

Apéndice A

Listado de piezas y materiales de Cube-2.0

Todas las piezas que componen a CUBE-2.0 se listan a continuación:

- 4 servomecanismos modelo Futaba 3003.
- 4 carcasas del servo, con sus correspondientes engranajes¹. Estas piezas, junto con los servos permiten contruir una articulación con doble eje.
- Varillas roscadas de 4mm de diámetro:²
 - 8 de 4.6cm, para la unión de los servos con las carcasas del doble eje.
 - 8 de 7.5cm, para la unión del cuerpo de servo con el resto de piezas.
 - 2 varillas de 9.5cm para unir un servo con la pieza de la cabeza.
- 96 tuercas para las varillas roscadas (24 para cada servo).
- Piezas de plástico transparente de 2mm de grosos:
 - 6 de tipo A.
 - 2 de tipo B.
 - 2 de tipo C.
 - 1 base de la cola.
 - 1 base de la cabeza.
 - 1 base para la tarjeta CT6811.
 - 1 base para la tarjeta BT6811.
- 64 tornillos de 3mm de diámetro y 1cm de longitud con sus tuercas correspondientes:

¹El servomecanismo Futaba 3003 es desmontable y se pueden comprar sus piezas por separado: engranajes, carcasas, tornillos,etc.

²En total son necesarios 116cm. Estas varillas se suelen vender en unidades de 1 metro

- 4 por cada pieza tipo A. En total: $6 \times 4 = 24$.
- 6 por cada pieza tipo B: $2 \times 6 = 12$.
- 6 por cada pieza tipo C: $2 \times 6 = 12$.
- 4 para cada una de las dos bases: $2 \times 4 = 8$.
- 4 para la base de la CT6811.
- 4 para la base de la BT6811.

- 8 escuadras de 1×1.5 cm, para la unión de las bases de la cola y la cabeza.
- 1 tarjeta CT6811, basada en el microcontrolador 68hc11 de Motorola.
- 1 tarjeta BT6811, basada en el mismo microcontrolador.
- 50cm de velcro adhesivo, para unir los servos a las almohadillas y las tarjetas BT6811 y CT6811 a la estructura mecánica.
- Una almohadilla de ratón, cortada en cuadrados de 4×5 cm.
 - Son necesarios 8 cuadrados de este tipo.

Apéndice B

Características de Cube-2.0

Todas las medidas se han realizado sobre el prototipo final, con toda la electrónica y pilas acopladas.

- **Dimensiones** en estado de reposo:

- **Longitud:** 38cm
- **Anchura:** 7.5cm
- **Altura:** 6cm

- **Peso:** 700gr

- **Alimentación:**

- *Electrónica:* 4.5-6v
- *Servos:* 5-6v

- **Consumo:**

- *Electrónica:* 40mA
- *Servos:* 800mA

- **Velocidad**, en varias condiciones:

- Para $A = 20$, $\lambda = 150$, $v = 5$:

$$V_G = 3,2 \frac{cm}{s}$$

- Para $A = 10$, $\lambda = 150$, $v = 5$:

$$V_G = 1,7 \frac{cm}{s}$$

- Para $A = 10$, $\lambda = 120$, $v = 5$:

$$V_G = 1,3 \frac{cm}{s}$$

Apéndice C

Programas para el 68hc11

C.1. Programa maestro

```
*****
* Programa para el módulo maestro *
*****
* Version para la RAM interna *
*****

* --- Puertos de entrada y salida

PORTA EQU $0
PORTB EQU $04
PORTC EQU $03
DDRC EQU $07

* --- Registros del SCI

BAUD EQU $2B
SCCR1 EQU $2C
SCCR2 EQU $2D
SCSR EQU $2E
SCDR EQU $2F

* --- Registros del SPI

PORTD EQU $08
DDRD EQU $09
SPCR EQU $28
SPDR EQU $2A
SPSR EQU $29

ORG $0000 * Dirección de comienzo del programa

inicio
```

```

        LDX    #$1000    * Para acceder a los registros de control

* ---- Configuración del SCI ----

wait    BRCLR  SCSR,X $40 wait    * espero a que se estabilice el SCI

        LDAA   #$22     * Poner 22 para velocidad de 7812 baudios
        STAA  BAUD,X    * Poner 30 para velocidad de 9600 baudios

* ---- Configuración del SPI (como maestro) ----

        LDAA   #$3C
        STAA  DDRD,X    * El SPI configura las salidas automáticamente

* El maestro lo ponemos con colector cerrado.

        LDAA   #$50     * Colector cerrado
        STAA  SPCR,X    * Activa en modo maestro el SPI

* El puerto C tienen que ser de entrada
        CLRA
        STAA  DDRC,X    * Puerto C de entrada

main
        JSR   leer_car   * leo carácter por el puerto serie
        JSR   enviar_spi * mando la orden a la BT6811
        BRA  main

*.....
* . Rutina que recibe un caracter .
* . por el puerto serie .
*.....

leer_car
        BRCLR  SCSR,X $20 leer_car
        LDAA  SCDR,X
        RTS

*.....
* . Rutina que envia un caracter .
* . por el puerto serie .
*.....

enviar_car
        BRCLR  SCSR,X $80 enviar_car
        STAA  SCDR,X
        RTS

*.....

```

```

*. Rutina que envía un dato por el SPI .
*.....
*. Si se quiere pausa entre datos      .
*. enviados llamar a: enviar_spi_pausa .
*.                                     .
*. Si no se quiere pausa entre datos   .
*. enviados llamar a: enviar_spi       .
*.....

enviar_spi_pausa
    PSHY                * pausa entre datos
    LDY    #$18         * valor de 150 microseg
pausa  DEY
    CPY    #$0
    BNE   pausa
    PULY

enviar_spi
    LDAB   PORTD,X
    ANDB   #$DF
    STAB   PORTD,X      * Mandamos activarse al esclavo

    STAA   SPDR,X      * introduzco el dato a mandar
espera  BRCLR SPSR,X $80 espera * ¿transmisión efectuada?

    LDAB   PORTD,X
    ORB    #$20
    STAB   PORTD,X      * Desactivamos al esclavo

    RTS

*.....
*. Rutina de pausa      .
*.....

esperar
    PSHY
    LDY    #$C400
sigue  DEY
    CPY    #$0
    BNE   sigue
    PULY
    RTS

    END

```

C.2. Programa esclavo

```

* .....
* . FUTABA. Version 1.0 .....
* .....
* . Programa para la gestion de 4 servomecanismos Futaba S3003 .....
* .....

*****
* Identificador del módulo esclavo *
*****

M_ES equ 'a'

*****
* Registros y Puertos del 68hc11 *
*****

* Puertos del 68hc11

PORTA equ $00
PORTB equ $04
PORTC equ $03
DDRC equ $07

* Registros del SPI

PORTD EQU $08
DDRD EQU $09
SPCR EQU $28
SPDR EQU $2A
SPSR EQU $29

* Registros de los comparadores

TCNT equ $0E ; valor del temporizador principal
TMSK1 equ $22
TFLG1 equ $23
TCTL1 equ $20
TOC1 equ $16 ; este registro es de 16 bits
TOC2 equ $18 ; este registro es de 16 bits
TOC3 equ $1A ; este registro es de 16 bits
TOC4 equ $1C ; este registro es de 16 bits
TOC5 equ $1E ; este registro es de 16 bits

*****
* Constantes del programa *
*****

```

```

PERIODO EQU 42000 ; Periodo de la señal cuadrada 21 mseg
EXTREMO1 EQU 600
CENTRO EQU 2600
EXTREMO2 EQU 4300
    
```

```

*****
* Máscaras de acceso *
*****
    
```

```

OC1 equ $80 ; Comparador 1
OC2 equ $40 ; Comparador 2
OC3 equ $20 ; Comparador 3
OC4 equ $10 ; Comparador 4
OC5 equ $08 ; Comparador 5
    
```

```

FUT1 equ $01 ; Bit donde se encuentra el FUTABA 1
FUT2 equ $02 ; Bit donde se encuentra el FUTABA 2
FUT3 equ $04 ; Bit donde se encuentra el FUTABA 3
FUT4 equ $08 ; Bit donde se encuentra el FUTABA 4
LED equ $10 ; Bit donde se encuentra el LED
    
```

```

*****
* Variables del programa *
*****
    
```

```

ORG $0
    
```

```

posi1 RMB 2
posi2 RMB 2
posi3 RMB 2
posi4 RMB 2
salidas RMB 1 ; máscara que indica las salidas hardware activas
    
```

```

*****
* ----- *
* | PROGRAMA PRINCIPAL | *
* ----- *
*****
    
```

```

ORG $F800 ; Programa para el microcontrolador E2
    
```

```

*****
* Inicialización del programa *
*****
    
```

```

inicializar
LDS #$FF ; inicializo el puntero de pila
LDX #$1000 ; para acceder a los registros de control internos
    
```



```

* --- Configuración del SPI (como esclavo) ----

    LDAA  #$3C      * SS de entrada
    STAA  DDRD,X   * El SPI configura las salidas automáticamente

    LDAA  #$40      * Colector cerrado
    STAA  SPCR,X   * Activa en modo esclavo el SPI

* --- Configuración del Puerto C como salida

    LDAA  #$FF
    STAA  DDRC,X

* --- Configuración de los comparadores ---

    CLRA
    STAA  TCTL1,X   ; Comparadores sin salida hardware
    LDAA  #OCL      ; Activo el comparador 1
    STAA  TMSK1,X

* --- Configuración de las salidas ---

    LDD   #2000
    STD   posi1
    STD   posi2
    STD   posi3
    STD   posi4

    CLRA                      ; indico que todos los motores activos
    STAA  salidas

    CLI                        ; permito las interrupciones

    LDAA  #$FF                ; @@
    STAA  PORTB,X             ; @@

*****
* BUCLE PRINCIPAL *
*****

inicio
    BSR  recibir_spi
    CMPA #M_ES                ; verifica que la trama sea para él
    BEQ  analizar
    BSR  recibir_spi
    BSR  recibir_spi
    BSR  recibir_spi
    BRA  inicio

analizar BSR  recibir_spi      ; leo dos caracteres por el puerto serie SPI
        CMPA #'1'

```



```

BSR recibir_spi ; Leer nºmotor ( $1, $2, $3, $4 )

* Si se llega aquí es que se ha solicitado servicio de posicionamiento
* Meter en Y el número (motor-1)*2
DECA ; A=motor-1
LSLA ; A=(motor-1)*2
TAB
CLRA
XGDY ; Y=(motor-1)*2

BSR recibir_spi ; Leer posición (0-255)
CLRB
LSRD
LSRD
LSRD
LSRD
ADDD #EXTREMO1 * D=(posición*16+EXTREMO1)
SEI
STD 0,Y * Almacenar posición
CLI
BRA inicio

```

```

* .....
* . Rutina que recibe un dato por el SPI .
* . La rutina espera hasta recibir el dato .
* . Entradas: Ninguna .
* . Salidas: El acumulador A contiene el dato .
* .....

```

```

recibir_spi
espera BRCLR SPSR,X $80 espera ; espero que el dato se haya recibido
LDAA SPDR,X
RTS

```

```

*****
* Rutinas de interrupción de los COMPARADORES *
*****

```

```

* .....
* . Comparador 1: Establece el principio de los pulsos .
* .....
* . Podemos poner BSET porque las interrupciones de los .
* . otros comparadores están desactivadas. En caso .
* . contrario habría que utilizar load y store. .
* .....

```

```

int_oc1
BSET TFLG1,X OC1 ; Poner a cero el flag de interrupción

LDD TCNT,X ; Actualizar comparador 1
ADDD #PERIODO ; próxima interrupción = ...

```

```

STD TOC1,X      ; ... tiempo_actual(TCNT) + Periodo

LDAA PORTB,X
ORA  salidas
ANDA #$18      ; Si salida 4 habilitada ponerla a
STAA PORTB,X   ; nivel alto
LDD  TCNT,X    ; Actualizar comparador 5 (FUTABA 4)
ADDD posi4     ; Establecer tiempo anchura del pulso
STD  TOC5,X

LDAA PORTB,X
ORA  salidas
ANDA #$1C      ; Si salida 3 habilitada ponerla a
STAA PORTB,X   ; nivel alto
LDD  TCNT,X    ; Actualizar comparador 4 (FUTABA 3)
ADDD posi3     ; Establecer tiempo anchura del pulso
STD  TOC4,X

LDAA PORTB,X
ORA  salidas
ANDA #$1E      ; Si salida 2 habilitada ponerla a
STAA PORTB,X   ; nivel alto
LDD  TCNT,X    ; Actualizar comparador 3 (FUTABA 2)
ADDD posi2     ; Establecer tiempo anchura del pulso
STD  TOC3,X

LDAA PORTB,X
ORA  salidas
ANDA #$1F      ; Si salida 1 habilitada ponerla a
STAA PORTB,X   ; nivel alto
LDD  TCNT,X    ; Actualizar comparador 2 (FUTABA 1)
ADDD posi1     ; Establecer tiempo anchura del pulso
STD  TOC2,X
    
```

* Activar las interrupciones de los comparadores de las salidas

```
BSET TMSK1,X $78
```

```
RTI
```

```

* .....
* . ¡OJO! No podemos poner BSET porque podría desactivar las otras .
* . interrupciones de los comparadores antes de ser atendidas .
* .....
* . Rutina de servicio de interrupcion del COMPARADOR 2 .
* . POSICION DEL FUTABA 1 .
* .....
    
```

int_oc2

```

LDAA #OC2      ; pongo a cero el flag de interrupción
STAA TFLG1,X
    
```

```

        BCLR PORTB,X FUT1 ; pongo a cero la salida hardware correspondiente
        BCLR TMSK1,X OC2  ; Deshabilitar interrupción del comparador
        RTI

* .....
* . Rutina de servicio de interrupcion del COMPARADOR 3 .
* . POSICION DEL FUTABA 2 .
* .....

int_oc3
        LDAA #OC3 ; pongo a cero el flag de interrupción
        STAA TFLG1,X

        BCLR PORTB,X FUT2 ; pongo a cero la salida hardware correspondiente
        BCLR TMSK1,X OC3  ; Deshabilitar interrupción del comparador
        RTI

* .....
* . Rutina de servicio de interrupcion del COMPARADOR 4 .
* . POSICION DEL FUTABA 3 .
* .....

int_oc4
        LDAA #OC4 ; pongo a cero el flag de interrupción
        STAA TFLG1,X

        BCLR PORTB,X FUT3 ; pongo a cero la salida hardware correspondiente
        BCLR TMSK1,X OC4  ; Deshabilitar interrupción del comparador
        RTI

* .....
* . Rutina de servicio de interrupcion del COMPARADOR 5 .
* . POSICION DEL FUTABA 4 .
* .....

int_oc5
        LDAA #OC5 ; pongo a cero el flag de interrupción
        STAA TFLG1,X

        BCLR PORTB,X FUT4 ; pongo a cero la salida hardware correspondiente
        BCLR TMSK1,X OC5  ; Deshabilitar interrupción del comparador
        RTI

*****
* Inicializacion de los vectores de interrupcion *
*****

        ORG $FFE0 ; coloco vector de interrupcion del comparador 5
v_comp5 FDB #int_oc5 ; FUTABA 4
    
```

```

v_comp4  ORG $FFE2      * coloco vector de interrupcion del comparador 4
         FDB #int_oc4  * FUTABA 3

v_comp3  ORG $FFE4      * coloco vector de interrupcion del comparador 3
         FDB #int_oc3  * FUTABA 2

v_comp2  ORG $FFE6      * coloco vector de interrupcion del comparador 2
         FDB #int_oc2  * FUTABA 1

v_comp1  ORG $FFE8      * coloco vector de interrupcion del comparador 1
         FDB #int_oc1

v_reset  ORG $FFFE      * coloco vector de interrupcion del reset
         FDB #inicializar

        END
```

Apéndice D

Álbum de fotos

CUBE-1.0

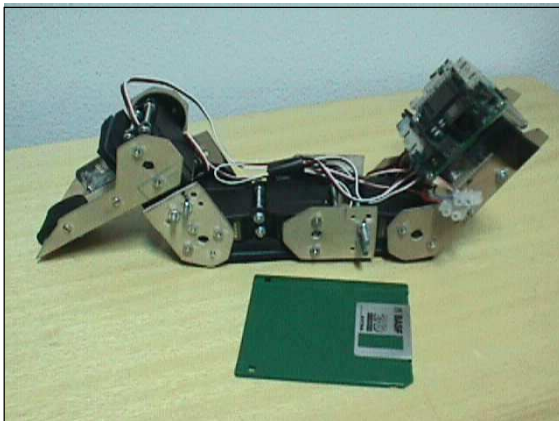


Foto 1: Cube-1.0 sobre una mesa

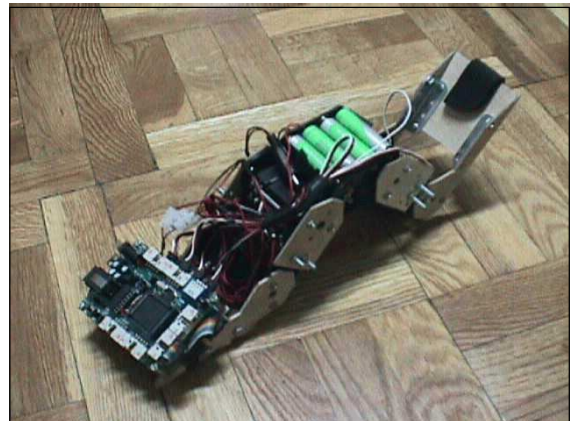


Foto 2: Cube 1.0 desplazándose sobre el suelo

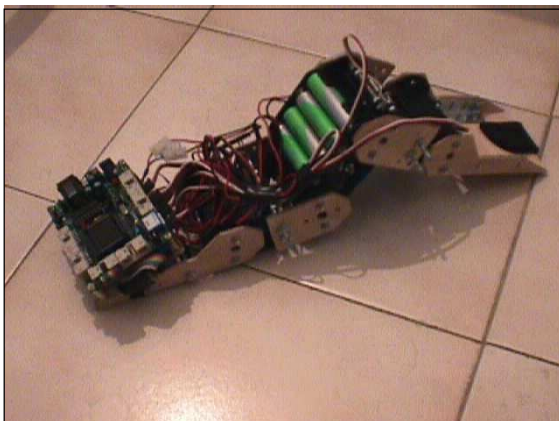


Foto 3: Cube 1.0 desplazándose por otra superficie

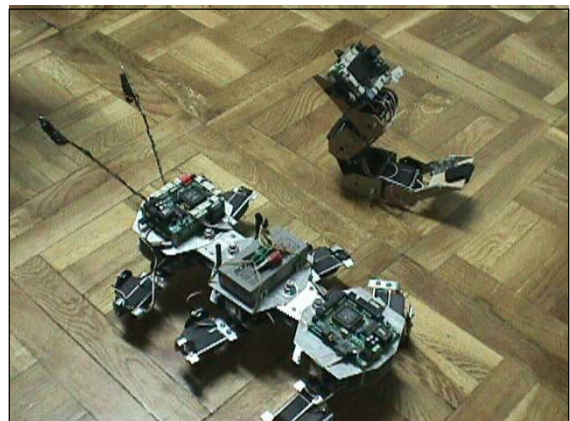


Foto 4: Cube 1.0 junto a la hormiga

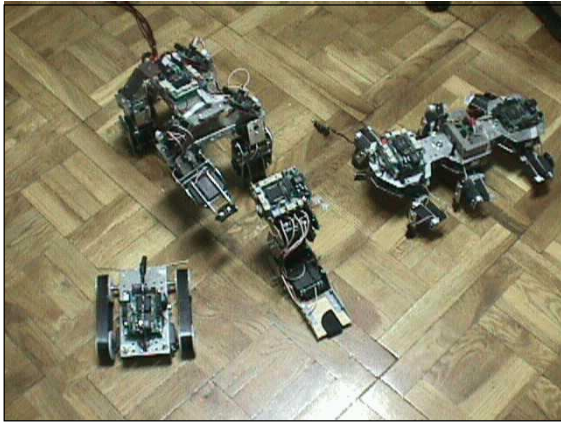


Foto 6: La granja de microbótica

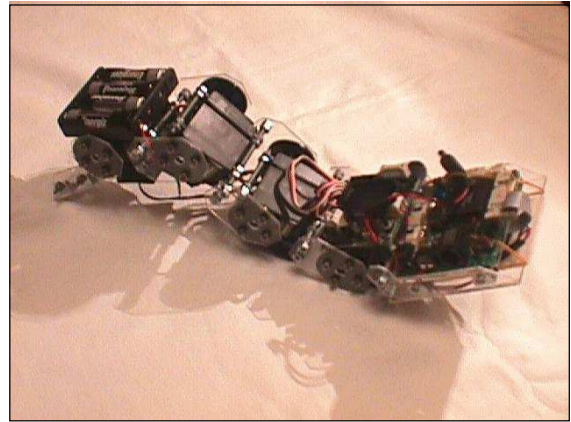


Foto 7: Cube 2.0 desplazándose sobre el suelo

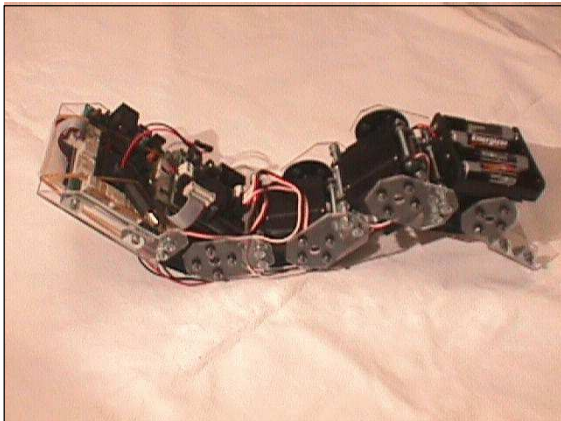


Foto 8: Cube 2.0 visto desde el otro lado

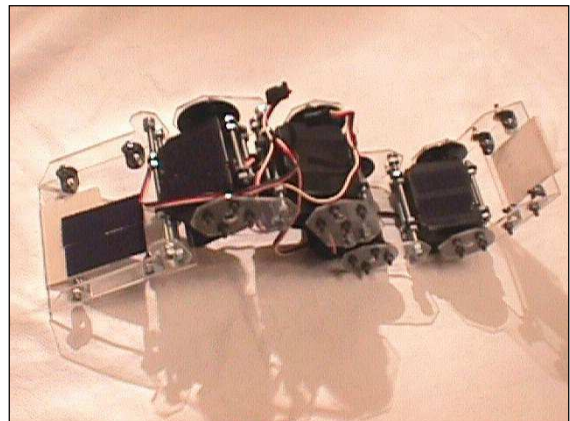


Foto 9: Estructura mecánica de Cube 2.0

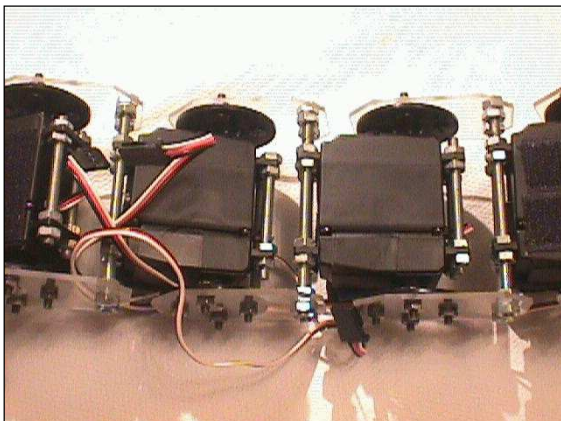


Foto 10: Detalle de la estructura mecánica

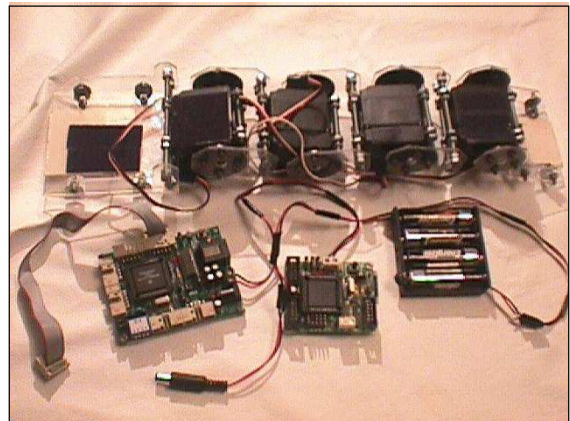


Foto 11: Estructura y electrónica de cube 2.0

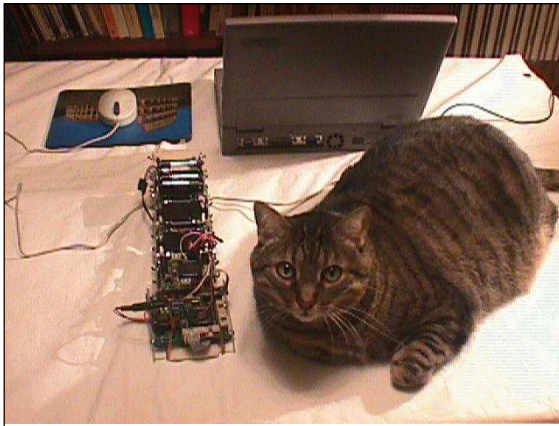


Foto 12: Entorno de trabajo: portátil, Cube y gato

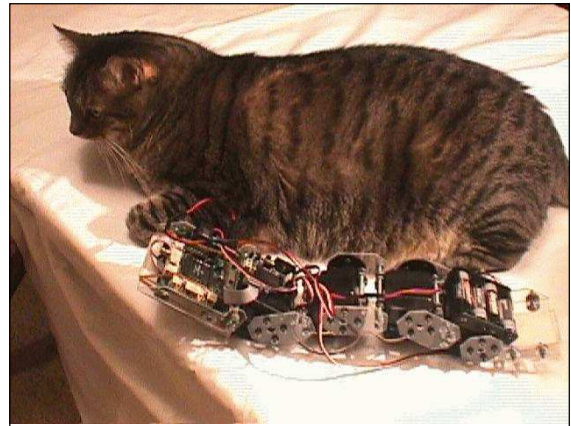


Foto 13: Cube al lado de Curro

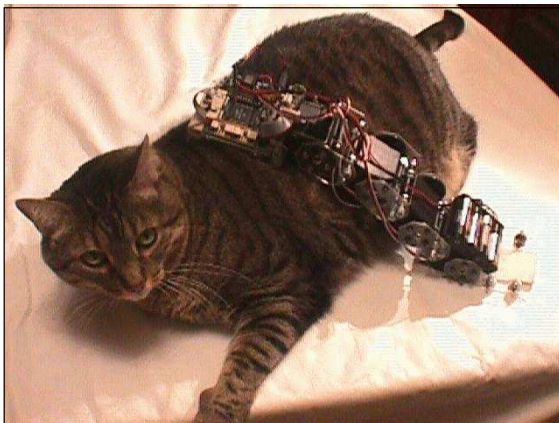


Foto 14: Cube quiere jugar con Curro



Foto 15: Cube sobrepasa a Curro, sin que se inmute

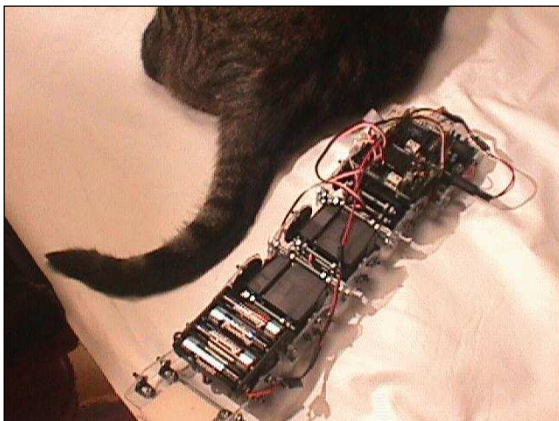


Foto 16: Comparación entre Cube y el rabo de Curro

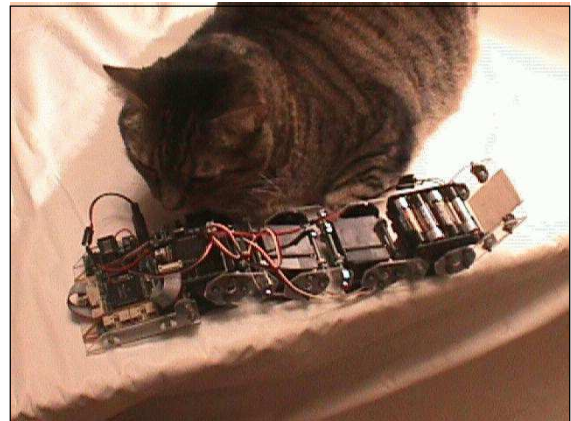


Foto 17: Curro olfateando a Cube. ¡Que bicho tan raro!

Bibliografía

- [1] S.J Industries inc [en línea]. <http://sjindustriesinc.com/ROBOTS/IRB90.htm>. [Consulta: 20 feb. 2001].
- [2] “An introduction to ROBOT technology”. Philippe Coiffet, Michael Chirouze. Ed. Kogan Page. 1983.
- [3] “Robótica industrial. Tecnología, programación y aplicaciones”. Mikell P. Groover, Mitchell Weiss. Ed. Mc Graw Hill. 1989.
- [4] “Robótica práctica”. Jose M^a Angulo. Ed. Paraninfo. 1986.
- [5] “R.U.R”, Son las siglas de “Rossum’s Universal Robots”.Karel Capek. 1917
- [6] “Sueños de Robot”. Isaac Asimov. Ed. P&J. 1992.
- [7] Robot humaniforme desarrollado por Honda [en línea]. <http://world.honda.com/robot>. [Consulta: 25 feb. 2001].
- [8] COG [en línea]. <http://caes.mit.edu/mup/html/cog.html>. [Consulta: 28 feb. 2001].
- [9] Robocup [en línea]. <http://www.robocup.org>. [Consulta: 2 mar. 2001].
- [10] Concurso de luchadores de sumo organizado por la AESS [en línea]. <http://www.aess.upc.es>. [Consulta: 2 mar. 2001].
- [11] Concurso luchadores de sumo organizado por el grupo de robótica de la EUPMT [en línea]. <http://www.eupmt.es/cra>. [consulta: 4 mar. 2001].
- [12] Alcabot 2000 [en línea]. <http://www.depeca.alcala.es/alcabot2000>. [Consulta: 6 mar. 2001].
- [13] Alcabot 2001 [en línea]. <http://www.alcabot.uah.es>. [Consulta: 6 mar. 2001].
- [14] Vehículo espacial enviado a marte [en línea]. <http://mpfwww.jpl.nasa.gov/MPF/rovercom/rovcom.html>. [Consulta: 8 mar. 2001].
- [15] AIBO [en línea]. http://www.aibo.com/ers_210. [Consulta: 9 mar. 2001].
- [16] “Diseño, construcción y control de un robot articulado mediante una red de microcontroladores”. Proyecto Fin de Carrera de D. Andrés Prieto-Moreno Torres. ETSIT-UPM. 2001.

- [17] UIUC Hexapod Project. Beckman institute for Advanced Science and Technology. University of Illinois at Urbana-Champaign [en línea]. <http://soma.npa.uiuc.edu/labs/nelson/hexapod.html>. [Consulta: 9 mar. 2001].
- [18] Robot Snake, en la Nasa [en línea]. <http://ic-www.arc.nasa.gov/ic/snakebot/hello.html>. [Consulta: 9 ene. 2001].
- [19] Empresa Microbótica, S.L [en línea]. <http://www.microbotica.es>. [Consulta: 5 sep. 2000].
- [20] “Microbótica”. Jose M^aAngulo Usategui. Ed. Paraninfo. 2000.
- [21] “Microcontrolador 68HC11: Fundamentos, recursos y programación”. C. Doblado, A. Prieto-Moreno, J. San Martín, J. González. Servicio de publicaciones de la ETSIT. 1998.
- [22] GMD [en línea]. <http://ais.gmd.de/BAR/snake.html>. [Consulta: 10 mar. 2001].
- [23] Dr. Gavin Miller [en línea]. <http://snakerobots.com>. [Consulta: 10 mar. 2001].
- [24] Mita Laboratory. <Http://www.ctrl.titech.ac.jp/ctrl-labs/mita-lab/snake/home.html>
- [25] Página índice con muchos enlaces a proyectos de serpientes robóticas [en línea]. <http://ais.gmd.de/~worst/snake-collection.html>. [Consulta: 8 ene. 2001].
- [26] Talleres y concursos impartidos por Microbótica, S.L [en línea]. <http://www.microbotica.es/taller.htm>. [Consulta: 5 sep. 2000].
- [27] “Física general 1”. José M. de Juana. Ed. Alhambra Universidad. 1990.
- [28] “Mobile Robots. Inspiration to implementation”. Joseph L. Jones. Bruce A. Seiger. Anita M.Flynn. Ed. A.K Peters. 1999.
- [29] “Desarrollo de aplicaciones Linux con GTK+ y GDK”. Eric Harlow. Prentice-Hall. 1999.
- [30] “Manual de referencia de C”. Herbert Schildt. Mc Graw Hill. 1989.
- [31] “Unix, programación avanzada”. Fco. Manuel Márquez. Ed. Ra-ma. 1996.
- [32] “Microlinux empotrado en arquitecturas digitales basadas en microcontroladores y lógica programable”. Borrador de PFC de D. Juan José San Martín Mazzucconi. ETSIT. UPM. 2001.
- [33] “Control de sistema empotrado basado en uClinux a través de http: desarrollo de servidor web e interfaces gráficas”. Borrador de PFC de D^a. Cristina Doblado Alcázar. ETSIT. UPM. 2001.