

Product Name	YUGA 9xo7 module
Number of Pages	14
Produce Version	V1.30
Date	2016/10/12

9X07 Module LINUX Integration User Manual

V1.30



Shang Hai YUGE Information Technology co., LTD

All rights reserved



Update records

Version	Date	Author	Description
V1.10	2016/07/11	YUGE document group	Initial
V1.20	2016/08/15	YUGE document group	Add the content about GobiNet
V1.30	2016/10/12	YUGE document group	Modify the AT commands and related instructions.

Shanghai YUGE



Contents

1. Linux system to load the YUGE 9X07 module USB serial port drive system components...	1
2. Linux system YUGE 9X07 module pppd dialing instructions.....	1
2.1 View device access.....	1
2.2 Filtering related interfaces.....	1
3. Linux system under the interactive AT process.....	3
4. Linux system pppd dial-up process.....	5
5. Linux system YUGE 9X07 module GobiNet dialing instructions.....	7
5.1 Add the GobiNet Drive System components.....	7
5.2 Checking Device Access.....	7
5.3 Filtering the RMNET Interface.....	8
5.4 Compile the GobiNet driver.....	9
5.5 Load the GobiNet driver.....	9
5.6 GobiNet Dial Related Commands.....	9
5.7 network card Get IP.....	10
5.8 View the IP address and network test.....	10
6. FAQ.....	11



1. Linux system to load the YUGE 9X07 module USB serial port drive system components

In Linux systems usually use USB to serial port driver. To add a driver, you need to configure the Linux kernel as follows:

```
cd kernel
```

```
make menuconfig
```

```
device drivers -> usb support -> usb serial converter support
```

Select the following components:

```
USB driver for GSM and CDMA modems
```

Check to save the configuration.

2. Linux system YUGE 9X07 module pppd dialing instructions

2.1 View device access

Use the command `lsusb` view USB device, confirm the discovery of equipment.

```
test@yuge-info:~$ lsusb
Bus 001 Device 003: ID 05c6:9025 Qualcomm, Inc. Qualcomm HSUSB Device
```

As shown above, the module default VID, PID is 0x05C6,0x9025.

2.2 Filtering related interfaces

9X07 module using pppd dialing mode, only need to load the AT port and modem port, other unrelated ports to be filtered, the following provides three solutions can be selected according to the actual situation.

1. kernel version supports blacklist, in `option.c` (path is generally `drivers/usb/serial/option.c`) `option_ids` add blacklist, will enable the driver to automatically skip the blacklist specified interface.

In the first file, find the array of `option_ids []`, Add the VID and PID of the YUGE module, VID 0x05C6 already exists, according to the actual situation, add the VID, PID according to the following method:

```
#define QUALCOMM_VENDOR_ID          0x05C6
#define YUGA_PRODUCT_9X07          0x9025
```



Set interface 0,1 not load the option driver, add the following:

```

/***** Add for YUGA 9X07 *****/
static const struct option_blacklist_info YUGA_9X07_blacklist = {
    .reserved = BIT(0) | BIT(1) | BIT(4),
};
/*****/

```

Add blacklist to the option_ids array:

```

static const struct usb_device_id option_ids[] = {
/***** Add for YUGA 9X07 modem *****/
{ USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07),
  .driver_info=(kernel_ulong_t)&YUGA_9X07_blacklist},
/*****/

```

2. For the kernel version does not support to set blacklist in the *option_ids* array, first add the module's VID and PID:

```

#define QUALCOMM_VENDOR_ID          0x05C6
#define YUGA_PRODUCT_9X07          0x9025

static const struct usb_device_id option_ids[] = {
/***** Add for YUGA 9X07 *****/
{ USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07) },
/*****/

```

In the *option_probe* function to determine the current interface number to filter:

```

/***** Add for YUGA 9X07 *****/
if(le16_to_cpu(serial->dev->descriptor.idVendor) == QUALCOMM_VENDOR_ID &&
    le16_to_cpu(serial->dev->descriptor.idProduct) == YUGA_PRODUCT_9X07 &&
    (serial->interface->cur_altsetting->desc.bInterfaceNumber <= 1 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 4)) {
    printk(KERN_INFO "Discover YUGA 9X07\n");
    return -ENODEV;
}
/*****/

```

3. For the users who use of drive of *usb-serial.ko* , add the module's VID and PID in array of *option_ids[]* in *option.c*:

```

#define QUALCOMM_VENDOR_ID          0x05C6
#define YUGA_PRODUCT_9X07          0x9025

static const struct usb_device_id option_ids[] = {
/***** Add for YUGA 9X07 *****/
{ USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07) },
/*****/

```



Adding the following judgment to filter the port at the start of `usb_serial_probe()` function in `usb-serial.c`:

```

/***** Add for YUGA 9X07 *****/
if(interface->cur_altsetting->desc.bInterfaceNumber <= 1 ||
    interface->cur_altsetting->desc.bInterfaceNumber == 4)
    return -EDOM;
/*****

```

After compilation is complete, Through the inquiry order: `ls/dev/ttyU*`, should have `ttyUSB0`, `ttyUSB1` two ports, as shown below:

```

test@yuge-info:~$ ls /dev/ttyU*
/dev/ttyUSB0 /dev/ttyUSB1

```

3. Linux system under the interactive AT process

1. Insert the USIM / SIM card into the application terminal and connect the 4G full-range antenna to the RF connector of the module. Module boot, load the USB driver, access USB port: `ttyUSB0`, `ttyUSB1`.

`ttyUSB0 -> Modem`

`ttyUSB1 -> AT`

2. Start the Linux system serial application `minicom`, use the following command:

```
#minicom -s
```

In the `minicom` menu, select "*Serial port setup*", configure the "*Serial device*" to `/dev/ttyUSB1` (module serial AT (`ttyUSB1`), Modem (`ttyUSB0`) can send AT command); modify the exit to the `minicom` menu, select "*Save setup as dfl*" After saving the configuration, select "*exit*" to exit the `minicom` configuration.

3. through the `minicom` send AT commands for system testing

```
#minicom
```

Will get the following return results:

```
Welcome to minicom 2.7
```

```
OPTIONS: I18n
```

```
Compiled on Jan  1 2014, 17:13:22.
```

```
Port /dev/ttyUSB1
```

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



Enter the following command to turn on echo:

ATE

If the system is working properly, you will get the following results:

OK

Enter the following command to query the product information:

ATI

You will get the following information:

Manufacturer: Yuga Co.,Ltd.

Model: CLM920_NC5

Revision: CLM920_NC5-V1 [Oct 11 2016 11:23:29]

IMEI: 868323022554940

+GCAP: +CGSM

Enter the following command to query the product APP version information:

AT+APPVER

You will get the following information:

APP_VERSION: Oct 11 2016 19:30:30

Enter the following command to query the PIN code:

AT+CPIN?

You will get the following information:

+CPIN: READY

Enter the following command to query the signal:

AT+CSQ

The following signal strength and bit error rate information are obtained:

+CSQ: 31,99

Enter the following command to query the system information:

AT^SYSINFO

You will get the following information:



`^SYSINFO: 2,3,0,9,1` // Note: When registering at SRLTE, this instruction returns 6-bit parameters.

Enter the following command to query the CS domain (SMS, voice service) registration status:

`AT+CREG?`

Will get the following registration information:

`+CREG: 0,1`

Enter the following command to query the PS domain (data field) registration status:

`AT+CGREG?`

Will get the following registration information:

`+CGREG: 0,1`

Enter the following command to query the network operator information:

`AT+COPS?`

Will get the following operator information (different operators to return information is different, the following to China Mobile SIM card as an example)

`+COPS: 0,0,"CHINA MOBILE CMCC",7`

4.Linux system pppd dial-up process

1. Repeat the module's USB load process and AT interactive process. To ensure that the module registers correctly to the network, the signal strength CSQ returns the first parameter above 9;
2. Confirm the Linux system with pppd and chat executable. If the system does not have pppd, please install kppp which have the pppd application (pppd 2.4.3, pppd 2.4.5 is recommended);
3. In telecom 2G and 3G mode, the dialing number can be unified with other standard, support dialing using * 99 #

Dial-up Internet access in two ways:

A)Directly use our dial-up script `yuga.lte-pppd` (default APN is empty, can be set as needed), pay attention to the implementation of the authority to the script;



B) Write the pppd script and the chat script separately:

(1) File of `/etc/ppp/peers/lte` , which reads as follows:

```
# Usage: root>pppd call lte&
/dev/ttyUSB0
115200
crtscts
modem
debug
nodetach
usepeerdns
noipdefault
defaultroute
user card
password card
connect '/usr/sbin/chat -s -v -f /etc/ppp/lte-connect-chat'
```

(2) File `/etc/ppp/lte-connect-chat` , which reads as follows:

```
#/etc/ppp/lte-connect-chat
#chat script for LTE.
TIMEOUT 15
ABORT "DELAYED"
ABORT "BUSY"
ABORT "ERROR"
ABORT "NO DIALTONE"
ABORT "NO CARRIER"
"" AT
OK ATE0
OK ATDT*99***1#
CONNECT
```

After two scripts written, execute the scripts `pppd call lte &` to dial-up Internet access.

NOTE:

① YUGE 9X07 module, a variety of network can be unified using chat script in the `ATDT*99***1#` dial-up

② The user name and password in pppd script is registered in the telecommunications 2G, 3G use, no impact on other networks.



4. Test the Internet connection

To test whether to connect to the Internet, use the following command:

```
# Ping 115.239.210.27
```

If the IP address can be Ping, but the Ping domain name is blocked, the following instructions:

```
# ping www.baidu.com
```

You need to add DNS (114.114.114.114) to */etc/resolv.conf*.

5. Disconnect the Internet connection:

A) Invoke the ppp-off script we provided

B) Use the command: `# killall pppd`

5. Linux system YUGE 9X07 module GobiNet dialing instructions

The 9x07 module supports GobiNet dialing. The module PID must be cut to 9091 using the command `AT + USBCFG = 9091`.

5.1 Add the GobiNet Drive System components

GobiNet driver requires usbnet driver support, so need to configure the Linux kernel, as follows:

```
cd kernel
```

```
make menuconfig
```

```
device drivers -> Network device support -> usb Network Adapters
```

Select the following components:

Multi-purpose USB Networking Framework

Select the save configuration, recompile the kernel.

5.2 Checking Device Access

Use `lsusb` view usb device, confirm the discovery of equipment.

```
test@yuge-info:~/Documents$ lsusb
Bus 001 Device 004: ID 05c6:9091 Qualcomm, Inc.
```

As shown above, the module's VID, PID is 0x05C6,0x9091.



5.3 Filtering the RMNET Interface

As the USB serial port and RMNET are non-standard CDC devices, you need to prevent the RMNET port is loaded by the USB serial port driver can not load RMNET port driver. The following provides three solutions, according to the actual situation choose to deal with.

1.To kernel version supports blacklist, option_ids in option.c need add blacklist, it can automatically skip the blacklist specified interface when driver loading .

First add the VID and PID of the module:

```
#define QUALCOMM_VENDOR_ID          0x05C6
#define YUGA_PRODUCT_9X07_RMNET    0x9091
```

Set interface 0,2,3 not loaded option drive, add the following:

```
/****** Add for YUGA 9X07 *****/
static const struct option_blacklist_info YUGA_9X07_RMNET_blacklist = {
    .reserved = BIT(0) | BIT(2) | BIT(3),
};
/*******/
```

Add blacklist to the option_ids array:

```
/****** Add for YUGA 9X07 modem *****/
{ USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07_RMNET),
  .driver_info=(kernel_ulong_t)&YUGA_9X07_RMNET_blacklist},
/*******/
```

2.For the kernel version does not support the option_ids array set blacklist, first add the module VID and PID:

```
#define QUALCOMM_VENDOR_ID          0x05C6
#define YUGA_PRODUCT_9X07_RMNET    0x9091
```

```
static const struct usb_device_id option_ids[] = {
    /****** Add for YUGA 9X07 RMNET *****/
    { USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07_RMNET) },
    /*******/
};
```

In the option_probe function to determine the current interface number to filter:

```
/****** Add for YUGA 9X07 RMNET *****/
if((le16_to_cpu(serial->dev->descriptor.idVendor) == QUALCOMM_VENDOR_ID &&
    le16_to_cpu(serial->dev->descriptor.idProduct) == YUGA_PRODUCT_9X07_RMNET &&
    (serial->interface->cur_altsetting->desc.bInterfaceNumber == 0 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber >= 2)) {
    printk(KERN_INFO"Discover the 2nd interface for YUGA 9X07 RMNET\n");
    return -ENODEV;
}
/*******/
```

3.For the users who use of drive of usb-serial.ko , need to add the module's VID and PID:

```
#define QUALCOMM_VENDOR_ID          0x05C6
#define YUGA_PRODUCT_9X07_RMNET    0x9091

static const struct usb_device_id option_ids[] = {
    /****** Add for YUGA 9X07 RMNET *****/
    { USB_DEVICE(QUALCOMM_VENDOR_ID, YUGA_PRODUCT_9X07_RMNET) },
    /*******/
};
```



adding the following judgment to filter the RMNET port at the start of `usb_serial_probe()` function in `usb-serial.c`:

```

/***** Add for YUGA 9X07 RMNET *****/
if(interface->cur_altsetting->desc.bInterfaceNumber == 0 ||
    interface->cur_altsetting->desc.bInterfaceNumber >= 2)
    return -EDOM;
/*****/

```

After compilation is complete, through the inquiry order: `ls/dev/ttyU*`, should have `ttyUSB0`, as shown below:

```

test@yuge-info:~$ ls /dev/ttyU*
/dev/ttyUSB0

```

`ttyUSB0` is AT port, you can send AT commands via `ttyUSB0`

5.4 Compile the GobiNet driver

GobiNet driver provided in the form of the original code, compiled by the user in their own system.

Extract the kernel source file to the relevant folder, such as `drivers/net/usb`. Execute the command `make` in directory of GobiNet which formed after decompression, `GobiNet.ko` file will generated under the directory.

5.5 Load the GobiNet driver

Load the GobiNet driver with the `insmod` command: `sudo insmod GobiNet.ko`.

Use the command `ifconfig` to view the network card information, if the `usb0` indicates that the driver loaded successfully, as shown in Fig.

```

usb0      Link encap:Ethernet  HWaddr 0a:76:22:f0:42:42
          inet6 addr: fe80::876:22ff:fef0:4242/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4048 (4.0 KB)

```

5.6 GobiNet Dial Related Commands

1. Before dialing, please check the following command to confirm that the module successfully registered to the network, the specific return results refer to the AT manual.

```
AT+CPIN?
```

```
AT^SYSINFO
```

```
AT+CSQ
```



```
AT+CGREG?
```

2. After confirming that the module is registered on the network, use the following command to GobiNet dial-up and query the connection.

```
AT$QCRMCALL=1,1,1,2,1
```

//3GPP initiates dialing, 3GPP2 uses the command AT\$QCRMCALL=1,1,1,1,1

//The following message will return if dial-up successfully

```
$QCRMCALL: 1, V4 // means: connected, the protocol is IPv4
```

```
AT$QCRMCALL? // query.
```

//After the connection is successful, the following information is returned

```
$QCRMCALL: 1, V4 // means: connected, the protocol is IPv4
```

3. Disconnect the dial

```
AT$QCRMCALL=0,1
```

5.7 network card Get IP

After the successful acquisition of IP, the IP will be assigned to the network card usb0 via DHCP. You can refer to the following commands:

```
udhcpc -i usb0
```

5.8 View the IP address and network test

Enter the command `ifconfig view usb0` IP address, as shown below:

```
usb0      Link encap:Ethernet  HWaddr 0a:76:22:f0:42:42
          inet addr:10.62.171.67  Bcast:10.62.171.71  Mask:255.255.255.248
          inet6 addr: fe80::876:22ff:fef0:4242/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:126 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:616 (616.0 B)  TX bytes:25108 (25.1 KB)
```

After that, you can test whether to connect to the Internet by *ping*.



6. FAQ

6.1 Q: I added the YUGE module's VID PID, why ls / dev / ttyU * still can not see the port?

First, make sure the module is powered on and USB is successfully connected. The commands lsusb or dmesg can be used to view the information of the connected USB devices. Make sure the module is connected to the system. Otherwise, make sure the hardware connection is correct.

After checking the module information by lsusb or dmesg, confirm that the added VID PID is correct. If check the correct, the final confirmation of the modified information is compiled into the system.

The above information is correct, you can see through the ls / dev / ttyU * port.

6.2 Q: How to manually send AT command through the echo cat ?

To ttyUSB2 sent AT command, for example, according to the following command (through ctrl + c exit)

```
sudo echo -en "AT\r\n" > /dev/ttyUSB2;cat /dev/ttyUSB2
test@yuge-info:~$ sudo echo -en "ATE\r\n" > /dev/ttyUSB2;cat /dev/ttyUSB2

OK
^C
test@yuge-info:~$ sudo echo -en "AT\r\n" > /dev/ttyUSB2;cat /dev/ttyUSB2
AT
OK
```

6.3 Q: Why can not get the initiative to report information under Linux?

You can not add the module VID PID in generic.c, you need to add in the option.c file to confirm that the module is loaded as GSM modem.