
MARS TCP 通信接口协议



杭州锐讯科技有限公司

中国 杭州

版本历史

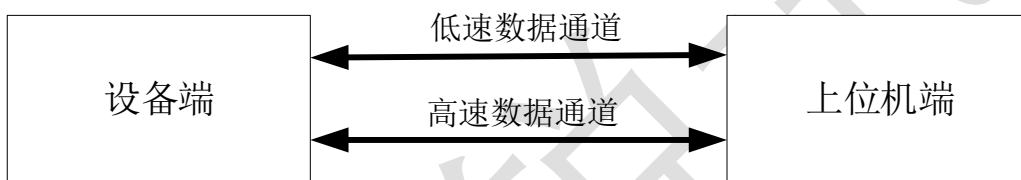
| 版本 | 日期 | 备注 |
|------|------------|---------|
| V1.0 | 2021-04-23 | 初版 |
| V1.1 | 2021-09-24 | 增加数据包示例 |
| | | |
| | | |
| | | |
| | | |

1、综述

本文描述设备端（指 MARS 设备）与上位机（指控制设备）间的 TCP 数据通信细节。

1.1 数据通道

设备端和上位机端间存在两个通道，分别是用于配置和查看设备运行的低速数据传输通道和实现原始采集数据实时上传的高速数据传输通道。



低速数据通道传输配置或查询等低速率数据帧，采用一应一答的方式进行，设备端接收到命令数据帧后，会在超时时间内回复。如果上位机在超时时间内未接收到回复帧，则需要重发，如果达到重发次数后设备仍未回复，则认为通道链路断开或者设备故障。

高速数据通道则用于设备端将采集的原始数据实时发送给上位机端。高速数据通道需要较高的带宽来实时传输大量数据。

1.2 通信接口

采用 TCP 通信协议，命令端口和数据端口分开。设备端为 TCP 服务器端，上位机为 TCP 客户端，上位机主动发起 TCP 连接，控制通道及数据通道的使用方法一致。端口分配是固定的，命令端口为 7777，数据端口为 7778。

1.3 数据帧结构

对具体的数据进行封帧处理，具体的帧结构如下：

| 序号 | 字段名称 | 字节数 | 说明 |
|----|--------|-----|--|
| 1 | 开始字节 | 2 | 固定为 0xFE 0xFE |
| 2 | 帧长度 | 2 | 整个帧的字节数，包含开始字节和校验和 |
| 3 | 协议版本 | 2 | 协议的版本号说明： 高字节代表协议大版本 低字节代表协议小版本 |
| 4 | 帧事务号 | 1 | 用于确认数据帧发送的顺序，发送方填充该事务号，回复方使用该事务号进行回复 |
| 5 | 源地址 | 1 | 发出本数据帧的设备地址编号 |
| 6 | 目的地址 | 1 | 期望接收本数据帧的设备地址编号 |
| 7 | 帧类型 | 1 | 帧类型字段说明： 最高位代表通信方向，置 0 表示上位机至设备端，置 1 表示设备端至上位机； 次高位代表错误指示，置 0 表示是正常的回复帧，置 1 表示是一个错误回复帧； 余下 6 位代表本数据帧类型。 |
| 8 | CRC 检验 | 2 | 数据 CRC 检验值 计算整个数据帧的 CRC 值 |
| 9 | 数据内容 | 变长 | 需要发送的数据内容 |

协议约定：

1. 每帧数据长度不超过 1200 个字节；
2. 如未特殊说明，数据内容都为小端字节序；
3. 协议版本暂定为 0x0001；
4. 数据内容中的保留部分应填充 0，以保持兼容性；
5. 源地址和目的地址保留，使用零填充；
6. 数据内容必须 2 字节对齐；
7. CRC 计算见第 3 章。

帧类型定义

| 值 | 说明 |
|------|---------|
| 0x00 | 心跳帧 |
| 0x01 | 配置帧 |
| 0x02 | 实时预览数据帧 |

2、数据帧描述

以下数据帧描述只包含数据内容的描述。

2.1 低速数据通道数据帧

控制通道数据帧包括链路维护帧（心跳帧）和配置帧。控制通道数据帧遵循一问一答的模式，由上位机主动发起，设备完成处理后进行回复。自定义的低速数据通道数据帧可以约定无需回复。

链路维护

心跳帧用于进行数据链路的维护。心跳帧由上位机主动发起，每隔一段时间需要主动发送一帧。

心跳帧字段说明（帧类型 0x00）

| 序号 | 字段名称 | 字节数 | 说明 |
|----|--------|-----|-----------------|
| 1 | 特定标识 | 4 | 0x12345c5c |
| 2 | 保留 | 4 | |
| 3 | 当前时间信息 | 4 | 当前的 4 字节 UTC 时间 |

设备端在接收到上位机发送的心跳帧后，立刻回复，如果上位机连续几次在超时时间内未接收到心跳回复帧，则认为链路断开。

设备端同时维护对心跳帧的计数，如果超过一定时间未收到过上位机的心跳

帧，会认为链路断开。

心跳帧回复帧字段说明（帧类型 0x80）

| 序号 | 字段名称 | 字节数 | 说明 |
|----|---------|-----|---|
| 1 | 设备本地时间 | 4 | 设备当前的 4 字节 UTC 时间 |
| 2 | 采样状态 | 1 | 0: 设备未配置任何有效采样计划（指设备未配置周期或分段采样任务） 1: 设备处于采样过程中（数据采集中） 2: 处于有效采样计划中的等待采样启动状态（处于周期或分段采样时的空闲时间状态） 3: 启动采样失败，处于失败后的重试状态 4: 启动失败，不再重试 其他值保留 |
| 3 | 保留 | 3 | 0 |
| 4 | 已采样时长 | 4 | 本周或本时段开始采样时刻到当前时刻的时间间隔 |
| 5 | 剩余存储容量 | 4 | 设备当前剩余存储容量，单位 MB |
| 6 | 设备可配置状态 | 1 | 0: 可配置，此时可以对设备进行配置操作 1: 配置任务执行中，此时不可以对设备进行配置操作 2: 设备启动中，此时不可对设备进行配置操作 3: 设备忙，此时不可对设备进行配置操作 其他值保留 |

| | | | |
|----|--------|----|---|
| 7 | 设备异常状态 | 1 | 0: 设备不存在异常状态 1: 设备当前时间信息存在异常（与主设备时间存在较大差异，现约定大于 10 秒） 其他值保留 |
| 8 | 保留 | 6 | 保留字段 |
| 9 | 电池电压 | 4 | 低 2 字节有效，单位为 mV |
| 10 | 存储容量 | 4 | 设备配置的总存储容量，单位为 MB |
| 11 | 错误码 | 4 | 为 0 表示没有错误，其他值代表特定的错误 |
| 12 | 错误参数 | 4 | 与错误码结合，指示设备当前错误信息 |
| 13 | 保留 | 32 | 保留字段 |

系统配置

上位机使用系统配置帧配置设备运行参数及控制设备的运行，并通过设备的配置回复帧获取设备运行状态。

系统配置帧字段说明（帧类型 0x01）

| 序号 | 字段名称 | | 字节数 | 说明 |
|-----|--------|-----|-----|---------------------|
| 1 | 配置参数个数 | | 1 | 本配置帧包含的配置参数个数 |
| 2 | 保留 | | 3 | 保留字段 |
| 3 | 配置参数 1 | 类型 | 2 | 指示配置参数的类型，详见“配置参数表” |
| | | 保留 | 2 | 保留字段 |
| | | 值 | 4 | 配置参数帧 |
| 4 | 配置参数 2 | ... | | |
| ... | | | | |

配置参数表

| 类型 | 值 |
|----|---|
|----|---|

| 值 | 说明 | |
|----|--------------|--|
| 0 | 旨在读取配置参数 | 0 |
| 1 | 设置系统时间 | 4 字节 UTC 时间 |
| 2 | 配置采样模式 | 0: 手动采样模式 1: 分段采样模式 2: 周期采样模式 其他值保留 |
| 6 | 采样率 | 设备所支持的采样率 其他值保留 |
| 7 | 增益 | 0: 增益为 0dB 1: 增益为 20dB 2: 增益为 26dB 3: 增益为 30dB 其他值保留，如果设备不支持可变增益，则需要设置为 0 |
| 8 | 用于控制设备执行对应命令 | 详见“系统命令表” |
| 9 | 设置设备 IP 地址 | 4 字节表示的 IP 地址 10.13.1.11:0xA(BYTE3)-0xD(BYTE2)-0x1(BYTE1)-0xB(BYTE0) |
| 10 | 设置设备网关地址 | |
| 11 | 设置设备子网掩码 | |
| 12 | 预览通道位掩码 | 每个 bit 位代表一个通道，如果为 1 表示该预览通道实时数据，否则关闭该通道的预览，共 96 个通道。 如果为 UNAS 设备，则只存在一个通道，该值不需要配置。 |
| 13 | | |
| 14 | | |
| 15 | 初始化存储器操作 | 固定为 0。初始化存储器操作将重置存储器。 注意，所有存储卡上的数据会被清除，且不可恢复！ |
| 44 | 单文件存储时间 | 单位为 s，表示保存在存储卡上的单个 |

| | | |
|-----------|------------|---|
| | | 文件存储的数据时间。 |
| 45 | 周期采样开始时间 | 采样模式为周期采样时使用，从“开始时间”开始，每“采样周期”时间内采样“周期内采样时长”，直到“结束时间”时停止。 “周期内采样时长”要小于“采样周期”至少 10s，保证设备有足够的时间结束采样。 |
| 46 | 周期采样结束时间 | |
| 47 | 采样周期 | |
| 48 | 周期内采样时长 | |
| 49+2(N-1) | 分段 N 的开始时间 | 采样模式为分段采样时使用，N 为 1~10（即最大支持配置 10 个分段采样）。每一个分段采样的结束时间必须大于开启时间，且下一分段的开始时间必须大于上一分段的结束时间 10s 以上，保证设备有足够的时间结束采样。 所有分段时间设置必须按时间先后顺序排列。 |
| 50+2(N-1) | 分段 N 的结束时间 | |

系统命令表

| 值 | 说明 |
|---|--|
| 0 | 停止采样（无论出于何种模式的采样过程） |
| 1 | 开始采样（设备处于手动采样模式下有效） |
| 2 | 重启设备 |
| 5 | 确认设备关机，如果需要关机，需要先发送“设备关机允许”命令，等收到回复后发送“确认设备关机”。UNAS 设备不支持关机操作。 |
| 6 | 设备关机允许。UNAS 设备不支持关机操作。 |

如果设备设置系统时间成功，则会立刻回复配置成功应答帧，具体数据内容如下：

配置成功应答帧字段说明（帧类型 0x81）

| 序号 | 字段名称 | 字节数 | 说明 |
|----|------------|-----|------------------------------|
| 1 | 保留 | 12 | 保留字段 |
| 2 | 设备 ID | 4 | 当前设备的 ID 信息，4 个 ASCII 码表示 |
| 3 | 单文件持续时间 | 4 | 单位为 s，表示保存在存储卡上的单个文件存储的数据时间。 |
| 4 | 存储器总容量 | 4 | 单位为 MB，表示设备配置的存储器总容量 |
| 5 | 存储器剩余容量 | 4 | 单位为 MB，表示设备当前剩余存储器总容量 |
| 6 | 保留 | 4 | 保留字段 |
| 7 | 当前采样率 | 4 | 参考”配置参数表”中采样率说明 |
| 8 | 当前增益 | 4 | 参考”配置参数表”中增益说明 |
| 9 | 采样通道数 | 4 | 当前设备采样通道数 |
| 10 | 采样位宽 | 4 | 当前设备支持的采样位宽 |
| 11 | 保留 | 4 | 保留字段 |
| 12 | 当前采样模式 | 4 | 参考”配置参数表”中采样模式说明 |
| 13 | 周期采样参数 | 16 | 周期采样开始时间，4 字节整型 |
| | | | 周期采样结束时间，4 字节整型 |
| | | | 采样周期，4 字节整型 |
| | | | 周期内采样时长，4 字节整型 |
| 14 | 分段采样参数 | 80 | 分段 N 的开始时间，N 为 1~10 |
| | | | 分段 N 的结束时间 |
| 15 | 保留 | 40 | |
| 16 | 设备当前 IP 地址 | 4 | 参考”配置参数表”中网络地址说明 |
| 17 | 设备当前网关地址 | 4 | |
| 18 | 设备当前子网掩码 | 4 | |
| 19 | 保留 | 40 | |
| 20 | 预览通道位掩码 | 12 | 参考”配置参数表”中预览通道位掩码 |

| | | | |
|----|----|---|------|
| | | | 说明 |
| 21 | 保留 | N | 保留字段 |

设备端接收到系统配置帧后，会在指定超时时间内回复，如果上位机在超时时间内未接收到回复帧，则需要重发，如果达到重发次数后设备仍未回复，则认为数据链路断开或者设备故障，上位机需要进入重新连接流程。

所有的配置操作，如果处理正确，均使用上述消息将所有设备信息返回。

有任何一个配置操作失败，均回复错误应答。如果一条消息有多个配置项，则下述错误响应中不包含的配置操作项表示成功。

系统配置失败回复帧字段说明（帧类型 0xC1）

| 序号 | 字段名称 | | 字节数 | 说明 |
|-----|----------|-------|-----|-------------|
| 1 | 失败配置参数个数 | | 1 | |
| 2 | 保留 | | 3 | |
| 3 | 失败配置参数 1 | 类型 | 2 | 指示配置参数的类型 |
| | | 失败原因 | 2 | 详见“配置失败原因表” |
| | | 设备当前值 | 4 | 配置参数帧 |
| 4 | 失败配置参数 2 | ... | | |
| ... | | | | |

配置失败原因表

| 值 | 说明 |
|---|----------------------|
| 1 | 操作不支持，表示没有对应的命令 |
| 2 | 参数不支持，表示设备不支持该配置参数的值 |
| 3 | 操作失败，表示设备未能正确完成配置操作 |
| 4 | 操作不被允许，表明设备正在忙 |

2.2 高速数据通道数据帧

设备端采集的实时预览数据经由高速数据通道主动推送到上位机，上位机收到数据之后不需要确认。TCP 连接数据端口后，设备就持续输出，直到采样停止。

实时预览数据

实时预览数据为设备采集到的原始数据，数据格式由设备硬件决定。现阶段固定为 24 位大端格式。

如果使能了实时预览数据上传，设备端在采集完一定量数据后，会立刻打包推送。

实时预览数据帧字段说明（帧类型 0x82）

| 序号 | 字段名称 | 字节数 | 说明 |
|----|--------|-----|--|
| 1 | 保留 | 1 | 保留 |
| 2 | 数据格式指示 | 1 | Bit0~2: 采样位宽，单位为字节，固定为 3； Bit3: 数据域 0 为小端，数据域 1 为大端，固定为 1； Bit4~7: 保留。 |
| 3 | 保留 | 2 | 保留 |
| 4 | 数据域长度 | 2 | 预览数据的字节长度 |
| 5 | 预览状态指示 | 1 | Bit0: 丢包指示，如果为 1 表示：如果一段时间内存在数据产生速率大于数据链路所能提供的最大传输速度，就会存在原始数据被覆盖的情况。上位机根据这个信息可以知晓因数据链路传输速率不够而导致预览数据不完整。 Bit1~7: 保留。 |
| 6 | 保留 | 2 | 保留 |
| 7 | 样点数偏移 | 8 | 8 字节无符号数，为当前数据包从 0 开始（每次采样开始复位）的按照样点数计算的偏移。 如果是多通道采集，设备打包时保证所有数据 |

| | | | |
|---|---------|----|-------------------------------------|
| | | | 通道的同样偏移位置的样点数据包含在同一个数据帧中； |
| 8 | 预览通道位掩码 | 12 | 参考”配置参数表”中预览通道位掩码说明 |
| 9 | 预览数据 | 变长 | 预览数据为大端补码格式，每个数据点三个字节（24 位），按通道依次保存 |

示例

fe fe 0c 04 01 00 44 00 00 82 20 90 02 0b 44 05 e4 03 00 00 60 bd 0a 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 0a bd 60 0a bd 61 0a bd 62 0a bd 63 0a bd 64 0a bd 65 0a bd 66 0a bd 67 0a bd 68 0a bd 69 0a bd 6a 0a bd 6b 0a bd 6c 0a bd 6d 0a bd 6e 0a bd 6f 0a bd 70 0a bd 71 0a bd 72 0a bd 73 0a bd 74 0a bd 75 0a bd 76 0a bd 77 ...

fe fe 为开始字节

0c 04 表示这是一个包含 1036 字节的数据帧

01 00 表示本数据帧的协议版本为 0x0001

44 表示本数据帧的数据号为 0

00 00 地址信息，暂时不用

82 表示本数据帧为实时预览数据

20 90 为本数据帧的 CRC 检验值

02 保留，不用关心

0b 00001011b 采样位宽为 3 字节, 数据域为大端, 实时数据

44 05 保留，不用关心

03 e4 为预览数据字节长度, 996 个字节

00 00 保留，不用关心

60 bd 0a 00 00 00 00 00 样点数偏移，表示本数据包第一个样点偏移为

703840

01 00 00 00 00 00 00 00 00 00 00 00 00 00 预览通道位掩码，表示当前只有通道 1 处于预览状态中

剩余字节表示预览数据内容，示例中为自增数据

3、CRC 参考实现

以下是 CRC 校验 C 语言参考代码：

```
#define MSG_SCRAMBLER_NUMBER 0x5A5C
int32_t Msg_CheckCrc(uint8_t *pMsg, uint32_t len)
{
    uint16_t *pData = (uint16_t *)pMsg;
    uint32_t num = len/(sizeof(uint16_t));
    uint32_t i;
    uint16_t crc = 0;

    for(i = 0; i < num; i++)
    {
        crc ^= pData[i];
    }

    if(crc == MSG_SCRAMBLER_NUMBER)
    {
        return 0;
    }
    else
    {
        return -1;
    }
}

uint16_t Msg_CalcCrc(uint8_t *pMsg, uint32_t len)
{
    uint16_t *pData = (uint16_t *)pMsg;
    uint32_t num = len/(sizeof(uint16_t));
    uint32_t i;
```

```
uint16_t crc = 0;

for(i = 0; i < num; i++)
{
    crc ^= pData[i];
}

return crc^MSG_SCRAMBLER_NUMBER;
}
```