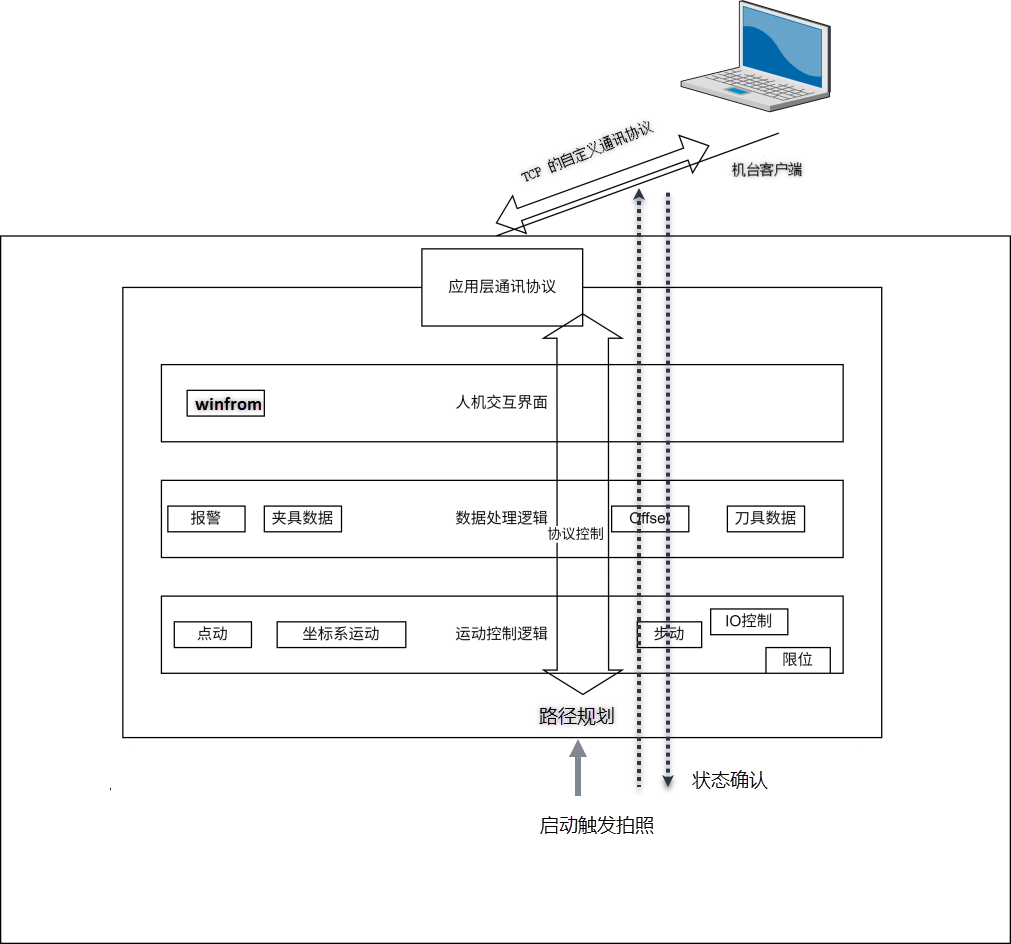
### 铜粒子定位磨削系统软件开发文档

#### 一、概述

本控制软件是针铜粒子铣床开发的上位机控制系统，具备高性能、高可靠性和良好的人机交互体验。系统采用模块化架构，集成了铜粒子路径规划、运动控制、报警管理、数据记录等核心功能，能够高效支持铜板粒子铣削任务。

#### 二、系统架构与技术实现

##### 1.总体结构



**图 1.1 软件结构图**

**系统采用分层架构，主要包括：**

**硬件控制层**：通过 TCP/IP协议与3D相机进行通讯，实现触发采集点云数据。

**路径规划层：封装路径规划及气缸避让算法，实现一次采集，输出打磨路径坐标，加工数据匹配，气缸避让等多维数据包**

**业务逻辑层**：封装各类报文解析、自动生成报文、运动模式、流程控制、报警检测及数据管理。

**界面交互层**：基于 winform技术实现可视化人机界面，提升操控便利性。

**远程通讯层**：通过基于 TCP 的自定义通讯协议，与机台应用系统实现高效交互。

##### 远程通讯协议

本系统设计并实现了一套基于 TCP 协议的 **高效定制通讯协议**，用于上位机与机台控制系统之间的数据交换。

**协议描述**：

本标准的基础传输层构建在TCP/IP协议上，应用层的协议与具体的传输网络无关，应用层的具体协议详见下文。本标准与通讯介质无关。

* 1. 应答模式

完整的命令由请求方发起、响应方应答组成，具体步骤如下：

1. 请求方发送请求命令给响应方；
2. 响应方接到请求后，向请求方发送请求应答（握手完成）；
3. 请求方收到请求应答后，等待响应方回应执行结果；如果请求方未收到请求应答，按请求回应超时处理；
4. 响应方执行请求操作；
5. 响应方发送执行结果给请求方；
6. 请求方收到执行结果，命令完成；如果请求方没有收到执行结果，按执行超时处理。
   1. 超时重发机制

* 一个请求命令发出后在规定的时间内未收到回应，视为超时：
* 超时后重发，重发超过规定次数后仍未收到回应视为通讯不可用，通讯结束；
* 超时时间根据具体的通讯方式和任务性质可自定义；
* 超时重发次数根据具体的通讯方式和任务性质可自定义。

请求方在收到请求回应后规定时间内未收到返回数据或命令执行结果，认为超时，命令执行失败，请求操作结束。

* 数据结构

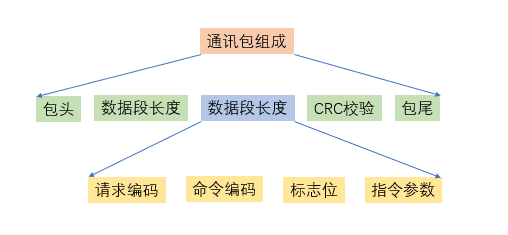


图1 通讯协议数据结构

2.1 通讯包结构组成

表1 通讯包结构组成表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 描述 |
| 包头 | 字符 | 2 | 固定为## |
| 数据段长度 | 十进制整数 | 4 | 数据段的ASCII字符数，例如：长255，则写为“0255” |
| 数据段 | 字符 | 0≤n≤1024 |  |
| CRC校验 | 十六进制整数 | 4 | 数据段的校验结果，CRC校验算法见附录A，接收到一条命令，如果CRC错误，执行结束 |
| 包尾 | 字符 | 2 | 固定为<CR><LF>(回车、换行) |

2.2 数据段结构组成

表2 数据段结构组成表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 长度 | 描述 |
| 请求编码QN | 字符 | 20 | 精确到毫秒的时间戳：QN=YYYYMMDDhhmmsszzz，用来唯一标示一次命令交互 |
| 命令编码CN | 字符 | 6 | CN=命令编码，命令编码取值详见表6 命令编码表 |
| Flag | 整数 | 8 | Flag=标志位，这个标志位包含标准版本号、数据是否应答。   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | V6 | V5 | V4 | V3 | V2 | V1 | V0 | A |   V6~V0：标准版本号；Bit：000000表示本次标准修订版本号。  A：命令时候应答；Bit：1-应答，0-不应答。 |
| 指令参数CP | 字符 | 0≤n≤990 | CP=&&数据区&& |
|  |  |  |  |

2.3 数据区

2.3.1 结构定义

字段与其值用’=’连接；在数据区中，同一项目的不同分类值用’，‘来分隔，不同项目之间用’；‘来分隔。

2.3.2 字段定义

2.3.2.1 字段名

字段名要区分大小写，单词的首个字符为大写，其他部分为小写。

2.3.2.2 数据类型

C4: 表示最多4位的字符型字符串，不足4位按实际位数；

N5: 表示最多5位的数字型字符串，不足5位按实际位数；

N14.2 用可变长字符串形式表达的数字型，表示14为整数和2位小数，带小数点，带符号，最大长度为18；

YYYY： 日期年，如2016表示2016年；

MM： 日期月，如09表示9月；

DD： 日期日，如24表示24日；

hh： 时间小时；

mm： 时间分钟；

ss： 时间秒；

zzz： 时间毫秒；

2.3.2.3 字段对照表

字段对照表如表3所示，表3中“宽度”仅包含该字段的内容长度。

表3 字段对照表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 描述 | 字符集 | 宽度 | 取值及描述 |
| QnRtn | 请求回应代码 | 0-9 | N3 | 取值详见表5《请求命令返回表》 |
| ExeRtn | 执行结果回应代码 | 0-9 | N3 | 取值详见表4《执行结果定义表》 |
| D | 刀具直径 | 0-9 | -- | 单位mm |
| R | 刀具转速 | 0-9 | -- | 单位rpm |
| S | 行刀间距 | 0-9 | -- | 单位mm |
| H | 吃刀深度 | 0-9 | -- | 单位mm |
| V | 进给速度 | 0-9 | -- | 单位mm/min |
| T | 切削总深 | 0-9 | -- | 单位mm |
| I | 工件表面高度 | 0-9 | -- | 单位mm |
| SX | 起始点X位置 | 0-9 | -- | 单位mm |
| SY | 起始点Y位置 | 0-9 | -- | 单位mm |
| SZ | 起始点Z位置 | 0-9 | -- | 单位mm |
| EX | 结束点X位置 | 0-9 | -- | 单位mm |
| EY | 结束点Y位置 | 0-9 | -- | 单位mm |
| EZ | 结束点Z位置 | 0-9 | -- | 单位mm |
| No | 第几条指令 | 0-9 | -- | 表明当前指令序号。 |
| Count | 指令总数 | 0-9 | -- | 表明当前机台存在待执行指令数量。 |
| State | 机台状态 | A~Z | C2 | 第一位表明机台运行状态，  R：运行；  P：暂停；  A：报警；  F：空闲；  I：初始化完成；  第二位表明机台控制方式，  L：本地，  R：远程 |
| DA | 数据响应 | 0-9 | C3 | 引用接收数据命令编码 |
| IOBits | IO地址列 | 0-9&A~F | C5 | 取值详见 **附录B** |
| CKSM | 校验和 | 0-9&A~F | C16 | 64位数据校验和 |

* 代码定义

3.1 执行结果定义

表4 执行结果定义表

|  |  |  |
| --- | --- | --- |
| 编号 | 描述 | 备注 |
| 1 | 执行成功 |  |
| 2 | 执行失败，但不知道原因 |  |
| 3 | 命令请求条件错误 |  |
| 4 | 通讯超时 |  |
| 5 | 系统繁忙不能执行 |  |
| 6 | 系统故障 |  |
| 100 | 没有数据 |  |

3.2 请求命令返回

请求命令返回如表5所示。

表5 请求命令返回表

|  |  |  |
| --- | --- | --- |
| 编号 | 描述 | 备注 |
| 1 | 准备执行请求 |  |
| 2 | 请求被拒绝 |  |
| 3 | Flag错误 |  |
| 4 | QN错误 |  |
| 5 | CN错误 |  |
| 6 | CRC检验错误 |  |
| 100 | 未知错误 |  |

3.3 命令编码（可扩充）

对应“图4通讯协议数据结构”中的命令编码。

3.3.1 类别划分

共有四类命令（即请求命令、上传命令、通知命令和交互命令），命令编码分为以下四组：

100~199表示初始化命令和参数命令编码；

200~299表示数据命令编码；

300~399表示控制命令编码；

900~999表示交互命令编码。

3.3.2 命令编码方法

命令编码用3为阿拉伯数字表示，如表6所示。

表6 命令编码表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 命令名称 | 命令编码 | | 命令类型 | 描述 |
| 上位向机台 | 机台向上位 |
| 初始化命令 | | | | |
| 设备初始化 | 100 |  | 请求命令 | 用于上位机通知现场机执行初始化动作 |
| 预留初始化命令 |  |  |  | 预留命令范围101-110 |
| 参数命令 | | | | |
| 查询机台工艺参数 | 111 |  | 请求命令 | 用于提取机台的工艺参数 |
| 上传机台工艺参数 |  | 111 | 上传命令 | 用于上传机台的工艺参数 |
| 设置机台工艺参数 | 112 |  | 请求命令 | 用于设置机台的工艺参数 |
| 预留参数命令 |  |  |  | 预留命令范围113-199 |
| 数据命令 | | | | |
| 取机台状态 | 201 |  | 请求命令 | 用于启动机台上传运行状态 |
| 上传机台状态 |  | 201 | 上传命令 | 用于机台上传当前设备状态 |
| 停止查看机台状态 | 202 |  | 通知命令 | 用于停止机台上传运行状态 |
| 取机台动作列表 | 203 |  | 请求命令 | 用于查询机台待执行的动作列表 |
| 清除机台动作 | 204 |  | 请求命令 | 用于删除之前添加的指令动作，当No=-1的情况下，删除所有指令动作。 |
| 查询DI状态列表 | 205 |  | 请求命令 | 用于查看机台DI状态，使用IOBits表明要查询的DI地址。 |
| 查询动作校验和 | 206 |  | 请求命令 | 查询机台中动作列表所生成的校验和 |
| 添加指令 | | | | |
| 快速移动 | 210 |  | 请求命令 |  |
|  |  | 210 | 上传命令 | 用于回复上位机的指令查询 |
| 切割移动 | 211 |  | 请求命令 |  |
|  |  | 211 | 上传命令 | 用于回复上位机的指令查询 |
| DO开 | 212 |  | 请求命令 |  |
|  |  | 212 | 上传命令 | 用于回复上位机的指令查询 |
| DO关 | 213 |  | 请求命令 |  |
|  |  | 213 | 上传命令 | 用于回复上位机的指令查询 |
| 预留数据命令 |  |  |  | 预留命令范围214~299 |
| 控制命令 | | | | |
| 执行动作 | 301 |  | 请求命令 | 用于上位机启动机台按照添加的指令动作进行执行。 |
| 暂停动作 | 302 |  | 请求命令 | 用于上位机暂停当前机台的指令动作。 |
| DO开 | 303 |  | 请求命令 | 用于上位机控制机台打开IOBits中指定的DO |
| DO关 | 304 |  | 请求命令 | 用于上位机控制机台关闭IOBits中指定的DO |
| 全部夹紧 | 305 |  | 请求命令 | 用于上位机机控制机台打开**所有**夹具 |
| 全部松开 | 306 |  | 请求命令 | 用于上位机机控制机台关闭**所有**夹具 |
| 停止 | 307 |  | 请求命令 | 用于上位机控制机台停止生产流程 |
| 预留控制命令 |  |  |  | 预留命令范围308~399 |
| 交互命令 | | | | |
| 请求应答 |  | 901 |  | 用于机台回应接收的上位机请求命令是否有效 |
| 执行结果 |  | 902 |  | 用于机台回应接收的上位机请求命令执行结果 |
| 通知应答 | 903 | 903 |  | 回应通知命令 |
| 数据应答 | 904 | 904 |  | 数据应答命令 |
| 预留交互命令 |  |  |  | 预留命令范围905~999 |

#### 请求顺序

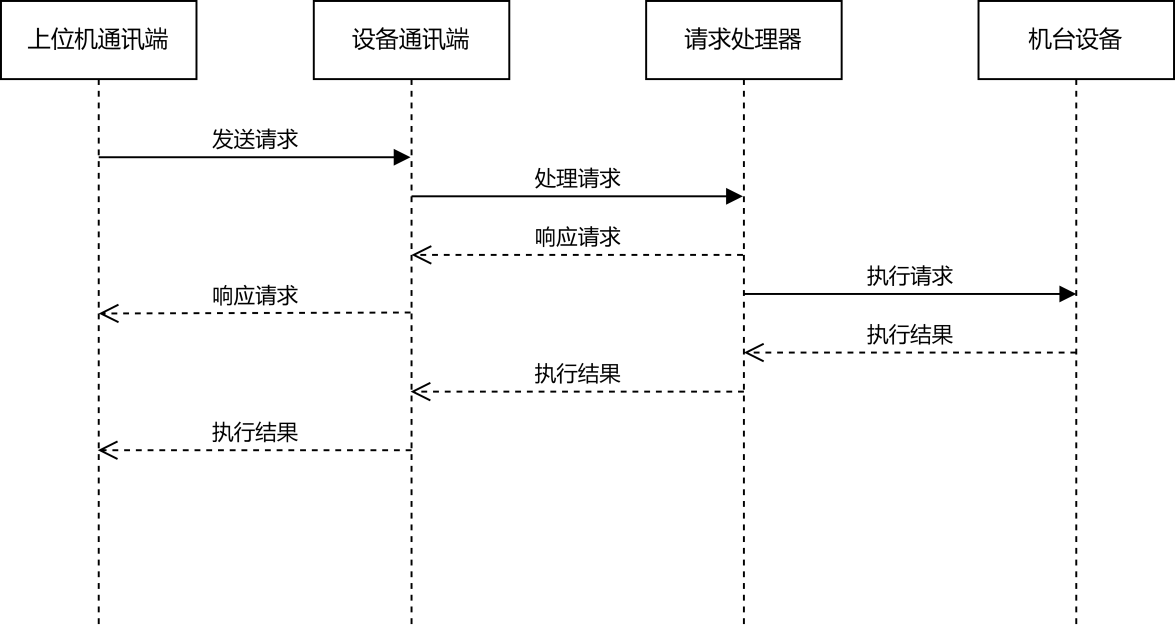


图2 请求顺序图

## 通讯命令示例和拆分包及应答机制示例

表7 设备初始化

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “设备初始化” | QN=20241231115731779;CN=100;Flag=1;CP=&&&& |
| 机台 | 返回“请求应答” | QN=20241231115731779;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20241231115731779;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| 执行过程 | 1. 上位机发送 “设备初始化”命令； 2. 机台检查请求报文并返回请求应答；   3、机台接收“设备初始化”命令并执行 设备初始化流程；  4、机台完成设备初始化流程后返回“执行结果”；  5、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表8设置机台工艺参数

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “设置机台工艺参数” | QN=20241231133614817;CN=112;Flag=1;CP=&&D=71.189,H=43.206,V=7.811,S=26.761,R=45.826&& |
| 机台 | 返回“请求应答” | QN=20241231133614817;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20241231133614817;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| D | | 刀具直径 |
| H | | 切削总深 |
| V | | 进给速度 |
| S | | ⾏刀间距 |
| R | | 刀具转速 |
| 执行过程 | 1. 上位机发送 “设置机台工艺参数”命令；   2、机台检查请求报文并返回请求应答；  3、机台接收“设置机台工艺参数”命令并执行设置机台工艺参数流程；  4、机台完成工艺参数设置后返回“执行结果”；  5、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表9 查询机台工艺参数

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “查询机台工艺参数”命令 | QN=20241231133903587;CN=111;Flag=1;CP=&&&& |
| 机台 | 返回“请求应答” | QN=20241231133903587;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“工艺参数” | QN=20241231133903587;CN=111;Flag=0;CP=&&CP=&&D=71.189,H=43.206,V=7.811,S=26.761,R=45.826&& |
| 上位机 | 返回“数据应答” | QN=20241231133903587;CN=904;Flag=0;CP=&&DA=000111&& |
| 机台 | 返回“执行结果” | QN=20241231133903587;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| D | | 刀具直径 |
| H | | 切削总深 |
| V | | 进给速度 |
| S | | 行刀间距 |
| R | | 刀具转速 |
| D | | 刀具直径 |
| DA | | 应答数据类型 |
| 执行过程 | 1、上位机发送 “查询机台工艺参数”命令；  2、机台接收“查询机台工艺参数”命令并返回请求应答；  3、机台上传本地的工艺参数并开始等待上位机的数据应答；  4、上位机接收“工艺参数列表后”，返回应答的数据应答；  5、机台接受到数据应答后，返回执行结果；  6、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表10 添加执行动作

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “快速移动”/“切割移动” | QN=20241231134233533;CN=210;Flag=0;CP=&&SX=24.450,SY=41.353,SZ=77.725,EX=7.679,EY=97.946,EZ=61.347&& |
| 机台 | 返回“请求应答” | QN=20241231134233533;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20241231134233533;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| SX | | 起始点 X 位置 |
| SY | | 起始点 Y 位置 |
| SZ | | 起始点Z位置 |
| EX | | 结束点 X 位置 |
| EY | | 结束点 Y位置 |
| EZ | | 结束点 Z 位置 |
| 执行过程 | 1. 上位机发送 “快速移动”/“切割移动”命令； 2. 机台检查请求报文并返回请求应答；   3、“快速移动”/“切割移动”被添加到机台的待执行动作列表；  4、机台返回“执行结果”；  5、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表11查询动作列表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “查询动作列表” | QN=20241231134344169;CN=203;Flag=1;CP=&&&& |
| 机台 | 返回“请求应答” | QN=20241231134344169;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 上传“快速移动” | QN=20241231134344169;CN=000210;Flag=0;CP=&&SX=24.450,SY=41.353,SZ=77.725,EX=7.679,EY=97.946,EZ=61.347,NO=0&& |
| 机台 | 上传“切割移动” | QN=20241231134344169;CN=000211;Flag=0;CP=&&SX=97.091,SY=71.921,SZ=29.566,EX=76.677,EY=85.985,EZ=83.485,NO=1&& |
| 上位机 | 返回“数据应答” | QN=20241231134344169;CN=904;Flag=0;CP=&&DA=210;NO=0&& |
| 上位机 | 返回“数据应答” | QN=20241231134344169;CN=904;Flag=0;CP=&&DA=211;NO=1&& |
| 机台 | 返回“执行结果” | QN=20241231134344169;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| SX | | 起始点 X 位置 |
| SY | | 起始点 Y 位置 |
| SZ | | 起始点Z位置 |
| EX | | 结束点 X 位置 |
| EY | | 结束点 Y位置 |
| EZ | | 结束点 Z 位置 |
| DA | | 应答数据类型 |
|  | NO | | 表明当前指令的执行序号 |
| 执行过程 | 1、上位机发送 “查询动作列表”命令；  2、机台接收“查询动作列表”命令并返回请求应答；  3、机台依次上传本地的所有带执行动作；  4、上位机接收设备上传的所有的“快速移动”/“切割移动”，返回数据应答；  5、机台接受到所有数据应答后，返回执行结果；  6、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表12执行动作

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “执行动作”指令 | QN=20250107193416210;CN=301;Flag=0;CP=&&&& |
| 机台 | 返回“请求应答” | QN=20250107193416210;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20250107193416210;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| 执行过程 | 1. 上位机发送 “执行动作”命令；   2、机台检查请求报文并返回请求应答；  2、机台接收“执行动作”命令并依据自身状态判断是否执行，若允许执行则开始执行并返回请求结果，若不能执行则返回相应（系统繁忙/故障）的**执行代码**；  4、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表8暂停动作

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “暂停动作” | QN=20250107175100769;CN=302;Flag=0;CP=&&&& |
| 机台 | 返回“请求应答” | QN=20250107175100769;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20250107175100769;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| 执行过程 | 1. 上位机发送 “暂停动作”命令；   2、机台检查请求报文并返回请求应答；  3、机台停止生产流程  3、返回执行结果  4、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表9清除动作

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “清除机台动作” | QN=20250107193646591;CN=204;Flag=1;CP=&&NO=1&& |
| 机台 | 返回“请求应答” | QN=20250107193646591;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20250107193646591;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| NO | | 需要删除的指令序号 |
| 执行过程 | 1. 上位机发送 “清除机台动作”命令；   2、机台检查请求报文并返回请求应答；  3、依据指令中的NO对待执行动作队列中的运动指令进行删除  3、返回执行结果  4、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表 10 追加DO 开

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “追加DO开” | QN=20250208170508745;CN=212;Flag=0;CP=&&IOBits=10F01&& |
| 机台 | 返回“请求应答” | QN=20250208170508745;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20250208170508745;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| IOBits | | 需要关闭的 4\*5(5个HEX字符) 个 IO的地址列表，当对应编号的bit值为1时则表示在该DO输出打开信号 |
| 执行过程 | 1. 上位机发送 “追加DO开”命令；   2、机台检查请求报文并返回请求应答；  2、机台将该IO指令追加到动作执行队列  3、返回执行结果  4、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表 11 追加DO 关

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “追加DO关” | QN=20250208170508745;CN=213;Flag=0;CP=&&IOBits=10F01&& |
| 机台 | 返回“请求应答” | QN=20250208170508745;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20250208170508745;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| IOBits | | IO地址列表 |
| 执行过程 | 1. 上位机发送 “追加DO关”命令；   2、机台检查请求报文并返回请求应答；  3、机台将该IO指令追加到动作执行队列  3、返回执行结果  4、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表 12 DO开

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “DO开” | QN=20250208171123414;CN=303;Flag=0;CP=&&IOBits=0EF21&& |
| 机台 | 返回“请求应答” | QN=20250208171123414;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20250208171123414;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| IOBits | | IO地址列表 |
| 执行过程 | 1、上位机发送 “DO开”命令；  2、机台检查请求报文并返回请求应答；  3、机台依据指定的0~n 个有效的DO地址进行DO开操作  4、返回执行结果  5、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表 13 DO关

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “DO关” | QN=20250208171830972;CN=304;Flag=0;CP=&&IOBits=10101&& |
| 机台 | 返回“请求应答” | QN=20250208171830972;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 返回“执行结果” | QN=20250208171830972;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| IOBits | | IO地址列表 |
| 执行过程 | 1. 上位机发送 “追加DO关”命令；   2、机台检查请求报文并返回请求应答；  3、机台依据指定的0~n 个有效的DO地址进行DO关闭操作  4、返回执行结果  5、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

表 14 查询设备DI状态列表

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 项目 | | 示例/说明 |
| 使用命令 | 上位机 | 发送 “取DI列表”命令 | QN=20250208171945129;CN=205;Flag=1;CP=&&IOBits=FFFFF&& |
| 机台 | 返回“请求应答” | QN=20250208171945129;CN=901;Flag=0;CP=&&QnRtn=1&& |
| 机台 | 上传上位机查询的的设备IO状态 | QN=20250208171945129;CN=205;Flag=0;CP=&&IOBits=0EE20&& |
| 上位机 | 对设备上传的DI状态列表进行数据响应 | QN=20250208171945129;CN=904;Flag=0;CP=&&DA=205&& |
| 机台 | 返回“执行结果” | QN=20250208171945129;CN=902;Flag=0;CP=&&ExeRtn=1&& |
| 使用字段 | QnRtn | | 请求应答结果 |
| ExeRtn | | 请求执行结果 |
| IOBits | | IO地址列表 |
| DA | | 数据响应 |
| 执行过程 | 1、上位机发送 “查询DI列表”命令；  2、机台接收命令并返回请求应答  3、返回执行结果  4、上位机接收“执行结果”，根据执行结果标志 ExeRtn 的值判断请求是否完成，请求执行完毕； | | |

附 录 A

（规范性附录）

循环冗余校验（CRC）算法

CRC校验（Cyclic Redundancy Check）是一种数据传输错误检查方法。本标准采用ANSI CRC16， 简称CRC16。

CRC16 码由传输设备计算后加入到数据包中。接收设备重新计算接收数据包的CRC16码，并与接 收到的CRC16码比较，如果两值不同，则有误。

CRC16 校验字节的生成步骤如下：

1) CRC16 校验寄存器赋值为0xFFFF；

2) 取被校验串的第一个字节赋值给临时寄存器；

3) 临时寄存器与CRC16校验寄存器的高位字节进行“异或”运算，赋值给CRC16校验寄存器；

4) 取CRC16校验寄存器最后一位赋值给检测寄存器；

5) 把CRC16校验寄存器右移一位；

6) 若检测寄存器值为1，CRC16校验寄存器与多项式0xA001进行“异或”运算，赋值给CRC16 校验寄存器；

7) 重复步骤4~6，直至移出8位；

8) 取被校验串的下一个字节赋值给临时寄存器；

9) 重复步骤3~8，直至被校验串的所有字节均被校验；

10) 返回CRC16校验寄存器的值。 校验码按照先高字节后低字节的顺序存放。

CRC校验算法示例：

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

函 数: CRC16\_Checkout

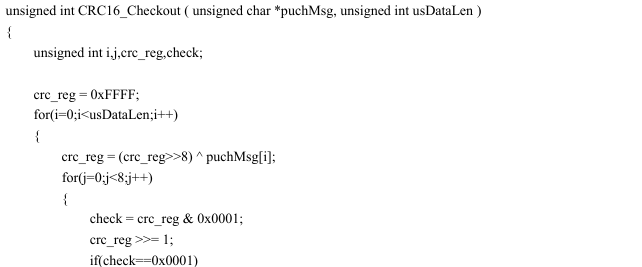
描 述: CRC16循环冗余校验算法。

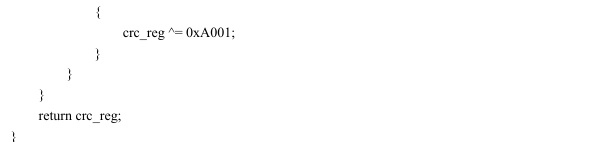
参 数 一: \*puchMsg：需要校验的字符串指针

参 数 二: usDataLen：要校验的字符串长度

返 回 值: 返回CRC16校验码

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





附 录 B

IO设备取值说明

使用5位16进制来对IO设备进行编址，每位16进制可以表示4个IO设备 4\*5 即共可表示20个IO设备。

表1机台IO设备编址

|  |  |  |
| --- | --- | --- |
| 编号 | 值(16进制字符串) | 含义 |
| 0 | 0X00001 | 气缸1 |
| 1 | 0X00002 | 气缸2 |
| 2 | 0X00004 | 气缸3 |
| 3 | 0X00008 | 气缸4 |
| 4 | 0X00010 | 气缸5 |
| 5 | 0X00020 | 气缸6 |
| 6 | 0X00040 | 气缸7 |
| 7 | 0X00080 | 气缸8 |
| 8 | 0X00100 | 气缸9 |
| 9 | 0X00200 | 气缸10 |
| 10 | 0X00400 | 气缸11 |
| 11 | 0X00800 | 气缸12 |
| 12~19 |  | 预留IO地址 |

1. 允许同时操控多个DO设备，例如若需要打开气缸 1~12 可以使用 位运算 ‘|’对各个气缸IO的地址值进行连接运算。

wps

1. 查询DI列表时同理，指令可以仅仅查询关注的0~n|{n<20} 个DI信号，例如需要查询气缸 {2,6,8}

C:/Users/xinchenglei/AppData/Local/Temp/wps.AFADGdwps

##### UI界面设计

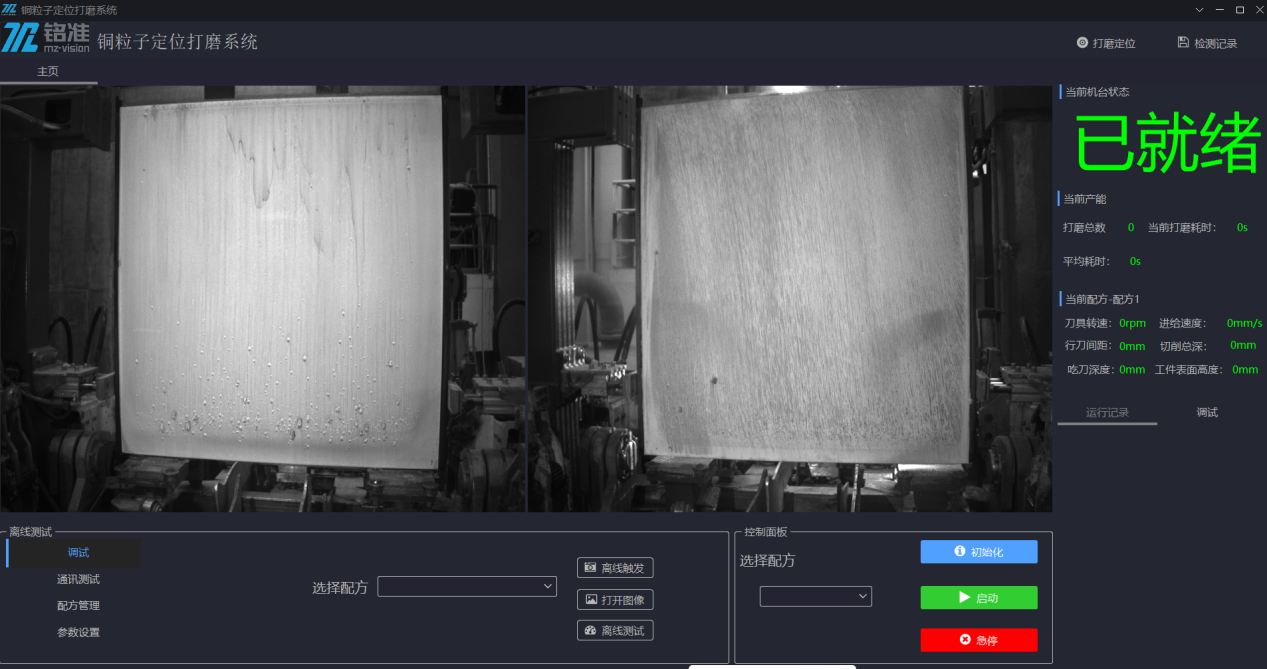
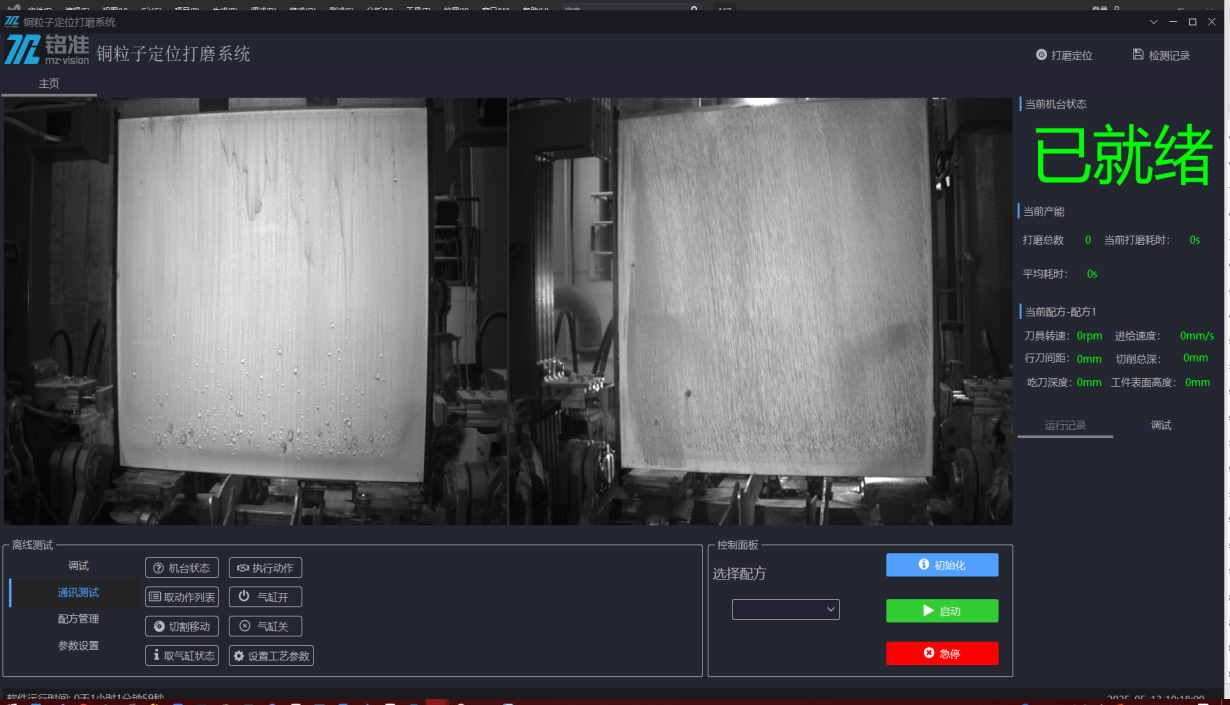
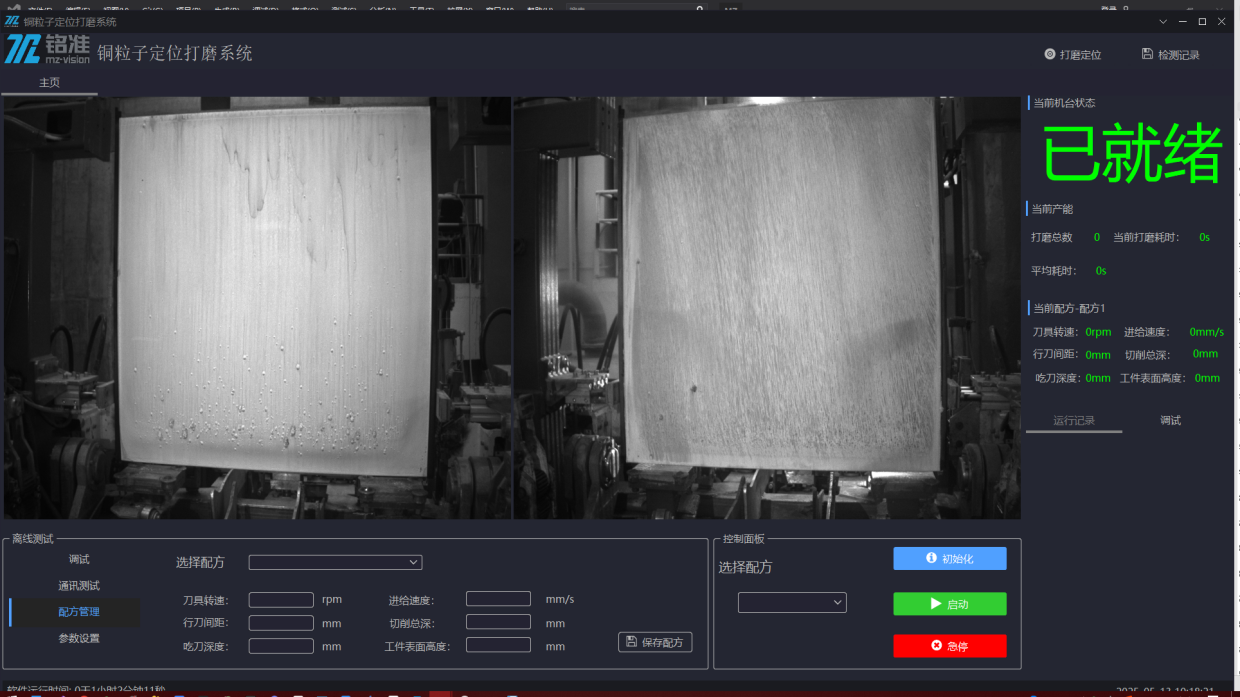


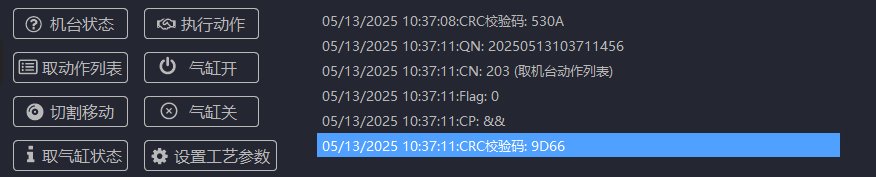
图3 界面效果图

人机交互界面因具备如下基本功能：

* 1. 因能显示3D相机实时画面，及触发后的路径规划图。
  2. 因能直观的显示机台当前状态，后台以轮训的形式获取机台状态，当机台异常时，应能采取相关连锁机制，禁止触发和远控的相关功能，确保安全。
  3. 如图所示：
* 应能统计打磨总数，打磨耗时，等相关统计数据，
* 需要显示当前配方；
* 需要输出日志及报警信息（包括但不限于监测机台异常状态及软件本身报警）；
* 需要具备离线调试功能；
* 需要具备通讯测试功能；
* 需要具备系统参数设置功能；
* 需要具备配方管理功能；
* 需要具备打磨数据统计功能，输出表格功能；







**后台交互逻辑：**