四川大學

SICHUAN UNIVERSITY

计算机网络与分布式系统 开发过程文档

题　　目	基于 TCP 和 UDP 的实时通讯系统
学生姓名	林世龙
课　　程	计算机网络与分布式系统
学　　号	2018141231089
专　　业	力学软件工程交叉实验班
指导教师	宋万忠

1. 说明

1.1 项目名称

基于 TCP 和 UDP 的实时通信系统

1.2 项目需求

1） 系统要正确响应用户发出的消息，将其上传到服务器，分发给客户端。
2） 通讯系统至少可以满足三个客户端同时登录、通讯的需求。
3） 用户可以对即时通讯系统的工作端口以及 ip 进行设置。

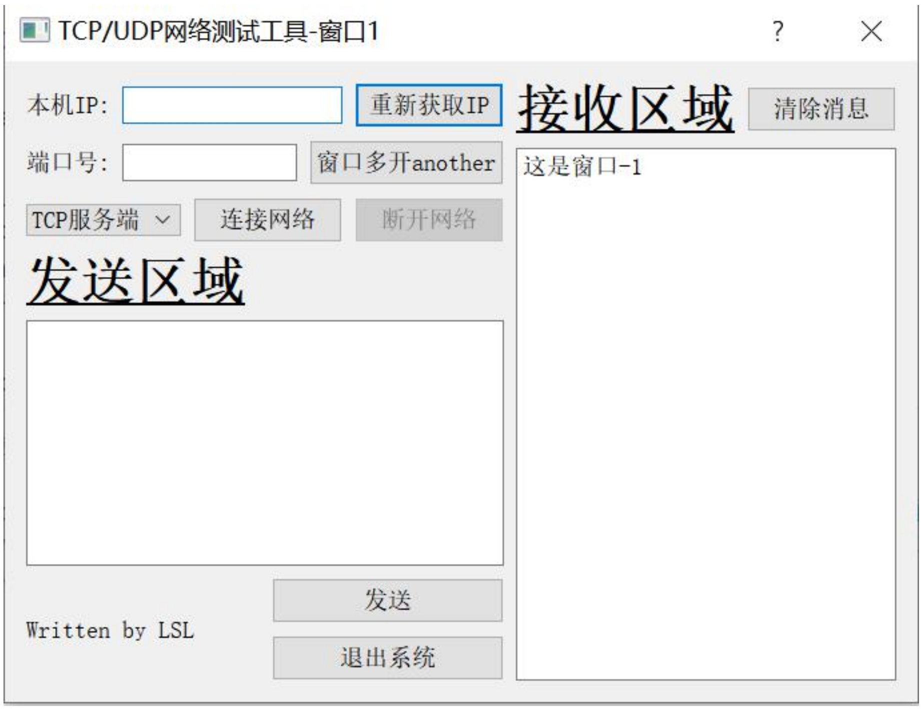
1.3 开发环境

操作系统：Windows 10
开发工具：VScode
Python 版本：3.6
实现语言：python

2. 项目实现

2.1. 界面设计

本项目是使用的 PYQT5 来进行的图形化界面设计，设计如图所示



这是窗口-1

Written by LSL

具体代码实现

```python
def ui_translate(self):
    self.setWindowTitle(self._translate("TCP-UDP", "TCP/UDP网络测试工具-窗口%s" % self.num))
    self.comboBox_tcp.setItemText(0, self._translate("TCP-UDP", "TCP服务端"))
    self.comboBox_tcp.setItemText(1, self._translate("TCP-UDP", "TCP客户端"))
    self.comboBox_tcp.setItemText(2, self._translate("TCP-UDP", "UDP服务端"))
    self.comboBox_tcp.setItemText(3, self._translate("TCP-UDP", "UDP客户端"))
    self.pushButton_link.setText(self._translate("TCP-UDP", "连接网络"))
    self.pushButton_unlink.setText(self._translate("TCP-UDP", "断开网络"))
    self.pushButton_get_ip.setText(self._translate("TCP-UDP", "重新获取IP"))
    self.pushButton_clear.setText(self._translate("TCP-UDP", "清除消息"))
    self.pushButton_send.setText(self._translate("TCP-UDP", "发送"))
    self.pushButton_exit.setText(self._translate("TCP-UDP", "退出系统"))
    self.pushButton_else.setText(self._translate("TCP-UDP", "窗口多开another"))
    self.label_ip.setText(self._translate("TCP-UDP", "本机IP:"))
    self.label_port.setText(self._translate("TCP-UDP", "端口号:"))
    self.label_sendto.setText(self._translate("TCP-UDP", "目标IP:"))
    self.label_rev.setText(self._translate("TCP-UDP", "接收区域"))
    self.label_send.setText(self._translate("TCP-UDP", "发送区域"))
    self.label_written.setText(self._translate("TCP-UDP", "Written by LSL"))
```

2.2. UDP 通讯的实现

主要是由五个函数来实现的 UDP 通讯的功能。一个是 UDP 服务器的开启函数 UDP_Server_start()。一个是 UDP 客户端的开启函数 UDP_Client_start()。

```python
def udp_server_start(self):
    self.udp_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    try:
        port = int(self.lineEdit_port.text())
        address = ('', port)
        self.udp_socket.bind(address)
    except Exception as ret:
        msg = '请检查端口号\n'
        self.signal_write_msg.emit(msg)
    else:
        self.sever_th = threading.Thread(target=self.udp_server_concurrency)
        self.sever_th.start()
        msg = 'UDP服务端正在监听端口:{}\n'.format(port)
        self.signal_write_msg.emit(msg)
```

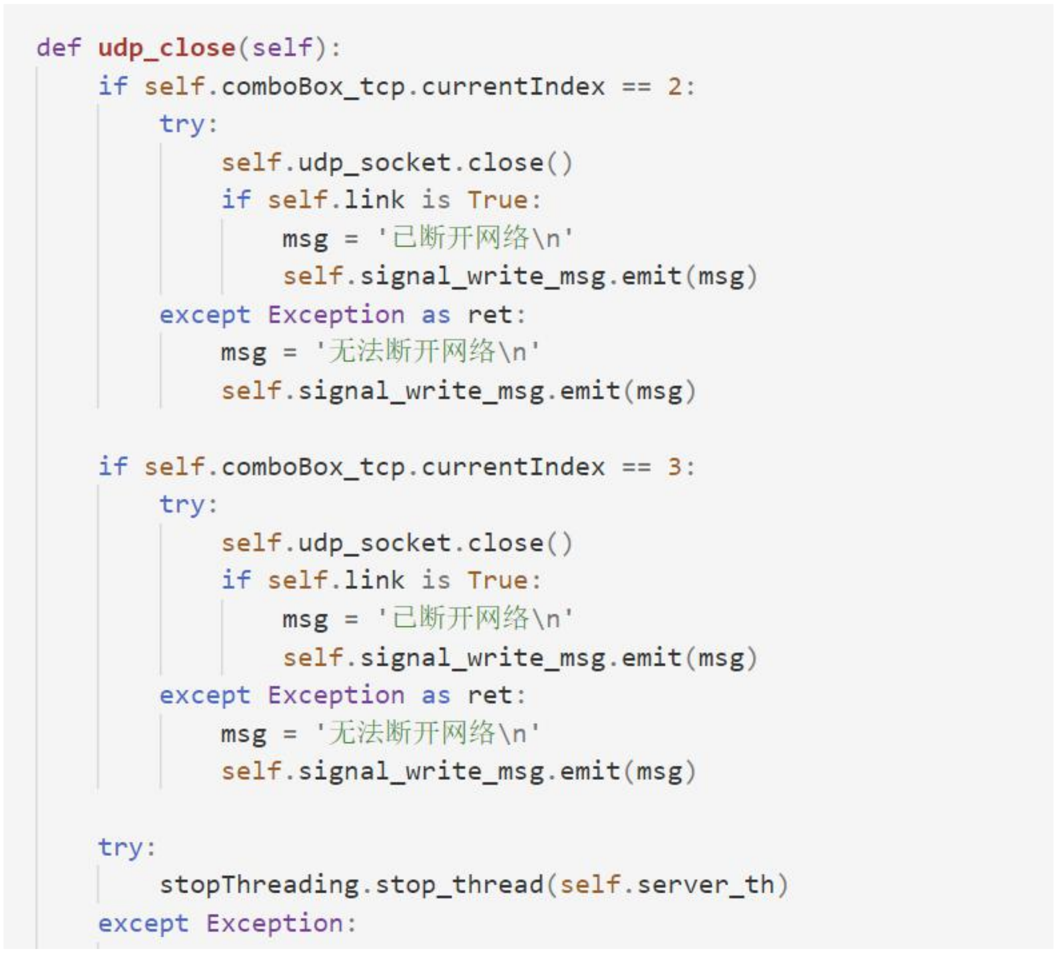
```python
def udp_client_start(self):
    self.udp_socket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
    try:
        self.address = (str(self.lineEdit_ip_send.text()),(int(self.lineEdit_port.text())))
    except Exception as ret:
        msg = '请检查目标IP，目标端口号\n'
        self.signal_write_msg.emit(msg)
    else:
        msg = 'UDP客户端已启动\n'
        self.signal_write_msg.emit(msg)
```

由 udp_server_concurrency(self)。这个函数来实现 CLient 的多线程操作

```python
def udp_server_concurrency(self):
    while True:
        recv_msg, recv_addr = self.udp_socket.recvfrom(1024)
        msg = recv_msg.decode('utf-8')
        msg = '来自IP:{}端口:{}:\n{}\n'.format(recv_addr[0],recv_addr[1],msg)
        self.signal_write_msg.emit(msg)
```

最后是 UDP 信息的发送和图形界面的关闭

```python
def udp_send(self):
    if self.link is False:
        msg = '请选择服务，并点击连接网络\n'
        self.signal_write_msg.emit(msg)
    else:
        send_msg = (str(self.textEdit_send.toPlainText())).encode('utf-8')
        if self.comboBox_tcp.currentIndex() == 2:
            msg = 'UDP服务端无法发送，请切换为UDP客户端\n'
            self.signal_write_msg.emit(msg)
        if self.comboBox_tcp.currentIndex() == 3:
            self.udp_socket.sendto(send_msg,self.address)
            msg = 'UDP客户端已发送\n'
            self.signal_write_msg.emit(msg)
```

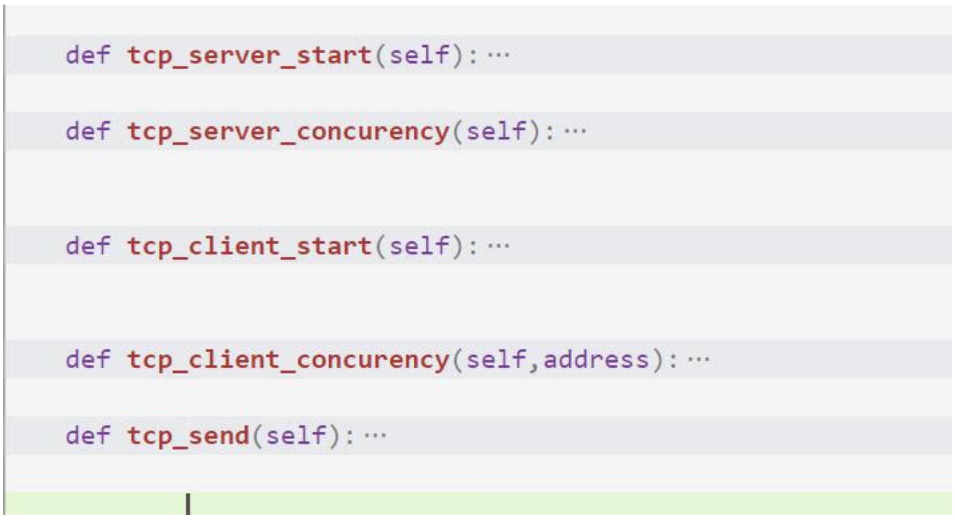


```python
def udp_close(self):
    if self.comboBox_tcp.currentIndex == 2:
        try:
            self.udp_socket.close()
            if self.link is True:
                msg = '已断开网络\n'
                self.signal_write_msg.emit(msg)
        except Exception as ret:
            msg = '无法断开网络\n'
            self.signal_write_msg.emit(msg)

    if self.comboBox_tcp.currentIndex == 3:
        try:
            self.udp_socket.close()
            if self.link is True:
                msg = '已断开网络\n'
                self.signal_write_msg.emit(msg)
        except Exception as ret:
            msg = '无法断开网络\n'
            self.signal_write_msg.emit(msg)

    try:
        stopThreading.stop_thread(self.server_th)
    except Exception:
```

2.3. TCP 通讯的实现



```python
def tcp_server_start(self): ⋯

def tcp_server_concurency(self): ⋯


def tcp_client_start(self): ⋯


def tcp_client_concurency(self,address): ⋯

def tcp_send(self): ⋯
def tcp_server_start(self):
    self.tcp_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    self.tcp_socket.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR,1)
    self.tcp_socket.setblocking(False)
    try:
        port = int(self.lineEdit_port.text())
        self.tcp_socket.bind(('',port))
    except Exception as ret:
        msg = '请检查端口号\n'
        self.signal_write_msg.emit(msg)
    else:
        self.tcp_socket.listen()
        self.server_th = threading.Thread(target = self.tcp_server_concurency)
        self.server_th.start()
        msg = 'TCP服务正在监听端口:%s\n' % str(port)
        self.signal_write_msg.emit(msg)
```

```python
def tcp_server_concurency(self):
    while True:
        try:
            client_socket,client_address = self.tcp_socket.accept()
        except Exception as ret:
            pass
        else:
            client_socket.setblocking(False)
            self.client_socket_list.append((client_socket,client_address))
            msg = 'TCP服务端已连接IP:%s端口:%s\n' % client_address
            self.signal_write_msg.emit(msg)
        for client,address in self.client_socket_list:
            try:
                recv_msg = client.recv(1024)
            except Exception as ret:
                pass
            else:
                if recv_msg:
                    msg = recv_msg.decode('utf-8')
                    msg = '来自IP:{}端口:{}:\n{}\n'.format(address[0],address[1],msg)
                    self.signal_write_msg.emit(msg)
                else:
                    self.tcp_socket.close()
                    self.reset()
                    mag = '从服务器断开连接\n'
                    self.signal_write_msg.emit(msg)
                    break
```

```python
def tcp_client_start(self):
    self.tcp_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    try:
        address = (str(self.lineEdit_ip_send.text()),int(self.lineEdit_port.text()))
    except Exception as ret:
        msg = '请检查目标IP，目标端口\n'
        self.signal_write_msg.emit(msg)
    else:
        try:
            msg = '正在连接目标服务器\n'
            self.signal_write_msg.emit(msg)
            self.tcp_socket.connect(address)
        except Exception as ret:
            msg = '无法连接服务器\n'
            self.signal_write_msg.emit(msg)
        else:
            self.client_th = threading.Thread(target=self.tcp_client_concurency,args=(address,))
            self.client_th.start()
            msg = 'TCP客户端已连接IP:{}%s端口:%s\n' % address
            self.signal_write_msg.emit(msg)
```

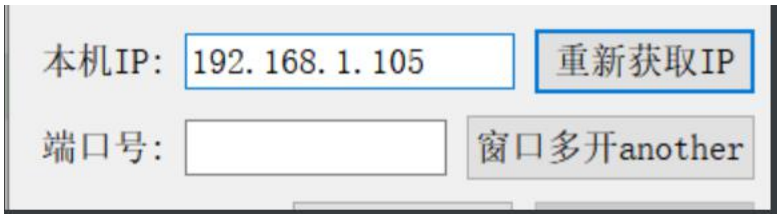
```python
def tcp_client_concurency(self,address):
    while True:
        recv_msg = self.tcp_socket.recv(1024)
        if recv_msg:
            msg = recv_msg.decode('utf-8')
            msg = '来自IP:{}端口:{}:\n{}\n'.format(address[0],address[1],msg)
            self.signal_write_msg.emit(msg)
        else:
            self.tcp_socket.close()
            self.reset()
            msg = '从服务器断开连接\n'
            self.signal_write_msg.emit(msg)
            break
```

```python
def tcp_send(self):
    if self.link is False:
        msg = '请选择服务，并点击连接网络\n'
        self.signal_write_msg.emit(msg)
    else:
        try:
            send_msg = (str(self.textEdit_send.toPlainText())).encode('utf-8')
            if self.comboBox_tcp.currentIndex() == 0:
                for client,address in self.client_socket_list:
                    client.send(send_msg)
                msg = 'TCP服务端已发送\n'
                self.signal_write_msg.emit(msg)
            if self.comboBox_tcp.currentIndex == 1:
                self.tcp_socket.send(send_msg)
                msg = 'TCP客户端已发送\n'
                self.signal_write_msg.emit(msg)
        except Exception as ret:
            msg = '发送失败\n'
            self.signal_write_msg.emit(msg)
```

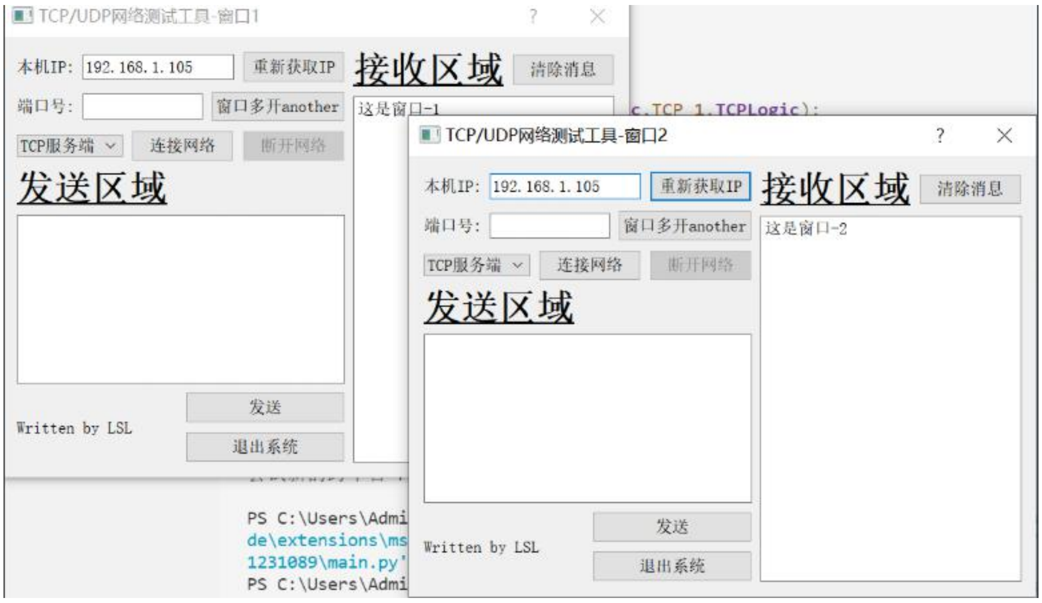
3. 项目展示

一. TCP 实现的功能

1. 获取本机的 IP，若是选择服务器可以设置端口号

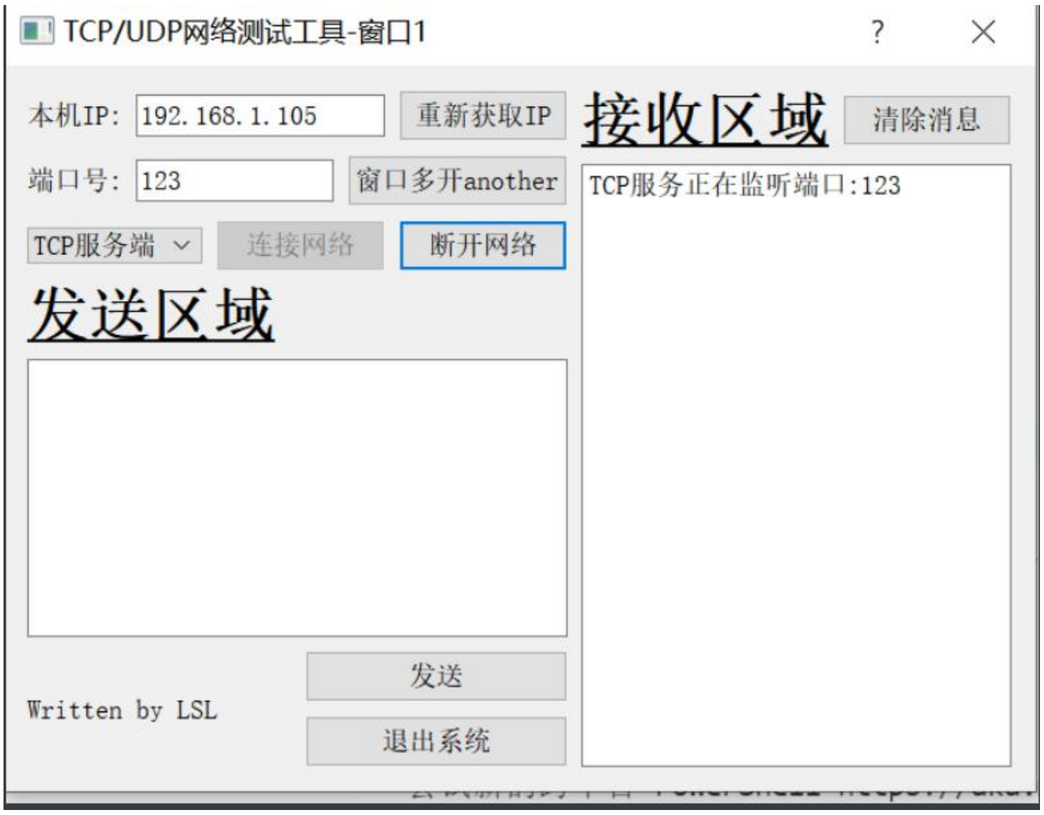


2. 窗口多开功能，可以在一台主机上开启多个窗口。点击窗口上的多开窗口 another，点击 YES，即开开启新的窗口



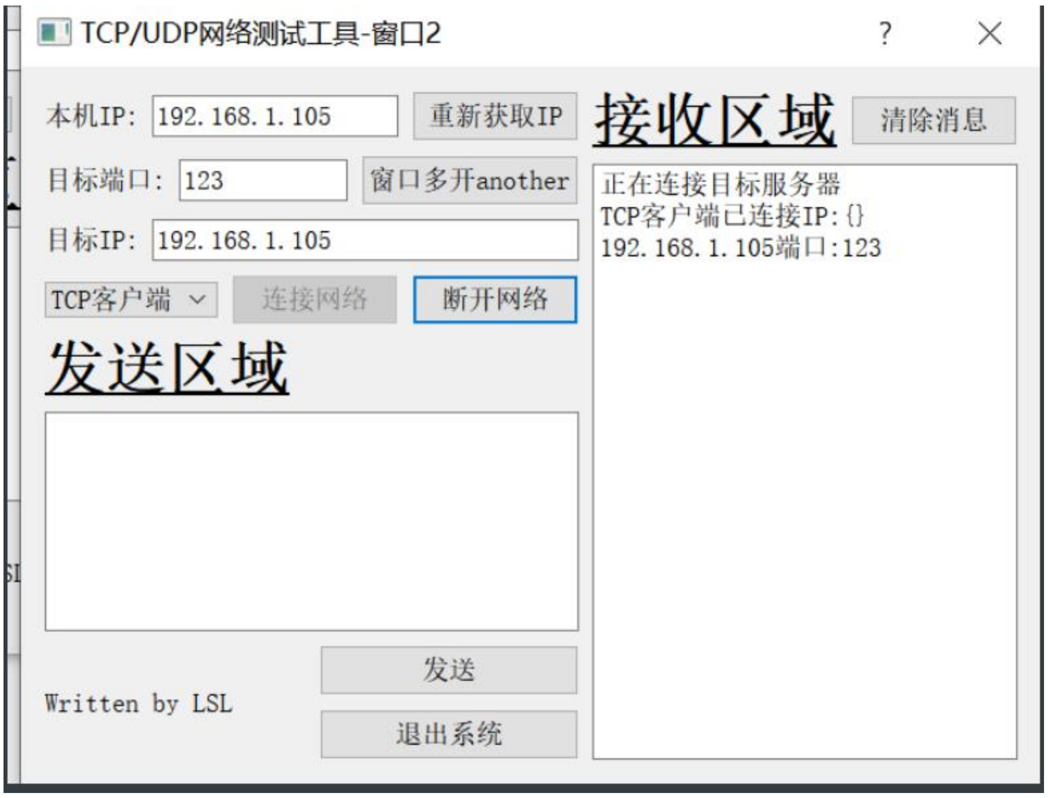
```
c.TCP_1.TCPLogic):
PS C:\Users\Admi
de\extensions\ms
1231089\main.py'
PS C:\Users\Admi
```

3. TCP 服务器设置好端口号，点击连接网络，即可开启 TCP 服务器



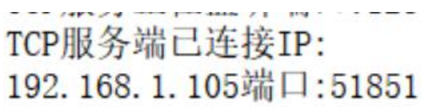
TCP服务正在监听端口:123

4. TCP 客户端设置好目标端口号，目标 IP，点击连接网络。即可开启 TCP 客户端



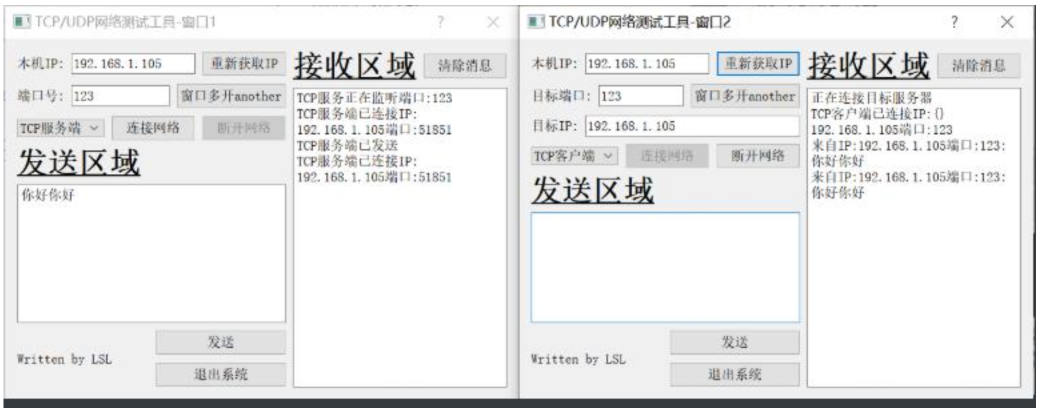
正在连接目标服务器
TCP客户端已连接IP:{}
192.168.1.105端口:123

TCP 服务器显示连接成功



```
TCP服务端已连接IP:
192.168.1.105端口:51851
```

5. TCP 服务器发送消息



TCP/UDP网络测试工具-窗口1

本机IP: 192.168.1.105
端口号: 123
TCP服务端

发送区域

你好你好

接收区域

```
TCP服务端正在监听端口:123
TCP服务端已连接IP:
192.168.1.105端口:51851
TCP服务端已发送
TCP服务端已连接IP:
192.168.1.105端口:51851
```

Written by LSL

TCP/UDP网络测试工具-窗口2

本机IP: 192.168.1.105
目标端口: 123
目标IP: 192.168.1.105
TCP客户端

发送区域

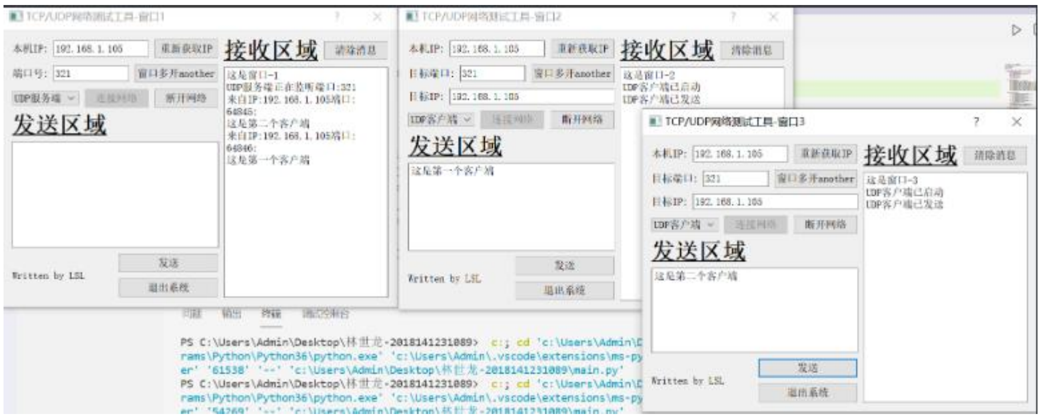
接收区域

```
正在连接目标服务器
TCP客户端已连接IP: ()
192.168.1.105端口:123
来自IP:192.168.1.105端口:123:
你好你好
来自IP:192.168.1.105端口:123:
你好你好
```

Written by LSL

二. UDP 实现的功能

1. 在下拉框中选择 UDP 服务器，设置好对应的端口号，连接网络
2. 选择 UDP 客户端，设置好目标端口号和目标 IP。点击连接网络
3. 在多个客户端中发送消息



TCP/UDP网络测试工具-窗口1

本机IP: 192.168.1.105
端口号: 321
UDP服务端

发送区域

接收区域

```
这是窗口-1
UDP服务端正在监听端口:321
来自IP:192.168.1.105端口:
64845:
这是第二个客户端
来自IP:192.168.1.105端口:
64846:
这是第一个客户端
```

Written by LSL

TCP/UDP网络测试工具-窗口2

本机IP: 192.168.1.105
目标端口: 321
目标IP: 192.168.1.105
UDP客户端

发送区域

这是第一个客户端

接收区域

```
这是窗口-2
UDP客户端已启动
UDP客户端已发送
```

Written by LSL

TCP/UDP网络测试工具-窗口3

本机IP: 192.168.1.105
目标端口: 321
目标IP: 192.168.1.105
UDP客户端

发送区域

这是第二个客户端

接收区域

```
这是窗口-3
UDP客户端已启动
UDP客户端已发送
```

Written by LSL

```
PS C:\Users\Admin\Desktop\林世龙-2018141231089>  c:; cd 'c:\Users\Admin\D
rams\Python\Python36\python.exe' 'c:\Users\Admin\.vscode\extensions\ms-py
er' '61538' '--' 'c:\Users\Admin\Desktop\林世龙-2018141231089\main.py'
PS C:\Users\Admin\Desktop\林世龙-2018141231089>  c:; cd 'c:\Users\Admin\D
rams\Python\Python36\python.exe' 'c:\Users\Admin\.vscode\extensions\ms-py
er' '54269' '--' 'c:\Users\Admin\Desktop\林世龙-2018141231089\main.py'
```