

# G6 系列网关使用说明书

欢迎您选择并使用 Alinkwise [G6]系列 LoRaWAN 网关!

本快速使用说明书旨在为您提供一个清晰、简洁的指引，帮助您在最短的时间内完成网关的硬件安装、网络配置及基础操作。无论您是经验丰富的物联网开发者，还是初次接触 LoRaWAN 技术的新手，都可以通过本指南轻松上手，快速搭建稳定、可靠的物联网数据通道。

在开始之前，请确保您已准备好所有必要的设备和网络环境。让我们一起开启高效、便捷的物联网连接之旅!

## ALINKWISE LoRaWAN智能网关

极致连接 · 智能物联 · 高效集成

### 标准LoRaWAN(1.0.4)协议

远程连接Chirpstack、TTN、腾讯连连等标准LoRaWAN 服务器

### 全球 LoRaWAN频段支持

CN470\EU433\IN865\EU868\US915\AU915\AS923 标准频段

### 内置 LoRaWAN NS

无需搭建 外部NS 服务器  
开箱即用

### 三重网络链路模式

以太网 \ WIFI 热点 \ 4G 蜂窝网  
支持后备网络切换

### 标准 数据接口

MQTT / HTTP / Websocket /  
TCP 方式数据获取

### 标准 MODBUS 支持

MODBUS -TCP \ MODBUS -RTU  
标准协议数据直连

### 智能可编程™扩展功能

自定义 API 数据格式 / 终端荷载JSON转码/  
终端状态联动 / 定时数据采集任务

### Alinkwise可视化 OS

网关/终端通信实时监控  
支持手机/平板/电脑三端使用



# 设备简介 (Product Introduction)

**G6 系列网关** 是专为室内物联网应用打造的智能 LoRaWAN 网关。它深度集成了 **LoRaWAN 网关 + Network Server (NS) + Application Server (AS)** 功能于一体。本产品不仅支持标准的 **透传/转发模式** (连接 TTN、ChirpStack 等云平台)，同时支持了 **内置解码模式 (内置 NS)**，赋予网关强大的边缘计算能力。用户无需部署复杂的服务器即可在本地完成数据解析、逻辑控制和协议转换，是构建高效、灵活物联网系统的理想核心设备。

## 核心功能与技术优势

- **本地/云端传输数据，灵活部署：**
  - **远程 NS 模式：** 作为标准网关，无缝对接第三方主流 LoRaWAN 平台。
  - **内置 NS 模式：** 内置数据处理服务，直接在网关内完成数据解密与解析，输出用户可读的 JSON 数据。
- **强大的边缘计算与数据接口：**
  - **多协议 API：** 提供丰富类型的 API 接口，支持 **MQTT、HTTP、TCP、WebSocket** 等主流物联网协议，并可通过 JavaScript 脚本自定义报文格式。
  - **JS 本地联动控制：** 支持多终端设备间的联动控制（如传感器触发开关），逻辑在网关本地运行，无需云端参与，响应更快更稳定。
  - **JS 脚本编码：** 支持用户编写 JavaScript 脚本自定义数据解码规则。
  - **定时自动采集：** 支持主动向 Class C 终端下发采集指令，实现定时轮询。
- **Modbus TCP/RTU 直连采集：**
  - **全面支持 Modbus 映射：** 独有的寄存器映射功能，可将 LoRaWAN 无线数据映射为标准 Modbus 寄存器设备。
  - **Modbus-TCP 直连：** 独有的寄存器映射功能，可将 LoRaWAN 无线数据映射为 Modbus 寄存器，通过以太网/WIFI 轻松接入传统 SCADA、PLC 或组态软件。
  - **Modbus-RTU 直连：** 支持通过 RS485 接口直接连接 PLC 或工业显示屏，无需网络转换即可获取标准协议数据，完美适配传统工业现场。
- **多链路网络保障：**
  - 支持 **以太网、Wi-Fi、4G 三种联网方式**，并支持优先级配置（如断网自动切换 4G），保障数据传输稳定可靠。

## 适用应用场景

- **工业物联网 (IIoT)：** 利用 Modbus RTU/TCP 功能，通过 RS485 或局域网将无线传感器无缝集成到 PLC、HMI 或工业控制系统中。
- **智能楼宇与办公：** 快速搭建温湿度监测、智能照明与空调控制网络。
- **中小型独立网络：** 无需云服务器，在局域网或离线环境下即可独立运行完整的 LoRaWAN 网络。
- **快速原型开发：** 开发者无需搭建复杂后台，仅凭一台网关即可完成端到端的系统验证。

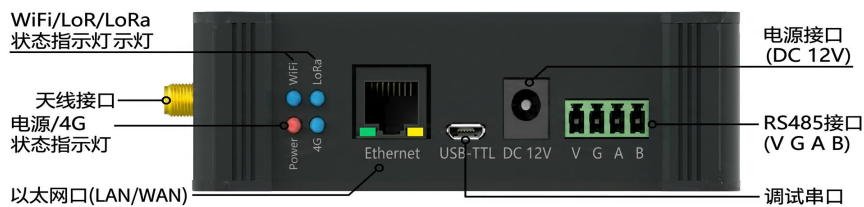
## 包装清单 (Package List)

打开包装盒后，请对照下表仔细核对箱内物品。如有缺失或损坏，请立即联系您的设备供应商。

### 发货清单

 <p><b>1. 网关主体 - 1台:</b> G6系列网关主机</p>	 <p><b>2. LoRa吸盘天线 - 1根:</b> 用于LoRaWAN通信，底部带吸盘</p>	 <p><b>3. WiFi吸盘天线 - 1根:</b> 用于WiFi通信，底部带吸盘</p>	 <p><b>4. 4G吸盘天线 - 1根:</b> (选配) 仅限选购了4G功能的型号提供</p>
 <p><b>5. 电源适配器 - 1个:</b> 标配DC 12V电源适配器 (注: POE电源为选配项)</p>	 <p><b>6. 壁挂安装套件 - 1套:</b> 包含挂耳及安装螺丝/膨胀螺钉</p>	 <p><b>7. 合格证/保修卡图片</b> 1份: 产品出厂合格证明及保修证</p>	

## 电气接口及指示灯(Hardware Details)



接口标识	名称	描述
Ethernet	以太网口	RJ45 接口，用于连接有线局域网/互联网。
DC 12V	电源接口	标准 DC 圆孔插座，连接标配的 12V/2A 电源适配器。
V G A B	RS485/电源端子	4 Pin 工业接线端子，定义如下： • V: DC 12V • G: GND (地) • A: RS485_A (+) • B: RS485_B (-) (主要用于连接 Modbus RTU 设备)
USB-TTL	调试串口	Micro-USB 接口，用于底层调试（需使用带串口芯片的数据线）。
Antenna	天线接口	SMA 接口，用于连接 LoRa、WiFi 及 4G 天线（请根据标签对应连接）。

### LED 指示灯状态参考

指示灯图标与名称	颜色与状态	含义	详细说明
 电源	红灯常亮	供电正常	检测到设备正常运行。
	关闭	电压不足/未供电/损坏	检查电源或设备状态。
 WiFi	蓝灯闪烁	模块启动中	无线模块正在初始化。
	蓝灯蓝色	Wi-Fi成功启动/正常工作	连接已建立并处于活动状态。
	关闭	自动失激/未启用	模块初始化失败已禁用。
 LoRa	常亮色色	射频模块成功启动/ 正常工作	LoRa射频模块已激活。
	关闭	启动中/启动失败	模块正在初始化或遇到错误。
 4G	慢闪（短亮，长灭）	正在搜索4G网络	亮0.2秒 / 灭1.8秒。
	慢闪（长亮，短灭）	网络已注册/待机中	亮1.8秒 / 灭0.2秒。
	快速闪烁	正在执行4G数据传输	数据传输正在进行中。

# 硬件安装指南 (Hardware Installation)

## 连接天线与 SIM 卡 (Antenna & SIM Installation)

正确安装天线是保障网关信号覆盖范围的关键，请务必按照标识对应安装，切勿接反。

### 1. 安装天线

请检查天线吸盘底部或胶棒上的标签，将其拧紧到网关侧面对应的 SMA 接口上：

- **LoRa 天线 (关键) :**
  - **识别标识:** 吸盘底部标有 “LoRa” 字样；胶棒天线上标有 “TX-XXX” (XXX 代表频率) 字样。
  - **安装位置:** 连接到网关标有 LoRa 的接口。
- **WiFi 天线:**
  - **识别标识:** 吸盘底部标有 “2.4G” 或 “WiFi” 字样；胶棒天线上标有 “WiFi” 字样。
  - **安装位置:** 连接到网关标有 WiFi 的接口。
- **4G 天线 (选配) :**
  - 仅在您选购了 4G 版本时提供。请将其连接到网关标有 4G/LTE 的接口。

**注意:** 请确保天线接头已拧紧，避免接触不良导致信号衰减。建议将吸盘天线吸附在较高的位置以获得更好的覆盖效果。

### 2. 插入 SIM 卡 (仅限 4G 版)

如果您需要使用 4G 蜂窝网络进行回传，请在通电前安装 SIM 卡。

- **卡槽位置:** 位于网关机身侧面。
- **SIM 卡规格:** 请使用 **标准大卡 (Standard SIM)**。
- **操作:** 将芯片朝向正确方向推入卡槽，直到听到 “咔哒” 一声锁定。

(注: 如果您未选配 4G 功能或仅使用有线/WiFi 联网, 请忽略此步骤)

## 连接电源与网络 (Power & Network Connection)

### 1. 连接电源

网关支持 DC 适配器供电 (标准) 或 PoE 供电 (选配), 请根据您的设备型号选择:

- **使用 DC 电源适配器 (标准方式) :**
  1. 取出包装盒内的 **12V 电源适配器**。
  2. 将圆形插头插入网关标有 “**DC 12V**” 的电源接口。

3. 将适配器接入 100-240V 交流电源插座。

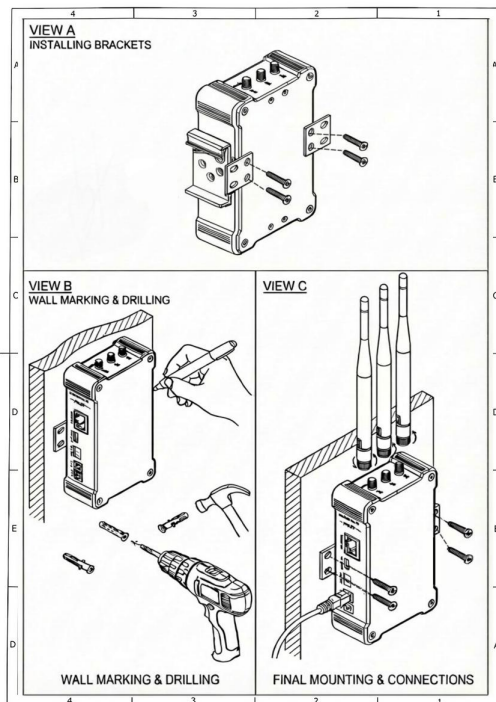
● **使用 PoE 供电 (仅限选配型号) :**

- 如果您的网关支持 PoE 功能且您拥有 PoE 交换机/供电模块, 可直接通过网线连接 Ethernet 口进行供电, 无需连接 DC 适配器。

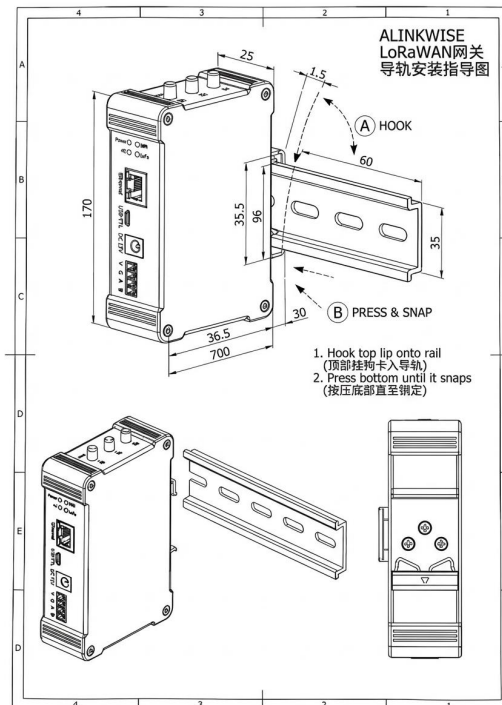
**上电状态检查:**

接通电源后, 请观察面板上的 Power 指示灯。若指示灯呈 **红色常亮**, 表示供电正常。

## 网关安装方式 (Installation method)



A 挂壁安装方式



B 导轨安装方式

# 网关登录与通用设置 (Initial Login & Settings)

## 首次访问网关 (First Access)

用户可以通过**以太网**或**WIFI**登录网关的操作界面进行配置和调试。由于通过网线接入路由器时，网关通常会动态获取一个未知的 IP 地址。为了方便您快速访问，我们推荐您首次使用时，**通过网关自带的 Wi-Fi AP 热点**直接连接并进入管理后台。

请按照以下步骤操作：

### 第一步：连接 Wi-Fi 热点

请打开电脑或手机的无线网络设置，搜索并连接 G6 网关发出的默认 wifi 热点：

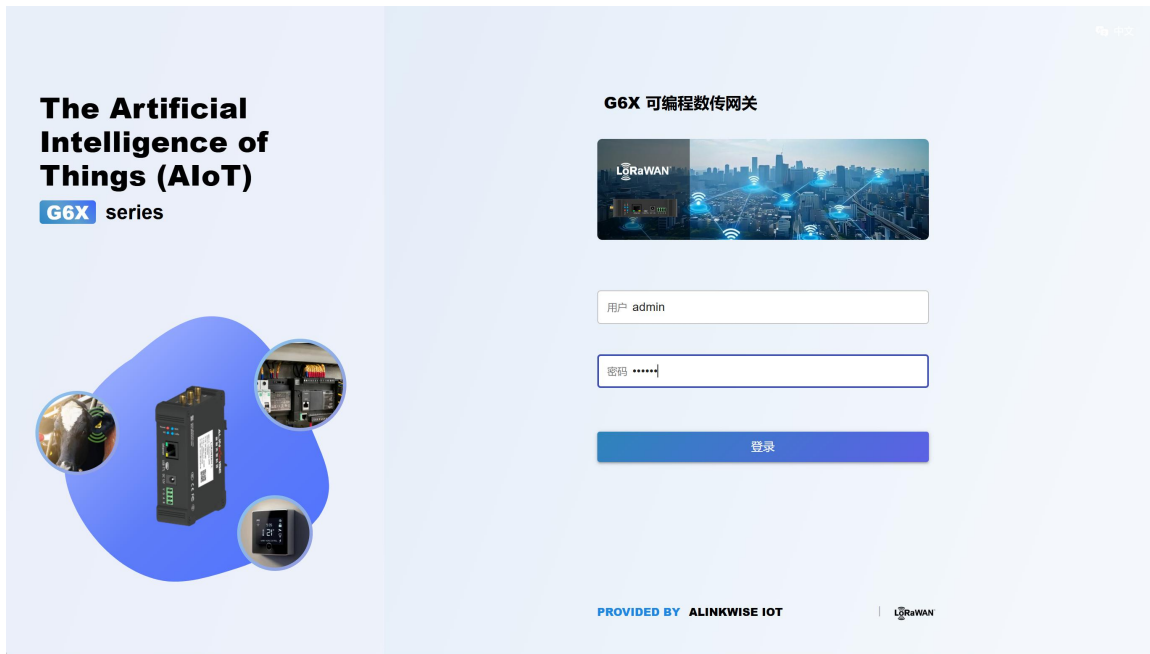
- **WIFI 热点名称 (SSID): Alinkwise\_G6\_xxxx**
  - (注: xxxx 为网关 EUI 地址的后四位, 您可以在设备标签上找到)
- **WIFI 默认密码: 12345678**



### 第二步：登录管理界面

连接成功后，网关会为您分配一个内部 IP。请打开浏览器（推荐使用 Google Chrome），进行以下操作：

1. 在浏览器地址栏输入网关的固定管理地址（注意包含端口号）：
  - 192.168.188.251:8080
2. 在弹出的登录界面中输入默认账户信息：
  - 用户名: admin
  - 密码: 123456
3. 点击“登录”，即可进入网关可视化管理后台。



## 网关通用配置 (General Configuration)

1. 此步骤共有三项配置：
2. 无线射频配置（网关和节点之间建立射频通信）
3. 网络连接配置（用于网关和应用服务器建立通信）
4. 数据处理与传输（用于确定网关的模式，方便后续操作）

## 无线射频配置 (Radio)



您可以在“设置”->“无线频段”页面对网关的 LoRaWAN 射频参数进行配置。正确的无线射频配置是确保网关能成功接收终端设备（节点）数据的关键。

## 网关频段 (Frequency Plan)

5. 首先，您需要选择一个正确的“无线频段”，网关和传感器才能在同一个“区域”通信。网关支持的频段由内部射频硬件决定。您必须在购买设备前，就确认并选择符合您部署区域法规的硬件型号。

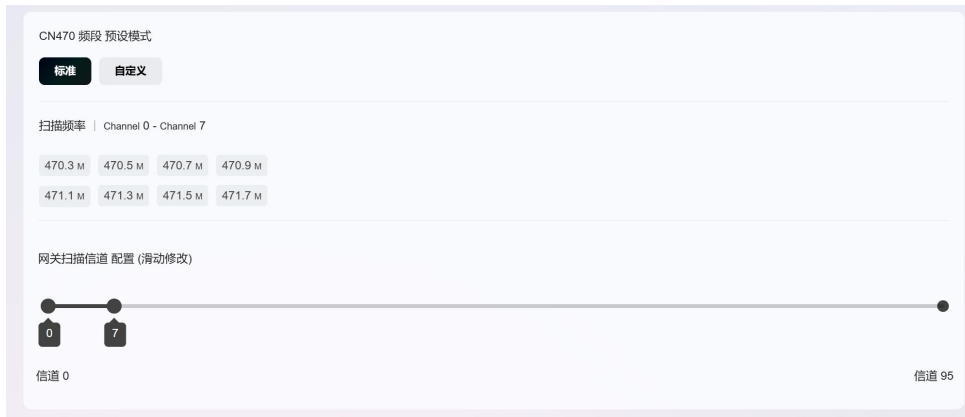


- **频段选择：**界面上会列出支持的所有工作频段（例如：CN470、EU433、IN865、EU868 等）。由于受硬件限制，请勿随意切换工作频段，如需切换可联系售后人员。

## 网关频率&信道 (Frequencies & Channels)

6. 我们需要配置网关具体要收听哪些“信道”（Channel）的节点数据。每个信道对应一个具体的频率（MHz），信道是网关和节点之间通信的射频参数，需要设置为一致。

- **标准模式 (Standard):**
  - **这是绝大多数场景会使用的模式，强烈推荐您使用！**
  - 在此模式下，网关会自动计算一组连续的（8个/16个）标准信道。例如，在 CN470 下，它会默认使用 0 到 7 共 8 个信道。您可以通过滑动滑块来快速选择需要监听信道范围。
  - 扫描频率：展示所选每个监听信道对应的详细频率（MHZ）。



- **自定义模式 (Custom):** 此模式允许您手动输入特定的频率值, 适用于非标或私有化的网络部署。切换到自定义模式后, 您会看到一个名为 “自定义频率配置” 的新区域。
  - Channel 1, Channel 2, ... Channel 8: 这代表网关拥有的 8 个独立的接收 “频道”。您需要为每一个您想使用的频道手动指定一个频率。
  - 输入框: 每个频道后面都有一个输入框, 用于填写频率值。

自定义频率 配置			
Channel 1	Channel 2	Channel 3	Channel 4
470300000	470500000	470700000	470900000
Channel 5	Channel 6	Channel 7	Channel 8
471100000	471300000	471500000	471700000

## 应用并保存配置



到页面最下方, 您会看到一个黑色的 “应用” 按钮。  
点击 “应用” 按钮, 系统会将您的新配置保存到网关中。

## 网络连接配置 (Internet)

本章节将指导您如何将 G6 网关连接至互联网, 以实现数据的回传与交互。您可以在 “设置” -> “网络连接” 页面对网关的网络连接参数进行配置。网关支持 **有线以太网 (Ethernet)**、**无线局域网 (WiFi)** 及 **4G 蜂窝网**

络三种联网方式，并支持**多链路备份与优先级管理**。通过“网络优先级”功能，您可以指定一种网络模式作为主用链路，其余网络作为备用链路。当主用网络发生故障（如网线断开）时，网关会自动无缝切换至备用网络，确保数据传输不中断。



## 以太网 有线网络 (ETHERNET)

有线网络因其高稳定性和抗干扰能力，通常作为网关的首选连接方式。请按照以下步骤在页面上完成配置：

### 1. 设置 以太网 网络优先级

在页面的“网络优先级”区域，如果您希望网关通过以太网传输数据，请勾选“以太网·优先”。



- 选择模式：在卡片下方的单选框中，勾选“以太网·优先”。

### 2. 配置以太网属性

向下滑动至“以太网 网络属性”区域，进行具体参数设置。

**以太网 网络属性**

有线网络设备 描述  
以太网 WAN 网口 (有线连接)

网络设备 MAC  
40:D6:3C:01:7D:38

网关IP 分配  
**自动** 手动

网关IP 地址  
192.168.1.46

网关 子网掩码  
255.255.255.0

网络 默认路由地址  
192.168.1.1

### IP 分配方式选项:

**IP 分配方式 修改将在网关重启后生效, 请在应用成功后手动重启网关**

- **自动 (DHCP)** —— [默认推荐]

适用场景: 网关直接连接到开启了 DHCP 功能的交换机或路由器。

“网关 IP 地址”、“子网掩码”和“默认路由地址”将自动填充, 无需手动输入。如图所示, 自动分配 IP 模式下, 网关已成功被分配 IP 192.168.1.46。

- **手动 (Static IP)**

适用场景: 企业内网要求固定 IP, 或网络未开启 DHCP 服务 (如直连电脑)。

填写参数: 切换后, 下方的参数栏变为**可编辑状态**, 请准确填写网络管理员提供的参数:

-网关 IP 地址: 例如 192.168.1.100 (需确保不与局域网其他设备冲突)。

-网关 子网掩码: 通常为 255.255.255.0。

-网络 默认路由地址: 即上级路由器的网关地址, 例如 192.168.1.1。

### 3. 保存并使新配置生效

**检测到设备配置被修改**  
点击保存以应用最新配置

**应用** **取消**

配置完成后, 页面最下方, 您会看到一个黑色的“应用”按钮。

点击“应用”按钮, 系统会将您的新配置保存到网关中。

**IP 分配方式 修改将在网关重启后生效, 请在应用成功后手动重启网关**

## WIFI 热点 网络 (WIFI STA)

当设备部署环境中无法布线网线，但有可用的局域网 WiFi 覆盖时，您可以将网关作为无线终端（STA 模式）连接至现场的主路由器，实现无线上网。

### 1. 设置 WIFI 网络优先级

在页面的“网络优先级”区域，如果您希望网关通过 WIFI 传输数据，请勾选“WIFI 网络·优先”。

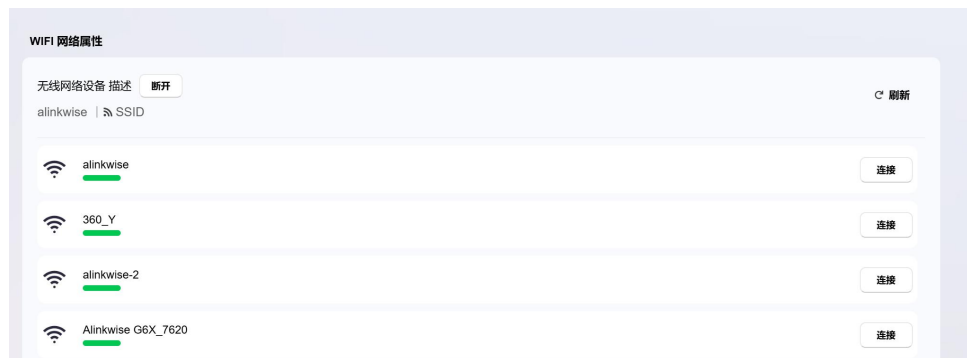


- 选择模式：在卡片下方的单选框中，勾选“WIFI 网络·优先”。

### 2. 扫描并连接 WIFI 网络

向下滚动至“WIFI 网络属性”区域，按照以下步骤建立连接：

扫描网络：点击列表右侧的“刷新”按钮。网关会自动扫描周围环境中的无线网络，并在下方列表中显示。



选择信号源：在列表中找到您需要连接的 WiFi 名称（SSID，例如图中的 alinkwise 或 360\_Y）。

发起连接：点击目标网络右侧的“连接”按钮。

输入密码：（系统会弹出密码输入框）请输入该 WiFi 的正确密码并确认。



### 3. 确认连接状态与断开

连接成功：配置正确后，在“无线网络设备描述”下方，会显示当前已连接的 WiFi 名称（如 SSID:alinkwise）。

断开连接：如果您需要更换 WiFi 或停止使用无线网络，可以点击当前 SSID 旁边的“断开”按钮。

### 3. 保存并使新配置生效



配置完成后，页面最下方，您会看到一个黑色的“应用”按钮。

点击“应用”按钮，系统会将您的新配置保存到网关中。

## 4G 蜂窝网络 (4G cellular network)

在没有有线宽带或 WiFi 覆盖的户外、偏远地区，或者需要将 4G 作为高可靠性的独立备份链路时，您可以使用 4G 模块进行联网（需设备出厂选配 4G 硬件功能）。

异常提示：如果界面显示“！SIM 卡未检测”或状态显示为“未插入”，说明网关未识别到卡片。请检查卡片是否插反、松动，或者是否在带电状态下热插拔（**若是，请重启网关**）。

### 2. 设置 4G 网络优先级

在页面的“网络优先级”区域，如果您希望网关通过 4g 网传输数据，请勾选“4G 网络·优先”。



- 选择模式：在卡片下方的单选框中，勾选“4G 网络·优先”。

### 3. 查看 4G 状态信息与高级设置

SIM 卡状态 / IMSI / ICID：显示 SIM 卡的内置识别码。只要显示出具体数字，即代表 SIM 卡读取成功。

#### 蜂窝网 APN 设置：

普通手机卡/通用物联网卡：网关默认开启自动拨号（如显示 3gnet 等默认值），一般无需手动修改。

专网卡/定向卡：如果运营商要求配置特定的接入点（APN），请在此框中填写运营商提供的 APN 字符串。



#### 4. 保存并使新配置生效



配置完成后，页面最下方，您会看到一个黑色的“应用”按钮。  
点击“应用”按钮，系统会将您的新配置保存到网关中。

保存并等待网络连接后，页面顶部的“网络连接”状态区域，将会点亮4G的绿色指示灯，表示4G链路已拨号成功并正在传输数据。

## 数据处理及传输 (Data Process)



G6 系列网关集成了边缘计算能力，支持两种不同的 LoRaWAN 数据处理架构。在左侧菜单栏进入“设置”->“数据传输”页面，您可以根据实际的物联网系统架构，灵活选择工作模式。

### 远程 NS 模式 ( Packet Forwarder)

网关将作为标准的数据包转发器 (Packet Forwarder) 运行。此时，网关不再负责解析或解密终端数据，而是将接收到的底层 LoRa 射频加密包，原封不动地通过网络转发给外部的第三方 LoRaWAN Network Server (如 ChirpStack、The Things Network、腾讯连连、AWS 等)，由外部服务器完成设备的鉴权与数据解析。

在此模式下，您无需在网关本地的“节点管理”中添加终端设备。所有的设备注册、入网管理和数据解码，均需在您的外部 LoRaWAN 服务器平台上进行。

## 1. 选择外部 NS 通信协议

根据您的远端 LoRaWAN 服务器的支持情况，选择合适的底层转发协议（点击单选框进行切换）：



- **Semtech UDP GWMP (经典 UDP 模式)**

业界最通用、兼容性最广的传统 UDP 转发协议。通常只需配置服务器的 IP 地址以及上行/下行端口（默认通常为 1700）。

### 填写转发地址

在展开的“Semtech Udp GWMP 转发地址”输入框中，准确填写您外部 LoRaWAN 服务器的接入地址与通信端口。

**输入格式：** IP 地址:端口号 或 域名:端口号。

**参数示例：** 如 39.98.253.192:1700。

**配置提示：** 1700 是 Semtech UDP 协议在业界最常用的默认数据收发端口。填写前，请务必确认您的服务器已在云端安全组或本地防火墙中放行了该 UDP 端口。

- **ChirpStack MQTT Bridge**

专为 ChirpStack V3 设计，或兼容 ChirpStack Gateway Bridge 数据格式的 MQTT 代理服务器。

通过 MQTT 协议将网关与服务器连接，相比 UDP 在复杂网络环境下（如需穿透 NAT 防火墙）连接更稳定。

### 配置转发服务器参数

选中协议后，下方会展开对应的参数填写区域。请根据外部服务器的接入要求进行配置：

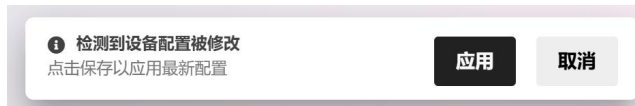
- **转发地址 (Broker Address):** 输入服务器的连接协议、IP 或域名及端口。

- 格式通常为 tcp://IP:端口 (如 tcp://39.98.253.192:1883) 。如果是加密传输 (格式应为 ssl://IP:端口 (如 ssl://39.98.253.192:8885) ) 。
- 用户名 / 密码: 填写连接该 MQTT 代理服务器所需的身份验证信息 (如 gateway / hA2QUmW7kDpINMEq) 。若服务器允许匿名接入则留空。

## 2. 连接测试与保存

验证连通性: 所有参数填写并粘贴证书后, 务必点击页面最下方的“测试”按钮。网关将发起一次模拟连接, 并在页面上提示验证结果。如果测试失败, 请检查地址、端口是否开放, 以及证书格式是否包含多余的空格或换行。

## 3. 保存并使新配置生效



配置完成后, 页面最下方, 您会看到一个黑色的“应用”按钮。点击“应用”按钮, 系统会将您的新配置保存到网关中。

## 内置 NS 模式 (Built-in NS)



网关本身作为 LoRaWAN Network Server (NS), 直接在网关内部完成终端设备的入网鉴权、数据去重、解密及 Payload (有效载荷) 解码。

**适用场景:** 用户没有或不需要搭建自己的 LoRaWAN 服务器 (如 ChirpStack、TTN), 希望直接获取终端解析后的明文 JSON 数据 (如温度、湿度具体数值), 并推送到自己的业务系统中。

### 1. 选择内置 NS 通信协议

根据您远端 LoRaWAN 服务器的支持情况, 选择合适的底层转发协议 (点击单选框进行切换):



## 内置 Alink API (Data Interface)

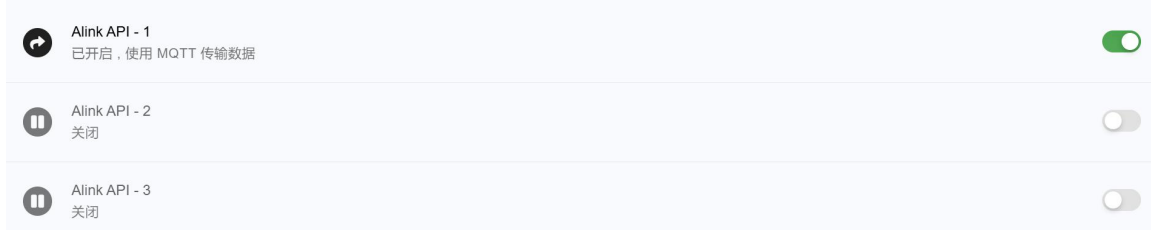
当网关工作在“内置 NS 解码模式”，网关内部将终端射频数据处理为标准的 JSON 格式明文后，通过 Alink API 输出数据至用户。Alink API 支持通过 **MQTT**、**HTTP**、**TCP**、**WS (WebSocket)** 这 4 种主流网络协议，将数据安全、实时地传输至您的服务器。



网关提供 3 个完全独立的 API 接口（API-1、API-2、API-3），支持同时向多个不同服务器推送数据。以下以配置 Alink API - 1 使用最常用的 MQTT 协议 为例进行详细说明。

### 1. 开启接口与协议选择

开启 API 通道：将 Alink API - 1 右侧的开关拨至开启状态（显示绿色）。



选择接口协议：**开启后**，link API - 1 项目区域**将会变为可点击**，点击后下滑出现“Alink API 1 接口协议”区域，根据您的应用服务器支持情况，选择对应的数据推送方式：

Alink API 1 接口协议

**MQTT** HTTP TCP WS

- **MQTT (推荐)**：作为 MQTT Client 连入消息队列，支持双向通信（上报与下发），是物联网场景下最标准且稳定的推荐协议。
- **HTTP**：网关通过 HTTP/POST，将节点上报数据单向 POST 推送至您的服务器 URL。
- **TCP**：网关内置 TCP 客户端（Client），主动连接您指定的远端 TCP 服务器建立持久连接，实现底层流数据的双向传输，适用于对接传统的后台系统。
- **WS (WebSocket)**：通过 WebSocket 协议建立全双工长连接。支持服务器主动下发指令，适合高频、实时的 Web 端数据交互场景。

## 2. 配置 MQTT 服务器连接参数

在此区域填写您自有应用服务器（MQTT Broker）的接入信息：

Alink API 1 接口协议

**MQTT** HTTP TCP WS

MQTT 服务器 地址

tcp://39.98.253.192:1883 [» 连接测试](#)

MQTT 用户名

user1

MQTT 密码

password1

MQTT ClientID

639c6847407c7c4cNS1

- **MQTT 服务器 地址**：输入标准的主机地址与端口号，格式为 协议://IP 或域名:端口。例如截图中的 tcp://39.98.253.192:1883。若使用加密连接，协议头通常为 ssl://。
- **MQTT 用户名 / 密码**：填写您的 MQTT 代理服务器分配的身份验证凭证（如 user1 / password1）。若服务器允许匿名接入则留空。
- **MQTT ClientID**：MQTT 客户端的唯一标识符。系统会默认生成一个基于网关 MAC 地址的唯一 ID（如 639c6847...NS1），通常无需修改。若需修改，请确保其在您的 MQTT Broker 中是绝对唯一的，否则会导致连接频繁掉线重连。

- 网络连通性测试：填写完毕后，强烈建议点击地址栏右侧的“» 连接测试”按钮。网关会尝试连接服务器，提示成功后方可进行下一步。

### 3. MQTT 主题 (Topic) 配置与通信机制 (核心)

MQTT 依赖“主题”来进行消息路由。网关内置了 MQTT 客户端，您需要明确网关发布 (Publish) 和订阅 (Subscribe) 的主题，以便与您的应用服务器精准对接。

开启“MQTT 网关主题 (推荐)”开关后，您可以自定义以下四类主题路径 (建议保持默认的唯一路径结构)：

- 上报数据 主题 (Uplink)

网关负责发布 (Publish)，用户服务器负责订阅 (Subscribe)。

所有终端节点上传的原始数据/业务数据 (如温湿度值、电量等)，网关解析后都会推送到此主题。

- 下发数据 主题 (Downlink)

网关负责订阅 (Subscribe)，用户服务器负责发布 (Publish)。

当您的业务系统需要向节点下发数据 (如控制指令) 时，需将指令按指定 JSON 格式发送至此主题，网关收到后会通过 LoRa 射频下发给对应终端。

- 入网主题 (Join)

网关负责发布 (Publish)，用户服务器负责订阅 (Subscribe)。

当 LoRaWAN 终端设备发起 OTAA 入网请求并成功加入网络时，网关会在此主题推送入网成功状态及相关密钥信息日志。

#### ACK 主题 (ACK)

网关负责发布 (Publish)，用户服务器负责订阅 (Subscribe)。

当服务器下发确认类指令 (Confirmed Downlink) 后，终端回复的应答 (ACK) 信号会推送到此主题，供服务器确认指令是否成功送达。

**(进阶选项) MQTT 节点主题：**默认为关闭状态。若开启，网关将不再把所有终端数据混在一个“网关主题”里，而是为每一个终端设备生成独立的收发主题 (通常以终端的 DevEUI 作为主题内容进行区分)，适用于对数据分流隔离有严格要求的系统架构。

### 4. 高级安全与证书 (客户端证书)

客户端证书：默认为关闭状态。如果您的 MQTT 服务器启用了双向 TLS/SSL 严格认证 (例如对接 AWS IoT、阿里云等平台)，请开启此选项，并在展开的输入框中粘贴您的 CA 根证书、客户端证书 (Client Certificate) 及客户端私钥 (Client Key)。

### 3. 保存并使新配置生效



配置完成后，页面最下方，您会看到一个黑色的“应用”按钮。  
点击“应用”按钮，系统会将您的新配置保存到网关中。

## 内置 NS 服务器参数配置 (Ns Config)

在“内置 NS 解码模式”下，网关充当了完整的 LoRaWAN 网络服务器。为了适配不同厂家的终端设备或满足特定的射频规范，您可以对底层网络参数进行精细化调整。通常情况下，保持默认设置即可满足绝大多数应用场景。

启用网关心跳、连接、状态、断开MQTT主题

心跳秒数  
60

心跳主题接口  
API-1  API-2  API-3

网关状态主题  
gateway/event/stats

入网请求间隔 (s)  
5

RX1窗口间隔 (s)  
1

RX2 发送频率 (Mhz)  
509.3

RX2 发送速率 (SF)  
SF7 SF8 SF9 SF10 SF11 SF12

RX2 发送带宽 (BW)  
125KHZ 250KHZ 500KHZ

如果您的项目有特殊需求，请参考以下说明进行修改：

### 1. 网关状态与心跳配置

用于向您的业务系统（通过 MQTT）定时汇报网关自身的在线健康状态。

- 心跳秒数：设定网关发送“心跳包”的时间间隔。默认 60 秒。

- 心跳主题接口：选择心跳数据通过哪个 Alink API 通道推送（如 API-1）。开启后，您的服务器就可以通过订阅该通道的主题，来监控网关是否断网。

网关状态主题：网关发布心跳数据的专用 MQTT Topic 路径（例如：gateway/xxx/event/status）。您的服务器需订阅此主题。

该报文用于网关运行状态监测，不涉及具体设备或业务数据。

通过该心跳状态报文，用户系统可以实时掌握网关的在线情况，用于运行监控和运维管理。

上发报文 MQTT 默认主题（可自定义）：gateway/event/stats

上发报文 JSON 数据格式（不可修改）：

```
{
  "gatewayID": "639c6847407c7c4c",    // 网关唯一标识 (Gateway EUI)
  "time": "2025-05-18T10:28:25Z",    // 心跳上报时间 (UTC, ISO 8601 格式)
  "status": "00 天 16 时 09 分 13 秒" // 网关运行时长
}
```

## 2. 入网时序配置

入网请求间隔 (s)：规定终端（OTAA 模式）在发送 Join 请求后，网关在多少秒之后下发入网许可（Join-Accept）。默认通常为 5 秒。此参数必须与终端设备的底层固件设定严格一致，否则终端将因错过接收窗口而导致入网失败。

## 3. 窗口 1 (RX1) 时序配置

RX1 窗口间隔 (s)：终端在发送完普通上行数据后，打开第一个接收窗口（RX1）等待网关下发指令的时间间隔。默认通常为 1 秒。此时使用的频率和速率由终端决定。

## 4. 窗口 2 (RX2) 射频参数配置 (核心控制窗口)

如果网关在 RX1 窗口未能成功下发指令，或者针对 Class C 类型的常听设备，网关将使用 RX2 窗口进行数据下发。RX2 的参数是固定的，如果更改，必须确保网关与所有被控终端的 RX2 参数完全对齐。

- RX2 发送频率 (Mhz)：设置网关下发控制指令的固定频点。以中国区为例，常见默认为 505.3 MHz。
- RX2 发送速率 (SF - 扩频因子)：扩频因子决定了通信距离和速率。数值越大（如 SF12），通信距离越远、抗干扰能力越强，但传输速率越慢，占用空中时间越长。

截图示例中选择的是 SF12，这能保证最大的下行覆盖范围，代价是下发耗时较长。

RX2 发送带宽 (BW)：射频通道的带宽。通常室内和园区场景默认为 125KHz。

### 排障提示:

如果您发现业务系统能够成功推送控制指令到网关，但在网关的“下行记录”中显示发送成功后，终端却没有任何动作响应，请重点检查：终端设备内部配置的 RX2 频率、SF 和 BW 是否与此处的设置（如 505.3 / SF12 / 125K）完全一致。

## 节点管理与通信 (Initial Login & Settings)

### 注册第一个设备 (Register the first device)

当网关配置为“内置 NS 解码模式”时，用户必须将 LoRaWAN 终端设备（如传感器、控制器等）的网络参数注册到网关中，网关才能合法接收并解析其上传的数据。

○ 如果您在上一节配置了“远程 NS (网关转发模式)”，请直接在您的第三方 LoRaWAN 服务器（如 ChirpStack、TTN）上添加设备，无需阅读此章节。

- 进入左侧导航栏的“节点” -> “我的终端”，即可进行设备的添加与状态管理。

在“我的终端”页面中，点击 **终端管理** 按钮以呼出“注册新的终端”配置弹窗。请根据终端厂商提供的设备参数清单，依次配置以下核心信息：





### 新的终端名称

- 自定义该终端的标识名称（如：测试终端），用于在列表中快速识别和区分不同位置或功能的设备。

### 终端地址 (DEVEUI)

- 填写终端设备的全球唯一 64 位标识符（通常为 16 位十六进制字符，如 A000000000000001）。此参数固定不变，由设备出厂设定。

### 终端 CLASS (工作模式选择)

- A 模式 (Class A)：异步通信模式。终端平时处于休眠状态，仅在发送上行数据后，短暂开启两个接收窗口以接收网关下发指令。此模式功耗极低，适用于电池供电的传感器（如温湿度计、门磁等）。
- C 模式 (Class C)：同步通信模式。终端的接收窗口几乎持续保持开启，网关可随时向其下发控制指令，响应无延迟。此模式功耗较高，适用于外部常供电的被控设备（如智能插座、空调控制面板等）。

### 网络接入方式 (入网模式选择)

- ABP 模式（激活即用，推荐）：

选中此模式后，需手动填写 **通信地址 (DevAddr)**、**网络会话密钥 (NwkSKey)** 和 **应用会话密钥 (AppSKey)**。终端无需发送入网请求，上电即可与网关进行加密通信，连接速度快，适用于室内固定网络环境。

- OTAA 模式（空中激活）：

选中此模式后，参数输入框将自动切换，仅需填写 **入网密钥 (AppKey)**。

终端在首次工作或重启时，必须向网关发起 Join 请求，双方动态协商生成会话密钥。安全性更高，但需确保终端**成功执行入网操作后方可通信**。

确认各项参数填写无误后，点击右下角的“注册”按钮完成添加。

### 批量添加与管理终端

针对多设备部署场景，注册弹窗右上角提供了快捷操作入口：



**批量添加**：点击后可下载标准的 CSV 模板文件。将多台设备的 DevEUI、密钥及工作模式录入表格后上传，即可实现几十至上百台终端的一键快速注册。

**批量管理**：提供对已注册终端的统一管理功能，支持批量修改设备的 CLASS 模式、离线时间判定阈值，或执行批量删除操作。

设备注册成功后，将在左侧“我的终端”列表中列出。

### 左侧基础状态看板

显示终端名称、设备参数（如 ABP | Class A）以及当前在线状态。若超过设定的周期未收到数据，状态将显示为灰色 Offline（离线）。同时统计历史累计的上行数据包数量。



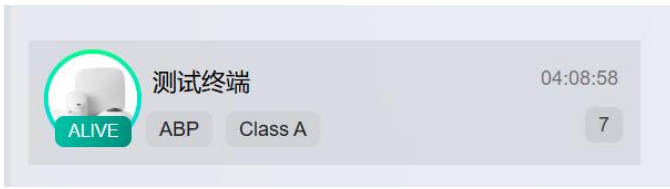
如需注销该设备，可点击右上角的“删除终端”按钮。

## 进行第一次通信 (Initiate the first communication)

当终端设备成功注册并上电工作后，它会按照自身设定的周期向网关发送 LoRa 无线数据（或由用户主动触发上报数据）。在网关的“我的终端”列表中点击具体的设备，即可进入“终端信息”详情页实时监控数据交互状态。

### 1. 查看终端通信概览

在“终端概览”选项卡中，您可以直观地掌握该设备的核心通信指标：



在线状态：设备名称旁显示绿色的 Alive 标签，表示网关近期已成功接收到该设备的数据。

- **上行数据 / 下行数据**：展示网关最近一次收到该终端的数据包（如 1234567890abcd0e，长度 8 Bytes）及最近一次向该终端发送的指令内容，并标注距今的时间间隔（如 4 分前）。
- **数据趋势图**：以折线图的形式，展示该终端最近几次上报数据的无线信号质量（包含 RSSI 信号强度和 SNR 信噪比），可用于评估终端安装位置的信号覆盖情况。
- **荷载转码**：若配置了 JS 解析脚本，此处将展示解析后的明文数据。

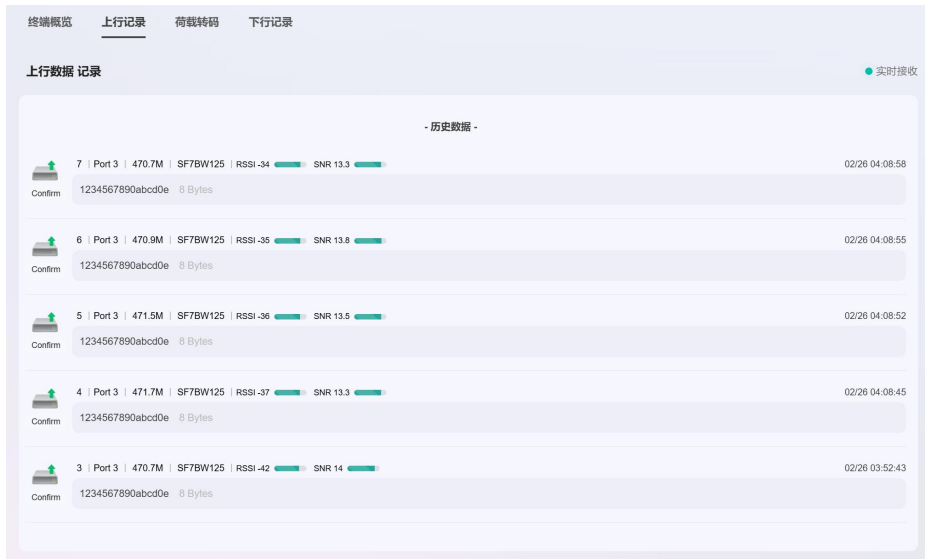


## 2. 查看详细上行记录

切换至“上行记录”选项卡，您可以查看该终端的历史数据记录：

数据包详情：每条记录均包含详尽的射频参数，例如：

- 包序号 (FCnt)：如 7、6、5，正常情况下应递增。
- 端口 (Port)：如 Port 3，应用层端口号。
- 频率与速率：如 470.7M、SF7BW125，表示该数据包使用的物理频点和扩频因子。
- 信号质量：如 RSSI -34（数值越大越好，通常在 -40 到 -120 之间）、SNR 13.3。
- 数据载荷 (Payload)：如 1234567890abcd0e 原始 16 进制数据。



### 常见故障排查：终端发送数据，网关无显示

如果在确认终端设备已上电并触发了发送动作，但在网关的“上行记录”或“终端概览”中始终无法看到任何数据，请按照以下顺序进行排障：

#### 第 1 步：核对终端注册参数（最常见错误）

检查入网模式：终端实际使用的是 ABP 还是 OTAA？必须与网关上的配置完全一致。

核对密钥 (Key)：进入“终端参数配置”页面，逐字核对 DevEUI、DevAddr、NwkSKey、AppSKey 或 AppKey。任何一个字母或数字的输入错误（如把 0 错看成 O，或包含了多余的空格），都会导致网关因无法解密而直接丢弃该数据包。

#### 第 2 步：检查物理射频频段

频点一致性：检查终端设备的工作频段（如 CN470）是否与网关的“无线频段”配置（如 470.3 ~ 471.7 MHz）匹配或包含在内。若终端发送的频率不在网关的监听范围内，网关将无法在空中捕捉到信号。

天线状态：确认网关的 LoRa 天线已牢固拧紧，且终端设备距离网关不要过远，也尽量避免紧贴网关（过近可能导致信号饱和失真）。

#### 第 3 步：确认 OTAA 设备的入网状态

如果您配置的是 OTAA 模式，设备必须先完成 Join（入网）请求并得到网关回应，才能发送业务数据。

排查方法：让终端重新上电或触发强制配网，然后在网关的“实况” -> “系统日志”或相关 MQTT 订阅主题中，观察是否有该设备的 Join 记录。如果一直未能 Join 成功，请回到第 1 步和第 2 步检查 AppKey 和频段。

## 使用 ALink API 推送数据 (Push data)

当您在网关的“上行记录”中成功看到终端上报的数据后，恭喜您，网关与传感器之间的 LoRaWAN 无线物理链路已经成功配置并正常工作。

接下来的最后一步，就是如何让您自己的业务系统（如物联网云平台、本地工控软件、手机 App 后台）拿到这些数据，并实现对设备的监控。

### 使用 ALink API 与业务系统交互

网关接收到终端的射频数据后，会将其打包为标准的 JSON 格式。只要您在前面（参考 3.6 章节）成功配置并开启了 ALink API（如 MQTT 接口），网关就会自动充当“桥梁”，在您的业务系统与底层终端之间双向搬运数据。

以下以最常用的 MQTT 协议为例，说明您的业务系统应该如何操作：

#### 1. 业务系统接收数据（上行 Uplink）

操作动作：您的业务服务器作为 MQTT 客户端，需要向 MQTT Broker 订阅（Subscribe）网关的上报主题（例如 /gateway/639c.../ns/up1）。

数据呈现：当传感器发来数据时，您的服务器会立刻收到一条 JSON 格式的消息。

默认数据结构解析：

```
{
  "applicationId": "0", // 应用 ID
  "applicationName": "G6x_LoRaWAN_Uplink", // 应用名称
  "deviceName": "测试终端 OTAA", // 设备名称
  "devEui": "9cc466c19fe2da25", // 设备唯一标识 EUI
  "fCnt": 2, // 上行帧计数
  "fPort": 3, // 数据端口号
  "payload": "EjRWeJCrzQ4=", // 原始上行数据 (Base64 编码)
  "data": "1234567890abcd0e", // 原始上行数据 (字符编码)
  "rxInfo": { // 接收信息
    "gatewayId": "639c6847407c7c4c", // 接收网关 EUI
    "rssi": -39, // 信号强度
    "snr": 13.8 // 信噪比
  },
  "txInfo": { // 无线参数
```

```
"frequency": 471.7, // 工作频点 (MHz)
"modulation": {
  "lora": {
    "spreadingFactor": "SF7BW125", // 扩频因子/带宽
    "codeRate": "CR_4_5" // 编码率
  }
}
}
```

## 2. 业务系统控制设备 (下发 Downlink)

当您的业务系统需要向设备发送控制指令 (例如: 打开智能阀门、调整上报周期) 时:

操作动作: 您的业务服务器需要向 MQTT Broker 的下发主题 (例如 /gateway/639c.../ns/down1) 发布 (Publish) 一条指令。

指令格式要求: 您发布的指令必须是 JSON 格式, 且明确指定目标设备的 EUI 和要下发的 16 进制控制码。示例结构如下:

JSON

```
{
  "deveui": "9fe2da259fe2da25", // 设备唯一标识 EUI
  "data": "1234567890abcdef", // 自定义下发数据内容 (字符编码)
  "fPort": 2, // 数据端口号, 默认 2
  "retry": 1, // 数据下发次数, 默认 1
  "confirmed": false // 终端是否需要在下一次上报中确认
}
```

网关执行: 网关在“下发主题”中监听到这条消息后, 会将其转换为 LoRa 射频信号, 在终端下一次开启接收窗口时 (Class A) 或立即 (Class C) 发送给该终端。

### 进阶预告: 告别 16 进制乱码 (荷载转码输出)

您可能会发现, 通过默认 API 收到的终端数据 (如 0100098e15d104) 是一段可读性较差的 16 进制数据。如果您的业务系统不想去写复杂的解析代码, 希望直接收到类似于 {"温度": 25.2, "湿度": 48.6} 这样直观的明文数据, 该怎么办?

网关内置了强大的“荷载转码 (JS 脚本解析)”功能。您只需在网关内部编写一段简单的 JavaScript 规则, 网关就能提前替您的服务器把数据翻译好再推送过去!

(具体如何编写解析脚本并将乱码转换为直观的温湿度数值, 请参考下一章节: 网关数据流控与编程。)

# 网关数据处理器与编程

G6 网关是一个强大的边缘计算节点。在左侧菜单栏的“应用” -> “数据流控” 模块中，**网关提供了多个独立的数据处理器 (Processor)**。您可以根据业务需求，通过编写 JavaScript (JS) 脚本，对流入、流出网关的数据格式进行任意的裁剪、重组和逻辑判断，从而实现与各种非标业务系统的“零代码改造”对接。

## 射频上行数据处理器 (Uplink)

网关射频上行处理器：获取节点上报的 LoRaWAN 原始射频数据，在网关将其通过 ALink API 推送给您的业务服务器之前，允许您使用自定义的 JS 脚本对最终输出的 JSON 报文格式进行自定义排版。



**典型应用场景：**您的现有云平台只接受特定格式的 JSON 数据（如必须包含 applicationName 字段，或不需要展示复杂的信号强度数组）。此时，您无需修改云平台代码，只需在网关射频上行处理器中编写一段 JS 脚本，即可让网关直接输出平台所需的任何数据格式。

### 1. 开启射频上行处理器与配置接口

- 1.进入“应用” -> “数据流控” 页面，在左侧处理器列表中点击“网关射频上行 处理器”。
- 2.确保该处理器右侧的状态显示为绿色的“已开启”。

已开启

- 3.在右侧主看板的“上报传输 API” 选项卡中，选择您希望输出数据的 API 通道（例如 Alink API - 1），并将其开关拨至“已启用”。

已启用

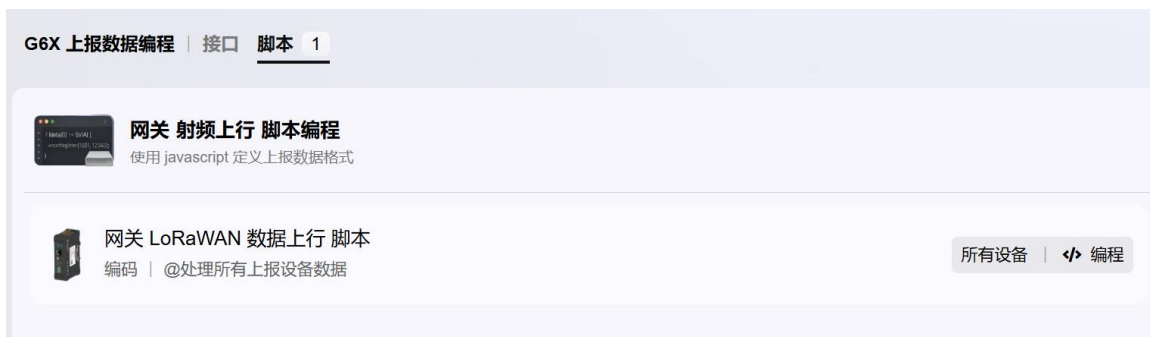
- 4.此时，当有节点上报数据时，在下方的“上报传输 记录” 列表中，您将能实时看到网关即将推送给您服务器的数据格式。



## 2. 编写上行数据重组脚本 (JS 编程)

G6 网关在出厂时已配备了标准的[LoRaWAN 射频数据帧]JSON 输出脚本。如果您对默认输出的 JSON 格式不满意，可以通过脚本进行重构。

1. 在主看板顶部切换至“脚本”选项卡。
2. 此处显示了“网关 射频上行 脚本编程”模块，该脚本全局生效于“所有设备”。点击右侧的“</> 编程”按钮。
3. 系统将弹出“LoRaWAN 输出序列化”的代码编辑器窗口。



脚本编写逻辑说明：

系统会默认提供一个名为 G6x\_Output 的 JS 函数。网关会将底层原始的射频数据字符串 (uplink\_Lorawan) 作为入参传给这个函数，您需要在这个函数内将其解析、重新拼装成新的 JSON 对象，并最终 return 输出。

```
LoRaWAN 输出序列化
1 // 自定义网关的射频数据输出 API 数据结构
2 var G6x_Output = function (uplink_Lorawan) {
3
4     const uplink = JSON.parse(uplink_Lorawan);
5
6     let data = {
7         // 设备信息
8         applicationId: uplink.app.appid,
9         applicationName: "G6x_LoRaWAN_Uplink",
10        deviceName: uplink.app.motename,
11        devEui: uplink.app.moteeui,
12
13        // LoRaWAN MAC 层信息
14        fCnt: uplink.app.userdata.seqno,
15        fPort: uplink.app.userdata.port,
16
17        // Payload (Base64)
18        payload: uplink.app.userdata.payload,
19        // Data (string)
20        data: uplink.app.userdata.payloadstr,
21
22        // 接收网关信息
23        rxInfo: {
24            gatewayId: uplink.app.gwrx[0].eui,
25            rssi: uplink.app.gwrx[0].rssi,
26            snr: uplink.app.gwrx[0].lsnr,
27        },
28
29        // 发射参数
30        txInfo: {
31            frequency: uplink.app.motetx.freq,
32            modulation: uplink.app.motetx.modulation,
33        }
34    };
35
36    return JSON.stringify(data);
37 };
```

操作建议：

- 按需裁剪：如果您的服务器不需要 rxInfo 或 txInfo 等复杂的射频参数，您可以直接在代码中将这些段落删除，以减少网络传输的流量消耗。
- 增加自定义字段：您可以随时在 let data = {} 结构中增加您业务系统所需的固定字段（如 tenantId: "1001"）。
- 保存生效：修改完代码后，点击右下角的“保存”按钮。

## 射频下行数据处理器 (Downlink)

网关射频下行处理器的作用与上行处理器正好相反。它的核心功能是：获取您的业务服务器通过 API（如 MQTT）下发给网关的控制指令，并在网关将其转换为无线射频信号发给终端之前，允许您使用自定义的 JS 脚本对下发的 JSON 数据结构进行反向解析和适配。



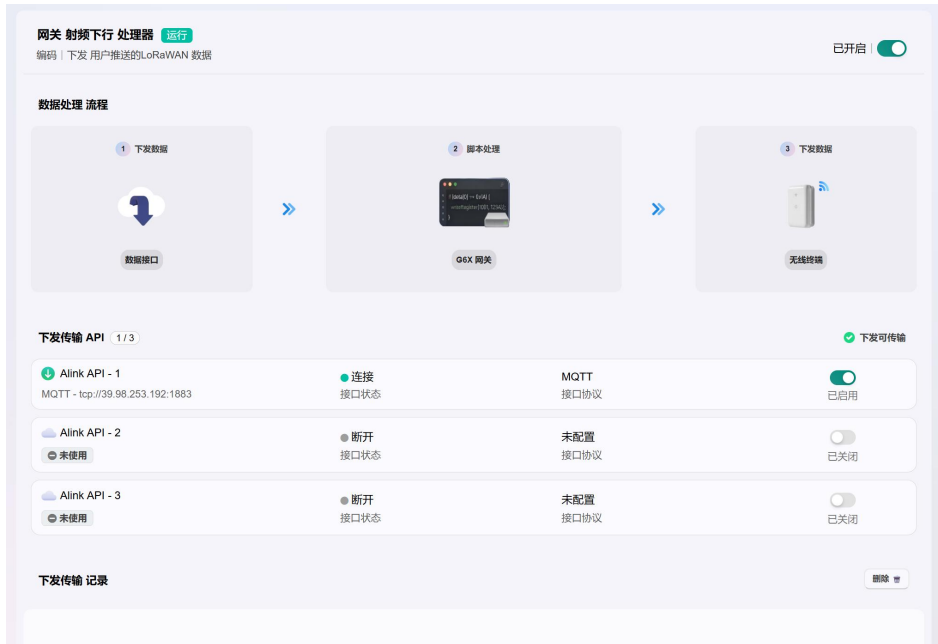
**典型应用场景：**您的云平台原本是为 WiFi 或 4G 设备设计的，下发的控制指令格式为 {"deviceId": "A001", "command": "open"}。而 LoRaWAN 网关底层要求严格的 {"devEui": "xxx", "data": "十六进制指令"} 格式。此时，您只需在此处编写一段 JS 脚本，将您的业务指令“转换”成网关需要的格式，从而免去修改云平台底层代码的麻烦。

## 1. 开启射频下行处理器

1. 在“应用”->“数据流控”页面左侧，点击“网关射频下行处理器”。
2. 确保该处理器右侧的状态显示为绿色的“已开启”。

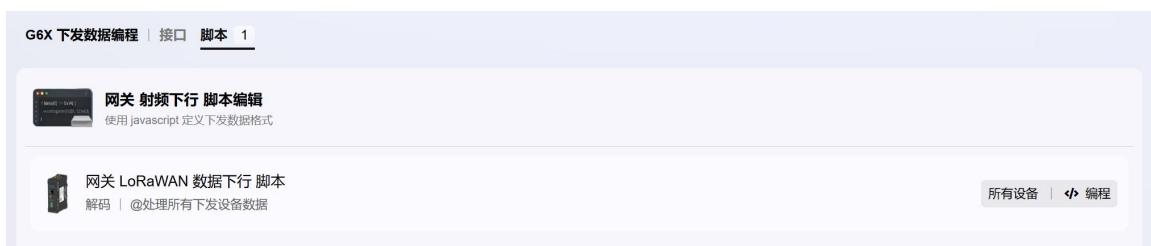
已开启

3. 此时，当有接口数据下发时，在下方的“下发传输记录”列表中，您将能实时看到网关转换的下发数据格式。



## 2. 编写下行数据适配脚本 (JS 编程)

1. 在主看板顶部切换至“脚本”选项卡，
2. 点击“</> 编程”按钮，进入“LoRaWAN 下发序列化”代码编辑器。



脚本编写逻辑说明：

系统提供了一个名为 G6x\_Input 的核心函数。您的业务系统通过 API 下发的所有数据（无论格式如何），都会以字符串形式作为 downlink\_Lorawan 参数传入此函数。您需要在此函数中提取关键信息，并严格按照网关底层要求的格式返回一个至少包含 Down\_Eui 和 Down\_Payload 的 JSON 对象。

默认脚本示例解析（参考截图）：



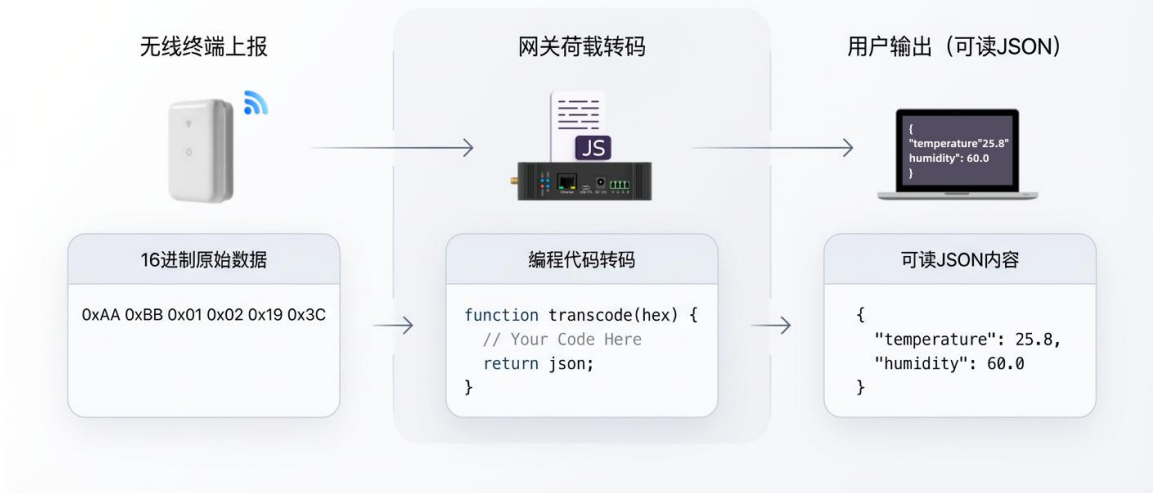
```
1 //自定义解析脚本
2 var G6x_Input = function(downlink_Lorawan) {
3   const down = JSON.parse(downlink_Lorawan);
4
5   // 解码下发数据
6   let data = {
7     Down_Eui: down.devEui,
8     Down_Payload: down.data,
9     // Down_Ack: down.confirmed,
10    // Down_Port: down.fPort,
11    // Down_Retry: 1,
12  };
13  return JSON.stringify(data);
14  };
```

操作建议：

- 字段映射（Mapping）：如果您的云平台下发的字段叫 mac\_address，只需在代码中修改为 Down\_Eui: down.mac\_address 即可。
- 指令转换：如果您的云平下发的是明文指令（如 {"action": "turn\_on"}），您甚至可以在脚本内部写 if..else 逻辑，将其转换为对应的 16 进制控制码：if(down.action === "turn\_on") { data.Down\_Payload = "01050000FF008C3A"}。这种边缘计算能力极大降低了云平台的开发复杂度。
- 修改完成后，点击“保存”生效。您可以在“下发传输记录”列表中监控到指令下发的执行情况。

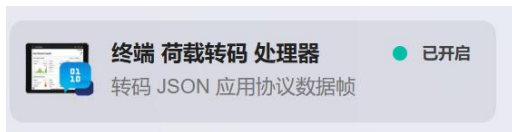
## 终端 荷载转码 处理器 (JSCode)

### 网关荷载转码：从原始数据到可读信息的智能转换



LoRaWAN 传感器上传的业务数据 (Payload) 通常是 16 进制字符串 (例如 01030E0113...)。G6 网关支持您在网关本地植入各传感器厂家的解码规则 (JS 脚本)，将接收到的 16 进制数据直接“翻译”成直观的 JSON 明文 (例如: {"temp": 25.5, "humi": 60.2}) 后再推送给云端。

### 1. 开启处理器 & 接口配置

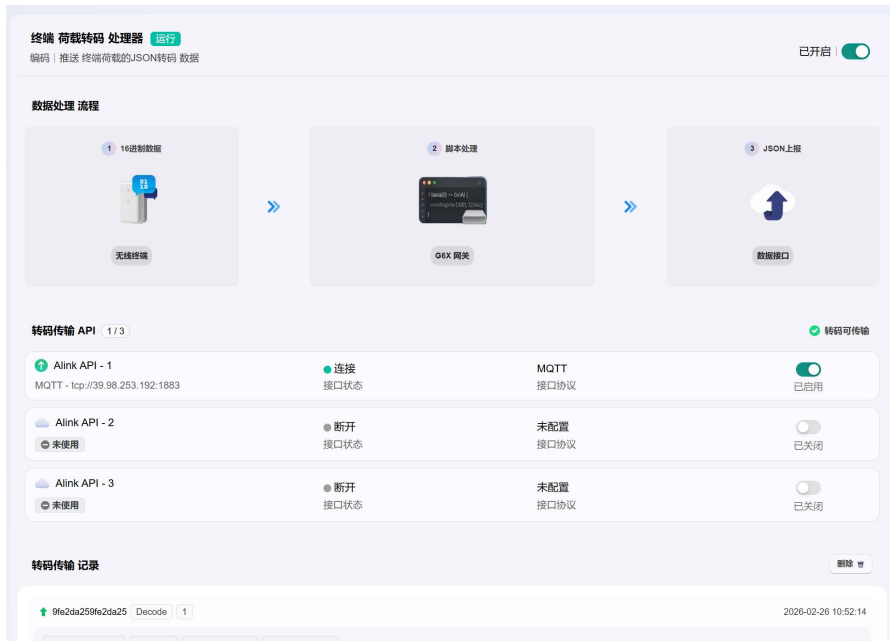


1. 在“应用” -> “数据流控” 页面左侧，点击“终端 荷载转码 处理器”。
2. 确保该处理器右侧的状态显示为绿色的“已开启”。

已开启

3. 在右侧主看板的“转码传输 API” 选项卡中，选择您希望输出数据的 API 通道 (例如 Alink API - 1)，并将其开关拨至“已启用”。

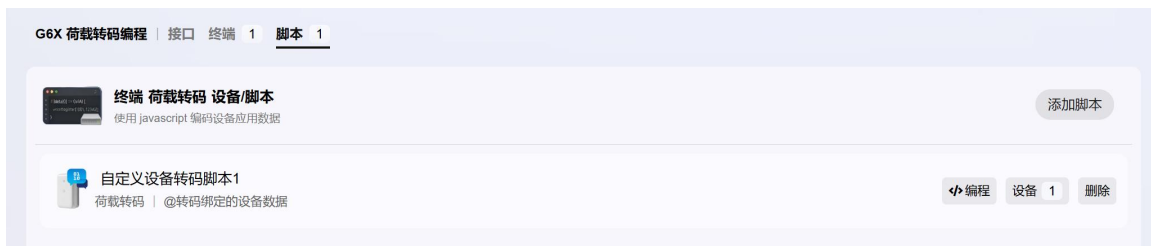
已启用



## 2. 编写传感器专用的转码脚本

不同厂家、不同型号的传感器，其 16 进制数据的排列规则完全不同。因此，您需要针对具体的传感器型号编写解码脚本。

1. 在主看板顶部切换至“脚本”选项卡，点击“添加脚本”或对现有脚本点击“</> 编程”。



2. 进入代码编辑器。系统提供了一个名为 G6x\_Decode 的核心函数，并内置了一个 parse 工具函数，用于快速截取 16 进制字节并转换为整数。

示例转码脚本解析（参考截图，以某温湿度传感器为例）：

```
自定义设备转码脚本1

1 // 自定义解析脚本
2 var G6x_Decode = function (byte, device) {
3
4     const device_t = JSON.parse(device);
5
6     if (device_t.port < 255) {
7         // 【选用】 根据数据中的 LoRaWAN fport 进行解码
8     }
9
10    let val_1 = parse(byte, 2, 2) / 10; // 室内温度
11    let val_2 = parse(byte, 4, 2) / 10; // 设定温度
12
13    let data = {
14        name: device_t.name || "未知设备",
15        port: device_t.port,
16        temp: val_1,
17        humi: val_2,
18    };
19
20    return JSON.stringify(data);
21 };
22
23 // 工具函数：提取指定字节范围的16进制字符串并转为整数
24 function parse(frame, startByte, byteLength) {
25     const start = startByte * 2;
26     const end = start + byteLength * 2;
27     const hexStr = frame.slice(start, end);
28     return parseInt(hexStr, 16);
29 }
```

保存 关闭

3. 编辑完成后，点击“保存”。

### 3. 将脚本与对应的终端设备“绑定”（关键步骤）

荷载转码脚本是区分设备型号的。如果编写了脚本但没有绑定对应的终端，转码将不会生效。

1. 在“脚本”选项卡页面中，找到您刚刚编写的脚本列表项，点击右侧的设备按钮。
2. 在脚本列表项下滑的设备绑定区域中，点击“绑定设备”（或点击“添加更多设备”）。
3. 在弹出的“绑定设备到脚本”窗口中，勾选属于该型号的传感器终端（如测试终端），点击“绑定”。



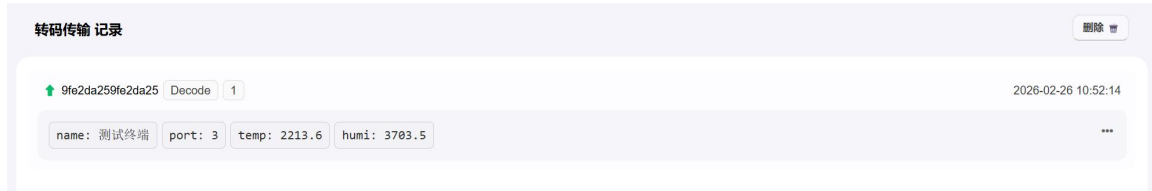
切换至顶部的“终端”选项卡，这里罗列了网关下所有的设备，您可以直观地查看哪些设备处于“未绑定”状态，并为它们逐一指派解码脚本。

#### 4. 验证转码输出结果

绑定成功后，返回该处理器的“接口”选项卡。

当传感器再次上报数据时，向下滚动查看“转码传输记录”。

此时您会发现，原本的 16 进制字符串已经变成了直观的业务数据结构（例如截图所示的 name: 测试终端，temp: 2213.6，humi: 3703.5）。您的业务服务器订阅此 API 接口后，将直接收到这些直观的 JSON 报文，大大降低了云端的开发工作量。



## 终端联动控制 处理器 (Linkage)

### 网关智能联动：本地编程，即时响应的逻辑控制中心

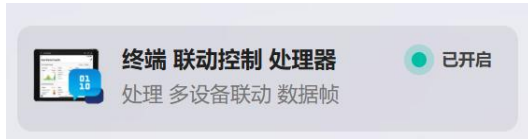


终端联动控制处理传统的物联网架构中，联动逻辑通常由云端服务器执行：传感器上报数据 -> 云端判断 -> 云端下发控制指令。这种方式依赖外部网络，一旦断网联动就会失效。

终端联动控制处理器 赋予了网关在本地执行复杂逻辑的能力。您可以直接在网关内部编写 JS 逻辑脚本。当“触发设备”（如温湿度传感器）上报数据时，网关会立即执行脚本进行状态判断；若满足条件，网关将直接在本地向“响应设备”（如空调控制器）下发控制指令，全程无需云端服务器参与，实现真正的“断网可用”与极低延迟。

**注意：**被控制的“响应设备”必须是 Class C 类型（常供电），否则无法及时接收网关下发的指令。

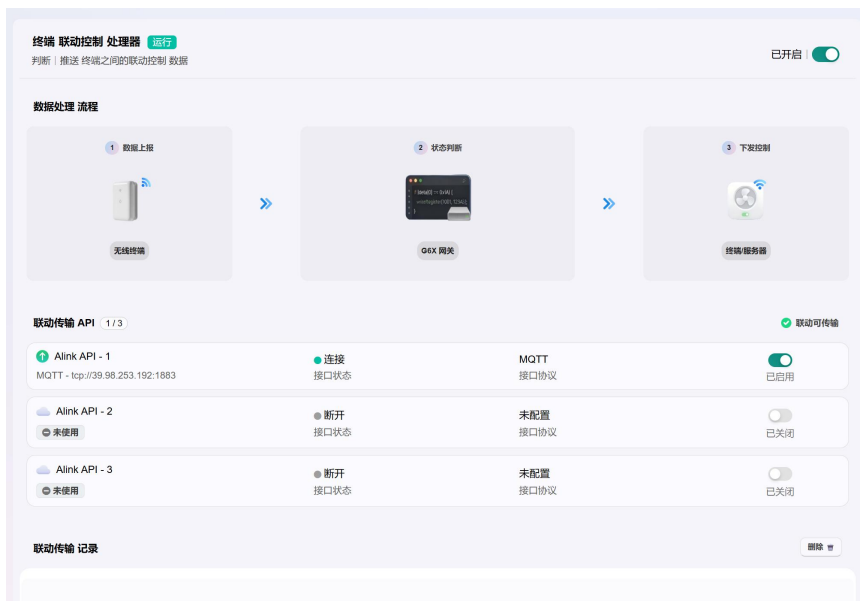
## 1. 开启联动处理器与配置输出接口



1. 在“应用”->“数据流控”页面左侧，点击进入“终端 联动控制 处理器”。
2. 确保该处理器右侧的状态显示为绿色的“已开启”。

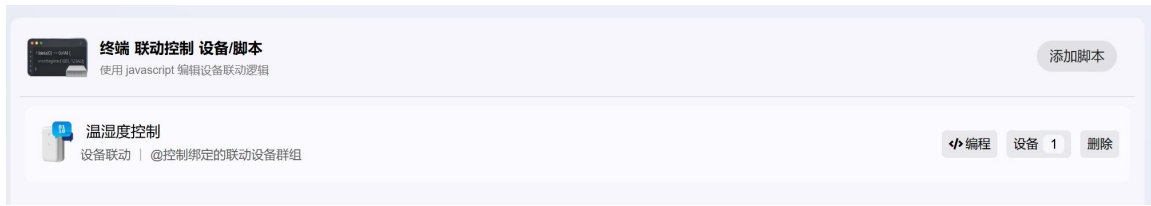


3. 联动传输 API (可选)：如果您希望云端服务器也能“知道”网关刚刚执行了一次本地联动，可以启用 API。网关在执行联动时，会通过此接口向云端推送一条自定义的“联动日志”。选择您希望输出数据的 API 通道（例如 Alink API - 1），并将其开关拨至“已启用”。



## 2. 编写联动逻辑脚本 (JS 编程)

1. 在顶部切换至“脚本”选项卡，点击“添加脚本”，为其命名（如“温湿度控制”）。
2. 点击“</> 编程”进入代码编辑器。



系统提供了一个名为 G6x\_Control 的核心函数。当“触发设备”上报数据时，该函数会被调用。  
联动脚本示例解析（参考截图，以“温度超标自动关空调”为例）：



3. 编写完成后点击“保存”。

### 3. 绑定联动套件（触发端与响应端配对）

这是联动控制中关键的一步。您必须告诉网关：“谁”的数据用来触发这段脚本，这段脚本产生的指令又该发给“谁”。

1. 在“脚本”选项卡中找到刚才编写的脚本，点击下方的“绑定群组”（或操作栏的“设备”）。

2. 在弹出的“联动套件设备绑定”窗口中：

- 左侧【触发设备】：勾选提供数据的传感器（支持单选或多选）。例如勾选 J32\_环境监测。
- 右侧【响应设备】：勾选接收指令的被控终端（必须是 Class C 设备）。

3. 点击“绑定”完成配对。

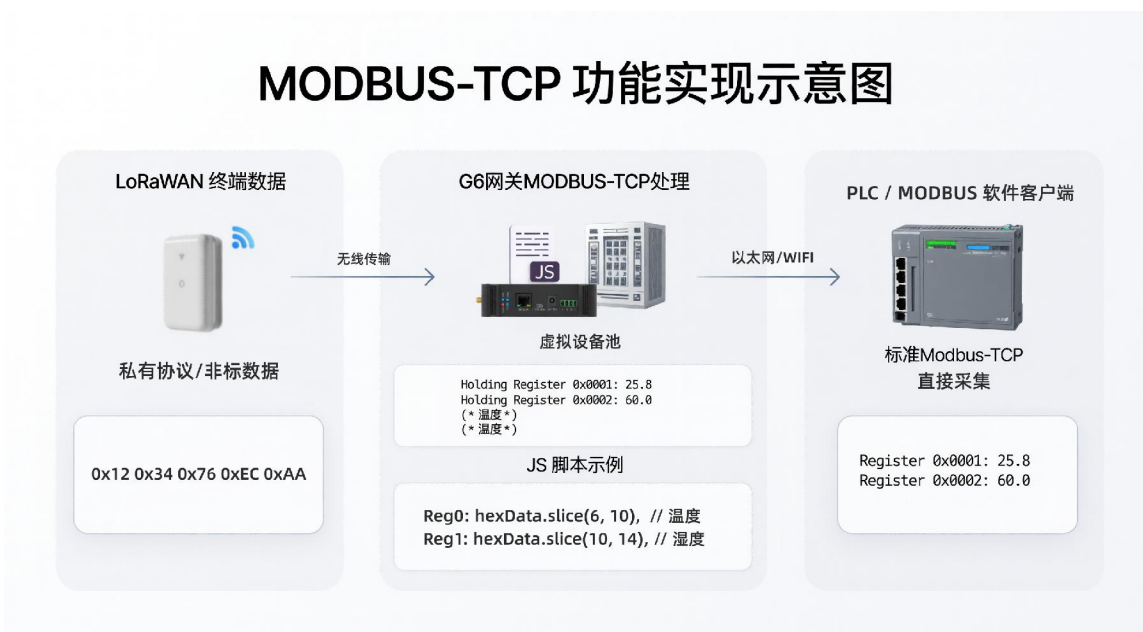


注：您可以在“终端联动 群组”列表中查看到已经形成的配对关系，如：触发设备 J32\_环境监测 对应 响应设备【J32\_环境监测】（此截图示例中，设备将自己的数据作为判断依据，又将控制指令发给了自己，即“自触发自控制”的闭环）。

至此，一个脱离云端的本地自动化控制链路就彻底建立完成了！只要传感器上报温度，网关就会秒级响应并执行您的控制策略。

## MODBUS-TCP 服务器(MODBUS-TCP)

在工业自动化领域，PLC、SCADA 等上位机系统通常只支持标准的 Modbus 协议。而 LoRaWAN 终端上报的数据往往是厂家自定义的非标结构。



为了解决这一问题，G6 网关内置了 MODBUS-TCP 服务器。它会建立一个“虚拟 Modbus 设备池”。通过编写 JS 脚本，网关会将 LoRaWAN 终端上报的数据，放置到虚拟 Modbus 设备的保持寄存器（Holding

Registers) 中。这样，传统的 Modbus-TCP 主机（如 PLC）只需通过读取虚拟设备，就能像读取普通 Modbus 从站一样，轻松读取到 LoRa 传感器的数值，甚至可以通过写入寄存器来控制 LoRa 终端。

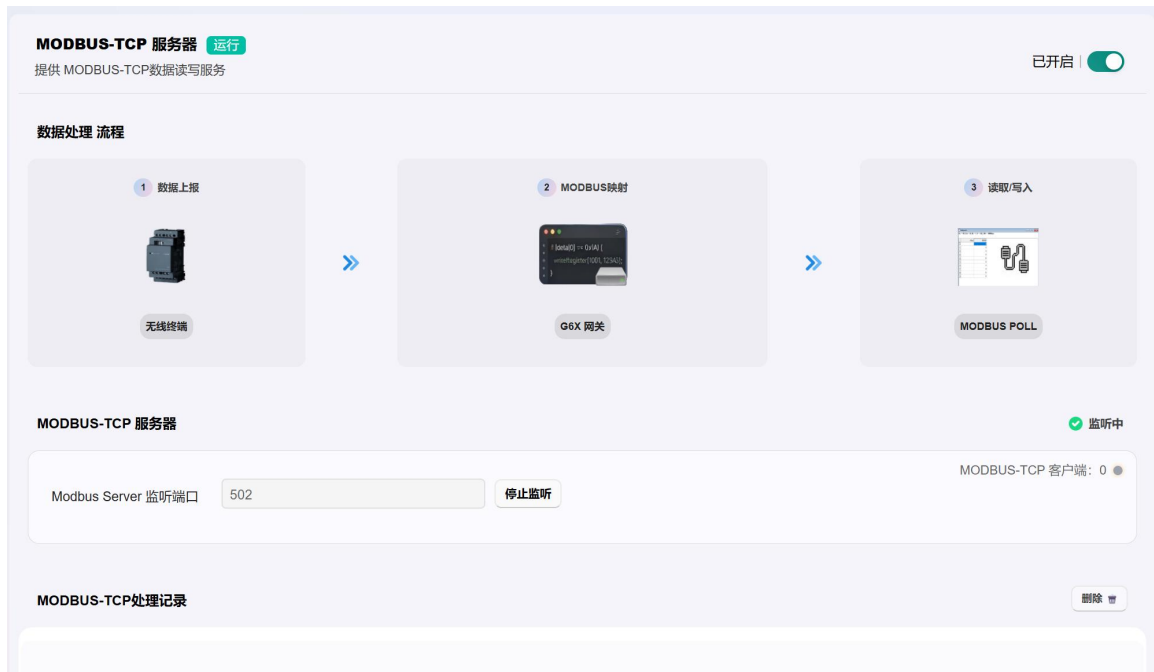
## 1. 开启 MODBUS-TCP 服务并监听端口

1. 在“应用”->“数据流控”页面左侧，点击进入“MODBUS-TCP 服务器”处理器。
2. 确保该处理器右侧的状态显示为绿色的“已开启”。

已开启 |

3. 在下方的“MODBUS-TCP 服务器”配置栏监听端口：默认填写 502（这是 Modbus-TCP 协议的标准默认端口）。点击开始监听按钮。

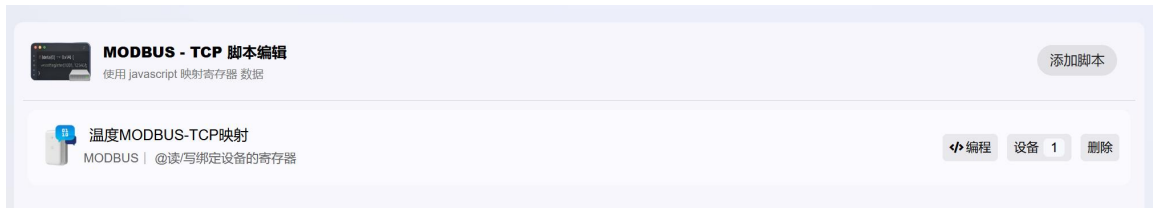
确认右侧状态显示为绿色的“监听中”，此时网关已准备好接受外部 PLC 或测试工具（如 Modbus Poll）的 TCP 连接。



## 2. 编写寄存器映射脚本 (JS 编程)

由于不同终端的数据结构不同，我们需要通过脚本告诉网关：把传感器数据的哪一段，放到哪一个寄存器里。

1. 在顶部切换至“脚本”选项卡，点击“添加脚本”，为其命名（如“温度 MODBUS-TCP 映射”）。
2. 点击“</> 编程”进入代码编辑器。



系统提供了两个核心函数：G6x\_Modbus\_Read（处理数据上报入库）和 G6x\_Modbus\_Write（处理上位机下发指令）。

映射脚本示例解析（参考截图，以多参数环境监测仪为例）：

编辑完成后点击“保存”。

### 3. 绑定终端并分配“Modbus 从站 ID”

要让 PLC 能够准确读取指定的传感器，必须为该传感器分配一个独立的从站 ID (Unit ID)（取值范围 1-255）。

**注意：**每个 Modbus 寄存器映射脚本，建议只绑定同一种类型的终端设备。因为不同设备的报文长度和含义不同，混用脚本会导致寄存器数据混乱。

1. 在“脚本”选项卡中找到刚才编写的映射脚本。
2. 点击右侧的“设备”或下方的“绑定设备”。
- 3.



4. 在弹出的“MODBUS 设备 添加”窗口中：

- 从站 ID (1-255): 手动输入一个不冲突的 ID 号，例如填写 1。
- 勾选设备：在列表中选中对应的终端（例如 J32\_环境监测）。

5. 点击“添加设备”。



此时，您在界面的绑定列表中可以看到：设备 J32\_环境监测 已经成功绑定了脚本，且分配的从站地址为 1。

#### 4. 外部系统读取数据验证

至此，网关侧的配置全部完成。您可以打开电脑上的 Modbus Poll 测试软件（或真实的 PLC）：

建立连接：选择 Modbus TCP/IP，输入网关的局域网 IP 地址，端口 502。

设置读取参数：

Slave ID（从站号）填 1（即上一步分配的 ID）。

Function 选 03 Read Holding Registers（读取保持寄存器）。

Address（起始地址）填 0。

Quantity（读取数量）填 5。

结果确认：连接成功后，您将在软件的地址 0~4 中，直接看到由网关“映射”过来的环境监测数值（以 10 进制或 16 进制显示）。

## MODBUS-RTU 处理器(MODBUS-RTU)

如果您的现场环境没有以太网，需要使用传统的 RS485 总线 将网关直接连接到 PLC、HMI（触摸屏）或各种串口工控设备上，则需要使用 MODBUS-RTU 服务器。

# MODBUS-RTU 功能实现示意图



在工业自动化领域，PLC、SCADA 等上位机系统通常只支持标准的 Modbus 协议。而 LoRaWAN 终端上报的数据往往是厂家自定义的非标结构。

为了解决这一问题，G6 网关内置了 MODBUS-RTU 服务器。它会建立一个“虚拟 Modbus 设备池”。通过编写 JS 脚本，网关会将 LoRaWAN 终端上报的数据，放置到虚拟 Modbus 设备的保持寄存器（Holding Registers）中。这样，传统的 Modbus-RTU 主机（如 PLC）只需通过读取虚拟设备，就能像读取普通 Modbus 从站一样，轻松读取到 LoRa 传感器的数值，甚至可以通过写入寄存器来控制 LoRa 终端。

## 1. 开启 MODBUS-RTU 服务 & 配置串口参数

在使用此功能前，请确保您已使用 RS485 线缆将网关的 A 和 B 端子正确连接至上位机的对应端子。

1. 在“应用”->“数据流控”页面左侧，点击进入“MODBUS-RTU 服务器”处理器。
2. 确保该处理器右侧的状态显示为绿色的“已开启”。

已开启

2. 在下方的“Modbus Rtu 串口配置”栏中，根据您上位机（PLC/触摸屏）的实际通讯参数进行严格对齐设置：

- 波特率：默认 9600（可根据实际情况修改）。
- 数据位：默认 8。
- 停止位：默认 1。
- 校验位：默认 无校验 (None)。

MODBUS-RTU 服务器 ● 串口已开启

Modbus Rtu 串口配置 停止监听

波特率  
9600

数据位  
8

停止位  
1

校验位  
无校验

参数确认无误后，点击右侧的“开始监听”按钮。

## 2. 编写寄存器映射脚本 (JS 编程)

由于不同终端的数据结构不同，我们需要通过脚本告诉网关：把传感器数据的哪一段，放到哪一个寄存器里。

1. 在顶部切换至“脚本”选项卡，点击“添加脚本”，为其命名（如“温度 MODBUS-RTU 映射”）。
2. 点击“</> 编程”进入代码编辑器。

G6X MODBUS - RTU编程 | 接口 终端 1 脚本 1

**MODBUS - RTU 脚本编辑** 添加脚本  
使用 javascript 映射寄存器 数据

**温度MODBUS-RTU映射** 编程 设备 1 删除  
MODBUS | @读写绑定设备的寄存器

系统同样提供了 G6x\_Modbus\_Read 和 G6x\_Modbus\_Write 两个核心函数。

```
温度MODBUS-RTU映射
1 // 自定义 MODBUS-RTU 读取脚本
2 var G6x_Modbus_Read = function(hexData, id, port) {
3   let data = {}; // 默认空对象
4
5   data = {
6     UnitId: id,
7     Reg0: hexData.slice(6, 10), // 温度
8     Reg1: hexData.slice(10, 14), // 湿度
9     Reg2: hexData.slice(14, 18), // PM2.5
10    Reg3: hexData.slice(18, 22), // PM10
11    Reg4: hexData.slice(22, 26), // CO2
12  };
13
14  return JSON.stringify(data);
15 };
16
17 // 自定义 MODBUS-RTU 下发脚本
18 var G6x_Modbus_Write = function(hexData) {
19   let data = {
20     Down_Payload: hexData.slice(0, 4),
21   };
22
23   return JSON.stringify(data);
24 };
25
```

保存 关闭

3. 编辑完成后点击“保存”。

### 3. 绑定终端并分配“Modbus 从站 ID”

为了让通过 485 串口连接的 PLC 能够区分多个无线传感器，需要为它们分配从站地址。

1. 在“脚本”选项卡中找到刚才编写的映射脚本。
2. 点击下方的“绑定设备”。
3. 在弹出的“MODBUS-RTU 设备添加”窗口中：



- 从站 ID (1-255): 输入一个在整个 485 总线上不冲突的地址，例如填写 1。
- 勾选设备: 选中对应的终端（例如 J32\_环境监测）。



4. 点击“添加设备”。

### 4. 外部系统读取验证

使用 USB 转 485 线连接网关与电脑。打开 Modbus Poll 测试软件。

建立连接：连接方式选择 Serial Port (串口)，选择正确的 COM 口，并设置与网关相同的波特率（如 9600, 8, N, 1）。

读取数据：

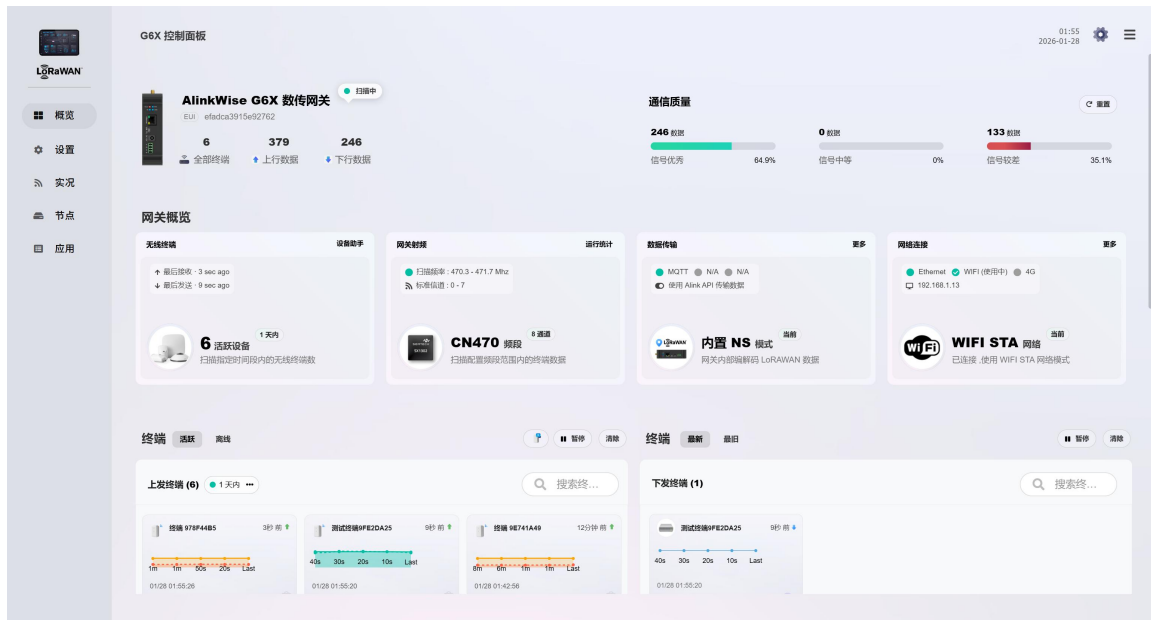
Slave ID (从站号) 填 1。

Function 选 03 Read Holding Registers。

开始读取。如果硬件接线和参数都正确，您就可以在串口的另一端读到无线的 LoRa 传感器数据了。

## 附件 1：网关概览页面 (Dashboard Overview)

成功登录后，您将看到直观的 **G6X 控制面板**。这里集中展示了网关的运行健康度、终端状态监控、射频数据模式及通信质量统计，是日常监控最常用的界面。



界面主要分为以下几个核心区域：

### 1. 数据统计区域

网关信息：显示 网关当前使用名称 (可自定义)、网关 EUI 以及 LoRa 射频状态 (显示 “扫描中” 绿色标签表示射频工作正常)。

数据统计：

- 全部终端：自网关运行/重置统计以来，接收到的所有 LoRaWAN **上报终端**的数量。
- 上行数据 (Uplink)：自网关运行/重置统计以来，网关接收到的上行数据包总数。
- 下行数据 (Downlink)：自网关运行/重置统计以来，网关发送给终端的指令包总数。
- 通信质量：网关接收数据的信号质量分布 (优秀/中等/较差)，帮助您判断网关/终端安装位置是否合理。

- 点击通信质量右侧的 **重置** 按钮，可重新开始统计 上行数据 / 下行数据 / 通信质量。



数据统计区域

## 2. 网关概览区域

该区域实时显示网关的 终端活跃数 / 射频工作参数 / 数据传输模式 / 网络链路状态



**无线终端卡片 (Devices):** 显示活跃设备数量及最后一次数据收发时间，方便快速判断是否有设备掉线。



- **最后接收:** 网关接收最后一次上行数据的时间记录，N/A 为从未收到过数据
- **最后发送:** 网关发送最后一次下行数据的时间记录，N/A 为从未发送过数据
- **活跃设备:** 统计指定时间内，**上报过数据的终端数量**，如图，即在 24 小时 (1 天内)，有 6 台活跃设备 (一天内上报过数据)。
- **指定时间** 可点击【**上发终端**】区域右侧的 **1 天内** 按钮进行修改

**网关射频卡片 (Radio):** 显示当前工作的 LoRaWAN 频段及信道配置，确保与您的终端频率一致。



- **扫描频率**：网关当前使用的 [起始频率-结束频率]，**终端与网关的频点对应才能成功通信。**
- **标准信道**：LoRaWAN 协议规定的标准信道序号，用于快速对应频率范围。
- **XXX 频段**：硬件支持的大范围物理频率区间，此项需要硬件支持，所以需要在设备购买前确定。如 EU433/CN470/CN779/EU868/US915/AU915/KR920/AS923

**数据传输卡片 (Mode)**：显示网关当前数据传输模式，以及对应传输模式的链路状态。



- **链路状态**：左上角 [状态灯]+链路协议。绿色表示可用/灰色表示不可用（断开或未配置）；使用内置 NS 模式时，3 个状态灯分别对应 Alink API-1,API-2,API-3 的链路状态。
- **数据模式**：显示当前网关使用的数据模式，可选择 **内置 NS 模式 (解码)** / **远程 NS 模式 (仅转发)**

**网络连接卡片 (Radio)**：显示网关当前使用的联网方式，以及网络链路的可用性。



- **网络状态**：网关当前的网络可用性，**[状态灯]+网络类型**。以及显示正在优先使用的网络模式。
- **设备 IP**：网关被分配的本地网络 IP 地址。

### 3. 终端概览区域

在网关概览页面的下半部分，提供了一个直观的“终端监控看板”。该区域显示网关已处理的**上行/下行终端**的通信状态统计信息，该看板以卡片的形式，实时展示了当前接入网关的所有终端设备的通信健康状态，帮助您快速定位可能存在的网络故障。

#### 1.状态筛选：

- **活跃**：仅显示在时间筛选内有数据上报的正常设备（如 时间筛选为 10 分钟，显示 10 分钟内，**上报过数据**的所有终端）。
- **离线**：仅显示时间筛选范围内，尚未上报任何数据的故障设备（如 时间筛选为 10 分钟，显示 10 分钟内，**未上报过数据**的所有终端）。

#### 2.时间筛选：

- 点击下拉菜单，可选择统计的时间范围（如 10 分钟、1 小时、1 天内 等）。选择后，下方卡片中的“消息统计”和“图表”将仅会显示**该时间段内活跃终端（该时间内上报过数据）**的数据表现。

#### 3.搜索与控制：

- **搜索框**：支持通过终端名称或 DevEUI 模糊搜索特定设备。
- **暂停/清除**：点击“暂停”可暂时冻结界面的数据刷新，方便截图或分析；点击“清除”可清空当前的统计缓存（不影响历史数据）。



## 设备通信详情卡片

每个注册的终端都会生成一张独立的状态卡片（如图所示），卡片内的各项信息含义如下：

1. 设备名称与状态：左上角显示您为设备自定义的名称（如 测试终端 OTAA）。右上角实时显示该设备最后一次通信距今的时间（如 9 秒 前 ，绿色箭头代表上行数据）。

2. 最近数据概览 (时间轴图表)

这是一条上报数据的动态时间轴（如 40s - 30s - 20s - 10s - Last，直观展示该节点的最近 5 个数据包，上报之间的间隔为 10 秒，即该节点的上报周期为 10 秒）。

实线 (直线)：代表接收信号强度指示 (RSSI)。

虚线：代表信噪比 (SNR)。

颜色指示 (核心)：图表的颜色直观反映了当前的无线通信质量，帮助您快速判断是否需要人工干预：

**绿色**：通信质量优秀。信号强且底噪低，链路非常稳定，无需任何关心。

**黄色**：通信质量一般。信号可能存在一定的衰减或干扰，但不影响正常的数据传输。

**红色**：通信质量较差。处于随时可能断连的边缘，需要排查调整（例如：检查天线是否被金属遮挡、增加网关数量或调整设备的安装位置）。

3.最后上报时间

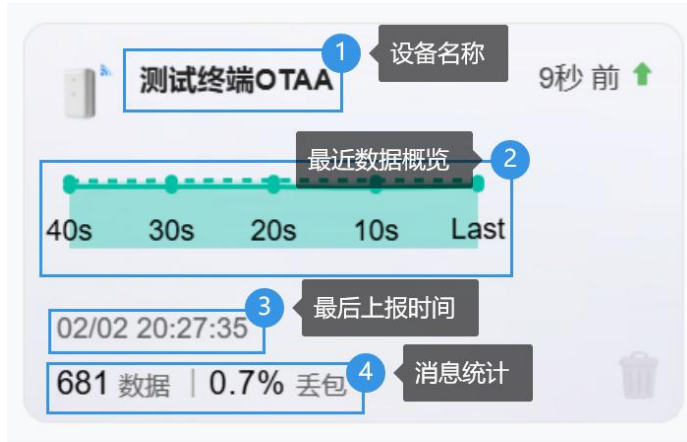
精确记录网关最后一次收到该终端数据的具体日期和时分秒（如 02/02 20:27:35），用于精确对账。

4.消息统计 (通信质量核心指标)

数据总量：显示在所选时间筛选范围内（如最近 10 分钟内），网关共收到该设备的有效数据包总数（如 681 数据）。

丢包率：这是评估无线信号质量的最关键指标。通过对比设备上报的帧序号（FCnt）的连续性，网关自动计算出的丢包比例（如 0.7% 丢包）。

通过这个看板，用户无需深入复杂的日志，只需一眼扫过卡片的颜色和丢包率，就能迅速掌握数十个传感器的网络健康全貌。



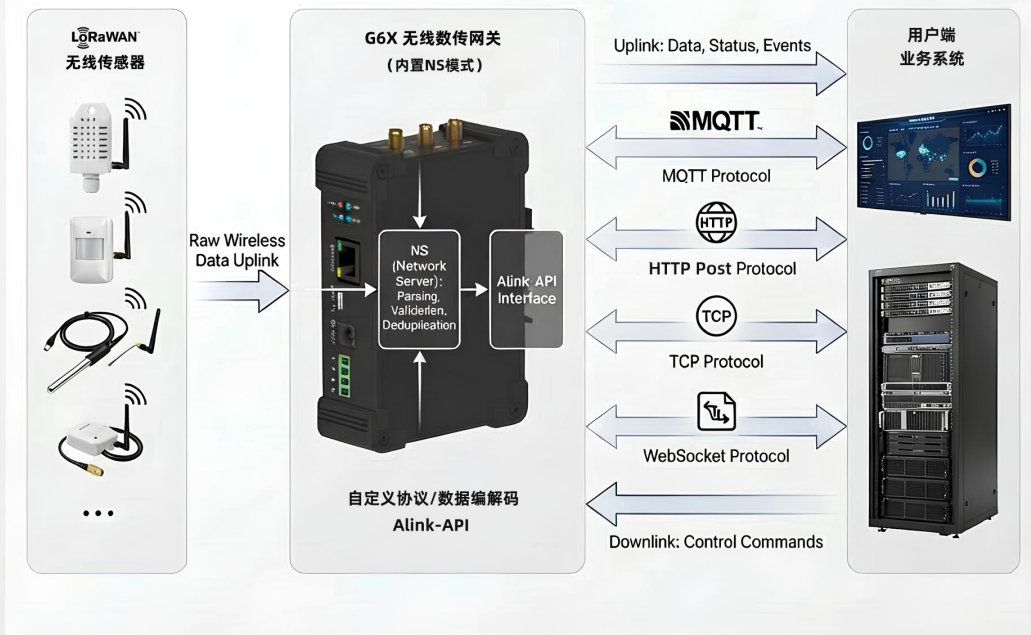
## 附件 2：网关 API 接口协议及报文定义

### 内置 NS 模式 | 数据接口 (Alink-API)

G6xx 网关内置 **NS (Network Server, 网络服务器)** 功能，可对接入的 LoRaWAN 无线节点进行统一管理与数据处理。节点的数据在进入网关后，会先由网关内部的 NS 进行解析、校验和编码，再通过网关自带的 Alink-API 接口与用户系统进行数据交互。

Alink-API 支持使用 **MQTT / HTTP / TCP / WebSocket** 协议，用户可以方便地获取节点的**上行数据、设备状态、事件信息**，同时也可以向节点发送**下行指令**，实现业务系统与 G6xx 网关设备之间的双向通信。

## G6X Gateway: Alink-API 数据接口



### Alink-API 协议类型配置

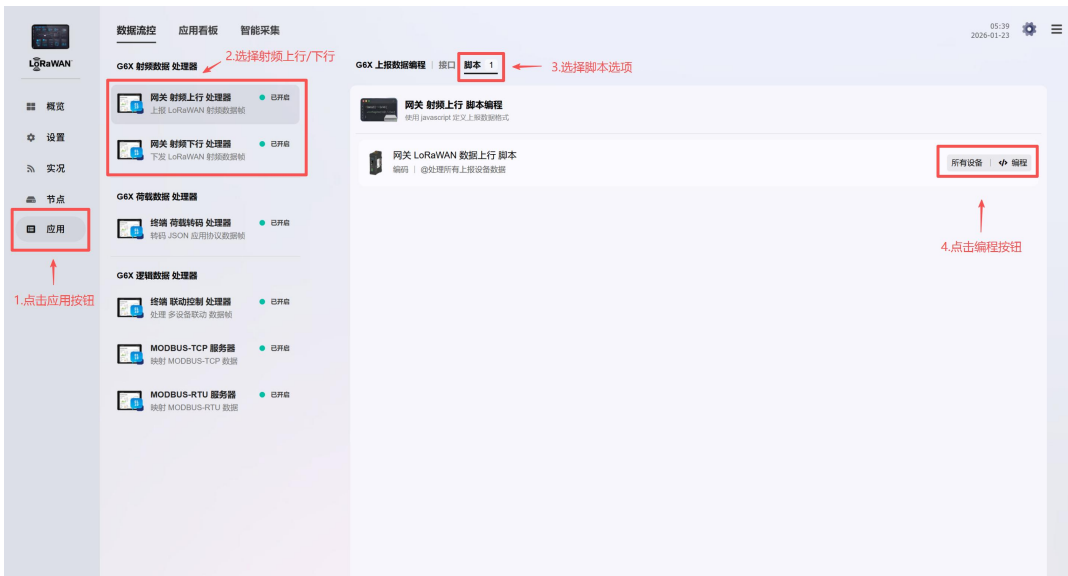
在网关配置页面最左侧的主导航栏中，点击 设置>数据传输>网关数据模式(内置 NS)

在同一页面向下滑动，即可找到“Alink API 数据接口”标题。在此区域，您可以开启 Alink API 并配置协议 (MQTT/HTTP/TCP/WS) 类型及服务器地址，数据主题等参数。



## Alink-API 报文内容自定义 可编程

在网关配置页面最左侧的主导航栏中，点击 应用>网关 射频上行/下行处理器>脚本 在同一页面右侧，即可找到“编程”按钮。点击按钮后，您可以开启 Alink API 代码编辑器，通过 Javascript 脚本编码，使网关产生和发送 自定义 上行/下行 数据格式报文。



## 终端 - 射频上行报文 / Uplink



当 LoRaWAN 终端设备完成数据采集并向网关上报时，网关在成功接收到该数据后，会生成一条 LoRaWAN 射频上行报文 (Uplink)。

该数据报文由网关内置 NS 统一处理后，通过 API 接口上报给用户系统。

通过该报文，用户系统可获取设备上报的数据内容及通信状态，用于业务处理、数据展示和设备监控。

上行报文 MQTT 默认主题 (可自定义)：/gateway/{网关 EUI}/ns/up1

上行 JSON 报文 默认格式 (可自定义)：

```
{
  "applicationId": "0", // 应用 ID
  "applicationName": "G6x_LoRaWAN_Uplink", // 应用名称
  "deviceName": "测试终端 OTAA", // 设备名称
  "devEui": "9cc466c19fe2da25", // 设备唯一标识 EUI
  "fCnt": 2, // 上行帧计数
  "fPort": 3, // 数据端口号
  "payload": "EjRWeJCrzQ4=", // 原始上行数据 (Base64 编码)
  "data": "1234567890abcd0e", // 原始上行数据 (字符编码)
  "rxInfo": { // 接收信息
    "gatewayId": "639c6847407c7c4c", // 接收网关 EUI
    "rssi": -39, // 信号强度
    "snr": 13.8 // 信噪比
  },
  "txInfo": { // 无线参数
    "frequency": 471.7, // 工作频点 (MHz)
    "modulation": {
      "lora": {
        "spreadingFactor": "SF7BW125", // 扩频因子/带宽
        "codeRate": "CR_4_5" // 编码率
      }
    }
  }
}
```

## 终端 - 射频下行报文 / Downlink

MQTT

TCP

WS

可编程

当用户系统需要向 LoRaWAN 终端设备发送控制指令或配置数据时，需要向网关推送一条 LoRaWAN 射频下行报文 (Downlink)。

该数据帧通过用户下发，由 API 接口**下发**给网关内置 NS 进行处理，并发送至指定的终端设备。  
通过该报文，用户系统可以实现对终端设备的远程控制和参数配置。

下发报文 MQTT 默认主题（可自定义）： /gateway/{网关 EUI}/ns/down1

下发报文 JSON 默认格式（可自定义）：

```
{
  "deveui": "9fe2da259fe2da25",           // 设备唯一标识 EUI
  "data": "1234567890abcdef",           // 自定义下发数据内容（字符编码）
  "fPort": 2,                            // 数据端口号，默认 2
  "retry": 1,                            // 数据下发次数，默认 1
  "confirmed": false                     // 终端是否需要在下一次上报中确认
}
```

## 终端 - OTAA 入网报文 / Join

MQTT

HTTP

TCP

WS

当 LoRaWAN 终端设备采用 OTAA 方式完成入网流程，并成功加入网络后，网关会生成一条 OTAA 入网事件报文（Join Event）。

该报文用于通知用户系统设备已成功入网，通过 API 接口**上报**给用户系统。

通过该事件报文，用户系统可以感知设备入网状态，用于设备初始化、状态更新及运维管理。

上报报文 MQTT 默认主题（可自定义）： /gateway/{网关 EUI}/ns/join

上报报文 JSON 数据格式（不可修改）：

```
{
  "ApplicationID": "0",                   // 应用 ID
  "ApplicationName": "JoinEvent",        // 应用名称
  "DeviceName": "测试终端 OTAA",        // 设备名称
  "DevEUI": "9cc466c19fe2da25",         // 设备唯一标识 EUI
  "DevAddr": "c0152925",                 // OTAA 入网为设备分配的短地址
  "Time": "2026-01-20T23:53:25.475Z"    // OTAA 入网时间
}
```

## 终端 - 确认回复报文 / Ack

MQTT

HTTP

TCP

WS

当 LoRaWAN 终端设备接收到一条 需要确认的下发数据帧（Confirmed Downlink）后，终端会在下一次上行通信时携带确认标志进行回复。

网关在接收到该上行数据后，将整理为一条确认信息（Ack Event）并通过 API 上报给用户系统。  
通过该事件数据帧，用户系统可判断下发指令是否被终端成功接收，用于下发结果确认和通信状态分析。

上报报文 MQTT 默认主题（可自定义）：/gateway/{网关 EUI}/ns/ack

上报报文 JSON 数据格式（不可修改）：

```
{
  "applicationID": "0", // 应用 ID
  "applicationName": "AckEvent", // 应用名称
  "deviceName": "测试终端 OTAA", // 设备名称
  "devEUI": "9cc466c19fe2da25", // 设备唯一标识 EUI
  "fCnt": 10, // 确认的下行计数
  "acknowledged": true, // 确认标志，固定为 true
  "time": "2026-01-21T02:31:07.656Z" // 确认时间
}
```

## 终端 - 状态查询 / Request+Response

### MQTT

当用户系统需要查询当前网关下已接入设备的状态和基础信息时，仅可通过 MQTT 请求方式向网关下发设备信息查询请求（Request）。

网关在接收到请求后，会对请求进行处理，并通过 MQTT 返回对应的设备信息响应数据（Response）。  
通过该请求与响应机制，用户系统可以实时获取设备运行状态，用于设备管理、状态监测和运维分析。

### 第一步：用户通过 MQTT 主题下发 Request 请求

下发请求 MQTT 数据主题（不可修改）：gateway/{网关 EUI}/api/request

下发请求 JSON 数据格式（不可修改）：

```
{
  "id":"req-1", // 请求 ID，网关将回复相同 ID
  "method":"GET", // 请求方法
  "path":"/api/cot/Dev/DeviceList", // 请求的接口路径
  "headers":{}, // 头信息，默认为空
  "body": "", // 请求体，默认为空
  "query":{} // 查询参数，默认为空
}
```

## 第二步：网关 MQTT 主题上报 Response 回复用户

上报回复 MQTT 数据主题（不可修改）：gateway/{网关 EUI}/api/response

上报回复 JSON 数据格式（不可修改）：

```
{
  "id": "req-1", // 响应的请求 ID, 与 Request 中的 id 一致
  "status_code": 200, // 请求处理结果状态码
  "headers": {
    "Access-Control-Allow-Origin": "*", // 响应头信息
    "Content-Type": "application/json" // 数据格式
  },
  "body": {
    "code": 0, // 业务处理结果 (0 表示成功)
    "data": [
      {
        "name": "J32_环境监测", // 设备名称
        "devEUI": "9135bfd79dfdafb3", // 设备唯一标识 EUI
        "lastSeen": "N/A", // 最近一次上行时间
        "lastSeenDown": "", // 最近一次下行时间
        "isOtaa": true, // 是否为 OTAA 入网方式
        "fCntUp": 0, // 上行数据帧计数
        "class": "C", // 设备类型 (Class A/B/C)
        "alive": false, // 当前是否在线
        "id": 0, // 预留字段
        "javaName": "" // 预留字段
      },
      {
        "name": "测试终端 OTAA",
        "devEUI": "9cc466c19fe2da25",
        "lastSeen": "2026/01/21 10:31:07",
        "lastSeenDown": "",
        "isOtaa": true,
        "fCntUp": 13,
        "class": "A",
        "alive": true,
        "id": 1768962667,
        "javaName": ""
      }
    ]
  }
}
```

```
],  
  "msg": "ok"  
},  
"error": ""  
}
```

## 网关 - 状态报文 / Status

MQTT

HTTP

TCP

WS

网关在正常运行过程中，会按照设定的周期主动上报心跳状态报文，用于告知用户系统当前网关仍处于在线运行状态。

该报文用于网关运行状态监测，不涉及具体设备或业务数据。

通过该心跳状态报文，用户系统可以实时掌握网关的在线情况，用于运行监控和运维管理。

上发报文 MQTT 默认主题（可自定义）：gateway/event/stats

上发报文 JSON 数据格式（不可修改）：

```
{  
  "gatewayID": "639c6847407c7c4c", // 网关唯一标识 (Gateway EUI)  
  "time": "2025-05-18T10:28:25Z", // 心跳上报时间 (UTC, ISO 8601 格式)  
  "status": "00 天 16 时 09 分 13 秒" // 网关运行时长  
}
```