



HRS3301 Programmer's Guide

HRS3301是一个光学数字式心率传感器，通过I2C接口读取原始数据，在MCU的程序中计算出心率的值。本文主要描述如何在软件层面实现心率的计算。建议的主程序流程图如下：

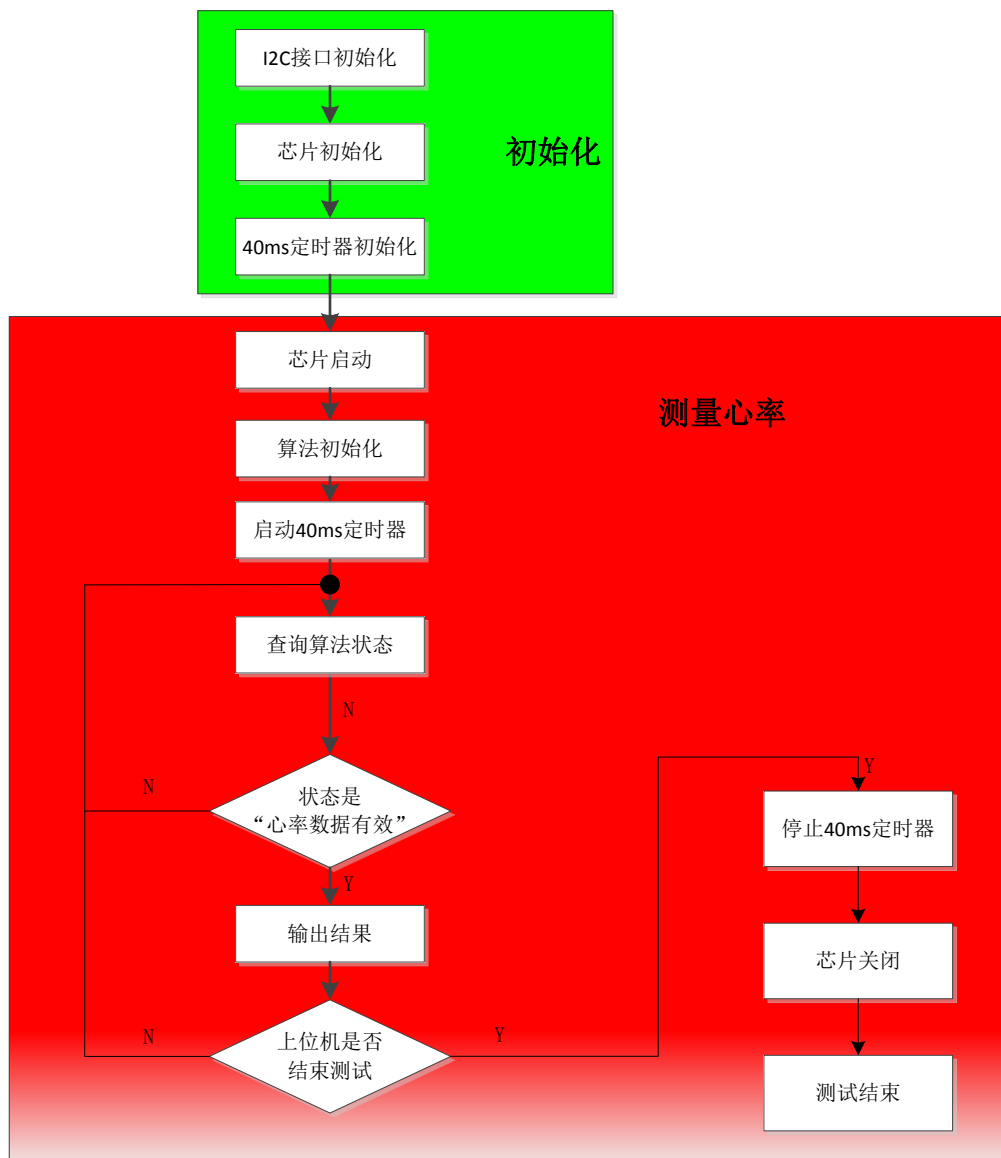


Figure 1. 主程序建议流程图

函数和主要变量说明

HRS3301 主要调用的函数和使用的变量如下表：

Table 1. 函数和主要变量说明



函数	描述
bool Hrs3300_chip_init();	初始化芯片配置，请在程序初始化阶段调用一次
void Hrs3300_chip_enable();	使能芯片，传感器开始工作，在心率测量的开始调用
void Hrs3300_chip_disable();	关闭芯片，传感器进入休眠
void Hrs3300_alg_open();	打开心率算法，初始化算法
void Hrs3300_bp_alg_open();	打开血压算法，初始化算法
void Hrs3300_set_exinf(uint8_t age, uint8_t height, uint8_t weight, uint8_t gender, uint8_t ref_sbp, uint8_t ref_dbp);	传入血压特征值: age: 年龄, height (身高): cm Weight (体重): kg, gender (性别): 0 男, 1 女, sbp, dbp (为参考血压值, 默认可以为0)
void heart_rate_meas_timeout_handler (void * p_context)	心率 40ms 定时处理函数, 主要完成将 PPG 信号和 gsensor 信号传送到算法和输出算法结果的作用
void blood_presure_meas_timeout_handler (void * p_context)	血压 20ms 定时处理函数, 主要完成将 PPG 信号传送到算法和输出算法结果的作用
uint16_t Hrs3300_read_hrs(void)	读取 PPG 信号函数
hrs3300_results_t Hrs3300_alg_get_results();	查询心率算法状态, 在测量过程中应该一直用此函数查询
hrs3300_bp_results_t Hrs3300_alg_get_bp_results();	查询血压算法状态, 在测量过程中应该一直用此函数查询
bool Hrs3300_alg_send_data(int16_t new_raw_data, int16_t als_raw_data, int16_t gsen_data_x, int16_t gsen_data_y, int16_t gsen_data_z)	定时处理函数, 在 40ms 定时器中调用此函数 其中的 new_raw_data 是 PPG 的原始数据, 调用函数 Hrs3300_read_hrs()可获得, als_raw_data 为光强度信息, gsen_data_x, gsen_data_y, gsen_data_z 是 gsensor 数据, 格式为补码, 对应的满量程是±2g
bool Hrs3300_bp_alg_send_data(int16_t	定时处理函数, 在 20ms 定时器中调用此函数



new_raw_data)	
<pre>typedef enum { MSG_ALG_NOT_OPEN = 0x01, MSG_NO_TOUCH = 0x02, MSG_PPG_LEN_TOO_SHORT = 0x03, MSG_HR_READY = 0x04, MSG_ALG_TIMEOUT = 0x05, MSG_SETTLE = 0x06 } hrs3300_msg_code_t; typedef struct { hrs3300_msg_code_t alg_status; uint32_t data_cnt; uint8_t hr_result; uint8_t hr_result_qual; } hrs3300_results_t;</pre>	结构 hrs3300_results_t 用于表示算法状态和结果, 具体含义如下: hrs3300_msg_code_t 表示目前算法状态: MSG_ALG_NOT_OPEN: 算法未打开 MSG_NO_TOUCH: 脱落 MSG_PPG_LEN_TOO_SHORT: 数据不足 MSG_HR_READY: 心率数据有效 MSG_ALG_TIMEOUT: 超时 MSG_SETTLE: 正在稳定过程中 alg_status: 表示目前算法的状态 data_cnt: 表示接收到的原始数据个数, 每 40ms 增加 1 hr_result: 计算的心率值 hr_result_qual: 上报的心率值的质量
<pre>typedef enum { MSG_BP_ALG_NOT_OPEN = 0x01, MSG_BP_NO_TOUCH = 0x02, MSG_BP_PPG_LEN_TOO_SHORT = 0x03, MSG_BP_READY = 0x04, MSG_BP_ALG_TIMEOUT = 0x05, MSG_BP_SETTLE = 0x06 } hrs3300_bp_msg_code_t; typedef struct { hrs3300_bp_msg_code_t bp_alg_status;</pre>	结构 hrs3300_bp_results_t 用于表示算法状态和结果,具体含义如下: hrs3300_bp_msg_code_t 表示目前算法状态: MSG_BP_ALG_NOT_OPEN: 算法未打开 MSG_BP_NO_TOUCH: 脱落 MSG_BP_PPG_LEN_TOO_SHORT: 数据不足 MSG_BP_READY: 血压数据有效 MSG_BP_ALG_TIMEOUT: 超时 MSG_SETTLE: 正在稳定过程中 bp_alg_status: 表示目前血压算法的状态 sbp: 收缩压计算结果 dbp: 舒张压计算结果



uint8_t	sbp;	
uint8_t	dbp;	
} hrs3300_bp_results_t;		

心率移植步骤

1. 根据平台先填写 hrs3300.c 中的 i2c 读写函数， i2c 地址为 0x44(7bit)

2. 初始化

芯片上电请延时 5ms

```
Hrs3300_chip_init(); // initial hrm chip
```

```
Hrs3300_chip_enable();
```

```
Hrs3300_alg_open(); // hrs alg initial
```

3. 开始定时器 40ms 调用

```
heart_rate_meas_timeout_handler(void * p_context)
```

大概 10s 输出心率值 alg_results.hr_result

4. 停止心率测试

先停止定时器

```
Hrs3300_chip_disable();
```

说明：如果需要测试动态心率请合入 gsensor 三轴数据， gsensor 数据格式统一为原始数据 16 位（如为 12 位请左移 4 位）， 格式为补码，请检查：静止状态下±2G 对应最大量程为：±16384， ±4G 对应最大量程为：±8192， ±8G 对应最大量程为：±4096

如果是±8G 请保持不变， 如果是±4G 请除以 2， 如果是±2G 请除以 4

PS：只需检查静止时传入算法函数 Hrs3300_alg_send_data 最大轴是否对应为±4096 左右即可

血压移植步骤

一. 小血压版本

1.移植步骤跟心率移植步骤相同

2.打开 BP_CUSTDOWN_ALG_LIB 宏

大概 20s 输出血压值 bp_alg_results.sbp, bp_alg_results.dbp

二. 大血压版本移植



1.初始化

芯片上电请延时 5ms

```
Hrs3300_chip_init();    // initial hrm chip
```

```
Hrs3300_chip_enable();
```

```
Hrs3300_set_exinf(0,0,0,0,0,0);
```

```
Hrs3300_bp_alg_open(); // bp alg initial
```

2.开始定时器 20ms 调用

```
void blood_presure_meas_timeout_handler(void * p_context)
```

大概 30s 输出血压值 bp_alg_results.sbp, bp_alg_results.dbp

3.停止血压测试

先停止定时器

```
Hrs3300_chip_disable();
```

程序示例

Main.c

.....

```
#include "hrs3300.h"
```

```
#include "hrs3300_alg.h"
```

```
#include "hrs3300_i2c.h"
```

.....

```
int main(void)
```

```
{    // Initialize.
```

```
    twi_master_init();    // initial i2c
```

```
    lis3dh_opr_init();    // initial gsensor
```

```
    opr_oled_init();    // initial oled
```

```
    timers_init();
```

```
    // Start heart rate measurement
```

```
    Hrs3300_chip_init();    // initial hrm chip
```



```
Hrs3300_chip_enable();  
  
Hrs3300_alg_open();  
  
application_timers_start();  
  
} //main
```



版本历史

版本	日期	说明
0.1	2016-11-10	初始版本
0.2	2016-11-29	将 timer 和 I2C 部分移出库
0.3	2017-2-13	动态算法适配
0.4	2017-3-2	加入血压函数和数据结构
0.5	2017-3-4	加入心率和血压移植步骤
0.6	2017-3-16	加入心率和 gsensor 移植说明
0.7	2017-8-5	加入小版本血压移植说明
0.8	2017-10-15	修改 gsensor 参数为 8G 量程