



# **Guide to Kernel Driver Integration in Linux for Huawei Modules**

Issue            2.0  
Date             2014-12-12

## **Copyright © Huawei Technologies Co., Ltd. 2014. All rights reserved.**

No part of this manual may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd. and its affiliates ("Huawei").

The product described in this manual may include copyrighted software of Huawei and possible licensors. Customers shall not in any manner reproduce, distribute, modify, decompile, disassemble, decrypt, extract, reverse engineer, lease, assign, or sublicense the said software, unless such restrictions are prohibited by applicable laws or such actions are approved by respective copyright holders.

## **Trademarks and Permissions**



HUAWEI,

HUAWEI, and



are trademarks or registered trademarks of Huawei Technologies Co., Ltd.

Other trademarks, product, service and company names mentioned may be the property of their respective owners.

## **Notice**

Some features of the product and its accessories described herein rely on the software installed, capacities and settings of local network, and therefore may not be activated or may be limited by local network operators or network service providers.

Thus, the descriptions herein may not exactly match the product or its accessories which you purchase.

Huawei reserves the right to change or modify any information or specifications contained in this manual without prior notice and without any liability.

## **DISCLAIMER**

ALL CONTENTS OF THIS MANUAL ARE PROVIDED "AS IS". EXCEPT AS REQUIRED BY APPLICABLE LAWS, NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE MADE IN RELATION TO THE ACCURACY, RELIABILITY OR CONTENTS OF THIS MANUAL.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL HUAWEI BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, OR LOSS OF PROFITS, BUSINESS, REVENUE, DATA, GOODWILL SAVINGS OR ANTICIPATED SAVINGS REGARDLESS OF WHETHER SUCH LOSSES ARE FORSEEABLE OR NOT.

THE MAXIMUM LIABILITY (THIS LIMITATION SHALL NOT APPLY TO LIABILITY FOR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH A LIMITATION) OF HUAWEI ARISING FROM THE USE OF THE PRODUCT DESCRIBED IN THIS MANUAL SHALL BE LIMITED TO THE AMOUNT PAID BY CUSTOMERS FOR THE PURCHASE OF THIS PRODUCT.

## **Import and Export Regulations**

Customers shall comply with all applicable export or import laws and regulations and be responsible to obtain all necessary governmental permits and licenses in order to export, re-export or import the product mentioned in this manual including the software and technical data therein.

## **Privacy Policy**

To better understand how we protect your personal information, please see the privacy policy at <http://consumer.huawei.com/privacy-policy>.



## About This Document

### Revision History

Document Version	Date	Chapter	Change Description
V1.0	2010-11-29		Completed the draft
V1.0.1	2011-05-03		Add the method of modifying the kernel files to support new PIDs and to enable the autosuspend feature
V1.2.1	2011-08-25		Add the method to solve upgrade problem in android system used by MC509
V1.2.3	2011-11-10		Enable zero Packet feature.
V1.2.4	2013-01-31		Update the comment scope
V1.2.6	2013-09-06	2	Updated Table 2-1 Linux kernel driver architecture supporting Huawei modules in Android
		All	Added the description related to the CDC ECM Driver
		All	Added the product scope of LTE
		5	Added chapter Appendix
V1.2.7	2013-11-13	All	Updated the document title
V1.2.8	2013-11-30	4.1.1	Updated supporting declarations for Huawei modules
V1.2.9	2013-12-30	4.3	Added 4.3 Delay Time for Selective Suspend Mode
V2.0	2014-12-12	All	Added the supported kernel system and kernel version
		All	Updated description of USB and CDC ECM driver integration
		All	Added description of CDC MBIM and power management driver integration
		8	Added make menuconfig Configuration



Document Version	Date	Chapter	Change Description
		9	Added FAQ
		10	Added Abbreviations



---

# Contents

---

<b>1 Purpose</b>	<b>7</b>
<b>2 Scope</b>	<b>8</b>
<b>3 Overview</b>	<b>10</b>
<b>4 USB Serial Port Driver Integration</b>	<b>11</b>
4.1 Context	11
4.2 Source Code File Modification	11
4.2.1 Adding Macro Definitions	11
4.2.2 Adding the Zero Packet Mechanism	30
4.3 Compilation Configuration	31
<b>5 CDC ECM Driver Integration</b>	<b>33</b>
5.1 Context	33
5.2 Compilation Configuration	33
<b>6 CDC MBIM Driver Integration</b>	<b>34</b>
6.1 Context	34
6.2 Compilation Configuration	34
<b>7 Power Management Integration</b>	<b>35</b>
7.1 Power Management Integration for the USB Serial Port Driver	35
7.1.1 Source Code File Modification	35
7.1.2 Compilation Configuration	37
7.2 Power Management Integration for the CDC ECM Serial Port Driver	37
7.3 Power Management Integration for the CDC MBIM Serial Port Driver	38
7.4 Delay Time for Selective Suspend Mode	38
<b>8 make menuconfig Configuration</b>	<b>40</b>
8.1 Compilation Configuration Options for USB Serial Port Driver	40
8.2 Configuration Options Related to PPP Dial-up	41
8.3 Compilation Configuration Options for CDC ECM Driver	42
8.4 Compilation Configuration Options for CDC MBIM Driver	44
8.5 Configuration Options for Power Management	46
<b>9 FAQ</b>	<b>48</b>
9.1 How to check whether the correct USB serial port driver exists in the kernel?	48



---

9.2 How to check whether the correct CDC ECM serial port driver exists in the kernel? .....	48
9.3 How to obtain the port mapping information of the board? .....	49
9.4 What can I do if the port number does not start from ttyUSB0? .....	50
9.5 How do I manually load the USB serial port driver? .....	50
9.6 How do I provide required logs if port mapping fails or the corresponding ports are not found? ..	50
<b>10 Abbreviations .....</b>	<b>51</b>



# 1 Purpose

---

This guide instructs the kernel driver integration development for Huawei modules based on Linux operating system (OS), such as Android, Ubuntu and Chrome OS. It is intended for the driver developers of the products based on Linux kernel OS.



# 2 Scope

Item	Description	
Applicable operating systems (OSs)	Linux, Android, and Chrome	
Applicable network standards	WCDMA, CDMA, and LTE	
Inapplicable kernel version	Versions earlier than 2.6.12	Not applicable
Applicable kernel versions	2.6.12–2.6.21	Supports USB serial port driver integration (see chapter 4 "USB Serial Port Driver Integration")
	2.6.22–2.6.31	<ul style="list-style-type: none"><li>• Supports USB serial port driver integration (see chapter 4 "USB Serial Port Driver Integration")</li><li>• Supports CDC ECM driver integration (see chapter 5 "CDC ECM Driver Integration")</li></ul>
	2.6.32–3.8	<ul style="list-style-type: none"><li>• Supports USB serial port driver integration (see chapter 4 "USB Serial Port Driver Integration")</li><li>• Supports CDC ECM driver integration (see chapter 5 "CDC ECM Driver Integration")</li><li>• Supports power management integration (see chapter 7 "Power Management Integration")</li></ul>





Item	Description	
	3.9 or later	<ul style="list-style-type: none"><li>• Supports USB serial port driver integration (see chapter 4 "USB Serial Port Driver Integration")</li><li>• Supports CDC ECM driver integration (see chapter 5 "CDC ECM Driver Integration")</li><li>• Supports CDC MBIM driver integration (see chapter 6 "CDC MBIM Driver Integration")</li><li>• Supports power management integration (see chapter 7 "Power Management Integration")</li></ul>



# 3 Overview

---

This document describes how to modify the Linux kernel codes so customer systems (such as the Android system) can properly load Huawei modules.

Two types of drivers may be required for Huawei modules in Linux.

- Huawei-developed port: If this type of port is used, the general drivers must have adaptation data integrated before being used for Huawei modules. The adapted drivers are named **option**.
- General port: If this type of port (such as the ECM and MBIM ports) is used, the general drivers can be directly used for Huawei modules.

Set the kernel compilation configuration items for all drivers to make sure the drivers can be compiled in kernels.



# 4 USB Serial Port Driver Integration

After the USB serial port driver is integrated in Linux, Huawei modules can provide the most basic functions.

## 4.1 Context

The USB serial port driver involves Huawei-developed ports, including the Modem, PCUI, Diag, GPS, and GPS Control ports.

- Modem port: used for interaction between Linux and a Huawei module for PPP-Modem dial-up connections and data services.
- PCUI port: used for interaction between Linux and a Huawei module for AT command execution.
- Diag port: used to capture logs on a Huawei module.
- GPS and GPS Control ports: used to send GPS commands and obtain GPS NMEA information.

USB serial port driver integration requires modification of the following Linux kernel source code files:

```
linux_src/drivers/usb/serial/option.c
linux_src/drivers/usb/serial/usb_wwan.c
linux_src/include/linux/usb.h
```

## 4.2 Source Code File Modification

### 4.2.1 Adding Macro Definitions

Step 1 (Optional) If the following macro definitions are not contained in the **usb.h** file under **linux\_src/include/linux**, add them, as shown in Figure 4-1 :

```
#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)
```

**Figure 4-1** Adding macros to the usb.h file

```
#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)
```

Step 2 Add the following statements to the ID list of **static const struct usb\_device\_id option\_ids[]** in the **option.c** file under **linux\_src/drivers/usb/serial** to match device data:

 **NOTE**

- Make sure the following statements are added in the **option.c** file. Otherwise, USB port mappings of the module may be abnormal.
- The items in the brackets correspond to the VID, InterfaceClass, InterfaceSubClass, and InterfaceProtocol, respectively.

```
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0xff, 0xff) },
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0x01, 0x01) },
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0x01, 0x02) },
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0x01, 0x03) },
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0x01, 0x04) },
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0x01, 0x05) },
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0x01, 0x06) },
{ USB_VENDOR_AND_INTERFACE_INFO(HUAWEI_VENDOR_ID, 0xff, 0x01, 0x31) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x32) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x33) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x34) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x35) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x36) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x61) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x62) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x63) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x64) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x65) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x66) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x0A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x0B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x0D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x0E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x0F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x3A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x3B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x3D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x3E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x3F) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x6A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x6B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x6D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x6E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x6F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x10) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x12) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x13) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x14) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x15) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x17) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x18) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x19) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x1A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x1B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x1C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x1D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x48) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x49) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x4A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x4B) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x4C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x4D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x72) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x73) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x74) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x75) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x78) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x79) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x7A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x7B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x7C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x01,  
0x7D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x01) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x02) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x03) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x04) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x05) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x06) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x31) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x32) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x33) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x34) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x35) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x36) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x61) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x62) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x63) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x64) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x65) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x66) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x0A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x0B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x0D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x0E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x0F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x3A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x3B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x3D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x3E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x3F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x6A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x6B) },
```





```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x6D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x6E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x6F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x10) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x12) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x13) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x14) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x15) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x17) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x18) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x19) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x1A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x1B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x1C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x1D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x48) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x49) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x4A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x4B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x4C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x4D) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x72) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x73) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x74) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x75) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x78) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x79) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x7A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x7B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x7C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x02,  
0x7D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x01) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x02) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x03) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x04) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x05) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x06) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x31) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x32) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x33) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x34) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x35) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x36) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x61) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x62) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x63) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x64) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x65) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x66) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x0A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x0B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x0D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x0E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x0F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x3A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x3B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x3D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x3E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x3F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x6A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x6B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x6D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x6E) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x6F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x10) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x12) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x13) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x14) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x15) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x17) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x18) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x19) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x1A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x1B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x1C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x1D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x48) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x49) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x4A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x4B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x4C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x4D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x72) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x73) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x74) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x75) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x78) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x79) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x7A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x7B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x7C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x03,  
0x7D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x01) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x02) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x03) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x04) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x05) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x06) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x31) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x32) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x33) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x34) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x35) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x36) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x61) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x62) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x63) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x64) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x65) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x66) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x0A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x0B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x0D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x0E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x0F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x3A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x3B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x3D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x3E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x3F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x6A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x6B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x6D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x6E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x6F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x10) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x12) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x13) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x14) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x15) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x17) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x18) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x19) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x1A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x1B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x1C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x1D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x48) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x49) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x4A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x4B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x4C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x4D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x72) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x73) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x74) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x75) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x78) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x79) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x7A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x7B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x7C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x04,  
0x7D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x01) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x02) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x03) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x04) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x05) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x06) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x31) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x32) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x33) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x34) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x35) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x36) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x61) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x62) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x63) },
```





```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x64) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x65) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x66) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x0A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x0B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x0D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x0E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x0F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x3A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x3B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x3D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x3E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x3F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x6A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x6B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x6D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x6E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x6F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x10) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x12) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x13) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x14) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x15) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x17) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x18) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x19) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x1A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x1B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x1C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x1D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x48) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x49) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x4A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x4B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x4C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x4D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x72) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x73) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x74) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x75) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x78) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x79) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x7A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x7B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x7C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x05,  
0x7D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x01) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x02) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x03) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x04) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x05) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x06) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x31) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x32) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x33) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x34) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x35) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x36) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x61) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x62) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x63) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x64) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x65) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x66) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x0A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x0B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x0D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x0E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x0F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x3A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x3B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x3D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x3E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x3F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x6A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x6B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x6D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x6E) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x6F) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x10) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x12) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x13) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x14) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x15) },
```



```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x17) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x18) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x19) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x1A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x1B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x1C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x1D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x48) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x49) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x4A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x4B) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x4C) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x4D) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x72) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x73) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x74) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x75) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x78) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x79) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x7A) },  
  
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,  
0x7B) },
```

```
{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,
0x7C) },

{ USB_VENDOR_AND_INTERFACE_INFO (HUAWEI_VENDOR_ID, 0xff, 0x06,
0x7D) },
```

## 4.2.2 Adding the Zero Packet Mechanism

As required by the USB protocol, add the mechanism for processing zero packets during transmission to the **usb\_wwan.c** file under **linux\_src/drivers/usb/serial**:

Step 1 Define the values of **bcdUSB** and **vid**, as shown in Figure 4-2 .

Add **#define HW\_bcdUSB 0x0110** to define the value of **bcdUSB**.

Add **#define HUAWEI\_VENDOR\_ID 0x12d1** to define the value of **vid**.

**Figure 4-2** Defining the values of bcdUSB and vid

```
#define N_IN_URB 4
#define N_OUT_URB 4
#define IN_BUFLen 4096
#define OUT_BUFLen 4096
#define HW_bcdUSB 0x0110
#define HUAWEI_VENDOR_ID 0x12d1
```

The details are as follows:

```
#define HW_bcdUSB 0x0110

#define HUAWEI_VENDOR_ID 0x12d1
```

Step 2 Add **struct usb\_host\_endpoint \*ep=NULL;** to the **usb\_wwan\_write** function, as shown in Figure 4-3 .

**Figure 4-3** Adding struct usb\_host\_endpoint \*ep=NULL;

```
int usb_wwan_write(struct tty_struct *tty, struct usb_serial_port *port,
const unsigned char *buf, int count)
{
    struct usb_wwan_port_private *portdata;
    struct usb_wwan_intf_private *intfdata;
    int i;
    int left, todo;
    struct urb *this_urb = NULL; /* spurious */
    struct usb_host_endpoint *ep=NULL;
```

The details are as follows:

```
struct usb_host_endpoint *ep=NULL;
```

Step 3 Add the following statements to the **usb\_wwan\_write** function, as shown in Figure 4-4 :

```
if ((HUAWEI_VENDOR_ID == port->serial->dev->descriptor.idVendor)
&& (HW_bcdUSB != port->serial->dev->descriptor.bcdUSB)) {
    ep = usb_pipe_endpoint(this_urb->dev, this_urb->pipe);
```

```

        if(ep && (0 != this_urb->transfer_buffer_length)
            && (0 == this_urb->transfer_buffer_length %
                ep->desc.wMaxPacketSize)){
            this_urb->transfer_flags |= URB_ZERO_PACKET;
        }
    }
}

```

**Figure 4-4** Adding statements to the sb\_wwan\_write function

```

memcpy(this_urb->transfer_buffer, buf, todo);
this_urb->transfer_buffer_length = todo;

if((HUAWEI_VENDOR_ID == port->serial->dev->descriptor.idVendor)
    && (HW_bcdUSB != port->serial->dev->descriptor.bcdUSB)){
    ep = usb_pipe_endpoint(this_urb->dev, this_urb->pipe);
    if(ep && (0 != this_urb->transfer_buffer_length)
        && (0 == this_urb->transfer_buffer_length % ep->desc.wMaxPacketSize)){
        this_urb->transfer_flags |= URB_ZERO_PACKET;
    }
}

spin_lock_irqsave(&intfdata->susp_lock, flags);
if (intfdata->suspended) {
    usb_anchor_urb(this_urb, &portdata->delayed);
    spin_unlock_irqrestore(&intfdata->susp_lock, flags);
}

```

## 4.3 Compilation Configuration

In the `.config` file under `linux_src`, modify the configuration options related to the USB serial port driver and PPP dial-up connections as follows:

- Configuration options related to the USB serial port driver:

```

CONFIG_USB_SERIAL=y
CONFIG_USB_SERIAL_OPTION=y
CONFIG_USB_SERIAL_WWAN=y

```



### NOTE

For details about how to modify the configuration options related to the USB serial port driver from the `make menuconfig` user interface, see section 8.1 "Compilation Configuration Options for USB Serial Port Driver".

- Configuration options related to PPP dial-up connections:

```

CONFIG_PPP=y
CONFIG_PPP_MULTILINK=y
CONFIG_PPP_FILTER=y
CONFIG_PPP_ASYNC=y
CONFIG_PPP_SYNC_TTY=y
CONFIG_PPP_DEFLATE=y
CONFIG_PPP_BSDCOMP=y

```



**NOTE**

For details about how to modify the configuration options related to the PPP dial-up connections from the make menuconfig user interface, see section 8.2 "Configuration Options Related to PPP Dial-up".



# 5 CDC ECM Driver Integration

---

## 5.1 Context

The CDC ECM driver is a general ECM driver for Huawei modules without the need for code modifications. Only compilation configuration is required. CDC ECM driver integration is applicable to kernel version 2.6.22 or later. After the CDC ECM driver is integrated in Linux, Huawei modules can provide data services.

For kernel versions earlier than 2.6.22, to use data services using PPP-Modem dial-up connections, set the configuration options related to the PPP dial-up connections in Linux (for details, see section 4.3 "Compilation Configuration"). This method also applies to kernel version 2.6.22 or later.

## 5.2 Compilation Configuration

In the `.config` file under `linux_src`, modify the configuration options related to the CDC ECM driver as follows:

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



### NOTE

For details about how to modify the configuration options related to the CDC ECM driver from the make menuconfig user interface, see section 8.3 "Compilation Configuration Options for CDC ECM Driver".

# 6 CDC MBIM Driver Integration

---

## 6.1 Context

The CDC MBIM driver is a general driver for Huawei modules without the need for code modifications. Only compilation configuration is required. CDC MBIM driver integration is applicable to kernel version 3.9 or later.

## 6.2 Compilation Configuration

In the `.config` file under `linux_src`, modify the configuration options related to the CDC MBIM driver as follows:

```
CONFIG_USB_USBNET=y  
CONFIG_NETDEVICES=y  
CONFIG_USB_NET_CDC_MBIM=y
```



### NOTE

For details about how to modify the configuration options related to the CDC MBIM driver from the `make menuconfig` user interface, see section 8.4 "Compilation Configuration Options for CDC MBIM Driver".



# 7 Power Management Integration

---

Power management integration is applicable to kernel version 2.6.32 or later.

## 7.1 Power Management Integration for the USB Serial Port Driver

Linux kernels in version 2.6.32 or later versions have the selective suspend feature for USB serial port drivers. Therefore, to enable this feature and the USB serial port drivers will support the selective suspend feature for power management.

### 7.1.1 Source Code File Modification

Step 1 In the **option.c** file under **linux\_src/drivers/usb/serial**, find the functions to which the structure **attach** (under **option\_1port\_device**) direct.

In different versions, the functions directed are different. In version 2.6.35, the function to which **attach** direct is **usb\_wwan\_startup()**, as shown in Figure 7-1 .

**Figure 7-1** Finding the functions directed

```
static struct usb_serial_driver option_1port_device = {
    .driver = {
        .owner = THIS_MODULE,
        .name = "option1",
    },
    .description = "GSM modem (1-port)",
    .usb_driver = &option_driver,
    .id_table = option_ids,
    .num_ports = 1,
    .probe = option_probe,
    .open = usb_wwan_open,
    .close = usb_wwan_close,
    .dtr_rts = usb_wwan_dtr_rts,
    .write = usb_wwan_write,
    .write_room = usb_wwan_write_room,
    .chars_in_buffer = usb_wwan_chars_in_buffer,
    .set_termios = usb_wwan_set_termios,
    .tiocmget = usb_wwan_tiocmget,
    .tiocmset = usb_wwan_tiocmset,
    .attach = usb_wwan_startup,
    .disconnect = usb_wwan_disconnect,
    .release = usb_wwan_release,
    .read_int_callback = option_instat_callback,
#ifdef CONFIG_PM
    .suspend = usb_wwan_suspend,
    .resume = usb_wwan_resume,
#endif
};
```

Step 2 In the `usb_wwan.c` file under `linux_src/drivers/usb/serial`, define `usb_wwan_startup()`. Then, add the following statements, as shown in Figure 7-2, to enable the **selective suspend** feature: (Before adding `usb_wwan_startup()`, add `#define HUAWEI_VENDOR_ID 0x12d1`.)

```
if (serial->dev->descriptor.idVendor == HUAWEI_VENDOR_ID) {
    if ( 0 != (serial->dev->config->desc.bmAttributes & 0x20)) {
        usb_enable_autosuspend(serial->dev);
    }
}
```

**Figure 7-2** Adding statements to enable the selective suspend feature

```
int usb_wwan_startup(struct usb_serial *serial)
{
    int i, j, err;
    struct usb_serial_port *port;
    struct usb_wwan_port_private *portdata;
    u8 *buffer;

    dbg("%s", __func__);

    if (serial->dev->descriptor.idVendor == HUAWEI_VENDOR_ID) {
        if ( 0 != (serial->dev->config->desc.bmAttributes & 0x20)) {
            usb_enable_autosuspend(serial->dev);
        }
    }
}
```



**NOTE**

The `reset_resume` function is used to reset a suspended USB device. If this function is removed, skip step 3.

- Step 3 In the **option.c** file under **linux\_src/drivers/usb/serial**, add the following statement to the **option\_driver** structure to invoke the **reset\_resume** function, as shown in Figure 7-3 :

```
.reset_resume = usb_serial_resume,
```

**Figure 7-3** Adding the statement to invoke the reset\_resume function

```
static struct usb_driver option_driver = {
    .name = "option",
    .probe = usb_serial_probe,
    .disconnect = usb_serial_disconnect,
#ifdef CONFIG_PM
    .suspend = usb_serial_suspend,
    .resume = usb_serial_resume,
    .reset_resume = usb_serial_resume,
    .supports_autosuspend = 1,
#endif
    .id_table = option_ids,
    .no_dynamic_id = 1,
};
```

## 7.1.2 Compilation Configuration

In the **.config** file under **linux\_src**, modify the configuration options of the Linux kernel as follows:

```
CONFIG_USB_SUPPORT=y
```

```
CONFIG_USB =y
```

```
CONFIG_PM_RUNTIME=y
```

```
CONFIG_USB_SUSPEND=y
```

## 7.2 Power Management Integration for the CDC ECM Serial Port Driver

To enable selective suspend feature, add the contents enclosed in the red rectangle in the following figure into the **usbnet\_probe()** function in the **linux\_src/drivers/net/usb/usbnet.c**. As shown in the following figure, first add the macro of **#define HUAWEI\_VENDOR\_ID 0x12d1**.

```
usb_set_intfdata (udev, dev);

if(xdev->descriptor.idVendor == HUAWEI_VENDOR_ID){
    if( 0 != (xdev->config->desc.bmAttributes & 0x20)){
        usb_enable_autosuspend(xdev);
    }
}

netif_device_attach (net);

if (dev->driver_info->flags & FLAG_LINK_INTR)
    netif_carrier_off(net);

return 0;
```

The details are as follows:

```
if(xdev->descriptor.idVendor == HUAWEI_VENDOR_ID){
    if( 0 != (xdev->config->desc.bmAttributes & 0x20)){
        usb_enable_autosuspend(xdev);
    }
}
```

## 7.3 Power Management Integration for the CDC MBIM Serial Port Driver

To enable selective suspend feature, add the contents enclosed in the red rectangle in the following figure into the **usbnet\_probe(.....)** function in the **linux\_src/drivers/net/usb/usbnet.c**. As shown in the following figure, first add the macro of **#define HUAWEI\_VENDOR\_ID 0x12d1**.

```
usb_set_intfdata (udev, dev);
if(xdev->descriptor.idVendor == HUAWEI_VENDOR_ID){
    if( 0 != (xdev->config->desc.bmAttributes & 0x20)){
        usb_enable_autosuspend(xdev);
    }
}
netif_device_attach (net);
if (dev->driver_info->flags & FLAG_LINK_INTR)
    netif_carrier_off(net);
return 0;
```

The details are as follows:

```
if(xdev->descriptor.idVendor == HUAWEI_VENDOR_ID){
    if( 0 != (xdev->config->desc.bmAttributes & 0x20)){
        usb_enable_autosuspend(xdev);
    }
}
```

## 7.4 Delay Time for Selective Suspend Mode

When the power management is enabled, the delay time for selective suspend is 2s by default. That is to say, if the USB communication between the terminal equipment and module is idle for more than 2s, the USB HOST will make USB module go into suspend automatically. The customer can change the delay time based on actual needs and information in Table 7-1 .

**Table 7-1** Minimum delay time for selective suspend mode

Module Name	Minimum Delay Time for Selective Suspend Mode	Reason
MU736	5s	The GPS data submission cycle will be relatively long for the first time. Setting the minimum delay time as 5s guarantees sufficient time for data submission.

Modifications need be made in the information in the red rectangle shown in Figure 7-4 .

**Figure 7-4** Modifying area

```
static int nousb; /* Disable USB when built into kernel image */
#ifdef ONFIG_USB_SUSPEND
static int usb_autosuspend_delay = 2; /* Default delay value,* in seconds */
module_param_named(autosuspend, usb_autosuspend_delay, int, 0644);
MODULE_PARM_DESC(autosuspend, "default autosuspend delay");
```

The details are as follows:

```
static int usb_autosuspend_delay = 2; /* Default delay value,* in
seconds */
```

Change the value of **usb\_autosuspend\_delay** to a desired delay time.

 **NOTE**

- Make sure the delay time is longer than the GPS data submission cycle; otherwise the product may enter suspend mode before a GPS data submission cycle ends.
- See AT^WPDFR in module's AT Command Interface Specification for the time settings of the GPS data submission cycle. By default, the GPS data submission cycle is 1s, and the minimum delay time for the selective suspend mode is 2s.

# 8 make menuconfig Configuration

## 8.1 Compilation Configuration Options for USB Serial Port Driver

```
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

General setup --->
[*] Enable loadable module support --->
--* Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) ---->
Executable file formats / Emulations --->
--* Networking support --->
Device Drivers --->
Firmware Drivers --->
v(++)

<Select> < Exit > < Help >
```

```
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus ---.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

^(-)
Sonics Silicon Backplane --->
[*] Multifunction device drivers --->
[*] Voltage and Current Regulator Support --->
<M> Multimedia support --->
Graphics support --->
<M> Sound card support --->
[*] HID Devices --->
[*] USB support --->
<*> MMC/SD/SDIO card support --->
<M> Sony MemoryStick card support (EXPERIMENTAL) --->
v(++)

<Select> < Exit > < Help >
```



```
USB support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
*** USB Imaging devices ***
<M> USB Mustek MDC800 Digital Camera support
<M> Microtek X6USB scanner support
*** USB port drivers ***
<M> USS720 parport driver
<Y> USB Serial Converter support --->
*** USB Miscellaneous drivers ***
<M> EMI 6|2m USB Audio interface support
<M> EMI 2|6 USB Audio interface support
<M> ADU devices from Ontrak Control Systems
v(+)
<Select> < Exit > < Help >
```

```
USB Serial Converter support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
[ ] USB Secure Encapsulated Driver - Padded
<M> USB Siemens MPI driver
<M> USB Sierra Wireless Driver
<M> USB Symbol Barcode driver (serial mode)
<M> USB TI 3410/5052 Serial Driver
<M> USB REINER SCT cyberJack pinpad/e-com chipcard reader
<M> USB Xircom / Entegra Single Port Serial Driver
<Y> USB driver for GSM and CDMA modems
<M> USB ZyXEL omni.net LCD Plus Driver
<M> USB Opticon Barcode driver (serial mode)
v(+)
<Select> < Exit > < Help >
```

## 8.2 Configuration Options Related to PPP Dial-up

```
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
General setup --->
[*] Enable loadable module support --->
-- Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
-- Networking support --->
<Y> Device Drivers --->
Firmware Drivers --->
v(+)
<Select> < Exit > < Help >
```

```
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
[*] Multiple devices driver support (RAID and LVM) --->
[*] Fusion MPT device support --->
    IEEE 1394 (FireWire) support --->
<M> I2O device support --->
[*] Macintosh device drivers --->
[*] Network device support --->
[*] ISDN support --->
< > Telephony support --->
    Input device support --->
    Character devices --->
v(+)
<Select> < Exit > < Help >
```

```
Network device support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
[ ] Use MMIO instead of PIO
<M> SysKonnnect FDDI PCI support
<M> PLIP (parallel port) support
[*] PPP (point-to-point protocol) support
    [*] PPP filtering
    <M> PPP support for async serial ports
    <M> PPP support for sync tty ports
    <M> PPP Deflate compression
    [*] PPP BSD-Compress compression
    <M> PPP over ATM
v(+)
<Select> < Exit > < Help >
```

### 8.3 Compilation Configuration Options for CDC ECM Driver

```
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
General setup --->
[*] Enable loadable module support --->
--*-- Enable the block layer --->
    Processor type and features --->
    Power management and ACPI options --->
    Bus options (PCI etc.) --->
    Executable file formats / Emulations --->
--*-- Networking support --->
    [*] Device Drivers --->
    Firmware Drivers --->
v(+)
<Select> < Exit > < Help >
```



```
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
[*] Multiple devices driver support (RAID and LVM) --->
[*] Fusion MPT device support --->
    IEEE 1394 (FireWire) support --->
<M> I2O device support --->
[*] Macintosh device drivers --->
[*] Network device support --->
[*] ISDN support --->
< > Telephony support --->
    Input device support --->
    Character devices --->
v(+)
<Select> < Exit > < Help >
```

```
Network device support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
    WiMAX Wireless Broadband devices --->
[*] USB Network Adapters --->
[ ] PCMCIA network device support --->
[*] Wan interfaces support --->
[*] ATM drivers --->
    *** CAIF transport drivers ***
<M> CAIF TTY transport driver
<*> FDDI driver support
<M> Digital DEFTA/DEFEA/DEFFA adapter support
[ ] Use MMIO instead of PIO
v(+)
<Select> < Exit > < Help >
```

```
USB Network Adapters
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
<M> USB KLSI KL5USB101-based ethernet device support
<M> USB Pegasus/Pegasus-II based ethernet device support
[*] Multi-purpose USB Networking Framework
<M> ASIX AX88xxx Based USB 2.0 Ethernet Adapters
(M) CDC Ethernet support (smart devices such as cable modems)
<M> Davicom DM9601 based USB 1.1 10/100 ethernet devices
<M> SMSC LAN75XX based USB 2.0 gigabit ethernet devices
<M> SMSC LAN95XX based USB 2.0 10/100 ethernet devices
<M> GeneSys GL620USB-A based cables
<M> NetChip 1080 based cables (Laplink, ...)
v(+)
<Select> < Exit > < Help >
```



```
USB Network Adapters
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module <>

<M> USB KLSI KL5USB101-based ethernet device support
<M> USB Pegasus/Pegasus-II based ethernet device support
<*> Multi-purpose USB Networking Framework
<M> ASIX AX88xxx Based USB 2.0 Ethernet Adapters
[*] CDC Ethernet support (smart devices such as cable modems)
<M> Davicom DM9601 based USB 1.1 10/100 ethernet devices
<M> SMSC LAN75XX based USB 2.0 gigabit ethernet devices
<M> SMSC LAN95XX based USB 2.0 10/100 ethernet devices
<M> GeneSys GL620USB-A based cables
<M> NetChip 1080 based cables (Laplink, ...)
v(+)
```

<Select> < Exit > < Help >

## 8.4 Compilation Configuration Options for CDC MBIM Driver

```
Linux/x86 3.9.0 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module <>

^(-)
General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
-* Networking support --->
[*] Device Drivers --->
Firmware Drivers --->
v(+)
```

<Select> < Exit > < Help > < Save > < Load >

```
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module <>

^(-)
<M> Generic Target Core Mod (TCM) and ConfigFS Infrastructure --
[*] Fusion MPT device support --->
IEEE 1394 (FireWire) support --->
<M> I2O device support --->
[*] Macintosh device drivers --->
[*] Network device support --->
Input device support --->
Character devices --->
-* I2C support --->
-* SPI support --->
v(+)
```

<Select> < Exit > < Help > < Save > < Load >



```
Network device support
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
<M> PPP over L2TP
<M> PPP support for async serial ports
<M> PPP support for sync tty ports
<M> SLIP (serial line) support
[*] CSLIP compressed headers
[*] Keepalive and linefill
[*] Six bit SLIP encapsulation
[*] USB Network Adapters --->
[*] Wireless LAN --->
WiMAX Wireless Broadband devices --->
v(+)
<Select> < Exit > < Help > < Save > < Load >
```

```
USB Network Adapters
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
<M> USB CATC NetMate-based Ethernet device support
<M> USB KLSI KL5USB101-based ethernet device support
<M> USB Pegasus/Pegasus-II based ethernet device support
<M> USB RTL8150 based ethernet device support
[*] Multi-purpose USB Networking Framework
<M> ASIX AX88xxx Based USB 2.0 Ethernet Adapters
<M> ASIX AX88179/178A USB 3.0/2.0 to Gigabit Ethernet
{M} CDC Ethernet support (smart devices such as cable modems)
<M> CDC EEM support
-*- CDC NCM support
v(+)
<Select> < Exit > < Help > < Save > < Load >
```

```
USB Network Adapters
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
^(-)
<M> USB Pegasus/Pegasus-II based ethernet device support
<M> USB RTL8150 based ethernet device support
[*] Multi-purpose USB Networking Framework
<M> ASIX AX88xxx Based USB 2.0 Ethernet Adapters
<M> ASIX AX88179/178A USB 3.0/2.0 to Gigabit Ethernet
{M} CDC Ethernet support (smart devices such as cable modems)
<M> CDC EEM support
-*- CDC NCM support
[*] CDC MBIM support
<M> Davicom DM9601 based USB 1.1 10/100 ethernet devices
v(+)
<Select> < Exit > < Help > < Save > < Load >
```



## 8.5 Configuration Options for Power Management

```
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

General setup --->
[*] Enable loadable module support --->
--* Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
--* Networking support --->
Device Drivers --->
Firmware Drivers --->
File systems --->
Kernel hacking --->
Security options --->
--* Cryptographic API --->
[*] Virtualization --->
Library routines --->
---
v(+)
```

**<Select>** < Exit > < Help >

```
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] Power Management support
[*] Power Management Debug Support
[*] Extra PM attributes in sysfs for low-level debugging/testing
[ ] Verbose Power Management debugging
[*] Suspend to RAM and standby
[*] Test suspend/resume and wakealarm during bootup
[*] Hibernation (aka 'suspend to disk')
() Default resume partition
[*] Run-time PM core functionality
[*] ACPI (Advanced Configuration and Power Interface) Support --->
[*] SFI (Simple Firmware Interface) Support --->
<M> APM (Advanced Power Management) BIOS support --->
CPU Frequency scaling --->
--* CPU idle PM support
```

**<Select>** < Exit > < Help >

```
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

General setup --->
[*] Enable loadable module support --->
--* Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
--* Networking support --->
Device Drivers --->
Firmware Drivers --->
File systems --->
Kernel hacking --->
Security options --->
--* Cryptographic API --->
[*] Virtualization --->
Library routines --->
---
v(+)
```

**<Select>** < Exit > < Help >



```
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
v(+)
```

```
[*] SPI support --->
   PPS support --->
-*  GPIO Support --->
{M} Dallas's 1-wire support --->
-*  Power supply class support --->
[*] Hardware Monitoring support --->
-*  Generic Thermal sysfs driver --->
[*] Watchdog Timer Support --->
   Sonics Silicon Backplane --->
[*] Multifunction device drivers --->
[*] Voltage and Current Regulator Support --->
<M> Multimedia support --->
   Graphics support --->
<M> Sound card support --->
[*] HID Devices --->
[*] USB support --->
<*> MMC/SD/SDIO card support --->
v(+)
```

```
<Select> < Exit > < Help >
```

```
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
v(+)
```

```
--- USB support
[*] Support for Host-side USB
[ ] USB verbose debug messages
[*] USB announce new devices
   *** Miscellaneous USB options ***
[ ] USB device filesystem (DEPRECATED)
[*] USB device class-devices (DEPRECATED)
[*] Dynamic USB minor allocation
[*] USB runtime power management (suspend/resume and wakeup)
<*> USB Monitor
<M> Support WUSB Cable Based Association (CBA)
   Enable CBA debug messages
   *** USB Host Controller Drivers ***
<M> Cypress C67x00 HCD support
<*> EHCI HCD (USB 2.0) support
   Root Hub Transaction Translators
<M> OXU210HP HCD support
v(+)
```

```
<Select> < Exit > < Help >
```

# 9

## FAQ

### 9.1 How to check whether the correct USB serial port driver exists in the kernel?

Open **Terminal** and run the command **dmesg** to check the kernel log information. If the following information (or similar information) exists in the kernel log, the correct USB serial port driver has been integrated into the kernel.

```
[1558586.308060] usb 1-1.2: new high-speed USB device number 7 using ehci-pci
[1558586.402563] usb 1-1.2: New USB device found, idVendor=12d1, idProduct=1404
[1558586.402568] usb 1-1.2: New USB device strings: Mfr=3, Product=2, SerialNumber=0
[1558586.402571] usb 1-1.2: Product: HUAWEI MOBILE WCDMA EM770W
[1558586.402574] usb 1-1.2: Manufacturer: HUAWEI Technology
[1558586.404738] option 1-1.2:1.0: GSM modem (1-port) converter detected
[1558586.404916] usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB0
[1558586.405016] option 1-1.2:1.1: GSM modem (1-port) converter detected
[1558586.405168] usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB1
[1558586.405260] option 1-1.2:1.2: GSM modem (1-port) converter detected
[1558586.405389] usb 1-1.2: GSM modem (1-port) converter now attached to ttyUSB2
[1558586.405459] option 1-1.2:1.3: GSM modem (1-port) converter detected
```

### 9.2 How to check whether the correct CDC ECM serial port driver exists in the kernel?

Open **Terminal** and run the command **dmesg** to check the kernel log information. If the information in the red rectangle in the following figure exists in the kernel log, the correct CDC ECM serial port driver has been integrated into the kernel.



```
226.168555] usb 2-1.2: USB disconnect, device number 3
226.170773] cdc_ether 2-1.2:2.0 eth0: unregister 'cdc_ether' usb-0000:00:1d:0
-1.2, CDC Ethernet Device
226.177183] option1 ttyUSB0: GSM modem (1-port) converter now disconnected fr
om ttyUSB0
226.177198] option 2-1.2:2.2: device disconnected
257.419920] usb 2-1.2: new high-speed USB device number 4 using ehci-pci
257.536485] usb 2-1.2: New USB device found, idVendor=12d1, idProduct=1573
257.536489] usb 2-1.2: New USB device strings: Mfr=2, Product=3, SerialNumber
=4
257.536493] usb 2-1.2: Product: HUAWEI Mobile
257.536496] usb 2-1.2: Manufacturer: HUAWEI Technology
257.536498] usb 2-1.2: SerialNumber: 0123456712ABCA17
257.595410] cdc_ether 2-1.2:2.0 eth0: register 'cdc_ether' at usb-0000:00:1d:
-1.2, CDC Ethernet Device, 00:1e:10:1f:00:00
257.608340] option 2-1.2:2.2: GSM modem (1-port) converter detected
257.608735] usb 2-1.2: GSM modem (1-port) converter now attached to ttyUSB0
```

### 9.3 How to obtain the port mapping information of the board?

- Step 1 Run the command **dmesg** to check whether the driver for Huawei modules has been loaded successfully. If the information in the red rectangle in the following figure exists in the kernel log, the correct driver has been loaded successfully. (The value of **idProduct** varies with the actual product.)

```
226.168555] usb 2-1.2: USB disconnect, device number 3
226.170773] cdc_ether 2-1.2:2.0 eth0: unregister 'cdc_ether' usb-0000:00:1d:0
-1.2, CDC Ethernet Device
226.177183] option1 ttyUSB0: GSM modem (1-port) converter now disconnected fr
om ttyUSB0
226.177198] option 2-1.2:2.2: device disconnected
257.419920] usb 2-1.2: new high-speed USB device number 4 using ehci-pci
257.536485] usb 2-1.2: New USB device found, idVendor=12d1, idProduct=1573
257.536489] usb 2-1.2: New USB device strings: Mfr=2, Product=3, SerialNumber
=4
257.536493] usb 2-1.2: Product: HUAWEI Mobile
257.536496] usb 2-1.2: Manufacturer: HUAWEI Technology
257.536498] usb 2-1.2: SerialNumber: 0123456712ABCA17
257.595410] cdc_ether 2-1.2:2.0 eth0: register 'cdc_ether' at usb-0000:00:1d:
-1.2, CDC Ethernet Device, 00:1e:10:1f:00:00
257.608340] option 2-1.2:2.2: GSM modem (1-port) converter detected
257.608735] usb 2-1.2: GSM modem (1-port) converter now attached to ttyUSB0
```

- Step 2 Run the command **ls /dev/ttyUSB\*** to query the device file names of Huawei modules' ports (such as the Modem and PCUI ports)

```
root@localhost:~/linux-3.9# ls /dev/ttyUSB*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 /dev/ttyUSB4 /dev/ttyUSB5
```

For most of Huawei modules, the device file name of the Modem port is **/dev/ttyUSB0**, the device file name of the Diag port is **/dev/ttyUSB1** and the device file name of the PCUI port is **/dev/ttyUSB2**.

## 9.4 What can I do if the port number does not start from ttyUSB0?

Check the ttyUSB port usage. Check whether the ttyUSB port is released when the module is disconnected.

## 9.5 How do I manually load the USB serial port driver?

Confirm that the compilation configuration described in chapter 4 "USB Serial Port Driver Integration" is complete.

Step 1 Query the module VID and PID.

In Linux, run the **lsusb** command to obtain the VID and PID. If this command is not supported, query the VID and PID in Windows or follow the instructions in the module documentation.

If the VID is **12D1** and PID **1573**, the module is identified correctly.

Step 2 Open **Terminal** and run **echo "12d1 1573" >/sys/bus/usb-serial/drivers/option1/new\_id**.

Step 3 Run **dmesg** or **ls /dev/ttyUSB\***.

\* specifies the port number.

## 9.6 How do I provide required logs if port mapping fails or the corresponding ports are not found?

Step 1 Open **Terminal**, run the command **dmesg**, and save the output to the **dmesg.txt** file.

Step 2 Run the command **ls -l /sys/bus/usb/drivers/** and save the output to the **logcat.txt** file. Make sure the **option** folder is available under **/sys/bus/usb/drivers/** where the **logcat.txt** file is placed.

Step 3 Run the command **ls -l /sys/bus/usb/drivers/option/\*\*/** and save the output to the **logcat.txt** file.

Step 4 Run the command **cat /sys/bus/usb/drivers/option/\*\*/bInterface\*** and save the output to the **logcat.txt** file.

Step 5 Run command the **cat /proc/bus/usb/devices** and save the output to the **logcat.txt** file. If the command cannot be executed, skip this step.

# 10 Abbreviations

---

Abbreviations	Full Spelling
CDC	Communications Device Class
CDMA	Code Division Multiple Access
ECM	Ethernet Networking Control Model
GPS	Global Positioning System
LTE	Long Term Evolution
MBIM	Mobile Broadband Interface Model
OS	Operating system
PPP	Point-to-Point Protocol
USB	Universal Serial Bus
WCDMA	Wideband Code Division Multiple Access