



南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

电子信息工程课程设计实验报告

心电信号采集与分析

作者: 马子轩 学号: 9161040G0826

同组人: 赖梓扬 学号: 9161040G0821

学院: 电子工程与光电技术学院

专业(方向): 电子信息工程

指导者: 王志华

评阅者: _____

2019 年 9 月

目 次

| | |
|--|----|
| 1 设计目的..... | 1 |
| 2 设计要求..... | 1 |
| 2.1 基本要求..... | 1 |
| 2.2 提高要求..... | 1 |
| 3 设计内容及步骤..... | 2 |
| 3.1 心电信号的产生..... | 2 |
| 3.2 将 DSP 实验箱的 OUT3 端口连接示波器..... | 2 |
| 3.3 接通 DSP 实验箱电源，根据液晶显示屏显示的提示信息进行操作..... | 2 |
| 3.4 心电信号的分析参照标准..... | 4 |
| 3.5 MATLAB 程序编写以及数据分析原理..... | 5 |
| 4 硬件设计原理..... | 6 |
| 4.1 前置放大模块..... | 6 |
| 4.1.1 OP07 芯片作用..... | 6 |
| 4.1.2 AD620 芯片作用..... | 7 |
| 4.2 右腿驱动模块..... | 7 |
| 4.3 滤波模块..... | 8 |
| 4.3.1 高通滤波器..... | 8 |
| 4.3.2 低通滤波器..... | 9 |
| 4.3.3 滤波电路..... | 9 |
| 4.4 电平抬升模块..... | 10 |
| 5 软件设计原理（独立自主完成）..... | 12 |
| 5.1 原代码分析..... | 12 |
| 5.1.1 LED 显示..... | 12 |
| 5.1.2 AD 采样..... | 12 |
| 5.2 信号处理的实现..... | 12 |
| 5.2.1 算法简析..... | 12 |
| 5.2.2 代码实现及注释..... | 13 |

| | |
|--------------------------------|----|
| 5.3 将算得的数据显示到 LCD 显示屏..... | 14 |
| 5.3.1 静态显示..... | 14 |
| 5.3.2 动态显示..... | 14 |
| 5.3.3 部分代码..... | 14 |
| 5.4 心率的计算..... | 15 |
| 5.4.1 算法简析..... | 15 |
| 5.4.2 部分代码..... | 15 |
| 6 实验结果..... | 16 |
| 6.1 Matlab 对心电信号分析结果 | 16 |
| 6.1.1 隔直量化后的心电图与频域图..... | 16 |
| 6.1.2 加窗后的心电图与频谱..... | 16 |
| 6.1.3 结果显示..... | 17 |
| 6.2 利用 DSP 分析输入信号并显示到显示屏 | 17 |
| 6.2.1 信号发生器产生心电信号..... | 17 |
| 6.2.2 示波器上显示的 ADDA 波形 | 18 |
| 6.2.3 LCD 输出信号处理结果..... | 18 |
| 6.3 误差分析..... | 18 |
| 7 实验感想..... | 19 |
| 8 附录: matlab 程序源码..... | 20 |

1 设计目的

通过对心电信号的采集分析处理，可以有效地监测人的心脏和血压的健康状况。本次课程设计利用 DSP 实验装置首先采集心电传感器模块输出的心电信号，通过 USB 接口传输到计算机中，利用 MATLAB 编程对心电信号进行处理，显示出心电图波形，并对心电信号进行频谱分析。

2 设计要求

2.1 基本要求

1. 熟悉信号采集工作原理；
2. 利用任意信号发生器产生心电信号；
3. 运用 DSP 数据采集实验装置采集心电信号，并将采集的数据送到计算机中存储；
4. 利用 Matlab 编程对心电信号进行处理，显示出心电图，并得出相关心电参数；
5. 撰写课程设计报告。

2.2 提高要求

1. 修改 DSP 程序，采用实验装置采集正弦信号并进行处理，在 LCD 上显示信号频率、幅度等信息；
2. 修改 DSP 程序，在实验装置上分析、处理心电信号，得出相关心电参数，并在 LCD 显示处理结果。

3 设计内容及步骤

3.1 心电信号的产生

接通心电信号采集器以及传感器，连接到示波器，按照图 3.1 所示把三个电极放在身体各部分，调节示波器可以看到心电信号，也可以利用信号发生器（图 3.2）产生心电信号（以 Sa 函数代替），利用示波器测量确认后，通过连接电缆将心电信号采集器或者信号发生器的输出连接到的 C2000DSP 实验箱的 INPUT1 端口。

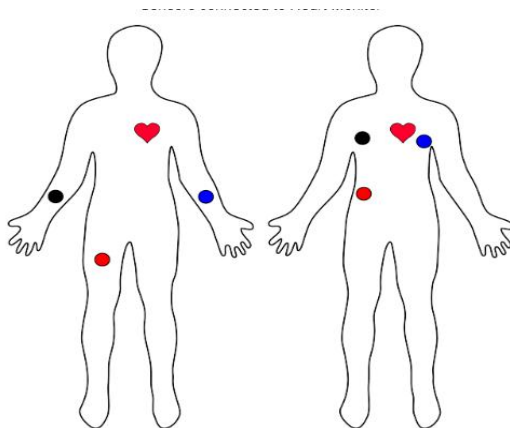


图 3-1 心电信号采集器连接位置

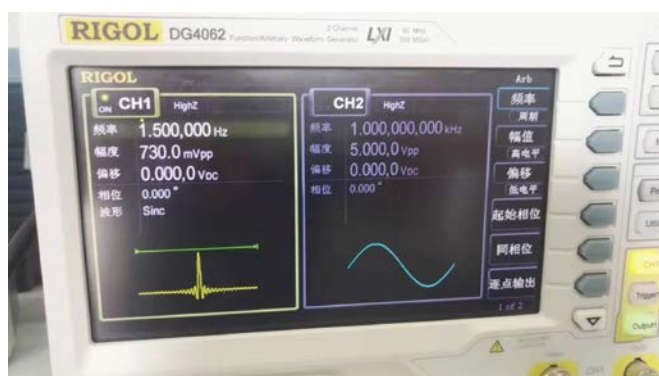


图 3-2 信号发生器产生心电信号

3.2 将 DSP 实验箱的 OUT3 端口连接示波器

3.3 接通 DSP 实验箱电源，根据液晶显示屏显示的提示信息进行操作

1. 上电后，首先选择 4（AD），按 ENTER 键确认；
2. 通过数字键选择采样频率（符合那奎斯特采样定理,本次试验中采用 300Hz），按 ENTER 键确认；
3. 选择“1”保存，通过主机上的采集软件，可将采集的数据通过 USB 线上传到主机。选则“2”不保存，可通过 DSP 试验箱的 OUT3 接口，通过示波器观察波形，若系统正常，

应该能够看到跟信号发生器输出一致的波形，以此来验证电路系统的正确性；

4. 若在 3) 选择“1”保存后，主机提示安装 USB 驱动，正确安装驱动后，打开主机上的数据采集软件，会出现如图所示界面：

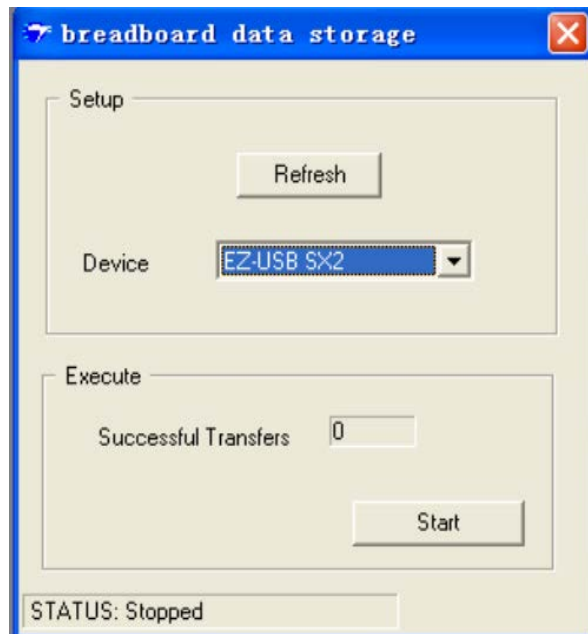


图 3-3 数据采集软件界面示意图

5. 点击“start”,开始数据传输，若系统工作正常，Successful Transfers 后会显示“5”，表明收到 5 个数据包，若显示信息不是 5，则将 DSP 试验箱断电，重新开始。

6. 若 5) 正常，则主机会产生一个数据文件 USB.DAT，这就是 ADC 采集的数据，共 1024 个采样点，每个采样点为 12 位有效数字，表示为 2 个字节，高 8 位在前（其中高 4 位为 0），低 8 位在后。

7. 在 PC 端接收实验箱传输的采样数据，利用编好的 MATLAB 程序进行数据处理。

8. 观察 PC 端 MATLAB 显示结果，逐步调试改进程序。

9. 修改程序并对数据进行分析最后给出健康分析。

10. 重新采集两组数据进行对比试验。两组信号分别为：

1) 2Hz,600Mv

2) 1Hz,400mV

3.4 心电信号的分析参照标准

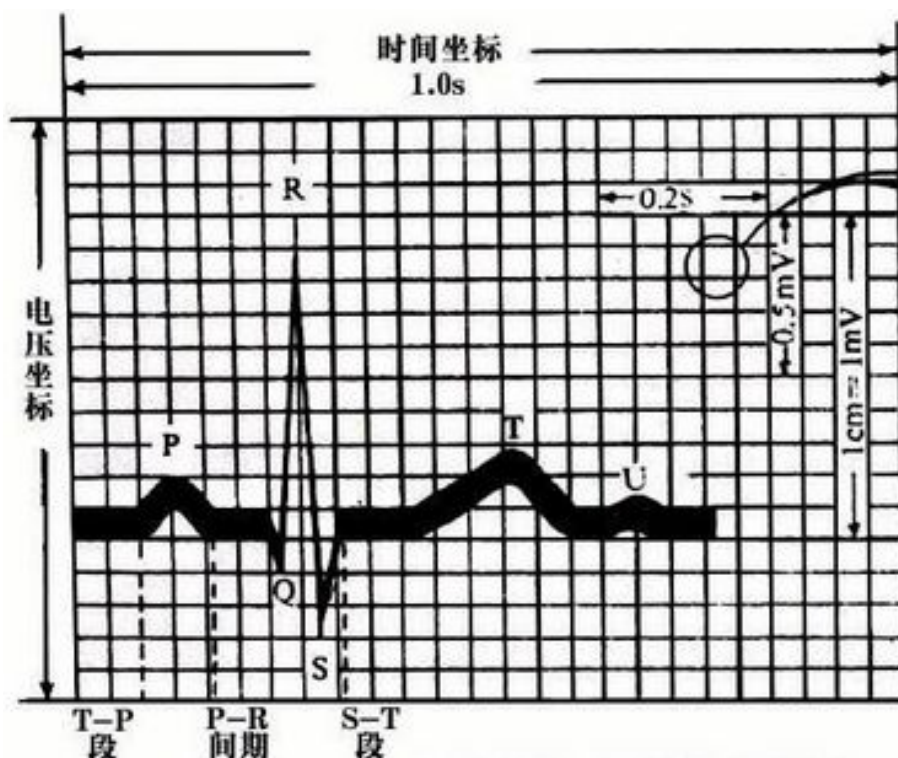


图 3-4 正常心电图波形示意图

【心电图的各个指标正常要求范围如下】：

1. PR 间期：（120～200）ms。
2. QRS 宽度：（60～100）ms。
3. QT 间期：（340～430）ms（跟心率很相关，此为对应 60～100bpm 的 QT 间期正常最高值）。
4. QTC 间期：< 440ms（QTc 为心率校正的 QT 间期= QT/\sqrt{RR} , 临界 QTc 值为 440～460ms, >460ms 判断为 QT 延长, <350ms 为缩短）。
5. ST 段：（-0.05～0.3）mv，（超过正常范围下移常见于心肌缺血或劳损，上移多见于急性心肌梗塞、急性心包炎等）。
6. P 波幅度 $\leq 0.25\text{mv}$ ，宽度 $\leq 0.11\text{s}$ 。
7. Q 波幅度 \leq 同导联 1/4R 波振幅 宽度 $\leq 0.04\text{s}$ 。
8. QRS 波群较复杂，一般可认为 0.5mv～2.0mv。
9. T 波幅度 \geq 同导联 1/10R 波幅度，胸前导联 T 波幅度高达 1.2mv～1.5mv，T 波低平或者倒置常见于心肌缺血、低血钾等。
10. U 波：振幅很小，在胸前导联特别是 V3 较清楚，可高达 0.2～0.3 毫伏。

11. 正常窦性心率 60~100bpm (对应的 RR 间期为 1s~0.6s)。

3.5 MATLAB 程序编写以及数据分析原理

1. 实验数据在通过 DSP 实验箱进行数据采集时会产生电噪声, 影响数据分析, 故在数据频谱分析前需要进行隔直、归一化与滤波, 实验中滤除直流分量可以使用参考语句 $data = data - mean(data)$, 滤波采用加凯瑟窗 (具体程序见附录二)。

2. 心率即心脏在一分钟之内跳动的次数, 其产生次数表现在波形上即为一段时间内 R 峰出现的次数, 因而计算心率即要求使用 MATLAB 编程在处理过的数据中找到所有最大波峰 (使用 *findpeaks* 函数), 给出所有最大波峰的横纵坐标即时间和电压, 对相邻波峰出现的时间间隔取平均数 t , 使用公式:

$$\text{心率} = 60/t$$

3. Q/S 点的查找, 首先规定一定阈值, 在这个阈内, Q/S 点即为最大或者最小点, 此时使用函数 *max()* 即可得到相应的横纵坐标;

4. Q 波、T 波和 S 波的脉冲宽度, 这三个波可以视为正弦波的部分, 通过在一定范围内, 当数值开始出现连续性的变化时, 可以视为这个开始产生变化的点即为以上各波的起始点, 它们与波峰或者波谷点的横坐标差的二倍即为波的时间宽度, 进行转化, 得到脉冲宽度。

5. 血压的高低与心率的大小成正比

6. 健康分析综合各项指标进行。

4 硬件设计原理

来自导联电极的心电信号混有主要包括人体肌电呼吸等生物噪声、电极接触噪声、工频 50Hz 信号及其谐波等干扰；以及其它电子设备机器噪声及外界高频电磁干扰等噪声，其中工频 50Hz 干扰信号较强，主要是共模噪声。

按照设计要求，皮肤接触电极到分别通过 1.5m 长的屏蔽导联线与前置放大器相连接。由于信号线对屏蔽线的输入电容不完全对称，造成共模电压的不等量衰减，使得包括导联在内的放大器共模抑制比降低，从而使系统抑制干扰的能力下降。其中工频干扰引起的共模信号可能远大于心电信号，其影响尤为严重。而由于工频干扰频谱与正常心电信号混杂，又不宜采用工频陷波器滤除。

为有效地消除输入电路不对称而引起的电压分配效应所产生的共模干扰，采用屏蔽驱动和右腿驱动电路。从输入导联取出的共模电压送入屏蔽层（屏蔽层不接地）；同时送到右腿放大器反向放大，经一个限流电阻接到右腿电极，即等效为以人体为相加点的电压并联负反馈电路。抑制了共模干扰进入后续电路。为更好的抑制工频干扰，可以在右腿驱动电路加入低通滤波电路。满足将心电放大器含屏蔽导联线的共模抑制比提高到 80dB 的指标要求。

4.1 前置放大模块

从人体体表拾取的心电信号一般只有几个毫伏，为了提高其分辨率以便于后端显示和处理，首先需要对信号进行放大。在心电信号采集过程中，前置放大电路对心电信号的影响最大，为提高心电信号的性能，前置放大电路的放大倍数不能选择得太大（一般小于 20），否则会由于有较大的干扰信号（指电极的极化电压），致使放大器产生阻塞现象。

前置放大是整个信号放大最关键的环节，关系到整个模拟采集部分的工作性能。设计电路时必须把干扰因素减小到最小。它关系到前置放大电路的性能，因而它的选型非常重要。

前置放大器的性能并不是整个实际电路的性能，还必须辅以合理的电路结构来充分发挥前置放大器的作用。前置放大级最重要的电路参数为共模抑制比参数，很大程度上取决于电路的对称性，在生物电信号采集电路中，前置放大一般采用仪表放大器，主要有共模抑制比高，输入阻抗大、精度高、漂移小等原因。本设计使用的是 ADI 公司的 AD620。两个 OP07 起到阻抗变换的作用，其实就是电压跟随器和阻抗变换。

4.1.1 OP07 芯片作用

OP07 芯片是一种低噪声，非斩波稳零的双极性运算放大器集成电路。由于 OP07 具有非常低的输入失调电压（对于 OP07A 最大为 $25\ \mu\text{V}$ ），所以 OP07 在很多应用场合不需要额外

的调零措施。OP07 同时具有输入偏置电流低（OP07A 为 $\pm 2\text{nA}$ ）和开环增益高（对于 OP07A 为 300V/mV ）的特点，这种低失调、高开环增益的特性使得 OP07 特别适用于高增益的测量设备和放大传感器的微弱信号等方面。

特点有超低偏移： $150\text{ }\mu\text{V}$ 最大。低输入偏置电流： 1.8nA 。低失调电压漂移： $0.5\text{ }\mu\text{V}/^\circ\text{C}$ 。超稳定时间： $2\text{ }\mu\text{V/month}$ 最大。高电源电压范围： $\pm 3\text{V}$ 至 $\pm 22\text{V}$ 。

4.1.2 AD620 芯片作用

AD620 在本设计中是作为前级放大芯片使用的。它是一款低成本、高精度的仪表放大器，仅需要一个外部电阻来设置增益，增益范围为 1 至 10000。此外，AD620 采用 8 引脚 SOIC 和 DIP 封装，尺寸小于分立电路设计，并且功耗更低，因而非常适合电池供电及便携式应用。AD620 的基本特点为精确度高、使用简单、低噪声，此仪表放大器有高输入阻抗，高共模抑制比(CMR): 100dB ，低输入抵补电压(Input offset Voltage): 50uV ，低输入偏移电流(Input bias current): 1.0nA ，低消耗功率： 1.3 mA ，以及过电压保护等特性应用十分广泛。

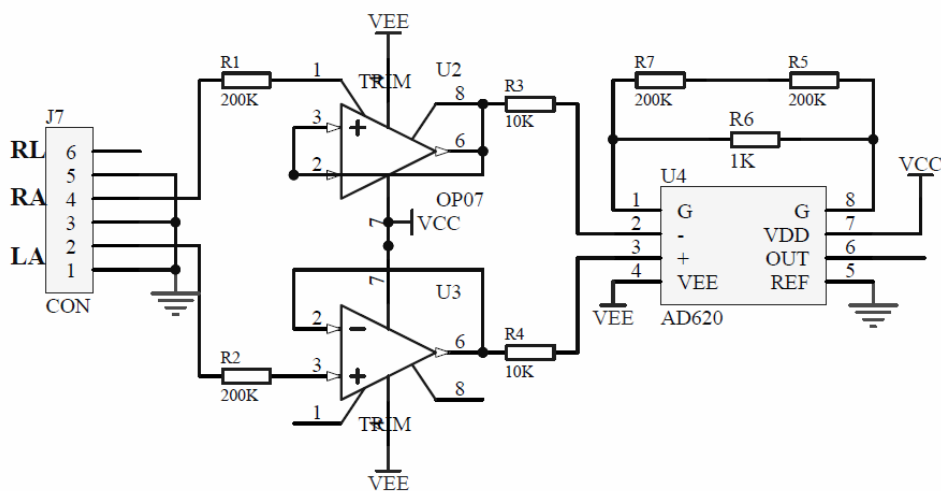


图 4-1 前置放大电路图

4.2 右腿驱动模块

人体接地是造成触电事故的一个重要原因，因此取消人体接地是最根本的安全用电措施。人体接地本来就是在没有高质量的放大器情况下采取少共模信号的应急措施。测量心电图时，如果病人右脚不接地，由于杂散分布电容的影响，病人身上将会产生很高的共模电压。因此，最理想的方法是设计出一种既能减少共模干扰又能取消人体接地的电路。右腿驱动的工作原理是将由人体体表获得的共模电压通过负反馈放大的方式输回人体，从而达到抵消共模干扰的作用，从根本上抑制共模电压。如图 4-2 所示。

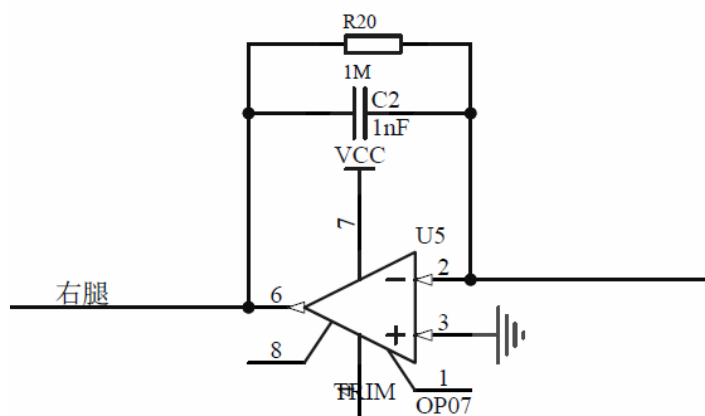


图 4-2 右腿驱动电路图

4.3 滤波模块

4.3.1 高通滤波器

由于心电信号微弱，需要多级放大，而多级直接耦合的直流放大器虽能满足要求，但多级直接耦合的直流放大器容易引起基线漂移。此外，由于极化电压存在的缘故，动态心电图机的直流放大器更不能采用多级直接耦合。本装置中，在两级放大器之间采用 RC 耦合电路，即时间常数电路，在隔离直流信号的同时达到高通滤波的效果。我们取时间常数约为 3.2s，这样可确定电阻、电容值，在两级之间组成高通滤波器。可得转折频率为：

$$f_c = \frac{1}{2\pi CR} = 0.05 \text{ Hz}$$

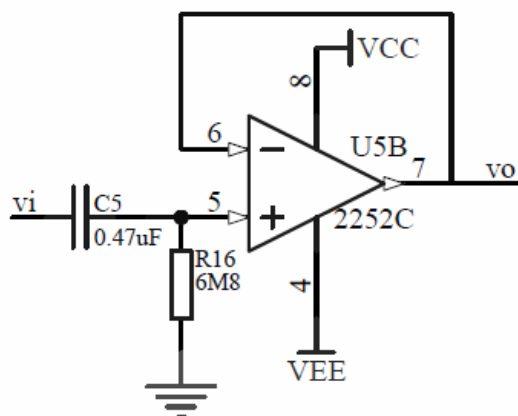


图 4-3 高通滤波电路

4.3.2 低通滤波器

由于电磁干扰越来越严重，所以心电信号在采集过程中不仅有 50Hz 的工频干扰和低频、直流分量的干扰，还有高于 100Hz 高频谐波的严重干扰，有必要进行低通滤波电路的设计。如图 4-4 所示

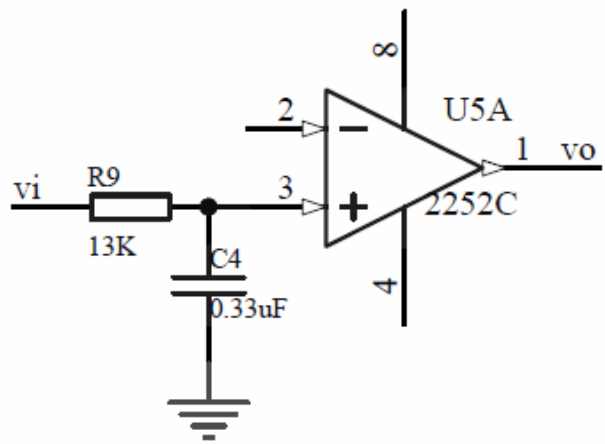


图 4-4 低通滤波电路

4.3.3 滤波电路

有分析可知滤波电路由高通滤波、低通滤波和 50Hz 陷波电路构成，滤波电路如图 4-5 所示。

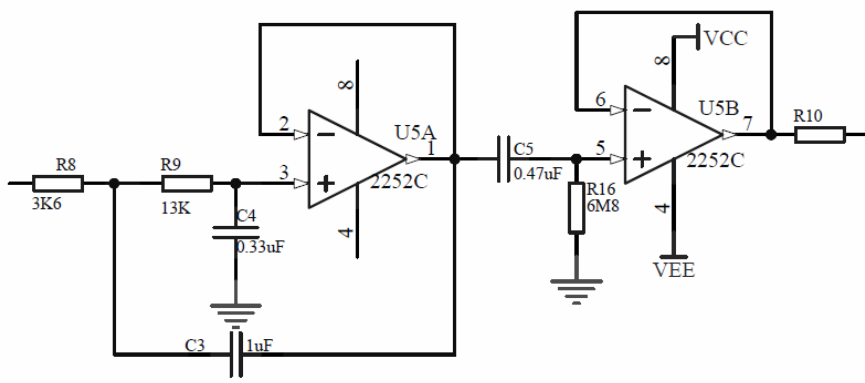


图 4-5 滤波电路

4.4 电平抬升模块

放大滤波后的心电信号将被送入 A/D 转换电路进行模数转换，而本系统选择的 AD 转换器是单 5V 供电的 MAX187，所以滤波之后需加一级电平抬升电路，将心信号抬升到 0V~+5V 的范围。电路设计很简单，如图 4-6 所示。

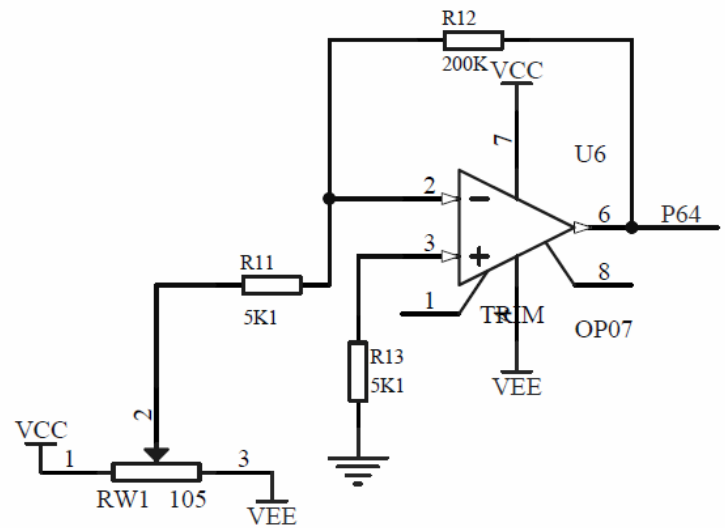


图 4-6 电平抬升电路

采用 A S I C 芯片的心电信号采集器的电路：

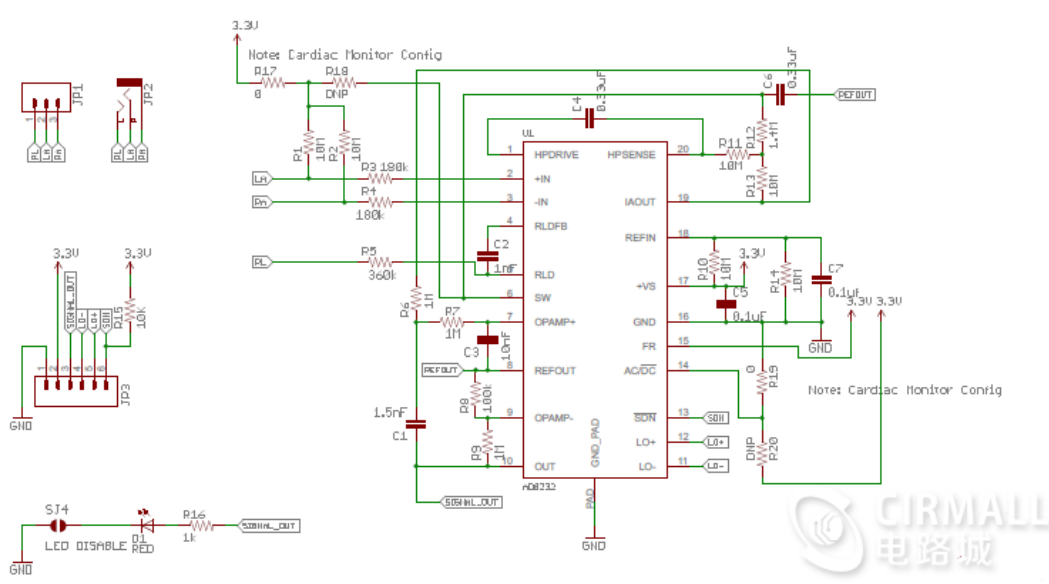


图 4-7 心电信号采集器电路

工作原理

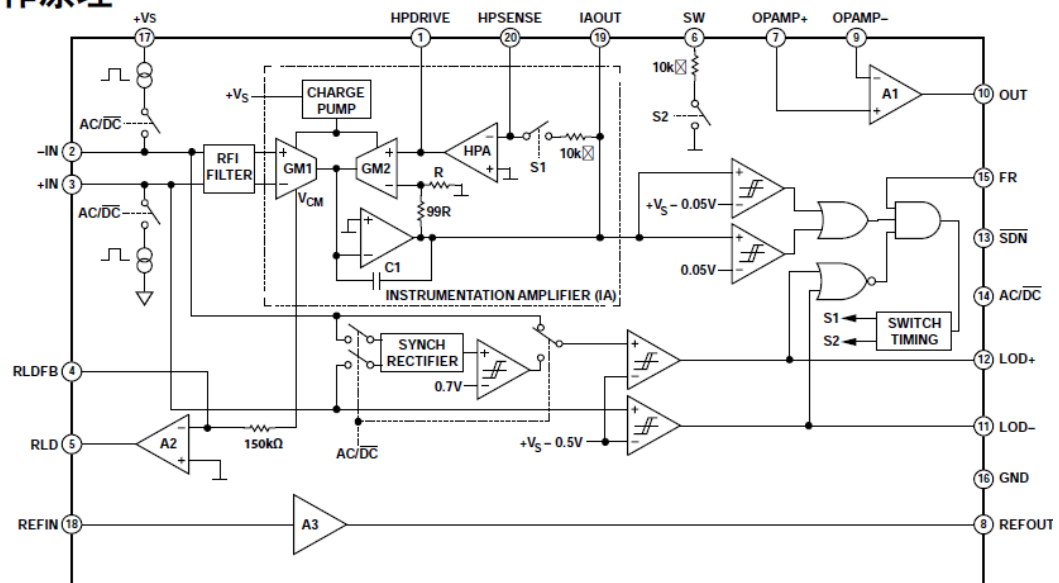


图 4-8 电路工作原理

5 软件设计原理（独立自主完成）

5.1 原代码分析

在实验提供的众多代码中，可以发现对本实验有用的程序文件主要有 3 个，分别是：LED.h，LED.c 和 AD.c。

5.1.1 LED 显示

LED.h 和 LED.c 两个文件用来控制 LED 的显示，提供了一些字模和接口程序。

其中字模由字模提取软件提取后，将二进制代码存放在“zimu”这个数组中。

在 LED.c 文件中定义了“void wr_hex(unsigned int code,unsigned int o_y,unsigned int o_x,unsigned short fanxian)”函数，只需给出字符显示的横纵坐标和在“zimu”中的位置，即可在 LED 显示屏上显示出字符。

5.1.2 AD 采样

AD.c 文件可以说为整个 DSP 程序的主程序，里面包含了菜单选择、AD 采样等我们需要的主要功能。

在 main 函数的 while 循环中，位于 645 行有一个“while(can!=6)”循环，该循环为菜单进入“AD -> 设定频率 -> 不保存”后，正在执行的循环函数。在这个函数中添加一些显示语句，就可以将想要显示的信号处理的结果全部显示在屏幕上。

位于 808 行的“interrupt void adc_isr(void)”函数为 AD 采样的中断服务函数。AD 每采样一次，就进入一次中断服务函数。在 adc_isr 函数中有一个 switch 语句，用于判断具体执行的函数。经分析，选择“保存”后，执行“case 33: AD_read();”；选择“不保存”后，执行“case 34: AD_DA();break;”

在“AD_DA()”这个函数中，有“xn=AdcRegs.ADCRESULT0;”这样一个语句。我们将 xn 输出在 LED 上后，发现“AdcRegs.ADCRESULT0”其实为 AD 采样的实时值，但是由于这个采样值的范围已经超过 int 范围（-32768~32767）。为了减少 bug 的出现，在本实验中，将 AdcRegs.ADCRESULT0 右移四位，并重新赋值给一个新的全局变量，从而实现对 AD 采样值的实时分析。

5.2 信号处理的实现

5.2.1 算法简析

因在上一节已经提到得到 AD 采样实时值的方式，于是对于信号数据的分析就相对简单。

1. 振幅

依据最大值最小值原理得到振幅的相对大小，然后分析得到的 AD 数值和实际的信号幅度，总结出线性的表达公式，进行换算后可得到最终结果。其中，AD 采样与实际信号幅度的线性表达式为：

$$\text{Value} * 10 = (\text{ADmax} - \text{ADmin}) * 15 - 600;$$

其中，ADmax 和 ADmin 分别为 AD 采样的最大值和最小值，Value 为转换得到的幅值，单位为 mV，*10 是为了方便利用整形（int）得到小数点后一位。

2. 频率

频率的计算采用阈值法，阈值的选取为信号的平均值。因为我们认为，无论是正弦信号还是心电信号，其平均值位置的 AD 采样结果是斜率最大的，不易产生相同结果。当前一采样结果小于阈值，本次采样结果大于阈值时，即认为函数经过了一个周期。通过周期计数器得到的结果和采样频率的关系，得出最终的频率和周期。

3. 实时刷新数据

实时刷新的问题主要在于最大值和最小值的刷新。因为 AD 采样的数据的不断出现，如果一直取数据的最大值和最小值，就会导致最大值只能增大，最小值只能减小的问题出现。当输入信号的幅度增大时会刷新，但是当输入信号的幅度减小时，程序无法做到正确显示输入信号的幅度。

我的解决方法是每 10 个信号周期后，将 AD 采样的最大值(AD_max)和最小值(AD_min)都降低 100，这样既可以保证数据可以实时刷新，又可以保证显示的数据不会有很大的跳变。

5.2.2 代码实现及注释

```
void AD_DA()
{
    GpioDataRegs.GPBDAT.bit.GPIOB3 = 1; //原程序中自带的代码
    xn=AdcRegs.ADCRESULT0; //原程序中自带的代码
    zn=AdcRegs.ADCRESULT0>>4; //将AdcRegs.ADCRESULT0右移4位，赋值给zn，得到AD采样实时值的缩小值
    GpioDataRegs.GPBDAT.bit.GPIOB3 = 0; //原程序中自带的代码
    * DAOUT=xn; //原程序中自带的代码
    if (zn_max<zn) zn_max=zn; //取zn中的最大值
    if (zn_min>zn) zn_min=zn; //取zn中的最小值
    Avg = (zn_max + zn_min) / 2; //通过计算得到zn平均值Avg，即为判断周期的阈值
    if (zn>Avg && zn_last < Avg) //如果前一个zn小于Avg且本次zn大于等于Avg，则认为这时一个周期的结点
    {
        flag_zn++; //zn标志位自加，这个标志位的目的是每10个周期刷新zn_max和zn_min
        if(flag_zn%10==0) //判断flag_zn是否是10的倍数
        {
            flag_zn=1; //将flag_zn置1
            zn_max-=100; //最大值减100，实现数据刷新
            zn_min+=100; //最小值加100，实现数据刷新
        }
        Period = Frequency_time*1000/adfreq_last; //计算周期，adfreq_last为采样频率的1/10，得到周期的单位是0.1ms
        //这是为了方便显示的时候可以直接用int将小数点后一位显示出来
        Frequency = adfreq_last*100/Frequency_time; //计算频率，得到频率的单位是0.1Hz
        Frequency_time = 0; //Frequency_time置0
    }
    else{Frequency_time++;} //Frequency_time自加，Frequency_time为周期计数器，每次AD采样加1
    //每当整周期是置0
    zn_last = zn; //将zn赋给zn_last，作为上一次的采样值
}
```

图 5-1 AD 部分代码

计算幅度的代码为：Value = (zn_max - zn_min)*15-600; //得到的单位为 0.1mV

5.3 将算得的数据显示到 LCD 显示屏

5.3.1 静态显示

所谓静态显示，就是首先在显示屏上显示一些不需要改变的东西，比如学号姓名、数据的名称、单位等。

在 LCD.c 文件中的“extern void menu_4(void)”函数中定义了执行 AD 采样时的菜单界面，我修改了部分代码，让菜单界面上显示了我和组员的姓名学号，以及一些数据的名称和单位。代码略。

5.3.2 动态显示

由于得到的数据（如周期、频率等）需要动态的显示在 LCD 屏幕上，所以需要不断的刷新屏幕上的内容，这就不能在“extern void menu_4(void)”函数中实现，需要在 AD.c 文件中的“while(can!=6)”循环中实现。

由于 LCD 的显示需要一位一位显示，不能将变量直接作为参数输入到“GUILCD_writeCharStr();”函数中，需要将变量中的某一位输入到这个函数中。于是就需要对变量取各个位的数字，这个是通过“/10”和“%10”的组合利用来实现的，具体实现方法请见代码。

5.3.3 部分代码

```
//AD 采样结果
GUILCD_writeCharStr(0x03,0x0A,zn%10, FALSE); //zn 个位
GUILCD_writeCharStr(0x03,0x08,zn/10%10, FALSE); //zn 十位
GUILCD_writeCharStr(0x03,0x07,zn/100%10, FALSE); //zn 百位
GUILCD_writeCharStr(0x03,0x06,zn/1000%10, FALSE); //zn 千位
GUILCD_writeCharStr(0x03,0x05,zn/10000%10, FALSE); //zn 万位
//显示频率
GUILCD_writeCharStr(0x05, 0x0B, 73, FALSE); //Hz
GUILCD_writeCharStr(0x05, 0x0A, Frequence%10, FALSE); //小数点后一位
GUILCD_writeCharStr(0x05, 0x09, 71, FALSE); //小数点
GUILCD_writeCharStr(0x05, 0x08, Frequence/10%10, FALSE);
GUILCD_writeCharStr(0x05, 0x07, Frequence/100%10, FALSE);
GUILCD_writeCharStr(0x05, 0x06, Frequence/1000%10, FALSE);
//显示幅度
GUILCD_writeCharStr(0x04, 0x0B, 72, FALSE); //mV
GUILCD_writeCharStr(0x04, 0x0A, Value%10, FALSE);
GUILCD_writeCharStr(0x04, 0x09, 71, FALSE); //小数点
GUILCD_writeCharStr(0x04, 0x08, Value/10%10, FALSE);
GUILCD_writeCharStr(0x04, 0x07, Value/100%10, FALSE);
GUILCD_writeCharStr(0x04, 0x06, Value/1000%10, FALSE);
```

```
//显示周期
GUILCD_writeCharStr(0x06, 0x0B, 74, FALSE);//ms
GUILCD_writeCharStr(0x06, 0x0A, Period%10, FALSE);
GUILCD_writeCharStr(0x06, 0x09, 71, FALSE);//小数点
GUILCD_writeCharStr(0x06, 0x08, Period/10%10, FALSE);
GUILCD_writeCharStr(0x06, 0x07, Period/100%10, FALSE);
GUILCD_writeCharStr(0x06, 0x06, Period/1000%10, FALSE);
GUILCD_writeCharStr(0x06, 0x05, Period/10000%10, FALSE);
```

5.4 心率的计算

5.4.1 算法简析

心率即为一分钟之内心跳的次数，转换到程序中，就是一分钟之内采到信号的周期数，即为频率的 60 倍，故心率=60*频率。

判断心率是否正确，根据 Matlab 例程中给出的判断标准，利用一个 if 语句判断心率是否在正常的范围内，并将比较结果输出到显示屏上。标准为，心率在 60 以下为心率过低，在 60~100 为正常，在 100 以上为心率过高。

5.4.2 部分代码

```
//显示心率;
GUILCD_writeCharStr(0x07, 0x06, Frequence*60/10%10, FALSE);
GUILCD_writeCharStr(0x07, 0x05, Frequence*60/100%10, FALSE);
GUILCD_writeCharStr(0x07, 0x04, Frequence*60/1000%10, FALSE);
//判断心率是否正常
if(Frequence*6<100 && Frequence*6>60)
{
    GUILCD_writeCharStr(0x07, 0x0C, 19, FALSE);//正
    GUILCD_writeCharStr(0x07, 0x0D, 78, FALSE);//常
}
else if(Frequence*6>100)
{
    GUILCD_writeCharStr(0x07, 0x0C, 79, FALSE);//过
    GUILCD_writeCharStr(0x07, 0x0D, 26, FALSE);//高
}
else
{
    GUILCD_writeCharStr(0x07, 0x0C, 79, FALSE);//过
    GUILCD_writeCharStr(0x07, 0x0D, 24, FALSE);//低
}
```

6 实验结果

6.1 Matlab 对心电信号分析结果

以 1.3Hz，500mV 的心电信号为例，进行分析，结果如下：

6.1.1 隔直量化后的心电图与频域图

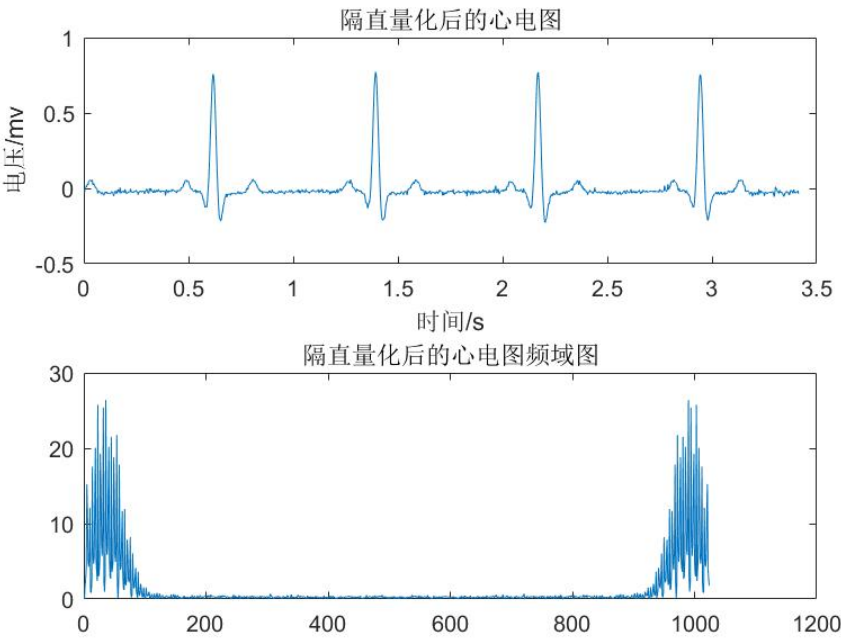


图 6-1 隔直量化后的心电图与频域图

6.1.2 加窗后的心电图与频谱

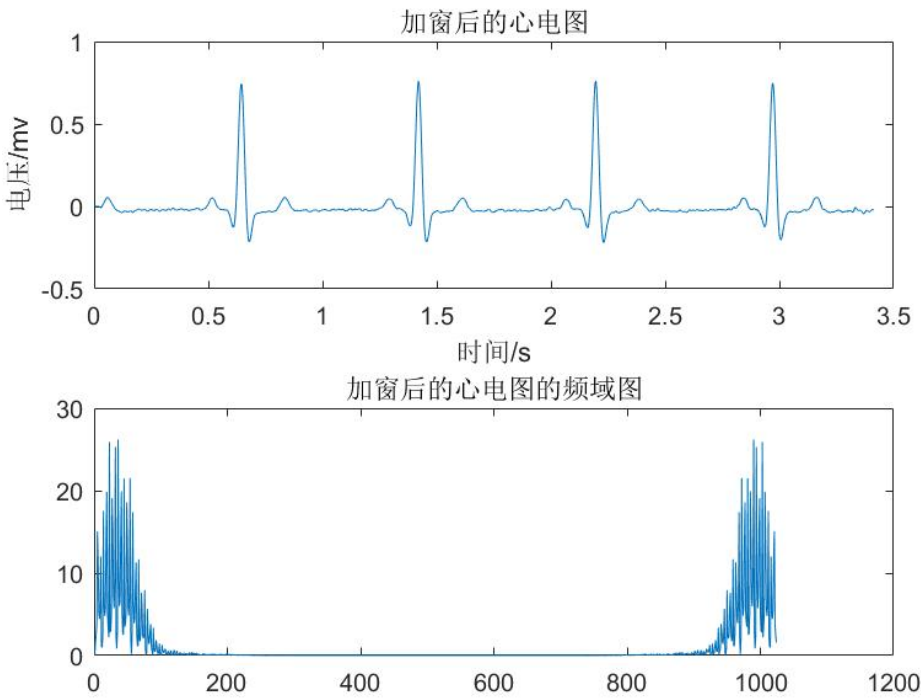


图 6-2 加窗后的心电图与频谱

6.1.3 结果显示

此人的心率(bpm)为:

77

正常人的心率在60-100bpm, 因此此人心率正常, 血压正常

心电图的信噪比为:

52.7391

此人的QRS宽度(s)为:

0.090667

正常人的QRS宽度在60-100ms, 因此此人QRS宽度正常

此人的P波幅度(mv)为:

0.012232

正常的P波幅度<0.25mv, 因此此人的P波幅度正常

此人的ST段范围为-0.21至0.05mv, 而正常的ST段范围为-0.05至0.3mv, 因此此人的ST段幅度范围下移, 不正常, 可能是心肌缺血或劳损

此人的P波宽度(s)为:

-0.066667

正常的P波宽度<0.11秒, 因此此人的P波宽度正常

此人的Q波宽度(s)为:

0.023333

正常的Q波宽度<0.04秒, 因此此人的Q波宽度正常

此人的PR宽度为(s)为:

0.12667

正常的PR宽度为120ms-200ms, 因此此人的PR宽度正常

此人的QT宽度为(s)为:

图 6-3 matlab 结果截图

6.2 利用 DSP 分析输入信号并显示到显示屏

6.2.1 信号发生器产生心电信号

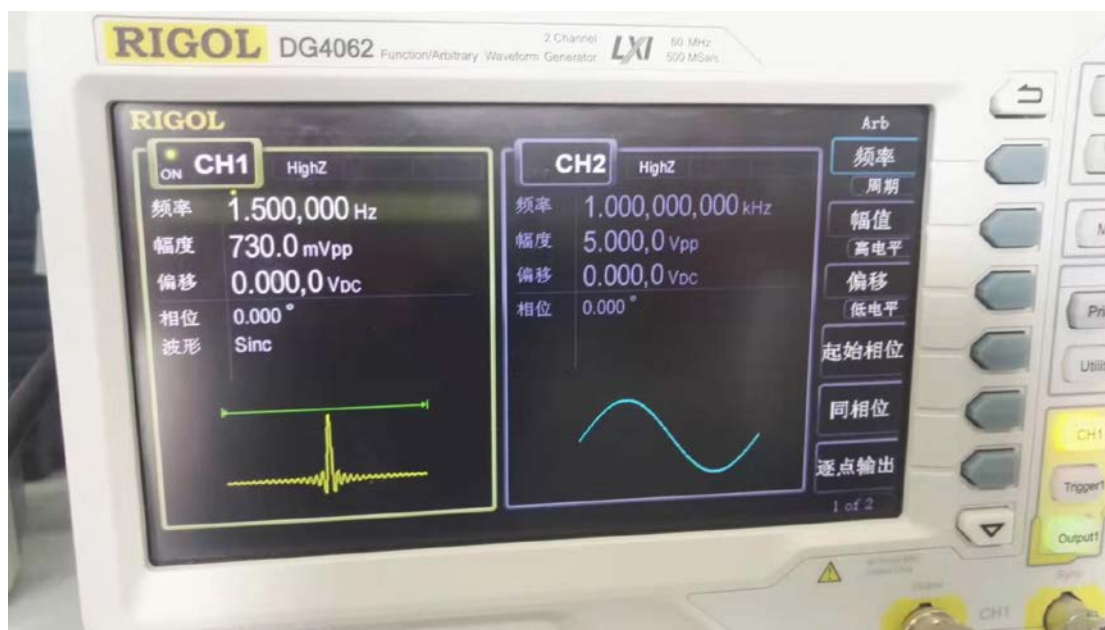


图 6-4 信号发生器产生的心电信号

6.2.2 示波器上显示的 ADDA 波形

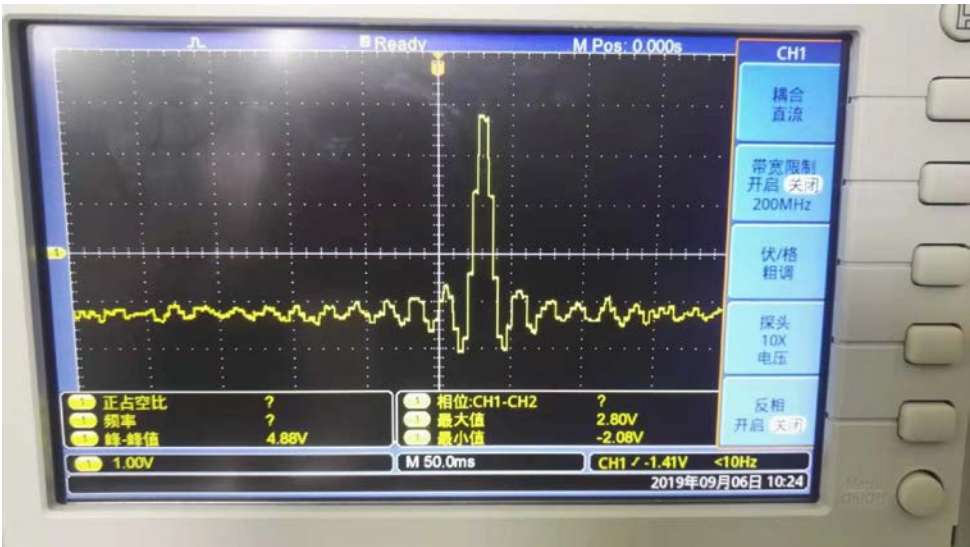


图 6-5 示波器上显示的 ADDA 波形

6.2.3 LCD 输出信号处理结果



图 6-6LCD 输出信号处理结果

6.3 误差分析

在本次实验中，matlab 程序可以很精确的计算出心电信号的各种参数；DSP 可以实时的处理 AD 采样得到的信号，其中频率、周期、心率的计算较为准确，幅度的计算稍有误差，这是由于 AD 采样的数值和幅度值进行转化的公示为线性公式，在推导线性公式的时候存在无法避免的误差。

7 实验感想

在本次实验过程中，体会颇丰。

能很清楚的感受到，本次实验和以往的各种实验的最大差别就是本次实验更侧重于学生对于原理的把握和对独立自主创新的鼓励。王老师在实验中更多的不是要求去实现那些大家都会却意义不大的功能，而是要求大家仔细研究程序，实现那些真正有用的功能。即使我们的算法简单，误差较大，但是由于实现了实时测频、测幅等这些很具有实际意义的功能，也得到了老师的表扬。

实验本应该就是理论和实践的结合，故实验内容也应该具有实践意义。在实验过程中，我们通过几天的努力，仔细推敲代码中每一行的作用，快速找到需要更改的地方，这个是非常重要的。由于我之前学过一些 TI 的单片机（MSP430F6638 系列）编程，故对整个程序架构还是比较有把握的。经过代码分析，算法分析，代码实现，bug 排除等多个步骤，我们最终实现了本报告中提到的全部功能，格外欣慰。

在实验结尾过程中，我们本想实现将波形显示在 LCD 屏幕上的功能，但由于种种原因，最终未能实现，这是本次实验的一点小小的遗憾。

最后，我想感谢在实验中一直给予我鼓励和帮助的王志华老师，感谢在实验过程中和我一起讨论并给与很大帮助的陈彪、许晓明、何超翔、李世源等同学。是你们的无私帮助，让我们的实验过程更有意义，是我们的精诚合作，让我们大家收获颇丰。

8 附录：matlab 程序源码

```
clc;
clear;
close all;
N=1024;
fs=300;
%hz
Ts=1/fs;
n=0:N-1;
fp=n/N*fs;
[FileName,PathName] = uigetfile('USB.dat','Select the USB.dat file');
f = fullfile(PathName,filesep,FileName);
fid = fopen(f,'r');
data = fscanf(fid,'%x');
fclose(fid);
data = data(1:2:end)*256 + data(2:2:end);%进制转换
data=data-mean(data);%隔直
dataal=data;
m=max(data);
n=min(data);
data=data/(m-n);%量化
datsgn1 = data;
subplot(2,1,1);
x = 1:1:1024;
x = x.*(1/300);
plot(x,datsgn1);      %时域波形
title('隔直量化后的心电图');
xlabel('时间/s');ylabel('电压/mv');
F1=fft(datsgn1);
subplot(2,1,2);
plot(abs(F1));
title('隔直量化后的心电图频域图');
%加滤波器
fp=0.05;fc=100;As=80;Ap=1;Fs=300;
wc=2*pi*fc/Fs;
wp=2*pi*fp/Fs;
wd=wc-wp;
beta=0.1102*(As-8.7);
N=ceil((As-7.95)/2.286/wd);
wn=kaiser(N+1,beta);
ws=(wp+wc)/2/pi;
b=fir1(N,ws,wn);
datsgn2=fftfilt(b,data);
F2=fft(datsgn2);
```



```

figure;
subplot(2,1,1);
x = 1:1:1024;
x = x.*(1/300);
plot(x,datsgn2);
title('加窗后的心电图');
xlabel('时间/s');ylabel('电压/mv');
subplot(2,1,2);
plot(abs(F2));
title('加窗后的心电图的频域图');
[ pks,locs]
findpeaks(datsgn2,'MINPEAKDISTANCE',50,'MINPEAKHEIGHT',0.5);
dis=diff(locs);
avrdis=mean(dis);
beat=round(fs/avrdis*60);
display('此人的心率(bpm)为:')
display(num2str(beat))
if beat>=60&&beat<=100
display('正常人的心率在 60-100bpm，因此此人心率正常,血压正常')
else
    if beat<60
display('此人的心率此人心率过低，血压低')
    else
        display('此人的心率此人心率过高，血压高')
    end
end
end
%计算信噪比
pdatsgn1=datsgn1.*datsgn1;
pdatsgn2=datsgn2.*datsgn2;
snr=sum(pdatsgn2)/(sum(pdatsgn1)-sum(pdatsgn2));
display('心电图的信噪比为:')
display(num2str(snr))

%Q 波
Q=[0 0 0 0 0];
Qloc=[0 0 0 0 0];
dat=[datsgn2(90:100)];
dat=abs(dat);
[Q(1),loc1]=max(dat);
loc1=loc1+89;
dat=[datsgn2(307:317)];
dat=abs(dat);
[Q(2),loc2]=max(dat);
loc2=loc2+306;
dat=[datsgn2(521:531)];
dat=abs(dat);

```

=

```

[Q(3),loc3]=max(dat);
loc3=loc3+520;

dat=[datsgn2(737:747)];
dat=abs(dat);
[Q(4),loc4]=max(dat);
loc4=loc4+736;
dat=[datsgn2(950:960)];
dat=abs(dat);
[Q(5),loc5]=max(dat);
loc5=loc5+949;
Qloc(1)=loc1;
Qloc(2)=loc2;
Qloc(3)=loc3;
Qloc(4)=loc4;
Qloc(5)=loc5;
VQ=sum(Q)/5;
%S 波
S=[0 0 0 0 0];
Sloc=[0 0 0 0 0];
dat=[datsgn2(113:123)];
dat=abs(dat);
[S(1),loc1]=max(dat);
loc1=loc1+112;
dat=[datsgn2(328:338)];
dat=abs(dat);
[S(2),loc2]=max(dat);
loc2=loc2+327;
dat=[datsgn2(543:553)];
dat=abs(dat);
[S(3),loc3]=max(dat);
loc3=loc3+542;
dat=[datsgn2(757:767)];
dat=abs(dat);
[S(4),loc4]=max(dat);
loc4=loc4+756;
dat=[datsgn2(972:982)];
dat=abs(dat);
[S(5),loc5]=max(dat);
loc5=loc5+971;
Sloc(1)=loc1;
Sloc(2)=loc2;
Sloc(3)=loc3;
Sloc(4)=loc4;
Sloc(5)=loc5;
vs=sum(S)/5;

```

```

qrs=Sloc-Qloc;
Averageqrs=sum(qrs)/fs/5;
display('此人的 QRS 宽度 (s) 为:')
display(num2str(Averageqrs))
if Averageqrs>=0.06&&Averageqrs<=0.1
    display('正常人的 QRS 宽度在 60-100ms, 因此此人 QRS 宽度正常')
else
display('正常人的 QRS 宽度在 60-100ms, 因此此人 QRS 宽度不正常')
end
% p 点
P=[0 0 0 0 0];
p1=[0 0 0 0 0];
data2=[datsgn2(53:77)];
[P(1),location1]=max(data2);
p1location=53+location1;
data2=[datsgn2(272:296)];
[P(2),location2]=max(data2);
p2location=272+location2;
data2=[datsgn2(491:515)];
[P(3),location3]=max(data2);
p3location=491+location3;
data2=[datsgn2(710:734)];
[P(4),location4]=max(data2);
p4location=710+location4;
data2=[datsgn2(929:953)];
[P(5),location5]=max(data2);
p5location=929+location5;
p1(1)=p1location;
p1(2)=p2location;
p1(3)=p3location;
p1(4)=p4location;
p1(5)=p5location;
% t 点
T=[0 0 0 0 0];
t1=[0 0 0 0 0];
data2=[datsgn2(140:180)];
[T(1),tlocation1]=max(data2);
t1location=53+tlocation1;
data2=[datsgn2(359:399)];
[T(2),tlocation2]=max(data2);
t2location=359+tlocation2;
data2=[datsgn2(578:618)];
[T(3),tlocation3]=max(data2);
t3location=578+tlocation3;
data2=[datsgn2(797:837)];
[T(4),tlocation4]=max(data2);

```

```

t4location=797+tlocation4;
data2=[datsgn2(1016:1024)];
[T(5),tlocation5]=max(data2);
t5location=1016+tlocation5;
t1(1)=t1location;
t1(2)=t2location;
t1(3)=t3location;
t1(4)=t4location;
t1(5)=t5location;
vp=sum(P)/5;
vt=sum(T)/5;
display('此人的 P 波幅度(mv)为:')
display(num2str(vp))
if vp<0.25
    display('正常的 P 波幅度<0.25mv, 因此此人的 P 波幅度正常')
else
    display('正常的 P 波幅度<0.25mv, 因此此人的 P 波幅度不正常')
end
display('此人的 ST 段范围为-0.21 至 0.05mv,而正常的 ST 段范围为-0.05 至 0.3mv,
因此此人的 ST 段幅度范围下移, 不正常, 可能是心肌缺血或劳损')
time=[datsgn2(48:58)];
average=sum(time)/11;
value=[datsgn2(60:70)];
value=value-average;
value=value.*value;
[minvalue,locp]=min(value);
locp=locp+59;
Ptime=(p1location-locp)*2/fs;
display('此人的 P 波宽度(s)为:')
display(num2str(Ptime))
if Ptime<0.11
    display('正常的 P 波宽度<0.11 秒, 因此此人的 P 波宽度正常')
else
    display('正常的 P 波宽度<0.11 秒, 因此此人的 P 波宽度不正常')
end
value=[datsgn2(85:95)];
value=value-average;
value=value.*value;
[minvalue,locq1]=min(value);
locq1=locq1+84;
value=[datsgn2(90:100)];
value=value-average;
value=value.*value;
[minvalue,locq2]=min(value);
locq2=locq2+89;
Qtime=(locq2-locq1)/fs;

```

```

display('此人的 Q 波宽度(s)为:')
display(num2str(Qtime))
if Qtime<0.11
    display('正常的 Q 波宽度<0.04 秒, 因此此人的 Q 波宽度正常')
else
display('正常的 Q 波宽度<0.04 秒, 因此此人的 Q 波宽度不正常')
end
PRtime=(locq1-locp)/fs-Ptime;
display('此人的 PR 宽度为(s)为:')
display(num2str(PRtime))
if PRtime<=0.2&&PRtime>=0.12
    display('正常的 PR 宽度为 120ms-200ms, 因此此人的 PR 宽度正常')
else
display('正常的 PR 宽度为 120ms-200ms, 因此此人的 PR 宽度不正常')
end
value=[datsgn2(169:179)];
value=value-average;
value=value.*value;
[minvalue,loct2]=min(value);
loct2=loct2+168;
QTtime=(loct2-locq1)/fs;
display('此人的 QT 宽度为(s)为:')
display(num2str(QTtime))
if QTtime<=0.43&&PRtime>=0.34
    display('正常的 QT 宽度为 340ms-430ms, 因此此人的 QT 宽度正常')
else
display('正常的 QT 宽度为 340ms-430ms, 因此此人的 QT 宽度不正常')
end
if beat>=60&&beat<=100
display('此人身体健康')
else
display('此人不健康')
end

```