

★ 服务热线: 400-615-1233
★ 配套精品教学资料包
★ www.huatengedu.com.cn

DANPIANJI C YUYAN XIANGMUSHI JIAOCHENG

单片机C语言项目式教程 (第2版)



策划编辑: 高 宇
责任编辑: 陈坤朋
封面设计: 黄燕美



定价: 59.80元



“十四五”职业教育国家规划教材

单片机C语言项目式教程
第2版

主编 于宝明 龚运新 于炜芳

北京邮电大学出版社



X-B

DANPIANJI C YUYAN XIANGMUSHI JIAOCHENG

单片机C语言项目式教程 (第2版)

主编 于宝明 龚运新 于炜芳



北京邮电大学出版社
www.buptpress.com



“十四五” 职业教育国家规划教材



DANPIANJI C YUYAN XIANGMUSHI JIAOCHENG

单片机C语言项目式教程

(第2版)

主编 于宝明 龚运新 于炜芳



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书包括 15 个项目,分别为单片机识别与展望、单片机开发工具、单片机最小系统设计、单片机内部结构、C51 程序设计基础、C51 程序设计、LED 彩灯定时器延时控制、LED 彩灯中断控制器制作、LED 彩灯串行控制器制作、串行存储器扩展设计、I/O 端口扩展电路设计、D/A 和 A/D 转换器电路接口设计、单片机电动机控制器制作、步进电机控制器制作和单片机产品开发实例。

本书既可作为高等职业教育机电一体化技术、电气自动化技术、电子信息工程技术等专业的教材,也可供单片机应用开发技术人员参考。

图书在版编目(CIP)数据

单片机 C 语言项目式教程 / 于宝明, 龚运新, 于炜芳
主编. -- 2 版. -- 北京: 北京邮电大学出版社, 2024.
(2026 重印). -- ISBN 978-7-5635-7366-0
I. TP368.1; TP312.8
中国国家版本馆 CIP 数据核字第 20242962VT 号

策划编辑: 高 宇 责任编辑: 陈坤朋 封面设计: 黄燕美

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 三河市骏杰印刷有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 19 插页 1

字 数: 393 千字

版 次: 2012 年 5 月第 1 版 2024 年 10 月第 2 版

印 次: 2026 年 3 月第 3 次印刷

ISBN 978-7-5635-7366-0

定 价: 59.80 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

服务电话:400-615-1233

本书二维码资源列表

项目	页码	类型	具体内容	项目	页码	类型	具体内容
一	1	视频	“中国龙芯之母”黄令仪	五	77	课件	C51 程序设计基础
	1	视频	大国工匠·匠心报国		85	源程序	计算正整数次幂函数
	2	课件	单片机识别与展望		103	源程序	求一元二次方程的根
	2	视频	AT89S51 单片机引脚		104	源程序	观察方程求解结果
	9	拓展	认识单片机		105	源程序	计算某月的天数
二	13	课件	单片机开发工具		106	源程序	观察天数求解结果
	14	视频	Keil C51 软件的使用		108	源程序	求自然数累加和
	39	视频	Proteus 软件的使用		108	源程序	while 语句结果
	42	拓展	QTH 调试实例		109	源程序	求自然数累加和
	43	拓展	使用项目管理方式开发		110	源程序	do-while 语句结果
	45	视频	另一种闪烁的 LED 灯电路		111	源程序	求自然数累加和
	47	拓展	单片机学习网站		112	源程序	求自然数累加和
三	48	视频	电路原理图设计		112	源程序	求等差数列
	49	视频	单片机最小系统概述		113	源程序	观察 goto 语句结果
	49	课件	单片机最小系统设计		114	源程序	终止循环
	49	手册	74LS373		115	源程序	return 语句使用
	54	源程序	控制指示灯顺序点亮		115	源程序	观察 return 语句结果
	56	视频	8 只 LED 灯原理图搭建		118	手册	三极管 9013
	57	视频	16 只 LED 灯原理图搭建		121	源程序	摄氏温度转华氏温度
	57	视频	24 只 LED 灯原理图搭建		124	课件	C51 程序设计
	58	视频	32 只 LED 灯原理图搭建	124	源程序	一重循环	
	58	拓展	元器件的装配方式与布局	124	源程序	二重循环	
	58	视频	PCB 设计	124	源程序	三重循环	
四	61	视频	芯片改变世界	六	125	源程序	用 while 实现二重循环
	62	课件	单片机内部结构		125	源程序	用 do-while 实现二重循环
	62	视频	单片机内部结构		126	源程序	指示灯顺序点亮
	73	源程序	置位 128 位		129	源程序	实现指示灯顺序点亮
	74	手册	AT89C2051		130	源程序	指示灯多向顺序点亮 1

续表

项目	页码	类型	具体内容	项目	页码	类型	具体内容
六	132	源程序	指示灯多向顺序点亮 2	十	198	源程序	AT24C16 读写子程序
	134	源程序	指示灯多向顺序点亮 3		200	动画	SPI 总线技术
	134	源程序	指示灯多向顺序点亮 4		200	动画	串行传输比较
	136	源程序	字节的 8 位进行置位操作		207	手册	AT93C46
	137	源程序	4 个字节进行置位操作	十一	217	课件	I/O 端口扩展电路设计
	139	源程序	输出 Hello World		218	源程序	开关量 I/O 设计
	140	源程序	做加法		226	手册	七段数码管
七	144	课件	LED 彩灯定时器延时控制	十一	226	动画	七段 LED 译码器
	149	源程序	初始化程序		226	视频	另一种液晶显示原理图
	150	源程序	延时 1 秒子程序(查询方式)		228	源程序	动态显示程序
	150	源程序	延时 1 秒子程序(中断方式)		232	源程序	矩阵键盘输入子程序
	152	源程序	T0 工作于方式 0	十二	252	课件	D/A 和 A/D 转换器电路接口
	153	源程序	T0 工作于方式 1		252	手册	DAC0832
	154	源程序	T0 工作于方式 2		254	源程序	运放输出锯齿波电压
	155	源程序	T0 工作于方式 3		256	源程序	图形显示器光栅移动位置
八	160	课件	LED 彩灯中断控制器制作	十三	259	源程序	用中断法读取数据
	160	拓展	中断及中断源		264	课件	单片机电动机控制器制作
	169	源程序	使用一个外部中断		265	源程序	单台电动机控制器
	171	源程序	同时用两个外部中断		266	源程序	多台电动机控制器
	173	源程序	从右向左依次点亮灯	270	视频	直流稳压电源的使用	
九	176	视频	华为 5G 技术	十四	273	课件	步进电机控制器制作
	177	课件	LED 彩灯串行控制器制作		273	动画	步进电机工作原理
	182	手册	MAX232		281	视频	步进电机的另一种原理图
	185	视频	另一种串行通信仿真	282	视频	示波器的使用	
	187	拓展	USB 通用串行总线技术	十五	285	课件	单片机产品开发实例
	188	动画	串行通信		285	手册	74LS138
	191	拓展	单片机串行口应用举例		293	视频	PCB 布局
十	193	课件	串行存储器扩展设计		293	拓展	单片机软硬件抗干扰
	193	动画	I ² C 总线技术				

单片机是一种将中央处理器(CPU)、存储器、输入/输出(I/O)接口、定时器、计数器等集成在一块芯片上的微型计算机系统。单片机体积小,制造成本较低,同时具有较高的可靠性,能够在恶劣的工业环境中稳定工作,具备强大的处理能力和丰富的接口。单片机被广泛应用于自动化控制、传感器接口、数据采集与监控、过程控制与调节、智能仪表与监控、通信与网络连接等工业场景,同时也可被应用于智能家居、交通运输和医疗保健等生活场景。

党的二十大报告指出:“建设现代化产业体系。坚持把发展经济的着力点放在实体经济上,推进新型工业化,加快建设制造强国、质量强国、航天强国、交通强国、网络强国、数字中国。实施产业基础再造工程和重大技术装备攻关工程,支持专精特新企业发展,推动制造业高端化、智能化、绿色化发展。巩固优势产业领先地位,在关系安全发展的领域加快补齐短板,提升战略性资源供应保障能力。推动战略性新兴产业融合集群发展,构建新一代信息技术、人工智能、生物技术、新能源、新材料、高端装备、绿色环保等一批新的增长引擎。”

单片机是一门理论性、实践性和综合性都很强的学科,需要以模拟电子技术、数字电子技术、电气控制技术、电力电子技术等学科作为知识背景,并与计算机软、硬件有机结合。

本书是作者多年理论、实践教学及产品研发经验的结晶。在编写过程中,始终将理论知识与产品开发有机结合,从单片机最小系统开始,逐步扩展功能,从小到大,从简单到复杂,采用理论—仿真—实践的顺序编写,编写思路系统、完整、清晰。

本书从实用角度出发,加强了设计性环节的指导,所有程序均可在 Keil C 仿真软件中调试成功,具有较强的可读性,便于学生自学;扩展电路可在 Proteus 软件中仿真成功,更接近真实的工作环境。若条件许可,教师可以将教学安排在计算机机房或多媒体教室进行,边讲解边演示,结合多媒体课件,使教学内容直观形象、通俗易懂,特别是进行软件仿真、硬件仿真与产品模拟时,效果会更好。

本书在此次再版中,除了对第 1 版内容进行修改、订正、更新外,还融入了思政元素,主要讲述了中国单片机产业发展的新技术、新应用,培养学生的民族自豪感和艰苦奋斗的创业精神。

本书包括 15 个项目,分别为单片机识别与展望、单片机开发工具、单片机最小系统设计、单片机内部结构、C51 程序设计基础、C51 程序设计、LED 彩灯定时器延时控制、LED 彩灯中断控制器制作、LED 彩灯串行控制器制作、串行存储器扩展设计、I/O 端口扩展电路设计、D/A 和 A/D 转换器电路接口设计、单片机电动机控制器制作、步进电机控制器制作和单片机产品开发实例。

本书由于宝明、龚运新、于炜芳任主编。

由于编者水平有限,书中难免存在疏漏和不足之处,敬请广大读者批评指正。

目前, MCS-51 系列单片机在我国电子、计算机、自动化、电气控制、机电一体化、智能仪器仪表等行业得到了广泛应用, 各高职高专院校的相关专业均开设了相关课程。单片机是一门理论性、实践性和综合性都很强的学科, 它需要模拟电子技术、数字电子技术、电气控制、电力电子技术等学科作为知识背景, 同时它也是计算机软、硬件有机结合的产物。

本教材是作者多年理论、实验教学及产品研发经验的结晶。在教材编写过程中, 始终将理论、实验、产品开发三者有机结合, 从单片机最小系统开始, 逐步扩展功能, 从小到大, 从简单到复杂, 采用理论—仿真—实践顺序编写, 给学生一种系统、完整、清晰的学习思路。

本教材从实用角度出发, 加强了设计性环节的指导, 所有程序均可在 Keil C 仿真软件中调试成功, 增加了知识的真实性和可读性, 便于自学; 扩展电路可在 Proteus 软件中仿真成功, 更接近真实的工作环境。若条件许可, 教学可以安排在计算机机房或多媒体教室进行, 边讲解边演示, 结合多媒体课件, 使教学内容直观形象、通俗易懂, 特别是进行软件仿真、硬件仿真与产品模拟时, 效果会更好。

本教材包括 15 个项目, 分别为单片机识别与展望、单片机开发工具、单片机最小系统设计、单片机内部结构、C51 程序设计基础、C51 程序设计、LED 彩灯定时器延时控制、LED 彩灯中断控制器制作、LED 彩灯串行控制器制作、串行存储器扩展设计、I/O 端口扩展电路设计、D/A 和 A/D 转换器电路接口设计、单片机电动机控制器制作、步进电机控制器制作和单片机产品开发实例。

本教材由无锡科技职业学院龚运新副教授任主编。具体编写分工如下: 项目一和项目四由鄂东职业技术学院徐波编写, 项目二由西安技师学院高德龙编写, 项目三、项目十和项目十二由咸阳职业技术学院党世红编写, 项目五、项目六和项目九由龚运新编写, 项目七和项目八由湖北黄冈职业技术学院秦佳编写, 项目十一和项目十五由无锡职业技术学院罗惠敏编写, 项目十三和项目十四由无锡城市职业技术学院任靖福编写。全书由许昌职业技术学院宁玉伟副教授主审并提出宝贵意见。

由于本教材涉及的专业面较广, 以及编者水平有限, 书中难免存在一些疏漏和不足之处, 敬请读者批评指正。

编 者

项目一 单片机识别与展望	1	项目实践	73
相关知识	2	拓展知识	74
一、芯片引脚排列识别	2	复习思考题	75
二、芯片引脚功能	3	项目五 C51 程序设计基础	76
三、I/O 端口内部结构	5	相关知识	77
项目实践	8	一、C 语言基础知识	77
拓展知识	8	二、函数	84
复习思考题	12	三、运算符与表达式	96
项目二 单片机开发工具	13	项目实践	100
相关知识	14	拓展知识	117
一、软件仿真	14	复习思考题	122
二、硬件仿真	39	项目六 C51 程序设计	123
三、在线仿真	41	相关知识	124
项目实践	44	一、循环程序	124
拓展知识	45	二、单片机 I/O 端口控制程序	126
复习思考题	47	项目实践	135
项目三 单片机最小系统设计	48	拓展知识	135
相关知识	49	复习思考题	142
一、单片机最小系统概述	49	项目七 LED 彩灯定时器延时	
二、单片机最小系统的应用	51	控制	143
项目实践	56	相关知识	144
拓展知识	58	一、定时/计数器的使用方法	144
复习思考题	60	二、定时/计数器的初值计算	147
项目四 单片机内部结构	61	三、定时/计数器的初始化	149
相关知识	62	项目实践	151
一、MCS-51 系列单片机内部结构	62	拓展知识	156
二、MCS-51 系列单片机存储器		复习思考题	158
结构	66		



项目八 LED 彩灯中断控制器

制作	159
相关知识	160
一、单片机中断知识	160
二、MCS-51 系列单片机中断系统的初始化	166
项目实践	169
拓展知识	174
复习思考题	175

项目九 LED 彩灯串行控制器

制作	176
相关知识	177
一、串行口控制寄存器	177
二、串行口工作方式	178
三、波特率	180
项目实践	181
拓展知识	187
复习思考题	191

项目十 串行存储器扩展设计 ...

相关知识	193
一、I ² C 总线	193
二、AT24C 系列串行 E ² PROM ...	196
项目实践	197
拓展知识	200
复习思考题	215

项目十一 I/O 端口扩展电路

设计	216
相关知识	217
一、开关量 I/O 设计	217
二、8255 芯片的使用	221
项目实践	225
拓展知识	234
复习思考题	249

项目十二 D/A 和 A/D 转换器

电路接口设计	251
相关知识	252
一、D/A 转换器电路接口设计 ...	252
二、A/D 转换器电路接口设计 ...	256
项目实践	259
拓展知识	260
复习思考题	262

项目十三 单片机电动机控制器

制作	263
相关知识	264
一、单台电动机控制器制作	264
二、多台电动机控制器制作	266
项目实践	269
拓展知识	270
复习思考题	271

项目十四 步进电机控制器制作 ...

相关知识	273
一、步进电机概述	273
二、D306 三相混合式步进电机驱动器	274
三、步进电机控制系统	278
项目实践	281
拓展知识	282
复习思考题	283

项目十五 单片机产品开发实例 ...

相关知识	285
一、硬件设计	285
二、软件设计	287
项目实践	293
拓展知识	293
复习思考题	295

参考文献.....



项目一

单片机识别与展望

知识目标

- (1)了解各种单片机的外部结构及封装形式；
- (2)了解 I/O 端口的内部结构；
- (3)掌握各引脚的功能。

技能目标

- (1)学会运用识别单片机各引脚的方法；
- (2)灵活使用各引脚。

学思园地

84 岁老太研龙芯,突破壁垒扬国威

随着中国经济的发展,中美贸易战爆发,芯片“卡脖子”成为中国发展面临的难题之一。面对美国的刁难,中国从来不会示弱。一位 84 岁的老太太黄令仪,呕心沥血,终于研究出龙芯,打破了美国在芯片方面的技术垄断,为国家做出了巨大贡献。青春气贯长虹,进取压倒苟且,奋斗二字,应当刻在每个中国人的基因里。我们不能辜负先烈们为这块土地所付出的血与泪,我们一定会让祖国母亲走向强大,永远立于世界民族之林。



视频
“中国龙芯之
母”黄令仪

项目描述

本项目只限于单片机芯片的识别,芯片的制造和封装是微电子制造技术专业专门学习的内容。对于单片机应用的技术人员而言,只限于单片机如何应用,而且在应用时主要掌握芯片的使用方法、封装形式和各引脚的功能。封装形式在制作单片机产品的印制电路板时要用到,各引脚的功能在设计产品时用到。



视频
大国工匠·匠
心报国



相关知识

一、芯片引脚排列识别

单片机芯片有针脚式封装(DIP)与表面贴片式封装(SMD)两大类。制作单片机产品时,先按芯片封装形式在电子CAD软件中设计好印制电路板(PCB),然后送到专门制作PCB的厂家制作出PCB,再在焊接场地焊接好芯片,组装成电子产品。针脚式封装的元件体积较大,电路板必须钻孔才能安装元件,焊接时需要人工插入单片机芯片,再经过锡炉或喷锡(也可手焊)焊接好芯片。这种芯片成本较高,一般用于小批量的手工焊接产品。较新的设计都是采用体积小的表面贴片式元件,这种元件不必钻孔,用钢膜将半熔状锡膏倒在PCB上,再把芯片放上,用热风枪加热,即可将芯片焊接在PCB上。这种封装加工简单,焊接容易,成本较低,一般用于大批量的自动贴片机焊接产品。

单片机采用40引脚的双列直插封装方式时,封装及外形如图1-1所示,图1-1(a)为实物图,图1-1(b)为原理图库中的器件图,图1-1(c)为PCB封装库中的封装图。为了识别引脚,芯片都开有缺口,识别时将芯片缺口朝上,引脚向下放置于桌面上,左边第一脚即为引脚“1”,有的芯片在开口的同时还在引脚“1”处刻有圆圈作为标记,按此标记逆时针依次排列引脚。

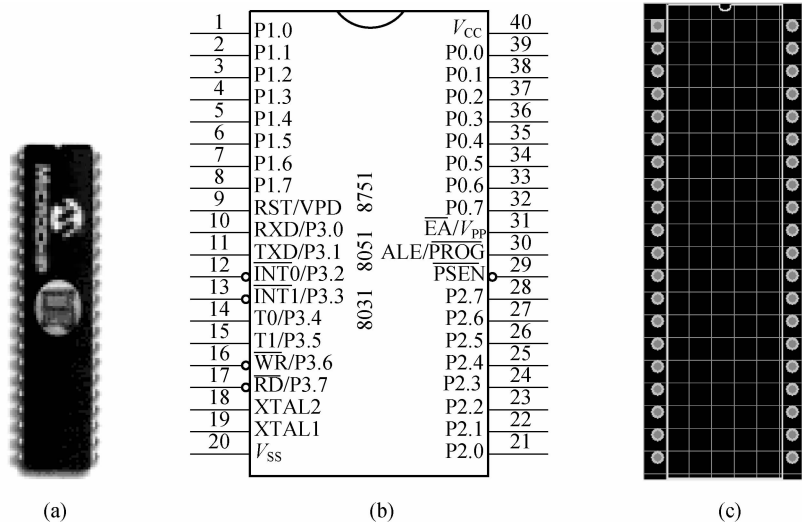


图 1-1 单片机双列直插封装图

单片机贴片封装如图1-2所示,图1-2(a)为原理图库中的器件图,图1-2(b)为PCB封装库中的封装图。为了识别引脚,方形芯片有一个角被切除,识别时将芯片缺角朝左上方,引脚向下放置于桌面上,左边第一脚即为引脚“1”,有的芯片在切角的同时还在引脚“1”处刻有圆圈作为标记。



课件

单片机识别与
展望



视频

AT89S51 单
片机引脚

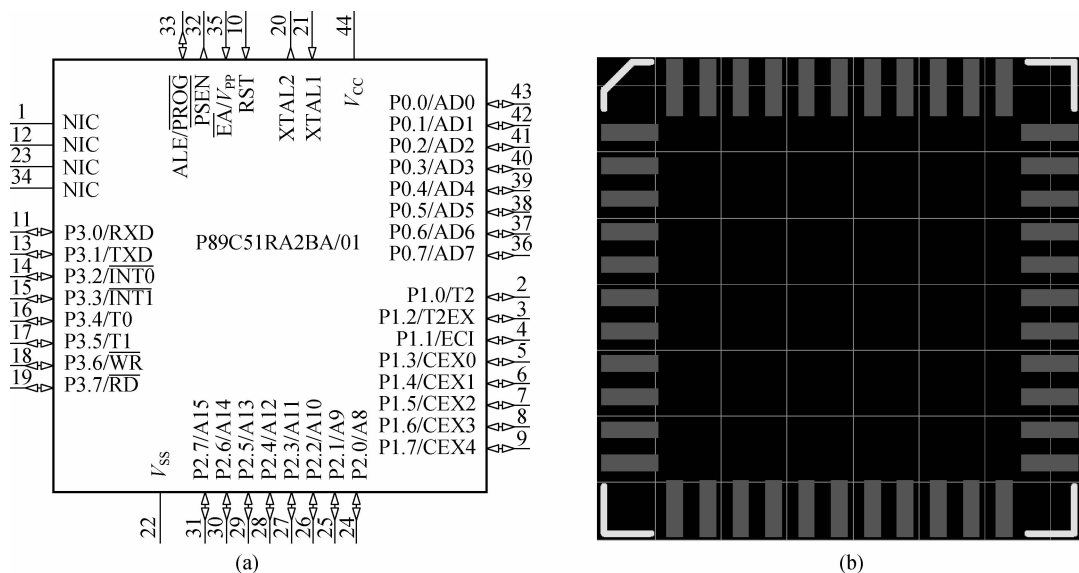


图 1-2 单片机贴片封装图

二、芯片引脚功能

图 1-1 所示以 51 芯片为内核的单片机的 40 个引脚功能说明如下。

1. 主电源引脚 V_{SS} 和 V_{CC}

1) V_{SS}

V_{SS} (第 20 引脚)接地。

2) V_{CC}

正常操作时, V_{CC} (第 40 引脚)为 +5 V 电源。

2. 外接晶振引脚 XTAL1 和 XTAL2

1) XTAL1

XTAL1(第 19 引脚)为内部振荡电路反相放大器的输入端,是外接晶体的一个引脚。当采用外部振荡器时,此引脚接地。

2) XTAL2

XTAL2(第 18 引脚)为内部振荡电路反相放大器的输出端,是外接晶体的另一个引脚。当采用外部振荡器时,此引脚接外部振荡源。

3. 控制或与其他电源复用引脚

1) RST/VPD

当振荡器运行时,在 RST/VPD(第 9 引脚)上出现两个机器周期的高电平(由低到高跳变),将单片机复位;在 V_{CC} 掉电期间,此引脚可接备用电源,由 VPD 向内部提供备用电源,以保持单片机内部 RAM 中的数据。

2) ALE/ $\overline{\text{PROG}}$

正常操作时, ALE/ $\overline{\text{PROG}}$ (第 30 引脚)为 ALE 功能(允许地址锁存),用于把地址的低



字节锁存到外部锁存器。ALE 引脚以不变的频率(振荡器频率的 1/6)周期性地发出正脉冲信号,可用作对外输出的时钟或用于定时。每当访问外部数据存储器时,将跳过一个 ALE 脉冲,ALE 端可以驱动(吸收或输出电流)8 个 LSTTL 负载。对于 EPROM 型单片机,在 EPROM 编程期间,此引脚接收编程脉冲($\overline{\text{PROG}}$ 功能)。

3) $\overline{\text{PSEN}}$

$\overline{\text{PSEN}}$ (第 29 引脚)为外部程序存储器读选通信号输出端,在从外部程序存储器读取指令(或数据)期间, $\overline{\text{PSEN}}$ 在每个机器周期内两次有效。 $\overline{\text{PSEN}}$ 同样可以驱动 8 个 LSTTL 负载。

4) $\overline{\text{EA}}/V_{\text{PP}}$

$\overline{\text{EA}}/V_{\text{PP}}$ (第 31 引脚)为内部程序存储器和外部程序存储器选择端。当 $\overline{\text{EA}}/V_{\text{PP}}$ 为高电平时,访问内部程序存储器;当 $\overline{\text{EA}}/V_{\text{PP}}$ 为低电平时,则访问外部程序存储器。对于 EPROM 型单片机,在 EPROM 编程期间,此引脚上加 21 V 编程电源(V_{PP})。

4. 输入/输出引脚

1) P0 口

P0 口(P0.0~P0.7,对应芯片第 39~32 引脚)是一个 8 位漏极开路型双向 I/O 端口,在访问外部存储器时,它是分时传送的低字节地址和数据总线,P0 口能以吸收电流的方式驱动 8 个 LSTTL 负载。

2) P1 口

P1 口(P1.0~P1.7,对应芯片第 1~8 引脚)是一个带有内部提升电阻的 8 位准双向 I/O 端口,能驱动(吸收或输出电流)4 个 LSTTL 负载。

3) P2 口

P2 口(P2.0~P2.7,对应芯片第 21~28 引脚)是一个带有内部提升电阻的 8 位准双向 I/O 端口,在访问外部存储器时,它输出高 8 位地址。P2 口可以驱动(吸收或输出电流)4 个 LSTTL 负载。

4) P3 口

P3 口(P3.0~P3.7,对应芯片第 10~17 引脚)是一个带有内部提升电阻的 8 位准双向 I/O 端口,能驱动(吸收或输出电流)4 个 LSTTL 负载。P3 口还用于第二功能,具体内容参见表 1-1。

表 1-1 P3 口的第二功能

引脚功能	第二功能	引脚功能	第二功能
P3.0	RXD——串行输入(数据接收)口	P3.4	T0——定时器 0 外部输入
P3.1	TXD——串行输出(数据发送)口	P3.5	T1——定时器 1 外部输入
P3.2	$\overline{\text{INT0}}$ ——外部中断 0 输入	P3.6	$\overline{\text{WR}}$ ——外部数据存储器写选通信号输出
P3.3	$\overline{\text{INT1}}$ ——外部中断 1 输入	P3.7	$\overline{\text{RD}}$ ——外部数据存储器读选通信号输入

图 1-2 所示贴片封装单片机共有 44 只引脚,其中有 4 只空引脚,其他引脚功能与上述相同。



三、I/O 端口内部结构

I/O 端口又称为 I/O 接口,也称为 I/O 通道或 I/O 通路,是 MCS-51 系列单片机对外部实现控制和信息交换的必经之路。I/O 端口有串行和并行之分,串行 I/O 端口一次只能传送一位二进制信息,并行 I/O 端口一次能传送一组二进制信息。

1. 串行 I/O 端口

8051 单片机有一个全双工的可编程串行 I/O 端口。这个串行 I/O 端口既可以在程序控制下将 CPU 的 8 位并行数据变成串行数据一位一位地从发送数据线 TXD 发送出去,也可以把串行接收到的数据变成 8 位并行数据送给 CPU,而且这种串行发送和串行接收既可以单独进行,也可以同时进行。

8051 单片机串行发送和串行接收利用了 P3 口的第二功能,即利用 P3.1 引脚作为串行数据的发送线 TXD,P3.0 引脚作为串行数据的接收线 RXD,具体内容参见表 1-1。串行 I/O 端口的电路结构还包括串行口控制器 SCON、电源及波特率选择寄存器 PCON 和串行数据缓冲器 SBUF 等,它们都属于特殊功能寄存器。其中 PCON 和 SCON 用于设置串行口工作方式和确定数据的发送和接收波特率,SBUF 实际上由两个 8 位寄存器组成,一个用于存放待发送的数据,另一个用于存放接收到的数据,起着数据的缓冲作用,这些将在以后的项目中加以详细介绍。

2. 并行 I/O 端口

MCS-51 系列单片机设有 4 个 8 位双向 I/O 端口(P0 口、P1 口、P2 口、P3 口),每一个 I/O 端口都能独立地用做输入或输出。P0 口为三态双向口,能驱动 8 个 LSTTL 负载;P1 口、P2 口、P3 口为准双向口(在用作输入线时,端口锁存器必须先写入“1”,故称为准双向口),负载能力为 4 个 LSTTL 电路。

1) P0 口功能

P0 口位结构如图 1-3 所示,包括一个输出锁存器,两个三态缓冲器,一个输出驱动电路和一个输出控制端。输出驱动电路由一对场效应管组成,其工作状态受输出端控制。输出控制端由一个与门、一个反相器和一个转换开关 MUX 组成。对 8051 单片机来讲,P0 口既可作为 I/O 端口,又可作为地址/数据总线使用。

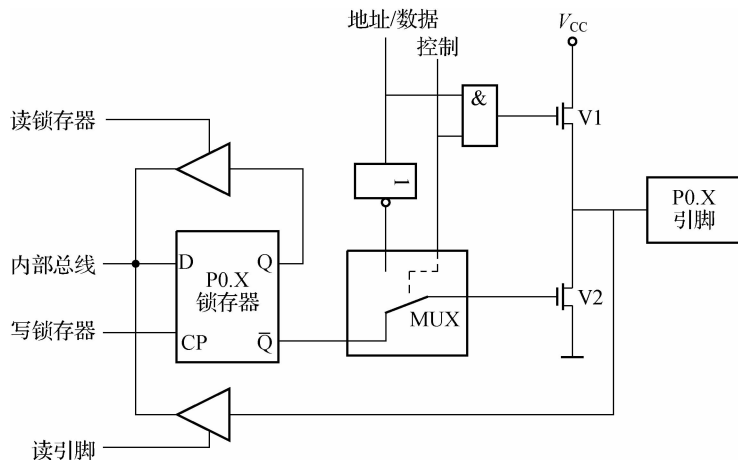


图 1-3 P0 口位结构



(1) P0 口作地址/数据总线使用。若从 P0 口输出地址或数据信息,此时控制端应为高电平,转换开关 MUX 将反相器输出端与输出级场效应管 V2 接通,同时与门开锁,内部总线上的地址或数据信号通过与门去驱动 V1 管,又通过反相器去驱动 V2 管,这时内部总线上的地址或数据信号就传送到 P0 口的引脚上。工作时低 8 位地址与数据线分时使用 P0 口。低 8 位地址由 ALE 信号的负跳变使它锁存到外部地址锁存器中,而高 8 位地址由 P2 口输出。

(2) P0 口作通用 I/O 端口使用。对于有内部 ROM 的单片机,P0 口也可以作通用 I/O 端口,此时控制端为低电平,转换开关把输出级与锁存器的 Q 端接通,同时因与门输出为低电平,V1 管处于截止状态,输出级为漏极开路电路,在驱动 NMOS 电路时应外接上拉电阻。P0 口作输入口用时,应先将锁存器写“1”,这时输出级两个场效应管均截止,可作高阻抗输入,通过三态输入缓冲器读取引脚信号,从而完成输入操作。

(3) P0 口上的“读—修改—写”功能。图 1-3 上面一个三态缓冲器是为了读取锁存器 Q 端的数据,Q 端与引脚的数据是一致的,结构上这样安排是为了满足“读—修改—写”指令的需要。这类指令的特点是:先读锁存器,随之可能对读入的数据进行修改再写入端口。这类指令同样适合 P1 口、P2 口和 P3 口,其操作是:先将端口的全部 8 位数读入,再通过指令修改某些位,然后将新的数据写回到锁存器中。

2) P1 口功能

P1 口是一个有内部上拉电阻的准双向口,位结构如图 1-4 所示。

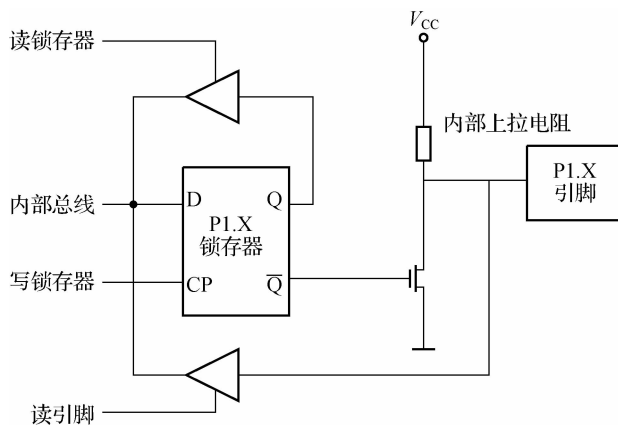


图 1-4 P1 口位结构

(1) P1 口作通用 I/O 端口使用。P1 口的每一只引脚都能独立用作输入线或输出线。作输出时,将“0”写入锁存器,场效应管导通,输出线为低电平,即输出为“0”。作输入时,必须先将“1”写入锁存器,使场效应管截止。该引脚由内部上拉电阻提拉成高电平,同时也能被外部输入源拉成低电平,即当外部输入“1”时该引脚为高电平,而输入“0”时,该引脚为低电平。P1 口作输入时,可被任何 TTL 电路和 MOS 电路驱动,由于具有内部上拉电阻,也可以直接被集电极开路和漏极开路电路驱动,不必外加上拉电阻。

(2) P1 口其他功能。P1 口在 EPROM 编程和验证程序时,输入低 8 位地址。在 8032/8052 单片机中,P1.0 引脚和 P1.1 引脚是多功能的:P1.0 引脚可作定时/计数器 2 的外部计数触发输入端,P1.1 引脚可作定时/计数器 2 的外部控制输入端。

3) P2 口功能

P2 口位结构如图 1-5 所示,引脚上拉电阻同 P1 口。在结构上,P2 口比 P1 口多一个输出控制部分。

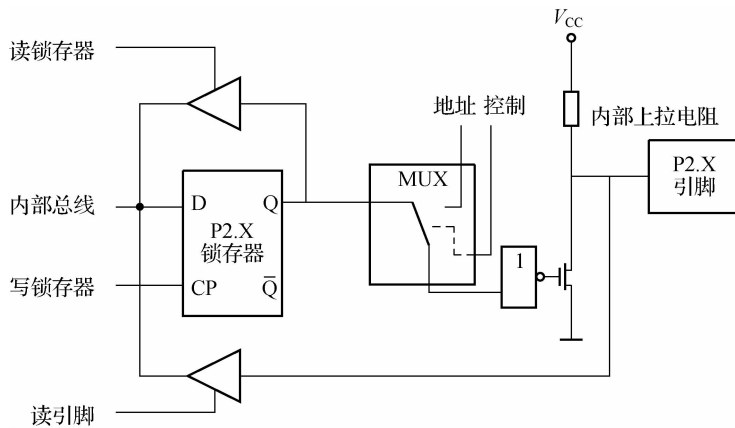


图 1-5 P2 口位结构

(1)P2 口作通用 I/O 端口使用。当 P2 口作通用 I/O 端口使用时,是一个准双向口,此时转换开关 MUX 倒向左边,输出级与锁存器接通,引脚可接 I/O 设备,其 I/O 操作与 P1 口完全相同。

(2)P2 口作地址总线使用。当系统中接有外部存储器时,P2 口用于输出高 8 位地址 A15~A8。这时在 CPU 的控制下,转换开关 MUX 倒向右边,接通内部地址总线。P2 口的引脚状态取决于片内输出的地址信息,这些地址信息来源于 PCH、DPH 等。在外接程序存储器的系统中,由于访问外部存储器的操作连续不断,P2 口不断送出地址高 8 位。在由 8031 单片机构成的系统中,P2 口一般只作地址总线使用,不作 I/O 端口直接连外部设备。

4) P3 口功能

P3 口是一个多用途的端口,也是一个准双向口,作为第一功能使用时,其功能同 P1 口。P3 口位结构如图 1-6 所示。

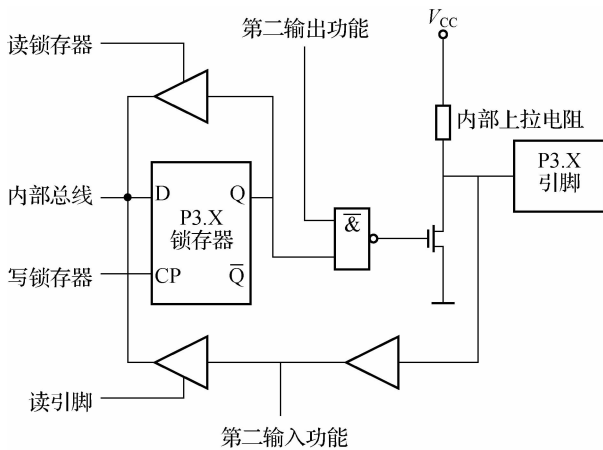


图 1-6 P3 口位结构



作为第二功能使用时，P3 口每一位功能定义见表 1-1。P3 口的第二功能实际上就是系统具有控制功能的控制线，此时相应 P3 口的锁存器必须为“1”状态。与非门的输出由第二功能输出线的状态确定，从而 P3 口的状态取决于第二功能输出线的电平。在 P3 口的引脚信号输入通道中有两个三态缓冲器，第二功能的输入信号取自第一个缓冲器的输出端，第二个缓冲器仍是第一功能的读引脚信号缓冲器。

每个 I/O 端口内部都有一个 8 位数据输出锁存器和一个 8 位数据输入缓冲器，四个数据输出锁存器与端口号 P0、P1、P2 和 P3 同名，皆为特殊功能寄存器。因此，CPU 数据从并行 I/O 端口输出时可以得到锁存，数据输入时可以得到缓冲。

四个并行 I/O 端口作为通用 I/O 端口使用时，共有写端口、读端口和读引脚三种操作方式。写端口实际上就是输出数据，是将累加器 A 或其他寄存器中的数据传送到端口锁存器中，然后自动从端口引脚线上输出。读端口不是真正的从外部输入数据，而是将端口锁存器中的输出数据读取后存到 CPU 的累加器。读引脚才是真正的输入外部数据的操作，是从端口引脚线上读取外部的输入数据。I/O 端口的上述三种操作是通过指令或程序来实现的，这些将在以后的项目中详细介绍。

项目实践

1. 参观元器件买卖市场

熟悉当地元器件买卖市场，自己购买各种集成块，查看并了解各种元器件。

2. 熟悉各种元器件网站，会查各种资料

搜索三家以上元器件销售网站，了解多家集成块生产厂家，最低收集 10 家，了解各生产厂家的产品质量和价格，以及每种器件的技术参数、制作工艺、制造设备等知识。下载各种资料，增加自己的相关知识。与此同时，在计算机中安装各种仿真软件、课件和动画，以供学习钻研，多做多练，培养学习能力和动手能力。

3. 熟悉元器件

认真查看购买回来的单片机集成块，比较不同集成块之间的主要区别。用万用表测试每一个脚对地(GND)的电阻，记录下每个阻值，便于以后检修，因为集成块损坏时这些阻值会发生改变。

拓展知识

单片微型计算机简称为单片机。它是微型计算机发展中的一个重要分支，它以其独特的结构和性能，越来越广泛地应用到工业、农业、国防、网络、通信以及人们的日常工作、生活领域中。

单片机在一块芯片上集成了中央处理部件(CPU)、存储器(RAM、ROM)、定时/计数器和各种 I/O 端口(并行 I/O 端口、串行 I/O 端口和 A/D 转换器)等部件。由于单片机通常是实时控制应用而设计和制造的，所以也将其称为微控制器(MCU)。

1. 单片机的基本知识

每一种单片机的设计都包括以下几个方面。



- (1) 指令及与指令对应的电路(芯片)和代码。
- (2) 固化程序的编程器。
- (3) 硬件仿真器。

设计完成后,由有关生产厂家生产出产品(芯片、编程器、仿真器),再由开发人员开发出市场所需要的应用产品。在这些开发的芯片中,有些芯片是公开使用的,在市场上能买到的芯片就属这种类型;有些芯片是不公开的,如军工产品和各大公司开发的专用产品。公开使用的芯片又分为两类:一类不能加密,一类可加密。

单片机种类很多,但不管哪种单片机,厂家都要配套提供编程器(固化程序用)、硬件仿真器(调试程序用)、指令系统和芯片使用说明书,没有这些很难进行二次开发,除非能破解芯片。因而,对产品开发人员来说,所要做的工作就是按厂家提供的方法使用芯片,并且能够按产品功能要求设计电路、编写程序、做成产品。对产品维修使用人员来说,知道芯片的使用方法和产品电路的工作原理,会维修使用即可。

2. 单片机的发展概况

单片机自问世以来,性能不断提高和完善,其资源不仅能满足很多应用场合的需要,而且具有集成度高、功能强、速度快、体积小、功耗低、使用方便、性能可靠、价格低廉等特点,因此,在工业控制、智能仪器仪表、数据采集和处理、通信系统、网络系统、汽车工业、国防工业、高级计算器具、家用电器等领域的应用日益广泛,并且正在逐步取代现有的多片微机应用系统。单片机的潜力越来越被人们所重视,特别是用 CMOS 工艺制成的各种单片机,由于功耗低、使用的温度范围大、抗干扰能力强、能满足一些特殊要求的应用场合等特点,不仅扩大了单片机的应用范围,也进一步促进了单片机技术的发展。

自 1976 年 9 月 Intel 公司推出 MCS-48 系列单片机以来,单片机就受到了广大用户的欢迎。因此,有关公司都争相推出各自的单片机,如 GI 公司推出的 PIC1650 系列单片机, Rockwell 公司推出的与 6502 微处理器兼容的 R6500 系列单片机,它们都是 8 位机,片内有 8 位中央处理器(CPU)、并行 I/O 端口、8 位定时/计数器和容量有限的存储器(RAM、ROM)以及简单的中断功能。

1980 年,Intel 公司在 MCS-48 系列单片机的基础上又推出了高性能的 MCS-51 系列单片机。这类单片机均带有串行 I/O 端口,定时/计数器为 16 位,片内存储(RAM、ROM)容量相应增大,并有优先级中断处理功能。MCS-51 系列单片机扩大了功能和寻址范围,是当时单片机应用的主流产品。

1982 年,Mostek 公司和 Intel 公司先后又推出了性能更高的 16 位单片机 MK68200 和 MCS-96 系列,NS 公司和 NEC 公司也分别在原有 8 位单片机的基础上推出了 16 位单片机 HPC16040 和 μ PD783 \times \times 系列。1987 年 Intel 公司又推出了性能比 8096 单片机高两倍的 CMOS 型单片机 80C196,1988 年推出带 EPROM 的 87C196 单片机。由于 16 位单片机的推出时间较迟、价格昂贵、开发设备有限等多种原因,至今还未得到广泛应用。而 8 位单片机已能满足大部分应用的需要,因此,在推出 16 位单片机的同时,高性能的新型 8 位单片机也不断问世。

目前,国际市场的 8 位、16 位单片机系列已有很多。在我国使用较多的系列是 Intel 公司的产品,其中以 MCS-51 系列单片机应用尤为广泛,其价格也越来越低,性能也越来越好。



拓展
认识单片机



单片机技术正以惊人的速度向前发展,就市场上已出现的单片机而言,其技术革新与进步主要表现在以下几个方面。

1) CPU 的发展

增加 CPU 的字长或提高时钟频率均可提高 CPU 的数据处理能力和运算速度。CPU 的字长目前有 8 位、16 位和 32 位,时钟频率高达 20 MHz 的单片机也已出现,还有的 8 位单片机其算术逻辑运算部件(ALU)是 16 位,内部采用 16 位数据总线。另外,单片机内部采用双 CPU 结构能大大提高处理能力,由于片内有两个 CPU 能同时工作,能更好地处理外围设备的中断请求,克服了单 CPU 在多重高速中断响应时的失效问题。同时,由于双 CPU 可以共享存储器和 I/O 端口的资源,因此,还可更好地解决信息通信问题,如 Intel 公司的 8044 单片机,它的内部实际上是由 8051 单片机和 SIU 通信处理机组成的,由 SIU 来管理 SDLC 的通信,这样既加快了通信处理的速度,也减轻了 8051 单片机的处理负担。

2) 片内存储器的发展

(1) 扩大存储容量。早期单片机的片内存储器,一般 RAM 为 64~128 B,ROM 为 1~2 KB,寻址范围为 4 KB。新型单片机片内 RAM 为 256 B,ROM 多达 16 KB。新型单片机的寻址范围可扩大到 64 KB,甚至 128 KB(其中随机存储器 RAM 容量为 64 KB,只读存储器 ROM 容量为 64 KB)。内部 ROM 分可擦除和一次性可编程(OTP)两种,前者价高,技术开发时使用,后者价低,开发成功后,一次性固化在产品上使用,须注意的是,一次性固化在产品上使用的必须是成熟产品,否则会造成经济损失。

(2) 片内 EPROM 开始 E²PROM 化。早期单片机内 ROM 有的采用可擦除式的只读存储器——EPROM,然而 EPROM 必须要高压编程,紫外线擦除,给使用带来不便。近年来推出的电擦除可编程只读存储器——E²PROM,可在正常工作电压下进行读写,并能在断电的情况下保持信息不丢失,因此,有些厂家已开始用 E²PROM 替代原来的片内 EPROM。由于写入 E²PROM 的数据能永久保存,因此,有些厂家已开始将 E²PROM 用作片内 ROM,甚至用作片内通用寄存器,这样可省去备用电池。

(3) 闪速存储器。随着 CMOS 工艺的改进和提高,闪速存储器在不断发展和完善,应用越来越广,容量越来越大,价格越来越低。闪存技术在各个领域得到应用,如 ATMEL 公司将闪存技术应用到单片机中,生产出了带闪速存储器的 AT89 系列单片机,对小系统,外部可以不用扩展存储芯片,从而使得只用单片机就能构成一个完整的控制系统。PIC 系列单片机也有带闪存的存储芯片。

(4) 串行存储器。I²C 总线的快速发展,使得串行数据存储器在容量和存储速度上有了很大的提高,由于它体积小、引脚少、价格低,因此得到了广泛的应用。

(5) 片内程序的保密措施。为了使片内 EPROM(E²PROM)内容不被复制,一些厂家对片内 EPROM(E²PROM)采用加锁技术。如 Intel 公司的 8X252 单片机,加锁后的 EPROM(E²PROM)中的程序只能供片内 CPU 读取,不能从片外读取,否则必须先开锁,开锁时,CPU 先自动擦除 EPROM(E²PROM)中的信息,从而达到程序保密的目的。

3) 片内 I/O 端口功能

最初的单片机,片内只有并行 I/O 端口、定时/计数器,它们的功能较弱,实际应用中往往需要通过特殊的接口扩展功能,从而也增加了应用系统结构的复杂性。

近年来,新型单片机内的接口,从类型和数量上都有很大的发展,这不仅大大提高了单



片机的功能,而且使系统的总体结构也大大简化了。如有些单片机的并行 I/O 端口能直接输出大电流和高电压,可直接用于驱动荧光显示管(VFD)、液晶显示器(LCD)和数码显示管(LED)等,应用系统中就不再需要外部驱动电路。又如有些单片机片内含有 A/D 转换器,在一些实时控制系统中可省掉外部 A/D 转换器。

目前,在单片机中已出现的各类新型接口有数十种,如 A/D 转换器、D/A 转换器、DMA 控制器、CRT 控制器、LCD 驱动器、LED 驱动器、VFD 驱动器、正弦波发生器、声音发生器、字符发生器、波特率发生器、锁相环、频率合成器、脉宽调制器等。虽然一个单片机内只含若干种接口,但其功能比初期的单片机强得多,因此,可用它作为高速主机的通用外设接口。

目前,单片机种类繁多,功能多样,将外围电路尽量集中在芯片内,使其成为名副其实的单片机,这也成为一种发展趋势。

4) 单片机在工艺上的提高

单片机的制造工艺直接影响其性能。早期的单片机采用 PMOS 工艺,随后逐渐采用 NMOS、HMOST 和 CMOS 工艺。目前,8 位单片机中有一半产品已 CMOS 化,也已开始推出 16 位单片机 CMOS 型产品。采用 CMOS 工艺的单片机,其工作电源范围较宽,功耗大小与电源电压成正比,所以降低电源电压即可降低功耗;但是降低电压会减慢指令执行速度,即降低单片机的运算速度,故一般希望在一定速度的前提下尽量降低工作电压,减小功耗。

随着新型单片机片内接口电路的增多,外引脚也增多,为减少外引脚线,目前主要采用两种方式:一种是采用新颖的通信总线以减少外引线,一种是改进外封装,如采用扁平引脚封装 FP(flat package)、方形引脚封装 QIP(quad in line package)等。

5) 片内固化应用软件和系统软件

将一些应用软件和系统软件固化于片内 ROM 中,可以简化应用程序的编制工作,为开发和应用提供方便。如 RUP1-44 系列单片机,把通信控制软件固化在片内,使通信程序大大简化。又如 Intel 公司在有的 MCS-51 系列单片机内固化了 PL/M-51 语言,在 8052 BH 单片机中固化了 BASIC 解释程序,不仅可用汇编语言,还可用 BASIC 语言编程,其 BASIC 语言系统比基本 BASIC 有所扩充,增加了许多适合控制用的语句、命令、运算符等,而且还允许 BASIC 语言和汇编语言互相调用,需要快速控制时,可用汇编语言,如采样、A/D 转换等,在作复杂的数据运算时,可用汇编语言来调用 BASIC 中现成的运算子程序,既能满足速度方面的要求,又能简化编程。

单片机的技术还在不断发展,新型单片机还将不断涌现,当前单片机的产量占整个微机(包括一般的微处理器)产量的 80% 以上。在我国,低档 8 位单片机(如 8048 单片机)于 20 世纪 80 年代初就开始应用,目前已转向高档 8 位单片机(如 MCS-51 系列单片机、Z8 系列单片机等)的应用,也有不少企业已转向 16 位单片机的开发和应用。

3. 单片机的应用

单片机在一块芯片上集成了一台微型计算机所需的 CPU、存储器、I/O 部件和时钟电路等,因此它具有体积小、使用灵活、成本低、易于产品化、抗干扰能力强、可在各种恶劣环境下可靠地工作等特点,特别是它应用面广、控制能力强,使它在工业控制、智能仪表、外设控制、家用电器、机器人、军事装置等方面得到了广泛的应用。

单片机主要用于以下几方面。



1) 控制系统中的应用

控制系统的工作环境恶劣,各种干扰也强,特别是工业控制系统,而且往往要求实时控制,故要求控制系统工作稳定、可靠、抗干扰能力强。单片机最适合用于控制领域,如炉子恒温控制、电镀生产线自动控制等。

2) 智能仪表中的应用

用单片机制作测量、控制仪表,能使仪表向数字化、智能化、多功能化和柔性化发展,并使监测、处理、控制等功能一体化,使仪表重量大大减轻,便于携带和使用,同时降低了成本,提高了性价比,如数字式RLC测量仪、智能转速表、计时器等。

3) 智能产品

单片机与传统的机械产品结合,使传统机械产品结构简化、控制智能化,构成新型的机、电、仪一体化产品,如数控车床、智能电动玩具、各种家用电器和通信设备等。

4) 在智能计算机外设中的应用

在计算机应用系统中,除通用外部设备(键盘、显示器、打印机)外,还有许多用于外部通信、数据采集、多路分配管理、驱动控制等的接口。如果这些外部设备和接口全部由主机管理,势必造成主机负担过重、运行速度降低,并且不能提高对各种接口的管理水平。如果采用单片机专门对接口进行控制和管理,则主机和单片机就能并行工作,这不仅能大大提高系统的运算速度,而且单片机还可对接口信息进行预处理,以减少主机和接口间的通信密度,提高接口控制管理的水平,如绘图仪控制器,磁带机、打印机的控制器等。

目前,国外的单片机应用已相当普及,国内虽然从1980年才开始着手开发应用,但至今已拥有数十家专门生产单片机开发系统的工厂或公司,越来越多的科技工作者投身到单片机的开发和应用中,并且在程序控制、智能仪表等方面涌现出大量的科技成果,可以预见,单片机在我国必将有更为广阔的发展前景。



复习思考题

1. 什么是单片机?它与一般微型计算机在结构上有何区别?
2. 单片机的发展大致可分为几个阶段?各阶段的单片机功能特点是什么?



项目二

单片机开发工具

知识目标

- (1) 熟悉 Keil μ Vision3 软件各菜单的功能；
- (2) 掌握 Proteus 软件的基本操作方法；
- (3) 了解 QTH 系列单片机仿真器开发应用程序的过程。

技能目标

- (1) 灵活运用软件仿真开发软件；
- (2) 灵活运用硬件仿真开发软件；
- (3) 灵活运用在线硬件仿真开发系统。

学思园地

工欲善其事,必先利其器

孔子曰“工欲善其事,必先利其器”,其为人们揭示了一个简单而重要的道理,那就是不管做什么事,都要先做好充分的准备工作。“磨刀不误砍柴工”“宜未雨而绸缪,毋临渴而掘井”,说的都是这方面的道理。不管做什么事,只有事先准备好了称手的工具,做好了充分的准备,才能高效地完成工作,取得较好的成绩。

在芯片领域,没有工具将一事无成,没有集成开发环境(integrated development environment, IDE)就不能进行代码编写、分析、编译、调试等工作,没有程序烧录器(下载器),编写成功后的程序就无法写入芯片内部。所以在进行芯片学习之前,先要掌握开发工具的使用方法。

项目描述

单片机是一门综合性学科,需要模拟电子技术和数字电子技术作为先导知识,还要结合计算机使用基础知识,同时,单片机又是一门实践性很强的专业技术,使用的设备较多。对于初学者来说,没有进行理论学习、软件仿真、硬件仿真、程序固化、实物制作的全过程,很难入门,所以在学习过程中首先要掌握单片机的开发工具的使用方法。



课件
单片机开发
工具



相关知识

一、软件仿真

Keil C51 软件是 MCS-51 系列单片机应用开发软件,其模拟调试功能很强,基本上包括了在线仿真器的单步、跟踪、检查和修改功能,并且还能模拟产生各种中断和 I/O 端口应答过程。Keil C51 软件集可视化编辑、编译、调试、仿真于一体,支持汇编、PLM 和 C 语言的混合编程,功能强大、界面友好、易学易用。下面主要介绍 Keil μ Vision3 软件的使用方法。

1. Keil μ Vision3 软件概述

Keil μ Vision3 软件是流行的单片机开发环境之一,其安装的方法同一般软件的安装。安装完成后将在 Windows 桌面生成一个 Keil μ Vision3 的图标。单击“开始”→“程序”→Keil μ Vision3 即可运行该软件,也可双击 Keil μ Vision3 的图标运行该软件。Keil μ Vision3 的主窗口如图 2-1 所示,有菜单栏、工具栏、源代码窗口、项目窗口和输出窗口。



视频

Keil C51 软件的使用

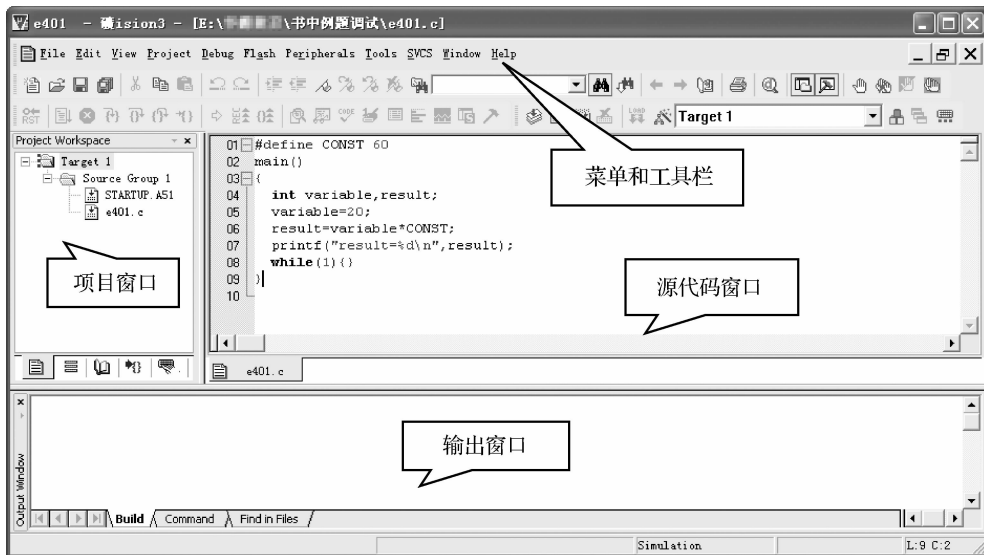


图 2-1 Keil μ Vision3 主窗口

1) Keil μ Vision3 软件的主菜单

Keil μ Vision3 软件的主菜单提供各种操作菜单,如编辑操作、项目维护、开发工具选项设置、调试程序、窗口选择和处理以及在线帮助等,下面分别介绍。

(1) File 菜单和命令见表 2-1。

表 2-1 File 菜单和命令

菜 单	工具条	快捷键	描 述
New		Ctrl+N	创建新文件
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

续表

菜 单	工具条	快捷 键	描 述
Save all			保存所有文件
Save as			另外取名保存
Device Database			维护器件库
Print Setup			打印页面属性设置
Print		Ctrl+P	打印当前文件
Print Preview			打印预览
Exit			退出 Keil μ Vision3

(2) Edit 菜单和命令见表 2-2。

表 2-2 Edit 菜单和命令

菜 单	工具条	快捷 键	描 述
Undo		Ctrl+Z	取消上次操作
Redo		Ctrl+Shift+Z	重复上次操作
Cut		Ctrl+X	剪切所选文本
Copy		Ctrl+C	复制所选文本
Paste		Ctrl+V	粘贴
Indent Selected Text			右移一个制表位的距离
Unindent Selected Text			左移一个制表位的距离
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			清除当前文件的所有标签
Find		Ctrl+F	在当前文件中查找文本
Replace		Ctrl+H	替换特定的字符
Find in Files		Ctrl+ Shift+F	在多个文件中查找
Incremental Find		Ctrl+I	增量式查找
Goto Matching Brace			转到对应的括号

(3) View 菜单和命令见表 2-3。

表 2-3 View 菜单和命令

菜 单	工具条	描 述
Status Bar		显示/隐藏状态
File Toolbar		显示/隐藏文件工具栏
Build Toolbar		显示/隐藏编译工具栏
Debug Toolbar		显示/隐藏调试工具栏



续表

菜单	工具条	描述
Project Window		显示/隐藏项目窗口
Output Window		显示/隐藏输出窗口
Source Browser		打开资源浏览器
Disassembly Window		显示/隐藏反汇编窗口
Watch & Call Stack Window		显示/隐藏观察和堆栈窗口
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
Toolbox		显示/隐藏自定义工具
Periodic Window Update		程序运行时刷新调试窗口
Workbook Mode		显示/隐藏窗口框架模式
Include Dependencies		包含
Options		设置颜色、字体、快捷键等

(4) Project 菜单和命令见表 2-4。

表 2-4 Project 菜单和命令

菜单	工具条	快捷键	描述
New Project			创建新项目
Import μ Vision1 Project			转化 μ Vision1 的项目
Open Project			打开一个已经存在的项目
Close Project			关闭当前的项目
Target Environment			定义工具、包含文件和库的路径
Targets Groups Files			维护一个项目的文件组 and 文件
Select Device for Target			选择对象的 CPU
Remove			从项目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	编译修改过的文件并生成应用
Rebuild all Target Files			重新编译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

(5) Debug 菜单和命令见表 2-5。

表 2-5 Debug 菜单和命令

菜 单	工具条	快捷 键	描 述
Start/Stop Debug Session		Ctrl+F5	开始/停止调试模式
Go		F5	运行程序,直到遇到一个中断
Step		F11	单步执行程序,进入子程序运行
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
Stop Running		Esc	停止程序运行
Breakpoints			打开断点对话框
Insert/Remove Breakpoint			设置/取消当前行的断点
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
Enable/Disable Trace Recording			使能/禁止程序运行轨迹
View Trace Records			显示程序运行过的指令
Memory Map			打开存储器空间配置对话框
Performance Analyzer			打开设置性能分析的窗口
Inline Assembly			对某行重新汇编,修改汇编代码
Function Editor			编辑、调试函数和调试配置文件

(6)Peripherals 菜单和命令见表 2-6。

表 2-6 Peripherals 菜单和命令

菜 单	工具条	描 述
Reset CPU		复位 CPU
I/O-Ports		打开片上外围器件的设置对话框
Interrupt		打开中断观察窗口
Serial		打开串行口观察窗口
Timer		打开定时/计数器观察窗口
A/D Converter		A/D 转换器
D/A Converter		D/A 转换器
I ² C Controller		I ² C 控制器
CAN Controller		CAN 控制器
Watchdog		“看门狗”



(7)Tools 菜单和命令见表 2-7。

表 2-7 Tools 菜单和命令

菜 单	描 述
Setup PC-Lint	配置 Gimpel Software 的 PC-Lint 程序
Lint	用 PC-Lint 处理当前编辑的文件
Lint all C Source Files	用 PC-Lint 处理项目中的 C 源代码文件
Setup Easy-Case	配置 Siemens 的 Easy-Case 程序
Start/Stop Easy-Case	运行/停止 Siemens 的 Easy-Case 程序
Show File (Line)	用 Easy-Case 处理当前编辑的文件
Customize Tools Menu	添加用户程序到工具菜单中

(8)SVCS 菜单见表 2-8。

表 2-8 SVCS 菜单

菜 单	描 述
Configure Version Control	配置软件版本控制系统的命令

(9)Window 菜单见表 2-9。

表 2-9 Window 菜单

菜 单	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
Close All	关闭所有窗口

(10)Help 菜单见表 2-10。

表 2-10 Help 菜单

菜 单	描 述
Help Topics	打开在线帮助
About μ Vision	显示版本信息和许可证信息

2)Keil μ Vision3 中的调试器

Keil μ Vision3 中集成了一种新型调试器——Debug,它可以进行纯软件模拟仿真和硬件目标板在线仿真,使用之前应该先对其进行适当配置。单击 Project 菜单,选择 Options for Target 选项,弹出 Options for Target 对话框,切换到如图 2-2 所示 Debug 选项卡。

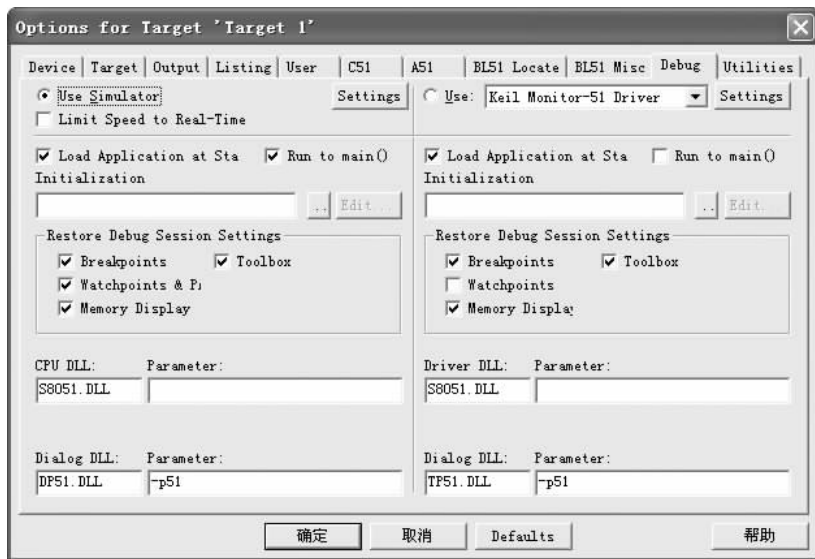


图 2-2 Debug 选项卡

在该选项卡中选中单选按钮 Use Simulator, 可采用软件模拟方式进行仿真。这种模式下, 可以在没有任何单片机硬件的情况下仅用一台计算机实现对单片机应用程序的仿真模拟。选中 Use 下拉列表框前的单选按钮, 单击下拉按钮可选择不同器件对其仿真。例如, 选择 Intel 8052 可仿真 8052 单片机内部定时器 T2; 选择 Philips 80C552 可仿真 80C552 单片机内部 A/D 转换器的功能; 选择 Monitor-51 Driver 可采用硬件目标板方式进行仿真。选择硬件目标板方式进行仿真时, 要购买硬件仿真器, 还要安装 Keil 公司提供的专用监控程序 Monitor-51, 有了这两项后可实现对硬件目标系统的在线仿真。单击 Settings 按钮, 弹出如图 2-3 所示对话框。单击 Port 右侧下拉按钮选择计算机的串口名, 单击 Baudrate 右侧下拉按钮选择需要的通信波特率。对话框中间有四个复选框, 是设定调试过程中数据的缓存选项, 选中时可以加快数据在计算机屏幕上的显示速度, 若希望直接观察单片机内部定时/计数器、I/O 端口各引脚电平状况以及外部扩展端口实际数据的变化, 则不要选中这些复选框。选中 Serial Interrupt 复选框可以在运行程序过程中通过 Debug 调试器的 Stop 按钮或计算机键盘的 Esc 键停止程序的运行, 但这时程序中不能使用单片机的串行口, 同时不能禁止特殊功能寄存器 IE 中的 EA 位。

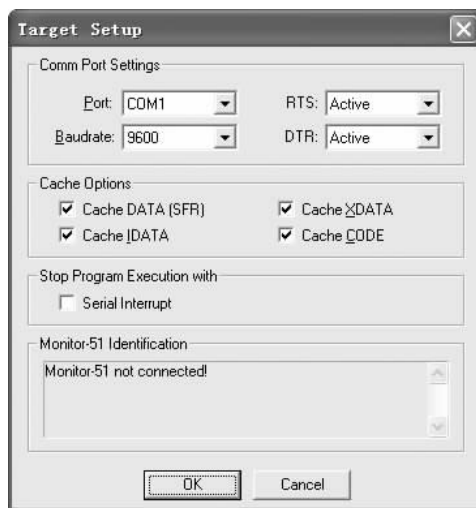


图 2-3 Target Setup 对话框

选中 Debug 选项卡中的 Load Application at Startup 复选框, 在启动 Debug 时将自动装入程序, 选中 Ran to main() 复选框, 程序将从复位入口运行到 main() 函数处, 通常这两个复选框都需要选中以便于调试。在 Initialization 文本框内可以输入一个带路径的初始化文件



名,该文件的内容为 Debug 的各种调试命令。可以在启动调试时一次执行。单击 Edit 按钮可以在编辑窗口打开初始化文件进行编辑。四个复选框 Breakpoints、Watchpoints、Memory Display 和 Toolbox 分别用于在启动 Debug 时自动恢复上次调试过程中所设置的断点、观察点,性能分析器,存储器及工具箱的显示状态,如果希望启动 Debug 时能够使用在编辑源程序文件时就设置的断点,应该选中这些复选框。CPU DLL 文本框、Driver DLL 文本框、Dialog DLL 文本框和 Parameter 文本框是根据项目配置时从器件库中所选择的单片机 CPU 器件,由 Keil μ Vision3 自动设置的内部驱动程序及参数,一般不要轻易改动。

Debug 选项配置完成之后,打开或建立新项目并编译通过后,选择 Debug 菜单中的 Start/Stop Debug Session 选项,即可启动 Debug。启动 Debug 后,项目窗口分配如图 2-4 所示:寄存器窗口用于显示程序调试过程中单片机内部寄存器状态的变化情况;主调试窗口用于显示源程序,窗口左边的箭头指向当前程序语句,每执行一条语句箭头会自动向下移动,便于观察程序当前执行点,如果项目中包含有多个程序文件,执行过程中将自动切换到不同文件显示;命令窗口用于输入各种调试命令;存储器窗口用于显示程序调试过程中单片机的存储器状态;观察窗口用于显示局部变量和观察点的状态。此外,在主调试窗口还可以显示反汇编窗口、串行窗口以及性能分析窗口,通过选择 View 菜单中的相应选项,或单击相应工具按钮,可以切换窗口。

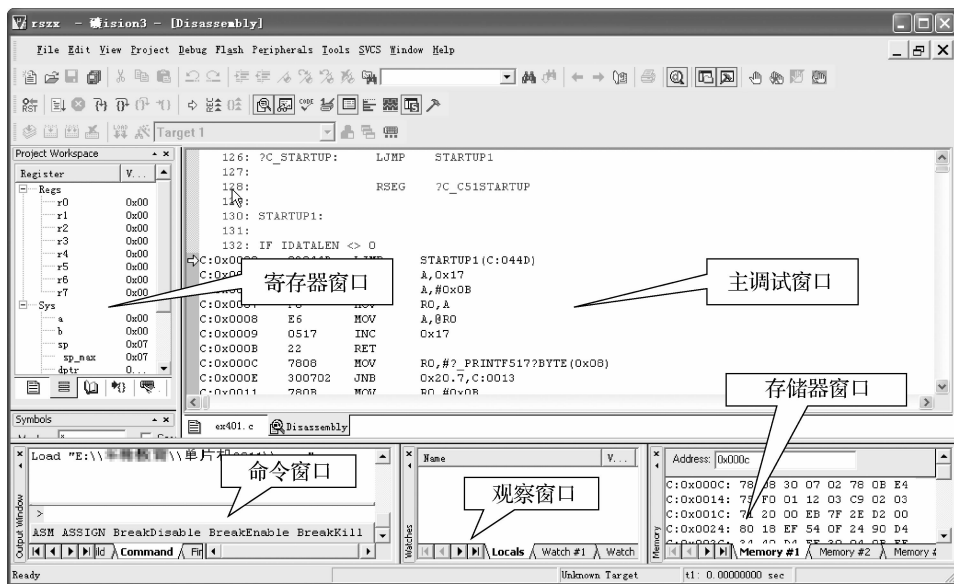


图 2-4 Keil μ Vision3 项目窗口分配

(1)命令窗口。调试状态下,单击 View 菜单,选择 Output Window 选项,弹出输出窗口,单击 Command 标签,可出现如图 2-5 所示命令窗口。

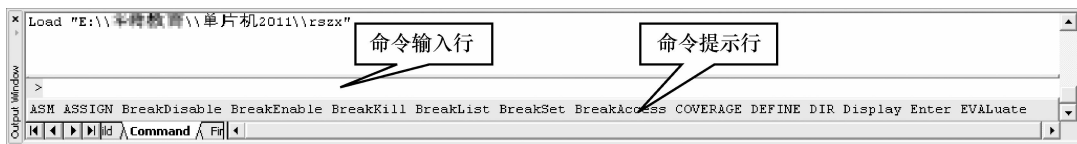


图 2-5 调试状态下命令窗口

窗口中有一个显示提示符“>”的命令输入行,其中可以输入各种命令字,如装入程序的目标文件,运行、设置观察点或断点等。在命令输入行内输入命令字并按 Enter 键,该命令将立即被执行,执行结果显示在输出窗口中,若命令输入错误,窗口中将显示出错信息。Keil μ Vision3 内部集成了一个方便实用的命令语法产生器,在输入命令时自动将所有可选的命令字、命令需要的参数等显示在窗口下边的命令提示行内。按 Tab 键可滚动显示提示行上的内容,输入提示行上某个命令字中的大写字母可直接输入该命令。在命令输入时提示行上的显示内容会自动减少,只剩下一个命令字时,输入空格即可输入该命令字,输入命令字后提示行上还将显示该命令所要求的各种参数,以避免出错。Keil μ Vision3 允许在输入命令字时带有 C 语言格式的注释。命令行可通过按键来改变命令字的输入,各按键功能见表 2-11。

表 2-11 按键功能

按 键	功 能
Enter	执行所输入的命令
Backspace	删除光标前面的一个字符
Del	删除光标处的一个字符
Esc	中止命令输入并开始新的命令输入行
Home	置光标于输入行首
End	置光标于输入行尾
Tab	滚动显示帮助行上的信息
←	光标左移一格
→	光标右移一格
↑	光标上移一行
↓	光标下移一行

(2)反汇编窗口。单击 View 菜单,选择 Disassembly Window 选项,弹出如图 2-6 所示的反汇编窗口,窗口中显示已经装入软件中的程序的汇编指令、反汇编代码及其地址。

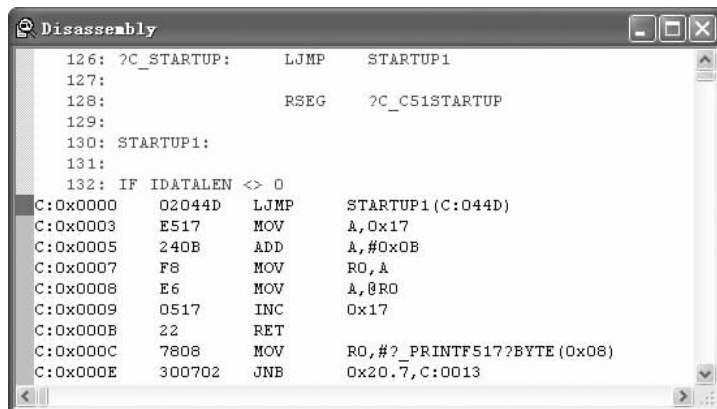


图 2-6 反汇编窗口



当采用单步或断点方式运行程序时,反汇编窗口的显示内容会随指令的执行而滚动。在反汇编窗口中右击可弹出如图 2-7 所示的快捷菜单。

①快捷菜单第一栏。该菜单第一栏中共有三个选项,用于选择窗口内反汇编内容的显示方式。选择 Mixed Mode 选项,系统采用高级语言与汇编语言混合方式显示;选择 Assembly Mode 选项,系统采用汇编语言方式显示;选择 Inline Assembly 选项,系统用于程序调试中的“在线汇编”。使用“在线汇编”时,先将光标定位在反汇编窗口中的希望行上,再选择该选项,弹出如图 2-8 所示的对话框,其中 Current Instruction 文本框显示的是指定行当前的汇编指令,Current Assembly Address 文本框显示的是当前指令行的地址,在 Enter New Instruction 文本框中可以输入汇编语言指令,完成后按 Enter 键,新输入的指令将取代指定行上原来的指令,实现“在线汇编”功能。

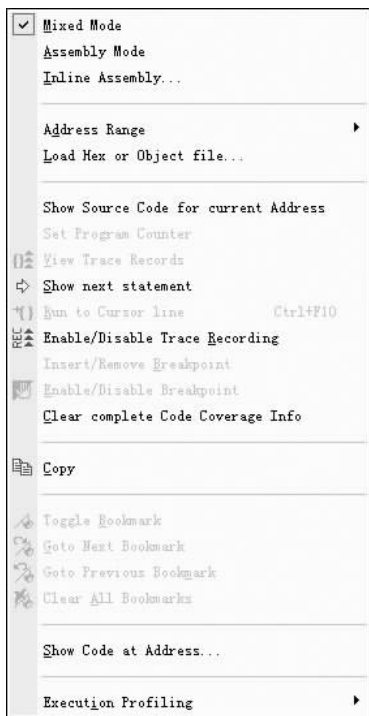


图 2-7 反汇编窗口快捷菜单

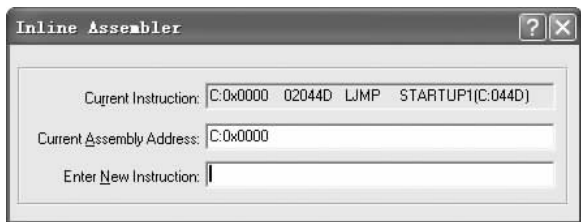


图 2-8 Inline Assembler 对话框

②快捷菜单第二栏。快捷菜单第二栏共有两个选项,Address Range 选项用于显示程序的地址范围;Load Hex or Object file 选项用于重新将 Hex 或 Object 文件装入 Keil μ Vision3 中进行调试。

③快捷菜单第三栏。快捷菜单第三栏共有九个选项:Show Source Code for current Address 选项用于显示源代码和目前地址;Set Program Counter 选项用于设置程序计数器;View Trace Records 选项用于在反汇编窗口显示指令执行的历史记录,该选项只有在 Enable/Disable Trace Recording 选项被选中并且已经执行过程序指令的情况下才能起作用,这时反汇编窗口将以不同颜色显示出已经被执行过指令的历史记录,将光标向上移动可以查看更多历史记录,寄存器窗口的 Regs 选项卡中的显示内容也随之发生变化;Show next statement 选项用于显示下一条将被执行的指令;Run to Cursor line 选项用于将程序执行到

当前光标所在的那一行,使用方法是先将光标定位在需要暂停的程序上,然后选择该选项,程序会执行到指定行暂停;Insert/Remove Breakpoint 选项用于插入或删除程序执行时的断点,使用方法是先将光标定位在希望插入断点的程序行上,然后选择该选项,将在选定行插入一个断点,对于已经存在的断点,选择该选项可删除选定的断点;Enable/Disable Breakpoint 选项用于激活或禁止已经存在的断点;Clear complete Code Coverage Info 选项用于清除代码覆盖信息。

④快捷菜单第四栏。快捷菜单第四栏只有一个 Copy 选项,用于复制反汇编窗口中的内容。具体使用方法是:先选定反汇编窗口中需要复制的内容,右击选择该选项即可将选定的内容复制到剪贴板中,然后再将其粘贴到文件中。这一功能对于保存“在线汇编”时临时修改了的程序文件特别有用,因为“在线汇编”仅在调试程序时对装入软件中的汇编代码起作用,对源文件并未进行修改,退出调试之后,“在线汇编”的作用将不复存在。

⑤快捷菜单第五栏。快捷菜单第五栏用于文本标记操作,其中各个选项与菜单中“文件标记”选项的功能相同。

⑥快捷菜单倒数第二栏的 Show Code at Address 选项用于显示指定地址处的程序代码,选择该选项时反汇编窗口将立即切换到指定地址处的程序代码窗口。

(3)观察窗口。View 菜单的 Watch & Call Stack Window 选项用于调试状态下观察窗口的显示/隐藏切换。观察窗口有四个标签,分别是 Locals、Watch # 1、Watch # 2 和 Call Stack。图 2-9 所示为 Locals 窗口,窗口中显示程序调试过程中当前局部变量的使用情况。单击 Watch # 1 标签可切换到 Watch # 1 窗口显示程序中已经设置了的观察点在调试过程中的当前值。在 Locals 窗口或 Watch # 1 窗口中右击可改变局部变量或观察点的值,按十六进制(hex)或十进制(decimal)方式显示。单击 Call Stack 标签可切换到 Call Stack 窗口显示程序执行过程中对子程序的调用情况。

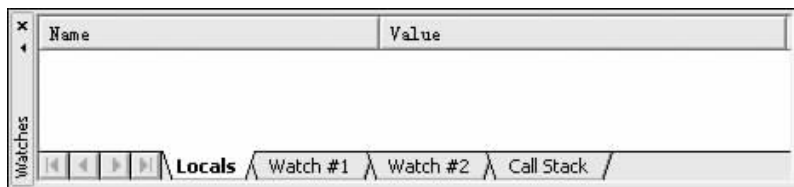


图 2-9 Locals 窗口

(4)存储器窗口。View 菜单的 Memory Window 选项用于系统存储器空间的显示/隐藏切换,选择该选项,弹出如图 2-10 所示存储器窗口。在 Address 文本框中输入存储器地址,将立即显示对应存储器空间的内容。输入地址时要指定存储器类型(C、D、I、X 等)。窗口有 Memory # 1、Memory # 2、Memory # 3 和 Memory # 4 四个标签,每个标签可指定不同的地址空间。

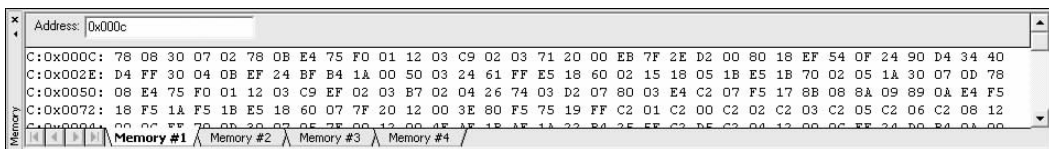


图 2-10 存储器窗口



(5)代码覆盖窗口。View 菜单中的 Code Coverage Window 选项用于代码覆盖窗口的显示/隐藏切换。选择该选项,弹出如图 2-11 所示代码覆盖性能分析窗口。性能分析窗口中显示的是指定程序模块的代码执行情况。在 Current 文本框内输入指定的程序模块名,该模块中各函数包含的指令条数及其已经被执行指令的百分数将会在窗口中显示出来。Update 按钮用于对显示值进行更新,Reset 按钮用于复位被执行指令的百分数。

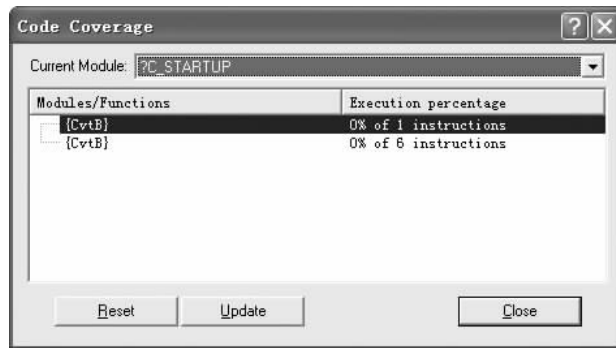


图 2-11 代码覆盖窗口

(6)性能分析窗口。View 菜单中的 Performance Analyzer Window 选项用于性能分析窗口的显示/隐藏切换,选择该选项,弹出如图 2-12 所示性能分析窗口。性能分析窗口用于显示指定程序模块被调用的次数及执行时间,分析结果以棒状图形式显示在窗口中,通过窗口内的一个运行性能统计标尺很容易了解该程序模块的运行性能。窗口中有六个文本框:min 文本框指定程序模块所消耗的最小时间;max 文本框指定程序模块所消耗的最大时间;avg 文本框指定程序模块所消耗的平均时间;total 文本框指定程序模块所消耗的总时间;% 文本框指定程序模块所消耗的时间占全部运行时间的百分数;count 文本框指定程序模块被调用的次数。

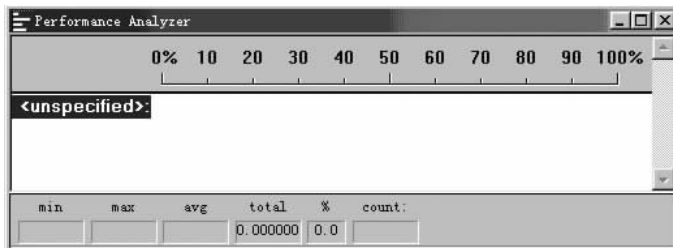


图 2-12 性能分析窗口

在性能分析窗口中右击弹出快捷菜单,共四个选项,其中,Reset PA 选项用于复位所有程序模块的性能分析值;Active PA 选项用于启动/停止性能分析;Update Times 选项用于更新程序执行时间;Setup PA 选项用于设置性能分析窗口的内容。选择 Setup PA 选项,弹出如图 2-13 所示的对话框。其中,Current PA Ranges 文本框显示的是当前已经设定的性能分析程序模块,单击 Kill All 按钮将删除已设定的全部模块,也可以选中其中某一模块再单击 Kill Selected 按钮删除该模块,在 Define Performance Analyzer 文本框中输入需要进行分析的程序模块名,再单击 Define 按钮可将其设定为一个性能分析程序模块,Function

Symbols 文本框中显示了当前所有可用的程序模块,将鼠标指向其中某个模块名并双击,选定的程序模块名将立即出现在 Define Performance Analyzer 文本框中,再单击 Define 按钮即可将其设定为一个性能分析程序模块。

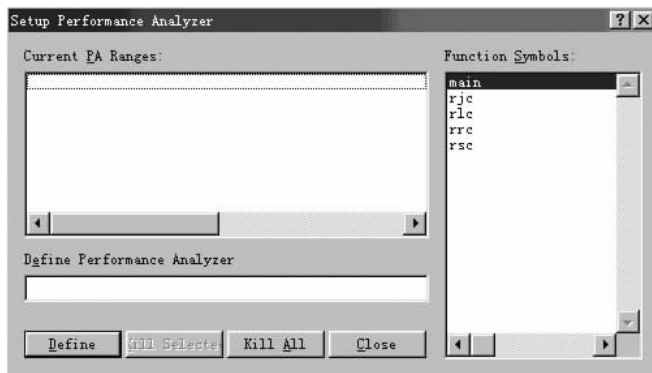


图 2-13 Setup Performance Analyzer 对话框

(7)符号窗口。View 菜单中的 Symbol Window 选项用于符号窗口的显示/隐藏切换,选择该选项,弹出如图 2-14 所示的符号窗口,窗口中显示的是当前程序中的各种符号。在 Mask 文本框中输入希望显示的符号名,对应的符号信息将立即显示在窗口中。选中 Case Sensitive 复选框可区分大小写。

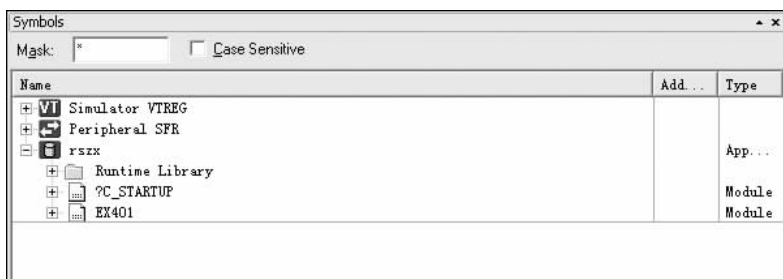


图 2-14 符号窗口

(8)串行窗口。View 菜单中的 Serial Window #1(Serial Window #2)选项用于串行窗口 1(2)的显示/隐藏切换。串行窗口在进行调试程序时十分有用,如果程序中调用了 C51 的库函数 scanf()和 printf(),则必须利用该窗口来完成 scanf()函数的输入操作,printf()函数的输出结果也将显示在该窗口中。利用串行窗口可在程序仿真调试过程中实现人机对话,或者对程序运行结果进行实时显示。在串行窗口中右击弹出快捷菜单,可按需要将窗口内容以十六进制或二进制显示,也可随时清除显示内容。串行窗口中可保持最近 8 KB 串行输入/输出数据,并可进行翻滚显示。

(9)工具箱。View 菜单中的 Toolbox 选项用于工具箱窗口的显示/隐藏切换,选择该选项,弹出如图 2-15 所示的窗口,单击 Update Windows 按钮将对当前显示内容进行更新。



图 2-15 工具箱窗口

在该窗口中,可以根据需要自定义其他命令按钮。自定义按钮可以通过在命令窗口中选择 Define Button 选项来实现,其格式为



```
Define Button "button_label", "button_command"
```

其中, button_label 是按钮名, button_command 必须是一个有效的 Keil μ Vision3 命令。

例 2-1 定义工具箱按钮。

```
>Define Button "go to main" //定义“go to main”按钮
```

```
>Define Button "clr dptr", "dptr=0" //定义“clr dptr”按钮
```

进行正确定义后,对应的按钮将立即出现在工具箱窗口中,每个自定义命令按钮前自动冠以一个编号,以便于删除该按钮。随着自定义按钮数目的增加,工具箱窗口将自动增大。在命令窗口中选择 Kill Button 选项可以删除自定义命令按钮,预定义按钮 Update Windows 前面没有编号,因此它是不能被删除的。

例 2-2 删除工具箱自定义按钮。

```
>Kill Button 1 //删除编号为 1 的自定义按钮
```

```
>Kill Button 2 //删除编号为 2 的自定义按钮
```

(10)周期更新。View 菜单中的 Periodic Window Update 选项用于对各窗口的显示内容进行周期性更新。在进行程序调试时,若希望从观察窗口中看到某个变量值的改变情况,必须选择该选项。

3) Debug 菜单功能

在 Debug 中可以进行两种类型的代码调试:带调试信息的源程序代码调试和 Hex 代码调试。前者允许调试过程中显示高级语言源程序语句,后者仅能显示基本汇编语言指令。在成功完成项目编译连接之后,通过 Debug 菜单进入程序调试状态,在调试状态下仍可通过调试主窗口进行源程序的编辑修改,这也是 Debug 的一大特点,可以根据当前调试结果修改源程序,但修改后不能立即进行调试,要先退出当前调试状态,重新编译连接成为新的目标代码再次装入之后才能调试。

Debug 菜单中的 Start/Stop Debug Session 选项用于启动/停止调试功能,启动之前应先确定是采用软件模拟仿真还是采用硬件目标板仿真。启动调试功能之后,项目窗口自动切换到 Regs 窗口,显示当前单片机内部各寄存器的状态,寄存器状态将随着程序语句或指令的执行而变化。

Debug 菜单第二栏的各选项用于控制目标代码的执行方式。选择 Go 选项,程序从当前地址处开始全速运行,遇到断点或选择 Stop Running 选项时停止。Step 选项用于单步执行,在高级语言显示方式下选择一次该选项执行一条 C 语言语句,在汇编语言显示方式下则执行一条指令,遇到函数调用(高级语言方式)或子程序调用(汇编语言方式)语句时,将跟踪进入函数或子程序中执行。选择 Step Over 选项,程序从当前地址处开始执行一条语句,对于函数或子程序调用语句不跟踪进入被调用函数,而是将整个函数或子程序与调用语句一起一次执行。在调用函数或子程序过程中,选择 Step Out of Current Function 选项,函数或子程序从当前地址处开始执行并返回到调用函数或子程序的下一条语句,该选项对于非函数或子程序以及采用硬件目标板仿真时无效。选择 Run to Cursor Line 选项,程序从当前地址处开始执行到光标所在行。Stop Running 选项用于停止运行程序。

Debug 菜单第三栏各选项用于程序中的断点管理。断点功能对于程序的仿真调试是十分重要的,它可在某个特定地址或是满足某种特定条件下暂停程序,以便观察、了解程序的运行状况,查找或排除错误。选择 Breakpoints 选项将弹出如图 2-16 所示的 Breakpoints 对

话框。Current Breakpoints 文本框用于显示当前已经设置的断点列表,Keil μ Vision3 自动为每个断点分配一个内部序号,序号前面有一个复选框,选中时使能该断点,否则禁止该断点。禁止断点不同于删除断点,它仅仅暂时禁止断点发挥作用,需要时可以重新激活该断点,让它重新发挥作用;Expression 文本框用于输入断点表达式;Count 文本框用于输入断点通过次数,例如,当 Count 的值等于 2 时,表示在第二次运行到该断点时停止程序运行或执行规定的命令;Command 文本框用于输入当程序执行到断点时需要执行的命令;Access 栏用于规定断点的存取类型,选中 Read 复选框表示读,选中 Write 复选框表示写,同时选中这两个复选框表示读写;Size 栏用于规定存取断点的长度,选中 Bytes 复选框时,按断点表达式的值从第一个地址开始计算其字节长度。

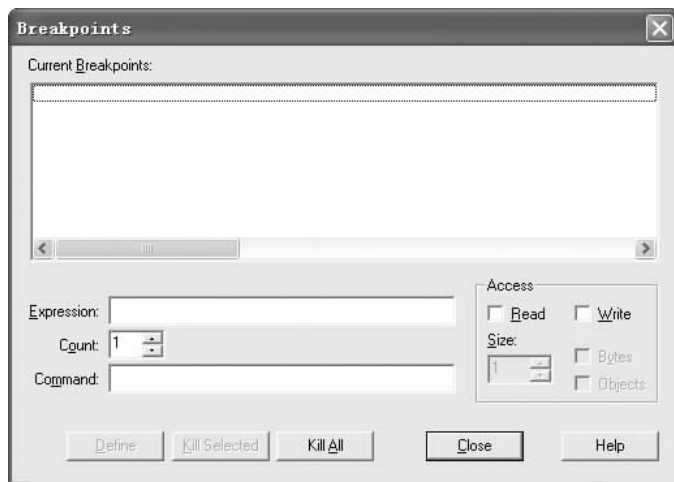


图 2-16 Breakpoints 对话框

常用断点有三种:执行断点(execution breakpoint)、条件断点(conditional breakpoint)和存取断点(access breakpoint)。它们在使用中各有优缺点。若断点表达式为一个特定代码地址,则为执行断点,程序每当运行到该地址时被暂停,如函数地址或语句行号。执行断点必须设置在正确的代码地址处,对于某些代码地址只能设置一次断点,不能重复定义,利用执行断点可以检查程序流程是否正确。若断点表达式不是代码地址,即为条件断点。对于条件断点,每一条指令执行结束后重新计算其逻辑表达式,若计算逻辑结果为假(“0”值),继续执行程序,若计算逻辑结果为真(非“0”值),则暂停执行程序。利用条件断点可以检查程序中的一些特殊错误,但可能会降低程序的执行速度。带有存取类型(Read、Write)标志的断点为存取断点,用于捕捉对于非法存储器地址的存取操作。

双击 Current Breakpoints 文本框中某个断点,与该断点有关的信息将在窗口下边的相关栏中显示出来。选中某个断点后单击 Kill Selected 按钮,可以立即删除该断点,单击 Kill All 按钮将删除全部断点。设置新断点时,可直接在 Expression 文本框内输入断点表达式,在 Command 文本框和 Count 文本框内输入有效的 C51 命令串和计数值(如果 Count 栏为空,则断点计数值默认为 1),对于存取断点还需要在 Access 栏选定相应的存取属性,然后单击 Define 按钮,新断点即设置完成。当 Command 文本框中输入了命令串时,该命令串将在程序到达断点时被立即执行,而程序并不在断点处暂停。若需要在断点处暂停程序,必须在



命令串中增加表达式“_BREAK_=1”。断点命令串为可选项,并非每个断点都需要命令串,因此 Command 文本框可以为空,此时每当程序执行到断点时都将会暂停。下面举几个设置断点的例子。

例 2-3 设置执行断点。

Expression 文本框:main

Command 文本框:printf("main has been reached\n"),_BREAK_=1

Count 文本框:1

在程序中标号“main”处设置一个执行断点,当程序执行到标号“main”时将自动执行命令串“printf("main has been reached\n)”,同时在命令窗口内输出命令串执行结果。由于命令串中存在表达式“_BREAK_=1”,因此程序在断点处被暂停。

例 2-4 设置执行断点。

Expression 文本框:\MEASURE\110

Command 文本框:

Count 文本框:1000

在程序模块 MEASURE 的行号 110 处设置一个执行断点,计数值为 1 000。当程序在第 1 000 次执行到该断点地址时将被暂停,断点起作用之后计数值将变为 1。

例 2-5 设置条件断点。

Expression 文本框:(sindex > 0x0a && sindex < 0x25) || sindex == 0x133

Command 文本框:eval sindex

Count 文本框:1

当满足条件“ $0x0a < sindex < 0x25$ 或 $sindex == 0x133$ ”时程序将被暂停,并执行命令串“eval sindex”,命令执行结果显示在命令窗口中,执行完命令串后继续运行程序。

例 2-6 设置条件断点。

Expression 文本框:\$ == time0 && sindex > 5

Command 文本框:

Count 文本框:1

当程序执行到函数 timer0() 并且当变量 $sindex > 5$ 时,暂停执行程序。

设置条件断点时,其表达式必须表示一个存储器地址和存储器类型,并且要遵循两条规则:表达式结果必须具有唯一的存储器类型;只允许使用 &、&&、<、<=、>、>=、==、!= 等运算符。

例 2-7 设置存取断点。

Expression 文本框:sindex && sindex == 0x133

Command 文本框:

Access 栏:选中 write 复选框

Count 文本框:1

当程序向变量 sindex 进行写入,且当变量 sindex 的值为 0x133 时暂停执行。

除了可以通过上述断点设置窗口来设置断点外,还可以直接通过 Debug 菜单来设置断点。进入调试状态后,选择 Debug 菜单中的 Insert/Remove Breakpoint 选项,可在编辑窗口当前光标所在文件行上插入或删除一个断点;选择 Debug 菜单中的 Enable/Disable Breakpoint

选项,可激活/禁止当前光标所指向的一个断点;选择 Debug 菜单中的 Disable All Breakpoints 选项,将禁止所有已经设置的断点;选择 Debug 菜单中的 Kill All Breakpoints 将删除所有已经设置的断点;选择 Debug 菜单中的 Show Next Statement 选项将在编辑窗口中显示下一条将要被执行的语句。

Debug 菜单中的 Enable/Disable Trace Recording 选项用于激活/禁止程序调试时跟踪指令执行的历史记录,在该选项被激活并且已经执行过程序的情况下,选择 View Trace Records 选项,可以从汇编窗口查看源程序指令执行的历史记录。

Debug 菜单中的 Memory Map 选项用于进行软件模拟仿真时设置存储器空间映像,选择该选项,弹出如图 2-17 所示 Memory Map 对话框。Current Mapped 文本框用于显示所有已被设置的存储器空间映像,选中其中一项并单击 Kill Selected Range 按钮将删除选定的存储器空间映像,删除后不能对该部分存储器进行操作,否则将在命令窗口显示 access violation 错误;Map Range 文本框用于希望设置存储器空间映像的地址范围,格式为

字母前缀:起始地址,字母前缀:终止地址

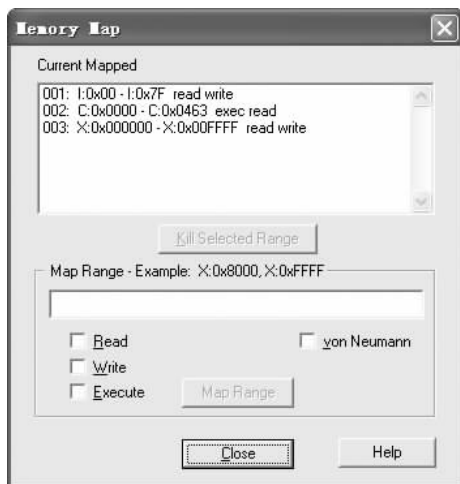


图 2-17 Memory Map 对话框

Keil μ Vision3 软件的存储器空间映像可高达 16 MB,采用以下字母来区分不同的存储器空间映像,见表 2-12。

表 2-12 存储器空间映像设置

字母前缀	存储器空间映像
I(D)	256 B 内部数据存储器(DATA、IDATA)空间,地址范围为 0x00~0xFF
X	64 KB 外部数据存储器(XDATA)空间,地址范围为 0x0000~0xFFFF; 16 MB 外部数据存储器(HDATA)空间,地址范围为 0x000000~0xFFFFFF
C	64 KB 代码存储器(CODE)空间,地址范围为 0x0000~0xFFFF; 16 MB 代码存储器(ECODE)空间,地址范围为 0x000000~0xFFFFFF
B0~B31	分组代码空间,B0 地址范围为:0x0000~0xFFFF;B31 地址范围为:0x0000~0xFFFF



除了可以设置以上标准单片机存储器空间映像之外,Keil μ Vision3 还可以设置四个自定义存储空间映像,字母前缀为 S、T、U、V,地址范围为 0x0000~0xFFFF。

存储器空间映像还带有 Read、Write、Execute 或 von Neumann 属性,带有属性的存储器空间映像只能进行与之相应的操作。内部数据存储空间具有默认的 Read、Write 属性。von Neumann 属性用于将外部数据存储器与代码存储器相重叠,此时对于任何外部数据存储器(XDATA)空间的存取操作,同时发生在代码存储器(CODE)空间。需要注意的是,设置 von Neumann 属性的存储器空间映像应同时具有 Read 和 Write 属性,代码存储器空间不能被设置为 von Neumann 属性。单击 Map Range 按钮即完成存储器空间映像设置。

虽然 Keil μ Vision3 软件支持 16 MB 存储器空间映像,但通常只需要对程序所使用的存储器空间范围进行设置,存储器空间映像设置得过大降低 Keil μ Vision3 的运行速度。启动调试功能并装入目标程序之后,Keil μ Vision3 软件将自动按当前项目创建时对存储器的要求设置存储器空间映像。一般来说,自动设置的存储器空间映像已经可以满足程序调试要求,只有在一些特殊场合才需要重新进行设置。

例 2-8 设置内部数据存储器空间映像。

Map Range 文本框输入“I:0x00,I:0xFF”,选中 Read 复选框和 Write 复选框。

例 2-9 将外部数据存储器空间 0x8000~0xFFFF 与代码存储器空间 0x8000~0xFFFF 相重叠。

Map Range 文本框输入“I:0x8000,I:0xFFFF”,选中 Read 复选框、Write 复选框和 von Neumann 复选框。

例 2-9 中,在外部数据存储器地址 0x8000~0xFFFF 内进行读写操作与在代码存储器地址 0x8000~0xFFFF 内进行读写操作的效果相同。

Debug 菜单中的 Performance Analyzer 选项,用于设定需要进行性能分析的某段程序的地址范围(简称 PA 范围)。

Debug 菜单中的 Inline Assembly 选项,用于程序调试过程中的在线汇编。

Debug 菜单中的 Function Editor(Open Ini File)选项,用于编辑或创建 C51 的初始化文件,初始化文件中可以包含各种 C51 命令及调试函数。选择该选项,弹出如图 2-18 所示窗口,窗口中的 Open 按钮用于打开已有的初始化文件;New 按钮用于创建一个新的初始化文件;Save 按钮和 Save As 按钮用于保存当前编辑的初始化文件;Compile 按钮用于编译并加载当前编辑的初始化文件,如果发生错误,将在 Compile Errors 文本框中显示相应的错误号,同时光标自动跳转到第一个错误所在的行上,以利于编辑修改。对于初始化文件中包含的 C51 命令,编译时将会自动执行。

4) Peripherals 菜单功能

目前,MCS-51 系列单片机已有 400 多个品种和型号,不同型号具有不同的外围集成功能,Keil μ Vision3 通过内部集成器件库实现对各种单片机外围集成功能的模拟仿真,在调试状态下可以通过 Peripherals 菜单来观察仿真结果。Peripherals 菜单的选项内容会根据选用器件库中不同器件而有所变化,图 2-19 所示为选用 8052 单片机器件后的 Peripherals 菜单。