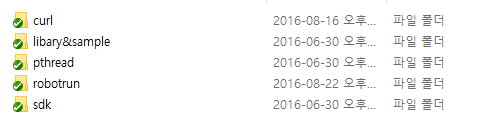
**구성**

* 1. 하드웨어
     1. 본 로봇 구동 프로그램은 아래의 하드웨어의 Spec에 맞춰 작성되었으며 다른 하드웨어의 동작 여부는 보장하지 않습니다.
        1. 상체 / 머리 관절부 MX-28, AX-12A
        2. 구동부 MX-106, AX-18A (변경)
        3. 디스플레이 wave share e-ink display
  2. 소프트웨어
     1. 로봇 구동 프로그램은 다음 OS에서 구동 가능하도록 제작 되었으며 다른 OS에서의 동작 여부는 보장하지 않습니다.
        1. 윈도우즈 (Windows 7 / 10)
        2. Raspberry PI (jessie / jessie lite)
  3. 제공되는 데이터
     1. 3DMax Model Data (로봇 에니메이션 제작을 위하여 만든 Dummy Data)
     2. 3DMax Animation Exporter (maxexporter.ms : 3D Max에서 제작한 에니메이션을 export 하기 위한 Exporter)
     3. 에니메이션 샘플 데이터 (XML포멧)
     4. 로붓 구동 프로그램 소스코드 / Exporter 소스코드

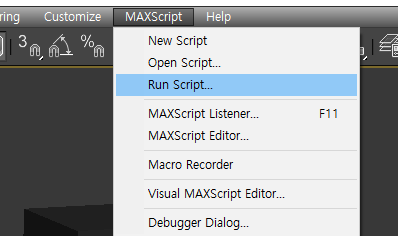
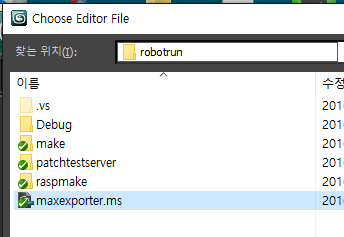
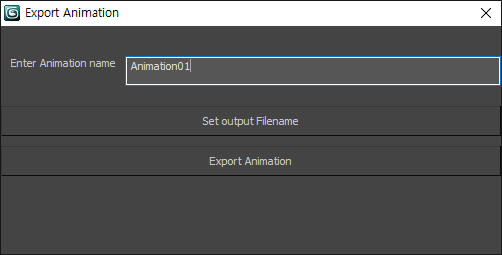
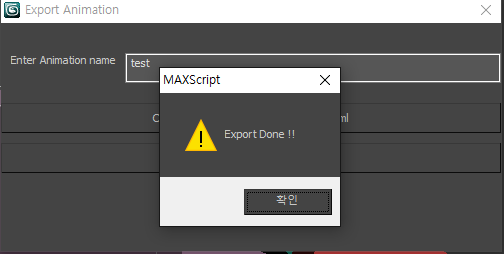
**소스코드 빌드 방법**

* 1. 아래 그림의 폴더가 정상적으로 있는가 확인
     1. 
  2. 윈도우즈
     1. Visual Studio (2012 / 2015)를 사용하며 C++ 코드로 작성되어 있슴.
        1. Visual Studio 해당 버전에 맞는 Solution File을 연후 빌드 한다.
           1. \robotrun\robotrun\_2012.sln (Visual Studio 2012용 Solution)
           2. \robotrun\robotrun\_2015.sln (Visual Studio 2015용 Solution)
  3. Raspberry PI
     1. 아래의 빌드 순서를 따라 빌드 준비를 한다. (아래의 순서는 root 권한으로 수행)
        1. sudo apt-get install build-essential
        2. libary&sample\DynamixelSDK-master\c++\build\linux32 에서 make 수행
        3. 빌드 성공시 sudo make install 수행
     2. robotrun\raspmake 에서 make 수행

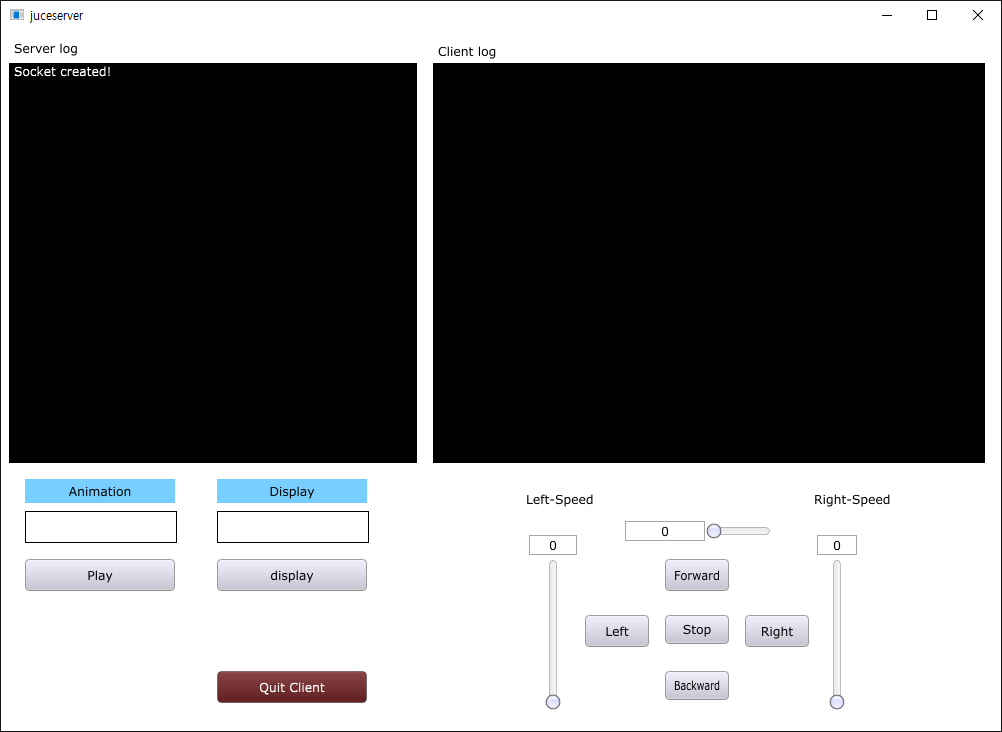
**Robot 설정**

* 1. setup.xml (해당 항목에 대하여 수정가능, 내용 추가 / 삭제 금지)
     1. 각 구동 모터 / 디스플레이 / 하드웨어 설정을 위한 설정 파일
        1. jointdevicename / displaydevicename : 윈도우즈용 시리얼 포트
        2. linuxjointdevicename / linuxdisplaydevicename : 라즈베리 시리얼 포트
        3. jointbaudrate / linuxjointbaudrate : 윈도우 / 라즈베리 시리얼 속도
        4. displaybaudrate : 디스플레이용 시리얼 속도
        5. serveraddress : 커멘더 서버 주소 (PORT는 8888 사용)
           1. 현재 테스트용 서버로 테스트 가능
        6. parts (각 디바이스 설정)
           1. type : joint (관절) , wheel (바퀴) , display (디스플레이)
           2. id : device ID
           3. torque : 1 토크 있슴 (joint / wheel)
           4. torquelimit : 토크 제한 값 (wheel)
           5. cwlimit : 시계방향 제한각 (joint)
           6. ccwlimit : 반 시계방향 제한각 (joint)
           7. initpos : 초기 위치 (joint)
           8. p-param : PID , P 값 (joint / wheel)
           9. i-param  : PID , I 값 (joint / wheel)
           10. d-param  : PID , D 값 (joint / wheel)

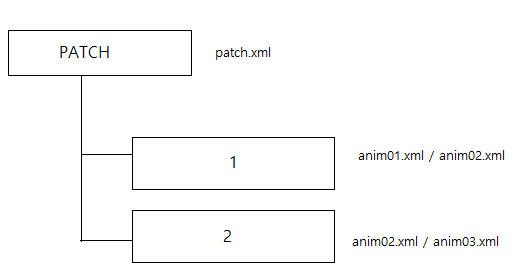
**에니메이션 데이터 제작 / 익스포트**

* 1. 제공된 3DMax Dummy Model 데이터를 이용하여 에니메이션 데이터를 제작합니다. 에니메이션 제작에는 다음과 같은 제한을 갖고 있습니다.
     1. 에니메이션 제작에 실수를 방지하기 위하여 Dummy Model의 각 관절은 제한 각을 설정하였습니다.
        1. setup.xml에서도  Dummy Model에서 설정한 값을 기준으로 3DMax Dummy Model에도 제한을 설정했습니다.
           1. 두 항목의 제한 각이 같아야 같은 결과물을 기대 할 수 있습니다.
        2. 만일 이 값의 수정이 필요하면 아래의 기타 항목 참고하시기 바랍니다.
        3. **원하는 초기 값을 전달해 주시면 조정하여 전달하겠습니다.**
     2. **자연스러운 에니메이션을 위하여 에니메이션 제작은 기본 포즈에서 시작하여 에니메이션 마지막에 기본 포즈로 복귀하여야 합니다**.
        1. 아니여도 상관은 없지만 동작 연결 시 자연스럽지 않습니다.
        2. 기본 포즈에 대하여 정해준 내용이 없기 때문에 임의의 포즈로 설정 하였습니다.
     3. 너무 빠른 동작 전환은 재생할 수 없습니다.
        1. 관절에 사용된 모터의 성능의 한계로 인하여 너무 빠른 변화의 에니메이션은 재생이 불가 합니다.
        2. 117.07 RPM이 최대 값이므로 초당 702도(Degree) 보다 빠르게 움직일 수 없습니다. (MX-28기준)
  2. 에니메이션 익스포트
     1. 제작이 완료된 에니메이션을 실제 사용하기 위하여 Export 합니다.
     2. 3DMax 메뉴중 Run Script을 클릭
        1. 
     3. 제공된 익스포터 선택
        1. 
     4. 에니메이션 이름을 입력
        1. 
     5. Set Output Filename에서 저장할 파일 이름 선택(확장자는 xml , 예 : Animation01.xml)
     6. Export Animation 클릭하여 Export 수행
        1. 

**Robot 운용 / 테스트**

* 1. 제공된 테스트 서버를 시작한다.
     1. 
     2. 테스트 서버에서는 다음의 기능테스트가 가능합니다
        1. 에니메이션 플레이
        2. E-Paper 그림 변경
        3. 구동 부 컨트롤 (전/후/좌회전/우회전)
        4. Robot 로그 모니터링
  2. 윈도우용 client
     1. Robotclient 폴더를 연다
     2. PC에 Robot연결 확인 후 해당 폴더로 이동하여 robotrun\_2015.exe 실행
  3. Raspberry PI
     1. /home/pi/robot 폴더로 이동
     2. 라즈베리에 Robot연결 확인 후 해당 폴더로 이동하여 ./start.sh 실행
  4. **좌측 Edit box**에 에니메이션 이름 넣은후 play 버튼으로 해당 에니메이션 플레이
  5. **우측 Edit box** 디스플레이할 그림의 alias 넣은후 버튼으로 그림 표시
  6. 오른쪽 컨트롤들을 이용하여 이동 테스트 가능
     1. **왼쪽/오른쪽 슬라이드**로 회전시 각 바퀴의 속도 제어
     2. **중앙 슬라이드**를 조정하여 전진/후진 속도 제어

**온라인 패치 / 패치 데이터 작성**

* 1. 패치 서버가 운용될 IP Address (Domain name) 준비
  2. Http 서버 준비 (Apache / NginX ...) / 패치 서버 폴더 구성
  3. 패치 작동 순서
     1. local 머신의 패치 정보를 갖고 있는 patch.xml 데이를 읽고 파싱 (각 패치 항목의 목록과 버전 정보) / patch.xml은 없어도 무방 (패치시 다운로드 함).
     2. 패치 서버에 접속하여 서버에 있는 patch/patch.xml 을 읽고 파싱
     3. 서버의 patch.xml 정보와 local patch.xml을 비교하여 다운로드 해야할 목록 구성
     4. 다운로드 할 목록 다운로드후 local 머신에 저장
     5. 다운 로드 완료하면 local 머신의 patch.xml 을 최신 정보로 업데이트 후 local 에 저장
     6. 다운로드 한 데이터 로드
  4. 패치 서버의 폴더 구성
     1. 
  5. patch.xml 내용
     1. patch.xml의 내용은 다음과 같다
        1. version : 파일의 버전 (local의 버전 보다 서버 버전이 높으면 다운로드 받는다)
        2. name : 파일 이름
        3. path : 다운로드 경로 (이름 포함 Full path)
        4. crc : 파일의 CRC32 값 (검증용)
  6. patchserver 폴더내의 Sample 내용을 참고

**Animation.xml 설명**

<data>

<animation filename="test01.xml"/>

<animation filename="test02.xml"/>

<animation filename="yes.xml"/>

<animation filename="no.xml"/>

</data>

* + 1. animation.xml은 로봇에서 사용할 모든 에니메이션 파일의 목록이 정의 되어있습니다
    2. 로봇이 구동하면 animation.xml 파일을 읽어 나열되어 있는 모든 에니메이션을 로딩합니다
    3. 해당 에니메이션 (ex : yes.xml)안의 내용 중 name 에 해당되는 이름으로 에니메이션을 재생합니다.

**display.xml 설명**

<data>

<picture name="test1" left="EYE1.BMP" right="EYE1.BMP"/>

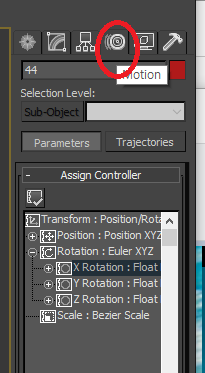
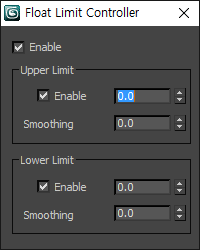
<picture name="test2" left="EYE1.BMP" right="EYE2.BMP"/>

<picture name="test3" left="EYE2.BMP" right="EYE2.BMP"/>

</data>

* + - 1. Display.xml에는 로봇의 e-paper에 표시될 그림의 목록이 정의 되어있습니다
      2. e-paper는 좌,우 두개이기 때문에 각각에 표시될 파일명이 name으로 묶여 있고 이 name기반으로 e-paper에 그림을 표시합니다
      3. 예: test2를 표시하라 🡪 왼쪽 : EYE1.BMP 오른쪽: EYE2.BMP 표시
      4. 해당 그림파일들은 미리 e-paper의 nand flash에 로딩되어 있어야 합니다.

**기타**

* 1. 제한각 설정에 따른 3DMax데이터 설정
     1. 제한각 변경은 다음과 같이 수정합니다.
        1. 
        2. 해당 축을 더블클릭하여 제한각을 설정한다.
        3. 
        4. 수정한 값을 setup.xml의 해당 관절에 적용한다.
  2. 현재 에니메이션 플레이중인 경우 다른 에니메이션 플레이 명령이 수신되면 현재 에니메이션 플레이 종료 후 플레이 됩니다.