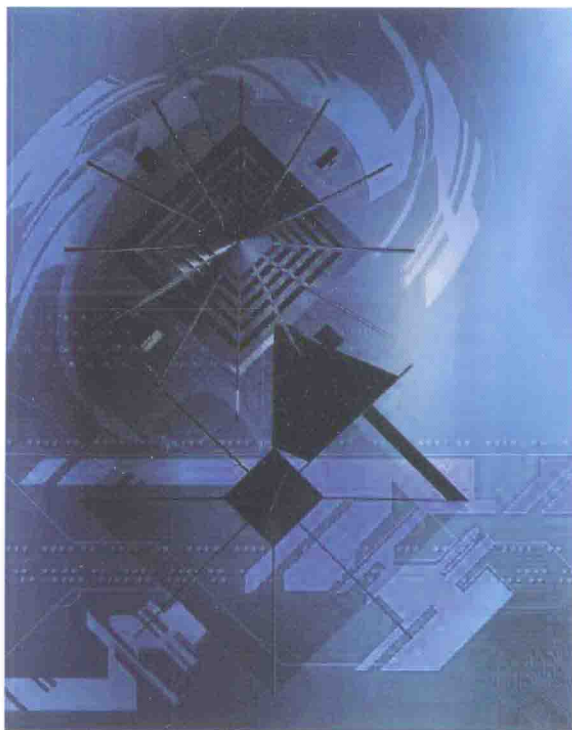


嵌入式系统开发基础

——基于8位单片机的C语言 程序设计 (第二版)

- ◆ 嵌入式控制系统的定义和研究方法
- ◆ RISC微处理器、CISC微处理器简介
- ◆ IAP、ISP技术
- ◆ 光电隔离技术
- ◆ 常用串行芯片接口

人机界面设计



侯殿有 葛海淼 编著



清华大学出版社

高等学校计算机应用规划教材

嵌入式系统开发基础

——基于8位单片机的C语言程序设计

(第二版)

侯殿有 葛海淼 编著

清华大学出版社

内 容 简 介

嵌入式系统大多具有小巧轻薄的特点,程序代码不是很长,系统采用 8 位单片机即可满足要求。本书详细介绍了使用 8 位单片机和 C 语言进行嵌入式系统设计的方法。对于单片机技术的最新发展,如 RISC(Reduced Instruction Set Computing, 精简指令集)微处理器、SoC(System on Chip, 片上系统)设计技术、IAP(In Application Programming, “应用现场”可调试功能)和 ISP(In System Programming, 在系统编程)功能也做了简单介绍。许多串行芯片工作体积小,能耗低,适合嵌入式系统,特别是掌上产品的使用要求,本书对此做了重点介绍。

嵌入式控制系统人机界面设计是进行嵌入式控制系统设计首先遇到的问题,也是难点。本书在详细介绍 LCD 显示汉字、曲线和 ASCII 码原理的基础上,给出了一个通用字模提取和建立小字库程序以及 3 种典型 LCD 显示驱动程序,这些资料对初学者和从事嵌入式开发工作的人员有很大的实用价值。

本书配套的电子课件、配套实验讲义、各章的习题答案和部分工具软件可以到 <http://www.tupwk.com.cn/downpage> 网站下载。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式系统开发基础:基于 8 位单片机的 C 语言程序设计 / 侯殿有,葛海淼 编著. —2 版. —北京:清华大学出版社,2014

(高等学校计算机应用规划教材)

ISBN 978-7-302-36957-8

I. ①嵌… II. ①侯… ②葛… III. ①微型计算机—系统开发—高等学校—教材 ②C 语言—程序设计—高等学校—教材 IV. ①TP360.21 ②TP312

中国版本图书馆 CIP 数据核字(2014)第 135539 号

责任编辑:胡辰浩 袁建华

装帧设计:孔祥峰

责任校对:曹 阳

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 装 者:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:19.25

字 数:481 千字

版次:2014年8月第2版

印 次:2014 年 8 月第1次印刷

印 数:1~3000

定 价:38.00 元

产品编号:057821-01

第二版前言

从 1979 年 Intel 公司首次发布 MCS-51 单片机产品以来,已经过去 30 多年了,在这 30 多年中,单片机世界发生了翻天覆地的变化,出现了许多功能强大,能满足各种嵌入式产品需求的专用或通用嵌入式微处理器。

RISC(Reduced Instruction Set Computing, 精简指令集)微处理器是在 CISC(Complex Instruction Set Computing, 复杂指令集)微处理器基础上,近几年才发展起来的先进技术。

RISC 和 CISC 是目前设计制造微处理器的两种典型技术,虽然它们都试图在体系结构、操作运行、软硬件、编译时间等诸多因素中做出某种平衡,以求达到高效的目的,但采用的方法不同,因此,在很多方面差异很大。

RISC CPU 包含有较少的单元电路,因而面积小、功耗低;RISC 微处理器结构简单、布局紧凑、设计周期短、易于采用最新技术、用户易学易用。所有这些特点,都使 RISC 技术受到众多微处理器厂家的青睐。

随着半导体产业进入超深亚微米乃至纳米加工时代,在单一集成电路芯片上就可以实现一个复杂的电子系统,诸如手机芯片、数字电视芯片、DVD 芯片等。在未来几年内,上亿个晶体管、几千万个逻辑门有望在单一芯片上实现。

SoC(System on Chip, 片上系统)设计技术始于 20 世纪 90 年代中期,随着半导体工艺技术的发展,IC 设计者能够将愈来愈复杂的功能集成到单硅片上,SoC 正是在集成电路(IC)向集成系统(IS)转变的大方向下产生的。SoC 是集成电路发展的必然趋势,是技术发展的必然,是 IC 产业的未来。

此外,CPU 的工作频率从最初的几 MHz,发展到现在的几十到上百 MHz。片上程序存储器容量从原来的最大 4KB ROM/EPROM 做到现在的 4KB Flash、甚至达到 64KB Flash,使得单片机可以不使用仿真器在线编程,即具有 IAP(In Application Programming,“应用现场”可调试功能)和 ISP(In System Programming,在系统编程,无须将存储芯片从嵌入式设备拔出即可对其编程,简称 ISP)功能。

许多单片机片上集成了 WDT 技术(watchdog,看门狗定时器)、SPI(Serial Peripheral Interface, 串行外设接口)、A/D 转换电路,使单片机无须扩展就是一个嵌入式开发系统。

还有的单片机采用多核结构,提高了浮点运算速度,增强了数字信号处理能力。

单片机工作电压范围更宽,出现了工作电压 2.2~6V 的微处理器,特别适合电池供电的手持式嵌入式产品。

在专用微处理器方面也出现了许多新产品。例如,日本脉冲马达株式会社(npm, nippon pulse motor co., ltd)的 pcl6045b 运动控制微处理器芯片,是一种通过总线接收命令、并产生脉冲控制步进电机或脉冲驱动型伺服电机的 cmos 专用微处理器,可以广泛应用于数控机床、机器人等数控机械的运动控制中。

德州仪器公司 TI 的 MSP430, 为高整合、高精度的微处理器系统, 是目前具有最低功耗的 flash 16-bits RISC 微控制器。可方便地实现心电信号的采集、处理、存储和打印。

美国 Veridicom 公司的 FPS100 固态指纹传感器是一种直接接触的指纹采集器件, 它是一种低功耗、低价格、高性能的电容式指纹提取微控制器。

这方面例子很多, 不一一列举。

面对如此众多的单片机产品, 初学者如何去学习和掌握呢? 我们研制一个嵌入式产品如何选择做为核心的微处理器呢?

我们知道, 单片机产品种类繁多, 功能各异, 但它们大多是从 MCS-51 单片机基础上发展而来, 只是在功能上进行了增减, 指令系统基本相同, 因此, 我们只要掌握了 MCS-51 单片机的硬件结构和软件编程, 遇到其他处理器就比较容易处理了。

对于单片机选型, 要综合考虑其功能指标。主频过高, 当然运算速度快, 但主频过高系统耗能就高, 电磁干扰加大, 且容易发热, 导致系统稳定系数下降, 因此, 运算速度满足要求即可, 不必盲目追求主频过高产品。

现在出现了许多串行芯片, 这些芯片一般体积小, 能耗低, 适合嵌入式系统小、巧、轻、灵、薄的使用要求, 特别是掌上产品的要求。但串行芯片软件编程比并行芯片复杂, 时序要求严格, 特别是运行速度比并行芯片慢, 因此, 到底使用哪种芯片要综合考虑。

虽然 8 位单片机功能很强, 可以处理大多数嵌入式问题, 但对于多媒体、多线程、与 Internet 的连接、图像处理等较复杂问题就比较困难, 此时, 应选择功能更强的单片机, 如以 ARM 为代表的 32 位单片机, 它有多种型号, 以适应不同的应用场合。

嵌入式控制系统人机界面设计是我们进行嵌入式控制系统设计首先遇到的问题, 本书对此进行了详细讨论, 这部分内容是本书的重点。

本书资料主要来源是: 多年 MCS-51 单片机 C 语言程序设计教学中使用的资料、作者在 20 多年科研工作和指导学生参加各种大赛中积累的经验、程序、网上资源、一些公司的产品使用说明书或技术资料。

作者对使用或借鉴资料的个人或公司表示感谢。本书由侯殿有编写 1~11 章; 葛海淼编写 12~17 章。

对本书的意见和建议请与我们联系, 联系信箱是 huchenhao@263.net, 联系电话是 010-62796045。

设计人机界面, 要提取字模建立小字库, 本书通用字模提取程序功能强大、使用方便, 可从 <http://www.tupwk.com.cn/downpage> 网站下载, 使用密码是 194512125019。

目

录

第1章 嵌入式控制系统概论.....1

- 1.1 单片机和嵌入式控制系统的
定义、嵌入式系统的分类.....1
- 1.2 MCS-51 单片机在嵌入式
控制系统中的地位和作用.....2
- 1.3 嵌入式控制系统的研究方法.....2
 - 1.3.1 交叉编译环境 Keil C.....2
 - 1.3.2 Keil C51 的安装.....3
- 1.4 程序的编辑、编译、调试和
运行.....5
 - 1.4.1 建立项目.....5
 - 1.4.2 项目的运行模式.....6
 - 1.4.3 项目的编译模式.....7
 - 1.4.4 项目的调试.....8
- 1.5 系统软件资源.....10
- 1.6 习题.....12

第2章 MCS-51 单片机系统和 系统扩展.....13

- 2.1 MCS-51 系列单片机.....13
- 2.2 MCS-51 单片机的外部
引脚和总线.....14
 - 2.2.1 输入/输出引脚.....14
 - 2.2.2 MCS-51 单片机的控制线.....15
 - 2.2.3 MCS-51 单片机的片外
总线.....15
 - 2.2.4 MCS-51 单片机存储器
结构.....16
- 2.3 MCS-51 单片机的最小系统.....21
 - 2.3.1 8051/8751 的最小系统.....21
 - 2.3.2 8031 最小应用系统.....22
- 2.4 MCS-51 单片机系统扩展.....22

2.4.1 存储器扩展概述.....22

2.4.2 存储器扩展的讨论.....23

2.5 输入/输出扩展和使用.....24

2.5.1 简单 I/O 接口扩展.....24

2.5.2 I/O口在TTL电路中使用.....26

2.5.3 I/O口在外围设备中使用.....28

2.6 习题.....30

第3章 STC 89C51/89C52

单片机介绍.....31

3.1 89C51/89C52 单片机资源 和使用.....31

3.1.1 89C51/89C52 单片机
片内资源.....31

3.1.2 89C52 单片机程序调试.....33

3.2 89C52 最小系统和仿真器 使用.....34

3.2.1 89C52 最小系统.....34

3.2.2 仿真器使用.....34

3.3 习题.....35

第4章 C51 基本语句.....36

4.1 C 语言的特点及程序结构.....36

4.1.1 C语言的特点.....36

4.1.2 C语言和C51的程序结构.....36

4.2 C51 数据类型.....38

4.2.1 字符型(字节型) char.....38

4.2.2 int 整型.....38

4.2.3 long 长整型.....38

4.2.4 float 浮点型.....38

4.2.5 指针型.....38

4.2.6 特殊功能寄存器型.....39

4.2.7 位类型.....39

4.3 C51 的运算量.....40	5.3 函数的嵌套与递归.....65
4.3.1 常量.....40	5.4 局部变量和全局变量.....66
4.3.2 变量.....41	5.4.1 局部变量.....66
4.3.3 存储模式.....44	5.4.2 全局变量.....67
4.3.4 绝对地址的访问.....44	5.5 习题.....68
4.4 C51 的运算符及表达式.....46	第 6 章 C51 构造数据类型.....69
4.4.1 赋值运算符.....46	6.1 数组.....69
4.4.2 算术运算符.....46	6.1.1 一维数组.....69
4.4.3 关系运算符.....47	6.1.2 字符数组.....70
4.4.4 逻辑运算符.....47	6.2 指针.....71
4.4.5 “位”运算符.....48	6.2.1 指针的概念.....71
4.4.6 复合赋值运算符.....48	6.2.2 指针变量的定义.....72
4.4.7 逗号运算符.....49	6.2.3 指针变量的引用.....72
4.4.8 条件运算符.....49	6.3 结构.....73
4.4.9 指针与地址运算符.....49	6.3.1 结构与结构变量的定义.....73
4.5 表达式语句及复合语句.....50	6.3.2 结构变量的引用.....74
4.5.1 表达式语句.....50	6.4 联合.....76
4.5.2 复合语句.....50	6.4.1 联合的定义.....76
4.6 C51 的输入输出.....50	6.4.2 联合变量的引用.....77
4.6.1 格式输出函数 printf().....51	6.5 枚举.....77
4.6.2 格式输入函数 scanf().....51	6.6 习题.....78
4.7 C51 程序基本结构与 相关语句.....52	第 7 章 MCS-51 可编程并行 I/O 接口.....79
4.7.1 C51 的基本结构.....52	7.1 可编程并行 I/O 接口 8255A.....79
4.7.2 if 语句.....54	7.1.1 8255A 的结构和工作方式.....79
4.7.3 switch/case 语句.....55	7.1.2 8255A 与 MCS-51 单片机的 硬件接口与编程.....84
4.7.4 while 语句.....56	7.2 可编程 I/O 扩展接口 8155.....86
4.7.5 do while 语句.....56	7.2.1 8155 的结构和工作方式.....86
4.7.6 for 语句.....57	7.2.2 8155 与 MCS-51 单片机的 连接和软件编程.....88
4.7.7 循环的嵌套.....57	7.3 步进电机控制电路.....90
4.7.8 break 和 continue 语句.....58	7.4 输入输出程序编写.....92
4.7.9 return 语句.....58	7.5 习题.....94
4.8 习题.....59	
第 5 章 C51 函数.....61	
5.1 函数的定义.....61	
5.2 函数的调用与声明.....63	

第8章 MCS-51单片机的中断系统..... 95	
8.1 中断的基本概念.....95	10.5.1 串行口的编程步骤.....118
8.2 MCS-51 单片机的中断系统.....96	10.5.2 串行口的应用实例.....119
8.2.1 MCS-51 单片机的中断源.....96	10.6 RS232 和 RS422、RS485
8.2.2 优先级控制.....97	通信.....124
8.2.3 中断响应.....99	10.6.1 RS232 通信.....124
8.2.4 中断应用举例.....100	10.6.2 RS-422 与 RS-485
8.3 习题.....101	串行接口.....125
第9章 MCS-51 定时器/计数器	10.7 习题.....127
接口.....102	第11章 MCS-51 与键盘、
9.1 定时器/计数器接口概述.....102	显示器的接口.....128
9.1.1 定时/计数器的主要特性.....102	11.1 MCS-51 单片机与键盘
9.1.2 定时/计数器 T0、T1 的	接口.....128
结构及工作原理.....102	11.1.1 独立式键盘.....128
9.2 定时/计数器的工作方式	11.1.2 行列式键盘.....130
寄存器和控制寄存器.....103	11.2 MCS-51 单片机与 LED
9.2.1 定时/计数器的方式	显示器接口.....133
寄存器 TMOD.....103	11.2.1 LED 显示器的结构与
9.2.2 定时/计数器的控制	原理.....133
寄存器 TCON.....104	11.2.2 LED 数码管显示器的
9.3 定时/计数器的工作方式.....105	译码方式.....134
9.4 定时/计数器的初始化	11.2.3 LED 数码管的显示.....135
编程及应用.....106	11.2.4 LED 显示器与单片机的
9.4.1 定时/计数器的编程.....106	接口.....135
9.4.2 定时/计数器的应用.....106	11.3 串行键盘/显示芯片
9.5 习题.....111	HD7279A 介绍.....140
第10章 MCS-51单片机串行接口.....112	11.3.1 HD7279A 简介.....140
10.1 通信的基本概念.....112	11.3.2 HD7279A 命令时序.....144
10.2 MCS-51 单片机串行口	11.3.3 HD7279A 与 MCS-51
功能与结构.....113	单片机接口.....145
10.3 串行口的工作方式.....115	11.3.4 HD7279A 驱动程序.....146
10.3.1 方式 0.....115	11.4 习题.....148
10.3.2 方式 1.....116	第12章 MCS-51 与常用
10.3.3 方式 2 和方式 3.....116	串行芯片接口.....150
10.4 串行口波特率计算.....117	12.1 MCS-51 单片机与 I ² C
10.5 串行口的编程和应用.....118	总线芯片接口.....150
	12.1.1 I ² 总线简介.....150

12.1.2	I^2C 总线与 MCS-51 单片机接口.....	153	14.1.4	屏幕上“打点”	190
12.1.3	CAT24WCXX 与单片机 的接口与编程	154	14.1.5	汉字显示概述	191
12.2	MCS-51 单片机与串行 日历时钟芯片接口	159	14.2	汉字字符集介绍	193
12.2.1	串行日历时钟芯片 DS 1302 简介	159	14.3	汉字的内码	193
12.2.2	DS1302 的输入输出	164	14.4	内码转换为区位码	194
12.3	单总线(1-wire)数字温度 传感器的接口	169	14.5	字模提取与小字库建立	194
12.3.1	DS18B20简介	169	14.5.1	用 C 语言提取 16×16 点阵字模	194
12.3.2	DS18B20的内部结构	170	14.5.2	24×24 点阵字模的 C 语言提取程序	197
12.3.3	DS18B20 的温度 转换过程	173	14.5.3	用 Delphi 提取字模和 建立小字库	200
12.3.4	DS18B20 的软件 驱动程序	175	14.5.4	通用字模提取程序 MinFonBase 使用说明	210
12.4	习题	177	14.6	汇编语言字模与 C 语言 字模互相转换	211
第 13 章	MCS-51 与 D/A、A/D 的接口	178	14.6.1	汇编语言字模转换为 C 语言字模	211
13.1	A/D、D/A 转换原理及 常用芯片介绍	178	14.6.2	C 语言字模转换为 汇编语言字模	213
13.1.1	D/A 转换器概述	178	14.7	自造字符和自造图形 点阵方法	216
13.1.2	A/D 转换器原理	179	14.7.1	自造字符点阵方法	216
13.2	PCF8591 8 位 A/D 和 D/A 转换芯片	180	14.7.2	自造图形点阵方法	216
13.2.1	PCF8591 一般介绍	180	14.8	习题	217
13.2.2	PCF8591 软件编程	181	第 15 章	T6963C 的汉字字符显示	218
13.3	习题	187	15.1	T6963C 的一般介绍	218
第 14 章	汉字和西文字符显示原理	188	15.1.1	T6963C 的硬件特点	218
14.1	英文字符在计算机中的 表示	188	15.1.2	T6963C 的引脚说明 及功能	219
14.1.1	ASCII 码	188	15.1.3	T6963C 的状态字	220
14.1.2	英文字符的显示	189	15.2	T6963C 指令系统	221
14.1.3	其他西文字符在计算机中 的存储和显示	190	15.2.1	指针设置指令	221
			15.2.2	控制指令	222
			15.2.3	数据读写指令	224
			15.2.4	屏操作指令	225

15.2.5	位操作指令	225
15.3	T6963C 和单片机的连接	225
15.3.1	直接连接	225
15.3.2	间接连接	226
15.4	T6963C 的驱动程序	227
15.5	T6963C 的内嵌字符表	240
15.6	习题	241
第 16 章 KS0108 液晶显示器		
	驱动控制	242
16.1	KS0108 液晶显示器概述	242
16.1.1	KS0108 的硬件特点	242
16.1.2	KS0108 与微处理机的 接口	244
16.1.3	KS0108 的电源和对 比度调整	244
16.2	KS0108 的指令系统	245
16.2.1	显示开/关指令	245
16.2.2	行列设置命令	246
16.2.3	数据和状态读写命令	246
16.3	KS0108 的软件驱动程序	247
16.4	ASCII 8×8 字符库	269
16.5	习题	271

第 17 章		HD61830 液晶显示器	
		驱动控制	272
17.1	HD61830 液晶显示器		
		概述	272
17.2	HD61830 的指令系统		274
17.2.1	方式控制指令		274
17.2.2	显示域设置指令		275
17.2.3	光标设置指令		276
17.2.4	数据读写指令		277
17.2.5	“位”操作指令		277
17.3	HD61830 液晶显示器		
		驱动控制程序	277
17.3.1	HD61830 液晶显示器		
		显示 RAM 结构	277
17.3.2	软件程序		278
17.4	HD61830 CGRAM 字符		
		代码表	294
17.5	习题		294
参考文献			296

第1章 嵌入式控制系统概论

嵌入式控制系统设计是当前 IT 行业最热门的话题之一。什么是嵌入式控制系统？它们如何分类？各类系统的设计方法又有什么不同？本章将结合最常用的 MCS-51 单片机来一一回答这些问题。

1.1 单片机和嵌入式控制系统的定义、嵌入式系统的分类

单片机就是在一片半导体硅片上，集成了中央处理单元(CPU)、存储器(RAM/ROM)和各种 I/O 接口的微型计算机。这样一块集成电路芯片，具有一台微型计算机的功能，因此被称为单片微型计算机，简称单片机。

有些单片机功能比较齐全，我们称之为通用单片机；有些单片机则是专门为某一应用领域研制的，突出某一功能，例如专门的数控芯片、数字信号处理芯片等，我们称之为专用单片机。有时我们也把这两种单片机统称为微处理器。

单片机主要应用在测试和控制领域，由于单片机在使用时，通常处于测试和控制领域的核心地位，并嵌入其中，因此我们也常把单片机称为嵌入式控制器(Embedded MicroController Unit)，把嵌入某种微处理器或单片机的测试和控制系统称为嵌入式控制系统(Embedded Control System)。

在本书后面的叙述中，单片机和嵌入式控制器的意义是相同的。

嵌入式控制系统在航空航天、机械电子、家用电器、自动控制等各个领域都有广泛的应用，特别是家用电器领域更是嵌入式控制系统最大的应用领域，MP3、MP4、MP5、数码相机、扫描仪、个人 PC、车载电视、DVD、PDA(掌上电脑)等，到处都可以看到嵌入式控制系统的应用。

随着超大规模集成电路工艺和集成制造技术的不断完善，单片机的硬件集成度也在不断提高，已经出现了能够满足各种不同需要、具有各种特殊功能的单片机。在 8 位单片机得到广泛应用的基础上，16 位单片机和 32 位单片机也应运而生，特别是以 ARM 技术为基础的 32 位精减指令系统单片机(RISC Microprocessor)，由于其性能优良、价格低廉，大有取代 16 位单片机而成为高档主流机型的趋势。

嵌入式控制系统由于其内核嵌入的微处理器不同，在应用上大致可分为两个层次：在系统简单、要求不高，成本低的应用领域，大多采用以 MCS-51 为代表的 8 位单片机；随着嵌入式控制系统与 Internet 的逐步结合，PDA、手机、路由器、调制解调器等复杂的高端应用对嵌入式控制器提出了更高的要求，在少数高端应用领域，以 ARM 技术为基础的 32 位精减指令系统单片机得到越来越多的青睐。嵌入式控制系统的高端应用领域又分为代嵌入式操作

系统支持和不代嵌入式操作系统支持两种情况。

1.2 MCS-51 单片机在嵌入式控制系统中的地位和作用

1980 年, Intel 公司在 MCS-48 单片机基础上推出了 MCS-51 单片机, MCS-51 单片机包括 3 个基本型 8031、8051、8751, 还包括 3 个 CMOS 工艺的低功耗型 80C31、80C51、87C51。

虽然它们是 8 位单片机, 但是它们品种多、兼容性好、功能强、价格低廉、性能稳定和使用方便, 特别是设计和应用资料齐全, 因此受到广大工程技术人员的青睐, 成为我国应用最为广泛的机种。在今后相当长的一段时间, MCS-51 单片机还是嵌入式控制系统的主流机型。

由于 MCS-51 单片机技术先进, 性能稳定, 世界上许多大的半导体公司也在根据 Intel 公司技术生产 MCS-51 单片机或改进型 MCS-51 单片机。因此, MCS-51 单片机已经成为 8 位单片机的实际技术标准, 也是嵌入式控制系统中使用最多的嵌入式控制器。

在计算机技术飞跃发展的今天, 16 位和 32 位单片机已经出现并逐步得到推广应用, 但 MCS-51 单片机的应用依然非常广泛。MCS-51 单片机的设计思想在 16 位和 32 位单片机中得到了进一步的继承和发展。

我们掌握了 MCS-51 单片机的 C 语言程序设计方法, 可以完全满足一般嵌入式控制系统的设计要求, 因为嵌入式控制系统大多具有小、巧、轻、灵、薄的特点, 中小简单系统占嵌入式控制系统的绝大多数, 少数高端应用我们遇到的较少。另外, 掌握了 8 位嵌入式控制系统的设计方法也为进一步学习 16 位和 32 位嵌入式控制系统打下基础。

1.3 嵌入式控制系统的研究方法

1.3.1 交叉编译环境 Keil C

作为嵌入式控制器的单片机, 不管是 8 位单片机, 还是 16 位单片机或是 32 位单片机, 由于受其本身资源限制, 其应用程序都不能在其本身开发, 要开发其应用程序, 还需要一台通用计算机, 如常用的 IBM-PC 机或兼容机, Windows 95/98/2000 或 XP 操作系统, 16MB 以上内存, 20MB 以上硬盘存储空间(运行交叉编译环境 Keil C 的最低配置)。我们也称这台通用计算机为“宿主机”, 称作为嵌入式控制器的单片机为“目标机”, 应用程序在“宿主机”上开发, 在“目标机”上运行。“目标机”和“宿主机”之间利用计算机并口或 USB 口通过一台叫做“仿真器”的设备相连, 编译好的计算机可以识别的目标程序(二进制代码程序)可以从“宿主机”传到“目标机”, 这也叫程序下载, 也可以从“目标机”传到“宿主机”, 叫程序上传。应用程序通过“仿真器”的下载和上传, 在“宿主机”上反复修改, 这个过程叫做“调试”。调试好的应用程序, 在“目标机”上编译成“宿主机”可以直接执行的机器码文件, 通过一台叫“固化器”的设备下载并固化到“目标机”的程序存储器中(8 位单片机常用的程序存储器是 EPROM 或 Flash), 整个下载过程, 叫做烧片, 也叫做程序固化。

程序固化是单片机开发的最后一步,以后“宿主机”和“目标机”就可以分离,“宿主机”的任务完成。“目标机”可以独立执行嵌入式控制器的任务。嵌入式控制系统开发过程如图 1-1 所示。

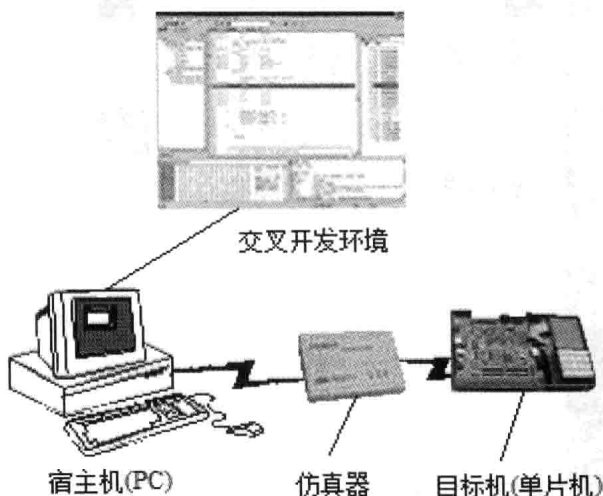


图 1-1 嵌入式控制系统开发过程

通过以上叙述可知,在“宿主机”上运行的开发工具软件的功能非常重要,我们也称这套开发工具软件为交叉编译环境或集成开发环境,交叉编译环境首先应具有类似“Word”的功能,对我们用 C 语言编写的程序进行编辑,同时它还具有调试和编译功能,可以把调试好的应用程序编译成“目标机”可以直接执行的机器码文件。

在我国, MCS-51 单片机的开发大多使用德国 Keil 公司的 μ Vision2/3 或南京伟福的 Wave6000, μ Vision2/3 也叫 Keil C51, 是一款非常优秀的 MCS-51 开发工具,它功能强大、使用方便,特别是运行稳定、抗干扰和防病毒能力强,给使用者留下了深刻印象。

在清华大学出版社网站<http://www.tupwk.com.cn/download>上可下载本书免费学习参考资料,内有 Keil C, 供读者下载学习使用。Wave6000 可以从南京伟福官方网站<http://www.wave-cn.com>免费下载。

1.3.2 Keil C51 的安装

打开单片机编译器文件夹,再打开 setup 子文件夹,出现图 1-2 所示的画面,双击 setup.exe 文件,出现如图 1-3 所示的选择安装类型对话框,第一次安装,选择第一项。单击 Next 按钮,出现如图 1-4 所示的选择安装版本对话框,选择 Full Version, 系统就开始安装,确定安装路径“C:\Keil”和同意版权协议后,系统还要求输入产品系列号,系列号在 UP51V701.TXT 文件中。

接着我们在如图 1-5 中单击 Browse 按钮,在上一级文件夹中找到 PK51 专业开发软件路径 C51addon 文件夹,选中并确定,出现如图 1-6 所示的画面,继续单击 Next 按钮即可一步步完成安装。

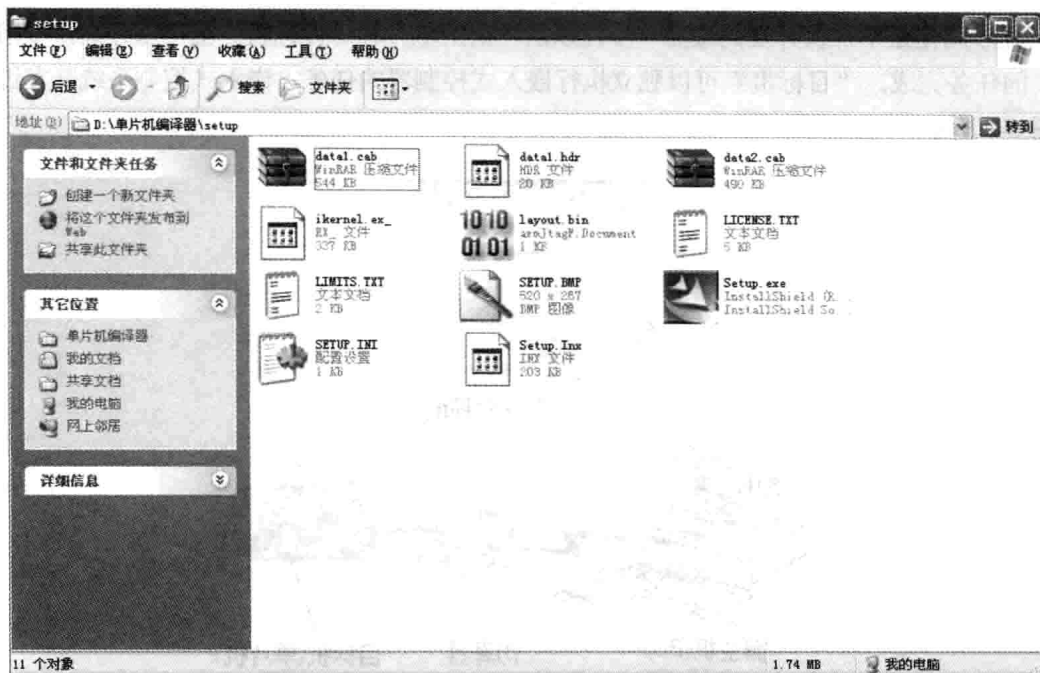


图 1-2 Keil C 安装初始画面

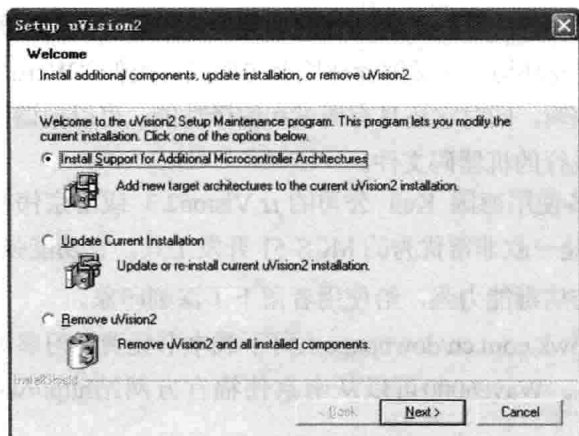


图 1-3 选择安装类型

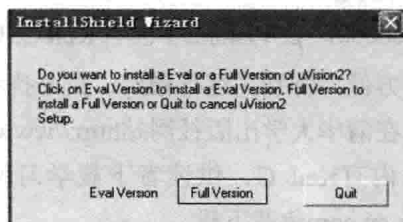


图 1-4 选择安装版本

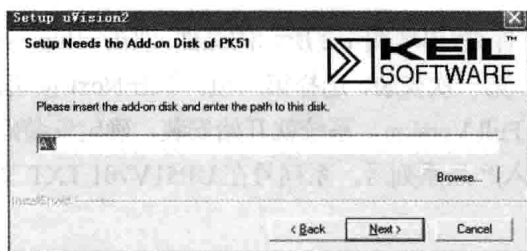


图 1-5 安装 PK51 专业开发软件

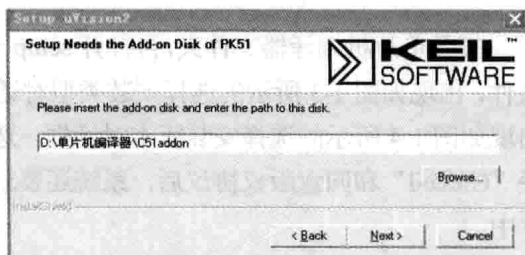


图 1-6 PK51 专业开发软件路径

1.4 程序的编辑、编译、调试和运行

1.4.1 建立项目

MCS-51 单片机程序的编辑和调试均在交叉编译环境 Keil C51 中完成, 非常方便。Keil C 的快捷方式图标如图 1-7 所示, 可以把它放在桌面上以方便使用。双击该图标, 就会出现图 1-8 所示的交叉编译环境 Keil C 的主界面, MCS-51 单片机程序的编辑和调试均在此界面中完成, 今后我们会经常在此界面上工作。



图 1-7 Keil C51 的快捷方式图标

Keil C 对程序的编辑、编译和调试, 都是以“项目”为单位来进行的, 在一个项目中可以包含后缀为.C 的 C 语言源文件、后缀为.h 的 C 语言头文件、后缀为.A 的汇编语言文件、后缀为.o 的机器码文件(C 语言文件经编译后形成的文件)、后缀为.LIB 的库文件(一个库文件中保存同一类功能的一些文件, 这些文件又可以是后缀为.C 的 C 语言源文件、后缀为.h 的 C 语言头文件、后缀为.A 的汇编语言文件、后缀为.o 的机器码文件, 还可以是另一个后缀为.LIB 的库文件)。

Keil C 在对“项目”进行编辑时, 会根据每一个程序的不同后缀名调用不同的编译工具分别把它们转换为后缀为.o 的机器码文件, 然后再调用链接工具文件 Link 根据“项目”结构把它们链接成一个统一的后缀为.exe 的可执行文件。

因此, 使用 Keil C 进行嵌入式控制系统程序开发时, 首先要新建一个项目, 在开发环境主菜单中, 选择 Project | New Project 命令, 就会出现如图 1-9 所示的“新建项目”对话框, 我们给项目起个名字: HELLO, 名字的后缀 Uv2 是系统自动加的, 表示这是一个 Keil C 项目。

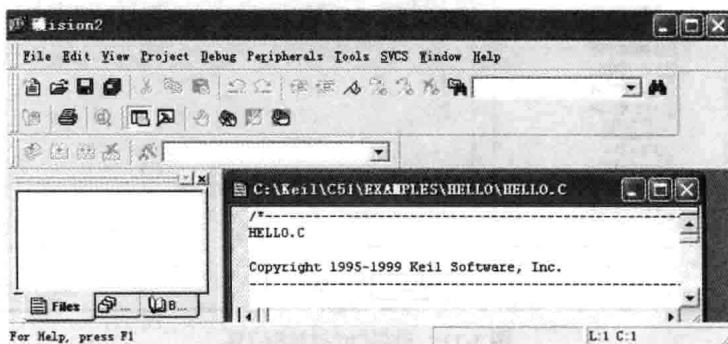


图 1-8 Keil C 的集成开发环境

选择好保存路径, 单击“保存”按钮即可新建项目。之后出现“选择设备”对话框, 如图 1-10 所示, 就是让我们为项目选择一款单片机, 例如我们选择 Intel 公司的 8031AH, 就会

出现该设备的描述信息,如图 1-11 所示,确认后返回主界面,即完成了新建项目的工作。

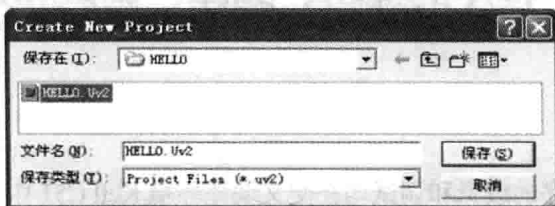


图 1-9 新建 Keil C 项目

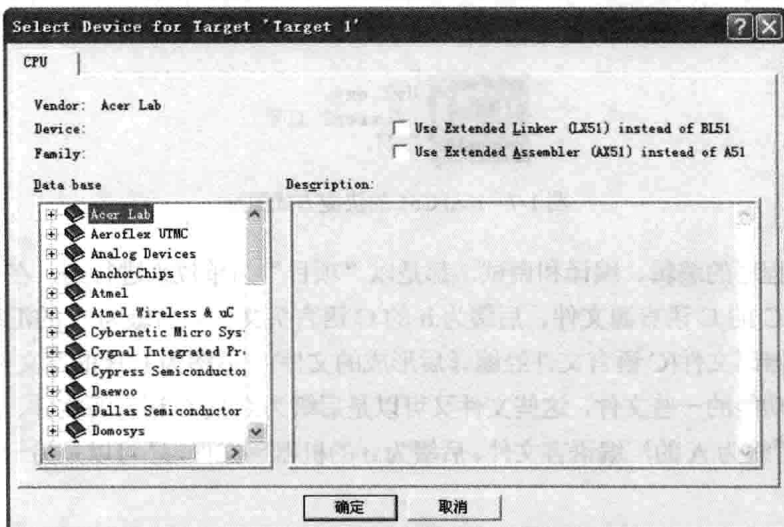


图 1-10 “选择设备”对话框

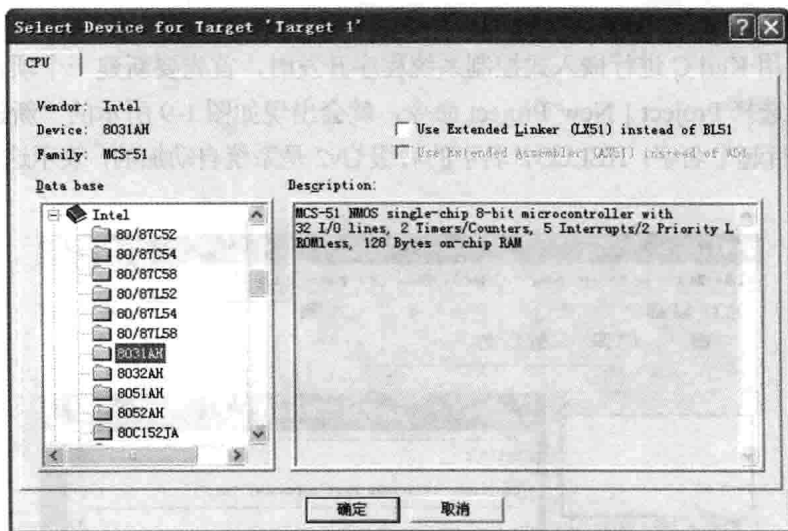


图 1-11 设备功能描述信息

1.4.2 项目的运行模式

为了使项目能够在最优化情况下运行,Keil C 对项目进行了分类,编译后代码小于 2K 的

项目为小模式(Small 模式),其他为中模式(Compact 模式)或大模式(Large 模式)。

单片机虽然功能很强,但本身资源毕竟有限,特别是片上数据存储器,只有 128B(51 系列)或 256B(52 系列),有时候我们必须要在片外对数据存储器进行扩展。在项目为小模式时,只使用片上数据存储器就满足系统要求了,程序用到的变量或函数调用时用到的参数可以放在片上数据存储器中,这种情况下项目占用系统资源少、运行速度快。

虽然小模式占用系统资源少、运行速度快,但代码容量太小,在工程上一般采用大模式(Large 模式)。大模式允许数据存储器和程序代码分别为 64K,完全可以满足嵌入式控制系统的要求。中模式实际使用比较少,本书不做介绍。我们后面的例子程序均采用大模式。

在编译项目之前,要先确定使用的模式,可按如下步骤进行:

在主界面中,右击 Target1(对象 1),从弹出的快捷菜单中选择“Options for target ‘target1’”命令,打开“对象 1 设置”对话框,设置 Memory Model 为大模式:“Large Variables in XDATA”(大模式,变量放片外数据存储器),如图 1-12 所示。

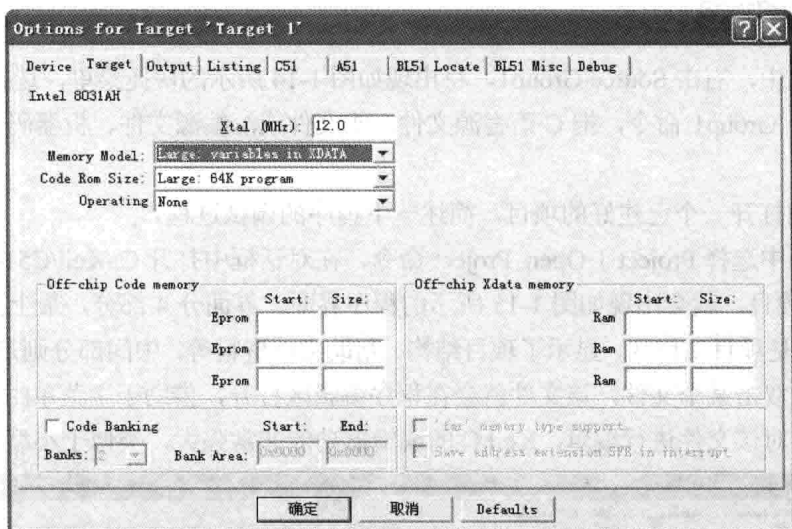


图 1-12 编译选大模式对话框

1.4.3 项目的编译模式

前面已经讲过,Keil C 可以将项目编译成后缀为.o 的机器码(也叫目标码)文件,也可以将多个目标码文件通过 Link 链接成一个后缀为.exe 的可执行文件。

如果将项目编译成后缀为.o 的机器码文件后不能直接执行,那么还要和其他后缀为.o 的机器码文件通过 Link 链接成一个后缀为.exe 的可执行文件才可执行。

一般我们的项目都较小,希望将项目编译成后缀为.o 的机器码文件后,编译器直接调用 Link 将其链接成后缀为.exe 的可执行文件,这可按如下步骤进行。

在“Options for target ‘target1’”(对象 1 设置)对话框中,选中 Output 选项卡,然后选中 Debug Information 和 Create Hex Files 复选框即可,如图 1-13 所示。至此,为项目选设备和该项目编译器设置完成。