MINI PROJECT REPORT
On
2D ROBOTIC ARM CONTROL WITH ARDUINO

BY
Praveen S(15EC07)
Shivprasad meena(15EC12)

MINOR PROJECT REPORT SUBMITTED TO DEPARTMENT OF ELECTROINCS AND COMMUNICATION ENGINEERING INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD
APRIL 2018

2D ROBOTIC ARM

A PROJECT REPORT

Submitted in partial fulfillment of the
requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by:
Praveen S(15EC07)

Shiv Prasad meena (15EC12)

Guided by:
Dr.Jagdish dn
Asst.prof IIIT DWD

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD
APRIL 2018

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **"2D ROBOTIC ARM"** submittedin partial fulfillment for the award of the degree of Bachelor of Technology in '**Electronics and Communication**' completed under the supervision of **Dr.Jagdish,** IIIT Dharwad is an authentic work.

Further, we declare that we have not submitted this work for the award of any other degree elsewhere.

Signature and name of the student(s) with date

CERTIFICATE by MINI PROJECT Guide

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of Mini Project Guides with dates and their designation

Preface

This report on "**2D ROBOTIC ARM**" is prepared under the guidance of **Dr.Jagdish**

Through this report we have controlling with arduino uno IDE and 2 servo motors.

Praveen S
Shiv Prasad meena
B.Tech. IIIrd Year
Discipline of **Electronics and Communication Engineering**
IIIT Dharwad

Acknowledgements

We would like to thank **Dr.Jagdish** for his kind support and valuable guidance.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Praveen S
Shivprasad meena
B.Tech. IIIrd Year
Discipline of **Electronics and communication engineering**
IIIT Dharwad

Abstract

Today, technology is developing in the same direction in line with rapidly increasing human needs. The work done to meet these needs makes life easier every day, and these studies are concentrated in robotic arm studies. Robot arms work with an outside user or by performing predetermined commands. Nowadays, the most developed field of robot arms in every field is the industry and medicine sector.

Designed and realized in the project, the robot arm has the ability to move in 4 axis directions with 5 servo motors. Thanks to the holder, you can take the desired material from one place and carry it to another place, and also mix it with the material it receives. While doing this, robot control is provided by connecting to the android application via Bluetooth module connected to Arduino Nano microcontroller.

Table of Contents

Candiate Declaration.....1

Supervisior Certificate.....2

PREFACE.....3

ACKNOWLEDGEMENT.....4

ABSTRACT.....5

1.Introduction.....6

2. THEORETICAL INFRASTRUCTURE

3 Servo Motors

4. Arduino Uno Microcontroller

Circuit Diagram (Above Board) 6

5 working principle

6 Basic model of 2d Robotic arm

7 CODE

8 COMPONENTS

Introduction

All about robotic: Robotic is the branch of technology that deal with design, construction,operations,and applications of robots. Robot is electro mechanical machine which is guided by electronic circuitry or computer programme to perform various task.

A robotic arm is a robotic manipulator, usually programmable, With functions similar to the human arm.

In the project, Arduino Nano microcontroller written in C language is programmed and servo motor control is provided. Thus, it is possible to perform the desired operations by means of the elements located on the Arduino without any circuit construction other than the circuit where the servo motor inputs are located. For the mechanical part, the robot arm is drawn with the SolidWorks program and the dimensions of the robot arm are specified. A 5V power supply is also preferred for the robot to work

2.THEORETICAL INFRASTRUCTURE

The theoretical background of the project is examined below as main headings and subheadings.

3 Servo;

Detects the operation error of a mechanism, provides feedback and corrects faults. The servo motor can have alternating current (AC), direct current (DC) or stepper motors. In addition to these, there are drive and control circuits. Servo motors are the kinds of motors that can fulfill the commands we want. They can operate steadily even at very small or very large speeds. In these motors, the large moment can be obtained from the small size.

Servo motors are used in control systems such as fast operation, excessive axis movement, condition control and so on. Servo motors are the last control element of a mechanism. They are highly sensitive and servo motors are used in conjunction with electronic or programmable circuits. These engines are divided into AC and DC. When the AC servo motors are brushless type motors, the servo motors brush. Servo motors are mostly three cables. These are a red cable for power, black for grounding and yellow cables for control (data, data). One of the servomotors used in the production phase of the project is shown in Fig.1.

Fig .1

In the project Tower, Pro SG90 Mini servo motor is used. Some features of this servo motor; versatile operation, 10 µs pulse width control, VP-P: 3-5 V
Square wave and working voltage of 4.8-6V. The used servomotor has a working voltage of 0.12 s / 60 ° and a torque of 1.2-1.6 kg/cm at low operating voltage.
Servo motors are controlled according to the signal condition. In doing so, the supplied pulse width modulated (PWM) signal is used with the data bus. Each servo motor is controlled by a PWM signal at 10-20 ms and at 0.5-1.5 ms.the position of the motor shaft is determined according to the duration (tk) of this signal at logic 1. These;

- When tk = 0.5 ms, the motor shaft rotates to the end,
- When tk= 0,5- 1 ms, the position of the motor shaft is in the middle,
- When tk= 1 - 1.5 ms, the motor shaft turns to the right,
- When tk= 10-20 ms (when the same signal is given again) it remains in its old position,

The position control of these motors is determined using the required pulses. The servo motors DC used in the project are kept at about 5V during operation.

4.Arduino

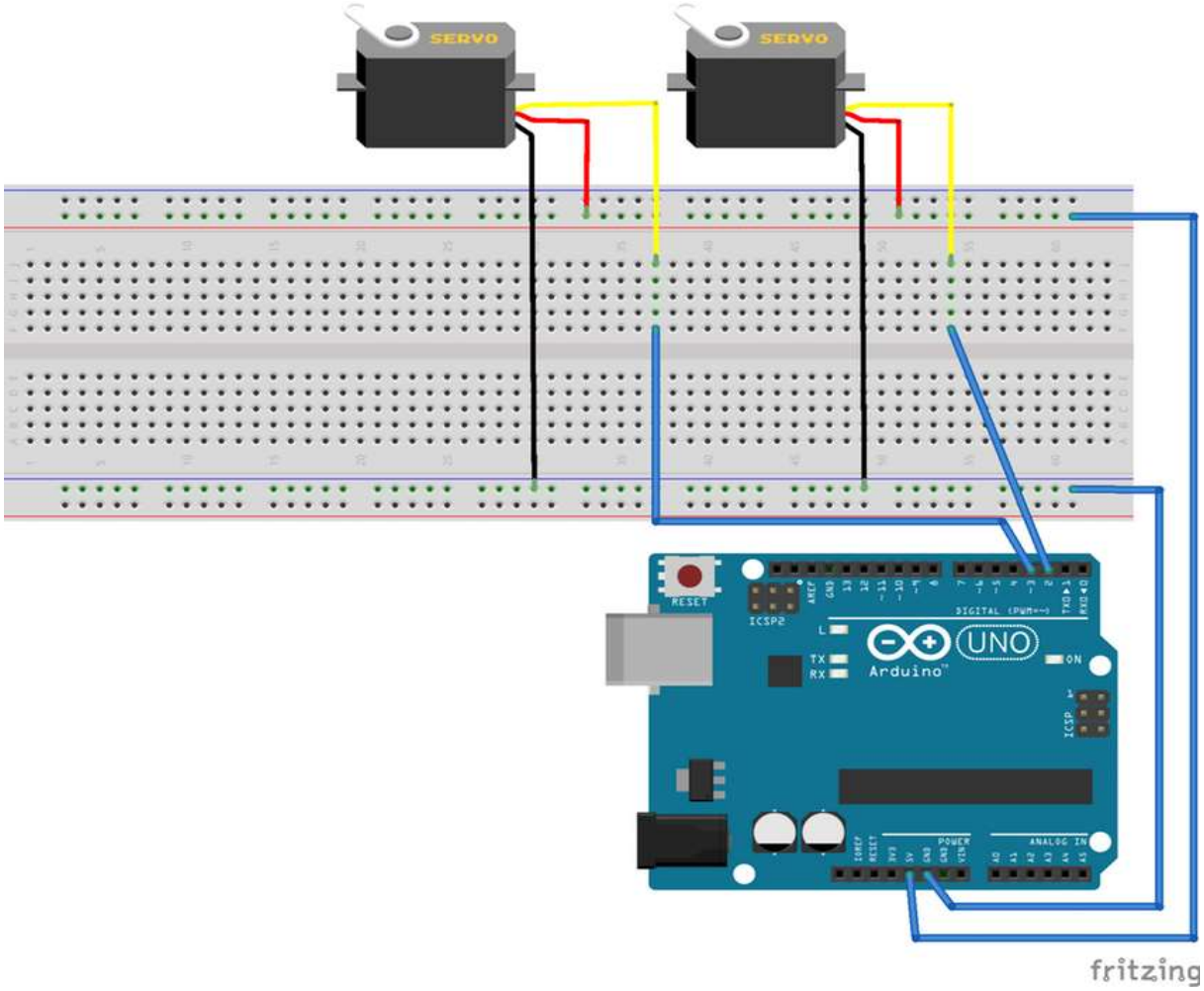
The high-performance Microchip 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byteoriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter.

Arduino uno

Circuit Diagram (Above Board)

Circuit; Servo inputs, Arduino pin inputs, Servo motors are activated in this way, the Arduino uno connections and power supply connections are shown. Thanks to this

circuit we use, it is possible to distribute the 5V from the power supply to the servo motors.



5.Power:

The power supply selected for feeding the control circuit of the servomotors is capable of delivering the same current even if all the synchronous servomotors are operating. When all servo motors are operated at the same time, they draw 0.5A current. In addition, 5 V was needed for the Arduino used for robot movement in the project. This requirement is provided by a 5V power supply.

6.**Working principle**:

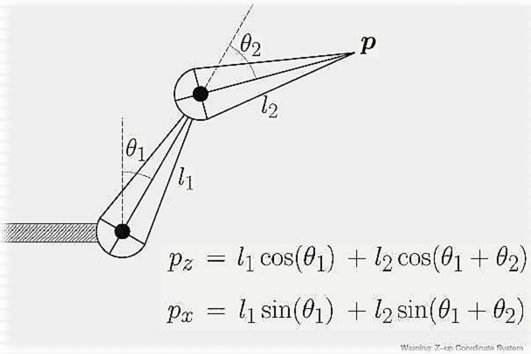
- Imagine a point located at (x,y). If you wanted to rotate that point around the origin, the coordinates of the new point would be located at (x',y').
- $x' = x\cos\theta - y\sin\theta$
- $y' = y\cos\theta + x\sin\theta$
- Where θ is the angle of rotation

FORWARD KINEMATICS

- To find out the position and orientation of the end-effector of the manipulater, forward kinematics is used.

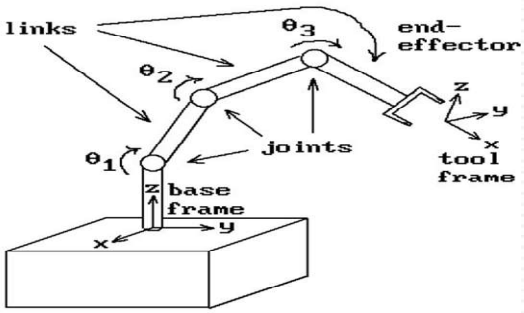
FORWARD KINEMATICS

- To find out the position and orientation of the end-effector of the manipulater, forward kinematics is used.

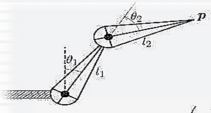


TOOL FRAME &BASE FRAME

- We generally describe the position of the manipulator by giving a description of the tool frame, which is attached to end effector, relative to the base frame, which is attached to the nonmoving base of the manipulator.



INVERSE KINEMATICS



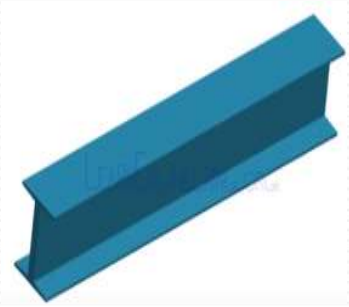
$$\theta_2 = \cos^{-1}\left(\frac{p_z^2 + p_x^2 - l_1^2 - l_2^2}{2l_1 l_2}\right)$$

$$\theta_1 = \frac{-p_z l_2 \sin(\theta_2) + p_x(l_1 + l_2\cos(\theta_2))}{p_x l_2 \sin(\theta_2) + p_z(l_1 + l_2\cos(\theta_2))}$$

- For a given position and orientation of the tool frame, values for the joint variables can be calculated via the inverse kinematics.

For an endpoint position and orientation given in Cartesian coordinates, the process of finding the values that the joint variables must take is called inverse kinematics. Reverse kinematics can be difficult to solve according to the advanced kinematics. In many cases, it is necessary to use techniques that do not guarantee a solution and include trial and error

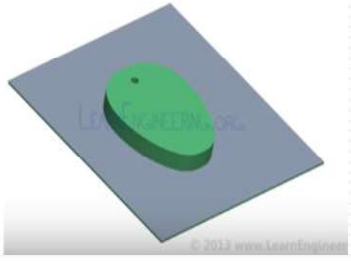
DEGREE OF FREEDOM



- 6 D.O.F
- 3 TRANSLATIONAL X Y Z
- 3 ROTATIONAL θ_X θ_Y θ_Z

DEGREE OF FREEDOM

PLANER MECHANISM

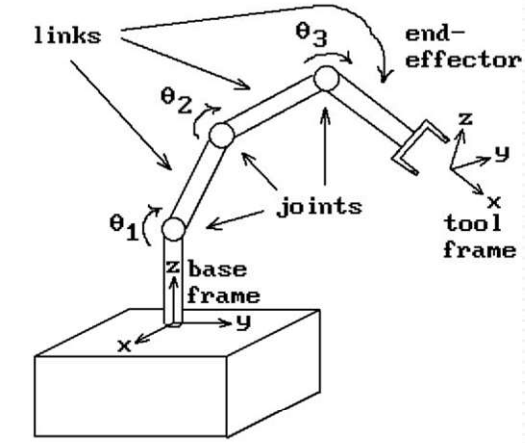


3 D.O.F

2 TRANSLATIONAL X Y

ROTATIONAL θ_Z

BASIC MODEL OF ROBOTIC ARM



ROBOT ARM CONTROL

The connection box is made to distribute the 5V voltage from the supply source to the servo motors. In doing so, servo motor inputs, Arduino pin inputs, and communication circuit elements are used. The mechanical part of the robot arm is designed by combining the pre-selected parts appropriately.

7 USED COMPONENTS :

1. ARDUINO UNO
2. 2 SERVO MOTORS
3. 2 SCALES
4. MALE FEMALE CONNECTORS
5. USB CABLE

CONCLUSION:

Robotic arms, many areas are developable. Thanks to the robotic arms, many tasks are made easier and the resulting error level has been reduced to a minimum. For example; some pharmacy-based drug-giving robots and a projected robot arm have been developed. In addition to this, the ability to move the robot arm is further increased. Despite the fact that the robotic arm made by this project is of prototype quality it has a quality that can be improved for more robotic systems.

the purpose of the project is to provide control of 2 axes moving robot arm design and this robot arm with a suitable microcontroller.

During the process of making and developing the project, a lot of theoretical knowledge has been transferred to the practice and it has been ensured that it is suitable for the purpose of the project

REFERENCES

1.Introductions to robotics

2. WMHW Kadir, RE Samin, BSK Ibrahim. Internet controlled a robotic arm. Procedia Engineering. 2012.

3. Electric Electronic Technology-Step and Servo Motors, SVET, 2007

Programming code:

```
#include <Servo.h>


Servo BasePan, GripTilt, ShoulderTilt, ElbowTilt, WristTilt; // create servo object to control a servo.
//Servo counting goes from 0 to 4
//Grip = 0  Wrist = 1  Elbow = 2   Shoulder = 3  Base = 4

int pos, CurrentAngle;
int i;


//****** Setup IK variables ******
#define PI 3.14159265

float ShoulderLength = 9.0;
float ElbowLength = 13.0;
```

```
float WristLength = 4.0;
float Hypot, Slope, CirclePointX, CirclePointY;
float x1 = 5.0;//Target end point.10.6
float y1 = 5.0;//Target end point.7.5
float x2 = 0.0;//origin start point.
float y2 = 0.0;//origin start point.
float A, B, C,
    PosXAnswer,   //positive X portion of Quadratic Equation.
    PosYAnswer,   //positive Y portion of Qudratic Equation.
    NegXAnswer,   //negative X portion of Quadratic Equation.
    NegYAnswer,   //negative Y portion of Quadratic Equation.
    Angle_A,      //Angle located at origin (0,0).
    Angle_A_Temp, //Second 1/2 of Angle located at origin (0,0).Shoulder servo setting.
    Angle_B,      //Elbow servo setting.
    Angle_C;      //Wrist servo setting subtracted from 180.

//*********************************


void setup()
{

  Serial.begin(9600);



  pinMode( 9, OUTPUT); //Base pin.
  pinMode( 10, OUTPUT); //Shoulder pin.
  pinMode( 11, OUTPUT); //Elbow pin.
  pinMode( 8, OUTPUT); //Wrist pin.
  pinMode( 7, OUTPUT); //Grip
```

```
  BasePan.attach(9);  // attaches the servo on pin 9 to the servo object
  GripTilt.attach(7);  // attaches the servo on pin 7 to the servo object
  ShoulderTilt.attach(10);  // attaches the servo on pin 10 to the servo object
  ElbowTilt.attach(11);  // attaches the servo on pin 11 to the servo object. Can not use
Pin 11.
  WristTilt.attach(8);// attaches the servo on pin 8 to the servo object


  // Initialize servos...
  Serial.println("Initializing Servos...");
  ResetServos();

}//end setup.


void loop()
{


  for ( x1 = 3; x1 < 14; x1 = x1 + 1) {

    for (y1 = 1; y1 < 15; y1 = y1 + 1) {


      Serial.println("********************");
      Serial.println();
      Serial.print("Point: (");
      Serial.print(x1);
      Serial.print(",");
```

```
      Serial.print(y1);
      Serial.println(")");
      if ((x1 < ShoulderLength) && (y1 < ShoulderLength)) {
        Special_Calc_Point();

      }//end if x1 & y1 < ShoulderLength.
      else {

        Calc_Point();

      }//end else.
    }//end y1 for.
  }//end x1 for.


}//end Loop.



void Special_Calc_Point() {
  //*****************************************************************
  //********** Starting Special Circle Calcs ***********************
  //*****************************************************************

  Hypot = sqrt(sq(x1) + sq(y1)) + WristLength;
  A = ElbowLength;//4;
  B = Hypot;//B_Value;//6.7; //check here for nan.
  C = ShoulderLength;//6;
```

```
  //*** Calculate angle A ***//
  Angle_A = acos((sq(B) + sq(C) - sq(A)) / (2 * B * C)) * (180 / PI);
  //    Angle_A_Temp = acos((sq(x1) + sq(Hypot) - sq(y1)) / (2 * x1 * Hypot))*(180 / PI);
//Second 1/2 of Angle A from X axis upto Shoulder arm.
  //    Angle_A = Angle_A + Angle_A_Temp; //Second 1/2 of Angle A from X axis plus
Shoulder arm angle.



  //*** Calculate angle B ***//
  Angle_B =  acos((sq(C) + sq(A) - sq(B)) / (2 * A * C)) * (180 / PI);

  //*** Calculate angle C ***//
  Angle_C = acos((sq(A) + sq(B) - sq(C)) / (2 * A * B)) * (180 / PI);
  //  Angle_C = 180 - Angle_C; //must calc larger opposite side angle.

  Angle_A = Angle_A * (1 + (Angle_A / (Angle_B + Angle_C))); // Angle A is
recalculated due to the servo gearbox used.

  if (Angle_A < 35) {// Angle A is recalculated due to the servo gearbox used.
    Angle_A = 35;
  }//End if <35.



  if (!isnan(Angle_A)) {      //If Angle is a valid number.
    Serial.println(Angle_A);
    ShoulderTilt.write(Angle_A);
    delay(1000);
  }


  if (!isnan(Angle_B)) { //If Angle is a valid number.
```

```
  CurrentAngle = ElbowTilt.read();

  if (Angle_B > CurrentAngle) {
    Serial.print(CurrentAngle);
    Serial.print(" < ");
    Serial.println(Angle_B);
    for (i = CurrentAngle; i < Angle_B; i = i + 1) //if Angle_B > CurrentAngle.
    {

      ElbowTilt.write(i);
      delay(5);
    }//end for CurrentAngle.

  }//end if Angle_B > CurrentAngle.

  else
  {
    Serial.print(CurrentAngle);
    Serial.print(" > ");
    Serial.println(Angle_B);
    for (i = CurrentAngle; i > Angle_B; i = i - 1) //if Angle_B < CurrentAngle.
    {

      ElbowTilt.write(i);
      delay(5);
    }//end for CurrentAngle.
  }//end else.

} //end if !isnan.
```

```
if (!isnan(Angle_C)) { //If Angle is a valid number.

  CurrentAngle = WristTilt.read();

  if (Angle_C > CurrentAngle) {
    Serial.print(CurrentAngle);
    Serial.print(" < ");
    Serial.println(Angle_C);
    for (i = CurrentAngle; i < Angle_C; i = i + 1) //if Angle_C > CurrentAngle.
    {

      WristTilt.write(i);
      delay(5);
    }//end for CurrentAngle.

  }//end if Angle_C > CurrentAngle.

  else
  {
    Serial.print(CurrentAngle);
    Serial.print(" > ");
    Serial.println(Angle_C);
    for (i = CurrentAngle; i > Angle_C; i = i - 1) //if Angle_C < CurrentAngle.
    {

      WristTilt.write(i);
      delay(5);
    }//end for CurrentAngle.
  }//end else.
```

```
  } //end if isnan.


}//end Special_Calc_Circle.


void Calc_Point() {
  //*********** Line Calcs **********************************
  Slope = (y2 - y1) / (x2 - x1);
  Serial.print("Slope: ");
  Serial.println(Slope, 4);


  //*********** Target Circle Calcs **********************************
  A = 1 + sq(y1 / x1);
  B = (-2 * x1) + (-2 * y1 * Slope);
  C = sq(y1) + sq(x1) - sq(WristLength);



  //*************************************************************
  //************ Quadratic Equation ******************************
  //*************************************************************

  /*
  Use: sqrt (-1);  // Set to not a number...nan.
  */

  Serial.println();
```

```
  //************************//
  //*** Negative Portion ***//
  //************************//

  NegXAnswer = ((-1 * B) - (sqrt(sq(B) - (4 * A * C)))) / (2 * A); //Minus portion of
Quadratic Equation.
  NegYAnswer = Slope * NegXAnswer;

  if (!isnan(NegXAnswer)) {

    Serial.print("Negative Point: ");
    Serial.print(NegXAnswer);
    Serial.print(",");
    Serial.println(NegYAnswer);
    Serial.println();

    Calc_Circle(NegXAnswer, NegYAnswer);
  }//end if Negative !isnan.

  else
  {


    //************************
    //*** Positive Portion ***
    //************************

    PosXAnswer = ((-1 * B) + (sqrt(sq(B) - (4 * A * C)))) / (2 * A); //Plus portion of
Quadratic Equation.
    PosYAnswer = Slope * PosXAnswer;

    if (!isnan(PosXAnswer)) {
```

```
    Serial.print("**** Positive Point: ");
    Serial.print(PosXAnswer);
    Serial.print(",");
    Serial.println(PosYAnswer);
    Serial.println();

    Calc_Circle(PosXAnswer, PosYAnswer);

   }//end if Positive !isnan.

  }//end else.

}//end Calc_Point.



void Calc_Circle(float BXValue, float BYValue) {
  //*************************************************************
  //********** Starting Circle Calcs ********************************
  //*************************************************************

  Hypot = sqrt(sq(BXValue) + sq(BYValue));
  A = ElbowLength;//4;
  B = Hypot;//B_Value;//6.7; //check here for nan.
  C = ShoulderLength;//6;



  // Angle_A_Temp = acos((sq(x1) + sq(Hypot) - sq(y1)) / (2 * x1 * Hypot)) * (180 / PI);
//Second 1/2 of Angle A from X axis upto Shoulder arm.
  //  Angle_A = Angle_A + Angle_A_Temp; //Second 1/2 of Angle A from X axis plus
Shoulder arm angle.
```

```
  //*** Calculate angle B ***//
  Angle_B =  acos((sq(C) + sq(A) - sq(B)) / (2 * A * C)) * (180 / PI);


  //*** Calculate angle C ***//
  Angle_C = acos((sq(A) + sq(B) - sq(C)) / (2 * A * B)) * (180 / PI);
  Angle_C = 180 - Angle_C; //must calc larger opposite side angle.


  //*** Calculate angle A ***//
  Angle_A = acos((sq(B) + sq(C) - sq(A)) / (2 * B * C)) * (180 / PI);

  Angle_A = Angle_A * (1 + (Angle_A / (Angle_B + Angle_C))); // Angle A is
recalculated due to the servo gearbox used.

  if (Angle_A < 35) {
    Angle_A = 35;
  }//End if <35.


  if (!isnan(Angle_A)) { //If Angle is a valid number.
    Serial.println(Angle_A);
    ShoulderTilt.write(Angle_A);
    delay(1000);
  }

  if (!isnan(Angle_B)) { //If Angle is a valid number.

    CurrentAngle = ElbowTilt.read();

    if (Angle_B > CurrentAngle) {
      Serial.print(CurrentAngle);
```

```
   Serial.print(" < ");
   Serial.println(Angle_B);
   for (i = CurrentAngle; i < Angle_B; i = i + 1) //if Angle_B > CurrentAngle.
   {

     ElbowTilt.write(i);
     delay(5);
   }//end for CurrentAngle.

  }//end if Angle_B > CurrentAngle.

  else
  {
   Serial.print(CurrentAngle);
   Serial.print(" > ");
   Serial.println(Angle_B);
   for (i = CurrentAngle; i > Angle_B; i = i - 1) //if Angle_B < CurrentAngle.
   {

     ElbowTilt.write(i);
     delay(5);
   }//end for CurrentAngle.
  }//end else.

} //end if !isnan.



if (!isnan(Angle_C)) { //If Angle is a valid number.

  CurrentAngle = WristTilt.read();
```

```
  if (Angle_C > CurrentAngle) {
    Serial.print(CurrentAngle);
    Serial.print(" < ");
    Serial.println(Angle_C);
    for (i = CurrentAngle; i < Angle_C; i = i + 1) //if Angle_C > CurrentAngle.
    {

      WristTilt.write(i);
      delay(5);
    }//end for CurrentAngle.

  }//end if Angle_C > CurrentAngle.

  else
  {
    Serial.print(CurrentAngle);
    Serial.print(" > ");
    Serial.println(Angle_C);
    for (i = CurrentAngle; i > Angle_C; i = i - 1) //if Angle_C < CurrentAngle.
    {

      WristTilt.write(i);
      delay(5);
    }//end for CurrentAngle.
  }//end else.

 } //end if !isnan.


}//end Calc_Circle.
```

```
//*************************************************************
//                ResetServos
//*************************************************************
void ResetServos()
{

  ShoulderTilt.write(65);//was 50.60.35.55
  delay(2000);


  ElbowTilt.write(90 );//was 60.70.50.90.40.90
  delay(500);


  WristTilt.write(90);//was 60.50.40
  delay(500);


  GripTilt.write(90); //was 40,35,90,20,60.10.45
  delay(500);


} //End ResetServos.
```