服务器篇:

服务器端

HTcpServer

链接端(客户端)

HConnect

------------------------------------------------------------------------------------

HTcpServer ID=ip

std::map<ip,HConnect> list;存储所有连接

包含多个HConnect连接(每个连接HConnect ID=ip)

HNetApp

std::map<ip,HTcpServer\*> list; 存储所有服务器端

这样的优点就是从任何客户端过来的数据，可以解析后再从原来的链路上发回去。

报文采用统一显示窗口，所有报文都在同一个窗口显示

正常报文

发出报文格式收到报文用color1，发出报文用color2颜色

(2018-4-11 00:00:00)A网(心跳)>> 85 00 00 00 00

(2018-4-11 00:00:00)A网(全遥信)<<86 00 00 00…..

2018-4-11 00:00:00 from:198.120.0.111 <<

如果解析勾上(之前的报文是不能解析的，只能解析后面):

(2018-4-11 00:00:00)A网(心跳)>> 85(标识码:心跳) 00 00(报文长度) 00(类型) 00(监控)

错误报文(不解析红色)

2018-4-11 00:00:00)A网(错误报文)>> 85(标识码:心跳) 00 00(报文长度) 00(类型) 00(监控)

ui->textEdit->setHtml("这个是<span style=' color:#ff0000;'>红色</span>");

"\([\(]from[:0-9.]+[\)]\)"; from:198.120.0.111

"\\b[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}\\b"; 2018-00-00 19:00:23

"\([\(]from[:0-9.\u4e00-\u9fa5]+[\)]\)"; (from:198.120.0.111:心跳)

"2018-11-20 19:18:12 (from:198.120.0.111) (type:心跳) << 85 00 00 00 "; ----1

"[\u4e00-\u9fa5a-zA-Z0-9\_ :<-\(]+<<[0-9 ]+"; --对应上面的

网络设计

三个部分：

1.显示界面报文部分

2.网络收发报文部分

3.规约处理报文部分

由于qt不建议主界面放在线程里面，所以主界面框架为主进程，网络收发、规约处理等放在多线程里面进行处理。

相关类

HNetMonitor 是网络报文显示类

HNetThread 网络线程类

HNetManager 网络管理类

HTcpServer 服务器类

HTcpConnect 客户端类

HProtocol 规约处理类

HNetHandle 网络总处理类

HTcpServer创建服务器端 连接客户端，等待客户端

该类创建一个map列表存储客户端信息。

HTcpConnect 搭建客户端链路，实现报文的接受和发送。

报文接受之后，调用HNetManager接口来进行处理。

HNetManager 管理所有网络的类，实现报文接受和发送的转发

管理HTcpServer类(主备双机)

HTcpConnect接受到报文之后交由此类进行处理，即将报文投入到与规约类构建的共享队列里面。同时定时处理读取发送队列的发送报文，扔给对应的HTcpConnect进行处理

HNetHandle

创建HNetMonitor类、HNetThread类（网络线程类）、HProtocol类（规约处理线程类）

HNetThread 创建 HNetManager类

共享队列：

1.显示报文(消息)共享队列，是所有类与HNetMonitor之间的共享通道

2.网络报文接受消息类（规约与网络之间的）

3.网络报文发送消息类(规约与网络之间的)

发送和接收流程

1.当网络接收到报文之后，调用HNetManager的handle\_receive来处理报文(即添加到共享队列中)，然后规约进行处理。

2.当需要发送报文的时候，只需要组织好报文发送 到发送报文共享队列即可。HNetManager定时处理发送报文队里，当有报文存在的时候，取出报文，然后查看主备网的情况，主网通则主HTcpServer发送，不通调用备用HTcpServer发送。注意：HTcpServer发送一般都是全连接发送，如果要定死链接发送，则需要记录ip相关信息。HTcpServer发送时，调用HTcpConnect的addSend和ProcessSend进行发送