

G403LTE-FAG 说明书

[最大下行速率：150MBPS/上行速率：50MBPS]

[USB 2.0 高速接口]

[增强型 AT 命令集]

日期：2017-04-03

版本：1-1

四川无涯智能科技有限公司

wanyong0523@qq.com | 专业的物联网模块、设备提供者

目录

1. 产品简介.....	2
2. 产品概述.....	3
2.1. 主要优势.....	3
2.2. 主要特征.....	3
2.3. 特殊特性.....	4
2.4. 接口.....	4
2.5. 规格参数.....	4
2.6. 电气参数.....	5
2.7. 尺寸描述.....	6
3. Linux 系统驱动安装指导.....	7
3.1. 添加 USB 串口驱动.....	7
3.1.1 添加 VID 和 PID.....	7
3.1.2 添加 Zero Packet 机制.....	7
3.1.3 添加 Reset Resume.....	9
3.1.4 扩大 URBs.....	10
3.1.5 使用 GobiNet 或者 QMI 驱动.....	10
3.1.6 修改内核配置.....	11
3.2. 添加 GobiNet 驱动.....	12
3.2.1 修改驱动源码.....	12
3.2.2 修改内核配置.....	14
3.3. 添加拨号软件.....	14
3.4. 修改配置文件和测试.....	14
4. 联系方式.....	15

1. 产品简介

G403LTE-FAG 模块是无涯智能最近推出的 LTE 模块。



G403LTE-FAG 模块采用 LTE 3GPP Rel.10 技术，支持最大下行速率 150Mbps 和最大上行速率 50Mbps，可实现从 3G 网络向 4G 网络的轻松平滑过渡。G403LTE-FAG 模块包含 LCC 和 Mini_PCie 两个版本，它能够向后兼容现存的 EDGE 和 GSM/GPRS 网络，以确保在缺乏 3G 和 4G 网络的偏远地区也能正常工作。

G403LTE-FAG 模块支持多输入多输出技术（MIMO），即在发射端和接收端分别使用多个发射天线和接收天线，使信号通过发射端与接收端的多个天线传送和接收，从而降低误码率，改善通信质量。同时，它结合了高速无线连接与内置多星座高精度定位 GPS/GLONASS/BeiDou/Galileo/QZSS 接收器。

G403LTE-FAG 模块内置丰富的网络协议，集成多个工业标准接口，多种操作系统和软件

功能(Windows XP/Windows Vista/Windows7/Windows 8/8.1/Linux/Android)，极大地拓展了G403LTE-FAG模块在M2M领域的应用范围，如中央处理单元、路由器、数据卡、平板电脑、车载、安防以及工业级PDA领域。

2. 产品概述

2.1. 主要优势

- 各网络制式的全面覆盖
- 标准的 Mini PCIe 封装模块，为客户设计及使用提供最大便利
- MIMO 技术满足无线通信系统对数据速率和连接的可靠性要求
- 高集成多星座 GNSS 接收机提供快速、精准的定位
- 轻松实现从 3G 向 4G 网络的平滑过渡

2.2. 主要特征

频段	FDD LTE: B1/B3/B8
	TDD LTE: B38/B39/B40/B41
	TDSCDMA: B34/B39
	WCDMA: B1/B8
	GSM: 900/1800
LTE 版本	3GPP E-UTRA Release 10
带宽	1.4/3/5/10/15/20MHz
天线	DL MIMO, 支持 Rx-diversity
供电电压	3.3V~ 3.6V, 典型值：3.3V
工作温度	-40°C ~ +85°C
尺寸	51.0mm × 30.0mm × 4.9mm
重量	约 10.5g
AT 命令版本	3GPP TS27.007 和增强型 AT 命令

2.3. 特殊特性

驱动	USB Serial	Windows XP, Windows Vista, Windows 7, Windows 8/8.1, Windows 10, Linux 2.6 或更新版本, Android 4.0/4.2/4.4/5.0
	RIL*	Android 4.0/4.2/4.4/5.0
	NIDS	Windows XP, Windows Vista, Windows 7, Windows 8/8.1, Windows 10
	Linux qmi wwan	Linux 3.4 或更新版本
	Gobinet	Linux 2.6 或更新版本
DFOTA	软件空中升级系统	
GNSS	GPS/GLONASS/BeiDou/Galileo/QZSS	

2.4. 接口

USB	USB 2.0 高速接口, 480Mbps
USIM	1.8V/3V
网络指示	×2, NET_STATUS 和 NET_MODE
UART	×1 UART
复位	低电平
PWRKEY	低电平
天线接口	×3 (主天线, 分集天线和 GNSS 天线接口)
ADC	×2

2.5. 规格参数

数据	LTE	LTE-FDD: Max 150Mbps (DL) Max 50Mbps (UL) LTE-TDD: Max 130Mbps (DL) Max 35Mbps (UL)
	DC-HSPA+	Max 42Mbps (DL) Max 5.76Mbps (UL)

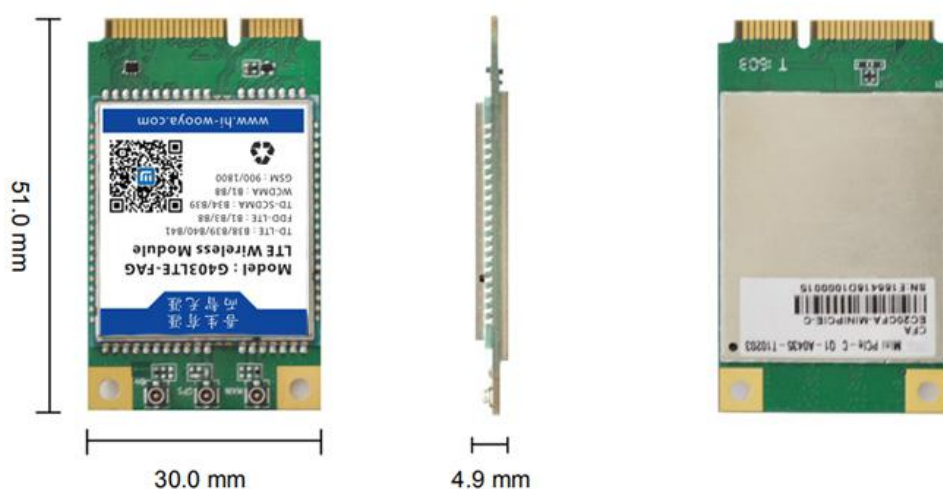
	UMTS	Max 384Kbps (DL) Max 384Kbps (UL)
	TD-SCDMA	Max 4.2Mbps (DL) Max 2.2Mbps (UL)
	CDMA	Max 3.1Mbps (DL) Max 1.8Mbps (UL)
	EDGE	Max 236.8Kbps (DL) Max 236.8Kbps (UL)
	GPRS	Max 85.6Kbps (DL) Max 85.6Kbps (UL)
音频	语音编解码	HR, FR, EFR, AMR, AMR-WB
	回音机制	回声消除 噪声抑制
eCall*	车辆事故紧急服务	
VoLTE	数字语音和 VoLTE (Voice over LTE) (可选)	
协议栈	TCP/ UDP/ PPP/ FTP/ HTTP/ NTP/ PING/ QMI/ NITZ/ CMUX/ HTTPS*/ SMTP*/ MMS*/ FTPS*/ SMTPS*/ SSL*/ FILE*	

2. 6. 电气参数

输出功率	Class 3 (23dBm±2dB) for LTE FDD
	Class 3 (23dBm±2dB) for LTE TDD
	Class 2 (24dBm +1/-3dB) for TD-SCDMA
	Class 3 (24dBm +1/-3dB) for WCDMA
	Class 3 (24dBm±1dB) for CDMA BC0
	Class E2 (27dBm ±3dB) for EDGE 900MHz
	Class E2 (26dBm ±3dB) for EDGE 1800MHz
	Class 4 (33dBm ±2dB) for GSM 900MHz
	Class 1 (30dBm ±2dB) for GSM 1800MHz

功耗	20uA @Power off
	3mA @Sleep, 典型值
	20mA @Idle state, 典型值
灵敏度	FDD B1: -97dBm (10M)
	FDD B3: -95dBm (10M)
	FDD B8: -94dBm (10M)
	TDD B38: -97dBm (10M)
	TDD B39: -97dBm (10M)
	TDD B40: -97dBm (10M)
	TDD B41: -96dBm (10M)
	WCDMA B1: -111dBm
	WCDMA B8: -110dBm
	TDSCDMA B34: -109dBm
	TDSCDMA B39: -109dBm
	CDMA BC0: -109dBm
	GSM 900: -109dBm
	GSM 1800: -109dBm

2.7. 尺寸描述



3. Linux 系统驱动安装指导

本章的目的，是给 G403LTE-FAG 全网通 4G 模块添加驱动并且正常拨号上网。本章是以 ChaosCalmer1505 OpenWrt 源码为例，对应的内核为 linux-3.18.45。当然，如果您是其他版本的 Linux 系统，也可参考本章内容进行修改，大同小异。

3.1. 添加 USB 串口驱动

如果您想给 G403LTE-FAG 全网通 4G 模块添加 USB 串口驱动，请仔细阅读该章节，否则请跳过。

当我们成功的给 G403LTE-FAG 全网通 4G 模块添加了 USB 串口驱动后，我们可以在设备节点中查看到以下设备节点信息。

```
# ls /dev/ttyUSB*  
  
# ttyUSB0/ttyUSB1/ttyUSB2...
```

接下来给大家详细讲解，如何添加 USB 串口驱动。

3.1.1 添加 VID 和 PID

为了能识别到 G403LTE-FAG 全网通 4G 模块，我们需要按照下面的方法添加 VID 和 PID。

文件：[KERNEL]/drivers/usb/serial/option.c

```
static const struct usb_device_id option_ids[] = {  
#if 1 //Added by Wooya  
    { USB_DEVICE(0x05C6, 0x9215) }, /* Wooya G403LTE-FAG */  
#endif
```

如果你的内核代码中存在下列代码，请将它们删除。它们将干扰 G403LTE-FAG 全网通 4G 模块 USB 串口驱动。

文件：[KERNEL]/drivers/usb/serial/qcserial.c

```
{USB_DEVICE(0x05c6, 0x9215)}, /* Acer Gobi 2000 Modem device (VP413) */
```

文件：[KERNEL]/drivers/net/usb/qmi_wwan.c

```
{QMI_GOBI_DEVICE(0x05c6, 0x9215)}, /* Acer Gobi 2000 Modem device (VP413) */
```

3.1.2 添加 Zero Packet 机制

在传输大量数据的时候，USB 协议需要添加处理 Zero Packet 的机制。

对于高于 2.6.34 的内核版本，需要做如下修改。

文件: [KERNEL]/drivers/usb/serial/usb_wwan.c

```
static struct urb *usb_wwan_setup_urb(struct usb_serial_port *port,
                                     int endpoint,
                                     int dir, void *ctx, char *buf, int len,
                                     void (*callback) (struct urb *))
{
    struct usb_serial *serial = port->serial;
    struct urb *urb;

    urb = usb_alloc_urb(0, GFP_KERNEL); /* No ISO */
    if (!urb)
        return NULL;

    usb_fill_bulk_urb(urb, serial->dev,
                      usb_sndbulkpipe(serial->dev, endpoint) | dir,
                      buf, len, callback, ctx);

#ifdef CONFIG_USB_ZERO_PACKET
    //Added by Wooya for Zero Packet
    if (dir == USB_DIR_OUT) {
        struct usb_device_descriptor *desc = &serial->dev->descriptor;
        if (desc->idVendor == cpu_to_le16(0x05C6) && desc->idProduct ==
cpu_to_le16(0x9090))
            urb->transfer_flags |= URB_ZERO_PACKET;
        if (desc->idVendor == cpu_to_le16(0x05C6) && desc->idProduct ==
cpu_to_le16(0x9003))
            urb->transfer_flags |= URB_ZERO_PACKET;
        if (desc->idVendor == cpu_to_le16(0x05C6) && desc->idProduct ==
cpu_to_le16(0x9215))
            urb->transfer_flags |= URB_ZERO_PACKET;
        if (desc->idVendor == cpu_to_le16(0x2C7C))
            urb->transfer_flags |= URB_ZERO_PACKET;
    }
#endif

    return urb;
}
```

对于低于 2.6.34 的内核版本，需要做如下修改。

文件: [KERNEL]/drivers/usb/serial/usb_wwan.c

```

/* Helper functions used by option_setup_urbs */
static struct urb *option_setup_urb(struct usb_serial *serial, int endpoint,
int dir, void *ctx, char *buf, int len,
void (*callback)(struct urb *))
{
.....
usb_fill_bulk_urb(urb, serial->dev,usb_sndbulkpipe(serial->dev, endpoint) | dir,
buf, len, callback, ctx);
#if 1 //Added by Quectel for Zero Packet
if (dir == USB_DIR_OUT) {
struct usb_device_descriptor *desc = &serial->dev->descriptor;
if (desc->idVendor == cpu_to_le16(0x05C6) && desc->idProduct ==
cpu_to_le16(0x9090))
urb->transfer_flags |= URB_ZERO_PACKET;
if (desc->idVendor == cpu_to_le16(0x05C6) && desc->idProduct ==
cpu_to_le16(0x9003))
urb->transfer_flags |= URB_ZERO_PACKET;
if (desc->idVendor == cpu_to_le16(0x05C6) && desc->idProduct ==
cpu_to_le16(0x9215))
urb->transfer_flags |= URB_ZERO_PACKET;
if (desc->idVendor == cpu_to_le16(0x2C7C))
urb->transfer_flags |= URB_ZERO_PACKET;
#endif
return urb;
}

```

3.1.3 添加 Reset Resume

对于一些 USB 控制器，当主控进入休眠或者待机模式时，它将掉电或者重启。而当主控退出休眠或者待机模式时，USB 主控将不能重启 USB 设备，反而，它将执行 Reset Resume，因此我们需要添加如下内容。

对于高于 3.4 的内核版本，需要做如下修改。

文件：[KERNEL]/drivers/usb/serial/option.c

```

static struct usb_serial_driver option_lport_device = {
.....
#ifdef CONFIG_PM
.suspend          = usb_wwan_suspend,
.resume           = usb_wwan_resume,

```

```
#if 1 //Added by Wooya
    .reset_resume      = usb_wwan_resume,
#endif
#endif
};
```

对于低于 3.4 的内核版本，需要做如下修改。

文件：[KERNEL]/drivers/usb/serial/option.c

```
/* Driver structure we register with the USB core */
static struct usb_driver usb_serial_driver = {
    .name = "usbserial",
    .probe = usb_serial_probe,
    .disconnect = usb_serial_disconnect,
    .suspend = usb_serial_suspend,
    .resume = usb_serial_resume,
#if 1 //Added by Wooya
    .reset_resume = usb_serial_resume,
#endif
    .no_dynamic_id = 1,
    .supports_autosuspend = 1,
};
```

3.1.4 扩大 URBs

如果您的内核版本低于 2.6.29，您需要扩大 URBs 来提高传输速度。

文件：[KERNEL]/drivers/usb/serial/option.c

```
#define N_IN_URB 4
#define N_OUT_URB 4 //Wooya 1
#define IN_BUFLEN 4096
#define OUT_BUFLEN 4096 //Wooya 128
```

3.1.5 使用 GobiNet 或者 QMI 驱动

如果您需要使用 GobiNet 或者 QMI 驱动，为了防止系统将 G403LTE-FAG 全网通 4G 模块的接口 4 当作 USB 转串口设备，需要添加以下代码。

对于高于 2.6.30 的内核版本，需要做如下修改。

文件：[KERNEL]/drivers/usb/serial/option.c

```
static int option_probe(struct usb_serial *serial,
```

```
        const struct usb_device_id *id)
{
.....
#ifdef 1    //Added by Wooya
//Wooya G403LTE-FAG's interface 4 can be used as USB Network device
    if (serial->dev->descriptor.idVendor == cpu_to_le16(0x05C6) &&
        serial->dev->descriptor.idProduct == cpu_to_le16(0x9215)
        && serial->interface->cur_altsetting->desc.bInterfaceNumber >= 4)
        return -ENODEV;
#endif

    /* Store device id so we can use it during attach. */
    usb_set_serial_data(serial, (void *)id);

    return 0;
}
```

对于低于 2.6.30 的内核版本，需要做如下修改。

文件：[KERNEL]/drivers/usb/serial/option.c

```
static int option_startup(struct usb_serial *serial)
{
.....
    dbg("%s", __func__);
#ifdef 1    //Added by Wooya
//Wooya G403LTE-FAG's interface 4 can be used as USB Network device
    if (serial->dev->descriptor.idVendor == cpu_to_le16(0x05C6) &&
        serial->dev->descriptor.idProduct == cpu_to_le16(0x9215)
        && serial->interface->cur_altsetting->desc.bInterfaceNumber >= 4)
        return -ENODEV;
#endif
.....
}
```

3.1.6 修改内核配置

经过上面对源码的修改以后，我们还需要进行相关内核的配置，才能真正支持 USB 串口驱动。

总的来说，我们需要将以下内容，配置进内核即可。

```
kmod-usb-serial    kmod-usb-serial-option    kmod-usb-serial-wwan    usb-modeswitch
```

kmod-usb-net

那么怎么进行配置呢？我们以 ChaosCalmer1505 OpenWrt 源码为例。

1). 首先在 ChaosCalmer1505 OpenWrt 源码顶层目录执行 `make menuconfig` 命令进入配置界面。

```
# make menuconfig
```

2). 配置 `kmod-usb-serial`

```
Kernel modules --->
  USB Support --->
<*> kmod-usb-serial..... Support for USB-to-Serial converters
```

3). 配置 `kmod-usb-serial-option`

```
Kernel modules --->
  USB Support --->
<*> kmod-usb-serial-option..... Support for Option HSDPA modems
```

4). 配置 `kmod-usb-serial-wwan`

```
Kernel modules --->
  USB Support --->
-* kmod-usb-serial-wwan..... Support for GSM and CDMA modems
```

5). 配置 `kmod-usb-net`

```
Kernel modules --->
  USB Support --->
<*> kmod-usb-net..... Kernel modules for USB-to-Ethernet convertors
```

6). 配置 `usb-modeswitch`

```
Utilities --->
<*> usb-modeswitch..... USB mode switching utility
```

7). 最后，保存退出配置界面即可

3.2. 添加 GobiNet 驱动

如果您要给 G403LTE-FAG 全网通 4G 模块添加 GobiNet 驱动，请仔细阅读本章节，否则请跳过。

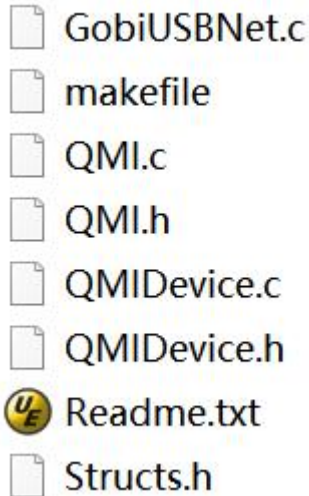
当 GobiNet 驱动识别到 G403LTE-FAG 全网通 4G 模块时，会创建一个网络设备节点和 QMI 通道。这个网络设备节点叫做 `ethX` (如果内核版本低于 2.6.39 则叫 `usbX`)，这个 QMI 通道叫 `/dev/qcqmIX`。这个网络设备节点用于数据传输和 QMI 通道用于 QMI 消息传输。

接下来给大家介绍如何添加 GobiNet 驱动。

3.2.1 修改驱动源码

G403LTE-FAG 全网通 4G 模块添加 GobiNet 驱动源码由我们公司提供。我们需要拷贝这些驱动文件到 “[`KERNEL`]/drivers/net/usb/” ([`KERNEL`]/drivers/usb/net/ 如果内核版本低

于 2.6.22)。



接下来需要修改当前目录下的 Makefile 文件，具体如下。

```
#
# Makefile for USB Network drivers
#

obj-$(CONFIG_USB_CATC) += catc.o
obj-$(CONFIG_USB_KAWETH) += kaweth.o
obj-$(CONFIG_USB_PEGASUS) += pegasus.o
obj-$(CONFIG_USB_RTL8150) += rtl8150.o
obj-$(CONFIG_USB_RTL8152) += rtl8152.o
obj-$(CONFIG_USB_HSO) += hso.o
obj-$(CONFIG_USB_NET_AX8817X) += asix.o
asix-y := asix_devices.o asix_common.o ax88172a.o
obj-$(CONFIG_USB_NET_AX88179_178A) += ax88179_178a.o
obj-$(CONFIG_USB_NET_CDCETHER) += cdc_ether.o
obj-$(CONFIG_USB_NET_CDC_EEM) += cdc_eem.o
obj-$(CONFIG_USB_NET_DM9601) += dm9601.o
obj-$(CONFIG_USB_NET_SR9700) += sr9700.o
obj-$(CONFIG_USB_NET_SR9800) += sr9800.o
obj-$(CONFIG_USB_NET_SMSC75XX) += smsc75xx.o
obj-$(CONFIG_USB_NET_SMSC95XX) += smsc95xx.o
obj-$(CONFIG_USB_NET_GL620A) += gl620a.o
obj-$(CONFIG_USB_NET_NET1080) += net1080.o
obj-$(CONFIG_USB_NET_PLUSB) += plusb.o
obj-$(CONFIG_USB_NET_RNDIS_HOST) += rndis_host.o
obj-$(CONFIG_USB_NET_CDC_SUBSET) += cdc_subset.o
obj-$(CONFIG_USB_NET_ZAURUS) += zaurus.o
obj-$(CONFIG_USB_NET_MCS7830) += mcs7830.o
obj-$(CONFIG_USB_USBNET) += usbnet.o
obj-$(CONFIG_USB_NET_INT51X1) += int51x1.o
obj-$(CONFIG_USB_CDC_PHONET) += cdc-phonet.o
obj-$(CONFIG_USB_NET_KALMIA) += kalmia.o
obj-$(CONFIG_USB_IPHETH) += ipheth.o
obj-$(CONFIG_USB_SIERRA_NET) += sierra_net.o
obj-$(CONFIG_USB_NET_CX82310_ETH) += cx82310_eth.o
obj-$(CONFIG_USB_NET_CDC_NCM) += cdc_ncm.o
obj-$(CONFIG_USB_NET_HUAWEI_CDC_NCM) += huawei_cdc_ncm.o
obj-$(CONFIG_USB_VL600) += lg-vl600.o
obj-$(CONFIG_USB_NET_QMI_WWAN) += qmi_wwan.o
obj-$(CONFIG_USB_NET_CDC_MBIM) += cdc_mbim.o

obj-y += GobiNet.o
GobiNet-objs := GobiUSBNet.o QMIDevice.o QMI.o
```

3.2.2 修改内核配置

经过上面对源码的修改以后，我们还需要进行相关内核的配置，才能真正的使能 GobiNet 驱动。

那么怎么进行配置呢？我们以 ChaosCalmer1505 OpenWrt 源码为例。

1). 首先在 ChaosCalmer1505 OpenWrt 源码顶层目录执行 `make kernel_menuconfig` 命令进入配置界面。

```
# make kernel_menuconfig
```

2). 接下来将下面 2 个选项配置进内核，然后保存退出即可。

```
Device Drivers --->
  [*] USB support --->
    <*> Support for Host-side USB

Device Drivers --->
  [*] Network device support --->
    <*> USB Network Adapters --->
      <*> Multi-purpose USB Networking Framework
```

3.3. 添加拨号软件

G403LTE-FAG 全网通 4G 模块 GobiNet 驱动对应的拨号软件由我们公司提供(对应压缩包为 `quectel_cm.zip`)。将该压缩包拷贝到 `package` 目录，然后进行解压和配置。

1). 解压

```
# tar xvf quectel_cm.zip
```

2). 通过 `make menuconfig` 将拨号软件配置进系统

```
# make menuconfig
```

```
Base system --->
<*> quectel_cm..... quectel_cm communication utility
```

3.4. 修改配置文件和测试

4. 联系方式

公 司：四川无涯智能科技有限公司

网 址：<http://www.hi-wooya.com>

客户支持中心：<http://forum.hi-wooya.com/>

邮 箱：wanyong0523@qq.com

电 话：18576787029 QQ：317312379

公司文化：吾生有涯，而智无涯