// Rio Process를 처리할 Thread

// Rio에게 주행 준비 신호 송신

<RioThreadPoint>

<<SendPacketpredrive

* SOCKET s: sock\_base[i]
* 매개변수로 1 또는 2
* 매개변수 emp

두번째 매개변수가 1: Rio에게 pre drive(1) 송신

두번째 매개변수가 2: Rio에게 pre drive(2) 송신

<<Sendpacketspeedprof

* SOCKET s: sock\_base[i]
* angle: recv\_angle
* speed: recv\_speed

Rio에게 speed profile 송신

<< SendPacketDrivestart

* SOCKET s: sock\_base[i]
* Start:1
* emp

Rio에게 drive start(1) 송신

<< SendPacketDrivestart

* SOCKET s: p->clientSock
* start:1
* emp

server에게 drive start(1) 송신

isDriveSt = true; // Drive start Set

p->m\_bRelease = true; // NTCurlingDlg에 주행 시작을 알리기 위한 변수 set

// Server로 부터 패킷을 받는 Thread

<RecvThreadPoint>

SOCKET sock = p->clientSock;

<< RecvClassify

1. 패킷이 PRE\_MODE형일 경우

<<RecvPacketPremode

server에게 Pre Mode 전송

p->m\_pre: 연습모드일 때는 1(true), 경기모드일 때는 0(false)

Thrower: p->m\_mode == robot \_id.

{

isThrower=true;

카메라 폴대를 접기 위해 m\_bSetCam을 reset(false)

<<SendPacketRioreset

rio에게 Rio ReSet 전송

}

skip : p->m\_mode != robot\_id

{

카메라 폴대를 펴기 위해 m\_bSetCam을 set(true)

isThrower=false;

}

// Rx,Rio 양쪽 모두 주행 준비 신호 송신

<<SendPacketPremode

TX와 Rio에게 Pre Mode 전송

<<SendPacketStoneInfoAck

Server에게 Pre Mode Ack 전송

1. 패킷이 CALL\_STONE\_INFO형일 경우

skip일 경우

{

Server에게 Call Stone Info 전송

TX에게 Call Stone Info 전송

}

<<SendPacketStoneInfoAck(매개변수2:call stone info)

Server에게 Call Stone Info Ack전송

1. 패킷이 CALL\_ROBOT\_INFO형일 경우

thrower일 경우

{

Server에게 Call robot Info 전송

TX에게 Call robot Info 전송

}

1. 패킷이 ROBOT\_MODE형일 경우

<<RecvPacketThrowmode

Server에게 robot mode 전송

Thrower: p->m\_mode == robot \_id.

{

isThrower=true;

카메라 폴대를 접기 위해 m\_bSetCam을 reset(false)

<<SendPacketRioreset

rio에게 Rio ReSet 전송

}

skip : p->m\_mode != robot\_id

{

isThrower=false;

카메라 폴대를 펴기 위해 m\_bSetCam을 set(true)

}

TX에게 robot mode 전송

1. 패킷이 RELEASE형일 경우

thrower이고 isReady(robot info call수신)인 경우,

{

p->m\_recvRelease = true;

<<RecvPacketRelease

Server에게 release 전송

p->m\_bCW = isRight;

pos.x와 tar\_pos.x의 차이점 알아보기

}

skip이지만 연습모드인 경우

{

<<RecvPacketRelease

Server에게 release 전송

pos.x와 tar\_pos.x의 차이점 알아보기

}

<<SendPacketStoneInfoAck(매개변수 4: release)

Server에게 release ack 전송

1. 패킷이 FLAG형일 경우

SKIP이라면

{

TX에게 flag 전송

}

1. 패킷이 STONE\_INFO\_ACK형일 경우

SKIP이라면

{

TX에게 stone info ack 전송

}

1. 패킷이 EMERGENCY형일 경우

<<SendPacketEMERGENCY

TX에게 emergency 전송

<<SendPacketpredrive

rio에게 pre drive(5)전송

1. 패킷이 RESTART형일 경우

<<SendPacketRestart

Tx에게 Packet Restart 전송

<<SendPacketpredrive

rio에게 pre drive(1)전송

isReady = false;// ready 신호(robot info call 수신 시 set, release end 시 reset)

isDriveSt = false; // drive start 신호

isRioRst = false; // Rio reset 신호

p->m\_fHogOffset = 0;

p->m\_bRelease = false;

p->m\_bReleaseEnd = false;

p->m\_bSetCam = false;

1. 패킷이 MY\_TURN형일 경우

<<RecvPacketMyturn

server에게 my turn전송

<<SendPacketMyturn

server에게 my turn전송

1. 패킷이 INFO\_TIME형일 경우

// 모드 송신 (Joystic 조작용)

<SendMode>

if (prep != 0) // 입력 모드가 0이 아니면 Rio로 모드 송신

<<SendPacketpredrive

* Socket s: sock\_base[i]
* prep
* 0

<data>

헤더설정

run 패킷형

preparation의 주소를 4칸씩 나누어 넣는다.

emp의 주소를 4칸씩 나누어 넣는다.

Send

Rio에게 pre drive(%d), prep 송신

<< SendPacketDrivestart

* Socket s: sock\_base[i]
* Start:0
* Emp:0

Rio에게 drive start(0) 송신

// 주행 값 송신 (Joystic 조작용)

TestMove

<<SendPacketSpeedProf

* Socket s: sock\_base[i]
* Angle: m\_fTestAngle
* Speed: m\_fTestSpeed

Rio에게 speed profile 송신

<< SendPacketDrivestart

* Socket s: sock\_base[i]
* Start:1
* Emp:0

Rio에게 drive start(1) 송신

// TX에게 각도&위치 정보 요청 (Joystic 조작용)

<CallAngle>

<<SendPacketCallrobotinfo

* Socket s: sock\_base[i]
* isDist: 0

<data>

헤더설정

Call\_robot\_info 패킷형

두번째 매개변수를 넣는다.

Send

TX에게 call robot info 송신

// TX에게 각도&위치 정보 요청 (Joystic 조작용)

<SendAngle>

<<SendPacketTargetangle

* Socket s: sock\_base[i]
* t\_angle: 1.5
* emp: 0

<data>

헤더설정

TARGET\_ANGLE 패킷형

t\_angle의 주소를 4칸씩 나누어 넣는다.

emp의 주소를 4칸씩 나누어 넣는다.

Send

Rio에게 target angle 송신

<SendCalib>

<<SendPacketCalibAngle

* Socket s: sock\_base[i]
* d1: 1
* d2: 0

<data>

헤더설정

Calib\_angle 패킷형

D1의 주소를 4칸씩 나누어 넣는다.

D2의 주소를 4칸씩 나누어 넣는다.

Send

Rio에게 call angle 송신

<SendEmergency>

<< SendPacketDrivestart

* Socket s: sock\_base[i]
* Start: 0
* Emp: 0

Rio에게 drive start 송신

<<SendPacketEMERGENCY

* Socket s: sock\_base[i]
* emp1: 0
* emp2: 0

<data>

헤더설정

emergency 패킷형

emp1의 주소를 4칸씩 나누어 넣는다.

emp2의 주소를 4칸씩 나누어 넣는다.

Send

TX에게 emergency 송신