

目 录

产品介绍

入门指南

 订购型号

 资源下载

 硬件资源

 软件资源

 机械参数

 电源接口

 调试串口

 按键功能

 串口

 USB

 4G和5G网络功能

 LED指示灯

 MicroSD卡功能

 以太网

 文件传输

 网络优先级设置

 配置系统时间和时区

 NTP时间同步

 烧写系统

 看门狗功能

应用开发

 配置应用开发环境

 开机自启动应用

 修改ssh服务的端口

 切换ssh服务

 配置静态路由

产品介绍

GW201 是海兔核推出的一款工业领域轻量级的网关类产品，本网关采用高性能的工业级ARM应用处理器，搭载嵌入式Linux操作系统作为业务应用软件的支撑平台。同时提供满足工业领域常用的功能接口，RS232、RS485、以太网接口以及无线通讯功能，可同时连接串口设备、以太网设备和无线通讯设备，实现数据的处理或传输。用户可实现数据采集、多种数据协议的格式转换、数据处理转发至无线网络、虚拟专网、本地存储等功能于一体。本网关可以连接多个厂家的物联网平台 SCADA 系统，可以通过 RS232/RS485/以太网口连接智能仪表、PLC 等智能设备，其主要功能 是处理上行、下行的信息，完成不同协议的转发、通过 MQTT 方式连接物联网平台。

产品特点

系统性能强悍

采用低功耗、高性能的工业级温宽嵌入式计算模块，主频最大可达 1000Mhz

适配嵌入式Linux操作系统，可扩展性强，方便开发者实现多样功能

提供SDK开发包，轻松调用系统各种接口及资源

可以接入第三方云平台、如阿里云（LinkWan平台、边缘计算平台、智能网关平台）、百度天工等云平台

支持主流语言的开发和运行环境，Python、Golang、Rust

支持 iptables 防火墙，访问控制

支持 NTP 网络授时

支持 VPN Client和Server（PPTP，L2TP，IPSEC）

支持 Web 方式管理和配置网关

支持安全加密芯片（选配功能）

外设功能易用

支持导轨式和壁挂式安装方式，安装灵活方便

支持 Web 方式管理和配置网关接口功能

模块化设计，方便安装不同类别的4G或LoRa通讯模块

提供接线端子的 RS232、RS485信号，方便现场施工接线安装

支持 10/100 Mbps 以太网接口

多种无线通讯功能，WiFi、4G、LoRa节点/LoRa网关/Zigbee

安装方式支持导轨和壁挂

设备稳定可靠

产品介绍

主要器件均采用一线大厂品牌，提升稳定性

支持 DC 9~24V 宽电压输入，具备防反接保护、过流保护

通讯接口均采用电气隔离设计，能有效抑制干扰，保护电路不受危险电压和电路危害

工业级温宽设计，可在严苛的环境中稳定运行

双网口设计，支持内网和外网隔离，保证内网安全

WDT 看门狗设计，加固系统稳定不宕机

具备升级功能

应用场景

- 环保
 - 智能水务
 - 环境监测
- 智慧社区（园区）
 - 供水监测
 - 烟感监测
 - 路灯控制

水表/电表/燃气表监测

- 智慧电力
 - 通讯规约转换
 - 光伏站逆变监测
 - 输变电监测
- 农业种植
 - 智能灌溉
 - 土壤墒情
 - 水培养殖监测
 - 大棚温湿度检测

产品介绍

- 智能风机
- 虫情监测
- 农业养殖
 - 智能养殖/畜牧监测
 - 畜牧动物定位监测
 - 智能放牧管理
 - 智能饲养
 - 水产养殖环境监测

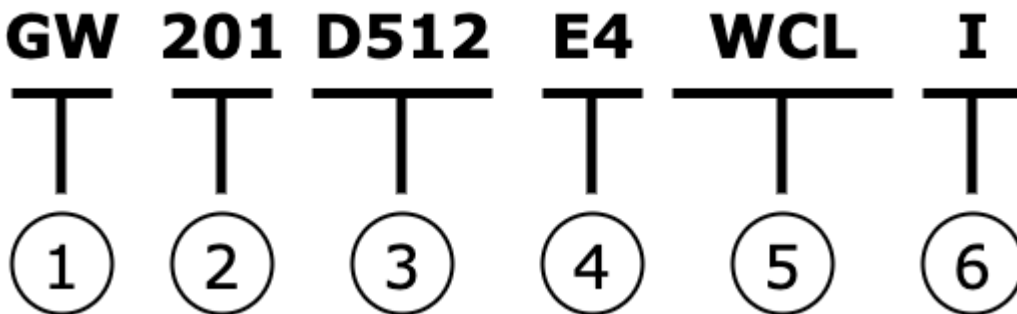
入门指南

入门指南

订购型号

规格型号

湃兔核提供以下标准产品供用户选择使用，如对CPU型号、存储器大小等存在特殊要求，需要选配LoRa，4G模块等请联系湃兔核的销售，或发邮件咨询，support@i2som.com。



项目	描述
产品型号	GW201-D512E4-LI
CPU性能	单核ARM Cortex-A7, 主频最大1000Mhz
内存容量	512MB
硬盘容量	4GB
无线配置	4G Cat1/Cat4
工作温度	-40°~ +85°C
订购链接	

装箱清单

当您开箱时请保管好包装材料，以便日后需要转运时使用。清单如下：

1. GW201网关主机， 1台
2. 4G天线， 1根
3. 电源2Pin接线端子， 1个
4. 串口通讯9Pin 接线端子， 1个

资源下载

资源下载

更新记录

GW201-V0.1

<https://i2somtech.cowtransfer.com/s/b5a639a3bb6b47>

硬件资源

硬件特性

- ST STM32MP13系列高性能工业级处理器，嵌入式计算核心板
 - 单核ARM® Cortex®-A7内核，运行频率最高可达1000 MHz
 - 32KB L1 I-cache, 32KB D-cache
 - 128KB L2 cache
- 运行内存：512MByte DDR3L，高达533MHz的DDR3L时钟
- 数据存储：eMMC Flash 8GB
- 2路10/100/1000M自适应以太网口，标准RJ45接口 15KV TVS保护
- 串行通讯接口
 - 1路 RS232 串口，接线端子，内置ESD保护，全隔离保护设计
 - 2路 RS485 串口，接线端子，内置ESD保护，全隔离保护设计
- 2个LED指示灯分别为“PWR”，“RUN”
- 支持4G网络，采用标准SMA天线接口（内孔外螺纹），卡槽为Micro SIM
- 一路USB Host接口
- 调试口：内置USB转TTL调试口，
- 内置RTC芯片
- 1个自弹TF卡座，支持TF卡扩展存储
- 1个FUN功能按键
- 看门狗：
 - 内置双独立IWDG，支持空闲唤醒和掉电唤醒
 - 外置独立IWDG，带有启动使能
- 电源：
 - 输入电压：9~36VDC，推荐使用12VDC/1.5A

硬件资源

▼ 机械特性: ~ 2000

- 机械特性
 - 外壳金属材质
 - 尺寸: 143mmx88mmx26mm
 - 防护等级: IP63
- 工作环境
 - 工作温度: -40°C~+55°C
 - 工作湿度: 5%~95%

CPU: STM32MP13x ARM 32bit Cotex-A7, 主频1GHz

32KB L1 I-cache, 32KB D-cache

128KB L2 cache

内存保护, 防止非法访问控制;

代码隔离机制, 提供运行时保护;

多种功能, 确保在产品生命周期内平台通过验证;

完整的安全生态系统。

内存: 512Mbyte DDR3L, 高达533MHz的DDR3L时钟

FLASH: eMMC 8Gbyte

加密: 支持PRNG/DES/3DES/AES/SHA/HMAC加密, 最高256位加密模式

看门狗:

内置双独立IWDG, 支持空闲唤醒和掉电唤醒

外置独立IWDG, 带有启动使能

RS232:

1路RS232通讯端口, 内置ESD保护, 全隔离保护设计

RS485: 2路RS485通讯端口, 内置ESD保护, 全隔离保护设计

调试口: 内置USB转TTL调试口, 波特率: 115200, 数据位: 8, 停止位: 1, 校验位: none, 流控: 无

USB- HOST接口

功能按键FUN

网络:

硬件资源

2路10M/100M/1000M自适应工业以太网，标准RJ45接口 15KV TVS保护

无线功能：

具有分集接收功能的 5G NR/LTE-FDD/LTE-TDD/WCDMA无线通信模块

支持 5G NR、LTE-FDD、LTE-TDD、HSDPA、HSUPA、HSPA+、WCDMA 网络数据连接。

2个SIM卡接口，4个天线接口

传输速度：达到相应功能的标准速度

电源：

输入电压：9~36VDC，推荐使用12VDC/1.5 A

单机功耗：< 9W

机械特性

外壳金属材质

尺寸：143 x 88 x 26mm

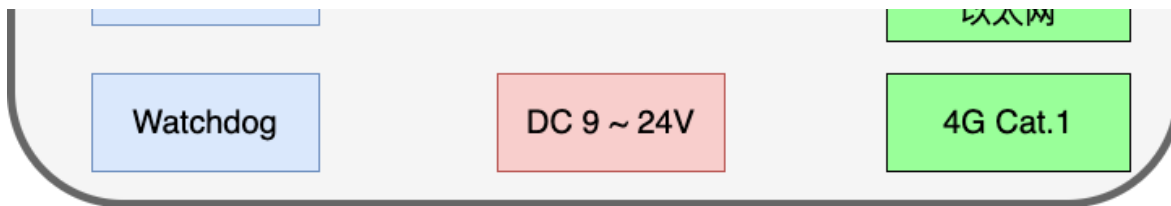
防护等级：IP63

- 工作环境
 - 工作温度：-40℃~+55℃
 - 工作湿度：5% ~ 95%

硬件框图

GW201网关采用了一体化单板方式进行设计，具备丰富的外设接口和内部资源。功能框图如下：

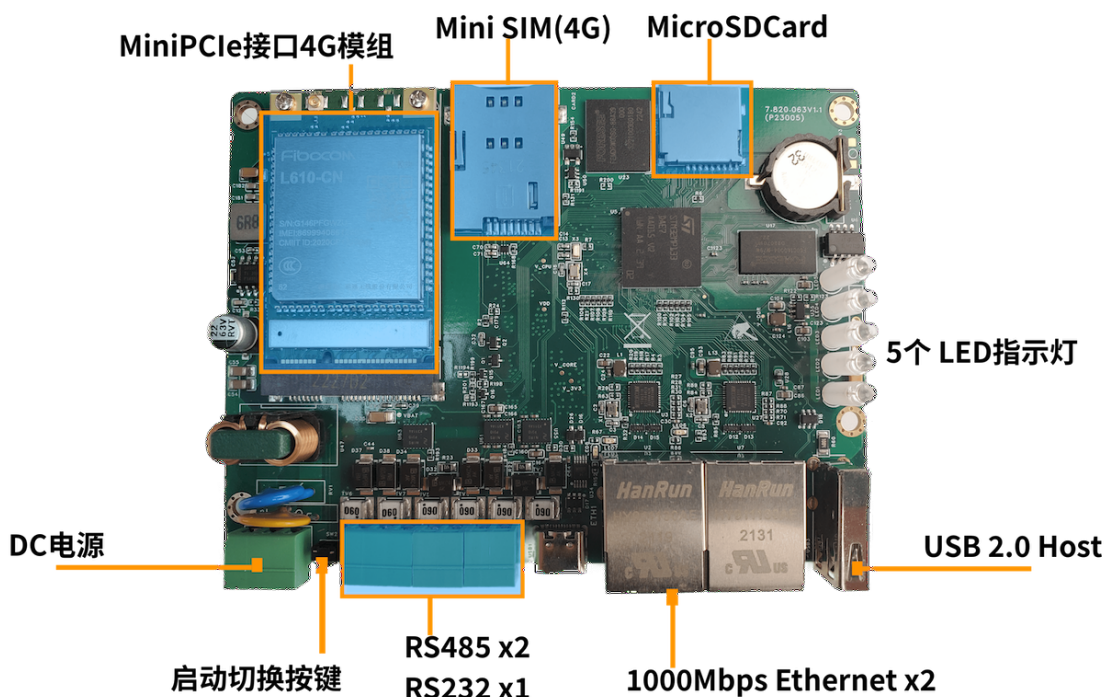




功能框图说明

- 绿色框为通讯接口
- 红色为电源
- 蓝色为板载内部功能

接口功能



接口	说明
电源接口	直流 9 ~ 24V输入
调试串口	1路USB Type-C接口的调试串口，用于Linux系统调试及维护管理功能
指示灯	5个指示灯。一个电源指示灯，一个系统运行状态灯，一个4G网络指示灯
以太网口	双路10/100/1000 Mbps以太网，标准RJ45座
USB 接口	一路USB 2.0 Host Type-A型接口
通讯串口	一路RS232串口，两路RS485串口
4G 天线接口	SMA接口

接口 启动设备按键	说明 可选择系统从不同设备启动，支持eMMC Flash和SDCard
Micro SD卡接口	一个Micro SD卡接口，最高可扩展32GB存储
Mini SIM卡接口	一个Mini SIM卡接口，用于安装移动网络SIM卡

供电参数

- 标准电源: DC 12V/1.5A
- 支持供电范围: DC 9~24V

功耗表现

测试项	平均功耗	说明
系统待机	12V/110mA	未启用4G,未插以太网网线
系统待机, 启动4G模组	12V/150mA	4G连接网络
系统待机, 启动4G模组, 双网口连接网线	12V/265mA	4G连接网络, 两个网口使用iperf3跑满带宽, 总带宽只有1Gbps

物理特性

- 外壳: 金属外壳
- PCB外形尺寸: 116 x 86 x 1.6mm (不包括天线和安装件)
- 外壳外形尺寸: 143 x 88 x 26 mm (不包括天线和安装件)
- 重量: 约722g (未安装天线)
- 工作温度: -40~+85°C
- 相对湿度: 95%(无凝结)

软件资源

GW201网关预装定制的嵌入式Linux操作系统，Linux内核版本为4.14。满足POSIX标准或类UNIX平台的应用程序。针对系统特有的硬件设备或接口，内核提供了简单、易用的驱动接口，可加速用户的应用程序开发。

系统软件结构

GW201网关的Linux系统共分为3部分，分别为Bootloader、Linux内核和RootFS。

- **Bootloader**，用于引导系统启动。本产品选用的是U-Boot，全称 Universal Boot Loader，是遵循GPL条款的开放源代码项目。UBoot不仅能引导Linux内核的启动，同时支持更新升级、Recovery系统的启动
- **Linux内核**是整个操作系统的核心部件，面向最底层，负责整个硬件的驱动，同时提供整个系统所需的核心功能，更好的服务应用运行的条件和需求
- **RootFS**是用于文件或目录结构的存储和管理，也是提供用户应用程序存放的空间位置。对于不同应用功能的需求，都需要RootFS提供相应功能的文件。

应用软件功能

支持Redis 内存KV数据库

支持Thrift RPC通讯框架

支持CAN接口开发库，libsocketcan

支持多种 WAN 连接方式，包括静态 IP,DHCP, PPPOE, 3G/UMTS/4G/LTE, dhcp-4G

支持 4G网络和有线网络的优先级切换功能

支持 VPN client (PPTP, L2TP, IPSEC) (注: 仅 VPN 版支持)

支持 VPN sever (PPTP, L2TP, IPSEC) (注: 仅 VPN 版支持)

支持远程管理, SNMP、SSH、Web管理

支持本地和远程在线升级, 导入导出配置文件

支持 NTP 网络授时, 默认使用chrony软件, 可以作为客户端和服务端

支持国内外多种 DDNS

支持 VLAN, MAC 地址克隆

WIFI 支持 802.11b/g/n/ac,支持 WIFI AP、AP Client, 中继器, 中继桥接等多种工作模式 (可选)

WIFI 支持 WEP,WPA, WPA2 等多种加密方式, MAC 地址过滤

支持多种上下线触发模式, 包括短信、电话振铃、串口数据、网络数据触发上下线模式

软件资源

支持 APN/VPDN

支持多路 DHCP server 及 DHCP client, DHCP 捆绑 MAC 地址, DDNS, 防火墙, NAT, DMZ 主机, QoS, 流量统计,实时显示数据传输速率

支持 TCP/IP、UDP、FTP（可选）、HTTP 等多种网络协议

本地存储支持闪存、TF 卡（TF 卡与 USB 应用二选一）

连接湃兔核云服务，通过 Web 方式访问云平台，实现远程管理

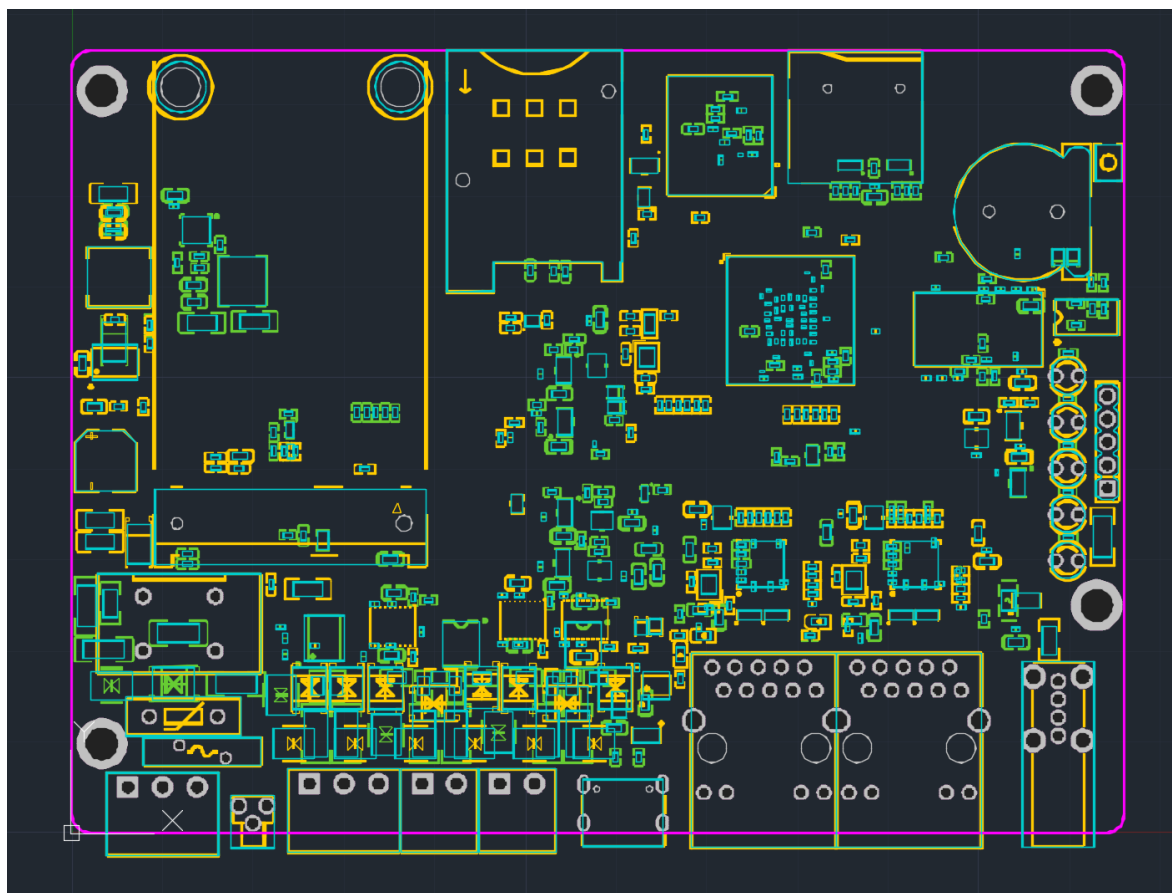
维护通道：维护人员在便携 PC 上运行客户端后，可以直接远程维护现场设备

内置多种通讯协议（Modbus RTU/Modbus TCP、西门子、欧姆龙、三菱等 PLC 私有协议等），适应多种现场设备，通过 MQTT+SSL/TLS 方式连接物联网平台

高性能硬件方案，可以接入第三方平台，比如普奥云、微软 Azure-iot 平台、腾讯云、阿里云（LinkWan 平台、边缘计算平台、智能网关平台）等云平台

机械参数

GW201网关机械尺寸如下图所示：



更详细的尺寸参数请下载以下连接的CAD文件参考

AutoCAD格式：

电源接口

电源接口，输入电压范围DC9~36V，推荐使用DC12V。

PCB丝印为P6，Pin从左往右。实物图如下

FG	V-	V+
屏蔽地、保护地，可不接	电源负极	电源正极

调试串口

调试串口（Console UART）

Console UART口对外接口为Type-C座，内部集成USB转串口芯片，方便用户直接使用串口进行调试与维护。

默认调试串口通信参数如下

- 波特率：115200
- 数据位：8bit
- 停止位：1bit
- 校验方式：None

电脑端可以使用串口工具配置对应参数后，即可与网关建立串口通信。

按键功能

功能按键

实物图如下

- FUN: 可编程按键。按键上电启动SD卡中系统更新文件系统

Linux系统下，此按键可作为常规输入设备。

功能按键的设备节点/dev/input/event0

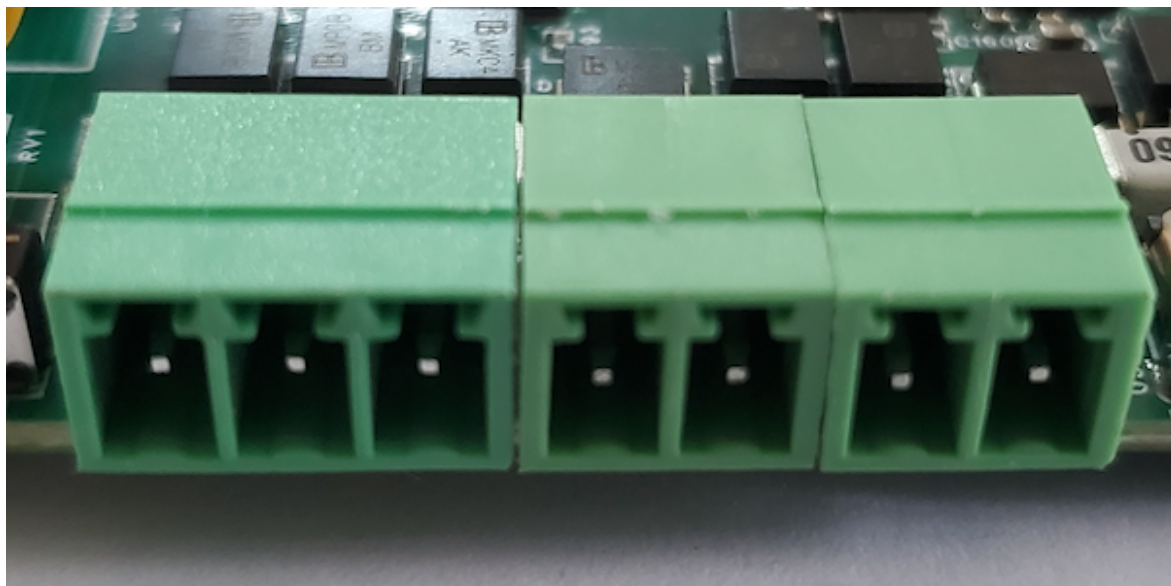
以下是使用evtest命令测试按键，按下和释放功能。

```
evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 257 (BTN_1)
Properties:
Testing ... (interrupt to exit)
Event: time 1651194534.021372, type 1 (EV_KEY), code 257 (BTN_1), value 1
Event: time 1651194534.021372, ----- SYN_REPORT -----
Event: time 1651194534.182318, type 1 (EV_KEY), code 257 (BTN_1), value 0
Event: time 1651194534.182318, ----- SYN_REPORT -----
```

串口

GW201网关有2路RS485串口，1路RS232串口。

实物图如下



左侧3Pin端子为RS232，右侧两组2Pin端子为RS485。右侧两个RS485，从左往右，丝印标识依次是P7和P5。

RS232接口

丝印P4 Linux设备节点 /dev/ttySTM3

RS232的3Pin端子，从左往右顺序的信号表示如下

GND	TX	RX
RS232通讯隔离地	RS232通讯TX端口	RS232通讯RX端口

RS485接口

RS232的3Pin端子，从左往右顺序的信号表示如下

- 丝印P7 Linux设备节点 /dev/ttySTM1
- 丝印P5 Linux设备节点 /dev/ttySTM2

RS485的两个3Pin端子，从左往右顺序的信号表示如下

B	A	B	A
P7的RS485通讯B端口	P7的RS485通讯A端口	P5的RS485通讯B端口	P5RS485通讯A端口

USB

网关支持1路USB HOST功能，连接器为Type-A型。可连接其他USB device设备。

以下使用U盘的方式，插入U盘后，看到提示信息

```
[ 594.650631] usb 1-1: new high-speed USB device number 3 using ehci-platform
[ 594.869170] usb-storage 1-1:1.0: USB Mass Storage device detected
[ 594.883531] scsi host0: usb-storage 1-1:1.0
[ 595.922896] scsi 0:0:0:0: Direct-Access    Generic- SD/MMC          1.00 PQ: 0 ANSI:
4
[ 595.943198] sd 0:0:0:0: Attached scsi generic sg0 type 0
[ 596.813488] sd 0:0:0:0: [sda] 30916608 512-byte logical blocks: (15.8 GB/14.7 GiB)
[ 596.823126] sd 0:0:0:0: [sda] Write Protect is off
[ 596.827492] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 596.850165] sda: sda1
[ 596.865177] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

此时可以挂载存储设备

```
mount /dev/sda1 /mnt
```

卸载存储设备

```
umount /mnt
```

4G和5G网络功能

GW201网关有两个硬件版本

- MiniPCIE版本，默认是Mini PCI-E连接器，可以安装4G Cat1/Cat4模组和5G模组
- 5G版本，默认焊接为移远RG200U 5G模组

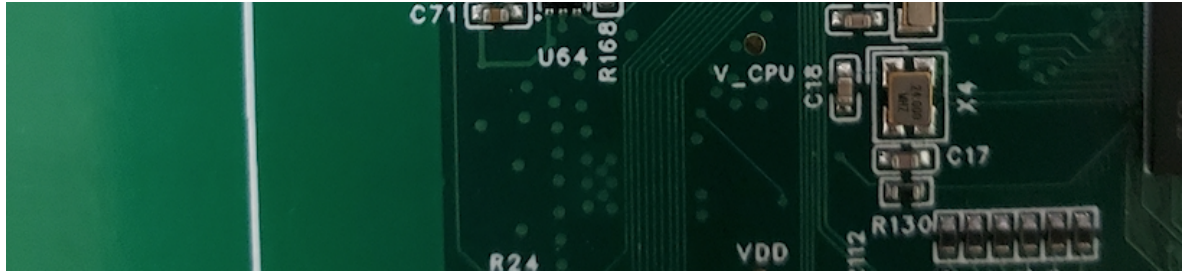
默认出货为MiniPCIE版本，如有定制需求，请联系销售或技术支持。

MiniPCIE版本

GW201 MiniPCIE版本有一个SIM卡座，可以插入Standard SIM卡。外壳上的天线SMA连接在模组上即可。

安装SIM卡时，SIM卡的缺口朝内，芯片朝下来安装，如下图。





MiniPCIe插槽也可以安装5G RedCap模组，目前仅测试了移远RG200U模组。

/dev/ttyUSB2是AT命令端口

使用方法如下

```
// use NCM mode
AT+QCFG="usbnet",5

// use network card mode
AT+QCFG="nat",0

// 关闭 Ethernet 网卡
AT+QCFG="ethernet",0

// 配置 PDP 通路为 2，并进行 PDP 激活操作，立刻生效，同时开启开机自动拨号功能，启用自动重连功能。
AT+QNETDEVCTL=2,3,1

// 查询配置2的网卡状态
AT+QNETDEVSTATUS=2

// 配置完后，需要重启模组
AT+CFUN=1,1
```

GW201 5G版本

GW201 5G版本有两个SIM卡座，板载4个SMA母头接口，外壳的SMA天线座可依次连接上即可。

SIM卡插口

编号	标识符	功能说明
1	SIM1, SIM2	SIM卡接口，支持移动、联通卡、电信

5G模块的天线连接方法

5G天线有四根，4个标准SMA母头接口，用于连接外置天线或者射频馈线。

标识符	功能说明	5G天线标识
ANT0	主天线1	LMH
ANT1	分集天线	MH

ANT2 标识符	主天线2 功能说明	MH 5G天线标识
ANT3	分集天线	LMH

5G速率

5G

5G SA Sub-6 下行2 Gbps; 上行1 Gbps

5G NSA Sub-6 下行2.2 Gbps; 上行575 Mbps

LTE 下行600 Mbps; 上行150 Mbps

5G RedCap

5G SA Sub-6 下行226 Mbps; 上行120 Mbps

LED指示灯

指示灯实物如下



LED灯功能说明 (从上向下的顺序)

- RUN: 系统运行心跳状态
- NET: 无线网络状态指示
- SVR: 用户自定义，系统可编程
- DATA: 用户自定义，系统可编程
- PWR: 电源指示

SVR指示灯操作方法

```
// 点亮  
echo 1 > /sys/class/leds/srv/brightness;  
  
// 熄灭  
echo 0 > /sys/class/leds/srv/brightness
```

DATA指示灯操作方法

```
// 点亮  
echo 1 > /sys/class/leds/data/brightness  
  
// 熄灭  
echo 0 > /sys/class/leds/data/brightness
```

MicroSD卡功能

GW201网关侧面提供了一路Micro SD卡接口，采用标准的MicroSD卡插座，PCB丝印为J6“SD Card”。安Micro SD卡需开发者自行购买，不随本产品提供。Micro SD卡可用于系统调试或者系统升级，当系统启动完成后，用户可以对SD卡进行读写操作，使得用户数据可以灵活选择保存在Micro SD存储卡中。

实物图如下：

Micro SD卡使用方法

插入MicroSD卡，并按压至听到弹簧卡扣的“咔嚓”声，说明卡片已经安装正常。此时系统会检测到卡片，并输出提示信息到串口。

```
mmc0: host does not support reading read-only switch, assuming write-enable
mmc0: new high speed SDHC card at address 59b4
mmcblk0: mmc0:59b4 SMI 7.44 GiB
mmcblk0: p1
```

系统检测到卡片，设备名为mmcblk0，有一个分区p1。同样，使用mount命令查看是否已经挂载。

```
mount | grep mmcblk0
/dev/mmcblk0p1 on /run/media/mmcblk0p1 type vfat (rw,relatime,gid=6,mask=0007,dmask=0007,allow_utime=0020,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
```

这里使用grep命令来过滤，只看mmcblk0设备是否挂载成功。上面的信息说明，mmcblk0p1分区挂载在/run/media/mmcblk0p1目录下，接下来就可以对该目录操作，实现对MicroSD卡的读写。

操作完后，使用umount命令卸载目录。

以太网

接口说明

GW201网关提供两路以太网接口，两个以太网均支持10/100/1000Mbps速率的网口。连接器物理形式为两个单独网口RJ45插座，每个网口均有两个LED指示灯，用来指示以太网的工作情况。指示灯在不接入网线时均不亮，当接入网线后，黄灯亮，表示连接正常，当有数据传输时，绿灯闪烁。

接口实物图如下：

Linux系统接口定义

PCB丝印	Linux系统中设备名
右侧为ETH0	eth0
左侧为ETH1	eth1

网络常用命令

查看系统中的网络设备

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr c2:83:23:40:bf:7b
          inet6 addr: fde3:c0c9:1c0a:0:c083:23ff:fe40:bf7b/64 Scope:Global
          inet6 addr: fe80::c083:23ff:fe40:bf7b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 errors:0 dropped:12 overruns:0 frame:0
          TX packets:47 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2132 (2.0 KiB)  TX bytes:8406 (8.2 KiB)

eth1      Link encap:Ethernet  HWaddr 2e:ec:c8:cc:a2:fb
          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)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  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)
```

配置IP地址

```
# ifconfig eth0 192.168.1.100
```

配置网关

```
# route add default gw 192.168.1.1
```

配置DNS域名解析服务

```
# echo "nameserver 8.8.8.8" > /etc/resolv.conf  
# echo "nameserver 114.114.114.114" >> /etc/resolv.conf
```

配置完成后就可以使用ping命令来测试开发板与互联网的连通性。

```
# ping bing.com  
PING bing.com (204.79.197.200): 56 data bytes  
64 bytes from 204.79.197.200: icmp_seq=0 ttl=118 time=12.205 ms  
64 bytes from 204.79.197.200: icmp_seq=1 ttl=118 time=12.017 ms  
64 bytes from 204.79.197.200: icmp_seq=2 ttl=118 time=11.565 ms  
64 bytes from 204.79.197.200: icmp_seq=3 ttl=118 time=12.116 ms
```

网卡配置持久

GW201网关使用Systemd作为系统服务的管理工具，默认网卡是DHCP模式，自动从网关出获取IP，网关信息。如果需要手动配置网络信息，可以在/etc/systemd/network下编写配置文件。

/etc/systemd/network/eth0.network的静态配置文件内容如下

```
[Match]  
Name=eth0  
  
[Network]  
Address=192.168.5.100/24  
Gateway=192.168.5.50  
DNS=8.8.8.8
```

网口动态配置文件的内容如下

```
[Match]  
Name=eth0  
  
[Network]  
DHCP=yes
```

配置文件写好后，使用systemctl命令重启服务即可。

```
systemctl restart systemd-networkd
```

建议执行reboot命令来重启，而不是按Reset复位按键。

文件传输

开发者需要把文件复制到网关，或者从网关复制文件到电脑，可以使用三种方式。

- 网络文件传输：macOS或Linux系统使用scp命令，Window系统是用winscp工具
- MicroSD卡文件传输：使用MicroSD卡和读卡器
- U盘文件传输：使用U盘存储器

网络文件传输

macOS或Linux的scp命令

```
// 单个文件，从本地到网关
scp hello.txt root@192.168.1.100:~/

// 目录，从本地到网关
scp -r hello root@192.168.1.100:~/

// 单个文件，从网关到本地
scp root@192.168.1.100:~/hello.txt hello.txt

// 目录，从网关到本地
scp -r root@192.168.1.100:~/hello hello
```

MicroSD卡文件传输

SD卡的文件系统格式建议为FAT32或Ext4。FAT32可以在Windows, Linux系统下正常读写，Ext4只能在Linux系统下读写。

插入SD卡到网关的TF卡槽位置，在系统中挂载即可读写使用，使用完后必须先卸载后再拔出。

```
mount -t vfat /dev/mmcblk0p1 /mnt

cp hello.txt /mnt
cp /mnt/hello.txt .

umount
```

U盘文件传输

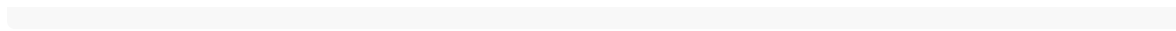
U盘的文件系统格式建议为FAT32或Ext4。FAT32可以在Windows, Linux系统下正常读写，Ext4只能在Linux系统下读写。

插入U盘到网关的USB Host接口上，在系统中挂载即可读写使用，使用完后必须先卸载后再拔出。

```
mount -t vfat /dev/sda1 /mnt

cp hello.txt /mnt
cp /mnt/hello.txt .

umount
```



网络优先级设置

网络设备优先级

DHCP方式的优先级

```
[DHCP]  
RouteMetric = 200
```

静态方式的优先级

```
[Route]  
Gateway = 192.168.1.1  
Metric = 200
```

Metric的数字越小，优先级越高。

比如以太网eth0和4G设备的usb0设备，4G优先配置如下

4G设备是usb0，使用DHCP方式获取IP地址

```
[Match]  
Name=usb0  
  
[Network]  
DHCP=yes  
  
[DHCP]  
RouteMetric=100
```

以太网设备eth0，使用静态IP方式，配置如下

```
[Match]  
Name=eth0  
  
[Network]  
Address=192.168.1.100/24  
  
[Route]  
Gateway=192.168.1.1  
Metric=200
```

配置系统时间和时区

初始系统默认为UTC时区，设置时区为上海

```
timedatectl set-timezone Asia/Shanghai
```

设置时间并存储到RTC中

```
date -s "2023-01-02 12:13:14"  
hwclock -w
```

这样再次上电后，系统会从RTC中读取时间，再设置到系统中。

NTP时间同步

NTP 是网络时间协议（Network Time Protocol），它用来同步网络设备的时间的协议。

配置时间服务器

配置文件为/etc/ntp.conf，使用编辑器工具打开后，修改server字段。

```
# This is the most basic ntp configuration file
# The driftfile must remain in a place specific to this
# machine - it records the machine specific clock error
driftfile /var/lib/ntp/drift
# This should be a server that is close (in IP terms)
# to the machine. Add other servers as required.
# Unless you un-comment the line below ntpd will sync
# only against the local system clock.
#
# server time.server.example.com
#
# Using local hardware clock as fallback
# Disable this when using ntpd -q -g -x as ntpdate or it will sync to itself
server cn.ntp.org.cn
server 127.127.1.0
fudge 127.127.1.0 stratum 14
# Defining a default security setting
restrict -4 default notrap nomodify nopeer noquery
restrict -6 default notrap nomodify nopeer noquery

restrict 127.0.0.1 # allow local host
restrict ::1 # allow local host
```

第14行是增加的时间服务器，可以填写域名或IP地址。

修改后，启动NTP服务

```
systemctl start ntpd
```

启用开机自动启动服务

```
systemctl enable ntpd
```

烧写系统

制作烧写SD卡

使用提供的SD卡烧写镜像文件，使用usbit工具制作烧写SD卡。

USB Image Tool工具使用步骤

1. 运行USB Image Tool.exe程序
2. 插入SD读卡器
3. 左侧点击读卡器
4. 下面点击"Restore"按钮
5. 弹出文件选择对话框，选择"所有文件"，找到对应镜像文件raw.gz，点击"打开"按钮
6. 弹出恢复镜像文件的提示，选择"Yes"按钮
7. 等待左下角进度条结束

烧写步骤

制作烧写SD卡，按如下步骤操作，自动进入烧写系统，并完成系统的更新。

1. 插入GW201网关的MicroSD卡槽
2. 按下按键的同时，连接电源，等两秒后，松开按键

此时系统自动进入烧写系统，并开始自动完成系统的更新。烧写过程中请查看调试串口，是否烧写成功。

烧写成功后，RUN运行灯为常亮。烧写失败为常灭。这里显示update system success表示烧写成功。**烧写成功后，RUN运行灯为常亮。烧写失败为常灭。**

```
[ 19.802588] updatepkg[365]: [INFO ] : SWUPDATE started : Software Update started !
[ 44.827097] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : Configuration file /etc/fw_env.config wrong or corrupted
[ 88.835380] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : Configuration file /etc/fw_env.config wrong or corrupted
[ 88.871050] updatepkg[365]: [INFO ] : SWUPDATE running : Installation in progress
[ 283.303563] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : resize2fs 1.46.5 (30-Dec-2021)
[ 283.341002] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : The filesystem is already 65536 (1k) blocks long. Nothing to do!
[ 283.380846] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : resize2fs 1.46.5 (30-Dec-2021)
[ 283.411079] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : The filesystem is already 16384 (1k) blocks long. Nothing to do!
```

```
[ 283.450763] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : resize2fs 1.46.5 (30-Dec-2021)
[ 323.358697] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : resize2fs 1.46.5 (30-Dec-2021)
[ 385.461968] updatepkg[365]: [ERROR] : SWUPDATE failed [0] ERROR : Configuration file /etc/fw_env.config wrong or corrupted
[ 385.502158] updatepkg[365]: [INFO ] : SWUPDATE successful ! SWUPDATE successful !
[ 385.681990] updatepkg[365]: [INFO ] : No SWUPDATE running : Waiting for requests...
[ 386.069358] updatepkg[284]: update system success!

ST OpenSTLinux - Console - (A Yocto Project Based Distro) 4.0.4-snapshot gw135 ttySTM0
```

看门狗功能

系统内置看门狗功能

默认由systemd来管理

/etc/systemd/system.conf配置文件中开启如下配置

```
RuntimeWatchdogSec=60s  
RebootWatchdogSec=10min
```

RuntimeWatchdogSec定义的是看门狗的超时时间，

ShutdownWatchdogSec定义的是看门狗超时之后的重启系统时间。

测试让系统产生panic错误，导致systemd服务异常而无法喂狗，此时看门狗到达时间后会重启系统

```
echo c > /proc/sysrq-trigger
```

'c' 是预定义的SysRq功能之一，触发Crashdump，产生kernel panic

应用开发

应用开发

配置应用开发环境

GW201网关提供了可以开发应用的SDK工具包，名称为i2SOM-yocto-glibc-x86_64-i2som-image-gateway-cortexa7t2hf-neon-toolchain-2.2-r0.sh。

本节以安装i2SOM-yocto-glibc-x86_64-i2som-image-gateway-cortexa7t2hf-neon-toolchain-2.2-r0.sh工具包为例。

在“资源下载”页面下载工具包后，确认有执行权限，然后安装即可。这里将安装路径设置为“/opt/i2SOM-yocto-adk-core/2.2-r0”。

```
$ chmod a+x i2SOM-yocto-glibc-x86_64-i2som-image-gateway-cortexa7t2hf-neon-toolchain-2.2-r0.sh
$ ls -lh
-rwxr-xr-x 1 i2somdev i2somdev 199M Nov 29 10:20 i2SOM-yocto-glibc-x86_64-core-image-base-cortexa7hf-neon-toolchain-2.2-r0.sh
$ ./i2SOM-yocto-glibc-x86_64-core-image-base-cortexa7hf-neon-toolchain-2.2-r0.sh
i2SOM Yocto SDK installer version 2.2-r0
=====
Enter target directory for SDK (default: /opt/i2SOM-yocto/2.2-r0): /opt/i2SOM-yocto-adk-core/2.2-r0
You are about to install the SDK to "/opt/i2SOM-yocto-adk-core/2.2-r0". Proceed[Y/n]? y
[sudo] password for i2somdev:
Extracting SDK.....done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.
$ . /opt/i2SOM-yocto-adk-core/2.2-r0/environment-setup-cortexa7hf-neon-i2som-linux-gnueabi
```

验证开发工具是否安装正确，显示版本信息就表示工具链安装和加载正确。

```
$ source /opt/i2SOM-yocto-adk-core/2.2-r0/environment-setup-cortexa7hf-neon-i2som-linux-gnueabi
$ $CC --version
arm-i2som-linux-gnueabi-gcc (GCC) 6.2.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

这里使用的是\$CC变量，这个变量是指向gcc工具链的，但使用时请直接使用\$CC，而不是arm-i2som-linux-gnueabi-gcc，直接使用会缺失部分参数配置，导致应用无法编译的情况。每次编译应用前，再加载环境变量文件即可。

测试HelloWorld应用

编写hello.c文件，内容如下：

```
#include <stdio.h>
int main(int argc, char* argv)
{
    printf("Hello, world!");
}
```

编译hello应用

```
$CC hello.c -o hello
```

配置应用开发环境

```
file hello
hello: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, for GNU/Linux 3.2.0, BuildID[sha1]=b6a98060e4d0423c65a477d0a0d74cb183441648, not stripped
```

这里必须要用\$CC变量来编译，不能直接用gcc名称。然后复制在i2C-6ULX开发板后，直接运行就可以了。

编译应用例程

例程包里有很多应用程序，使用cmake管理，编译方式如下，编译前需要先加载sdk工具链。

```
cd linux-example
mkdir build
cd build
cmake ../
make
```

开机自启动应用

GW201网关支持启动自定义的应用程序，系统默认使用systemd service服务方式，也可以使用以前sysv服务的rc-local文件。

systemd service服务方式

系统默认使用systemd管理系统服务，所以添加自启动应用需要编写配置文件

```
# vi /lib/systemd/system/demoapp.service

[unit]
Description=app demo
After=multi-user.target

[Service]
Type=simple
user=root
ExecStart=/usr/local/startapp.sh
#Restart=always

[Install]
WantedBy=multi-user.target
```

编写好后，配置服务启动

```
systemctl enable demoapp.service
```

rc-local方式

系统也支持使用 `/etc/rc.local` 来启动自己的应用程序，将需要执行的命令或应用写在 `/etc/rc.local` 文件中，然后修改 `/lib/systemd/system/rc-local.service` 文件，增加最后两行

```
[Unit]
Description=/etc/rc.local Compatibility
Documentation=man:systemd-rc-local-generator(8)
ConditionFileIsExecutable=/etc/rc.local
After=network.target

[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
RemainAfterExit=yes
GuessMainPID=no

[Install]
WantedBy=multi-user.target
```

然后执行

开机自启动应用

```
systemctl daemon-reload  
systemctl enable rc-local.service
```

修改ssh服务的端口

默认使用sshd.socket提供SSH服务。

sshd socket服务的端口

sshd.socket服务的端口配置文件为 `/lib/systemd/system/sshd.socket`

```
vi /lib/systemd/system/sshd.socket
[Unit]
Conflicts=sshd.service
Wants=sshdgenkeys.service

[Socket]
ExecStartPre=/bin/mkdir -p /var/run/ssh
ListenStream=22
Accept=yes

[Install]
WantedBy=sockets.target
```

修改ListenStream参数为需要的端口号，然后重新启动服务

```
systemctl daemon-reload
systemctl restart sshd.socket
```

sshd service服务的端口

sshd.service服务的配置文件是 `/etc/ssh/sshd_config`

```
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

默认端口为22，修改好重新启动sshd.service就可以了

```
systemctl restart sshd.service
```

切换ssh服务

socket服务与service服务的区别

sshd.service模式会在后台保持一个sshd的守护进程，每当有ssh连接要建立时，就创建一个新进程，比较适合SSH下有大量流量的系统。

sshd.socket方式也是在每次要建立新的ssh连接时生成一个守护进程的实例，不过监听端口则是交给了systemd来完成，意味着没有ssh连接的时候，也不会有sshd守护进程运行，大部分情况下，使用sshd.socket服务更为合适。

另外，通过使用.socket文件来管理需要监听端口的服务，可以直接通过systemctl来查看一些网络相关的信息，如监听的端口、目前已经接受的连接数、目前正连接的连接数等。

切换sshd的socket服务与service服务

如果目前使用的是sshd.socket服务，而想切换至sshd.service服务，可以执行如下命令：

```
systemctl disable sshd.socket  
systemctl enable sshd.service  
systemctl stop sshd.socket  
systemctl start sshd.service
```

如果目前使用的是sshd.service服务，而想切换至sshd.socket服务，可以执行如下命令：

```
systemctl disable sshd.service  
systemctl enable sshd.socket  
systemctl stop sshd.service  
systemctl start sshd.socket
```

配置静态路由

配置子路由表的默认网关

等同于

```
ip route add default via 222.222.222.1 dev eth1 table 222
```

Table 字段表示该子路由表的数字 ID。
network文件中添加如下

```
[Route]  
Table=222  
Gateway=222.222.222.1
```

配置子路由表的本网段路由

等同于:

```
ip route add 222.222.222.0/24 dev eth1 src 222.222.222.222 table 222
```

network文件中添加如下

```
[Route]  
Table=222  
Destination=222.222.222.0/24
```

配置到该子路由表的策略

等同于:

```
echo "200 222" >> /etc/iproute2/rt_tables && \  
# ip rule add from 222.222.222.222 table 222
```

network文件中添加如下

```
[RoutingPolicyRule]  
Table=222  
Priority=200  
From=222.222.222.222
```

配置不生效，需要配置

```
[Match]  
Name=eth0
```

配置静态路由

[Network]

DHCP=yes

[Route]

Gateway=192.168.0.10

Destination=10.0.0.0/8

GatewayOnLink=yes