

# 天津大学

## 软件工程综合实践结课报告

### 基于红外热像传感器的WEB开发



学 院	智能与计算学部
专 业	软件工程
年 级	2019级
姓 名	刘格伦 郑睿恺 周子涵 樊振宇 李世军
学 号	3019244337 3019244342 3019244335 3019244338 3019244341

2021 年 9 月 9 日

# 目 录

第一章	引言	1
1.1	背景	1
1.2	开发目的	1
第二章	项目需求	2
2.1	硬件需求	2
2.2	服务器端需求	2
2.3	前端需求	2
2.4	系统运行环境	2
第三章	项目设计与实现	3
3.1	总体设计	3
3.2	硬件系统设计	3
3.3	后端服务器设计	4
3.4	前端页面设计	5
3.5	数据库设计	5
第四章	测试与开发过程	6
4.1	传播延时方面	6
4.2	WiFi连接方式	6
4.3	数据接收服务	7
4.4	插值算法的选择	7
4.5	温度区间色彩	7
4.6	图片的产生和输出	7
4.7	服务器的部署	8

4.8	前端页面的改进	8
第五章	成果展示	9
第六章	总结与展望	12
6.1	总结	12
6.2	展望	12
第七章	团队成员分工描述	13
第八章	异构计算的uid及实验截图	14

## 第一章 引言

### 1.1 背景

2019年12月以来，湖北省武汉市部分医院陆续发现了多例有华南海鲜市场暴露史的不明原因肺炎病例，现已证实为2019新型冠状病毒感染引起的急性呼吸道传染病。新型冠状病毒感染，初期症状可能和感冒差不多，可能有咳嗽，咳痰，发热，胸闷呼吸道症状。可能有腹痛，腹泻消化道症状。可能有乏力，四肢酸痛全身症状。可能有眼红，流泪眼部症状。因此，为了更好的防控防治，无论是个人及家庭，或是社区、车站、医院等地方都急需测温来筛查发热病人，因此测温仪需求量大增。

红外测温仪由光学系统、光电探测器、信号放大器及信号处理、显示输出等部分组成。光学系统汇聚其视场内的目标红外辐射能量，视场的大小由测温仪的光学零件及其位置确定。红外能量聚焦在光电探测器上并转变为相应的电信号。该信号经过放大器和信号处理电路，并按照仪器内定的算法和目标发射率校正后转变为被测目标的温度值。与传统接触式温度计相比，红外线测温仪有着响应时间快、非接触、准确测量、使用安全及使用寿命长等优点，结合成熟的软件技术，更能使其发挥传统测温仪难以实现的优越性能。

目前应用较为广泛的主要是单点式的非接触红外测温系统，虽然能满足一般需求，但由于没有一个对人体温度的全面感知，存在漏检问题。红外热像仪是红外测温仪的升级版本，它可以将人体表面的温度发布用彩色图像的形式输出到显示器或屏幕上，让我们可以直接“看见”温度，不同的色彩代表着不同的温度，温度高低一目了然。红外成像仪适用于人流量很大的公共场合的人群排查，在保证更高精度的同时，也提高了测温的效率。本课题提出使用MLX90640红外热像传感器的热像体温测量方案。

### 1.2 开发目的

本项目主要目标为：以MLX90640传感器和ESP32完成红外数据的采集，在Web端实时显示热成像与关键数据，同时可以对出入人员的信息进行登记与查询。适合放置于商场入口、公司大门等人流量多，对测温需求大的地点。

## 第二章 项目需求

### 2.1 硬件需求

- 实现ESP32和红外传感器的数据采集方案
- 实现利用ESP32的WiFi模块将采集得到的温度数据发送到服务器

### 2.2 服务器端需求

- 通过socket接收ESP32传输的数据
- 使用插值算法将32\*24的图片变成512\*384像素
- 实现红外温度到可视颜色的伪彩色编码转换
- 绘制热成像图片，提取关键信息
- 对数据库进行操作

### 2.3 前端需求

- 实时显示生成的热成像图片
- 实现登录、信息登记、信息查询等数据库操作
- 美观的界面

### 2.4 系统运行环境

- 开发工具: IntelliJ IDEA , Arduino
- 前端: Html + JSP + CSS
- 后端: Java + Servlet
- 服务器: 阿里云 + Tomcat

## 第三章 项目设计与实现

### 3.1 总体设计

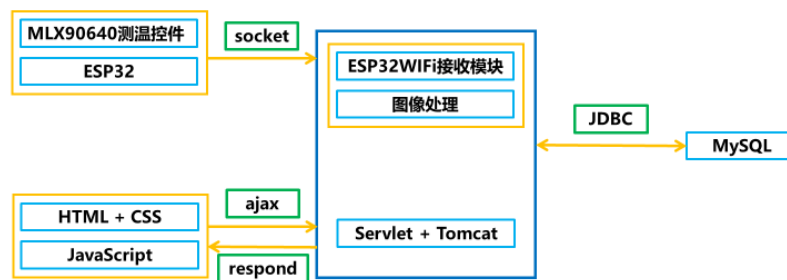


图 3-1 总体设计

### 3.2 硬件系统设计

硬件系统由 MLX90640 传感器和 ESP32 组成，开发环境为 arduino。

主要分为两个部分来实现，首先是对传感器的数据进行采集。这一部分通过在网上收集资料，按照说明调用相应的 API 接口，引入 *MLX90640\_API.cpp*，*MLX90640\_I2C\_driver.cpp* 中的函数来实现。

```

1 // 读取全部校正参数
2
3 int MLX90640_DumpEE( uint8_t slaveAddr , uint16_t *eeData )
4
5 // 解析校正参数为计算参数
6
7 int MLX90640_ExtractParameters( uint16_t * eeData ,
8     paramsMLX90640 *mlx90640 )
9
10 // 读取一帧计算用实时数据
11
12 int MLX90640_GetFrameData( uint8_t slaveAddr , uint16_t *
  
```

```

    frameData)
12
13 // 得到外壳温度
14
15 float MLX90640_GetTa(uint16_t *frameData, const
    paramsMLX90640 *params)
16
17 // 返回电压值VDD
18
19 float MLX90640_GetVdd(uint16_t *frameData, const
    paramsMLX90640 *params)
20
21 // 返回绝对温度
22
23 void MLX90640_CalculateTo(uint16_t *frameData, const
    paramsMLX90640 *params, float emissivity, float tr,
    float *result)

```

另一部分的内容就是使用WiFi将获取到的数据发送出去。这里用到了ESP32的WiFi库，使用WiFi.begin()方法，传入WiFi名称和密码就可以连接。

对于需要更换网络环境这一问题，改进了WiFi的连接。启动时首先尝试连接上一次连接的WiFi，在一定时间内如果没有连接成功，则转为热点模式，此时可以通过手机连接热点，访问ESP32开设的小服务器，利用事先编写好的html代码生成的简易页面，输入WiFi名称和密码，使其能够连接上新的WiFi。

发送数据用到的是WiFiClient，将得到的浮点数数组\*100转为整形，拼接为字符串后后调用WiFiClient.println()方法即可将其发送出去。

### 3.3 后端服务器设计

后端服务器主要由两部分构成。第一部分是利用Java的socket通信接收ESP32通过WiFi发送的数据，将其读入float数组以待后续使用。Insertdata类实现了二元和三元插值算法，将32\*24像素的图像扩大至512\*384像素并且满足图像及其轮廓符合人眼的生理观感。PixelConversion类实现了温度和可视颜色的伪彩编码转换，其中我们调整了温度在色条上的分布，使得其更加满足人体测温的需要，并且使得图像更加美观Draw类实现了图像的绘制。将图像转换为Base64编码，使其能够在Web上显示。

另一部分利用了Servlet来实现，经过TomCat处理后的前端发送的http请求，由Servlet进行接收，并发送携带图片base64编码的respond，从而能够将生成的图

片的Base64编码持续发送给前端接收，实现热成像图片的实时显示。

### 3.4 前端页面设计

编写html，实现web前端最基本的框架。编写的同时也调用css对其进行美化。再根据相关功能编写script中的函数以实现表单传递，发送请求等功能。script中主要使用了ajax来进行数据的收发。

### 3.5 数据库设计

表 3-1 admin表

变量	<i>id</i>	<i>name</i>	<i>account</i>	<i>password</i>	<i>tel</i>	<i>email</i>
类型	int	varchar	varchar	varchar	varchar	varchar

表 3-2 peoplelist表

变量	<i>id</i>	<i>address</i>	<i>name</i>	<i>tel</i>	<i>id<sub>n</sub>um</i>	<i>tem</i>	<i>date</i>
类型	int	varchar	varchar	varchar	varchar	varchar	date



## 第四章 测试与开发过程

### 4.1 传播延时方面

- 最初方案：对float数组元素逐个传输，传完一次数据约耗时2s。
- 原因是每传输一次数据都会调用一次Wifi模块，经过测试，调用WiFi模块是占用整个传输时间最大比例的，所以后续改进都从减少调用Wifi模块次数的角度考虑。
- 改进方案：将float数组拼接为字符串，传完一次数据约耗时0.6s。
- 该方案确实减少了Wifi模块调用时间从而大大减少了传输时间。但是在对数据进行类型转换的时候，由float型（4字节）转换为对应的string型（8字节）时，数据的字节数发生了增长，后续改进都针对数据字节规模进行了考虑。考虑尝试使用UDP进行传输，但是UDP丢包问题可能使得服务器在单位时间内接收的数据量更少，得不偿失。后又考虑把几次数据合为一个字符串进行传输，但是在网络协议中包的大小有着明确的限制，不论合成几次数据都会分为多个包进行发送，并且合成数据时也浪费了大量的时间，经过考虑后，放弃了UDP的传输方式。
- 最终方案：在ESP32上对float做乘100运算后类型转换为int，而后拼为字符串进行传输，传完一组数据约耗时0.3s。
- 这个方法降低了转换后字符串的长度，并且后续处理数据也更为简单方便。

### 4.2 WiFi连接方式

- 最初方案：使用ESP32开设热点即AP模式，接收设备连接进行数据接收。
- 但是由于ESP32本身没有网络环境，外部设备连接后会使得设备无法访问外部网络，最后放弃了这个方案。
- 改进方案：ESP32调用自身WiFi模块，利用事先输入好的Wifi名称和密码连接对应网络，外部设备连接同一个wifi网络，在局域网内进行数据传输。
- 适用范围提升了许多，但是每次更换网络环境，需要重新烧写ESP32，十分繁琐。
- 第二次改进：ESP32的AP模式可以使其作为服务器向外部连接的提供一个html网页来输入想要连接的网络名称和密码。
- 虽然适配了复杂多变的网络环境，但是每次断电，启动都需要输入一次环境配置，也不够人性化。

- 最终方案：保存上次连接的WiFi信息，启动后先尝试连接上次连接的WiFi，不成功再开启AP模式，用外部设备连接，输入新的网络配置。

### 4.3 数据接收服务

- 最初方案：使用Java编写的Service服务器，与ESP32建立Socket连接。
- 每次ESP32重新启动时，会递交新的Socket，但是之前的服务器仍然在等待上一次建立连接的Socket的输入流，只能重启服务器才能再次进行连接。
- 最终方案：建立两个Service对象交替使用，在一个服务器建立连接后，另一个服务器依然在监听Socket请求，当收到新的请求后，结束上一个服务器的线程，并且再次开启一个新的线程用于监听。

### 4.4 插值算法的选择

- 最初方案：简单的取平均值算法。
- 速度很快，计算压力较小，实现了图像分辨率的增加，但是计算的参照物少，精度不高，生成的图像比较模糊，细节缺失较多。
- 最终方案：二元和三元插值算法。
- 相较于之前的算法得出的图像锯齿感有所减少。尽管三元插值算法在图像细节上可能有较好的呈现，但是因为其运算过程复杂，可能导致数据从传输到最终显示在前端页面上有着较大的延迟，所以最后选择了耗时较少的二元插值算法，让图像质量和显示延时取得较好的平衡。

### 4.5 温度区间色彩

- 最初方案：简单的让温度在色彩条带上均匀分布。
- 由于温度跨度较大，而实际测温时的温度区间差值较小，使得人和环境的温度都落在色带上较为接近的位置，最终渲染出来的图像颜色较为接近，区分度不高。原本设想温度低于某一个阈值时就将其颜色置为纯色，使得背景颜色不影响主要测温物体颜色的呈现。但是由于测温物体边缘的温度有时与环境相差无几，细节缺失较多，最后呈现出来的图像也并没有达到预想的效果。
- 最终方案：根据实际情况，将人体体温附近的温度在色彩区间上占比增加，调整了其他温度区间的色彩区间占比，显著提升了成像的质量。

### 4.6 图片的产生和输出

- 最初方案：将经过处理后的数据封装入image对象中，在本地生成图片，再将本地生成的图片转为Base64编码。
- 但是图片刷新速率快，每次运行产生大量图片，给调试者带来了不必要

的麻烦。并且由于生产图片需要根据路径存放，也不易于服务器的移植。

- 最终方案：将生成图片封装的image对象直接转换为字节流的形式，再对字节流进行Base64编码的转换。
- 最终使得图片流式传输，不占用本地资源，同时也节省了图片生成及读取的时间，减少了前端页面的延迟。

#### 4.7 服务器的部署

- 最初方案：服务器部署于本机。
- 必须通过局域网才能访问，硬件部分每次重新启动服务器也需要重新启动。即使通过端口映射到学校网络实现连接学校网络即可访问，也并不是十分稳定。
- 最终方案：将服务器部署在阿里云ESC实例上，使其可以通过公网ip访问。
- 只要有网络环境和浏览器的设备即可访问。

#### 4.8 前端页面的改进

- 最初方案：在进行html编写时采用固定像素的方式布局页面。
- 在常见PC端可以正常展现画面，但是在其他移动设备上页面效果非常僵化，页面布局混乱。
- 最终方案：进行整个页面的重新设计，设定html网页布局，分配好容器，元素大小使用容器相对值进行设定。
- 使得Web适配了PC端、移动端，在不同的设备上使用也能获得较为良好的使用体验，真正实现了多端访问的需求。

## 第五章 成果展示

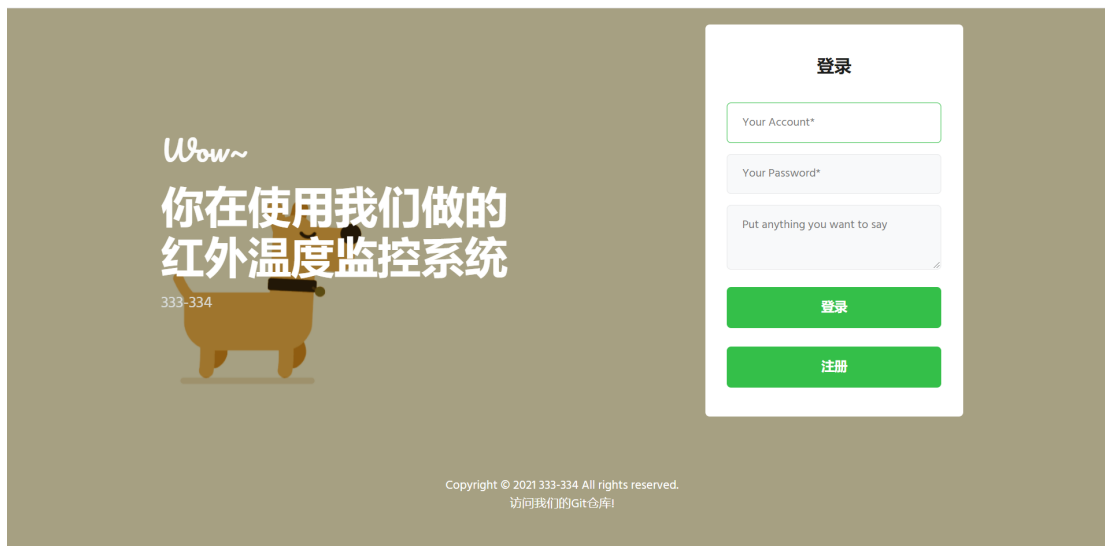


图 5-1 登录页面

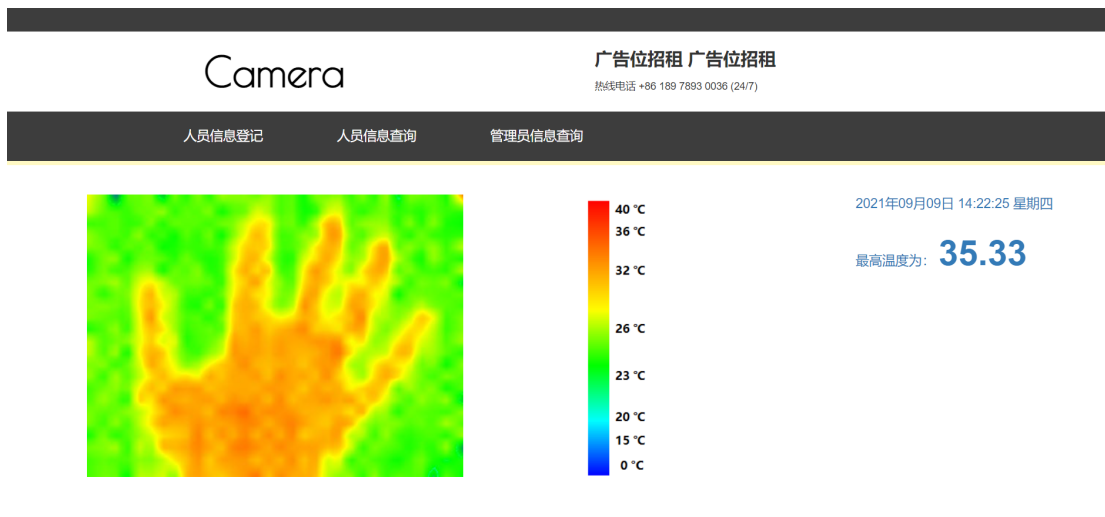


图 5-2 主界面

### 查询结果

ID:1 ,姓名:刘格伦 ,电话:18978930036 ,账号:Liugelun ,电话18978930036 ,邮箱lgl@tju.edu.cn

---

ID:2 ,姓名:郑睿恺 ,电话:13600807587 ,账号:Zzrk ,电话13600807587 ,邮箱Moranzrk@tju.edu.cn

---

图 5-3 管理员信息查询

Camera

广告位招租 广告位招租  
热线电话 +86 189 7893 0036 (24/7)

人员信息登记 人员信息查询 管理员信息查询

#### 请输入该用户的基本信息

家庭住址	姓名	联系方式
<input type="text" value="天津大学北洋园校区梅园一街"/>	<input type="text" value="樊振宇"/>	<input type="text" value="13333333333"/>
身份证件	实时体温	日期
<input type="text" value="000000000000000000"/>	<input type="text" value="36.52"/>	<input type="text" value="2021/9/9"/>

提交

About Us

Tianjin, Tianjin University  
Software Engineering Class 4  
333-334  
Tel 见招租电话  
Git <https://github.com/Caffrey-Liu/333-334>

Stay in Touch

Subscribe

Camera




图 5-4 用户信息登记



图 5-5 用户信息查询

## 第六章 总结与展望

### 6.1 总结

在本次实践中，我们实现了基于MLX90640传感器的Web平台开发，实现了基本的功能。对本次实践的全过程进行一个总结如下：

- 由于缺乏项目的开发经验，起初进行开发时不知道从何入手。开始时没有明确的方向，各个模块都有所进展，却没有解决根本的前后端如何通信这样的问题。因此在前后端整合上花费了大量的时间。
- 通过这一次的开发过程，理解了如计算机网络，数据库这一类基础课程的重要性。在学习课本上理论知识的同时，将这些知识结合编程语言，结合到实际的项目中去，是学习这些课程必须要做到的。
- 总的来说，这次实践串联起之前所学的理论知识，并且让我们自己了解了许多实际项目中需要考虑的问题，收获颇丰。

### 6.2 展望

本次实践基本实现了项目的需求，但目前还不够完善，存在可以拓展的功能，有以下几个想法：

- 超过一定温度时，发出警报，如弹窗，发出声音等。
- 增加一个摄像头，进行人脸识别，实现来往人员信息的自动登记。

## 第七章 团队成员分工描述

### 刘格伦

- 实现红外温度到可视颜色的伪彩色编码转换
- 建立服务器，对收到的数据进行处理
- 实现热成像图的绘制
- 实现前后端的通信
- 前端页面的美化

### 郑睿恺

- 实现ESP32和红外传感器的数据采集方案
- 实现和改进ESP32的WiFi通信
- 实现二元和三元插值算法
- LaTeX文档的编写

### 樊振宇

- 数据库的建立，sql语句的编写
- 协助完成LaTeX文档的编写

### 周子涵

- 协助完成前端页面
- 协助完成LaTeX文档的编写

### 李世军

- 英文文献的翻译
- 协助完成LaTeX文档的编写



## 第八章 异构计算的uid及实验截图

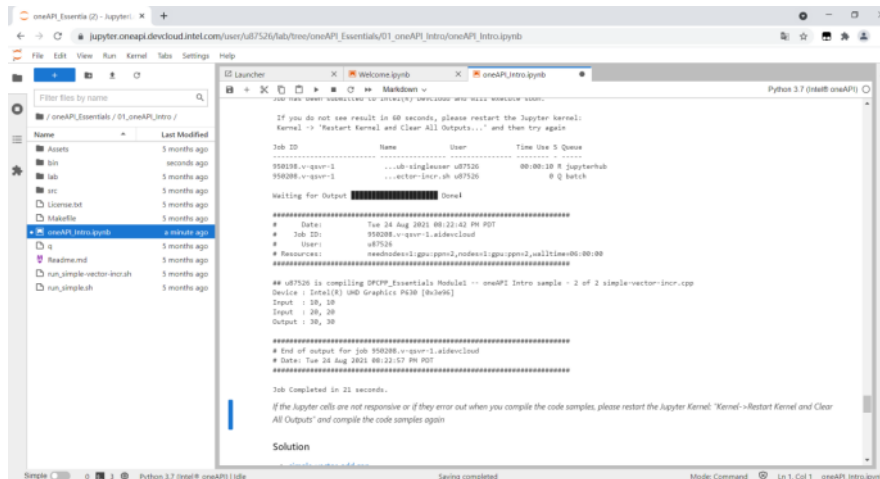


图 8-1 郑睿恺截图uid=u87526

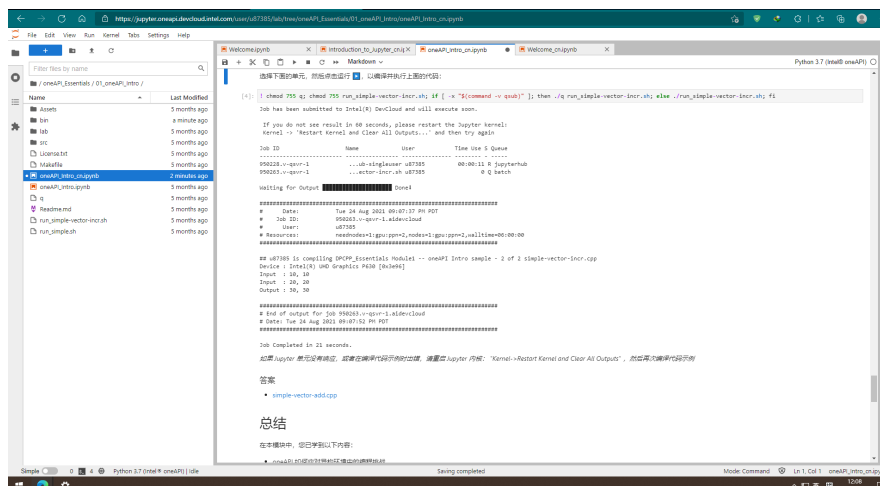


图 8-2 刘格伦截图uid=u87385

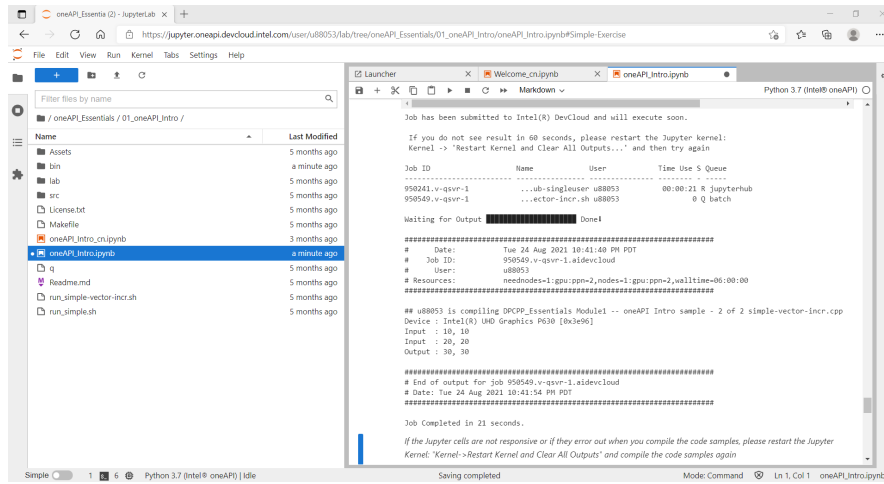


图 8-3 樊振宇截图uid=u88053

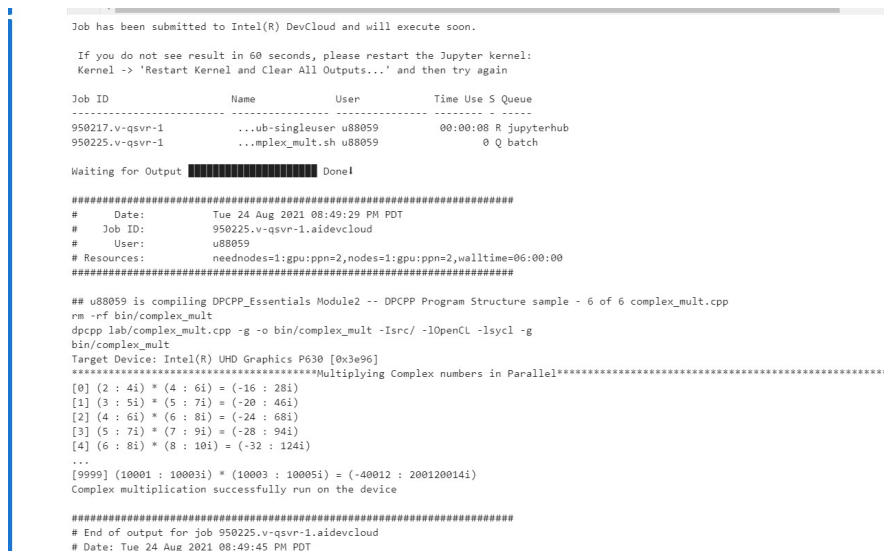


图 8-4 李世军截图uid=u88059

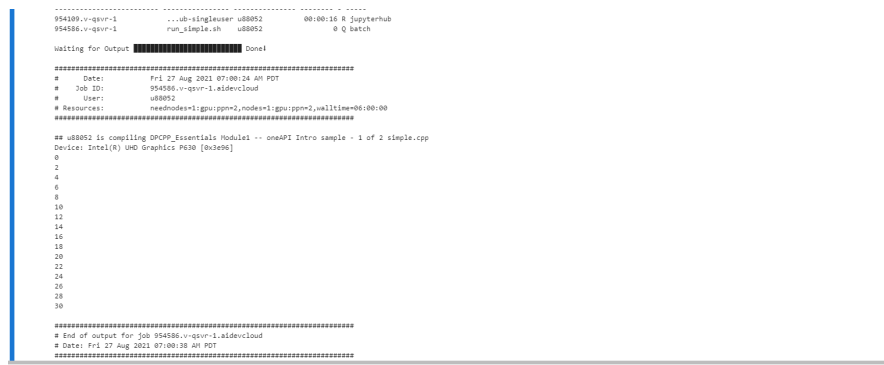


图 8-5 周子涵截图uid=88052