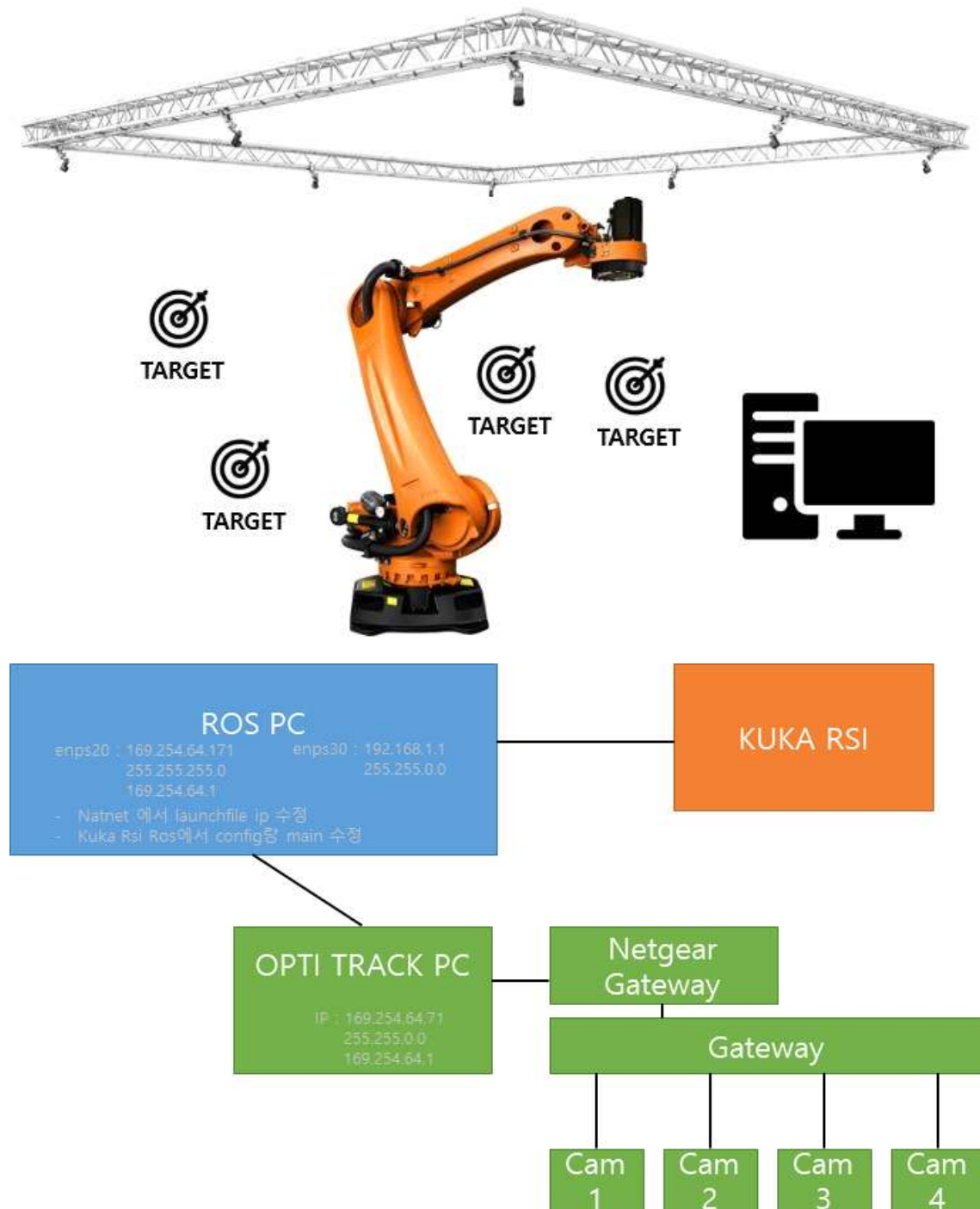
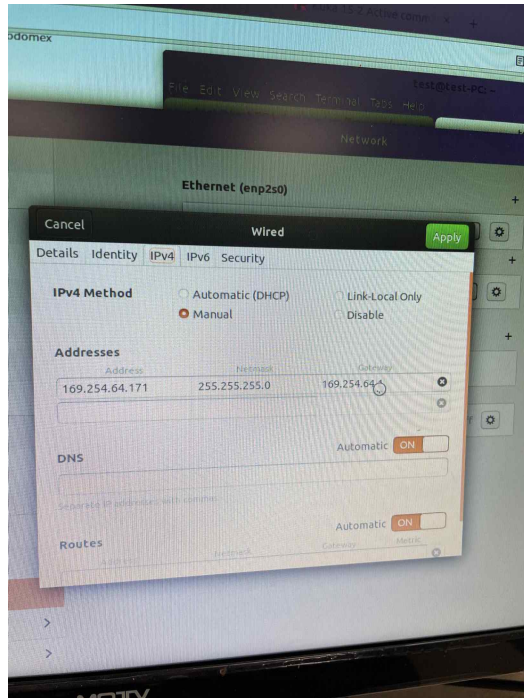


Fluent Manufacturing System Manual



- Environment Setting

1. ROS - OPTI TRACK



natnet_ros.launch

```
<?xml version="1.0"?>
```

```
<launch>
```

```
  <!-- LOGGING -->
```

```
    <arg name="log_internals" default="false"/>
```

```
    <arg name="log_frames" default="false"/>
```

```
    <arg name="log_latencies" default="false"/>
```

```
  <!-- CONNECTION -->
```

```
    <arg name="serverIP" default="169.254.64.71"/>
```

```
    <arg name="clientIP" default="169.254.64.171"/>
```

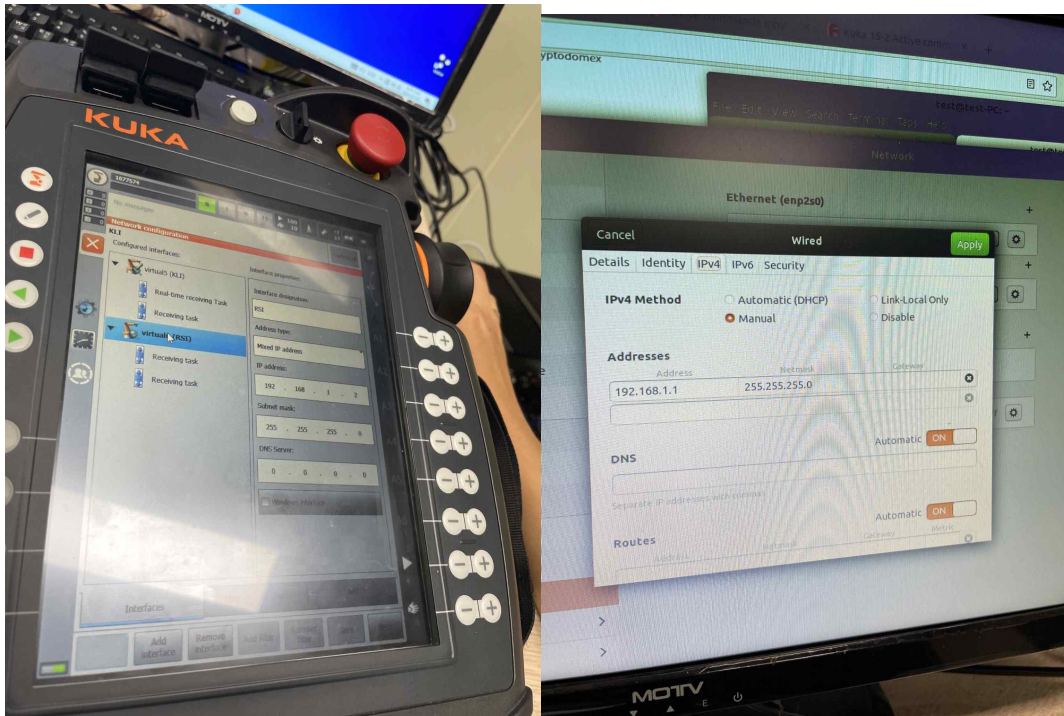
```
    <arg name="serverType" default="multicast"/>
```

```
    <arg name="multicastAddress" default="239.255.42.99"/>
```

```
    <arg name="serverCommandPort" default="1510"/>
```

```
    <arg name="serverDataPort" default="1511"/>
```

2. ROS - KUKA



config.yaml

Server settings.

server:

IP address to bind to on the host PC.

ip_address: "192.168.1.1"

Port to bind to on the host PC.

port: 59152

Buffer size used for receiving data from the robot controller.

buffer_size: 1024

main.cpp

// Load server address setting.

std::string server_address;

nh.param<std::string>("server/ip_address",

server_address,

"192.168.1.1");

3. KUKA Controller에 krc files 넣기 (올바른 위치에 있어야 함)

1. github 페이지 있는 krc 파일은 예전 버전이라 실행이 안됨 (로봇컨트롤러에서)

2. sucess 폴더내 있는 파일은 로봇 움직이는데 성공한파일

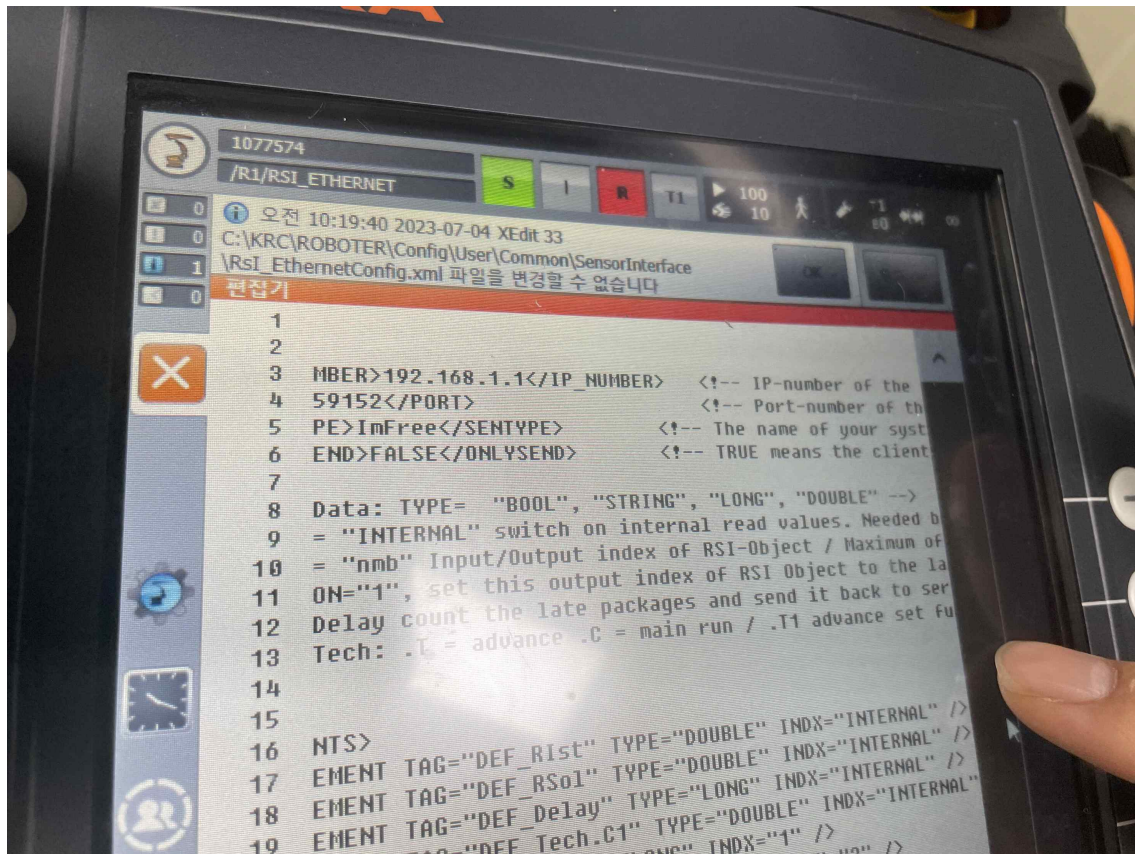
[KRC:\R1\Program 에서 실행함]

- RSI_Ethernet_YJ2.src

.ptp \$pos_act 가 없어서 안됨

[C:\KRC\ROBOTER\Config\User\Common\SensorInterface에 복사함]

- RSI_Ethernet + 세팅

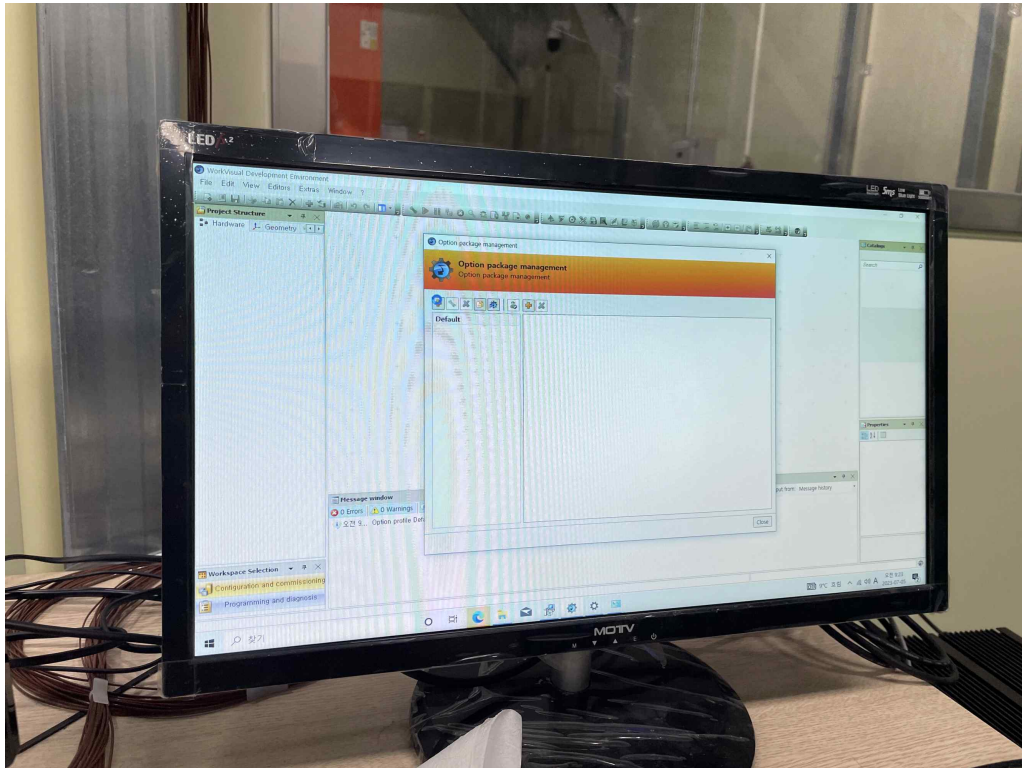


- RSI_EthernetConfig

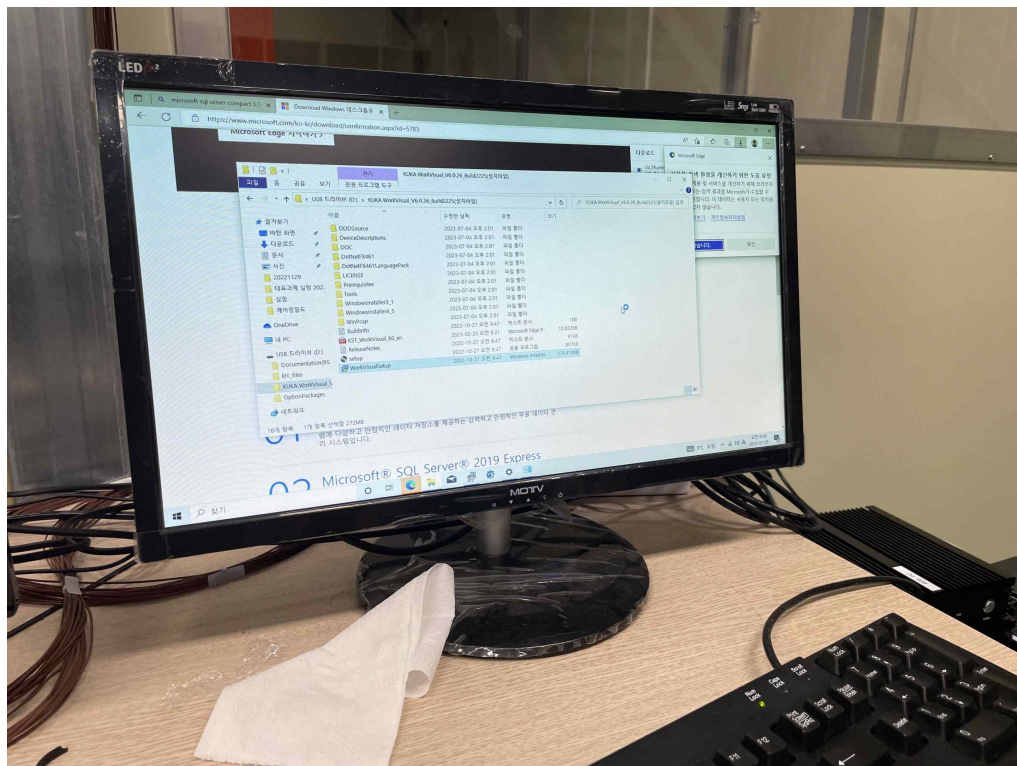
안됐던 이유는 \$POSCORRMON 블록이 없어서 안됨

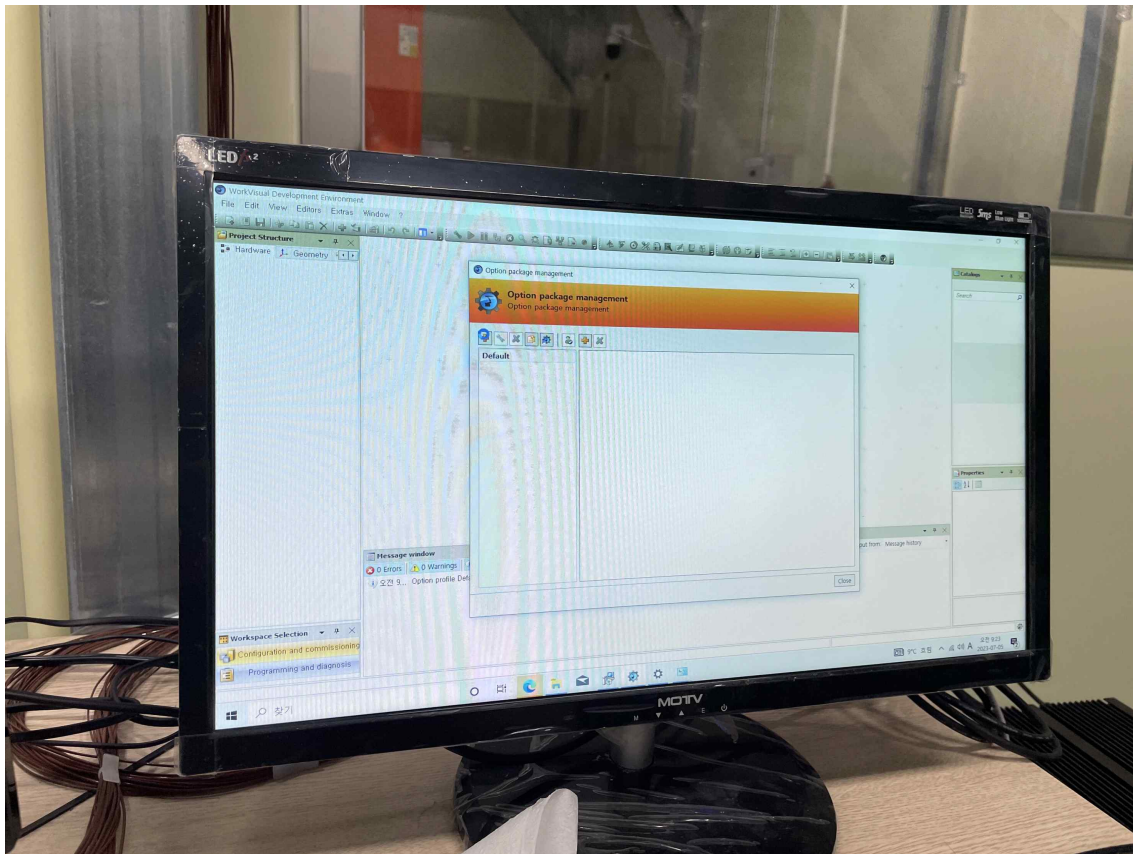
-> Work Visual에서 수정

1.(KUKA 이더넷 케이블 연결)



2.

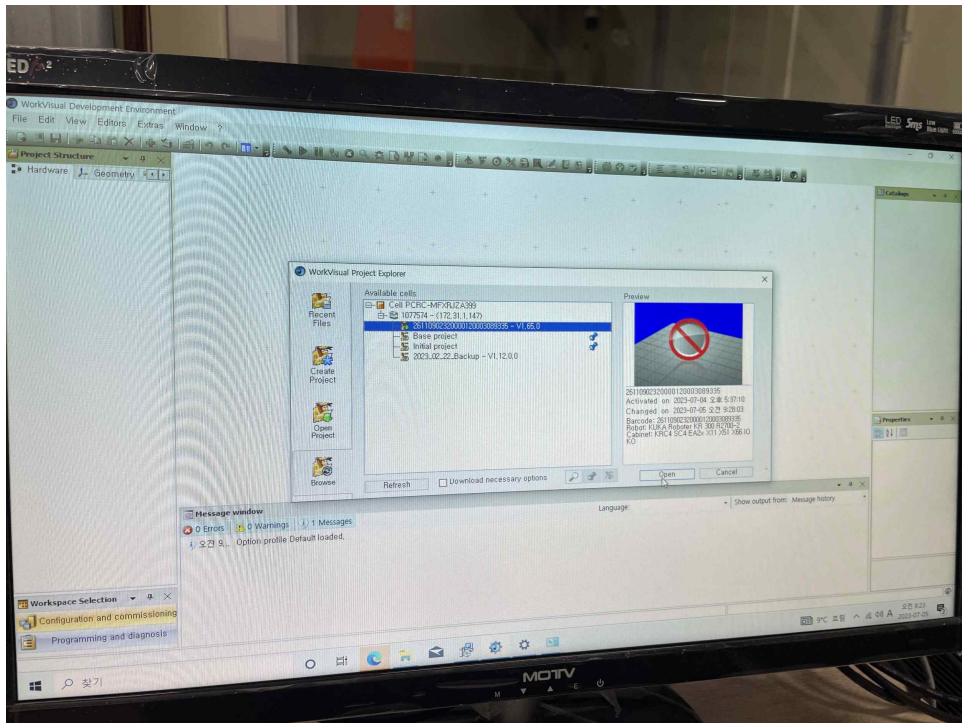




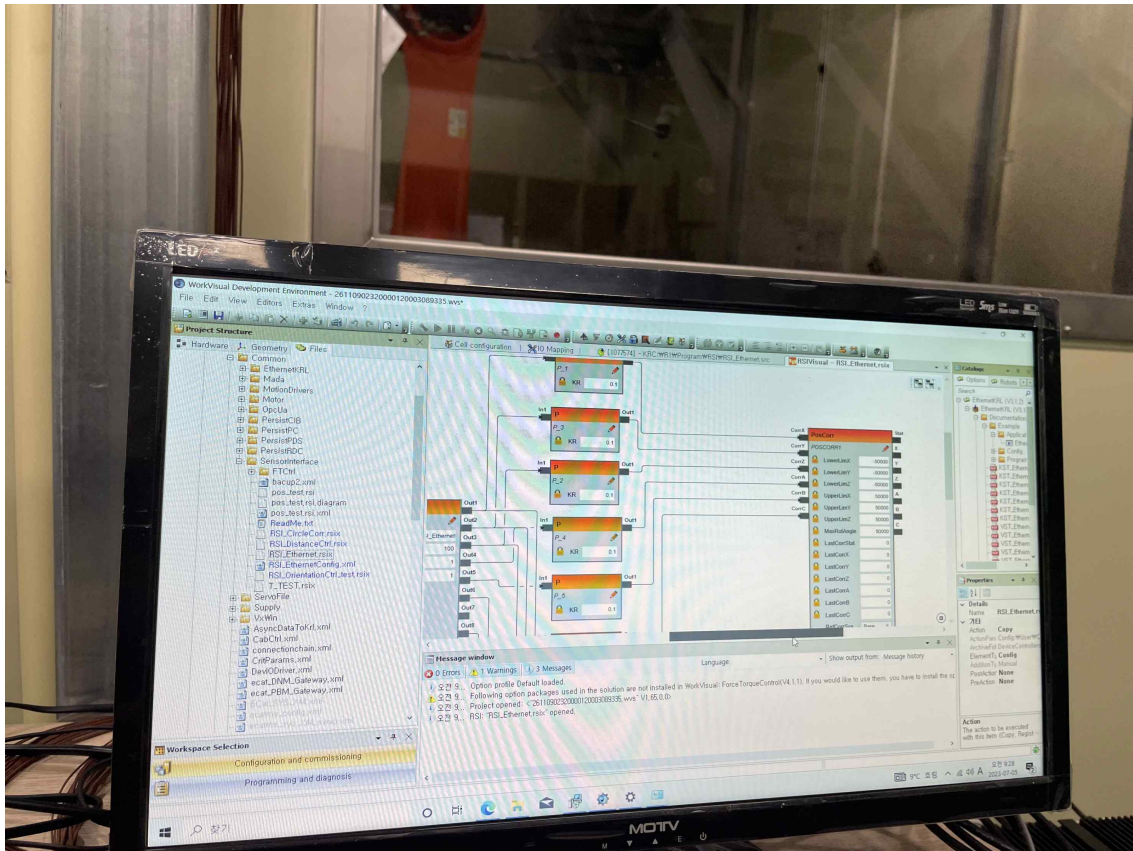
3. 더하기 그림 클릭해서 필요한 라이브러리들 불러와야 함.

5개 불러오고 나서 체크하고 reload

4. (확실 x)



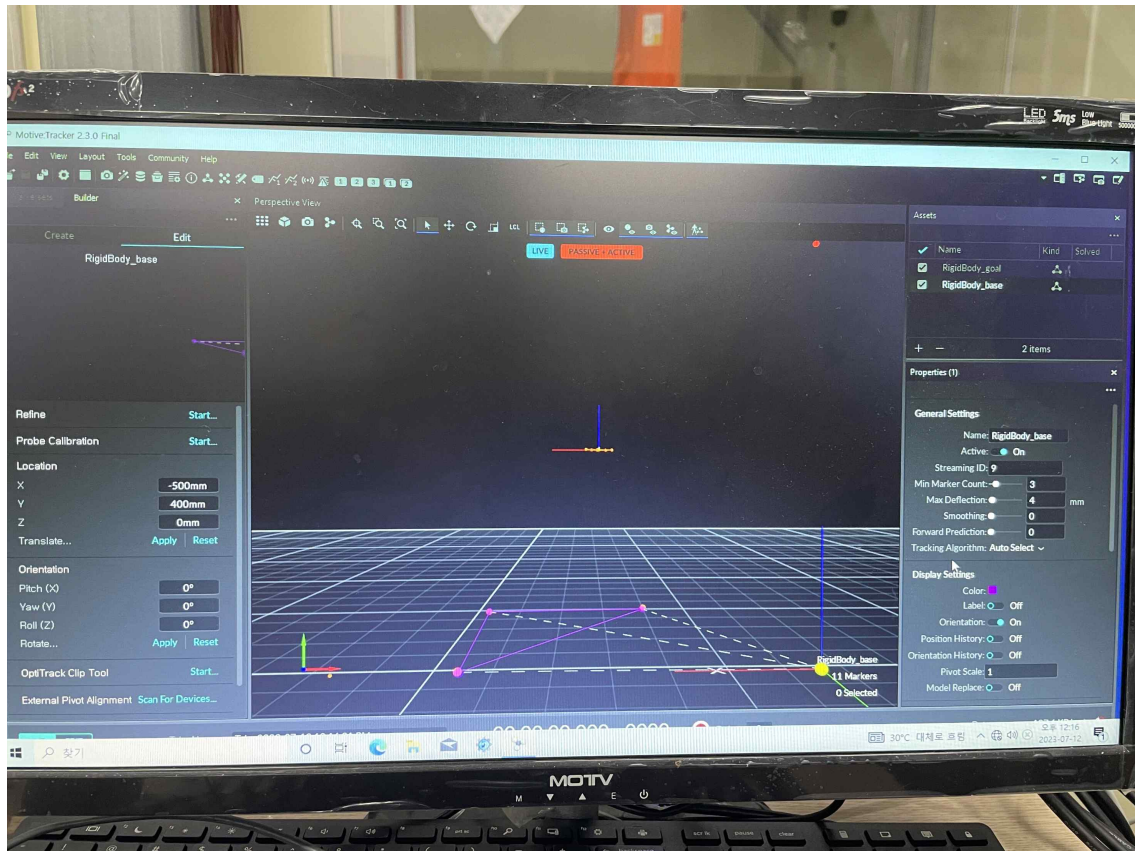
5. Work Visual에서 rsix 파일 수정.



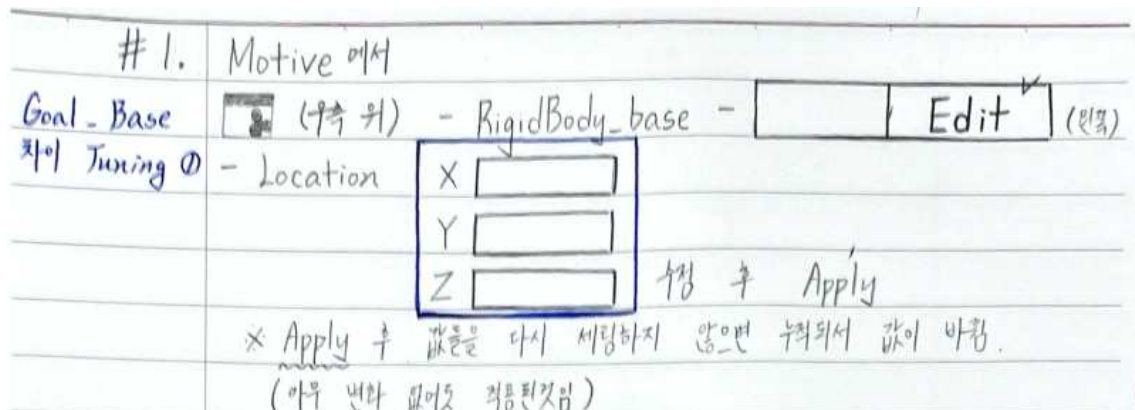
컴파일 하고, 선 연결 하면 실행 준비 완료

실행

1. Window (Opti track PC)



위 화면에서,

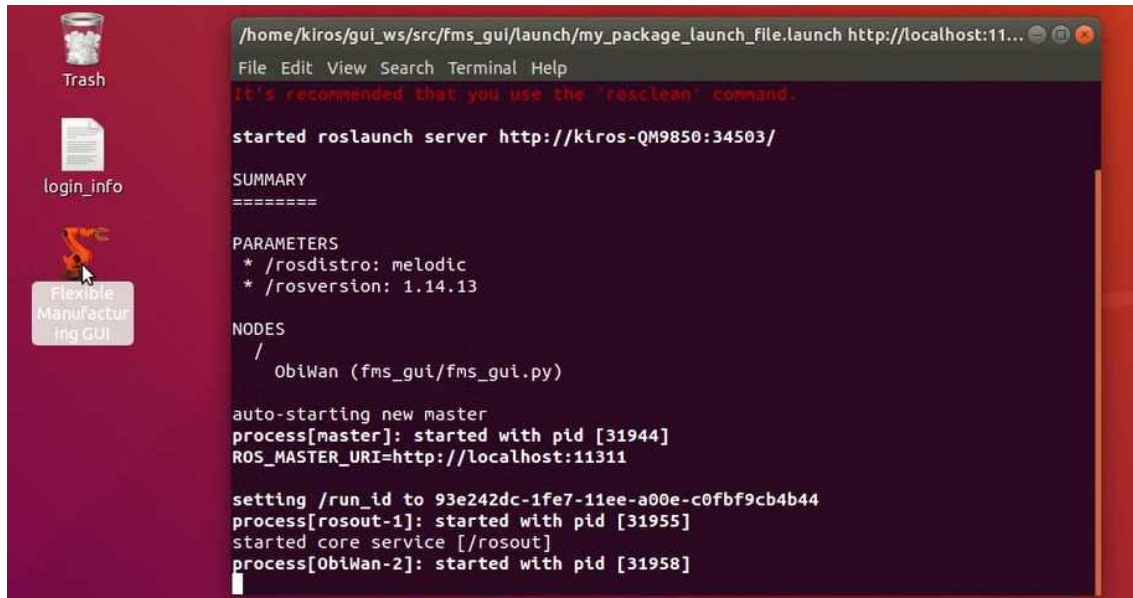


하고 나서, View -> Data Streaming

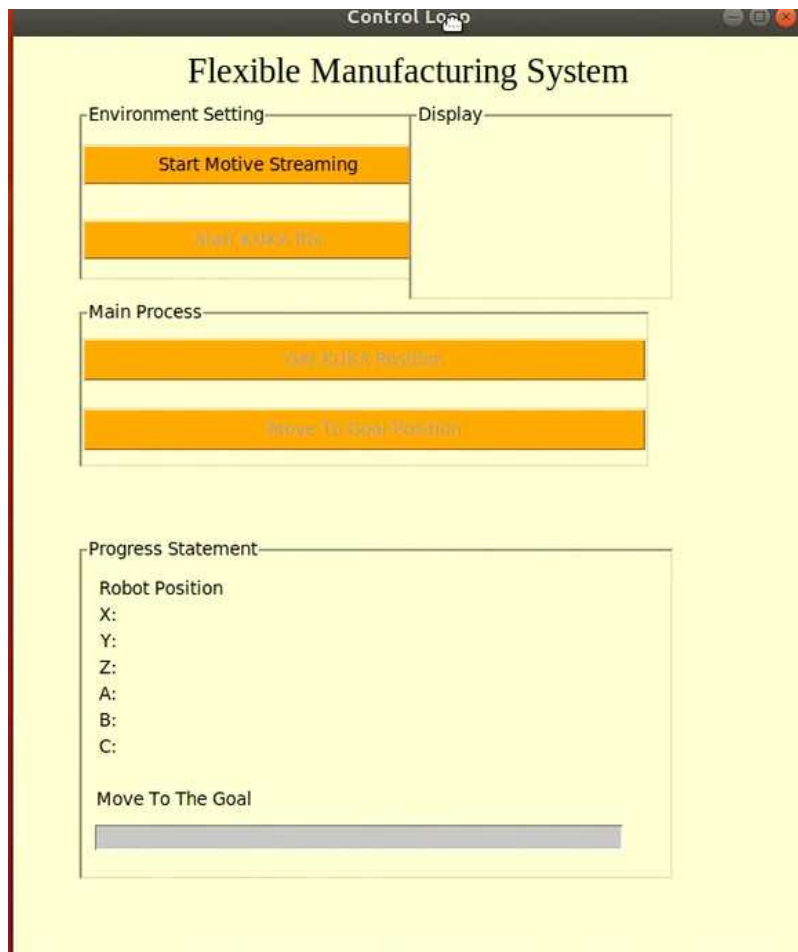
: 여기에 뜨는 ip로 전송중.

2. ROS PC (GUI)

Flexible Manufacturing GUI 클릭



GUI에서



Start Motive Streaming : natnet 실행하는 launch파일 실행함

Start KUKA RSI : KUKA RSI ROS 실행

KUKA Controller 이때 작동

[KRC:\R1\Program 에서 실행함]

- RSI_Ethernet_YJ2.src

Select 하고 뒤에 안전 버튼 누르면서 ► , Confirm all

~ RSI MOVECORR() 갈때까지

그 다음, Get KUKA Position 눌렀을 때 Robot Position을 받아오면 통신이 잘 되고 있는 것

Move To The Goal을 눌러서 원하는 물체로 이동.

- 이걸 누르면, get_topic_srv_client_using-TF.launch를 실행시킴.

: natnet_ros에서 RigidBody_base를 subscribe해서, lookupTransform이라는 함수로 RigidBody_base에서 RigidBody_goal 까지의 Transform을 계산하고 KUKA의 data형식으로 변환함. 그리고 trajectory를 계산해서 전송 하는데, 아래 설명에 따라서 Goal과 End effector의 거리가 멀 경우 tuning할 수 있음.

4. Move To Goal Position

: sim_ws의 - motive2kuka - get_topic_srv_client_using-TF.launch

• trajectory에 관련된 parameter 설정 가능.

time 92 • src의 get_topic_srv_client_using-TF.cpp 실행시킴

sub = nh.subscribe("/natnet_ros/RigidBody-base/pose"...)

Transform을 subscribe

122 transformStamped = tfBuffer.lookupTransform

("RigidBody-base", "RigidBody-goal", ros::Time(0));

Goal_Base 4이 : base에서 goal을 바라본 transform 계산 ... 변경

Tuning @ 145 Kuka-Pose G.k = {x.D * 1000, y.D * 1000, x.D * 1000 + 100, euler_yaw * 180.0/M.PI, euler_pitch * 180.0/M.PI, euler_roll * 180/M.PI}

154 std::vector<double> thetaend = {G.k.x, G.k.y, G.k.z, 135, 0, 175}

(좌표계에서 각도만 쓰면 안됨) end-effector에서 임의의 설정 (x, y, z)

166 launch file에서 가려진 parameter들을 가지고 initial pose (I.k) 다

Goal pose (G.k) 사이의 trajectory 생성.

• cpp 수정하면 catkin_make source devel/setup.bash.

* Gear Torque Exceed Axis A6 : Base Tuning Young green

하사보면 특정 지점에서 여러 발생.