

成绩	
----	--



课程设计说明书

题目名称：人体感应智能温控风扇设计

实训名称：单片机原理及应用课程设计

分 院：机械与电气信息学院

专 业：电气工程及其自动化

姓 名：潘智勇

班 级：电气本 1602

学 号：1603032026

指导教师：王克成

课程设计任务书

设计题目	人体感应智能温控风扇设计						
学生姓名	潘智勇	专业	电气工程及其自动化	班级	电气本1602	学号	1603032026
设计要求	(1) 人来风扇转，人走风扇停； (2) 根据外界温度不同，控制不同速度的风扇； (3) 根据设计内容，设计系统硬件结构，画出系统流程图； (4) 应用Keil软件开发应用系统并仿真调试。						
设计任务	(1) 对需求进行分析进行实验设计； (2) 编辑程序； (3) 程序调试和撰写课程设计说明书初稿； (4) 提交课程设计说明书、硬件原理图、程序。						
参考文献	[1] 陆坤. 电子设计技术[M]. 电子科技大学出版社, 1997. [2] 谢自美. 电子线路设计·实验·测试[M]. 华中理工大学出版社, 2001. [3] 吴金戎, 沈庆阳. 8051单片机实践与应用[M]. 清华大学出版社, 2003.						
各阶段时间安排、应完成的主要工作	(1) 选题准备时间: 2019年6月10日-2019年月10日 (2) 设计时间: 2019年6月11日-2019年6月14日 (3) 电路组装调试时间: 2019年6月15日-2019年6月18日 (4) 设计说明书撰写时间: 2019年6月19日-2019年6月20日 (5) 验收时间: 2019年6月21日						
<div style="display: flex; justify-content: space-between;"> <div> <p>任务下达时间: 2019年6月9日</p> <p>任务完成时间: 2019年6月21日</p> </div> <div style="text-align: right;"> <p>指导教师签字:</p> </div> </div>							

人体感应智能温控风扇设计

摘 要

随着科技的进步，人们的生活水平有了很大的提高，空调已经进入很多家庭，但是它给人们带来了方便的同时，也会病状出现，作为一种老牌的电器的电风扇具有价格便宜、摆放方便、体积轻巧等特点，我们提出了一种新型的电风扇，相对于过去的电风扇，此种电风扇更加智能化，其根据室内温度自动开启和关闭，十分方便;人性化的设计将会大大提高此种新型电风扇的市场竞争力

在设计进程中，通过查阅大量资料，巩固和加强了自己所学的专业知识，圆满完成智能感应温控风扇设计

关键词：STC89C52；液晶屏显示；PWM 控制；温度传感器；人体感应

目 录

前 言	1
第 1 章 系统前的分析	2
1.1 电路设计构思	2
1.2 设计满足的基本功能	2
1.3 芯片选择	2
第 2 章 总设计原理及其元件选择	4
2.1 设计电路框图	4
2.2 显示部分电路设计	4
2.3 检测传感器电路设计	5
2.3.1 温度传感器电路设计	5
2.3.2 人体红外传感器电路设计	7
2.4 时钟部分电路设计	8
2.5 复位部分电路设计	9
2.6 驱动电路设计	9
2.7 元件选择	10
第 3 章 软件设计	12
3.1 主程序模块的设计	12
3.2 程序语言及开发环境	12
3.3 PWM 的程序设计	13

第 4 章 成果展示	14
4.1 PCB 正面图	14
4.2 PCB 背面图	14
4.3 成果图	15
4.4 成员焊接图	15
结 论	16
设计体会	17
参考文献	18
附 录	19

前 言

单片微型计算机简称单片机，又称微控制器，是微型计算机的一个重要分支。单片机是 20 世纪七十年代中期发展起来的一种大规模集成电路芯片，是集 CPU、RAM、ROM、I/O 接口和终端系统与同一硅片的器件。20 世纪八十年代以来单片机发展迅速各类新产品不断涌现出现许多新产品，出现了许多高性能新型机种现已成为工业控制和各控制领域的支柱产业之一。由于单片机功能强、体积小、可靠性好、价格便宜等独特优点因而受到人们的高度重视并取到了一系列的科研成果，成为传统工业技术改造和新产品更新换代的理想机种，并具有广阔的发展前景。

智能感应温控风扇设计是一种基于 STC89C52 单片机设计的人体热释电感应器控制电扇开关与外界温度值触发 PWM 控制电风扇转速。电路由包括马达驱动电路，温度检测电路，人体热释电传感器电路，电源电路，按键电路，液晶屏显示电路组成。

本设计运用所学的单片机知识，将单片机与普通风扇相结合设计了智能感应温控风扇，具有显示直观、省电、精度高，智能化控制等优良特点，在家，商业广场中应用广泛。

第1章 系统前的分析

1.1 电路设计构思

本设计构思中包括硬件电路的设计和系统程序的设计。其硬件电路的设计主要包括单片机的选择、显示电路的选择与设计、按键电路的选择与设计、时钟电路的选择与设计、复位电路的选择与设计，传感器的选择，马达驱动电路的选择和设计等。系统程序的设计包括系统程序设计思路、系统资源的分配、C 语言程序的设计等。

1.2 设计满足的基本功能

- (1) 感应到人来风扇转动，人走风扇停止。
- (2) 检测到外界温度，并显示在液晶屏 1602 上
- (3) 按键实现上下限设置，设定范围为 0 到 99 度
- (4) 根据外界不同温度值，控制不同转速风扇档位 0,1,2,3

1.3 芯片选择

STC89C52 是 STC 公司生产的一种低功耗、高性能 CMOS8 位微控制器，具有 16K 字节系统可编程 Flash 存储器。STC89C52 使用经典的 MCS-51 内核。在单芯片上，拥有灵巧的 8 位 CPU 和在系统可编程 Flash，使得 STC89C52 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。

STC89C52 引脚分布图如图 1-1

1	P1.0	VCC	40
2	P1.1	P0.0	39
3	P1.2	P0.1	38
4	P1.3	P0.2	37
5	P1.4	P0.3	36
6	P1.5	P0.4	35
7	P1.6	P0.5	34
8	P1.7	P0.6	33
9	RST/VPT	P0.7	32
10	P3.0/RXD	$\overline{\text{EA}}/\text{VPP}$	31
11	P3.1/TXD	ALE/ $\overline{\text{P}}\overline{\text{R}}\overline{\text{O}}\overline{\text{E}}$	30
12	P3.2/ $\overline{\text{I}}\overline{\text{N}}\overline{\text{T}}0$	/P/SE/N	29
13	P3.3/ $\overline{\text{I}}\overline{\text{N}}\overline{\text{T}}1$	P2.7	28
14	P3.4/T0	P2.6	27
15	P3.5/T1	P2.5	26
16	P3.6/ $\overline{\text{W}}\overline{\text{R}}$	P2.4	25
17	P3.7/ $\overline{\text{R}}\overline{\text{D}}$	P2.3	24
18	XTAL1	P2.2	23
19	XTAL2	P2.1	22
20	VSS	P2.0	21

图 1-1 STC89C52 引脚分布图

第2章 总设计原理及其元件选择

2.1 设计电路框图

设计总框图如图 2-1：

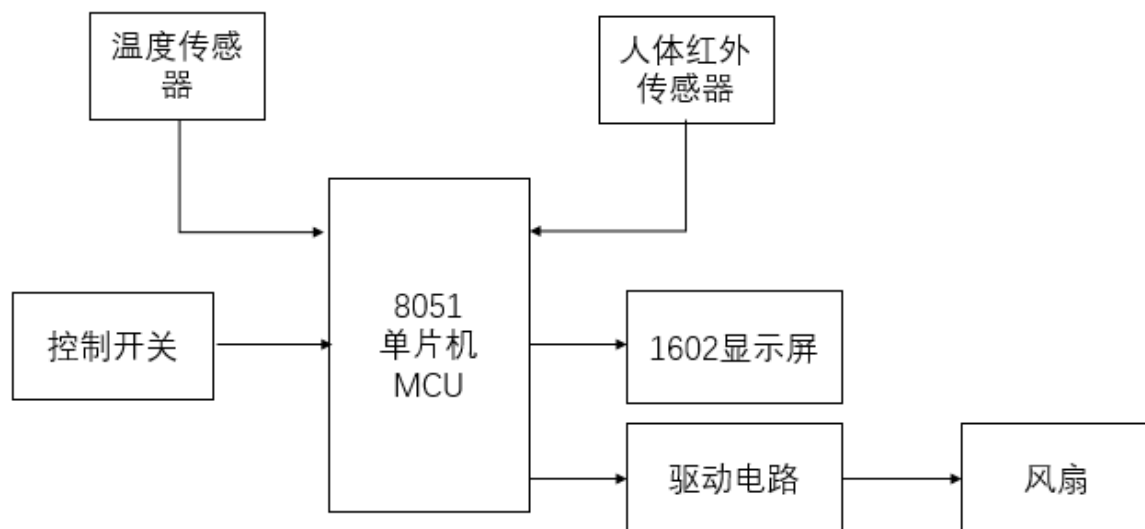


图 2-1 设计总框图

2.2 显示部分电路设计

1602 液晶也叫 1602 字符型液晶，它是一种专门用来显示字母、数字、符号等的点阵型液晶模块。它由若干个 5X7 或者 5X11 等点阵字符位组成，每个点阵字符位都可以显示一个字符，每位之间有一个点距的间隔，每行之间也有间隔起到了字符间距和行间距。

1602 显示电路管脚显示图如图 2-2 所示：

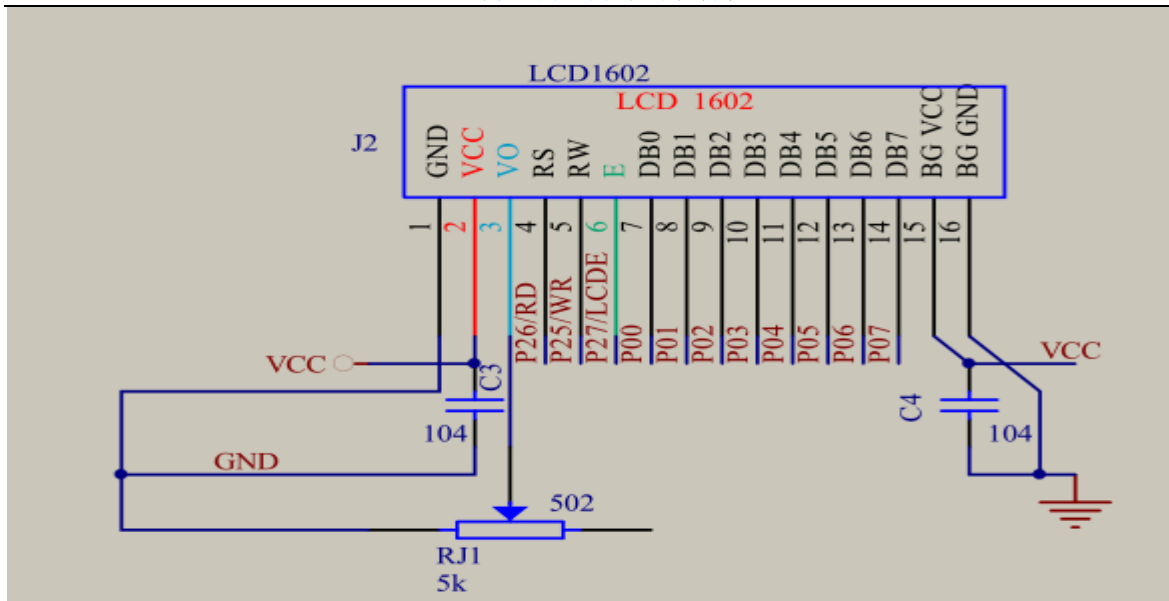


图 2-2 1602 显示电路管脚显示图

2.3 检测传感器电路设计

2.3.1 温度传感器电路设计

温度传感器的种类众多，在应用与高精度、高可靠性的场合时 DALLAS（达拉斯）公司生产的 DS18B20 温度传感器当仁不让。超小的体积，超低的硬件开销，抗干扰能力强，精度高。DS18B20 的主要特征：单总线数据通信；最高 12 位分辨率，精度可达 ± 0.5 摄氏度，12 位分辨率时的最大工作周期为 750 毫秒。可选择寄生工作方式。检测温度范围为 $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ 内置 EEPROM，限温报警功能 64 位光刻 ROM，内置产品序列号，方便多机挂接。

DS18B20 引脚排列图如图 2-3：

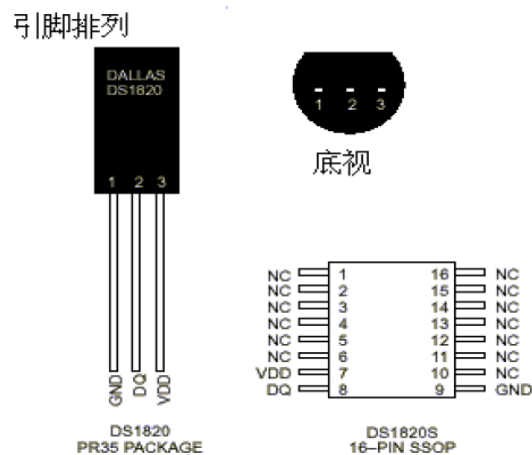


图 2-3 DS18B20 引脚排列图

DS18B20 的引脚说明如下：

GND:地

DQ: 数据 I/O

VDD :电源

DS18B20 接口电路图如图 2-4 所示

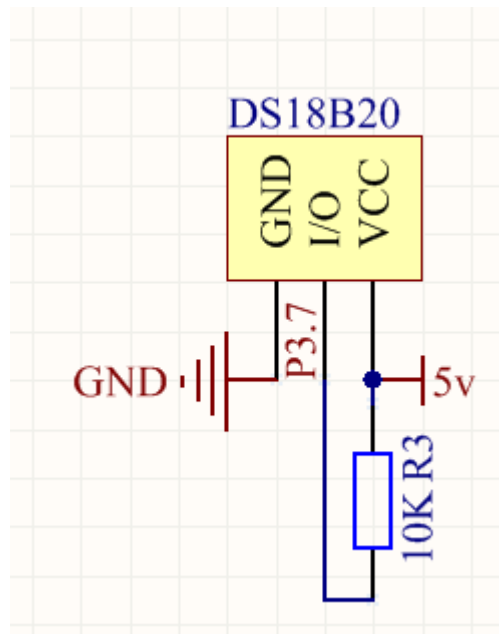


图 2-4 DS18B20 接口电路图

DS18B20 有两种供电方式，一是用电源供电的方式，此时 DS18B20 的 1 脚接地、2 脚为信号线、3 脚来接电源。另一个是寄生电源的供电方式，单片机的端口接总线，为了保证在有效 DS18B20 的时钟周期内能提供足够电流，可用一个三极管对总线进行上拉。本设计用电源供电的方式。

DS18B20 的部分温度转换值如表 2-1 所示

表 2-1 部分温度转换值

温度	输入（2 进制）	输出（16 进制）
+125℃	0000 0111 1101 0000	07D0H
+85℃	0000 0101 0101 0000	0550H
+25.0625℃	0000 0001 1001 0001	0191H
+10.125℃	0000 0000 1010 0010	00A2H
+0.5℃	0000 0000 0000 1000	0008H
0℃	0000 0000 0000 0000	0000H
-0.5℃	1111 1111 1111 1000	FFF8H

-10.125℃	1111 1111 0101 1110	FF5EH
-25.0625℃	1111 1111 0101 1110	EE6FH
-55℃	1110 1110 0110 1111	FE90H

2.3.2 人体红外传感器电路设计

HC-SR501 是基于红外线技术的自动控制模块，采用德国原装进口 LHI778 探头设计，灵敏度高，可靠性强，超低电压工作模式，广泛应用于各类自动感应电器设备，尤其是干电池供电的自动控制产品。

1、全自动感应:人进入其感应范围则输出高电平，人离开感应范围则自动延时关闭高电平，输出低电平。

2、光敏控制（可选择，出厂时未设）可设置光敏控制，白天或光线强时不感应。

3、温度补偿(可选择，出厂时未设)：在夏天当环境温度升高至 30~32℃，探测距离稍变短，温度补偿可作一定的性能补偿。

4、两种触发方式：（可跳线选择）

a、不可重复触发方式:即感应输出高电平后，延时时间段一结束，输出将自动从高电平变成低电平；

b、可重复触发方式：即感应输出高电平后，在延时时间段内，如果有人体在其感应

范围活动，其输出将一直保持高电平，直到人离开后才延时将高电平变为低电平（感应模块检测到人体的每一次活动后会自动顺延一个延时时间段，并且以最后一次活动的时间为延时时间的起始点）。

5、具有感应封锁时间(默认设置:2.5S 封锁时间)：感应模块在每一次感应输出后（高电平变成低电平），可以紧跟着设置一个封锁时间段，在此时间段内感应器不接受任何感应信号。此功能可以实现“感应输出时间”和“封锁时间”两者的间隔工作，可应用于间隔探测产品；同时此功能可有效抑制负载切换过程中产生的各种干扰。(此时间可设置在零点几秒—几十秒钟)。

6、工作电压范围宽：默认工作电压 DC4.5V-20V。

7、微功耗:静态电流<50 微安，特别适合干电池供电的自动控制产品。

8、输出高电平信号：可方便与各类电路实现对接。

9、感应距离 3 到 7 米

HC-SR501 接口电路的电路图如图 2-5

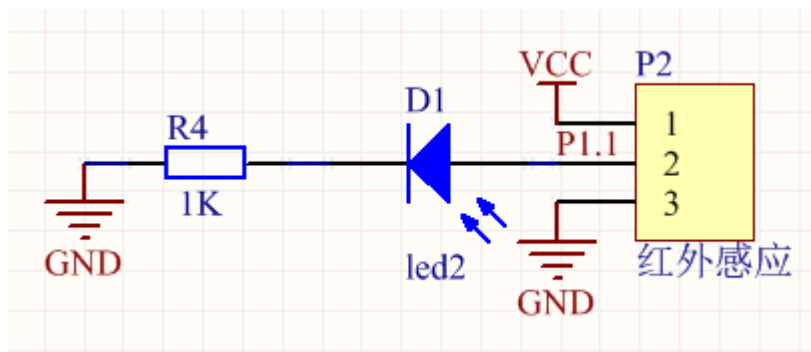


图 2-5 HC-SR501 接口电路的电路图

HC-SR501 的引脚说明如下：

GND:地

OUT: 输出脚

VDD :电源

人来 out 脚输出高电平灯亮，人走 out 脚输出低电平灯灭。

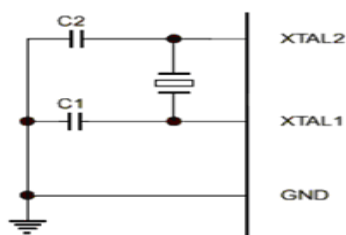
2.4 时钟部分电路设计

单片机的时钟信号用来提供单片机内各种微操作的时间基准，80C51 片内设有一个由反向放大器所构成的振荡电路，XTAL1 和 XTAL2 分别为振荡电路的输入和输出端，80C51 单片机的时钟信号通常用两种电路形式得到：内部振荡方式与外部振荡方式。外部方式的时钟很少用，若要用时，只要将 XTAL1 接地，XTAL2 接外部振荡器就行。对外部振荡信号无特殊要求，只要保证脉冲宽度，一般采用频率低于 12MHz 的方波信号。

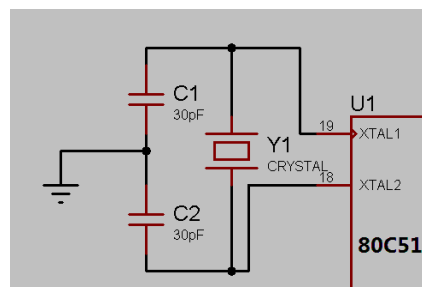
时钟发生器把振荡频率两分频，产生一个两相时钟信号 P_1 和 P_2 供单片机使用。 P_1 在每一个状态 S 的前半部分有效， P_2 在每个状态的后半部分有效。本设计采用的内部振荡方式，内部振荡方式所得的时钟信号比较稳定，实用电路中使用较多。本设计系统的时钟电路如图 2—6 所示。只要按照图 2.6 所示电路进行设计连接就能使系统可靠起振并能稳定运行。图中，电容器 C_1 、 C_2 起稳定振荡频率、快速起振的作用，电容值一般为 5~33pF。但在时钟电路的实际应用中一定要注意正确选择其大小，并保证电路的对称性，尽可能匹配，选用正牌的瓷片或云母电容，如果可能的话，温度系数尽可能低。本设计中采用大小为 30pF 的电容和

12MHz 的晶振。

时钟振荡电路如图 2-6



(a) 时钟振荡原理图



(b) 时钟振荡电路连接图

图 2-6 时钟振荡电路

2.5 复位部分电路设计

复位引脚 RST 通过一个斯密特触发器与复位电路相连，斯密特触发器用来抑制噪声，在每个机器周期的 S5P2，斯密特触发器的输出电平由复位电路采样一次，然后才能得到内部复位操作所需要的信号。

复位电路通常采用上电自动复位和按钮复位两种方式。

该设计采用加电直接复位，复位电容采用 10uF，电阻 10000 欧，为了节省元件，没有采用上电加按键模式。加电瞬间，RES 管脚为高电平。通过电阻回路放电，使电压逐渐降为零，从而实现了复位功能。

复位电路如图 2-7 所示：

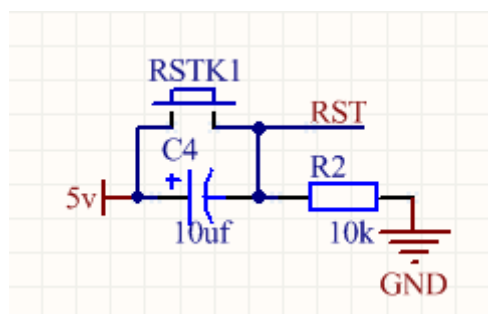


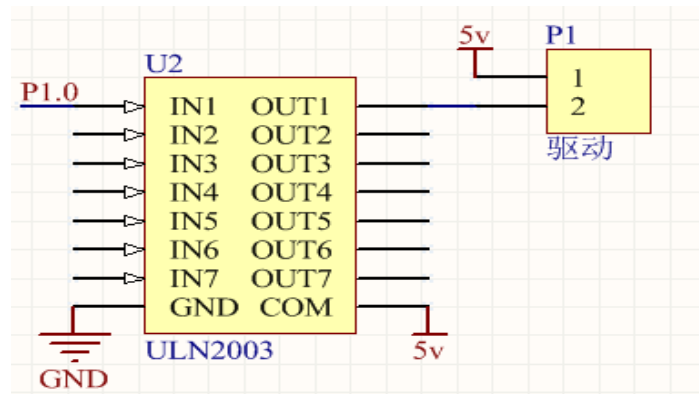
图 2-7 复位电路

2.6 驱动电路设计

ULN2003 是一个单片高电压、高电流的达林顿晶体管阵列集成电路。它是由 7 对 NPN 达林顿管组成的，它的高电压输出特性和阴极箝位二极管可以转换

感应负载。单个达林顿对的集电极电流是 500mA。达林顿管并联可以承受更大的电流。此电路主要应用于继电器驱动器，字锤驱动器，灯驱动器，显示驱动器（LED 气体放电），线路驱动器和逻辑缓冲器。本设计中驱动风扇，风扇驱动电流是 500mA，考虑到单片机引脚驱动能力，所以使 ULN2003 芯片来驱动

ULN2003 电路如图 2-8



如图 2-8 ULN2003 电路

2.7 元件选择

元件列表如表 2-2 所示：

表 2-2 元件列表

序号	名称	型号及规格	数量
1	电阻	10k	2
2	电阻	1k	2
3	LED		2
4	Uln2003		1
5	保险丝		1
6	lcd1602		1
7	轻触按键		5
8	自锁开关		1
9	晶振	12M	1
10	电位器	1k	1

11	单片机	STC89C52	1
12	瓷片电容	104	4
13	瓷片电容	33	2
14	电解电容	10uf 25v	1
15	pcb	7cm*8cm	1

第 3 章 软件设计

3.1 主程序模块的设计

主程序流程图如图3-1

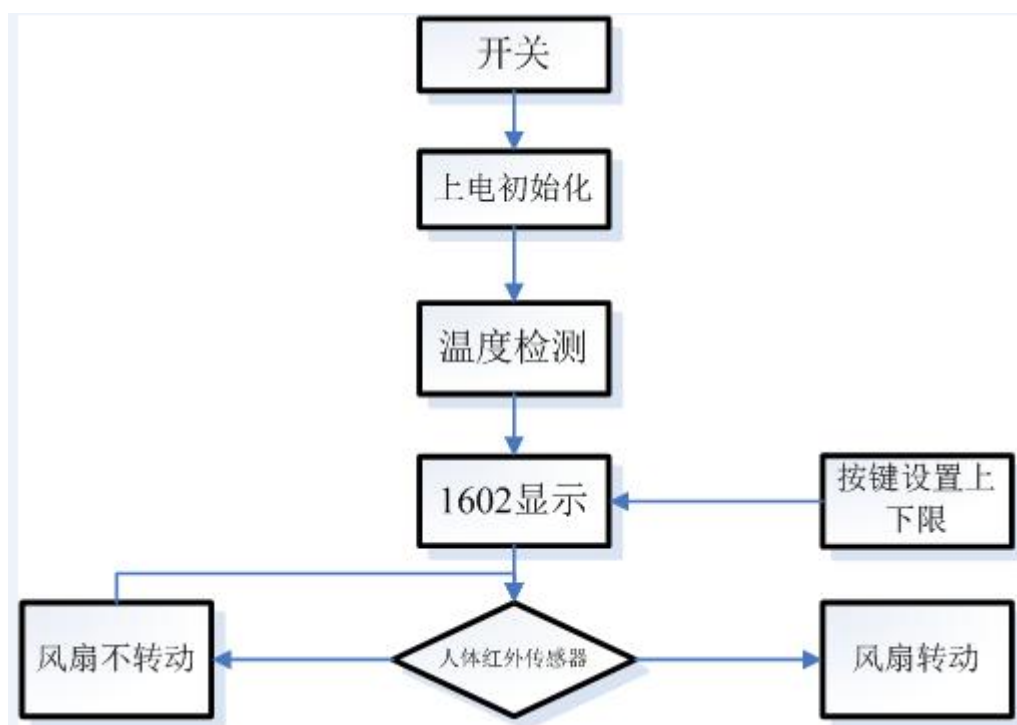


图 3-1 主程序流程图

3.2 程序语言及开发环境

C 语言是一种计算机程序设计语言，它既具有高级语言的特点，又具有汇编语言的特点。它由美国贝尔实验室的 Dennis M. Ritchie 于 1972 年推出，1978 年后，C 语言已先后被移植到大、中、小及微型机上，它可以作为工作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它的应用范围广泛，具备很强的数据处理能力，不仅仅是在软件开发上，而且各类科研都需要用到 C 语言，适于编写系统软件、三维、二维图形和动画，具体应用例如单片机以及嵌入式系统开发。

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开

发系统，与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境将这些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP 等操作系统。如果你使用 C 语言编程，那么 Keil 几乎就是你的不二之选，即使不使用 C 语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。

3.3 PWM 的程序设计

PWM 是一种对模拟信号电平进行数字编码的方法。通过高分辨率计数器的使用，方波的占空比被调制用来对一个具体模拟信号的电平进行编码。PWM 信号仍然是数字的，因为在给定的任何时刻，满幅值的直流供电要么完全有(ON)，要么完全无(OFF)。电压或电流源是以一种通(ON)或断(OFF)的重复脉冲序列被加到模拟负载上去的。通的时候即是直流供电被加到负载上的时候，断的时候即是供电被断开的时候。只要带宽足够，任何模拟值都可以使用 PWM 进行编码。

本设计中我们通过编程设置定时器溢出时间和周期来实现对 PWM 的频率设定，在通过软件设置驱动电机的高电平的占空比来实现风扇调速。

第 4 章 成果展示

4.1 PCB 正面图

PCB 正面图如图 4-1

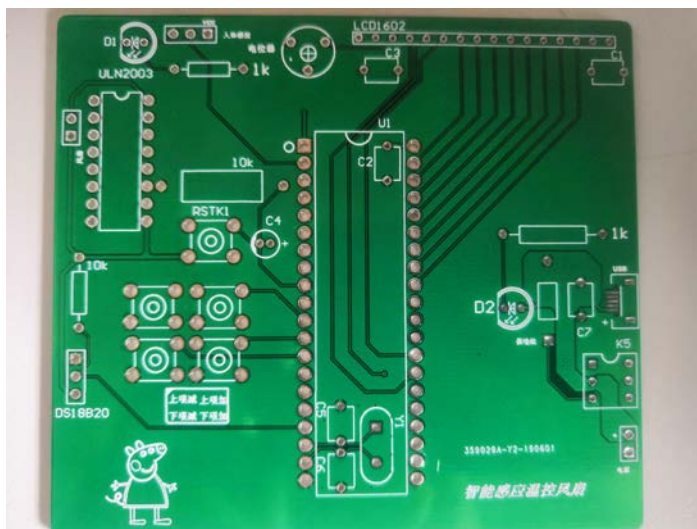


图 4-1 PCB 正面图

4.2 PCB 背面图

PCB 背面图如图 4-2

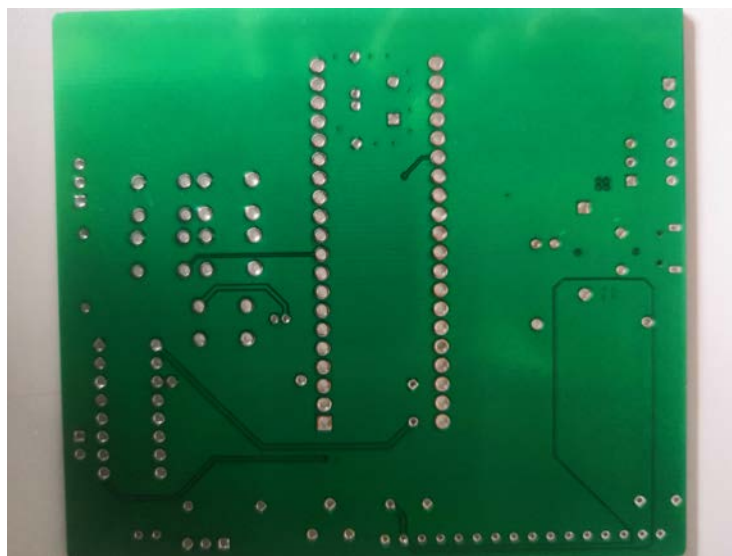


图 4-2 PCB 背面图

4.3 成果图

成果图如图 4-3

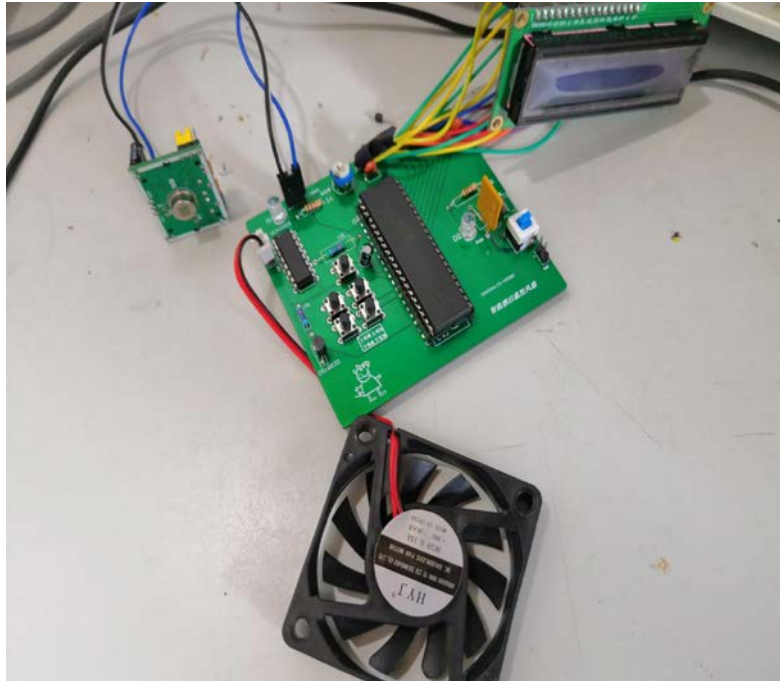


图 4-3 成果图

4.4 成员焊接图

成员焊接图如图 4-4



图 4-4 成员焊接图

结 论

本次课程设计中，重新巩固了单片机理论课时，感觉到的内容很多，知识点很杂、很繁琐。通过自己的努力也更进一步掌握了单片机的内容构造和工作原理，以及接外部电路的情况。当然光有理论知识那只是“纸上谈兵”，还需实际动手去实践。真正把所学的用到日常生活中，理论联系实际，做出实物模型。

这次单片机课程设计，我们设计的是人体感应智能温控风扇设计，通过这次课程设计我感觉到要想做成功，必须花时间多做准备，查阅大量资料，每个过程都很繁琐,硬件要找温度传感器，lcd1602，人体感应模块，按键模块的学习，软件要学 PWM 学习，1602 显示学习，都要认真地分析每一步每一个模块要实现大的功能，然后先测试在进行单步调试细微调整，我在画板子的时候不能太想当然，还要仔细比对封装大小，轻易打板子会导致反复返工，所以打板子前要认真细微。

设计体会

透过这次设计，我懂得了学习的重要性，了解到理论知识与实践相结合的重要好处，学会了坚持、耐心和努力，这将为自己今后的学习和工作做出了最好的榜样。单片机的课程设计是很有好处的，对理解单片机工程师有深一步了解，更重要的是如何把自己平时所学的东西应用到实际中。

虽然自己对于这门课懂的并不多，很多基础的东西都还没有很好的掌握，觉得很难，也没有很有效的办法透过自身去理解，但是靠着这一次课程设计，在小组同学的帮忙和讲解下，渐渐对这门课逐渐产生了些许的兴趣，自己开始主动学习并逐步从基础慢慢开始弄懂它。

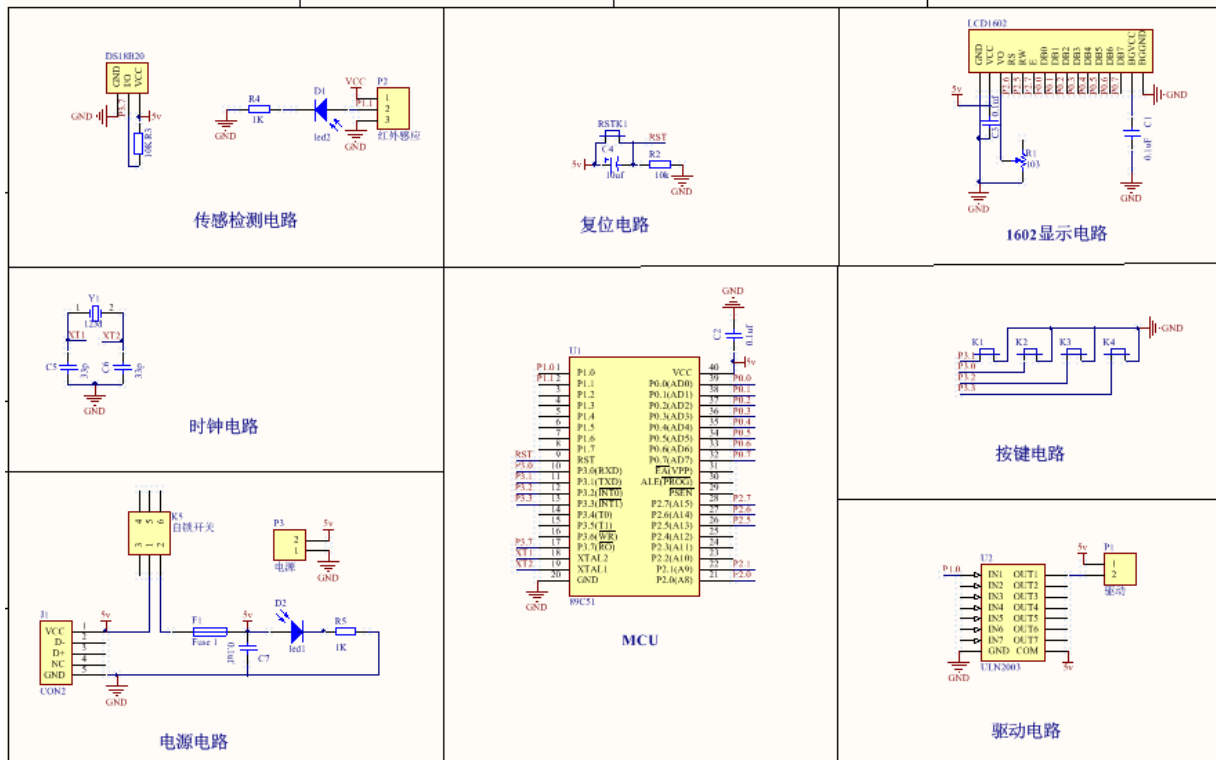
我认为这个收获就应说是相当大的。觉得课程设计反映的是一个从理论到实际应用的过程，但是更远一点能够联系到以后毕业之后从学校转到踏上社会的一个过程。小组人员的配合相处，以及自身的动脑和努力，都是以后工作中需要的。

参考文献

- [1] 胡汉才. 单片机原理及其接口技术[M]. 清华大学出版, 1996.
- [2] 蔡美琴. MCS-51 系列单片机系统及其应用[M]. 高等教育出版社, 2004.2.
- [3] 付家才. 单片机控制工程实践技术[M]. 化学工业出版社, 2004.5.
- [4] 潘新民. 微型计算机控制技术[M]. 人民邮电出版社, 1999.9.
- [5] 余锡存. 单片机原理及接口技术[M]. 西安电子科技大学出版社, 2000.7.
- [6] 雷丽文. 微机原理与接口技术[M]. 电子工业出版社, 1997.2.
- [7] 周立功. 增强型 80C51 单片机速成与实战[M]. 北京航空航天大学出版社, 2004.5.
- [8] 何立民. 单片机应用技术选编[M]. 北京航空航天大学出版社, 2004.3.
- [9] 李华. MCS-51 系列单片机实用接口技术[M]. 北京航空航天大学出版社, 1993.
- [10] 周航慈. 单片机应用程序设计技术[M]. 北京航空航天大学出版社, 1991.
- [11] 张志良. 单片机原理与控制技术[M]. 机械工业出版社, 2001.7.

附 录

总电路图如图附录-1 所示



图附录-1 总电路图

软件设计程序

```
#include "reg51.h"
```

```
#include "lcd.h"
```

```
#include "temp.h"
```

```
sbit key1 = P3^1; //设置上限加
```

```
sbit key2 = P3^0; //设置上限减
```

```
sbit key3 = P3^2; //设置下限加
```

```
sbit key4 = P3^3; //设置下限减
```

```
sbit fun_pwm = P1^0;
```

```
sbit Irstatus = P1^1;
```



```

void LcdDisplay(int);
code char NUM[] = "0123456789";
uchar string[] = "H:36 L:30"; //默认初始化的上限下限值
int wendu;
int H = 36;
int Hmax = 99;
int L = 30;
int Lmin = 0;
uchar high_pwm;
uchar pwm_t=0;
void timer0_init();
void keyross();
void main()
{
    uchar i;
    int temp;
    LcdInit();           //初始化 LCD1602
    LcdWriteCom(0x88); //写地址 80 表示初始地址第 8 个位置
    LcdWriteData('C');
    LcdWriteCom(0x40+0x80);
    for(i=0;i<9;i++)
    {
        LcdWriteData(string[i]);
    }
    while(1)
    {
        wendu = Ds18b20ReadTemp();
        LcdDisplay(wendu);
        Lcd1602_Delay1ms(30);
        keyross();
        timer0_init();
    }
}

```

```
temp=wendu*0.0625; //去看 165 行代码的注释
if(temp >= H && Irstatus == 1) //温度值超过上限， 风扇全速运行 3 挡
{
    LcdWriteCom(0x8b);
    LcdWriteData(NUM[3]);
    TR0=1;
    high_pwm = 255;
}
else if(temp >= ((H-L)/2 + L) && temp <= H && temp >= L && Irstatus ==
1)//温度值超过一半没有超过上限值， 风扇运行 2 挡
{
    LcdWriteCom(0x8b);
    LcdWriteData(NUM[2]);
    TR0=1;
    high_pwm = 150;
}
else if(temp >= L && temp <= ((H-L)/2 + L) && Irstatus == 1)//温度值超过下
限值没有超过一半值， 风扇运行 1 挡
{
    LcdWriteCom(0x8b);
    LcdWriteData(NUM[1]);
    TR0=1;
    high_pwm = 75;
}

else if(temp < L || Irstatus == 0)//温度值没超过下限值， 风扇不运行
{
    LcdWriteCom(0x8b);
    LcdWriteData(NUM[0]);
    TR0=0;
    high_pwm = 255;
```

```

        fun_pwm = 0;
    }

}

}

/*****T0 初始化*****/
void timer0_init()
{
    TMOD |= 0x01;//工作方式 1， 16 位
    TH0=0xF4;
    TL0=0xDA;
    TR0=1;
    ET0=1;
    EA=1;
}

/*****按键处理函数*****/
void keyross()
{
    if(key1==0)
    {
        Lcd1602_Delay1ms(30);
        if(key1==0)
        {
            H++;
            if(H > Hmax)//上限
            {
                LcdWriteCom(0x42+0x80);
                LcdWriteData('-');
                LcdWriteCom(0x43+0x80);
                LcdWriteData('-');
            }
        }
    }
}

```

```
        else
        {
            LcdWriteCom(0x42+0x80);
            LcdWriteData(NUM[H/10]);
            LcdWriteCom(0x43+0x80);
            LcdWriteData(NUM[H%10]);
        }
    }
    while(!key1);
}
if(key2==0)
{
    Lcd1602_Delay1ms(30);
    if(key2==0)
    {
        H--;
        if(H < L)
        {
            LcdWriteCom(0x42+0x80);
            LcdWriteData('-');
            LcdWriteCom(0x43+0x80);
            LcdWriteData('-');
        }
        else
        {
            LcdWriteCom(0x42+0x80);
            LcdWriteData(NUM[H/10]);
            LcdWriteCom(0x43+0x80);
            LcdWriteData(NUM[H%10]);
        }
    }
}
```

```
        while(!key2);
    }
    if(key3==0)
    {
        Lcd1602_Delay1ms(30);
        if(key3==0)
        {
            L++;
            if(L > H)
            {
                LcdWriteCom(0x47+0x80);
                LcdWriteData('-');
                LcdWriteCom(0x48+0x80);
                LcdWriteData('-');
            }
            else
            {
                LcdWriteCom(0x47+0x80);
                LcdWriteData(NUM[L/10]);
                LcdWriteCom(0x48+0x80);
                LcdWriteData(NUM[L%10]);
            }
        }
        while(!key3);
    }

    if(key4==0)
    {
        Lcd1602_Delay1ms(30);
        if(key4==0)
        {
```

```

        L--;
        if(L < Lmin)
        {
            LcdWriteCom(0x47+0x80);
            LcdWriteData('-');
            LcdWriteCom(0x48+0x80);
            LcdWriteData('-');
        }
        else
        {
            LcdWriteCom(0x47+0x80);
            LcdWriteData(NUM[L/10]);
            LcdWriteCom(0x48+0x80);
            LcdWriteData(NUM[L%10]);
        }
    }
    while(!key4);
}
}

```

/******: LCD 显示读取到的温度******/

void LcdDisplay(int temp) //lcd 显示

```
{
```

uchar ds18b20_datas[] = {0, 0, 0, 0, 0}; //定义显示 ds18b20 显示数

组

```
float tp;
```

```
if(temp < 0) //当温度值为负数
```

```
{
```

```
    LcdWriteCom(0x80); //写地址 80 表示初始地址
```

```
    LcdWriteData('-'); //显示负
```

```

//因为读取的温度是实际温度的补码，所以减 1，再取反求出原码
temp=temp-1;
temp=~temp;
tp=temp;
temp=tp*0.0625*100+0.5;
//留两个小数点就*100，+0.5 是四舍五入，因为 C 语言浮点数转换为
整型的时候把小数点
//后面的数自动去掉，不管是否大于 0.5，而+0.5 之后大于 0.5 的就是
进 1 了，小于 0.5 的就
//算由.5，还是在小数点后面。
}
else
{
    LcdWriteCom(0x80);    //写地址 80 表示初始地址
    LcdWriteData('+');    //显示正
    tp=temp;//因为数据处理有小数点所以将温度赋给一个浮点型变
量
    //如果温度是正的那么，那么正数的原码就是补码它本身
    temp=tp*0.0625*100+0.5;
    //留两个小数点就*100，+0.5 是四舍五入，因为 C 语言浮点数转
换为整型的时候把小数点
    //后面的数自动去掉，不管是否大于 0.5，而+0.5 之后大于 0.5 的
就是进 1 了，小于 0.5 的就
    //算加上 0.5，还是在小数点后面。
}
ds18b20_datas[0] = temp / 10000;
ds18b20_datas[1] = temp % 10000 / 1000;
ds18b20_datas[2] = temp % 1000 / 100;
ds18b20_datas[3] = temp % 100 / 10;
ds18b20_datas[4] = temp % 10;

```

```
//ds18b20 的显示数据
LcdWriteCom(0x82);      //写地址 80 表示初始地址
LcdWriteData('0'+ds18b20_datas[0]); //百位

LcdWriteCom(0x83);      //写地址 80 表示初始地址
LcdWriteData('0'+ds18b20_datas[1]); //十位

LcdWriteCom(0x84);      //写地址 80 表示初始地址
LcdWriteData('0'+ds18b20_datas[2]); //个位

LcdWriteCom(0x85);      //写地址 80 表示初始地址
LcdWriteData('.');      //显示 ‘.’

LcdWriteCom(0x86);      //写地址 80 表示初始地址
LcdWriteData('0'+ds18b20_datas[3]); //显示小数点

LcdWriteCom(0x87);      //写地址 80 表示初始地址
LcdWriteData('0'+ds18b20_datas[4]); //显示小数点
}

/*****: T0 中断函数*****/
void timer0(void) interrupt 1 using 1
{
    TH0=0xF4;
    TL0=0xDA;
    pwm_t++; //定时周期+1
    if(pwm_t == 255) //一个脉冲到了
    {
        pwm_t = 0; //定时周期重新计数
    }
    if(pwm_t <= high_pwm) //当周期和高电平占空比相等时
```



```
{  
    fun_pwm = 1;//使能 PWM  
}  
else  
{  
    fun_pwm = 0;//失能 PWM  
}  
}
```

指导教师评语：

该同学按照单片机课程设计任务书的要求，认真撰写了课程设计说明书，并且完成了单片机硬件电路的设计、组装、调试，编写了相应的程序设计，实现了基本功能，完成了课程设计的任务。

在软件设计方面还可以做的更好些，实现更加丰富的功能；在制作工艺水平上还有待进一步提高。

课程设计报告成绩：_____，占总成绩的比例 40%。

课程设计其它环节成绩：

环节名称： 考勤，成绩：_____，占总成绩的比例 20%。

环节名称： 实物验收，成绩：_____，占总成绩的比例 40%。

总成绩：_____。

指导教师签字：

年 月 日