



C# & Robot

1) 2) 로봇 ...

계획 [1/2]

- 1. C# 설치
 - Visual Studio C# 설치, 닷넷 프레임 환경 구성
 - 로봇 기초 교육 (설치 시간동안)
 - 간단한 이론, 로봇 / 모터 / 프로그래밍 툴 의 종류 등)
 - 프로그래밍을 배우면 뭐가 가능한가?
 - 프로그래밍은 CPU(컴퓨터의 머리)에서 어떻게 해석되는지?
 - 맞춰기 프로그래밍 (스트리밍: 카메라의 영상을 인터넷으로)
- 2. C# 기초
 - 단축키, 메시지 창 띄우기
 - 변수의 종류 및 사용
 - GUI 배치 쉽게 하기
 - 타이머, 텍스트박스, 레이블, 콤보박스, 리스트박스, 체크박스 사용법
 - 실습: 버튼을 클릭한 계산기 만들기
- 3. 오픈지그웨어 사용법 기초
 - DLL 설치, 오픈지그웨어의 소스 내부를 들여다 보기
 - 메시지 히스토리 (디버깅메세지 - 파일기록)
 - 파라미터 사용 (텍스트 박스 등에 초기값 유지하기)
 - 타이머 클래스 사용
- 4. 오픈지그웨어 특수 기능 사용하기
 - 카메라출력, 스트리밍 서버 (바탕화면 전송, 카메라영상전송), 스트리밍 영상 가져오기
 - 가상키보드 (외부 / 내부) 사용, 프로그램간 글자 주고받기
 - 다른 프로그램 강제 실행하기 / 강제 종료하기 (메모장, 그림판 등)
 - 내 프로그램 중복 실행 방지 등
 - 파일 관리
 - 시리얼 통신 / 소켓통신
- 5. 오픈지그웨어 2D
 - 간단한 파형 실험 그래프 만들기
 - 2D 로 그림 그리기
 - 만들어진 2D 그림을 3차원 회전해 보자.
- 6. 오픈지그웨어 3D
 - 내 프로그램 내의 가상공간 만들기
 - 점, 선, 면을 만들어 보자
 - box, 원기둥, 구 를 만들어 보자
 - stl 파일을 로드하자.

계획 [2/2]

- 7. 오픈지그웨어 모터 제어하기 (쿠루쿠루의 모터 총동원)
 - 모터 클래스 선언 (오픈지그웨어 몬스터 라이브러리)
 - 속도제어 / 위치제어
 - 모터 설명 : 컨트롤테이블 & 온라인 메뉴얼 확인 , 작동 원리)
- 8. 3D 모델링 & 모션 툴 사용 (로봇팔 만들기)
 - 스켈레톤 모델링 (로봇팔 간단 설계)
 - 만들어진 모델링을 이용한 모션 제작
 - 제작된 모션을 몬스터 라이브러리로 제어해 보자 .
 - " 복습 : 10 분만에 로봇팔 모델링 만들고 제어해 보기 "
- 9. 더 예쁘게 만들어 보자
 - Fusion 360 을 이용한 모델링 기초 (기대마시길 나도 잘 모름 ... _-;))
 - STL 만들고 3D 프린팅 (메이커 스페이스 활용)
 - 만들어진 STL 용량 줄이기 작업 & 내 모델링에 넣어보자 .
- 10. 실습
 - 프로그램에 3D 띄우고 제어하기
 - 3D 와 로봇 , 같이 제어하기
- 11. 모션 만들기 심화
 - 모션 만들기 심화
 - R+Motion2 툴과 혼용하는 방법 (상호 복사 / 붙여넣기)
 - Microsoft Excel 활용하기 (일반복사 , 수식 복사)
 - 엑셀만으로 로봇 제어하기
- 12. 최종실습
 - 조이스틱 + 로봇팔 제어
- 13. 종료 & 배우지 못한 부분 설명 (실습없이 설명만 ...)
 - 오픈 지그웨어 내의 Forward / Inverse Kinematics
 - 3D 모델링을 그리면 자동으로 만들어지는 수식 살펴보기
 - Inverse Kinematics 를 풀었을 때의 제어 차이점
 - Inverse Kinematics 가 들어있는 ojw 파일을 이용한 로봇팔 제어

진행 목차

- 1. 모터의 직접 제어
- 2. 모델링 툴 → Show()

```
Ojw.CTools CTool = new Ojw.CTools();  
CTool.ShowTools_Modeling(1.0f);
```

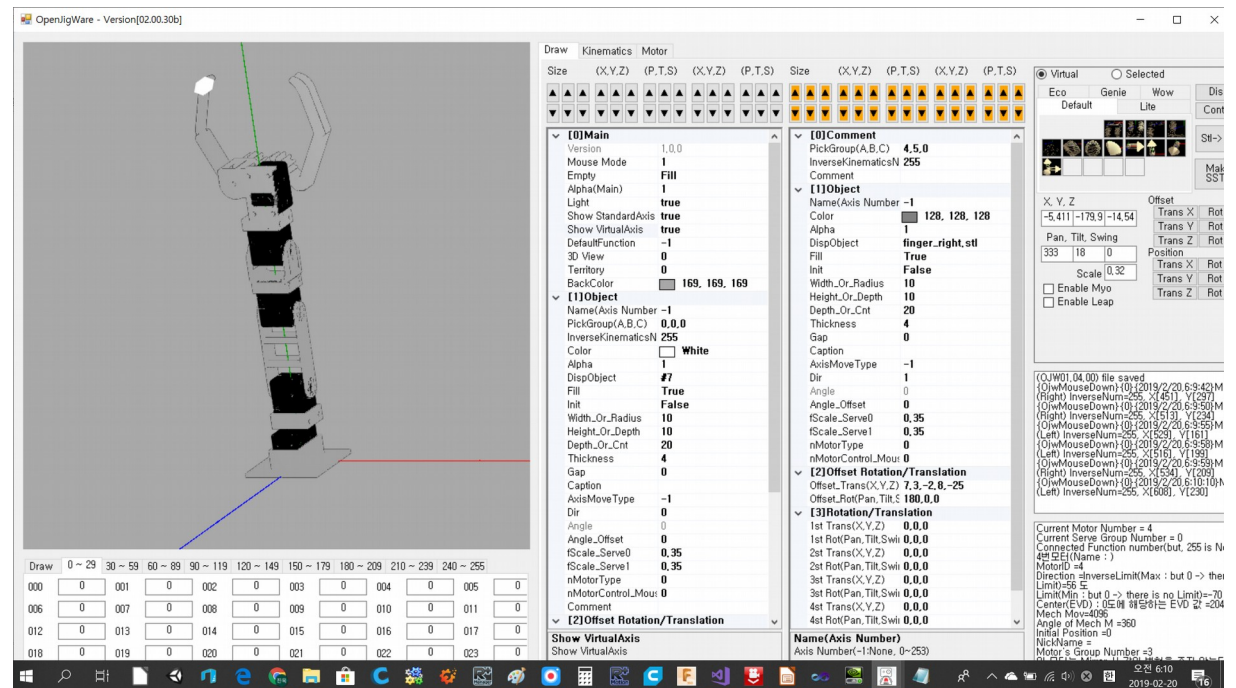
- 3. 모션 툴 → Show()

```
Ojw.CTools_Motion CTool = new Ojw.CTools_Motion();  
CTool.ShowTools(1.0f);
```

-

3D 모델링을 위한 디자이너 툴

- 3D 라이브러리 추가
 - Tao ...
- ~~3D~~ 그리기
- 간단히 살펴 보기

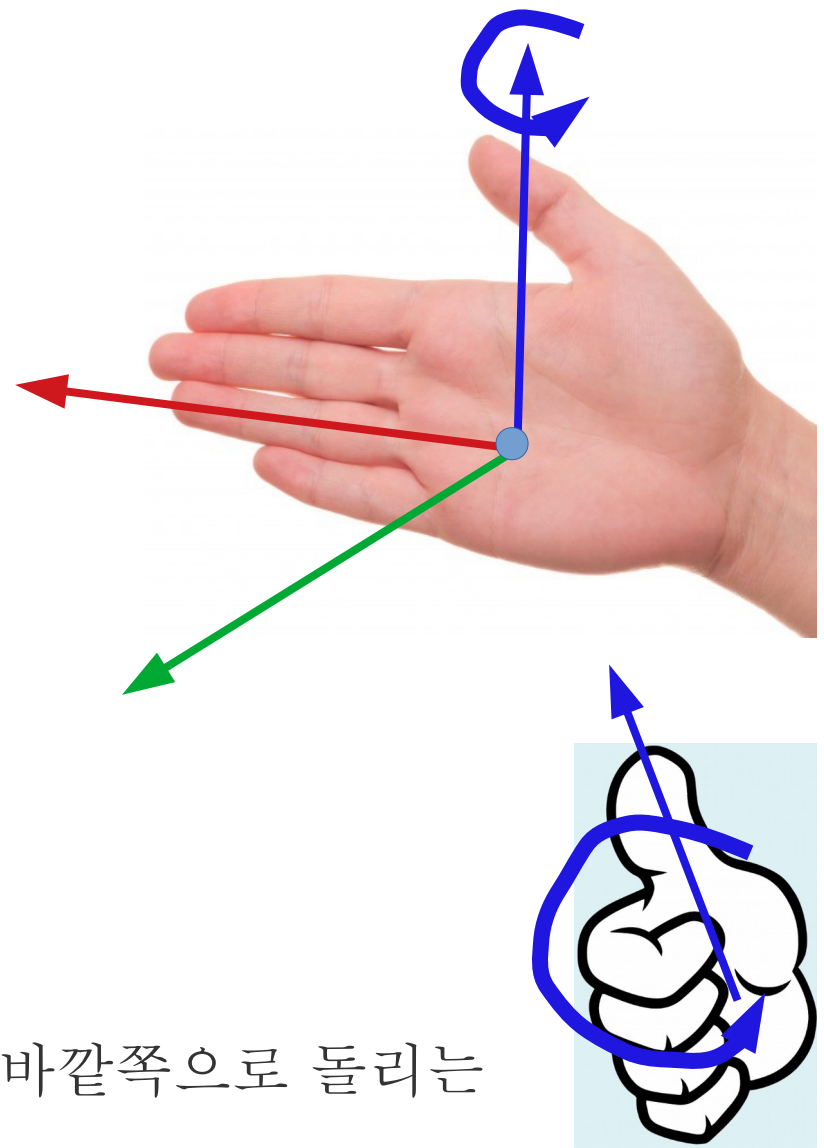


STL 입하기

- stl 모델링을 만들어 보자
 - Fusion 360
- 만들어진 모델링을 그대로 사용하면 안된다.
 - 용량 줄이기
- Translation, Rotation
- Offset Translation, Offset Rotation

3D 모델링을 그려보자

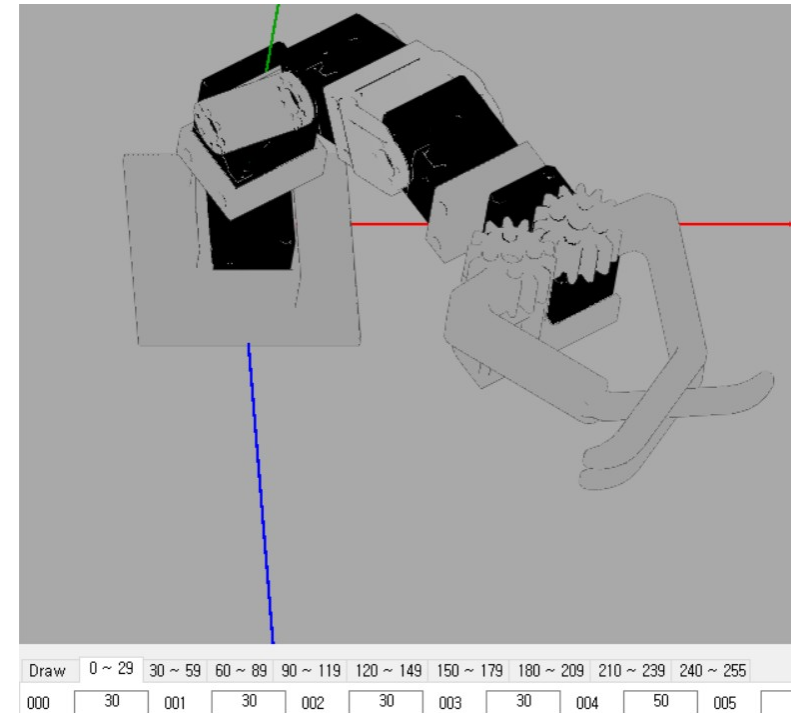
- 이것만 기억하자.
 - 엄지는 모터의 회전축
 - 말아쥐는 방향이 모터의 회전 방향
- 3D 그리기
 - A : 빨간색으로 움직이는 값
 - D : 파란색으로 움직이는 값
 - Alpha : 빨간색 축을 기준으로 손목을 바깥쪽으로 돌리는 방향의 각도값
 - Theta : 파란색 축을 기준으로 손가락을 말아쥐는 방향의 각도 값



모델링 기반 모터제어

```
[0,0,0,-90],[-1,0] // 축의 회전
[0,78,0,0],[1,0] // - Axis1
[0,0,90,90],[-1,0]
[0,0,90,0],[-1,0]
[80,0,0,0],[2,0] // - Axis2
[65,0,0,0],[3,0] // - Axis3
//[100,0,0,0],[4,0] // Axis4
```

```
//[0,0,0,90],[-1,0]
//[0,0,0,0],[5,0] // Axis5
//[50,0,135,0],[-1,0,0]
//[60,0,-105,0],[-1,0,0]
//[60,0,-50,0],[-1,0,0]
//[60,0,0,0],[-1,0,0]
//[60,0,50,0],[-1,0,0]
//[50,0,105,0],[-1,0,0]
//[0,0,0,0],[5,1] // Axis5
//[0,0,0,0],[5,1] // Axis5
//[50,0,90,0],[-1,0,0]
//[60,0,105,0],[-1,0,0]
//[60,0,50,0],[-1,0,0]
```



• 3D 모델링을 그려보자

- (+) 방향의 각도를 주었을때 움직임의 방향이 주어진 3D 모델링과 같이 움직이도록 세팅하자.(우측상단 그림 참조)
- 각 링크의 길이를 78, 80, 65 mm 로 정의한다.

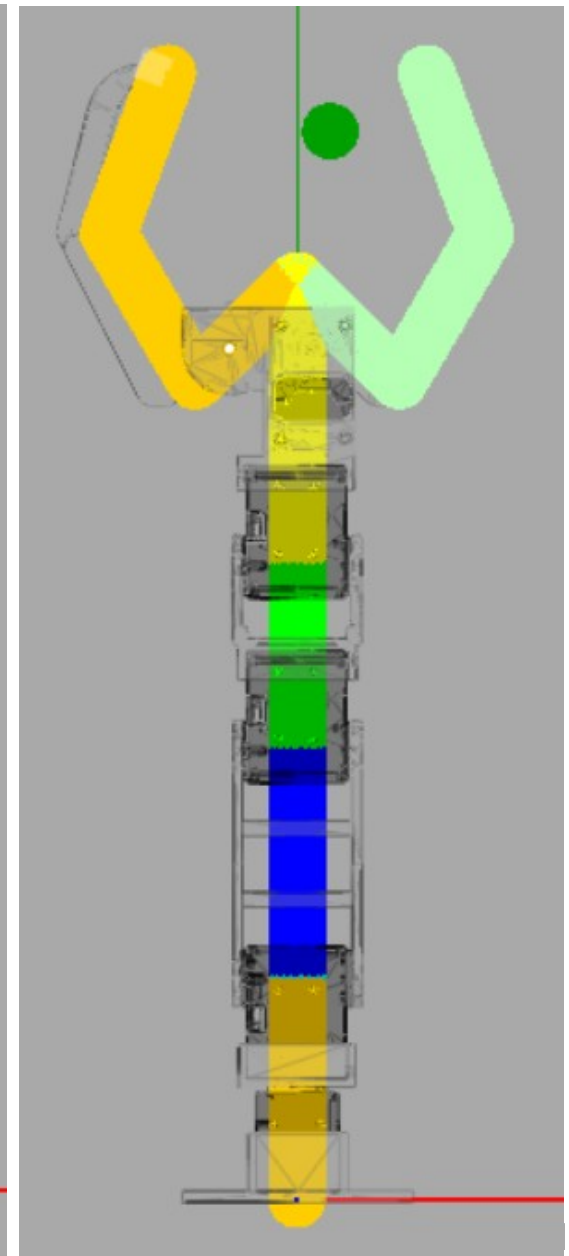
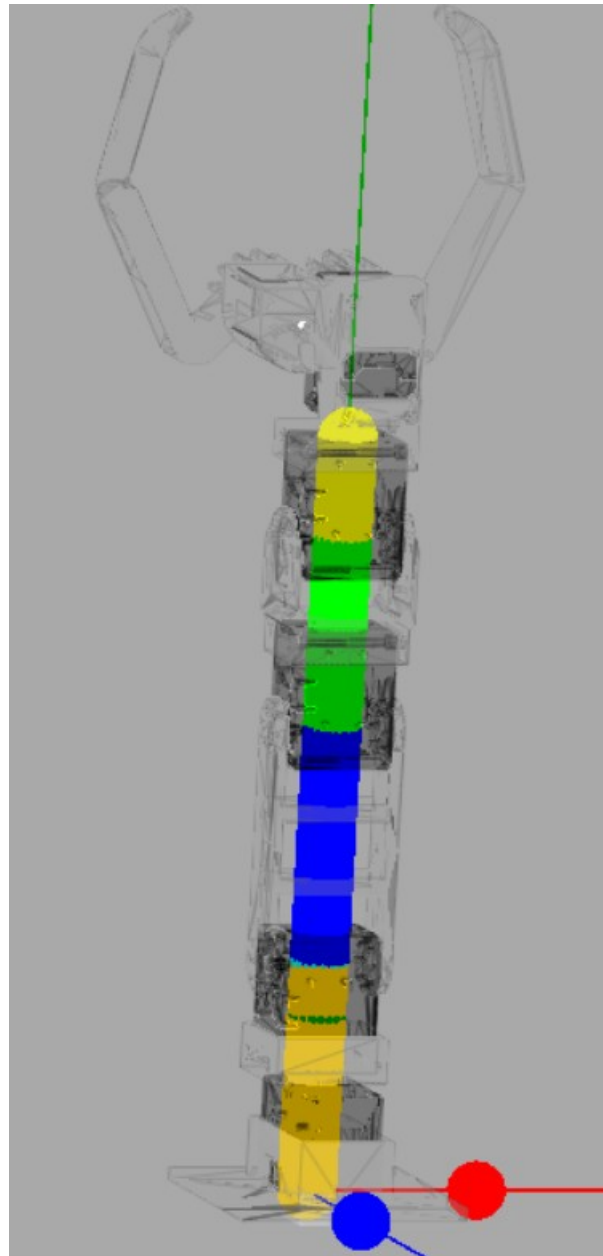
Forward / Inverse Kinematics

- 붙여넣기

- Forward

[0,0,0,-90],[-1,0] // 축의 회전
[0,78,0,0],[1,0] // - Axis1
[0,0,90,90],[-1,0]
[0,0,90,0],[-1,0]
[80,0,0,0],[2,0] // - Axis2
[65,0,0,0],[3,0] // - Axis3
//[100,0,0,0],[4,0] // Axis4

//[0,0,0,90],[-1,0]
//[0,0,0,0],[5,0] // Axis5
//[50,0,135,0],[-1,0,0]
//[60,0,-105,0],[-1,0,0]
//[60,0,-50,0],[-1,0,0]
//[60,0,0,0],[-1,0,0]
//[60,0,50,0],[-1,0,0]
//[50,0,105,0],[-1,0,0]
//[0,0,0,0],[5,1] // Axis5
//[0,0,0,0],[5,1] // Axis5
//[50,0,90,0],[-1,0,0]
//[60,0,105,0],[-1,0,0]
//[60,0,50,0],[-1,0,0]



Forward / Inverse Kinematics

- 붙여넣기

– Inverse

```
bk2=t3  
bk1=t2  
bk3=t4
```

```
fx=y-78  
fy=x  
fz=z
```

```
up=80  
dn=65
```

```
// Up 모터의 각도 알아내기  
tup=atan2(fz,fy)
```

```
// clipping //  
// 작으면 1, 같거나 크면 0  
a=0  
tupresult0 = abs(abs((((fz-a)/abs(fz-a)) + 1))/abs((((fz-a)/abs(fz-a)) + 1))-1)  
tupresult1 = tup/* pow(-1,(tupresult0 + 1)) + 90 //fz 가 앞으로 나오면 (0 보다 크면) 음의각, 뒤로가면 양의 각을 취하도록 ...  
tupresult2 = tupresult1 - 360 * abs((((tupresult1-360)/abs(tupresult1-360)) + 1))/abs((((tupresult1-360)/abs(tupresult1-360)) + 1)))  
b=-360  
tupresult3 = tupresult2 + 360 * abs(abs((((tupresult2-b)/abs(tupresult2-b)) + 1))/abs((((tupresult2-b)/abs(tupresult2-b)) + 1))-1)  
c=180  
tupresult4 = (tupresult3-360) * abs((((tupresult3-c)/abs(tupresult3-c)) + 1))/abs((((tupresult3-c)/abs(tupresult3-c)) + 1))) + (tupresult3) *  
abs(abs((((tupresult3-c)/abs(tupresult3-c)) + 1))/abs((((tupresult3-c)/abs(tupresult3-c)) + 1))) - 1)  
t1=tupresult4
```

```
// 이제 Wing 과 Down 모터의 각을 알아보자  
// Down  
d=sqrt(((fx * fx) + (fy * fy) + (fz * fz)), 2)  
tdn=acos(((up * up) + (dn * dn) - (d * d)) / (2 * up * dn))
```

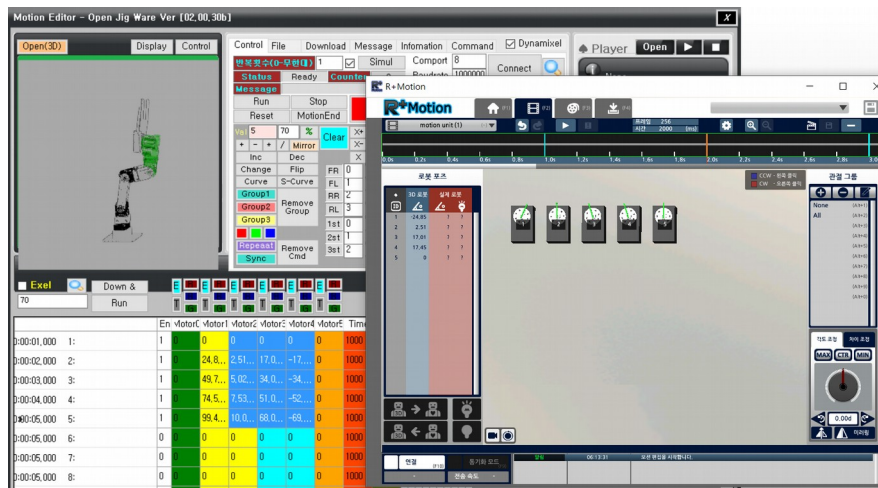
```
t3 = 180 - tdn
```

```
// Wing  
dy=sqrt(((fy * fy) + (fz * fz)), 2)  
phi=atan(dy/fx)//atan2(fx, dy)  
t2=phi - acos(((up*up) + (d*d) - (dn*dn)) / (2 * d * up))
```

```
t4=bk3+(bk1-t2)+(bk2-t3)
```

모션 툴 활용

- 모션을 만들어 저장해 보자
 - 만든 모션을 내가 만든 프로그램에서 동작해 보자.
- 팁
 - R+Motion 과의 호환
 - R+Motion 에서 오픈지그웨어 모션툴로 복사
 - R+Motion 에서 Control + C 로 복사
 - 오픈지그웨어 모션툴에서 Control+Shift+R 로 붙여넣기
 - 오픈지그웨어 모션툴 에서 R+Motion 으로 복사
 - 오픈지그웨어 모션툴에서 Control+R 로 복사
 - R+Motion 에서 Control + V 로 붙여넣기



STL Import / Export

• Import

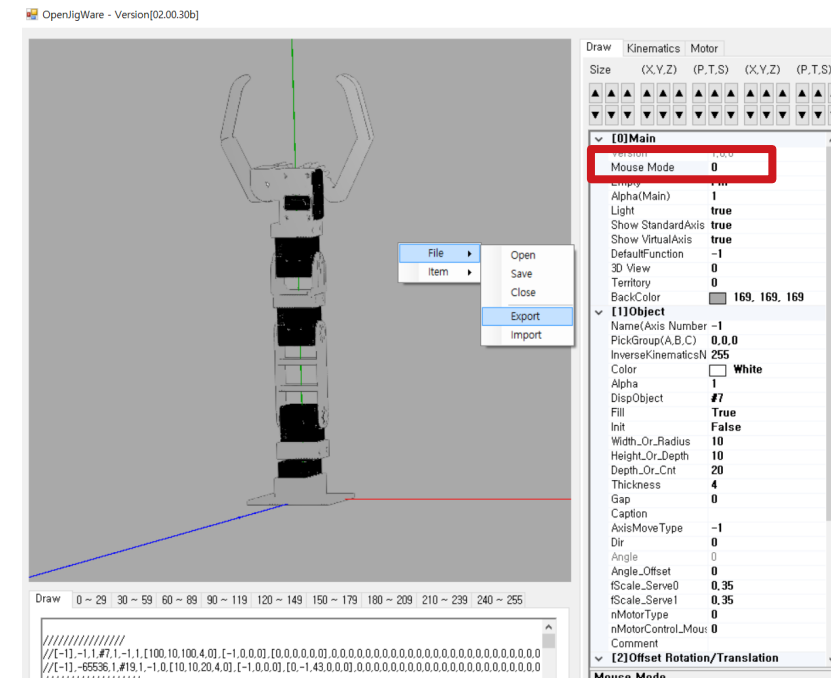
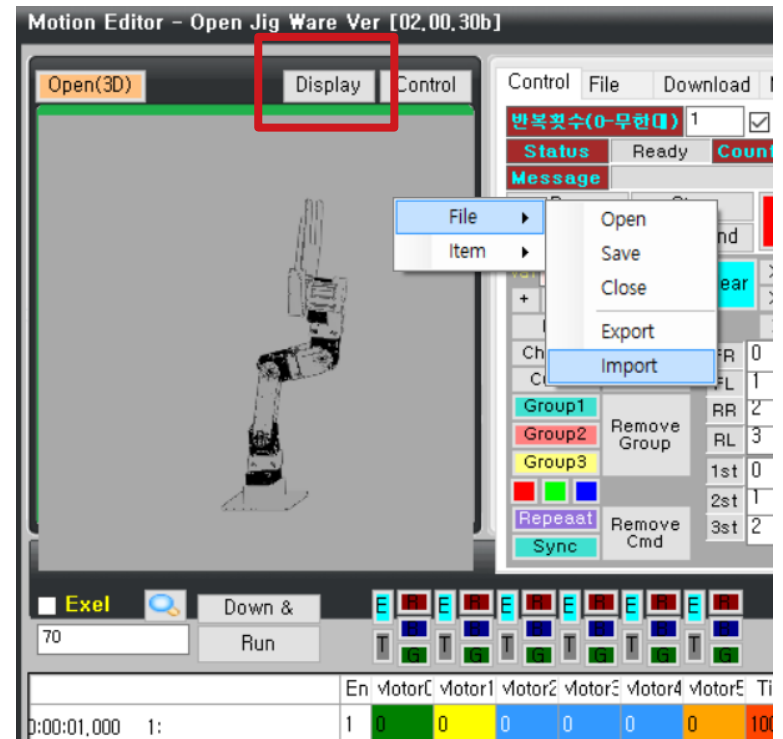
- 앓, 내가 만든 프로그램에 어떤 STL 파일이 필요한지 모른다면?

• Export

- 만들어 놓은 모델링에서 사용된 STL 파일들만 뽑아서 따로 저장하고 싶다면?

• 주의사항

- “Mouse Mode” 가 Display(“0”) 모드로 되어 있어야 한다. [Control 모드는 ‘1’]
- 코드로는 ...
 - `m_C3d.Prop_Set_Main_MouseControlMode(0);`
 - `m_C3d.Prop_Update_VirtualObject();`



오랫동안 꿈을 그리는 사람은

마침내 그 꿈을 닮아간다 .

- 앙드레 말로

감사합니다 .