



朝阳电流源



型号：4INC-CK-HL(X)

功能：，接电流表，通过手调输出直流电源，也可计算机编程进行控制。

输入电压 220V，50Hz

输出量程：5-90V，0-2.5A(直流)

更为详细的仪器说明和软件操作如下，实验室服务器上有程序代码：

4NIC-CK-2-10000w 程控电源

基本信息

产品名称： 4NIC-CK-2-10000w 程控电源

所属类别： AC/DC 系列

输入电压： 110/220/380VAC





2-10000W		
功能与特点	工作环境温度	电参数 (测试环境温度 25 °C)
<p>界面设有独特的软关机功能, 按关闭按钮, 电源输出为零, 再按打开按钮, 输出即恢复原来设定值。具有强大的记忆、存储功能, 掉电时, 数据不会丢失。还有查询、打印功能。方便试验和档案管理。手动与计算机控制互用, 即使计算机出现故障时, 电源仍能够正常工作。RS232 接口的设计, 方便与计算机进行通信, 通过计算机对程控电源进行各种操作, 可以近行远程电源控制、特制高亮度长寿命数码显示、PVC 面膜款式多样随您定、故障告警等。低纹波、进口高集成芯片为核心、设计裕量宽、100 % 漂移检测、满足世界各国标准 AC 输入要求。</p> <p>保护</p> <p>过流保护</p> <p>短路保护</p> <p>过压保护</p> <p>过热保护</p>	商业品: 0 °C +50 °C	输 入
	工业品: -25 °C +50 °C	电压范围——110/220/380VAC ±10%
	军用品: -40 °C +55 °C	单相 / 三相
	冷却形式: 自然 / 隧道风冷	频 率——50Hz/400Hz
	安全参数	效 率——75-85 % 典型 80 %
	隔离电压	输 出
	输入对输出: AC1000V/ 分钟 (漏电流 ≤ 10mA)	电 压——DC0-300V 任选
	输入对外壳: AC1500V/ 分钟 (漏电流 ≤ 10mA)	电 流——0-2A
	绝缘电阻	功 率——100W
	输入对外壳: DC1000V ≥ 200MΩ	电压精度——≤ 1 %
	输入对输出: DC1000V ≥ 200MΩ	电压调整率——≤ 0.5 %
	输出对外壳: DC250V ≥ 200MΩ	电流调整率——≤ 1%(5V ≤ 2%)
	输出对输出: DC250V ≥ 200MΩ	纹 波——Vrms ≤ 0.1mV(Uo < 48V) Vp-p ≤ 1mV(Uo < 48V)
	典型应用	过流保护——限流式
	模拟电路各类放大器	注: 附带 4NIC-CK 程控电源安装盘、数据线。
数字电路 工业仪表 交通运输 通讯设备 科研试验等	 <p>致力于创新的 不断追求</p>	

电源通讯协议 (V2.0)

1. 通讯参数

9600,N,8,1 物理接口 RS485



2. 电源控制帧定义

头	地址	控制命令高	3 字节数据域	校验	结束符
0X5E	0~255	参见下表	数据或填充字 (0XF0)	XOR	0X0D

3. 电源应答帧定义

头	地址	控制命令高	3 字节数据	校验	结束符
0X5E	0~255	参见下表	数据或电源状态字	XOR	0X0D

4. 控制命令

命令	说明	备注
0XA0	电源开	上位机的数据为填充字符 电源 MCU 返回的数据为状态字
0XA1	电源关	
0XA2	自检	
0XA3	升压	
0XA4	降压	
0XA5	升流	
0XA6	降流	



0XA7	写控制电压	上位机下传的是实际数
0XA8	写控制电流	据电源 MCU 返回原数据
0XA9	读电压	上位机的数据为填充字
0XAA	读电流	符
0XAB	读电源温度	电源 MCU 返回相应数据
0XAC	写预置输出电压	
0XAD	写预置输出电流	
0XAE	预置广播控制输出	广播命令(地址字段为 0), 数据域为三个字节 电源预置状态

5. 数据值定义

A. 非广播命令数据域

数据域部分 3 个字节的 BCD, 表示数据值. 小数点固定为 3 位。例如：
当发读电压值时，返回的数据为 025000，表示电压值为 25.000V。当
发读电流值时，返回数据为 010000，表示电流值为 10.000A。当发读
温度值时，返回的数据为 010000，则温度值为 10.000 度。0XA00000
表示无传感。

在写控制电压和控制电流时的数据值也遵循这一格式。

填充字:0XF0, 上位机命令不带数据时填充无意义的字符, 3 个字节的
0XF0.

B. 广播命令数据域



三个字节每位代表控制电源开关输出状态

位 0: 电源 1 通道控制状态。0-电源状态不变;1-电源预置输出

位 1: 电源 2 通道控制状态。0-电源状态不变;1-电源预置输出

...

位 23: 电源 24 通道控制状态。0-电源状态不变;1-电源预置输出

6. 校验 XOR

除头字段和结束符以外的所有数据异或值。

7. 故障代码

电源正常: 0X00009A;

过热保护: 0X00009B;

短路保护: 0X00009C;

过流保护: 0X00009D;

过压保护: 0X00009E;

关机状态: 0X00009F

8. 地址

电源地址。0 地址为广播地址。



函数功能如同函数名所示，结合程控电源的数据帧阅读本程序

```
void openpower()
{
    _data[0]=0x5E;
    _data[1]=0x04;
    _data[2]=0xA0;
    _data[3]=0x06;
    _data[4]=0x55;
    _data[5]=0x35;
    _data[6]=0xc2;
    _data[7]=0x0d;
    while(send_i<8)
    {
        if(SciaTx_Ready() == 1)
        {
            SciaRegs.SCITXBUF = _data[send_i];
            send_i++;
        }
    } //while
    send_i=0;
    delay_loop();
}

void closepower()
{
    _data[0]=0x5E;
    _data[1]=0x04;
    _data[2]=0xA1;
    _data[3]=0x06;
    _data[4]=0x55;
    _data[5]=0x35;
    _data[6]=(_data[1]^_data[2]^_data[3]^_data[4]^_data[5]);
    _data[7]=0x0d;
    while(send_i<8)
    {
        if(SciaTx_Ready() == 1)
        {
            SciaRegs.SCITXBUF = _data[send_i];
            send_i++;
        }
    } //while
    send_i=0;
    delay_loop();
}
```



```
void setdata()
{
    _data[0]=0x5E;
    _data[1]=0x04;
    _data[2]=0xA8;
    _data[3]=0x00;
    _data[4]=data_i[data_i_index];
    _data[5]=0x00;
    _data[6]=(_data[1]^_data[2]^_data[3]^_data[4]^_data[5]);
    _data[7]=0x0d;
    send_i=0;
    while(send_i<8)
    {
        if(SciaTx_Ready() == 1)
        {
            SciaRegs.SCITXBUF = _data[send_i];
            send_i++;
        }
    } //while
    send_i=0;
    delay_loop();
}

void setdatamax()
{
    _data[0]=0x5E;
    _data[1]=0x04;
    _data[2]=0xA7;
    _data[3]=0x09;
    _data[4]=0x00;
    _data[5]=0x00;
    _data[6]=0xAA;
    _data[7]=0x0d;
    while(send_i<8)
    {
        if(SciaTx_Ready() == 1)
        {
            SciaRegs.SCITXBUF = _data[send_i];
            send_i++;
        }
    } //while
    send_i=0;

    delay_loop();
    _data[3]=0x00;
```



```
_data[4]=0x25;  
_data[5]=0x00;  
_data[6]=0x89;  
    while (send_i<8)  
    {  
        if(SciaTx_Ready() == 1)  
        {  
            SciaRegs.SCITXBUF = _data[send_i];  
            send_i++;  
        }  
    } //while  
    send_i=0;  
  
    delay_loop();  
  
}
```