

直流电源使用安全注意事项

在安装、维护我司直流电源产品时，为避免出现安全隐患， 请注意以下安全条例

1. 在使用本产品前， 请仔细阅读产品使用说明书。
2. 负责安装、维护我司设备的人员， 必须先经过培训， 了解各种安全注意事项。
3. 本产品应放置、安装在远离液体的区域， 以防止液体进入设备内部造成短路 。
4. 本产品放置、使用过程中要确保设备内无凝露。
5. 产品搬运过程中， 注意防止划伤或跌落砸伤。
6. 产品上电调试过程中， 请穿戴好电工绝缘防护设备， 避免电击危险。
7. 禁止带电安装、拆除带电模块或线缆。
8. 严禁在雷电、雨、雪、大风等恶劣天气下安装、使用设备。
9. 禁止在易燃易爆区域内使用。
10. 产品运行过程中请勿触摸产品外壳。
11. 建议产品运行结束 10分钟内不要触摸产品外壳， 避免烫伤或者电击危险。
12. 设备贴有铭牌， 其中包含产品相关的重要参数信息， 严禁人为涂改和损坏。
13. 非专业人士请勿打开产品内部， 避免内部高压电击危险。

直流电源DC20K-100介绍

产品特点

1. 高环境可靠性：大面积灌胶防护，防盐雾、防潮湿、防霉变等显著提高。
2. 内置放电电路：自动泄放残留电荷，简化系统设计，提高系统安全性。
3. 三相无中线交流输入：简化系统设计，消除中线大电流风险，降低系统成本。

产品功能

1. 输入限功率控制
输出功率与输入电压的关系。当输入电压在 $380V_{ac} \pm 10\%$ 之间时，单机可以输出最大功率 20 KW。
- 2 . 输出恒功率控制
额定输入电压时， 电源允许输出功率为 20KW，模块输出电压与输出电流的关系参考V-I曲线。
3. 温度限功率
55℃环境温度以下，电源满功率输出20kW。
4. 具有无级限流功能；限流点在 1A~200A 范围内可调。
- 5 . 输出电压调节调整最小调节步距为 0.1Vdc。
6. 输入过/欠压保护
三相交流输入电压小于 323Vac或者大于 520Vac时电源将停止工作
7. 过压保护点可设置 ，设置范围为 10Vdc~100Vdc，出厂默认值为 100Vdc，过压保护后需要重启可开机。
8. 过温保护
环境过温保护点为 70℃。当内部温度过高，电源过温度保护关机；温度恢复后才能开机工作。
9. 其他保护，电源短路时保护，风扇故障时保护。
10. 通讯接口采用 RS485 modbus 协议通信和网口协议。

技术规格表

类别	名称	参数
环境条件	工作温度	-40℃~+75℃， 55℃ 以上需降额使用
	储存温度	-40℃~+70℃
	相对湿度	≤95%RH，无冷凝
	冷却方式	风冷
	大气压力	79kPa~106kPa
	IP 防护等级	IP20
交流输入	输入制式	三相+PE
	电压范围	320Vac~520Vac
	额定电压	380Vac
	最大电流	40A
	电网频率	45Hz~65Hz
	额定频率	50Hz/60Hz
直流输出	有效电压范围	2Vdc~100Vdc
	电流范围	1~200A（限流点可以设）
	稳压精度	<±1%+0.5%FS（10~100V， 0~20MHz）
	稳流精度	≤±1%+0.5%FS(输出负载 20%~100%额定范围)
	纹波系数	<±0.5%+0.5%FS（10~100V， 0~20MHz）
	电网调整率	≤±0.1%（输入交流电压范围为 320V~520V）
安规参数	接地电阻电压纹波峰值因数 THD	≤5% @50%~100%满载输出功率 接地电阻≤0.1Ω， 应能承受电流≥80A
	漏电流	漏电流≤3.5mA
	绝缘电阻	直流部分、交流部分对外壳之间以及交流部分对直流部分之间的绝缘电阻≥10MΩ
机械参数	尺寸	26.6cm（高）×48cm（宽）×65cm（深不含端子和把手）
	重量	≤35kg

设备图片参考：（触摸屏）

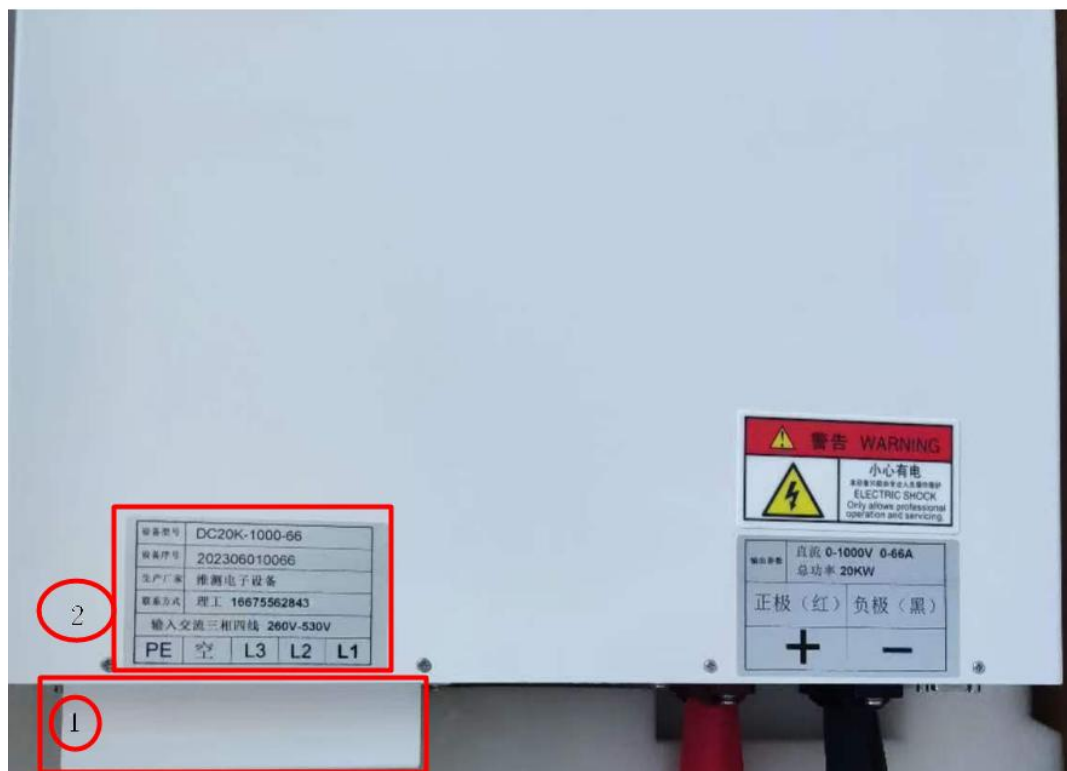


设备安装

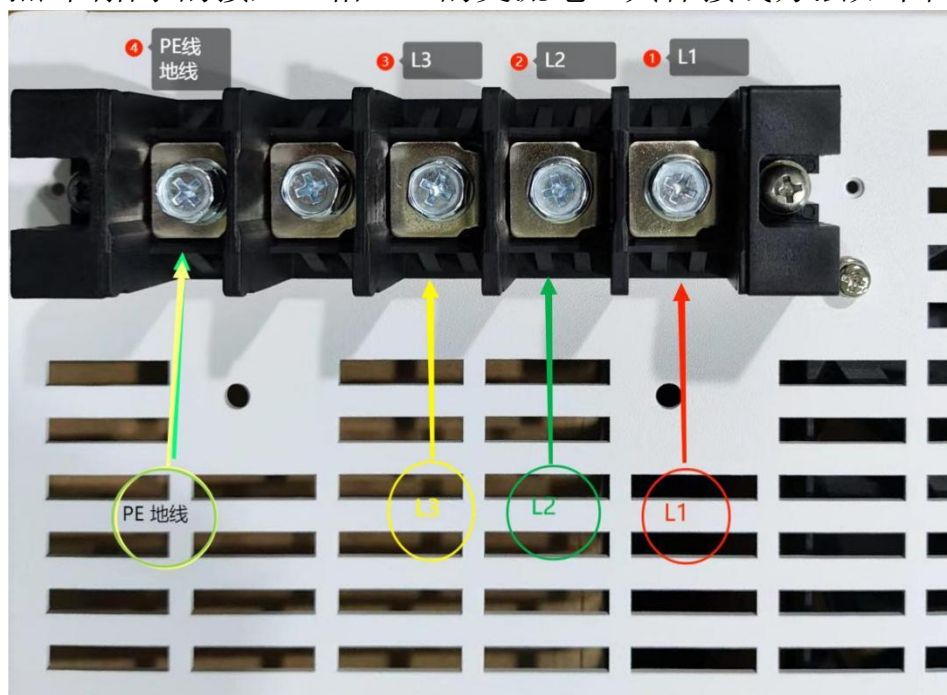
1.1从包装箱子中取出直流电源



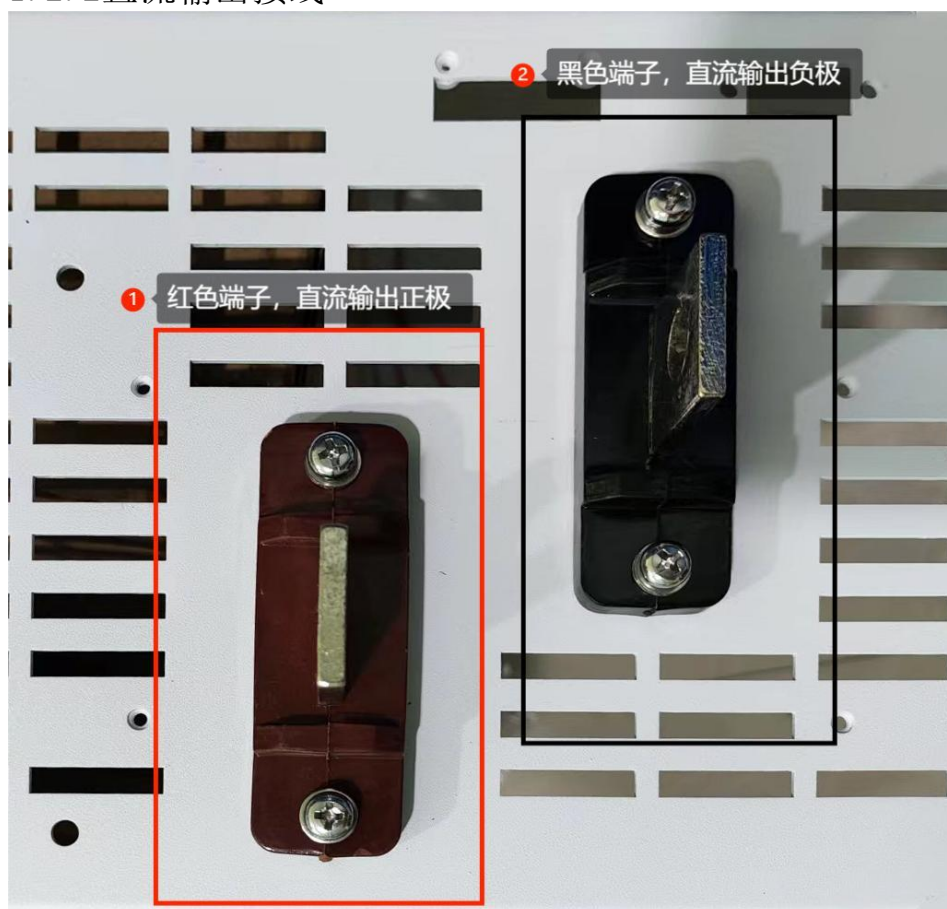
1.2直流电源通电，交流输入接线



1. 2. 1在直流电源尾部，红框1所示，用十字螺丝刀打开保护盖，依照2图指示的接入三相380V的交流电。具体接线方法如下图



1. 2. 2直流输出接线



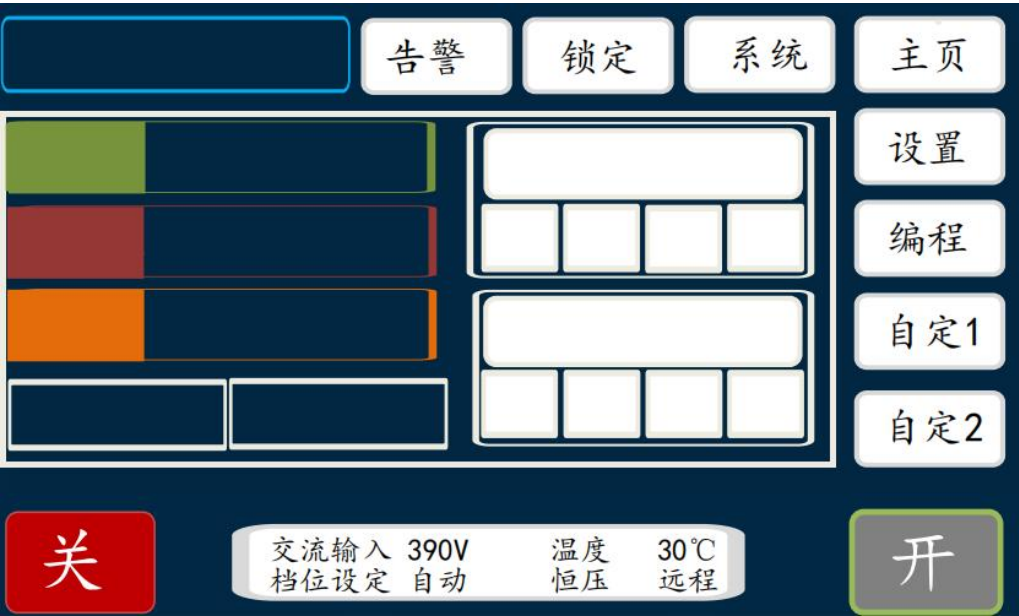
2.设备上电开机

2.1接好线路之后，在直流电源前面板，开机。

2.2开机自检，约10秒



2.2.1自检完成后，到主界面

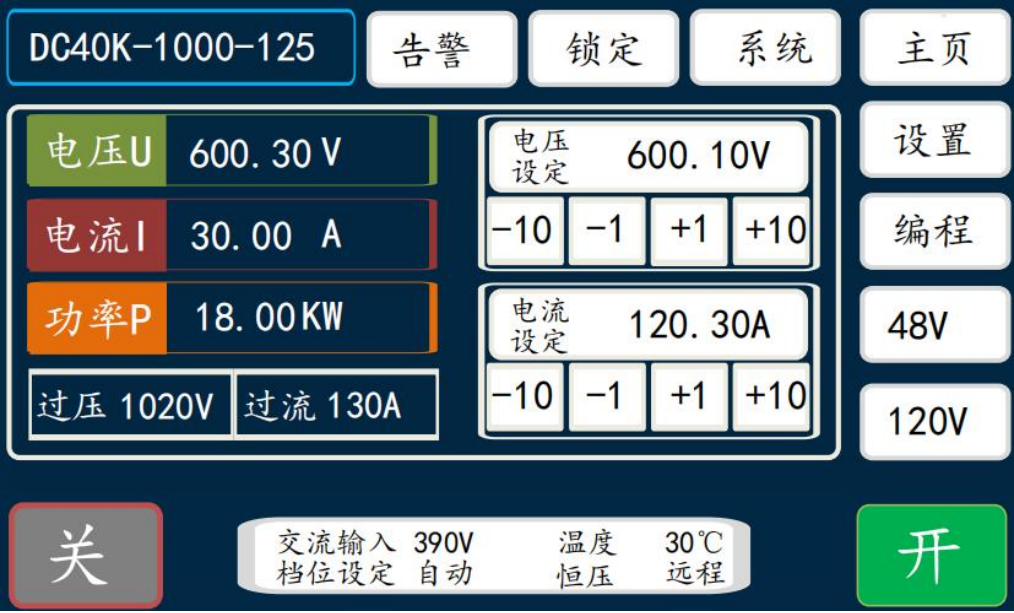


2.2.2输出电压设置

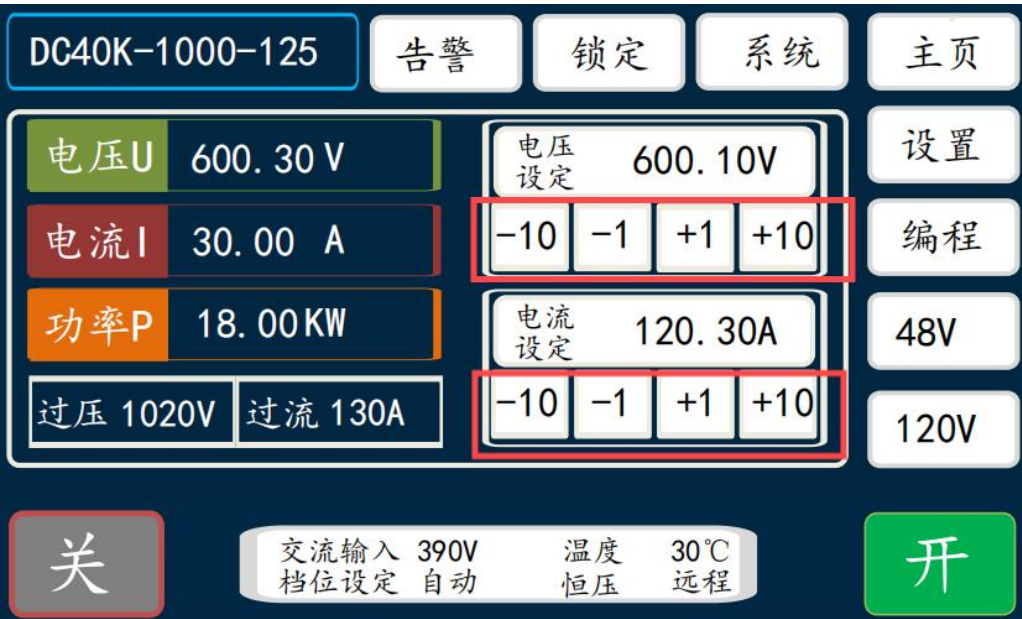
在主界面状态下，按下电压设定框，会出现小键盘，在小键盘上输入需要的电压，在小键盘按一下确认键，电压设置成功，按下开 /关 按钮即可输出，输出状态下开按钮底色会从灰色变成绿色。

通过数字小键盘按键输入需要的电压，如600.10V，按键顺序6 0 0 . 1 0 之后再按下确认

按下屏幕 开 按键有直流输出时，按键会变为绿色，电源输出工作。如下图



2. 2. 3在输出状态下可以微调按键调整输出电压。下图红色框框中的-10 -1 +1 +10，可以进行当前电压加10V 加1V 减10V 减1V的操作，按下-10 后设定电压会调整为 $600.10-10.00=590.10V$ ，电流设置同理



2.3自定义电压和电流，及时调用。在主界面按下设置进入设置界面再按下自定义按键

485串口

网口

保护

自定义

保存

主页

电压(V)

电流(A)

自定义0

自定义1

自定义2

在自定义出设定常用的电压 电流信息可以在主页直接调用比如
自定义0 电压5 电流10
自定义1 电压48 电流100
自定义2 电压120 电流120
按屏幕上保存键，保存后重启上电
开机默认电压5V 电流10A 红框部分自定义电压生效

DC40K-1000-125

告警

锁定

系统

主页

电压U

600.30 V

电流I

30.00 A

功率P

18.00 KW

过压 1020V

过流 130A

电压设定

600.10V

-10

-1

+1

+10

电流设定

120.30A

-10

-1

+1

+10

设置

编程

48V

120V

关

交流输入 390V

档位设定 自动

温度 30℃

恒压 远程

开

2.4直流电源本地编程设置，主界面按一下编程按键。

电压U

电流I

功率P

当前步:

运行循环:

运行时间:

步循环:

步数:

步	电压(V)	电流(I)	时间(s)

编程设定

步:

电压:

电流:

时间:

保存

主页

确认

编程设定框 编辑需要的电压电流跳变，可以实现100步编辑，

比如实现5次跳变100次 设定如下

设定 步 1 电压100 电流30 时间30 按保存按键

设定 步 2 电压390 电流100 时间60 按保存按键

设定 步 3 电压220 电流100 时间180 按保存按键

设定 步 4 电压112 电流80 时间300 按保存按键

设定 步 5 电压88 电流80 时间999 按保存按键

完成编程后在步循环处输入100 步数处输入5 再按确认按键 运行按键就可以输出已经编辑的程序。

电压U

V

电流I

A

功率P

KW

当前步:10

运行循环:12

运行时间:9999m

步循环:22

步数:100

步	电压(V)	电流(I)	时间(s)
1	100	30	30
2	390	100	60
3	220	100	180
4	112	80	300
5	88	80	999

编程设定

步:

电压:

电流:

时间:

结束

运行

主页

编辑

2.5直流电源通讯设置，支持RS 485 MODBUS RTU和 LAN MODBUS TCP协议，按设置界面，选择需要设置的通讯。设定好后按下保存。

485设置

485串口网口保护自定义保存主页

波特率：9600192003840057600115200

地址：

RS 485 MODBUS协议

网口设置

485串口网口保护自定义保存主页

IP地址：

端口地址：

协议类型：TCP

工作模式：客户端服务器

2.6直流电源保护设置，过电压，过电流，过功率设置。按下设置按键，再按保护按键进入。设定好后按下保存。

485串口网口保护自定义保存主页

过电压 (V):

过电流 (A):

过功率(W):

2.7设备信息查看，按键系统可以进入系统界面查看设备相关信息以及帮助信息。

额定选项界面序列号保存主页

型号

设备型号

额定电压

额定电流

额定功率

模块电流

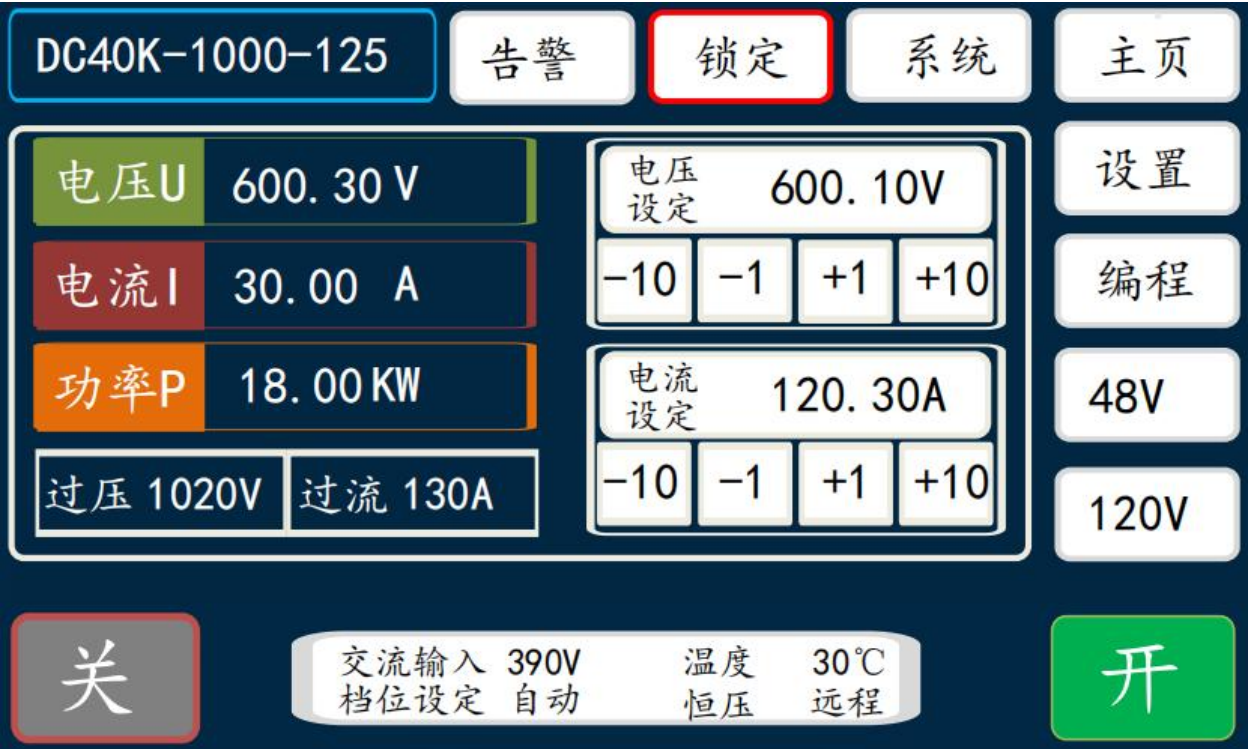
交流电压

NO

K

KP

2.8系统锁定按键，在长时间工作调试中，防止中途有人打断试验，设置运行后，可以按住锁定按键3秒，可以锁定屏幕其他地方，防止误操作。再次长按3秒可以解除锁定状态。如下图



操作手册提供操作思路参考流程，实际设备在不断升级中，会有和操作手册不匹配的地方，请及时联系厂家更新。

直流电源modbus协议

目录

1 功能特性	错误！未定义书签。
1.1 外观特性	错误！未定义书签。
2 MODBUS 通讯协议	15
2.1 协议简介	15
2.2 通信协议介绍	15
2.3 地址码	15
2.4 功能码	15
2.4.2 读取保持寄存器 (0x03)	16
2.4.3 写入保持寄存器 (0x10)	16
2.5 数据区	16
2.6 错误校验码 (CRC校验)	16
2.7 协议寄存器介绍	17
2.7.1 读取保持寄存器 (0x03)	17
2.7. 2 写入保持寄存器(0x10)	17
3 实例	18
3.1 控制流程	18
3.1 读输入寄存器 (0x03)	18
3.3 写保持寄存器 (0x10)	18
3.3.1 进入远程	18
3.3.2 写电压5V	19
3.3.3 写电压1A	19
3.3.4 写开机	20
3.3.5 写关机	20
3.3.6 写本地	20

2 MODBUS 通讯协议

2.1协议简介

通讯接口采用RS485 ，默认波特率9600，无校验。
通信协议为 MODBUS-RTU ；
设备地址默认为： 85；
支持功能码 0x03、0x10。
电压/电流精度, 1位小数位。

2.2通信协议介绍

信息传输为异步方式，Modbus-RTU模式以11位的字节为单位

字格式（串行数据）	10位二进制
起始位	1位
数据位	8位
奇偶校验位	无
停止位	1位

数据帧结构：

数据帧间隔	地址码	功能码	数据区	CRC校验
3.5字节以上	1字节	1字节	N字节	2字节

发送数据前要求数据总线静止时间即无数据发送时间大于 3.5（例如：波特率为 9600 时为 5ms）消息发送至少要以 3.5 个字节时间的停顿间隔开始，整个消息帧必须作为一连续的数据传输流，如果在帧完成之前有超过 3.5 个字节时间的停顿时间，接收设备将刷新不完整的消息并假定下一字节是一个新消息的地址域。同样地，如果一个新消息在小于 3.5 个字符时间内接着前个消息开始，接收的设备将认为它是前一消息的延续。

2.3地址码

地址码是每次通讯信息帧的第一字节（8位），从1到247。这个字节表明由用户设置地址的从机将接收由主机发送来的信息。每个从机都必须有唯一的地址码，并且只有符合地址码的从机才能响应回送信息。当从机回送信息时，回送数据均以各自的地址码开始。主机发送的地址码表明将发送到的从机地址，而从机返回的地址码表明回送的从机地址。相应的地址码表明该信息来自于何处。

2.4功能码

功能码为每次通讯信息帧传送的第二个字节，ModBus通讯规约可定义的功能码为1到127。作为主机请求发送，通过功能码告诉从机应执行什么动作。作为从机响应，从机返回的功能码与从主机发送来的功能码一样，并表明从机已响应主机并且已进行相关的操作。本机仅支持0x03、0x10功能码。

功能码	定 义	操 作（二进制）
0x03	读保持寄存器（可写）	读取一个或多个保持寄存器（可写）的数据
0x10	写多个寄存器	把多组二进制数据写入多个寄存器

2.4.2读取保持寄存器（0x03）

0x03读取功能主机格式

地址码	功能码	寄存器起始地址	寄存器地址数量n	CRC校验码
1字节	1字节	2字节	2字节	2字节

0x03读取功能从机返回格式

地址码	功能码	返回字节数2*n	寄存器数据	CRC校验码
1字节	1字节	1字节	2*n个字节	2字节

2.4.3写入保持寄存器（0x10）

0x10写单个寄存器功能主机格式

地址码	功能码	寄存器地址	寄存器数量n	字节数量	寄存器数据	CRC校验码
1字节	1字节	2字节	2字节	寄存器数x2字节	寄存器数量n x2字节	2字节

0x06写单个寄存器功能从机返回格式

地址码	功能码	寄存器地址	寄存器数据	CRC校验码
1字节	1字节	2字节	2字节	2字节

2.5数据区

数据区包括需要由从机返送何种信息或执行什么动作，这些信息可以是数据（如：开关量输入/输出、模拟量输入/输出、寄存器等等）、参考地址等。例如，主机通过功能码03告诉从机返回寄存器的值（包含要读取寄存器的起始地址及读取寄存器的长度），则返回的数据包括寄存器的数据长度及数据内容。

2.6错误校验码（CRC校验）

主机或从机可用校验码进行判别接收信息是否正确。由于电子噪声或一些其它干扰，信息在传输过程中有时会发生错误，错误校验码（CRC）可以检验主机或从机在通讯数据传送过程中的信息是否有误，错误的信息可以放弃（无论是发送还是接收），这样增加了系统的安全和效率。MODBUS通讯协议的CRC（冗余循环码）包含2个字节，即16位二进制数。

CRC码由发送设备（主机）计算，放置于发送信息帧的尾部。接收信息的设备（从机）再重新计算接收到信息的CRC，比较计算得到的CRC是否与接收到的相符，如果两者不相符，则表明出错。CRC校验码发送时低位在前，高位在后。

CRC码的计算方法：

- 预置1个16位的寄存器为十六进制FFFF（即全为1）；称此寄存器为CRC寄存器；
- 把第一个8位二进制数据（既通讯信息帧的第一个字节）与16位的CRC寄存器的低8位相异或，把结果放于CRC寄存器；
- 把CRC寄存器的内容右移一位（朝低位）用0填补最高位，并检查右移后的移出位；
- 如果移出位为0：重复第3步（再次右移一位）；如果移出位为1：CRC寄存器与多项式A001（1010 0000 0000 0001）进行异或；
- 重复步骤3和4，直到右移8次，这样整个8位数据全部进行了处理；
- 重复步骤2到步骤5，进行通讯信息帧下一个字节的处理；
- 将该通讯信息帧所有字节按上述步骤计算完成后，得到的16位CRC寄存器的高、低字节进行交换；

最后得到的CRC寄存器内容即为CRC码。

2.7协议寄存器介绍

保持寄存器用命令码0x03读取，用命令码0x10写入。

2.7.1读取保持寄存器（0x03）

（单个寄存器地址内的数据为双字节型数据）

序号	名称	地址(16进制)	字数(U16)	范围	小数位	单位	读写	备注
1	输出电压	0x01	2	0-U _{max}	3	mV	R	
3	输出电流	0x03	2	0-I _{max}	3	mA	R	
5	输出功率	0x05	2		0	—	R	

2.7.2写入保持寄存器(0x10)

（单个寄存器地址内的数据为双字节型数据）

序号	名称	地址	字数(U16)	范围	小数点	单位	读写	备注
11	电压设定值	0x11	2	0-U _{max}	3	mV	R/W	
13	电流设定值	0x13	2	0-I _{max}	3	mA	R/W	
15	输出开关	0x15	2	0, 1	0	—	R/W	0开机；1 关机
17	远程	0x17	2	0, 1	0	—	R/W	0本地；1 远程

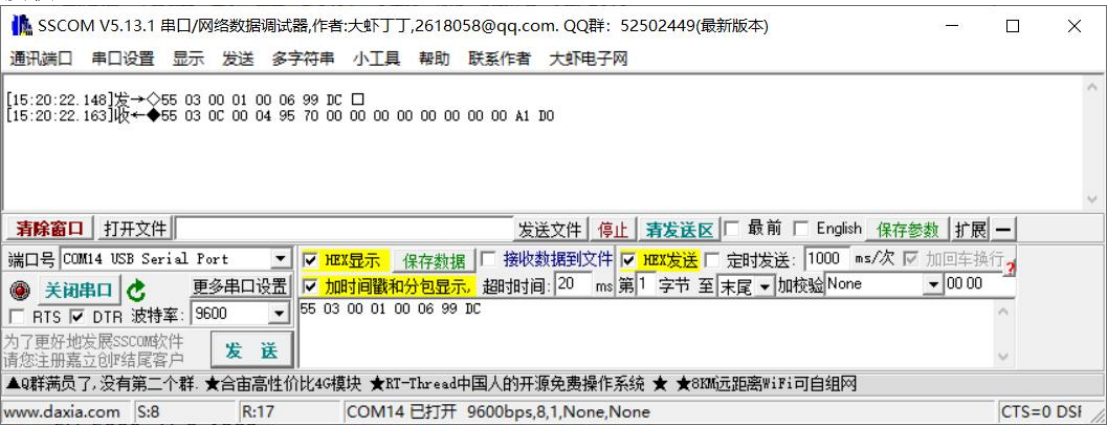
3实例

3.1控制流程

- 1、进入远程：写入远程模式，1
- 2、写入电压设置
- 3、写入电流设置
- 4、输出开关，0
- 5、读取输出电压电流状态
- 6、输出开关，1
- 7、退出远程：写入远程模式，0

3.1读输入寄存器（0x03）

查询电压、电流、功率
发送55 03 00 01 00 06 99 DC
接收0x00049570=300400mV=300.4V



3.3写保持寄存器（0x10）

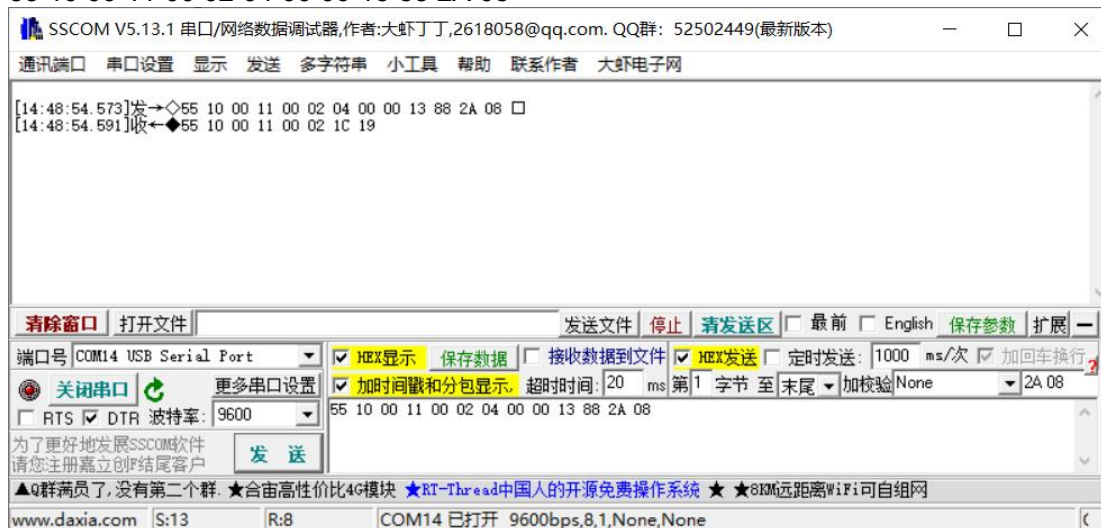
3.3.1进入远程
55 10 00 17 00 02 04 00 00 00 01 66 B4



3.3.2写电压5V

5V=5000mV=0x1388

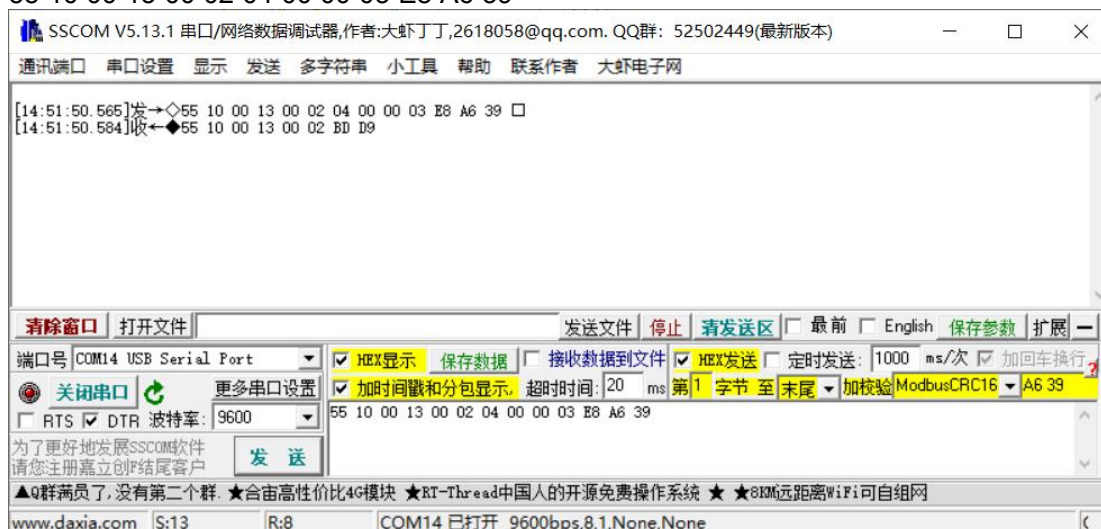
55 10 00 11 00 02 04 00 00 13 88 2A 08



3.3.3写电压1A

1A=1000mA=0x03E8

55 10 00 13 00 02 04 00 00 03 E8 A6 39



3.3.4写开机

55 10 00 15 00 02 04 00 00 00 00 26 AD



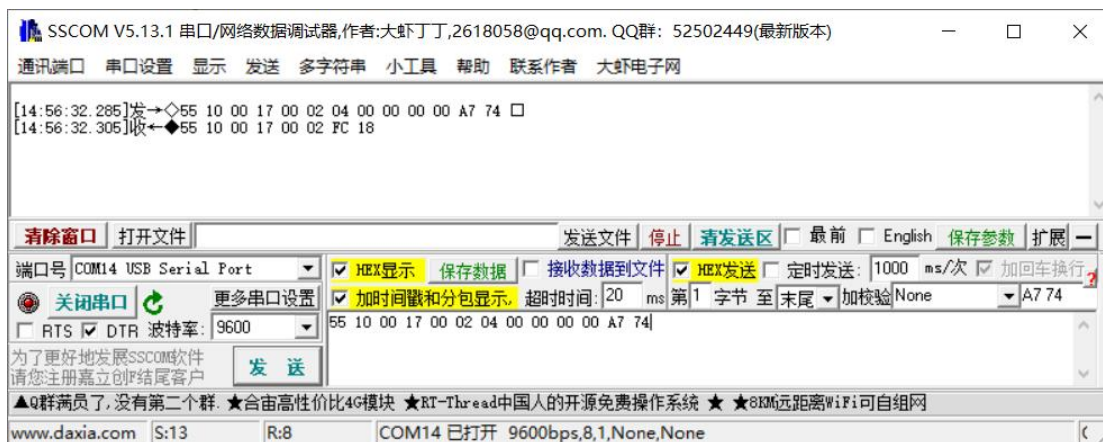
3.3.5写关机

55 10 00 15 00 02 04 00 00 00 01 E7 6D



3.3.6写本地

55 10 00 17 00 02 04 00 00 00 00 A7 74



网口Modbus TCP协议介绍

目录

- 1配置信息21
- 1 Modbus TCP21
 - 1.1 Modbus TCP与其他Modbus变体的区别22
 - 1.2 Modbus TCP协议原理解析22
 - 1.2.1 通信模型22
 - 1.2.2 数据传输方式22
 - 1.2.3 报文结构22
 - 1.2.4 MBAP头22
 - 1.2.5 PDU22
 - 1.2.6 常用功能码23
 - 1.2.7 数据映射23
 - 2协议寄存器介绍23
 - 2.1读取保持寄存器（0x03）23
 - 2.2写入保持寄存器(0x10)23
- 3实例24
 - 3.1控制流程24
 - 3.2读输入寄存器（0x03）24
 - 3.3写保持寄存器（0x10）24
 - 3.3.1进入远程24
 - 3.3.2写电压5V25
 - 3.3.3写电流1A25
 - 3.3.4写开机26
 - 3.3.5写关机26
 - 3.3.6写本地26

1配置信息

默认IP: 192.168.1.222
端口: 502
通讯协议: ModbusTCP
服务端

1 Modbus TCP

Modbus TCP（也称为Modbus TCP/IP）是Modbus协议的一个变种，基于TCP/IP协议栈在以太网上进行通信。它继承了Modbus协议的所有优点，同时利用以太网的高带宽和稳定性，提供更快速和可靠的数据传输。

1.1 Modbus TCP与其他Modbus变体的区别

Modbus协议有多种实现方式，包括：

- Modbus RTU：基于串行通信（如RS-485），数据传输效率较低，适用于点对点或点对多点的简单网络。
 - Modbus ASCII：同样基于串行通信，但使用ASCII码表示数据，便于调试，但效率更低。
 - Modbus TCP：基于以太网，支持更高的数据传输速率和更复杂的网络拓扑。
- 与传统的Modbus RTU相比，Modbus TCP消除了串行通信的限制，支持更大的网络规模和更高的数据传输速度，适用于现代工业自动化系统。

1.2 Modbus TCP协议原理解析

1.2.1 通信模型

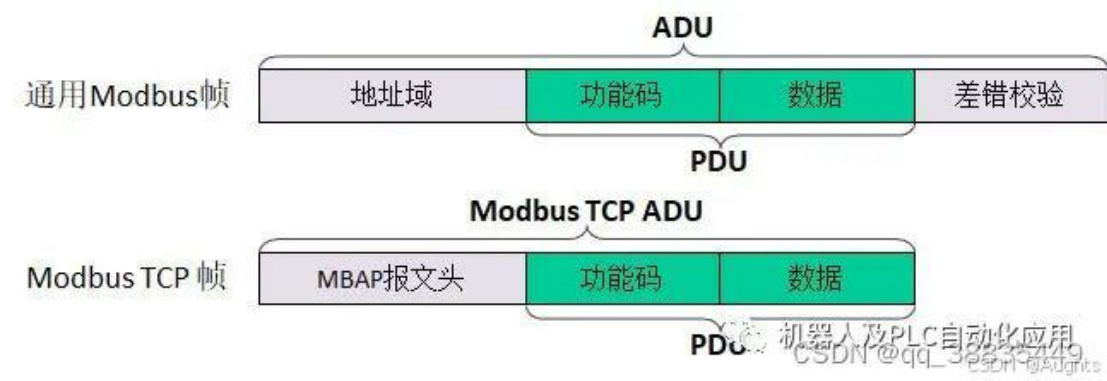
Modbus TCP采用客户端-服务器（Master-Slave）的通信模型。客户端发起请求，服务器响应请求。一个网络中可以有多个客户端和服务端，服务器通常是PLC、传感器、仪表等工业设备。

1.2.2 数据传输方式

Modbus TCP基于TCP协议，采用面向连接的通信方式。通信过程包括建立连接、数据传输和断开连接。数据传输过程中，客户端和服务端通过TCP端口（默认502）进行数据交换。

1.2.3 报文结构

Modbus TCP的报文由两部分组成：



MBAP头（Modbus Application Protocol Header）

PDU（Protocol Data Unit）

1.2.4 MBAP头

MBAP头包含7个字节，用于管理传输层的数据。具体结构如下：

域	长度	描述
事务处理标识符	2字节	可以理解为报文的序列号，一般每次通信之后就要加1以区别不同的通信数据报文
协议标识符	2字节	00 00表示Modbus TCP协议
长度	2字节	表示接下来的字节长度，单位字节
单元标识符	1字节	串行链路或其他总线上连接的远程从站地址

- 事务标识符（Transaction Identifier）：2字节，用于匹配请求和响应。
- 协议标识符（Protocol Identifier）：2字节，固定为0，表示Modbus协议。
- 长度字段（Length Field）：2字节，表示后续PDU的长度。
- 单元标识符（Unit Identifier）：1字节，用于识别设备，特别在串行通信中用于区分不同的从设备。

1.2.5 PDU

PDU包含功能码和数据部分。

功能码（Function Code）：1字节，指示请求的操作类型。
数据（Data）：变长，根据功能码不同，数据内容和长度不同。

1.2.6 常用功能码

以下是一些常用的Modbus TCP功能码：

- 03：读取保持寄存器（Read Holding Registers）
- 16：写多个寄存器（Write Multiple Registers）

1.2.7 数据映射

Modbus协议通过地址映射实现数据的读写。主要包括：

保持寄存器（Holding Registers）：用于存储16位的读写数据（如设定值）。

2协议寄存器介绍

保持寄存器用命令码0x03读取，用命令码0x10写入。

2.1读取保持寄存器（0x03）

（单个寄存器地址内的数据为双字节型数据）

序号	名称	地址(16进制)	字数(U16)	范围	小数位	单位	读写	备注
1	输出电压	0x01	2	0-U _{max}	3	mV	R	
3	输出电流	0x03	2	0-I _{max}	3	mA	R	
5	输出功率	0x05	2		0	—	R	
7	预留	0x07	2					

2.2写入保持寄存器(0x10)

（单个寄存器地址内的数据为双字节型数据）

序号	名称	地址	字数(U16)	范围	小数点	单位	读写	备注
11	电压设定值	0x11	2	0-U _{max}	3	mV	R/W	
13	电流设定值	0x13	2	0-I _{max}	3	mA	R/W	
15	输出开关	0x15	2	0, 1	0	—	R/W	0开机；1 关机
17	远程	0x17	2	0, 1	0	—	R/W	0本地；1 远程

3实例

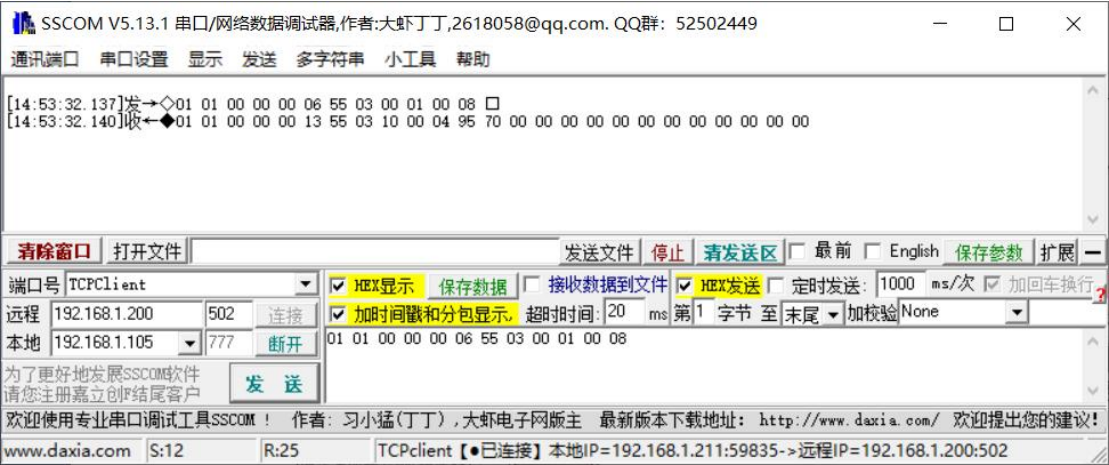
建议客户端连接

3.1控制流程

- 1、进入远程：写入远程模式，1
- 2、写入电压设置
- 3、写入电流设置
- 4、输出开关，0
- 5、读取输出电压电流状态
- 6、输出开关，1
- 7、退出远程：写入远程模式，0

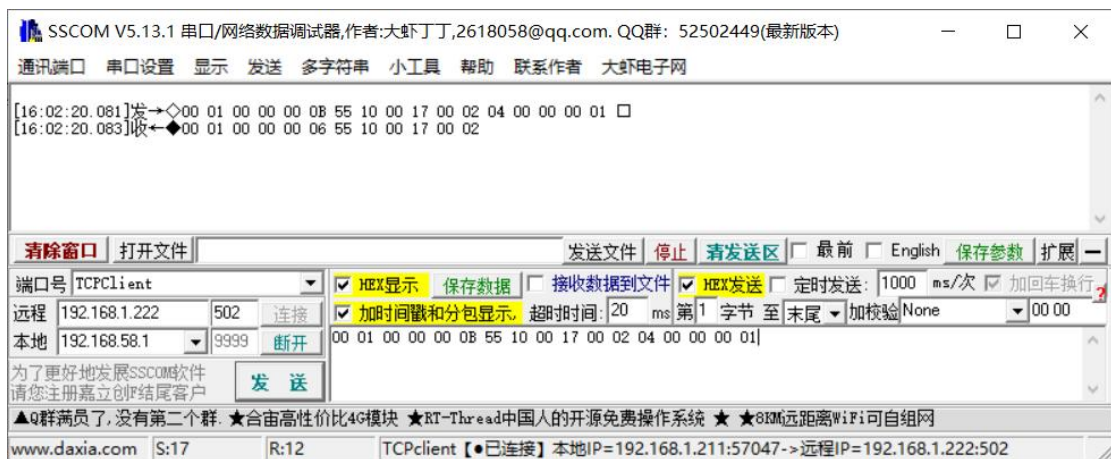
3.2读输入寄存器（0x03）

查询电压、电流、功率
发送01 01 00 00 00 06 55 03 00 01 00 08
接收0x00049570=300400mV=300.4V



3.3写保持寄存器（0x10）

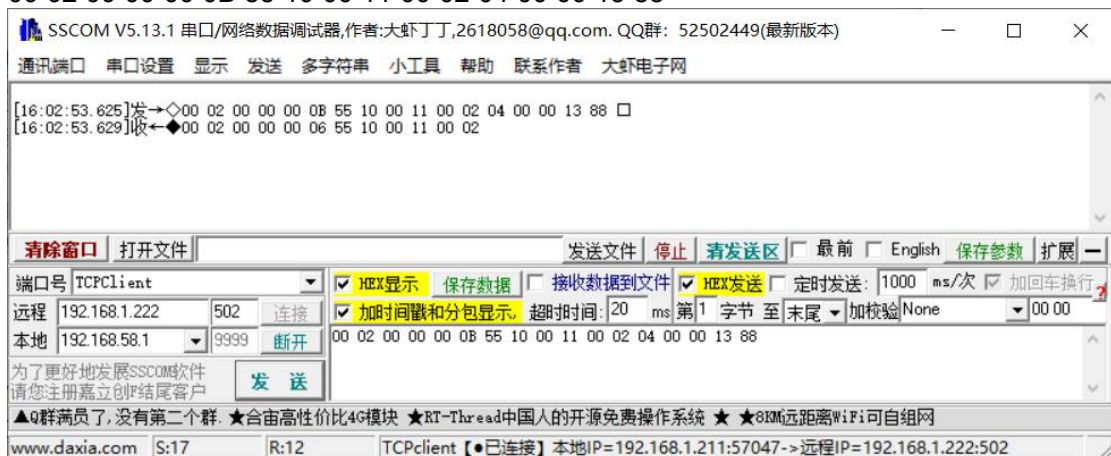
3.3.1进入远程
00 01 00 00 00 0B 55 10 00 17 00 02 04 00 00 00 01



3.3.2写电压5V

5V=5000mV=0x1388

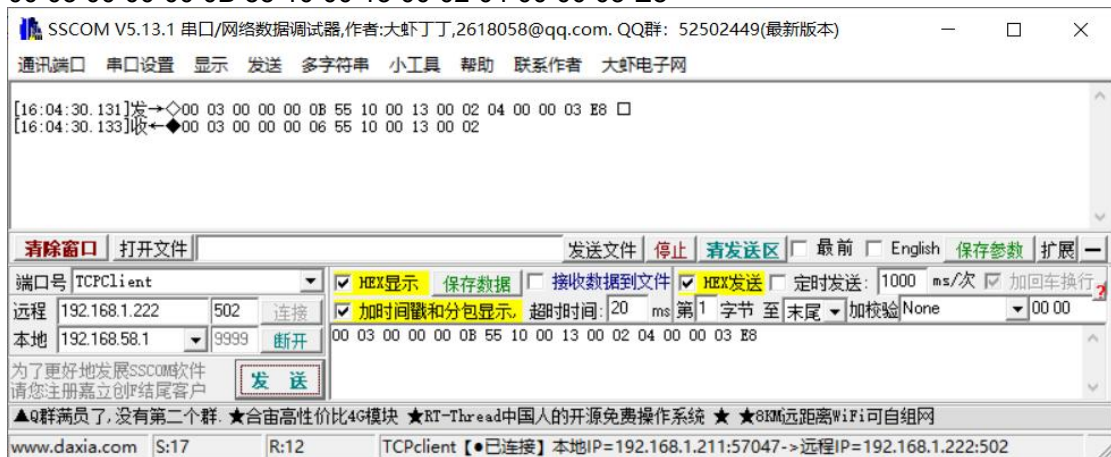
00 02 00 00 00 0B 55 10 00 11 00 02 04 00 00 13 88



3.3.3写电流1A

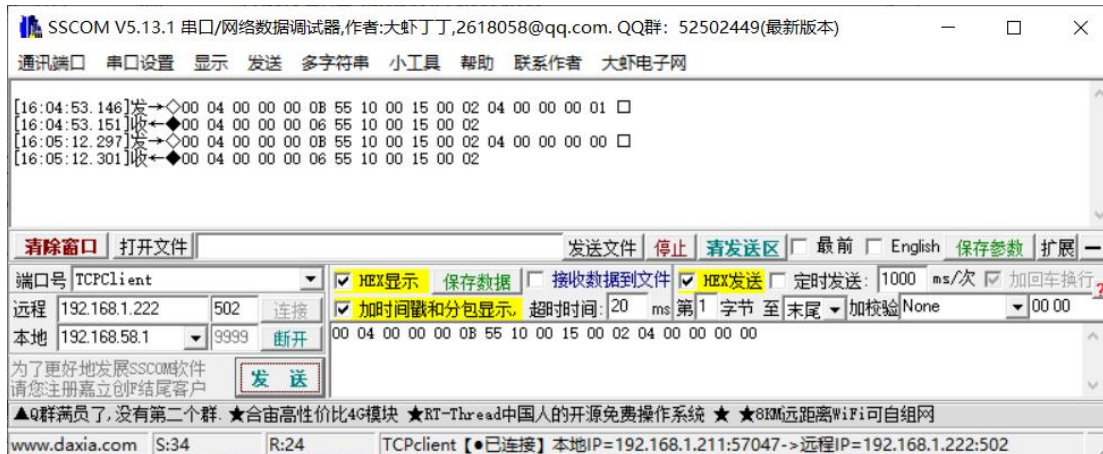
1A=1000mA=0x03E8

00 03 00 00 00 0B 55 10 00 13 00 02 04 00 00 03 E8



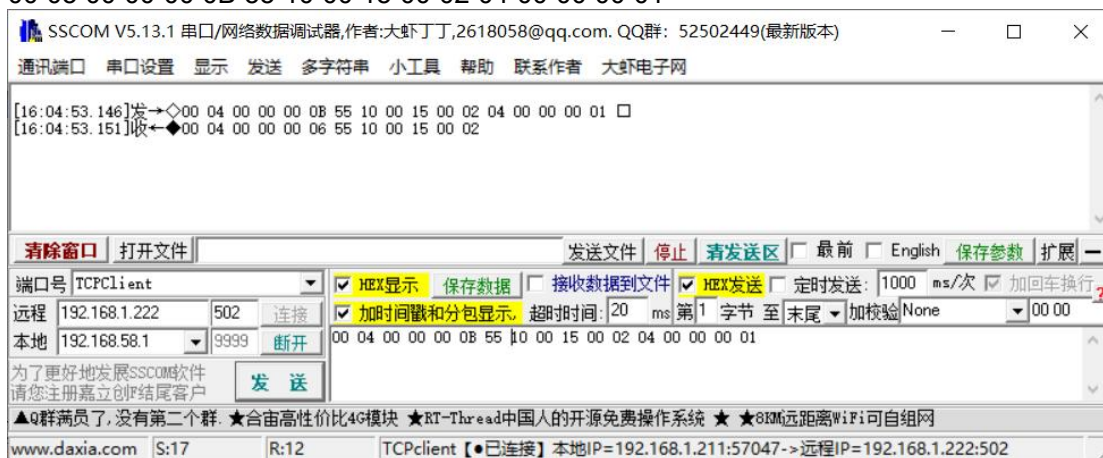
3.3.4写开机

00 04 00 00 00 0B 55 10 00 15 00 02 04 00 00 00 00



3.3.5写关机

00 05 00 00 00 0B 55 10 00 15 00 02 04 00 00 00 01



3.3.6写本地

00 06 00 00 00 0B 55 10 00 17 00 02 04 00 00 00 00

