

Laboratorio de Robótica
Curso 2007-2008
(2º semestre)

Universidad Autónoma de Madrid
Escuela Politécnica Superior

1. Presentación
2. Normas del laboratorio
3. Prácticas
4. GPBOT: Kit Básico de Robótica
5. Programación básica (puertos)
6. Programación avanzada (serie y timer)

Andrés Prieto-Moreno Torres

andres@iearobotics.com

andres.prietomoreno@uam.es

Tutorías:

Por mail en cualquier momento

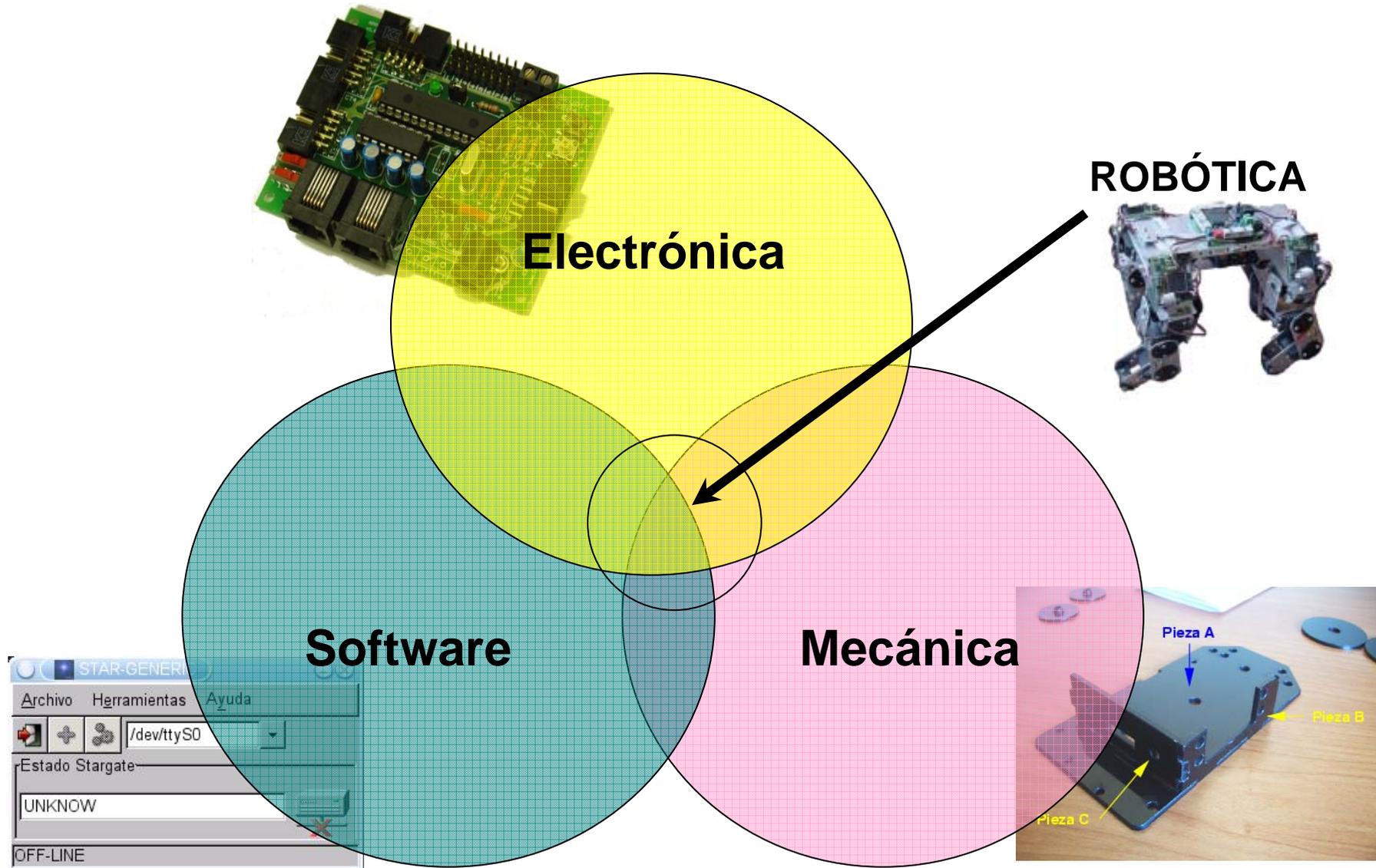
Presenciales:

Miércoles (grupo A): 18:00 – 20:00

Viernes (grupo XX): 16:00 – 18:00

Web asignatura: Guillermo Glez. de Rivera (coordinador Laboratorio)

<http://www.ii.uam.es/~gdrivera/robotica/robotica.htm>



- Se aprueba asistiendo a la P0 y aprobando la práctica final con nota >5
- Las prácticas se entregan según el calendario de la web
- Las prácticas se hacen en grupos de hasta 3 personas:
 - Se evalúa independientemente a cada alumno
 - Pueden tener distinta nota.
- Habrá material de la UAM que se dará en préstamo

- La asistencia no es obligatoria pero si recomendable
- Cada grupo hace las prácticas en el horario asignado
- No hay cambios de grupo -> Guillermo G. de Rivera
- **Las prácticas se evalúan en el laboratorio**
- Copia:
 - Copiado y copiador suspensos
 - Cuidado con dejar la práctica en el ordenador
 - Cuidado con el código de internet

- P0: Explicación y puesta en marcha del entorno
- P1: Diseño y construcción de un robot móvil capaz de:
 - Realizar un recorrido previamente marcado en un PC (PC-Robot)
 - Realizar un recorrido marcado en el suelo e informar al PC (Robot-PC)

Material:

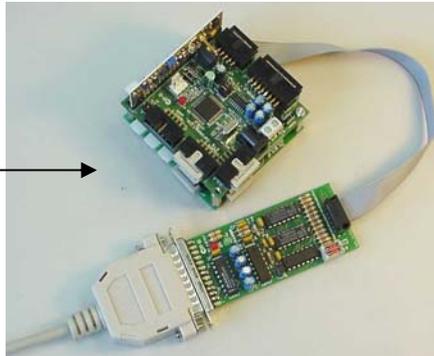
4 Sensor Infrarrojos: CNY70
Portapilas plano 4 unidades
Jack de alimentación macho 2.5
Cables de sección 0.25mm
2 motores Futaba S3003 o equivalente
Piezas para hacer la estructura

Se valorará la iniciativa de cada grupo.
La UAM posee cierto material para préstamo.

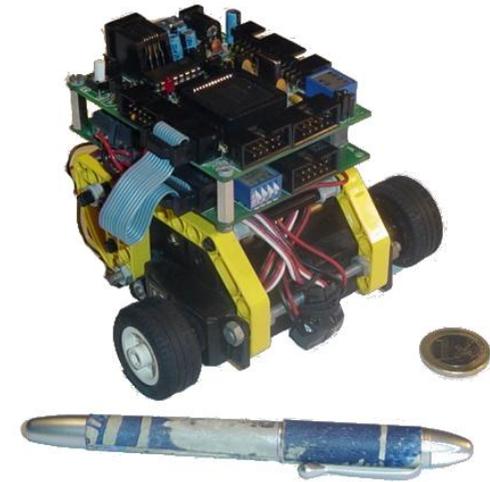


Herramientas Software

RS232



Entrenadora



Estructura Robot

Herramientas Software

Windows

Editor Programmer's Notepad

Compilador SDCC (compilador de C)

Descarga con Prog08sz (Pemicro)



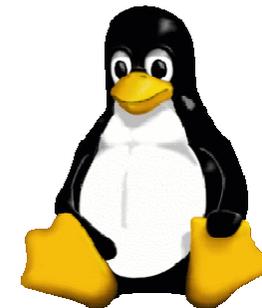
Linux

Editor Anjuta, Vim, emacs

Compilador SDCC (make)

Descarga con GPDOWN (learobotics)

<http://arantxa.ii.uam.es/~gdrivera/robotica/cuadernos/c10/c10.html>



The screenshot displays a Windows desktop environment with three main windows:

- Programmers Notepad 2 - [ledp.c]:** Shows the source code for a C program named `ledp.c`. The code includes comments in Spanish and a function `delay_init` for configuring a timer. The code is as follows:


```

/*-----*/
/* ledp.c (c) Juan Gonzalez, Marzo 2004 */
/*-----*/
/* Ejemplo que hace parpadear un led conectado */
/* al bit 0 del puerto B */
/* Se cambia el estado del led cada 0.5 segs */
/*-----*/
/* LICENCIA GPL */
/*-----*/
#include "mc68hc908gp32.h"

/*-----*/
/* Inicializacion del temporizador. se debe */
/* llamar esta funcion antes de hacer cualquier */
/* pausa. */
/* Se configura el TIM1 para active el flag de */
/* overflow cada 100ms */
/*-----*/
void delay_init(void)
{
    /*-----*/
    /*- Configurar el temporizador */
    /*-----*/
    T1SC = 0x06; // Prescaler: Div entre 64
    T1MODH = 0x0E; // Establecer modulo 0x0EF7
    T1MODL = 0xF7;
}

/*-----*/
/* Rutina de pausa, en unidades de 100ms */
/*-----*/

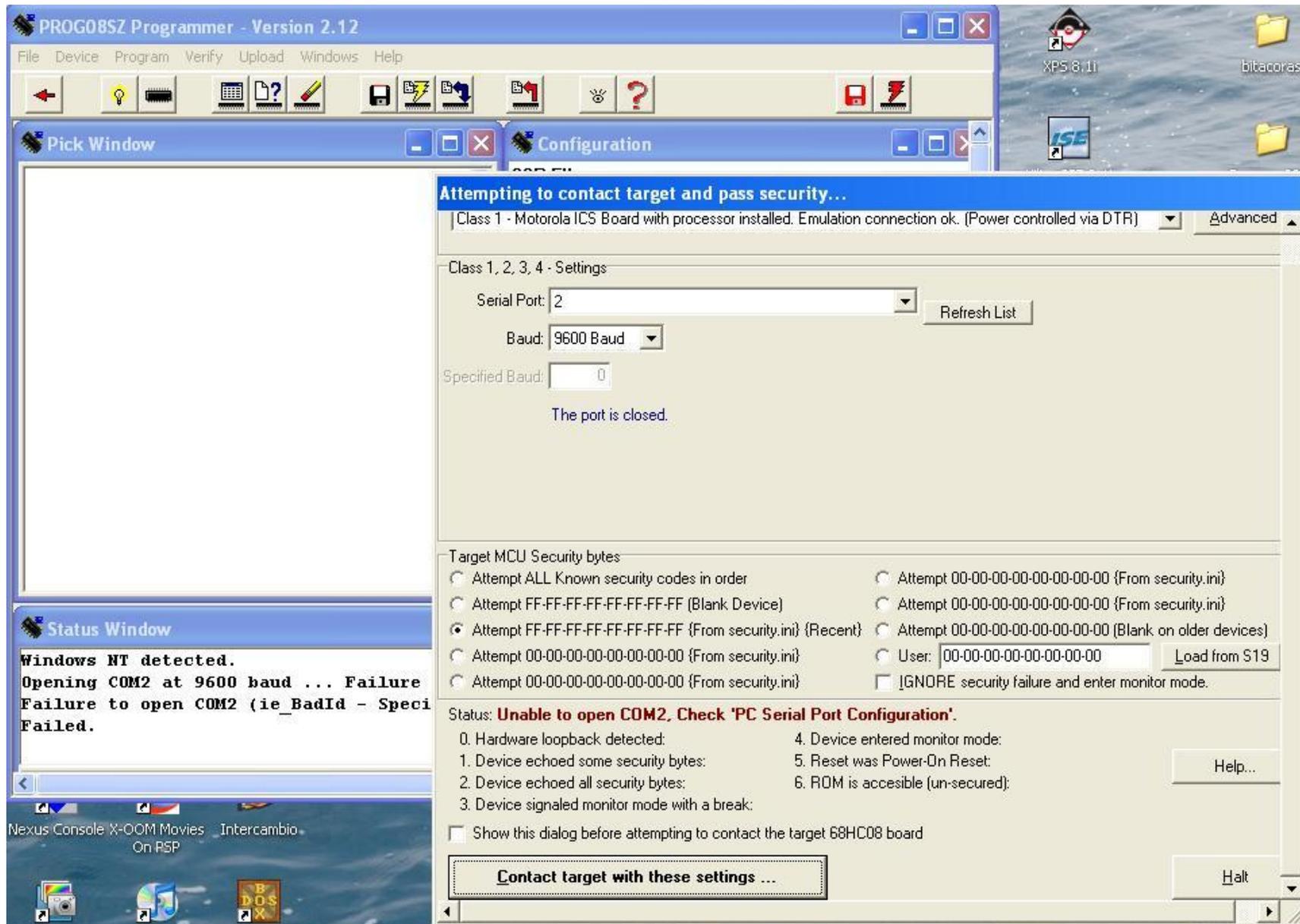
```
- Output:** Shows the command executed in the terminal:


```

> "C:\Archivos de Programa\SDCC\bin\sdcc.exe" -mhc08 --stack-loc 0x023f ledp.c
> Process Exit Code: 0

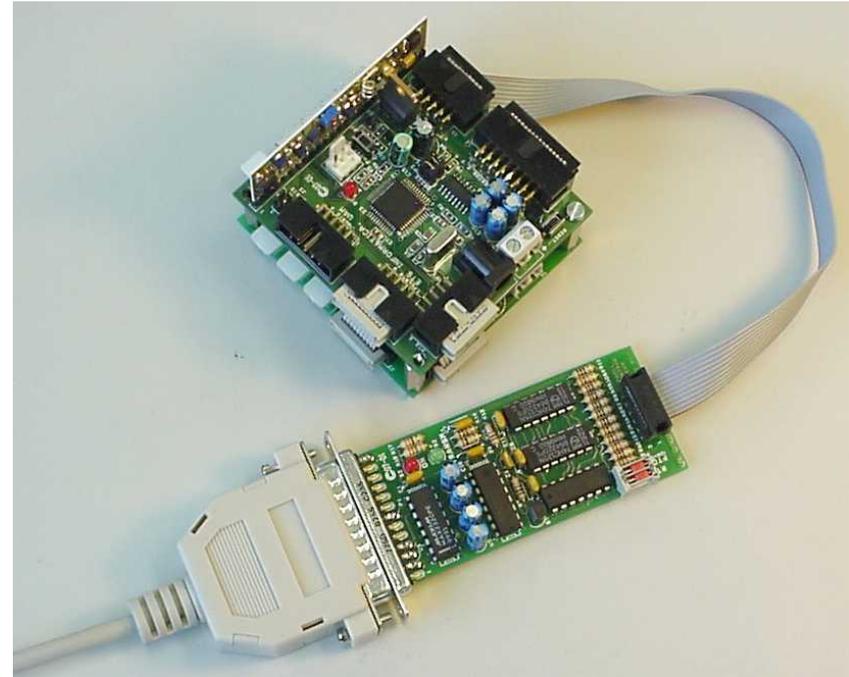
```
- Options:** A dialog box for configuring the compiler tool. It shows the 'Scheme' set to 'C / C++' and a list of tools:

Name	Command	Params
sdcc	C:\Archivos de Pro...	-mpic14 -p16f8...
sdcc con libreria auxilia...	%d\crear_ejecuta...	%n "%d"
sdcc GPBOT	C:\Archivos de Pro...	-mhc08 --stack-...
- Propiedades de Edit Tool:** A dialog box for configuring the 'sdcc GPBOT' tool. It shows the following settings:
 - Name: `sdcc GPBOT`
 - Command: `C:\Archivos de Programa\SDCC\bin\sdcc`
 - Folder: `%d`
 - Parameters: `-mhc08 --stack-loc 0x023f %f`
 - Shortcut: `Ninguno`
 - Save: `None`
 - This tool will modify the current file.

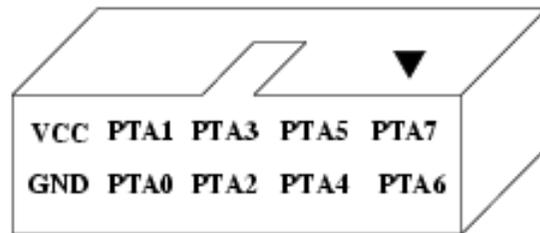


Micro Motorola MC68HC908GP32

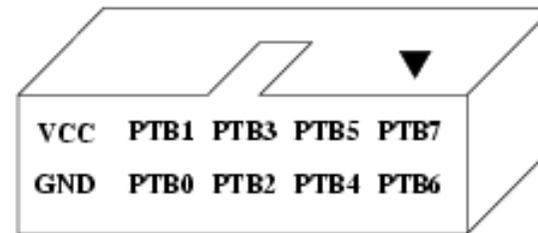
- Reloj a 9.8 Mhz
- 32Kb de memoria FLASH
- 512 bytes memoria RAM
- Comunicaciones SPI, SERIE
- 2 temporizadores de 16 bits
- 8 conversores AD
- Pines IO
- Adaptado para programar en C



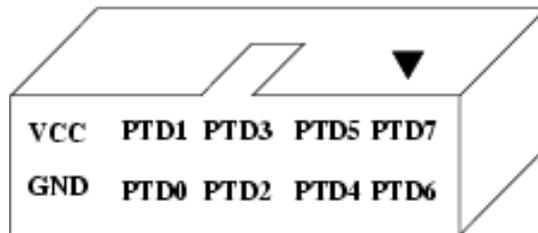
PUERTO A



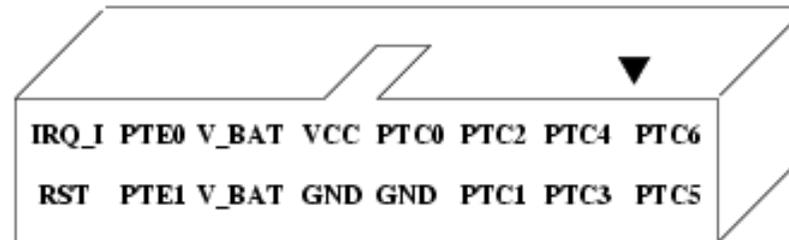
PUERTO B



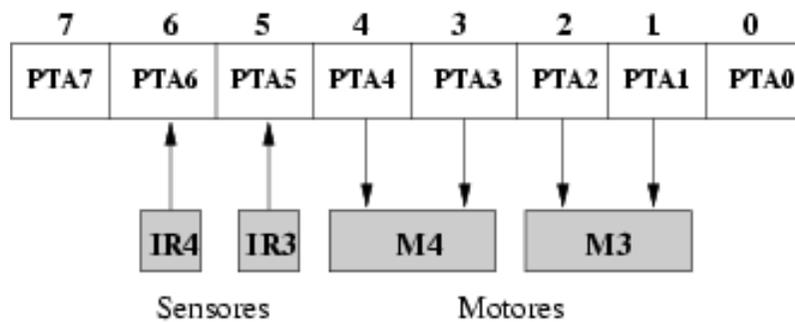
PUERTO D



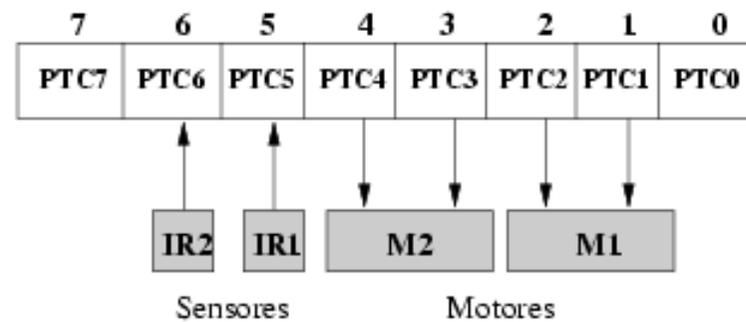
PUERTO C y E



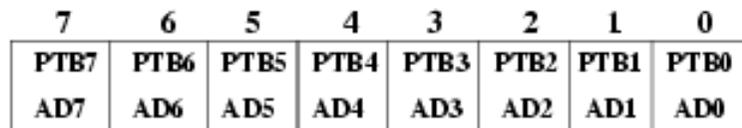
PUERTO A



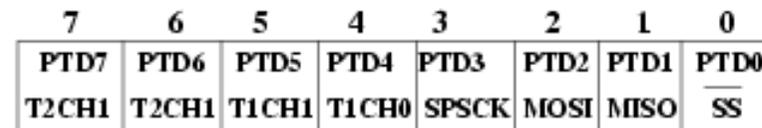
PUERTO C



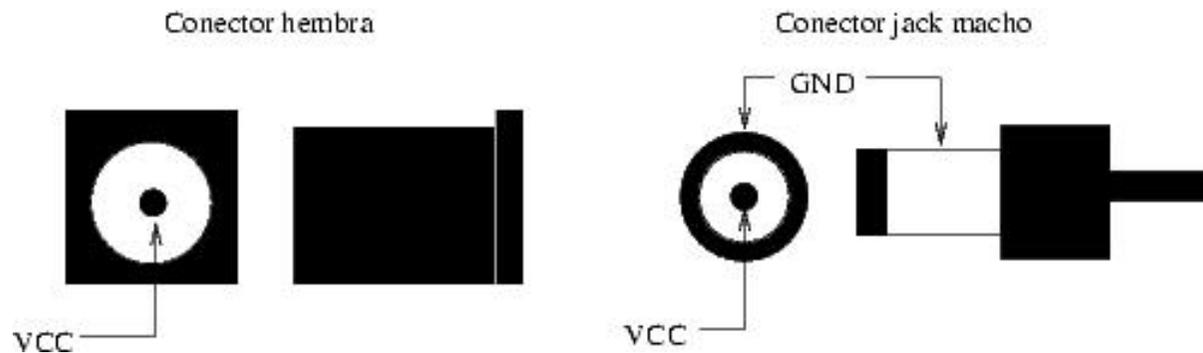
PUERTO B



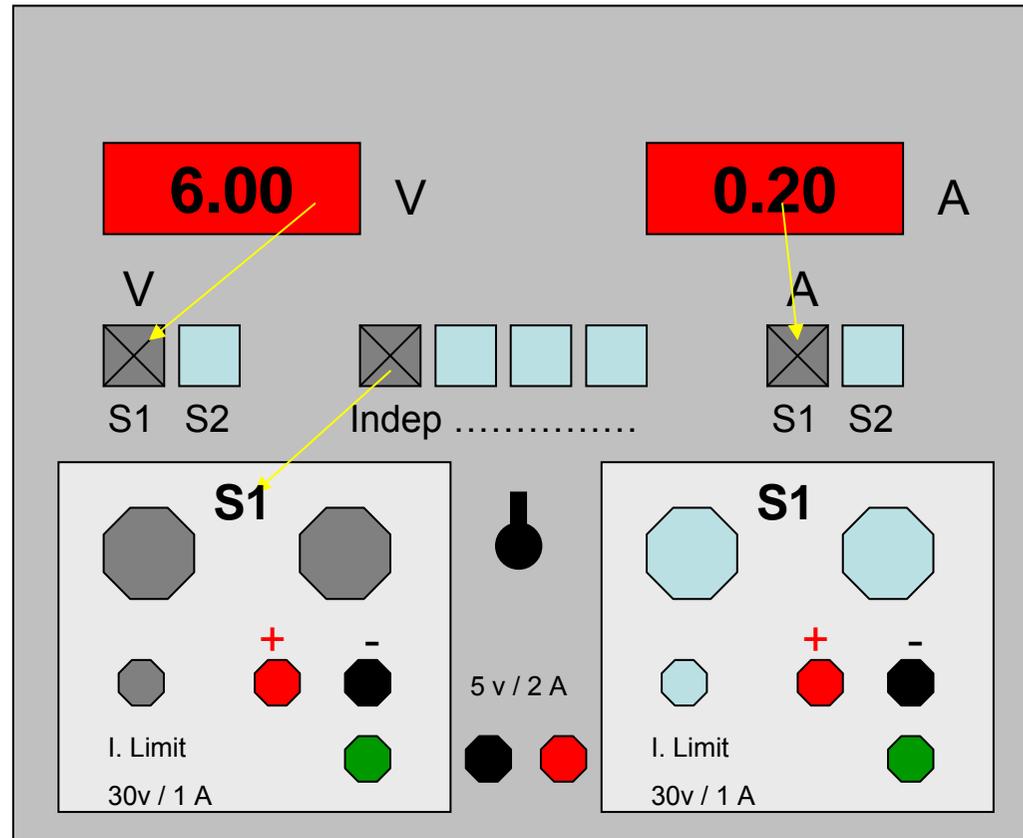
PUERTO D

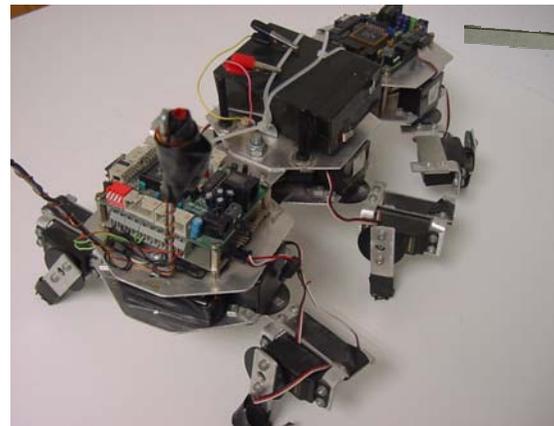
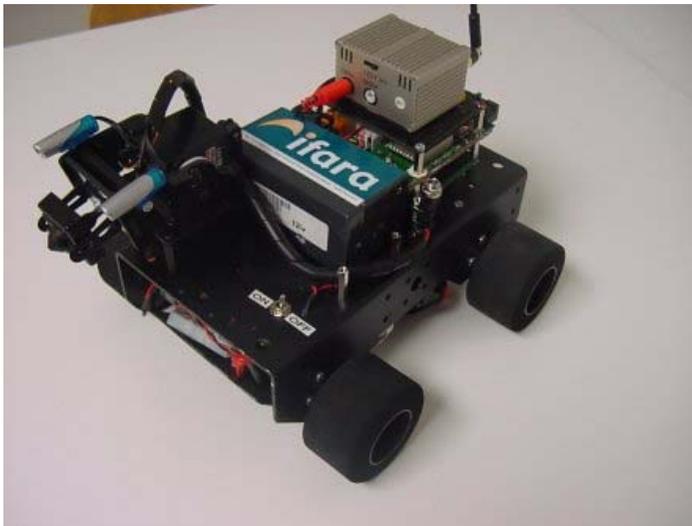
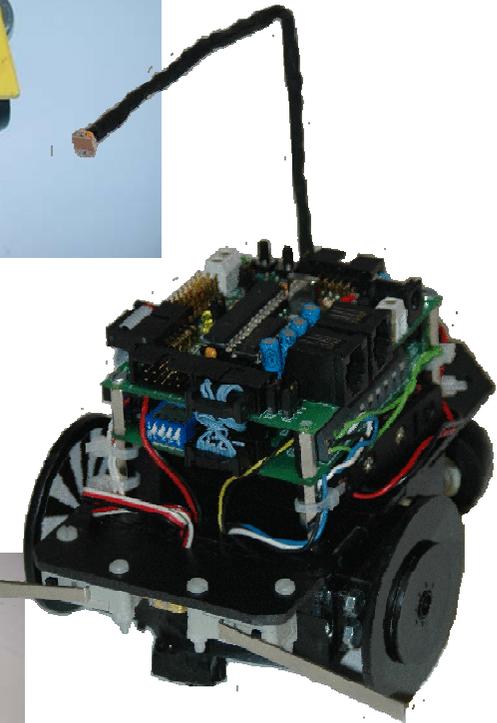
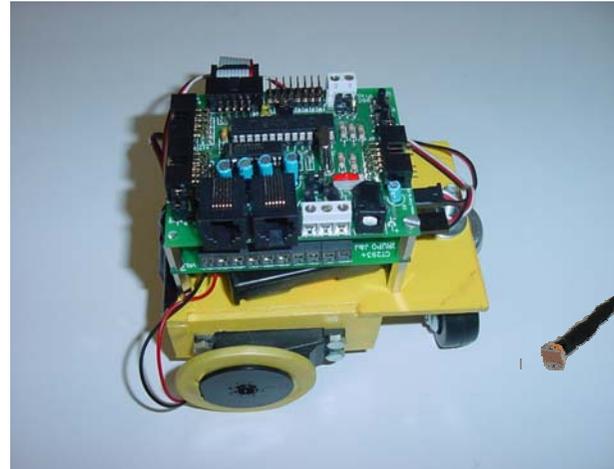


Antes de enchufar verificar la tensión de la fuente



Alimentación entre 5.5 y 9 voltios. Recomendable 6v





1. Puesta en marcha del entorno (1 semana)

- Conseguir la placa entrenadora GPBOT
- Fabricar el cable de conexión a la Fuente de alimentación.
- Instalar el software y probar con algún ejemplo básico. (WEB)

2. Construir el robot (2 semanas)

- Adaptar los servos. Se van a necesitar dos motores.
- Montar los sensores CNY70. Se van a necesitar 4 sensores.
- Adaptar la electrónica a la estructura.
- Para aprender hacer el programa de la línea negra.

3. Comunicar el robot con el PC (2 semanas)

- Construir el cable serie
- Usar programa de ejemplo para probar el cable y el entorno. (WEB)
- Programar en C un pequeño programa cliente-servidor. Por ejemplo mover el robot con teclas desde un terminal serie.

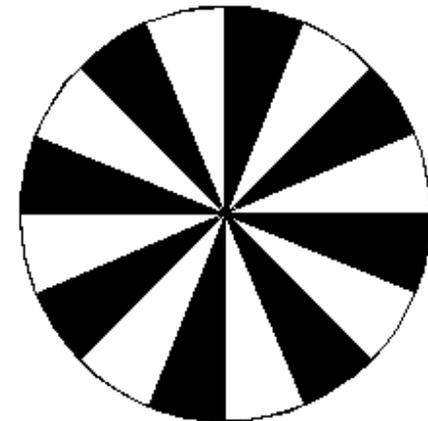
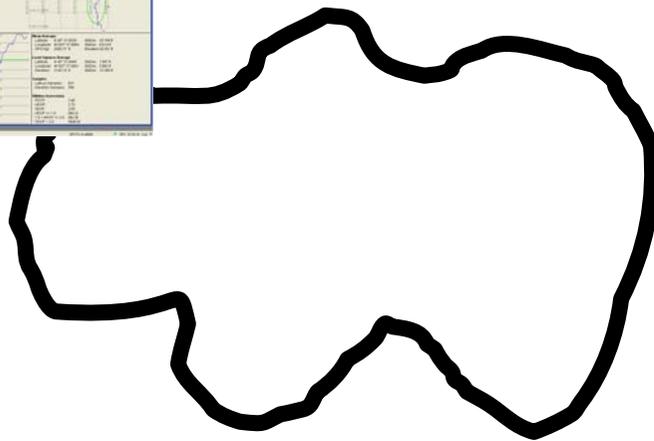
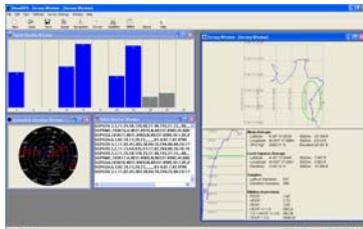
Paso 4: Hacer el programa del micro que lee los encoders. (3 semanas)

Dos funciones principales:

- Llevar la cuenta de pasos con el encoder e informar al PC
- Mover rueda un determinado número de pasos que se le indica desde el PC

Paso 5: Con esas dos funciones podrán hacer el resto de la práctica. (4 semanas)

- Realizar un recorrido previamente marcado en un PC (PC-Robot)
- Realizar un recorrido marcado en el suelo e informar al PC (Robot-PC)



Trucaje Futaba:

<http://www.iearobotics.com/proyectos/cuadernos/ct2/ct2.html>

Conexión CNY70 y Motor a la GPFAZ

<http://arantxa.ii.uam.es/~gdrivera/robotica/curso0607/p1/p1-doc.html>

Construcción cable serie

<http://arantxa.ii.uam.es/~gdrivera/robotica/curso0506/p2/p2-doc.html>

Ejemplos de Programación

<http://www.iearobotics.com/personal/juan/proyectos/gpbot/gpbot.html>

1. Presentación
2. Normas del laboratorio
3. Prácticas
4. GPBOT: Kit Básico de Robótica

5. Programación básica (puertos)
6. Programación avanzada (serie y timer)

1. Puesta en marcha del entorno (1 semana)

- Conseguir la placa entrenadora GPBOT
- Fabricar el cable de conexión a la Fuente de alimentación.
- Instalar el software y probar con algún ejemplo básico. (WEB)

2. Construir el robot (2 semanas)

- Adaptar los servos. Se van a necesitar dos motores.
- Montar los sensores CNY70. Se van a necesitar 4 sensores.
- Adaptar la electrónica a la estructura.
- Para aprender hacer el programa de la línea negra.

3. Comunicar el robot con el PC (2 semanas)

- Construir el cable serie
- Usar programa de ejemplo para probar el cable y el entorno. (WEB)
- Programar en C un pequeño programa cliente-servidor. Por ejemplo mover el robot con teclas desde un terminal serie.

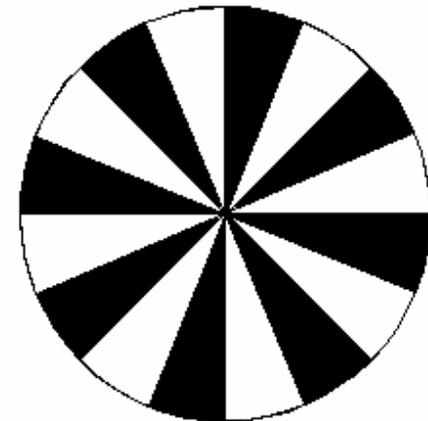
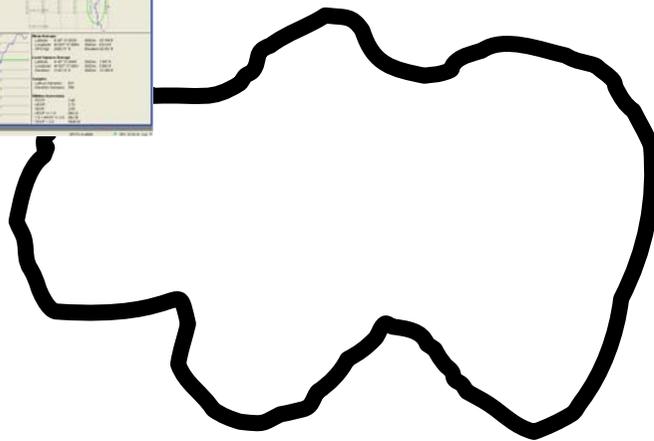
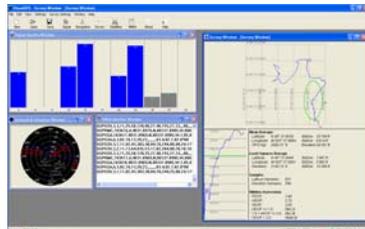
Paso 4: Hacer el programa del micro que lee los encoders. (3 semanas)

Dos funciones principales:

- Llevar la cuenta de pasos con el encoder e informar al PC
- Mover rueda un determinado número de pasos que se le indica desde el PC

Paso 5: Con esas dos funciones podrán hacer el resto de la práctica. (4 semanas)

- Realizar un recorrido previamente marcado en un PC (PC-Robot)
- Realizar un recorrido marcado en el suelo e informar al PC (Robot-PC)



Trucaje Futaba:

<http://www.iearobotics.com/proyectos/cuadernos/ct2/ct2.html>

Conexión CNY70 y Motor a la GPFAZ

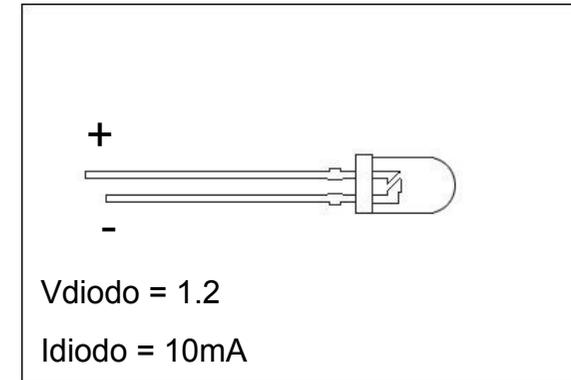
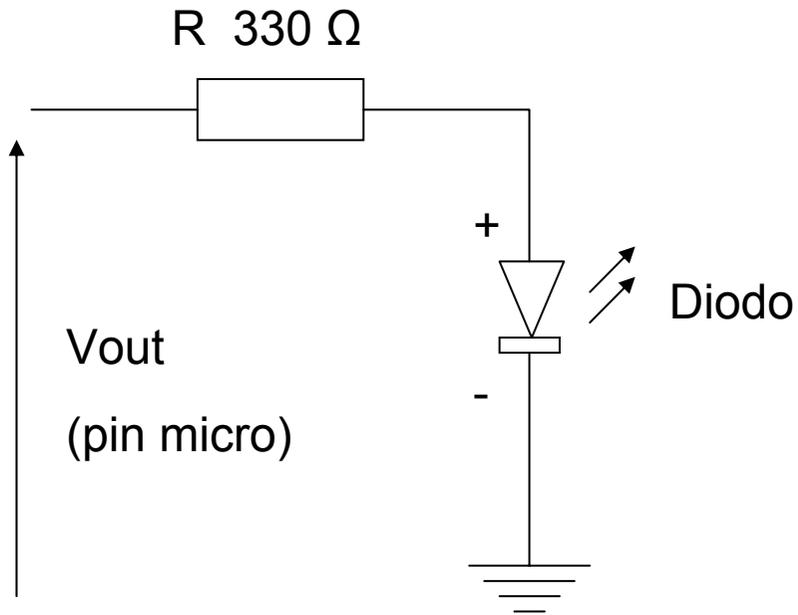
<http://arantxa.ii.uam.es/~gdrivera/robotica/curso0607/p1/p1-doc.html>

Construcción cable serie

<http://arantxa.ii.uam.es/~gdrivera/robotica/curso0506/p2/p2-doc.html>

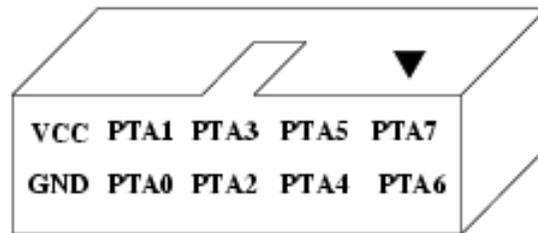
Ejemplos de Programación

<http://www.iearobotics.com/personal/juan/proyectos/gpbot/gpbot.html>

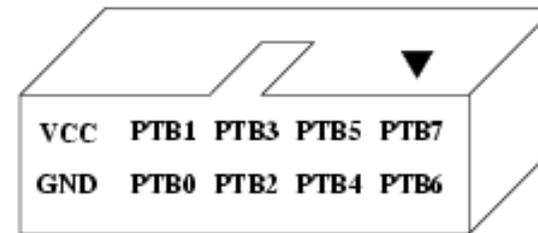


$$R = \frac{V_{out} - V_{diodo}}{I_{diodo}} = 330 \Omega$$

PUERTO A



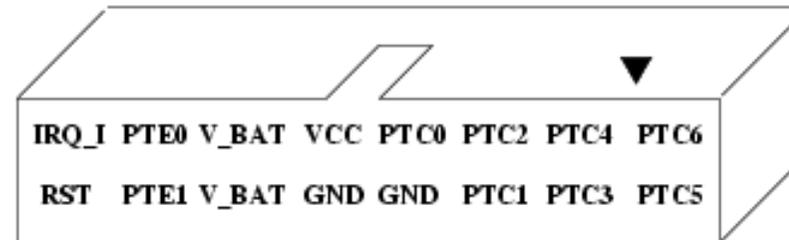
PUERTO B



PUERTO D

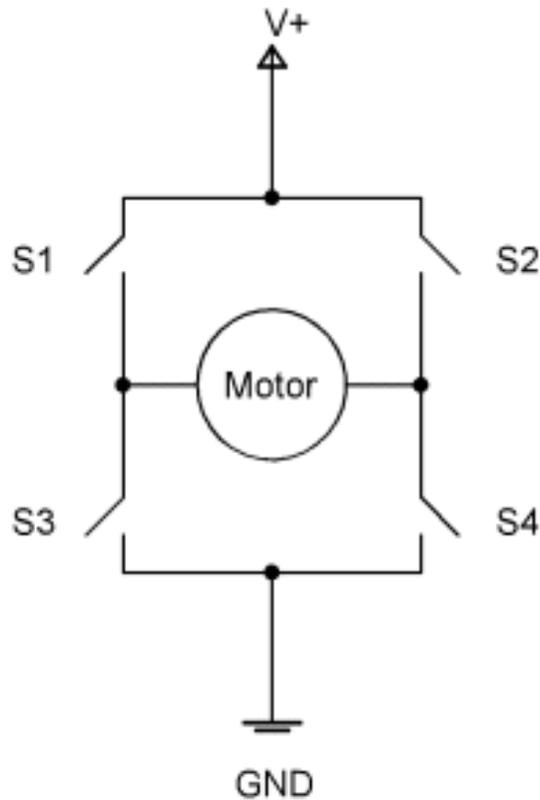


PUERTO C y E

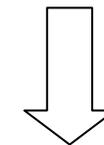


Software de prueba [portb_sal.c](#)

Puente en H: Controlador 1 motor

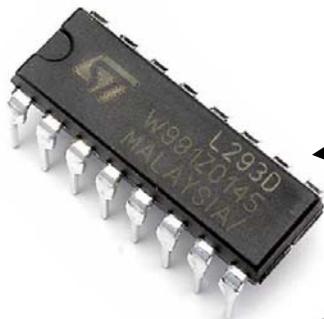


S1	S3	S2	S4	
On	Off	Off	On	Izq
Off	On	On	Off	Der
On	On	X	X	Error
X	X	On	Off	Error
On	Off	On	Off	Stop



Simplificando

A	B	
on	Off	Izq
off	on	Der
Off	Off	Stop

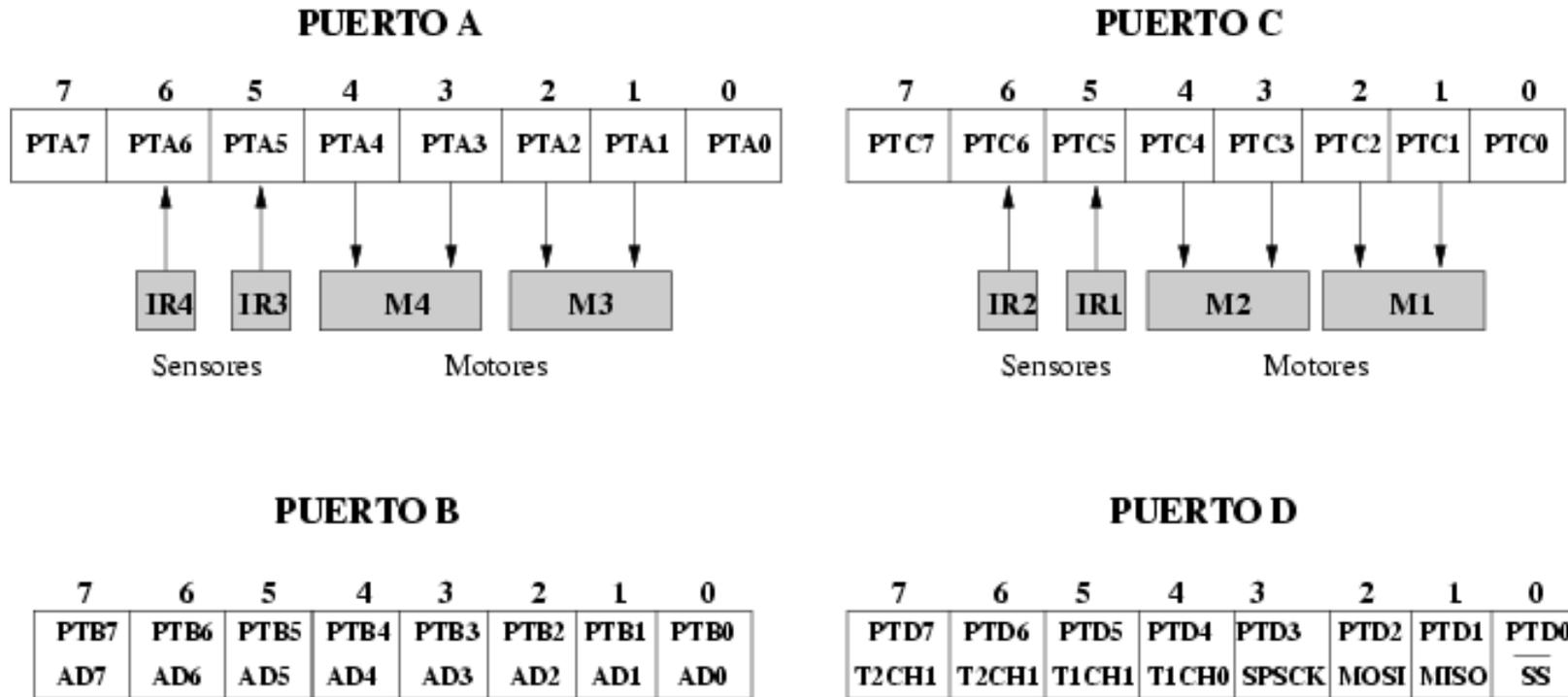


x2

L293D ofrece 2 P-H para dos motores

<http://www.learobotics.com/proyectos/cuadernos/ct2/ct2.html>

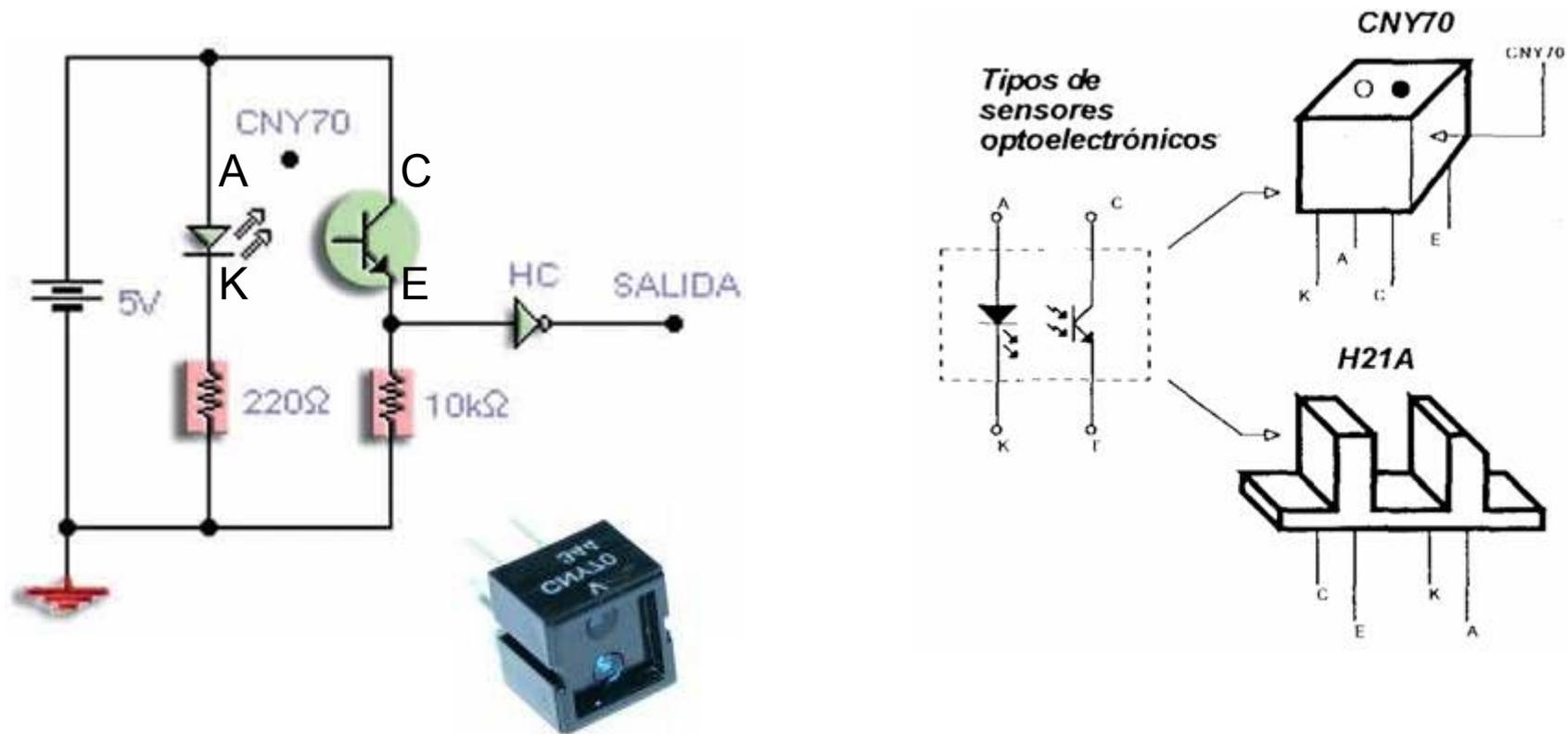




Software de prueba [motor_on.c](#)

Ejemplo de lectura de "Pin de entrada" usando un CNY70

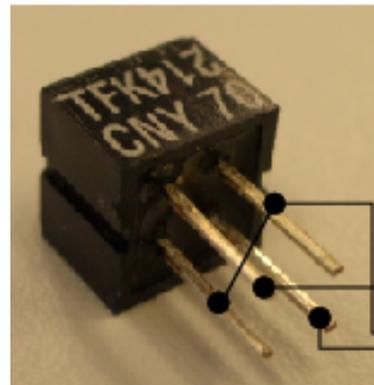
Ejemplo mezclando el futaba con el CNY70



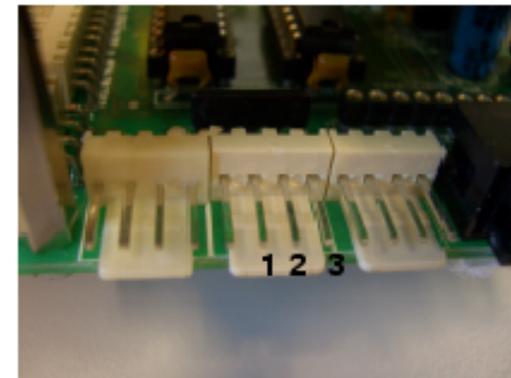
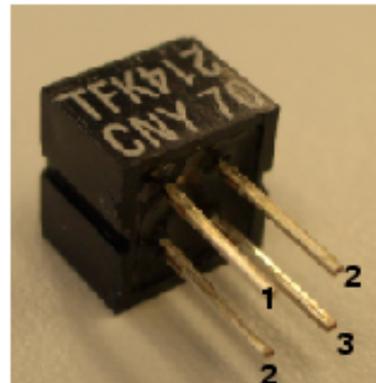
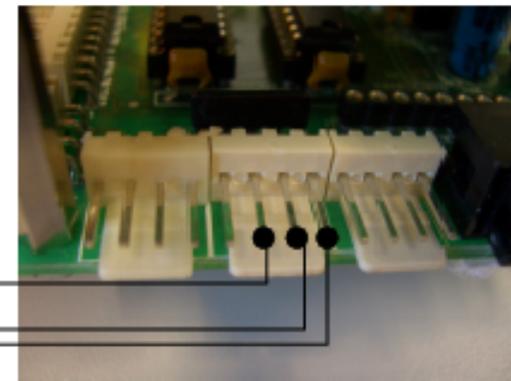
<http://arantxa.ii.uam.es/~gdrivera/robotica/curso0607/p1/p1-doc.html>



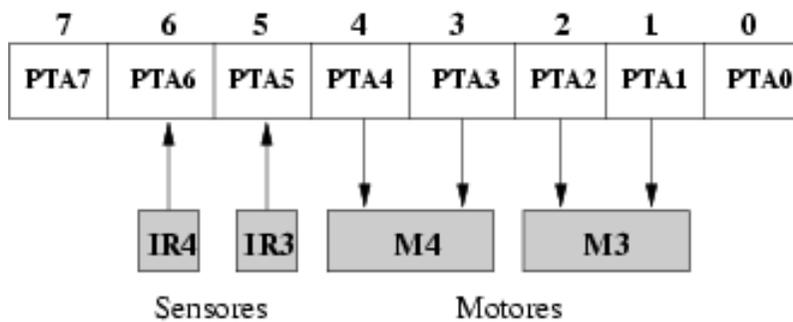
Sensor CNY70



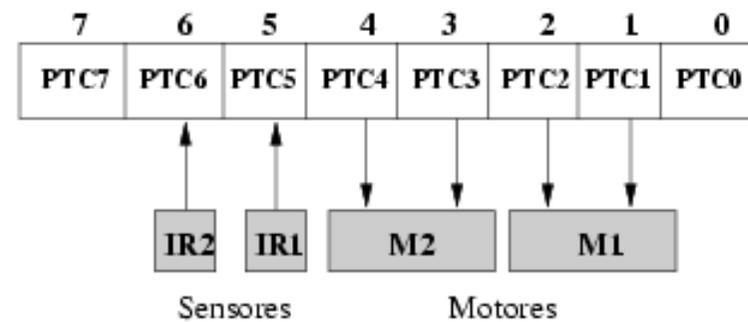
GP_IFAZ



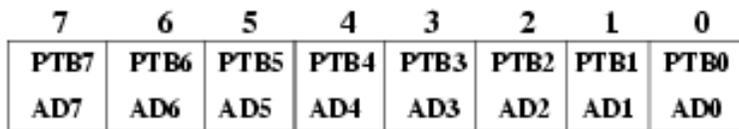
PUERTO A



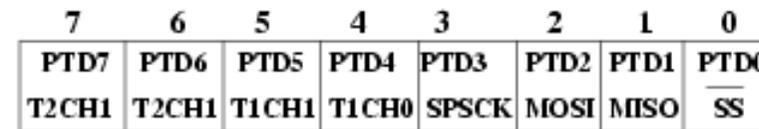
PUERTO C

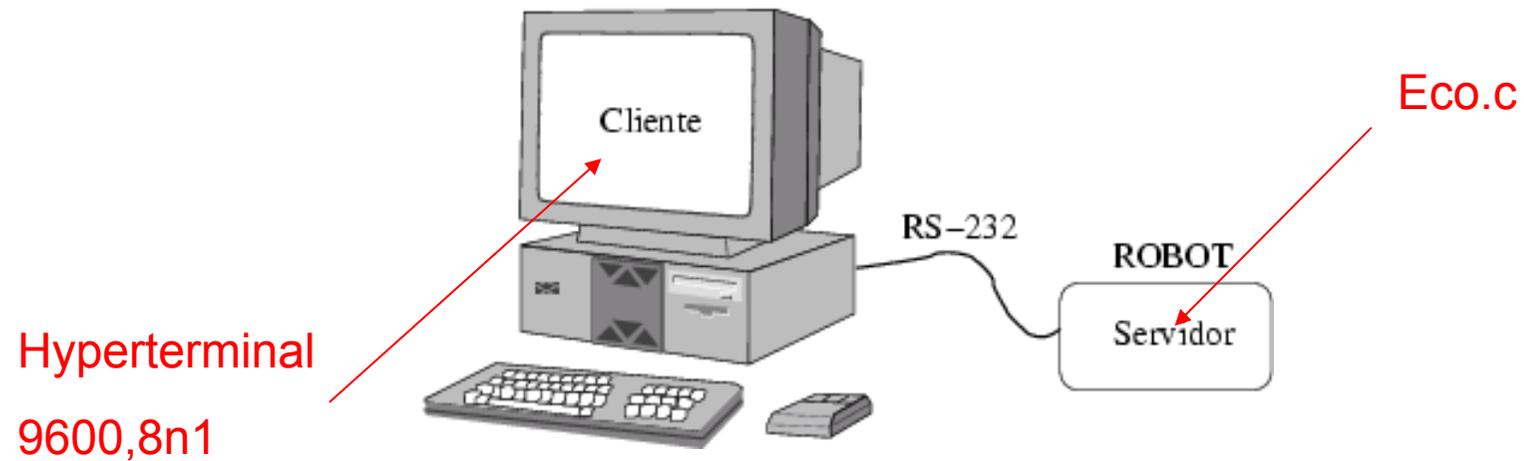


PUERTO B

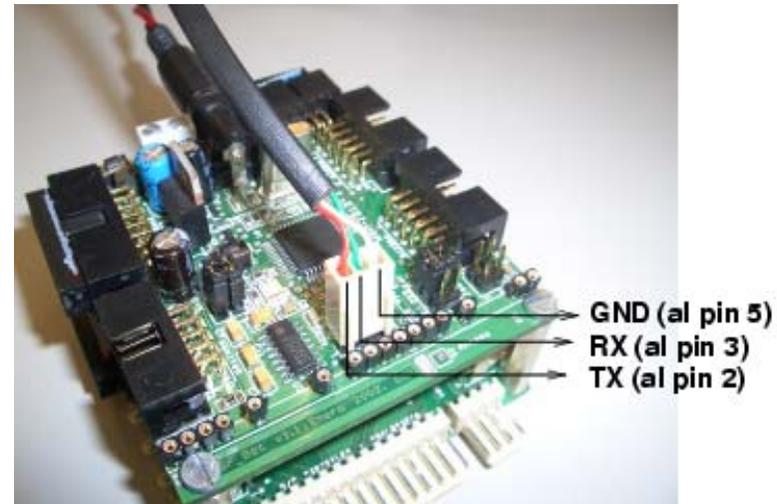
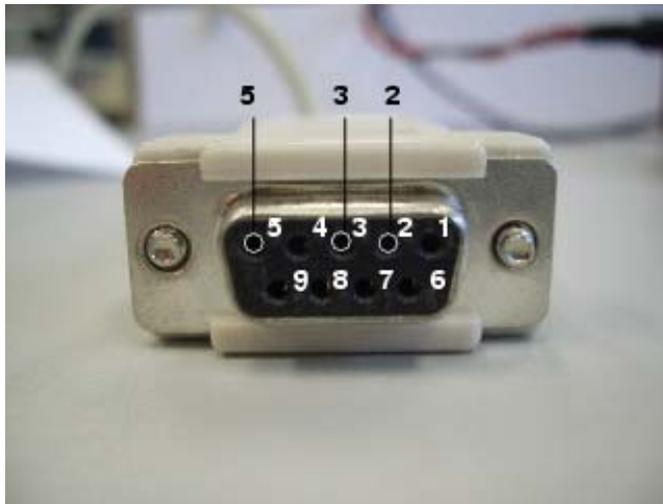


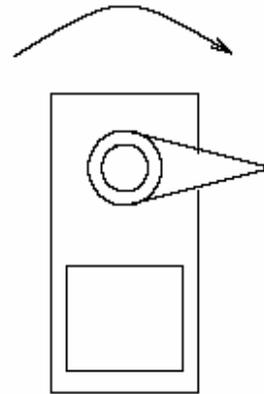
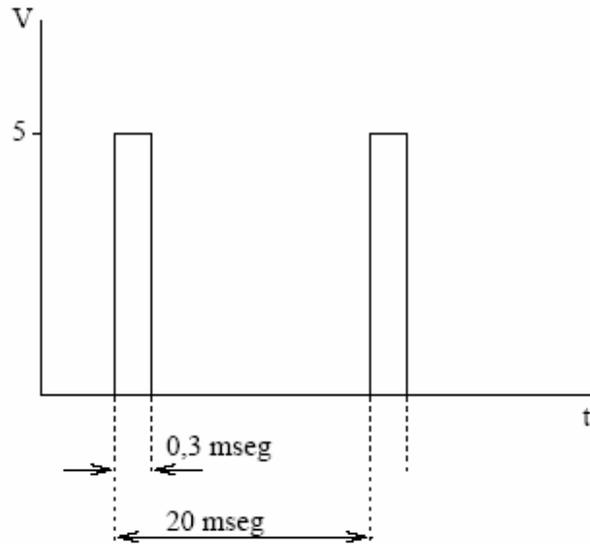
PUERTO D



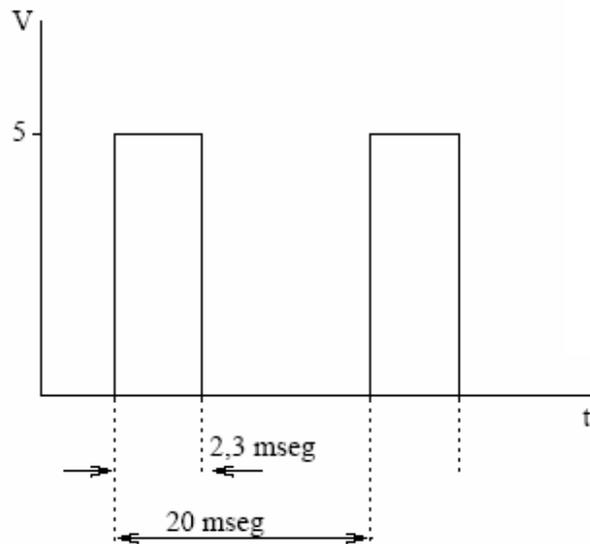


<http://arantxa.ii.uam.es/~gdrivera/robotica/curso0506/p2/p2-doc.html>





Futaba S3003 Analógico



	Futaba 3003
Velocidad	0.23 seg/60°
Par	3.2 Kg-cm
Tamaño	40 x 20 x 36 mm
Peso	37 g