****

五邑大学实验报告

**网络编程课程**

**2023~2024年度 第1学期**

**题目：基于TCP-SOCKET编程的智能CPU监控硬件**

**院系： 智能制造学部**

**班级： 230733**

**学号： 3223000181**

**姓名： 洪培煜**

**目录**

[1．概论 3](#_Toc148562854)

[2．硬件设计 3](#_Toc148562855)

[3．软件设计 4](#_Toc148562856)

[2.0 Wifi配网 4](#_Toc148562857)

[2.1 天气感知 5](#_Toc148562858)

[2.2 CPU检测 6](#_Toc148562859)

[2.3 模拟蓝牙键盘控制 9](#_Toc148562860)

[4．运行与测试 10](#_Toc148562861)

[4.1 项目地址 10](#_Toc148562862)

[4.2 视频演示 10](#_Toc148562863)

[5．总结与心得 11](#_Toc148562864)

# 1．概论

·本项目是基于ESP32C3的软硬件结合的项目，单片机的控制程序和上位机的TCP 传输数据程序，相比之下,上位机的编写是较为简单的，做一个关于显示电脑参数的硬件，可以通过TCP传回显示CPU的运行速度，可以通过Wifi获取到Http上的天气数据，也可以把当成蓝牙键盘来进行操作电脑。

图示

描述已自动生成

# 2．硬件设计

硬件上有ESP32C3为主控IC，是一颗wifi芯片，可以为后续TCP 上的数据传输提供了可能，和分别240x320和128x160的屏幕组成,硬件成本要不到40元。在PCB设计通过排针和排母，和核心板主控连接。



# 3．软件设计

3.0 Wifi配网

自动配网功能是通过WiFi设置是储存在它的闪存系统中的。因此在启动ESP8266并连接WiFi时，它都会尝试使用闪存系统中储存的信息来进行WiFi连接。这个避免是我们在连接WiFi的时候，不同的环境去连接WiFi，需要重新烧录代码的情况， 通过TCP协议对主控IC连接WiFi名字和密码进行修改，再次重新连接。

3.1 天气感知

天气感知是通过是调用心知天气的Http接口，返回的Json数据进行解析的，得到准确的温度信息和城市。然后通过单片机对传过来的数据进行处理，并显示在LCD屏上。

桌子上放了游戏机

描述已自动生成

3.2 CPU检测

电脑性能数据的传输，先由单片机对电脑热点的建立连接，后由Python编写的上位机的程序，通过同一局域网下连接单片机的IP地址和端口来进行连接，传输电脑CPU，GPU的占用率给单片机进行显示。

图片包含 图形用户界面

描述已自动生成

Python 上位机的调用 psutil ，socket , PyQt5, FluentUI来进行编写，通过psutil来读取电脑CPU 的性能，然后通过socket来进行对单片机来通信，FluentUI是主要来界面美化UI的界面，可以通过简单的调用就可以调出想要的效果。

**def** **\_\_init\_\_**(self, text: str, parent=None):

super().\_\_init\_\_(parent=parent)

self.label = QLabel(text, self)

self.hBoxLayout = QHBoxLayout(self)

self.label.setAlignment(Qt.AlignCenter)

self.hBoxLayout.addWidget(self.label, 1, Qt.AlignCenter)

self.setObjectName(text.replace(' ', '-'))

self.IPAddress = PushButton('IP地址', self)

self.IPAddress.move(120, 40)

self.IPAddress.resize(100, 35)

self.IPAddressLineEdit = LineEdit(self)

self.IPAddressLineEdit.move(250, 40)

self.IPAddressLineEdit.resize(200, 2500)

self.IPAddressLineEdit.setClearButtonEnabled(True)

self.setObjectName(text.replace(' ', '-'))

self.PortName = PushButton('端口', self)

self.PortName.move(120, 100)

self.PortName.resize(100, 35)

self.PortNameLineEdit = LineEdit(self)

self.PortNameLineEdit.move(250, 100)

self.PortNameLineEdit.resize(200, 2500)

self.PortNameLineEdit.setClearButtonEnabled(True)

self.TCPconnectSuccessButton = PushButton('连接', self)

self.TCPconnectSuccessButton.clicked.connect(self.TCPconnectSuccess)

self.TCPconnectSuccessButton.move(250, 200)

self.TCPconnectSuccessButton.resize(200, 60)

self.loginButton = PushButton('断开', self)

self.loginButton.clicked.connect(self.TCPconnectSuccess)

self.loginButton.move(250, 300)

self.loginButton.resize(200, 60)

**def** **TCPconnectSuccess**(self):

**print**(self.IPAddressLineEdit.text())

InfoBar.success(

title='TCP',

content="TCP通道已建立",

orient=Qt.Horizontal,

isClosable=True,

position=InfoBarPosition.TOP,

duration=2000,

parent=self

)

# 打印内存占用率

**print**(f"内存占用率: {memory\_usage.percent}%")

# 获取CPU使用率

cpu\_usage = psutil.cpu\_percent(interval=1) # 可以指定时间间隔

**print**(f"CPU使用率: {cpu\_usage}%")

total\_memory\_gb = round(memory\_usage.total / (2 \*\* 30), 2)

used\_memory\_gb = round(memory\_usage.used / (2 \*\* 30), 2)

free\_memory\_gb = round(memory\_usage.free / (2 \*\* 30), 2)

percent\_memory = round(memory\_usage.percent, 2)

**print**(f"GPU总内存: {total\_memory\_gb} GB")

**print**(f"GPU已使用: {used\_memory\_gb} GB")

**print**(f"GPU可用内存: {free\_memory\_gb} GB")

**print**(f"GPU使用率: {percent\_memory}%")

message = "GPU使用率: " + str(percent\_memory) + "%"

**print**(message)

# 1 创建客户端套接字对象tcp\_client\_1

# 参数介绍：AF\_INET 代表IPV4类型, SOCK\_STREAM代表tcp传输协议类型 ,注：AF\_INET6代表IPV6

tcp\_client\_1 = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

# 2 通过客户端套接字的connect方法与服务器套接字建立连接

# 参数介绍：前面的ip地址代表服务器的ip地址，后面的61234代表服务端的端口号 。

tcp\_client\_1.connect(("192.168.137.249", 2333))

# tcp\_client\_1.connect((self.IPAddressLineEdit.text(), self.PortNameLineEdit.text()))

# 将编号好的数据存到变量send\_data中，注：encode(encoding='utf-8)是将数据转换成utf-8的格式发送给服务器

send\_data = ("CPU use:" + str(cpu\_usage) + "%\n"+"GPU use"+ str(percent\_memory) + "%\n\r").encode(encoding='utf-8')

# send\_data = ("GPU\r").encode(encoding='utf-8')

# 3 通过客户端套接字的send方法将数据发送给服务器

tcp\_client\_1.send(send\_data)

# 4 通过客户端套接字的recv方法来接受服务器返回的数据存到变量recv\_data中，1024是可接收的最大字节数。

recv\_data = tcp\_client\_1.recv(1024)

# 将接收到的服务器数据recv\_data通过decode方法解码为utf-8

**print**(recv\_data.decode(encoding='utf-8'))

# # 5 最后关闭客户端套接字连接

tcp\_client\_1.close()

图形用户界面, 应用程序

描述已自动生成

3.3 模拟蓝牙键盘控制

蓝牙键盘控制是通过单片机上的上下左右来进行控制电脑，往上面是提高音量，向下是减低音量，向中间按是禁音，向右是打开浏览器搜索，向左是打开文件管理器。

电脑显示屏

中度可信度描述已自动生成

# 4．运行与测试

## 4.1 项目地址

<https://gitee.com/its-a-slap/Monitoring-Hardware>

<https://github.com/Yu-1120/Monitoring-Hardware>

## 4.2 视频演示

[五邑大学-基于TCP的智能电脑性能监控硬件\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1KH4y1d7iv/?spm_id_from=333.1296.top_right_bar_window_history.content.click)

# 5．总结与心得

这个项目是我一直想做的，但又没啥时间，趁着这次大作业，随便做了，项目不难，只是把之前学过的东西整合到一起去了。意义上之前的项目的积累，才有了这个项目。

这个项目的第一个缺点是主控芯片外置Flash 只有4M-Flash，需要我们在代码上进行优化。尽可能压缩代码编译后体积变小。第二个硬件是连接主要是通过排针排母连接的方式无法把体积做到最小。