第1章 大转折——从学习单片机到应用单片机

如何从单片机的学习飞跃到单片机的应用呢?需要两个条件:一是掌握一定水平的单 片机知识和电子技术基础知识,二是具有勇于实践的精神。

单片机的应用是以单片机应用系统的开发为前提的。如图 1-1 所示为一个单片机应用 系统——踢足球机器人,其内部的单片机能够检测到机器人在球场中的坐标和球的位置, 并控制机器人按照既定的比赛计划完成移动和控球等动作。明显地,要利用单片机制作一 个踢足球机器人,需要进行大量反复的设计和调试。



图 1-1 踢足球的机器人(图片来自 wikipedia.org)

要建造房屋,就需要有一定数量的砖头,同样,设计单片机系统就需要掌握一定数量 的功能模块。如踢足球机器人的运动需要电机来驱动,其功能模块就是电机;机器人之间 需要通信,其功能模块就是无线通信。只有学习了这些功能模块,才能拥有"砖头"去完 成整个系统。本书的目的就是把"砖头"交到将要进行单片机系统开发的朋友手上。

在本章中,首先来看单片机的学习与应用之间有什么区别与联系。本章的内容有:

- ◆ 如何利用单片机
- ◆ 哪些工具可以帮助我们
- ◆ 调试乃成功之母
- ◆ 实例点拨——无线鼠标

1.1 如何利用单片机

不知道如何使用单片机、如何利用单片机进行系统设计,大概是对将要设计的系统功能和结构不明确造成的。因此,在利用单片机进行任何设计之前,需要有一个十分清晰的系统功能和结构的概念。

例如,要设计一个防盗用的玻璃打碎报警器,系统的功能明确为:当报警器检测到玻

璃被打破的声音时,驱动警灯发光、警铃报警。如图 1-2 所示,当报警器的话筒接收到玻 璃打破的、异于平常的尖锐声音时,单片机便对这种声音进行检测,一旦确认为玻璃破碎 的声音,则让警灯和警铃发出警示信号。



图 1-2 玻璃打碎报警器

明确系统是检测到玻璃破碎的声音便进行报警的功能之后,就要思考系统的结构。系统结构讨论的是系统的功能模块和模块之间的关系。在报警器中,系统的功能模块包括:

◆ 话筒。将声音信号转换成电信号,这是信号采集模块。

◆ 单片机。分析声音信号,判断是否为打碎玻璃的声音,这是信号处理分析模块。

◆ 警灯和警铃。发出警报,这是报警的执行模块。

有了功能模块,并掌握了接口方法,就可以规划它们之间的关系。它们之间的关系呈现在纸上就是系统的结构框图,如图 1-3 所示(放大、A/D 转换的内容在后面的章节会有详细介绍)。



图 1-3 玻璃打碎报警器的系统结构框图

在如图 1-3 所示的系统结构框图中,话筒把声音信号转换成电信号,经过放大器的放 大、A/D 转换器的数字化后输入单片机中进行处理和分析,如果判断出是玻璃破碎的声音 信号,则驱动警灯和警铃发出警报。

根据系统结构框图,就可以开始设计实际系统的电路原理图(简称电路图)。电路图 是对结构框图进一步的细化,用实际的电子器件取代结构框图中的模块,并使用导线具体 化模块之间的关系,如图 1-4 所示。对照图 1-3 可以发现,电路图与系统框图是对应的, 从左到右被虚线分成话筒、放大器、A/D转换器、单片机、警灯和警铃等电路模块。



图 1-4 玻璃打碎报警器的系统电路图

有的读者感觉图 1-4 太复杂,并惊讶它是怎么设计出来的,这种感觉表明这些读者对 单片机系统设计是颇有兴趣的。本书将会用大量的笔墨来介绍如何设计类似图 1-4 的单片 机系统电路图,目前只需要把系统框图和电路图结合起来了解一下即可。

注意,系统框图(图1-3)的单片机模块中的白色字体的注释为:信号处理分析,控制 报警,这是单片机将要完成的任务之一。想象一下,当报警器工作时,环境中有许多噪声, 如马路上的汽车喇叭、人的说话声、小猫/小狗的叫声、他人轻轻敲击玻璃的声音等,这些 声音与玻璃被打碎时的"噼里啪啦"响声是截然不同的,如果使用示波器观察图 1-4 中的 节点 T 的信号,可以比较一般背景噪声(如图1-5(b)所示)和玻璃被打碎时的声音的波 形(如图1-5(a)所示)。





人类经过几千万年的进化之后,已经能轻易地区别一般背景噪声和玻璃打碎声,但单 片机这个思维、思考能力不及人类大脑亿分之一的电子器件既不会听,也不会想,它怎么 进行判断呢?所以,在电路图需要有 A/D 转换器把波形数字化后输入单片机中,这些数字 化后的信号就能够被单片机接受,通过程序对信号的处理和分析,就会分辨出到底是不是 玻璃被打碎的声音,如果是,就驱动警灯和警铃报警。

单片机对信号的处理和分析以及控制报警的时机要靠程序来完成,这就是单片机系统 开发中一个重要的环节——程序开发。程序的开发可借助一些开发工具软件来完成,如 KEIL 公司的嵌入式系统程序设计软件 μVision 等。1.2 节将介绍包括 μVision 在内的可用于单片 机开发的工具。

金思考一下

 防盗用的玻璃打碎报警器的系统功能是什么?系统的功能模块有哪些?各个模块的 作用是什么?

2. 联系图 1-3 和图 1-4, 分析系统框图和电路图是如何对应的。

3.单片机能不能直接处理和分析如图 1-5 所示的信号波形?如果不能,信号应该做什 么转换后再输入单片机中?

1.2 哪些工具可以帮助我们

工欲善其事,必先利其器。要开发单片机系统,还需要借助一些工具。本节以单片机 控制1支发光二极管闪烁的系统为例,介绍单片机开发过程中需要的几种代表性工具,掌 握这些工具即可顺利完成大部分单片机系统的开发。

1. 单片机设计参考书——获取系统设计的思路

我们生活的年代离牛顿发现万有引力定律的时代已经十分久远了,现在身边有各种各 样的现成理论和实践实例可以参考,何不拿来进行修改优化并融入自己的设计理念后构建 成所需的系统?在本书中就可以找到许多有用的电路图、代码和应用实例等,这对于提高 系统开发的效率是很有帮助的。

例如,单片机控制1支发光二极管的系统比较简单,可以参考图1-6。这个电路图是单 片机最小系统,由单片机、晶振、复位、电源与外设组成,任何单片机应用系统都是在单 片机最小系统上衍生出来的。

在设计单片机系统时,难免会使用到模拟和数字电路的知识,在必要时,还可以参考 这方面的参考书,如《电子设计从零开始》(清华大学出版社,ISBN 978-7-302-11509-0)。 另外,有的读者感觉单片机的基础知识还有些欠缺,可以参考 2008 年 1 月出版的《51 单 片机应用从零开始》(清华大学出版社,ISBN 978-7-302-16247-6)一书,这本书出版一年 之内即重印,我们已经收到几百个朋友的邮件,许多好的建议已经被采用到这本书中。

初学电子技术或单片机系统设计的朋友也许不知道类似图 1-4 和图 1-6 的电路图是怎么 画出来的,这时可以参考介绍电路图设计的书籍,如《电路设计与仿真——基于 Multisim 8 和 Protel 2004》(清华大学出版社),从中能获得丰富的电路设计、仿真、电路图绘制和 印刷电路板设计方面的知识。

• 4 •

此外,本书的支持网站——电路飞翔网(www.circuitfly.com)在广大朋友的建议和帮助下,开设了单片机的学习交流平台,并不断更新其中的资料和信息,希望它能成为广大朋友学习单片机的一把"利器"。

有了以上这几本参考书和电路飞翔网站的帮助,单片机系统设计的学习就会变得轻车 熟路了。



图 1-6 单片机控制 1 支发光二极管的系统电路图

2. 单片机程序开发软件——µVision

μVision 是国际上比较流行的单片机程序开发和仿真软件,目前最新的版本是 μVision3。它可用于 C51、ARM、C16x、ST10 及 C25 等单片机的程序设计和仿真。它支 持世界主流单片机生产商的单片机程序开发和软件仿真(见附录 F),可以在网上搜索 μVision3 试用版下载试用。

软件仿真就是在软件中对单片机程序的运行过程进行观察,检查它是否按照设计目的 对外围器件进行正确的控制。

下面简单介绍如何利用 μ Vision 帮助单片机程序的开发。首先,运行 μ Vision 软件(本书以 μ Vision3 版本为例,书中均略称为 μ Vision),选择 Project 菜单中的 New Project 命令,如图 1-7 所示。

🕼 Hision3		
Ele Edt Yew Proj	ect Debug Flash Perpherals Tools 2VCS Window Help	
1200	New Project	
← → 02 (4	Import jalision1 Project Open Project 新建工程命令	
0	Glose Project	
Project Workspace	Components, Environment, Bools Select Devices for Target Target, L' Pergram Rem Options for Target Target 1'	
(*)	Quild target	17
10 A	Rebuild all target files	
×	Stop byld	

图 1-7 新建工程

 μ Vision弹出一个保存新建工程的对话框,为了方便回顾和记忆将工程命名为Prj_flash,

其中, Prj 是 Project 的缩写, flash 的意思是"闪烁", 代表是一个发光二极管闪烁的工程, 然后在硬盘上选择路径保存新建的工程文件。

µVision 随即弹出一个目标器件选择对话框,在其中选择所使用的单片机型号,如 Atmel 公司的 AT89S51 单片机。在对话框左侧的器件数据库中找到 Atmel 的文件夹由 ◆ Atmel,单 击文件夹旁边的 "+"号,从展开的树中找到 □ AT89S51,单击该图标,这时右侧出现对应 的器件描述信息,如图 1-8 所示。单击 OK 按钮,弹出一个确认对话框,提示是否把 8051 的启动代码也复制到工程文件夹中并向工程中添加一个代码文件,单击 No 按钮,此处暂时不需要向工程中添加这些代码。



图 1-8 器件选择对话框

在 μVision 左侧的 Project Workspace 窗格中生成了一个空的工程文件夹,要向其中添加单片机系统运行所需要的汇编程序。单击工具栏中的 窗按钮(或选择 File 菜单中的 New 命令),在工作区域生成了一个新的编辑窗口 Text1,如图 1-9 所示。



图 1-9 在工程中新建汇编程序文件

在编辑窗口中用键盘输入控制发光二极管闪烁的程序,如程序 1-1 所示。注意,在输

• 6 •

入程序的过程中,最好关闭中文输入法而在纯英文输入法下输入指令和标点符号,在需要 中文注释时再打开中文输入法输入,这样可以避免中文标点符号可能引发的汇编错误。

ORG OOH ; 设置起始地址 START: CPL P1.0 ; P1.0 口电平反相 CALL DELAY ; 调用延时子程序 JMP START ; 跳到 START 标号处重新开始执行 DELAY: ; 延时子程序段(500ms) MOV R3,#50 D1: MOV QINZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ; 返回主程序 END ; 汇编程序结束	程序 1-1:	单片机控制) P1.0 口上的发光	6_	こ极管闪烁(对应图 1-6)	
START: CPL P1.0 ; P1.0 口电平反相 CALL DELAY ; 调用延时子程序 JMP START ; 跳到 START 标号处重新开始执行 DELAY: ; 延时子程序段 (500ms) MOV R3,#50 D1: MOV R4,#20 D2: MOV R5,\$ DJNZ R5,\$ JDNZ R5,\$ JDNZ R3,D1 RET ; 返回主程序 END ; 汇编程序结束		ORG	00H	;	设置起始地址	
CPL P1.0 ; P1.0 口电平反相 CALL DELAY ; 调用延时子程序 JMP START ; 跳到 START 标号处重新开始执行 DELAY: ; 延时子程序段(500ms) MOV R3,#50 D1: MOV R4,#20 D2: MOV R5,#248 JJNZ R5,\$ JJNZ R4,D2 DJNZ R3,D1 RET ; 返回主程序 END ; 汇编程序结束	START:					
CALL DELAY ; 调用延时子程序 JMP START ; 跳到 START 标号处重新开始执行 DELAY: ; 延时子程序段(500ms) MOV R3,#50 D1: MOV R4,#20 D2: MOV R5,#248 DJNZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ; 返回主程序 END ; 汇编程序结束		CPL	P1.0	;	P1.0 口电平反相	
JMP START ; 跳到 START 标号处重新开始执行 DELAY: ; 延时子程序段 (500ms) MOV R3,#50 D1: MOV R4,#20 D2: MOV R5,#248 DJNZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ; 返回主程序 END ; 汇编程序结束		CALL	DELAY	;	调用延时子程序	
DELAY: ; 延时子程序段(500ms) MOV R3,#50 D1: MOV R4,#20 D2: MOV R5,#248 DJNZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ; 返回主程序 END ; 汇编程序结束		JMP	START	;	跳到 START 标号处重新开始执行	
MOV R3,#50 D1: MOV R4,#20 D2: MOV R5,#248 DJNZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ; 返回主程序 END ; 汇编程序结束	DELAY:			;	延时子程序段(500ms)	
D1: MOV R4,#20 D2: MOV R5,#248 DJNZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ;返回主程序 END ;汇编程序结束		MOV	R3,#50			
D2: MOV R5,#248 DJNZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ;返回主程序 END ;汇编程序结束	D1:	MOV	R4,#20			
DJNZ R5,\$ DJNZ R4,D2 DJNZ R3,D1 RET ;返回主程序 END ;汇编程序结束	D2:	MOV	R5,#248			
DJNZ R4,D2 DJNZ R3,D1 RET ;返回主程序 END ;汇编程序结束		DJNZ	R5,\$			
DJNZ R3,D1 RET ;返回主程序 END ;汇编程序结束		DJNZ	R4,D2			
RET ;返回主程序 END ;汇编程序结束		DJNZ	R3,D1			
END ; 汇编程序结束		RET		;	返回主程序	
		END		;	汇编程序结束	

输入完成后,单击 按钮,在保存对话框中将该汇编程序文件命名为 flash.asm,表明 是一个控制发光二极管闪烁的汇编程序。保存的路径默认为 Prj_flash 工程文件夹中,然后 单击"保存"按钮,完成汇编程序文件的保存。这时,输入的程序按助记符、操作数和注 释被μVision 自动标记了 3 种颜色。接下来把保存好的汇编程序文件添加到 Prj_flash 工程 中,方法是:单击左侧工程工作区 Project Workspace 中图标^[1] Target 1 前的"+"号,打开 目录树。接着在 Source Group 1 文件夹上单击鼠标右键,在弹出的快捷菜单中选择 Add Files to Group 'Source Group 1'命令,意思是"向源代码组中添加文件",如图 1-10 所示。



图 1-10 添加汇编程序文件到工程中去

此时,μVision 弹出添加文件对话框,如果对话框中没有出现刚才保存好的汇编程序文件 flash.asm,是因为在文件类型下拉列表框中默认显示的是*.c(C51 程序文件)文件。在文件类型下拉列表框中选择 Asm Source file (*.s*; *.src; *.a*)选项,就会出现 flash.asm 文件,如图 1-11 所示。选中 flash.asm 文件,单击 Add 按钮,完成添加。最后单击 Close 按钮,关闭对话框。





添加完成之后,在 Project Workspace 中的 Source Group 1 文件夹左侧多出了一个 "+" 号,单击 "+"号,看到 flash.asm 文件已经被添加到工程中。以.asm 为后缀的文件均为汇 编程序文件,asm 是 assembly (汇编)的缩写。

单击工具栏中的目标选项按钮▲(或选择 Project→Options for Target 'Target 1'命令), µVision 将打开一个目标选项设置对话框,如图 1-12 所示。按照图 1-12 (a)和图 1-12 (b) 所示修改 Target 选项卡中的 Xtal (MHz)参数和 Output 选项卡中的 Create HEX File 参数, 表明仿真晶振频率为 12MHz,并在汇编成功后生成.Hex 文件。这个以.Hex 为后缀的文件正 是将来下载到单片机中的执行代码文件,这是唯一能被单片机"认知"和执行的文件。稍 后将看到在工程文件夹中生成了一个以.Hex 为后缀名的执行代码文件。



(a) 设置仿真晶振频率

(b) 选中生成.Hex 文件功能

图 1-12 修改目标选项设置

单击工具栏中的按钮重建所有目标文件按钮⁴⁴(或选择 Project→Rebuild all target files 命令),对程序 1-1 进行汇编和执行代码文件的生成。如果顺利,输出窗口提示信息的最 后一行出现的是 "0 Error(s), 0 Warning(s)",说明汇编成功,如图 1-13 所示。

× +	Build target 'Target 1'开始建立目标 assembling flash.asm正在汇编 flash.asm 文件 linking	^
Output Nindow	Program Size: data=8.0 xdata=0 code=19 程序大小: 数据=8.0 外部数据=0 代码=19 creating hex file from "Prj_flash"为工程生成十六进制文件 "Prj_flash" - 0 Error(s), 0 Warning(s)工程: 0 个错误, 0 个皆告	

图 1-13 输出窗口中的提示信息

以上工作顺利完成之后,表明程序在语法上是没有问题的,可以进入 µVision 软件仿 真环境中对程序的设计进行验证。方法是:单击@按钮(或选择 Debug→Start/Stop Debug Session 命令,或按 Ctrl+F5 组合键)进入软件仿真,其中一些按钮、按钮的功能和窗口如 图 1-14 所示。



图 1-14 µVision 软件的仿真环境

单击器按钮打开逻辑分析窗口,如图 1-15 所示,参考图中的序号顺序进行设置:单击 Setup__按钮,在弹出的对话框中,输入分析对象 P1.0 后关闭对话框;然后单击 建按钮运行程 序,将会看到逻辑分析窗口指示的 P1.0 口的电平以 500ms 为间隔进行跳变,如图 1-16 所 示(调整放大/缩小按钮观察到舒服的波形),说明程序是正确的。如果在如图 1-6 所示的 单片机实际系统中运行该程序,P1.0 口上的发光二极管就会以 500ms 为间隔闪烁。

如果汇编成功, µVision 会生成以.Hex 为后缀的执行代码文件,此时,可以在 Windows

的资源管理器中找到该工程的文件夹,并在其中找到文件 **English.hex**(如果没有,可在文件夹的"详细视图"中观察哪一个文件的类型为 HEX 文件)。





这里介绍两个实现单片机硬件电路仿真的软件——Proteus 和 Multisim MCU,它们分

别是 Labcenter Electronics 公司和 National Instruments 公司开发的单片机仿真软件,其界面 如图 1-17(a) 和图 1-17(b) 所示。



(a) Proteus 的界面



图 1-17 单片机硬件电路仿真软件

这两款单片机仿真软件功能相似,可以在其中连接单片机和外围器件,以组成单片机 系统,然后向单片机加载.Hex 文件(执行代码)并运行,即可在软件中观察单片机运行的 效果。如果加载的是程序 1-1,就会看到软件中的发光二极管闪烁。

这两款软件功能十分强大,除了能对 51 单片机仿真外,还支持 AVR、PIC 等单片机的仿真。目前,Proteus 在国内比较流行,可以试着在网上搜索一下看是否有试用版可以下载试用。

通过 Proteus 或 Multisim MCU 可以完成单片机系统硬件电路的仿真,在不搭建硬件平台的情况下,就可以对所设计的单片机系统电路和程序进行评估,这种方式能极大地提高单片机系统开发的效率。

但对于初学者来说,应当多做一些实际的实验,即在面包板或万用板上通过实际连接 电子器件来搭建单片机系统,这样能遇到许多实际的问题并通过努力找到解决方法,可不 断积累经验。这种能力的培养在电子技术学习中特别重要,所以本书没有过多地讨论这些 软件的使用。当较深刻地掌握单片机并完成一定数量的实际系统之后,可以利用这些软件 来提高系统开发的效率。

4.入门级单片机硬件平台——面包板、万用板、开发板

刚开始设计单片机系统时,面包板是一个很好的帮手,可以对系统电路和程序进行验证。常用的面包板的一面有许多插孔,其每一列的上、下5个插孔是分别导通的,列与列之间则相互绝缘。使用较硬的导线或面包板专用插线可以实现电气的连接,如图1-18(a)所示。

另外一种用于电子技术实验的板子叫万用板,板上有密密麻麻、规则排列的过孔,在 板子的一侧,每个过孔对应有1个焊盘,以供实验时焊接器件和导线,如图1-18(b)所示。 在实验的初级阶段,推荐使用面包板,因为使用面包板不需要焊接,容易修改电气连接和 插拔器件。



(a)面包板用于电子技术实验 (b)万用板用于电子技术实验(图片来自 simtec.co.uk) 图 1-18 用于电子技术实验的器材

5. 系统实验——用下载线向单片机下载程序

按如图 1-6 所示的电路图在面包板上完成电路的连接,即可用下载线把单片机和计算 机连接起来,向单片机下载.Hex 文件(执行代码)。下载线可以直接买到,也可以自制, 制作和使用方法可参考附录 D。

按如图 1-19 所示完成下载连接。单片机的第 9 管脚 RST 与下载接口的 RST 口相连, 使下载线接管单片机的复位控制。按照附录 D 中介绍的过程,在计算机中使用 Atmel MCU ISP 软件加载刚才在汇编时生成的执行代码文件 Pri_flash.hex, 然后下载到单片机中。下载完 成后,拔下下载线就会看到发光二极管 D1 闪烁。



图 1-19 下载程序

许多读者在阅读《51单片机应用从零开始》一书时提到,目前许多计算机已经找不到 并口或者串口了,是否可以使用 USB 口来完成下载程序呢?答案是肯定的,我们后来研制 了一种可用于 USB 口下载的 51 单片机 ISP 下载线(USB 型),外观如图 1-20 所示。

为了方便大家使用,这个下载线由一条 USB 线和转换器组成。下载线一头插入计算机 的 USB 口,另一头插入转换器的 USB 口。而转换器下端是一个下载接口,可插入开发板

中。当然开发板还需要装配一个与下载接口兼容的插座才行。



图 1-20 51 单片机 ISP 下载线(USB 型)

关于 51 单片机 ISP 下载线(USB 型)的介绍,可参考附录 D。

一名思考一下

1. 在图 1-6 中,发光二极管 D1 与单片机的哪个 I/O 口相连?该 I/O 口输出高电平或 低电平时发光二极管的状态有什么不同?

2.请在计算机上安装一个单片机程序开发软件 μVision,根据本节的介绍学习如何用 它来开发单片机程序。

3.除了使用逻辑分析功能对 I/O 口的电平进行观察外,试着使用 μVision 仿真环境中 的菜单 Peripherals→ I/O-ports→ Port 1 来观察 P1 口的电平变化。

4. 请使用下载线把程序 1-1 下载到单片机中,观察系统的运行效果。

1.3 调试乃成功之母

1.2 节展现了一个最简单的单片机系统的设计与制作过程,其中包括了以下几个步骤。

(1)系统功能的规划: 解决系统到底实现什么功能、产生什么效果、需要什么功能部件来实现的问题。

(2) 硬件设计:把功能部件的规划体现在电路图上。

(3) 搭建硬件平台:在实际中连接器件,可使用面包板、万用板或开发板等。

(4) 软件设计: 在硬件结构基础上,设计实现系统功能的程序,并进行软件仿真。

(5)下载程序到单片机并调试:利用下载线等工具把程序下载到单片机中,并比较系统运行的效果与设计是否相符。

在单片机控制发光二极管闪烁的例子中,以上5个步骤可以用图1-21(a)~图1-21(e)来表述。



(a) 系统功能的规划:闪烁,间隔 500ms

(b)硬件设计:系统电路图



(d)软件设计:在汇编语言开发环境中编写汇编程序并仿真 (c) 搭建硬件平台:实际连接电子器件



(e) 下载程序到单片机并调试: 用下载线把程序下载到单片机中

图 1-21 系统开发过程

如果一切顺利,程序下载完成之后,就会看到发光二极管闪烁。但有时事情并不总是 那么顺利,最后运行的效果可能得不到预期的效果,如发光二极管根本没有点亮或被点亮 而没有闪烁。可以从以下几个方面防止单片机系统最终效果与设计的不符:

(1)完善的系统规划。仔细考虑系统要实现的功能,特别是在复杂系统中,功能部件 很多,如有液晶屏、键盘、传感器等,如何将它们有效地组织起来实现特定功能要在设计

之初十分明确。

(2)电路图的组织。有时应用到的电子器件,如液晶屏、缓冲器等,不一定原来使用 过,所以要认真地学习器件的使用手册,看其中有没有设计的帮助和实例。应当把系统中 所有电子器件都弄清楚之后,再把它们有机地与单片机组织起来,完成单片机系统电路的 设计。

(3)硬件的连接。无论使用面包板、万用板或其他方式连接实际电路,错误经常会出现。如原来应该连接的两个器件没有连接或错连,或焊接过程中发生虚焊等,所以连接硬件时要千万小心。

(4)程序的设计。在电路图所示的单片机与外围器件的关系上编写单片机程序。本书 有很多单片机本身和外围器件的程序供参考。完成编写后,可以在 μVision 的软件仿真功 能中先对程序进行调试。

(5)系统整体运行。程序通过仿真后,即可下载到单片机中,软件和硬件联合起来运行,看是否与设计相符。

以上 5 个环节只要忽略任何一个,系统的运行就可能达不到设计的要求。在单片机系 统开发中,如果出现错误,可以在这 5 个环节中"顺藤摸瓜",找到并排除错误。这个"顺 藤摸瓜"的过程就是调试的过程,它是系统开发成功之母。

多思考一下

1. 单片机系统设计与制作的过程有哪几步?

2.结合单片机控制 1 支发光二极管闪烁的例子,分析如何经过这几个步骤完成一个简 单的单片机系统?

3. 可以从哪几个方面入手来完成单片机系统的调试?

1.4 实例点拨——无线鼠标

鼠标作为计算机必不可少的部件,它从一开始的滚轮式发展到光电式,今天又出现了 无线鼠标,如图 1-22(a)所示。无线鼠标系统由无线鼠标和 USB 接收器组成。

无线鼠标的最大特点是鼠标不用拖着一根信号线,从而使鼠标的运动和使用更加灵活。 无线鼠标的移动、按钮按下等操作,通过鼠标内置的无线发射模块发送到接收器上,接收 器接收到数据后,通过 USB 口向计算机传递鼠标的操作信息,如图 1-22(b)所示。从图 中看到,在无线鼠标和接收器上都应用了单片机。

在无线鼠标中,集成了许多单片机系统应用的技术,如单片机与外界的射频通信技术、 鼠标光电传感器与单片机的接口技术、单片机与 USB 口接口技术等。由于现在的知识积累 还不够,这里就初步了解无线鼠标的整个设计和制作过程,当完成本书学习之后再回过头 来自制一个无线鼠标就不困难了。





1. 系统功能的规划

无论是有线或无线鼠标,它的功能都是在计算机中作为位置选择设备,通过检测鼠标的相对位移,取得二维运动的坐标并反映在计算机屏幕上。鼠标除移动外,还有左键、右键和滑轮等用于对计算机界面的操作,如图 1-23 所示。其中,鼠标底面的光电传感器用于鼠标二维运动的坐标检测。



图 1-23 鼠标的结构

无线鼠标在一般光电鼠标的基础上添加了一个射频发射模块,相应地添加了一个接收器把无线鼠标发射过来的数据通过 USB 口输入计算机。无线鼠标包括的硬件模块有:鼠标 左键、鼠标右键、滑轮按键、滑轮、光电传感器和射频发射。

除射频发射模块以外,其他模块都是可以在鼠标表面找到的。结合图 1-22 (b) 所示的 无线鼠标结构——鼠标+单片机+无线发射模块,可以规划系统结构框图如图 1-24 所示,单 片机为系统的核心,左键、右键、滑轮键、滑轮、光电传感器和射频发射共 6 个模块都在 单片机的控制下工作,电池向整个系统提供工作电源。

图 1-24 中的箭头还指明了各模块与单片机之间的关系,如左键被按下时,向单片机发送信号,是一个输入模块,箭头朝向单片机;而单片机把鼠标二维运动坐标输出到射频发

射模块上,射频发射模块是一个输出模块,箭头朝向外设,依此类推。

例如,当按下鼠标左键时,单片机接收到这个操作信号,并向射频发射模块发送指令 和数据,指令是"射频发射模块,你把数据从天线发射出去",数据则是"左键被按下", 这样,接收器就会接收到左键的按下操作信号。



图 1-24 无线鼠标系统结构框图

而接收器的结构相对简单一些,还是参考图 1-22,可以规划无线鼠标的接收器系统结构框图,如图 1-25 所示。接收器的核心仍然是单片机,该单片机与图 1-24 中的单片机不同,它具有与计算机 USB 口进行接口和"对话"的本领,这类能与 USB 口通信的单片机称为 USB 单片机,许多单片机生产商都推出了具有 USB 接口通信功能的单片机。这一部分功能比较简单,即单片机把射频接收模块接收到的无线鼠标发送来的鼠标操作信息通过 USB 口告诉计算机。



图 1-25 无线鼠标接收器系统结构框图

2. 硬件设计

将系统规划出来之后,即可开始硬件电路的设计,这是考验单片机系统工程师水平的 环节。如何选择单片机和外围器件来完成系统结构框图中的"方框",从而实现系统的运 行,在这里都需要考虑。本书正是介绍这些"方框"背后具体器件的应用知识,以及这些 器件如何与单片机组合成一个实际的应用系统。 在本小节中,无法展开介绍如何把图 1-24 和图 1-25 变成具体的电路图。但是为了满 足大家对无线鼠标制作的好奇心,这里直接给出无线鼠标和接收器的电路图,并深入浅出 地讲解电路图中蕴含的信息。如果完成本书的学习,并完成一定数量的实验,理解和设计 这样一个系统将不是什么问题。

如图 1-26 所示是无线鼠标的系统电路图。单片机之外的功能模块——鼠标按钮、射频 发射、光电传感器和滑轮都被阴影包围起来。图中具有相同网络标号的端口在电气上是导 通的,如单片机 21 管脚的网络标号为 SW_LEFT,它与鼠标按键模块中的左键的端口 SW_ LEFT 同名,所以它们之间在电气上是连接的。这样一来会发现阴影部分的功能模块都连接 到单片机的某些管脚上,这就体现了单片机是系统核心的地位。

以单片机为中心,整个系统的工作过程大致为:光电传感器检测鼠标运动的坐标并输入到单片机中,滑轮和鼠标按键如果触发,也会向单片机输入相应的信息,单片机实时把 这些信息通过射频发射模块的天线发送出去。

接收器的电路图如图 1-27 所示,鼠标运动和按键等信息被接收器的天线接收,并通过 射频接收模块的处理后经由 USB 单片机输入单片机的 USB 口。

3.搭建硬件平台

如图 1-26 和图 1-27 所示的无线鼠标的电路需要搭建出来才能进行下一步的工作。和 前面讲到的一样,可以选择面包板、万用板、开发板或自行设计印刷电路板来完成所有器 件的连接。



图 1-26 无线鼠标系统电路图



图 1-27 接收器电路图

4. 软件设计

无线鼠标系统中,需要分别设计图 1-26 和图 1-27 中的两个单片机的程序,这两个单 片机控制各自的外围器件。如图 1-26 所示的单片机根据其硬件结构,其程序流程图如图 1-28 所示。对于这个流程图,先不要着急全部理解,本书后面会逐步展开相关的知识点。



图 1-28 无线鼠标程序流程图

5. 下载程序

当程序代码完成后,需要经过反复的软件仿真并结合硬件进行调试,确定系统的运行效果与设计相符,即可将程序下载到单片机,这样整个系统的开发即完成,如图 1-29 所示。



(a) 无线鼠标





