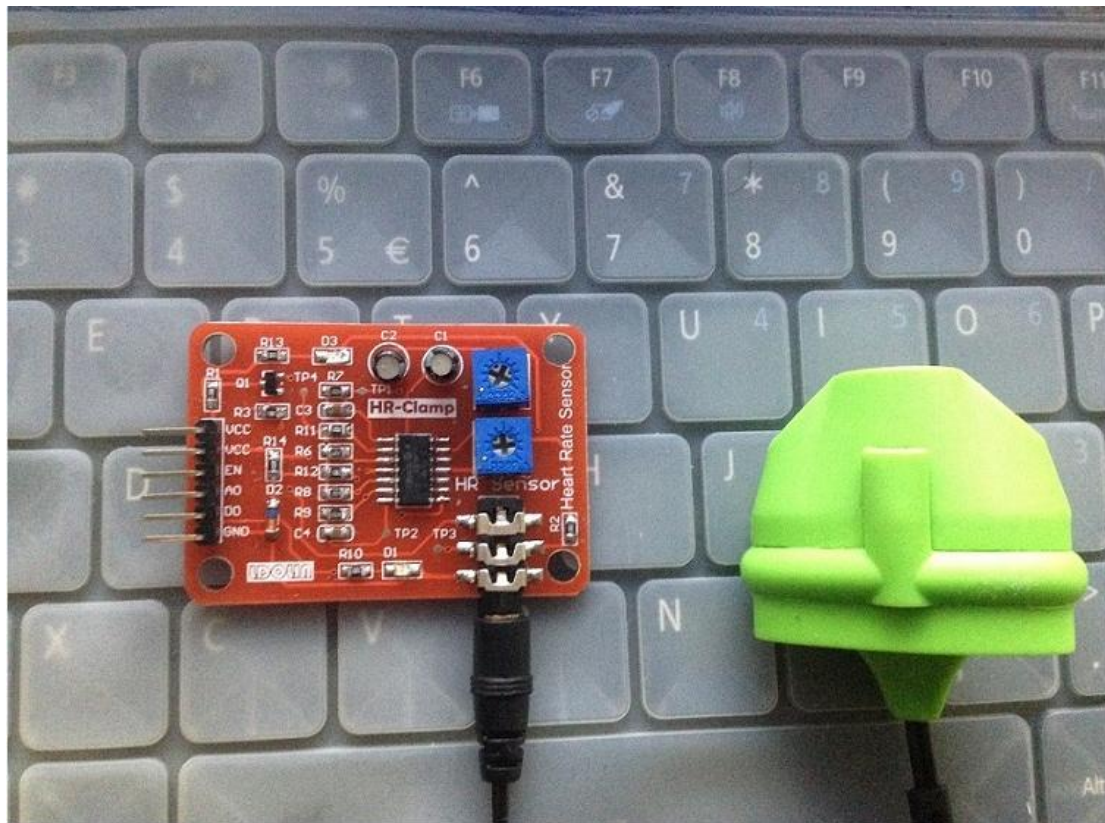
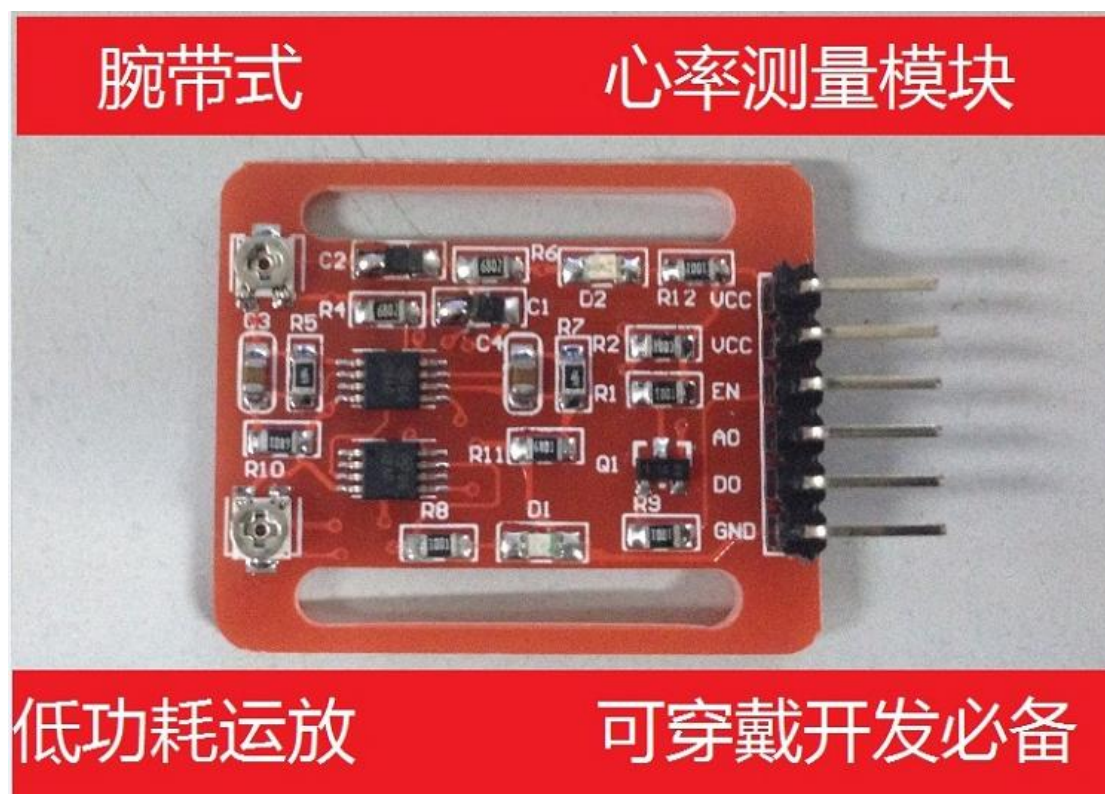


指夹式 心率测量模块



腕带式 心率测量模块



淘宝店铺地址 <http://ldolin.taobao.com/>

模块介绍

HR-Clamp 是 Heat Rate Clamp 的缩写，此版本的心率传感器模块采用的是指夹式获取指尖心率数据，传感器采样原理是通过高穿透率的 IR 红外发射管发射不可见光，照射指尖，红外发射管对面是对应的红外接收管。在心跳不同时间，指尖的毛细血管的血量是不一样的，发射管的红外光透过手指，红外接收管接收到经过信号放大从而根据透过手指的红外光判断心率。

淘宝地址 Idolin.taobao.com

HR-Wristband 是 腕带式版本的心率测量模块,心率指夹式传感器是通过 穿透式测量方法测量心率,而腕带式传感器是通过皮肤反射式测量心率,无疑此种方法传感器输出的信号更小,测量难度稍大

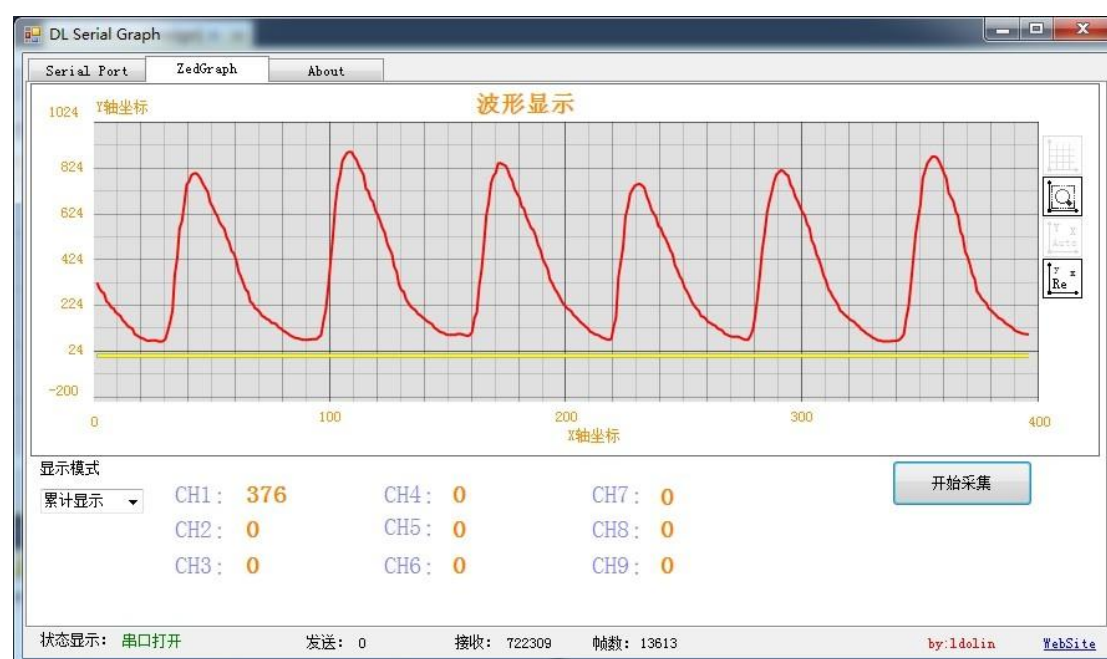
这两种方案基本都需要 **人体静止测量**,比如 指夹式 腕带式 测量过程中弯曲一个手指都会受到影响

两种模块对比

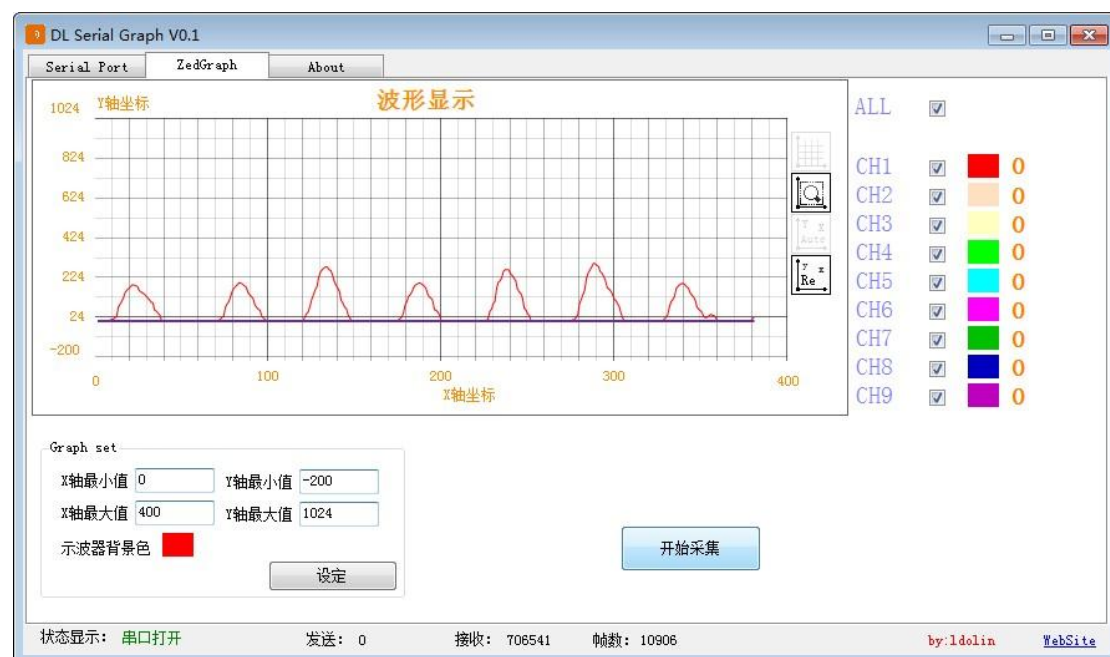
指夹式:测试最为灵敏 而且抗干扰能力比较强,测量到的心率波形比较稳定 波形也很好

腕带式:测量也比较灵敏,运放放大倍数比较大 抗干扰能力不是很强,

指夹式 心率测量模块 AO 端输出波形



腕带式 心率测量模块 AO 端输出波形



模块电器参数

输入 VCC 3V-5V

输出 AO 信号可调

输出 DO 信号为高低电平

引脚功能介绍

VCC, GND 电源接口

EN 选通引脚: 连接 VCC 选通, 悬空或者连接 GND 不选通,

控制 IR 发射

AO 模拟信号输出，输出为心率波形，可通过示波器，单片机 AD 采集信号。

DO 数字信号输出，通过模块上运放的比较电路输出高低电平的心率信号，方便心率计数。

使用方法

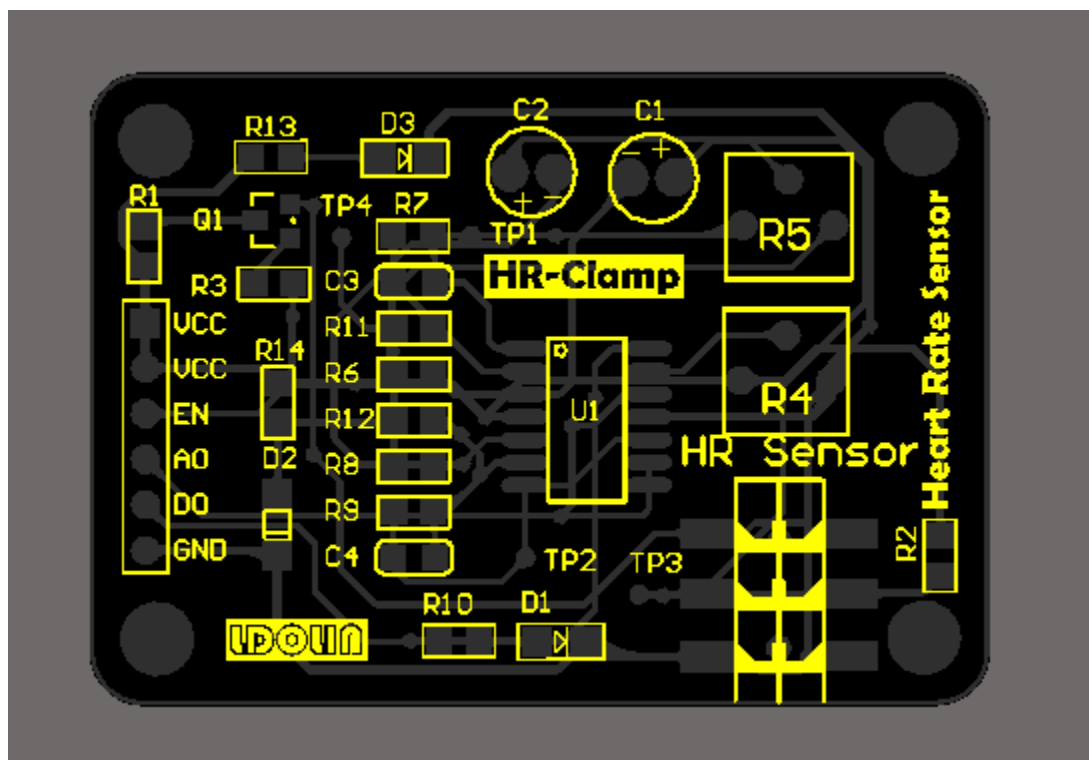
心率计与 Arduino 连接方式如下，

Arduino 上发送代码（见附件）

串口示波器使用简单，

1. 连接好模块，并将 Arduino 插入 PC 后打开串口示波器，
2. 选择串口号，打开串口，
3. ZedGraph 界面，点击开始采集即可采集到心率波形

调整心率波形

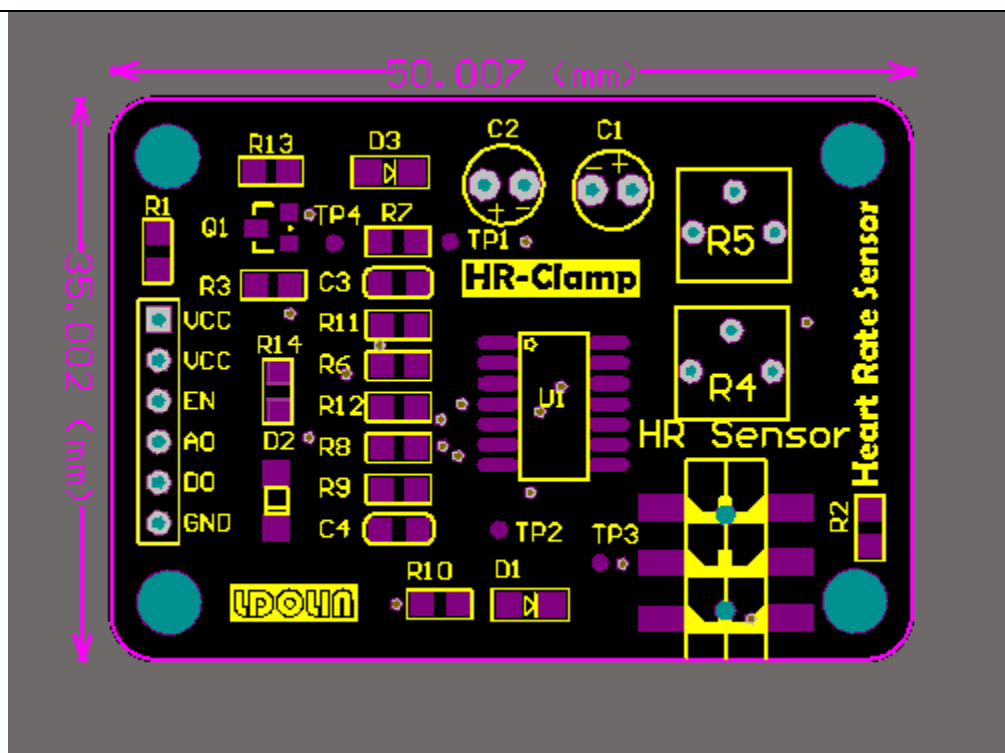


通过模块上的 变阻器 R5 调整运放的放大倍数，从而达到调整心率波形的目的

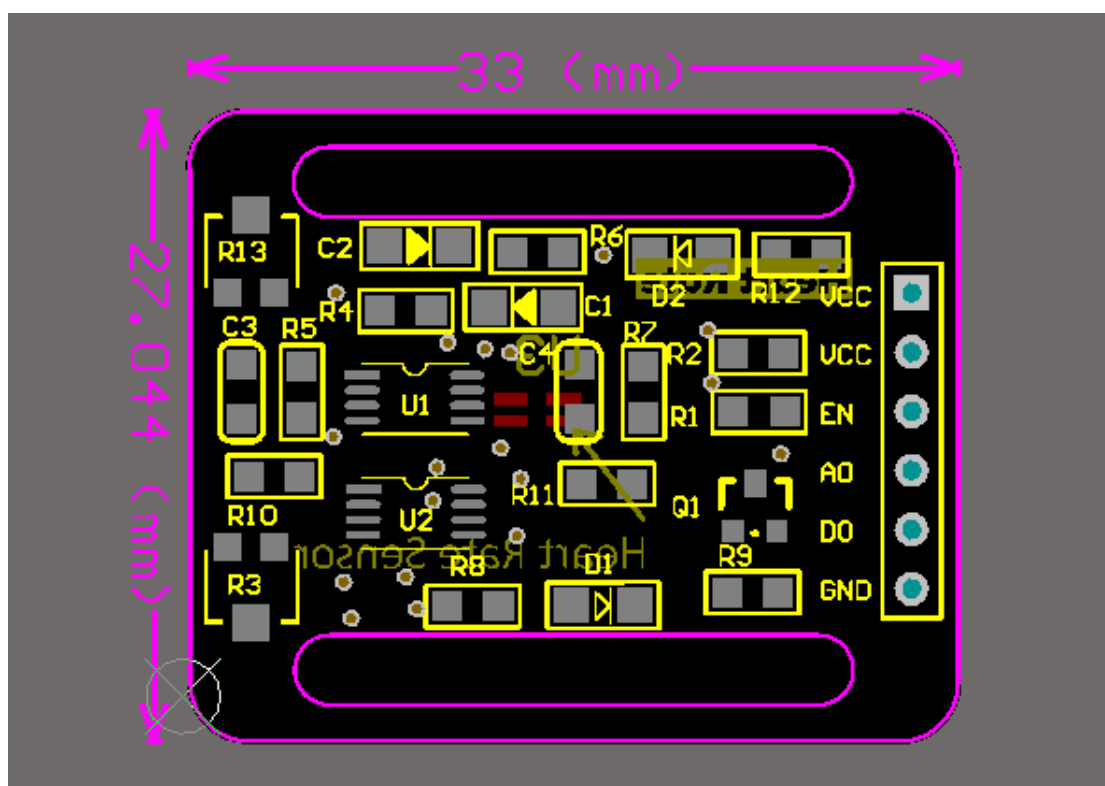
调整心率计模块的变阻器 R4 可以调节比较放大器的比较电压，从而调节 DO 输出（当放大倍数较小时，模拟输出 AO 端电压较低，比较电压比较大，只有模拟输出端 AO 大于比较电压时 DO 才会输出高电平）

模块尺寸

指夹式 心率测量模块 尺寸如下



腕带式 心率测量模块 尺寸如下



附件

Arduino 应用程序采集心率模块心率信号程序（配套 DL Graph 串口示波器使用） 串口示波器图片如下

```
/* AnalogReadSerial Reads an analog input on pin 0, prints the result to the serial monitor    This
example code is in the public domain. */
void setup()
{
    Serial.begin(115200);
}
void loop()
{
    int sensorValue = analogRead(A0);
    Serial.write(0x57);
    Serial.write(0x43);
    Serial.write(18);
    Serial.write(0xa1);

    Serial.write(sensorValue>>8);
    Serial.write(sensorValue);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);

    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);

    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);
    Serial.write(0);

    /*Serial.println(sensorValue,DEC);
    Serial.println(",");
```



```
Serial.println(1000,DEC);
Serial.println(",");
/*Serial.println(22);
Serial.println(",");
Serial.println(0); */

delay_x(5);
}
void delay_x(uint32_t millis_delay)
{
uint16_t micros_now = (uint16_t)micros();
while (millis_delay > 0)
{
if (((uint16_t)micros() - micros_now) >= 1000)
{ millis_delay--; micros_now += 1000; }
}
}
```