



LGW/1TG/33G 网关

用户手册

V1.0



上海良信电器股份有限公司

使用须知

在使用本装置前，请仔细阅读本手册并在使用时务必注意以下几点。

注意：

- 本装置必须由已阅读本手册的专业人员进行操作与维护。
- 在对本装置进行任何内部或外部操作前，必须切断所有输入信号和电源，并确保电压互感器的次级端子没有短路，电流互感器的次级端子没有开路。
- 务必使用合适的电压检测装置来确定本装置各部位无电压。
- 提供给本装置的电参数需在额定范围内。
- 当本装置工作时，请勿接触装置端子。
- 使用本装置的通信功能时，请接入安全的通信网络。

下述情况会导致装置损坏或工作异常：

- 工作环境超范围。
- 辅助电源电压超范围。
- 配电系统频率超范围。
- 信号输入超过最大额定值。
- 电流或电压输入极性不正确。
- 未按要求接线。

未经本公司合法的书面许可，不得复制、传播本手册的任何内容。公司对本手册中的任何错误、疏漏而导致或产生的后果概不负责，本手册内容如有变更，恕不另行通知。如果您需要本手册，可以向本公司的技术服务部门。

目录

1	概述.....	7
2	产品选型.....	7
3	行业应用.....	8
4	技术参数表.....	9
4.1	基础性能.....	9
4.2	接口性能.....	10
4.3	可信平台模块.....	10
4.4	机械特性.....	10
4.5	设备功率.....	10
4.6	电源.....	11
4.7	环境温湿度.....	11
4.8	指示灯.....	11
4.9	EMC 指标.....	11
4.10	物理特性.....	11
4.11	认证.....	12
4.12	操作系统.....	12
4.13	网络互联.....	12
4.14	安全.....	12
4.15	可靠性.....	12
4.16	数据采集协议（*DSA）.....	12
4.17	维护管理.....	13
5	产品结构及接线.....	14
5.1	开关量输入接口（Digital Input）.....	15
5.2	开关量输出接口（Digital Output）.....	15
5.3	USB.....	15
5.4	LED.....	16
5.5	用户可编程按键.....	17
5.6	直流输入.....	18
5.7	SIM 卡插槽.....	18
5.8	MicroSD 卡槽.....	18

5.9	恢复出厂按键.....	18
5.10	天线接口.....	19
5.11	mSATA 硬盘接口	19
6	入门.....	20
6.1	连接到 LGW/1TG/33G.....	20
6.1.1	通过 SSH Console 进行连接.....	20
6.1.2	Linux 用户	21
6.1.3	Windows 用户	22
6.2	用户账户管理.....	23
6.2.1	切换到 root 用户	23
6.2.2	创建和删除用户账户	23
6.2.3	禁用默认的用户账户	24
6.3	网络管理和系统管理.....	24
6.3.1	基于 IEOS 的 web 管理	24
6.3.2	登录 web.....	25
6.3.3	网络管理.....	25
6.3.4	配置以太网接口.....	25
6.3.5	配置蜂窝拨号.....	27
6.3.6	配置 Wi-Fi Station.....	30
6.3.7	配置静态路由.....	30
6.3.8	配置防火墙.....	31
6.3.9	配置 DNS.....	31
6.3.10	网络诊断	32
6.3.11	配置 OpenVPN 服务器	33
6.3.12	配置 OpenVPN 客户端	36
6.4	系统管理.....	38
6.4.1	基础配置.....	38
6.4.2	固件升级.....	40
6.4.3	其他.....	40
6.5	状态.....	41
6.5.1	设备信息.....	41
6.5.2	蜂窝拨号状态信息.....	41
6.5.3	Wi-Fi Station 状态信息.....	42
6.5.4	DHCP Server 状态信息	42

6.5.5	路由状态信息.....	42
6.5.6	防火墙状态信息.....	43
6.5.7	日志信息.....	43
6.5.8	基于 Linux 的命令行管理	44
6.6	网络管理.....	45
6.6.1	设置一个静态 IP 地址.....	45
6.6.2	设置一个动态 IP 地址.....	46
6.7	系统管理.....	46
6.7.1	查询固件版本.....	46
6.7.2	查看可用磁盘空间.....	47
6.7.3	查询产品型号信息.....	47
6.7.4	调整时间.....	48
6.7.5	设置时区.....	48
6.7.6	使用 tzselect 命令	48
6.7.7	使用 localtime 文件.....	49
7	外设接口的高级配置.....	50
7.1	串口.....	50
7.1.1	更改串口设置.....	50
7.1.2	查看串口信息.....	53
7.1.3	设置 COM1 串口的波特率.....	53
7.1.4	设置 COM2 串口的波特率.....	53
7.1.5	设置 COM3 串口的波特率.....	53
7.1.6	设置 COM4 串口的波特率.....	54
7.2	USB 接口	54
7.3	Micro SD.....	54
7.4	mSATA 硬盘	56
7.5	CAN 总线接口.....	57
7.5.1	配置连接 CAN1 接口.....	57
7.5.2	配置连接 CAN2 接口.....	57
7.6	IO 调试	58
7.7	GPS	58
7.8	开关机.....	59
7.8.1	关闭设备.....	59
8	安全.....	60

8.1	sudo 机制.....	60
8.2	防火墙.....	60
8.3	TPM2.0.....	61
9	系统恢复出厂设置及更新.....	61
9.1	恢复出厂设置.....	61
9.2	系统升级.....	61
10	编程指南.....	62
10.1	IO 编程指南	66
10.1.1	DI0~DI3 sys/class/gpio/gpio487~sys/class/gpio/gpio490	67
10.1.2	(sys/class/gpio/gpioxxx/value)	67
10.2	Led 编程指南	68
10.3	交叉编译.....	72
10.4	AI 算力卡.....	74
10.5	编译内核模块.....	74
10.5.1	在设备上直接编译	74

1 概述

LGW/1TG/33G 是一款高性能多接口带 AI 扩展的边缘计算网关，最高可扩展 26 TOPS AI 算力，并支持 TensorRT/cuDNN/ VisionWorks/OpenCV AI 框架。其搭载 ARM Cortex-A55@2.0GHz 四核处理器，提供强劲的计算平台。该产品采用发行版 Linux 系统，向用户提供灵活多样的二次开发环境。支持安全特性如 Secure Boot 和 TPM2.0，保障软件及数据安全。

2 产品选型

L GW / □ □ / □ □ □

1 2 3 4 5 6 7

示例：LGW/1TG/12G

序号	序号说明	
1	系列代号	用“L”表示
2	产品代号	通讯网关用“GW”表示（网关的英文“Gateway”缩写）
3	产品类型	物联型用“1”表示
4	接口类型	4G 用“G”表示 TCP 用“T”表示 4G+TCP 用“TG”表示
5	功能细分	轻型用“1”表示 功能型用“2”表示 能源型用“3”表示
6	串口数量	串口数量≤2 个用“1”表示 2 个<串口数量≤4 个用“2”表示 4 个<串口数量≤8 个用“3”表示 串口数量>8 个用“4”表示
7	安装方式	导轨用“G”表示 耳挂用“E”表示

3 行业应用

- 国家机关办公建筑与大型公建建筑能耗监测系统；
- 可再生能源监测系统；
- 节约型校园节能监管系统；
- 水质在线监测系统；
- 污染源在线监测系统；
- 智慧农业、林业；
- 油田设备在线监测；
- 输变电设备在线监测系统；
- 配电室在线监测；
- 机房在线监测；

4 技术参数表

4.1 基础性能

CPU: Quad-core Cortex-A55@2.0GHz

GPU: Mail-G52 2EE

NPU: 1.0 TOPS, 支持扩展 Hailo-8 M.2 AI 模块, 最高达 26 TOPS

内存: 4GB, 可扩展

FLASH: 16GB eMMC, 可扩展

4.2 接口性能

以太网端口：4x 10/100/1000Mbps 以太网端口

串行接口：4x RS-232/485/422+8x RS485，工业端子

I/O：4x DI，4x DO

CAN：2x CAN 2.0A/B

Wi-Fi：STA，802.11ac/a/b/g/n，2.4G/5G 双频

蓝牙：BLE 5.0

GPS：支持 GPS、北斗、格洛纳斯定位

USB：2x USB2.0

HDMI：1x HDMI 2.0

扩展接口：1x mSATA（用于扩展 SSD），1x B-Key 3042（用于扩展 Hailo AI）

复位按键：针孔式复位按键

SIM 卡座：2x 标准 SIM

TF 卡：支持 Micro SD

4.3 可信平台模块

TPM：TPM2.0

4.4 机械特性

安装方式：导轨，壁挂

外壳：金属外壳

防护等级：IP30

尺寸：203 x 122 x 49 mm

4.5 设备功率

典型值（OS 空闲态）：4W

最大值（满载）：8W

4.6 电源

电源输入：DC 12-48V，防反接保护

电源接口：工业端子

4.7 环境温湿度

工作温度：-20~ 70℃

环境湿度：5~ 95%（无凝霜）

存储温度：-40 ~ 85℃

4.8 指示灯

LED：4G/5G， 信号强度指示灯（3 颗：L1,L2,L3） SIM1,SIM2,User1,User2,
PWR, STATUS, WARN, Error

4.9 EMC 指标

静电：EN61000-4-2, level 3

辐射电场：EN61000-4-3, level 3

脉冲电场：EN61000-4-4, level 3

浪涌：EN61000-4-5, level 3

传导骚扰：EN61000-4-6 ,level 3

工频磁场：EN61000-4-8, 水平方向/垂直方向 400A/m (>level 2)

振荡波抗扰度：EN61000-4-12, level 3

4.10 物理特性

防震：IEC60068-2-27

跌落：IEC60068-2-32

振动：IEC60068-2-6

4.11 认证

CE、FCC、IC、PTCRB、Verizon Wireless、AT&T

4.12 操作系统

操作系统: Yocto/Linux (Debian 11, Kernel 5.10)

文件系统: 完整的 Debian Core 根文件系统

包管理器: Debian 软件包管理器

4.13 网络互联

网络制式: 5G SA/NSA, LTE Cat4, LTE Cat6 (注: 分型号适配不同网络)

网络接入: APN、VPDN

接入认证: CHAP/PAP

LAN 协议: ARP、Ethernet

WAN 协议: 支持静态 IP、DHCP

4.14 安全

Secure Boot: 支持

4.15 可靠性

备份功能: 支持双 SIM 卡

链路在线检测: 多级链路检测, 断线自动连接

内嵌看门狗: 设备运行自检技术, 设备运行故障自修复

4.16 数据采集协议 (*DSA)

工业协议: Modbus RTU Master/Slave, Modbus TCP Master/Slave, EtherNet/IP,

ISO on TCP, OPC UA Client/Server, Mitsubishi MC 3C/3E/3C OverTCP,
Mitsubishi CPU Port, FINSUDP, HostLink, PPI
电力协议: DLT645-2007, IEC101/104, DNP3.0
其他协议: BACnet, CNC

4.17 维护管理

升级方式: 支持专有升级机制, 利用本地或远程方式进行固件升级

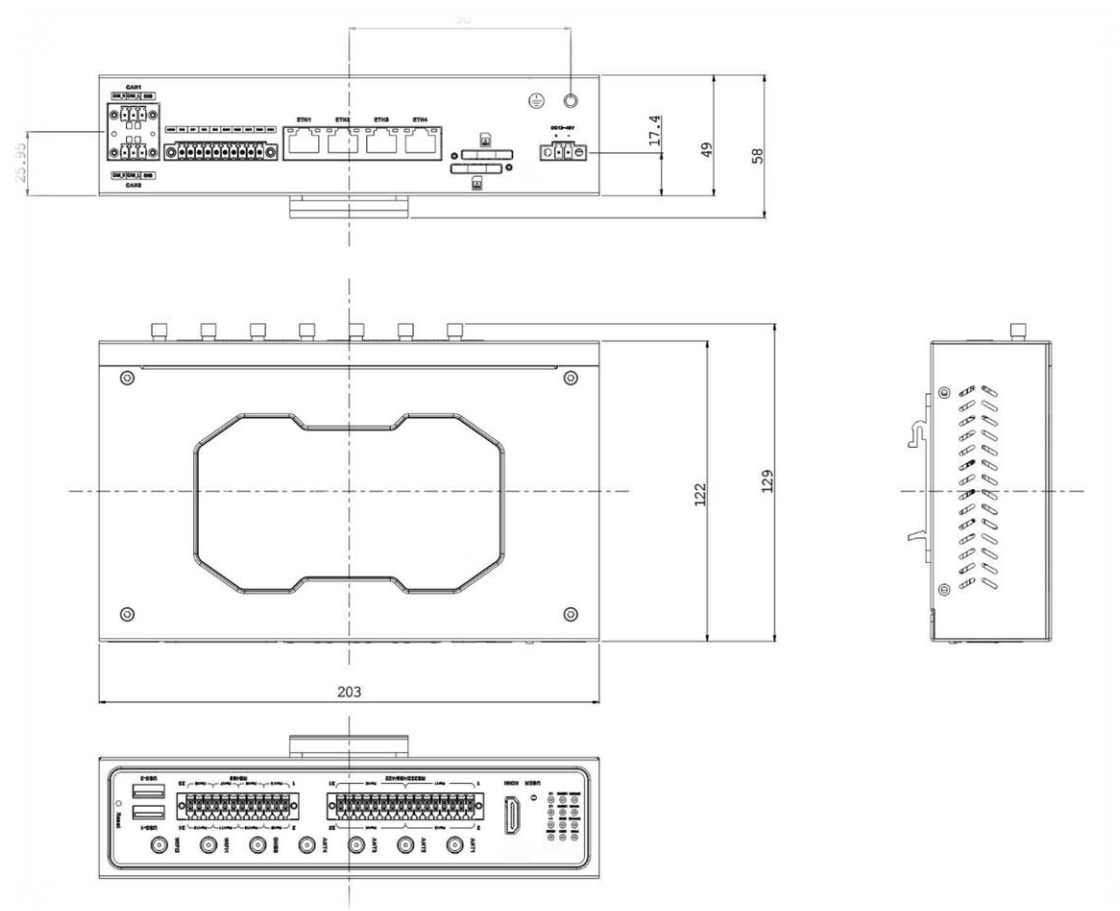
配置方式: WEB

日志功能: 支持本地系统日志、远程日志输出, 重要日志掉电保存

远程管理: 支持 DeviceLive 或 HTTP、HTTPS、SSH 等方式:

平台功能: 支持基于云的参数配置、容器管理、应用和固件管理, 助您进行设备的
远程 管理和应用部署

5 产品结构及接线



5.1 开关量输入接口（Digital Input）

接口标识	功能	描述
COM	DI公共端	4 路数字量输入 DI, 干接点状态 “1”：闭合干接点状态 “0”：断开 隔离 3000VDC
DI0	数字量输入 0 号接口	
DI1	数字量输入 1 号接口	
DI2	数字量输入 2 号接口	
DI3	数字量输入 3 号接口	

5.2 开关量输出接口（Digital Output）

接口标识	功能	描述
GND	D0接地端	4 路数字量输出 DO, 隔离3000VDC
D00	数字量输出 0 号接口	
D01	数字量输出 1 号接口	
D02	数字量输出 2 号接口	
D03	数字量输出 3 号接口	

5.3 USB

LGW/1TG/33G提供两个 USB 2.0 Host 接口。



5.4 LED



LGW/1TG/33G 有12个LED灯，分别对电源及系统运行状态进行指示。

标识	名称	定义
PWR	电源指示灯	上电常亮
STAT	系统运行状态指示灯	当系统正常启动后，STATUS闪烁，如果系统启动阶段发生异常导致系统启动失败；或者恢复出厂操作尚未完成时，STATUS长灭。
WARN	警告指示灯	当系统发生警告异常，系统升级或者恢复出厂尚未完成时，WARN灯闪烁。
ERR	错误指示灯	当系统发生严重错误，系统升级或者恢复出厂尚未完成时，Error灯闪烁。
SIM1	SIM1卡指示灯，	选中SIM卡1进行拨号时常亮，选择SIM卡2拨号或者关闭拨号时，长灭。
SIM2	SIM1卡指示灯，选中则常亮	选中SIM卡2进行拨号时常亮，选择SIM卡1拨号或者关闭拨号时，长灭。
USER1	用户可编程指示灯1	默认熄灭，可由用户编程控制
USER2	用户可编程指示灯2	默认熄灭，可由用户编程控制
4G/5G	蜂窝网连接状态指示灯	拨号成功后常亮
L1	蜂窝网信号强度	见蜂窝网信号强度指示灯说明
L2	蜂窝网信号强度	见蜂窝网信号强度指示灯说明
L3	蜂窝网信号强度	见蜂窝网信号强度指示灯说明

蜂窝网信号强度指示灯

LED	无信号	信号弱 (RSSI < -90)	信号中等 (-90 ≤ RSSI < -70)	信号强 (RSSI ≥ -70)
L1	灭	亮	亮	亮
L2	灭	灭	亮	亮
L3	灭	灭	灭	亮

除了L1, L2, L3信号灯的组合来指示蜂窝信号强度，还有一组LED组合来标识恢复出厂的过程。

LED	状态	
WARN	闪烁	
ERROR	闪烁	
STATUS	熄灭	

执行恢复出厂设置后，系统会进行一次重启，重启完成后，恢复出厂并没有完成，此时WARN灯和ERROR闪烁，STATUS熄灭，在这个状态下不能对设备断电，否则可能会导致一些文件丢失而影响系统功能。这个状态会持续30秒，当恢复出厂完成后，WARN和ERROR熄灭，STATUS闪烁。

5.5 用户可编程按键

LGW/1TG/33G提供API接口，用户可以调用API接口检测可编程按键的状态，然后实现自己的按键逻辑。



5.6 直流输入

LGW/1TG/33G支持12~48V直流输入



5.7 SIM 卡插槽

LGW/1TG/33G支持2个SIM卡插槽，SIM卡需要在断电状态下安装，将SIM卡按压插入卡槽即可。



5.8 MicroSD 卡槽

LGW/1TG/33G有一个MicroSD卡的卡槽，SD不支持热插拔，需要在断电情况下插拔。插入SD卡，给设备上电后，系统会自动挂载所有的分区。

5.9 恢复出厂按键

有一个复位按钮用于系统恢复出厂。参考《恢复出厂设置》进行操作。



5.10 天线接口

LGW/1TG/33G共有7个天线接口，不同型号标配的天线数量不同，将天线拧入相应的天线接口完成天线安装。



标识	名称
ANT1	4G LTE主天线/5G天线
ANT2	4G LTE分集接收天线/5G天线
GNSS	GNSS天线
ANT3	5G天线
ANT4	5G天线
WiFi1	WiFi天线
WiFi2	WiFi天线

5.11 mSATA 硬盘接口

LGW/1TG/33G支持mSata硬盘，出厂默认不配带mSata硬盘，如果用户有大容量的存储需求，需要自购mSata硬盘。

6 入门

在本章中，我们将介绍基于Arm结构的边缘计算机LGW/1TG/33G的基础配置。

6.1 连接到 LGW/1TG/33G

您需要一台计算机，用于连接到LGW/1TG/33G并登录到命令行界面。可以通过以太网网线的方式连接。出厂默认的用户名和密码：

Username: edge Password: security@edge

LGW/1TG/33G设备出厂默认创建了root，但是登录被禁用。如果需要使用root用户，请手动修改系统配置

，输入`sudo -s` 切换至root用户。用户edge在sudo组中，所以您可以在edge用户下使用`sudo`来执行系统级命令。有关其他细节，请参阅第5章中的sudo机制部分。

提示：

当出现`command not found`时，输入`sudo -s`切换至root用户或使用`sudo`命令来操作。

注意：

出于安全考虑，我们建议您禁用默认的用户账户，并创建您自己的用户帐户。

6.1.1 通过 SSH Console 进行连接

LGW/1TG/33G支持通过以太网的SSH连接。使用以下默认IP 地址连接到LGW/1TG/33G。

端口	默认IP
ETH 1	192.168.3.100
ETH 2	192.168.4.100
ETH 3	192.168.5.100
ETH 4	192.168.6.100

6.1.2 Linux 用户

提示:

这些步骤适用于您在 Linux PC 上连接到 LGW/1TG/33G。请不要将这些步骤应用于 LGW/1TG/33G 设备本身。在您运行 ssh 命令之前，请确保将 PC 的以太网口 IP 地址配置在特定范围内。

ETH1: 192.168.3.0/24, ETH2: 192.168.4.0/24, ETH3: 192.168.5.0/24, ETH4: 192.168.6.0/

在 Linux PC 上使用 ssh 命令访问 LGW/1TG/33G 的 ETH1 端口。

```
user@PC:~$ ssh edge@192.168.3.100
```

输入 **yes** 继续完成连接。

```
The authenticity of host '192.168.3.100 (192.168.3.100)' can't be established.  
ECDSA key fingerprint is SHA256:jqiLREbTX6Ut2whNFdpLvCcQfM3KUnl3Ta7/dWppBCU.  
Are you sure you want to continue connecting (yes/no)? yes
```

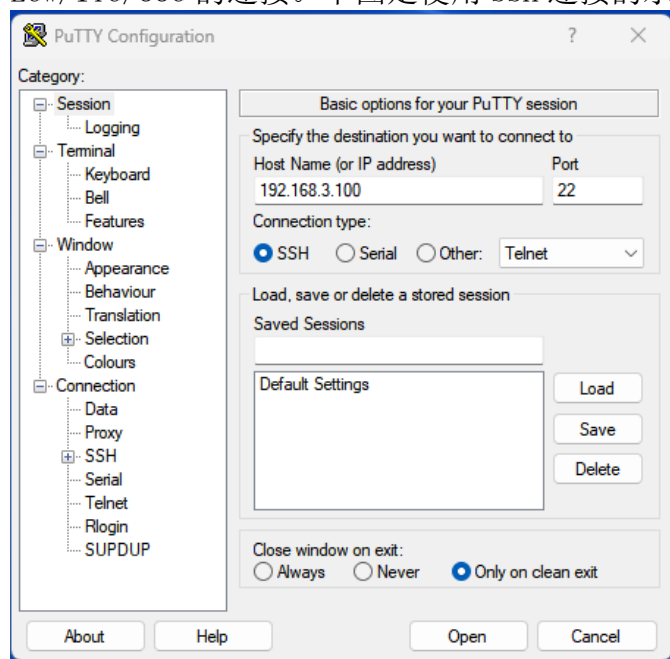
当界面出现终端提示符 `edge@edge-computer:~$`，可输入 `shell` 命令时，连接成功。

6.1.3 Windows 用户

提示:

这些步骤适用于您在 Windows PC 上连接 LGW/1TG/33G。请不要将这些步骤应用于 LGW/1TG/33G 设备本身。请在您的 Windows PC 机上做出如下步骤:

点击链接 <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>, 下载 PuTTY (免费的软件), 在 Windows 环境下以 SSH 命令的方式来建立与边缘计算机 LGW/1TG/33G 的连接。下图是使用 SSH 连接的示例:



6.2 用户账户管理

6.2.1 切换到 root 用户

您可以使用 `sudo -s` 的命令来切换到 root 用户。出于安全考虑，不要在 root 权限下操作所有的命令。

提示：

点击链接获取更多关于 `sudo` 命令的信息。

<https://wiki.debian.org/sudo>

注意：

您可能在非 root 权限下使用一些管道或者重定向行为的时候得到 “permission denied” 的提示，此时您必须使用 `'sudo su -c'` 来取代 `'>','<','>>','<<','etc'` 等命令，需要包含完整命令的单引号。

6.2.2 创建和删除用户账户

您可以使用 `useradd` 和 `userdel` 命令来创建和删除用户的账户。请务必在主界面使用这些命令来设置该账户的相关访问权限。以下是如何在 `sudo` 组中创建 `test1` 的示例（`test1` 用户的默认登陆环境是 `bash`，其主目录是 `/home/test1`）

```
edge@edge-computer:~$ sudo useradd -m -G sudo -s /bin/bash test1
```

修改 `test1` 的密码，用 `passwd` 命令，输入新密码然后重复输入新密码以确认更改

```
edge@edge-computer:~$ sudo passwd test1
New password:
Retype new password:
passwd: password updated successfully
```

如果要删除用户 `test1`，请使用命令 `userdel`

```
edge@edge-computer:~$ sudo userdel test1
```

6.2.3 禁用默认的用户账户

注意：

在禁用默认帐户之前，应首先创建一个用户帐户

使用 passwd 命令来锁定默认用户账户，使 edge 用户无法登录

```
edge@edge-computer:~$ sudo passwd -l edge
passwd: password expiry information changed.
```

对 edge 用户进行解除锁定

```
edge@edge-computer:~$ sudo passwd -u edge
passwd: password expiry information changed.
```

6.3 网络管理和系统管理

LGW/1TG/33G 基于 debian 11 系统，因此可以使用 Linux 原生的命令进行网络管理和系统管理；为了便于用户配置，提供 web 界面，用户可以方便的通过 web 进行网络管理和系统管理，但需要注意的是，当启用 IEOS 功能后，IEOS 会接管网络管理和系统管理，此时通过 Linux 原生命令进行网络管理和系统管理可能会失效；设备出厂默认启用 IEOS，如果用户需要基于 Linux 原生的命令行进行网络管理和系统管理，需要先关闭 IEOS。

6.3.1 基于 IEOS 的 web 管理

IEOS 提供 web 界面，用户可以通过 web 进行以太网口 ip 地址，蜂窝拨号，Wi-Fi Station，DHCP Client/Server，静态路由

，防火墙等网络配置；也可以对系统时间，时区，固件升级和系统重启等进行操作；另外用户可以通过 DeviceLive 平台对 LGW/1TG/33G 设备进行远程监控和管理。

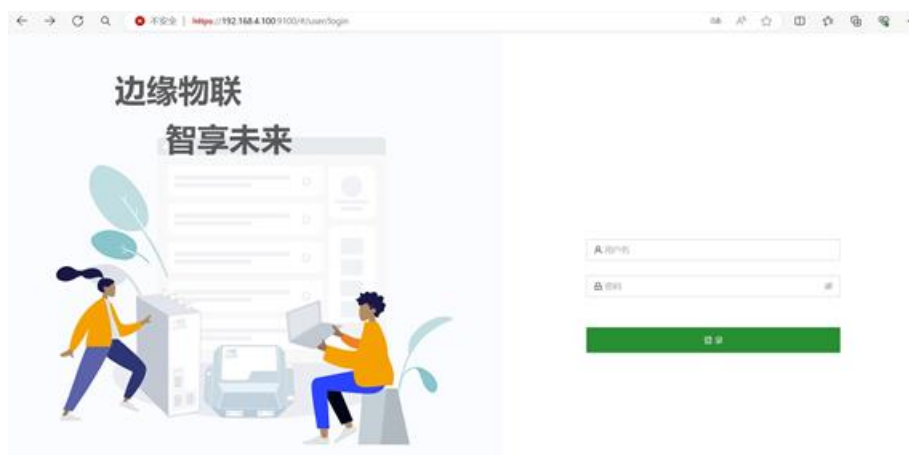
IEOS 采用状态和配置分离的设计方案，分为网络管理，系统管理和状态三大功能板块。网络管理菜单和系统管理菜单下只能进行网络和系统的相关配置，状态信息需要统一到状态页面查看。

重要说明：使用 IEOS 程序管理网络配置和系统配置的时候，如果同时使用 Linux 原生命令，二者可能会相互影响，导致异常的运行状态。建议 IEOS 支持的配置都通过 IEOS web 来实现管理，IEOS 不支持的配置，比如 VPN，可以结合 Linux 原生命令一起实现配置目标。

6.3.2 登录 web

考虑到用户的程序可能会需要使用 HTTP/HTTPS 标准端口号 80/443，IEOS 使用 9100 端口号做为 HTTPS 连接的端口，不支持通过 HTTP 访问；当用户使用 HTTP 访问 web 的时候，会自动跳转到通过 HTTPS。本文档以通过 eth2 的默认地址 192.168.4.100 为例来进行说明。用户在浏览器中输入 192.168.4.100:9100 后跳转到登录页面

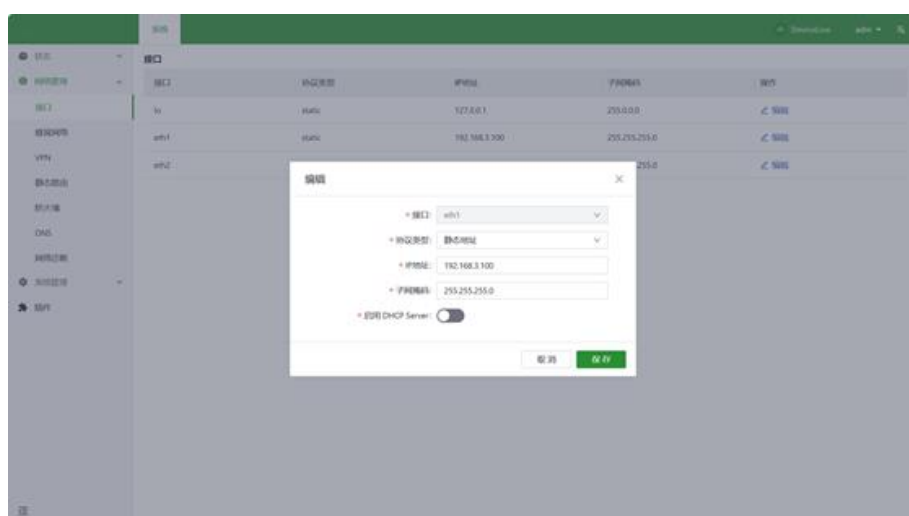
重要说明：IEOS 程序启用时，会预留一些端口号供内部通信使用，预留端口号范围 9100 ~ 9200，启用 IEOS 后，客户的程序应该避开使用这些端口号，否则可能会产生冲突导致功能异常。



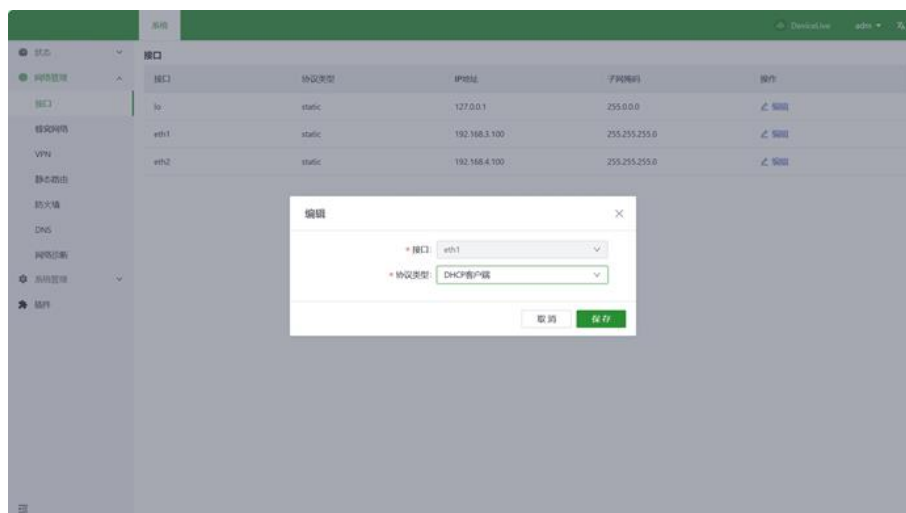
6.3.3 网络管理

6.3.4 配置以太网接口

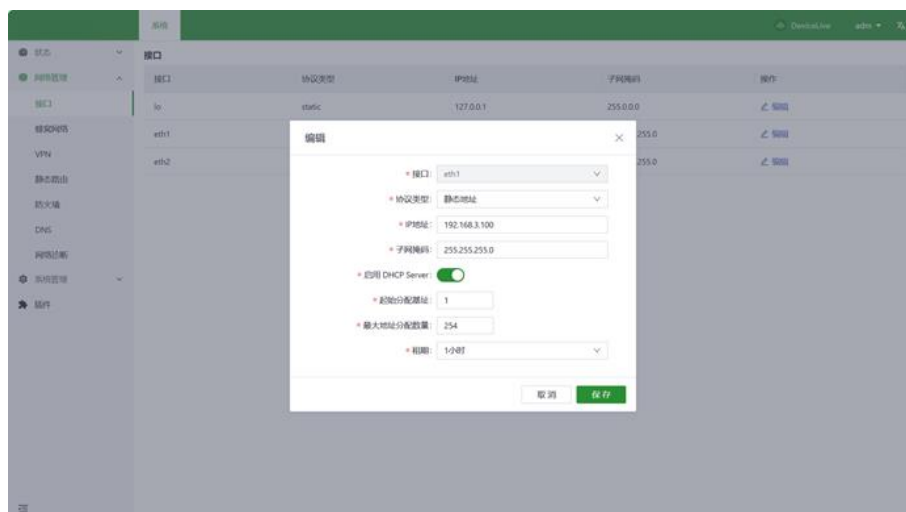
给 eth1 接口配置静态 IP 地址



给 eth1 接口配置 DHCP Client



在 eth1 接口上启动 dhcp server 功能，给 eth1 的下挂设备分配地址



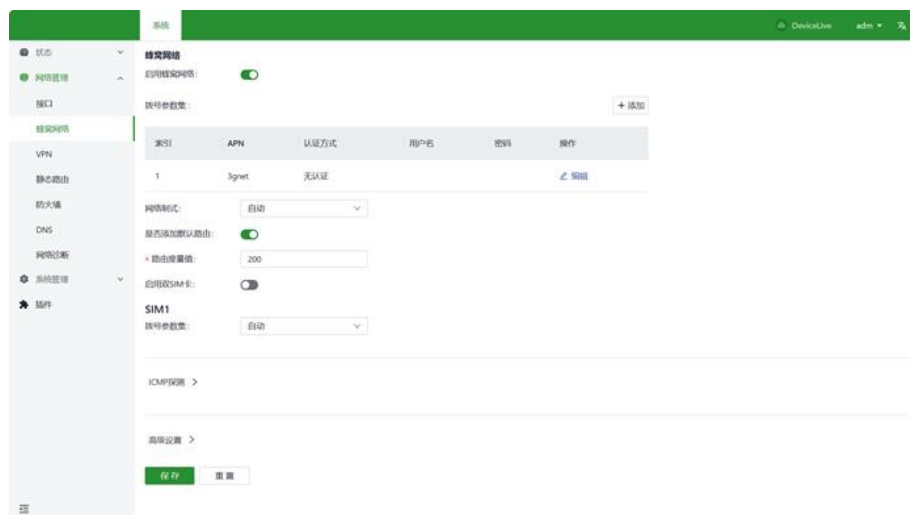
DHCP Server 配置参数说明：

启用 **DHCP Server**：DHCP Server 功能的开关

起始分配地址：DHCP Server 地址池起始基址，网段 + 起始地址 = 地址池的起始 ip 地址。在截图中，eth1 的网段是 192.168.3.0/24，基址是 1，则地址池的起始地址是 192.168.3.1/24.

最大地址分配数量：地址池中的最大地址数量。租期：租期时间

6.3.5 配置蜂窝拨号



蜂窝网参数说明：

启用蜂窝网络：蜂窝功能的开关；默认是开启的状态。

拨号参数集：拨号参数集，用于专网卡拨号时配置 APN，用户名，密码以及认证方式等信息。如果不是专网卡，通常不需要修改这里的配置。拨号参数集最多可以添加 10 条记录。

网络制式：蜂窝的网络制式，可以选择 3G，4G 等相关网络制式，比如 LTE，WCDMA 等。如果不清楚选哪种网络制式，则选择自动；程序会自动选择一个最合适的网络制式。默认值是自动。

是否添加默认路由：启用添加默认路由功能，开启后，当拨号成功后，会添加一条蜂窝口的默认路由。默认是开启的。

路由度量值：蜂窝口默认路由的度量值，当蜂窝口，Wi-Fi 和以太网口都配置了默认路由的时候，度量值最小的生效。

网络制式:	自动
是否添加默认路由:	☒
* 路由度量值:	200
启用双SIM卡:	☒
选择主卡:	SIM1
最大拨号次数:	3
SIM1	
拨号参数集:	自动
SIM2	
拨号参数集:	自动

启用双 SIM 卡： 启用双卡功能。EC942 为了提高网络的可靠性，支持双卡单拨。设备上需要插入 2 张 sim 卡，开启该功能后，如果 sim1 卡因为欠费导致拨号失败，会自动切换到 sim2 卡进行拨号。默认是关闭的。

选择主卡： 主 sim 卡，拨号时会优先使用选中的 sim 卡进行拨号，拨号失败达到一定次数时，在切换到另外一张 sim 卡进行拨号，默认先使用 sim1 进行拨号。

最大拨号次数： 开启双卡单拨功能后，当前拨号 sim 卡拨号达到指定次数，切换到另外一张 sim 卡进行拨号。

拨号参数集： sim 卡选用的拨号参数集，默认值是自动。通常专网卡通常需要配置拨号参数集，并在此处选择拨号参数集的 Index。

PIN Code： sim 卡的 PIN 码。

ICMP探测

▼

ICMP探测服务器:	
* 间隔时间:	60 秒 (1-86400)
* 超时时间:	5 秒 (1-86400)
* 最大重试次数:	3
严格探测:	☐

无线蜂窝网络比较复杂，有些时候会出现拨号假连接的情况，即拨号状态是成功的，但却 ping 不通目标地址；发生这些状况的时候，重新拨号一次就可以恢复正常。IEOS 蜂窝拨号支持 ICMP 探测以发现假连接。建议使用蜂窝联网的客户开启 ICMP 探测，这样出现假连接的时候也可以快速恢复。

ICMP 探测参数：

ICMP 探测服务器： ICMP 探测地址；可以配置 2 个探测地址，只要有 1 个地址探测成功，则说明蜂窝没有出现假连接。2 个地址都没有配置的时候，ICMP 探测功能处

于关闭状态。

间隔时间：间隔多久进行一次 ICMP 探测。

超时时间：ICMP 探测超时时间，等待多久没有收到探测响应报文，则认为此次探测失败

最大重试次数：最大探测次数；当探测失败达到这个值的时候，触发重新拨号。

取值范围[1, 5]

严格探测：是否开启严格探测。关闭严格探测时，探测程序在每个探测周期会检测蜂窝接口接收的报文是否发生变化，如果有变化，说明蜂窝网络是通的，则不会发送 ICMP 报文进行探测，这样可以节省一些流量；如果开启探测，则无论蜂窝接口接收报文的数量是否发生变化，都会周期性的发送 ICMP 探测报文。默认是关闭的。

高级设置 ▾

启用调试模式： ☐

启用无限重拨： ☒

重拨间隔： 秒 (0-3600)

信号查询间隔： 秒 (0: 禁用)

高级配置中是一些不常用的设置选项。

启用调试模式：是否启用 debug 功能，开启后，日志中会增加一些拨号相关的调试信息，默认关闭。启用无限重拨：开启无限重拨。某些情况下，拨号会处于异常状态，通过重启系统可以恢复正常；默认无限重拨是关闭的，当拨号失败一定次数后，会重启系统尝试恢复。由于拨号默认是开启的，有些客户在不插 sim 卡的情况下，拨号失败，系统会重启，针对这种情况，可以开启无限重拨；这样无论拨号失败多少次，系统都不会重启。

重播间隔：拨号间隔；但某次拨号失败时，进行下一次拨号之前等待的时间。

信号查询间隔：信号查询间隔。信号不好时，可能会产生假连接的问题；这时重新拨号有一定概率可以解决假连接的问题。拨号程序会定期检测信号强度，这里配置的是信号检测周期。

6.3.6 配置 Wi-Fi Station

WiFi

启用 WiFi: ☒

接口类型: ☒ 客户端 ☐ 接入点

* 客户端 SSID: 扫描

是否添加默认路由: ☐

认证方式: 无认证

SSID 扫描

信道	SSID	BSSID	安全	信号(dBm)	操作
暂无数据					

保存 重置

启用 Wi-Fi: 使能开关; 默认关闭

客户端 SSID: 需要连接的 ssid, 可以手动输入; 也可以通过扫描按钮获取附近可连的 ssid 是否添加默认路由: 是否启用添加默认路由功能, 开启后, 当 wifi 连接成功后, 会添加一条 wlan 口的默认路由。默认是开启的。

路由度量值: wifi 口默认路由的度量值, 当蜂窝口, Wi-Fi 和以太网口都配置了默认路由的时候, 度量值最小的生效。

认证方式: 认证方式, 支持无认证, WPA-PSK, WPA2-PSK, WPA-PSK/WPA2-PSK Mixed

加密方式: 加密方式; 支持 CCMP, TKIP, TKIP 和 CCMP

WPA/WPA2 PSK 密钥: 密钥信息

6.3.7 配置静态路由

这里配置的是以太网的静态路由, 当同时配置了以太网, 蜂窝, wifi 的默认路由时, metric 值最小的那一条默认路由生效。需要保证默认路由的 Metric 值是不同的。

静态路由配置参数：

接口：静态路由的出接口

目标网络：目标网络

目标网络掩码：目标网络掩码网关：

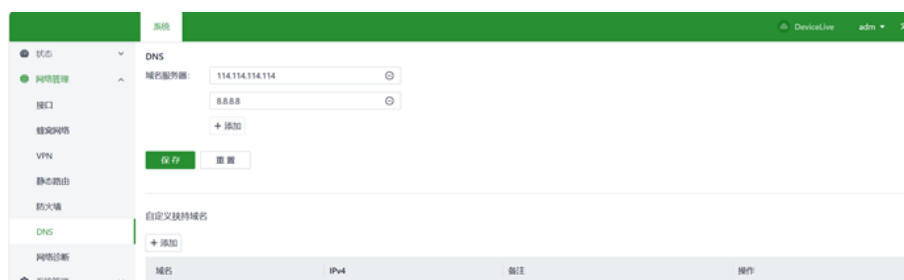
下一跳地址路由度量值：静态路由的度量值

6.3.8 配置防火墙



目前只支持使用 iptables 命令进行配置。

6.3.9 配置 DNS



域名服务器：DNS Server 地址，最多支持配置 4 个

自定义挟持域名：域名劫持功能，可以实现 IP 地址和域名之间的绑定。



6.3.10 网络诊断

网络诊断支持 ping，traceroute 和 nslookup 功能。

状态

网络管理

接口

蜂窝网络

VPN

静态路由

防火墙

DNS

网络诊断

系统管理

插件

网络诊断

Ping

* 目标:

协议: ipv4

开始

Traceroute

* 目标:

协议: ipv4

开始

Nslookup

* 目标:

开始

6.3.11 配置 OpenVPN 服务器

OpenVPN 的配置, 可以通过 web 页面手动配置, 也可以通过导入配置文件的方式进行配置。

1. 手动配置

客户端配置

服务器端配置

配置方式:

手动配置

* 服务端名称:

server

状态:

☒

* 接口类型:

tun

网络类型:

subnet

* 虚拟网络:

5.5.5.0

* 网络掩码:

255.255.255.0

* CA证书:

上传

ca.crt

下载

删除

* 公钥证书:

上传

server.crt

下载

删除

* 私钥证书:

上传

server.key

下载

删除

* DH参数:

上传

dhparam.pem

下载

删除

高级设置

* 服务器地址:

0.0.0.0

* 协议类型:

udp

* 服务器端口:

1194

加密算法:

AES-128-CBC

哈希算法:

SHA3-512

密钥保持:

☒

隧道保持:

☒

LZO压缩:

关闭

客户端最大数量:

20

用户:

nobody

用户组:

nogroup

* 日志级别:

3

* 连接检测时间间隔:

60

* 连接检测超时时间:

300

允许客户端到客户端的通信:

☒

推送到客户端:

route 172.20.0.0 255.255.0.0

服务端名称：唯一标识服务器端配置的字符串，只能包含英文字母和数字，并以字母开头，长度不超过 64 个字符。

状态：开启或关闭，用于控制 OpenVPN 服务器的启动或停止。**接口类型：**目前只支持 tun 模式。

网络类型：支持 p2p, net30 和 subnet。p2p 和 net30 用于建立点对点的连接，subnet 用于建立虚拟局域网。

虚拟网络：subnet 模式下，需要配置虚拟网络的网段信息；如果是 p2p 或 net30 配置的则是本端 IP 地址和远端 IP 地址。

网络掩码：subnet 模式下，需要配置虚拟网络的子网掩码信息。

CA 证书：此处需要导入 CA 证书文件

公钥证书：此处需要导入服务器的公钥证书文件私钥

证书：此处需要导入服务器的私钥证书文件

DH 参数：此处需要导入服务器的 DH 参数文件

高级设置中可以配置一些改动频率比较低的参数。

服务器地址：OpenVPN 监听的 IP 地址，默认监听 0.0.0.0，即监听所有 IP 地址

协议类型：支持 udp 和 tcp，默认 udp。

服务器端口：服务器监听的端口号，默认 1194。

加密算法：配置用于保证数据传输机密性的加密算法。**哈希算法：**配置用于保证数据传输完整性的哈希算法。

密钥保持：启用密钥保持后，即使 OpenVPN 因网络异常出现短暂断开也能快速恢复，无需重新握手。

隧道保持：启用隧道保持后，可以防止隧道因短暂网络波动而关闭。

LZO 压缩：是否启用 LZO 压缩。

客户端最大数量：配置同时连接 OpenVPN Server 的客户端数量。

用户：指定运行 OpenVPN 进程的用户

用户组：指定运行 OpenVPN 进程的用户组

日志级别：取值范围 1~11，数值越大，日志信息越详细；可以在日志页面，ALL 类型日志中查看 OpenVPN 日志。

连接检测时间间隔：发送心跳包检测 VPN 连接是否存活的周期

连接检测超时时间：如果在超时时间内未收到对方响应，判定连接已断开并尝试重启允许客户端到客户端的通信：**是否允许客户端和客户端相互通信**

推送到客户端：可以推送路由和 DNS 等信息到客户端。

2. 导入配置

状态

网络管理

接口

蜂窝网络

LoRaWAN

GPS

VPN

静态路由

客户端配置 服务器端配置

配置方式: 导入配置

* 服务端名称: server

状态: ☒

* 接口类型: tun

* 配置文件: 上传

保存 重置

服务端名称：唯一标识服务器端配置的字符串，只能包含英文字母和数字，并以字母开头，长度不超过 64 个字符。

状态：开启或关闭，用于控制 OpenVPN 服务器的启动或停止。 **接口类型：**目前只支持 tun 模式。

配置文件：导入 OpenVPN Server 配置文件，配置文件中需要包含 CA 证书，服务器证书和服务器密钥等信息。

6.3.12 配置 OpenVPN 客户端

1. 手动配置

添加

×

配置方式：

手动配置

* 客户端名称：

client1

状态：

* 接口类型：

tun

网络类型：

subnet

OpenVPN服务器：

10.5.30.193 1194 udp

+ 添加

* 日志级别：

3

* CA证书：

上传

* 公钥证书：

上传

* 私钥证书：

上传

高级设置

▼

加密算法：

AES-128-CBC

哈希算法：

SHA3-512

密钥保持：

隧道保持：

LZO压缩：

开启

用户：

nobody

用户组：

nogroup

* 连接检测时间间隔：

60

* 连接检测超时时间：

300

取消

保存

1

36

客户端名称：唯一标识服务器端配置的字符串，只能包含英文字母和数字，并以字母开头，长度不超过 64 个字符。

状态：开启或关闭，用于控制 OpenVPN 服务器的启动或停止。接口类型：目前只支持 tun 模式。

网络类型：支持 p2p, net30 和 subnet。p2p 和 net30 用于建立点对点的连接，subnet 用于建立虚拟局域网。当网络类型是 p2p 或 net30 时，需要配置本端 IP 地址和远端 IP 地址。

OpenVPN 服务器：配置 OpenVPN 服务器的 IP 地址，端口号以及协议信息，格式为：ip port proto，例如：

192.168.3.200 1194 udp，最多可以配置 10 个服务器地址。OpenVPN 客户端会按照顺序一次尝试

建立连接，直到成功建立 OpenVPN 隧道。

日志级别：取值范围 1~11，数值越大，日志信息越详细；可以在日志页面，ALL 类型日志中查看 OpenVPN 日志。

CA 证书：此处需要导入 CA 证书文件

公钥证书：此处需要导入客户端的公钥证书文件 私钥证书：此处需要导入客户端的私钥证书文件 高级设置中可以配置一些改动频率比较低的参数。

加密算法：配置用于保证数据传输机密性的加密算法。哈希算法：配置用于保证数据传输完整性的哈希算法。

密钥保持：启用密钥保持后，即使 OpenVPN 因网络异常出现短暂断开也能快速恢复，无需重新握手。

隧道保持：启用隧道保持后，可以防止隧道因短暂网络波动而关闭。

LZO 压缩：是否启用 LZO 压缩。

用户：指定运行 OpenVPN 进程的用户

用户组：指定运行 OpenVPN 进程的用户组

连接检测时间间隔：发送心跳包检测 VPN 连接是否存活的周期

连接检测超时时间：如果在超时时间内未收到对方响应，判定连接已断开并尝试重启

2. 导入配置

编辑

✕

配置方式:

导入配置

▼

* 客户端名称:

client1

状态:

☒

* 接口类型:

tun

▼

* 配置文件:

⬆️ 上传

取消

保存

客户端名称：唯一标识客户端端配置的字符串，只能包含英文字母和数字，并以字母开头，长度不超过 64 个字符。

状态：开启或关闭，用于控制 OpenVPN 客户端的启动或停止。接口类型：目前只支持 tun 模式。

配置文件：导入 OpenVPN Client 配置文件，配置文件中需要包含 CA 证书，客户端证书和客户端密钥等信息。

6.4 系统管理

6.4.1 基础配置

云管理

状态

网络管理

系统管理

基础设置

软件升级

其他

云管理

启动云服务:

☒

* 云服务器:

DeviceLive

▼

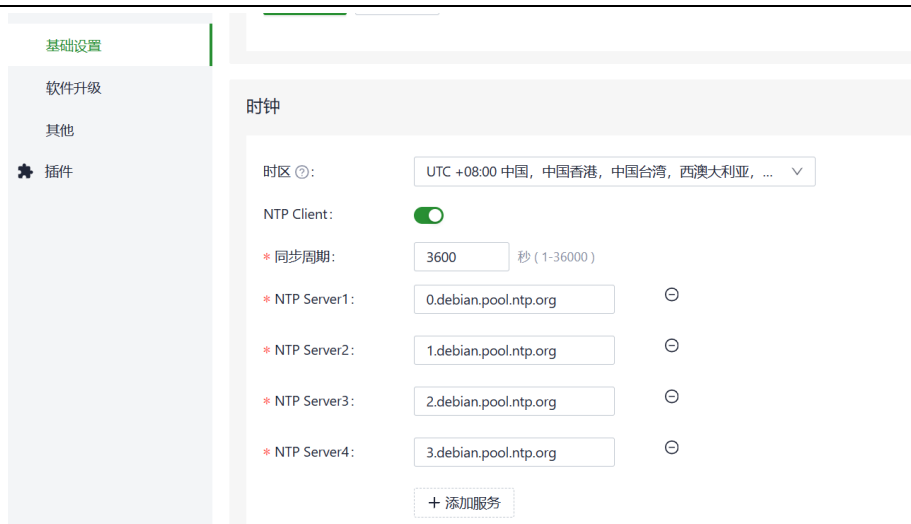
保存

重置

启动云服务： 对接 DeviceLive 平台的使能开关；DeviceLive 是设备的远程监控和管理平台；

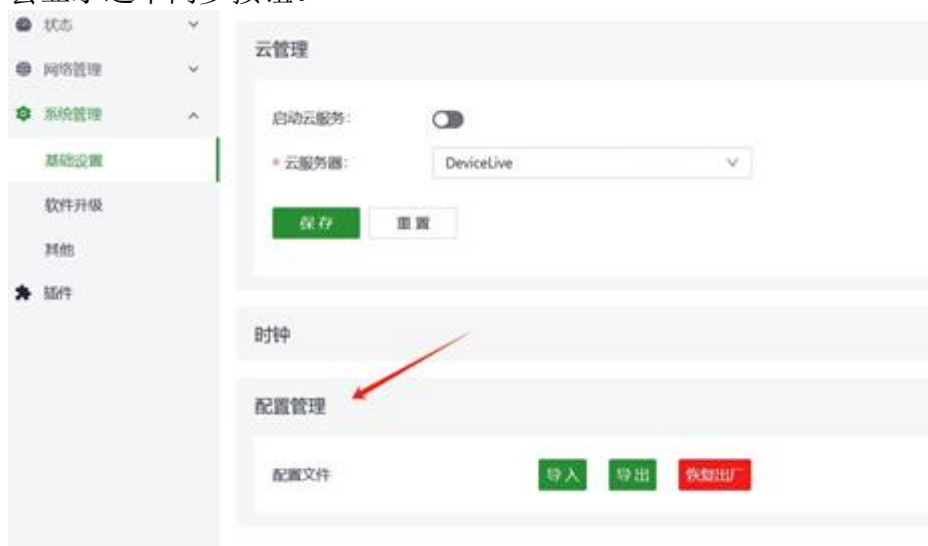
云服务器： DeviceLive 平台有 2 个地址：一个是国内平台的地址，另外一个海外平台地址；这里选择连接哪个平台。

时区和 NTP client



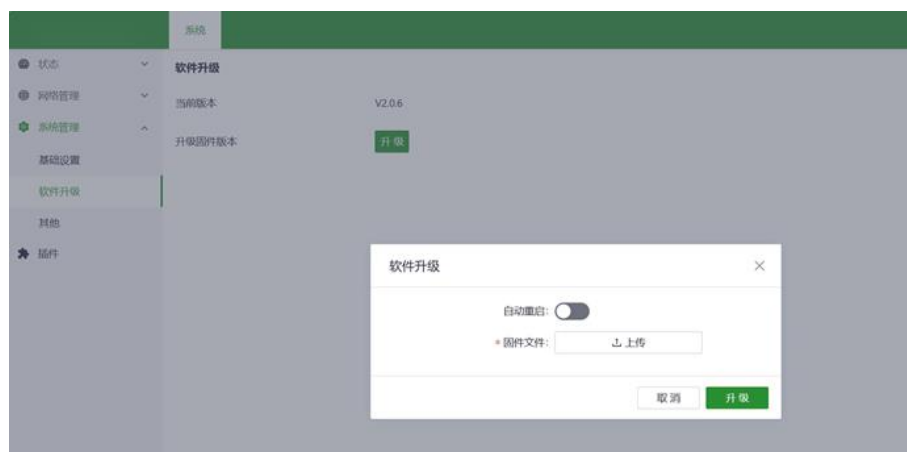
最多可以配置 10 个 NTP Server 地址，程序周期性的依次向每个服务器地址发送同步请求，同步成功后，将系统时间写入 RTC，并不再继续向后面的 NTP 服务器发送同步请求。

除了使用 NTP 同步时间，在 Device Info 状态页面有一个同步按钮可以手动同步时间，但只有当设备时间和本地时间(访问设备所使用的电脑的时间)相差超过 3s 时，才会显示这个同步按钮。



这里支持配置的导入，导出以及恢复出厂。

6.4.2 固件升级



自动重启选项默认是关闭的，升级固件之后，需要手动重启系统才能生效；开启自动重启选项后，升级固件成功后会自动重启系统。

6.4.3 其他

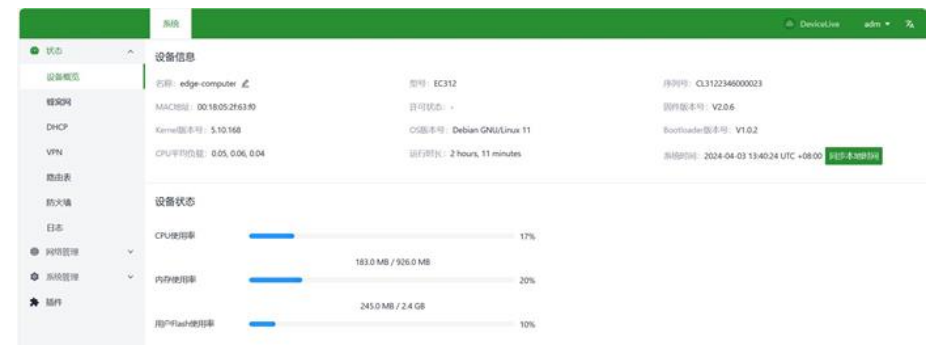


这个页面有重启系统和重置系统 2 个功能。重置系统需要慎重使用，重置系统功能会将系统的配置状态和文件系统状态恢复到和出厂时一致，也就是说用户安装的软件也会被清除。

6.5 状态

6.5.1 设备信息

设备信息状态页面显示主机名，设备型号，序列号，固件版本，内核版本，文件系统版本以及 CPU，内存和磁盘空间的使用概况。



6.5.2 蜂窝拨号状态信息

蜂窝拨号状态页面显示当前拨号使用的 sim 卡，IMEI，IMSI，ICCID，信号强度，以及拨号获取的 IP 地址，DNS 等信息。



6.5.3 Wi-Fi Station 状态信息

Wi-Fi 状态页面显示 Wi-Fi 连接成功后获取的 IP 地址，网关以及 DNS 等信息。

WiFi 状态		
接口类型: STA	状态: 未连接	接口名称: -
IP地址: -	子网掩码: -	连接时长: -
网关: -	DNS: -	MTU: -
信号强度: -		

6.5.4 DHCP Server 状态信息

DHCP Server 状态页面显示设备做为 DHCP Server，分配出去的 IP 地址，客户端主机名，客户端主机 MAC 以及过期时间。

DHCP 分配			
主机名	主机ip	主机mac	租约过期时间
暂无数据			

6.5.5 路由状态信息

路由状态页面显示 IPv4 直连路由，静态路由以及路由邻居等信息。

路由列表			
活动IPv4路由			
接口	对象	IPv4网关	跃点数
docker0	172.17.0.0/16	-	-
eth1	192.168.3.0/24	-	-
eth2	192.168.4.0/24	-	-
IPv4网络邻居			
IPv4地址	MAC地址	接口	
192.168.4.101	74:5d:22:b7:a9:28	eth2	

6.5.6 防火墙状态信息

防火墙状态信息显示过滤规则，IP 地址映射规则等信息。

状态

设备概览

配置网

DHCP

VPN

路由表

防火墙

日志

网络管理

系统管理

插件

filternatmangleraw

防火墙状态

表 Filter

链 INPUT (策略: ACCEPT 数据包: 134 流量: 891K)

数据包	流量	对象	协议	入口	出口	源地址	目标地址	选项
13656	5351K	ACCEPT	all	lo	*	0.0.0.0/0	0.0.0.0/0	/^ \$a3 ^/
3647	802K	input_rule	all	*	*	0.0.0.0/0	0.0.0.0/0	/^ \$a3: Custom in...
3513	793K	ACCEPT	all	*	*	0.0.0.0/0	0.0.0.0/0	dststate RELATED,E...
63	3276	syn_flood	tcp	*	*	0.0.0.0/0	0.0.0.0/0	tcp_flags 0x17/0x0...

链 FORWARD (策略: ACCEPT 数据包: 0 流量: 0)

数据包	流量	对象	协议	入口	出口	源地址	目标地址	选项
0	0	forwarding_rule	all	*	*	0.0.0.0/0	0.0.0.0/0	/^ \$a3: Custom fo...
0	0	ACCEPT	all	*	*	0.0.0.0/0	0.0.0.0/0	dststate RELATED,E...

链 OUTPUT (策略: ACCEPT 数据包: 0 流量: 0)

数据包	流量	对象	协议	入口	出口	源地址	目标地址	选项
13656	5351K	ACCEPT	all	*	lo	0.0.0.0/0	0.0.0.0/0	/^ \$a3 ^/

6.5.7 日志信息

日志页面可以查看系统日志，用户日志以及设置查看的日志等级，包括 Error, Info, Debug 等级别。日志也可以下载到本地。

状态

设备概览

配置网

DHCP

VPN

路由表

防火墙

日志

网络管理

系统管理

插件

类别: System 级别: 全部 关键字: 查询 重置

50行 不刷新

级别	时间	内容
正常	2024-04-03 14:11:09	dia[2891]
正常	2024-04-03 14:11:10	dia[2891]
正常	2024-04-03 14:11:10	dia[2891]: +CME ERROR: 10
正常	2024-04-03 14:11:10	dia[2891]
正常	2024-04-03 14:11:10	dia[2891]
正常	2024-04-03 14:11:10	dia[2891]: +CME ERROR: 10
正常	2024-04-03 14:11:10	dia[2891]
正常	2024-04-03 14:11:10	dia[2891]
正常	2024-04-03 14:11:10	dia[2891]: +CME ERROR: 10
正常	2024-04-03 14:11:10	dia[2891]
正常	2024-04-03 14:11:12	dia[2891]
正常	2024-04-03 14:11:12	dia[2891]: +CME ERROR: 10
正常	2024-04-03 14:11:12	dia[2891]
正常	2024-04-03 14:11:12	dia[2891]

6.5.8 基于 Linux 的命令行管理

使用 Linux 命令行进行网络配置和系统配置的时候，首先需要关闭 IEOS 程序。IEOS 是通过 systemctl 来管理的，关闭 IEOS 的方式如下：

```
systemctl stop ieos_daemon
```

这种关闭只针对本次启动生效，系统重启后，IEOS 程序依然会启动，禁止 IEOS 程序开机启动的方式如下：

```
systemctl disable ieos_daemon
```

重要说明：关闭 IEOS 之后，拨号，Wi-Fi 等无线联网功能需要用户基于 Linux 原生命令来实现，而且也无法实现对接 DeviceLive 平台对设备进行远程管理。

6.6 网络管理

6.6.1 设置一个静态 IP 地址

如果要为 LGW/1TG/33G 设置静态 IP 地址,

通过命令 `vim /etc/network/interfaces.d/eth1`

或 `vim /etc/network/interfaces.d/eth2` 修改对应的网络配置文件来对以太网接口的默认网关，地址，网络以及子网掩码进行设置。这里以给 `eth2` 口设置静态 IP 来举例说明：

```
auto eth2
iface eth2 inet static
    address 192.168.3.100
    netmask 255.255.255.0
    gateway 192.168.3.254
```

修改接口 IP 配置后，执行 `/etc/init.d/networking restart`，重启网络服务，使配置生效。

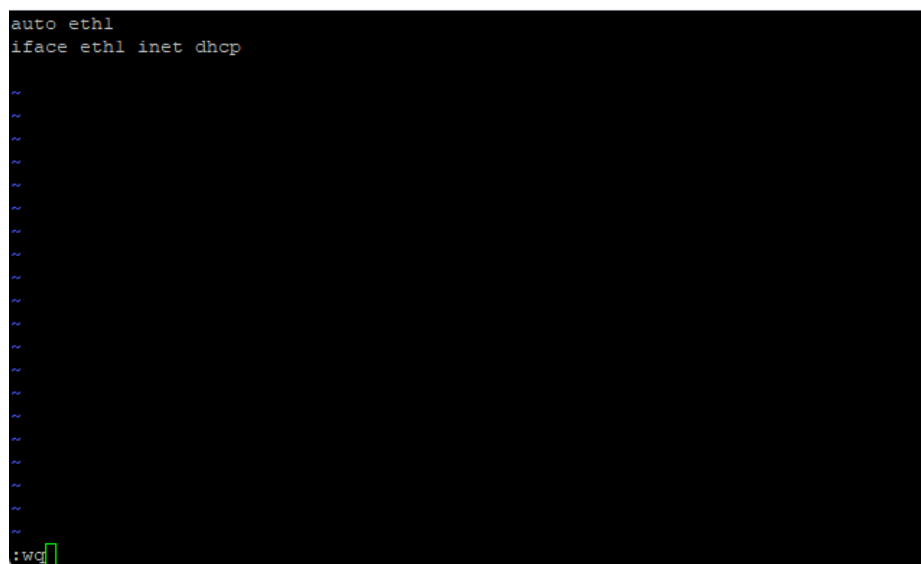
6.6.2 设置一个动态 IP 地址

如果要为 LGW/1TG/33G 设置动态 IP 地址，

通过命令 `vim /etc/network/interfaces.d/eth1`

或 `vim /etc/network/interfaces.d/eth2` 修改对应的网络配置文件，在 `inet` 后设置为 DHCP 将自动获取 IP 地址。

这里以给 `eth1` 口设置动态 IP 来举例说明。



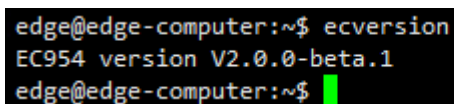
The screenshot shows a terminal window with a black background and white text. The first two lines of the configuration are `auto eth1` and `iface eth1 inet dhcp`. Below these lines, there are several lines of tilde characters (~) representing empty space in the editor. At the bottom left, the text `:wq` is visible, indicating the user has pressed the 'w' and 'q' keys to save and quit the editor.

修改接口 IP 配置后，执行 `/etc/init.d/networking restart`，重启网络服务，使配置生效。

6.7 系统管理

6.7.1 查询固件版本

要检查 LGW/1TG/33G 的计算机固件版本，请键入：



The screenshot shows a terminal window with a black background and white text. The first line shows the command `edge@edge-computer:~$ ecversion`. The second line shows the output `EC954 version V2.0.0-beta.1`. The third line shows the prompt `edge@edge-computer:~$` followed by a green cursor.

加上-a 的选项可以看到完整的版本信息:

```
edge@edge-computer:~$ ecversion -a
EC954 version V2.0.0-beta.1 Build 2023127
edge@edge-computer:~$
```

6.7.2 查看可用磁盘空间

要确定可用驱动器空间的量, 请使用带有-h 选项的 df 命令。系统将返回按文件系统分解的驱动器空间量。LGW/1TG/33G 产品, 用户可用的磁盘分区是/dev/mmcblk0p8。下面是一个示例:

```
edge@edge-computer:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mmcblk0p7  26M   14K   23M   1% /custom
/dev/mmcblk0p8  5.8G  241M   5.2G   5% /userdata
overlay         5.8G  241M   5.2G   5% /
devtmpfs        1.9G   8.0K   1.9G   1% /dev
tmpfs           2.0G   16K   2.0G   1% /tmp
tmpfs           2.0G    0   2.0G   0% /dev/shm
tmpfs           2.0G   18M   1.9G   1% /run
tmpfs           5.0M   4.0K   5.0M   1% /run/lock
tmpfs           2.0G    0   2.0G   0% /sys/fs/cgroup
tmpfs           391M   4.0K  391M   1% /run/user/108
tmpfs           391M   4.0K  391M   1% /run/user/1001
```

6.7.3 查询产品型号信息

devinfo 工具可以查看产品的型号信息

```
edge@edge-computer:~$ sudo devinfo
model=EC942
alias=
serialnumber=CL9422343000019
partnumber=LQA8-W-G
```

6.7.4 调整时间

LGW/1TG/33G 有两个时间设置。一个是系统时间，另一个是由 LGW/1TG/33G 的硬件所保持的 RTC（实时时钟）时间。使用 `date` 命令来查询当前的系统时间或设置一个新的系统时间。使用 `hwclock` 命令查询当前的 RTC 时间或设置一个新的 RTC 时间。

使用命令 `date MMDDhhmmYYYY` 来设置系统时间：MM：月 DD：日 hh：时 mm：分 YYYY：年，使用以下命令可以将 RTC 时间设置为系统时间

```
edge@edge-computer:~$ sudo date 030115052023
Wed Mar  1 15:05:00 CST 2023
```

```
edge@edge-computer:~$ sudo hwclock
2023-03-01 15:05:55.192961+08:00
```

点击以下的链接来获取更多有关日期和时间的详细信息：

[https://www.debian.org/doc/manuals/system-administrator/ch-sysadmin-](https://www.debian.org/doc/manuals/system-administrator/ch-sysadmin-time.html)
[time.html https://wiki.debian.org/DateTime](https://wiki.debian.org/DateTime)

6.7.5 设置时区

有两种方法可以配置 LGW/1TG/33G 的时区。一个是使用命令 `tzselect`。另一个是使用 `/etc/localtime` 文件。

6.7.6 使用 `tzselect` 命令

当键入 `tzselect` 命令后，会进入到选择地区的界面先选择大致的区域（按大洲大洋划分）
输入大洲或大洋前面的数字

```
edge@edge-computer:~$ tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent, ocean, "coord", or "TZ".
1) Africa
2) Americas
3) Antarctica
4) Asia
5) Atlantic Ocean
6) Australia
7) Europe
8) Indian Ocean
9) Pacific Ocean
10) coord - I want to use geographical coordinates.
11) TZ - I want to specify the time zone using the Posix TZ format.
```


再选择该大洲或大洋下的国家

```
#? 4
Please select a country whose clocks agree with yours.
1) Afghanistan      18) Israel           35) Palestine
2) Armenia           19) Japan            36) Philippines
3) Azerbaijan        20) Jordan           37) Qatar
4) Bahrain           21) Kazakhstan       38) Russia
5) Bangladesh        22) Korea (North)    39) Saudi Arabia
6) Bhutan            23) Korea (South)    40) Singapore
7) Brunei            24) Kuwait           41) Sri Lanka
8) Cambodia          25) Kyrgyzstan       42) Syria
9) China             26) Laos             43) Taiwan
10) Cyprus           27) Lebanon          44) Tajikistan
11) East Timor       28) Macau            45) Thailand
12) Georgia          29) Malaysia         46) Turkmenistan
13) Hong Kong        30) Mongolia         47) United Arab Emirates
14) India            31) Myanmar (Burma)  48) Uzbekistan
15) Indonesia        32) Nepal            49) Vietnam
16) Iran             33) Oman             50) Yemen
17) Iraq             34) Pakistan
```

按照以上步骤获取到中国时区关键字 Asia/Shanghai，执行以下命令设置时区

```
#? 9
Please select one of the following time zone regions.
1) Beijing Time
2) Xinjiang Time
#? 1

The following information has been given:

    China
    Beijing Time

Therefore TZ='Asia/Shanghai' will be used.
Selected time is now:  Wed Mar  1 15:07:31 CST 2023.
Universal Time is now: Wed Mar  1 07:07:31 UTC 2023.
Is the above information OK?
1) Yes
2) No
#? Yes
Please enter a number in range.
#? 1

You can make this change permanent for yourself by appending the line
    TZ='Asia/Shanghai'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
Asia/Shanghai
```

6.7.7 使用 localtime 文件

本地时区存储在/etc/localtime 中，如果没有为 TZ 环境变量设置值，则被 GNU 库用于 C（glibc）。此文件或者是/usr/share/zoneinfo/file 的副本，或者是指向它的符号链接。若 LGW/1TG/33G 无法找到

/usr/share/zoneinfo/ 文件，请从该网站上下载您所需要的时区信息文件（<https://www.iana.org/time-zones>），并在 LGW/1TG/33G 中重新链接到本地时间文件。

注意：

成功下载所需要的时区信息文件后，进行解压，之后使用 zic 命令编译生成对应的二进制文件，生成的时区文件为“/usr/share/zoneinfo/自定义时区文件名”。

7 外设接口的高级配置

在本章中,我们将介绍基于 Arm 结构的边缘计算机 LGW/1TG/33G 的外设接口的高级配置。

7.1 串口

LGW/1TG/33G 有 8 个串口, 前 4 个串口支持 RS-232、RS-422 和 RS-485 多种串口模式, 默认模式为 RS-485, 使用 `ih_uart_ctl` 命令可以切换串口模式。后 4 个串口固定为 RS-485 模式。

P1 对应的设备节点是 `/dev/ttyCOM1` P2 对应的设备节点是 `/dev/ttyCOM2` P3 对应的设备节点是 `/dev/ttyCOM3` P4 对应的设备节点是 `/dev/ttyCOM4`

```
root@edge-computer:~# ih_uart_ctl
Usage:
    ih_uart_ctl [OPTIONS]

OPTIONS
    help
        Help info of how to use ih_uart_ctl command.

    port_num=<num>
        num:1,2,3,4
        1 --> /dev/ttyCOM1
        2 --> /dev/ttyCOM2
        3 --> /dev/ttyCOM3
        4 --> /dev/ttyCOM4
        show the port mode
        example: ih_uart_ctl port_num=1

    uart_mode=<num>
        num:0,1,2
        0 --> RS232
        1 --> RS485
        2 --> RS422
        set the uart port mode
        if you do not input port number, the default port num is 1
        example: ih_uart_ctl uart_mode=0
        example: ih_uart_ctl port_num=0 uart_mode=0
```

7.1.1 更改串口设置

用 `stty` 命令查看和设置串口

通过键入 `sudo stty --help` 查看详细命令内容:

```

edge@edge-computer:~$ sudo stty --help
Usage: stty [-F DEVICE | --file=DEVICE] [SETTING]...
or: stty [-F DEVICE | --file=DEVICE] [-a|--all]
or: stty [-F DEVICE | --file=DEVICE] [-g|--save]
Print or change terminal characteristics.

Mandatory arguments to long options are mandatory for short options too.
-a, --all          print all current settings in human-readable form
-g, --save         print all current settings in a stty-readable form
-F, --file=DEVICE  open and use the specified DEVICE instead of stdin
--help            display this help and exit
--version          output version information and exit

Optional - before SETTING indicates negation. An * marks non-POSIX
settings. The underlying system defines which settings are available.

Special characters:
* discard CHAR    CHAR will toggle discarding of output
eof CHAR          CHAR will send an end of file (terminate the input)
eol CHAR          CHAR will end the line
* eol2 CHAR       alternate CHAR for ending the line
erase CHAR        CHAR will erase the last character typed
intr CHAR         CHAR will send an interrupt signal
kill CHAR         CHAR will erase the current line
* lnext CHAR      CHAR will enter the next character quoted
quit CHAR         CHAR will send a quit signal
* rprnt CHAR      CHAR will redraw the current line
start CHAR        CHAR will restart the output after stopping it
stop CHAR         CHAR will stop the output
susp CHAR         CHAR will send a terminal stop signal
* swtch CHAR      CHAR will switch to a different shell layer
* werase CHAR     CHAR will erase the last word typed

Special settings:
N                set the input and output speeds to N bauds
* cols N         tell the kernel that the terminal has N columns
* columns N      same as cols N
* [-]drain       wait for transmission before applying settings (on by default)
ispeed N         set the input speed to N
* line N         use line discipline N
min N            with -icanon, set N characters minimum for a completed read
ospeed N         set the output speed to N
* rows N         tell the kernel that the terminal has N rows
* size           print the number of rows and columns according to the kernel
speed           print the terminal speed
time N          with -icanon, set read timeout of N tenths of a second

Control settings:
[-]clocal        disable modem control signals
[-]cread         allow input to be received
* [-]crtcts      enable RTS/CTS handshaking
csN              set character size to N bits, N in [5..8]
[-]cstopb        use two stop bits per character (one with '-')
[-]hup           send a hangup signal when the last process closes the tty
[-]hupcl        same as [-]hup

[-]parenb        generate parity bit in output and expect parity bit in input
[-]parodd        set odd parity (or even parity with '-')
* [-]cmspar      use "stick" (mark/space) parity

```

```

Input settings:
[-]brkint      breaks cause an interrupt signal
[-]icrnl       translate carriage return to newline
[-]ignbrk      ignore break characters
[-]igncr       ignore carriage return
[-]ignpar       ignore characters with parity errors
* [-]imaxbel    beep and do not flush a full input buffer on a character
[-]inlcr       translate newline to carriage return
[-]inpck       enable input parity checking
[-]istrip      clear high (8th) bit of input characters
* [-]iutf8      assume input characters are UTF-8 encoded
* [-]iucllc     translate uppercase characters to lowercase
* [-]ixany      let any character restart output, not only start character
[-]ixoff       enable sending of start/stop characters
[-]ixon        enable XON/XOFF flow control
[-]parmrk      mark parity errors (with a 255-0-character sequence)
[-]tandem      same as [-]ixoff

```

```

Output settings:
* bsN          backspace delay style, N in [0..1]
* crN          carriage return delay style, N in [0..3]
* ffN          form feed delay style, N in [0..1]
* nlN          newline delay style, N in [0..1]
* [-]ocrnln     translate carriage return to newline
* [-]ofdel      use delete characters for fill instead of NUL characters
* [-]ofill      use fill (padding) characters instead of timing for delays
* [-]olcuc      translate lowercase characters to uppercase
* [-]onlcr      translate newline to carriage return-newline
* [-]onlret     newline performs a carriage return
* [-]onocr      do not print carriage returns in the first column
* [-]opost      postprocess output
* tabN         horizontal tab delay style, N in [0..3]
* tabs         same as tab0
* -tabs        same as tab3
* vtN          vertical tab delay style, N in [0..1]

```

```

Local settings:
[-]crterase    echo erase characters as backspace-space-backspace
* crtkill      kill all line by obeying the echoprt and echoe settings
* -crtkill     kill all line by obeying the echoctl and echok settings
* [-]ctlecho    echo control characters in hat notation ('^c')
[-]echo        echo input characters
* [-]echoctl    same as [-]ctlecho
[-]echoe       same as [-]crterase
[-]echok       echo a newline after a kill character
* [-]echoke     same as [-]crtkill
[-]echonl      echo newline even if not echoing other characters
* [-]echoprt    echo erased characters backward, between '\' and '/'
* [-]extproc    enable "LINEMODE"; useful with high latency links
* [-]flusho     discard output
[-]icanon      enable special characters: erase, kill, werase, rpnt
[-]iexten      enable non-POSIX special characters
[-]isig        enable interrupt, quit, and suspend special characters

```

```

[-]noflsh      disable flushing after interrupt and quit special characters
* [-]prterase   same as [-]echoprt
* [-]tostop     stop background jobs that try to write to the terminal
* [-]xcase      with icanon, escape with '\' for uppercase characters

```

```

Combination settings:
* [-]LCASE      same as [-]lcase
cbreak         same as -icanon
-cbreak        same as icanon
cooked         same as brkint ignpar istrip icrnl ixon opost isig
               icanon, eof and eol characters to their default values
-cooked        same as raw
crt            same as echoe echoctl echoke
dec            same as echoe echoctl echoke -ixany intr ^c erase 0177
               kill ^u
* [-]decctlq    same as [-]ixany
ek             erase and kill characters to their default values
evenp         same as parenb -parodd cs7
-evenp        same as -parenb cs8
* [-]lcase      same as xcase iuclc olcuc
litout        same as -parenb -istrip -opost cs8
-litout       same as parenb istrip opost cs7
nl            same as -icrnl -onlcr
-nl           same as icrnl -inlcr -igncr onlcr -ocrnl -onlret
oddp         same as parenb parodd cs7
-oddp        same as -parenb cs8
[-]parity     same as [-]evenp
pass8        same as -parenb -istrip cs8
-pass8       same as parenb istrip cs7
raw          same as -ignbrk -brkint -ignpar -parmrk -inpck -istrip
               -inlcr -igncr -icrnl -ixon -ixoff -icanon -opost
               -isig -iuclc -ixany -imaxbel -xcase min 1 time 0
-raw         same as cooked
sane         same as cread -ignbrk brkint -inlcr -igncr icrnl
               icanon texten echo echoe echok -echonl -noflsh
               -ixoff -iutf8 -iuclc -ixany imaxbel -xcase -olcuc -ocrnl
               opost -ofill onlcr -onocr -onlret nl0 cr0 tab0 bs0 vt0 ff0
               isig -tostop -ofdel -echoprt echoctl echoke -extproc -flusho,
               all special characters to their default values

Handle the tty line connected to standard input. Without arguments,
prints baud rate, line discipline, and deviations from stty sane. In
settings, CHAR is taken literally, or coded as in ^c, 0x37, 0177 or
127; special values ^- or undef used to disable special characters.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report stty translation bugs to <https://translationproject.org/team/>
Full documentation at: <https://www.gnu.org/software/coreutils/stty>
or available locally via: 'info '(coreutils) stty invocation'

```

7.1.2 查看串口信息

```

edge@edge-computer:~$ sudo stty -a -F /dev/ttyCOM1
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S;
susp = ^Z; rprnt = ^R; werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 hupcl -cstpb cread clonal -crtscs
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff -iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt echoctl echoke -flusho -extproc
edge@edge-computer:~$

```

7.1.3 设置 COM1 串口的波特率

```
sudo stty -F /dev/ttyCOM1 ispeed 9600 ospeed 9600 cs8
```

7.1.4 设置 COM2 串口的波特率

```
sudo stty -F /dev/ttyCOM1 ispeed 9600 ospeed 9600 cs8
```

7.1.5 设置 COM3 串口的波特率

```
sudo stty -F /dev/ttyCOM3 ispeed 9600 ospeed 9600 cs8
```

7.1.6 设置 COM4 串口的波特率

```
sudo stty -F /dev/ttyCOM4 ispeed 9600 ospeed 9600 cs8
```

注意:

关于 stty 命令的详细信息可在以下链接中获得

<http://www.gnu.org/software/coreutils/manual/coreutils.html>

7.2 USB 接口

LGW/1TG/33G 提供两个 USB 2.0 Host 接口，主要用于扩展存储设备、接鼠标和键盘。

LGW/1TG/33G 支持 USB 存储设备热插拔。它会自动挂载所有的分区。LGW/1TG/33G 会把所有 USB 存储设备分区挂载到 /mnt/ 路径下，挂载文件夹的命名格式为 usb_<node>_<num>。其中，<node>是分区的设备节点名称，<num>可以是 0~9 的数字。

```
edge@edge-computer:~$ sudo mount|grep "/mnt"
overlay on /mnt type overlay (rw,relatime,lowerdir=/mnt,upperdir=/userdata/v1//mnt_
rw/upper,workdir=/userdata/v1//mnt_rw/work)
/dev/sda1 on /mnt/usb_sda1_0 type vfat (rw,nodev,noatime,fmask=0022,dmask=0022,code
page=936,iocharset=cp936,shortname=mixed,errors=remount-ro)
```

注意:

在断开 USB 大容量存储设备之前，请记住输入 sync 同步命令，以防止数据丢失。当您断开存储设备的连接时，请从 /media/* 目录下退出。如果您留在 /media/usb* 中，则自动卸载过程将会失败。如果发生这种情况，请键入 umount /media/usb* 以手动卸载设备

7.3 Micro SD

需要将 SD 卡的分区格式化为 FAT32 格式，格式化的方法如下：

- SD卡容量 ≤ 32GB：SD卡插入读卡器，然后将读卡器插入使用Windows系统的电脑。大部分Windows系统只支持将容量小于32GB的SD卡格式为FAT32系统，格式化的时候会自动创建一个分区；也可以在Linux系统下格式化为FAT32文件系统，方法同SD卡容量 > 32GB。
- SD卡容量 > 32GB：SD卡容量大于32GB时，需要在Linux下格式化FAT32格式。SD卡插入读卡器，然后将读卡器插入使用Linux系统的电脑或者安装虚拟机Linux虚拟机的电脑。
 - a. 执行fdisk -l查看SD卡对应的设备节点，示例中是/dev/sda

b. 执行 `fdisk /dev/sda` 进行分区 (p 查看当前分区, d 删除已有分区, n 新创建分区, w 保存更改), 创建分区 `/dev/sda1`

c. 执行 `sudo mkfs.vfat -F 32 /dev/sda1` 将分区格式化为 FAT32 格式。

d. 格式化之后, 将 sd 卡从读卡器上取下来, 在设备断电的情况下插入设备。

LGW/1TG/33G 支持 micro SD 存储卡不支持热插拔。它会自动挂载所有的分区。LGW/1TG/33G 会把所有 micro SD 存储卡分区挂载到 `/mnt/` 路径下, 挂载文件夹的命名格式为 `sd_<node>_<num>`。其中, `<node>` 是分区的设备节点名称, `<num>` 是 0~9 的数字。

```
edge@edge-computer:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        3.5G  2.7G  566M  83% /
devtmpfs         1.9G   8.0K   1.9G   1% /dev
/dev/mmcblk0p9   8.2G  473M   7.8G   6% /userdata
overlay          8.2G  473M   7.8G   6% /var
overlay          8.2G  473M   7.8G   6% /etc
overlay          8.2G  473M   7.8G   6% /home
overlay          8.2G  473M   7.8G   6% /root
overlay          8.2G  473M   7.8G   6% /sbin
overlay          8.2G  473M   7.8G   6% /bin
overlay          8.2G  473M   7.8G   6% /usr
overlay          8.2G  473M   7.8G   6% /lib
overlay          8.2G  473M   7.8G   6% /tmp
overlay          8.2G  473M   7.8G   6% /mnt
overlay          8.2G  473M   7.8G   6% /opt
overlay          8.2G  473M   7.8G   6% /media
overlay          8.2G  473M   7.8G   6% /system
overlay          8.2G  473M   7.8G   6% /boot
overlay          8.2G  473M   7.8G   6% /srv
overlay          8.2G  473M   7.8G   6% /vendor
tmpfs            1.9G   32K   1.9G   1% /dev/shm
tmpfs            1.9G   8.9M   1.9G   1% /run
tmpfs            5.0M   4.0K   5.0M   1% /run/lock
tmpfs            1.9G     0   1.9G   0% /sys/fs/cgroup
/dev/mmcblk0p7   126M   13M  107M  11% /oem
/dev/mmcblk1p2   3.7G  800M   2.8G  23% /mnt/sd_mmcblk1p2_0
/dev/mmcblk1p1   69M   26M   44M  37% /mnt/sd_mmcblk1p1_0
tmpfs            378M     0  378M   0% /run/user/108
tmpfs            378M     0  378M   0% /run/user/1001
```


7.4 mSATA 硬盘

(1) 登录系统，执行 `sudo fdisk -l`，找到自己硬盘分区，如下图是 `/dev/sda1`

```
edge@edge-computer:~$ sudo fdisk -l
Disk /dev/ram0: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes

Disk /dev/sda: 119.2 GiB, 128035676160 bytes, 250069680 sectors
Disk model: Lenovo SSD SL700
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x1a63708d

Device            Boot Start      End  Sectors  Size Id Type
/dev/sda1          2048 250069679 250067632 119.2G 83 Linux

Disk /dev/mmcblk0: 14.6 GiB, 15634268160 bytes, 30535680 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 902C0000-0000-4864-8000-6B9300002304

Device            Start      End  Sectors  Size Type
/dev/mmcblk0p1     16384    24575    8192     4M unknown
/dev/mmcblk0p2     24576    32767    8192     4M unknown
/dev/mmcblk0p3     32768    98303   65536    32M unknown
/dev/mmcblk0p4     98304   163839   65536    32M unknown
/dev/mmcblk0p5    163840   229375   65536    32M unknown
/dev/mmcblk0p6    229376  10715135 10485760    5G unknown
/dev/mmcblk0p7   10715136 10977279   262144   128M unknown
/dev/mmcblk0p8   10977280 13074431   2097152    1G unknown
/dev/mmcblk0p9   13074432 30535615 17461184   8.3G unknown
```

(2) 将分区格式化为需要使用的文件系统，比如ext4

```
edge@edge-computer:~$ sudo mkfs.ext4 /dev/sda1
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 31258454 4k blocks and 7815168 inodes
Filesystem UUID: 59e1a7cf-0044-4453-9886-518aee27fc2a
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

(3) 创建挂载点，比如 `/mnt/sda1`

(4) 编辑 `vi /etc/fstab` 文件，

在末尾行加上/dev/sda1 /mnt/sda1 ext4 defaults,

nofail,x-systemd.device-timeout=1s 0 0, 添加后如下图所示:

/dev/sda1: 设备分区, 需要用户根据实际情况进行配置

/mnt/sda1: 挂载点, 需要用户根据实际情况进行配置

ext4 硬盘分区的文件系统格式, 需要用户根据实际情况进行配置

defaults,nofail,x-systemd.device-timeout=1s 0 0 固定配置, 推荐使用这样的配置, 用户也可以根据需求自行修改。

```
UNCONFIGURED FSTAB FOR BASE SYSTEM
/dev/mmcblk0p7 /oem ext2 defaults 0 0
#/dev/mmcblk0p8 /userdata ext2 defaults 0 0
/dev/sda /mnt/sda ext4 defaults,nofail,x-systemd.device-timeout=60s 0 0
```

7.5 CAN 总线接口

LGW/1TG/33G 的 CAN 端口支持两个 CAN 总线, CAN1 对应设备 can0, CAN2 对应设备 can1

7.5.1 配置连接 CAN1 接口

默认情况下, CAN 端口将被初始化。如果需要任何其他配置, 请使用 ip 链接命令检查 CAN 设备。要检查 CAN 设备的状态, 请使用 ip 链接命令:

```
edge@edge-computer:~$ ip link show can0
2: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can
edge@edge-computer:~$
```

要配置 CAN 设备, 请使用 ip link set
can0 down 来先关闭设备 sudo ip link
set can0 down

然后配置比特率 (以下是 125k 比特率示例):

```
sudo ip link set can0 type can bitrate 125000
```

最后重新开启设备

```
sudo ip link set can0 up
```

7.5.2 配置连接 CAN2 接口

默认情况下, CAN 端口将被初始化。如果需要任何其他配置, 请使用 ip 链接命令检查 CAN 设备。要检查 CAN 设备的状态, 请使用 ip 链接命令:

```
edge@edge-computer:~$ ip link show can1
3: can1: <NOARP,ECHO> mtu 16 qdisc noop state DOWN mode DEFAULT group default qlen 10
    link/can
edge@edge-computer:~$
```

要配置 CAN 设备，请使用 `ip link set can1 down` 来先关闭设备

`sudo ip link set can1 down` 然后配置比特率（以下是 125k 比特率示例）：

`ip link set can1 type can bitrate 125000 dbitrate 1250000 fd on`

最后重新开启设备 `sudo ip link set can1 up`

7.6 IO 调试

LGW/1TG/33G 支持 4 路 DI 和 4 路 DO。当您要使用 IO 口的时候请键入 `dio_mgmt` 命令用于控制 io 的输入输出。`dio_mgmt` 的用法：

```
edge@edge-computer:~$ sudo dio_mgmt
Usage:
    dio_mgmt <OPTIONS>

OPTIONS
    help
        Help info of how to use dio_mgmt.
    show <DI0|DI1|DI2|DI3|DO0|DO1|DO2|DO3>
        Show the status of digital IO.
    set <DO0|DO1|DO2|DO3> <LOW|HIGH>
        Set the status of digital out IO.
edge@edge-computer:~$
```

当需要设置某位 IO 口为高或者低时，可以键入命令

`dio_mgmt set D<I/O><number><HIGH/LOW>`

```
edge@edge-computer:~$ sudo dio_mgmt set DO2 LOW
edge@edge-computer:~$ sudo dio_mgmt show DI0
LOW
edge@edge-computer:~$
```

通过键入 `dio_mgmt show D<I/O><number>` 打印对应的 IO 口的电平信息。

7.7 GPS

LGW/1TG/33G 部分型号内部集成 GPS 模块，数据串口节点为 `/dev/ttyS9`。如果要查看 GPS 的详细信息，有两种方式可以进行查看：

- 1、使用 `stty` 设置串口节点，键入 `cat` 直接输出源数据

```

edge@edge-computer:~$ sudo stty -F /dev/ttyS9 ispeed 115200 ospeed 115200 cs8
edge@edge-computer:~$ sudo cat /dev/ttyS9
$GNGGA,,,,,0,00,25.5,,,,,*64

$GNGLL,,,,,V,N*7A

$GPGSA,A,1,,,,,,,,,25.5,25.5,25.5*02

$BDGSA,A,1,,,,,,,,,25.5,25.5,25.5*13

$GLGSA,A,1,,,,,,,,,25.5,25.5,25.5*1E

$GPGSV,1,1,00*79

$BDGSV,1,1,00*68

$GLGSV,1,1,00*65

$GNRMC,,V,,,,,,,,,N*4D

$GNVTG,,,,,,,,N*2E

$GNZDA,,,,,*56

```

2、键入 gnss 命令直接输出解析后的时间、经纬度等信息

```

edge@edge-computer:~$ sudo gnss

```

7.8 开关机

7.8.1 关闭设备

您可以使用 Linux 命令来关闭设备上运行的所有软件并停止系统。但是，在运行此命令后，CPU、RAM 和存储设备等主要组件将掉电。

```

edge@edge-computer:~$ sudo shutdown -h now

```

8 安全

在本章中，我们将介绍基于 ARM 结构的边缘计算机 LGW/1TG/33G 的安全机制。

8.1 sudo 机制

在 LGW/1TG/33G 中，为了获得更好的安全性，根用户被禁止使用。Sudo 是一个程序，让系统管理员允许被同意的用户作为根用户或其他用户执行一些命令。最基本的准则是给予尽可能少的特权完成工作。使用 sudo 比使用 root 身份的会话打开更安全，原因有很多，包括：

- 不用知道root密码（sudo会提示当前用户的密码），可授予普通用户特权
- 通过sudo运行需要特权的命令很容易，其余时间，都作为无特权用户工作，从而降低由于错误操作可能造成的损害。
- 有些系统级命令不能直接对用户使用，如下示例输出所示：

```
edge@edge-computer:~$ ifconfig
-bash: ifconfig: command not found
edge@edge-computer:~$ sudo ifconfig
br-97e4c72399e1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.18.0.1 netmask 255.255.0.0 broadcast 172.18.255.255
    inet6 fe80::42:eff:fe07:c8aa prefixlen 64 scopeid 0x20<link>
    ether 02:42:0e:07:c8:aa txqueuelen 0 (Ethernet)
    RX packets 663 bytes 81645 (79.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 595 bytes 107290 (104.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:c4:58:eb:69 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

8.2 防火墙

Netfileter/iptables（以下简称 iptables）是 nuix/linux 系统自带的优秀且完全免费的基于包过滤的防火墙工具、它的功能十分强大、使用非常灵活、可以对流入、流出及流经服务器的数据包进行精细的控制。

8.3 TPM2.0

TPM 代表 “Trusted Platform Module”（可信平台模块），它是一种硬件安全模块，旨在提供计算机系统的安全和加密功能。它是一个安全微控制器，可以在计算机系统中嵌入或作为独立硬件设备出售。它包含一个加密协处理器，用于存储加密密钥、数字证书和其他安全数据，以及支持多种加密算法和安全协议。在 LGW/1TG/33G 上，已经集成了标准的 TPM2.0 协议栈和 TPM2.0 工具供用户使用。

9 系统恢复出厂设置及更新

在本章中，我们将介绍基于 Arm 结构的边缘计算机 LGW/1TG/33G 如何恢复出厂设置及更新。

9.1 恢复出厂设置

有两种方法可以恢复出厂设置：

- 1、通过键入命令，系统会自动重启并恢复出厂设置。

```
edge@edge-computer:~$ sudo update reset
```

- 2、通过按键恢复出厂设置：

- 长按恢复出厂设置按键10-20s，看到warn灯长亮。
- 当warn灯长亮以后，松开恢复出厂设置按键。
- 松开恢复出厂设置按键后，error灯闪烁几次后系统开始重启并执行恢复出厂设置
- 系统重启后，warn灯和error灯闪烁，status熄灭；大约30s后，当warn灯和error灯都停止闪烁，同时status开始闪烁，则系统完成恢复出厂设置。

9.2 系统升级

准备一个 U 盘(micro SD 卡)。如果该 U 盘(SD 卡)有多个分区，请使用第一个分区。建议不做多个分区。U 盘(SD 卡)分区需要格式为 FAT32 格式。本文档以升级 LGW/1TG/33G-V2.0.0.img 为例进行说明。

- 在U盘（SD卡）根目录下创建一个空的LGW/1TG/33G_img目录，将公司发布的 LGW/1TG/33G-V2.0.0.img 文件和md5.txt文件放到LGW/1TG/33G_img目录下。
- 确认md5.txt 文件中仅有 LGW/1TG/33G-V2.0.0.img 一行的 md5 哈希值。

LGW/1TG/33G不支持多个img镜像ota升级。

- 在电脑上正常退出U盘（SD卡）。注意不要直接拔u盘，要在桌面选择“退出”或“弹出”操作。
- 将U盘（SD卡）插入目标LGW/1TG/33G设备。目标设备会自动校验LGW/1TG/33G-V2.0.0.img文件，并进行OTA升级。升级过程中WARN和ERROR灯会有相应的显示。等WARN和ERROR恢复正常后，升级操作即完成。因为img文件较大，升级时间较长，请耐心等待。
- 升级结束后，LGW/1TG/33G会把升级中的关键信息写入LGW/1TG/33G_img目录下的日志文件。请查看相关文件。

10 编程指南

LGW/1TG/33G 提供了一个 JSON 格式的设备信息描述文件，需要操作 IO，LED，串口等外设的客户，可以通过查询设备描述信息文件获取这些外设的设备节点信息。

设备描述信息文件路径：/tmp/ieos/etc/system_info.json，内容如下：

```
{
  "device_info": { "model_info": {
    "model": "XXXXX",
    "pn": "I0-FQ58", "sn": "xxxxxxxxxxxxxxxx",
    "oem": "xxxxx",
    "features": ";io;cell-FQ58;wlan;ble;gps;"
  },
  "software_info": { "boot_loader": "1.0.2",
    "kernel": "5.10.160-00001-g406d1811beab-dirty", "version": "V2.0.0-beta.1",
    "OS": "Debian GNU/Linux 11 (bullseye)"
  },
  "hardware_info": { "arch": "arm64",
    "soc": "rk3568", "interface": {
      "eth": [
        {
          "iface_name": "eth1", "iface_mac": "00:33:44:11:00:01"
        },
        {
          "iface_name": "eth2", "iface_mac": "00:33:44:11:00:02"
        }
      ]
    }
  }
}
```

```
{
  "iface_name": "eth3", "iface_mac": "00:33:44:11:00:03"
},
{
  "iface_name": "eth4", "iface_mac": "00:33:44:11:00:04"
}
],
"wlan": [
  {
    "iface_name": "wlan0", "iface_mac": "94:A4:08:8A:30:CD"
  },
  {
    "iface_name": "wlan1", "iface_mac": "96:A4:08:8A:30:CD"
  }
],
"ble": [
]
},
"gpio": [
  {
    "gpio_name": "cellular_power", "dev_node": "/sys/class/gpio/gpio0"
  },
  {
    "gpio_name": "sim_switch", "dev_node": "/sys/class/gpio/gpio19"
  },
  {
    "gpio_name": "msata_power", "dev_node": "/sys/class/gpio/gpio20"
  },
  {
    "gpio_name": "gnss_power", "dev_node": "/sys/class/gpio/gpio110"
  },
  {
    "gpio_name": "ble_power", "dev_node": "/sys/class/gpio/gpio220"
  }
],
"user_key": [
```

```
{
  "user_key_name": "USER", "dev_node": "/sys/class/gpio/gpio95"
}
],
"uart": [
  {
    "uart_name": "P1", "dev_node": "/dev/tty01"
  },
  {
    "uart_name": "P2", "dev_node": "/dev/tty02"
  },
  {
    "uart_name": "P3", "dev_node": "/dev/tty03"
  },
  {
    "uart_name": "P4", "dev_node": "/dev/tty04"
  },
  {
    "uart_name": "P5", "dev_node": "/dev/tty05"
  },
  {
    "uart_name": "P6", "dev_node": "/dev/tty06"
  },
  {
    "uart_name": "P7", "dev_node": "/dev/tty07"
  },
  {
    "uart_name": "P7", "dev_node": "/dev/tty08"
  }
],

"io": {
  "di": [
    {
      "di_name": "DI1", "dev_node": "/sys/class/gpio/gpio487"
    },

```

```
{
  "di_name": "DI2", "dev_node": "/sys/class/gpio/gpio488"
},
{
  "di_name": "DI3", "dev_node": "/sys/class/gpio/gpio489"
},
{
  "di_name": "DI4", "dev_node": "/sys/class/gpio/gpio490"
}
],
"do": [
  {
    "di_name": "D01", "dev_node": "/sys/class/gpio/gpio491"
  },
  {
    "di_name": "D02", "dev_node": "/sys/class/gpio/gpio492"
  },
  {
    "di_name": "D03", "dev_node": "/sys/class/gpio/gpio493"
  },
  {
    "di_name": "D04", "dev_node": "/sys/class/gpio/gpio494"
  }
],
"led": [
  {
    "led_name": "USER1", "dev_node": "/sys/class/leds/user1"
  },
  {
    "led_name": "USER2", "dev_node": "/sys/class/leds/user2"
  },
  {
    "led_name": "4G/5G", "dev_node": "/sys/class/leds/cell"
  },
  {
```

```
"led_name": "SIM1", "dev_node": "/sys/class/leds/sim1"
},
{
"led_name": "SIM2", "dev_node": "/sys/class/leds/sim2"
},
{
"led_name": "WARN", "dev_node": "/sys/class/leds/warn"
},
{
"led_name": "ERROR", "dev_node": "/sys/class/leds/error"

},
{
"led_name": "STATUS", "dev_node": "/sys/class/leds/status"
},
{
"led_name": "L1", "dev_node": "/sys/class/leds/level1"
},
{
"led_name": "L2", "dev_node": "/sys/class/leds/level2"
},
{
"led_name": "L3", "dev_node": "/sys/class/leds/level3"
}
]
}
}
}
```

10.1 IO 编程指南

目前设备上一共 12 个 IO 接口:如设备面板上 DI0~DI3 为 4 个输入管脚;DO0~DO3 为 4 个输出管脚。根据设备描述信息文件/tmp/ieos/etc/system_info.json 可以获取到 IO 设备节点为:

10.1.1 DI0~DI3 `sys/class/gpio/gpio487~sys/class/gpio/gpio490`

D00~D03----- `sys/class/gpio/gpio491~sys/class/gpio/gpio494`

当您需要对 IO 接口进行编程时，直接操作后台设备节点下面的 value 值即可

10.1.2 (`sys/class/gpio/gpioxxx/value`)

案例：

当 D00 需要输出高电平时，可以直接向 `sys/class/gpio/gpio491/value` 写入 1

```
echo 1 > /sys/class/gpio/gpio491/value
```

当需要查看 DI0 的电平是，同样可以直接查看 `sys/class/gpio/gpio487/value` 的值

```
cat /sys/class/gpio/gpio487/value
```

完整 shell 脚本：

```
#!/bin/bash
```

```
gpio491="/sys/class/gpio/gpio491/value"
```

```
gpio492="/sys/class/gpio/gpio492/value"
```

```
gpio493="/sys/class/gpio/gpio493/value"
```

```
gpio494="/sys/class/gpio/gpio494/value"
```

```
# 当 D00 需要输出高电平时，可以直接向 sys/class/gpio/gpio491/value 写入 1
```

```
if [ -f "$gpio491" ]; then
```

```
echo 1 > /sys/class/gpio/gpio491/value else
```

```
echo "no file exit "$gpio491
```

```
fi
```

```
# 当 D01 需要输出低电平时，可以直接向 sys/class/gpio/gpio491/value 写入 1
```

```
if [ -f "$gpio492" ]; then echo 0 > $gpio492
```

```
else
```

```
echo "no file exit "$gpio492
```

```
fi
```

```
gpio487="/sys/class/gpio/gpio487/value"
```

```
gpio488="/sys/class/gpio/gpio488/value"
```

```
gpio489="/sys/class/gpio/gpio489/value"
```

```
gpio490="/sys/class/gpio/gpio490/value"
```

```
# 当需要查看 DI0 的电平是，同样可以直接查看 sys/class/gpio/gpio487/value 的值
```

```
if [ -f "$gpio487" ]; then cat $gpio487
```

```
else  
  
echo "no file exit "$gpio487  
  
fi
```

10.2Led 编程指南

设备上用户可以使用 USER1 和 USER2 两个灯进行状态提示,请查看灯签确认 USER1 和 USER2 两个灯的位置。

根据设备描述信息文件/tmp/ieos/etc/system_info.json 可以获取到 USER1 和 USER2 设备节点为:

```
user1: /sys/class/leds/user1 user2: /sys/class/leds/user2
```

/sys/class/leds/user1 目录下有一些控制文件用于控制 LED 的属性和状态:

/sys/class/leds/user1/brightness: 该文件用于控制 USER1 灯的亮或者灭。

写入 1 即是常亮, 写入 0 即是常灭。

/sys/class/leds/user1/trigger: LED 灯的触发器, 可写入 timer 表示定时器触发, 写入 none 表示取消触发器。

/sys/class/leds/user1/delay_on: 该文件表示 led 灯亮的时间, 以 ms 为单位。

/sys/class/leds/user1/delay_off: 该文件表示 led 灯灭的时间, 以 ms 为单位。

若配置了 trigger 为定时触发, 则 brightness 中的值将不再生效, 会自动变为 0。将文件路径中 user1 替换为 user2 即为控制 USER2 灯的操作。

案例:

当需要 USER1 灯常亮时, 将 1 写入 brightness 文件

```
echo 1 > /sys/class/leds/user1/brightness
```

当需要 USER1 灯闪烁时, 将 timer 写入 trigger 文件, 通过 delay_on 和 delay_off 控制亮和灭的时间

开始定时器

```
echo timer > /sys/class/leds/user1/trigger
```

亮 1 秒钟

```
echo 1000 > /sys/class/leds/user1/delay_on
```

灭 1 秒钟

```
echo 1000 > /sys/class/leds/user1/delay_off
```

完整 shell 脚本:

```
#!/bin/bash
```

```
USER1_BRIGHTHNESS="/sys/class/leds/user1/brightness"      USER1_TRIGGER="/sys/class/leds/user1/trigger"  
USER1_DELAY_ON="/sys/class/leds/user1/delay_on"  USER1_DELAY_OFF="/sys/class/leds/user1/delay_off"
```

```
USER2_BRIGHTHNESS="/sys/class/leds/user2/brightness"      USER2_TRIGGER="/sys/class/leds/user2/trigger"  
USER2_DELAY_ON="/sys/class/leds/user2/delay_on"  USER2_DELAY_OFF="/sys/class/leds/user2/delay_off"
```

```
# 点亮 USER1 灯
```

```
if [ -f "$USER1_BRIGHTHNESS" ]; then echo 1 > $USER1_BRIGHTHNESS
```

```
else
```

```
echo "no file exit "$USER1_BRIGTHNESS
fi

# 点亮 USER2 灯
if [ -f "$USER2_BRIGTHNESS" ]; then echo 1 > $USER2_BRIGTHNESS
else
echo "no file exit "$USER2_BRIGTHNESS
fi

# 设置 USER1 灯闪烁
if [ -f "$USER1_TRIGGER" ]; then echo timer > $USER1_TRIGGER
else
echo "no file exit "$USER1_TRIGGER
fi

# 设置 USER2 灯闪烁
if [ -f "$USER2_TRIGGER" ]; then echo timer > $USER2_TRIGGER
else
echo "no file exit "$USER2_TRIGGER
fi

# 设置 USER1 灯亮 1000ms
if [ -f "$USER1_DELAY_ON" ]; then echo 1000 > $USER1_DELAY_ON
else
echo "no file exit "$USER1_DELAY_ON
fi

# 设置 USER1 灯灭 1000ms
if [ -f "$USER1_DELAY_OFF" ]; then echo 1000 > $USER1_DELAY_OFF
else
echo "no file exit "$USER1_DELAY_OFF
fi

# 设置 USER2 灯亮 1000ms
if [ -f "$USER2_DELAY_ON" ]; then echo 1000 > $USER2_DELAY_ON
```

```
else

echo "no file exit "$USER2_DELAY_ON

fi

# 设置 USER2 灯灭 1000ms

if [ -f "$USER2_DELAY_OFF" ]; then echo 1000 > $USER2_DELAY_OFF

else

echo "no file exit "$USER2_DELAY_OFF

fi

# 关闭 USER1 闪烁

if [ -f "$USER1_TRIGGER" ]; then echo none > $USER1_TRIGGER

else

echo "no file exit "$USER1_TRIGGER

fi

# 关闭 USER2 闪烁

if [ -f "$USER2_TRIGGER" ]; then
```

10.3 交叉编译

用户自己的 c/c++ 程序可以通过在开发机器上使用交叉编译工具链进行交叉编译，然后将目标文件上传到 LGW/1TG/33G 设备上执行。

交叉编译工具压缩包：gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu.tar.gz 为交叉编译工具链配置环境变量的方法：

1、在开发机上将 gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu.tar.gz 解压到 /opt 路径下 (也可以解压到其他路径，步骤 2 中设置 PATH 环境变量的时候做相应的调整)

2、编辑 ~/.bashrc 文件，在文件最后添加一行 PATH=\$PATH:/opt/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu/bin

3、执行 source ~/.bashrc 使环境变量在当前终端生效；新打开的终端会自动生效。以经典的 hello world 程序为示例进行说明，创建如下目录和文件

```
mkdir ~/example
```

```
touch ~/example/hello.c touch ~/example/Makefile
```

~/example/hello.c 文件的内容如下：

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
printf("hello, world!
```

```
");
```

```
return 0;
```

```
}
```

~/example/Makefile 文件的内容如下:

```
# 定义目标文件名和源文件名

TARGET := hellworld

DIRS   := $(shell find . -maxdepth 3 -type d)

SRCS   := $(foreach dir,$(DIRS),$(wildcard $(dir)/*.c)) OBJS := $(SRCS:.c=.o)

CC=aarch64-linux-gnu-gcc # 定义编译器和编译选项

CFLAGS := -Wall -Wextra -g -Wno-unused-parameter


# 定义默认目标

all: $(TARGET)


# 定义目标文件的依赖关系和编译命令

$(TARGET): $(OBJS)

$(CC) $(CFLAGS) $(LIBS) $^ -o $@


# 定义将源文件编译成目标文件的命令

%.o: %.c

$(CC) $(CFLAGS) $(LIBS) -c $< -o $@


# 定义清除临时文件的命令

clean:

rm -f $(TARGET) $(OBJS)


# 声明伪目标 ".PHONY"

.PHONY: all clean
```

在~/example 目录下执行 make, 生成目标文件 helloworld

10.4 AI 算力卡

LGW/1TG/33G 部分型号内置了 Hailo-8 算力模组，系统预装了版本号为 4.16.0 的驱动和软件包，用户可以参考 Hailo 官网的文档进行开发：<https://hailo.ai/developer-zone>

10.5 编译内核模块

用户可以使用交叉编译或在设备上直接编译内核模块

10.5.1 在设备上直接编译

使用 `sudo -s` 命令切换到 root 用户

- 安装编译环境
 - `apt-get update`
 - `apt-get install -y build-essential`
 - `ln -s /usr/bin/python3 /usr/bin/python`
- 创建文件 `hello.c`

```
#include <linux/module.h> #include <linux/kernel.h>
```

```
int init_module(void) {  
    printk(KERN_INFO "Hello, world - this is the kernel speaking  
    ");  
    return 0;  
}
```

```
void cleanup_module(void) {  
    printk(KERN_INFO "Goodbye, world - leaving the kernel  
    ");  
}
```

-
- 创建文件Makefile
 - 执行make命令编译模块，会生成文件hello.ko
 - 执行insmod hello.ko安装模块

1.1.2 交叉编译

- 安装编译环境
 - 已测试过的系统：Ubuntu22.04
 - 解压交叉编译工具链：tar -xf gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu.tar.gz -C /opt/
 - 创建目录：mkdir linux-headers
 - 解压内核编译环境：tar -xf linux-headers.tar -C linux-headers
- 创建文件hello.c

```
#include <linux/module.h> #include <linux/kernel.h>

int init_module(void) {
    printk(KERN_INFO "Hello, world - this is the kernel speaking\n");
    return 0;
}

void cleanup_module(void) {
    printk(KERN_INFO "Goodbye, world - leaving the kernel\n");
}

MODULE_LICENSE("GPL");
```

- 创建文件Makefile

```
obj-m += hello.o

all:
    make -C linux-headers M=$(PWD) CROSS_COMPILE=/opt/gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-g

clean:
    make -C linux-headers M=$(PWD) CROSS_COMPILE=/opt/gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-g
```

- 执行make命令编译模块，会生成文件hello.ko，将此文件传输到设备中安装

上海良信电器股份有限公司
上海市浦东新区申江南路 2000 号
E/liangxin@sh-liangxin.com
T/021-68586699
F/021-23025796