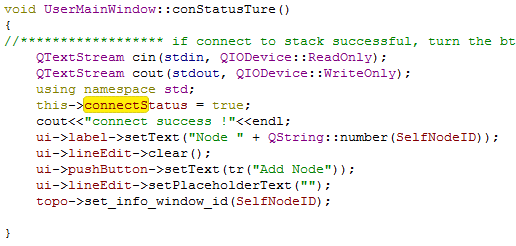
启动一个通信节点窗口之前，先隐藏config和operate两个button。然后调用connectServer函数与协议栈进程建立套接字连接。

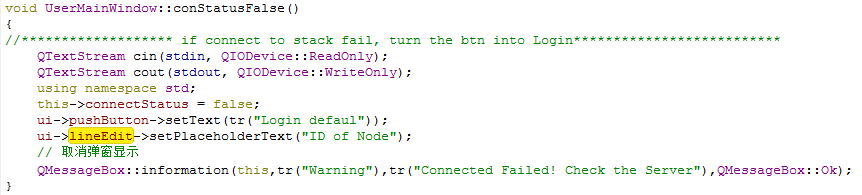
**如果与协议栈套接字连接成功：**

有一个槽函数，如果套接字建立成功，则调用conStatusTrue，表示该节点可以通信。（显示节点ID，并且可以添加其他节点进行通信。即只要跟协议栈连通，该节点界面就可以正常工作，但是协议栈不一定可以。）





如果连接失败，就不能通信：



**Add Node的button：**getConStatus表示本节点TCP的状态，建立成功才能添加其他节点（如果套接字没连接成功，则再连接一次，不会弹出chat窗口）。lineEdit的text是另一个通信节点。



调用AddNewNodeHandler的时候，会检查本节点的映射中是否有该新节点。如果已经有了就不做任何操作。没有的话就创建一个新的，并且放入映射。

然后调用ShowChatWindows显示该窗口。（就是从映射中找到该窗口的指针，然后show出来。）

关于本节点，与其他节点的Chat窗口的映射：

定义了一个映射：



该节点通信窗口的一个私有成员：



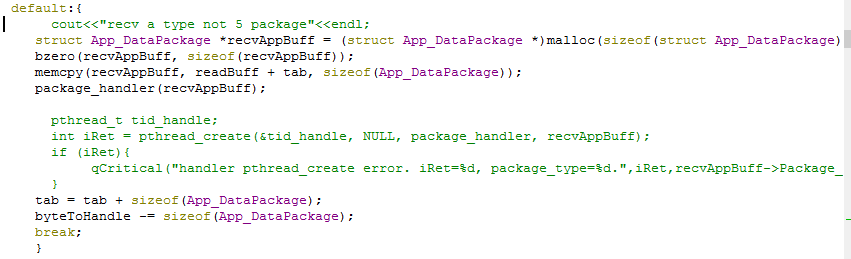
（接收图片窗口是节点窗口的成员，只有一个，接收的时候应该不区分源ID）

创建了一个Chat窗口之后，该窗口属于本节点的通信窗口。（Chat窗口中有一个ower\_指针指向该对象。一般需要发数据也是找到这个节点窗口，调用节点窗口的函数然后发送。）

**节点窗口收到其他节点的数据：**



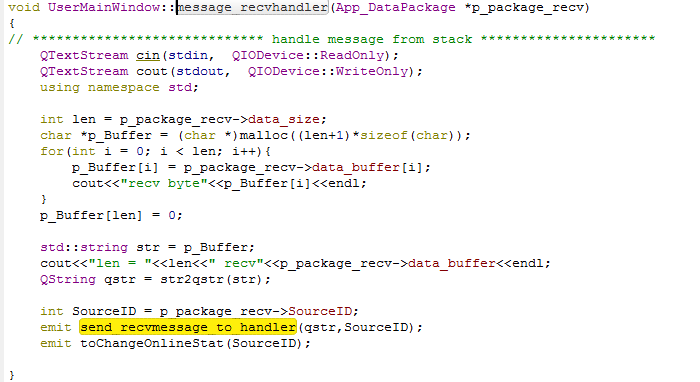
接收到数据时调用readData来处理数据，包类型是4，就读取一个App包的大小，然后调用package\_handler来处理。（这里的处理是：先在readData中通过包类型区分包大小，然后强转，再把指针放到package\_handler中。而package\_handler的传入参数是void\*，个人感觉意义不大，可能是编程过程中没注意到？）



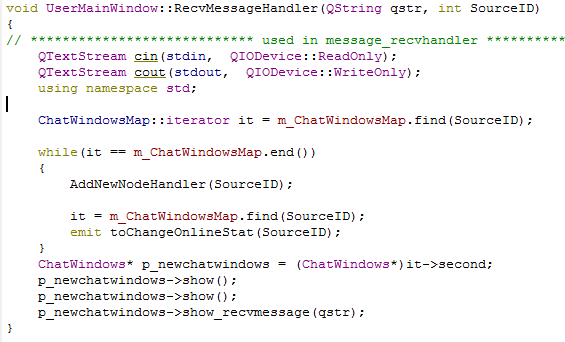
然后就调用message\_recvhandler来处理，



把App数据包解封，获得头部的sourceID和Data部分。然后发送信号，调用槽函数RecvMessageHandler

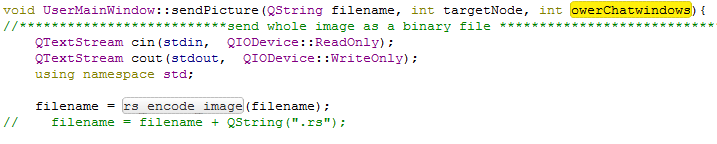


在m\_ChatWindowsMap中找到这个ID的目标点，如果没有找到就调用AddNewNodeHandler来生成并放入Map中。（这里的SourceID其实是目标节点，ChatWindow通过ower来找到本节点ID。）



**发送图片：**chat窗口点击发送以后，把需要发送的文件（还有源目标ID）传入节点窗口的sendPicture函数，然后在发送。

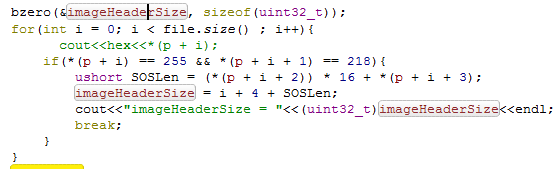
这里先调用python的函数"./python\_script/rs\_encode\_image.py "来加校验位。生成文件为filename + .rs ，即加上后缀rs。



然后读这个文件（QFile），把节点窗口关于图片包的一些数据包头初始化完成，



然后应该是设置图片头的大小。（jpg格式的图片头部有很多种标记码构成，SOS标记码是最后一个标记码。这个标记码开头用0xFFDA来表示开始，并且之后的两个字节表示这个标记码的长度。因此图片头的长度应该是i+2+SOSLen+2。其中，exif格式文件sos两个字节0xFFDA。）



两个图片数据的结构体（图片头和图片数据）



然后设置定时器，超时调用槽函数sendPictureTimer发图片：首先要检查数据是否发完（byteToSend是否大于零）

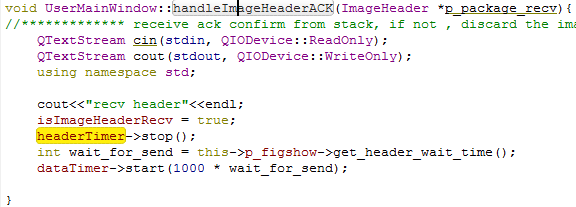
***没有发送完：***

**第一种情况**，图片头还没发送完：



就封装sendImageHeader，然后往本地套接字写入。并且设置ACK定时器（如果超时说明头部没有收到ACK，取消发送。）

如果收到了ACK（定义成了一个信号），会调用槽函数来停止这个定时器，并且设置定时器，继续调用sendPictureTimer，一直到图片头发完。（这样可以保证图片头必须受到，才发送data，不论图片头有多长）



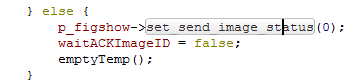
**第二种情况**，图片头已经发完，开始发图片data：（这里又检查了一下标志位isImageHeaderRecv是否满足条件，即是否收到了ACK）



和封装图片头差不多，封装data部分，然后往本地套接字写入。并且设置定时器，下一次发送DATA的时间。

***已经发送完了***

直接设置发送图片窗口状态为idle



这里的emptyTemp意思应该是把临时数据（如offset，数据包，图片头包等）都置零：

