

MCS-51单片机应用实验教程

沈放 何尚平 万彬主编

重庆大学出版社



内 容 提 要

本书从单片机实验教学和工程实际应用角度出发,主要讲解实验实践相关教学内容,包括单片机编程开发工具——Keil 集成开发环境、单片机 Proteus ISIS 仿真、单片机基础实验、单片机应用系统综合实例等,涵盖了单片机“理论学习—基础实验—应用系统设计与开发—软硬仿真—硬件制作—整体调试”整套开发过程,形式新颖,内容齐全,覆盖面广,实用性强,既可作为应用型本科、专科院校的单片机实验及实践类课程教材,也可作为单片机应用初学者以及单片机开发人员的实用参考书。

图书在版编目(CIP)数据

MCS-51 单片机应用实验教程 / 沈放, 何尚平, 万彬

主编. -- 重庆: 重庆大学出版社, 2019.1

高等院校应用型本专科通用规划教材

ISBN 978-7-5689-1417-8

I. ①M… II. ①沈… ②何… ③万… III. ①单片微型计算机—高等学校—教材 IV. ①TP368.1

中国版本图书馆 CIP 数据核字(2018)第 277708 号

MCS-51 单片机应用实验教程

主 编 沈 放 何尚平 万 彬

策划编辑:曾令维

责任编辑:曾令维 版式设计:曾令维

责任校对:刘志刚 责任印制:张 策

*

重庆大学出版社出版发行

出版人:易树平

社址:重庆市沙坪坝区大学城西路 21 号

邮编:401331

电话:(023) 88617190 88617185(中小学)

传真:(023) 88617186 88617166

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn(营销中心)

全国新华书店经销

重庆俊蒲印务有限公司印刷

*

开本:787mm×1092mm 1/16 印张:15.25 字数:382 千

2019年1月第1版 2019年1月第1次印刷

印数:1—3 000

ISBN 978-7-5689-1417-8 定价:38.00 元

本书如有印刷、装订等质量问题,本社负责调换

版权所有,请勿擅自翻印和用本书

制作各类出版物及配套用书,违者必究

前 言

单片机具有功能强、使用灵活、性价比高、体积小、面向控制等特点,广泛应用于工业控制、智能仪器仪表、现代传感器、数据采集与处理、机电一体化、消费电子、家用电器、通信、办公自动化、医疗器械、计算机控制等领域,自 20 世纪 70 年代问世以来,已经对人类社会做出了巨大贡献。

单片机作为典型的具有代表性的嵌入式系统,并随着计算机技术、微电子技术的高速发展,目前各类院校很多专业已经普遍开设了单片机原理及其相关课程,并作为学习各种微处理器、控制器的基础。

单片机原理作为一门实践性、技术性很强的课程,在学好基础知识、基本技能的同时,最终在于掌握实际应用,本书即是在以习近平新时代中国特色社会主义思想指导下,落实“新工科”建设新要求,从单片机实验教学和工程实际应用角度出发,主要讲解实验实践相关教学内容,包括单片机编程开发工具——Keil 集成开发环境、单片机 Proteus ISIS 仿真、单片机基础实验、单片机应用系统综合实例等,涵盖了单片机“理论学习—基础实验—应用系统设计与开发—软硬仿真—硬件制作—整体调试”整套学习开发过程,形式新颖,内容齐全,覆盖面广,实用性强,既可作为应用型本科院校(自动化、电气工程及其自动化、电子信息工程、应用电子技术、通信工程、测控技术与仪器、机电一体化、车辆工程等专业)、专科院校(机制、机电一体化、数控、模具等专业)的单片机实验及实践类课程教材,也可作为单片机应用初学者以及单片机开发人员的实用参考书。

本书主编由南昌大学科学技术学院沈放、何尚平和南昌职业学院万彬老师担任。全书共分 3 章,其中万彬编写了第 1 章,沈放编写了第 2 章,何尚平编写了第 3 章,全书由沈放负责统稿。

南昌大学科学技术学院吴静进对本书进行了认真审阅,许仙明、朱淑云、陈艳、吴敏(排名不分先后)等提出了许多宝贵的意见,特此致谢!

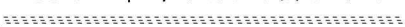
由于编者水平有限,书中错漏和不妥之处在所难免,恳请专家、同行老师和读者批评指正。

编 者

2018 年 9 月

目 录

第 1 章 Keil 集成开发环境及 Proteus ISIS 仿真	1
1.1 Keil 集成开发环境	2
1.1.1 Keil μ Vision2 工作环境	2
1.1.2 Keil 工程的创建	8
1.1.3 存储空间资源的查看和修改	21
1.1.4 变量的查看和修改	22
1.1.5 外围设备的查看和修改	23
1.2 Proteus ISIS 简介	24
1.2.1 Proteus ISIS 工作环境	24
1.2.2 电路原理图的设计与编辑	28
1.2.3 Proteus ISIS 与 Keil C51 的联调	36
第 2 章 单片机基础实验	38
2.1 实验系统介绍	38
2.1.1 实验系统特点	38
2.1.2 系统概述	38
2.1.3 系统电源	40
2.2 基本电路介绍	41
2.2.1 整机介绍	41
2.2.2 硬件资源	41
2.2.3 整机测试	42
2.2.4 单元电路原理及测试	42
2.2.5 扩展接口定义	54
2.2.6 扩展板的安装与使用	55
2.3 实验部分	57
2.3.1 EL-MUT-8051-Keil C 软件使用	57
2.3.2 P1 口输出输入实验	64
2.3.3 P1 口输入并输出实验	69
2.3.4 交通灯控制实验	73
2.3.5 I/O 口扩展实验	78
2.3.6 中断实验——有急救车的交通灯控制实验	81
2.3.7 定时器实验——循环彩灯实验	87
2.3.8 8255A 可编程并行接口实验一	91



2.3.9	8255A 可编程并行接口实验二——键盘实验	94
2.3.10	数码管显示实验	100
2.3.11	8279 键盘显示接口实验一	106
2.3.12	8279 键盘显示接口实验二	112
2.3.13	串行口实验一——单机实验	116
2.3.14	串行口实验二——双机实验	119
2.3.15	D/A 转换实验	127
2.3.16	A/D 转换实验	134
2.3.17	存储器扩展实验	138
2.3.18	8253 定时器实验	140
2.3.19	8259 中断控制器实验	143
2.3.20	CPLD 实验	148
2.3.21	LCD 显示实验	153
第3章	单片机应用系统综合实例	156
3.1	单片机应用系统的开发过程	156
3.2	单片机控制流水灯应用系统示例	164
3.3	单片机控制步进电机应用系统示例	173
3.4	单片机应用系统设计过程	181
3.5	基于单片机的数码管时钟系统设计	184
3.6	基于单片机的简易数字直流电压表系统设计	198
3.7	基于单片机的液显数字时钟系统设计	216
参考文献		238

第 1 章 Keil 集成开发环境及 Proteus ISIS 仿真

随着计算机技术、微电子技术的高速发展,单片机在工业控制、智能仪器仪表、现代传感器、数据采集与处理、机电一体化、消费电子、家用电器、通信、办公自动化、医疗器械、计算机控制等领域应用越来越广泛。单片机作为典型、具有代表性的嵌入式系统,其应用系统设计包括硬件电路设计和软件电路设计两个方面,学习和应用过程中必须软件、硬件结合。单片机系统调试通常分为软件调试、硬件测试和整机联调三个部分。

单片机自身不具备开发功能,必须借助于开发工具。目前,国内外推出了许多基于个人计算机的单片机软或硬开发平台。硬件开发平台方面诸如开发板、实验箱、仿真器、编程器、示波器、逻辑分析仪等,但其价格不菲,开发过程烦琐。在软件支持的前提下,应用最普遍的是软件仿真开发平台,其具有方便、快捷、节约的优点。

单片机应用系统软件仿真开发平台有两个常用的工具软件:Keil 和 Proteus ISIS。Keil 主要用于单片机源程序的编辑、编译、链接以及调试;Proteus ISIS 主要用于单片机硬件电路原理图的设计以及单片机应用系统的软、硬件联合仿真调试。

本章将以 Keil μ Vision2、Proteus ISIS Professional Vision7.7 SP2 版本为例详细介绍其在单片机开发中的应用方法,并通过一个实例详细介绍 Keil 与 Proteus ISIS 的联调使用方法。



1.1 Keil 集成开发环境

一般情况下单片机常用的程序设计语言有两种:汇编语言和 C 语言。

汇编语言具有执行速度快、占存储空间少、对硬件可直接编程等特点,因而特别适合对实时性要求比较高的情况使用。使用汇编语言编程要求程序设计人员必须熟悉单片机内部结构和工作原理,编写程序麻烦一些。

与汇编语言相比,C 语言在功能、结构性、可读性、可维护性、可移植性上都有明显优势,C 语言大多数代码被翻译成目标代码后,其效率和准确性方面和汇编语言相当。特别是 C 语言的内嵌汇编功能,使 C 语言对硬件操作更加方便,并且 C 语言作为自然高级语言,易学易用,尤其是在开发大型软件时更能体现其优势,因此在单片机程序设计中得到广泛应用。

Keil μ Vision2 是德国 Keil Software 公司推出的微处理器开发平台,可以开发多种 8051 兼容单片机程序。它可以被用于工程创建、管理、编辑、编译 C 源码、汇编源程序、链接、重定位目标文件和库文件、生成 HEX 文件、调试目标程序等完整的开发流程,具有丰富的库函数和功能强大的集成开发工具,全 Windows 操作界面,所以备受用户青睐。

1.1.1 Keil μ Vision2 工作环境

正确安装后,用鼠标左键双击计算机桌面上 KEIL μ VISION2 运行图标,或用鼠标左键分别单击计算机桌面上“开始”—“所有程序”—“KEIL μ VISION2”,即可启动 Keil μ Vision2,启动界面如图 1-1 所示,进入 Keil μ Vision2 集成开发环境后,其界面如图 1-2 所示。



图 1-1 Keil μ Vision2 启动界面

从图 1-2 可以看出,Keil μ Vision2 集成开发环境与其他常用的 Windows 窗口软件类似,设置有菜单栏、可以快速选择命令的按钮工具栏、工程窗口、源代码文件窗口、对话框、信息显示窗口。Keil μ Vision2 允许同时打开浏览多个源程序文件。

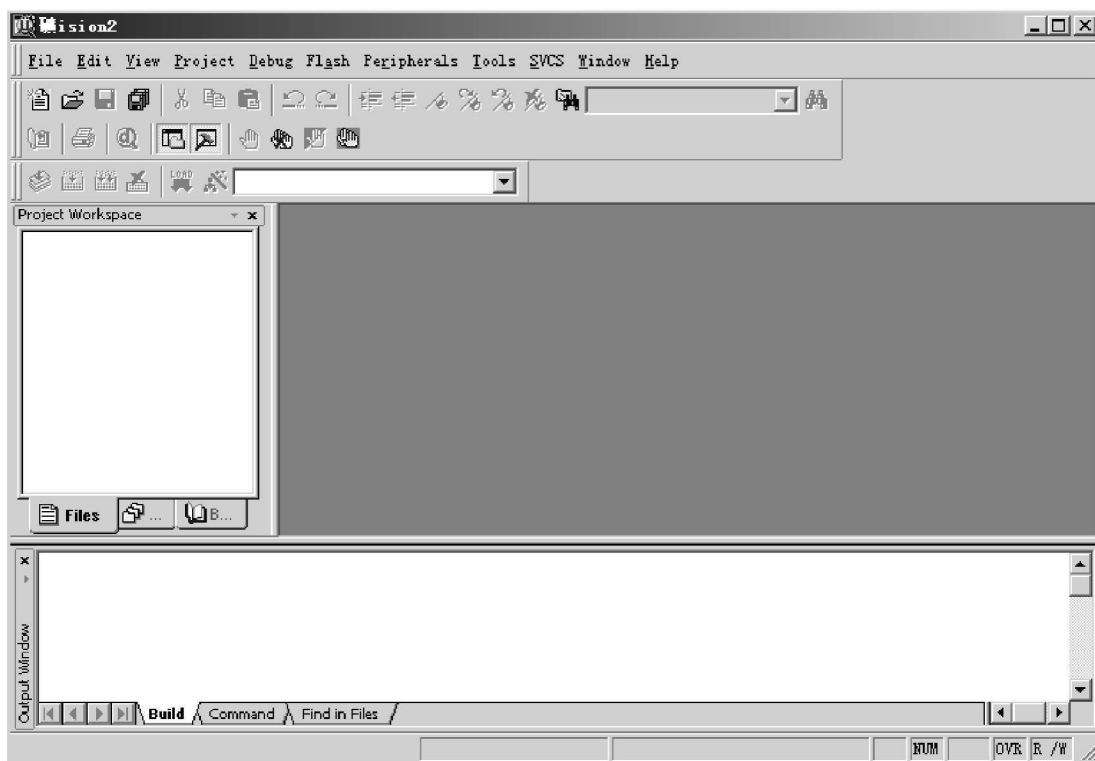
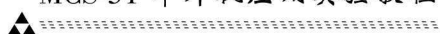


图 1-2 Keil μVision2 集成开发环境界面

Keil μVision2 IDE 提供了多种命令执行方式:①菜单栏提供了诸如文件(File)操作、编辑(Edit)操作、视图(View)操作、项目/工程(Project)操作、程序调试(Debug)、闪存(Flash)操作、片上外设寄存器设置和观察(Peripherals)、开发工具选项(Tools)、软件版本控制系统菜单(SVCS)、窗口选择和处理(Window)、在线帮助(Help)共 11 种操作菜单;②使用工具栏按钮可以快速地执行 μVision2 命令;③使用键盘快捷键也可以执行 μVision2 命令,键盘快捷键根据使用习惯等还可以重新设置。表 1-1 ~ 表 1-10 列出了 μVision2 菜单项命令、工具条图标、默认的快捷键以及对它们的描述。

表 1-1 文件菜单和命令(File)

菜单	工具条	快捷键	描 述
New		Ctrl + N	创建一个新的源程序文件
Open		Ctrl + O	打开已经存在的文件
Close			关闭当前文件
Save		Ctrl + S	保存当前文件(新建保存时需命名)
Save as			另外取名保存当前文件
Save all			保存所有文件
Device Database			维护器件库
Print Setup			设置打印机



续表

菜单	工具条	快捷键	描 述
Print		Ctrl + P	打印当前文件
Print Preview			打印预览
1-9			打开最近用过的文件
Exit			退出 μ Vision2 提示是否保存文件

表 1-2 编辑菜单和编辑器命令 (Edit)

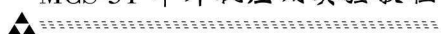
菜单	工具条	快捷键	描 述
Home			移动光标到本行的开始
End			移动光标到本行的末尾
Ctrl + Home			移动光标到文件的开始
Ctrl + End			移动光标到文件的结束
Ctrl + < -			移动光标到词的左边
Ctrl + - >			移动光标到词的右边
Ctrl + A			选择当前文件的所有文本内容
Undo		Ctrl + Z	撤销上次操作
Redo		Ctrl + Shift + Z	重复上次撤销的操作
Cut		Ctrl + X	剪切所选文本
		Ctrl + Y	剪切当前行的所有文本
Copy		Ctrl + C	复制所选文本
Paste		Ctrl + V	粘贴所剪切或复制的文本
Indent Selected Text			将所选文本右移一个制表键的距离
Unindent Selected Text			将所选文本左移一个制表键的距离
Toggle Bookmark		Ctrl + F2	设置/取消当前行的标签
Goto Next Bookmark		F2	移动光标到下一个标签处
Goto Previous Bookmark		Shift + F2	移动光标到上一个标签处
Clear All Bookmarks			取消当前文件的所有标签

续表

菜单	工具条	快捷键	描 述
Find		Ctrl + F	在当前文件中查找文本
		F3	向前重复查找
		Shift + F3	向后重复查找
		Ctrl + F3	查找光标处的单词
		Ctrl +]	寻找匹配的大括号、圆括号或方括号 (用此命令将光标放到大括号、圆括号或方括号的前面)
Replace		Ctrl + H	替换特定的字符
Find in Files			在多个文件中查找

表 1-3 视图命令 (View)

菜单	工具条	快捷键	描 述
Status Bar			显示/隐藏状态条
File Toolbar			显示/隐藏文件工具栏
Build Toolbar			显示/隐藏编译菜单条
Debug Toolbar			显示/隐藏调试工具栏
Project Window			显示/隐藏项目/工程窗口
Output Window			显示/隐藏输出窗口
Source Browser			显示/隐藏资源浏览器窗口
Disassembly Window			显示/隐藏反汇编窗口
Watch & Call Stack Window			显示/隐藏观察和访问堆栈窗口
Memory Window			显示/隐藏存储器窗口
Code Coverage Window			显示/隐藏代码报告窗口
Performance Analyzer Window			显示/隐藏性能分析窗口
Symbol Window			显示/隐藏字符变量窗口
Serial Window #1			显示/隐藏串口 1 的观察窗口
Serial Window #2			显示/隐藏串口 2 的观察窗口
Toolbox			显示/隐藏自定义工具箱
Periodic Window Update			在程序运行时周期刷新调试窗口



续表

菜单	工具条	快捷键	描 述
Workbook Mode			显示/隐藏工作簿模式
Include Dependencies			显示/隐藏头文件
Options			设置颜色、字体、快捷键和编辑器的选项

表 1-4 项目菜单和项目命令 (Project)

菜单	工具条	快捷键	描 述
New Project			创建新工程
Import Vision1 Project			转化 μ Vision1 的工程
Open Project			打开一个已经存在的工程
Close Project			关闭当前的工程
Target Environment			定义工具、包含文件和库文件的路径
Targets, Groups, Files			维护一个项目的对象文件组 and 文件
Select Device for Target			从设备数据库中选择对象的 CPU
Remove			从工程中移走一个组或文件
Options		Alt + F7	设置对象、组或文件的工具选项
File Extensions			选择不同文件类型的扩展名
Build Target		F7	编译链接修改过的文件并生成应用
Rebuild Target			重新编译链接所有的文件并生成应用
Translate		Ctrl + F7	编译当前文件
Stop Build			停止生成应用的过程
1-9			打开最近打开过的工程

表 1-5 调试菜单和调试命令 (Debug)

菜单	工具条	快捷键	描 述
Start/Stop Debugging		Ctrl + F5	开始/停止调试模式
Go		F5	运行程序,直到遇到一个断点
Step		F11	单步执行程序,遇到子程序则进入

续表

菜单	工具条	快捷键	描 述
Step over		F10	单步执行程序,跳过子程序
Step out of Current function		Ctrl + F11	执行到当前函数的结束
Run to Cursor line		Ctrl + F10	从程序指针处运行到光标处
Stop Running		Esc	程序停止运行
Breakpoints			打开断点对话框
Insert/Remove Breakpoint			插入/清除当前行的断点
Enable/Disable Breakpoint			使能/禁止当前行的断点
Disable All Breakpoints			禁止所有的断点
Kill All Breakpoints			取消所有的断点
Show Next Statement			显示下一条指令/语句
Enable/Disable Trace Recording			使能/禁止程序运行轨迹的记录
View Trace Records			显示执行过的指令
Memory Map			打开存储器空间配置对话框
Performance Analyzer			打开设置性能分析器的对话框
Inline Assembly			对某一个行重新汇编,并可以修改汇编代码
Function Editor			编辑调试函数和调试配置文件

表 1-6 外围器件菜单 (Peripherals)

菜单	工具条	快捷键	描 述
Reset CPU			复位 CPU
Interrupt			中断
I/O-Ports			I/O 口, Port 0 ~ Port 3
Serial			串行口
Timer			Timer 0 ~ Timer 2 定时器



表 1-7 工具菜单 (Tool)

菜单	工具条	快捷键	描 述
Customize Tools Menu			添加用户程序到工具菜单中

表 1-8 软件版本控制系统菜单 (SVCS)

菜单	工具条	快捷键	描 述
Configure Version Control			配置软件版本控制系统的命令

表 1-9 视窗菜单 (Window)

菜单	工具条	快捷键	描 述
Cascade			以互相重叠的形式排列文件窗口
Tile Horizontally			以不互相重叠的形式水平排列文件窗口
Tile Vertically			以不互相重叠的形式垂直排列文件窗口
Arrange Icons			排列主框架底部的图标
Split			把当前的文件窗口分割为几个
1-9			激活指定的窗口对象

表 1-10 帮助菜单 (Help)

菜单	工具条	快捷键	描 述
Help topics			打开在线帮助
About Vision			显示版本信息和许可证信息

1.1.2 Keil 工程的创建

使用 Keil μ Vision2 IDE 的项目/工程开发流程和其他软件开发项目的流程极其相似,具体步骤如下:

- (1) 新建一个工程,从设备器件库中选择目标器件(CPU),配置工具设置;
- (2) 用 C51 语言或汇编语言编辑源程序;
- (3) 用工程管理器添加源程序;
- (4) 编译、链接源程序,并修改源程序中的错误;
- (5) 生成可执行代码,调试运行应用。

为了介绍方便,下面以一个简单实例——单片机流水灯来介绍 Keil 工程的创建过程。

1. 源程序文件的建立

执行菜单命令 File→new 或者单击工具栏的新建文件按钮,即可在项目窗口的右侧打开一个默认名为 Text1 的空白文本编辑窗口,录入、编辑程序代码,在该窗口中输入以下 C 语言代码:

```
/* 定义头文件及变量初始化 */
#include <reg51.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
uchar temp = 0xFE;           //temp 中先装入 LED1 亮、LED2 ~ LED8 灭的数据
                              //(二进制的 11111110)
uchar count = 0x64;          //定义计数变量初值为 100,计数 100 个 10ms,即 1 s

/* T0 中断服务子程序 */
void timer0(void) interrupt 1 using 1
{
    TH0 = -5000/256;          //重装初值
    TL0 = -5000%256;
    count--;                  //1 s 时间未到,继续计数
    if(count == 0)
    {
        count = 0x64;         //1 s 时间到,重置 count 计数初值为 100
        temp = _crol_(temp,1); //将点亮的 LED 循环左移一位
    }
}

/* 主程序 */
void main(void)
{
    P1 = 0xff;                //初始状态,所有 LED 熄灭
    TMOD = 0x01;              //设置 T0 工作方式 1
    TH0 = -5000/256;          //设置 10 ms 计数初值
    TL0 = -5000%256;
    EA = 1;                   //开放总中断
    ET0 = 1;                  //开放 T0 中断
    TR0 = 1;                  //启动 T0
    while(1)                  //死循环
    {
        P1 = temp;            //把 temp 数据送 P1 口
    }
}
```



```

    }
}

```

上述程序的汇编代码如下:

```

        ORG      0000H           ;单片机上电后程序入口地址
        SJMP     START          ;跳转到主程序存放地址处
        ORG      000BH           ;定时器 T0 入口地址
        SJMP     TOSVR          ;跳转到定时器 T0 中断服务程序存放地址处
        ORG      0030H           ;设置主程序开始地址
START: MOV      SP, #60H         ;设置堆栈起始地址为 60H
        MOV      P1, #0FFH       ;初始状态,所有 LED 熄灭
        MOV      A, #0FEH        ;ACC 中先装入 LED1 亮、LED2 ~ LED8 灭的数据
                                   ;(二进制的 11111110)
        MOV      R0, #64H        ;计数 100 个 10ms,即 1 s
        MOV      TMOD, #01H      ;设置 T0 工作方式 1
        MOV      TH0, #0ECH      ;设置 10ms 计数初值
        MOV      TL0, #78H
        SETB     EA              ;开放总中断
        SETB     ET0             ;开放 T0 中断
        SETB     TR0             ;启动 T0
DISP: MOV      P1, A             ;把 ACC 数据送 P1 口
        SJMP     DISP

/* T0 中断服务子程序 */
TOSVR: MOV      TL0, #78H        ;重装初值
        MOV      TH0, #0ECH
        DJNZ     R0, LOOP        ;1 s 时间未到,继续计数
        MOV      R0, #64H        ;1 s 时间到,重置 R0 计数初值为 100
        RL       A               ;将点亮的 LED 循环左移
LOOP: RETI                    ;子程序返回
        END                      ;程序结束

```

μVision2 与其他文本编辑器类似,同样具有录入、删除、选择、复制、粘贴等基本的文本编辑功能。需要说明的是,源文件就是一般的文本文件,不一定使用 Keil 软件编写,可以使用任意文本编辑器编写,需要注意的是,Keil 的编辑器对汉字的支持不好,建议使用记事本之类的编辑软件进行源程序的输入,然后按要求保存,以便添加到工程中。

在编辑源程序文件过程中,为防止断电丢失,须时刻保存源文件,第一次执行菜单命令 File→Save 或者单击工具栏的保存文件按钮 ,将打开如图 1-3 所示的对话框,在“文件名”对话框中输入源文件的命名。注意必须加上后缀名(汇编语言源程序一般用 .ASM 或 .A51 为后缀名,C51 语言文件用 .c 为后缀名),这里将源程序文件保存为 Example.c。

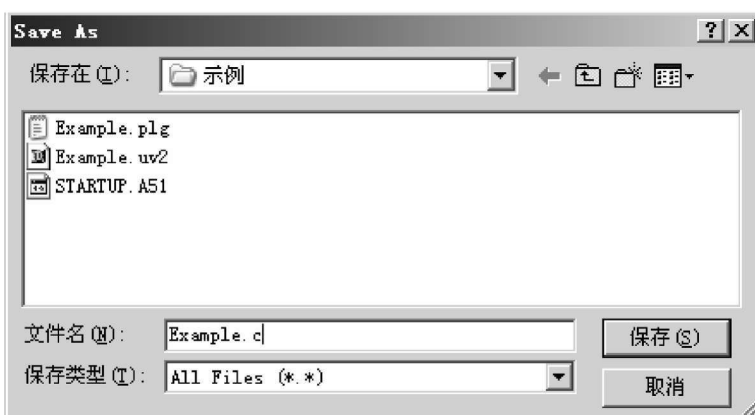


图 1-3 命名并保存新建源程序文件

2. 建立工程文件

Keil 支持数百种 CPU, 而这些 CPU 的特性并不完全相同, 在工程开发中, 并不是仅有一个源程序文件就行了, 还必须为工程选择 CPU, 以确定编译、汇编、链接的参数, 指定调试的方式, 有一些项目还会有多个文件组成等。因此, 为管理和使用方便, Keil 使用工程 (project) 这一概念, 即将源程序 (C51 或汇编)、头文件、说明性的技术文档等都放置在一个工程里, 只能对工程而不能对单一的源文件进行编译 (汇编) 和链接等操作。

启动 Keil μ Vision2 IDE 后, μ Vision2 总是打开用户上一次处理的工程, 要关闭它可以执行菜单命令 Project $\rightarrow$ Close Project。建立新工程可以通过执行菜单命令 Project $\rightarrow$ New Project, 此时将出现如图 1-4 所示的 Create New Project 对话框, 要求给将要建立的工程在“文件名”对话框中输入名字, 这里假定将工程文件命名为 Example, 并选择保存目录, 不需要扩展名。

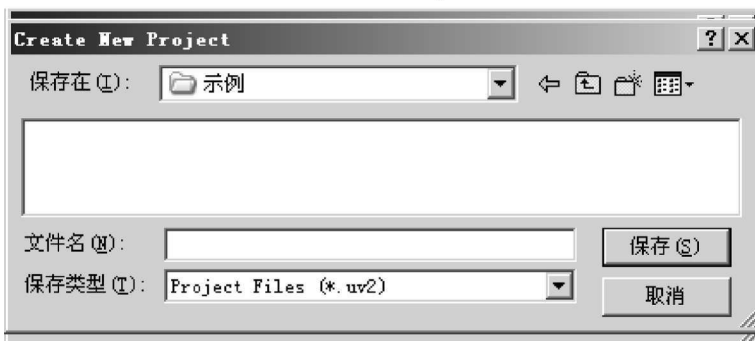


图 1-4 建立新工程

单击“保存”按钮, 打开如图 1-5 所示的 Select Device for Target ‘Target 1’ 的第二个对话框, 此对话框要求选择目标 CPU (即所用芯片的型号), 列表框中列出了 μ Vision2 支持的以生产厂家分组的所有型号的 CPU。Keil 支持的 CPU 很多, 这里选择的是 Atmel 公司生产的 AT89S51 单片机, 然后再单击“确定”按钮, 回到主界面。

另外, 如果在选择完目标 CPU 后想重新改变目标 CPU, 可以执行菜单命令 Project $\rightarrow$ Select Device for..., 在随后出现的目标设备选择对话框中重新加以选择。由于不同厂家许多型号的 CPU 性能相同或相近, 因此, 如果所需的目标 CPU 型号在 μ Vision2 中找不到, 可以选择其他公司生产的相近型号。

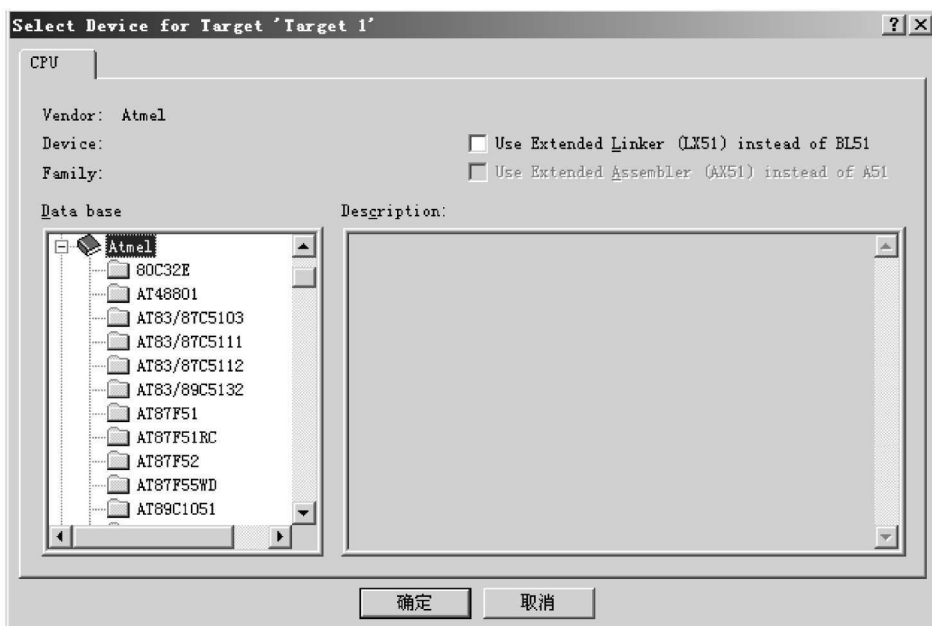


图 1-5 选择目标 CPU

3. 添加源程序文件到工程中

选择完目标 CPU 后,在工程窗口中,出现了“Target 1”,前面有“+”号,点击“+”号展开,可以看到下一层的“Source Group 1”,这时的工程还是一个空的工程,没有任何源程序文件,前面录入编辑好的源程序文件需手工添加,鼠标左键单击“Source Group 1”使其反白显示,然后,单击鼠标右键,出现一个下拉菜单,如图 1-6 所示,选中其中的“Add file to Group ‘Source Group 1’”,弹出一个对话框,要求添加源文件。注意,在该对话框下面的“文件类型”默认为 C SOURCE FILE(*. C),也就是以 C 为扩展名的文件,假如所要添加的是汇编源程序文件,则在列表框中将找不到,需将文件类型设置一下,单击对话框中“文件类型”后的下拉列表,找到并选中“ASM SOURCE FILE(*. A51, *. ASM)”,这样,在列表框中就可以找到汇编源程序文件了。

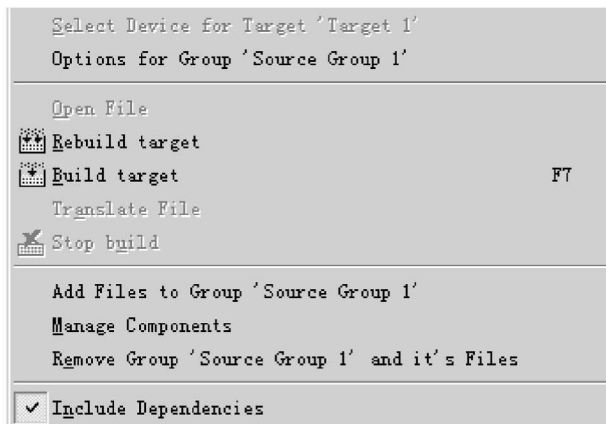


图 1-6 加入文件

双击 Example. c 文件,将文件加入工程,添加源程序文件后的工程如图 1-7 所示,注意,在