

图书基本信息

书名：<<51单片机实用C语言程序设计与典型实例>>

13位ISBN编号：9787508387949

10位ISBN编号：7508387945

出版时间：2009-7

出版时间：中国电力出版社

作者：杨国田，白焰，董玲 编著

页数：232

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

1980年,英特尔公司推出了后来广为普及的MCS - 51单片机,近30年来,其衍生系列不断涌现,从Atmel加入FLASHROM,到Phillips加入各种外设,再到后来的Cygnal推出C8051F,使得以8051为核心的单片机在各个发展阶段的低端应用中始终扮演着一个独特的角色,其地位不断提升,资源日渐丰富,历经30年仍在生机勃勃地延续,甚至在SoC时代仍赫然占有一席之地。

有资料显示,8位机始终是嵌入式低端应用的主要机型,而且在未来相当长的时间里仍会保持这个势头。

而8051系列,在8位单片机中形成了一道独特的风景线。

历史最长、常盛不衰、众星捧月、不断更新,形成了既具有经典性又不乏生命力的一个单片机系列。

可以说,Intel公司创建了8位机的经典系列结构。

究其原因,嵌入式系统嵌入到对象体系中,并在对象环境下运行,与对象领域相关的操作主要是对外界物理参数进行采集、处理,对外界对象实现控制,并与操作者进行人机交互等,而这些任务所要求的响应速度有限,且不会随时间变化。

在8位单片机能基本满足其响应速度要求后,数据宽度不成为技术发展的主要矛盾。

因此,8位单片机将稳定下来,其技术发展方向转为最大限度地满足对象的采集、控制、可靠性和低功耗等品质要求。

而对于8051系列,则由于其实施技术开放政策,使得这个系列历经沧桑而不老。

在相当长的历史时期里,单片机的开发以汇编语言为主,即使今天,高效的工作程序也大多不能完全摆脱汇编语言,这当然有其原因。

而且Borland公司创始人之一,因使用汇编语言撰写编译器开发出TurboPascal(首创In - MemorYCompiler)而震惊全世界的大名鼎鼎的AndersHeilsberg先生用他的传奇经历也证明了汇编语言可以做大事。

Anders先生使用汇编语言创造出了一度是全世界速度最快、品质一流的Pascal编译器,可见汇编语言的威力。

但是,在Anders离开了Borland之后,几乎没有人能够修改Anders的编译器,足见Anders先生的汇编语言功力何以出神入化,并非常人所能企及。

然而在今天,单片机应用系统开发开始走向大型化、复杂化,开发周期要求越来越短,汇编语言被认为枯燥而难以维护和调试,而C语言成了人们热捧的开发语言,尤其近年来C语言几乎是大学新生的必修课,这虽然没有直接降低单片机系统开发的门槛,但是却抬高了门槛外的土地。

当然,C语言是一种优秀的高级语言,它精练,接近硬件,复杂程度适当,数据类型丰富、明确。

## 内容概要

本书主要介绍8051系列单片机的c语言(C51)程序设计方法、应用系统及程序设计实例，共分为8051系列单片机基础、c语言基础、应用系统设计、设计案例等四个部分。

其中第三部分以实例方式介绍应用系统设计的有关问题。

包括人机界面设计以及常用接口器件的使用等，例如按钮、矩阵键盘、触摸屏、LCD显示器、串行A / D(D / A)器件、定时器、RS-232接口等；第四部分主要介绍多个应用系统设计案例，包括红外遥控系统、射频遥控系统、电视遥控器、步进电机驱动系统、PS2 / 串口鼠标接口系统等。

本书写作时注意展示C51与汇编语言的内在联系，这样有利于对C51的自如运用，同时在案例部分给出系统设计时的种种考虑，力图使读者具有身临其境的感觉。

此外，还详细介绍有关元器件参数计算的方法等，例如总线上拉电阻的阻值、电阻的额定功率核算等

。本书赠送1张光盘，内含书中涉及到的所有程序以及相关的素材文件。

本书可作为从事单片机及嵌入式开发的技术人员、51单片机开发者和初学者的学习参考书，也可作为高等院校相关专业研究生和高年级本科生、大学教师等教材。

## 书籍目录

前言第1章 51单片机基础 1.1 51系列MCU的结构组成 1.2 特殊功能寄存器 1.3 存储器系统组成及特点 1.4 并行I/O接口 1.5 定时器,计数器组成及特点 1.6 中断子系统 1.6.1 中断请求 1.6.2 中断响应过程 1.7 串行I/O接口(UART) 1.7.1 UART的工作模式 1.7.2 波特率设置 1.8 系统扩展 1.9 51系列单片机的指令系统 1.9.1 程序设计模型 1.9.2 寻址方式 1.9.3 指令分类 1.9.4 指令列表第2章 C51基础 2.1 C语言程序的基本结构 2.1.1 C语言的字符集 2.1.2 C语言词汇 2.2 数据类型和表达式 2.2.1 常量和变量 2.2.2 变量的类型 2.2.3 数组和指针 2.2.4 字符及字符串的表述 2.2.5 结构、联合和位段 2.2.6 自定义数据类型 2.3 基本运算 2.3.1 运算符的结合性与优先级 2.3.2 算术运算符 2.3.3 递变运算符 2.3.4 关系与逻辑运算符 2.3.5 位运算符 2.3.6 赋值运算符 2.3.7 逗号运算及其表达式 2.3.8 条件表达式 2.3.9 几个特殊运算符的说明 2.4 数据输入,输出及实现 2.4.1 printf( )函数 2.4.2 scanf函数 2.4.3 头文件 2.4.4 输入/输出与串行接口 2.5 基本语句 2.5.1 基本语句分类 2.5.2 流控制语句 2.6 函数 2.6.1 函数的定义与声明 2.6.2 函数的参数和返回值 2.6.3 函数的调用 2.6.4 main( )函数 2.6.5 函数中的变量 2.6.6 变量的存储类别 2.7 C语言预处理 2.7.1 宏定义 2.7.2 文件包含 2.7.3 条件编译 2.7.4 预处理命令和预定义宏名汇总表 2.8 C语言程序设计基础 2.8.1 算法及其描述 2.8.2 基本程序结构 2.9 C51对ANSI C的扩展与简化 2.9.1 内存区域 2.9.2 存储类型与存储模式 2.9.3 变量或数据类型 2.9.4 Keil C51指针 2.9.5 Keil C51函数 2.10 Keil C51程序设计 2.10.1 绝对地址访问 2.10.2 Keil C51与汇编的接口 2.10.3 Keil C51软件包中的通用文件 2.11 Keil C51库函数参考 2.11.1 内建库函数与普通库函数.....第3章 应用系统设计第4章 应用系统设计案例参考文献后记

## 章节摘录

插图：变量从作用域角度分为全局变量和局部变量，有利于程序的模块化和函数之间的相对独立性。从技术实现的角度来看，C语言引入了静态存储和动态存储两种方式，即让变量具有不同的生存期，对于实现全局和局部变量非常有利。

(1) 静态存储方式：变量在整个程序运行期间占有固定的内存地址。

(2) 动态存储方式：程序运行期间根据需要临时为变量分配内存。

在计算机中，整个内存除掉操作系统等系统程序占用的空间，剩余的空间均可由用户程序使用，称为用户存储空间，这部分空间又可细分为三个部分：(1) 程序区：用于存储用户程序，通常由操作系统加载，在单片机中则是事先固化的。

(2) 静态存储区：程序加载的同时即已分配的内存区，在单片机中则是程序编译后即已明确分配方案的数据存储区。

(3) 动态存储区：程序运行时可以随时申请或释放的内存区。

全局变量全部存放在静态存储区，程序开始执行时为其分配内存，程序结束时释放。

但在单片机中则是编译后确定内存地址，一般也不存在释放问题。

在整个程序运行期间，它们占据的存储单元固定不变，无需动态地分配和释放。

但是不能说静态存储区存放的就一定是全局变量。

极个别函数，因位其运行结果与历史有关，需要一种局部、但能长期保存的变量，这种局部变量就需要放到静态存储区。

而动态存储区（通常是堆栈）则存放以下数据：(1) 函数的形参。

(2) 自动变量（未加static声明的局部变量）。

(3) 函数调用时的现场保护和返回地址。

上述三种数据，在函数开始调用时分配（除自动变量外，其余在主调函数中即已分配）动态存储空间，函数结束时释放（除自动变量外，其余在回到主调函数后才彻底释放）这些空间。

在C语言中，每个变量和函数有两个属性：数据类型和数据的存储类别。两者全部说明后才完整地声明了一个变量，如果不明确说明，则系统自动取缺省值，前面的许多程序实例就是如此。

## 后记

作者1988年接触单片机，从依托.rP801B平台制作单片机仿真器开始，一直从事单片机、嵌入式系统研发工作，期间熟悉和使用过多种处理器的汇编语言，例如6502、8098、8048、8051、8086等同时，也使用过多种高级语言，例如C语言、Visual Basic、Visual Foxpro数据库系统、Visual C++等，开发软硬件产品数十种。

至今仍在不断学习，因为嵌入式系统研发是一个日新月异的世界。

作者在本书写作过程中大量参考了来自互联网的资料，但因为网站的资料多互相转载，很难搞清楚原始出处，因此，未在参考文献中列出，在此一并表示感谢，敬请谅解。

编辑推荐

《51单片机实用C语言程序设计与典型实例》由中国电力出版社出版。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>