

MT5700M-CN 5G 系列模组 **Linux 内核驱动集成指导** V200R001C20


文档版本：01
发布日期：2024-05-17

成都鼎桥通信技术有限公司

网址: <https://www.td-tech.com>
客户服务电话: 400 060 0808

版权所有©成都鼎桥通信技术有限公司 2024。保留一切权利。
非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明

 **TD Tech** 和其他商标均为成都鼎桥通信技术有限公司的商标。
本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受成都鼎桥通信技术有限公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,成都鼎桥通信技术有限公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。



目录

1 关于本文档.....	1
2 目的.....	2
3 范围.....	3
4 概要.....	4
5 USB 串口驱动集成.....	5
5.1 简要说明.....	5
5.2 使用 option.ko 模块进行 USB 驱动集成.....	5
5.2.1 修改文件 linux_src/drivers/usb/serial/option.c.....	5
5.2.2 内核的编译配置.....	6
5.3 使用 usbserial.ko 模块进行 USB 驱动集成.....	6
5.3.1 修改文件 linux_src/include/linux/usb.h 中添加如下宏：	7
5.3.2 修改文件 Linux_src/include/linux/usb/serial.h.....	7
5.3.3 修改文件 Linux_src/drivers/usb/serial/usb-serial.c.....	7
5.3.4 修改文件 Linux_src/drivers/usb/serial/generic.c.....	8
5.3.5 内核的编译配置.....	10
6 CDC ECM 驱动集成.....	11
6.1 简要说明.....	11
6.2 内核编译配置.....	11
7 NCM 驱动集成.....	12
7.1 简要说明.....	12
7.2 内核编译配置.....	12
8 RNDIS 驱动集成.....	13
8.1 简要说明.....	13
8.2 内核编译配置.....	13
9 make menuconfig 配置.....	14
9.1 USB 串口驱动编译配置项.....	14
9.2 CDC ECM 驱动编译配置项.....	16
9.3 CDC NCM 驱动编译配置项.....	17
10 拨号分配 IP.....	20
10.1 以 NCM 举例说明.....	20

11 FAQ..... 23

11.1 如何确认系统中是否已经存在正确的 USB 串口驱动..... 23

11.2 如何确认系统中是否已经存在正确的 CDC ECM 驱动..... 23

11.3 如何获取单板当前的端口映射情况信息..... 24

11.4 出现端口类似不是从 ttyUSB0 开始的情况如何处理..... 24

11.5 如何手动加载串口驱动..... 25

11.6 若无法映射端口或无法查找对应端口形态需要提供哪些 log..... 25

12 缩略语..... 26

1 关于本文档

修订记录

文档版本	发布日期	说明
01	2024-05-17	第一次发布

2 目的

本文档主要是针对MT5700M-CN 5G系列模组，基于Linux内核系统（如Android、Ubuntu和Chrome OS等）驱动集成开发活动进行相关的指导说明，主要面向基于Linux内核产品开发商的驱动开发人员。

3 范围

功能	描述	
支持的系统	Linux、Android和Chrome OS	
支持的模块制式	WCDMA/CDMA/LTE/NR	
支持的内核版本	2.6.12以前的版本	本文档不支持
	2.6.12 ~ 2.6.21	支持USB串口驱动集成（ 章节5 USB串口驱动集成 ）
	2.6.22 ~ 2.6.31	<ul style="list-style-type: none">支持USB串口驱动集成（章节5 USB串口驱动集成）支持CDC ECM驱动集成（章节6-CDC ECM驱动集成）
	2.6.32及以后的版本	<ul style="list-style-type: none">支持USB串口驱动集成（章节5 USB串口驱动集成）支持CDC ECM驱动集成（章节6-CDC ECM驱动集成）

4 概要

本文档主要介绍如何修改Linux内核代码，使客户系统（如Android等）能够正常加载模块。因本文档支持的系统都是基于Linux内核的系统，下文统一称Linux侧，基于Linux内核版本5.15.1进行修改并以此版本作为内核进行运行。

MT5700M-CN 5G在Linux侧使用的驱动需要进行内核编译配置项设置，确保驱动编译进内核。

本文中linux_src代表所下载的源代码根目录。

5 USB 串口驱动集成

5.1 简要说明

5.2 使用option.ko 模块进行USB驱动集成

5.3 使用usbserial.ko 模块进行USB驱动集成

5.1 简要说明

MT5700M-CN 5G商用版本默认接口，包括：PCUI、Diag、SerialB、SerialC、ECM、NCM、GPS。

- 各端口作用详见《 MT5700M-CN 5G系列模组USB接口应用指南 》。
- 修改USB串口驱动有以下两种方式(选择其一进行适配)。
 - option.ko(推荐)
 - usbserial.ko

下面分两部分对两种方式分别进行指导。



注意

选择其中一种方式即可，两种方式都使用将会导致驱动升级不成功。

5.2 使用 option.ko 模块进行 USB 驱动集成

这部分集成涉及修改的Linux内核源码文件为：

Linux_src/drivers/usb/serial/option.c

5.2.1 修改文件 linux_src/drivers/usb/serial/option.c

将下面代码添加至 option.c 文件中注释 /* Vendor and product IDs */之后

```
#define TDTECH_VENDOR_ID 0x3466
```

```
/* Vendor and product IDs */
```

```
#define TDTECH_VENDOR_ID 0x3466
```

将下面代码添加至 option.c 文件中static const struct usb_device_id option_ids[] 之中

```
static const struct usb_device_id option_ids[] = {  
{ USB_VENDOR_AND_INTERFACE_INFO(TDTECH_VENDOR_ID, 0xff, 0x06, 0x10) },  
{ USB_VENDOR_AND_INTERFACE_INFO(TDTECH_VENDOR_ID, 0xff, 0x06, 0x12) },  
{ USB_VENDOR_AND_INTERFACE_INFO(TDTECH_VENDOR_ID, 0xff, 0x06, 0x13) },  
{ USB_VENDOR_AND_INTERFACE_INFO(TDTECH_VENDOR_ID, 0xff, 0x06, 0x14) },  
{ USB_VENDOR_AND_INTERFACE_INFO(TDTECH_VENDOR_ID, 0xff, 0x06, 0x1B) },  
{ USB_VENDOR_AND_INTERFACE_INFO(TDTECH_VENDOR_ID, 0xff, 0x06, 0x1C) },  
}
```

5.2.2 内核的编译配置

文本配置方式，修改linux_src/目录下的.config文件。

- USB串口驱动相关的配置项：
CONFIG_USB_SERIAL=y
CONFIG_USB_SERIAL_OPTION=y
CONFIG_USB_SERIAL_WWAN=y

说明

USB串口驱动相关配置项在make menuconfig的图像界面配置方法，请参考[9.1 USB串口驱动编译配置项](#)。

5.3 使用 usbserial.ko 模块进行 USB 驱动集成

这部分集成涉及修改的Linux内核源码文件为：

```
Linux_src/include/linux/usb.h  
Linux_src/include/linux/usb/serial.h  
Linux_src/drivers/usb/serial/usb-serial.c  
Linux_src/drivers/usb/serial/generic.c
```

5.3.1 修改文件 linux_src/include/linux/usb.h 中添加如下宏：

```
#define USB_DEVICE_AND_INTERFACE_INFO(vend, prod, cl, sc, pr) \
    .match_flags = USB_DEVICE_ID_MATCH_INT_INFO | \
    | USB_DEVICE_ID_MATCH_DEVICE, \
    .idVendor = (vend), \
    .idProduct = (prod), \
    .bInterfaceClass = (cl), \
    .bInterfaceSubClass = (sc), \
    .bInterfaceProtocol = (pr) \

#define USB_VENDOR_AND_INTERFACE_INFO(vend, cl, sc, pr) \
    .match_flags = USB_DEVICE_ID_MATCH_INT_INFO | \
    | USB_DEVICE_ID_MATCH_VENDOR, \
    .idVendor = (vend), \
    .bInterfaceClass = (cl), \
    .bInterfaceSubClass = (sc), \
    .bInterfaceProtocol = (pr)
```

若部分内核已经添加，则无需添加此部分。详文如下：

```
#define USB_VENDOR_AND_INTERFACE_INFO(vend, cl, sc, pr) \
    .match_flags = USB_DEVICE_ID_MATCH_INT_INFO | \
    USB_DEVICE_ID_MATCH_VENDOR, \
    .idVendor = (vend), \
    .bInterfaceClass = (cl), \
    .bInterfaceSubClass = (sc), \
    .bInterfaceProtocol = (pr)
```

5.3.2 修改文件 Linux_src/include/linux/usb/serial.h

在文件头部添加如下代码：

```
/* td-tech: support for tdtech Module */
#define CONFIG_SUPPORT_TDTECH_MODULE
```

```
/* td-tech: support for tdtech Module */
#define CONFIG_SUPPORT_TDTECH_MODULE
```

5.3.3 修改文件 Linux_src/drivers/usb/serial/usb-serial.c

文件中函数 `usb_serial_probe()` 中添加添加下面两个代码：主要是为防止网卡接口以及debug版本下的ADB接口被误识别为USB虚拟串口

代码1：

```
#ifdef CONFIG_SUPPORT_TDTECH_MODULE

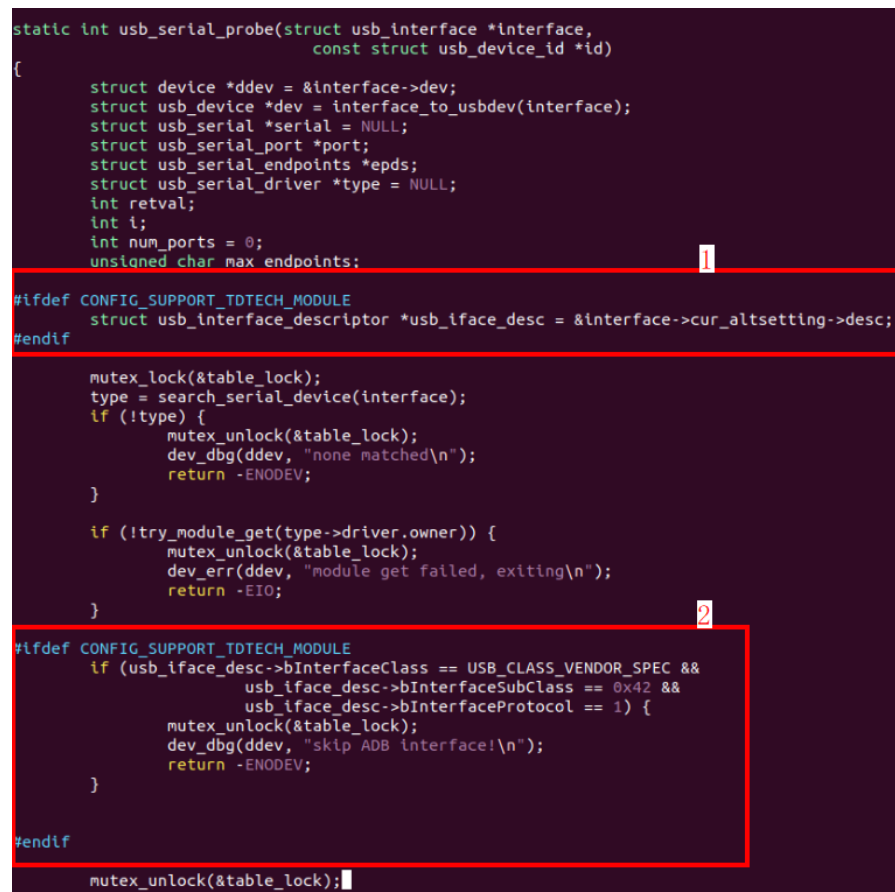
struct usb_interface_descriptor *usb_iface_desc = &interface->cur_altsetting->desc;

#endif
```

代码2:

```
#ifdef CONFIG_SUPPORT_TDTECH_MODULE

    if (usb_iface_desc->bInterfaceClass == USB_CLASS_VENDOR_SPEC &&
        usb_iface_desc->bInterfaceSubClass == 0x42 &&
        usb_iface_desc->bInterfaceProtocol == 1) {
        mutex_unlock(&table_lock);
        dev_dbg(ddev, "skip ADB interface!\n");
        return -ENODEV;
    }
#endif
```



```
static int usb_serial_probe(struct usb_interface *interface,
                           const struct usb_device_id *id)
{
    struct device *ddev = &interface->dev;
    struct usb_device *udev = interface_to_usbdev(interface);
    struct usb_serial *serial = NULL;
    struct usb_serial_port *port;
    struct usb_serial_endpoints *epds;
    struct usb_serial_driver *type = NULL;
    int retval;
    int i;
    int num_ports = 0;
    unsigned char max_endpoints;

#ifdef CONFIG_SUPPORT_TDTECH_MODULE
    struct usb_interface_descriptor *usb_iface_desc = &interface->cur_altsetting->desc;
#endif

    mutex_lock(&table_lock);
    type = search_serial_device(interface);
    if (!type) {
        mutex_unlock(&table_lock);
        dev_dbg(ddev, "none matched\n");
        return -ENODEV;
    }

    if (!try_module_get(type->driver.owner)) {
        mutex_unlock(&table_lock);
        dev_err(ddev, "module get failed, exiting\n");
        return -EIO;
    }

#ifdef CONFIG_SUPPORT_TDTECH_MODULE
    if (usb_iface_desc->bInterfaceClass == USB_CLASS_VENDOR_SPEC &&
        usb_iface_desc->bInterfaceSubClass == 0x42 &&
        usb_iface_desc->bInterfaceProtocol == 1) {
        mutex_unlock(&table_lock);
        dev_dbg(ddev, "skip ADB interface!\n");
        return -ENODEV;
    }
#endif

    mutex_unlock(&table_lock);
```

5.3.4 修改文件 Linux_src/drivers/usb/serial/generic.c

文件头部添加如下定义:

```
#ifdef CONFIG_SUPPORT_TDTECH_MODULES

    static struct usb_device_id generic_device_ids[10]; /*为后面需要添加VID、PID预留空间*/

#else
```

static struct usb_device_id generic_device_ids[2]; /*定义内核原来会有，将其放进条件判断中*/

#endif

```
#ifndef CONFIG_SUPPORT_TDTECH_NCM_MODULE
    static struct usb_device_id generic_device_ids[10];
#else
    static struct usb_device_id generic_device_ids[2]; /* Initially all zeroes. */
#endif
```

函数 **usb_serial_generic_register ()** 添加如下代码：

```
#ifndef CONFIG_SUPPORT_TDTECH_MODULE

    int i = 0;

    const __u16 vid_pid_group[][2] = {
        {vendor, product},
        {0x3466, 0x3301}
    };

    if (sizeof(generic_device_ids)/sizeof(generic_device_ids[0]) <
        sizeof(vid_pid_group)/sizeof(vid_pid_group[0])) {
        printk(KERN_ERR"%s : generic_device_ids[] overflow!\n",__func__);
        return -1;
    }

    for (; i < sizeof(vid_pid_group)/sizeof(vid_pid_group[0]); i++)
    {
        generic_device_ids[i].idVendor = vid_pid_group[i][0];
        generic_device_ids[i].idProduct = vid_pid_group[i][1];
        generic_device_ids[i].match_flags = USB_DEVICE_ID_MATCH_VENDOR |
        USB_DEVICE_ID_MATCH_PRODUCT;
    }

#else

    /* 其原生代码插入 */

#endif
```

```
int usb_serial_generic_register(void)
{
    int retval = 0;

#ifdef CONFIG_USB_SERIAL_GENERIC
#ifdef CONFIG_SUPPORT_TDTech_MODULE

    int i = 0;

    const __u16 vid_pid_group[][2] = {
        {vendor, product},
        {0x3460, 0x3301}
    };

    if (sizeof(generic_device_ids)/sizeof(generic_device_ids[0]) < sizeof(vid_pid_group)/sizeof(vid_pid_group[0])) {
        printk(KERN_ERR "%s : generic_device_ids[] overflow!\n", __func__);
        return -1;
    }

    for (; i < sizeof(vid_pid_group)/sizeof(vid_pid_group[0]); i++)
    {
        generic_device_ids[i].idVendor = vid_pid_group[i][0];
        generic_device_ids[i].idProduct = vid_pid_group[i][1];
        generic_device_ids[i].match_flags = USB_DEVICE_ID_MATCH_VENDOR | USB_DEVICE_ID_MATCH_PRODUCT;
    }
#else
    generic_device_ids[0].idVendor = vendor;
    generic_device_ids[0].idProduct = product;
    generic_device_ids[0].match_flags =
        USB_DEVICE_ID_MATCH_VENDOR | USB_DEVICE_ID_MATCH_PRODUCT;
#endif
    retval = usb_serial_register_drivers(serial_drivers,
        "usbserial_generic", generic_device_ids);
#endif
    return retval;
}
```

5.3.5 内核的编译配置

文本配置方式，修改linux_src/目录下的.config文件。

- USB串口驱动相关的配置项：
CONFIG_USB_SERIAL=y
CONFIG_USB_SERIAL_GENERIC=y
CONFIG_USB_FUNCTIONS_GENERIC=y

说明

USB串口驱动相关配置项在make menuconfig的图像界面配置方法，请参考[9.1 USB串口驱动编译配置项](#)。

6 CDC ECM 驱动集成

6.1 简要说明

6.2 内核编译配置

6.1 简要说明

CDC ECM驱动是MT5700M-CN 5G模块适配标准的ECM通用驱动，无需额外的代码修改，直接配置编译项即可。该功能支持内核2.6.22及以后的版本。

6.2 内核编译配置

文本配置方式，修改Linux内核的编译配置（在`linux_src/`目录下的`.config`文件中）。

CDC ECM驱动的相关配置项：

`CONFIG_USB_USBNET=y`

`CONFIG_USB_NET_CDCETHER=y`

`CONFIG_NETDEVICES=y`

下面这个参数适用于arm架构的PC，编译arm架构的时候请加上这个参数（较少场景使用）：

`CONFIG_USB_CONFIGFS_ECM=y`

说明

make menuconfig的图像界面配置，请参考[9.3 CDC NCM驱动编译配置项](#)。

7 NCM 驱动集成

7.1 简要说明

7.2 内核编译配置

7.1 简要说明

NCM驱动是MT5700M-CN 5G 5G模块适配标准的NCM通用驱动，无需额外的代码修改，直接配置编译项即可。该功能支持内核2.6.22及以后的版本。

7.2 内核编译配置

文本配置方式，修改Linux内核的编译配置（在`linux_src/`目录下的`.config`文件中）。

NCM驱动的相关配置项：

`CONFIG_USB_USBNET=y`

`CONFIG_NETDEVICES=y`

下面这个参数适用于arm架构的PC，编译arm架构的时候请加上这个参数（较少场景使用）：

`CONFIG_USB_CONFIGFS_NCM=y`

说明

make menuconfig的图像界面配置，请参考[9.3 CDC NCM驱动编译配置项](#)。

8 RNDIS 驱动集成

8.1 简要说明

8.2 内核编译配置

8.1 简要说明

RNDIS驱动是MT5701G-CN模块适配标准的RNDIS通用驱动，无需额外的代码修改，直接配置编译项即可。该功能支持内核2.6.22及以后的版本。

8.2 内核编译配置

文本配置方式，修改Linux内核的编译配置（在`linux_src/`目录下的`.config`文件中）。

NCM驱动的相关配置项：

`CONFIG_USB_USBNET=y`

`CONFIG_NETDEVICES=y`

`CONFIG_USB_NET_RNDIS_HOST=y`

下面这个参数适用于arm架构的PC，编译arm架构的时候请加上这个参数（较少场景使用）：

`CONFIG_USB_CONFIGFS_NCM=y`

说明

make menuconfig的图像界面配置，请参考[9.3 CDC NCM驱动编译配置项](#)。

9 make menuconfig 配置

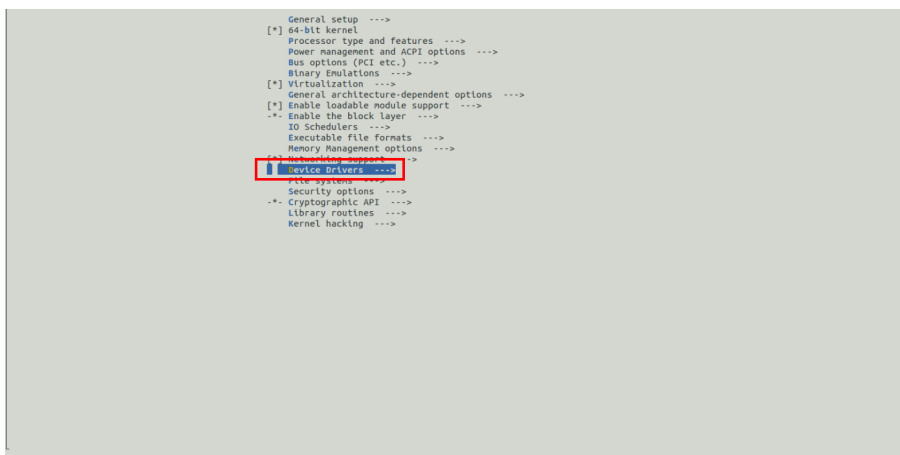
- 9.1 USB串口驱动编译配置项
- 9.2 CDC ECM 驱动编译配置项
- 9.3 CDC NCM驱动编译配置项

9.1 USB 串口驱动编译配置项

/usr/src/linux_src/ 目录下执行命令 “make menuconfig”

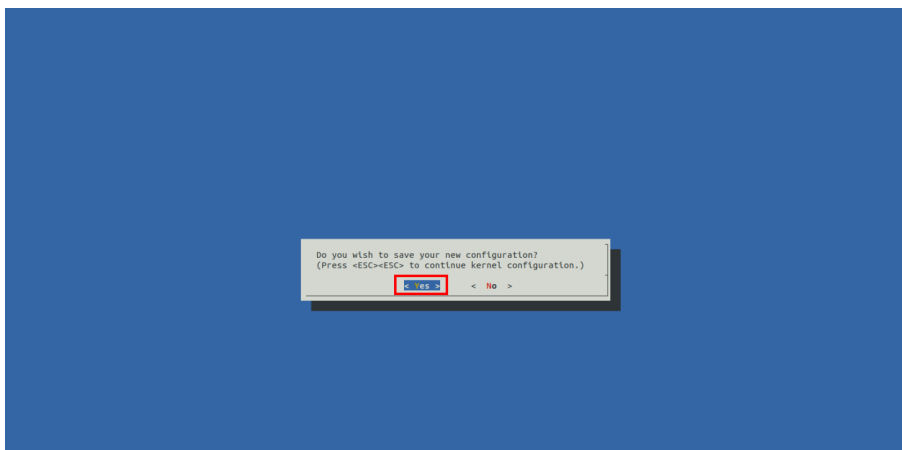
```
root@tdtech-Produ:/usr/src/linux-5.15.1# make menuconfig
```

出现界面中进行如下配置：





保存退出，自动更新 .config 文件



/usr/src/linux_src/ 目录下执行 make 进行编译

9.2 CDC ECM 驱动编译配置项

/usr/src/linux_src/ 目录下执行命令 “make menuconfig”

```
root@tdtech-Produ:/usr/src/linux-5.15.1# make menuconfig
```

出现界面中进行如下配置：



```

--- Network device support
[*] Network core driver support
<> Bonding driver support
<> Dummy net driver support
<> WireGuard secure network tunnel
<> EQL (serial line load balancing) support
[ ] Fibre Channel driver support
<> Intermediate Functional Block support
<> Ethernet team driver support ----
<> MAC-VLAN support
<> IP-VLAN support
<> Virtual extensible Local Area Network (VXLAN)
<> Generic Network Virtualization Encapsulation
<> Bare UDP Encapsulation
<> GPRS Tunneling Protocol datapath (GTP-U)
<> IEEE 802.1AE MAC-level encryption (MACsec)
<+> Network console logging support
<> Universal TUN/TAP device driver support
[ ] Support for cross-endian vnet headers on little-endian kernels
<> Virtual ethernet pair device
<> Virtual netlink monitoring device
<> ARCNet support ----
[*] Ethernet driver support ---
<> FDDI driver support
[ ] HIPPI driver support
<> General Instruments Surfbord 1000
<+> PHY Device support and Infrastructure ---
<+> RDIO bus device drivers ---
<+> PCS device drivers ---
<> PPP (point-to-point protocol) support
<+> SLIP (serial line) support
<+> USB Network Adapters ---
[*] Wireless LAN ---
[ ] Man Interfaces support ----
Wireless WAN ---
<> VMware VMXNET3 ethernet driver
<> FUJITSU Extended Socket Network Device driver
<> Simulated networking device
<> Fallover driver
[ ] ISDN support ----

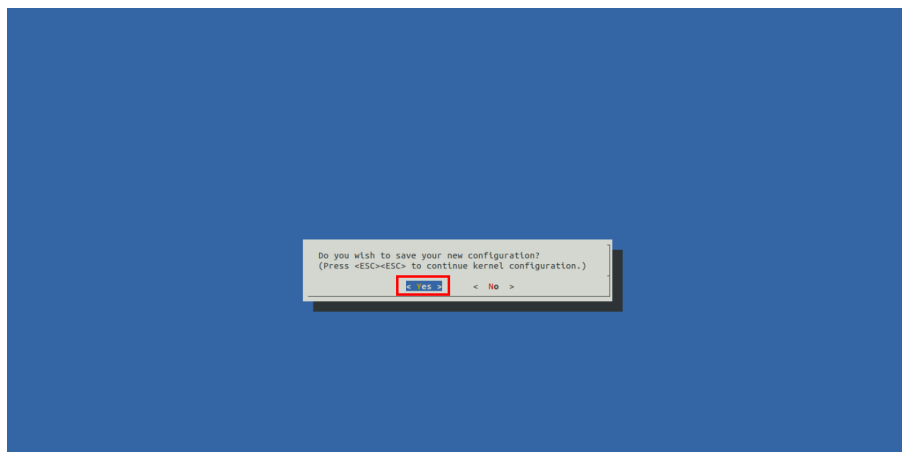
```

```

--- USB Network Adapters
<> USB CATC NetHate-based Ethernet device support (NEW)
<> USB KLSI KLSUSIM1-based ethernet device support (NEW)
<> USB Pegasus/Pegasus-II based ethernet device support (NEW)
<> USB RTL8150 based ethernet device support (NEW)
<> Realtek RTL8152/RTL8153 Based USB Ethernet Adapters (NEW)
<+> Microchip LAN78XX Based USB Ethernet Adapters (NEW)
<+> Multi-purpose USB Networking Framework
<+> ASIX AX88179/178A USB 3.0/2.0 to Gigabit Ethernet (NEW)
<+> ASIX AX88179/178A USB 3.0/2.0 to Gigabit Ethernet (NEW)
<+> CDC Ethernet support (smart devices such as cable modems)
<+> CDC Ethenet support (NEW)
<+> CDC NCM support (NEW)
<+> Huawei NCM embedded AT channel support (NEW)
<+> CDC NCM support (NEW)
<> Davicom DM96xx based USB 10/100 ethernet devices (NEW)
<> Corechip-sz SR9700 based USB 1.1 10/100 ethernet devices (NEW)
<> Corechip-sz SR9800 based USB 2.0 10/100 ethernet devices (NEW)
<> SMSC LAN75XX based USB 2.0 gigabit ethernet devices (NEW)
<> SMSC LAN95XX based USB 2.0 10/100 ethernet devices (NEW)
<> genesys GL620USB-A based cables (NEW)
<+> Netchip 1080 based cables (Laplink, ...) (NEW)
<> Prolific PL-2301/2302/25A1/27A1 based cables (NEW)
<> Netchip NC5730 based Ethernet adapters (NEW)
<> Host for RNDIS and ActiveSync devices (NEW)
<+> Simple USB Network Links (CDC Ethernet subset) (NEW)
[ ] All HSO32 based 'USB 2.0 Data Link' cables (NEW)
[ ] Anchorchips 2720 based cables (Xlrcon PGUNET, ...) (NEW)
[*] eTEK based host-to-host cables (Advance, Belkin, ...) (NEW)
[*] Embedded ADN Linux Links (UPag, ...) (NEW)
[ ] Epson 2888 based firmware (DEVELOPMENT) (NEW)
[ ] KT Technology KC2190 based cables (Instant) (NEW)
<+> Sharp Zaurus (stock ROMs) and compatible (NEW)
<> Conexant CXB2310 USB ethernet port (NEW)
<> Samsung Kalnia based LTE USB modem (NEW)
<> QMI WMAN driver for Qualcomm MSM based 3G and LTE modems (NEW)
<> Option USB High Speed Mobile Devices (NEW)
<> Intellon PLC based usb adapter (NEW)
<> Apple iPhone USB Ethernet driver (NEW)
<> USB-to-WMAN Driver for Sierra Wireless modems (NEW)
v(+)

```

保存退出，自动更新 .config 文件



/usr/src/linux_src/ 目录下执行 make 进行编译

9.3 CDC NCM 驱动编译配置项

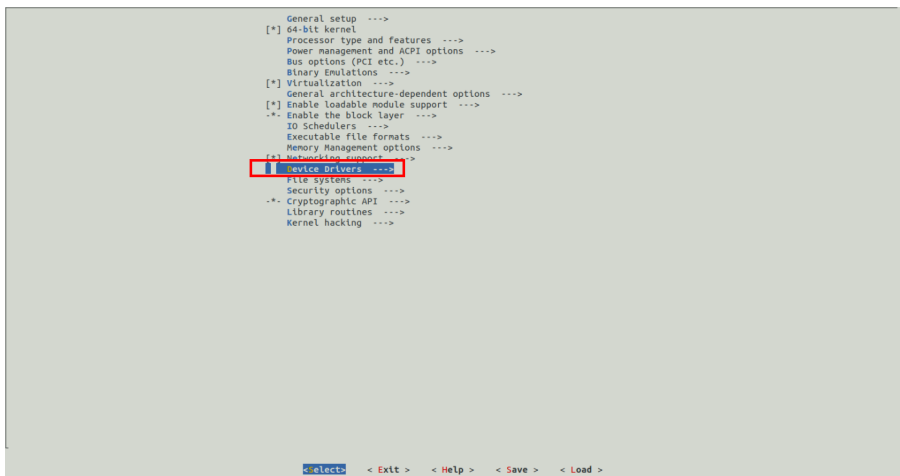
/usr/src/linux_src/ 目录下执行命令 “make menuconfig”

```

root@tdtech-Produ:/usr/src/linux-5.15.1# make menuconfig

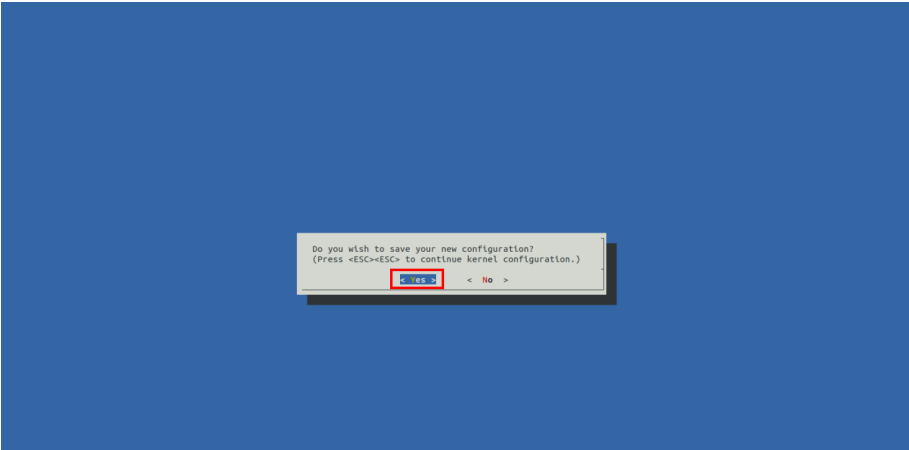
```

出现界面中进行如下配置：





保存退出，自动更新 .config 文件



/usr/src/linux_src/ 目录下执行 make 进行编译

10 拨号分配 IP

10.1 以 NCM 举例说明

10.1 以 NCM 举例说明

查看启动信息：dmesg 可以看到加载的PID；枚举出来的端口ttyUSB0, ttyUSB1, ttyUSB2, ttyUSB3, ttyUSB4；加载的网卡eth2(或者其他网卡名字)信息。

```
usb 1-10: new high-speed USB device number 4 using xhci_hcd
usb 1-10: config 2 interface 1 altsetting 1 has 2 endpoint descriptors, different from the interface descriptor's value: 3
usb 1-10: New USB device found, idVendor=3466, idProduct=3301, bcdDevice= 1.02
usb 1-10: New USB device strings: Mfr=13, Product=14, SerialNumber=15
usb 1-10: Product: TDTECH MT5700M-CN
usb 1-10: Manufacturer: TD Tech Ltd.
usb 1-10: SerialNumber: 57050M4315800093
cdc_ncm 1-10:2.0: MAC-Address: d0:a0:d6:72:a7:b5
cdc_ncm 1-10:2.0 eth0: register 'cdc_ncm' at usb-0000:00:14.0-10, CDC NCM, d0:a0:d6:72:a7:b5
usbserial_generic 1-10:2.2: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.2: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.2: generic converter detected
usb 1-10: generic converter now attached to ttyUSB0
usbserial_generic 1-10:2.3: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.3: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.3: generic converter detected
usb 1-10: generic converter now attached to ttyUSB1
usbserial_generic 1-10:2.4: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.4: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.4: generic converter detected
usb 1-10: generic converter now attached to ttyUSB2
usbserial_generic 1-10:2.5: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.5: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.5: generic converter detected
usb 1-10: generic converter now attached to ttyUSB3
usbserial_generic 1-10:2.6: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.6: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.6: generic converter detected
usb 1-10: generic converter now attached to ttyUSB4
cdc_ncm 1-10:2.0 enx0a0d672a7b5: renamed from eth0
```

检查端口：ls -all ttyUSB*

```
root@tdtech-Produ:/home/td-tech# ls -all /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 3月 2 16:40 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 3月 2 16:40 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 2 3月 2 16:40 /dev/ttyUSB2
crw-rw---- 1 root dialout 188, 3 3月 2 16:40 /dev/ttyUSB3
crw-rw---- 1 root dialout 188, 4 3月 2 16:40 /dev/ttyUSB4
```

赋予端口权限：chmod 777 /dev/ttyUSB*

```
root@tdtech-Produ:/home/td-tech# chmod 777 /dev/ttyUSB*
```

验证PCUI口正常：

开启双**Terminal**，一个输出打印结果：cat /dev/ttyUSB1，一个发送AT指令：echo -e "ate1\r\n" > /dev/ttyUSB1



查看网卡：ifconfig

```
root@tdtech-Produ:/home/td-tech# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
     ether 6c:0b:84:43:64:e5 txqueuelen 1000 (以太网)
     RX packets 65 bytes 6274 (6.2 KB)
     RX errors 0 dropped 0 overruns 0 frame 0
     TX packets 101 bytes 16974 (16.9 KB)
     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
     device interrupt 20 memory 0xf7d00000-f7d20000

eth1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
     ether 00:e0:4c:97:d4:b4 txqueuelen 1000 (以太网)
     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
     device interrupt 17

eth2: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
     ether 0c:5b:8f:27:9a:64 txqueuelen 1000 (以太网)
     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

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 (本地环回)
     RX packets 792 bytes 59061 (59.0 KB)
     RX errors 0 dropped 0 overruns 0 frame 0
     TX packets 792 bytes 59061 (59.0 KB)
     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

如果查不到网口eth2请执行命令：

ifconfig -a

ifconfig eth2 up

拨号：echo -e "at^ndisdup=1,1\r\n" > /dev/ttyUSB1



拨号之后请查看网卡状态(eth2已RUNNING)：ifconfig【注：如果没有获取到IP，输入命令：dhclient -d eth2】

```
root@tdtech-Produ:/home/td-tech# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 6c:0b:84:43:64:e5 txqueuelen 1000 (以太网)
    RX packets 66 bytes 6521 (6.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 195 bytes 33132 (33.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7d00000-f7d20000

eth1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:e0:4c:97:d4:b4 txqueuelen 1000 (以太网)
    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
    device interrupt 17

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.148.24.53 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::8579:db72:3780:5496 prefixlen 64 scopeid 0x20<link>
    inet6 240e:476:bb02:2bde:73cb:af45:2e33:620f prefixlen 64 scopeid 0x0<global>
    inet6 240e:476:bb02:2bde:1efb:b495:30b:e6d9 prefixlen 64 scopeid 0x0<global>
    ether 0c:5b:8f:27:9a:64 txqueuelen 1000 (以太网)
    RX packets 33 bytes 0030 (0.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 118 bytes 15324 (15.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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 (本地环回)
    RX packets 2138 bytes 157125 (157.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2138 bytes 157125 (157.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

测试ping 百度: ping www.baidu.com

```
root@tdtech-Produ:/home/td-tech# ping www.baidu.com
PING www.baidu.com (14.119.104.189) 56(84) bytes of data.
64 字节, 来自 14.119.104.189 (14.119.104.189): icmp_seq=1 ttl=54 时间=51.3 毫秒
64 字节, 来自 14.119.104.189 (14.119.104.189): icmp_seq=2 ttl=54 时间=61.2 毫秒
64 字节, 来自 14.119.104.189 (14.119.104.189): icmp_seq=3 ttl=54 时间=52.3 毫秒
64 字节, 来自 14.119.104.189 (14.119.104.189): icmp_seq=4 ttl=54 时间=52.7 毫秒
64 字节, 来自 14.119.104.189 (14.119.104.189): icmp_seq=5 ttl=54 时间=74.6 毫秒
64 字节, 来自 14.119.104.189 (14.119.104.189): icmp_seq=6 ttl=54 时间=65.2 毫秒
64 字节, 来自 14.119.104.189 (14.119.104.189): icmp_seq=7 ttl=54 时间=43.6 毫秒
```

11 FAQ

- 11.1 如何确认系统中是否已经存在正确的USB串口驱动
- 11.2 如何确认系统中是否已经存在正确的CDC ECM驱动
- 11.3 如何获取单板当前的端口映射情况信息
- 11.4 出现端口类似不是从ttyUSB0开始的情况如何处理
- 11.5 如何手动加载串口驱动
- 11.6 若无法映射端口或无法查找对应端口形态需要提供哪些log

11.1 如何确认系统中是否已经存在正确的 USB 串口驱动

打开Terminal，执行命令**dmesg**，查看内核的log信息。如查看到类似如下图内容，即说明设备正常加载。

```
usb 1-10: new high-speed USB device number 4 using xhci_hcd
usb 1-10: config 2 interface 1 altsetting 1 has 2 endpoint descriptors, different from the interface descriptor's value: 3
usb 1-10: New USB device found, idVendor=3466, idProduct=3301, bcdDevice= 1.02
usb 1-10: New USB device strings: Mfr=13, Product=14, SerialNumber=15
usb 1-10: Product: TDTECH MT5700M-CN
usb 1-10: Manufacturer: TD Tech Ltd.
usb 1-10: SerialNumber: 57050M4315800093
cdc_ncm 1-10:2.0: MAC-Address: d0:a0:d6:72:a7:b5
cdc_ncm 1-10:2.0 eth0: register 'cdc_ncm' at usb-0000:00:14.0-10, CDC NCM, d0:a0:d6:72:a7:b5
usbserial_generic 1-10:2.2: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.2: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.2: generic converter detected
usb 1-10: generic converter now attached to ttyUSB0
usbserial_generic 1-10:2.3: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.3: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.3: generic converter detected
usb 1-10: generic converter now attached to ttyUSB1
```

11.2 如何确认系统中是否已经存在正确的 CDC ECM 驱动

打开Terminal，执行命令**dmesg**。如出现下图红框部分信息，则说明存在正确的CDC ECM驱动。

```
usb 1-10: new high-speed USB device number 4 using xhci_hcd
usb 1-10: config 2 interface 1 altsetting 1 has 2 endpoint descriptors, different from the interface descriptor's value: 3
usb 1-10: New USB device found, idVendor=3466, idProduct=3301, bcdDevice= 1.02
usb 1-10: New USB device strings: Mfr=13, Product=14, SerialNumber=15
usb 1-10: Product: TDTECH MT5700M-CN
usb 1-10: Manufacturer: TD Tech Ltd.
usb 1-10: SerialNumber: 57050M4315800093
cdc_ncm 1-10:2.0: MAC-Address: d0:a0:d6:72:a7:b5
cdc_ncm 1-10:2.0 eth0: register 'cdc_ncm' at usb-0000:00:14.0-10, CDC NCM, d0:a0:d6:72:a7:b5
usbserial_generic 1-10:2.2: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.2: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.2: generic converter detected
usb 1-10: generic converter now attached to ttyUSB0
```

11.3 如何获取单板当前的端口映射情况信息

步骤1 执行命令dmesg，查看驱动是否加载成功。假如log信息中存在下图中红框部分信息，则说明驱动已经被成功加载（idProduct因产品的不同而不同）。

```
usb 1-10: new high-speed USB device number 4 using xhci_hcd
usb 1-10: config 2 interface 1 altsetting 1 has 2 endpoint descriptors, different from the interface descriptor's value: 3
usb 1-10: New USB device found, idVendor=3466, idProduct=3301, bcdDevice= 1.02
usb 1-10: New USB device strings: Mfr=13, Product=14, SerialNumber=15
usb 1-10: Product: TDTECH MT5700M-CN
usb 1-10: Manufacturer: TD Tech Ltd.
usb 1-10: SerialNumber: 57050M4315800093
cdc_ncm 1-10:2.0: MAC Address: d0:a0:d6:72:a7:b5
cdc_ncm 1-10:2.0 eth0: register 'cdc_ncm' at usb-0000:00:14.0-10, CDC NCM, d0:a0:d6:72:a7:b5
usbserial_generic 1-10:2.2: The "generic" usb-serial driver is only for testing and one-off prototypes.
usbserial_generic 1-10:2.2: Tell linux-usb@vger.kernel.org to add your device to a proper driver.
usbserial_generic 1-10:2.2: generic converter detected
usb 1-10: generic converter now attached to ttyUSB0
```

步骤2 当前模块设备的Modem和PCUI等端口的设备文件名称查询命令：ls -all /dev/ttyUSB*

```
root@tdtech-Produ:/usr/src/linux-5.15.1/drivers/usb/serial# ls -all /dev/ttyUSB*
crwxrwxrwx 1 root dialout 188, 0 2月 14 19:15 /dev/ttyUSB0
crwxrwxrwx 1 root dialout 188, 1 2月 14 19:15 /dev/ttyUSB1
crwxrwxrwx 1 root dialout 188, 2 2月 14 19:15 /dev/ttyUSB2
crwxrwxrwx 1 root dialout 188, 3 2月 14 19:15 /dev/ttyUSB3
crwxrwxrwx 1 root dialout 188, 4 2月 14 19:15 /dev/ttyUSB4
```

步骤3 对MT5700M-CN 5G模块，默认端口的对应关系举例如下：

Application Interface	/dev/ttyUSB0
PCUI	/dev/ttyUSB1
SerialB	/dev/ttyUSB2
SerialC	/dev/ttyUSB3
GPS	/dev/ttyUSB4

具体实际端口与ttyUSBx对应关系，建议至/sys/bus/usb/usbserial-generic/XXXXX/interface具体确认。

```
root@tdtech-Produ:/sys/bus/usb/drivers/usbserial_generic# cat 3-4\1.0/interface
TDTECH Connect - Application Interface
root@tdtech-Produ:/sys/bus/usb/drivers/usbserial_generic# cat 3-4\1.1/interface
TDTECH Connect - PC UI Interface
root@tdtech-Produ:/sys/bus/usb/drivers/usbserial_generic# cat 3-4\1.2/interface
TDTECH Connect - SerialB
root@tdtech-Produ:/sys/bus/usb/drivers/usbserial_generic# cat 3-4\1.3/interface
TDTECH Connect - SerialC
root@tdtech-Produ:/sys/bus/usb/drivers/usbserial_generic# cat 3-4\1.4/interface
TDTECH Connect - GPS Interface
```

----结束

11.4 出现端口类似不是从 ttyUSB0 开始的情况如何处理

需确定ttyUSB口的使用情况。在断开模块时，确认系统是否已释放ttyUSB0端口资源。

11.5 如何手动加载串口驱动

- 步骤1** 确认已配置添加编译驱动serial进入内核（参考章节5 USB串口驱动集成）。
- 步骤2** 查询模块的VID和PID（Linux系统上可通过输入lsusb获取，如果不支持该命令，可以通过windows或相关产品文档获得）。若得到模块的VID=3466，PID=3301，则说明MT5700M-CN 5G模块USB串口驱动加载成功。
- 步骤3** 打开Terminal，输入echo "3466 3301" >/sys/bus/usb-serial/drivers/generic/new_id。（这个方法是建立在系统有usbserial模块的基础上，通过下面命令查询“lsmod | grep usbserial*” ,如果有，直接使用命令；如果没有，输入命令“modprobe usbserial”进行模块加载）
- 步骤4** 输入dmesg或者ls -all /dev/ttyUSB*，查看加载是否成功。

----结束

11.6 若无法映射端口或无法查找对应端口形态需要提供哪些log

- 步骤1** 打开Terminal，执行命令dmesg并保存输出结果于dmesg.txt文档。
- 步骤2** 执行命令ls -l /sys/bus/usb/drivers/，保存输出结果于logcat.txt文档，确保此目录下有option选项。
- 步骤3** 执行命令ls -l /sys/bus/usb/drivers/usbserial-generic/**/, 保存输出结果于logcat.txt文档。
- 步骤4** 执行命令cat /sys/bus/usb/drivers/usbserial-generic /**/blInterface*，保存输出结果于logcat.txt文档。
- 步骤5** 如果能执行命令cat /proc/bus/usb/devices，将输出结果保存于logcat.txt文档（如果不能请忽略此命令）。

----结束

12 缩略语

缩略语	英文全名	中文解释
CDC	Communications Device Class	连接设备配置
ECM	Ethernet Networking Control Model	以太网控制模型
LTE	Long Term Evolution	长期演进
OS	Operating system	操作系统
USB	Universal Serial Bus	通用串行总线
NCM	Network Control Model	网络控制模型