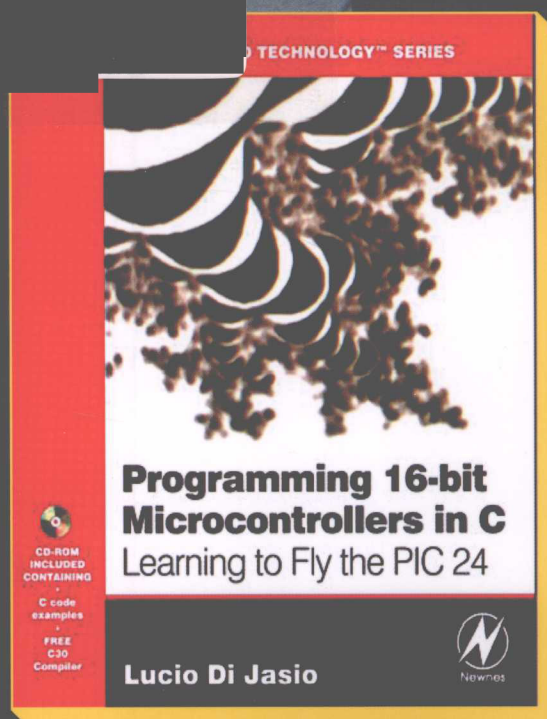


16位单片机C语言编程 基于PIC24

Programming 16-bit Microcontrollers in C
Learning to Fly the PIC 24

[意] Lucio Di Jasio 著
李中华 张雨浓 黄晓红 译



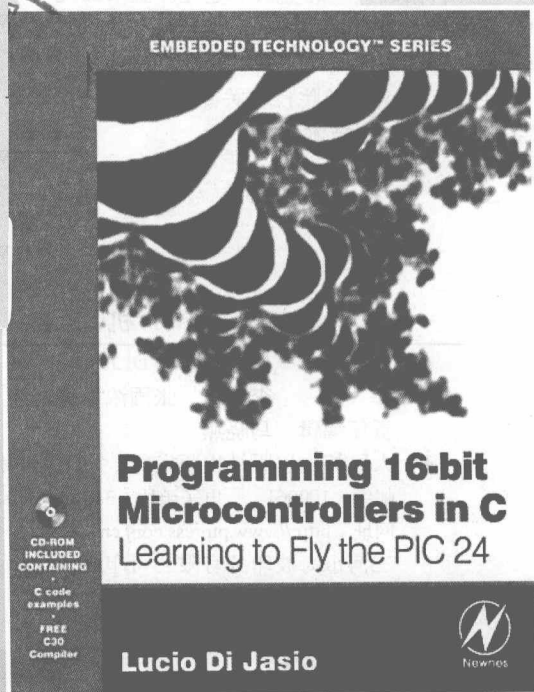
人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵电子与电气工程丛书

16位单片机C语言编程 基于PIC24

Programming 16-bit Microcontrollers in C
Learning to Fly the PIC 24



人民邮电出版社
北京

图书在版编目(CIP)数据

16位单片机C语言编程: 基于PIC 24/(意)贾西欧
(Jasio, L. D.) 著; 李中华, 张雨浓, 黄晓红译. —北
京: 人民邮电出版社, 2010.4
(图灵电子与电气工程丛书)
ISBN 978-7-115-22149-0

I. ①1 … II. ①贾… ②李… ③张… ④黄… III. ①
单片微型计算机—C语言—程序设计 IV. ①TP368.1
②TP312

中国版本图书馆CIP数据核字(2010)第007537号

内 容 提 要

本书是关于16位PIC微控制器C语言编程的经典著作, 采用飞行员训练教程的模式, 历经从“首次飞行”至“自由翱翔”的全训练过程。全书围绕PIC 24微控制器应用系统设计的C语言描述, 从PIC 24微控制器的基本C编程语法开始, 涵盖了PIC 24微控制器中断处理、存储器分配、通信接口、人机接口、视频处理、外围部件接口等模块的功能原理和C程序实现等内容。

本书即可作为高等院校相关专业本科生、研究生的课程教材, 也可供从事微控制器应用设计和嵌入式系统开发的工程技术人员参考。

图灵电子与电气工程丛书

16位单片机C语言编程: 基于PIC 24

-
- ◆ 著 [意] Lucio Di Jasio
 - 译 李中华 张雨浓 黄晓红
 - 责任编辑 马晓燕
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19
字数: 472千字 2010年4月第1版
印数: 1-3 000册 2010年4月河北第1次印刷
著作权合同登记号 图字: 01-2009-2898号
ISBN 978-7-115-22149-0
-

定价: 49.00元

读者服务热线: (010) 51095186 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

版 权 声 明

Programming 16-bit Microcontrollers in C: Learning to Fly the PIC24, by Lucio Di Jasio, ISBN: 978-0-7506-8292-3.

Copyright © 2007 by Elsevier Inc. All rights reserved.

Authorized Simplified Chinese translation edition published by the Proprietor.

ISBN: 978-981-272-185-3.

Copyright© 2009 by Elsevier (Singapore) Pte Ltd. All rights reserved.

Elsevier (Singapore) Pte Ltd.

3 Killiney Road

#08-01 Winsland House I

Singapore 239519

Tel: (65)6349-0200

Fax: (65)6733-1817

First Published 2010

2010 年初版

Printed in China by POSTS & TELECOM PRESS under special arrangement with Elsevier (Singapore) Pte Ltd. This edition is authorized for sale in China only, excluding Hong Kong SAR and Taiwan. Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书简体中文版由 Elsevier (Singapore) Pte Ltd. 授权人民邮电出版社在中华人民共和国境内（不包括香港特别行政区和台湾地区）出版与销售。未经许可之出口，视为违反著作权法，将受法律之制裁。

引 言

一直以来，我几乎天天都在想着写一本书，写写我这一辈子最钟爱的一件事——驾驶飞机遨游蓝天！希望通过这本书激励其他工程师能像我一样敢于冒险、实现梦想——学习飞行，成为私人飞机驾驶员。然而，有限的实际飞行经历还不能让我成为一名值得信赖的飞行专家。因此，当有机会撰写一本关于 Microchip 最新的 16 位 PIC24 微控制器的书籍时，我忍不住想尝试将编程和飞行结合起来。毕竟，学习飞行也要遵循一个成熟的训练过程，即人们熟悉新技能并超越自身极限的一个历程。它通常引导你通过一定的理论学习和实际操作，才能获得初级飞行员的资格。飞行员资格实际上只是一个崭新冒险过程的起点，有人说那是继续学习的资格。其实，飞行学习的过程和学习新的编程技巧或者掌握新型微控制器功能的过程是极其相似的。

我将这两个学习领域的平行式比拟贯穿于全书，并在每一章的参考文献中也介绍一些飞行读物。如果读者真的有这种飞行梦想，希望本书能激发起读者的好奇心，给读者以梦想成真的学习动力。

读者定位

我本该在这里告诉读者：在阅读本书的时候，你将会体验到很多有趣的软件和硬件实验，而且会学习到如何在全新的 16 位 RISC 处理器上从零开始使用 C 语言编程。但是，说实话，我没法这么说，因为这不是十分准确。真心希望读者在阅读本书的时候能够体验到更多的乐趣、感受更多的趣味实验。不过，读者必须做些准备工作并努力学习，才能消化本书内容，经过前几章介绍后内容难度会很快加大。

本书是为具有初级和中级编程能力的人员编写的，不适合纯粹意义上的“新手”。因此，本书不会从最基础的二进制数、十六进制符号以及编程基础知识开始讲授。不过，在介绍难度较大的项目之前，本书将简单地介绍 C 语言的编程基础，因为它和最新的通用 16 位微控制器应用密切相关。本书尤其适合下列 4 类人员。

- 嵌入式控制程序员：具有基于汇编语言的微控制器编程经验，但对 C 语言编程只有基本的认识。
- PIC 微控制器专家：对 C 语言编程有基本的了解。
- 学生或专业人员：对 PC 的 C（或 C++）编程有一定知识。
- 其他高手：鉴于程序员不喜欢被简单地分类，所以特意为读者创造了这个类别！

不同层次和经验的读者，都可以在各章中找到感兴趣的内容。本书将尽量保证在每一章中都安排关于 C 语言编程技巧和新型外围硬件设备的介绍。如果读者对相关内容都已经熟悉了，完全可以跳到每一章最后针对专家的部分，或者思考附加练习，甚至进一步地研究/阅读参考书目和网上链接。

本书将介绍以下内容。

- 嵌入式控制的 C 程序结构：循环，循环，再循环。
- 基本的计时和 I/O 操作。
- 使用 PIC24 的中断实现 C 语言的基本嵌入式控制的多任务。
- 新的 PIC24 外围设备（以下不分顺序）。
 - 输入捕获。
 - 输出比较。
 - 改变通知。
 - 主并行端口。
 - 异步串行通信。
 - 同步串行通信。
 - 模数转换。
- 如何控制 LCD 显示。
- 如何生成视频信号。
- 如何生成音频信号。
- 如何访问大容量媒介。
- 如何与 PC 实现大容量设备的文件共享。

本书结构

像飞行课程一样，本书由三部分组成。第一部分由 5 个难度逐渐递增的章节组成。其中每一章都会介绍 PIC24FJ128GA010 微控制器的一个基本外围硬件设备和一个 C 语言问题。而且，在每一章中至少会开发一个演示项目。开始时，这些项目需要使用 MPLAB SIM 软件仿真器，除了可能用到 Explorer16 演示板外，不需要其他真实的硬件设备。

本书的第二部分由 5 个章节组成。因为某些外围设备需要真实硬件进行测试，所以 Explorer16 演示板（或者相似的第三方设备）在这部分会变得更重要。

本书的第三部分包括 5 个内容更为丰富的章节。每一章的内容都建立在过去章节介绍的课程之上，同时因为开发项目的复杂性提高，还增加了新的外围设备内容。本书第三部分的项目需要用到 Explorer 16 演示板和基本的焊板知识（读者可能需要用到电烙铁）。如果读者想避免麻烦，或者尚没有基本的硬件焊接工具，可以从网站 <http://www.flyingthepic24.com> 上找到一个能满足全部演示项目需要的电路和元件扩展板。

每一章给出的所有源代码也已包含在本书附属资源内，读者可以登录图灵公司网站 (<http://www.turingbook.com>)，免费注册后下载。

不要误解了本书

本书不能代替 Microchip 公司出版的 PIC24 数据表、参考指南和程序员手册，也不能代替 MPLAB C30 编译器的用户指南及 Microchip 提供的所有程序库和相关软件工具。整本书会经常提及上述文件和工具，而且必要时将给出方框图和摘录。本书的叙述不能代替官方网站或者用

户指南上提供的信息。读者应该注意本书的表述是否和官方文档上有分歧，必须随时参考最新材料。如果确实发现有不一致的地方，请读者一定要用电子邮件告诉我。我将不胜感激，并且会在网站 <http://www.flyingthepic24.com> 上发布收到的所有纠错信息和实用提示。

本书也不能作为 C 语言的初级读本。虽然在前面的部分章节中对 C 语言作了一些回顾，但是读者可以在参考资料部分列出的课程和图书中找到更完善的介绍。

备忘录

无论是专业还是业余的飞行员，在每次飞行前或者飞行中，都会按照备忘录来完成每个操作。这并不是由于那些操作步骤长得无法记忆，又或者是飞行员的记忆力比其他人差。原因在于：研究表明人的记忆是会衰退的，尤其是在承受压力的时候，因此飞行员都会使用备忘录。使用备忘录也会使他们比其他专业人士犯的错误更少，飞行员可是将保证安全看得比什么都重要的。

当然，作为程序员，在使用 PIC24 进行开发编程时，即使多做或者忘记了做什么，也不会发生什么致命危险。不过，本书还是为读者准备了一些简单的常用编程和调试任务的备忘录。希望这些备忘录能在读者刚开始学习新 PIC24 工具时提供帮助——甚至在以后像作者一样同时面对不同厂家的开发环境和多个项目时，仍然能够派上用场。

目 录

第一部分 飞行人门

第 1 章 首飞	2
1.1 飞行计划	2
1.2 飞前备忘录	2
1.3 飞行	3
1.3.1 编译和连接	4
1.3.2 构建第一个项目	5
1.3.3 端口初始化	7
1.3.4 重测 PORTA	8
1.3.5 测试 PORTB	9
1.4 飞后小结	11
1.5 给汇编语言专家的提示	11
1.6 给 PIC 微控制器专家的提示	12
1.7 给 C 语言专家的提示	12
1.8 提示与技巧	12
1.9 练习	13
1.10 推荐书目	13
1.11 网上链接	13
第 2 章 模式循环	14
2.1 飞行计划	14
2.2 飞前备忘录	14
2.3 飞行	15
2.3.1 while 循环	15
2.3.2 动画模拟	17
2.3.3 使用逻辑分析器	20
2.4 飞后小结	22
2.5 给汇编语言专家的提示	22
2.6 给 PIC 微控制器专家的提示	23
2.7 给 C 语言专家的提示	23
2.8 提示与技巧	23
2.9 练习	23
2.10 推荐书目	24
2.11 网上链接	24
第 3 章 更多模式, 更多循环	25
3.1 飞行计划	25

3.2 飞前备忘录	25
3.3 飞行	25
3.3.1 do 循环	26
3.3.2 变量声明	26
3.3.3 for 循环	27
3.3.4 更多循环示例	28
3.3.5 数组	29
3.3.6 新的演示程序	29
3.3.7 使用逻辑分析器测试	31
3.3.8 使用 Explorer16 演示板	32
3.4 飞后小结	32
3.5 给汇编语言专家的提示	32
3.6 给 PIC 微控制器专家的提示	32
3.7 给 C 语言专家的提示	33
3.8 提示与技巧	33
3.9 练习	34
3.10 推荐书目	34
3.11 网上链接	34
第 4 章 数据类型	35
4.1 飞行计划	35
4.2 飞前备忘录	35
4.3 飞行	36
4.3.1 关于优化	37
4.3.2 测试	37
4.3.3 走近长整型	38
4.3.4 长整型数据乘法说明	39
4.3.5 双长整型数据的乘法	39
4.3.6 浮点型	39
4.4 给 C 语言专家的提示	40
4.5 飞后小结	42
4.6 给汇编语言专家的提示	43
4.7 给 PIC 微控制器专家的提示	44
4.8 提示与技巧	44
4.8.1 函数库	44
4.8.2 复数数据类型	44
4.9 练习	45
4.10 推荐书目	45

4.11 网上链接	45
第 5 章 中断	46
5.1 飞行计划	46
5.2 飞前备忘录	46
5.3 飞行	46
5.3.1 中断嵌套	50
5.3.2 陷阱	50
5.3.3 Timer1 中断的模板和 示例	50
5.3.4 Timer1 应用实例	51
5.3.5 Timer1 中断的测试	53
5.3.6 二级振荡器	55
5.3.7 实时时钟日历 (RTCC)	56
5.3.8 多个中断的管理	56
5.4 飞后小结	57
5.5 给 C 语言专家的提示	57
5.6 给汇编语言专家的提示	57
5.7 给 PIC 微控制器专家的提示	57
5.8 提示与技巧	57
5.9 练习	59
5.10 推荐书目	59
5.11 网上链接	59
第 6 章 剖析引擎	60
6.1 飞行计划	60
6.2 飞前备忘录	60
6.3 飞行	60
6.3.1 存储器空间分配	62
6.3.2 程序空间可视化	63
6.3.3 存储器分配	64
6.3.4 查看 MAP 文件	67
6.3.5 指针	69
6.3.6 堆	70
6.3.7 MPLAB C30 存储器模型	70
6.4 飞后小结	71
6.5 给 C 语言专家的提示	71
6.6 给汇编语言专家的提示	71
6.7 给 PIC 微控制器专家的提示	71
6.8 提示与技巧	72
6.9 练习	72
6.10 推荐书目	72
6.11 网上链接	72

第二部分 单 飞

第 7 章 通信	74
7.1 飞行计划	74
7.2 飞前备忘录	74
7.3 飞行	74
7.3.1 同步串行接口	75
7.3.2 异步串行接口	76
7.3.3 并行接口	77
7.3.4 使用 SPI 模块进行同步通信	77
7.3.5 测试读状态寄存器命令	79
7.3.6 写 EEPROM	82
7.3.7 读存储器内容	82
7.3.8 非易失性存储库	83
7.3.9 测试新的 NVM 库	85
7.4 飞后小结	87
7.5 给 C 语言专家的提示	87
7.6 给汇编语言专家的提示	87
7.7 给 PIC 微控制器专家的提示	88
7.8 提示与技巧	88
7.9 练习	89
7.10 推荐书目	89
7.11 网上链接	89
第 8 章 异步通信	90
8.1 飞行计划	90
8.2 飞前备忘录	90
8.3 飞行	90
8.3.1 UART 配置	92
8.3.2 发送和接收数据	93
8.3.3 测试串行通信程序	94
8.3.4 建立简单的控制库	96
8.3.5 测试 VT100 终端	98
8.3.6 使用串行端口作为调试工具	99
8.3.7 黑客帝国	99
8.4 飞后小结	101
8.5 给 C 语言专家的提示	101
8.6 给 PIC 微控制器专家的提示	102
8.7 提示与技巧	102
8.8 练习	103
8.9 推荐书目	103
8.10 网上链接	103

第 9 章 玻璃护航	104
9.1 飞行计划.....	104
9.2 飞前备忘录.....	104
9.3 飞行.....	104
9.3.1 HD44780 控制器的兼容性.....	105
9.3.2 并行主控制端口.....	107
9.3.3 LCD 模块控制的 PMP 配置.....	107
9.3.4 访问 LCD 显示的小函数库.....	108
9.3.5 高级 LCD 控制.....	111
9.4 飞后小结.....	113
9.5 给 C 语言专家的提示.....	113
9.6 提示与技巧.....	114
9.7 练习.....	114
9.8 推荐书目.....	114
9.9 网上链接.....	115
第 10 章 模拟的世界	116
10.1 飞行计划.....	116
10.2 飞前备忘录.....	116
10.3 飞行.....	117
10.3.1 首次转换.....	119
10.3.2 自动采样定时.....	119
10.3.3 开发演示程序.....	120
10.3.4 开发游戏.....	121
10.3.5 温度测量.....	123
10.3.6 Breath-Alizer 游戏.....	126
10.4 飞后小结.....	127
10.5 给 C 语言专家的提示.....	127
10.6 提示与技巧.....	127
10.7 练习.....	127
10.8 推荐书目.....	128
10.9 网上链接.....	128

第三部分 跨国飞行

第 11 章 输入捕捉	130
11.1 飞行计划.....	130
11.2 飞行.....	130
11.2.1 PS/2 通信协议.....	131
11.2.2 PIC24 连接 PS/2.....	132
11.2.3 输入捕捉.....	132
11.2.4 使用激励脚本测试输入	

捕捉方法.....	136
11.2.5 测试 PS/2 接收子程序.....	139
11.2.6 仿真.....	140
11.2.7 仿真器规范.....	142
11.2.8 另一种方法—— 变化通知.....	142
11.2.9 开销计算.....	146
11.2.10 第三种方法—— I/O 查询.....	147
11.2.11 测试 I/O 查询方法.....	151
11.2.12 方案性价比.....	153
11.2.13 完成接口：添加 FIFO 缓冲器.....	154
11.2.14 完成接口：解码按键码.....	158
11.3 飞后小结.....	160
11.4 提示与技巧.....	161
11.5 练习.....	161
11.6 推荐书目.....	161
11.7 网上链接.....	161
第 12 章 暗屏	162
12.1 飞行计划.....	162
12.2 飞行.....	162
12.2.1 产生合成视频信号.....	164
12.2.2 使用输出比较模块.....	168
12.2.3 存储器分配.....	170
12.2.4 图像串行化.....	171
12.2.5 构建视频模块.....	173
12.2.6 视频发生器测试.....	176
12.2.7 性能测定.....	178
12.2.8 暗屏.....	179
12.2.9 测试图样.....	179
12.2.10 描点.....	181
12.2.11 星夜.....	182
12.2.12 画线.....	183
12.2.13 Bresenham 算法.....	184
12.2.14 画数学函数图.....	187
12.2.15 二维函数可视化.....	188
12.2.16 分形几何.....	191
12.2.17 文本.....	197
12.2.18 测试 TextOnGPage 模块.....	200
12.2.19 开发文本页视频.....	201

12.2.20 测试文本页性能	209	14.2.9 测试 fopenM() 和 fcloseM()	253
12.3 飞后小结	211	14.2.10 向文件写入数据	255
12.4 提示与技巧	212	14.2.11 关闭文件, 第二次执行	259
12.5 练习	212	14.2.12 辅助函数	260
12.6 推荐书目	213	14.2.13 测试整个文件 I/O 模块	263
12.7 网上链接	213	14.2.14 代码大小	266
第 13 章 大容量存储	214	14.3 飞后小结	267
13.1 飞行计划	214	14.4 提示与技巧	267
13.2 飞行	214	14.5 练习	267
13.2.1 SD/MMC 卡物理接口	215	14.6 推荐书目	268
13.2.2 连接 Explorer16 演示板	215	14.7 网上链接	268
13.2.3 开始一个新项目	216	第 15 章 翱翔	269
13.2.4 选择 SPI 操作模式	217	15.1 飞行计划	269
13.2.5 在 SPI 模式发送命令	217	15.2 飞行	269
13.2.6 完成 SD/MMC 卡初始化	219	15.2.1 在 PWM 模式下使用 PIC OC 模块	271
13.2.7 从 SD/MMC 卡读取数据	221	15.2.2 将 PWM 用作数/模转换器 测试	273
13.2.8 向 SD/MMC 卡写入数据	223	15.2.3 产生模拟波形	274
13.2.9 使用 SD/MMC 接口模块	225	15.2.4 语音信息再生	276
13.3 飞后小结	228	15.2.5 媒体播放器	276
13.4 提示与技巧	228	15.2.6 WAVE 文件格式	277
13.5 练习	229	15.2.7 函数 play()	278
13.6 推荐书目	229	15.2.8 低级音频程序	283
13.7 网上链接	229	15.2.9 测试 WAVE 文件播放器	286
第 14 章 文件 I/O	230	15.2.10 优化文件 I/O	288
14.1 飞行计划	230	15.2.11 LED 剖析	288
14.2 飞行	231	15.2.12 发掘更多	290
14.2.1 扇区和簇	231	15.3 飞后小结	293
14.2.2 文件分配表 (FAT)	232	15.4 提示与技巧	294
14.2.3 根目录	233	15.5 练习	294
14.2.4 寻宝	234	15.6 推荐书目	294
14.2.5 打开一个文件	241	15.7 网上链接	294
14.2.6 从文件中读取数据	248		
14.2.7 关闭一个文件	251		
14.2.8 创建文件 I/O 模块	251		

第一部分

飞 行 人 门

第 1 章 首 飞

本章内容

- ▶ 编译和连接
- ▶ 创建第一个项目
- ▶ 端口初始化
- ▶ 重测 PORTA
- ▶ 测试 PORTB

每一个飞行学员的首飞通常都是模糊的回忆，充满着一系列短暂而强烈的感受，这包括：

- 初次起飞的冲击，尽管是由教练操作的；
- 在听了教练“能开车的人就能开飞机”的观点后，若要保持飞机直线飞行几分钟，依然会有嘴唇发白、手心冒汗的感受；
- 急性运动性眩晕，当教练回到驾驶位置执行降落任务，进行那个容易引起晕机的“侧滑”动作时，让人觉得跑道就要穿过侧面的机窗了。

对于每一个刚迈进嵌入式编程世界的新读者来说，第 1 章也会带给你类似的感觉。

1.1 飞行计划

每一次飞行都是有目的的，最好一开始就准备一份飞行计划。

本书的第一个项目是使用 16 位的 PIC24 微控制器。对于部分读者而言，可能会使用 MPLAB IDE 集成开发环境和 MPLAB C30 语言套件来开始第一个项目。即使读者以前从来没听说过 C 语言，也可能知道著名的“Hello World”编程例子。如果连这个例子也不知道，那么下面就加以简单地介绍。

第一本 C 语言的书是由 Kernighan 和 Ritchie 在几十年前编写的。从那时起，每一本正统的 C 语言书籍都会介绍一个在计算机屏幕上显示“Hello World”字样的示例程序。就算没有上千本，那么也应该有上百本的书籍延续了这个传统，本书也不例外。不过，本书会略有一些区别。实际上，本书之所以讨论可编程微控制器，是因为要设计嵌入式控制器应用。尽管可以假定任何个人电脑或工作站都会用到显示器，但在嵌入式控制领域这个假定并不成立。因此，本书的第一个嵌入式应用将关注更基础类型的输出端口——数字 I/O 引脚。在后面更深入的章节中，将会介绍 LCD 显示器和其他连接到串行口的终端设备。当然，到时将会给出比“Hello World”更有趣的实验。

1.2 飞前备忘录

每一次飞行都会有飞前检查——绕着飞机走走，检查油缸里的汽油以及机身上的机翼等。因此，这里应首先检查所需的组件是否已经准备和安装好（可从 Microchip 网站 <http://www>。

microchip.com/mplab 下载最新版本):

- ❑ MPLAB IDE, 免费的集成开发环境;
- ❑ MPLAB SIM, 软件仿真器;
- ❑ MPLAB C30, C 编译器 (免费的学生版)。

接下来, 要按照下面这个“创建新项目”备忘录, 使用 MPLAB IDE 创建一个新项目。

(1) 选择“Project→Project Wizard”激活新项目向导, 它会通过下面的步骤自动引导读者操作。

(2) 选择 PIC24FJ128GA010 器件, 单击 Next。

(3) 选择 MPLAB C30 编译器套件, 单击 Next。

(4) 创建一个新文件夹, 命名为“Hello”。将项目命名为“Hello Embedded World”, 单击 Next。

(5) 简单地单击下一个对话框的 Next——因为不需要从以前的项目或目录下复制任何源文件。

(6) 单击 Finish, 完成向导设置。

由于是第一个项目, 所以还需加上以下的步骤。

(7) 打开一个新的编辑窗口。

(8) 输入以下三行注释:

```
//
//   Hello Embedded World!
//
```

(9) 选择“File→Save As”, 将文件保存为“Hello.c”。

(10) 选择“Project→Save”, 保存项目。

1.3 飞行

现在要开始编写代码了。读者可能会不知所措, 尤其是从来没有使用 C 语言编写过嵌入式控制应用程序代码的读者。第一行代码应该是:

```
#include <p24fj128ga010.h>
```

这并不是严格的 C 语句, 但是这一伪指令会告诉编译器在继续运行程序之前, 先读取设备说明文件的内容。设备说明“.h”文件仅仅是一个长长的列表, 指明了所选 PIC24 模型中所有内部特殊功能寄存器 (SFR) 的名字和长度。如果 include 文件正确, 则文件中寄存器的名字恰恰是设备数据表上正在使用的。如果还有疑问, 可以打开文件查看——这是一个可以用 MPLAB 编辑器打开的简单文本文件。下面是 p24fj128ga010.h 文件的部分程序段, 其中定义了程序计数器和一些其他的特殊功能寄存器 (SFR):

```
...
extern volatile unsigned int   PCL __attribute__((__sfr__));
extern volatile unsigned char PCH __attribute__((__sfr__));
extern volatile unsigned char TBLPAG __attribute__((__sfr__));
extern volatile unsigned char PSVPAG __attribute__((__sfr__));
```

```
extern volatile unsigned int RCOUNT __attribute__((__sfr__));
extern volatile unsigned int SR __attribute__((__sfr__));
...
```

返回到“Hello.c”源文件，加上几行，定义函数 main()：

```
main()
{
}
}
```

尽管函数体是空的，不会执行任何操作，但它现在已经是一个完整的 C 语言程序了。在那两个花括号里面，很快将放入实现嵌入式控制应用的一些指令。

不管函数 main() 出现在什么位置，无论是在最开始的几行，还是在一个几十万行的文件的最后几行，它都标志微控制器（程序计数器）在上电复位或者其他复位后程序开始运行的位置。

需要注意的是，在进入函数 main() 之前，微控制器会执行连接器自动插入的一个较短的初始化代码段。这个代码段又被称作 c0 码。c0 码将实现基本的例行内务处理，包括微控制器栈的初始化以及其他事务。

现在的任务是启动一个或多个 I/O 引脚，即端口 A 的引脚 RA0~RA7。在汇编语言中，可以使用一些 mov 指令来将字面值（literal value）传送给输出端口。在 C 语言中，操作会变得更简单，即直接写入如下面例子中的“赋值语句”：

```
#include <p24fj128ga010.h>

main()
{
    PORTA = 0xff;
}
```

首先要注意的是，每一个 C 语句都是以分号结束的。另外，C 语句同数学方程很相似，不过它不是数学方程！

赋值语句的右边会先被计算。所得的结果（在这个例子中仅仅是一个字面值常量）会被保留，然后传送到左边的接收容器。在这个例子中，接收容器是微控制器的一个 16 位特殊功能寄存器（已在 .h 文件中定义过）。

注解 在 C 语言中，如果字面值前面有 0x，则表明它是十六进制数。否则编译器将假设其为默认的十进制数。相似地，0b 表示二进制数，而因为历史的原因，单个的“0”表示八进制数。（现在还有人使用八进制数吗？）

1.3.1 编译和连接

现在，我们已经写出了 main() 函数，也是第一个 C 程序的唯一的函数。下面，怎样把源程序转换成可执行的二进制代码呢？

那就使用 MPLAB 集成开发环境 (IDE) 吧, 它很容易上手! 只要用鼠标轻轻一点就可以啦。这个操作被称作项目构建。一系列又长又复杂的工序主要由以下两大步骤组成。

- ❑ 编译: 激活 C 编译器, 并生成目标代码文件 (.o)。这个文件暂时还不能执行。当大部分代码生成时, 所有的函数和变量的地址仍然是未定义的。实际上, 这又叫作可重定位的代码目标。如果有多个源文件, 对每个文件都会重复地执行这个步骤。
- ❑ 连接: 激活连接器, 并在内存空间中为每个函数和变量分配适当的位置。同时, 任意数量的预编译器目标代码文件和标准库函数都会在这个时候根据需要增加进来。连接器生成的几个输出文件实际上都是二进制可执行文件 (.hex)。

以上的这些步骤, 在读者单击项目 (Project) 菜单的选项 “Build All” 后, 就会以很快的速度执行完成。

如果选择命令行界面, 读者会欣喜地发现, 除了使用 MPLAB IDE 外, 还可使用其他的方法来激活编译器和连接器, 从而获得同样的结果, 不过, 还需要查找 MPLAB C 编译器的用户指南上的指令。本书的后续章节将一直使用 MPLAB IDE 界面和适当的备忘录来简化操作。

为了让 MPLAB 知道哪个 (或哪些) 文件需要编译, 应该把它 (们) 的名字 (在本例中是 Hello.c) 添加到项目的源文件列表 (Source Files List) 中。

为了使连接器能正确地为每个变量和函数分配地址, 需要向 MPLAB 提供指定设备的 “连接器脚本” 文件 (.gld) 的名称。正如 include (.h) 文件是用来告诉编译器指定设备的特殊功能寄存器 (SFR) 的名字 (和大小) 一样, 连接器脚本 (.gld) 文件是用来告诉连接器内存的预定义位置 (由设备数据表决定) 和提供基本的内存空间信息, 如闪存的可用空间大小、RAM 存储器的可用空间大小及其地址范围。

连接器脚本文件是一个简单的文本文件, 可以使用 MPLAB 编辑器来打开和检查。

下面是 p24fj128ga010.gld 文件的程序片段, 定义了程序计数器和一些特殊功能寄存器的地址:

```
...
PCL          = 0x2E;
_PCL         = 0x2E;
PCH          = 0x30;
_PCH         = 0x30;
TBLPAG       = 0x32;
_TBLPAG      = 0x32;
PSVPAG       = 0x34;
_PSVPA       = 0x34;
RCOUNT       = 0x36;
_RCOUNT     = 0x36;
SR           = 0x42;
_SR          = 0x42;
...
```

1.3.2 构建第一个项目

首先来回顾一下完成第一个演示项目所需的几个步骤。

- (1) 把当前源文件添加到 “Project Source Files (项目源文件)” 列表。

此时有 3 个备忘录可供选择，分别对应于使用 3 种不同的方法来实现相同的功能。在这里第一次用到的是：

- (a) 打开项目窗口，如果还没打开，选择“View→Project”；
 - (b) 使用编辑窗口的光标，单击右键激活编辑器的弹出（pop-up）菜单；
 - (c) 选择“Add to project”（添加项目）。
- (2) 把 PIC24 的“连接器脚本”文件添加到项目中。

按照下面这个“把连接器脚本添加到项目”备忘录来完成以下步骤。

- (a) 右键单击项目窗口中的连接器脚本列表；
- (b) 选择“Add file”（添加文件），浏览并选择 MPLAB 子目录 support/gld 下的“p24fj128ga010.gld”文件。

现在，项目窗口应该类似于图 1-1 所示。

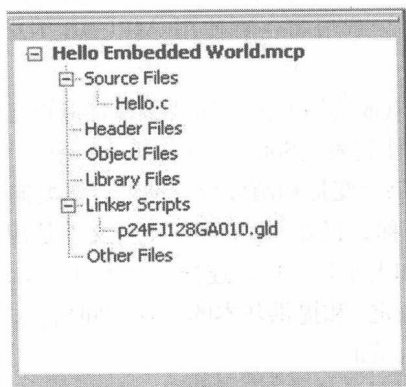


图 1-1 “Hello Embedded World”项目的 MPLAB IDE 项目设置窗口

- (3) 选择“Project→Build”功能，依次观察 C30 编译器和连接器的运行、生成的可执行代码以及 MPLAB IDE Build 窗口里的一些有用信息。

注解 “构建项目”备忘录包含的另外几个步骤，对于以后执行更加复杂的实例是很有帮助的。（如图 1-2 所示。）

- (4) 选择“Debugger（调试）→Select Tool（选择工具）→MPLAB SIM（MPLAB 仿真器）”，选择并激活仿真器作为本节的主要调试工具。注意：这个“MPLAB SIM 调试器设置”备忘录会提示读者如何正确地配置调试器。

如果一切正常，在试运行代码之前，还要先打开 Watch（监视）窗口，选择并添加 PORTA 特殊功能寄存器（输入或者在 SFR 组合框中选择 PORTA，然后单击“Add SFR”按钮）。（如图 1-3 所示。）