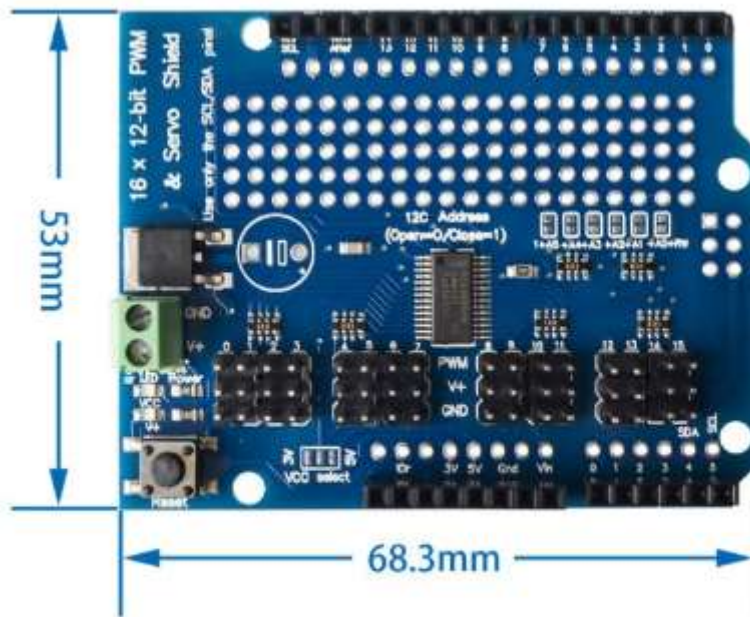


Brazo robot



Este brazo robótico lleva 5 servos incluidos. Con ellos podemos simular los movimientos del hombro (el de más abajo), el codo (el siguiente servo), la muñeca (dos servos, uno para el movimiento arriba-abajo y el otro para el movimiento derecha-izquierda) y el último servo para la pinza (o mano).

Material necesario:



Controlador para 16 servos, servo shield. Puedes comprarlo en Aliexpress (por unos 3€):

https://es.aliexpress.com/item/32823068267.html?spm=a2g0o.productlist.0.0.763a7bb2FC97YD&algo_pvid=667ea40b-12d9-4dcd-a8cb-b1a44b6d661f&algo_expid=667ea40b-12d9-4dcd-a8cb-b1a44b6d661f-2&btsid=c5e34248-3bf3-4d22-a9b7-f066a8eb764c&ws_ab_test=searchweb0_0,searchweb201602_10,searchweb201603_52



Alimentador de 5 voltios 5 amperios. Esto lo puedes comprar en cualquier lugar.



Chasis brazo robótico con servos. El de la imagen lleva 6 servos pero nosotros solo le montaremos 5 de ellos. Mira en la imagen al inicio de este documento y móntalo conforme está dicha imagen. Puedes comprarlo en Aliexpress (por unos 30€):

https://es.aliexpress.com/item/32953547342.html?src=google&src=google&albch=shopping&acnt=494-037-6276&isdl=y&slnk=&plac=&mtctp=&albbt=Google_7_shopping&aff_platform=google&aff_short_key=UneMJZvf&&albagn=888888&albcpr=2047572441&albag=80829465588&trgt=743612850714&crea=es32953547342&netw=u&device=c&gclid=CjwKCAiA5JnuBRA-EiwA-0ggPfEyONWU9I_r6nM4Kyp5E-TphwCZlmy6QrxRqUHI3RkyNp3DnRd6cRoC6qwQAvD_BwE&gclsrc=aw.ds

Y por supuesto un Martuino y algunos cables.

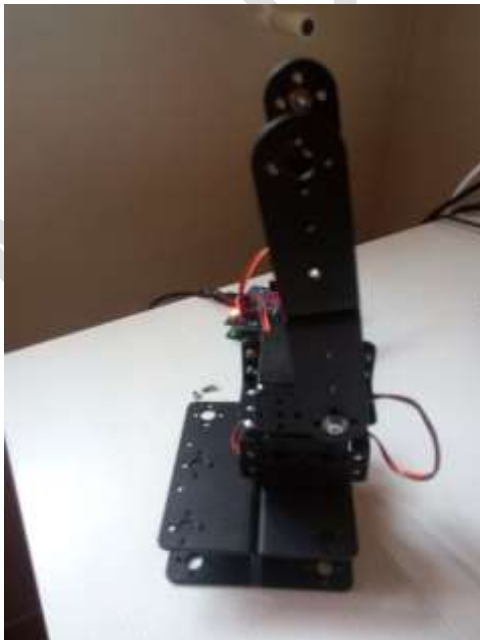
El Comienzo

Lo primero será montar el chasis del brazo robot siguiendo las instrucciones que lleva:

1. Montaremos el servo de más abajo:



2. Siguiendo servo y sus láminas de aluminio:



3. Siguiendo servo y sus láminas de aluminio:



4. Y la pinza:



Ahora conectaremos la tarjeta servos shield y los cables de los servos al Martuino:



Los servos irán conectados a la tarjeta servo shield del canal 0 al 4, siendo el 0 el servo de más abajo (el hombro) y así respectivamente para los servos conforme subimos en la estructura del brazo. El servo canal 4 será la pinza. La tarjeta shield la tenemos que alimentar con 5 voltios que además soporten una corriente alta, de al menos 5 amperios. Si esto no es así cuando funcionen varios servos a la vez podría tirar abajo la alimentación y reiniciar el Martuino. Aprovecharemos esta alimentación de 5 voltios para alimentar también al Martuino. Insertar la tarjeta servo shield sobre el Martuino y alimentaremos las dos tarjetas con una fuente de 5 voltios. Si el alimentador lleva conector redondo, lo cortaremos e insertaremos los cables directamente en la tarjeta servo shield y al Martuino por Vin (+) y GND (-). Ojo con la polaridad, asegúrate en el alimentador cual es el positivo y cual el negativo y conéctalo correctamente.

El programa

Os ponemos el código del programa para que funcione el brazo robótico. Hemos elegido unos comandos para mover cada servo y un ángulo que corresponderá con el trazado de cada servo. A través de una conexión wifi realizaremos el control del brazo. El control lo haremos desde un P.C. con un programa como el [TeraTerm](#), gratuito, o desde un móvil Android con un **Telnet Client** que encontraras en el Play store.

Todos los comandos que nos hemos inventado empiezan por una letra mayúscula:

- H Hombro (servo inferior del todo)
- C Codo (siguiente servo siguiendo hacia la pinza)
- M giro muñeca (siguiente servo siguiendo hacia la pinza)
- A Muñeca arriba-abajo (siguiente servo siguiendo hacia la pinza)
- P Pinza abre o cierra (servo de la pinza)
- D retardo
- R reset.

Seguido de tres números entre 000 y 180, correspondiente a los grados de giro, luego un - y tres números más correspondientes al retardo en cada grado a mover el servomotor (en miliseg.).

Ejemplo:

H090-020 Mueve el hombro 90 grados y en cada grado espera 20 milisegundos.

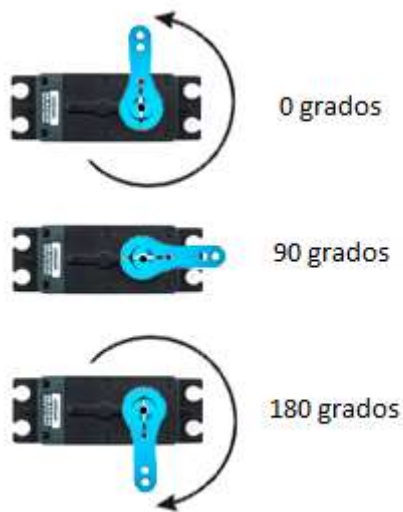
Si el retardo = 000 será el máximo retardo (1 segundo). Si el retardo es 001 será el mínimo posible (1 milisegundo).

El comando D se utiliza para hacer paradas después de cualquier movimiento. Va seguido de 3 números y también es en milisegundos.

D100 Para los movimientos durante 100 milisegundos.

El comando R hace que el brazo vuelva a su estado origen, el mismo que cuando se alimenta el Martuino.

En todos los servos que lleva el brazo, corresponden los 0 grados cuando están totalmente girados hacia un lado y 180 grados cuando lo están hacia el lado contrario.



Por lo tanto a la hora de posicionar cualquiera de las partes del brazo hay que tener en cuenta el ángulo que acompaña al comando. Podemos hacer pruebas con los ángulos para cada uno de los servos por separado.

Para la pinza solo será necesario enviar P000 para cerrarla y P001 para abrirla. No necesita tiempo.

En este caso Martuino se iniciará como un punto de acceso, es decir, que él crea su propio Wi-Fi como si de un router se tratara. Tendrás que conectarte a él, buscando un wi-fi llamado "BRAZOROBOT" y de contraseña "brazorobot".

Ahora veamos el código Martuino:

```
/*  
  
Brazo robotico  
  
Circuit:  
  
* Martuino  
  
creado Ago 2017 por Vicente Martinez  
  
*/  
  
#include <WiFi.h>  
  
#include <Wire.h>  
  
#include <Adafruit_PWMServoDriver.h>  
  
// your network name also called SSID  
char ssid[] = "BRAZOROBOT";  
// your network password  
char password[] = "brazorobot";  
Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver();  
WiFiServer server(23);  
  
boolean alreadyConnected = false; // whether or not the client was connected  
previously  
  
unsigned int pos0=110; // ancho de pulso en cuentas para pocion 0°  
unsigned int pos180=470; // ancho de pulso en cuentas para la pocion 180°  
unsigned int pos90=290;  
unsigned int manocerrada = 500;  
unsigned int manoabierta = 200;
```

```
unsigned int mano = 4;
unsigned int muneca = 2;
unsigned int munecag = 3;
unsigned int codo = 1;
unsigned int hombro = 0;
unsigned int angulohombro = 90;
unsigned int angulocodo = 90;
unsigned int angulomuneca = 90;
unsigned int angulomunecag = 90;
unsigned int angulopinza = manocerrada;
unsigned int angulohombroprox = 90;
unsigned int angulocodoprox = 90;
unsigned int angulomunecaprox = 90;
unsigned int angulomunecagprox = 90;
unsigned int angulopinzaprox = manocerrada;
unsigned int tiempohombro = 1;
unsigned int tiempocodo = 1;
unsigned int tiempomunecag = 1;
unsigned int tiempomuneca = 1;
unsigned int tiempopinza = 1;
char bufrec[50];
char mese[3];
int z = 0;
int indicebuf = 0;

void setup() {
```

```
//Initialize serial and wait for port to open:
Serial.begin(19200);
servos.begin();
servos.setPWMPFreq(56); // Analog servos run at ~56 Hz updates
servos.setPWM(hombro,0,pos90);
delay(1000);
servos.setPWM(codo,0,pos90);
delay(1000);
servos.setPWM(muneca,0,pos90);
delay(1000);
servos.setPWM(munecag,0,pos90);
delay(1000);
servos.setPWM(mano,0,manocerrada);
angulohombro = 90;
angulocodo = 90;
angulomuneca = 90;
angulomunecag = 90;
angulopinza = 0;
//Opción 1: Como Access Point AP
Serial.print("Arrancando wifi...");
WiFi.beginNetwork(ssid, password);
Serial.println("Hecho.");

// Arrancamos el servidor en el puerto 23:
server.begin();
}
```

```

void loop() {

  // wait for a new client:

  WiFiClient client = server.available();

  // when the client sends the first byte, say hello:

  if (client) {

    if (!alreadyConnected) {

      // clear out the input buffer:

      client.flush();

      Serial.println("NUEVO CLIENTE CONECTADO");

      alreadyConnected = true;

    }

    if (client.available() > 0) {

      // read the bytes incoming from the client:

      char thisChar = client.read();

      // echo the bytes back to the client:

      server.write(thisChar);

      bufrec[indicebuf] = thisChar;

      indicebuf++;

      if (thisChar == '\r'){

        for(int x=0;x<indicebuf;x++){

          if (bufrec[x] == 'H'){//Movimiento hombro

            mese[0] = bufrec[x+1];

            mese[1] = bufrec[x+2];

            mese[2] = bufrec[x+3];

          }

        }

      }

    }

  }

}

```

```

angulohombroprox = atoi(mese);

mese[0] = bufrec[x+5];

mese[1] = bufrec[x+6];

mese[2] = bufrec[x+7];

tiempohombro = atoi(mese);

client.println("");

client.print("H");

client.println(angulohombroprox);

actuarhombro(tiempohombro);

}

if (bufrec[x] == 'C'){//Movimiento codo

mese[0] = bufrec[x+1];

mese[1] = bufrec[x+2];

mese[2] = bufrec[x+3];

angulocodoprox = atoi(mese);

mese[0] = bufrec[x+5];

mese[1] = bufrec[x+6];

mese[2] = bufrec[x+7];

tiempocodo = atoi(mese);

client.println("");

client.print("C");

client.println(angulocodoprox);

actuarcodo(tiempocodo);

}

if (bufrec[x] == 'M'){//Movimiento muñeca giro

mese[0] = bufrec[x+1];

```

```
mese[1] = bufrec[x+2];
mese[2] = bufrec[x+3];
angulomunecagprox = atoi(mese);
mese[0] = bufrec[x+5];
mese[1] = bufrec[x+6];
mese[2] = bufrec[x+7];
tiempomunecag = atoi(mese);
client.println("");
client.print("M");
client.println(angulomunecagprox);
actuargiromuneca(tiempomunecag);
}
if (bufrec[x] == 'A'){//Movimiento muñeca arriba-abajo
mese[0] = bufrec[x+1];
mese[1] = bufrec[x+2];
mese[2] = bufrec[x+3];
angulomunecaprox = atoi(mese);
mese[0] = bufrec[x+5];
mese[1] = bufrec[x+6];
mese[2] = bufrec[x+7];
tiempomuneca = atoi(mese);
client.println("");
client.print("A");
client.println(angulomunecaprox);
actuarmuneca(tiempomuneca);
}
```

```
if (bufrec[x] == 'P'){//Movimiento pinza
    mese[0] = bufrec[x+1];
    mese[1] = bufrec[x+2];
    mese[2] = bufrec[x+3];
    angulopinzaprox = atoi(mese);
    mese[0] = bufrec[x+5];
    mese[1] = bufrec[x+6];
    mese[2] = bufrec[x+7];
    tiempopinza = atoi(mese);
    client.println("");
    client.print("P");
    client.println(angulopinzaprox);
    actuarpinza(tiempopinza);
}
if (bufrec[x] == '?'){//angulos
    client.println("");
    client.print("H");
    client.println(angulohombroprox);
    client.print("C");
    client.println(angulocodoprox);
    client.print("M");
    client.println(angulomunecagprox);
    client.print("A");
    client.println(angulomunecaprox);
    client.print("P");
    client.println(angulopinzaprox);
```

```
}  
  
if (bufrec[x] == 'R'){//Reset  
    resetservos();  
  
    client.println("");  
  
    client.print("H");  
  
    client.println(angulohombroprox);  
  
    client.print("C");  
  
    client.println(angulocodoprox);  
  
    client.print("M");  
  
    client.println(angulomunecaprox);  
  
    client.print("A");  
  
    client.println(angulomunecaprox);  
  
    client.print("P");  
  
    client.println(angulopinzaprox);  
}  
  
if (bufrec[x] == 'D'){//Delay  
    mese[0] = bufrec[x+1];  
    mese[1] = bufrec[x+2];  
    mese[2] = bufrec[x+3];  
  
    delay(atoi(mese));  
  
    client.println("");  
  
    client.print("D");  
  
    client.println(atoi(mese));  
}  
}  
  
indicebuf = 0;
```



```
}  
}  
}  
}
```

```
void printWifiStatus() {  
    // print the SSID of the network you're attached to:  
    Serial.print("SSID: ");  
    Serial.println(WiFi.SSID());  
  
    // print your WiFi shield's IP address:  
    IPAddress ip = WiFi.localIP();  
    Serial.print("IP: ");  
    Serial.println(ip);  
  
    // print the received signal strength:  
    long rssi = WiFi.RSSI();  
    Serial.print("Nivel de señal (RSSI):");  
    Serial.print(rssi);  
    Serial.println(" dBm");  
}
```

```
void actuarmuneca(int tiempo){  
    if (angulomunecaprox > 180){  
        angulomunecaprox = 180;  
    }  
}
```

```

if (angulomunecaprox<0){
    angulomunecaprox = 0;
}
if (angulomunecaprox>angulomuneca){
    for (int y=angulomuneca;y<angulomunecaprox;y++){
        z = (y * 2)+ pos0;
        servos.setPWM(muneca,0,z);
        delay(tiempo);
    }
}else{
    while (angulomunecaprox<angulomuneca){
        z = (angulomuneca * 2)+ pos0;
        servos.setPWM(muneca,0,z);
        delay(tiempo);
        angulomuneca--;
    }
}
angulomuneca = angulomunecaprox;
}
void actuarhombro(int tiempo){
if (angulohombroprox > 180){
    angulohombroprox = 180;
}
if (angulohombroprox<0){
    angulohombroprox = 0;
}
}

```

```

if (angulohombroprox > angulohombro){
  for (int y=angulohombro;y<angulohombroprox;y++){
    z = (y * 2)+ pos0;
    servos.setPWM(hombro,0,z);
    delay(tiempo);
  }
}else{
  while (angulohombroprox<angulohombro){
    z = (angulohombro * 2)+ pos0;
    servos.setPWM(hombro,0,z);
    delay(tiempo);
    angulohombro--;
  }
}
angulohombro = angulohombroprox;
}

void actuargiromuneca(int tiempo){
  if (angulomunecagprox > 180){
    angulomunecagprox = 180;
  }
  if (angulomunecagprox<0){
    angulomunecagprox = 0;
  }
  if (angulomunecagprox > angulomunecag){
    for (int y=angulomunecag;y<angulomunecagprox;y++){
      z = (y * 2)+ pos0;

```

```

servos.setPWM(munecag,0,z);
delay(tiempo);
}
}else{
while (angulomunecagprox<angulomunecag){
z = (angulomunecag * 2)+ pos0;
servos.setPWM(munecag,0,z);
delay(tiempo);
angulomunecag--;
}
}
angulomunecag = angulomunecagprox;
}
void actuarcodigo(int tiempo){
if (angulocodoprox > 180){
angulocodoprox = 180;
}
if (angulocodoprox<0){
angulocodoprox = 0;
}
if (angulocodoprox > angulocodo){
for (int y=angulocodo;y<angulocodoprox;y++){
z = (y * 2)+ pos0;
servos.setPWM(codo,0,z);
delay(tiempo);
}
}
}

```

```

}else{
    while (angulocodoprox<angulocodo){
        z = (angulocodo * 2)+ pos0;
        servos.setPWM(codo,0,z);
        delay(tiempo);
        angulocodo--;
    }
}
angulocodo = angulocodoprox;
}

void resetservos(){
    angulohombro = 90;
    angulocodo = 90;
    angulomuneca = 90;
    angulomunecag = 90;
    angulopinza = 0;

    servos.setPWM(hombro,0,pos90);
    delay(250);
    servos.setPWM(codo,0,pos90);
    delay(250);
    servos.setPWM(muneca,0,pos90);
    delay(250);
    servos.setPWM(munecag,0,pos90);
    delay(250);
    servos.setPWM(mano,0,manocerrada);
}

```

```
}  
void actuarpinza(int tiempo){  
  if (angulopinzaaprox > 180){  
    angulopinzaaprox = 180;  
  }  
  if (angulopinzaaprox<0){  
    angulopinzaaprox = 0;  
  }  
  if (angulopinzaaprox == 0){  
    servos.setPWM(mano,0,manocerrada);  
  }else{  
    servos.setPWM(mano,0,manoabierta);  
  }  
  angulopinza = angulopinzaaprox;  
}
```

Este es todo el programa necesario.

Una vez grabado desde Energia-Martuino podremos desconectar el usb y cambiar el jumper de la placa Martuino Vusb a Vin:



Posición normal para grabar por el USB a Martuino.

Para que se alimente desde la alimentación de 5 voltios debemos pasarlo al otro lado.

Al alimentar el brazo este se posiciona en un lugar que está memorizado en el código en el void setup():

```
servos.setPWM(hombro,0,pos90);
```

```
servos.setPWM(codo,0,pos90);
```

```
servos.setPWM(muneca,0,pos90);
```

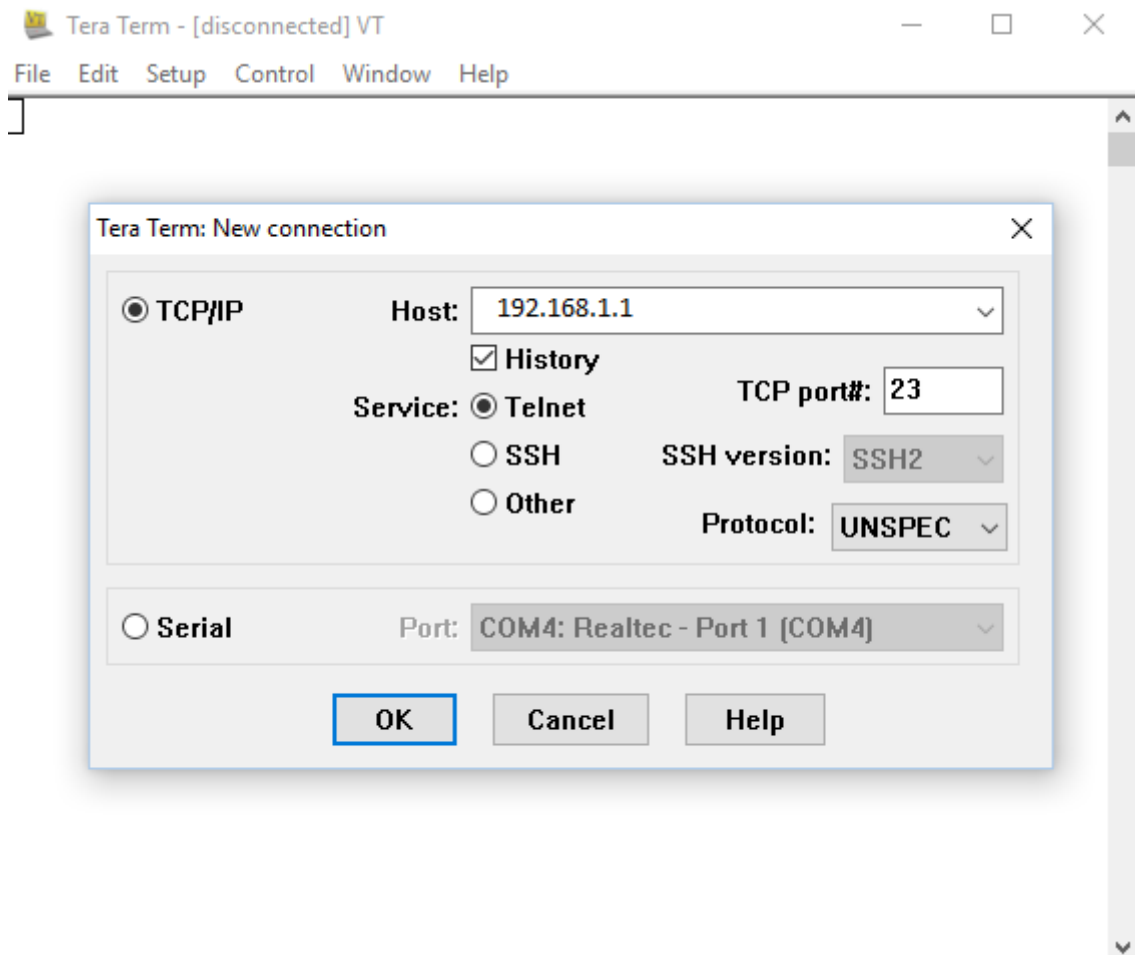
```
servos.setPWM(munecag,0,pos90);
```

```
servos.setPWM(mano,0,manocerrada);
```

Más o menos ponemos todos los servos en su posición central. Si quieres que cuando arranque alguna otra posición, solo tienes que cambiar estas líneas de código.

El control

Una vez en marcha todo y conectados al wi-fi del brazo, abriremos el programa TeraTerm y tendremos esta pantalla:



Pondremos la I.P. 192.168.1.1 y pincharemos sobre Telnet y le daremos a OK.

El programa TeraTerm viene configurado con el fondo negro y letras blancas, si quieres que te aparezca como en la imagen entra en Setup → Window... y pulsa sobre reverse. Después podrás ajustar el tamaño de la letra en Setup → Font.

Una vez conectado sobre la pantalla de Teraterm pulsa en Mayúsculas los comandos indicados y verás moverse el brazo. Realiza pruebas con cada servo para familiarizarte con los ángulos.

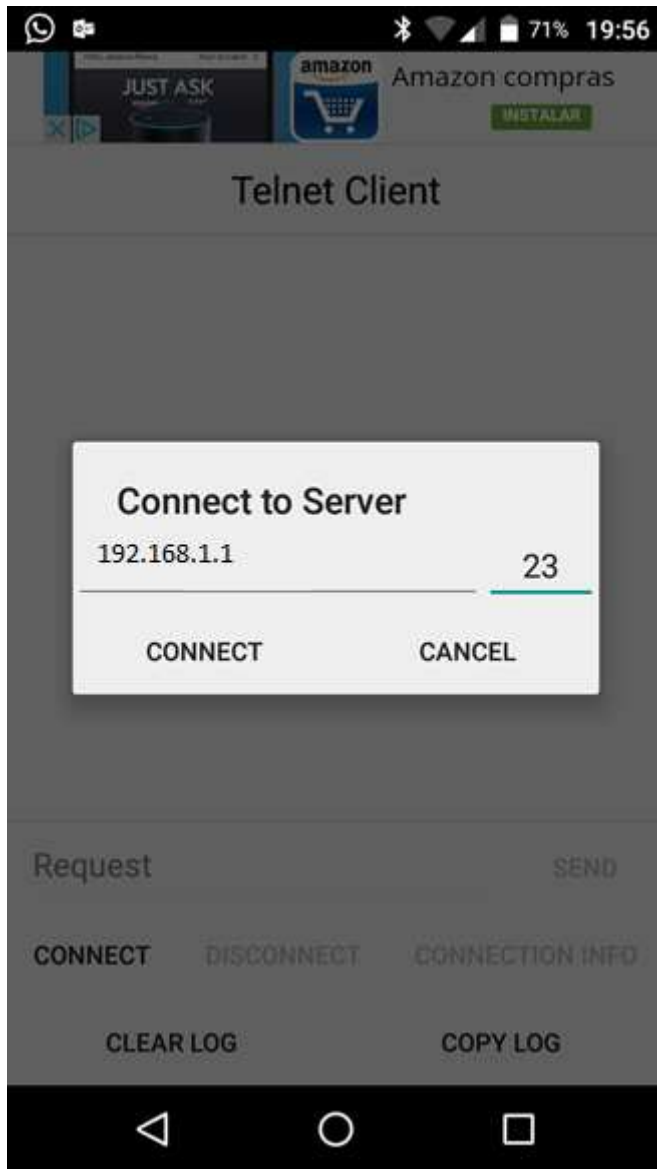
Con Teraterm también puedes enviar un archivo de texto con varias maniobras juntas. Por ejemplo guarda este texto en un archivo txt:

```
M010-020
D500
P001
D500
H120-020
D500
C010-020
D500
A060-020
D500
P000
D500
C040-020
D500
H090-020
D500
C000-020
D500
P001
D500
C090-020
D500
M090-020
D500
A090-020
D500
R
```

Este archivo envía al robot a una posición desde su base, recoge una goma de borrar y la lleva a otro lugar de la base del brazo. Nos referimos a base como el plano sobre el que se asienta el brazo. Está pensado el trayecto del brazo desde su posición inicial, cuando arranca al alimentarlo. Si no está en esta posición porque has movido algún servo haciendo pruebas, envía el comando R para posicionarlo en su estado inicial antes de enviar este archivo.

Ahora en Teraterm estando conectado al brazo, ves al menú File -> Send file y selecciona el archivo creado con anterioridad. Verás los movimientos que realiza el brazo automáticamente.

Desde Android podrás descargarte el programa telnet Client. Una vez instalado lo abres y pones la I.P. 192.168.1.1 y puerto 23 y le das a connect:



Y una vez conectado deberás escribir un comando y “send” para enviárselo al brazo robótico. Envía un comando con cada “send”.

Disfruta de este brazo robótico.