**J75上下位机通信协议**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

版本：0.0.1：添加初始化故障标识码

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. 上位机发送至下位机（驱动器）通信协议

* 通信方式采用RS422串口通信，五线制（含地线）
* 通讯周期为20ms
* 波特率115200
* 上位机发送至下位机数据协议包如表1所示

表1. 上位机发送至下位机数据协议包

|  |  |  |  |
| --- | --- | --- | --- |
| 内容 | 定义 | 长度（字节） | 备注 |
| 0x5a5a | 包头 | 2 |  |
|  | 包长 | 1 | 一个标识符加上与之对应的指令位（共3个字节）为一个长度 |
|  | 标识符1 | 1 |  |
|  | 指令位1 | 2 |  |
|  | 标识符2 | 1 |  |
|  | 指令位2 | 2 |  |
|  | 。。。。 |  |  |
|  | 。。。。 |  |  |
|  | 标识符n | 1 |  |
|  | 指令位n | 2 |  |
|  | CRC16位校验 | 2 |  |
| 0xa5a5 | 包尾 | 2 |  |

* 标识符与指令位定义如表1.1 所示

表1.1 标识符与指令位定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标识符的值 | 标识符含义 | 对应指令位的值 | 对应指令位的含义 | 备注 |
| 0x01 | 动作指令 | 0 | 停止转动指令 |  |
| 1 | 开始转动指令 |  |
| 0x02 | 转速指令 | 0xXXXX | 目标转速的值(rpm) |  |

例子：

0x5a 0x5a 0x02 0x01 0x00 0x01 0x02 0x00 0xff 0x.. 0x.. 0xa5 0xa5

包头 包长 动作指令 开始转动 转速指令 目标转速 CRC16 包尾

1. 下位机（驱动器）发送至上位机通信协议

* 通信方式采用RS422串口通信，五线制（含地线）
* 通讯周期为20ms
* 波特率115200
* 下位机发送至上位机数据协议包如表2所示

表1. 上位机发送至下位机数据协议包

|  |  |  |  |
| --- | --- | --- | --- |
| 内容 | 定义 | 长度（字节） | 备注 |
| 0x5a5a | 包头 | 2 |  |
|  | 包长 | 1 | 一个标识符加上与之对应的指令位（共3个字节）为一个长度 |
|  | 标识符1 | 1 |  |
|  | 数据位1 | 2 |  |
|  | 标识符2 | 1 |  |
|  | 数据位2 | 2 |  |
|  | 。。。。 |  |  |
|  | 。。。。 |  |  |
|  | 标识符n | 1 |  |
|  | 数据位n | 2 |  |
|  | CRC16位校验 | 2 |  |
| 0xa5a5 | 包尾 | 2 |  |

* 标识符定义如表2.1 所示

表2.1 标识符定义

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标识符的值 | 标识符含义 | 对应数据位的值 | 对应数据位的含义 | 备注 |
| 0x01 | 当前实际转速 | 0xXXXX | 当前电机实际转（rpm） |  |
| 0x02 | 系统报警信息 | 0x00 | 无报警信息，一切正常 |  |
| xxxx xxxx xxxx xxx1(binary) | 系统过流 |  |
| xxxx xxxx xxxx xx1x(binary) | 母线过压 |  |
| xxxx xxxx xxxx x1xx(binary) | 系统过温 |  |
| xxxx xxxx xxxx 1xxx(binary) | 软件故障 |  |
| xxxx xxxx xxx1 xxxx(binary) | 初始化故障 |  |

例子：

0x5a 0x5a 0x01 0x01 0x00 0x01 0x.. 0x.. 0xa5 0xa5

包头 包长 当前实际转速 实际转速的值 CRC16 包尾

1. CRC校验C语言源码

int CalCrc(int crc, const char \*buf, int len){

int x;

int i;

for(i = 0; i < len; ++i){

x = ((crc >> 8) ^ buf[i]) & 0xff;

x ^= x >> 4;

crc = (crc << 8) ^ (x << 12) ^ (x << 5) ^ x;

crc &= 0xffff;

}

return crc;

}