

# **EI400H 高速分选卡编程手册**

# 目录

一、基本方案说明 .....	5
二、基本工作流程 .....	6
三、主要程序流程 .....	6
四、例程 .....	9
五、重点说明 .....	11
5.1、转盘上元件个数限制 .....	11
5.2、元件的安全距离设定 .....	11
5.4、每个拍照的元件都要调用 Plt_SetBlow 指令 .....	11
5.5、编码器方向问题 .....	11
5.6、检测传感器有效沿说明 .....	12
5.7、元件后压缓冲区的延时时间特别说明 .....	13
5.8、脉冲模式与编码器模式的设置注意 .....	13
5.9、吹气方式说明 .....	14
5.10、OK 吹气模式设定说明 .....	15
六、安装调试建议 .....	15
6.1、配线 .....	15
6.2、配置轴参数 .....	16
6.3、调试轴的运动速度 .....	16
6.4、调试检测装置有效沿 .....	17
6.5、调试相机与吹气口的有效电平与触发时长 .....	17
6.5、初步配置分选参数 .....	18
6.6、调试相机和吹气口位置 .....	18
6.7、测量被测元件的大致长度 .....	19
七、分拣指令详细说明 .....	20
指令 1 Plt_SortCameraBlowConfig .....	22
指令 2 Plt_get_SortCameraBlowConfig .....	23
指令 3 Plt_SetBlow .....	24
指令 4 Plt_StartSort .....	25
指令 5 Plt_ClearSort .....	25
指令 6 Plt_SetCameraTriggerCount .....	25
指令 7 Plt_GetCameraTriggerCount .....	25
指令 8 Plt_SetBlowTriggerCount .....	25
指令 9 Plt_GetBlowTriggerCount .....	26
指令 10 Plt_GetSortStatus .....	26
指令 11 Plt_GetSortDistance .....	26
指令 12 Plt_GetWorkpieceLength .....	27
指令 13 Plt_SetSortOutPut .....	27
指令 14 Plt_SetBufferDelayTime .....	27
指令 15 Plt_GetBufferDelayTime .....	28
指令 16 Plt_GetAbnormalWorkpieceCount .....	28
指令 17 Plt_GetWorkPieceLatchPosition .....	28
指令 18 Plt_GetWorkPiecCentralPosition .....	28

指令 19	Plt_SetBlowProtectPos .....	29
指令 20	Plt_GetBlowProtectPos .....	29
指令 21	Plt_GetBlowStatus .....	29
指令 22	Plt_SetBlowMode .....	29
指令 23	Plt_GetBlowMode .....	30
指令 24	Plt_SetOKPiecesCountMode .....	30
指令 25	Plt_GetOKPiecesCountMode .....	30
指令 26	Plt_SetOneCycleMaxCount .....	30
指令 27	Plt_GetOneCycleMaxCount .....	31
指令 28	Plt_GetCurAngle .....	31
指令 29	Plt_GetWorkPiecCentralBufferPosition .....	31
指令 30	Plt_SetCameraOutMap .....	31
指令 31	Plt_GetCameraOutMap .....	32
指令 32	Plt_SetBlowOutMap .....	32
指令 33	Plt_GetBlowOutMap .....	32
指令 34	Plt_SetBlowOutTime .....	32
指令 35	Plt_GetBlowOutTime .....	33
指令 36	Plt_SetCameraOutTime .....	33
指令 37	Plt_GetCameraOutTime .....	33
指令 38	Plt_SetBlowOutLogic .....	33
指令 39	Plt_GetBlowOutLogic .....	34
指令 40	Plt_SetCameraOutLogic .....	34
指令 41	Plt_GetCameraOutLogic .....	34
指令 42	Plt_GeLatchRepeatPos .....	34
指令 43	Plt_SetBlowExtern .....	35
八、其它通用指令（部分）说明 .....	36	
指令 1	Plt_CardOpen .....	37
指令 2	Plt_CardClose .....	38
指令 3	Plt_CardReset .....	38
指令 4	Plt_AxSetPulseOutMode .....	38
指令 5	Plt_AxSetPPU .....	39
指令 6	Plt_AxGetMotionStatus .....	39
指令 7	Plt_AxSetvelParms .....	39
指令 8	Plt_AxMoveVel .....	40
指令 9	Plt_AxMoveRel .....	40
指令 10	Plt_AxMoveAbs .....	40
指令 11	Plt_AxChangeVel .....	41
指令 12	Plt_IoReadInputByBit .....	41
指令 13	Plt_IoWriteOutputByBit .....	41
指令 14	Plt_IoReadOutputByBit .....	41
指令 15	Plt_IoReadAllInput .....	42
指令 16	Plt_IoReadAllOutput .....	42
指令 17	Plt_IoWriteAllOutput .....	42
指令 18	Plt_AxSetsvonPort .....	42

指令 19	Plt_AxSetEncoderInMode .....	43
指令 20	Plt_AxSetEncoderPosition .....	43
指令 21	Plt_AxGetEncoderPosition .....	44
指令 22	Plt_AxMotionStop .....	44
指令 23	Plt_CardMotionEmgStop .....	44
指令 24	Plt_AxConfigLatchParms .....	44
指令 25	Plt_AxReadLatchParms .....	44
指令 26	Plt_AxGetLatchFlagStatus .....	45
指令 27	Plt_AxClearLatchStatus .....	45
指令 28	Plt_AxGetLacthPosition .....	45
指令 29	Plt_AxGetLatchNum .....	46
指令 30	Plt_AxConfigSpecialInputParms .....	46
指令 31	Plt_AxReadSpecialInputParms .....	46
指令 32	Plt_AxReadSpecialInputStatus .....	46
指令 33	Plt_AxGetsvonPort .....	47
指令 34	Plt_AxGetRdyPort .....	47
指令 35	Plt_AxSetErcPort .....	47
指令 36	Plt_AxGetErcPort .....	47
指令 37	Plt_AxSetCmmandPosition .....	48
指令 38	Plt_AxGetCmmandPosition .....	48
指令 39	Plt_SetHostWatchDog .....	48
指令 40	Plt_GetHostWatchDog .....	48
指令 41	Plt_FeedHostWatchDog .....	49
指令 42	Plt_InitHostWatchDog .....	49
指令 43	Plt_SetDiCountMode .....	49
指令 44	Plt_GetDiCountMode .....	49
指令 45	Plt_SetDlInitCountValue .....	50
指令 46	Plt_GetDiCountValue .....	50
指令 47	Plt_set_none_ok_blow_mode .....	50
指令 48	Plt_get_none_ok_blow_mode .....	50
指令 49	Plt_set_output_enable .....	51
指令 50	Plt_get_output_enable .....	51
指令 51	Plt_SetCameraLatchPos .....	51
指令 52	Plt_GetCameraLatchPos .....	52
指令 53	Plt_SetBlowLatchPos .....	52
指令 54	Plt_GetBlowLatchPos .....	52
九、接线说明 .....	53	
9.1 接线图 .....	53	
十、错误码（函数返回值） .....	56	

# 一、基本方案说明

1、视觉高速分拣方案如下图所示

元件经过检测装置自动进行检测，自动触发相机进行拍照，根据上位机提供的图像识别结果，自动控制吹气口吹气。

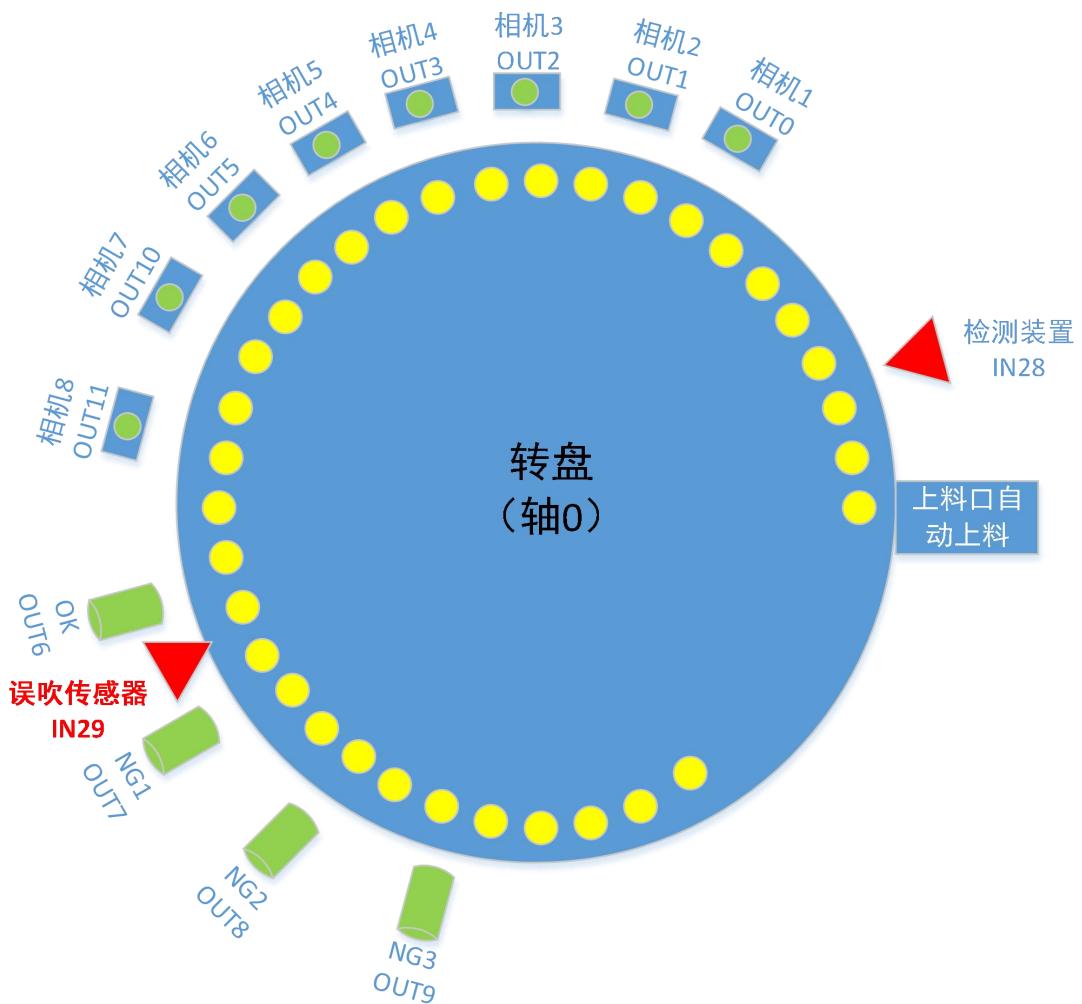


图 1.1 单盘分选卡方案

2、EI400H 支持双系统分选。第一个系统进料检测装置为 IN28, 转盘轴为轴 0；第二个系统进料检测装置为 IN29, 转盘轴为轴 2。

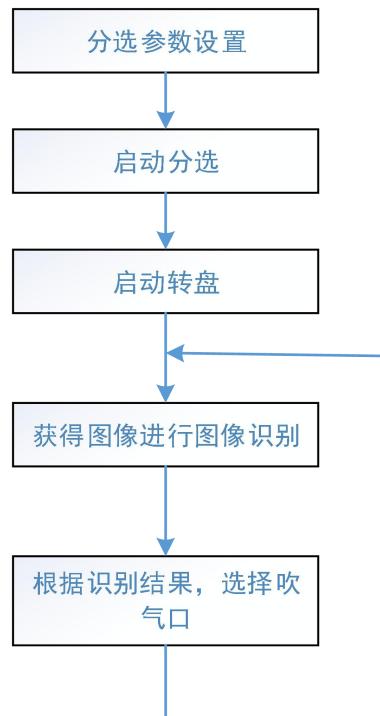
3、EI400H 相机拍照输出口和吹气输出口不是固定的，通过映射函数配置。拍照输出口映射配置函数是 P1t\_SetCameraOutMap，吹气输出口映射配置函数是 P1t\_SetBlowOutMap。输出口 out0 到 out23 通过以上两个函数映射为拍照输出口或者吹气输出口，可以是 24 个都是拍照输出口，也可以 24 个都是吹气输出口。

## 二、基本工作流程

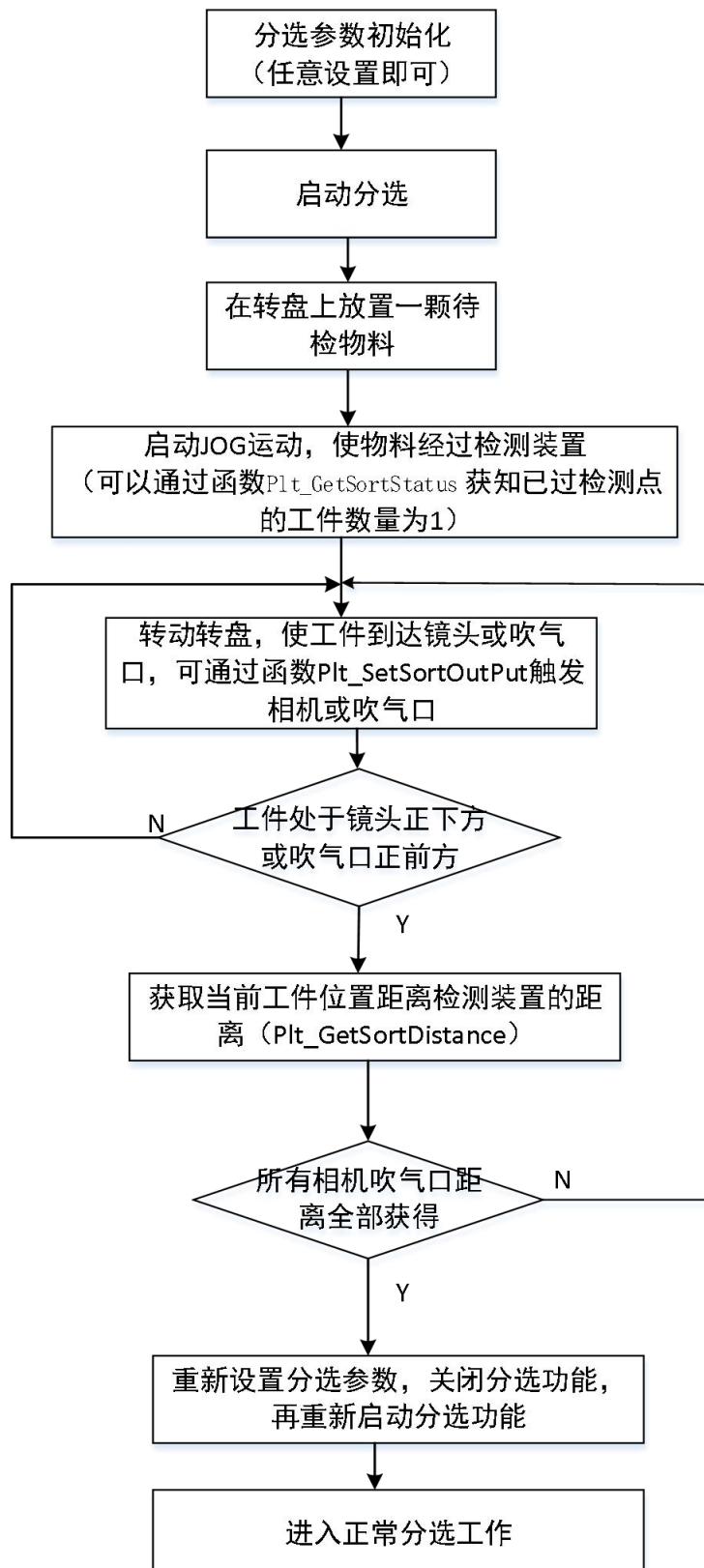
- 1) 启动前的准备。
  - a) 设置检测装置的有效触发沿，上升沿还是下降沿有效
  - b) 设置相机数量及各相机到检测装置的距离
  - c) 设置吹气口数量及各吹气口到检测装置的距离
  - d) 设置相机触发电平及触发保持时间
  - e) 设置吹气口触发电平及触发保持时间
  - f) 设置工件大小参数（最大宽度、最小宽度、最小脉冲间距、最长时间间距）
- 2) 启动分选功能
- 3) 启动转盘运动
- 4) 根据视觉识别的结果决定使用哪个吹气口，调用 Plt\_SetBlow 设置对应吹气口即可。  
不断循环步骤4持续根据视觉结果调用Plt\_SetBlow选择吹气口，元件到达之后控制卡自动控制吹气。

## 三、主要程序流程

正常工作时程序流程如下图：



调试相机和吹气口到检测装置的距离时的流程如下：



调用函数 `Plt_GetSortDistance` 获取的距离是当前位置到检测装置的准确距离，只要使元件准确处于相机下方，这个值基本是不用调整的。

当然，如果需要调整，在使用中也可以调用分选参数设置函数进行微调。



## 四、例程

```
Int main()
{
    struct_sort_config_parms sort_config_params;
    struct_sort_status sort_status;
    uint32 camera_count_last = 0;

    //初始化控制卡
    .....

    //设置分选配置参数
    sort_config_params.check_logic = 0;    //检测装置以上升沿作为触发沿（0 表示上升沿）

    sort_config_params.camera_count = 6;    //设置使用 6 个相机
    sort_config_params.camera_time = 4;    //设置相机触发时间（单位 500us），4 表示 2ms
    sort_config_params.camera_reverse = 0;  //相机触发时输出低电平

    sort_config_params.blow_count = 4;      //设置使用 4 个吹气口
    sort_config_params.blow_time = 2;       //设置吹气口触发时间（单位 500us），2 表示 1ms;
    sort_config_params.blow_reverse = 1;    //吹气口触发时输出高电平

    sort_config_params.width_max = 400;    //设置元件的最大宽度（单位：脉冲），超过这个宽度的过滤掉，不予处理
    sort_config_params.width_min = 100;    //设置元件的最小宽度（单位：脉冲），小于这个宽度的过滤掉，不予处理

    //当两个元件之间的脉冲距离或者时间距离小于设置最小间距，后面的这个元件都不予处理。
    sort_config_params.distance_min = 150; //设置两元件之间的最小间距（单位：脉冲）,
    sort_config_params.time_min= 10;     //设置两元件之间的最小间距（单位：500us），10 表示 5ms;

    uint32 camera_pos [8];
    uint32 blow_pos[4];
    //设置各个相机相对检查点的距离
    camera_pos[0] =1000;
    camera_pos[1] =2000;
    camera_pos[2] =3000;
    camera_pos[3] =4000;
    camera_pos[4] =5000;
    camera_pos[5] =6000
```

```

//设置各个吹气口相对检查点的距离
blow_pos [0] =10000;
blow_pos [1] =11000;
blow_pos [2] =12000;
blow_pos [3] =13000

//调用分选配置参数设置函数
Plt_SortCameraBlowConfig(cardid, sort_config_params, camera_pos, blow_pos);

//启动分选
Plt_StartSort(cardid);

//启动转盘（相关速度设置此处省略了）
//注意，一定要保证编码器是正向计数
Plt_AxMoveVel(cardid, 0, 1);

While(1)
{
    if( 获得一个元件的图像识别结果)
    {
        //假设 n 表示图像识别结果
        //n=1 表示 OK, n=2 表示 NG1, n=3 表示 NG2,
        Switch (n)
        {
            Case 0:
                Plt_SetBlow(cardid,0); //对应元件不触发吹气口
            Case 1:
                Plt_SetBlow(cardid,1); //对应元件触发吹气口1 (OK)
            Case 2:
                Plt_SetBlow(cardid,2); //对应元件触发吹气口 2 (NG1)
            Case 3:
                Plt_SetBlow(cardid,3); //对应元件触发吹气口 3 (NG2)
            Default:
                Plt_SetBlow(cardid,0);
        }
    }
}

}

```

## 五、重点说明

### 5.1、转盘上元件个数限制

玻璃盘上的元件个数（从进料传感器到最后一个 NG 吹气之间）的数量不要大于 1000

### 5.2、元件的安全距离设定

元件最小间距（脉冲）和元件最小间距（时间），元件间距小于这两个参数的任何一个都会被判为无效元件，不予处理。

压缓冲区的延时时间参数设置为元件最小间距（时间）。

例如元件最小间距（时间）为 10ms 时，同时设置“压缓冲区的延时时间”为 10ms。

元件最小间距（脉冲）和元件最小间距（时间）的设定函数

Plt\_SortCameraBlowConfig

压缓冲区的延时时间参数的设定函数

Plt\_SetBufferDelayTime

### 5.4、每个拍照的元件都要调用 Plt\_SetBlow 指令

每个进行拍照的元件都要指定吹气口，如果不需要实际吹气，就指定吹气口为 0。

### 5.5、编码器方向问题

转动方向正确时，需要保证编码器一定是正向计数。

如果编码器是负向计数，可以通过下面的方法进行调整：

方法 1：将接头的 EA+与 EB+互换，EA-与 EB-互换。

序	名称	I/O	说 明	序	名称	I/O	说 明
1	24V_GND	0	24V 输出电源地	14	+24V	0	24V 电源输出
2	ALM	I	伺服报警信号	15	CLEAR	0	驱动报警清除
3	SEVRON	0	伺服使能输出	16	INP	I	驱动到位信号
4	EA-	I	编码器输入 A-	17	EA+	I	编码器输入 A+
5	EB-	I	编码器输入 B-	18	EB+	I	编码器输入 B+
6	EZ-	I	编码器输入 Z-	19	EZ+	I	编码器输入 Z+
7	+5V	0	内部 5V 电源输出	20	5V_GND	0	内部 5V 输出电源地
8	保留	/	/	21	5V_GND	0	内部 5V 输出电源地
9	DIR+	0	方向信号+	22	DIR-	0	方向信号-
10	5V_GND	0	内部 5V 输出电源地	23	PUL+	0	脉冲信号+
11	PUL-	0	脉冲信号-	24	5V_GND	0	内部 5V 输出电源地
12	保留	/	/	25	保留	/	/
13	5V_GND	0	内部 5V 输出电源地				

方法 2：修改伺服的编码器输出方向，以松下 A5 伺服为例。

Pr0.12 原来是 0，则修改为 1，原来是 1，则修改为 0.

Pr0.12 *	脉冲输出逻辑反转			设定范围	单位	标准出厂设定	关联模式																					
	0 ~ 3	—	0 P S T F																									
设置脉冲输出的 B 相逻辑和输出源。通过本参数可对 B 相脉冲逻辑取反，改变 A 相脉冲和 B 相脉冲的相位关系。																												
全闭环控制时，输出源可选择编码器或光栅尺中的任意一个。全闭环控制之外，可选择编码器。																												
<脉冲输出逻辑反转>																												
<table border="1"> <thead> <tr> <th>Pr0.12</th> <th>B 相逻辑</th> <th>输出源</th> <th>CCW 方向动作时</th> <th>CW 方向动作时</th> </tr> </thead> <tbody> <tr> <td rowspan="2">【0】</td> <td rowspan="2">非反转</td> <td>编码器</td> <td>A 相 </td> <td>A 相 </td> </tr> <tr> <td>光栅尺</td> <td>B 相 </td> <td>B 相 </td> </tr> <tr> <td rowspan="2">1</td> <td rowspan="3">反转</td> <td>编码器</td> <td>A 相 </td> <td>A 相 </td> </tr> <tr> <td>光栅尺</td> <td>B 相 </td> <td>B 相 </td> </tr> </tbody> </table>								Pr0.12	B 相逻辑	输出源	CCW 方向动作时	CW 方向动作时	【0】	非反转	编码器	A 相 	A 相 	光栅尺	B 相 	B 相 	1	反转	编码器	A 相 	A 相 	光栅尺	B 相 	B 相 
Pr0.12	B 相逻辑	输出源	CCW 方向动作时	CW 方向动作时																								
【0】	非反转	编码器	A 相 	A 相 																								
		光栅尺	B 相 	B 相 																								
1	反转	编码器	A 相 	A 相 																								
		光栅尺	B 相 	B 相 																								
<b>注 意</b> 设定值 2、3 仅为全闭环控制时有效。全闭环控制之外请设定为 0、1。																												

## 5.6、检测传感器有效沿说明

```

typedef struct
{
    .....
    uint8 check_logic;      //检测装置有效沿 0: 上升沿有效 1: 下降沿有效
    .....
} struct_sort_config_parms;

typedef struct
{
    .....
    uint8 check_logic;      //检测装置有效沿 0: 上升沿有效 1: 下降沿有效
    .....
}

```

```
.....  
}struct_sort_config_parms_e;//分选配置参数
```

## 5.7、元件后压缓冲区的延时时间特别说明

功能简述：此延时时间用于保证将判断合适的元件压入缓冲区，而不会漏掉。

参数说明：

- a、单位：秒，delay\_time = 0.01，表示延时 10ms。
- b、小于 从检测点到相机 1 所需的时间。例如根据速度和检测点到相机 1 所需时间为 0.2s，可以设置延时时间为 0.1s
- c、大于等于 工件最小时时间间距，例如最小工件间距为 0.005s，可以设置延时时间 0.1s。

函数：

第一组分选的设置和查询函数：

```
Plt_SetBufferDelayTime(uint16 cardid,double delay_time);  
Plt_GetBufferDelayTime(uint16 cardid,double *delay_time);
```

第二组分选的设置和查询函数：

```
Plt_SetBufferDelayTime_e(uint16 cardid,double delay_time);  
Plt_GetBufferDelayTime_e(uint16 cardid,double *delay_time);
```

## 5.8、脉冲模式与编码器模式的设置注意

脉冲模式和编码器模式需要匹配上。

以松下 A5 伺服为例。

如果 A5 伺服的脉冲输入和编码器输出使用出厂默认参数。

对应控制卡的设置

脉冲模式：模式 4 或 5

编码器模式：模式 3（AB 相 4 倍计数）

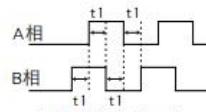
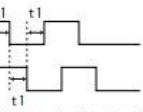
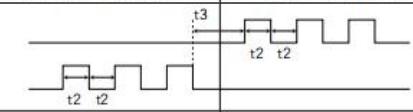
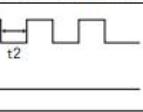
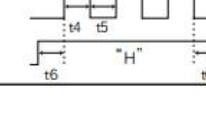
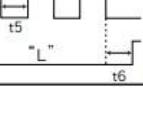
脉冲模式 4 还是 5 的选择方法：

当设置 4 时，如果控制卡控制电机进行正向运动（指令位置增加），同时编码器计数在减小，那就需要将脉冲模式修改为模式 5.

重点说明，一般情况下编码器模式都要选择模式 3，确保指令位置计数增加值与编码器反馈计数增加值一致。指令位置向前运动了 10000 个脉冲，编码器计数也要增加 10000。

松下 A5 伺服的默认设置如下：

### ■指令脉冲的输入形态

Pr0.06 (指令脉冲) 极性设置 设置值	Pr0.07 (指令脉冲) 输入模式设置 设置值	指令脉冲形式	信号名称	正方向指令	负方向指令
【0】	0 或者 2	90° 相位差 2相脉冲 (A相+B相)	PULS SIGN	A相 B相 B相比A相超前90° 	
	【1】	正方向脉冲序列 + 负方向脉冲序列	PULS SIGN		
	3	脉冲序列 + 符号	PULS SIGN		

## 5.9、吹气方式说明

功能说明：

一般情况下，吹气时间长度由固定时间确定，每次吹气都是相同的时长。

吹气时间长度也可以不由时间确定，可以由距离确定，此时由编码器计数值来确定吹气时间长度。例如吹气 100 个编码器计数，当编码器速度为 1K/s 时，吹气时长就是 0.1s；当编码器速度额为 10K/s，吹气时长就是 0.01s。

设置吹气保持模式 `blow_mode=0`: 吹气保持设定时间   `blow_mode=1` 吹气保持设定编码器计数个数

第一组分选所用函数：

`Plt_SetBlowMode(uint16 cardid,uint16 blow_num,uint16 blow_mode);`

`Plt_GetBlowMode(uint16 cardid,uint16 blow_num,uint16 *blow_mode);`

第二组分选所用函数：

`Plt_SetBlowMode_e(uint16 cardid,uint16 blow_num,uint16 blow_mode);`

`Plt_GetBlowMode_e(uint16 cardid,uint16 blow_num,uint16 *blow_mode);`

参数说明：

`blow_num:` 吹气口编号，0-23 //根据实际映射的吹气口对应

`blow_mode:` 吹气时长定义模式选择

`blow_mode=0:` 吹气保持设定时间

`blow_mode=1:` 吹气保持设定编码器计数个数

## 5.10、OK 吹气模式设定说明

功能说明：

通常情况下，判断为 OK 的元件通过一个固定的吹气口输出。

某些情况下，判断为 OK 的元件需要通过两个吹气口轮流输出；例如 OUT8 输出 1000 个 OK 元件，OUT9 输出接下来的 1000 个元件，然后继续由 OUT8 输出 1000 个元件。这样方便在分选的过程中同时进行包装。

第一组分选所用函数：

```
Plt_SetOKPiecesCountMode(uint16 cardid,uint16 mode,uint32 max_count);
```

```
Plt_GetOKPiecesCountMode(uint16 cardid,uint16 *mode,uint32 *max_count);
```

第二组分选所用函数：

```
Plt_SetOKPiecesCountMode_e(uint16 cardid,uint16 mode,uint32 max_count);
```

```
Plt_GetOKPiecesCountMode_e(uint16 cardid,uint16 *mode,uint32 *max_count);
```

参数说明：

mode:

mode=0：（默认模式）一个 OK 料盒计数没限制。

mode=1：两个OK料盒，当其中一个OK料盒计数满后，切换到另外一个料盒。此种模式下，OK使用每组分选功能的前两个吹气口（具体的物理吹气口通过Plt\_SetBlowOutMap函数指定），两个吹气口轮流吹气次数由max\_count定义

mode=2：一个 OK 料盒计数（max\_count 定义）满之后暂停，此种模式下 OK 料盒固定在第一个吹气口（第一组使用 OUT8，第二组使用 OUT20）

max\_count:

mode 设置为 1 或 2 时，max\_count 生效。

其它说明：

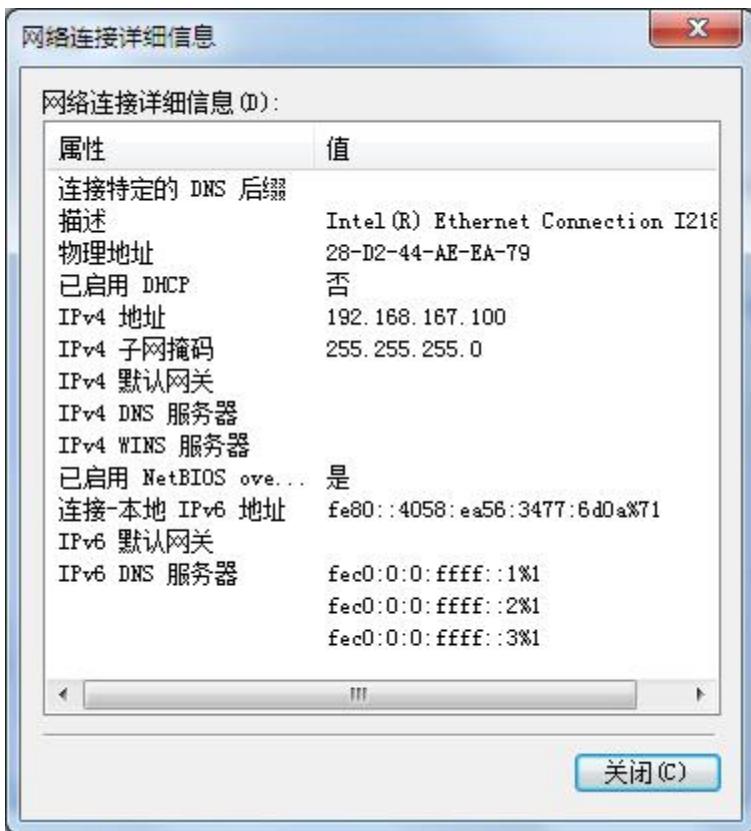
可以通过 Plt\_GetBlowTriggerCount 函数了解各吹气口触发计数，来判断包装计数是否满了。

## 六、安装调试建议

### 6.1、配线

参考第九章进行接线

设置 PC 的 IP 地址为 192.168.167.100(192.168.167.\*\*都可以，不要使用 192.168.167.120，这是控制卡的默认 IP)



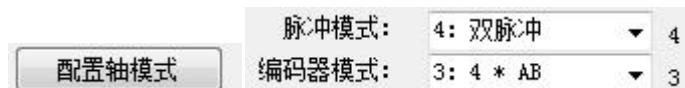
如果使用调试软件进行调试，请打开调试软件，点击“”，建立连接之后可以进行下面的配置

## 6.2、配置轴参数

配置完的结果：

控制轴进行正向运动 10000 步，转盘朝需要的方向转动，编码器计数也增加 10000.

注意：伺服电机要先使能伺服。



## 6.3、调试轴的运动速度

步骤 1、设置轴的速度，注意速度单位是 脉冲/ms。

步骤 2、启动轴进行速度运动。

步骤 3、如果速度合适，停止轴运动进入步骤 4

如果速度不合适，停止轴运动，回到步骤 1

步骤 4、记录速度值

可以使用调试软件



正转/反转：按下启动启动转盘进行速度运动，弹起停止运动

启动转盘：启动转盘进行正向速度运动

停止转盘：停止转盘运动

## 6.4、调试检测装置有效沿

没有元件在光纤处时，记录光纤传感器输入（IN28 或 IN29）口电平 level1。

元件在光纤处时，记录光纤传感器输入（IN28 或 IN29）口电平 level2。

level1 与 level2 不同，说明光纤传感器工作正常

如果 level2 是低电平（输入口状态为 0），则元件进入光纤区域时，IN28 或 IN29 会从高电

平 1 变化到低电平 0。此时需要设置检测装置有效沿为下降沿有效（check\_logic = 1）

如果 level2 是高电平（1），则设置检测装置有效沿为上升沿有效（check\_logic = 0）

检测 (IN28) 口状态 1:高电平      检测 (IN29) 口状态 1:高电平

参数所属结构体：

```
typedef struct
{
    .....
    uint8 check_logic; //检测装置有效沿 0: 上升沿有效 1: 下降沿有效
    .....
}struct_sort_config_parms;//分选配置参数
```

## 6.5、调试相机与吹气口的有效电平与触发时长

调试相机输出口，先接好线。

设置相机电平和时长。

相机数量:	8	相机时间:	2	单位: 500us
相机电平:	0: 输出低电平			

设置吹气电平和时长。

吹气时间:	40	单位: 12.5us
吹气电平:	0: 输出低电平	

点击下图中的按钮，触发各个吹气口输出，如果电平不对，修改上面的吹气电平。

8 吹气0到检测点距离	90000	set	8	清零	0:时间	2
9 吹气1到检测点距离	95000	set	9	清零	0:时间	2
10 吹气2到检测点距离	96000	set	10	清零	0:时间	2
11 吹气3到检测点距离	97000	set	11	清零	0:时间	2

## 6.5、初步配置分选参数

此时可以配置分选参数，点击“设置分选参数”进行配置，“get”可以回读是否设置正确了。

## 6.6、调试相机和吹气口位置

启动分选

启动转盘运动

当有效元件元件通过传感器后，过检测点数量会加 1，（下图中检测到一个元件）

过检测点数量： 1

通过正转和反转控制元件到达目标位置（例如相机 1 处）

点击“获取元件到检测点距离”，可以获得检测传感器到相机 0 的距离

0                  6090

点击“相机 0 到检测点距离”旁的“set”按钮，将距离值转移到输入框中

6090

重复上面的方式可以设定各相机和吹气口的距离

设定完成后，需要重新设定分选参数。

## 6.7、测量被测元件的大致长度

状态查询中可以观察到元件的长度

状态查询:

```
分选轴: 停止 分选轴速度: 0
辅助轴1位置: 0 速度: 0
    过检测点数量: 1
    过相机数量1
    吹气数量1
相机计数: 1 1 1 1 -1 1 1 1
吹气计数: 1 0 0 0
锁存位置: 26185
中心位置: 26285
轴0实时位置: 35202
轴0位置反馈: 35202
元件长度= 201
```

# 七、分拣指令详细说明

第一组分选功能指令：

指令	说明
Plt_SortCameraBlowConfig	设置分选参数
Plt_get_SortCameraBlowConfig	读取分选参数
Plt_SetBlow	指定吹气口
Plt_StartSort	启动高速分选功能
Plt_ClearSort	停止高速分选功能
Plt_SetCameraTriggerCount	设置相机触发计数
Plt_GetCameraTriggerCount	读取相机触发计数
Plt_SetBlowTriggerCount	设置吹气口触发计数
Plt_GetBlowTriggerCount	读取吹气口触发计数
Plt_GetSortStatus	获取分选状态
Plt_GetSortDistance	获取当前位置元件位置相对检测装置的距离
Plt_GetWorkpieceLength	获取刚刚检测到的元件的长度
Plt_SetSortOutPut	手动触发相机口或吹气口
Plt_SetBufferDelayTime	设置检测到元件后压缓冲区的延时时间
Plt_GetBufferDelayTime	读取检测到元件后压缓冲区的延时时间
Plt_GetAbnormalWorkpieceCount	读取由于距离太近等导致的异常元件计数值
Plt_GetWorkPieceLatchPosition	读取元件锁存位置值
Plt_GetWorkPiecCentralPosition	读取元件中心点位置值(锁存边沿+元件长度的一半)
Plt_SetBlowProtectPos	设置吹气保护距离，此距离为相对于分选位置锁存的距离
Plt_GetBlowProtectPos	读取吹气保护距离，此距离为相对于分选位置锁存的距离
Plt_SetBlowMode	设置吹气保持模式
Plt_GetBlowMode	读取吹气保持模式
Plt_SetOKPiecesCountMode	设置 OK 料统计模式
Plt_GetOKPiecesCountMode	读取 OK 料统计模式
Plt_SetOneCycleMaxCount	转盘转一圈编码器计数
Plt_GetOneCycleMaxCount	转盘转一圈编码器计数
Plt_GetCurAngle	读取转盘当前角度
Plt_GetWorkPiecCentralBufferPosition	读取缓冲区元件中心点位置值(锁存边沿+元件长度的一半)

Plt_SetCameraOutMap	拍照输出口映射配置
Plt_GetCameraOutMap	读取当前配置的拍照输出口映射
Plt_SetBlowOutMap	吹气输出口映射配置
Plt_GetBlowOutMap	读取当前配置的吹气输出口映射
Plt_GetBlowStatus	获取包含多吹、少吹及下发指令状态的所有吹气口的详细信息
Plt_SetBlowOutTime	在线调整吹气时间
Plt_GetBlowOutTime	读取吹气时间
Plt_SetCameraOutTime	在线调整拍照时间
Plt_GetCameraOutTime	读取拍照时间
Plt_SetBlowOutLogic	在线调整吹气有效电平
Plt_GetBlowOutLogic	读取吹气有效电平
Plt_SetCameraOutLogic	在线调整拍照有效电平
Plt_GetCameraOutLogic	读取拍照有效电平
Plt_GeLatchRepeatPos	读取没有开分选时元件过正式分选口锁存到的元件开始边沿和结束边沿位置
Plt_SetBlowExtern	指定吹对应序号

## 第二组分选功能指令：

指令	说明
Plt_SortCameraBlowConfig_e	设置分选参数
Plt_get_SortCameraBlowConfig_e	读取分选参数
Plt_SetBlow_e	指定吹气口
Plt_StartSort_e	启动高速分选功能
Plt_ClearSort_e	停止高速分选功能
Plt_SetCameraTriggerCount_e	设置相机触发计数
Plt_GetCameraTriggerCount_e	读取相机触发计数
Plt_SetBlowTriggerCount_e	设置吹气口触发计数
Plt_GetBlowTriggerCount_e	读取吹气口触发计数
Plt_GetSortStatus_e	获取分选状态
Plt_GetSortDistance_e	获取当前位置元件位置相对检测装置的距离
Plt_GetWorkpieceLength_e	获取刚刚检测到的元件的长度
Plt_SetSortOutPut_e	手动触发相机口或吹气口
Plt_SetBufferDelayTime_e	设置检测到元件后压缓冲区的延时时间

Plt_GetBufferDelayTime_e	读取检测到元件后压缓冲区的延时时间
Plt_GetAbnormalWorkpieceCount_e	读取由于距离太近等导致的异常元件计数值
Plt_GetWorkPieceLatchPosition_e	读取元件锁存位置值
Plt_GetWorkPiecCentralPosition_e	读取元件中心点位置值(锁存边沿+元件长度的一半)
Plt_GetWorkPiecCentralBufferPosition_e	读取缓冲区元件中心点位置值(锁存边沿+元件长度的一半)
Plt_SetBlowProtectPos_e	设置吹气保护距离，此距离为相对于分选位置锁存的距离
Plt_GetBlowProtectPos_e	读取吹气保护距离，此距离为相对于分选位置锁存的距离
Plt_SetBlowMode_e	设置吹气保持模式
Plt_GetBlowMode_e	读取吹气保持模式
Plt_SetOKPiecesCountMode_e	设置 OK 料统计模式
Plt_GetOKPiecesCountMode_e	读取 OK 料统计模式
Plt_SetOneCycleMaxCount_e	转盘转一圈编码器计数
Plt_GetOneCycleMaxCount_e	转盘转一圈编码器计数
Plt_GetCurAngle_e	读取转盘当前角度
Plt_SetCameraOutMap_e	拍照输出口映射配置
Plt_GetCameraOutMap_e	读取当前配置的拍照输出口映射
Plt_SetBlowOutMap_e	吹气输出口映射配置
Plt_GetBlowOutMap_e	读取当前配置的吹气输出口映射
Plt_GetBlowStatus_e	获取包含多吹、少吹及下发指令状态的所有吹气口的详细信息
Plt_SetBlowOutTime_e	在线调整吹气时间
Plt_GetBlowOutTime_e	读取吹气时间
Plt_SetCameraOutTime_e	在线调整拍照时间
Plt_GetCameraOutTime_e	读取拍照时间
Plt_SetBlowOutLogic_e	在线调整吹气有效电平
Plt_GetBlowOutLogic_e	读取吹气有效电平
Plt_SetCameraOutLogic_e	在线调整拍照有效电平
Plt_GetCameraOutLogic_e	读取拍照有效电平
Plt_GeLatchRepeatPos_e	读取没有开分选时元件过正式分选口锁存到的元件开始边沿和结束边沿位置
Plt_SetBlowExtern_e	指定吹对应序号

**备注：第二组分选指令的详细介绍，请参考第一组的对应指令。**

指令 1 Plt\_SortCameraBlowConfig

指令原型	<pre>Plt_SortCameraBlowConfig(uint16 cardid, struct_sort_config_parms sort_config_params, int32 *camera_pos, int32 *blow_pos)</pre>
指令说明	设置分选参数
指令参数	
cardid	卡号
sort_config_params	<pre>typedef struct {     uint16 camera_count; //相机数量, 有效范围: [1, 32]     uint32 camera_time; //相机触发保持时间, 单位为 500us     uint8 camera_reverse; //相机触发电平 0: 输出低电平 1: 输出高电     uint16 blow_count; //吹气装置数量, 有效范围: [1, 32]     uint32 blow_time; //吹气装置触发保持时间, 单位为 500us     uint8 blow_reverse; //吹气装置触发电平 0: 输出低电平 1: 输出高     uint8 check_logic; //检测装置有效沿 0: 上升沿有效 1: 下降沿     uint32 width_max; //工件最大宽度, 单位: 【pulse】     uint32 width_min; //工件最小宽度, 单位: 【pulse】     uint32 distance_min; //工件最小脉冲间距 , 单位: 【pulse】     uint32 time_min; //工件最长时间间距 , 单位: 【500us】 } struct_sort_config_parms;</pre>
camera_pos	相机位置数组
blow_pos	吹气口位置数组
指令返回值	
详细说明	<ol style="list-style-type: none"> <li>1、相机位置数组 camera_pos 的大小要等于或大于相机数量 camera_count</li> <li>2、吹气位置数组 blow_pos 的大小要等于或大于吹气口数量 blow_count</li> <li>3、相机触发保持时间 camera_time 和吹气装置触发保持时间 blow_time 的时间单位是 <b>500us</b>, 例, 设定相机触发保存时间为 2, 则相机口输出电平保持 1ms</li> <li>4、检测装置有效沿 check_logic 根据检测传感器来设置</li> <li>5、触发电平根据相机和吹气口实际情况来设置</li> <li>6、工件最大宽度和最小宽度, 工件在这两个尺寸限度之内被判定为正常元件, 两个尺寸之外的元件被抛弃不做处理</li> <li>7、最小脉冲间距和最长时间间距, 用来限定两个元件之间的最小距离, 如果两个元件之间的间距小于这两个限定值, 则被抛弃</li> <li>8、没有使用的相机和吹气输出口, 不能作为通用输出口使用</li> </ol>

## 指令 2 Plt\_get\_SortCameraBlowConfig

指令原型	<pre>Plt_get_SortCameraBlowConfig(uint16 cardid, struct_sort_config_parms *sort_config_params, int32 *camera_pos, int32 *blow_pos)</pre>
------	--

指令说明	读取分选参数
指令参数	
cardid	卡号
sort_config_params	<p>分选配置参数</p> <pre>typedef struct {     uint16 camera_count; //相机数量，有效范围：[1, 32]     uint32 camera_time; //相机触发保持时间，单位为 500us     uint8 camera_reverse; //相机触发电平 0: 输出低电平 1: 输出高电平     uint16 blow_count; //吹气装置数量，有效范围：[1, 32]     uint32 blow_time; //吹气装置触发保持时间，单位为 500us     uint8 blow_reverse; //吹气装置触发电平 0: 输出低电平 1: 输出高电平     uint8 check_logic; //检测装置有效沿 0: 上升沿有效 1: 下降沿有效     uint32 width_max; //工件最大宽度，单位：【pulse】     uint32 width_min; //工件最小宽度，单位：【pulse】     uint32 distance_min; //工件最小脉冲间距，单位：【pulse】     uint32 time_min; //工件最长时间间距，单位：【500us】 } struct_sort_config_parms;</pre>
camera_pos	相机位置数组
blow_pos	吹气口位置数组
指令返回值	

### 指令 3 Plt\_SetBlow

指令原型	Plt_SetBlow(uint16 cardid, uint16 blow_num)
指令说明	指定吹气口
指令参数	
cardid	卡号
blow_num	<p>吹气口号 1-24代表吹气口1-24, 根据实际映射吹气口数对应, 0表示不吹气 注意: 每一个拍照的元件都要指定吹气口, 不需要吹气的就设置吹气口为0</p>
指令返回值	
详细说明	<p>假设 OK 的元件要吹到吹气口 1 NG1 的元件吹到吹气口 2 NG2 的元件吹到吹气口 3</p> <p>如果一个元件图像识别判定是 OK 的, 则调用 Plt_SetBlow (卡号, 1) 如果一个元件图像识别判定是 NG1 的, 则调用 Plt_SetBlow (卡号, 2) 如果一个元件图像识别判定是 NG2 的, 则调用 Plt_SetBlow (卡号, 3) 如果一个元件图像识别异常, 或者不需要处理, 则调用 Plt_SetBlow (卡号, 0)</p>

号, 0), 这个元件就抛弃, 到后面收集处进行收集
----------------------------

#### 指令 4 Plt\_StartSort

指令原型	Plt_StartSort(uint16 cardid)
指令说明	启动分选功能
指令参数	
cardid	卡号
指令返回值	

#### 指令 5 Plt\_ClearSort

指令原型	Plt_ClearSort(uint16 cardid)
指令说明	停止分选功能
指令参数	
cardid	卡号
指令返回值	

#### 指令 6 Plt\_SetCameraTriggerCount

指令原型	Plt_SetCameraTriggerCount(uint16 cardid, uint16 camera_num, uint32 camera_trigger_count)
指令说明	设置相机触发计数
指令参数	
cardid	卡号
camera_num	相机号
camera_trigger_count	相机触发计数值
指令返回值	

#### 指令 7 Plt\_GetCameraTriggerCount

指令原型	Plt_GetCameraTriggerCount(uint16 cardid, uint16 camera_num, uint32 *camera_trigger_count, uint16 count)
指令说明	获得相机触发计数
指令参数	
cardid	卡号
camera_num	相机号
camera_trigger_count	相机触发计数值
count	一次连续获得的相机个数
指令返回值	

#### 指令 8 Plt\_SetBlowTriggerCount

指令原型	Plt_SetBlowTriggerCount(uint16 cardid, uint16 blow_num, uint32
------	--

	blow_trigger_count)
指令说明	设置吹气口触发计数
指令参数	
cardid	卡号
blow_num	吹气口号
blow_trigger_count	吹气口触发计数值
指令返回值	

#### 指令 9 Plt\_GetBlowTriggerCount

指令原型	Plt_GetBlowTriggerCount(uint16 cardid, uint16 blow_num, uint32 *blow_trigger_count, uint16 count)
指令说明	获取吹气口触发计数
指令参数	
cardid	卡号
blow_num	吹气口号
blow_trigger_count	吹气口触发计数值
count	一次连续获得的吹气口个数
指令返回值	

#### 指令 10 Plt\_GetSortStatus

指令原型	Plt_GetSortStatus(uint16 cardid, struct_sort_status *sort_status)
指令说明	获取分选状态
指令参数	
cardid	卡号
sort_status	<pre>//分选状态 typedef struct {     uint32 command_blow_count;           //发送吹气指令数量     uint32 piece_cross_camera_count;    //穿过所有相机的工件数量     uint32 piece_find_count;            //已过检测点的工件数量 } struct_sort_status;</pre>
指令返回值	
详细说明	<p>command_blow_count 反馈控制卡收到的 Plt_SetBlow 指令数量</p> <p>piece_cross_camera_count 经过最后一个相机的元件数量</p> <p>piece_find_count 经过检测传感器的元件数量</p>

#### 指令 11 Plt\_GetSortDistance

指令原型	Plt_GetSortDistance(uint16 cardid, int32 *distance)
指令说明	获取当前位置元件位置相对检测装置的距离

指令参数	
cardid	卡号
distance	当前位置元件位置相对检测装置的距离
指令返回值	
详细说明	<p>设置当前位置距离检测装置的距离</p> <p>1、此命令方便在调试中快速获得相机和吹气口的位置参数</p> <p>2、此命令的第一步是保证有一个元件通过检测口，第二步使元件到达指定相机或吹气口下方；然后调用此命令即可</p>

#### 指令 12 Plt\_GetWorkpieceLength

指令原型	Plt_GetWorkpieceLength(uint16 cardid, int32 *length);
指令说明	获取刚刚检测到的元件的长度
指令参数	
cardid	卡号
length	元件的长度
指令返回值	
详细说明	<p>调试时，当一个元件经过了检测传感器之后，可以通过这条指令来了解元件长度（单位是脉冲）</p> <p>1、此命令只返回刚刚经过检测口的元件的长度</p> <p>2、此命令用于在调试中了解元件的尺寸</p> <p>3、当没有元件经过检测口时，将通过返回值报错</p>

#### 指令 13 Plt\_SetSortOutPut

指令原型	Plt_SetSortOutPut(uint16 cardid, uint16 io_num, uint8 logic, uint32 hold_time)
指令说明	手动触发相机口或吹气口
指令参数	
cardid	卡号
io_num	IO口号： 0-5对应相机1到相机6， 10-11对应相机7到相机8 6-9对应吹气口1到吹气口4
logic	有效电平
hold_time	触发时间（单位：500us） 当hold_time=0，时，则一直保持设置的电平， 当hold_time !=0,hold_time后电平翻转回设置前的电平 hold_time = 1，电平保持时间500us hold_time = 4，电平保持时间1ms
指令返回值	
详细说明	调试中，用于手动触发相机和吹气口

#### 指令 14 Plt\_SetBufferDelayTime

指令原型	Plt_SetBufferDelayTime(uint16 cardid, double delay_time)
指令说明	设置检测到元件后压缓冲区的延时时间
指令参数	
cardid	卡号
delay_time	检测到元件后压缓冲区的延时时间 (单位: s)
指令返回值	

#### 指令 15 Plt\_GetBufferDelayTime

指令原型	Plt_GetBufferDelayTime(uint16 cardid, double *delay_time)
指令说明	读取检测到元件后压缓冲区的延时时间
指令参数	
cardid	卡号
delay_time	检测到元件后压缓冲区的延时时间
指令返回值	

#### 指令 16 Plt\_GetAbnormalWorkpieceCount

指令原型	Plt_GetAbnormalWorkpieceCount(uint16 cardid, uint32 *count)
指令说明	读取由于距离太近等导致的异常元件计数值 (通过检测传感器, 而被抛弃不会触发相机的元件数量) )
指令参数	
cardid	卡号
count	由于距离太近等导致的异常元件计数值
指令返回值	

#### 指令 17 Plt\_GetWorkPieceLatchPosition

指令原型	Plt_GetWorkPieceLatchPosition(uint16 cardid, int32 *pos)
指令说明	读取元件锁存位置值
指令参数	
cardid	卡号
pos	元件锁存位置值
指令返回值	

#### 指令 18 Plt\_GetWorkPiecCentralPosition

指令原型	Plt_GetWorkPiecCentralPosition(uint16 cardid, int32 *pos)
指令说明	读取元件中心点位置值(锁存边沿+元件长度的一半)
指令参数	
cardid	卡号
pos	元件中心位置值
指令返回值	

#### 指令 19 Plt\_SetBlowProtectPos

指令原型	Plt_SetBlowProtectPos(uint16 cardid, int32 protect_pos);
指令说明	设置吹气保护距离，此距离为相对于分选位置锁存的距离
指令参数	
cardid	卡号
protect_pos	吹气保护距离
指令返回值	

#### 指令 20 Plt\_GetBlowProtectPos

指令原型	Plt_GetBlowProtectPos(uint16 cardid, int32 *protect_pos);
指令说明	读取吹气保护距离，此距离为相对于分选位置锁存的距离
指令参数	
cardid	卡号
protect_pos	吹气保护距离
指令返回值	

#### 指令 21 Plt\_GetBlowStatus

指令原型	Plt_GetBlowStatus(uint16 cardid, uint32 *blow_more, uint32 *blow_little, uint32 *blow_cmd)
指令说明	读取吹气状态
指令参数	
cardid	卡号
blow_more	多吹指令计数
blow_little	少吹指令计数
blow_cmd	上位机调用吹气指令计数
指令返回值	

#### 指令 22 Plt\_SetBlowMode

指令原型	Plt_SetBlowMode(uint16 cardid, uint16 blow_num, uint16 blow_mode)
指令说明	设置吹气保持模式
指令参数	
cardid	卡号
blow_num	吹气口编码 0--23,根据实际映射吹气口对应
blow_mode	<b>blow_mode=0:</b> 吹气保持设定时间 <b>blow_mode=1</b> 吹气保持设定编码器计数个数
指令返回值	

#### 指令 23 Plt\_GetBlowMode

指令原型	Plt_GetBlowMode(uint16 cardid, uint16 blow_num, uint16 *blow_mode)
指令说明	读取吹气保持模式
指令参数	
cardid	卡号
blow_num	吹气口编码 0--23,根据实际映射吹气口对应
blow_mode	<b>blow_mode=0:</b> 吹气保持设定时间 <b>blow_mode=1</b> 吹气保持设定编码器计数个数
指令返回值	

#### 指令 24 Plt\_SetOKPiecesCountMode

指令原型	Plt_SetOKPiecesCountMode(uint16 cardid, uint16 mode, uint32 max_count)
指令说明	设置 OK 料统计模式
指令参数	
cardid	卡号
mode	<b>mode=0:</b> 一个 OK 料盒计数没限制, <b>mode=1:</b> 两个 OK 料盒, 当其中一个 OK 料盒计数满后, 切换到另外一个料盒 <b>mode=2:</b> 一个 OK 料盒 计数满之后暂停
max_count	<b>mode=1和mode=2模式下</b> , 料盒计数满时的计数值
指令返回值	

#### 指令 25 Plt\_GetOKPiecesCountMode

指令原型	Plt_GetOKPiecesCountMode(uint16 cardid, uint16 *mode, uint32 *max_count)
指令说明	读取 OK 料统计模式
指令参数	
cardid	卡号
mode	<b>mode=0:</b> 一个 OK 料盒计数没限制, <b>mode=1:</b> 两个 OK 料盒, 当其中一个 OK 料盒计数满后, 切换到另外一个料盒 <b>mode=2:</b> 一个 OK 料盒 计数满之后暂停
max_count	<b>mode=1和mode=2模式下</b> , 料盒计数满时的计数值
指令返回值	

#### 指令 26 Plt\_SetOneCycleMaxCount

指令原型	Plt_SetOneCycleMaxCount(uint16 cardid, uint32 max_count)
------	--

指令说明	设置转盘转一圈编码器计数
指令参数	
cardid	卡号
max_count	转盘转一圈编码器计数
指令返回值	

#### 指令 27 Plt\_GetOneCycleMaxCount

指令原型	Plt_GetOneCycleMaxCount (uint16 cardid, uint32 *max_count)
指令说明	读取转盘转一圈编码器计数
指令参数	
cardid	卡号
max_count	转盘转一圈编码器计数
指令返回值	

#### 指令 28 Plt\_GetCurAngle

指令原型	Plt_GetCurAngle (uint16 cardid, float *max_count)
指令说明	读取转盘当前角度
指令参数	
cardid	卡号
max_count	转盘当前角度 (0--360)
指令返回值	

#### 指令 29 Plt\_GetWorkPiecCentralBufferPosition

指令原型	Plt_GetWorkPiecCentralBufferPosition (uint16 cardid, uint16 pos_num, uint16 *act_pos_num, int32 *pos);
指令说明	读取缓冲区元件中心点位置值(锁存边沿+元件长度的一半)
指令参数	
cardid	卡号
pos_num	读取的个数, 当 pos_num 小于等于 1 时,都是 1 个值
act_pos_num	实际返回有效个数 当 pos_num 大于缓冲区有效个数时, act_pos_num 返回缓冲区有效个数
pos	元件中心点位置值
指令返回值	

#### 指令 30 Plt\_SetCameraOutMap

指令原型	Plt_SetCameraOutMap (uint16 cardid, uint16 cameracount, uint16 *camera_out_map);
------	--

指令说明	设置相机对应 IO 口映射
指令参数	
cardid	卡号
cameracount	相机数量
camera_out_map	相机对应映射输出口
指令返回值	

#### 指令 31 Plt\_GetCameraOutMap

指令原型	Plt_GetCameraOutMap(uint16 cardid, uint16 *cameracount, uint16 *camera_out_map);
指令说明	读取相机对应 IO 口映射
指令参数	
cardid	卡号
cameracount	相机数量
camera_out_map	相机对应映射输出口
指令返回值	

#### 指令 32 Plt\_SetBlowOutMap

指令原型	Plt_SetBlowOutMap(uint16 cardid, uint16 blowcount, uint16 *blow_out_map);
指令说明	设置吹气口对应 IO 口映射
指令参数	
cardid	卡号
blowcount	吹气口数量
blow_out_map	吹气口对应映射输出口
指令返回值	

#### 指令 33 Plt\_GetBlowOutMap

指令原型	Plt_GetBlowOutMap(uint16 cardid, uint16 *blowcount, uint16 *blow_out_map);
指令说明	读取吹气口对应 IO 口映射
指令参数	
cardid	卡号
blowcount	吹气口数量
blow_out_map	吹气口对应映射输出口
指令返回值	

#### 指令 34 Plt\_SetBlowOutTime

指令原型	Plt_SetBlowOutTime(uint16 cardid, uint16 blow_index, uint32 blow_time);
------	---

指令说明	在线调整吹气时间
指令参数	
cardid	卡号
blow_index	吹气号
blow_time	吹气装置触发保持时间，单位为 500us
指令返回值	

#### 指令 35 Plt\_SetBlowOutTime

指令原型	Plt_SetBlowOutTime(uint16 cardid, uint16 blow_index, uint32 *blow_time);
指令说明	读取吹气时间
指令参数	
cardid	卡号
blow_index	吹气号
blow_time	吹气装置触发保持时间，单位为 500us
指令返回值	

#### 指令 36 Plt\_SetCameraOutTime

指令原型	Plt_SetCameraOutTime(uint16 cardid, uint16 camera_index, uint32 camera_time);
指令说明	在线调整拍照时间
指令参数	
cardid	卡号
camera_index	相机号
camera_time	camera_time 相机触发保持时间，单位为 500us
指令返回值	

#### 指令 37 Plt\_GetCameraOutTime

指令原型	Plt_GetCameraOutTime(uint16 cardid, uint16 camera_index, uint32 *camera_time);
指令说明	读取拍照时间
指令参数	
cardid	卡号
camera_index	相机号
camera_time	camera_time 相机触发保持时间，单位为 500us
指令返回值	

#### 指令 38 Plt\_SetBlowOutLogic

指令原型	Plt_SetBlowOutLogic(uint16 cardid, uint16 blow_index, uint16 blow_logic);
------	---

指令说明	在线调整吹气有效电平
指令参数	
cardid	卡号
blow_index	吹气号
blow_logic	吹气装置触发输出电平 0: 低电平 1: 高电平
指令返回值	

#### 指令 39 Plt\_GetBlowOutLogic

指令原型	Plt_GetBlowOutLogic(uint16 cardid, uint16 blow_index, uint16 *blow_logic);
指令说明	读取吹气有效电平
指令参数	
cardid	卡号
blow_index	吹气号
blow_logic	吹气装置触发输出电平 0: 低电平 1: 高电平
指令返回值	

#### 指令 40 Plt\_SetCameraOutLogic

指令原型	Plt_SetCameraOutLogic(uint16 cardid, uint16 camera_index, uint16 camera_logic);
指令说明	在线调整拍照有效电平
指令参数	
cardid	卡号
camera_index	相机序号
camera_logic	相机触发输出电平 0: 低电平 1: 高电平
指令返回值	

#### 指令 41 Plt\_GetCameraOutLogic

指令原型	Plt_GetCameraOutLogic(uint16 cardid, uint16 camera_index, uint16 *camera_logic);
指令说明	读取拍照有效电平
指令参数	
cardid	卡号
camera_index	相机序号
camera_logic	相机触发输出电平 0: 低电平 1: 高电平
指令返回值	

#### 指令 42 Plt\_GeLatchRepeatPos

指令原型	P1t_GeLatchRepeatPos(uint16 cardid, uint16 *ltc_state, uint16 *ltc_cnt, int32 *ltc_pos);
指令说明	读取没有开分选时元件过正式分选口锁存到的元件开始边沿和结束边沿位置
指令参数	
cardid	卡号
ltc_state	锁存状态 2: 锁存完成 1: 锁存进行中 0: 锁存停止 3: 锁存失败
ltc_cnt	锁存个数
ltc_pos	锁存位置
指令返回值	

#### 指令 43 P1t\_SetBlowExtern

指令原型	P1t_SetBlowExtern(uint16 cardid, uint16 blow_num, uint32 blow_index);
指令说明	指定吹对应序号的料
指令参数	
cardid	卡号
blow_num	吹气口号 1-24代表吹气口1-24, 根据实际映射吹气口数对应, 0表示不吹气 注意: 每一个拍照的元件都要指定吹气口, 不需要吹气的就设置吹气口为0
blow_index	料序号, 从 0 开始
指令返回值	

## 八、其它通用指令（部分）说明

指令	说明
Plt_CardOpen	初始化控制卡
Plt_CardClose	关闭控制卡
Plt_CardReset	控制卡硬件复位
Plt_AxSetPulseOutMode	设定脉冲输出模式
Plt_AxSetPPU	设定脉冲当量
Plt_AxGetMotionStatus	读取指定轴的运动状态
Plt_AxSetvelParms	速度参数设置
Plt_AxMoveVel	启动JOG运动
Plt_AxMoveRel	点位运动（相对位置）
Plt_AxMoveAbs	点位运动（绝对位置）
Plt_AxChangeVel	在线变速
Plt_IoReadInputByBit	读取输入口的状态
Plt_IoWriteOutputByBit	设置输出口的状态
Plt_IoReadOutputByBit	读取输出口的状态
Plt_IoReadAllInput	读取输入端口的值
Plt_IoReadAllOutput	读取输出端口的值
Plt_IoWriteAllOutput	设置输出端口的值
Plt_AxSetsvonPort	输出SEVON信号
Plt_AxSetEncoderInMode	设置编码器计数模式
Plt_AxSetEncoderPosition	设置编码器位置
Plt_AxGetEncoderPosition	读取编码器位置
Plt_AxMotionStop	轴停止
Plt_CardMotionEmgStop	所有轴立即停止
Plt_AxConfigLatchParms	锁存方式配置
Plt_AxReadLatchParms	读取锁存方式配置
Plt_AxGetLatchFlagStatus	读取锁存标志
Plt_AxClearLatchStatus	清除锁存标志
Plt_AxGetLacthPosition	读取锁存位置
Plt_AxGetLatchNum	查询已经锁存的数据个数
Plt_AxConfigSpecialInputParms	配置EMG\ERC\INP\ALM等信号和滤波时间
Plt_AxReadSpecialInputParms	读取EMG\ERC\INP\ALM等信号和滤波时间的配置
Plt_AxReadSpecialInputStatus	读取轴专用输入信号状态
Plt_AxGetsvonPort	获取SEVON信号
Plt_AxGetRdyPort	获取RDY信号

Plt_AxSetErcPort	控制ERC信号输出电平
Plt_AxGetErcPort	获取控制ERC信号输出电平
Plt_AxSetCmmandPosition	设置轴的指令位置
Plt_AxGetCmmandPosition	查询轴的指令位置
Plt_SetHostWatchDog	设置看门狗参数
Plt_GetHostWatchDog	读取看门狗参数
Plt_FeedHostWatchDog	在超时报警时间范围内不断调用此函数，以防止超时报警
Plt_InitHostWatchDog	看门狗功能复位，复位后如果需要再次启动看门狗功能，需要重新配置看门狗
Plt_SetDiCountMode	设置输入IO计数模式
Plt_GetDiCountMode	读取输入IO计数模式
Plt_SetDlInitCountValue	设置输入IO计数初始值
Plt_GetDiCountValue	读取输入IO计数值
Plt_set_none_ok_blow_mode	设置是否启用所有非OK料常吹模式
Plt_get_none_ok_blow_mode	读取是否启用所有非OK料常吹模式
Plt_set_output_enable	设置输出口是否启用输出，默认情况下启用输出，此功能在对应的输出口被配置为拍照口或吹气口的条件下有效
Plt_get_output_enable	读取输出口是否启用输出，默认情况下启用输出，此功能在对应的输出口被配置为拍照口或吹气口的条件下有效
Plt_SetCameraLatchPos	配置相机在料的哪个位置拍照
Plt_GetCameraLatchPos	读取配置相机在料的哪个位置拍照
Plt_SetBlowLatchPos	配置吹气口在料的哪个位置吹气
Plt_GetBlowLatchPos	读取吹气口在料的哪个位置吹气

### 指令 1 Plt\_CardOpen

指令原型	Plt_CardOpen(uint16 TotalCards, uint16 *CardIdArray, uint16 *Section, uint16* Host_id)
指令说明	初始化控制卡
指令参数	
TotalCards	卡的数量，只有一张卡，设置为1
CardIdArray	卡的ID列表， 可以设置第一张卡的ID为0，后续通过ID识别卡，通常控制函数的第一个参数就是这个ID号。
Section	IP地址的第三段，默认出厂是167 要求将PC机网口地址设为同一段，例如192.168.167.99
Host_id	IP地址的第四段，默认出厂是120
指令返回值	

**例子**

```

public static UInt16 GcardID = new UInt16();
UInt16 TotalCards = 1;
UInt16[] CardNameArray = new UInt16[8];
CardNameArray[0] = 0;
GcardID = CardNameArray[0];
UInt16[] Section = new UInt16[8];
UInt16[] Host_id = new UInt16[8];
Section[0] = 167;
Host_id[0] = 120;
ret = PLT.Plt_CardOpen(TotalCards, CardNameArray, Section, Host_id);
//开卡后可以通过卡的ID来控制对应卡
PLT.Plt_AxMoveAbs(GcardID, axis, position);

```

**指令 2 Plt\_CardClose**

指令原型	Plt_CardClose(void)
指令说明	关闭控制卡
指令参数	
指令返回值	

**指令 3 Plt\_CardReset**

指令原型	Plt_CardReset(void)
指令说明	控制卡硬件复位
指令参数	
指令返回值	

**指令 4 Plt\_AxSetPulseOutMode**

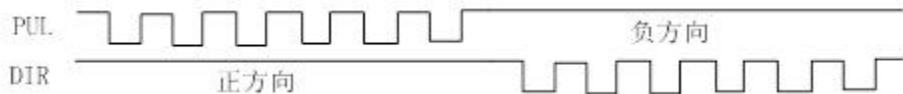
指令原型	Plt_AxSetPulseOutMode(uint16 cardid, uint16 axis, uint16 mode)
指令说明	设定脉冲输出模式，影响指定轴的输出脉冲波形
指令参数	
Card	卡号
Axis	轴号
Mode	脉冲模式，主要脉冲模式见下图 0—脉冲+方向； 4—双脉冲 脉冲+方向模式由0, 1, 2, 3共四种，一般选择0或2，根据实际情况实验确认。 双脉冲有4、5两种。一般选4即可
指令返回值	

备注：

脉冲模式 0 对应的波形



脉冲模式 4 对应的波形



#### 指令 5 Plt\_AxSetPPU

指令原型	Plt_AxSetPPU(uint16 cardid, uint16 axis, double PPU)
指令说明	设置脉冲当量
指令参数	
card	卡号
axis	轴号
PPU	脉冲当量值, 如果不调用此函数, 默认各轴的脉冲当量是1 应用在分选模式时, 不需要设置, 使用默认值1。
指令返回值	

#### 指令 6 Plt\_AxGetMotionStatus

指令原型	Plt_AxGetMotionStatus(uint16 cardid, uint16 axis, uint16 *motionstatus)
指令说明	读取轴运动状态
指令参数	
cardid	卡号
axis	轴号
motionstatus	轴运动状态 (0: 运动中, 1: 停止)
指令返回值	

#### 指令 7 Plt\_AxSetvelParms

指令原型	Plt_AxSetvelParms(uint16 cardid, uint16 axis, struct_vel_plan_parms axisplanparms)
指令说明	设置速度参数
指令参数	
cardid	卡号
axis	轴号
axisplanparms	速度规划参数  typedef struct { double start_vel; //起始速度, 单位: 【ppu/ms】 double max_vel; //最大速度, 单位: 【ppu/ms】 }

	<pre>double end_vel; //停止速度, 单位: 【ppu/ms】 double acc; //加速度,单位: [ppu/ms<sup>2</sup>]; double dec; //减速度,单位: [ppus/ms<sup>2</sup>]; double smooth_factor; //S段比例因子, 范围: [0-1] }struct_vel_plan_parms;//速度规划参数</pre>
指令返回值	
详细说明	<p>注意这里的速度单位是基于 ms 的。正常计算速度时是以秒为单位的，写入时，需要除以 1000，转换为以 ms 为单位。</p> <p>这里的加速度的单位是 ms<sup>2</sup> 的，正常计算速度时是以 s<sup>2</sup> 为单位的，写入时，需要除以 1000000，转换为以 ms<sup>2</sup> 为单位。</p> <p>S 段比例因子尽量使用 0 。</p>

#### 指令 8 Plt\_AxMoveVel

指令原型	Plt_AxMoveVel(uint16 cardid, uint16 axis, uint16 direction)
指令说明	启动 JOG 运动
指令参数	
cardid	卡号
axis	轴号
direction	JOG运动的方向, 0: 负方向, 1: 正方向
指令返回值	

#### 指令 9 Plt\_AxMoveRel

指令原型	Plt_AxMoveRel(uint16 cardid, uint16 axis, double dist)
指令说明	点位运动（相对位置）
指令参数	
cardid	卡号
axis	轴号
Dist	移动距离
指令返回值	

#### 指令 10 Plt\_AxMoveAbs

指令原型	Plt_AxMoveAbs(uint16 cardid, uint16 axis, double position)
指令说明	点位运动（绝对位置）
指令参数	
cardid	卡号
axis	轴号
position	移动目标点位置
指令返回值	

#### 指令 11 Plt\_AxChangeVel

指令原型	Plt_AxChangeVel(uint16 cardid, uint16 axis, double maxvel)
指令说明	启动 JOG 运动
指令参数	
cardid	卡号
axis	轴号
maxvel	新的速度
指令返回值	

注意：可以变速到 0，虽然轴的运动停止下来了，但是轴的状态任然是运动状态，要退出运动状态，必须调用停止指令。

#### 指令 12 Plt\_IoReadInputByBit

指令原型	Plt_IoReadInputByBit(uint16 cardid, uint16 bitno, uint16 *active_level)
指令说明	读取输入口状态（读取单个输入口）
指令参数	
Cardid	卡号
Bitno	输入口编号
active_level	输入口状态
指令返回值	
详细说明	IN28 和 IN29 在分选方案中有特殊用途，不要作为通用输入使用。

#### 指令 13 Plt\_IoWriteOutputByBit

指令原型	Plt_IoWriteOutputByBit(uint16 cardid, uint16 bitno, uint16 active_level)
指令说明	单个输出口输出控制
指令参数	
Cardid	卡号
Bitno	输出口编号
active_level	输出口电平
指令返回值	
详细说明	OUT0-OUT11 在分选方案中有特殊用途，不要作为通用输出使用。

#### 指令 14 Plt\_IoReadOutputByBit

指令原型	Plt_IoReadOutputByBit(uint16 cardid, uint16 bitno, uint16 *active_level)
指令说明	单个输出口状态查询
指令参数	
Cardid	卡号

Bitno	输出口编号
active_level	输出口电平
指令返回值	

#### 指令 15 Plt\_IoReadAllInput

指令原型	Plt_IoReadAllInput (uint16 cardid, DWORD *active_level_1, DWORD *active_level_2)
指令说明	同时获取所有输入口的状态
指令参数	
cardid	卡号
active_level_1	输入口0到31的状态
active_level_2	输入口32到63的状态
指令返回值	

#### 指令 16 Plt\_IoReadAllOutput

指令原型	Plt_IoReadAllOutput (uint16 cardid, DWORD *active_level)
指令说明	一次查询全部的输出口状态
指令参数	
cardid	卡号
active_level	Bit0到bit31分别代表输出口0到31的状态
指令返回值	

#### 指令 17 Plt\_IoWriteAllOutput

指令原型	Plt_IoWriteAllOutput (uint16 cardid, DWORD active_level)
指令说明	手动触发相机口或吹气口
指令参数	
cardid	卡号
active_level	Bit0到bit31分别代表输出口0到31的状态
指令返回值	

#### 指令 18 Plt\_AxSetsvonPort

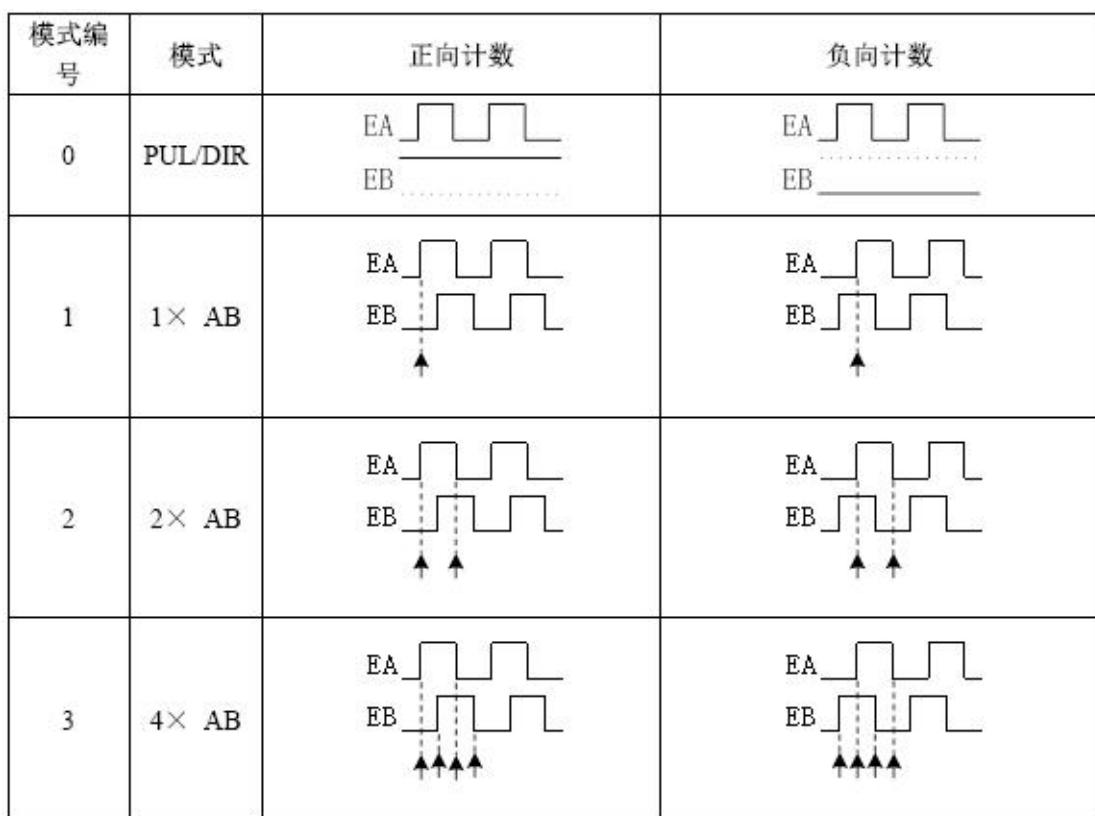
指令原型	Plt_AxSetsvonPort (uint16 cardid, uint16 axis, uint16 active_level)
指令说明	控制指定轴的 Sevon 的输出
指令参数	
cardid	卡号
axis	轴号
active_level	有效电平

指令返回值	
详细说明	分选方案中，只有轴 0 可以正常控制，axis 只能设为 0

### 指令 19 Plt\_AxSetEncoderInMode

指令原型	Plt_AxSetEncoderInMode(uint16 cardid, uint16 axis, uint16 mode)
指令说明	设置编码器计数模式
指令参数	
cardid	卡号
axis	轴号
mode	编码器计数模式（可选择模式0到模式3，详细见下面的备注）
指令返回值	

备注：



### 指令 20 Plt\_AxSetEncoderPosition

指令原型	Plt_AxSetEncoderPosition(uint16 cardid, uint16 axis, double pos)
指令说明	设置编码器位置值
指令参数	
cardid	卡号
axis	编码器编号
pos	编码器位置值
指令返回值	

#### 指令 21 Plt\_AxGetEncoderPosition

指令原型	Plt_AxGetEncoderPosition(uint16 cardid, uint16 axis, double * pos)
指令说明	设置编码器位置值
指令参数	
cardid	卡号
axis	轴号
pos	编码器位置值
指令返回值	

#### 指令 22 Plt\_AxMotionStop

指令原型	Plt_AxMotionStop(uint16 cardid, uint16 axis, uint16 stop_mode)
指令说明	轴停止
指令参数	
cardid	卡号
axis	轴号
stop_mode	停止模式 0: 减速停止 1: 立即停止
指令返回值	

#### 指令 23 Plt\_CardMotionEmgStop

指令原型	Plt_CardMotionEmgStop(uint16 cardid)
指令说明	所有轴立即停止
指令参数	
cardid	卡号
axis	轴号
指令返回值	

#### 指令 24 Plt\_AxConfigLatchParms

指令原型	Plt_AxConfigLatchParms(uint16 cardid, uint16 axis, struct_axis_latch_parms ltcparms)
指令说明	锁存方式配置
指令参数	
cardid	卡号
axis	轴号
ltcparms	<pre>typedef struct {     uint16 active_level;//0: 上升沿有效 1: 下降沿有效     uint16 latch_method;//锁存方式; 0: 单次锁存; 1: 连续锁存     uint16 latch_source;//锁存源; 0: 锁存理论位置; 1: 锁存编码器位置; }struct_axis_latch_parms;//位置锁存配置</pre>
指令返回值	

#### 指令 25 Plt\_AxReadLatchParms

指令原型	Plt_AxReadLatchParms(uint16 cardid, uint16 axis, struct_axis_latch_parms *ltcparms)
指令说明	读取锁存方式配置
指令参数	
cardid	卡号
axis	轴号
ltcparms	<pre>typedef struct {     uint16 active_level;//0: 上升沿有效 1: 下降沿有效     uint16 latch_method;//锁存方式; 0: 单次锁存; 1: 连续锁存     uint16 latch_source;//锁存源; 0: 锁存理论位置; 1: 锁存编码器位置; }struct_axis_latch_parms;//位置锁存配置</pre>
指令返回值	

#### 指令 26 Plt\_AxGetLatchFlagStatus

指令原型	Plt_AxGetLatchFlagStatus(uint16 cardid, uint16 axis, uint16 *latchstatus)
指令说明	读取锁存标志
指令参数	
cardid	卡号
axis	轴号
latchstatus	1:有锁存数据 0: 没有锁存数据
指令返回值	

#### 指令 27 Plt\_AxClearLatchStatus

指令原型	Plt_AxClearLatchStatus(uint16 cardid, uint16 axis)
指令说明	清除锁存标志
指令参数	
cardid	卡号
axis	轴号
指令返回值	

#### 指令 28 Plt\_AxGetLacthPosition

指令原型	Plt_AxGetLacthPosition(uint16 cardid, uint16 axis, long latch_num, double *position)
指令说明	读取锁存位置
指令参数	
cardid	卡号
axis	轴号
latch_num	位置索引 latch_num为-1, 读单次锁存值, latch_num>=0 按索引读锁存

	值
position	锁存位置
指令返回值	

#### 指令 29 Plt\_AxGetLatchNum

指令原型	Plt_AxGetLatchNum(uint16 cardid,uint16 axis, long *num)
指令说明	查询已经锁存的数据个数
指令参数	
cardid	卡号
axis	轴号
num	锁存个数
指令返回值	

#### 指令 30 Plt\_AxConfigSpecialInputParms

指令原型	Plt_AxConfigSpecialInputParms(uint16 cardid,uint16 axis, struct_special_input_parms specinput)
指令说明	配置 EMG\ERC\INP\ALM 等信号和滤波时间
指令参数	
cardid	卡号
axis	轴号
specinput	专用输入配置
指令返回值	

#### 指令 31 Plt\_AxReadSpecialInputParms

指令原型	Plt_AxReadSpecialInputParms(uint16 cardid,uint16 axis, struct_special_input_parms *specinput);
指令说明	读取 EMG\ERC\INP\ALM 等信号和滤波时间的配置
指令参数	
cardid	卡号
axis	轴号
specinput	专用输入配置
指令返回值	

#### 指令 32 Plt\_AxReadSpecialInputStatus

指令原型	Plt_AxReadSpecialInputStatus(uint16 cardid,uint16 axis, struct_special_input_status *inputstatus);
指令说明	读取轴专用输入信号状态
指令参数	
cardid	卡号

axis	轴号
inputstatus	专用输入状态
指令返回值	

### 指令 33 Plt\_AxGetsvonPort

指令原型	Plt_AxGetsvonPort(uint16 cardid,uint16 axis,uint16 *active_level);
指令说明	获取 SEVON 信号
指令参数	
cardid	卡号
axis	轴号
active_level	SERV-ON输出电平值 0:低电平 1: 高电平
指令返回值	

### 指令 34 Plt\_AxGetRdyPort

指令原型	Plt_AxGetRdyPort(uint16 cardid,uint16 axis,uint16 *active_level);
指令说明	获取 RDY 信号
指令参数	
cardid	卡号
axis	轴号
active_level	RDY信号输入值 0:低电平 1: 高电平
指令返回值	

### 指令 35 Plt\_AxSetErcPort

指令原型	Plt_AxSetErcPort(uint16 cardid,uint16 axis,uint16 active_level);
指令说明	控制 ERC 信号输出电平
指令参数	
cardid	卡号
axis	轴号
active_level	ERC信号输出值 0:低电平 1: 高电平
指令返回值	

### 指令 36 Plt\_AxGetErcPort

指令原型	Plt_AxGetErcPort(uint16 cardid,uint16 axis,uint16 *active_level);
指令说明	获取控制 ERC 信号输出电平
指令参数	
cardid	卡号
axis	轴号
active_level	ERC信号输出值 0:低电平 1: 高电平
指令返回值	

### 指令 37 Plt\_AxSetCmmandPosition

指令原型	Plt_AxSetCmmandPosition(uint16 cardid, uint16 axis, double pos);
指令说明	设置轴的指令位置
指令参数	
cardid	卡号
axis	轴号
pos	指令位置值
指令返回值	

### 指令 38 Plt\_AxGetCmmandPosition

指令原型	Plt_AxGetCmmandPosition(uint16 cardid, uint16 axis, double * pos);
指令说明	查询轴的指令位置
指令参数	
cardid	卡号
axis	轴号
pos	指令位置值
指令返回值	

### 指令 39 Plt\_SetHostWatchDog

指令原型	Plt_SetHostWatchDog(uint16 cardid, uint16 enable, uint16 watchtime, uint16 do_total_num, uint16 *do_num, uint16 *do_logic);
指令说明	设置看门狗参数
指令参数	
cardid	卡号
enable	看门狗使能标志 1: 看门狗使能 0: 看门狗禁止 默认是禁止
watchtime	超时报警时间, 单位[ms]
do_total_num	超时报警时需要设置的输出IO口总数
do_num	超时报警时需要设置的输出IO口序号
do_logic	超时报警时需要设置的输出各个IO口电平
指令返回值	

### 指令 40 Plt\_GetHostWatchDog

指令原型	Plt_GetHostWatchDog(uint16 cardid, uint16 *enable, uint16 *watchtime, uint16 *do_total_num, uint16 *do_num, uint16 *do_logic);
指令说明	读取看门狗参数
指令参数	
cardid	卡号
enable	看门狗使能标志 1: 看门狗使能 0: 看门狗禁止 默认是禁止

watchtime	超时报警时间，单位[ms]
do_total_num	超时报警时需要设置的输出IO口总数
do_num	超时报警时需要设置的输出IO口序号
do_logic	超时报警时需要设置的输出各个IO口电平
指令返回值	

#### 指令 41 Plt\_FeedHostWatchDog

指令原型	Plt_FeedHostWatchDog(uint16 cardid);
指令说明	在超时报警时间范围内不断调用此函数，以防止超时报警
指令参数	
cardid	卡号
指令返回值	

#### 指令 42 Plt\_InitHostWatchDog

指令原型	Plt_InitHostWatchDog(uint16 cardid);
指令说明	看门狗功能复位，复位后如果需要再次启动看门狗功能，需要重新配置看门狗函数
指令参数	
cardid	卡号
指令返回值	

#### 指令 43 Plt\_SetDiCountMode

指令原型	Plt_SetDiCountMode(uint16 cardid, uint16 bitno, uint16 count_mode, double filter_time);
指令说明	设置输入 IO 计数模式
指令参数	
cardid	卡号
bitno	输入io号
count_mode	计数模式 0: 禁止输入 IO 计数 1: 上升沿计数 2: 下降沿计数
filter_time	输入IO滤波时间 单位: s 最小值 0.001s
指令返回值	

#### 指令 44 Plt\_GetDiCountMode

指令原型	Plt_GetDiCountMode(uint16 cardid, uint16 bitno, uint16 *count_mode, double *filter_time);
指令说明	读取输入 io 计数模式
指令参数	
cardid	卡号
bitno	输入io号

count_mode	计数模式 0: 禁止输入 IO 计数 1: 上升沿计数 2: 下降沿计数
filter_time	输出IO滤波时间 单位: s 最小值 0.001s
指令返回值	

#### 指令 45 Plt\_SetDiInitCountValue

指令原型	Plt_SetDiInitCountValue(uint16 cardid, uint16 bitno, uint32 init_value);
指令说明	设置输入 IO 计数初始值
指令参数	
cardid	卡号
bitno	输入io号
init_value	输入io计数初始值
指令返回值	

#### 指令 46 Plt\_GetDiCountValue

指令原型	Plt_GetDiCountValue(uint16 cardid, uint16 bitno, uint32 *count_value);
指令说明	读取输入 IO 计数值
指令参数	
cardid	卡号
bitno	输入io号
count_value	当前输入io计数值
指令返回值	

#### 指令 47 Plt\_set\_none\_ok\_blow\_mode

指令原型	Plt_set_none_ok_blow_mode(uint16 cardid, uint16 sys_id, uint16 blow_mode);
指令说明	设置是否启用所有非 OK 料常吹模式
指令参数	
cardid	卡号
sys_id	分选系统序号 取值范围[0,1] 第一组分选 sys_id = 0;第二组分选 sys_id = 1;
blow_mode	1:启用所有非ok料常吹模式,0: 不启用
指令返回值	

#### 指令 48 Plt\_get\_none\_ok\_blow\_mode

指令原型	Plt_get_none_ok_blow_mode(uint16 cardid, uint16 sys_id, uint16 *blow_mode);
指令说明	读取是否启用所有非 OK 料常吹模式
指令参数	
cardid	卡号

sys_id	分选系统序号 取值范围[0,1] 第一组分选 sys_id = 0;第二组分选 sys_id = 1;
blow_mode	1:启用所有非ok料常吹模式,0: 不启用
指令返回值	

#### 指令 49 Plt\_set\_output\_enable

指令原型	Plt_set_output_enable(uint16 cardid, uint16 bitno, uint16 output_enable);
指令说明	设置输出口是否启用输出， 默认情况下启用输出，此功能在对应的输出口被配置为拍照口或吹气口的条件下有效
指令参数	
cardid	卡号
bitno	输出口序号 取值范围[0,31]
output_enable	0是打开， 1是关闭
指令返回值	

#### 指令 50 Plt\_get\_output\_enable

指令原型	Plt_get_output_enable(uint16 cardid, uint16 bitno, uint16 *output_enable);
指令说明	读取输出口是否启用输出， 默认情况下启用输出，此功能在对应的输出口被配置为拍照口或吹气口的条件下有效
指令参数	
cardid	卡号
bitno	输出口序号 取值范围[0,31]
output_enable	0是打开， 1是关闭
指令返回值	

#### 指令 51 Plt\_SetCameraLatchPos

指令原型	Plt_SetCameraLatchPos(uint16 cardid, uint16 sys_id, uint16 camera_index, double pos);
指令说明	配置相机在料的哪个位置拍照
指令参数	
cardid	卡号
sys_id	分选系统序号 取值范围[0,1] 第一组分选 sys_id = 0;第二组分选 sys_id = 1;
camera_index	相机号
pos	取值范围[0,1] 默认情况下是0.5 即中心位置 0: 开始边沿 1: 结束边沿
指令返回值	

#### 指令 52 Plt\_GetCameraLatchPos

指令原型	Plt_GetCameraLatchPos(uint16 cardid, uint16 sys_id, uint16 camera_index, double *pos);
指令说明	读取配置相机在料的哪个位置拍照
指令参数	
cardid	卡号
sys_id	分选系统序号 取值范围[0,1] 第一组分选 sys_id = 0;第二组分选 sys_id = 1;
camera_index	相机号
pos	取值范围[0,1] 默认情况下是0.5 即中心位置 0: 开始边沿 1: 结束边沿
指令返回值	

#### 指令 53 Plt\_SetBlowLatchPos

指令原型	Plt_SetBlowLatchPos(uint16 cardid, uint16 sys_id, uint16 blow_index, double pos);
指令说明	配置吹气口在料的哪个位置吹气
指令参数	
cardid	卡号
sys_id	分选系统序号 取值范围[0,1] 第一组分选 sys_id = 0;第二组分选 sys_id = 1;
blow_index	吹气号
pos	取值范围[0,1] 默认情况下是0.5 即中心位置 0: 开始边沿 1: 结束边沿
指令返回值	

#### 指令 54 Plt\_GetBlowLatchPos

指令原型	Plt_GetBlowLatchPos(uint16 cardid, uint16 sys_id, uint16 blow_index, double *pos);
指令说明	读取吹气口在料的哪个位置吹气
指令参数	
cardid	卡号
sys_id	分选系统序号 取值范围[0,1] 第一组分选 sys_id = 0;第二组分选 sys_id = 1;
blow_index	吹气号
pos	取值范围[0,1] 默认情况下是0.5 即中心位置 0: 开始边沿 1: 结束边沿
指令返回值	

# 九、接线说明

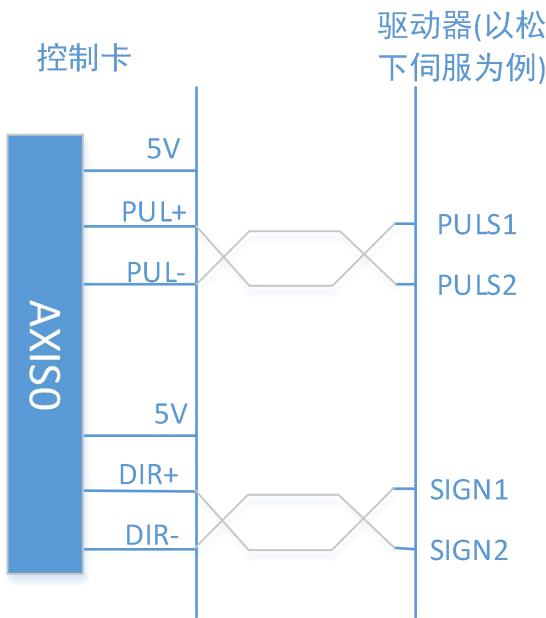
## 9.1 接线图

AXIS 0 接头信号定义表

序	名称	I/O	说 明	序	名称	I/O	说 明
1	24V_GND	0	24V 输出电源地	14	+24V	0	24V 电源输出
2	ALM	I	伺服报警信号	15	CLEAR	0	驱动报警清除
3	SEVRON	0	伺服使能输出	16	INP	I	驱动到位信号
4	EA-	I	编码器输入 A-	17	EA+	I	编码器输入 A+
5	EB-	I	编码器输入 B-	18	EB+	I	编码器输入 B+
6	EZ-	I	编码器输入 Z-	19	EZ+	I	编码器输入 Z+
7	+5V	0	内部 5V 电源输出	20	5V_GND	0	内部 5V 输出电源地
8	保留	/	/	21	5V_GND	0	内部 5V 输出电源地
9	DIR+	0	方向信号+	22	DIR-	0	方向信号-
10	5V_GND	0	内部 5V 输出电源地	23	PUL+	0	脉冲信号+
11	PUL-	0	脉冲信号-	24	5V_GND	0	内部 5V 输出电源地
12	保留	/	/	25	保留	/	/
13	5V_GND	0	内部 5V 输出电源地				

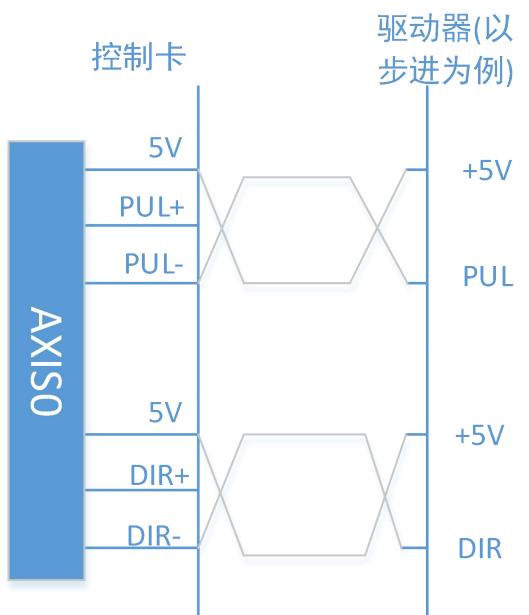
### 1、转盘电机驱动器的接线方法

当采用差分接法时，请按下图接线



差分接线方式

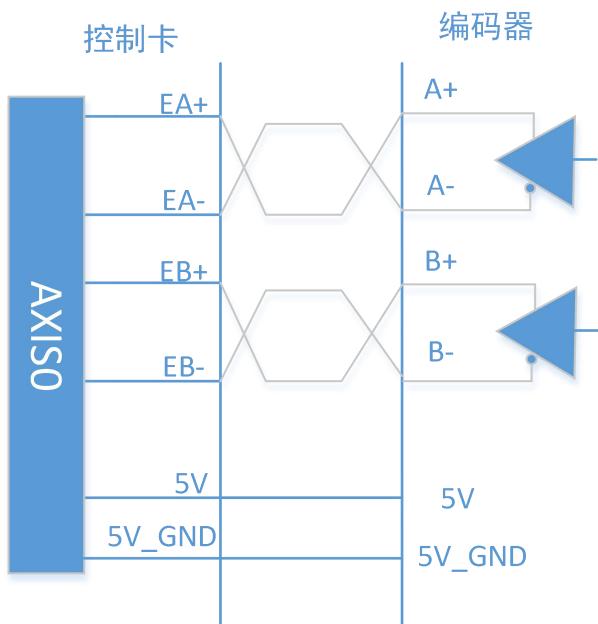
由于驱动器接口原因无法使用差分接法时，请使用下图的单端接法。



单端接线方式

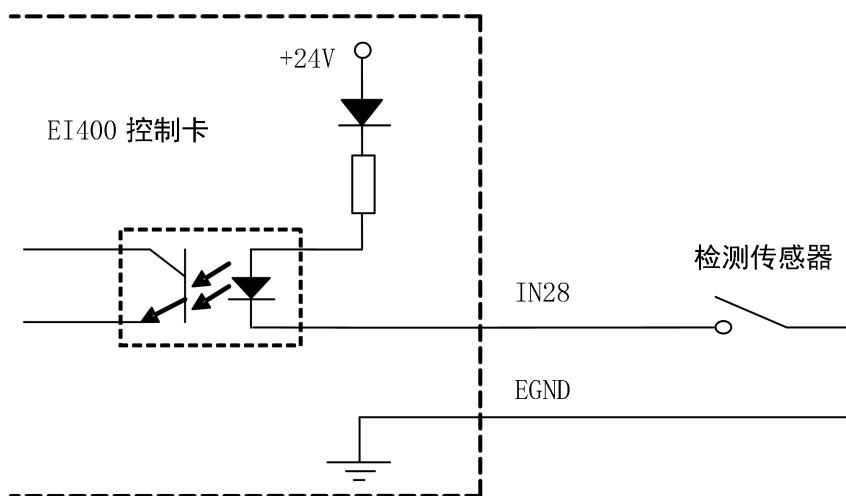
## 2、转盘编码器的接线方法

请按下图接线方式连接编码器，如果编码器需要单独供电的话，尽量使用端子上的 5V 电源为编码器供电。

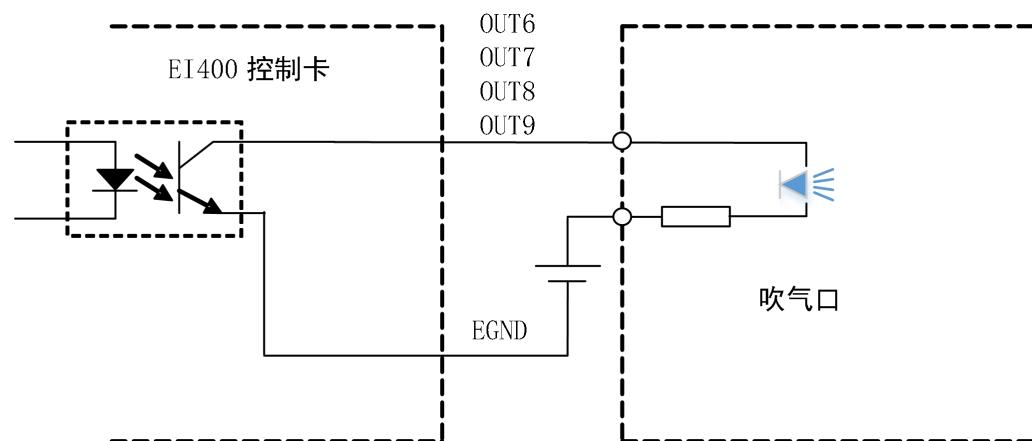
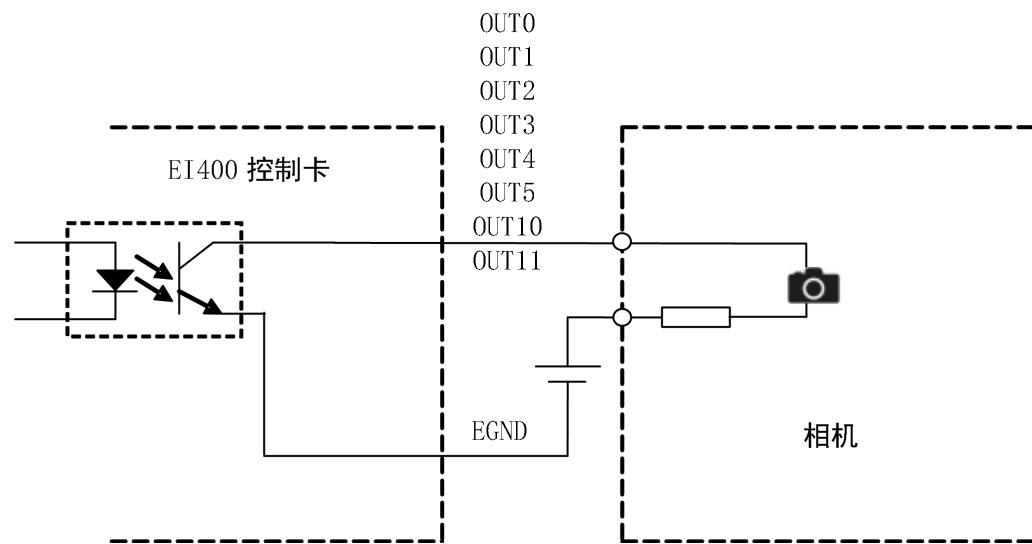


## 3、检测装置的接线方式

请按下图接入检测装置



#### 4、相机和吹气口的接线方式



## 十、错误码（函数返回值）

返回值	说明	备注
0	指令调用正确	
1	保留	
2	当前固件不支持此指令	
3	当前固件不支持此参数	
4	当前固件不支持此回零模式	
5	比较通道超限，有效范围是[0,3]	
6	比较队列超限，有效范围是[0,15]	
7	比较动作参数超限，有效范围是[0,15]	
8	手轮通道错误，通道允许值只有 0	
9	IO 映射参数错误	
10	输入 IO 端口号超限	
11	输出 IO 端口号超限	
12	保留	
13	保留	
14	保留	
15	保留	
16	保留	
17	保留	
18	保留	
19	高速锁存器范围超限，有效范围[0,1]	
20	比较缓冲区满	
21	保留	
22	保留	
23	正软限位	
24	负软限位	
25	轴运动中	
26	轴数超限	
27	保留	
28	当前轴不是点位模式	
29	目标位置超限	
30	当前轴不是 JOG 模式	
31	减速停止不允许在线变速	
32	立即停止不允许在线变速	
33	非点位模式不能在线变位	
34	减速停止不允许在线变位	
35	立即停止不允许在线变位	
36 到 63	保留	
64	回零轴号范围超限	

65	保留	
66	回零最大速度小于等于零	
67	回零最大加速度小于等于零	
68	回零最大减速度小于等于零	
69	回零距离错误	
70	保留	
71	速度规划减速度小于等于零	
72	速度规划起跳速度小于零	
73	速度规划最大速度小于等于零	
74	速度规划终点速度小于零	
75	速度规划长度小于等于零	
76	速度规划模式非 T 模式或 S 模式	
77	速度规划加速度小于等于零	
78	速度规划减速度小于等于零	
79 到 94	保留	
95	相机序号大于等于 32	
96	相机拍照时间参数设置小于等于零	
97	吹气序号大于等于 32	
98	吹气时间参数设置小于等于零	
99	物料最大宽度参数小于等于零	
100	物料最小宽度参数小于等于零	
101	物料最小宽度参数大于最大宽度参数	
102	吹气位置小于等于拍照位置	
103	拍照参数和吹气参数未设置	
104	物料过位置错误	
105	当前没有物料锁存	
106	当前位置大于吹气位置	
107	输出口序号无效	
108	吹气位置大于编码器位置	
109	缓冲区满	
110	没有物料	
111	输出口映射配置错误	
112	输出映射配置 IO 数量大于 32	
113	元件还没到最后一个相机或没有锁存到元件，就调用吹气指令	
114	第一组相机无效	
115	高速比较口启用	
116	坏料吹气物理输出口序号大于等于 32	
117	坏料吹气物理输出口序号已经被用作相机口或吹气口	
118	坏料吹气逻辑输出口无效	
200	密码错误	
201	数据报错误	
202	PCI 通讯错误	

203	轴在运动中，不能进行此操作	
204	轴非空闲状态，不能设置脉冲当量	
205	卡号超限	
206	脉冲输出模式范围超限，有效范围是[0,5]	
207	保留	
208	保留	
209	不是有效的卡种类	
210	相对于轨迹起点滞后输出模式范围超限，有效范围[0,1]	
211	相对于轨迹起点提前输出模式范围超限，有效范围[0,1]	
212	比较动作范围超限，有效范围[0,19]	
213	负软限位大于正软限位	
214	限位使能参数有误，可取的值为 0, 1,2,3	
215	限位逻辑参数有误，可取的值为 0, 1,2,3	
216	限位模式参数有误，可取的值为 0, 1,2,3	
217	减速停止输入 IO 超限，合理范围[0,31]	
218	总卡数为零	
219	开卡错误	
220	二维比较位置重合	
221	可计数的输入 IO 范围超限，合理范围[0, 1,5,]	
222	滤波参数小于零	
223	映射的 IO 种类超限，合理范围[0,7]	
224	被映射的 IO 种类超限，合理范围[0,6]	
225	保留	
226	以太网通信失败	
227	DII 没有初始化	
401	JOG 运动最大速度小于等于零	
402	JOG 运动加速度小于等于零	
403	JOG 运动减速度小于等于零	
404	JOG 运动 S 曲线因子小于等于零	
405	JOG 运动在线变速或变运动方向失败	
406	JOG 运动非运动状态下不能在线变速	
501	点位运动最大速度小于等于零	
502	点位运动加速度小于等于零	
503	点位运动减速度小于等于零	
504	点位运动 S 曲线因子小于等于零	
505	点位运动在线变速或变运动方向失败	
506	点位运动非运动状态下不能在线变速	
507	点位运动有效长度为零	