

5.网络部分

网络编程实现断线重连

1. 使用TCP通信，在 .pro 文件中添加 network 模块

```
QT += network
```

2. 添加头文件

```
#include <QTcpSocket>
#include <QTimer>
```

创建网络套接字，定时器

```
private:
    QTcpSocket msocket;
    QTimer mtimer;
private slots:
    void timer_connect(); // 网络连接
    void stop_connect(); // 停止连接
    void start_connect(); // 开始连接
```

3. 具体实现

```
/* 在 FaceAttendance 构造函数中*/

// QTcpSocket 当断开连接时: disconnected信号, 连接成功: connected信号
// connect2, 当断开连接时, 定时发送信号
connect(&msocket, &QTcpSocket::disconnected, this,
        &FaceAttendance::start_connect);
// connect3, 连接成功, 停止连接
connect(&msocket, &QTcpSocket::connected, this,
        &FaceAttendance::stop_connect);

// 定时连接服务器, connect1
connect(&mtimer, &QTimer::timeout, this, &FaceAttendance::timer_connect); // 定
```

时器， 定时时长， ， 运行

// 启动定时器

mtimer.start(5000); //每5s连接一次， 直到连接成功

```
void FaceAttendance::timer_connect()
```

```
{
```

```
    //连接服务器
```

```
    msocket.connectToHost("IP号",端口号); // 具体看本机
```

```
    qDebug() << "正在连接服务器";
```

```
}
```

```
void FaceAttendance::stop_connect()
```

```
{
```

```
    mtimer.stop(); //停止定时器， 不在发送信号
```

```
    qDebug() << "连接服务器成功";
```

```
}
```

```
void FaceAttendance::start_connect()
```

```
{
```

```
    mtimer.start(5000); // 定时器设置5s， 5s后connect1发送连接信号
```

```
    qDebug() << "断开服务器";
```

```
}
```

4. 结果

正在连接服务器

正在连接服务器

正在连接服务器

正在连接服务器

正在连接服务器

连接服务器成功

断开服务器

正在连接服务器

连接服务器成功

断开服务器

TCPUDP连接工具

下载方式: [TCP/UDP测试工具下载及使用教程_tcp &udp测试工具-CSDN博客](#)

新建一个服务器



