

```

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

Servo s1;
Servo s2;
Servo s3;
Servo s4;
Servo s5;
Servo s6;

int d=15;//servo step delay
int i=0;
int p1=90;
int p2=0;
int p3=180;
int p4=0;
int p5=90;
int p6=0;//open

int ppd=-10;//grippper default world angle

//positions in calculations
int P1=0;
int P2=0;
int P3=0;
int P4=0;
int P5=90;
int P6=0;

//trigonometry
//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=29;
float Yi=12;
float Zi=15;

//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

int incoming[3];

int type;

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

  s1.attach(8);
  s2.attach(2);
  s3.attach(11);
  s4.attach(13);
  s5.attach(5);
  s6.attach(9);

  s1.write(p1);
  s2.write(p1);
  s3.write(p3);
  s4.write(p4);
```

```

s5.write(p5);
s6.write(p6);

}

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

  delay(1000);

  while (Serial.available()>=4)
  {
    for(int i=0;i<4;i++)
    {
      incoming[i]=Serial.read();
    }
    Xi=incoming[0];
    Yi=incoming[1];
    Zi=incoming[2];
    type=incoming[3];

    play();
  }
}

```

```

void action()
{
  for(i=0;i<180;i++)
  {
    if (p1<P1)
    {
      p1=p1+1;
      s1.write(p1);
    }
    if (p1>P1)
    {
      p1=p1-1;
      s1.write(p1);
    }

    if (p2<P2)
    {
      p2=p2+1;

```

```

        s2.write(p2);
    }
    if (p2>P2)
    {
        p2=p2-1;
        s2.write(p2);
    }

    if (p3<P3)
    {
        p3=p3+1;
        s3.write(p3);
    }
    if (p3>P3)
    {
        p3=p3-1;
        s3.write(p3);
    }

    if (p4<P4)
    {
        p4=p4+1;
        s4.write(p4);
    }
    if (p4>P4)
    {
        p4=p4-1;
        s4.write(p4);
    }

    if (p5<P5)
    {
        p5=p5+1;
        s5.write(p5);
    }
    if (p5>P5)
    {
        p5=p5-1;
        s5.write(p5);
    }

    delay (d);
    s5.write(90);
}

}

void total_reset()
{

```

```
s1.write(90);
s2.write(0);
s3.write(180);
s4.write(0);
s5.write(90);
s6.write(0);
```

```
}
```

```
void reset()
```

```
{
```

```
    for(i=0;i<30;i++)
```

```
    {
```

```
        if(p4<P4+30)
```

```
        {
```

```
            p4=p4+1;
```

```
            s4.write(p4);
```

```
        }
```

```
        if(p4>P4+30)
```

```
        {
```

```
            p4=p4-1;
```

```
            s4.write(p4);
```

```
        }
```

```
        delay(d);
```

```
    }
```

```
    for(i=0;i<180;i++)
```

```
    {
```

```
        if(p1<90)
```

```
        {
```

```
            p1=p1+1;
```

```
            s1.write(p1);
```

```
        }
```

```
        if(p1>90)
```

```
        {
```

```
            p1=p1-1;
```

```
            s1.write(p1);
```

```
        }
```

```
        if(p2<0)
```

```
        {
```

```
            p2=p2+1;
```

```
            s2.write(p2);
```

```
        }
```

```
        if(p2>0)
```

```
        {
```

```
            p2=p2-1;
```

```

        s2.write(p2);
    }

    if(p3<180)
    {
        p3=p3+1;
        s3.write(p3);
    }
    if(p3>180)
    {
        p3=p3-1;
        s3.write(p3);
    }

    if(p4<0)
    {
        p4=p4+1;
        s4.write(p4);
    }
    if(p4>0)
    {
        p4=p4-1;
        s4.write(p4);
    }
    s5.write(90);
    delay(d);
}

}

void calculate_position()
{
    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();

```

```

}

```

```

void play()
{
    calculate_position();
    delay(500);
}

```



```

    action();
    delay(500);
    for(p6=0;p6<=180;p6++)//close
{
    s6.write(p6);
}
    delay(500);

    delay(500);

    if (type==1)
    {

        Xi=-29;
        Yi=8;
        Zi=10;
    }

    if (type==2)
    {
        Xi=29;
        Yi=8;
        Zi=10;
    }

    calculate_position();
    delay(500);
    action();
    delay(500);
    for(p6=180;p6>=0;p6--)//open
{
    s6.write(p6);
}
    delay(500);

    delay(500);
    reset();
    delay(500);

    Xi=incoming[0];
    Yi=incoming[1];
    Zi=incoming[2];
    type=incoming[3];
}

```