



BRAZO ROBOT ARDUINO

I.E.S. PADRE MANJÓN



DPTO. TECNOLOGÍA INFORMÁTICA

2017/2018

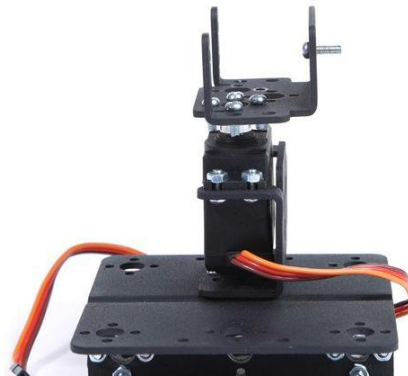
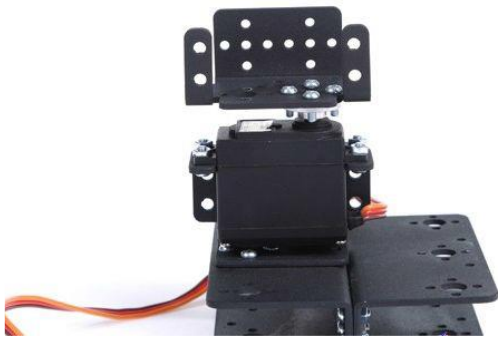
José Miguel Capilla Ruiz

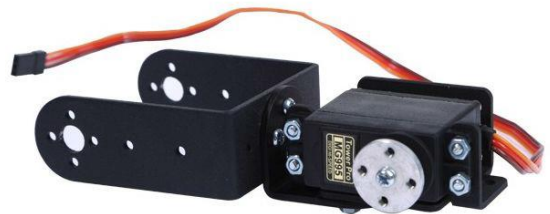
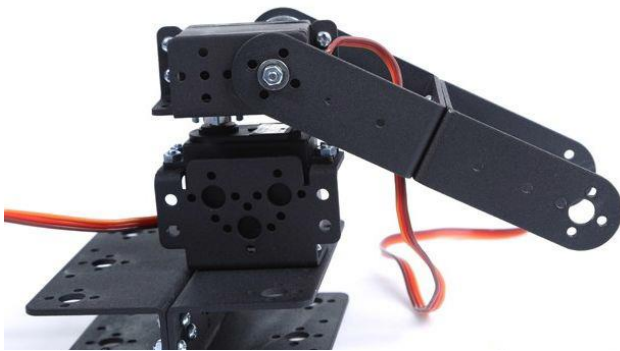
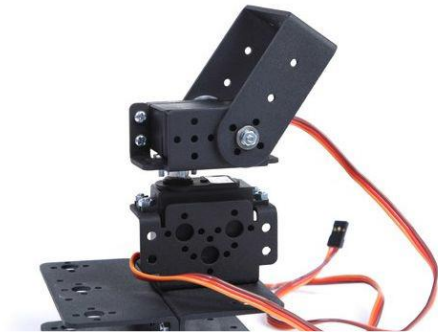
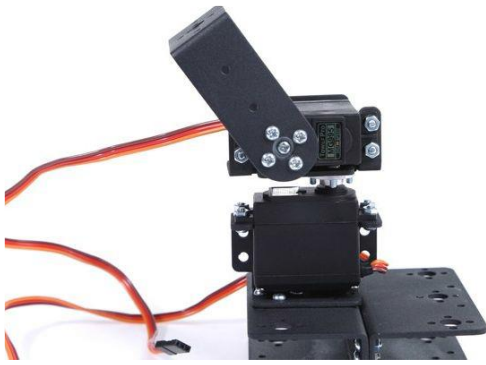
PROYECTO ROBOT

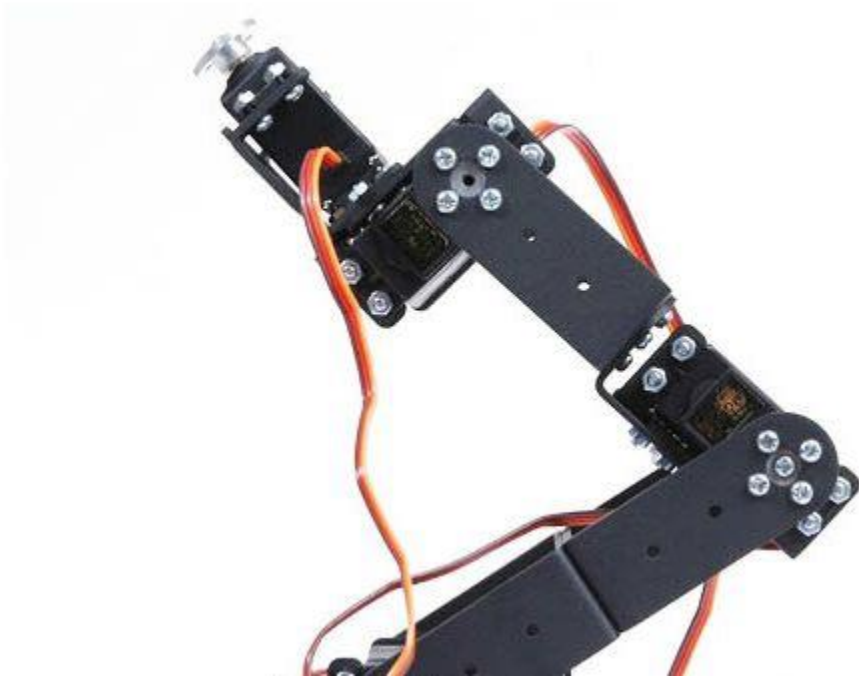
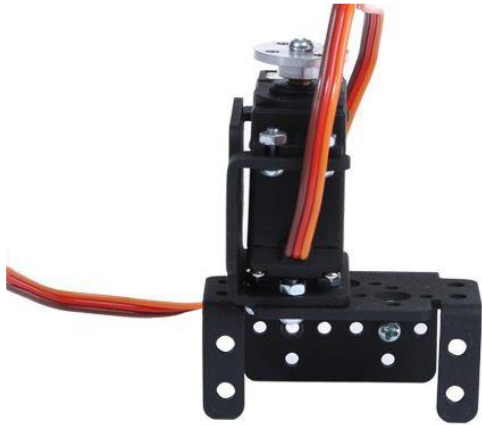
Contenido

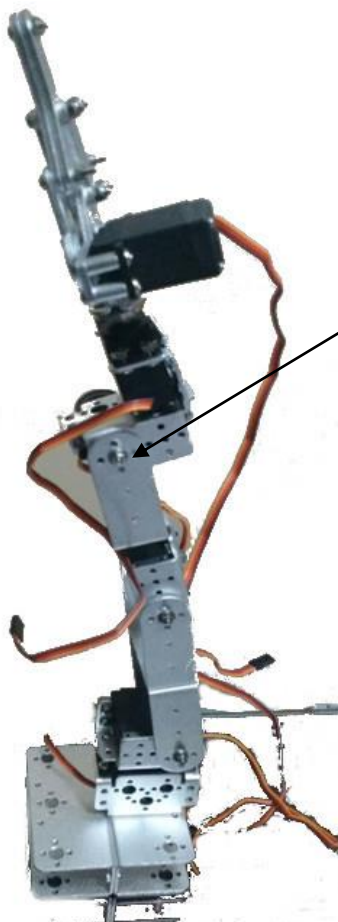
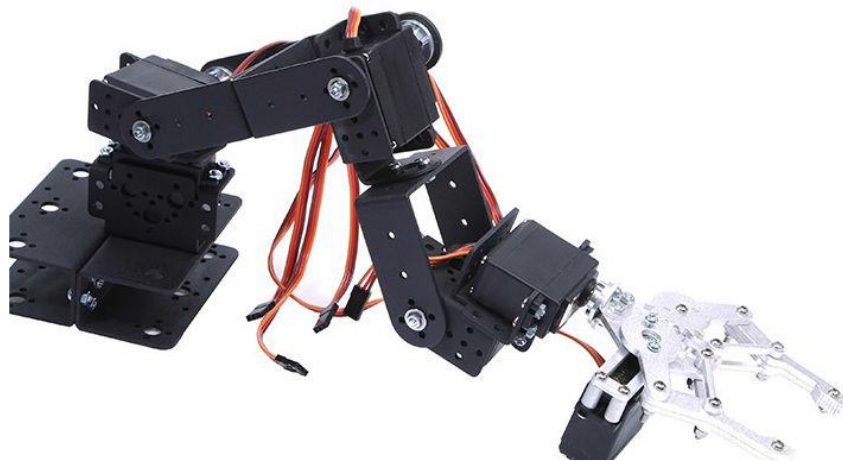
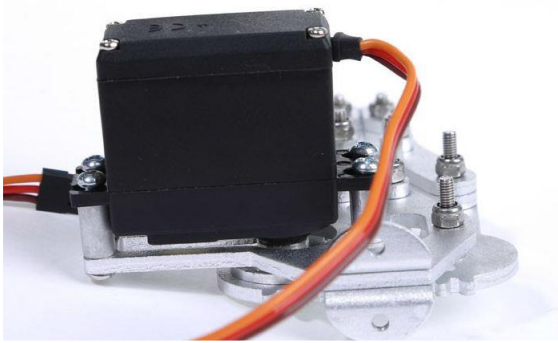
MATERIAL Y MONTAJE	2
MOTOR MG996R	6
CALIBRACIÓN DE SERVOS:.....	6
ALIMENTACIÓN DE 5 SERVOS:	7
MONTAJE PLACA.	7
PROGRAMACIÓN	8
INICIAR MOTORES:.....	8
//CONTROL DE TECLADO.....	9
POSICIÓN (X,Y,Z) Y CÁLCULO ÁNGULOS.....	11
POSICIONAMIENTO x,y,z. Programa	12
RESOLUCIÓN DE MOVIMIENTO BRUSCO	14
Posicionamiento siguiendo los puntos almacenados en un array.....	17
Depurando el programa anterior para mover los motores a la vez grado a grado.	20
CONTROL BLUETOOTH	22
Módulo HC-06	22
INTERFACE.....	22
CÓDIGO APPINVENTOR:.....	23
ÁREA DE TRABAJO DEL BRAZO ROBOT EN ÁREA POSITIVA.....	24

MATERIAL Y MONTAJE.









Motor eliminado por razones de peso.



MOTOR MG996R

Especificaciones:

- Torque: 9.4kg/cm (4.8v); 11kg/cm (6.0v)
- Velocidad: 0.14 s/60°
- Rango de rotación: 180°
- Peso: 55 g
- Dimensiones: 40.6 x 19.8 x 42.9 mm
- Tipo de engranaje: Metal
- Tipo de conector: Hembra
- Modulación: Analógica
- Periodo: 20 ms
- Voltaje: 4.8 ~ 6 v ~ 7.2v
- Ancho pulso: 1.0 ~ 2.5 ms
- Temperatura de operación: 0 ~ 55 °C



CALIBRACIÓN DE SERVOS:

Después del montaje, aunque los servos se ajustaron a 90°, los ángulos de cada uno en posición vertical y centrada, han quedado según la tabla:

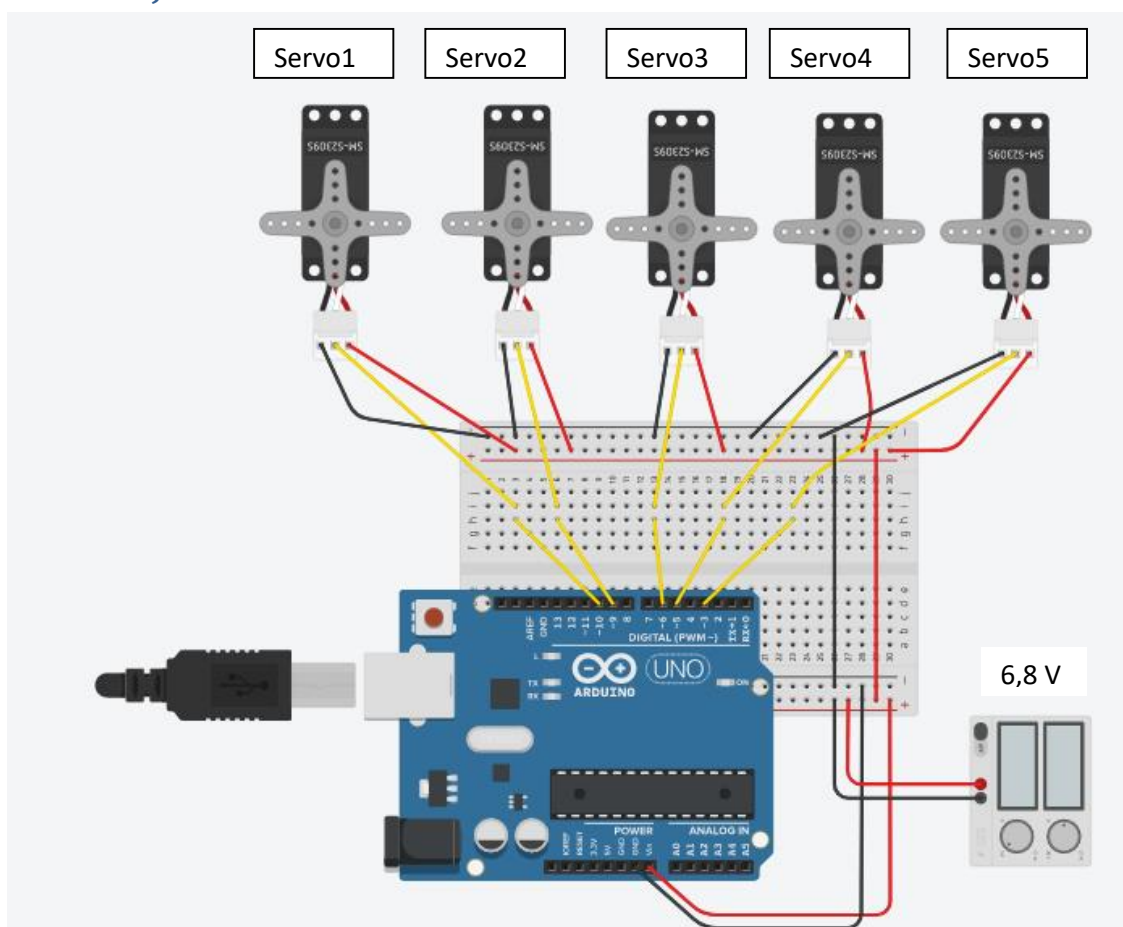
		PIN	ÁNGULOS		
SERVO 1	PINZAS	10	125° Abiertas		180° Cerradas
SERVO 2.	GIRO PINZAS	9	180°	105°	0°
SERVO 3	ARTIC. CABEZA Brazo 2	6	0°(adelante)	110°	180°(atrás)
SERVO 4	ARTIC. INFERIOR Brazo 1	5	180°(adelante)	90°	0°(atrás)
SERVO 5	GIRO BASE	3	180°	105°	0°

ALIMENTACIÓN DE 5 SERVOS:

Para alimentación de los 5 servos, usaremos una fuente de alimentación regulable externa de 1 A y 6,7 V. Se ha comprobado que en situaciones de torque excesivo, los servos no responden adecuadamente. (En este caso quitaremos la alimentación situando los servos en posición inicial manualmente). Aumentando el voltaje a 7,5V funciona adecuadamente. Usando una batería Ni-Cd de 7.2 V y 1600mAh el resultado es mejor.



MONTAJE PLACA.



PROGRAMACIÓN

INICIAR MOTORES:

```
//inicia motores
#include <Servo.h>

Servo servo1,servo2,servo3,servo4,servo5;

void setup()
{
  servo1.attach(10); //180 cerradas 125 abiertas
  servo2.attach(9); //105 centro 180 antihorario 0 horario
  servo3.attach(6); //110 recto, 0 adelante 180 atras
  servo4.attach(5); //90 recto 180 adelante 0 atras
  servo5.attach(3); //105 centro 180 antihorario 0 horario

  servo1.write(180);delay(100); //pinzas cerradas
  servo2.write(105);delay(100); //giro pinzas centro
  servo3.write(110);delay(100); //brazo s3 pinzas vertical
  servo4.write(90);delay(100); //brazo s4 articula base vertical
  servo5.write(105); delay(100); //giro base centro
}

void loop()
{
  servo1.write(125); delay(1000); //pinzas abiertas
  servo1.write(180); delay(1000); //pinzas cerradas
}
```

Inconveniente: El **servo5** soporta todo el peso del brazo robótico, produciendo balanceos. Sería recomendable una plataforma giratoria.

CONTROL DE TECLADO

```

#include <Servo.h>

Servo servo1,servo2,servo3,servo4,servo5;

int n = 0; //Entradas serie

int pinza=180,giropinza=105,brazo_s3=110,brazo_s4=90,girobase=105;

void setup(){
  servo1.attach(10); //180 cerradas 125 abiertas
  servo2.attach(9); //105 centro 180 antihorario 0 horario
  servo3.attach(6); //110 recto, 0 adelante 180 atrás
  servo4.attach(5); //90 recto 180 adelante 0 atrás
  servo5.attach(3); //105 centro 180 antihorario 0 horario
  servo1.write(180); delay(100); //pinzas
  servo2.write(105); delay(100); //giro pinzas
  servo3.write(110); delay(100); //articula pinzas
  servo4.write(90); delay(100); //articula base
  servo5.write(105); delay(100); //giro base

  Serial.begin(9600); //Ativando puerto serie
  Serial.flush(); //Limpiando memoria puerto serie
  //A Z pinzas, S X giro pinzas, D C giro brazo1, F V giro brazo2, G B base, R incia posición
}

void loop(){
  if(Serial.available(>0){ //Test puerto serie si recibe datos
    n = Serial.read(); //lectura de datos
    Serial.print("Valor leído: ");
    Serial.println(n); //Escribe el valor leído en consola
    switch (n)
    {
      case 65://A
        if (pinza>125) pinza=pinza-5;    break;

      case 90://Z
        if (pinza<180)pinza=pinza+5;    break;
      case 83://S
        if (giropinza>10)giropinza=giropinza-5;    break;
      case 88://X

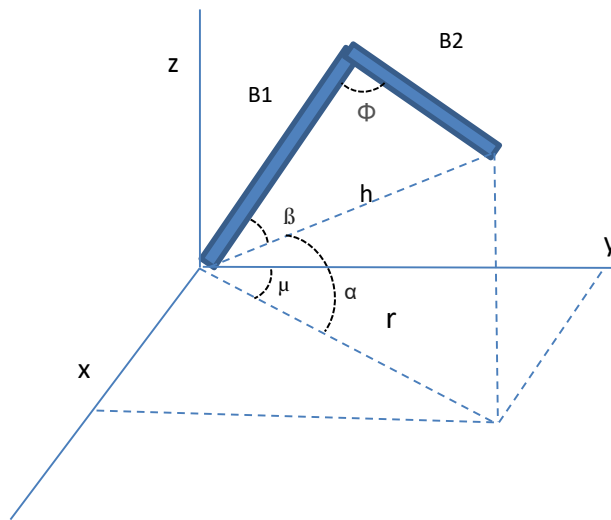
```

```
    if (giropinza<175) giropinza=giropinza+5;    break;
case 67://C
    if (brazo_s3>5) brazo_s3=brazo_s3-5;    break;
case 68://D
    if (brazo_s3<170) brazo_s3=brazo_s3+5;    break;
case 70://F
    if (brazo_s4>5) brazo_s4=brazo_s4-5;    break;
case 86://V
    if (brazo_s4<180) brazo_s4=brazo_s4+5;    break;
case 71://G
    if (girobase>15) girobase=girobase-5;    break;
case 66://B
    if (girobase<170) girobase=girobase+5;    break;
case 82:
    pinza=180,giropinza=105,brazo_s3=110,brazo_s4=90,girobase=105;    break;
}

servo1.write(pinza); delay(25);
servo2.write(giropinza); delay(25);
servo3.write(brazo_s3); delay(25);
servo4.write(brazo_s4); delay(25);
servo5.write(girobase); delay(25);

Serial.flush();
}
}
```

POSICIÓN (X,Y,Z) Y CÁLCULO ÁNGULOS



$$B1=10 \text{ cm} \quad B2=15 \text{ cm}$$

$$r^2 = x^2 + y^2 \quad h = \sqrt{r^2 + z^2} \quad \alpha = \text{arctg}\left(\frac{z}{r}\right)$$

$$\mu = \text{arctg}\left(\frac{x}{y}\right)$$

$$B_2^2 = h^2 + B_1^2 - 2hB_1\cos\beta \quad \beta = \arccos\left(\frac{h^2 + B_1^2 - B_2^2}{2hB_1}\right)$$

$$(\text{servo4}) \text{ángulo } B_1 = \alpha + \beta$$

$$h^2 = B_1^2 + B_2^2 - 2B_2B_1\cos\phi \quad \phi = \arccos\left(\frac{B_1^2 + B_2^2 - h^2}{2B_2B_1}\right)$$

$$(\text{servo3}) \text{ángulo } B_2 = \phi$$

$$\text{ángulo de la Base}(\text{servo5}) \mu = \text{arctg}\left(\frac{x}{y}\right)$$

POSICIONAMIENTO x,y,z. Programa

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

```
#include <math.h>
```

```
Servo servo3,servo4,servo5;
```

```
float MiACos(float R)
```

```
{
  float S, C, T;
  C = R;
  S = sqrt(1 - pow(R , 2));
  return atan2(S,C);
}
```

```
float MiASen(float R)
```

```
{
  float S, C;
  S = R;
  C = sqrt(1 - pow(R , 2));
  return atan2(S,C);
}
```

```
void posicion(int x,int y, int z, int &s3, int &s4, int &s5)
```

```
//Dasdas las coordenadas x,y,z calcula s3 brazo superior,s4 brazo inferior y s5 base
```

```
{
  float h,r,alfa,beta,fi,mu,b1,b2;
  b1=10;
  b2=15;
  r=pow(x*x+y*y,0.5);
  alfa=atan(z*1.0/r);
  h=pow(z*z+r*r,0.5);
  beta=MiACos((h*h+b1*b1-b2*b2)/(2*h*b1));
  s4=(alfa+beta)*180/3.1415;
  fi=MiACos((b1*b1+b2*b2-h*h)/(2*b1*b2));
  s3=fi*180/3.1415;
  s5=atan(x*1.0/y)*180/3.1415;
  return;
}
```



```
void movimiento(int s3,int s4,int s5)  
{  
  servo4.write(180-s4);//ajuste de ángulos al robot  
  delay(50);  
  if(s3<70) s3=70;  
  servo3.write(s3-70);//ajuste de ángulos al robot  
  delay(50);  
  servo5.write(105-s5);  
  delay(50);  
  return;  
}
```

```
void setup()
```

```
{  
  servo3.attach(6);  
  servo4.attach(5);  
  servo5.attach(3);  
}
```

```
void loop()
```

```
{  
  int x,y,z,s3,s4,s5;  
  
  posicion(0,0,25,s3,s4,s5);  
  movimiento(s3,s4,s5);  
  delay(1000);  
  posicion(10,15,5,s3,s4,s5);  
  movimiento(s3,s4,s5);  
  delay(1000);  
}
```

Inconveniente: Cambio de posición demasiado brusca.

RESOLUCIÓN DE MOVIMIENTO BRUSCO

```

#include <Servo.h>

#include <math.h>

Servo servo1,servo2,servo3,servo4,servo5;

int pos_servo3=110; int pos_servo4=90; int pos_servo5=105;

float MiACos(float R)
{
    float S, C, T;
    C = R;
    S = sqrt(1 - pow(R , 2));
    return atan2(S,C);
}

float MiASen(float R)
{
    float S, C;
    S = R;
    C = sqrt(1 - pow(R , 2));
    return atan2(S,C);
}

void posicion(int x,int y, int z, int &s3, int &s4, int &s5)
//Dasdas las coordenadas x,y,z calcula ang1 brazo inferior, ang2 brazo superior y ang3 de la
base
{
    float h,r,alfa,beta,fi,mu,b1,b2;
    b1=10;
    b2=15;
    r=pow(x*x+y*y,0.5);
    alfa=atan(z*1.0/r);
    h=pow(z*z+r*r,0.5);
    beta=MiACos((h*h+b1*b1-b2*b2)/(2*h*b1));
    s4=(alfa+beta)*180/3.1415;
    fi=MiACos((b1*b1+b2*b2-h*h)/(2*b1*b2));
    s3=fi*180/3.1415;
    s5=atan(x*1.0/y)*180/3.1415;
    return;
}

```

```

void movimiento(int s3,int s4,int s5,int &pos_servo3, int &pos_servo4, int &pos_servo5)
{
  s4=180-s4;
  if (s3<70) s3=70;s3=s3-70;
  s5=105-s5;
  //Cada for doble asegura si el angulo actual es mayor o menor
  //que la posición donde ir y además ralentiza el servo
  for(int i=pos_servo5;i<=s5;i++) {  servo5.write(i); delay(20); }
  for(int i=pos_servo5;i>=s5;i--) {  servo5.write(i); delay(20); }
  for(int i=pos_servo4;i<=s4;i++) {  servo4.write(i); delay(20); }
  for(int i=pos_servo4;i>=s4;i--) {  servo4.write(i); delay(20); }
  for(int i=pos_servo3;i<=s3;i++) {  servo3.write(i); delay(20); }
  for(int i=pos_servo3;i>=s3;i--) {  servo3.write(i); delay(20); }
  pos_servo3=s3;
  pos_servo4=s4;
  pos_servo5=s5;
  return;
}

```

```

void setup()
{
  servo1.attach(10); //180 cerradas 130 abiertas
  servo2.attach(9); //105 centro 160 antihorario 40 horario
  servo3.attach(6); //brazo2
  servo4.attach(5); //brazo1
  servo5.attach(3); //base
  //Iniciar motores
  servo1.write(180);  delay(100);
  servo2.write(105);  delay(100);
  servo3.write(110);  delay(100);
  servo4.write(90);   delay(100);
  servo5.write(105);  delay(100);
}

```

```
void loop()
{

  int x,y,z,s3,s4,s5;
  servo1.write(125);delay(100);

  posicion(0,15,-2,s3,s4,s5);
  //ajuste de ángulos al robot
  movimiento(s3,s4,s5,pos_servo3,pos_servo4,pos_servo5);
  delay(100);servo1.write(180);delay(1000);

  //SEGUNDO MOVIMIENTO

  posicion(-18,0,10,s3,s4,s5);
  //ajuste de angulos al robot
  movimiento(s3,s4,s5,pos_servo3,pos_servo4,pos_servo5);

  delay(100);
  servo2.write(20);
  delay(1500);
  servo1.write(125);
  delay(1000);
  servo2.write(105);
  delay(1500);
}
```

Posicionamiento siguiendo los puntos almacenados en un array.

```

#include <Servo.h>
#include <math.h>
Servo servo1,servo2,servo3,servo4,servo5;
int pos_servo3=110;
int pos_servo4=90;
int pos_servo5=105;
float MiACos(float R)
{
    float S, C, T;
    C = R;
    S = sqrt(1 - pow(R , 2));
    return atan2(S,C);
}
float MiASen(float R)
{
    float S, C;
    S = R;
    C = sqrt(1 - pow(R , 2));
    return atan2(S,C);
}
void posicion(int x,int y, int z, int &s3, int &s4, int &s5)
//Dasdas las coordenadas x,y,z calcula s3 brazo superior,s4 brazo inferior y s5 base
{ float h,r,alfa,beta,fi,mu,b1,b2;
    b1=10;    b2=15;
    r=pow(x*x+y*y,0.5);
    alfa=atan(z*1.0/r);
    h=pow(z*z+r*r,0.5);
    beta=MiACos((h*h+b1*b1-b2*b2)/(2*h*b1));
    s4=(alfa+beta)*180/3.1415;
    fi=MiACos((b1*b1+b2*b2-h*h)/(2*b1*b2));
    s3=fi*180/3.1415;
    s5=atan(x*1.0/y)*180/3.1415;
    return;
}

```



```

void movimiento(int s3,int s4,int s5,int &pos_servo3, int &pos_servo4, int &pos_servo5)
{
  s4=180-s4;
  if (s3<70) s3=70;s3=s3-70;
  s5=105-s5;
  //cada for doble asegura si el angulo actual es mayor o menor
  //que la posición donde ir y además ralentiza el servo
  for(int i=pos_servo5;i<=s5;i++) {  servo5.write(i); delay(20); }
  for(int i=pos_servo5;i>=s5;i--) {  servo5.write(i); delay(20); }
  for(int i=pos_servo4;i<=s4;i++) {  servo4.write(i); delay(20); }
  for(int i=pos_servo4;i>=s4;i--) {  servo4.write(i); delay(20); }
  for(int i=pos_servo3;i<=s3;i++) {  servo3.write(i); delay(20); }
  for(int i=pos_servo3;i>=s3;i--) {  servo3.write(i); delay(20); }
  pos_servo3=s3;
  pos_servo4=s4;
  pos_servo5=s5;
  return;
}

```

void setup()

```

{
  servo1.attach(10); //180 cerradas 130 abiertas
  servo2.attach(9); //105 centro 160 antihorario 40 horario
  servo3.attach(6); //brazo2
  servo4.attach(5); //brazo1
  servo5.attach(3); //base

  //Iniciar motores
  servo1.write(180); delay(100); //pinzas
  servo2.write(105); delay(100); //giro pinzas
  servo3.write(110); delay(100); //articula pinzas
  servo4.write(90); delay(100); //articula base
  servo5.write(105); delay(100); //giro base
}

```

```
void loop()  
{  
  int xy[5][3]={-7,15,2,7,1,20,7,10,2,-7,15,20,-7,1,2};  
  int x,y,z,s3,s4,s5;  
  for (int j=0;j<5;j++)  
  {  
    posicion(xy[j][0],xy[j][1],xy[j][2],s3,s4,s5);  
    movimiento(s3,s4,s5,pos_servo3,pos_servo4,pos_servo5);  
  
  }  
}
```

Depurando el programa anterior para mover los motores a la vez grado a grado.

```
#include <Servo.h>
#include <math.h>
Servo servo1,servo2,servo3,servo4,servo5;
int pos_servo3=110; int pos_servo4=90; int pos_servo5=105;
float MiACos(float R)
{
float S, C, T;
C = R;
S = sqrt(1 - pow(R , 2));
return atan2(S,C);
}
float MiASen(float R)
{
float S, C;
S = R;
C = sqrt(1 - pow(R , 2));
return atan2(S,C);
}
void posicion(int x,int y, int z, int &s3, int &s4, int &s5)
//Dasdas las coordenadas x,y,z calcula s3 brazo superior,s4 brazo inferior y s5 base
{
float h,r,alfa,beta,fi,mu,b1,b2;
b1=10;
b2=15;
r=pow(x*x+y*y,0.5);
alfa=atan(z*1.0/r);
h=pow(z*z+r*r,0.5);
beta=MiACos((h*h+b1*b1-b2*b2)/(2*h*b1));
s4=(alfa+beta)*180/3.1415;
fi=MiACos((b1*b1+b2*b2-h*h)/(2*b1*b2));
s3=fi*180/3.1415;
s5=atan(x*1.0/y)*180/3.1415;
return;
}
```

```

void movimiento(int s3,int s4,int s5,int &pos_servo3, int &pos_servo4, int &pos_servo5)
{
  s4=180-s4;
  if (s3<70) s3=70;s3=s3-70;
  s5=105-s5;
  //Usaremos un bucle while
  do
  {
    if(s5<pos_servo5) pos_servo5--;    if(s5>pos_servo5) pos_servo5++;
    servo5.write(pos_servo5); delay(10);
    if(s4<pos_servo4) pos_servo4--;    if(s4>pos_servo4) pos_servo4++;
    servo4.write(pos_servo4); delay(10);
    if(s3<pos_servo3) pos_servo3--;    if(s3>pos_servo3) pos_servo3++;
    servo3.write(pos_servo3); delay(10);
  }
  while(s3!=pos_servo3 || s4!=pos_servo4 || s5!=pos_servo5 );
  return;
}

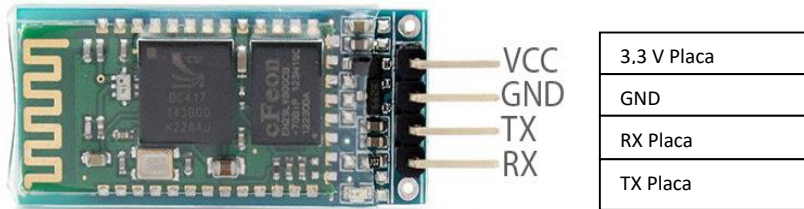
void setup()
{
  servo1.attach(10); //180 cerradas 130 abiertas
  servo2.attach(9); //105 centro 160 antihorario 40 horario
  servo3.attach(6); //brazo2
  servo4.attach(5); //brazo1
  servo5.attach(3); //base
}

void loop()
{
  int xy[5][3]={0,0,25,7,7,15,-7,7,15,0,10,10,0,20,5}; //0,0,25 Inicio motores
  int x,y,z,s3,s4,s5;
  for (int j=0;j<5;j++)
  {
    posicion(xy[j][0],xy[j][1],xy[j][2],s3,s4,s5);
    movimiento(s3,s4,s5,pos_servo3,pos_servo4,pos_servo5);
  }
}

```

CONTROL BLUETOOTH

Módulo HC-06



VCC, Voltaje positivo de alimentación. A 3,3 V

GND, Voltaje negativo de alimentación, se tienen que conectar al GND del Arduino.

TX, Pin de Transmisión de datos, por este pin el HC-06 transmite los datos que le llegan desde la PC o Móvil mediante bluetooth, este pin debe ir conectado al pin RX del Arduino.

RX, pin de Recepción, a través de este pin el HC-06 recibirá los datos del Arduino los cuales se transmitirán por Bluetooth, este pin va conectado al Pin TX del Arduino.

Usando el programa de **control de teclado (con el cable usb desconectado y alimentando el HC-06 a 3,3V de la placa)**, usaremos la siguiente aplicación appinventor para controlar el brazo robot:

INTERFACE



CÓDIGO APPINVENTOR:

```

when ListPicker1 .BeforePicking
do
  if BluetoothClient1 . Available
  then
    set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do
  set ListPicker1 . Selection to call BluetoothClient1 .Connect
  address ListPicker1 . Selection

when Abrir .Click
do
  call BluetoothClient1 .SendText
  text " A "

when Cerrar .Click
do
  call BluetoothClient1 .SendText
  text " Z "

when Gder .Click
do
  call BluetoothClient1 .SendText
  text " S "

when Gizq .Click
do
  call BluetoothClient1 .SendText
  text " X "

when M3atras .Click
do
  call BluetoothClient1 .SendText
  text " D "

when M3avanza .Click
do
  call BluetoothClient1 .SendText
  text " C "

when Reset .Click
do
  call BluetoothClient1 .SendText
  text " R "

when Baseizq .Click
do
  call BluetoothClient1 .SendText
  text " B "

when M4atras .Click
do
  call BluetoothClient1 .SendText
  text " F "

when Baseder .Click
do
  call BluetoothClient1 .SendText
  text " G "

when M4avanza .Click
do
  call BluetoothClient1 .SendText
  text " V "

```

ÁREA DE TRABAJO DEL BRAZO ROBOT EN ÁREA POSITIVA

esfera - GeoGebra

Superficie

$a = \text{Superficie} \left(25 \cos(t) \text{ sen}(v), 25 \text{ sen}(v) \text{ sen}(t), 25 \cos(v), t, 0, \pi, v, 0, \frac{\pi}{2} \right)$

$\rightarrow \begin{pmatrix} 25 \cos(t) \text{ sen}(v) \\ 25 \text{ sen}(v) \text{ sen}(t) \\ 25 \cos(v) \end{pmatrix}$

$b = \text{Superficie} \left(10 \cos(t) \text{ sen}(v), 10 \text{ sen}(v) \text{ sen}(t), 10 \cos(v), t, 0, \pi, v, 0, \frac{\pi}{2} \right)$

$\rightarrow \begin{pmatrix} 10 \cos(t) \text{ sen}(v) \\ 10 \text{ sen}(v) \text{ sen}(t) \\ 10 \cos(v) \end{pmatrix}$

