东软睿道

项目实训总结报告

项 目 名 称:云端医疗监护系统

学 校 名 称:北京理工大学

项 目 讲 师:姜侃

姓 名:梁瑛平

学 号:1120182525

实 训 日 期: 2019年9月7日-9月17日

**1 引言**

随着信息技术逐步渗透各行各业，云端医疗监护系统也逐渐成为了热门的课题。如何较好地完成云端医疗系统的每个版块功能成为了课题的核心。

1.1项目概述

本项目的诞生源于2020年暑期计算机实训课程。其背景是，云端医疗监控作为一门较为新兴的技术，在新冠疫情期间发挥了能够保障医护人员安全、提高监护效率的显著作用。因此通过设计该项目，一方面利用实训锻炼自身的应用层能力，一方面尝试为该领域提供一点素材，以期云端医疗监护系统能够早日在各医院广泛应用。本项目得到了姜侃老师所在企业提供的多个素材，包括但不限于端口、CS端、导入数据库端的多个DEMO及姜侃老师对课题的细致分析和深入讲解。

1.2 需求分析

正如前文所提到的，随着疫情期间社会各层面提高了对云端医疗监护领域的重视，多样化的需求也呈现在了设计师面前。因此，我们需要设计出一些符合特定监护需求的蕴端医疗监护系统。

新冠肺炎是本项目背景的一个切入点。每位住院患者都需要配备监护仪监控其心率等生理参数，而在类似的突发大型卫生事件下，患者数量远超平日，为医疗系统带来巨大压力。主要体现在以下两点：监护仪若由家属来监控，由于家属不是专业人员，存在对信息的误解进而影响治疗；若由医生来监控，每位患者都需配备医生，对医院的医生数量提出很高要求，并且效率很低。因此需求是将患者群体中大量分散、孤立的数据在云端集成为整体系统。一方面可以缓解医生的压力，远程监控数据方便快捷，且少量医生可以实时查看大量患者的数据；另一方面方便对数据做统计分析，有助于在长期改进医疗水平。

1.3 运行环境

软件环境：Windows，Python

硬件环境：电脑

**2 项目设计**（小三黑体，居中）（以个人主要负责内容为主）

2.1 设计思路

我们的设计思路从上述需求分析入手，即主要任务是集成分散患者的信息。第一步通过监护仪获取患者的面部数据，收集患者的眼部闭合程度和心率，并将每台监护仪的数据通过串口通信发送到网卡，即客服端，在客户端记录了患者的详细信息。接下来在网络连接模块中通过socket建立网络通信传输，将客户端数据发送到服务端。网络连接模块是我们工程的主体模块，客户端和服务端均设计了功能丰富、使用便捷的界面。服务端支持多线程，来自多个客户端的数据可以同时在主界面进行绘制和展示。最后，服务端的数据将被写入数据库中供存档和后续分析。

2.2 模块功能介绍

我主要负责UI界面和功能设计以及Socket通信模块。

2.2.1 客户端界面：

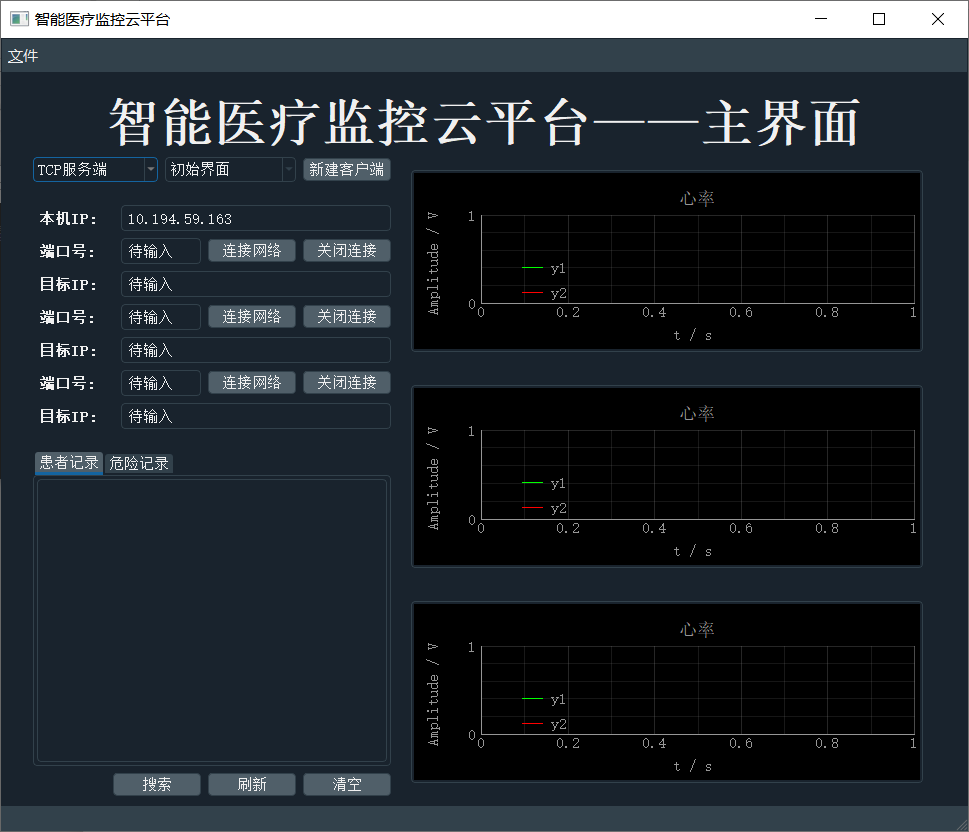
客户端（即Client）使用PyQt5进行界面布局和按键功能的开发。界面布局使用Qt Designer进行布局的大体设计，即首先将布局分为四部分：1.标题栏；2.网络连接区域；3.患者信息记录区域；4.患者实时视频模块。



其中标题栏使用QMenu和QAction实现了下拉框和功能响应，用于手动修改患者信息。网络连接区域使用QLineEdit和QLabel，提供连接服务器所用IP和端口的输入和接收功能，【连接网络】按键用于连接服务器，连接服务器后启动Socket将患者的基本信息和实时数据传输到主界面中。患者信息记录区域使用QTabWidget进行布局，用于在界面中记录患者的基本信息，点击【细节】按键可以在弹出窗口中查看患者的详细信息。患者实时视频模块通过连接设备摄像头获取实时的视频，通过QImage和QPixmap将视频进行解码和展示，通过Dlib库进行人脸检测和心率识别。

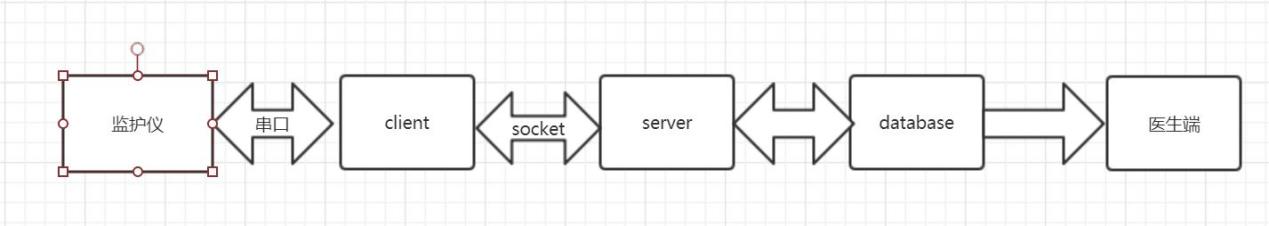
2.2.1 医生端界面：

客户端（即Sever）使用PyQt5进行界面布局和按键功能的开发。界面布局使用Qt Designer进行布局的大体设计，同样将布局分为四部分：1.标题栏；2.网络连接区域；3.患者信息记录区域；4.患者实时数据模块。

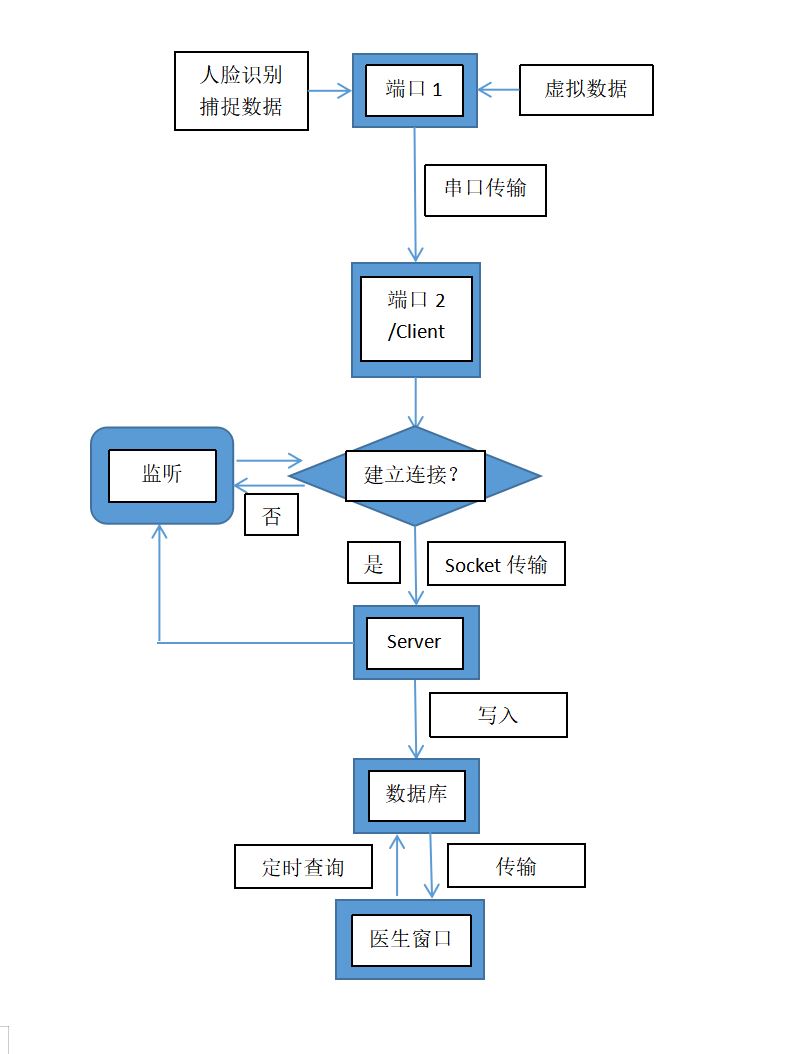


其中标题栏使用QMenu和QAction实现了下拉框和功能响应，用于手动修改患者信息。网络连接区域使用QLineEdit和QLabel，提供启用TCP服务所用IP和端口的输入和接收功能，【连接网络】按键用于启动TCP服务，启动后启动开启单独的线程实时监听客户端传入的患者的基本信息和实时数据并进行可视化的展示；【关闭连接】用于手动终止客户端的连接。患者信息记录区域使用QTabWidget进行布局，用于在界面中记录患者的基本信息，点击【细节】按键可以在弹出窗口中查看患者的详细信息。患者实时数据模块接收客户端传入的数据，通过PyQtGraph进行坐标系建立和波形绘制。

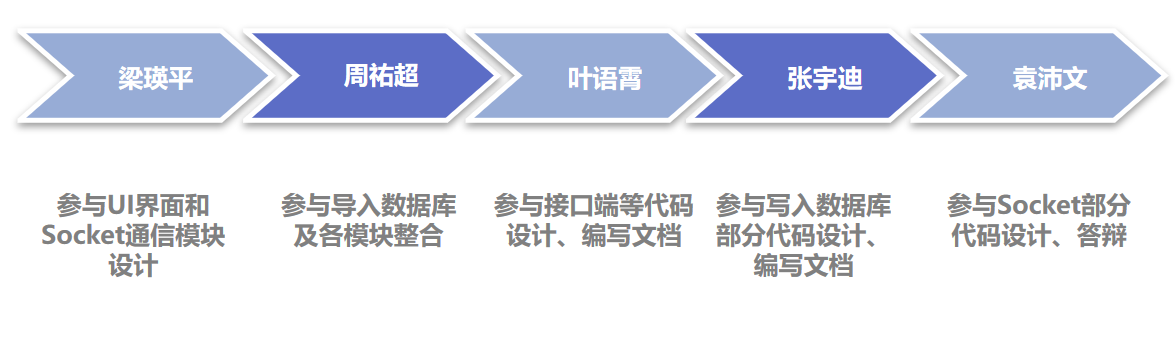
2.3 模块结构图



2.4 程序流程图



2.5 功能设计分工

**3 详细设计**

本项目中各个部分的关系和流程已在2.3及2.4中展示，因此这里只展示各个部分所涉及的技术和核心功能。具体如下：

1. 心率等指标的检测方法设计

在本部分，项目与计算机视觉中相关技术相结合。运用了基于人眼特征的心率检测和人眼闭合程度检测。意在用真实的检测方式取代假数据，在之后的步骤中体现实时性。在患者检测方面提供了两个指标：心率和闭眼程度。前者基于脉搏跳动时对变化的血液对光的吸收量的测量来实现，后者则是基于对人眼的识别技术实现。

1. 数据传输设计

本项目的数据传输设计分为两部分，串口传输及socket传输，以此模拟真实设备传输的情况。涉及的技术即为两者与pyqt的结合。

串口部分的数据传输功能将数据从检测设备发送至客户端。Socket部分的核心功能为将客户端的数据传输到服务器端。具体的实现流程包含server端监听，client端连接，server端接收并连接。本项目的多线程体现在此部分即为设置服务器使其可以同时与多个客户端相连。

1. 数据库的设计

数据库的设计基于本项目的多线程功能。在服务器接收到来自不同客户端的数据后将其写入数据库。数据库中设计二维表popvalue。其中三个属性分别为graph\_popvalues(心率)、eye\_popvalues(闭眼程度)、time\_popvalues(时间)。

第i个元组即为来自第i个线程的数据。便于绘制时对数据库的读取。所涉技术包含对sql语句和pyqt的结合。

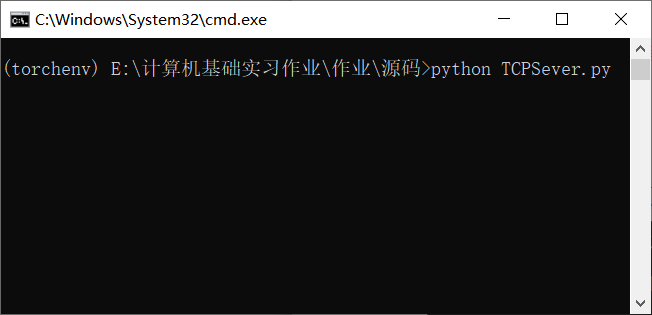
1. 界面的设计

界面的设计分为主界面和终端两个部分。两个部分的界面均按照位置分为三个区域，保证界面简洁清晰。左上区域显示网络连接信息：包括本机ip、目标ip及端口号。左下区域显示患者个人信息便于查询。右侧区域为波形显示区域，主界面支持并列显示多个患者的波形。

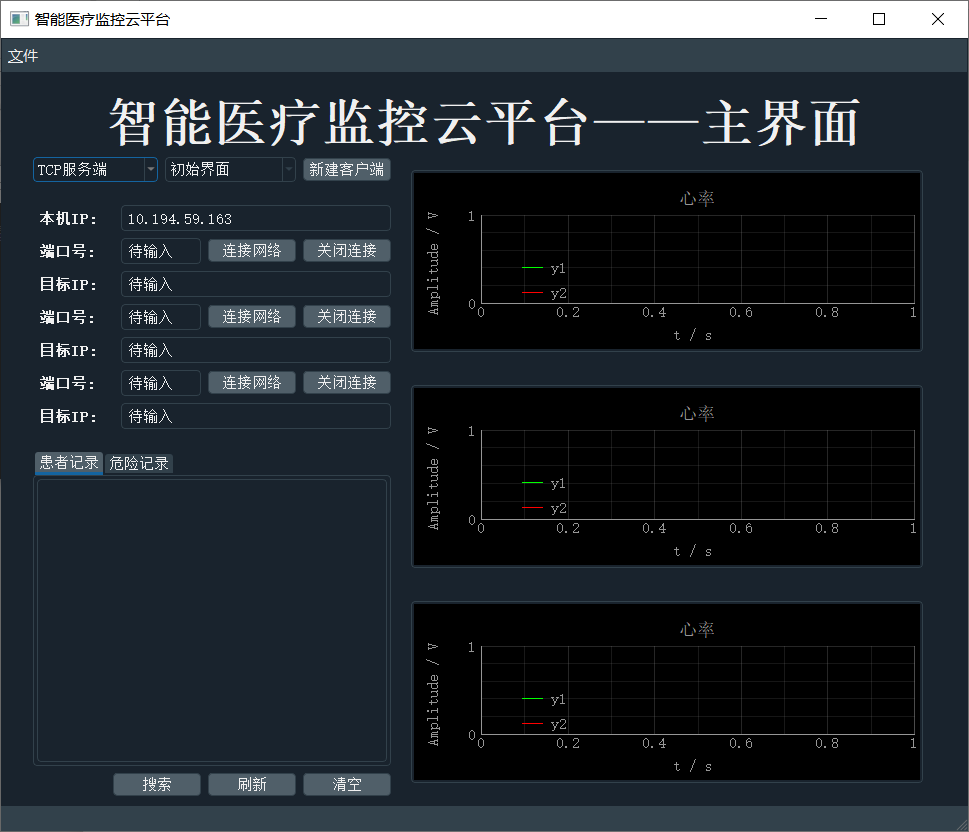
**4 实现方案**（小三黑体，居中）

4.1 启动Sever：

python TCPSever.py



运行后弹出Sever主界面

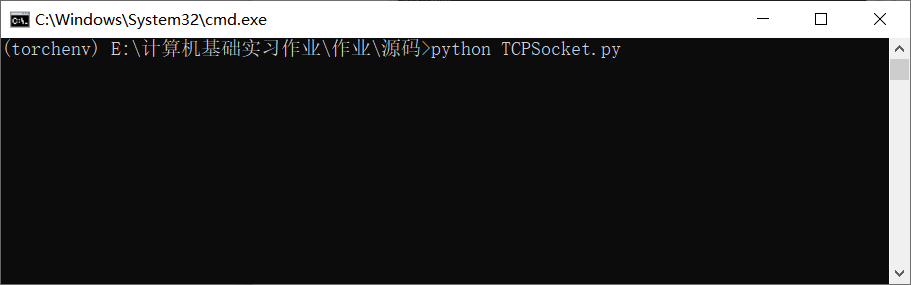


在左上侧区域输入端口号和目标IP，点击【连接网络】开启服务



4.1 启动Sever：

新建终端，输入：python TCPSocket.py



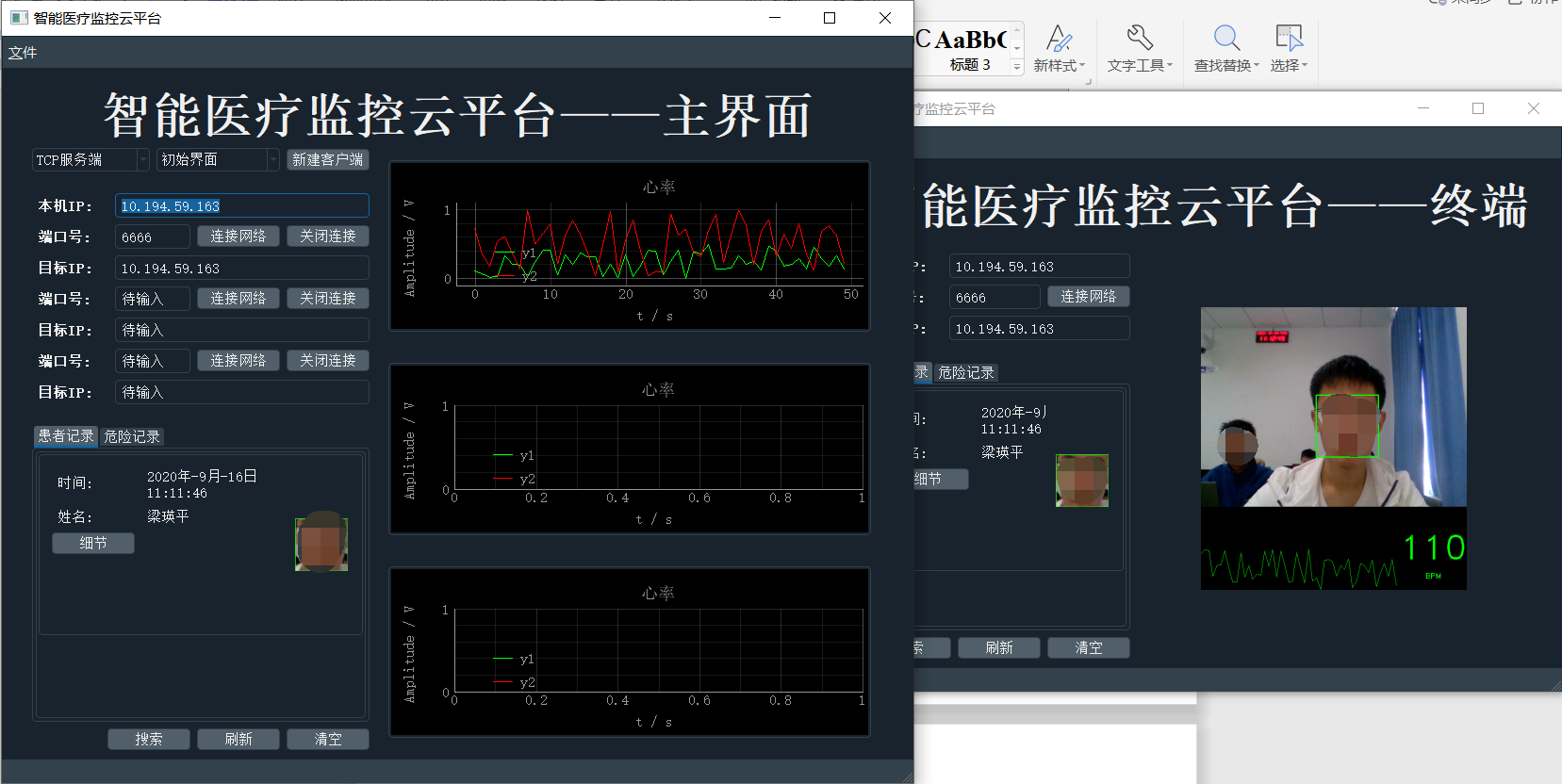
运行后弹出Socket主界面



在左上侧区域输入端口号和目标IP，点击【连接网络】开启通信



主界面接收消息并进行波形绘制



也可以开启多个Socket服务

****

**5 系统测试**（小三黑体，居中）



**6 心得体会**（小三黑体，居中）

算通过学习，我们对软件开发有了更深刻的认识，通过实战项目的演练我们锻炼了自己的编码能力、工程能力和团队沟通能力。