成都业贤科技有限公司

数控 RS232 通讯协议

V3.0

He Fahong

2012-8-30

该协议写明了通讯基本格式，具体的通讯命令字符串可以在我们提供的软件 EasyHost 里查询。查询方法请阅读本协议的 3.3 小节。

本文档是成都业贤科技有限公司产品的 RS232 通讯协议。

1 解释

1.1 下位机：具备某个特定功能，硬件上相对独立，具有数字控制单元的设备。比如，数字温度控制器，数控恒流源驱动器等。

1.2 上位机：通过串口与下位机通讯，通过软件控制下位机的设备，通常指计算机，也可能是单片机。

1.3 下位机的内部逻辑结构：每个下位机有多个独立的子模块；每个子模块有不同的属性集合。为了便于管理，每个属性都定义成 1 个参数。因此每个子模块的全部特性可以用 1 个参数集合来完全表征。通过子模块名加上参数名，就可以迅速定位到某个子模块的某个属性。

2 通讯语法

通讯命令有设定命令、查询命令、保存命令。

2.1 设定命令。

设定命令用于设定某个参数的值。

2.1.1 设定命令语法如下：

MODULENAME:PARAMNAME=PARAMVALUE\r

2.1.2 整个设定命令由四部分组成：模块参数名、等号、设置值、结束符。

2.1.3 含义为：模块:参数=值\r

例如：

TC1:TCADJUSTTEMP=25\r

含义是把子模块 TC1 的参数 TCADJUSTTEMP（该参数的含义是温度控制器的调节温度）设置为 25。

2.1.4 并非每个参数都可以被设置。某些参数只能被查询。某些参数不对外公开。

2.1.5 下位机接收到设定命令后，会执行相关操作，然后返回执行结果信息。返回信息的格式如下：

CMD:REPLY=ERRORCODE\r

其中 ERRORCODE 代表错误信息码。

2.2 查询命令。

查询命令用于查询某个参数的值。

2.2.1 语法如下：

MODULENAME:PARAMNAME?\r

2.2.2 整个查询命令由三部分组成：模块参数名、问号、结束符。

2.2.3 含义为：模块:参数? \r

例如：

TC1:TCADJUSTTEMP?\r

含义是查询子模块 TC1 的参数 TCADJUSTTEMP 的值。

2.2.4 绝大部分参数都可以被查询。

2.2.5 下位机接收到查询命令后，会执行相关操作，然后返回执行结果信息。如果查询命令正确，返回信息的格式如下：

MODULENAME:PARAMNAME=PARAMVALUE\r

比如：

向下位机发送 TC1:TCADJUSTTEMP?\r

下位机则返回 TC1:TCADJUSTTEMP=25\r

如果查询命令错误，返回信息的格式如下：

CMD:REPLY=ERRORCODE\r

其中 ERRORCODE 代表错误信息码。

2.3 保存命令。

保存命令用于保存某个参数的设定值。

2.3.1 语法如下：

MODULENAME:PARAMNAME!\r

2.3.2 整个保存命令由三部分组成：模块参数名、叹号、结束符。

2.3.3 含义为：模块:参数! \r

例如：TC1:TCADJUSTTEMP!\r

含义是把子模块 TC1 的参数 TCADJUSTTEMP 的当前设定值保存，下次开机时读取保存值。

2.3.4 只有可以设定的值才能保存，但并非所有可设定的值都可以保存。

2.3.5 下位机接收到保存命令后，会执行相关操作，然后返回执行结果信息。返回信息的格式如下：

CMD:REPLY=ERRORCODE\r

其中 ERRORCODE 代表错误信息码。

3 语法说明

3.1 “MODULENAME”：子模块名称。该名称的具体定义由下位机确定。

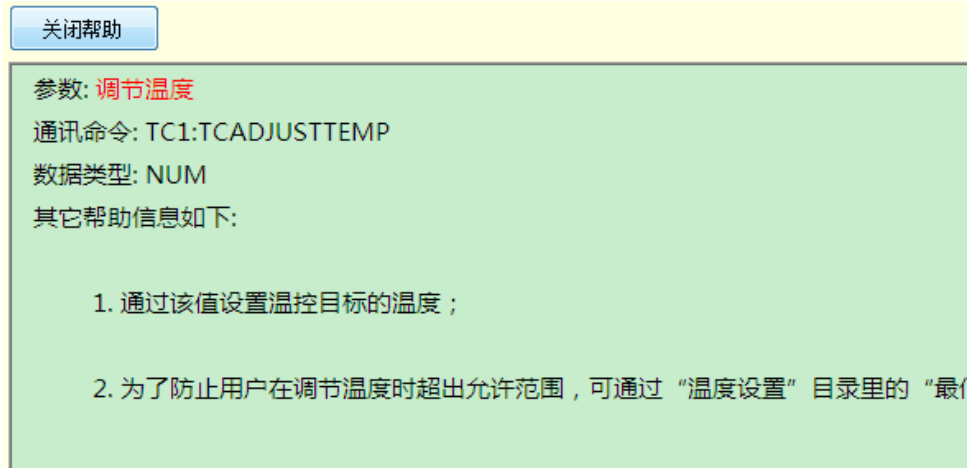
子模块名称举例：温度控制器中的温度控制子模块 1，TC1；温度控制

器中的温度控制子模块 2，TC2；恒流驱动器中的子模块 LD1，LD2；下位机中的电路板温度监视子模块，PCB；存储器子模块，MEMORY。

3.2 “PARAMNAME”：参数名称。该名称的具体定义由下位机确定。用冒号分隔开子模块名称和参数名称。

3.3 在软件 EasyHost 最新版（V6 及以上）里，点击每行参数后面的“帮助”按钮。

在弹出的帮助界面里，可以查询到该参数对应的子模块名称和参数名称。



3.4 “=”：用“=”标明是设定命令。

3.5 “?”：用“?”标明是查询命令。

3.6 “!”：用“!”标明是保存命令。

3.7 “\r”：用于表明 1 个命令结束的回车符，是 1 个字节的单字符。ASCII 码为 0x0D。关于“\r”和转义字符的更多信息，请在网络上搜索。

3.8 ERRORCODE：错误信息码。用于标明下位机在执行命令的过程中产生的错误信息，可以帮助用户了解其所使用的通讯命令的执行情况。

4 通讯规则

4.1 所有的通讯命令都是大小写敏感的。通常情况下，我司的下位机的通

讯命令都是大写。

4.2 每个通讯命令，无论是计算机发向下位机的，还是下位机返回计算机的，其结尾都含有符号“\r”来标明该命令完整结束（即每个命令字符串的最后一个 ASCII 码必然是 0x0D）。无该结束符号的为非法命令，不被识别。

4.3 所有命令均以 ASCII 码方式发送。通讯协议本身都是以字符串方式定义的，字符串在计算机中存储和传输时一般使用的是 ASCII 码（ASCII 码的更多信息可以网络搜索）。比如用于查询温控器调节温度的命令 **TC1:TCADJUSTTEMP?\r**，在 C 语言程序中以字符串格式表达为"TC1:TCADJUSTTEMP?\r"，程序执行时会通过串口把字符串里每个字符的 ASCII 码依次发送出去：0x54, 0x43, 0x31, 0x3A, 0x54, 0x43, 0x41, 0x44, 0x4A, 0x55, 0x53, 0x54, 0x54, 0x45, 0x4D, 0x50, 0x3F, 0x0D。0xHH 表示 16 进制格式下，ASCII 码值为 HH。字母“T”的 ASCII 码是 0x54，字母“C”的 ASCII 码是 0x43，……，字母“?”的 ASCII 码是 0x3F，结束回车符（即“\r”）的 ASCII 码为 0x0D。

4.4 设置命令中的数字或者下位机返回的数字同样以 ASCII 码方式发送。比如设置温控器调节温度的命令 **TC1:TCADJUSTTEMP=25.01\r**，目的是要把温度设置在 25.01 摄氏度，在 C 语言程序中以字符串格式表达为"TC1:TCADJUSTTEMP=25.01\r"，串口依次发送的字节数据如下：0x54, 0x43, 0x31, 0x3A, 0x54, 0x43, 0x41, 0x44, 0x4A, 0x55, 0x53, 0x54, 0x54, 0x45, 0x4D, 0x50, 0x3D, 0x32, 0x35, 0x2E, 0x30, 0x31, x0D。前面这句描述里用 16 进制下的 8 位格式表示数据，即每个字节单独表示，共 23 个字节。最后一段 ASCII 码中（有下划线这一段），字母“=”的 ASCII 码是 0x3D，数字“2”的 ASCII 码是 0x32，数字“5”的 ASCII 码是 0x35，小数点“.”的 ASCII 码是 0x2E，数字“0”的 ASCII 码是 0x30，数字“1”的 ASCII 码是 0x31，结束回车符“\r”的 ASCII 码为 0x0D。

4.5 一条通讯命令的内部、前面不能有空格。

4.6 每两条发向下位机的通讯命令的发送时间间隔必须大于 50ms，小于这

个时间间隔的命令可能不被下位机受理。为了避免发送的命令未被处理导致错误，可以在发送设置命令后，等 50ms 再发送一次查询命令检查设置是否完成。

4.7 地址码：紧接着命令添加“@X”，其中 X 为欲通讯的下位机串口地址。单个下位机时，可以不用该功能。

4.8 校验码：紧接着地址码后添加“#YY”，YY 为异或校验和。

4.9 一般情况下，可以不使用地址码和校验码。下位机自动调整回复命令的格式和上位机发送命令的格式一样。

5 错误信息码：ERRORCODE。

5.1 CMD:REPLY=0， 设定/查询/保存命令未找到子模块名称，或参数名称。

5.2 CMD:REPLY=1，设定命令正确执行返回。

5.3 CMD:REPLY=2，设定/查询/保存命令未找到参数名称。

5.4 CMD:REPLY=3，设定/查询/保存命令被禁止。

5.5 CMD:REPLY=4，设定命令参数值超范围。

5.6 CMD:REPLY=5，其它或未知错误。

5.7 CMD:REPLY=6，命令格式语法错误。

5.8 CMD:REPLY=7，通讯命令里有校验错误。

5.9 CMD:REPLY=8，保存正确执行。

6 软件

6.1 通常情况下，用户键盘输入的是字符，软件的文本框也显示的是字符。但该串口通讯协议要求每个命令后都要有特定字符“\r”结尾，给处理带来了一定难度，如果用户输入“\r”，通常会被软件识别为两个字符“\”和“r”。

6.2 我司提供的软件 EasyCom 在发送和接收时会自动添加/删除该结束符（无需用户输入），方便用户在电脑上调试下位机。用户可使用该软件来熟悉我司的串口通讯协议。

6.3 如果使用网络上的其它串口工具软件，则需要用户手动输入结束符；因此很多软件无法用于我司模块的通讯调试。

6.4 部分第三方的串口工具支持转义方式，可以用于我司串口的调试，比

如 eagleCom。如下图所示，选中“\hh HEX 转换”，则可以在发送命令后用\0D 来表示结束字符“\r”，软件在读取到字符“\”时自动把后续的两个字符识别成 1 个字符的 ASCII 码。

7 串口地址

7.1 如果只需要操作 1 个下位机，则用户无需关心该部分内容。

7.2 每个串口具有 1 个地址（0～254 之间）。串口地址标明该串口的地址信息。255 为广播地址。

7.3 上位机发送过来的命令地址如果和串口地址两者相同，则该串口会响应上位机的命令；如果不同，则表示自己不是上位机想通讯的目标，不会响应上位机的命令。串口地址默认为 0。

7.4 常规情况下，如果上位机发送的命令没有地址后缀，则默认该命令是和上一次通讯的下位机继续通讯。

7.5 带地址后缀的命令格式为：

MODULENAME:PARAMNAME=PARAMVALUE@X\r

下位机返回命令也会在末尾加上“@X”，其中 X 为通讯的下位机的地址。

7.6 该地址功能可用于多下位机的并联，简介如下：

7.6.1 下位机地址设定。通过软件 EasyUI 或者 UIM 模块设置下位机地址。

7.6.2 连接方法如下：

7.6.2.1 上位的 TXD 直接连接到多台下位机的 RXD；

7.6.2.2 上位机的地和下位机的地相连；

7.6.2.3 上位机的 RXD 接 10k 电阻连接到地；

7.6.2.4 多台下位机的 TXD 经过或门后，再和上位机的 RXD 相连；

7.6.2.5 可选用我司的多下位机连接配件：串口复用模块。

7.6.3 控制方法如下：发送带地址后缀的命令和下位机通讯。

7.7 关于下位机并联的进一步信息，请参见具体产品的用户手册，或者参考我司应用笔记 7：1 个串口控制多个独立的 TCM 系列温控器。

8 校验码

8.1 通常情况下，不使用校验码对通讯没有影响。在某些高可靠性应用中，需要保证每次串口发送命令和接收命令都必须完全正确，此时需要使用校验。使用校验码时，必须使用地址功能，如果是默认的下位机，则可以使用默认地址 0。

8.2 比如“TC1:TCSW=1@0#50\r”命令含义为：打开地址为 0 的温控器的 TC1 的输出开关，异或校验码为 50。下位机会对接收到的字符串和校验码进行比对校验，如果校验码正确，则执行命令。如果命令正确执行，下位机回复“CMD:REPLY=1@0#7D\r”。

8.3 校验码计算从命令第一个字符开始，到“#”结束。校验码初值为 0，把它和所有字符都依次进行异或计算，得到 1 个字节的校验码，把它的 16 进制表达转换成两个字符添加到命令末尾。校验码字符必须大写。

8.4 比如“TC1:TCSW=1@0#”的校验结果为 16 进制的 0x50，则把数字 50 转换成两个字符“5”和“0”添加在命令末尾：“TC1:TCSW=1@0#50\r”。

9 应用

9.1 用户设置下位机时，不建议每次都通过发送命令的方式设置所有参数；建议先使用我司的上位机软件先设置/保存好使用过程中不会改变的通用参数，然后在用户自己的上位机/单片机程序里发送命令设置经常变化的参数。

10 Modbus 支持

10.1 部分新产品支持 Modbus RTU。需要在 EasyUI/UIM, EasyHost 里通过单独的选项设置打开 Modbus 支持功能；

10.2 支持选项共 3 个等级：0，完全不支持 Modbus RTU；1，系统优先以自定义协议检测上位机发送的命令，没有结束符的命令再送去 Modbus 监测；2，系统优先进行 Modbus RTU 的 CRC 监测，不符合的情况下才进行自定义协议监测。

10.3 目前仅支持三个 Modbus RTU 命令：0x03（读保持寄存器），0x04（读只读或输入寄存器），0x10（写保持寄存器）。注意，0x10 命令只是修改参数值，修改值并不会保存，即下次上电后消失；我司自定义协议里的保存功能还未在 Modbus 支持里实现。

10.4 产品对 Modbus RTU 的支持并不涉及全部参数，可支持的参数列表可以在 EasyCom 软件里查看。例如我司的产品 TCM1031 V5.5，使用 EasyCom 连接后，输入命令 PCRS232:COMPUSHLIST=1 即可查看支持参数列表:

10.5 查询列表命令的模块名可能为 RS232，PCRS232，UIRS232 中的一个，视产品接口而定，TCM1031 的两个串口分别是 PCRS232 和 UIRS232。

10.6 列表里面显示了参数对应的寄存器地址，数据类型，数据长度（单位字节）。寄存器地址为 3xxx 的为只读参数，可以用 0x04 命令读取；寄存器地址为 4xxx 的为可读写参数，可以用 0x03 命令读取，0x10 命令写新值。寄存器地址是模块为了支持 Modbus 虚拟出来的，每批次的产品的寄存器地址组织顺序可能略有变化，请使用前先验证。

10.7 比如上图中最靠前的 3 个参数：实际温度 TCACTTEMP 寄存器地址为 3001（10 进制），数据类型为 float 浮点数，数据长度为 4 字节，寄存

器数量为 2（Modbus 协议里每个寄存器为 16 位，即 2 字节）；输出状态 TC0E 的寄存器地址位 3003（即上一个参数的地址 3001+上一个参数的寄存器数量 2），数据类型为 UINT16，数据长度为 2 字节，寄存器数量为 1，所以下一个参数 TCLED 的寄存器地址为 3003+1=3004；

```
TC1:TCACTTEMP.REG=3001_FLOAT_4
TC1:TC0E.REG=3003_UINT16_2
TC1:TCLED.REG=3004_UINT16_2
```

10.8 Modbus 里广播站号为 0，我司自定义协议里广播地址为 255，因此我们做如下规定，Modbus 站号 = 我司协议串口地址 + 1；比如我司的模块，默认出厂串口地址为 0，使用 Modbus RTU 通讯时，默认站号为 1。

10.9 支持多个寄存器连读或者连续写，但是要求被读参数本来是连续的，起始寄存器地址必须刚好是某个参数的寄存器地址，总寄存器长度为 n 个完整的参数寄存器长度。如 10.7 里所示的 3 个参数就可以连续读。

10.10 使用时请注意，Modbus 命令中的寄存器地址、寄存器数量的字节序都是高字节在前，低字节在后；CRC 结果却是低字节在前，高字节在后。

10.11 支持的数据类型 1：无符号数 uint16_t，占用两个字节，刚好占用 1 个 modbus 寄存器，传输时高字节在前，低字节在后。

10.12 支持的数据类型 2：无符号数 uint32_t，占用 4 个字节（假设为 ABCD），占用 2 个 modbus 寄存器，传输时高字节在前，低字节在后，传输顺序为 ABCD。

10.13 支持的数据类型 3：单精度浮点数 float，占用 4 个字节，使用 IEEE754 标准表示（仍然假设表示浮点数的 4 个字节为 ABCD），占用 2 个 modbus 寄存器，传输时高字节在前，低字节在后，传输顺序为 ABCD。

10.14 示例：以 MODBUS RTU 来读取 TC1:TCACTEMP，站号为 1，命令为 0x04，寄存器地址为 3001（16 进制：0x0BB9），寄存器数量为 0x0002，所以在 EasyCom 中输入为\01\04\0B\09\00\02，最后的 MODBUS 校验码选择 EasyCom 自动添加（本次命令字符串的校验码添加为\A2\0A），如果是其它串口助手，请手动添加校验码。

10.15 上图下位机接受 Modbus 命令后的返回值中，0x01 为站号，0x04 为命令号，下一个 0x04 为字节数，0x41 0xC7 0xCE 0xB3 为返回的参数值（代表单精度浮点数 24.9759，可以在网上通过浮点数计算器得到），0x4B 0x90 为校验码。

11 修订历史

11.1 2012-8-25，v1.0，完成基本版本。

11.2 2012-9-28，v1.0.1，修改部分文字描述。

11.3 2013-4-29，v2.0，增加串口地址功能。

11.4 2014-7-21，v2.1，增加命令末尾附加地址的新协议内容。

11.5 2014-8-12，v2.2，增加了校验命令格式。

11.6 2014-12-16，v2.2.1，修改部分文字描述。

11.7 2015-1-28，v2.2.2，重新描述了地址功能。

11.8 2015-12-10，v2.2.3，通讯规则里增加更加详细的说明。

11.9 2016-4-34，v2.2.3，增加调试软件的说明。

11.10 2016-12-03，v2.2.3，增加 EasyHost 软件里提供了参数命令的说明。

11.11 2017-4-14，v2.2.3，增加了第三方串口工具软件的说明。

11.12 2017-8-12，v2.2.3，增加了协议里数字发送方式的说明。

11.13 2020-10-13，v3.0，增加了 Modbus 协议支持。

11.14 2022-11-15，v3.0，增加了 Modbus 实例。