**机器人控制系统设计文档**

**目录**

[1 机器人控制系统总体设计 1](#_Toc43300342)

[2 机器人系统配置和参数管理软件 1](#_Toc43300343)

[2.1 系统配置和参数管理软件总体设计 1](#_Toc43300344)

[2.2 机器人配置模块 3](#_Toc43300345)

[2.3 传感器配置模块 7](#_Toc43300346)

[2.4 监控调试模块 8](#_Toc43300347)

[3 机器人示教器 8](#_Toc43300348)

[3.1 示教器软件结构设计 8](#_Toc43300349)

[3.1 示教器软件界面设计 9](#_Toc43300350)

[4 机器人控制运行系统 11](#_Toc43300351)

[4.1 控制器总体架构设计 11](#_Toc43300352)

[4.2 机器人语言解释器 11](#_Toc43300353)

[4.3 网络通信与数据交互 13](#_Toc43300354)

**1 机器人控制系统总体设计**

本设计的工业机器人控制系统硬件框架如图1.1。整个机器人控制器系统在硬件上主要由示教器、上位机、运行控制器、伺服系统及电机、I/O设备和外部传感器设备组成。运行控制器与伺服系统和I/O设备间采用工业实时以太网总线的形式进行交互，而示教器、上位机以及外部传感器设备与运行控制器之间采用Socket方式进行通信。

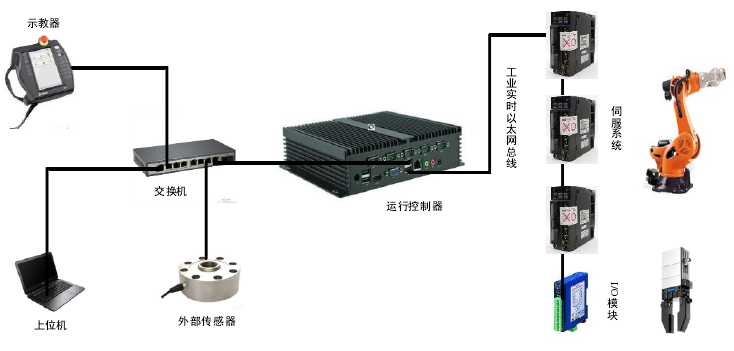


图1.1 机器人控制系统硬件框架

**2 机器人系统配置和参数管理软件**

机器人厂商通常提供两类人机交互平台：示教器和机器人集成开发环境。示教器侧重于在控制器实际运行过程中完成对机器人运动的控制、调试与监控；而机器人集成开发环境侧重于对机器人工程的编辑与配置、控制器系统内用户任务的调试和机器人运动数据的离线调试与分析。本机器人系统配置与参数管理软件作为机器人集成开发环境的子集，设计实现了适用于本控制器系统的机器人系统配置、控制器外部传感器接口配置及对控制器系统运行监控和管理功能。

**2.1 系统配置和参数管理软件总体设计**

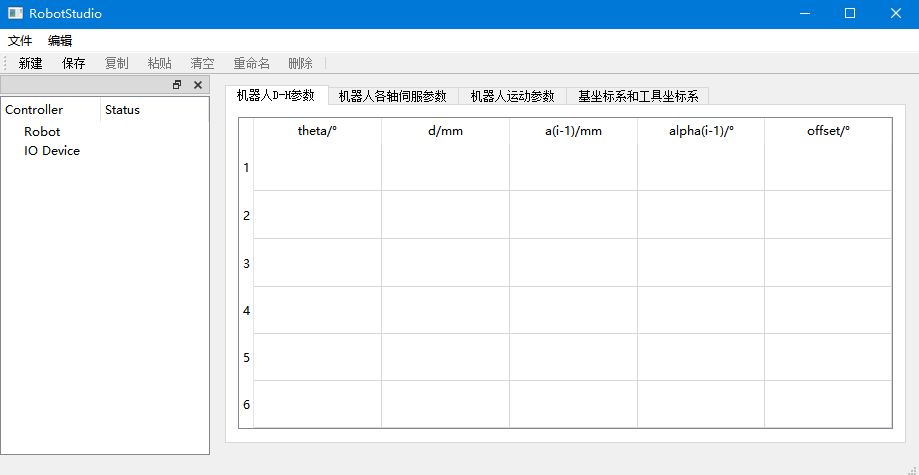
本系统配置与参数管理软件命名为RobotStudio，采用 Qt 框架进行开发，主要运行于 PC 平台。同时出于便于开发和维护的目的，采用模块化的设计思想，按功能划主要划分为机器人配置模块、外部传感器配置模块、监控调试模块和网络通信模块，并运用 Qt 提供的信号-槽机制处理模块内及模块间的通信。

整个配置管理软件包含两个线程，一个为 GUI 主线程，主要完成所有的显示任务和用户的输入响应；另一个为网络通信线程，主要完成系统配置与参数管理软件与控制器系统的所用信息交互任务。图2.1为机器人系统配置和 管理软件总体结构图。机器人系统参数配置模块主要根据机器人本体参数配置生成 RobotConfig.xml 文件，用户可以新建机器人实例或者加载原先创建的机器人实例来开始机器人系统的配置工作；传感器配置模块则用于配置实时通信接口使得控制器运行系统能够与各类传感器通信；监控调试模块用于与控制器运行系统进行数据交互，并将相关数据显示在界面上。



图2.1 机器人系统配置和管理软件总体结构图

整个配置管理软件界面是 Qt 提供的 QMainWindow 类的一个对象，图 2.2 为主界面布局示意图，其中包含了菜单栏、工具栏、状态栏以及用 QDockWidget 和 QTreeWidget 实现的工程树形结构和用QStackedWidget 实现的中心部件，软件中各模块的主要工作区就位于中心部件。



2.2 为主界面布局示意图

可扩展标记语言（xml）文件以纯文本的格式进行存储，提供了一种独立于硬件与软件的数据存储方法，这让创建的不同的应用程序共享数据变得更加容易。基于这个特性，将配置的机器人系统参数以及外部传感器配置信息均以 xml 文件的形式存储，从而控制器系统能够获取系统配置数据。

**2.2 机器人配置模块**

本模块的机器人配置参数包括机器人D-H参数、机器人各轴伺服参数、机器人运动参数以及机器人的基坐标系和工具坐标系，其各子界面如图2.1-2.4所示，界面间的切换使用QTabWidget实现。



图2.1机器人D-H参数子界面



图2.2机器人各轴伺服参数子界面

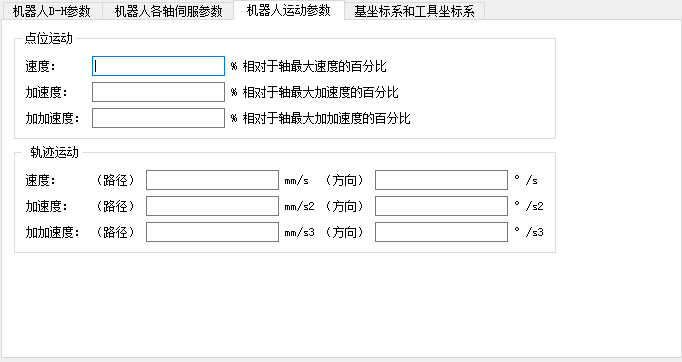


图2.3机器人运动参数子界面

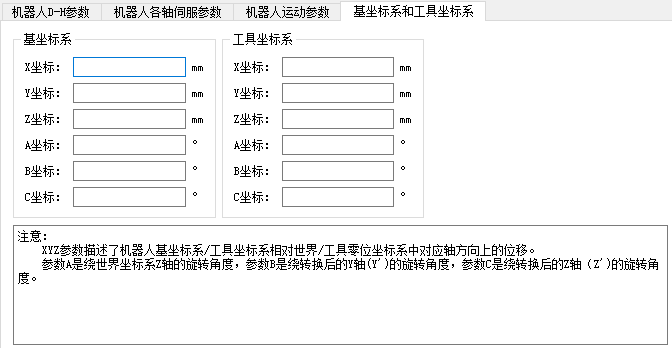


图2.3机器人基坐标系和工具坐标系子界面

机器人本体配置参数如表2.1所示。

表2.1机器人本体配置参数

|  |  |  |
| --- | --- | --- |
| 参数名称 | 配置属性名及参数解释 | 取值说明 |
| 机器人名称 | name：使用的机器人名称 | 无 |
| 机器人类型 | type：机器人类型 | 普通六轴机器人、其他 |
| 机器人D-H参数 | theta：运动关节角  d：机械臂连杆偏距  a：机械臂连杆长度  alpha：机械臂杆件扭角  offset：关节角偏移量 | 根据实际机器人本体参数说明并建立各关节坐标系确定  单位：度、毫米 |
| 机器人各轴极限参数 | maxPos：正向最大角度  minPos：负向最大角度  maxVel：负向最大角度  maxAcc：最大加速度  maxDec：最大减速度  maxJerk：最大加加速度 | 根据机器人电机性能和机械安装确定（由机器人厂商提供准确参数）  单位：度、毫米 |
| offset：关节零点相对伺服零点的偏移量  coupling：各轴间的耦合关系  direction：各关节的方向 | 根据建立的机器人坐标系与伺服电机的零点和运动方向确定 |
| 机器人运动参数 | velAxis：轴运动速度  accAxis：轴运动加速度  jerkAxis=：轴运动加加速度  velPath：路径速度  accPath：路径加速度  jerkPath：路径加速度  velOri：方向速度  accOri：方向加速度  jerkOri：方向加加速度 | 根据实际需要设置机器人运动时的最大速度、加速度、加加速度 |
| 机器人基坐标系相对世界坐标系的位置 | base：基坐标系  tool：工具坐标系 | 根据现场作业环境设置，当极坐标系与世界坐标系重合时，各坐标数值为0 |

机器人系统参数配置模块配置所得的配置文件固定名为 RobotConfig.xml。通过使用TinyXML的类库，可以方便的生成所需的配置文件。RobotConfig.xml 配置文件中每个机器人实例主要包括<BaseInfo>、<DH>、<Servo>、<Dynamic>和<Coordinate>节点，分别对应机器基本信息、机器人D-H参数、机器人各轴极限参数、机器人运动参数和机器人基坐标系相对世界坐标系的位置，其中各个节点的属性即为配置软件中所需用户填写或选择的内容。图2.3为生成的配置文件 RobotConfig.xml 的各部分示例内容。

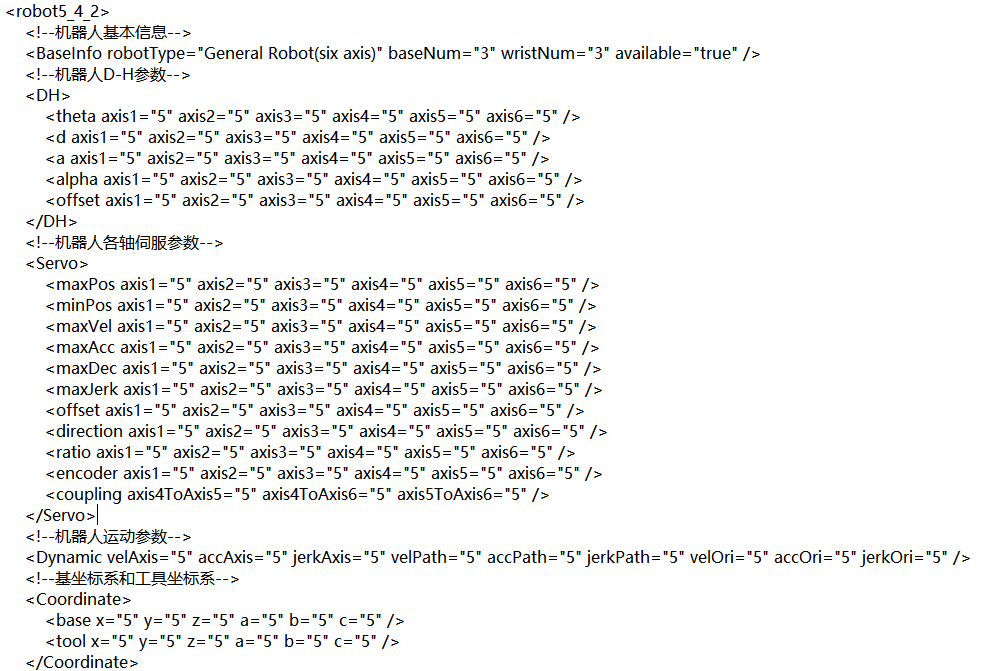


图2.3 配置文件示例图

每次打开RobotStudio软件时，会加载配置文件，更新界面上的数据。在配置过程中，会检测每个机器人实例的所有配置是否都已完成，如果是则在机器人名称后标示“available”，表示该机器人实例可用。除此之外，还可新建多个机器人实例，并能对其进行复制、粘贴、删除等操作，配置文件也会随之更新。

**2.3 传感器配置模块**

**2.4 监控调试模块**

**3 机器人示教器**

**3.1 示教器软件结构设计**

机器人的控制系统采用上下位机结构，其中示教器是上位机，而运动控制器是下位机，两者之间通过Socket进行通信并交互数据，完成对机器人的操作和控制。整个示教器软件设计为七个模块，分别为七个模块，分别为用户自定义界面模块、配置管理模块、变量管理模块、项目管理模块、程序管理模块、位置管理模块、信息报告模块。示教器软件的整体结构以及和下位机的联系如图3.1所示。



图3.1 示教器软件结构及与下位机的联系

在设计示教器软件结构之后，需要对示教器系统软件的运行流程进行规划，本系统中软件的运行流程图如图3.2所示。首先是应用程序的初始化，其中包括人机界面的初始化、通信初始化；然后，创建通信线程用于数据传输，创建按键线程用于数据更新。此后，示教器开始与下位机进行连接通信，这步骤可以从下位机同步获得运动参数、各关节参数、系统等信息，以及一些需要的IO数据。在示教器与下位机进行同步的时候，如果示教器接收到了下位机发送的应答命令，就说明所有参数信息已经同步完成了。通信线程定期获取下位机传送的机器人的位姿等信息，并将这些数据实时显示出来，跟踪机器人的运动轨迹。当有触控操作的时候，示教器则将触控需要完成的命令任务发送给下位机，让其执行。当有下位机数据到来的时候，示教器需要对数据进行处理并显示。

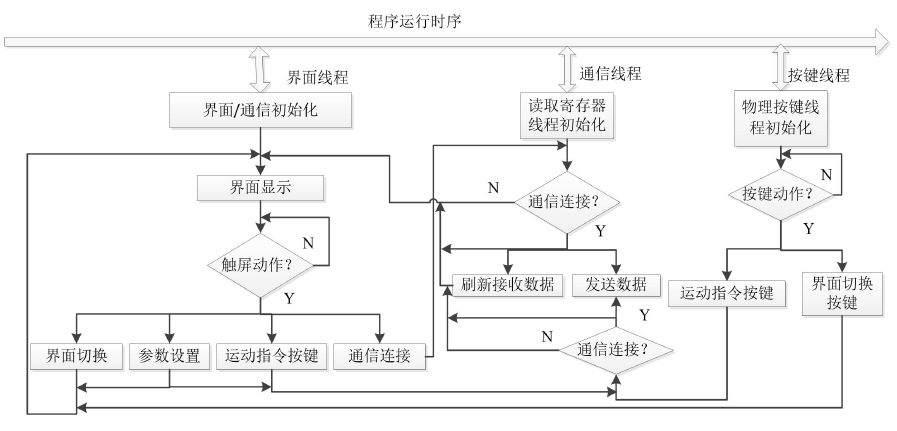


图3.2示教器软件运行流程图

**3.1 示教器软件界面设计**

设计的机器人示教器的主界面如图3.3所示。



图3.3示教器主界面

图左侧分别为状态和配置配置管理部分，而右侧按钮为机器人动作操作按钮，底部的按钮时调节部分。左侧4个状态显示栏表示了系统运行状态，系统正常启动为RUN状态栏变绿，并显示为RUN\_ON，发生错误时EPR\_OFF会变绿，并显示为EPR\_ON。机器人上电时PWR\_OFF会变绿，并显示为PWR\_ON。

左侧7个按钮，分别为7个模块界面的切换按钮，自上而下依次为用户自定义模块、配置管理模块、变量管理模块、项目管理模块、程序管理模块、位置管理模块、信息报告模块。右侧机器人动作操作部分，通过按“+”与“-”按键可以在编程或者点动时调节机器人的坐标位置，点击换页按钮可以翻到下一页(附加轴页)。开始和停止按钮与程序运行和停止有关。

底部伺服上电按钮用于机器人上电或下电，切换坐标按钮用于切换机器人坐标系(轴坐标系、世界坐标系、工具手坐标系)，单步/连续按钮用于切换程序进入单步模式还是连续模式。V-和V+用于调节机器人运动速度。

在主界面顶部，有一个状态栏，它包含了机器人的操作模式、机器人状态及名称、坐标系、运动调节速度、项目程序名称、程序状态及执行模式、急停开关的状态、使用者等级等有关机器人系统状态方面的信息。具体如图3.2所示。其他详细介绍请阅读示教器使用说明书。



图3.4主界面状态栏部分

**4 机器人控制运行系统**

**4.1 控制器总体架构设计**

RobotStudio（机器人系统配置和参数管理软件）新建机器人工程

输入机器人各项参数配置，保存生成xml文件和其他配置文件（外部传感器、通信配置等）

采用FTP文件传输协议将系统配置文件上传至服务端（即控制器），同时也可以从服务端下载文件至本地客户端

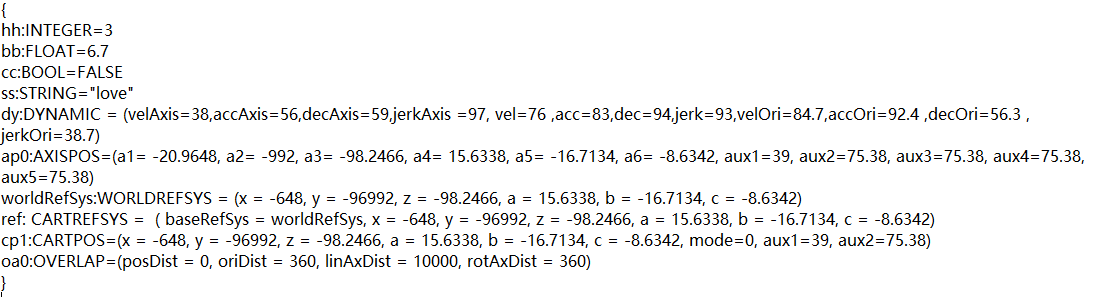
控制器启动后读取配置文件，根据配置文件中的参数初始化，等待示教器端启动，并与示教器进行数据交互，控制机器人运动

**4.2 机器人语言解释器**

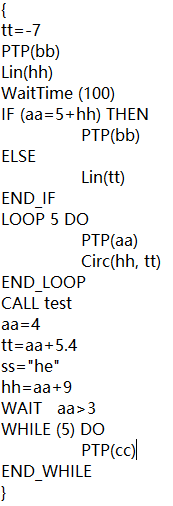
工业机器人控制语言用于编写机器人的示教程序，因此机器人控制语言决定了机器人系统的编程能力，而其编程能力决定了机器人能够提供的功能。

本控制系统根据工业机器人运动控制的需求定义了工业机器人控制语言。工业机器人控制语言包括10种数据类型、15种机器人指令，并且可以继续扩展。然后根据该语言的文法介绍了示教程序解释执行模块的实现：工业机器人控制语言解释器。解释器包括词法分析、语法分析、语义分析等模块，其中词法分析和语法分析借助开源工具flex和bison完成。

数据文件示例如下：



程序文件示例如下：



**4.3 网络通信与数据交互**

在控制器实际运行过程中，需要与示教器交互的数据如下：

//示教器发送给控制器的数据

int enableState = 0; //机器人使能状态，0：未使能，1：使能

int operateMode = 0; //机器人操作模式，0：手动，1：自动

int runState = 0; //机器人运行状态，0：暂停，1：运行中

int step = 0; //程序执行的方式，0：连续，1：单步

int jog = 0; //手动示教时所选坐标系，0：轴关节坐标系，1：世界坐标系，2：工具手坐标系

int coordinate = 0; //手动示教时选中的坐标，0：未示教，1-6：第x个坐标

int upOrDown = 0; //手动示教时对选中坐标的增大或减小，0：减小，1：增大

int percentage = 20; //相对最大运动参数的百分比

int progLine = 0; //程序执行到的行数

string dataContent = "{}"; //data文件的内容

string progContent = "{}"; //prog文件的内容

//控制器发送给示教器的数据

AXISPOS axisPos = { 0 }; //机器人在轴关节坐标系下的位置坐标

CARTPOS cartPos = { 0 }; //机器人在世界坐标系下的位置坐标

DRIVERPOS driverPos = { 0 }; //机器人在伺服编码器中的位置坐标

DYNAMIC dynamic = { 0 };

int hasReachSingularity = 0; //是否到达机器人的奇异点，0：否，1：是

vector<pair<int, string>> infoReport; //信息报告的内容

网络通信连接的流程如下：

