750_slm868_blacklist = {
+       .reserved = BIT(4) | BIT(5),
+};
+
+/*[MEIG-zhaopf-2019-11-04]add for meig modem supported }*/
static conststructusb_device_idoption_ids[] = {
       /*[MEIG-zhaopf-2019-11-04]add for meig modem supported {*/
// If the SLM790 module is an ECM version, you need to change the following
meig_slm790_blacklist to meig_slm790_ecm_blacklist+
{ USB_DEVICE(MEIG_VENDOR_ID, MEIG_PRODUCT_SLM790),
+          .driver_info = (kernel_ulong_t)&meig_slm790_blacklist },
+       { USB_DEVICE(MEIG_QCM_VENDOR_ID, MEIG_QCM_PRODUCT_SRM815_SLM750_SLM868),
+          .driver_info = (kernel_ulong_t)&meig_srm815_slm750_slm868_blacklist },
+       { USB_DEVICE(MEIG_VENDOR_ID, MEIG_PRODUCT_SRM815),
+          .driver_info = (kernel_ulong_t)&meig_srm815_slm750_slm868_blacklist },
+       { USB_DEVICE(MEIG_VENDOR_ID, MEIG_PRODUCT_SRM815_ECM),
+          .driver_info = (kernel_ulong_t)&meig_srm815_slm750_slm868_blacklist },
       /*[MEIG-zhaopf-2019-11-04]add for meig modem supported }*/
```

4.3 Compile and Load the driver

User can compile “option.ko” and use the “insmod command” to load the driver.

After loading the driver successfully, for example, when the SRM815 module is used, the device will generate 4 virtual ports:

```
kvim:/ # ls -la /dev/ttyUSB*
crw-rw-r-- 1 radio radio 188,   0 2020-04-16 09:41 /dev/ttyUSB0
crw-rw-r-- 1 radio radio 188,   1 2020-04-16 09:41 /dev/ttyUSB1
crw-rw-r-- 1 radio radio 188,   2 2020-04-16 10:16 /dev/ttyUSB2
crw-rw-r-- 1 radio radio 188,   3 2020-04-16 09:41 /dev/ttyUSB3
```

Figure 1 Virtual Port Example

5 PPP Dialing

5.1 Add PPP driver in the Kernel

Add the PPP driver in the kernel configuration file, for example:

```
+++ b/osdrv/opensource/kernel/linux-3.18.y/arch/arm/configs/hi3520dv400_full_defconfig
@@ -2439,4 +2439,10 @@ CONFIG_USB_SERIAL_OPTION=y
 CONFIG_USB_NET_CDCETHER=y
 CONFIG_USB_USBNET=y
 CONFIG_USB_NET_MEIG_CDC_NCM=y
+CONFIG_PPP=y
+CONFIG_PPP_FILTER=y
+CONFIG_PPP_MULTILINK=y
+CONFIG_PPP_ASYNC=y
+CONFIG_PPP_SYNC_TTY=y
+CONFIG_PPP_DEFLATE=y
```

5.2 PPPd Script Preparation

```
#Unzip the script ppp_script_for_linux.tar.gz to the /etc/ppp directory
tar xzvfppp_script_for_linux.tar.gz-C /etc/
# Add executable permissions to the script
chmod 755 -R /etc/ppp
#Modify port
# Modify the port number in the file /etc/ppp/peers/gprs-dial to be consistent with the
actual
example: /dev/ttyUSB1
```

5.3 PPP Dialing

5.3.1 Set APN

APN must be set before dialing, user can use the AT command of "**AT+CGDCONT**" to set APN.

For example:

China Mobile—CMNET

China Telecom—CTNET OR CTLTE

China Unicom—3GNET

```
OK
at+cgdcont=1,"IPV4V6","cmnet"
OK
at+cgdcont?
+CGDCONT: 1,"IPV4V6","cmnet","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: 38,"IPV6","v2x_ip","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,,,0
+CGDCONT: 39,"IPV6","v2x_non_ip","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
at+cgdcont=1,"IPV4V6","ctnet"
OK
at+cgdcont?
+CGDCONT: 1,"IPV4V6","ctnet","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: 38,"IPV6","v2x_ip","0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0",0,0,0,0,,,,,,,,,"",,,,0
+CGDCONT: 39,"IPV6","v2x_non_ip","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
```

Figure 2 Set APN

5.3.2 Make Dialing

Before dialing, check whether the port in `/etc/ppp/peers/gprs_dial` is correct or not. If it is inconsistent with the actual one, user needs to modify it before using it.

```
pppd call gprs_dial
```

```
Script /usr/sbin/chat -s -v -f /etc/ppp/ppp-on-dialer finished (pid 12784), status = 0x0
Serial connection established.
using channel 3
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB1
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x54b3e1ca> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x8 <asyncmap 0x0> <auth chap MD5> <magic 0xf7971c9e> <pcomp> <accomp>]
No auth is possible
sent [LCP ConfRej id=0x8 <auth chap MD5>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x54b3e1ca> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x9 <asyncmap 0x0> <magic 0xf7971c9e> <pcomp> <accomp>]
sent [LCP ConfAck id=0x9 <asyncmap 0x0> <magic 0xf7971c9e> <pcomp> <accomp>]
sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [LCP DiscReq id=0xa magic=0xf7971c9e]
rcvd [LCP ProtRej id=0xb 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03 2f]
Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x2]
sent [IPCP ConfNak id=0x2 <addr 0.0.0.0>]
rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x3]
sent [IPCP ConfAck id=0x3]
rcvd [IPCP ConfNak id=0x2 <addr 10.154.69.135> <ms-dns1 211.137.130.2> <ms-dns2 211.137.130.18>]
sent [IPCP ConfReq id=0x3 <addr 10.154.69.135> <ms-dns1 211.137.130.2> <ms-dns2 211.137.130.18>]
rcvd [IPCP ConfAck id=0x3 <addr 10.154.69.135> <ms-dns1 211.137.130.2> <ms-dns2 211.137.130.18>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing existing default route via 192.168.147.2
local  IP address 10.154.69.135
remote IP address 10.64.64.64
primary   DNS address 211.137.130.2
secondary DNS address 211.137.130.18
Script /etc/ppp/ip-up started (pid 12797)
Script /etc/ppp/ip-up finished (pid 12797), status = 0x0
```

Figure 3 PPP Dialing

5.4 PPPd Verification

Chinese Customers can make a PING to a public IP address, for example:

PING 114.114.114.114

PINGwww.baidu.com

Overseas customers can make a PING to a public IP address, for example:

PING 8.8.8.8

PING www.google.com

6 ECM Dialing

6.1 Load ECM Driver

The ECM driver is generally loaded by default in the linux kernel.If not, it can be loaded as follows for linux PC.

```
modprobeusbnet
modprobecdc_ether
```

For embedded linux environment, user canconfig the usbnet and ecm switches in the kernel configuration

```
CONFIG_USB_USBNET=y
CONFIG_USB_NET_CDCETHER=y
```

6.2 ECM Dialing

Normally, the ECM version of the module is automatically dialing by default, similar to plug and play, and it usually generates a network port named `ethx`or `usbx`.

For example, when generating the network card `usb0`, user can check whether it has obtained the IP through `ifconfig usb0`. If user has obtained it, user can directly PING to verify.

For some embedded linux platforms that cannot automatically request DHCP, customer can use `udhcpc`, `dhclient`, `dhcpcd`and other tools to obtain and set IP information.

For example:

```
#Note that whether udhcpc can successfully set IP, gateway, dns and other information will
depend on the configuration script,
Default Parth: "/etc/udhcpc/default.script" user can also specifiy the script by -s paramter
udhcpc -i usb0 -s /etc/udhcpc/default.script
```

7 NCM Dialing

For modules that support NCM dialing，user needs to compile and load MeiG NCM driver. In the meantime, if user already load Option driver, user needs to follow Ch.4 to shield the network port of the module, otherwise it will cause the NCM driver to fail to find the port.

7.1 Compile and Load Driver

```
#Unzip driver
tar xzvf MeiG_NCM_V0.5.1.tar.gz

#Compile on PC
cd MeiG_NCM_V0.5.1
make

#Load Driver
insmodmeig_cdc_driver.ko

#Enable network card
ifconfig usb0 up
```

For embedded linux devices, user can add drivers as follows,copy the driver file `meig_cdc_driver.c` to the driver directory `drivers/net/usb/`,and modify `Kconfig` and `Makefile` as follows, compile and update the kernel after modification.

- drivers/net/usb/Kconfig

```
--- a/osdrv/opensource/kernel/linux-3.18.y/drivers/net/usb/Kconfig
+++ b/osdrv/opensource/kernel/linux-3.18.y/drivers/net/usb/Kconfig
@@ -262,6 +262,20 @@config USB_NET_HUAWEI_CDC_NCM
            To compile this driver as a module, choose M here: the module will be
calledhuawei_cdc_ncm.ko.

+config USB_NET_MEIG_CDC_NCM
+       tristate "Meig NCM embedded AT channel support"
+       depends on USB_USBNET
+       select USB_WDM
+       select USB_NET_CDC_NCM
+       help
+            This driver supports meige-style NCM devices, that use NCM as a
+            transport for other protocols, usually an embedded AT channel.
+            Good examples are:
+            * MEIG SLM790
+
+            To compile this driver as a module, choose M here: the module will be
+            called meig_cdc_driver.ko.
+
```

- drivers/net/usb/Makefile

```
--- a/osdrv/opensource/kernel/linux-3.18.y/drivers/net/usb/Makefile
+++ b/osdrv/opensource/kernel/linux-3.18.y/drivers/net/usb/Makefile
@@ -37,4 +37,4 @@obj-$(CONFIG_USB_NET_HUAWEI_CDC_NCM)  += huawei_cdc_ncm.o
obj-$(CONFIG_USB_VL600)              += lg-vl600.o
obj-$(CONFIG_USB_NET_QMI_WWAN) += qmi_wwan.o
obj-$(CONFIG_USB_NET_CDC_MBIM) += cdc_mbim.o
-
+obj-y  += meig_cdc_driver.o
```

After the driver is loaded successfully, when the module is used, a network card named usbX will appear, usually it is at usb0 port.

7.2 NCM Dialing

After the network card is successfully generated, user needs to use a serial port tool such as `minicom` to send thecommand `AT^NDISDUP` to make dialing, at this time, user needs to make sure that the USB to serial port driver has been adapted,(drivers/usb/serial/option.c), for example:

```
#Set APN, for example:
AT+CGDCONT=1,"IPV4V6","cmnet"
#Dialing
AT^NDISDUP=1,1
#Terminate dialing
AT^NDISDUP=1,0
```

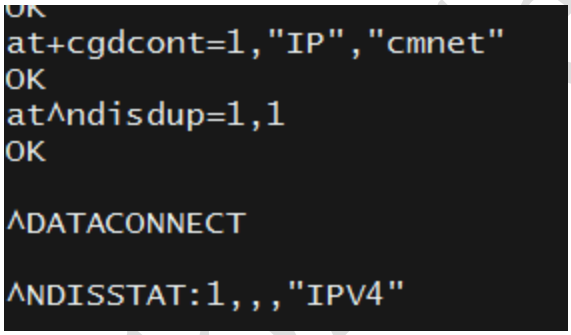


Figure 4 NDIS Dialing

After dialing, the dhcp client will automatically request IP information. If the current platform does not support it, user can manually use dhcp clients such as udhcpc, dhtool, dhcpcd, and dhclient to request.

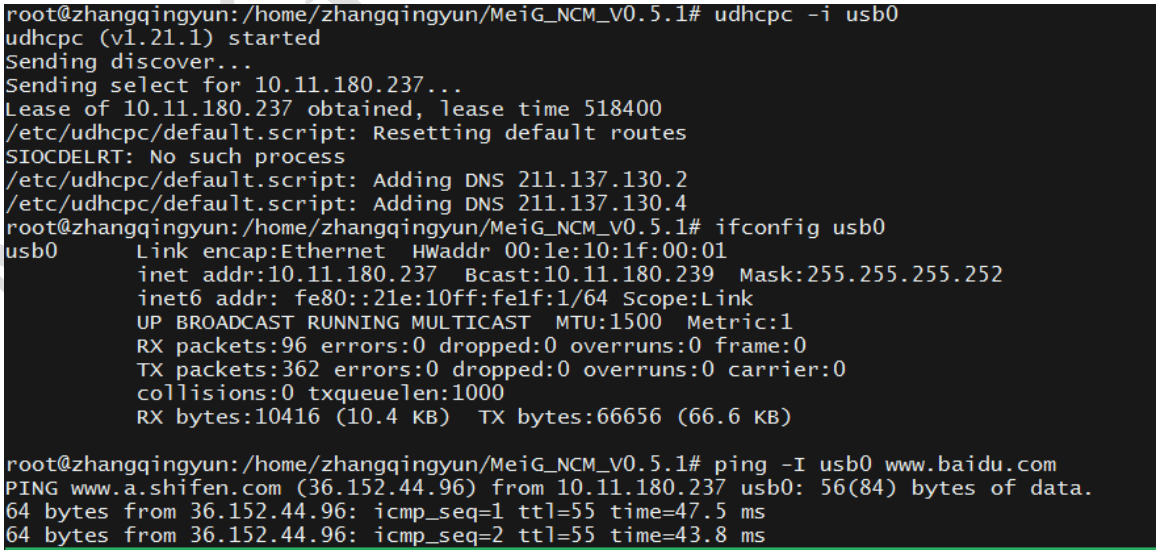


Figure 5 Network Connectivity Verification

8 Gobinet Single Dialing

8.1 Load Gobinet Driver

Unzip driver

```
tar xzvf Meig_GobiNet_Driver_V1.4.1.tar.gz
```

For the PC environment, execute the script in the driver directory to load all the drivers

```
source go_gobi.sh
```

For embedded linux environment, cross compilation is required

```
#Compile driver
#make -C [Kernel path] M=[Gobinet driver path] CROSS_COMPILE=[ Cross compiler tool prefix] modules
make -C /home/zhaopf/work/linux-4.14.148 M=/home/zhaopf/work/release/GobiNet modules CROSS_COMPILE=aarch64-linux-android-
#Generate driver: GobiNet.ko
#Load it on the corresponding platform
insmodGobiNet.ko

#Compile the dial tool
cd meig-cm
make CROSS_COMPILE=aarch64-linux-android-
# Generate the dial tool meig-cm and copy it to the machine
```

8.2 Gobinet Dialing Verification

8.2.1 Use CM to dial

User must compile the MeiG-CM tool before usage, then using the compiled MeiG-CM to dial

```
#Parameter Description:
#-sSet APN
#-6 supports ipv4v6 protocols
#-iSpecify the name of the network card, for the case where part of the network card name is modified
#For example, set China Mobile APN:
meig-cm -s cmnet
#For example, set China Telecom APN:
meig-cm -s ctnet
#For example, set China Unicom APN:
meig-cm -s 3gnet
```

8.2.2 Use AT command to dial

Using AT to dial depends on the DHCP client. For the ubuntusystem, user can use the following command to install, such as:

```
sudo apt-get install udhcpc
```

Single IPv4 dial:

```
#Enable network card
ifconfig<network card name> up

#Issue command through AT port
AT$QCRMCALL=1,1,1

#Request dhcp
udhcpc -i<network card name>
```

Single IPv6 dial:

```
#Enable network card
ifconfig<network card name> down

# Issue command through AT port
AT$QCRMCALL=1,1,2

#Enable network card, get IPV6 address automatically
ifconfig<network card name> up
```

IPv4v6 dial:

```
#Disable network card
ifconfig<network card name> down

# Issue command through AT port
AT$QCRMCALL=1,1,3

#Enable network card and request DHCP immediately
ffconfig<network card name> up
udhcpc -i<network card name>
```

Note:

Because of the NetworkManagerservice is availableon the ubuntusystem; there is no need to manually send DHCP requests. User can directly enable/disable the network card, and the system will send the DHCP request by itself. For example:

```
#Disable network card
Ifconfig<network card name> down

# Issue command through AT port
AT$QCRMCALL=1,1,3

#Enable network card
ifconfig<network card name> up
```

9 Gobinet Multi-Channel Dialing

9.1 Load Driver

Decompress driver

```
tar xzvf GobiNet_MultiRmNet_v2.0.1.tar.gz
```

For the PC environment, execute the script in the driver directory to load all the drivers and prepare the dial-up environment

```
source go_gobi.sh
```

For embedded linux environment, cross-compilation is required

```
#Compile driver
#make -C [kernel path] M=[Gobinet driver path] CROSS_COMPILE=[ Cross compiler tool prefix] modules
make  -C  /home/xxx/work/linux-4.14.148  M=/home/zhaopf/work/release/GobiNet  modules CROSS_COMPILE=aarch64-linux-android-
#Generate driver GobiNet.ko
#Load the driver on the corresponding platform
insmodGobiNet.ko

#In order to activate the dhcp client, user needs to initiate a dhcp request immediately after dialing so as to dial successfully, as a result,user can start the background process in advance, set multiple dial channels per request,maximum 4 channels, for example:

#1st channel
ifconfig bmwan0 up
udhcpc -f -t 0 -i bmwan0 -x hostname:test-C -O 121 > /dev/null &

#2nd channel
ifconfig bmwan1 up
udhcpc -f -t 0 -i bmwan1 -x hostname:test-C -O 121 > /dev/null &

#3rd channel
ifconfig bmwan2 up
udhcpc -f -t 0 -i bmwan2 -x hostname:test-C -O 121 > /dev/null &

#4th channel
ifconfig bmwan3 up
udhcpc -f -t 0 -i bmwan3 -x hostname:test-C -O 121 > /dev/null &
```

9.2 Verify Gobinet Multi-Channel Dialing

Multi-channel dialing needs to use serial port tool(such as `minicom`) to set each APN first through the command of "AT+CGDCONT", and then use "AT$QCRMCAL" to dial. Please refer to the appendix for the detailed usage of the two commands.

For example, use China Mobile SIM card to set 4 channles dialing:

Set APN:

```
at+cgdcont=1,"IP","cmnet"
OK
at+cgdcont=3,"IP","APN2"
OK
at+cgdcont=4,"IP","APN3"
OK
at+cgdcont=5,"IP","APN4"
OK
at+cgdcont?
+CGDCONT: 1,"IP","cmnet","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
+CGDCONT: 3,"IP","APN2","0.0.0.0",0,0,0,0
+CGDCONT: 4,"IP","APN3","0.0.0.0",0,0,0,0
+CGDCONT: 5,"IP","APN4","0.0.0.0",0,0,0,0
```

Figure 6 Set Multi-channel APN

Make multi-channle dialing

Take a note that the 2nd parameter <Instance> value should be the actual network card ID+1; the 5th parameter <profile number> is the APN ID.

For example, AT$QCRMCALL=X,2,X,X,1means network ID=1 and it's APN will be used, the network interface is bmwan3.

```
at$qcrmcall=1,1,1,2,1
$QCRMCALL: 1, V4

OK
at$qcrmcall=1,2,1,2,3
$QCRMCALL: 2, V4

OK
at$qcrmcall=1,3,1,2,4
$QCRMCALL: 3, V4

OK
at$qcrmcall=1,4,1,2,5
$QCRMCALL: 4, V4

OK
```

Figure 7 Multi-channel dialing

10 QMI_WWAN Dialing

Decompress driver

```
tar xzvf Meig_Qmi_wwan_Driver_V1.0.1.tar.gz
```

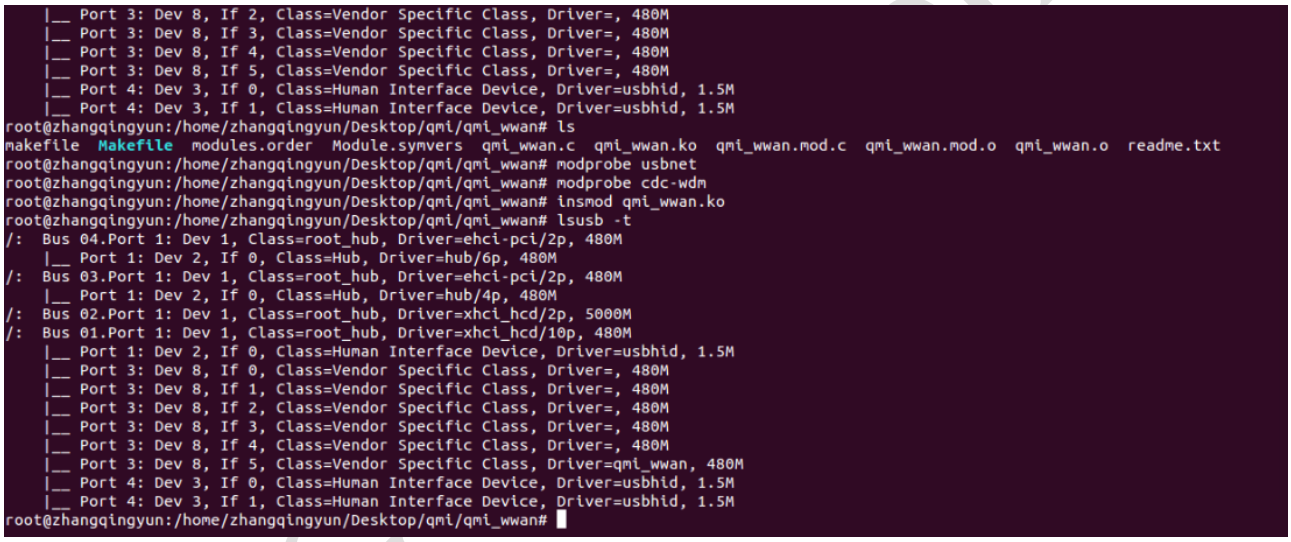
For linux pc, go to qmi_wwan folder,just make and qmi_wwan.ko will generate.

modprobe usbnet

modprobe cdc-wdm

insmod qmi_wwan.ko .

you will find qmi_wwan driver have been correct loaded , like the following picture.



Note:
Some low-level kernels do not support qmi_wwan.

10.1 Add Kernel Configuration Items

```
CONFIG_USB_WDM=y
CONFIG_USB_NET_DRIVERS=y
#If user wants to compile into a module, then user can set it to:
CONFIG_USB_NET_QMI_WWAN=m
CONFIG_USB_NET_QMI_WWAN=y
```

10.2 Adding MeiG Module Into the Driver

staticconststructusb_device_id products[] /* Append at the end

```
--- a/drivers/net/usb/qmi_wwan.c
+++ b/drivers/net/usb/qmi_wwan.c
@@ -1351,7 +1351,8 @@ static intqmi_wwan_resume(structusb_interface *intf)
       {QMI_GOBI_DEVICE(0x1199, 0x901b)},     /* Sierra Wireless MC7770 */
       {QMI_GOBI_DEVICE(0x12d1, 0x14f1)},     /* Sony Gobi 3000 Composite */
       {QMI_GOBI_DEVICE(0x1410, 0xa021)},     /* Foxconn Gobi 3000 Modem device (Novatel
E396) */

+{QMI_FIXED_INTF(0x05c6, 0xf601, 5)},    /* Meig SLM868 */
+      {QMI_FIXED_INTF(0x2dee, 0x4d22, 5)},   /* Meig SRM815 */
       { }                                    /* END */
 };
 MODULE_DEVICE_TABLE(usb, products);
```

10.3 Compile the Dial Tool

qmi_wwan needs to use meig-cm tool to dial. The compiling method of Meig-cm tool is as following：

```
#Compile the dial tool
cd meig-cm

#Compiling methoing in PC system
Make

#Compiling the tool in embedded system, needs cross compiling
make CROSS_COMPILE=aarch64-linux-android-

#After compiling the dialing tool of meig-cm, copy it into the machine
```

After inserting the module, a network card named wwan0 will be generated.

10.4 Dialing

Using the compiled meig-cm tool to dial

```
#Parameter description:
#-sSet APN
#-6Support ipv4v6 both protocols
#-iSet network card name, in case the network card name was modified

#For example, set China Mobile SIM card APN:
meig-cm -s cmnet
```

11 MBIM Dialing

11.1 Add Kernel Configuration Items

```
CONFIG_USB_NET_DRIVERS=y
CONFIG_USB_NET_CDC_NCM=y
# If user wants to compile into a module, then user can set it to:
CONFIG_USB_NET_CDC_MBIM=m
CONFIG_USB_NET_CDC_MBIM=y
```

After modifying the above configuration items, the kernel will support mbim driver by default.

11.2 Dialing

Currently, Windows and ubuntu18 and above versions of mbim are driver-free and can be enabled directly on the network connection.

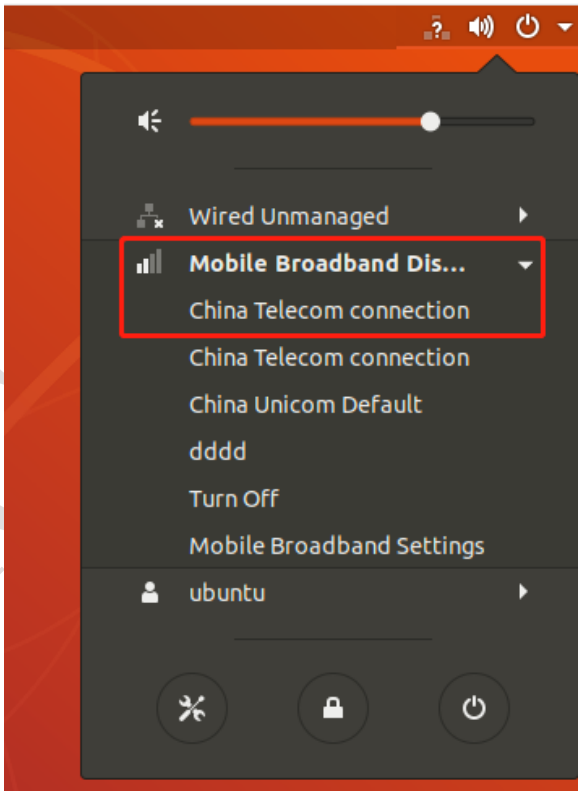


Figure 8 MBIM Dialing

12 RNDIS Dialing

RNDIS dialing is similar to ECM dialing, and is generally an automatic dialing method.

12.1 Add Kernel Configuration Items

The ECM driver is generally loaded by default in the linux kernel;if not, it can be loaded as follows for linux PC,

```
modprobeusbnet
modprobecdc_ether
modproberndis_host
```

For embedded linux environment, you can turn on the following switches in the kernel configuration, and verify after compiling and updating the kernel.

```
CONFIG_USB_USBNET=y
CONFIG_USB_NET_CDCETHER=y
CONFIG_USB_NET_RNDIS_HOST=y
```

12.2 Dialing

Normally, the RNDIS version of the module automatically dials by default, similar to plug and play, and usually generates a network port named `usbX`.

For example, if the generated network card is `usb0`, you can check whether you have obtained the IP through `ifconfig usb0`. If you have obtained it, you can directly ping for verification.

For some embedded linux platforms that cannot automatically request dhcp, you can use `udhcpc`, `dhclient`, `dhcpcd`and other tools to obtain and set ip information.

For example:

```
#Note that whether udhcpc can successfully set ip, gateway, dns and other information, depends on the configuration script,
Default path:"/etc/udhcpc/default.script". User can also specify the script through the -s parameter
udhcpc -i usb0 -s /etc/udhcpc/default.script
```

13 PCIe Dialing

For embedded platforms, user needs to use a cross-compilation tool chain and kernel to compile and generate the mhi driver (pcie_mhi.ko), which is used to generate the network card device.

13.1 Prepare and Load the Driver

Before dialing, use lspci to check whether the VID and PID information of the module is detected or not. If not, check whether the module is connected properly.

For example:

```
~ # busyboxlspci
03:00.0 Class ff00: 17cb:0306
```

Load mhi driver

```
~ # insmodpcie_mhi.ko
# Will generate the following device nodes
~ # ls /dev/mhi_*
/dev/mhi_BHI      /dev/mhi_DIAG     /dev/mhi_DUN      /dev/mhi_LOOPBACK  /dev/mhi_MBIM
```

13.2 Dialing

```
#Single IPv4 Dialing
~ #./meig-cm -d /dev/mhi_MBIM
#IPv4v6 dialing
~ #./meig-cm -d /dev/mhi_MBIM -4 -6
#After dialing is successful, the generated network card is mhi0, use can PING to verify
ping -I mhi0 www.baidu.com
```

14 SIM Card Hot Swap Support

For modules that support SIM card hot swap function, user can use the following AT commands to enable them. Note:

After setting, restart the module to take effect

```
# Enable SIM card hot swap function, the detection pin is active at low level
AT+MGCFG=2,1,0

# Enable hot plugging of SIM card, and the detection pin is active at high level
AT+MGCFG=2,1,1

# Disable SIM card hot swap
AT+MGCFG=2,0,0
```

15 IPv6 Functional Verification

At present, not all MeiGmodules support IPv6 function, user needs to confirm with MeiGFAE team in advance.

15.1 IPv6 Connectivity Verification

You can use the ping6 command to ping the IPv6 address to verify. The following addresses are known to be available:

DNS server of Beijing University of Posts and Telecommunications
2001:da8:202:10::36
2001:da8:202:10::37

DNS server of Beijing University of Science and Technology
2001:da8:208:10::6

```
root@56iqDS:/etc # ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.187.213.130  P-t-P:10.64.64.64  Mask:255.255.255.255
          inet6 addr: 240e:bf:d427:f0df:acb1:4d03:d9fc:c534/64 Scope: Global
          inet6 addr: fe80::acb1:4d03:d9fc:c534/10 Scope: Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1280  Metric:1
          RX packets:47 errors:0 dropped:0 overruns:0 frame:0
          TX packets:95 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:4113 TX bytes:6416

root@56iqDS:/etc # ping6 2001:da8:202:10::36
PING 2001:da8:202:10::36(2001:da8:202:10::36) 56 data bytes
64 bytes from 2001:da8:202:10::36: icmp_seq=1 ttl=46 time=200 ms
64 bytes from 2001:da8:202:10::36: icmp_seq=2 ttl=46 time=98.7 ms
64 bytes from 2001:da8:202:10::36: icmp_seq=3 ttl=46 time=86.6 ms
^C
--- 2001:da8:202:10::36 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 86.621/128.648/200.561/51.091 ms
root@56iqDS:/etc #
```

Figure 9 IPv6 Ping

15.2 IPv6 Functional Test

Visit the address in the web browser http://www.test-ipv6.com/, user can verify IPv6 support.

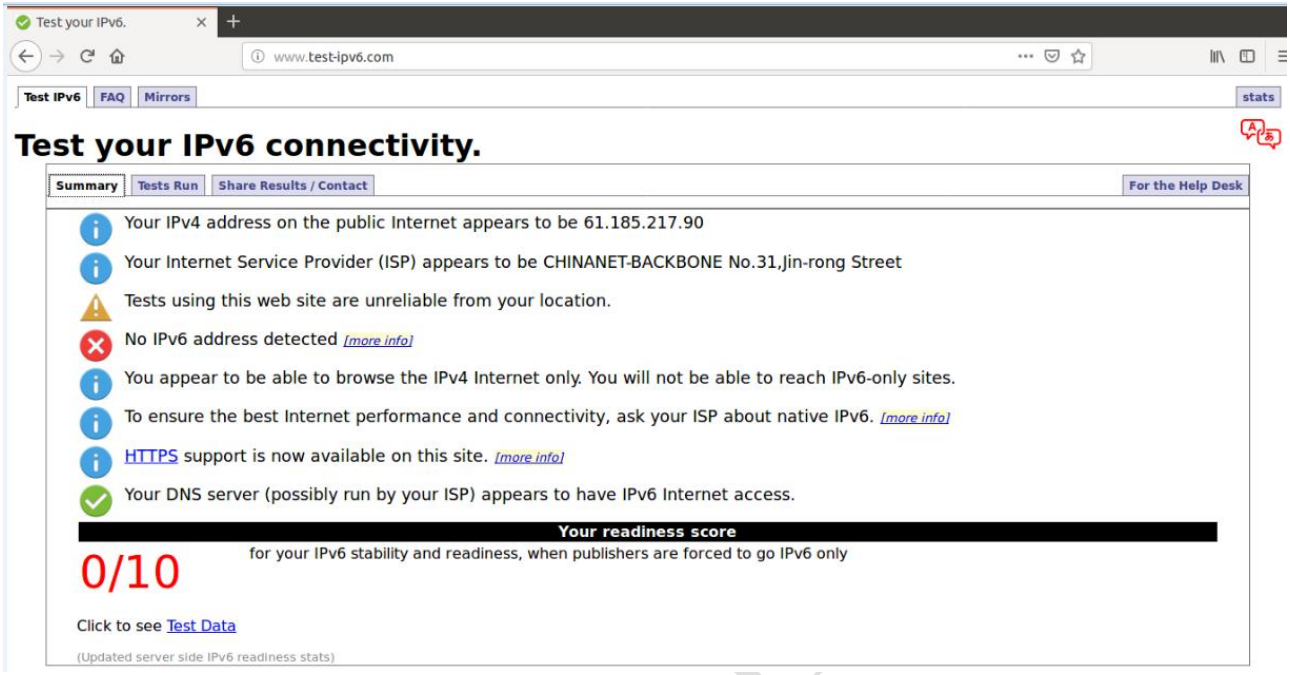


Figure 10 IPv6 Connectivity test

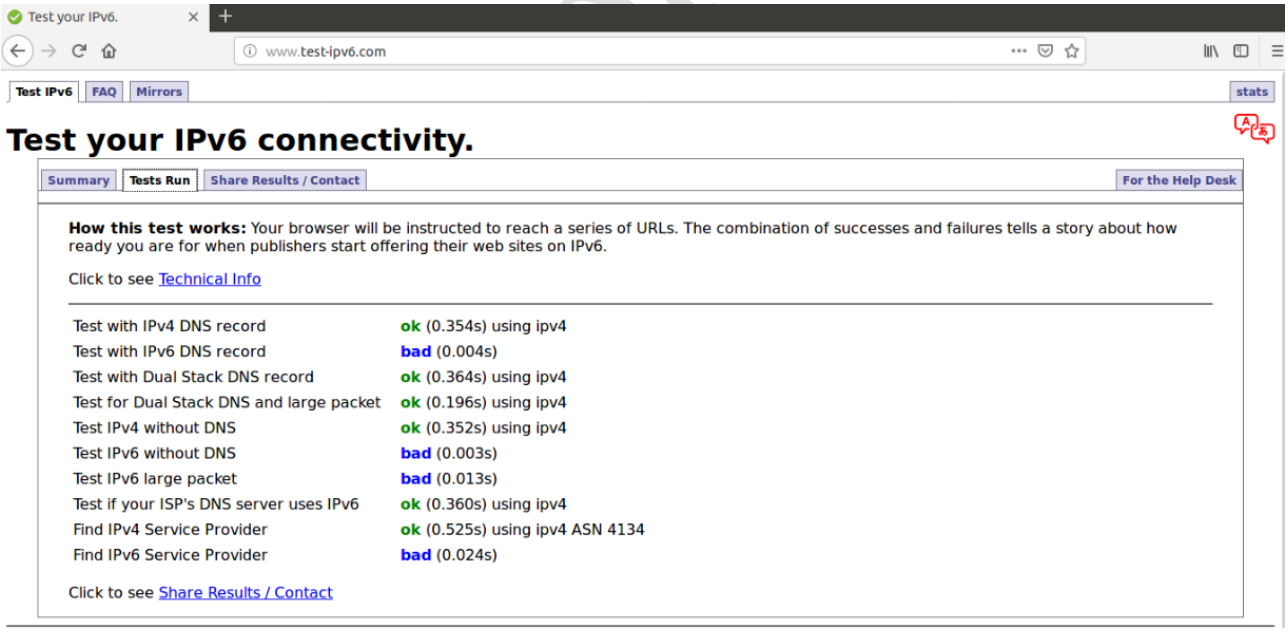


Figure 11 Connectivity test

16 FAQ

16.1 Whether the Module is Connected Normally

Use lsusb to view the vendor id and product id of all connected usb devices to confirm whether the module is connected. Such as:

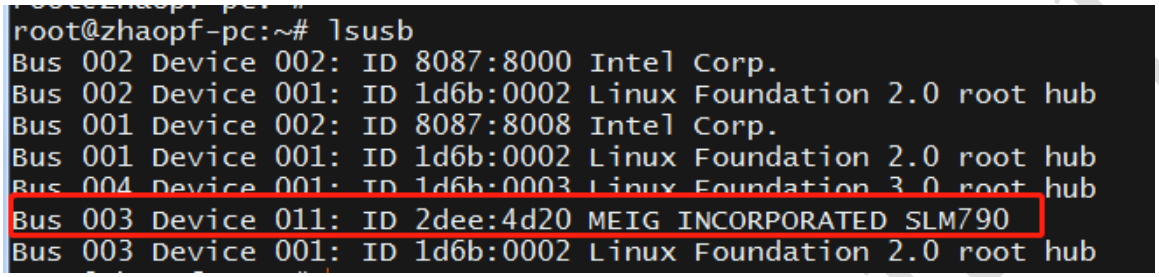


Figure 12 Verify USB Device

If the corresponding module is not detected, firstly check whether the USB port connected to the module is in host mode. If it is a host, user needs to check whether the power supply of the module is normal, and measure to confirm whether the module is turned on. If the module is turned on, user needs to continue to check whether the usb wiring is normal and make sure about it.

16.2 SIM Card Connection Status

Use AT command “AT+CPIN?” to query the SIM card status

```
AT> AT+CPIN?
AT< +CPIN: READY
```

16.3 Signaling Verification

Use AT command “AT^HCSQ?” to check the device signaling quality status

```
AT> AT^HCSQ?
AT< ^HCSQ: 0,0,"LTE",54,14,52,186
```

16.4 Device Registration Verification

Use AT command “AT+COPS?” to query whether device registered to network or not, for example:

```
AT> AT+COPS=3,0;+COPS?;+COPS=3,1;+COPS?;+COPS=3,2;+COPS?
AT< +COPS: 0,0,"004300 003F",7
AT< +COPS: 0,1,"00 003F",7
AT< +COPS: 0,2,"46011",7
```

16.5 USB Driver Verification

If there is no `/dev/ttyUSB*` device, user needs to check whether the option driver is loaded

```
lsmod | grep option
```

17 Appendix

17.1 AT+CGDCONTPDP Context Definition Command

Using the write command, parameters can be defined for the PDP context, which is identified by the local context identification parameter <cid>. The special form of the setting command +CGDCONT=<cid> will make the value of the context number <cid> an undefined value. The test command returns a composite value. If the MT supports several PDP types <PDP_type>, the parameter value range of each <PDP_type> is returned on a separate line.

17.2 AT$QCRMCALL NDI Dialing Command

This command is a RMNET-based dialing command to connect and disconnect the data.

Note:
Please refer to MeiG Module AT Commands Manual in detail for above AT commands

17.3 AT^NDISDUP NDIS Dialing

This command is used for NDIS dialing

- at^ndisdup=1,1：NDIS dialing
- at^ndisdup=1,0：DisconnectNDIS dialing.

This command is only use in NDIS port mode.

- **Syntax**

Command	Response
^NDISDUP=<pdpid>,<connect>[,<APN>[<username>[,<passwd>[,<authpref>]]]]	<CR><LF>OK<CR><LF> In case of Error: <CR><LF>+CME ERROR: <err><CR><LF>
^NDISDUP?	<CR><LF>OK<CR><LF>
^NDISDUP=?	GU Mode： <CR><LF>^NDISDUP：(list of supported<pdpid>s),<0-1><CR><LF><CR><LF>OK<CR><LF> GUL Mode： <CR><LF>^NDISDUP：(list of supported <pdpid>s),<0-1><CR><LF><CR><LF>OK<CR><LF>

- **Parameter Description**

Parameter	Description
<pdpid>	Integer type, PDP context identifier. GU:1~16 (Currently only supports 11, and can be expanded to 16 in the future) GUL:1~20
<connect>	Integer value set the connection status. The values are as follows: 0: Disconnect; 1: Setup connection
<APN>	String type, access point name, 0~99byte
<username>	String type, username, 0~255byte
<passwd>	String type, password, 0~255byte
<authpref>	Integer value, authentication protocol. The values are as follows: 1: PAP; 2: CHAP; 3: MsChapV2 (Not support for now)

- **Example**

- NDIS Dialing

AT^NDISDUP=1,1
OK

^DATACONNECT

^NDISSTAT:1,,,"IPV4"

- Query command

AT^NDISDUP?
OK

- Test Command

AT^NDISDUP=?
^NDISDUP：(1-20),(0-1)
OK