机器人控制器基本功能设计：

1. 记忆功能：存储作业顺序、运动路径、运动方式、运动速度和与生产工艺有关的信息。
2. 示教功能：离线编程，在线示教，间接示教。在线示教包括示教盒和导引示教两种。
3. 与外围设备联系功能：输入和输出接口、通信接口、网络接口、同步接口。
4. 坐标设置功能：有关节、绝对、工具、用户自定义四种坐标系
5. 人机接口：示教盒、操作面板、显示屏。
6. 传感器接口：位置检测、视觉、触觉、力觉等。
7. 位置伺服功能：机器人多轴联动、运动控制、速度和加速度控制、动态补偿等。
8. 故障诊断安全保护功能：运行时系统状态监视、故障状态下的安全保护和故障自诊断。

|  |  |  |
| --- | --- | --- |
| 控制器功能 | 解释说明 | 完成情况 |
| 参数配置 | 配置机器人DH、伺服、运动、坐标系等参数，并生成xml文件供控制器读写 | 已完成 |
| 手动示教 | 分为关节空间坐标系的示教和直角坐标系（世界坐标系和工具坐标系）的示教 | 关节空间坐标系的示教已完成，直角坐标系的示教因为没有底层API支持，需做特殊处理 |
| 离线编程 | 根据作业任务编写机器人运行程序，生成对应程序文件和变量文件，该文件存储有作业顺序、运动路径、运动方式、运动速度等信息 | 已完成，可支持约三十种控制指令的编辑 |
| 程序编译 | 对程序文件和变量进行词法、语法、语义分析和解释 | **基本完成** |
| 逻辑控制 | 点击示教器上的某个按键时，控制器应做哪些响应 | 已完成 |
| 数据传输 | 控制器与示教器间所需数据的定义和传输 | 已完成 |
| 位置检测 | 示教器上动态显示机器人在各坐标系下的实时位置数据等 | 已完成 |
| 使能状态设置 | 控制机器人的上使能和下使能 | 已完成 |
| 操作模式设置 | 切换机器人的操作模式为手动或自动 | 已完成 |
| 程序暂停和继续 | 暂停和继续程序的执行，需对线程的控制做处理 | 已完成 |
| 程序执行方式的设置 | 程序执行方式分为单步和连续 | 已完成 |
| 坐标系设置 | 切换参考坐标系，包括关节、世界、工具和用户自定义坐标 | 未完成 |
| 速度和加速度控制 | 根据程序指令的参数或示教器上的速度加减按钮调整机器人的运动速度和加速度 | 已完成 |
| 子程序调用 | 调用同一项目下的其他程序并执行 | **已完成** |
| 多程序平行运行与终止 | 新建线程并发执行或终止同一项目下的其他程序 | **已完成** |
| 功能整合 | 整合以上功能，进行联调 | 未完成 |

|  |  |  |
| --- | --- | --- |
| 指令组 | 具体指令 | 解释说明 |
| 运动指令组 | **PTP** | **点位运动指令** |
| **Lin** | **直线运动指令** |
| Circ | 圆弧运动指令 |
| **PTPRel** | **点位插补相对偏移指令** |
| **LinRel** | **直线插补相对偏移指令** |
| StopRobot | 停止机器人运动并且丢弃已经计算好的插补路径 |
| RefRobotAxis | 标定回零位置，针对单一轴 |
| RefRobotAxisAsync | 允许多轴同时回零，把当前机器人所有轴位置作为各轴的零位 |
| MoveRobotAxis | 对机器人的单一轴进行运动 |
| 设置指令组 | Dyn | 设置机器人运动参数 |
| Dynovr | 设置机器人运动的动态倍率参数 |
| Ramp | 设置速度规划方式 |
| Refsys | 设置参考系 |
| Tool | 设置工具坐标系相对法兰盘的位置 |
| 系统功能指令组 | WaitTime | 设置机器人等待时间 |
| Stop | 停止所有激活程序的执行 |
| Circ | 发出一条信息通知 |
| Warning | 发出一条警告信息 |
| Error | 发出一条错误消息 |
| 流程控制指令组 | **CALL** | **调用指令** |
| WAIT | 等待指令 |
| IF | 条件跳转控制指令 |
| WHILE | 在满足条件时循环控制指令 |
| LOOP | 循环次数控制指令 |
| **RUN** | **并发运行子程序指令** |
| **KILL** | **终止子程序指令** |
| RETURN | 终止本程序指令 |
| **GOTO** | **跳转执行指令** |

**子语句解释器：**

//运动指令组

void PTP\_interpreter(ast \*a);

void Lin\_interpreter(ast \*a);

void Circ\_interpreter(ast \*a);

void PTPRel\_interpreter(ast \*a);

void LinRel\_interpreter(ast \*a);

void StopRobot\_interpreter(ast \*a);

void RefRobotAxis\_interpreter(ast \*a);

void RefRobotAxisAsync\_interpreter(ast \*a);

void MoveRobotAxis\_interpreter(ast \*a);

//设置指令组

void Dyn\_interpreter(ast \*a);

void Dynovr\_interpreter(ast \*a);

void Ramp\_interpreter(ast \*a);

void Refsys\_interpreter(ast \*a);

void Tool\_interpreter(ast \*a);

//系统功能指令组

void WaitTime\_interpreter(ast \*a);

void Stop\_interpreter(ast \*a);

void Notice\_interpreter(ast \*a);

void Warning\_interpreter(ast \*a);

void Error\_interpreter(ast \*a);

//流程控制指令组

void CALL\_interpreter(ast \*a);

void WAIT\_interpreter(ast \*a);

void IF\_interpreter(ast \*a);

void WHILE\_interpreter(ast \*a);

void LOOP\_interpreter(ast \*a);

void RUN\_interpreter(ast \*a);

void RETURN\_interpreter(ast \*a);

void GOTO\_interpreter(ast \*a);

**机器人控制指令**：

//机器人正解

CARTPOS forwardKinematics(AXISPOS);

//机器人逆解，axisPos为上一次计算得到的逆解，以便求出距离上次逆解最近的新逆解

AXISPOS inverseKinematics(CARTPOS cartPos, AXISPOS axisPos);

//根据位置的类型和是否设置dynamic参数，运动指令有几种不同的形式

void PTP(AXISPOS pos, DYNAMIC dyn);

void PTP\_AP\_DYN(AXISPOS pos, DYNAMIC dyn);

void PTP\_CP(CARTPOS pos);

void PTP\_CP\_DYN(CARTPOS pos, DYNAMIC dyn);

void Lin\_AP(AXISPOS pos);

void Lin\_AP\_DYN(AXISPOS pos, DYNAMIC dyn);

void Lin\_CP(CARTPOS pos);

void Lin\_CP\_DYN(CARTPOS pos, DYNAMIC dyn);

void Circ\_AP\_AP(AXISPOS pos1, AXISPOS pos2);

void Circ\_AP\_CP(AXISPOS pos1, CARTPOS pos2);

void Circ\_CP\_AP(CARTPOS pos1, AXISPOS pos2);

void Circ\_CP\_CP(CARTPOS pos1, CARTPOS pos2);

void Circ\_AP\_AP\_DYN(AXISPOS pos1, AXISPOS pos2, DYNAMIC dyn);

void Circ\_AP\_CP\_DYN(AXISPOS pos1, CARTPOS pos2, DYNAMIC dyn);

void Circ\_CP\_AP\_DYN(CARTPOS pos1, AXISPOS pos2, DYNAMIC dyn);

void Circ\_CP\_CP\_DYN(CARTPOS pos1, CARTPOS pos2, DYNAMIC dyn);

//PTP插补相对偏移指令

void PTPRel\_AP(AXISPOS pos);

void PTPRel\_CP(CARTPOS pos);

void PTPRel\_AP\_DYN(AXISPOS pos, DYNAMIC dyn);

void PTPRel\_CP\_DYN(CARTPOS pos, DYNAMIC dyn);

//Lin插补相对偏移指令

void LinRel\_AP(AXISPOS pos);

void LinRel\_CP(CARTPOS pos);

void LinRel\_AP\_DYN(AXISPOS pos, DYNAMIC dyn);

void LinRel\_CP\_DYN(CARTPOS pos, DYNAMIC dyn);

//停止机器人运动并且丢弃已经计算好的插补路径,stopMode=0时机器人立即停止，stopMode=1时机器人减速停止

void StopRobot(int stopMode = 0);

//标定回零位置，针对单一轴。当机器人到达零点后，保存当前机器人轴位置作为该轴的零位,角度单位为°。如果未设置角度参数，则把当前轴位置设为零位。

void RefRobotAxis(int axis);

void RefRobotAxis(int axis, double angle);

void RefRobotAxis\_DYN(int axis, DYNAMIC dyn);

void RefRobotAxis\_DYN(int axis, double angle, DYNAMIC dyn);

//允许多轴同时回零，把当前机器人所有轴位置作为各轴的零位

void RefRobotAxisAsync();

void RefRobotAxisAsync\_DYN(DYNAMIC dyn);

//对机器人的单一轴进行运动

void MoveRobotAxis(int axis, double angle);

void MoveRobotAxis\_DYN(int axis, double angle, DYNAMIC dyn);

//设置指令组

//设置dynamic参数

void Dyn(DYNAMIC dyn);

//设置机器人运动的动态倍率参数

void Dynovr(int ovr);

//设置速度规划方式，0为S型速度规划（默认），1为梯形速度规划

void Ramp(int type);

//设置参考系，为后续运行的位置指令设定为一个新的参考坐标系，如果程序中没有设定参考坐标系，系统默认参考系为世界坐标系

void Refsys(CARTREFSYS crs);

//设置工具坐标系相对法兰盘的位置

void Tool(CARTSYS cs);

//系统功能指令组

//设置机器人等待时间

void WaitTime(double duration);

//停止所有激活程序的执行，stopMode==0时机器人立即停止，stopMode==1时机器人减速停止

void Stop(int stopMode = 0);

//发出一条信息通知,信息显示信息报告栏中

void Notice(char \*text);

void Notice1(char \*text, auto para1);

void Notice2(char \*text, auto para2);

//发出一条警告信息

void Warning(char \*text);

void Warning1(char \*text, auto para1);

void Warning2(char \*text, auto para2);

//发出一条错误消息

void Error(char \*text);

void Error1(char \*text, auto para1);

void Error2(char \*text, auto para2);