

Keil μ Vision 与 C51 语言基础

用 C 语言编写的源代码程序不能直接使用,需要对该程序进行编译,最后生成 .hex 文件。因此首先要进行程序源代码的编写、编译以及调试。目前这些工作都在集成化的软件开发环境下完成。目前大多使用 Keil 软件来完成 C51 程序源代码的编辑、调试以及编译,最终生成目标代码 .hex 文件,并加载到单片机中运行。本章介绍如何使用 Keil μ Vision4 集成化开发平台,结合 C51 语言基础知识,为 C51 程序设计与开发打下基础。

3.1 Keil μ Vision4 集成开发环境

在单片机的程序开发中,最常使用的是 Keil μ Vision 系列软件,它是德国 Keil Software 公司开发的用于 51 系列兼容单片机的软件开发系统。Keil μ Vision 编译器主要为 8051 单片机软件开发提供开发环境,支持多个公司的 8051 单片机芯片,同时集编辑、编译、仿真等功能为一体,具有强大的软件调试功能。

3.1.1 Keil μ Vision4 简介

Keil μ Vision4 具有 Windows 风格的可视化操作界面,集成了丰富的库函数和各种编译工具。Keil μ Vision4 能够完成 51 系列单片机以及与 51 系列兼容的绝大部分类型单片机的程序设计和仿真。Keil μ Vision4 是一个非常优秀的编译器,用户常使用 Keil μ Vision4 来开发工程、调试程序,最终生成的 .hex 文件是可执行的代码文件,受到广大单片机设计者的广泛使用,其主要特点如下。

- (1) 支持汇编语言、C51 语言等多种单片机程序设计语言。
- (2) 可视化的项目管理,界面友好。
- (3) 支持最为丰富的产品线,可以使用 51 系列及其兼容内核的单片机。
- (4) 具有完善的编译、连接工具。
- (5) 具备丰富的仿真调试功能,可以仿真并口、串口、A/D、D/A、定时器/计数器以及中断等资源。
- (6) 可以和外部仿真器联合使用,支持在线调试。
- (7) 支持在一个工作空间中进行多项目的程序设计。
- (8) 支持多级代码优化。

3.1.2 Keil μ Vision4 界面概览

Keil μ Vision4 安装完成后,可在桌面上看到 Keil μ Vision4 软件的快捷图标,双击该快捷图标,即可启动该软件。

Keil μ Vision4 集成开发环境提供了良好的用户界面,对于一个 μ Vision 项目,其界面如图 3-1 所示。由于采用了标准的 Windows 界面,Keil μ Vision4 集成开发环境由菜单栏、工具栏、文件编辑窗口、项目管理窗口和输出窗口等组成。

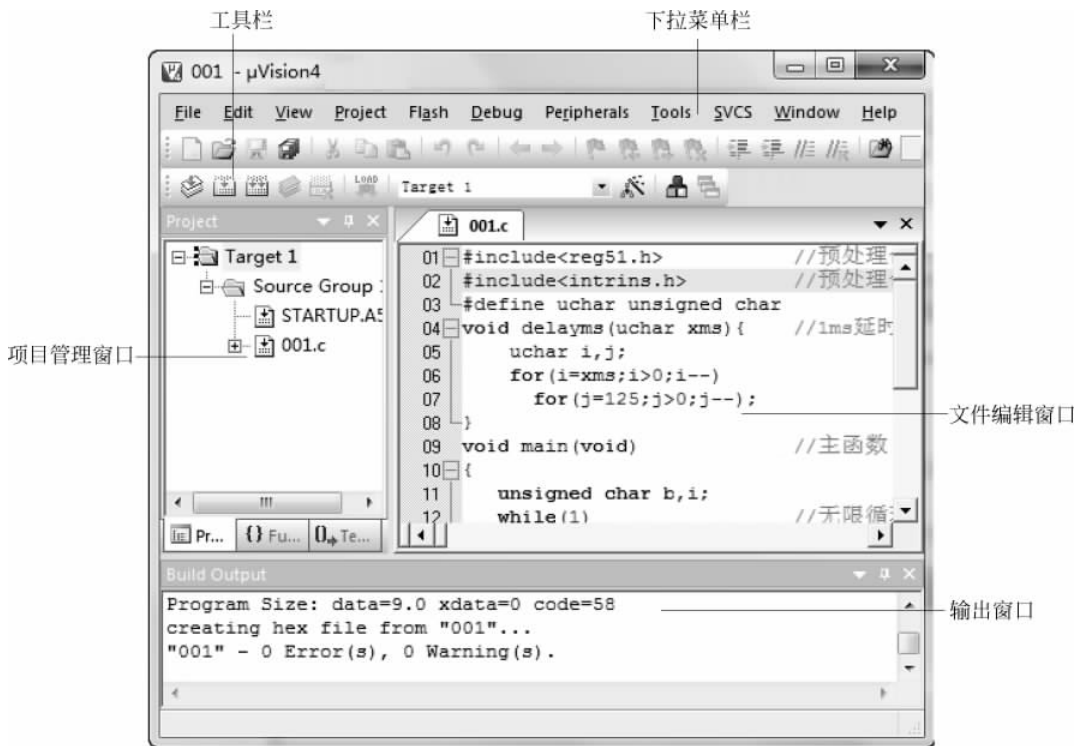


图 3-1 μ Vision4 集成开发环境用户界面

其中,Keil μ Vision4 的菜单栏和工具栏提供了项目操作、编辑操作、编译调试以及帮助等各种常用操作。项目管理窗口用来管理整个 C51 项目,包括头文件、源文件等。文件编辑窗口则显示了源代码编辑及其他一些窗口。输出窗口则显示了该项目的编译链接信息。另外,Keil μ Vision4 集成开发环境还提供了多种不同用途的显示窗口,可以用于反汇编的查看、堆栈数据查看、变量查看以及仿真波形图等。这些窗口是 Keil μ Vision4 集成开发环境使用的基础,下面介绍一些在程序设计及仿真调试中常用的窗口及操作。

3.2 Keil μ Vision4 的 C51 开发流程

Keil μ Vision4 软件的初始界面如图 3-2 所示,可以在该集成开发环境中编写 C51 程序,并下载到硬件电路中执行。下面通过一个实例来讲解如何在 Keil μ Vision4 集成开发

环境下进行单片机 C51 语言的程序设计。

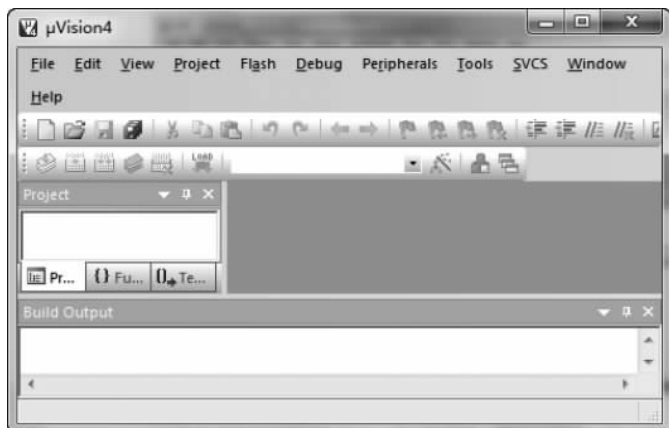


图 3-2 Keil μ Vision4 启动界面

3.2.1 创建项目

在使用 Keil 对程序进行调试、编译之前, C51 程序需要在一个项目中进行管理。因此, 需要首先创建一个新的项目, 操作步骤如下所示。

(1) 在 Keil μ Vision4 集成开发环境中, 选择 Project $\rightarrow$ New μ Vision Project 命令, 如图 3-3 所示。

(2) 单击 New μ Vision Project 命令, 弹出 Create New Project 窗口, 如图 3-4 所示。在该窗口中, 需在“文件名”文本框中输入新建项目的名字, 并选择需要保存的目录。

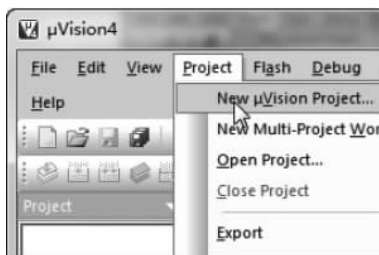


图 3-3 选择新建项目命令

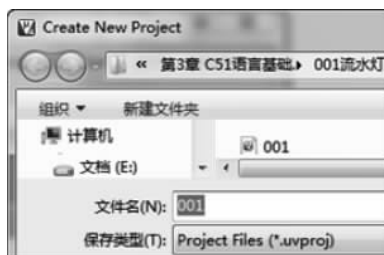


图 3-4 输入项目名称

(3) 单击“保存”按钮, 此时弹出选择单片机型号对话框, 如图 3-5 所示。用户可以在其中选择所使用的单片机型号。例如, 选择 Atmel 公司的单片机 AT89C51, 此时在 Description 栏中将会显示该单片机的资源。此时选择的单片机型号也可以在项目建立后重新更改。

(4) 选择完毕后, 单击“确定”按钮, 此时弹出提示信息窗口, 如图 3-6 所示。该提示信息询问是否将适合该单片机型号的 AT89C51 启动代码添加到项目中。对于 C51 程序, 选择添加即可。

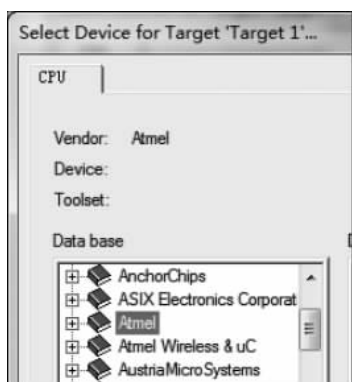


图 3-5 单片机型号对话框

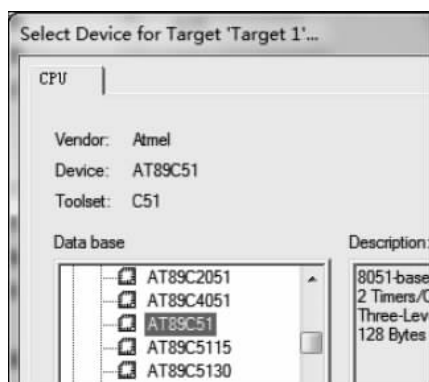


图 3-6 将启动代码添加到项目中

(5) 单击“是”按钮,选择添加,此时项目创建完毕,但是还没有任何源文件,属于一个空项目。

3.2.2 创建源程序文件

当一个新的项目创建完成后,需要将编写的用户源代码添加到这个项目中,用户添加程序文件通常有两种形式:一种是新建文件,另一种是添加已创建的文件。

新建文件的流程如下。

(1) 选择 File→New 命令,此时工作区中弹出一个新的源代码编辑窗口,如图 3-7 所示。用户可以在其中输入 C51 的源代码。

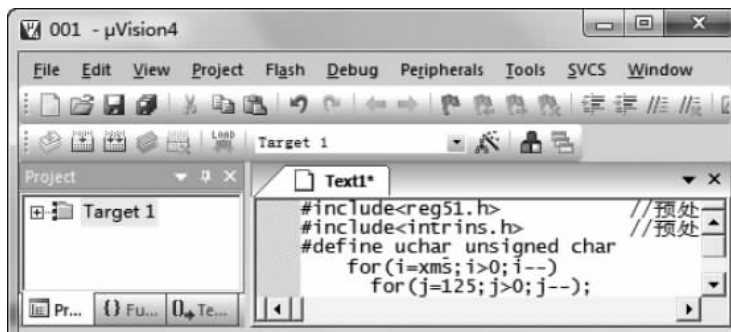


图 3-7 源代码编辑窗口

(2) C51 源代码输入完毕后,可以选择 File→Save 命令,弹出 Save As 窗口,如图 3-8 所示。在“文件名”文本框中输入 001.c,然后单击“保存”按钮。如果用汇编语言编程,则文件夹的扩展名应为 001.asm。

(3) 在项目管理窗口中,右击 Source Group 1 标签,选择 Add Files to Group ‘Source Group 1’命令,如图 3-9 所示,在弹出的对话框中选择刚才保存的 C51 源文件,如图 3-10 所示。单击 001 文件后,单击 Add 按钮,再单击 Close 按钮,文件添加就已经完成了。此时的项目管理窗口中的 001.c 文件已经出现在 Source Group 1 目录下了,如图 3-11 所示。

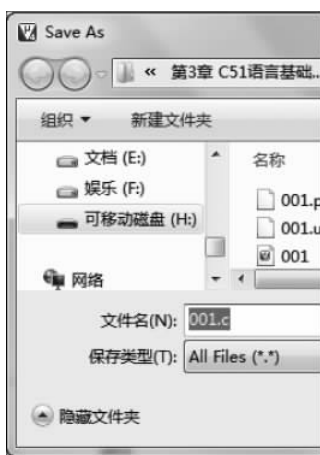


图 3-8 Save As 窗口

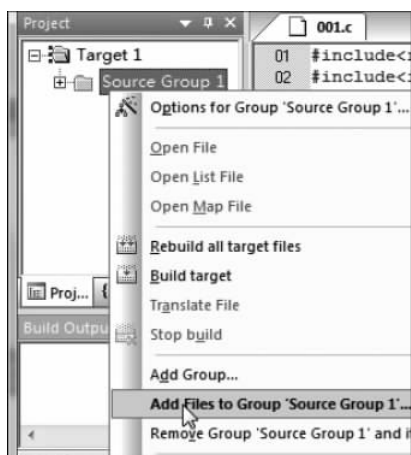


图 3-9 添加文件



图 3-10 保存的源文件

3.2.3 程序编译与调试

项目及 C51 源文件准备好后便可以编译项目了。Keil μ Vision4 集成开发环境提供多个源代码编译命令,分别介绍如下。

- (1) 选择 Project→Translate... .c 命令,可以完成对当前 C51 源代码的翻译。
- (2) 选择 Project→Build target 命令,即可对该项目进行编译。
- (3) 选择 Project→ReBuild all target files 命令,可以重新编译所有的项目文件。

单击快捷按钮中的 ,对当前文件进行编译,如果源程序无误,则编译完成后,将在输出窗口中显示编译结果,如图 3-11 所示。

3.2.4 仿真调试

Keil μ Vision4 集成开发环境中提供了强大的程序仿真功能。在一个程序下载到单片机硬件中执行之前,可以先对程序进行仿真,以确保源程序符合要求。下面就介绍如何使用 Keil μ Vision4 来进行单片机 C51 语言程序的仿真调试。操作步骤如下。

- (1) 选择 Debug→Start/Stop Debug Session 命令(或单击开始/停止调试的工具按钮

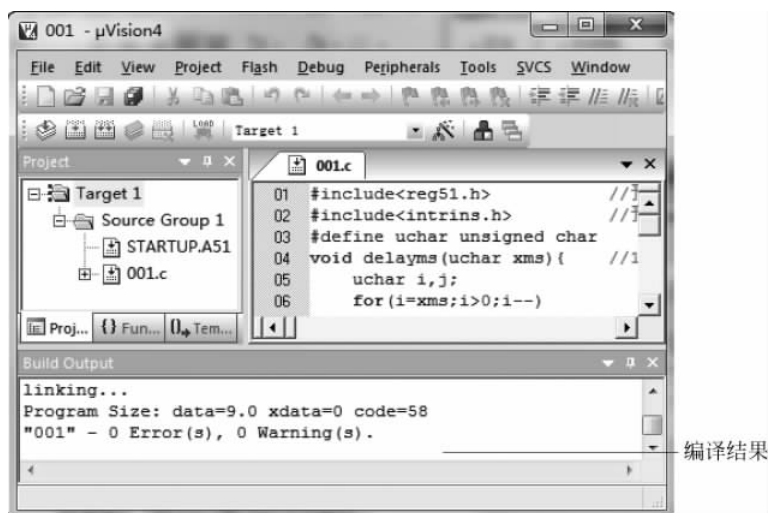


图 3-11 输出窗口显示编译结果

此时, Keil μ Vision4 进入仿真调试模式, 如图 3-12 所示。在该模式下, 显示出反汇编窗口、主调试窗口、寄存器窗口、串行输出窗口、堆栈窗口(可查看变量、堆栈、存储器、符号观察等窗口)。若不需要查看反汇编窗口, 可直接单击该窗口右上角的 $\times$ 按钮关闭即可。

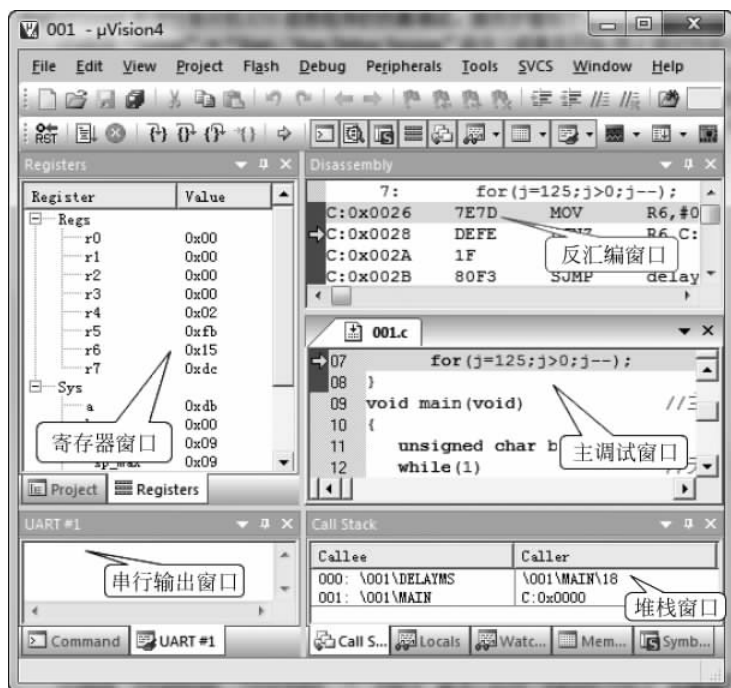


图 3-12 仿真调试模式

(2) 选择 Peripherals→I/O-Ports→Port 0 命令, 打开 Parallel Port 0 仿真窗口, 如图 3-13 所示。

(3) 选择 Peripherals→I/O-Ports→Port 1 命令,打开 Parallel Port 1 仿真窗口,如图 3-14 所示。

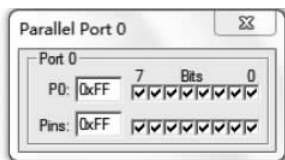


图 3-13 P0 口仿真窗口

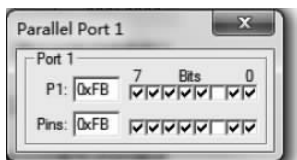


图 3-14 P1 口仿真窗口

(4) 单击  工具按钮,开始单步执行程序。可以观察到 P0 口由右至左的低电平逐个变化,同时也可以查看 P1、P2、P3 口的引脚电平状态。

在程序调试状态下,可运用工具按钮进行单步、跟踪、断点、全速运行等方式的调试,也可观察单片机资源的状态,例如程序存储器、数据存储器、特殊功能寄存器、变量寄存器及 I/O 口的状态。这些工具按钮大多是与菜单栏命令 Debug 下拉菜单中的各项子命令一一对应的,只是工具按钮要比下拉菜单命令使用起来更加方便快捷。

(5) 常用的工具按钮功能介绍如下。

: 控制编译输出窗口开/关。用于输出项目编译链接的信息,如图 3-11 所示。如果程序在编译链接的过程中有错误,则显示具体的错误信息及其位置。同时,编译输出窗口还显示输出代码的大小。

: 控制反汇编窗口开/关。用于显示 C51 代码的反汇编代码,如图 3-15 所示。反汇编窗口在程序运行或调试模式下才会出现。反汇编窗口中列出了 Keil μ Vision4 编译器对当前 C51 代码的反汇编结果。可以选择两种语言同时在文件编辑窗口显示及运行,也可选择汇编程序在单独窗口显示及运行。

: 控制符号观察窗口开/关。如图 3-16 所示,显示了程序运行时所有函数和模块的公共符号、当前模块或函数的局部符号、代码行号和当前载入应用所定义的 CPU 特殊功能寄存器(SFR)。符号观察窗口也需要在程序调试或运行时才能显示。

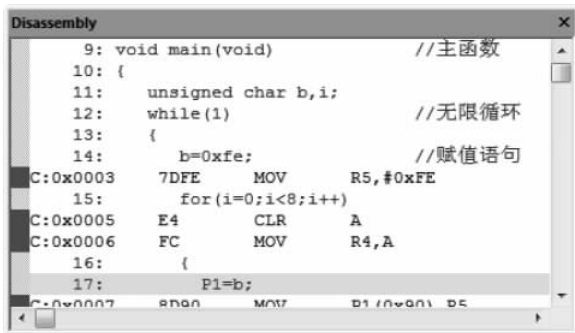


图 3-15 反汇编代码窗口

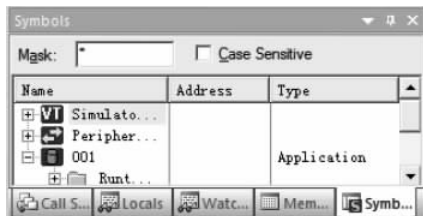


图 3-16 观察窗口

: 控制 CPU 寄存器窗口开/关。如图 3-17 所示,CPU 寄存器窗口显示了 CPU 寄存器中的值。CPU 寄存器窗口在程序运行和调试状态下显示,用户可以单击项目管理窗

口下的 Regs 标签来显示。左边的工程窗口给出了常用的寄存器 r0~r7 以及 a、b、sp、dptr、PC、psw 等特殊功能寄存器的值,这些值会随着程序的执行发生相应的变化。

: 控制调用窗口开/关。如图 3-18 所示,用于显示程序执行过程中对子程序的调用情况,观察该窗口也是在程序执行和调试模式下。

: 控制变量观察窗口开/关。如图 3-19 所示,用于显示程序执行过程中变量的值。

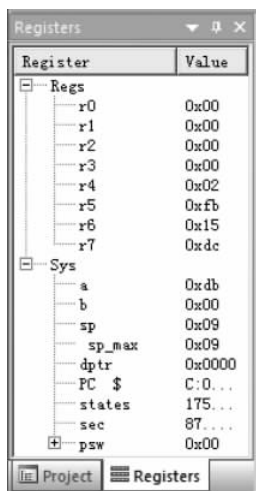


图 3-17 CPU 寄存器窗口

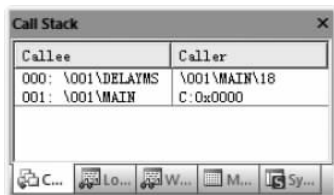


图 3-18 调用窗口

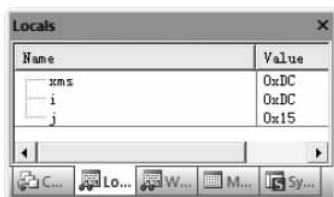


图 3-19 变量观察窗口

: 控制存储器窗口开/关。如图 3-20 所示,存储器窗口用于显示不同存储器的数据,存储器窗口也是在程序运行和调试状态才有的。存储器窗口的地址栏处输入“0000H”后按回车键,则可以查看单片机片内程序存储器的内容,单元地址前有“C:”表示程序存储器。如果在存储器窗口的地址栏处输入“D: 00H”后按回车键,则可以查看单片机片内数据存储器的内容,单元地址前有“D:”表示数据存储器。

: 控制串行口窗口开/关。如图 3-21 所示,用于显示串行口的 I/O 信息,它只在程序运行或调试状态下显示。

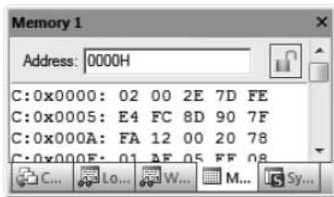


图 3-20 存储器窗口



图 3-21 串行口窗口

: 控制逻辑分析窗口开/关。如图 3-22 所示,用于显示程序运行时变量的波形图,逻辑分析窗口只在程序运行或调试状态下显示。

(6) 各调试功能的工具按钮如下。

: 调试状态进入/退出。

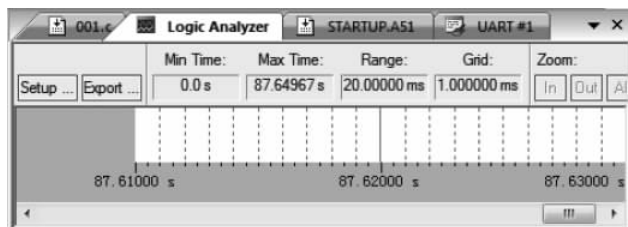


图 3-22 逻辑分析窗口

：复位 CPU。在程序不改变的情况下，使程序重新开始运行，单击本按钮即可。执行此命令后程序指针返回到 0000H 地址单元。另外，一些内部特殊功能寄存器在复位期间也将重新赋值。例如，A 将变为 00H，SP 变为 07H，DPTR 变为 0000H，P3~P0 口变为 FFH。

：全速运行。单击本按钮，即可实现全速运行程序。若程序中已经设置断点，程序将执行到断点处，并等待调试指令。在全速运行期间，不允许对任何资源进行查看，也不接受其他命令。

：单步跟踪。可以单步跟踪程序，每执行一次此命令，程序将运行一条指令。当前的指令用黄色箭头标出，每执行一步箭头都会移动，已执行过的语句呈绿色。

：单步运行。本按钮实现单步运行程序，此时单步运行命令将把函数和函数调用当作一个实体来看待，因此单步运行是以语句（该语句不管是单一命令行还是函数调用）为基本执行单元。

：执行返回。在用单步跟踪命令跟踪到子函数或子程序内部时，使用本按钮即可将程序的 PC 指针返回到调用此子程序或函数的下一条语句。

：运行到光标行。

：停止程序运行。

(7) 断点操作的快捷按钮如下。

在程序调试中常常要设置断点，一旦执行到该断点程序运行即停止，可在断点处观察有关变量值，以确定问题所在。

：插入/清除断点。

：使能/禁止断点，开启或暂停光标所在行的断点功能。

：清除所有的断点设置。

：禁止所有断点，暂停所有断点。

此外，插入或清除断点最简单的方法，即将光标移至需要插入或清除断点的行首双击即可。上述的 4 个快捷命令均可从 Debug 的下拉菜单中找到。

通过上面的仿真操作可以看出，这段源代码完全符合预期的功能。合理使用 Keil μ Vision4 集成开发环境的仿真调试功能，不仅能够确保源程序的正确性，同时还可以在没有任何硬件的基础上学习 C51 语言及单片机程序开发，从而加快学习速度。

3.2.5 项目设置

项目创建完毕后，还需对项目进行进一步的设置，以满足要求。右击项目管理窗口的

Target 1, 选择 Options for Target 'Target 1' 命令, 如图 3-23 所示, 即出现对项目进行设置的对话框, 如图 3-24 所示。该对话框下有多个页面, 通常需要设置的有两个, 一个是 Target 页面, 另一个是 Output 页面, 其余设置取默认值即可。

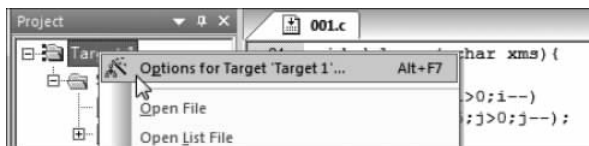


图 3-23 项目设置选择

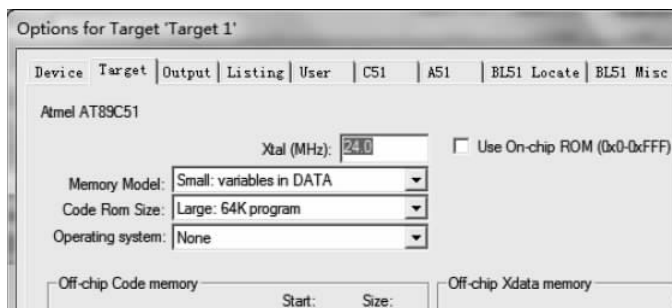


图 3-24 项目设置对话框

1. Target 页面

(1) Xtal(MHz): 设置晶振频率值, 默认值是所选目标 CPU 的最高可用频率值, 可以根据需要重新设置。该设置与最终产生的目标代码无关, 仅用于软件模拟调试时显示程序执行时间。正确设置该数值可使显示时间与实际所用时间一致, 一般将其设置成与硬件目标样机所用的频率相同, 如果没必要了解程序执行的时间, 也可以不设置。

(2) Memory Model: 设置 RAM 的存储器模式, 有 3 个选项。

① Small: 所有变量都在单片机的内部 RAM 中。

② Compact: 可以使用 1 页外部 RAM。

③ Large: 可以使用全部外部的扩展 RAM。

(3) Code Rom Size: 设置 ROM 空间的使用, 即程序的代码存储器模式, 有 3 个选项。

① Small: 只使用低于 2KB 的程序空间。

② Compact: 单个函数的代码量不超过 2KB, 整个程序可以使用 64KB 程序空间。

③ Large: 可以使用全部 64KB 程序空间。

(4) Use On-chip ROM: 是否仅使用片内 ROM 选项。注意, 选中该项并不会影响最终生成的目标代码量。

(5) Operating: 操作系统选项。Keil 提供了两种操作系统: Rtx tiny 和 Rtx full。通常不选用操作系统, 所以选用默认项 None。

(6) Off-chip COD MEMORY: 用以确定系统扩展的程序存储器的地址范围。