

## Plataforma Robótica Controlada de Forma Remota Mediante una Red I2C de Microcontroladores

José Manuel Andújar Márquez

Dep. de Ingeniería Electrónica, Sistemas Informáticos y Automática, Univ. de Huelva, Palos de la Frontera (Huelva), andujar@diesia.uhu.es

Tomás de Jesús Mateo Sanguino

Dep. de Ingeniería Electrónica, Sistemas Informáticos y Automática, Univ. de Huelva, Palos de la Frontera (Huelva), tomas.mateo@diesia.uhu.es

Francisco José Aguilar Nieto

Dep. de Ingeniería Electrónica, Sistemas Informáticos y Automática, Univ. de Huelva, Palos de la Frontera (Huelva), franciscojose.aguilar@alu.uhu.es

Juan José Chica Barrera

Dep. de Ingeniería Electrónica, Sistemas Informáticos y Automática, Univ. de Huelva, Palos de la Frontera (Huelva), juanjose.chica@alu.uhu.es

### RESUMEN

Este artículo describe los resultados obtenidos en el desarrollo de una plataforma de pruebas, con el objetivo de ensayar y poner a punto diferentes estrategias de control remoto, visión artificial y adquisición y procesamiento de señales, las cuales están dedicadas al control de un vehículo de bajo peso equipado con un brazo robótico manipulador denominado VANTER (Vehículo Autónomo No Tripulado Especializado en Reconocimiento). Este vehículo está actualmente en desarrollo.

El objetivo fundamental de este trabajo es el diseño, implementación y pruebas de una red de microcontroladores que, mediante conexión RF, ha de permitir el control remoto del VANTER.

Para la comunicación punto a punto entre estos dispositivos y el intercambio de datos de telecomandos y telemetría, se ha implementado un protocolo de redirección de mensajes orientado a conexión, entre el interfaz virtual de usuario y la red de microcontroladores, usando como enlace el canal UHF y el bus I2C.

**Palabras Clave:** Plataforma robótica, interfaz virtual, control remoto, red de microcontroladores, bus I2C.

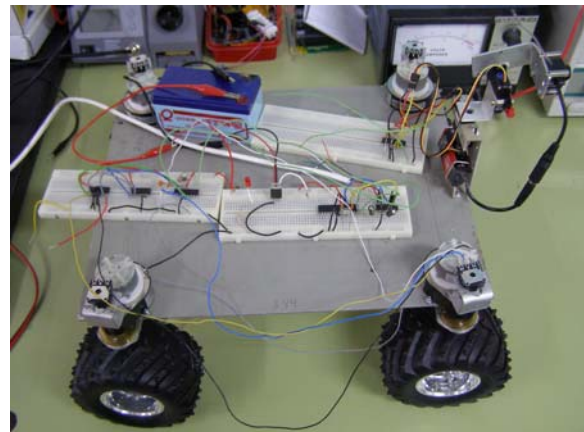


Figura 1: Plataforma de prueba dedicada a ensayar estrategias de control remoto y visión artificial del vehículo robótico VANTER.

### 1 INTRODUCCIÓN

La implementación de aplicaciones para la robótica mediante el uso de microcontroladores es algo largamente conocido en la ingeniería. La disponibilidad de dispositivos cada vez más completos, junto con el desarrollo de nuevas librerías de alto nivel, hacen posible la programación, fundamentalmente en lenguaje C de sus nuevas funcionalidades [3]. Todo ello permite diseñar una arquitectura hardware basada en una red de microcontroladores dedicados al control de tareas concretas. El reducido tamaño de estos dispositivos

con encapsulados SMD [9], hace posible además incluirlos abordo de los elementos a controlar o monitorizar, reduciendo el número de líneas de señal y alimentación hacia partes móviles como articulaciones o motores.

Este trabajo muestra los resultados obtenidos en el desarrollo de una plataforma de ensayo, dedicada a controlar un vehículo de bajo peso equipado con un brazo robótico manipulador llamado VANTER. El objetivo fundamental reside en aportar un sistema de control remoto basado en un módem UHF, directamente conectado a un microcontrolador maestro. Éste forma parte de una red de microcontroladores esclavos con tareas específicas sobre el hardware a controlar (motores y encoders). Estos dispositivos llevan los mensajes de información o control a través de la red I2C y son gobernados por un cliente remoto que monitoriza y ejecuta las tareas desde un interfaz virtual.

Existen pocas referencias dedicadas a la comunicación de microcontroladores en una red maestro-esclavo basada en el bus I2C [12]. De forma comercial, el bus I2C se utiliza típicamente para el control de pequeños dispositivos como pantallas LCD, teclados o sensores mediante un único microcontrolador [5]. Algunos trabajos de investigación describen el diseño de un protocolo de comunicación en una red I2C aislada, formada por microcontroladores que monitorizan datos de sensores mediante una configuración maestro-esclavo [11], o el uso de un PC empotrado que se comunica mediante RS-232 con hardware I2C y microcontroladores [8].

El artículo está dividido en varias secciones que desarrollan los puntos a tratar desde una perspectiva global del sistema hasta sus partes individuales, concretando en las implementaciones de hardware y software realizadas. Se describen sus características y

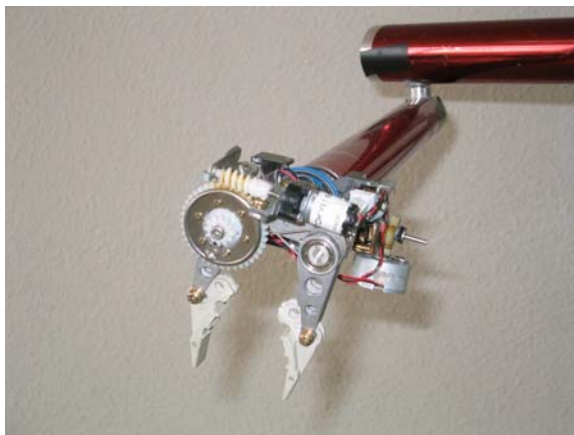


Figura 2: Muestra del brazo robótico montado en el vehículo VANTER.

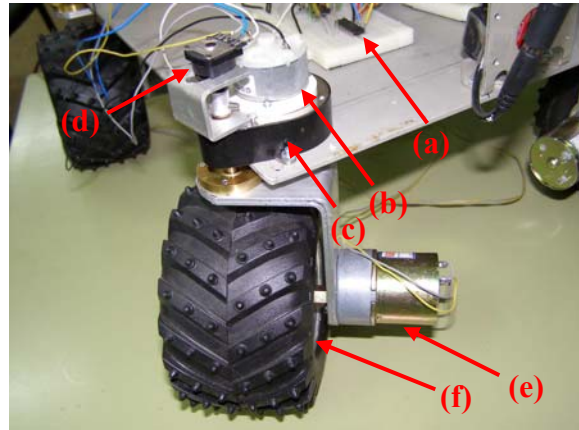


Figura 3: Detalle de un microcontrolador esclavo que controla un conjunto de motores y encoders.

funciones, incluyendo figuras y esquemas de los sistemas. Finalmente, se consideran algunas conclusiones y líneas actualmente en desarrollo.

## 2 VEHÍCULO AUTÓNOMO NO TRIPULADO ESPECIALIZADO EN RECONOCIMIENTO

El propósito final que ha sido marcado en el proyecto, es el desarrollo de un prototipo de robot móvil dotado de brazo manipulador denominado VANTER (Vehículo Autónomo No Tripulado Especializado en Reconocimiento) [2], con el objetivo de ensayar y poner a punto diferentes estrategias de control remoto, visión artificial y adquisición y procesamiento de señales. El desarrollo del prototipo ha sido dividido en varias fases de diseño, ensamblaje y programación del hardware.

Estas fases implican la implementación de una plataforma móvil con posicionamiento y tracción independiente (fig.1), un brazo robótico manipulador (fig.2), sistemas de control y adquisición de datos basados en motores y sensores (fig.3), una red de microcontroladores maestros-esclavos con comunicación a alta velocidad (fig.5), un interfaz virtual de control remoto en tiempo real mediante radioenlace (fig.8) y un sistema de visión artificial mediante realimentación de imágenes de una micro cámara actualmente en desarrollo.

Anteriormente se ha probado con éxito el desarrollo de un interfaz virtual que controla el brazo manipulador de forma remota [1]. Este interfaz ha sido programado mediante una herramienta gráfica que permite incluir de una forma sencilla elementos de control y monitorización de cada uno de los grados de libertad del brazo manipulador.

### 3 CONTROL DEL HARDWARE DE BAJO NIVEL

Gracias al pequeño tamaño de los microcontroladores, es posible incluirlos abordo de los elementos a controlar, reduciendo el número de cables de señal y alimentación que habría que llevar hacia las partes móviles del vehículo robótico como articulaciones o ruedas. Por ello, se tiene un sistema donde cada grupo de circuitos tiene su propio microcontrolador dedicado al control del hardware, creando así una red de dispositivos que es necesario comunicar y sincronizar.

En la figura 3 se muestra el detalle de una de las ruedas controladas por cada microcontrolador esclavo (a). Este dispositivo genera señales PWM que controlan sendos motores de corriente continua por cada microcontrolador, a través de los circuitos en puente. El motor del ángulo de posición (b) está acoplado a una caja reductora en el eje del motor (c), cuya posición es leída por un encoder basado en un potenciómetro de película continua (d). El motor de avance de la rueda (e) [7] proporciona una velocidad de 120 r.p.m y el encoder basado en un sensor óptico reflexivo (f) obtiene la posición del giro.

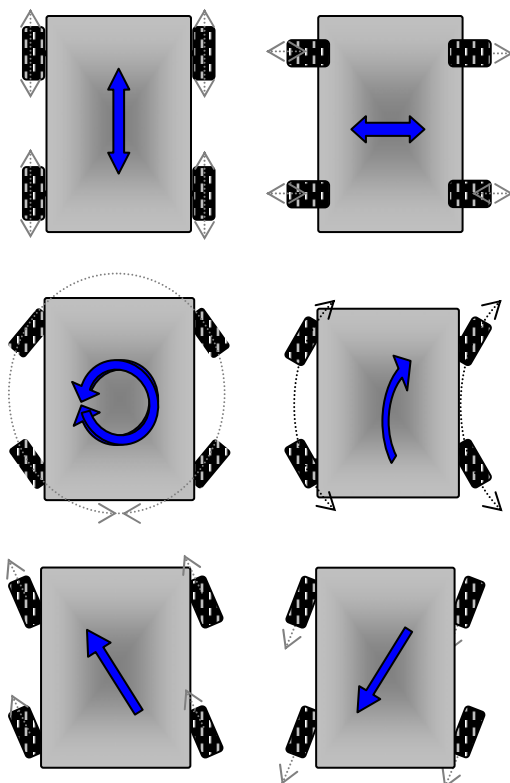


Figura 4: Ejemplos con distintas combinaciones posibles de las posiciones de las ruedas para el desplazamiento.

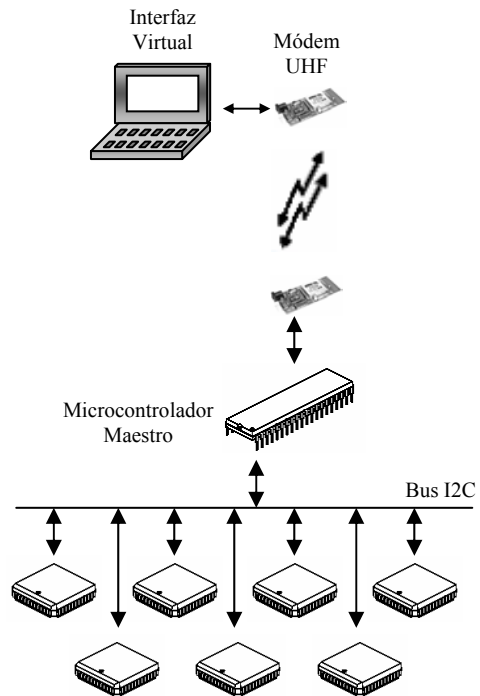


Figura 5: Estructura de la arquitectura hardware de la red de microcontroladores.

Gracias al control individual de cada grupo de microcontroladores, es posible ofrecer distintas combinaciones de las posiciones de las ruedas que nos permitan distintos modos de desplazamiento. Estos modos de desplazamiento son programados en alto nivel, pudiendo seleccionarse a través del interfaz virtual de usuario, la combinación que mejor se adapte a las necesidades del movimiento.

En la figura 4 se puede observar una muestra de distintos desplazamientos. De izquierda a derecha y de arriba hacia abajo podemos ver: movimiento de avance y retroceso lineal, movimiento de desplazamiento lateral, movimiento de rotación, movimiento de giro a la derecha con avance, desplazamiento con ángulo hacia delante y movimiento de retroceso hacia la izquierda.

### 4 DISEÑO DE LA RED DE MICROCONTROLADORES

El esquema de la figura 5 muestra la red de microcontroladores de 8 bits de la serie PIC16F87X de la casa Microchip [10] que han sido utilizados. La red está diseñada en torno a un bus I2C que ha sido configurado en modo rápido a una velocidad de 400 Khz. El modo de direccionamiento elegido permite la comunicación con un número máximo de 128 dispositivos distintos, de los cuales se han necesitado hasta el momento 1 dirección para el maestro y 4

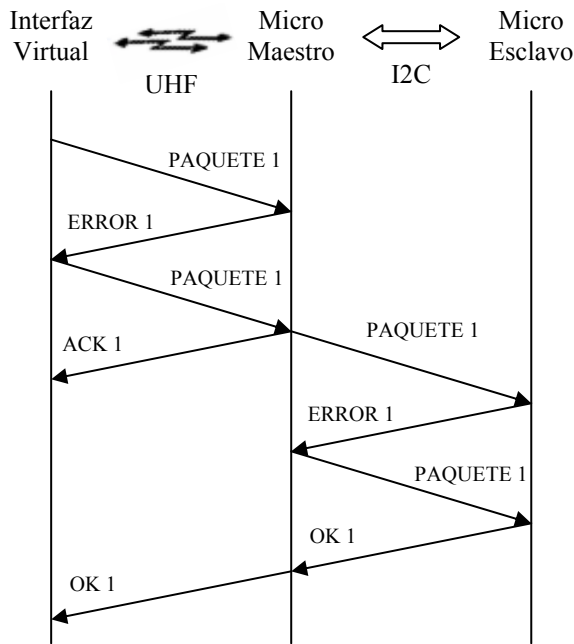
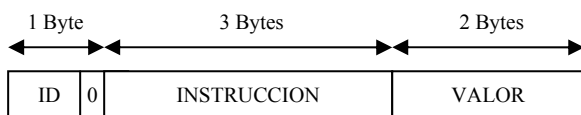


Figura 6: Flujo de datos entre el canal UHF y el bus I2C.

direcciones para los esclavos, quedando el resto de direcciones libres para usos futuros en el prototipo final. Obviamente, la cantidad de direcciones libres permite una gran posibilidad de expansión de la plataforma.

Para la comunicación entre los distintos elementos, se ha diseñado un protocolo orientado a conexión basado en paquetes de error y confirmación ACK (fig. 6). Este protocolo implementa un mecanismo de redirección de mensajes entre el interfaz virtual de usuario y la red de dispositivos esclavos a través del microcontrolador maestro, usando como enlace el canal UHF y el bus I2C.

Dependiendo del camino en el flujo de datos, se tendrán distintos formatos de paquetes. En el caso del envío de paquetes de datos en la comunicación UHF, se ha definido un formato de instrucción identificado por el LSB = 0 del primer octeto:

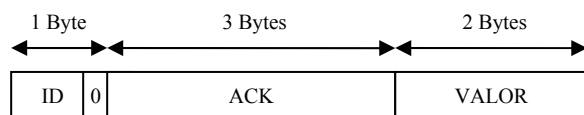


El campo ID define la dirección de 7 bits del microcontrolador de destino; el campo INSTRUCCIÓN define el comando de 3 caracteres en formato ASCII y el campo VALOR configura el

elemento esclavo según la instrucción anterior. Estos caracteres ASCII pueden ser:

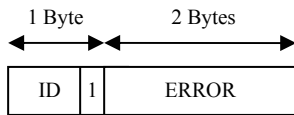
- M1A: solicita que el motor 1 avance a una cierta velocidad, expresada en el campo VALOR en revoluciones por minuto.
- M2R: solicita que el motor 2 retroceda a una cierta velocidad, expresada en el campo VALOR en revoluciones por minuto.
- M3S: solicita la parada del motor 3. El campo VALOR debe tener los 2 bytes = 0.
- M4G: solicita que el motor 4 gire el ángulo especificado en campo VALOR.
- M1V: solicita el cambio de velocidad del motor 1, pero no el sentido.
- M2C: solicita que el motor 1 cambie de sentido, pero manteniendo la velocidad.
- M3X: solicita que el motor 3 avance una distancia, expresada en el campo VALOR en centímetros.
- OK: comando enviado cuando una operación solicitada ha finalizado con éxito.
- EV1: pregunta la velocidad actual del motor 1. En el paquete ACK se devuelve la velocidad en el campo VALOR.
- EG2: pregunta el ángulo actual del motor 1. En el paquete ACK se devuelve la velocidad en el campo VALOR.

Un caso particular de este paquete es la respuesta afirmativa (mediante un mensaje de confirmación respecto de una solicitud anterior) desde un microcontrolador esclavo. Para ello, el campo ID debe contener la dirección del esclavo, el campo INSTRUCCIÓN debe contener los caracteres ACK y el campo VALOR representa el estado de la variable solicitada por el comando previo. Esto es:

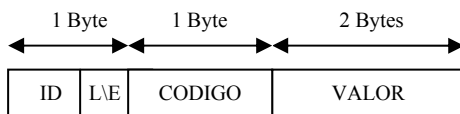


En el caso del envío de paquetes de error, se ha definido un formato de instrucción identificado por el LSB = 1 del primer octeto. El campo ID define la dirección de 7 bits del microcontrolador de origen; el campo ERROR define el tipo de error mediante un valor numérico cuyos códigos pueden ser:

- 0: Dispositivo no conectado.
- 1: Instrucción no válida.
- 2: Instrucción no soportada por el microcontrolador.
- 3: Velocidad fuera de rango.
- 4: Ángulo fuera de rango.



Cuando los paquetes de telecomandos o telemetría entran en el ámbito de la red de microcontroladores, se realiza una conversión de los paquetes que implica la adaptación del campo ID y del LSB del primer octeto (utilizado como bit de lectura/escritura en el bus I2C). Además, el campo de INSTRUCCIÓN se codifica en un único byte. De esta forma el formato del paquete es el siguiente:



## 5 PROGRAMACIÓN DE LOS MICROCONTROLADORES

La programación de los PICs se ha realizado con software de libre distribución, usando funciones desarrolladas en lenguaje C pertenecientes al compilador PCW de la casa CCS [4] (fig. 7). Cada fichero hexadecimal obtenido se ha cargado en el módulo grabador [13] y usando los controladores USB de la placa cargadora [6], se han grabado los programas de control en los microcontroladores conectados a un zócalo ZIF de 40 pines.

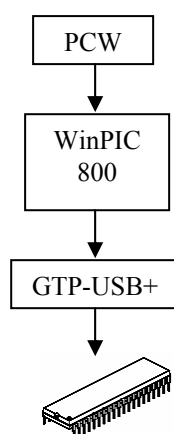


Figura 7: Módulos de programación usados en el cargador USB de los microcontroladores.

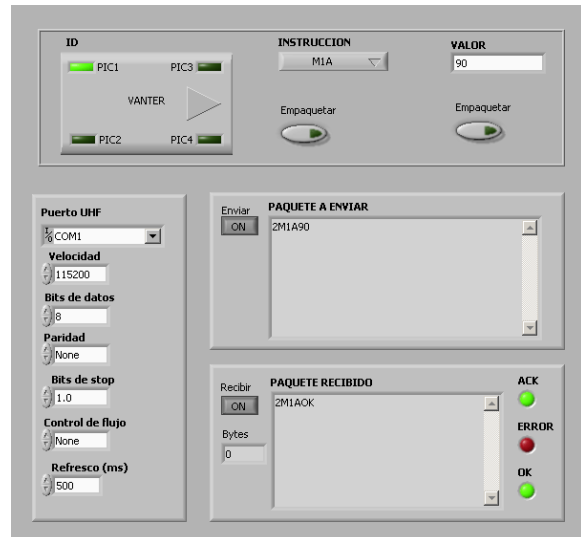


Figura 8: Interfaz virtual de usuario que monitoriza y controla la comunicación UHF.

## 6 INTERFAZ VIRTUAL Y COMUNICACIÓN REMOTA UHF

La interacción en tiempo real entre el cliente remoto y la red de microcontroladores, se realiza mediante una comunicación inalámbrica punto a punto entre el interfaz virtual de usuario (fig. 8) y el microcontrolador maestro por medio de un transmisor UHF [14]. Esta comunicación sirve para el envío de mensajes que transportan datos de telecomandos y de telemetría en modo half-dúplex como si de un cable virtual se tratara.

El protocolo orientado a conexión desarrollado evita la pérdida de paquetes por falta de cobertura o colisiones en el medio, debiendo respetar un tiempo mínimo de 0,5ms antes de volver a realizar otra transmisión. La detección de la señal portadora disponible en el conector de datos del módulo UHF hace posible un control básico de acceso al medio CSMA/CD.

El software desarrollado tanto en el interfaz virtual como en el microcontrolador maestro, empaqueta los mensajes en tramas no mayores de 96 bytes antes de mandarlo por el canal UHF. El módulo transmisor [15] tiene un microprocesador que controla la sintonización RF y proporciona una velocidad de transmisión real de 100 Kbps.

El interfaz virtual de usuario está dividido en varias zonas funcionales. En la zona inferior izquierda del panel principal se dispone de un área de configuración del puerto RS-232, que por defecto se



ha establecido con 1 bit de inicio, 8 bits de datos y 1 bit de parada. En el área superior se dispone de 3 zonas que facilitan la elección de los campos de los paquetes mediante displays y menús desplegables. Una vez empaquetado el mensaje, éste puede enviarse y visualizarse en la zona central del panel principal, obteniéndose los paquetes recibidos en la zona inferior.

## 7 CONCLUSIONES Y LÍNEAS ACTUALMENTE EN DESARROLLO

En este trabajo se ha presentado una plataforma robótica desarrollada con el fin de ensayar y poner a punto, mediante una red de microcontroladores, estrategias de control remoto vía RF.

Esta plataforma constituye un paso más en el proceso de desarrollo de un prototipo de vehículo robótico dotado de brazo manipulador de acrónimo VANTER (Vehículo Autónomo No Tripulado Especializado en Reconocimiento).

VANTER está siendo desarrollado con la idea de cumplir dos objetivos: (1) servir de multiplataforma para hacer ensayos reales de estrategias de control (remoto o no), así como de visión artificial y de adquisición y procesamiento de señales y (2) una vez totalmente operativo, ser un vehículo robótico dotado de sistemas estándares abiertos, con aplicaciones en campos donde tanto las dimensiones como las condiciones de toxicidad ambiental, desaconsejen la presencia de operadores humanos como en minas o gaseoductos por ejemplo.

Durante su desarrollo VANTER se está mostrando como una herramienta excelente de aprendizaje, así como un banco de pruebas real sobre el que están trabajando alumnos de Proyectos Fin de Carrera y doctorandos. La razón estriba en que todo el desarrollo hardware (articulaciones, estructuras, motores, sensores, arquitectura y electrónica) y software de comunicaciones y control está siendo realizado en el grupo de investigación.

## Referencias

- [1] Andújar Márquez, J.M., Mateo Sanguino, T.J. and Aguilar Nieto, F.J., "Virtual Interface for Controlling a Remote Handle Rover", *Proc. IADAT-aci2005 Conf.*, pp.224-228, 2005
- [2] Andújar Márquez, J.M., Mateo Sanguino, T.J. y Aguilar Nieto, F.J., "Control de un Vehículo Robótico Dotado de Brazo Manipulador Mediante un Interfaz Virtual de Usuario", XXVI Jornadas de Automática p.p.743-748, 2005
- [3] Cánovas López, A. y Dorado, V., "Manual de Usuario del Compilador PCW de CCS". Disponible en: [http://picmania.garcia-cuervo.com/recursos/Manual\\_Compilador\\_CCS\\_PICC.pdf](http://picmania.garcia-cuervo.com/recursos/Manual_Compilador_CCS_PICC.pdf)
- [4] C Compiler Reference Manual. Custom Computer Services Inc., 2005. Disponible en: <http://www.ccsinfo.com>
- [5] Display LCD Serie I2C 4x20. Disponible en : <http://www.superrobotica.com>
- [6] GTP-USB[plus]. Disponible en: <http://perso.wanadoo.es/siscobf/index.html>
- [7] Motor DC Reductor de 12V y 120 RPM. Disponible en: <http://www.superrobotica.com>
- [8] O. González Martínez, "Proyecto: Electrococo", Seminario de Microrobots, Univ. de Alcalá de Henares, 2006
- [9] Packaging Specification. Microchip Technology Inc., 2006. Disponible en: <http://ww1.microchip.com/downloads/en/PackagingSpc/00049AF.pdf>
- [10] PIC16F87X Data Sheet. 28/40-Pin 8-bit CMOS Flash Microcontrollers. Microchip Technology Inc., 2001. Disponible en: <http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>
- [11] S. Bowling and R.L. Fischer, "An I2C Network Protocol for Environmental Monitoring", Microchip Technology Incorporated, 2002
- [12] The I2C-Bus Specification version 2.1, January 2000. Philips Semiconductors. Disponible en: [http://www.semiconductors.philips.com/products/interface\\_control/i2c/](http://www.semiconductors.philips.com/products/interface_control/i2c/)
- [13] WinPIC800 v 3.58. Disponible en : <http://perso.wanadoo.es/siscobf/index.html>
- [14] WIZ-SML-1A Instruction Manual, Rev. A, 01-04-2002. AUREL S.P.A. <http://www.aurel.it/>
- [15] XTR-434xxx Instruction Manual, Rev. A, 30-10-2001. AUREL S.P.A. <http://www.aurel.it/>