PROYECTO FINAL



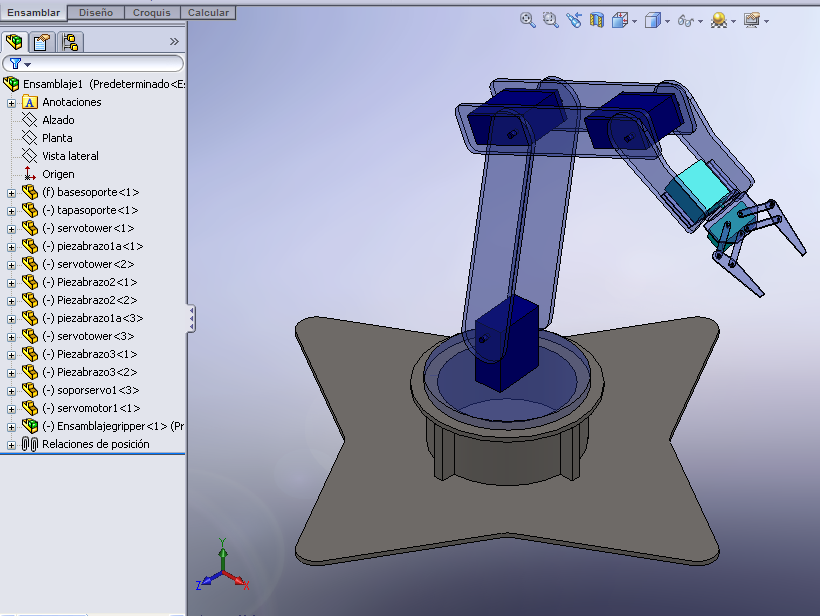
Universidad Nacional de Ingeniería

Facultad de Ingeniería Mecánica

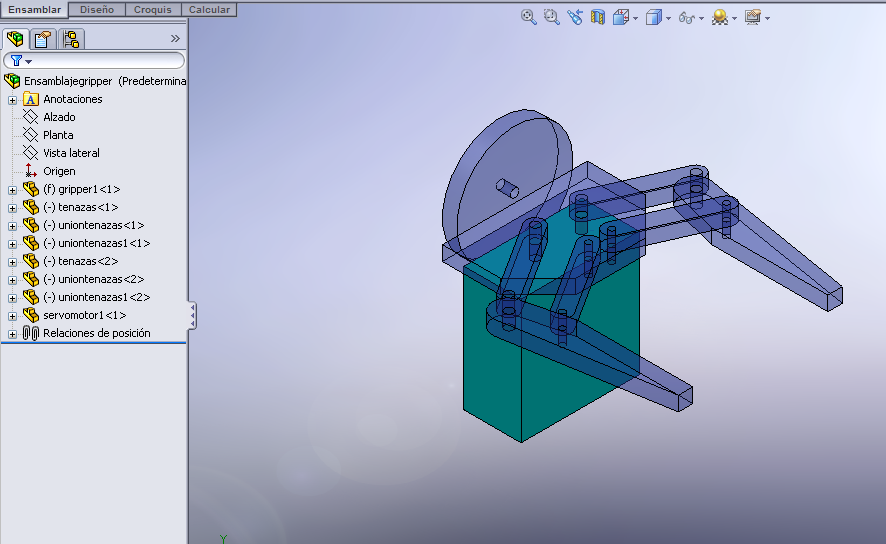
|  |  |
| --- | --- |
| PROFESOR | Ing. Vinces Ramos, Nikolai |
| CURSO | Análisis y control de robots |
| CÓDIGO | MT517 |
| SECCIÓN | B |
| PERÍODO ACADÉMICO: | 2012-II |
| INTEGRANTES | Monroe Diaz, Kevin  Meléndez Rodríguez Terry Andy  Saturno Rivera Briner  Villacorta Nervi Joaquín  Contreras Martínez Dimel Arturo  Veramendi Espinoza, Roberto  Mucha soto jesus miguel  2012 |
|  |  |

**DISEÑO EN SOLIDWORKS**

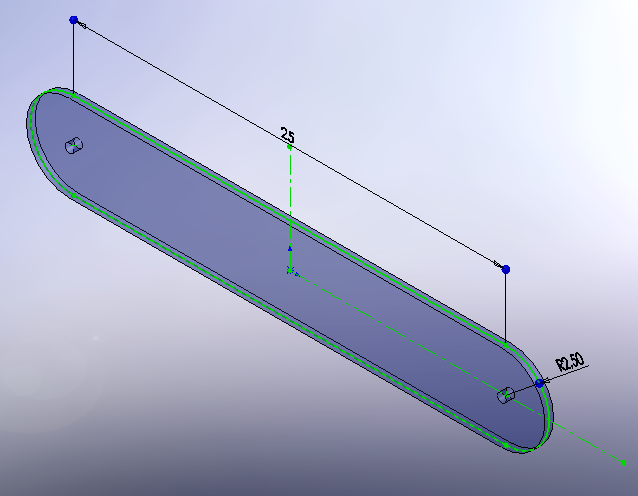
Brazo puma:

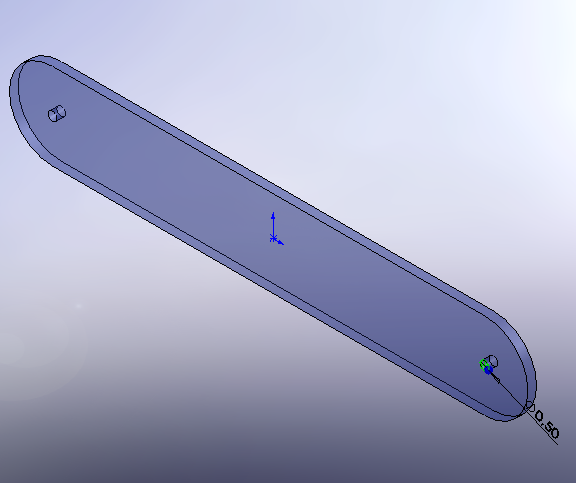


Gripper:

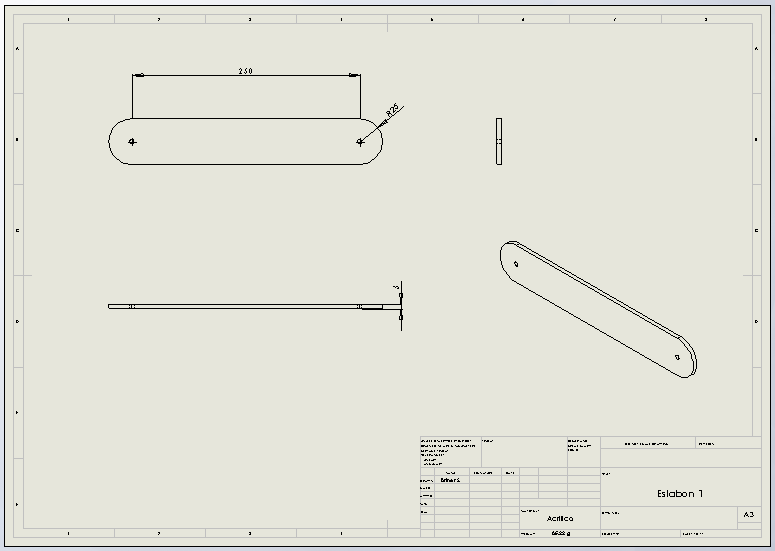


Eslabón 1:

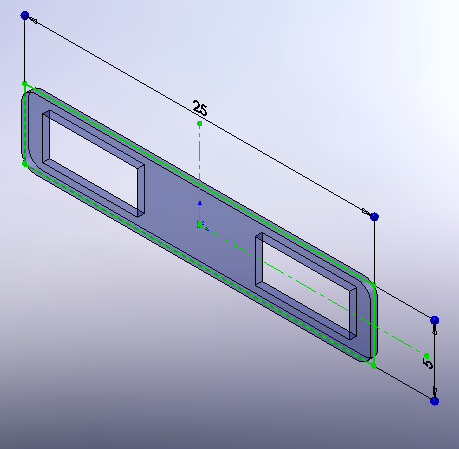


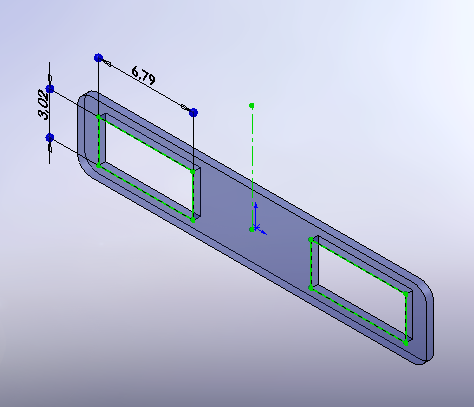


PLANO ESLABON 1:

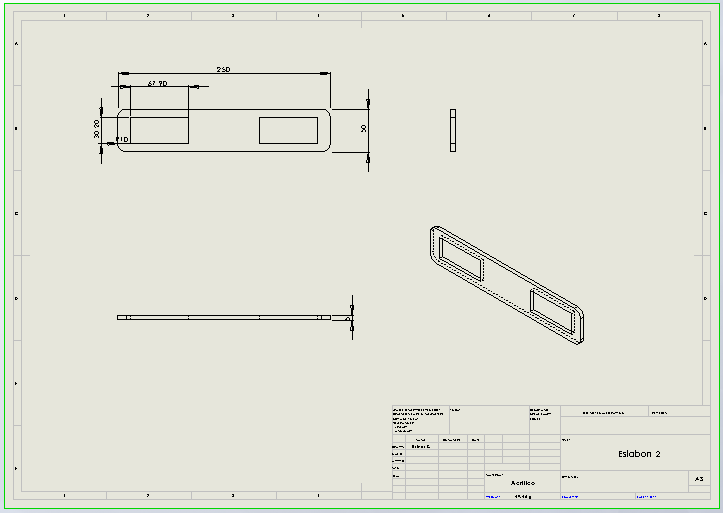


Eslabón 2:

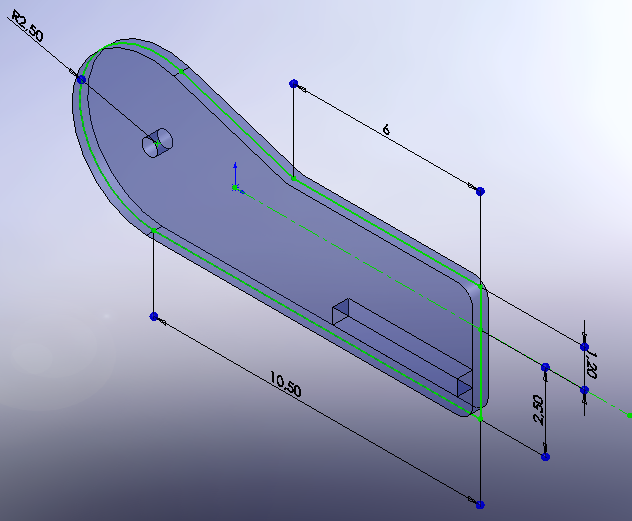


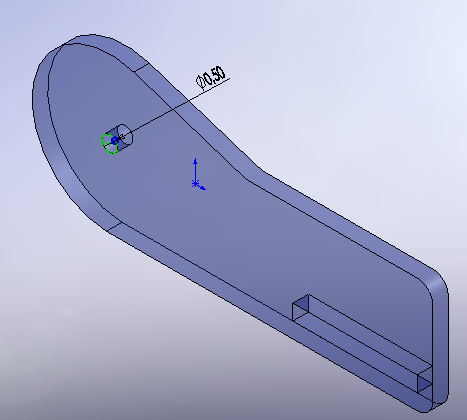


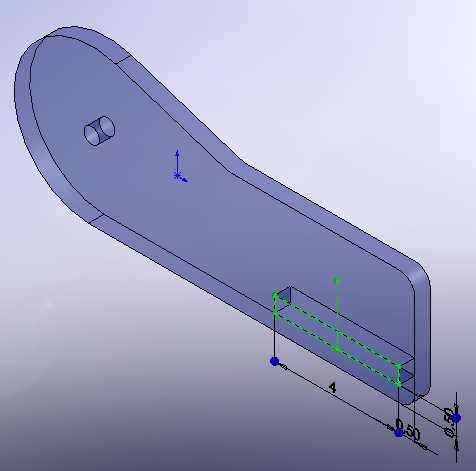
PLANOS ESLABON 2:



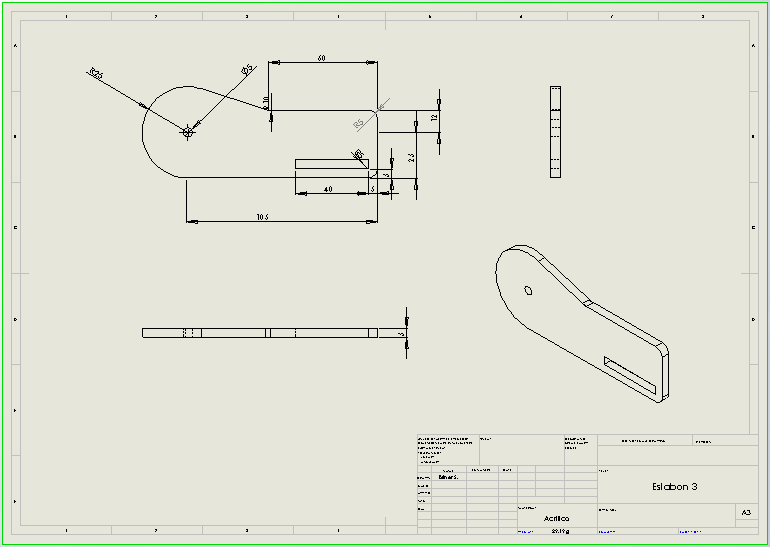
Eslabón 3:



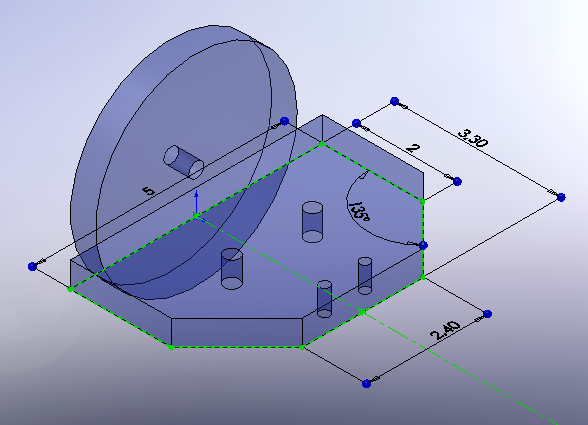


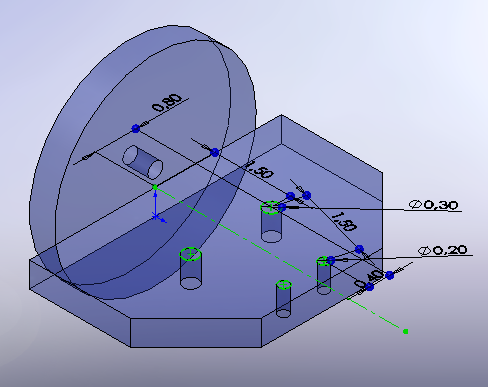


PLANOS ESLABON 3:

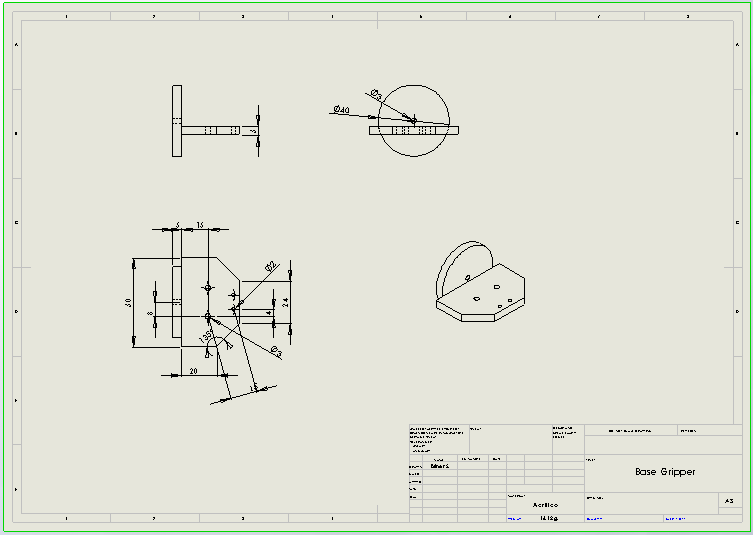


BASE GRIPPER:

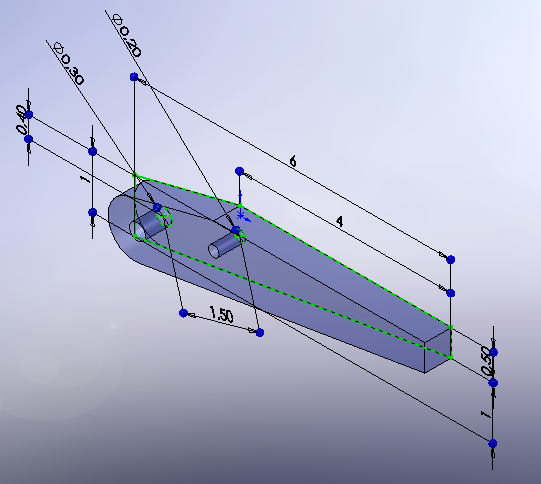




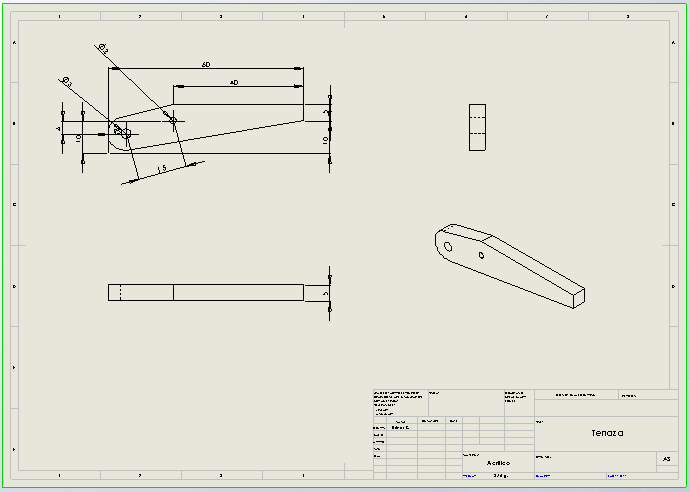
PLANOS BASE GRIPPER:



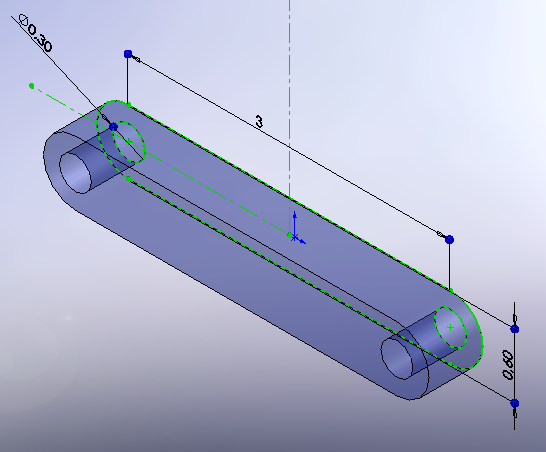
Tenazas:



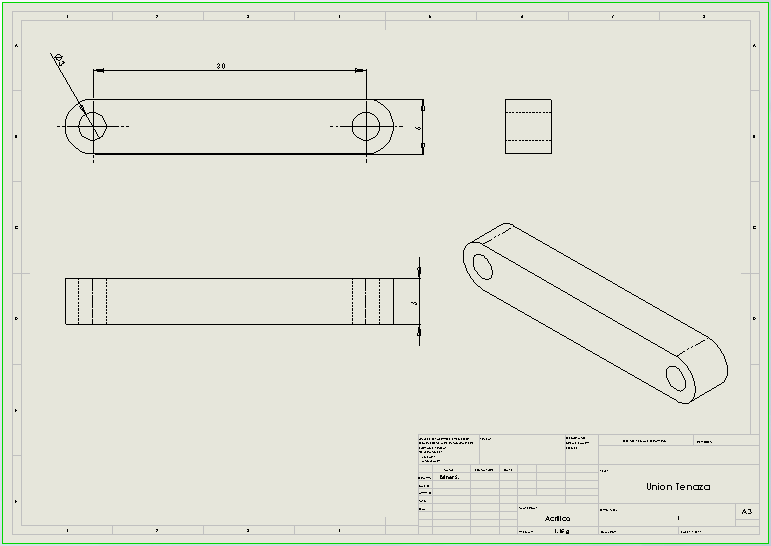
PLANOS TENAZAS:



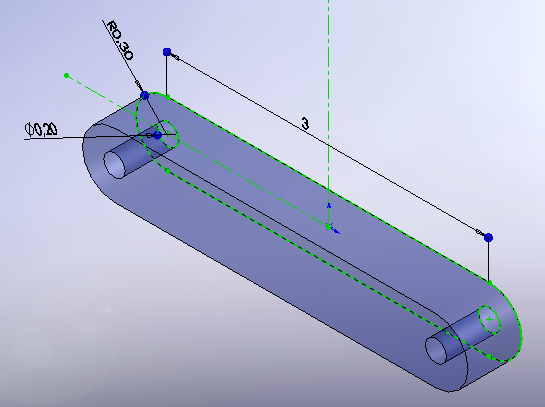
UNION 1 de TENAZAS:



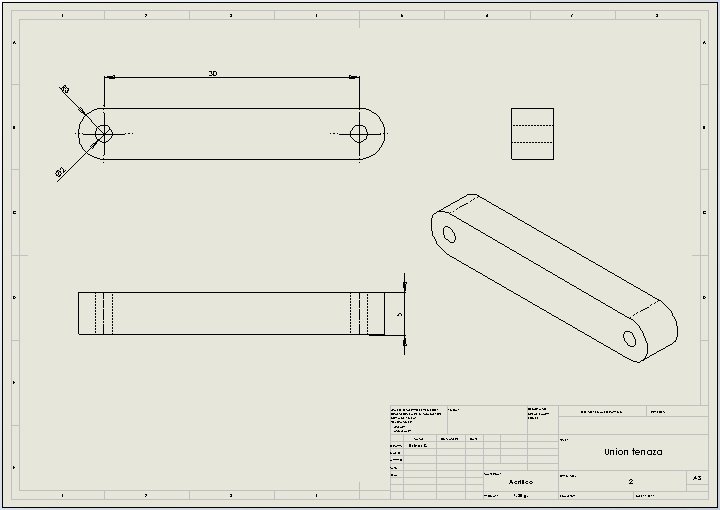
PLANOS DE UNION 1 de TENAZAS:



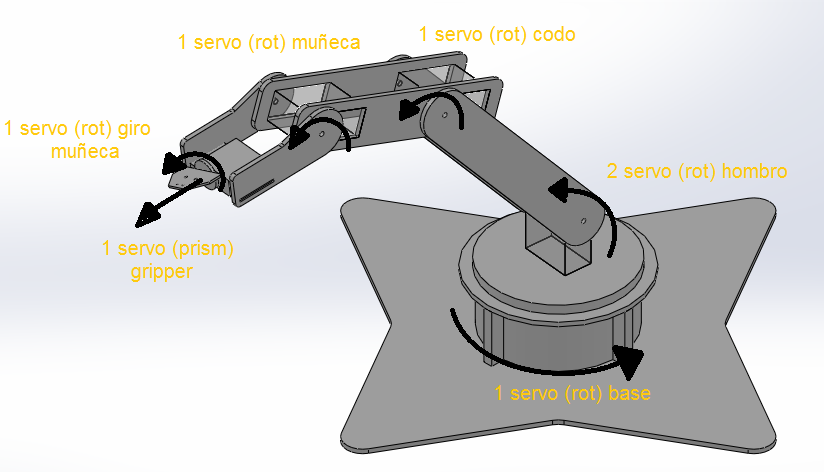
UNION 2 de TENAZAS:



PLANOS DE UNION 2 de TENAZAS:



**GRADOS DE LIBERTAD DEL BRAZO PUMA (5 G.L. 1 on-off)**



Material: Acrílico (densidad 1.19 g/cm^3)

**SERVO MOTOR A USAR**

* 1 servo (rot) base - 22kg: masa = 180g “9805MG”
* 2 servo (rot) hombro – 12kg c/u: masa = 55g “MG945”
* 1 servo (rot) codo – 22kg: masa =180g “9805MG”
* 1 servo (rot) muñeca - 22 kg: masa =180g “9805MG”
* 1 servo (rot) giro muñeca 7kg: masa =55g “MG955H”
* 1 servo (prim) gripper 7kg: masa =55g “9805MG”

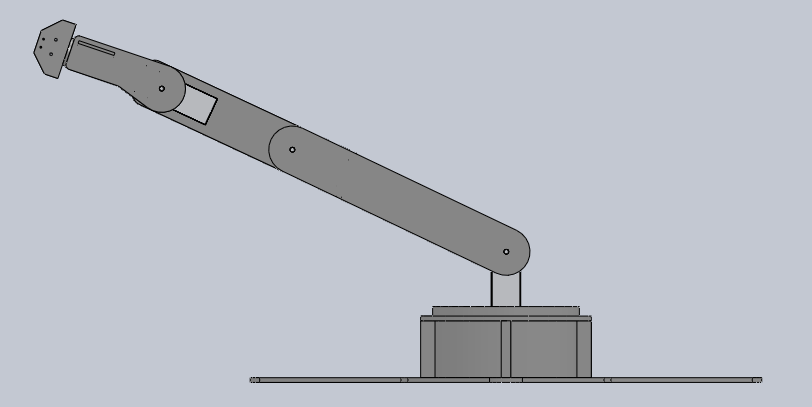
**Calculo de masas en cada Eslabón:**

Eslabón 1: masa =

Eslabón 2: masa =

Eslabón 3: masa =

**Calculo de Momentos:**



M1

M2

M3

25 cm

17 cm

8 cm

**MATERIALES UTILIZADOS**

Servomotores:

Son motores de corriente continua que tienen la propiedad de controlar su posición, es decir,

Habiéndola alcanzado primero para luego estabilizarla

Servomotor TowerPro MG995

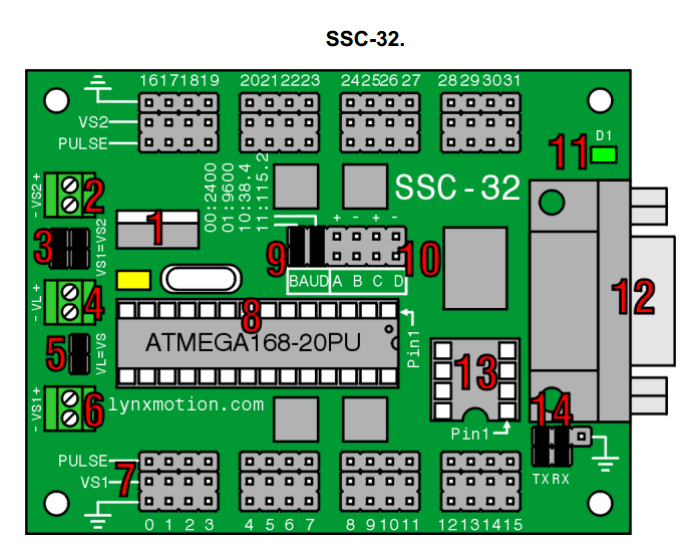
Dimensiones: 40.7 x 19.7 x 42.9

|  |  |
| --- | --- |
| Dimensiones | 40.7 |
| Peso | 38 gramos |
| Torque | 8.5 kg/cm |
| Voltaje de operación | 4.8 – 7.2 V |
| Velocidad de operación | 0.20s/60º |
| Temperatura de operación |  |
| Engranajes |  |

**Placa controladora SSC-32:**

Es una tarjeta provista de un microcontrolador Atmel (modelo ATMEGA168-20PU) que nos permitirá establecer de intermedio entre la interfaz de computadora y los servomotores. Mediante esta tarjeta es posible el control y sincronización de hasta un máximo de 32 servomotores, como se puede apreciar en la siguiente figura:

Descripción del hardware;



Se observan algunos componentes numerados, cuya función se procede a mencionar:

1. Regulador de tensión: Suministra una salida de 5 voltios en corriente continua a partir de una entrada de 5.5 voltios. Aunque es capaz de suministrar 500 mA, se recomienda un uso de 250 mA para evitar sobrecalentamientos.
2. Alimentación de los canales de los servos que van del orden 16 a 31. Aquí,
3. Jumpers de conexión de VS1 a VS2, cuando se requiera alimentar todos los servos con una sola batería.
4. Entrada de alimentación lógica para los componentes electrónicos del circuito. En la práctica, se comprueba que es importante separar las alimentaciones de los servomotores y de la lógica, debido a los posibles ruidos electromagnéticos.
5. Jumper de alimentación común del microcontrolador y los servomotores.

6. Terminal de alimentación de los canales de los servos (del 0 al 15), donde se sigue la misma convención de colores.

7.

8. Zócalo de ubicación del microcontrolador, donde se indica la correcta orientación que debe tener el mismo.

9. Cuatro jumpers que permiten graduar la tasa de baudios, tal como se ve en el siguiente ejemplo:

|  |  |  |
| --- | --- | --- |
| Jumpers | Tasa de baudios | Uso |
| 0 0 | 2400 | Procesadores de menor velocidad |
| 0 1 | 9600 | Procesadores de menor velocidad |
| 1 0 | 38400 | Comunicación Atom/Stamp |
| 1 1 | 115200 | Comunicación con el PC, actualización de firmware |

10. Entradas (A,B,C,D) para la entrada digital de lectura. Cuenta para este fin con resistencias de tipo pull-up de 50k.

11. LED que indica el correcto funcionamiento del procesador y la transmisión de datos.

12. Entrada para el conector de cable DB9

13. Zócalo para una memoria EEPROM de 8 pines, la cual debe ser compatible con el firmware 2.01 GP.

14. Este es el puerto a nivel TTL. Mediante dos jumpers (como se muestra) es posible habilitar tanto la comunicación serial vía DB9 o sino mediante un TTL con otro microcontrolador.

**Tipos y grupos de comandos:**

Segùn la hoja de datos, todos los comandos SSC-32 deben terminar en un carácter de retorno de carro.

Se

pueden ejecutar varios comandos del mismo tipo de forma simultánea en un *Grupo de comandos*. Todos los comandos

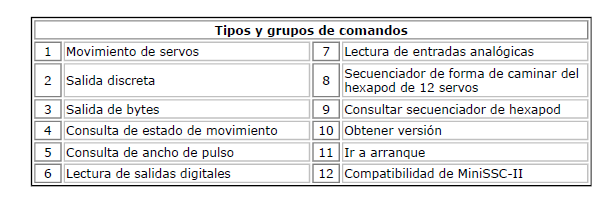
incluidos en un grupo de comandos se ejecutarán después de recibir el retorno de carro. Los comandos de tipos diferentes no

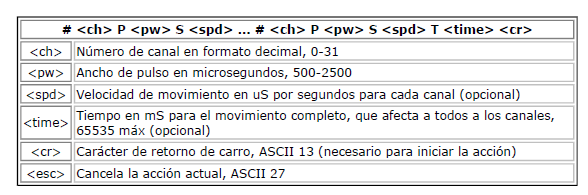
pueden combinarse dentro del mismo grupo de comandos. Además, los argumentos numéricos para todos los comandos SSC-

32 deben ser cadenas ASCII de los números decimales, por ejemplo, "1234". Algunos comandos admiten números negativos,

por ejemplo, "-5678". Se proporcionarán ejemplos de programación más adelante. El formato de ASCII no es sensible al uso de

mayúsculas/minúsculas. Use cuantos bytes sean necesarios. Se ignorarán los espacios, tabuladores y saltos de líneas.





Ejemplo de movimiento de servo: "#5 P1600 S750 <cr>"

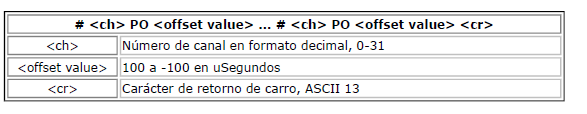
El ejemplo moverá el servo en el canal 5 a la posición 1600. Se moverá desde su posición actual a una velocidad de 750uS por

segundo hasta que alcance el destino del comando. Para comprender mejor el argumento de velocidad, tenga en cuenta que

un desplazamiento de 1000 uS equivale a un giro de 90° aproximadamente. Un valor de velocidad de 100 uS por segundo

significa que el servo tardará 10 segundos en girar 90°. Asimismo, un valor de velocidad de 2000 uS por segundo equivale a

500 mS (medio segundo) para girar 90°.

**Software de compensación de posición**

**CALCULOS PARA GENERAR EL MOVIMIENTO**

**SERVOS Y SU POSICION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SERVO | P-PLACA SSC32 | ANGULO | CAPACIDAD | UBICACION |
| S1 | P0 | Β1 | 15 Kg.cm | BASE |
| S2 | P1 | Β2 | 15 Kg.cm | HOMBRO |
| S3 | P2 | Β3 | 15 Kg.cm | HOMBRO |
| S4 | P3 | Β4 | 15 Kg.cm | CODO |
| S5 | P4 | Β5 | 15 Kg.cm | MUÑECA |
| S6 | P5 | Β6 | 5 Kg.cm | GRIPPER |

Conversion P – Angulo

Px=500+100\* Βx/9;

**ECUACIONES CINEMATICAS**

DENAVIT HATENBERG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gdl | Θ | d | a | Α |
| 1 | Θ1 | L1 | 0 | 90 |
| 2 | Θ2 | 0 | L2 | 0 |
| 3 | Θ3 | 0 | L3 | 0 |
| 4 | Θ4 | 0 | L4 | 0 |

L1=12;

L2=20;

L3=12;

L4=12;

Puntos del extremo del Robot

P0=[0,0,0];

P1=[0,0,L1];

P2=[L2\*cos(q2)\*cos(q1),L2\*cos(q2)\*sin(q1),L1+L2\*sin(q2)];

P3=[(L2\*cos(q2)+L3\*cos(q2+q3))\*cos(q1),(L2\*cos(q2)+L3\*cos(q2+q3))\*sin(q1),L1+L2\*sin(q2)+L3\*sin(q2+q3)];

P4=[(L2\*cos(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*cos(q1),(L2\*cos(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*sin(q1),L1+L2\*sin(q2)+L3\*sin(q2+q3)+L4\*sin(q2+q3+q4)]

* *La Matriz de Transformación se calcula a partir de :*

A partir de la fórmula matricial general:

* *Matriz de Jacobiano:*

a=-(L2\*sin(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*sin(q1);

b=-L2\*cos(q1)\*sin(q2)-L3\*cos(q1)\*sin(q2+q3)-L4\*cos(q1)\*sin(q2+q3+q4);

c=-L3\*cos(q1)\*sin(q1+q3)-L4\*cos(q1)\*sin(q2+q3+q4);

d=-L4\*cos(q1)\*sin(q2+q3+q4);

e=(L2\*sin(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*cos(q1);

f=-L2\*sin(q1)\*sin(q2)-L3\*sin(q1)\*sin(q2+q3)-L4\*sin(q1)\*sin(q2+q3+q4);

g=-L3\*sin(q1)\*sin(q1+q3)-L4\*sin(q1)\*sin(q2+q3+q4);

h=-L4\*sin(q1)\*sin(q2+q3+q4);

i=0;

j=L2\*cos(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4);

k=L3\*cos(q2+q3)+L4\*cos(q2+q3+q4);

l=L4\*cos(q2+q3+q4);

J=[a b c d;e f g h;i j k l]

**CINEMATICA INVERSA**

for i=1:6

q1=Q(1,1);

q2=Q(2,1);

q3=Q(3,1);

q4=Q(4,1);

Px=(L2\*cos(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*cos(q1);

Py=(L2\*cos(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*sin(q1);

Pz=L1+L2\*sin(q2)+L3\*sin(q2+q3)+L4\*sin(q2+q3+q4);

P=[Px;Py;Pz]

Po=Pi+(Pf-Pi)\*i/6

a=-(L2\*sin(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*sin(q1);

b=-L2\*cos(q1)\*sin(q2)-L3\*cos(q1)\*sin(q2+q3)-L4\*cos(q1)\*sin(q2+q3+q4);

c=-L3\*cos(q1)\*sin(q1+q3)-L4\*cos(q1)\*sin(q2+q3+q4);

d=-L4\*cos(q1)\*sin(q2+q3+q4);

e=(L2\*sin(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4))\*cos(q1);

f=-L2\*sin(q1)\*sin(q2)-L3\*sin(q1)\*sin(q2+q3)-L4\*sin(q1)\*sin(q2+q3+q4);

g=-L3\*sin(q1)\*sin(q1+q3)-L4\*sin(q1)\*sin(q2+q3+q4);

h=-L4\*sin(q1)\*sin(q2+q3+q4);

i=0;

j=L2\*cos(q2)+L3\*cos(q2+q3)+L4\*cos(q2+q3+q4);

k=L3\*cos(q2+q3)+L4\*cos(q2+q3+q4);

l=L4\*cos(q2+q3+q4);

J=[a b c d;e f g h;i j k l]

%matriz para la pseudoinversa

M=J'\*J

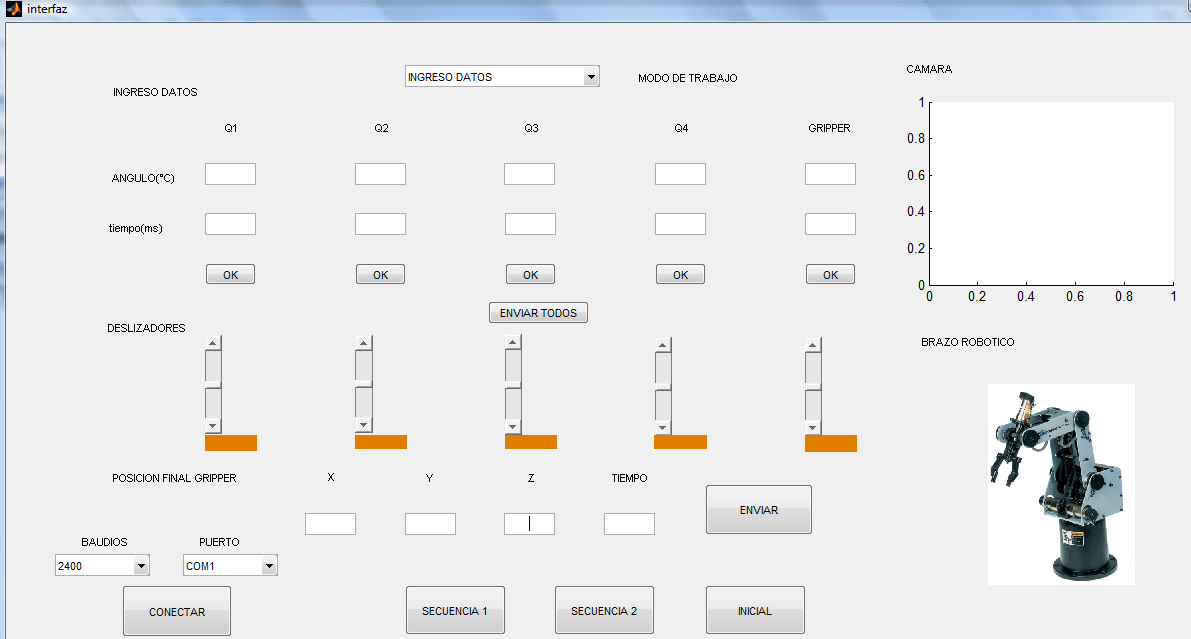
Qf=Q+inv(M)\*J'\*(Po-P)

Q=Qf;

end

**INTERFAZ PARA EL CONTROL DEL BRAZO ROBOTICO**

Esta interfaz, fue realizada en matlab como GUI.



Mediante esta interfaz se puede controlar al brazo robotico de distintos modos :

1. Ingreso de Datos :

Para controlar los angulos de las articulaciones individualmente o en grupo , ingresando la posicion y tiempo exacto .

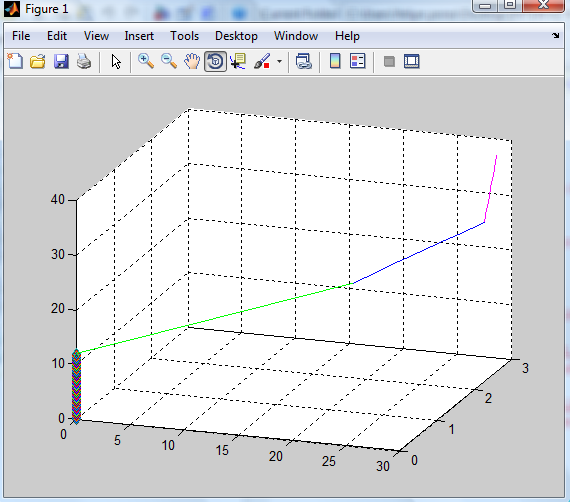
1. Deslizadores :

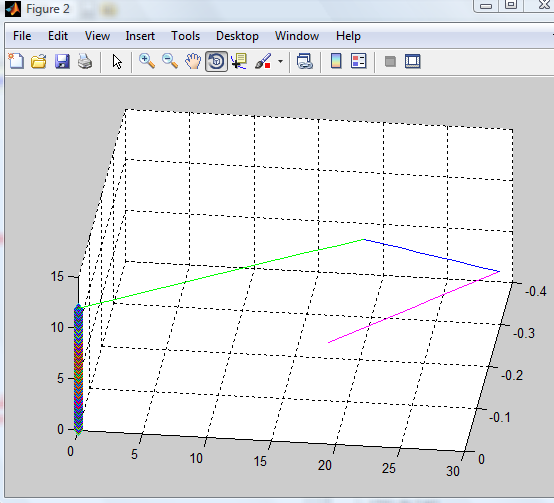
Se puede controlar cada angulo de articulación de manera continua , deslizando la barra.

1. Ingresando Posición Final :

Se ingresa las 3 coordenadas de la posición final deseada además del tiempo que demorara en llegar al objetivo.

Ademas entrega dos figuras que muestran la posición inicial del brazo y la posición final.





1. Secuencias :

Se realizan 2 secuencias distintas pre-establecidas .

**Código de las Secuencias**

a=1700;

fprintf(SerPIC,'%s',['#0P' int2str(500) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#1P' int2str(a) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#2P' int2str(3000-a) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#3P' int2str(1500) 'T' int2str(1500) 13]);

fprintf(SerPIC,'%s',['#4P' int2str(2000) 'T' int2str(1000) 13]);

fprintf(SerPIC,'%s',['#5P' int2str(1000) 'T' int2str(1000) 13]);

b=2050;

pause(5);

fprintf(SerPIC,'%s',['#0P' int2str(2000) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#1P' int2str(b) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#2P' int2str(3000-b) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#3P' int2str(1500) 'T' int2str(3000) 13]);

fprintf(SerPIC,'%s',['#4P' int2str(1500) 'T' int2str(1000) 13]);

pause(3);

fprintf(SerPIC,'%s',['#5P' int2str(1600) 'T' int2str(3000) 13]);

pause(5);

a=1500;

fprintf(SerPIC,'%s',['#0P' int2str(500) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#1P' int2str(a) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#2P' int2str(3000-a) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#3P' int2str(1700) 'T' int2str(3000) 13]);

fprintf(SerPIC,'%s',['#4P' int2str(2000) 'T' int2str(1000) 13]);

pause(3);

fprintf(SerPIC,'%s',['#5P' int2str(1400) 'T' int2str(3000) 13]);

pause(5);

a=1500;

fprintf(SerPIC,'%s',['#0P' int2str(500) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#1P' int2str(a) 'T' int2str(2000) 13]);

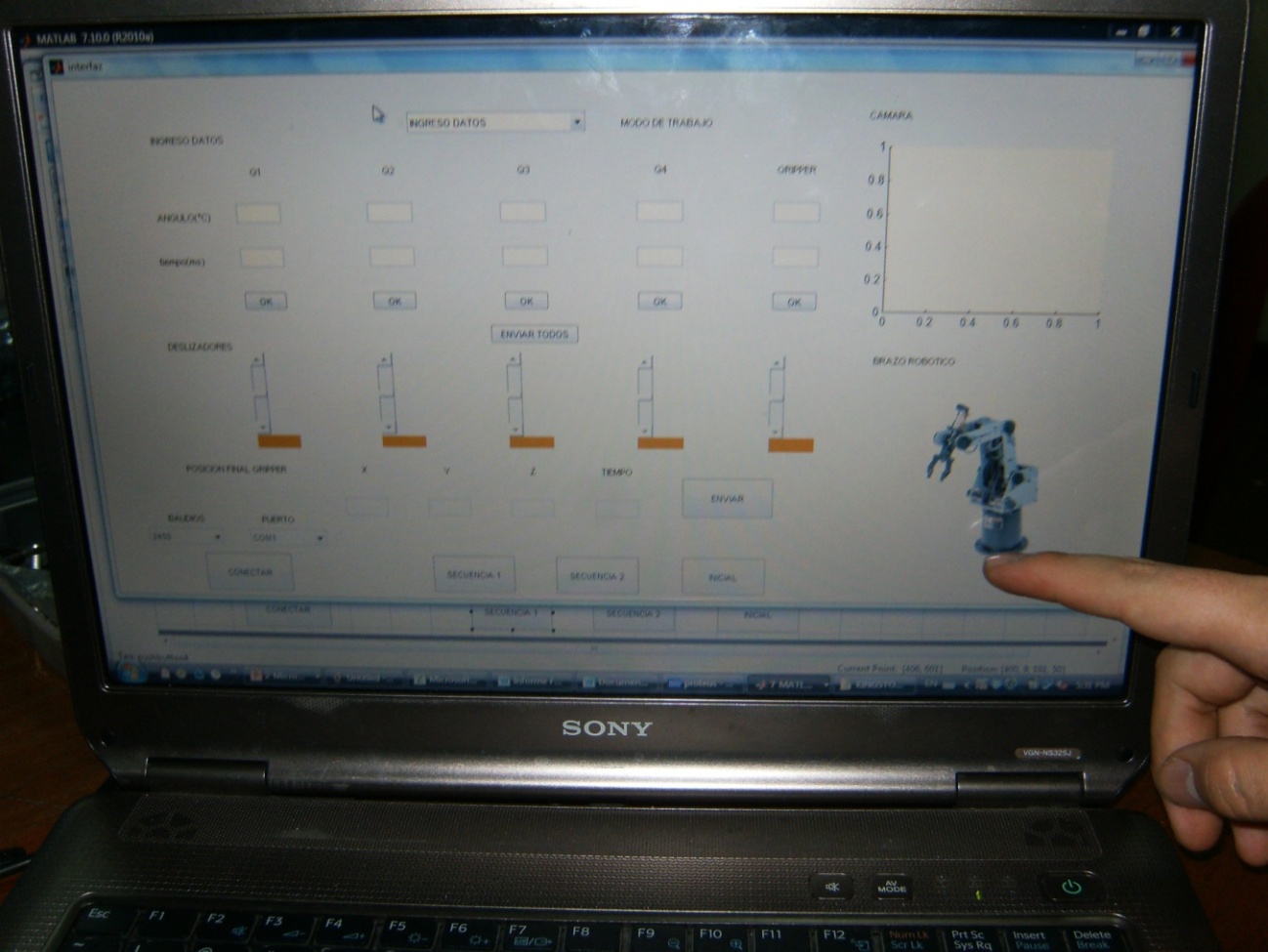
fprintf(SerPIC,'%s',['#2P' int2str(3000-a) 'T' int2str(2000) 13]);

fprintf(SerPIC,'%s',['#3P' int2str(1700) 'T' int2str(3000) 13]);

fprintf(SerPIC,'%s',['#4P' int2str(2000) 'T' int2str(1000) 13]);

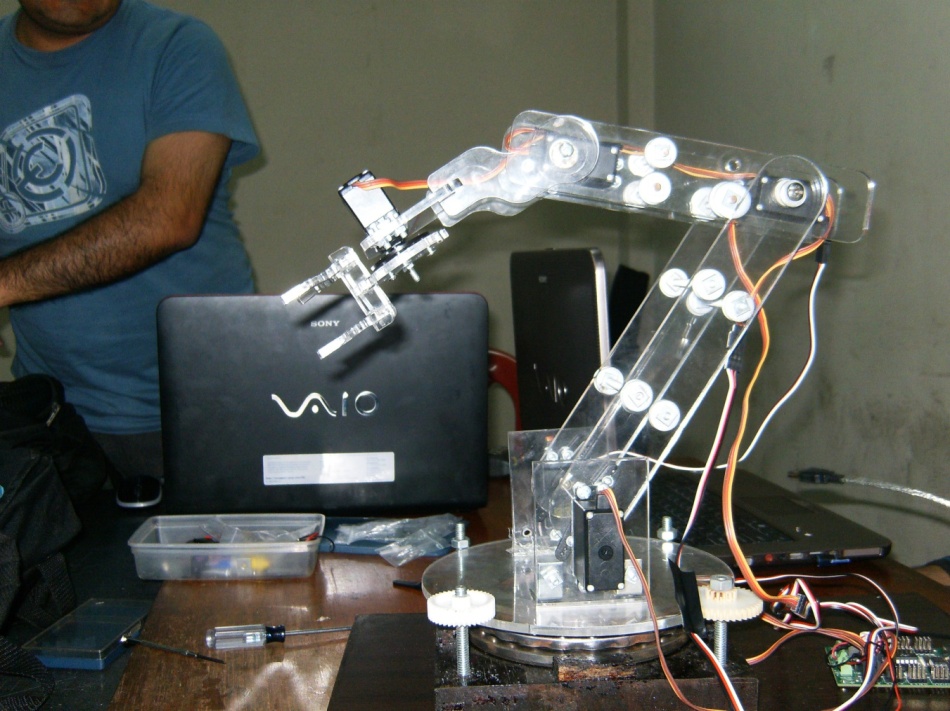
pause(3);

fprintf(SerPIC,'%s',['#5P' int2str(700) 'T' int2str(3000) 13]);

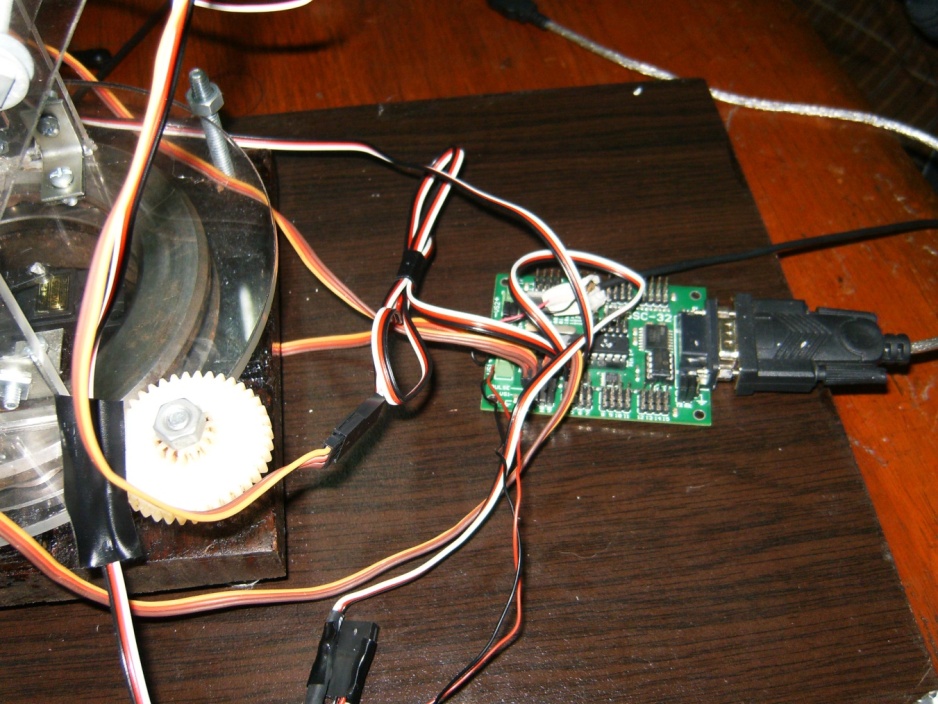


**IMPLEMENTACION**

**Controlando el brazo robotico**



**TARJETA CONTROLADORA DE SERVOS SSC32**



**FUENTE DE VOLTAJE UTILIZADA**

