

**毕业设计（论文）(字体)**

**题 目：** 3D打印机及其上位机控制系统设计

学 院**：**  航空制造工程学院

专业名称**：**  机械设计制造及其自动化

班级学号： 15031419

学生姓名： 刘传佳

指导教师： 洪连环

2O19年 六月

# 引言

## 研究背景及意义

3D打印（3D Printing）技术是近几年来兴起的一种三维立体快速成型技术，发展至今已受到人们的广泛关注[1]。近年来，3D打印这项高端的技术已经逐渐融入到人们的生活中[2]，相比于传统机械制造行业的车、铣、钳、焊、铸造等诸多传统加工工艺[3]，3D打印技术具有高效、节能、环保、高精度等诸多优势。[4]所以，集绿色环保与先进制造技术于一身的3D打印技术必将在制造业大放光彩。

3D打印技术在国内掀起了一股创新热，针对产品的3D打印效果展示和3D可视化呈现在国内获得了广泛的应用[5]。机器的大批量生产虽然可以满足人们的正常的生活需求，但是这种满足于正常生活需求的产品是建立在消除个性差异，拔掉冒尖的刺儿的基础之上的，正如上个世纪五十年代计划经济时大工厂为供销社所设计的产品是为了解决温饱问题的一样。如今，改革开放日益深入，中国经济飞速发展，人们日益增长的物质需求和日益丰富的精神需求势必要求个性丰富的产品的出现，最好是个性定制的产品，那些在产品的创新上滞后而跟不上时代的发展的企业已经或在不久的将来被市场所淘汰，所以为了缓和供给与需求之间不平衡的矛盾，3D打印这项革新传统减材加工的增材制造技术为消费者的个性化生活赢得了一步棋。所以，回归3D打印的基础技术讨论和研究具有重要的意义，3D打印的基础技术当然包含它的机械机构本体设计。

3D打印机主要分为工业级和桌面级两种。工业级打印机相比较于桌面级的打印机，精度更高，体积更大，价格更高[5]，相比较之下，桌面级的3D打印机更便于携带，小巧灵活，物美价廉，更适合消费者人群和教学活动。从教学和研究的角度看，选择桌面级的3D打印机的机型作为机械设计的样本将是个不错的选择。

为了方便调试3D打印机的状态和检测其工作过程，有必要设计一个可以与3D打印机的下位机进行交互的上位机系统。实验室虚拟仪器集成环境（LabVIEW），是美国国家仪器公司在20世纪80年代开发的创新软件产品，是图形化软件开发环境。发展至今，LabVIEW主要应用在自动化测试与验证系统、嵌入式检测和控制系统、仪器控制、数据采集与处理领域[6]。它利用简单的图形编程方式代替复杂而烦琐的语言编程，使得工程技术人员可以快速地将自己的程序用图形的形式“画”出来，从而减轻了工程技术人员的工作量，工作效率得到了显著的提高[7]。综上以上优点，使用LabVIEW可以大大降低开发成本和缩短开发周期，同时能增大软件开发的灵活性，其他开发者能更加容易在原来的基础上进一步完善。

## 国内外研究现状

### 国内研究现状

我国对于3D打印技术的研究较国外起步较晚，但随着3D打印技术在国外各领域的应用中所展现的巨大潜力，迅速引起了国内大批科研工作者、工业领域专家的重点关注，并使得我国的3D打印技术得到了迅猛的发展[8]。例如华中科技大学历经十多年研制出全球最大的“3D打印机”，其可加工零件长宽最大尺寸均达到1.2米 [9]；西安交通大学自主研发了三维打印机喷头, 并在光固化成型系统、成型材料等方面也取得了突破性进展, 使其成型精度达到了0.2mm[10]。目前，该技术也在我国得到了广泛的应用，市场占比也在不断增加，但目前我国的 3D 打印技术的发展还不是特别成熟和完善，还是主要应用于科研，没有实现对其的普遍应用，与发达国家相比还是有一定的不足和差距[11]。目前，国产 3D 打印机在打印精度、打印速度、打印尺寸和软件支持等方面还难以满足商用的需求[12]。

国内 3D 打印产业发展呈现加速增长态势，但发展不够均衡，技术侧重点受地域经济影响较为明显。北京、陕西、上海三地是国内 3D 打印专利申请量最多的省市，3D 打印的研究上均侧重于生物体制造、塑料成型、图像数据处理、电数字数据处理[13]。中国沿海城市目前已经将快速成型技术广泛应用于商业，而我国更发达的香港台湾地区3D打印设备的使用更早, 服务领域更为广泛, 但主要用于技术方面, 而不是用来研究3D打印技术[14]。

### 国外研究现状

在国外，3D打印，特别是熔丝制造广泛应用于原型制作和低成本定制部件的制造。然而，目前的应用熔丝制造的3D打印机具有有限的喷嘴状态监测技术，以最小化喷嘴堵塞误差。喷嘴堵塞是当前熔融长丝制造的3D打印机中最重要的工艺误差之一，并且它在几何公差，表面粗糙度和机械性能方面影响原型部件的质量。因此在这种情况下提出了一种加载振动传感器的熔丝3D打印的喷嘴状态监测技术，简要描述如下。首先，在熔融长丝制造挤出机中的支撑液化器的杆安装件被建模为由过程力系统激发的梁。确定边界条件，并分析直接和鲍登类型的熔丝制造挤出机的施加力。其次，设计并制造了一台带有固定挤出机和移动平台的新型3D打印机，用于进行喷嘴状态监测实验。第三，通过降低喷嘴挤出温度来模拟喷嘴堵塞，这导致长丝在喷嘴内壁周围部分凝固。第四，通过Direct和Bowden类型的熔融长丝制造挤出机在挤出聚乳酸，丙烯腈-丁二烯-苯乙烯和SemiFlex长丝期间测量棒料的振动来进行多组实验。当前研究的结果表明，通过测量挤出机的杆安装振动，可以使用加速计传感器监测熔融长丝制造3D打印机中的喷嘴堵塞。所提出的技术可以有效地用于监视熔融长丝制造3D打印机中的喷嘴堵塞，因为它基于基本过程建模[15]。

巨大的市场潜力以及增材制造技术的经济，地缘政治和其他影响，将不可避免地引起攻击分子的注意，从个人到国家行为者。由于3D打印机依赖于[计算机化](https://www.sciencedirect.com/topics/computer-science/computerization)，因此容易受到各种攻击。这是“野外”检测到的[网络物理系统](https://www.sciencedirect.com/topics/engineering/cyber-physical-systems)的一系列攻击以及研究文献中假设的攻击所支持的。其中包括对[工业控制](https://www.sciencedirect.com/topics/engineering/industrial-control)系统的攻击，最先进的技术[汽车](https://www.sciencedirect.com/topics/social-sciences/motor-vehicles)和无人驾驶和载人航空系统。从这些例子得出的结论是，对增材制造系统和增材制造技术的滥用就在眼前。工业级的3D打印机越来越多地用于为[重要系统](https://www.sciencedirect.com/topics/computer-science/important-system)制造[功能性部件](https://www.sciencedirect.com/topics/computer-science/functional-part)。但是，由于3D打印机依赖于[计算机化](https://www.sciencedirect.com/topics/computer-science/computerization)而容易受到各种攻击。更重要的是，3D打印机本身不是目标，而是将该3D打印机作为启动后续攻击的临时点[16]。

增材制造是一种很有前景但非常危险的技术。研究机构和供应商社区必须更加关注[防御战略](https://www.sciencedirect.com/topics/computer-science/defensive-strategy)和机制的发展，以减轻3D打印机用作武器时可能产生的严重影响[16]。

## 课题的设计内容及方案

### 课题的设计内容

本设计主要负责机械结构本体设计，LabVIEW设计上位机控制系统设计，G代码解析工作，上位机与上位机对接工作。主要研究内容有以下几个方面：

（1）设计3D打印机的机械结构；

（4）G代码解析；

（3）上位机界面设计和界面服务程序设计；

（4）上位机与下位机通信对接。

### 课题分析与设计方案

（1）课题分析

1. 设计3D打印机的机械结构

首先搭建一个小三轴运动平台，然后将小三轴运动平台改装为简单的3D打印机平台，只需要将3D打印所需的熔丝的挤出机加装到小三轴的一根横梁上。安装在小三轴设备上的一个挤出头在理论上可以在三维空间上到达任意一个点，如果把所需要的打印的三维图形分解成一系列的在运动空间上连续的点，那么经过挤出头挤出的熔丝点在空间中按照一定规则连续运动时去填充这些点而堆积三维实体就可以实现打印肉眼可见的三维模型。

1. G代码解析

下位机的运动的基础是一系列连续的坐标点，如果没有被提供这些点，这个3D打印机械设备也就是如同失去灵魂的空壳子。这些点经过整合成一个文件就是需要上位机提供给下位机的轨迹坐标文件，而现有的文件只有G代码文件，从这里可以看出必须设计一个独立的G代码解析算法模块，其输入是G代码文件，输出是轨迹坐标文件。所以，如何编写G代码解析算法将是3D打印机的上位机设计的必由之路，也是核心所在。

G代码解析算法的设计将分成两个部分进行，第一，从G代码中获取关键信息；第二，设计插补算法，将获取的关键信息作为其输入。

1. 上位机界面和界面服务程序设计

上位机与下位机通信采用串口通信协议，所以在界面上需要串口设置区，同时为了将向下位机传送轨迹坐标文件，需要添加选择轨迹坐标文件的控件，除此之外，为了突出上位机监视功能和上位机与下位机的交互作用，有必要添加显示运动状态的控件，也就是需要显示坐标，还要添加一些三轴快速移动的按钮。为了方便学习，讨论和研究，添加读和写缓冲区，这样就可以清楚直观地知道串口通信过程中所传输的数据。

1. 上位机与下位机通信对接

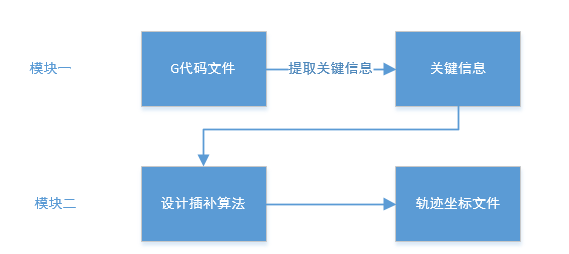
设计上位机与下位机的通信命令码。

（2）设计所需的系统环境与软件说明

1. 系统要求：Windows 10 X64；
2. 三维制图软件：SolidWorks 2016；
3. 上位机设计软件：LabVIEW 2018;
4. G代码解析算法设计软件与语言：Visual Studio 2019与C语言；
5. 代码托管平台：Github;
6. 毕业设计项目托管网址：https://github.com/Twsa/Graduation-Project/tree/master/3Dprinter\_labview。

（3）G代码解析算法设计

G代码解析算法将分成两个子模块进行设计，第一个子模块的功能是从G代码文件中获取关键信息，例如从G代码“G01 X200 Y300”取出关键信息“200，300”；第二个子模块的功能是利用关键信息进行插补，也就是设计插补算法。模块之间的连接关系见图1.3。



**图1.3 G代码解析算法模块**

（4）设计上位机与下位机的通信命令码

上位机与下位机的通信命令码包括发送的命令与接收的命令。

发送的命令：1）X、Y、Z轴快移命令；2）运动轨迹坐标文件。

接收的命令：1）指示发送运动轨迹坐标文件的命令；2）运动状态信息。

（5）上位机界面设计

按照（1）中的分析，这里提供概念设计方案如图1.1。

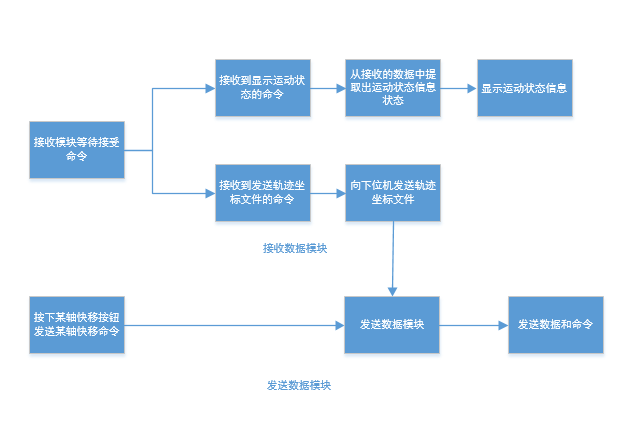


**图1.1 上位机界面概念设计**

（6）上位机界面的服务程序设计

上位机界面的服务程序主要包括两个部分，一个是接收模块，另一个是发送模块，上位机发送轨迹坐标文件给下位机是以上位机接收到下位机的传给上位机的发送轨迹坐标文件的命令为前提的。接收模块除了接收发送文件的命令码，还会接收显示运动状态的信息的命令码，这个命令码中就包含有运动状态的信息，如果接收模块接收到该命令码，接下就可以从该命令码中提取数据并显示在显示控件中。

上面所述的服务逻辑可见图1.2。

**图1.2 上位机界面后台服务逻辑**

（7）调试仿真

各个模块分开调试仿真，调试通过后组装各个模块，然后进行总装后的调试与仿真。调试仿真通过之后然后进行与下位机的对接工作。

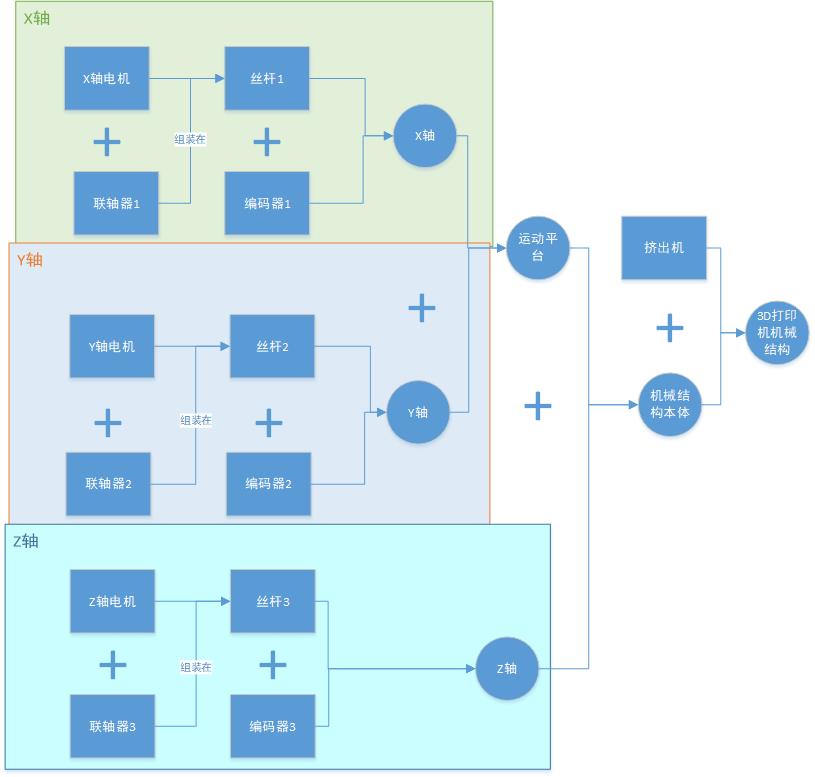
（8）与下位机握手对接调试

进行最终的在实物的调试，首先测试串口通信的是否正常对接，其次测试通信命令码的在实现操作上的正确性，最后完成在实物上的调试。

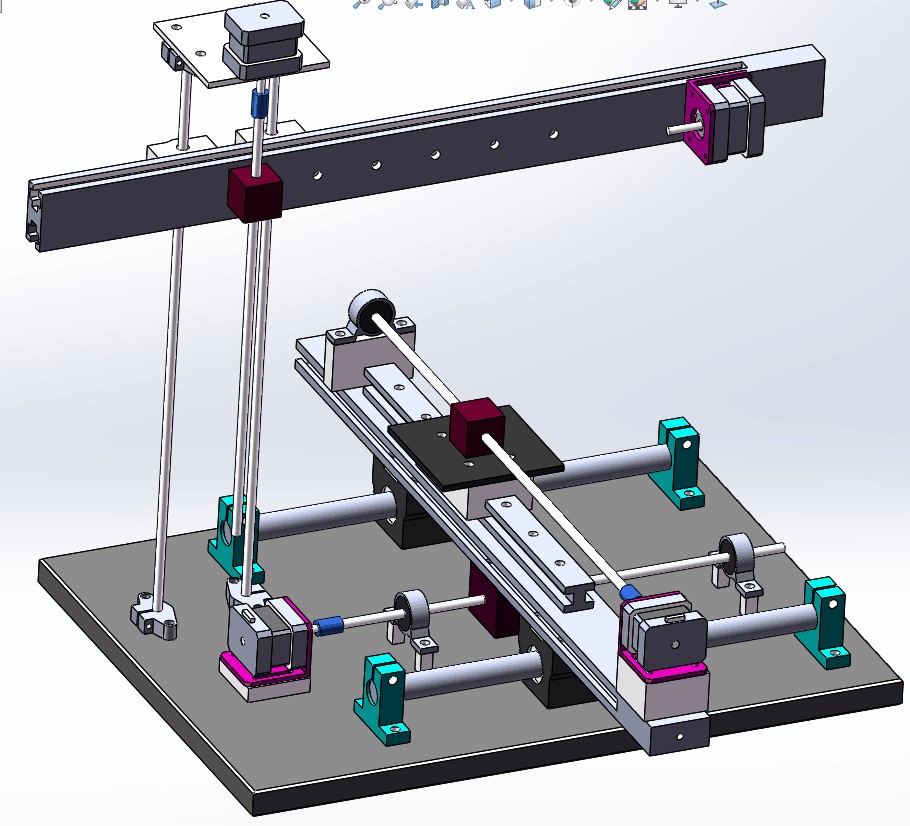
# 机械结构设计

## 2.1 机械结构设计概述

机械结构设计的流程见图2.1，从图2.1中可以看出，机械结构设计主要是设计机械结构本体，机械结构本体设计分成三个部分进行，这三个部分就是分别设计三轴运动平台的基本组成的X、Y、Z轴。其中每一个轴上的零件都是相同的配置，这些零件有电机，联轴器，丝杆和编码器。联轴器安在电机的轴末端成为一个电机-联轴器单元。电机-联轴器单元安装在丝杆上，然后在丝杆的末端套上编码器就构成了某一转轴。三个转轴组装完成后，然后将X轴搭在Y轴上组成一个平面运动平台。运动平台在加装在Z轴上就成了可以升降的平台，也就完成了机械本体结构的组装。最后机械本体结构的X轴的末端加装挤出机，这就完成了简易的3D打印机的机械结构的组装。组装完成的效果图可见图2.2的机械结构的装配图。



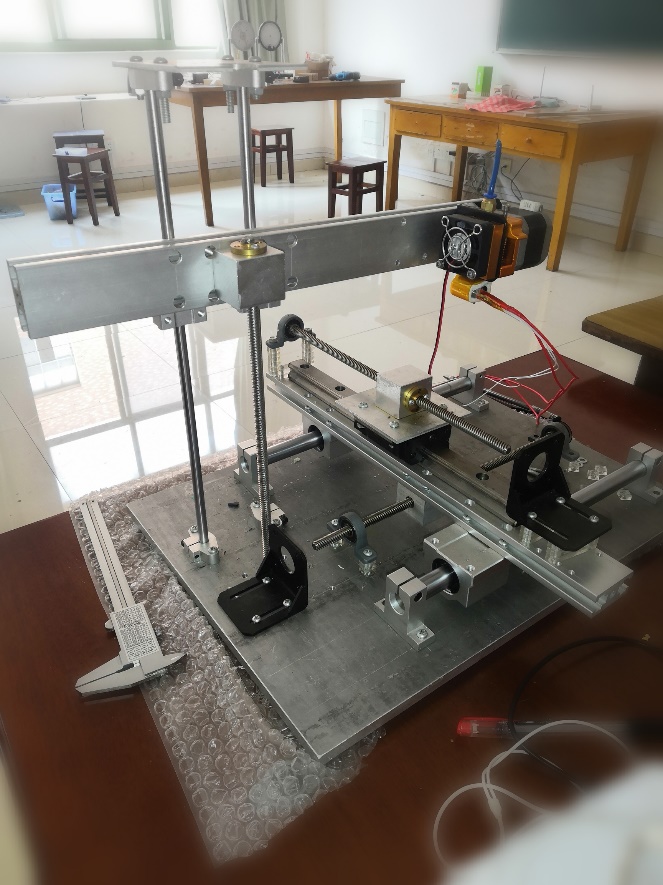
**图2.1 机械结构设计框图**



**图2.2 机械结构的装配图**

图2.3是按照设计制作而成的实物图。X轴位于最下面，Y轴与X轴相垂直且在X轴上面随X轴运动。Z轴采用悬臂结构可上下运动。

X轴和Z轴导轨选用标准的光杆滑轨，Y轴导轨选用直线导轨，底板选用了一块10mm厚度的硬铝板，Z轴悬臂和Y轴基座使用2040型铝合金型材，机械部分剩余的设计工作即为将这些零件按功能要求连接。



**图2.3 机械结构实物图**

## 2.2 机械本体结构各部件的制作

机械装置各组成部件的制作过程具体如下：

（1）X轴的制作：X轴由两根光杆导轨支座，步进电机电机座和丝杆轴承组成。

（2）底板的制作：底板的功能是固定X轴两根光杆导轨的支座，两根Z轴光杆导轨的支座，X轴步进电机电机座和X轴丝杆轴承座。利用所选零件的安装孔位与尺寸，对底板进行钻孔加工，并且对孔洞进行攻丝螺纹。

（3）Y轴横梁的制作：Y轴横梁需要连接X轴两根光杆导轨的滑块，并在其上面安装直线导轨，电机座和丝杆轴承座。利用以上零件的安装孔位，对Y轴横梁进行钻孔加工。

（4）Z轴悬臂梁的制作：Z轴悬臂梁一端由Z轴两根光杆导轨的滑块安装固定。根据Z轴滑块的安装孔位尺寸，对Z轴悬臂梁进行钻孔加工。

（5）Z轴上顶板的制作：Z轴上顶板安装在Z轴两根竖直光杆的顶端，Ｚ轴上顶板安装Ｚ轴电机，故Z轴上顶板根据光杆的安装孔位与电机安装孔位钻孔加工。

## 2.3 机械本体结构各部件的装配

根据装配图完成机械本体的装配。为了装置的配合良好，保证导轨的运行顺畅且基准正确，在安装时需要做到如下三点：

（1）X轴光杆导轨的安装：一根作为X轴主导轨，另一根作为副导轨，主导轨安装有两个滑块，作为滑动基准，副导轨安装一个滑块，作为辅助。安装时，不断调节副导轨螺钉安装间隙位置以达到X轴导轨最好的顺畅效果且不改变主导轨位置。

（2）X轴丝杆与轴承座的安装：利用安装孔的间隙，不断调整轴承座的位置，直到顺畅转动并带动Y轴横梁移动。

（3）Z轴的安装同Ｘ轴导轨的安装：分为主导轨与副导轨，以主导轨为基准进行安装。

（4）挤出机的安装：在X轴上钻两个孔，作为固定挤出机固定架固定所用，把将固定架安装在X轴上，装上挤出机。

至此，机械结构设计的工作完成。

# 3．G代码解析算法设计

## 3.1 G代码解析概述

数控代码由数控指令构成，而G代码也是一种数控程序指令，是一种按照严格规则编写的数控程序语言。一般情况下，一条指令只能执行一个动作，若干条指令以一定的逻辑结构组成一个程序段，完成某种加工操作。因为在数控机床的加工过程中，数控程序虽然涉及到多种功能指令自，但是起主要功能的是G功能指令字，所以数控代码也简称G代码[36]

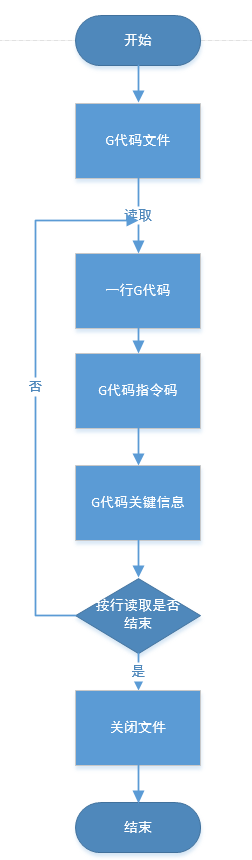
G代码指令一般由特定的功能指令字（一般大写）外加阿拉伯数字组。[37]本文只涉及简单的，常用的G代码。本设计只使用了直线插补G01指令,和圆弧插补G02、G03指令。G代码解析算法的设计首先需要将G代码中的关键信息提取出来，然后利用关键信息设计插补算法。

插补一般建立在增量插补的基础上，常用的由脉冲增量插补和数据采样插补。所谓脉冲增量插补（又称基准脉冲插补或行程标量插补），就是每次插补结束仅向各运动坐标输出一个控制脉冲，因此各坐标仅产生一个脉冲当量或行程增量。脉冲增量插补方法很多，应用较多的是逐点比较法和数字积分法。数字采样插补算法又称为数据增量插补、时间分割插补或时间标量插补，主要又直线函数法、扩展数字积分法、二阶递归算法等。[38]

本文采用了逐点比较法插补算法。逐点比较法，又称醉步式近似法，即走一步看一看，边找边走，宛如醉人的脚步，其基本思想是被控对象按要求的轨迹运动时。每走一步都要和规定的轨迹比较，一句比较的结果决定下一移动的方向，使被对象向减小偏差的方向并趋向终点移动。逐点比较法是以折线来逼近给定轨迹的，只要将脉冲当量取得足够小，就可以达到精度的要求。下面就介绍G代码的解析算法设计过程。

## 3.2 G代码关键信息提取程序设计

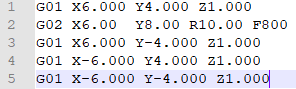
按照如图3.1的过程图提取G代码的关键信息，首先准备一个G代码文件，然后按行读取到一行G代码，然后从这行G代码中读出其G代码指令，接着按照这个G代码指令的对应格式提取出G代码的关键信息，然后判断按行读取是否结束，如果G代码文件全部被读取就结束G代码的读取，如果没有就紧接着上一行读取下一行。结束对文件的操作后关闭文件。下面将分为几个子模块分别进行讨论，它们分别是G代码文件准备与读入，按行读取G代码，获取G代码指令和关键信息，关闭文件。



**图3.1 G代码关键信息提取程序过程**

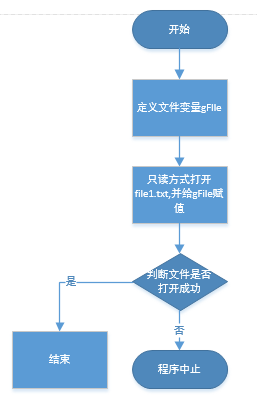
1. G代码文件的准备与读入

为测试准备的G代码文件是一个名为file1.txt文件，如图3.2为filel.txt文件写入的G代码内容。从图3.2可以看到有直线插补指令G01和圆弧插补G02，满足设计的要求。

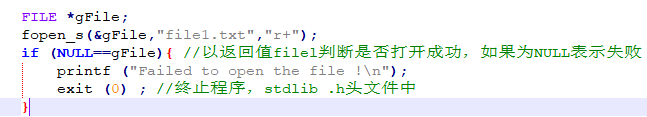


**图3.2 G代码文件的准备**

然后需要在C程序中实现对G代码文件的读入。如图3.4是G代码文件读入程序段，其流程图可见图3.3。该程序段首先定义了一个文件类型的指针变量，使用fopen\_s库函数打开文件，文件的第三个参数的设置为“+r”，说明对文件的操作为只读以防止G代码文件被修改。接下来判断文件是否打开，如果文件打开继续执行后面的程序，如果没有正常打开，那么打印错误信息，中止程序。



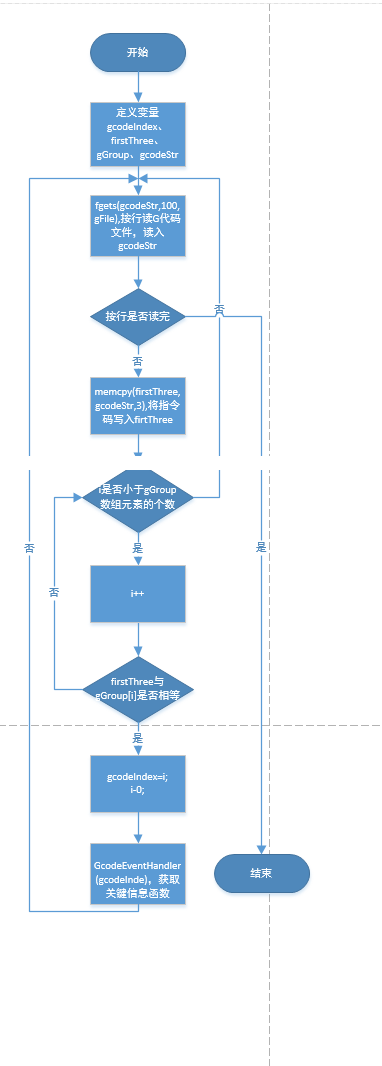
**图3.3 G代码文件读入程序段流程图**

**图3.4 G代码文件读入程序段**

（1）按行读取G代码



**图3.4 按行读取G代码程序段**



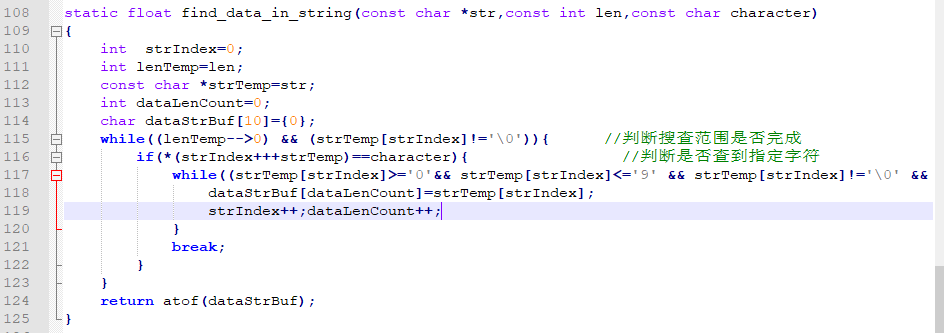
**图3.5 按行读取G代码流程图**

如图3.4是按行读取G代码程序段，图3.5是按行读取G代码流程图这段程序首先按行读取G代码文件，然后判断G代码按行读取是否完成，如果完成就结束，如果没有完成就从读出来的一行G代码提取指令码，然后在数组gGroup中搜索该指令码的类型并获取该指令码在数组中的下标，该下标在读取关键信息的函数中作为参数输入。这就完成了一行函数的读取工作，读到一行G代码，就调用一个读取关键信息的函数，紧接着读取下一行G代码，重复上述操作直至按行读取完成。

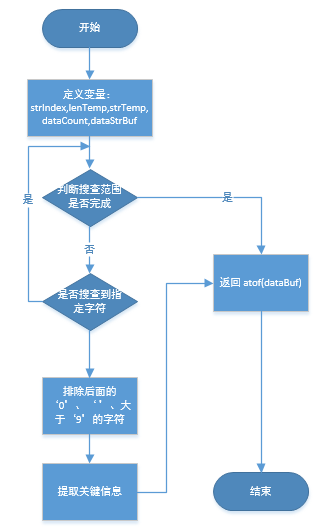
（3）获取G代码关键信息

获取G代码关键信息的函数是find\_data\_in\_string，下面将注重分析这个函数。该函数实现如图3.6。该函数实现的流程图如图3.7。

find\_data\_in\_string函数实现的功能是在一个字符串中获取特定的标志头后面的数据，这个函数有3个参数分别是str,len和character，str代表需要读取的G代码字符串，len代表需要读取的字符串的长度，character代表需要获取的数据前置标识字符。比如G代码“G01 X200 Y200 Z100”，str代表这个整段字符串，len就是这个字符串的长度14，如果想获取X轴的坐标，那么character就是‘X’，如果需要将X、Y、Z轴的坐标全部提取出来，只要重复调用这个函数，下面就介绍这个函数的工作流程。

 首先进入第一个循环，循环条件是判断搜查的字符串的是否搜查完成，也就是检索是否到达末端。如果超出搜查范围，则立马结束，否则继续判断是否查找指定的标志字符，如果没有找到，继续上述的查找；如果找到，提取标志字符后面的数据，提取后面的数据是需要保值数据是数字，排除其他类型的数据。数据提取成功后返回关键信息数据，结束函数。

**图3.6 find\_data\_string函数实现**



**图3.7 find\_data\_string函数实现流程图**

至此，G代码关键信息提取工作可以告一段落了。接下来是利用G代码中的关键信息设计逐点插补算法（直线插补和圆弧插补）算法。

## 3.3 插补算法设计

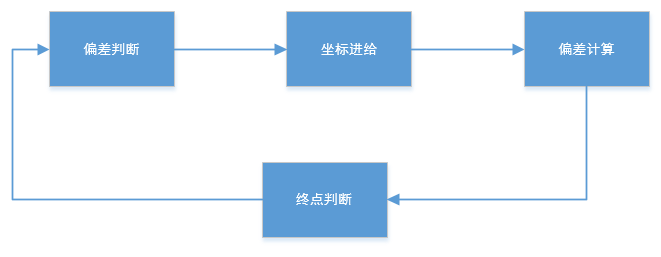
本设计采用的是逐点比较法插补算法（直线插补和圆弧插补）。逐点比较法的插补过程可按以下四个步骤进行：

（1）偏差判断：根据被控对象当前位置是在给定轮廓上，还是在给定轮廓上方或下方确定进给方向。

（2）坐标进给：使被控对象向给定堆积趋近，即向减少误差方向移动。

（3）偏差计算：计算被控对象新位置与给定轨迹之间的偏差，作为下一步判别的依据。

（4）终点判断：判断是否到达终点，若到达，结束插补；否则，继续以上四个步骤（图3.8）。[38]

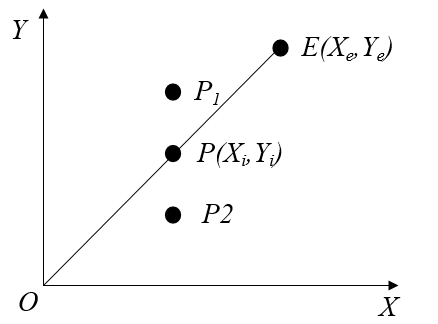


**图3.8 逐点比较法插补过程**

### 3.3.1 逐点比较法直线插补

（1）偏差计算公式

假设给定的轨迹为图3.9所示的第一象限的直线段*OE*，取其起点*O*为坐标原点，终点坐标*A(Xe,Ye)*已知，*P*点*(Xi,Yi)*为动点。



**图3.8 第一象限直线插补**

若点*P*在直线段*OE*上面，则下列关系成立:

|  |  |  |
| --- | --- | --- |
|  |  | （3.1） |

由此可以定义直线插补的偏差函数如下：

|  |  |  |
| --- | --- | --- |
|  |  | （3.2） |

当Fi≥0时，表明P点在OE直线段上或在OE直线段上方，应沿+X方向进给一步，步进后的新的坐标值为

|  |  |  |
| --- | --- | --- |
|  |  | （3.3） |

新的偏差为

|  |  |  |
| --- | --- | --- |
|  |  | （3.4） |

若Fi＜0表明P点在OE直线段下方，应向+Y方向进给一步，步进后的新坐标值为

|  |  |  |
| --- | --- | --- |
|  |  | （3.5） |

新的偏差为

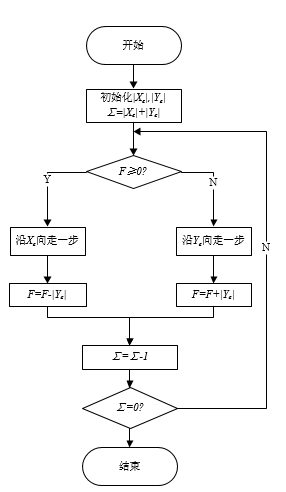
|  |  |  |
| --- | --- | --- |
|  |  | （3.6） |

（2）终点判别法

逐点比较的终点判别有很多方法，下面只介绍算法中所使用的方法。这种方法就是用一个终点计数器寄存X和Y两个坐标，从起点到达终点的总步数和∑=|Xe|+|Ye|。当在X或者Y坐标方向进给时，计数长度减1，当计数长度减到0时，即 =0式，到达终点，停止插补。

（3）不同象限的直线插补计算

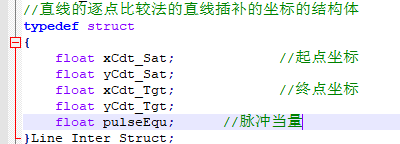
上面讨论的为第一象限的直线插补计算方法，其他三象限的直线插补计算可使用相同原理。图3.9所示为四象限直线插补流程图。



**图3.9 四象限直线插补流程图**

（4）程序设计

如图3.10，先定义一个直线插补的数据结构，该结构中包含有起点坐标，终点坐标和脉冲当量。



**图3.10 逐点比较法直线插补数据结构**

然后编写逐点比较法直线插补算法函数，算法函数如图3.11。

**图3.11 逐点比较法直线插补函数**

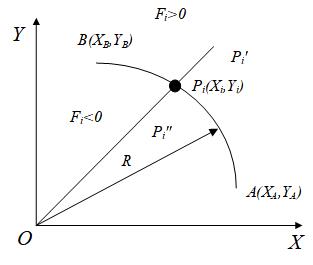
### 3.3.2 逐点比较法圆弧插补

（1）偏差计算公式

若加工半径为R的圆弧AB，将坐标原点定在圆心上，如图3.10所示。对于任意加工点Pi（Xi，Yi），其偏差函数Fi，可表示为

|  |  |  |
| --- | --- | --- |
|  |  | （3.7） |

为了简化公式（3.7）的计算；要采用递推式或迭代式。圆弧加工可分为顺时针加工或逆时针加工，以此相对应的便有逆圆插补和顺圆插补两种方式，本文在设计算法时只设计了逆圆插补的情况。下面就是第一象限的圆弧（图3.10），对其偏差函数的递推公式加以推导。



**图3.12 逐点比较法圆弧插补**

若Fi≥0，应向-X方向进一步，到达Pi+1点，其坐标值为Xi+1=Xi-1，Yi+1=Yi。新点的偏差为：

|  |  |  |
| --- | --- | --- |
|  |  | （3.8） |

若Fi＜0,应向+Y方向进给一步，到Pi+1点，其坐标值为Xi+1=Xi，Yi+1=Yi+1。新点的坐标的偏差：

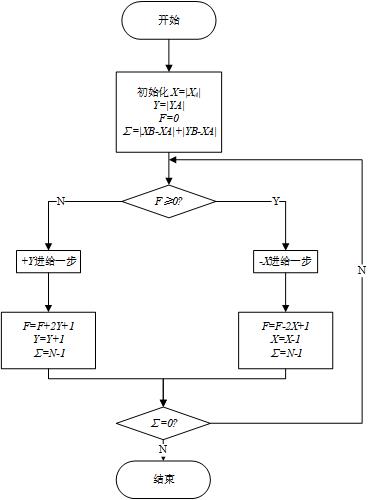
|  |  |  |
| --- | --- | --- |
|  |  | （3.9） |

（2）不同象限圆弧插补

上面讨论的时第一象限圆弧的逆圆插补，其他三个象限的圆弧逆圆插补可以采用相同的原理进行操偶组，如图3.12是四象限圆弧插补流程图。

（3）终点判别方法

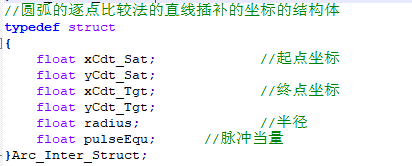
圆弧插补的终点判别方法与直线插补相同。可将从起点到达终点的X、Y轴进给步数的总和存入一个计数器∑=|XB-XA|+|YB-YA|,每进给一步，从∑中减去1，当∑=0时发出终点到达信号。



**图3.13 四象限圆弧插补流程**

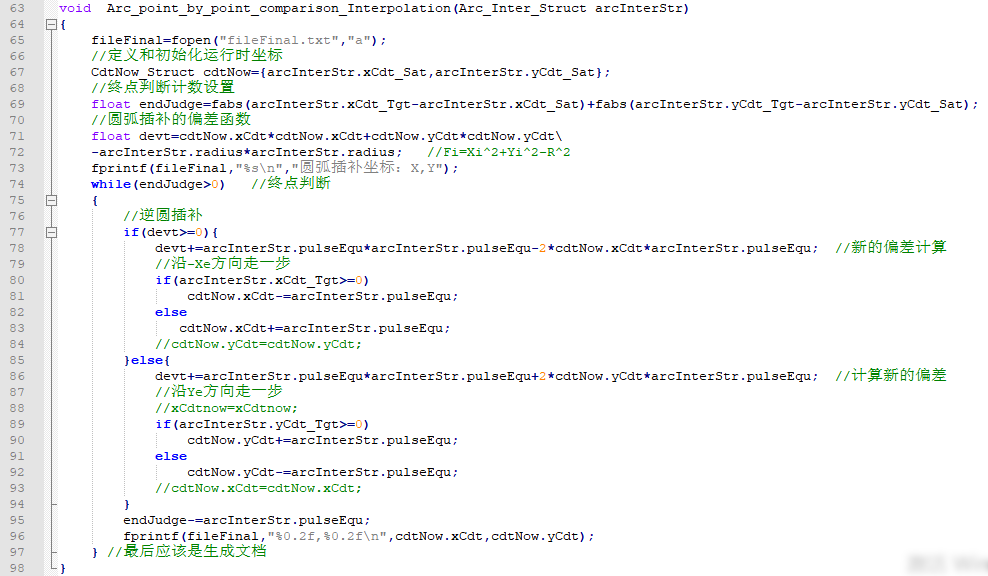
（4）程序设计

如图3.14先定义一个圆弧插补的数据结构，该数据结构包含有起点坐标，终点坐标，半径和脉冲当量。



**图3.14 圆弧插补数据结构**

然后编写逐点比较法圆弧插补逆圆插补算法函数，算法函数如图3.15。



**图3.14 圆弧插补逆圆插补算法函数**

至此，插补算法设计工作完成。

# 4．设计上位机与下位机通信命令码

（1）下位机向上位机发送的命令码

1）接受轨迹坐标文件命令：1$

2）发送运动状态命令：2$X21,Y31.00,Z42.00,F300.00

（2）上位机向下位机发送的命令码：

1）X轴快移：Xmove

2）Y轴快移：Y\_move

3）Z轴快移：Z\_Move

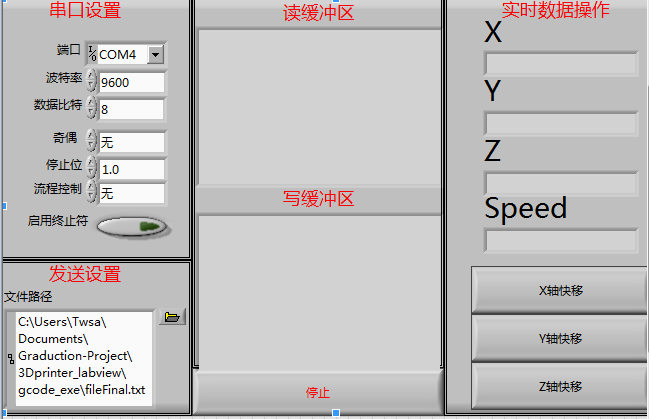
4）运动轨迹坐标文件

# 5．上位机界面设计和界面服务程序设计

## 5.1上位机界面设计

### 5.1.1 上位机界面设计概述

上位机的界面设计如图3.1所示，从设计的界面可以看出，整个界面被分成5个部件，这5个部件分别是串口设置，发送设置，读缓冲区，写缓冲区和实时数据操作。串口设置部件是实现对串口的初始化所设计的控件布局；发送设置部分实现就是查找轨迹坐标文件的路径这一个控件的布局；读、写缓冲区部件分别是显示在串口通信中的数据的接收、发送的数据；实时数据操作这个部分实现对状态信息显示控件和操作下位机快速移动的控件的实现。



**图4.1 上位机界面**

### 5.1.2 上位机界面各部件的制作与总装

在LabVIEW中新建工程，保存工程并命名为final.lvproj，在工程中新建vi文件，保存并命名为final.vi，上位机的界面设计就在final.vi 的前面板进行。

（1）串口设置部件的制作

如图3.2是串口设置部件的界面设计布局，从图中可以看到设计上所采用的背景以灰色色调为主。这种灰色色调的背景可以通过LabVIEW中修饰控件来实现，可以从在LabVIEW的前面板的空白区域右击而出现的控件的对话框中选择修饰控件，修饰控件有许多不同的样式可以选择，本设计全部采用“垂直平滑盒”这个样式。添加完修饰控件后要记得将修饰控件向后方移动，因为端口操作相关的控件需要添加在修饰的上面，也就是在视图的位置处于修饰控件的前方，如果修饰控件没有向后面移动，端口操作相关的控件是会被修饰控件所覆盖，也就看不到了。为了实现修饰控件向后面的移动，需要点击菜单栏中重新排序的按钮下拉中的移至后面，当然也可以将串口操作相关的控件移至前面，只是你会发现因串口相关的控件较多而繁杂无聊罢了。

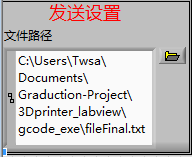
从图3.2中可以看出串口操作相关控件需要7个，其中只有三类控件，端口是一类，波特率、数据比特、奇偶、停止位、流程控制是一类，还有一类是启动终止符。端口选择的控件是I/O控件类中VIS A资源名称控件；第二类选择的控件是数值控件类中的数值输入控件；启动终止符选择的是布尔控件类中的开关按钮控件。最后将各个控件按照如图3.2所示的顺序布局。其他部件的制造除了选择的控件不同之外，其他操作一样，所以在下面的关于其他部件的制作的描述中将着重于控件的选择。



**图4.2 串口设置部件**

（2）发送设置部件的制作

如图3.3是发送设置部件。发送设置部件的控件只有一个，这个控件就是字符串与路径控件类中文件路径输入控件。



**图4.3发送设置部件**

（3）读、写缓冲区部件的制作

如图3.4是读、写缓冲区部件。读、写缓冲区部件的两个控件是一样的，都是字符串与路径控件类中的字符串显示控件。



**图4.4 读、写缓冲区部件**

（4）实时数据操作部件的制作

如图3.5是 实时数据操作部件。这个部件由两种控件组成，X、Y、Z轴的数据显示控件是字符串与路径控件类中的字符串显示控件，X、Y、Z轴的快速移动按钮控件式是布尔控件类中的确定按钮。其中需要说明的是需要将确定按钮的属性中的操作由默认的选项改成“保持转换直至释放”，这样就可以实现长按按钮而实现快速移动而不至于按一下按钮只移动一个步距。



**图4.5 实时数据操作部件**

（4）上位机界面各部件总装

首先需要添加一个各个部件总装的背景，同样选择“垂直平滑盒子”修饰样式作为背景，同时需要将背景移至最后面，然后将各个部件按图3.1的布局安放好，将部件安放在恰当的位置之后用修饰控件类里的粗分隔线将各个部件分隔开，这就完成了上位机界面的各个部件的总装。

至此上位机的界面设计完成。

## 5.2 上位机界面服务程序设计

### 5.2.1 上位机界面服务程序设计概述

上位机与下位的交互采用了串口通信协议。上位机与下位机的交互分成读和写这两种的数据操作模式，所以在设计上位机的服务程序时就这两种数据的操作模式设计了两个模块，就是读数据模块和写数据模块。其余的数据处理操作都是围绕着串口的读和写展开的，所以串口的读与写将是展开其他工作的前提和基本。

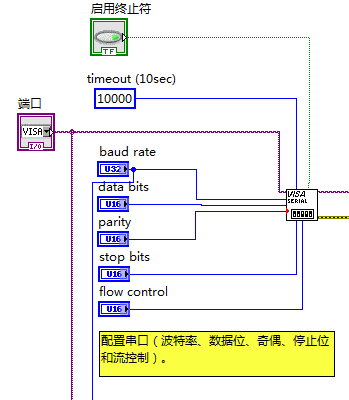
在本设计中利用了VISA对串口进行操作，VISA（Virtual Instruments Software Architecture，VISA虚拟仪器软件架构是NI公司为了确保多个厂商具有协同工作能力以及降低仪器设备的开发时间而联合其他许多虚拟仪器厂商在1993年推出的一个高效的，具有标准性质的软件架构，从而使得一套仪器控制程序适用于任何硬件平台成为可能，使得开发者不需要关注特定硬件平台的架构，只需要调用VISA结构，并为特定平台配置相应的参数即可，减轻了开发者的负担。

       通过VISA用户能与大多数仪器总线连接，包括GPIB、USB、串口等等，无论底层是何种硬件接口，用户只需要面对统一的编程接口—VISA。在这里将介绍如何使用VISA对串口进行操作。

### 5.2.2读数据模块

（1）设置串口

首先将VISA串口模块调用出来，配置串口的相应的设置如图4.6。这些设置有选择端口，是否启动终止符，超时检测，波特率，数据比特，停止位，流控制和奇偶校验。

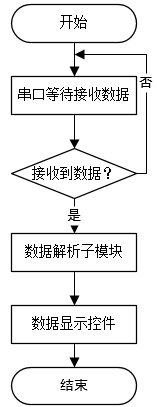


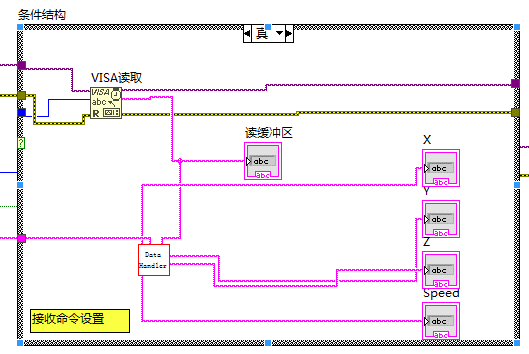
**图4.6 配置串口**

（2）读数据

读数据，也就是接收数据。接收数据的流程图如图4.7，接收数据的程序框图如图4.8。默认设置串口等待接收数据，判断是否接收到数据，如果没有接收到数据跳回到接收数据，如果已经接收到数据，将接收到的数据作为数据解析子模块的输入执行数据解析子模块，去除分隔符并提取接收到的数据中的可用信息，最后将可用信息显示在显示控件的面板上。

接下来介绍数据解析模块的实现流程。



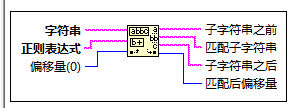
**图4.8 数接收据流程图**

**图4.9 数接收据程序框图**

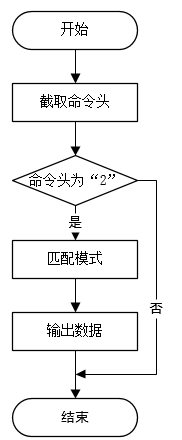
（3）数据解析子模块

上位机与下位机的通信命令码有两条，例如命令1：1$；命令2：2$X200,Y300,Z400,F800。数据解析子模块就是解析这两个命令，并从这些命令中提取有用信息，这些有用信息，比如说在命令1中是命令头“1”，在命令2中是“200”，“300”，“400”和“800”。在下面将详细介绍这个数据解析流程。

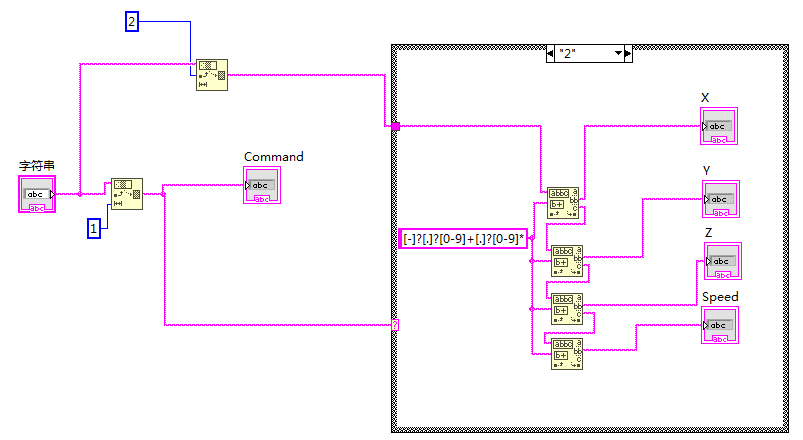
数据解析子模块的流程图如图4.11。在下面的程序框图中（图4.12）使用了匹配模式函数，函数的原型如图4.10，该函数的一个输入项“字符串”就是需要解析的源字符串，另一个输入项“正则表达式”就是解析源字符串的规则，这个输入项也是个字符串。还有一个输入项“偏移量”表示源字符串开始解析的位置，默认在第一个位置，在这里使用默认不作更改，不接线。匹配模式函数的输出项有“子字符串之前”，“匹配子字符串”，“子字符串之后”和“匹配后的偏移量”，假如源字符串是“X200,Y300,Z300,F100”，偏移量为0，正则表达式的规则是匹配字母后面的数字，那么“子字符串之前”是没有的，因为偏移量是0，匹配到的字符“200”就是有用信息，子字符串是“,Y300,Z300,F100”,匹配后的偏移量就是4，如果需要继续提取继“匹配子字符串”之后的字符串中的有用信息，可以将“子字符串之后”作为源字符串输入至匹配模式函数，然后继续上述操作即可。



**图4.10 匹配模式函数**



**图4.11数据解析子模块流程图**

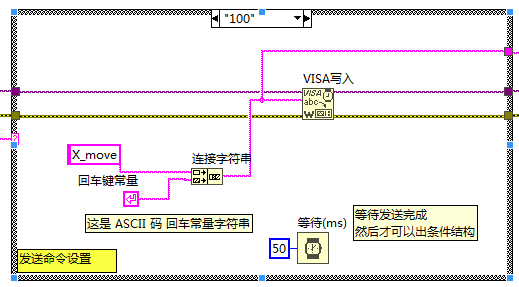


**图4.12 数据解析子模块程序框图**

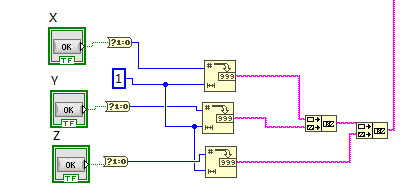
### 5.2.2写数据模块

（1）快速移动子模块

如图4.12，向下位机发送快速移动的命令的操作被圈在一个条件结构里面，这个条件结构的触发条件需要X、Y、Z按钮来触发，然而按钮只能触发布尔值的变化，而条件结构中的条件只能是字符串，这就要设计触发条件。如图4.13是触发条件的设计程序框图。在4.13的程序框图中，首先使用“布尔转数字”函数将布尔值转化成数字，然后使用“数字转字符串”函数，将数字转化成字符串，最后使用“拼接字符串函数”按顺序把转化后的字符串拼接成一个字符串，这个字符串就可以作为图4.12中的条件结构之条件。

 图4.12的条件结构中，就是当X轴快速移动的按钮被按下的情况，在这种情况下向下位机发送“X\_move”命令，如果是Y轴快速移动的按钮被按下则发送“Y\_move”，否则就发送“Z\_move”命令，其他操作在其他条件分支中式一样的。

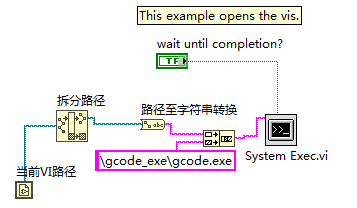
**图4.12 快速移动子模块程序框图**



**图4.13 设计触发条件**

（2）调用G代码解析算法子模块

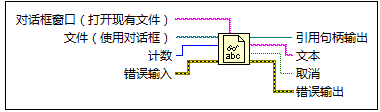
G代码解析算法的执行文件是gcode.exe文件。下下面的程序框图（图4.14）中，首先寻找到当前程序框图文件的路径，然后将这个路径末端，也就是该程序框图的文件名拆分下来，然后将拆分后的路径转化成字符串，使用“拼接字符串函数”将gcode.exe文件的相对路径拼接，这就找到了相对于当前vi路径的相对路径上的gcode.exe的路径。最后将gcode.exe作为输入，调用一个“系统执行”函数，该函数就可以执行gcode.exe文件，生成运动轨迹坐标文件。



**图4.14 调用G代码解析算法子模块程序框图**

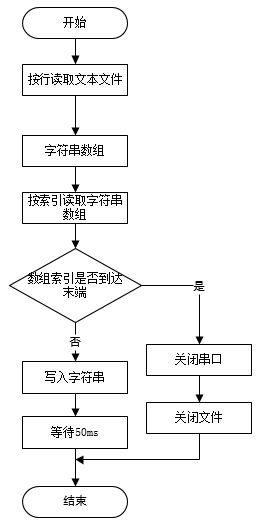
（3）**按行读取文本文件**子模块

这里主要是使用了一个“读文本文件”函数，函数原型如图4.15。这个函数的功能是从字节流文件中读取指定数目的字符或行。在调用该函数右击可以选择按行读取文件，如果还“计数”项设置为-1就可以按行读取整个文件，用这种方式来读文件，其输出将是一个元素为各行的文本内容的字符串数组。

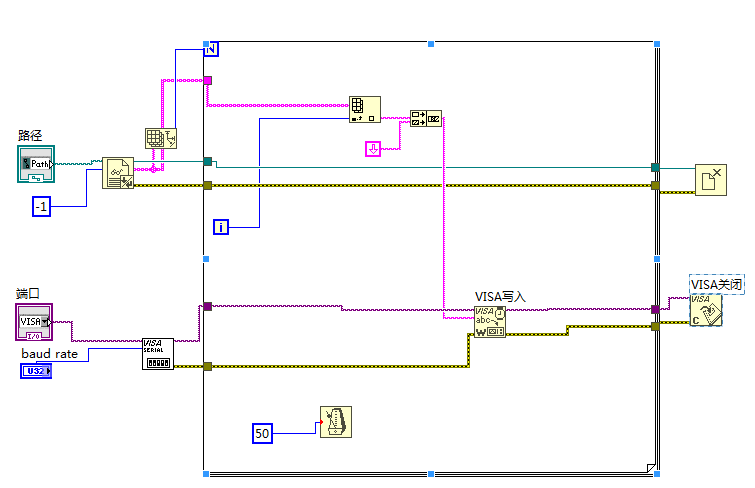


**图4.15 读文本文件函数**

如图4.16是按行读取文本文件流程图。正如上述，使用按行读取模式的“读文本文件”函数得到输出是一个字符串数组。所以，如图4.17程序框图使用for循环来按索引字符串数组来发送一个个字符串。发送每个字符串的时候都要使用延时，等待发送完成。如果字符串索引到达数组末端就关闭串口和关闭文件。



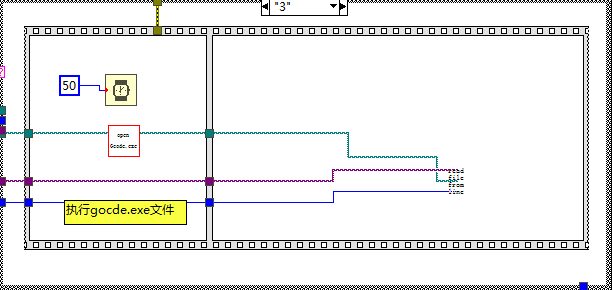
**图4.16 按行读取文本文件流程图**



**图4.17 按行读取文本文件程序框图**

（4）按行发送轨迹坐标

如图4.18是按行发送轨迹坐标子模块程序框图，从图中可以看出使用了一个顺序结构，在顺序结构的第一个分支调用了子模块“调用G代码解析算法子模块”，执行gcode.exe生成轨迹坐标文件；在顺序结构的第二个分支中调用了子模块“按行读取文本文件”，按行发送轨迹坐标。

至此，界面的服务程序中的读模块与写模块的编写工作到此为止。

**图4.18 按行发送轨迹坐标子模块程序框图**

# 6．仿真测试

## 6.1 G代码解析算法测试

（1）预设测试数据

（2）测试结果

## 6.2上位机仿真测试