

104 规约(2002版)报文解析

1、初始化

主站发:68 04 07 00 00 00

目的:给子站发请求链路状态命令。

子站回答: 68 04 0B 00 00 00 目的:子站向主站响应链路状态。

子站回答: 68 0E 00 00 00 00 46 01 04 00 01 00 00 00 00 00

目的:初始化结束。

2、对时

时钟同步命令一般不在 104 中应用,因为网络路由的延时永远不定(随机) ,导致对时不准。

主站发: 68 14 2C 00 6A 00 67 01 06 00 01 00 00 00 00 E5 3F 00 0F 09 0C 04目的:向子站发送对时报文。 357 毫秒 16 秒 0 分 15 小时 9 日 12 月 4 年

3、总召唤

主站发: 68 0E 00 00 06 00 64 01 06 00 01 00 00 00 00 14

目的:向地址为 01 的子站发总召唤命令。

子站回答: 68 0E 08 00 02 00 64 01 07 00 01 00 00 00 00 14

目的:子站响应总召唤。

子站回答: 68 2D 0A 00 02 00 01 A0 14 00 01 00 01 00 00 01 00 01 00 00 00

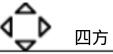
目的:子站向主站以 ASDU1方式连续上送全遥信,此为第一帧。

报文解析:

| 68 2D 0A | |
|----------|--|
| 00 02 00 | |
| 01 | 数据类型(ASDU方式) |
| A0 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,20 转成十进制 =32) |
| 14 00 | 传送原因 |
| 01 00 | 子站地址 |
| 01 00 00 | 起始点号 |
| 00 | 点号 1的遥信状态(分) |
| 01 | 点号 2的遥信状态(合) |
| 00 | 点号 3的遥信状态(分) |
| | |
| 00 | 点号 32的遥信状态(分) |

子站回答: 68 2D 0C 00 02 00 01 A0 14 00 01 00 21 00 00 00 00 00 00 00 00 00 00

目的:子站继续上送全遥信的下一帧。



.....

子站回答: 68 2D 20 00 02 00 01 A0 14 00 01 00 61 01 00 00 00 00 00 00 00 00 00

目的:子站向主站上送全遥信的最后一帧。

子站回答: 68 AD 22 00 02 00 0D A0 14 00 01 00 01 40 00 B2 0B 4B 42 00 00 00 00

目的: 子站向主站以 ASDU13方式(浮点数) 上送全遥测中的第一帧。 (有些调度要求用 ASDU9(码值)上送,见下文补充)

报文解析:

| 3100 7 7 9 1 1 7 1 | |
|--------------------|--|
| 68 AD | |
| 22 00 | |
| 02 00 | |
| 0D | 数据类型(ASDU方式) |
| A0 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,20 转成十进制 =32) |
| 14 00 | 传送原因 |
| 01 00 | 子站地址 |
| 01 40 00 | 起始点号(16385) |
| B2 0B | 点号 16385的遥测值 |
| 4B 42 | |
| 00 | |
| 00 00 | 点号 16386的遥测值 |
| 00 00 | |
| 00 | 00 |
| | |
| | |

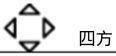
目的:子站继续上送全遥测的下一帧。

.

子站回答: 68 17 32 00 02 00 0D 82 14 00 01 00 01 41 00 00 00 00 00 00 00 00 00

00 00

目的:子站向主站上送全遥测的最后一帧。



子站回答: 68 0E 34 00 02 00 64 01 0A 00 01 00 00 00 00 14

目的:子站以 ASDU100响应主站,指明总查询结束。

补充:有些主站要求遥测用 ASDU9方式上送,举例报文如下

子站回答: 68 6D 9E 00 06 00 09 A0 14 00 02 00 01 40 00 ED 01 00 76 01 00 00 00

00 00 00 00 00 00 00 00 00

目的:子站向主站以 ASDU9方式(码值)上送全遥测中的第一帧。

报文解析:

| 68 6D | |
|----------|--|
| 9E 00 | |
| 06 00 | |
| 09 | 数据类型(ASDU方式) |
| A0 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,20 转成十进制 =32) |
| 14 00 | 传送原因 |
| 02 00 | 子站地址 |
| 01 40 00 | 起始点号(16385) |
| ED 01 | 点号 16385的遥测值 |
| 00 | |
| 76 01 | 点号 16386的遥测值 |
| 00 | 00 |
| | |

4、 变化遥测

子站回答: 68 1A 36 00 02 00 0D 02 03 00 01 00 01 40 00 22 5D 87 41 00 A4 40 00

22 5D 87 41 00

目的:有变化数据并以 ASDU13方式上送变化遥测

报文解析:



| 68 1A 36 | |
|----------|--|
| 00 02 00 | |
| 0D | 数据类型(ASDU方式) |
| 02 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,02 转成十进制 =02) |
| 03 00 | 传送原因 |
| 01 00 | 子站地址 |
| 01 40 00 | 点号 (16385) |
| 22 5D | 点号 16385的遥测值 |
| 87 41 | |
| 00 | |
| A4 40 00 | 点号 (16548) |
| 22 5D | 点号 16548的遥测值 |
| 87 41 | |
| 00 | 00 |

子站回答: 68 16 B0 00 04 00 09 02 03 00 02 00 01 41 00 E9 02 00 02 41 00 29 02

00

目的:有变化数据并以 ASDU9方式上送变化遥测

报文解析:

| 68 16 B0 | |
|----------|--|
| 00 04 00 | |
| 09 | 数据类型(ASDU方式) |
| 02 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,02 转成十进制 =02) |
| 03 00 | 传送原因 |
| 02 00 | 子站地址 |
| 01 41 00 | 点号(16641) |
| E9 02 | 点号 16641的遥测值 02 E9 转成十进制 = 745 |
| 00 | |
| 02 41 00 | 点号(16642) |
| 29 02 | 点号 16642的遥测值 |
| 00 | 00 |

主站发 68 04 43 00 00 00

目的:召唤变化数据。

子站回答: 68 04 83 00 00 00

目的:此刻没有变化数据。

5、 变化遥信

当站内有相应遥信点变位时

WATCHBL 给提示 Get MSG_LON_YX_CHANGE Index:1 Val:0

Get MSG_LON_YX_CHANGE Index:2 Val:1 Get MSG_LON_YX_CHANGE Index:3 Val:0

子站回答: 68 16 32 00 02 00 01 03 03 00 01 00 02 00 00 00 03 00 00 01 04 00 00

00

目的:以 ASDU1方式上送变位遥信(2号点置 0,3号点置 0,4号点置 0,)。

报文解析:

| 68 16 32 | |
|----------|--|
| 00 02 00 | |
| 01 | 数据类型(ASDU方式) |
| 03 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,03 转成十进制 =03) |
| 03 00 | 传送原因 |
| 01 00 | 子站地址 |
| 02 00 00 | 点号(02) |
| 00 | 点号 02的遥信状态,为分 |
| 03 00 00 | 点号(03) |
| 01 | 点号 03的遥信状态,为合 |
| 04 00 00 | 点号(04) |
| 00 | 点号 04的遥信状态,为分 |

.

WATCHBL Get MSG_LON_YX_CHANGE Index:2 Val:0

子站回答: 68 0E 36 00 02 00 01 01 03 00 01 00 03 00 00 00

目的:以 ASDU1方式上送变位遥信(3号点置 0)。

6、SO的处理

分为 3 字节时标和 7 字节时标两种:

子站回答: 68 11 24 00 2A 00 02 01 01 00 01 00 01 00 00 00 7E 3E 18

目的:以 ASDU2方式上送 SOE(3字节时标)。

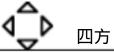
报文解析:

| 68 11 24 | |
|----------|--|
| 00 2A 00 | |
| 02 | 数据类型(ASDU方式) |
| 01 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,01 转成十进制 =01) |
| 01 00 | 传送原因 |
| 01 00 | 子站地址 |
| 01 00 00 | 点号(1) |
| 00 | 点号 1的遥信状态,为分 |
| 7E 3E | 毫秒 (低字节在前) |
| 18 | 分 |

子站回答: 68 15 9C 01 2E 00 1E 01 03 00 02 00 85 04 00 01 80 6D 13 11 1E 0C 06

目的:以 ASDU30方式上送 SOE(7字节时标)。

报文解析:



| 68 15 9C | |
|----------|--|
| 01 2E 00 | |
| 1E | 数据类型(ASDU方式) |
| 01 | 可变结构限定词 (低 7 位表示一组遥信的个数 ,03 转成十进制 =03) |
| 03 00 | 传送原因 |
| 02 00 | 子站地址 |
| 85 04 00 | 点号 (1157) |
| 01 | 点号 1157的遥信状态,为合 |
| 80 6D | 毫秒 |
| 13 | 分 |
| 11 | 时 |
| 1E | 日 |
| 0C | 月 |
| 06 | 年 |

7、遥控

主站发: 68 0E 04 00 9C 00 2E 01 06 00 02 00 42 60 00 82

目的:对地址为 02的子站发遥控预置。

报文解析:

| 68 0E 04 | | |
|----------|---|---|
| 00 9C 00 | | |
| 2E | 数据类型(ASDU方式) | |
| 01 | 可变结构限定词 | |
| 06 00 | 传送原因 | |
| 02 00 | 子站地址 | |
| 42 60 00 | 点号(60 42 转成十进制 = 24642) | |
| 82 | → 转为二进制 1000 0010 最高位 1表示预置(0表示执行),最低两位 10表示控 | 空 |
| | 合(01表示控分) | |

子站回答: 68 0E 9C 00 06 00 2E 01 07 00 02 00 42 60 00 82

目的:子站上送遥控预置成功的反校报文。

.

主站发: 68 0E 06 00 9E 00 2E 01 06 00 02 00 42 60 00 02

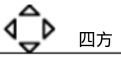
目的:发遥控执行命令。

子站回答: 68 0E 9E 00 08 00 2E 01 07 00 02 00 42 60 00 02

目的:执行确认。

子站回答: 68 0E A0 00 08 00 2E 01 0A 00 02 00 42 60 00 02

目的:一次遥控过程结束。



IEEE754浮点数

2008年 04月 15日 星期二 07:01

一:概述

IEEE754浮点数格式

短实数也称单精度数 符号位 1 位, 阶码 8 位, 尾数 23 位长实数也称双精度数 符号位 1 位, 阶码 11 位, 尾数 52 位

42E48000

第一步, 化为 2 进制

0100 0010 1110 0100 1000 0000 0000 0000

第二步: 因为 IEEE754使用 1 个符号,8 个阶码,23 个尾数,我们分别在上面提取这些内容 .

符号:第一位:0 表示正数

阶码:2-9 位:10000101 为 133, 实际的幂值为 133-127=6

第三步: 根据公式写出实际数值大小为 1110010.01, 化为十进制为:114.25

二:基础知识扩展

值 存储为 指数偏移量

real*4 1位符号位 (s) 、8位指数 (e) , 23位尾数 (m, 共 32位) 127(7FH)

real*8 1位符号位 (s) 、11位指数 (e) ,52位尾数 (m,共 64位) 1023(3FFH)

real*10 1 位符号位 (s) 、15 位指数 (e) ,64 位尾数 (m, 共 80 位) 16383(3FFFH)

计算公式:

V=(-1)^s*2^E*M

V=(-1)^s * 2^(1 - 指数偏移量)*(尾数)

(当指数位为 0 时)

V=(-1)^s * 2^(指数位数值 - 指数偏移量) * (1+ 尾数)

(当指数位不全为 0或不全

情况 1:当 e(各位)为全 '0' 时:

E=1-(2^(e (位数)-1)-1); <===>E=1指数偏移量

M=m

如: real*4 是 8 位, E=1-(2^(8-1)-1)=1-127=-126

在 real*4 时:

V=(-1)^s*2^(-126)*m

在 real*8 时:

V=(-1)^s*2^(-1022)*m

情况 2:当 e(各位)不为全 '0' 且不为全 '1' 时:

E=e(值)-(2^(e (位数)-1)-1); <===>E指数位数值-指数偏移量

```
M=1+m
```

在 real*4 时:

V=(-1)^s*2^(e (值)-127)*(1+m)

在 real*8 时:

V=(-1)^s*2^(e (值)-1023)*(1+m)

三:将浮点格式转换成十进制数

[例 3.1]:

0x00280000 (real*4)

转换成二进制

0000000001010000000000000000000

我们将其分段:

符号位 指数部分(8位) 尾数部分

0 0000000 010100000000000000000

符号位 =0;

指数部分 =0;

尾数部分 M为 m:

该浮点数的十进制为:

(-1)^0*2^(-126)*0.3125

=3.6734198463196484624023016788195e-39

[例 3.2]:

0xC04E00000000000(real*8)

转换成二进制

符号位 指数部分(11位) 尾数部分

符号位 =1;指数 =1028, 因指数部分不为全 '0' 且不为全 '1',则:尾数部分 M为 1+m:

该浮点数的十进制为:

(-1)^1*2^(1028-1023)*1.875

=-60

四:将十进制数转换成浮点格式(real*4)

[例 4.1]:

26.0

十进制 26.0 转换成二进制

11010.0

规格化二进制数

1.10100*2^4

计算指数

4+127=131



[例 4.2]:

0.75

十进制 0.75 转换成二进制

0.11

规格化二进制数

1.1*2^-1

计算指数

-1+127=126

符号位 指数部分 尾数部分

0 01111110 100000000000000000000000

以单精度(real*4)浮点格式存储该数

0011 1111 0100 0000 0000 0000 0000 0000

0x3F40 0000

[例 4.3]:

-2.5

十进制 -2.5 转换成二进制

-10.1

规格化二进制数

-1.01*2^1

计算指数

1+127=128

符号位 指数部分 尾数部分

1 10000000 010000000000000000000000

以单精度(real*4)浮点格式存储该数

1100 0000 0010 0000 0000 0000 0000 0000

0xC020 0000