OJW MENUAL

**목차**

I. 소개

1. Open Jig Ware 란?

2. 어떤 로봇들이 가능한가?

3. 개발동기

4. 개발환경

5. 특징

II. 설치

1. Download

2. 사용 예

III. 일반

1. InputBox

2. Message

3. Convert

4. Math

5. Encrypt

6. File

7. Timer

8. Register

IV. Kinematics

V. 그리기

1. 2D 그리기

2. 3D 그리기

VI. 통신

1. Serial

2. Socket

VII. 모터제어

1. Herculex

2. MX28

3. DC2408

VIII. Mouse

IX. 시간 그래프

X. Struct

XI. Streaming

XII. System

XIII. 응용

**그림목차**

[그림 I-1]

[그림 I-2]

[그림 I-3]

[그림 III-1]

[그림 III-2]

[그림 III-3]

[그림 III-4]

[그림 III-5]

[그림 III-6]

[그림 III-7]

[그림 III-8]

[그림 III-9]

[그림 III-10]

[그림 III-11]

[그림 III-12]

[그림 III-13]

[그림 III-14]

[그림 III-15]

[그림 III-16]

[그림 III-17]

[그림 III-18]

[그림 III-19]

[그림 III-20]

[그림 III-21]

[그림 III-22]

[그림 III-23]

[그림 III-24]

[그림 III-25]

[그림 III-26]

[그림 III-27]

[그림 III-28]

[그림 III-29]

[그림 VI-1]

[그림 VI-2]

[그림 VI-3]

[그림 VI-4]

[그림 VI-5]

[그림 IX-1]

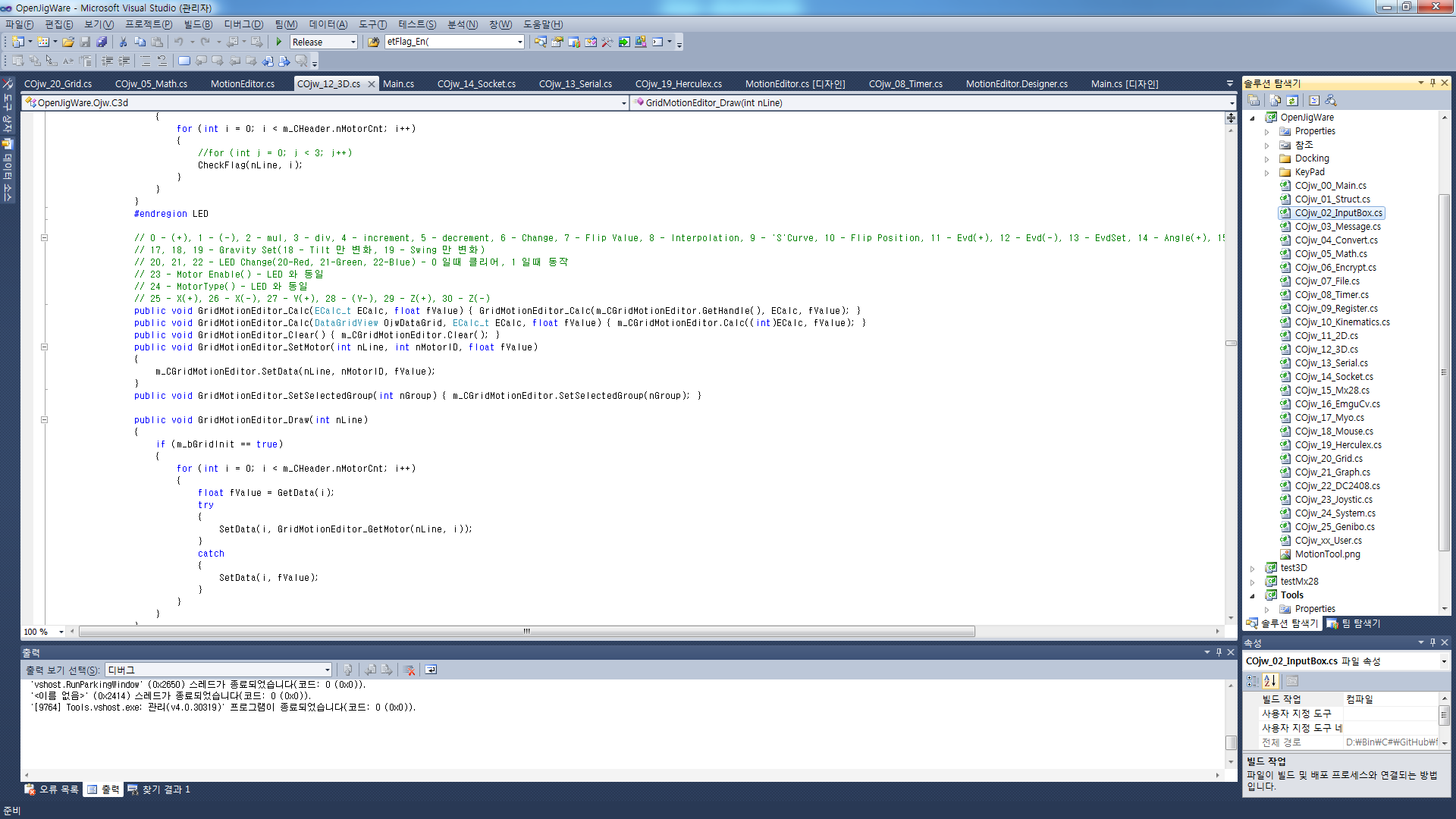
[그림 IX-2]

[그림 X-1]

[그림 X-2]

Open Jig Ware

1. 소개
   1. Open Jig Ware 란?
      1. 개발용 DLL 로 미들웨어의 역할을 수행.
      2. 장비 혹은 지능형 모션 개발자와 교육용에 타깃.
         1. 교육용
            1. 선생님과 학생이 구분해서 사용 가능

선생님 : 3D 로 로봇 만들기, 수식 입력, 모터 정의 및 상관 관계 입력

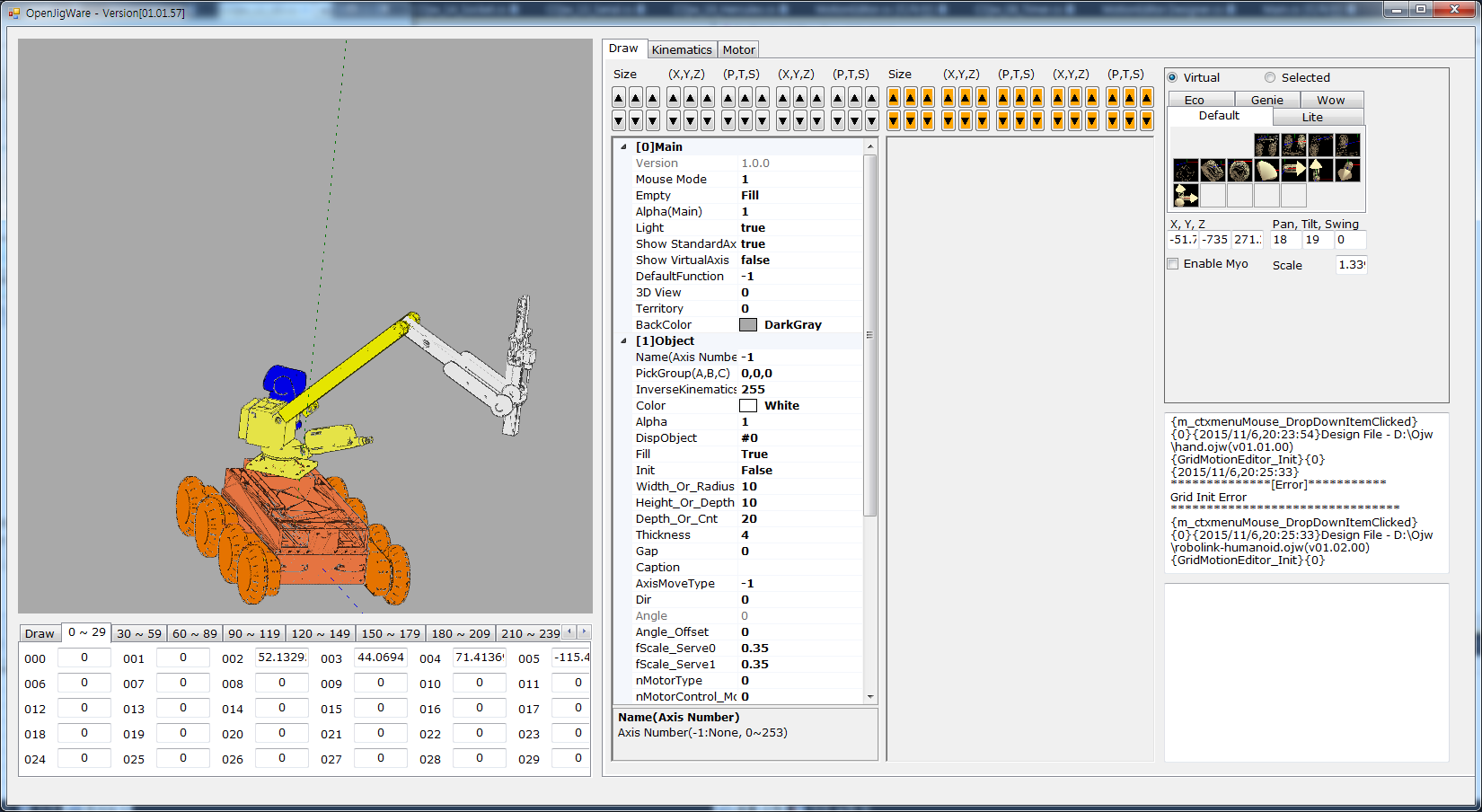
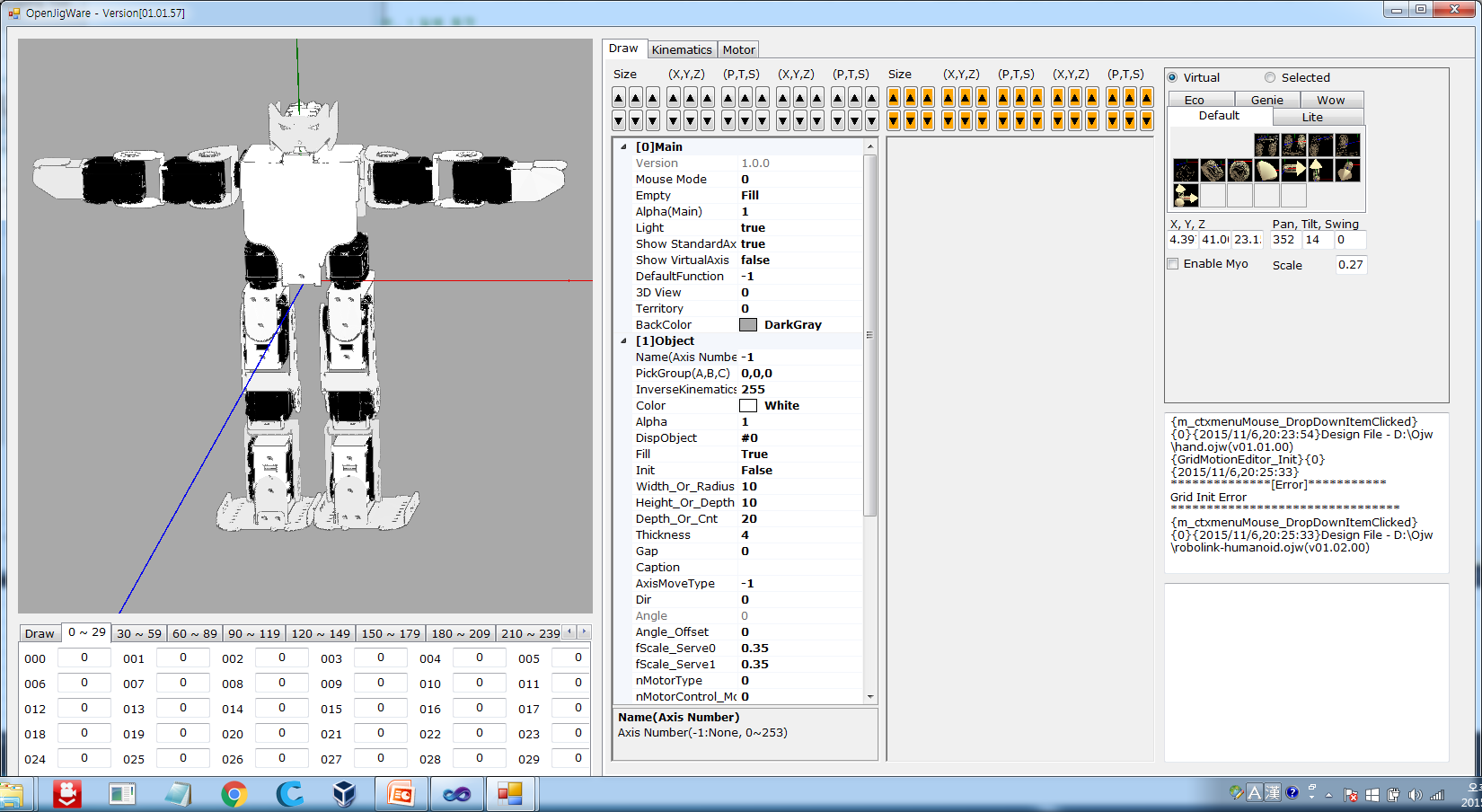
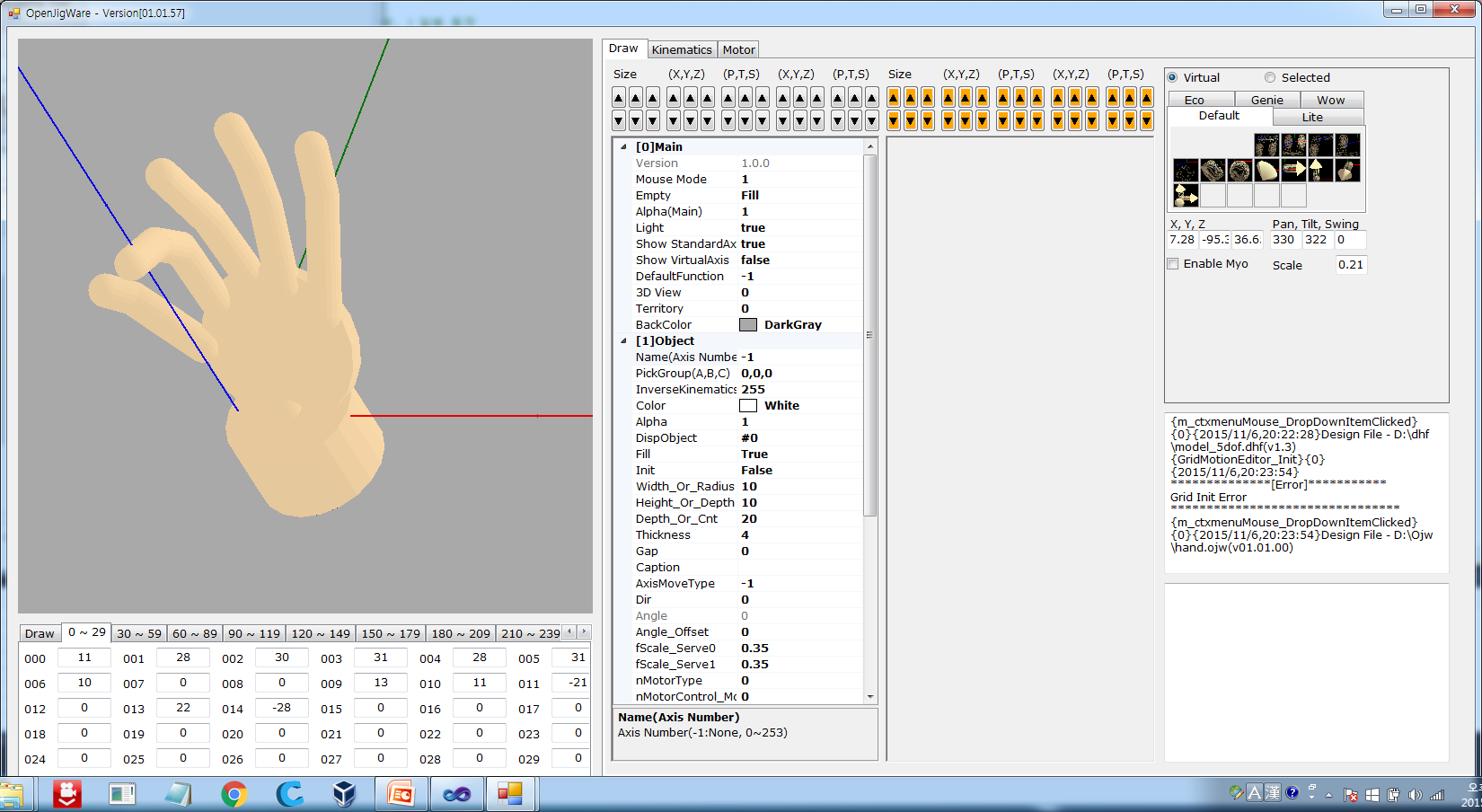
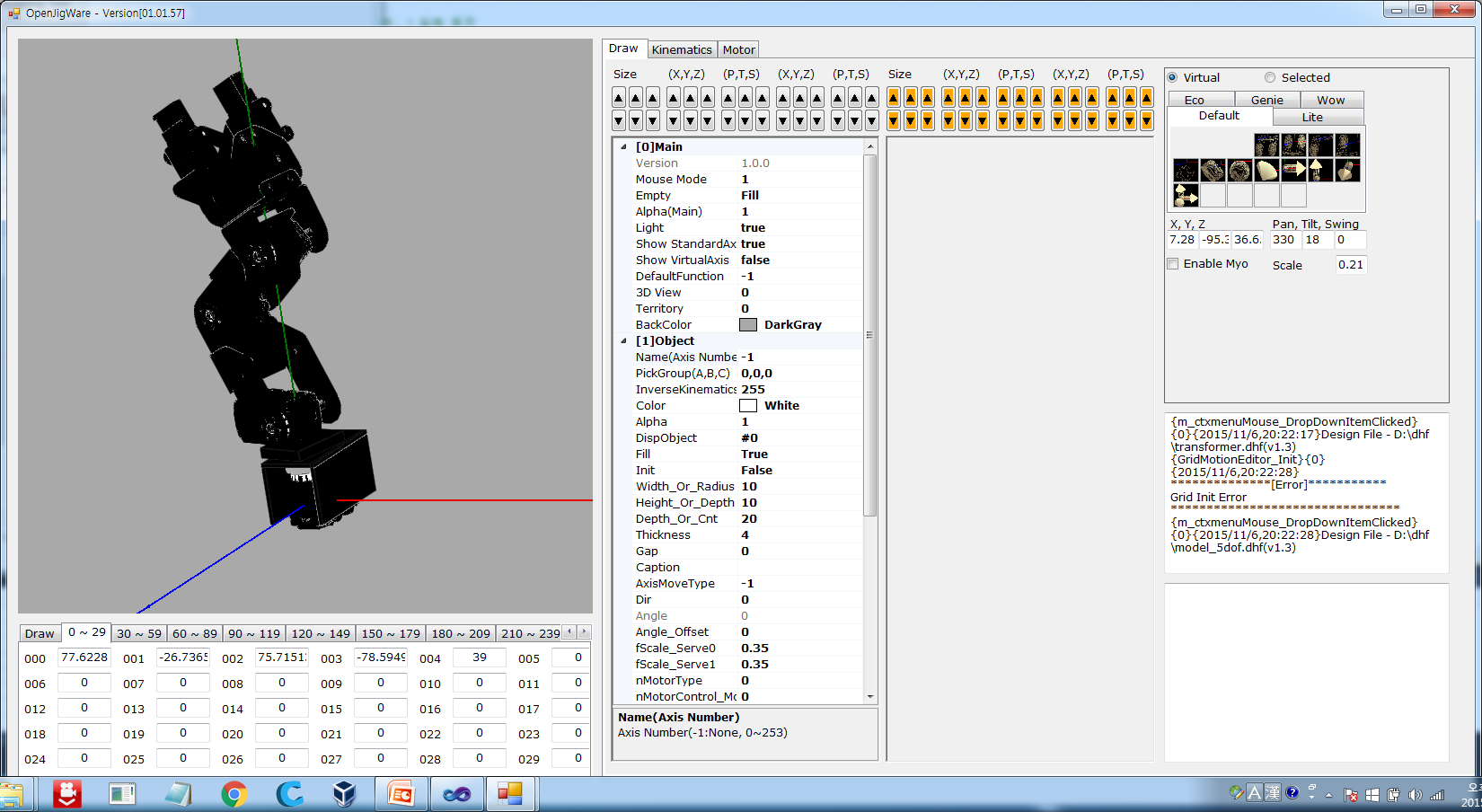
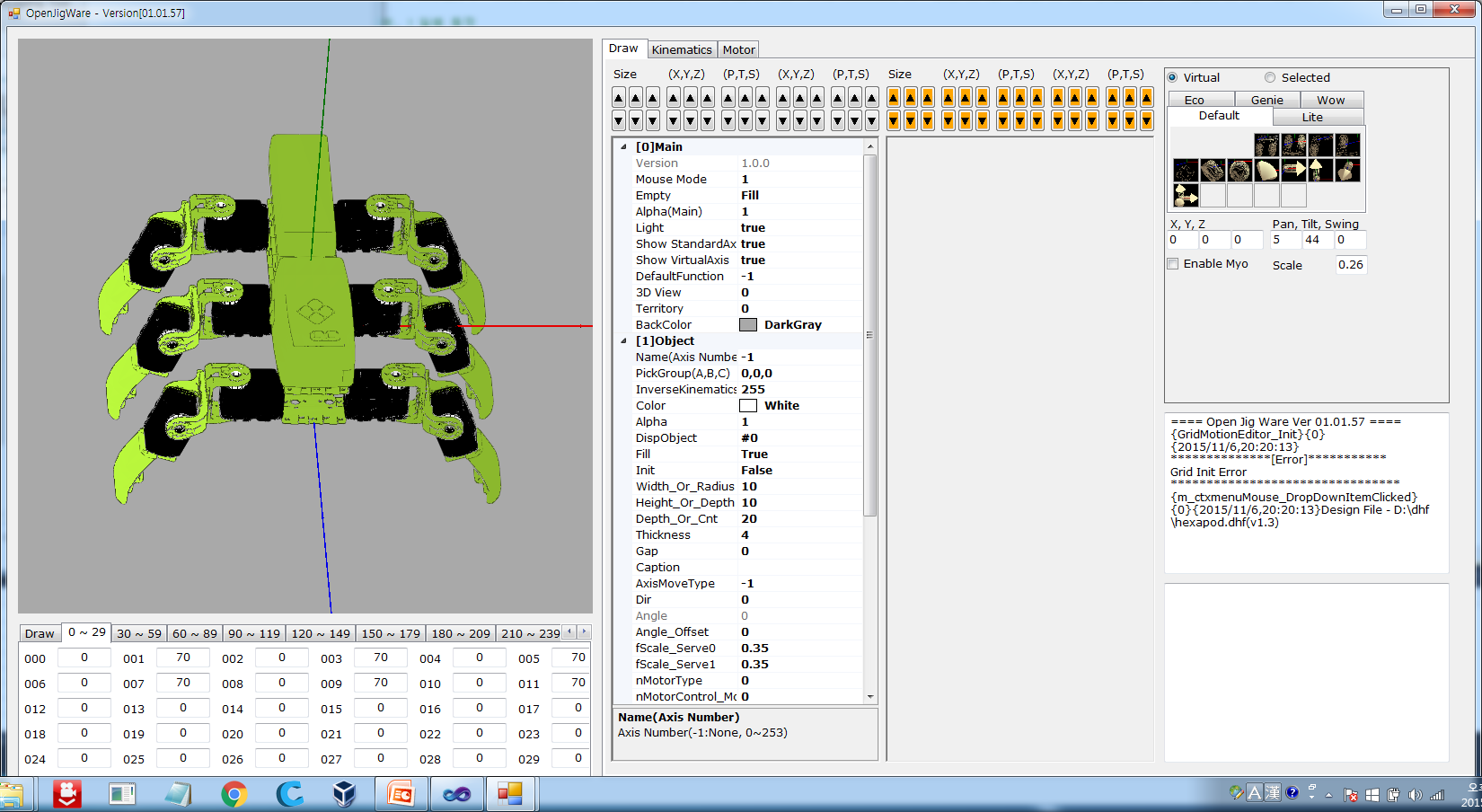
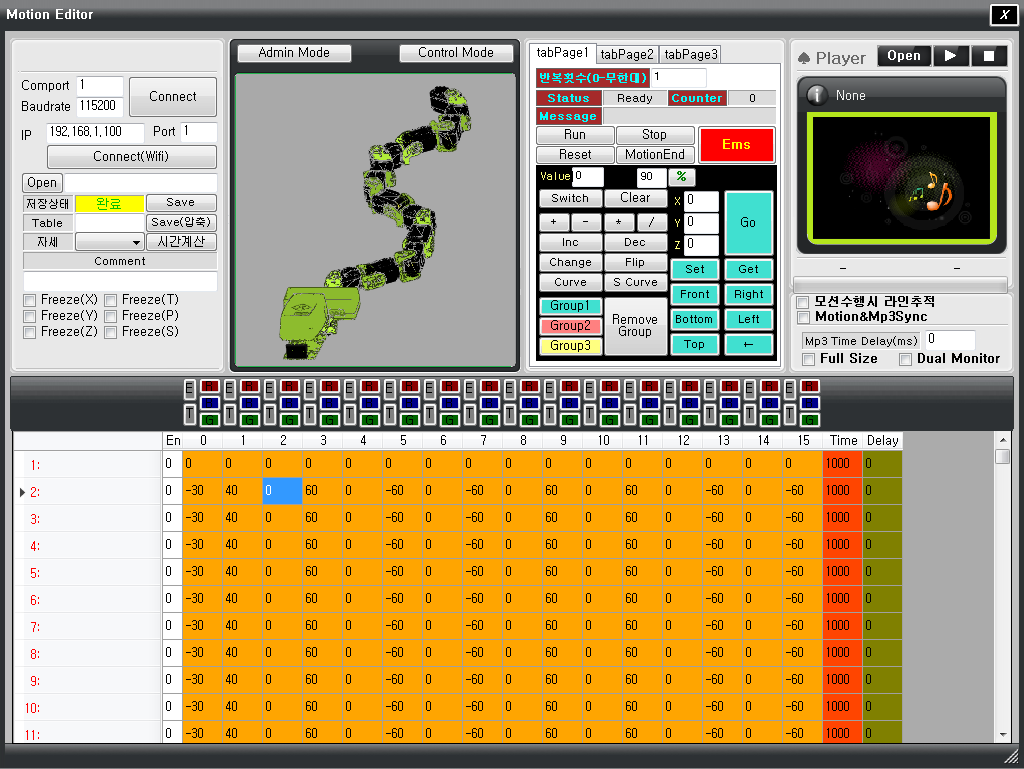
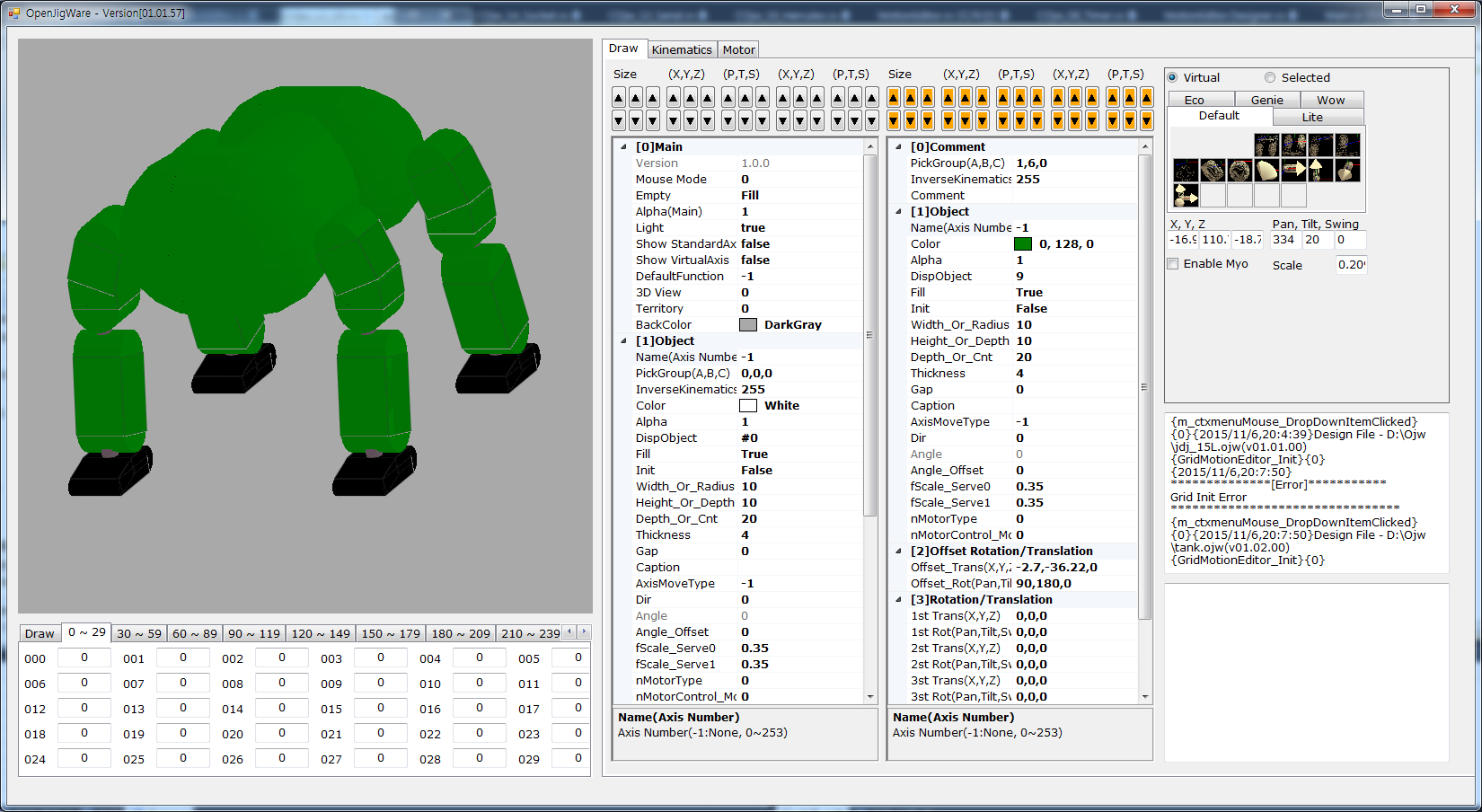
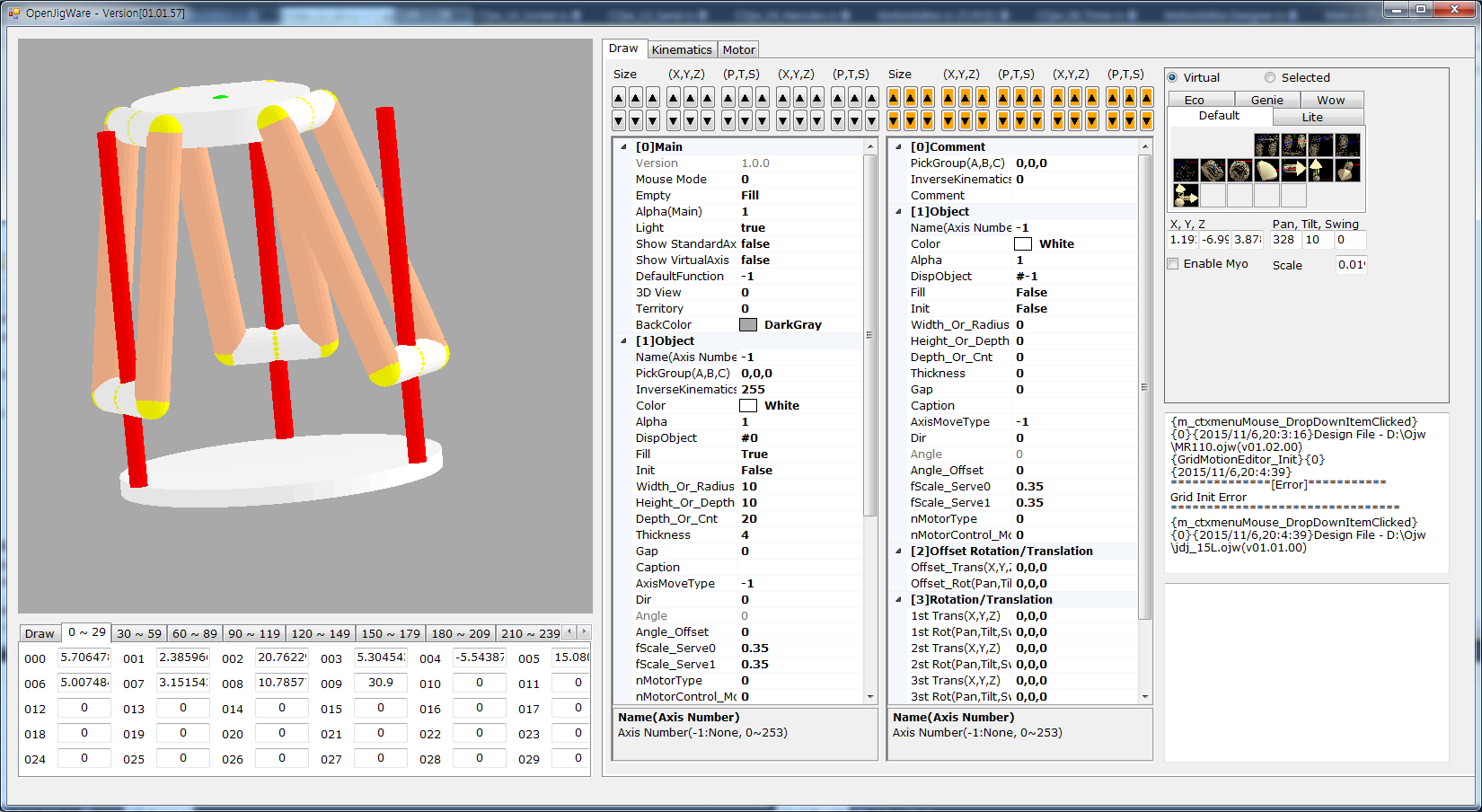
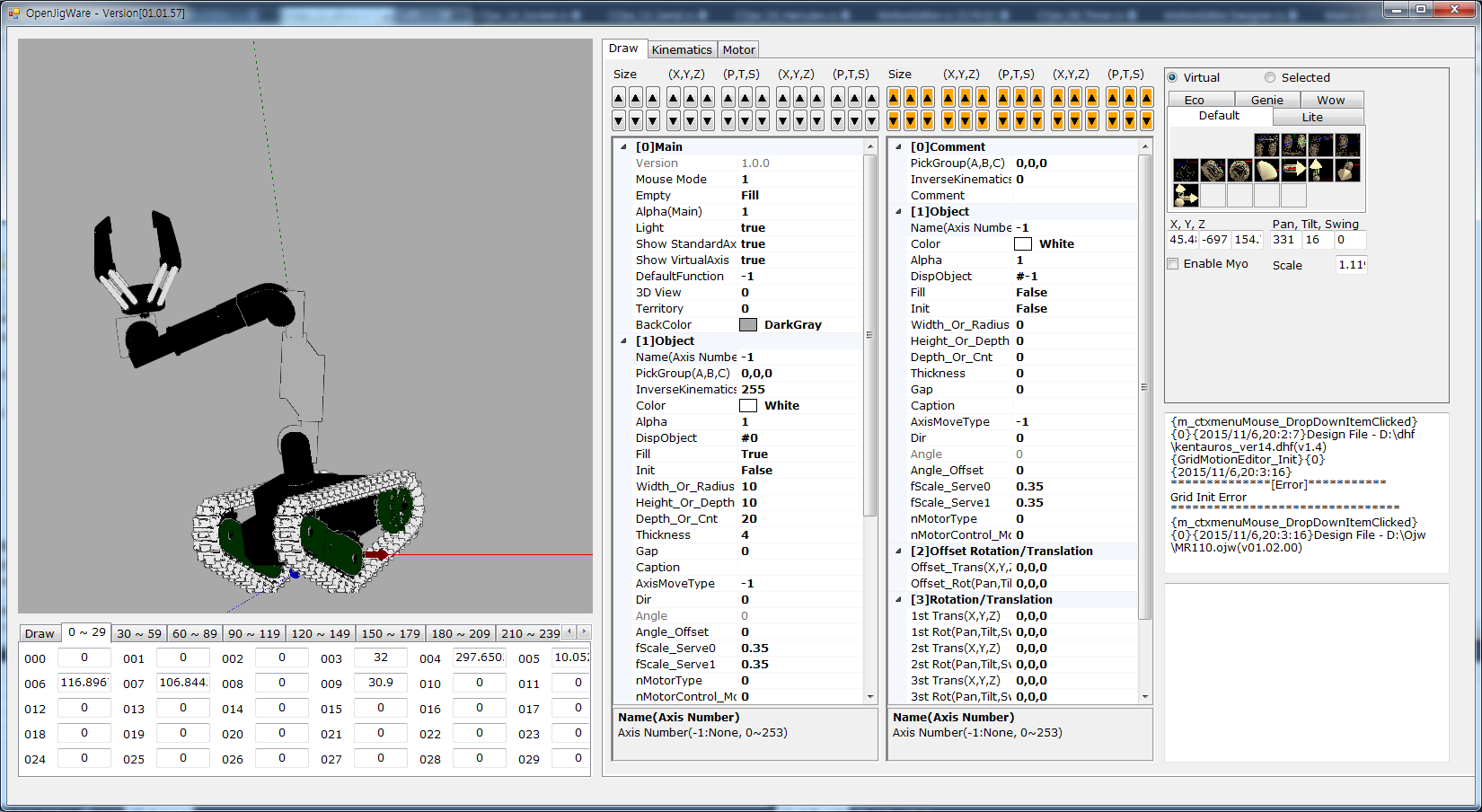
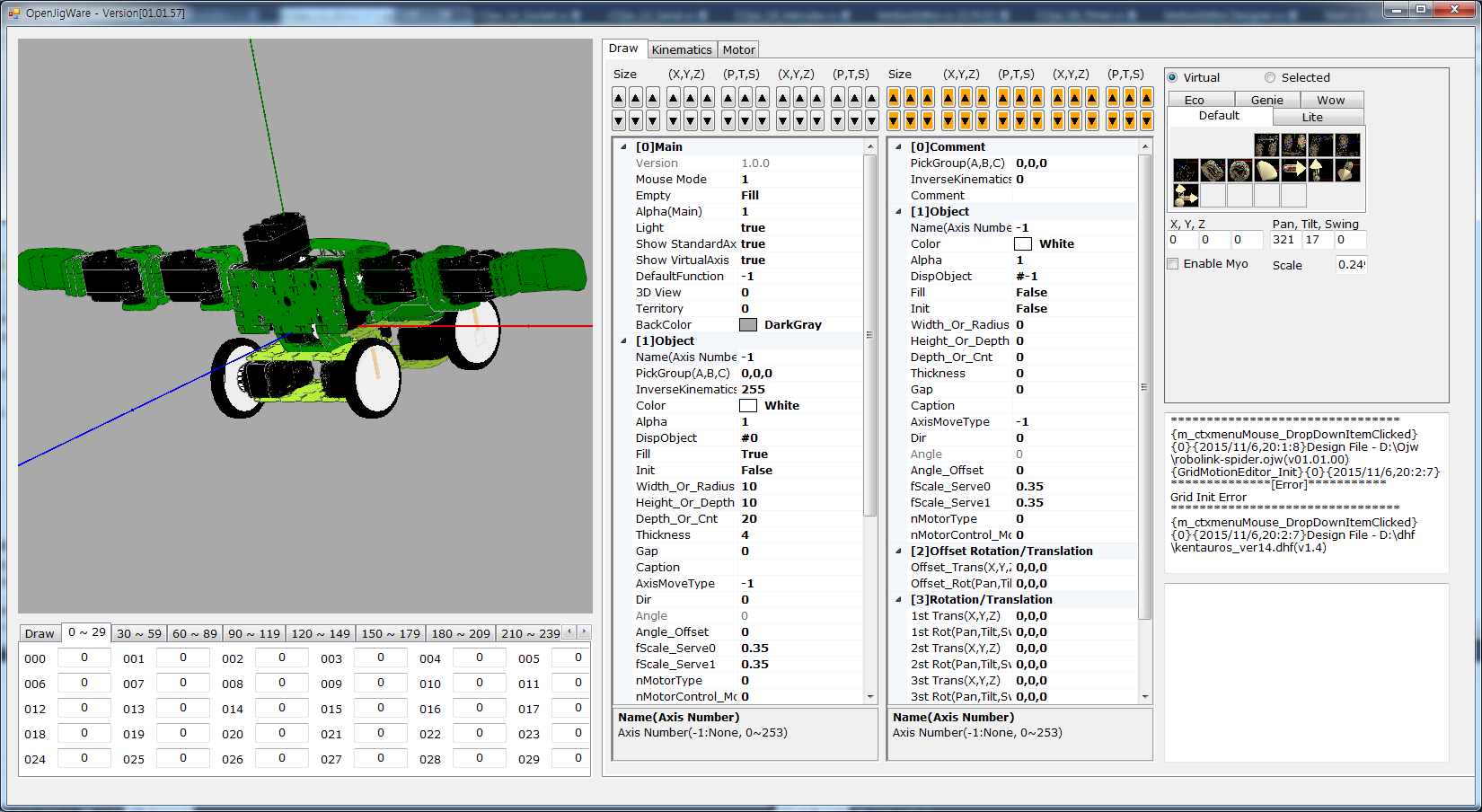
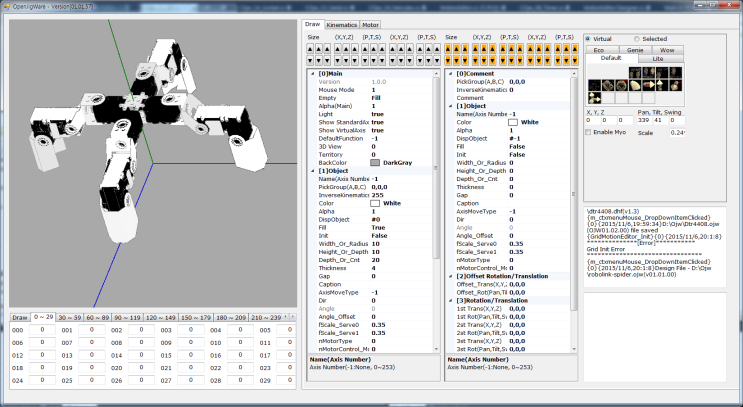
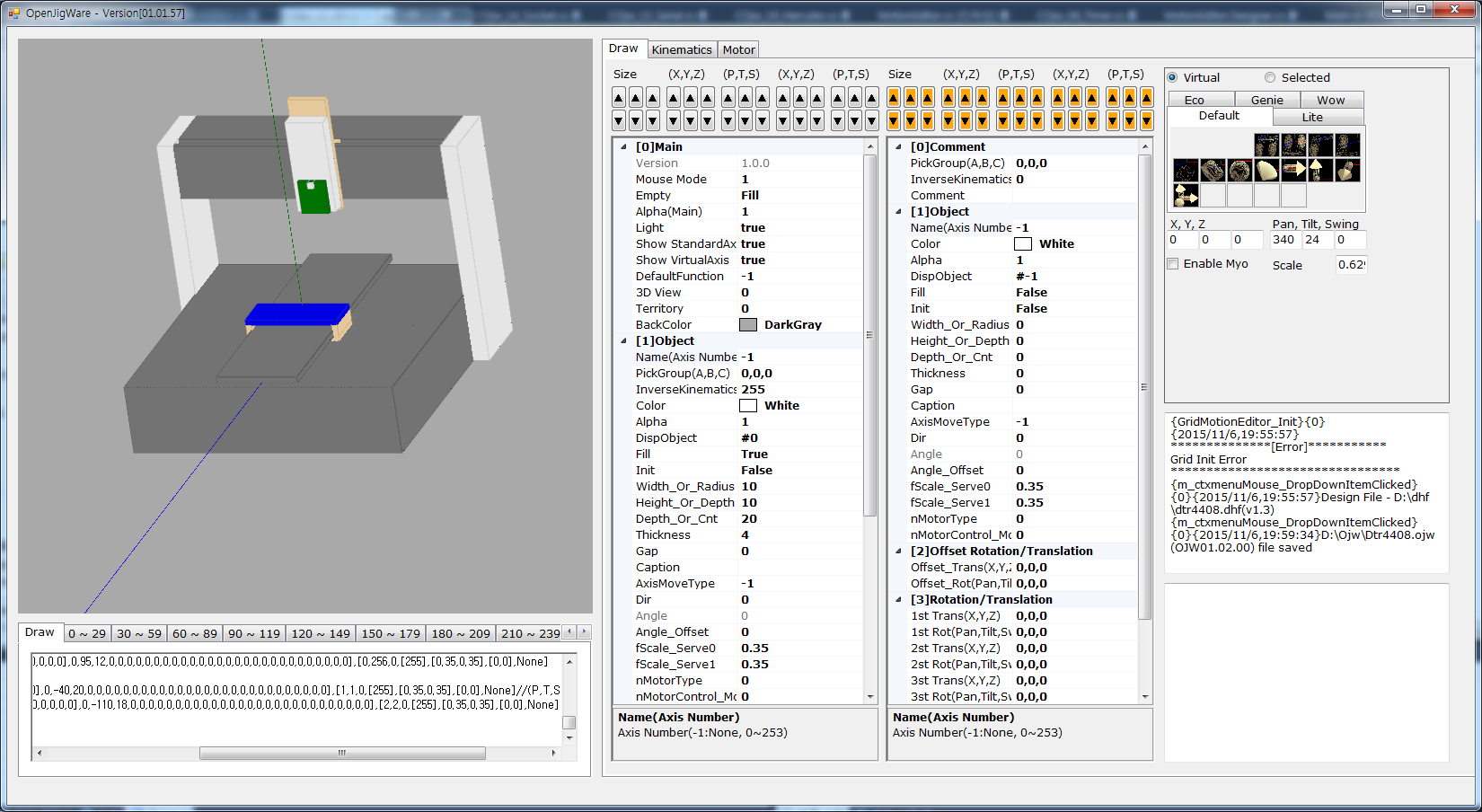
다양한 로봇을 학생들에게 접하게 할 수 있고 로봇의 수식을 학생들에게 실제적으로 적용해서 보일 수 있다.

클릭하는 로봇의 부품에 따라 회전, 직교로 동작할지 Forward/Inverse 등의 수식이 적용될지, 혹은 클릭이 되지 않게 적용할지를 선택할 수 있다.

Forward Kinematics 를 직접 눈으로 예를 들어 보이면서 설명이 가능하다.

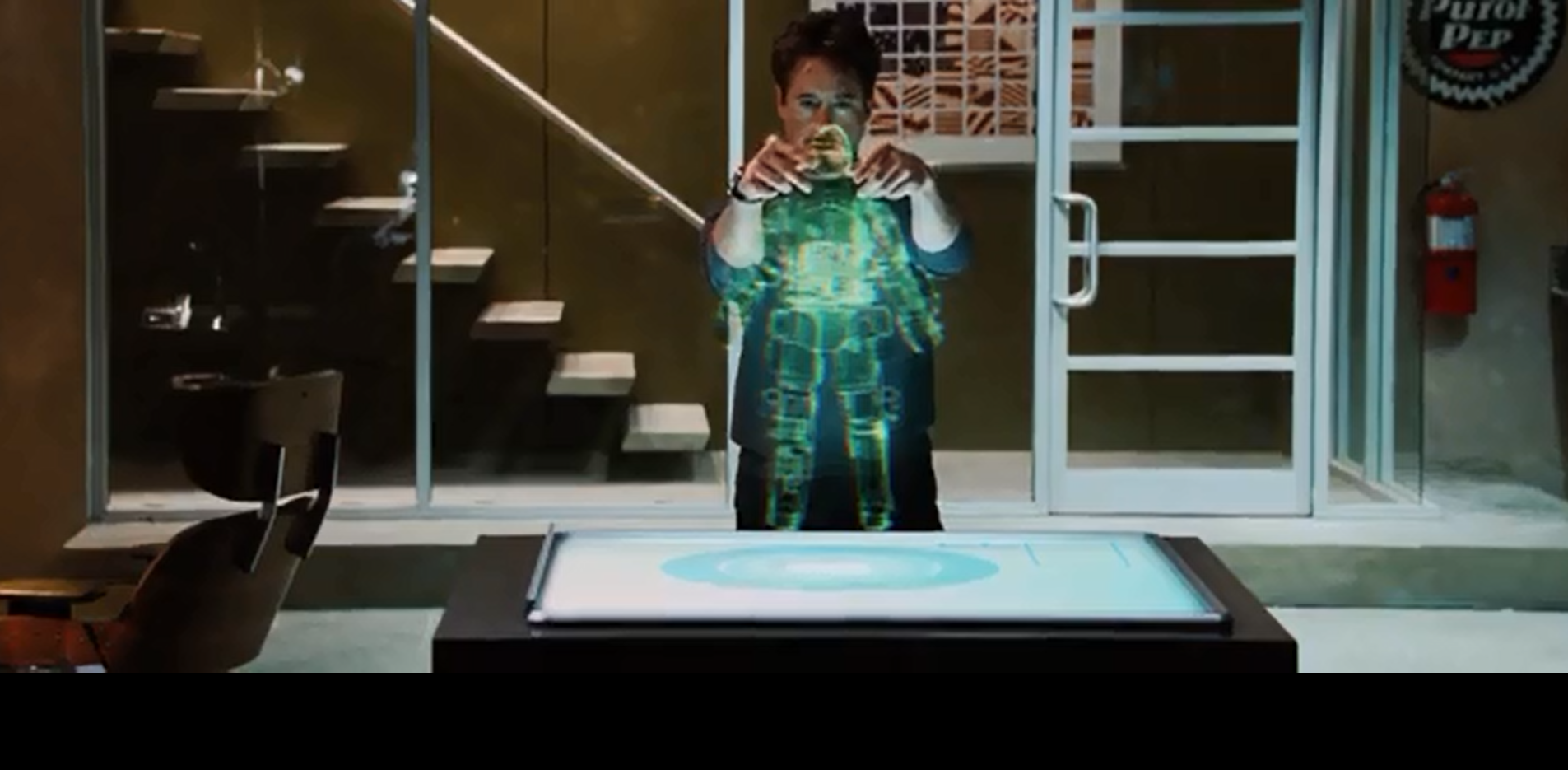
[그림 I-1]

* 1. 어떤 로봇들이 가능한가?



[그림 I-2]

* 1. 개발동기
     1. 어느 날 던져진 화두
        1. 모든 로봇에 적용될 수 있는 무언가가 없다.
        2. 휴머노이드건 바퀴형 로봇이건 직교좌표로봇이건 모든 로봇에 적용될 수 있는 것이 있다면?
        3. 로봇은 수식으로 이루어진 수식 덩어리. 그렇다면 그 수식만 바꿔주면 될거 아닌가?
        4. Kinematics 를 교육하기란 너무 어렵고 또한 이걸 위해 실물을 가져다 놓기엔 비용이 너무 많이 든다.



[그림 I-3]

* 1. 개발환경
     1. Microsoft Visual studio 2010 & C#(with Tao frame work : OpenGL)
  2. 특징
     1. 최초 개발은 모션 편집용 프로그램으로 작성하여 모션 제작 및 튜닝에 최적화 되어 있으며 Jig 및 장비 개발에 최적화 되어 있다.
     2. 3D 모델링의 빠른 구현으로 복잡한 형태도 숙련자의 경우 2시간 정도면 구현 완료
        1. 가상에서 보면서 조립하면 끝.
     3. 기타 세부적인 적들을 정의하는데 시간이 걸림
        1. 수식 / 실제 사용할 모터 기어비 등 자신이 넣고 싶은 정보가 많은 수록 시간이 걸린다.
     4. 제어기능의 빠른 구현
        1. 모델링이 완성되어 있는 경우 전체 코드 5줄만 추가하면 준비 완료.
     5. 교육적 내용 포함
        1. D-H Notation 파라미터만 입력하면 3D 모델링 및 수식이 자동으로 생성된다.
           1. 자동으로 생성되는 3D 모델링 – 실제 사용시는 보이지 않는 스켈레톤 모델로 이걸 활용하여 세부 그림을 완성해 나가면 쉽다.
           2. D-H 수식 자동완성

파라미터만 입력하면 수식의 현 결과를 3D 와 행렬 함수로 바로 확인이 가능하다.

확인된 행렬 함수는 자동으로 내부에서 사용되고 있으므로 프로그래밍적으로 필요는 없고 따로 논문이나 기타 문서등에 활용 수식으로 첨부하면 논문이나 개발문서 작성에 좀더 빠르게 적용 가능하다.

1. 설치
   1. Download
      1. 아래의 링크에서 소스 포함 다운로드가 가능하다.
         1. [Https://github.com/ojw5014/OpenJigWare](https://github.com/ojw5014/OpenJigWare)
         2. 다운받은 DLL 중 OpenJigware.dll 을 C# 의 참조에 넣어주고 상단에 “using OpenJigWare;” 라 작성하면 준비 완료. - 아래의 “사용 예” 참조
            1. 단, 3D 모델링을 활용하려면 tao\*.dll 등을 같이 참조에 넣어주어야 한다.
      2. 사용 예
         1. Visual C# 에서의 사용
         2. Visual C++에서의 사용
2. 일반
   1. InputBox : 메세지를 받는 창을 출력하자.

**Created Date: 2016.01.01**

**Modified Date: 2016.01.01**

**revision 1**

**키워드:**

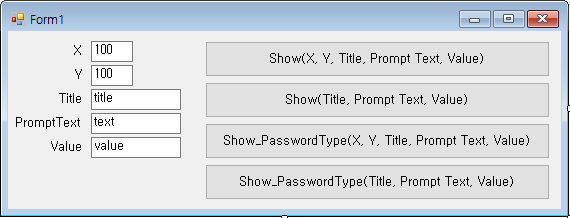
**개발 환경: Microsoft windows, Visual studio 2010 C#**

**1. InputBox**

  윈도우를 사용하다 보면 글자를 입력 받는 Input Box 를 사용하게됩니다. 매번 API 로 작성하기보다 현 오픈지그웨어 내의 함수를 사용해서 보다 편리하게 사용할 수 있습니다.

**2. 프로그램 예제**

Visual Studio 2010, C# 기준으로 아래와 같이 화면을 구성하도록 합니다.



[그림 III-1]

Input box 는 간단합니다.

4가지의 기능밖에 없습니다.

첫번째 버튼은 보시다 시피 뜨는 창의 위치를 조절할 수 있는 버튼으로

두번째는 기본위치에 그냥 쉽게 띄우기...

세번째와 네번째는 동일하지만 패스워드 입력 시 편하도록 만들었습니다.

  1) 첫번째 버튼

    첨부의 OpenJigWare.dll 을 참조에 넣어두고 (강좌 1 참조) 아래와 같이 버튼 이벤트 코드를 각각 작성하면 완성입니다.

// For Use

//  1. 참조(reference) - 참조추가(add reference) - 찾아보기(searching tab) - DLL 선택(OpenJigWare.dll)

//  2. add "using OpenJigWare" as follow

using OpenJigWare;

namespace Ex1\_InputBox

{  
    public partial class Form1 : Form

    {

        public Form1()

        {

            InitializeComponent();

        }

        private void btnInputBox\_Show\_Click(object sender, EventArgs e) // (1)

        {

            String strValue = txtValue.Text;

            if (Ojw.CInputBox.Show(Ojw.CConvert.StrToInt(txtX.Text), Ojw.CConvert.StrToInt(txtY.Text), txtTitle.Text, txtPrompt.Text, ref strValue) == DialogResult.OK)

            {

                txtValue.Text = strValue;

            }

        }

        private void btnInputBox\_Show2\_Click(object sender, EventArgs e) // (2)

        {

            String strValue = txtValue.Text;

            if (Ojw.CInputBox.Show(txtTitle.Text, txtPrompt.Text, ref strValue) == DialogResult.OK)

            {

                txtValue.Text = strValue;

            }

        }

        private void btnInputBox\_Show\_Pswd\_Click(object sender, EventArgs e)

        {

            String strValue = txtValue.Text;

            if (Ojw.CInputBox.Show\_PasswordType(Ojw.CConvert.StrToInt(txtX.Text), Ojw.CConvert.StrToInt(txtY.Text), txtTitle.Text, txtPrompt.Text, ref strValue) == DialogResult.OK)

            {

                txtValue.Text = strValue;

            }

        }

        private void btnInputBox\_Show2\_Pswd\_Click(object sender, EventArgs e)

        {

            String strValue = txtValue.Text;

            if (Ojw.CInputBox.Show\_PasswordType(txtTitle.Text, txtPrompt.Text, ref strValue) == DialogResult.OK)  
            {

                txtValue.Text = strValue;

            }

        }

    }

}

(1) InputBox.Show( x 위치, y 위치, 제목글자, 제목밑의 글자, ref 입력받을 문자)  => 참고로 입력할 문자 위치에 초기 값을 지정해서 넣어 둘 수 있습니다.

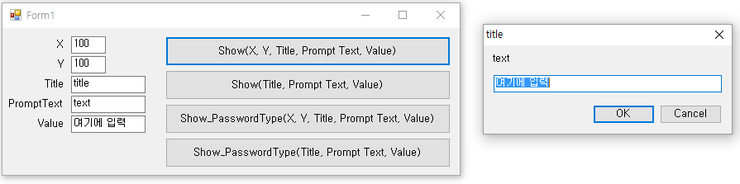
(2) InputBox.Show( 제목글자, 제목밑의 글자, ref 입력받을 문자) => 훨씬 간소화 된 InputBox

(3) (1)과 동일하지만 입력 받는 문자가 \* 로 표기되어 보이지 않게 해 준다. 패스워드 입력에 적당

(4) (2)와 동일하지만 입력 받는 문자가 \* 로 표기되어 보이지 않게 해 준다. 패스워드 입력에 적당

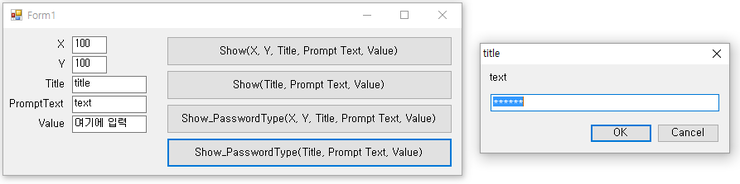
실행화면은 각기 다음과 같습니다.

(1) (2) ->



[그림 III-2]

(3) (4) ->



[그림 III-3]

*End.*

*written by Jinwook On*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [[오픈 지그 웨어] 강좌 2 - InputBox 사용 (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/13790) |**작성자** [쿠루쿠루](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=ojw5014)

* 1. Message : 에디터 창에 디버그 메세지를 출력하고 파일로 저장하자.

**Created Date: 2016.01.01**

**Modified Date: 2016.01.01**

**revision 1**

**키워드:  Open Jig Ware**

**개발 환경: Microsoft windows, Visual Studio 2010 C#**

**1. Message**

  각기 프로그램에서 Printf 나 Cout, 혹은 Trace 등 여러가지 형태의 디버깅 메세지를 다루어 보셨을 것으로 생각됩니다.

리눅스건 윈도우건 항상 프로그래밍을 하면 **History 를 기록을 해야 합니다**.(프로그래머를 꿈꾸신 다면 기본중의 기본으로 여기셔야 할 겁니다. 매우 귀찮죠. -\_-;;;)

Normal 한 상황에서도, 장비를 오픈했는지, 혹은 에러가 언제 어디서 어떻게 발생했는지, 혹은 몇개가 발생했는지를 항상 기록을 해야 합니다.

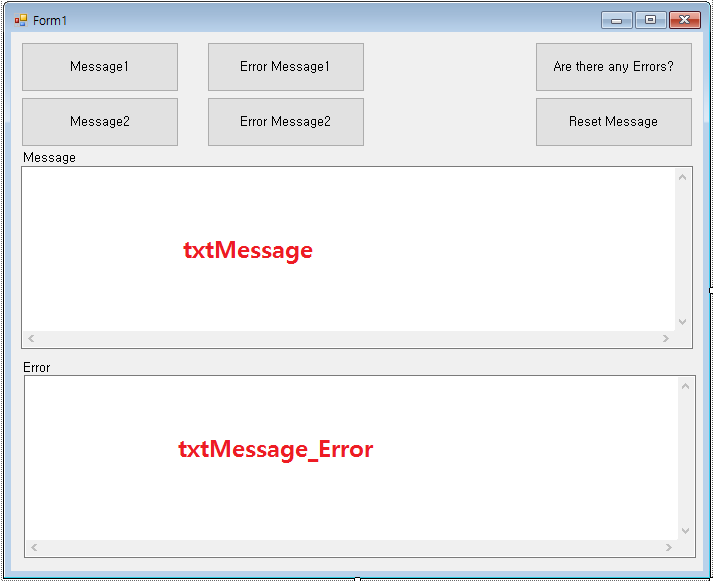
이 Message 기능은 이러한 모든 것들은 간단하게 관리 할 수 있도록 도와줄 것입니다.

모든 명령은 Ojw.CMessage. ... 으로 시작합니다.

앞으로 있을 예제에는 이 명령 구문을 사용해서 메세지가 출력될 수 있도록 강의를 진행하도록 하겠습니다.

**2. 프로그램 예제**

아래와 같이 버튼과 TextBox 들을 배치한다.



[그림 III-4]

  1) 위 구성에서 txtMessage 에 모든 History Message를 출력할 것이다.

  2) txtMessage\_Error 에는 에러상황만 출력이 되도록 할 것이다.

  3) 앞의 두 버튼(Message1~2) 는 각기 다른 테스트 메세지를 출력하도록 할 것이다.

  4) Error Message 1 ~ 2 의 버튼은 각기 다른 에러 메세지를 출력하도록 할 것이다.

  5) Are there any Errors? 버튼으로 몇개의 에러가 있는지 확인할 것이다.

  6) 현재의 누적 기록되어 있는 에러 상태를 리셋할 것이다.

  7) 이 모든 상황을 파일로 기록할 수 있도록 할 것이다.

**3. 실행 코드**

  1). 참조 - 참조추가 - 찾아보기 - DLL 선택(OpenJigWare.dll)  
  2). using OpenJigWare 를 프로그램 상단에 작성 하면 OpenJigWare 사용준비 끝. (강좌 1 참조)

Form\_Load 이벤트에 다음과 같이 기록해 주면 메세지 박스 설정 완료

Ojw.CMessage.Init(txtMessage); // 메세지 박스 기본 출력위치 지정

Ojw.CMessage.Init( 텍스트박스 ) 는 지정한 텍스트 박스에 모든 메세지 상황을 기록하겠다는 뜻이다.

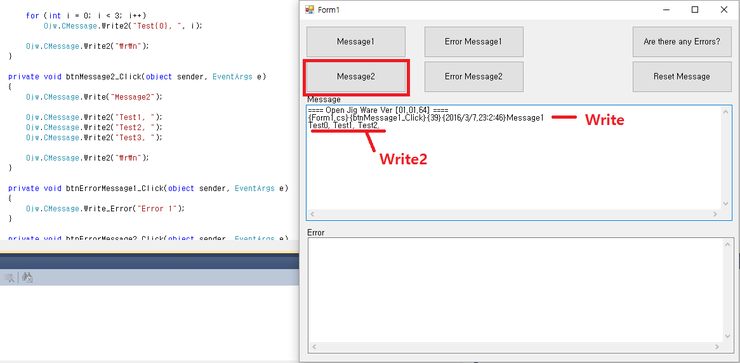
행여 Error 만 따로 기록하고 싶다면 다음과 같은 라인을 추가해 주면 된다.

Ojw.CMessage.Init\_Error(txtMessage\_Error); // 메세지 박스 에러 출력위치 지정

이후부터 아래의 명령을 사용하여 메세지를 출력하면 된다.

Ojw.CMessage.Write( "메세지", arg0, arg1, ... argn );

Ojw.CMessage.Write2() Write 명령과 동일하나 줄바꿈이 없다.출력 메세지의 차이를 각기 확인해 보자



[그림 III-5]

btnMessage2\_Click 함수에 각기 Write 와 Write2 가 사용되어 있고 우측의 프로그램이 출력 결과를 비교해 주고 있다.

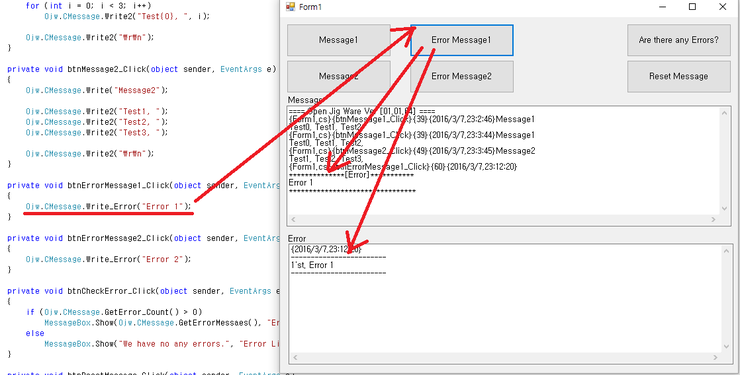
Write 는 해당 함수가 실행되는 위치, 시간등을 비교적 자세히 출력하고 있고 완료 후 자동 줄바꿈을 하고 있다.

Write2 는 줄바꿈이 없어 직접 지령해 주어야 하고 보조 메세지가 없으나 대신 자유로이 사용할 수 있다.

Wriet / Write2 는 둘다 기본 속성으로 txtMessage 에 출력하도록 되어 있지만 출력 위치를 지정해서 별도로 출력하는 것도 가능하다.

  -> Write(TextBox, ....) : 좌측의 형식으로 할 시 지정한 텍스트 박스에 메세지 출력이 가능하다.

Ojw.CMessage.Write\_Error( "메세지", arg0, arg1, ... argn ); // 메세지 박스 에러 출력



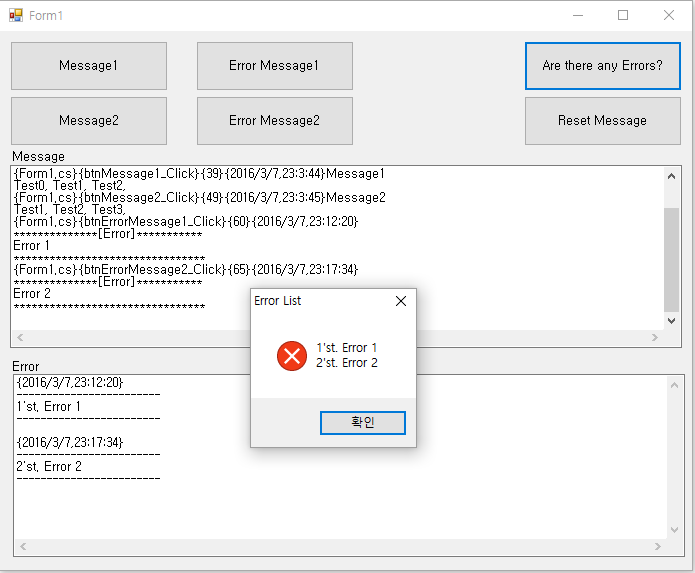
[그림 III-6]

Message History 지정한 곳에는 에러를 포함한 메세지 전체가, Error Message 지정한 곳에는 에러의 카운트와 함께 에러만 기록이 됩니다.

이제 메세지 에러 상황을 한번 체크해 보도록 하겠습니다. 이후의 진행은 편의상 존칭을 사용하지 않도록 하겠습니다.

Ojw.CMessage.GetError\_Count() 로 검사 했을 때 Error 가 있다면 0보다 큰 값이 나온다.

Ojw.CMessage.GetErrorMessaes() 로 현재까지의 누적 에러메세지를 가져올 수 있다.



[그림 III-7]

Are there any errors? 버튼을 눌러 메세지를 확인해 보았다.

Ojw.CMessage.Reset() 을 실행하면 여태까지 기록된 모든 에러의 카운트와 메세지가 초기화 된다.

**4. History file 기록**

자, 여기까지 메세지를 출력을 하였다면 출력한 메세지를 파일로 만들어 지도록 해 보는 것은 어떻겠는가?

사실 이렇게 모든 메세지가 파일로 기록되는 상황은 반드시 필요하다. 장비를 만들다 보면, 혹은 로봇을 만들다 보면 내가 지켜보지 않는 가운데 메세지가 어떻게 발생되었는지 궁금한 경우가 있다.

이러한 프로그래밍은 꽤 귀찮은 작업이지만 없어서는 안되는 정말 필요한 작업이라고 할 수 있다.

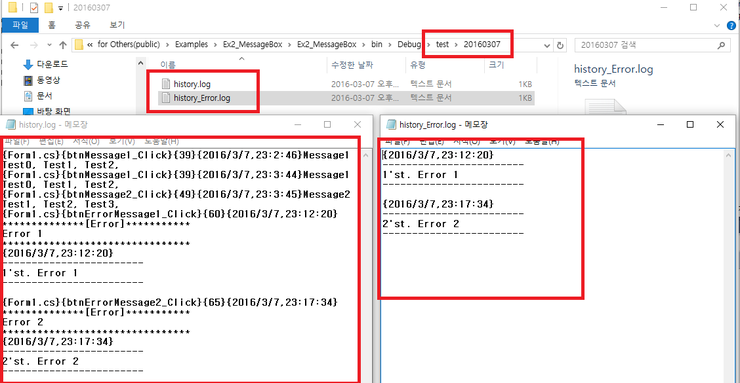
Open Jig Ware 에서는 다음의 라인을 초기화 구문 뒤에 살쩍 넣어주면 된다.(앞에 넣어도 되긴 하지만 모양상... ^^;;;)

Ojw.CMessage.Init\_File(true); // Save History files(Default : false)

Ojw.CMessage.Init\_FilePath(Application.StartupPath + "\\" + "test"); // change the history file path, (default = current)

첫번째 라인은 파일을 기록을 하겠다고 선언하는 것으로 이것만 하면 기본적으로 현재의 사용자 프로그램 내부에서 History 라는 폴더를 만들고 매일매일 폴더를 만들어서 별개의 텍스트 로그 파일을 기록하게 된다.

두번째 라인은 저장되는 폴더를 사용자가 임의로 지정하고 싶을 때 사용한다.



[그림 III-8]

위의 그림에서 보다시피 지정한 경로에 오늘 날짜로 파일이 생성된 것을 볼수 있고 각기 History Message 와 Error Message 로 구분해서 저장된 것을 확인 할 수 있다.

그리고 여기까지... 걱정하지 마시라.

초기 선언만 2~3줄 선언해 놓으면 나머지는 알아서 자동으로 생성해 놓는다. 사용자가 신경쓸 필요가 없다는 얘기이다.

메세지 기능은 더 있지만 기본적으로 여기까지만 알면 현업에서도 충분히 활용 가능할 것이다.

그리고 약간 보너스로 설명 하자면...

 1) Message 가 저장되듯이 이미지가 저장되는

Ojw.CMessage.SaveImageFile(string strFileName, Bitmap bmp);

함수가 있다.

 2) 그리고 에러 메세지를 마지막 메세지만 확인 하려면

Ojw.CMessage.GetLastErrorMessage();

를 이용하면 된다.

  3) 마지막으로 에러 카운트 말고 에러가 있는지 없는지 확인하려면

Ojw.CMessage.IsError();

함수도 있다.

이 외에도 다양한 기능들이 있으니 찾아보는 재미도 쏠쏠할 것으로 생각된다.

이 정도만 사용해도 별 무리없이 History 기능을 사용할 수 있을 것이다.

이상입니다. 지난밤 부터 예전 회사에서의 미처 못끝낸 업무로 인해 밤을 지세웠더니 헤롱헤롱 하군요.

모두들 행복한 오로카 되시길...

**[출처]** [[오픈 지그 웨어] 강좌 3 - Message (\*\*\*중요\*\*\*) (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/13791) |**작성자** [쿠루쿠루](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=ojw5014)

* 1. Convert

**Created Date: 2016.01.01**

**Modified Date: 2016.01.01**

**revision 1**

**키워드:  Open Jig Ware, Convert**

**개발 환경:  Microsoft windows, Visual Studio 2015 C#**

 안녕하세요. Open Jig Ware프로젝트에 참여하게 된 SSM입니다. 메뉴얼 작업을 맡게되었습니다. 아직 프로그래밍을 접한지 얼마되지 않았기 때문에 배우는 입장입니다. 부족한 부분이 많더라도 이해해 주시고 많은 조언 부탁드립니다.

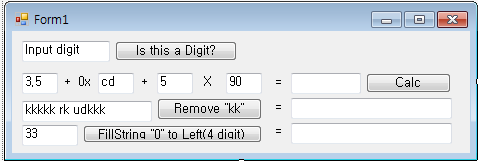
 아래의 강좌는 쿠루쿠루님의 **'[오픈 지그 웨어] 강좌' 를 참조하여 만든것입니다.**

**1. Convert**

 Convert는 각종 변환 함수를 모아놓은 것 입니다.

**2. 프로그램 예제**

이번에 다룰 예제에는 총 4가지의 기능이 들어있습니다. 하지만 예제를 자세히 알아보기 전에 참조추가를 해주셔야 예제를 실행했을 때 프로그램이 제대로 작동하기 때문에 반드시 참조추가를 해주세요.



[그림 III-9]

Visual Studio 2015(C#)을 이용하여, 위와 같은 형태로 만들어 줍니다.

// For Use

//  1. 참조 - 참조추가 - 찾아보기 - DLL 선택(OpenJigWare.dll)

//  2. add "using OpenJigWare" as follow

using OpenJigWare;

"**using OpenJigWare;**" 를 반드시 적어주세요.(자세한 내용은 <http://cafe.naver.com/openrt/13764>를 참고)

namespace Ex3\_Convert

{  
    public partial class Form1 : Form

    {

        public Form1()

        {

            InitializeComponent();

        }

        private void button1\_Click(object sender, EventArgs e) // [1]

        {

            if (Ojw.CConvert.IsDigit(txtIsDigit.Text) == true)

            {

                MessageBox.Show("Yes");

            }

            else

            {

                MessageBox.Show("Not");

            }

        }

        private void btnTest\_Click(object sender, EventArgs e) // [2]

        {

            float fValue = Ojw.CConvert.StrToFloat(txtFloat.Text);

            int nHex = Ojw.CConvert.HexStrToInt(txtHex.Text);

            int nValue = Ojw.CConvert.StrToInt(txtInt.Text);

            double dValue = Ojw.CConvert.StrToDouble(txtDouble.Text);

            float fRet = fValue + (float)nHex + (float)nValue \* (float)dValue;

            txtResult.Text = Ojw.CConvert.FloatToStr(fRet);

        }

        private void btnRemoveString\_Click(object sender, EventArgs e) // [3]

        {

            txtResult\_RemoveString.Text = Ojw.CConvert.RemoveString(txtRemoveString.Text, "kk");

        }

        private void btnTest2\_Click(object sender, EventArgs e) // [4]

        {

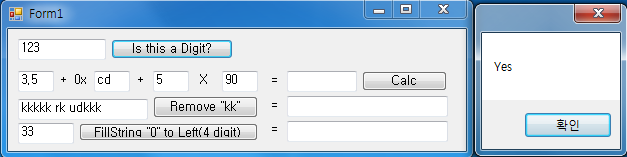
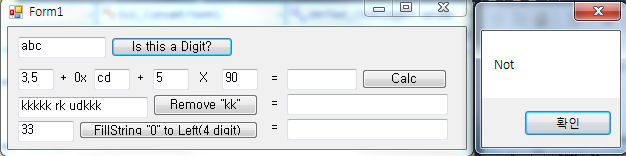
            txtResult2.Text = Ojw.CConvert.FillString(txtTest2.Text, "0", 4, false);

        }

    }

}

[1] 'Is this a Digit?'버튼의 소스코드입니다. text창에 숫자를 입력하면 "Yes"를, 아닌 경우 "Not"이 출력되는 기능입니다.

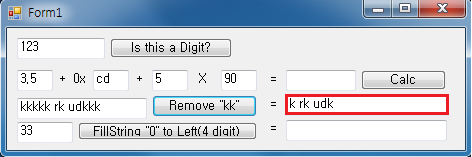
 [그림 III-10]

[2] 'Calc'버튼의 소스코드 입니다. 숫자+숫자+숫자\*숫자의 값이 나옵니다.(단 두번째 숫자는 16진수)



[그림 III-11]

[3] 'Remove "kk"'버튼의 기능은 말그대로 텍스트창에 써져있는 "kk"를 제거해 주는 기능입니다. 단, k가 단독으로 존재할때는 없어지지 않습니다.



[그림 III-12]

[4] 'FillString "0" to Left(4 digit)'버튼의 기능은 4자리수 숫자의 앞쪽 빈자리에 0을 채워넣어주는 기능을 합니다. 백의 자리숫자면 앞에 0을 한개, 십의 자리면 두개, 일의 자리면 3개를 넣어줍니다.



 [그림 III-13]

**3. Convert Source**

이제 예제에 사용된 변환함수들을 살펴보도록 하겠습니다.

[1]

Ojw.CConvert.IsDigit(char cValue)  // 텍스트창에 입력한 문자가 숫자인지 아닌지 확인합니다.

Ojw.CConvert.IsDigit(string strValue)

// Check Numeric or ...

            public static bool IsDigit(char cValue) { if (!Char.IsNumber(cValue)) return false; else return true; }

            public static bool IsDigit(string strValue)

            {

                bool bSimbol = false;

                bool bPoint = false;

                int nPoint = -1;

                foreach (char cItem in strValue)

                {

                    if ((cItem == '-') || (cItem == '+'))

                    {

                        if (bSimbol == true) return false;

                        bSimbol = true;

                    }

                    else

                    {

                        if (nPoint >= 0)

                        {

                            if (cItem == '.')

                            {

                                nPoint++;

                                bPoint = true;

                            }

                            else bPoint = false;

                        }

                        if (cItem != '.')

                        {

                            bSimbol = false;

                            if (!Char.IsNumber(cItem))

                                return false;

                        }

                    }

                    if (nPoint < 0) nPoint = 0;

                }

                if (bSimbol == true) return false;

                else if (nPoint > 1) return false;

                else if (bPoint == true) return false;

                return true;

            }

[2]

Ojw.CConvert.StrToFloat(String strData)  // string형 데이터를 float형으로 바꿔줍니다.

            public static float StrToFloat(String strData)

            {

                try

                {

                    return float.Parse(strData); // string형 문자열 받아서 float형 숫자로 반환

                }

                catch

                {

                    return 0.0f;  // 문자열이 숫자가 아닌 경우 float형 숫자 '0'으로 반환

                }

            }

Ojw.CConvert.HexStrToInt  // 16진수 문자열을 받아서 10진수 숫자로 바꿔줍니다.

// HEXA string -> Numeric

            public static int HexStrToInt(String strData)

            {

                try

                {

                    return int.Parse(**S\_StrTrim**(strData), System.Globalization.NumberStyles.HexNumber);  
                }  // string형 문자열(16진수)을 받아서 int형 숫자로 변환

                catch

                {

                    return 0;  // 예외가 생긴 경우 숫자 '0'으로 취급

                }

            }

// remove letters '$' and space(Kor: '$' 로 표시된 문자에서 공백과 '$' 문자 제거)

            public static String**S\_StrTrim**(String strData)

            {

                try

                {

                    String strTmp;

                    strTmp = strData.Trim();

                    strTmp = strTmp.Trim('$');

                    strTmp = strTmp.Trim();

                    strTmp = strTmp.PadLeft(2, '0');

                    //strTmp = strTmp.Substring(strTmp.Length - 2);

                    return strTmp;

                }

                catch

                {

                    return "";

                }

            }

Ojw.CConvert.StrToInt  // string형 데이터를 int형으로 바꿔줍니다.

            public static int StrToInt(String strData)

            {

                try

                {

                    return int.Parse(strData);  // string형 문자열을 받아서 int형 숫자로 반환

                }

                catch

                {

                    return 0; // 예외가 생긴 경우 '0'을 반환

                }

            }

Ojw.CConvert.StrToDouble  // string형 데이터를 double형으로 바꿔줍니다.

            public static double StrToDouble(String strData)

            {

                try

                {

                    return double.Parse(strData);  // string형 문자열을 받아서 double형 숫자로 반환

                }

                catch

                {

                    return 0.0;  // 예외가 있는 경우 숫자 '0'을 반환

                }

            }

Ojw.CConvert.FloatToStr  // float형 데이터를 string형으로 바꿔줍니다.

            public static String FloatToStr(float fData)

            {

                try

                {

                    return Convert.ToString(fData);  // float형 숫자를 받아서 string형 문자열로 변환

                }

                catch

                {

                    return null;  // 예외가 있는 경우 null(0)을 반환

                }

            }

[3]

Ojw.CConvert.RemoveString

// Partially replacing characters(Kor: 부분적 문자 삭제)

            public static string RemoveString(string strText, String strRemoveData)

            {

                string strResult = strText;

                int nIndex;

                int nStart = 0;

                int nW = strRemoveData.Length;

                while ((nIndex = strResult.IndexOf(strRemoveData, nStart)) >= 0)

                {

                    StringBuilder stb = new StringBuilder(strResult);

                    strResult = stb.Remove(nIndex, nW).ToString();// strResult.Remove(nIndex, nW);

                    if (nIndex >= nStart)

                        nStart = nIndex;

                    else break;

                }

                return strResult;

            }

[4]

Ojw.CConvert.FillString

            public static string FillString(String strOrg, String strSpace, int nStringWidth, bool bRight)

            {

                String strRet = ""; ;

                for (int i = 0; i < (nStringWidth - strOrg.Length); i++) strRet += strSpace;

                if (bRight == true) strRet = strOrg + strRet;

                else strRet = strRet + strOrg;

                return strRet;

            }

아직 소스코드에 대한 분석이 끝나지 않았습니다. 이 부분은 앞으로 또다른 메뉴얼들을 작성하면서 함께 수정해나갈 예정입니다.

*End.*

*written by Sanghyeok Lee*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [[오픈 지그 웨어] 강좌 5 - Convert (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15068) |**작성자** [SSM](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=dlfjsdksk2)

* 1. Math

**Created Date: 2016.08.28**

**Modified Date: 2016.09.03**

**revision 6**

**키워드: 오픈지그웨어, OpenJigWare, Math**

**개발 환경: Microsoft Windows, Visual Studio C# 2015**

**1. Math Class**

안녕하세요. 이번에 오픈지그웨어 메뉴얼을 작성하게 된 ThinkPoint입니다. 일단 이 프로젝트에 참가하게 해주시고 아낌없는 지원을 해주신 온진욱 선생님과 로보링크 분들, 오픈지그웨어 프로젝트 관계자님들께 감사드립니다. 아직 배우고 있는 학생이라 프로그래밍과 코딩하는 법도 잘 모르고 많이 부족합니다만, 메뉴얼은 계속 업데이트 할 예정이므로 잘못된 점이나 개선될 수 있는 부분이 있으면 댓글로 달아주시기 바랍니다.

Math Class는 수학 관련 함수들을 담아놓은 클래스입니다. 오픈지그웨어를 이용하면서 수학적인 변환을 많이 사용하실 텐데, 그 부분에서 유용하게 쓰실 수 있습니다. 파일 이름은 COjw\_05\_Math.cs입니다.

클래스는 Ojw.CMath 입니다.

**2. 상수, 함수 설명**

**※상수**

Ojw.CMath.PI; // Math.Pi;와 같습니다. 원주율(파이)를 나타냅니다.

Ojw.CMath.\_PI; // 위와 같습니다

Ojw.CMath.\_ZERO; // 0.000001을 나타냅니다. 0으로 계산할 때(나누는 등) 에러가 뜨는 것을 방지하는 용도로 쓰입니다.

Ojw.CMath.ZERO(); // 위와 같습니다.

**예시**

예시는 예전 강의에서 진행했던 Ojw.Message.Write를 써서 메세지박스에 위의 값이 나타나도록 설정하였습니다. 아래는 사용한 코드와 실행 결과입니다.

        private void btnMessage\_Click(object sender, EventArgs e)

        {

            //PI

            double dPI = Ojw.CMath.PI;

            string sPI = Ojw.CConvert.DoubleToStr(dPI);

            Ojw.CMessage.Write(sPI);

            //Zero

            double dZero = Ojw.CMath.Zero();

            string sZero = Ojw.CConvert.DoubleToStr(dZero);

            Ojw.CMessage.Write(sZero);

        }



[그림 III-14]

**※반올림, 각도 변환**

Ojw.CMath.Round(double x); //Math.Round(double x);와 같습니다. 가장 가까운 정수로 반올림합니다.

Ojw.CMath.Round(double x, int nPos); //Math.Round(double x, int nPos);와 같습니다. 소수점 nPos자리까지 반올림합니다.

Ojw.CMath.Deg2Rad(double x); //단위가 도(degree, º)인 x를 라디안(rad)으로 바꿉니다. x / 180.0 \* PI

Ojw.CMath.D2R(double dValue); //위와 같습니다.

Ojw.CMath.AngleToRadian(double t); //위와 같습니다.

Ojw.CMath.Rad2Deg(double x); //단위가 라디안(rad)인 x를 도(degree, º)로 바꿉니다.

Ojw.CMath.R2D(double dValue); //위와 같습니다.

Ojw.CMath.RadianToAngle(double t); //위와 같습니다.

**예시**

위와 같은 방식을 이용하였습니다. 아래는 사용한 코드와 실행 결과입니다.

        private void btnMessage2\_Click(object sender, EventArgs e)

        {

            //Round

            double dx = Ojw.CMath.Round(5.457, 2);

            string sx = Ojw.CConvert.DoubleToStr(dx);

            Ojw.CMessage.Write(sx);

            //Deg2Rad

            double drad1 = Ojw.CMath.Deg2Rad(180);

            string srad1 = Ojw.CConvert.DoubleToStr(drad1);

            Ojw.CMessage.Write(srad1);

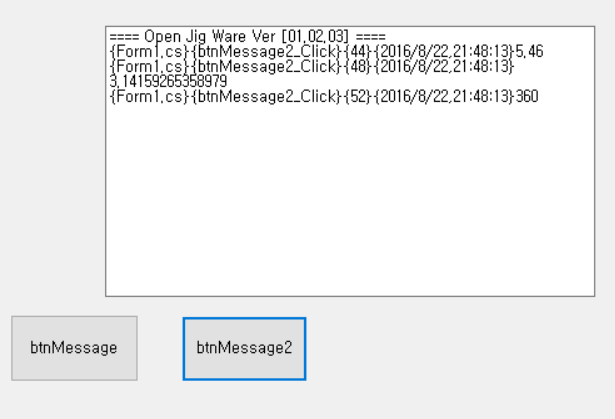
            //Rad2Deg

            double ddeg1 = Ojw.CMath.R2D(2\*Ojw.CMath.PI);

            string sdeg1 = Ojw.CConvert.DoubleToStr(ddeg1);

            Ojw.CMessage.Write(sdeg1);

        }



[그림 III-15]

OpenJigWare의 Math클래스의 가장 큰 특징중 하나는 모든 삼각함수의 단위가 **각도(degree)**라는 것입니다.

**※삼각함수 관련**

Ojw.CMath.Sin(double t); //Math.Sin(double t);와 같은 기능이지만, t는 각도(degree)를 받습니다. 사인함수입니다.

Ojw.CMath.Cos(double t); //Math.Cos(double t);와 같은 기능이지만, t는 각도(degree)를 받습니다. 코사인함수입니다.

Ojw.CMath.Tan(double t); //Math.Tan(double t);와 같은 기능이지만, t는 각도(degree)를 받습니다. 탄젠트함수입니다.

Ojw.CMath.ASin(double t); //Math.ASin(double t);와 같은 기능이지만, 반환값은 각도(degree)로 표현됩니다. 아크사인함수(역삼각함수)이고, 반환값은 -*90도* *≤θ≤90도*입니다. t는 -1~1의 사이값이어야 하고 아니라면 NaN이 반환됩니다.

Ojw.CMath.ACos(double t); //Math.ACos(double t);와 같은 기능이지만, 반환값은 각도(degree)로 표현됩니다. 아크코사인함수(역삼각함수)이고, 반환값의 범위는 0*도* *≤θ≤180도*입니다. t는 -1~1의 사이값이어야 하고 아니라면 NaN이 반환됩니다.

Ojw.CMath.Hypotenuse(double x1, double y1); //좌표평면에서 x좌표를 x1, y좌표를 y1으로 설정하여 그 빗변의 길이(원점까지의 거리)로 반환됩니다. 즉, x1의 제곱과 y1의 제곱의 합의 제곱근으로 반환됩니다.

Ojw.CMath.ASinXY(double x1, double y1); //좌표평면에서 x좌표를 x1, y좌표를 y1으로 설정하여 극좌표 방향(x축에서 시계 반대 방향)으로 그 빗면의 각도(degree)를 반환합니다. 반환갑의 범위는0*도* *≤θ≤360도*입니다.

Ojw.CMath.ACosXY(double x1, double y1); //위와 같습니다.

Ojw.CMath.ASin\_Plane\_23(double t); //  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.CMath.ASin\_Plane\_4(double t); //  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.CMath.ASin\_Plane(int nPlane, double t); //  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.CMath.ACos\_Plane(int nPlane, double t); //  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.CMath.ATan\_Calc(double t); //Math.Atan(double t);와 같은 기능이지만, 반환값은 각도(degree)로 표현됩니다. 아크탄젠트함수(역삼각함수)이고, 반환값의 범위는 *-90도≤θ≤90도*입니다. t가 +무한대라면 반환값은 90도, t가 -무한대라면 반환값은 -90도 입니다.

Ojw.CMath.ATan(double t); //위와 같은 기능이지만, 반환값의 범위가 *0도≤θ<180도*입니다(위의 음수 반환값에 180도만큼 더합니다.). 반환값은 각도(degree)로 표현됩니다.

Ojw.CMath.ATan2(double x\_Down\_1, double y\_Up\_1); //Math.Atan2(double y, double x);와 비슷하고, Ojw.CMath.ASinXY(double x1, double y1);와 같은 기능을 가집니다. 좌표평면에서 x좌표를 x\_Down\_1, y좌표를 y\_Up\_1으로 설정하여 Ojw.CMath.ATan(y\_Up\_1/x\_Down\_1);의 값을 반환합니다. 또한 위와 마찬가지로 각도(degree)로 반환되고, 반환값의 범위는 *0도≤θ≤360도*입니다.

**예시**

위와 같은 방식을 이용하였습니다. 아래는 사용한 코드와 실행 결과입니다.

        private void btnMessage3\_Click(object sender, EventArgs e)

        {

            //Tan

            double dtan1 = Ojw.CMath.Tan(45);

            string stan1 = Ojw.CConvert.DoubleToStr(dtan1);

            Ojw.CMessage.Write(stan1);

            //ACos

            double dacos1 = Ojw.CMath.ACos(1);

            string sacos1 = Ojw.CConvert.DoubleToStr(dacos1);

            Ojw.CMessage.Write(sacos1);

            //Hypotenus

            double dhy1 = Ojw.CMath.Hypotenuse(3, 4);

            string shy1 = Ojw.CConvert.DoubleToStr(dhy1);

            Ojw.CMessage.Write(shy1);

            //ASinXY

            double dasinxy1 = Ojw.CMath.ASinXY(-1, 1);

            string sasinxy1 = Ojw.CConvert.DoubleToStr(dasinxy1);

            Ojw.CMessage.Write(sasinxy1);

            //ACosXY

            double dacosxy1 = Ojw.CMath.ACosXY(-1, 1);

            string sacosxy1 = Ojw.CConvert.DoubleToStr(dacosxy1);

            Ojw.CMessage.Write(sacosxy1);

            //ATan

            double datan1 = Ojw.CMath.ATan(-Math.Sqrt(3));

            string satan1 = Ojw.CConvert.DoubleToStr(datan1);

            Ojw.CMessage.Write(satan1);

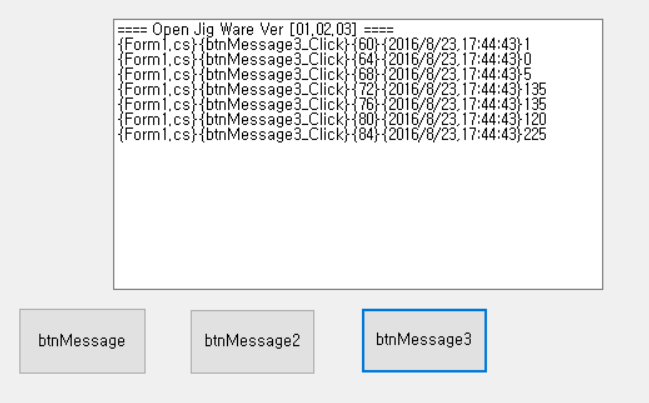
            //ATan2

            double datan2 = Ojw.CMath.ATan2(-1,-1);

            string satan2 = Ojw.CConvert.DoubleToStr(datan2);

            Ojw.CMessage.Write(satan2);

        }



[그림 III-16]

**※기타계산 함수**

Ojw.CMath.Distance(double x, double y); //Ojw.CMath.Hypotenuse(double x1, double y1); 와 같은 기능입니다. 좌표평면에서 x좌표를 x1, y좌표를 y1으로 설정하여 그 빗변의 길이(원점까지의 거리)로 반환됩니다. 즉, x1의 제곱과 y1의 제곱의 합의 제곱근으로 반환됩니다.

Ojw.CMath.VectorNorm(double x, double y); //위와 같습니다.

Ojw.CMath.Norm(double x, double y); //위와 같습니다.

Ojw.CMath.Pow(double x); //x의 제곱을 반환합니다.

Ojw.CMath.Pow(double x, double s); //x의 s제곱을 반환합니다. Math.Pow(x, s);와 같습니다.

Ojw.CMath.Dot(SVector3D\_t u, SVector3D\_t v); //3차원의 점을 나타내는 SVector3D\_t 구조체의 점곱(내적, 스칼라곱)을 반환합니다. SVector3D\_t 구조체에 관해서는 Struct 강좌에서 알려드리도록 하겠습니다.

Ojw.CMath.Norm(SVector3D\_t v); //원점에서부터 3차원 점을 나타내는 구조체 v의 좌표까지의 거리를 나타냅니다.

Ojw.CMath.Distance(SVector3D\_t u, SVector3D\_t v); //3차원의 점을 나타내는 구조체 u, v의 좌표 사이의 거리를 반환합니다.

**예시**

위와 같은 방식을 이용하였습니다. 아래는 사용한 코드와 실행 결과입니다.

        private void btnMessage4\_Click(object sender, EventArgs e)

        {

            //Distance

            double dd1 = Ojw.CMath.Distance(3,4);

            string sd1 = Ojw.CConvert.DoubleToStr(dd1);

            Ojw.CMessage.Write(sd1);

            //Pow

            double dpow1 = Ojw.CMath.Pow(2, 4);

            string spow1 = Ojw.CConvert.DoubleToStr(dpow1);

            Ojw.CMessage.Write(spow1);

            //Dot(struct)

            Ojw.SVector3D\_t dot1 = new Ojw.SVector3D\_t(1, 2, 3);

            Ojw.SVector3D\_t dot2 = new Ojw.SVector3D\_t(3, 2, 1);

            double ddot3 = Ojw.CMath.Dot(dot1, dot2);

            string sdot3 = Ojw.CConvert.DoubleToStr(ddot3);

            Ojw.CMessage.Write(sdot3);

            //Norm(struct)

            double dnorm1 = Ojw.CMath.Norm(dot1);

            string snorm1 = Ojw.CConvert.DoubleToStr(Ojw.CMath.Pow(dnorm1));

            Ojw.CMessage.Write(snorm1);

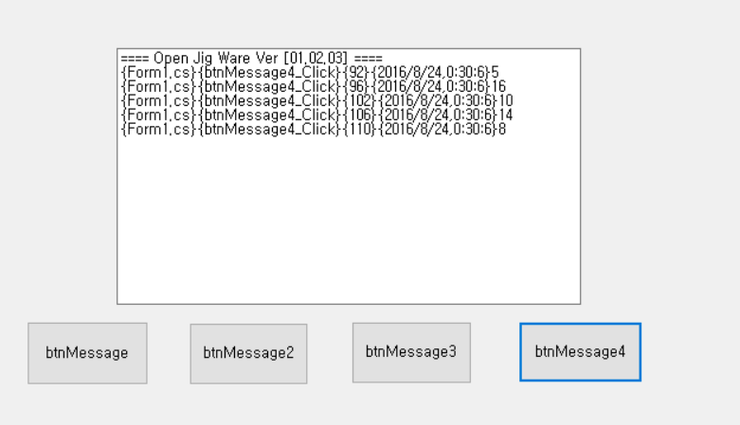
            //Distance(struct)

            double dd2 = Ojw.CMath.Distance(dot1, dot2);

            string sd2 = Ojw.CConvert.DoubleToStr(Ojw.CMath.Pow(dd2));

            Ojw.CMessage.Write(sd2);

        }



[그림 III-17]

**※행렬, DH Notation 계산**

Ojw.CMath.CalcMatrix(int nLine, double[,] adS0, double[,] adS1, out double[,] adRes);   
//2차원 nLine차수의 정사각 행렬을 adS0와 adS1에서 위쪽 부터 떼어내어 그 행렬끼리 순서대로 곱하여 adRes에 반환합니다. CalcMatrix함수 자체는 bool 형식이며, 계산에 성공하면 true를 반환하고, 계산에 실패하면 false를 반환합니다. 행렬 adS0와 adS1의 원소의 수는 반드시 둘 다 nLine의 제곱보다 작지 않아야 하고, 그렇지 않으면 계산에 실패합니다.

※행렬의 종류에 따라 오류가 있을 수 있습니다.

Ojw.CMath.CalcMatrix\_Str(int nLine, String[,] aStrS0, String[,] aStrS1, out String[,] aStrRes);  
//2차원 nLine차수의 정사각 행렬을 aStrS0와 aStr1에서 위쪽 부터 떼어내어 그 행렬끼리 순서대로 곱하여 aStrRes에 반환합니다. aStrS0와 aStr1는 string의 배열 변수이며, 위 함수는 문자로 된 행렬을 가상으로 계산해서 문자로 출력합니다. CalcMatrix\_Str함수 자체는 bool 형식이며, 계산에 성공하면 true를 반환하고, 계산에 실패하면 false를 반환합니다. 행렬 adS0와 adS1의 원소의 수는 반드시 둘 다 nLine의 제곱보다 작지 않아야 하고, 그렇지 않으면 계산에 실패합니다.  
※행렬의 종류에 따라 오류가 있을 수 있습니다.

Ojw.CMath.CalcT(double dA, double dAlpha, double dD, double dTheta, out double[,] adT);  
//4개의 DH Parameters를 받고, DH Representation 4x4 행렬을 adT에 반환합니다. dAlpha와 dTheta의 각도값은 각(degree)으로 받습니다. CalT함수 자체는 bool 형식이며, true를 반환합니다.

Ojw.CMath.CalcT\_Str(double dA, double dAlpha, double dD, double dTheta, String strMotName, out String[,] astrT);  
//실제 DH Presentation의 식을 가상으로 계산해서 문자열로 반환하는 함수입니다.  4개의 DH Parameters와 모터 이름을 받고, DH Representation 4x4 행렬을 string의 배열로 astrT에 반환합니다. 모터 이름은 theta의 계산식에 "strMotName+theta"와 같은 형식으로 표현됩니다. dAlpha와 dTheta의 각도값은 각(degree)으로 받습니다. CalT\_Str함수 자체는 bool 형식이며, true를 반환합니다.

Ojw.CMath.CalcRot(double dAngleX, double dAngleY, double dAngleZ, double[,] adSrc, out double[,] adRot);  
//adSrc행렬에 회전행렬을 곱한 후 adRot로 반환합니다. x축 회전각도, y축 회전각도, z축 회전각도와 4x4행렬 adSrc를 받고 4x4 행렬을 adRot에 반환합니다. dAngleX, dAngleY, dAngleZ의 각도값은 각(degree)으로 받습니다. CalT함수 자체는 bool 형식이며 계산 후 true를 반환합니다.   
※현재 일부 오류가 있습니다.

Ojw.CMath.CalcRot(double dAngleX, double dAngleY, double dAngleZ, ref double dX, ref double dY, ref double dZ);  
//3차원의 점을 회전시켜 좌표로 반환합니다. x축 회전각도, y축 회전각도, z축 회전각도와 3차원 점의 x, y, z좌표를 참조로 받고 다시 dX, dY, dZ로 반환합니다. dAngleX, dAngleY, dAngleZ의 각도값은 각(degree)으로 받습니다. 3차원 좌표계는 오른손 좌표계를 사용하고 회전 방향은 오른손 법칙 방향입니다. 함수 자체는 bool 형식이며 계산 후 true를 반환합니다.   
※현재 일부 오류가 있습니다.

Ojw.CMath.MakeMatrix\_Rot(double dAngleX, double dAngleY, double dAngleZ, out double[,] adRot);  
//3차원 4x4 회전행렬을 반환합니다. x축 회전각도, y축 회전각도, z축 회전각도를 받고 4x4 행렬을 adRot에 반환합니다. dAngleX, dAngleY, dAngleZ의 각도값은 각(degree)으로 받습니다. 함수 자체는 bool 형식이며 계산 후 true를 반환합니다.  
※현재 일부 오류가 있습니다.

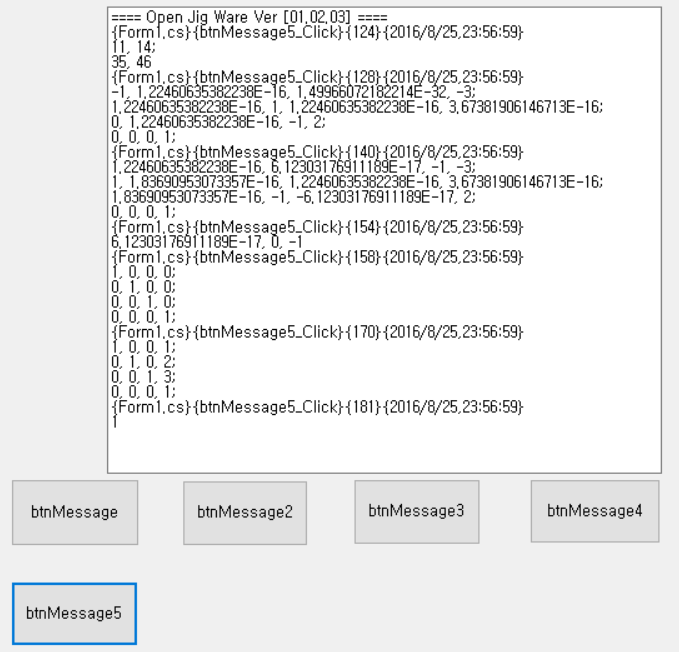
Ojw.CMath.MakeMatrix(double dX, double dY, double dZ, out double[,] adMatrix);  
//x, y, z값의 받고 4x4행렬을 adMatrix에 반환합니다. 행렬은 C# 표현방식으로 다음과 같습니다.  
{ {1, 0, 0, dX}, {0, 1, 0, dY}, {0, 0, 1, dZ}, {0, 0, 0, 1} }  
함수 자체는 bool 형식이며 계산 후 true를 반환합니다.

Ojw.CMath.CompareMatrix(int nLine, double[,] adSrc0, double[,] adSrc1);  
//adSrc0과 adSrc1에서 왼쪽 위에서부터 nLine x nLine 만큼의 정사각행렬을 떼어내어 원소를 0.001까지 비교합니다. 함수는 bool 형식이며 0.001까지 같다면 true, 아니라면 false를 반환합니다.

**예시**

위와 같은 방식을 이용하였습니다. 아래는 사용한 코드와 실행 결과입니다.

        private void btnMessage5\_Click(object sender, EventArgs e)  
        {  
            //CalcMatrix  
            double[,] mat1 = { { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 9, 10, 11, 12 }, { 13, 14, 15, 16 } };  
            double[,] mat2 = { { 1, 2, 3, 4 }, { 5, 6, 7, 8 }, { 12, 11, 10, 9 }, { 16, 15, 14, 13 } };  
            double[,] mat3;  
            bool calcMat1 = Ojw.CMath.CalcMatrix(2, mat1, mat2, out mat3);  
            string sMat3\_00 = Ojw.CConvert.DoubleToStr(mat3[0, 0]);  
            string sMat3\_01 = Ojw.CConvert.DoubleToStr(mat3[0, 1]);  
            string sMat3\_10 = Ojw.CConvert.DoubleToStr(mat3[1, 0]);  
            string sMat3\_11 = Ojw.CConvert.DoubleToStr(mat3[1, 1]);  
            Ojw.CMessage.Write("\r\n{0}, {1}; \r\n{2}, {3}", sMat3\_00, sMat3\_01, sMat3\_10, sMat3\_11);  
            //CalcT  
            double[,] mat4;  
            bool calcT1 = Ojw.CMath.CalcT(3, 180, 2, 180, out mat4);  
           Ojw.CMessage.Write("\r\n{0}, {1}, {2}, {3}; \r\n{4}, {5}, {6}, {7}; \r\n{8}, {9}, {10}, {11}; \r\n{12}, {13}, {14}, {15};"   
            Ojw.CMessage.Write("\r\n{0}, {1}; \r\n{2}, {3}", sMat3\_00, sMat3\_01, sMat3\_10, sMat3\_11);  
                , Ojw.CConvert.DoubleToStr(mat4[0, 0]), Ojw.CConvert.DoubleToStr(mat4[0, 1])  
                , Ojw.CConvert.DoubleToStr(mat4[0, 2]), Ojw.CConvert.DoubleToStr(mat4[0, 3])  
                , Ojw.CConvert.DoubleToStr(mat4[1, 0]), Ojw.CConvert.DoubleToStr(mat4[1, 1])  
                , Ojw.CConvert.DoubleToStr(mat4[1, 2]), Ojw.CConvert.DoubleToStr(mat4[1, 3])  
                , Ojw.CConvert.DoubleToStr(mat4[2, 0]), Ojw.CConvert.DoubleToStr(mat4[2, 1])  
                , Ojw.CConvert.DoubleToStr(mat4[2, 2]), Ojw.CConvert.DoubleToStr(mat4[2, 3])  
                , Ojw.CConvert.DoubleToStr(mat4[2, 2]), Ojw.CConvert.DoubleToStr(mat4[2, 3])  
                , Ojw.CConvert.DoubleToStr(mat4[3, 0]), Ojw.CConvert.DoubleToStr(mat4[3, 1])  
                , Ojw.CConvert.DoubleToStr(mat4[3, 2]), Ojw.CConvert.DoubleToStr(mat4[3, 3]));  
            //CalcRot(matrix) ??(회전이 적용이 안됨)  
            double[,] mat5;  
            bool rot1 = Ojw.CMath.CalcRot(0 , 90, 0, mat4, out mat5);  
            Ojw.CMessage.Write("\r\n{0}, {1}, {2}, {3}; \r\n{4}, {5}, {6}, {7}; \r\n{8}, {9}, {10}, {11}; \r\n{12}, {13}, {14}, {15};"  
                , Ojw.CConvert.DoubleToStr(mat5[0, 0]), Ojw.CConvert.DoubleToStr(mat5[0, 1])  
                , Ojw.CConvert.DoubleToStr(mat5[0, 2]), Ojw.CConvert.DoubleToStr(mat5[0, 3])  
                , Ojw.CConvert.DoubleToStr(mat5[1, 0]), Ojw.CConvert.DoubleToStr(mat5[1, 1])  
                , Ojw.CConvert.DoubleToStr(mat5[1, 2]), Ojw.CConvert.DoubleToStr(mat5[1, 3])  
                , Ojw.CConvert.DoubleToStr(mat5[2, 0]), Ojw.CConvert.DoubleToStr(mat5[2, 1])  
                , Ojw.CConvert.DoubleToStr(mat5[2, 2]), Ojw.CConvert.DoubleToStr(mat5[2, 3])  
                , Ojw.CConvert.DoubleToStr(mat5[3, 0]), Ojw.CConvert.DoubleToStr(mat5[3, 1])  
                , Ojw.CConvert.DoubleToStr(mat5[3, 2]), Ojw.CConvert.DoubleToStr(mat5[3, 3]));  
            //CalcRot(coordinate)   
            double dX = 1;  
            double dY = 0;  
            double dZ = 0;  
            bool rot2 = Ojw.CMath.CalcRot(0, 90, 0, ref dX, ref dY, ref dZ);  
            Ojw.CMessage.Write("\r\n{0}, {1}, {2}", dX, dY, dZ);  
            //MakeMatrix\_Rot ??(CalcRot(matrix)에 문제가 있음)  
            double[,] mat6;  
            bool makeMat\_Rot1 = Ojw.CMath.MakeMatrix\_Rot(0, 0, 90, out mat6);  
            Ojw.CMessage.Write("\r\n{0}, {1}, {2}, {3}; \r\n{4}, {5}, {6}, {7}; \r\n{8}, {9}, {10}, {11}; \r\n{12}, {13}, {14}, {15};"  
                , Ojw.CConvert.DoubleToStr(mat6[0, 0]), Ojw.CConvert.DoubleToStr(mat6[0, 1])  
                , Ojw.CConvert.DoubleToStr(mat6[0, 2]), Ojw.CConvert.DoubleToStr(mat6[0, 3])  
                , Ojw.CConvert.DoubleToStr(mat6[1, 0]), Ojw.CConvert.DoubleToStr(mat6[1, 1])  
                , Ojw.CConvert.DoubleToStr(mat6[1, 2]), Ojw.CConvert.DoubleToStr(mat6[1, 3])  
                , Ojw.CConvert.DoubleToStr(mat6[2, 0]), Ojw.CConvert.DoubleToStr(mat6[2, 1])  
                , Ojw.CConvert.DoubleToStr(mat6[2, 2]), Ojw.CConvert.DoubleToStr(mat6[2, 3])  
                , Ojw.CConvert.DoubleToStr(mat6[3, 0]), Ojw.CConvert.DoubleToStr(mat6[3, 1])  
                , Ojw.CConvert.DoubleToStr(mat6[3, 2]), Ojw.CConvert.DoubleToStr(mat6[3, 3]));  
            //MakeMatrix  
            double[,] mat7;  
            bool makeMat1 = Ojw.CMath.MakeMatrix(1, 2, 3, out mat7);  
            Ojw.CMessage.Write("\r\n{0}, {1}, {2}, {3}; \r\n{4}, {5}, {6}, {7}; \r\n{8}, {9}, {10}, {11}; \r\n{12}, {13}, {14}, {15};"  
                , Ojw.CConvert.DoubleToStr(mat7[0, 0]), Ojw.CConvert.DoubleToStr(mat7[0, 1])  
                , Ojw.CConvert.DoubleToStr(mat7[0, 2]), Ojw.CConvert.DoubleToStr(mat7[0, 3])  
                , Ojw.CConvert.DoubleToStr(mat7[1, 0]), Ojw.CConvert.DoubleToStr(mat7[1, 1])  
                , Ojw.CConvert.DoubleToStr(mat7[1, 2]), Ojw.CConvert.DoubleToStr(mat7[1, 3])  
                , Ojw.CConvert.DoubleToStr(mat7[2, 0]), Ojw.CConvert.DoubleToStr(mat7[2, 1])  
                , Ojw.CConvert.DoubleToStr(mat7[2, 2]), Ojw.CConvert.DoubleToStr(mat7[2, 3])  
                , Ojw.CConvert.DoubleToStr(mat7[3, 0]), Ojw.CConvert.DoubleToStr(mat7[3, 1])  
                , Ojw.CConvert.DoubleToStr(mat7[3, 2]), Ojw.CConvert.DoubleToStr(mat7[3, 3]));  
            //CompareMatrix  
            bool compMat1 = Ojw.CMath.CompareMatrix(2, mat1, mat2);  
            Ojw.CMessage.Write("\r\n{0}", Ojw.CConvert.BoolToStr(compMat1));  
        }



[그림 III-18]

1.83690953073357E-16등은 매우 작은 숫자로 사실상 0을 의미합니다.

**※RPM 계산**

Ojw.CMath.CalcRps(float fDeltaAngle, float fMillisecond);//각도(degree)와 밀리초를 받고 rps(rotate per second, 초당 회전수)를 반환합니다.  
※검증이 되지 않았습니다.

Ojw.CMath.CalcRpsCalcTime(float fDeltaAngle, float fRps);//각도(degree)와 rps(rotate per second, 초당 회전수)를 받고 회전 시간을 반환합니다.  
※검증이 되지 않았습니다.

**※필터**

Ojw.CMath.LowPassFilter(float fAlpha, float fResult\_Prev, float fCurrent\_Value);//  
들어오는 입력의 갑작스런 변화를 자제시켜주는 로우패스 필터입니다. float으로 값을 받고 float으로 반환합니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.CMath.LowPassFilter\_vd(double fAlpha, double fResult\_Prev, double fCurrent\_Value);//  
들어오는 입력의 갑작스런 변화를 자제시켜주는 로우패스 필터입니다. double으로 값을 받고 double으로 반환합니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

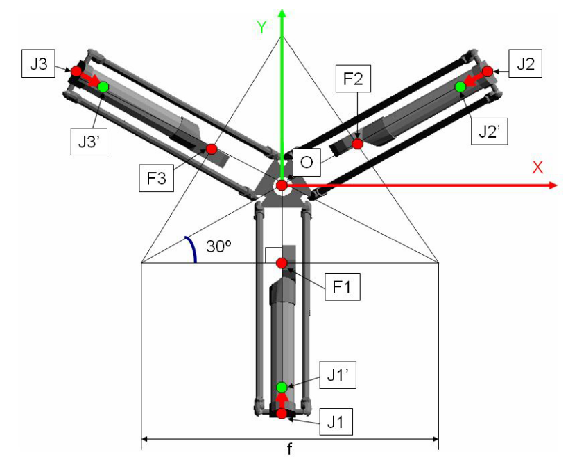
Ojw.CMath.HighPassFilter(float fAlpha, float fResult\_Prev, float fCurrent\_Value, float fPrevious\_Value);//  
※아직 검증되지 않았습니다.

Ojw.CMath.HighPassFilter\_vd(double fAlpha, double fResult\_Prev, double fCurrent\_Value, double fPrevious\_Value);//  
※아직 검증되지 않았습니다.

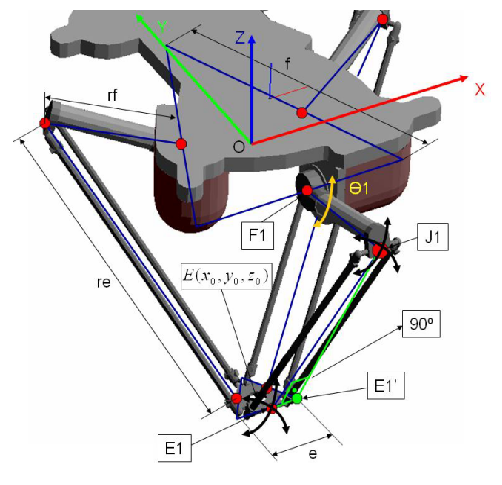
Ojw.CMath.ComplementaryFilter(short[] asAcc, short[] gyrData, ref float fRoll, ref float fPitch);//  
※아직 검증되지 않았습니다.

이 Delta Parallel 3 DOF Kinematics의 계산과 그림은 이동현 차장(동부로봇)님의 논문을 참고하였습니다.

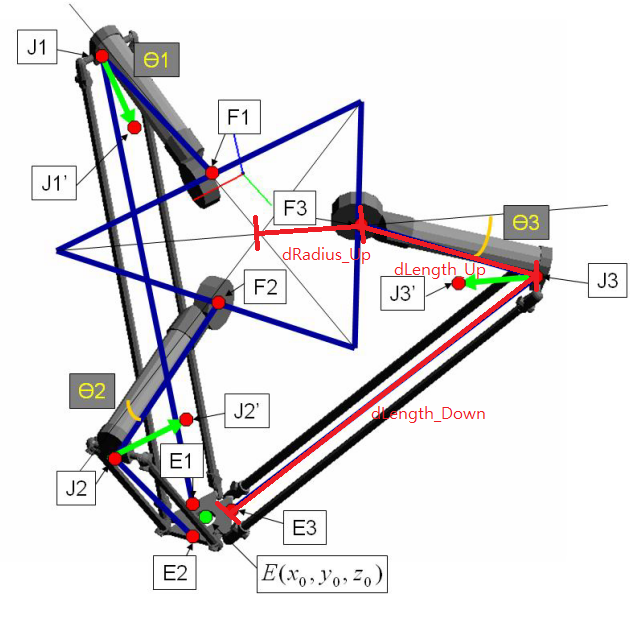
**※Delta Parallel 3 DOF Kinematics**



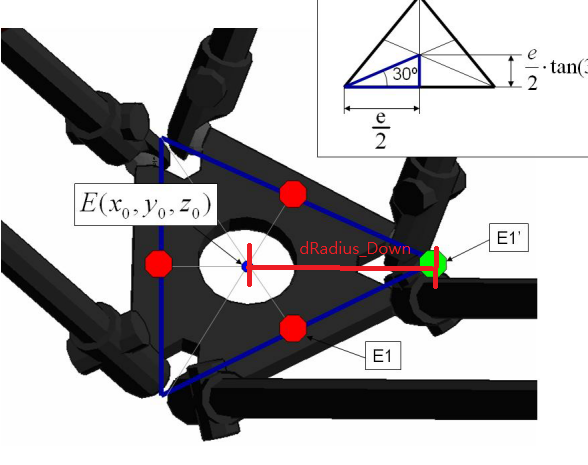
[그림 III-19]



[그림 III-20]

Ojw.CMath.Delta\_Parallel\_Init(double dRadius\_Up, double dLength\_Up, double dLength\_Down, double dRadius\_Down);//   
평행 델타 로봇의 운동학을 풀기 위한 기본 정보를 입력합니다. dRadius\_Up은 윗판 반지름 길이, dLength\_Up은 윗링크 반지름 길이, dLength\_Down은 아랫링크 반지름 길이, dRadius\_Down은 아랫판 반지름 길이를 의미합니다.

[그림 III-21]



[그림 III-22]  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.CMath.Delta\_Parallel\_InverseKinematics(double dX, double dY, double dZ, out double dAngle0, out double dAngle1, out double dAngle2);//  
x, y, z의 좌표 정보를 입력하면 각도를 산출해 내는 역운동학을 풀어냅니다. 계산 후 \_Parallel\_InverseKinematics 함수 자체는 true를 반환합니다. \_Parallel\_Init을 하지 않으면 false를 반환합니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.CMath.Delta\_Parallel\_ForwardKinematics(double dM0, double dM1, double dM2, out double dX, out double dY, out doubledZ);//  
각도를 입력하면 좌표 정보를 산출해 내는 정운동학을 풀어냅니다. 계산 후 \_Parall\_ForwardKinematics 함수 자체는 true를 반환합니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

**[참고자료]**

[[1] 마이크로소프트 참조 카탈로그](https://msdn.microsoft.com/ko-kr/library/)

<https://msdn.microsoft.com/ko-kr/library/>

[[2] 이동현 차장(동부로봇)님의 논문자료](http://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&ved=0ahUKEwiD2Nv_lsbKAhWBFqYKHe7FCzIQFghsMAk&url=http%3A%2F%2Fwww.jonmartinez.neositios.com%2Fdownloads.php%3Fid%3D1012189%26dId%3D0%26fId%3D61194&usg=AFQjCNENsRA6axGz4V_9XgRi7-QU03j-_A&bvm=bv.112454388,d.dGY&cad=rja)

<http://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&ved=0ahUKEwiD2Nv_lsbKAhWBFqYKHe7FCzIQFghsMAk&url=http%3A%2F%2Fwww.jonmartinez.neositios.com%2Fdownloads.php%3Fid%3D1012189%26dId%3D0%26fId%3D61194&usg=AFQjCNENsRA6axGz4V_9XgRi7-QU03j-_A&bvm=bv.112454388,d.dGY&cad=rja>

*End.*

*written by Yeseung Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [**[오픈 지그 웨어] 강좌 6 - Math** (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15070)|**작성자** [**ThinkPoint**](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=rhgkrsus1)

* 1. Encrypt

**Created Date: 2016.08.28**

**Modified Date: 2016.08.28**

**revision 1**

**키워드: 오픈지그웨어, 암호화, Encrypt, OpenJigWare**

**개발 환경: Microsoft windows, Visual studio C#**

안녕하세요.

"오픈지그웨어" 의 매뉴얼팀 김세훈입니다.

이번주에 저는 Encrypt 부분 작성을 맡았습니다.

**1. Encrypt**

Encrypt는 말그대로 암호화에 쓰입니다. 이 부분의 코드는 전체적으로 디폴트 마스터 패스워드, 디폴트 패스워드를 설정하는 부분과 헤더를 확인해 암호화 되어있는지 확인하는 부분, 암호화를 해제하는 부분으로 되어있습니다.

**2. 코드 예제 - 마스터 패스워드**

Encrypt는 말그대로 암호화에 쓰입니다. 이 부분의 코드는 전체적으로 디폴트 마스터 패스워드, 디폴트 패스워드를 설정하는 부분과 헤더를 확인해 암호화 되어있는지 확인하는 부분, 암호화를 해제하는 부분으로 되어있습니다.

        public class CEncryption  
        {  
            private static String m\_strMasterKey = "OJW5014"; // Default Master Key

m\_strMasterKey 에 마스터 패스워드가 설정됩니다.

            public static bool SetMasterKey(String strOldKey, String strNewKey)  
            {  
                if (m\_strMasterKey == strOldKey)  
                {  
                    m\_strMasterKey = strNewKey;  
                    return true;  
                }  
                return false;  
            }  
            public static void SetEncrypt(String strSet)  
            {  
                m\_bEncryptionEnable = (strSet == m\_strMasterKey) ? true : false;  
            }  
            public static bool IsEncrypt() { return m\_bEncryptionEnable; }

마스터 패스워드를 바꿀 수 있습니다.

**2. 코드 예제 - 패스워드**

**private static String m\_strLetter ="skfktakfTkadlebdrnlrdpekfdkanswkdhkfhtjfmtkakteldkslgkfTo.dlfjswjsckfhdjflsqortjddlslfmrhwjghfgqodltueh,akckasowpEmemftlfjvuelahtgkfshalgkslfk.sodlfmfdnlgkdudjdutqlsjrutofhtmamfduemfqwkfkfaodrkshsl,tkfkaakekgodutnqlslruskffhTnapvusdkszpgkrhwugkfEkfkalxlfk.";**

m\_strLetter에 일반 패스워드를 지정할 수 있습니다.

참고로 디폴트 키는 훈민정음(나랏말싸미...)을 영타로 친 것 입니다.

**3. 코드 예제 - 암호화 확인및 해제**

            public static byte[] Encryption(bool bEncryptionMode, byte[] byteData)  
            {  
                int nHeaderSize = 2; // "!" , oxff  
                if ((m\_bEncryptionEnable == false) && (bEncryptionMode == false)) return byteData;  
                // Check it first, header is ("!" , oxff) or not  
                if (byteData.Length < nHeaderSize) return byteData;  
                if ((byteData[0] == '!') && ((char)(byteData[1]) == 0xff))  
                {  
                    if (bEncryptionMode == true) return byteData  
                    // It'll change from encryption to normal  
                    else  
                    {  
                        // Remove Encryption Header  
                        byte[] byteData2 = new byte[byteData.Length - nHeaderSize];  
                        Array.Copy(byteData, nHeaderSize, byteData2, 0, byteData2.Length);  
                        Array.Resize<byte>(ref byteData, byteData2.Length);  
                        Array.Copy(byteData2, byteData, byteData2.Length);  
                        byteData2 = null;

                        // Remove Encryption  
                        for (int i = 0; i < byteData.Length; i++)  
                            byteData[i] = (byte)LetterCode2Letter(i % (m\_strLetter.Length), byteData[i]);  
                    }  
                }

암호화 여부를 확인하는 방법은 헤더를 확인하는 것입니다. 이 코드에서는 oxff를 그것으로 사용하고 있습니다.

암호화여부를 확인하고, 암호화 되어있다면 해제 명령만 받고 암호화 되어있지 않다면 두가지 명령을 다 받습니다.

암호화를 할때는 보안이 필요없지만 해제시엔 보안이 필요합니다.

해제시엔 암호화 헤더또한 제거됩니다. 물론 암호화적용시엔 헤더가 추가되고요.

부족한 설명 봐주셔서 감사합니다. 향후 피드백을 통해 내용수정/추가할 예정입니다.

*End.*

*written by Sehoon Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [MANUAL - OpenJigWare.Encrypt (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15069) |**작성자** [버질](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=dusk2box)

* 1. File

**Created Date: 2016.09.26**

**Modified Date: 2016.09.26**

**revision 2**

**키워드: 오픈지그웨어, OpenJigWare, File**

**개발 환경: Microsoft Windows, Visual Studio Express C# 2015**

**1. File Class**

안녕하세요. ThinkPoint입니다. 이번에는 파일 클래스의 메뉴얼을 작성하도록 하겠습니다.

파일 클래스는 파일의 존재 여부를 체크하거나, 파일의 이름 또는 경로, 확장자 등을 불러오는 등의 파일 관련된 작업을 할 수 있는 클래스 입니다. 파일 이름은 COjw\_07\_File.cs입니다.

클래스는 Ojw.CFile 입니다.

**2. 함수 설명**

public static bool Ojw.CFile.IsFile(String strFile);  
파일이 있는지 없는지 확인합니다. 있으면 true, 아니면 false를 반환합니다. strFile에는 파일 이름 혹은 파일 경로가 들어갑니다.

public static bool Ojw.CFile.IsFiles(params String[] pstrFile);  
1차원 배열로 나열하며, 모두 있으면 true, 하나라도 없으면 false를 반환합니다.

public static String Ojw.CFile.GetTitle(String strPath);  
파일의 이름만 가져옵니다.

public static String Ojw.CFile.GetName(String strPath);  
확장자까지 합쳐서 파일의 이름을 가져옵니다.

public static String Ojw.CFile.GetPath(String strPath);  
파일의 경로를 가져옵니다.

public static String Ojw.CFile.GetExe(String strPath);  
파일의 확장자를 가져옵니다.

public static string Ojw.CFile.CheckAndMakeFolder(string strPath);  
strPath의 경로에 폴더가 있는지 확인하고, 없으면 그 경로의 폴더를 만듭니다.

public static string Ojw.CFile.CheckAndMakeFolder(string strPath, bool bSeparation, bool bYear4, bool bMonth2, bool bDay2,bool bTime);  
strPath의 경로에 폴더가 있는지 확인하고, 없으면 그 경로의 폴더를 만듭니다. 각각  
 bYear4, bMonth2, bDay2, bTime이 true이면 현재 년도, 월, 일, 시간의 이름을 가지고 strPath의 하위폴더를 만듭니다. true를 하지 않은 변수는 적용되지 않습니다. 만약 15th/Jan/2015, 05:33:00 pm의 시간에 bSeparation이 true라면  strPath의 하위폴더가 "2015\\01\\15\\173300\\"만큼 생기고, bSeparation이 false라면 strPath의 하위폴더 "20150115173300"가 하나 생깁니다.

public int Ojw.CFile.Load(int nCnt, String strFileName);  
파일의 내용을 리스트 메모리로 옮겨줍니다.

public bool Ojw.CFile.Save(String strFileName);  
리스트의 내용을 파일로 저장합니다.

public bool Ojw.CFile.GetData\_ListBox(ref ListBox lst);  
리스트박스 Ist의 데이터를 불러옵니다.

public String Ojw.CFile.CheckData(int nIndex);  
nIndex의 번호에 있는 데이터를 불러옵니다.

public String Ojw.CFile.GetData\_String(int nIndex);  
nIndex의 번호에 있는 데이터를 String으로 불러옵니다.

public int Ojw.CFile.GetData\_Int(int nIndex);  
nIndex의 번호에 있는 데이터를 int로 불러옵니다.

public float Ojw.CFile.GetData\_Float(int nIndex);  
nIndex의 번호에 있는 데이터를 float으로 불러옵니다.

public double Ojw.CFile.GetData\_Double(int nIndex);  
nIndex의 번호에 있는 데이터를 double으로 불러옵니다.

public bool Ojw.CFile.GetData\_Bool(int nIndex);  
nIndex의 번호에 있는 데이터를 bool으로 불러옵니다.

public bool Ojw.CFile.SetData\_String(int nIndex, String strValue);  
nIndex의 번호에 있는 데이터를 String의 "s" + strValue로 바꿔줍니다.

public bool Ojw.CFile.SetData\_Int(int nIndex, int nValue);  
nIndex의 번호에 있는 데이터를 Int의 "n" + nValue로 바꿔줍니다.

public bool Ojw.CFile.SetData\_Float(int nIndex, float fValue);  
nIndex의 번호에 있는 데이터를 float의 "f" + fValue로 바꿔줍니다.

public bool Ojw.CFile.SetData\_double(int nIndex, double dValue);  
nIndex의 번호에 있는 데이터를 double의 "d" + dValue로 바꿔줍니다.

public bool Ojw.CFile.SetData\_Bool(int nIndex, bool bValue);  
nIndex의 번호에 있는 데이터를 bool의 "b" + bValue로 바꿔줍니다.

*End.*

*written by Yeseung Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [**[오픈 지그 웨어] 강좌 13 - File** (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15236)|**작성자** [**ThinkPoint**](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=rhgkrsus1)

* 1. Timer

**Created Date: 2016.01.01**

**Modified Date: 2016.01.01**

**revision 1**

**키워드:  Open Jig Ware, Timer**

**개발 환경: Microsoft windows, Visual Studio 2015 C#**

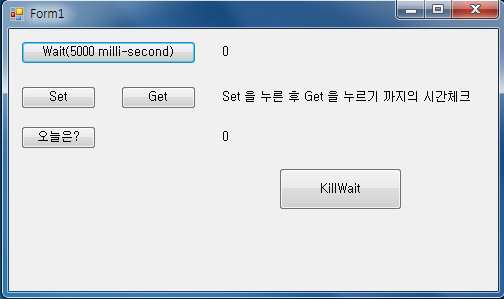
 안녕하세요. SSM입니다. 저번 'Convert 강좌'에 이어서 Timer강좌를 맡게되었습니다.

**1. Timer**

 Timer는 시간이나 날짜 등을 측정하는데 필요한 함수들을 만들어서 모아놓은 것 입니다.

**2. 프로그램 예제**

 이번에 다룰 예제는 총 4가지 기능을 가지고 있습니다. Timer도 다른 클래스들과 마찬가지로 참조추가 하는 것을 잊지 마시기 바랍니다.



[그림 III-23]

Visual Studio 2015(C#)을 이용하여, 위와 같은 형태로 만들어 줍니다.

// For Use  
//  1. 참조 - 참조추가 - 찾아보기 - DLL 선택(OpenJigWare.dll)  
//  2. add "using OpenJigWare" as follow  
using OpenJigWare;

"**using OpenJigWare;**" 를 반드시 적어주세요.(자세한 내용은 <http://cafe.naver.com/openrt/13764>를 참고)

namespace Ex4\_Timer  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }

        private void Form1\_Load(object sender, EventArgs e)  
        {  
        }

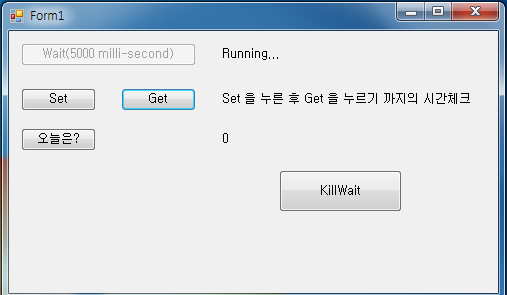
        private void btnWait\_Click(object sender, EventArgs e)  // [1]  
        {  
            btnWait.Enabled = false;  
            lbWait.Text = "Running...";  
            // Wait sample  
            Ojw.CTimer.Wait(5000);  
            lbWait.Text = "End";  
            btnWait.Enabled = true;  
        }

        private Ojw.CTimer m\_CTId\_0 = new Ojw.CTimer();  // [2]  
        private void btnSet\_Click(object sender, EventArgs e)  
        {  
            m\_CTId\_0.Set();  
        }  
        private void btnGet\_Click(object sender, EventArgs e)  
        {  
            long lData = m\_CTId\_0.Get();  
            lbGet.Text = lData.ToString() + " ms";  
        }

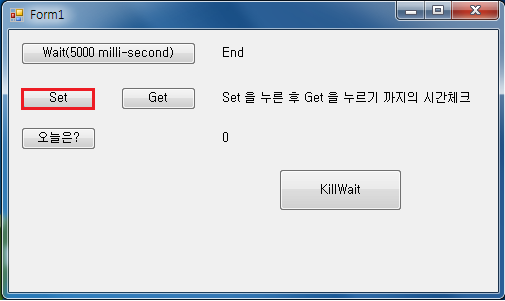
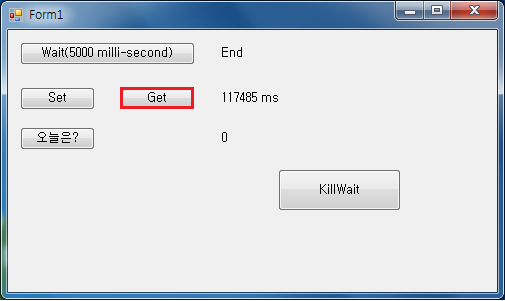
        private void btnCurrent\_Click(object sender, EventArgs e)  // [3]  
        {  
            lbCurrent.Text = Ojw.CTimer.GetYear() + "/" +  
                Ojw.CTimer.GetMonth() + "/" +  
                Ojw.CTimer.GetDay() +  
                " " +  
                Ojw.CTimer.GetHour() + ":" +  
                Ojw.CTimer.GetMinute() + ":" +  
                Ojw.CTimer.GetSecond();  
        }

        private void btnKill\_Click(object sender, EventArgs e)  // [4]  
        {  
            Ojw.CTimer.KillWait();  
        }  
    }  
}

[1] 버튼을 누르면 5초를 세는 기능입니다.

[그림 III-24][그림 III-25]

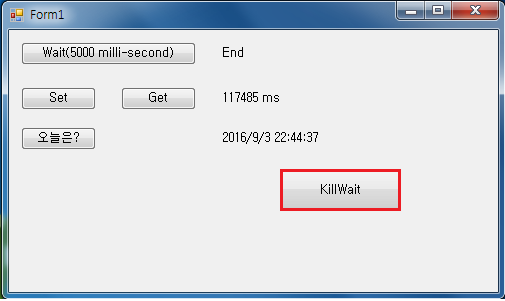
[2] 'Set'버튼을 누른 후 부터 'Get'버튼을 누르기 까지의 시간을 체크합니다.

[그림 III-26][그림 III-27]

[3] 현재의 년/월/일 시:분:초 에대한 정보가 나옵니다.

[그림 III-28]

[4] 'Wait'버튼과 'Set'버튼을 눌렀을 때 'KillWait'버튼을 누르면 중단됩니다.

[그림 III-29]

**3. Timer Source**

위에 있는 예제를 하나하나 파헤쳐 보겠습니다.

[1]

        private void btnWait\_Click(object sender, EventArgs e)  
        {  
            btnWait.Enabled = false;  // 'Wait'버튼을 누를 수 없게 합니다.  
            lbWait.Text = "Running...";  // 버튼 옆에 "Running..."이라는 텍스트 문자를 띄웁니다.  
            // Wait sample  
            Ojw.CTimer.Wait(5000);  // 5초 동안 위의 상태가 지속됩니다.  
            lbWait.Text = "End";  // "Running..."이라는 텍스트 문자를 "End"로 바꿔줍니다.  
            btnWait.Enabled = true;  // 다시'Wait'버튼을 누를 수 있게 합니다.  
        }

[2]

        private Ojw.CTimer m\_CTId\_0 = new Ojw.CTimer();  // 초기화합니다.  
        private void btnSet\_Click(object sender, EventArgs e)  
        {  
            m\_CTId\_0.Set();  // 컴퓨터의 현재 날짜와 시간을 가져와서 Timer를 생성합니다.  
        }  
        private void btnGet\_Click(object sender, EventArgs e)  
        {  
            long lData = m\_CTId\_0.Get();  // Timer 생성 후 현재까지의 시간 값을 IData에 저장합니다.  
            lbGet.Text = lData.ToString() + " ms";  // IData에 저장된 값을 텍스트 문자로 내보내고 "ms"를 붙여줍니다.  
        }

public static void Set(int nHandle)  
{  
Set\_Tick(nHandle, DateTime.Now.Ticks);  
}

public static long Get(int nHandle)  
{  
    if (m\_abTimer[nHandle] == true)  
    {  
        DateTime tmrTemp = DateTime.Now;  
        long temp = (long)tmrTemp.Ticks \* 100 / 1000000;  
        long temp\_Gap = temp - m\_alTimer[nHandle];  
        return temp\_Gap;  
    }  
    else return (long)(0x7fffffffffffffff);  
}

[3]

        private void btnCurrent\_Click(object sender, EventArgs e)  
        {  
            lbCurrent.Text = Ojw.CTimer.GetYear() + "/" +  // 현재 년도를 텍스트로 출력 후 "/"도 출력합니다.  
                Ojw.CTimer.GetMonth() + "/" +  // 현재 달을 텍스트로 출력 후 "/"를 출력합니다.  
                Ojw.CTimer.GetDay() +  // 현재 일을 텍스트로 출력합니다.  
                " " +                           // 한 칸 띄어줍니다.  
                Ojw.CTimer.GetHour() + ":" +  // 현재 시각을 텍스트로 출력한 후 ":"를 출력합니다.  
                Ojw.CTimer.GetMinute() + ":" +  // 현재 분을 텍스트로 출력한 후 ":"를 출력합니다.  
                Ojw.CTimer.GetSecond();  // 현재 초를 텍스트로 출력합니다.  
        }  // 위 과정을 거치면 현재의 '년/월/일 시:분:초' 형태로 텍스트가 출력됩니다.

public static int GetYear() { return DateTime.Now.Year; }  
public static int GetMonth() { return DateTime.Now.Month; }  
public static int GetDay() { return DateTime.Now.Day; }  
public static int GetHour() { return DateTime.Now.Hour; }  
public static int GetMinute() { return DateTime.Now.Minute; }  
public static int GetSecond() { return DateTime.Now.Second; }  
// 0 - 일(Sun), 1 - 월(Mon), 2 - 화(Tue), 3 - 수(Wed), 4 - 목(Thu), 5 - 금(Fri), 6 - 토(Sat), => -1 - 에러(Error)  
public static int GetWeek() { try { return int.Parse(DateTime.Now.DayOfWeek.ToString("d")); } catch { return -1; } }

[4]

        private void btnKill\_Click(object sender, EventArgs e)  
        {  
            Ojw.CTimer.KillWait();  
        }

public static void Wait(long t)  
{  
    long temp\_t = t, temp;  
    long temp\_1 = 0;  
    DateTime tmrTemp = DateTime.Now;  
    long temp\_2 = (long)tmrTemp.Ticks \* 100 / 1000000;  
    do  
    {  
        if ((m\_bProgEnd == true) || (m\_bProgEnd == true) || (m\_bStop == true) || (m\_bKillWait == true)) break;  
        // Delay  
        tmrTemp = DateTime.Now;  
        temp = (long)tmrTemp.Ticks \* 100 / 1000000;  
        Application.DoEvents();  
        temp\_1 = temp - temp\_2;  
    } while (temp\_1 <= temp\_t);  
    m\_bKillWait = false; // Only 1'st use(Kor: KillWait 은 한번만 수행한다.)  
}  
private static bool m\_bKillWait = false;  
public static bool IsKillWait() { return m\_bKillWait; }  
public static void KillWait() { m\_bKillWait = true; }

[4]항목은 아직 어려워서 분석하지 못했습니다. 질문 후에 수정하겠습니다.

*End.*

*written by Sanghyeok Lee*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [[오픈 지그 웨어] 강좌 8 - Timer (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15113) |**작성자** [SSM](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=dlfjsdksk2)

* 1. Register

1. Kinematics
2. 그리기
   1. 2D 그리기
   2. 3D
      1. 모델링 그리기
      2. 모터 정의하기
      3. 수식 정의하기
      4. 내부 툴 사용하기
         1. 3D 모델링 툴
         2. 모션 편집기
3. 통신
   1. Serial

**Created Date: 2016.01.01**

**Modified Date: 2016.01.01**

**revision 1**

**키워드:  Open Jig Ware, Serial**

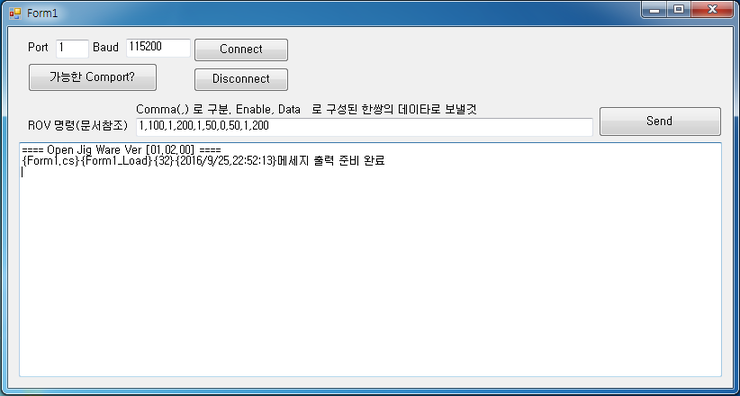
**개발 환경: Microsoft windows, Visual Studio 2015 C#**

**1. Serial**

  Serial 통신을 하기 위해 필요한 함수들을 모아놓은 Class입니다.

**2. 프로그램 예제**

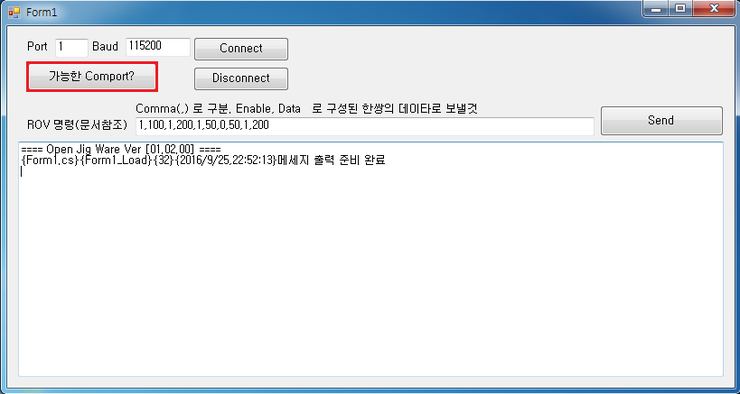
 이번에 다룰 Serial 예제는 총 4개의 버튼으로 구성되어 있습니다. 우선 각 버튼의 기능들에 대해 알아보도록 하겠습니다.



[그림 VI-1]

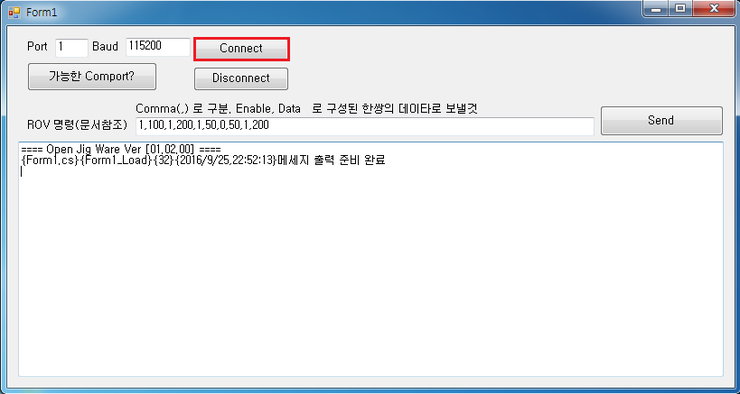
 Visual Studio 2015 C#을 이용하여, 위와 같이 인터페이스를 구성해줍니다.

[1] 가능한 Comport를 찾는 기능입니다.



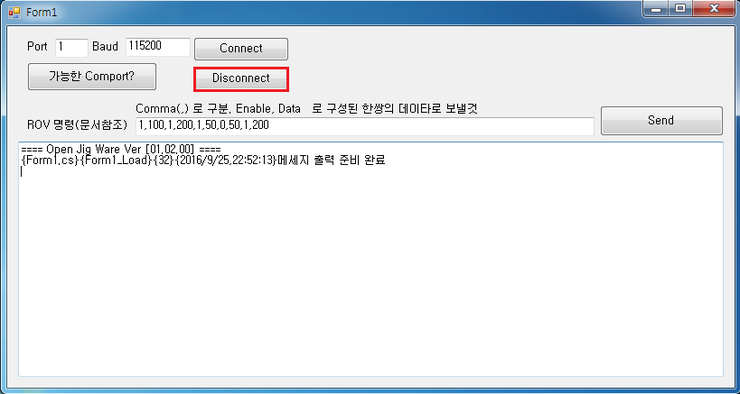
[그림 VI-2]

[2] 사용자가 원하는 Port와 Baud값을 설정하여 연결합니다.



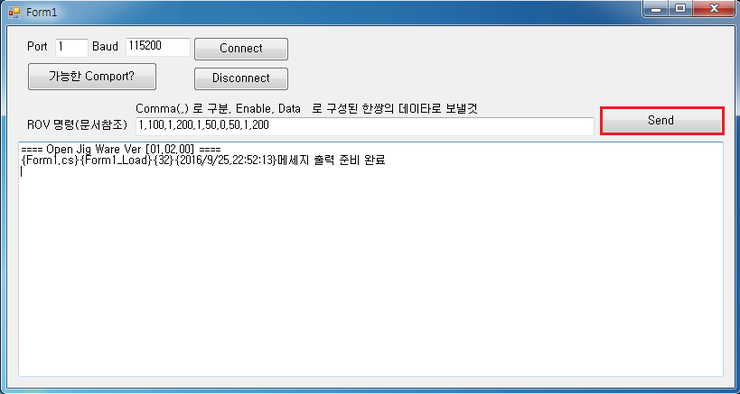
[그림 VI-3]

[3] 연결을 해제합니다.



[그림 VI-4]

[4] 텍스트창에 입력한 문자를 보내는 기능입니다.



[그림 VI-5]

**3. 소스코드 분석**

// Step 1 - 사용하기위한 선언  
using OpenJigWare;  
// 쓰레드를 사용하기 위해 선언  
using System.Threading;

namespace SerialTest  
{  
    public partial class Form1 : Form  
    {  
        // Step 2 - 변수 선언  
        private Ojw.CSerial m\_CSerial = new Ojw.CSerial();  // Ojw.CSerial의 이름을 m\_CSerial로 지정함  
                                                                                   // 그 후 초기화  
        // 데이타를 받아올 쓰레드 선언  
        Thread m\_thReceive = null;  // Thread값을 계속해서 갱신

        public Form1()  
        {  
            InitializeComponent();  // 초기화  
        }

        private void Form1\_Load(object sender, EventArgs e)  // 인터페이스가 로드되면  
        {  
            Ojw.CMessage.Init(txtMessage);  // 'txtMessage'라는 텍스트 박스를 사용 선언  
            Ojw.CMessage.Write("메세지 출력 준비 완료");  // 큰 따옴표 안의 메세지를 출력  
        }

Ojw.CMessage.Init();  
            // set the text box handle for writing history messages  
            public static void Init(TextBox txt)   
            {   
                m\_txtMessage = txt;  
                Write2("==== Open Jig Ware Ver [{0}] ====\r\n", SVersion\_T.strVersion);  
            }  
  
Ojw.CMessage.Write();  
            // Write -> write all informations and enter  
            public static void Write(string msg, params object[] objects)  
            {  
                try  
                {  
                    m\_nMessageStack++;  
                    if (objects != null)   
                    {  
                        if (objects.Length > 0)  
                        {  
                            StringBuilder sb = new StringBuilder();  
#if \_USING\_DOTNET\_3\_5  
                            sb.Remove(0, sb.Length);  
#else  
                            sb.Clear(); // Dotnet 4.0 이상에서만 사용  
#endif  
                            sb.AppendFormat(msg, objects);  
                            msg = sb.ToString();  
                        }  
                    }  
                    OjwDebugMessage(m\_bFile, m\_bFunction, m\_bLine, m\_bTime, true, msg);  
                }

        private void btnConnect\_Click(object sender, EventArgs e)  // Connect버튼을 누르면 아래를 실행  
        {  
            int nPort = Ojw.CConvert.StrToInt(txtPort.Text); // 문자를 숫자로...  
            int nBaudrate = Ojw.CConvert.StrToInt(txtBaudrate.Text);  
            if (m\_CSerial.IsConnect() == true)  // 만약 연결이 가능하다고 확인되면 아래를 출력  
            {  
                Ojw.CMessage.Write("이미 연결되어 있습니다. 굳이 다시 연결할 필요 없습니다.");  
                return;  
            }  
            if (m\_CSerial.Connect(nPort, nBaudrate) == true)  // 연결이 되어있으면 조건문 실행  
            {  
                Ojw.CMessage.Write("연결 성공!!!, Port={0}, BaudRate={1}", nPort, nBaudrate);  
                // Thread 를 돌려 데이타를 받을 함수를 지정하자.  
                m\_thReceive = new Thread(new ThreadStart(Thread\_Receive));  
                m\_thReceive.Start();  
            }  
            else Ojw.CMessage.Write("연결 실패");  
        }

 Ojw.CConvert.StrToInt();  
            public static int StrToInt(String strData)  
            {  
                try  
                {  
                    return int.Parse(strData);  // String형 데이터를 int형으로 바꿔서 반환  
                }  
                catch  
                {  
                    return 0;  
                }  
            }  
  
m\_CSerial.IsConnect()  
            public bool IsConnect() { if (m\_SerialPort == null) return false; return m\_SerialPort.IsOpen; }  
            // 포트에 연결이 되었는지 안되었는지 확인하는 소스  
  
m\_CSerial.Connect()  
            public bool Connect(int nPort, int nBaudRate)  
            //(int nPort, int nBaudRate, int nParity, int nDataBits, int nStopBits)  
            {  
                m\_SerialPort.PortName = "COM" + nPort.ToString();  
                m\_SerialPort.BaudRate = nBaudRate;  
                m\_SerialPort.Parity = Parity.None;  
                m\_SerialPort.DataBits = 8;  
                m\_SerialPort.StopBits = StopBits.One;  
                m\_SerialPort.ReceivedBytesThreshold = 1;  
                //m\_SerialPort.ReadExisting  
                //m\_SerialPort.ReadBufferSize = 256;  
                try  
                {  
                    m\_SerialPort.Open();  
                    if (IsConnect() == true)  // 연결이 불가능한 경우 catch로 넘어간다.  
                    {  
                          
                    }  
                }  
                catch(Exception ex)  
                {  
                    CMessage.Write\_Error("Port Open Error - " + ex.ToString());  
                    //m\_bConnect = false;  
                }  
                return IsConnect();  
            }

        #region Thread  
        private void Thread\_Receive()  
        {  
            Ojw.CMessage.Write("[Thread] is Started");  
            while (m\_CSerial.IsConnect() == true)  // 연결이 가능한 경우 while문을 실행  
            {  
                try  
                {  
                    if (m\_CSerial.GetBuffer\_Length() > 0)  
                    {  
                        byte data = m\_CSerial.GetByte();  
                        Ojw.CMessage.Write2("{0}", (char)data);  
                    }  
                }  
                catch  
                {  
                    Ojw.CMessage.Write("[Thread] Port Closed");  
                }  
            }  
            Ojw.CMessage.Write("[Thread] Closed Thread");  
        }  
        #endregion Thread

m\_CSerial.GetBuffer\_Length()  
            public int GetBuffer\_Length()  
            {  
                try  
                {  
                    return (((IsConnect() == true) && (m\_bClassEnd == false)) ? m\_SerialPort.BytesToRead : 0);  
                }  
                catch  
                {  
                    return 0;  
                }  
            }  
m\_CSerial.GetByte();  
            public byte GetByte() { return (byte)((IsConnect() == true) ? m\_SerialPort.ReadByte() : 0); }  
  
Ojw.CMessage.Write2  
            public static void Write2(string msg, params object[] objects)  
            {  
                try  
                {  
                    m\_nMessageStack++;  
                    if (objects != null)  
                    {  
                        if (objects.Length > 0)  
                        {  
                            StringBuilder sb = new StringBuilder();  
#if \_USING\_DOTNET\_3\_5  
                            sb.Remove(0, sb.Length);  
#else  
                            sb.Clear(); // Dotnet 4.0 이상에서만 사용  
#endif  
                            sb.AppendFormat(msg, objects);  
                            msg = sb.ToString();  
                        }  
                    }   
                    OjwDebugMessage(false, false, false, false, false, msg);  
                }  
                catch (Exception e)  
                {  
                    MessageBox.Show("[Message]" + e.ToString() + "\r\n");  
                }  
            }

        private void btnDisconnect\_Click(object sender, EventArgs e)  
        {  
            if (m\_CSerial.IsConnect() == true)  // 연결가능한 상태일때 아래를 실행  
            {  
                m\_CSerial.DisConnect();  
                Ojw.CMessage.Write("통신 종료");  
            }  
            else Ojw.CMessage.Write("통신이 오픈되어 있지 않습니다.");

// 연결 불가능한 경우 텍스트 메세지 출력        }

        }

m\_CSerial.DisConnect();  
            public void DisConnect() { if (IsConnect() == true) { m\_SerialPort.Close(); } }  
            //연결된 상태인 경우 시리얼 포트를 닫는다.

        // 보너스 기능 - 내 컴에 연결 가능한 컴포트가 무엇이 있는지 알아보자.  
        private void btnCheckComport\_Click(object sender, EventArgs e)  
        {  
            string [] pstrComport = null;  
            Ojw.CRegistry.GetSerialPort(out pstrComport, true, true);  
  
            if (pstrComport != null)  
            {  
                Ojw.CMessage.Write("발견된 Comport =>");  
                for (int i = 0; i < pstrComport.Length; i++)  
                {  
                    Ojw.CMessage.Write2(pstrComport[i]);  
                    Ojw.CMessage.Write2("\r\n"); // 줄바꿈을 하자  
                }  
            }  
            else Ojw.CMessage.Write("발견된 통신 포트가 없습니다.");  
        }

연결가능한 포트를 찾아주는 기능

        #region Send  
        private void btnSend\_Click(object sender, EventArgs e) // Send버튼을 누르면 아래를 실행  
        {  
            string[] pstrData = txtData.Text.Split(',');  
            int[] pnData = new int[pstrData.Length];  
            int i = 0;  
            Ojw.CMessage.Write("Data Send=>");  
            foreach (string strData in pstrData)  
            {  
                pnData[i] = Ojw.CConvert.StrToInt(strData);  
                Ojw.CMessage.Write2("0x{0},", Ojw.CConvert.IntToHex(pnData[i], 2));  
                i++;  
            }  
            Ojw.CMessage.Write2("\r\n");  
            if (m\_CSerial.IsConnect() == true)  // 포트가 연결가능한 상태이면 아래를 실행  
            {  
                SendValue(pnData);  
            }  
            else  // 아닌 경우 아래의 텍스트 메세지를 출력  
                Ojw.CMessage.Write("통신이 연결되지 않아 보내지 못했습니다.");  
        }  
        private void SendValue(params int[] pnData)  
        {  
            if (pnData.Length % 2 != 0)  
            {  
                Ojw.CMessage.Write\_Error("Data Size error");  
                return;  
            }  
            int nSize\_Header = 5;  
            int nSize\_Data = pnData.Length / 2 \* 3;  
            byte[] pBuffer = new byte[nSize\_Header + nSize\_Data];  
            byte[] pbyteTmp;  
            int i = 0;  
            int nCmd = 0x01;  
  
            // Header  
            pBuffer[i++] = (byte)0x02;  
            // Command  
            pBuffer[i++] = (byte)(nCmd & 0xff);  
            byte byCheckSum = pBuffer[i - 1];  
            // Size  
            pBuffer[i++] = (byte)(nSize\_Data & 0xff);  
            byCheckSum ^= pBuffer[i - 1];  
  
            for (int nPos = 0; nPos < pnData.Length / 2; nPos++)  
            {  
                pBuffer[i++] = (byte)(pnData[nPos \* 2] & 0xff);  
                byCheckSum ^= pBuffer[i - 1];  
                pbyteTmp = Ojw.CConvert.ShortToBytes((short)pnData[nPos \* 2 + 1]);  
                pBuffer[i++] = pbyteTmp[0];  
                byCheckSum ^= pBuffer[i - 1];  
                pBuffer[i++] = pbyteTmp[1];  
                byCheckSum ^= pBuffer[i - 1];  
            }  
            // CheckSum  
            pBuffer[i++] = byCheckSum; // Checksum  
            // Tail  
            pBuffer[i++] = (byte)0x03;  
            m\_CSerial.SendPacket(pBuffer);  
        }  
        #endregion Send  
  
        private void Form1\_FormClosing(object sender, FormClosingEventArgs e)  
        {  
            if (m\_CSerial.IsConnect() == true) m\_CSerial.DisConnect();  
        }  
    }

}

*End.*

*written by SangHyeok Lee*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [[오픈 지그 웨어] 강좌 12 - Serial (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15230) |**작성자** [SSM](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=dlfjsdksk2)

* 1. Socket

**Created Date: 2016.09.17**

**Modified Date: 2016.09.17**

**revision 1**

**키워드: 오픈지그웨어, 소켓, socket, OpenJigWare**

**개발 환경: Microsoft windows, Visual studio C#**

안녕하세요.

"오픈지그웨어" 의 매뉴얼팀 김세훈입니다.

이번주에 저는 socket부분 작성을 맡았습니다.

**1. Socket**

Socket은 라즈베리 파이나 아두이노같은 외부와 통신하는 소켓 통신 부분입니다. 원래는 구현이 어렵지만 시리얼 통신 수준으로 끌어내렸습니다. 네트워크 케이블 혹은 무선 네트워크로 연결되어 인터넷망을 이용합니다. 로직 구성은 서버와 연결하고, 클라이언트 생성, 담당 소켓 생성 하는 것과 그 이후 스레드를 닫고 서버와의 연결을 해제하는 것으로 되어있습니다. 이후 코드 예제 설명은 코드가 쓰인 순서대로 입니다.

**2. 코드예제 - 서버와 연결**

            public bool sock\_connected()  
            {  
                if (sock\_started() == false) return false;  
                if (m\_tcpServer\_Client == null) return false;  
                return m\_tcpServer\_Client.Connected;  
            }

서버가 클라이언트에 연결되어 있는지 확인하고

            public bool sock\_started()  
            {  
                if (m\_tcpServer == null) return false;  
                return true;  
            }

서버 구동이 시작되었는지 확인하고

            public bool sock\_get\_auth() { return m\_bAuth; }

인증여부를 확인합니다.

         public bool sock\_start(int nA, int nB, int nC, int nD, int nPort)  
            {  
                String strIP = nA.ToString() + "," + nB.ToString() + "," + nC.ToString() + "," + nD.ToString();  
                return sock\_start(strIP, nPort);  
            }

소켓을 생성합니다.

                    m\_tcpServer = new TcpListener(addr, nPort);  
                    m\_tcpServer.Start();]

서버

(클라이언트 생성시 스레드를 생성하는 부분은 주석처리 되었습니다. 원할 시 활성화 가능합니다.)

            public void WaitClient(bool bBlockingMode)  
            {  
                m\_tcpServer\_Client = m\_tcpServer.AcceptTcpClient();  
                m\_tcpServer\_Client.Client.Blocking = bBlockingMode;  
                m\_tcpServer\_Client.NoDelay = false;  
                m\_bwServer\_outData = new BinaryWriter(new BufferedStream(m\_tcpServer\_Client.GetStream()));  
                m\_bwServer\_inData = new BinaryReader(new BufferedStream(m\_tcpServer\_Client.GetStream()));

                CMessage.Write("준비완료");

                m\_nServer\_Seq = 0;

                m\_nCntAuth = 0;  
                m\_bAuth = false;

                m\_bThread\_Server = true;  
            }

클라이언트가 붙으면 담당 소켓을 생성

**3. 코드 예제 - 연결 종료**

**m\_bStartedServerThread = false;**

**스레드 종료  
                if (m\_bStartedServer == true) m\_tcpServer.Stop();  
               if (m\_tcpServer.Server.Connected == true) m\_tcpServer.Stop();  
                bool bConnected = sock\_connected();  
                bool bStarted = sock\_started();  
                if (bConnected == true) m\_tcpServer\_Client.Close();  
                if (bStarted == true) m\_tcpServer.Stop();**

**m\_tcpServer = null;**

**m\_bAuth = false;  
                m\_nCntAuth = 0;  
            }**

**열려있는 포트 닫음(Server)**

            public int sock\_get\_int16()  
            {  
  
                int nData;  
                byte[] byteData = sock\_get\_bytes(2);  
                nData = (byteData[2] << 8);  
                nData += (byteData[3]);  
                if (sock\_connected()) return nData; //2 Bytes  
                else return 0;  
            }

2byte 정수값을 반환

           public int sock\_get\_int32()  
            {  
                int nData;  
                byte[] byteData = sock\_get\_bytes(4);  
                nData = (byteData[0] << 24);  
                nData += (byteData[1] << 16);  
                nData += (byteData[2] << 8);  
                nData += (byteData[3]);  
  
                if (sock\_connected()) return nData; //4 Bytes  
                else return 0;  
            }

4byte 정수값을 반환

**4. 코드예제 - CSocket 클래스 정의**

            public NetworkStream stream;            // 네트워크 스트림  
            public int m\_nPort = 0;                        // 포트번호

            public bool m\_bConnect = false;          // 서버 접속 플래그  
            TcpClient m\_tcpClient;                       // TCP 클라이언트

           public bool Connect(String strIP, int nPort)  
            {  
                m\_tcpClient = new TcpClient();       // TCP 클라이언트 생성

                        m\_bConnect = false;  
                        return false;  
                    }  
                    m\_tcpClient.EndConnect(IResult);

서버에 접속 실패시

           public bool Connect(String strIP, int nPort, bool bBlockingMode)  
            {  
                m\_tcpClient = new TcpClient();       // TCP 클라이언트 생성

                try  
                {  
                    m\_tcpClient.Connect(strIP, nPort);  
                    m\_tcpClient.Client.Blocking = bBlockingMode;  
                }

서버 IP 주소와 포트 번호를 이용해 접속 시도

*End.*

*written by Sehoon Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [[오픈 지그 웨어] 강좌 11 - Socket (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15180) |**작성자** [버질](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=dusk2box)

1. 모터제어
   1. Herculex
   2. MX28
   3. DC2408
2. Mouse
3. 시간그래프

**Created Date: 2016.09.11**

**Modified Date: 2016.09.11**

**revision 1**

**키워드: 오픈지그웨어, OpenJigWare,  Graph**

**개발 환경: Microsoft Windows, Visual Studio Express C# 2015**

**1. Graph Class**

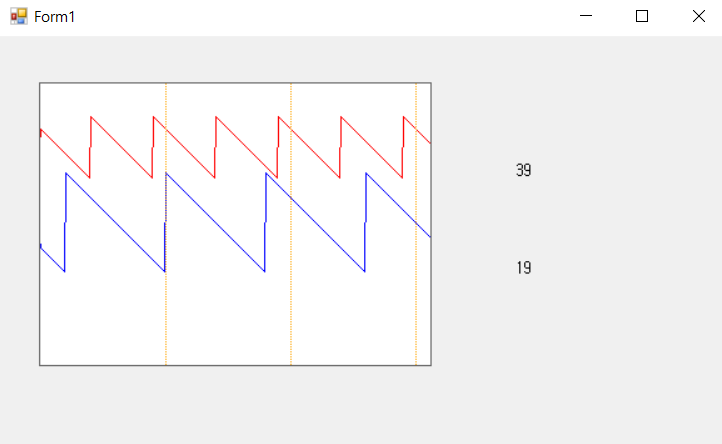
안녕하세요. ThinkPoint입니다. 이번에는 그래프 클래스의 메뉴얼을 작성하도록 하겠습니다.

Graph 클래스는 이름에서 볼 수 있다시피 그래프를 그리기 위한 클래스입니다. 파일 이름은 COjw\_21\_Graph.cs입니다.

클래스는 Ojw.CGraph 입니다.

**2. 예제**

이번에는 예제를 살펴보고 이 클래스가 어떤 것인지 알아보겠습니다.



[그림 IX-1]

디버그를 하면 이런식으로 그래프가 나타나게 됩니다. 그래프는 왼쪽에서 오른쪽으로 이동하고 오른쪽의 숫자는 가장 왼쪽 가장자리 그래프의 현재 수치를 나타냅니다. 위쪽 빨간색 그래프의 현재 수치는 39, 아래 파랜색 그래프의 현재 수치는 19입니다.

다음은 이 예제의 소스코드입니다.

        public Form1()  
        {  
            InitializeComponent();  
        }  
        private Ojw.CGraph m\_CGrap = null;//그래프를 초기화합니다.  
  
        private void Form1\_Load(object sender, EventArgs e)  
        {  
            // Graph  
            m\_CGrap = new Ojw.CGraph(lbDisp, lbDisp.Width, Color.White, null, Color.Red, Color.Blue);//새 그래프를 설정합니다. lbDisp는 그래프를 그리는 창이고 lable입니다. lbDisp.Width는 lbDisp의 너비이고, Color.White는 배경색, null의 자리에는 백그라운드 이미지를 bitmap으로 열 수 있습니다. 그리고 오른쪽에는 그래프의 색을 지정할 수 있습니다. 백그라운드 이미지 자리 뒤에 그래프의 수만큼 색이 있어야 합니다.  
            timer1.Enabled = true;//timer를 활성화 시킵니다.  
        }  
  
        private int m\_nTest0 = 0, m\_nTest1 = 0;  
        private void timer1\_Tick(object sender, EventArgs e)  
        {  
            // Making some random data for testing  
            int nTarget = 50;  
            m\_nTest0 = (m\_nTest0 + 1) % nTarget;//타이머에 시간 데이터를 만듭니다.  
            int nTarget2 = 80;  
            m\_nTest1 = (m\_nTest1 + 1) % nTarget2;//타이머에 시간 데이터를 만듭니다.  
  
            lbTest0.Text = m\_nTest0.ToString();//시간데이터를 string으로 변환합니다.  
            lbTest1.Text = m\_nTest1.ToString();//시간데이터를 string으로 변환합니다.  
  
            // push your datas...  
            m\_CGrap.Push(m\_nTest0, m\_nTest1);//시간데이터를 입력하면 그래프를 그리기 위해 계산하는 함수입니다. 시간데이터의 개수가 그래프의 개수를 결정합니다.  
  
            // Drawing...  
            m\_CGrap.OjwDraw();//실제 그래프를 그리는 함수입니다.  
        }



[그림 IX-2]

**3. 함수 설명**

public CGraph(Label lbDraw, int nHistorySize, Color cBackColor, Bitmap bmpBack, params Color[] acModel);  
//이 함수는 클래스와 이름이 같습니다. 그래프의 설정을 담당합니다. lbDraw는 그래프를 그리는 창이고 lable으로 받습니다. nHistorySize는 그림의 너비이고, cBackColor는 배경색, bmpBack의 자리에는 백그라운드 이미지를 Bitmap으로 열 수 있습니다. 그리고 acModel은 그래프의 색을 1차원 배열로 지정할 수 있습니다. acModel자리에 그래프의 수만큼 색을 받아야 합니다.

public void Push(params int [] anData); //시간데이터를 anData에 1차원 배열로 입력하면 원소의 수만큼 그래프를 계산합니다.

public void OjwDraw(); //Push에서 계산한 것을 바탕으로 그래프를 실제로 그려냅니다.

※자세한 사항은 나중에 업데이트 할 예정입니다.

**[참고자료]**

[[1] 마이크로소프트 참조 카탈로그](https://msdn.microsoft.com/ko-kr/library/)

<https://msdn.microsoft.com/ko-kr/library/>

*End.*

*written by Yeseung Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [**[오픈 지그 웨어] 강좌 10 - Graph** (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15164)|**작성자** [**ThinkPoint**](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=rhgkrsus1)

1. Struct

**Created Date: 2016.09.04**

**Modified Date: 2016.09.04**

**revision 3**

**키워드: 오픈지그웨어, OpenJigWare, Struct**

**개발 환경: Microsoft Windows, Visual Studio C# 2015**

**1. Struct Class**

안녕하세요. ThinkPoint입니다. 이번에는 Struct 클래스를 다루어 보도록 하겠습니다.

Struct 클래스는 로봇과 관련된 수학 정보, 모터 정보, 모션 프레임 등 여러가지 구조체들을 다룰 수 있습니다. 구조체에 대한 정보는 <https://msdn.microsoft.com/ko-kr/library/0taef578.aspx>에서 알 수 있습니다. 파일 이름은 COjw\_01\_Struct.cs입니다.

클래스는 Ojw에서 바로 쓰시면 됩니다.

**2. 구조체 설명**

**※구조체**

Ojw.SEncryption\_t  
byte[]의 1차원 배열을 SEncryption\_t.byteEncryption로 받습니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.SPoint2D\_t  
2차원 좌표의 정보(x좌표, y좌표와 데이터)를 int로 받습니다. x좌표는 SPoint2D\_t.x, y좌표는 SPoint2D\_t.y, 데이터는 SPoint2D\_t.Data로 받습니다.

Ojw.SPoint3D\_t  
3차원 좌표의 정보(x좌표, y좌표, z좌표)를 int로 받습니다. 각각의 좌표를 SPoint3D\_t.x, SPoint3D\_t.y, SPoint3D\_t.z, 또는 Ojw.SPoint3D\_t(int nX, int nY, int nZ)로 받습니다.

Ojw.SVector\_t  
2차원 좌표의 정보(x좌표, y좌표)를 double로 받습니다. 각각의 좌표를 SVector\_t.x, SVector\_t.y 또는 Ojw.SVector\_t(double x,double y)로 받습니다. 또한 +, -, \*, / 연산자로 두개의 다른 SVector\_t의 각각의 x좌표 끼리, y좌표 끼리 연산할 수 있습니다. +는 덧셈, -는 뺄셈, \*는 곱셈, /는 나눗셈 입니다.

Ojw.SVertex\_t  
위의 Ojw.SVector\_t와 완전히 같습니다.

Ojw.SVector3D\_t  
3차원 좌표의 정보(x좌표, y좌표, z좌표)를 float 으로 받습니다. 각각의 좌표를 SVector3D\_t.x, SVector3D\_t.y, SVector3D\_t.z 또는 Ojw.SVector\_t(float x, float y, float z)로 받습니다. 또한 +, -, \*, / 연산자로 두개의 다른 SVector\_t의 각각의 x좌표 끼리, y좌표 끼리, z좌표 끼리 연산할 수 있습니다. +는 덧셈, -는 뺄셈, \*는 곱셈, /는 나눗셈 입니다.

Ojw.SVertex3D\_t  
위의 Ojw.SVector3D\_t와 완전히 같지만, double로 받습니다.

Ojw.STrackD\_t  
모터의 3차원 좌표와 연결된 모터 번호, 회전 방향 등을 받습니다. x좌표를 STrackD\_t.fX, y좌표를 STrackD\_t.fY, z좌표를 STrackD\_t.fR로 float으로 받습니다. 회전해야 할 좌표 기준점을 x좌표는 STrackD\_t.fCenter\_X, y좌표는 STrackD\_t.fCenter\_Y로 float으로 받습니다. 또한 연결된 모터 번호를 STrackD\_t.nConnectedAxis, 회전 모드를 STrackD\_t.nMode, 회전 방향을 STrackD\_t.nDir로 int로 받습니다. 회전모드에서 0은 변화없음, 1은 회전이고, 회전방향은 0은 정방향, 1은 역방향입니다. Ojw.STrackD\_t(float x, float y, float z, int axis, float fCx, float fCy, int mode, int dir)으로 값을 받을수도 있습니다. 여기서 axis는 STrackD\_t.nConnectedAxis, fCx는 STrackD\_t.fCenter\_X, fCy는 STrackD\_t.fCenter\_Y, mode는 STrackD\_t.nMode, dir은 STrackD\_t.nDir입니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.SAngle3D\_t  
모터 각도의 pan, tilt, swing을 float으로 받습니다. pan은 SAngle3D\_t.pan, tilt는 SAngle3D\_t.tilt, swing은 SAngle3D\_t.swing입니다. 또는 Ojw.SAngle3D\_t(float pan, float tilt, float swing)으로도 pan, tilt, swing을 받을 수 있습니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.SMotionTable\_t  
모션을 위한 정보들을 받습니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.SMotion\_t  
모션을 위한 정보들을 받습니다.  
※자세한 사항은 나중에 업데이트 할 예정입니다.

Ojw.SDhT\_t  
DH Notation을 위한 2차원 배열을 double[,]로 SDhT\_t.adT에 받습니다.

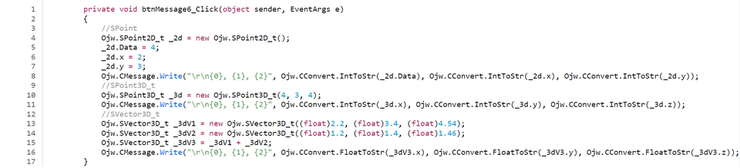
Ojw.SDhT\_Str\_t  
DH Notation을 위한 2차원 배열을 String[,]으로 SDhT\_t.aStrT에 받습니다.

Ojw.SOjwCode\_t  
※자세한 사항은 나중에 업데이트 할 예정입니다.

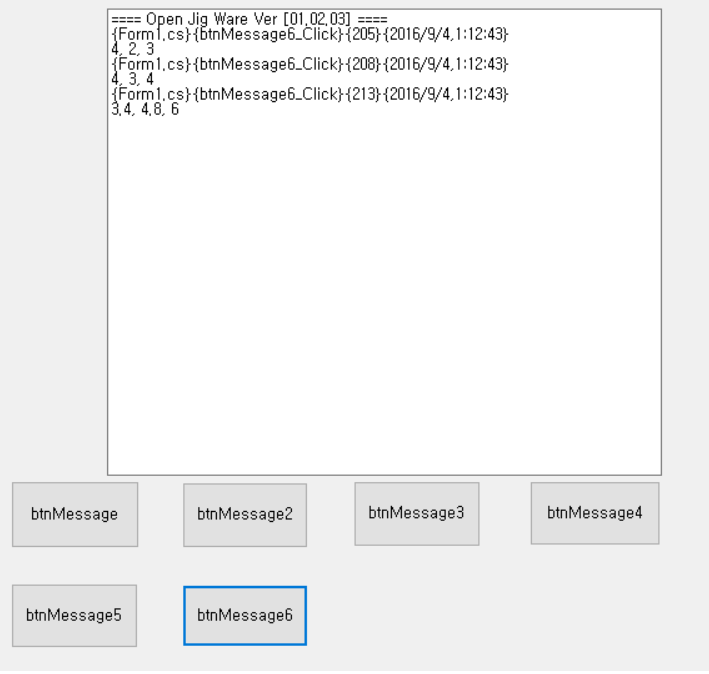
**※예제**

아래는 몇가지 예제와 실행 결과입니다.

        private void btnMessage6\_Click(object sender, EventArgs e)  
        {  
            //SPoint  
            Ojw.SPoint2D\_t \_2d = new Ojw.SPoint2D\_t();  
            \_2d.Data = 4;  
            \_2d.x = 2;  
            \_2d.y = 3;  
    Ojw.CMessage.Write("\r\n{0}, {1}, {2}", Ojw.CConvert.IntToStr(\_2d.Data), Ojw.CConvert.IntToStr(\_2d.x), Ojw.CConvert.IntToStr(\_2d.y));  
            //SPoint3D\_t  
            Ojw.SPoint3D\_t \_3d = new Ojw.SPoint3D\_t(4, 3, 4);  
   Ojw.CMessage.Write("\r\n{0}, {1}, {2}", Ojw.CConvert.IntToStr(\_3d.x), Ojw.CConvert.IntToStr(\_3d.y), Ojw.CConvert.IntToStr(\_3d.z));  
            //SVector3D\_t  
            Ojw.SVector3D\_t \_3dV1 = new Ojw.SVector3D\_t((float)2.2, (float)3.4, (float)4.54);  
            Ojw.SVector3D\_t \_3dV2 = new Ojw.SVector3D\_t((float)1.2, (float)1.4, (float)1.46);  
            Ojw.SVector3D\_t \_3dV3 = \_3dV1 + \_3dV2;  
      Ojw.CMessage.Write("\r\n{0}, {1}, {2}", Ojw.CConvert.FloatToStr(\_3dV3.x), Ojw.CConvert.FloatToStr(\_3dV3.y), Ojw.CConvert.FloatToStr(\_3dV3.z));  
        }



[그림 X-1]



[그림 X-2]

**[참고자료]**

[[1] 마이크로소프트 참조 카탈로그](https://msdn.microsoft.com/ko-kr/library/)

<https://msdn.microsoft.com/ko-kr/library/>

*End.*

*written by Yeseung Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]**  [**[오픈 지그 웨어] 강좌 7 - Struct** (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15114)|**작성자** [**ThinkPoint**](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=rhgkrsus1)

1. Streaming

**Created Date: 2016.09.02**

**Modified Date: 2016.09.02**

**revision 1**

**키워드: 오픈지그웨어, 스트리밍, streaming, OpenJigWare**

**개발 환경: Microsoft windows, Visual studio C#**

안녕하세요.

"오픈지그웨어" 의 매뉴얼팀 김세훈입니다.

이번주에 저는 streaming부분 작성을 맡았습니다.

**1. Streaming**

Streaming은 영상을 가져와 재생하는 데에 쓰입니다. 이 부분의 코드는 전체적으로 비디오 플레이어의 크기와 위치등 속성을 설정하는 부분과 영상을 여는 부분, 영상을 끄는 부분으로 되어있습니다.

**2. 코드예제 - 플레이어 속성 설정**

**private bool Init(Control ctrlDisp, int nX, int nY, int nWidth, int nHeight)  
            {  
                try  
                {  
                    if (IsReady == false) return false;  
                    m\_afgPlayer.Left = nX;  
                    m\_afgPlayer.Top = nY;  
                    m\_afgPlayer.Width = nWidth;  
                    m\_afgPlayer.Height = nHeight;  
                    m\_afgPlayer.Visible = true;  
                    m\_ctrlDisp = ctrlDisp;  
                    m\_ctrlDisp.Controls.Add(m\_afgPlayer);  
                    return true;**

m\_afgPlayer에 폭과 너비등 속성을 지정합니다

**3. 코드예제 - 동영상 가져오기**

**private bool Open(Control ctrlDisp, IVideoSource source, int nX, int nY, int nWidth, int nHeight)**

파일을 여는 부분입니다

**{  
                try  
                {  
                    if (IsReady == false) return false;  
                    if (m\_afgPlayer == null) m\_afgPlayer = new AForge.Controls.VideoSourcePlayer();  
                    // set busy cursor  
                    //this.Cursor = Cursors.WaitCursor;**

**// stop current video source  
                    if (IsStreaming == true) Close();**

**if (Init(ctrlDisp, nX, nY, nWidth, nHeight) == false) return false;**

이 부분을 통해 중복실행을 막습니다.

**// start new video source  
                    m\_afgPlayer.VideoSource = source;  
                    m\_afgPlayer.Start();**

**//this.Cursor = Cursors.Default;  
                    return true;  
                }**

그 후 새로운 비디오를 실행합니다.

**4. 코드예제 - 동영상 끄기**

**private bool Close()  
            {  
                try  
                {  
                    if (IsReady == false) return false;  
                    //ctrlDisp.Controls.IndexOf(m\_afgPlayer);//.Remove(m\_afgPlayer);  
                    if (m\_afgPlayer.VideoSource != null)**

끝을 내라는 신호를 줍니다.

**{  
                        m\_afgPlayer.SignalToStop();  
                        Ojw.CTimer CTmr = new CTimer();  
                        CTmr.Set();  
                        while (IsStreaming) { if (CTmr.Get() < 3000) Ojw.CTimer.Wait(100); else break; }**

그 후 스트리밍이 종료되었는지 확인하기 위해 3초의 딜레이를 줍니다.

**if (IsStreaming) m\_afgPlayer.Stop();**

종료가 확인 되면

**m\_afgPlayer.VideoSource = null;  
                        m\_ctrlDisp.Controls.Clear();  
                        m\_ctrlDisp = null;  
                        m\_afgPlayer = null;  
                        return true;  
                    }  
                    return false;  
                }**

플레이어를 종료합니다.

*End.*

*written by Sehoon Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [[오픈 지그 웨어] 강좌 9 - Streaming (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15115) |**작성자** [버질](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=dusk2box)

1. System

**Created Date: 2016.09.26**

**Modified Date: 2016.09.26**

**revision 1**

**키워드: 오픈지그웨어, 시스템, system, OpenJigWare**

**개발 환경: Microsoft windows, Visual studio C#**

안녕하세요.

"오픈지그웨어" 의 매뉴얼팀 김세훈입니다.

이번주에 저는 system부분 작성을 맡았습니다.

**1. System**

System은 shutdown, 중복실행제어 같은 시스템 제어 관련 코드입니다. 중복실행을 확인하는 부분, 중복실행을 허용 또는 불허하는 부분, 프로세스를 죽이는 부분으로 되어있습니다.

**2. 코드예제 - 중복실행 확인**

       System.Diagnostics.Process ps = new System.Diagnostics.Process();  
       String strTitle = Ojw.CFile.GetTitle(strProgram);  
       Process[] processes = System.Diagnostics.Process.GetProcesses();  
       bool bStarted = false;  
                  
       foreach (System.Diagnostics.Process process in processes) { if (process.ProcessName == strTitle) bStarted = true; }

중복실행 체크

                if (bStarted == true)  
                {  
                    Ojw.CMessage.Write("[warning]Still program is running... Can't run it. Check process first");  
  
                    return true;  
                }

중복 실행시

                else  
                {  
                    Ojw.CMessage.Write("Program duplication checking : OK");  
                }  
                return false;

중복실행 아닐시

**3. 코드 예제 - 프로세스 Kill**

         nRunningMode == 0 : Normal(중복허용, 안죽임), 1 : killothers(다른것 다 죽이고 혼자 살아남음), 2 : 중복시 안띄움)

                bool bRunning = false;  
                foreach (System.Diagnostics.Process process in processes) { if (process.ProcessName == strTitle) { if (nRunningMode == 1) process.CloseMainWindow(); else bRunning = true; } }  
                if (bRunning == true)  
                {  
                    if (nRunningMode != 2)  
                    {  
                        ps.StartInfo.FileName = strProgram;  
                        if (strArgument != null) ps.StartInfo.Arguments = strArgument;  
                        ps.Start();  
                    }  
                }  
                else  
                {  
                    ps.StartInfo.FileName = strProgram;  
                    if (strArgument != null) ps.StartInfo.Arguments = strArgument;  
                    ps.Start();  
                }

같은 이름을 가진 process를 모두 kill함

*End.*

*written by Sehoon Kim*

[*http://oroca.org*](http://oroca.org/)

[https://licensebuttons.net/l/by/4.0/88x31.png](http://creativecommons.org/licenses/by/4.0/deed.ko)

이 저작물은 [크리에이티브 커먼즈 저작자표시 4.0 국제 라이선스](http://creativecommons.org/licenses/by/4.0/deed.ko)에 따라 이용할 수 있습니다.

**[출처]** [OpenJigWare 강좌 14 - System (오픈소스 소프트웨어 & 하드웨어: 로봇 기술 공유 카페 (오로카))](http://cafe.naver.com/openrt/15237) |**작성자** [버질](http://cafe.naver.com/openrt.cafe?iframe_url=/CafeMemberNetworkView.nhn%3Fm=view%26memberid=dusk2box)

1. 응용