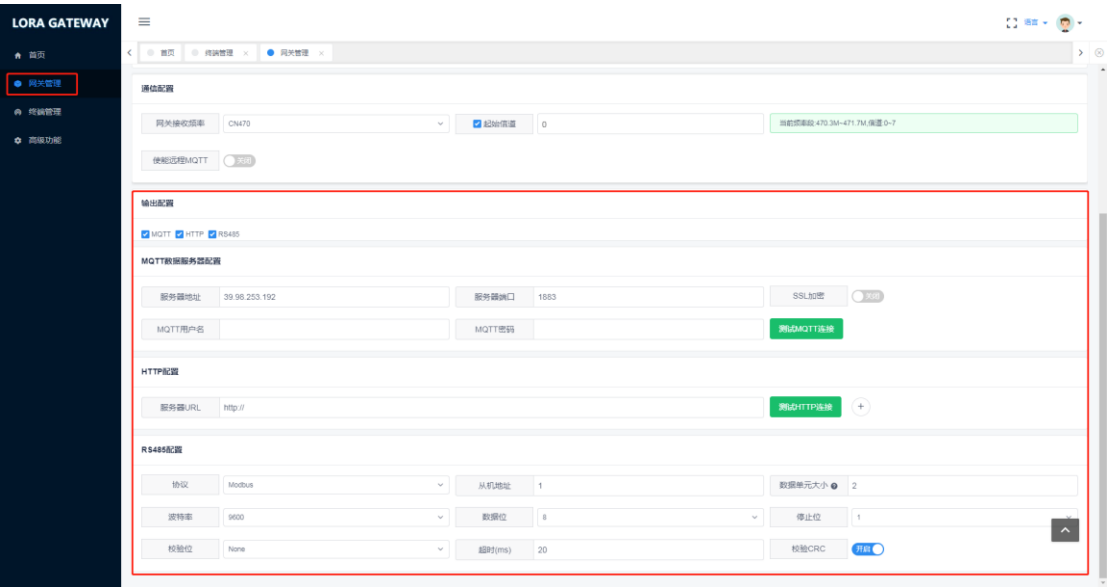


边缘网关接口文档

1.接口说明

边缘网关目前仅开放数据接口，数据接口支持 HTTP、MQTT 和 RS485 三种方式，HTTP 数据仅支持推送，因为一般边缘网关部署和应用服务器不会在一个局域网中，一般是边缘网关能直接访问到服务器，但服务器不能直接访问到边缘网关。因此我们建议使用 MQTT 形式的数据上发和下发。RS485 需室内网关硬件支持即可，有 Modbus 和 Private 两种通信协议。

其接口配置在网关管理->网关配置中，如下图



输出配置是选择 MQTT、HTTP 和 RS485 三种方式，出厂默认勾选 MQTT，以及配置好了我们测试服务器的地址。若三个都不勾选则数据只在网关内部，可通过网关实时数据(原始数据)和终端实时数据(解析后的数据)查看。

2.数据接口

2.1 MQTT 数据接口

2.1.1 主题介绍

边缘网关原生接口：

/gateway/{gatewayId}/up

//上行数据，订阅

/gateway/{gatewayId}/down //下行数据，发布

以下两个接口是兼容标准开源(chirpstack)服务器的接口：

application/0/device/{DevEUI}/rx //上行数据，订阅

application/0/device/{DevEUI}/tx //下行数据，发布

其中{gatewayId}为网关 EUI 在，在网关管理->网关配置中可以查看。

{DevEUI}为终端 EUI。两类接口可混合使用。

订阅时{gatewayId}和{DevEUI}可以变为加号“+”，为 MQTT 的模式匹配，表示订阅所有网关及终端数据。

2.1.2 终端上行数据

2.1.2.1 原生接口

主题 /gateway/{gatewayId}/up

数据格式如下：

```
{
  "app": {
    "moteeui": "00000000995477ea", //终端 EUI
    "dir": "up", //上行
    "userdata": { //用户数据
      "seqno": 1, //上行计数
      "port": 12, //端口
      "payload": "MTIz" //base64 后的应用数据
    },
    "motetx": { //终端发送的射频信息
      "freq": 471.5, //频率，单位 MHz
      "modu": "LORA", //LORA 调制方式
      "datr": "SF10BW125", //扩频因子及带宽
      "codr": "4/5", //编码率
      "adr": false //固定 false
    },
    "gwrx": [ //网关接收信息
      {
        "eui": "6eb1b8ae65155a4d", //网关 EUI
        "time": "2022-04-07T10:01:44.522700", //收包时间
        "timefromgateway": true,
        "chan": 6,
        "rfch": 1,
        "rssi": -23, //信号强度
        "lsnr": 13 //信噪比
      }
    ]
  }
}
```

```

    ]
  }
}

```

2.1.2.2 兼容接口

主题 application/0/device/{DevEUI}/rx

数据格式如下：

```

{
  "applicationID": "0", //固定 0
  "applicationName": "0", //固定 0
  "deviceName": "00000000995477ea", //终端 EUI
  "devEUI": "00000000995477ea", //终端 EUI
  "rxInfo": [ //网关接收信息
    {
      "gatewayID": "6eb1b8ae65155a4d", //网关 EUI
      "name": "Edge", //固定 Edge
      "time": "2022-04-07T10:01:44.5227+08:00", //收包时间
      "rssi": -23, //信号强度
      "LoRaSNR": 13 //信噪比
    }
  ],
  "txInfo": { //终端发送的射频信息
    "frequency": 471500000, //频率，单位 Hz
    "dr": 0 //数据速率，0 表示 SF12 直到 5 表示 SF7
  },
  "adr": false, //固定 false
  "fCnt": 1, //上行计数
  "fPort": 12, //端口
  "data": "MTlz", //base64 后的应用数据
  "devAddr": "995477ea" //终端短地址
}

```

2.1.3 下发数据

2.1.3.1 原生接口

主题 /gateway/{gatewayId}/down

数据格式如下：

```

{
  "app": {

```

```

    "moteeui": "00000000995477ea", //终端 EUI
    "token": 5053000, //非必填, CN470 默认 505300000Hz 和 SF12, token 是频率和
    SF 的组合, 不推荐, 为兼容老设备保留, 建议用 freq 和 sf 字段代替
    "freq": 505300000, //非必填, CN470 默认 505300000Hz
    "sf": 12, //非必填, CN470 默认 SF12, 取值范围 7-12
    "bandwidth": 0, //非必填, 默认 0 表示 125k 带宽, 1 表示 250k, 2 表示 500k 带
    宽
    "userdata": {
        "port": 2, //端口
        "payload": "dGVzdA==" //base64 后的应用数据
    },
}
}

```

对于 ClassA 类型的终端, 下发频率、扩频因子 (SF)、带宽这三个参数是无效的, 因为 LoRaWAN 协议中规定 ClassA 类型终端这三个参数是根据终端上行数据来计算的。

对于 ClassC 类型的终端, CN470 默认下发频率是 505300000 和 SF12, 带宽 125k, 若有特定应用可修改对应参数。

注: token 的最后一位表示 SF,取值范围是 0-5 依次对于 SF12-SF7,剩下的高位是表示频率 kHz, 如, 下发频率是 505300000Hz, SF7 的话则 token 为 5053005。已不推荐使用, 若同时存在 token、freq、sf 字段, 优先使用 token

2.1.3.2 兼容接口

主题 application/0/device/{DevEUI}/tx

数据格式如下:

```

{
    "devEUI": "00000000995477ea", 非必填, 不填时取主题上的 DevEUI
    "fPort": 1, //端口
    "data": "aGVsbG8=" //base64 后的应用数据
}

```

兼容接口下发的频率、扩频因子、带宽均为默认参数, 同原生接口

2.1.4 组播下发数据

主题同上面的单播下发 /gateway/{gatewayId}/down

数据格式如下:

```

{
    "app": {
        "groupAddr": "ffffff", //组播地址, 必须去掉 moteeui 字段, 如果存在则以单播下
        发, 仅该字段与单播不同
        "token": 5053000, //非必填, CN470 默认 505300000Hz 和 SF12, token 是频率和
        SF 的组合, 不推荐, 为兼容老设备保留, 建议用 freq 和 sf 字段代替
        "freq": 505300000, //非必填, CN470 默认 505300000Hz
    }
}

```

“sf”: 12, //非必填, CN470 默认 SF12, 取值范围 7-12
“bandwidth”: 0, //非必填, 默认 0 表示 125k 带宽, 1 表示 250k, 2 表示 500k 带

宽

```
"userdata": {  
  "port": 2, //端口  
  "payload": "dGVzdA==" //base64 后的应用数据  
},  
}
```

2.2 HTTP 上行数据接口

对于 HTTP 接口方式需填入自定义应用服务器的 URL 地址, 该 URL 需支持 POST 方法, 若存在认证机制, 如 token, 可以点击添加请求头, 在 HTTP 的 Header 中添加对应的信息。

由于终端上行数据是采用网关推送的方式, 要求 URL 为在创建应用时输入的自定义 HTTP 服务器地址和接口名称, 如 <http://example.com/uplink>

要求该接口支持 POST 方法

默认只推送为原生接口的上行数据格式, 数据格式同 MQTT 接口。

若要推送标准开源接口的上行数据格式, 需在 url 上加 compatOpen=true 参数。

例如: <http://example.com/uplink?compatOpen=true>

2.3 RS485 数据接口

2.1 Modbus

Modbus 目前仅支持 0x03、0x04 和 0x10 这三个功能码, 其他功能码会返回错误。如果 CRC 校验错误将不返回任何数据, 可以在设置中关闭 CRC 校验。从机地址也可在网关设置中配置。

其中终端寄存器数, 在 Modbus 中是设定一个终端可以表示几个寄存器, 默认 1 个终

端 1 一个寄存器，所有终端的按序号连续排列，寄存器地址也是连续的，例如设定每个终端有 4 个寄存器，则映射如下：

终端序号	寄存器地址映射
0	00 01 02 03
1	04 05 06 07
2	08 09 0A 0B

每个终端的第 1 个寄存器地址=终端序号*终端寄存器数

当执行 0x03 读多个寄存器时，寄存器开始地址建议从终端的第 1 个寄存器地址开始读，也可从任意地址开始，如从序号为 0 的终端的 03 开始读 2 个寄存器，则会把序号为 0 的终端最后 1 个寄存器和序号为 1 的终端的第 1 个寄存器数据返回，共返回 4 字节数据。如果超出寄存器映射地址则返回错误，如从 0B 开始读两个寄存器。若序号为 1 终端缺失，则该地址映射也缺失，读取该地址会返回错误。

当执行 0x04 读多个寄存器时，每个终端默认固定一个寄存器，表示终端是否在线。

当执行 0x10 写多个寄存器时，寄存器开始地址必须从终端的第 1 个寄存器地址开始写，且写入的寄存器数必须是终端寄存器数的整数倍，即可从 00 开始写入 4*n 个寄存器，若从 00 开始写入 3 个寄存器会返回错误。

其中终端序号在终端管理中可查看

2.1.1 读取终端数据

以下操作都假定每个终端有 4 个寄存器
主机发送

从机地址	功能码	寄存器起始地址	寄存器个数	CRC 低位	CRC 高位
0x01	0x03	0x00 00	0x00 08	0x44	0x0C

完整的序列命令：01 03 00 00 00 08 44 0C，该命令会读取序号为 0 和 1 两个终端的数据

网关回复

从机地址	功能码	数据字节数	寄存器数据	CRC 低位	CRC 高位
0x01	0x03	0x10	39 30 61 62 63 64 65 66	0x82	0xC9

			31 32 33 34 35 36 37 38		
--	--	--	-------------------------	--	--

完整的序列数据：01 03 10 39 30 61 62 63 64 65 66 31 32 33 34 35 36 37 38 82 C9

其中数据为每个终端最后一次上行的数据，即将序号为 0 和 1 的两个终端最后一次上行数据拼接为一个数据包返回。

2.1.2 写入终端数据

主机发送

从机地址	功能码	寄存器起始地址	寄存器个数	数据字节数	写入数据	CRC 低位	CRC 高位
0x01	0x10	0x00 00	0x00 08	0x10	39 30 61 62 63 64 65 66 31 32 33 34 35 36 37 38	0x7F	0x5D

完整的序列命令：01 10 00 00 00 08 10 39 30 61 62 63 64 65 66 31 32 33 34 35 36 37 38 7F 5D

网关回复

从机地址	功能码	寄存器起始地址	寄存器个数	CRC 低位	CRC 高位
0x01	0x10	0x00 00	0x00 08	0xC1	0xCF

完整的序列数据：01 10 00 00 00 08 C1 CF

上例中会给序号为 0 和 1 的终端分别下发 39 30 61 62 63 64 65 66 和 31 32 33 34 35 36 37 38，在网关管理中可查看下发数据，注意仅 ClassC 类型终端会立即下发，ClassA 类型终端会等终端有上报后才会下发。

The screenshot displays the 'LORA GATEWAY' management interface. The 'Gateway Management' tab is selected, and the 'Gateway Real-time Data' sub-tab is active. A table lists the following data:

数据方向	数据类型	终端地址	计数	端口	MIC校验	上传时间
> downlink	UnconfirmedDataDown	00232074	2	1	b96714e	2022-08-25 09:54:32.444
> downlink	UnconfirmedDataDown	0023161a	2	1	03181ca4	2022-08-25 09:54:30.781

2.1.3 读取终端在线状态

主机发送

从机地址	功能码	寄存器起始地址	寄存器个数	CRC 低位	CRC 高位
0x01	0x04	0x00 00	0x00 06	0x70	0x08

完整的序列命令：01 04 00 00 00 06 70 08，该命令会读取序号为 0 到 5 六个终端的在线状态

网关回复

从机地址	功能码	数据字节数	寄存器数据	CRC 低位	CRC 高位
0x01	0x04	0x0C	00 01 00 01 00 01 00 00 00 00 00 00	0x8C	0x1B

完整的序列数据：01 04 0C 00 01 00 01 00 01 00 00 00 00 00 8C 1B

例子中，前 3 个寄存器位值为 1，表示前 3 个终端在线，后 3 个为 0，表示离线

2.2 Private

Private 协议与 Modbus 不同之处在于，Modbus 是请求响应式，即必须要主机发送指令，从机才会回复，而 Private 协议是终端数据上行主动推送，给终端下发数据后也无回复。

另 Private 中是将每个终端序号作为地址，一次推送或下发一个终端数据，终端序号为小端模式

2.2.1 终端上行数据

从机地址	终端序号	数据长度	数据	CRC 低位	CRC 高位
0x01	0x01 00	0x33	31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31	F9	EB

完整包结构：01 01 00 33 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 F9 EB

2.2.2 终端下行数据

从机地址	终端序号	数据长度	数据	CRC 低位	CRC 高位
0x01	0x01 00	0x03	01 02 03	1A	F4

完整包结构：01 01 00 03 01 02 03 1A F4

附录：MQTT 测试工具

推荐使用 mqtt.fx 工具，目前最新版 1.7.1。

官方下载地址为 <http://www.jensd.de/apps/mqttfx/1.7.1/>

可根据操作系统自行下载

a) 配置

