

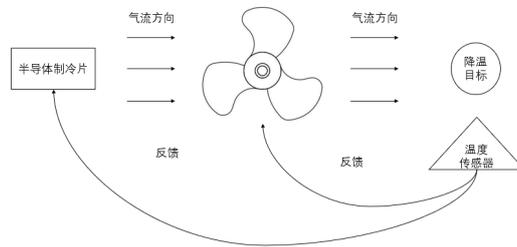
基于 Arduino 开发的自动调速风扇

陈寅聪 17307110004

一、课题设计

1.1 整体设计结构

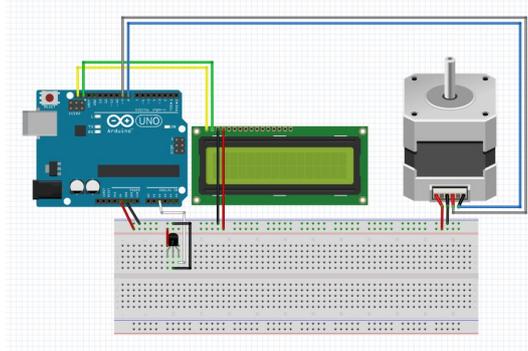
本实验中开发的自动调速风扇主要由三个部分组成：风扇部分（由电机、扇叶以及电机驱动组成）、感温部分（温度传感器）以及冷源部分（半导体制冷片）。其中 Arduino 主要控制风扇部分以及感温部分，完成自动调速的功能；感温部分可以将测温的结果反馈给风扇部分与冷源部分，调节他们的工作效率；冷源部分为风扇提供冷气源，使得风扇吹出的风的温度更低，降温效果更好。整体设计结构的示意图如图一所示。



图一 自动调速风扇的结构示意图

1.2 温度反馈机制

本实验中利用 Arduino 控制风扇部分以及感温部分，其结构图如图二所示。



图二 温度反馈机制的结构示意图

图中的元件包括 Arduino 开发板，液晶显示屏，电机（未显示扇叶），温度传感器。温度传感器将读取到的数据（电压信号）发送给 Arduino 开发板，在其中通过程序实现信号的转化，将电压信号变换为对应的温度信号，显示在液晶显示屏上；于此同时，程序根据读取到的温度信号改变与电机相连的端口的电平，以此控制电机的占空比，改变风扇的转速。

1.3 半导体制冷原理

半导体制冷器是指利用半导体的热-电效应制取冷量的器件，又称热电制冷器。用导体连接两块不同的金属，接通直流电，则一个接点处温度降低，另一个接点处温度升高。若用一个 N 型半导体和一个 P 型半导体代替金属，效应就大得多。接通电源后，上接点附近产生电子空穴对，内能减小，温度降低，向外界吸热，称为冷端。另一端因电子空穴对复合，内

能增加，温度升高，并向环境放热，称为热端。一对半导体热电元件所产生的温差和冷量都很小，实用的半导体制冷器是由很多对热电元件经并联、串联组合而成，也称热电堆。单级热电堆可得到大约 60℃的温差，即冷端温度可达-10~-20℃。半导体制冷器具有无噪声、无振动、不需制冷剂、体积小、重量轻等特点，且工作可靠，操作简便，易于进行冷量调节。但它的制冷系数较小，电耗量相对较大，故它主要用于耗冷量小和占地空间小的场合。

二、实验过程

2.1 硬件组装

本实验中选取的 Arduino 开发板型号为 Arduino UNO，温度传感器为 LM35（集成于 UNO 传感器拓展板上），液晶显示屏为 LCD1602A（含 IIC/I2C 接口 LCD1602 转接板），风扇电机为 L9110（含驱动模块 140C04），将这些元件根据图二进行组装，其中温度传感器与模拟信号接口 A2 连接，电机与数字信号接口 D8、D9 连接，液晶显示屏与 SDA 接口以及 SCL 接口连接。

冷源部分的硬件包括 S-120-12 电源（用于将220V 交流电压转换为12V 直流电压），半导体制冷片（含冷端平台以及用于半导体制冷片热端散热的风扇）。将半导体制冷片与电源用导线连接，实物图如图三所示。



图三 冷源部分硬件组装实物图

将其他部分与冷源部分进行组合，将风扇用简易支架悬挂在制冷平台的上方，实物图如图四所示。



图四 所有硬件硬件组装实物图

2.2 程序编写

在程序编写部分,主要需要实现的功能为从温度传感器中读取电压信号并转化为温度数据,并由该温度数据控制风扇的运行。具体程序如图五所示。

```
int potPin = 0;
int IA = 9;
int IB = 8;
byte speed = 128;
#include <LiquidCrystal_I2C.h> //引用I2C库
LiquidCrystal_I2C lcd(0X27,16,2);

void setup()
{
  Serial.begin(9600);
  lcd.init(); // 初始化LCD
  lcd.backlight(); //设置LCD背景等高
  pinMode(IA,OUTPUT);
  pinMode(IB,OUTPUT);
}

void loop()
{
  float val;
  float datc;
  float datf;
  val = analogRead(2);
  datc = (125 * val)/256;
  Serial.print("Temperature:");
  Serial.print(datc);
  Serial.println("C");
  lcd.setCursor(0,0); //设置显示指针
  lcd.print("Celcius:");
  lcd.print(datc); //输出字符到LCD1602上
  lcd.setCursor(0,1);
  lcd.print("Fan Speed:");
  speed = map(val,0,128,64, 255);
  lcd.print(map(speed,64,255,0,100));
  lcd.print("%");
  analogWrite(IA, speed);
  analogWrite(IB, 0);
  delay(500);
}
```

图五 Arduino 程序

蓝色方框中的程序的作用为将程序初始化,并将于电机相连的两个接口设置为输出。

红色方框中的程序从模拟信号接口 A2 中读取电压信号,并根据电压信号计算对应的温度数据,转换公式为:

$$T = \frac{500}{2^{10}} \times Signal = \frac{125}{256} Signal$$

将转换得到的温度数据储存在变量之中,并显示在液晶显示屏上。

黄色方框中的程序将温度数据与风扇转速关联起来,温度越高,风扇转速越快,并将风扇的转速以百分比的形式显示在液晶显示屏上。

程序的运行效果如图六所示。



图六 程序运行结果。液晶屏上的显示为：“Celsius: 26.86 Fan Speed: 42%”

2.3 参数调试

在参数调试部分，主要是调试风扇转速与温度的对应关系。风扇电机除了电源线以及地线接口外还有两个接口 INA 与 INB，当 INA 的电平高于 INB 时，风扇倒转；当 INB 的电平高于 INA 时，风扇正转。由于结构的限制，在本实验中需设置风扇倒转才能将冷源部分的冷空气吹出。风扇电机的占空比正比于 INA 与 INB 两接口之间的电位差，因此只需设置 INB 接口的电位始终为 0，就可以通过调节 INA 接口的电位（范围为0到255）控制风扇的转速。

经过试验，当 INA 接口的电位为 255 时，风扇的转速达到最大；当 INA 接口的电位小于 64 时，由于电机内部的阻尼影响，风扇无法转动起来，因此设置控制风扇转动速率的参数范围为 64 到 255 的整数。再考虑到风扇平时的使用范围为20°C到40°C，因此对温度传感器得到的电压信号进行进一步缩放，具体的转换函数为：

$$\frac{speed - 64}{255 - 64} = \frac{Signal}{128}$$

当气温在27°C附近时，风扇的转速为满功率运行时的转速的42%，比较符合预期。

三、实验结果和分析

利用温度计测量冷源部分的冷却效果。将半导体制冷片接通电源后，测得在30s内，冷端平台的温度降至-5.3°C，降温效果与预期相符，但平台上出现了较多的来自空气中的冷凝水，这可能对降温效果有一定的影响。

利用温度计测试风扇的降温效果。未开启风扇时，利用温度计测得的气温显示为28.4°C；调节参数使得风扇达到满功率运行，再将温度计置于风扇的风道中，测得的温度为27.6°C，降温幅度为0.8°C；再次调节参数使得风扇按照预设的自动调速模式运行，此时液晶显示屏上显示风扇的转速为满功率运行时的转速的46%，将温度计置于风扇的风道中，测得的温度为27.9°C，降温幅度为0.5°C；将冷源部分开启，并设定风扇按照预设的自动调速模式运行，此时液晶显示屏上显示风扇的转速为满功率运行时的转速的46%，将温度计置于风扇的风道中，测得的温度为24.7°C，降温幅度为3.7°C。

观察以上结论可以发现，仅仅依靠风扇达到的降温幅度并不明显，这可能是由于风扇吹出的风仍为室温的温度，因此不能带来很好的降温效果。加入冷源部分以后，降温效果相较于之前有着明显的提升，达到类似空调的局部效果，但装置总体积较小，方便移动，功耗也比空调更低，可以在天气不太炎热的情况下达到非常好的消暑作用。

实验中所使用的风扇电机模块功率较小，风扇的扇叶半径也较小，因此只能在很小的一块区域内有较好的降温效果。如果使用功率更高、面积更大的风扇模块，则可以在降温效果

以及降温范围上都有很大的提升，但也会对 Arduino 驱动造成一定的压力，不适用于本实验中。

四、实验结论

本实验利用 Arduino 模块实现了自动控温风扇的设计，可以根据环境温度自动调节风扇的转速，并在一定温度以下停止工作。实验中还加入了用半导体制冷片制成的冷源部分以提升风扇的降温效果，同时也保证了装置体积小、功耗小的优点。

五、参考文献

- [1] 基于 ArduinoUNOR3 的智能调速风扇 <https://www.cnblogs.com/dapangsen/p/7929052.html>
- [2] 半导体制冷 en.wikipedia.org/wiki/Thermoelectric_cooling