

5.5 项目 12——脉搏监测仪

这个项目采用超亮红外(IR)LED 和光敏晶体管来探测手指的脉搏,一个红色 LED 会随着脉搏闪动。

本项目所使用的元器件及设备见表 5.3。

表 5.3 元器件及设备

元器件及设备	说 明	附 录
	Arduino Diecimila 或 Duemilanove 板或兼容板	1
D ₁	5mm 红色 LED	23
D ₂	5mm 红外 LED 发射器,波长 940nm	26
R ₁	56k Ω ,0.5W 金属膜电阻	12
R ₂	270 Ω ,0.5W 金属膜电阻	6
R ₃	390 Ω ,0.5W 金属膜电阻	4
T ₁	红外光敏三极管(波长与 D ₂ 相同)	36

5.5.1 硬 件

脉冲监视器是这样工作的:由于发光 LED 是在手指的一面,而光敏三极管在手指的另一面,光敏三极管用来获取发射的光通量,当血压脉动通过手指时,光敏三极管的电阻会有微小的变化。

本项目的原理电路图如图 5.13 所示,面包板布局图如图 5.15 所示。我们选择了一个阻值非常高的电阻 R₁,因为通过手指的大部分光被吸收,因此希望光敏三极管足够灵敏。可以通过实验来选择电阻以得到最佳的结果。

最重要的是尽可能屏蔽进入光敏三极管的杂散光。这一点对于家里的灯光来说特别重要,因为家里的灯光大多是以 50Hz 或 60Hz 波动的,因此会给微

弱的心跳信号增加相当大的噪声。

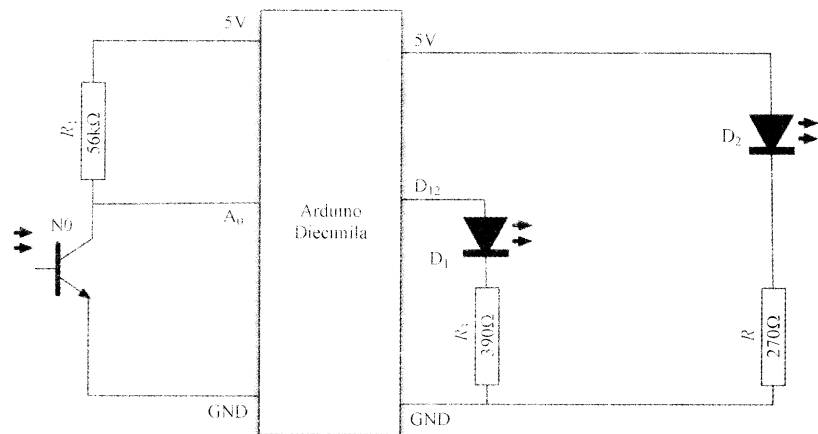


图 5.13 项目 12 的原理电路图

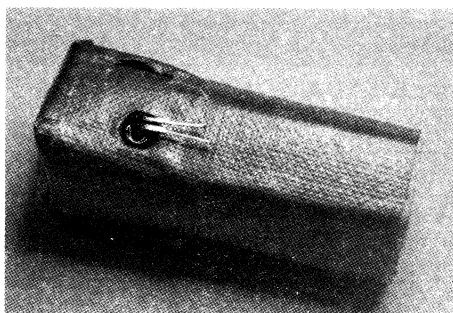


图 5.14 心跳监测仪的传感器管套

由于这个原因，光敏三极管和 LED 是放置在套管中或用套管固定的瓦楞纸中，这种结构如图 5.14 所用。

在管子相对的两面钻两个 5mm 的孔，将 LED 插入其中一边的孔中，光敏三极管插入另一边的孔中，将短引线焊接到 LED 和光敏三极管上，然后再用一层套管将管子、引线等所有东西固定。在用胶带粘贴前一定要仔细检查带有颜色的导线与 LED 和光敏三极管引线是否正确连接。

这个项目(图 5.15)的面包板原型非常简单，最后的“指套”如图 5.16 所示。

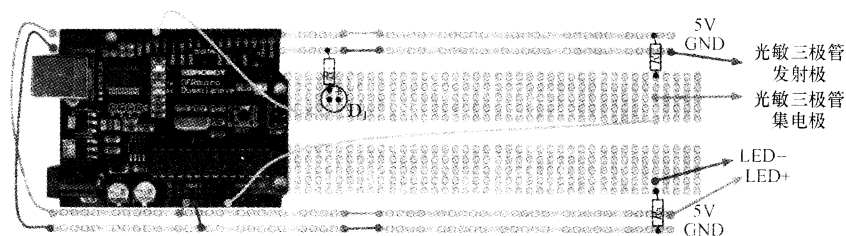


图 5.15 项目 12 的面包板布局图

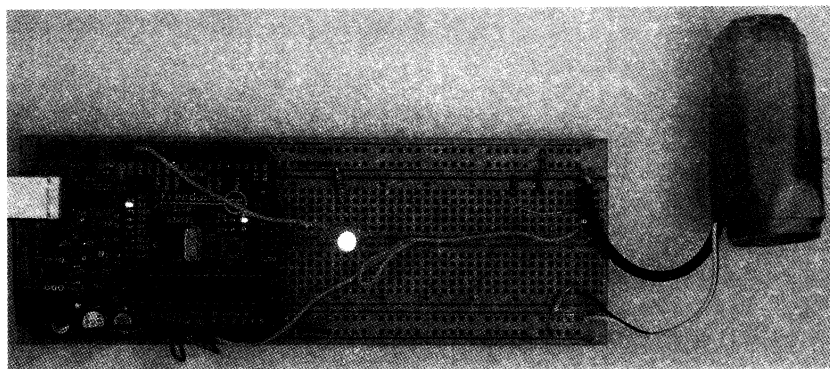


图 5.16 项目 12:脉搏监测仪

5.5.2 软件

要调试这个项目的软件是相当复杂的。第一步甚至不是运行整个最终程序,而是运行测试程序。这个测试程序将采集脉搏数据,然后将数据粘贴到电子表格和图表中,以测试数据平滑算法(更多的说明见后)。

在项目 12 的程序清单中提供了测试程序。

项目 12 程序清单——测试程序:

```
int ledPin = 13;
int sensorPin = 0;

double alpha = 0.75;
int period = 20;
double change = 0.0;

void setup()
{
    pinMode(ledPin, OUTPUT);
    Serial.begin(115200);
}

void loop()
{
    static double oldValue = 0;
    static double oldChange = 0;
    int rawValue = analogRead(sensorPin);
    double value = alpha * oldValue
```

```

    + (1 - alpha) * rawValue;

    Serial.print(rawValue);
    Serial.print(",");
    Serial.println(value);

    oldValue = value;
    delay(period);
  }

```

这个程序从模拟输入端读取原始数据并且进行平滑处理,然后将这些原始数据和平滑后的数据写至串口监视软件(Serial Monitor),通过串口监视软件可以获取这两组数据,并且将它们粘贴到电子表格用以进一步分析。注意,这个串口监视软件的通信速率设置为最高,以减小传输数据带来的延迟影响。当打开串口监视软件时,需要将串行通信速度调到115 200波特。

平滑功能采用了一种称为“leaky integration”的方法,可以通过代码了解这种方法,我们用如下代码完成这种平滑:

```
double value = alpha * oldValue + (1 - alpha) * rawValue;
```

变量 alpha 是大于 0 小于 1 的数,其大小决定了数据平滑的程度。

把手指放入传感器管套,打开串口监视软件,让程序运行 3~4s,这样可以采集到几次脉搏跳动。

然后,将采集数据的文本粘贴到电子表格中。此时可能需要采用列分隔符,即逗号。平滑后的最终数据和原始数据显示在图 5.17 所示的电子表格的两列。

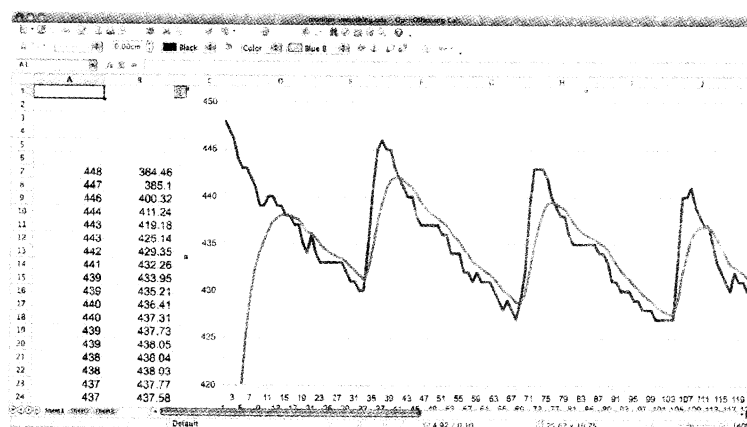


图 5.17 粘贴到电子表格中的心跳监测仪的测试数据

从模拟端口获取的原始数据表现为更加波动的曲线,而平滑后的曲线明显地已经去除了大部分的噪声。如果平滑后的曲线显示出明显的噪声,特别是虚假峰值会干扰监测,这时可以通过减小 α 值来增加平滑程度。

一旦为传感器装置找到了合适的 α 值,就可以把这个值传输到实际程序中,并且切换到实际程序而不是测试程序。实际程序清单如下所示。

项目 12 程序清单:

```
int ledPin = 13;
int sensorPin = 0;

double alpha = 0.75;
int period = 20;
double change = 0.0;

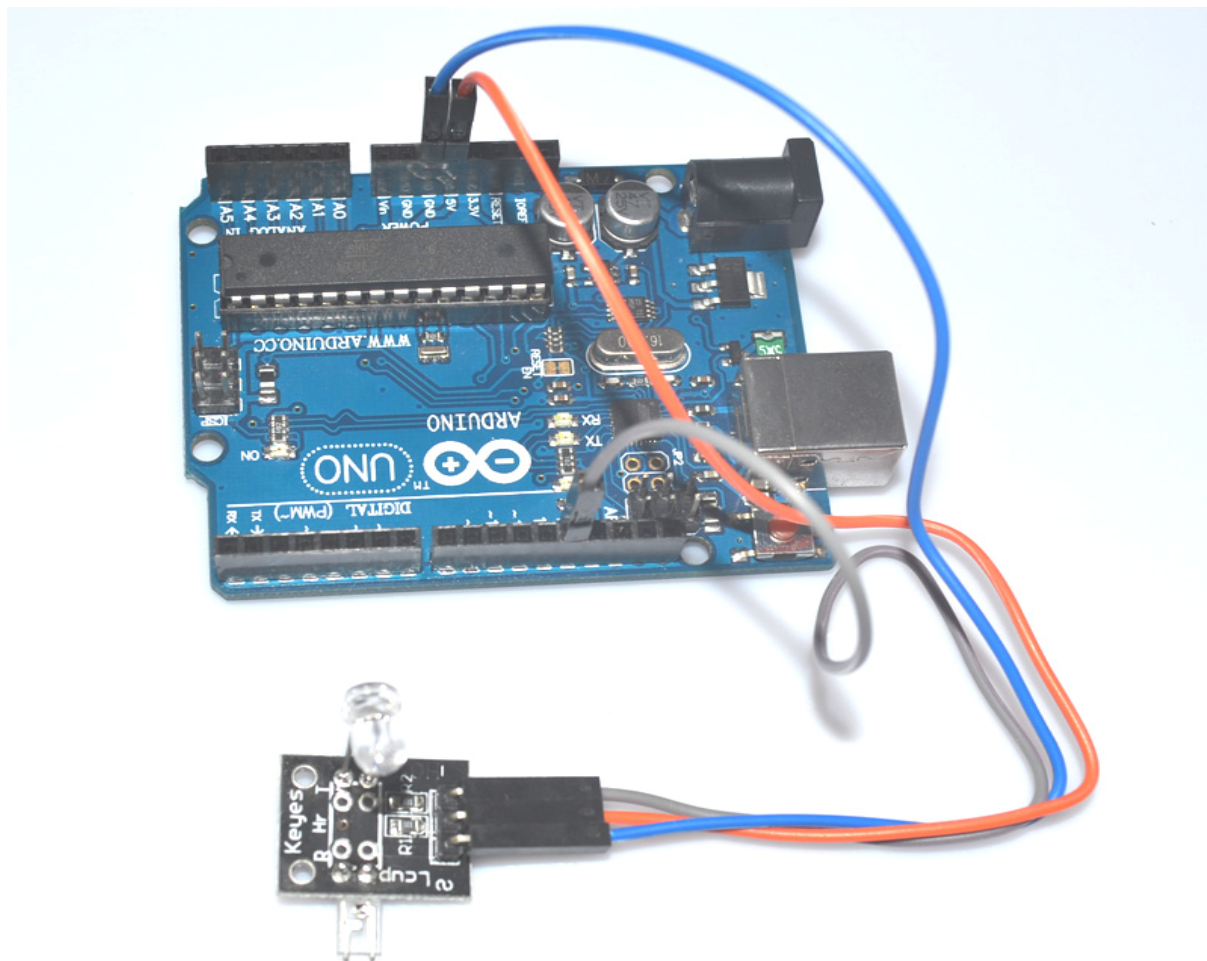
void setup()
{
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    static double oldValue = 0;
    static double oldChange = 0;
    int rawValue = analogRead(sensorPin);
    double value = alpha * oldValue
        + (1 - alpha) * rawValue;
    change = value - oldValue;

    digitalWrite(ledPin, (change < 0.0 && oldChange > 0.0));

    oldValue = value;
    oldChange = change;
    delay(period);
}
```

现在唯一的问题就是检测峰值。从图 5.17 我们可以看出,如果持续跟踪前面读取的数据,会发现数据是逐渐增加直到出现反向变化,即向着逐渐减小的方向变化。因此,如果每当前面的数据向正的方向变化而新的数据向负的方向变化时点亮 LED,就会从 LED 在每一个脉冲峰值处得到一个短脉冲。



基于 Arduino 的趣味电子制作

5.5.3 项目集成

项目 12 的测试程序和实际程序都在 Arduino Sketchbook 中,按照安装指南将其下载到板子上(见第 1 章)。

正如前面提到的,要使这个项目工作起来有一定难度。你可能会发现必须把手指放在正好合适的位置才会开始产生脉冲。如果有任何问题,按照前面的描述运行测试程序,核实探测器是否有脉冲,并且平滑因子 α 足够是否低。

作者指出,这个装置不适合用于任何实际的医学应用。