

# 单片机仿真系统指导书

阎有运 郭顺京 成凌飞 编著

焦作工学院电气工程系

2001年10月

# 前 言

本书由两部分组成，一部分是单片机系统的仿真，另一部分是仿真系统目标机。

单片机系统仿真采用的是伟福 E2000/T 型仿真器，其主要功能是：通用仿真器（1-16 位，15M 总线速度）、硬件测试仪、运行时间统计、逻辑笔（选配件）、Windows 版本，DOS 版本双平台、事件断点，断点记数、跟踪器、影子存储器、全空间程序/数据时效分析器等。

编程器采用的是 TOP851 通用编程器，是专为开发 51 系列单片机和烧写各类存储器的普及型机，采用 RS232 串口与 PC 机连接通信，抗干扰性能好，可靠性高，特别适合烧写各种一次性（OTP）和电擦除器件。

仿真系统目标机（数字钟）作为一个例子，其目的是熟悉单片机的开发与研制的过程，熟悉仿真系统和编程器的使用，达到能够独立地利用单片机开发各种系统的能力。

---

## 仿真系统实验指导书

单片机仿真开发系统作为单片机重要开发手段，一般说来应具有如下最基本功能：

1. 用户样机硬件电路的诊断与检查；
2. 用户样机程序的输入与修改；
3. 程序的运行、调试（单步运行、设置断点运行）、排错、状态查询等功能；
4. 将程序固化到 EPROM 芯片中。

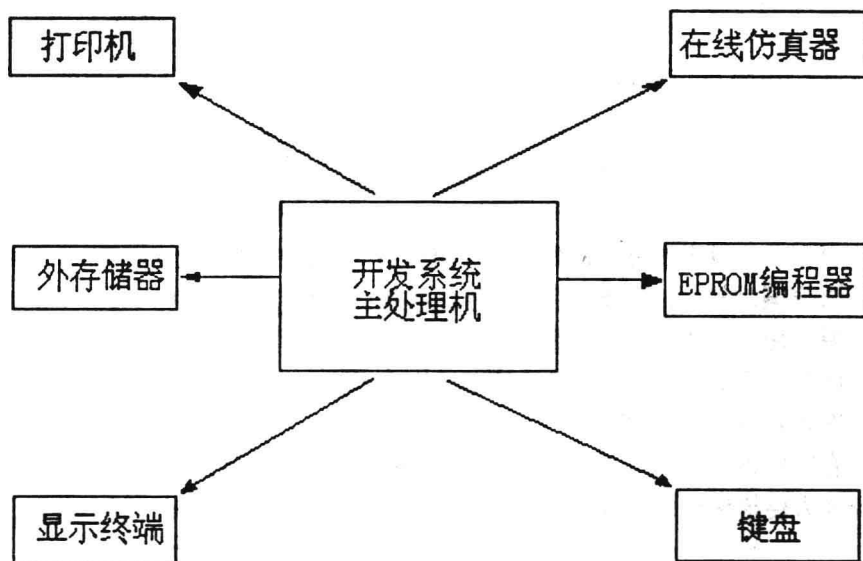
不同的开发系统都必须具备上述功能，但对于一个较完善的开发系统还应具备：

1. 有较全的开发软件。最好配有高级语言（PL/M、C 等），用户可用高级语言编制软件；由开发系统编译连接生成目标文件、可执行文件。同时要求用户可用汇编语言编制应用软件；开发系统自动生成目标文件；并配有反汇编软件，能将目标程序转换成汇编语言程序；有丰富的子程序库可供用户选择调用。
2. 有跟踪调试、运行的能力。开发系统占单片机硬件资源尽量最少。
3. 为方便模块化软件调试，还应配备软件转储、程序文本打印功能及设备。

---

## 仿真系统基本组成

一个完整的开发系统有硬件和软件两大部分组成，下图是一个典型的开发系统硬件配置：



### 1、主处理机

它是开发系统的核心，一般是一个带有监控程序的单板机或一个带有操作系统的微计算机。开发系统的硬件动作及软件运行全由它控制。

### 2、控制台

控制台包括键盘和显示终端两部分，是实现人机对话的必备部件。

### 3、外存储器

通常是指硬磁盘或软磁盘，主要用来存放开发系统的系统软件和暂存用户的应用程序。

### 4、打印机

---

打印用户系统资料和程序清单。

#### 5、在线仿真器

这是开发系统的关键部件，缺少这一部分不能称为开发系统。在单片机开发系统上编好应用程序后，一项重要的工作就是调试应用程序，而且最好是在用户环境中调试程序。在线仿真器正是起到了这个作用。它为用户程序提供了调试环境，把目标硬件（用户样机）同开发系统联系起来，使用户可以在开发的实际目标硬件环境下，全面调试用户目标程序。

#### 6、EPROM 编程器

将调试好的用户程序固化到 EPROM，使用户样机脱离开发系统自己运行。

#### 7、开发系统软件

开发系统软件一般由编辑程序、汇编（或编译）程序及动态调试程序三部分组成。其中编辑程序和汇编程序的运行并不需要在线仿真器的配合，而动态调试程序则要在与在线仿真器配合才能运行。

## 用户样机的开发调试过程

完成一个用户样机，首先要完成硬件组装工作，然后进入软件设计、调试和硬件调试阶段。硬件组装就是在设计、制作完毕的印刷板上焊好元件与插座，然后就可利用开发工具进行软件设计、调试和硬件调试工作。

用户样机软件设计调试过程如下图：

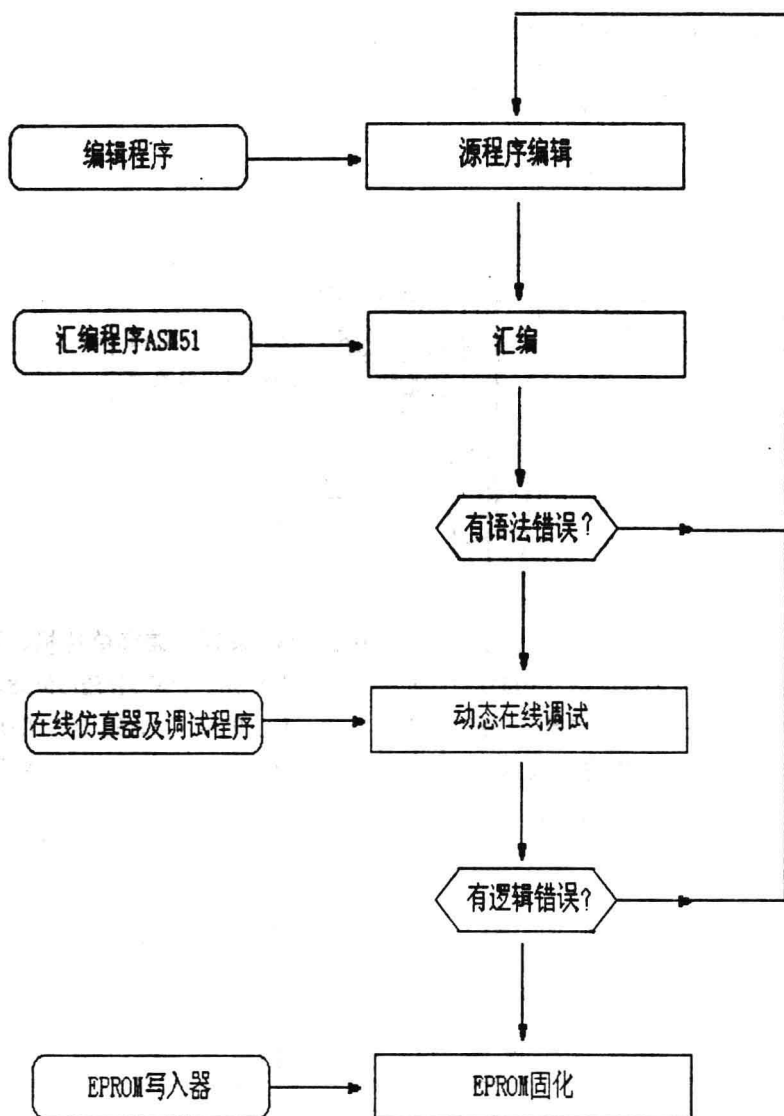
第一步，建立用户程序。用户按照开发系统的编辑软件所要求的格式、语法的规定，把源程序输入到开发系统中，并存在磁盘上。

第二步，在开发系统机上，利用汇编软件对用户输入的源程序进行汇编，变成可执行的目标代码。在汇编过程中，如果用户源程序有语法错误，则返回到第一步进行修改，再进行汇编，直到语法错误全部纠正为止。

第三步，动态在线调试。这一步对用户程序进行调试。上述第一、二是纯粹的软件运行过程，而这一步必须要有在线仿真器的配合，才能对用户程序进行调试。发现逻辑错误，返回到第一步修改，直到逻辑错误全部纠正。在调试过程中，硬件调试与软件调试不能完全分开。

---

第四步，将调试好的用户程序通过 EPROM 编程器固化在 EPROM 中。

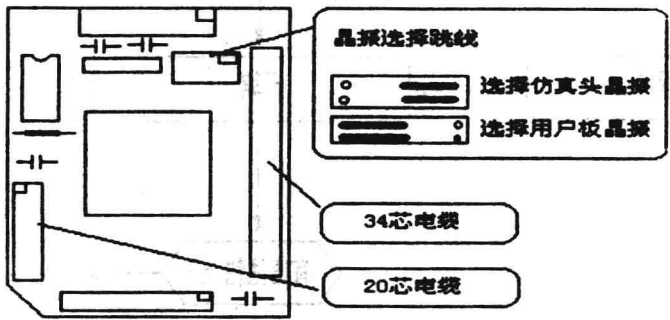


用户样机软件设计、调试

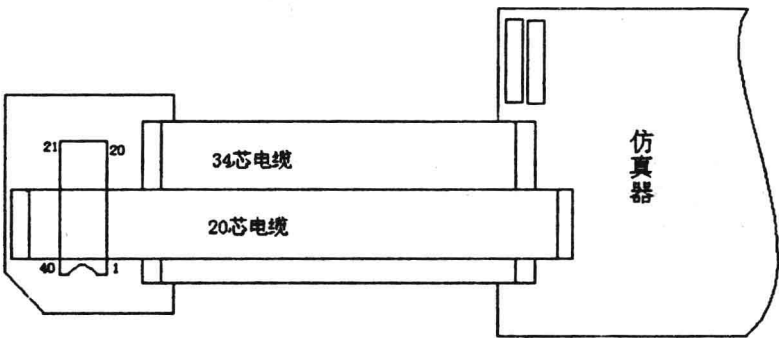
# 伟福仿真系统使用说明

## 一、仿真头介绍

### 1、POD8X5X 仿真头

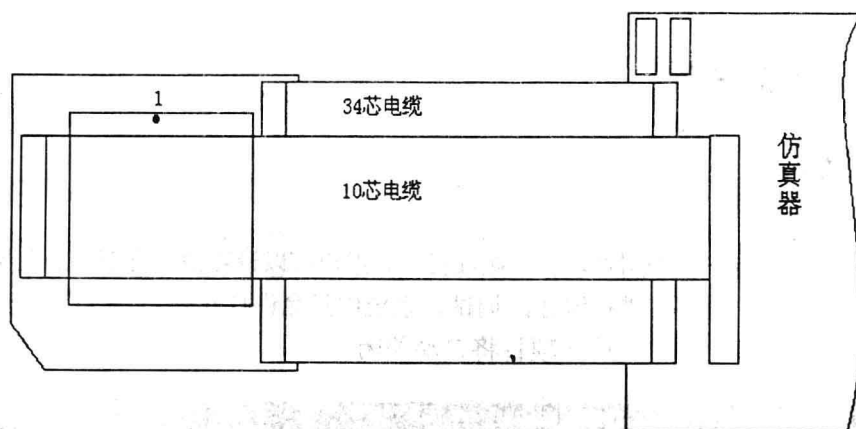
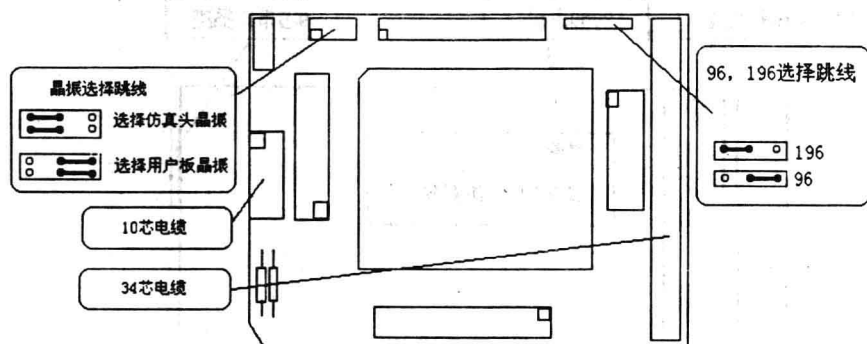


POD8X5X 仿真头用于仿真 INTEL8031/8051 系列及兼容单片机，可仿真 CPU 种类为 8031/32、8051/52、8751/52/54/55/58、89C51/52/55/58、89C1051/2051/4051、华邦的 78E51/52/54/58、LG 的 97C51/52/1051/2051。配有 40 脚 DIP 封装的转接座。选配 2051 转接可仿真 20 脚 DIP 封装的 XXC1051/2051/4051CPU。



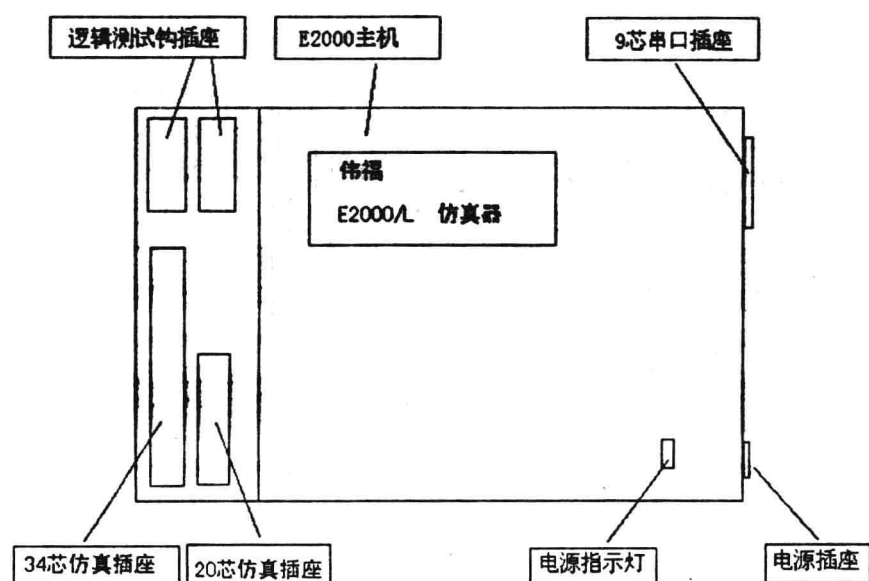
E2000 /51仿真器与POD8X5X连接图

## 2、POD196KB/KC 仿真头



E2000仿真器与POD196KC连接图

## 二、 仿真器介绍

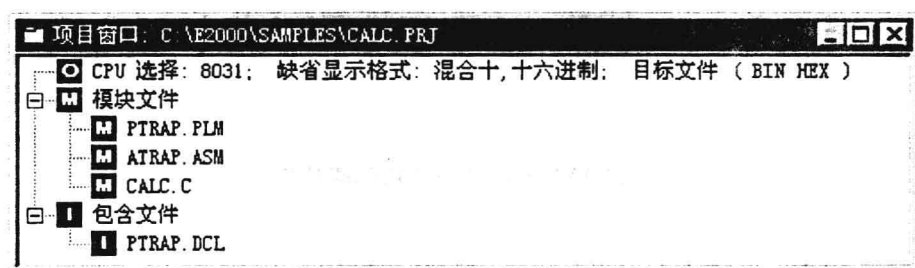


## 三、 开发环境

### ◆文件

#### 打开项目

打开一个用户项目，在项目中，用户可以设置仿真类型。加入用户程序，进行编译，调试。系统中只允许打开一个项目，打开一个项目，前一项目将自动关闭。



---

伟福开发环境的项目文件包括仿真器设置, 模块文件, 包含文件. 仿真器设置包括仿真器类型, 仿真头 (POD) 类型, CPU 类型, 显示格式和产生的目标文件类型可以用以下几种方法设置仿真器

- 在项目窗口中双击第一行, 将打开仿真器设置窗口, 对仿真器进行设置.
- 按鼠标右键, 在弹出菜单中选择[仿真器设置].
- 主菜单 真器/仿真器设置.

#### 加入模块文件

- 按鼠标右键, 在弹出菜单中选择"加入模块文件"
- 主菜单 项目/加入模块文件

#### 加入包含文件

- 按鼠标右键, 在弹出菜单中选择"加入包含文件"
- 主菜单 项目/加入包含文件

如何将以前单文件方式仿真转为 WINDOWS 下的项目方式进行仿真

- 一、主菜单 文件/新建项目, 在新建项目时, 前一个项目自动关闭.
- 二、加入模块文件时, 选择以前的程序文件名, 将文件加入项目.
- 三、将项目存盘.
- 四、编译, 运行, 调试项目.

### 保存项目

将用户项目存盘. 用户在编译项目时, 自动存盘.

主菜单 文件/保存项目

### 新建项目

当用户开始新的任务时, 应新建一个项目, 在项目中, 设置所用仿真器类型, POD 类型, 加入用户程序 (模块).

### 项目另存为

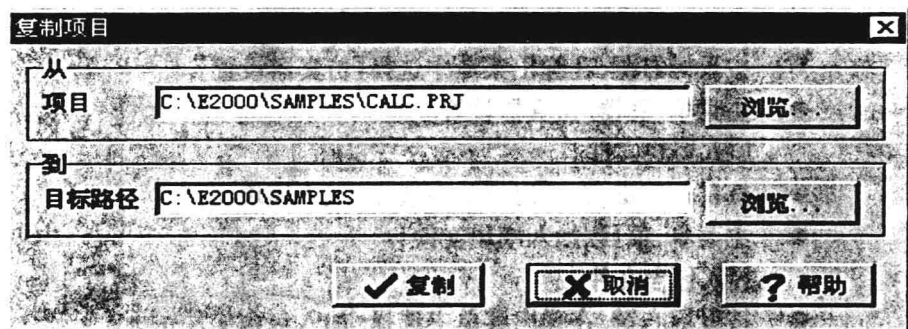
将项目换名存盘,

### 复制项目

复制项目, 用户可以将项目中的所有模块 (用户程序) 备份到另

一个地方。在多模块项目中,用复制项目功能,可以避免用户因为少复制某些模块,而造成项目编译不能通过.方便用户对程序进行管理.

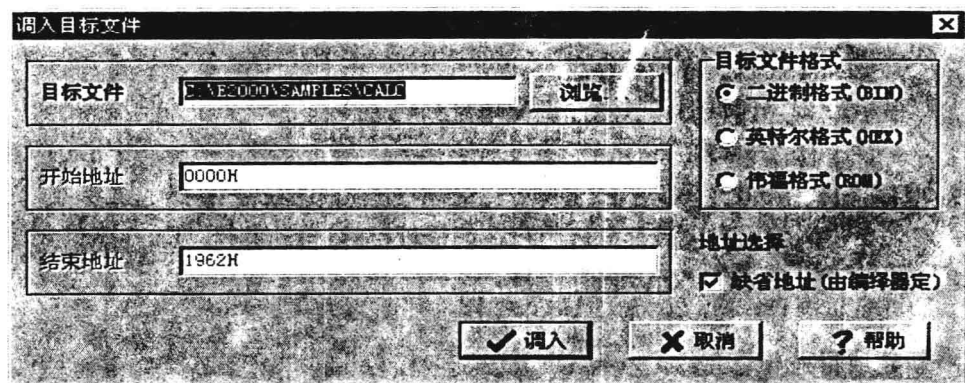
方法:主菜单 文件/复制项目



[从项目]中为当前被复制项目,包括项目中各模块,包含文件,如果不是复制当前项目,可以通过[浏览]找到所要复制的项目,[到目标路径]中为项目复制到何处,可以通过其后的[浏览]指定将项目复制到其它地方.

### 调入目标文件

装入用户已编译好目标文件。系统支持两种文件格式:BIN, HEX



(图:调入目标文件对话框)

调入已经编译好的目标文件

目标文件格式有二种:

二进制 (BIN): 由编译器生成的二进制文件,也就是程序的机器码

英特尔格式(HEX)：由英特尔定义的一种格式,用 ASCII 码来存储编译器生成的二进制代码,这种格式包括地址,数据和校验地址选择一般为缺省地址(由编译器定).如果想在当前项目已编译好的二进制代码中插入一段其它代码,可以去掉“缺省地址”前的选择,然后填入开始插入的地址和结束地址.用调入目标文件的方法,可以调试已有的二进制代码程序.而不需要源程序.

直接调入目标文件进行仿真的方法是：

- 1、建项目.
- 2、在新建的项目中,设置仿真器类型,仿真头类型,CPU 类型.
- 3、调入目标文件.(不要用加入模块方式,而是直接调入文件)
- 4、打开 CPU 窗口,在 CPU 窗口中就可以看见目标文件反汇编生成的程序.
- 5、程序停在与 CPU 相关的地址上(51 系列停在 0000H 处,96 系列停在 2080H)
- 6、这样就可以单步或全速调试程序了.

## 保存目标文件

将用户编译生成的目标文件存盘。

保存目标文件

目标文件: C:\E2000\SAMPLES\CALC 浏览...

开始地址: 0000H

结束地址: 1906H

目标文件格式:  
☒ 二进制格式(BIN)  
☐ 英特尔格式(HEX)

地址选择:  
☒ 缺省地址(由编译器定)

保存 取消 帮助

对于按项目方式仿真的用户,系统将程序编译正确后,会根据用户在 仿真器/仿真器设置/目标文件 中设置的格式,将生成的二进制代码存盘.如果用户是用调入目标文件方式进行仿真,并且对目标码进行了修改,就可以用[文件/保存目标文件]方式存盘.

---

目标文件可以存成两种格式：

二进制(BIN)、英特尔格式(HEX). 由英特尔定义的一种格式, 用 ASCII 码来存储编译器生成的二进制代码, 这种格式包括地址, 数据和校验地址选择一般为缺省地址(由编译器定). 如果想要存盘的目标文件是由“调入目标文件”方式装入, 而不是由系统编译产生的代码, 并已经修改, 最好指定它的开始地址和结束地址, 因为代码不是编译系统产生的. 系统不知道文件有多长, 无法指定开始和结束地址. 自己指定地址的方法是: 去掉“缺省地址”前的选择. 然后填入开始插入的地址和结束地址.

形, 跟踪器窗口

## ◆ 搜索

### 查找

在当前窗口中查找符号, 字串. 可以指定区分大小写方式, 全字匹配方式, 可以向上 / 向下查找。

### 替换

在当前窗口查找相应文字, 并替换成指定的文字, 可以指定区分大小写方式和全字匹配方式查找, 可以在指定处替换, 也可以全部替换。

### 查找下一个

查找文字符号下一次出现的地方

### 项目中查找

在项目所有模块(文件)中查找符号, 字串. 在项目所包含的文件比较多时, 用此方法可以很方便地查到字串在什么地方出现。

### 转到指定行

将光标转到程序的某一行。

### 转到指定地址/标号

将光标转到指定地址或标号所在的位置。

### 转到当前 PC 所在行

将光标转到 PC 所在的程序位置。

## ◆ 项目

### 编译

编译当前窗口的程序。如有错误, 系统将会指出错误所在的位置。

---

## **全部编译**

全部编译项目中所有的模块（程序文件），包含文件。如有错误系统会指出错误所在位置。

## **加入模块文件**

在当前项目中添加一个模块。

## **加入包含文件**

在当前项目中添加一个包含文件

# **◆ 执行**

## **全速执行**

运行程序

## **跟踪**

跟踪程序执行的每步，观察程序运行状态。

## **单步**

单步执行程序，与跟踪不同的是，跟踪可以跟踪到函数或过程的内部，而单步执行则不跟踪到程序内部。

## **执行到光标处**

程序从当前 PC 位置，全速执行到光标所在的行。如果光标所在行没有可执行代码。则提示“这行没有代码”

## **暂停**

暂停正在全速执行的程序。

## **复位**

终止调试过程，程序将被复位。如果程序正在全速执行，则应先停止。

## **设置 PC**

将程序指针 PC，设置到光标所在行。程序将从光标所在行开始执行。

## **编辑观察项**

观察变量或表达式的值，可以将需要检查和修改的值或表达式放到观察窗口里以便检查和修改。

[表达式]: 用于输入用户所要求值的表达式。(例)

[重复次数]: 如果表达式为某一存储变量, 重复次数表示以此变量开始的连续 N 个地址的值。(例)

[显示格式]: 指定用何种方式显示表达式的值。

[存储区域]: 指明变量所在的区域。

[显示类型]: 指定表达式为何种类型的变量。

[缺省方式显示]: 按照高级语言定义的方式显示。

[存储器内容]: 以内存方式显示观察内容, 也就是按地址顺序显示变量值, 与变量类型无关

[求值]: 对表达式求值, 并按显示格式显示在窗口内。

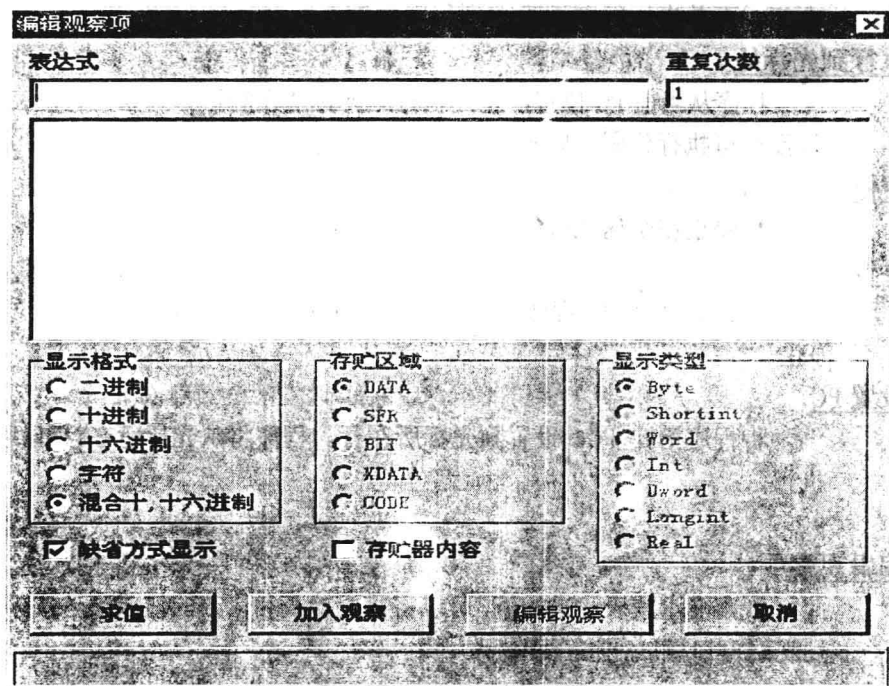
[加入观察]: 将表达式加入观察窗口中, 以便随时察看。

[编辑观察]: 当修改过窗口内容后, 按此键后, 替代观察窗口中的原观察项, 如果选择‘加入观察’, 则会在观察窗口中另加一个变量的观察项, 以两种格式观察一个变量。

[取消]: 关闭编辑观察项窗口

### 设置/取消断点

将光标所在行设为断点, 如果该行原来已为断点, 则取消该断点。



有四种方法可以在光标处设置断点:

1. 将光标移到编辑窗口内, 行左边的空白处, 光标变成右上指箭头, 双击鼠标左键, 可以设置/取消断点.
2. 使用 Ctrl-F8 快捷键, 可以在光标所在行设置/取消断点
3. 右击鼠标, 弹出菜单, 选择 设置/取消断点,
4. 主菜单 执行/设置取消断点, 也可以用 Alt-R / B 菜单快捷设置取消断点. 断点设置好后, 当前行左空白处出现 X 图. 如果断点有效, 则当前行背景色为红, 如背景色为绿, 说明当前程序行不产生代码, 断点无效.

### 清除全部断点

清除程序中所有的断点。让程序全速执行

## ◆窗口

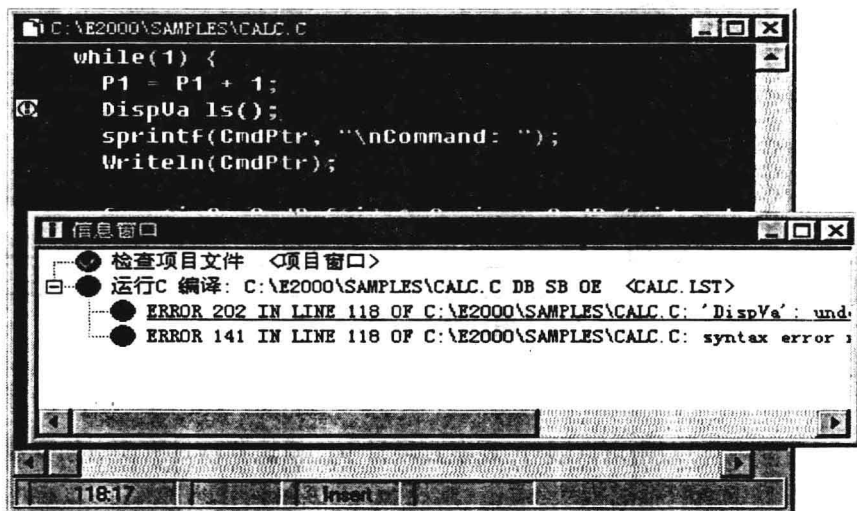
### 项目窗口

打开项目窗口, 以便在项目中加入模块或包含文件。

### 信息窗口

显示系统编译输出的信息。如果程序有错, 会以图标形式指出,

● 表示错误, ● 表示警告, ● 表示通过



在编译信息行会有相关的生成文件, 双击鼠标左键, 或击右键在弹出菜