



UML+OOPC 嵌入式C语言开发精讲

高焕堂 著



电子工业出版社
www.eip.com.cn
http://www.phd.com.cn

介 简 容 内

UML+OOPC 嵌入式 C 语言开发精讲

高焕堂 著

容内稿全更代路立并本委外编制复失式同并以界不，可并登未

版权所有，侵权必究。

图书在版编目 (CIP) 数据

UML+OOPC 嵌入式 C 语言开发精讲 / 高焕堂著. — 北京: 电子工业出版社, 2008.9

ISBN 978-7-121-07108-9

I. U... II. 高... III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 123456 号

UML+OOPC 嵌入式 C 语言开发精讲

作者: 高焕堂

出版: 电子工业出版社

地址: 北京市西城区百万庄大街 22 号

邮编: 100037

电话: (010) 88252888

网址: www.eel.com.cn

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

质量投诉电话: 010-88252888 电子邮箱: qing@eel.com.cn

服务热线: (010) 88252888

内 容 简 介

OOPC 通过将面向对象程序设计技术与 C 语言相结合,以提升 C 语言的简洁性、易读性和重复使用性,进而提升嵌入式系统分析及架构设计的技术能力。本书由浅入深,从 C 语言的复习开始,然后讲述 C 语言如何与 OOP 相结合,接着从面向对象技术进入 UML,最后教读者从实践应用出发,活用 UML+OOPC 开发流程,做好系统分析和架构设计,实现高质量的嵌入式软件系统。本书的实例通过定制头文件,为 ANSI-C 和 Keil C51 添加类、接口的定义机制,来说明面向对象程序设计技术与 C 语言相结合的优点和特色。通过详细介绍相关的类、接口、UML 和基本模式,以深入进行 UML 分析及设计,让读者精确掌握系统需求和设计优美的系统架构,以求大幅提高嵌入式系统的可靠性和质量。本书适合广大嵌入式系统及应用开发人员阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

UML+OOPC 嵌入式 C 语言开发精讲 / 高焕堂著. —北京: 电子工业出版社, 2008.9

ISBN 978-7-121-07108-9

I. U… II. 高… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 106330 号

责任编辑: 陈元玉

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 35.25 字数: 550 千字

印 次: 2008 年 9 月第 1 次印刷

印 数: 5 000 册 定 价: 68.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

序 言

近年来, C 语言类书籍的销售量扶摇直上, 起因于在数码家电、手机、数字化汽车等产业中, 嵌入式 (Embedded) 软件应用愈来愈广。而嵌入式软件开发所使用的语言中, C 语言仍约占 80% 多。

由于嵌入式软件应用愈来愈广, 软件质量决定了数码产品的稳定性和可靠度, 因此, 如何提升 C 程序的简洁性、易读性及重复使用性, 乃是当今软件业的热门话题。例如, 世界知名的麦肯锡 (McKinsey) 顾问公司, 在 2006 年的报告 (“Getting better software to manufactured products”) 中, 呼吁嵌入式软件业必须积极提升其系统分析及架构设计的技术能力, 才能解决使用软件愈来愈多的数码产品的信赖度问题。

如何解决上述问题呢? 其方向已经很清楚了, 就是让 C 语言与面向对象程序设计 (Object-Oriented Programming, 简称 OOP) 技术相结合。就像当今的其它主流计算机语言 (如 VB.NET、C#、Java 等) 一样。由于当今的世界标准系统分析与架构设计的建模语言——UML, 也是基于面向对象技术而发展出来的, 因此, 一旦 C 语言与面向对象技术相结合了, 也就是与 UML 结合了, 便能逐渐提升系统分析与设计的质量。因此笔者在出版《精通 ANSI-C 语言》一书之后, 继续编写本书, 期望陪伴众多 C 程序员能更上层楼, 强化系统分析及架构设计的能力, 以适应日益热络的嵌入式系统开发市场的需要。

也许你会问: 在 1986 年时, 贝尔 (Bell) 实验室已经将 C 语言与面向对象技术结合成为 C++ 语言了, 为何还需要 OOPC 呢? 其答案是 C++ 语言有些贪心, 将整套的面向对象技术涵括进去, 导致 C++ 的效率远比单纯 C 语言慢了许多。由于嵌入式软件所能使用的硬件资源大都极为有限, 对程序执行效率斤斤计较; 所以在今天嵌入式软件开发上, 使用最广的仍是 C 语言。

而本书所介绍的面向对象 C 语言并不是一种新的语言, 它只运用单纯 C 语言的宏 (Macro) 技巧, 实现了面向对象的基本技术, 让系统分析与设计阶段的 UML 模型能与 C 程序紧密对应, 以提升 C 程序的质量。此外, 这些宏在编译阶段就被翻译为单纯 C 程序代码了, 仍然保持其单纯 C 的高效率, 符合嵌入式软件环境的需要。本书由浅入深分为 4 篇 (共 26 章):

第 1 篇——复习 ANSI-C 语言。

第2篇——介绍面向对象概念及 OOPC。详细说明了 ANSI-C 如何与 OOP 技术相结合成为面向对象 C 语言（简称 OOPC）。

第3篇——介绍 UML。

第4篇——UML + OOPC 实用示例。从实例演练中，运用 UML+OOPC 开发流程，做好系统分析和架构设计，实现高质量的嵌入式软件系统。

第5篇——面向对象 Keil C51 语言：在单片机（SOC）上的应用。

愿本书能陪伴你驰骋于嵌入式领域中，并鸿图大展。

高焕堂

谨识于 2008 年元月

联系博文视点

您可以通过如下方式与本书的出版方取得联系。

读者信箱: reader@broadview.com.cn

投稿信箱: bvtougao@gmail.com

北京博文视点资讯有限公司 (武汉分部)

湖北省 武汉市 洪山区 吴家湾 邮科院路特 1 号 湖北信息产业科技大厦 1402 室

邮政编码: 430074

电话: (027) 87690813 传真: (027) 87690813 转 817

若您希望参加博文视点的有奖读者调查, 或对写作和翻译感兴趣, 欢迎您访问:

<http://bv.csdn.net>

关于本书的勘误、资源下载及博文视点的最新书讯, 欢迎您访问博文视点官方博客:

<http://blog.csdn.net/bvbook>

目 录

第 1 篇 复习 ANSI-C 语言	1
第 1 章 嵌入式时代最划算策略	3
1.1 迎接高质量的嵌入式软件时代	4
1.2 基于 Turbo C 2.0 环境的评估	4
1.2.1 加入第 1 个类	4
1.2.2 加入第 2 个类	6
1.2.3 加入第 3 个类	9
1.2.4 加入第 4 个类	11
1.2.5 评估统计图	13
1.3 基于 Visual C++ 环境的评估	14
1.4 LW_OOPC 与 C++ 的比较和评估	17
第 2 章 C 程序的基本组成	19
2.1 认识 C 语言与 LW_OOPC 语言	20
2.1.1 C 语言的身世背景	20
2.1.2 C 影响 C++、Java、C# 等语言文化	20
2.1.3 用 C 语言编写面向对象 (Object-Oriented) 程序	21
2.1.4 面向对象概念让 UML 与 C 携手合作	22
2.2 函数: C 程序的基本结构	25
2.2.1 指令、函数与程序	25
2.2.2 函数间的调用 (Call)	25
2.2.3 库函数	26
2.3 变量的概念	26
2.3.1 数据分类与变量	26
2.3.2 声明变量	27
2.4 变量的声明格式	27
2.5 如何输出数据	28
2.6 如何传递参数	29
2.7 如何替函数和变量命名	30

第3章 C语言的数据类型	33
3.1 基本数据类型	34
3.2 整数类型	34
3.2.1 short int 类型	35
3.2.2 long int 类型	36
3.3 无符号整数	37
3.3.1 无符号字符 (unsigned char)	38
3.3.2 无符号短整数 (unsigned short int)	38
3.3.3 无符号 (长) 整数 (unsigned int)	39
3.4 整数的输出格式	41
3.5 字符类型	43
3.5.1 一般字符	43
3.5.2 控制字符	45
3.6 浮点数类型	46
3.6.1 float 类型	46
3.6.2 double 类型	48
第4章 C的数据运算	51
4.1 基本运算符	52
4.2 算术及赋值运算	52
4.3 关系运算	56
4.4 逻辑运算符	57
4.5 算术赋值运算符	59
4.6 加1及减1运算符	60
4.7 取地址运算符	61
4.8 按位运算符	62
4.8.1 &、 、^ 及~ 运算符	63
4.8.2 <<及>>运算符	65
4.8.3 按位赋值运算符	66
4.9 类型转换运算符	67
第5章 决策与循环	69
5.1 逻辑运算与决策	70
5.2 嵌套的 if 指令	71
5.3 多选1的抉择	72

5.4	while 循环	74
5.5	for 循环	75
5.5.1	基本格式	75
5.5.2	各式各样的 for 循环	76
5.6	do 循环	78
第 6 章	C 语言的指针	79
6.1	认识指针	80
6.1.1	指针是什么	80
6.1.2	指针的声明	80
6.1.3	指针的指针	81
6.2	传递指针参数	82
6.3	函数回传指针	83
6.4	函数指针	84
第 7 章	结构 (struct) 及动态内存分配	87
7.1	C 语言的结构 (struct)	88
7.2	结构指针	90
7.3	传递结构参数	91
7.4	结构内的函数指针	92
7.4.1	先介绍 typedef 指令	92
7.4.2	复习函数指针	93
7.4.3	把函数指针放入结构里	94
7.4.4	让函数存取结构里的数据细项	94
7.5	动态内存分配	97
7.5.1	malloc() 及 free() 函数	97
7.5.2	calloc() 及 realloc() 函数	100
第 8 章	外部变量与静态函数	101
8.1	变量的储存种类	102
8.2	自动变量	103
8.3	外部变量	103
8.4	外部静态变量	104
8.5	extern 种类	107
8.6	静态函数	110

第 9 章 数组与字符串	113
9.1 数组的意义	114
9.2 1 维数组	114
9.3 1 维数组与指针	115
9.4 2 维数组与多维数组	120
9.5 2 维数组与指针	121
9.6 数组参数	123
9.6.1 1 维数组参数	123
9.6.2 2 维数组参数	124
9.7 为数组赋初值	125
9.8 使用字符串	126
9.8.1 何谓字符串	126
9.8.2 给予字符串初值	126
9.9 库字符串函数	127
9.10 传递字符串参数	129
第 10 章 预处理程序	131
10.1 预处理程序的工作	132
10.2 使用宏	132
10.2.1 宏常数	132
10.2.2 #define 与 typedef 的区别	134
10.2.3 带参数的宏	135
10.2.4 取消宏	138
10.3 添加头文件	139
10.4 条件性编译	140
10.4.1 条件性编译概述	140
10.4.2 条件性定义	141
10.5 认识 MISOO 的 lw_oopc.h 宏文件	143
10.5.1 复习重要的 C 宏	143
10.5.2 使用 lw_oopc.h 头文件	145
第 2 篇 介绍面向对象概念及 OOPC	149
第 11 章 认识对象 (Object)	151
11.1 自然界的对象 (Natural Object)	152
11.1.1 对象 (Object)	152

11.1.2	信息 (Message)	152
11.1.3	事件 (Event)	152
11.2	软件对象 (Software Object)	153
11.2.1	“抽象”的意义	153
11.2.2	抽象表示	153
11.2.3	数据和函数	153
11.2.4	历史的足迹	154
11.3	对象与函数	155
11.3.1	函数的角色	155
11.3.2	对象与类	156
11.3.3	类的用途: 描述对象的共同特点	156
11.4	对象与类	157
11.4.1	类的用途	157
11.4.2	定义类	158
11.5	对象指针	161
11.6	构造器 (Constructor)	162
11.7	类设计的实例说明	163
11.7.1	以电灯 (Light) 类为例	163
11.7.2	以数学矩阵 (Matrix) 类为例	164
第 12 章	对象沟通方法	167
12.1	“信息传递”沟通方法	168
12.2	“信息传递”示例说明	169
12.2.1	分析与设计	169
12.2.2	设计 OOPC 类	169
12.2.3	生成 OOPC 对象	170
12.3	以 OOPC 实现: 使用 Turbo C	170
12.4	以 OOPC 实现: 使用 VC++ 2005	173
第 13 章	对象沟通实例	181
13.1	以向量类封装 1 维数组	182
13.1.1	定义 Vector 类	182
13.1.2	运用 malloc() 库函数	183
13.1.3	运用 #define 语句	184
13.1.4	运用 void* 指针	185

13.2	以矩阵类封装 2 维数组	189
13.2.1	定义 Matrix 类	189
13.2.2	Matrix 对象包含 Vector 对象	190
第 14 章	认识接口 (Interface)	197
14.1	如何定义接口	198
14.2	多个类实现同一接口	201
14.3	以接口实现多态性 (Polymorphism)	204
14.4	一个类实现多个接口	208
第 15 章	接口应用实例	213
15.1	电池接口的用意	214
15.2	设计电池接口	214
15.3	以 OOPC 实现接口设计	216
第 16 章	集合对象链表 (Linked List)	221
16.1	认识集合对象	222
16.2	以 OOPC 实现 LList 集合类	224
16.3	应用实例说明	229
第 17 章	LW_OOPC 宏的设计思维	235
17.1	前言	236
17.2	从 ANSI-C 出发	236
17.3	运用 C 的结构	237
17.4	设计构造器	238
17.5	运用函数指针	239
17.6	运用 C 宏	240
17.6.1	定义宏: CLASS(类名称)	241
17.6.2	定义宏: CTOR (类名称)	242
17.7	定义接口 (Interface) 宏	243
17.8	定义 CTOR2()构造器宏	246
17.9	将宏独立成 lw_oopc.h 头文件	248
第 3 篇	介绍 UML	251
第 18 章	认识 UML	253
18.1	UML: 世界标准对象模型语言	254
18.2	UML 的演化	254
18.3	UML 的基本元素	256

18.4 UML 的图示	256
第 19 章 UML 类图	259
19.1 为什么需要面向对象思维	260
19.2 为什么需要设计类	261
19.3 为什么要描述类间的关系	263
19.3.1 类间的组合关系	263
19.3.2 类间的结合关系	265
19.4 为什么要绘制 UML 类图	267
19.5 如何绘制 UML 类图	270
19.5.1 EA 的类图	270
19.5.2 StarUML 的类图	272
19.5.3 JUDE 的类图	272
19.6 如何得到类	274
19.6.1 从对象归类而得到类	274
19.6.2 从领域概念 (Domain Concepts) 找到类	276
第 20 章 UML 用例图	279
20.1 为什么需要用例图	280
20.2 用例的内涵是什么	280
20.2.1 用例图表达 What 及 Who	281
20.2.2 用例描述表达 How 及 When	282
20.3 用例与对象的密切关系	283
20.4 用例的经济意义	284
20.5 用例间的关系	286
20.5.1 ud Bill-1 UC 描述: 结账	288
20.5.2 ud Bill-1 UC 描述: 买汉堡套餐	288
20.5.3 ud Bill-2 UC 描述: 结账	289
20.5.4 ud Bill-2 UC 描述: 买汉堡套餐	289
20.5.5 ud Bill-2 UC 描述: 赠送玩具	289
20.6 企业用例与系统用例	290
20.6.1 上、下层级的用例	290
20.6.2 上、下层级用例的美妙关联	290
20.6.3 从“上层用例”导出“下层用例”的步骤	292
20.7 如何绘制 UML 用例图	295

第 21 章 UML 序列图	299
21.1 UML 序列图的意义	300
21.2 UML 序列图的语法	301
21.3 如何绘制 UML 序列图	302
21.4 从序列图落实到 OOPC 类	304
21.4.1 共享类的考虑	305
21.4.2 对应到类	306
21.4.3 对应到函数	306
21.4.4 对方对象→信息名称(参数)	308
21.5 UML 序列图示例说明	309
21.5.1 绘制用例图	309
21.5.2 绘制类图	310
21.5.3 绘制序列图	311
21.5.4 对应到 OOPC 类	312
第 22 章 UML 对象状态图	315
22.1 Why 状态图	316
22.2 简介 UML 状态图	316
22.2.1 状态、事件与转移	316
22.2.2 活动	318
22.2.3 复合状态	319
22.2.4 子机状态	320
22.2.5 历史状态	321
22.2.6 决策	323
22.2.7 汇合	323
22.2.8 并行	324
22.2.9 同步	324
22.3 使用 UML 绘图工具	325
22.3.1 绘图区	325
22.3.2 工具箱	326
22.3.3 画一个状态	326
22.3.4 状态转移	327
22.3.5 活动的表示	328
22.4 如何以 OOPC 实现 UML 状态图	329

22.4.1	举例：以小灯状态为例	329
22.4.2	举例：以冰箱的状态为例	332
22.4.3	举例：以银行账户（Account）的状态为例	334
第 4 篇 UML+OOPC 实用示例		341
第 23 章 UML+OOPC 实用示例之一		343
23.1	形形色色的涂鸦程序	344
23.2	涂鸦程序示例说明	345
23.3	涂鸦系统分析与设计	346
23.3.1	绘制系统用例（Use Case）图	346
23.3.2	绘制类图	348
23.3.3	绘制 Scribble 状态图	349
23.3.4	绘制序列图	350
23.3.5	用例：“涂鸦”	350
23.3.6	用例：“播放”	351
23.4	涂鸦程序的实现：使用 OOPC 语言	352
第 24 章 UML+OOPC 实用示例之二		365
24.1	认识“录音”概念和技术	366
24.1.1	认识 PCM 规格	366
24.1.2	设定录音格式	366
24.1.3	设定缓冲区格式	367
24.1.4	将音频数据写入.wav 声音文件	367
24.1.5	使用 Win32 所提供的录音 API	368
24.2	单纯“录音”的示例分析	368
24.2.1	绘制系统用例图	368
24.2.2	绘制类图	368
24.2.3	绘制序列图	370
24.3	“录音/播放”示例的分析	370
24.3.1	绘制系统用例图	370
24.3.2	绘制类图	372
24.3.3	绘制序列图	372
24.4	“录音/播放”示例的实现	372
第 25 章 UML+OOPC 实现示例之三		385
25.1	层次分析（AHP）法简介	386

25.2	AHP 的分析步骤.....	387
25.3	如何得到权数值.....	388
25.3.1	成对相比.....	388
25.3.2	从“成对比值”算出“权数值”.....	391
25.3.3	“成对比值”的一致性检验.....	396
25.4	“AHP”示例分析与设计.....	398
25.4.1	绘制系统用例图.....	398
25.4.2	绘制类图.....	399
25.4.3	绘制序列图.....	400
25.5	“AHP”示例的实现：使用 OOPC.....	402
25.5.1	准备决策数据.....	402
25.5.2	以 OOPC 编写 AHP 程序.....	404
第 26 章	UML+OOPC 实用示例之四.....	413
26.1	什么是半加器.....	414
26.2	设计一个“位计算器”.....	415
26.2.1	以软件模拟硬件的意义.....	415
26.2.2	设计单位计算器的操作画面.....	416
26.2.3	设计单位计算器的 UML 状态图.....	417
26.3	实现位计算器：使用 OOPC.....	418
第 5 篇	面向对象 Keil C51 语言：在单片机（SOC）上的应用.....	427
第 27 章	替 Keil C51 黄袍加身.....	429
27.1	以 200 Bytes 代价换得优雅架构.....	430
27.2	3 种弹性又高雅的写法.....	431
27.2.1	动态型（Dynamic，昵称为豪华型）.....	431
27.2.2	静态型（Static，昵称为标准型）.....	431
27.2.3	纯粹静态型（Pure Static，昵称为精简型）.....	432
27.3	静态（Static）型写法及其评估.....	432
27.4	纯粹静态（Pure Static）型写法及其评估.....	441
27.5	动态（Dynamic）型写法及其评估.....	444
第 28 章	Keil C51 的特殊数据类型.....	449
28.1	8051 的 CODE 存储区.....	450
28.2	8051 的 DATA 存储区.....	450
28.3	Keil C 的存储模式.....	451

28.4 Keil C 的专用数据类型	452
第 29 章 以 Keil C51 定义类	455
29.1 定义类	456
29.2 构造器 (Constructor)	456
29.3 Keil C51 类设计之实例说明	457
29.3.1 分析与设计	457
29.3.2 以 Keil C 实现红绿灯控制程序	459
29.3.3 红绿灯类的另一种写法	461
第 30 章 应用范例一	463
30.1 以 Toggle Light 电灯为例	464
30.1.1 分析与设计	464
30.1.2 设计 OOPC 类	464
30.1.3 以 OOPC 实现: 使用 Keil C	465
30.2 以红绿灯控制系统为例	469
30.2.1 分析与设计	469
30.2.2 以 LW_OOPC 实现: 使用 Keil C	473
第 31 章 应用范例二	479
31.1 界面用途: 从硬件的 PnP 谈起	480
31.1.1 硬件端口 (Port) 就是接口	480
31.1.2 软件接口	480
31.2 LED 显示器控制设计 (1)	481
31.2.1 分析与设计	481
31.2.2 UML 图示	482
31.2.3 以 Keil C 实现范例	483
31.3 LED 显示器控制设计 (2)	486
31.3.1 分析与设计	486
31.3.2 以 Keil C 实现范例	487
31.4 LED 显示器控制设计 (3)	489
第 32 章 应用范例三	495
32.1 模式观念	496
32.2 软件设计模式	496
32.2.1 Why 设计模式	496
32.2.2 设计模式的起源	497