

HY-R16TC XX

16 路热电偶温度测量模块

使用说明书 (V2.3)

1、概述

本模块采用高精度 32 位 AD 芯片+ARM32 位工业级 MCU，精度高，抗干扰好。支持 B、C、E、J、K、N、R、S、T 九种热电偶型号测温，自由配置，可广泛用于各种温度测量场合。

具有以下特点：

- ✧ 具有宽电源供电 DC9-30V；
- ✧ 32 位高精度 AD 高分辨率，误差 $\leq \pm 0.1^{\circ}\text{C}$ （热电偶为 K 型时的采样误差，排除热电偶本身误差后的数值），超小温度漂移；
- ✧ 支持 B、C、E、J、K、N、R、S、T 多种热电偶温度传感器。
- ✧ 采样周期具有 0.22s, 0.56s, 1s, 2s 四种速率可设置；
- ✧ 具有奇校验、偶校验、无校验、2 停止位等多种通讯格式可自由设定；
- ✧ 具有 $^{\circ}\text{C}$ （摄氏度）与 $^{\circ}\text{F}$ （华氏度）两种温度单位的数据寄存器可自由读取；
- ✧ 热电偶输入、电源、通讯三种相互隔离，可靠性高；
- ✧ 可以用 RS485 或以太网做为通信接口，当采用以太网版本时，可同时使用 RS485 接口，使模块同时拥有两个通讯口，可用于冗余高可靠场合；
- ✧ 可灵活自选 Modbus-RTU 或 Modbus-TCP 工业通信协议，与各种组态屏、工控软件以及模组进行可靠通信。

2、产品主型号

HY-R16TC-84C2

16 路测温，RS485 接口；

HY-R16TC-64C2

16 路测温，以太网接口+RS485；

3、性能指标

表 3.1 性能参数表

测量参数		
项目	参数	说明
支持热电偶类型	B、C、E、J、K（出厂默认）、N、R、S、T	通过修改 0x0200--0x020F 寄存器选择型号，同一模块每个通道可单独设置，可同时用不同的热电偶）
测温范围	50~1700 $^{\circ}\text{C}$	B 型
	0~2320 $^{\circ}\text{C}$	C 型
	-200~900 $^{\circ}\text{C}$	E 型
	0~750 $^{\circ}\text{C}$	J 型
	-200~1250 $^{\circ}\text{C}$	K 型（出厂默认）
	-270~1300 $^{\circ}\text{C}$	N 型
	0~1450 $^{\circ}\text{C}$	R 型
	0~1450 $^{\circ}\text{C}$	S 型
	-200~350 $^{\circ}\text{C}$	T 型
精度误差 （以 K 型作测试，排除了热电偶本身误差与冷端补偿误差之后的精	$\pm 0.1^{\circ}\text{C}$	2 秒更新速率（出厂默认）
	$\pm 0.1^{\circ}\text{C}$	1 秒更新速率
	$\pm 0.3^{\circ}\text{C}$	0.56 秒更新速率
	$\pm 3^{\circ}\text{C}$	0.22 秒更新速率

度)		
冷端误差	±0.4℃	实际测温误差=精度误差+冷端误差+热电偶本身误差（非简单相加）
分辨率	0.1℃	
温度漂移	≤±200ppm/℃	
数据更新速率	0.22 秒、0.56 秒、1 秒、2 秒(默认)	4 种刷新时间供选择 速度越快，精度会越低 修改 0x0081 寄存器调整刷新时间
电气参数		
项目	参数	说明
隔离耐压	>1500V DC	电源、通信、采集电路三者相互隔离
供电电源	+9V~+30V DC	
额定功耗	≤0.6W	HY-R16TC-84C2
	≤1.2W	HY-R16TC-64C2 带网口型
通讯参数		
项目	参数	说明
输出接口	RS485	HY-R16TC-84C2
	网口+RS485	HY-R16TC-64C2
通讯协议	Modbus-RTU 或 Modbus-TCP	可通过修改 0x01FA 寄存器配置
通讯波特率	4800、9600、19200、38400、57600、115200bps 供选择	RS485 口通过修改 0x0051 寄存器调整，出厂默认为 9600
		网口与主控芯片间波特率通过修改 0x005A 寄存器调整，出厂默认为 115200，一般用户不需调整这个；
数据格式	8 个数据位； 可自由配置无校验/奇校验/偶校验、1 位停止位/2 位停止位。	RS485 口通过修改 0x0052 寄存器调整，出厂默认无奇偶校验、1 位停止位；
		网口与主控芯片间波特率通过修改 0x005B 寄存器调整，出厂默认无奇偶校验、1 位停止位；
RS485 口出厂参数： 地址为 1 号,波特率 9600,无校验,8 个数据位，1 个停止位;默认 Modbus-RTU 协议		
RJ45 网口出厂参数： TCP server 模式，IP:192.168.0.7,端口号:20108;默认 Modbus-TCP 协议 网页登录用户名:admin,登录密码:admin，可修改参数； 也可以用专用工具软件修改参数。		
其它参数		
项目	参数	说明
工作温度	-40℃~+70℃	
工作湿度	<90%	相对湿度
外观尺寸	120*110.6*45.5mm	

4、外形尺寸图

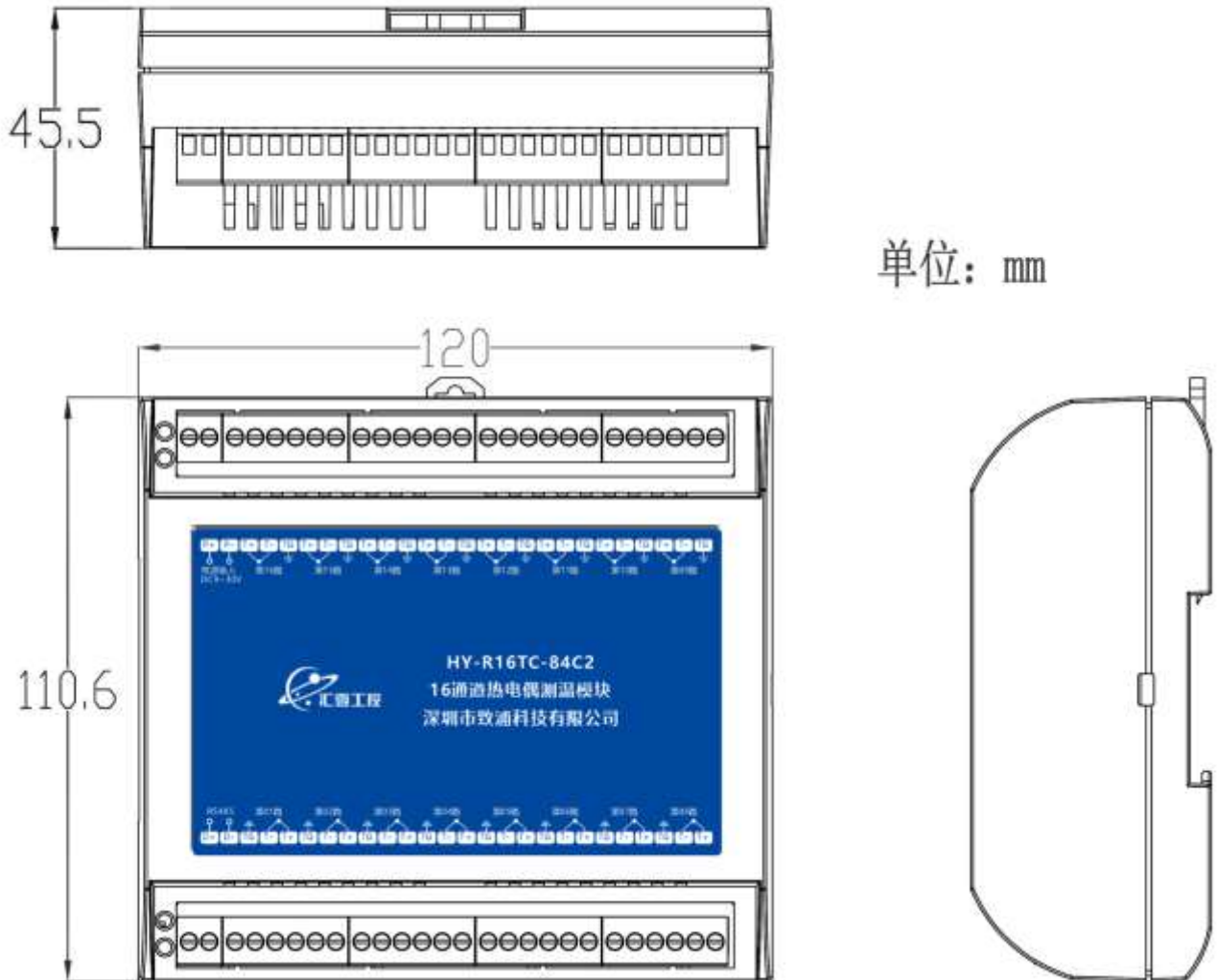


图 4.1

5、按键与端子定义图

5.1 一键复位

一键复位功能能在设置出错时，一键复位至出厂状态，步骤如下：

- a. 打开外壳，按下 PCB 上的 SET 键不松开；
- b. 重新上电或按一下复位键；
- c. 此时保持 SET 键不松开，RUN 指示会先亮 1 秒，然后熄灭 2 秒；
- d. 当出现 RUN 指示灯慢闪时，如果松开 SET 键，则复位通讯设置；
- e. 如果想复位其它设置，则在出现 RUN 灯慢闪后，一直按住 SET 键不松开（约 30 秒），直到 RUN 熄灭，此时会复位所有设置，包括：通讯、采样速率、量程、电网设置等等，但不会复位校正参数。

5.2 端子定义：（热电偶屏蔽线端子可以不接）

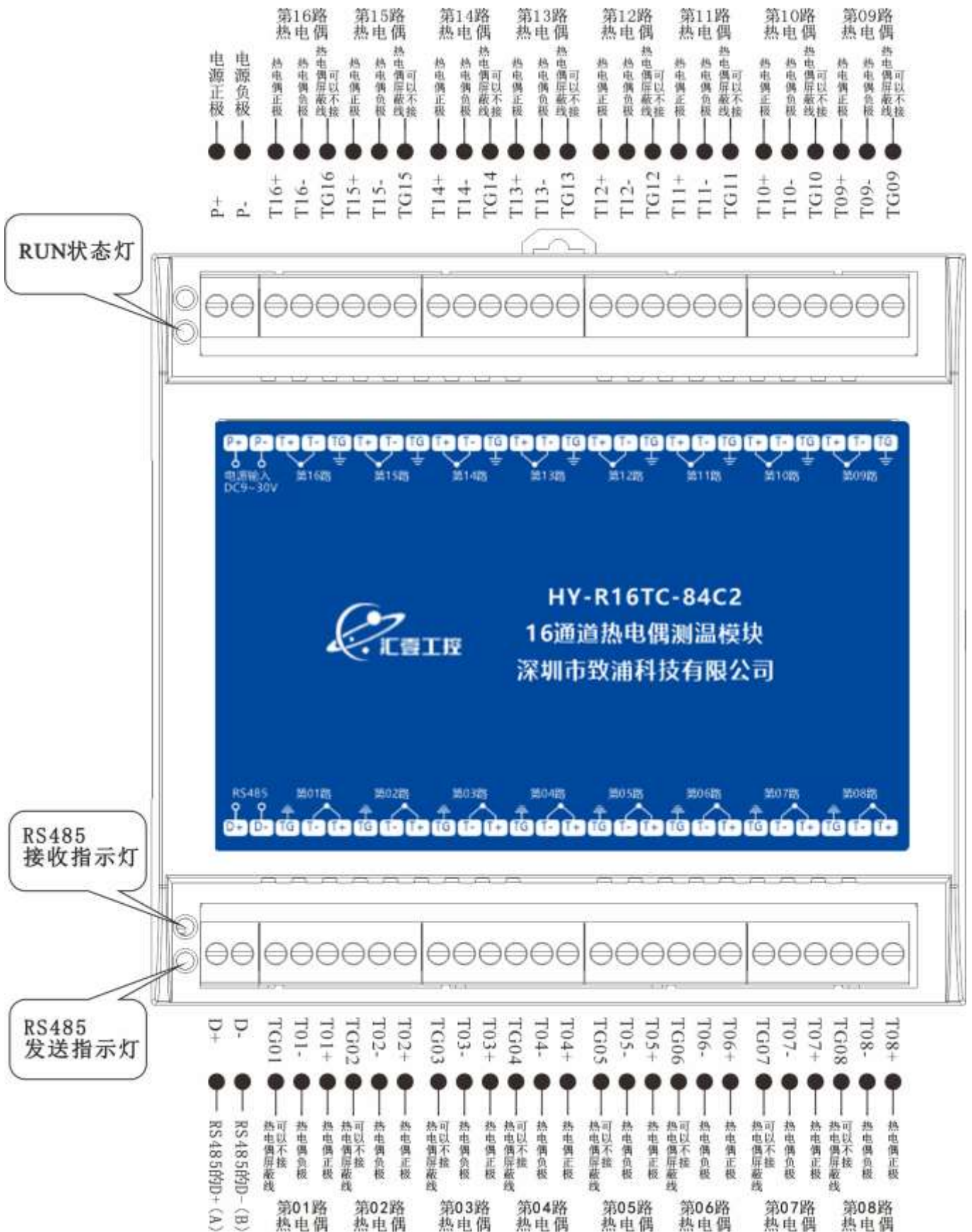


图 5.2.1: RS485 接口版本接线图

5.3 各端子符号说明:

表 5.3.1 端子符号说明

符号	定义	说明
P+	电源正极	
P-	电源负极	
D+	RS485 接口 D+	也称 A 端
D-	RS485 接口 D-	也称 B 端
T+、T01+、T02+... T16+	热电偶正端输入	
T-、T01-、T02-... T16-	热电偶负端输入	
TG、TG01、TG02... TG16	热电偶外屏蔽线端	屏蔽线端需要根据现场需求连接，一般低干扰场合不需要接；同时，有些场合屏蔽线已接大地或设备地，则须看耦合情况接入此屏蔽线端。
RUN 指示灯	运行指示	会根据更新速率闪烁
RXD1	RS485 接收指示灯	收到数据会闪烁
TXD1	RS485 发送指示灯	发送数据会闪烁

6、寄存器说明

6.1 温度寄存器（支持用 03 和 04 功能码读，不能改写）

表 6.1.1 温度寄存器表

寄存器内容		寄存器地址 (十进制)	对应 PLC 或 组态软件 配置地址	寄存器数据说明
摄氏度 格式	第 1 通道温度值	0000	40001	1、数据是 16 位有符号二进制数据(2 字节)，高字节在前，低字节在后。 2、采集的温度分辨率为 0.1 摄氏度。 3、温度值 = 寄存器的数据 ÷ 10 (请参考下表例子)。 4、当数值为 0x7FFF 时，表示传感器断开或过温；0x8000 时，表示传感器短路或超低温。
	第 2 通道温度值	0001	40002	
	第 3 通道温度值	0002	40003	
	.	.	.	
	.	.	.	
	.	.	.	
第 15 通道温度值	0014	40015		
第 16 通道温度值	0015	40016		
华氏度 格式	第 1 通道温度值	0176	40177	1、数据是 16 位有符号二进制数据(2 字节)，高字节在前，低字节在后。 2、采集的温度分辨率为 0.1 华氏度。 3、温度值 = 寄存器的数据 ÷ 10 4、当数值为 0x7FFF 时，表示传感器断开或过温；0x8000 时，表示传感器短路或超低温。
	第 2 通道温度值	0177	40178	
	第 3 通道温度值	0178	40179	
	.	.	.	
	.	.	.	
	.	.	.	
第 15 通道温度值	0190	40191		

第 16 通道温度值	0191	40192	
------------	------	-------	--

数据举例：比如数值 16 进制时为 0x058E，转成 10 进制为 1422，则温度为 142.2℃；

比如数值 16 进制时为 0xFDA3，第 16 位为 1，则为负数，转成 10 进制为-605，则温度为-60.5℃。

6.2 配置字寄存器

此类寄存器只能用 03 功能码读或 06 与 16 功能码写，见表如下：

表 6.2.1 配置字寄存器表

寄存器地址 (Hex)	保持寄存器内容	寄存器个数	寄存器 状态	数据范围
0050H	地址	1	读/写	地址(此值可填 1-253,254 与 255 为广播地址)(默认 01)
0051H	波特率 (RS485 口)	1	读/写	0000 设置波特率-115200bps 0001 设置波特率-9600bps(默认) 0002 设置波特率-19200bps 0003 设置波特率-38000bps 0004 设置波特率-2400bps 0005 设置波特率-4800bps 0006 设置波特率-9600bps 0007 设置波特率-19200bps 0008 设置波特率-38400bps 0009 设置波特率-57600bps 000A 设置波特率-115200bps
0052H	奇偶校验 (RS485 口)	1	读/写	0000 无校验, 1 个停止位(默认) 0001 奇校验, 1 个停止位 0002 偶校验, 1 个停止位 0003 无校验, 2 个停止位 0004 奇校验, 2 个停止位 0005 偶校验, 2 个停止位
0055H	模块名称--高	1	读/写	默认:5A54H (ZT 的 ASCII 码)
0056H	模块名称--中	1	读/写	默认:3136H (16 的 ASCII 码)
0057H	模块名称--低	1	读/写	默认:5443H (TC 的 ASCII 码)
0058H	软件版本	1	读	3033: 03 的 ASCII 码
0059H	软件子版本	1	读	3031: 01 的 ASCII 码
005AH	网口与 MCU 通讯速率	1	读/写	同 0051H
005BH	网口与 MCU 通讯奇偶 校验	1	读/写	同 0052H
0081H	采样速率	1	读/写	0--刷新时间 2 秒 (默认) 1--刷新时间 1 秒 2--刷新时间 0.56 秒 3--刷新时间 0.22 秒 刷新时间越短, 精度越低。

0082H	电网配置	1	读/写	其值为： 50--适用频率为 50HZ 的电网 60--适用频率为 60HZ 的电网
0083H	校正标志	1	读/写	其值为 0xA5F0 时，表示出厂已校正
01FAH	通讯协议定义	1	读/写	详见附件 《如何切换 Modbus-RTU 与 Modbus-TCP 协议》
0200H--020FH	热电偶型号设置	16	读/写	对应 1 至 16 路传感器型号。其值为： 0--B 型热电偶 1--C 型热电偶 2--E 型热电偶 3--J 型热电偶 4--K 型热电偶 5--N 型热电偶 6--R 型热电偶 7--S 型热电偶 8--T 型热电偶 出厂默认为 K 型热电偶
02C0H~02CFH	客户温度补偿	16	读/写	对应 1 至 16 通道温度补偿，用户可填入正负数值修正温度误差。 此值单位为:0.1℃ 实际温度=测量温度+此补偿值 如要改写此值，必须向 8000H 寄存器写 0x0A，改写完后，向 8000H 寄存器写 0x05 加锁保护。
8000H	温度补偿与报警值写保护	1	读/写	向此寄存器写 0x0A 才能改写温度补偿寄存器；改写完后，写 0x05 锁存，并生效温度补偿

版本： V2.3 2022.08.14 更新

附件 1

Modbus-RTU 通讯协议

Modbus-RTU 通讯协议举例

如下所有命令都是以硬件地址为 01 来举例说明；

1. 读模块配置字寄存器命令（03 功能码）

主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
2	03	功能码	1
3	00 55	数据起始寄存器地址，高 8 位在前，低 8 位在后；参照“配置字寄存器表”	2
4	00 02	读取寄存器个数，高 8 位在前，低 8 位在后 （此列读取 2 个寄存器数据）	2
5	D4 1B	CRC 校验码（低位在前，高位在后）	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
2	03	功能码	1
3	04	返回的数据字节个数，2 个寄存器*2	1
4	35 39 30 39	读取的寄存器数据，每 2 个字节表示一个寄存器数据，高位在前，低位在后；第 1 个寄存器数据在前	可变
5	F1 E0	CRC 校验码（低位在前，高位在后）	2

2. 读 8 路温度命令（支持 04 功能码, 字节读）

主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
2	03	功能码	1
3	00 00	起始通道序号，高 8 位在前，低 8 位在后；参照“温度寄存器表”	2
4	00 08	读取 8 个通道温度，高 8 位在前，低 8 位在后	2
5	44 0C	CRC 校验码（低位在前，高位在后）	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址, 可变 (1-253, 254 与 255 为广播地址) (此列为 01 设备地址)	1
2	03	功能码	1
3	10	返回的数据字节个数, 8 个寄存器*2	1
4	01 37 03 05 00 00 ...	读取的寄存器数据, 每 2 个字节表示一个寄存器数据, 高位在前, 低位在后; 第 1 个寄存器数据在前, 数据还原参照“温度寄存器表”	可变
5	CRC	CRC 校验码 (低位在前, 高位在后)	2

3. 配置寄存器修改命令:
3.1 单个寄存器修改命令 (06 功能码, 每次只能修改一个寄存器, 举例修改通讯地址)
主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址, 可变 (1-253, 254 与 255 为广播地址) (此列为 01 设备地址)	1
2	06	功能码	1
3	00 50	寄存器地址, 高 8 位在前, 低 8 位在后, 参照“配置字寄存器表”	2
4	00 02	寄存器数据, 参照“配置字寄存器表”	2
5	08 1A	CRC 校验码 (低位在前, 高位在后)	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址, 可变 (1-253, 254 与 255 为广播地址) (此列为 01 设备地址)	1
2	06	功能码	1
3	00 50	寄存器地址, 返回相同	2
4	00 02	寄存器数据, 返回相同	2
5	08 1A	CRC 校验码, 返回相同	2

3.2 连续修改多个寄存器命令 (16 功能码, 举例修改各通道补偿值)
主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址, 可变 (1-253, 254 与 255 为广播地址) (此列为 01 设备地址)	1
2	10	功能码	1
3	04 40	起始寄存器, 高 8 位在前, 低 8 位在后 参照“配置字寄存器表”	2
4	00 04	写入寄存器长度, 高 8 位在前, 低 8 位在后 (此列写入 4 个寄存器)	2

5	08	写入字节长度（写入寄存器长度 x2）	1
6	00 00 00 01 00 03 00 06	写入的数据, 每 2 个字节表示一个寄存器数据, 高位在前, 低位在后; 此列表示把 0440H 寄存器写入数据 0000H, 0441H 寄存器写入数据 0001H, 0442H 寄存器写入数据 0003H, 0443H 寄存器写入数据 0006H	按序列 8 表示的字节数
7	F4 03	CRC 校验码（低位在前, 高位在后）	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	01	从设备地址, 与主设备发送报文保持一致	1
2	10	功能码	1
3	04 40	起始寄存器, 高 8 位在前, 低 8 位在后 与主设备发送的报文相同	2
4	00 04	写入寄存器长度, 高 8 位在前, 低 8 位在后 与主设备发送的报文相同	2
5	C1 2E	CRC 校验码（低位在前, 高位在后）	2

附件 2:

Modbus-TCP 通讯协议

如下所有命令都是以硬件地址为 01 来举例说明；

1 读模块配置字寄存器命令（03 功能码）

主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文（此处以 3D 46 为列）	2
2	00 01	表示协议标识符（此处以 00 01 为列）	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
5	03	功能码	1
6	00 55	数据起始寄存器地址，高 8 位在前，低 8 位在后；参照“配置字寄存器表”	2
7	00 02	读取寄存器个数，高 8 位在前，低 8 位在后（此列读取 2 个寄存器数据）	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 07	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 7 个字节的数据）	2
4	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
5	03	功能码	1
6	04	返回的数据字节个数，2 个寄存器*2	1
7	35 39 30 39	读取的寄存器数据，每 2 个字节表示一个寄存器数据，高位在前，低位在后；第 1 个寄存器数据在前	可变

2 读 8 路温度命令（支持 04 与 03 功能码，字节读）

主设备发送报文

序列	数据举例	数据说明	字节数
----	------	------	-----

	(16 进制)		
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文（此处以 3D 46 为列）	2
2	00 01	表示协议标识符（此处以 00 01 为列）	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
5	03	功能码 03 或 04	1
6	00 00	起始通道序号，高 8 位在前，低 8 位在后；参照“温度寄存器表”	2
7	00 08	读取 8 个通道温度，高 8 位在前，低 8 位在后	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符，与主设备发送报文保持一致	2
3	00 13	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 19 个字节的数据）	2
4	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
5	03	功能码 03 或 04	1
6	10	返回的数据字节个数，8 个寄存器*2	1
7	01 37 03 05 00 00 ...	读取的寄存器数据，每 2 个字节表示一个寄存器数据，高位在前，低位在后；第 1 个寄存器数据在前，数据还原参照“温度寄存器表”	可变

3 配置寄存器修改命令：

3.1 单个寄存器修改命令（06 功能码，每次只能修改一个寄存器，举例修改通讯地址）

主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符，一般每次通信之后将被要求加 1 以区别不同的通信数据报文（此处以 3D 46 为列）	2
2	00 01	表示协议标识符（此处以 00 01 为列）	2
3	00 06	为数据长度，用来指示接下来数据的长度，高 8 位在前，低 8 位在后（此列表示后面跟随有 6 个字节的数据）	2
4	01	从设备地址，可变（1-253, 254 与 255 为广播地址）（此列为 01 设备地址）	1
5	06	功能码	1
6	00 50	寄存器地址，高 8 位在前，低 8 位在后，参照“配置字寄存器表”	2
7	00 02	寄存器数据，参照“配置字寄存器表”	2

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符, 应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符, 与主设备发送报文保持一致	2
3	00 06	为数据长度, 用来指示接下来数据的长度, 高 8 位在前, 低 8 位在后 (此列表示后面跟随着 6 个字节的数据)	2
4	01	从设备地址, 可变 (1-253, 254 与 255 为广播地址) (此列为 01 设备地址)	1
5	06	功能码	1
6	00 50	寄存器地址, 返回相同	2
7	00 02	寄存器数据, 返回相同	2

3.2 连续修改多个寄存器命令 (16 功能码, 举例修改各通道补偿值)

主设备发送报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符, 一般每次通信之后将被要求加 1 以区别不同的通信数据报文 (此处以 3D 46 为列)	2
2	00 01	表示协议标识符 (此处以 00 01 为列)	2
3	00 0F	为数据长度, 用来指示接下来数据的长度, 高 8 位在前, 低 8 位在后 (此列表示后面跟随着 6 个字节的数据)	2
4	01	从设备地址, 可变 (1-253, 254 与 255 为广播地址) (此列为 01 设备地址)	1
5	10	功能码	1
6	04 40	起始寄存器, 高 8 位在前, 低 8 位在后 参照“配置字寄存器表”	2
7	00 04	写入寄存器长度, 高 8 位在前, 低 8 位在后 (此列写入 4 个寄存器)	2
8	08	写入字节长度 (写入寄存器长度 x2)	1
9	00 00 00 01 00 03 00 06	写入的数据, 每 2 个字节表示一个寄存器数据, 高位在前, 低位在后; 此列表示把 0440H 寄存器写入数据 0000H, 0441H 寄存器写入数据 0001H, 0442H 寄存器写入数据 0003H, 0443H 寄存器写入数据 0006H	按序列 8 表示的字节数

从设备返回正确报文

序列	数据举例 (16 进制)	数据说明	字节数
1	3D 46	为此次通信事务处理标识符, 应答报文要求与先前对应的主设备发送报文保持一致	2
2	00 01	表示协议标识符, 与主设备发送报文保持一致	2
3	00 06	为数据长度, 用来指示接下来数据的长度, 高 8 位在前, 低 8 位在后 (此列表示后面跟随着 6 个字节的数据)	2
4	01	从设备地址, 与主设备发送报文保持一致	1

5	10	功能码	1
6	04 40	起始寄存器，高 8 位在前，低 8 位在后 与主设备发送的报文相同	2
7	00 04	写入寄存器长度，高 8 位在前，低 8 位在后 与主设备发送的报文相同	2

附件 3:

如何切换 Modbus-RTU 与 Modbus-TCP 协议

(本说明适用 HY-R16xx 与 HY-R08xx 全系列产品)

如何在产品中切换 Modbus-TCP、Modbus-RTU、自定义协议以及用 Modbus-RTU 扩展下联模块?

A. 只需要用 06 功能码修改 0x1FA 寄存器就可改变串口的通信协议和工作方式。

B. 0x1FA 寄存器为 16 位寄存器，每 4 位对应一个通讯口设置，列表如下:

表 (1)

0x1FA 寄存器位	对应产品通 讯接口序号	对应产品通信接口	数据含义代码 (16 进制)
Bit3:Bit0	第一通讯口	RS485 口	0x0--从机 Modbus-RTU 协议 (RS485 出厂 默认协议)
Bit7:Bit4	第二通讯口	以太网口	0x1--从机 Modbus-TCP 协议(以太网口出 厂默认协议)
			0x4--从机自定义协议 1
			0x6--从机自定义协议 2

C. 注意: 因为所有通讯口的协议格式存储在同一个寄存器 (0x1FA) 的不同位上(16 位 2 个字节), 而我们用 06 或 16 功能码修改时, 是按字节修改的, 所以在修改一个通讯口的协议时, 要把其它通讯口的原协议代码保留填入, 否则会同步修改。

D. 举例, 当 RS485 口为 Modbus-RTU 协议时, 通过 RS485 口更改通讯协议:

➢ RS485 口保持 Modbus-RTU 协议不变, 以太网口协议修改为 Modbus-TCP, 则需发送命令如下:

命令: 01 06 01 FA 00 10 A9 CB(返回相同代码即修改成功), 解析如下表:

设备地址	功能码	改写的寄存器		改写的的数据		CRC校验码	
		高8位	低8位	高8位 (Bit15:Bit8)	低8位 (Bit7:Bit0)	高8位	低8位
01	06	01	FA	00 ↙ ↘ 第4通讯口 第3通讯口 格式 格式	10 ↙ ↘ 第2通讯口 第1通讯口 格式 格式	A9	CB

注：表中第4通讯口与第3通讯口未用到，填0就可以了。

➤ 当需要把 RS485 由当前通讯协议 Modbus-RTU 更改为 Modbus-TCP 协议，以太网口通讯协议改为 Modbus-RTU 时，，则需发送命令如下：

命令：01 06 01 FA 00 01 69 C7(返回相同指令即修改成功)；解析如下表：

设备地址	功能码	改写的寄存器		改写的的数据		CRC校验码	
		高8位	低8位	高8位	低8位	高8位	低8位
01	06	01	FA	00 ↙ ↘ 第4通讯口 第3通讯口 格式 格式	01 ↙ ↘ 第2通讯口 第1通讯口 格式 格式	69	C7

E. 举例，由 Modbus-TCP 协议更改为 Modbus-RTU：

➤ RS485 口与以太网口当前通讯协议为 Modbus-TCP，如要全改成 Modbus-RTU 协议，则需要发命令：

命令：00 00 00 00 00 06 01 06 01 FA 00 00(返回相同代码即修改成功)；解析如下表：

事务标示符		协议标示符		数据长度		设备地址	功能码	改写的寄存器		改写的的数据	
高8位	低8位	高8位	低8位	高8位	低8位			高8位	低8位	高8位	低8位
00	00	00	00	00	06	01	06	01	FA	00 ↙ ↘ 第4通讯口 第3通讯口 格式 格式	00 ↙ ↘ 第2通讯口 第1通讯口 格式 格式

附 4:

网络接口模块测试与设置方法

1、网口功能特点:

- ❖ 10/100Mbps 自适应以太网接口, 支持 AUTO-MDIX 网线交叉直连自动切换;
- ❖ 工作模式可选择 TCP Serve、TCP Client、UDP Client、UDP Server、Httpd Client;
- ❖ 自定义心跳包机制, 保证连接真实可靠, 可用来检测死连接;
- ❖ 自定义注册包机制, 可检测连接状态, 识别模块, 也可做自定义包头;
- ❖ TCP Server 模式下, 连接 Client 的数量可在 1 到 16 个之间任意设置, 默认 4 个, 已连接 Client 的 IP 可在内置网页状态界面显示, 按连接计算发送/接收数据;
- ❖ TCP Server 模式下, 当连接数量达到最大值时, 新连接是否踢掉旧连接可设置;
- ❖ 支持 TCP Client 短连接功能, 短连接断开时间自定义;
- ❖ 支持超时重启(无数据重启)功能, 重启时间自定义;
- ❖ TCP 连接建立前, 数据缓存是否清理可设置;
- ❖ DHCP 功能, 能够自动获取 IP;
- ❖ MAC 地址可修改, 出厂烧写全球唯一 MAC, 支持自定义 MAC 功能;
- ❖ DNS 功能, 域名解析; DNS 服务器地址可自定义;
- ❖ 支持虚拟串口, 可提供配套的虚拟串口软件;
- ❖ 可以跨越网关, 交换机, 路由器运行; 可以工作在局域网, 也可访问外网;

**网口出厂默认参数: 工作模式: TCP Serve; IP: 192.168.0.7; 端口号: 20108; 用户名: admin; 密码: admin
与主芯片通信波特率 115200pbs, 数据位 8 位, 1 位停止位, 无奇偶校验。**

2、模块工作方式设置(可网页登录设置或用专用的设置软件方式):

2.1 自带内置的网页服务器, 与常规的网页服务器相同, 用户可以通过网页登录设置参数也可以通过网页查看模块的相关状态。网页服务器的端口号可设置, 默认为 80。

默认首页为当前状态界面, 每隔 10s 刷新一次, 显示模块工作状态:

网络发送总数: 通过网络发送数据可以判断 模块发送多少数据到外网;

网络接收总数: 通过接收计数可以判断有多少数据从网络发向模块;

已连接远端 IP/ 网络发送/ 接收: 通过此项, 可以看到 模块 与哪一个设备进行连接, 该连接发送和接收的数据量有多少, 目前只支持 5 个连接状态显示。

UDP Server 模式下, 只显示发送/接收数据, 不显示连接 IP。



图一、网页工作状态显示页面



图 2、模块参数网页设置页面

2.2 可至我司网站下载专用的设置工具软件，设置更直观快捷。

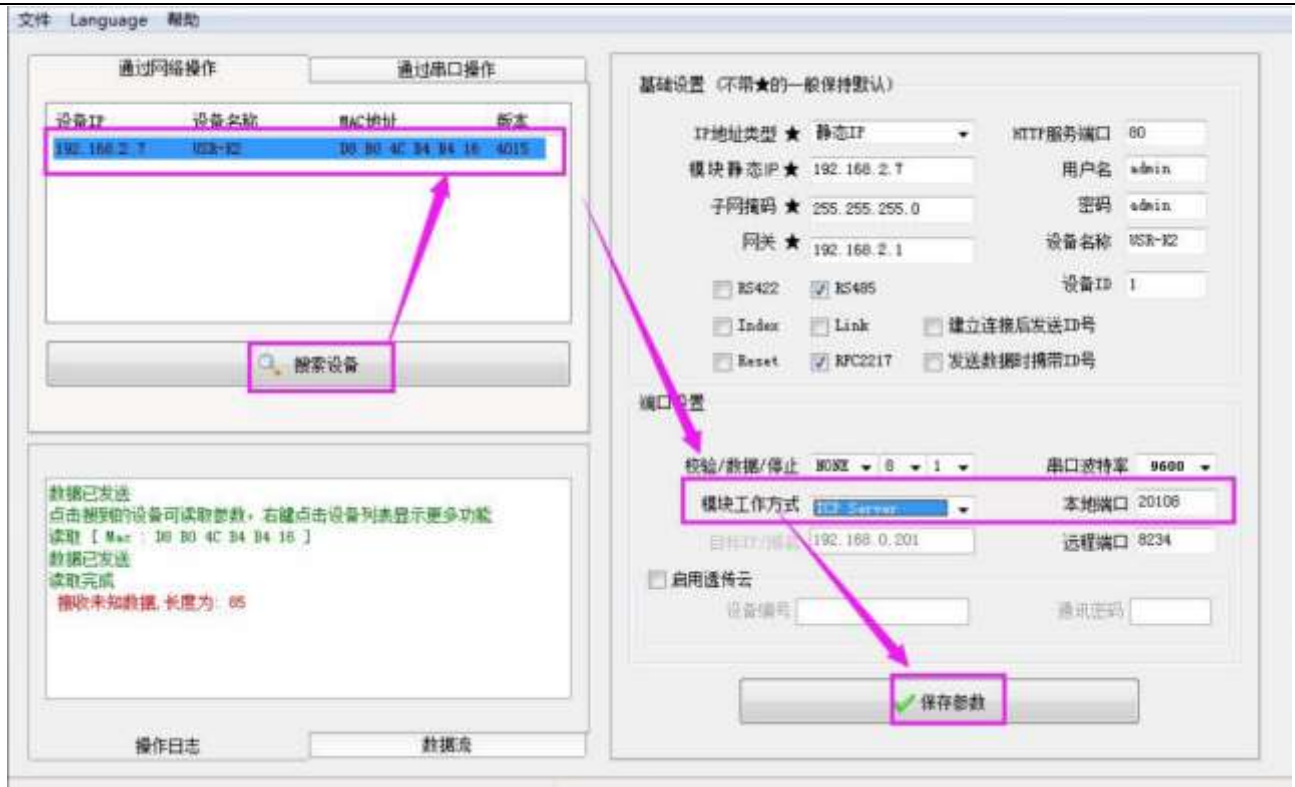


图 3、模块参数软件设置页面（可到本公司官网下载“网络设置软件”）

3、TCP Serve 模式通讯实例

模块设置为 TCP Serve 模式，IP 为 192.168.2.7，端口为 20108 的情况下，打开调试助手软件（本软件可以在本公司网站下载“串口调试助手”）按以下页面设置，本地 IP 需选择正错的本机电脑 IP；

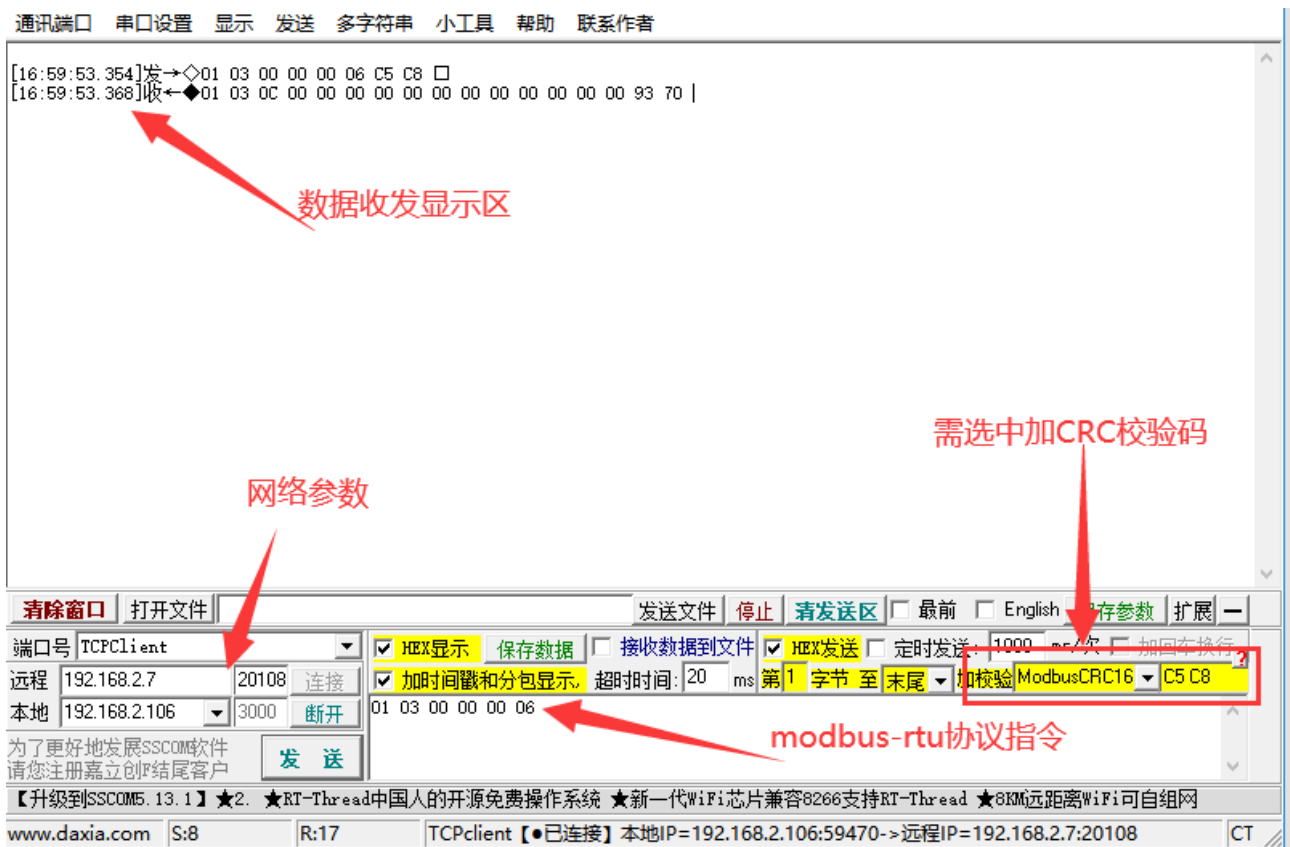


图 4、modbus-rtu 协议指令测试页面

通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者

```
[16:44:13.827]发->◇00 00 00 00 00 06 01 03 00 00 00 06 □  
[16:44:13.839]收←◆00 00 00 00 00 0F 01 03 0C 00 00 00 00 00 00 00 00 00 00 00
```

通讯成功后的发与收数据,如果通讯不成功只有发,没有收的数据

网络连接参数

清除窗口 打开文件 发送文件 停止 清发送区 最前 English 保存参数 扩展

端口号 TCPClient HEX显示 保存数据 接收数据到文件 HEX发送 定时发送: 1000 ms/次 加回车换行

远程 192.168.2.7 20108 连接
本地 192.168.2.106 3000 断开

加时间戳和分包显示, 超时时间: 20 ms 第1 字节 至 末尾 加校验 None

00 00 00 00 00 06 01 03 00 00 00 06

modbus-tcp协议指令

为了更好地发展SSCOM软件
请您注册嘉立创VIP会员

【升级到SSCOM5.13.1】★2. ★RT-Thread中国人的开源免费操作系统 ★新一代WiFi芯片兼容8266支持RT-Thread ★8KM远距离WiFi可自组网

www.daxia.com S:12 R:21 TCPclient 【●已连接】本地IP=192.168.2.106:59470->远程IP=192.168.2.7:20108 CT

图 5、modbus-tcp 协议指令测试页面