Deliver to： NEC Electronics Hong Kong Limited

Deliver from： Beijing NEC IC Design Corp， Ltd

NEC Tools 78Kx 系列
用户手册

VERSION 1.0

Date Published Jun， 2004

前言

目的

本手册的目的是向用户讲解 NEC 开发工具包——NEC Tools 的基本概念和操作。通过本手册的指导，用户可以实际使用 NEC Tools 中的 PM plus 和 SM78K0 plus、SM78K0s plus 工具，分别针对 78K0 和 78K0s 系列芯片进行编程开发活动。

预备知识

本手册面向的读者是准备开发基于 78K0 和 78K0s 芯片的软件的初学者。读者需要具备一定的 C、汇编和微处理器（MCU）的基础知识。

章节

本手册分为以下几个章节：

- 安装
- 范例
- 简介
- 编译
- 调试

术语

78Kx: 78K0 和 78K0s 在本手册中统称为 78Kx。在 NEC开发工具中，78K0 系列 CPU 和 78K0s 系列 CPU 各有一套独立的开发工具。

参考文档

本手册参考的文档都是 NEC 正式发布的文档，可以在以下网址索引到：http：//www.necel.com/en/search/index.html#document

参考文档

名称	编号
CC78K0 UM	U16613E
CC78K0s UM	U16654E
RA78K0 UM	U16629E
RA78K0s UM	U16656E
SM78K V2.52 UM	U16768E
PMplus E5.10 UM	U16569E

目录

第 1 章 安装开发工具

本章将介绍如何安装 NEC 工具包 ra78Kx/PM plus、cc78Kx 和 SM78Kx plus。

1.1 安装 ra78Kx/PM plus

1.1.1 安装包内容

ra78Kx 安装包中包括 ra78Kx 和 PM78Kx plus 两个软件包：

ra78Kx：

ra78Kx 是用于 NEC 78Kx 系列微处理器的汇编编译工具包。它可将汇编源程序编译为 78Kx 芯片的目标代码。

PM plus：

PM plus 是一个集成开发环境平台，用来有效地开发 NEC 的 8/16 位微控制器 78Kx 系列的用户程序。PM plus 包括一个 Project Manager 和一个屏幕编辑器，提供了一系列的操作功能，如编辑器功能、编译器功能、开发向导功能等。

1.1.2 安装步骤

运行 ra78Kx 安装包中的 setup 文件，开始安装。

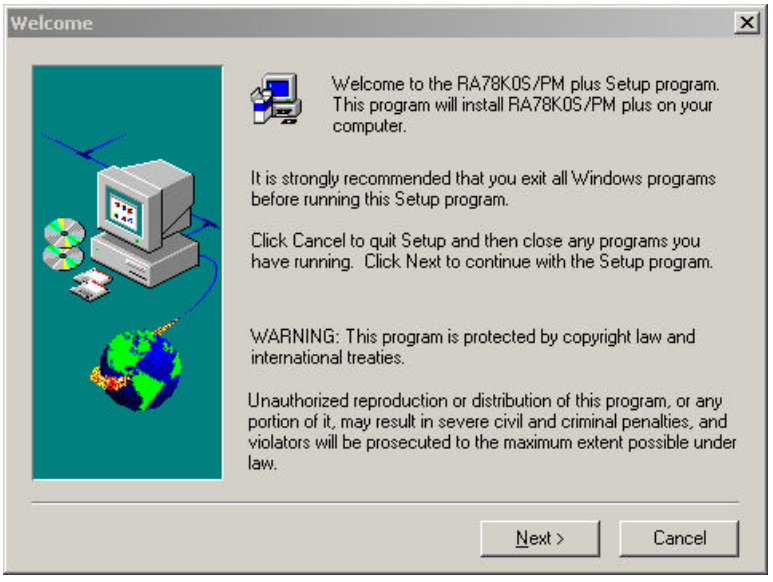

图 1-1 Welcome 对话框

点击 Next 按钮，出现 Software License Agreement 对话框。

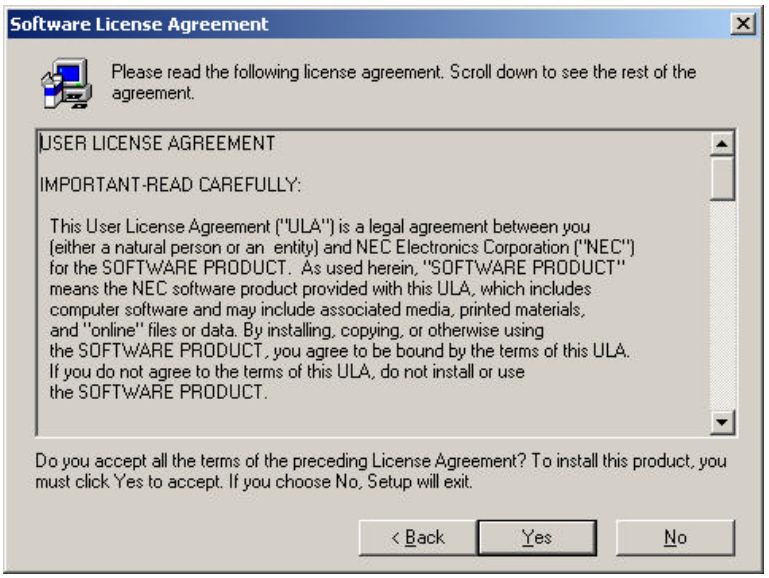

图 1-2 Software License Agreement 对话框

点击 Yes 按钮，出现 Product ID 对话框。

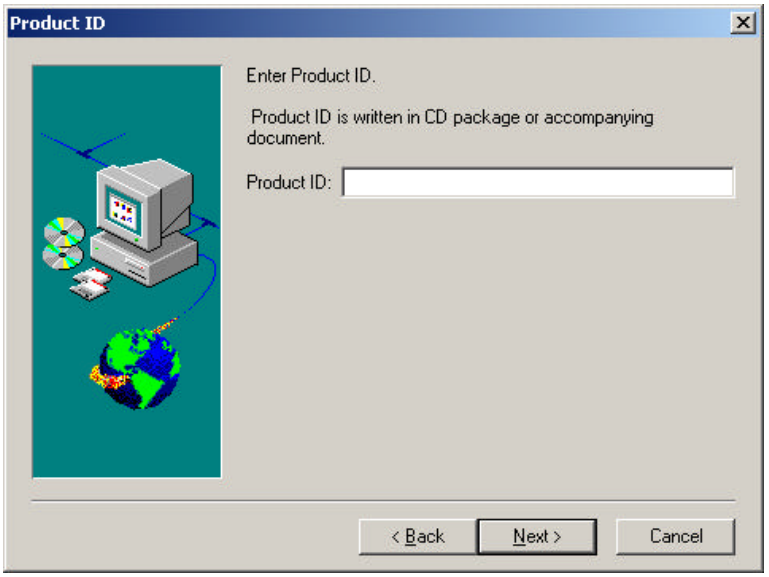

图 1-3 Product ID 对话框

输入序列号，点击 Next 按钮，出现 Select Components 对话框。选择要安装的组件，默认选择 ra78Kx Assembler Package 和 PM plus 以及它们的相关文档。

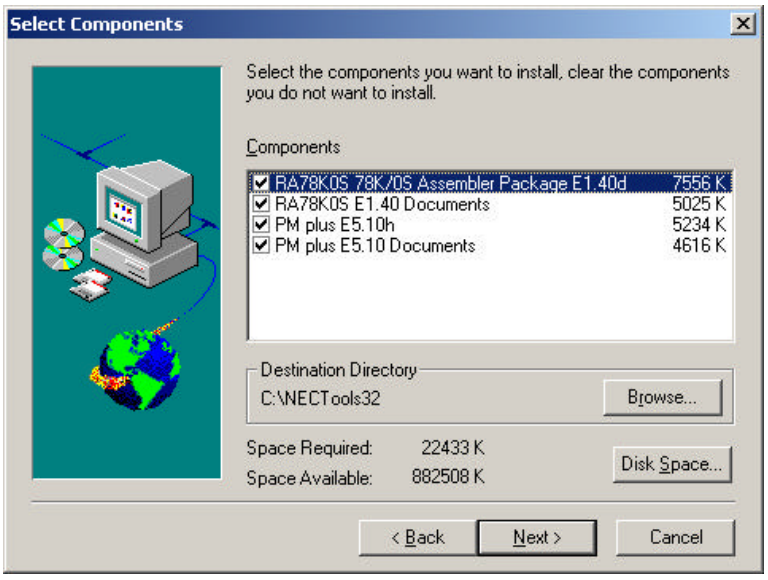

图 1-4 Select Components 对话框

点击 Browse 按钮，指定要安装的路径，默认路径是“C：\NECTools32”。点击 Next，出现 Select Program Folder 对话框。默认设置是“NEC Tools32 ”，也可以修改为其他的目录名。

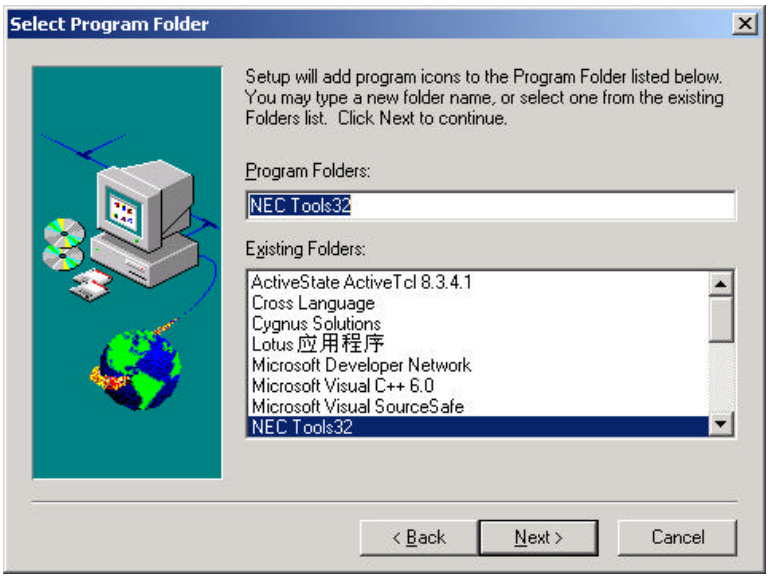

图 1-5 Select Program Folder 对话框

点击 Next，出现 Start Copying Files 对话框。检查当前的设置。

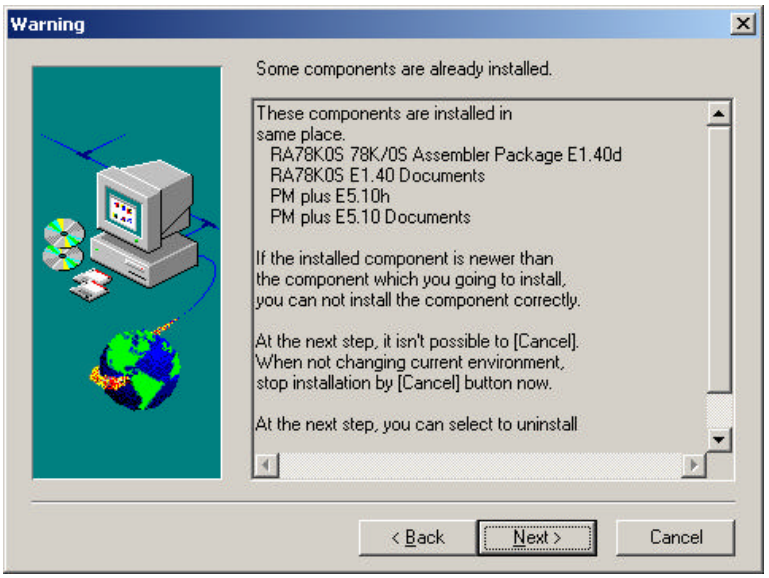

图 1-6 Start Copying Files 对话框

点击 Next 开始拷贝文件，等待拷贝完成。

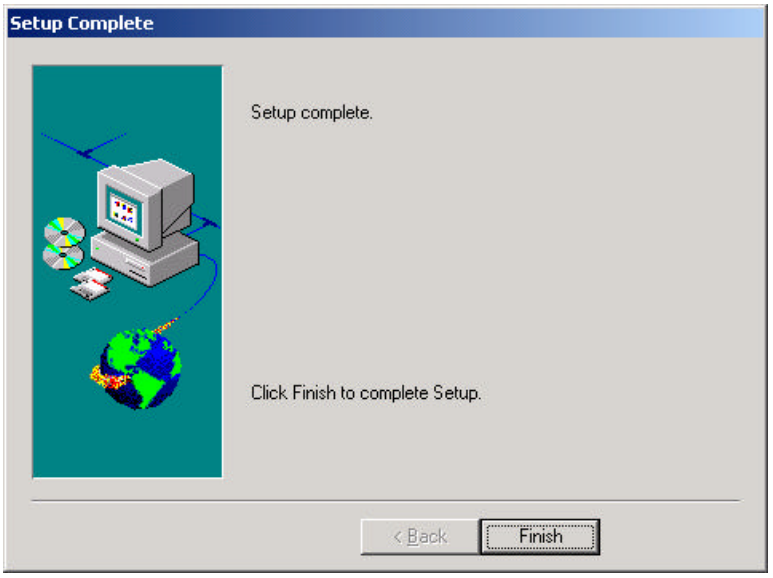

图 1-7 Setup Complete 对话框

出现 Setup Complete 对话框，点击 Finish 按钮，安装就完成了。

1.2 安装 cc78Kx

1.2.1 安装包内容

cc78Kx 是用于 NEC 78Kx 系列微处理器的 c 语言编译工具包。它具有以下特征：

- C 语言库符合 ANSI 标准
- 支持程序 ROM 存储方式
- 集成多种编译功能，灵活易用

1.2.2 安装步骤

运行 cc78Kx 安装包中的 setup 文件，开始安装。

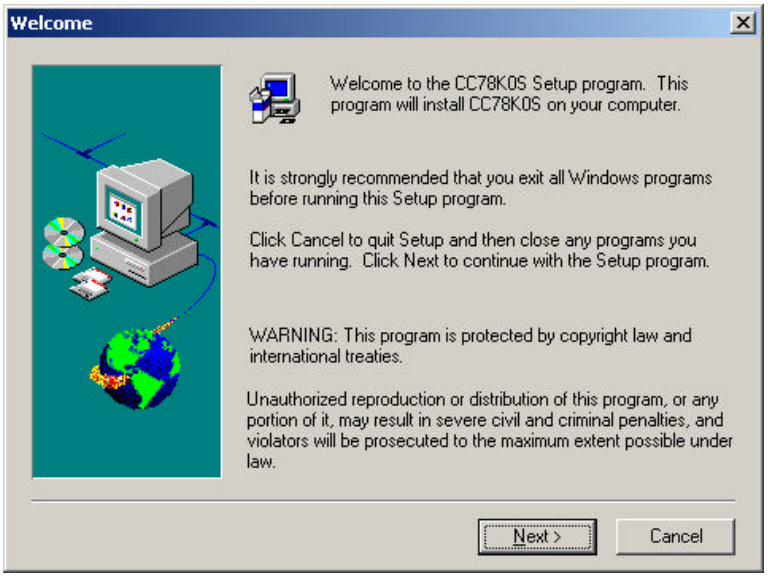

图 1-8 Welcome 对话框

点击 Next 按钮，出现 Software License Agreement 对话框。

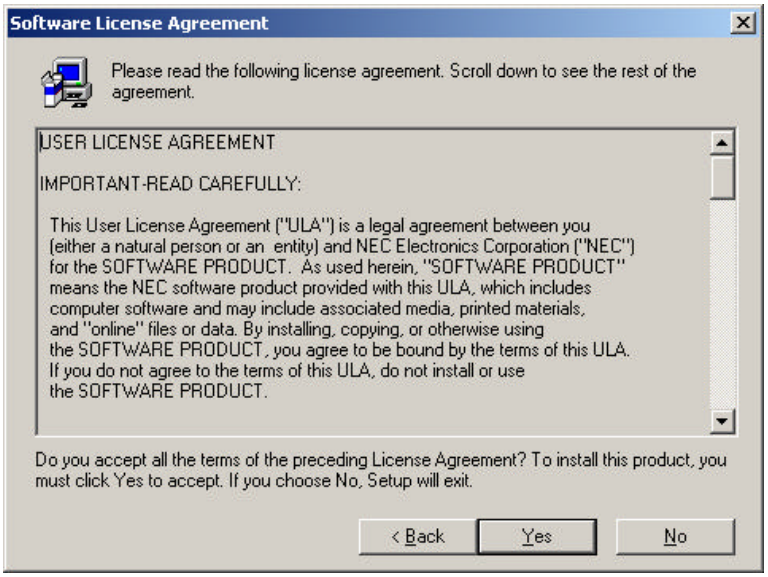

图 1-9 Software License Agreement 对话框

点击 Yes 按钮，出现 Product ID 对话框。

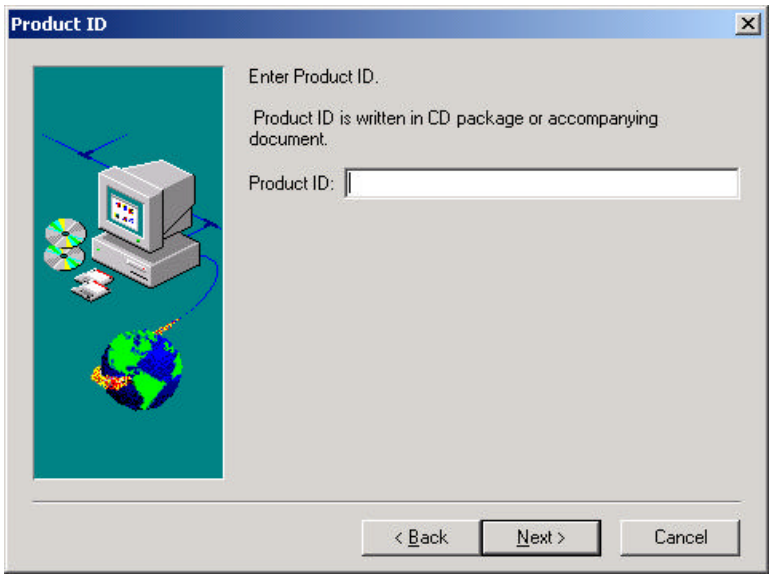

图 1-10 Product ID 对话框

输入序列号，点击 Next 按钮，出现 Select Components 对话框。选择要安装的组件，默认选择 cc78Kx C Compiler 和它的相关文档。

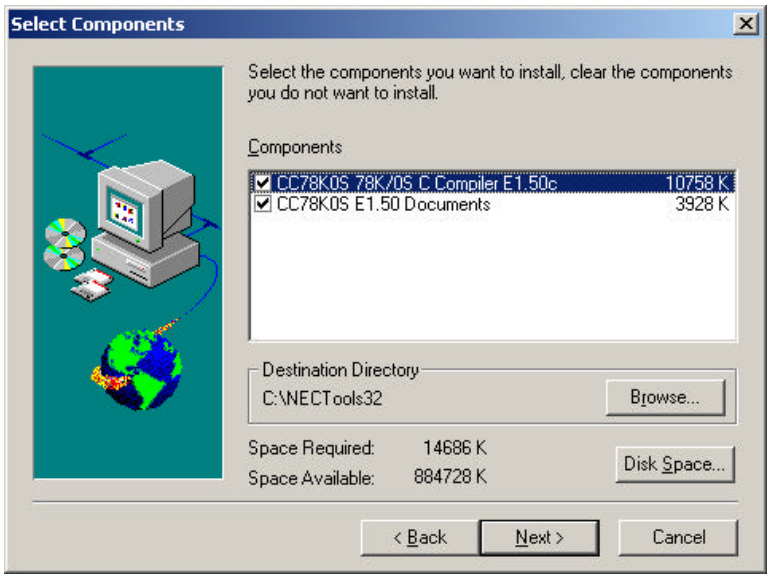

图 1-11 Select Components 对话框

点击 Browse 按钮，指定要安装的路径，默认路径是“C：\NECTools32”。点击 Next，出现 Select Program Folder 对话框。默认设置是“NEC Tools32”，也可以修改为其他的目录名。

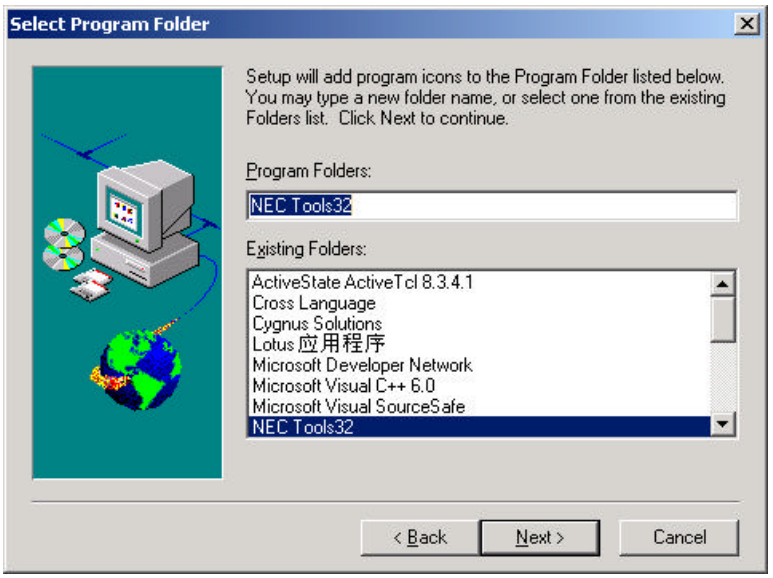

图 1-12 Select Program Folder 对话框

点击 Next，出现 Start Copying Files 对话框。检查当前的设置。

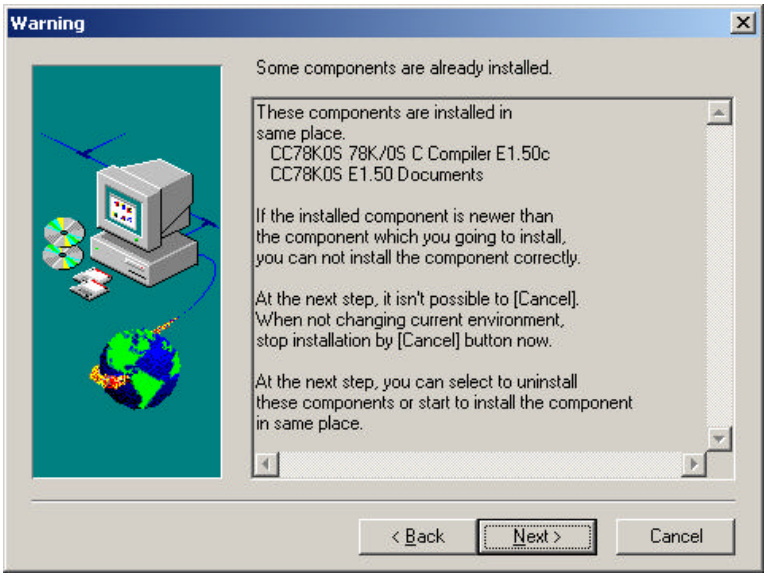

图 1-13 Start Copying Files 对话框

点击 Next 开始拷贝文件，等待拷贝完成。

1.3 安装 SM78Kx PLUS

1.3.1 安装包内容

SM78Kx plus 是 NEC 提供的调试软件，来帮助用户查找 bug，观察程序执行结果。它仿真了 NEC 的多种 CPU 和芯片，给用户提供了一个集仿真、调试、性能分析于一体的功能强大的软件平台。

1.3.2 安装步骤

执行 SM78Kx plus 的 Setup.exe 程序，出现 Welcome 对话框。

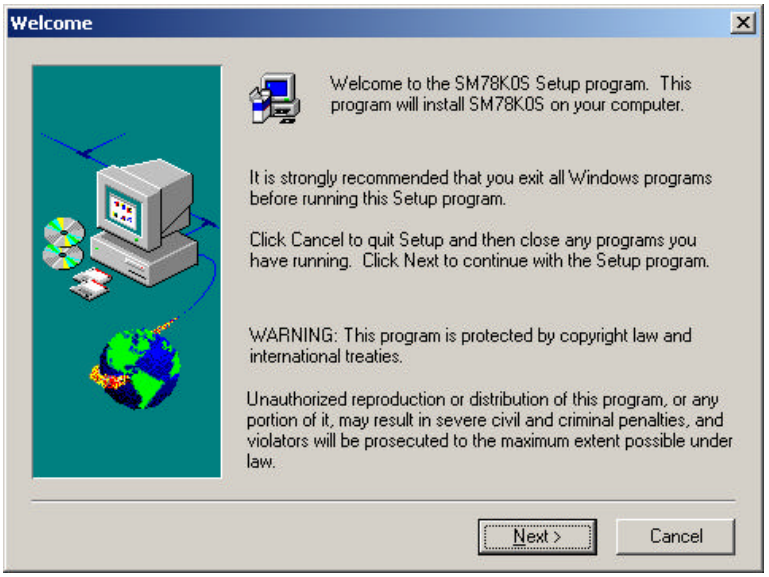

图 1-14 Welcome 对话框

点击 Next 按钮，出现 Software License Agreement 对话框。

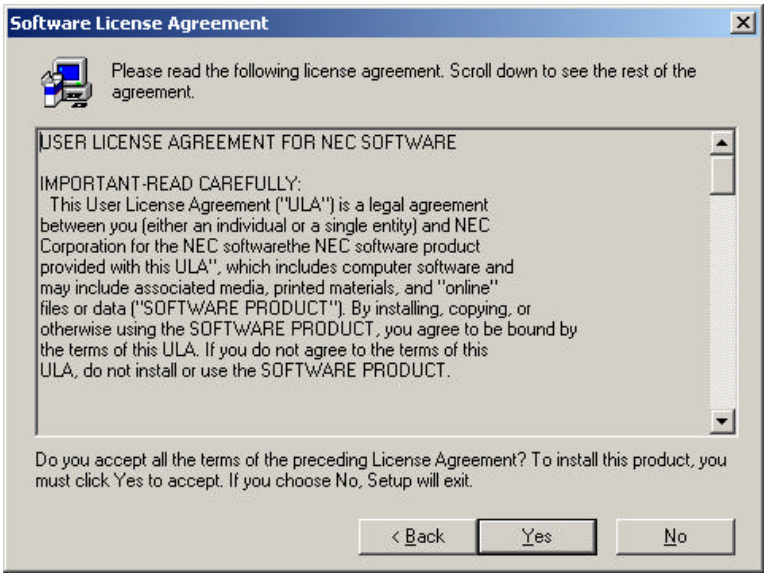

图 1-15 Software License Agreement 对话框

点击 Yes 按钮，出现 Product ID 对话框。

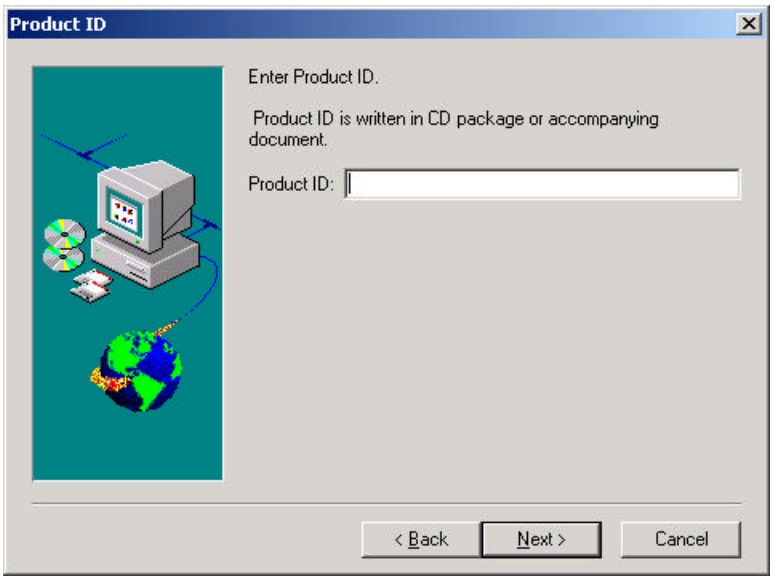

图 1-16 Product ID 对话框

输入序列号，点击 Next 按钮，出现 Select Components 对话框。

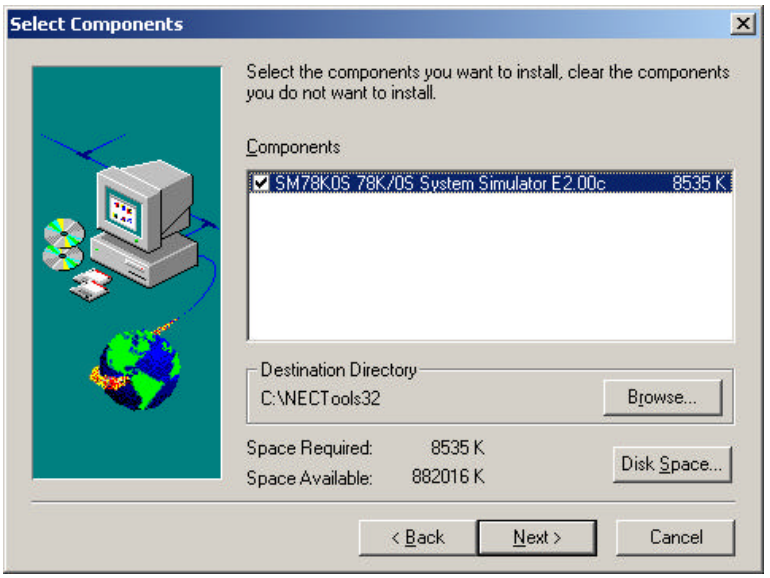

图 1-17 Select Components 对话框

选择要安装的组件。

点击 Browse 按钮，指定要安装的路径，默认路径是“C：\NECTools32”。点击 Next，出现 Select Program Folder 对话框。默认设置是“NEC Tools32 ”，也可以修改为其他的目录名。

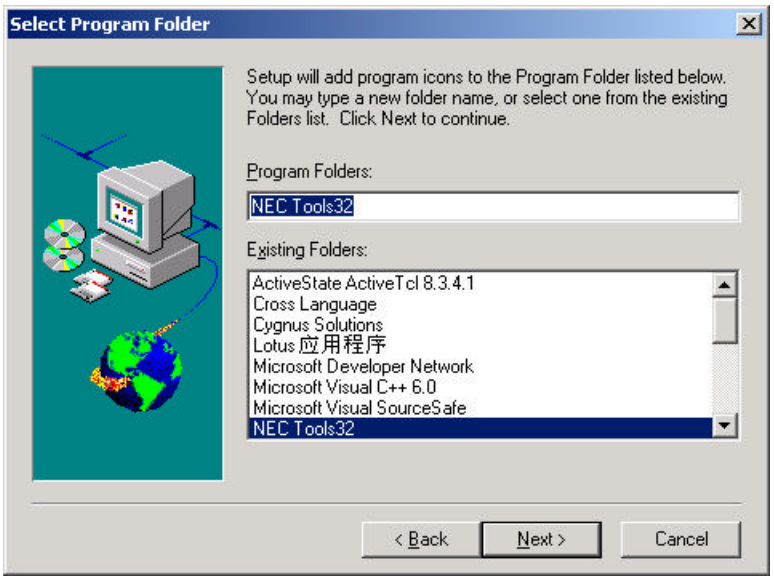

图 1-18 Select Program Folder 对话框

点击 Next，出现 Start Copying Files 对话框。检查当前的设置。

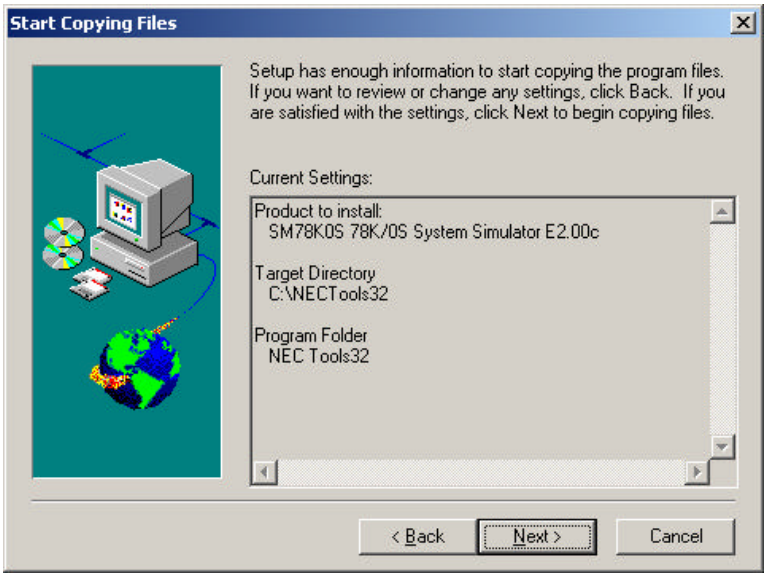

图 1-19 Start Copying Files 对话框

点击 Next 开始拷贝文件，等待拷贝完成。

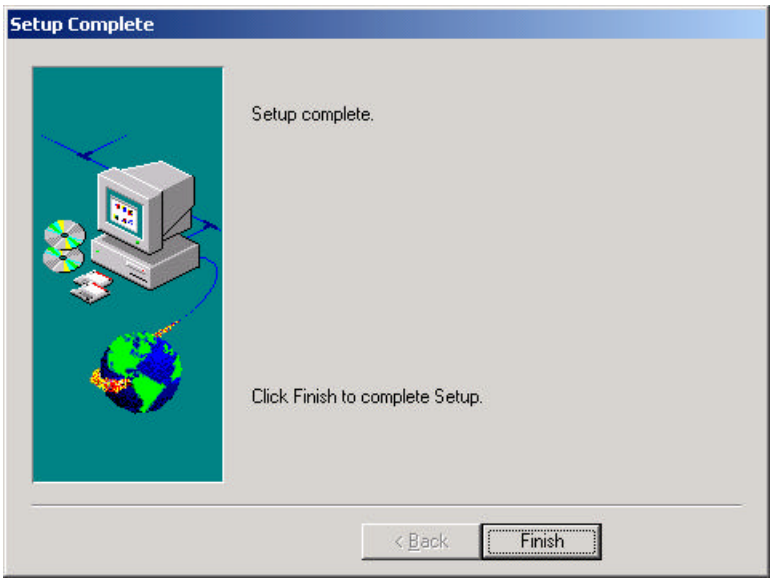

图 1-20 Setup Complete 对话框

出现 Setup Complete 对话框，点击 Finish 按钮，安装就完成了。

1.4 安装设备文件

1.4.1 简介

设备文件（Device file）是描述芯片信息的二进制文件。每一个设备文件描述了一个或一组芯片。

在编译程序的过程中，编译工具需要读设备文件，来得到相应芯片的内存和寄存器信息。

在用 PM plus 调试程序的过程中，PM plus 也需要读设备文件，来得到芯片的配置信息。

用户可以到以下网址下载各个芯片的设备文件：

http://www.necel.com/micro/ods/eng/tool/DeviceFile/list.html

1.4.2 安装步骤

在开始菜单中找到 NEC Tools32 中的 Device File Installer 图标，运行它，打开 Device File Installer 工具。

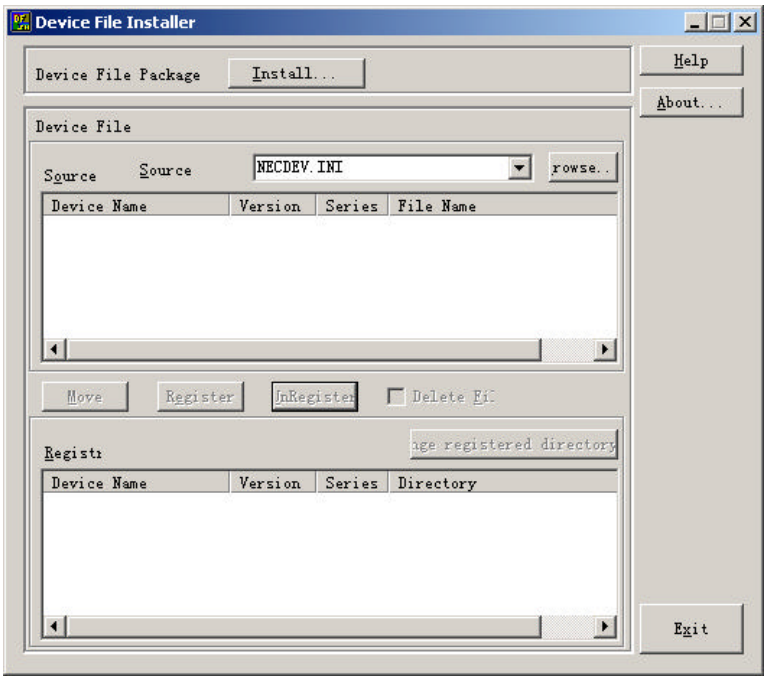

图 1-21 Device File Installer 界面

点击 Browse 按钮，选择设备文件（.78k）所在的路径。一般是 NECTools32 目录下的 dev 目录。

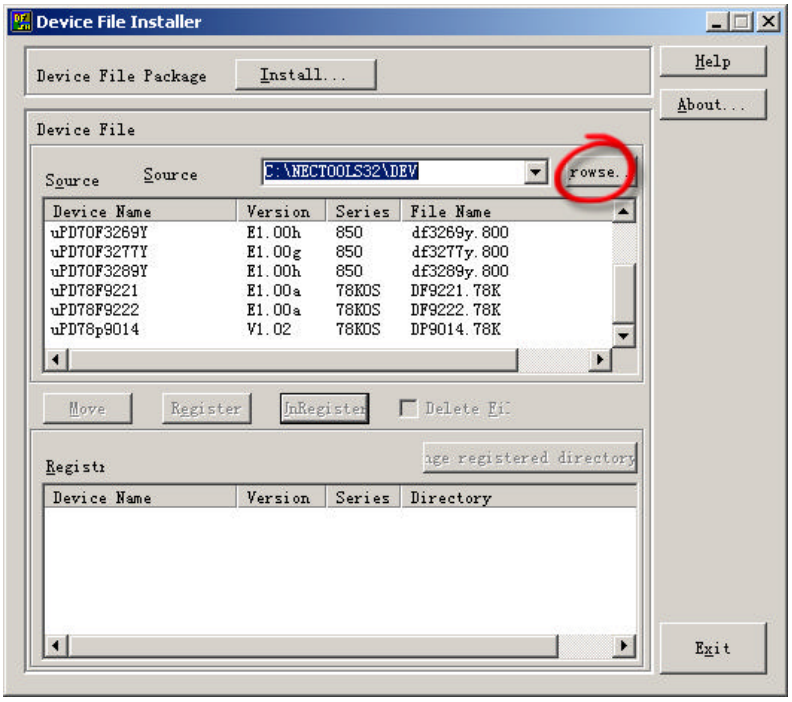

图 1-22 选择设备文件路径

在列表中选中一个或多个设备文件。

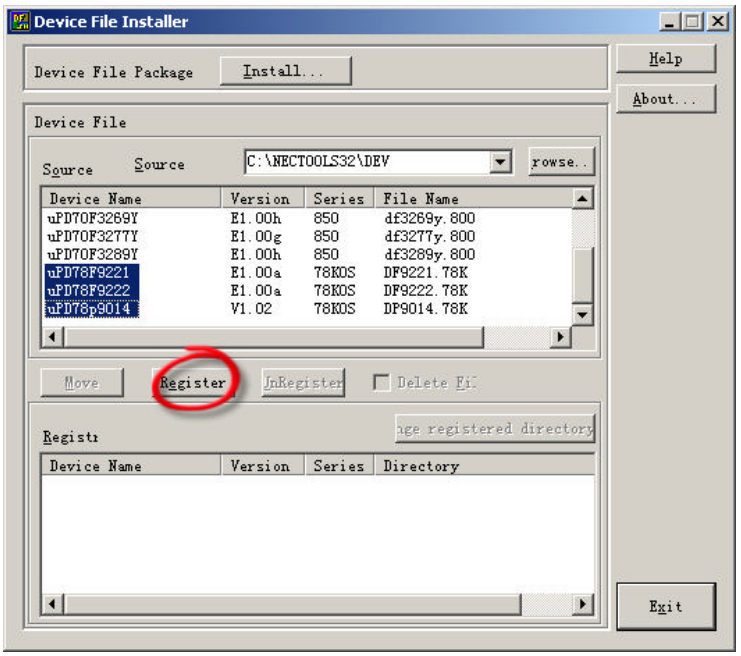

图 1-23　**选择设备文件**

点击 Register **按钮，执行安装。**

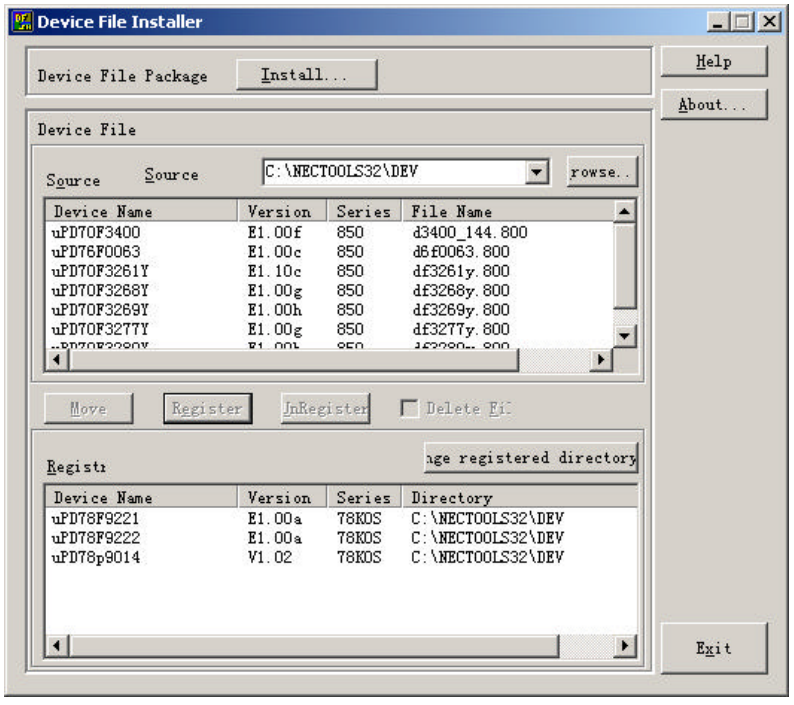

图 1-24　**设备文件安装完成**

第 2 章 应用范例

2.1 环境

本 demo 程序使用的芯片是 78k0s 系列的 78f9222。请参考手册《78K0S/KA1+ User's Manual》（文档编号 U16898E），以获得该芯片的信息。

本 demo 是专为演示 PM plus 和 SM plus 所做。编译 demo 需要 PM plus 5.10 以上版本。调试 demo 需要 SM plus 1.00 以上版本。

2.2 编译

本 demo 已经编译完成，可以直接运行。

如果用户希望重新编译，须先将 demo 目录拷贝到 NECTools32 目录下，因为编译时需要链接 NECTools32/lib78k0s 目录下的 s0s.rel 文件作为 startup 文件。

用户可以使用 PM plus 将它重新编译，操作如下：

用 PM plus 打开工程文件 demo_tp.prw。

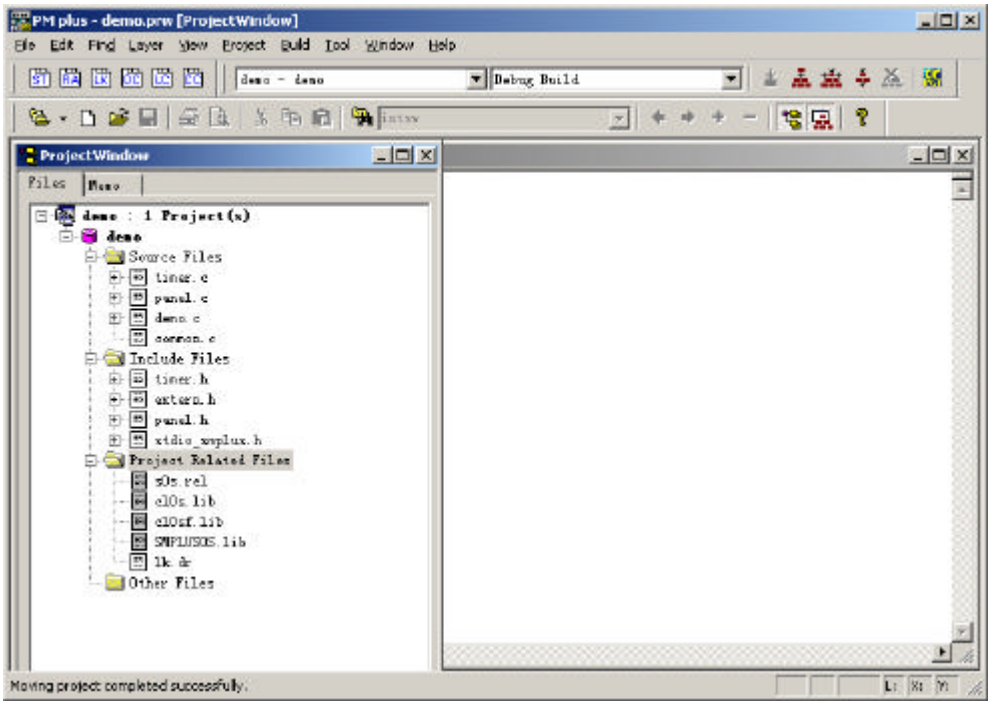

图 2-1 用 PM plus 打开工程

此工程中，所有链接和编译选项已经设置完成。用户只需点击“Rebuild”按钮即可。

2.3 运行

2.3.1 启动 SM78K0s plus

打开 SM78K0s plus 程序，在 chip 一栏选择 78F9222。如果下拉框没有 78F9222 的选项，说明该芯片的设备文件没有安装，请先安装设备文件。具体方法请参考 1.4 节。

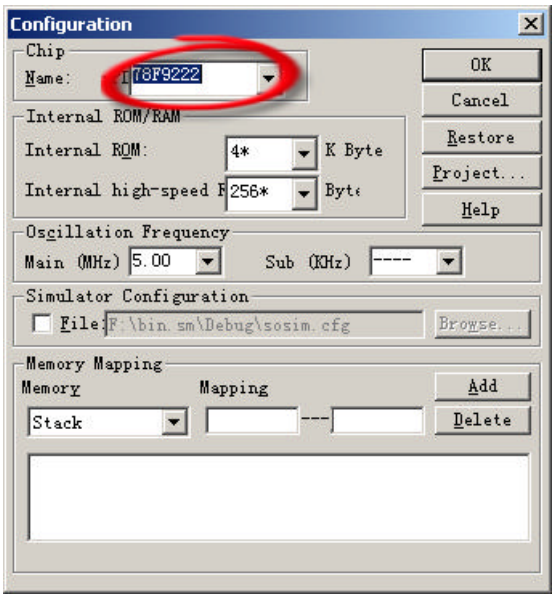

图 2-2 选择芯片

2.3.2 打开工程

用 SM78Kx plus 打开工程文件 demo.prj，可以看到以下的工程界面。

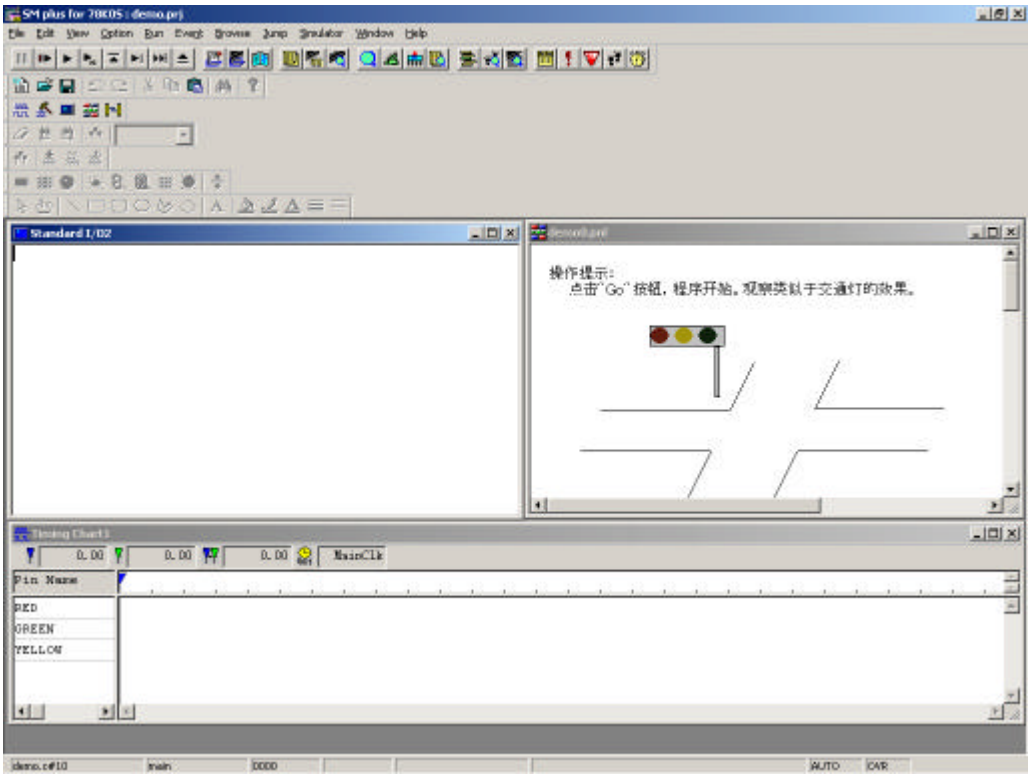

图 2-3 demo 工程界面

界面中各窗口作用如下：

1）I/O 面板

I/O 面板模拟了一些真实的器件，供用户搭建应用场景。

本 demo 的 I/O 面板中有 3 个控件：红、黄、绿灯。还有一些装饰线和方框，模拟了一个交通灯的场景。

图 2-4 I/O 面板

2）Standard I/O

Standard I/O 显示标准输出函数的输出。为标准输入函数提供输入界面。

在本 demo 运行时，程序会向 Standard I/O 窗口输出一些提示信息。

图 2-5 Standard I/O 窗口

3）时序图

时序图显示芯片的端口输出信息。

本 demo 中的时序图显示了连接到三盏灯的三个端口的输出情况。当某个端口输出值为高时，相应的灯就会亮；输出值为低时，相应的灯会灭。

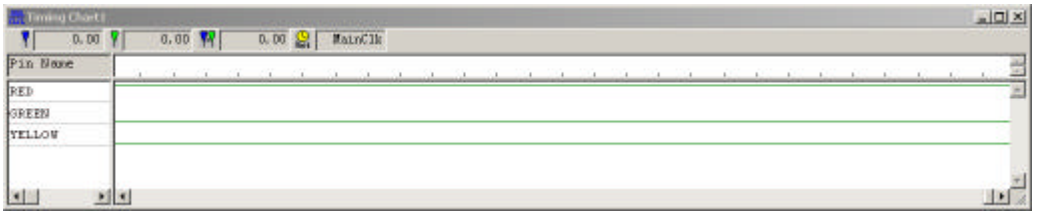

图 2-6 时序图

2.3.3 运行和观测

step1：

点击 SM78Kx plus 的 Go 按钮，运行程序。

可以观察到交通灯在工作，红灯和绿灯轮流亮/灭，黄灯会闪烁。

图 2-7 交通灯工作

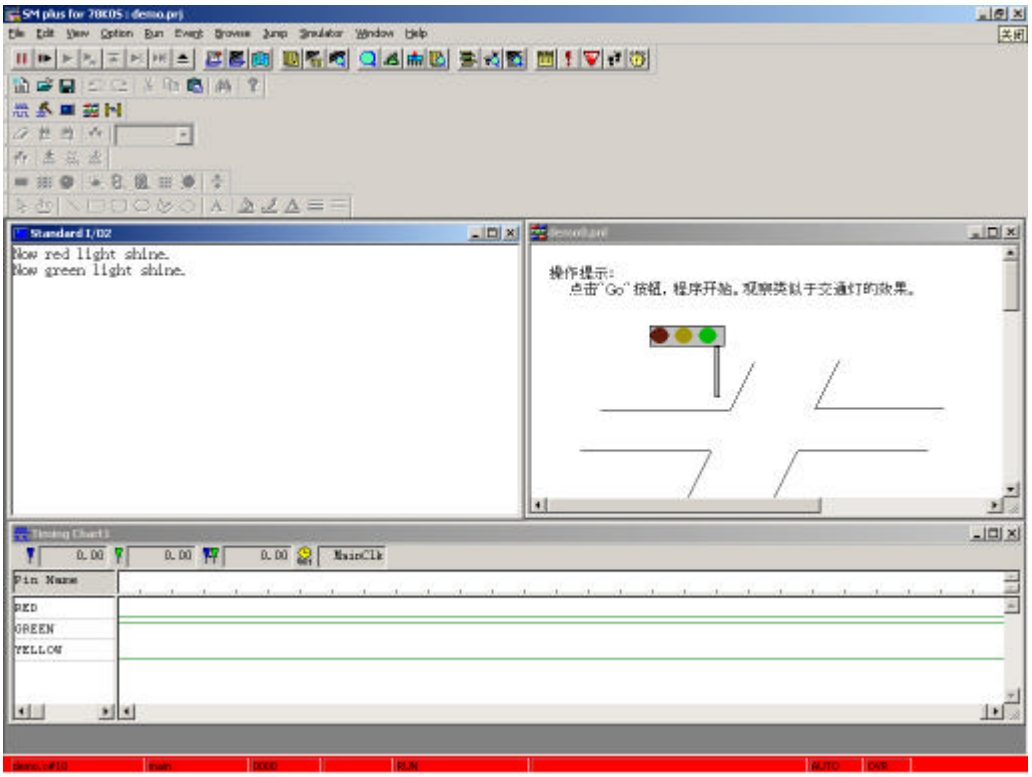

图 2-8 运行时的 demo 全图

第 3 章 NEC Tools 开发环境简介

本章将简要介绍手册中涉及到的 NEC Tools 开发工具。

3.1 概述

NEC Tools 开发环境包含了 NEC 系列微处理器（MCU）开发所需要的多种工具，如编译工具、链接工具、调试工具等。

本手册将介绍 NEC 开发工具中用于 78Kx 系列 CPU 的编译和仿真调试工具。

3.2 集成编译环境 PM plus

见第 4 章。

3.3 仿真调试环境 SM plus

见第 5 章

3.4 C 编译器 cc78Kx

cc78Kx 已经集合到 PM plus 平台中，用户可以使用 PM plus 编译工程，也可以抛开 PM plus，直接使用 cc78Kx 的命令行方式编译工程。

cc78Kx 它读入 C 源文件，编译生成二进制文件，或者是汇编文件。

cc78Kx 的输入文件包括：

file.c C 源文件

cc78Kx 的输出文件包括：

file.asm 汇编源文件

模块文件

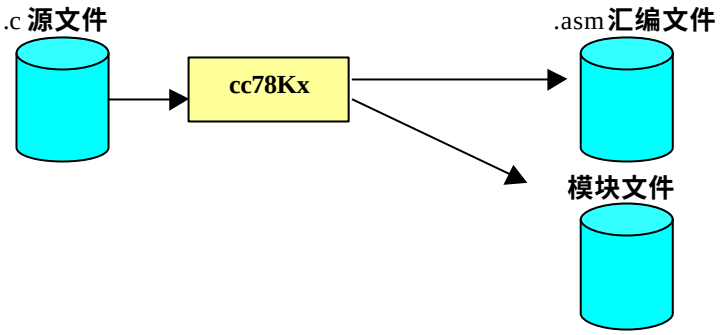

图 3-1 cc78Kx 工作示意图

3.5 汇编编译器 ra78Kx

ra78Kx 已经集合到 PM plus 平台中，用户可以使用 PM plus 编译工程，也可以抛开 PM plus，直接使用 ra78Kx 的命令行方式编译工程。

ra78Kx 读入汇编源文件，编译成目标文件，目标文件中的段和变量待定位，直到它们通过链接工具链接之后，才分配确定的内存地址。

ra78Kx 的输入文件包括：

file.asm　　汇编文件

as78Kx 的输出文件包括：

file.rel　　目标文件

file.rrn　　列表文件

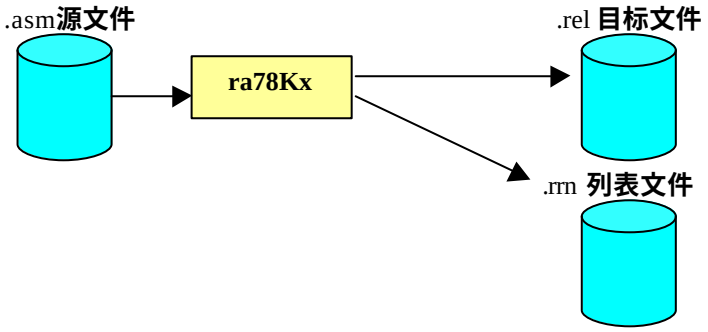

图 3-2　ra78Kx 工作示意图

3.6 链接器 lk78Kx

lk78Kx 已经集合到 PM plus 平台中，用户可以使用 PM plus 链接工程，也可以抛开 PM plus，直接使用 lk78Kx 的命令行方式链接工程。

一般情况下，用户程序由多个源文件组成。当用 cc78Kx 和 ra78Kx 来处理源文件以后，已经生成了多个目标文件（.rel），目标文件中的段和寄存器地址未定。此时用 lk78Kx 链接它们，生成一个可执行的模块文件（.lmf）。

除了源文件，lk78Kx 在链接过程中还需要读入段链接文件（Directive file，扩展名.dr），以获取段地址和大小信息；读入设备文件（Device file，扩展名.78k），以获得特殊寄存器和芯片内存信息；读入库文件（Library file，扩展名.lib），以获得库函数代码。

lk78Kx 的输入文件包括：

file.rel　　目标文件

file.dir　　段链接文件

file.78k　　设备文件

file.lib　　库文件

lk78Kx 的输出文件包括：

file.lmf　　可执行模块文件

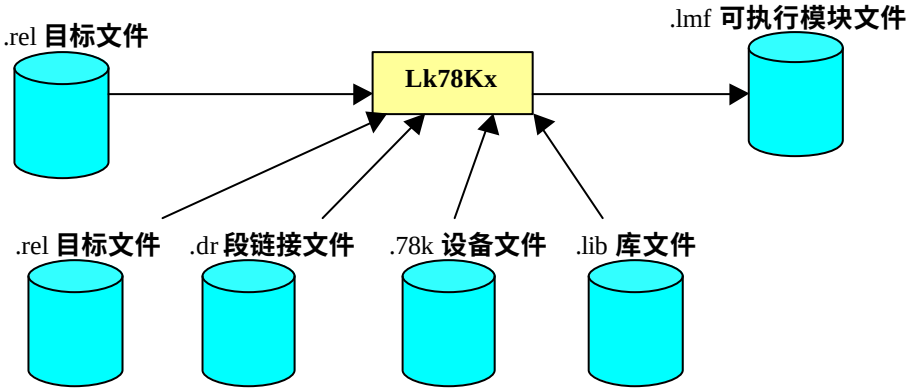

图 3-3　lk78Kx 工作示意图

3.7　目标文件转化器　oc78Kx

用户可以调用 oc78Kx 来转化可执行模块文件（.lmf）格式为可执行文本文件（.hex），并且可以输出符号表文件（.sym）。

oc78Kx 支持以下 hex 标准格式：Intel 格式。

oc78Kx 的输入文件包括：

file.lmf　　可执行二进制文件

oc78Kx 的输出文件包括：

file.hex　　可执行文本文件

file.sym　　符号表文件

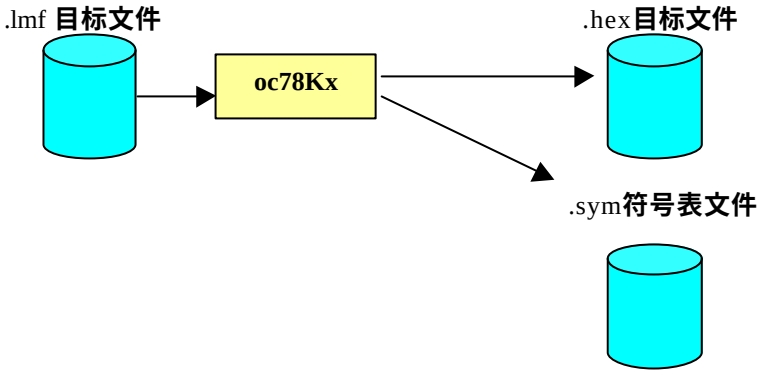

图 3-4　oc78Kx 工作示意图

3.8　库生成器　lb78Kx

lb78Kx 已经集合到 PM plus 平台中，用户也可以抛开 PM plus，直接调用 lb78Kx 来将多个二进制文件（.rel）链接为库文件。

lb78Kx 的输入文件包括：

file.rel　　目标文件

lb78Kx 的输出文件包括：

file.lib　　库文件

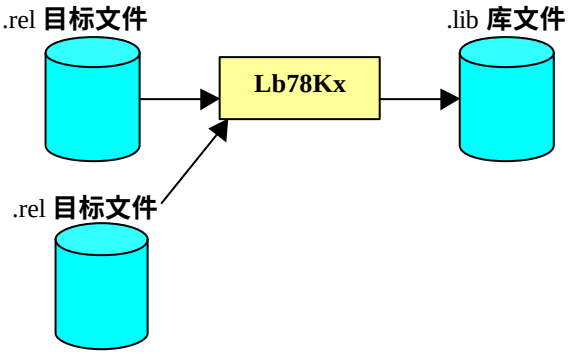

图 3-5　lb78Kx 工作示意图

第 4 章 集成开发环境

本章介绍如何用 PM plus 编译用户程序。

4.1 使用 PM plus 编译

4.1.1 启动 PM plus

在开始菜单的程序里找到 NEC Tools 的 PM plus 图标，并执行。将会弹出如下界面。如果不是第一次执行，PM plus 会自动加载最后一次执行的 Workspace。

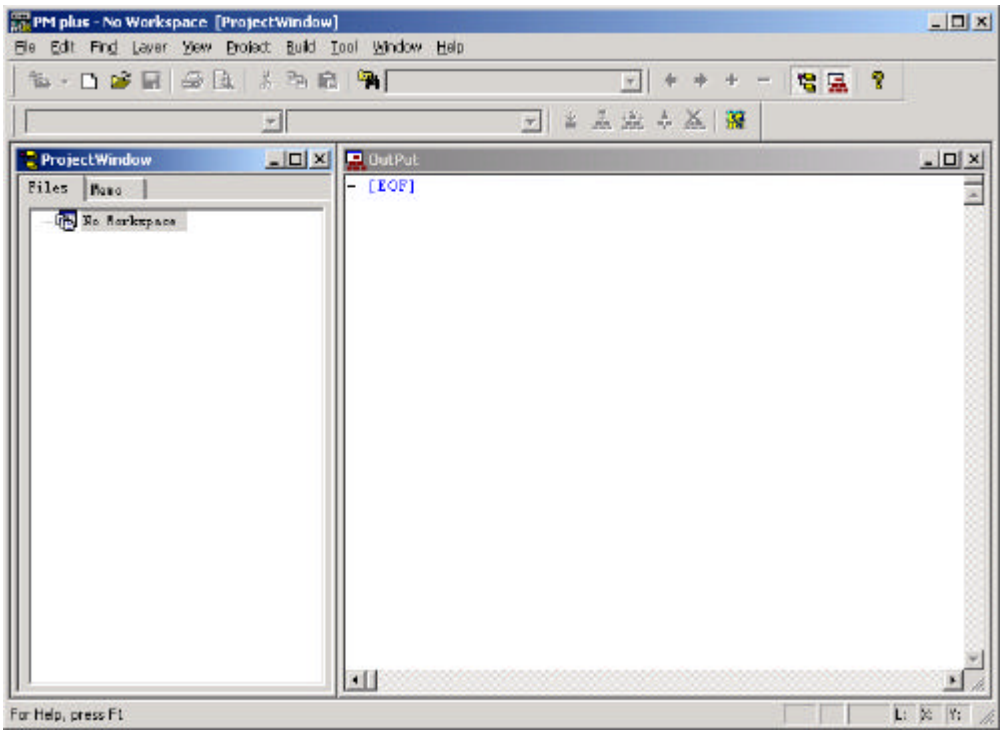

图 4-1 启动 PM plus

4.1.2 工程术语

Project：

Project 是被 PM plus 管理的一个单元，指在 PM plus 环境下开发的一个应用系统。PM plus 将每一个应用系统用到的源文件、目标设备名称、模块的工具选项以及编辑器或调试器等信息保存到一个项目文件（*.prj）里。编译或调试都是在项目单元里进行的，因此，要编译或调试的项目必须被设置为“当前项目(active project)”,可以通过选择 Project 菜单中的 Select Active Project…菜单项进行设置。

Workspace：

Workspace 是一个管理多个项目（project）文件的单元，PM plus 把这些项目文件的文件名保存到一个 workspace 文件（*.prw）中。

Project Group：

多个已注册的项目可以和相关的项目组成一个组，这个组就是 Project Group**。注意一个项目组中所有要注册的项目必须是使用同一个设备的项目。**

IDL file：

IDL file **是一个保存** PM plus **层次关系的文件。**

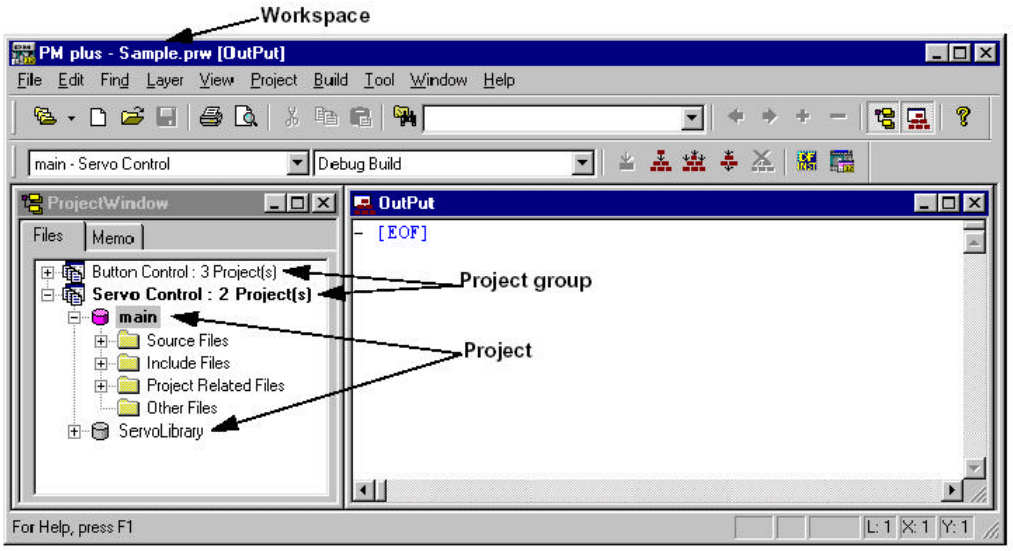

图 4-2 PM plus **主窗体**

4.1.3 创建 Workspace

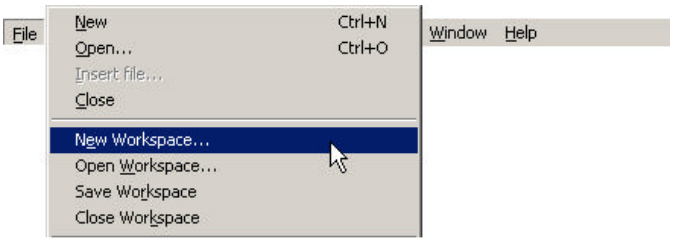

图 4-3 **新建** Workspace

要使用 PM plus **管理项目，之前必须先建立一个** Workspace**。选择** File **菜单中的** New Workspace...**，将打开创建对话框。**

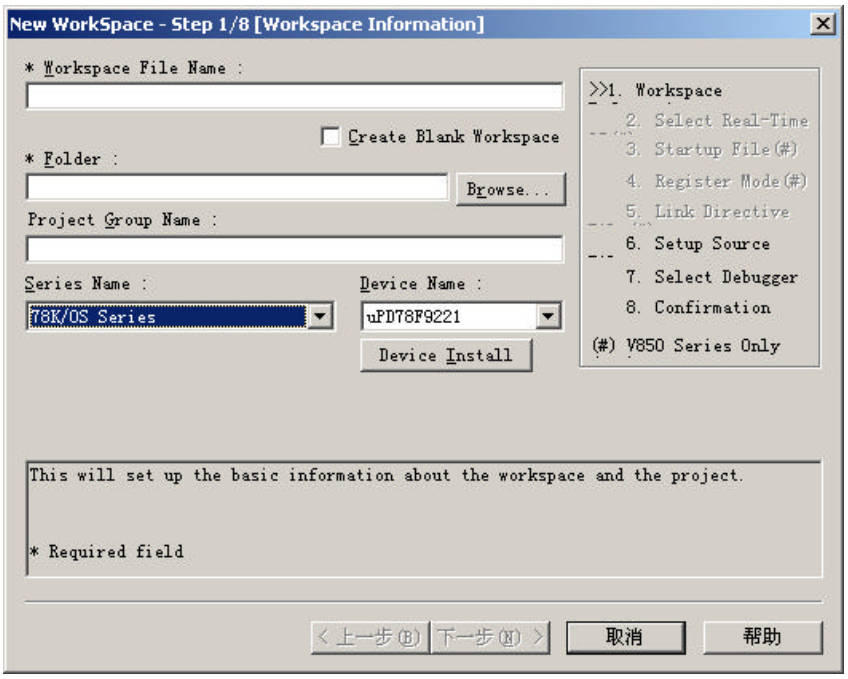

图 4-4 New WorkSpace – Step 1/8 对话框

Step1/8 对话框中设置如下内容：

Workspace File Name 创建的 Workspace 名字。这个名字将显示在创建好的 Workspace 图标上。

Folder 所有工程文件的存放目录。

Project 将加入的 Project Group 名字。

Series Name 新 Project 的 CPU 类型。下拉菜单提供已安装的所有 CPU 类型。

Device Name 新 Project 对应的设备类型。下拉菜单提供已安装的所有设备类型。

填好设置，点击下一步按钮，出现创建对话框 Step6/8。

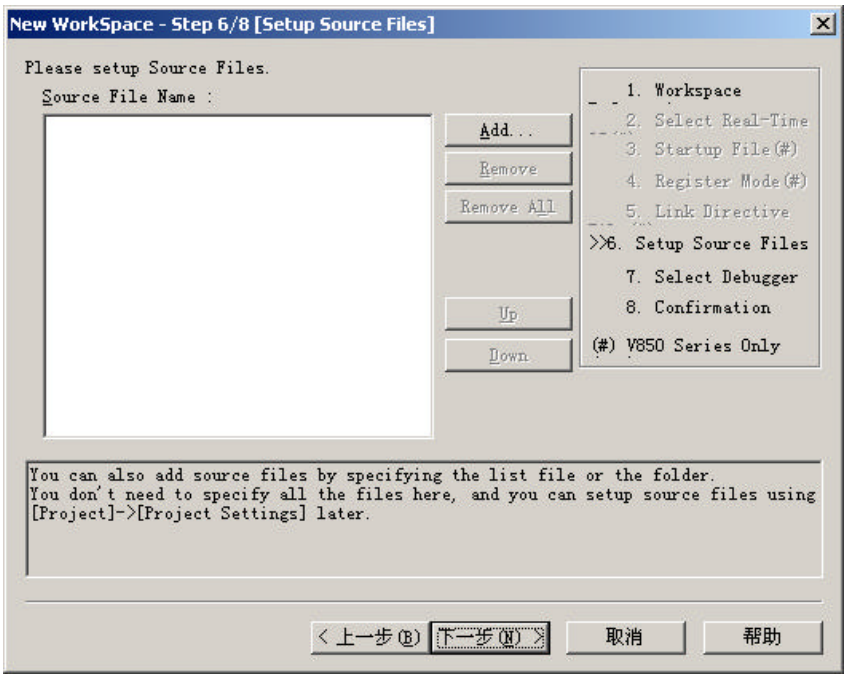

图 4-5 New WorkSpace – Step 6/8 **对话框**

对话框 Step6/8 **中添加源文件。点击“** Add **”按钮来添加源文件，可以是** C **或汇编文件。点击对话框下一步按钮，出现创建对话框** Step7/8。

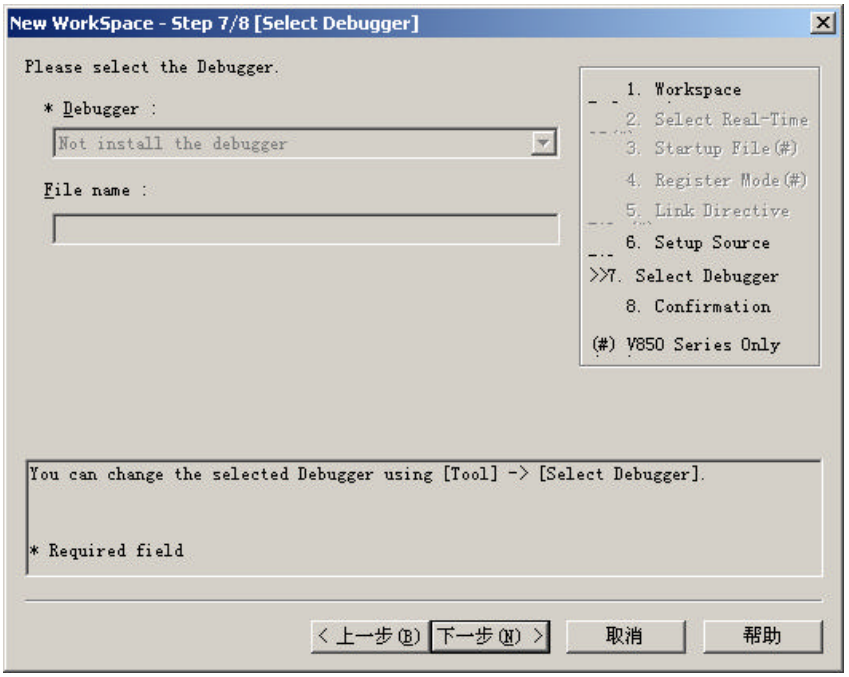

图 4-6 New WorkSpace – Step 7/8 **对话框**

对话框 Step7/8 **中选择调试器，下拉菜单提供了已安装的所有调试器。**
点击对话框下一步按钮，出现创建对话框 Step8/8。

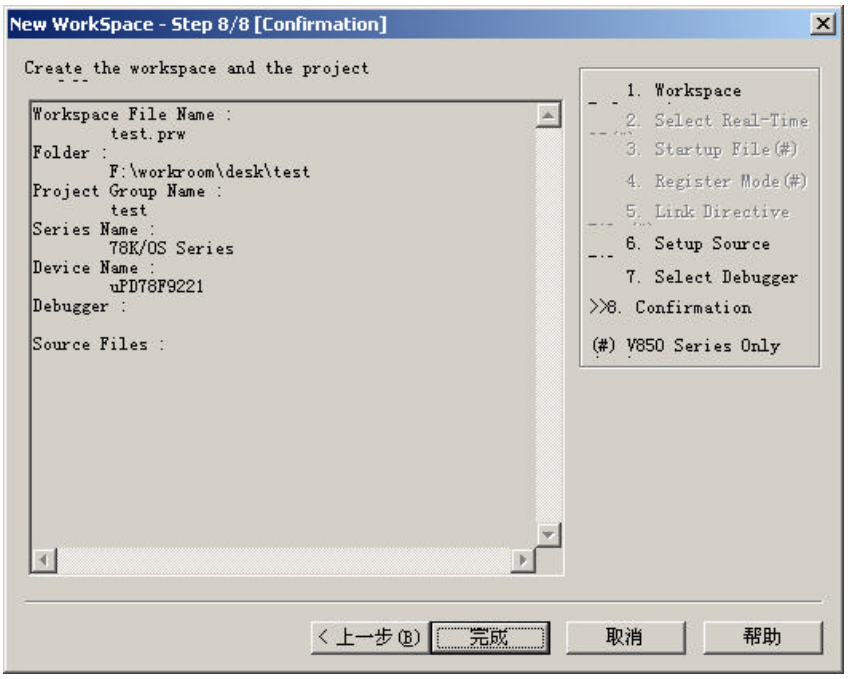

图 4-7 New WorkSpace – Step 8/8 对话框

对话框 Step8/8 是创建完成信息对话框。

点击完成，一个新的 WorkSpace 就建立好了。

4.1.4 读取 Workspace

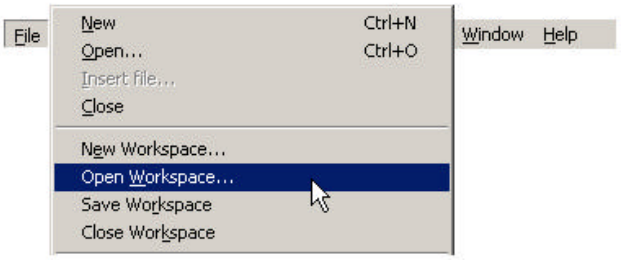

图 4-8 读取 Workspace

如果已有创建过的工程，选择 File 菜单中的 Open Workspace，打开 Open Workspace 对话框。

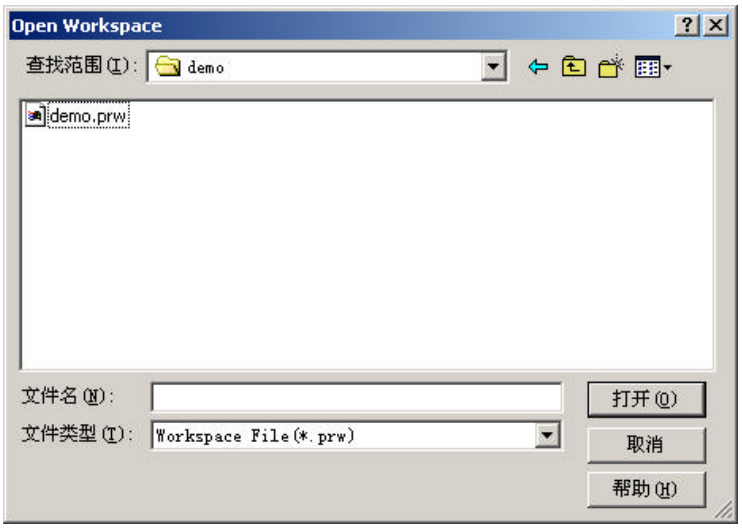

图 4-9 Open Workspace 对话框

选择 Wokspace 文件（*.prw），点击打开按钮，就可以读取相应的 Workspace。

4.1.5 Build工程（Project）

Build 一个工程，根据工程设置的不同，将产生二进制代码文件（*.lmf），或是库文件（*.lib）。

Build 工程只需点击 Build 按钮  ，或是从菜单中选择 Build->Build。

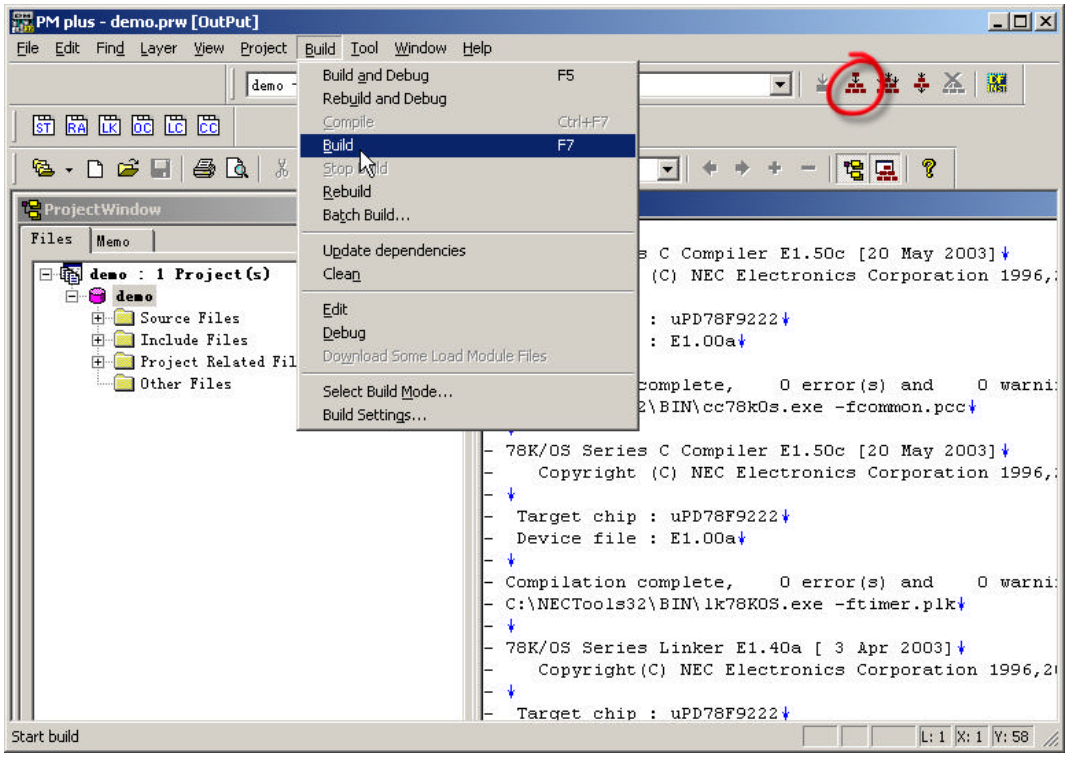

图 4-10 Build 工程

4.2 选项设置

PMplus 为高级用户提供了丰富的选项设置功能。比较重要的有编译选项、汇编选项和链接选项。

编译选项

选择 Tool 菜单中的 Compiler Options…，可以打开 Compiler Options 对话框。提供了编译、优化、预处理等选项的设置。

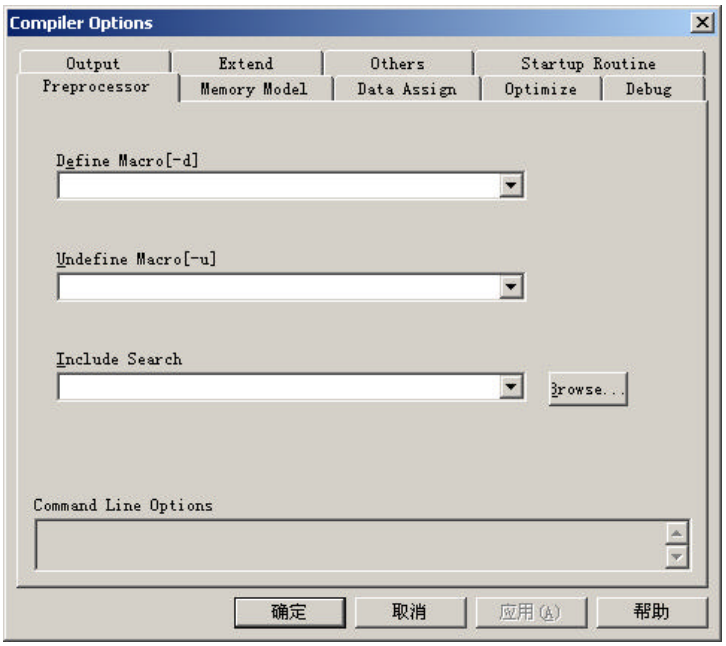

图 4-11 Compiler Options 对话框

汇编选项

选择 Tool 菜单中的 Assembler Options…，打开 Assembler Options 对话框。提供了编译汇编语言的扩展选项。

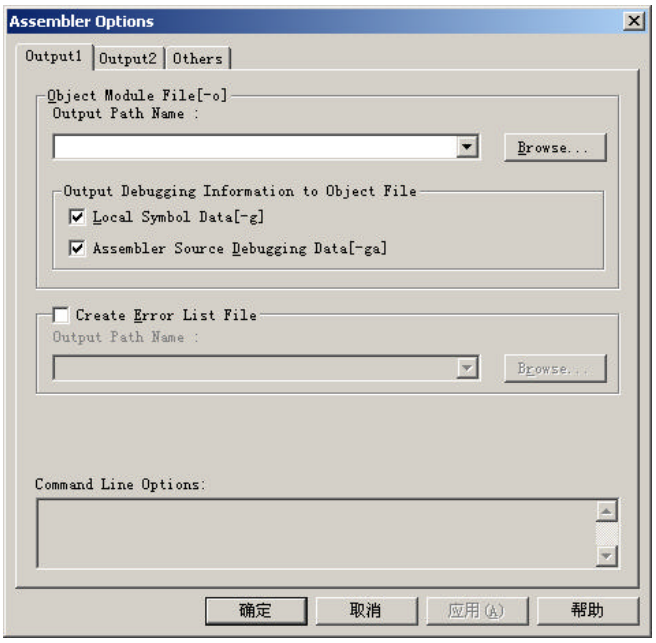

图 4-12 Assembler Options 对话框

链接选项

选择 Tool 菜单中的 Linker Options…，打开 Linker Options 对话框。提供了段链接文件、用户自定义库等的设置选项。

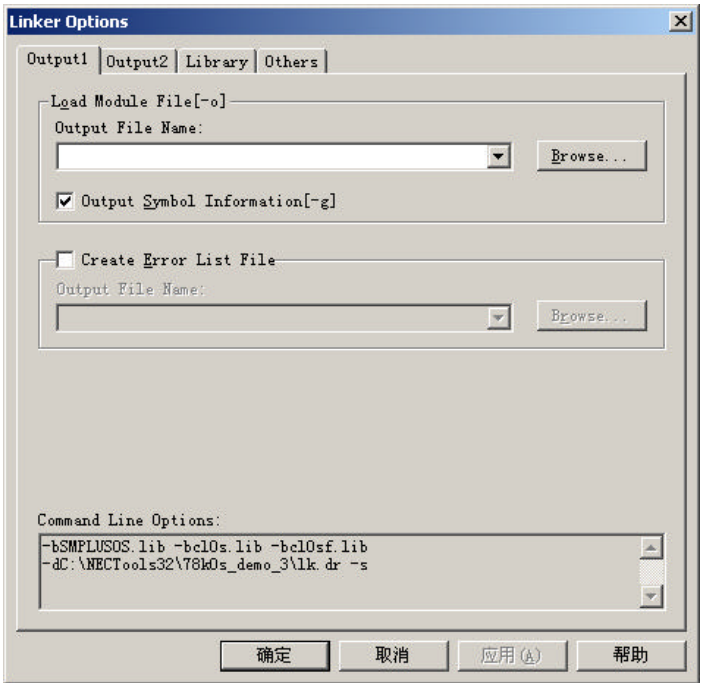

图 4-13 Linker Options 对话框

第 5 章 仿真调试环境

本章介绍如何用 SM78Kx plus 调试用户程序。

5.1 基本步骤

下面按顺序介绍调试的基本步骤。

5.1.1 启动

在开始菜单里找到 NEC Tools32 的 SM78Kx plus 图标，执行它，将启动 SM78Kx plus。

执行 SM78Kx plus 之后，自动弹出启动设置对话框，由用户配置 SM78Kx plus 执行时的环境。配置内容如下：

Chip SM78Kx plus 使用的芯片型号，下拉框提供了已安装的所有芯片。

Internal ROM/RAM 执行时芯片的内部 ROM 和 RAM 大小。默认值根据所选的芯片而变化。用户可以修改内存大小。

Oscillation Frequency CPU 的主时钟和副时钟的频率值。

Memory Mapping 外部内存设置，下拉框提供了可添加的内存类型。

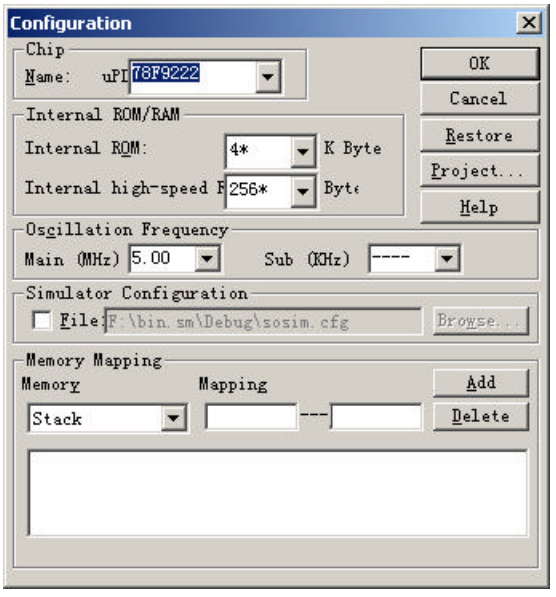

图 5-1 Configuration 对话框

设置完成后，点击 OK 按钮，将显示 SM78Kx plus 的主窗体。

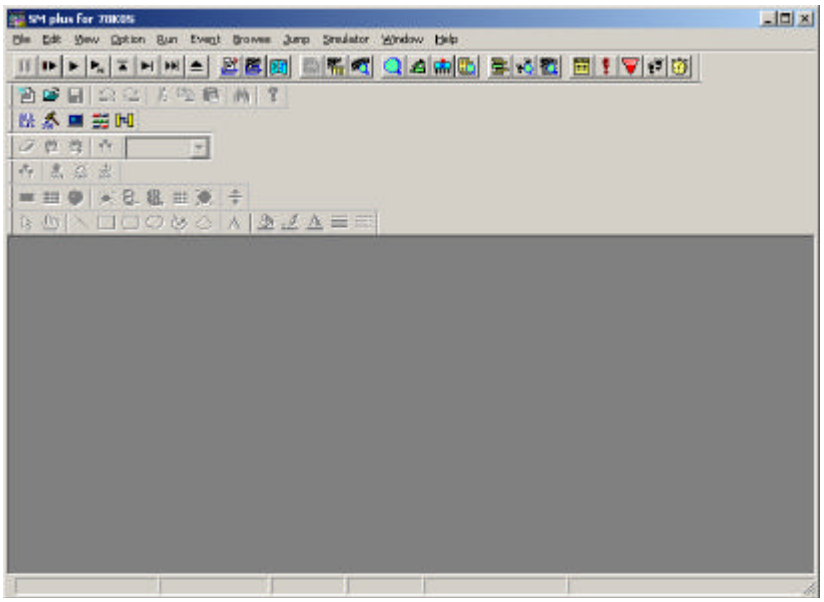

图 5-2 SM78Kx plus 主窗体

5.1.2 加载目标文件

目标文件就是由用户程序编译成的二进制文件，扩展名是*.lmf。关于编译目标文件的方法，详见第 4 章。

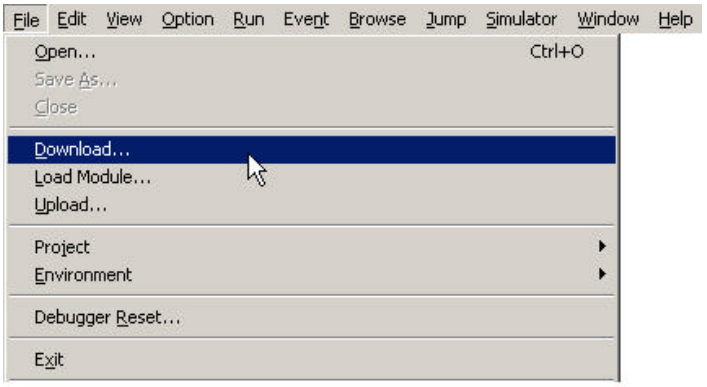

图 5-3 加载目标文件

要加载目标文件，选择 File 菜单中的 Download…，打开 Download 对话框。

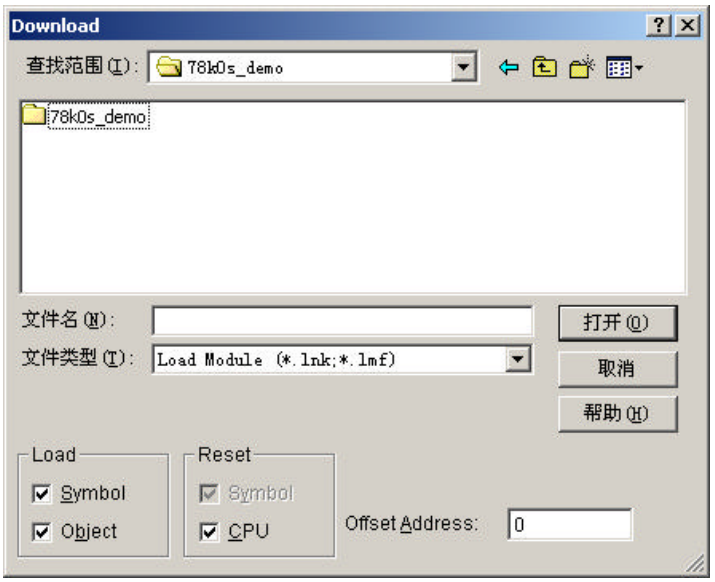

图 5-4 Download 对话框

选择要加载的目标文件（*.lmf），点打开按钮，该文件就被加载到 SM78Kx plus 中。

5.1.3 加载源文件

源文件是对应于目标文件的 C 和汇编源文件。

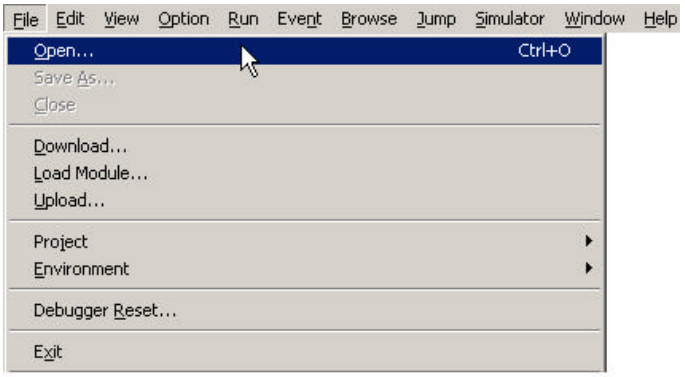

图 5-5 加载源文件

要加载源文件，选择 File 菜单中的 Open…，打开 Open 对话框。

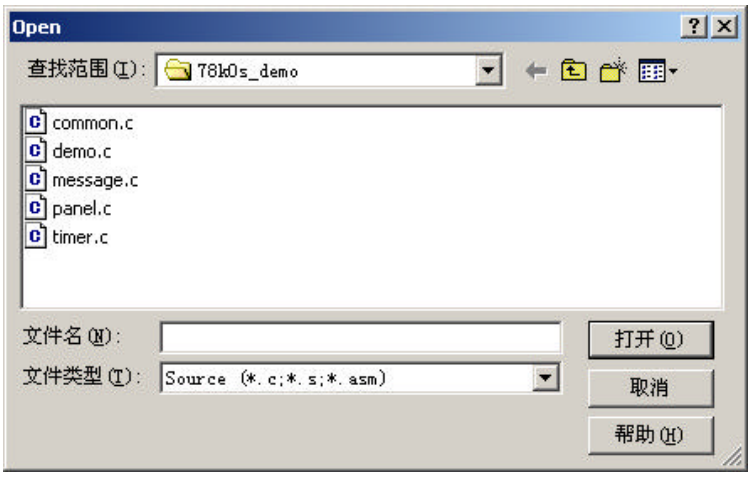

图 5-6 Open 对话框

选择要加载的源文件，点打开按钮，该文件就被加载到 SM78Kx plus 中。显示源代码窗口。

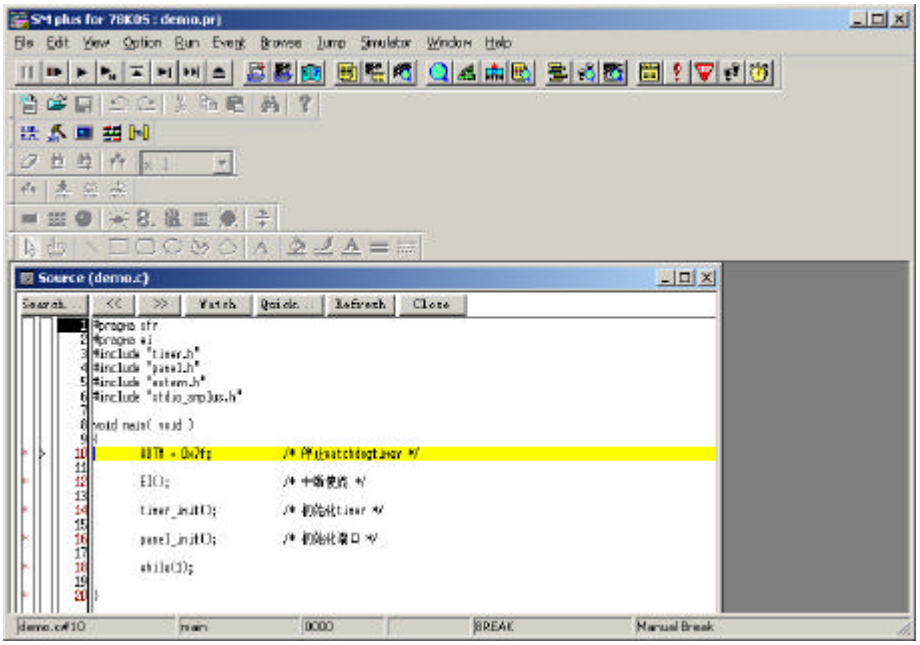

图 5-7 源代码窗口

在 SM78Kx plus 的源代码窗口中，右栏显示代码，左栏显示调试信息。

调试信息栏的数字和符号意义如下：

数字 表示行号。

***** 表示本行是指令行。可以在有“ * ”的行设断点。

> 表示当前程序执行的指令行。当前行显示为黄色。

5.1.4 加载工程文件

工程文件（*.prj）包含了调试一个工程的所有信息，包括断点、event、加载文件、窗口位置和各种设置。通过加载一个预先保存好的工程文件，可以方便的恢复上次调试的所有环境。

要加载工程文件，选择 File 菜单中的 Project -> Open，打开 Open 对话框。

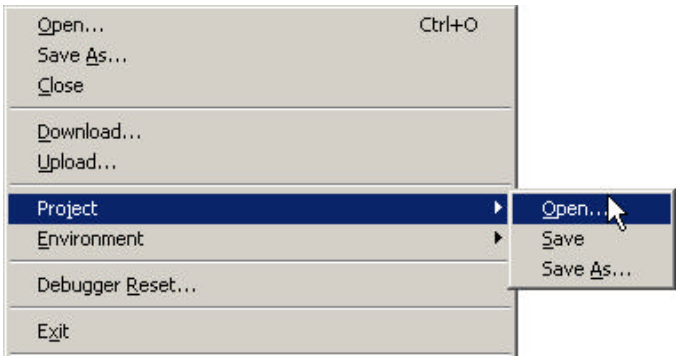

图 5-8 加载工程文件

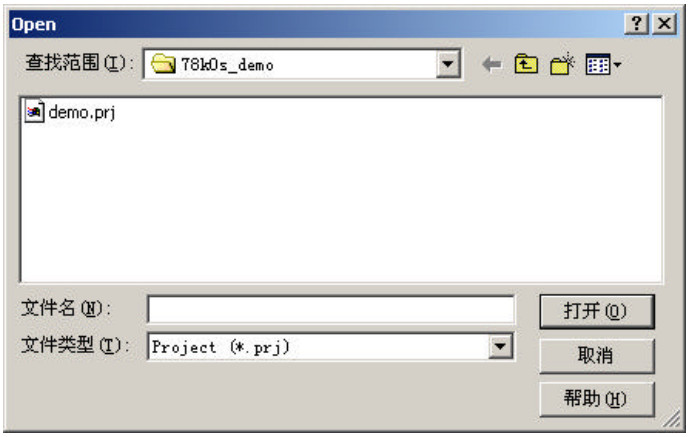

图 5-9 Open 对话框

选择要加载的文件名，点击打开按钮，该工程就被加载进来了。

5.1.5 在源文件中设置断点

可以设置断点的行号左边都有一个“*”标识。将鼠标箭头移到行号左边的“*”处，点击鼠标左键就在该行设置了断点。这样，在行号左边能看到一个红色的“B”，表示设置了一个断点，断点行显示为红色。

设置断点后，程序将在断点所在的指令行停止。

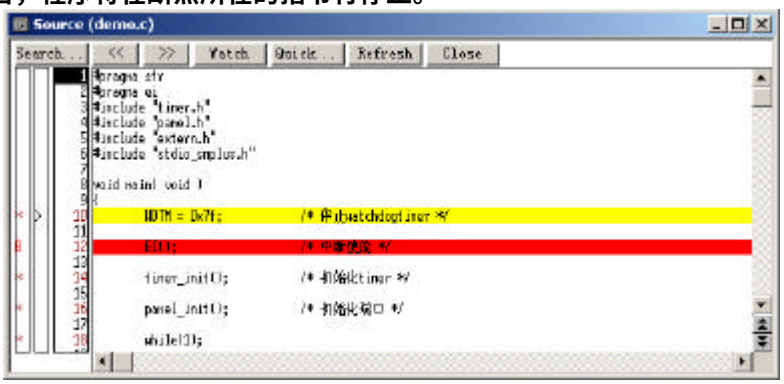

图 5-10 设置断点

5.1.6 执行程序

要执行程序，点击主窗体工具栏上的执行按钮  ，或是从 Run 菜单中选择 Go。

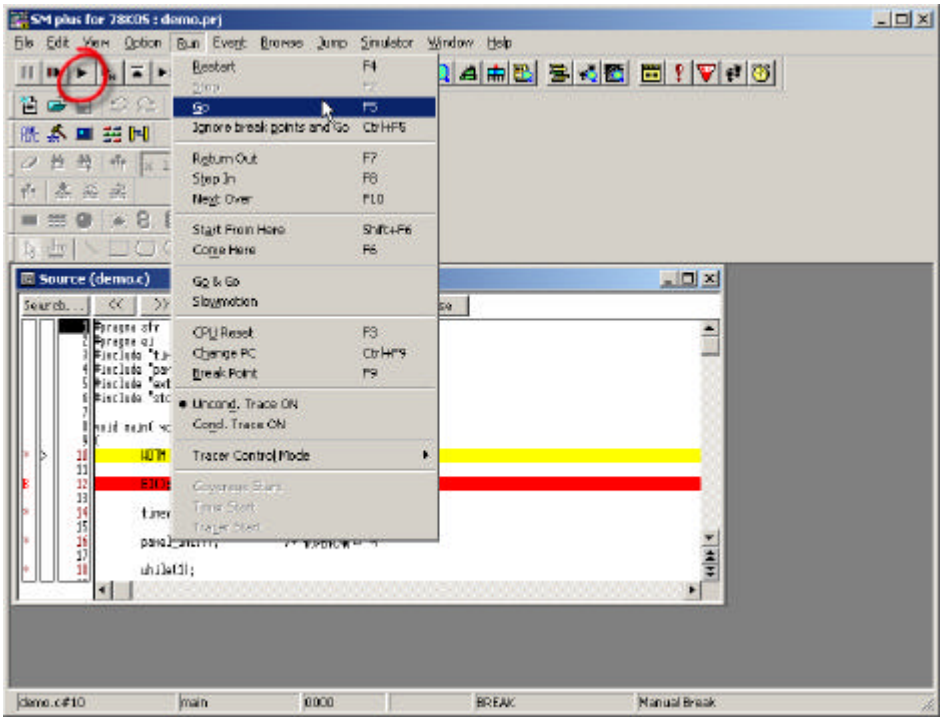

图 5-11 执行程序

程序执行到断点行将停止。要去掉断点，只需把鼠标箭头移到“ B ”上，点击鼠标左键即可。

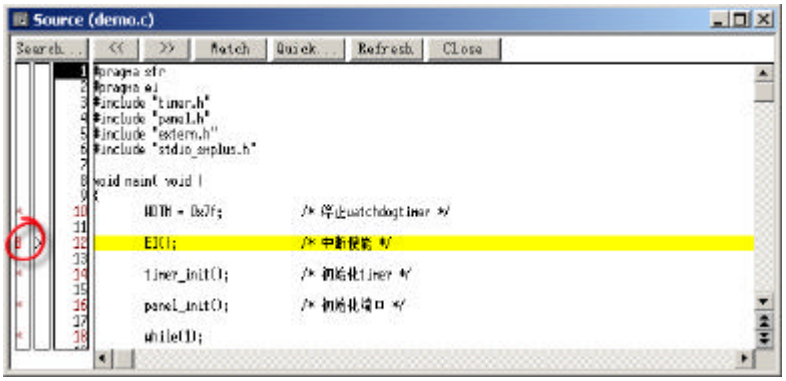

图 5-12 清除断点

5.1.7 单步执行

点击主窗体工具栏上的 Step In 按钮  ，或是选择 Run 菜单的 Step in，进行单步执行。

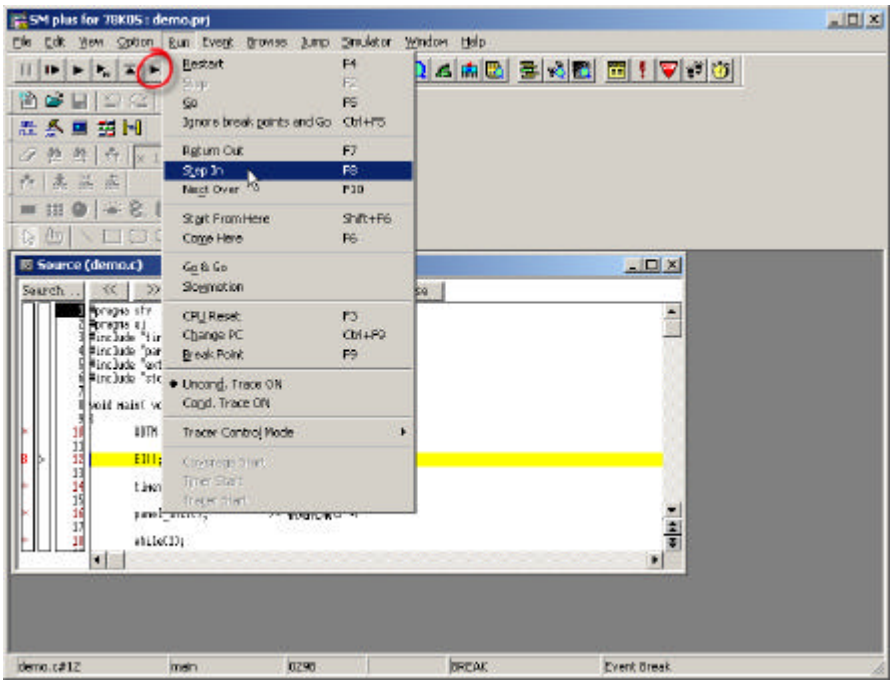

图 5-13　单步执行

Step In 按钮  把源文件的一行作为一步来执行，即便遇到一个被调用的函数，也是一一行地执行，进入函数的实现代码。

Next Over 按钮  也是把源文件的一行作为一步来执行，但执行一个被调用的函数时，整个函数作为一步，跳过函数实现代码。

5.1.8　停止执行

在程序执行期间，按停止按钮  ，或是选择 Run 菜单中的 Stop，可以停止程序的执行。再点击执行按钮  后，程序将继续从停止的位置执行。

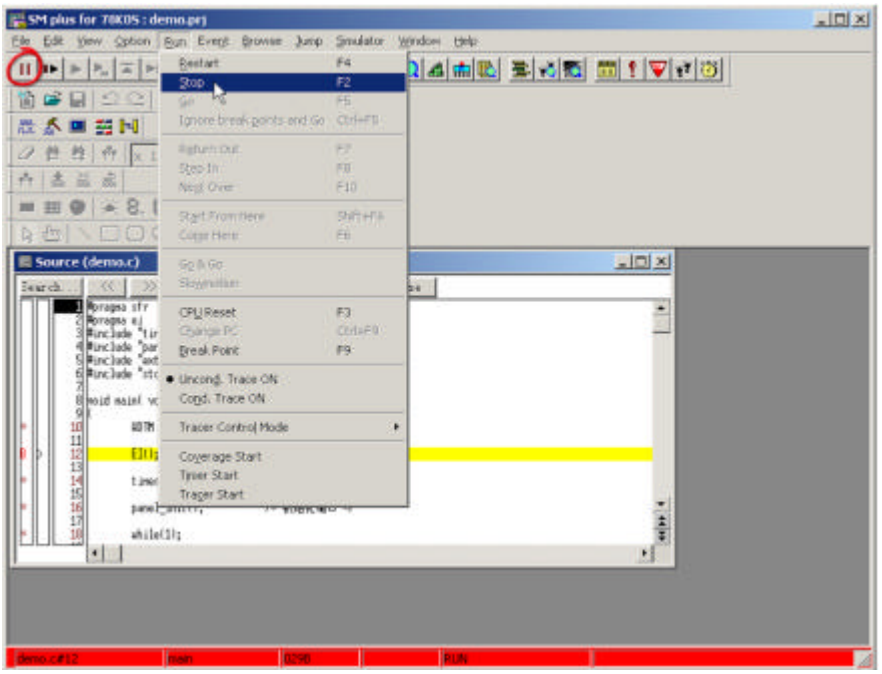

图 5-14 停止执行

5.1.9 重启

无论程序是否执行，都可以执行重启操作。重启的方法是按下重启按钮  ，或是选择 Run 菜单中的 Restart。

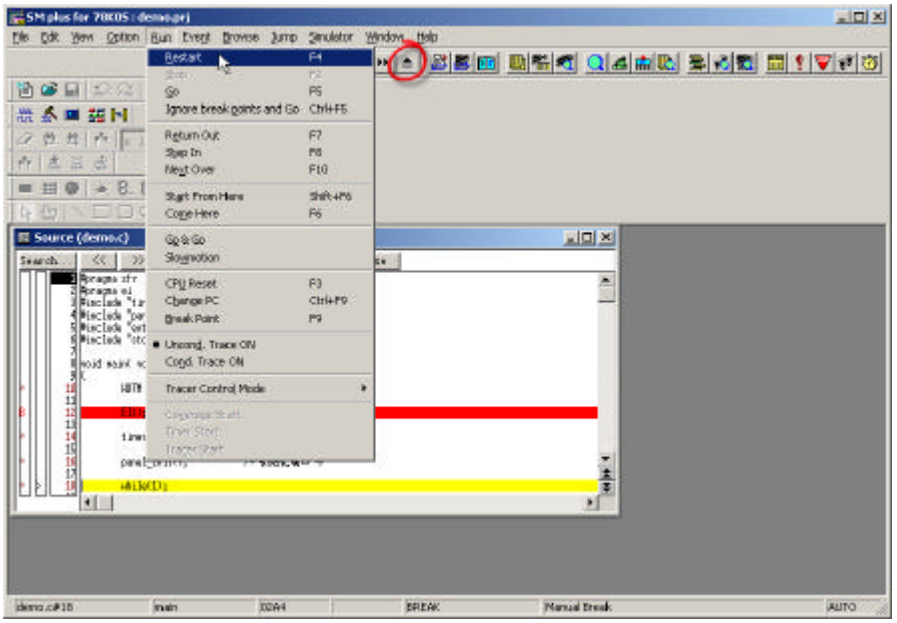

图 5-15 重启

重启操作将使 CPU 和各外围 Macro 都重启。所有通用和特殊寄存器恢复初始值，程序也从最初一行开始执行。

建议不要在程序运行时进行重启操作，而是先停止程序，后重启。

5.1.10 观察和修改变量值

开启 Watch 窗口：

点击主窗体工具栏上的 Watch **按钮**  **，或是选择** Browse **菜单的** Watch**，开启** Watch **对话框。**

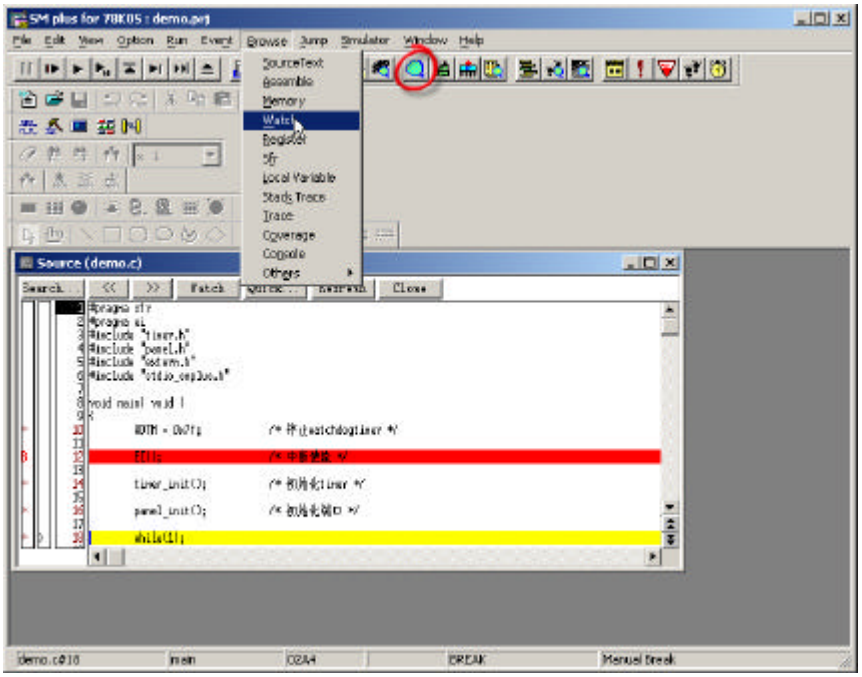

图 5-16 **打开** Watch **窗口**

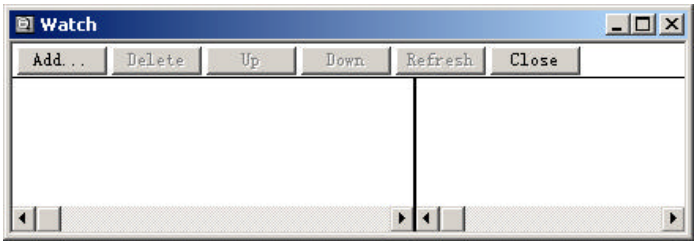

图 5-17 Watch **窗口**

第一次打开的 Watch **窗口里内容为空。**

观察变量：

点击 Watch **窗口的** Add…**按钮，弹出添加变量对话框。**

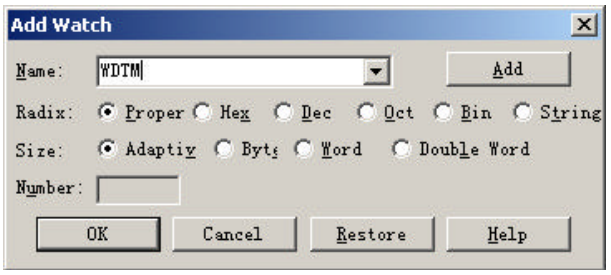

图 5-18 添加变量

在对话框的 Name 栏填入变量名，选择 OK，就可以观测到该变量的当前值。随着程序的执行，变量的值也会随时刷新。

图 5-19 观察变量

修改变量值：

在 Watch 窗口的显示/设置变量值区域，输入新的值按回车键，即可修改该变量的值。通过这种操作，可以在调试中很方便的测试变量的各种数值，而不必重新修改和编译程序。

被修改的数值显示为红色。

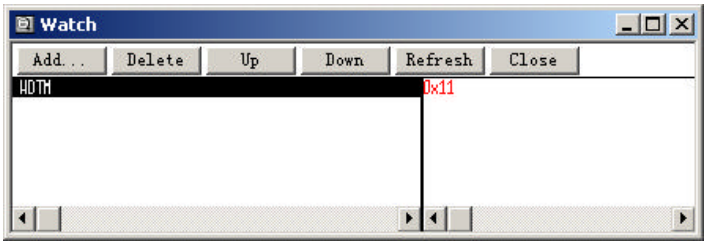

图 5-20 修改变量值

5.1.11 观察和修改寄存器值

开启寄存器窗口：

点击寄存器窗口图标  ，或是选择 Browse 菜单中的 Register，开启寄存器窗口。

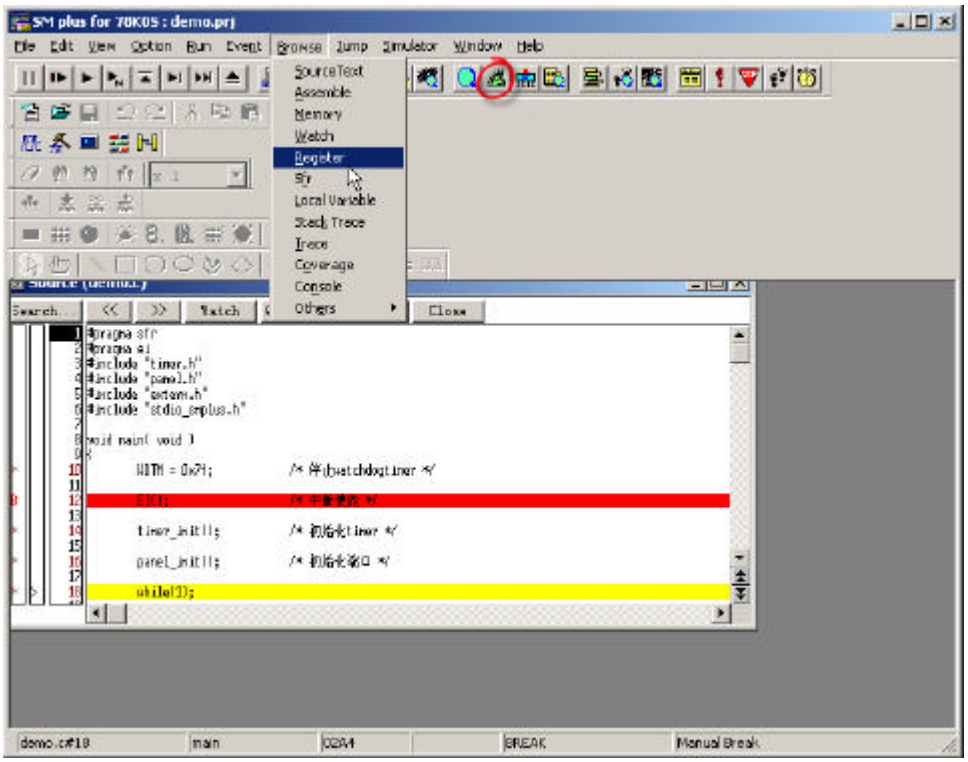

图 5-21　开启寄存器窗口

观察寄存器值：

双击带“+”的寄存器的名字，可以看寄存器各个位的值。

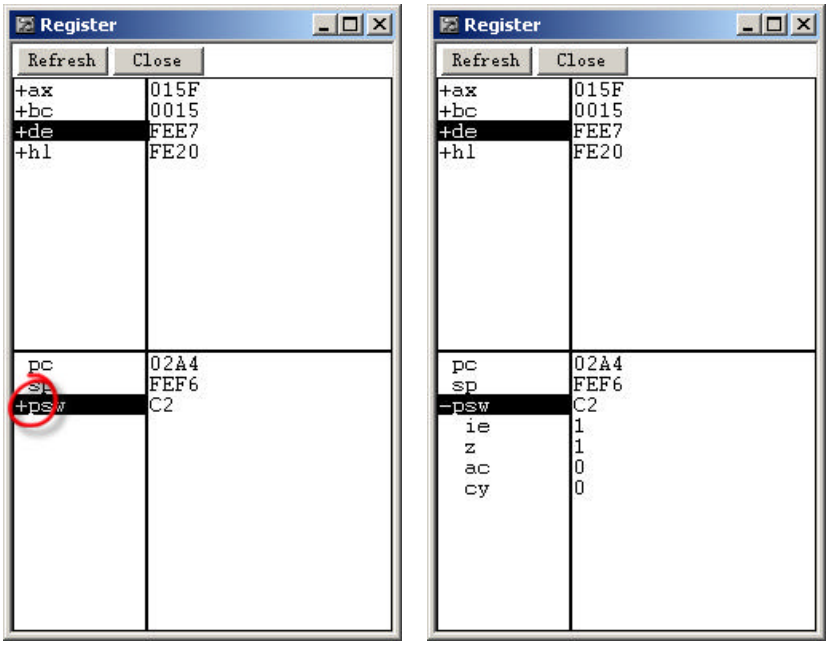

图 5-22　寄存器窗口

修改寄存器值：

在寄存器窗口，可以修改寄存器的值，修改后的值显示为红色，按回车键确认修改。

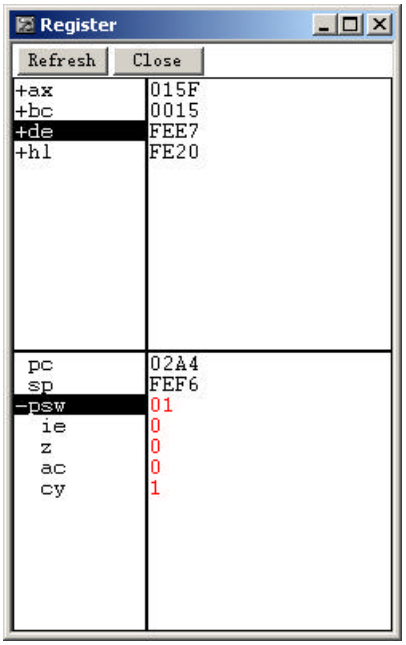

图 5-23 修改寄存器值

5.1.12 观察汇编代码

SM78Kx plus 可以显示程序的反汇编代码。点击汇编窗口按钮  ，或是选择 Browse 菜单中的 Assemble，打开汇编窗口。

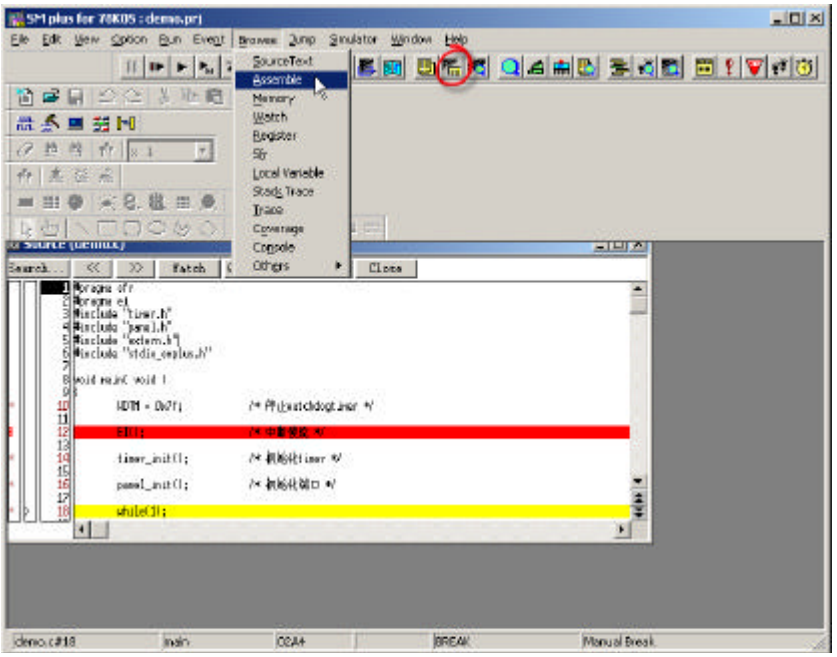

图 5-24 打开汇编窗口

与源代码窗口类似，左侧是调试信息栏，其中“*”符号表示汇编行，“>”符号表示当前执行指令行，数字表示指令地址。

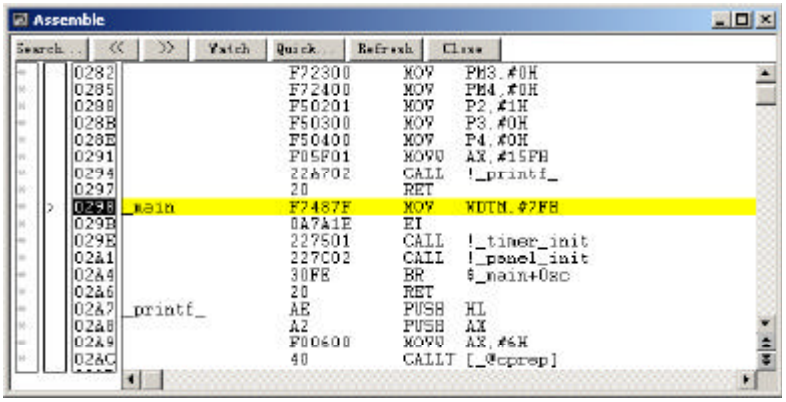

图 5-25 汇编窗口

5.1.13 设置汇编断点

在地址行左边的“*”区域点击鼠标左键，可以加入一个断点，断点处有一个红色“B”标识。被设为断点的行显示为红色。

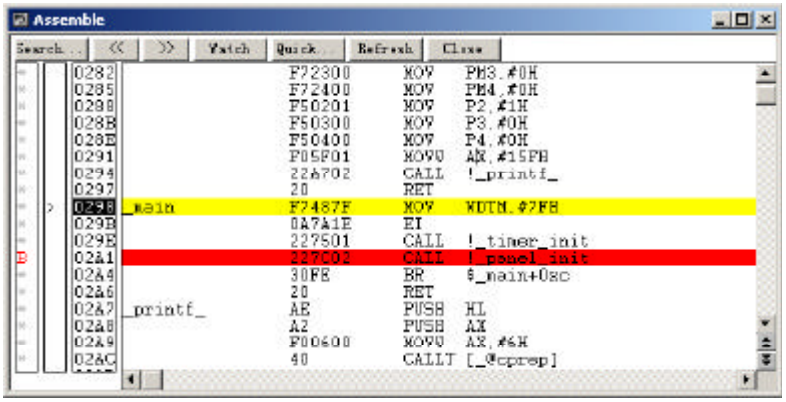

图 5-26 设置断点

点击主窗体工具栏上的  按钮执行，程序执行到断点时停止。标识了“B”的黄色一行表明程序执行的当前行。

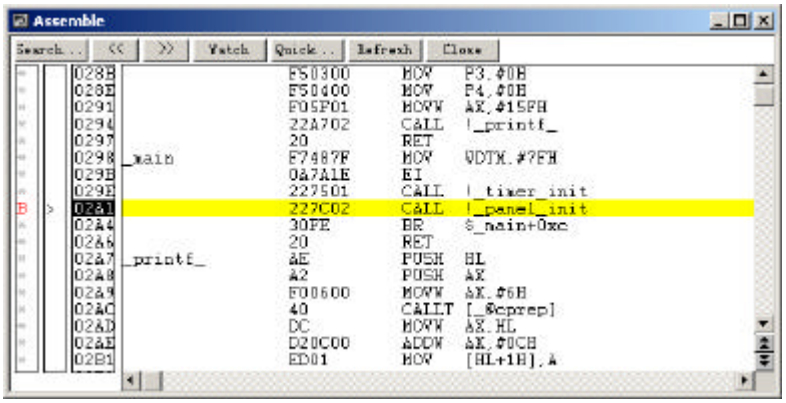

图 5-27 执行到断点

5.1.14 修改汇编代码

跟在 Watch 窗口修改变量值类似，在 Assemble 窗口可以修改指令，修改的指令为红色，按回车键确认修改。

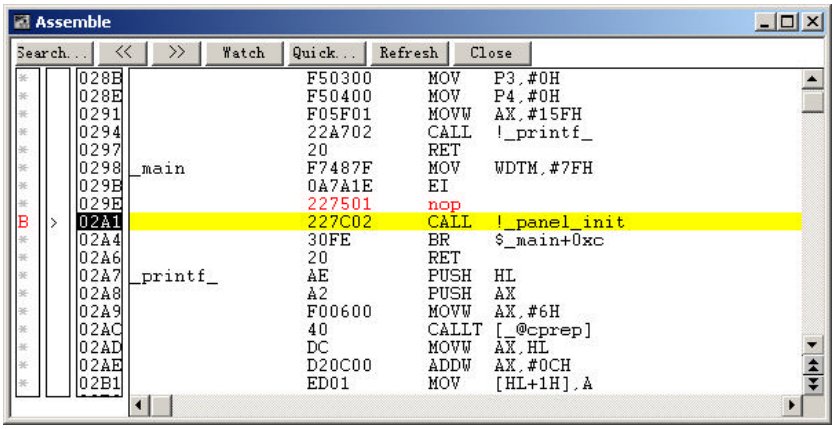

图 5-28 修改指令

5.1.15 退出 SM78Kx plus

要退出 SM78Kx plus，选择 File 菜单中的 Exit，出现退出提示对话框。

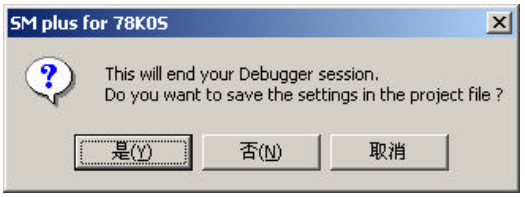

图 5-29 SM78Kx plus for 78Kx 对话框

说明是否保存 SM78Kx plus 当前调试环境设置为一个调试工程文件，如果点击 Yes，下次启动 SM78Kx plus 时，可以使用当前的环境设置。

5.2 高级调试功能

本部分内容是在 5.1 的基础上，介绍一些 SM78Kx Plus 提供的高级调试功能，可以帮助用户深入的调试程序，和检测程序性能。

5.2.1 Event（事件）

在 SM78Kx plus 中，Event 被定义成各种调试动作的触发条件。

基于 Event 可以设置断点，Event 还可以作为启动或停止 Trace 的条件等。通过 Event 管理器（Event Manager）对 Event 进行管理，设置 Event。也可以把几个 Event 组成一组作为一个条件，作为更复杂的触发条件。

5.2.1.1 Event 设置

选择 Event 菜单中的 EventManager，打开 Event Manager 窗口。

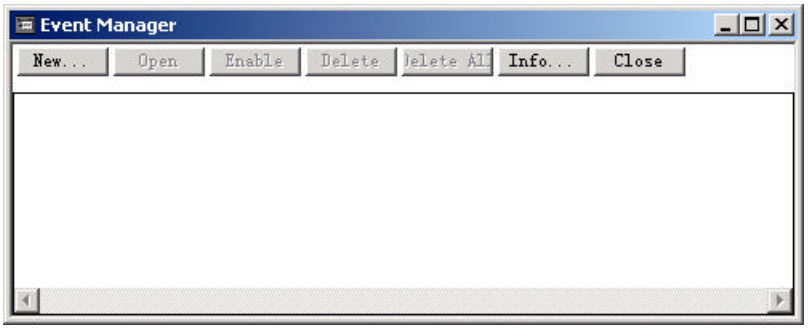

图 5-30 Event Manager 窗口

在 Event Manager 窗口中，点击 New…按钮，出现 New Event 窗口，选择 Event…，打开 Event 对话框。

图 5-31 New Event 对话框

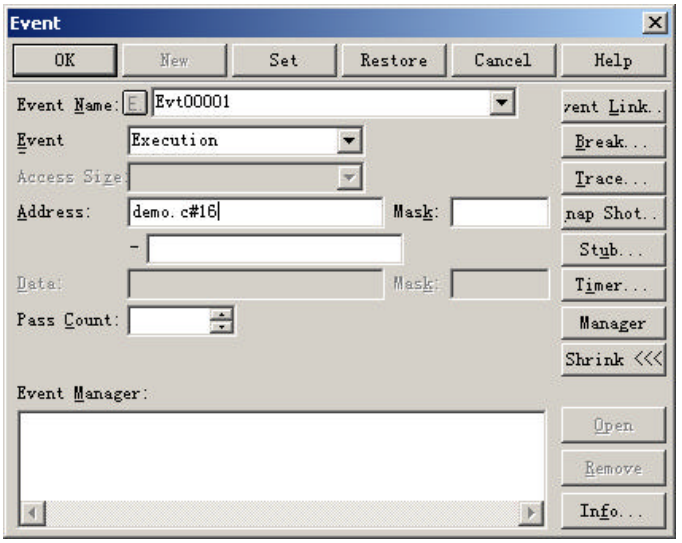

图 5-32 Event 窗口

设置内容为：

Event Name　　新的 Event 名。

Event	Event 的触发类型。
Access Size	传输数据的位宽条件。位宽不符合的传输，将不能触发 Event。
Address, Mask	地址范围条件，地址的 mask 值。
Data, Mask	传输的数据条件，数据的 mask 值。
Pass Count	条件重复次数。表示发生多少次触发条件，才触发事件。默认为 1，即第一次发生触发条件，就触发事件。

点击 Set 按钮，如果没有提示错误信息，Evt00001 就设置好了，可以在 Event Manager 窗口看到 Evt00001。如果 Source 窗口或 Assemble 窗口打开，在设置了事件的行前有“E”标识。

最多可以设置 256 条事件。

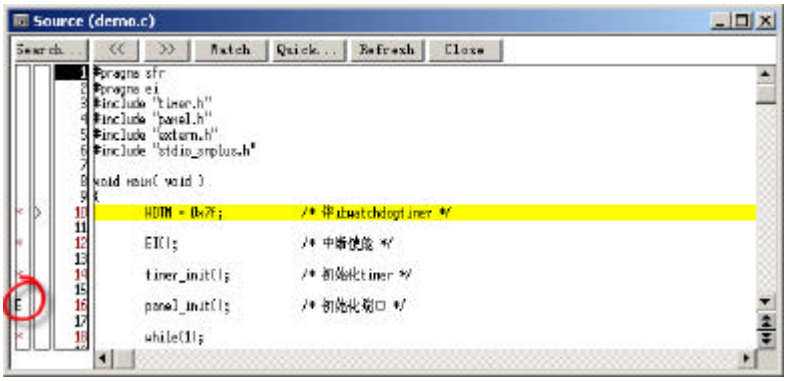

图 5-33 Source（demo.c）窗口 （设置事件）

5.2.1.2 Event 链设置

Event 链是由多个 Event 链接成的事件。最多可以由 4 个 Event 串成链。Event 链的触发条件是所有 Event 先后触发。

在 New Event 对话框中，点击 Event Link…按钮，打开 Event Link 窗口。

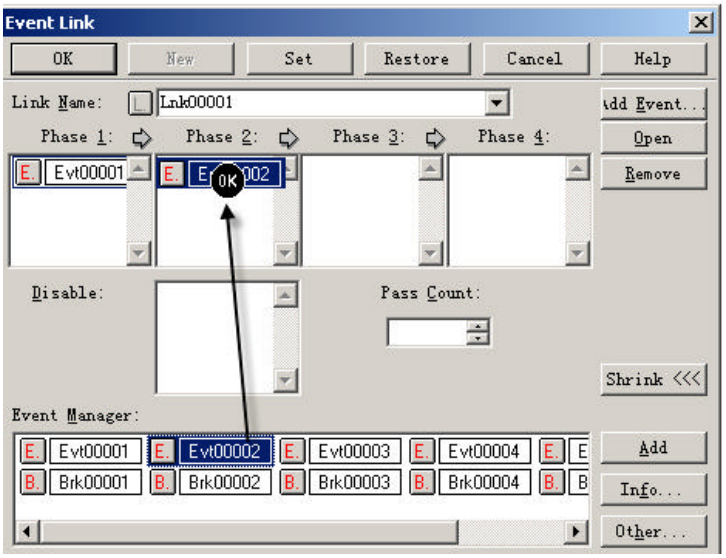

图 5-34 Event Link 窗口

如果设置了 Event 或设置了 Event 的链接条件，都能在 Event Link 窗口中的 Event Manager 列表框内看到。从下方的 Event 列表中将 Event 拖放到上方的事件链表中，点击 Set 就建立了事件链。

5.2.2 断点

除了 5.1.5 节中介绍的设置断点的方法。还可以用 Event 和 Event 链来设置断点，这样断点的触发条件可以定义的非常灵活。

先用 5.2.1.1 节的方法设置一个 Event。然后在 New Event 对话框中，点击 Break…按钮，打开 Break 窗口。

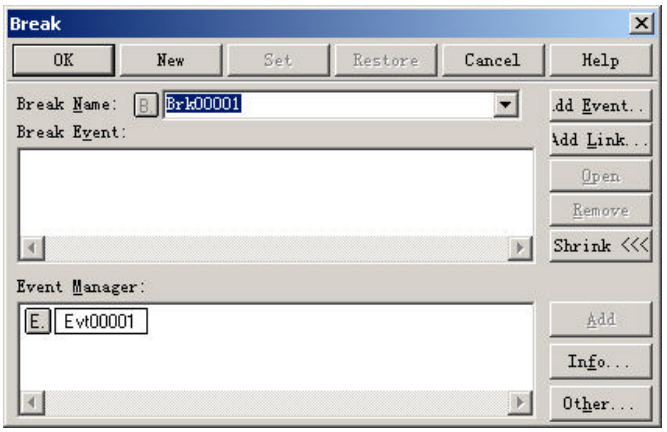

图 5-35 Break 窗口

通过拖动的方法，选择 Event，设置断点条件。

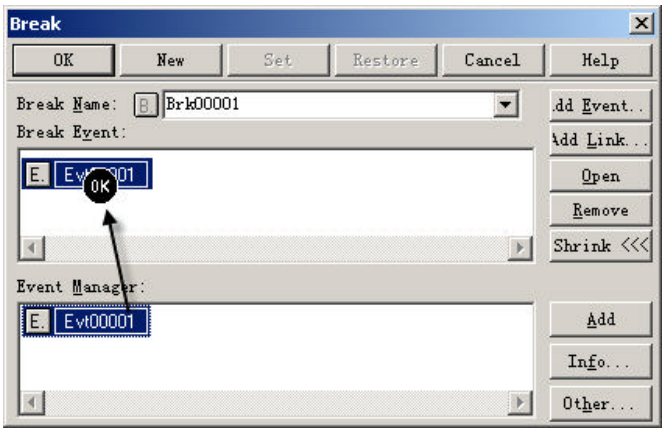

图 5-36 拖动事件

点击 Set 按钮，断点 Event 条件设置完成，Break 窗口 Event Manager 列表框中，Brk00001 图标变为红色。

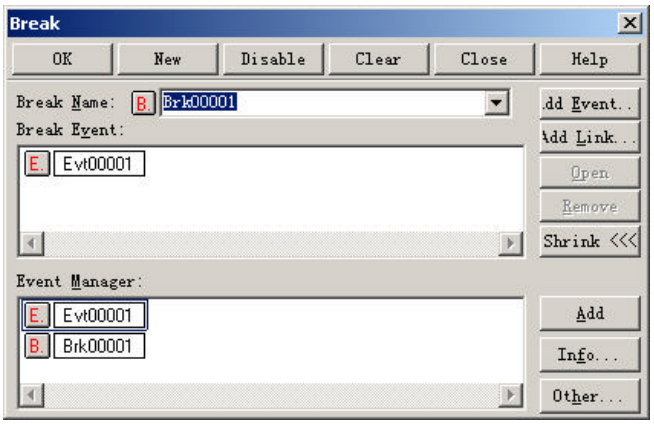

图 5-37 断点设置完成

5.2.3 Timer（计时）

SM78Kx plus 提供了 Timer 功能，使用户能测量执行全部或部分程序需要的时间。

以下图为例，假设我们要计算一段代码（12 行到 16 行）执行的时间。

首先要开启计时功能，选择菜单 Option 中的 Timer on。然后用 5.2.1.1 节中的方法设置 2 个事件，这两个事件将用于 Timer 的开始和结束触发。设置完毕后在 Source 窗口的 12 行和 16 行处会显示“ E ”标识，反汇编窗口中相应起始行处也有显示。

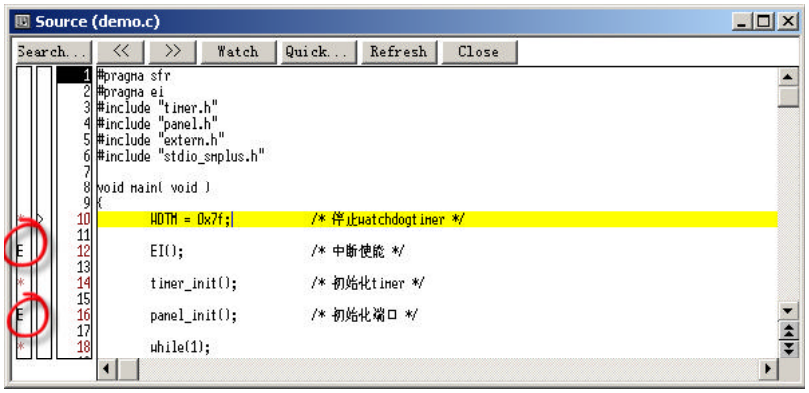

图 5-38 设置事件

选择 Event 菜单中的 Timer…，打开 Timer 对话框。

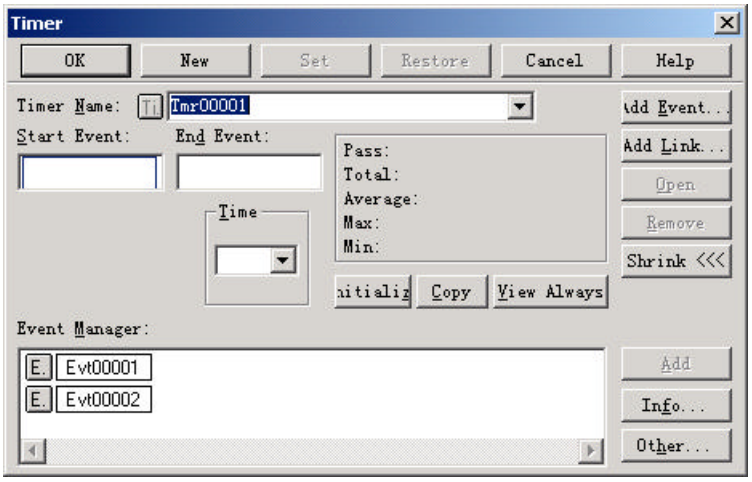

图 5-39 Timer 窗口

在 Timer 对话框中，设置 Evt00001 为 Start Event，设置 Evt00002 为 End Event，这些事件可以直接从 Event Manager 列表框中拖动，放到相应的对话框内。设置好 Timer 的名字 Tmr00001，点击 Set 按钮，Tmr00001 图标在 Event Manager 列表框中变成红色。

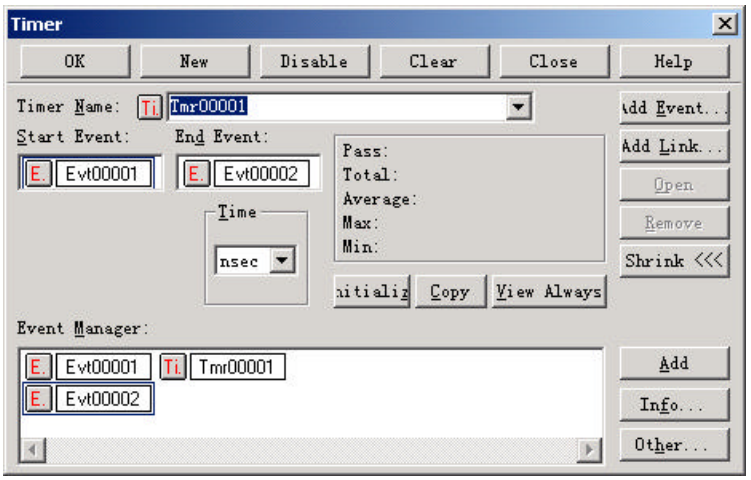

图 5-40 Timer 设置完毕

在这一例子中设了一个断点，以使运行中的程序停下来。

执行程序。当程序执行到 12 行时触发 Evt00001，开始计数；当程序执行到 16 行时触发 Evt00002，结束计数；当程序执行到断点时程序停止，此时可以从 Timer 对话框中观察到计数结果。

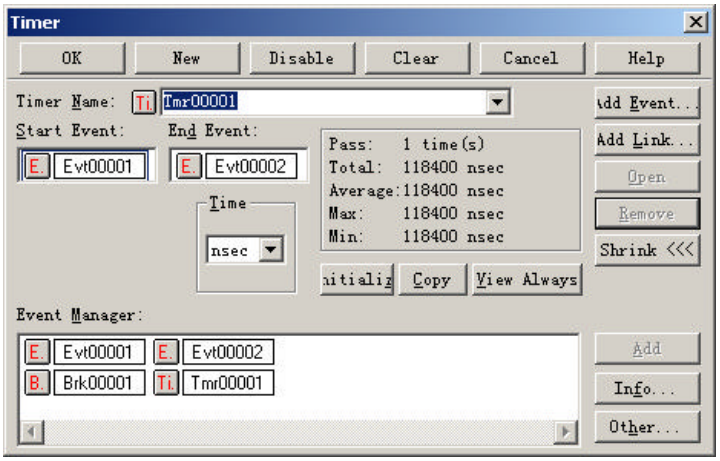

图 5-41 计数结果

5.2.4 Trace（追踪）

SM plus 提供了 Trace 功能，可以记录执行过程中程序运行轨迹，和内存的变化的情况。

Trace 存储器具有一个环形缓冲区域，记录最近的数据，数据包括程序执行的地址、数值和跳转信息。当写入数据超过最大容量时，最近的数据被保留，超出容量的旧数据从缓冲器中删除。

要使用 Trace 功能，只需选择 Option 菜单中的 Trace on，开启 Trace 功能。当程序开始执行时，相关信息就被记录下来了。

要观察 Trace 结果，点击 Trace 按钮，或是选择 Browse 菜单中的 Trace，打开 Trace 窗口。

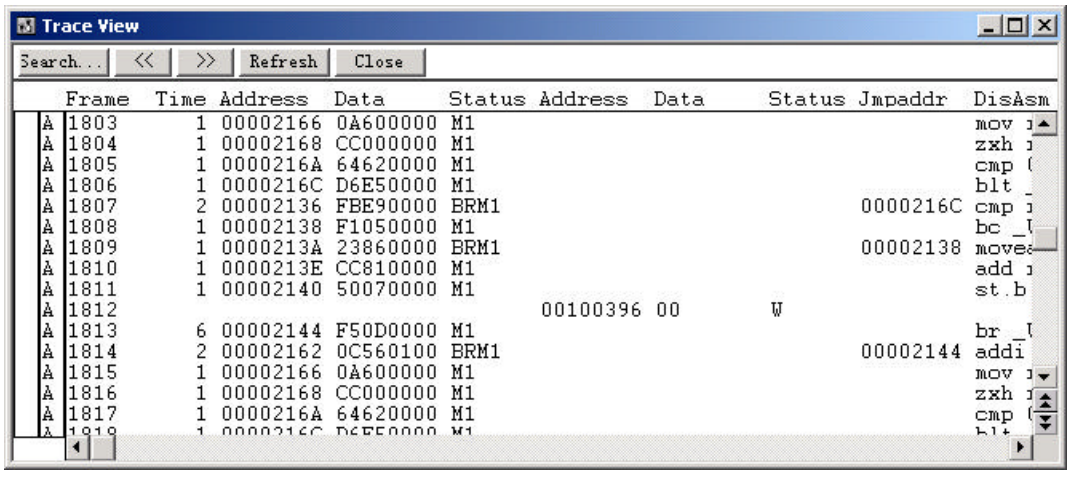

图 5-42 Trace 窗口

5.2.5 Timing Chart（时序图）

SM plus 提供 Timing Chart 窗口，可以显示芯片的端口信号。

点击 Timing Chart 按钮  ，或是选择 Simulator 菜单中的 Timing Chart，打开 Timing Chart 窗口。

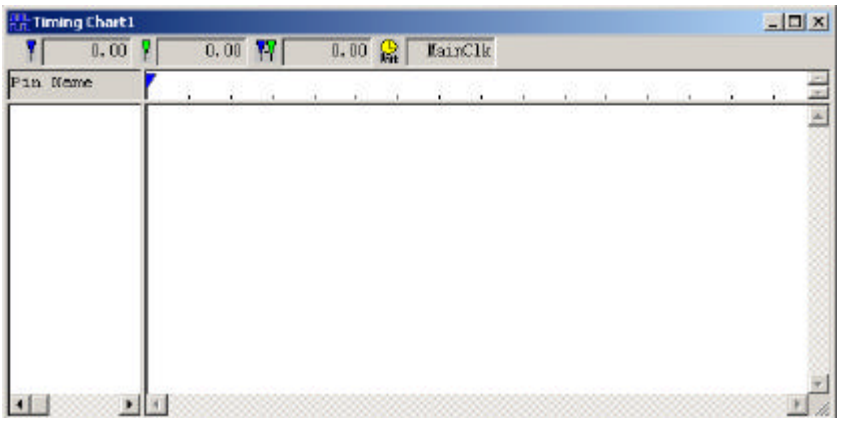

图 5-43 Timing Chart 窗口

选择 Edit 菜单中的 Select Pin，打开选择端口的对话框。

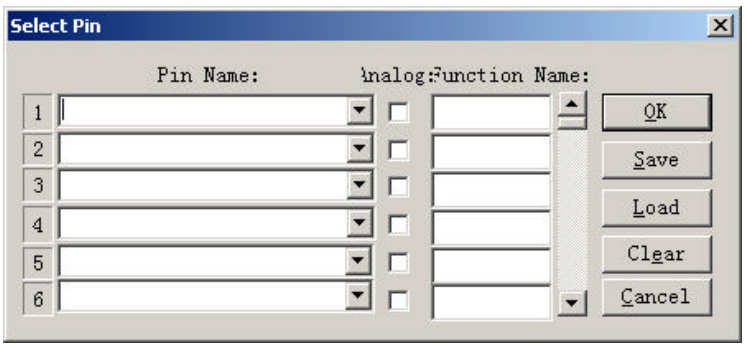

图 5-44 选择端口

Pin Name 选择的端口。下拉框提供可选的所有端口。

Analog 选中表示该端口是模拟值输出。

Function Name 端口的别名。这个名字由用户输入，将显示在 Timing Chart 窗口中。默认为端口名。

设置完毕点击 OK，相应的端口就被添加到 Timing Chart 窗口中。当程序执行时，从 Timing Chart 中可以观测端口的数值变化。

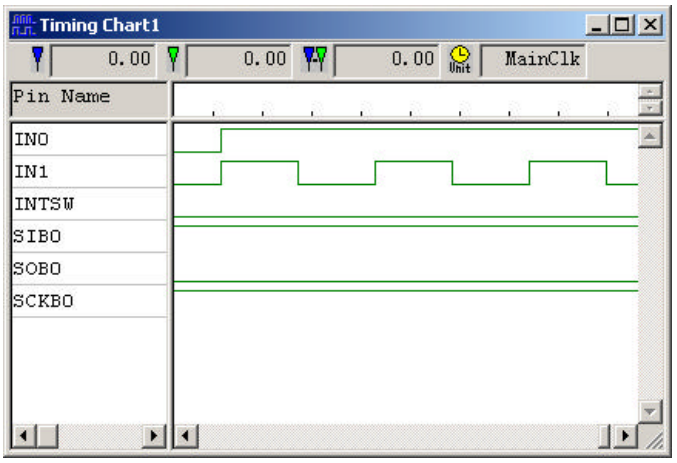

图 5-45 观测时序

5.2.6 Signal Data Editor（信号编辑器）

SM plus 提供信号编辑器，可以编辑各个端口的输入信号。

点击 Signal Data Editor 按钮  ，或是选择 Simulator 菜单中的 Signal Data Editor，打开信号编辑器窗口。

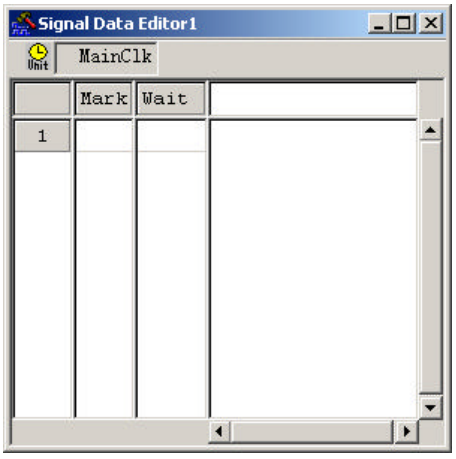

图 5-46 信号编辑器

5.2.6.1 添加端口

选择 Edit 菜单中的 Select Pin，打开添加端口的对话框。

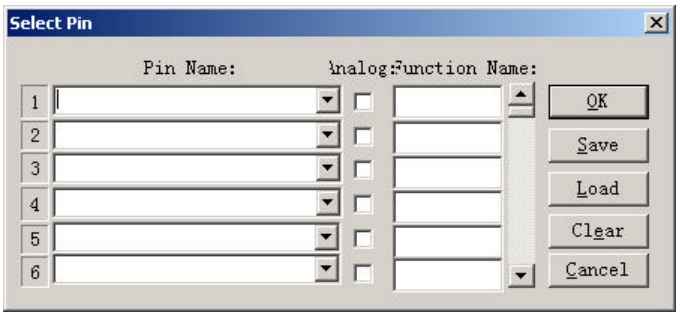

图 5-47　添加端口

Pin Name　　选择的端口。下拉框提供可选的所有端口。

Analog　　选中表示该端口是模拟值输出。

Function Name　　端口的别名。这个名字由用户输入，将显示在 Timing Chart 窗口中。默认为端口名。

设置完毕点击 OK，相应的端口就被添加到信号编辑器窗口中。

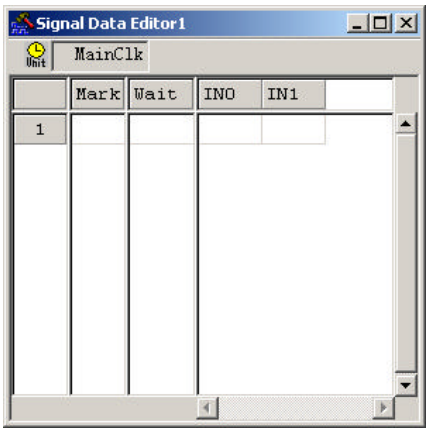

图 5-48　添加端口完成

5.2.6.2　编辑输入信号

在图 5-48 的“Wait”栏可以编辑等待时间，单位为 clock。

在右侧的“IN0”、“IN1”两个端口栏可以编辑信号值，数字端口的值只能为 1 或者 0，模拟端口的值可以是 0 到 5000 的任意整数。黄色的行表示当前输入行。

编辑完毕如下图所示：

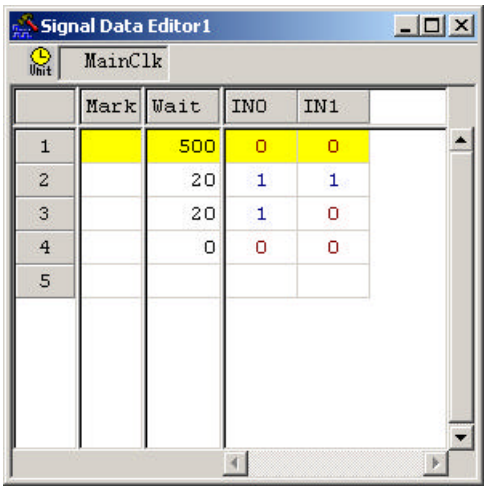

图 5-49　编辑输入信号

5.2.6.3　循环控制

通过设置循环控制，可以使信号自动循环输入。

在栏中想要设置为循环起点的位置，点鼠标右键弹出菜单，选择 Loop Start。

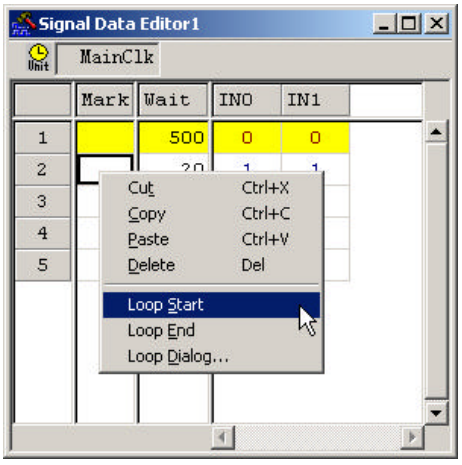

图 5-50　设置循环起点

在栏中想要设置为循环终点的位置，点鼠标右键弹出菜单，选择 Loop End，循环设置完成。

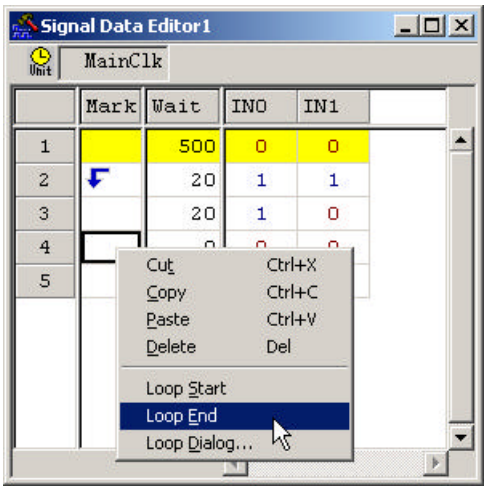

图 5-51 设置循环终点

5.2.6.4 启动输入

在信号编辑器窗口处于激活状态时，点击开始输入按钮，或是选择 Edit 菜单中的 Signal Input->Start，信号开始输入。

5.2.6.5 停止输入

在信号编辑器处于输入状态时，点击停止输入按钮，或是选择 Edit 菜单中的 Signal Input->Stop，信号停止输入。

5.2.6.6 复位输入

点击复位按钮，或是选择 Edit 菜单中的 Signal Input->Reset，输入信号复位。

5.2.7 I/O Panel（I/O 面板）

SM plus 的 I/O Panel 中提供了许多虚拟的外部设备，比如按钮、灯、LED、电机、变压器等。用户可以用它们建立一个虚拟的应用环境。

5.2.7.1 创建控件

点击 I/O Panel 按钮，或是选择 Simulator 菜单中的 I/O Panel，打开 I/OPanel 窗口。

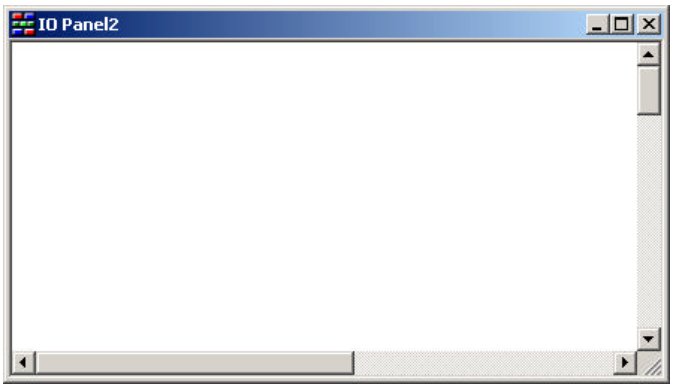

图 5-52 I/O 面板

SM plus 提供了多种控件，可以像绘图一样放置到 I/O 面板上。从界面上的 Parts 工具栏可以选择这些控件，或是在激活 I/O 面板后从 Parts 菜单中选择。

图 5-53 Parts 工具栏

除了控件，SM plus 还提供了绘制线、多边形、文本的功能，使 I/O 面板的显示更加丰富。从界面上的 Figure 工具栏可以选择这些控件，或是在激活 I/O 面板后从 Figure 菜单中选择。

图 5-54 Figure 工具栏

下图是一个已经拖放好控件的 I/O 面板。

图 5-55 包含控件的 I/O 面板

5.2.7.2 连接控件

按照上一节的方法操作，创建一个包含各种控件的 I/O 面板。这些控件需要与芯片上的端口相连，才能使用。

在任何一个控件的图像上双击鼠标左键，或是在选中控件后选择 View 菜单中的 Properties，

都可以打开属性对话框。以图 5-55 中的 I/O 面板为例，双击面板上的按钮控件，将会弹出下图所示的属性对话框。

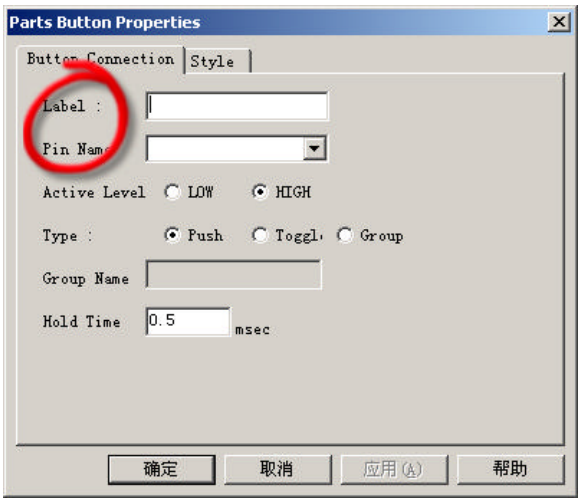

图 5-56　按钮属性对话框

不同种类的控件，属性对话框也不同，但它们都有 Label 和 Pin Name 这两个最基本的编辑框。

Label　　控件的显示名。

Pin Name　　与控件相连的端口。下拉框提供了所有可以选择的端口。

设置好 Label 和 Pin Name 两栏，点击 OK，控件就连接完毕，可以工作了。

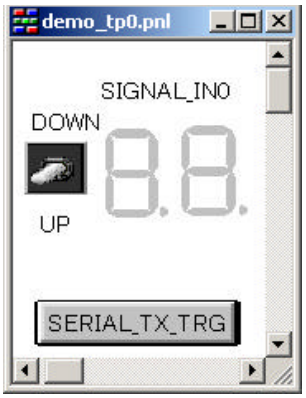

图 5-57　连接好的控件

5.2.7.3　使用控件

控件可分为两种，输入信号的控件和输出信号的控件。以图 5-57 为例，按钮和开关控件输入信号，7 段 LED 显示输出信号。

对于输出信号的控件，当程序执行时，如果有相连的信号输出，7 段 LED 就会相应的亮或者灭。

对于输入信号的控件，当程序执行时，由用户点击来发出输入信号。但是在初始情况下，I/O 面板为编辑状态，此时鼠标为十字箭头形状，用户点击控件只是选中，而不是按键的动作操作。

如下图所示：

图 5-58 控件处于编辑状态

要改编辑状态为仿真状态,点击 Simulation Mode 按钮,或是选择 Figure 菜单的 Simulation Mode 选项。此时鼠标为手形状，用户点击控件将是按键动作，激活输入信号。如下图所示：

图 5-59 控件处于仿真状态

图 5-60 使用中的 I/O 面板

5.2.8 Standard I/O

Standard I/O 是 SM plus 提供的标准输入输出接口。用户程序中的 printf 函数将会把信息输出到 Standard I/O 窗口上，而 scanf 函数从 Standard I/O 窗口获得输入。

点击 Standard I/O 按钮，或是选择 Simulator 菜单中的 Standard I/O，打开 Standard I/O 窗口。

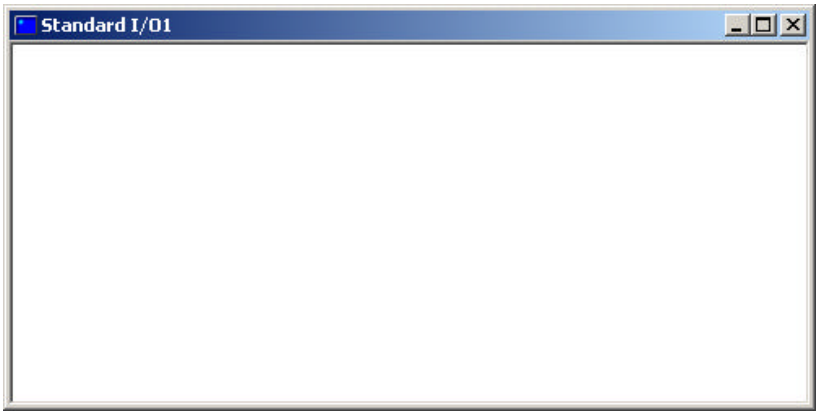

图 5-61 Standard I/O 窗口

5.2.9 Serial GUI

Serial GUI 模拟了一个串口的终端，可以向它连接的串口设备发送和接收数据。一个 Serial GUI 窗口可以连接一个串口设备。

5.2.9.1 启动 Serial GUI

点击 Serial 按钮  ，或是选择 Simulator 菜单中的 Serial，打开 Serial GUI 窗口。

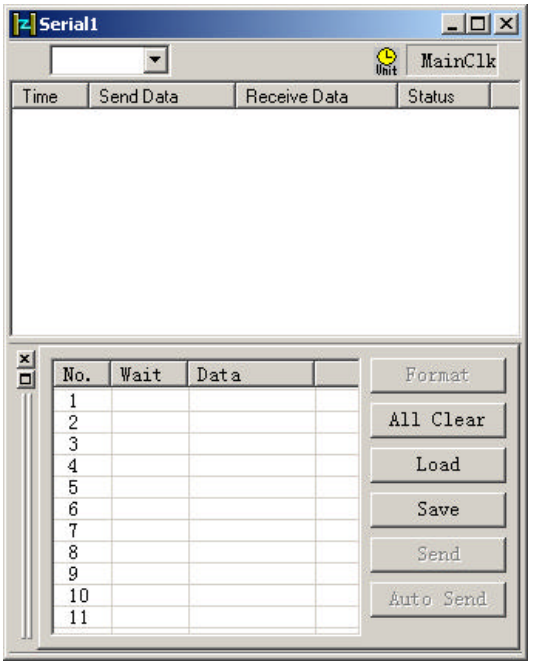

图 5-62 Serial GUI 窗口

5.2.9.2 连接串口设备

可供连接的串口已经在设备文件中设置，用户只需在下拉菜单中选择即可。

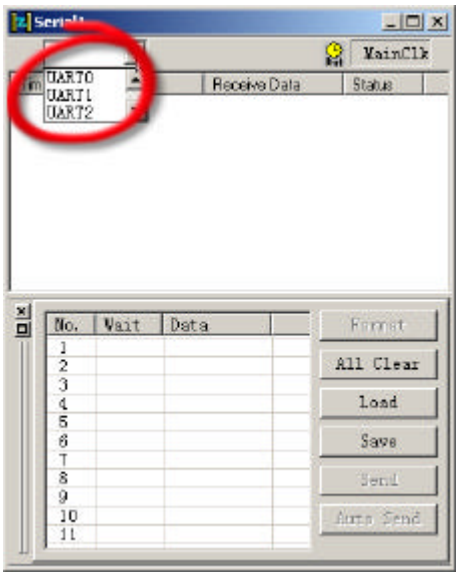

图 5-63 连接串口设备

5.2.9.3 接收数据

程序执行后，当与 Serial GUI 相连的串口设备发送出数据时，Serial GUI 将显示接收到的数据。

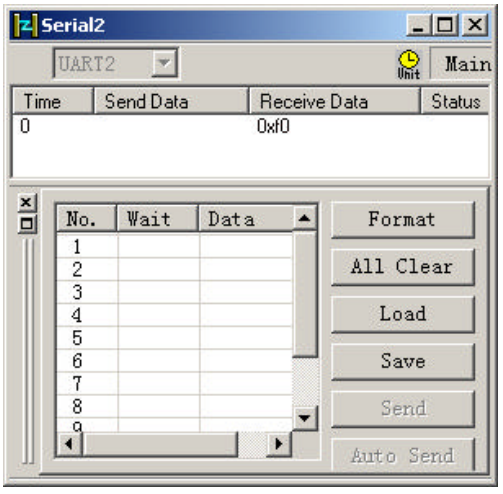

图 5-64 接收数据

5.2.9.4 发送数据

在 Serial GUI 的发送数据栏填写发送输入，然后点击 Send 按钮。数据就发送到所连接的串口中了。

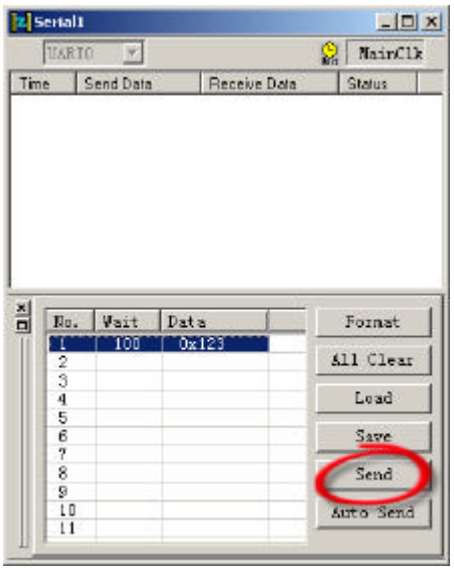
Serial1
UART0
MainClk
Time
Send Data
Receive Data
Status
No.
Wait
Data
1
100
0x123
2
3
4
5
6
7
8
9
10
11
Format
All Clear
Load
Save
Send
Auto Send

图 5-65　发送数据