

案例解说

虚拟仪器典型控制应用

六种典型控制系统，贴近工程实践

- 单片机 ●PLC ●PCI数据采集卡
- USB数据采集板 ●远程I/O模块 ●智能仪器



配套光盘 超值配套光盘，内含

- ◆ 完整案例程序——提供所有控制系统的完整源程序，并经编译和实际测试。
- ◆ 程序运行录屏——录制程序的运行过程，便于读者了解程序操作使用方法。
- ◆ 系统测试录像——录制、讲解系统的硬件组成，以及软件、硬件测试过程。
- ◆ 丰富学习资源——提供各公司硬件驱动程序、手册，以及图片等丰富资源。

李江全
李玲
刘媛媛
等编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

案例解说控制应用精品丛书

案例解说虚拟仪器 典型控制应用

李江全 李 玲 刘媛媛 等编著

電子工業出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从工程应用的角度出发,通过6种典型的计算机控制系统(包括基于单片机、PLC、PCI数据采集卡、USB数据采集板、远程I/O模块、智能仪器),使用目前流行的虚拟仪器编程语言LabVIEW和LabWindows/CVI,对工业控制系统中的4类典型应用,即模拟量输入(AI)、模拟量输出(AO)、数字量输入(DI)和数字量输出(DO)的程序设计方法进行了详细的讲解。

本书内容丰富,每个案例提供具体的设计任务、详细的操作步骤、完整的程序代码,注重解决工程实际问题。可供各类自动化、计算机应用、机电一体化等专业的大学生、研究生学习计算机控制技术,也可供计算机控制系统研发的工程技术人员参考。

为方便读者学习,本书提供超值配套光盘,内容包括所有案例的源程序、软硬件资源、程序运行录屏、系统测试录像等。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

案例解说虚拟仪器典型控制应用/李江全等编著. —北京:电子工业出版社,2011.5

(案例解说控制应用精品丛书)

ISBN 978-7-121-13406-7

I. ①案… II. ①李… III. ①计算机控制系统 IV. ①TP273

中国版本图书馆CIP数据核字(2011)第074974号

责任编辑:陈韦凯 特约编辑:蒲 玥

印 刷:涿州市京南印刷厂

装 订:涿州市桃园装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本:787×1092 1/16 印张:23.25 字数:592千字

印 次:2011年5月第1次印刷

印 数:4000册 定价:49.00元(含光盘1张)



凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

计算机控制技术在通信、遥感、无损检测、智能仪器、工业自动控制等工程领域有着广泛的应用。在开发计算机控制系统时,程序设计是很多技术人员要面临的问题。在高校和科研院所,有众多的研究人员在使用各种计算机控制系统,他们都迫切需要相关的书籍来帮助他们学习相关的编程技术。

本书从工程应用的角度出发,通过几种典型的计算机控制系统(包括基于单片机、PLC、PCI 数据采集卡、USB 数据采集板、远程 I/O 模块、智能仪器等),使用目前流行的虚拟仪器编程语言 LabVIEW 和 LabWindows/CVI,对工业控制系统中的 4 类典型应用,即模拟量输入(AI)、模拟量输出(AO)、数字量输入(DI)和数字量输出(DO)的程序设计方法进行了详细的讲解。

LabVIEW 采用了工程人员熟悉的术语、图标等图形化符号来代替常规基于文字的语言程序。它把复杂、烦琐、费时的语言编程简化成用菜单或图标提示方法来完成某些功能的选择功能图标,并提供用线条把各种功能图标连接起来的简单图形编程方式。利用 LabVIEW,用户可通过定义和连接代表各种功能模块的图标,方便迅速地创建虚拟仪器。

LabWindows/CVI 是 32 位的面向计算机测控领域的虚拟仪器软件开发平台,它是以 ANSI C 为核心的交互式虚拟仪器开发环境,它将功能强大的 C 语言与测控技术有机结合,具有灵活的交互式编程方法和丰富的库函数,为开发人员建立检测系统、自动测试环境、数据采集系统、过程监控系统等提供了理想的软件开发环境。

书中提供的测控应用实例都有详细的操作步骤,读者可以在实验室动手操作,按步骤去实现虚拟仪器的各种测控功能。实践操作性强是本书的特色。

本书弥补了虚拟仪器设计同类书籍在测控实践方面的缺憾,因此,对 LabVIEW 和 LabWindows/CVI 在测控领域的学习者有很好的参考价值。

为方便读者学习,本书提供超值配套光盘,内容包括所有案例的源程序、软硬件资源、程序运行录屏、系统测试录像等。

本书由塔里木大学李平编写第 1 章,刘媛媛编写第 3、4 章;石河子大学龚立娇编写第 2 章,李江全编写第 5 章及附录,李玲编写第 6 章;克拉玛依职业技术学院柳力文编写第 7、8 章;全书由李江全教授担任主编并统稿,李玲、刘媛媛担任副主编。参与编写、程序设计等工作的人员还有田敏、李宏伟、郑瑶、郑重、朱东芹、任玲、王洪坤、汤智辉、胡蓉、王平等老师。电子开发网、北京研华科技、西安达泰电子、石河子大学电气工程实验中心等单位或公司为本书的编写提供了宝贵的技术支持和帮助,借此机会对他们致以深深的谢意。

由于编者水平有限,书中难免存在不妥或错误之处,恳请广大读者批评指正。

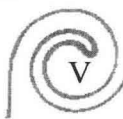
编著者

2011 年 2 月



目 录

第 1 章 基于单片机开发板的控制应用	1
1.1 系统设计说明	1
1.1.1 设计任务	1
1.1.2 硬件系统	2
1.2 数据采集与控制程序设计	4
1.2.1 模拟量输入	4
1.2.2 模拟量输出	25
1.2.3 数字量输入	40
1.2.4 数字量输出	52
第 2 章 基于 PLC 的控制应用	61
2.1 系统设计说明	61
2.1.1 设计任务	61
2.1.2 硬件系统	62
2.2 数据采集与控制程序设计	70
2.2.1 模拟量输入	70
2.2.2 模拟量输出	86
2.2.3 数字量输入	95
2.2.4 数字量输出	115
第 3 章 基于 NI 数据采集卡的控制应用	135
3.1 系统设计说明	135
3.1.1 设计任务	135
3.1.2 硬件系统	136
3.2 数据采集与控制程序设计	144
3.2.1 模拟量输入	144
3.2.2 数字量输入	164
3.2.3 数字量输出	171
第 4 章 基于研华数据采集卡的控制应用	179
4.1 系统设计说明	179
4.1.1 设计任务	179
4.1.2 硬件系统	180
4.2 数据采集与控制程序设计	185





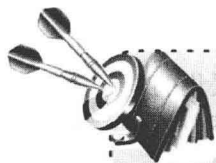
4.2.1	模拟量输入	185
4.2.2	模拟量输出	199
4.2.3	数字量输入	210
4.2.4	数字量输出	220
第 5 章	基于 USB 数据采集板的控制应用	231
5.1	系统设计说明	231
5.1.1	设计任务	231
5.1.2	硬件系统	232
5.2	数据采集与控制应用程序设计	238
5.2.1	模拟量输入	238
5.2.2	模拟量输出	242
5.2.3	数字量输入	245
5.2.4	数字量输出	249
第 6 章	基于远程 I/O 模块的控制应用	253
6.1	系统设计说明	253
6.1.1	设计任务	253
6.1.2	硬件系统	254
6.2	数据采集与控制应用程序设计	265
6.2.1	模拟量输入	265
6.2.2	模拟量输出	272
6.2.3	数字量输入	278
6.2.4	数字量输出	286
第 7 章	基于智能仪器的控制应用	294
7.1	系统设计说明	294
7.1.1	设计任务	294
7.1.2	硬件系统	294
7.2	温度测量与控制程序设计	300
7.2.1	单台智能仪器温度测控	300
7.2.2	多台智能仪器温度测控	310
7.2.3	网络温度监控	322
第 8 章	基于声卡的数据采集	327
8.1	声卡的基本常识	327
8.1.1	声卡的作用与特点	327
8.1.2	声卡的构造与设置	328
8.1.3	声卡的主要技术参数	330
8.2	基于声卡的数据采集程序设计	331



8.2.1 利用 LabVIEW 实现声卡数据采集.....	331
8.2.2 利用 LabWindows/CVI 实现声卡数据采集.....	339
附录 A 控制系统的输入与输出.....	345
附录 B LabWindows/CVI 串口通信函数.....	349
附录 C LabWindows/CVI 数据采集函数库.....	353
参考文献.....	362



第1章 基于单片机开发板的控制应用



目前,在许多单片机应用系统中,上、下位机分工明确,作为下位机核心器件的单片机往往只负责数据的采集和通信,而上位机(一般为个人计算机,简称PC)通常以基于图形界面的Windows系统为操作平台。为便于查询和保存数据,还需要数据库的支持。这种应用的核心是数据通信,它包括单片机和上位机之间,客户端和服务端之间,以及客户端和客户端之间的通信,而单片机和上位机之间的数据通信则是整个系统的基础。

单片机和PC的通信是通过单片机的串口和PC串口之间的硬件连接实现的。

1.1 系统设计说明

1.1.1 设计任务

分别利用Keil C51、汇编语言编写程序实现单片机数据采集与控制;分别利用LabVIEW和LabWindows/CVI编写程序实现PC与单片机自动化控制。任务要求如下。

1. 模拟电压输入

单片机开发板接收变化的模拟电压(范围:0~5V)并在数码管上显示(保留1位小数);PC接收单片机发送的电压值(十六进制,1个字节),转换成十进制形式,以数字、曲线的方式显示。

2. 模拟电压输出

在PC程序界面中输入一个数值(范围:0~10),发送到单片机开发板,在数码管上显示(保留1位小数),并通过模拟电压输出端口输出同样大小的电压值。

3. 数字量输入

将单片机开发板数字量输入端口与地短接或断开，产生数字信号 0 或 1 送到单片机数码管上显示，单片机再将数字信号发送到 PC 显示。

4. 数字量输出

PC 发出开关指令（0 或 1）传送给单片机开发板，驱动相应的继电器动作。

单片机与 PC 通信，在程序设计上涉及两个部分的内容：一是单片机的 C51 数据采集和控制程序；二是 PC 的串口通信程序和各种功能程序。

1.1.2 硬件系统

1. 线路连接

基于单片机开发板的数据采集与控制系统结构如图 1-1 所示。

工作过程为：作为下位机的单片机实时采集测量到的电压值，并将采集的电压数据显示在数码管上，同时采集的电压值通过串口传送到上位 PC。上位机收到下位机传送来的电压数据，在显示屏上显示。上位 PC 设置电压值，通过串口发送到单片机系统，单片机数码管显示该电压，并通过模拟电压输出端口输出。上位 PC 发出开关指令传送给单片机系统，驱动继电器动作。电气开关产生开关信号，发送到 PC 显示。

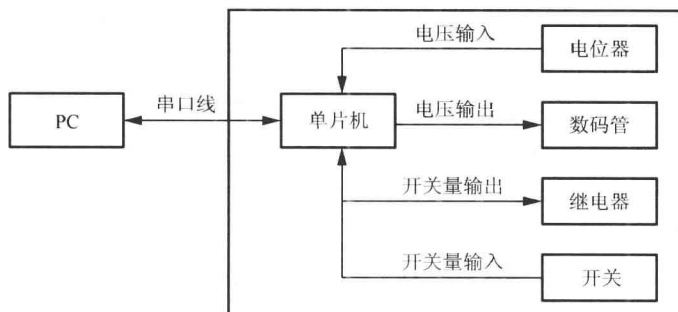


图 1-1 PC 与单片机组成的数据采集与控制系统框图

如图 1-2 所示，单片机开发板与 PC 数据通信采用 3 线制，将单片机开发板 B 的串口与 PC 串口的 3 个引脚（RXD、TXD、GND）分别连在一起，即将 PC 和单片机的发送数据线 TXD 与接收数据 RXD 交叉连接，两者的地线 GND 直接相连。

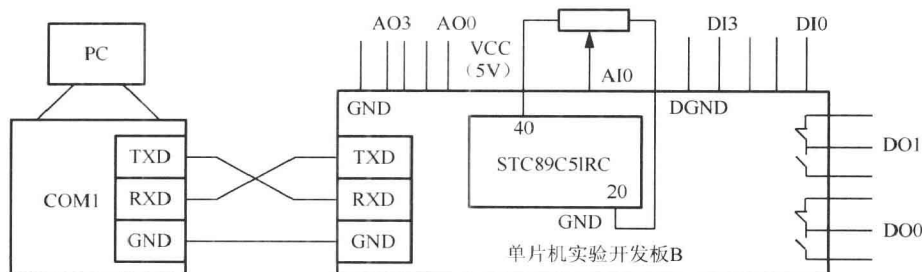


图 1-2 PC 与单片机开发板 B 组成数据采集与控制系统



由于单片机的 TTL 逻辑电平和 RS-232C 的电气特性完全不同, RS-232C 的逻辑 0 电平规定在+3~+15V 之间, 逻辑 1 电平在-3~-15V 之间, 因此, 在将 PC 和单片机的 RXD 和 TXD 交叉连接时必须进行电平转换, 单片机开发板 B 使用的是 MAX232 电平转换芯片。

模拟电压输入: 直接采用单片机的 5V 电压输出 (40 和 20 引脚)。将电位器两端与 STC89C51RC 单片机的 40 和 20 引脚相连, 电位器的中间端点 (输出电压 0~5V) 与单片机开发板 B 的模拟量输入口 AI0 相连。

模拟电压输出: 不需连线。使用万用表直接测量单片机开发板 B 的 AO0, AO1, AO2, AO3 端口与 GND 端口之间的输出电压。

数字量输入: 使用杜邦线将单片机开发板 B 的 DI0, DI1, DI2, DI3 端口与 DGND 端口连接或断开即可。

数字量输出: 不需连线, 直接使用单片机开发板 B 的继电器和指示灯。

2. 单片机开发板 B 简介

单片机开发板 B 是电子开发网专为单片机初学者设计并开发的一种实验兼开发板, 开发这个产品的目的就是帮助单片机初学者快速学会单片机技术。在自学单片机的过程中, 通过做一系列的实验, 从而比较容易地领会单片机那些枯燥、难懂的专业术语, 而且这款实验开发板弥补了市场上常见的单片机实验板的一些不足, 有针对性地面向最终的实用控制功能, 包括模拟量输入与输出接口、数字量输入与输出接口, 增加了实用的继电器接口, 可以使实验板能够直接用于控制各种负载, 成为一个实用化的嵌入式控制系统。

图 1-3 是单片机开发板 B 的实物图。

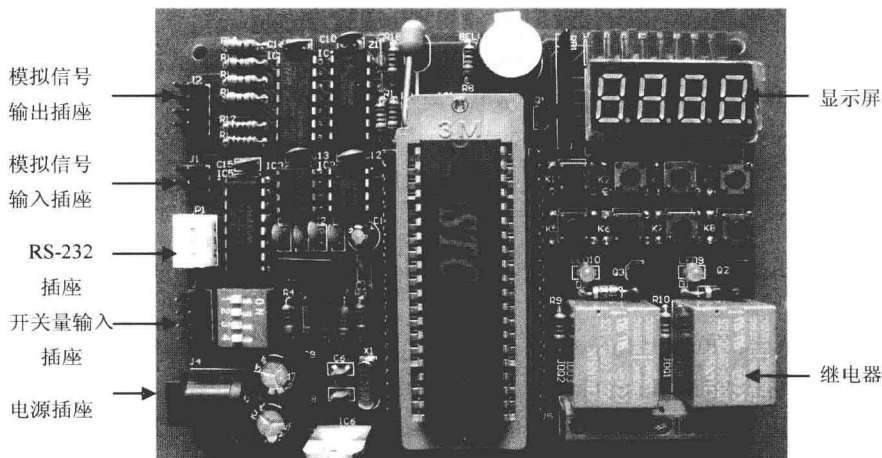


图 1-3 单片机开发板 B 实物图

单片机开发板 B 可以做很多实验, 如: 模拟电压输入与输出、开关量输入与输出、红外线遥控器编码分析仪、通用频率计、温度测控等。

有关单片机开发板 B 的详细信息请查询电子开发网 <http://www.dzkfw.com>。



1.2 数据采集与控制程序设计

1.2.1 模拟量输入

1.2.1.1 利用 Keil C51 实现单片机模拟电压输入

Keil C51 软件是众多单片机应用开发的优秀软件之一，它集编辑、编译、仿真于一体，支持汇编、PLM 语言和 C 语言的程序设计，界面友好，易学易用。

启动 Keil C51，几秒后出现编辑界面。

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，出现 Create New Project 对话框。然后选择你要保存的路径、文件夹，输入工程文件的名字，如 pc_com（后缀名默认），单击“保存”按钮。

这时会弹出一个 Select Device for Target “Target 1”对话框，要求选择单片机的型号，可以根据用户使用的单片机来选择，Keil C51 几乎支持所有的 51 核的单片机。这里选择 Atmel 的 89C51。选择 89C51 之后，右边一栏是对这个单片机的基本说明，然后单击“确定”按钮。

2. 编写程序

单击 File 菜单，在下拉菜单中单击 New 选项。此时光标在编辑窗口里闪烁，这时可以输入用户的应用程序了，但笔者建议首先保存该空白的文件。

单击菜单上的 File，在下拉菜单中选中 Save As 选项，在“文件名”栏右侧的编辑框中，输入欲使用的文件名，同时，必须输入正确的扩展名，如 pc_com.c，然后单击“保存”按钮。

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)。

回到编辑界面后，单击“Target 1”前面的“+”号，在“Source Group 1”上单击右键，弹出菜单，然后单击 Add File to Group “Source Group 1”。

选中 pc_com.c，单击 Add 按钮，再单击 Close 按钮。此时注意到“Source Group 1”文件夹中多了一个子项“pc_com.c”。子项的多少与所增加的源程序的多少相同。

现在，请输入 C 语言源程序。

在输入程序时，就会看到事先保存待编辑文件的好处了，即 Keil C51 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。

3. 编译程序

单击 Project 菜单，在下拉菜单中单击 Options for Target “Target 1”选项，出现对话框，



选择 Output 选项卡, 选中 Create HEX Files 选项, 单击“确定”按钮。

再单击 Project 菜单, 在下拉菜单中单击 Built Target 选项 (或者使用快捷键 F7), 进行编译。若有错误会在 Output 窗口提示, 可根据此提示, 找出错误并修改, 直至编译通过, 如图 1-4 所示。

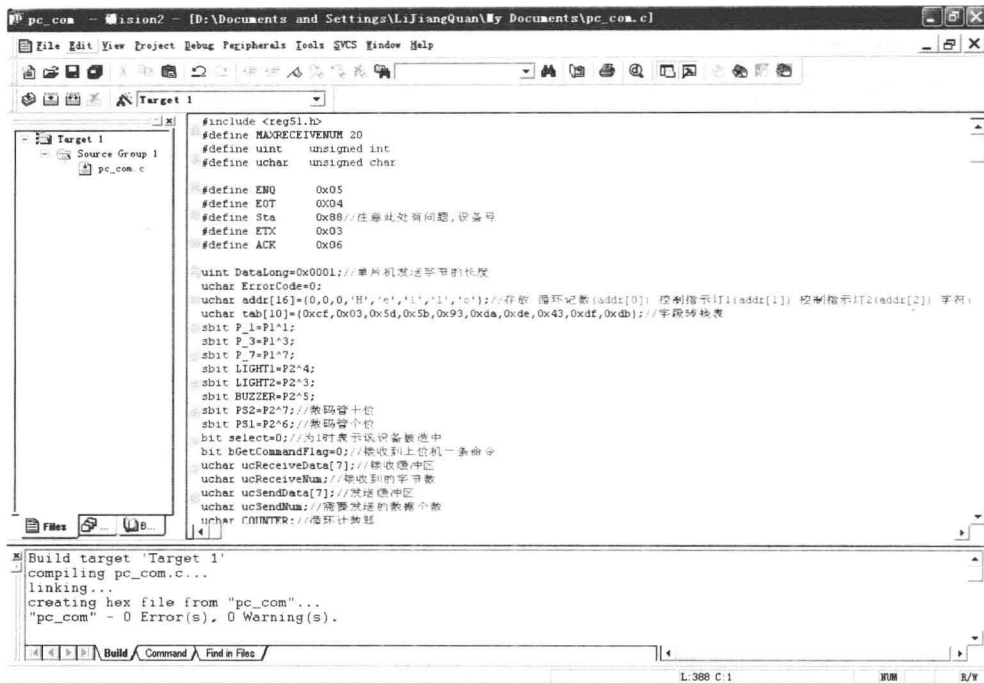


图 1-4 Keil C51 编译界面

至此, 用 Keil C51 做了一个完整工程。其中, 生成一个编程器烧写文件 pc_com.hex。以下是完成单片机模拟电压输入的 C51 参考程序:

```

**程序功能: 模拟电压输入, 显示屏显示 (保留 1 位小数), 并以十六进制形式发送给 PC
** 晶振频率: 11.0592MHz
** 线路→单片机开发板 B
***** /
#include <REG51.H>
#include <intrins.h>
/*****TLC0832 端口定义*****/
sbit ADC_CLK=P1^2;
sbit ADC_DO=P1^3;
sbit ADC_DI=P1^4;
sbit ADC_CS=P1^7;
/*****数码显示 键盘接口定义*****/
sbit PS0=P2^4; //数码管小数点后第一位
sbit PS1=P2^5; //数码管个位
sbit PS2=P2^6; //数码管十位

```



```
sbit PS3=P2^7;           //数码管百位
sfr P_data=0x80;         //P0 口为显示数据输出口
sbit P_K_L=P2^2;         //键盘列
sbit JDQ1=P2^0;          //继电器 1 控制
sbit JDQ2=P2^1;          //继电器 2 控制

//字段转换表
unsigned char tab[]={0xfc,0x60,0xda,0xf2,0x66,0xb6,0xbe,0xe0,0xfe,0xf6,
0xee,0x3e,0x9c,0x7a,0x9e,0x8e};

unsigned char adc_change(unsigned char a); //操作 TLC0832
unsigned int htd(unsigned int a);         //进制转换函数
void display(unsigned int a);             //显示函数
void delay(unsigned int);                 //延时函数
void main(void)
{
    unsigned int a,temp;
    TMOD=0x20;           //定时器 1—方式 2
    TL1=0xfd;
    TH1=0xfd;            //11.0592MHz 晶振,波特率为 9600
    SCON=0x50;           //方式 1
    TR1=1;               //启动定时
    while(1)
    {
        temp=adc_change('0')*10*5/255;
        for(a=0;a<200;a++)//显示,兼有延时的作用
            display(htd(temp));
        //SBUF=(unsigned char)(temp>>8);           //将测量结果发送给 PC
        //while(TI!=1);
        //TI=0;
        SBUF=(unsigned char)temp;
        while(TI!=1);
        TI=0;
        if(temp>45)
            JDQ1=0;           //继电器 1 动作
        else
            JDQ1=1;           //继电器 1 复位
        if(temp<5)
            JDQ2=0;           //继电器 2 动作
        else
            JDQ2=1;           //继电器 2 复位
    }
}

/*****数码管显示函数*****/
/*函数原型:void display(void)
```



```

/*函数功能:数码管显示
/*调用模块:delay()
/*****
void display(unsigned int a)
{
    bit b=P_K_L;
    P_K_L=1;                //防止按键干扰显示
    P_data=tab[a&0x0f];      //显示小数点后第1位
    PS0=0;
    PS1=1;
    PS2=1;
    PS3=1;
    delay(200);
    P_data=tab[(a>>4)&0x0f]|0x01;    //显示个位
    PS0=1;
    PS1=0;
    delay(200);
    //P_data=tab[(a>>8)&0x0f];      //显示十位
    PS1=1;
    //PS2=0;
    //delay(200);
    //P_data=tab[(a>>12)&0x0f];      //显示百位
    //PS2=1;
    //PS3=0;
    //delay(200);
    //PS3=1;
    P_K_L=b;                //恢复按键
    P_data=0xff;            //恢复数据口
}
/*****
; 函数名称:adc_change
; 功能描述:TI 公司 8 位 2 通 adc 芯片 TLC0832 的控制时序
; 形式参数:config(无符号整型变量)
; 返回参数:a_data
; 局部变量:m、n
*****/
unsigned char adc_change(unsigned char config)//操作 TLC0832
{
    unsigned char i,a_data=0;
    ADC_CLK=0;
    _nop_();
    ADC_DI=0;
    _nop_();

```

```

ADC_CS=0;
_nop_();
ADC_DI=1;
_nop_();
ADC_CLK=1;
_nop_();
ADC_CLK=0;
    if(config=='0')
    {
        ADC_DI=1;
        _nop_();
        ADC_CLK=1;
        _nop_();
        ADC_DI=0;
        _nop_();
        ADC_CLK=0;
    }
    else if(config=='1')
    {
        ADC_DI=1;
        _nop_();
        ADC_CLK=1;
        _nop_();
        ADC_DI=1;
        _nop_();
        ADC_CLK=0;
    }
    ADC_CLK=1;
    _nop_();
    ADC_CLK=0;
    _nop_();
    ADC_CLK=1;
    _nop_();
    ADC_CLK=0;
    for(i=0;i<8;i++)
    {
        a_data<<=1;
        ADC_CLK=0;
        a_data+=(unsigned char)ADC_DO;
        ADC_CLK=1;
    }
    ADC_CS=1;
    ADC_DI=1;

```



```

    return a_data;
}
/*****十六进制转十进制函数*****/
/*函数原型:uint htd(uint a)
/*函数功能:十六进制转十进制
/*输入参数:要转换的数据
/*输出参数:转换后的数据
/*****/
unsigned int htd(unsigned int a)
{
    unsigned int b,c;
    b=a%10;
    c=b;
    a=a/10;
    b=a%10;
    c=c+(b<<4);
    a=a/10;
    b=a%10;
    c=c+(b<<8);
    a=a/10;
    b=a%10;
    c=c+(b<<12);
    return c;
}
/*****延时函数*****/
/*函数原型:delay(unsigned int delay_time)
/*函数功能:延时函数
/*输入参数:delay_time (输入要延时的时间)
/*****/
void delay(unsigned int delay_time) //延时子程序
{for(;delay_time>0;delay_time--)
{}
}

```

4. 烧写程序

程序经过调试运行后就可以将其烧写进单片机了。STC 系列单片机在线下载程序只需要用串口连接到单片机上就可以了。用串口线连接 PC 与单片机实验板,将编写好的汇编程序用 Keil μ Vision3 编译生成 HEX 文件就可以实现程序的简便烧写。

到网站 <http://www.mcu-memory.com> 下载 STC 单片机 ISP 下载编程软件。按照提示在计算机上运行该程序,其界面如图 1-5 所示。

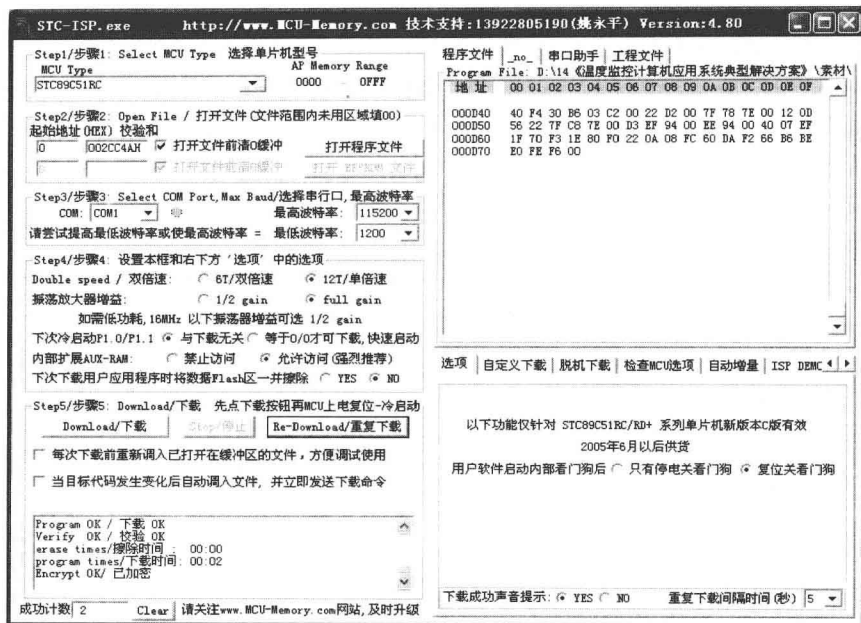


图 1-5 单片机烧写程序

烧写程序步骤如下。

连接单片机开发板与 PC 后，实验板电源先不要接通。

接着按下面步骤完成单片机程序的在线下载：

第一步：选择要下载程序的单片机型号。

第二步：打开编译完成要下载到单片机中扩展名为 HEX 的文件。

第三步：选择与实验板连接的串口。

第四步：选择合适的通信波特率（这步可以省略）。

第五步：将该选项选中（选中的目的是便于程序的调试，若下载的是调试通过的程序，此步可省略）。

第六步：单击该按钮。

第七步：接通实验板电源。

几秒后就可以将程序下载到单片机并运行了。

程序烧写进单片机之后，就可以给单片机开发板通电了，这时数码管上将会显示检测的电压值。

5. 串口通信调试

在进行串口开发之前，一般要进行串口调试，经常使用的工具是“串口调试助手”程序。它是一个适用于 Windows 平台的串口监视、串口调试程序。它可以在线设置各种通信速率、通信端口等参数；既可以发送字符串命令，也可以发送文件；可以设置自动发送/手动发送方式；可以十六进制显示接收到的数据等。从而提高串口开发效率。

打开“串口调试助手”程序（ScomAssistant.exe），首先设置串口号为 COM1、波特率为 9600、校验位为 NONE、数据位为 8、停止位为 1（注意：设置的参数必须与单片机设