一种三轴写字机械手系统的设计与实现（目录）

# ABSTRACT

# 第一章 绪论

# 第二章 机械手系统设计

## 2.1 机械手硬件模型

## 2.2 机械手软件模型

# 第三章 机械手硬件系统设计

## 3.1 数字信号处理芯片TSM320F812

## 3.2 仿真器

## 3.2 上位机PC

## 3.3 步进电机驱动板

## 3.4 舵机和步进电机

# 第四章 机械手软件系统设计及实现

## 4.1 软件开发环境

### 4.1.1 Microsoft Visual Studio

### 4.1.2 CCS

## 4.2 上位机UI设计及实现

本文希望提供一个友好的交互界面，用来提供任意汉字的输入入口以及一些动作执行时的基本设置。例如，舵机的初始位置决定了写字用笔所在的初始高度位置，在开始写字之前，笔应该是处于抬起状态，而准备写字的时候，笔应该是落下的状态，这些可以通过上位机进行微调。写字的速度对应步进电机的转速，转速越快，完成一笔动作需要的时间则越短。本文提供两种写字模式，分别为快速和慢速，在开始写字之前也可以在上位机对应设置。交互界面中还包括一组按钮，其中有显示数据信息的功能，也有当作开关使用。界面的实现由MFC程序编写，如下图xxxx所示。



图xxxx 上位机交互界面

上图xxxx左下角是一个MFC编辑框，在界面中用于实现显示功能。图中所有的“1”所在的位置代表了“北”字。

### 4.2.1 交互界面使用到的控件

设计界面时，使用到了MFC程序提供的很多控件，包括静态文本框、复选框、按钮、单选按钮、进度条和编辑框。其中，静态文本框可以显示静态的文本信息，还可以实现显示图像，如图中北京大学校徽就是通过静态文本框实现的。滑块用来实现微调的效果，滑动滑块，右边显示的百分比即为当前的调整状态。编辑框一方面用来输入汉字，另一方面用来动态显示汉字对应的点阵信息。按钮则分别对应这某种功能，如显示数据、发送数据及退出等等。空间说明如下表yyyy所示。

表yyyy 界面使用控件说明表

|  |  |  |  |
| --- | --- | --- | --- |
| 对象 | 控件ID | 说明 | 对应变量 |
| 静态文本框 | IDC\_STATIC\_PHOTO | 显示北京大学校徽 |  |
| 静态文本框 | IDC\_STATIC\_TITLE | 显示静态文本内容 |  |
| 编辑框 | IDC\_WORD\_EDIT | 输入汉字入口 | m\_Word/m\_Edit |
| 按钮 | IDC\_APPLY\_BUTTON | 应用，向下位机发送数据 |  |
| 按钮 | IDC\_EXIT\_BUTTON | 退出程序 |  |
| 滑块 | IDC\_SLIDER1 | 设置舵机初始位置 |  |
| 单选按钮 | IDC\_STATIC |  | Speed |
| 快速 | IDC\_FAST\_RADIO | 设置快速模式 | &Fast |
| 慢速 | IDC\_SLOW\_RADIO | 设置慢速模式 | &Slow |
| 设置 | IDC\_SETUP\_BUTTON |  |  |
| 按钮 | IDC\_EDIT\_COMNUM | 输入串行通信端口号 | portnums |
| 按钮 | IDC\_BUTTON\_OPENCOM | 打开串口 |  |
| 按钮 | IDC\_BUTTON\_SHOWDATA | 显示 |  |

### 4.2.2 各控件功能实现

## 4.3 上位机寻址算法

上位机界面提供了友好的用户接入点，是供用户当作控制台使用的。下位机是写字机器人的动作执行控制器，而下位机本身却并不能识别出一个汉字，即用户输入的信息无法直接提供给下位机使用。因此，在给定输入汉字之后，使用输入数据的关键在于先实现输入信息的转换。将用户的输入转换成下位机能直接利用的信息，这部分由上位机的相关算法实现。

### 4.3.1 汉字库

本文所使用到的汉字库是以文本形式存在的一份比较大的文件。

### 4.3.2 深度优先搜索

写汉字的时候，不仅仅要考虑一笔一划的先后顺序，还要保证完整的写完一笔对应的距离。按照汉字的书写习惯，一般从汉字左部或者上部开始书写，对于这点，在搜索“1”的时候可以选择从靠左上的位置开始，比较棘手的地方在于完整的一笔包含了多少个“1”以及这些“1”具体对应的位置。本文采用深度优先搜索算法来获取并记录汉字对应的所有的“1”出现的位置。

深度优先搜索算法是针对图的遍历而引出的一种遍历算法，它的主要思想是以一个未被访问过的点作为起始点，沿该点能连通的边走到未访问过的下一个顶点。在本文中，此处的“边”用四个方向来代替，即上下左右相邻的四个点，若这四个点也为“1”，则表示可以从该点访问到对应相邻点。若访问到某一个点时，当前没有未被访问的点了，则回溯到上一个顶点，继续试探访问别的点，直到所有的顶点都被访问过。

## 4.4 下位机硬件资源驱动

下位机驱动软件主要指驱动DSP2812控制板承载的一些硬件资源，包括设置DSP2812系统时钟频率，驱动并设置GPIO口的输入输出方向，关闭看门狗，驱动DSP2812串口并设置工作方式及波特率，设置DSP2812事件管理器的工作寄存器组、驱动事件管理器生成想要的PWM信号，最后为了协调任务的先后顺序，采用中断的控制方式，因此，还需要设置DSP2812中断控制寄存器。

### 4.4.1设置系统时钟

### 4.4.2 驱动GPIO口

### 4.4.3 关闭看门狗

### 4.4.4 打开串口

### 4.4.5 使用事件管理器A和B

### 4.4.6 中断控制方式

## 4.5 上下位机通信算法

由于DSP2812下位机控制板自带串口，而上位机本身也带有串口资源，本文中上下位机通信通过串口来实现。

## 4.6 下位机动作控制算法

在下位机上电启动之后，首先按一定的先后顺序驱动将会使用到的板载硬件资源，填充相应的寄存器值，生成固定频率的PWM方波信号。然后按照接收到的坐标数据，控制脉冲产生的个数。DSP片上软件流程图如下图xxxx所示。



图xxxx DSP2812片上软件流程图

# 第五章 系统整体调试及实现效果

## 5.1 路径寻址

## 5.2 上下位机通信

## 5.3方波信号产生

## 5.4 动作次序和方向

## 5.5 移动距离

## 5.6 实现效果

# 致谢

# 参考文献