

WH-GM5 Linux 驱动适配说明书

文件版本：Ver 1.0.2



目录

WH-GM5 Linux 驱动适配说明书.....	1
1. 前言.....	3
1.1. 适用范围.....	3
1.2. 缩略词.....	3
1.3. 文档概要.....	3
2. WH-GM5 的模块使用前的 USB 模式设定.....	3
2.1. 模式设定.....	3
1、查询系统的设定格式：AT+ZNCARD=?.....	3
2、查询当前的 USB 模式：AT+ZNCARD?.....	3
3、设定模块的 USB 模式：AT+ZNCARD=<mode>.....	4
3. Linux 内核修改.....	4
3.1. 源码添加.....	4
3.2. 修改内核配置.....	4
4. Linux 环境下使用模块.....	5
4.1. 使用步骤.....	5
4.1.1. 使用前说明.....	5
4.1.2. 插入 GM5 4G 模块.....	6
4.2. ttyUSB 排序.....	7
4.3. 操作指令.....	8
4.4. 注意事项.....	9
4.4.1. 禁止 Modem Manager 访问串口.....	9
4.4.2. 使用 udhcpc 命令无法自动分配 IP.....	10

1. 前言

1.1. 适用范围

此文档适用于需要测试 WH-GM5 模块 4G 功能的测试人员、开发人员、FAE 及客户。

1.2. 缩略词

ECM(Ethernet Control Model): USB 虚拟的以太网控制卡, Linux 默认支持。

RNDIS(Remote Network Driver Interface Specification): 远程网络接口规范。微软定义的网卡协议, 自 Windows7 以来默认支持。

VID: Vendor ID。

PID: Product ID。

1.3. 文档概要

此文档描述了 WH-GM5 模块的 4G 功能使用说明, 使用 USB 接口连接 Linux 开发板与 GM5 模块, 按照 AT 指令进行网络注册操作, 实现 4G 上网功能。

2. WH-GM5 的模块使用前的 USB 模式设定

2.1. 模式设定

WH-GM5 模块产品使用 AT 命令进行设置。由于模块产品特殊性, 不支持工作模式动态切换。**所以当用户切换模式后需重启模块才能生效。**

1、查询系统的设定格式: AT+ZNCARD=?

```
[2020-08-21_18:26:22:729]AT+ZNCARD=?  
[2020-08-21_18:26:22:729]+ZNCARD: <0, 1, 3>  
[2020-08-21_18:26:22:729]OK
```

有以下几种模式:

0: only charge;

3: RNDIS and serials;

1: ECM and serials;

2、查询当前的 USB 模式: AT+ZNCARD?

```
[2020-08-21_18:27:08:529]AT+ZNCARD?
[2020-08-21_18:27:08:529]+ZNCARD: 3
[2020-08-21_18:27:08:529]OK
```

3、设定模块的 USB 模式: AT+ZNCARD=<mode>

```
[2020-08-21_18:27:33:729]AT+ZNCARD=3
[2020-08-21_18:27:33:729]OK
```

使用 Linux 系统的底板建议将 GM5 模块设置为 ECM 模式。

3. Linux 内核修改

使用的 linux 版本需高于 3.19.x。

3.1. 源码添加

修改的文件为 drivers/usb/serial/option.c, 在 option_ids 里增加:

```
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d10, 0xff, 0, 0) }, /* Eight Serials */
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d11, 0xff, 0, 0) }, /* netdev and four serials */
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d12, 0xff, 0, 0) }, /* diag device */
```

```
{ USB_DEVICE(TPLINK_VENDOR_ID, TPLINK_PRODUCT_MA180),
  .driver_info = (kernel_ulong_t)&net_intf4_blacklist },
{ USB_DEVICE(TPLINK_VENDOR_ID, 0x9000), /* TP-Link MA260 */
  .driver_info = (kernel_ulong_t)&net_intf4_blacklist },
{ USB_DEVICE(CHANGHONG_VENDOR_ID, CHANGHONG_PRODUCT_CH690) },

{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d10, 0xff, 0, 0) }, /* Eight Serials */
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d11, 0xff, 0, 0) }, /* netdev and four serials */
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d12, 0xff, 0, 0) }, /* diag device */

{ USB_DEVICE_AND_INTERFACE_INFO(0x2001, 0x7d01, 0xff, 0x02, 0x01) }, /* D-Link DWM-156 (variant) */
{ USB_DEVICE_AND_INTERFACE_INFO(0x2001, 0x7d01, 0xff, 0x00, 0x00) }, /* D-Link DWM-156 (variant) */
{ USB_DEVICE_AND_INTERFACE_INFO(0x2001, 0x7d02, 0xff, 0x02, 0x01) },
{ USB_DEVICE_AND_INTERFACE_INFO(0x2001, 0x7d02, 0xff, 0x00, 0x00) },
{ USB_DEVICE_AND_INTERFACE_INFO(0x2001, 0x7d03, 0xff, 0x02, 0x01) },
{ USB_DEVICE_AND_INTERFACE_INFO(0x2001, 0x7d03, 0xff, 0x00, 0x00) },
{ USB_DEVICE_AND_INTERFACE_INFO(0x07d1, 0x3e01, 0xff, 0xff, 0xff) }, /* D-Link DWM-152/C1 */
{ USB_DEVICE_AND_INTERFACE_INFO(0x07d1, 0x3e02, 0xff, 0xff, 0xff) }, /* D-Link DWM-156/C1 */
{ USB_DEVICE_INTERFACE_CLASS(0x2020, 0x4000, 0xff) }, /* OLICARD300 - MT6225 */
{ USB_DEVICE(INOVIA_VENDOR_ID, INOVIA_SEW858) },
{ USB_DEVICE(VIA TELECOM_VENDOR_ID, VIA TELECOM_PRODUCT_CDS7) },
```

3.2. 修改内核配置

1. 添加 USB 串口 GSM 和 CDMA 驱动选项。

```
.config - Linux/x86 3.10.0 Kernel Configuration
> Device Drivers > USB support > USB Serial Converter support
    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 Safe Serial (Encapsulated) Driver
    <> USB Siemens MPI driver
    <> USB Sierra Wireless Driver
    <> USB Symbol Barcode driver (serial mode)
    <> USB TI 3410/5052 Serial Driver
    <> USB REINER SCT cyberJack pinpad/e-com chipcard reader
    <> USB Xircom / Entegra Single Port Serial Driver
    [*] USB driver for GSM and CDMA modems
    <> USB ZyXEL omni.net LCD Plus Driver
    <> USB Opticon Barcode driver (serial mode)

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

http://blog.csdn.net/u011728480
```

2. 启动 USB 网络支持

```
.config - Linux/x86 3.10.0 Kernel Configuration
> Device Drivers > Network device support > USB Network Adapters
    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 <>

    <> USB KLSI KL5USB101-based ethernet device support
    <> USB Pegasus/Pegasus-II based ethernet device support
    <> USB RTL8150 based ethernet device support
    <> Realtek RTL8152 Based USB 2.0 Ethernet Adapters
    [*] Multi-purpose USB Networking Framework
    [*] ASIX AX88xxx Based USB 2.0 Ethernet Adapters
    [*] ASIX AX88179/178A USB 3.0/2.0 to Gigabit Ethernet
    [*] CDC Ethernet support (smart devices such as cable modems)
    <> CDC EEM support
    [*] CDC NCM support

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

http://blog.csdn.net/u011728480
```

3. 保存设置后，重新编译内核并烧录到开发板中。

4. Linux 环境下使用模块

4.1. 使用步骤

4.1.1. 使用前说明

Linux 下 Kernel 自带 USB 串口以及 ECM 驱动，比较新的版本可能还自带 RNDIS 驱动。但是 Linux 使用 USB 网卡建议直接用 ECM 驱动。

Linux 下 ECM 驱动是通过 CLASS ID 来识别的，所以在 Linux 下不需要额外装驱动也不需要其他操作就可以被识别。USB 串口没有标准 CLASS，在 Linux 要根据 PID/VID 去识别。

如果 GM5 驱动编译进内核则忽略下面两步操作，如果将 GM5 驱动编译为外部模块则需要按照以下两步操作：

1. 挂载 option 驱动
\$ /sbin/modprobe option
2. 把 GM5 模块可能用到的模式加入 option 驱动的认识列表
\$ echo 1782 4d10 > /sys/bus/usb-serial/drivers/option1/new_id
\$ echo 1782 4d11 > /sys/bus/usb-serial/drivers/option1/new_id
\$ echo 1782 4d12 > /sys/bus/usb-serial/drivers/option1/new_id
注：1782 4d12 可以不加，这个模式通常只在 Windows 下使用。

4.1.2. 插入 GM5 4G 模块

将模块与开发板通过 USB 接口进行连接，均上电后，驱动会自动进行识别：

```
root@imx6ulevk:~# usb 1-1.1: new high-speed USB device number 4 using ci_hdc
usb 1-1.1: config 1 interface 0 altsetting 0 endpoint 0x81 has an invalid bInterval 32, changing to 9
cdc_ether 1-1.1:1.0 usb0: register 'cdc_ether' at usb-ci_hdc.1-1.1, CDC Ethernet Device, 02:4b:b3:b9:eb:e5
option 1-1.1:1.2: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB0
option 1-1.1:1.3: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB1
option 1-1.1:1.4: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB2
option 1-1.1:1.5: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB3
option 1-1.1:1.6: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB4
option 1-1.1:1.7: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB5
option 1-1.1:1.8: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB6
option 1-1.1:1.9: GSM modem (1-port) converter detected
usb 1-1.1: GSM modem (1-port) converter now attached to ttyUSB7
```

使用\$ lsusb 可以看到此时连接的 USB 设备，VID 为 1782，PID 为 4d11。

```
root@imx6ulevk:~# lsusb
Bus 001 Device 003: ID 0bda:b720
Bus 001 Device 004: ID 1782:4d11
Bus 001 Device 002: ID 0424:2514
Bus 001 Device 001: ID 1d6b:0002
```

使用\$ ifconfig -a，可以看到所有的网络接口，其中 usb0 则是 GM5 模块生成的网络接口。

```

sit0      Link encap:IPv6-in-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb0      Link encap:Ethernet  HWaddr 02:4B:B3:B9:EB:E5
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr AC:5D:5C:C6:C3:08
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

```

注：如果没有出现该接口，可以按照上文介绍的内容更改模块 usb 模式为 ECM。

4.2. ttyUSB 排序

开发板接入模块后就可以看到 `ttyUSB{*}` device。

```

root@imx6ulevk:/# ls /dev/ttyUSB*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 /dev/ttyUSB4 /dev/ttyUSB5 /dev/ttyUSB6 /dev/ttyUSB7

```

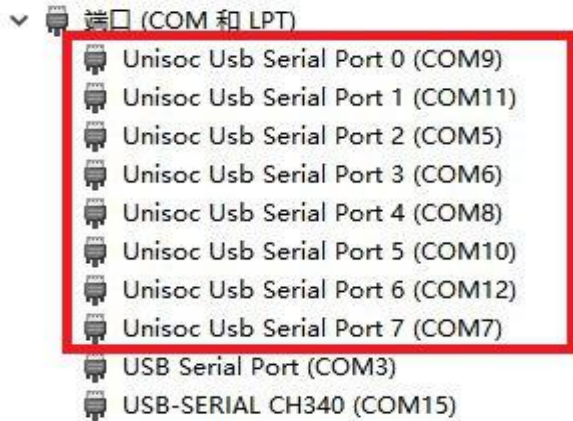
因为 Linux 下 tty 设备不显示设备名称，所以需要使用 `udevinfo` 命令来确认具体设备对应端口。可以使用 `$ udevadm info -n /dev/ttyUSB{0-7}`，来查看 `ttyUSB*` 的信息。

```

root@imx6ulevk:/# udevadm info -n /dev/ttyUSB0
P: /devices/platform/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb1/l-1/l-1.3/l-1.3:1.2/ttyUSB0/tty/ttyUSB0
N: ttyUSB0
S: serial/by-id/usb-UNISOC_UNISOC-8910-if02-port0
S: serial/by-path/platform-ci_hdrc.1-usb-0:1.3:1.2-port0
E: DEVLINKS=/dev/serial/by-id/usb-UNISOC_UNISOC-8910-if02-port0 /dev/serial/by-path/platform-ci_hdrc.1-usb-0:1.3:1.2-port0
E: DEVNAME=/dev/ttyUSB0
E: DEVPATH=/devices/platform/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb1/l-1/l-1.3/l-1.3:1.2/ttyUSB0/tty/ttyUSB0
E: ID_BUS=usb
E: ID_MM_DEVICE_IGNORE=1
E: ID_MODEL=UNISOC-8910
E: ID_MODEL_ENC=UNISOC-8910
E: ID_MODEL_ID=4d11
E: ID_PATH=platform-ci_hdrc.1-usb-0:1.3:1.2
E: ID_PATH_TAG=platform-ci_hdrc.1-usb-0_1_3_1_2
E: ID_REVISION=0000
E: ID_SERIAL=UNISOC_UNISOC-8910
E: ID_TYPE=generic
E: ID_USB_DRIVER=option
E: ID_USB_INTERFACES=:020600:0a0000:ff0000:
E: ID_USB_INTERFACE_NUM=02
E: ID_VENDOR=UNISOC
E: ID_VENDOR_ENC=UNISOC
E: ID_VENDOR_ID=1782
E: MAJOR=188
E: MINOR=0
E: SUBSYSTEM=tty
E: USEC_INITIALIZED=877837575

```

其中的 `ID_USB_INTERFACE_NUM=${id}`：对于 `ID_MODEL_ID=4d10`，id 是 00-07；对于 `ID_MODEL_ID=4d11`，id 是 02-09。从小到大依次对应 Windows 端口下的 Port0 - Port7。



如果想给固定端口指定名称的话可以通过增加 udev rules 来解决。在 /etc/udev/rules.d/ 路径下增加 98-gm5-usb-udev.rules。其内容如下：

```
SUBSYSTEM=="tty",ATTRS{idVendor}=="1782",ATTRS{idProduct}=="4d10",ENV{ID_USB_INTERFACE_NUM}=="00",SYMLINK+="gm5_port0"
SUBSYSTEM=="tty",ATTRS{idVendor}=="1782",ATTRS{idProduct}=="4d11",ENV{ID_USB_INTERFACE_NUM}=="02",SYMLINK+="gm5_port0"
```

注：给串口起别名，这样/dev/ttyUSB0 就会有一个别名为 /dev/gm5_port0。重新上电后可以看到/dev/ttyUSB0 的状态。想要增加其他端口可参考此方法。

```
root@imx6ulevk:~# ls -l /dev/ |grep ttyUSB0
lrwxrwxrwx    1 root    root        7 Apr 26 19:23 gm5_port0 -> ttyUSB0
crw-rw----    1 root    dialout   188,  0 Apr 26 19:23 ttyUSB0
root@imx6ulevk:~#
```

4.3. 操作指令

使用 \$ ifconfig -a 查看网络接口，如果当前网络接口中有 usb0，则按照以下步骤进行操作：

如果模块内部自动注网，则不需要执行下面两条指令；如果模块不会自动进行注网操作，则执行以下两条指令进行注网：

1. \$ echo "AT+CGATT=1" > /dev/gm5_port0 或者 \$ echo "AT+CGATT=1" > /dev/ttyUSB0
2. \$ echo "AT+CGACT=1,1" > /dev/gm5_port0 或者 \$ echo "AT+CGACT=1,1" > /dev/ttyUSB0

注网操作完成后，输入以下指令：

```
$ ifconfig eth0 down
$ ifconfig eth1 down
$ ifconfig wlan0 down
$ ifconfig usb0 up
$ udhcpc -i usb0
```

注：该文件系统需支持 udhcpc 命令。


```
root@imx6ulevk:~# ifconfig eth0 down
root@imx6ulevk:~# ifconfig eth1 down
root@imx6ulevk:~# ifconfig wlan0 down
root@imx6ulevk:~# ifconfig usb0 up
root@imx6ulevk:~# udhcpc -i usb0
udhcpc (v1.24.1) started
Sending discover...
Sending select for 10.165.160.64...
Lease of 10.165.160.64 obtained, lease time 30840
/etc/udhcpc.d/50default: Adding DNS 211.136.17.107
root@imx6ulevk:~#
```

此时的 usb0 网络接口状态如下：

```
root@imx6ulevk:~# ifconfig usb0
usb0      Link encap:Ethernet  HWaddr 02:4B:B3:B9:EB:E5
          inet addr:10.165.160.64  Bcast:10.165.160.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:656 (656.0 B)  TX bytes:6834 (6.6 KiB)
```

使用\$ ping 命令可查看此时的联网状态：

```
root@imx6ulevk:~# ping www.baidu.com
PING www.baidu.com (39.156.66.18): 56 data bytes
64 bytes from 39.156.66.18: seq=0 ttl=53 time=42.867 ms
64 bytes from 39.156.66.18: seq=1 ttl=53 time=50.170 ms
64 bytes from 39.156.66.18: seq=2 ttl=53 time=52.745 ms
64 bytes from 39.156.66.18: seq=3 ttl=53 time=44.194 ms
64 bytes from 39.156.66.18: seq=4 ttl=53 time=42.519 ms
64 bytes from 39.156.66.18: seq=5 ttl=53 time=48.985 ms
64 bytes from 39.156.66.18: seq=6 ttl=53 time=43.489 ms
64 bytes from 39.156.66.18: seq=7 ttl=53 time=46.605 ms
64 bytes from 39.156.66.18: seq=8 ttl=53 time=68.569 ms
64 bytes from 39.156.66.18: seq=9 ttl=53 time=42.917 ms
64 bytes from 39.156.66.18: seq=10 ttl=53 time=61.536 ms
64 bytes from 39.156.66.18: seq=11 ttl=53 time=48.919 ms
64 bytes from 39.156.66.18: seq=12 ttl=53 time=64.410 ms
```

使用\$ route 命令可以查看当前路由情况：

```
root@imx6ulevk:~# route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
default          10.165.160.1    0.0.0.0         UG    10    0      0 usb0
10.165.160.0    *               255.255.255.0   U      0    0      0 usb0
```

4.4. 注意事项

4.4.1. 禁止 Modem Manager 访问串口

通常在 Linux 桌面 PC 上，modem manager 可能会自作主张操作 ttyUSB 设备。解决方法是在 /etc/udev/rules.d/98-gm5-usb-udev.rules 中添加如下内容：

#禁止 Modem Manager 访问串口

```
ATTRS{idVendor}=="1782", ATTRS{idProduct}=="4d10", ENV{ID_MM_DEVICE_IGNORE}="1"
```

```
ATTRS{idVendor}=="1782", ATTRS{idProduct}=="4d11", ENV{ID_MM_DEVICE_IGNORE}="1"
```

```
ATTRS{idVendor}=="1782", ATTRS{idProduct}=="4d12", ENV{ID_MM_DEVICE_IGNORE}="1"
```

4.4.2. 使用 udhcpc 命令无法自动分配 IP

如果使用 `$ udhcpc -i usb0`，出现“Sending discover...”且长时间不能进行动态分配 IP，则需要检查一下 SIM 卡是否正常插上，SIM 卡是否欠费或者是模块是否正常注网。

5. 更新历史

版本	更新内容	更新时间
V1.0.0	版本 V1.0.0 创立	2020-07-21
V1.0.2	更新设置模式方式	2020-08-21