



普通高校“十三五”规划教材

Eastsoft.

东软载波单片机应用 C 程序设计

上海东软载波微电子有限公司 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

“十三五”规划教材

Eastsoft.

东软载波单片机应用 C 程序设计

上海东软载波微电子有限公司 编著



北京航空航天大学出版社

内 容 简 介

从 C 语言设计及实现的角度,为读者深入解析 C 语言的各种细节。本书以《C 语言参考手册》为基础,既力求精确描述,又不乏通俗解释,并配有详细例程,期待成为国内第一本面向 C 程序员的语言参考手册。本书涉及领域较广,包括计算机系统结构、程序语言概念、程序设计基础、软件工程、嵌入式开发等。同时,本书又是一本专门针对 C 语言深度剖析的参考手册,不但囊括了 C 语言的传统知识点,还深入阐述了类型系统、寻常类型转换规则、声明形式等高级主题。

本书既合适作为高等院校学生、广大 C 程序开发人员的参考手册,也可以帮助从事 C 编译器设计人员更好地理解 C 语言标准。

图书在版编目(CIP)数据

东软载波单片机应用 C 程序设计 / 上海东软载波微电子有限公司编著. -- 北京: 北京航空航天大学出版社, 2017.6

ISBN 978-7-5124-2446-3

I. ①东… II. ①上… III. ①单片微型计算机—C 语言—程序设计 IV. ①TP368.1②TP312.8

中国版本图书馆 CIP 数据核字(2017)第 133408 号

版权所有,侵权必究。

东软载波单片机应用 C 程序设计

上海东软载波微电子有限公司 编著

责任编辑 胡晓柏 张楠

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:26 字数:554 千字

2017 年 9 月第 1 版 2017 年 9 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2446-3 定价:65.00 元

前言

本书的写作目的源于上海东软载波微电子有限公司 HRCC 编译器设计团队的实践工作。HRCC 编译器是一款面向嵌入式应用的 C 编译器,她以 C89 标准为基础结合 HR 系列单片机特性设计而成。自 2010 年 6 月 HRCC 编译器正式发布以来,有数百家嵌入式系统方案公司及高等院校实验室正在使用其开发各类系统方案,是真正意义上完全独立自主设计的商业级 C 编译器。在编译器开发与支持工作中,发现不少拥有多年 C 程序开发经验的工程师其实并没有真正了解 C 语言。面对一些“奇怪”的语法问题,他们经常使用自认为最有效的方式——回避。我们曾经很多次拿着《C 语言参考手册》向用户深入解释错误原因,但显然这种方式过于低效。为此,我们经过大量的经验总结,决定编写 C 语言书籍,帮助客户深入、精确理解 C 语言。

尽管 C 语言已经问世 40 多年,但关于 C 语言精确描述的书籍却非常稀缺。目前,堪称最精确描述 C 语言的书籍莫过于《C 语言参考手册》,其在每位 C 编译器设计者心目中的地位是无可比拟的。不过,遗憾的是绝大多数 C 程序开发人员并非该书的预期读者。正如该书前言所述:We expect our readers to already understand basic programming concepts, and many will be experienced C programmers。作为 HRCC 编译器设计团队,我们每位研发人员都曾无数次拜读《C 语言参考手册》及 C89/C99 标准,深知其“晦涩”并非普通开发人员可以接受,面对每个细节都必须反复推敲,否则稍有不慎就可能引起误解。而市面上大部分的“经典”C 语言教材可能只涉及了其中 30%左右的语法点,还不乏一些错误观点存在。为此,我们团队期待结合对 C 语言的理解及多年编译器设计经验,从程序语言设计及实现的角度,深入浅出为读者解析那些 C 语言的“细枝末节”。

本书涉及领域较广,包括计算机系统结构、程序语言概念、程序设计基础、软件开发技术、嵌入式应用以及程序调试技术等,期待为读者展示软件开发的完整过程。在 C 语言知识方面,本书不但讲述了表达式、语句、函数、数组、指针等基本语言点,还深入描述了类型系统、寻常类型转换规则、声明形式等市面上大部分教材几乎从未涉足的话题。

本书所使用的软件开发环境、程序源代码都可以从上海东软载波微电子有限公司官方网站(www.essemi.com)免费获得。由于时间仓促,笔者水平有限,书中难免有不足和疏漏之处,欢迎广大读者批评和指正。作者联系方式:

E-mail: c_program@essemi.com

最后,特别感谢上海东软载波微电子有限公司软件部卢昊参与全书编写工作,潘松、陈立权、陈光胜校审全稿,并提出宝贵意见。

上海东软载波微电子有限公司软件部

目 录

第 1 章 计算机系统	1
1.1 计算机结构简介	1
1.1.1 冯·诺依曼结构	1
1.1.2 哈佛结构	2
1.2 处理器内核	3
1.2.1 指令系统	4
1.2.2 乘法器和除法器	5
1.2.3 程序计数器	6
1.2.4 堆 栈	6
1.3 程序存储器	7
1.3.1 直接修改寄存器	7
1.3.2 跳转指令	8
1.3.3 FLASH 自编程	9
1.4 数据存储器	12
1.4.1 直接寻址	13
1.4.2 间接寻址	14
1.4.3 通用存储器特殊寻址	16
第 2 章 C 语言基础	17
2.1 命令式语言	17
2.2 程序基本结构	19
2.2.1 注 释	20
2.2.2 预处理	20
2.2.3 函 数	20
2.2.4 变 量	21
2.2.5 表达式与控制结构	21
2.3 C 语言标准	22
2.3.1 传统 C	22

2.3.2	C89 标准	22
2.3.3	C95 标准	23
2.3.4	C99 标准	23
2.3.5	C1x 标准	24
2.3.6	GB 标准	25
2.4	语言的语法	25
2.4.1	文 法	25
2.4.2	文法推导	27
2.5	本书约定	28
第 3 章	表达式	29
3.1	变 量	29
3.1.1	类 型	29
3.1.2	变量声明	33
3.1.3	变量初始化	34
3.2	标识符	35
3.3	常 量	36
3.3.1	整数常量	37
3.3.2	浮点数常量	37
3.3.3	字符常量	39
3.3.4	字符串常量	40
3.4	表达式	40
3.4.1	运算符的优先级	41
3.4.2	算术表达式	42
3.4.3	赋值表达式	44
3.4.4	逗号表达式	47
3.4.5	关系表达式	47
3.4.6	判等表达式	48
3.4.7	逻辑表达式	48
3.4.8	条件表达式	49
3.4.9	按位表达式	50
3.4.10	移位表达式	51
3.4.11	增值、减值运算	52
3.4.12	类型转换表达式	53
3.5	sizeof 表达式	54
3.6	类型转换:表示形式转换 *	54

3.6.1	表示形式改变	55
3.6.2	整型之间转换	55
3.6.3	其他标量类型转换为整型	56
3.6.4	转换为浮点类型	56
3.6.5	转换为结构/联合类型	57
3.6.6	转换为枚举类型	57
3.6.7	转换为数组/函数类型	57
3.6.8	转换为指针	57
3.7	类型转换:寻常转换 *	58
第4章	语 句	64
4.1	表达式语句	64
4.2	复合语句	65
4.2.1	作用域	65
4.2.2	可见性	67
4.2.3	重复可见性	67
4.3	选择语句	68
4.3.1	if 语句	68
4.3.2	switch 语句	72
4.4	循环语句	76
4.4.1	while 语句	76
4.4.2	do 语句	77
4.4.3	for 语句	78
4.4.4	程序实例	79
4.5	无条件转移语句	85
4.5.1	break 与 continue 语句	85
4.5.2	goto 语句	87
4.6	名字重载 *	89
第5章	数 组	91
5.1	数组概述	91
5.2	一维数组	92
5.2.1	一维数组声明	92
5.2.2	数组下标	93
5.2.3	程序实例	93
5.3	多维数组	96

目 录

5.3.1	二维数组声明	97
5.3.2	程序实例	97
5.3.3	多维数组声明	99
5.4	数组初始化	100
5.4.1	一维数组初始化	100
5.4.2	二维数组初始化	100
5.4.3	多维数组初始化	102
5.5	数组存储	102
5.5.1	存储映射	102
5.5.2	选择器	103
5.5.3	边界检查	104
5.6	排序与查找 *	105
5.6.1	排 序	105
5.6.2	查 找	109
第 6 章	结 构	113
6.1	结构概述	113
6.1.1	引 例	113
6.1.2	结构标签	114
6.1.3	结构初始化	116
6.1.4	结构成员选择	116
6.2	结构存储	117
6.2.1	存储布局	118
6.2.2	内存对齐 *	118
6.3	结构赋值	121
6.4	程序实例	122
6.5	位 域	128
6.5.1	位域概述	128
6.5.2	位域声明	130
6.5.3	位域布局	131
6.5.4	匿名位域	132
6.6	结构声明 *	134
第 7 章	联合与枚举	137
7.1	联 合	137
7.1.1	联合概述	137

7.1.2 联合存储	140
7.1.3 联合赋值	141
7.1.4 联合应用	142
7.2 枚 举	147
7.2.1 枚举声明	148
7.2.2 枚举运算	150
第8章 指 针	152
8.1 指针概述	152
8.1.1 指针概念	152
8.1.2 指针声明	153
8.1.3 基本运算	153
8.1.4 通用指针	155
8.2 指针与数组	155
8.2.1 现象与本质	156
8.2.2 指针的算术运算	156
8.2.3 指针的其他运算	158
8.2.4 指针访问数组	161
8.2.5 程序实例	165
8.3 指针与结构	171
8.4 动态存储分配	174
8.4.1 运行时环境	174
8.4.2 动态分配与释放	175
8.4.3 堆区管理	176
8.4.4 悬空指针	178
8.4.5 内存泄漏	179
8.5 字符串	180
8.5.1 字符串存储	181
8.5.2 字符串变量	182
8.5.3 字符串的基本操作	182
第9章 函 数	186
9.1 引 例	186
9.2 函数定义	188
9.2.1 定义位置	188
9.2.2 基本形式	188

9.2.3	程序实例	189
9.3	参 数	193
9.3.1	参数声明	193
9.3.2	参数传递	195
9.3.3	实参求值顺序	197
9.3.4	程序实例	198
9.4	返回值	202
9.4.1	return 语句	203
9.4.2	函数返回类型	203
9.5	函数原型	204
9.5.1	函数原型基础	204
9.5.2	函数原型兼容	205
9.5.3	函数原型一致性	205
9.6	函数指针 *	207
9.6.1	基本形式	207
9.6.2	函数指针的运算	208
6	第 10 章 预处理	210
10.1	预处理概述	210
10.1.1	预处理过程	210
10.1.2	预处理命令	211
10.1.3	预处理词法	212
10.2	宏定义与替换	215
10.2.1	类似对象的宏	215
10.2.2	类似函数的宏	216
10.2.3	宏与函数的差异	217
10.2.4	取消宏定义	219
10.2.5	预定义宏	219
10.2.6	重新扫描	221
10.2.7	递归展开	221
10.3	条件编译	222
10.3.1	# if... # endif 结构	222
10.3.2	# if... # else... # endif 结构	223
10.3.3	# elif 命令	224
10.3.4	defined 运算符	225
10.3.5	# ifdef、# ifndef 命令	225

10.4 文件包含	226
10.4.1 一般形式	226
10.4.2 搜索路径	226
10.4.3 搜索路径设置	227
10.5 特殊命令	228
10.5.1 ##运算符	229
10.5.2 #运算符	231
10.5.3 #line 命令	231
10.5.4 #error 命令	232
10.6 程序实例	233
第 11 章 声 明	241
11.1 存储类别	241
11.1.1 变量存储类别	241
11.1.2 函数存储类别	244
11.2 类型限定符	245
11.2.1 const 限定符	245
11.2.2 volatile 限定符	247
11.2.3 扩展限定符	248
11.3 声明模型 *	250
11.3.1 声明前部	250
11.3.2 声明器	252
11.3.3 声明模型	254
11.3.4 类型别名	258
11.4 小 结	261
11.5 程序实例 *	261
11.5.1 动态规划	261
11.5.2 实例:最长公共子序列	262
11.5.3 实例:构造最大因式	265
第 12 章 软件项目开发	269
12.1 模块设计	269
12.1.1 编译与链接	270
12.1.2 模块设计	273
12.1.3 耦合度	277
12.1.4 内聚度	279

12.1.5	小 结	280
12.2	模块实现	281
12.2.1	模块接口	281
12.2.2	混合语言编程	284
12.2.3	信息隐藏	288
12.2.4	程序实例	289
12.3	代码规范	301
12.3.1	整体布局	301
12.3.2	注 释	302
12.3.3	命 名	303
第 13 章	嵌入式程序设计	305
13.1	中断服务	305
13.1.1	中断源	305
13.1.2	中断向量	306
13.1.3	中断优先级	308
13.1.4	中断处理	308
13.1.5	中断服务的实现	309
13.2	输入与输出	310
13.2.1	输入电压	311
13.2.2	端口方向设定	312
13.2.3	按键检测	312
13.2.4	程序实例	312
13.3	定时器	316
13.3.1	工作原理	316
13.3.2	程序实例	317
13.4	模/数转换	321
13.4.1	工作原理	321
13.4.2	程序实例	322
13.5	异步串行通信	328
13.5.1	通用异步收发器	328
13.5.2	常用的异步通信协议	330
13.5.3	程序实例	331
第 14 章	集成开发环境	335
14.1	iDesigner 概述	335

14.1.1	iDesigner 安装	335
14.1.2	菜单与工具	336
14.1.3	解决方案与项目	337
14.2	编码辅助功能	338
14.2.1	成员信息	338
14.2.2	函数信息	339
14.2.3	定义与引用	339
14.2.4	智能显示	340
14.2.5	大纲显示	342
14.3	编译生成	342
14.3.1	生 成	342
14.3.2	清 理	342
14.3.3	重新生成	343
14.3.4	生成结果	343
14.4	调 试	343
14.4.1	配置调试环境	343
14.4.2	执行控制	344
14.4.3	调试窗口	345
14.5	编 程	346
第 15 章	iDesigner 应用实例	348
15.1	生成项目	348
15.1.1	创建项目	348
15.1.2	编译项目	350
15.1.3	优化选项	351
15.2	调试项目	352
15.2.1	启动调试	352
15.2.2	变量监视功能	353
15.2.3	浮动监视	353
15.2.4	内存窗口	354
15.2.5	PC 断点	355
15.2.6	禁用与删除断点	356
15.2.7	条件断点	356
15.2.8	调用堆栈	357
15.2.9	反汇编	357
15.2.10	调试反汇编	358

15.2.11 跑表窗口	359
15.3 编程工具	359
附录 A 语言文法	363
附录 B ASCII 字符集	376
附录 C HR 系列单片机指令集	377
参考文献	401

第 1 章

计算机系统

计算机系统堪称是 20 世纪人类创造的最复杂的系统之一。随着个人微型计算机的诞生,计算机已经从实验室走向了人们的日常生活。除了最为人所熟知的微型个人计算机(即 personal computer,简称 PC)之外,那些体积小巧、性能各异的嵌入式计算机更是浸润在我们生活的每个角落。

关于计算机系统结构(computer architecture)的概念,最早由 IBM 计算机帝国的缔造者 G.M.Amdahl 于 1964 年提出的,其定义为:程序员所看到的计算机系统的属性,即概念性结构和功能特性。**概念性结构**主要指计算机各基本部件(如微处理器、存储设备、IO 外设等)的逻辑组织结构,但并不涉及这些部件的硬件设计与实现。而**功能特性**主要是指计算机的指令系统及其执行模型,包括数据表示、寻址技术、寄存器定义、指令系统、存储系统、中断系统、输入输出系统以及机器的工作状态等。

本章将依托于上海东软载波微电子有限公司自主研发的 HR 系列单片机,概要介绍计算机的基本结构。

1.1 计算机结构简介

在过去的半个世纪中,计算机硬件、软件技术的飞速发展都是令人惊叹不已的。如今,一台普通智能手机的性能已经超过了 20 年前的个人计算机了,而前者体积可能只是后者的百分之一。随着时间的推移,计算机的实现技术及性能指标都会不断提升,但系统内在的本质不会改变。几乎所有主流计算机系统都由相似的硬件和软件组成,它们的组织结构并不存在本质差异,其所扮演的角色及执行的功能也没有太多变化。本节将介绍两种典型的计算机结构:冯·诺依曼结构、哈佛结构。

1.1.1 冯·诺依曼结构

1946 年,被后世誉为“计算机之父”的冯·诺依曼(von Neumann)教授提出了一种将程序指令存储器和数据存储器合并在一起的计算机结构,即**冯·诺依曼结构**(也被称为**普林斯顿架构**),其逻辑结构如图 1-1 所示。直到今天,绝大多数个人计算机及嵌入式计算机仍然沿用冯·诺依曼结构,例如,Intel x86、8051、ARM7、Cortex-M0、MIPS 等。冯·诺依曼结构的主要特点如下:

第1章 计算机系统

- (1) 存储器是字节固定、线性编址的一维结构,每个地址是唯一定义的;
- (2) 机器运行由指令形式的低级机器语言驱动。指令顺序执行,即按照指令在存储器中存储的顺序依次执行,分支结构由转移指令完成;
- (3) 以运算器为中心,数据存储及输入输出都必须通过运算器;
- (4) 运算器、存储器及输入输出设备之间的协同工作由控制器集中控制。

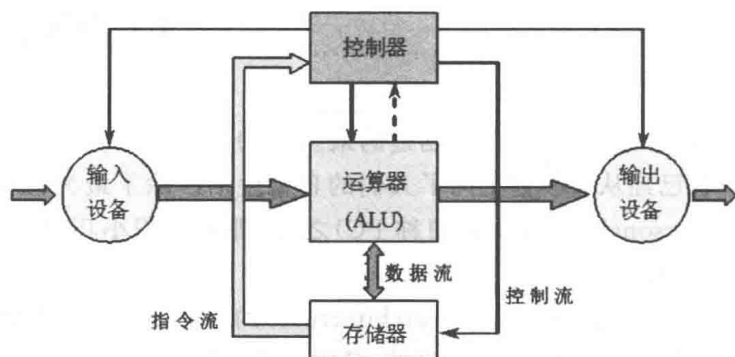


图 1-1 冯·诺依曼计算机结构

1.1.2 哈佛结构

哈佛结构是一种将程序指令和数据空间独立存储的计算机系统,其目的是为了缓解程序运行时的存储访问瓶颈,如图 1-2 所示。与传统冯·诺依曼结构不同,哈佛结构通过两条独立的总线分别用于取指令及取数据,因此从架构上解决了这两种行为在总线访问上的冲突问题。在嵌入式应用领域,基于哈佛结构的主流处理器包括:Microchip 公司的 PIC、摩托罗拉公司的 MC68、Atmel 公司的 AVR 以及 ARM 公司的 ARM9、ARM10、Cortex-M3 等。

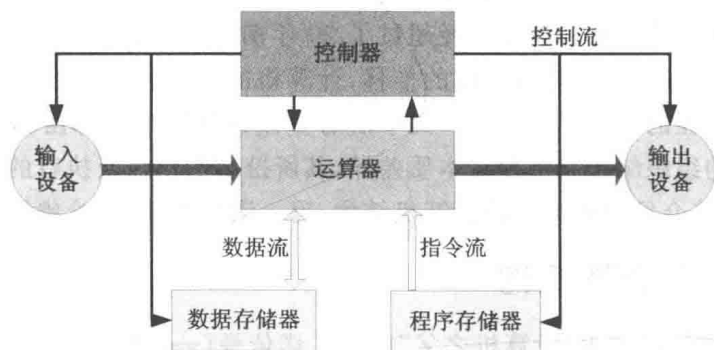


图 1-2 哈佛计算机结构

由于是通过两条独立系统总线,哈佛结构的处理器允许指令字长与数据字长不同。例如,ARM 公司 Cortex-M3 处理器的 Thumb 指令字长是 16 位,而数据字长