```
#include<Servo.h>
#include<Wire.h>
#include<Math.h>


int  p1=0;
int  p2=0;
int  p3=0;
int  p4=0;
int  p5=0;
int  p6=0;//open

int  ppd=0;//grippper  default  world  angle


//trigonometery
//coc
double pp1c=0;
double pp2c=0;
double pp3c=0;
double pp4c=0;
double pp5c=0;
double pp6c=0;
double ppdc=0;

//sine
double pp1s=0;
double pp2s=0;
double pp3s=0;
double pp4s=0;
double pp5s=0;
double pp6s=0;
double ppds=0;



//coordinates

//gripper  coordinates
float X=0;
float Y=0;
float Z=0;

//pick  up  coordinates
float  Xi=-31;
float Yi=9;
float Zi=8;

//place  coordinates
```

```
float Xo=0;
float Yo=0;
float Zo=0;

//arm  lengths
float  x1=11.25;//base  height
float  x2=9;//humerous
float  x3=8.25;//ulna
float  x4=18.75;//wrist  &  gripper
float  cf=1.5;//correction  factor



void setup() {
  // put your setup code here, to run once:
   Wire.begin(9600);
   Serial.begin(9600);

}

void  loop()
{
  // put your main code here, to run repeatedly:

        ppds=sin(ppd*PI/180);
        ppdc=cos(ppd*PI/180);

       double d=atan(Yi/Xi);
        if(Xi<0)
      {
         p1=-(d*180/PI);
      }
       if(Xi>0)
      {
          p1=180-(d*180/PI);
      }
       if (Xi==0)
      {
         p1=90;
      }


         float  base=((sqrt((Xi*Xi)+(Yi*Yi)))-(x4*ppdc))+cf;
        float  height=Zi-(x1+(x4*ppds));
         float  hype=sqrt((base*base)+(height*height));

       if(base>0)
      {
```

```
    double Q1=(atan(height/base))*180/PI;
      double  Q2=(acos(((hype*hype)+(x2*x2)-(x3*x3))/(2*hype*x2)))*180/PI;
      double  Q3=(acos(((hype*hype)-(x2*x2)-(x3*x3))/(2*x2*x3)))*180/PI;
   p2=180-(Q1+Q2);
   p3=Q3+90;
   p4=p2+p3-ppd-180;



   pp1c=cos(p1*PI/180);
   pp1s=sin(p1*PI/180);
   pp2c=cos(p2*PI/180);
   pp2s=sin(p2*PI/180);
    pp3c=cos((p3+(p2-90))*PI/180);
    pp3s=sin((p3+(p2-90))*PI/180);
     pp4c=cos((p4-((p3+(p2-90))-90))*PI/180);
     pp4s=sin((p4-((p3+(p2-90))-90))*PI/180);

   int ppp2=180-p2;
   int ppp3=270-p2-p3;

    double ppp2c=cos(ppp2*PI/180);
    double ppp3c=cos(ppp3*PI/180);
    double ppp2s=sin(ppp2*PI/180);
    double ppp3s=sin(ppp3*PI/180);



     X=((x2*ppp2c)+(x3*ppp3c)+(x4*ppdc))*pp1c;
     Y=((x2*ppp2c)+(x3*ppp3c)+(x4*ppdc))*pp1s;
    Z=x1+(x2*ppp2s)+(x3*ppp3s)+(x4*ppds);

}

 if (base<0)
{


    double Q1=(atan(height/base))*180/PI;
      double  Q2=(acos(((hype*hype)+(x2*x2)-(x3*x3))/(2*hype*x2)))*180/PI;
      double  Q3=(acos(((hype*hype)-(x2*x2)-(x3*x3))/(2*x2*x3)))*180/PI;
   p2=Q1-Q2;
   p3=Q3+90;
   p4=p2+p3-ppd-180;



   pp1c=cos(p1*PI/180);
   pp1s=sin(p1*PI/180);
```

```
     pp2c=cos(p2*PI/180);
     pp2s=sin(p2*PI/180);
      pp3c=cos((p3+(p2-90))*PI/180);
      pp3s=sin((p3+(p2-90))*PI/180);
       pp4c=cos((p4-((p3+(p2-90))-90))*PI/180);
       pp4s=sin((p4-((p3+(p2-90))-90))*PI/180);

     int ppp2=180-p2;
     int ppp3=270-p2-p3;

      double ppp2c=cos(ppp2*PI/180);
      double ppp3c=cos(ppp3*PI/180);
      double ppp2s=sin(ppp2*PI/180);
      double ppp3s=sin(ppp3*PI/180);



       X=((-x2*pp2c)+(x3*ppp3c)+(x4*ppdc))*pp1c;
       Y=((-x2*pp2c)+(x3*ppp3c)+(x4*ppdc))*pp1s;
      Z=x1+(x2*pp2s)+(x3*ppp3s)+(x4*ppds);

  }

      Serial.print("X= ");
      Serial.print(X);
      Serial.print("  |Y= ");
      Serial.print(Y);
      Serial.print("  |Z= ");
      Serial.print(Z);
      Serial.print("  |P1= ");
      Serial.print(p1);
      Serial.print("  |P2= ");
      Serial.print(p2);
      Serial.print("  |P3= ");
      Serial.print(p3);
      Serial.print("  |P4= ");
      Serial.print(p4);
      Serial.println();


}
```