

更多原创作品尽在电路城: <http://www.cirmall.com/>

Arduino 入门教程(8)—温度报警器

在上一节中，我们认识了一个发声元件——蜂鸣器，也做了一个简单的小报警器。是不是还不过瘾呢？这次我们要做一个更实际的应用——温度报警器。当温度到达我们设定的限定值时，报警器就会响。我们可以用于厨房温度检测报警等等，各种需要检测温度的场合。这个项目中，除了要用到蜂鸣器外，还需要一个 LM35 温度传感器。

我们这里将头一回接触传感器，传感器是什么？简单的从字面上的理解就是，一种能感知周围环境，并把感知到的信号转换为电信号的感应元件。感应元件再把电信号传递给控制器。就好比人的各个感官，感知周围环境后，再信息传递给大脑是一样的道理。

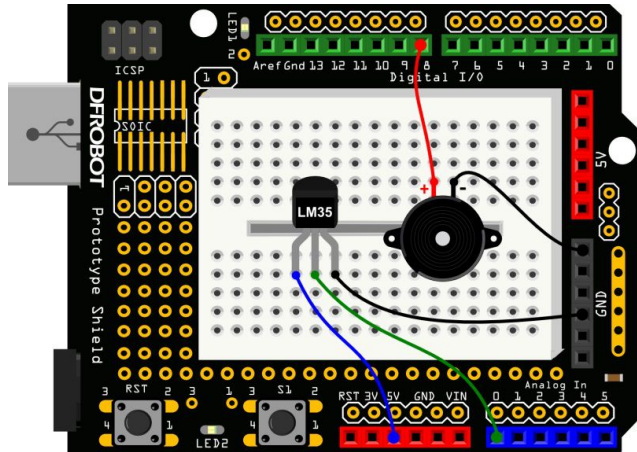
所需元件

- 1× 蜂鸣器
- 1× LM35 温度传感器

STEP 1：硬件连接

在接 LM35 温度传感器时，注意三个引脚的位置，从左至右依次接 5V、Analog 0、GND，如我们下图所示。

更多原创作品尽在电路城: <http://www.cirmall.com/>



STEP 2 : 输入代码

完成硬件连接后, 打开 Arduino IDE,输入下面这段代码。

```
1. float sinVal;
2. int toneVal;
3. unsigned long tepTimer ;
4.
5. void setup(){
6.     pinMode(8, OUTPUT);          // 蜂鸣器引脚设置
7.     Serial.begin(9600);          //设置波特率为 9600 bps
8. }
9.
10. void loop(){
11.     int val;                      //用于存储 LM35 读到的值
12.     double data;                  //用于存储已转换的温度值
13.     val=analogRead(0);            //LM35 连到模拟口, 并从模拟口读值
14.     data = (double) val * (5/10.24); // 得到电压值, 通过公式换成温度
15.
16.     if(data>27){                  // 如果温度大于 27, 蜂鸣器响
17.         for(int x=0; x<180; x++){
18.             //将 sin 函数角度转化为弧度
19.             sinVal = (sin(x*(3.1412/180)));
20.             //用 sin 函数值产生声音的频率
21.             toneVal = 2000+(int(sinVal*1000));
22.             //给引脚 8 一个
```

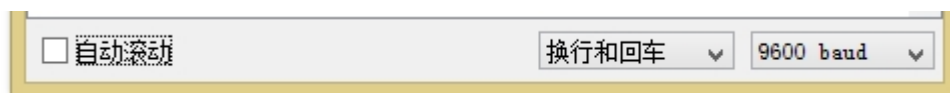
更多原创作品尽在电路城: <http://www.cirmall.com/>

```
23.         tone(8, toneVal);
24.         delay(2);
25.     }
26. } else { // 如果温度小于 27, 关闭蜂鸣器
27.     noTone(8); //关闭蜂鸣器
28. }
29.
30. if(millis() - tepTimer > 500){ // 每 500ms, 串口输出一温度值
31.     tepTimer = millis();
32.     Serial.print("temperature: "); // 串口输出"温度"
33.     Serial.print(data); // 串口输出温度值
34.     Serial.println("C"); // 串口输出温度单位
35. }
36. }
```

成功下载完程序后, 打开 Arduino IDE 的串口监视器。

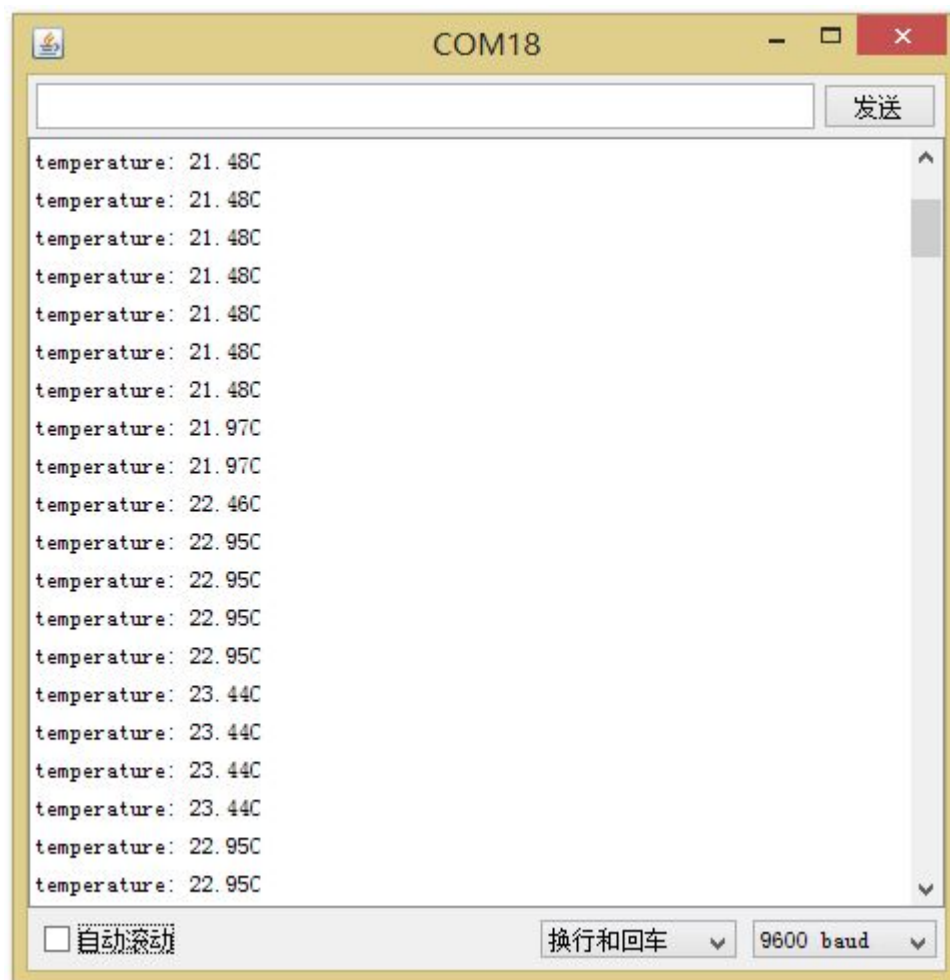


设置串口监视器的波特率为 9600。



就可以直接从串口中读取温度值, 并尝试升高周围环境温度, 或者用手直接接触 LM35 使其升温, 串口可以很直观的看到温度有明显的变化。

更多原创作品尽在电路城: <http://www.cirmall.com/>



蜂鸣器工作的条件是，一旦检测到环境温度大于 27 度，蜂鸣器鸣响，环境温度小于 27 度，则关闭蜂鸣器。

STEP 3: 代码回顾

这段代码与报警器一节的大部分内容是相同的，就直接讲讲不同的吧！

setup()函数的第一句，我们想必已经很熟了，设置蜂鸣器为输出模式，有人可能会问为什么 LM35 不用设置呢？LM35 是个模拟量，模拟量不需要设置引脚模式。pinMode 只用于数字引脚。

Arduino 的通信伙伴——串口

更多原创作品尽在电路城: <http://www.cirmall.com/>

串口是 Arduino 和外界进行通信的一个简单的方法。每个 Arduino 都至少有一个串口, UNO 分别与数字引脚 0(RX)和数字引脚 1(TX)相连。所以如果要用到串口通信的, 数字 0 和 1 不能用于输入输出功能。

串口可用的函数也有好多, 可用查看语法手册。我们这里就先介绍几个常用的:

```
1. Serial.begin(9600);
```

这个函数用于初始化串口波特率, 也就是数据传输的速率, 是使用串口必不可少的函数。直接输入相应设定的数值就可以了, 如果不是一些特定的无线模块对波特率有特殊要求的话, 波特率设置只需和串口监视器保持一致即可。我们这里就只是用于串口监视器。

```
1. val=analogRead(0);
```

这里用到了一个新函数——`analogRead(pin)`。

这个函数用于从模拟引脚读值, `pin` 是指连接的模拟引脚。Arduino 的模拟引脚连接到一个了 10 位 A/D 转换, 输入 0~5V 的电压对应读到 0~1023 的数值, 每个读到的数值对应的都是一个电压值。

我们这里读到的是温度的电压值, 是以 0~1023 的方式输出。而我们 LM35 温度传感器每 10mV 对应 1 摄氏度。

```
1. data = (double) val * (5/10.24);
```

从传感器中读到的电压值, 它的范围在 0~1023, 将该值分成 1024 份, 再把结果乘以 5, 映射到 0~5V, 因为每度 10mV, 需要再乘以 100 得到一个 double 型温度值, 最后赋给 `data` 变量。

后面进入一个 if 语句, 对温度值进行判断。这里的 if 语句与之前讲的有所不同。if...else 用于对两种情况进行判断的时候。

if...else 语句格式:

更多原创作品尽在电路城: <http://www.cirmall.com/>

if(表达式){

语句 1 ;

} else{

语句 2 ;

}

表达式结果为真时, 执行语句 1, 放弃语句 2 的执行, 接着跳过 if 语句, 执行 if 语句的下一条语句; 如果表达式结果为假时, 执行语句 2, 放弃语句 1 的执行, 接着跳过 if 语句, 执行 if 语句的下一条语句。无论如何, 对于一次条件的判断, 语句 1 和语句 2 只能有一个被执行, 不能同时被执行。

回到我们的代码, if 中的语句就省略不说了, 不明白的可以回看前一节:

```
1.   if(data>27){
2.       for(int x=0; x<180; x++){
3.           .....
4.       }
5.   } else {
6.       .....
7.   }
```

进入 if 判断, 对 data 也就是温度值进行判断, 如果大于 27, 进入 if 前半段, 蜂鸣器鸣响。否则, 进入 else 后的语句, 关闭蜂鸣器。

除了不断检测温度进行报警, 我们还需要代码在串口实时显示温度。这里又用到 millis() 函数 (项目三中有说明), 利用固定的机器时间, 每隔 500ms 定时向串口发出数据。

那串口收到数据后, 如何在串口监视器上显示呢? 就要用到下面的两句语句:

```
1.   Serial.print(val);
2.   Serial.println(val);
3.
```

更多原创作品尽在电路城：<http://www.cirmall.com/>

`print()`的解释是，以我们可读的 ASCII 形式从串口输出。

这条命令有多种形式：

- (1) 数字则是以位形式输出 (例 1)
- (2) 浮点型数据输出时只保留小数点后两位 (例 2)
- (3) 字符和字符串则原样输出，字符需要加单引号 (例 3)，字符串需要加双引号 (例 4)。

例如：

- (1) `Serial.print(78);` 输出 “78”
- (2) `Serial.print(1.23456);` 输出 “1.23”
- (3) `Serial.print('N');` 输出 “N”
- (4) `Serial.print("Hello world.");` 输出 “Hello world.”

不仅有我们上面这种形式输出，还可以以进制形式输出，可以参看语法手册。

`println()`与 `print()`区别就是，`println()`比 `print()`多了回车换行，其他完全相同。

串口监视器输出还有一条语句比较常见的是 `Serial.write()`，它不是以 ASCII 形式输出，而是以字节形式输出，感兴趣的可以查看语法手册。

代码中，可能有一处会不太明白：

```
1. Serial.print(data);
```

有人会问，`data` 不是字符串吗？怎么输出是数字呢？不要忘了，这是我们前面定义的变量，它其实就是代表数字，输出当然就是数字啦！

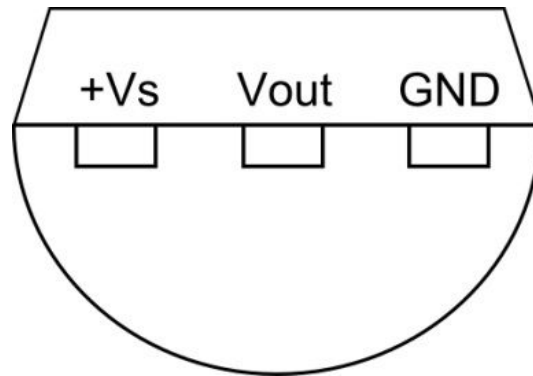
STEP 4: 硬件回顾

LM35

更多原创作品尽在电路城：<http://www.cirmall.com/>

LM35 是一种常见的温度传感器，使用简便，不需要额外的校准处理就可以达到 $\pm 1/4^{\circ}\text{C}$ 的准确率。

我们看一下 LM35 引脚示意图，Vs 接入电源，Vout 是电压输出，GND 接地。



计算公式： $V_{out} = 10\text{mV}/^{\circ}\text{C} * T^{\circ}\text{C}$ （温度范围在 $+2^{\circ}\text{C} \sim 40^{\circ}\text{C}$ ）

可以做些其他的事

将我们上面的温度报警器再结合 LED 灯。在不同的温度范围设置不同颜色灯，并伴随不同频率的声音。

比如：温度小于 10 或者大于 35，亮红灯，蜂鸣器发出比较急促的声音。

温度在 25~35 之间，亮黄灯，蜂鸣器伴随相对缓和的声音。

温度在 10~25 之间，亮绿灯，关闭蜂鸣器。