

# 浙江大学实验报告

专业: 生物医学工程  
姓名: \_\_\_\_\_  
学号: \_\_\_\_\_  
日期: 2025.6.3  
地点: 东 1B-416

课程名称: 微机原理及其应用 指导老师: 陈星 成绩: \_\_\_\_\_  
实验名称: 智能系统设计 实验类型: 微机实验 同组学生姓名: \_\_\_\_\_

## 一、实验目的和要求

### 1. 实验目的

本实验旨在通过设计与调试温控风扇系统，深入了解 51 单片机的 C 语言编程方法及其控制系统的构成。通过实际操作，掌握如何利用温度传感器的数据来控制风扇的转速，并在数码管上实时显示温度值及风扇的 PWM 占空比。通过本实验，增强对单片机控制系统的理解，培养实践动手能力和解决实际问题的能力。

### 2. 实验要求

- 1) . 基于实验七，使用 C 语言编写程序，根据温度传感器检测到的温度响应值，在 15°C-40°C 范围内，按每升高 1°C 增加 4% 的占空比来控制风扇转速，低于 15°C 时风扇停止转动，高于 40°C 时风扇满功率转动。
- 2) . 实现温度检测值和风扇控制 PWM 的占空比在数码管上的实时显示。
- 3) . 考虑到采样数据可能含有噪声，对 AD 采样的数据进行滤波以提高数据准确性。
- 4) . 完成中断方式下的程序设计。
- 5) . 对编写的程序先在 proteus 上仿真验证，由于 proteus 里没有 XPT2046 的芯片，只需仿真数码管显示和 PWM 调速控制功能（AD 值可以给固定值）。后续用 keil 软件编译程序生成 HEX 文件，下载到开发板进行实机验证调试。

## 二、实验原理

### 1. 直流电机工作原理

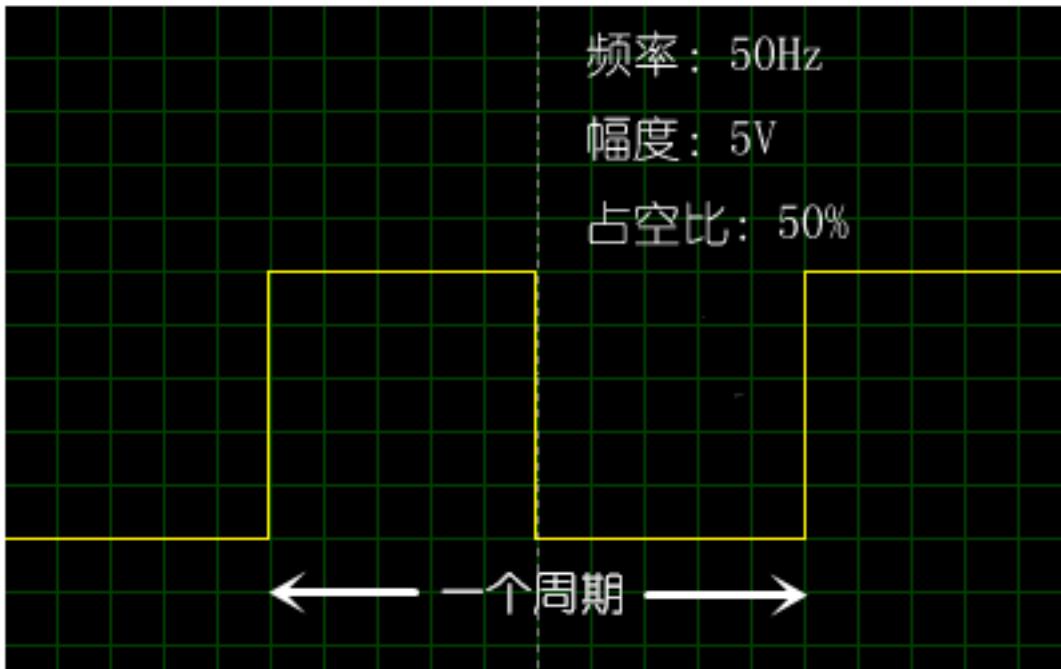
直流电机是一种将电能转换为机械能的装置。一般的直流电机有两个电极，当电极正接时，电机正转，当电极反接时，电机反转。直流电机主要由永磁体（定子）、线圈（转子）和换向器组成。

由于直流电机是大功率负载，流过的电流很大，直接连到 I/O 上会损坏单片机。因此需要配合驱动电路使用。常用的有 H 桥驱动电路（可控制电机正反转）和大功率器件直接驱动（达林顿管驱动）。开发板上面使用的是大功率驱动器件 ULN2003D 芯片（达林顿管）驱动直流电机。

### 2. PWM

脉冲宽度调制（Pulse Width Modulation, PWM）是一种通过调整脉冲宽度来控制电子设备的技术，广泛应用于电机控制、电源管理、音频信号处理和 LED 调光等多个领域。通过改变脉冲的占空比，PWM 能够在不改变电压幅值的情况下，调节输出功率，从而达到控制设备的目的。

PWM 技术的核心在于“脉冲宽度”和“周期”。占空比是脉冲宽度与周期的比值，通过调整这个比例，可以控制输出功率的平均值。例如，在电机控制中，更高的占空比通常意味着电机转速的增加。



### 3. 生成 PWM 波形原理

51 单片机生成 PWM 波形的核心在于在合适的时机操控 I/O 口翻转电平。方法 1：使用一个定时器，设置定时器以一定的时间溢出中断，在中断服务函数中进行计数，当计数值小于比较值（比较值由占空比来计算得到）时 I/O 口输出高电平，当计数值大于比较值后 I/O 口输出低电平。当计数值达到上限值后，清空计数值并再次将 I/O 口输出高电平。循环往复，便会生成 PWM 波形。改变上限值和比较值可以改变 PWM 波的周期和占空比。这种方法也被称为比较值法。

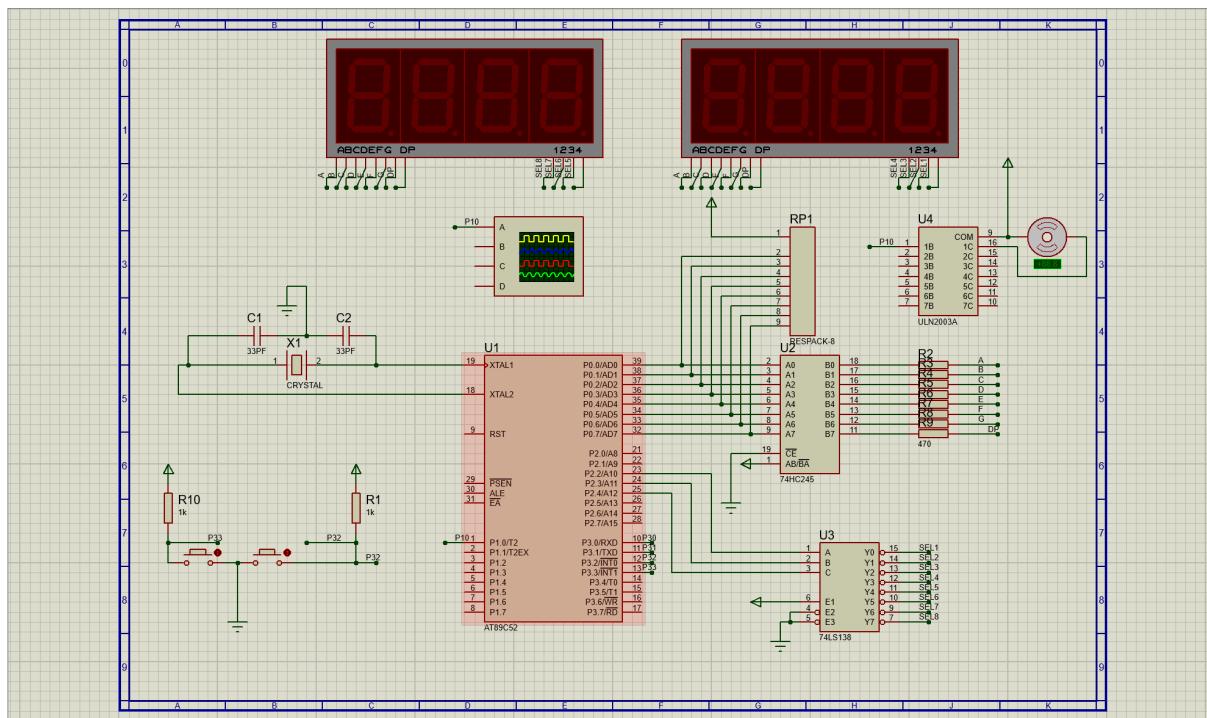
方法 2：使用两个定时器，第一个定时器的溢出时间设定为 PWM 的周期，第二个定时器的溢出时间根据 PWM 的占空比设定。当第一个定时器时间溢出中断后，在中断服务程序中将 I/O 口输出高电平，同时开启第二个定时器，等待第二个定时器产生溢出中断后，将 I/O 口输出改为低电平，并关闭第二个定时器。循环往复，便会产生 PWM 波形。改变第一个定时器和第二个定时器的溢出时间可以改变 PWM 波形的周期和占空比。这种方法也被称为双定时器法。

## 三、实验设备及器材

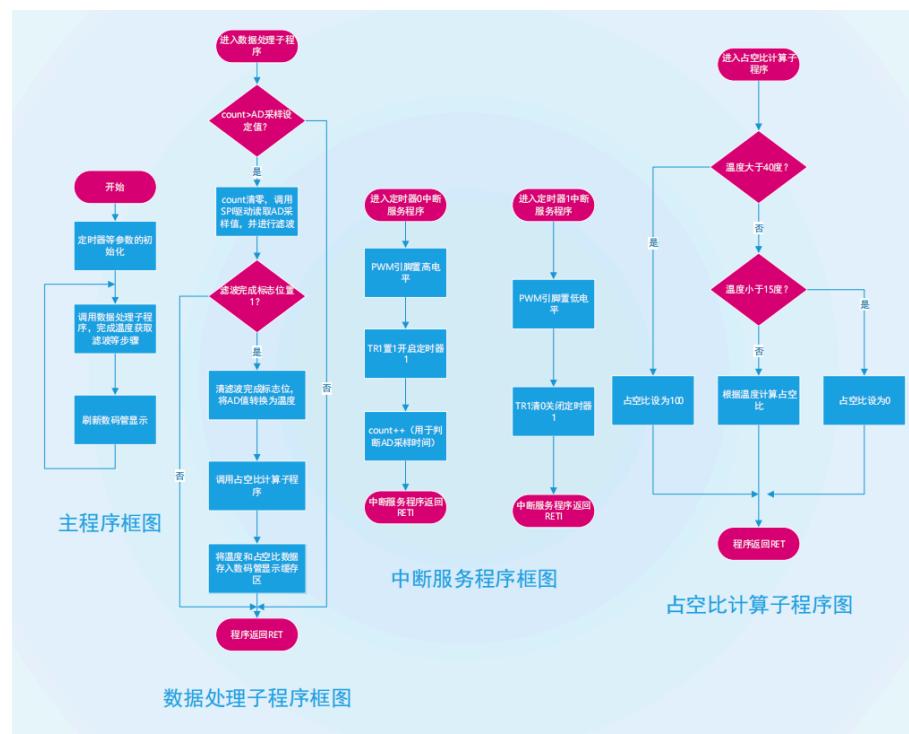
计算机，开发板：HC6800 - ES V2.0 开发板，USB 转串口线（开发板自带），直流电机

## 四、实验步骤和方法

### 1. 硬件电路设计



### 2. 软件流程图



### 3. 程序代码

main.c 文件如下 XPT2046.c,XPT2046.h 文件同实验 7

```
1 #include "reg52.h" //此文件中定义了单片机的一些特殊功能寄存器
2 #include "XPT2046.h"
3
4 typedef unsigned int u16; //对数据类型进行声明定义
5 typedef unsigned char u8;
6
7 sbit LSA=P2^2;
8 sbit LSB=P2^3;
9 sbit LSC=P2^4;
10 sbit PWM=P1^0;
11 bit DIR;
12
13 u16 count, value, timerl, pw;
14
15 u8 disp[8];
16 u8 sent[8];
17 u8 code smgduan[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
18 u8 code smgduan1[10]={0xBf,0x86,0xDB,0xCf,0xE6,0xED,0xFd,0x87,0xFF,0xEF};
19 ****
20 * 函数名      : TimerInit
21 * 函数功能    : 初始化
22 ****
23 void TimerInit(){
24     TMOD |=0X10;
25
26     TH1=0xFF;
27     TL1=0xFF;
28
29     ET1=1;
30     EA=1;
31     TR1=1;
32 }
33 ****
34 * 函数名      : delay
35 * 函数功能    : 延时函数, i=1时, 大约延时10us
36 ****
37 void delay(u16 i)
38 {
39     while(i--);
40 }
41 ****
42 * 函数名      : datapros()
43 * 函数功能    : 数据处理函数
44 * 输入        : 无
45 * 输出        : 无
46 ****
47 void datapros()
48 {
49     u16 temp;
50     u16 digit;
51     static u8 i;
52     if(i==50)
53     {
54         i=0;
55         digit = Read_AD_Data(0xD4); // AIN1 热敏电阻
56         temp = (digit - 2030)*0.86;
57         pw = (int)(temp-150)*4;
58
59         i++;
60     }
61     disp[2]=smgduan[temp%1000/100];//百位
62     disp[1]=smgduan1[temp%1000%100/10];//十位
63     disp[0]=smgduan[temp%100%10];
64
65     disp[7]=smgduan1[pw/100];//百位
66     disp[6]=smgduan[pw%100/10];//十位
67     disp[5]=smgduan[pw%100%10];
68 }
69 ****
70 ****
71 }
```

```
74 * 函数名          :DigDisplay()
75 * 函数功能        :数码管显示函数
76 * 输入           : 无
77 * 输出           : 无
78 ****
79 void DigDisplay()
80 {
81     u8 i;
82     for(i=0;i<8;i++)
83     {
84         switch(i) //位选, 选择点亮的数码管,
85         {
86             case(0):
87                 LSA=0;LSB=0;LSC=0; break;//显示第0位
88             case(1):
89                 LSA=1;LSB=0;LSC=0; break;//显示第1位
90             case(2):
91                 LSA=0;LSB=1;LSC=0; break;//显示第2位
92             case(3):
93                 LSA=1;LSB=1;LSC=0; break;//显示第3位
94             case(4):
95                 LSA=0;LSB=0;LSC=1; break;
96             case(5):
97                 LSA=1;LSB=0;LSC=1; break;
98             case(6):
99                 LSA=0;LSB=1;LSC=1; break;
100            case(7):
101                LSA=1;LSB=1;LSC=1; break;
102        }
103        P0=disp[i];//发送数据
104        delay(10); //间隔一段时间扫描
105        P0=0x00;//消隐
106    }
107 }
108 void shine()
109 {
110     if(timer1>1000)
111     {
112         timer1=0;
113     }
114     if(timer1<pw)
115     {
116         PWM=1;
117     }
118     else
119     {
120         PWM=0;
121     }
122 }
123 ****
124 * 函数名          : main
125 * 函数功能        : 主函数
126 * 输入           : 无
127 * 输出           : 无
128 ****
129 void main()
130 {
131     TimerInit();
132     while(1)
133     {
134         datapros(); //数据处理函数
135         DigDisplay();//数码管显示函数
136         shine();
137     }
138 }
139 ****
140 * 函数名          : Timer
141 * 函数功能        : 定时器中断响应函数
142 * 输入           : 无
```

```

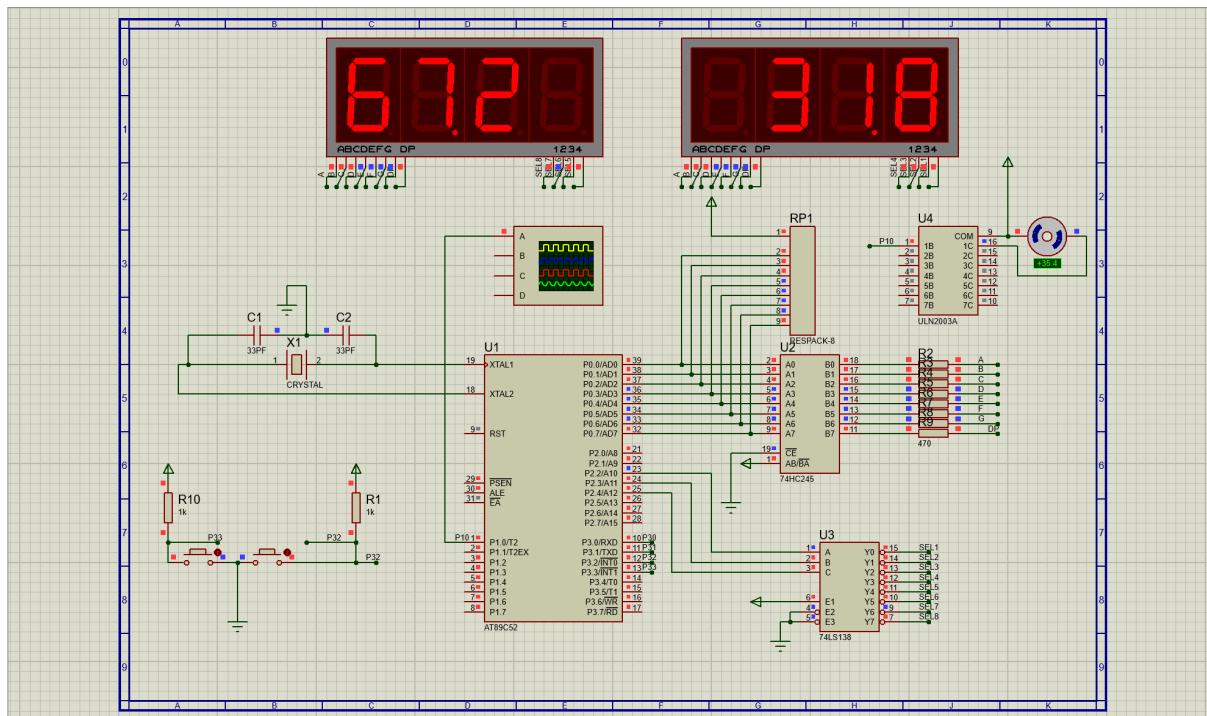
144 * 输    出      : 元
145 ****
146 void Timer(void) interrupt 3
147 {
148     TH1=0XFF;
149     TL1=0XFF;
150     timer1++;
151 }

```

## 五、实验结果与分析

### 1. 硬件电路设计

仿真后显示的温度值和占空比百分数如图



### 2. 软件仿真运行结果

显示在开发板上如图



在调试模式下，温度从 15 摄氏度变化为 40 摄氏度，占空比不断升高，风扇转速也随之变化



## 六、讨论与心得

历经一个长学期后，微机实验也迎来了尾声。从夏学期开始，我们所做的实验是以 C 语言为主，其实这更符合现实中单片机的应用场景。但是无论是理论课还是实验指导，对于 C51 的学习和理解都比较少，缺乏一个系统性的学习，基本是基于例程来补充和修改代码的，我觉得这可能是这门课程后续需要调整和改进的地方。

总体来看，这门课让我初步入门了 51 单片机开发板的设计、调试、应用方法，虽然是一些比较简单应用，但是仍然让我感受到了单片机的魅力。当然正像老师所说的，学习这门课不能为了 51 而 51，还是需要把单片机的原理和底层逻辑搞明白。后续我会尝试自购其他单片机的开发板和芯片，尝试开发其他程序，进一步巩固微机和单片机的相关知识。

实验课的总体体验还是比较好的，收获也比较大，感谢所有的老师、助教和帮助过我的同学！