Proyecto Informatico I

Servo brazo de 2 articulaciones con pinza

**Alumna: Ludmila Jazmin Rovner**

**Curso: 4° 1° Computación TT**

**Grupo: Ludmila Rovner, Thiago Ramirez, Jostin Navarro y Tomas Herrero.**

**Proyecto: Movimiento de servo brazo de 3 articulaciones con pinza.**

****

Viernes 27.09.24

**DOF: Geometría y grados de Libertad.**

❤ **DOF eje Z:** Permite orientar el brazo de manera horizontal.

❤ **DOF eje X-Y:** Con 1 solo DOF en el eje X-Y sólo podemos girar el brazo por lo que los movimientos se limitan en un círculo del plano X-Y, contando el DOF del eje Z sería una semiesfera.

★ 2 DOF X-Y: Triángulo articulado que puede posicionarse en una zona restringida por las longitudes de los tramos del brazo. No se puede elegir el ángulo con el que se llega al punto elegido. La solución a la cinemática**\*** es más fácil y única, para cada posición posible.

★ 3 DOF X-Y: Cuadrilátero articulado, mecanismo frecuente para resolver la cinemática inversa de un robot articulado. Posiciona el extremo en cualquiera de los puntos de alcance y el ángulo del ataque del efector. La resolución de su cinemática es sencilla y única si definimos como parámetro el ángulo de ataque, sino hay infinitas soluciones.

❤ **DOF garra:** Técnicamente no es un DOF porque no participa en la resolución de la posición del brazo pero es el encargado de abrir y cerrar la garra.

❤ **DOF adicionales:** Se pueden agregar más DOF a lo largo del brazo para mayor movilidad. Por ejemplo, en el extremo se pueden agregar uno o dos que funcionan como una “muñeca”.

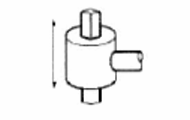
**\***Cinemática: estudio de la descripción del movimiento de un cuerpo, dividiéndolo entre movimientos de traslación y rotación.

Viernes 04.10.24

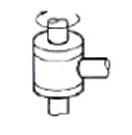
**Estructura mecánica de un brazo robótico.**

Un brazo robótico está hecho por una serie de eslabones unidos con articulaciones que le dan movimiento a los eslabones.

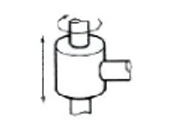
Tipos de articulaciones:

❤ **Prismática:** Movimiento sobre el eje Y con un grado de libertad.

❤ **Rotacional:** Movimiento sobre el eje Z con un grado de libertad.



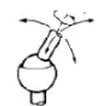
❤ **Cilíndrica:** Movimiento sobre los ejes Y y Z con 2 grados de libertad.



❤ **Planar:** Movimiento sobre los ejes Y y X con 2 grados de libertad.

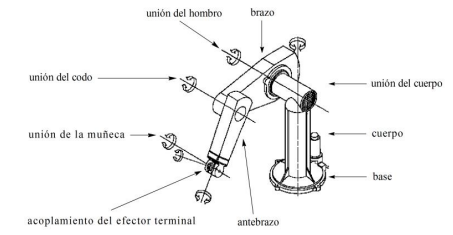


❤ **Esférica:** Movimiento sobre los ejes Y, X y Z con 3 grados de libertad.



Martes 08.10.24

**Partes de un brazo robótico:**



**Elementos terminales:**

El elemento terminal es el eslabón final, el que interactúa con los elementos que rodean al robot. Estos se dividen por:

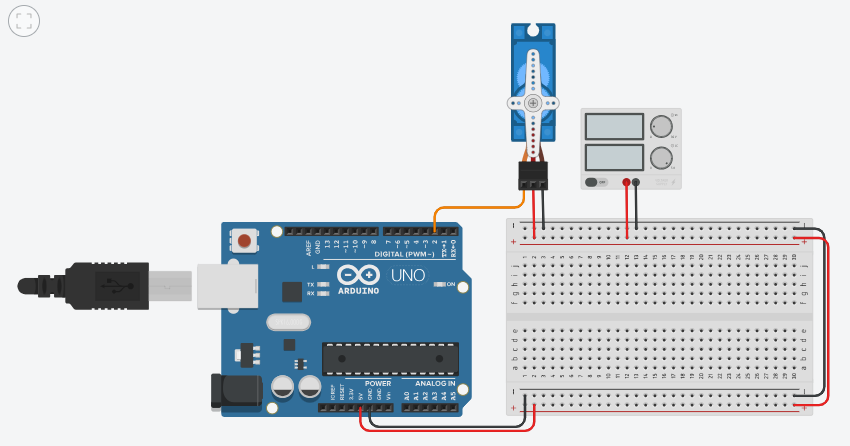
❤ **Herramienta de sujeción:** se usan para agarrar objetos. El ejemplo más común es la garra, pero también se pueden utilizar elementos de agarre electromagnético o ventosas.

❤ **Herramientas:** se usan para que el robot realice tareas específicas. Lo más común es que estos vengan con un aparato portaherramientas que nos permite cambiar las herramientas para mayor productividad.

Para el proyecto que vamos a realizar con mis compañeros, vamos a necesitar una herramienta de sujeción, como una pinza.

Viernes 18.10.24

**Emprolije las conecciones y el código que hizo Thiago:**

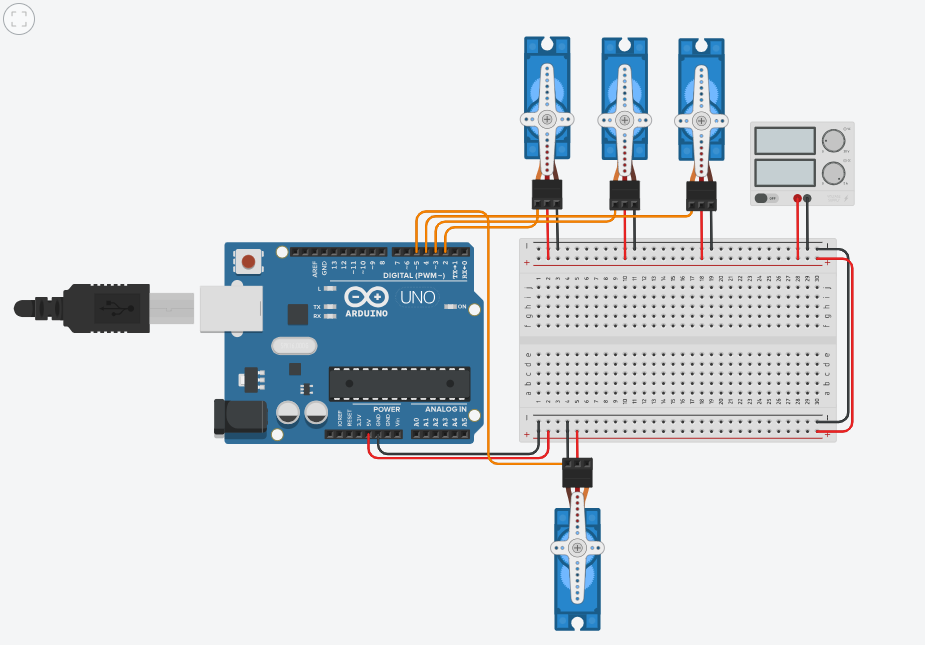


| #include <Servo.h> Servo servo.1; //Se declara el servomotor int variable = 0; //Se declara la variable void setup()  { servo1.attach(2); //Se define el pin delay(1000); servo1.write(90): //Se define la posición inicial del servo } void loop() { while (variable<1){  servo1.write(0);  delay(1000);  servo1.write(90);  delay(1000);  servo1.write(180);  delay(1000)  } } |
| --- |

Este código lo utilizamos para aprender junto con mis compañeros para aprender a utilizar los servo motores desde tinkercad. Así realizamos los primeros movimientos de los servos, llevándolos a 3 posiciones distintas según los grados que le indicamos.

Viernes 25.10.24

**Conexión de los 4 servo motores.**

****

Realice la conección en tinkercad de otros 3 servo motores para intentar un código en el que podamos mover más de un servo motor a la vez.

**Código base de la última conexión**

| #include <Servo.h> Servo servo1; Servo servo2;  Servo servo3;  Servo servo4;   int bro = 0;   void setup() {  servo1.attach(2);  servo2.attach(3);  servo3.attach(4);  servo4.attach(5); } void loop() {  bro = 90;  servo1.write(bro);   delay(2000);   Servo2.write(bro);  delay(2000);   Servo3.write(bro);  delay(2000);   Servo4.write(bro);  bro = 180;  servo1.write(bro);   delay(2000);   Servo2.write(bro);  delay(2000);   Servo3.write(bro);  delay(2000);   Servo4.write(bro);  bro = 0;  servo1.write(bro);   delay(2000);   Servo2.write(bro);  delay(2000);   Servo3.write(bro);  delay(2000);   Servo4.write(bro);  } |
| --- |

Este código lo realizamos con mi compañero de grupo Thiago Ramirez para mover varios servos al mismo tiempo y a distintas posiciones.

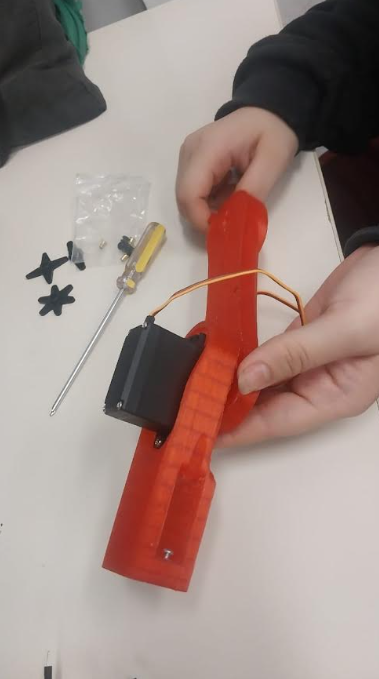
**❤** El código no funcionó porque declare los nombres de los servomotores con s y a la hora de escribir el código escribí los nombres con S. Corrección:

| #include <Servo.h> Servo servo1; Servo servo2;  Servo servo3;  Servo servo4;   int bro = 0;   void setup() {  servo1.attach(2);  servo2.attach(3);  servo3.attach(4);  servo4.attach(5); } void loop() {  bro = 90;  servo1.write(bro);   delay(2000);   servo2.write(bro);  delay(2000);   servo3.write(bro);  delay(2000);   servo4.write(bro);  bro = 180;  servo1.write(bro);   delay(2000);   servo2.write(bro);  delay(2000);   servo3.write(bro);  delay(2000);   servo4.write(bro);  bro = 0;  servo1.write(bro);   delay(2000);   servo2.write(bro);  delay(2000);   servo3.write(bro);  delay(2000);   servo4.write(bro);  } |
| --- |

**Construcción del Brazo**

Después de utilizar Tinkercad para aprender a conectar los servomotores, como definirlos y como enviarles señales desde nuestro código pudimos empezar con el proyecto en su parte física. Comenzamos armando el brazo robótico, con mis compañeros nos dividimos las partes del brazo para agilizar el trabajo y poder terminar el armado más rápido..

Yo me encargue del primer servo motor, el que ocupa el lugar del “codo” en el servo brazo, uniendo la mitad del brazo. Una vez que terminé con esa parte, inicié la instalación del servomotor que va en la parte de la “muñeca” de nuestro robot.

****

Junto con Jostin Navarro y Thiago Ramirez intentamos realizar la primera conexión de uno de los servomotores para poder moverlo, pero esta no funcionó. En realidad, la conexión si funcionaba pero el servomotor se movía de manera muy lenta, casi imperceptible.

Después de intentar con distintos códigos, cambiando cables, el servo y la placa arduino, llegamos a la conclusión de que el problema era que la computadora no era una fuente de alimentación suficiente como para poder mover nuestros servomotores.

**Jueves 31.10.24**

El día de hoy concurrimos con uno de mi compañero (Thiago Ramirez) al laboratorio durante el horario de Base de datos (Con consentimiento de la profesora Sofía Marmol) para continuar con el proyecto.

**Ajuste de los servos**

Tuvimos que desarmar el servo brazo porque los servomotores no estaban en la posición correcta. Pasamos lo que quedó del horario ajustando nuestros servomotores para que todos queden en la posición 0. Luego de ajustar todos los motores, volví a armar todo el brazo robótico.

También probamos un código para mover distintos servomotores a la vez y llevarlos a distintas posiciones indicadas en el mismo código, pero el robot de volvió loco y no funcionó.

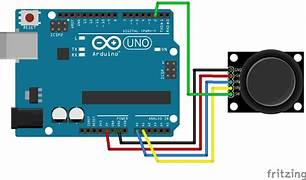
| #include <Servo.h> Servo Giro; Servo Hombro; Servo Codo; Servo Muneca1; Servo Muneca2; Servo Pinza;  void setup() {  Giro.attach(2);  Hombro.attach(3);  Codo.attach(4);  Muneca1.attach(5);  Muneca2.attach(6);  Pinza.attach(7); } void loop() {    Giro.write(0);//Movimiento 1  delay(2000);  Hombro.write(0);  delay(2000);  Codo.write(180);  delay(2000);  Muneca1.write(0);  delay(2000);  Muneca2.write(0);  delay(2000);  Pinza.write(0);  delay(2000);    Giro.write(90);//Movimiento 2  delay(2000);  Hombro.write(90);  delay(2000);  Codo.write(0);  delay(2000);  Pinza.write(90);  delay(2000);  Muneca2.write(90);//Movimiento 3  delay(2000);  Codo.write(90);  delay(2000);  Hombro.write(45);  delay(2000);  Pinza.write(0);  delay(2000);    Muneca2.write(0);//Movimiento 4  delay(2000);  Codo.write(180);  delay(2000);  Hombro.write(90);  delay(2000);  Giro.write(0);  delay(2000);    Hombro.write(45);//Movimiento 5  delay(2000);  Codo.write(80);  delay(2000);  Pinza.write(0);  delay(2000);    Muneca2.write(0);//Movimiento 6  delay(2000);  Codo.write(180);  delay(2000);  Pinza.write(0);  delay(2000);  Hombro.write(180);  delay(2000);  Codo.write(0);  delay(2000); } |
| --- |

**Investigación del joystick**

Junto con Thiago Ramirez realizamos la investigación sobre el funcionamiento, las conexiones y cómo escribir el código para controlar el servo brazo con un joystick.

El pin GND del joystick se conecta, valga la redundancia, al GND de la placa de arduino, el pin VCC se conecta al 5V del arduino. Los pines sobre los ejes X e Y (X-Axis e Y-Axis) se conectan a las salidas analogicas del arduino, como el A0 y el A1. Y el pin digital D2 se conecta a las salidas digitales de la placa.

Cuando la palanca del joystick se mueve, este envía señales eléctricas que son traducidas y leídas por el microcontrolador de la placa de arduino. Estas señales controlan la dirección y ángulos en los que se mueven los servomotores de nuestro proyecto.



[**https://programarfacil.com/blog/arduino-blog/joystick-con-arduino/**](https://programarfacil.com/blog/arduino-blog/joystick-con-arduino/)

[**https://www.wexterhome.com/curso-arduino/modulo-joystick-y-entradas-analogicas/#google\_vignette**](https://www.wexterhome.com/curso-arduino/modulo-joystick-y-entradas-analogicas/#google_vignette)

[**https://rodolfoferro.xyz/connect-joystick-to-arduino/**](https://rodolfoferro.xyz/connect-joystick-to-arduino/)

Estas son las fuentes que utilizamos para nuestra investigación.

**Código del joystick**

También investigamos en google como realizar un código para que el joystick funcione.

| void setup(){  pinMode(9, OUTPUT); } void loop(){ xAxis = analogRead(A0) yAxis = analogRead(A1);  if (xAxis > 512) {  digitalWrite(9, HIGH); } else if (xAxis < 512){  digitalWrite(9, LOW); } else {  digitalWrite(9, LOW); } }int xAxis, yAxis; |
| --- |

**Construcción del servo brazo**

Luego de terminar mi horario y que Thiago (rata traicionera) se vaya y me deje sola, me quede en el horario de la noche para terminar la construcción del robot.Fue difícil terminarlo porque había varios de los tornillos que traían los servomotores eran muy chiquitos para los agujeros del brazo, así que pasamos como media hora buscando tornillos que puedan adaptarse al servo brazo junto con el profesor Gonzalo Consorti. Una vez que los encontramos, termine de unir casi todas las articulaciones con los servomotores.

**Fuente de poder**

Intentamos distintas fuentes de poder ajenas a la computadora para intentar mover los servos. Primero utilizamos una que movía el servo directamente pero el profesor no sabía cómo utilizarla, yo menos. Así que perdimos una clase casi entera en el turno noche porque esa fuente no sirvió de nada.

**Jueves 07.11.24**

El día de la fecha me presenté en el laboratorio previo a mi horario escolar (De 09:00 a 13:30) para seguir con el proyecto informático y con la carpeta de campo.

**Construcción del servo brazo**

Terminé de instalar el último servomotor con los tornillos encontrados en la última clase para poder finalizar, de una vez, el armado del brazo robótico.

**Normas APA**

Actualice la carpeta de campo luego de la última publicación del profesor Gonzalo Consorti sobre las normas APA para cumplir con los requisitos de la entrega.

**Fuente de poder**

Luego de intentar la última fuente y que no funcione comenzamos a utilizar la siguiente fuente para intentar darle vida a nuestro pobre brazo disfuncional:



Esta fuente permite alimentar una protoboard de 400 a 830, suministra alimentaciones de 5V o 3.3V. Es compatible con las protoboards que tengan rieles o líneas de VCC – GND. Es un dispositivo que suministra energía (voltaje y corriente) a los circuitos montados en una placa de pruebas. Conecta la fuente a una toma de corriente o batería, y distribuye voltaje (5V, 3.3V) y GND a las líneas de alimentación de la protoboard.

Código del Joystick

El profesor nos ayudó a realizar un código con el que podamos mover los servomotores con las señales analógicas enviadas por los joysticks.

| #include <Servo.h>  Servo myservo; // create servo object to control a servo // twelve servo objects can be created on most boards  int pos = 0; // variable to store the servo position  void setup() {  pinMode(A5,INPUT);  pinMode(A4,INPUT);  myservo.attach(9); // attaches the servo on pin 9 to the servo object  int pos = 0;  Serial.begin(9800); }  void loop() {   Serial.print( "x:" );  Serial.print( analogRead(A0) );  Serial.print( " y:" );  Serial.println( analogRead(A1) );    if(analogRead(A1) <= 800 && pos <= 170)  {  pos += 2;  myservo.write(pos); // tell servo to go to position in variable 'pos'  delay(15);  }else if(analogRead(A0) <= 800 && pos >= 20)  {  pos -= 2;  myservo.write(pos); // tell servo to go to position in variable 'pos'  delay(15);  } } |
| --- |

Aun así no terminaba de funcionar completamente porque estábamos utilizando un joystick para mover un solo servomotor, por lo que el joystick funcionaba en L. O sea que únicamente se movía hacia un lado del eje X y un lado del eje Y.

**Jueves 14.11.24**

Luego de que el profesor nos ayude arreglando los problemas de posición del joystick, probamos un nuevo código para mover 2 servomotores:

| Serial.print( "x:" );  Serial.print( analogRead(A2) );  Serial.print( " y:" );  Serial.println( analogRead(A3) );    if(analogRead(A2) <= 800 && pos <= 170)  {  pos += 2;  myservo2.write(pos);   delay(15);  }else if(analogRead(A3) <= 800 && pos >= 20)  {  pos -= 2;  myservo2.write(pos);   delay(15);  } } |
| --- |

Con este código podiamos moverlos pero los servos realizaban movimientos muy bruscos y nuestro robot intentaba matarse así que corregimos las posiciones que definimos en el código para ajustarlas mejor a los problemas que maneja Adolfo.

| if(analogRead(A2) <= 350 && pos <= 170)  {  pos += 2;  myservo2.write(pos);   delay(15);  }else if(analogRead(A3) <= 900 && pos >= 20)  {  pos -= 2;  myservo2.write(pos);   delay(15);  } } |
| --- |

**Viernes 22.11.24**

**Fuente de poder**

La última fuente que intentamos funcionaba para mover uno o dos servos pero seguía sin ser suficiente para mover todas las articulaciones de brazo robótico(Adolf) de una vez, así que el profesor nos dio una nueva fuente(la batería) para seguir probando.

**Joystick**

También agregamos una parte nueva al códuigo para imprimir todos los movimientos que realicen los joysticks, así podemos ver y controlar las posiciones y los ángulos en los que se mueven las distintas articulaciones de nuestros servos

| void loop() {  Serial.print ( "x:" ) ;  Serial.print ( analogRead(A0) ) ;  Serial.print ( " y:" ) ;  Serial.print ( analogRead(A1) ) ;  Serial.print ( " w:" ) ;  Serial.print ( analogRead(A2) ) ;  Serial.print ( " z:" ) ;  Serial.print ( analogRead(A3) ) ;   Serial.print ( "\t\t") ;    Serial.print ( "G:" ) ;  Serial.print (posG ) ;  Serial.print ( " " ) ;  Serial.print ( "H:" ) ;  Serial.print ( posH ) ;  Serial.print ( " " ) ;  Serial.print ( "C:" ) ;  Serial.print (posC ) ;  Serial.print ( " " ) ;  Serial.print ( "M1:" ) ;  Serial.print ( posM1 ) ; |
| --- |

Perdimos mucho tiempo intentando realizar lo de imprimir los movimientos del joystick así que no llegamos a realizar la conección en arduino de todo el servo. Lo dejamos para la siguiente clase.

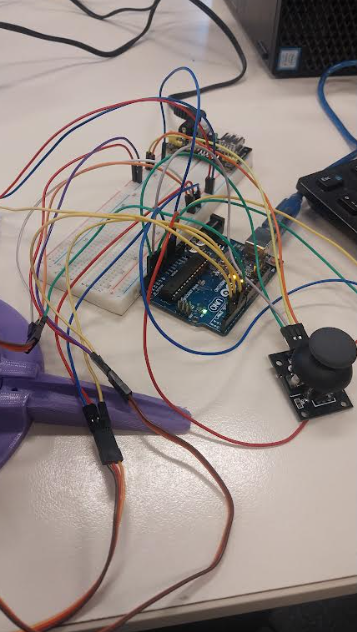
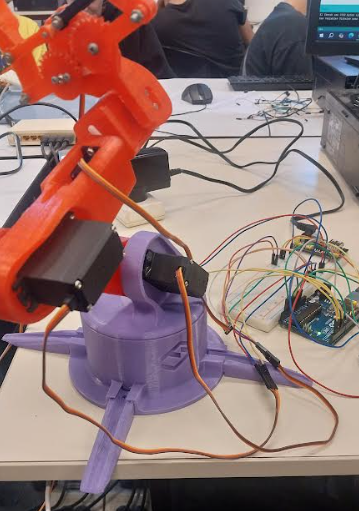
Viernes 29.11.24

Volvimos a realizar el código de la siguiente clase porque nos olvidamos de guardarlo. También le agregamos una última parte al código que nos permite utilizar el botón digital del joystick para utilizar la garra del servo brazo. Esta se cerrará cuando el botón esté presionado y se abrirá cuando deje de presionarse.

| Servo pinza; int pinBoton = 13;  bool boton = false;  void setup() {  pinMode(13,INPUT\_PULLUP);   pinza.attach(12);  Serial.begin(9800); }  void loop() {   if( digitalRead(13) == LOW )  {  pinza.write(0);  }else if( digitalRead(13) == HIGH )  {  pinza.write(50);  } |
| --- |

**Conexión final**

Realizamos la última conexión del servo brazo a la protoboard para probar el código final del proyecto, que nos permitiría mover todas las articulaciones del brazo robótico utilizando el joystick.



**Código Final**

Terminamos de unir todos los códigos funcionales que desarrollamos a lo largo del proyecto, como el código en el que movemos los servo motores con ayuda de los joysticks, en el que imprimimos los movimientos de estos, el código en el que utilizamos el botón digital del joystick para manejar la garra; y así pudimos finalizar todas las funciones propuestas para nuestro proyecto.

O sea que con la conexión que realizamos el día de hoy y la unión de todos los códigos escritos por ahora, finalmente podemos decir que terminamos nuestro proyecto informático.

**Código:**

| #include <Servo.h>  Servo giro;  Servo hombro; Servo codo; Servo muneca1; Servo muneca2; Servo pinza;  int posG = 100; int posH = 30; int posC = 130; int posM1 = 50;  int posM2 = 50; int posP = 50;   void setup() {    pinMode(A0,INPUT);  pinMode(A1,INPUT);  pinMode(A2,INPUT);  pinMode(A3,INPUT);  pinMode(A4,INPUT);  pinMode(13,INPUT\_PULLUP);      giro.attach(7);  hombro.attach(8);  codo.attach(9);  muneca1.attach(10);  muneca2.attach(11);  pinza.attach(12);     Serial.begin(9800); }  void loop() {  //joystick uno(giro X & hombro Y)    Serial.print ( "x:" ) ;  Serial.print ( analogRead(A0) ) ;  Serial.print ( " y:" ) ;  Serial.print ( analogRead(A1) ) ;  Serial.print ( " w:" ) ;  Serial.print ( analogRead(A2) ) ;  Serial.print ( " z:" ) ;  Serial.print ( analogRead(A3) ) ;  Serial.print ( " m:" ) ;  Serial.print ( analogRead(A4) ) ;   Serial.print ( "\t\t") ;    Serial.print ( "G:" ) ;  Serial.print (posG ) ;  Serial.print ( " " ) ;  Serial.print ( "H:" ) ;  Serial.print ( posH ) ;  Serial.print ( " " ) ;  Serial.print ( "C:" ) ;  Serial.print (posC ) ;  Serial.print ( " " ) ;  Serial.print ( "M1:" ) ;  Serial.print ( posM1 ) ;  Serial.print ( " " ) ;  Serial.print ( "M2:" ) ;  Serial.print ( " " ) ;  Serial.print ( posM2 );   Serial.print ( "\t\t") ;  Serial.println ( digitalRead(13) ) ;    if(analogRead(A0) <= 350 && posG <= 170)  {  posG += 1;  giro.write(posG);  delay(50);  }else if(analogRead(A0) >= 900 && posG >= 20){  posG -= 1;  giro.write(posG);  delay(50);  }     if(analogRead(A1) <= 350 && posH <= 170)  {  posH += 1;  hombro.write(posH);  delay(50);  }else if(analogRead(A1) >= 900 && posH >= 20){  posH -= 1;  hombro.write(posH);  delay(50);  }   if(analogRead(A2) <= 350 && posC <= 170)  {  posC += 1;  codo.write(posC);  delay(50);  }else if(analogRead(A2) >= 900 && posC >= 20){  posC -= 1;  codo.write(posC);  delay(50);  }      if(analogRead(A3) <= 350 && posM1 <= 170)  {  posM1 += 1;  muneca1.write(posM1);  delay(30);  }else if(analogRead(A3) >= 1000 && posM1 >= 20){  posM1 -= 1;  muneca1.write(posM1);  delay(50);  }   if(analogRead(A4) <= 350 && posM2 <= 170)  {  posM2 += 1;  muneca2.write(posM2);  delay(30);  }else if(analogRead(A4) >= 1000 && posM2 >= 20){  posM2 -= 1;  muneca2.write(posM2);  delay(50);  }   if( digitalRead(13) == LOW )  {  pinza.write(0);  }else if( digitalRead(13) == HIGH )  {  pinza.write(50);  }  } |
| --- |

* Ludmila Rovner 4° 1° C t.t -

No se olvide de descansar y tomar agua.