Ứng dụng MFC (Visual C++) trong mô phỏng Robot và hệ Cơ điện tử

Bài 8:
Đọc và hiển thị file STL

PHẠM MINH QUÂN

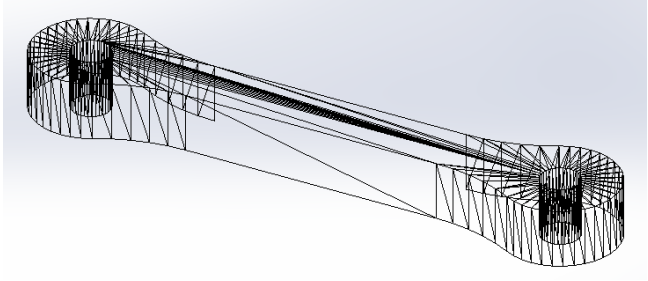
mquan.ph@gmail.com

Nội dung

1. Định dạng file .STL
2. Xuất file .STL từ phần mềm Solidworks
3. Sử dụng file .STL với đồ họa OpenGL
4. Mô phỏng Robot dùng file .STL

1. Định dạng file .STL

- File STL lưu trữ thông tin vật thể 3D thành tập hợp các mặt tam giác.

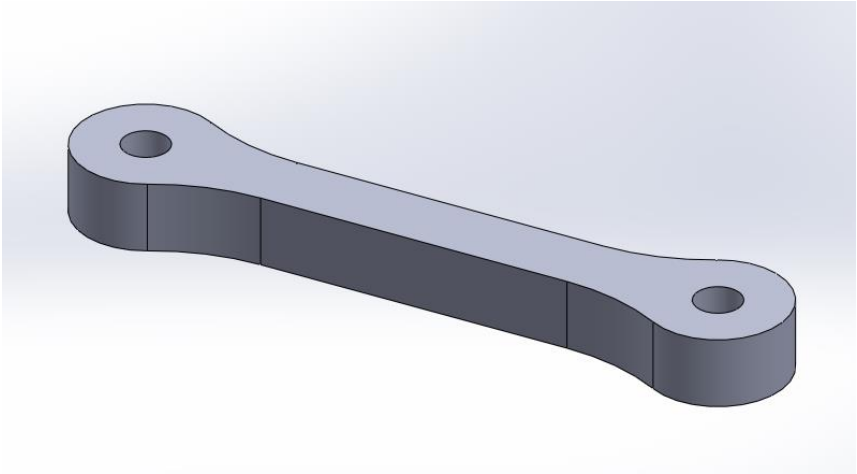


- Có 2 kiểu file STL: ASCII (Text) và Binary.
- Nội dung một file STL kiểu ASCII:

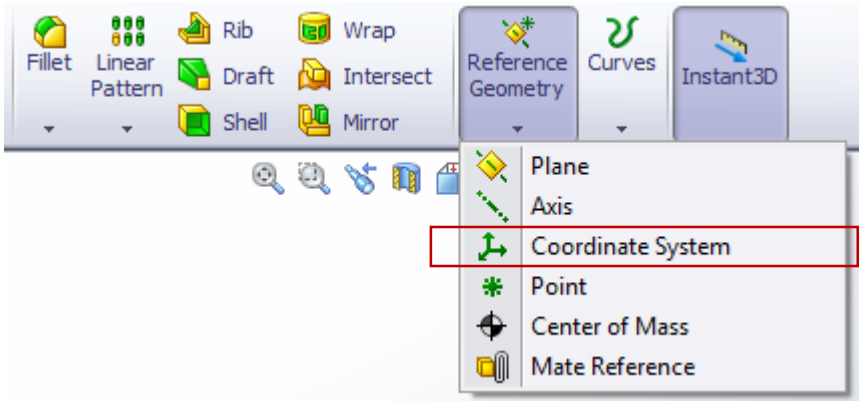
```
solid Link1                                                          // solid <tên vật thể>
   facet normal 0.000000e+000 -1.000000e+000 0.000000e+000           // Tọa độ vector pháp tuyến của mặt tam giác (hướng ra bên ngoài vật thể)
      outer loop
         vertex 2.269193e+001 6.705523e-007 -2.225557e+001           // Tọa độ các đỉnh của tam giác
         vertex 3.122153e+001 6.705523e-007 -1.910571e+001
         vertex 8.660254e+000 6.705523e-007 -5.000000e+000
      endloop
   endfacet
   facet normal 0.000000e+000 -1.000000e+000 0.000000e+000
      outer loop
         vertex 3.420201e+000 6.705523e-007 9.396927e+000
         vertex 5.000000e+000 6.705523e-007 8.660254e+000
         vertex 2.401519e+002 6.705523e-007 -1.736482e+000
      endloop
   endfacet
   ...
endsolid                                                             // Kết thúc file
```

2. Xuất file STL từ phần mềm Solidworks

- Mở file Solidworks

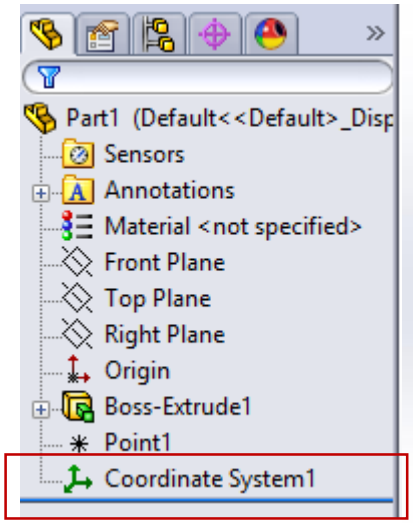


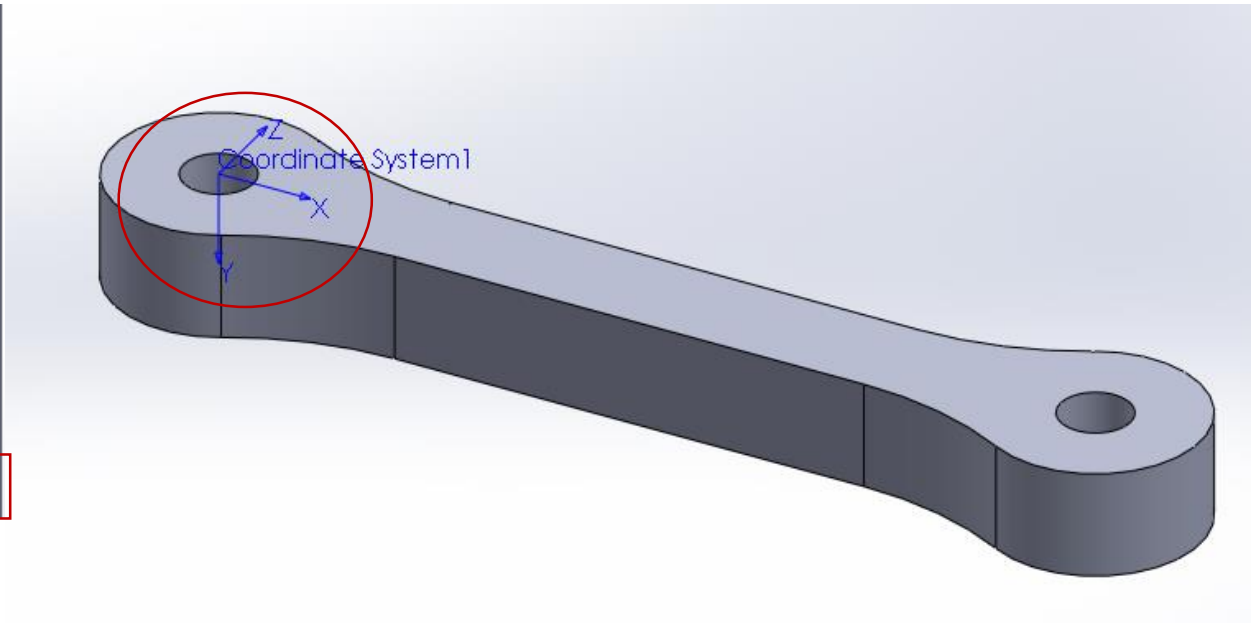
- Vẽ hệ tọa độ tham chiếu



2. Xuất file STL từ phần mềm Solidworks

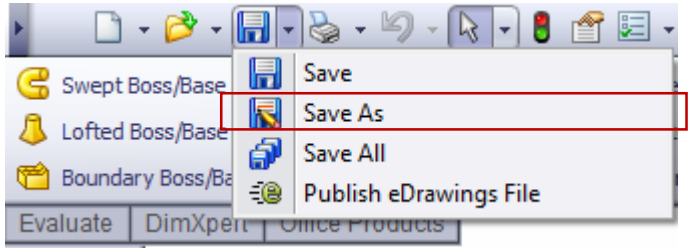
- Vẽ hệ tọa độ tham chiếu



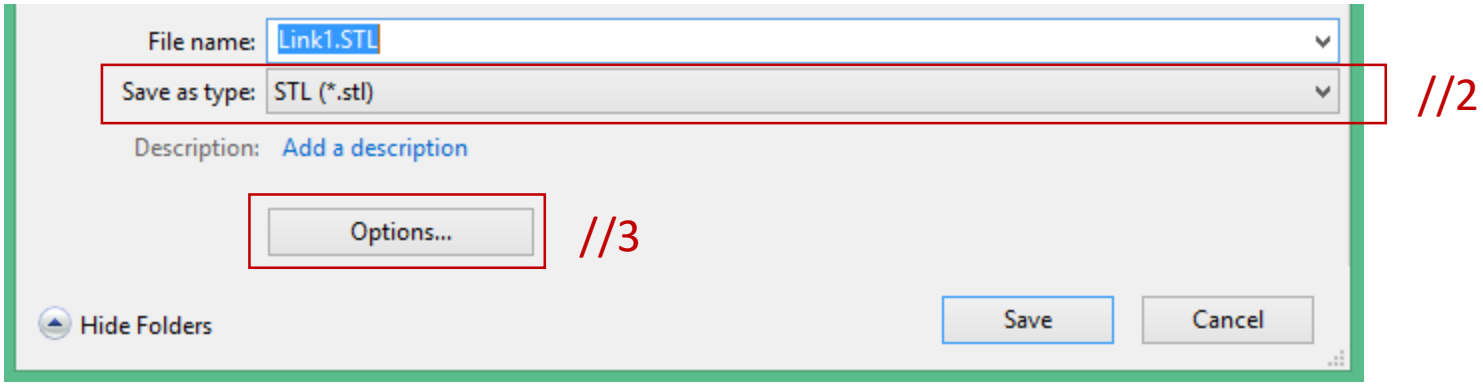


2. Xuất file STL từ phần mềm Solidworks

➢ Chọn Save as STL file



//1

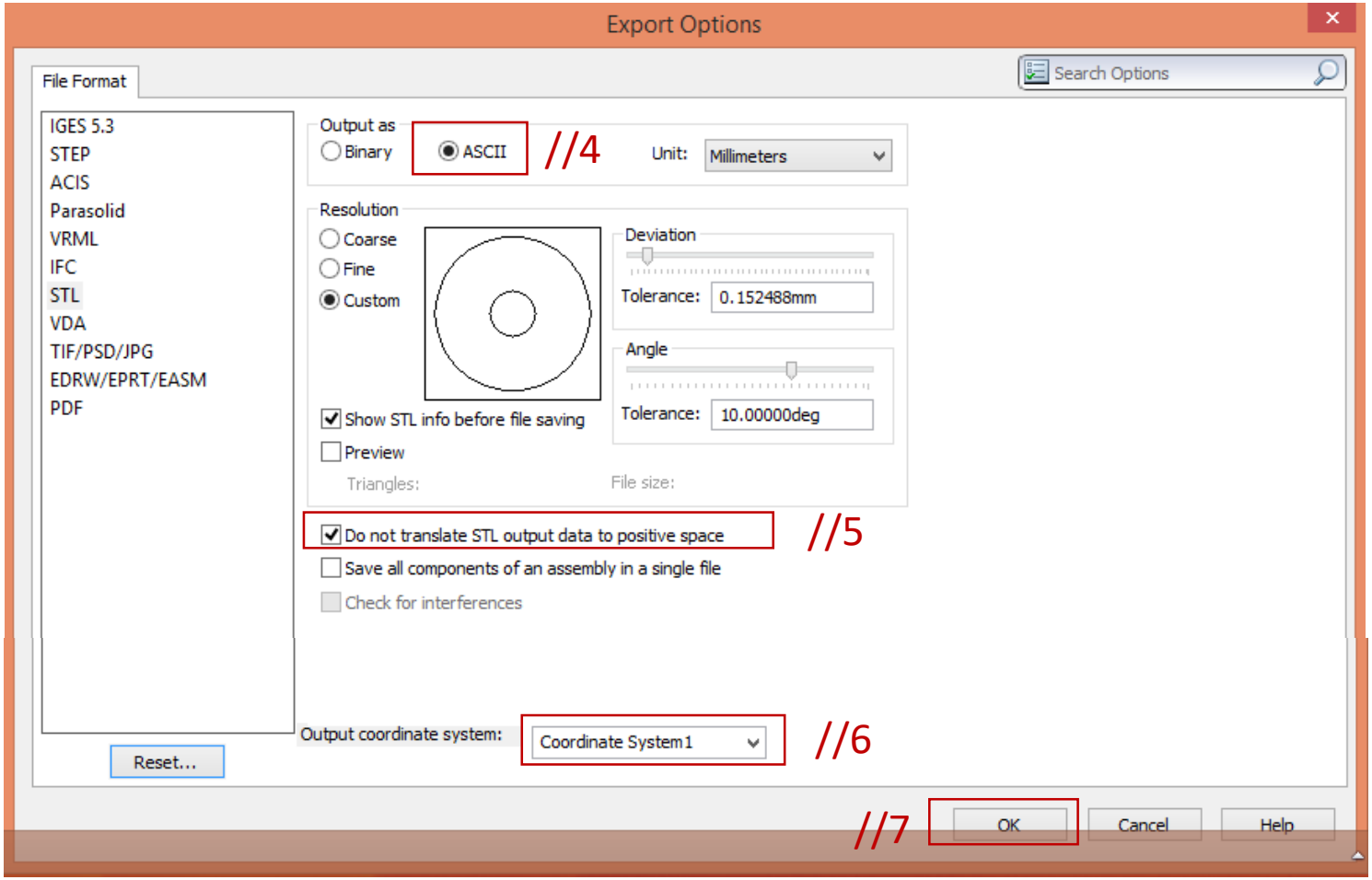


//2

//3

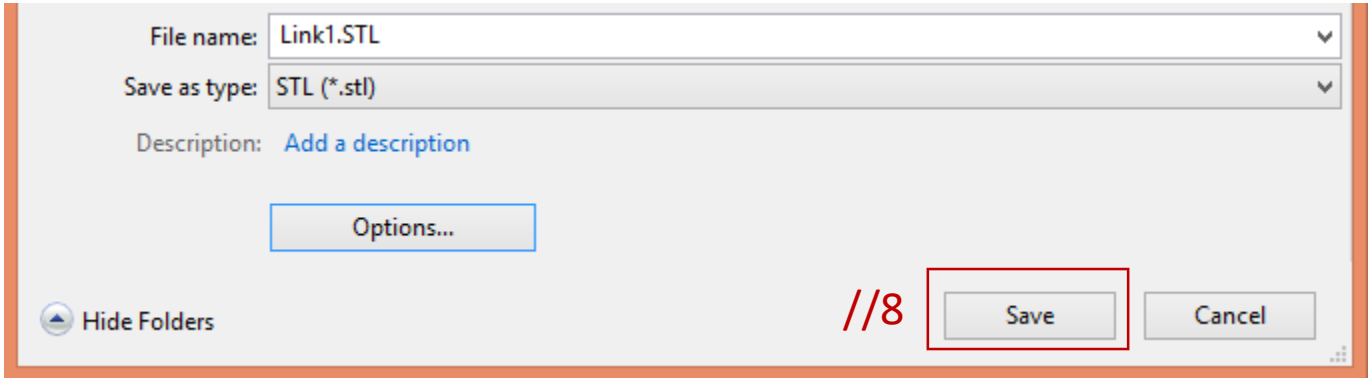
2. Xuất file STL từ phần mềm Solidworks

➢ Thay đổi một số Tùy chọn



2. Xuất file STL từ phần mềm Solidworks

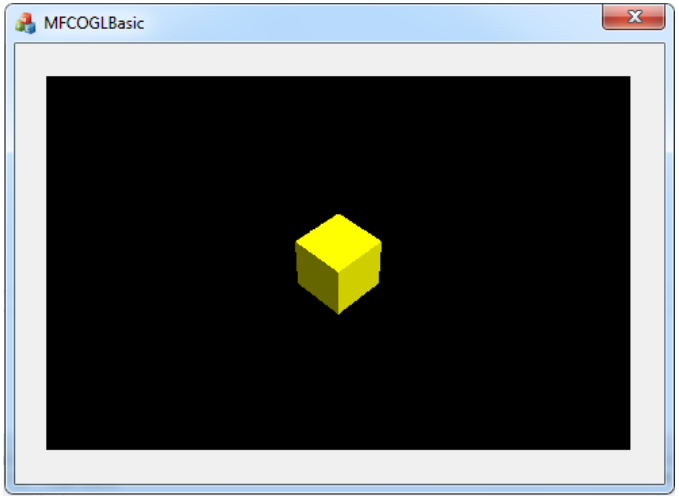
➢ Lưu file STL



3. Sử dụng file STL với đồ họa OpenGL

❑Tạo project và chuẩn bị dữ liệu

- ➢ Xây dựng chương trình MFC và thiết lập sẵn khung vẽ OpenGL như trong bài 6

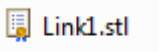


- ➢ Thêm các file GlobalFunctions.h và GlobalFunctions.cpp để khai báo và định nghĩa các hàm toàn cục

```
#include "stdafx.h"
#include "GlobalFunctions.h"
```

//Các file tiêu đề cần "include" trong GlobalFunctions.cpp

- ➢ Copy một file STL kiểu ASCII vào thư mục của project



//Ví dụ: file có tên Link1.stl

3. Sử dụng file STL với đồ họa OpenGL

❑Xây dựng các hàm xử lý dữ liệu STL

➢ Khai báo và định nghĩa hàm "*đọc file STL, lưu dữ liệu vào mảng*"

// Khai báo trong GlobalFunctions.h,
định nghĩa trong GlobalFunctions.cpp

```
bool ReadSTLFile(char *filename,float * & coord,long & count)
{
    float v1,v2,v3;
    float* vertex=NULL;
    long i=0;
    FILE *f;
    fopen_s(&f,filename,"rt");
    if(f== NULL) return false;
    else
    {
        vertex= (float*)realloc(vertex,0);
        while(fgetc(f)!= '\n'){}
        while(1)
        {
            if(fscanf_s(f,"%*s%*s%f%f%f",&v1,&v2,&v3)<3) break;
            vertex= (float*) realloc(vertex,(i+1)*12*sizeof(float));
            vertex[12*i]= v1; vertex[12*i+1]= v2; vertex[12*i+2]= v3;
            fscanf_s(f,"%*s%*s");
            for(int j=1;j<4;j++)
            {
                fscanf_s(f,"%*s%f%f%f",&v1,&v2,&v3);
                vertex[12*i+3*j]= v1; vertex[12*i+3*j+1]= v2; vertex[12*i+3*j+2]= v3;
            }
            fscanf_s(f,"%*s%*s");
            i++;
        }
        count= i*12;
        coord = vertex;
        vertex = NULL;
        fclose(f);
        return true;
    }
}
```

3. Sử dụng file STL với đồ họa OpenGL

❑Xây dựng các hàm xử lý dữ liệu STL

➢ Khai báo và định nghĩa hàm "*vẽ hình từ dữ liệu STL đã lưu trong mảng*"

```
void DrawSTLData(float *a, long c);
```
// File GlobalFunctions.h

```
void DrawSTLData(float *a, long c)
{
    int i,j;
    for(i=0; i<c/12; i++)
    {
        j=i*12;
        glBegin(GL_TRIANGLES);
        glNormal3f(a[j],a[j+1],a[j+2]);
        glVertex3f(a[j+3],a[j+4],a[j+5]);
        glVertex3f(a[j+6],a[j+7],a[j+8]);
        glVertex3f(a[j+9],a[j+10],a[j+11]);
        glEnd();
    }
}
```
// File GlobalFunctions.cpp

3. Sử dụng file STL với đồ họa OpenGL

❑Vẽ hình từ file STL

- Khai báo biến để lưu trữ mảng dữ liệu STL và số phần tử của mảng đó

```
float* STLvertex;
long STLcount;
```

// File OpenGLControl.h

- Đọc dữ liệu STL từ file ra và lưu vào mảng

```
#include "GlobalFunctions.h"
```

// File OpenGLControl.cpp

```
COpenGLControl::COpenGLControl(void)
{
    m_fPosX = 0.0f;     // X position of model in camera view
    m_fPosY = 0.0f;     // Y position of model in camera view
    m_fZoom = 10.0f;    // Zoom on model in camera view
    m_fRotX = 45.0f;     // Rotation on model in camera view
    m_fRotY = -45.0f;     // Rotation on model in camera view

    ReadSTLFile("Link1.STL", STLvertex, STLcount);
}
```

3. Sử dụng file STL với đồ họa OpenGL

❑Vẽ hình từ file STL

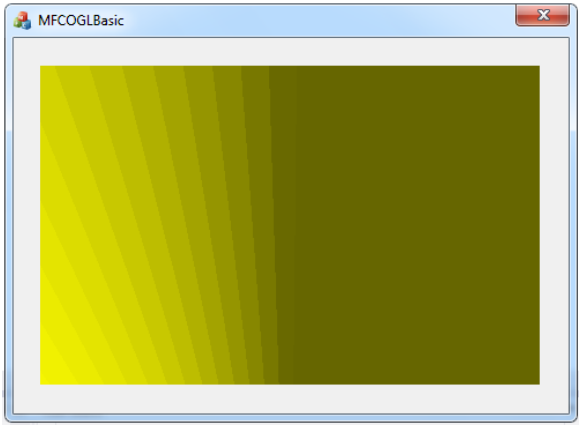
- ➢ Vẽ hình đồ họa OpenGL từ dữ liệu STL lưu trong mảng

// File OpenGLControl.cpp

```
void COpenGLControl::oglDrawScene(void)
{
    // Clear color and depth buffer bits
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    GLfloat mat_color[] = { 1.0, 1.0, 0.0 };
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0 };

    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_color);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialf(GL_FRONT, GL_SHININESS, 30);

    DrawSTLData(STLvertex, STLcount);

    // Swap buffers
    SwapBuffers(hdc);
}
```

Kết quả chạy:

//Ví dụ: hình trong file Link1.stl có kích thước ~ 150



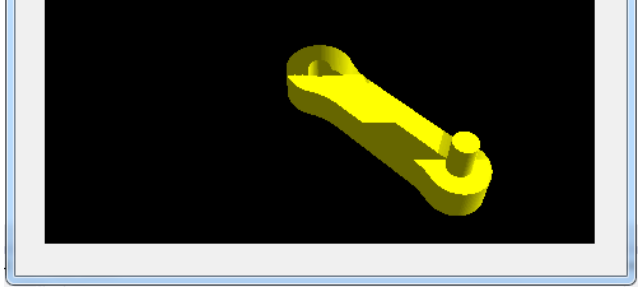
3. Sử dụng file STL với đồ họa OpenGL

❑Vẽ hình từ file STL

➢ Điều chỉnh các thông số OpenGL (zoom, zNear…) cho phù hợp với kích thước hình vẽ

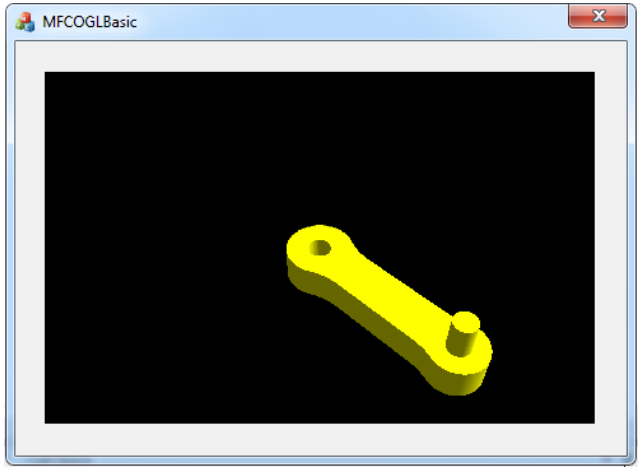
- Điều chỉnh zoom

```
COpenGLControl::COpenGLControl(void)
{
    m_fPosX = 0.0f;     // X position of model in camera view
    m_fPosY = 0.0f;     // Y position of model in camera view
    m_fZoom = 500.0f;   // Zoom on model in camera view
    m_fRotX = 45.0f;    // Rotation on model in camera view
    m_fRotY = -45.0f;    // Rotation on model in camera view

    ReadSTLFile("Link1.STL", STLvertex, STLcount);
}
```



- Điều chỉnh zNear, zFar

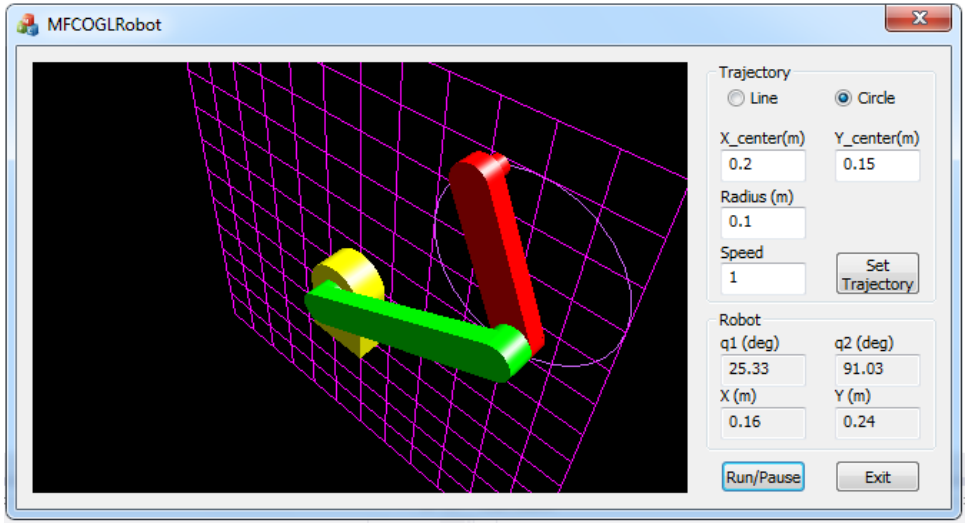
```
void COpenGLControl::OnSize(UINT nType, int cx, int cy)
{
    ...

    // Set our current view perspective
    gluPerspective(35.0f, (float)cx / (float)cy, 5.0f, 2000.0f);

    ...

}
```



4. Mô phỏng Robot dùng file STL

❑Tạo project và chuẩn bị dữ liệu

- Xây dựng chương trình mô phỏng Robot với đồ họa OpenGL như trong bài 7



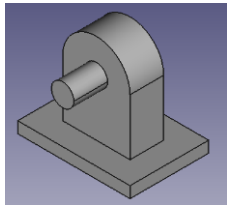
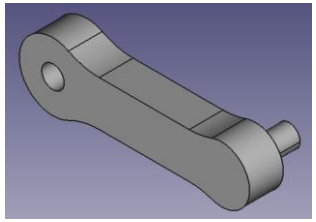
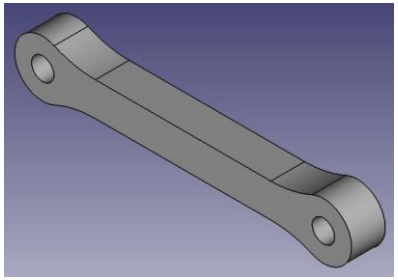
- Thêm các file GlobalFunctions.h và GlobalFunctions.cpp để khai báo và định nghĩa các hàm toàn cục:
  - *Đọc file STL và lưu dữ liệu ra mảng,*
  - *Vẽ hình từ dữ liệu STL đã lưu trong mảng*

  như hướng dẫn trong mục 3

4. Mô phỏng Robot dùng file STL

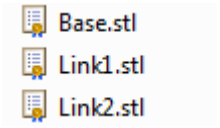
❑Tạo project và chuẩn bị dữ liệu

- Vẽ các part của robot 2 khâu phẳng, xuất ra định dạng file STL kiểu ASCII như hướng dẫn trong mục 2

(*Ví dụ: - Độ dài các khâu: L1 = 150mm, L2 = 250mm; Bề dầy khâu = 30mm*
- Hệ trục tọa độ của mỗi khâu đặt tại khớp xoay đầu tiên của khâu đó, trục khớp là trục Z…)

- Copy các file STL trên vào thư mục của project



4. Mô phỏng Robot dùng file STL

❑Vẽ robot từ các file STL

- ➢ Thêm các biến trong lớp Robot

```
    float * STLvertex[3];
    long STLcount[3];
```

// File Robot.h

- ➢ Thêm code vào hàm khởi tạo lớp Robot để đọc dữ liệu từ file STL

// File Robot.cpp

```
#include "GlobalFunctions.h"

Robot::Robot(void)
{
    l1=0.25;
    l2=0.15;
    q1=0; q2=0;
    FwdKin();
    ReadSTLFile("Base.stl",STLvertex[0],STLcount[0]);
    ReadSTLFile("Link1.stl",STLvertex[1],STLcount[1]);
    ReadSTLFile("Link2.stl",STLvertex[2],STLcount[2]);
}
```

4. Mô phỏng Robot dùng file STL

❑Vẽ robot từ các file STL

➢ Sửa code hàm vẽ Robot (*tùy theo kích thước và hệ trục tọa độ của các chi tiết trong file STL*)

// File Robot.cpp

```
void Robot::DrawRobot()
{
    GLfloat base_color[] = { 1.0, 1.0, 0.0 };
    GLfloat link1_color[] = { 0.0, 1.0, 0.0 };
    GLfloat link2_color[] = { 1.0, 0.0, 0.0 };
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0 };

    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, base_color);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialf(GL_FRONT, GL_SHININESS, 30);

    GLfloat th=30;      // Thickness of link
    glPushMatrix();
        DrawSTLData(STLvertex[0], STLcount[0]);
        glRotatef(q1*180/3.14, 0.0, 0.0, 1.0);
        glTranslatef(0.0, 0.0, th);
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, link1_color);
        DrawSTLData(STLvertex[1], STLcount[1]);
        glTranslatef(l1, 0.0, -th);
        glRotatef(q2*180/3.14, 0.0, 0.0, 1.0);
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, link2_color);
        DrawSTLData(STLvertex[2], STLcount[2]);
    glPopMatrix();
}
```

4. Mô phỏng Robot dùng file STL

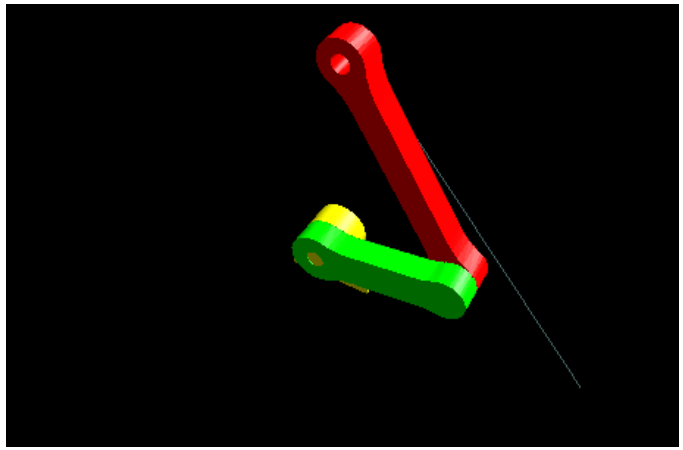
❑Vẽ robot từ các file STL

- ➢ Điều chỉnh các thông số của Robot, Trajectory, và các thông số thiết lập OpenGL cho phù hợp với kích thước của các chi tiết trong file STL

// File OpenGLControl.cpp

```
COpenGLControl::COpenGLControl(void)
{
    m_fPosX = 0.0f;     // X position of model in camera view
    m_fPosY = 0.0f;     // Y position of model in camera view
    m_fZoom = 800.0f;   // Zoom on model in camera view
    m_fRotX = 45.0f;    // Rotation on model in camera view
    m_fRotY = -30.0f;    // Rotation on model in camera view
    Rb1.Setconfig(150,250);
    Trj1.SetLine(300,-100,0,200,1);
    Trj1.SetCircle(200,150,100,1);
}

...
gluPerspective(35.0f, (float)cx / (float)cy, 10.0f, 2000.0f);
...
```

Kết quả chạy:



4. Mô phỏng Robot dùng file STL

❑Vẽ robot từ các file STL

➢ Hiệu chỉnh thêm chương trình để được kết quả như mong muốn:

- Vẽ thêm các ô lưới trên mặt phẳng X-Y

// File OpenGLControl.cpp

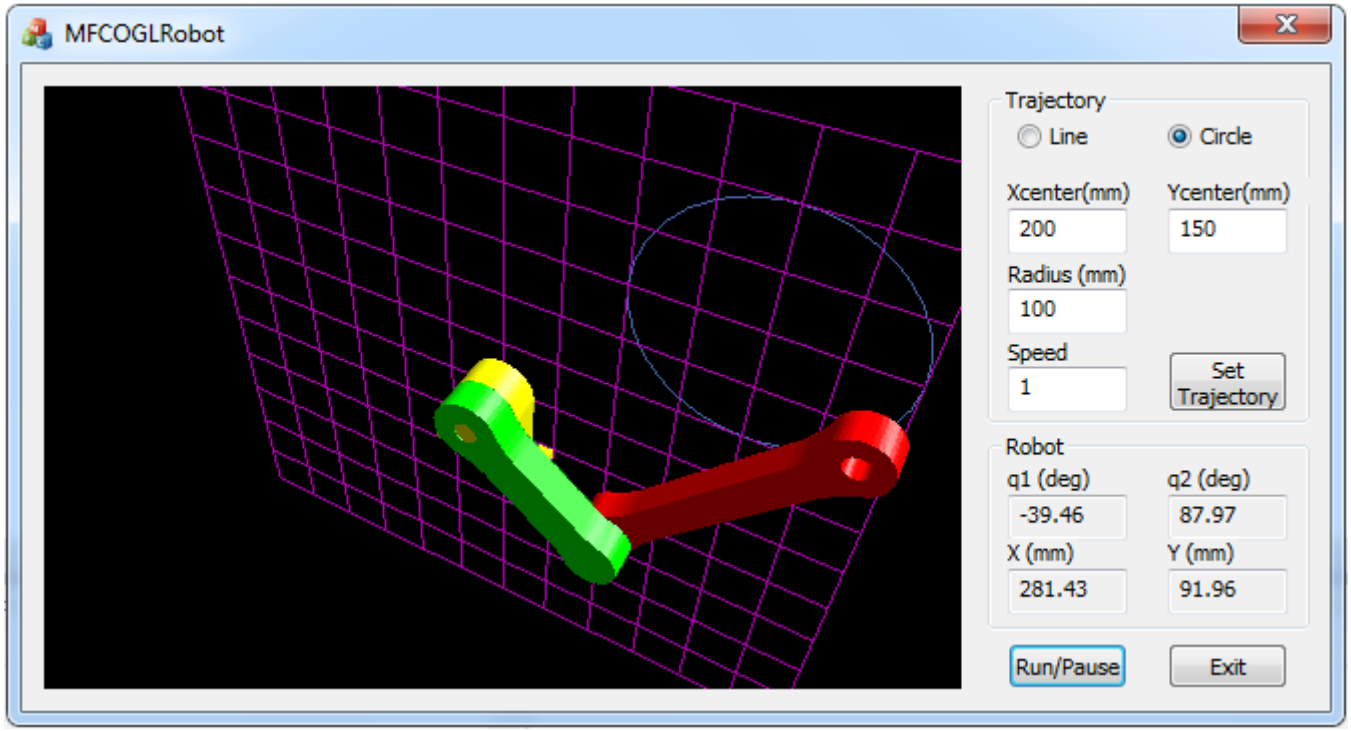
```
GLfloat grid_color[] = { 1.0, 0.0, 1.0 };
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, grid_color);
float step = 50;
int num = 6;
for (int i = -num; i <= num; i++)
{
    glBegin(GL_LINES);
    glVertex3f(i*step, -num*step, 0);
    glVertex3f(i*step, num*step, 0);
    glVertex3f(-num*step, i*step, 0);
    glVertex3f(num*step, i*step, 0);
    glEnd();
}
```

- Chỉnh sửa đơn vị của các thông số chiều dài cho hợp lý

4. Mô phỏng Robot dùng file STL

❑Vẽ robot từ các file STL

Kết quả chạy:



Hết Bài 8