程序调试开发过程总结

该项目程序编写采用增量模型进行开发，即优先确保基础功能的实现，在有余力的情况下对项目功能进行进一步的开发完善。

项目开发版本说明

1.0.0 实现MLX90614测温模块温度数据的串口打印

1.1.0 添加蓝牙HC05模块，实现MLX90614测温模块温度数据的远程显示

1.2.0 实现MAX30102心率模块的脉搏波测定和心率测定数据的远程显示

1.2.1 添加温度校准曲线修正数据，优化平均心率计算方法

1.3.0 添加APP远程按键控制功能，实现远程功能切换

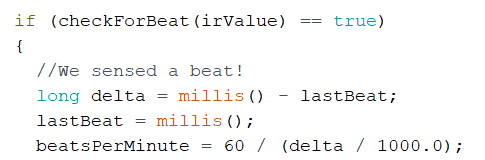
1.4.0 添加LED/蜂鸣器模块，实现心率有光/有声显示

2.0.0 优化硬件连线，增加用户友好性操作提示

2.1.0 增加代码注释，优化变量名，程序规范化处理，上传github

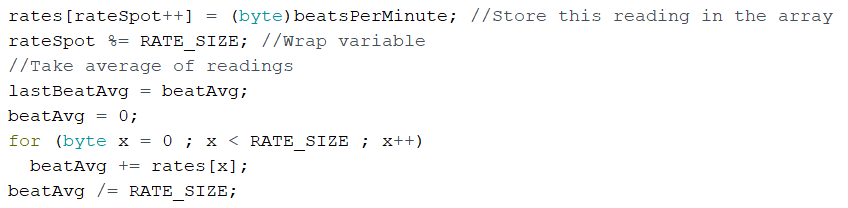
下面就关键人体参数的计算方法作进一步说明。

1. 瞬时心率检测方法



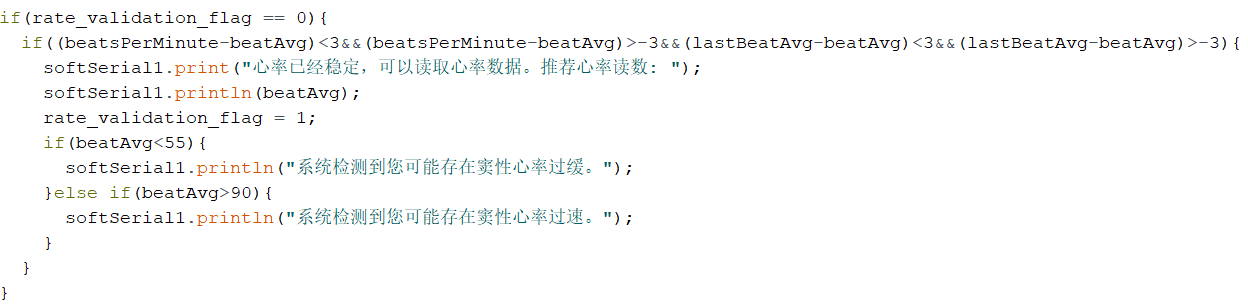
程序调用模块自带库函数checkForBeat(irValue)，传入检测得到的IR值，用作差法判断IR值变化是否显著，若显著则视为检测到一次心跳，通过计算当次心跳和上一次心跳的时间差以计算瞬时心率。

1. 平均心率计算方法



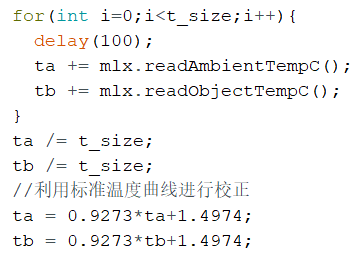
平均心率的计算建立在瞬时心率的基础上，方法是将多次有效瞬时心率（大于20小于255）取平均，一般来说4次以上有效瞬时心率的平均值结果已经比较优异，在实验中为了结果的稳定性我们选定该次数为10。

1. 心率稳定判断方法



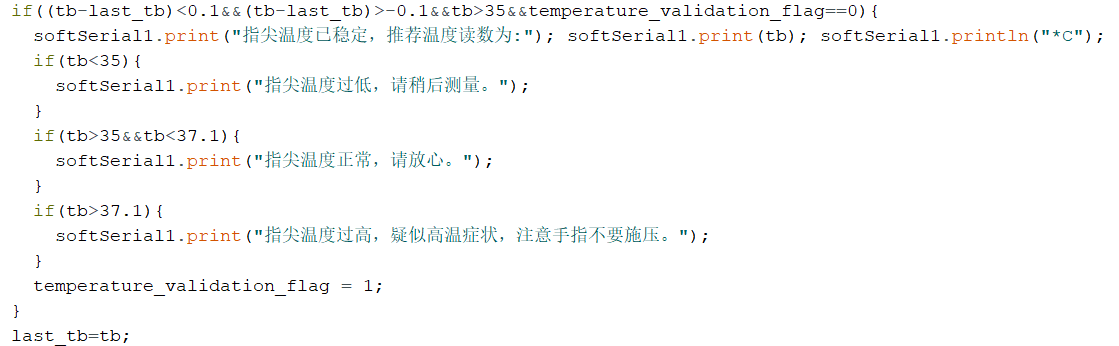
我们认为，当平均心率与有效瞬时心率相差的绝对值小于3，且上一次平均心率与当前平均心率小于3时，用户心率数据已稳定，可以读数。同时，我们将根据读数给出针对性提示。

1. 指端温度校正方法



单次红外测温数据使用MLX90614提供的库函数，我们认为，单次红外测温数据受基线漂移以及指端微颤影响较大。为此我们每三秒测定30次温度数据并求平均，以期获得有统计学意义的指端温度数据，同时，利用先前温度矫正实验得到的线性温度映射，对原始温度进行校正，得到实际指端温度。

1. 指端温度稳定判断方法



我们，当前指端温度与三秒前指端温度相差不超过0.1时，指端温度数据趋于稳定，用户可以进行读数。同时，我们将根据稳定的指端温度读数给用户针对性的提示。

遇到的软硬件问题和解决办法

1. IIC总线通信问题
   1. 我们首先遇到的问题就是IIC总线的通信问题，在多个设备并联IIC总线的情况下，出现了严重的IIC通信问题，各传感器元件之间存在通信的相互干扰，导致程序宕机，停止于Wire.h库中的endTransmission()函数。
   2. 由于开发时间受限的原因，最终我们也没有彻底解决这个问题，而是采用了转换硬件连线而不让各SCL,SDA线并联的方法来规避通信干扰。
2. 程序短路问题
   1. 由于心率传感器元件需要用手指直接触碰光电元件，加上元件十分集成，导致手指表面有电流流经，引发各引脚短接导致短路。
   2. 我们采用保鲜膜覆盖的方法，防止短路的同时，尽可能不影响远红外线的透射率。
3. 指端压力不均问题
   1. 由于指端对传感器的压力直接影响远红外光的透过率，而由于肌紧张的作用，人的主动施压不可能保持长时间的压力均一。
   2. 为此为了使一段时间内的指端压力保持一致，我们采用松紧带包裹手指的方法，对手指主动施压，得到了令人满意的结果。
4. 传感器元件不稳固问题
   1. 在最初的实验中为了调试于校准方便，我们均采用杜邦线和传感器直接相连，但是这样易导致设备连线混乱等问题。
   2. 我们重新设计了各元件的排布，并全部采用面包板插接的方式提高传感元件的牢固性，使检测结果的准确性得以进一步提升。