**主机网络协议**

**v1.4**

**注意：协议中非单字节内容填充都遵循memcpy原则**

* **报文定义：**
  + 报文一般格式：

|  |  |  |
| --- | --- | --- |
| **7E7E7E** | 3B | 报头 |
| **LEN** | 2B | 整个报文长度，short |
| **TYPE** | 1B | 报文/应答类型，ASCII字符 |
| **DATA** |  | 报文数据 |
| **CHKSUM** | 1B | 校验和（从LEN到DATA, 8位无符号相加溢出不管） |

* + 命令报文/应答：

上位机向采集器发送的命令，采集器都有相应的应答报文回送；

一般的命令应答报文定义如下：

**TYPE：同命令类型**

DATA：

|  |  |  |
| --- | --- | --- |
| **A** | 1B | 0：错误；1：成功；2：校验错 |

* **命令列表：**

|  |  |  |  |
| --- | --- | --- | --- |
| **r** | 读取量程 | 0x72 |  |
| **R** | 设置量程 | 0x52 |  |
| **t** | 同步时间 | 0x74 |  |
| **d** | 读取主机实时数据 | 0x64 |  |
| **D** | 读取从机实时数据 | 0x44 |  |
| **i** | 设置4~20mA输出大小 | 0x69 |  |
| **j** | 下载4~20mA输出的标定a和k系数 | 0x6A |  |
| a | 下载4~20mA输入的a系数 | 0x61 |  |
| K | 下载4~20mA输入的k系数 | 0x6B |  |
| **e** | 设置继电器动作命令 | 0x65 |  |
| **h** | 设置远端机是否在线 | 0x68 |  |
| **w** | 设置有线传输还是无线433 | 0x77 |  |
| **p** | 设置报警上下限阈值 | 0x70 |  |
| **g** | 发送4G数据 | 0x67 |  |
| **y** | 需要发送到远端机的数据 | 0x79 |  |

* **网络连接：**
  + 读取量程：

**TYPE：r**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |

发送r命令示例数据：

0x7E 0X7E 0X7E 0X09 0X00 0X72 0X01 0X00 CRC； CRC需计算，见第一个表格

其中0x7E 0X7E 0X7E 为开头固定、0x09 0x00 是长度，先发低字节后发字节； 0x72是‘r’字母的ascii码值； 0x01 是total；0x00 是index；最后加个CRC 一个字节，从LEN到DATA按**字节**相加, 8位溢出不管 unsigned char 格式累加 即可

应答：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Range | 64B | save\_modbus.p.range\_low[0]开始的64字节，即float range\_low[8]; float range\_top[8]; |  |

返回r数据示例：

0x7E 0X7E 0X7E 0X47 0X00 0X72 64字节数据 CRC；

0x47，0x00为长度 64字节+7=0x47； 0x72 命令不变返回，64字节数据为Range数据，存储方式都是低字节在前，高字节在后。符合常规电脑和ARM的存储模式

* + 设置量程：

**TYPE：R**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| Ch | 1B | 当前选择往下发送的通道好，0~7 |  |
| Is\_top | 1B | 量程低端还是高端，0：float range\_low[8]；1：float range\_top[8] |  |
| Range | 4B | Float，新的量程，每次只能设一个 |  |

0x7E 0X7E 0X7E 0X0F 0X00 0X52 0X01 0X00 0x00 0x01 float(50) CRC；

设置第0个通道的量程高端为50， 50是float型，小端模式放入4个字节内

应答：标准应答

0x7E 0X7E 0X7E 0X08 0X00 0X52 0X01 CRC；标准返回 0x01为返回状态，标准返回雷同，以下列出。

* + 同步时间：

**TYPE：t**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| Year | 1B | 年，例如18代表 2018年（当前年-2000） |  |
| Month | 1B | 月 |  |
| Day | 1B | 日 |  |
| Hour | 1B | 时 |  |
| Minute | 1B | 分 |  |
| Sec | 1B | 秒 |  |

0x7E 0X7E 0X7E 0X0F 0X00 0X74 0X01 0X00 0x12 0x06 0x05 0x08 0x0F 0x07 CRC；

修改设置时间为2018年6月5日 08点15分07秒

应答：标准应答

* + 读取主机实时数据命令：

**TYPE：d**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |

0x7E 0X7E 0X7E 0X09 0X00 0X64 0X01 0X00 CRC；

应答：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| slave\_data | sizeof(SLAVE\_DATA) | 返回给电脑 slave\_data 结构体内的数据 |  |

* + 读取保存在主机上的从机实时数据命令：（主机不停的从远端机获取的数据）

**TYPE：D**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |

0x7E 0X7E 0X7E 0X09 0X00 0X44 0X01 0X00 CRC；

应答：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| remote\_device | sizeof(REMOTE\_DEVICE)\*3 | 返回给电脑 remote\_device 结构体内的数据，目前是3个从机，所以数据量乘以3 |  |

* + 设置4~20mA输出大小（2个通道）：

**TYPE：i**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| Flag|ch | 1B | Flag：  0x80：标定状态，发下去多少就是多少，  0x00：运行状态，发下去的电流值经过标定系数修正后输出  Ch：  0x00：第0个通道  0x01：第1个通道  这个字节的参数是需要或操作后发送的 | 0x00  0x01  0x80  0x81 |
| iout | 4B | Float，新的电流输出值，每次只能设一个 |  |

0x7E 0X7E 0X7E 0X0E 0X00 0X69 0X01 0X00 0x80 float(iout) CRC；

标定状态第0个通道的4~20mA输出，标定时用。

应答：标准应答

* + 标定4~20mA输出（2个通道）：

**TYPE：j**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| ch | 1B | 0x00：第一个通道  0x01：第二个通道 |  |
| a | 4B | Float，标定系数a 模式：y=a+k\*x |  |
| k | 4B | Float、标定系数k |  |

a = (small\_out \* 5 - big\_out) / 4;

k = (big\_out - small\_out) / 16;

其中，small\_out为设置4mA输出时用电流表测得的值（标定模式）；big\_out为设置20mA时测得值

0x7E 0X7E 0X7E 0X12 0X00 0X6A 0X01 0X00 0x00 float(a) float(k) CRC；设置第0个通道的a和k系数，4~20mA输出标定的a和k

应答：标准应答

* + 标定4~20mA输入（4个通道）：

**TYPE：a/k**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| ch | 1B | 0~3 共4个通道 |  |
| a/k | 4B | Float，标定系数a 或 k 模式：y=a+k\*x; a,k需分开发 |  |

a = (float)4.0 - (float)16.0 \* i\_z / (i\_f - i\_z);

k = (float)16.0 / (i\_f - i\_z);

其中，i\_z为接入4mA标准信号时仪表测得的电流值，可通过d命令读取(slave\_data.I[8],该值保存的是未修正的电流值)；i\_f为接入20mA是仪表测得的电流值

0x7E 0X7E 0X7E 0X0E 0X00 0X61 0X01 0X00 *0x00* float(a) CRC；设置第0个通道的a系数

0x7E 0X7E 0X7E 0X0E 0X00 0X6B 0X01 0X00 *0x00* float(k) CRC；设置第0个通道的k系数

应答：标准应答

* + 设置报警阈值3参数命令（4个通道）：

**TYPE：p**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| ch | 1B | 0~3 共4个通道 |  |
| AlarmUP | 4B | 报警上限 float |  |
| AlarmDown | 4B | 报警下限 float |  |
| TargetPos | 4B | 目标位置 float |  |

0x7E 0X7E 0X7E 0X0B 0X00 0X70 0X01 0X00 0X00 (float)alarmup (float)alarmdown (float)targetpos CRC；

设置第0个通道的3个参数。

应答：标准应答

* + 设置4G参数和发送4G数据：

**TYPE：g**

发送：

DATA：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 字节数 | 解释 | 取值 |
| Total | | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | | 1B | 当前发送的是第几个包，从0开始 | 0 |
| Flag | | 1B | 0：设置4G模块AT参数，包括云IP、端口、网址、用户名、密码  1：发送数据到云端 | 该位对下方数据定义有影响 |
| Send\_Datas  (长度就是实际发送的字符串长度) | Send\_byARM | 1B | 1：由ARM来发送4G数据到云端  0：ARM不执行发送云操作 | Flag=0时，则具有左侧意义；  Flag=1时，这区域就是实际发送云端的数据 |
| Web\_ip | 32B | 云端IP和端口，例子：118.31.40.229,80 逗号隔开 |
| Web\_url | 64B | 云端网址，例子：/dongtu/batchData |
| Web\_user | 16B | 用户名，默认空 |
| Web\_passwd | 16B | 密码，默认空 |

当Flag=1时，发送如下类似数据：

code=0025&section=01&pile=a1&batchData=2018/05/30/23:11:11,c1,20,30,4,50,60,80;2018/05/30/23:12:11,c1,30,40,6,50,160,170;2018/05/30/23:13:11,c1,20,30,7,50,260,270

这是初期测试用的数据上传，用户根据需要上传

应答：标准应答

* + 设置继电器动作命令（8个通道）：

**TYPE：e**

发送：

DATA：（type1）

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| ch | 1B | 0~7 共8个通道 |  |
| Status | 1B | 0：开，1：合 设置继电器的状态 |  |

0x7E 0X7E 0X7E 0X0B 0X00 0X65 0X01 0X00 0X00 0X01 CRC；

设置第0个通道的继电器状态为高电平输出状态（+12VDC）

**以上例子是单个通道的设置继电器方法，下面列举多个继电器同时动作的方法：**

DATA：（type2）

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| ch | 1B | 固定为0xFF |  |
| Status | 1B | Bit0：继电器1，bit7：继电器8  0：开，1：合 设置继电器的状态 |  |

0x7E 0X7E 0X7E 0X0B 0X00 0X65 0X01 0X00 0XFF 0X0F CRC；

设置继电器1到继电器4为闭合状态，继电器5-继电器8位打开状态

应答：标准应答

* + 设置远端机在线命令：

**TYPE：h**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| device | 3B | 3个远端机的状态，0：不在线 ，1：在线。 |  |

0x7E 0X7E 0X7E 0X0C 0X00 0X68 0X01 0X00 0x01 0x01 0x01 CRC；

设置3台远端机都在线

应答：标准应答

* + 设置主机采用哪种模式传输数据，和远端机之间：

**TYPE：w**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| Has\_line | 1B | 0：433M无线 ，1：有线。 |  |

0x7E 0X7E 0X7E 0X0A 0X00 0X77 0X01 0X00 0x01 CRC； 设置主机为485有线传输

应答：标准应答

* + 上位机和远端机之间通讯：

**TYPE：y**

发送：

DATA：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 字节数 | 解释 | 取值 |
| Total | 1B | 总共需要发送几次网络包，从1开始。网络通讯一包有字数上限需拆分发送，一包推荐不超过1200字节 | 1 |
| Index | 1B | 当前发送的是第几个包，从0开始 | 0 |
| Fun | 1B | 子功能类型 |  |
| inData | \*B | 数据区，长度根据子功能类型确定 |  |

子功能：模块查询0x01

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x01：查询远端机是否在线 |
| inData | 0B | 无 |

0x7E 0X7E 0X7E 0X0A 0X00 0X79 0X01 0X00 0x01 CRC；发送完后，需要扫描3个远端机，预估3秒以上等到超时才能返回应答

返回应答：

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x01 |
| Mode1 | 1B | 远端机类型，固定0x01;如果是0，代表第1个地址的设备不在线 |
| Mode2 | 1B | 远端机类型，固定0x01;如果是0，代表第2个地址的设备不在线 |
| Mode3 | 1B | 远端机类型，固定0x01;如果是0，代表第3个地址的设备不在线 |

0x7E 0X7E 0X7E 0X0B 0X00 0X79 0x01 0x01 0x00 0x01 CRC；返回3个在线状态列表：1:0:1

**注意，返回包中都不含有Total，index项，以下同样；**

子功能：继电器控制0x02

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x02：设置远端继电器状态 |
| Addr | 1B | 目标远端机地址（2开始-4结束，目前3个远端机） |
| Relay | 1B | 0：断开，1：合起（+12V输出） |

0x7E 0X7E 0X7E 0X0C 0X00 0X79 0X01 0X00 0x02 0x02 0x01 CRC；设置0x02地址的远端机继电器为合起状态

返回应答：

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x02 |
| Addr | 1B | 远端机地址 |
| Status | 1B | 1：成功，0：失败 |

0x7E 0X7E 0X7E 0X0A 0X00 0X79 0x02 0x02 0x01 CRC；返回数据包成功例子

子功能：屏幕控制0x03

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x03：设置远端机屏幕 |
| Addr | 1B | 目标远端机地址（2开始-4结束，目前3个远端机） |
| On | 1B | 0：关闭屏幕，1：亮起屏幕 |

返回应答：

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x03 |
| Addr | 1B | 远端机地址 |
| Status | 1B | 1：成功，0：失败 |

子功能：远端机油压设置0x04

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x04：设置远端机油压数据 |
| Addr | 1B | 目标远端机地址（2开始-4结束，目前3个远端机） |
| Oil | 4B | 上位机动态设置给远端机的油压数据（float） |

返回应答：

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x04 |
| Addr | 1B | 远端机地址 |
| Status | 1B | 1：成功，0：失败 |

子功能：远端机8个通道显示设置0x05

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x05：设置远端机8个测量通道数据（相对位移） |
| Addr | 1B | 目标远端机地址（2开始-4结束，目前3个远端机） |
| Ch | 32B | 远端机8个通道的数据（float\*8） |

返回应答：

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x05 |
| Addr | 1B | 远端机地址 |
| Status | 1B | 1：成功，0：失败 |

子功能：远端机8个通道pc软件二次标定

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x06：设置远端机8个测量通道标定系数 |
| Addr | 1B | 目标远端机地址（2开始-4结束，目前3个远端机） |
| Ch | 1B | 远端机8个通道的通道号，0~7 |
| a | 4B | 标定a系数 y=a+kx; float |
| k | 4B | 标定k系数 |

返回应答：

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x06 |
| Addr | 1B | 远端机地址 |
| Status | 1B | 1：成功，0：失败 |

子功能：远端机第8个通道配套继电器用作报警上下限控制的报警值设置

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x07：设置远端机8个测量通道标定系数 |
| Addr | 1B | 目标远端机地址（2开始-4结束，目前3个远端机） |
| Ch | 1B | 哪个通道控制继电器，0~7中选一个，默认7，即最后一个 |
| AlarmUP | 4B | 报警上限 float |
| AlarmDown | 4B | 报警下限 float |
| TargetPos | 4B | 目标位置 float |

返回应答：

|  |  |  |
| --- | --- | --- |
| Fun | 1B | 0x07 |
| Addr | 1B | 远端机地址 |
| Status | 1B | 1：成功，0：失败 |

**以上协议是目前程序具备的响应，用户可自定义添加删除或修改协议，ARM端程序协议修改添加见 lwip->tcp\_rec\_send.c 文件的tcp\_usrserver\_recv（\*\*\*）函数内；**

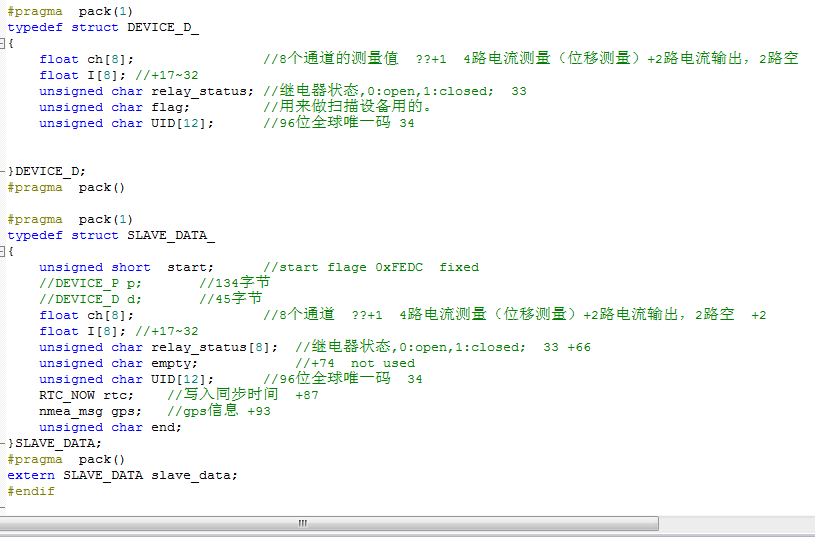
**数据交互的几个结构体，usr\_define.h和rs485.h文件下：**

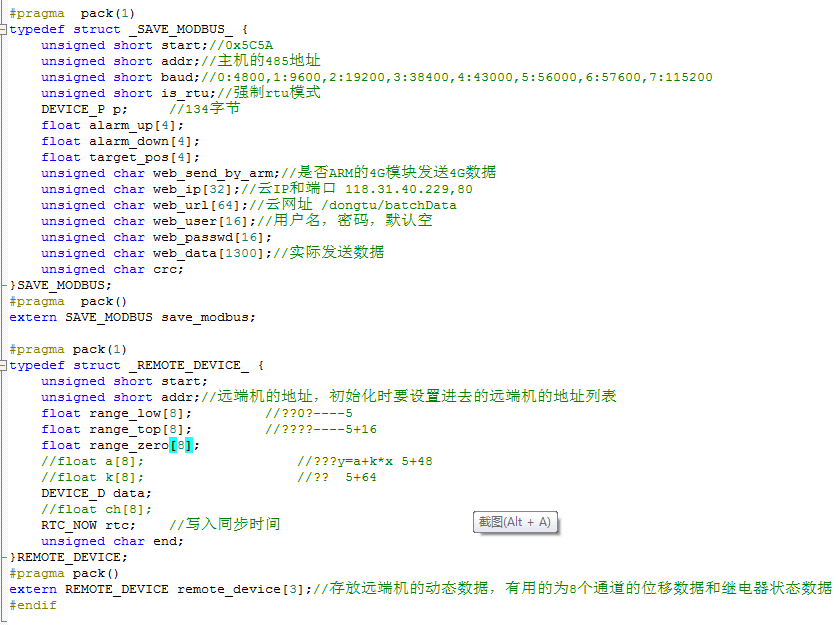
**REMOTE\_DEVICE： 存放远端机实时数据的**

**SLAVE\_DATA：主机设备实时数据存放点**

**SAVE\_MODBUS：主机参数存放点**

****

****

****