

项目 B 结题报告

2017011048 电 75 牛腾腾

一、项目设计

1. 项目立项

1.1. 项目名称

基于 Pulse Sensor 的心率测量系统。

1.2. 项目应用背景

心率以及心跳波形对健康状况的判断有一定的指导意义，如今市场上也有很多提供心率测量功能的智能穿戴设备，本项目旨在设计一种实用简单、成本低廉的心率测量系统，能够得到心率值以及心跳波形。

1.3. 需求分析

1.3.1. 硬件需求

Pulse Sensor 心率脉搏传感器、蓝牙模块、LCD 显示屏、MSP430G2553 单片机及扩展板

1.3.2. 软件需求

蓝牙上位机，用于绘制心电图的软件，蓝牙、LCD 模块等的驱动程序、信号处理程序

2. 功能描述

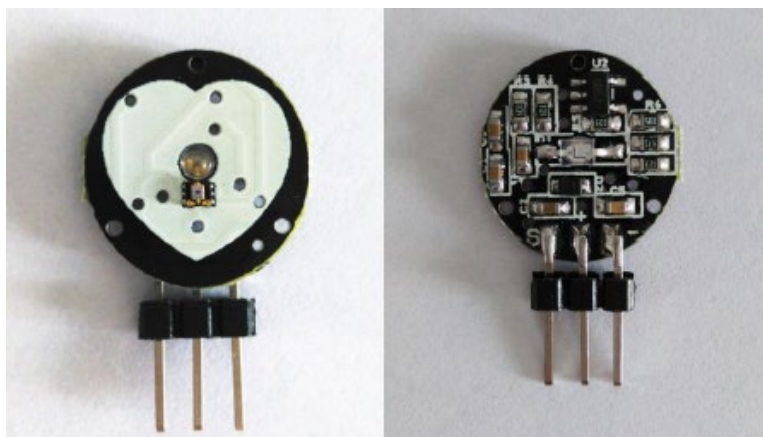
2.1. 本项目所设计的心率测量系统有三个预期功能：

1. 监测、计算得到心率的值，并显示在 LCD 液晶显示屏上
2. LED 灯随着心跳的亮灭
3. 通过蓝牙通信模块将心跳波形传至上位机，绘制出心电图。

2.2. 关键模块说明

1 . Pulse Sensor 心率脉搏传感器

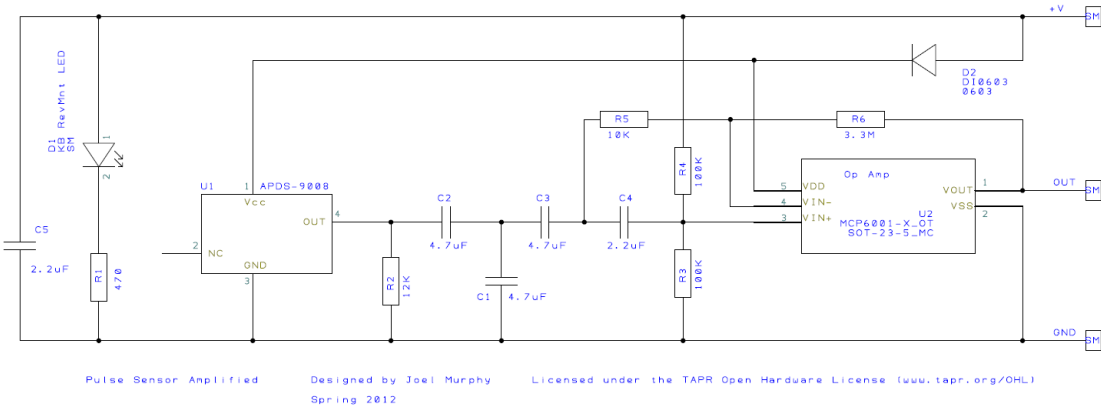
Pulse Sensor 是一款用于心率测量、脉搏波形测量和 HRV 分析的光电反射式模拟传感器。将其佩戴于手指、耳垂等处，即可采集处脉搏的模拟信号。通过 MSP430G2553 将该模拟信号转换为数字信号，通过简单计算即可得到心率数值，此外还可以将脉搏波形和心率数值上传至电脑进行显示。



Pulse Sensor 实物图

传感器参数：

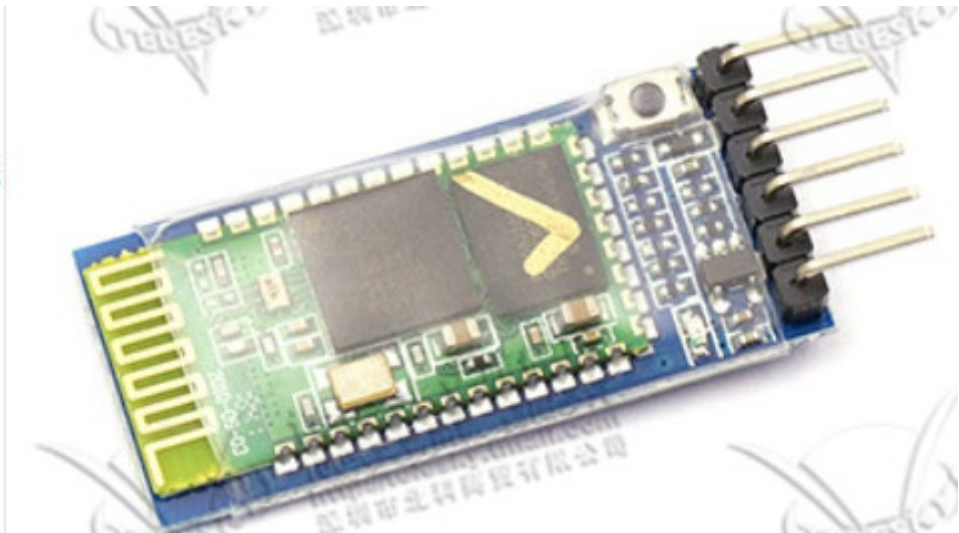
电路板直径	16mm	信号放大倍数	330 倍
电路板厚度	1.2mm	输出信号范围	0~Vcc
LED 峰值波长	515nm(绿光)	电流大小	~4mA(5V 下)
供电电压	3.3-5V	分辨率	1bpm
检测信号类型	光反射信号(PPG)	采样率	500Hz(由程序设定)
输出信号类型	模拟电压信号		



Pulse Sensor 硬件原理图

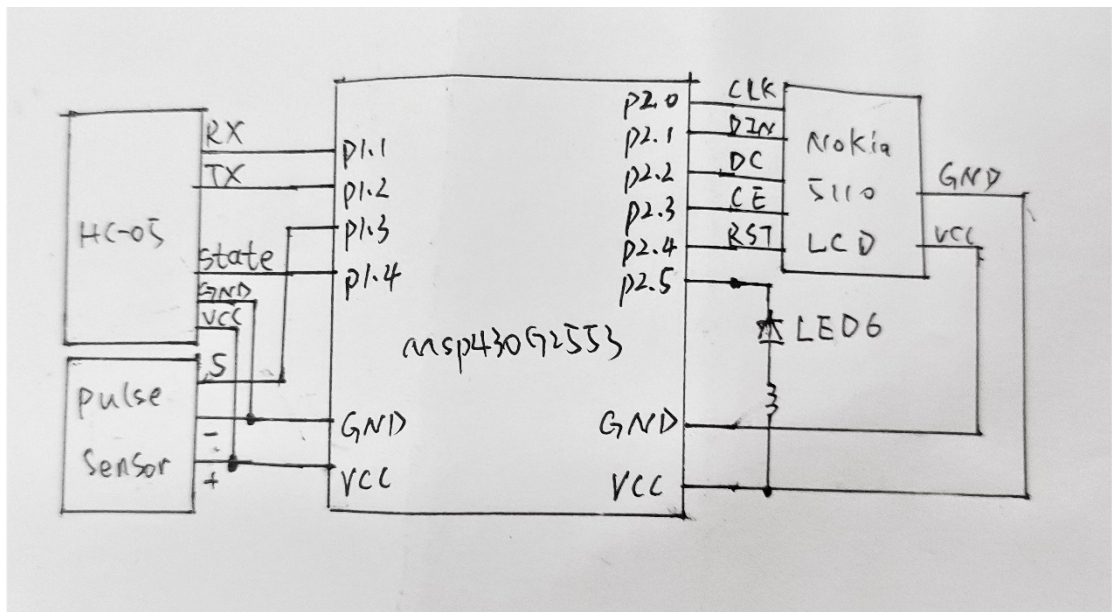
2 . HC-05 蓝牙模块

HC-05 蓝牙串口通信模块，是基于 Bluetooth Specification V2.0 带 EDR 蓝牙协议的数传模块。无线工作频段为 2.4GHz ISM，调制方式是 GFSK。模块最大发射功率为 4dBm，接收灵敏度-85dBm，板载 PCB 天线，可以实现 10 米距离通信。



HC-05 蓝牙模块实物图

3.1. 硬件原理图

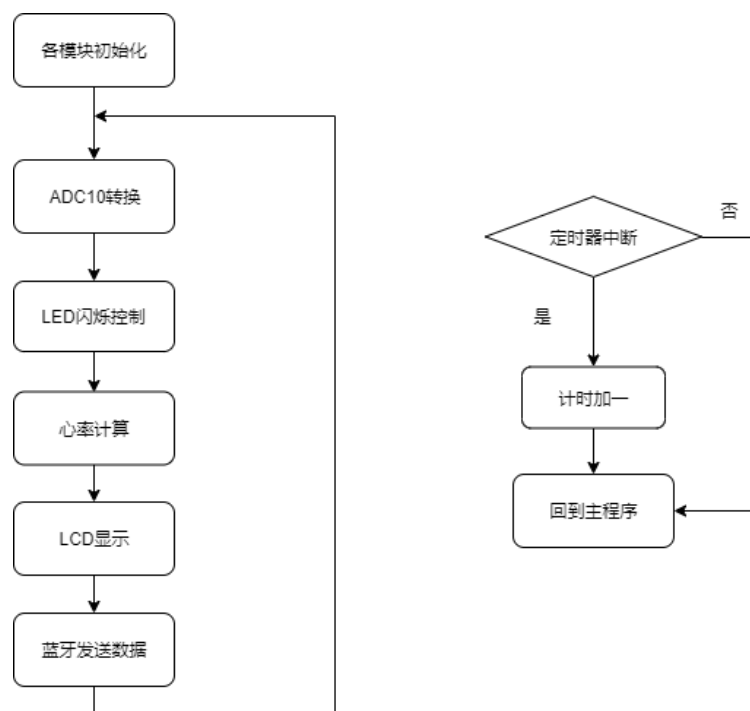


4. 软件设计

4.1. 软件模块划分

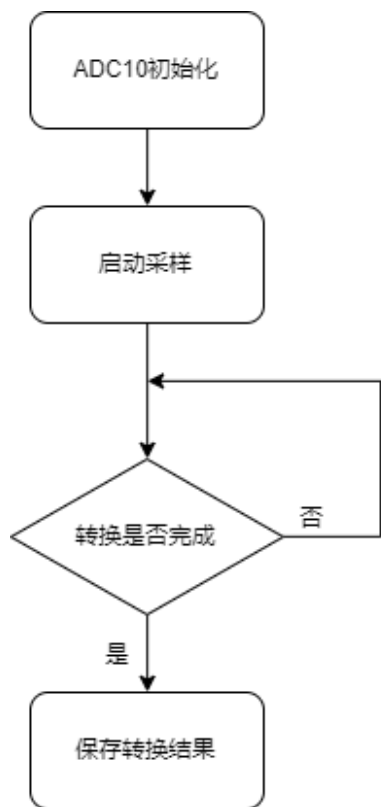
根据实现流程，可以分为 6 个模块，分别为模数转换模块、LED 控制模块、心率计算模块、蓝牙发送模块、LCD 显示模块、波形显示模块。

4.2. 总体流程图

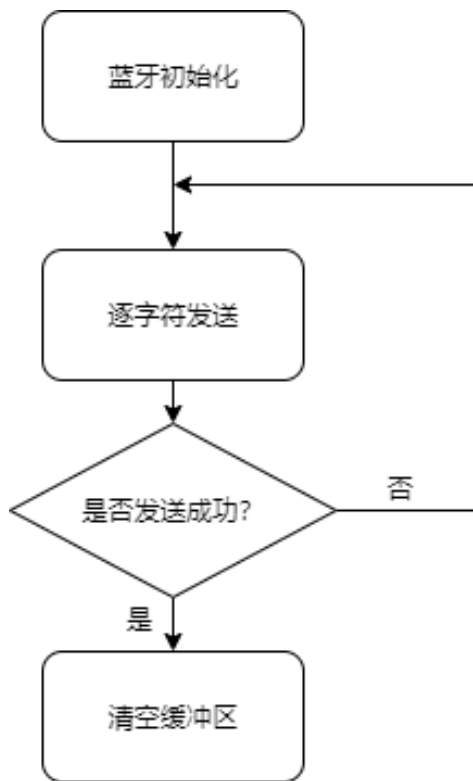


4.3. 各模块流程图

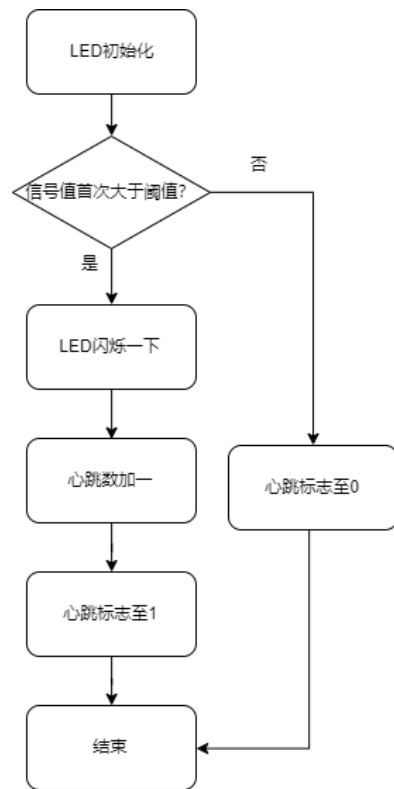
模数转换模块



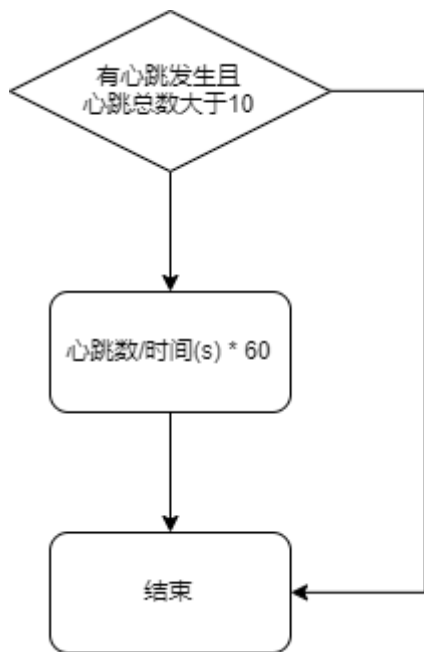
蓝牙发送模块



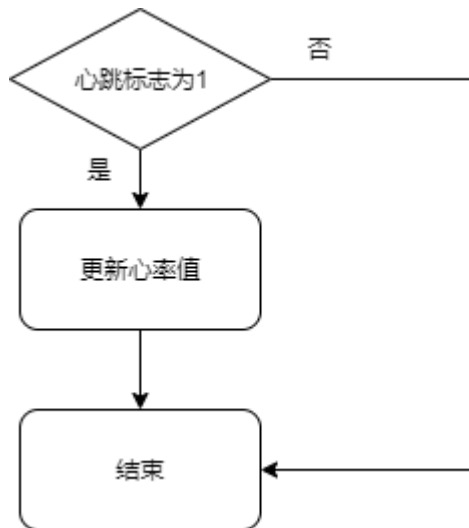
LED 控制模块



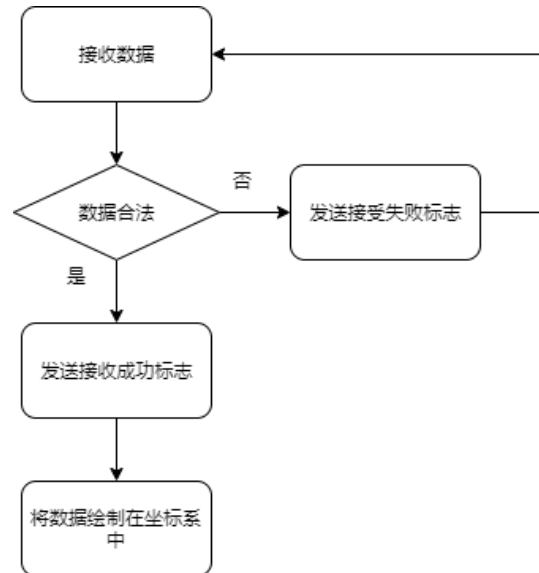
心率计算模块



LCD 显示模块



波形显示模块 (PC 上位机)



二、项目展示部分程序清单

1. 展示部分功能说明

1. LED 灯根据心跳的频率跳动;
每检测到一次心跳, LED 闪烁一下, 持续时间 1ms。
2. LCD 显示 BPM 的值;
 1. 手指未放置在传感器上时, LCD 显示“Please put your finger on the sensor and wait for a while.”
 2. 手指放置在传感器但是计算得到的心率值不正常 (低于 40 或高于 160), LCD 显示“Please wait for a while...”
 3. 手指放置在传感器且心率值正常, LCD 显示“BPM: 心率值”。
3. 蓝牙发送 signal、BPM、IBI 数据
上位机软件实时显示心电图, BPM、IBI 数据

2. 程序清单

1. main 函数

```
1  #include <msp430.h>
2  #include "ADC10.h"//ADC10模块
3  #include "LED.h"//LED模块
4  #include "USART0.h"//串口通信模块
5  #include "my_string.h"//使用到的字符串处理函数
6  #include "nokia5110.h"//LCD模块
7
8
9  unsigned int heart_signal, heart_signal_buffer;//心跳信号, 心跳信号缓存 (上一次的心跳信号)
10 unsigned int heartbeat_flag, bluetooth_state;//心跳标志, 蓝牙连接状态
11 unsigned int IBI, BPM;
12 unsigned int time_ms; //全局计时器 (检测到心跳时清零)
13
14 char info_to_send[10], buffer[10], info_to_print[20];//蓝牙发送字符串、缓冲字符串、LCD显示字符串
15
16 void init();//系统初始化
17 void GPIO_init();//GPIO初始化
18 void send_signal();//蓝牙发送心跳信号
19 void send_bpm();//蓝牙发送BPM、IBI
20 void TA0_init();//定时器初始化
21 void print_bpm();//LCD显示BPM值
22
23 void main(void) {
24     WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer
25     init();//系统初始化
26     while(1) {
27         bluetooth_state = (P1IN & BIT4);//检测蓝牙连接状态
28         LED_blink();//检测是否有心跳发生, LED闪烁
29         if(heartbeat_flag == 1) { //检测到有心跳发生
30             IBI = time_ms % 1000;//两次心跳发生的间隔 (IBI) 计算, 单位:ms
31             BPM = 6000 / IBI;//BPM计算, 单位:次/min
32             time_ms = 0;//清零全局计时器
33             print_bpm();//LCD显示BPM值
34
35             _DINT();//关中断总控位
36             send_bpm();//蓝牙发送BPM、IBI的值
37             _EINT();//开中断总控位
38         }
39         if(time_ms > 400) {
40             //长时间无心跳信号产生, LCD显示信息
41             LCD_write_english_string(0, 0, "Please put your finger on the sensor and wait for a while.");
42             time_ms = 0;//全局计时器清零
43         }
44     }
45 }
```

2. 1ms 中断 (完成心跳数据的采集和发送)

```
47 #pragma vector=TIMER0_A0_VECTOR
48 __interrupt void ISR_1ms() { //1ms中断
49     time_ms = time_ms + 1;
50     heart_signal_buffer = heart_signal;
51     heart_signal = ADC_get();//将采集到的心跳信号转化为数字量
52     send_signal();//蓝牙发送心跳信号signal
53 }
```

3. 其它函数

init() 系统初始化	<pre> 55 void init() { 56 ADC_init(); 57 USARTA0_init(); 58 GPIO_init(); 59 TA0_init(); 60 LCD_init(); 61 62 //上电显示信息 63 LCD_write_english_string(0, 0, 64 "Please put your finger on the sensor and wait for a while."); 65 } </pre>
GPIO_init() GPIO 初始化	<pre> 67 void GPIO_init() { 68 //LED P2.5 69 P2SEL &= ~BIT5; P2SEL2 &= ~BIT5; 70 P2DIR = BIT5; P2OUT = BIT5; 71 //ADC10 P1.3 72 P1SEL = BIT3; 73 //蓝牙连接状态 P1.4 74 P1SEL &= ~BIT4; P1SEL2 &= ~BIT4; 75 P1DIR &= ~BIT4; 76 } </pre>
TA0_init() 定时器初始化	<pre> 78 void TA0_init() { 79 BCSCTL1 = CALBC1_1MHZ; //初始化SMCLK为1MHz 80 DCOCTL = CALDCO_1MHZ; 81 BCSCTL2 = DIVS_0; //控制输出时钟不分频 82 //设置TA0定时器与TA捕捉/比较器0单元 83 TA0CTL = ID_0 + TACLRL + TASSEL_2 + MC_1; //设置TA控制寄存器的值 84 TA0CCR0 = 10869; //TA比较控制寄存器0值为10869 85 TA0CCTL0 = CCIE; //比较相等时CCR0中断允许 86 87 _EINT(); //允许中断 88 } </pre>
send_signal() 蓝牙发送心跳信号	<pre> 90 void send_signal() { 91 if(blueetooth_state != 0) { 92 //蓝牙已连接 93 //发送signal,数据格式为"S" + str(signal) 94 strcpy(info_to_send, "S"); 95 strcat(info_to_send, itoa(heart_signal, buffer)); 96 strcat(info_to_send, "\r\n"); 97 USARTA0_send(info_to_send); 98 } 99 } </pre>
send_bpm() 蓝牙发送BPM、IBI数据	<pre> 101 void send_bpm() { 102 if(blueetooth_state != 0) { 103 //蓝牙已连接 104 105 //发送BPM,数据格式为"B" + str(BPM) 106 strcpy(info_to_send, "B"); 107 strcat(info_to_send, itoa(BPM, buffer)); 108 strcat(info_to_send, "\r\n"); 109 USARTA0_send(info_to_send); 110 111 //发送IBI,数据格式为"Q" + str(IBI) 112 strcpy(info_to_send, "Q"); 113 strcat(info_to_send, itoa(IBI, buffer)); 114 strcat(info_to_send, "\r\n"); 115 USARTA0_send(info_to_send); 116 } 117 } </pre>

print_bpm() LCD 显示 BPM 值	<pre> 119 void print_bpm() { 120 LCD_clear(); 121 //LCD_write_english_string(5, 0, " "); 122 //LCD_write_english_string(5, 1, " "); 123 if((BPM <= 160) && (BPM >= 40)) { 124 //LCD显示BPM,数据格式为"BPM: " + str(BPM) 125 strcpy(info_to_print, "BPM: "); 126 strcat(info_to_print, itoa(BPM, buffer)); 127 strcat(info_to_print, " "); 128 LCD_write_english_string(0, 1, info_to_print); 129 } else { 130 LCD_write_english_string(0, 1, "Please wait for a while..."); 131 } 132 } </pre>
LED_blink() 检测是否有 心跳产生, LED 闪烁 (位于 LED.c)	<pre> 7 void LED_blink() { 8 if((heart_signal >= THRESHOLD) && (heart_signal_buffer <= THRESHOLD)) { 9 //检测到心跳 10 blink(); 11 heartbeat_flag = 1; 12 } else { 13 heartbeat_flag = 0; 14 } 15 } </pre>
blink() LED 闪烁 (位于 LED.c)	<pre> 17 //LED闪烁函数 18 void blink() { 19 P2OUT &= ~BIT5; 20 delay_ms(1); 21 P2OUT = BIT5; 22 } </pre>
delay_ms(uin t) 1ms 延时 (位于 LED.c)	<pre> 24 //1ms delay函数 25 void delay_ms(unsigned int time_ms) { 26 while(time_ms--) 27 __delay_cycles(1087); 28 } </pre>

三、调试过程中遇到的难点问题及解决方案

1. 蓝牙传 BPM、IBI 输数据所需时间较长，在这个过程中触发 1ms 中断，中断子程会采集心率数据并通过蓝牙发送 Signal（心跳信号）数据，但这样会破坏预先定义好的通信协议，使得 PC 机解码不了 BPM、IBI 和 signal 的值。解决方案是，在发送 BPM、IBI 时禁止中断，这样可能会少发送几次 signal 的数据，但是能够有效解决上述问题。
2. 数据采集频率太低，导致绘制出的心电图波形不够平滑，解决方案：使用 1ms 中断采集数据，1ms 采集一次数据。

四、项目的不足之处和可改进之处

1. 项目的功能有些单一，可以考虑加入血氧含量测量等其他健康相关功能。
2. 心电图需要通过 PC 机上位机软件进行绘制，不方便。可以考虑加入一块 OLED 屏幕，直接通过单片机显示波形。
3. 项目所用心率传感器受环境影响太大，测量质量时好时坏，可以考虑使用更加优质的传感器。