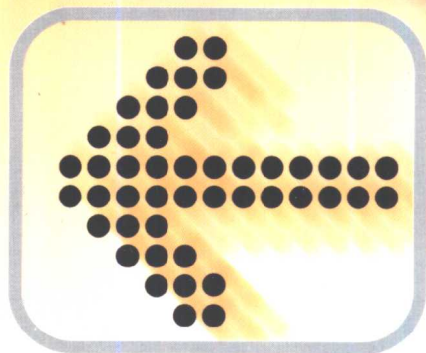
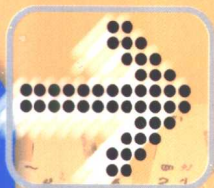




基于C语言

编程

MCS-51 单片机原理与应用

张培仁 主编

孙占辉 张欣 张村峰 编著



清华大学出版社

基于 C 语言编程 MCS-51 单片机原理与应用

张培仁 主编

孙占辉 张欣 张村峰 编著

清华大学出版社

(京)新登字 158 号

内 容 提 要

本书针对 MCS-51 系列单片机,利用 C 语言和汇编语言,以 Franklin C51 编译器和 Franklin L51 连接器为例,讲解单片机原理及应用。

书中,首先简要介绍 MCS-51 系列单片机的原理;然后讲解 C 语言编程的优点及编程注意事项,突出强调 C 语言和汇编语言混合编程时应注意的问题;最后结合作者多年的实践经验,通过大量的实例程序,手把手教读者学习如何通过编程实现应用接口电路、通信、控制等各种单片机功能。

本书适合用做大专院校电子类或计算机专业的教学用书,同时还可以作为从事该方面专业工作的科研参考用书。

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

基于 C 语言编程 MCS-51 单片机原理与应用 / 张培仁等编著. —北京:

清华大学出版社, 2002

ISBN 7-302-06177-7

I. 基… II. 张… III. ①单片微型计算机, MCS-51 ② C 语言-程序设计

IV. TP368.1

中国版本图书馆 CIP 数据核字 (2002) 第 105899 号

出版者: 清华大学出版社 (北京清华大学校内, 邮编 100084)

印刷者: 北京市耀华印刷有限公司

发行者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 26.5 字数: 644 千字

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数: 0001~5000

书 号: ISBN 7-302-06177-7

定 价: 38.00 元

前 言

应出版社要求，中国科学技术大学模式识别与智能控制教研室的老师和研究生共同编写了该书。自20世纪80年代初以来，本教研室一直从事嵌入式微处理器的研究和教学工作，有丰富的教学与科研经验。本书所有电路接口和绝大部分程序都在科研项目中使用过。

中国科学技术大学自动化系张培仁教授对本书进行了精心策划、分章，选定了本书内容，负责全书的审查校对。

本书第1、第2章单片机原理部分由张欣编写；第3章至第11章C语言编程的语法、数据结构、语句和函数分类、C51的使用以及C51和汇编语言混合编程应注意的问题由孙占辉编写。第12章至17章包括C51内部资源编程、输入输出接口、人机交互以及CAN模块的设计，由孙占辉、张村峰和房玉东共同编写。张培仁教授、杨兵、颜进军、凌来根、陈雪、樊杰和叶雪峰提供了本书各章的电路图和各種芯片控制程序。

本书编写过程中得到了教研室的马云老师和朱东杰老师的大力帮助，同时参加了部分章节的编写工作，在此一并表示感谢。

由于时间仓促，书中错误之处在所难免，敬请见谅！

张培仁

2002年10月15日

目 录

第1章 嵌入式控制器概述	1
1.1 嵌入式微控器的近来发展	1
1.2 单片机发展趋势	2
1.3 单片机内部结构	2
1.4 单片机与微处理器	3
1.5 单片机产品	4
1.6 单片机的应用	4
1.7 单片机的开发平台	5
第2章 MCS-51单片机的硬件结构和指令系统	8
2.1 MCS-51单片机的主要功能特点	8
2.2 MCS-51单片机内部结构分析	9
2.3 MCS-51单片机的引脚功能	11
2.3.1 时钟电路	12
2.3.2 控制信号	13
2.3.3 I/O端口	14
2.3.4 MCS-51单片机管脚的应用特性	19
2.4 MCS-51的存储器组织	19
2.4.1 MCS-51程序存储器	21
2.4.2 数据存储器	21
2.4.3 专用寄存器（SFR）	25
2.4.4 外部数据存储器	28
2.5 MCS-51 CPU时序	28
2.5.1 机器周期，状态，相位	28
2.5.2 典型指令的取指和执行时序	29
2.6 MCS-51低功耗运行方式	30
2.6.1 HMOS型单片机的掉电运行方式	30
2.6.2 CHMOS型单片机的掉电运行方式与待机方式	31
2.7 MCS-51内部程序存储器的写入、校验和加密	32
2.7.1 8751内部EPROM的写入和擦除	32
2.7.2 内部程序存储器加密位	33
2.8 MCS-51指令系统	33

第3章 C语言与8051单片机	41
3.1 8051的编程语言.....	41
3.2 C语言的特点与C语言向8051的移植.....	42
3.3 C51编译器.....	43
3.4 C51程序结构.....	44
第4章 C51基本词法	48
4.1 标识符.....	48
4.2 关键字.....	49
4.3 常量.....	49
4.4 字符串.....	50
4.5 运算符.....	50
4.5.1 单目运算符.....	52
4.5.2 双目运算符.....	54
4.5.3 三目运算符.....	59
4.5.4 赋值运算符.....	59
4.5.5 逗号运算符.....	60
4.5.6 函数参数运算符.....	60
4.5.7 数组下标运算符.....	60
4.5.8 结构/联合成员运算符.....	61
4.5.9 结构/联合指针运算符.....	61
4.6 分隔符.....	61
4.6.1 方括号对分隔符.....	61
4.6.2 圆括号对分隔符.....	62
4.6.3 花括号对分隔符.....	62
4.6.4 逗号分隔符.....	62
4.6.5 分号分隔符.....	63
4.6.6 冒号分隔符.....	63
4.6.7 省略号分隔符.....	64
4.6.8 星号分隔符.....	64
4.6.9 等号分隔符.....	64
4.6.10 ^分隔符.....	64
4.6.11 预处理器伪指令符号#.....	65
4.7 空白符.....	65
4.8 注释符.....	65
第5章 C51数据结构	67
5.1 数据与数据结构.....	67
5.2 常量.....	68
5.2.1 整数常量.....	69

5.2.2 浮点数常量.....	69
5.2.3 枚举常量.....	70
5.2.4 字符常量.....	71
5.3 变量.....	72
5.3.1 变量说明.....	72
5.3.2 变量的数据类型.....	75
5.3.3 8051特殊功能寄存器和特殊功能位.....	77
5.4 C51构造数据类型.....	78
5.4.1 数组.....	79
5.4.2 结构.....	82
5.4.3 共用体.....	86
5.4.4 枚举.....	87
5.5 指针.....	88
5.5.1 指针的基本概念.....	88
5.5.2 变量的指针和指向变量的指针变量.....	89
5.5.3 数组指针和指向数组的指针变量.....	93
5.5.4 指向结构类型数据的指针.....	97
5.5.5 Franklin C51的指针类型.....	100
第6章 C51的存储.....	103
6.1 8051单片机的存储器.....	103
6.2 C51存储类型与8051单片机存储器.....	105
6.3 存储模式.....	106
第7章 C51流程控制语句.....	109
7.1 C语言程序的基本结构.....	109
7.1.1 顺序结构及其流程图.....	109
7.1.2 选择结构及其流程图.....	109
7.1.3 循环结构及其流程图.....	111
7.2 选择结构流程控制语句.....	111
7.2.1 if 语句.....	112
7.2.2 switch/case语句.....	114
7.3 循环结构流程控制语句.....	115
7.3.1 goto语句及其构成的循环.....	116
7.3.2 while语句.....	117
7.3.3 do-while语句.....	118
7.3.4 for循环语句.....	119
7.3.5 循环的嵌套.....	122
7.3.6 break和continue语句.....	123

第8章 C51函数	125
8.1 C51程序的构成	125
8.2 函数的分类	126
8.2.1 从用户角度划分	126
8.2.2 从函数定义形式划分	126
8.3 函数的说明	127
8.3.1 函数说明的格式	127
8.3.2 函数说明	128
8.3.3 一般的函数定义举例	133
8.4 函数的参数和函数返回值	135
8.4.1 形式参数和实际参数	135
8.4.2 函数的返回值	136
8.5 函数的调用	138
8.5.1 函数调用的一般形式	138
8.5.2 函数调用的方式	138
8.5.3 对被调用函数的说明	138
8.5.4 函数的嵌套调用	140
8.5.5 函数的递归调用	141
8.5.6 用函数指针变量调用函数	142
8.6 数组、指针与函数调用	143
8.6.1 数组作为函数的参数	143
8.6.2 指针作为函数的参数	145
8.6.3 返回指针类型数据的函数	148
8.7 内部函数和外部函数	149
8.7.1 内部函数	149
8.7.2 外部函数	149
8.8 C51库函数介绍	150
8.8.1 intrins.h	150
8.2.2 absacc.h	152
第9章 C51预处理器	154
9.1 包含文件伪指令	154
9.2 伪指令宏	155
9.2.1 简单宏	155
9.2.2 参数宏	156
9.2.3 条件宏	156
9.2.4 预定义宏	157
9.2.5 宏释放	158
9.2.6 宏体中的#符和##符	158

9.3 条件编译伪指令	159
9.4 编译器伪指令	159
9.5 #error伪指令	160
第10章 C51和汇编语言	161
10.1 C51和汇编语言的性能比较	161
10.2 C51和汇编语言的混合编程	167
10.2.1 C51语言关于函数定义和函数调用的规定	168
10.2.2 C51与汇编函数互调代码	168
第11章 开发C51程序	173
11.1 C51程序基本概念	173
11.1.1 程序的组成	173
11.1.2 常用名词	173
11.1.3 C51文件扩展名	174
11.2 C51程序开发流程	175
11.3 C51编译	176
11.3.1 汇编方法	176
11.3.2 编译方法	181
11.4 C51库和连接/定位器	183
11.4.1 库	183
11.4.2 连接/定位器	184
11.5 编译连接与存储空间	188
11.5.1 覆盖	188
11.5.2 共享	188
11.6 程序优化	189
第12章 8051内部资源的C51编程	191
12.1 输入/输出端口	191
12.2 定时器/计数器	193
12.2.1 定时器/计数器的控制寄存器和状态寄存器	194
12.2.2 定时器/计数器的计数初值计算	197
12.2.3 应用编程实例	198
12.3 中断	202
12.3.1 8051中断系统	202
12.3.2 控制中断的一些寄存器	202
12.3.3 响应中断	204
12.3.4 中断服务程序的编制	205
12.3.5 中断编程应用实例	205
12.4 串行口	210

12.4.1	与串行口控制有关的寄存器	210
12.4.2	串行口工作模式	212
12.4.3	串行口应用实例	214
第13章	8051外部扩展资源的C51编程	223
13.1	可编程I/O接口控制芯片8255	223
13.1.1	8255简介	223
13.1.2	8255工作说明	225
13.1.3	8255与8051的接口和编程	230
13.2	用串行口扩展I/O端口	231
13.2.1	扩展输出口	231
13.2.2	扩展输入口	233
13.3	可编程外围定时器芯片8253	235
13.3.1	8253内部结构和引脚	235
13.3.2	8253的工作方式和控制字	237
13.3.3	8253与8051单片机的接口和编程	239
13.4	可编程接口芯片8155	243
13.4.1	8155芯片的结构和引脚	243
13.4.2	8155芯片的命令字、状态字	245
13.4.3	8155与8051的接口和编程	247
13.5	实时时钟/日历芯片PCF8583	249
13.5.1	I ² C总线	249
13.5.2	PCF8583时钟日历芯片	251
13.5.3	PCF8583的应用	257
第14章	C51的输入编程	263
14.1	开关量的输入	263
14.1.1	开关量输入要解决的问题	264
14.1.2	开关量输入实例	264
14.2	计数器的设计	266
14.3	用单线数字温度传感器DS18B20实现温度测量	268
14.3.1	DS18B20概述	269
14.3.2	DS18B20应用实例	273
14.4	AD7715数据采集	277
14.4.1	AD7715概述	277
14.4.2	AD7715应用实例	284
第15章	C51的输出编程	288
15.1	开关量的输出	288
15.1.1	开关量输出一一般结构	288

15.1.2 开关量输出实例	288
15.2 DA接口	290
15.2.1 MAX532概述	290
15.2.2 MAX532的应用	292
第16章 单片机人机交互C51程序设计	296
16.1 键盘程序设计	296
16.1.1 键盘的类型	296
16.1.2 键盘输入程序的设计方法	297
16.2 LCD液晶显示器	299
16.2.1 液晶显示器的结构和工作原理	299
16.2.2 LCD液晶显示器的驱动	299
16.2.3 LCD显示模块	299
第17章 用8051单片机实现CAN总线接口模块	310
17.1 开发背景	310
17.2 角控制器执行机构结构分析及模块设计思路	311
17.3 硬件设计	312
17.3.1 CAN总线接口部分	313
17.3.2 CPU、复位芯片、电源管理及地址译码	318
17.3.3 I/O接口部分	322
17.4 软件设计	326
17.4.1 CAN总线通信程序	326
17.4.2 看门狗芯片初始化程序	334
17.4.3 I/O管理函数	335
17.4.4 控制算法程序	336
17.4.5 模块程序清单	337
17.5 小结	352
附录A 具有51内核的单片机	353
A.1 MCS-51系列单片机	353
A.2 AT89系列单片机	353
A.3 DALLAS DS80C320单片机	354
A.4 WINBOND W78C31单片机	354
附录B C51库函数	355
附录C C51编译命令行控制选项和控制伪命令	363
C.1 简介	363
C.2 编译命令行	363
C.2.1 一次性使用的编译控制伪指令	364

C.2.2 可多次使用的编译控制伪指令	371
附录D C51编译器使用错误提示	375
D.1 致命错误	375
D.2 语法及语义错误	377
D.2.1 错误格式	377
D.2.2 错误信息及可能发生的原因	377
附录E L51连接定位器使用错误提示	389
E.1 前言	389
E.2 L51警告	389
E.3 L51错误	391
E.4 L51致命错误	393
E.5 异常信息	395
附录F C51的极限值	397
附录G QTH仿真开发系统使用指南	398
G.1 QTH仿真器系统设置	398
G.2 QTH-8052F系统设置	399
G.3 QTH调试器的主窗口	400
G.4 基本调试技巧	401
G.4.1 装入代码进行反汇编调试	401
G.4.2 使用源程序窗口调试程序	401
G.4.3 查看内存单元内容	406
G.5 源程序的编译及连接	408
G.5.1 源程序的编译及连接	408
G.5.2 创建工程文件	408

第1章 嵌入式控制器概述

1.1 嵌入式微控器的近来发展

嵌入式控制器主要分两类：嵌入式微控制器和嵌入式微处理器。

嵌入式微控制器（Embedded Microcontrollers）往往是在一硅片上集成CPU、A/D、D/A、PWM口、I²C、CAN网络口等。我国称为单片机系统。国内这种叫法主要是侧重于其功能。因为构成计算机的三要素：CPU、存储器、外设位于一个芯片之中，所以称为单片计算机。它追求的方向是整体化、小型化和廉价。

嵌入式微处理器（Embedded Microprocessors）与前者区别在于：后者是由原微处理器发展而来的，它不附加内部存储器。它依靠片外存储器通过三总线连接起来。指令系统也是原微处理器的指令系统，不需要重新学习。这类芯片有：80188，80186，8680，V20，V40等。数字信号处理DSP也属于后者。本书主要介绍嵌入式微控制器。

目前世界上集成电路生产厂商几乎都有自己的微控制器系列产品。不同厂商微控制器的指令不同，开发装置也不兼容。但是从近年来的发展来看，8051系列可能最终形成微控制器的工业标准。这是因为Intel公司向不同厂家转让8051微处理器的生产权。目前8051系列有百余种派生芯片。它们既保留8051核心结构又增加了各个厂商的专用功能或在原来功能基础上给予补充。这样，速度有所提高，芯片低功耗，封装尺寸又有很大改进，从而形成广泛的系列。同时，大部分用户已习惯使用它，市场上又向用户提供了软件包和硬件接口。

20世纪90年代以后逐步开发了16位微控制器80C251和80C51XA，与8051指令兼容。这样就可以重复使用和推广以前开发的程序库。

微控制器8051系列的C编译器经过多年使用已日趋成熟，可达到专业化水平的C编译器也被广大用户所认可。另外一个原因是，只有8051系列被多家公司成功研制出实时多任务操作系统（RTOS），RTOS对用户有效管理应用程序的运行、合理分配CPU时间和资源都带来了极大的方便，为用户开发复杂应用系统、实现软件产业化带来了好处。开发8051系列的公司很多，目前芯片价格很低，因此可选择性也就最强。

16位微控制器MCS-8096在20世纪80年代中期制造出来，世界著名微控制器生产商Intel和PHILIPS分别于20世纪90年代初推出16位机80C251和80C51XA两个系列。Intel公司又把原来8096进一步改造，产生了80C196/296的新产品。SIEMENS公司生产了RISC系列的SAB167和面向工业控制的C166系列。

嵌入式微控制器在处理速度、寻址能力、兼容性、支持多任务等方面都有很大进步。在速度上已达到20MHz以上。这主要是由于在结构上采用流水线设计。一般可以同时执行几条指令，单周期执行也缩短到100ns，中断响应不超过400ns，有的CPU还增加了DSP处理

器。在寻址能力上，各公司都达到24位线性空间寻址。寻址方式支持间接、扩展和相对位寻址。微控制器一般都提供两个地址空间，即程序空间和数据空间。芯片中资源越来越丰富，有Flash ROM、RAM、A/D、PWM、I²C、CAN等专用功能模块。OTP EPROM也得到了广泛使用。RAM空间可达1K到几K。有的芯片有两个双工UART。芯片的电源管理越来越合理，功耗越来越低。Flash ROM一般写1000次没有什么问题。工作电压具有灵活性，一般2.7V~5V都可以工作。为了适应移动通信相关的低电压，芯片开始采用电源智能化管理技术。

微控制器性能大幅度的提高，也带来了单片机开发水平的提高。

1.2 单片机发展趋势

单片机正朝着多层次用户、多品种、多规格、高性能的方向发展。

1. 高档单片机性能不断提高

- **CPU能力加强** CPU能力加强主要体现在数据处理速度和精度提高方面。一般通过以下措施来实现：增加CPU的字长，扩充硬件，提高主频，提高总线速度，扩充指令系统和提高效率。
- **内部资源增加** 单片机的内部资源除了CPU以外，还包括各种类型的存储器和I/O端口。程序存储器包括：掩膜式ROM、EPROM、E²PROM或FLASH。容量最大可达到几十KB。RAM也可达到几KB。I/O端口包括并行、串行、定时器/计时器并配有A/D、D/A、PWM、LED、LCD驱动接口等。
- **寻址范围增加** 目前最高可寻址几十MB。

2. 超小型，低功耗，廉价

- 微巨型单片机，目前已推出了运算速度1.2亿次/秒，CPU字长32位，可运行64位浮点运算的单片机。
- 指令系统从复杂指令系统向简易指令系统过渡。
- 单片机开发系统向多用户、C编译、在线实时开发方向发展。

1.3 单片机内部结构

单片机内部含有计算机的基本功能部件。

1. 中央处理机

中央处理机（CPU）是单片机的核心部件，一般按字长分为4位机、8位机、16位机、32位机等。CPU也可以根据处理速度、中断和实时控制功能而有不同指标。

2. 存储器

按Intel公司的设计方式分为只读（程序）存储器（ROM）和随机访问（数据）存储器（RAM）。一般单片机内部ROM在64KB以内。数据存储器RAM一般几KB以内。

3. I/O 端口和特殊功能部件

一般单片机有1~2个异步串行通信口，有多个并行口，有1~3个定时器/计时器，还有数量不等的A/D、D/A、PWM、DMA HSIO。

4. 单片机应用系统

单片机基本系统包括1片含有ROM和RAM、CPU和I/O端口的单片机，再加上外围复位电路、供电电源、输入输出设备及时钟电路。

单片机扩展系统除基本系统以外，还包括片外EPROM、片外RAM、串行口电平转换电路、并行口功率驱动电路，以及A/D、D/A等芯片。

1.4 单片机与微处理器

微处理器一般分为最大工作模式和最小工作模式。最大工作模式微处理器通过总线控制器、中断控制器来控制地址总线、数据总线和CPU控制总线，从而与片外存储器（芯片或硬盘）和I/O设备交换数据。微处理器的最小工作模式是直接控制地址总线、数据总线和控制总线。前者的控制方法要复杂些，后者的控制方法要简单些。但所有三总线都在微处理器的片外。

单片机基本系统的三总线都在芯片内运行。单片机扩展系统和微处理器最小工作模式控制方法是相似的。但是在单片机扩展系统中，数据总线和地址总线低位常常是分时复用的。

由于微处理器一般只和CPU、指令译码器与一些缓冲器、寄存器密切相关，所以它的主频速度比单片机高，处理速度快。但是组成系统结构复杂。一般微处理器都把程序存放在片外的程序存储器中，把数据存放在片外数据存储器中。单片机系统一般把CPU、ROM、RAM和I/O集成在一个芯片中，所以系统要简单得多。

微处理器的开发平台比较完善，一般可以用相应的PC机开发。使用高级语言开发，微处理器能很好地支持多用户操作系统。单片机系统由于EPROM不大，一般8位机、16位机只支持单用户系统（有的16位机支持多用户系统）。32位机才能很好支持多用户系统。单片机一般用汇编语言、C语言共同开发，并使用相应的专用开发系统。

微处理器目前大部分已使用简易指令系统了。单片机有的使用复杂指令系统，有的使用简易指令系统。

微处理器一般已使用哈佛流水结构，指令执行周期比单片机要少，执行一条加法指令的时间一般是几十ns。

1.5 单片机产品

Intel公司早在1976年就推出8位单片机8048, 20世纪80年代又推出了8051系列的单片机。目前MCS-51单片机结构已成为一种世界最通用的单片机, 它的结构和指令系统已转让给世界上百家芯片公司。

MCS-51系列产品有: 8051AH, 8051BH, 8751AH, 8052系列, 80C51, 87C51系列。它们的主要区别在于: 程序存储器大小、型号不同; RAM大小、定时器多少、I/O多少、主频的速度不同以及其他一些特殊功能不同。但是基本结构和指令系统是相同和兼容的。

与Intel公司MCS-51系列基本相同和兼容的有: PHILIPS公司生产的8031AH、80C31AH、80C451、80C550、80C851、80C552、80C652、80C528; ATMEL公司生产的89C51、89C52、89C55系列芯片; ADI公司生产的ADUC812、ADUC816、ADUC824。

不论Intel公司还是PHILIPS公司、ATMEL公司或ADI公司, 他们一开始都是用总线方式设计CPU来进行数据传递的。但是Motorola公司在开始设计单片机时, 主要使用组合逻辑的方法设计CPU中数据流的传递。因此Motorola公司的单片机68HC05、68HC705系列与MCS-51系列有一些不同: 对于双字节数据, 是低位地址还是高位地址在前; I/O设备的读写控制信号形式; 单片机在串行通信时, 使用SPI(发送线、接收线、时钟线)三线制还是使用Intel的两线制SCI(发送线、接收线); 串行数据是低位数据先输出还是高位数据先输出; 以及采用何种指令系统等等。由于Motorola公司主要使用组合逻辑设计芯片, 使芯片在同样工艺条件下完成相同功能所需晶体管数目比使用总线结构设计芯片要少, 这样芯片合格率提高, 硬件成本降低。所以Motorola公司的单片机首先在通信领域获得推广。但是由于组合逻辑实现模块化比较困难, 逻辑电路修改工作量加大。而总线结构的设计正好相反, 虽然硬件结构会多使用一些晶体管, 但各部分设计容易模块化, 易修改, 更新快, 从而使Intel公司逐步占领CPU市场。当然, 经过二十多年发展, 芯片设计相互取长补短, 现在已难以说明哪种芯片是用组合逻辑方法设计的, 哪种芯片是用总线方法设计的。但是, 在我们用MCS-51单片机的CPU与Motorola公司设计的I/O芯片接口时, 要特别注意其外特性的不同点。在使用Motorola公司提供的一些参考程序时也要特别注意这些区别。

1.6 单片机的应用

与一般PC机相比, 单片机具有下面一些特点:

- 体积小, 单片机“小而全”。
- 可靠性高, 三总线在芯片内部, 不易受干扰。
- 控制功能强, 控制直接, 速度也快。
- 使用方便, 性能价格比高。
- 容易产品化。比较容易保护开发者的知识产权。

单片机也有不如PC机的地方：速度比PC机慢，开发系统专用，存储容量比较小。因此，单片机广泛应用于以下场所：各种智能仪表、机电一体化设备，实时要求高的控制系统底层，另外在分布式多机控制系统、现场总线控制系统以及各种小型、移动控制系统中也常常使用单片机。

1.7 单片机的开发平台

单片机自20世纪70年代末问世以来，其应用得到迅速发展。以往开发单片机使用的基本工具包括仿真器、汇编器、编程器等。20世纪80年代的8051系列、6805系列、8096/98系列其内部资源基本在64K内，而大多数应用程序的代码长度都在十几K。进入20世纪90年代，随着大规模集成电路技术的高速发展，世界各主要半导体公司推出了面向21世纪的微控制器，如INTEL296、251、PHILIPSP51XA、MOTOROLA68300系列等。这些芯片采用流水线结构和宽总线（16条数据总线或2个内部总线），寻址能力可达6MB~32MB，片内ROM/EPROM/FLASH ROM可达28MB~512KB。为了充分有效地开发这些微控制器的内部资源，提高应用系统的性能指标，单片机开发平台技术也有很大提高。

由实时多任务操作系统（RTOS）、基于嵌入式系统的C语言编译器（COMPILER）、实时在线仿真器（ICE）组成的高性能工具集合称为单片机开发平台。单片机开发平台使单片机开发走向标准化、产业化，降低了风险，提高了效率。以往要进行单片机的开发工作，对开发者有多方面的要求：既要精通软件、硬件，又要具有系统设计概念，还要具备一定的电子线路工艺方面的知识和经验。开发单片机的另一大问题是：由于基于汇编语言编程，开发只能孤军作战，设计出来的代码可读性差、可继承性差。基于汇编的开发是一种个体手工生产方式，这种落后工具对于比较简单的系统还可以对付，但面对21世纪的高性能微控制器，将会遇到很大的困难。采用单片机开发平台，简单地说，用RTOS替代了单片机应用程序中的主程序，可以使多个程序员并行工作，写出相对独立的应用任务，也可以在RTOS的基础上购买专家库函数，组成高效、可靠、有竞争力的嵌入式系统。在开发方式上也产生了根本变化，将系统分析员、系统设计员与程序员分离。系统分析员定义应用系统的功能要求，系统设计员设计电路硬件并在RTOS基础上定义软件功能模块。程序员只要有C语言的基础，就可以在开发平台上完成复杂的应用系统。这样对程序员的要求降低，分工更加明确。一个复杂系统可以由多人共同开发，使嵌入式软件程序的开发进入规模化和产业化。高性能的实时在线ICE具备实时映像存储器（RAM）和海量数据等跟踪功能，对于复杂程序调试是必不可少的。

总之，嵌入式系统工业是一种技术密集、资金密集、而又高度分散的工业，它不可能像PC软件产业那样垄断在少数大公司手中，这就给广大单片机开发者带来了巨大的机会和广阔的发展前景。

1. 实时多任务操作系统

目前嵌入式应用的一个发展方向是采用实时多任务操作系统（Real Time Operating