#include <Servo.h>

#define SERVO\_BASE 2

#define SERVO\_SHOULDER 3

#define SERVO\_ELBOW 4

#define SERVO\_GRIPPER 5

Servo myservo\_1; // create servo object to control a servo

Servo myservo\_2;

Servo myservo\_3;

Servo myservo\_4;

unsigned char Data\_String[25],Data\_Index = 0,New\_Data\_Rec\_Flag = 0;

unsigned int Received\_Servo\_Value[4],Final\_Servo\_Val[4];

unsigned char Index\_i = 0,Index\_j = 0,Counter\_to\_Refresh = 0;

void setup() {

Serial.begin(9600);

myservo\_1.attach(SERVO\_BASE); // attaches the servo on pin 2 to the servo object

myservo\_2.attach(SERVO\_SHOULDER); // attaches the servo on pin 3 to the servo object

myservo\_3.attach(SERVO\_ELBOW); // attaches the servo on pin 4 to the servo object

myservo\_4.attach(SERVO\_GRIPPER); // attaches the servo on pin 5 to the servo object

myservo\_1.write(90);

delay(200);

myservo\_2.write(90);

delay(200);

myservo\_3.write(90);

delay(200);

myservo\_4.write(90);

delay(200);

Received\_Servo\_Value[0] = 90; // Default values

Received\_Servo\_Value[1] = 90;

Received\_Servo\_Value[2] = 90;

Received\_Servo\_Value[3] = 45;

Final\_Servo\_Val[0] = 90; // Default values

Final\_Servo\_Val[1] = 90;

Final\_Servo\_Val[2] = 90;

Final\_Servo\_Val[3] = 45;

}

void loop()

{

if(New\_Data\_Rec\_Flag==0)

{

if (Serial.available()) // check whether bluetooth data is available

{

// read incoming serial data:

char inChar = Serial.read(); // read bluetooth data one by one

//Serial.print(inChar);

if(inChar==0x0A) // End character of bluetooth data

{

Data\_String[Data\_Index] = inChar;

Data\_Index = 0;

New\_Data\_Rec\_Flag = 1;

}

else

{

if(inChar!=0x2C) // removing ascii value, except for 0x2C which is ascii for ,

{

inChar = inChar - 0x30; // removing Ascii value of 0, so that we can get exat value

}

Data\_String[Data\_Index] = inChar;

Data\_Index++;

}

}

}

if(New\_Data\_Rec\_Flag==1)

{

Received\_Servo\_Value[0] = 0;

Received\_Servo\_Value[1] = 0;

Received\_Servo\_Value[2] = 0;

Received\_Servo\_Value[3] = 0;

for(Index\_i = 0,Index\_j = 0;;)

{

if(Data\_String[Index\_j]==0x2C)

{

Index\_j++;

Index\_i++;

Serial.print("A ");

}

else if(Data\_String[Index\_j]==0x0A)

{

New\_Data\_Rec\_Flag = 0;

Serial.print("B ");

break;

}

else

{

Received\_Servo\_Value[Index\_i] = Received\_Servo\_Value[Index\_i] \* 10 + Data\_String[Index\_j];

Index\_j++;

Serial.print("C ");

}

}

Serial.print(Received\_Servo\_Value[0]);

Serial.print(" ");

Serial.print(Received\_Servo\_Value[1]);

Serial.print(" ");

Serial.print(Received\_Servo\_Value[2]);

Serial.print(" ");

Serial.println(2\*Received\_Servo\_Value[3]);

}

Counter\_to\_Refresh++;

delay(1);

if(Counter\_to\_Refresh >= 10) // delay of 10 msec = 1 msec \* 10, this will allow smooth movement of servos

{

Counter\_to\_Refresh = 0;

if(Received\_Servo\_Value[0]!=Final\_Servo\_Val[0])

{

if(Received\_Servo\_Value[0]>Final\_Servo\_Val[0])

{

Final\_Servo\_Val[0]++;

}

if(Received\_Servo\_Value[0]<Final\_Servo\_Val[0])

{

Final\_Servo\_Val[0]--;

}

myservo\_1.write(180 - Final\_Servo\_Val[0]); // adjuted as per app

}

if(Received\_Servo\_Value[1]!=Final\_Servo\_Val[1])

{

if(Received\_Servo\_Value[1]>Final\_Servo\_Val[1])

{

Final\_Servo\_Val[1]++;

}

if(Received\_Servo\_Value[1]<Final\_Servo\_Val[1])

{

Final\_Servo\_Val[1]--;

}

myservo\_2.write(Final\_Servo\_Val[1]);

}

if(Received\_Servo\_Value[2]!=Final\_Servo\_Val[2])

{

if(Received\_Servo\_Value[2]>Final\_Servo\_Val[2])

{

Final\_Servo\_Val[2]++;

}

if(Received\_Servo\_Value[2]<Final\_Servo\_Val[2])

{

Final\_Servo\_Val[2]--;

}

myservo\_3.write(Final\_Servo\_Val[2]);

}

if(Received\_Servo\_Value[3]!=Final\_Servo\_Val[3])

{

if(Received\_Servo\_Value[3]>Final\_Servo\_Val[3])

{

Final\_Servo\_Val[3]++;

}

if(Received\_Servo\_Value[3]<Final\_Servo\_Val[3])

{

Final\_Servo\_Val[3]--;

}

myservo\_4.write(180 - (2\*Final\_Servo\_Val[3])); // adjuted as per app

}

}

}