# 协议说明

1. 协议由Header(0xE1), Length(整个命令的字节数量),

Action(命令触发的动作),

Arg(参数, 参数的数量和Action有关,可能为0),

Check(异或校验, 只对Arg进行校验, 如果参数只有一个则等于Arg1, 如果没有参数则等于0)

End(0xEF)

1. 参数的格式.

为了防止参数的值和Header(0xE1), End(0xEF)相等导致检测出错,参数部分的最高位固定为0(即一个字节中只有7bit是有效的,参数部分的字节数值不能大于127),如果需要传的字节输大于127的参数则需要分成2个字节

1. 协议解析.

例如. 0xe1 0x0c 0x88 0x12 0x11 0x08 0x03 0x02 0x04 0x7a check 0xef

0xe1 : Header

0x0c : 命令长度=12个字节

0x88 : Action=0x88表示这是状态的协议

0x12 : arg1 温度[13:7]

0x11 : arg2 温度[6:0]

0x08 : arg3 磁编码器1[13:7]

0x03 : arg4 磁编码器1[6:0]

0x02 : arg5 磁编码器2[13:7]

0x04 : arg6 磁编码器2[6:0]

0x7a : arg7 各个位置开关的状态, 具体见”状态上传协议”说明

check = arg1 ^ arg2 ^ arg3 ^ arg4 ^ arg5 ^ arg6 (没有参数:check=0, 只有一个参数:check=arg1)

0xef : End

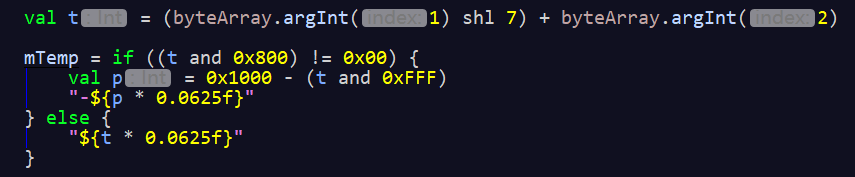
# 状态上传协议(0x88)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字节流 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 定义 | Header | 整条命令的长度 | 命令  类型 | arg1 | arg2 | arg3 | arg4 | arg5 | arg6 | arg7 | 校验字节 | End |
| 值 | 0xE1 | 12 | action(0x88) | 温度[13:0] | | 磁编码器1[13:0] | | 磁编码器2[13:0] | | 各个位置开关 | xor check | 0xEF |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| arg7功能定义 | | | | | | | |
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| 保留 | 空压机检测状态位置开关  0:空压机吸到货物 | 大门位置开关  1:大门关闭 | 升降电机顶部位置开关  0:升降电机位于顶部 | 保留 | 冰箱门关闭位置开关  0:关闭 | 冰箱门打开位置开关  0:打开 | 保留 |

注意:

1. 不需要发命令去查状态
2. 位置开关状态变化的时候状态会自动上传(不超过500ms)
3. 即使状态不发生变化,每隔5s也会上传一次状态
4. 温度数据的解析: (Kotlin语言)



温度解析流程

t = (arg1 << 7) + arg2

如果t的第11位置1则表示这是负数 温度 = -(0x1000 - t) \* 0.0625

如果t的第11位为0则表示这是正数 温度 = t \* 0.0625

# 出货协议(0x06)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字节流 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | .. | .. | 4+2n | 4+2n |
| 定义 | Header | 整条命令的长度 | 命令类型 | arg1 | arg2 | arg3 | arg4 | arg5 | arg6 | .. | .. | 校验字节 | End |
| 值 | 0xE1 | len | action | 超时时间[13:0] | | row1 | col1 | row2 | col2 | .. | .. | xor check | 0xEF |

返回 🡺

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字节流 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 定义 | Header | 整条命令的长度 | 命令类型 | arg1 | arg2 | status | 校验字节 | End |
| 值 | 0xE1 | len | 0x86 | row(当前正在出货的货道) | col(当前正在出货的货道) | 状态 | xor check | 0xEF |

status: 0 => 出货成功

1 => 本货道出货失败,还要继续往下一个货道出货

2 => 全部货道出货都失败

说明:

超时时间[13:0]单位ms, 表示单个电机的超时时间(机械臂, 冰箱门电机, 升降电机),

**机械手臂**寻找目标点的时间超过了超时时间 => 失败

**冰箱**打开或者关闭的时间超过了超时时间 => 失败

**机械臂上升或者下降**的时间超过了超时时间 => 失败,

建议超时时间=10000(即10s)

货道

表1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| row\col | 1 | 2 | 3 | 4 |
| 1 | 1-1 | 1-2 | 1-3 | 1-4 |
| 2 | 2-1 | 2-2 | 2-3 | 2-4 |
| 3 | 3-1 | 3-2 | 3-3 | 3-4 |
| 4 | 4-1 | 4-2 | 4-3 | 4-4 |

假如冰箱的货道分布如表1所示.

上位机发出的出货命令: **(单货道出货的情形)**

0xe1 0x09 0x06 timeOut >> 7 timeOut & 0x7F 0x01 0x02 check 0xef

下位机收到命令发出:

0xe1 0x06 0x6F status, check, 0xef

status = 0x00 表示命令解析成功 将要开始出货

status = 0x01 表示命令Header或者End错误

status = 0x02 表示发送的命令字节数量与Length不符

status = 0x03 表示校验出错

机器开始出货以后发出:

0xe1 0x08 0x86 0x01 0x02 status check 0xef

status = 0x00, 出货成功

status = 0x01, 0x01-0x02出货失败

status = 0x02, 全部货道出货都失败

**(多货道出货的情形)**

0xe1 0x0b 0x06 timeOut >> 7 timeOut & 0x7F **0x02 0x03** **0x03 0x01** check 0xef

下位机收到命令后发出:

0xe1 0x06 0x6F status, check, 0xef

如果命令解析成功,会先对2-3进行出货,如果尝试对2-3出货2次都失败会发出:

0xe1 0x08 0x86 **0x02 0x03** 0x01 check 0xef 表示2-3出货失败

接着对3-1进行出货,如果对尝试对3-1出货2次都失败会发出:

0xe1 0x08 0x86 **0x03 0x01** 0x01 check 0xef 表示3-1出货失败

0xe1 0x08 0x86 **0x03 0x01** 0x02 check 0xef 表示全部货道都是出货失败,出货流程结束

如果对2-3进行出货的时候出货就已经成功了会发出:

0xe1 0x08 0x86 **0x02 0x03** 0x00 check 0xef 表示2-3出货成功 出货流程结束

如果对2-3出货时出货失败,但是3-1出货成功了会发出:

0xe1 0x08 0x86 **0x02 0x03** 0x01 check 0xef 表示2-3出货失败

0xe1 0x08 0x86 **0x03 0x01** 0x00 check 0xef 表示3-1出货成功 出货流程结束

## 开电子锁协议(0xF):

0xE1 0x05 0x0F 0x00 0xEF