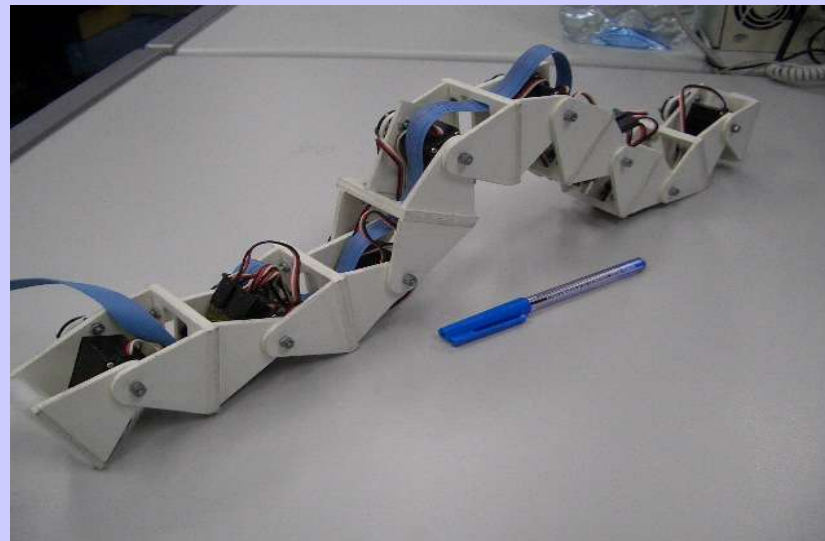
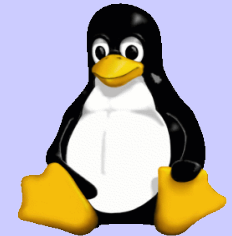


Robótica y Linux: Cómo se hizo *Cube Revolutions*



Juan González Gómez

Escuela Politécnica Superior
Universidad Autónoma de Madrid

La charla sobre Robótica y Linux normalmente la damos dos personas...

Una de ellas, Andrés Prieto-Moreno, acaba de ser padre:



Desde recién nacida se está familiarizando con el software libre:



ÍNDICE



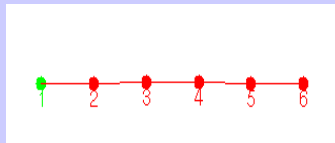
- Introducción



- Mecánica



- Electrónica



- Software

Hypercube

- Trabajo Futuro

Introducción

Robótica

Dos grandes áreas:

- **Manipulación:** Capacidad de modificación de los objetos del entorno



Robótica Industrial

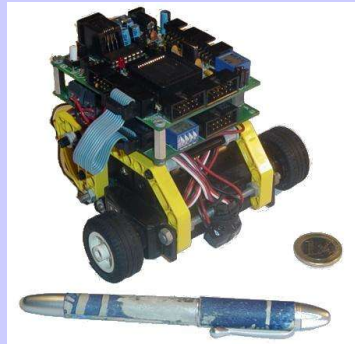
- **Locomoción:** Facultad para desplazarse de un lugar a otro



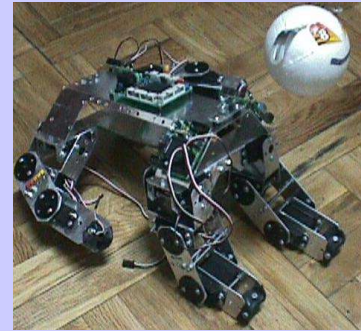
Robótica Móvil

Clasificación Robots

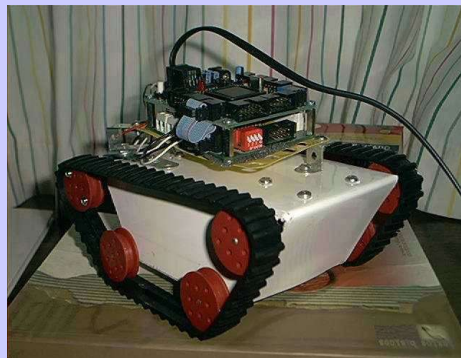
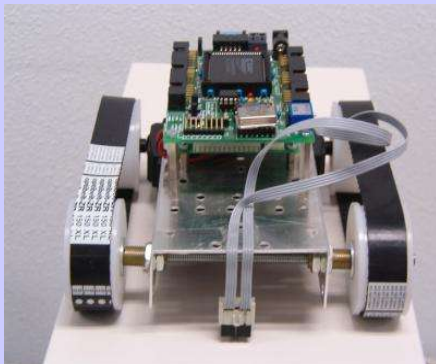
- Con ruedas:



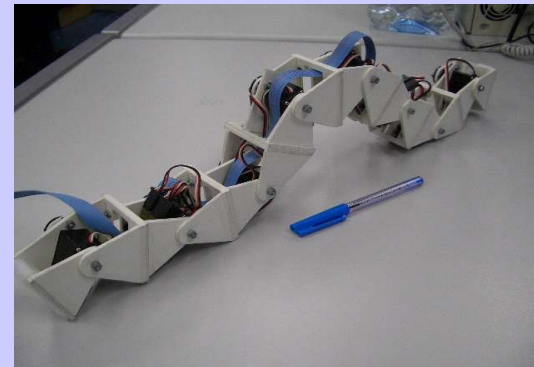
- Con patas:



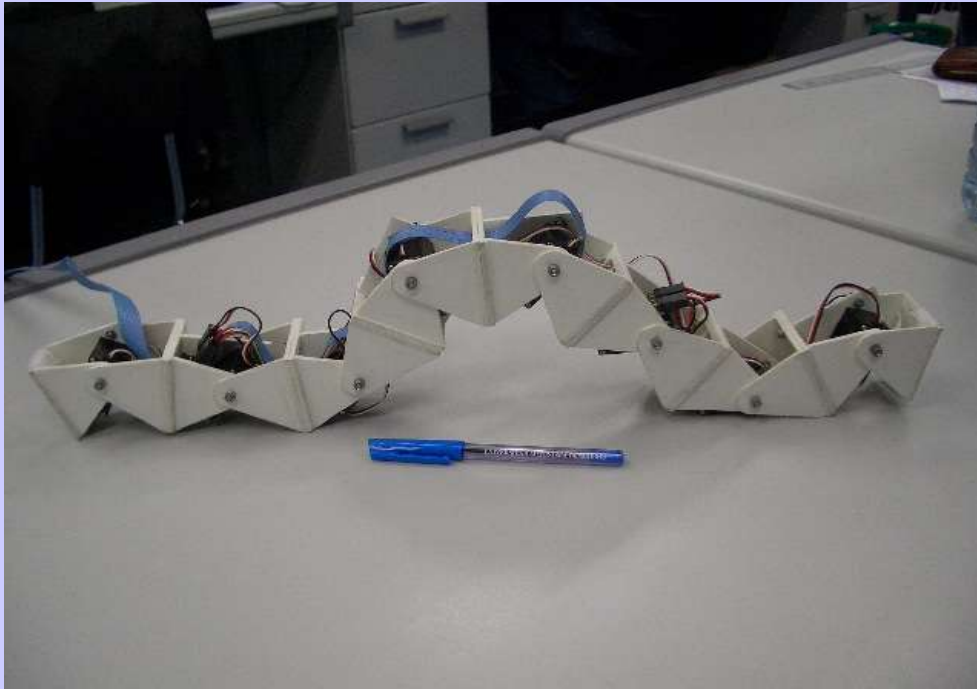
- Con orugas:



- Otros:



Cube Revolutions

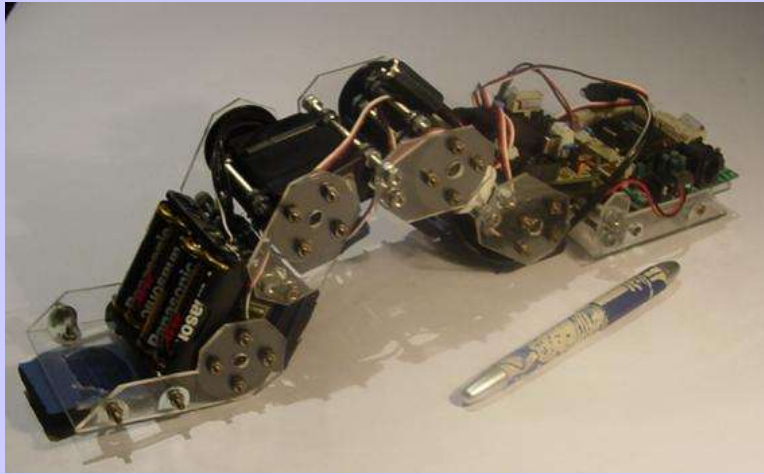


- **Robot ápodo Modular:** Constituido por 8 módulos iguales
- **Abierto:** Toda la información está disponible para que cualquiera lo pueda construir o modificar: planos y software.
- **Aplicaciones:**
 - Estudio de la locomoción
 - Acceso a zonas de difícil acceso

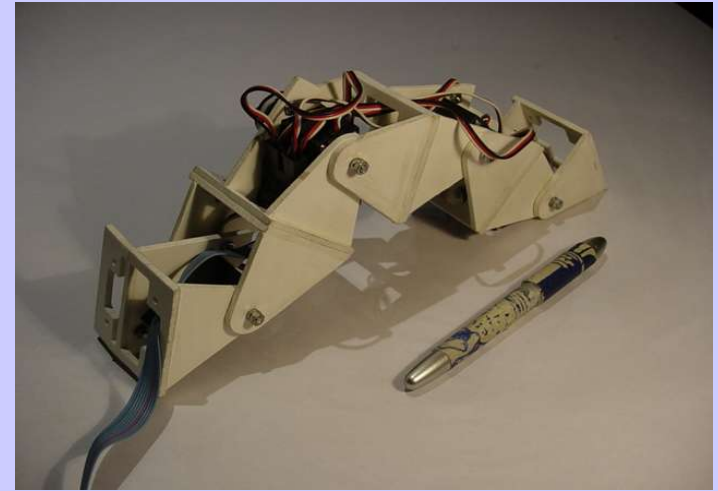


Evolución

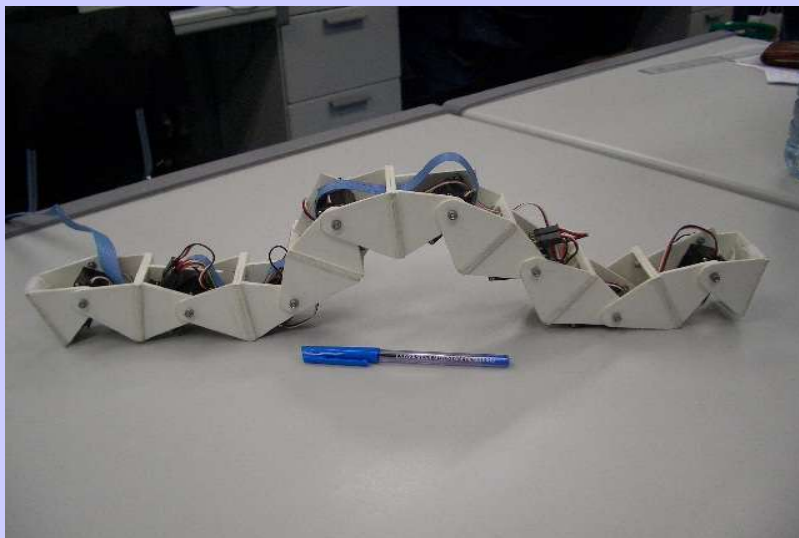
- Cube (2001)



- Cube Reloaded (2003)



- Cube Revolution (2004)

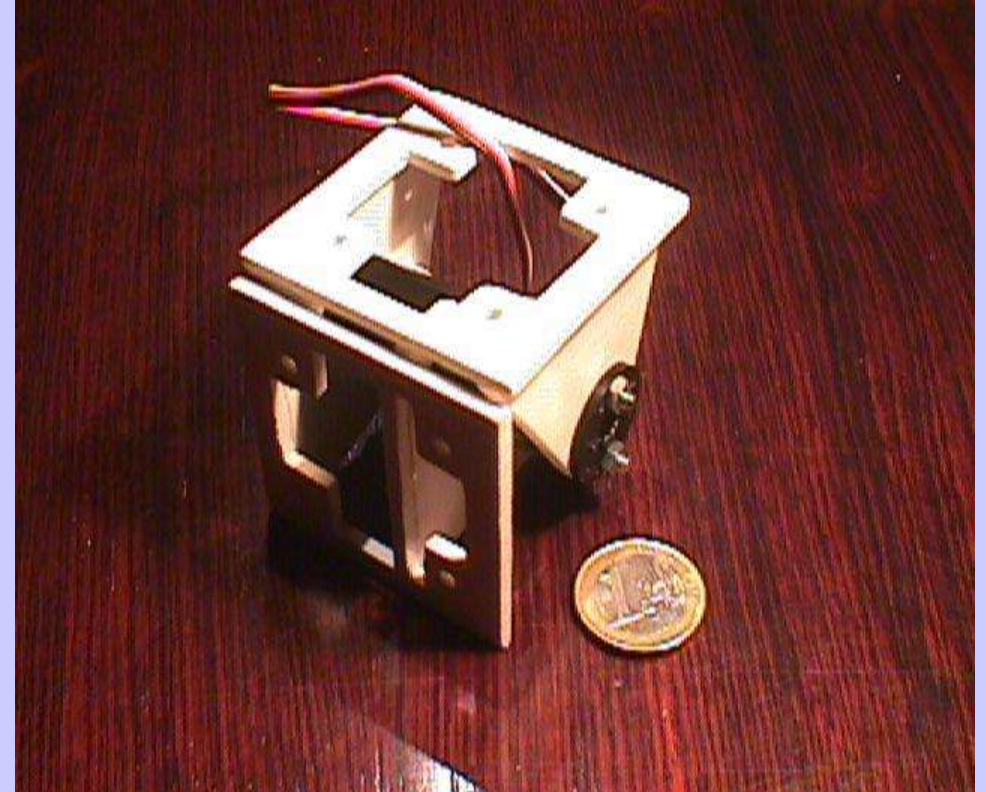
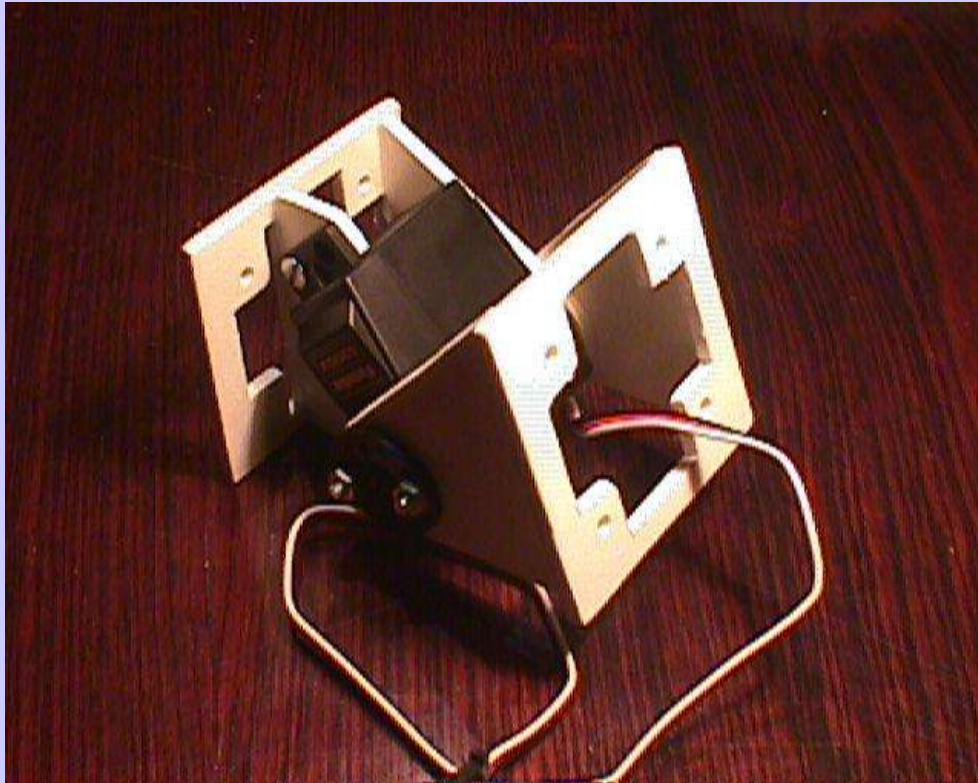


- Hypercube (200?)

Hypercube

Mecánica

Módulos Y1



- Módulos Y1
- Material: pvc expandido de 3mm
- Se utilizan servos del tipo Futaba 3003
- Conexión en fase y desfase

Vídeo

Planos

- Para "compartir" el robot, necesitamos tener los planos
- ¿Qué herramientas usar para el diseño de estos planos?
- Alternativas que se tenían:

AUTOCAD

- 
- Muy profesional
 - Muy extendido
 - Lo mejor de lo mejor....

- 
- Precio de la licencia "proporcional" a lo bueno que es...
 - Sólo Windows
 - Limitaciones a la compartición

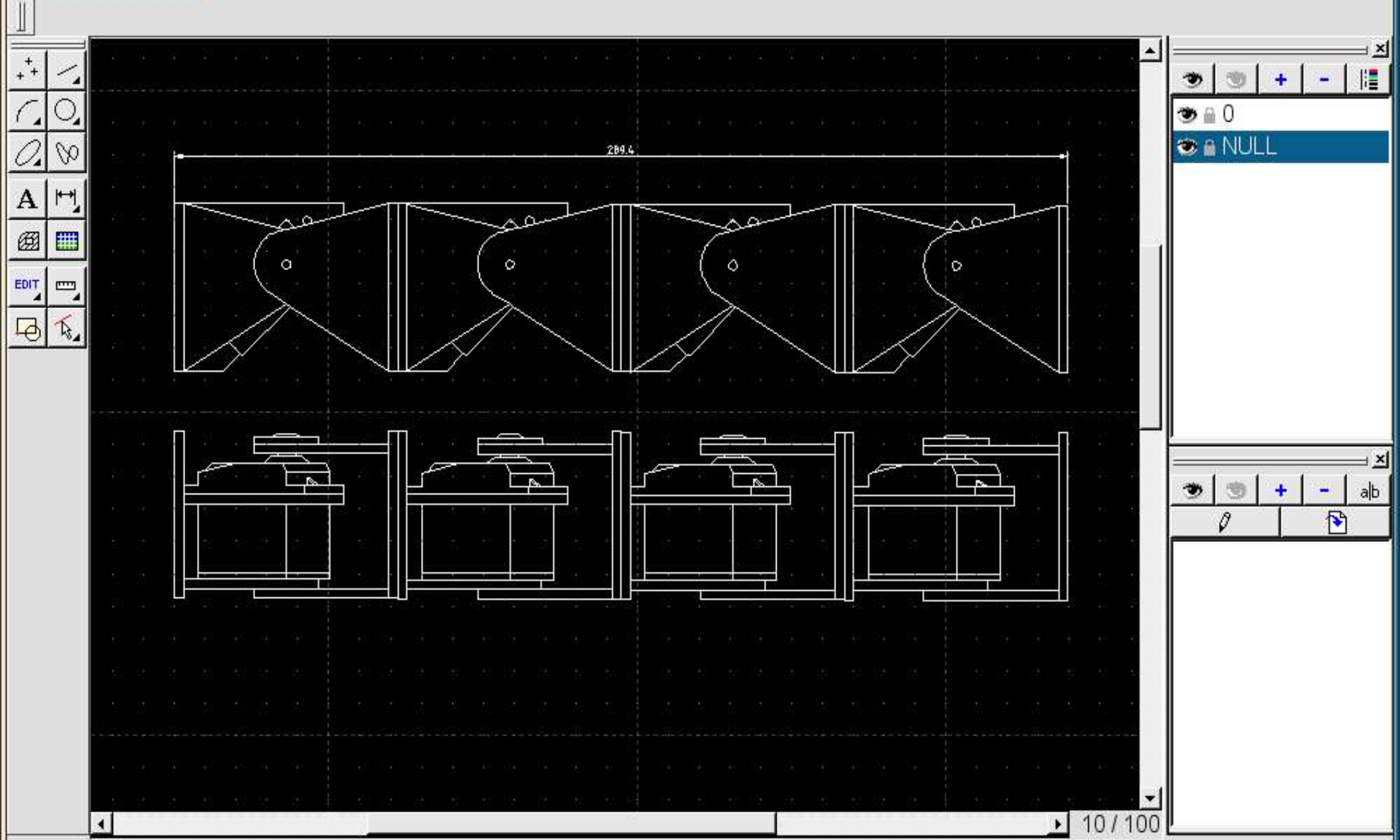
QCAD

- Poco conocido
 - No profesional
- 

- LIBRE
- 

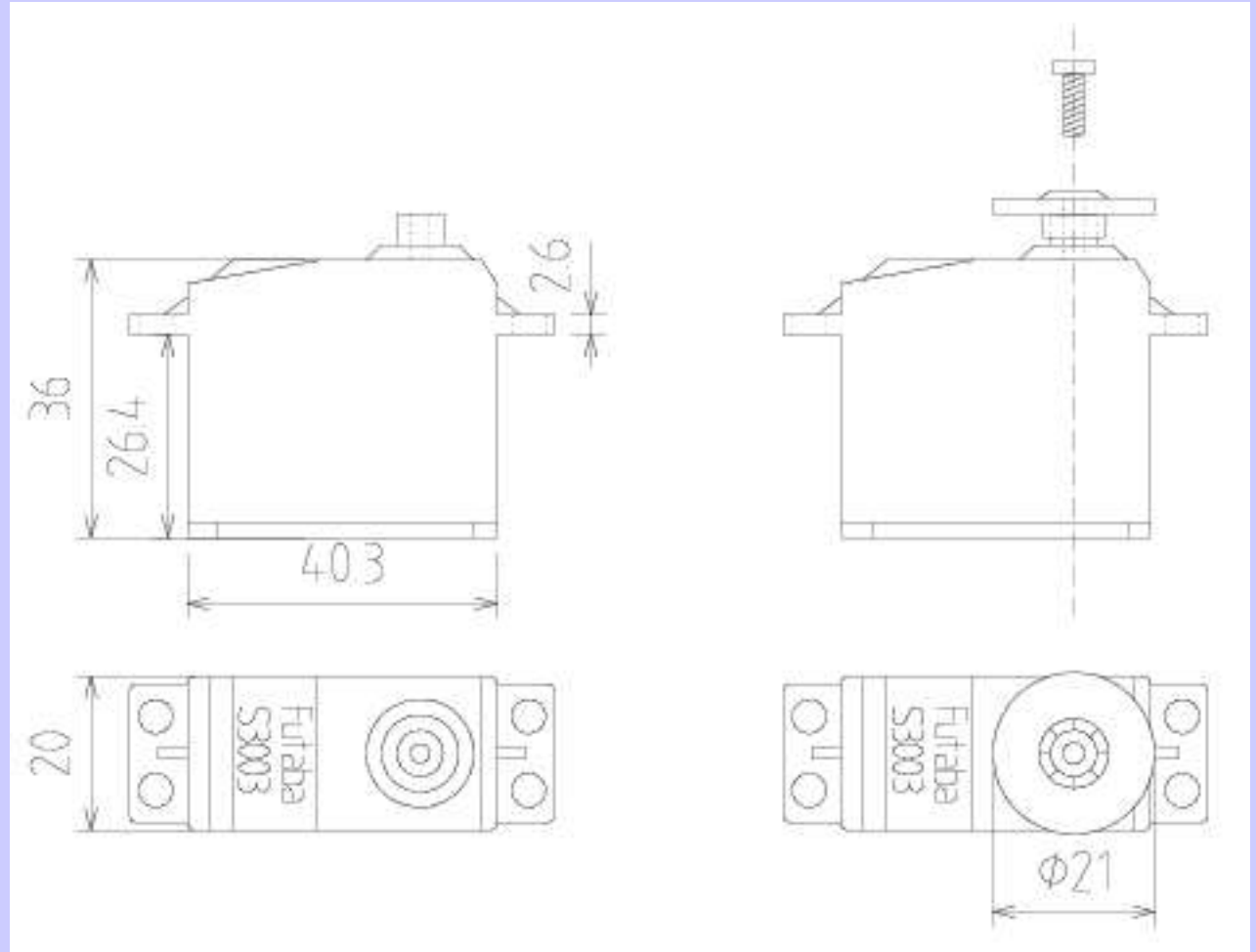


Por Capa — Por Capa Por Capa



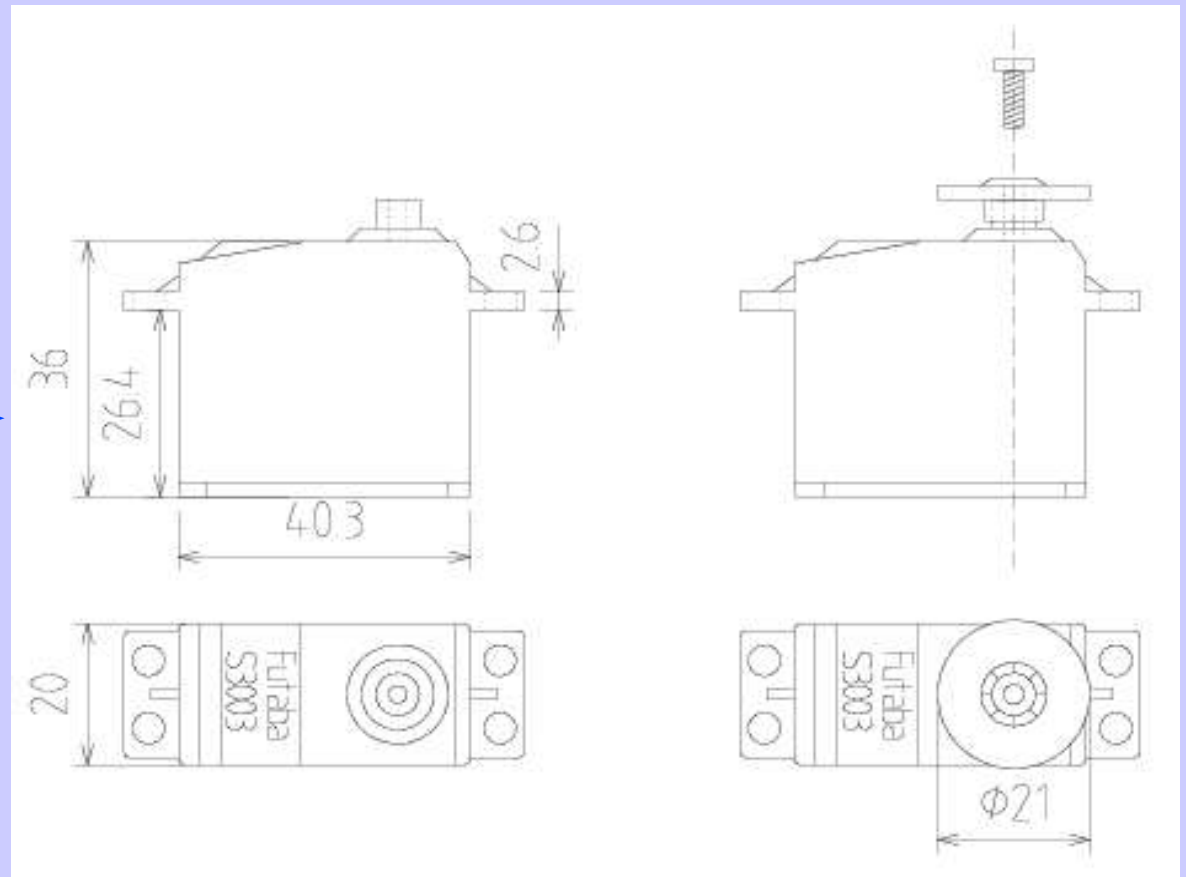
QCAD

- Con QCAD es más fácil compartir los planos
- Versión actual: 2.0.4
- Ya sí es profesional...
- Diseño en 2D
- Formato DXF (Autocad lo puede importar)
- Disponible en Debian (apt-get install qcad)



Diseño con QCAD

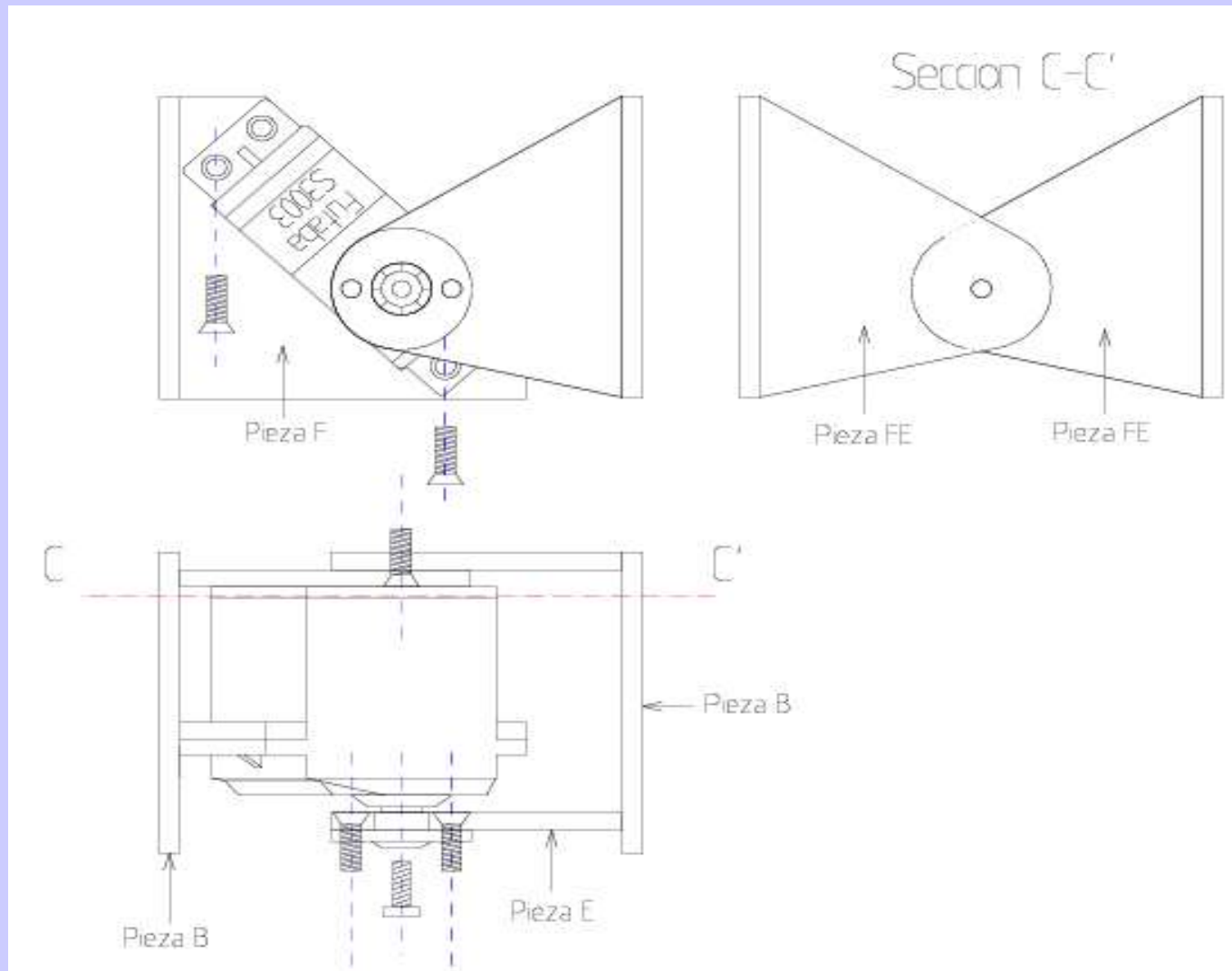
- **PASO 1:** Tener los planos del Servo Futaba 3003



- Además, así la "comunidad" dispone de los planos, para otros usos

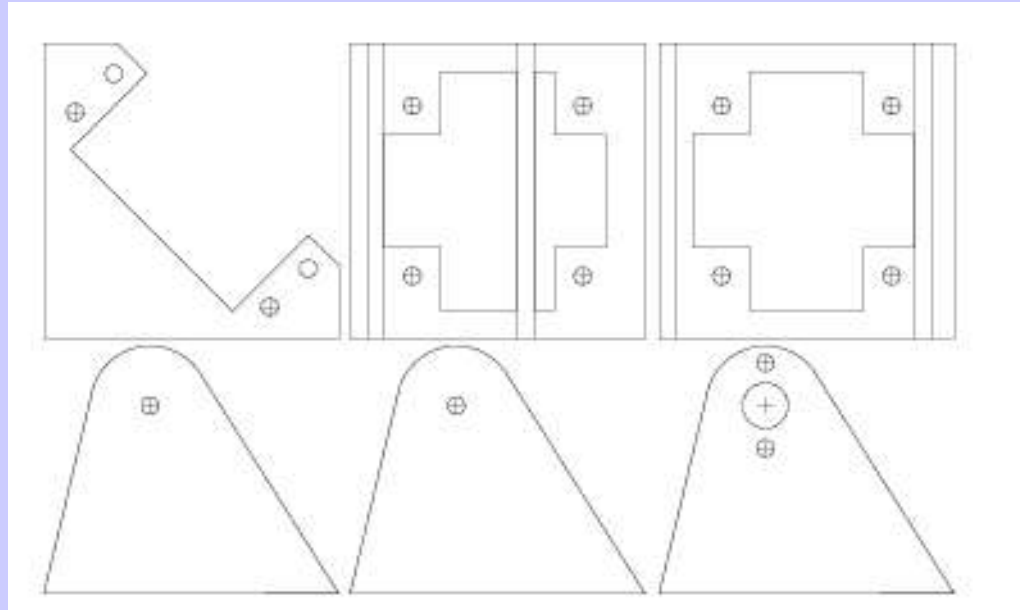
Diseño con QCAD

- **PASO 2:** Diseño del módulo



Diseño con QCAD

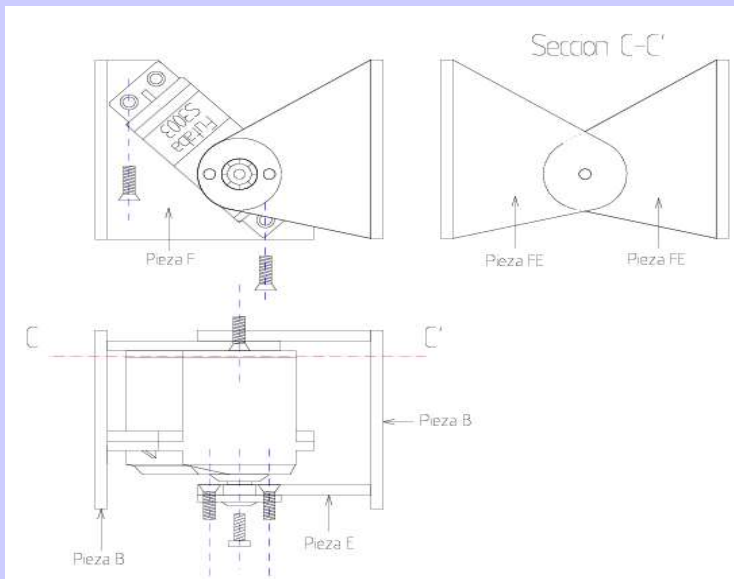
- **PASO 3:** Creación de la plantilla con las piezas



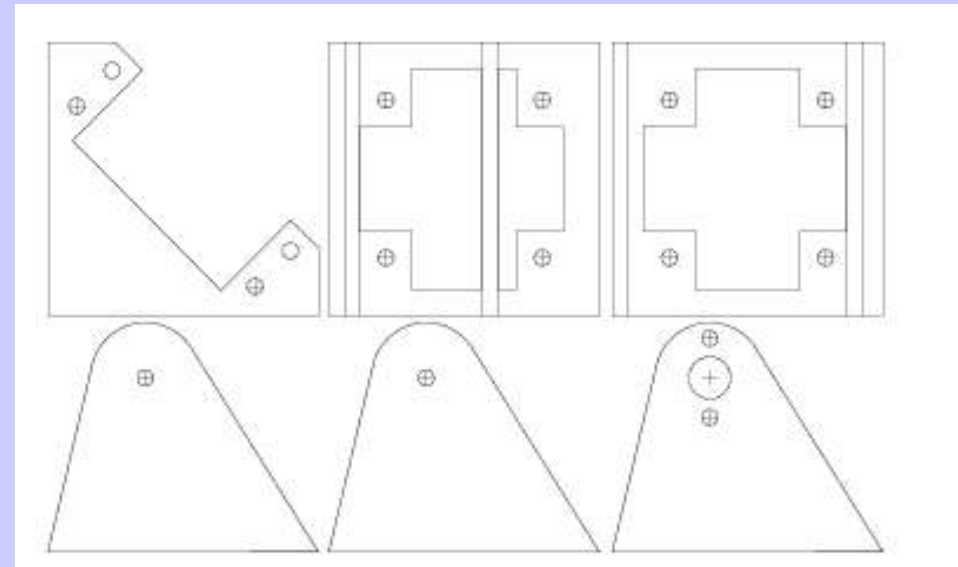
- **PASO 4:** Construcción
 - Se imprimen en papel de pegatina
 - Se pega sobre una plancha de PVC
 - Se cortan las piezas
 - Se monta el módulo

Montaje

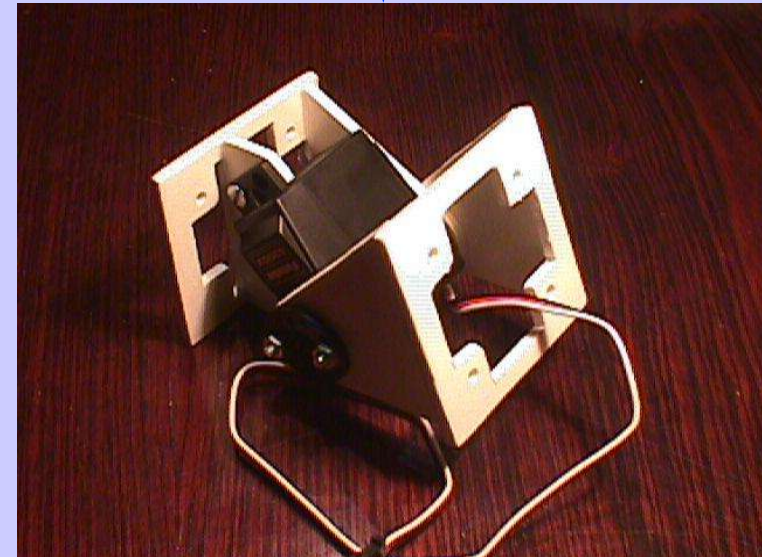
Planos



Plantillas

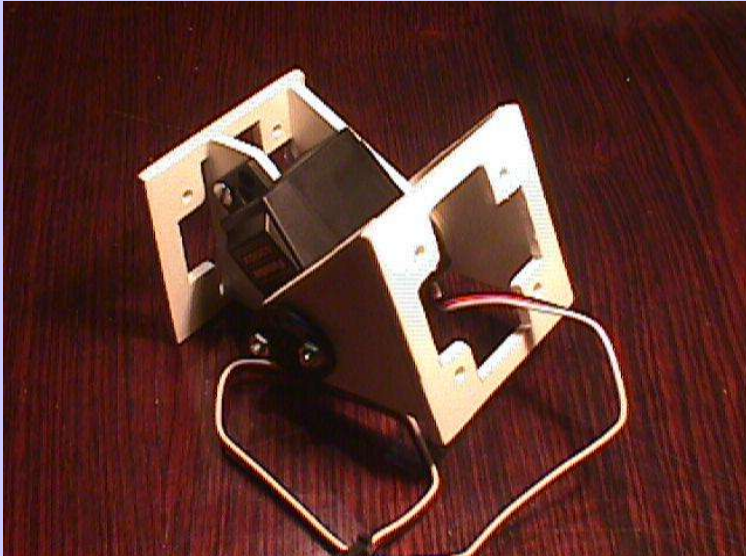


- El corte puede ser "manual"
- Para prototipos más avanzados: Corte por láser

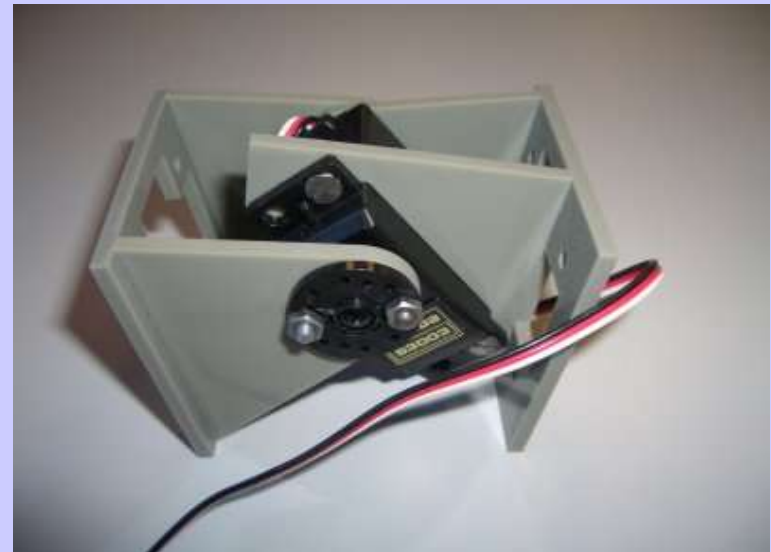


Montaje

Versión artesanal



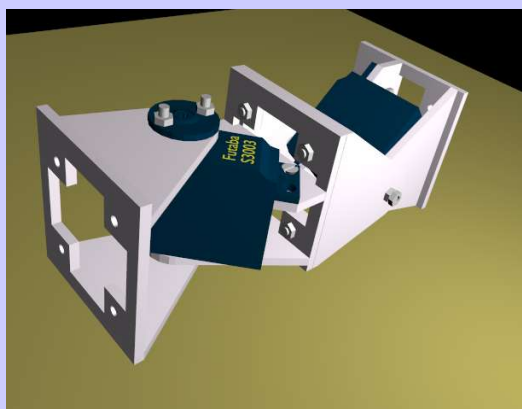
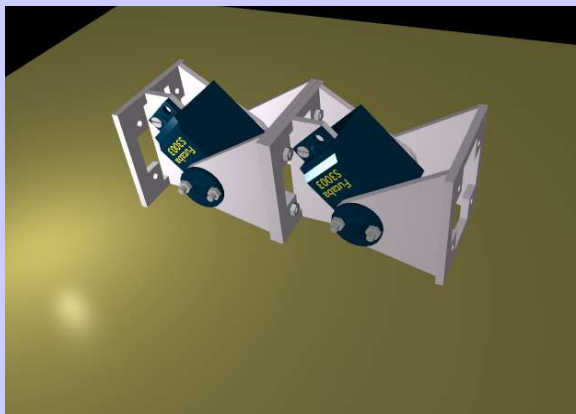
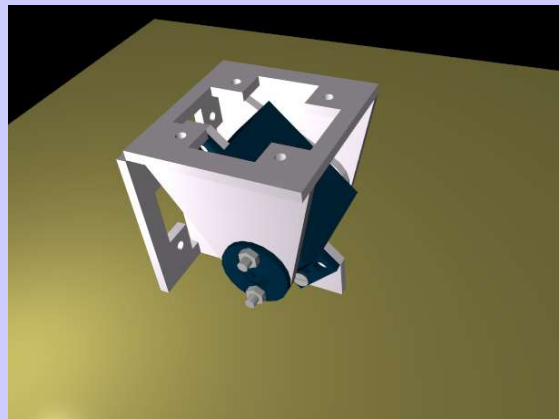
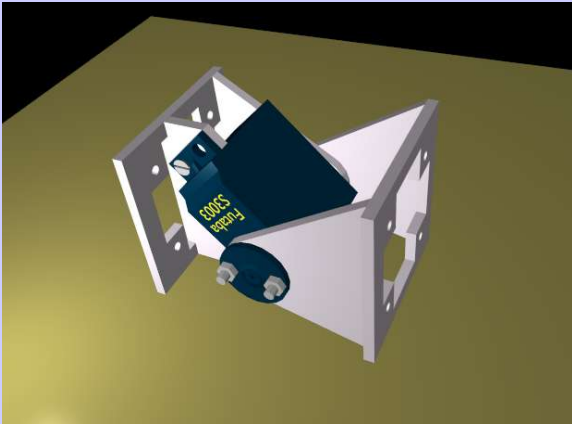
Corte por láser



Demo de QCAD

Modelo 3D

- Los planos están muy bien para la construcción
- Para hacerse una "idea" del aspecto del módulo, es mejor hacer un modelo en 3D

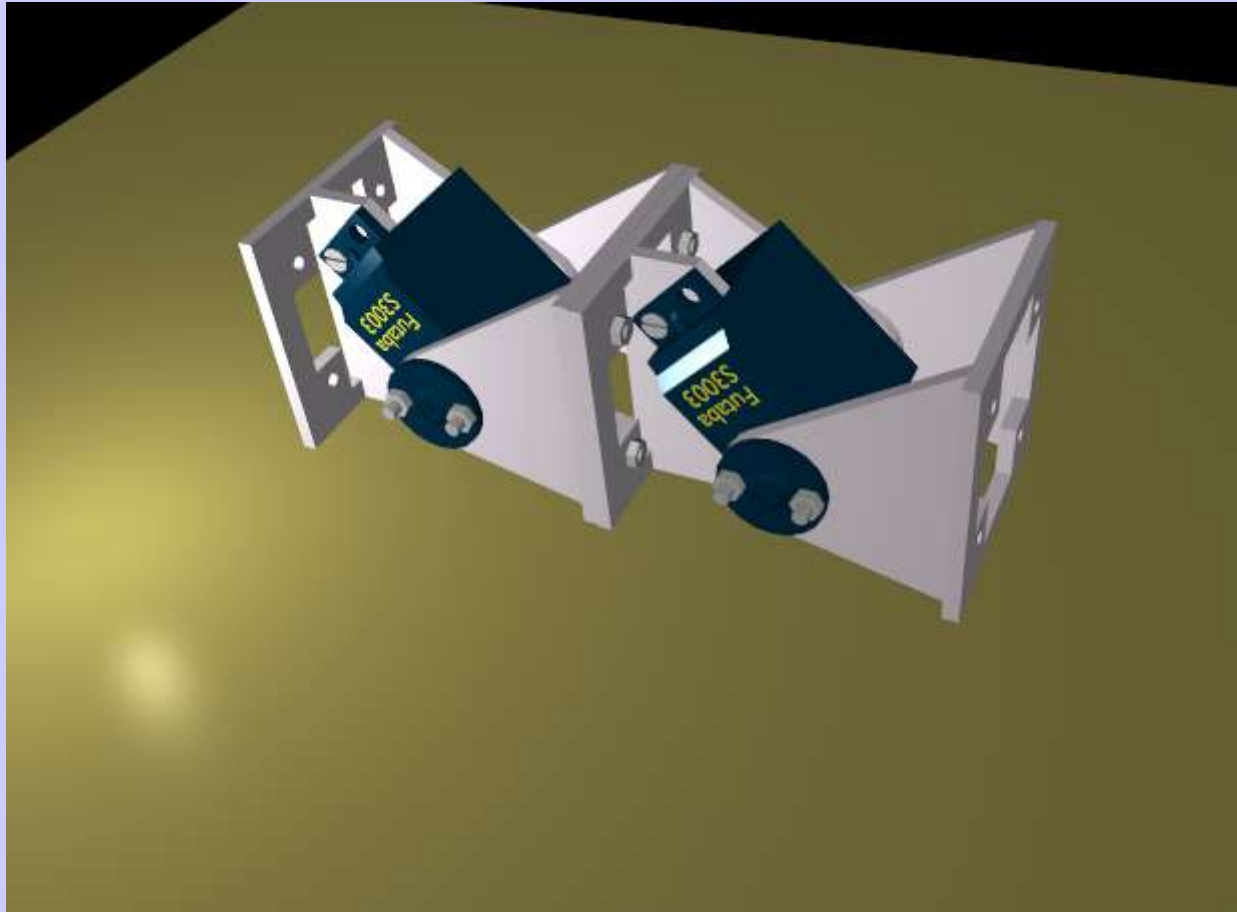


BLENDER



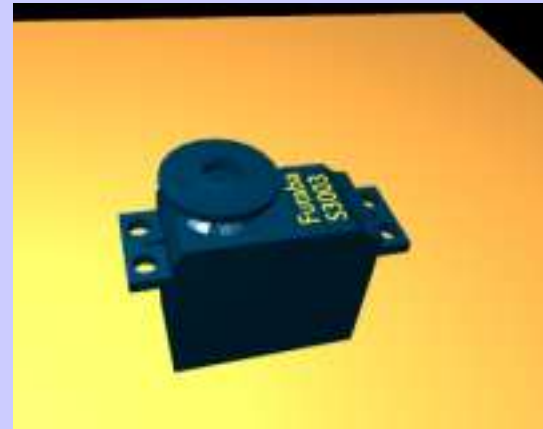
BLENDER

- Modelado 3D y animación
- Programa bajo licencia GPL
- Multiplataforma
- Disponible en Debian (`apt-get install blender`)

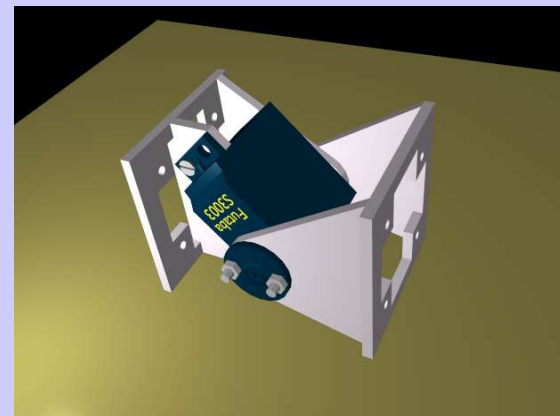
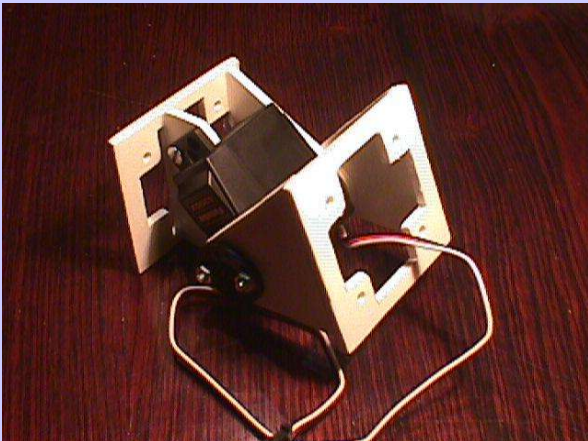


BLENDER

- Se importan las plantas de las piezas, desde QCAD y se les da altura
- Primero fue necesario crear el modelo virtual de los servos

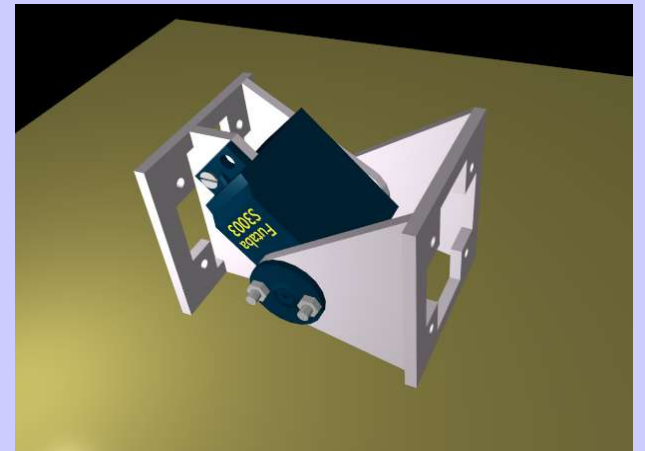
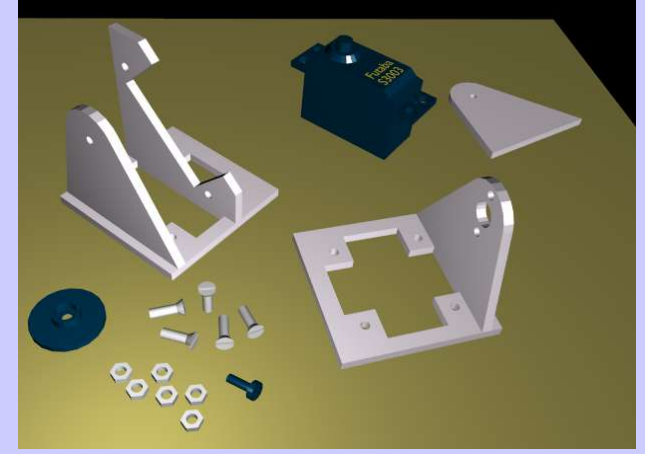
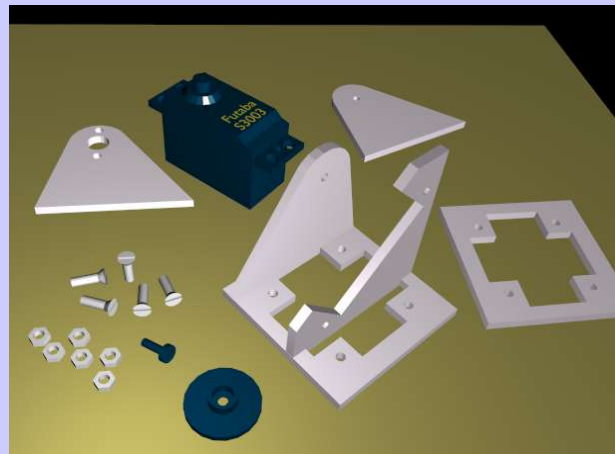
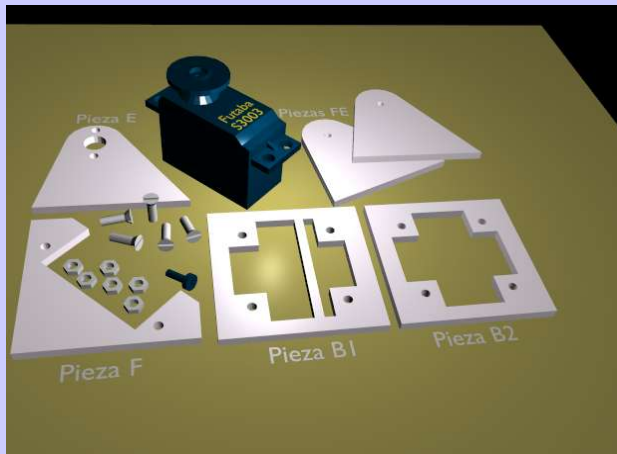


- Despues se modelaron las piezas del módulo Y1



Documentación

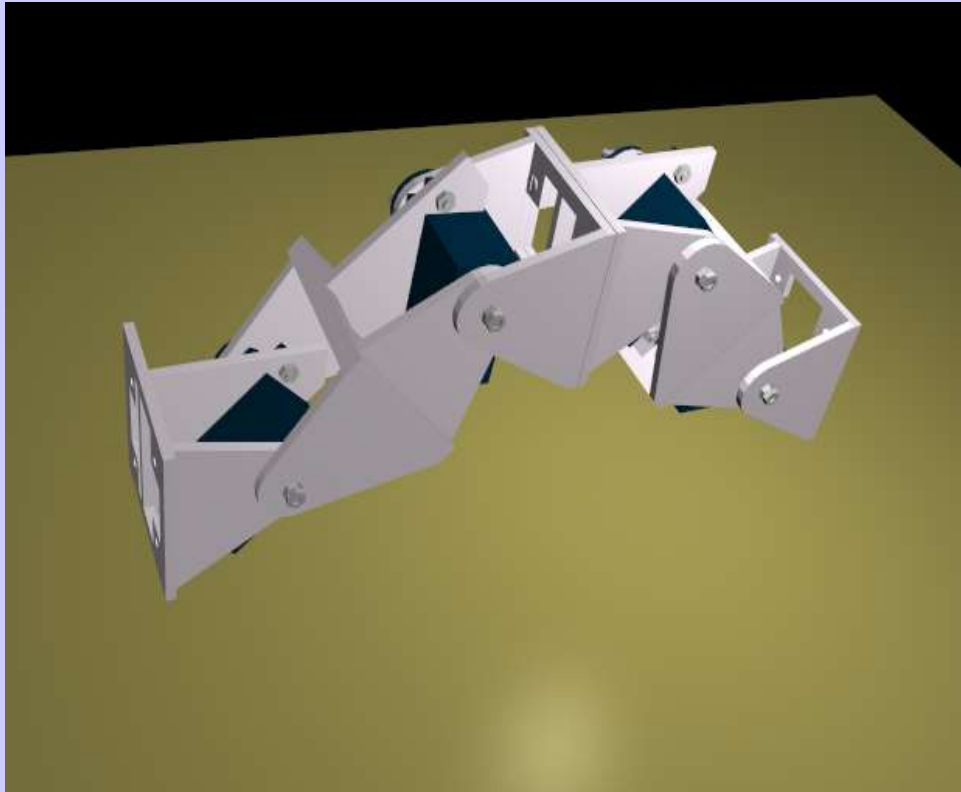
- Los modelos en 3D son muy útiles para documentar...
- Ej: Proceso de montaje de los módulos Y1



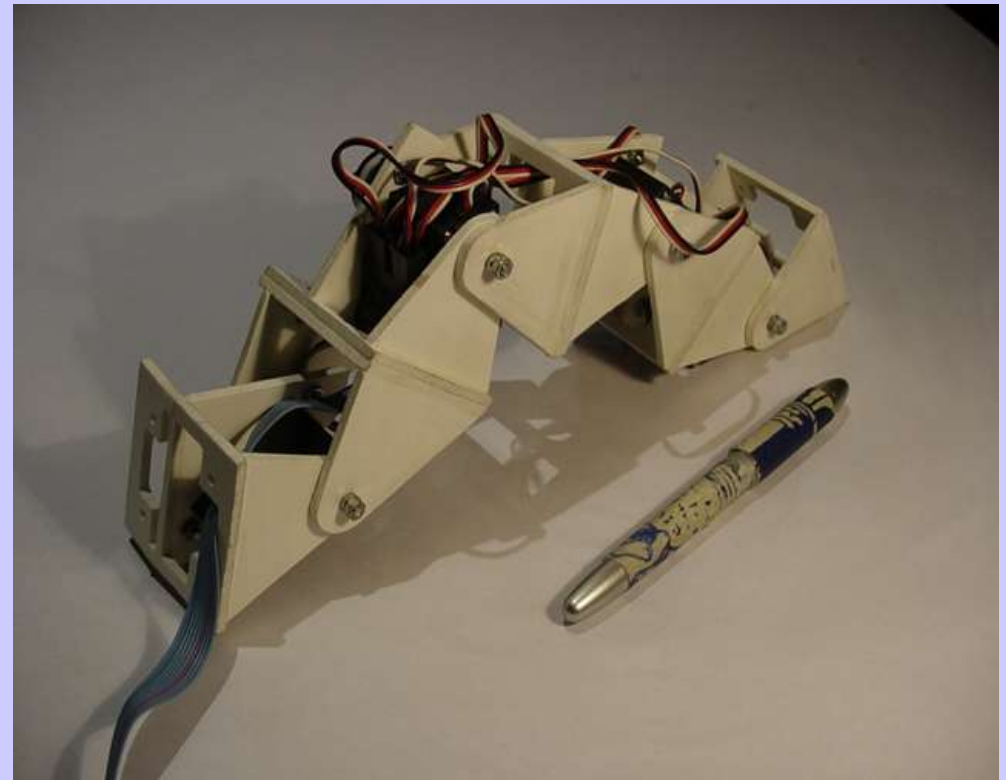
Documentación

- También resulta muy útil para ver la pinta que tendrá el robot antes de construirlo

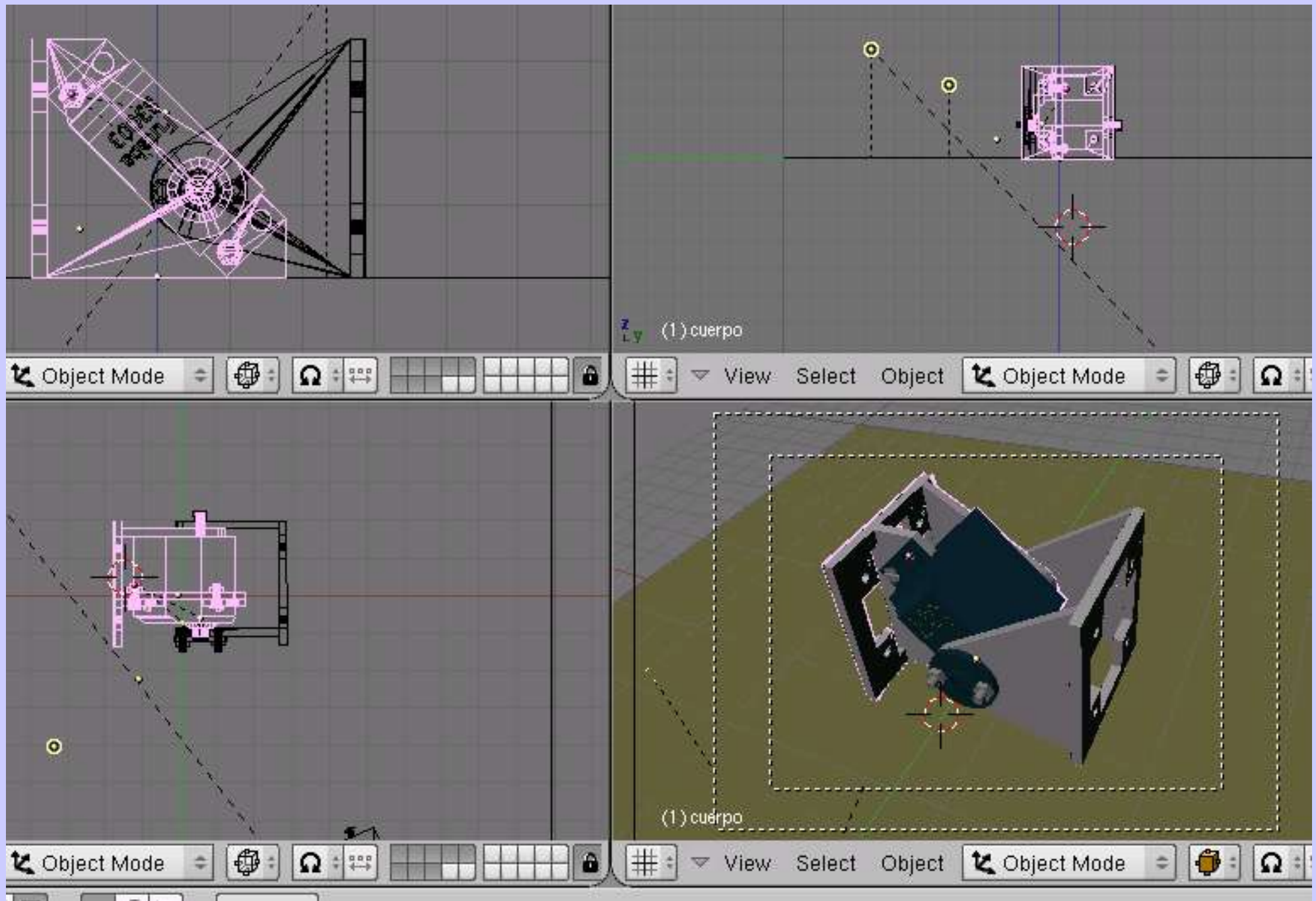
Gusano virtual



Gusano real

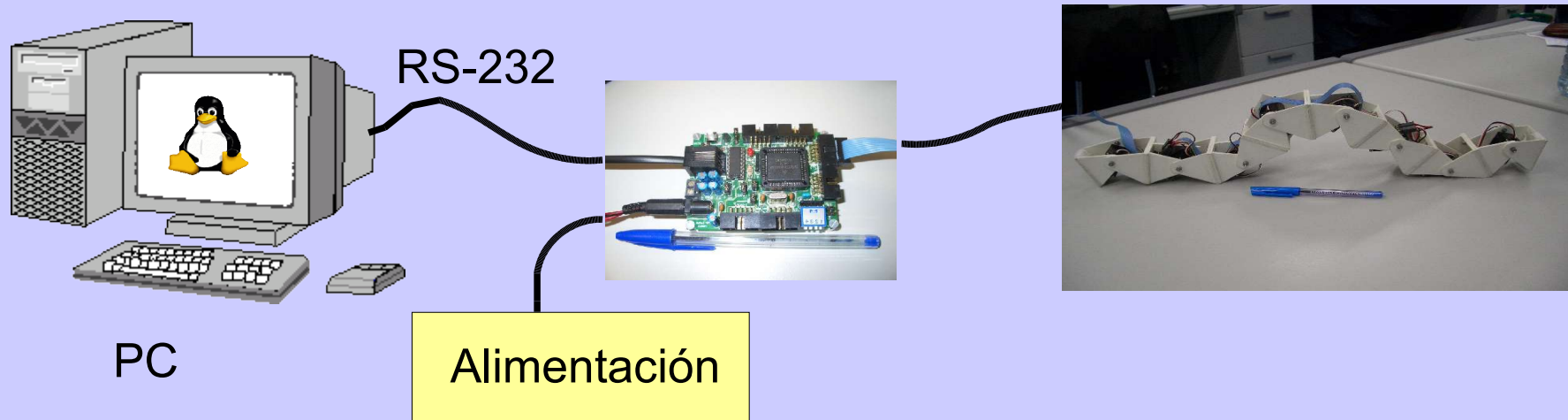


Veamos los módulos con Blender...



Electrónica

Arquitectura



- El gusano se controla desde el PC
- Lo manejamos como si fuese un periférico
- La electrónica se conecta al puerto serie por un lado y directamente a los servos por el otro
- Incluye un microcontrolador (PIC o 6811)

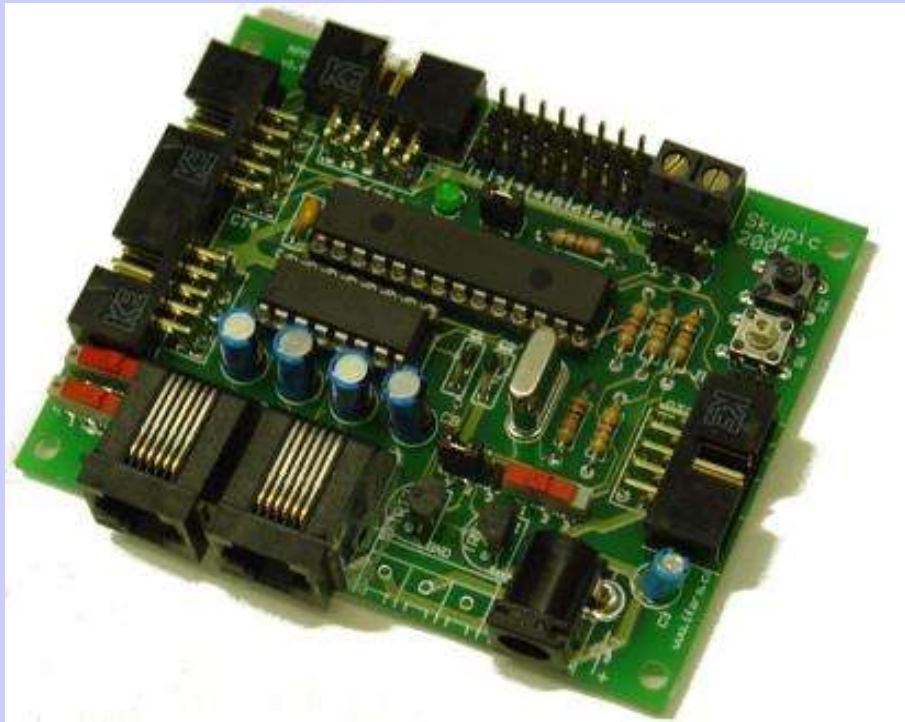
Tarjeta Skypic

- Por ejemplo se puede usar la tarjeta Skypic
- Basada en el PIC16F876
- **ES HARDWARE LIBRE**
 - Disponibles todos los esquemas
 - Se conceden permisos para su uso, modificación, distribución y fabricación.

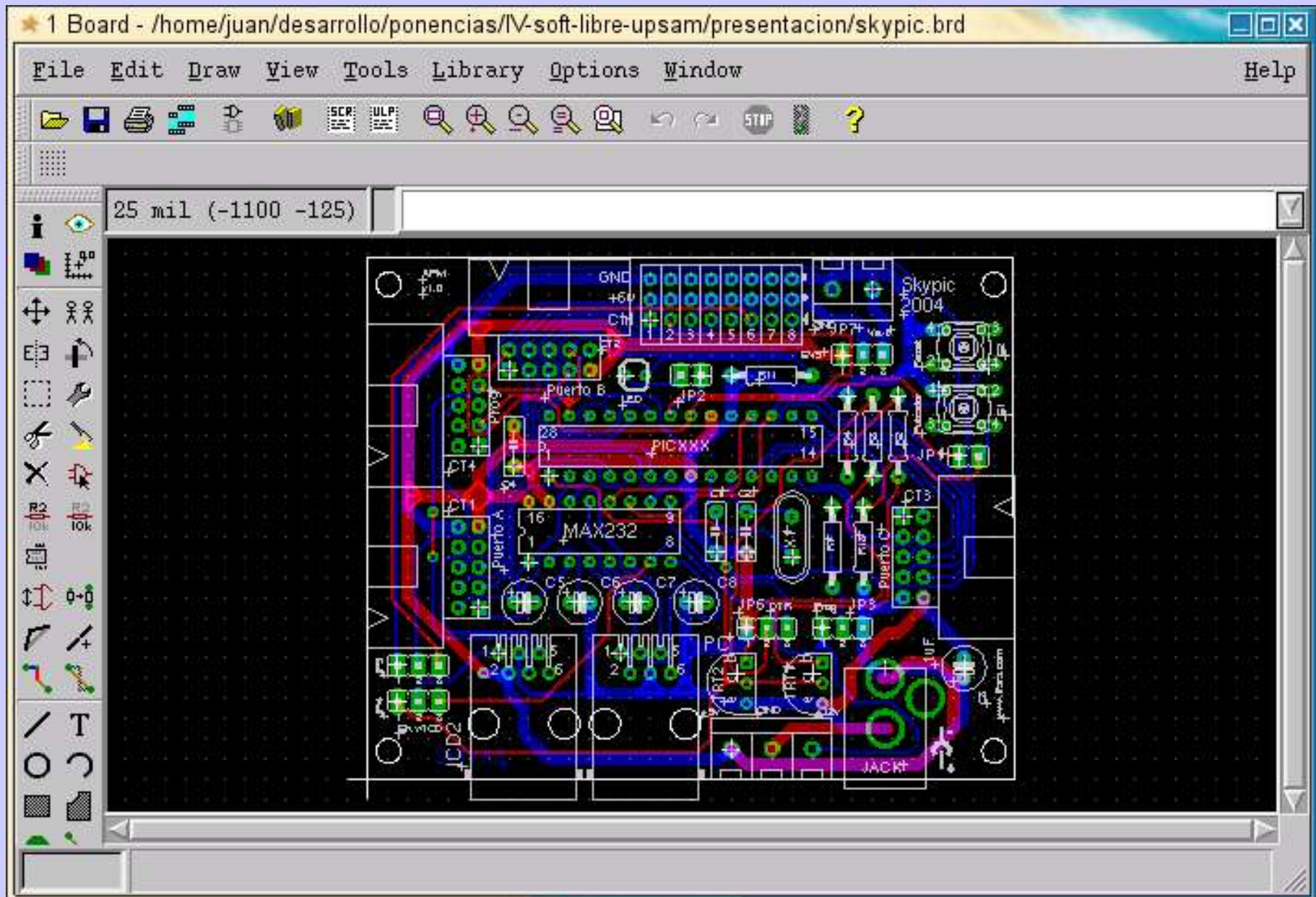


Herramienta diseño electrónico

- ¿Qué herramienta usar para diseño del hardware?
- El diseño de la Skypic se ha hecho con el **Eagle**
- **No es libre**, pero hay versión para Linux (disponible en Debian, en el repositorio non-free)



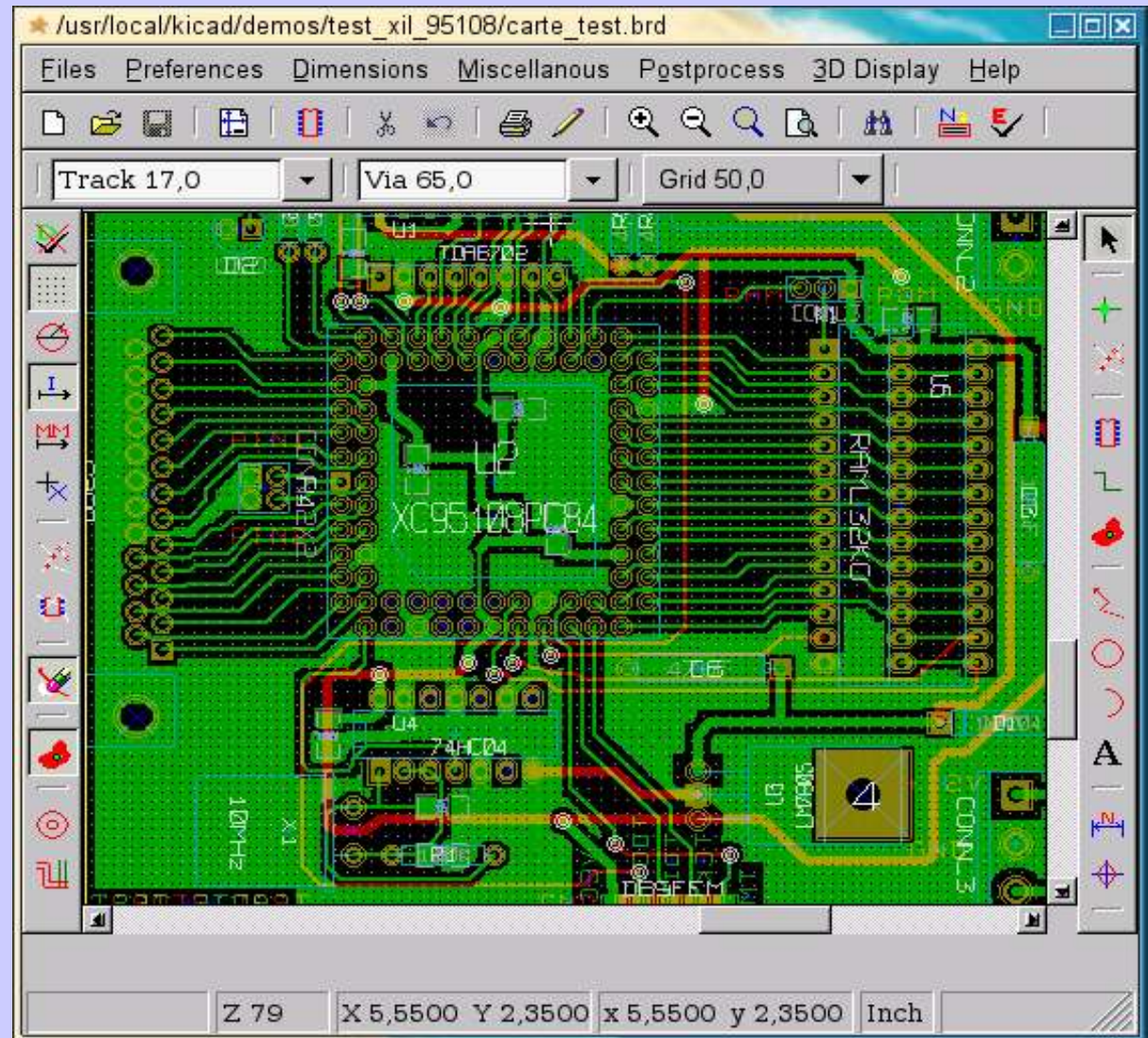
Herramienta diseño electrónico



Herramienta diseño electrónico

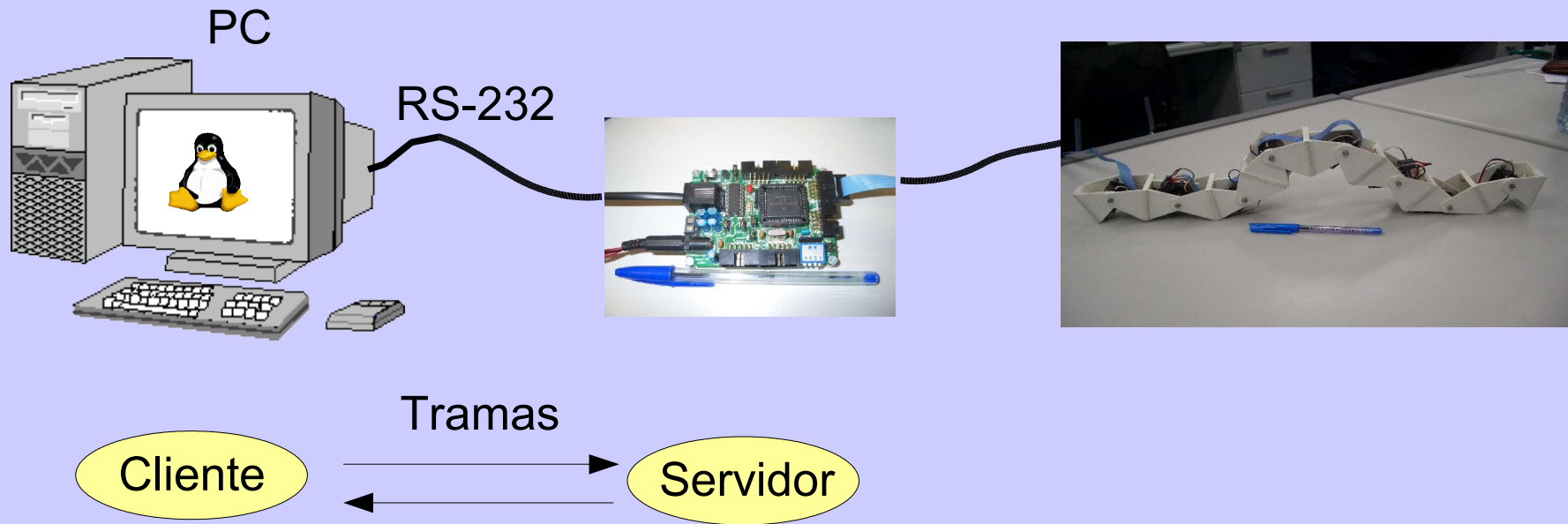
- Sin embargo, hace poco se ha liberado el programa **KICAD**
- Es totalmente profesional
- Licencia GPL!!!!!!

Demo de KICAD



Software

Arquitectura



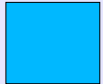
- Los servos son "periféricos" del PC
- El software se comunica con la electrónica mediante tramas que se envían por el puerto serie
- El servidor actual permite manejar hasta 8 servos. Este software NO es específico para Cube Revolutions. Se puede usar para programar cualquier robot que tenga hasta 8 servos

Tramas para manejo de servos

Cliente

Servidor

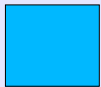
PING



PONG

• Servicio de PING

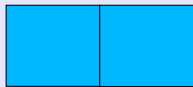
ID



RID

• Servicio de
identificación

ENABLE



• Servicio de
habilitación de servos

POS



• Servicio de
posicionamiento de
servos

API en C para posicionamiento de servos

- `sg_open()`

Abrir puerto serie

- `sg_tramas_init()`

Inicialización

- `sg_servos8_init()`

- `sg_ping()`

Servicio PING

- `sg_id()`

Servicio de identificación

- `sg_servos8_enable()`

Habilitación de servos

- `sg_servos8_pos1_grados()`

Posicionamiento

Programa "hola servo"

```
#include <stargate/stargate.h>
#include "sg-tramas-servos8.h"

int main (void)
{
    int serial_fd;  /*-- Descriptor puerto serie */

    serial_fd=sg_serial_open("/dev/ttyS0");  // Abrir puerto serie

    sg_tramas_init(serial_fd);                //-- Inicializar
    sg_servos8_init(serial_fd);

    sg_servos8_enable(0x03);                  //-- Habilitar servo 1 y 2

    sg_servos8_pos1_grados(1,90);              //-- Posicionar en un extremo
    sg_servos8_pos1_grados(2,90);

    sleep(2);

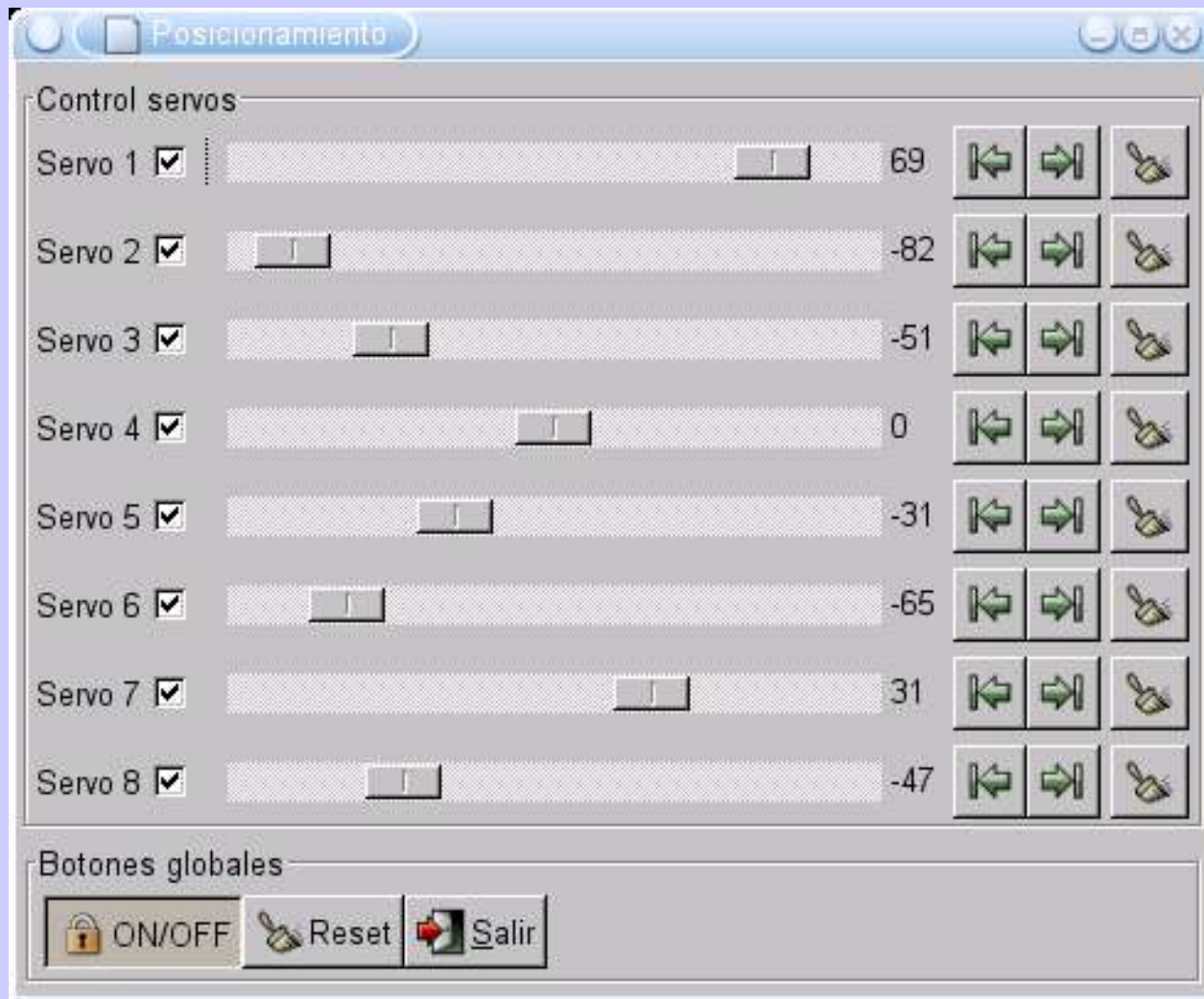
    sg_servos8_pos1_grados(1,-90);             //-- Posicionar en otro extremo
    sg_servos8_pos1_grados(2,-90);

    close(serial_fd);                          //-- Cerrar puerto serie
}
```

Programa "hola servo"

DEMO

Manejando servos desde interfaz gráfica



- La API permite establecer la posición de cualquiera de los 8 servos
- No es un software específico para CUBE: es válido para otros robots articulados

DEMO

Probando Cube



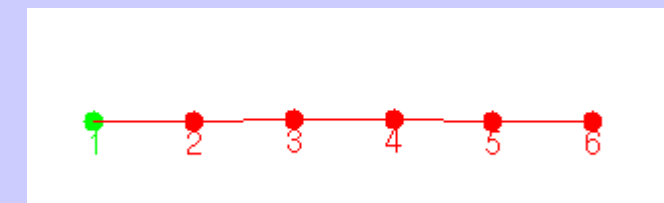
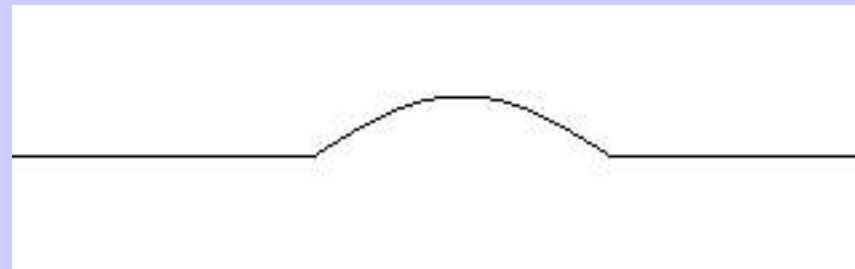
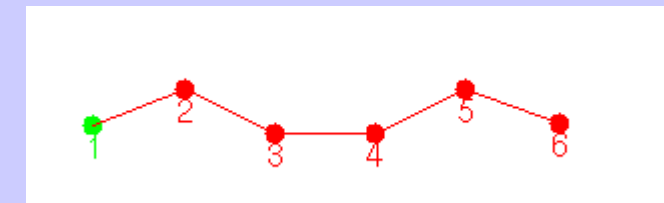
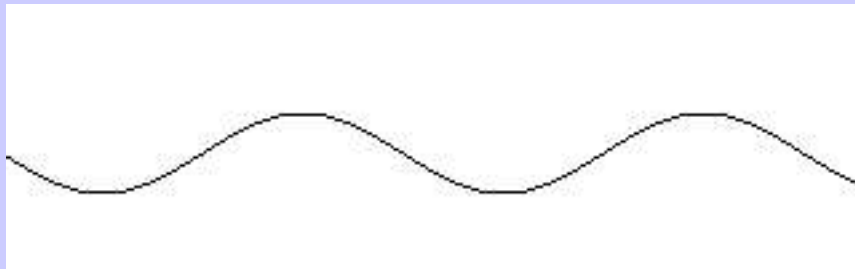
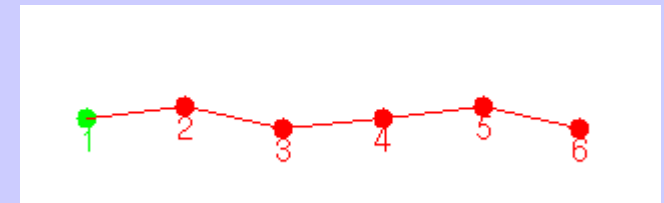
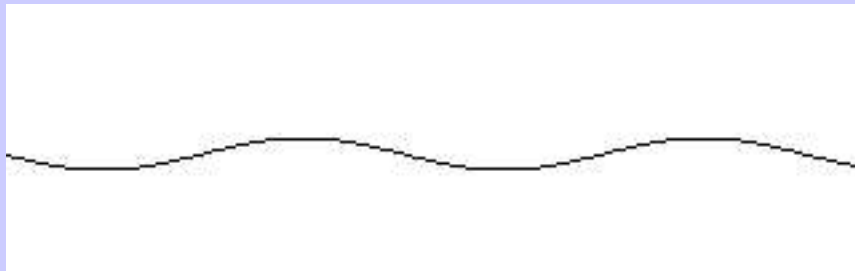
DEMO



Software para la Locomoción

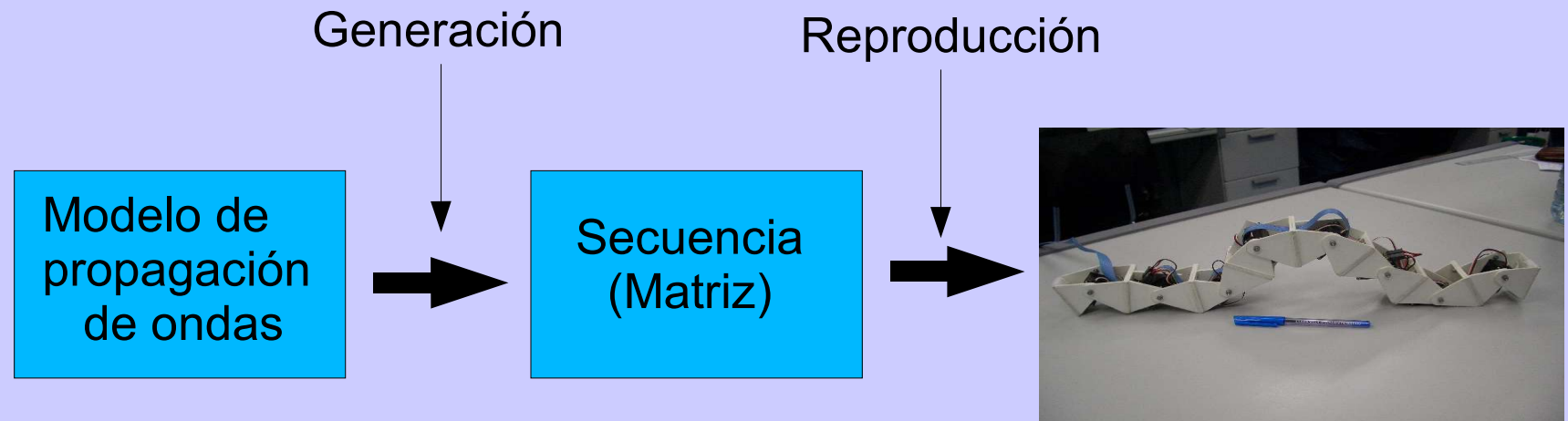
Generación del movimiento

- A partir del **tipo de onda**, **amplitud** y **longitud de onda**, se consiguen secuencias de movimiento diferentes



Locomoción

- La **locomoción** se implementa según el siguiente diagrama de bloques:



DEMO 2

Resumen del software

- MECANICA

QCAD (GPL)

BLENDER (GPL)

- ELECTRÓNICA

TARJETA SKYPIC (GPL)

EAGLE (no libre)

KICAD (GPL, futuro)

- SOFTWARE

Lenguajes: C y C#

GTK y GTK# (GPL)

Mono (GPL)

Enlaces

Algunas direcciones web son difíciles de recordar... Para acceder a toda la información hacer lo siguiente:

- 1) [Ir a Google](#)
- 2) [Teclear: "Cube Reloaded"](#)
- 3) [Pinchar en voy a tener suerte](#)

También podéis acceder directamente a la dirección:

www.iearobotics.com

Álbum de Fotos

Experimentos con un Gusano de Seda



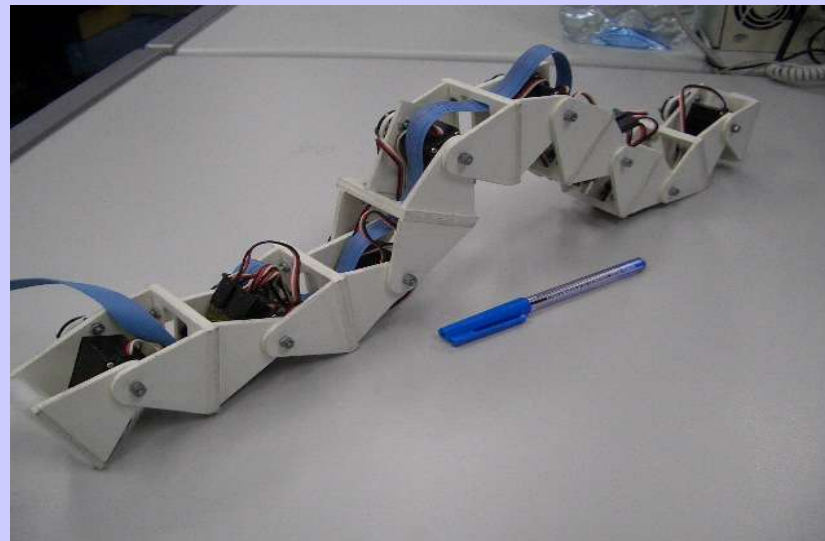
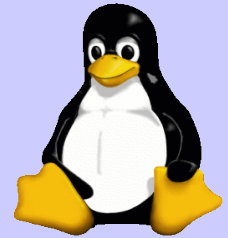
El gusano de seda descansando, una vez finalizada la jornada laboral en el laboratorio



En los congresos de Robótica hay mucho estrés...



Robótica y Linux: Cómo se hizo *Cube Revolutions*



Juan González Gómez

Escuela Politécnica Superior
Universidad Autónoma de Madrid