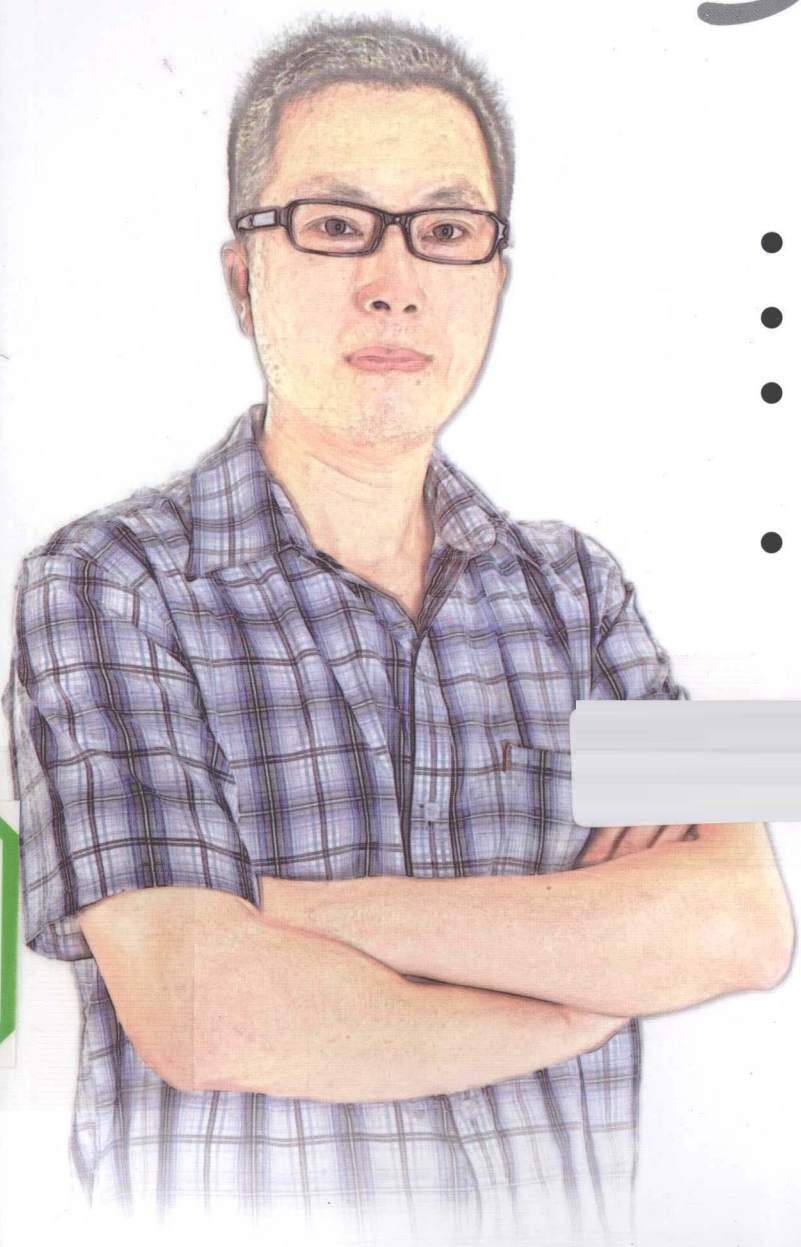


嵌入式产品分析 与设计



- 注重实际能力养成
- 涵盖产品分析与设计全流程
- 全面结合了当代嵌入式系统项目开发必须掌握的要素
- 让嵌入式产品设计变得简单

王真星 著

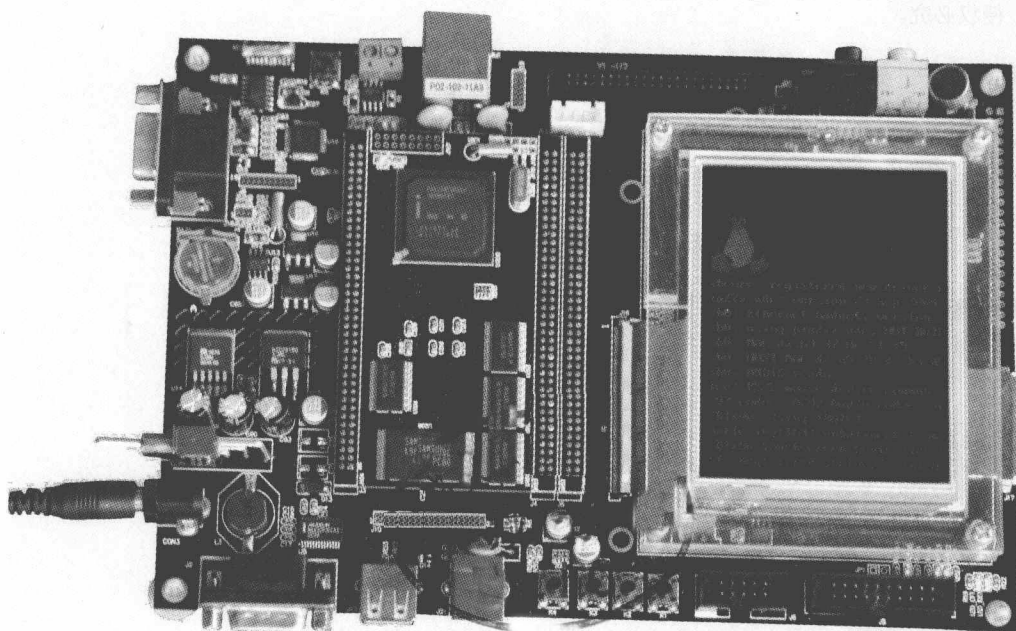


电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

嵌入式产品分析与设计

王真星 著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以多个嵌入式产品的开发为主线,遵循从简单到复杂的开发理念,使读者能全面了解和掌握近年来嵌入式产品开发的主要内容,具体包括软件和硬件的开发方法。本书分为两部分:第一部分包括第1~3章,主要介绍单片机产品开发;第二部分包括第4~9章,主要介绍ARM产品开发。

本书对工程技术人员的产品开发具有很好的参考价值。

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

嵌入式产品分析与设计 / 王真星著. —北京:电子工业出版社, 2013.10

ISBN 978-7-121-20618-4

I. ①嵌… II. ①王… III. ①微型计算机—系统设计 IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2013)第 120139 号

策划编辑:王敬栋(wangjd@phei.com.cn)

责任编辑:徐 萍

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

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:24 字数:614 千字

印 次:2013 年 10 月第 1 次印刷

印 数:4 000 册 定价:68.00 元

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 zltz@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

嵌入式产品设计对实践的要求非常高，只学习理论而不实际动手，所学的知识就无法内化为自己真正的、能运用自如的技能。嵌入式产品设计要求设计人员必须熟练掌握模拟电子技术、数字电子技术、单片机技术、计算机接口技术、计算机软件技术、操作系统原理、电力电子技术、自动控制技术方面的知识，并且要有非常丰富的实践经验。要学会复杂嵌入式产品的设计，需要一步一步经历从简单到复杂，不断犯错误、不断修正错误的过程。只有走过弯路、犯过错误，才能意识到问题，才能真正掌握这方面的知识。嵌入式产品的更新非常快，技术几年一变，只有不断地研究分析最新产品，不断地学习，才能设计出符合要求的、在市场上有竞争力的产品。

本书的特点在于：特别注重读者的实际能力，通过对多个已经量产的产品的分析和设计，使工程技术人员熟悉开发稳定的、能大批量生产的产品的整个流程。为了设计先进、稳定、可靠的产品，设计产品之前就需要了解市场的同类产品是如何实现的，按照价格约束确定要设计的产品，包括如何选择元件、如何批量采购、如何布板、如何与贴片代加工工厂配合，等等。元件的选择除了要了解价格，还要选择市场上的主流产品，而不要选择不容易买到的产品。如果元件太贵，则要研究能否用替代的方案降低成本。成本核算始终要贯穿于设计的全过程。除了产品稳定外，只有把成本控制在合理的范围内，所设计的产品才算是成功的。

考虑到产品的大批量生产背景，本书以消费电子产品的开发为主线，涵盖了嵌入式产品设计的主要内容，从简单到复杂，介绍嵌入式产品分析、设计和生产的全过程；从需求分析到线路图纸的设计，到 PCB 的设计，再到元件清单制作、贴片加工、软件设计、操作系统移植，直至系统在实际环境下的正确性验证，以及针对产品的知识产权保护等，对企业研发新产品具有很好的参考价值。

当前嵌入式产品开发人员的实际情况是：一部分具有硬件背景，还有一部分具有软件开发背景。对于前者，他们缺乏软件项目开发的知识，对操作系统、软件工程、面向对象、C++、网络通信、服务器、数据库等不太了解，因此在工作中不大会用到这些东西，许多还停留在 C 语言甚至汇编语言阶段；对于后者，他们不太了解计算机硬件，不会设计原理图，不会画电路板。因此，目前对软、硬件都比较熟练的开发人员较为稀缺。许多单位购买现有的评估板，然后在现成的平台上开发，最后在开发板的基础上重新布线，方案设计无法做到最优，产品自然缺乏竞争力，使嵌入式系统“量身定制”的目标大打折扣。本书的目的就在于力图改变这种局面，使开发人员同时掌握软件和硬件开发技术，开发出受市场欢迎

的、高质量的电子产品。

本书的内容如下：

第1章 讲解与 CPU 接口密切相关的外围数字和模拟输入/输出电路的设计，是设计嵌入式产品的必备知识。

第2章 带领读者设计一个简单的足部按摩器产品。本章首先分析现有样品，在功能分析的基础上带领读者进行开发，包括从硬件和软件的开发到器件的选择、焊接加工等大批量生产的整个流程。

第3章 在掌握了简单产品开发的基础上，深入一步介绍乒乓自动发球机的开发。与第2章相比，这个产品用到了主机和遥控器，因此一个产品包含两部分的设计，总体难度有所提高。

第4章 在学习了单片机的产品开发后，要进入 ARM 的产品开发，但两者差别较大。从单片机到 ARM 并非立即就能切入，因此本章对 ARM 的相关知识进行了介绍，重点是开发环境和启动代码。

第5章 在具备 ARM 的基础知识以后，本章介绍了一些外设接口。这些外设是开发产品时最常用的，必须熟练掌握。

第6章 在对国外拖地机器人产品样品进行硬件和功能分析的基础上，讲解用 ARM Cortex 的 STM32 开发没有操作系统的拖地机器人产品。本章主要介绍拖地机器人的基础硬件和软件设计，并在最后给出了拖地机的专利撰写样例。通过对本章的学习，读者能够熟悉目前基于 ARM 控制器产品开发的大致流程。

第7章 讨论 Linux 下的文件系统。本章为基于 Linux 操作系统的产品开发做好准备。

第8章 介绍嵌入式 Web 开发的相关知识，对 Web、CGI、数据库、Socket 通信及 HTTP 编程进行介绍，并给出如何融合这些技术进行实际远程控制的方案。

第9章 在前两章的基础上，介绍基于嵌入式 Linux 系统的家庭智能网关的开发，同时还提供了此产品的专利申请案例。

本书的出版得到上海第二工业大学校级重点学科（培育）建设项目“计算机应用技术 XXKPY1301”的资助。感谢刘中原、石林祥、陈林为本书的出版提供的帮助，徐小方对本书写作提供的支持。这里要特别感谢我的母亲，她听到我要把这么多年来开发的产品系统地写为书稿，认为此乃利益人民大众之事，非常支持。

本书作者的联系邮箱为 superwang2002@hotmail.com，读者也可访问 <http://www.armv.cn> 进行技术讨论。

作者

目 录

第 1 章 把好两头是关键——输入与输出基础知识	1
1.1 输出	2
1.1.1 最简单的 I/O 输出	2
1.1.2 感性负载对 I/O 输出的影响	3
1.1.3 隔离输出	4
1.1.4 输出电压匹配	6
1.2 输入	7
1.2.1 输入上拉电阻的确定	7
1.2.2 模拟输入的问题	9
1.2.3 模拟输入的等效网络	10
1.2.4 模拟小信号处理	11
第 2 章 最简单的例子——足疗机控制器设计	14
2.1 功能需求分析	15
2.1.1 人机输入分析	15
2.1.2 运行模式分析	16
2.1.3 电路板组成部分分析	17
2.1.4 足疗控制器功能划分	20
2.2 基于成本约束的控制电路板原理图设计	20
2.2.1 成本控制相关的考虑	20
2.2.2 CPU 控制板的设计	21
2.3 CPU 控制板的 PCB 设计	22
2.3.1 PCB 及贴片的成本考虑	22
2.3.2 焊接要求	23
2.4 价格成本核算	24
2.5 控制器开关电源设计	26
2.5.1 线性电源的设计	26
2.5.2 开关电源设计基本原理	27
2.5.3 用 PI Expert 设计足疗机的开关电源变压器	30
2.5.4 开关变压器的设计	37

2.5.5	控制集成电路部分	40
2.5.6	开关电源的 PCB 设计	44
2.6	控制器其他部分设计	45
2.7	给 PCB 代工厂提交的资料	48
2.8	足疗机软件设计	51
2.8.1	红外通信设计	52
2.8.2	PWM 产生	56
2.8.3	定时扫描显示	57
2.8.4	键盘处理程序	61
2.8.5	间断运行模式程序	62
2.8.6	定位程序	64
2.8.7	主控子程序	65
2.8.8	主控程序	70
第 3 章	更进一步——乒乓发球机产品设计	74
3.1	需求分析	75
3.1.1	遥控器需求分析	76
3.1.2	主板需求分析	77
3.2	硬件功能设计和实现	79
3.2.1	落点的实现	79
3.2.2	发球个数的实现	81
3.2.3	红外接收的实现	83
3.2.4	供球电动机正反转和调速的实现	84
3.3	主控板硬件原理图设计	90
3.4	红外遥控发射硬件设计	94
3.4.1	遥控器硬件要求分析	94
3.4.2	液晶的选择	94
3.4.3	遥控器主板设计	95
3.4.4	遥控器红外发射的调制	97
3.4.5	遥控器的外观	98
3.5	软件设计规划	99
3.5.1	合理安排中断优先级	99
3.5.2	主控程序总体结构	101
第 4 章	质的飞跃——从单片机到 ARM 产品开发	110
4.1	嵌入式系统和 ARM	111
4.1.1	JTAG 接口	112
4.1.2	JTAG 标准	112
4.1.3	JTAG 硬件控制器	113
4.2	JTAG 仿真器制作	114

4.2.1	ARM 的调试结构	114
4.2.2	JTAG 仿真环境	114
4.2.3	自制简易仿真器	115
4.2.4	JTAG 仿真器硬件制作	115
4.2.5	JTAG 仿真器驱动软件	120
4.3	ADS 开发套件	124
4.3.1	在 ADS 1.2 中使用简易 JTAG 仿真头调试	124
4.3.2	ADS 中程序的调试	128
4.4	ARM 启动代码和 Bootloader	130
4.4.1	启动代码主要构成	131
4.4.2	启动代码实例分析	133
4.5	从 ADS 1.2 到 Realview MDK	140
4.5.1	工具结构的改进	140
4.5.2	分散加载文件	142
4.5.3	C 库函数的差异	143
4.5.4	开发环境迁移实例	143
第 5 章	ARM 常用外设接口	149
5.1	SPI 接口	150
5.2	模块式 LCD 的 SPI 接口设计	151
5.2.1	128×64 点阵显示器	151
5.2.2	模块引脚说明	152
5.2.3	模块式 LCD 硬件连接	154
5.2.4	模块式 LCD 内部存储器	155
5.2.5	SPI 接口 LCD 显示程序	158
5.3	SPI Flash Memory 编程	164
5.3.1	SPI Flash 硬件接口	164
5.3.2	AT45DB041 的软件接口函数	166
5.4	I ² C 接口	170
5.4.1	上拉电阻与传输速率	171
5.4.2	I ² C 总线三种信号	172
5.4.3	软件仿真 I ² C 示例	174
5.4.4	带 I ² C 硬件控制器的程序	179
第 6 章	基于 STM32 的室内导航家用拖地机	181
6.1	对 mint5200 进行拆解	183
6.2	设计方案	185
6.2.1	外观的修改	185
6.2.2	机械设计	186
6.2.3	室内导航方案的选择	187

6.2.4	导航系统方案设计	188
6.2.5	红外通信方案设计	192
6.2.6	保证直线行走的设计方案	194
6.2.7	方案的合理性分析	195
6.3	总体设计	197
6.4	硬件设计	198
6.5	软件设计	204
6.5.1	关于 STM32 固件库	204
6.5.2	异常信号的处理	204
6.5.3	电动机控制部分	206
6.5.4	红外和噪声波载波发生	208
6.5.5	PID 电动机速度控制	210
6.5.6	MPU6050 陀螺仪及姿态解算	214
6.5.7	有关清洁覆盖算法分析	216
6.5.8	规则动作库	220
6.6	拖地机产品样机	221
6.7	拖地机专利撰写举例	222
第 7 章	基于 OS 层面 ARM 必备知识——嵌入式 Linux 文件系统	227
7.1	Linux 文件系统简介	228
7.1.1	ext2 和 INODE	229
7.1.2	虚拟文件系统 (VFS)	230
7.2	注册文件系统	233
7.3	安装文件系统	233
7.4	在虚拟文件系统中搜寻文件	235
7.5	卸载文件系统	235
7.6	/proc 文件系统	235
7.7	设备特殊文件	236
7.8	常见的 Flash 文件系统	236
7.8.1	Flash 的特点	236
7.8.2	JFFS2	238
7.8.3	YAFFS 文件系统	240
7.9	根文件系统	240
第 8 章	将设备联网——嵌入式 Web Server 的实现	245
8.1	Web 基础知识	246
8.1.1	HTTP 协议	246
8.1.2	HTTP 请求	246
8.1.3	HTTP 应答	247
8.2	面向电子商务的 B/S 结构	248

8.3	Web Server 制作网页	249
8.4	CGI 工作原理	250
8.4.1	环境变量	251
8.4.2	CGI 标题和 GET/POST	252
8.4.3	CGI 程序的开发	254
8.4.4	几种常用数据库接口	255
8.4.5	几种常用 CGI 及其 Web 开发语言	255
8.5	JavaScript 脚本	257
8.5.1	JavaScript 的语句及语法	259
8.5.2	JavaScript 编程举例	262
8.6	socket 通信	264
8.6.1	TCP Socket 编程举例	266
8.6.2	UDP Socket 编程举例	271
8.6.3	HTTP 请求中 Client 与 Server 的交互过程	275
8.6.4	一个简单的 Web 服务器例子	276
8.7	嵌入式 Web 服务器 Boa 的特点	280
8.7.1	Boa 的功能实现	281
8.7.2	Boa 的移植步骤	284
8.7.3	CGIC 库的移植	286
8.7.4	HTML 模板的制作	287
8.7.5	一个综合的 Web 测试实验	288
8.8	通过网络远程控制开发板上的灯	301
第 9 章	基于 Linux 的家庭网关设计	306
9.1	产品开发背景	307
9.2	功能需求	307
9.3	家庭网关设计	309
9.3.1	网络通信设计	310
9.3.2	网关软件架构	311
9.3.3	关于视频硬件设计	312
9.3.4	系统整体的硬件设计	312
9.3.5	家庭网关系统包含的软件	313
9.4	硬件平台设计	315
9.5	嵌入式 Web 开发概述	317
9.5.1	Mizi Linux	317
9.5.2	Boa 小型 Web 服务器	317
9.5.3	SQLite 轻型数据库	318
9.5.4	网关的软件平台构造	318
9.6	软件环境搭建步骤	319

9.6.1	烧写 Mizi Linux	319
9.6.2	搭建交叉编译环境	319
9.6.3	Boa 移植	319
9.6.4	SQLite 移植	321
9.6.5	SQLite 使用方法与常用命令	322
9.6.6	建立数据库	323
9.7	CGI 程序设计与实现	326
9.7.1	CGI 与客户端的通信机制	326
9.7.2	程序中读写 SQLite 数据库	330
9.7.3	RS-485 串口读写	333
9.7.4	指令格式定义	337
9.7.5	CGI 脚本举例	340
9.8	前台网页设计与实现	342
9.8.1	HTML 简介	342
9.8.2	CSS 简介	344
9.8.3	JavaScript 简介	347
9.8.4	Ajax 技术	350
9.9	网关使用说明书	355
9.9.1	产品概述	355
9.9.2	用户登录	356
9.9.3	主界面	356
9.9.4	家电控制	357
9.9.5	系统设置	360
9.10	家庭网关产品的知识产权保护问题	365
参考文献		371

第 1 章

把好两头是关键——输入与输出基础知识

很多嵌入式学习者对 CPU 和编程很熟悉，但对输入和输出的处理并不是十分清楚，这里首先来谈谈如何正确处理输入和输出，因为输入和输出是处理器与外界交互的主要部分，处理不好会直接影响到产品性能和系统稳定性。

本章主要学习如何对输入和输出电路进行处理，输入和输出电路其实与不同的外界条件密切相关。同样的电路，在一种外界条件下是合理的，但在另外一种外界条件下却可能是不合理的。通过本章的学习，读者将掌握以下几个关键知识点：

- 三极管饱和导通式开关输出；
- 感性负载的输出；
- 高频 PWM 输出；
- 隔离式输出；
- 不同供电电压芯片之间的输出匹配；
- 普通数字输入电路；
- 模拟小信号的输入；
- 差分输入电路。

1.1 输出

1.1.1 最简单的 I/O 输出

CPU 的 I/O 输出用来给外设提供信号, 或者驱动外部执行设备。因此, 合理设计 I/O 输出电路是产品稳定的必要条件。输出电路的设计需要考虑驱动功率大小、频率、噪声等因素。以下首先介绍最基本的开关量输出, 条件是输出开关频率很低。图 1-1 所示为最简单的驱动电路。

图 1-1 是最简单的驱动电路, 由单片机出来的 OUT 信号通过 R1 驱动 VT, R1 和 R2 的关系要按照 VT 所处的开关状态计算。也就是说, VT 只有两种状态: 断开与饱和导通, 不允许出现放大状态。假设电压 V_{CC} 为 10V, 电阻上承受的功率超过 2W。如果三极管处于放大状态, 我们假设通过调节 R1 让三极管处于 c、e 之间有 5V 的压降, 这样流过 R2 的电流是 5/47 安培 (A)。三极管的功率为 0.53W, 由于 8050 三极管没有散热装置, 因此在短时间内将很快发热, 最后发烫, 通过热辐射方式散热, 如果安装在封闭机箱中, 很快就会烧毁。

图 1-1 中给定负载 R2, 其实是要计算 R1, 根据 $I_c = \beta \cdot I_b$, 饱和情况下, VT 的压降约为 0.2V, $I_c \approx (V_{CC} - 0.2)/R_2 \approx 0.209A$ 。假设放大倍数 β 为 60, I_b 取 3.5mA, 单片机输出的 OUT 电压为 4.8V, 那么计算出通过 R1 的电流是 $(4.8 - 0.7)/R_1 = 0.41mA$ 。因此图 1-1 的设计不合理, 在给定负载情况下, 要求通过 R1 的电流至少是 3.5mA, 而实际只有 0.41mA, 会导致三极管烧毁。那么如何调整呢? 可以通过降低 R1 的阻值, 增大 I_b 实现。现在要求至少 3.5mA, 因此 $R_1 = (4.8 - 0.7)/0.0035 = 1.171k\Omega$, 考虑裕余量, 设计 R1 的阻值为 1k Ω 。

以上的前提条件是开关动作不频繁, 如果每 1s 内连续执行 10 次, 则上述线路还是有问题的。问题出在什么地方呢? 实际是由三极管从截止状态到饱和状态的过程必须经过放大状态引起的。因为经过放大状态是不可避免的, 所以为了降低损耗, 要尽量减少在放大状态停留的时间。如果输出的频率很高, 三极管同样会发热烧毁。图 1-2 所示为三极管从截止到饱和的过程, 特别要注意的是其中的转换过程。

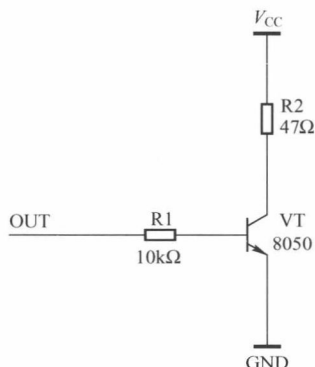


图 1-1 最简单的驱动电路

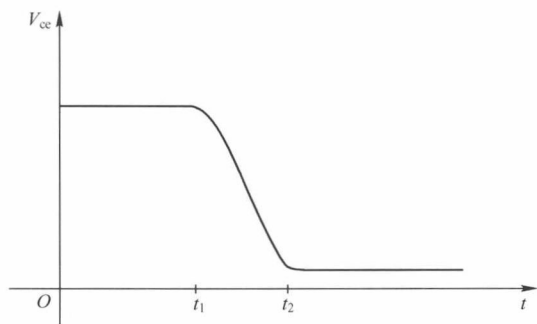


图 1-2 三极管从截止到饱和的过程

由图 1-2 可知, t_1 和 t_2 之间是三极管处于放大的状态, 其时间长短与目前三极管输出的电流大小, 输入的电流大小, b、e 之间的电容大小都有关系, 时间越长, 对输出管越不利。对于输出频率高、电流大的场合, 除了选用适合设计参数的晶体管外, 还需要在电路上进行辅助的设计, 用来尽可能减少变换的时间。也就是让信号变化期间变得陡直。

为了让信号变换显得陡直, 需要从三极管的特性进行分析。三极管的集电极载流子受基极电流的影响, 只要增大开通时的基极电流就能加快转换, 如图 1-3 所示。可以看到, 增加电容 C1 后, 当 OUT 是高电平时, 由于 C1 电压不能突变, 将迅速通过 C1 向 VT 提供基极电流, 这时的基极电流主要通过 C1 提供, 随着时间的推移, C1 充电后承受电压, 基极电流将通过 R1 提供。

以上电路虽然解决了转换时间问题, 但带来的副作用是对 CPU 输出引脚的冲击较大, 因为开通瞬间 VT 的基极回路阻抗基本是 VT 自身的 R_{be} , 一种优化的方式是给 C1 串联电阻。

图 1-4 所示是改进后的基极驱动电路, 其中, R3 降低了开通瞬间的冲击。但是 R3 的选择必须合适, 过大无法起到 C1 应有的作用, 过小会带来冲击。

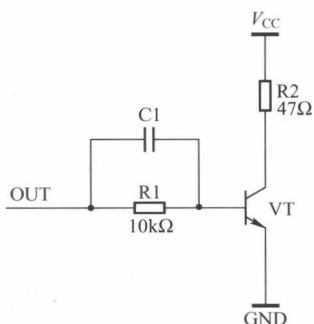


图 1-3 加快转换的方法

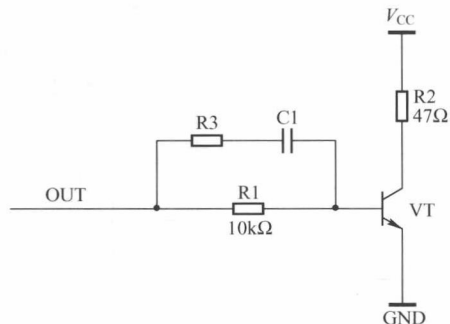


图 1-4 改进的基极驱动电路

1.1.2 感性负载对 I/O 输出的影响

前面提到的是单片机对电阻性负载的驱动问题。对于电感性负载, 情况又是怎样的呢? 图 1-5 中 VT 驱动了一个继电器, 会有什么情况发生呢?

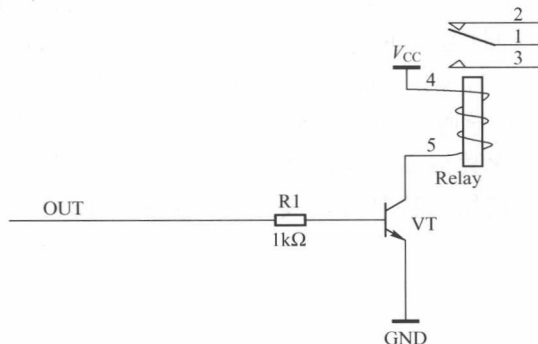


图 1-5 驱动感性负载

一开始可能很正常,但过了一段时间,不知道什么原因,VT 坏了。可能有人认为是 VT 质量不好,换一个 VT 修好了,以后也可能正常了很长一段时间,最后 VT 还是会损坏。原因就在于继电器线包其实是电感,当断开时有一个高压加在 VT 上面,非常容易击穿损坏 VT。因为电感的电流不能突变,开通的时候没有问题,电流是慢慢上升的,当 VT 关断时,电感电流不能突变,根据 $u=L \cdot di/dt$ 可知当电感量 L 一定时,VT 的关断导致电流突然消失, di/dt 很高,电压 u 也就很高,方向是上负下正,它和电源本来的电压 V_{CC} 叠加到三极管 VT 上,极有可能超过 VT 能够承受的击穿电压,导致 VT 击穿。因此,改进思路是让 VT 关闭时电感电流不会发生突变。从图 1-6 可以看到,当 VT 关闭后继电器通过 D1 续流,不会引起高压。

但是,如果 OUT 输出频率很高,图 1-6 是有问题的。

如图 1-7 所示,输出需要驱动一个直流电动机,如果直流电动机只是处于开与关两种状态,则没有问题。但如果需要对电动机进行降压调速,那么问题就来了。

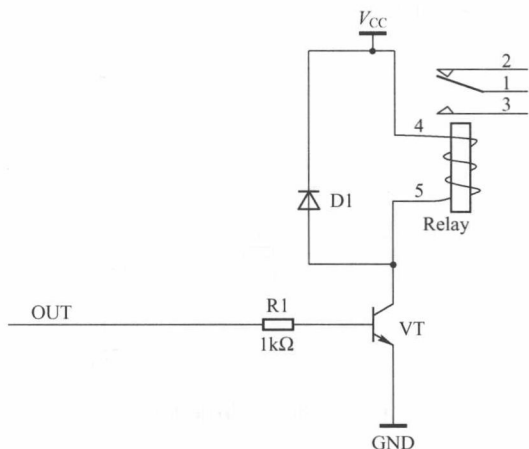


图 1-6 电感并接续流管

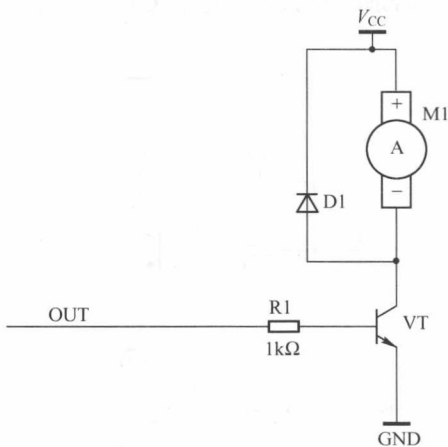


图 1-7 电动机驱动

假设电动机 M1 只是做实验用的小电动机,电流为 200mA 左右,图 1-7 是可以用的。但假设驱动的电动机电流在 1A 以上,并且 V_{CC} 是 15V,这时情况就大不一样。D1 必须仔细选择,否则会烧毁 VT。原因是 D1 若选用普通二极管,则反向恢复时间很长。假设频率是 100kHz,反向恢复时间将非常重要。当电动机在调速运行时,如果 VT 关闭,电流通过 D1 续流,然后 VT 导通,由于 D1 存在反向恢复时间,不能马上截止。这就严重影响了 VT,当 VT 导通时负载等效阻抗很小,VT 的电流很大。由于开关频率很高,这个阶段重复出现的次数很多,导致 VT 发热烧毁。为此,必须选择快恢复的二极管。在低压大电流场合,选用肖特基管,其反向恢复时间很短;在高压场合,则选择快恢复管。

1.1.3 隔离输出

当输出需要控制强电的时候,要采用隔离。根据设计要求的不同,隔离方式有光电隔

离和继电器隔离。继电器隔离主要是开关量，需要控制外设的启停场合。光电隔离主要用于频率较高且控制频繁的场合。这里主要讲解光电隔离的设计。

图 1-8 是基本的光电隔离电路。电路分为两个部分：光耦输入部分和输出部分。它们之间通过光传输信息，不存在电气上的关系。光耦输入端和普通的发光二极管驱动完全一样，一般电流为 20mA。从输出端可以看到输出用的电源的电压大小与地和输入完全不同，通过隔离方式，图 1-8 所示电路把电压从单片机的 5V 转换成 30V 的输出，也就是 Port 端的电压是 0 和 30V 两种状态。

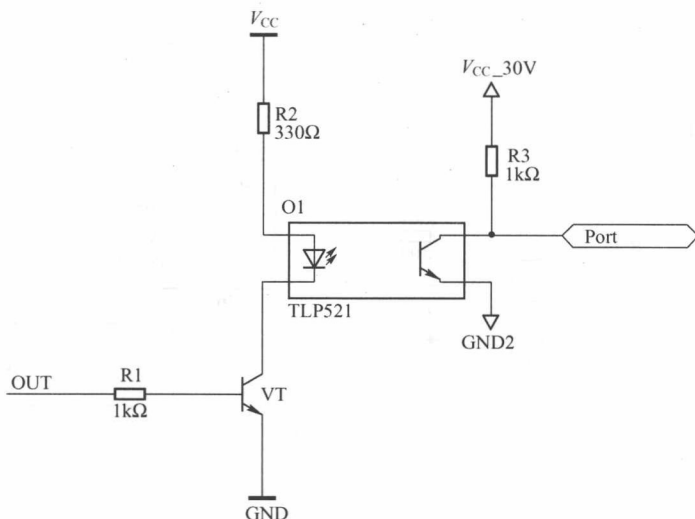


图 1-8 光电隔离基本电路

以上电路对于低频率很适合，但在频率很高的情况下就存在问题。当频率很高时，由于光耦延迟等因素，输出的波形不再完全跟随输入。

图 1-9 是光耦延迟对信号的影响。其中横坐标是时间轴，纵坐标是电压大小，下面的二维坐标表示光耦输入电压和时间的关系，上面的二维坐标表示光耦输出电压和时间的关系。当信号频率很高的时候，这个影响非常明显，会导致输出信号波形畸变。为了减小信号畸变，需要采用高速光耦，如 6N136 等。

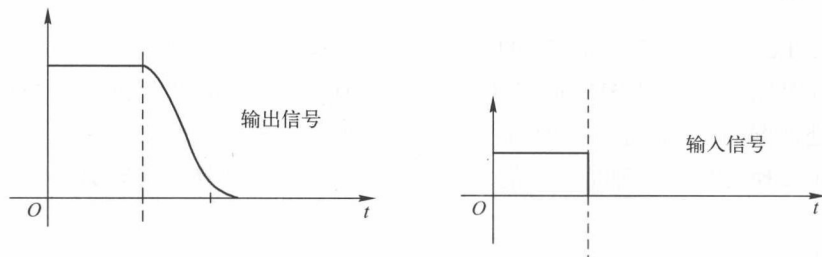


图 1-9 光耦延迟影响

在中央处理器的隔离通信应用中，处理器必须与外界电气隔离。如图 1-10 所示，通信采用 RS-485，波特率要求为 115 200bps 这里如果用 TLP521 低速光耦，肯定无法满足要求，信号会发生畸变，与远程机器无法正常收发。图中采用了 6N136 高速光耦，能顺利解决高速通信的问题。其中有一个 PC817 是低速光耦，是用来对通信方向进行切换的，也就是让通信处于发送或者接收状态，无须用高速光耦，为的是降低成本。

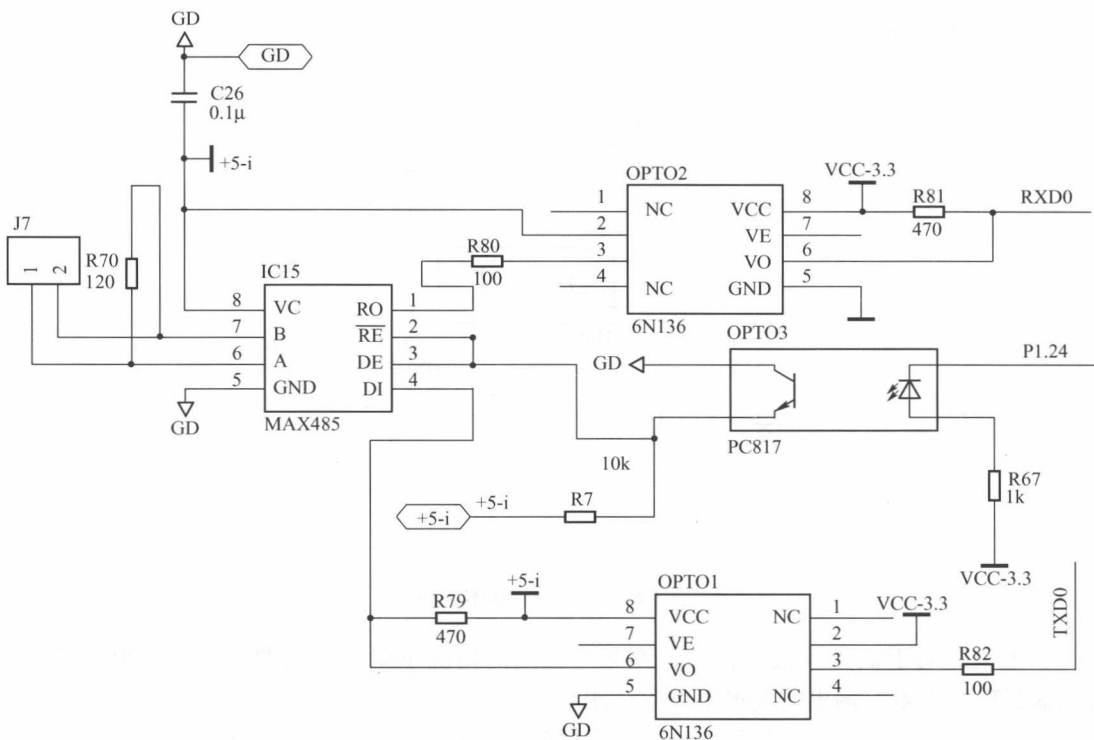


图 1-10 一个高速光耦隔离通信的例子

1.1.4 输出电压匹配

目前很多 PCB 上是 3.3V 和 5V 的器件并存，在输出时如何将两者的电压匹配呢？这里分别讲述从 3.3V 到 5V 和从 5V 到 3.3V 器件之间的连接。图 1-11 是通过三极管进行电压变换，将左边处理器的 3.3V 输出变换后，在 IC1 变为 5V 输入。如果不用三极管，也可以选用专门的电压转换芯片，原理是一样的。而 5V 到 3.3V 的转换比较容易，直接通过电阻分压即可。

输出讲完后，下面介绍输入的处理。