编　　制：北京万邦易嵌科技有限公司-嵌入式研发部肖龙

版　　本：V2.0

编制日期：2017 年 1 月 10 日

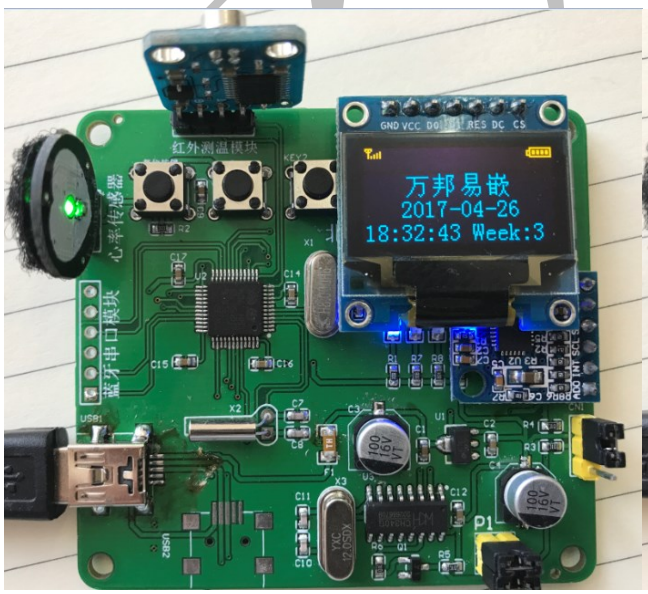
修改日期：2017 年 5 月 13 日

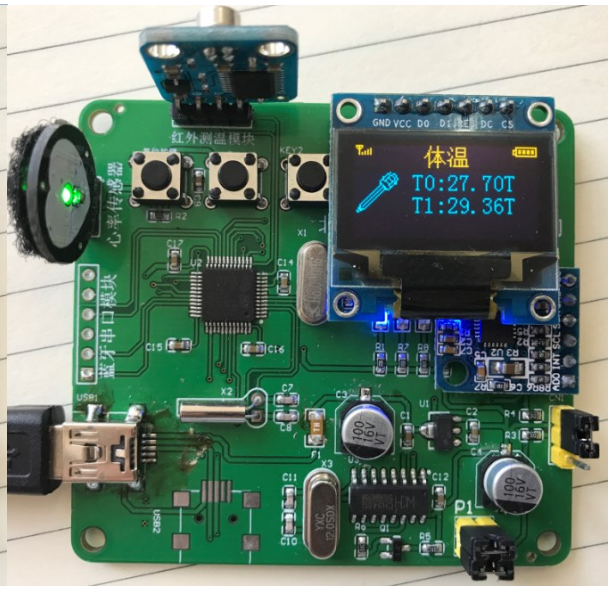
智能穿戴开发板用户手册

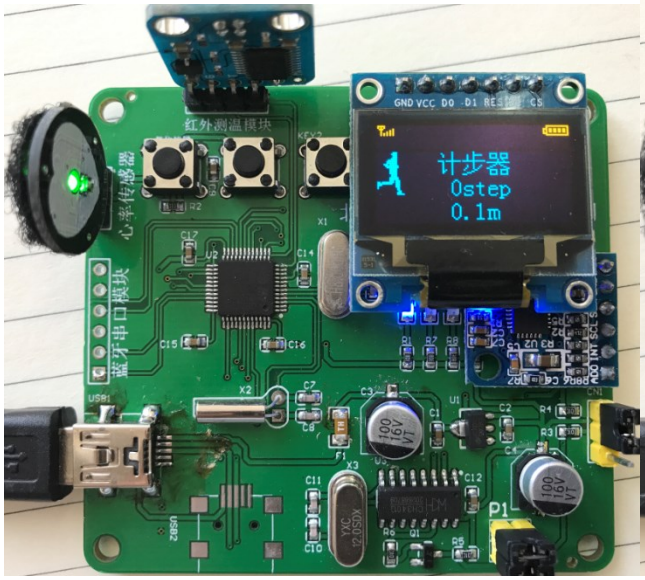
一、智能穿戴开发板功能介绍

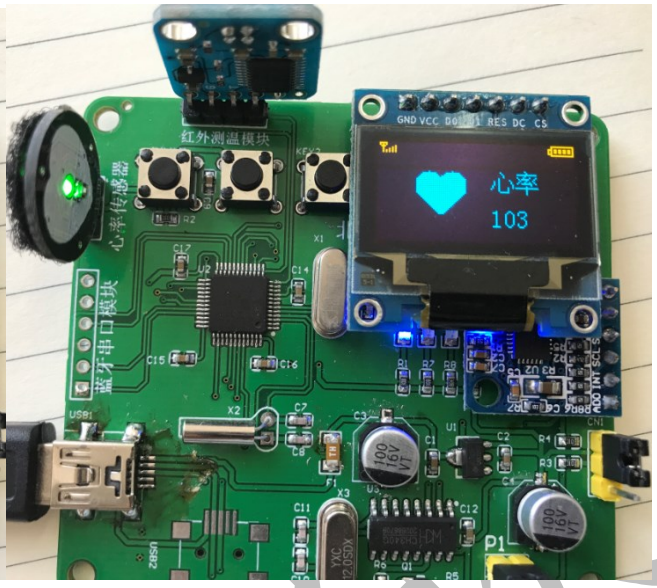
本开发板采用了 STM32F103C8T6 作为核心 CPU，包含了心率传感器、红外线测温传感器、计步传感器、OLED 显示屏等主要模块。按键包含了 3 个，1 个按键用于复位操作，其他 2 个按键用于切换显示屏的显示页面。LED 灯包含 3 盏，1 盏用于电源指示灯，其他 2 个可以用于表示程序运行状态。

开发板的电源可以通过 USB 进行供电，程序可通过串口下载(USB 转串口)。









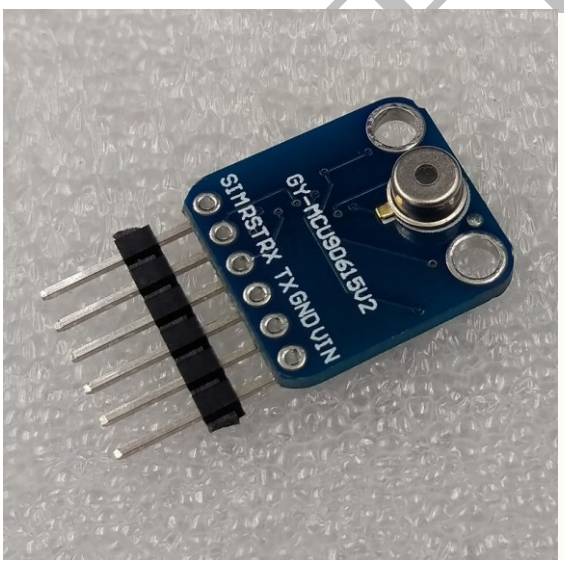
二、开发板功能模块使用介绍

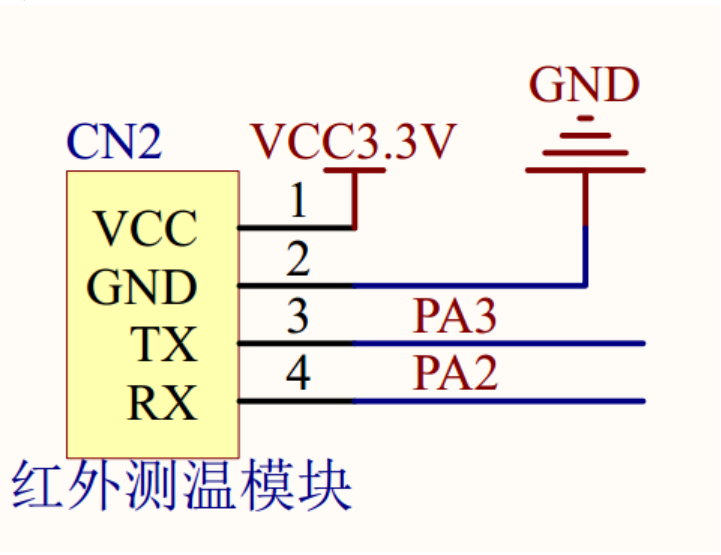
2.1 红外线测温模块使用说明

2.1.1 概述

GY-MCU90615 是一款低成本红外温度模块。工作电压 3-5v 功耗小，体积小。其工作原理， 是通过单片机读取红外温度度数据，串口（TTL 电平）通信方式输出。

串口的波特率有 9600bps 与 115200bps 有连续输出与询问输出两种方式，可适应不同的工作环境，与所有的单片机及电脑连接。





2.1.2 产品特点 技术参数

（1）、体积小

（2）、高性价比

（3）、 串口通信格式

2.1.3 产品应用

（1）、 人体测温

（2）、 发热物体表面温度检测

（3）、 非接触温度检测

2.1.4 引脚说明

Pin1	VCC	电源+ （3v-5v）
Pin 2	GND	电源地
Pin3	TX	串口数据发送 TXD
Pin 4	RX	串口数据接收 RXD
Pin 5	RST	内部使用，不需要连接，悬空
Pin 6	SIM	内部使用，不需要连接，悬

2.1.5 通信协议

- 串口发送命令字节：

（1）、串口通信参数（默认波特率值 115200 bps，可通过软件设定）

波特率： 9600 bps 校验位： N 数据位： 8 停止位： 1

波特率： 115200 bps 校验位： N 数据位： 8 停止位： 1

（2）、模块输入命令，由外部控制器发送至 GY-MCU90615 模块（十六进制）

指令格式：帧头+指令+校验和(8bit)（如自动读取温度指令=**0xA5+0x45+0xEA**）

初始化代码示例：

```
UsartInit(9600);
SendByte(0xA5);
SendByte(0x45);
SendByte(0xEA);
```

- 串口接收

（1）、串口通信参数（默认波特率值 115200 bps，可通过软件设定）

波特率： 9600 bps 校验位： N 数据位： 8 停止位： 1

波特率： 115200 bps 校验位： N 数据位： 8 停止位： 1

（2）、模块输出格式，每帧包含 9 个字节（十六进制）：

①.Byte0: 0x5A 帧头标志

②.Byte1: 0x5A 帧头标志

③.Byte2: 0X45 本帧数据类型（0X45：温度数据）

④.Byte3: 0x04 数据量（以下 4 个数据 2 组为例）

⑤.Byte4: 0x00~0xFF 数据 1 高 8 位

⑥.Byte5: 0x00~0xFF 数据 1 低 8 位

⑦.Byte6: 0x00~0xFF 数据 2 高 8 位

⑧.Byte7: 0x00~0xFF 数据 2 低 8 位

⑨.Byte8: 0x00~0xFF 校验和（前面数据累加和，仅留低 8 位）

（3）、数据计算方法

温度计算方法 ：

温度=高 8 位<<8|低 8 位（结果为实际角度乘以 100）

例：发送指令： A5 45 EA ，接收到一帧数据：**<5A- 5A- 45- 04- 0C- 78- 0D- 19- A7 >**

表示 TO（有符号 16bit，表示目标温度）： TO=0x0C78/100=31.92 ℃

表示 TA（有符号 16bit，表示环境温度）： TO=0x0D19/100=33.53 ℃

- 使用方法

该模块为串口输出数据，使用者通过串口连接后，发送输出指令，例如 0xA5+0x45+0xEA 给模块，模块将连续输出温度数据；如想通过查询输出可发送 0xA5+0x15+0xBA 给模块，每发送一次，模块将返回一次温度数据，查询频率应低于 10hz，如需高于 10hz 请使用连续输出模式，即发送 0xA5+0x45+0xEA 指令；

- 模块连接

注意：模块 I/O 是 TTL 电平，可以直接与单片机串口连接，可以直接与 PL2303,CH340,FT232 等芯片连接，但不能与电脑九针串口直接连接。

2.1.6 接收温度程序设计

```
u8 Receive_ok;
u8 rebuf[20]={0};
void RxTempInfo(void)
{
    static uint8_t i=0;
    if(USART2->SR&1<<5)        //判断接收标志
    {
        rebuf[i++]=USART2->DR;//读取串口数据，同时清接收标志
        if(rebuf[0]!=0x5a)      //帧头不对
            i=0;
        if((i==2)&&(rebuf[1]!=0x5a))//帧头不对
            i=0;

        if(i>3)//i 等于 4 时，已经接收到数据量字节 rebuf[3]
        {
            if(i!=(rebuf[3]+5))//判断是否接收一帧数据完毕
                return ;
            switch(rebuf[2])    //接收完毕后处理
            {
                case 0x45:
```

```
                    if(!Receive_ok)//当数据处理完成后才接收新的数据
                    {
                        Receive_ok=1;//接收完成标志
                    }
                    break;
                case 0x15:break;
                case 0x35:break;
            }
            i=0;//缓存清 0
        }
    }
}
```

2.1.7 转换温度示例

```
void GetTempInfo(void)
{
    float TO=0,TA=0;
    u8 sum=0,i=0;
    for(sum=0,i=0;i<(rebuf[3]+4);i++)
    {
        sum+=rebuf[i];
    }
    if(sum==rebuf[i])//校验和判断
    {
        TO=(float)((rebuf[4]<<8)|rebuf[5])/100;   //得到真实温度
        TA=(float)((rebuf[6]<<8)|rebuf[7])/100;   //得到真实温度
    }
    printf("TO: %f\r\n",TO);
    printf("TA: %f\r\n",TA);
}
```

2.2 OLED 显示屏使用说明

2.2.1 OLED 简介

OLED，即有机发光二极管（ Organic Light Emitting Diode）。 OLED 由于同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、可用于挠曲性面板、使用温度范围广、构造及制程较简单等优异之特性，被认为是下一代的平面显示器新兴应用技术。

LCD 都需要背光，而 OLED 不需要，因为它是自发光的。这样同样的显示 OLED 效果要来得好一些。以目前的技术，OLED 的尺寸还难以大型化，但是分辨率确可以做到很高。在此我们使用的是中景园电子的 0.96 寸 OLED 显示屏，该屏有以下特点：

1）0.96 寸 OLED 有黄蓝，白，蓝三种颜色可选；其中黄蓝是屏上 1/4 部分为黄光，下 3/4 为蓝；而且是固定区域显示固定颜色，颜色和显示区域均不能修改；白光则为纯白，也就是黑底白字；

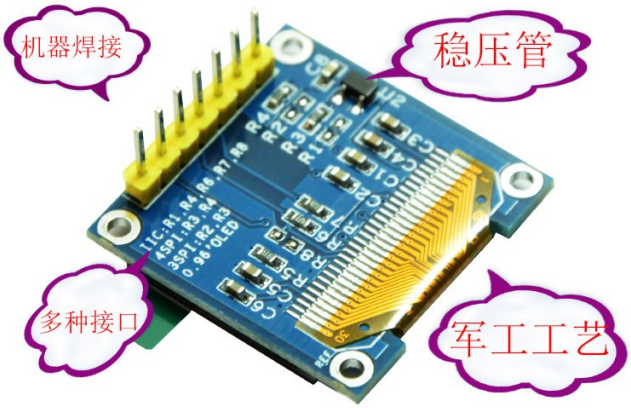
蓝色则为纯蓝，也就是黑底蓝字。

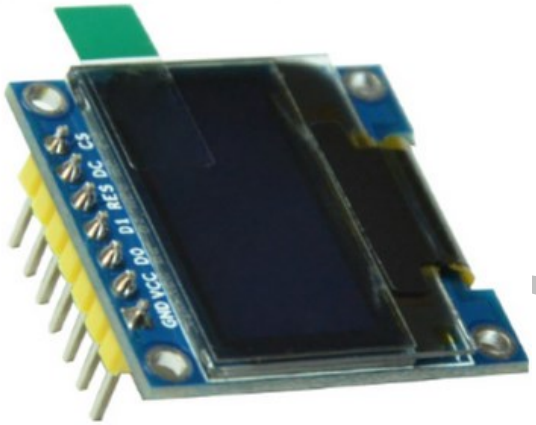
2）分辨率为 128*64

3）多种接口方式；OLED 裸屏总共种接口包括：6800、8080 两种并行接口方式、3 线或 4 线的串行 SPI 接口方式、 IIC 接口方式（只需要 2 根线就可以控制 OLED 了！），这五种接口是通过屏上的 BS0~BS2 来配置的。

4）OLED 屏开发了两种接口的 Demo 板，接口分别为七针的 SPI/IIC 兼容模块，四针的 IIC 模块。

2.2.2 0.96 寸 OLED 屏外观

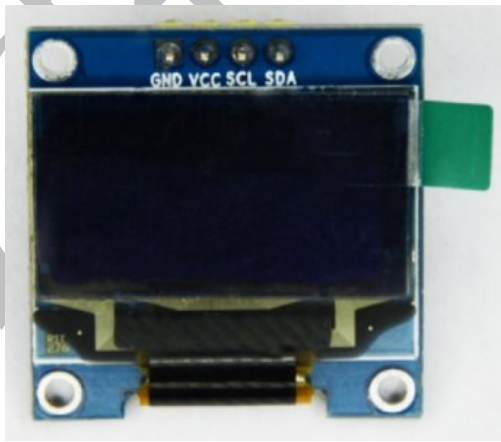




2.2.3 IIC 接口模块

- 模块接口定义

1. GND 电源地
2. VCC 电源正（3～5.5V）
3. SCL OLED 的 D0 脚，在 IIC 通信中为时钟管脚
4. SDA OLED 的 D1 脚，在 IIC 通信中为数据管脚



2.2.4 SPI/IIC 接口模块

- 模块接口定义

1. GND 电源地
2. VCC 电源正（3～5.5V）
3. D0 OLED 的 D0 脚，在 SPI 和 IIC 通信中为时钟管脚
4. D1 OLED 的 D1 脚，在 SPI 和 IIC 通信中为数据管脚

5. RES OLED 的 RES#脚，用来复位（低电平复位）

6. DC OLED 的 D/C#E 脚，数据和命令控制管脚

7. CS OLED 的 CS#脚，也就是片选管脚

2.2.5 0.96 寸 OLED 驱动 IC

本屏所用的驱动 IC 为 **SSD1306**；其具有内部升压功能；所以在设计的时候不需要再专一设计升压电路；当然了本屏也可以选用外部升压，具体的请详查数据手册。

SSD1306 的每页包含了 128 个字节，总共 8 页，这样刚好是 **128*64** 的点阵大小。这点与 1.3 寸 OLED 驱动 IC SSD1106 稍有不同，SSD1106 每页是 132 个字节，也是 8 页。所以在用 0.96 寸 OLED 移植 1.3 寸 OLED 程序的时候需要将 0.96 寸的显示地址向右偏移 2，这样显示就正常了；否则在用 1.3 寸的时候 1.3 寸屏右边会有 4 个像素点宽度显示不正常或是全白，这点大家注意一下。其它的 SSD1306 和 SSD1106 区别不大。

2.2.6 七针 SPI/IIC OLED 模块使用方法

七针 SPI/IIC OLED 模块共有七个管脚，1～7 分别为 GDN、VCC、D0、D1、RES、DC、CS 此模块支持四线 SPI、三线 SPI、IIC 接口；由 OLED 的数据手册我们可以知道 0.96 寸 OLED 裸屏是支持四种五种不同接口的；除了前面的三种还有 6800、8080 并口方式；由于这两种接口占用数据线比较多；而且不太常用，所以模块在设计的时候没有引出来。

10 11 12	BS0 BS1 BS2	I	**Communicating Protocol Select** These pins are MCU interface selection input. See the following table:

	BS0	BS1	BS2
I^2C	0	1	0
3-wire SPI	1	0	0
4-wire SPI	0	0	0
8-bit 68XX Parallel	0	0	1
8-bit 80XX Parallel	0	1	1

由上图可以看出来；模块的通信接口是通过 BS0,BS1,BS2 三个管脚来配置的。

- 本店所设计的模块默认是 SPI 接口：

1. GND 电源地

2. VCC 电源正（3～5.5V）

3. D0 OLED 的 D0 脚，在 SPI 和 IIC 通信中为时钟管脚

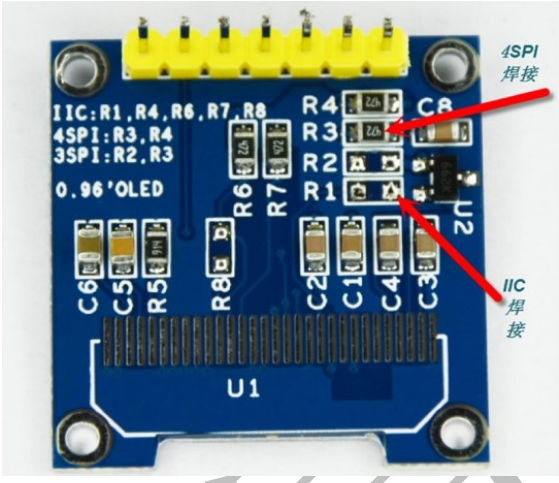
4. D1 OLED 的 D1 脚，在 SPI 和 IIC 通信中为数据管脚

5. RES OLED 的 RES#脚，用来复位（低电平复位）

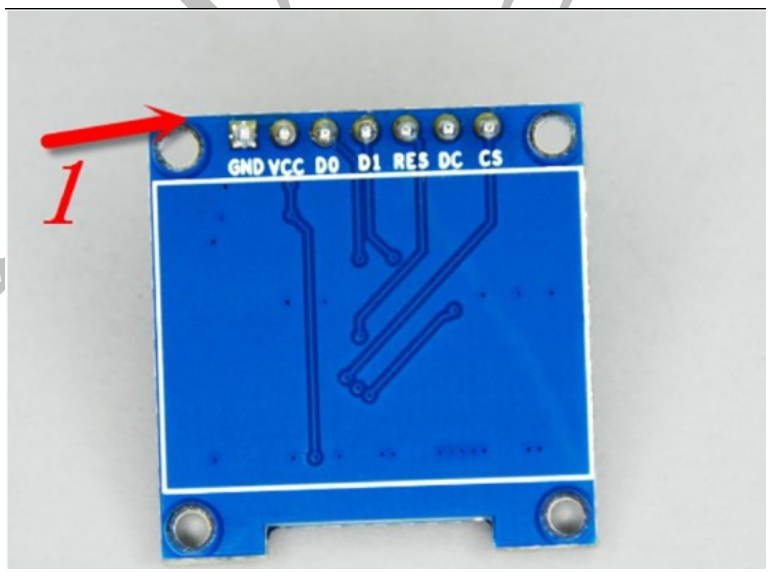
6. DC OLED 的 D/C#E 脚，数据和命令控制管脚

7. CS OLED 的 CS#脚，也就是片选管脚

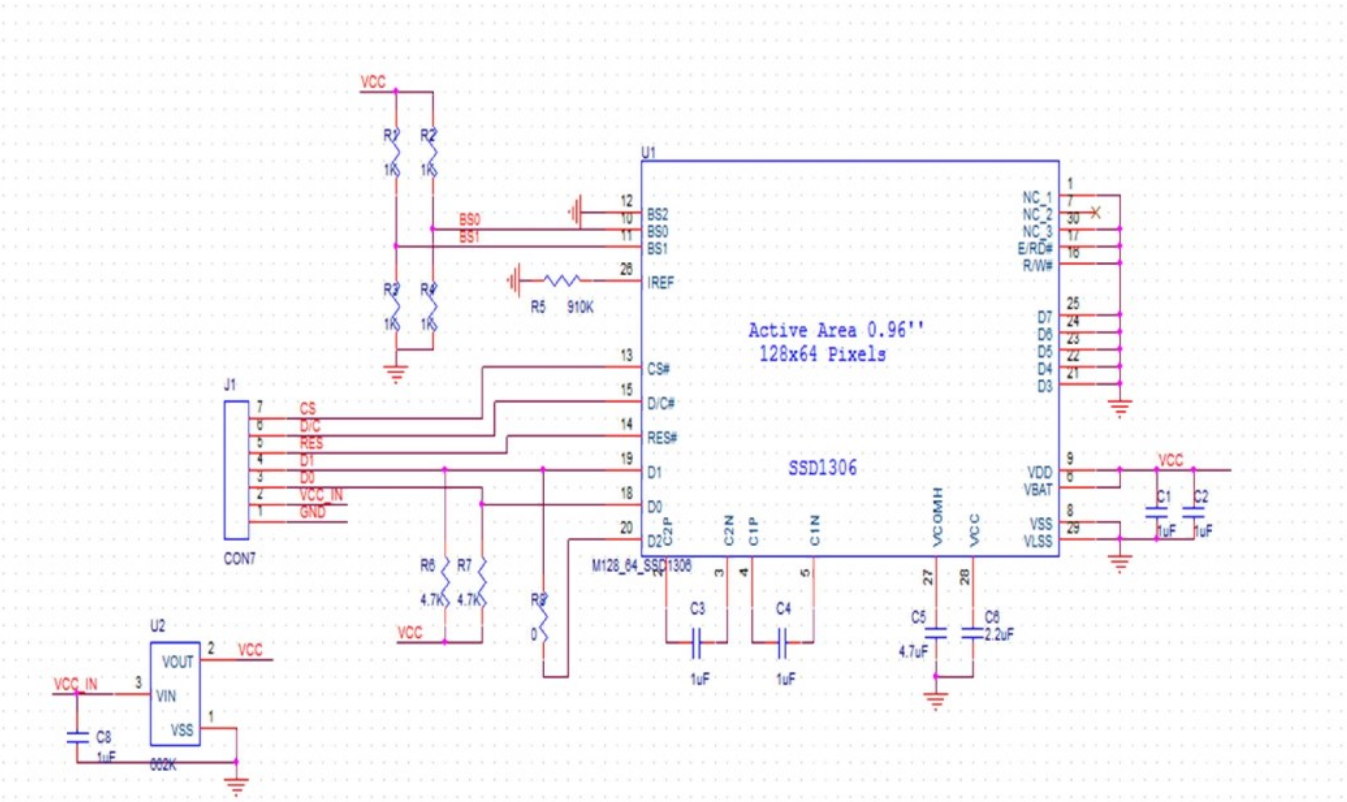
大家在使用的时候一定注意，如下图所示。在 SPI 接口中 R1,R2,R8 三个电阻是不焊接的，如果大家想用 IIC 接口的话需要将 R3 换到 R1 上，R8 可以焊接也可不焊接。



七针模块正面丝印如下，大家在接线的时候要注意不要接错了；特别是想用 IIC 接口的朋友们，在 IIC 接口中需要将 RES 接高电平，可以与 VCC 对接,使 OLED 复位脚一直操持高电平，也就是不复位的状态；同时需要将 DC,CS 接电源地；此时 IIC 通信中只需要 GND ,VCC,D0(时钟信号)，D1(数据信号)四根线了。如果大家感觉这样比较麻烦；建议大家直接选用四针的 IIC 接口模块

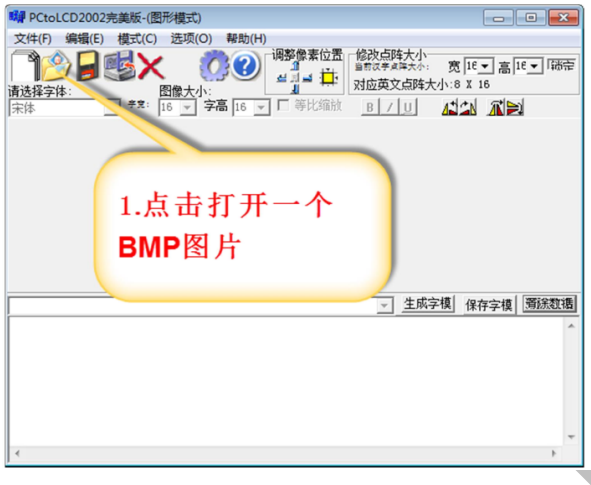


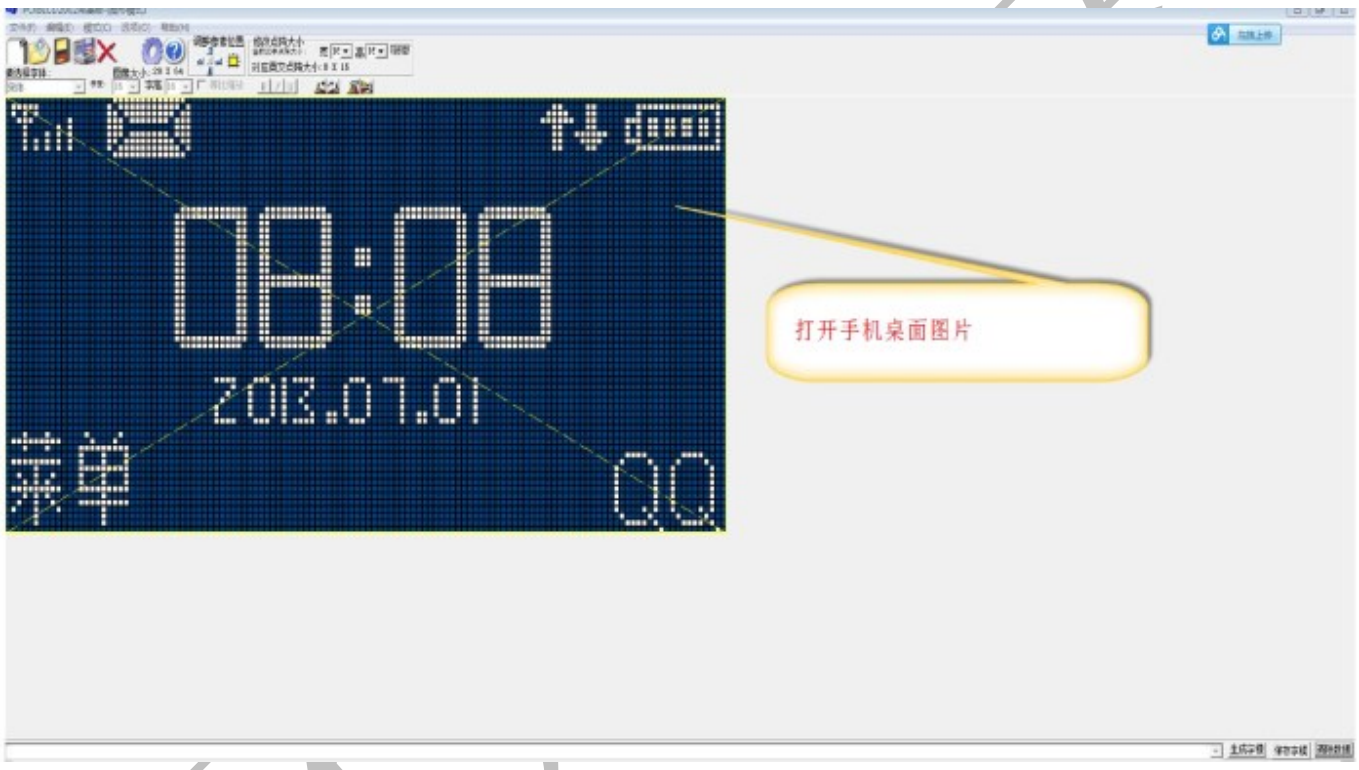
2.2.7 OLED 显示屏原理图

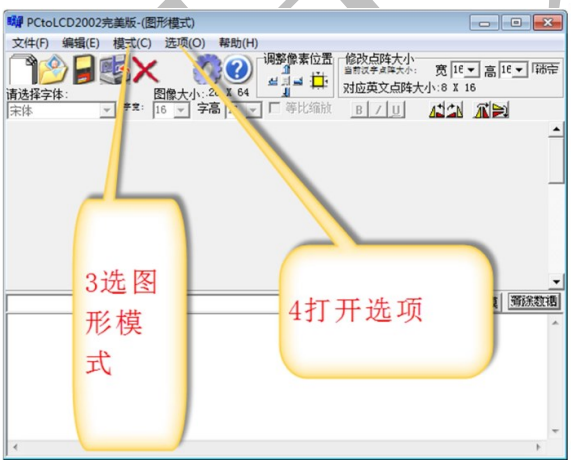


2.2.8 图片取模

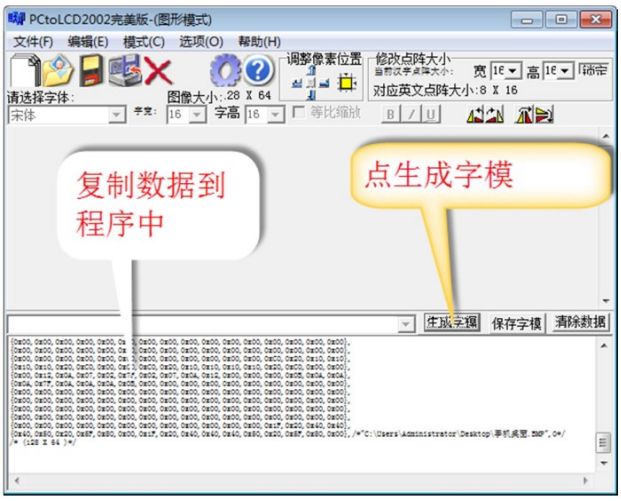
打开 PCtoLCD2002.exe 软件


PCtoLCD2002完美版-(图形模式)
文件(F) 编辑(E) 模式(C) 选项(O) 帮助(H)
1.点击打开一个
BMP图片
生成字模
保存字模

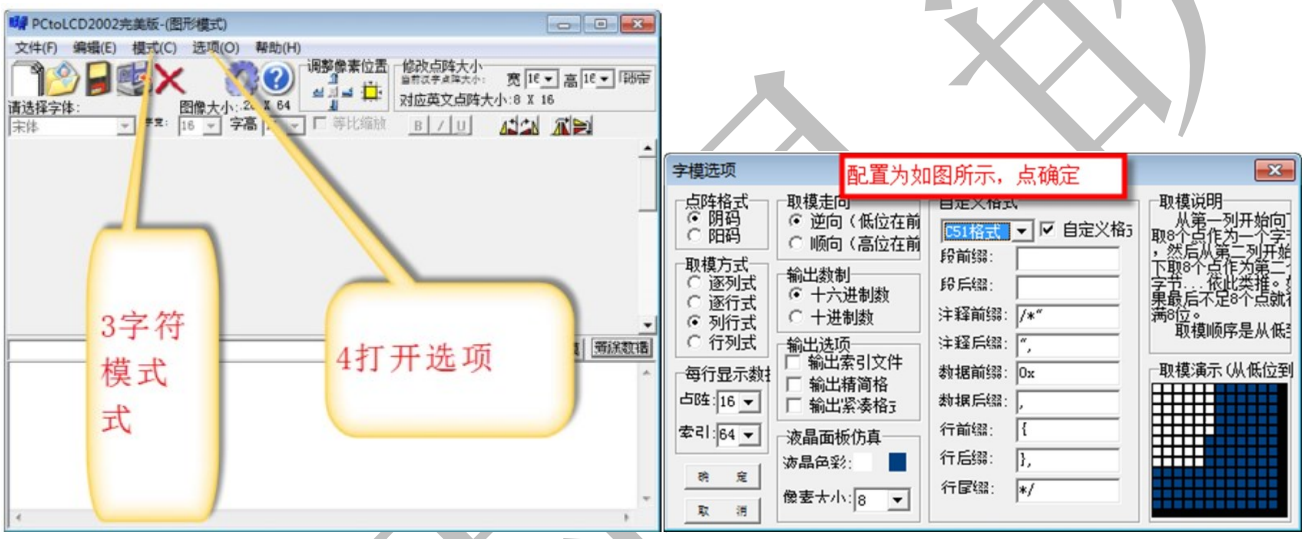

08:08
2013.07.01
菜单
打开手机桌面图片


PCtoLCD2002完美版-(图形模式)
3选图形模式
4打开选项


字模选项
如图
点阵格式
阴码
阳码
取模方式
逐列式
逐行式
列行式
行列式
取模走向
逆向（低位在前
顺向（高位在前
输出数制
十六进制数
十进制数
输出选项
输出索引文件
输出精简格
液晶面板仿真
液晶色彩:
像素大小:
自定义格式
C51格式
段前缀:
段后缀:
注释前缀:
注释后缀:
数据前缀:
数据后缀:
行前缀:
行后缀:
行尾缀:
取模说明
取模演示（从低位到
确 定
取 消



2.2.9 汉字字模的生成方法



2.2.10 OLED 程序移植

主要移植 oled.h 文件中定义的接口。将 define 后面替换为自己开发板的对应的引脚即可。

```
//-----------------OLED端口定义----------------
#define OLED_SCLK_Clr() GPIOA->ODR&=~(1<<5)
#define OLED_SCLK_Set() GPIOA->ODR|=1<<5

#define OLED_SDIN_Clr() GPIOA->ODR&=~(1<<7)
#define OLED_SDIN_Set() GPIOA->ODR|=1<<7

#define OLED_RST_Clr() GPIOB->ODR&=~(1<<0) //RES
#define OLED_RST_Set() GPIOB->ODR|=1<<0

#define OLED_DC_Clr() GPIOB->ODR&=~(1<<1)//DC
#define OLED_DC_Set() GPIOB->ODR|=1<<1

#define OLED_CS_Clr()  GPIOA->ODR&=~(1<<4)//CS
#define OLED_CS_Set()  GPIOA->ODR|=1<<4

#define OLED_CMD  0 //写命令
#define OLED_DATA 1 //写数据
```

2.2.11 关键的函数

//向 SSD1106 写入一个字节。
//dat:要写入的数据/命令
//cmd:数据/命令标志 0,表示命令;1,表示数据;

```
void OLED_WR_Byte(u8 dat,u8 cmd)
{
    u8 i;
    if(cmd)
      OLED_DC_Set();
    else
      OLED_DC_Clr();
    OLED_CS_Clr();
    for(i=0;i<8;i++)
    {
        OLED_SCLK_Clr();
        if(dat&0x80)
            OLED_SDIN_Set();
        else
            OLED_SDIN_Clr();
        OLED_SCLK_Set();
        dat<<=1;
    }
    OLED_CS_Set();
    OLED_DC_Set();
}

//设置坐标的位置(x 范围: 0~127    ,    y 的范围:0~63)
//注意: 8 行为一页,共 64 行即 8 页
void OLED_Set_Pos(unsigned char x, unsigned char y)
{
    OLED_WR_Byte(0xb0+y,OLED_CMD);
    OLED_WR_Byte(((x&0xf0)>>4)|0x10,OLED_CMD);
    OLED_WR_Byte((x&0x0f)|0x01,OLED_CMD);
}
```

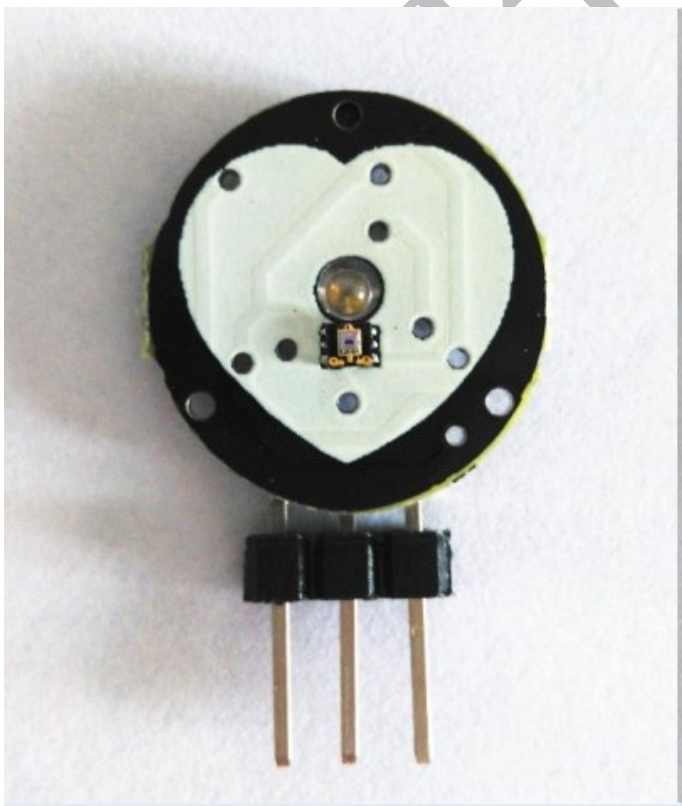
表 4-4 DDRAM 地 址 表

	CS1=1, CS2=0							CS1=0, CS2=1							
Y=	0	1	2	3	..	062	63	0	1	2	3	..	62	63	行号
X=0	DB0 ↓ DB7						DB0 ↓ DB7	DB0 ↓ DB7	横向0~127					DB0 ↓ DB7	0 ↓ 7
↓	DB0 ↓ DB7						DB0 ↓ DB7	DB0 ↓ DB7						DB0 ↓ DB7	8 ↓ 55
X=7	DB0 ↓ DB7	纵向0~7					DB0 ↓ DB7	DB0 ↓ DB7						DB0 ↓ DB7	56 ↓ 63

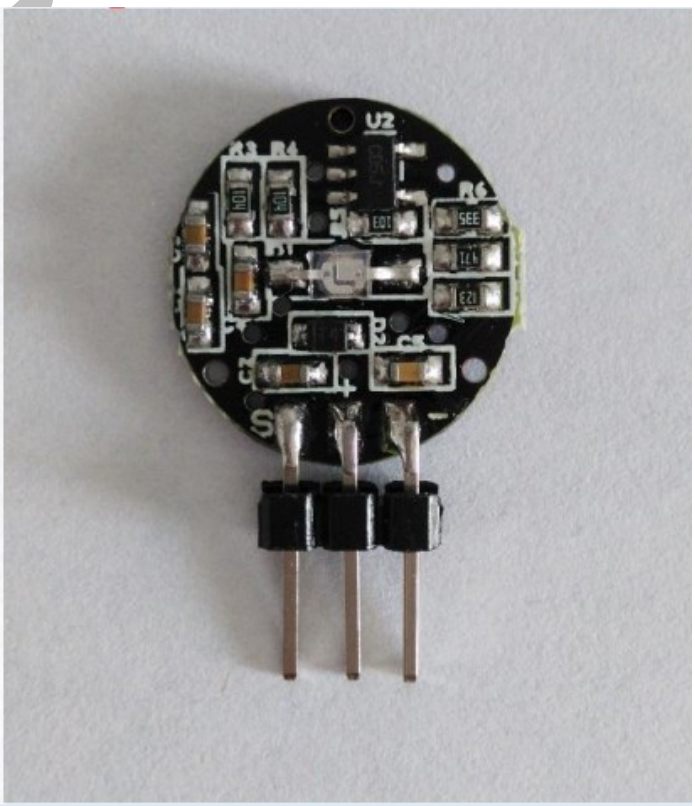
2.3 心率模块使用说明

2.3.2 产品简介

PulseSensor 是一款用于脉搏心率测量的光电反射式模拟传感器。将其佩戴于手指、耳垂等处，通过导线连接可将采集到的模拟信号传输给 Arduino 等单片机用来转换为数字信号，再通过 arduino 等单片机简单计算后就可以得到心率数值，此外还可将脉搏波形通过串口上传到电脑显示波形。 PulseSensor 是一款开源硬件， 目前国外官网上已有其对应的 arduino 程序和上位机 Processing 程序， 其适用于心率方面的科学研究和教学演示，也非常适合用于二次开发。 传感器实物如下图：

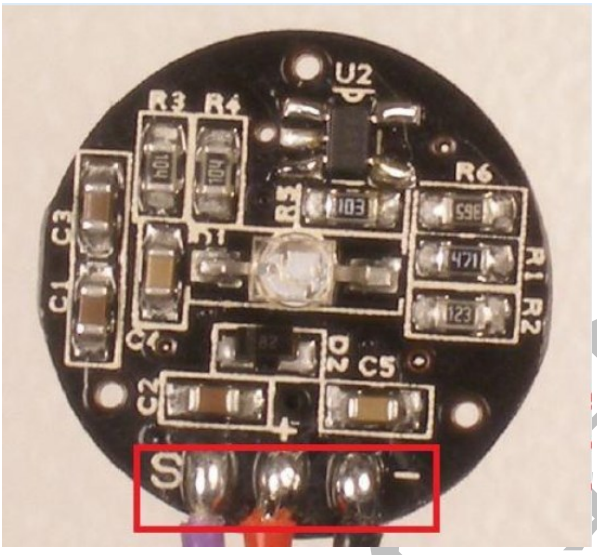


正面（手指接触面）



反面（非手指接触面）

特别提醒：传感器背面是电子元件，请不要用手指直接接触， 以免静电或汗液造成背面器件损坏。 可以在背面粘贴黑色粘扣， 正面粘贴透明膜来保护传感器。



传感器的接口一共 3 个， 如上图红框内所示。 请大家千万不要根据线的颜色来自行推测， 而要根据电路板的背面标识来分辨。

红框中的 3 根线，标有 S 的为模拟信号输出线（最左边）； 标有+的为电源输入线（中间）；标有-的为地线（最右边） 。

总结一下：

S → 脉搏信号输出（要接单片机 AD 接口）

+ → 5v(或 3.3v)电源输入

- → GND 地

2.3.3 传感器参数介绍

电路板直径： 16mm

电路板厚度： 1.2mm

LED 峰值波长： 515nm

供电电压： 3.3~5v

检测信号类型：光反射信号(PPG)

输出信号类型：模拟信号

信号放大倍数： 330 倍

输出信号大小： 0~VCC

电流大小： ~4ma(5v 下)

2.3.4 原理介绍

传统的脉搏测量方法主要有三种：一是从心电信号中提取；二是从测量血压时压力传感器测到的波动来计算脉率；三是光电容积法。前两种方法提取信号都会限制病人的活动，如果长时间使用会增加病人生理和心理上的

不舒适感。而光电容积法脉搏测量作为监护测量中最普遍的方法之一，其具有方法简单、佩戴方便、可靠性高等特点。

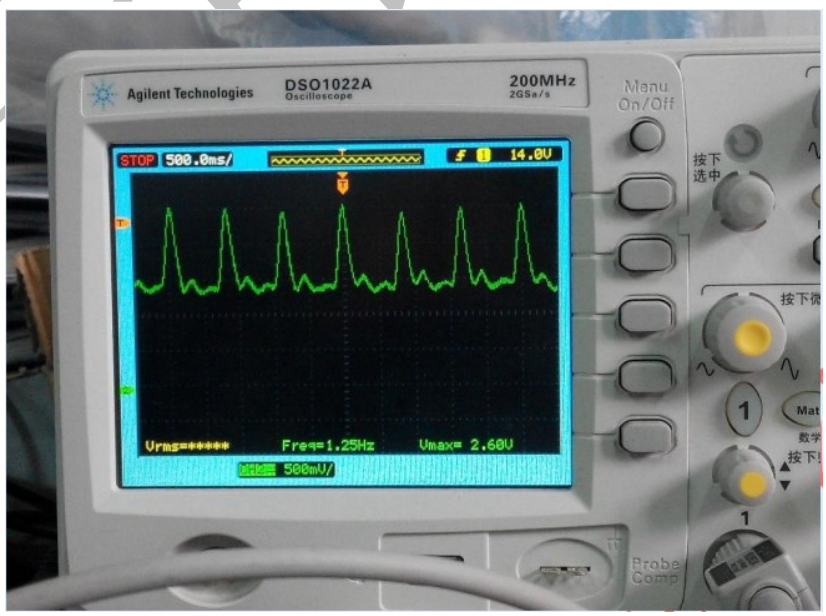
光电容积法的基本原理是利用人体组织在血管搏动时造成透光率不同来进行脉搏测量的。其使用的传感器由光源和光电变换器两部分组成，通过绑带或夹子固定在病人的手指或耳垂上。光源一般采用对动脉血中氧和血红蛋白有选择性的一定波长（500nm~700nm）的发光二极管。当光束透过人体外周血管，由于动脉搏动充血容积变化导致这束光的透光率发生改变，此时由光电变换器接收经人体组织反射的光线，转变为电信号并将其放大和输出。由于脉搏是随心脏的搏动而周期性变化的信号，动脉血管容积也周期性变化，因此光电变换器的电信号变化周期就是脉搏率。

根据相关文献和实验结果， 560nm 波长左右的波可以反映皮肤浅部微动脉信息，适合用来提取脉搏信号。本传感器采用了峰值波长为 515nm 的绿光 LED，型号为 AM2520，而光接收器采用了 APDS-9008， 这是一款环境光感受器，感受峰值波长为 565nm，两者的峰值波长相近，灵敏度较高。此外，由于脉搏信号的频带一般在 0.05~200Hz 之间， 信号幅度均很小，一般在毫伏级水平，容易受到各种信号干扰。在传感器后面使用了低通滤波器和由运放 MCP6001 构成的放大器，将信号放大了 330 倍，同时采用分压电阻设置直流偏置电压为电源电压的 1/2，使放大后的信号可以很好地被单片机的 AD 采集到。

整个心率传感器的结构如下图：

2.3.5 传感器使用说明

心率传感器可以直接使用示波器测量。 按照传感器引脚顺序（参看前面第二章内容） 接好电源引脚和地引脚。 将示波器探头接到输出引脚（S） ， 同时务必将示波器的时间扫描档放置到 200ms 左右，电压扫描档放置到 2V 左右,示波器探头设置为直流耦合（DC） 。 如果档位不正确是无法看到正确的波形的！ 如下图是正确波形的显示，此时说明传感器是好的。



由于传感器使用的是固定倍数的放大器， 而人体生理信号是微弱信号，细微的差异会导致放大后的信号产生巨大的差别。 所以上图的波形只是理想情况下的波形，每个人的实际效果会略有区别。当人体的脉搏信号较

强时，放大后的信号有可能达到或超过电源电压，此时只要改变手指与传感器的接触，就会使波形恢复正常，而且这一现象并不会影响心率的计算。

- 注意事项：

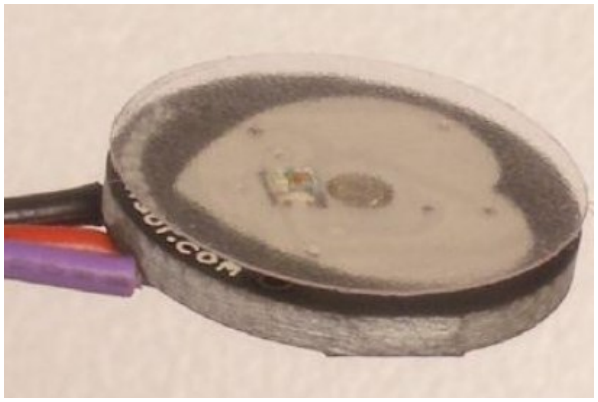
心率值准确、脉搏波形完好都与传感器和手指的接触有很大关系， 为了获得良好的体验效果

1、保持指尖与传感器接触良好

2、不要太用力按，否则血液循环不畅会影响测量结果

3、保持镇静，测量时身体不要过多移动,这个会影响测量结果

4、不要用冰凉的手指进行测试,因为血液循环不好会让测量结果不准确



2.3.6 心率计算算法

```
int BPM;                                    // 用于保存脉冲速率
int Signal;                                 // 持有的原始数据
int IBI = 600;
unsigned char Pulse = false;
unsigned char QS = false;
int rate[10];
unsigned long sampleCounter = 0;
unsigned long lastBeatTime = 0;
int P =512;
int T = 512;
int thresh = 512;
int amp = 100;
unsigned char firstBeat = true;
unsigned char secondBeat = false;
/*
      定时器 2 中断服务函数 用于周期性采集心率值
*/
void TIM2_IRQHandler(void)
{
      uint16_t runningTotal=0;
      uint8_t i;
      uint16_t Num;

      if(TIM2->SR&1<<0)
```

```
{
        //读取到的值右移 2 位，12 位-->10 位
    Signal = Get_AdcCHx_DATA(1)>>2;
    sampleCounter += 2;
    Num = sampleCounter - lastBeatTime;

    //发现脉冲波的波峰和波谷
    //    find the peak and trough of the pulse wave
    if(Signal < thresh && Num > (IBI/5)*3)
    {
        if (Signal < T)
        {
            T = Signal;
        }
    }

    if(Signal > thresh && Signal > P)
    {
        P = Signal;
    }

    //开始寻找心跳
    //当脉冲来临的时候，signal 的值会上升
    if (Num > 250)
    {
        if ( (Signal > thresh) && (Pulse == false) && (Num > (IBI/5)*3) )
        {
            Pulse = true;
            //LED0(0);
            IBI = sampleCounter - lastBeatTime;
            lastBeatTime = sampleCounter;

            if(secondBeat)
            {
                secondBeat = false;
                for(i=0; i<=9; i++)
                {
                    rate[i] = IBI;
                }
            }

            if(firstBeat)
            {
                firstBeat = false;
                secondBeat = true;
                return;
            }
```

```
                for(i=0; i<=8; i++)
                {
                    rate[i] = rate[i+1];
                    runningTotal += rate[i];
                }

                rate[9] = IBI;
                runningTotal += rate[9];
                runningTotal /= 10;
                BPM = 60000/runningTotal;
                QS = true;
            }
        }

        //脉冲开始下降
        if (Signal < thresh && Pulse == true)
        {
            Pulse = false;
            amp = P - T;
            thresh = amp/2 + T;
            P = thresh;
            T = thresh;
        }

        //没有检测到脉冲，设置默认值
        if (Num > 2500)
        {
            thresh = 512;
            P = 512;
            T = 512;
            lastBeatTime = sampleCounter;
            firstBeat = true;
            secondBeat = false;
        }
    }
    TIM2->SR&=0x0; //清中断标志
}
```

2.4 MPU6050 模块使用说明

2.4.1 产品简介

MPU6050 计步模块是一款高性能三轴加速度+三轴陀螺仪的六轴传感器。 该模块采用 InvenSense 公司的 MPU6050 芯片作为核心， 该芯片内部整合了 3 轴陀螺仪和 3 轴加速度传感器，并可利用自带的数字运动处理器（DMP: Digital Motion Processor）硬件加速引擎，通过主 IIC 接口，向应用端输出姿态解算后的数据。有

了 DMP，我们可以使用 InvenSense 公司提供的运动处理资料库，非常方便的实现姿态解算，降低了运动处理运算对操作系统的负荷，同时大大降低了开发难度。

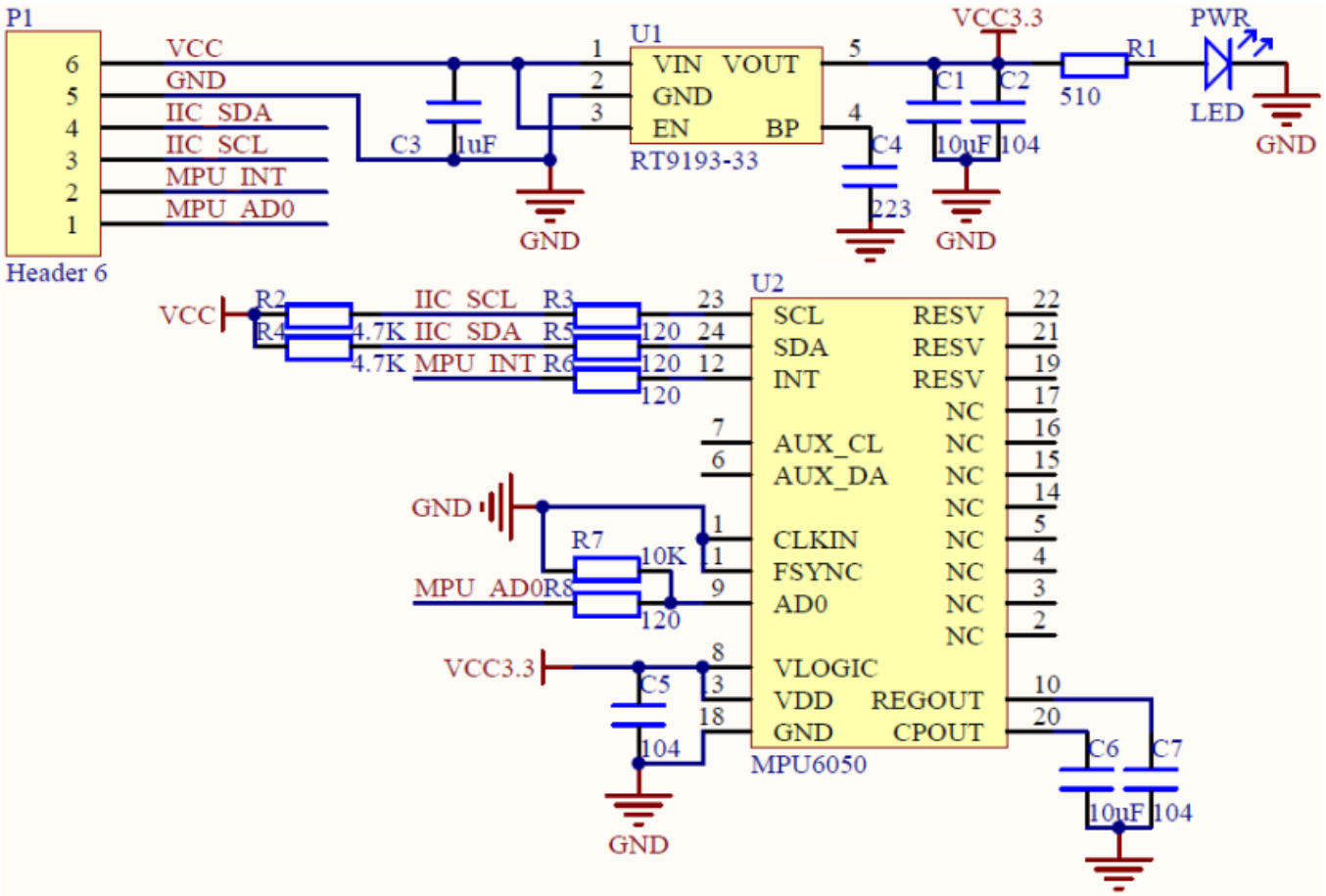
MPU6050 模块具有：体积小、自带 DMP、自带温度传感器、支持 IIC 从机地址设置和中断、兼容 3.3V/5V 系统、使用方便等特点

MPU6050 模块块各项参数如表所示。

项目	说明
接口特性	3.3V/5V
通信接口	IIC 接口
通信速率	400Khz（Max）
测量维度	加速度：3 维　　陀螺仪：3 维
加速度测量范围	±2/±4/±8/±16g
陀螺仪测量范围	±250/±500/±1000/±2000°/秒
ADC 位数	16 位
分辨率	加速度：16384LSB/g(Max)　陀螺仪：131LSB/(°/s)(Max)
输出速率	加速度：1Khz (Max)　　陀螺仪：8Khz (Max)
姿态解算输出速率	200Hz (Max)
温度传感器测量范围	-40℃~85℃
温度传感器精度	±1℃
工作温度	-40℃~85℃
模块尺寸	16mm*18mm

MPU6050 六轴传感器模块外观如图所示：

MPU6050 六轴传感器模块原理图介绍:



模块自带了 3.3V 超低压差稳压芯片，给 MPU6050 供电，因此外部供电可以选择： 3.3V / 5V 都可以。 模块通过 P1 排针与外部连接，引出了 VCC、 GND、IIC_SDA、 IIC_SCL、 MPU_INT 和 MPU_AD0 等信号，其中，IIC_SDA 和 IIC_SCL 带了 4.7K 上拉电阻，外部可以不用再加上拉电阻了，另外 MPU_AD0 自带了 10K 下拉电阻，当 AD0 悬空时，默认 IIC 地址为（0X68）。

MPU6050 六轴传感器模块通过一个 1*6 的排针（P1）同外部电路连接，各引脚的详细描述如表所示：

序号	名称	说明
1	VCC	3.3V/5V 电源输入
2	GND	地线
3	IIC_SDA	IIC 通信数据线
4	IIC_SCL	IIC 通信时钟线
5	MPU_INT	中断输出引脚
6	MPU_AD0	IIC 从机地址设置引脚： ID：0X68(悬空/接 GND) ID：0X69(接 VCC)

模块仅通过一个 IIC 接口与外部通信，并可以通过 MPU_AD0 设置模块的 IIC 地址，当 MPU_AD0 悬空接 GND 的时候，模块的 IIC 从机地址为： 0X68；当 MPU_AD0 接 VCC 的时候，模块的 IIC 从机地址为：0X69。

2.4.2 MPU6050 简介

MPU6050 是 InvenSense 公司推出的全球首款整合性 6 轴运动处理组件，相较于多组件方案，免除了组合陀螺仪与加速器时之轴间差的问题，减少了安装空间。

MPU6050 内部整合了 3 轴陀螺仪和 3 轴加速度传感器，并且含有一个第二 IIC 接口（本模块未引出），可用于连接外部磁力传感器，并利用自带的数字运动处理器（DMP: DigitalMotion Processor）硬件加速引擎，通过主 IIC 接口，向应用端输出完整的 9 轴融合演算数据。有了 DMP，我们可以使用 InvenSense 公司提供的运动处理资料库，非常方便的实现姿态解算，降低了运动处理运算对操作系统的负荷，同时大大降低了开发难度。

MPU6050 的特点包括：

① 以数字形式输出 6 轴或 9 轴（需外接磁传感器）的旋转矩阵、四元数(quaternion)、欧拉角格式(Euler Angle forma)的融合演算数据（需 DMP 支持）

② 具有 131 LSBs/° /sec 敏感度与全格感测范围为±250、±500、±1000 与±2000°/sec 的 3 轴角速度感测器(陀螺仪)

③ 集成可程序控制，范围为±2g、±4g、±8g 和±16g 的 3 轴加速度传感器

④ 移除加速器与陀螺仪轴间敏感度，降低设定给予的影响与感测器的飘移

⑤ 自带数字运动处理(DMP: Digital Motion Processing)引擎可减少 MCU 复杂的融合演算数据、感测器同步化姿势感应等的负荷

⑥ 内建运作时间偏差与磁力感测器校正演算技术， 免除了客户须另外进行校正的需求

⑦ 自带一个数字温度传感器

⑧ 带数字输入同步引脚(Sync pin)支持视频电子影相稳定技术与 GPS

⑨ 可程序控制的中断(interrupt)，支持姿势识别、摇摄、画面放大缩小、滚动、快速下降中断、 high-G 中断、零动作感应、触击感应、摇动感应功能

⑩ VDD 供电电压为 2.5V±5%、 3.0V±5%、 3.3V±5%； VLOGIC 可低至 1.8V± 5%

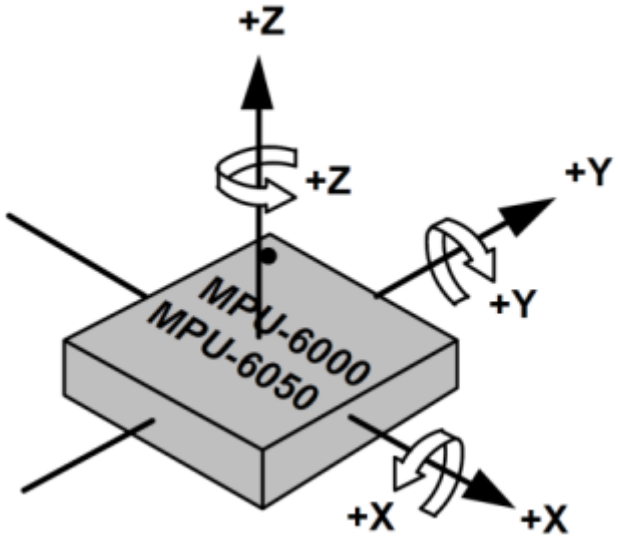
⑪ 陀螺仪工作电流： 5mA，陀螺仪待机电流： 5uA；加速器工作电流： 500uA，加速器省电模式电流：40uA@10Hz

⑫ 自带 1024 字节 FIFO，有助于降低系统功耗

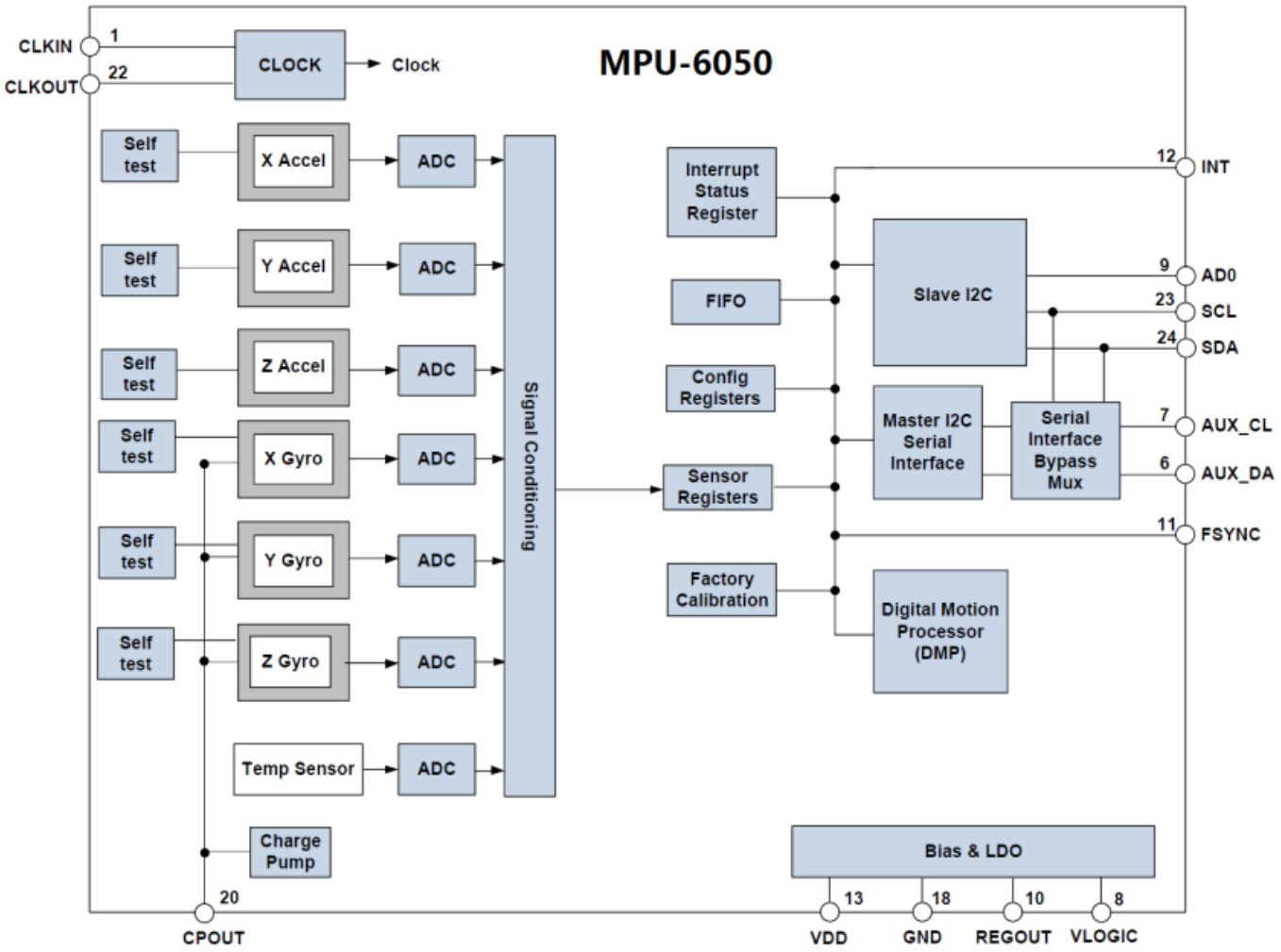
⑬ 高达 400Khz 的 IIC 通信接口

⑭ 超小封装尺寸： 4x4x0.9mm（QFN）

MPU6050 传感器的检测轴如图 2.2.1 所示：



MPU6050 的内部框图如图所示：



其中，SCL 和 SDA 是连接 MCU 的 IIC 接口，MCU 通过这个 IIC 接口来控制 MPU6050，另外还有一个 IIC 接口： AUX_CL 和 AUX_DA，这个接口可用来连接外部从设备，比如磁传感器，这样就可以组成一个九轴传感器。VLOGIC 是 IO 口电压，该引脚最低可以到 1.8V，我们一般直接接 VDD 即可。 AD0 是从 IIC 接口（接 MCU）的地址控制引脚，该引脚控制 IIC 地址的最低位。如果接 GND，则 MPU6050 的 IIC 地址是：0X68，如果接 VDD，则是 0X69，注意：这里的地址是不包含数据传输的最低位的（最低位用来表示读写）！！