UC20&EC20 Linux

GobiNet User Guide

UMTS/HSPA/LTE Module Series

Rev. UC20&EC20_Linux_GobiNet_User_Guilde_V1.0

Date: 2015-02-27

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Office 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233
Tel: +86 21 5108 6236
Mail: info@quectel.com

Or our local office, for more information, please visit:
http://www.quectel.com/support/salesupport.aspx

For technical support, to report documentation errors, please visit:
http://www.quectel.com/support/techsupport.aspx
Or Email: Support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2015-02-27	Joe WANG	Initial

Contents

Table Index

1 Introduction

This document mainly introduces how to use the Quectel module's NDIS interface in Linux OS with GobiNet driver. When GobiNet driver is attached with Quectel module, it will create a network device and a QMI channel. The network device is named as ***ethX***, and QMI channel is named as ***/dev/qcqmiX***. The network is working for data transmission, and QMI channel is working for QMI message interaction.

The following table shows the products that this document supports.

Table 1: Supported Products

Product	PID	VID	Supported
UC20	0x9003	0x05c6	√
EC20	0x9215	0x05c6	√

2 Integrate GobiNet in Linux

In order to make use of NDIS interface for Quectel module in Linux, you must integrate the GobiNet driver source code in Linux kernel at first.

2.1. Add Source Code in Kernel

Please retrieve the appropriate kernel source code version for your system and unpack/install it.

Put the GobiNet driver source code in **($_KERNELROOT)/drivers/net/usb** directory. The **($_KERNELROOT)** should be your own root directory of kernel source code.

2.2. Cancel Serial Driver Blinding on NDIS Interface

Before integrating the GobiNet driver, you must cancel the USB serial driver binding for NDIS interface.

Open the file in the root directory of source code:

($_KERNELROOT)/drivers/usb/serial/option.c

Then modify the source code in ***option_probe***. The following modifying example is made in kernel 3.6.

```
//file: <kernel path>/drivers/usb/serial/option.c
static int option_probe(struct usb_serial *serial,
                        const struct usb_device_id *id)
{
    struct usb_wwan_intf_private *data;
    struct option_private *priv;
    struct usb_interface_descriptor *iface_desc =
                        &serial->interface->cur_altsetting->desc;
    struct usb_device_descriptor *dev_desc = &serial->dev->descriptor;

    /*
     * D-Link DWM 652 still exposes CD-Rom emulation interface in modem
     * mode.
```

```
 */
if (dev_desc->idVendor == DLINK_VENDOR_ID &&
      dev_desc->idProduct == DLINK_PRODUCT_DWM_652 &&
      iface_desc->bInterfaceClass == 0x08)
      return -ENODEV;

/* Bandrich modem and AT command interface is 0xff */
if ((dev_desc->idVendor == BANDRICH_VENDOR_ID ||
      dev_desc->idVendor == PIRELLI_VENDOR_ID) &&
      iface_desc->bInterfaceClass != 0xff)
      return -ENODEV;
/*
 * Don't bind reserved interfaces (like network ones) which often have
 * the same class/subclass/protocol as the serial interfaces.  Look at
 * the Windows driver .INF files for reserved interface numbers.
 */
if (is_blacklisted(
      iface_desc->bInterfaceNumber,
      OPTION_BLACKLIST_RESERVED_IF,
      (const struct option_blacklist_info *) id->driver_info))
      return -ENODEV;
/*
 * Don't bind network interface on Samsung GT-B3730, it is handled by
 * a separate module.
 */
if (dev_desc->idVendor == SAMSUNG_VENDOR_ID &&
      dev_desc->idProduct == SAMSUNG_PRODUCT_GT_B3730 &&
      iface_desc->bInterfaceClass != USB_CLASS_CDC_DATA)
      return -ENODEV;

// Add to cancel the binding of UC20 NDIS interface -> start
      if (dev_desc->idVendor == cpu_to_le16(0x05c6) &&
            dev_desc->idProduct == cpu_to_le16(0x9003) &&
            iface_desc->bInterfaceNumber == 0x04)
            return -ENODEV;
// Add to cancel the binding of UC20 NDIS interface -> end

// Add to cancel the binding of EC20 NDIS interface -> start
      if (dev_desc->idVendor == cpu_to_le16(0x05c6) &&
            dev_desc->idProduct == cpu_to_le16(0x9215) &&
            iface_desc->bInterfaceNumber == 0x04)
            return -ENODEV;
// Add to cancel the binding of EC20 NDIS interface -> end
```

```
    data = kzalloc(sizeof(struct usb_wwan_intf_private), GFP_KERNEL);
    if (!data)
        return -ENOMEM;

    priv = kzalloc(sizeof(*priv), GFP_KERNEL);
    if (!priv) {
        kfree(data);
        return -ENOMEM;
    }

    priv->bInterfaceNumber = iface_desc->bInterfaceNumber;
    data->private = priv;

    if (!is_blacklisted(iface_desc->bInterfaceNumber,
            OPTION_BLACKLIST_SENDSETUP,
            (struct option_blacklist_info *)id->driver_info)) {
        data->send_setup = option_send_setup;
    }
    spin_lock_init(&data->susp_lock);

    usb_set_serial_data(serial, data);

    return 0;
}
```

2.3. Add GobiNet into Kernel Tree

Kconfig file defines the config menu for Linux kernel, the following steps introduce how to add the GobiNet module into kernel tree.

1. Add ***"config USB_GOBI_NET"*** option in menu section of **Kconfig** file:

($_KERNELROOT)/drivers/net/usb/Kconfig

Details are shown as below:

```
config USB_GOBI_NET
    tristate "Gobi USB Net driver for Quectel module"
    help
    Support Quectel module.

    A modem manager with support for GobiNet is recommended.
```

```
To compile this driver as a module, choose M here: the module will be called GobiNet.
```

2. Modify the **makefile** to make sure the driver source code can be compiled:

($_KERNELROOT)/drivers/net/usb/Makefile

Add following lines at the end of **makefile**:

```
obj-$(CONFIG_USB_GOBI_NET)          +=   GobiNet.o
GobiNet-objs      :=GobiUSBNet.o QMIDevice.o QMI.o MPQMUX.o
```

2.4. Build Kernel and Compile the Driver

From the unpacking root directory of your kernel type, configure the kernel by command:

```
#make menuconfig
```

Please select the menu as the following steps:

Device Drivers --->
[*] Network device support --->
[*] USB Network Adapters --->
[M] Gobi USB Net driver for Quectel module

Then build kernel and install the new kernel image.

Step 1:

```
#make bzImage
```

Step 2:

```
#make modules
```

Step 3:

```
#make modules_install
```

Step 4:

```
#make install
```

Now reboot your system, the driver will be loaded.

3 Configure and Use GobiNet

3.1. Configure DHCP for GobiNet Driver

GobiNet driver will activate NDIS interface as ***"ethX"*** network device in Linux system. The driver only supports DHCP temporarily and you need to configure your network by modifying:

/etc/network/interfaces

Please add the following lines in the file.

```
auto eth1
iface eth1 inet dhcp
```

NOTES

1. The descriptor of new network device depends on your own system. You should replace the ***"eth1"*** with your device descriptor.
2. In some Linux versions, this step is not needed.

3.2. Configure GobiNet Feature

In order to use the autoconnect function, you can open USB AT port and send AT command:

```
AT+QCFG="rmnet/autoconnect",1
OK
```

After finishing the above configuration, you must reboot the module.

3.3. Use GobiNet Driver

After successfully loading the GobiNet driver, a network device would be created in your system.

If no devices as ***"ethX"*** are created in your system, please check the existence of the kernel module:

#lsmod | grep GobiNet

If no entries are found, load the kernel module manually with root privileges:

#modprobe GobiNet

If an error is returned, such as:

#FATAL: Module GobiNet not found

This means that the kernel module is not present in your system. Please refer to the Chapter 2 for details.

If devices have been created in your system, the system can auto connect with the mobile internet under the autoconnect mode.

You can use ***"ifconfig"*** command to check whether the new network device ***"ethX"*** gets a correct IP address.

4 Appendix A Reference

Table 2: Terms and Abbreviations

Abbreviation	Description
OS	Operating System
PID	Product ID
VID	Vendor ID
NDIS	Network Driver Interface Specification