

# Diseño y construcción de un brazo robot

## Sección 1 La historia

Capítulo 1: Preámbulo, antecedentes e historia

## Sección 2: El montaje

Capítulo 1: Un brazo fuerte

Capítulo 2: Ensamblaje del brazo

Capítulo 3: Para asir

Capítulo 4: Conexiones

Capítulo 5: A prudente distancia

## Sección 3: Circuito de control y software

Capítulo 1: Primeras ideas

Capítulo 2: Purificando el soft

Capítulo 3: La placa final (por el momento)

## Sección 1 La historia

### Capítulo 1: Los orígenes

Todos los que nos hemos metido en este mundillo de la robótica hemos tenido que empezar y para ello hemos buscado un diseño sencillo, que fuéramos capaces de construir con pocos conocimientos, que fuera económico y sencillo.

Esto, mas que un documento “tutorial” es la historia, de cómo monté mi primer (y hasta ahora único) elemento robótico.

Así que os pido una pequeña concesión, antes de seguir directamente a los planos, dejadme explicaros una pequeña historia....

Nací en 1961. En esa época no había ni robots ni ordenadores ni nada de nada. La tele y poco más. Todo eso era cosa, como mucho, de ciencia ficción.

Jajajaja ¿sabéis? Yo invente el PC. Bueno mas o menos (que mas quisiera) .

Cuando tenia 7 u 8 años (no me acuerdo con exactitud) un familiar tenia una academia (que no colegio, al menos en la concepción actual) y tenia un aula con maquinas de escribir (mecánicas... prodigio de la técnica). Me dejaban sentarme y “jugar” con ellas y yo pensaba... joer... si has de picar 100 veces lo mismo... ¿Por qué no inventar “algo” (pensaba en algo para insertar lateralmente...que casual) que guarde esas pulsaciones para posteriormente que el “parato” lo haga solo”

Peasso de crac. Visionario que era yo, jejejeje... si hubiera podido aprovecharlo ahora en vez de hablar de Roberto Puertas se hablaría de Jose Luis Dominguez. Si es que siempre he sido un científico aficionado.

Bueno, eso solo era una anécdota para haceros ver que en mi caso el tema de la técnica me viene de lejos.

Con 13-14 años devoraba (que no leía) revistas-novelas-películas de ciencia ficción... computadoras, robots.. etc, todo valía para “satisfacer” mi imaginación.

Quise estudiar, por supuesto, electrónica pero no quedaban plazas así que empecé mi segunda opción, delineante. Una vez metido en eso ya no cambie y de alguna manera me aleje un poco del tema tecnológico.

Si embargo continuaba mi interés por estos temas e incluso, en plan “amateur” efectuaba algunos montajes sencillos que me pasaban amigos.

Pero la realidad te empuja, así que me eche novia, me case, me busque una profesión, funde una familia... vamos lo habitual...

Y un día, con 23 años, allá por 1983 mis sueños se hicieron realidad... aparecieron unas extrañas maquinas que (a decir de la sociedad de la época nos iban a dejar a todos en el paro).

Lo tenia olvidado durante años (la “responsabilidad” ya se sabe) pero aquello fue la chispa que hizo renacer aquellas fantasías de adolescente... no eran solo historias, películas... eran REALIDAD.. los ordenadores “EXISTÍAN”.

Inmediatamente intente adquirir uno... imposible... con la economía de postguerra de toda familia recién formada y unos sueldos “justitos”, pensar en comprar un ordenador que valía (entonces) el sueldo de un año era una utopía.

Leía todo lo que salía (y podía adquirir) revistas de ordenadores, de electrónica, de robótica, etc...

Con el tiempo me prestaron un Spectrum... conseguí comprar un MSX... después lo cambié todo por un 8088... y me “subí” en el mundo PC (aproximadamente en 1988.. la época de los “pioneros”). Como anécdota, sentaba entonces a mi hija de 3 meses al ordenador para que aporreara las teclas. Ahora tiene 19 y ooooh casualidades... estudia informática... ¿Por qué será? Si yo no creo haberla influido para nada (te lo juro por Snoopy)... solo me pidió un ordenador ya con 10 años y le dije “no hay problema... móntatelo”... le di las piezas y lo logro, un 486.

Bueno, volvamos al tema... en aquella época aparecían mil publicaciones semanales y mensuales de la novedad (que era) la informática. Revistas, secciones de los diarios, enciclopedias, etc.

En esto que un familiar se empezó a hacer una enciclopedia del tema (Mi Computer, 10 tomos, editorial Orbis, 1983). Yo no podía pagármela, así que cada semana le pedía prestado el fascículo en cuestión para leerlo... Spectrum... Comodore 64, Z80, Basic, Logo... etc... el futuro al fin y al cabo.

Esa enciclopedia para mi que era una traducción de una publicación inglesa (hablaban sobretodo de un trasto llamado BBC Micro que JAMÁS he visto) y todo lo que decía, por tanto, estaba a años luz de España.

En aquella enciclopedia empezaron una sección de bricolaje... circuitos básicos, herramientas, etc...

Y luego un momento (me quede impresionado) en que presentaban unas "cosas" de Lego (en España entonces Lego solo eran 4 piezas de un puzzle y poco mas) que eran un robot que se podía programar desde un ordenador... increíble...

Como eso solo se podía (que yo sepa) comprar fuera, automáticamente paso al "cajón mental" de las utopías y la vida siguió.

Pero en el capítulo siguiente apareció un brazo robot "en serio"... y te lo explicaban así... tan sencillo... (Con un "móntatelo tu mismo como el actual Bricomania)... inconcebible...

Pero estudie los diseños y joer.. Eran viables... venia todo... planos, circuitos, software... así que me puse manos a la obra y empecé a pensar en montarlo. Las piezas principales eran, por supuesto, 4 servos, así que me fui a una tienda de modelismo y pregunte ¿Cuánto vale un servo?

El dependiente me contesto "uno sencillito entre 6.000 y 10.000 pesetas". Joer ya estamos otra vez con el tema de la "economía de guerra"... Necesitaba 4 y uno era el 20% de mi sueldo... gastarme esa cantidad (llegábamos justos a fin de mes) implicaba que la parienta me podría ajusticiar directamente sin juicio previo...

Así que como tantas otras cosa paso al cajón de "utopías... verificar de aquí a cien años" y el tema quedo aparcado.

Muchos años después y tras una especie de crisis de identidad necesite anímicamente "hacer algo... montar algo"... y me acorde de aquel bracito de hacia 20 años.

Volví a consultar (ya en internet) el precio de los famosos "motorcitos" y mira que cosas.. valian 10€... joer que chollo... mi casa ya esta aposentada, ya no hay economía de guerra... puedo pagar eso sin problemas... PUEDO montar ese brazo...

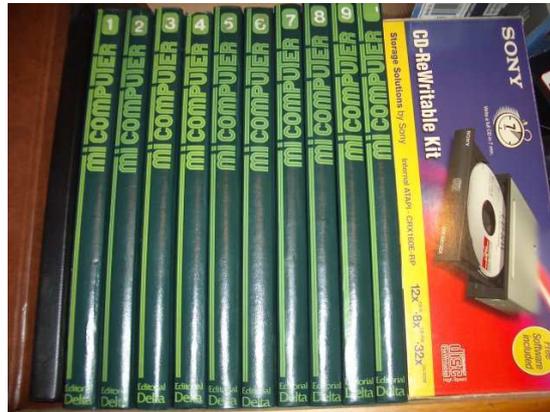
Pero las historias felices solo pasan en las películas de Disney... Busque las fotocopias que había echo... y habían desaparecido... parece ser que en una de las ultimas limpiezas familiares aquellos "papeles-añejos-y-amarronaos-que-no-sirven-para-nada" habían ido a parar al reciclaje.

Sin mucha esperanza pregunte al familiar que había sido de aquella antigua enciclopedia y me dijeron que no estaba... que la habían tirado... ¿o quizás se la llevo uno de sus hermanos cuando se casó? ... los hermanos (ambos actualmente informáticos) no se acordaban de nada...o quizás estaba en el altillo de casa de sus padres.. no sabían...

Así que armado de valor y como última opción fui a casa de sus padres para revisar el susodicho altillo... y BINGO... allí estaba... esperándome a MÍ desde hacia muchos, muchos años... la enciclopedia con las secciones de bricolaje que yo

quería.

Vista actual de esa enciclopedia, hábilmente “requisada” del atillo por aquí el narrador de la historia, que reposa actualmente en MI estantería.



Que experiencia... volver a tener aquello en mis manos... repasar pagina a pagina lo que decía.. lo que había cambiado el mundo esos años... ver las fotos... coño... si este es el Billy Gates con granos... y este el Steve Jobs con pelo... (si las queréis ver las escaneo, la enciclopedia en cuestión es un autentico incunable)

Y aquí están... los planos del brazo robot... pero a ver... el código... es de Basic “propietario”... los circuitos son añejos (si estos ordenadores no tenían ni puertos Lpt ni Com...) aquí hay mucho trabajo para hacer...

Sin embargo hay una buena información de lo que es un servo... de lo que requiere... seguro que esta en internet... y si añadido al circuito XX un controlador YY y busco en internet un esquema del LPT y le pongo y le quito.....y le pongo y le quito.... Y así, tras más de 20 años, nació MI PROYECTO.

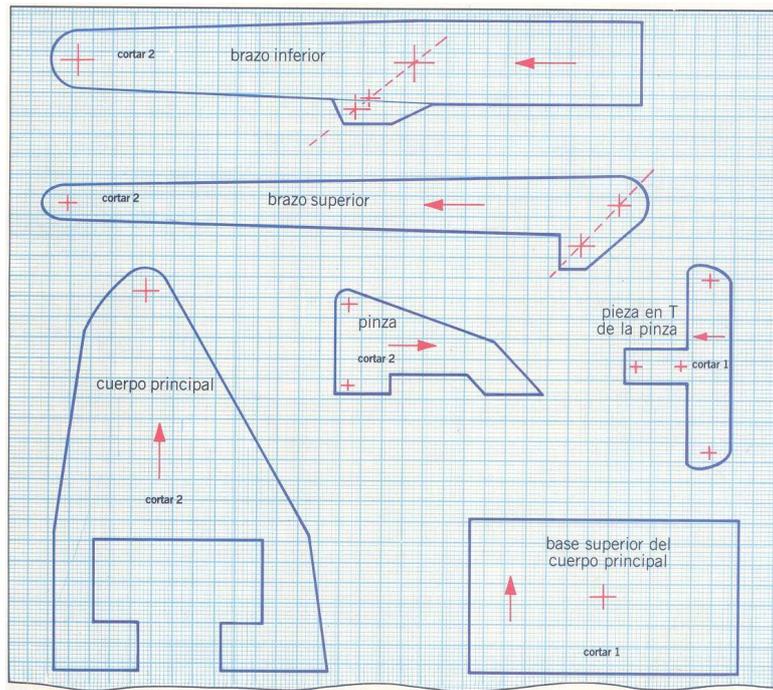
### **Sección 1: El diseño (Extracto de la enciclopedia “Mi computer, editorial Orbis, 1983”)**

*Dada la antigüedad del artículo original, totalmente obsoleto, para una mejor comprensión de este artículo se especifica el original en su totalidad en azul, indicando mediante textos en cursiva y color verde los cambios realizados para su adaptación a 2005, fecha en que comencé el montaje.*

## Sección 2: El montaje

### Capítulo 1: Un brazo fuerte

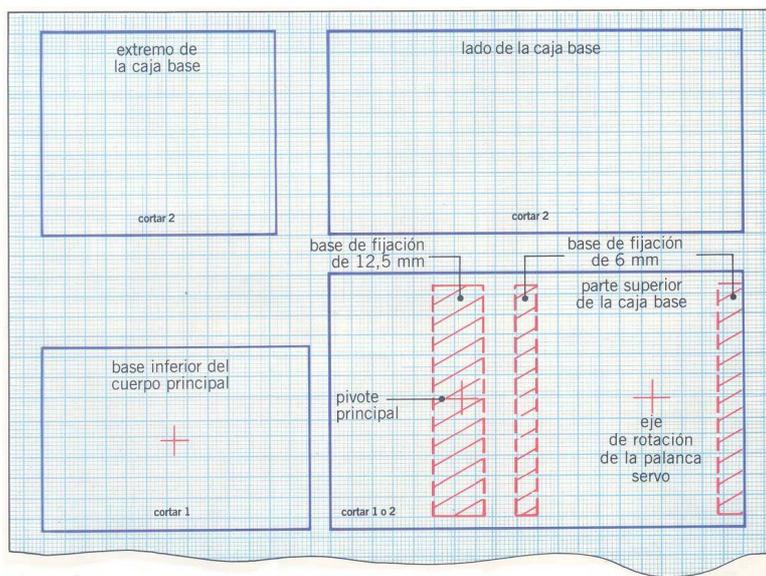
Iniciamos nuestro proyecto proporcionando los patrones de las partes del brazo y la lista de componentes



1574

Los componentes necesarios para este proyecto probablemente habrán de obtenerse de diversas fuentes. Las tiendas de electrónica y de modelos a escala proporcionarán los componentes electrónicos y móviles, pudiéndose adquirir el resto en ferreterías.

La base y los miembros del brazo se deben construir con un material ligero y rígido que se pueda cortar y moldear fácilmente. Probablemente el material más accesible sea la madera contrachapada, aunque también se podría utilizar una lámina delgada de perspex o un material plástico rígido. Suponiendo que para la estructura se emplee madera contrachapada, necesitaremos un cuadrado de 50 x 50 cm, de 4 mm de espesor. Lo usaremos junto con unos patrones para cortar los componentes de la estructura. Para actuar como bases de fijación necesitaremos tres trozos de madera de 55 mm (dos de 6 mm de espesor y uno de 12,5 mm de espesor).



<p><i>Haciendo un trabajo de ingeniería inversa, milimetrando el diseño, pase todo esto a CAD imprimible en dos Din A4. Se adjunta el plano de todas las piezas en archivos DWG y PDF.</i></p> <p><i>Para traspasarlos a la madera, yo simplemente los imprimí, los puse encima de del tablero y con una vulgar aguja traspase todos y cada uno de los vértices.</i></p> <p><i>Después, con ayuda de una regla y apoyándome en los agujeritos dibuje con un lápiz las diferentes piezas.</i></p> <p><i>Lo recorte todo con una sierra para marquetería. Para "igualar" las piezas usé lima y papel de lija.</i></p> <p><i>En el caso de las piezas "críticas" es decir la base y los brazos, para que las piezas me quedaran especialmente "gemelas" primero las recorte con un cierto margen y después realice los taladros para los ejes. Posteriormente uní las piezas con un tornillo y tuerca de M3 y las lijé y pulí unidas. Así me quedaron perfectamente iguales.</i></p> <p><i>El soporte y la base inferior la pegué con cola de carpintero hasta conseguir esto (se aprecia en la parte posterior de la base un orificio "inútil" pero que sirvió en su momento para unir las piezas con el susodicho tornillo):</i></p> 	<p>Cortado de los patrones Haga calcos de los patrones de la estructura, fíjelos sobre el tablero de contrachapado y córtelos. Observe que algunos patrones se usan dos veces para producir pare de componentes. Empareje los pares de los brazos y únalos antes de taladrarlos. Lije con cuidado cada uno de los componentes para eliminar las asperezas de los bordes.</p>
<p>Para el brazo necesitaremos cuatro servos digitales de 5 V; con este fin son ideales los motores Futaba S128. En el mismo sitio donde se adquieran los motores, se deben comprar discos plásticos de 30 mm para colocar en el extremo de los husillos del motor. El mejor sitio tal vez sean las tiendas de modelos a escala. Si desea un brazo más potente, entonces habrá de emplear motores más potentes que el Futaba S128.</p> <p>Para el motor necesitamos una fuente de alimentación eléctrica CD de 5 V. Aunque los tres micros que utilizaremos para controlar el brazo poseen fuentes de alimentación de 5 V</p>	<p>circuitos electrónicos a niveles TTL. La corriente máxima que se debe tomar de estas fuentes de 5 V es de 100 mA. Dado que un servomotor en carga máxima tomara 200 mA, no es aconsejable alimentar los servos del brazo desde la puerta de usuario. Por consiguiente, necesitamos una fuente de alimentación externa. Lo ideal sería un transformador de CD de 5V, pero una alternativa menos onerosa es utilizar un conjunto unido de tres pilas de 1,5 V. La salida total de 4,5 V será suficiente para activar el sistema.</p>

en la puerta para ampliación o para el usuario, estos voltajes están diseñados para alimentar

Se necesita un trozo de verovoard de 9 agujeros por 20 franjas para montar conectores en los que enchufar los motores, y dos metros de cable plano de 20 vías para conectar el brazo al ordenador. Quienes posean un BBC Micro habrán de comprar un conector IDC de 20 vías, y los que posean un Commodore 64 comprarán un conector marginal de 0,15 pulgadas y 24 vías para conectar el brazo a la puerta de usuario. El brazo se unirá al Spectrum a través de la interface que hemos construido en Bricolaje para controlar el robot. En un próximo capítulo explicaremos en forma detallada las modificaciones necesarias en esta interface.

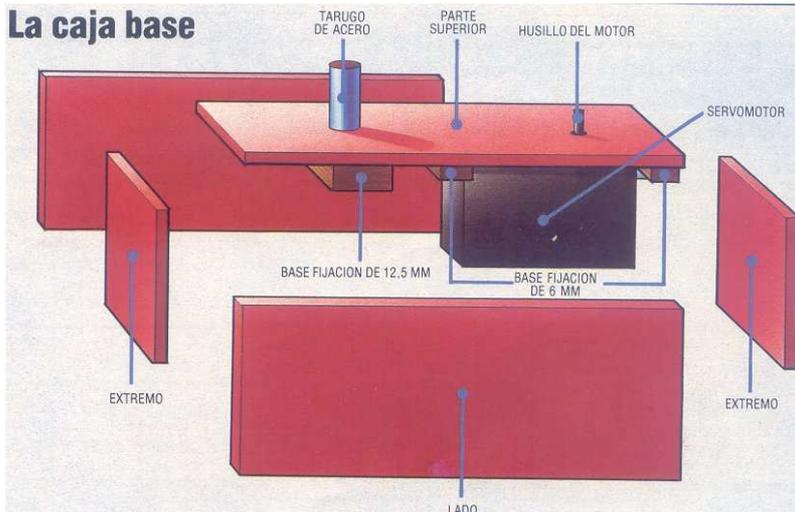
Para hacer los pivotes del brazo se requieren tres tubos de cobre de 75 mm de largo con un diámetro exterior de 4 mm, junto con tres trozos de tubo de cobre de 75 mm de largo y 5 mm de diámetro. Cada pivote se construirá colocando el tubo de 4 mm dentro del de 5 mm y recortando los extremos a la misma longitud. Por lo tanto, es importante que los tubos que compre se puedan colocar firmemente uno dentro de otro pero con suficiente amplitud como para que el tubo interior pueda girar libremente.

El montaje principal del brazo se une a la caja base mediante un cojinete de bolas.

El cojinete debe estar rebordeado y tener un diámetro interno de 6 mm. Estos cojinetes a menudo se utilizan para las ruedas de los modelos a escala y se pueden conseguir en tiendas de aeromodelismo. Para alargar el apoyo, también necesitaremos 25 mm de tarugo de acero de 6 mm. Este tarugo se fijara firmemente a la base y se encajara ajustadamente a través del cojinete.

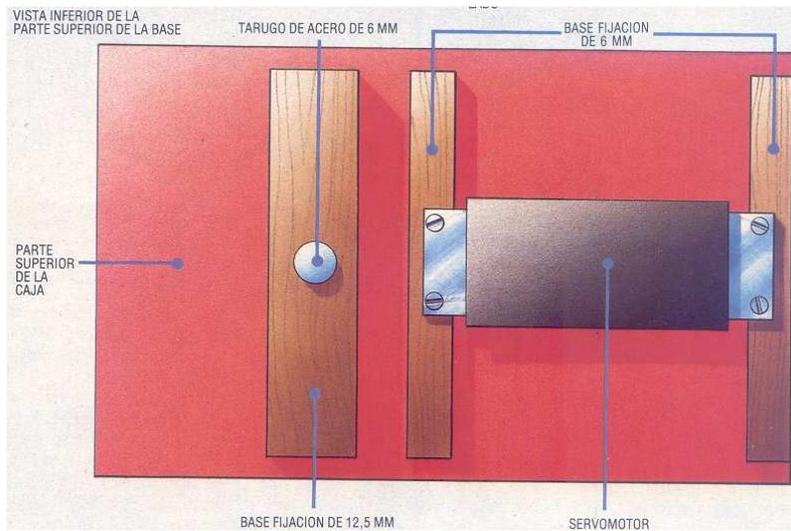
La pinza se opera desde un motor montado mas abajo en el ensamblaje del brazo. El motor debe conectarse a la pinza utilizando un cable flexible de 500 mm de largo (denominado a veces culebra). En esencia, este es un alambre de acero dentro de una manga de plástico. Los aviones a escala con frecuencia emplean este tipo de enlace para los alerones. Se necesitan 50 mm de alambre duro (el cable de piano es ideal) para abrir y cerrar la pinza, y este debe atarse al extremo de la culebra con un hilo de fusión de 5 A. Los miembros del brazo se mueven mediante un sistema de varillas. Para ello se utilizaran cuatro varillas de acero de 2 mm de grosor y 150 mm de longitud. Las varillas se pueden conectar a los discos del motor ya sea directamente o bien con pequeños cojinetes de bola.

### La caja base

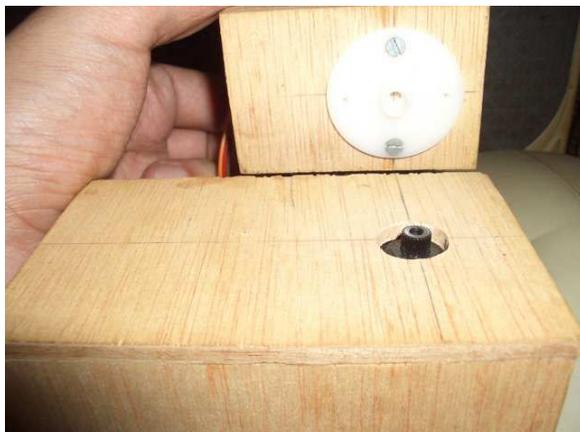


### Construcción de la caja base

Empiece por pegar con cola la base de fijación de 12,5 mm a la parte superior. Luego taladre los agujeros del pivote principal para alojar el tarugo de acero de 6 mm. Taladre atravesando la parte superior y la base de fijación, inserte el tarugo y péguelo bien en su sitio. Pegue las otras dos bases en su lugar, en la parte superior. Estas bases de fijación se utilizaran para montar uno de los servomotores. Taladre el otro agujero de la parte superior de modo que a través del mismo pueda deslizarse el husillo, y atornile el motor sobre las bases, utilizando 4 tornillos sin tuerca y arandelas de goma. Pegue los extremos y el lado de la caja. Una vez secos, lije las esquinas y los bordes.



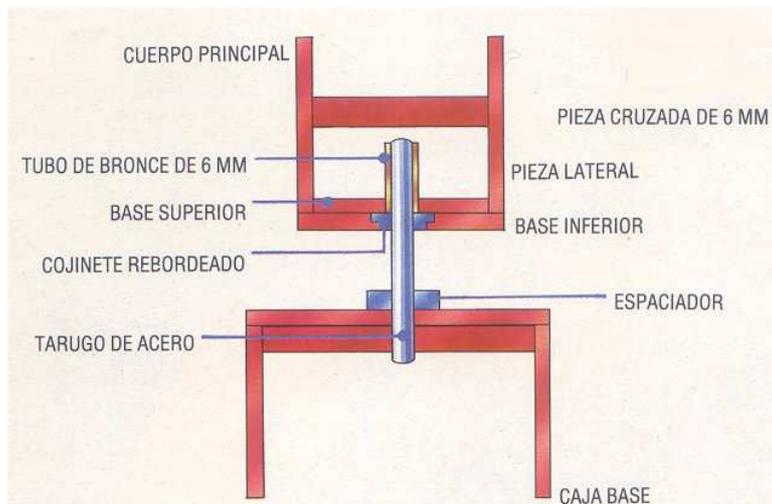
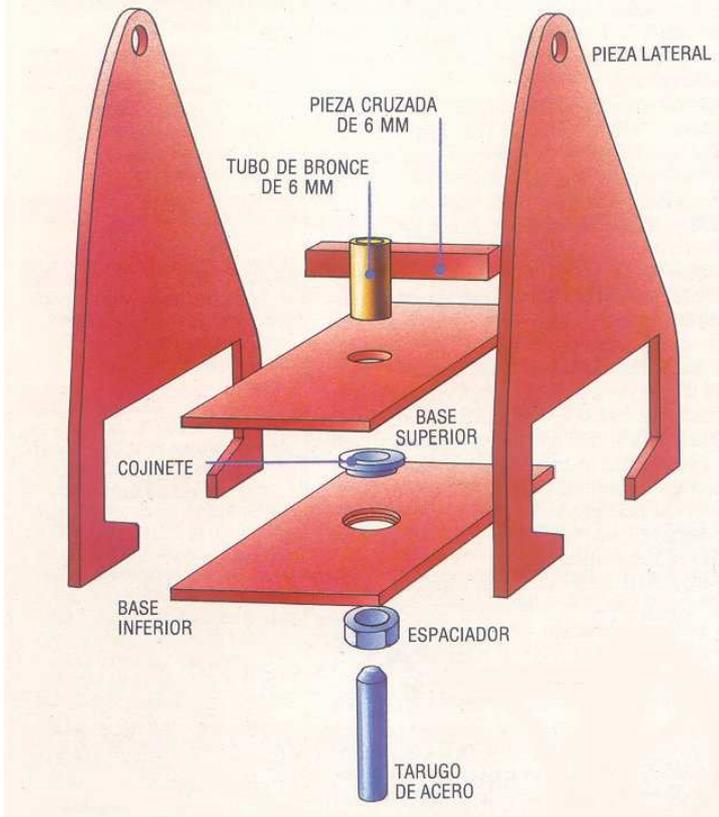
*La sujeción del servo en la caja base la solución de la misma manera, pero no instale el soporte ni el tarugo de acero. En su lugar instale el servo de forma que el propio eje fuera el apoyo de la base (ver fotos). Atornille el husillo del servo directamente a la base. Tiene el inconveniente de que el peso se carga sobre el eje del servo, pero tiene la ventaja de sus sencillez*



## Capítulo 2: Ensamblaje del brazo

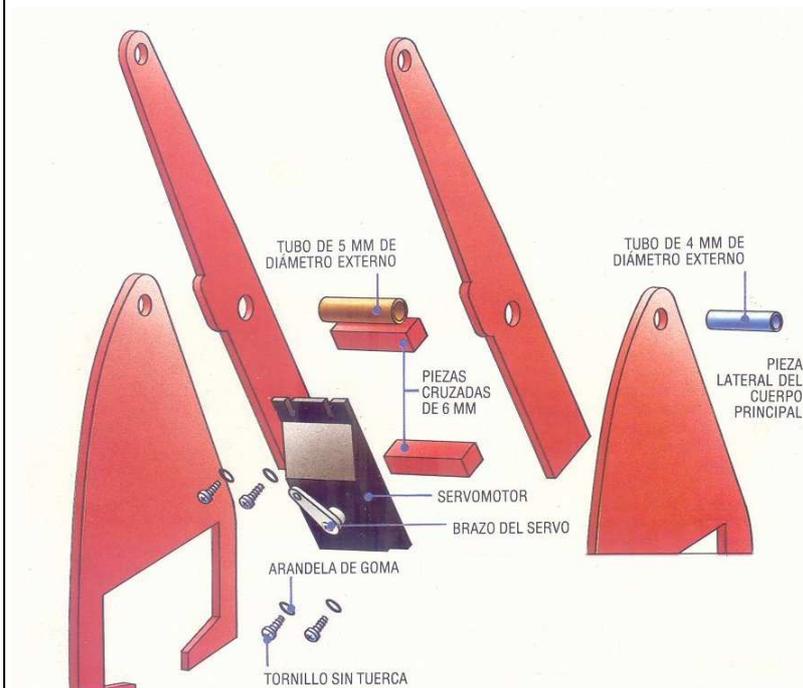
Ensamblaremos el cuerpo principal, la sección superior e inferior del brazo y efectuaremos las juntas del codo y hombro.

### Paso 1: Cuerpo principal



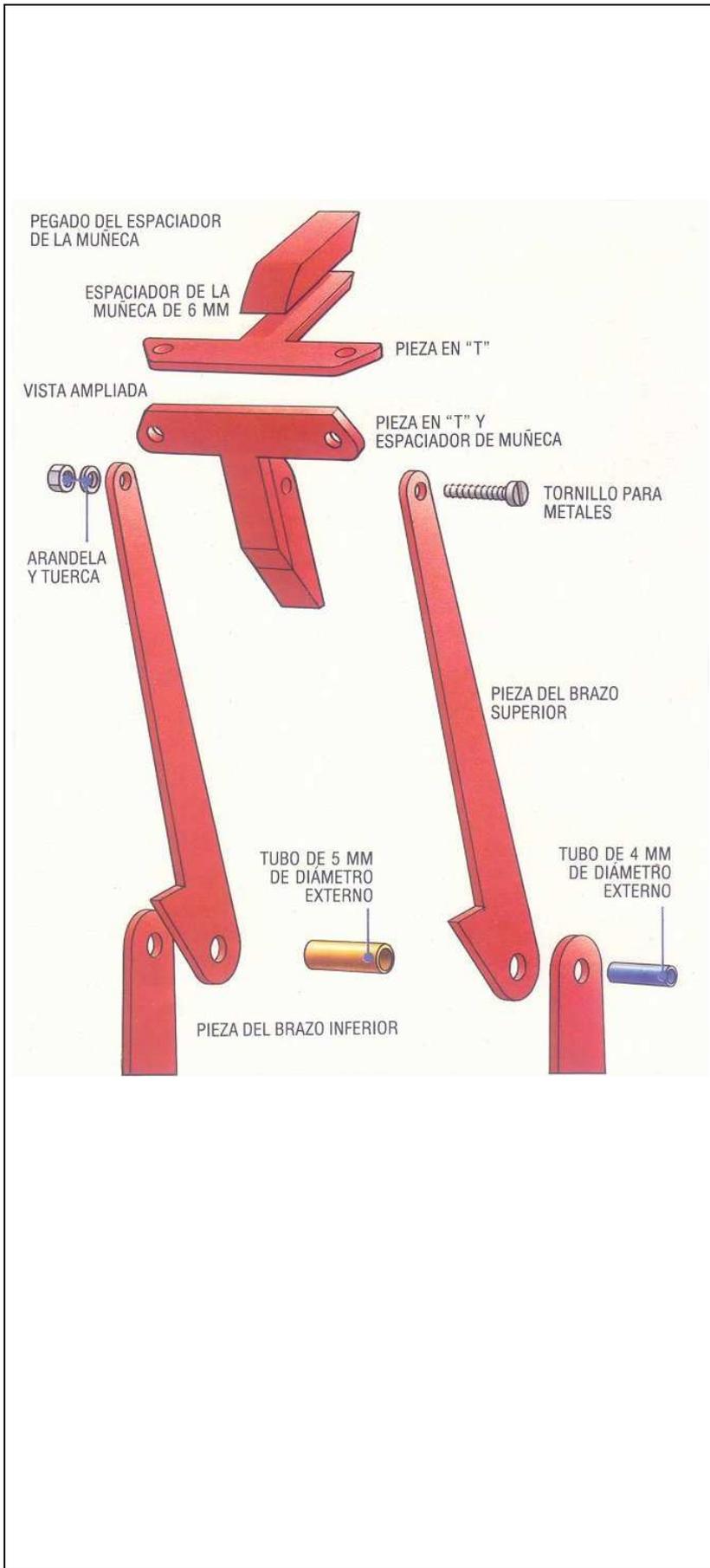
### Paso 1: Cuerpo principal

Taladre un agujero en la pieza inferior de la base (la más ancha de las dos piezas) para aceptar el cojinete. Utilizando una broca de mayor tamaño o un avellanador, bisele el labio superior del agujero para asentar en él el reborde del cojinete, como se ve en la ilustración. Después taladre un agujero de 6 mm de diámetro en la pieza de la base superior para recibir el trozo de 25 mm de tubo de 6 mm de diámetro interno. Con los agujeros alineados y el cojinete asentado sobre la pieza inferior de la base, pegue las dos piezas entre sí. Inserte el tubo de bronce en el agujero, desde arriba, y péguelo en su sitio. Cuando este ensamblaje esté seco se pueden añadir las piezas laterales del cuerpo. Perfore los agujeros en la parte de arriba de las piezas laterales de modo que acepten estrechamente un trozo de tubo de 4 mm de diámetro externo: se trata de un ajuste de fricción. Inserte un trozo de madera de 6x6 mm y 37 mm de largo a modo de pieza cruzada entre los dos laterales (colocada en la parte trasera del ensamblaje, en línea con el corte para el motor de la pieza lateral) y pegue el ensamblaje. Cuando éste esté seco, coloque un espaciador (p. ej., una tuerca grande) en el tarugo de acero que sobresale desde la parte superior de la caja base ensamblada anteriormente. Empuje el tarugo a través del cojinete y compruebe que el ensamblaje del cuerpo principal pueda girar libremente.



## Paso 2: Brazo inferior

Corte un trozo de 37 mm de tubo de bronce de 5 mm de diámetro externo y taladre agujeros en el medio de las piezas del brazo inferior para recibir este tubo. Asegúrese de obtener un ajuste apretado, puesto que este tubo de bronce formará parte de la junta del hombro. Perfore otros dos agujeros en el extremo piramidal de las piezas del brazo inferior para recibir el tubo de bronce de 4 mm de diámetro externo. Como antes, asegúrese de conseguir un ajuste de fricción. Trabajando en el medio, pase el tubo de 5 mm a través de los agujeros de las piezas del brazo inferior utilizando, como vemos en la ilustración, uno de los servomotores como espaciador. Corte dos trozos de 20 mm de largo de madera de 6 mm como piezas cruzadas y colóquelas entre las piezas del brazo inferior. Estas van justo debajo del tubo de bronce y en la parte de abajo del ensamblaje, y se emplean para montar el servomotor. Péguelos en su sitio y, una vez secos, monte el motor utilizando cuatro tornillos sin tuerca y arandelas de goma. Ya se puede unir el ensamblaje del brazo inferior al cuerpo principal, por la junta del hombro. Ahora inserte un trozo de 45 mm de largo de tubo de 4 mm a través de uno de los agujeros de la parte superior del cuerpo principal y a través del tubo de bronce que ya estaba colocado en el ensamblaje del brazo inferior. Asegúrese de que el ensamblaje del brazo inferior esté centrado. Marque la posición en la que parezca estar correctamente alineado, desmóntelo, y luego pegue el tubo de bronce más grande en su lugar en el ensamblaje del brazo inferior. Una vez seco, vuelva a ensamblar la junta del hombro.



**Paso 3: Brazo superior**

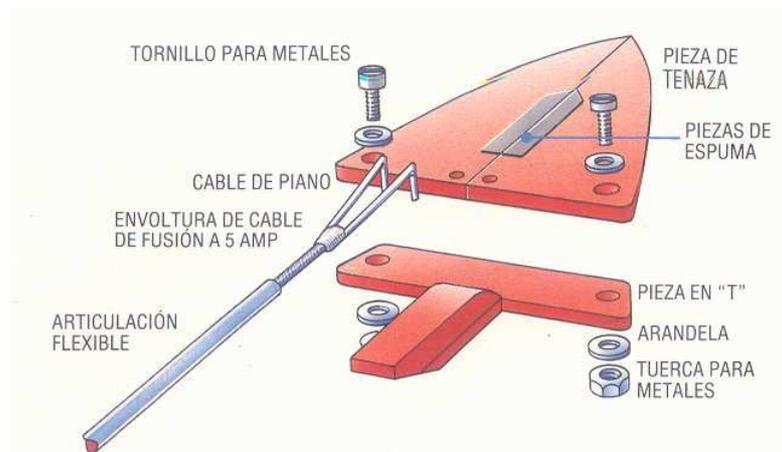
Perfore dos agujeros para recibir el tubo de bronce de 5 mm en los extremos más anchos de las piezas del brazo superior. Perfore otros dos agujeros en los otros extremos. Pegue un trozo de 25 mm de madera de 6x6 mm en la pieza en "T" y moldee los extremos. Éste actuará a modo de espaciador de la muñeca. Coloque la sección en "T" entre las dos piezas del brazo superior y taladre a través del espaciador de la muñeca de modo que a través del espaciador y de las secciones del brazo superiores se pueda pasar un tornillo para metales. Fije la pieza en "T" en su sitio con una arandela y un tornillo. En el otro extremo del brazo, empuje a través de los agujeros taladrados un trozo de 17 mm de tubo de bronce de 5 mm, y conecte el ensamblaje del brazo superior al brazo inferior en la juntura del codo pasando a través de la juntura, como antes, un trozo de 25 mm de tubo de 4 mm. Compruebe que el brazo superior esté centrado antes de desmontarlo y pegar en su sitio el tubo de bronce de diámetro mayor.

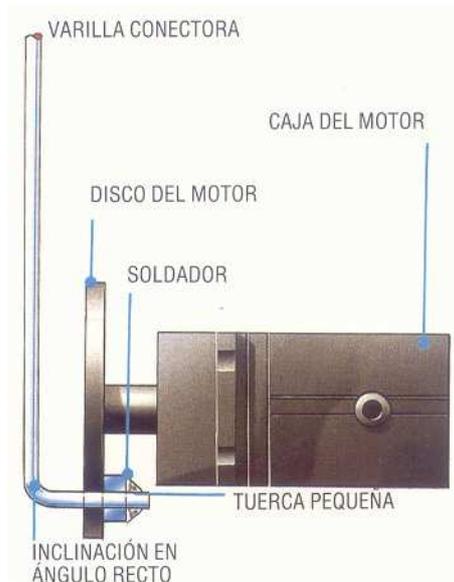
## Capitulo 3: Para asir

Llegados a este punto, nos corresponde diseñar el mecanismo de prensión y las conexiones necesarias para la "muscultura" del robot

### Paso 1: Ensamblaje de la pinza

Taladre agujeros en los centros señalados en la pieza en "T" y las piezas de tenaza para poder colocar los tornillos. Pegue tiras de espuma en la quijada de las piezas de tenaza a modo de relleno. Se deben perforar otros dos agujeros en las posiciones señaladas en los cortes del patrón, de modo que por el agujero quepa estrechamente un trozo de cable de piano rígido. Las piezas de tenaza se pueden ajustar a la pieza en "T" mediante los tornillos y las tuercas para metal. Inserte arandelas arriba y abajo de las piezas de tenaza de modo que cuando se ajusten los tornillos las piezas de tenaza queden libres para pivotar alrededor del tornillo. Tomando un extremo de la articulación flexible de 500 mm, fije dos trozos de 30 mm de cable de piano al cable central envolviendo los extremos del cable de piano y el cable central con alambre de fusión a 5 amp y sellándolo con soldadura. Incline hacia abajo los otros extremos de los cables de piano, formando ángulos rectos, y empujelos en los agujeros preparados en las piezas de tenaza. El cable central y los cables de piano deben formar una "Y". Con las quijadas de la pinza cerradas, ajuste la carcasa de la articulación con algodón y péguela en su sitio.





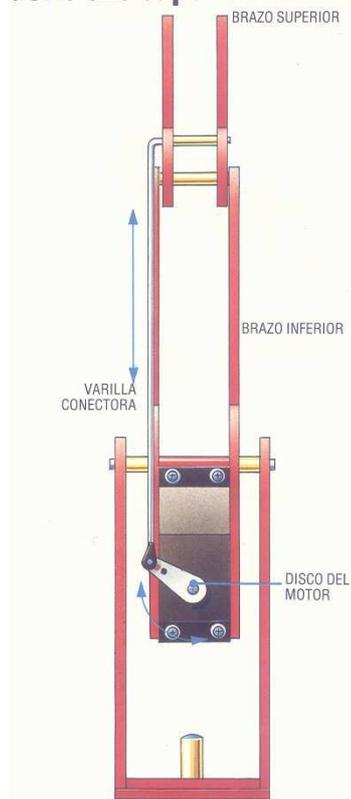
*En lugar de eso utilice algo bastante más moderno. En la tienda a la que fui me proporcionaron unos "enganches regulables" (paquete de 4 por 3,35€) que van fenomenal. Se trata de un soporte para la varilla mediante tornillo prisionero, tal y como se ve en algunas fotos anteriores.*

Conexión de las varillas a los discos

En el lugar donde adquiriera los servomotores podrá también comprar varillas conectoras especiales. Si no las consigue, la mejor forma de conectar una varilla de acero de 2 mm normal al disco del motor consiste en efectuar una inclinación en ángulo recto cerca del extremo de la varilla, pasándola a través del agujero del disco y soldando una pequeña tuerca en la parte posterior del disco. Esto se puede realizar con más facilidad con el disco fuera del husillo del motor.

Donde las varillas se colocan en agujeros de los miembros de madera del brazo, es una buena idea alinear el interior del agujero con una delgada pieza de tubo de cobre de un diámetro interno apenas más grande que el de la varilla conectora. Ello reducirá la abrasión de la madera cuando el brazo se mueva

## Paso 2: Varilla conectora del brazo superior

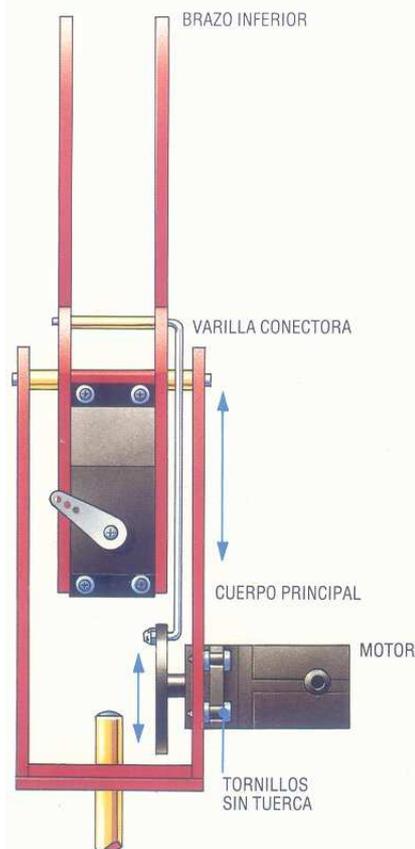


## Paso 2: Varilla conectora del brazo sup.

El brazo superior es controlado por el motor ya colocado en el ensamblaje del brazo inferior. El movimiento de rotación del husillo del motor se ha de convertir en un movimiento de tirar-empujar en el brazo superior. Ello se consigue ajustando un corto brazo de plástico (o "bocina") o un disco plástico de 30 mm al husillo. Ambos poseen un agujero por el cual se introducirá la varilla conectora. Aquí asumiremos que se están utilizando discos, pero el sistema funcionará perfectamente si en su lugar se emplean bocinas. Coloque el disco de 30 mm o la bocina en el husillo de modo que, cuando el motor gire completamente en sentido antihorario, el agujero esté justo a la izquierda de la posición de las seis en punto (imagínesse que el disco del motor es un cuadrante con las doce en punto situadas verticalmente

hacia arriba). Tome un trozo de varilla de acero de 2 mm de diámetro e incline los extremos formando ángulo recto de modo que la varilla quepa a través del agujero del disco del motor o la bocina y los dos agujeros preparados para ella en las piezas del brazo superior, cerca de la juntura del codo. La longitud de la varilla debe ser tal que los brazos superior e inferior formen un ángulo de 90° en el codo cuando el motor gire completamente en sentido antihorario. Compruebe la acción del brazo girando lentamente el disco del motor, a mano, en sentido horario. Pegue o suelde la varilla en su sitio en el disco.

### Paso 3: Varilla conectora del brazo inferior



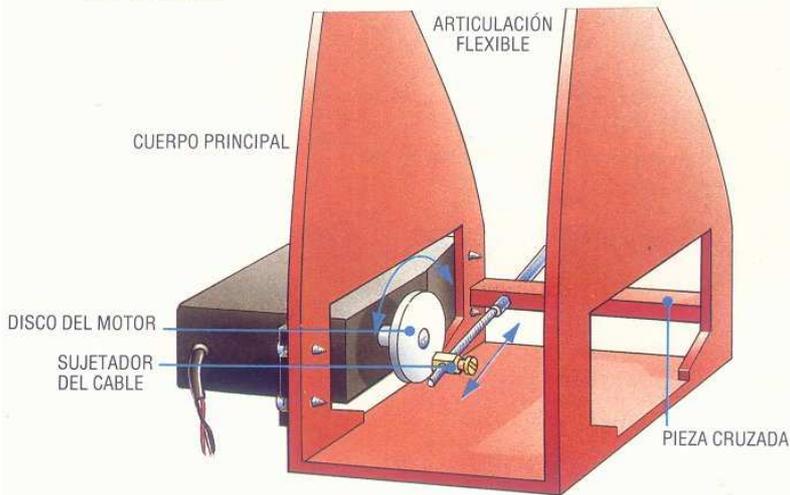
### Paso 3: Varilla conectara del brazo inferior

Monte los motores restantes en el cuerpo principal, como indica la ilustración. El motor de la derecha controla el movimiento del brazo inferior. Gire el motor de manera que quede por completo en sentido antihorario (visto desde el lado del disco del motor) y posicione el disco del motor de modo tal que el agujero de la circunferencia del disco quede justo a la derecha de la posición de las seis en punto. Ajuste un trozo de varilla de acero de 2 mm entre el disco del motor y los agujeros del brazo inferior que se hallan cerca de la juntura del hombro, como antes. La longitud de la varilla debe ser tal que el brazo inferior quede aproximadamente horizontal con el motor girado totalmente en contra de las agujas del reloj. Verifique la acción del brazo inferior haciendo girar a mano el disco del motor y pegue o suelde la varilla conectora al disco del motor.

### Paso 4: Conexión de la articulación al motor

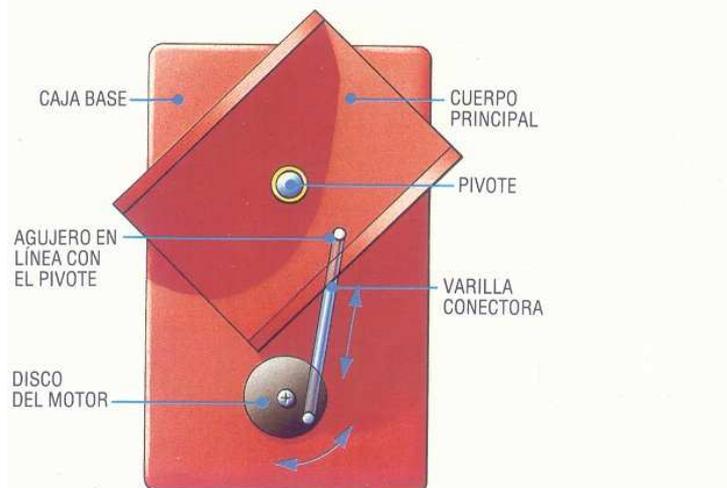
El motor montado a la izquierda del cuerpo principal se utiliza para abrir y cerrar la

## Paso 4: Conexión de la articulación al motor



pinza, empujando o tirando el cable central de la articulación flexible. Comience por taladrar un agujero a través de la pieza cruzada del cuerpo principal de modo que, como se ve en la ilustración, la articulación flexible se pueda deslizar a través del mismo desde atrás. Pegue la carcasa exterior de la articulación en este agujero. Cuando la cola esté seca, pase el cable central por este agujero, de modo que las quijadas de la pinza queden bien cerradas, y ajuste el extremo del cable al disco del motor. El agujero de la circunferencia del disco debe estar en las "nueve en punto" cuando el motor gire totalmente en sentido antihorario. El extremo del cable se puede fijar al disco del motor de varias maneras. Lo mejor es montar en el disco un pequeño prensacables. De lo contrario, se puede realizar una inclinación de 90° en el extremo del cable, y pegar o soldar en su sitio el extremo que se ha pasado a través del agujero del disco. Independientemente de cómo efectúe esta conexión, lo importante es que el motor pueda empujar y tirar del cable mientras vaya girando.

## Paso 5: Varilla del cuerpo principal



### Paso 5: Varilla del cuerpo principal

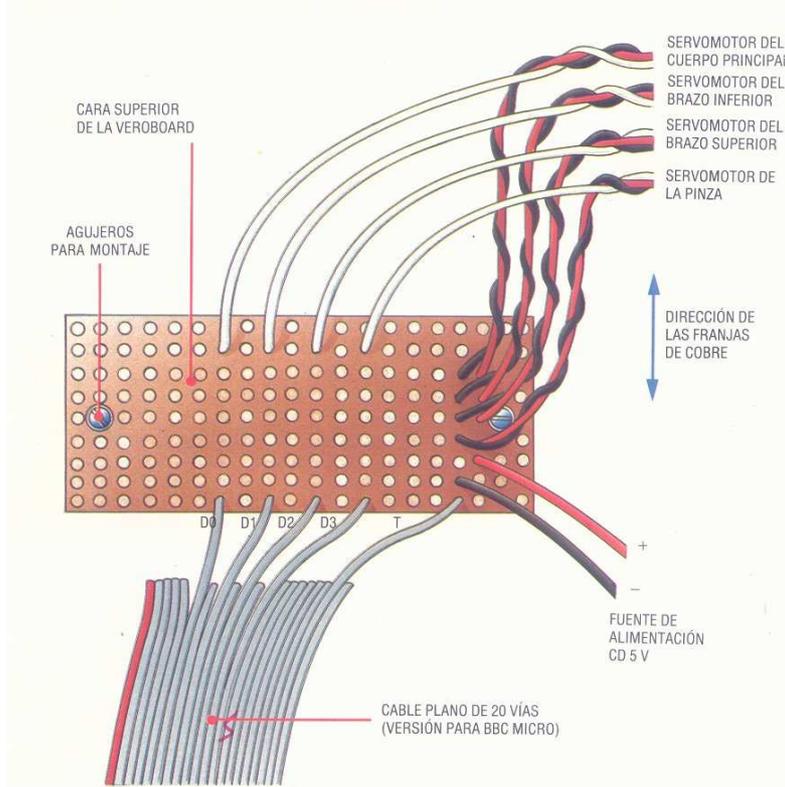
La última conexión a realizar es la del motor montado en la caja base al ensamblaje del cuerpo principal. Taladre un pequeño agujero en la base del cuerpo principal (necesitará correr el ensamblaje del tarugo de acero). El agujero se debe situar en línea con el pivote, junto al borde derecho (visto desde arriba) de la base del cuerpo principal. Realizado el agujero, vuelva a colocar el ensamblaje del cuerpo principal en el tarugo y gire hasta la posición de las dos en punto. Gire el motor por completo en sentido antihorario y posicione el disco de modo que el

	<p>agujero de la circunferencia quede justo a la derecha de las seis en punto. Coloque una varilla de conexión como antes y compruebe que el cuerpo principal quede libre para girar en sentido antihorario.</p>
--	--

## Capitulo 4: Conexiones

Habiendo cubierto ya las principales etapas de construcción de nuestro brazo-robot, sólo nos resta efectuar las conexiones eléctricas

### Paso 1: Conexiones de la veroboard

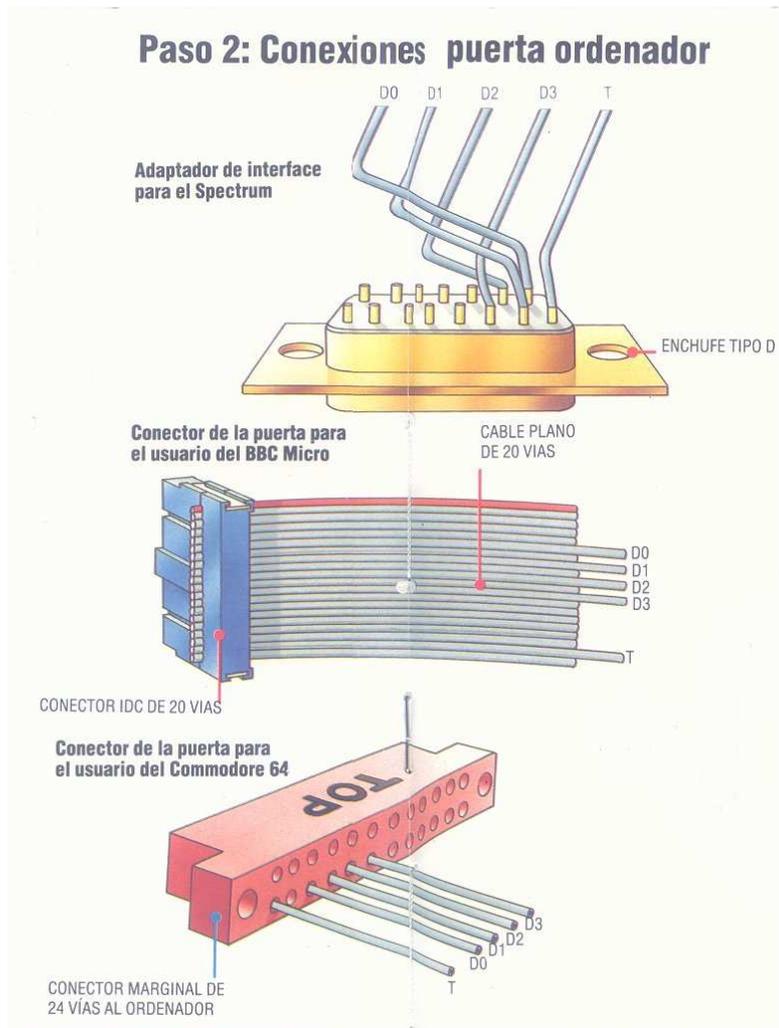


### Paso 1: Conexiones de la veroboard

Cada servomotor tiene tres cables unidos a él. En los servomotores Futaba que recomendamos para este proyecto, los colores de estos cables son: blanco para la línea de control, rojo para la línea de potencia positiva y negro para la línea de retorno común. Los cables del servomotor se deben unir al ordenador controlador y a la fuente de alimentación utilizando un rectángulo de veroboard de 20 franjas x9 aguj., que se montará en la parte posterior de la caja base del brazo. Corte la veroboard al tamaño adecuado y taladre dos agujeros para el montaje en las posiciones que se indican. Utilice la veroboard a modo de plantilla para marcar las posiciones de los agujeros en la parte posterior de la caja base y haga las perforaciones, los agujeros de la caja base y la veroboard se deben taladrar de modo que a través de ellos se pueda colocar un tornillo para metales. Haga otro agujero en la parte posterior de la caja base y haga pasar a través del mismo el grupo de cables del servomotor montados dentro de la base. Una entre sí los cuatro grupos de cables de los servos a la parte posterior de la caja base y suelde los cables en las posiciones que se indican. Las cuatro líneas de datos y la línea a tierra del ordenador se deben soldar en la parte de abajo de la veroboard. El diagrama ilustra un cable plano de 20 vías que se utiliza con el BBC Micro. Para el Spectrum y el Commodore se emplean

	<p>cinco cables separados, o un trozo de cable plano de cinco vías. Observe que las líneas de control blancas de los servomotores se sueldan en posiciones que se hallan directamente encima de cada una de las líneas de datos del ordenador, y que todas las líneas negras se sueldan en una única franja de cobre, encima de la conexión a tierra del ordenador y las dos conexiones de la fuente de alimentación. La tarea final de soldadura consiste en fijarle a la placa una fuente de alimentación de 5 V, como se observa en la ilustración. La fuente de alimentación de 5 V de que disponen el BBC Micro, el Spectrum o el Commodore 64 no es apropiada para alimentar los cuatro servomotores con una carga pesada. Debería buscarse, entonces, una fuente alternativa. Una pila de 4,5 V (o tres pilas de 1,5 V unidas con un sujetador de pilas) constituye una fuente de alimentación ideal. Si dispone de un transformador CD de 5 V, también lo puede utilizar. Si pretende emplear pilas, entonces debe soldar un sujetador de pilas en los extremos libres de las líneas de potencia que se extienden desde la veroboard. Si usted utiliza un transformador, debe instalar un conector de potencia en línea adecuado. El lado negativo de la fuente de alimentación comparte la misma franja de cobre con la conexión a tierra del ordenador y las líneas de retorno negras de los servomotores; el cable de potencia positivo se conecta a cada uno de los cables rojos de los servomotores a través de una franja de cobre común.</p>
	<p><b>Paso 2: Conexiones puerta ordenador</b> Después de realizar el cableado de la veroboard, debe añadirse el conector</p>

## Paso 2: Conexiones puerta ordenador



adecuado a la puerta del ordenador a los extremos libres de las líneas de datos de D0 a D3 y la línea de tierra. Para el BBC Micro debe utilizarse un cable de 20 vías estándar y un conector IDC. El Commodore 64 emplea un conector marginal de 0,15 pulgadas y 24 vías. Debido a que este conector se enchufa en la puerta para el usuario hacia arriba o hacia abajo indistintamente, antes de comenzar marque uno de los lados como ARRIBA. Las cinco líneas de la veroboard deben soldarse tal como se indica. Tanto el BBC Micro como el Commodore 64 poseen un sistema de circuitos incorporado para tratar las aplicaciones de control a través de una puerta para el usuario. El Spectrum, sin embargo, carece de un sistema de circuitos de este tipo, y debe construirse una interface especial para enchufar en su puerta de ampliación. En anteriores capítulos ya hemos diseñado y construido una interface de este tipo. Ahora los usuarios de un Spectrum habrán de remitirse a aquellos capítulos para construirse la interface. Un conector tipo D de 12 vías que se enchufa directamente en el robot proporciona las líneas de datos y de potencia de la interface. Podemos adaptar este conector para utilizarlo con el brazo-robot. Confeccione un enchufe tipo D de 15 vías empleando los extremos libres de las líneas de datos y tierra provenientes del trozo de veroboard del brazo, y añádale una cubierta para enchufe. Este adaptador permite enchufar el brazo-robot en la puerta para ampliación del Spectrum, a través de la placa de interface.

## Paso 3: Montaje del tablero

Cuando haya terminado todo el cableado y lo haya probado cuidadosamente, debe montar

<p><b>Paso 3: Montaje del tablero</b></p>	<p>la veroboard en la parte posterior de la caja base utilizando dos tornillos para metales ajustados mediante tuercas dentro de la base. Asegúrese de que los motores queden conectados a la veroboard en el orden correcto. La línea de datos DO (la más alejada hacia la izquierda) debe conectarse a la línea blanca del servomotor del cuerpo principal, montado en la caja base; 01 debe conectarse al servomotor del brazo inferior, montado a la izquierda del cuerpo principal; 02 controla el servomotor del brazo superior, montado junto a la juntura del hombro, y D3 controla el servomotor de la pinza, montado a la derecha del cuerpo principal.</p>
<p><i>Evidentemente todo el tema de conexionado lo cambie.</i></p> <p><i>El diseño original se basaba en unos interfaces previos (aquí solo esta su adaptación) para el Spectrum, Comodote 64 y BBC Micro.</i></p> <p><i>He omitido deliberadamente toda esa circuitería previa porque no tiene sentido, pero si alguien la necesita solo ha de pedírmelo y gustosamente se lo enviare.</i></p>	

## Capítulo 5: A prudente distancia

Nos corresponde diseñar el primer fragmento de software para controlar los movimientos del brazo-robot

Los cuatro motores que mueven el brazo están conectados a las cuatro líneas de datos del ordenador con las que está conectado el brazo en interface, que son también las que controlan los motores. En el caso del BBC Micro, estas cuatro líneas de datos parten de la puerta para el usuario utilizando las patillas de DO a D3. Debido a que los servomotores se basan en un flujo continuo de impulsos que les indican qué ángulo seguir, no se los puede controlar exclusivamente desde basic, sino que deben ser activados mediante un programa cuña en código máquina, que envíe un nuevo grupo de señales a los motores una vez cada dieciseisavo de segundo utilizando una interrupción. (Anteriormente ya hemos explicado detalladamente los principios de este método de control.) Se puede emplear el programa de control para servomotores múltiples ofrecido con anterioridad para comprobar la operación del brazo directamente desde el teclado y, si bien no es muy sofisticado, probará cada motor de forma independiente.

Una vez que el código cuña esté ejecutándose como tarea de fondo, será posible salir del programa de exploración de teclas del basic o colocar números directamente en las posiciones que controlan los ángulos de los motores.

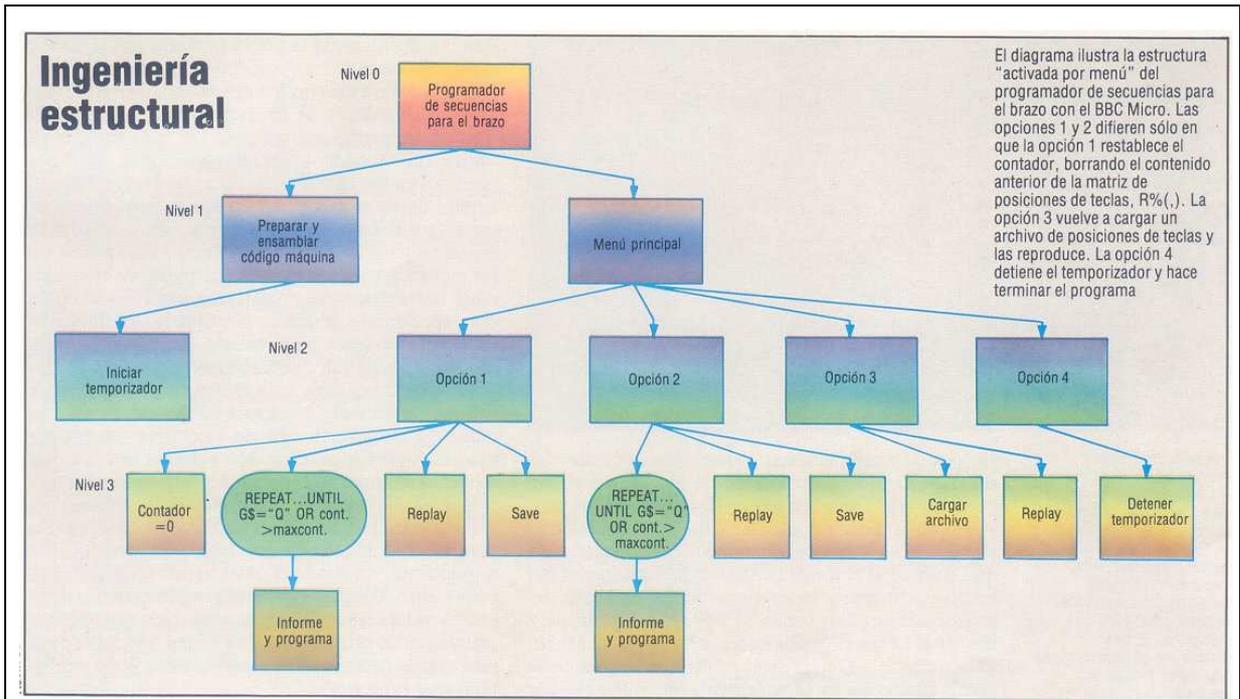
La dirección de la primera posición de memoria motor-ángulo es ángulo; motor 0 se establece utilizando el valor de esta posición.

La posición ángulo + 1 establece la posición del motor 1, y así sucesivamente. Recuerde que el motor 0 controla el movimiento de la cintura, el motor 1 el movimiento del hombro, y el motor 3 abre y cierra el mecanismo de prensión.

Si cualquier ángulo se incrementa o disminuye en una unidad, los motores lo pueden seguir fácilmente. Si se introducen cambios grandes, como de 0 a 128, entonces los motores pueden tardar unos instantes en reaccionar. Asimismo, observe cómo los efectos combinados de la inercia y "esponjosidad" del brazo hacen que éste se detenga de forma brusca, que recuerda la moda de la danza robótica.

Tras probar el brazo utilizando el sencillo programa de control, se hace evidente la necesidad de un programa más sofisticado. No obstante, decidir exactamente cómo programar el brazo probablemente sea la parte más difícil de toda la operación. Muchos robots industriales se programan moviendo la pinza manualmente a través de la secuencia que debe aprender el robot; el brazo graba de manera automática las posiciones clave de la secuencia. Lamentablemente, este método exige sofisticados elementos de hardware, como sensores de realimentación angular, de los que carece nuestro robot.

Pero incluso con equipos sofisticados, tales como los empleados para aplicaciones industriales, el



## Ingeniería estructural

El diagrama ilustra la estructura "activada por menú" del programador de secuencias para el brazo con el BBC Micro. Las opciones 1 y 2 difieren sólo en que la opción 1 restablece el contador, borrando el contenido anterior de la matriz de posiciones de teclas, R%(,). La opción 3 vuelve a cargar un archivo de posiciones de teclas y las reproduce. La opción 4 detiene el temporizador y hace terminar el programa brazo podría trabajar en entornos peligrosos en los que un operador no podría físicamente programar el brazo. En consecuencia, un método alternativo consiste en controlar el brazo a distancia desde el teclado, haciéndolo pasar por una serie de movimientos y guardando las posiciones clave de la secuencia mediante la pulsación de otra tecla, y registrando los cuatro ángulos de motor que definen cada posición en una matriz. La secuencia se puede reproducir simplemente recorriendo la matriz de posiciones de teclas con un factor de retardo adecuado. Las secuencias también se pueden guardar o cargar desde cinta o disco empleando un archivo secuencial para almacenar el contenido de la matriz. Éste es el procedimiento que utilizaremos. La primera etapa del diseño de nuestro software es, por lo tanto, asignar teclas para controlar las diversas funciones del brazo, de modo que pueda "enseñarsele" una secuencia de movimientos moviéndolo a distancia desde el teclado. Se pueden reasignar las teclas del cursor del BBC Micro para que controlen el movimiento izquierda/derecha del cuerpo principal y el movimiento arriba/abajo del brazo inferior. Las teclas A y Z se utilizarán para mover el brazo

valores de R%(,) utilizando valores previamente guardados en un archivo secuencial. Por tanto, tomado como un todo, el programa nos ofrece un método cabal para programar, editar y reproducir secuencias de movimientos del brazo.

### Un movimiento más uniforme

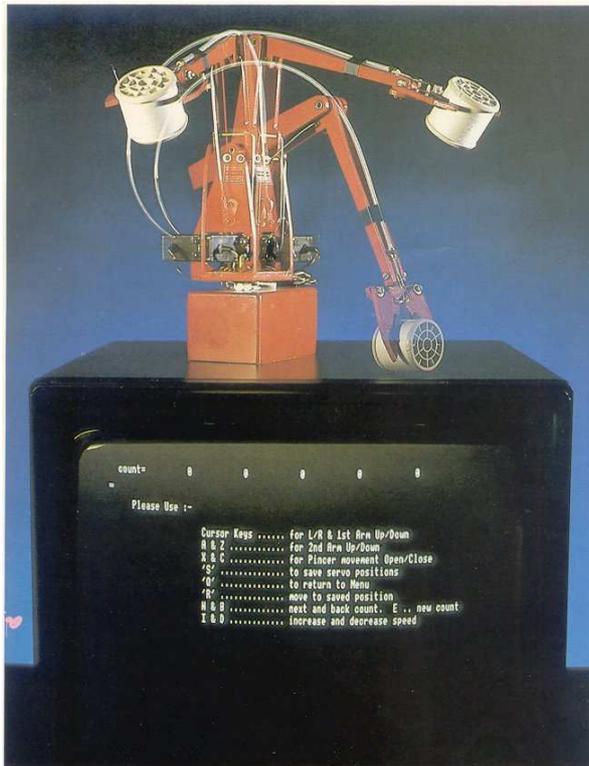
La rutina en código máquina para manipulación de interrupciones controla los ángulos de los servomotores analizando cuatro posiciones, que comienzan en la dirección ángulo. Cada una de estas posiciones contiene un valor que corresponde al ángulo que debe asumir cada motor. Una de las razones por las cuales el programa de prueba original produce movimientos discontinuos es que el programa en basic coloca (POKE) valores directamente en estas posiciones. Los grandes cambios en un ángulo producen un movimiento violento del motor, dado que éste intenta alcanzar la nueva posición con la mayor rapidez posible. Para conseguir que el brazo se desplace de forma más uniforme, deberíamos tratar de ajustar los valores de las cuatro posiciones ángulo sólo en pequeñas cantidades. Por este motivo se utiliza un segundo grupo de posiciones, que comienzan en nuevapos%, para aceptar los cambios de ángulo desde el programa en basic. Se puede añadir luego al programa una corta rutina en código máquina para aumentar o disminuir los valores de las cuatro posiciones ángulo (en pasos de una unidad) hasta que concuerden con los valores de la correspondiente posición nuevapos%. El

superior hacia arriba y hacia abajo, y las teclas X y C para abrir y cerrar las garras de la pinza. Las teclas I y D permiten aumentar o reducir la velocidad del brazo, de modo que las maniobras delicadas se puedan llevar a cabo lentamente, mientras los movimientos de barrido amplios se realizan con rapidez. La tecla S permite almacenar la posición actual del brazo en la matriz de posiciones de teclas, e Y retorna el brazo a la última posición almacenada. La secuencia se puede realizar para introducir cambios, ya sea hacia adelante empleando la tecla N, o bien hacia atrás, pulsando la tecla B; E controla movimientos a ciertos puntos de la secuencia. Utilizando estas tres últimas teclas, se puede editar cualquier secuencia de posiciones programada.

Los procedimientos informe y programa del listado le proporcionan una lista de las funciones de las teclas, además de explorar repetidamente el teclado en busca de una entrada.

Los otros procedimientos, replay y save, permiten reproducir o guardar en cinta o disco la secuencia retenida en la matriz de posiciones de teclas, R%(,), mientras que cargar archivo reasigna los

procedimiento moverservo simplemente llama al programa en código máquina que hace esto. La introducción de esta nueva rutina ofrece una ventaja adicional. Mediante la inserción de un bucle de demora dentro de este trozo de código máquina, podemos retardar o acelerar la velocidad a la cual se mueve el motor desde su antigua posición a su nueva posición, simplemente alterando el valor que da por terminado el bucle de demora. La posición &81 almacena este factor de retardo, que se puede alterar desde el basic al comienzo del procedimiento replay, permitiendo implementar a diferentes velocidades las secuencias de movimientos programados previamente. El código máquina para esta nueva rutina comienza en la línea 2000 del listado que ofrecemos.



#### Recogiendo las piezas

Comenzamos el proyecto del brazo-robot considerando una tarea determinada: recoger y trasladar un objeto del tamaño y peso de un carrete de hilo de coser. El programador de secuencias nos permite programarlo para que realice esta tarea, guardando los movimientos, si fuera necesario, en un archivo que se puede volver a cargar y reproducir

*Y, para terminar aquí teneis unas fotos de mi brazo completo:*



# Sección 3: El circuito de control

## Capítulo 1: Primeras ideas

Cuando me planteé hacer este montaje evidentemente apareció el tema del control.

Por supuesto no podía utilizar los interfaces descritos en el proyecto porque estaban pensados para otros equipos que no eran PC. Así que solo me quedaron 3 opciones, a saber:

- Puerto paralelo
- Puertos serie
- Puertos USB

Los puertos USB se que son una “red” de dos hilos por las que circulan datos y además desconozco como implementarlos en VB6, así que quedaron descartados.

Los puertos paralelos para mi eran un poco “tabu” (una vez una malfuncion de uno de esos puertos, sumados a un inútil (yo) que dijo, “bah, no pasa nada si cambio el cable de impresora con el ordenador conectado”... produjo un ordenador con el puerto destrozado y una impresora con la placa quemada (junto con una cierta cantidad de “humillo” y quedar sin luz en la casa).

Sabía que eran delicados y que requerían algún tipo de circuito adaptador, así que solo quedaron los puertos serie.

El funcionamiento de un puerto serie lo conocía por una aplicación que había realizado hacia años. Sabia que eran robustos, cortocircuitables (asunto muy importante, dada mi anterior experiencia con los paralelos), podían enviar datos o “pulsos” (depende de los pines) y además los implementaba directamente VB.

Así que si mas dudas empecé a imaginar como gestionar todo esto con un puerto COM.

Lo primero fue comprar un único servo, conectarlo e intentar gestionarlo desde Visual.

Buscando en internet encontré como son y se gestionaban los servos, a saber:

- Se alimentan por los hilos rojo-negro con 5 V
- Se les envía la posición mediante un pulso por el cable marrón (o blanco) cuya duración (0,5 a 2,5ms) indican la posición (0 a 180°). Este pulso se debe refrescar cada 20ms.

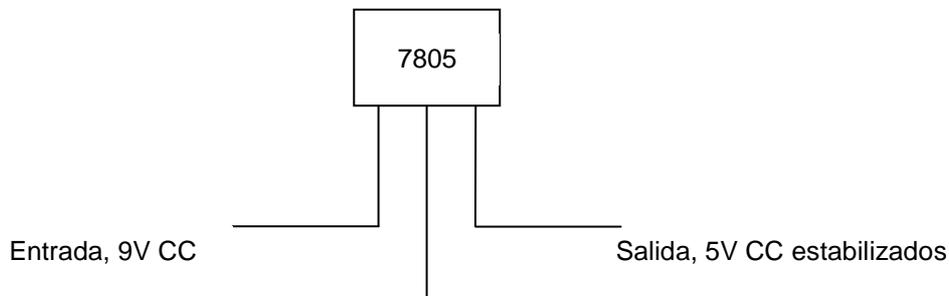
El puerto serie dispone de 8 pines “básicos”. Dos de ellos (2 y 3) son para transmisión/recepción de datos. Del resto, el 7 es el “común”, el 4, 6 y 8 son “entradas” y el 5 y 20 “salidas” (o el 4 y 6 salidas y el 5 y 20 entradas, según la configuración del propio puerto).

Todo eso es configurable desde el objeto "MsComm1" de visual así que en principio, no planteaba ningún problema.

El tema de la alimentación del servo a 5V tampoco representaba ningún problema ya que conocía el circuito integrado 7805 que sirve para regular una tensión de entrada y es comúnmente utilizado en alimentadores de corriente (dispone de 3 pines, entrada (> 5 V recomendable 9V), común y salida de 5V exactos).

Para la transformación de 220V CA (desde 2003 realmente 230V) a >5V CC se podría construir un alimentador de corriente o bien... en el despacho sobran varios alimentadores de modems, routers, móviles etc. ¿Por qué liarse en construir algo cuando ya lo tienes? elegí un alimentador de un antiguo modem con una entrada de 230V CA y una salida de 9V CC.

Para la estabilización simplemente prepare una placa con un 7805 (su precio es ínfimo y no tiene problemas de sobrecalentamiento, ni de cortocircuito, ni nada de nada... ideal para novatos) el montaje era algo así:



Bueno... un tema solucionado. Ahora solo faltaba como enviar los pulsos.

Sabia que las propiedades "RTSEnable" y "DTREnable" del objeto "MsComm1" de visual según las pusieras a "false" o "true" activaban o desactivaban los pines correspondientes del puerto.

En el despacho había un pequeño objeto que era un monitor de puerto serie así que lo conecte y empecé a probar manualmente.



Botón en visual... RTSEnable=true... se enciende el led.... Otro botón... RTSEnable=false... se apaga... PERFECTO.

Solo queda conectar el pin al cable marrón del servo y generar los pulsos por programa.

Pero OJO... el RS232 trabaja a 15V y el servo a 5V. Houston, tenemos un problema.

Se que existen conversores para puerto serie, pero mis conocimientos de electrónica son muy básicos, así que me estuve rascando la cabeza un rato, tuve una idea absurda.... y pensé, ¿por qué no?...

*Atención, nota informativa... los "puristas" electrónicos no deberían seguir leyendo, porque el montaje que viene a continuación puede ser perjudicial para su salud... en la vida se ha visto algo tan bruto...., pero, lo siento, entonces yo no daba para más.*

Me dije... si un 7805 convierte una tensión en otra.... y el puerto serie envía 15V (1 lógico) o no envía nada (0 lógico)... si pongo un 7805 después del pin del puerto serie... tendré una salida de 5V sincrónica con el puerto serie....

Así que me puse manos a la obra, prepare el montaje... y lo probé. Puse el tester a la salida del segundo 7805 y leí 5V o 0V según como pulsara yo en el programa... prueba conseguida.

Solo quedaba generar el soft para los pulsos, pero para eso ya tiene el visual temporizadores.

Prepare un objeto "Timer1" con un parámetro de 20 (es decir 20 ms) de forma que automáticamente se efectuaba una llamada a ese código cada 20 ms. La señal de refresco ya la tenía.

Solo había que poner el pin del serie a uno, durante el tiempo indicado por una variable, así que prepare otro objeto timer al cual yo le pasaba el valor y listo.

Pero vengaaaa... otro problema... el objeto timer de visual tiene un mínimo de 1 ms y no acepta decimales.

Así que cercanos al rango de 0,5 y 2,5 solo disponemos de 4 valores... 0, 1, 2 y 3... vamos a probar...

Arrancamos el invento... el led parpadea muy rápidamente, es el "refresco" de 20 ms... probamos un valor de 0... y no hace nada (claro). Le doy valor 1 y el servo se posiciona en... digamos... 70 grados... BIEN

Le doy valor 2 y se posiciona en aproximadamente 160 grados, BIEN

Le doy valor 3 y se va a "final de carrera" empieza a hacer ruidos raros... corto rápidamente. Evidentemente el servo se va a final de recorrido, se pasa de giro y empieza a forzar el potenciómetro.

Bueno, hagamos un alto para analizar el tema. El sistema FUNCIONA, el servo

FUNCIONA, el circuito FUNCIONA. Al soft le falta precisión. Hay que encontrar algo para que visual envíe datos mas rápido.

¿Y donde encuentro yo información de cosas raras para visual? Pues donde va a ser... en Internet.

## Capitulo 2: Purificando el soft

No me costo mucho encontrar una pagina en la cual explicaban que cualquier PC tiene un temporizador rápido, mejor dicho ultrarrápido.

<http://support.microsoft.com/default.aspx?scid=kb%3Bes%3B172338>

Hay un “contador” que se inicializa a 0 cuando arranca y se incrementa a la velocidad del reloj del procesador.

Evidentemente cuanto más rápido es el ordenador, más rápido se actualiza. Para controlarlo existen dos funciones. Una te da el valor del contador y otra la frecuencia de actualización.

Bueno... tengo un P4 a 1.600... supongo que será suficiente... hagamos un test... así que preparamos un pequeño código y lo lanzamos. Esto fue el resultado:

```
Start Value: 19731735,2034
End Value: 19731735,2049
QueryPerformanceCounter minimum resolution: 1/3579545 sec
```

Es decir... acabo de leer unos datos... de 0,00000027936511484 segundos (0,000279 milisegundos)

Suficiente no... mas que suficiente... no, más aun ES BRUTAL... jamás pensé que se pudiera conseguir algo tan rápido con VB6.

*Nota informativa, en experimentos posteriores comprobé que leer valores muy bajos es prácticamente imposible porque la propia llamada del sistema consume tiempo y a esos valores tan bajos puede que la propia llamada sea mucho mas que el tiempo que queremos leer. Como “control” os recomiendo que nunca uséis un bucle tipo “hasta que B sea igual a A” porque puede que cuando el bucle lea el valor ya haya “pasado” el contador. Es mejor un bucle tipo “hasta que B sea superior a A”*

Es decir preparamos el código que calcula el tiempo que necesitamos, lo dividimos (en mi caso) entre 3579545 y nos dará los “ciclos” que tenemos que esperar.

Activamos el pin, ejecutamos un bucle do...loop que lea el valor del contador y no acabe hasta que el contador no supere el valor adecuado yyyyyyyyyyy..... bingo el invento funciona perfectamente.

Con un slider deslizante voy aplicando el valor adecuado y el servo gira suavemente el ángulo indicado

Bueno... ¿suavemente?

Tiene un pequeño... bueno no tan pequeño “rateo”.... No lo dudes mas... es causado por el propio puerto serie... si utilizas (como has visto en muchas paginas) el puerto paralelo (como casi todos los roboticos no tendrás problemas)... (eso, como vi mas tarde no fue mas que otro error de apreciación, pero no adelantemos

acontecimientos...).

Así que me propuse en serio (y con miedo) acometer el diseño de una placa para gestión del puerto serie.

### Capitulo 3: La placa final

Una vez más Internet fue de gran ayuda. Entre muchas otras páginas encontré estas:

<http://www.infolaser.net/franpr/>

Donde el autor, Francisco Pulido Rubio nos deleita con un buen número de artículos técnicos muy interesantes. A destacar para las aplicaciones de robótica los siguientes:

[Control de motores Paso a Paso por el puerto paralelo del PC](#)

[Software de control de dos motores PAP con el ULN2803](#)

[Introducción al puerto paralelo del PC](#)

[Software de control de cuatro motores PAP con el 74LS574 y el ULN2803](#)

[Como modificar un servo para transformarlo en un motor con reductora de movimiento constante...](#)

[Controlar 8 servos RC por el puerto paralelo del PC...](#)

[Controladora Bipolar para motores Paso a Paso...](#)

[Programa que controla dos motores PAP por LPT...](#)

[Programa que monitorea el estado del puerto paralelo...](#)

[Programa que controla cuatro motores PAP por LPT...](#)

Y esta:

<http://www.pablin.com.ar/>

Esta más dedicada a la electrónica, dispone de los inestimables artículos siguientes:

[Cinco entradas digitales \(pulsadores\) desde el puerto paralelo](#)

[Control de motor paso a paso bifilar](#)

[Control de potencia con Relés](#)

[Control de potencia con Triac](#)

[Monitor para Puerto paralelo](#)

En primer lugar, en base a sus indicaciones, y echando mano de mi cajón de “cosas varias”, con unas resistencias y unos LEDs construí un monitor para puerto paralelo.

Como precaución adicional y dada mi nefasta experiencia con el puerto paralelo, puesto que el mío está en placa base, me imagine MI PLACA BASE generando humo y, la verdad... no me hizo ninguna gracia... Así que me fui a una tienda de informática y compré una tarjeta PCI de puerto paralelo. Si alguien tenía que morir, que fuese ella y no la placa base. Afortunadamente eso no sucedió y hoy dispongo de dos puertos paralelos.

Ahora nos encontramos con un problema, Visual no dispone de ninguna orden para leer o escribir datos en un puerto de E/S. Pero esto apenas represento ningún problema. Una vez más Internet fue de un valor incalculable. Hay muchas páginas en las que te ofrecen la dll “inpout32.dll” u otra similar que implementa las ordenes “out” (para escribir) y inp (para leer) los registros del puerto paralelo.

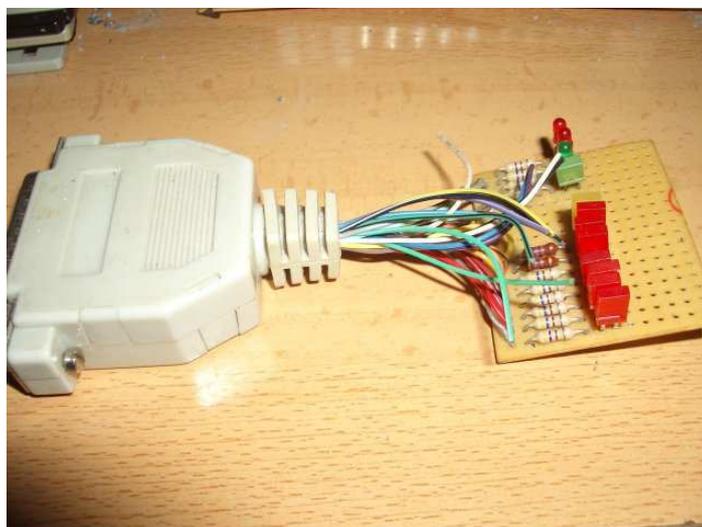
Así pues, uniendo las rutinas de “contador rápido” y la de “lectura/escritura de puertos” conseguí un código que funcionaba.

El código utilizado fue el siguiente:

Timer de refresco cada 20 ms (realmente lo puse variable con un slider)

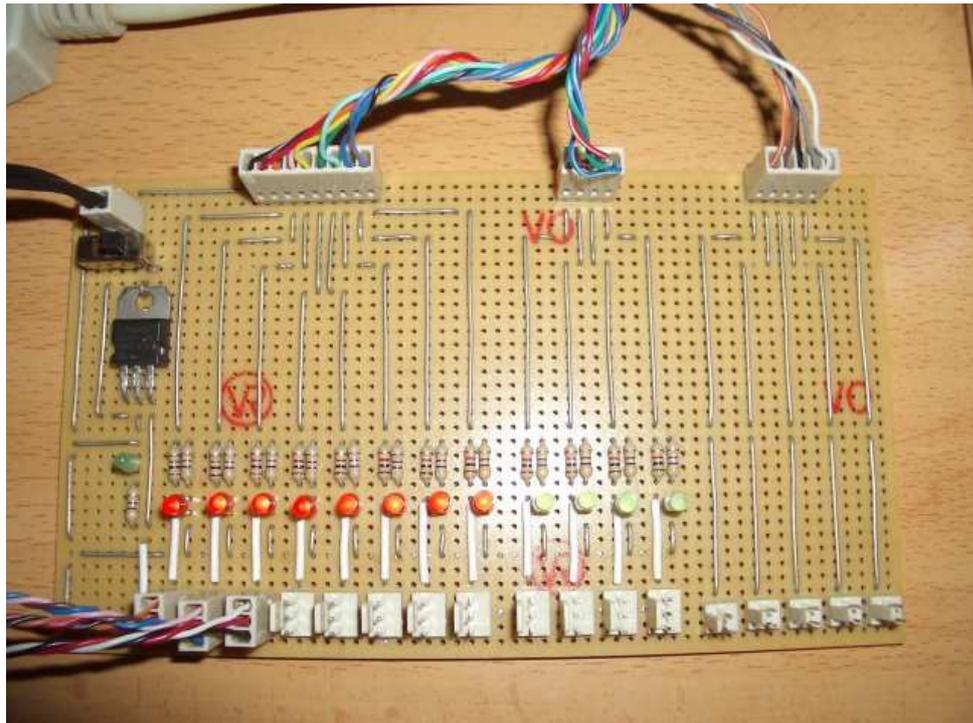
```
Private Sub TimerRefresco_Timer()  
TimerRefresco.Enabled = False 'impide que se realice otra llamada mientras se ejecuta este codigo  
For i = 0 To 7 'uno por cada una de las ocho salidas del puerto  
QueryPerformanceCounter ciclos  
valoractual = LargeIntToCurrency(ciclos)  
valorfinal = (valoractual + pasos(i))  
Select Case i 'pone a "1" logico el pin (i) del puerto paralelo  
Case Is = 0: BitDatos = BitDatos Xor 1: Out Puertobase, BitDatos  
Case Is = 1: BitDatos = BitDatos Xor 2: Out Puertobase, BitDatos  
Case Is = 2: BitDatos = BitDatos Xor 4: Out Puertobase, BitDatos  
Case Is = 3: BitDatos = BitDatos Xor 8: Out Puertobase, BitDatos  
Case Is = 4: BitDatos = BitDatos Xor 16: Out Puertobase, BitDatos  
Case Is = 5: BitDatos = BitDatos Xor 32: Out Puertobase, BitDatos  
Case Is = 6: BitDatos = BitDatos Xor 64: Out Puertobase, BitDatos  
Case Is = 7: BitDatos = BitDatos Xor 128: Out Puertobase, BitDatos  
End Select  
Do While valoractual < valorfinal  
QueryPerformanceCounter ciclos  
valoractual = LargeIntToCurrency(ciclos)  
Loop  
Select Case i 'pone a "0" logico el pin (i) del puerto paralelo  
Case Is = 0: BitDatos = BitDatos Xor 1: Out Puertobase, BitDatos  
Case Is = 1: BitDatos = BitDatos Xor 2: Out Puertobase, BitDatos  
Case Is = 2: BitDatos = BitDatos Xor 4: Out Puertobase, BitDatos  
Case Is = 3: BitDatos = BitDatos Xor 8: Out Puertobase, BitDatos  
Case Is = 4: BitDatos = BitDatos Xor 16: Out Puertobase, BitDatos  
Case Is = 5: BitDatos = BitDatos Xor 32: Out Puertobase, BitDatos  
Case Is = 6: BitDatos = BitDatos Xor 64: Out Puertobase, BitDatos  
Case Is = 7: BitDatos = BitDatos Xor 128: Out Puertobase, BitDatos  
End Select  
Next i  
TimerRefresco.Enabled = True  
End Sub
```

Con ese monitor estuve haciendo las pruebas de funcionamiento observando los parpadeos de los 8+4 LEDs, Todo fue perfecto, con lo que la transformación del código del puerto serie al puerto LPT fue absolutamente satisfactorio.



Una vez lo tuve en funcionamiento, me decidí a diseñar y construir la placa en cuestión. Solo era la suma de “controlar 8 servos con el puerto paralelo” más el “monitor para puerto paralelo” más la alimentación mediante un 7805 (esto ya de mi cosecha).

Aquí os dejo una foto de la placa acompañada de los diseños en PDF y CAD para poder construirla.



A partir de aquí solo quedaba probarlo.

No os podéis imaginar cual fue mi alegría y decepción al hacerlo...

Alegría porque funcionaba... perfectamente y además preparado para 8+4 servos y 5 entradas, un hito en mi.

Y decepción porque el famoso rateo no había desaparecido. Vaya... la primera vez que construyo un robot y tiene Parkinson... creo que ni el doctor Frankenstein lo hubiera hecho peor...

A ver ¿Qué falla? ¿el puerto paralelo? ¿el circuito? ¿el software? (porque coño siempre lo primero que pensamos es que nosotros somos los que nos equivocamos)

Al final llegue a la conclusión (equivocada) que los propios motores causaban algún tipo de interferencia) para lo que pensé en pedir ayuda “profesional” (“esto ya le debe haber pasado a alguien”, pensaba)

Después de dar muchas vueltas y sin encontrar solución encontré la página (que ya

conocéis):

<http://www.webdearde.com>

En la cual hice la consulta pertinente y la respuesta consensuada fue... que Windows no permite este tipo de uso "exclusivo y veloz" del puerto paralelo. Sus propios procesos internos (gestión del teclado, ratón, etc.) interfieren con el puerto. No hay solución...

Bueno... casi... porque buscando encontré una rutina:

<http://www.ilustrados.com/publicaciones/EpVAZVypZliRsNxUQB.php>

Como veréis la prioridad es una combinación de 2 parámetros, uno rige la clase del proceso (de 0 a 15) y el otro especifica su prioridad como subproceso (0 a 255)

Un proceso "normal" tiene 0 y 20H. Se pueden cambiar hasta 15 y 100H que es la máxima prioridad posible.

Para ello existen dos funciones

SetThreadPriority  
SetPriorityClass

Una explicación y el código en visualBasic esta en:

<http://www.lawebdelprogramador.com/preguntas/vercontestada.php?pagina=253&id=93&texto=Visual+Basic>

que aumenta la prioridad del proceso (el mío) hasta igualar la prioridad del propio sistema operativo. Eso redujo (redujo, que no elimino) mucho los temblores permitiendo seguir investigando

En estos momentos estoy preparándome para pasar a una placa con un pic (eso evitara el problema porque los pulsos los generara la placa y no el soft)

Y de alguna manera aquí acaba la historia... si he decepcionado a alguien porque esperaba un proyecto absolutamente completo y acabado, lo siento, pero como decía el de la tele "así son las cosas y así se las hemos contado".

No obstante creo que hay suficiente información para que sirva de guía y podáis experimentar vosotros mismos.