

DK04&DK04III通信协议

索引

DK04通信协议.....	03—14
DK04III补充通信协议.....	15—18

补充说明:

本文在原DK04通信协议的基础上增加了协议举例和DK04III补充通信协议。并对原DK04通信协议的错误进行了更正。
新增的部分用红字表示，请注意增加。

一、简介

本文仅介绍微机与DK04型监控模块之间的通信协议，有关DK04型监控模块和DK04型监控模块监控界面软件（CSUPC或WINCSU）的全面介绍请参考相应的用户手册。请使用本通信协议的人员仔细阅读本文的每一部分，并请认真阅读武汉洲际通信电源集团有限公司的电源使用手册。

本文只说明DK04型监控模块所运行的监控软件与对应的监控界面软件（CSUPC或WINCSU）之间的通信协议的要求，DK04型监控模块所适用的软件版本为3.0及以下，本通信协议的知识产权和本文的版权为武汉洲际通信电源集团有限公司所有，请使用本协议的单位和人员注意保密。

符号说明

在本文中，“PC”代表微机，“CSU”代表DK04型监控模块，“SMR”代表整流模块。

本协议的基本功能如下

建立和中断微机与监控模块之间的通信连接；
允许微机对监控模块传送一系列的特殊控制命令；
规定微机与监控模块之间数据交换的格式。

DK04型监控模块与远程监控PC的连接方式

DK04型监控模块与远程监控PC的连接方式有三种，这些连接方式在电源用户手册里有详细介绍，在本文里只作以下简单说明：

1. RS232方式

在DK04型监控模块的面板上有一个RS232标准的串行接口，通过此接口可以与本地的PC机通信，在本协议中称为RS232方式。

2. RS485方式

武汉洲际通信电源集团有限公司在向用户提供DK04型监控模块的同时，附带了一个称作“MIPS”（MODEM Interface and Power Supply）的接口模块。MIPS与监控模块之间的连接是RS485方式，MIPS上有RS232通信接口，可以直接接到计算机上，此种通信方式称为RS485方式。

3. MODEM方式

若将MIPS上的RS232接口接到MODEM上，那么DK04型监控模块可以通过电话网和远程监控计算机连接上，我们称之为MODEM方式。要注意的是，MODEM通信方式里，DK04型监控模块的参数项“MODEM启用”一定要设定为“1”，因为只有这样，CSU才可以对MODEM进行初始化操作。

注：在RS485和MODEM通信方式里，数据的传输格式是完全一样的，不同的是MODEM方式里要对MODEM进行操作。

通信的基本原则和数据格式

本协议所规定的通信方式为固定的数据格式和传输速率。数据格式为: 8位数据位加一位停止位, 无奇偶校验位, 以字符形式进行接收和发送, 以数据包和检查和的方式进行传输, 传输速率为 9600 bits/s, CSU以标准的AT命令对MODEM进行操作。通信所遵循的一条重要原则是“主从”通信方式, PC为“主”, CSU为“从”, CSU一般不主动向PC传送数据, 只有当CSU接收到PC的指令后才向PC传送相应的数据信息。但有一种情况列外, 即当CSU向PC回报(定时回报或故障回报)时, 首先由CSU向PC回报CSU的入口码, 并告之回报的类型, 这些过程完成后, 只有采用前面所述的“主从”通信方式PC才可以继续获得CSU的数据。

二、从PC发送到CSU的数据包的格式:

PC --> CSU

PC发送到CSU的数据包内的内容分别说明如下:

1. RS232方式

```
-----
| SyncChar | ByteCount | CommandID | Data | ChkSum |
| (AA hex) | (1 byte)   | (1 byte) | (n bytes) | (1 byte) |
-----
```

- SyncChar: 1 byte, CSU用来识别来自PC数据包的特定字符, 可称作“同步字符”, 其值必须是AA(hex)。
- ByteCount: 1 byte, 此数据包的字节数, 此字节数是从CommandID到ChkSum(包括它们本身)之间的字节个数。
- CommandID: 1 byte, 命令类型的识别字符, 可简称为“ID”, 具体意义在后面将有详细说明。
- Data: n bytes, 不同长度的数据流, 通常情况下, PC向CSU发送控制命令时, 数据包内不需要发送data, 在需要发送Data的数据包其对应的CommandID项上已做上“*”标记。
- ChkSum: 1 byte, 数据包的检查和。即将本数据包内从ByteCount到Data的最后一个字节依次相加, 所得结果即为ChkSum, 高位进位自然溢出丢失。

2. RS485和MODEM方式

在RS485和MODEM通信方式里, 数据包内的数据跟RS232通信方式相比, 增加了入口码(CSUAccessCode)项, 其余的和RS232方式完全相同。CSU本身的入口码在监控模块的面板上设置(其设定操作方法在用户手册上有介绍)。在此种通信方式里, 只有当PC向CSU发送的入口码和CSU本身的入口码一致时, CSU才响应PC发来的命令。

```
-----
| SyncChar | CSUAccessCode | ByteCount | CommandID | Data | ChkSum |
| (AA hex) | (3 bytes)    | (1byte)  | (1 byte) | (n bytes) | (1 byte) |
-----
```

CSUAccessCode: 3 bytes, 由七位十进制数组成的CSU入口码, 入口码在PC和CSU里以三个字节共24位的二进制数存储和传输, 通信时从低位字节到高位字节依次传送。在本协议中CSUAccessCode 都是这样定义的。

2005年6月2日

DK04&III型通信协议 4

注： ByteCount和ChkSum的定义与RS232方式里的定义是相同的, CSUAccessCode里的三个字节与其值的关系：

$$\text{CSUAccessCode} = \text{CSUAccessCode1} + \text{CSUAccessCode2} * 256 + \text{CSUAccessCode3} * 65536。$$

数据包内各种命令的描述（CommandID的各种取值的意义）

ReqSndStatID	100 = 要求CSU发送系统状态的数据包
ReqSndParamID	101 = 要求CSU发送监控模块参数的数据包
SetFloatID	102 = 设置CSU的浮充电压值 *
SetIBlimID	103 = 设置CSU的电池充电限流值 *
SetFSDAcI	104 = 设置CSU的交流电流传感器的满量程值 *
SetHhiID	105 = 设置CSU的系统输出电压过高告警值 *
SetVloID	106 = 设置CSU的系统输出电压过低告警值 *
SetBatTempID	107 = 设置CSU的电池温度过高告警值 *
SetEQperID	108 = 设置CSU的自动均充周期 *
SetEQdurID	109 = 设置CSU的自动均充持续时间 *
SetEQmanDurID	110 = 设置CSU的手动均充持续时间 *
SetEQtripID	111 = 设置CSU的自动启动均充的电压阈值 *
SetEQvID	112 = 设置CSU的均充电压值 *
SetLVDSID	113 = 设置CSU的LVDS（低压断路开关）自动断开的电压阈值 *
SetTempCoefID	114 = 设置CSU的电池充电的温度补偿系数 *
SetFSDiLID	115 = 设置CSU的负载电流传感器满量程值 *
SetFSDiBID	116 = 设置CSU的电池电流传感器满量程值 *
ReqSndBatEvID	117 = 要求CSU发送电池管理数据包
ReqSndLogID	118 = 要求CSU发送告警历史记录数据包
ReqSndSMRStatID	119 = 要求CSU发送所有整流模块的状态数据包
ReqSndSMRprmID	120 = 要求CSU发送某一整流模块的参数数据包 *
SetLastSMRaddr	121 = 设置整流模块个数 *
SetSMRparams	122 = 设置整流模块的参数 *
RstSMRHVSD	123 = 要求CSU对整流模块高压关机进行复位
ReqEQ	124 = 要求CSU进行均充
ReqFloat	125 = 要求CSU进行浮充
ReqSMRdisable	126 = 要求CSU关闭全部整流模块
ReqSMRenable	127 = 要求CSU开启全部整流模块
SetCa111	128 = 设置CSU的回叫电话号码1 *
SetCa112	129 = 设置CSU的回叫电话号码2 *
SetCa113	130 = 设置CSU的回叫电话号码3 *

SetModemEnable	131 = 允许或禁止启用MODEM *
SetCallOnError	132 = 允许或禁止启用故障回叫 *
SetCallDaily	133 = 允许或禁止启用定时回叫 *
SetDailyTime	134 = 设置CSU定时回叫的时间 *
GotPanic	135 = 向CSU证实已经得到CSU的故障回叫
GotDailyReport	136 = 向CSU证实已经得到CSU的定时回叫
SetCSUTime	137 = 设置CSU里的系统日期及时间 *

在标注“*”的CommandID后对应的Data数据的定义，与后面所讲的CSU传送到PC的监控模块参数数据包（ParamPacket）里相应数据项的定义是完全一样的。例如：

设置CSU的浮充电压值，SetFloatID的值是102，Data对应的内容是eeFloat，其数据类型是word，单位是100mV，取值范围是36V~65V（48V系统），18V~32V（24V系统）。

因为在本通信协议中，所要遵循的一重要原则是“主从”通信方式，所以若PC需要设置CSU的参数，可以这样处理，首先向CSU发送一参数设置命令，延时（大于5秒）后再读出CSU的参数，并与所要设定的值相比较，这样就可以判断参数设置是否成功。在PC向CSU传送控制命令时，也可以采取类似的方法，首先向CSU传送控制命令，延时（大于5秒）后再读出CSU或SMR的工作状态，这样就可以判断控制命令是否已成功执行。

重要建议： 因为CSU的参数对于整个电源系统的正常运行是至关重要的，所以在设定CSU的参数时一定要对其取值范围进行限制，首先要通过读CSU的状态数据包（StatusPacket）里的DIPval的值来确定是24V还是48V电源系统，然后确定相应的取值范围。在PC对CSU进行参数设置或PC向CSU传送控制命令时，PC程序里应加密码保护，以防止非法或未经授权的操作。

需要特别说明的是设置整流模块的参数的命令（SetSMRparams）对应的Data项的内容：

内容	字节数	数据类型	
NumofSMR	1	: byte	; 整流模块号码（255=所有整流模块）
FloatV,	2		; 浮充电压（100mV）
EQV,	2		; 均衡电压（100mV）
Vhi,	2		; 电压高告警（100mV）
Vlo,	2		; 电压低告警（100mV）
HVSD	2	: word	; 高压关机阈值电压（100mV）
Ilim	1	: byte	; 限流值（A）
AdjV	1	: ShortInt	; 均流调节值
SecEnable	1	: byte	; 保密功能开启（0表示不启用，1表示启用）

说明：若将整流模块的号码设置成255，这样就可以一次设置所有整流模块的参数，要注意的是，在设置所有整流模块的参数时，应将AdjV的值设定成120，因为对于每一个整流模块来说AdjV的值是不同的，当整流模块收到需要设定的AdjV值为120时，整流模块将不更改已有的AdjV值。

三、从CSU发送到PC的数据包的格式:

CSU -- > PC

CSU接收到PC要求发送数据包命令后，发送给PC的数据包内的数据格式:

1. RS232方式

```
-----
| SyncChar | PacketID | Data | ChkSum |
| (AA hex) | (1 byte) | (n bytes) | (1 byte) |
-----
```

SyncChar : 1 byte, PC用来识别来自CSU数据的特定字符, 其值必须为AA(hex)。

PacketID : 1 byte, 表示数据包内内容的特征代码, 可为下列种类,:

StatBlkID	20 = (CSU所发送的数据为系统状态的数据包)
ParamBlkID	21 = (CSU所发送的数据为监控模块参数的数据包)
BatEvBlkID	22 = (CSU所发送的数据为电池管理的数据包)
LogBlkID	23 = (CSU所发送的数据为告警历史记录的数据包)
SMRstatBlkID	24 = (CSU所发送的数据为整流模块状态的数据包)
SMRparamBlkID	25 = (CSU所发送的数据为某一整流模块参数的数据包)

Data : n bytes, 不同长度的数据流, 其内容根据不同的PacketID而各不相同, 对于CSU发送到PC的数据包里, Data均有其对应的内容, 其具体的定义在后面的章节有详细说明。

ChkSum : 1 byte, 检查和, 即将数据包内从同步字符SyncChar到Data内的最后的一个字节依次相加, 所得结果即为ChkSum, 高位进位自然溢出丢失。

2. RS485和MODEM方式

在RS485和MODEM通信方式里, 数据包内的数据跟RS232通信方式相比, 增加了入口码(CSUAccessCode)项, 其余的和RS232方式完全相同。CSU的入口码在监控模块的面板上设置(其设定操作方法在用户手册上有介绍)。

```
-----
| SyncChar | CSUAccessCode | PacketID | Data | ChkSum |
| (AA hex) | (3 bytes) | (1 byte) | (XX byte) | (1 byte) |
-----
```

在RS485和MODEM方式里, CSU可以主动向PC回报, 其数据包内的数据格式:

```
-----
| SyncChar | CallupID | CSUAccessCode | ChkSum |
| (AA hex) | (1 bytes) | (3 byte) | (1 byte) |
-----
```

CallupID: 表示CSU回报种类的特征代码, PC依照此字节判断CSU回报的种类。

PanicID 30 = 故障回叫

DailyReportID 31 = 定时回叫

CSUAccessCode: 3 bytes, 因为每一个CSU的入口码都可以设为不同, 所以PC可依照此入口码判断具体是哪一个CSU发来的回报。入口码由七位十进制数组成的, 在PC和CSU里以三个字节共24位的二进制数存储和传输, 通信时从低位字节到高位字节依次传送。

PC如何响应CSU的回报

当PC收到CSU的回报信息后, 应首先根据CallupID判断出回报的类型(故障回报或定时回报), 然后根据CSU传送过来的入口码(CSUAccessCode)判断具体是哪一个CSU发来的回报。这些过程完成后, PC应向CSU发送请求发送各种状态及参数数据包命令, 当PC知道了CSU的整个工作状态后, 应向CSU发送已收到故障回报或定时回报的证实信息(GotPanic或GotDailyReport), 因为CSU在收到PC发来已收到回报的证实之前认为PC没有收到回报, 并且继续向PC发送回报信息。最后PC应中断与CSU的通信关系。

各数据包数据(Datas)描述

各种数据类型的定义:

byte : 8位无符号数, 范围0~255

word : 16位无符号数, 范围0~65535, 低位字节先发送

ShortInt : 8位有符号数, 范围-128~127

SmallInt : 16位有符号数, 范围-32768~32767

在本协议中各项数据都是以定点数传送, 所以一定要注意各项数据的单位。例:

在StatusPacket数据包内, 系统电压(SysV)是一个word型的数据, 其单位是100mV, 所以假设传送到PC的数据是545, 那么, 实际的SysV是54.5V。

1. 系统状态数据包(StatusPacket):

内容	字节数	数据类型	
SysV,	2		; 系统电压(100mV)
ILoad,	2	: word	; 负载总电流(A)
IBat1,	2		; 电池组1的电流(A)
IBat2,	2		; 电池组2的电流(A)
IBat3,	2		; 电池组3的电流(A)
IBat4,	2	: SmallInt	; 电池组4的电流(A) (更正: 原为word)
VAcSys,	2		; 系统交流输入电压(V)
I1AcSys,	2		; 系统交流输入电流1(A)
I2AcSys	2	: word	; 系统交流输入电流2(A)
MD01val,	1		; 告警及状态代码1
MD02val,	1		; 告警及状态代码2
MD03val	1	: byte	; 告警及状态代码3

Btemp1,	1		; 电池组1温度 (°C)
Btemp2,	1		; 电池组2温度 (°C)
Btemp3,	1		; 电池组3温度 (°C)
Btemp4	1	: ShortInt	; 电池组4温度 (°C)
PortAval,	1		; 告警及状态代码
PortBval,	1		; 告警及状态代码
PortCval,	1		; 告警及状态代码
DIPval,	1		; 告警及状态代码
KB5val	1	: byte	; 告警及状态代码
Atemp	1	: ShortInt	; 环境温度 (°C)
HallAlarm	1	: byte	; 霍尔告警及状态代码 (#)
Spare	: array[1..4]	of byte	; 空 (未定义)
ChkSum	1	: byte	; 检查和

数据说明:

Btemp1, Btemp2, Btemp3, Btemp4和Atemp的有效范围是: -10~90°C, 若温度超出此范围应认为是温度传感器未安装。

MD01val的每一位所对应的意义, 当某一位为0时, 表示对应的告警或状态存在。

- Bit0 电池温度补偿功能启用
- Bit1 电池放电
- Bit4 整流模块故障 (整流模块关机)
- Bit5 整流模块告警 (整流模块没有关机的告警)
- Bit6 交流输入故障
- Bit7 监控模块的告警蜂鸣器的状态 1: 叫

MD02val的每一位所对应的意义, 当某一位为0时, 表示对应的告警或状态存在。

- Bit1 均衡充电状态
- Bit2 浮充状态
- Bit3 电池充电限流
- Bit4 电池温度过高
- Bit5 熔丝故障
- Bit6 电池开关断开
- Bit7 环境温度过高

MD03val的每一位所对应的意义, 当某一位为0时, 表示对应的告警或状态存在。

- Bit0 低压断路器 (LVDS) 闭合 (更正: 原为 '断开')
- Bit1 低压断路器 (LVDS) 断开 (更正: 原为 '闭合')
- Bit2 系统输出电压过低
- Bit3 系统输出电压过高

DIPval, CSU里功能设置开关的状态

Bit0	均衡充电功能	0: 启用	1: 不启用
Bit1	电池温度补偿功能	0: 启用	1: 不启用
Bit2	电池充电限流功能	0: 启用	1: 不启用
Bit3	操作CSU时的密码功能	0: 启用	1: 不启用
Bit4	系统接地方式	0: 正接地	1: 负接地
Bit5	系统的类型	0: 48V系统	1: 24V系统
Bit6	低压断路 (LVDS) 开关自动闭合功能	0: 启用	1: 不启用

KB5val

Bit2	均衡充电控制方式	0: 自动	1: 手动
------	----------	-------	-------

PortAval的每一位所对应的意义, 当某一位为1时, 表示对应的告警或状态存在。

Bit0	均衡充电
Bit1	环境温度过高
Bit2	整流模块主要告警 (两个或两个以上的模块关机)
Bit3	整流模块次要告警 (一个模块关机)
Bit4	交流输入故障
Bit5	系统输出电压过高
Bit6	系统输出电压过低

PortCval的每一位所对应的意义, 当某一位为1时, 表示对应的告警或状态存在。

Bit4	整流模块警告 (没有导致关机的告警)
Bit7	整流模块高压关机

HallAlarm的每一位所对应的意义, 当某一位为0时, 表示对应的告警或状态存在。

Bit0	霍尔1告警
Bit1	霍尔2告警
Bit2	霍尔3告警
Bit3	霍尔4告警
Bit4-Bit7	: 暂时保留未用

2. 监控模块参数数据包 (ParamPacket)

内容	字节数	数据类型
EeFloat,	2	; 浮充电压 (100mV)
eeIBlim,	2	; 电池充电限流 (A)
eeFSDiA,	2	; 交流电流传感器满量程 (A)
eeVhi,	2	; 系统输出电压过高告警 (100mV)
eeVlo,	2	; 系统输出电压过低告警 (100mV)
eeBatTemp,	2	; 电池温度高告警 (°C)
eeEQper,	2	; 均充周期 (星期)
eeEQdur,	2	; 均充持续时间 (小时)
eeEQmanDur,	2	; 手动均充持续时间 (小时)

eeEQtrip,	2		; 自动启动均充的电压阈值 (100mV)
eeEQv,	2		; 均衡电压 (100mV)
eeLVDS,	2		; 低压断路器自动断开的电压阈值 (100mV)
eeBTC,	2		; 电池温度补偿系数 (0.1mV/°C/每节电池)
eeFSDiL,	2		; 负载电流传感器满量程 (A)
eeFSDiB,	2		; 电池电流传感器满量程 (A)
eeLastSMR	2	: word	; 整流模块个数
eeCallNum1,	20	ASCII码: array[1..20] of char;	回报电话号码 1
eeCallNum2,	20	ASCII码: array[1..20] of char;	回报电话号码 2
eeCallNum3	20	ASCII码: array[1..20] of char;	回报电话号码 3
eeModemEnable,	1		; Modem启用? (0表示不启用, 1表示启用)
eeCallOnError,	1		; 故障回报启用? (0表示不启用, 1表示启用)
eeCallDaily,	1		; 定时回报启用? (0表示不启用, 1表示启用)
eeDailyMin,	1		; 定时回报时间的分钟值
eeDailyHour	1	: byte	; 定时回报时间的小时值
CSUTime	6	: TCSUtime	; 监控模块的日期与时钟
ChkSum	1	: byte	; 检查和

数据说明:

eeCallNum1, eeCallNum2, eeCallNum3这三个数据的类型是完全一样的, 都是由20个用ASCII码表示的字符组成的字符串, 在PC和CSU里和传输过程中也是用ASCII码表示的。此类数据的长度为固定的20个字节, 若电话号码的长度不足20位, 后面剩余的字节用空格的ASCII码填充, 在设置回报电话号码的方法中也是用此类方法处理。

TCSUtime的定义

内容	字节数	数据类型
Day,	1	
Month,	1	
Year,	1	
Cent,	1	
Hour,	1	
Min,	1	
Sec	1	: byte

3. 电池管理数据包 (BatEvPacket)

内容	字节数	数据类型	
eeBElastDur	2	: word	; 放电持续时间 (小时)
eeBElastRTim	6	: TCSUtime	; 最后一次放电时刻
eeBEavI1,	2		; 电池组1平均放电电流 (A)
eeBEavI2,	2		; 电池组2平均放电电流 (A)
eeBEavI3,	2		; 电池组3平均放电电流 (A)

eeBEavI4,	2		; 电池组4平均放电电流(A)
eeBEcumTim1,	2		; 电池组1累计放电时间(小时)
eeBEcumTim2,	2		; 电池组2累计放电时间(小时)
eeBEcumTim3,	2		; 电池组3累计放电时间(小时)
eeBEcumTim4,	2		; 电池组4累计放电时间(小时)
eeBEnumDis1,	2		; 电池组1累计放电次数
eeBEnumDis2,	2		; 电池组2累计放电次数
eeBEnumDis3,	2		; 电池组3累计放电次数
eeBEnumDis4,	2		; 电池组4累计放电次数
eeBEendV	2	: word	; 放电终止电压(V)
eeBElastEQTim	6	: TCSUtime	; 最后一次均充时刻
eeBEnumEQ	2	: word	; 均充次数
ChkSum	1	: byte	; 检查和

4. 告警历史记录数据包(LogPacket)

内容	字节数	数据类型	
LogSize	1	: byte	; 告警历史记录长度
Log	: array[1..100] of LogItem		; 各告警历史记录项的内容
ChkSum	1	: byte	; 检查和

数据说明:

LogItem (每一项告警历史记录项的内容)

内容	字节数	数据类型	
AlarmCode,	1		; 故障编码
Day,	1		; 发生故障日
Month,	1		; 发生故障月
Year,	1		; 发生故障年
Cent,	1		; 发生故障世纪
Hour,	1		; 发生故障时
Min,	1		; 发生故障分
Sec	1	: byte	; 发生故障秒

AlarmCode的定义

- 00: 整流模块告警
- 01: 交流输入故障
- 02: 熔丝故障
- 03: 电池开关断开
- 04: 环境温度过高
- 05: 电池温度过高
- 06: 空

- 07: 系统输出电压过高
- 08: 系统输出电压过低
- 09: 电池充电限流
- 10: 低压断路开关 (LVDS) 断开
- 11: 恢复正常
- 12: 整流模块通信故障

说明: 告警历史记录数据包内的字节数是固定的, 一共可以记录100条告警记录信息, 当告警记录的长度不足100条 (LogSize<100) 时, 剩余的告警记录项是用随机数来填充的, ChkSum将这些随机数同样计算在内。

5. 整流模块状态数据包 (SMRstatPacket)

内容	字节数	数据类型	
NumSMRs	1	: byte	; 整流模块个数
Status		: array[1..110] of SMRstatItem	; 各整流模块状态
ChkSum	1	: byte	; 检查和

数据说明:

SMRstatItem (每一个整流模块状态的内容)

内容	字节数	数据类型	
Alarm1,	1		; 故障编码1
Alarm2,	1		; 故障编码2
Alarm3,	1		; 故障编码3
Iout	1	: byte	; 输出电流 (A)

当某一位为1时, 表示存在其对应的告警。

Alarm1:

- Bit0 输出电压过高
- Bit1 输出电压过低
- Bit2 模块地址为0
- Bit3 输入开关未合
- Bit4 内部高压直流回路故障
- Bit5 电压回路控制放大器超出正常工作范围
- Bit6 内部温度高
- Bit7 内部控制回路调节电压故障

Alarm2:

- Bit0 输出限流
- Bit1 风扇故障
- Bit2 输出欠流
- Bit3 输入开关跳闸
- Bit4 手动关机
- Bit5 遥控关机
- Bit6 参考电压超出范围
- Bit7 通信故障

Alarm3:

- Bit0 高压关机
- Bit1 交流故障
- Bit2 功率限制
- Bit5 均衡充电
- Bit6 故障并关机
- Bit7 告警未关机

说明： 整流模块工作状态数据包内，可以表示110个模块的工作状态，当整流模块个数小于110时（NumSMRs<110），剩余的数据是用随机数来填充的。同样ChkSum同等对待这些随机数据。

6. 整流模块参数数据包（SMRparamPacket）

内容	字节数	数据类型	
FloatV,	2		; 浮充电压（10mV）
EQV,	2		; 均衡电压（10mV）
Vhi,	2		; 电压高告警（10mV）
Vlo,	2		; 电压低告警（10mV）
HVSD	2	: word	; 高压关机阈值电压（10mV）
Ilim	1	: byte	; 限流值（A）
AdjV	1	: ShortInt	; 均流调节值
SecEnable	1	: byte	; 保密功能开启（0表示不启用，1表示启用）
ChkSum	1	: byte	; 检查和

说明： 因为PC向CSU请求发送整流模块参数时，首先要给定一个整流模块的号码。这样，PC请求发送哪个模块的参数，那么随后收到的就是这个整流模块的参数数据包。这时，PC发送给CSU的请求命令里的 data 项是一个模块号码，具体格式如下：

```
-----
|SyncChar| ByteCount| ReqSendSMRprmID | NumberofSMR | ChkSum |
| (AA hex)| (1 byte) | (1 byte) | (1 byte) | (1 byte)|
-----
```

注: NumberofSMR的数据类型是byte

协议举例

1. 请求CSU发送系统状态的数据包

PC向CSU发送的数据包为: AA, 02, 64, 66

2. 设置CSU的浮充电压值为54.0V

分析: 协议中浮充电压的单位为0.1V, 故为540 (对应16进制为21C), 注意先送低字节, 后送高字节

PC向CSU发送的数据包为: AA, 04, 66, 1C, 02, 88

DK04III补充协议

Stat2B1kID	EQU	26	;系统状态2包
Param2B1kID	EQU	27	;交流参数数据包ID
StatACB1kID	EQU	28	;交流状态数据包ID
ReqSndStat2ID	EQU	176	;请求发送系统状态2包
ReqSndACStat2ID	EQU	179	;请求发送交流状态数据包ID
ReqSndParam2ID	EQU	177	;请求发送交流参数ID
SetParam2ID	EQU	178	;设定交流参数ID
LastID	EQU	180	

Stat2B1kID ;系统状态2包
 ILD1, ILD2, ILD3, ILD4, ILD5, ILD6, ILD7, ILD8,
 ILD9, ILD10, ILD11, ILD12, ILD13, ILD14, ILD15,
 ILD16, ILD17, ILD18, ILD19, ILD20 : SmallInt;
 ILD21, ILD22 : SmallInt;
 Kong1, Kong2 : word;
 XKD1 (1 byte)

ILD1-- ILD20 :代表负载分路1-20电流

ILD21, ILD22, Kong1, Kong2 :未用

XKD1的定义

- Bit0 ; 熔断器 (0表示正常)
- Bit1 ; 电池开关 (0表示闭合)
- Bit2 ; 防雷模块 (0表示正常)
- Bit3 ; 门禁 (0表示正常)
- Bit4 ; 烟雾 (0表示正常)
- Bit5 ; 油机 (0表示正常)
- Bit6 ; 空调 (0表示正常)
- Bit7 ; 地水 (0表示正常)

; *****

Param2B1kID ;交流参数数据包
 eeFSDiL1 (word), ; 直流负载分路满量程
 eeLDNUM (1 byte), ; 直流负载分路数

eeACNUM(1 byte), ; 三相交流采样板个数
 eeACVHI3(word), ; 三相交流电压过高告警
 eeACVLO3(word), ; 三相交流电压过低告警
 eeACFHI3(word), ; 三相交流频率过高告警
 eeACFLO3(word), ; 三相交流频率过低告警
 eeACIHI3(word), ; 三相交流电流过大告警
 eeBATNUM(1 byte), ; 电池组数
 eeBATDIS(word), ; 电池放电测试终止电压
 eeBATRATE(word), ; 电池组额定容量
 nil(2 bytes), nil(2 bytes), nil(2 bytes), nil(2 bytes), nil(2 bytes)

; *****

StatACBlkID ; 交流状态数据包

ACV1(word), ; 第一路三相交流输入U相电压
 ACV2(word), ; 第一路三相交流输入V相电压
 ACV3(word), ; 第一路三相交流输入W相电压
 ACI1(word), ; 三相交流输入U相电流
 ACI2(word), ; 三相交流输入V相电流
 ACI3(word), ; 三相交流输入W相电流
 ACFHZ(word), ; 第一路三相交流输入频率
 ACV21(word), ; 第二路三相交流输入U相电压
 ACV22(word), ; 第二路三相交流输入U相电压
 ACV23(word), ; 第二路三相交流输入U相电压
 ACFHZ2(word), ; 第二路三相交流输入频率
 XKD(1 byte),
 XKA1(1 byte),
 XKA2(1 byte),
 XKB1(1 byte),
 XKB2(1 byte),
 nil(2 bytes), nil(2 bytes), nil(2 bytes), nil(2 bytes), nil(2 bytes)

XKD ; 交流负载分路空气开关状态及交流屏给出的告警状态

Bit0 ; 交流屏输入空气开关状态(0表示断开)
 Bit1 ; 交流屏给出的交流缺相、电压过高或过低告警(0表示告警)
 Bit2 ; 交流屏输出空气开关6状态(0表示断开)
 Bit3 ; 交流屏输出空气开关5状态(0表示断开)
 Bit4 ; 交流屏输出空气开关4状态(0表示断开)
 Bit5 ; 交流屏输出空气开关3状态(0表示断开)
 Bit6 ; 交流屏输出空气开关2状态(0表示断开)
 Bit7 ; 交流屏输出空气开关1状态(0表示断开)

XKA1 ; 第一路三相交流告警标志1

- Bit0 ; U相电压过低 (1表示告警, 0表示正常)
- Bit1 ;
- Bit2 ; U相电压过高 (1表示告警, 0表示正常)
- Bit3 ;
- Bit4 ; V相电压过低 (1表示告警, 0表示正常)
- Bit5 ;
- Bit6 ; V相电压过高 (1表示告警, 0表示正常)
- Bit7 ;

XKA2 ; 第一路三相交流告警标志2

- Bit0 ; W相电压过低 (1表示告警, 0表示正常)
- Bit1 ;
- Bit2 ; W相电压过高 (1表示告警, 0表示正常)
- Bit3 ; 交流输入频率过低 (1表示告警, 0表示正常)
- Bit4 ; 交流输入频率过高 (1表示告警, 0表示正常)
- Bit5 ; 交流输入电流U相过大 (1表示告警, 0表示正常)
- Bit6 ; 交流输入电流V相过大 (1表示告警, 0表示正常)
- Bit7 ; 交流输入电流W相过大 (1表示告警, 0表示正常)

XKB1 ; 第二路三相交流告警标志1

- Bit0 ; U相电压过低 (1表示告警, 0表示正常)
- Bit1 ;
- Bit2 ; U相电压过高 (1表示告警, 0表示正常)
- Bit3 ;
- Bit4 ; V相电压过低 (1表示告警, 0表示正常)
- Bit5 ;
- Bit6 ; V相电压过高 (1表示告警, 0表示正常)
- Bit7 ;

XKB2 ; 第二路三相交流告警标志2

- Bit0 ; W相电压过低 (1表示告警, 0表示正常)
- Bit1 ;
- Bit2 ; W相电压过高 (1表示告警, 0表示正常)
- Bit3 ; 交流输入频率过低 (1表示告警, 0表示正常)
- Bit4 ; 交流输入频率过高 (1表示告警, 0表示正常)
- Bit5 ;
- Bit6 ;
- Bit7 ;

; *****

告警记录代码含义:

code = 18:	; 地水告警
24:	; 烟雾
25:	; 门禁
26:	; 防雷模块告警
27:	; 油机故障
28:	; 三相交流电压U相过高
29:	; 三相交流电压V相过高
30:	; 三相交流电压W相过高
31:	; 三相交流电压U相过低
32:	; 三相交流电压V相过低
33:	; 三相交流电压W相过低
34:	; 三相交流输入缺相、电压过高或过低
35:	; 空调停机
36:	; 三相交流频率过高
37:	; 三相交流频率过低
38:	; 三相交流输入U相电流过大
39:	; 三相交流输入V相电流过大
40:	; 三相交流输入W相电流过大
41:	; 霍尔1告警
42:	; 霍尔2告警
43:	; 霍尔3告警
44:	; 霍尔4告警

注: AlarmCode=13~17和19~23未定义