

使用Arduino的 实时心电指纹识别

《电子系统设计》课程项目

王润，田野，张正泉

目录

- 项目目的
- 思路介绍
- 数据采集
 - 树莓派方案
 - Arduino方案
- 数据处理
- 模型训练
- 实时识别
 - 识别流程
 - 识别选项
- 分工情况

项目目的

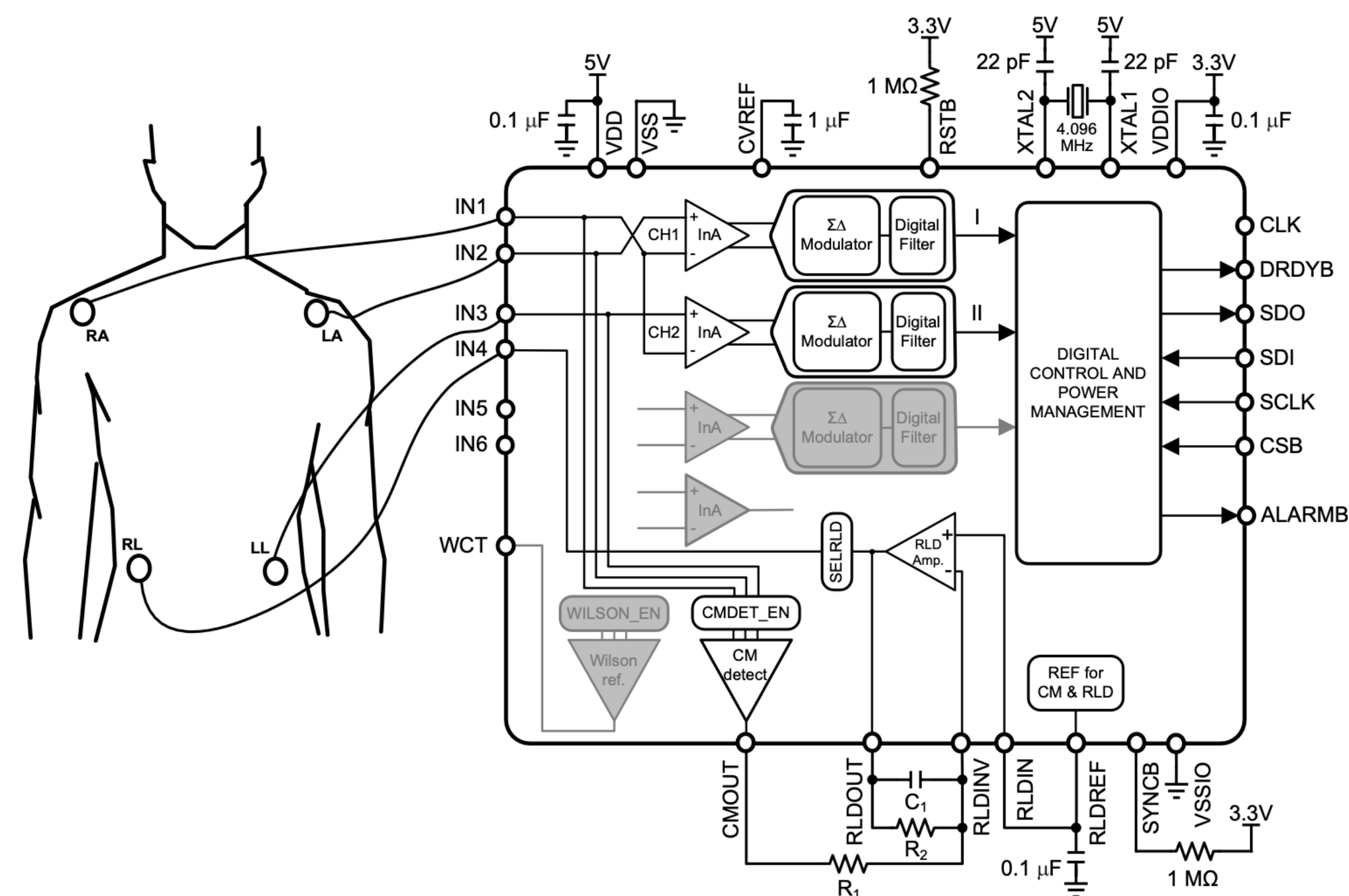
- 硬件上使用一款生理信号采集芯片采集心电数据
- 在上位机上编写程序训练模型，实现心电信号识别
- 具体要求
 - 人体上连接三导联信号采集线，信号持续将数据传入上位机，上位机进行识别
 - 模型识别成功率大于80%
 - 模型推断过程耗时少，实现实时推断

思路介绍

- 硬件部分
 - 使用ADS1293模拟前端采集心电数据
 - 使用树莓派通过SPI协议接收数据/使用Arduino接收数据通过串口上传电脑
- 软件部分
 - 数据清洗、滤波、分帧、添加类别信息
 - 训练机器学习模型进行分类
 - 实时处理并识别心电信号

数据采集

- ADS1293芯片
 - 3通道，24位模拟前端
 - 3导联连线方式
 - 采样频率853Hz
 - 使用SPI协议输出心电数据



ADS1293三导联接线图

SPI协议

- SPI是串行外设接口(Serial Peripheral Interface)的缩写。是 Motorola 公司推出的一种同步串行接口技术，是一种高速的，全双工，同步的通信总线。
- 根据这个协议，我们可以通过设置ADS1293的寄存器来输出采集到的数据。
- 优点
 - 支持全双工通信
 - 通信简单
 - 数据传输速度快
- 缺点
 - 没有指定的流控制，没有应答机制确认是否接收到数据，所以跟IIC总线协议比较在数据可靠性上有一定的缺陷。

SPI Protocol

A typical serial interface access cycle is exactly 16 bits long, which includes an 8-bit command field (R/WB + 7-bit address) to provide for a maximum of 128 direct access addresses, and an 8-bit data field. [Figure 28](#) shows the access protocol used by this interface. Extended access cycles are possible and they are described in the [Auto-incrementing Address](#) and [Streaming](#) sections.

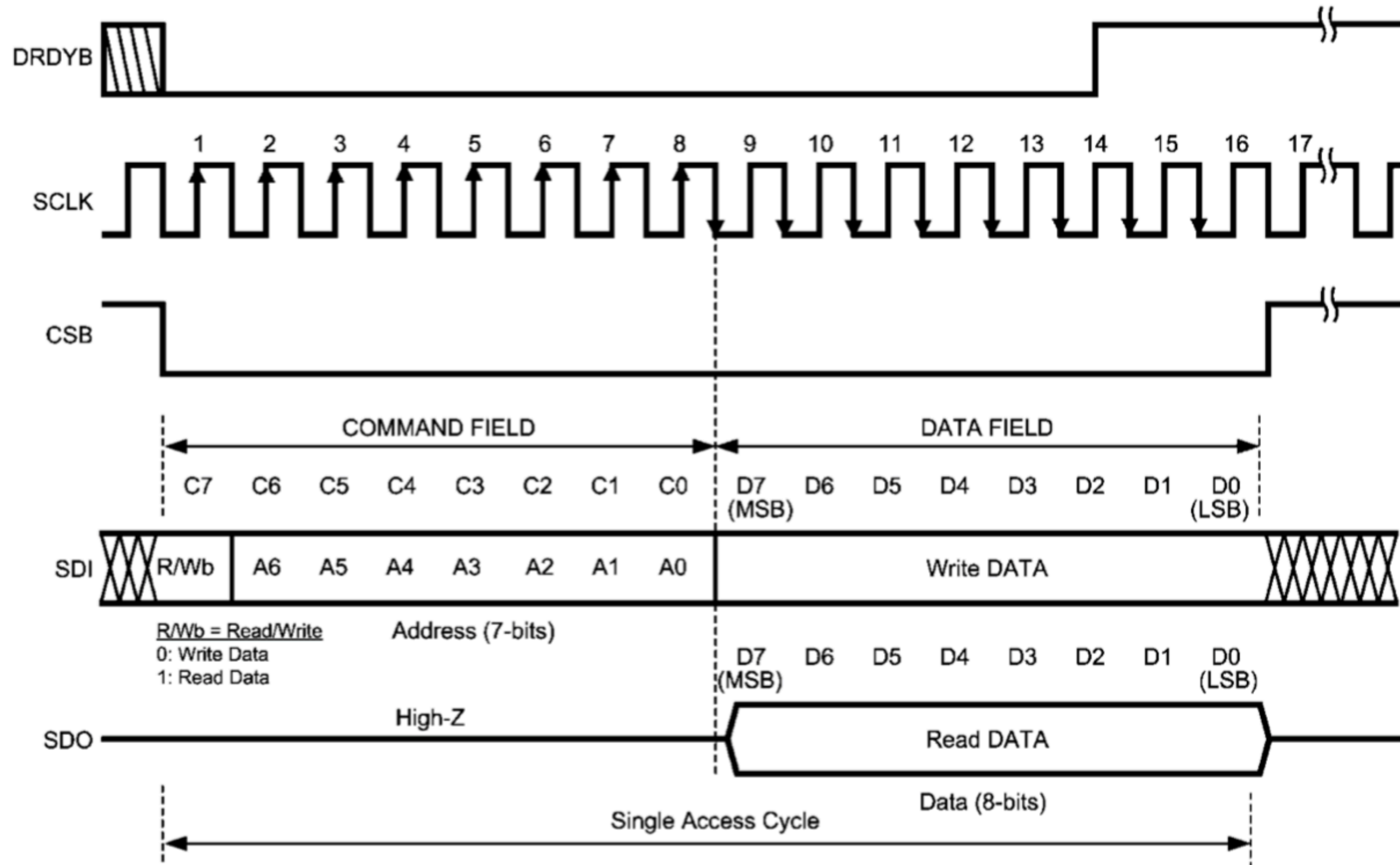
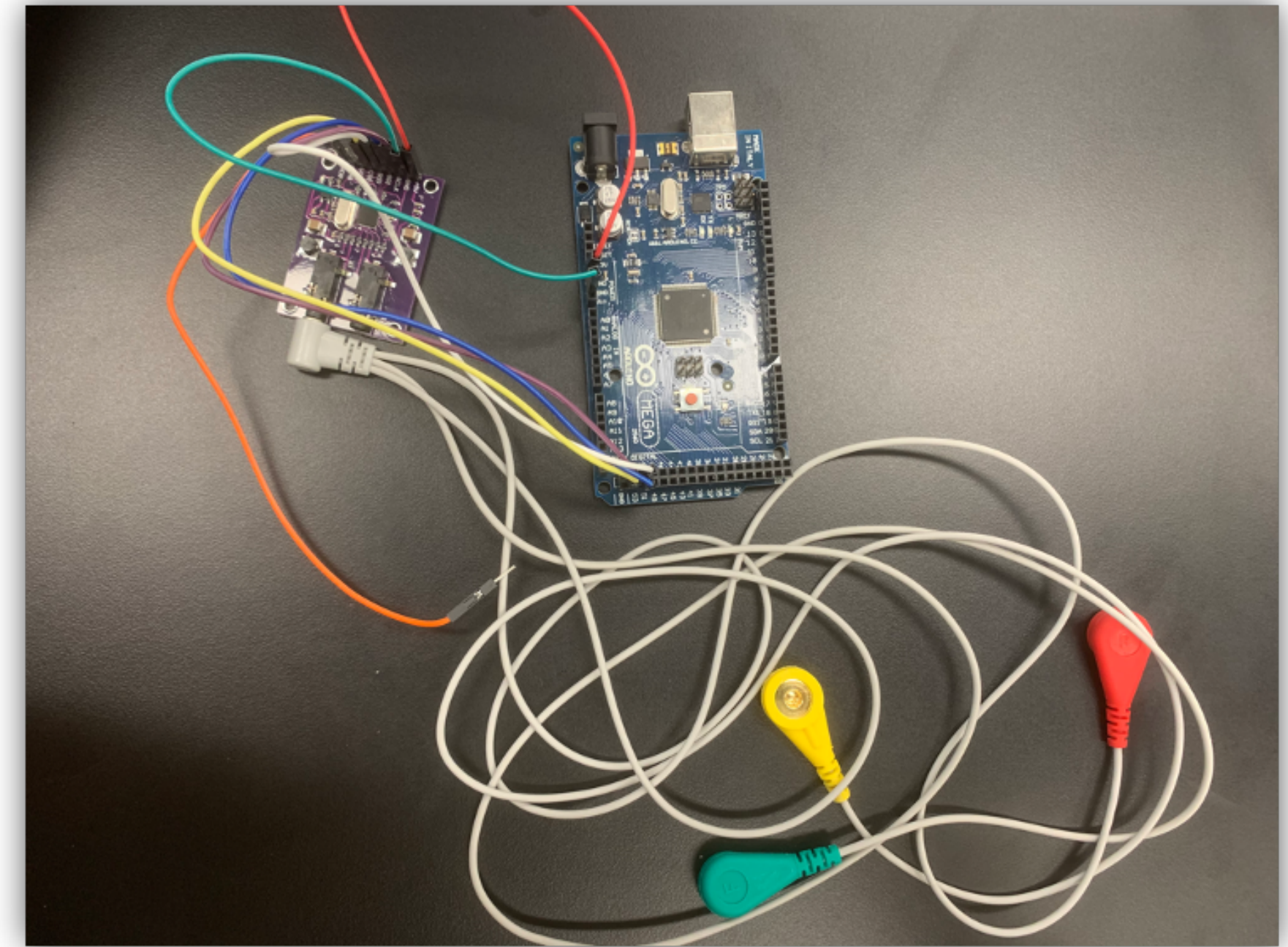


Figure 28. Serial Interface Protocol

硬件方案

- 接收ADS1293传出数据的方法
- 借助RPi.GPIO库与器件手册，使用树莓派的GPIO接口设定ADS1293寄存器，根据SPI协议读取数据（失败）
- 借助Arduino的SPI接口，使用Arduino提供的函数从ADS1293读取数据经过pyserial模块转入python处理。（成功）



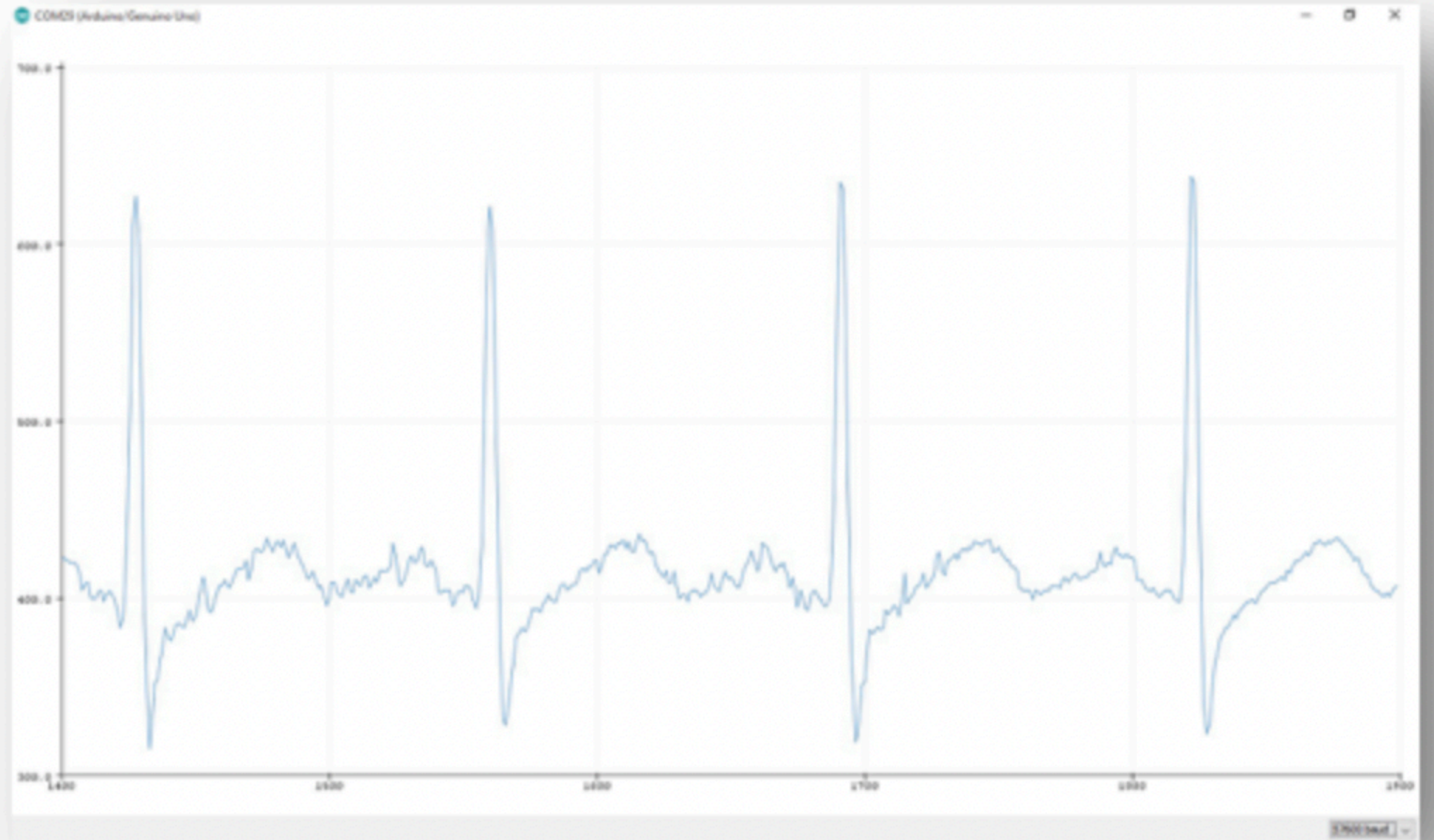
三导联、ADS1293与Arduino接线图

Pyserial

- Pyserial模块封装了python对串口的访问，为多平台的使用提供了统一的接口。
- 可以通过list_ports函数获得所有可用串口，也可以通过write或read函数对串口进行读写。
- 因为Arduino封装了数据传输过程，因此软件内部只能讲串口读入的数据绘制成图打印在屏幕上。我们需要使用pyserial来将数据从串口中读取出来并保存。
- 注意：arduino串口只有一个，因此我们不能同时观察波形和读取保存数据。

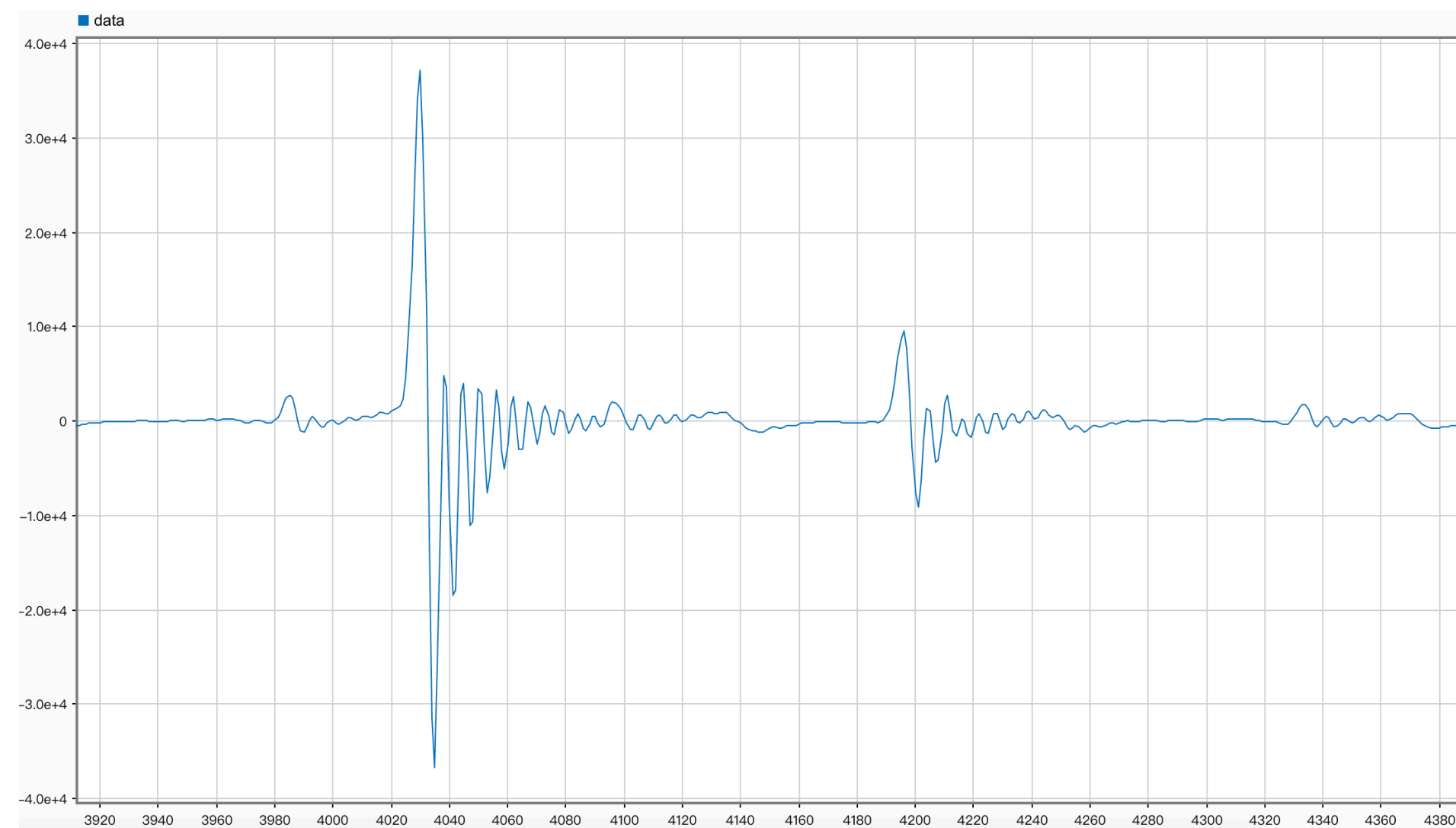
心电波形

- 如果芯片没有出现接触不良且贴片连接良好，那么arduino窗口中应该能看到正确的心电波形。

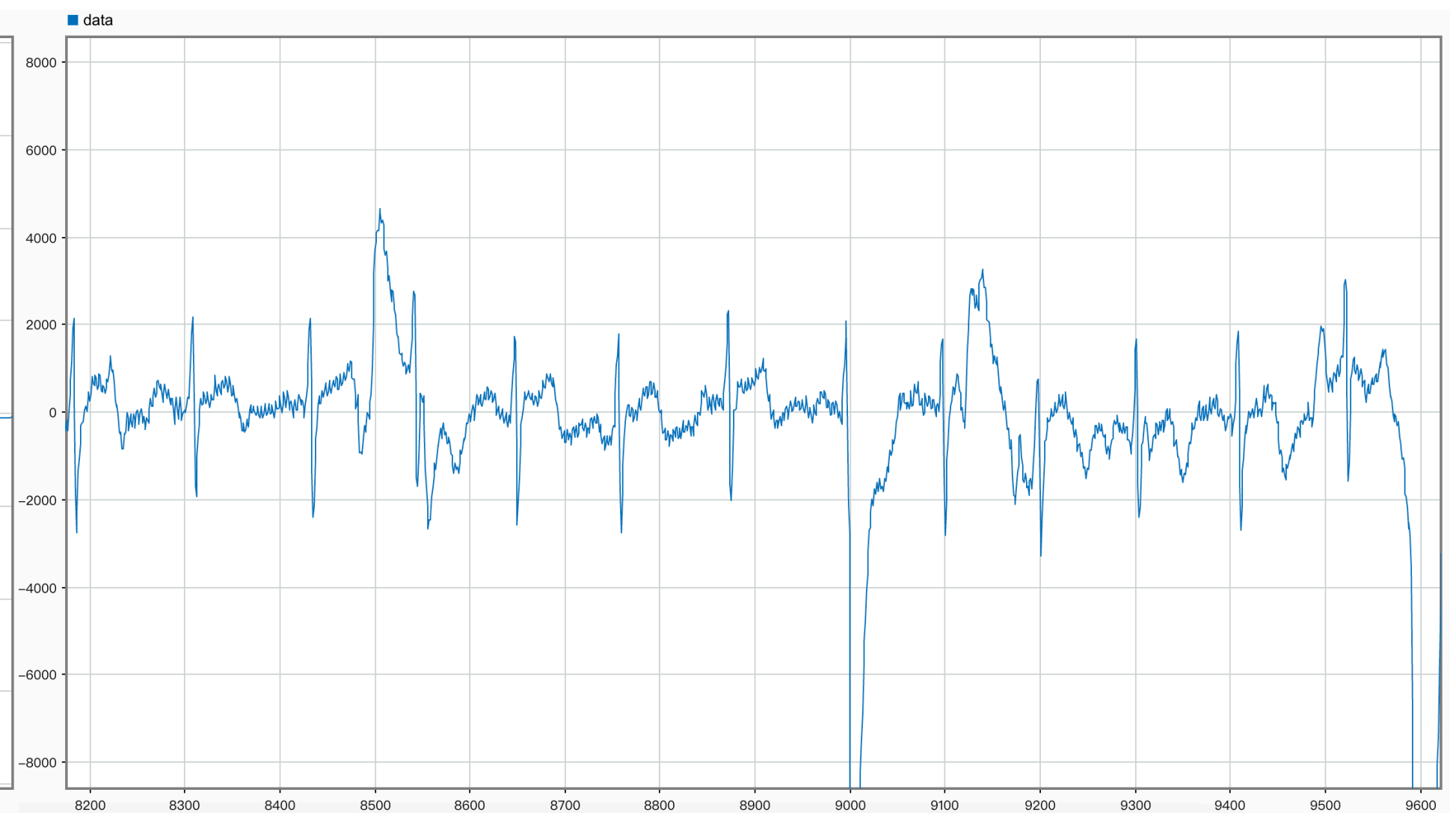


数据处理

- 去除质量较差部分
 - 24位量化下幅值超过3000的部分
 - 人工判断存在干扰的部分



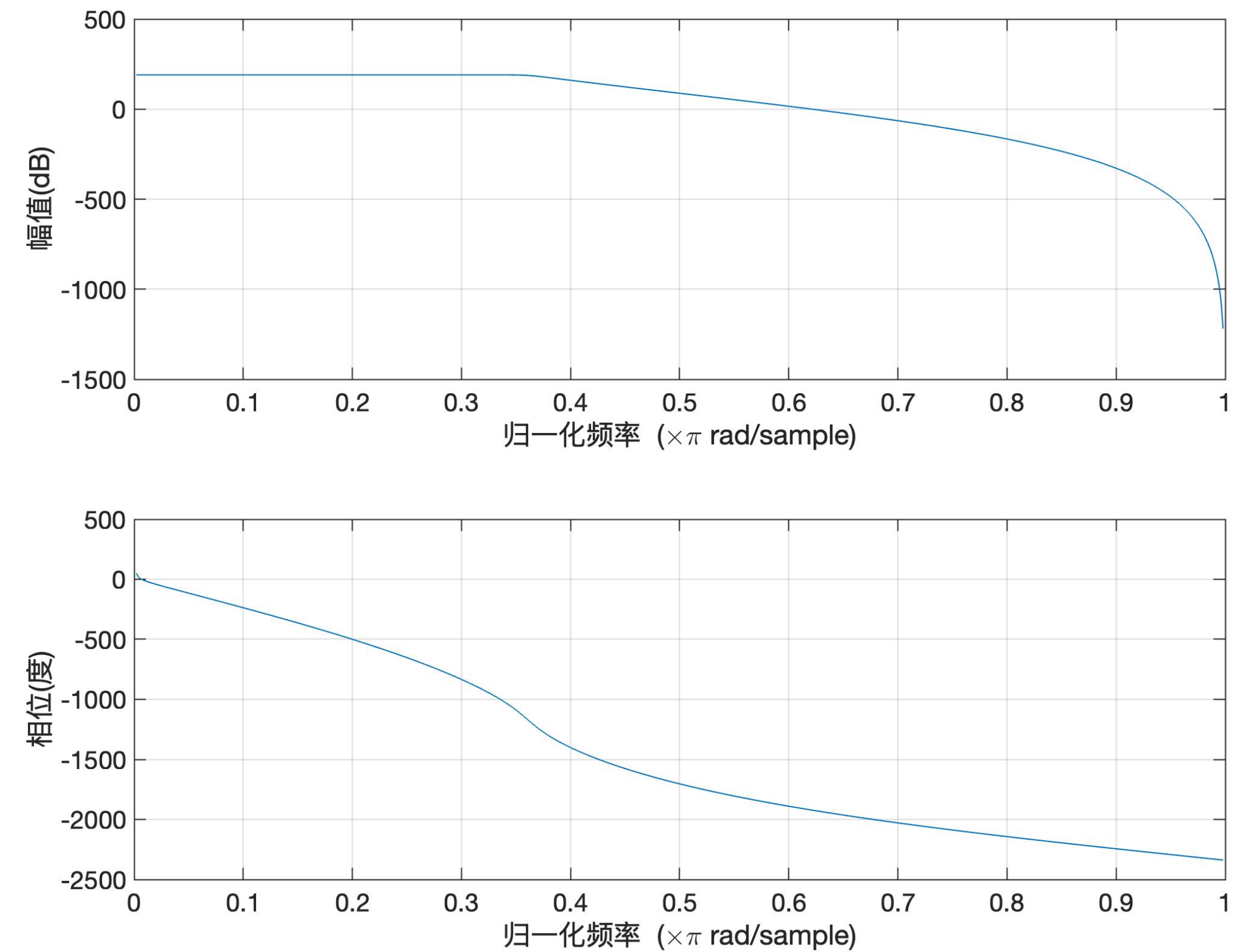
幅值过大



存在干扰

数据处理

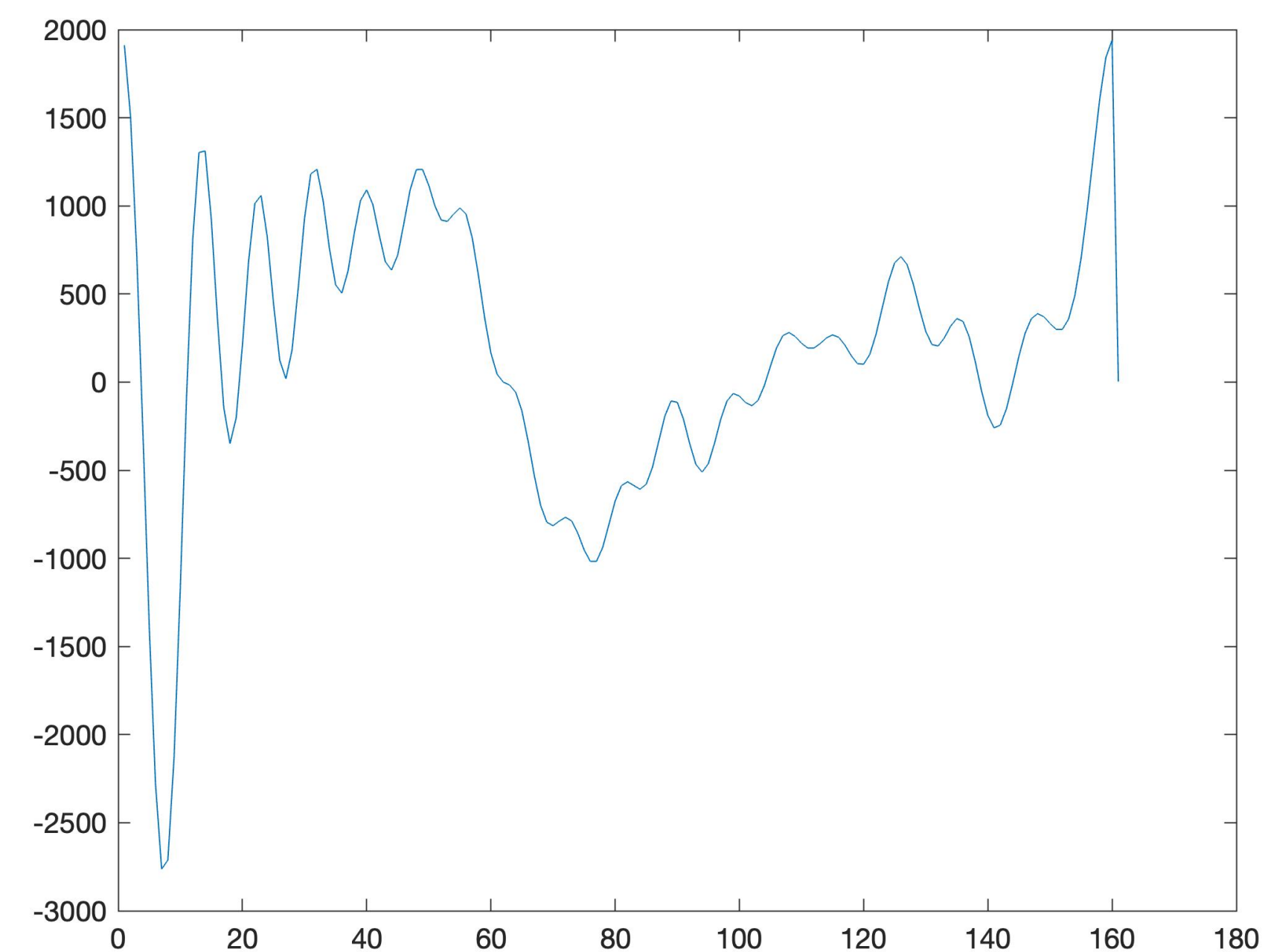
- 心电信号特点
 - 能量集中在0.25~35Hz
- 设计滤波器进行带通滤波
 - 通带0.05~150Hz
 - 采样频率853Hz



带通滤波器频率响应

数据处理

- 根据峰值进行分帧
 - 仅保留大于阈值的峰值
 - 阈值因人而异
- 帧处理
 - 使用三次样条插值至160样本点
 - 添加类别为最后一个样本点



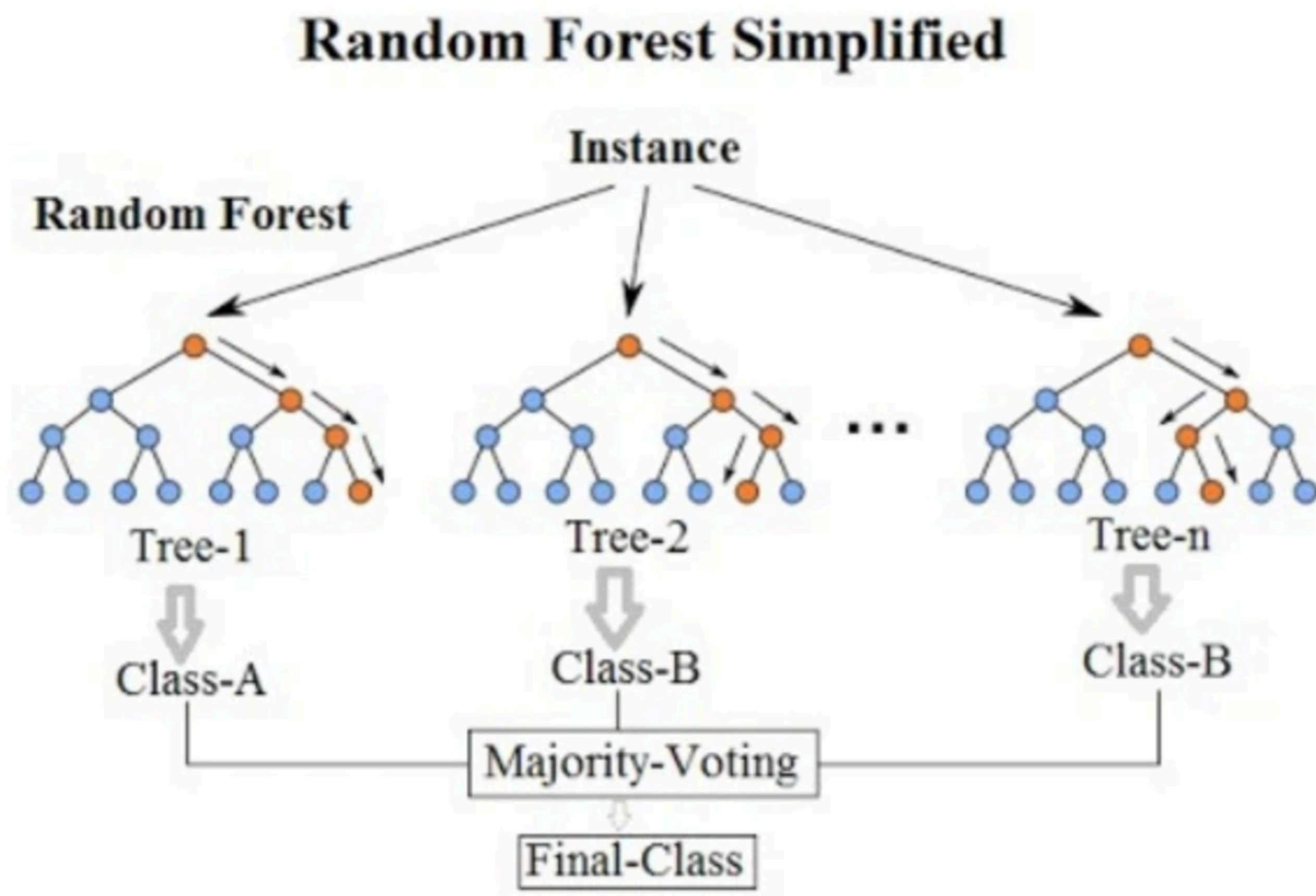
分帧结果，最后一个样本点为类别

模型训练

- 尝试使用随机森林模型、循环神经网络和卷积神经网络等方法
 - 输入每帧数据经样条插值后结果
 - 输出数据所属类别
- 测试集上准确率为89%（随机森林），98%（卷积神经网络）
- 综合考虑数据集样本数、样本幅度变化范围、推理和训练速度，使用随机森林模型。

随机森林

- 在机器学习中，**随机森林**是一个包含多个决策树的分类器，并且其输出的类别是由个别树输出的类别的众数而定。Leo Breiman和Adele Cutler发展出推论出随机森林的算法。这个术语是1995年由贝尔实验室的Tin Kam Ho所提出的随机决策森林（random decision forests）而来的。这个方法则是结合 Breimans 的 "Bootstrap aggregating" 想法和 Ho 的 "random subspace method" 以建造决策树的集合。



随机森林学习算法

- 根据下列算法而建造每棵树^[1]：
 - 用 N 来表示训练用例（样本）的个数， M 表示特征数目。
 - 输入特征数目 m ，用于确定决策树上一个节点的决策结果；其中 m 应远小于 M 。
 - 从 N 个训练用例（样本）中以有放回抽样的方式，取样 N 次，形成一个训练集（即bootstrap取样），并用未抽到的用例（样本）作预测，评估其误差。
 - 对于每一个节点，随机选择 m 个特征，决策树上每个节点的决定都是基于这些特征确定的。根据这 m 个特征，计算其最佳的分裂方式。
 - 每棵树都会完整成长而不会剪枝，这有可能在建完一棵正常树状分类器后会被采用）。

卷积神经网络

- 随机森林网络泛化能力有限，如果需要将系统映射到一个更大的数据集上，可以选择将模型修改为卷积神经网络。
- 卷积神经网络可以将一帧的数据视为一张图片，对其进行卷积、池化、正则化等运算，最终通过全连接层输出这帧数据所属的类别。通过卷积操作强大的特征提取能力，可以在大数据集上得到高准确率。缺点是推断速度和训练速度会有所下降。
- 经过多年发展，卷积神经网络有包括resnet在内多种结构可选。我们使用经典的卷积神经网络结构，在控制模型大小的情况下达到了98%的识别率。

实时识别

- 使用ADS1293与Arduino采集数据
- 从串口读取1000个样本数据
- 滤波、分帧、插值
- 送入随机森林分类器分类
- 取出现次数最多的类别为识别结果



实时识别流程

实时识别

- 最终实现一个可通过命令行调用的实时识别程序，可选参数如下

名称	缺省值	含义
model_path	tree.model	机器学习模型文件名
signal_length	1000	每次识别采集的信号长度
sos_path	sos.csv	带通滤波器sos矩阵文件名
peak_lower_limit	1300	分帧的阈值
frame_length	169	每帧信号在插值后的长度

实验流程

- 人体连接贴片，三导联连接ADS1293
- 采集数据，通过波形观察窗口中观察信号是否正常采集
 - 贴片存在接触不良情况
 - ADS1293模拟前端存在虚焊现象
- 关闭波形观察串口，解除对串口的占用
- 运行实时识别程序，对比输出结果与样本来源

实验总结

- 在本次实验过程中，我们顺利实现了硬件搭建，软件算法实现的全流程，基本实现实验需要。在进行硬件搭建过程中，我们尝试过树莓派以及MCU，但都遇到了问题，我们及时更换为Arduino进行开发。在软件算法上，尝试了多种算法模型，最后使用了最适合我们实验数据大小的随机森林。本实验的不足在于数据量较少，以及由于ADS1293存在虚焊现象，采集到的数据质量较差。

分工情况

- 田野：数据处理、实时识别程序编写、报告编写
- 王润：Arduino使用、树莓派使用、ADS1293连线
- 张正泉：神经网络数据处理、训练，ppt制作