

1585

2022 年嵌入式芯片与系统设计竞赛

基于深度学习的心电信号采集和分类识别装置



参赛人：束睿涵、李召熙、郭若鑫

2022 年 7 月 19 日

摘 要

本文基于 STM32 平台和数字滤波算法，设计了实时监测使用者心率、体温和运动数据的无线运动数据采集系统。该系统以屏幕和按键进行交互，采用 24bits 模拟前端芯片 ADS1292R 设计心电信号采集电路，利用 FIR 带通滤波器去除基线漂移和肌电信号干扰，获得稳定可靠的心电信号，通过阈值分割法测量 R 波周期，计算出精确的心率。并对得到的数据进行监督式学习算法进行对心电图信号的标签分类，再根据专家的心电图数据进行对比分析，得到较为精准的疾病信息或者心肌状态。

关键词： 健康监测 心电去噪 心电识别 数字滤波 深度学习 单片机

目 录

摘 要.....	2
第一部分 设计概述.....	5
1.1 设计目的.....	5
1.2 应用领域.....	5
1.3 主要技术特点.....	5
1.4 关键性能指标.....	5
1.5 主要创新点.....	5
第二部分 系统组成及功能说明.....	5
2.1 整体介绍.....	6
2.1.1 心电信号采集和分类识别装置节点硬件结构.....	6
2.1.2 心电信号采集和分类识别装置软件结构.....	7
2.2 系统硬件各模块介绍.....	7
2.2.1 MCU 及核心模块.....	8
2.2.2 RGB 液晶屏幕模块.....	8
2.2.3 心电 ECG 前端模块.....	9
2.2.4 温度传感器模块.....	9
2.2.6 无线网络模块.....	10
2.3 系统软件各模块介绍.....	10
2.3.1 系统分时复用模块设计.....	10
2.3.2 FIR 带通滤波器设计.....	11
2.3.3 心率检测模块设计.....	11
2.3.4 温度信号采集模块设计.....	11
2.3.5 屏幕显示模块设计.....	12
2.3.6 无线网络传输模块设计.....	12
2.4 深度学习与典型病理分析.....	12
2.4.1 KNN Analysis(K 近邻分类)监督学习.....	12
2.4.2 KNN 算法思想.....	13
2.4.3 KNN 算法步骤.....	13
2.4.4 KNN 算法核心知识.....	13
2.4.5 KNN 病例分析.....	14
2.5 图形界面设计.....	19
2.5.1 基本组成.....	19
2.5.2 外设及工作原理.....	19
2.5.3 图形引擎.....	21
2.5.4 软件架构(MVP).....	22
第三部分 完成情况及性能参数.....	22
3.1 系统测试所用仪器.....	23
3.2 系统心率测试.....	23
3.2.1 心率测试方案.....	23

3.2.2 心率测试结果与分析	24
3.3 系统温度测试	25
3.3.1 温度测试方案	25
3.3.2 温度测试结果与分析	25
3.4 系统功耗测试	26
3.5 性能测试总结	26
第四部分 总结	27
4.1 可扩展之处	27
4.2 心得体会	27
第五部分 参考文献	27
第六部分 附录	29

第一部分 设计概述

1.1 设计目的

首先，基于传统心电图分类的诊疗技术基本依靠人力完成。然而，在心脏疾病高发的当今，每天都有大量的患者去做心电图，完全依靠人力去诊断，效率十分低下。其次，心血管疾病是人类生命的最主要的威胁。而心电信号是诊断心血管疾病的主要依据，因此实时检测病人心电活动、设计自动采集存储病人心电信号的便携式系统具有重要意义。为了解决上述问题，基于心电图自动分类诊断技术开始出现。

1.2 应用领域

本产品类属智能健康监测设备，可应用于临床心血管疾病诊断以及自身心率健康检测。还可实现"互联网+医疗服务"，应用于远程医疗领域。

由于健康监测设备可以提供的实时监测的优点，所以它非常的适用于现在的医疗领域的慢性病管理方面。通过对健康数据的分析处理，健康监测设备可以帮助医生及时掌握患者病情变化情况及疾病发展规律，并进行针对性提前治疗。

1.3 主要技术特点

采用 24bits 模拟前端芯片 ADS1292R 设计心电信号采集电路，利用 FIR 带通滤波器去除基线漂移和心肌电信号干扰，获得稳定可靠的心电信号，通过阈值分割法测量 R 波周期，计算出精确的心率。

1.4 关键性能指标

- (1) 所采集温度信号准确度控制在 0.5°C 以内
- (2) 所采集心电信号分析准确度控制在 95% 以内，并可准确对比出 5 种常见心脏疾病
- (3) 采集心电信号经过后端处理后可以得到心电图，并在 TFT 屏上进行动态展示。

1.5 主要创新点

- (1) 与互联网、物联网结合，实时分析并传送个人健康数据，可实现远程医疗。
- (2) 与云计算与大数据结合，分析心电信号进行比对，实现深度学习与典型病例分析

第二部分 系统组成及功能说明

2.1 整体介绍

2.1.1 心电信号采集和分类识别装置节点硬件结构

本设计使用 Altium Designer 21.0.8 软件进行的原理图及 PCB 的绘制。

整个系统硬件采用模块如表 3-1 所示。

表 3-1 系统硬件模块清单

名称	型号	数量
MCU 模块	STM32F407	1
RGB 液晶屏幕模块	RGB043M2-800*480	1
心电 ECG 前端模块	ADS1292R	1
温度传感器模块	DHT11	1
无线网络模块	ESP8266 WIFI	1

系统整体硬件结构如图 3-1 所示。

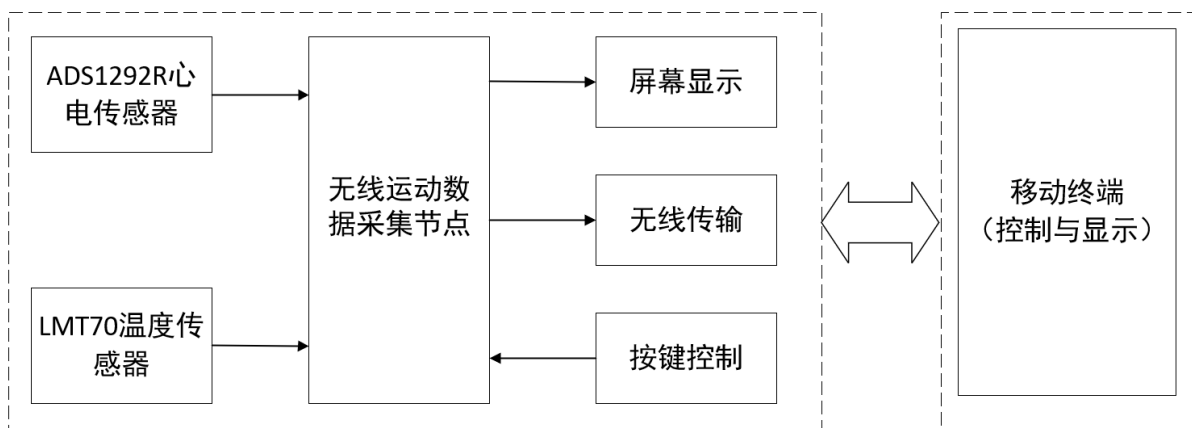


图 3-1 硬件整体结构图

基于 TI 用于生物电势测量的模拟前端芯片 ADS1292R，温度传感器 DHT11 综合设计制作心电信号智能检测仪，此节点采用外部适配器供电，能稳定采集和记录使用者的心电信息、空气温度，并通过 WIFI 模块 ESP8266 将数据传输智能判断心电情况且在移动设备（手机）端实时显示。

2.1.2 心电信号采集和分类识别装置软件结构

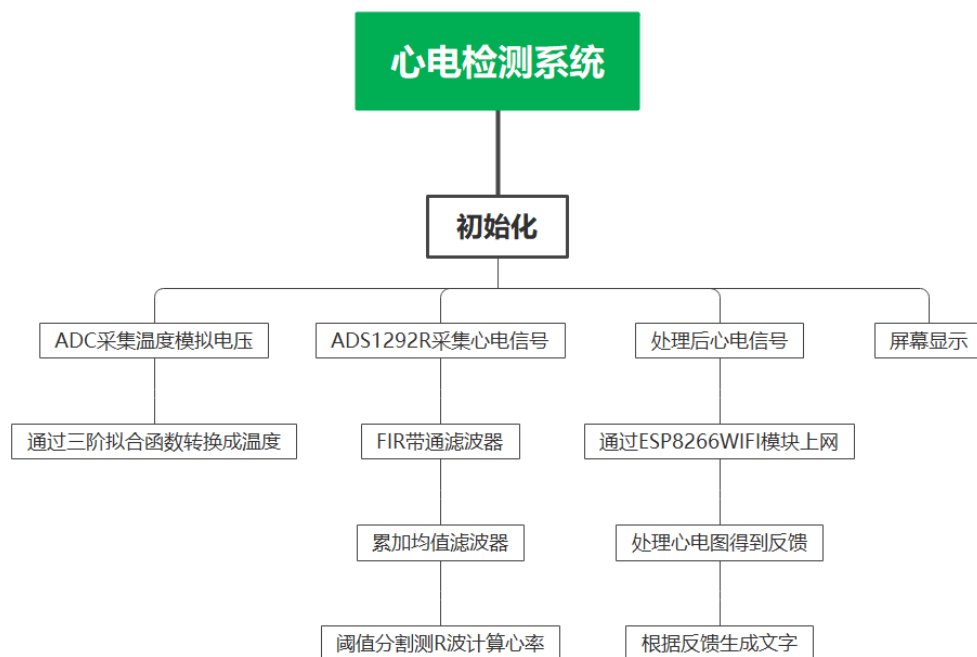


图 3-2 系统整体软件流程图

系统通过分时复用模块，来对各个任务进行控制。ADC 采集的心电信号，经过 FIR 滤波模块去除基线漂移和心肌电信号干扰之后，送入屏幕显示模块更新波形图像，同时数据也计入心率检测模块的移动平均滤波器中，在这里生成自适应的幅度阈值，并判断 R 波尖峰来测量实时心率。采集到心电信号以后，通过 ESP8266 进行云计算识别心电情况，并在移动端查看。温度采集模块通过定时器和 DMA，控制片内 ADC 采集温度传感器的输出电压，然后转换成温度。图形界面主要使用 touch-GFX，与用户进行交互。

2.2 系统硬件各模块介绍

装置的硬件设计，主要是以 STM32F407 作为主控搭建的控制板，将各个模板整合到一起，便于后期的使用和调试。

野火 STM32F407，通过 SPI 协议连接到 24 位 ADCADS1292R，外加部分 ADC 时序控制信号。通过串口连接到 WiFi，单总线 LTH11 输出的模拟电压信号，连接到片内 ADC 通道 4。

2.2.1 MCU 及核心模块

在本设计中，主控芯片选择使用 STM32F407ZGT6，主频 168M，在本设计中，使用主控集成的 SPI 控制器驱动 24 位 ADC 采集心电信号，片内 12bitADC 集合 DMA 控制器在内部定时器的驱动下采集温度的模拟电压，自带 LTDC 液晶控制器，可直接将显存中数据搬运到屏幕，而无需额外的屏幕控制芯片。片内集成 1M Flash，256k SRAM，外接 SDRAM 采用 16 位宽 32M 的 W9825G6JH，可使用 SDRAM 的部分空间作为显存。Flash 采用 8M 的 W25Q64，支持 SPI 通信，存储部分图片以及汉字库。同时 F407 自带 FPU 浮点运算单元，可以利用扩展 DSP 此快速进行 FFT、FIR 等信号处理操作。

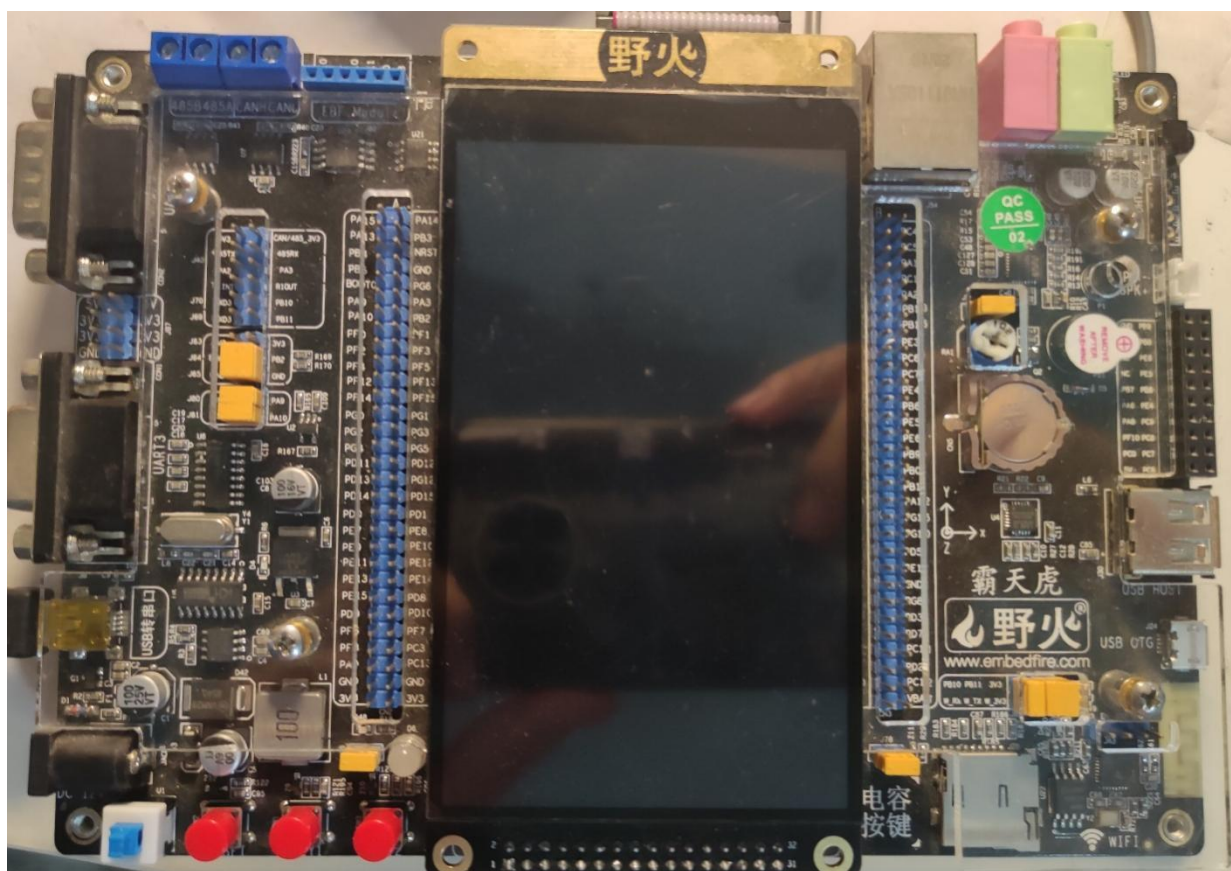


图 3-3-2 MCU 及核心模块

2.2.2 RGB 液晶屏幕模块

本设计中，屏幕采用的是 4.3 寸 RGB 接口的液晶屏，分辨率为 800*480 可通过间距 0.5mm 的 40P FPC 排线进行连接，支持 5 点电容触摸，触摸 IC 为 GT9147，使用 I2C 通信接口。支持 RGB888，向下兼容 RGB565。



图 3-3-3 RGB 液晶屏幕模块

2.2.3 心电 ECG 前端模块

心电 ECG 前端模选择的是 ADS1292R，是一款 24 位模数转换器，其具有低功耗、多通道、同步采样的特点，内部 PGA 可调增益范围为 1~12；功耗低至 335 μ W 每通道；多路输入复用器及其灵活，内部振荡器和基准可极大的简化设计，基本包含了便携式心电信号采集所需的所有功能。通过 SPI 协议驱动，采样率为 125SPS-8kSPS 可调，使用三导联线连接人体的 LA、RA、RL 位置的心电极贴。

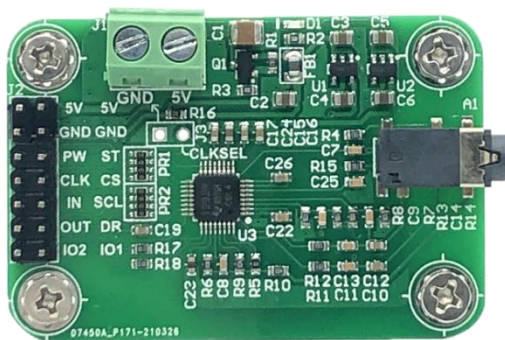


图 3-3-4 心电 ECG 前端模块

2.2.4 温度传感器模块

DHT11 数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有高的可靠性与卓越的长期稳定性。传感器包括一个电容式感湿元件和一个 NTC 测温元件，并与一个高性能 8 位单片机相连接

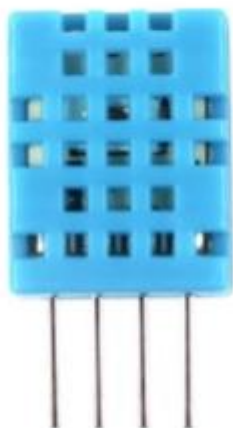


图 3-3-6 温度传感器模块

2.2.6 无线网络模块

无线网络模块，ESP8266WIFI 模块为安信可公司高性能串口 WIFI 模块。此模块具有体积小，功耗低，性能稳定，可靠性高的特点。该模块的内部嵌入了 TCP/IP 协议栈以便于与网络进行连接，利用该模块可以方便的将 WIFI 数据传输到串口，实现两者之间的数据传输转换传输。本设计中使用的型号为 ESP-12E，其串口速率可达 4Mbps，传输距离最远 80 米，工作电流为 15mA，支持 802.11b/g/n 协议



图 3-3-7 无线网络模块

2.3 系统软件各模块介绍

2.3.1 系统分时复用模块设计

系统分时复用模块的本质是根据用定时器产生的时基信号，来作为标准的时间单位，

已达到根据时间执行不同任务的功能。本次设计中使用的定时器为 32bits 的定时器 2，定时器 2 的时钟源来自于系统总线 APB1，当主频设置为 180M 时，APB1 Timer clocks 为 90M，定时器对其进行 9000 分频，计数值设置为 10 时，即可得到 1kHz 的时基信号，系统依据时基信号切换当前执行的任务。

2.3.2 FIR 带通滤波器设计

FIR 滤波器的参数主要是通过 MATLAB 自带的 fdatool 滤波器辅助设计工具生成的，带通滤波器的截止频率分别是 2Hz 与 45Hz，阶数为 129 阶，采用的是 hamming 窗作为加窗函数，当系统采样率为 250Hz 时，生成滤波器系数存入 MCU 中。

系统 FIR 滤波模块主要基于官方 DSP 库中 arm_fir_f32 函数运行，在系统初始化阶段，通过 arm_fir_init_f32 函数，初始化 FIR 滤波器阶数，设置好每次调用处理点数，传入滤波器系数，分配好 FIR 滤波器状态缓存，本系统中每次调用处理采样点数设置为 1，即每次采集到新数据则通过 FIR 滤波器输出以保证系统的实时性。ADC 采集到的信号经过此 FIR 带通滤波器模块之后发送给显示模块和心率测量模块计算心率。

2.3.3 心率检测模块设计

心率检测模块首先将经过 FIR 滤波器处理过的心电信号，通过一个 600 点的移动平均滤波器，求出信号两三个周期的平均值，然后向 R 波尖峰方向偏移 0.3 倍作为判断 R 波的阈值，当新的采样点数据传入之后，首先更新滑动平均的数据以更新阈值，使其具有自适应性。然后将当前传入值与阈值进行比较，记录第一次超过阈值的时间。由于当前未达到 R 波尖峰，所以会继续超过阈值一段时间，正常人体心跳为 60-120，所以两次 R 波最短时间间隔会大于 500ms。系统中设置检测两次超过阈值的最小时间为 400ms，当第二次 R 波超过阈值时，记录与第一次的时间差，从而求出瞬时心率。

2.3.4 温度信号采集模块设计

温度采集主要基于 STM32 片内 ADC 来完成，由定时器 6 产生 20ms 一次定时器更新事件来驱动 DMA，DMA 设置为从外设到内存，数据宽带为 Half Word，即 16bits，设置内存地址自动增加，则可以将每次采集到的数据存到同一个数组中的不同位置。数组存储满后，求出多次采用的平均值，再通过理论分析部分得到的三阶传递函数，将模拟电压转换成温度。

2.3.5 屏幕显示模块设计

屏幕显示控制主要基于 STM32 内部 LTDC 液晶控制器来完成,其内部还拥有 DMA2D 硬件加速器,可快速完成数据传输。在本设计中,选择颜色格式为 RGB565,显存空间从外部 RAM 中开始开辟,显示设置为单层,分辨率为 800*480,采用双缓存区,可以使画面更加流畅。

本设计中,显示模块要显示的数据主要有心率波形,温度、心率、运动步数等信息。其中心率波形采用幅度自适应模式,刷新一屏幕窗口数据的同时,记录本窗口内的最大和最小值,以此之差,与窗口高度的比值,作为下一窗口 y 轴缩放的倍数,即可始终使波形显示在窗口中,并铺满整个窗口。

2.3.6 无线网络传输模块设计

无线网络传输部分主要是基于 WiFi 串口模块 ESP8266 设计,将其内部提前烧录好官方固件,使用 AT 指令对其进行操作。在初始化阶段,先对 ESP8266 进行复位,完成后发送 NAME 和 PASSWD 连接提前设置好的 WiFi 热点;之后开始连接 MQTT 服务器,服务器地址已提前存入 MCU 中,端口号为 1883;之后向云服务器平台发送用于登陆平台的 ClientID、Device Name 和 Password,用户名和密码一一对应,在物联网平台绑定设备的阶段即可通过提供的设备证书,通过一定的加密算法计算出来。成功登陆到物联网平台之后,即可进行消息的发送和订阅。

物联网平台发送和订阅的数据格式主要是 JSON 格式,这是一种轻量级的独立的文本语言格式,其采用键值对的方式,将一个个标识与其对应的值联系起来。数据上传主要由系统分时复用模块控制,每次向服务器传输的数据主要包括使用者的心率信息,温度传感器采集到的温度,姿态检测模块计算出的运动步数,除此之外,心电图的波形数据也会以 JSON 数组的形式打包发送到服务器上。供移动显示终端查看与控制。

2.4 深度学习与典型病理分析

2.4.1 KNN Analysis(K 近邻分类)监督学习

KNN 算法又称 k 近邻分类(k-nearest neighbor classification)算法。它是根据不同特征值之间的距离来进行分类的一种简单的机器学习方法,它是一种简单但是懒惰的算法。他的训练数据都是有标签的数据,即训练的数据都有自己的类别。KNN 算法主要应用领域是对未知事物进行分类,即判断未知事物属于哪一类,判断思想是,基于欧几里得定理,判断未知事物的特征和哪一类已知事物的特征最接近。它也可以用于回归,通过找出一

个样本的 k 个最近邻居，将这些邻居的属性的平均值赋给该样本，就可以得到该样本的属性。最简单的机器学习算法之一。通过采用 KNN 算法模式结合云计算从而达到识别并分析病例数据的目的。

2.4.2 KNN 算法思想

KNN 算法用于分类的核心思想是：存在一个样本数据集合，也称训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每个数据与其所属分类的关系。输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据(最近邻)的分类标签。一般来说，我们只选择样本数据集中前 k 个最相似的数据，这就是 k 近邻算法中 k 的出处(通常 $k < 20$)。最后，我们选择 k 个最相似数据中出现次数最多的分类，作为新数据的分类。

KNN 算法用于回归的核心思想是：找到近邻的 k 个样本，然后取平均值作为未知样本的值，对其进行预测。

2.4.3 KNN 算法步骤

算法步骤如下：

- 1) 算距离：给定未知对象，计算它与训练集中的每个对象的距离；
- 2) 找近邻：圈定距离最近的 k 个训练对象，作为未知对象的近邻；
- 3) 做分类：在这 k 个近邻中出线次数最多的类别就是测试对象的预测类别。

2.4.4 KNN 算法核心知识

1) 距离或相似度的衡量

在 KNN 算法中常使用欧氏距离、曼哈顿距离和夹角余弦来计算距离从而来衡量各个对象之间的非相似度。在实际中使用哪一种衡量方法需要具体情况具体分析。对于关系型数据，常使用欧氏距离；对于文本分类来说，使用夹角余弦(cosine)来计算相似度就比欧式(Euclidean)距离更合适。

2) k 值的选取

在 KNN 算法中 k 的选取非常重要，KNN 分类的准备率对 K 值很敏感。不同的值有可能带来不同的结果。如果 K 选大了的话，可能求出来的 k 最近邻集合可能包含了太多隶属于其它类别的样本点，不具有代表性，最极端的就是 k 取训练集的大小，此时无论输入实例是什么，都只是简单的预测它属于在训练实例中最多的类，模型过于

简单，忽略了训练实例中大量有用信息。如果 K 选小了的话，结果对噪音样本点很敏感。在实际中，一般采用交叉验证（一部分样本做训练集，一部分做测试集）或者依靠经验的方法来选取 k 值。 k 值初始时取一个比较小的数值，之后不断来调整 K 值的大小来使得样本分类最优，最优时的 K 值即为所选值。 k 值一般为奇数。有一个经验规则： k 一般低于训练样本数的平方根。

3) 类别的判定

投票决定：少数服从多数，在 k 个近邻中哪个类别的点最多就分为哪类。

加权投票法：根据距离的远近，对近邻的投票进行加权，距离越近则权重越大（权重为距离平方的倒数）。这是考虑到各个对象的相似度有悬殊，不同距离的样本有可能对未知样本产生的影响不同。

4) k 个邻近样本的选取

在 KKN 算法中，整个样本集中的每一个样本都要与待测样本的进行距离的计算，然后在其中取 k 个最近邻。但这带来了巨大的距离计算量，这也就是懒惰算法所带来的计算成本。改进方案有两个：一个是对样本集进行组织与整理，分群分层，尽可能将计算压缩到在接近测试样本邻域的小范围内，避免盲目地与训练样本集中每个样本进行距离计算。另一个就是在原有样本集中挑选出对分类计算有效的样本，使样本总数合理地减少，以同时达到既减少计算量，又减少存储量的双重效果。KD 树方法采用的就是第一个思路，压缩近邻算法采用的是第二个思路。

2.4.5 KNN 案例分析

可检测病例分析表如图所示

表 3-2 病例分析一览表

诊断结论		
窦性心律	窦性心律	窦性心律
	窦性心律失常	窦性心律不齐
		窦性心动过速
		窦性心动过缓
快速型心律失常	早搏	室性早搏
		交界性早搏
		房性早搏
	心动过速	室性心动过速
		室上性心动过速
	扑动颤动	房扑
		房颤
	预激综合症	WPW
缓慢型心律失常	逸搏	室性逸搏
		房性逸搏
		交界性逸搏
	房室传导阻滞	一度房室传导阻滞
		二度房室传导阻滞
		三度房室传导阻滞
	室内传导阻滞	室内传导阻滞
	其他	停搏
		噪声过大

表 3-3 分析参数一览表

诊断结论		备注
分析参数	心率	
	P 波长度	
	PR 间期	
	QRS 间期	
	QT 间期	
	QTc 间期	
	T 波长度	

表 3-4 算法预测一览表

		医生诊断	
		异常_阳性 P	
算法预测	异常(阳性 P)	TP	FP
	正常(阴性 N)	FN	TN

• 灵敏度 (Recall) = $TP / (TP + FN)$

o 灵敏度的取值范围为 0 到 1，灵敏度越高，说明假阴性 (FN) 比例越低，即漏诊率（有病，算法未检出）越低

• 特异度 (Specificity) = $TN / (FP + TN)$

o 特异度的取值范围为 0 到 1，特异度越高，说明假阳性 (FP) 比例越低，即误诊率（没病，算法却告诉我有病）越低

• ROC 曲线下面积 (AUROC)：以 Recall 与 Specificity 为 X-Y 轴绘制 ROC 曲线（受试者工作特征曲线），曲线下面积 (AUROC) 用于衡量不同算法模型的整体优劣。

o AUROC 的取值范围为 0 到 1，AUROC 越大，说明算法整体表现越好

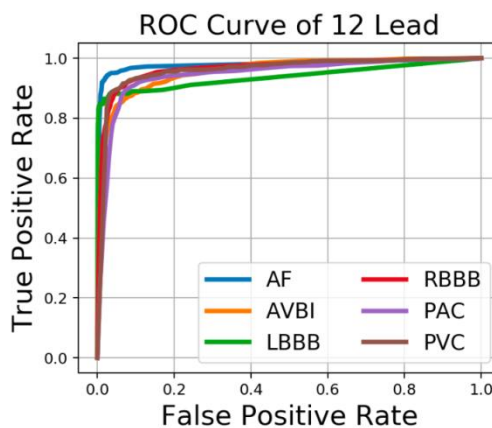
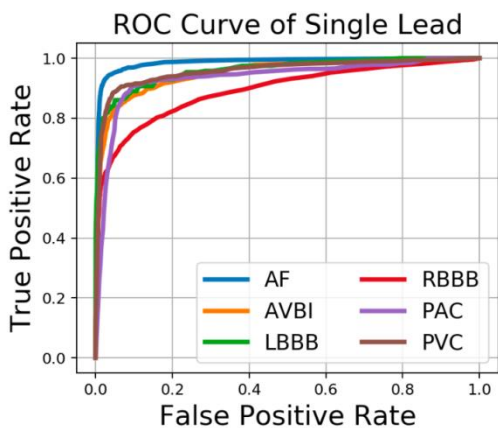
数据集一：CPSC

中国心电竞赛开源数据：The China Physiological Signal Challenge 2018: Automatic identification of the rhythm/morphology abnormalities in 12-lead ECGs.

- 样本量：6877 例，平均长度为 30 秒的 12 导联 ECG，取 I 导联为单导联测试数据
- 数据来源：11 家医院
- 标注方式：未知

测试结果 (AUROC)

	AF	AVBI	LBBI	RBBB	PAC	PVC
Single Lead	0.9857	0.9508	0.9597	0.8927	0.9343	0.9578
12-lead	0.9789	0.9579	0.9385	0.9655	0.9462	0.9609



数据集二

三甲医院心电科门诊数据 (私有)

- 样本量：测试数据集 15437 例
- 标注方式：从心电科医生的原始文本诊断中抽取

测试结果 (Recall、Specificity、AUROC)

诊断结论	标签	Recall	Specificity	AUROC
窦性心律	SN	0.9996	0.8182	0.9918
窦性心律不齐	SNA	0.7206	0.9692	0.9664
室性早搏	PVC	0.9138	0.9973	0.9943
房性早搏	PAC	0.7905	0.9910	0.9768
房扑	AFL	0.9091	0.9998	0.9997
房颤	AF	0.8824	1.0000	0.9998
左束支传导阻滞	LBBB	0.9524	0.9996	0.9997
右束支传导阻滞	RBBB	0.7778	0.9980	0.9934

2.5 图形界面设计

2.5.1 基本组成

本设计使用的嵌入式图形界面主要由 MCU、RAM、FLASH 和显示屏幕组成。MCU 的作用是从 FLASH 中读取图像数据存储在 RAM 中进行计算，并将像素数据从 RAM 传输到显示器上。FLASH 主要是用来存放图形数据，包括图像，文本等信息。RAM 主要是用来存放图像数据，以及充当显存的帧缓冲区。显示器是用来进行图像显示。

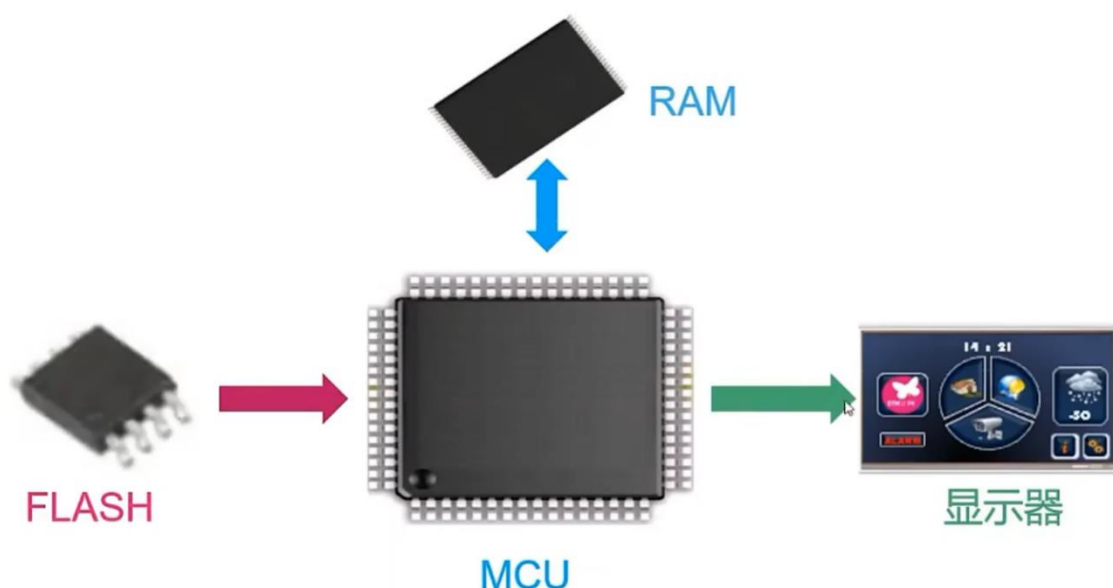


图 3-4 图形界面基本组成

2.5.2 外设及工作原理

LCD-TFT 显示控制器(LTDC)是 AHB 架构上的主设备，可以对内部和外部存储器进行读取访问。它有两个独立的图层，每层都有各自的 FIFO，从而使显示更加灵活。LTCD 控制器以 AHB 总线的速度自动从帧缓冲区提取数据，然后将图形数据存储在其中的一个 FIFO 中，通过整合之后将数据输出到显示器上。该系统架构可以使图形在没有任何 CPU 接入的情况下构建并绘制到屏幕上。在本设计中，LTDC 外设配置情况如图 3-6 所示。

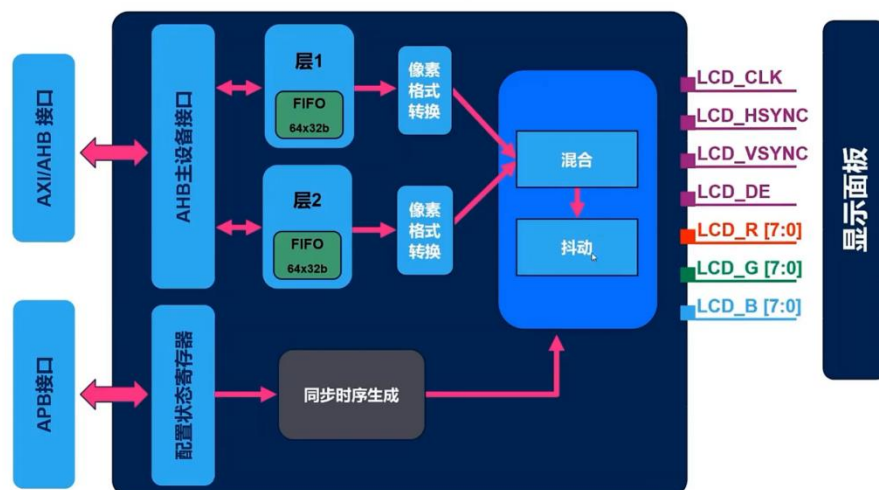


图 3-5 LCD-TFT 显示控制器 (LTDC)

Parameter Settings	Layer Settings	User Constants	NVIC Settings	Parameter Settings	Layer Settings	User Constants	NVIC Settings
Configure the below parameters :				Configure the below parameters :			
<div> <div>Search (Ctrl+F)</div> <div> <div>▼ Synchronization for Width</div> <div>Horizontal Synchronization Width1 pixels</div> <div>Horizontal Back Porch80 pixels</div> <div>Active Width800 pixels</div> <div>Horizontal Front Porch200 pixels</div> <div>HSync Width0</div> <div>Accumulated Horizontal Back Porch Width80</div> <div>Accumulated Active Width880</div> <div>Total Width1080</div> </div> <div> <div>▼ Synchronization for Height</div> <div>Vertical Synchronization Height1 lines</div> <div>Vertical Back Porch40 lines</div> <div>Active Height480 lines</div> <div>Vertical Front Porch22 lines</div> <div>VSycn Height0</div> <div>Accumulated Vertical Back Porch Height40</div> <div>Accumulated Active Height520</div> <div>Total Height542</div> </div> <div> <div>▼ Signal Polarity</div> <div>Horizontal Synchronization PolarityActive Low</div> <div>Vertical Synchronization PolarityActive Low</div> <div>Data Enable PolarityActive Low</div> <div>Pixel Clock PolarityNormal Input</div> </div> <div> <div>▼ BackGround Color</div> <div>Red0</div> <div>Green0</div> <div>Blue0</div> </div> </div>				<div> <div>Search (Ctrl+F)</div> <div> <div>▼ Windows Position</div> <div>Layer 0 - Window Horizontal Start0</div> <div>Layer 0 - Window Horizontal Stop800</div> <div>Layer 0 - Window Vertical Start0</div> <div>Layer 0 - Window Vertical Stop480</div> </div> <div> <div>▼ Pixel Parameters</div> <div>Layer 0 - Pixel FormatRGB565</div> </div> <div> <div>▼ Blending</div> <div>Layer 0 - Alpha constant for blending255</div> <div>Layer 0 - Default Alpha value0</div> <div>Layer 0 - Blending Factor1Alpha constant x Pixel Alpha</div> <div>Layer 0 - Blending Factor2Alpha constant x Pixel Alpha</div> </div> <div> <div>▼ Frame Buffer</div> <div>Layer 0 - Color Frame Buffer Start Adress0xD0000000</div> <div>Layer 0 - Color Frame Buffer Line Length (lma...800</div> <div>Layer 0 - Color Frame Buffer Number of Lines (...480</div> </div> <div> <div>▼ BackGround Color</div> <div>Layer 0 - Blue0</div> <div>Layer 0 - Green0</div> <div>Layer 0 - Red0</div> </div> <div> <div>▼ Number of Layers</div> <div>Number of Layers1 layer</div> </div> </div>			

图 3-6 LTDC 配置情况

图形加速器 Chrom-ART(DMA2D)提供了图形处理的硬件加速，作为面向图形的 DMA2D，它支持平面混合和像素格式转换，并且具有抗锯齿的特点模式。在本次设计中，使用内存到内存工作模式，它可以在不更改颜色格式的情况下，将部分或者整个源图像复制制到部分或者整个目标图像中去。本设计中 DMA2D 配置如图 3-8 所示。

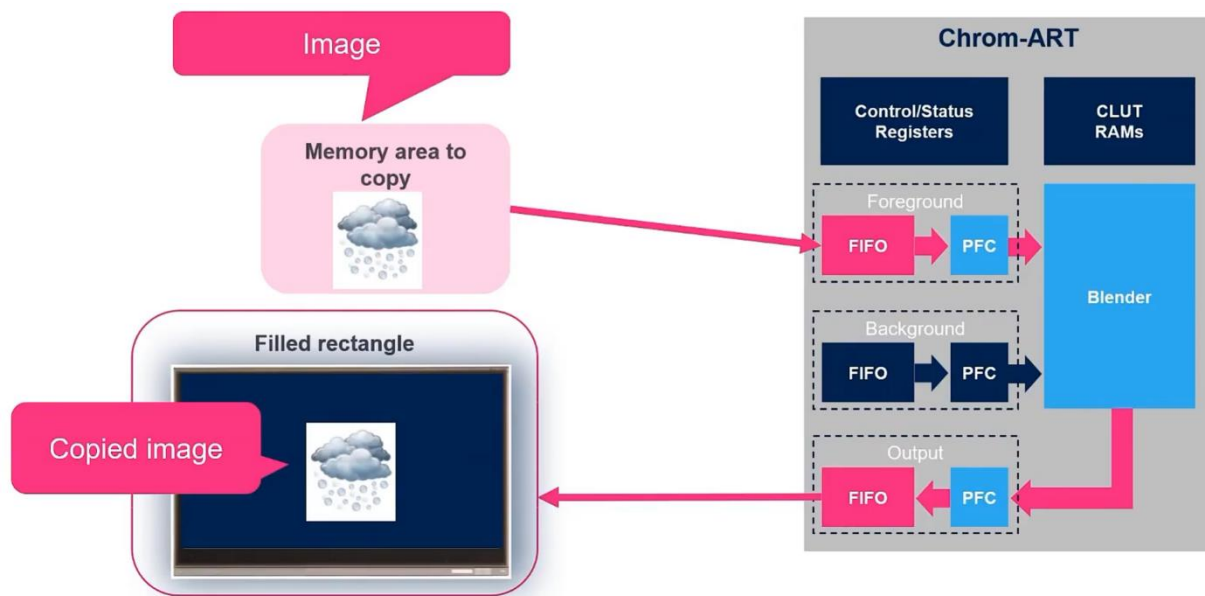


图 3-7 DMA2D 内存到内存工作模式示意图

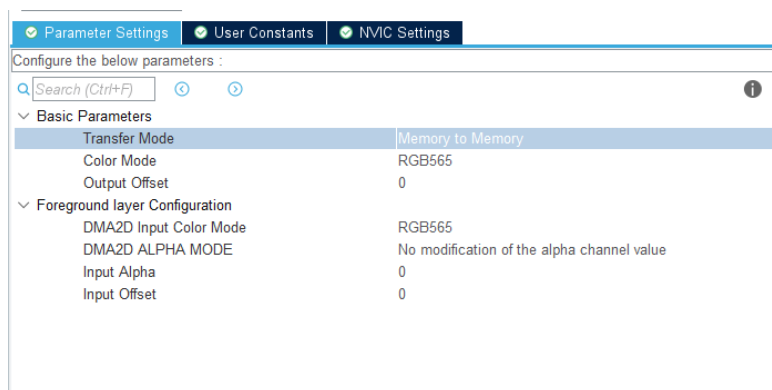


图 3-8 DMA2D 配置情况

图像显示数据流主要分为两步，第一步通过 MCU 内部 Chrom-ART 图形加速器将 FLASH 中的图片字体等图形资源搬运到 SDRAM 中来更新图形缓冲区数据。第二步通过 LTDC 控制器将帧缓冲区的数据传输到显示器上进行显示。

2.5.3 图形引擎

在本设计中，使用的图形引擎为 touch-GFX 图形引擎，它主要职责就是在一定的时间内对显示设备执行绘制操作。这个 touch-GFX 的图形引擎主要分为模式图形引擎和保留模式图形引擎两种。通过调用即时模式图形引擎的 API，应用程序可以直接将内容绘制

到屏幕上，应用程序负责在正确的时间调用正确的绘图操作。保留模式图形引擎则为用户提供了显示组件的抽象模型，用户只需操作抽象模型，touch-GFX 负责将模型转化为一组优化的渲染方案调用。touch-GFX 图形引擎主要包含三部分：收集、更新和渲染。收集主要是从触摸屏收集事件，接收来自后端的消息等。更新主要是对收到的事件做出反应，如更新对象模型的位置、动画、颜色等。渲染主要是重画模型中已更新的部分，并存储到帧缓冲区。

2.5.4 软件架构(MVP)

本设计中使用的软件架构是 Model-View-Presenter 的数据模式，是一种将代码分为类的结构化方法，将逻辑从图像中分离处理，可以对代码进行独立测试，使程序在重用、维护方面非常简便。在 MVP 的模式下，View 层的主要任务为负责界面的显示，Presenter 层主要进行的任务是逻辑处理和调用，Model 则提供来自后台的数据和作为切换界面时数据交换的介质。Model 中有一个 tick 函数，每帧会被调用一次，View 并不直接接触 Model，他们之间的交互是通过 Presenter 进行的。软件架构图如图 5-7 所示。本设计的主应用程序采用 C 语言编程，图形界面 MVP 架构采用 C++编程，图形界面与后端系统的通信是通过 Model 类实现的，后端系统可以从 UI 接收事件，也可以将事件传给 UI。在此架构中，可以有一个或者多个界面，并有多与之对应的 Presenter，每个时刻只有一个界面处于活跃状态，从而达到节省内存的目的。

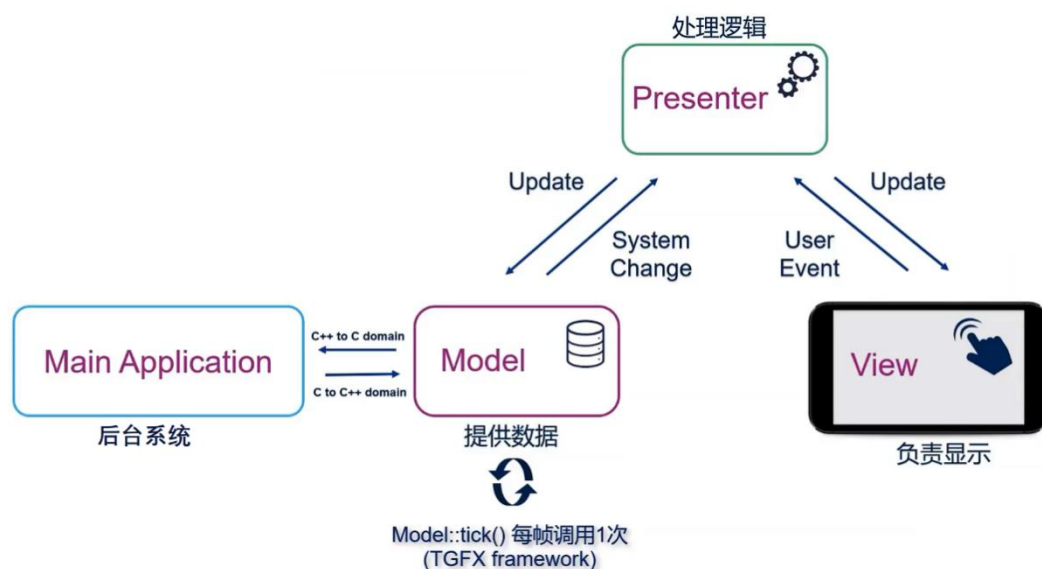


图 3-9 MVP 软件架构

第三部分 完成情况及性能参数

3.1 系统测试所用仪器

在系统测试中使用的主要仪器仪表如下：

表 4-1 测试用仪表

仪器名称	规格型号	数量	技术性能
示波器	安捷伦 DSO-X2012A	1	100MHz、2GSa/s
心电信号模拟器	FK550	1	输出心率信号 30~110 次/分钟
测温万用表	FLUKE 1587C	1	量程-40℃~537℃、分辨率 0.1 °C

3.2 系统心率测试

3.2.1 心率测试方案

(1) 标准心电信号检测：

标准心电信号测试中使用心电信号模拟仪(FK550)同时接入示波器与本传感器节点系统，将模拟器输出心率，示波器采集频率扩大 60 倍，本设计采集处理后计算出的心率进行对比。

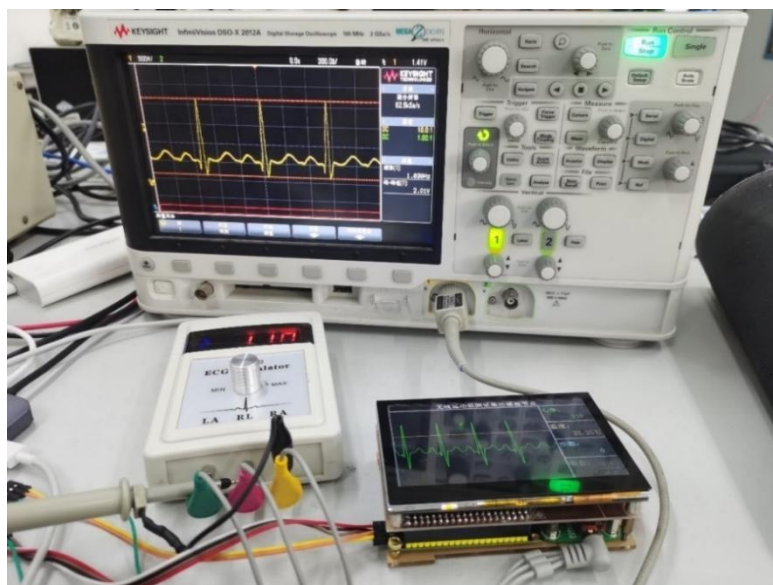


图 4-1 标准心电信号测试

(2) 人体心电测试：

由于实际的人体心电信号具有幅值小频率低的特点，很容易受到连线、仪器等多种因素干扰，因此不能简单地直接用示波器进行观察。人体测试中，使用三星 S10 系列手机上带有的心率传感器和小米手环作为参考，通过心电极片使用三导联线连接人体与本传感器节点系统，同时在人体手腕处佩戴小米手环，手指触摸手机上的心率传感器，三个设备同时测量心率，然后进行对比。



图 4-2(a) 人体心电测试



图 4-2(b) 手机心率传感器



图 4-2(c) 手环测量心率

3.2.2 心率测试结果与分析

在标准心电信号检测中,心电信号模拟仪(FK550)产生心率在 50-110 之间的标准心电信号,输入进示波器和本设计的心电检测接口,得到的结果如表 6-2 所示。可以看到,由于心电信号模拟仪产生的信号较为干净,没有其他噪声,且峰峰值在 2V 以上本设计可以很好的完成对其产生的心电信号的采样,除最后一组数据模拟器产生的心率 110.04 太接近 109,而本设计对数据处理并没有采用四舍五入,从而导致识别成 109 外,其余心率计算准确度均为 100%。

表 4-2 标准信号测试数据

模拟仪(FK550)输出心率	示波器频率 (单位 Hz)	示波器频率 扩大 60 倍	传感器节点 测试心率	误差
51	0.854	51.24	51	0%
64	1.073	64.38	64	0%
75	1.256	75.36	75	0%
87	1.459	87.54	88	0%
96	1.609	96.54	96	0%
105	1.761	105.66	105	0%
110	1.834	110.04	109	0.91%

在人体心电信号测试中,通过对比不同用户同时使用手机自带的心率传感器、小米手环和本设计来测量心率,得到的结果如表 6-3 所示,可知测量的结果比较准确,误差基本在 2%以内,满足要求的范围。

表 4-3 人体测试数据

测试人员	手机自带 心率传感器	小米手环	传感器节点 测试心率	与手机相比 误差	与手环相比 误差
人员 1	64	65	64	0%	1.54%
人员 2	70	70	71	1.43%	1.43%
人员 3	62	63	63	1.61%	0%
人员 4	58	59	58	0%	1.69%
人员 5	107	104	106	0.93%	1.92%

3.3 系统温度测试

3.3.1 温度测试方案

温度采集，将本无线传感器节点系统与测温万用表和温度计进行对比，将所有设备的探测头处于同一点，静止一会等待数据稳定之后测得的温度。调节测试的温度范围，多次测试之后进行对比。

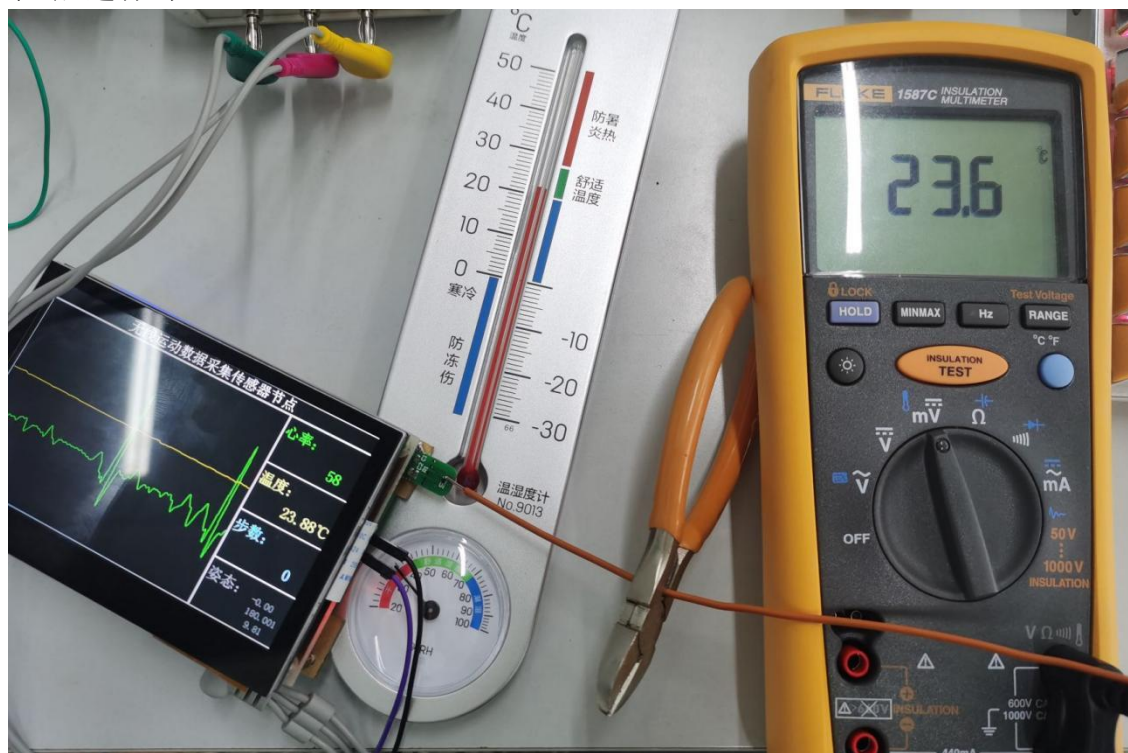


图 4-3 温度测试

3.3.2 温度测试结果与分析

在温度测试中，通过对比热电偶测温万用表，煤油温度计和本传感器节点同时测量同一部位的温度，得到的结果如表 2 所示，可以看到本设计测量的温度与温度计相比，误差

在 0.3°C 以内，与万用表相比误差在 0.2°C 以内。可知测量结果较为准确，在设计要求允许的范围内。

表 4-4 温度测试数据

温度计温度	测温万用表温度	传感器节点温度	温度计误差	与万用表误差与
23.9	24.0	24.19	0.29	0.19
25.6	25.4	25.32	0.28	0.08
28.5	28.5	28.35	0.15	0.15
29.2	29.1	29.15	0.05	0.05
31.4	31.3	31.21	0.19	0.09

3.4 系统功耗测试

功耗测试中，由于本系统供电仅需通过 USB Type-C 接口，使用 USB 电流表，在本系统供电线和充电宝直接串连，即可测试出系统供电的电压和电流，计算出系统功耗。



图 4-4 功耗测试

功耗测试的结果如图所示，可看到系统供电电压为 5.06V ，电流为 0.366A ，整体功率为 1.85W ，满足低功耗的要求。

3.5 性能测试总结

本部分主要对系统的各部分功能进行了对比测试，通过对比测试数据，可以看出系统的各部分功能均可很好的完成预期目标，误差也都在可控的范围内，系统整体功耗小于

2W，满足低功耗设计要求。

第四部分 总结

4.1 可扩展之处

本设计主要采用 STM32 单片机完成对使用者心率，体表温度，运动信息的采集，并通过 WiFi 将使用者的信息发送至移动设备进行显示。本设计在以下三个方面还有待完善。第一是心电信号采集，容易受到干扰。第二是运动信息采集准确度有待提高。第三是本设计与用户的交互界面有待优化。

具体的一些优化办法可以在心电信号采集的硬件部分单独隔离出来，将这部分电路重新绘制，之后加上屏蔽罩，尽可能最大限度的减少外接信号的干扰。或者采用更高级的软件滤波算法，比如卡尔曼滤波等，对数据进行最优估计，去除噪声和干扰信号的影响。而对于运动信息，可以换用更高主频的设备之后，将所有的姿态原始数据进行数字滤波处理，用单片机执行运动分析函数，解算出实时的姿态数据，在根据姿态数据重新计算步数。至于与用户交互的界面，可以在单片机上面跑 RTOS 系统之后移植 GUI，来带给用户如智能手机一般的体验。

4.2 心得体会

伴随着生活水平的提升，人们对关注自身健康的意识也在不断增强，具有健康监测、智慧医疗等功能便携式家庭健康设备正成为新的一个热点。而近年来，伴随着物联网，大数据，云计算和人工智能等多种技术的飞速发展，健康监测设备正在向智能便捷化发展，智能化的同时也存在着一些不足。比如由于现在健康监测设备行业发展正处于起步阶段，水平有限，各个厂商把重点放在设备的功能上，目前市面上的监测设备外观和材料的品质比较单一，而这也会影响用户的选择，根据调查显示，大部分购买健康监测设备的消费者为男性，占比可达到 85%。而且今后随着科技的进步，外观更漂亮，工艺更出色的健康监测产品将越来越吸引女性消费者的购买体验。还有受限于材料和能源技术，当前多数便携健康监测设备都有续航忧虑，这是当前所有移动设备中的短板，尚无有效解决方案。但随着传感器，集成电路技术的发展，未来将会出现更多集成了多种与生物信号采集有关传感器的 SoC 芯片，这必将大幅度降低系统功耗，延长移动健康检测设备的续航时间。总之，健康监测设备还有着巨大的发展潜力，未来将向着舒适性、便捷性、低功耗等方面深入发展，结合物联网万物互相，达到早期发现和早期诊断的目的，从而保证我们的身体健康。

第五部分 参考文献

- [1] 陈帝良,方震,赵湛,等. 集成多生理参数监测的终端设计[J]. 传感器与微系统,2016,35(1):95-97. DOI:10.13873/J.1000-9787(2016)01-0095-03.
- [2] 前瞻产业研究院.2021 中国可穿戴设备产业研究报告[J].智能制造, 2021,11(17):65-67.
- [3] 崔宇僊,金环. 便携式可穿戴医疗健康设备市场分析[J]. 时代金融,2018,(17):1.
- [4] Klingeberg,T.,Schilling,等.Mobile wearable device for long term monitoring of vital signs[J]. Practice.2012,106(2).89-96.doi:10.1016/j.cmpb.2011.12.009.
- [5] Kirk H, Shelley,Denis H,等.What is the best site for measuring the effect of ventilation on the pulse oximeter waveform?[J].Anesthesia and analgesia.2006,103(2).372-7,table Of cOntents.
- [6] 尹玉兴,刘振坚,项止武. 基于 Matlab 的 FIR 数字滤波器设计与仿真[J]. 工矿自动化,2011,37(12):60-63. DOI:32-1627/TP.20111128.1616.017.
- [7] 胡小华,李向攀,祁洋阳,等. 可穿戴式人体姿态检测系统设计[J]. 电子技术应用,2017,43(9):13-16. DOI:10.16157/j.issn.0258-7998.172804.
- [8] 魏阳,索忠伟,李文容. 数字滤波器在实时滤除心电干扰中的设计与应用[J]. 微计算机信息,2006,22(11):40-41,247. DOI:10.3969/j.issn.1008-0570.2006.11.014.
- [9] 商继敏,王海燕,蒋逢春,等. 最小二乘法对温度传感器测温数据 线性拟合及其应用[J]. 大学物理实验,2019,32(2):81-84. DOI:10.14139/j.cnki.cn22-1228.2019.02.023.
- [10]时琦. 基于 MPU6050 的四轴飞行器姿态角解算[J]. 科学技术创新,2019(4):38-39. DOI:10.3969/j.issn.1673-1328.2019.04.024.
- [11]古训,陈红. 基于 MPU6050 的软件与硬件姿态解算研究[J]. 电子世界,2019(22):5-7.
- [12]肖笛.可穿戴设备与人的融合[J].科技与创新,2015,24(10):95-97.

第六部分 附录

重要代码、推导过程等不便于在正文中体现的内容

ADS1292.c

```
//正常信号采集模式
u8 ADS1292_Single_Read(void)
{
    u8 res=0;
    Ads1292_Config2.Int_Test = INT_TEST_OFF;//关内部测试信号
    Ads1292_Ch1set.MUX = MUX_Normal_input;//普通电极输入
    Ads1292_Ch2set.MUX = MUX_Normal_input;//普通电极输入

    if(ADS1292_WRITE_REGBUFF())//写入寄存器
        res=1;
    delay_ms(10);
    return res;
}

//配置ads1292采集方式
u8 Set_ADS1292_Collect(u8 mode)
{
    u8 res;

    delay_ms(10);
    switch(mode)//设置采集方式
    {
        case 0:
            res =ADS1292_Single_Read();
            break;
        case 1:
            res =ADS1292_Single_Test();
            break;
        case 2:
            res =ADS1292_Noise_Test();
            break;
    }
    if(res)return 1; //寄存器设置失败
    ADS1292_Send_CMD(RDATAC); //启动连续模式
    delay_ms(10);
    ADS1292_Send_CMD(START); //发送开始数据转换（等效于拉高START引脚）
    delay_ms(10);
    return 0;
}
```



```
//设置通道1内部1mV测试信号
u8 ADS1292_Single_Test(void) //注意1292R开了呼吸解调，会对通道一的内部测试信号波形造成影响，这里只参考通道2即可，1292不受影响
{
    u8 res=0;
    Ads1292_Config2.Int_Test = INT_TEST_ON;//打开内部测试信号
    Ads1292_Ch1set.MUX=MUX_Test_signal;//测试信号输入
    Ads1292_Ch2set.MUX=MUX_Test_signal;//测试信号输入

    if (ADS1292_WRITE_REGBUFF())//写入寄存器
    {
        res=1;
        delay_ms(10);
        return res;
    }
    //设置内部噪声测试
    u8 ADS1292_Noise_Test(void)
    {
        u8 res=0;
        Ads1292_Config2.Int_Test = INT_TEST_OFF;//关内部测试信号
        Ads1292_Ch1set.MUX = MUX_input_shorted;//输入短路
        Ads1292_Ch2set.MUX = MUX_input_shorted;//输入短路

        if (ADS1292_WRITE_REGBUFF())//写入寄存器
        {
            res=1;
            delay_ms(10);
            return res;
        }
    }
}
```

Main（）函数

```
ES8266_MQTT_Init();
StatusReport();

ADS1292_PowerOnInit(); // ADS1292上电初始化
while(Set_ADS1292_Collect(0))//0 正常采集 //1 1mV1Hz内部测试信号 //2 内部短接噪声测试
{
    printf("1292寄存器设置失败\r\n");
    delay_ms(500);
    LED1=!LED1;
}

Value = Get_Adc();

while (1)
{
    /*串口*/
    if (USART1_RX_STA&0x8000)
    {
        len=USART1_RX_STA&0x3fff; //得到此次接收到的数据长度
        printf("\r\n您发送的消息为:\r\n");
        HAL_UART_Transmit(&huart1,(uint8_t*)USART1_RX_BUF,len,1000); //发送接收到的数据
        while(_HAL_UART_GET_FLAG(&huart1,UART_FLAG_TC)!=SET); //等待发送结束
        printf("\r\n\r\n");//插入换行
        USART1_RX_STA=0;
    }
    /*心率*/
    if(ads1292_recive_flag)
    {
        arm_fir(&Input_data2,&Output_data2);
        ECG_UI_refresh(&ecg_win,ecg_x_value++,Output_data2); //刷新示波器串口（以屏幕像素点为单位）
        printf("B: %8d,%8d,%d\r\n", (u32)Input_data2, (u32)Output_data2, heart_rate);
        ads1292_recive_flag=0;
    }
}
```

