实验指导书

配合三自由度气动机械手臂使用

V2.0

亿学通电子编著

北京纳克斯科技发展有限公司技术资料

http://www.61mcu.com

前言

“气压传动与控制”是一门工业、尤其是轻工业上应用非常普遍的技术。它以空气作为传动介质，将气体的压力能（势能）转化为机械能的一种传动方式，比液压传动成本低、清洁环保，比机械传动控制灵活、调节方便，并可与电气控制结合，形成便捷高效的电气混合控制。

三自由度气动机械手，作为一款高性价比、可重复拆装的气压传动控制类实验设备，创新性的采用了模块化的设计理念，从机械结构到电气控制部分，均为独立单元，方便了认知教学、机械拆装、编程实训、功能扩展等环节。在知识点设置上，涵盖了气缸/电磁阀控制、单片机或 PLC 编程、步进电机驱动、机械绘图与装配等多门类课程。不但适用于课堂教学实验，更适合用于课程设计、毕业设计等创新实践环节。以帮助同学们加深印象，提高实际动手能力,为融会贯通书本理论知识，分析解决工程实际问题奠定一个良好的基础。

为了方便老师和同学更好地学习和使用这款产品，亿学通电子基于本产品设计了十个实验。这些实验基本上涉及到了气动机械手的全部功能模块。

本书主要有三部分内容：

第一章单片机基础实验。

这章主要帮助同学们复习或学习掌握单片机的编程、编译、下载方法。学习如何通过单片机编程让机械手完成既定的功能。

第二章机械测绘相关实验。

这章主要是学习气动元件、控制器件、电机驱动等相关机电模块功能和原理。通过这些实验，可以帮助同学掌握气动机械手的硬件结构、机械装配、工作原理。以更好的学习掌握机电一体化产品的设计方法。

第三章综合扩展实验。

这章主要是针对综合应用而开展的实验，内容有，机械手完成搬运物体实验、机械手仿真实验、PLC 控制器扩展实验等等。设计这些实验的目的是让同学能将基础模块进行功能整合，可以完成一个较为系统的功能展示。

全书的实验由浅入深，由部分到整体，涉及了从基本结构到模块再到应用的全部实验，老师或者同学按照实验顺序从基本的编程到复杂的综合应用，一方面可以很好的学习和掌握气动机械手的基本知识，另一方面可以很容易的应用机械手进行设计，为以后的课程设计、毕业设计或者工作中的开发设计奠定了良好的基础。

在编写过程中难免会有错误纰漏，请广大读者朋友批评指正。

亿学通™电子

2010.10.10

实验注意事项：

气动三自由度教学机械手是一种综合性创新及实验设备，实验前务必仔细阅读实验指导书，了解实验内容，熟悉实验原理、接线方法及实验步骤，并根据实验要求独立完成必要的分析。实验连接电线和气管前必须先断开总电源及各分电源开关，严禁带电带压操作，电线和气管连接完毕，请实验指导老师检查确认，方可进行后续实验。

一、实验准备工作

1. 确保气路、电路的连接正确、可靠；

2. 确保机械手各模块之间连接正确；

3. 确保计算机、PLC、控制主板之间线路正确连接；

4. 准备好本次实验所使用的工具等。

二、实验过程

1. 每次实验，请指定一名受过使用培训的同学来启动气泵；

2. 在使用机械手的实验中，须先确保气泵气压是否设置在规定值（0.4-0.8MPa），然后开启气泵电源；

3. 确保操作人员不在机械手的工作空间内，以免被突然伸出的机械件碰伤；

4. 调节气缸上的气体调节阀，使气缸的动作处于理想速度；

5. 程序下载和运行实验时应注意转换连接线；

6. 程序编写或程序下载时，将控制板和电磁阀的接线脱离，当单片机程序运行的时候，再连接；

7. 控制板上面的接线需要联接或拔除、或控制板需要拆卸组装时，切记带电操作，以防造成电路短路、控制板接口损坏、单片机程序丢失等情况；

三、实验结束

1. 当实验结束时，请先将本设备的电源关闭，等全班同学都完成实验后，再将气泵关闭；

2. 整理实验台，相关工具等交实验老师存放；

3. 填写设备使用记录，若设备有故障，须报告老师以便检修。

亿学通™电子

2010．09．10

目录

第1章 单片机基础实验

实验一　单片机程序下载实验

【实验目的】

1. 学习并掌握 STC 单片机程序下载方法；
2. 通过实际操作，整理了解单片机在机电一体化中的应用。

【涉及知识面】

该实验涉及《单片机原理与应用》课程有关知识。

【实验工具】

计算机一台、三自由度气动机械手一台、空气压缩机一台（实验室共用）

【实验要求】

1. 会使用 STC-ISP 下载软件。
2. 下载程序后，使用手柄控制机械手运动。

【实验原理】

1. 单片机程序下载

随着科技发展单片机在我们生活中的应用越来越广泛，通信工程、工业产品、仪器仪表、汽车电子、安防网络、数字影像、消费娱乐。掌握单片机的开发也成为电子相关专业同学必备的技能。

什么是单片机呢？最简单的讲，单片机就是一种由数量众多晶体管组成的、能进行数学和逻辑运算的集成电路。就是把中央处理器 CPU（Central Processing Unit）、存储器（Memory）、定时器、I/0（Input/Output）接口电路等一些计算机的主要功能部件集成在一块集成电路芯片上的微型计算机。

怎么让单片机实现控制电磁阀、步进电机或其他功能呢？这需要单片机编程。C 语言课程就是完成这个工作的一部分。我们把单片机如何完成一个具体任务的程序放到单片机内部的程序存储器（ROM）中的过程，就是单片机程序下载。

单片机编程一般分为：程序编辑（使用 C/C++、汇编或操作系统编写的程序源代码）、程序编译（将用户编写的中高级语言生成单片机能识别的，如*.hex、*.bin 等格式的目标代码）、程序下载（将目标代码写入单片机程序存储器 ROM 中）。

程序的编辑和编译，将在单片机编程实验中介绍。今天只介绍我们如何将已经编译好的*.hex 目标代码写入单片机程序存储器。

2. 单片机程序下载工具和类型

可以完成单片机程序下载的工具，我们称之为下载器/编程器（也有称为烧写器、烧录器的）。

不同时期、不同型号的单片机，使用的下载器也不一样。

较早时期，程序下载（或烧写）都必须专门的程序编程器来完成。像最早的单片机教学中用到的 AT89CXX 系列单片机，编程电压需要 12V 以上，这个就必须要专门的编程器来完成单片机程序的下载，然后再将单片机安装到目标系统板上。现在很多 OTP（One Time Programmable）单片机还是用这种下载方式。

随着技术发展，尤其是程序存储器技术的发展，很多单片机都支持 ISP（In-system programmable）在系统可编程方式。就是不需要把单片机从目标系统板上取下来就可以直接从 PC 往单片机里面烧录程序。所以即使我们将芯片焊接在电路板上，只要留出和上位机接口的这个串口，就可以实现芯片内部存储器的改写，而无须再取下芯片。免去了调试时由于频繁地插入取出芯片对芯片和电路板带来的不便。

ISP 技术的优势是不需要编程器，是用简单的下载线就可以实现程序下载。

现在我们使用的 STC89C52 单片机就支持 ISP 程序下载。只需要一条串口线就能完成程序的下载，非常方便。

另外程序下载还有 JTAG、IAP 等方式，有兴趣的同学可以查阅相关资料。

【硬件连接】

首先我们为单片机主板接通电源、连接主板与计算机、连接手柄控制器，如图 1-1；

其次根据主板上方的提示，将按键 K3 按下置于“ON”、按键 K4 弹起置于“OFF”；

备注：为了方便用户程序下载，单片机主板留出了一路串口、一路 USB 口和计算机进行连接，用户可以自行选择一种接口完成程序下载。

（USB 口的设置，主要方便没有串口的计算机用户；如果使用 USB 口进行程序下载，用户需要安装“USB-串口驱动程序”）

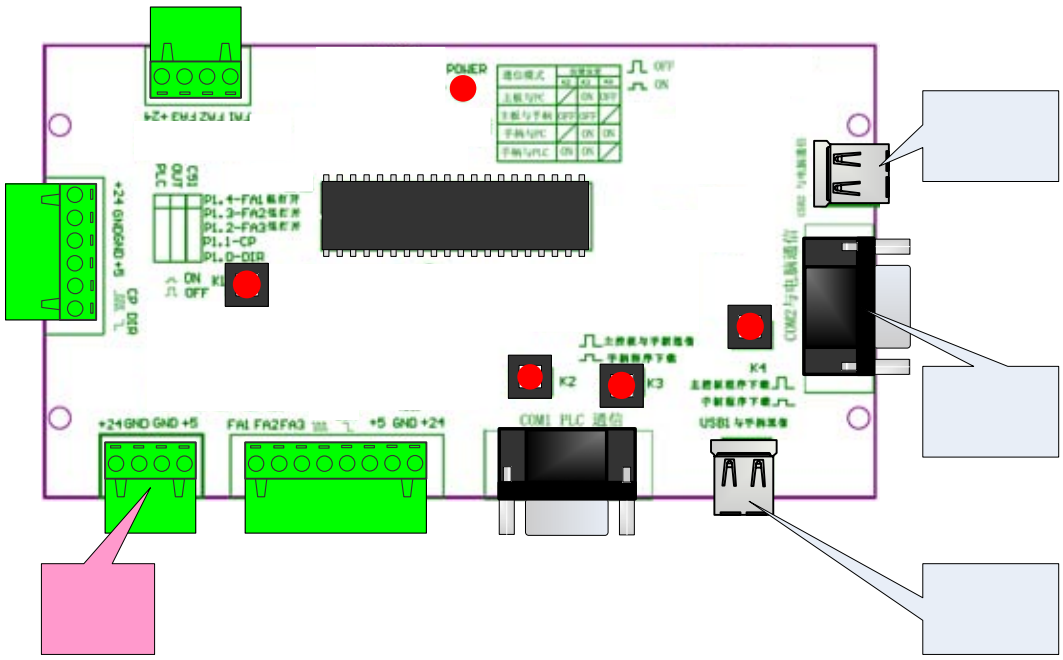


图 1-1 单片机连接图

【实验步骤】

1. 根据产品说明书，熟悉单片机控制板结构和按键设置；

2. 连接计算机和单片机控制板之间的下载线；

3. 检查没有问题，接通电源；

4. 在计算机上打开 STC-ISP 编程软件以及机械手演示程序目标代码：*.hex

我们在“三自由度气动机械手”资料光盘，或实验室电脑中找到 STC-ISP 软件。双击打开。（本软件是非安装版，解压后直接执行 STC-ISP.exe 即可）。图形界面如图 1-2 所示。

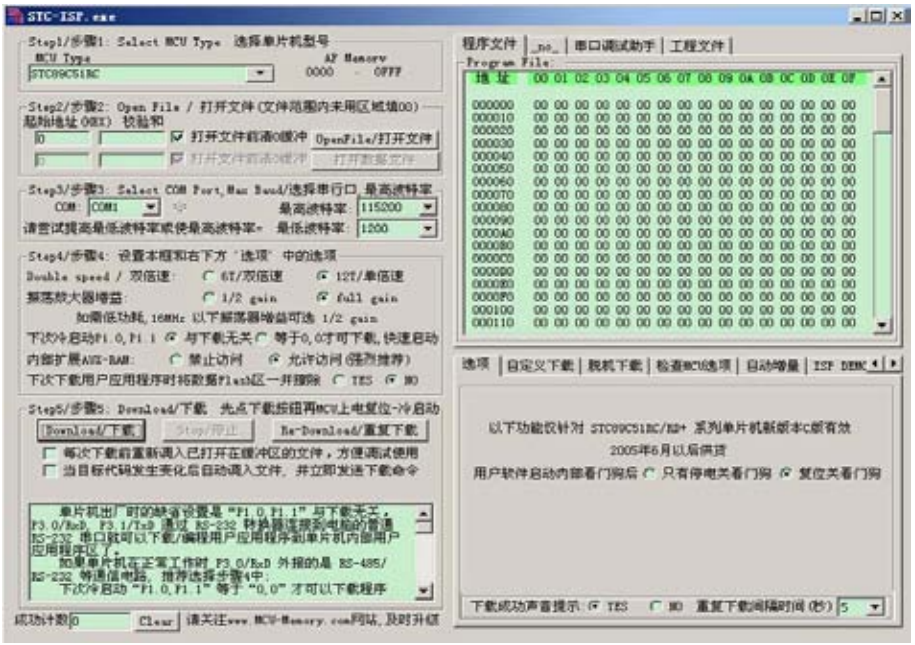


图 1-1　STC-ISP 编程软件界面

注：STC_ISP 是宏晶科技公司提供的使用串口直接将用户程序目标代码（.hex 文件）下载进 STC 单片机的编程软件。

5. 按照下面介绍的方法将程序下载到单片机；

图 1-3　在 MCU-Type 下拉菜单中选择单片机型号（STC89C52RC）

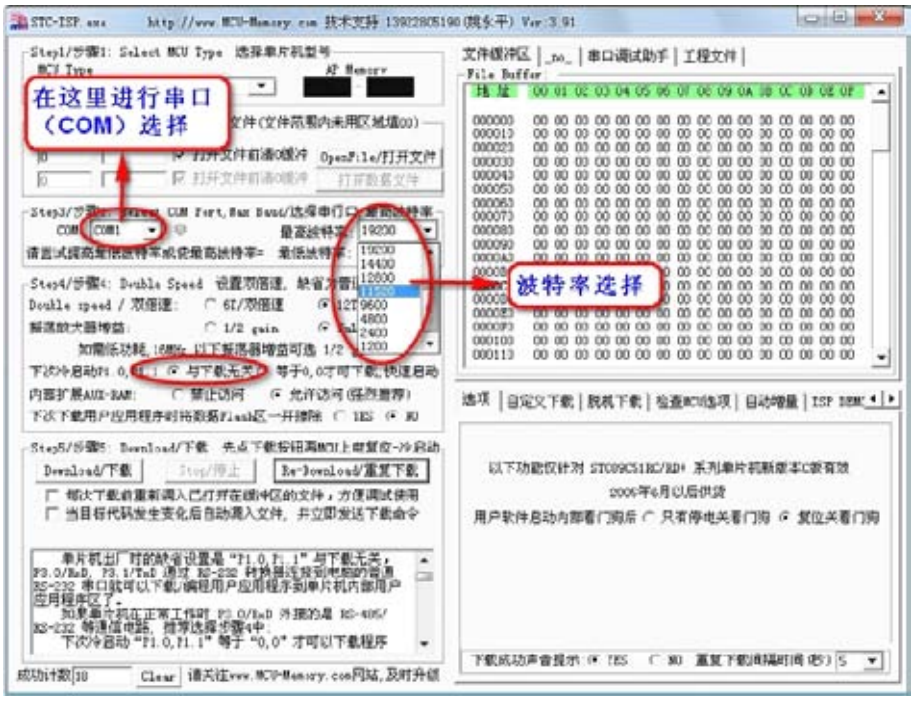


图 1-4　设置下载所使用的 COM 口和波特率

如果我们下载线连接正确，当我们打开 STC-ISP 下载软件后，软件将自动选择已经 COM 口。我们也可以自己手动设置。设置方法如下：

- 如果我们使用串口线完成程序下载，那么用到的 COM 口一般是 COM1 或 COM2；电机下拉菜单，选择即可；
- 如果使用 USB 口来完成程序下载，则需要到电脑设备管理器中查看虚拟后的 COM 口数值。

在“我的电脑\右键‘属性’\硬件\设备管理器”选项卡中，在“端口（COM 和 LPT）”中可以看到一个 USB-to-Serial 虚拟的 COM*口，这个 COM*口即是 STC-ISP 程序下载使用。如图 1-5 所示：

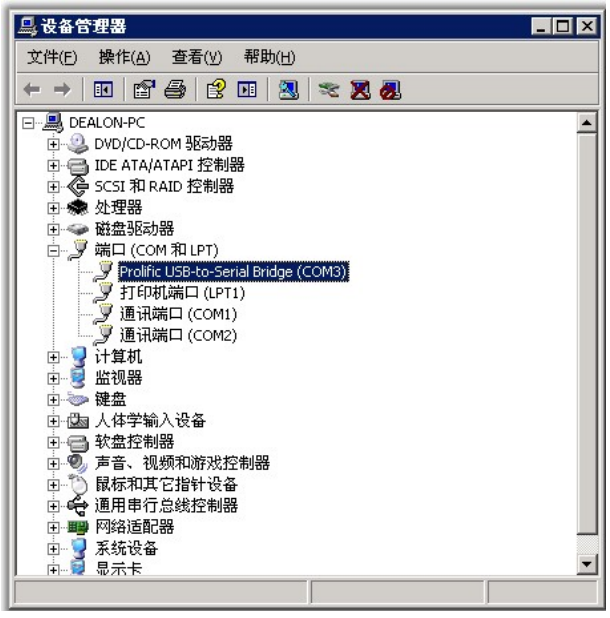


图 1-5　在设备管理器中查看串口

如果在“设备管理器”/“端口（COM 和 LPT）”中看不到图 3-5 中的信息，请检查单片机控制板是否上电、USB-串口驱动是否安装。

波特率表示的是计算机下载程序到单片机速度的快慢。较低的波特率在数据传输时有较高的稳定性，建议波特率不要设置太高。

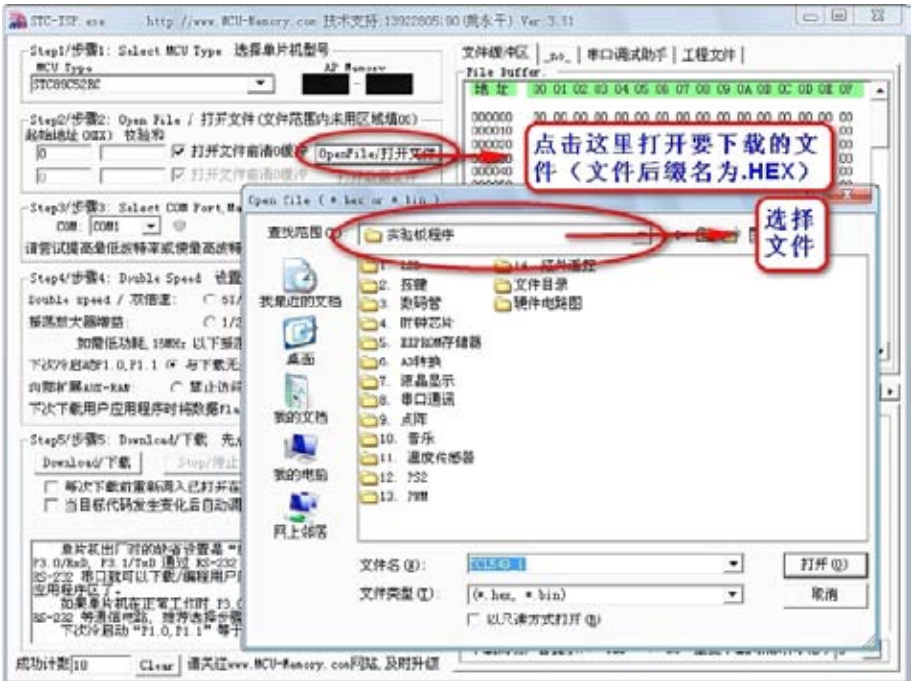


图 1-6　在计算机中找到机械手演示程序的目标代码：EX1_Borad_test.HEX

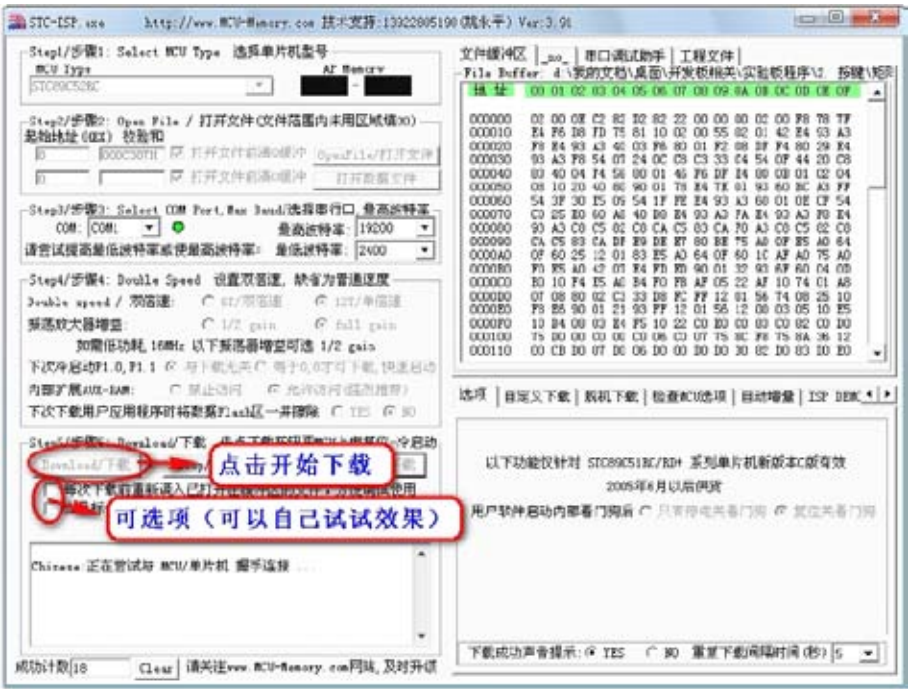


图 1-7　点击“Download/下载”按钮，按动电源开关 K1 键给单片机供电

我们在使用 STC-SIP 下载程序时，选择的是冷启动方式，即先点击程序下载按钮，再给单片机上电复位。所以下载程序前，需要将单片机断电，点击下载按钮后，再给单片机供电。

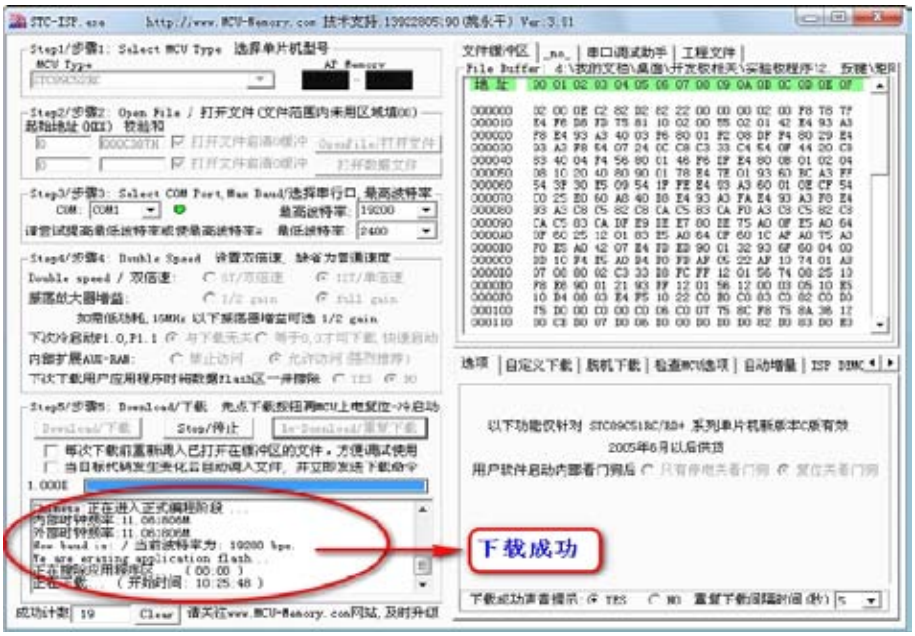


图 1-8　程序下载成功图片

备注：如果程序不能下载，请从以下几个方面找一下原因：

- COM 是否正确，如果使用串口线下载，可以更换 COM1 或 COM2 口尝试；如果使用 USB 下载线，请按照上面的方法查看实际的 COM 口数值，或者安装 USB 驱动；
- 如果 COM 设置正确，请查看，下载顺序是否正确：先点击下载，然后给单片机供电；
- 尝试降低波特率。

6.　关掉机械手总电源电源，连接电机驱动器、电磁阀接线，如图 1-9 所示位置；

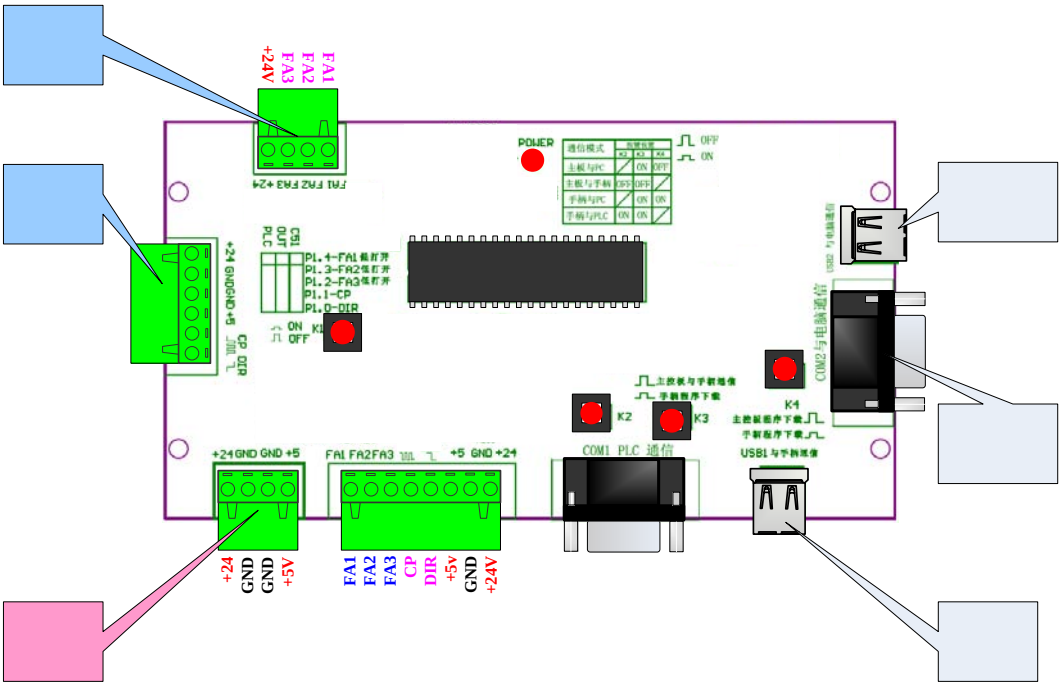


图 1-9　电磁阀和步进电机驱动接线

7.　设置按键 K2、K3 弹起，置于到“OFF”位置；接通手柄和单片机。

8. 检查所有电路和气动部件，确认没有问题，接通气源，打开电源开关；这个过程中，操作员要和机械手臂保持一米距离，防止接通气源时，机械手运动，伤到同学。
9. 使用手柄对机械手进行控制。

【范例路径】

三自由度机械手实验例程\Example\EX11_Download

【作业】

1. 根据单片机程序下载方法，为手柄下载程序。（EX1_Handle_test.HEX）
2. 查询不同型号单片机的程序下载方式，增加对单片机程序下载的了解。

实验二　单片机编程实验

【实验目的】

1. 学习并掌握单片机编程方法；
2. 了解单片机编程原理；

【涉及知识面】

该实验涉及《单片机原理与应用》课程有关知识。

【实验工具】

计算机一台、三自由度气动机械手一台

【实验要求】

1. 会使用 Keil C51 编程软件。
2. 编写简单的程序：hello.C。

【实验原理】

1. IDE 集成开发软件

在实验一中，我们介绍了单片机编程的过程，一般分为：

程序编辑（使用 C/C++、汇编或操作系统编写的程序源代码）；

程序编译（将用户编写的中高级语言生成单片机能识别的，如*.hex、*.bin 等格式的目标代码）；

程序下载（将目标代码写入单片机程序存储器 ROM 中）。

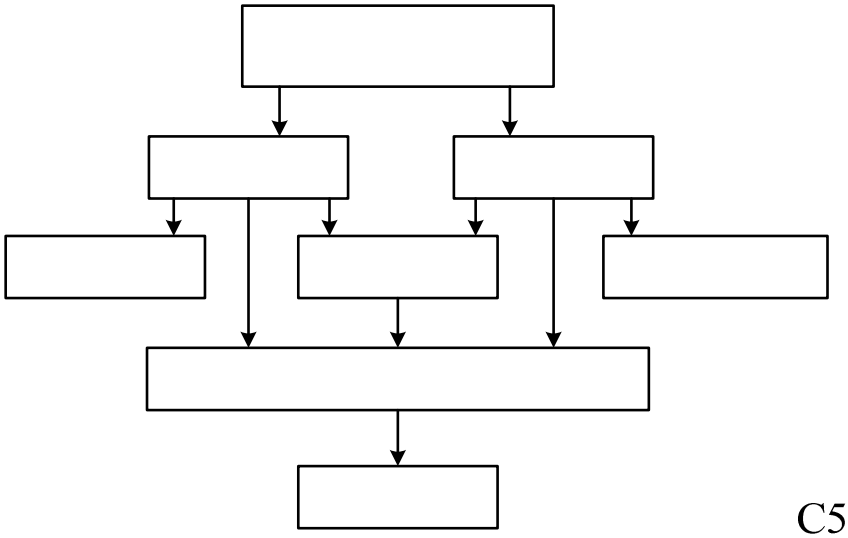
从单片机的学习中，我们知道，单片机是一个数字器件，只能识别 1、0 数据，我们使用 C/C++、汇编语言等编写出来的程序，单片机是不能识别的。而单片机能识别的数据格式，开发者写起来又非常吃力、可读性不强，这就促使了 IDE(Integrated Development，集成开发环境)软件的出现。

IDE 集成开发环境（简称 IDE）软件是用于程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面工具。该程序可以独立运行，也可以和其它程序并用。这极大方便了单片机开发。

常见的 IDE 有：51 单片机：Keil C 51、WAVE；凌阳单片机：unsp IDE2.0.0，Fortis IDE ；合泰单片机 HOLTEK：HT-IDE3000；PIC 单片机：MPLAB C18,PICC8.X 等。

2. KEIL C51 集成开发环境软件

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统（也可用于汇编语言编程）。 软件提供了丰富的库函数和功能强大的集成开发调试工具，全 Windows 界面。C 语言生成的目标代码效率非常高。在开发大型软件时更能体现高级语言的优势。



uVision/Ishell

C51编译器

图 2-1 Keil C 开发软件整体架构

Keil C51 开发软件，功能强大，整体架构如图 2-1 所示。

其中 uVision 与 Ishell 分别是 C51 for Windows 和 for Dos 的集成开发环境(IDE)，可以完成编辑、编译、连接、调试、仿真等整个开发流程。开发人员可用 IDE 本身或其它编辑器编辑 C 或汇编源文件，然后分别由 C51 及 A51 编译器编译生成目标文件(.OBJ)。

C库文件

LIB51

目标文件可由 LIB51 创建生成库文件，也可以与库文件一起经 L51 连接定位生成绝对目标文件(.ABS)。ABS 文件由 OH51 转换成标准的 Hex 文件，以供调试器 dScope51 或 tScope51 使用进行源代码级调试，也可由仿真器使用直接对目标板进行调试，也可以直接写入程序存储器中。

BL5

- C51 是 C 语言编译器，对 C 源文件(.C)进行编译；
- A51 是汇编语言编译器，对汇编源文件(.asm 或.a51)进行编译；
- L51 是 Keil C51 软件包提供的连接/定位器，其功能是将编译生成的 OBJ 文件与库文件连接定位生成绝对目标文件(.ABS)；

OH51

3. 单片机编程流程

单片机编程是用户使用中高级语言对单片机底层硬件进行操作的过程。所以我们在编程的时候，首先要对单片机硬件资源进行分配，尤其是单片机 IO 口的分配。例如在三自由度机械手主板的单片机在和步进电机驱动、电磁阀连接的时候，就用到了 P1.0—P1.4 五个 IO 口。我们只要在编程的时候，对这个五个 IO 口进行控制，就能很好的对步进电机和电磁阀进行控制。

其次要根据编写的程序，绘制程序流程图。

程序流程图是人们对解决问题的方法、思路或算法的一种描述。在编写程序时，尤其是编写功能较为复杂的程序时，绘制程序流程图可以快速的帮我们确定程序结构、算法和思路，是一个非常有效的工具。

例如，如果我们要编写一个这样的程序：让机械手，抓取一个工件，从 A 点旋转 90° 后搬运到 B 点，然后放下。这个程序就相对较为复杂，我们可以先绘制程序流程图，如图 2-2 所示：

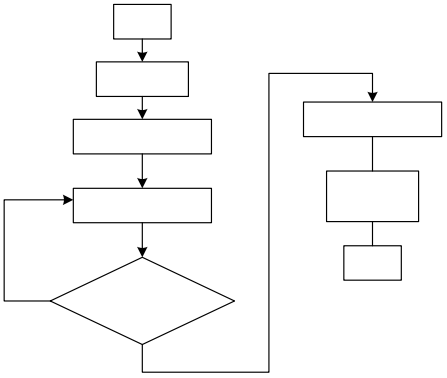


图 2-2 机械手搬运物体程序流程图

第三步我们就要根据分配好的硬件资源和程序流程图，编写程序。

这时候我们可以借助 Keil C51 的程序编辑平台，使用 C 或汇编语言，按照固有的语法格式，编写程序。

第四步使用 Keil C51 软件对编写程序进行编译，生成机器码后，对程序实现功能极性测试，如果不能满足要求，就修改程序，直到满足要求为止。

最后一步，就是将编译好的程序代码下载到单片机里面，完成编程。

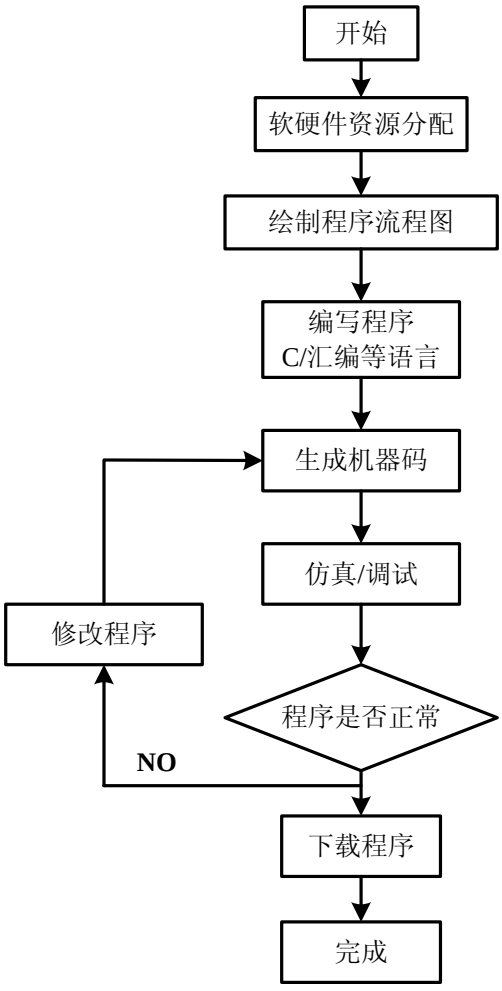


图 2-3 程序编程流程图

【硬件连接】

将单片机主板通过 USB2 口或 COM2 串口连接到计算机；

将按键 K3 按下置于“ON”、按键 K4 弹起置于“OFF”，完成主板和计算机的连接；

为单片机主板接通电源。

（由于本试验程序不对步进电机、气缸和手柄进行操作，所以其他接口不用连接；气泵也不需要工作）

【实验步骤】

双击打开 Keil C51 软件：

图 2-4 启动 Keil C51 时的软件界面

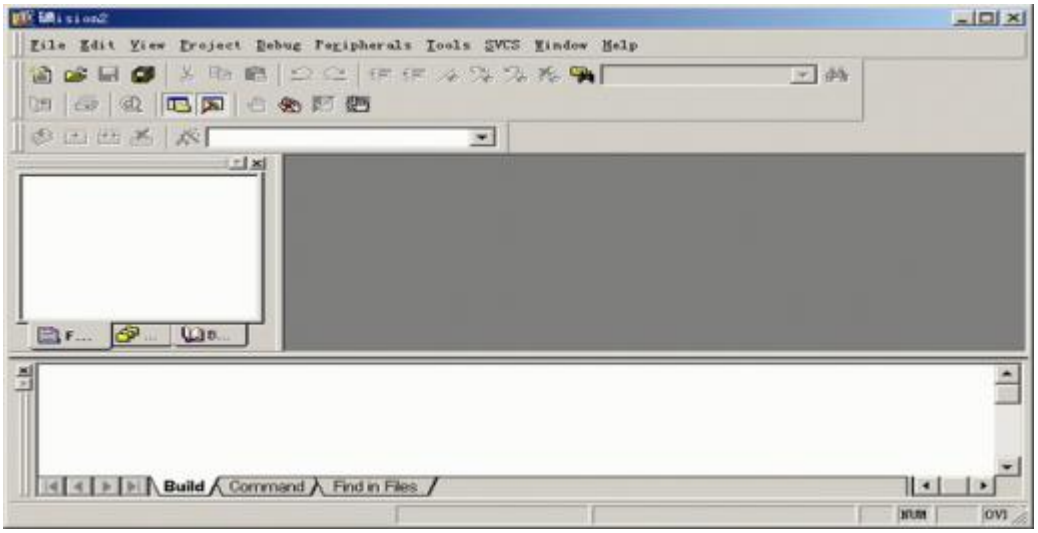


图 2-5 Keil C51 后的编辑界面

1. 建立一个新工程　单击 Project 菜单，在弹出的下拉菜单中选中 New Project 选项

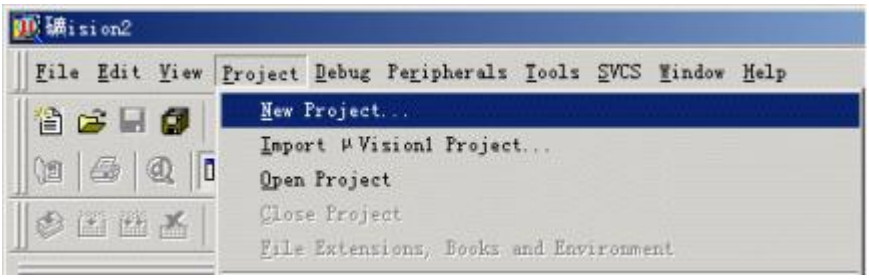


图 2-6　新建工程

2. 然后选择你要保存的路径,输入工程文件的名字,比如保存到 C51 目录里,工程文件的名字为 C51 如下图所示,然后点击保存

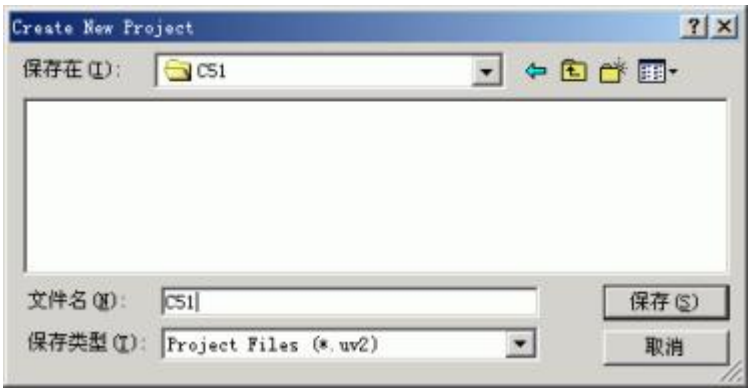


图 2-7　保存工程文件

3. 选择单片机型号

这时会弹出一个对话框,要求你选择单片机的型号,你可以根据你使用的单片机来选择,keil c51 几乎支持所有的 51 核的单片机。STC89C52 单片机是国产单片机，Keil C51 里面没有收录 ,我们可以选择 AT89C52，二者结构和功能完全一样。

如下图所示,选择 89C52 之后,右边栏是对这个单片机的基本的说明,然后点击确定.

图 2-8　选择单片机型号

4. 完成上一步骤后，屏幕如下图所示

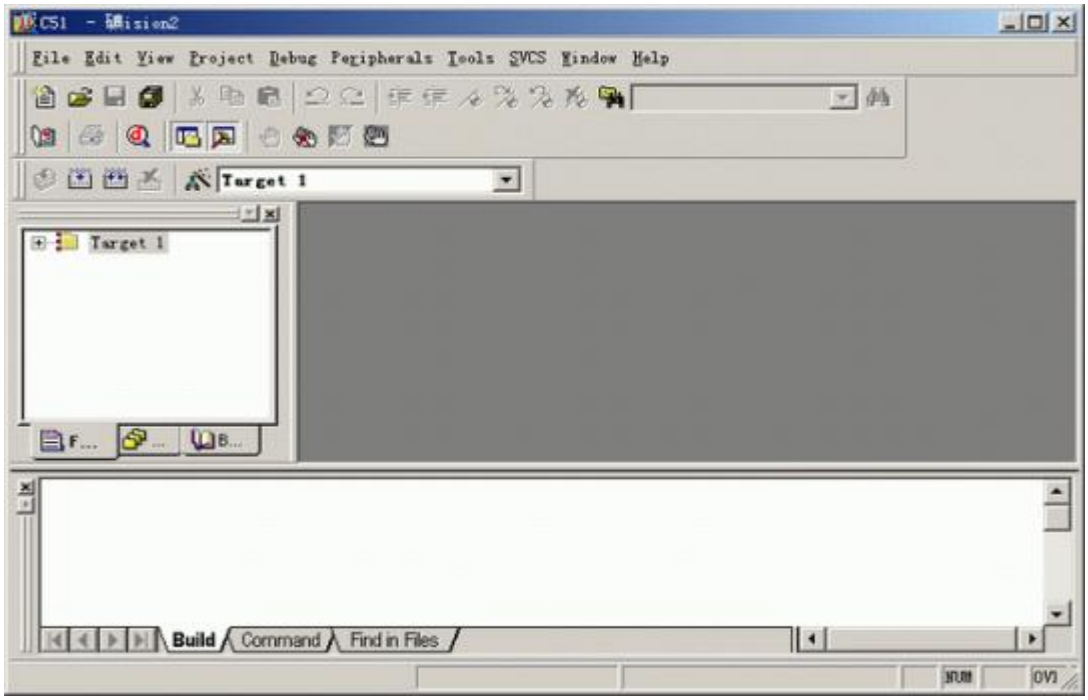


图 2-9　建立新工程界面

到现在为止，我们就建好了单片机工程文件。接下来就在这个工程下添加 C 或汇编程序。

5. 在下图中，单击“File”菜单，再在下拉菜单中单击“New”选项

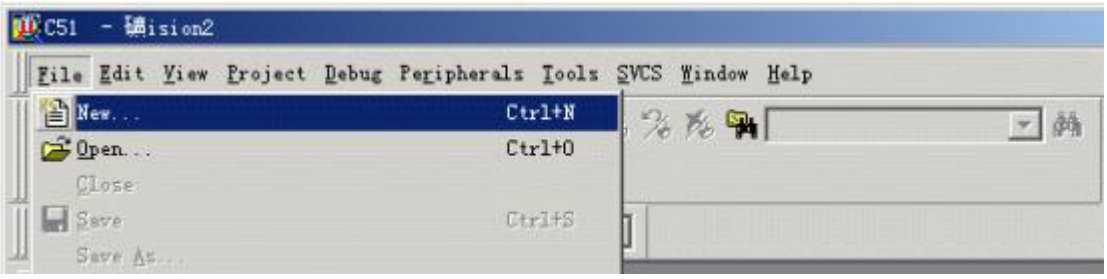


图 2-10　在工程下新建一个文件

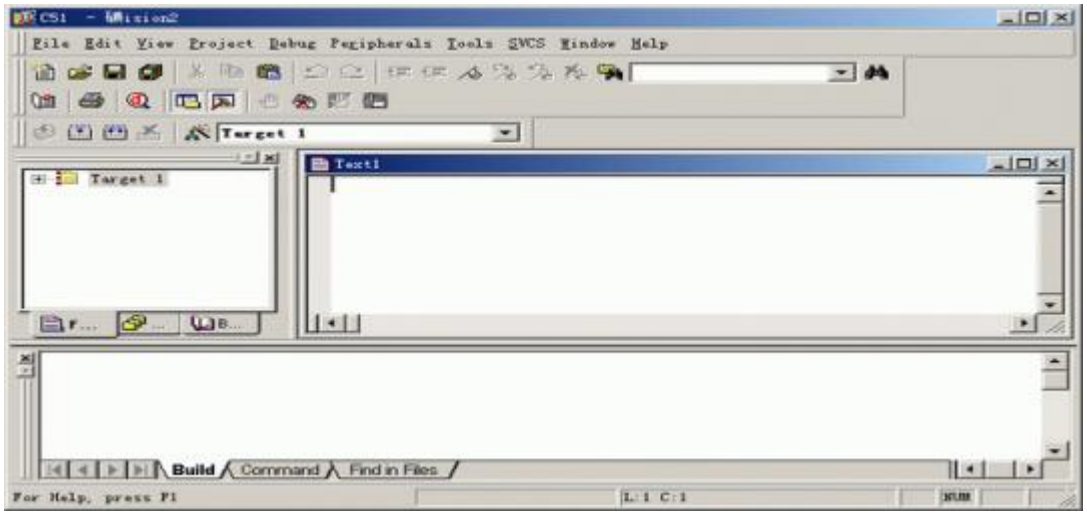


图 2-11　建立一个程序编辑窗口

此时光标在编辑窗口里闪烁，这时就可以键入用户的应用程序了。但我们最好先保存一下建立的新文件：单击菜单上的“File”，在下拉菜单中选中“Save As”选项单击，屏幕如下图所示。在“文件名”栏右侧的编辑框中，键入欲使用的文件名，同时，必须键入正确的扩展名。注意，如果用 C 语言编写程序，则扩展名为(.c)；如果用汇编语言编写程序，则扩展名必须为(.asm)。然后，单击“保存”按钮。

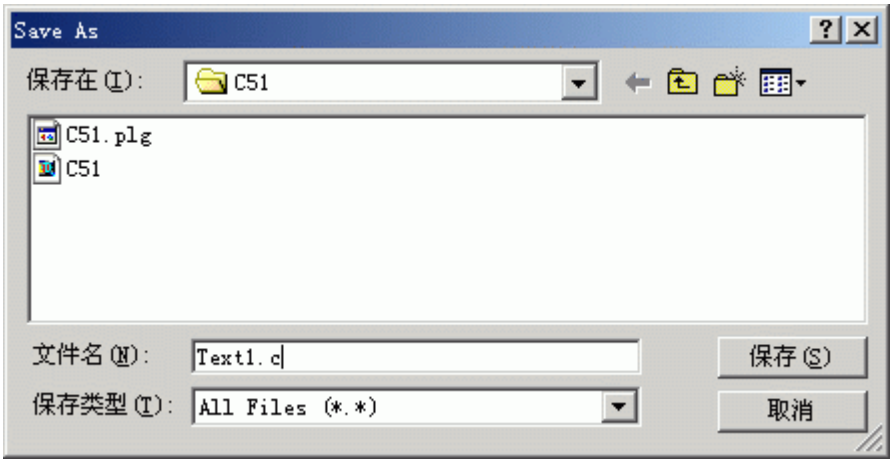


图 2-12　保存编写的程序文件

6. 回到编辑界面后，单击“Target 1”前面的“＋”号，然后在“Source Group 1”上单击右键，弹出如下菜单：

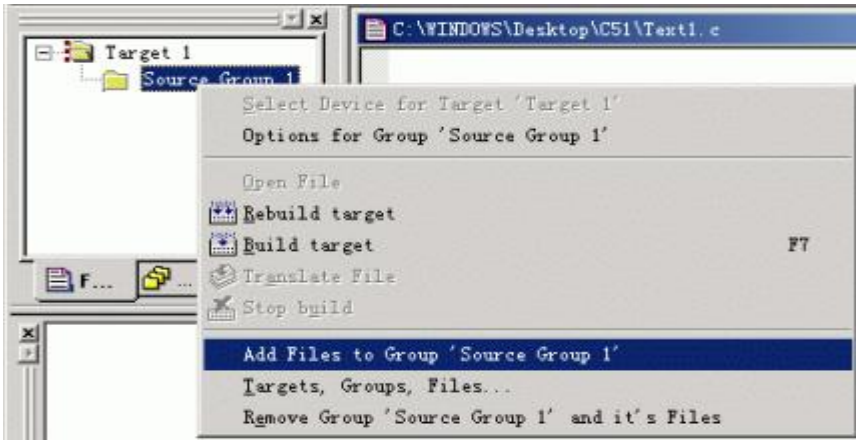


图 2-13　添加新编辑的 C 语言文件到工程文件中

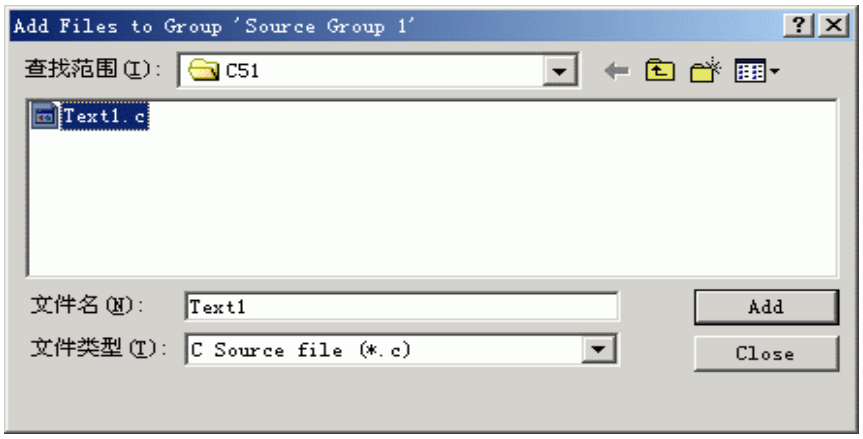


图 2-14　找到新建 C 语言文件路径，点击 ADD

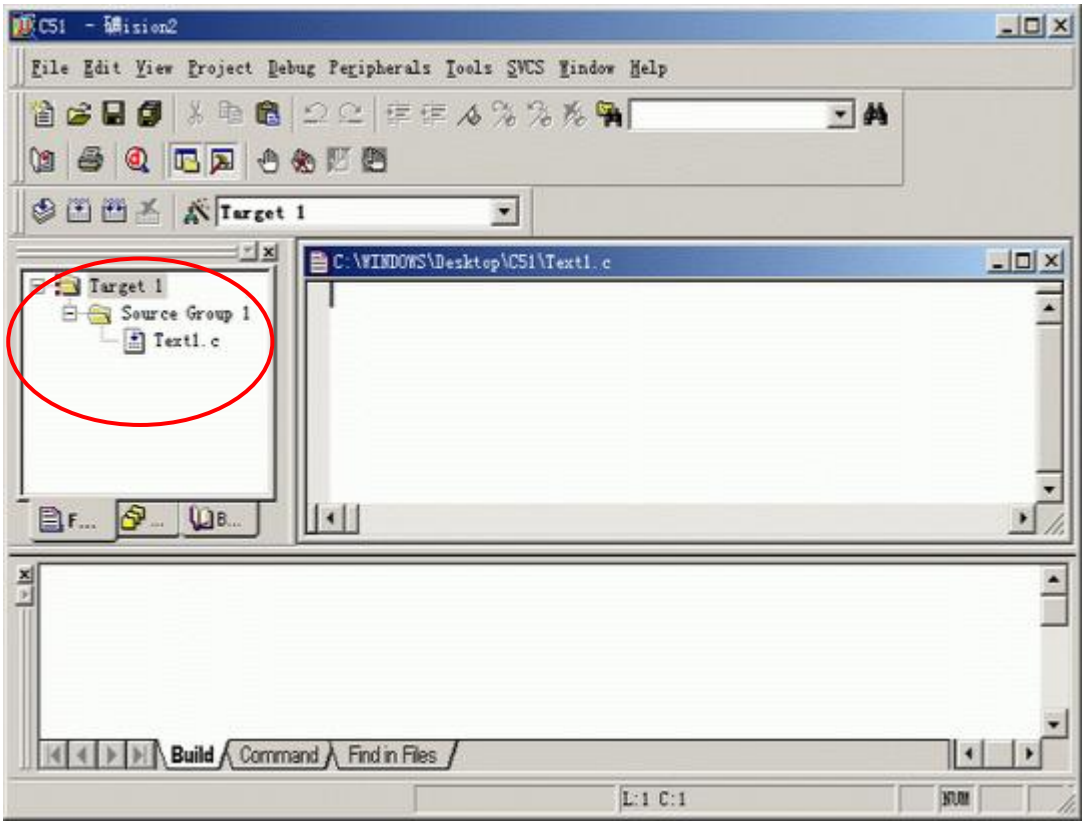


图 2-15　在工程文件 Target 1 下面就出现了新添加的文件：Text 1.c

备注：在工程文件下，我们可以添加多个文件 C、汇编文件、头函数等等。

7. 现在，请输入如下的 C 语言源程序:

```
#include <reg52.h>                    //包含文件
#include <stdio.h>
void main(void)                       //主函数
    {
        SCON=0x52;
        TMOD=0x20
        TH1=0xf3;
        TR1=1;                        //此行及以上 3 行为 PRINTF 函数所必须
        printf("Hello I am KEIL. \n");    //打印程序执行的信息
        printf("I will be your friend.\n");
        while(1);
    }
```

由于我们在建立 Text 1.c 后，立刻选择了保存，所以在我们输入上述程序时，Keil c51 软件会自动识别关键字，并以不同的颜色提示用户加以注意，这样会使用户少犯错误，有利于提高编程效率

程序输入完毕后，如下图所示

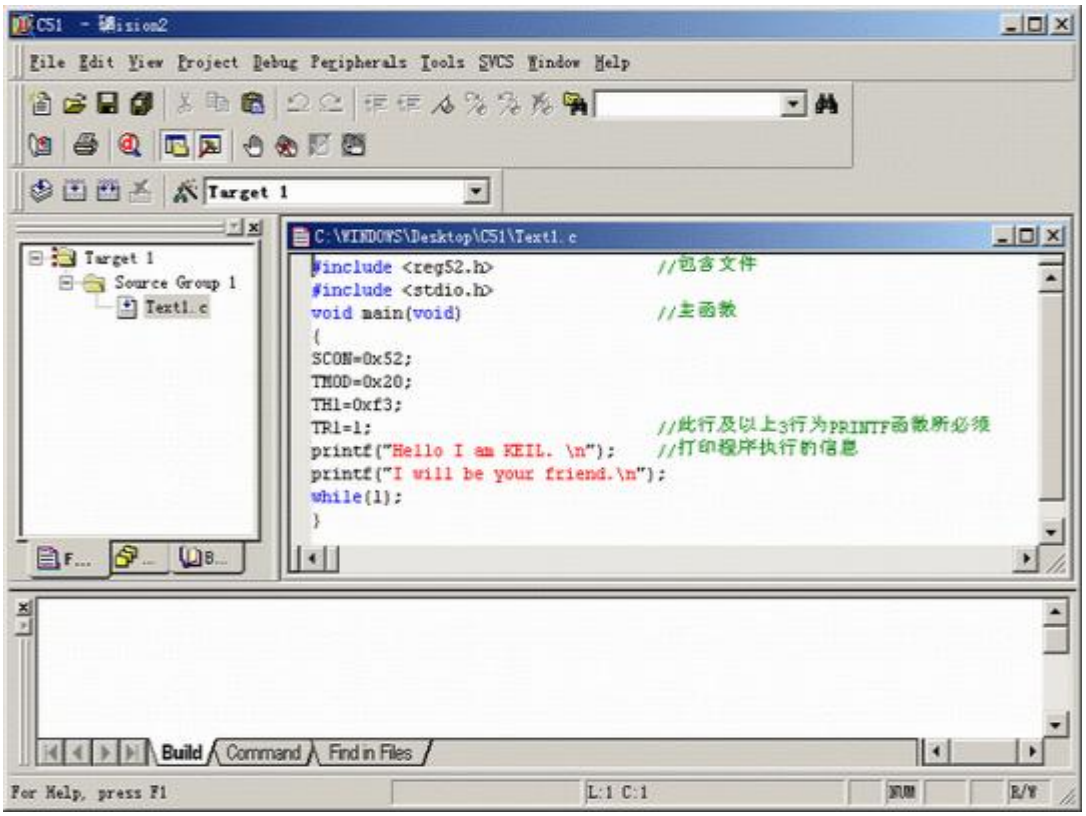


图 2-16　编写好的 Text 1.c 文件界面

8.　对编写的程序进行编译。

在上图中，单击“Project”菜单，再在下拉菜单中单击“Built Target”选项（或者使用快捷键 F7），编译成功后，再单击“Project”菜单，在下拉菜单中单击“Start/Stop Debug Session”（或者使用快捷键 Ctrl+F5）,屏幕如下所示

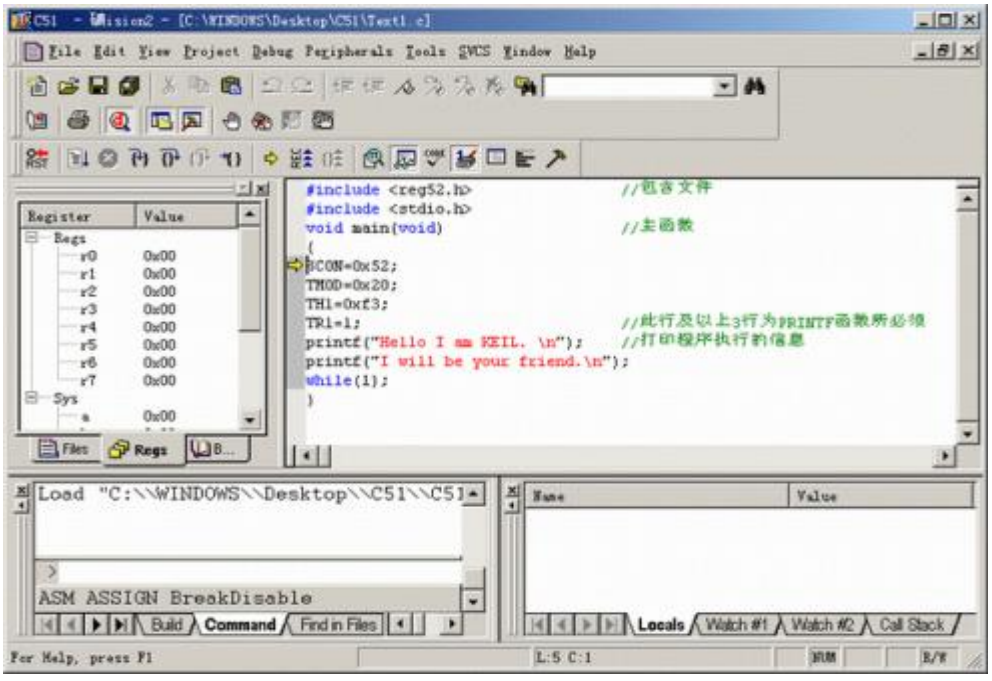


图 2-17　程序编译界面

9.　调试程序

在上图中，单击“Debug”菜单，在下拉菜单中单击“Go”选项，（或者使用快捷键 F5），然后再单击“Debug”菜单，在下拉菜单中单击“Stop Running”选项（或者使用快捷键 Esc）；再单击“View”菜单，再在下拉菜单中单击“Serial Windows #1”选项，就可以看到程序运行后的结果，其结果如下图所示

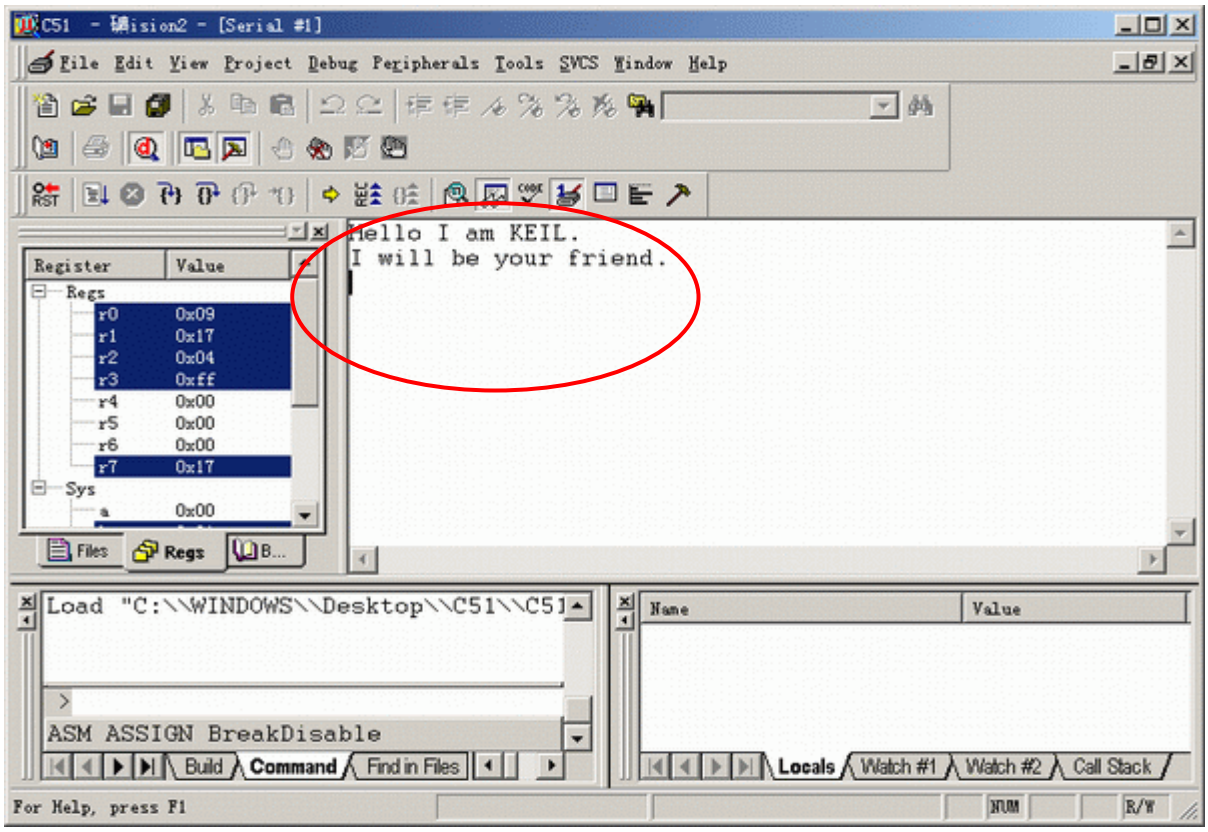


图 2-18　程序执行后的结果

至此，我们在 Keil C51 上做了一个完整工程的全过程。但这只是纯软件的开发过程，如何使用程序下载器看一看程序运行的结果呢？

10.　下载文件*.hex 生成

单击“Project”菜单，再在下拉菜单中单击“ Options for Target 'Target 1' ”在下图中，击“Output”中单击“Create HEX File” 选项，使程序编译后产生 HEX 代码。这样就可以通过我们实验一学习到的程序下载方法，将*.hex 文件下载到 STC89C52 单片机中。这样就完成了一次单片机程序的编写

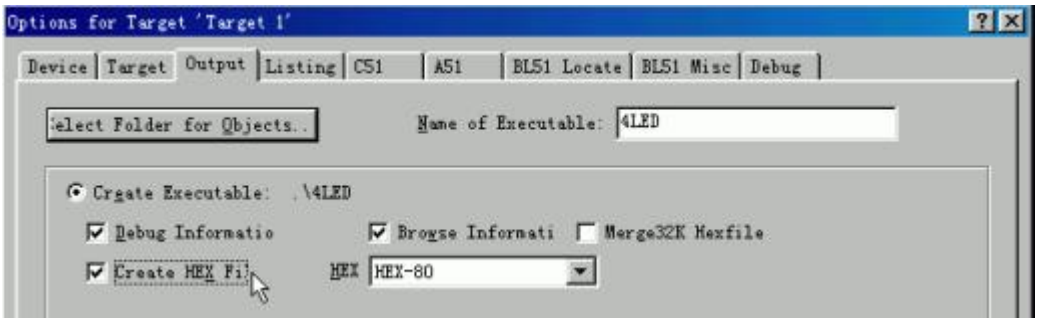


图 2-19　生成*.hex 文件

总结：

通过实验一程序下载，和本次实验的程序编程，我们就比较完整的完成了一次单片机程序开发。从开发的过程我们可以知道，单片机编程用到了两个软件：Keil C51 和 STC-ISP。前者软件是我们用来写程序、生成*.hex 目标代码的；后者软件是我们用来往单片机里面下载程序的。

不同厂家的单片机，程序编写和下载方法也略有不同，但开发过程是大同小异，只要我们掌握了这种方法，以后就可以很方便的学习、开发其他单片机产看。

【范例路径】

三自由度机械手实验例程\Example\EX12_Text1

【作业】

1. 建立一个工程文件，编写一个 C 语言程序，从 1 到 10 进行累加，并把计算结果保存在[Sum]单元里；
2. 建立一个工程文件，编写一个 C 语言程序，从 1 到 10 进行累加，并把计算结果保存在[Sum]单元里；

实验三　单片机控制电磁阀实验

【实验目的】

1. 学习电磁阀工作原理；
2. 学习单片机控制气缸运动的方法；

【涉及知识面】

该实验涉及《单片机原理与应用》、《气压传动》等知识

【实验工具】

计算机一台、三自由度气动机械手一台、气泵。

【实验要求】

1. 能看到电磁阀驱动电路图。
2. 会编写简单的程序控制机械手运动。

【硬件连接】

表 1 单片机 IO 与电磁阀对应表

硬件电路接口	单片机 IO 状态	气缸运动情况
FA1 — P1.4	0	电磁阀打开；水平气缸伸出；
	1	电磁阀关闭；水平气缸收缩；
FA2 — P1.3	0	电磁阀打开；竖直气缸伸出；
	1	电磁阀关闭；竖直气缸收缩；
FA3 — P1.2	0	电磁阀打开；手爪张开；
	1	电磁阀关闭；手爪闭合；

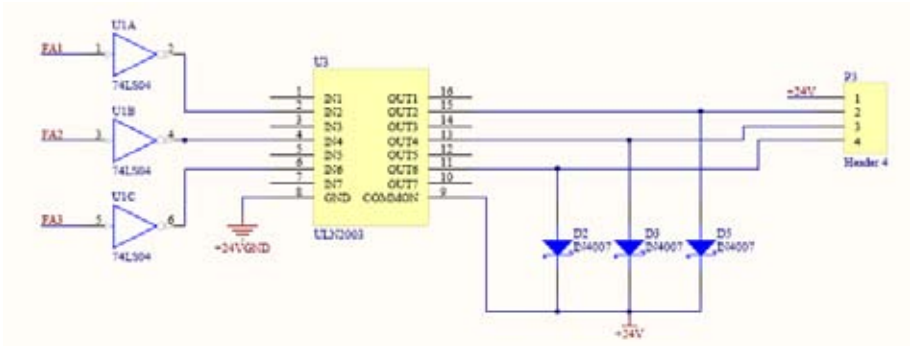


图 3-1 电磁阀驱动电路图

将单片机主板连接到计算机；将按键 K3 按下置于“ON”、按键 K4 弹起置于“OFF”；

按照图 3-1 将电磁阀连接到单片机主板上。

【实验原理】

气动电磁阀在系统中是属于控制元件，而不是执行机构。气动电磁阀一般分为直动式和先导式二种。他们的工作原理如下：。

首先是直动式：直动式的电磁阀是靠电磁线圈通电后的吸力来吸合一根铁芯，这根铁芯的上下移动会控制一个小孔的打开或者关闭状态，而小孔的打开或关闭状态能直接导致进气口和出气口相通或被隔离。

直动式的电磁阀由于受到电磁力的限制，小孔的通径不能做的很大，因此，直动式的电磁阀的流量一般很小。

然后是先导式，先导式的电磁阀是靠阀体内的一根滑芯的左右移动来实现 2 位，或者 3 位的机能，从而达到不同的工作状态的要求（气缸的推出或者退回）状态。滑芯的移动靠先导气来作用在他的二个端面，来推动滑芯的左右移动。

先导式的电磁阀的流量可以很大。

三自由度气动机械手选用的是三个二位五通单电控电磁阀，通过一个汇流板连在一起。分别控制三个气缸。工作电压为 24V。

【实验步骤】

1. 阅读电路原理图，连接单片机的硬件分配；
2. 建立一个新工程，编辑程序；
3. 将程序编译后下载到单片机里面；
4. 看机械手是否按照程序设定的功能运动。

【程序代码】

```
;=============================================================
;  文件名称：     Cylinder.ASM
;  功能描述:      控制电磁阀完成气缸运动控制。
;                 参考外围接线：FA1 — P1.4
;                               FA2 — P1.3
;                               FA3 — P1.2
;  文件来源：     《三自由度气动机械手实验指导书》第一章
;  IDE 环境：     KEIL C51
;  涉及的库：      无
;  组成文件:    cylinder.ASM
;  日期:          2010-3-25
;=============================================================

;==================定义电磁阀单片机控制接口==================
```

```
FA1 EQU P1.4        ;定义阀 1 控制接口，低电平时电磁阀打开。
FA2 EQU P1.3        ;定义阀 2 控制接口，低电平时电磁阀打开。
FA3 EQU P1.2        ;定义阀 3 控制接口，低电平时电磁阀打开。


        ORG     0000h
        AJMP    MAIN
        ORG     0030h

;================================================
;主程序功能：要求水平气缸伸出到最远端，竖直气缸伸到最低端，模拟手抓夹取工件后,
;            竖直气缸缩回最高端，水平气缸缩回最近端，模拟手抓放下工件，循环。
;================================================
MAIN:     MOV   P1,#0FFH    ;P1 口赋初值,防止上电初始打开电磁阀

          LCALL SHEN         ;调用水平气缸伸出子程序
          LCALL DEL1S        ;调用延时 1 秒子程序等待水平气缸伸到最远端
          LCALL XIA          ;调用竖直气缸向下伸出子程序
          LCALL DEL1S        ;调用延时 1 秒子程序等待垂直气缸伸到最低端
          LCALL KAI          ;调用手抓气缸打开子程序
          LCALL DEL1S        ;调用延时 1 秒子程序等待手抓打开
          LCALL HE           ;调用调用手抓闭合子程序
          LCALL DEL1S        ;调用延时 1 秒子程序等待手抓闭合
          LCALL SHANG        ;调用竖直气缸向上缩回子程序
          LCALL DEL1S        ;调用延时 1 秒子程序等待竖直气缸到达最高端
          LCALL SUO          ;调用水平气缸缩回最近端子程序
          LCALL DEL1S        ;调用延时 1 秒子程序等待水平气缸到达最近端
          LCALL KAI          ;调用手抓气缸打开子程序
          LCALL DEL1S        ;调用延时 1 秒子程序等待手抓打开

          LJMP   MAIN

;================================================
;水平气缸伸出子程序，手爪水平伸出。
SHEN:      CLR   FA1
           LCALL DEL2S

           RET
;================================================
;;水平气缸缩回子程序，手爪水平缩回。
SUO:       SETB    FA1
           LCALL DEL2S

           RET
```

```
;================================================
;竖直气缸缩回子程序，手爪上升。
SHANG:      SETB     FA2
            LCALL DEL2S

             RET

;================================================
;;竖直气缸伸出子程序，手爪下降。
XIA:        CLR    FA2
            LCALL DEL2S

            RET

;================================================
;;手爪气缸缩回子程序，手爪闭合。
HE:         SETB     FA3
            LCALL DEL2S

            RET

;================================================
;;手爪气缸缩回子程序，手爪张开。
KAI:        CLR    FA3
            LCALL DEL2S

            RET



;======延时 1S========
;^^^误差:相差 3 微秒^^^
DEL1S: MOV R5,#0A7H
DL1S0: MOV R6,#0ABH
DL1S1: MOV R7,#010H
       DJNZ R7,$
       DJNZ R6,DL1S1
       DJNZ R5,DL1S0
       RET
;======延时 2S========
;^^^误差:超出 222 微秒^^^
;======延时 2S========
;^^^误差:超出 731 微秒^^^
DEL2S: MOV R5,#0FFH
```

```
DL2S0: MOV R6,#0FDH
DL2S1: MOV R7,#02H
       DJNZ R7,$
       DJNZ R6,DL2S1
       DJNZ R5,DL2S0
       RET

       End
```

【范例路径】

三自由度机械手实验例程\Example\EX13_Cylinder

【作业】

1. 使用 C 语言完成电磁阀控制程序；
2. 思考：为什么在驱动电磁阀的时候，要在驱动电路前加一个反相器 74LS04?

实验四　单片机控制步进电机实验

【实验目的】

1. 学习步进电机驱动原理；
2. 学习步进电机编程方法；

【涉及知识面】

该实验涉及《单片机原理与应用》、《步进电机驱动》等知识

【实验工具】

计算机一台、三自由度气动机械手一台、气泵。

【实验要求】

1. 会使用、设置步进电机驱动器。
2. 会编写简单的程序控制步进电机转动。

【硬件连接】

表 1 单片机 IO 与步进电机驱动对应表

硬件电路接口	步进电机运动状态
CP — P1.1	定义步进电机脉冲控制接口；脉冲频率越高转动越快
DIR — P1.0	定义步进电机方向控制接口；0：正转；1：反转

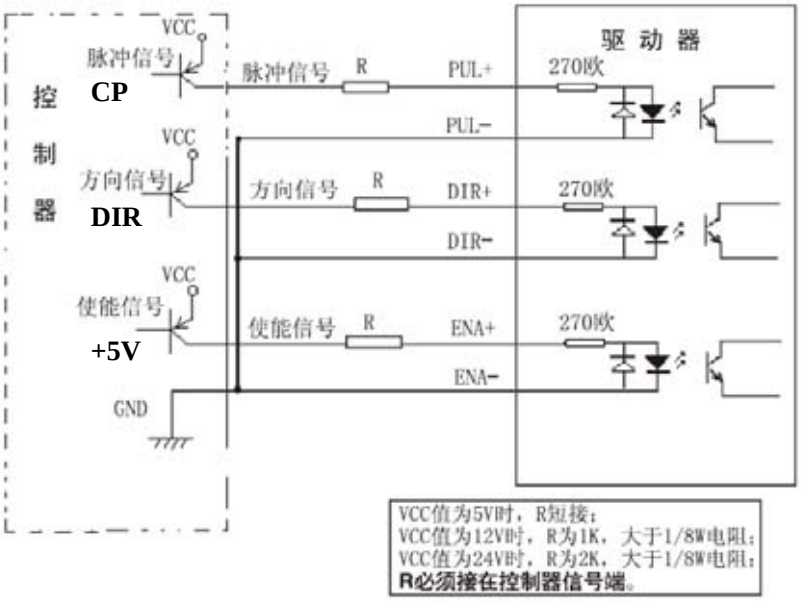


图 4-1 单片机和电机驱动器接口图

将单片机主板连接到计算机；将按键 K3 按下置于“ON”、按键 K4 弹起置于“OFF”；

按照图 4-1 将步进电机连接到单片机主板上。

【实验原理】

1. 步进电机工作原理

步进电动机是一种将电脉冲信号转换成角位移或线位移的机电元件。步进电动机的输入量是脉冲序列，输出量则为相应的增量位移或步进运动。正常运动情况下，它每转一周具有固定的步数；做连续步进运动时，其旋转转速与输入脉冲的频率保持严格的对应关系，不受电压波动和负载变化的影响。由于步进电动机能直接接受数字量的控制，所以特别适宜采用微机进行控制。

步进电动机不能直接接到工频交流或直流电源上工作，而必须使用专用的步进电动机驱动器，它由脉冲发生控制单元、功率驱动单元、保护单元等组成。

机械手的水平旋转电机，使用的是四线两相 1.8° 步距角的混合式步进电机。

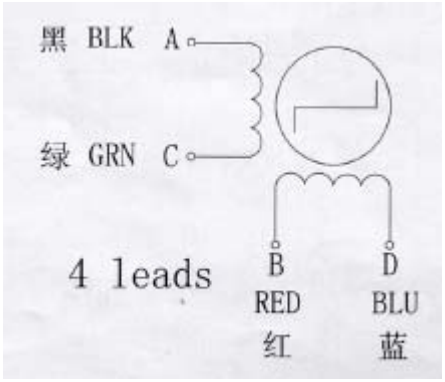


图 4-2 步进电机接线图

2. 步进电机驱动器设置

机械手选配的步进电机驱动器微步细分数有 15 种，最大步数为 25000Pulse/rev；其工作峰值电流范围为 1.0A－4.2A，输出电流共有 8 档，电流的分辨率约为 0.45A；具有自动半流，过压和过流保护等功能。

通过驱动器上面的拨码开关，我们可以设置驱动器的工作电流和细分数。具体的用户可以查阅步进电机驱动器说明书。

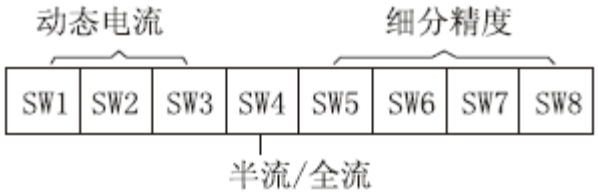


图 4-3 步进电机驱动器设置

【实验步骤】

5. 阅读电路原理图，连接单片机的硬件分配；

6. 按照实验二学习的单片机编程方法，建立一个新工程，编辑程序；

7. 将程序编译后下载到单片机里面；

8. 看机械手是否按照程序设定的功能运动。

【程序代码】

```
;=============================================================
;  文件名称：      Stepmotor.ASM
;  功能描述:       单片机控制步进电机转动。
;                  参考外围接线：CP — P1.1
;                                DIR— P1.0
;  文件来源：      《三自由度气动机械手实验指导书》第一章
;  IDE 环境：      KEIL C51
;  涉及的库：      无
;  组成文件:       stepmother.ASM
;  日期:           2010-3-25
;=============================================================

;================定义步进电机单片机控制接口================

        CP    EQU P1.1        ;定义步进电机脉冲控制接口
        DIR EQU P1.0          ;定义步进电机方向控制接口

        XIFENSHU EQU 22H      ;定义细分数存储单元

        ORG     0000h
        AJMP    MAIN
        ORG     0030h
;=================================================
;主程序功能：步进电机转动一定角度。
;=================================================
MAIN:     MOV     XIFENSHU,#8;细分数设置，此处为 8 细分，与步进电机驱动器设置有关。

          MOV     P1,#0FFH    ;P1 口赋初值

          MOV   R4,#18        ;设置步进电机运转的固定步距角个数，可以控制旋转角度，
                              ;旋转角度=(R4 的值+2)=固定步距角，加 2 是因为会有 2 个启动脉冲。
                          ;以步距角为 1.8°的电机为例，R4 设为 18，则它会旋转(18+2)*1.8°=36°

          SETB    DIR         ;将步进电机方向信号置位使步进电机顺时针旋转
                              ; （注：旋转方向与步进电机两相线圈接线有关）
;=================================================
;设置低频率启动脉冲，脉冲周期 12 毫秒，发送两个脉冲用于低速启动电机，这是计算旋转角度时需要加 2 的原因。
          MOV   R1,XIFENSHU ;将细分数计算在内。
ZUO1:
```

```
        CLR     CP              ;CP 端口输出低电平
        LCALL DEL7MS            ;低电平保持 7 毫秒
        SETB    CP              ;CP 端口输出高电平
        LCALL DEL5MS            ;高电平保持 5 毫秒
        CLR     CP              ;CP 端口输出低电平
        LCALL DEL7MS            ;低电平保持 7 毫秒
        SETB    CP              ;CP 端口输出高电平
        LCALL DEL5MS            ;高电平保持 5 毫秒
        DJNZ    R1,ZUO1         ;将细分数计算在内。
;================================================
JXZUO:
        MOV     R1,XIFENSHU ;将细分数计算在内。

JXZUO1:
 ;================================================
 ;设置步进电机运行脉冲：脉冲周期 5 毫秒，正常运转
        CLR     CP              ;CP 端口输出低电平
        LCALL DEL3MS            ;低电平保持 3 毫秒
        SETB    CP              ;CP 端口输出高电平
        LCALL DEL2MS            ;高电平保持 2 毫秒
        DJNZ    R1,JXZUO1       ;将细分数计算在内。

;================================================
        DJNZ    R4,JXZUO        ;步进电机运行脉冲数，R4 中值减 1 不为零则发一个脉冲，R4 中 198
                                ;加上前面 2 个，共 200 个。

        CLR     DIR             ;将步进电机方向信号置零使步进电机逆时针旋转
                                ;（注：方向与步进电机两相线圈接线有关）
;================================================
;设置低频率启动脉冲，脉冲周期 12 毫秒，发送两个脉冲用于低速启动电机，这是计算旋转角度时
需要加 2 的原因。
        MOV     R1,XIFENSHU ;将细分数计算在内。

YOU1:
        CLR     CP              ;CP 端口输出低电平
        LCALL DEL7MS            ;低电平保持 7 毫秒
        SETB    CP              ;CP 端口输出高电平
        LCALL DEL5MS            ;高电平保持 5 毫秒
        CLR     CP              ;CP 端口输出低电平
        LCALL DEL7MS            ;低电平保持 7 毫秒
        SETB    CP              ;CP 端口输出高电平
        LCALL DEL5MS            ;高电平保持 5 毫秒
        DJNZ    R1,YOU1         ;将细分数计算在内。
;================================================
JXYOU:
```

```
        MOV    R1,XIFENSHU    ;将细分数计算在内。

JXYOU1:
 ;=================================================
 ;设置步进电机运行脉冲：脉冲周期 5 毫秒，占空比为 3：2，正常运转。
        CLR    CP             ;CP 端口输出低电平
        LCALL DEL3MS          ;低电平保持 3 毫秒
        SETB   CP             ;CP 端口输出高电平
        LCALL DEL2MS          ;高电平保持 2 毫秒
        DJNZ   R1,JXYOU1      ;将细分数计算在内。
;=================================================
        DJNZ   R4,JXYOU       ;步进电机运行脉冲数，R4 中值减 1 不为零则发一个脉冲，R4 中 198，
加上前面 2 个，共 200 个。


        LJMP   MAIN


;======延时 1S========
;^^^误差:相差 3 微秒^^^
DEL1S: MOV R5,#0A7H
DL1S0: MOV R6,#0ABH
DL1S1: MOV R7,#010H
        DJNZ R7,$
        DJNZ R6,DL1S1
        DJNZ R5,DL1S0
        RET
;======延时 2S========
;^^^误差:超出 222 微秒^^^
;======延时 2S========
;^^^误差:超出 731 微秒^^^
DEL2S: MOV R5,#0FFH
DL2S0: MOV R6,#0FDH
DL2S1: MOV R7,#02H
        DJNZ R7,$
        DJNZ R6,DL2S1
        DJNZ R5,DL2S0
        RET
;======延时 7MS========
;^^^误差:相差 1 微秒^^^
 DEL7MS:
    MOV R6,#0D0H
DL7MS0:
    MOV R5,#0EH
    DJNZ R5,$
```

```
    DJNZ R6,DL7MS0
    RET
 ;======延时 3MS========
;^^^误差:相差 1 微秒^^^
 DEL3MS:     ;误差  -0.868055555556us
    MOV R6,#0FBH
DL3MS0:
    MOV R5,#04H
    DJNZ R5,$
    DJNZ R6,DL3MS0
    RET

;======延时 2MS========
;^^^误差:相差 1 微秒^^^
DEL2MS:
    MOV R6,#50H
DL2MS0:
    MOV R5,#0AH
    DJNZ R5,$
    DJNZ R6,DL2MS0
    RET
;======延时 5MS========
;^^^误差:相差 1 微秒^^^
DEL5MS: MOV R5,#0EEH
DL5MS0: MOV R6,#09H
        DJNZ R6,$
        DJNZ R5,DL5MS0
        RET

        end
```

【常见问题】

现象	可能问题	解决措施
电机不转	电源灯不亮	检查供电电路，正常供电
	电机轴有力	脉冲信息号弱，信号电流加大至7-16mA
	细分太小	选对细分
	电流设定是否太小	选对电流
	驱动器已保护	重新上电
	使能信号为低	此信号拉高或不接
	对控制信号不反应	未上电
电机转向错误	电机线接错	任意交换电机同一相的两根线（例如A+、A-交换接线位置）
	电机线有断路	检查并接对
报警指示灯亮	电机线接错	检查接线
	电压过高或过低	检查电源
	电机或驱动器损坏	更换电机或驱动器
位置不准	信号受干扰	排除干扰
	屏蔽地未接或未接好	可靠接地
	电机线有断路	检查并接对
	细分错误	设对细分
	电流偏小	加大电流
电机加速时堵转	加速时间太短	加速时间加长
	电机扭矩太小	选大扭矩电机
	电压偏低或电流太小	适当提高电压或电流

【范例路径】

三自由度机械手实验例程\Example\EX14_Stepmotor

【作业】

1. 使用 C 语言完成步进电机控制程序；
2. 思考：改变步进电机驱动器的细分数会什么什么现象？

实验五　手柄液晶显示实验

【实验目的】

1. 了解液晶显示原理；
2. 复习手柄程序下载方法；
3. 学习液晶显示器编程方法。

【涉及知识面】

该实验涉及《单片机原理与应用》、《点阵 LCD 编程》等知识

【实验工具】

计算机一台、三自由度气动机械手一台

【实验要求】

1. 会对手柄进行编程；
2. 能在液晶屏上修改显示的内容。

【硬件连接】

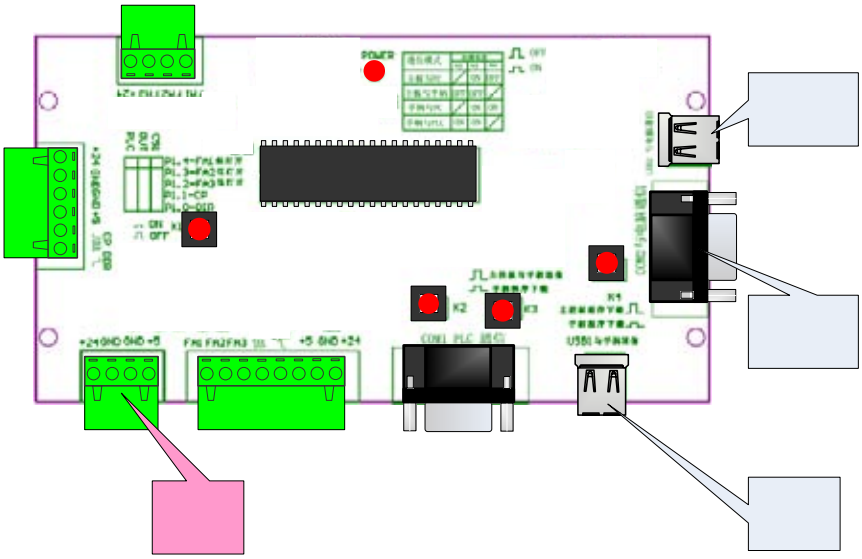


图 5-1　接线图

将单片机主板和计算机、手柄和单片机主板按照图 5-1 方式连接；

将按键 K3、K4 按下，置于“ON”的状态，实现手柄和 PC 机的连接；

P1.0 — DATO	P1.1 — DAT1
P1.2 — DAT2	P1.3 — DAT3
P1.4 — DAT4	P1.5 — DAT5
P1.6 — DAT6	P1.7 — DAT7
P2.4 — E	P2.5 — RW
P2.6 — RS	P3.2 — BACK

【实验原理】

手柄使用的液晶模组是一款蓝底白字，分辨率为 128*64 的并口点阵液晶。型号为 OCMJ4X8C。可显示 4 行汉字，每行可显示 8 个汉字。

引脚	名称	方向	说明	引脚	名称	方向	说明
1	VSS	-	GND (0V)	11	DB4	I/O	数据 4
2	VDD	-	Supply Voltage For Logic (+5V)	12	DB5	I/O	数据 5
3	VO	-	Supply Voltage For LCD (悬空)	13	DB6	I/O	数据 6
4	RS(CS)	I	H: Data L: Instruction Code (chip enable for serial mode)	14	DB7	I/O	数据 7
5	R/W(STD)	I	H: Read L: Write (serial data for serial mode)	15	PSB	I	H: Parallel Mode L: Serial Mode
6	E(SCLK)	I	Enable Signal，高电平有效 (serial clock)	16	NC	-	空脚
7	DB0	I/O	数据 0	17	/RST	I	Reset Signal，低电平有效
8	DB1	I/O	数据 1	18	NC	-	空脚
9	DB2	I/O	数据 2	19	LEDA	-	背光源正极 (+5V)
10	DB3	I/O	数据 3	20	LEDK	-	背光源负极 (0V)

图 5-2 液晶数据接口

液晶是单片机应用中较为复杂的外围模块。希望同学们能利用课余时间好好参考一下液晶的数据手册，以便更快的掌握液晶编程。

在本次实验例程中有一次修改液晶整屏显示内容的程序，同学可以通过修改该程序段，实现每次部分显示内容，也可再加一些其他功能，比如在液晶一个固定的位置显示 001~100，或者显示一幅图片等等，这些程序读者通过研读液晶的说明书，都可以写出来。

该液晶还有其他的显示控制方式，功能较多，使用起来较为方便，限于篇幅本实验只介绍了最常用的查表方式，只当抛砖引玉，其他的显示方式留给同学们去研究

【实验步骤】

1. 阅读手柄电路原理图，熟悉手柄单片机和液晶模组接口；
2. 阅读液晶模组数据手册，了解液晶模组的时序和指令；
3. 阅读例程；
4. 在例程基础上尝试修改液晶显示的内容。

【程序代码】

```
;=========================================================
;  文件名称：     SBLCD.ASM
```

```
;  功能描述:      在手柄液晶显示屏上显示四行字符。
;                 参考外围接线：P1.0 — DATO
;                               P1.1 — DAT1
;                               P1.2 — DAT2
;                               P1.3 — DAT3
;                               P1.4 — DAT4
;                               P1.5 — DAT5
;                               P1.6 — DAT6
;                               P1.7 — DAT7
;                               P2.4 — E            //使能
;                               P2.5 — RW
;                               P2.6 — RS
;                               P3.2 — BACK         //背光控制
;  文件来源：    《三自由度气动机械手实验指导书》第一章
;  IDE 环境：    KEIL C51
;  涉及的库：    无
;  组成文件:     SBLCD.ASM
;  日期:         2010-3-25
;============================================================


;================定义液晶的硬件连接（开始）==========================
BACK     EQU      P3.2  ;液晶背光控制脚，低电平时液晶背光打开，高电平关闭。
RS   EQU P2.4   ;RS 脚
RW   EQU P2.5   ;RW 脚
E    EQU P2.6   ;E 脚
PSB  EQU P3.3   ;PSB 脚
RST  EQU P3.5   ;RST 脚
LCDDATA EQU      P1     ;8 位数据脚
;================定义液晶的硬件连接（结束）==========================


;================定义液晶相关数据（开始）==========================
COUNT    EQU 34H    ;用于设置一次发送的字符个数
;================定义液晶相关数据（结束）==========================


                ORG     00H
                AJMP    MAIN
                ORG     30H

;================主程序（开始）==========================

;主程序功能：初始化液晶，然后将表格 1 中的字在液晶屏上显示，延时 5 秒钟，
```

```
;               在液晶屏上显示表格 2 中的字，延时 5 秒钟，然后返回程序开始，循环执行。
MAIN:
        NOP
        NOP                     ;几个空指令，无意义。

        CLR     BACK            ;打开液晶背光，背光低电平打开，高电平关闭。
        LCALL SETLCD            ;调用液晶初始化程序，初始化液晶
        MOV     DPTR,#TAB1      ;将指针指向 表格 1（TAB1）,为显示做准备
        LCALL DISPLAY           ;调用显示字符子程序，在整个屏幕显示表格 1 中的字符

        LCALL   DEL1S           ;调用 5 次延时 1s 子程序，在两组不同的现实内容之间延时 5 秒钟。
        LCALL   DEL1S
        LCALL   DEL1S
        LCALL   DEL1S
        LCALL   DEL1S

        MOV     DPTR,#TAB2      ;将指针指向 表格 2（TAB2）,为显示做准备
        LCALL   DISPLAY         ;调用显示字符子程序，在整个屏幕显示表格 2 中的字符，

        LCALL   DEL1S           ;调用 5 次延时 1s 子程序，在两组不同的现实内容之间延时 5 秒钟。
        LCALL   DEL1S
        LCALL   DEL1S
        LCALL   DEL1S
        LCALL   DEL1S

        LJMP    MAIN            ;延时结束，跳转到程序开始，继续循环执行。


;====================液晶初始化子程序（开始）======================
;液晶初始化子程序功能：用于在系统上电或者必要时初始化液晶，清空屏幕显示，设置光标的移动方式，
;                       使能显示，关掉光标并且不闪烁
;                       此程序没有入口参数，直接调用即可初始化液晶。
;根据不同的初始化需要，更改里面的命令参数即可修改初始化状态


SETLCD:
        SETBPSB         ;置高液晶的 PSB 脚
        SETBRST         ;置高液晶的 RST 脚

LGS0:   MOV     A,#34H          ;34H--扩充指令操作
```

```
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 34H，发送给液晶。
        MOV     A,#30H          ;30H--基本指令操作
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 30H，发送给液晶。
        MOV     A,#01H          ;清除显示
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 01H，发送给液晶。
        LCALL   DELAY2          ;调用一个短的延时程序，给液晶反应得时间。
        MOV     A,#06H          ;06H--指定在资料写入或读取时，光标的移动方向
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 06H，发送给液晶。
        MOV     A,#0CH          ;0CH--开显示,关光标,不闪烁
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 0CH，发送给液晶。
          RET                   ;液晶初始化子程序结束，返回。
;====================液晶初始化子程序（结束）======================
```

```
;====================液晶显示汉字和字符（开始）======================
;液晶显示汉字和字符子程序功能：在液晶上显示 4 行字，程序的入口参数为一个表格地址，表格中是要显示的字符的 ASCII 码。
;                               直接将表格的地址赋值给 DPTR 指针，然后调用此程序即可。


DISPLAY:
        MOV     COUNT,#40H      ;设置要显示的字符个数，因为整个液晶屏总共可以显示 40H（64）个字符，因此设为 40H（64）。

        MOV     A,#80H          ;设置显示的字符的起始地址，即从液晶上地址为 80H 的位置开始显示字符，80H 为液晶左上角开始的地址。
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 80H，发送给液晶。
DISPLAY1:
        CLR A                   ;清除 A 的内容，防止对下面的程序产生影响。

          MOVC  A,@A+DPTR       ;将表格中的内容取出来并赋值给 A，为发送做准备。
        LCALL   SEND_D          ;调用向液晶发送数据子程序，将 A 中的数据（即从表格中取出的数据），发送给液晶。
        INC  DPTR               ;将数据指针加一，指向表格中的下一个数据，以便取出下一个数据。
        DJNZ    COUNT,DISPLAY1;判断一下是否将数据完全取完，如果取完就向下执行，如果没有就继续取数据，COUNT 的值为需要取得字符

的个数。
        LCALL   DELAY3          ;数据已经取完，短延时一下，给予显示时间。

          RET                   ;子程序结束，程序返回。
;====================液晶显示汉字和字符（结束）======================
```

```
;====================向液晶发送数据子程序（开始）======================
;向液晶发送数据子程序功能:向液晶发送数据，入口参数为 A，将要发送的数据放入 A 中，然后调用此程序即可将要发送的数据发送到液晶。

SEND_D: LCALL   CHK_BUSY        ;调用检查忙碌子程序，发送数据前需先检查液晶是否忙碌，不忙碌才可发送数据。
    SETB    RS              ;置高液晶的 RS 脚
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

    CLR RW              ;置高液晶的 RW 脚
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

    MOV     LCDDATA,A       ;将要发送的数据送到数据端口（即 P1）。
      NOP
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    SETB    E               ;置高液晶的 E 脚
    NOP
    NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    CLR E               ;置低液晶的 E 脚
      NOP
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    NOP
    MOV     LCDDATA,#0FFH ;将数据端口全部设为高电平，防止误动作。
      NOP
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    RET                     ;子程序结束，返回。
;====================向液晶发送数据子程序（结束）======================


;====================向液晶写指令子程序（开始）======================
;向液晶写指令子程序功能:向液晶发送指令，入口参数为 A，将要发送的指令放入 A 中，然后调用此程序即可将要发送的指令发送到液晶。

SEND_I:
    LCALL   CHK_BUSY        ;调用检查忙碌子程序，发送指令前需先检查液晶是否忙碌，不忙碌才可发送指令。
    CLR RS              ;置高液晶的 RS 脚
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
      NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

    CLR RW              ;置高液晶的 RW 脚
```

```
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

        MOV  LCDDATA,A      ;将要发送的指令送到数据端口（即 P1）。

        NOP
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    SETB     E              ;置高液晶的 E 脚
    NOP
    NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    CLR E                ;置低液晶的 E 脚
        NOP
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    NOP
    MOV      LCDDATA,#0FFH ;将数据端口全部设为高电平，防止误动作。
        NOP
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    RET                     ;子程序结束，返回。


;====================检测液晶忙碌子程序（开始）======================
;检测液晶忙碌子程序功能:检测液晶是否忙碌，如果液晶不忙碌，子程序返回，
;                        如果液晶忙碌，会一直在子程序中等待，直到液晶不忙碌后再返回。
CHK_BUSY:
    CLR RS               ;置低液晶的 RS 脚

        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        NOP

    SETB     RW              ;置高液晶的 RW 脚
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        NOP

    SETB     E              ;置高液晶的 E 脚
    NOP
    NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。


    JB   P1.7,$          ;判断数据端口的最高位是高还是低，
                            ;如果是高电平，说明液晶忙碌，在此等待，如果为低电平，说明液晶不忙碌，继
续向下执行。
    CLR E                ;置低液晶的 E 脚
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        RET                     ;子程序结束，返回。
```

```
;====================检测液晶忙碌子程序（结束）======================


;短的延时子程序 1，延时很短不准确。
DELAY1:     MOV     R6,#06H
DEL11:      MOV     R7,#08H
DEL12:      DJNZ    R7,DEL12
            DJNZ    R6,DEL11
            RET


;短的延时子程序 2，延时很短不准确。
DELAY2:     MOV     R6,#0CH
DEL21:      MOV     R7,#18H
DEL22:      DJNZ    R7,DEL22
            DJNZ    R6,DEL21
            RET


;短的延时子程序 3，延时很短不准确。
DELAY3:
    MOV R6,#0DDH
DL30:
    MOV R5,#0CFH
    DJNZ R5,$
    DJNZ R6,DL30
    RET


;======延时 1S========
;^^^误差:相差 3 微秒^^^
DEL1S: MOV R5,#0A7H
DL1S0: MOV R6,#0ABH
DL1S1: MOV R7,#010H
        DJNZ R7,$
        DJNZ R6,DL1S1
        DJNZ R5,DL1S0
        RET


;;;要显示的字符的表格，建表格时要注意格式，每一行最多 8 个汉字或者 16 个英文字符。
;;;要注意 4 行的顺序与液晶显示位置的对应关系，要注意加引号，加引号后表格中的内容会是字符的 ASCII 码。
```

;;;要熟悉液晶的说明书，关于液晶显示位置的说明，液晶型号为 OCMJ4X8C，128X64 的。
;;;该例程只显示前两个表格，其余的作为参考。

```
TAB1:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
       DB   '液晶显示             '

TAB2:  DB    '青岛市黄岛          '
       DB   'ONETWOTHREE        '
       DB   '黄海之滨黄海学院'
       DB   '12345678             '

TAB3:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
       DB   '自动执行             '

TAB4:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
       DB   '学习                 '

TAB5:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
       DB   '水平臂缩回         '

TAB6:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
       DB   '竖直臂缩回         '

TAB7:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
       DB   '手爪合闭             '

TAB8:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
       DB   '水平臂伸出         '

TAB9:  DB   '      黄海学院      '
       DB   '当前状态             '
       DB   '三自由度机械手   '
```

```
        DB    '竖直臂伸出          '

TAB10:  DB    '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '手爪张开            '

TAB11: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '等待旋转设置        '

TAB12: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '向右旋转            '

TAB13: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '向左旋转            '

TAB14: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '旋转                '

TAB15: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '向右旋转            '

TAB16: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '向左旋转            '

TAB17: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
        DB    '水平臂收缩学习      '

TAB18: DB     '      黄海学院      '
        DB    '当前状态            '
        DB    '三自由度机械手      '
```

```
        DB    '竖直臂上升学习   '

TAB19: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '手爪合闭学习       '

TAB20: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '水平臂伸出学习   '

TAB21: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '竖直臂下降学习   '

TAB22: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '手爪松开学习       '

TAB23: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '旋转设置学习       '

TAB24: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '向右旋转学习       '

TAB25: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '向左旋转学习       '

TAB26: DB     '      黄海学院      '
        DB    '当前状态           '
        DB    '三自由度机械手   '
        DB    '水平旋转学习       '

    END
```

【范例路径】

三自由度机械手实验例程\Example\EX15_LCD

【作业】

1. 在液晶屏上显示自己的姓名和学号；
2. 思考：如何在液晶屏上画直线、矩形、圆形等几何图形？

实验六　手柄按键扫描显示实验

【实验目的】

1. 了解矩阵键盘的电路和编程方法；
2. 复习液晶模组编程知识；

【涉及知识面】

该实验涉及《单片机原理与应用》、《点阵 LCD 编程》、等知识

【实验工具】

计算机一台、三自由度气动机械手一台、手柄一台。

【实验要求】

1. 了解矩阵键盘的行扫描算法；
2. 能在例程基础上修改程序。

【硬件连接】

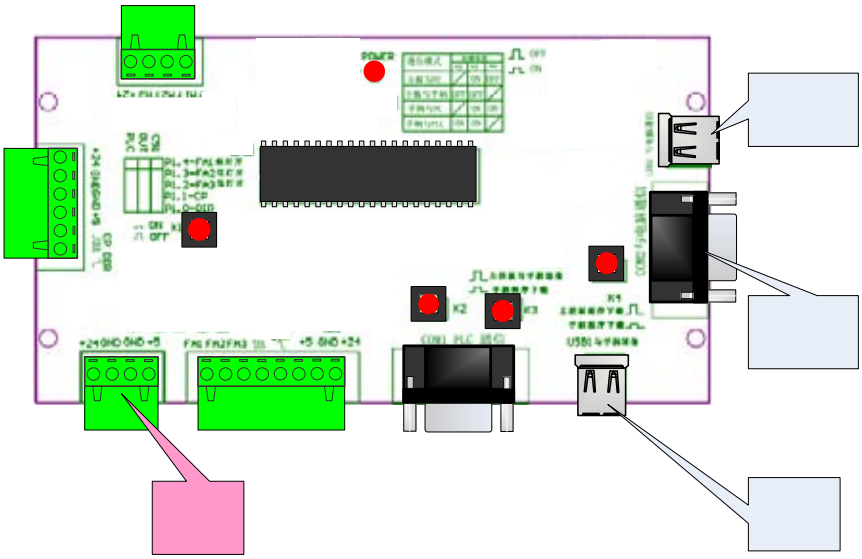


图 6-1 接线图

将单片机主板和计算机、手柄和单片机主板按照图 5-1 方式连接；

将按键 K3、K4 按下，置于“ON”的状态，实现手柄和 PC 机的连接；

P0.0 — P0.6 连接键盘矩阵的：H1— H7；

P0.7、P2.7 连接键盘矩阵的：L1、L2

P1.0 —P1.7 连接液晶模组的 DAT0 —DAT7；

P2.4 — E　　P2.5 — RW　　P2.6 — RS　　P3.2 — BACK

【实验原理】

1. 矩阵式键盘的结构与工作原理

当键盘按键的数量较多时，为了减少 I/O 口的占用，通常将按键排列成矩阵形式，如图 6-2 所示。

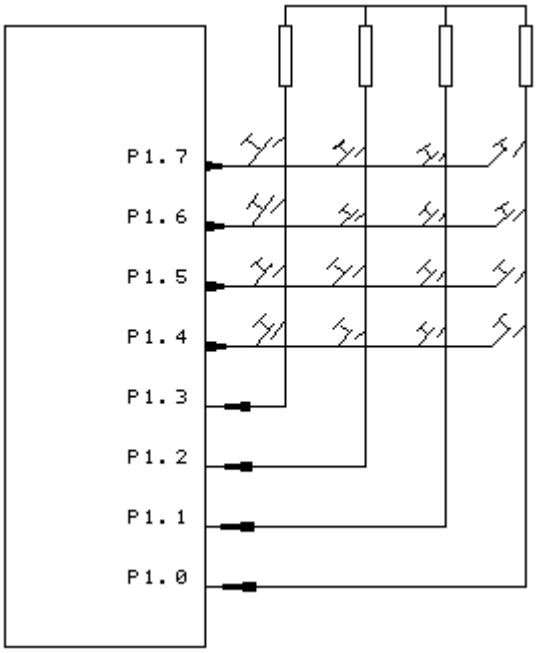


图 6-2 键盘矩阵接线图

在矩阵式键盘中，每条水平线和垂直线在交叉处不直接连通，而是通过一个按键加以连接。这样，一个端口（如 P1 口）就可以构成 4*4=16 个按键，比之直接将端口线用于键盘多出了一倍，而且线数越多，优势越明显，比如再多加一条线就可以构成 20 键的键盘，而直接用端口线则只能多出一键（9 键）。由此可见，在需要的键数比较多时，采用矩阵法来做键盘是合理的。

矩阵式结构的键盘显然比直接法要复杂一些，识别也要复杂一些，上图中，列线通过电阻接正电源，并将行线所接的单片机的 I/O 口作为输出端，而列线所接的 I/O 口则作为输入。这样，当按键没有按下时，所有的输出端都是高电平，代表无键按下。行线输出是低电平，一旦有键按下，则输入线就会被拉低，这样，通过读入输入线的状态就可得知是否有键按下了。具体的识别及编程方法如下所述。

2. 矩阵式键盘的按键识别方法

确定矩阵式键盘上何键被按下介绍一种“行扫描法”。

行扫描法又称为逐行（或列）扫描查询法，是一种最常用的按键识别方法，如图 6-2 所示键盘，即可使用行扫描法。

判断键盘中有无键按下：首先将全部行线 Y0-Y3 置低电平，然后检测列线的状态。只要有一列的电平为低，则表示键盘中有键被按下，而且闭合的键位于低电平线与 4 根行线相交叉的 4 个按键

之中。若所有列线均为高电平，则键盘中无键按下。

判断闭合键所在的位置：在确认有键按下后，即可进入确定具体闭合键的过程。其方法是：依次将行线置为低电平，即在置某根行线为低电平时，其它线为高电平。在确定某根行线位置为低电平后，再逐行检测各列线的电平状态。若某列为低，则该列线与置为低电平的行线交叉处的按键就是闭合的按键。

下面给出一个具体的例子：8031 单片机的 P1 口用作键盘 I/O 口，键盘的列线接到 P1 口的低 4 位，键盘的行线接到 P1 口的高 4 位。列线 P1.0-P1.3 分别接有 4 个上拉电阻到正电源+5V，并把列线 P1.0-P1.3 设置为输入线，行线 P1.4-P.17 设置为输出线。4 根行线和 4 根列线形成 16 个相交点。

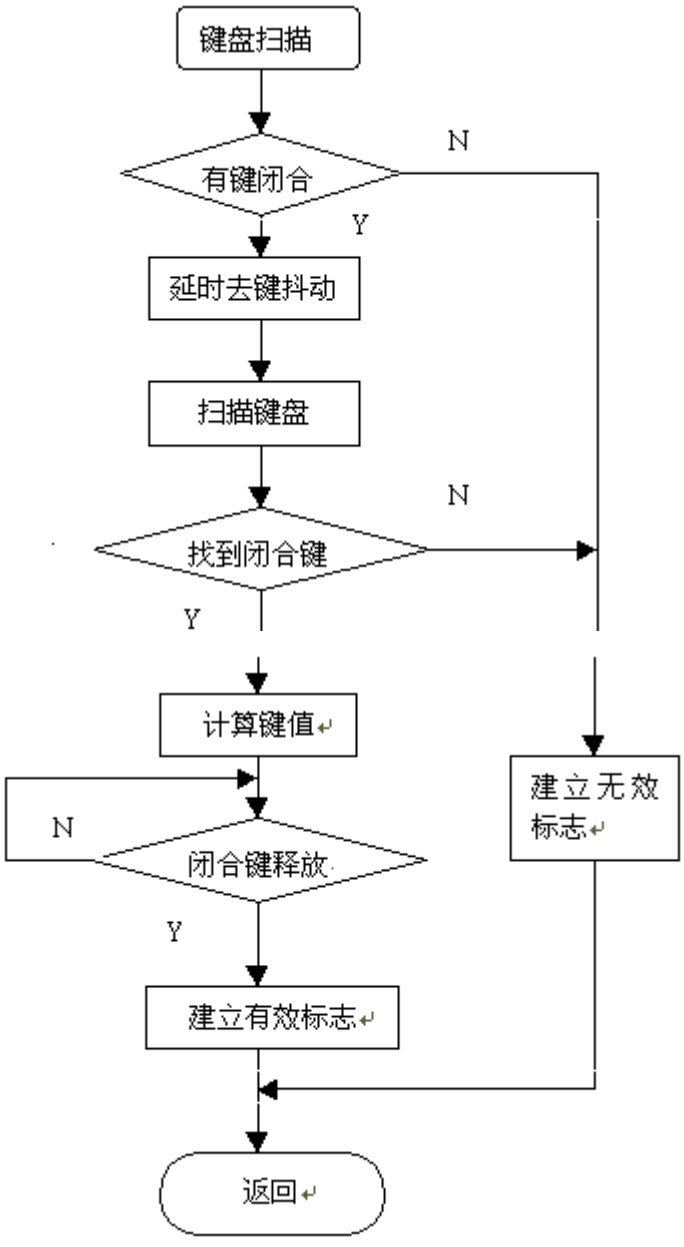


图 6-3 键盘矩阵接线图

(1) 检测当前是否有键被按下。

检测的方法是 P1.4-P1.7 输出全“0”，读取 P1.0-P1.3 的状态，若 P1.0-P1.3 为全“1”，则无键闭合，否则有键闭合。

(2) 去除键抖动。当检测到有键按下后，延时一段时间再做下一步的检测判断。

(3) 若有键被按下，应识别出是哪一个键闭合。方法是对键盘的行线进行扫描。P1.4-P1.7 按下述 4 种组合依次输出：

P1.7 :1 1 1 0　　P1.6:1 1 0 1　　P1.5:1 0 1 1　　P1.4:0 1 1 1

在每组行输出时读取 P1.0-P1.3，若全为“1”，则表示为“0”这一行没有键闭合，否则有键闭合。由此得到闭合键的行值和列值，然后可采用计算法或查表法将闭合键的行值和列值转换成所定义的键值

(4) 为了保证键每闭合一次 CPU 仅作一次处理，必须却除键释放时的抖动

【实验步骤】

1. 阅读手柄电路原理图，熟悉手柄单片机和液晶模组接口；
2. 阅读液晶模组数据手册，了解液晶模组的时序和指令；
3. 阅读例程；
4. 在例程基础上尝试修改液晶显示的内容。

【程序代码】

```
;===============================================================
;  文件名称：     SBAAC09.ASM
;  功能描述:      根据不同按键在液晶屏显示不同内容。
;                 参考外围接线：P1.0 — P1.7 连接 DAT0 — DAT7
;                               P2.4 — E            //使能
;                               P2.5 — RW
;                               P2.6 — RS
;                               P3.2 — BACK         //背光控制
;
;                               P0.0 — P0.6 连接键盘矩阵的：H1— H7；
;                               P0.7、P2.7 连接键盘矩阵的：L1、L2
;  文件来源：     《三自由度气动机械手实验指导书》第一章
;  IDE 环境：     KEIL C51
;  涉及的库：     无
;  组成文件:      SBAAC09.ASM
;  日期:          2010-3-25
;===============================================================


;================定义液晶的硬件连接（开始）==========================
BACK    EQU     P3.2  ;液晶背光控制脚，低电平时液晶背光打开，高电平关闭。
RS  EQU P2.4  ;RS 脚
RW  EQU P2.5  ;RW 脚
E   EQU P2.6  ;E 脚
```

```
PSB  EQU P3.3   ;PSB 脚
RST  EQU P3.5   ;RST 脚
LCDDATA EQU      P1      ;8 位数据脚
;================定义液晶的硬件连接（结束）==========================

;================定义矩阵按键的硬件连接（开始）==========================

H1        EQU      P0.0   ;矩阵按键的第一行
H2        EQU      P0.1   ;矩阵按键的第二行
H3        EQU      P0.2   ;矩阵按键的第三行
H4        EQU      P0.3   ;矩阵按键的第四行
H5        EQU      P0.4   ;矩阵按键的第五行
H6        EQU      P0.5   ;矩阵按键的第六行
H7        EQU      P0.6   ;矩阵按键的第七行


L1        EQU      P2.7   ;矩阵按键的第一列
L2        EQU      P0.7   ;矩阵按键的第二列
;================定义矩阵按键的的硬件连接（结束）==========================

;================定义液晶相关数据（开始）==========================
COUNT    EQU 34H    ;用于设置一次发送的字符个数
;================定义液晶相关数据（结束）==========================



                ORG     00H
                AJMP    MAIN
                ORG     30H
;================主程序（开始）==========================
;主程序功能：初始化液晶，显示一下表格 1 中的字符，然后扫描按键，如果没有按键按下将一直等待，
;                如果有按键按下，则显示按键的内容，延时 2 秒钟，然后返回主程序继续等待按键按下。
;        注意："开/关"按键不仅显示按键内容，还改变背光的状态，如果背光原先打开则关闭它，如果原先关闭则打开它。
    MAIN:        MOV      SP,#60H
                CLR      BACK              ;打开背光
                LCALL    SETLCD            ;初始化液晶
                MOV      DPTR,#TAB1
                LCALL    DISPLAY           ;显示表格 1 中的汉字
                LCALL    DEL2S             ;延时 2 秒钟

JIANCE:                                    ;矩阵按键扫描程序
```

```
        MOV     P0,#0FFH            ;先将要扫描的端口口全部置高，使输入更稳定。

        SETB    L2                  ;扫描第一列，将第二列置高。
        CLR     L1                  ;扫描第一列，将第一列置低。

        JNB     H1,SHEN             ;扫描"水平出"按键是否按下，如果按下跳转到"SHEN"程序处执行,否则向下执行。
        JNB     H2,XIA              ;扫描"垂直伸"按键是否按下，如果按下跳转到"XIA"程序处执行,否则向下执行。
        JNB     H3,KAI              ;扫描"手爪开"按键是否按下，如果按下跳转到"KAI"程序处执行,否则向下执行。
        JNB     H4,XZSZ             ;扫描"旋转设置"按键是否按下，如果按下跳转到"XZSZ"程序处执行,否则向下执行。
        JNB     H5,SPXZ             ;扫描"水平旋转"按键是否按下，如果按下跳转到"SPXZ"程序处执行,否则向下执行。
        JNB     H6, ZIDONG          ;扫描"自动执行"按键是否按下，如果按下跳转到"ZIDONG"程序处执行,否则向下执行。
        JNB     H7, XUEXI1          ;扫描"学习/回放"按键是否按下，如果按下跳转到"XUEXI1"程序处执行,否则向下执行。

        SETB    L1                  ;扫描第二列，将第一列置高。
        CLR     L2                  ;扫描第二列，将第二列置低。
        JNB     H1,SUO1             ;扫描"水平回"按键是否按下，如果按下跳转到"SUO1"程序处执行,否则向下执行。
        JNB     H2,SHANG1           ;扫描"垂直缩"按键是否按下，如果按下跳转到"SHANG1"程序处执行,否则向下执行。
        JNB     H3,HE1              ;扫描"手爪合"按键是否按下，如果按下跳转到"HE1"程序处执行,否则向下执行。
        JNB     H4,DU1811           ;扫描"-1.8°"按键是否按下，如果按下跳转到"DU1811"程序处执行,否则向下执行。
        JNB     H5,DU1821           ;扫描"+1.8°"按键是否按下,如果按下跳转到"DU1821"程序处执行,否则向下执行。
        JNB     H6,SHOUDONG1        ;扫描"手动操作"按键是否按下，如果按下跳转到"SHOUDONG1"程序处执行,否则向下执行。
        JNB     H7,BEIGUANG         ;扫描"开/关"按键是否按下,如果按下跳转到"BEIGUANG"程序处执行,否则向下执行。

        LCALL DEL200MS              ;延时 200MS，去除按键抖动。
        LJMP    JIANCE              ;没有按键按下，继续进行按键扫描。

XUEXI1:         LJMP    XUEXI   ;因为"JNB"指令的跳转范围有限，"XUEXI"程序已经超出"JNB"的跳转范围,所以加一个"XUEXI1"程序,作为跳转的
```

```
跳板.
SUO1:            LJMP   SUO      ;"SUO"的跳板。
SHANG1:          LJMP   SHANG    ;"SHANG"的跳板。
HE1:             LJMP   HE       ;"HE"的跳板。
DU1811:          LJMP   DU181    ;"DU181"的跳板。
DU1821:          LJMP   DU182    ;"DU182"的跳板。
SHOUDONG1:       LJMP   SHOUDONG  ;"SHOUDONG"的跳板。

;================"开/关"按键程序（开始）==========================
;程序功能：改变背光的状态，显示"开/关"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
 BEIGUANG:
                 CPL    BACK              ;背光设置,将背光开关取反，如果背光原先打开则关闭它，如果原先关
闭则打开它。
                 MOV    DPTR,#TAB2        ;将指针指向表格 2，显示"开/关"按键对应的内容。
                 LCALL  DISPLAY           ;显示表格 2 中的汉字
                 LJMP   JIANCE
;================"开/关"按键程序（结束）==========================


;================"水平出"按键程序（开始）==========================
;程序功能：显示"水平出"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
 SHEN:
                 MOV    DPTR,#TAB11       ;将指针指向表格 11，显示"水平出"按键对应的内容。
                 LCALL  DISPLAY           ;显示表格 11 中的汉字
                 LCALL  DEL2S             ;调用延时子程序，延时 2 秒钟
                 LJMP   JIANCE            ;跳转回按键检测程序
;================"水平出"按键程序（结束）==========================

;================"垂直伸"按键程序（开始）==========================
;程序功能：显示"垂直伸"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
XIA:
                 MOV    DPTR,#TAB12       ;将指针指向表格 12，显示"垂直伸"按键对应的内容。
                 LCALL  DISPLAY           ;显示表格 12 中的汉字
                 LCALL  DEL2S             ;调用延时子程序，延时 2 秒钟
                 LJMP   JIANCE            ;跳转回按键检测程序
;================"垂直伸"按键程序（结束）==========================

;================"手爪开"按键程序（开始）==========================
;程序功能：显示"手爪开"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
KAI:
                 MOV    DPTR,#TAB13       ;将指针指向表格 13，显示"手爪开"按键对应的内容。
```

```
                LCALL   DISPLAY          ;显示表格 13 中的汉字
                LCALL   DEL2S            ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE           ;跳转回按键检测程序
;================"手爪开"按键程序（结束）=========================


;================"旋转设置"按键程序（开始）=========================
;程序功能：显示"旋转设置"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
 XZSZ:
                MOV     DPTR,#TAB14      ;将指针指向表格 14，显示"旋转设置"按键对应的内容。
                LCALL   DISPLAY          ;显示表格 14 中的汉字
                LCALL   DEL2S            ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE           ;跳转回按键检测程序
;================"旋转设置"按键程序（结束）=========================

;================"水平旋转"按键程序（开始）=========================
;程序功能：显示"水平旋转"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
SPXZ:
                MOV     DPTR,#TAB15      ;将指针指向表格 15，显示"水平旋转"按键对应的内容。
                LCALL   DISPLAY          ;显示表格 15 中的汉字
                LCALL   DEL2S            ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE           ;跳转回按键检测程序
;================"水平旋转"按键程序（结束）=========================

;================"自动演示"按键程序（开始）=========================
;程序功能：显示"自动演示"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
ZIDONG:
                MOV     DPTR,#TAB16      ;将指针指向表格 16，显示"自动演示"按键对应的内容。
                LCALL   DISPLAY          ;显示表格 16 中的汉字
                LCALL   DEL2S            ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE           ;跳转回按键检测程序
;================"自动演示"按键程序（结束）=========================

;================"学习/回放"按键程序（开始）=========================
;程序功能：显示"学习/回放"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
XUEXI:
                MOV     DPTR,#TAB17      ;将指针指向表格 17，显示"学习/回放"按键对应的内容。
                LCALL   DISPLAY          ;显示表格 17 中的汉字
                LCALL   DEL2S            ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE           ;跳转回按键检测程序
;================"学习/回放"按键程序（结束）=========================
```

```
;================"水平回"按键程序（开始）==========================
;程序功能：显示"水平回"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
 SUO:
                MOV     DPTR,#TAB18       ;将指针指向表格 18，显示"水平回"按键对应的内容。
                LCALL   DISPLAY           ;显示表格 18 中的汉字
                LCALL   DEL2S             ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE            ;跳转回按键检测程序
;================"水平回"按键程序（结束）==========================


 ;================"垂直缩"按键程序（开始）==========================
;程序功能：显示"垂直缩"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
 SHANG:
                MOV     DPTR,#TAB19       ;将指针指向表格 19，显示"垂直缩"按键对应的内容。
                LCALL   DISPLAY           ;显示表格 19 中的汉字
                LCALL   DEL2S             ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE            ;跳转回按键检测程序
;================"垂直缩"按键程序（结束）==========================


;================"手爪合"按键程序（开始）==========================
;程序功能：显示"手爪合"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
 HE:
                MOV     DPTR,#TAB20       ;将指针指向表格 20，显示"手爪合"按键对应的内容。
                LCALL   DISPLAY           ;显示表格 20 中的汉字
                LCALL   DEL2S             ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE            ;跳转回按键检测程序
;================"手爪合"按键程序（结束）==========================


;================"-1.8°"按键程序（开始）==========================
;程序功能：显示"-1.8°"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
DU181:
                MOV     DPTR,#TAB21       ;将指针指向表格 21，显示"-1.8°"按键对应的内容。
                LCALL   DISPLAY           ;显示表格 21 中的汉字
                LCALL   DEL2S             ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE            ;跳转回按键检测程序
;================"-1.8°"按键程序（结束）==========================
```

```
;================"+1.8°"按键程序（开始）==========================
;程序功能：显示"+1.8°"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
DU182:
                MOV     DPTR,#TAB22      ;将指针指向表格 22，显示"+1.8°"按键对应的内容。
                LCALL   DISPLAY          ;显示表格 22 中的汉字
                LCALL   DEL2S            ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE           ;跳转回按键检测程序
;================"+1.8°"按键程序（结束）==========================


;================"手动操作"按键程序（开始）==========================
;程序功能：显示"手动操作"按键对应的内容，延时 2 秒钟.跳转回按键检测程序
;
 SHOUDONG:
                MOV     DPTR,#TAB23      ;将指针指向表格 23，显示"手动操作"按键对应的内容。
                LCALL   DISPLAY          ;显示表格 23 中的汉字
                LCALL   DEL2S            ;调用延时子程序，延时 2 秒钟
                LJMP    JIANCE           ;跳转回按键检测程序
;================"手动操作"按键程序（结束）==========================


;====================液晶初始化子程序（开始）=======================
;液晶初始化子程序功能：用于在系统上电时或者必要时初始化液晶，清空屏幕显示，设置光标的移动方式，
;                       使能显示，关掉光标并且不闪烁
;                       此程序没有入口参数，直接调用即可初始化液晶。
;
;
SETLCD:
          SETBPSB           ;置高液晶的 PSB 脚
          SETBRST           ;置高液晶的 RST 脚

LGS0:     MOV     A,#34H          ;34H--扩充指令操作
      LCALL   SEND_I        ;调用向液晶写指令子程序，将指令 34H，发送给液晶。
      MOV     A,#30H        ;30H--基本指令操作
      LCALL   SEND_I        ;调用向液晶写指令子程序，将指令 30H，发送给液晶。
      MOV     A,#01H        ;清除显示
      LCALL   SEND_I        ;调用向液晶写指令子程序，将指令 01H，发送给液晶。
```

```
        LCALL   DELAY2          ;调用一个短的延时程序，给液晶反应得时间。
        MOV     A,#06H          ;06H--指定在资料写入或读取时，光标的移动方向
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 06H，发送给液晶。
        MOV     A,#0CH          ;0CH--开显示,关光标,不闪烁
        LCALL   SEND_I          ;调用向液晶写指令子程序，将指令 0CH，发送给液晶。
           RET                  ;液晶初始化子程序结束，返回。
;====================液晶初始化子程序（结束）======================

;====================液晶显示汉字和字符（开始）======================
;液晶显示汉字和字符子程序功能：在液晶上显示 4 行字，程序的入口参数为一个表格地址，表格中是要显示的字符
的 ASCII 码。
;                               直接将表格的地址赋值给 DPTR 指针，然后调用此程序即可。
;
;


DISPLAY: MOV      COUNT,#40H    ;设置要显示的字符个数，因为整个液晶屏总共可以显示 40H（64）个字符，
因此设为 40H（64）。

      MOV       A,#80H          ;设置显示的字符的起始地址，即从液晶上地址为 80H 的地放开始显示字符，80H
为液晶左上角开始的地址。
      LCALL     SEND_I          ;调用向液晶写指令子程序，将指令 80H，发送给液晶。
DISPLAY1:
      CLR A                     ;清除 A 的内容，防止对下面的程序产生影响。

           MOVC     A,@A+DPTR      ;将表格中的内容取出来并赋值给 A.为发送做准备。
      LCALL     SEND_D          ;调用向液晶发送数据子程序，将 A 中的数据（即从表格中取出的数据），发送给
液晶。
      INC  DPTR                 ;将数据指针加一，指向表格中的下一个数据，以便取出下一个数据。
      DJNZ      COUNT,DISPLAY1;判断一下是否将数据完全取完，如果取完就向下执行，如果没有就继续取数据，
COUNT 的值为需要取得字符

的个数。
      LCALL     DELAY3          ;数据已经取完，短延时一下，给予显示时间。

           RET                  ;子程序结束，程序返回。
;====================液晶显示汉字和字符（结束）======================

;====================向液晶发送数据子程序（开始）======================
;向液晶发送数据子程序功能:向液晶发送数据，入口参数为 A，将要发送的数据放入 A 中，然后调用此程序即可将
要发送的数据发送到液晶。
;
;
;
```

```
SEND_D: LCALL    CHK_BUSY        ;调用检查忙碌子程序，发送数据前需先检查液晶是否忙碌，不忙碌才可发送数据。
    SETB     RS              ;置高液晶的 RS 脚
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

    CLR RW              ;置高液晶的 RW 脚
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

    MOV      LCDDATA,A      ;将要发送的数据送到数据端口（即 P1）。
       NOP
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    SETB     E              ;置高液晶的 E 脚
    NOP
    NOP                    ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    CLR E               ;置低液晶的 E 脚
       NOP
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    NOP
    MOV      LCDDATA,#0FFH ;将数据端口全部设为高电平，防止误动作。
       NOP
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    RET                    ;子程序结束，返回。
;====================向液晶发送数据子程序（结束）======================
```

```
;====================向液晶写指令子程序（开始）======================
;向液晶写指令子程序功能:向液晶发送指令，入口参数为 A，将要发送的指令放入 A 中，然后调用此程序即可将要发送的指令发送到液晶。

SEND_I:
    LCALL    CHK_BUSY        ;调用检查忙碌子程序，发送指令前需先检查液晶是否忙碌，不忙碌才可发送指令。
    CLR RS              ;置高液晶的 RS 脚
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

    CLR RW              ;置高液晶的 RW 脚
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
       NOP                   ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。

       MOV  LCDDATA,A      ;将要发送的指令送到数据端口（即 P1）。
```

```
        NOP
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    SETB    E               ;置高液晶的 E 脚
    NOP
    NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    CLR E               ;置低液晶的 E 脚
        NOP
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    NOP
    MOV     LCDDATA,#0FFH ;将数据端口全部设为高电平，防止误动作。
        NOP
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
    RET                     ;子程序结束，返回。


;====================检测液晶忙碌子程序（开始）======================
;检测液晶忙碌子程序功能:检测液晶是否忙碌，如果液晶不忙碌，子程序返回，
;                        如果液晶忙碌，会一直在子程序中等待，直到液晶不忙碌后再返回。
CHK_BUSY:
    CLR RS              ;置低液晶的 RS 脚

        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        NOP

    SETB    RW              ;置高液晶的 RW 脚
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        NOP

    SETB    E               ;置高液晶的 E 脚
    NOP
    NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。


    JB  P1.7,$          ;判断数据端口的最高位是高还是低，
                            ;如果是高电平，说明液晶忙碌，在此等待，如果为低电平，说明液晶不忙碌，继
续向下执行。
    CLR E               ;置低液晶的 E 脚
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        NOP                     ;空指令，短暂延时，等待液晶反映，液晶反映比较慢。
        RET                     ;子程序结束，返回。

;====================检测液晶忙碌子程序（结束）======================
```

```
;短的延时子程序 1，延时很短不准确。
DELAY1:  MOV      R6,#06H
DEL11:   MOV      R7,#08H
DEL12:   DJNZ     R7,DEL12
         DJNZ     R6,DEL11
         RET


;短的延时子程序 2，延时很短不准确。
DELAY2:  MOV      R6,#0CH
DEL21:   MOV      R7,#18H
DEL22:   DJNZ     R7,DEL22
         DJNZ     R6,DEL21
         RET


;短的延时子程序 3，延时很短不准确。
DELAY3:
      MOV R6,#0DDH
DL30:
      MOV R5,#0CFH
      DJNZ R5,$
      DJNZ R6,DL30
      RET

;================================

;================================================
;======延时 200MS========
;^^^误差:超出 153 微秒^^^
DEL200MS: MOV R5,#0FEH
DL200MS0: MOV R6,#09DH
DL200MS1: MOV R7,#01H
            DJNZ R7,$
            DJNZ R6,DL200MS1
            DJNZ R5,DL200MS0
            RET

;======延时 2S========
;^^^误差:相差 1017 微秒^^^
DEL2S: MOV R5,#0FEH
DL2S0: MOV R6,#0FAH
DL2S1: MOV R7,#0DH
          DJNZ R7,$
          DJNZ R6,DL2S1
```

```
        DJNZ R5,DL2S0
        RET

;======延时 1S========
DEL1S:    ;误差  -0.000000000227us
    MOV R7,#0DH
DL1S:
    MOV R6,#0F7H
DL1S0:
    MOV R5,#8EH
    DJNZ R5,$
    DJNZ R6,DL1S0
    DJNZ R7,DL1S
    NOP
    RET



TAB1:
        DB   '      黄海学院      '
        DB   '当前状态            '
        DB   '三自由度机械手   '
        DB   '等待按键按下      '

TAB2:  DB   '      黄海学院      '
        DB   '当前状态            '
        DB   '三自由度机械手   '
        DB   '开关键设置背光   '



TAB11:
        DB   '      黄海学院      '
        DB   '当前按键            '
        DB   '三自由度机械手   '
        DB   '水平出              '

TAB12:
        DB   '      黄海学院      '
        DB   '当前按键            '
        DB   '三自由度机械手   '
        DB   '垂直伸              '
```

```
TAB13:
        DB   '     黄海学院     '
        DB   '当前按键          '
        DB   '三自由度机械手    '
        DB   '手爪开            '

TAB14:
        DB   '     黄海学院     '
        DB   '当前按键          '
        DB   '三自由度机械手    '
        DB   '旋转设置          '

TAB15:
        DB   '     黄海学院     '
        DB   '当前按键          '
        DB   '三自由度机械手    '
        DB   '水平旋转          '

TAB16:
        DB   '     黄海学院     '
        DB   '当前按键          '
        DB   '三自由度机械手    '
        DB   '自动执行          '

TAB17:
        DB   '     黄海学院     '
        DB   '当前按键          '
        DB   '三自由度机械手    '
        DB   '学习回放          '

TAB18:
        DB   '     黄海学院     '
        DB   '当前按键          '
        DB   '三自由度机械手    '
        DB   '水平回            '

TAB19:
        DB   '     黄海学院     '
        DB   '当前按键          '
        DB   '三自由度机械手    '
```

```
        DB    '垂直缩              '

TAB20:
        DB    '     黄海学院      '
        DB    '当前按键           '
        DB    '三自由度机械手     '
        DB    '手爪合             '

TAB21:
        DB    '     黄海学院      '
        DB    '当前按键           '
        DB    '三自由度机械手     '
        DB    '-1.8°              '

TAB22:
        DB    '     黄海学院      '
        DB    '当前按键           '
        DB    '三自由度机械手     '
        DB    '+1.8°              '

TAB23:
        DB    '     黄海学院      '
        DB    '当前按键           '
        DB    '三自由度机械手     '
        DB    '手动操作           '

    END
```

【范例路径】

三自由度机械手实验例程\Example\ EX16_KEY_LCD

【作业】

1. 画出手柄按键的电路图；
2. 思考：如何设置按键的信号线使检测按键比较方便编程？

实验七　手柄和主板的串口通信实验

【实验目的】

结合《单片机原理与应用》及《工业机器人设计》课程及机械手控制器的原理图，设计基于双单片机的机械手控制，实现应用双单片机对机械手的控制。通过该实验可以使同学们认知如何进行双单片机之间的通讯协作，从而完成对机械手的控制，为后续应用打下基础。

【涉及知识面】

该实验涉及《单片机原理与应用》等知识

【实验工具】

计算机一台、三自由度气动机械手一台、手柄一台

【实验要求】

1. 熟练掌握单片机主板和手柄程序下载的方法；
2. 熟练掌握单片机主板与手柄通讯的设置方法；
3. 了解串口通讯协议。

【硬件连接】

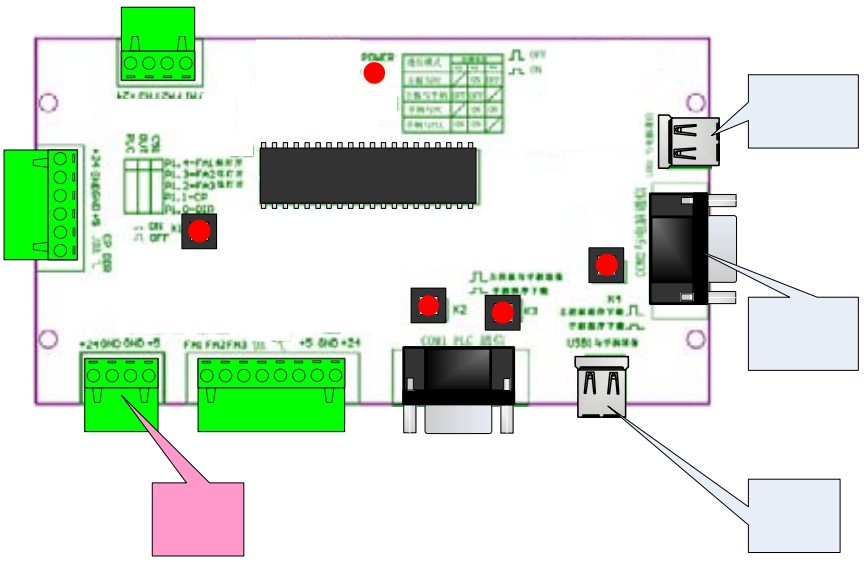


图 7-1 接线图

串口实验涉及到的接口转换较多，请同学们根据实验步骤并结合产品说明书，数量掌握。

【实验原理】

1. 矩阵式键盘的结构与工作原理

【实验步骤】

1. 将例程中的 UART10.HEX 文件下载到手柄；
2. 将例程中的 UART20.HEX 文件下载到单片机主板；
3. 将单片机主板和手柄进行连接；
5. 完成两个单片机之间的通讯。

【程序代码】

本实验程序代码有些长，请同学们到例程文件中打开，这里不再罗列。

【范例路径】

三自由度机械手实验例程\Example\EX17_UART

【作业】

1. 画出手柄按键的电路图；
2. 思考：如何设置按键的信号线使检测按键比较方便编程？

第2章 机械测绘相关实验

实验一　机械手拆装实验

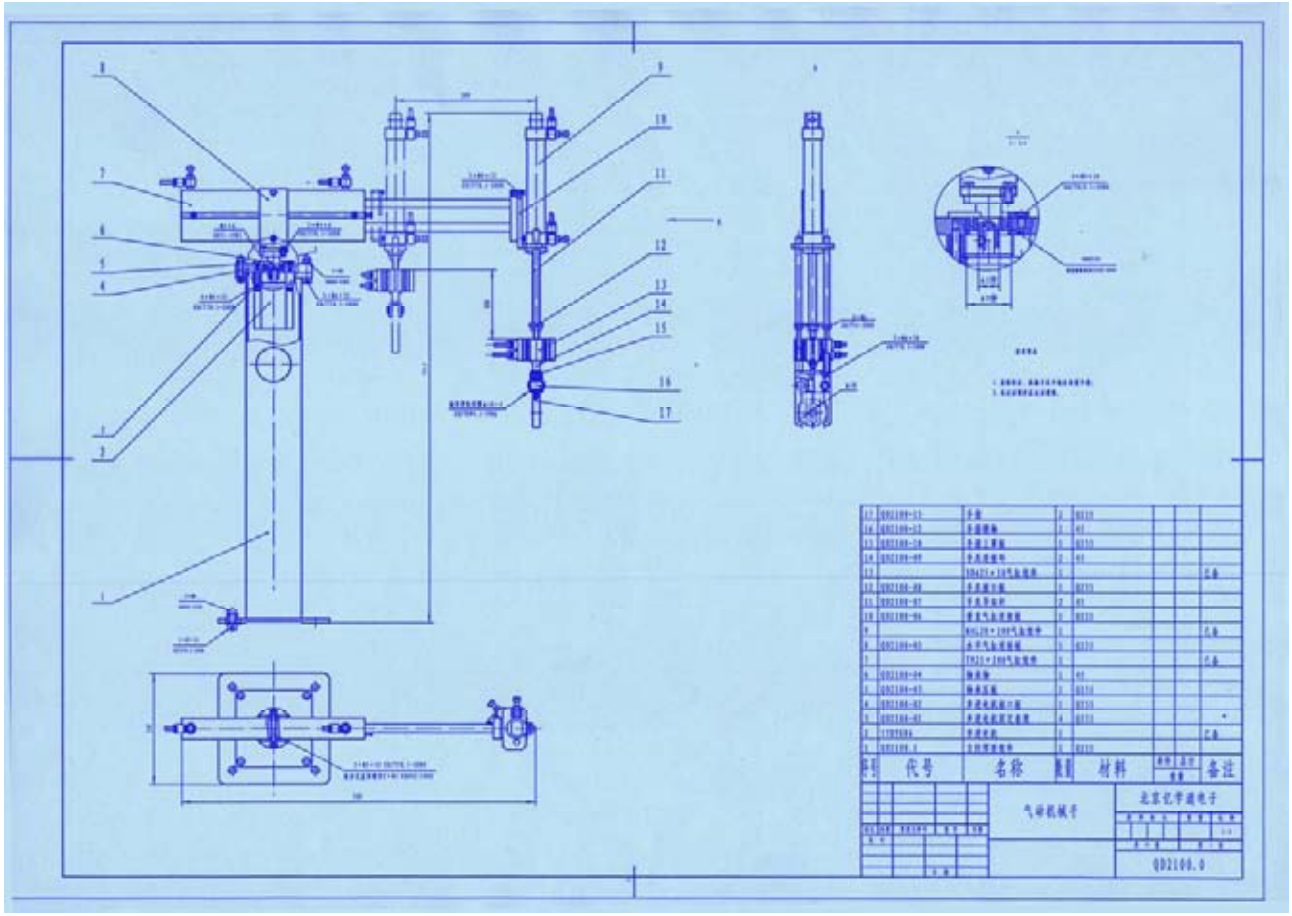


图 1-1 机械手图纸

【实验目的】

认知机械手的机械结构，学习机械手的设计方法，并在拆卸中体会结构设计对于设备装配及维护的重要性。

【涉及知识面】

该实验涉及《机械制图》、《互换性及技术测量基础》等课程的相关内容。

【使用工具】

活扳手，内六角扳手，一字形、十字形螺丝刀，橡皮锤,轴用弹性挡圈专用卡口钳等

【实验要求】

要求同学们在动手拆之前，先对机械手进行组成分析，并提出初步的拆卸顺序。经过老师审核后，找齐所需工具，学会如何应用这些工具，然后进行拆卸。拆卸完成后，要将零件归类，清洗，并对其中一些零件进行养护，养护好进行安装。安装好后，经指导老师检查，要进行运转调试。调试无误，完成实验报告。

【实验步骤】

1. 拔去所有气管及连接线(注意拔气管的正确方法);
2. 应用活扳手和内六角扳手卸去底板；
3. 应用内六角扳手将水平气缸连同安装在其上面的所有器件一起卸下；
4. 应用活扳手和内六角扳手将电机接口板及安装在上面的器件一同卸下；
5. 应用内六角扳手将电机四角的螺丝卸下；
6. 应用一字形螺丝刀将电机轴与轴承轴分开；
7. 将电机接口板翻转，使水平气缸连接板向下，将电机接口板卡在立柱口上，应用橡皮锤将轴承轴轻轻砸下；
8. 应用十字形螺丝刀将轴承盖从电机接口板上卸下；
9. 应用内六角扳手将水平气缸接口板与轴承轴分开；
10. 将电机接口板上露出全部轴承的面向下放置在立柱口上，两者中间垫上几层报纸，然后应用卸下的 M8 的内六角螺钉和橡皮锤将轴承卸下；
11. 应用轴承弹性挡圈专用卡口钳将手爪手指部分的四个弹性挡圈卸下；
12. 取出手指销轴；
13. 取出手指；
14. 应用内六角扳手将手指上罩板内的两个内六角螺钉卸下；
15. 旋转手指上罩板将其连同其内部的齿条从 SDA 气缸上卸下；
16. 用活扳手松动安装在导杆上的手爪接口板下面的螺母，然后应用活扳手卡住导杆上端的平面，将导杆从 SDA 气缸上的螺孔中拧出，注意两根导杆依次同时拧，每次一根导杆不超过 5 圈；
17. 拧下在 MAL20×100 气缸活塞杆下部的螺母，将手爪接口板卸下；
18. 卸下两根导杆；
19. 拧下气缸 MAL20×100 下部的薄六角螺母，将 MAL 气缸从垂直气缸连接板上卸下；
20. 应用内六角扳手将垂直气缸连接板将水平气缸顶端板上卸下，整个机械手拆卸完成。

【作业】

1. 思考：本机械手中应用的轴承为什么类型，为什么要选用这种类型？
2. 拆卸中出现的问题以及这些问题的解决；

实验二　机械手拆装实验

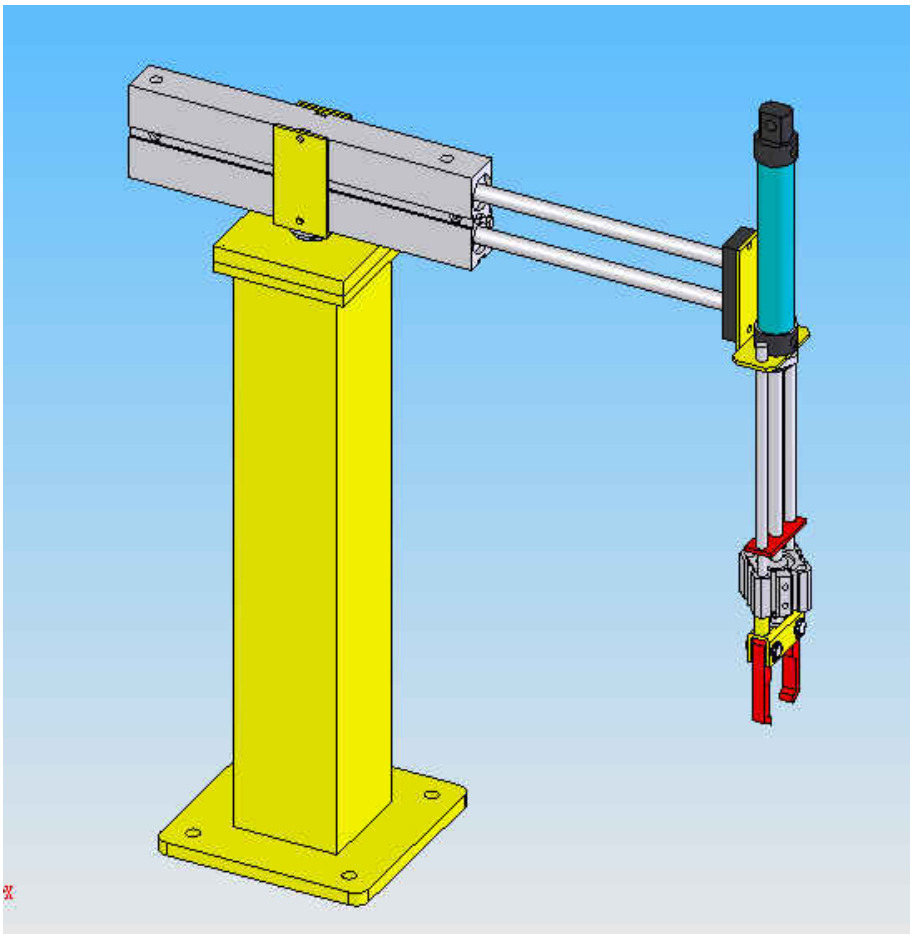


图 2-1 机械手测绘图

【实验目的】

结合学过的机械制图及互换性与技术测量基础等相关知识，对三自由度气动机械手进行测绘。测绘完成后，整理图纸，并与其他组同学的测绘结果进行比较，并弄明白不一致的原因，并将测绘图纸交指导老师批阅，着重检查零部件的标注信息。通过测绘，使同学们学会对一个设备的测绘方法。

【涉及知识面】

该实验涉及《机械制图》、《互换性与技术测量基础》等课程的相关内容。其中，主要是通过对机械手的测绘，学习如何对测绘的零部件进行正确的标注，包括尺寸、公差及零部件之间的配合关系。

【实验要求】

要求同学们在测绘之前，根据合理的拆卸顺序，对机械手进行快速拆卸。学习充分运用手中的

工具，对每个零件如何进行正确的测量，然后选择合适的表达方式，将该零件或部件表达出来。然后与其他组同学的测绘结果进行比较，并弄明白不一样的地方，判断正误，完成实验报告。

【实验步骤】

1. 按照工具表，领取工具，分组；
2. 对机械手进行快速拆卸；
3. 对拆卸下来的零部件进行清洗，擦拭，并进行分类；
4. 对零部件进行测绘；
5. 将测绘结果与其他组同学的测绘结果进行比较，找出不同，判断正误；
6. 安装机械手，交回工具，完成实验报告。

【范例路径】

三自由度机械手实验例程\Example\EX22_CAD_3D

【作业】

1. 使用 solidworks 或其他相关软件绘制三维视图，为后面的实验
2. 拆卸中出现的问题以及这些问题的解决；

实验三　机械手拆装实验

【实验目的】

结合《液压与气压传动》课程的相关知识，对机械手的各部分进行拆卸，对其中所涉及的气动元器件进行认知，画图，并连接气路。通过该实验使同学们认识该机械手中所应用到的各气动元件，学会其基本的应用方法，学习其养护知识。

【涉及知识面】

该实验涉及《液压与气动传动》课程的相关知识。主要涉及《液压与气压传动》中气泵、消声器、气压表、空气滤清器、减压阀（三联件）、电磁阀、三通、汇流板、消声塞堵、直通、减径多通、管塞、快速接头、气管、TN 型、MAL 型、SDA 型气缸等气动元器件。

【实验要求】

要求同学们利用学过的知识，辨识各个气动元件，并能正确连接，完成实验报告。

【实验步骤】

1. 对照机械手实物，认知各种气动元件，并画出其图形及符号；
2. 拆卸机械手的气动元器件部分；
3. 根据机械手连接，完成其各气动元件的连接原理图，并说明元件放置次序的原因；
4. 气路拆分与连接；
5. 对气动元器件进行养护；

安装机械手，在老师检查无误后，撰写实验报告。

【作业】

1. 按照上述步骤要求，画出各气动元件的图形符号，并用线条按真实的机械手装置进行连接；
2. 画出各气动元器件的等轴测图；
3. 写明各气动元器件的养护方法；
4. 思考：如果调节各气缸的运行速度。

第3章 综合扩展实验

实验一　基于 VB 的气动机械手动画仿真实验

【前言】

目前在虚拟设计和数控仿真程序的设计中，多采用 VC 和三维造型核心来实现对三维模型的显示和动态操作。由于 VC 较难掌握，涉及到图形渲染和操作，三维造型核心技术一般需要从国外购买，对于一般的开发应用来说因成本太高而缺乏可行性。

而 VB 作为一种完全面向对象的开发工具，在界面设计和数据库方面具有相当的优势，一般的设计人员也能够学习掌握。因此利用 VB 来实现三维模型的图形设计，将会在虚拟设计和仿真程序中发挥巨大的作用。

【实验目的】

了解 VB 编程以及 OPENGL 的操作指令代码；

结合《计算机高级语言编程》及机械手运动原理图，在计算机上应用动画，设计控制机械手的界面，使之完成机械手的信号输入与运动模拟。

【涉及知识面】

该实验涉及《计算机高级语言编程》或《Visual Basic 应用》等课程的相关知识，以及 VB 结合 OpenGL 技术，构造三维模型。

【实验工具】

计算机一台

【实验要求】

通过该实验，同学们可学习如何在计算机上应用高级语言进行与工业控制相关的界面设计，以及如何应用 OpenGL 技术，实现对机械手的动画仿真实验，为后续应用打下基础。

【实验原理】

1. 三维图形接口 OpenGL 简介

三维图形程序的开发大多是基于 OpenGL 来实现的。OpenGL 实际上是一个独立于窗口系统和操作系统的开放式三维图形标准，得到了众多计算机厂商的支持。作为一个优秀的三维图形接口，OpenGL 提供了丰富的绘图命令，利用这些命令能够开发出高性能、交互式的三维图形应用程序。

它与 VC 有着紧密的开发接口，但由于 VC 对于一般非计算机专业的工程技术人员来说难以掌握，因而给工程领域的仿真程序设计带来了很大的不便。但目前支持 VB 等开发工具的 OpenGL 开发库也开始出现。本文就是基于 VBOpenGL（vbogl.tlb）库来实现的，这是一个可免费使用的第三

方库，它封装了大量的底层 OpenGL 库函数，很大程度上简化了开发工作。

2. OpenGL 功能代码

（1）旋转操作：旋转操作是通过对模型场景视角的法向量按照一定的步幅来调整实现的。

```
diffx = Abs(SposX - X)
diffy = Abs(SposY - Y)
If diffx > diffy Then
If X > SposX Then
    ym = ym - X * 1.8  ‘逆时针旋转
  Else
    ym = ym + X *1.8  ‘顺时针旋转
  End If
End If
```

（2）平移操作 通过对模型场景的各个位置分量的调整来实现的。和旋转操作一样，首相是通过光标的当前位置和上次位置差的绝对值来判断光标的移动方向，然后根据判断结果对相应的位置分量 PosX 和 PosY 增加或减去一个指定的步幅。

```
If diffx > diffy Then
  If X > SposX Then
    PosY = PosY + 5  ‘上移
  Else
    PosY = PosY - 5  ‘下移
  End If
Else
  If Y > SposY Then
    PosX = PosX - 5  ‘左移
  Else
    PosX = PosX + 5  ‘右移
  End If
End If
End If
```

【实验步骤】

1. 对照机械手实物，编程绘制基于 VB 和 OpenGL 技术的机械手三维模型；

2. 进行计算机界面设计；

3. 对所设计的软件进行仿真调试；

4. 完成实验报告。

【三维模型设计】

1. 参数设计

表 1 参数设置表

参数	操作
01	顺时针旋转
02	逆时针旋转
03	水平出
04	水平回
05	垂直出
06	垂直回
07	手爪开
08	手爪合

2. 三维造型

三维模型可通过一般的三维造型软件 Solidworks 来设计，然后通过软件提供的图形接口输出为中性文件 STL。

在 VB 中建立三维图形环境之前，先要在“工程”菜单下通过“引用”子菜单下加入 VBOpenGL 库，然后在窗体上加入一个 PictureBox 控件作为三维模型的显示和操作区域，在 VB 环境中可通过读入 STL 文件并在绘图空间中重现的方法来显示三维模型。

对读入的 STL 文件按行分解，从中解析出每个面元，并存储到结构体数组中，接下来的工作就是通过 VBOpenGL 中的 glColor3d 函数对每个面元进行渲染和着色，并把渲染的结果以图形的形式填充到 PictureBox 控件的绘图区中，最终实现整个模型的显示，如下图所示。

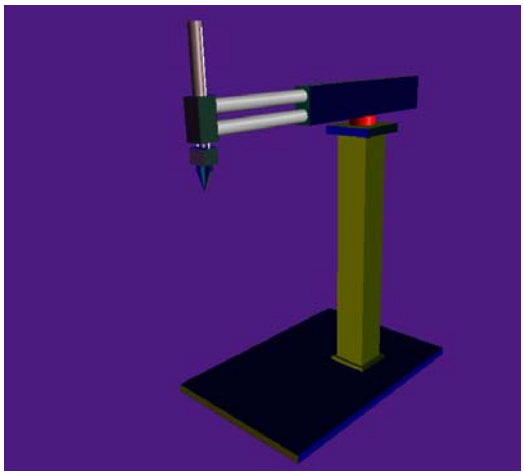


图 1-1　水平伸出

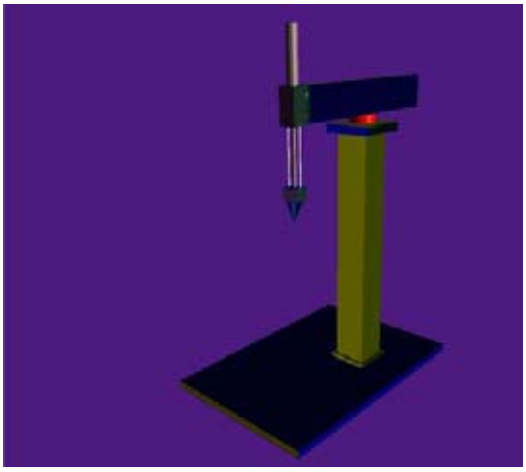


图 1-2 垂直伸出

3. 运动仿真

```
Private Sub MSComm1_OnComm()
Dim A As Variant
On Error Resume Next
Select Case MSComm1.CommEvent
Case comEvReceive
A = MSComm1.Input
Select Case A(0)
Case 1
CanShu = 3
Timer2.Enabled = True
Case 2
CanShu = 4
Timer2.Enabled = True
Case 3
CanShu = 5
Timer2.Enabled = True
Case 4
CanShu = 6
Timer2.Enabled = True
Case 5
CanShu = 7
Timer2.Enabled = True
Case 6
CanShu = 8
Timer2.Enabled = True
Case 7
If A(1) > 100 Then
CanShu = 1
```

```
JiaoDu = (A(1) - 100) * 1.8
Timer2.Enabled = True
Else
CanShu = 2
JiaoDu = (100 - A(1)) * 1.8
Timer2.Enabled = True
End If
End Select
End Select
End Sub
Private Sub Picture1_Click()
End Sub
Private Sub Timer1_Timer()
mQuadZ = mQuadZ + 0.5
rQuadX = rQuadX + 5
rQuadY = rQuadY + 5
rQuadZ = rQuadZ + 5

If mQuadZ = -7 Then
rQuadX = 20
rQuadY = -45
rQuadZ = 3
Timer1.Enabled = False
End If
End Sub

Private Sub Timer2_Timer()
Select Case CanShu
Case 1
If JiaoDu <> 0 Then
```

```
XuanZhuan = XuanZhuan - 1
JiaoDu = JiaoDu - 1
Else
Timer2.Enabled = False
End If
Case 2
If JiaoDu <> 0 Then
XuanZhuan = XuanZhuan + 1
JiaoDu = JiaoDu - 1
Else
Timer2.Enabled = False
End If
Case 3
ShenChu = ShenChu + 0.05
If ShenChu > 1.2 Then
ShenChu = 1.2
Timer2.Enabled = False
End If
Case 4
ShenChu = ShenChu - 0.05
If ShenChu <= 0 Then
ShenChu = 0
Timer2.Enabled = False
End If
Case 5
ShenSuo = ShenSuo + 0.02
If ShenSuo > 0.6 Then
ShenSuo = 0.6
Timer2.Enabled = False
End If
```

```
Case 6
ShenSuo = ShenSuo - 0.02
If ShenSuo < 0 Then
ShenSuo = 0
Timer2.Enabled = False
End If
Case 7
ShouZhua = ShouZhua + 3
If ShouZhua >= 30 Then
ShouZhua = 30
Timer2.Enabled = False
End If
Case 8
ShouZhua = ShouZhua - 3
If ShouZhua <= 0 Then
ShouZhua = 0
Timer2.Enabled = False
End If
End Select
End Sub
```

【范例路径】

三自由度机械手实验例程\Example\EX31_VB

【作业】

1. 按照上述步骤要求，画出机械手运动的原理图；
2. 了解 OpenGL 编程原理及步骤，构造三维实体模型；
3. 按照机械手的功能要求，在计算机界面上设定驱动数据输入口；
4. 编程，实现输入数据与机械手运动的联动。
5. 思考：如何使显示的三维模型更逼真。

实验二　基于 VB 的气动机械手计算机控制实验

【实验目的】

通过本实验，使同学们学会如何实现计算机对外部单片机的控制；

结合《单片机原理与应用》课程及《计算机高级语言编程》课程的相关内容，应用虚拟 USB 接口或串行口，通过编程实现单片机与计算机的通讯，并进行双方的数据驱动，为后续应用打下基础。

【涉及知识面】

该实验涉及《单片机原理与应用》的串行口通讯与《高级计算机编程》中的 Mscomm 控件的应用，并涉及到计算机高级语言编程中的 OpenGL 技术，以及计算机和单片机各自的通讯原理和之间的相互通信技术。

【实验工具】

计算机、三自由度气动机械手、气源

【实验要求】

要求同学们利用学过的知识，将计算机与机械手的控制板的串行口或虚拟 USB 口进行连接，在计算机上设计控制界面(采用 VB 和 OpenGL 三维图形技术进行编程)，完成对机械手的控制。

【硬件连接】

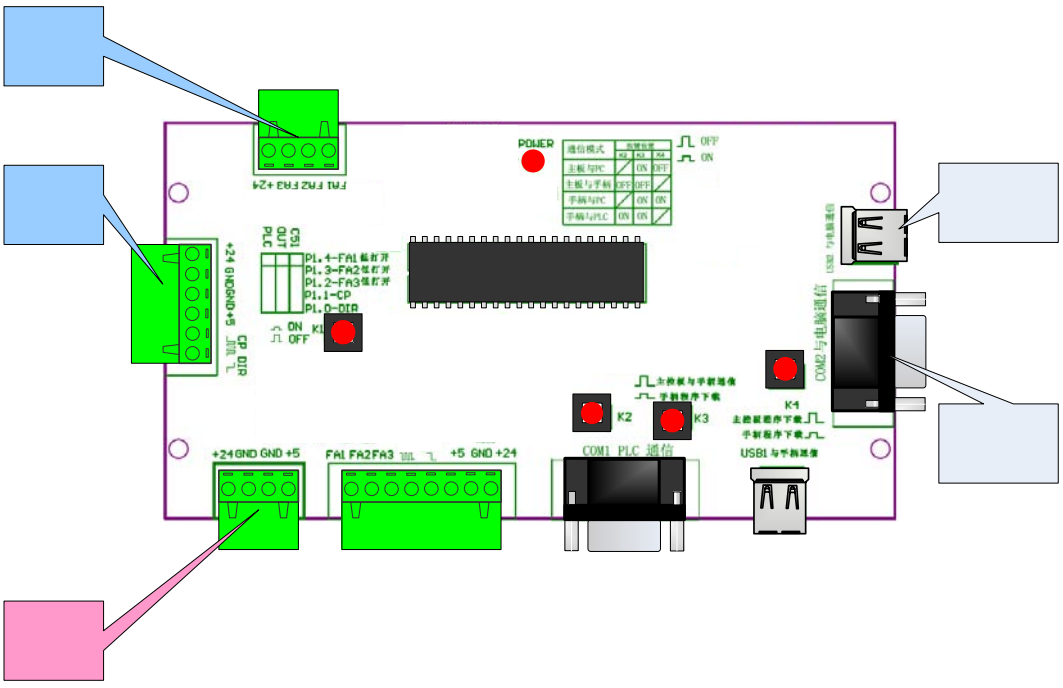


图 2-1 计算机与机械手硬件连接图

通过主板 USB2 或串口 COM2 口，连接计算机与机械手；

根据主板上方的提示，将按键 K3 按下置于“ON”、按键 K4 弹起置于“OFF”；使主板上的单片机与计算机进行通讯。

【实验步骤】

1. 应用串口线或 USB 口线，将机械手控制板与计算机进行连接；
2. 编写基于 VB 的机械手控制界面；
3. 双机通讯，实现计算机上驱动界面的数据对机械手控制板的驱动；
4. 编写程序进行调试，完成计算机对机械手的运动控制，同时实现虚拟机械手的同步运动；
5. 撰写实验报告。

【范例路径】

三自由度机械手实验例程\Example\EX31_VB

操作说明：

1. 在范例中，双击  打开 VB 程序；
2. 在软件左上角的菜单“串口->串口设置”中，选择计算机和单片机主板连接所使用的 COM 口（方法和程序下载 COM 口选择一样）；
3. 确定机械手和计算机之间、单片机主板和电磁阀/步进电机之间连线正确；
4. VB 程序界面中的按钮即可控制三自由度机械手完成：水平出、水平回、垂直出、垂直回、手抓开、手抓合等动作。同时软件中的三维机械手也做同步运动。

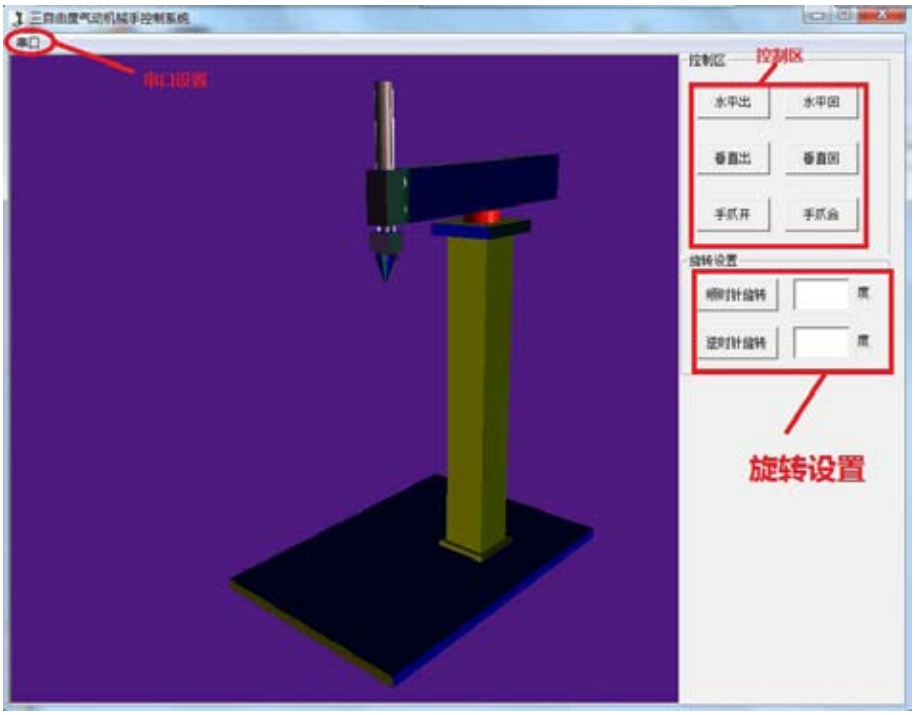


图 2-2 VB 软件界面

【作业】

1. 按照上述步骤要求，画出机械手运动的原理图；
2. 了解 OpenGL 编程原理及步骤，构造三维实体模型；
3. 按照机械手的功能要求，在计算机界面上设定驱动数据输入口；
4. 思考：如何将手柄和计算机进行连接，使用手柄对计算机 VB 虚拟机械手进行控制？

实验三　机械手搬运实验

【实验目的】

1. 利用所学知识，较系统的完成一次机械手运动控制；

【涉及知识面】

该实验涉及《单片机原理与应用》、《气压传动》等知识

【实验工具】

计算机一台、三自由度气动机械手一台、气泵。

【实验要求】

1. 机械手将较远的 A 点，搬运物体到反方向的 B 点。

【硬件连接】

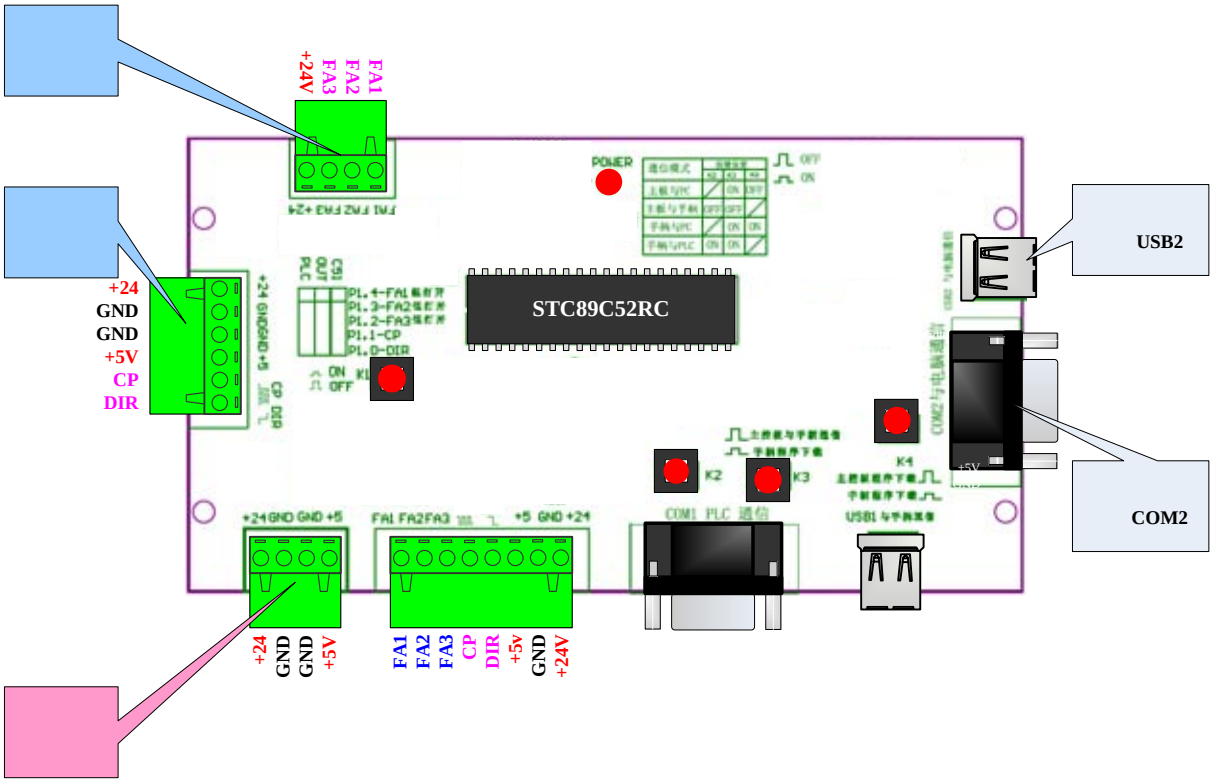


图 3-1 系统接线图

将单片机主板连接到计算机；将按键 K3 按下置于“ON”、按键 K4 弹起置于“OFF”；

【过程分析】

机械手水平前伸——>垂直下降——>打开手爪——>闭合手爪、抓起物体

垂直上升——>水平收缩——>水平转动——>停止转动

水平前伸——>垂直下降——>打开手爪、放下物体

【实验步骤】

1. 分解要完成的功能；
2. 绘制程序流程图；
3. 编写程序；
4. 编译下载程序，没有问题完成任务

【程序代码】

```
;=============================================================
;  文件名称：      ZHU.ASM
;  功能描述:      单片机编程实现机械手从 A 点搬运物体到 B 点。
;                 参考外围接线：FA1 — P1.4
;                               FA2 — P1.3
;                               FA3 — P1.2
;                                CP — P1.1
;                               DIR— P1.0
;  文件来源：     《三自由度气动机械手实验指导书》第三章
;  IDE 环境：     KEIL C51
;  涉及的库：      无
;  组成文件:      ZHU.ASM
;  日期:          2010-3-25
;=============================================================
;================定义电磁阀和步进电机单片机控制接口================


      FA1 EQU P1.4       ;定义阀 1 控制接口，低电平时电磁阀打开。
      FA2 EQU P1.3       ;定义阀 2 控制接口，低电平时电磁阀打开。
      FA3 EQU P1.2       ;定义阀 3 控制接口，低电平时电磁阀打开。
      CP  EQU P1.1       ;定义步进电机脉冲控制接口
      DIR EQU P1.0       ;定义步进电机方向控制接口


    XIFENSHU EQU 22H        ;定义细分数存储单元


       ORG     0000h
       AJMP    MAIN
       ORG     0030h
;=================================================
;主程序功能：要求水平气缸伸出到最远端，竖直气缸伸到最低端，模拟手抓夹取工件后,
;            竖直气缸缩回最高端，水平气缸缩回最近端，后旋转一定角度，然后水平
```

```
;                气缸再伸到最远端，竖直气缸伸到最低端，模拟手抓放下工件，循环。
;=================================================
MAIN:     MOV     XIFENSHU,#8;细分数设置，此处为 8 细分，与步进电机驱动器设置有关。

          MOV     P1,#0FFH     ;P1 口赋初值,防止上电初始打开电磁阀

          MOV    R4,#18        ;设置步进电机运转的固定步距角个数，可以控制旋转角度，
                               ;旋转角度=(R4 的值+2)=固定步距角，加 2 是因为会有 2 个启动脉冲。
                               ;以步距角为 1.8°的电机为例，R4 设为 18，则它会旋转(18+2)=1.8°
=36°。


          LCALL SHEN           ;调用水平气缸伸出子程序
          LCALL DEL1S          ;调用延时 1 秒子程序等待水平气缸伸到最远端
          LCALL XIA            ;调用竖直气缸向下伸出子程序
          LCALL DEL1S          ;调用延时 1 秒子程序等待垂直气缸伸到最低端
          LCALL KAI            ;调用手抓气缸打开子程序
          LCALL DEL1S          ;调用延时 1 秒子程序等待手抓打开
          LCALL HE             ;调用调用手抓闭合子程序
          LCALL DEL1S          ;调用延时 1 秒子程序等待手抓闭合
          LCALL SHANG          ;调用竖直气缸向上缩回子程序
          LCALL DEL1S          ;调用延时 1 秒子程序等待竖直气缸到达最高端
          LCALL SUO            ;调用水平气缸缩回最近端子程序
          LCALL DEL1S          ;调用延时 1 秒子程序等待水平气缸到达最近端
          SETB    DIR          ;将步进电机方向信号置位使步进电机顺时针/逆时针旋转
                               ; （注：旋转方向与步进电机两相线圈接线有关）
;=================================================
;设置低频率启动脉冲，脉冲周期 12 毫秒，发送两个脉冲用于低速启动电机，这是计算旋转角度时
需要加 2 的原因。
          MOV    R1,XIFENSHU ;将细分数计算在内。
ZUO1:
          CLR     CP           ;CP 端口输出低电平
          LCALL DEL7MS          ;低电平保持 7 毫秒
          SETB    CP           ;CP 端口输出高电平
          LCALL DEL5MS          ;高电平保持 5 毫秒
          CLR     CP           ;CP 端口输出低电平
          LCALL DEL7MS          ;低电平保持 7 毫秒
          SETB    CP           ;CP 端口输出高电平
          LCALL DEL5MS          ;高电平保持 5 毫秒
          DJNZ    R1,ZUO1      ;将细分数计算在内。
;=================================================
 JXZUO:
          MOV    R1,XIFENSHU ;将细分数计算在内。
JXZUO1:
 ;=================================================
```

```
;设置步进电机运行脉冲：脉冲周期 5 毫秒，正常运转
        CLR   CP            ;CP 端口输出低电平
        LCALL DEL3MS        ;低电平保持 3 毫秒
        SETB  CP            ;CP 端口输出高电平
        LCALL DEL2MS        ;高电平保持 2 毫秒
        DJNZ  R1,JXZUO1     ;将细分数计算在内。
;=================================================
        DJNZ  R4,JXZUO      ;步进电机运行脉冲数，R4 中值减 1 不为零则发一个脉冲，R4 中 198，
加上前面 2 个，共 200 个。


        LCALL DEL1S         ;调用延时 1 秒子程序，旋转到位后等待 1 秒
        LCALL SHEN          ;调用水平气缸伸出子程序
        LCALL DEL1S         ;调用延时 1 秒子程序等待水平气缸伸到最远端
        LCALL XIA           ;调用竖直气缸向下伸出子程序
        LCALL DEL1S         ;调用延时 1 秒子程序等待垂直气缸伸到最低端
        LCALL KAI           ;调用手抓气缸打开子程序
        LCALL DEL1S         ;调用延时 1 秒子程序等待手抓打开
        LCALL HE            ;调用调用手抓闭合子程序
        LCALL DEL1S         ;调用延时 1 秒子程序等待手抓闭合
        LCALL SHANG         ;调用竖直气缸向上缩回子程序
        LCALL DEL1S         ;调用延时 1 秒子程序等待竖直气缸到达最高端
        LCALL SUO           ;调用水平气缸缩回最近端子程序
        LCALL DEL1S         ;调用延时 1 秒子程序等待水平气缸到达最近端
        MOV   R4,#198       ;设置步进电机运转脉冲数
        CLR   DIR           ;将步进电机方向信号置位使步进电机逆时针/顺时针旋转
                            ; （注：方向与步进电机两相线圈接线有关）
;=================================================
;设置低频率启动脉冲，脉冲周期 12 毫秒，发送两个脉冲用于低速启动电机，这是计算旋转角度时
需要加 2 的原因。
        MOV   R1,XIFENSHU ;将细分数计算在内。
YOU1:
        CLR   CP            ;CP 端口输出低电平
        LCALL DEL7MS        ;低电平保持 7 毫秒
        SETB  CP            ;CP 端口输出高电平
        LCALL DEL5MS        ;高电平保持 5 毫秒
        CLR   CP            ;CP 端口输出低电平
        LCALL DEL7MS        ;低电平保持 7 毫秒
        SETB  CP            ;CP 端口输出高电平
        LCALL DEL5MS        ;高电平保持 5 毫秒
        DJNZ  R1,YOU1       ;将细分数计算在内。
;=================================================
   JXYOU:
        MOV   R1,XIFENSHU   ;将细分数计算在内。
  JXYOU1:
```

```
;=================================================
;设置步进电机运行脉冲：脉冲周期 5 毫秒，占空比为 3：2，正常运转。
        CLR   CP            ;CP 端口输出低电平
        LCALL DEL3MS        ;低电平保持 3 毫秒
        SETB  CP            ;CP 端口输出高电平
        LCALL DEL2MS        ;高电平保持 2 毫秒
        DJNZ  R1,JXYOU1     ;将细分数计算在内。
;=================================================
        DJNZ  R4,JXYOU      ;步进电机运行脉冲数，R4 中值减 1 不为零则发一个脉冲，R4 中 198，
加上前面 2 个，共 200 个。


        LJMP  MAIN
;=================================================
;水平气缸伸出子程序，手爪水平伸出。
SHEN:     CLR   FA1
          LCALL DEL2S

          RET
;=================================================
;;水平气缸缩回子程序，手爪水平缩回。
SUO:      SETB  FA1
          LCALL DEL2S

          RET

;=================================================
;竖直气缸缩回子程序，手爪上升。
SHANG:    SETB  FA2
          LCALL DEL2S

           RET

;=================================================
;;竖直气缸伸出子程序，手爪下降。
XIA:      CLR   FA2
          LCALL DEL2S

          RET

;=================================================
;;手爪气缸缩回子程序，手爪闭合。
HE:       SETB  FA3
          LCALL DEL2S
```

```
            RET

;=================================================
;;手爪气缸缩回子程序，手爪张开。
KAI:        CLR   FA3
            LCALL DEL2S

            RET



;======延时 1S========
;^^^误差:相差 3 微秒^^^
DEL1S: MOV R5,#0A7H
DL1S0: MOV R6,#0ABH
DL1S1: MOV R7,#010H
       DJNZ R7,$
       DJNZ R6,DL1S1
       DJNZ R5,DL1S0
       RET
;======延时 2S========
;^^^误差:超出 222 微秒^^^
;======延时 2S========
;^^^误差:超出 731 微秒^^^
DEL2S: MOV R5,#0FFH
DL2S0: MOV R6,#0FDH
DL2S1: MOV R7,#02H
       DJNZ R7,$
       DJNZ R6,DL2S1
       DJNZ R5,DL2S0
       RET
;======延时 7MS========
;^^^误差:相差 1 微秒^^^
 DEL7MS:
    MOV R6,#0D0H
DL7MS0:
    MOV R5,#0EH
    DJNZ R5,$
    DJNZ R6,DL7MS0
    RET
 ;======延时 3MS========
;^^^误差:相差 1 微秒^^^
 DEL3MS:    ;误差  -0.868055555556us
    MOV R6,#0FBH
```

```
DL3MS0:
    MOV R5,#04H
    DJNZ R5,$
    DJNZ R6,DL3MS0
    RET

;======延时 2MS========
;^^^误差:相差 1 微秒^^^
DEL2MS:
    MOV R6,#50H
DL2MS0:
    MOV R5,#0AH
    DJNZ R5,$
    DJNZ R6,DL2MS0
    RET
;======延时 5MS========
;^^^误差:相差 1 微秒^^^
DEL5MS: MOV R5,#0EEH
DL5MS0: MOV R6,#09H
        DJNZ R6,$
        DJNZ R5,DL5MS0
        RET

         End
```

【范例路径】

三自由度机械手实验例程\Example\EX33_A2B

【作业】

1. 改变机械手从 A 点搬运到 B 点所用时间；
2. 将程序改写为 C 语言。