**大件摆轮分拣机以太网通信协议**

版本：v2.0.0

目录

[一、通讯简介 3](#_Toc81579272)

[1.1 板卡连接示意图 3](#_Toc81579273)

[1.2 通讯协议格式 3](#_Toc81579274)

[1.3 通讯规则 3](#_Toc81579275)

[二、通讯命令详解 4](#_Toc81579276)

[2.1初始化信息(下料口分配表) 4](#_Toc81579277)

[2.2初始化信息(下料口到达光电时间表) 4](#_Toc81579278)

[2.2.1初始化信息 (摆轮数量与每台摆轮驱动器的数量) 5](#_Toc81579279)

[2.2.2初始化最小分拣间隔（小于间隔值不分拣） 6](#_Toc81579280)

[2.3.1初始化信息(下料口模组的控制参数) 7](#_Toc81579281)

[2.3.2初始化信息(左右分开角度控制) 7](#_Toc81579282)

[2.3.3 左右分角度控制参数读取 9](#_Toc81579283)

[2.4分拣命令 10](#_Toc81579284)

[2.5分拣结果回应 11](#_Toc81579285)

[2.5.1 包裹间距过小的检测结果输出 12](#_Toc81579286)

[2.6.1报警信息1,摆轮报警 12](#_Toc81579287)

[2.6.2 报警信息2，外部电机（皮带机）报警 13](#_Toc81579288)

[2.6.3报警信息3,摆轮电滚筒报警（无O带摆轮驱动器通信） 14](#_Toc81579289)

[2.6.4报警信息4,摆轮报警(4个伺服电机后2个报警) 14](#_Toc81579290)

[2.7发送给上位机的光电信号 15](#_Toc81579291)

[2.7.1发送给上位机包裹的长度信息（遮挡时间） 15](#_Toc81579292)

[2.8板卡心跳命令 16](#_Toc81579293)

[2.9测试下料光电触发时间命令-1 16](#_Toc81579294)

[2.10测试下料光电触发时间命令-2 16](#_Toc81579295)

[2.11读取板卡参数命令 17](#_Toc81579296)

[2.12写入板卡参数命令 18](#_Toc81579297)

[2.13初始化信息(靠边区域时间参数) 19](#_Toc81579298)

[2.14下料口开关信号 20](#_Toc81579299)

[2.15调速命令 21](#_Toc81579300)

[2.16双驱动器速度查询 21](#_Toc81579301)

[2.17 线体启停状态查询 22](#_Toc81579302)

[2.18 线体启停控制 23](#_Toc81579303)

[2.19 摆轮急停状态查询 24](#_Toc81579304)

[2.20 摆轮控制模式查询 24](#_Toc81579305)

[2.21包裹补格口数据上传上位机 25](#_Toc81579306)

[2.22版本号信息上传 25](#_Toc81579307)

[2.23光电滤波时间设置 26](#_Toc81579308)

[2.24超时信息追加输出 26](#_Toc81579309)

[2.26备用光电分拣结果上传（默认使用前一个光电做备用） 27](#_Toc81579310)

[2.27 从板未执行主板发送动作 28](#_Toc81579311)

[2.28.1分拣时是否加速功能初始化 28](#_Toc81579312)

[2.28.2 分拣时是否加速功能配置读取 29](#_Toc81579313)

[2.30备用光电分拣功能初始化 30](#_Toc81579314)

[2.31 驱动器站号组别设置 31](#_Toc81579315)

[2.32 驱动器站号组别读取 31](#_Toc81579316)

[2.33 上位机掉线强制异常口设置 32](#_Toc81579317)

[2.34 初始化信息丢失后的心跳 33](#_Toc81579318)

[2.35 使用过程中初始化格口信息 33](#_Toc81579319)

[2.36是否启用急停按钮 34](#_Toc81579320)

[3.0不允许分拣命令 35](#_Toc81579321)

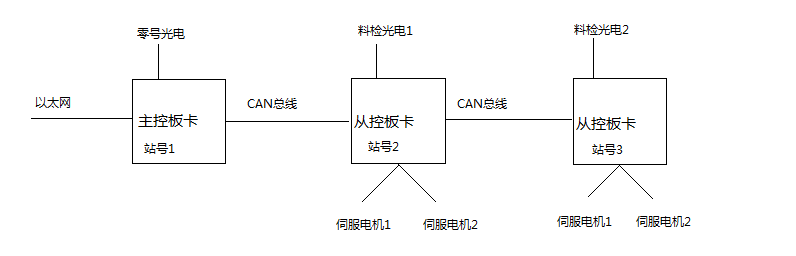
[3.1摆轮单机分拣结果回应 35](#_Toc81579322)

[3.2供包台是否允许供件信息上传 36](#_Toc81579323)

[\*关于重发机制的说明 37](#_Toc81579324)

## 一、通讯简介

### 1.1 板卡连接示意图



### 1.2 通讯协议格式

数据帧格式：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 帧头 | 帧序号 | 帧长度 | 帧校验 | 帧命令 | 数据区 |

* 帧 头：2个字节，区分每一帧数据，帧头约定为“0xAAAA”。
* 帧序号：4个字节，发送方每一帧自动加1，应答方应答时使用接收到发送方使用的帧序号。
* 帧长度：2个字节，表示整个帧的数据长度。
* 帧校验：1个字节，对帧数据中的帧命令和数据区的数据进行异或计算。
* 帧命令：2个字节，帧的命令类型代码，表示该帧数据的功能，发送命令最高位为0，应答方使用的应答命令为接收到的命令最高位或1。
* 数据区：每条命令的具体数据，详见命令说明。

**注**：通信协议帧中的数据全部为小端格式。

### 1.3 通讯规则

上位机系统采用UDP方式和下位板卡进行通讯。上位机使用固定IP：192.168.2.100 port：9000，板卡使用固定IP：192.168.2.110 port：9000。

## 二、通讯命令详解

### 2.1初始化信息(下料口分配表)

此协议: 自动供包摆轮共用

（1）发送时间

上位软件打开后主动发送.

（2）发送帧命令：**0x1A01**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char unloadPortTableNum;

sUnloadPortPara unloadPortParaTable[unloadPortTableNum];

} sPC2Board\_UnloadPortPara;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | unloadPortTableNum | 下料口个数 |
| sUnloadPortPara | unloadPortParaTable[unloadPortTableNum] | 具体的每个下料口的参数信息 |

每个下料口参数信息格式如下:

typedef struct{

unsigned char unload\_port\_num; //下料口号

unsigned char broad\_num; //从控站号

unsigned char unload\_port\_dir; //下料口方向

} sUnloadPortPara;

（4）应答帧命令：**0x9A01**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_UnloadPortPara\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.2初始化信息(下料口到达光电时间表)

此协议: 自动供包摆轮共用

（1）发送时间

上位软件打开后主动发送.

（2）发送帧命令：**0x1A02**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char unloadPhotoNum;

unsigned int32 unloadPortTime[unloadPhotoNum];

} sPC2Board\_UnloadPortTime;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | unloadPhotoNum | 下料口光电个数 |
| unsigned int32 | unloadPortTime[unloadPhotoNum] | 从零号光电开始到下料口光电的时间ms |

（4）应答帧命令：**0x9A02**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_UnloadPortTime\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.2.1初始化信息 (摆轮数量与每台摆轮驱动器的数量)

此协议: 自动供包摆轮共用

（1）发送时间

上位软件打开后主动发送.

（2）发送帧命令：**0x1A05**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char balance\_num;

unsigned char sSwingingWheelData[SwingingWheel\_Count];

} sPC2Board\_SwingingWheel\_Param;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | SwingingWheel\_Count | 摆轮数量 |
| sSwingingWheelData | SwingingWheelData [SwingingWheel\_Count] | 摆轮参数 |

typedef struct {

unsigned char SwingingWheel\_Index; //摆轮（板卡）序号;

unsigned char SwingingWheel\_Type;

//摆轮类型（1：单驱铁摆轮（或者单驱O带摆轮）、2: 双驱铁摆轮（或者双驱O带摆轮）、 3: 多驱铁摆轮（或者多驱O带摆轮）、 4: GSD195-M33P27 摩擦&包胶摩擦摆轮（多驱） 、5：GSD195-M24P27 摩擦&包胶摩擦摆轮（多驱） 、6： GSD195-M222P27 摩擦&包胶摩擦摆轮（多驱）、 7： GSD120-M2233P85 4K大包胶摆轮（双驱）、 8： GSD120-M2233P85 4K大包胶摆轮（多驱） 9：GSD120-M2222P85 6K大包胶摆轮、10：GSD90-M2223P59 自动供包小包胶摆轮(双驱) 、11： GSD90-M2223P59 自动供包小包胶摆轮(多驱)

摆轮类型名称解释：举例 （GSD195-M33P27 摩擦&包胶摩擦摆轮（GSD195代表单元直径，195一般都是大盘。M33，代表每组伺服电机带动的单元排数，33此处代表两个电机、每个电机带动三排，P27代表单元个数）

unsigned char Driver\_Count; //每台摆轮的驱动器数量

} sSwingingWheelData

（4）应答帧命令：**0x9A05**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_SwingingWheel\_Param\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.2.2初始化最小分拣间隔（小于间隔值不分拣）

（1）发送时间

上位软件打开后主动发送.（如果上位机没有发送或者发送的数值是0则不启用此功能）

（2）发送帧命令：**0x1A06**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int16 minInterval;

} sPC2Board\_MinInterval\_Param;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int16 | minInterval | 第一个包裹尾部到下个包裹头部的最小间隔时间 |

（4）应答帧命令：**0x9A06**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_MinInterval\_Param\_ACK;

### 2.3.1初始化信息(下料口模组的控制参数)

此协议: 自动供包摆轮共用

（1）发送时间

上位软件打开后主动发送.此参数需要写入Flash中。

（2）发送帧命令：**0x1A03**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char broad\_num;

unsigned char module\_num;

unsigned int16 photo\_dly;

unsigned int16 start\_dly;

unsigned int16 motor\_follow\_t;

} sPC2Board\_MotorCtrlPara;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | broad\_num | 板卡的站号 |
| unsigned char | unloadPhotoNum | 一个下料口包含的模组数(电机数) |
| unsigned int16 | photo\_dly | 光电触发延时时间 |
| unsigned int16 | start\_dly | 电机延时动作时间（伺服电机1，2的延时启动参数） |
| unsigned int16 | motor\_follow\_t | 跟随电机延时启动时间（伺服电机2在延时启动参数基础上再附加一个延时参数） |

（4）应答帧命令：**0x9A03**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_MotorCtrlPara\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

(注意:每一个主从控盒子都需要发送这样一个控制参数,而且可以分别配置)

### 2.3.2初始化信息(左右分开角度控制)

（1）发送时间

上位软件打开后主动发送.配置文件有此协议时不再发送上面的0x1A03协议。此参数需要写入Flash中。每个板卡的参数单独发送

（2）发送帧命令：**0x1A12**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char Balance\_id //板卡id

unsigned char unloadPhotoNum //一个下料口包含的模组数(电机数)

unsigned int16 zero\_start\_dly //0方向启动延时时间

unsigned int16 zero\_motor\_follow\_t1 //0方向跟随延迟时间1

unsigned int16 zero\_motor\_follow\_t2 //0方向跟随延迟时间2

unsigned int16 zero\_motor\_follow\_t3 //0方向跟随延迟时间3

unsigned int16 one\_start\_dly //1方向启动延时时间

unsigned int16 zero\_motor\_follow\_t1 //1方向跟随延迟时间1

unsigned int16 zero\_motor\_follow\_t2 //1方向跟随延迟时间2

unsigned int16 zero\_motor\_follow\_t3 //1方向跟随延迟时间3

unsigned int16 zero\_pulse\_1 //0方向脉冲1

unsigned int16 zero\_pulse\_2 //0方向脉冲2

unsigned int16 zero\_pulse\_3 //0方向脉冲3

unsigned int16 zero\_pulse\_4 //0方向脉冲4

unsigned int16 one\_pulse\_1 //1方向脉冲1

unsigned int16 one\_pulse\_2 //1方向脉冲2

unsigned int16 one\_pulse\_3 //1方向脉冲3

unsigned int16 one\_pulse\_4 //1方向脉冲4

} sPC2Board\_MotorCtrlPara;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | broad\_num | 板卡的站号 |
| unsigned char | unloadPhotoNum | 一个下料口包含的模组数(电机数) |
| unsigned int16 | zero\_start\_dly | 0方向启动延时时间 |
| unsigned int16 | zero\_motor\_follow\_t1 | 0方向跟随延迟时间1 |
| unsigned int16 | zero\_motor\_follow\_t2 | 0方向跟随延迟时间2 |
| unsigned int16 | zero\_motor\_follow\_t3 | 0方向跟随延迟时间3 |
| unsigned int16 | one\_start\_dly | 1方向启动延时时间 |
| unsigned int16 | zero\_motor\_follow\_t1 | 1方向跟随延迟时间1 |
| unsigned int16 | zero\_motor\_follow\_t2 | 1方向跟随延迟时间2 |
| unsigned int16 | zero\_motor\_follow\_t3 | 1方向跟随延迟时间3 |
| unsigned int16 | zero\_pulse\_1 | 0方向脉冲1 |
| unsigned int16 | zero\_pulse\_2 | 0方向脉冲2 |
| unsigned int16 | zero\_pulse\_3 | 0方向脉冲3 |
| unsigned int16 | zero\_pulse\_4 | 0方向脉冲4 |
| unsigned int16 | one\_pulse\_1 | 1方向脉冲1 |
| unsigned int16 | one\_pulse\_2 | 1方向脉冲2 |
| unsigned int16 | one\_pulse\_3 | 1方向脉冲3 |
| unsigned int16 | one\_pulse\_4 | 1方向脉冲4 |

（4）应答帧命令：**0x9A12**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_MotorCtrlPara\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

(注意:每一个主从控盒子都需要发送这样一个控制参数,而且可以分别配置)

### 2.3.3 左右分角度控制参数读取

（1）发送时间

上位软件打开后主动发送.

（2）发送帧命令：**0x1A13**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char Balance\_id //板卡id

} sPC2Board\_MotorCtrlPara\_Upload;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | broad\_num | 板卡的站号 |

（4）应答帧命令：**0x9A13**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char Balance\_id //板卡id

unsigned char unloadPhotoNum //一个下料口包含的模组数(电机数)

unsigned int16 zero\_start\_dly //0方向启动延时时间

unsigned int16 zero\_motor\_follow\_t1 //0方向跟随延迟时间1

unsigned int16 zero\_motor\_follow\_t2 //0方向跟随延迟时间2

unsigned int16 zero\_motor\_follow\_t3 //0方向跟随延迟时间3

unsigned int16 one\_start\_dly //1方向启动延时时间

unsigned int16 zero\_motor\_follow\_t1 //1方向跟随延迟时间1

unsigned int16 zero\_motor\_follow\_t2 //1方向跟随延迟时间2

unsigned int16 zero\_motor\_follow\_t3 //1方向跟随延迟时间3

unsigned int16 zero\_pulse\_1 //0方向脉冲1

unsigned int16 zero\_pulse\_2 //0方向脉冲2

unsigned int16 zero\_pulse\_3 //0方向脉冲3

unsigned int16 zero\_pulse\_4 //0方向脉冲4

unsigned int16 one\_pulse\_1 //1方向脉冲1

unsigned int16 one\_pulse\_2 //1方向脉冲2

unsigned int16 one\_pulse\_3 //1方向脉冲3

unsigned int16 one\_pulse\_4 //1方向脉冲4

} sPC2Board\_MotorCtrlPara\_Upload\_ack;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | broad\_num | 板卡的站号 |
| unsigned char | unloadPhotoNum | 一个下料口包含的模组数(电机数) |
| unsigned int16 | zero\_start\_dly | 0方向启动延时时间 |
| unsigned int16 | zero\_motor\_follow\_t1 | 0方向跟随延迟时间1 |
| unsigned int16 | zero\_motor\_follow\_t2 | 0方向跟随延迟时间2 |
| unsigned int16 | zero\_motor\_follow\_t3 | 0方向跟随延迟时间3 |
| unsigned int16 | one\_start\_dly | 1方向启动延时时间 |
| unsigned int16 | zero\_motor\_follow\_t1 | 1方向跟随延迟时间1 |
| unsigned int16 | zero\_motor\_follow\_t2 | 1方向跟随延迟时间2 |
| unsigned int16 | zero\_motor\_follow\_t3 | 1方向跟随延迟时间3 |
| unsigned int16 | zero\_pulse\_1 | 0方向脉冲1 |
| unsigned int16 | zero\_pulse\_2 | 0方向脉冲2 |
| unsigned int16 | zero\_pulse\_3 | 0方向脉冲3 |
| unsigned int16 | zero\_pulse\_4 | 0方向脉冲4 |
| unsigned int16 | one\_pulse\_1 | 1方向脉冲1 |
| unsigned int16 | one\_pulse\_2 | 1方向脉冲2 |
| unsigned int16 | one\_pulse\_3 | 1方向脉冲3 |
| unsigned int16 | one\_pulse\_4 | 1方向脉冲4 |

### 2.4分拣命令

（1）发送时间

上位机收到下料口光电信号后查找到相应的下料口然后发送分拣命令。

（2）发送帧命令：**0x1B01**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 msg\_id;

unsigned char unload\_port\_num;

unsigned int16 time\_delay;

unsigned int16 photo\_trigger\_time;（这个长度值V9会发送，其他版本不会发送）

} sPC2Board\_unload\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int32 | msg\_id | 下料命令序号 |
| unsigned char | unload\_port\_num | 下料口编号 |
| unsigned int16 | time\_delay | 从相机触发光电开始到收到下料信息经过的时间 |
| unsigned int16 | photo\_trigger\_time | 从相机触发光电上升沿到下降沿时间  （这个长度值V9会发送，其他版本不会发送） |

（4）应答帧命令：**0x9B01**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 package\_id;（包裹号,老版本板卡的ACK没有包裹号，能发送版本号的板卡会带有包裹号，需要3次最多重发功能配合使用）

} sPC2Board\_unload\_Data\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.5分拣结果回应

（1）发送时间

下位机执行下料动作后回应上位机

（2）发送帧命令：**0x1B02**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char msg\_num;

sMsgData msg\_data[msg\_num];

} sPC2Board\_unload\_Result;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | msg\_num | 回应的消息个数 |
| sMsgData | msg\_data[msg\_num] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char cmd\_type; //回应消息的类型(0:正常分拣 1:下料超时 2:下料口关闭 3:下料失败 4:包裹间距小于设定值 5:从板执行分拣动作超时(摆轮不回证版本不适用)) 6：叠件回流

unsigned int32 msg\_id; //下料命令序号

} sMsgData

（4）应答帧命令：0x9B02

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 package\_id;（包裹号,板卡收到ACK或者达到最大重发次数才会从队列中移除此包裹的分拣结果，需要3次最多重发功能配合使用）

} sPC2Board\_unload\_ Result\_ACK;

### 2.5.1 包裹间距过小的检测结果输出

（1）发送时间

板卡判断包裹间距过小后追加发送

（2）发送帧命令:0x1B04

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 pkg\_id;

unsigned int16 pkg\_length;

unsigned int16 interval\_before;

unsigned int16 interval\_after;

} sPC2Board\_Pkg\_Interval;

（4）应答帧命令：无需应答

### 2.6.1报警信息1,摆轮报警

此协议: 自动供包摆轮共用

（1）发送时间

板卡每隔1秒钟向上位机发送报警信息。

（2）发送帧命令：**0x1C01**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char error\_num;

sAlarmData error\_data[error\_num];

} sBoard2PC\_Alarm\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | error\_num | 有报警的控制盒的数量 |
| sAlarmData | error\_data[error\_num] | 每个控制盒的报警数据 |

报警数据格式如下:

typedef struct{

unsigned char broad\_num; //控制器站号

unsigned char error\_status; //错误状态(bit0:零号光电/料检光电,bit1:伺服电机1，bit2:伺服电机2,bit3:变频器,bit4:接近开关1,bit5:接近开关2,bit6:通信，bit7 相机一直给禁止供包信号

} sAlarmData

（4）应答帧命令：无需应答

### 2.6.2 报警信息2，外部电机（皮带机）报警

（1）发送时间

板卡每隔1秒钟向上位机发送报警信息。

（2）发送帧命令：**0x1801**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char error\_num;

sExMotorData error\_data[error\_num];

} sBoard2PC\_EXMotor\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | error\_num | 有报警外部电机总个数 |
| sAlarmData | error\_data[error\_num] | 每个外部电机的报警信息 |

报警数据格式如下:

typedef struct{

unsigned char exmotor\_count; //外部电机序号;

unsigned char error\_status; //错误状态（0x1：电机报警）

} sEXMotorAlarmData

（4）应答帧命令：无需应答

### 2.6.3报警信息3,摆轮电滚筒报警（无O带摆轮驱动器通信）

此协议: 自动供包摆轮共用

（1）发送时间

检测到有驱动器报警状态时上传，一秒一次,一次传一个从站的报警信息，无错误状态不上传。

（2）发送帧命令：**0x1401**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char board\_ID;

unsigned char error\_num;

sDriverData error\_data[error\_num];

} sBoard2PC\_Alarm\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | board\_ID | 有报警的站号 |
| unsigned char | error\_num | 有报警的电滚筒总数量 |
| sAlarmData | error\_data[error\_num] | 每个电滚筒的报警数据 |

报警数据格式如下:

typedef struct {

unsigned char driver\_count; //电滚筒序号;

unsigned char error\_status; //错误状态

0x01：滚筒电机堵转

0x02：滚筒电机过流

0x03：48V电源过压/欠压

0x04：滚筒驱动器报警

0x0f：滚筒驱动器未连接

0x10: 设置驱动器速度失败

0x20: 设置驱动器分组失败

0x40: 设置驱动器型号失败

} sDriverData

（4）应答帧命令：无需应答

### 2.6.4报警信息4,摆轮报警(4个伺服电机后2个报警)

此协议: 自动供包摆轮共用

（1）发送时间

板卡每隔1秒钟向上位机发送报警信息。

（2）发送帧命令：**0x1C02**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char error\_num;

sAlarmData error\_data[error\_num];

} sBoard2PC\_Alarm\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | error\_num | 有报警的控制盒的数量 |
| sAlarmData | error\_data[error\_num] | 每个控制盒的报警数据 |

报警数据格式如下:

typedef struct{

unsigned char broad\_num; //控制器站号

unsigned char error\_status; //错误状态(bit0:伺服电机3，bit1:伺服电机4 ,bit2:接近开关3,bit3:接近开关4)

} sAlarmData

（4）应答帧命令：无需应答

### 2.7发送给上位机的光电信号

此协议: 自动供包摆轮共用

（1）发送时间

板卡收到光电开关触发信号时向上位机发送该命令

（2）发送帧命令：**0x1D01**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char photoSwitch\_num;

} sBoard2PC\_Photoelectric\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | photoSwitch\_num | 光电开关序号 |

（4）应答帧命令：无需应答

### 2.7.1发送给上位机包裹的长度信息（遮挡时间）

(1)发送时间

板卡通过光电(一般通过0号光电)采集到包裹的长度后向上位机发送该命令

（2）发送帧命令：0x1D02

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int16 pkg\_length;

} sBoard2PC\_PKG\_LENGTH;

（4）应答帧命令：无需应答

### 2.8板卡心跳命令

此协议: 自动供包摆轮共用

（1）发送时间

板卡与上位机连续5秒未发生数据交互时发送。

（2）发送帧命令：**0x1150**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

} sBoard2PC\_Heart\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

（4）应答帧:无需应答

### 2.9测试下料光电触发时间命令-1

（1）发送时间

上位机向板卡发送进入测试模式命令

（2）发送帧命令：**0x1E01**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char test\_mode;

} sPC2Board\_Test\_Mode;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | test\_mode | 0:普通模式 1:测试模式 |

（4）应答帧命令：无需应答

### 2.10测试下料光电触发时间命令-2

（1）发送时间

板卡把每一个到下料口光电的时间发送到上位机

（2）发送帧命令：**0x1E02**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char photo\_index;

unsigned int32 photo\_time;

} sBoard2PC\_Test\_Mode;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | photo\_index | 下料口光电的序号(从1开始) |
| unsigned int32 | photo\_time | 包裹到达该下料口的时间ms |

（4）应答帧命令：无需应答

### 2.11读取板卡参数命令

此协议: 自动供包摆轮共用

（1）发送时间

仅在板卡boot模式中使用,用于调试。

（2）发送帧命令：**0x1F10**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

Unsigned char Board\_num; 板卡号

} sPC2Board\_Read\_Para;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

（4）应答帧命令：**0x9F10**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int16 motor\_find\_zero\_cnt[4];

unsigned int16 motor\_speed\_modulus;

unsigned int16 motor\_rolling\_pulse;

unsigned int16 drive\_available;

unsigned int16 motor\_slave\_para\_backup [3];

unsigned int16 photo\_delay \_para\_cnt[2];

unsigned int16 motor\_hold\_time;

unsigned int16 motor\_master\_para\_backup[3];

signed int16 compensation[4];

} sPC2Board\_Read\_Para\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int16 | motor\_find\_zero\_cnt[4] | 从站参数：电机1-4寻零脉冲数，电机寻零脉冲默认1600(铁摆轮)，O带摆轮需手动设置为3300。 |
| unsigned int16 | motor\_speed\_modulus | 从站参数：伺服电机转动速度系数。默认值3，范围3~33 |
| unsigned int16 | motor\_rolling\_pulse | 从站参数：摆动脉冲数（电机转动角度）默认值2000。 |
| unsigned int16 | drive\_available | 驱动器速度 |
| unsigned int16 | motor\_slave\_para\_backup[3] | 从站备用参数 |
| unsigned int16 | photo\_delay \_para\_cnt[2] | 主站参数：光电检测允许提前量，默认值200。光电检测允许延后量，值默认值450。 |
| unsigned int16 | motor\_hold\_time | 主站参数：电机摆出保持时间，默认值500 |
| unsigned int16 | motor\_ master\_para\_backup[3] | 主站boot参数备用预留 |
| signed int16 | Compensation[4] | 从站机械间隙补偿参数 |

### 2.12写入板卡参数命令

此协议: 自动供包摆轮共用

（1）发送时间

仅在板卡boot模式中使用,用于调试。

（2）发送帧命令：**0x1F11**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int16 motor\_find\_zero\_cnt[4];

unsigned int16 motor\_speed\_modulus;

unsigned int16 motor\_rolling\_pulse;

unsigned int16 motor\_slave\_para\_backup [4];

unsigned int16 photo\_delay \_para\_cnt[2];

unsigned int16 motor\_hold\_time;

unsigned int16 motor\_master\_para\_backup[3];

signed int16 compensation[4];

} sPC2Board\_Write\_Para;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int16 | motor\_find\_zero\_cnt[4] | 从站参数：电机1-4寻零脉冲数，电机寻零脉冲默认1600。 |
| unsigned int16 | motor\_speed\_modulus | 从站参数：伺服电机转动速度系数。默认值3，范围3~33 |
| unsigned int16 | motor\_rolling\_pulse | 从站参数：摆动脉冲数（电机转动角度） |
| unsigned int16 | drive\_available | 驱动器速度 |
| unsigned int16 | motor\_slave\_para\_backup[2] | 从站备用参数 |
| unsigned int16 | motorCtrlPara.start\_dly | 启动延时时间 |
| unsigned int16 | motorCtrlPara.motor\_follow\_t | 跟随时间 |
| unsigned int16 | photo\_delay \_para\_cnt[2] | 主站参数：光电检测向前允许误差量，光电检测向后允许误差量 |
| unsigned int16 | motor\_hold\_time | 主站参数：电机摆出保持时间 |
| unsigned int16 | motor\_ master\_para\_backup[3] | 主站boot参数备用预留 |
| signed int16 | Compensation[4] | 从站机械间隙补偿参数 |

（4）应答帧命令：**0x9F11**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_Write\_Para\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.13初始化信息(靠边区域时间参数)

（1）发送时间

上位软件打开后主动发送.

（2）发送帧命令：**0x1A04**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 keepSideDelayTime;

} sPC2Board\_KeepSidePara;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int32 | keepSideDelayTime | 相机触发光电到靠边区的时间（ms） |

（4）应答帧命令：**0x9A04**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_KeepSidePara\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.14下料口开关信号

（1）发送时间

上位软件在需要时主动发送给板卡.

（2）发送帧命令：**0x1B03**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char port\_num;

sPortClose portCloseData[port\_num];

} sPC2Board\_PortClose\_Cmd;

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | port\_num | 开关的下料口数量 |
| sPortClose | portCloseData[port\_num] | 下料口开关数据部分 |

下料口开关部分数据格式如下:

typedef struct{

unsigned char unload\_port\_num; //下料口号

unsigned char unload\_port\_close; //下料开关状态(0：开 1：关)

} sPortClose

（4）应答帧命令：**0x9B03**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_PortClose\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.15调速命令

此协议: 自动供包摆轮共用

（1）发送时间

上位软件在需要时主动发送给板卡.仅限于调节双驱动器驱动的电滚筒

（2）发送帧命令：**0x1901**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char driverSpeedValue;

} sPC2Board\_DriverSpeedValue\_cmd

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | driverSpeedValue | 驱动器速度值 |

驱动器速度值部分数据解释：

|  |  |
| --- | --- |
| 速度(M/S) | 调速指令3字节（速度只是对于现有摆轮63.5MM直径参考，准确的需要根据实际调节） |
| 1.5 | 18 |
| 1.8 | 1D（默认值） |
| 2.0 | 20 |
| 2.5 | 28 |
| 3 | 2F |
| 3.5 | 38 |

（4）应答帧命令：**0x9901**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_DriverSpeedValue\_cmd；

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.16双驱动器速度查询

此协议: 自动供包摆轮共用

（1）发送时间

上位软件在需要时主动发送给板卡.仅限于获取双驱动器驱动的电滚筒速度. 在驱动器速度改变的时候主动发送给上位机软件.

（2）发送帧命令：**0x1902**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_QueryDriverSpeed\_cmd

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

（4）应答帧命令：**0x9902**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

Unsigned char balance\_num; //摆轮数量

sSDriverSpeedData[balance\_num] ;

} sPC2Board\_DriverSpeedResult\_cmd；

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | balance\_ num | 从板卡数量 |
| sDriverSpeedData | sDriverSpeedData[balance\_ num] | 驱动器速度信息 |

驱动器速度信息

typedef struct{

unsigned char balance\_index;

unsigned char driverSpeedValue;

} sDriverSpeedData

### 2.17 线体启停状态查询

此协议: 自动供包摆轮共用

（1）发送时间

上位软件在需要时主动发送给板卡.在启停状态改变的时候主动发送给上位机软件.

（2） 发送帧命令：**0x1201**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_QueryIsRunning\_cmd

|  |  |  |
| --- | --- | --- |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

（4）应答帧命令：**0x9201**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char num;

srunstate IsRunning[num] ; //线体启停状态(0：停止 1：运行)

} sPC2Board\_ IsRunning \_cmd；

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | num | 摆轮数量 |
| srunstate | IsRunning[num] | 摆轮启停状态信息(0：停止 1：运行) |

typedef struct{

unsigned char balance\_index; 板卡号

unsigned char runstate\_Value; 启停状态信息

} srunstate;

### 2.18 线体启停控制

此协议: 自动供包摆轮共用

（1）发送时间：

上位机软件在需要控制线体启停的时候主动发送给板卡.

（2）发送帧命令：**0x1202**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char StartOrStop;

} sPC2Board\_StartOrStop\_cmd

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | StartOrStop | 启停状态(0：停止 1：运行) |

（4）应答帧命令：**0x9202**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_StartOrStop\_Ack

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.19 摆轮急停状态查询

此协议: 自动供包摆轮共用

（1）发送时间

上位软件在需要时主动发送给板卡.在急停状态改变的时候主动发送给上位机软件.

（2） 发送帧命令：**0x1301**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_QueryEStop\_cmd

|  |  |  |
| --- | --- | --- |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

（4）应答帧命令：**0x9301**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char num;

EStopState EStop[num] ; //线体急停状态(0：正常 1：急停)

} sPC2Board\_ EStop \_cmd；

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | num | 摆轮数量 |
| EStopState | EStop[num] | 摆轮启停状态信息(0：正常 1：急停) |

typedef struct{

unsigned char board\_index; 板卡号

unsigned char EStop\_Value; 急停状态信息

} s\_EStop\_State;

### 2.20 摆轮控制模式查询

此协议: 自动供包摆轮共用

（1）发送时间

上位软件在需要时主动发送给板卡查询.上电时板卡将各个从板卡的本地远程状态主动发给上位机

（2） 发送帧命令：**0x1501**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_QueryEStop\_cmd

|  |  |  |
| --- | --- | --- |
| 变量类型 | 变量名 | 含义 |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

（4）应答帧命令：**0x9501**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char num;

CModeInfo [num] ; //摆轮控制模式(0：本地 1：远程)

} sBoard2pc\_CModeInfo\_cmd；

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | num | 板卡数量 |
| s\_CModeInfo | CModeInfo[num] | 摆轮控制模式信息(0：本地 1：远程) |

typedef struct{

unsigned char board\_index; 板卡号

unsigned char Elocal\_Value; 本地远程状态信息

} s\_CModeInfo;

### 2.21包裹补格口数据上传上位机

（1）发送时间

现场操作员按下补格口按钮后发送对应格口数据给上位机。

（2）发送帧命令：0x1D03

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int16 SortPortID;

} sPC2Board\_CSP\_cmd

|  |  |  |
| --- | --- | --- |
| 变量类型 | 变量名 | 含义 |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int16 | SortPortID | 按钮代表的格口ID |

（4）应答帧命令：**无需应答**

### 2.22版本号信息上传

此协议: 自动供包摆轮共用

（1）发送时间

初始化完成后主动上传一次

（2）发送帧命令：0x1E03

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 version;

} sPC2Board\_ version;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int32 | version | 版本号（比如1.2.1.6 则 对应 0x06 0x01 0x02 0x01） |

（4）应答帧命令：无需应答

### 2.23光电滤波时间设置

（1）发送时间

上位机需要时发送，如果板卡收到的值不在合理范围内，则采用板卡内部默认值

（2）发送帧命令：0x1E04

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int16 FilteringTime;

} sPC2Board\_ FilteringTime;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int16 | FilteringTime | 滤波时长 |

（4）应答帧命令：0x9E04

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

}  sPC2Board\_ FilteringTime\_Ack;

### 2.24超时信息追加输出

（1）发送时间

包裹分拣超时后发送

（2）发送帧命令：0x1B07

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 package\_id;

unsigned char board\_id;

unsigned int32 Cumulative\_Time;

unsigned char msg\_num;

sMsgData msg\_data[msg\_num];

} sPC2Board\_OverTimePackageInfo;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int32 | package\_id | 包裹ID |
| unsigned char | board\_id | 板卡站号 |
| unsigned int32 | Cumulative\_Time | 累计计时 |
| unsigned char | msg\_num | 消息数据个数 |
| sMsgData | msg\_data[msg\_num] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char board\_id; //板卡站号ID

unsigned char check\_flag; //校验标志（0：提前，1：延后）

unsigned int16 RevisionTime; //修正时间

} sMsgData

（4）应答帧命令：无需应答

### 2.26备用光电分拣结果上传（默认使用前一个光电做备用）

（1）发送时间

下位机通过备用光电执行下料动作后回应上位机

（2）发送帧命令：**0x1B08**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char msg\_num;

sMsgData msg\_data[msg\_num];

} sPC2Board\_unload\_Result;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | msg\_num | 回应的消息个数 |
| sMsgData | msg\_data[msg\_num] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char cmd\_type; //回应消息的类型(0:正常分拣 1:下料超时 2:下料口关闭 3:下料失败 4:包裹间距小于设定值 5:从板执行分拣动作超时(摆轮不回证版本不适用))

unsigned int32 msg\_id; //下料命令序号

} sMsgData

（4）应答帧命令：0x9B08

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int32 package\_id;（包裹号,板卡收到ACK或者达到最大重发次数才会从队列中移除此包裹的分拣结果，需要3次最多重发功能配合使用）

} sPC2Board\_unload\_ Result\_ACK;

### 2.27 从板未执行主板发送动作

（1）发送时间

主板卡检测到从板卡未能执行主板卡命令时上传上位机记录。

（2）发送帧命令：0x1A09

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char board\_id;

}  sPC2Board\_InvalidBoard;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | board\_id | 未能执行主板命令的从板站号 |

（4）应答帧命令：无需应答

### 2.28.1分拣时是否加速功能初始化

（1）发送时间

上位机需要设置时发送，若本次初始化上位机没有发送给板卡，则所有分拣口按照默认值工作。另外此功能可以为每个分拣格口单独设置，为了方便上位机配置，假设6个分拣格口（序号1-6），上位机可以单独设置，2号和5号格口启用分拣加速1.5倍功能，1号3号4号6号格口使用默认值。格口号需要通过板卡ID和方向来定位。此参数需要写入Flash中。

1. 发送帧命令：0x1A0B
2. 数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char sMsgData;

}  sPC2Board\_IsAccelerate;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| sMsgData | msg\_data | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char BoardID; （分拣板卡ID）

unsigned char dir; （分拣的左右方向）

unsigned char SpeedUpType; (IO加速对应0 485加速对应1 ，不加速 2)

unsigned char SpeedupRatio; （加速比,0x0A为基准1.0倍速度,0x11为1.7倍速度）（对于只能加速或者不能加速的摆轮，0x0A为不加速，其余值均为加速）

} sMsgData

（4）应答帧命令：0x9A0B

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

}sPC2Board\_IsAccelerate\_Ack;

### 2.28.2 分拣时是否加速功能配置读取

（1）发送时间

上位机需要时主动发送。

（2）发送帧命令：0x1A14

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char Board\_id;

}  sPC2Board\_ IsAccelerate\_Upload;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | Board\_id | 板卡ID |

（4）应答帧命令：0x9A14

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char sMsgData;

}  sPC2Board\_IsAccelerate\_Upload\_ack;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| sMsgData | msg\_data | 消息数据部分 |

typedef struct{

unsigned char BoardID; （分拣板卡ID）

unsigned char dir; （分拣的左右方向）

unsigned char SpeedUpType; (IO加速对应0 485加速对应1)

unsigned char SpeedupRatio; （加速比,0x0A为基准1.0倍速度,0x11为1.7倍速度）（对于只能加速或者不能加速的摆轮，0x0A为不加速，其余值均为加速）

} sMsgData

### 2.30备用光电分拣功能初始化

（1）发送时间

上位机需要设置时发送，若上位机没有发送给板卡，则板卡按照默认值（启用）工作。另外此功能可以为每张从板卡单独设置，为了方便上位机配置，假设5张从板卡（序号2-6），上位机可以单独设置， 5号板卡进入备用光电分拣功能，3号4号6号板卡不启用此功能。

（2号从板卡 无法用备用光电分拣）

（2）发送帧命令：0x1A0C

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

Unsigned char msg\_num;

unsigned char sMsgData[msg\_num];

}  sPC2Board\_BackUp\_EnableStatus;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | msg\_num | 消息个数 |
| sMsgData | msg\_data[msg\_num] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char board\_id; （对应的板卡站号ID）

} sMsgData

（4）应答帧命令：0x9A0C

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

}sPC2Board\_BackUp\_EnableStatus\_Ack;

### 2.31 驱动器站号组别设置

（1）发送时间

上位机或者调试demo在有需要时发送，此参数需要掉电保存。

1. 发送帧命令：0x1A0D

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char BoardID; （板卡站号）

unsigned char GroupCount;（组别总数）

unsigned char sMsgData[GroupCount];

}  sPC2Board\_GroupInfo;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | BoardID | 从站ID号 |
| unsigned char | GroupCount | 组别总数 |
| sMsgData | msg\_data[GroupCount] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char Group\_ID;

unsigned char Driver\_Num;

unsigned char DriverID[Driver\_Num];

} sMsgData

（4）应答帧命令：0x9A0D

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char BoardID; （板卡站号）

}  sPC2Board\_GroupInfo\_Ack;

### 2.32 驱动器站号组别读取

（1）发送时间

上位机或者调试demo在有需要时发送，方便现场人员调试。

（2）发送帧命令：0x1A0E

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char BoardID;

}  sPC2Board\_GroupInfo\_Get;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | BoardID | 板卡站号 |

（4）应答帧命令：0x9A0E

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char BoardID; （板卡站号）

unsigned char GroupCount;（组别总数）

unsigned char sMsgData[GroupCount];

}  sPC2Board\_GroupInfo;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | BoardID | 从站ID号 |
| unsigned char | GroupCount | 组别总数 |
| sMsgData | msg\_data[GroupCount] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char Group\_ID;

unsigned char Driver\_Num;

unsigned char DriverID[Driver\_Num];

} sMsgData

### 2.33 上位机掉线强制异常口设置

（1）发送时间

上位机有需要时发送，目的是防止现场没有开启上位机就开始供包，导致漏扫。当收到的异常口数量为0时，代表取消强制异常掉线口。

（2）发送帧命令：0x1A0F

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char Port\_Num;

unsigned char PortID[Port\_Num];

}    sPC2\_SDS\_ErrorSP;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | Port\_Num | 异常口数量 |
| unsigned char | PortID[Port\_Num] | 异常口具体ID |

（4）应答帧命令：0x9A0F

typedef struct{

unsigned char cmd[11];

}  sPC2\_SDS\_ErrorSP\_ACK;

### 2.34 初始化信息丢失后的心跳

（1）发送时间

板卡检测到自身丢失了初始化信号后每5秒发送一次，发送此心跳时不需要发送正常心跳。

（2）发送帧命令：**0x1151**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

} sBoard2PC\_Heart\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

（4）应答帧:无需应答

### 2.35 使用过程中初始化格口信息

（1）发送时间

上位机距离上次初始化一定时间后（防止多次初始化），收到板卡自检丢失了初始化信息时发送。板卡收到此初始化消息时不需要进行初始化寻零操作。

（2）发送帧命令：**0x1A11**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char unloadPortTableNum;

sUnloadPortPara unloadPortParaTable[unloadPortTableNum];

} sPC2Board\_UnloadPortPara;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | unloadPortTableNum | 下料口个数 |
| sUnloadPortPara | unloadPortParaTable[unloadPortTableNum] | 具体的每个下料口的参数信息 |

每个下料口参数信息格式如下:

typedef struct{

unsigned char unload\_port\_num; //下料口号

unsigned char broad\_num; //从控站号

unsigned char unload\_port\_dir; //下料口方向

} sUnloadPortPara;

（4）应答帧命令：**0x9A11**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_UnloadPortPara\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 2.36是否启用急停按钮

（1）发送时间

配置Demo主动发送给板卡.急停按钮仅限于调节多驱电柜软急停时候使用，代码默认使用（1）急停功能。

（2）发送帧命令：**0x1A10**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char BoardID;

unsigned char Button\_Exist;

unsigned char Emergency\_Button;

} sPC2Board\_Emergency\_Button\_cmd

|  |  |  |
| --- | --- | --- |
| **变量类型** | 变量名 | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | BoardID | 板卡站号 |
| unsigned char | Emergency\_Button | 急停按钮值（0：不启用1：启用） |

（4）应答帧命令：**0x9A10**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char BoardID;

unsigned char Emergency\_Button;

} sPC2Board\_ Emergency\_Button\_cmd\_ack；

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 3.0不允许分拣命令

此协议: 自动供包摆轮专用

（1）发送时间

上位机收到相机信号发送给板卡部分包裹不允许分拣的信息

（2）发送帧命令：**0x1B05**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned int16 time\_delay;

unsigned int16 forbid\_unload\_time;

} sPC2Board\_forbid\_unload\_Data;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned int32 | msg\_id | 下料命令序号 |
| unsigned int16 | time\_delay | 从相机触发光电开始到收到下料信息经过的时间 |
| unsigned int16 | forbid\_unload\_time | 不允许分拣的持续时间 |

（4）应答帧命令：**0x9B05**

（5）数据格式定义

typedef struct{

unsigned char cmd[11];

} sPC2Board\_forbid\_unload\_Data\_ACK;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |

### 3.1摆轮单机分拣结果回应

此协议: 自动供包摆轮专用

（1）发送时间

下位机执行下料动作后反馈给上位机

（2）发送帧命令：**0x1B06**

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char msg\_num;

sMsgData msg\_data[msg\_num];

} sPC2Board\_unload\_Result;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | msg\_num | 回应的消息个数 |
| sMsgData | msg\_data[msg\_num] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char cmd\_type; //回应消息的类型(0:正常分拣1 表示 叠件回流 2 表示 不允许供包回流)

Unsigned int16 unload\_port\_id;//分拣口号

} sMsgData

（4）应答帧命令：无需应答

### 3.2供包台是否允许供件信息上传

此协议: 自动供包摆轮专用

（1）发送时间

初始化完成后主动上传一次所有供包台的状态，然后每当供包台允许供件时上传（小于500毫秒的状态改变不上传）

（2）发送帧命令：0x1A07

（3）数据格式定义

typedef struct{

unsigned char cmd[11];

unsigned char msg\_num;

sMsgData msg\_data[msg\_num];

} sPC2Board\_Supply;

|  |  |  |
| --- | --- | --- |
| **变量类型** | **变量名** | **含义** |
| unsigned char | cmd[11] | 详见1.1通讯协议格式 |
| unsigned char | msg\_num | 消息数据个数 |
| sMsgData | msg\_data[msg\_num] | 消息数据部分 |

消息部分数据格式如下:

typedef struct{

unsigned char Feed\_id; //供包机ID

unsigned char AllowSupply; //是否允许供包（0：不允许，1：允许）

} sMsgData

（4）应答帧命令：无需应答

### \*关于重发机制的说明

上位机和下位机重发间隔时间均为300毫秒一次

下位机重新发送时

第一次发送帧序号为0x00 0x00 0x00 0x00

第二次发送帧序号为0x01 0x00 0x00 0x00

第三次发送帧序号为0x02 0x00 0x00 0x00

上位机重发时帧序号不变，但是需要重新计算包裹延迟时间

|  |  |
| --- | --- |
| 版本 | 更新说明 |
| V1.0.04 | 增加到靠边区时间（3.3节） |
| V1.0.05 | 增加关闭下料口功能（3.4节）  变更分拣结果回应协议（2.5节） |
| V1.0.06 | 细化报警信息（2.6.1节,2.6.2节） |
| V1.0.07 | 新增可设置BOOT参数（3.1节,3.2节） |
| V1.0.08 | 需改增加2.6.3~2.6.4节无O带驱动器报警相关 |
| V1.0.09 | 新增2.4分拣命令photo\_trigger\_time字段 |
| V1.0.10 | 新增3.5双驱动器调速命令 |
| V1.0.11 | 新增3.6双驱动器获取速度命令  新增3.7摆轮线整体运行状态查询  BOOT参数无需进入测试模式即可读写 |
| V1.0.12 | 新增3.8 线体启停控制,新增分拣结果：下料失败 |
| V1.0.13 | 新增2.6.4摆轮报警2,新增 3.6驱动器速度查询3.7启停查询在状态改变时主动上报上位机 |
| V1.0.14 | 新增2.2.1 初始化摆轮信息；2.6.4新增接近开关报警；3.6驱动器速度反馈 具体到每一台摆轮的驱动器速度；3.1、 3.2 读写摆轮参数具体到读写具体某一台摆轮的参数 |
| V1.0.15 | 新增3.9 摆轮急停状态查询 |
| V1.0.16 | 新增2.2.2最小间隔设置命令，新增分拣结果：间隔过小 |
| V1.0.17 | 新增分拣结果：从板执行分拣动作超时 |
| V1.0.18 | 新增间距过小的包裹检测结果输出 |
| V1.0.19 | 新增包裹长度（遮挡时间）输出 |
| V1.0.20 | 新增板卡控制模式信息查询上传 |
| V1.0.21 | 新增包裹补格口数据上传上位机 |
| V1.0.22 | 新增不允许分拣命令和单机分拣结果命令 |
| V1.0.23 | 1. 新增板卡版本号输出 2. 新增上位机设置光电滤波时间功能 3. 新增供包台是否允许供件信息上传功能 4. 新增超时包裹数据追加上传功能 5. 包裹间距过小检测增加开关功能 6. 包裹分拣命令和分拣结果上传优化ACK方法 7. 新增从板未执行主板发送动作功能 8. 新增备用光电分拣结果上传功能 9. 新增光电信息初始化功能 |
| V1.0.24 | 1. 新增从板是否备用光电分拣功能 2. 新增格口是否加速分拣功能 |
| V1.0.25 | 1. 分拣时加速功能调整 2. 可以设置和读取板卡内部的驱动器组别和站号 3. 可以设置上位机掉线异常口 |
| V1.0.26 | 1.重新定义了上传和设置板卡boot参数 |
| V1.0.27 | 1.新增板卡自检异常心跳  2.新增板卡丢失初始化信息重新初始化功能  3.掉线异常口配置改动 |
| V1.0.28 | 新增双驱是否加速功能 |
| V1.0.29 | 1. 新增左右摆可以按照设定脉冲摆动 2. 新增机械间隙补偿功能 3. 新增摆轮站号代码可选择功能 |
| V1.0.30 | 1. 新增急停按钮是否启用配置 |
| V2.0.1 | 1.删除2.25光电档位配置  2.左右角度配置优化  3.是否加速功能优化  4.是否加速配置可读取  5.左右角度配置可读取 |