



## Vehículo con sistema de comunicaciones y control basado en microcontrolador para desarrollo de funcionalidades en docencia

Carlos Corrales Alba; Manuel Jesús López Sánchez  
Universidad de Cádiz

Departamento de Ingeniería en Automática, Electrónica, Arquitectura y Redes de Computadores

España

Correo-e: carlos.corrales@uca.es; manueljesus.lopez@uca.es

**Abstract.** En este trabajo se presenta un pequeño vehículo diseñado para su empleo en varias prácticas de laboratorio en la asignatura "Microprocesadores y Microcontroladores" del Grado en Ingeniería Radioelectrónica de la Universidad de Cádiz. A partir de dos motores, un chasis, un microcontrolador, la correspondiente circuitería de control y un transceptor NRF24L01, que trabaja en la banda ISM 2.4GHz, se construye un vehículo con el sistema de comunicaciones y control. Su gobierno se realiza mediante un joystick conectado a otro microcontrolador, empleando el software Niple de programación visual. Finalmente, se han incluido varias aplicaciones navales desarrolladas tomando como punto de partida este sistema.

### 1 Introducción

Son múltiples las aplicaciones que se pueden desarrollar para la docencia en microcontroladores, pero siempre hay que tener en cuenta el entorno de la asignatura. En este trabajo, se ha enfocado hacia una Ingeniería Radioelectrónica, y por tanto, ha de predominar el carácter electrónico y práctico de la misma frente a la programación informática. Es por ello por lo que se va a emplear el software *Niple* [6,7,8], con objeto de realizar una programación didáctica y visual del sistema. Aunque se va a trabajar con un vehículo, la práctica también se puede enfocar hacia cualquier aplicación naval en la que se necesite la transmisión o recepción de datos de forma digital o el control de un vehículo.

Así mismo, es importante la integración de trabajos y conocimientos de las distintas asignaturas que forman el Grado, con objeto de que el alumno obtenga una visión global de casos reales. Para ello, la metodología docente aplicada en esta práctica va buscando el inculcar al alumno una serie de principios y técnicas educativas que le sirvan para interpretar la realidad y proyectar sobre ella los significados que va construyendo (aprendizaje cognitivo), aprender lo relevante de un problema (aprendizaje significativo), descubrir los conceptos y sus relaciones y reordenarlos para adaptarlos a su esquema cognitivo (aprendizaje por descubrimiento), construir el conocimiento (aprendizaje constructivista) y, finalmente, crear un ambiente de motivación, reflexión, creatividad, trabajo en grupo, estudio de casos y discusión en grupo, favoreciendo el razonamiento lógico [1, 2, 3, 4, 5].

### 2 Descripción del sistema

Se ha diseñado un vehículo (Figura 2) a partir de dos motores y un chasis de metacrilato, similar a los que se pueden encontrar en tiendas especializadas. Se le ha añadido una placa protoboard, sobre la que se ha instalado toda la circuitería electrónica necesaria: microcontrolador, control de motores, módulo de comunicaciones y alimentación. Así mismo, en un entrenador de microcontroladores se ha instalado otro módulo de comunicaciones y un joystick analógico para el control del vehículo.



Figura 1: Aplicaciones navales del sistema

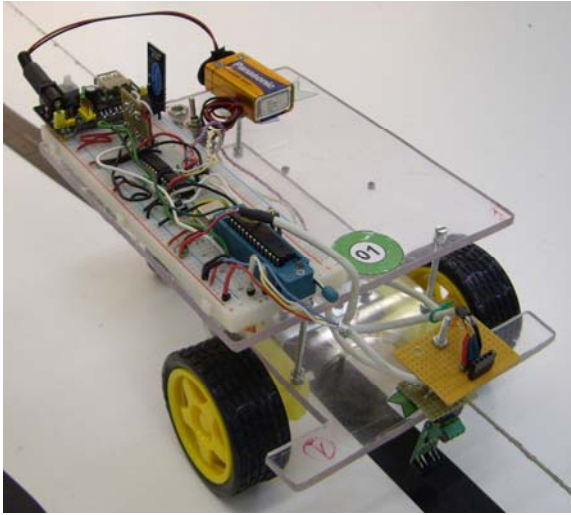


Figura 2: Vehículo radiocontrolado

### 3 Elementos del sistema

Los elementos empleados en esta práctica son los siguientes:

#### 3.1 Microcontrolador

Este trabajo se ha realizado en el laboratorio de "Sistemas Digitales y Control por Ordenador" de la Escuela de Ingenierías Marina, Náutica y Radioelectrónica de la Universidad de Cádiz, y se ha desarrollado sobre el entrenador-grabador *USB-PIC'School* de Ingeniería de Microsistemas Programados S.L. (Figura 3) (www.microcontroladores.com) empleando un microcontrolador *16F886* de *Microchip*.



Figura 3: Entrenador USB-PIC'School

Los recursos empleados en el microcontrolador del vehículo son, por una parte, tres bits del puerto A (RA2, RA1 y RA0) para el control de los motores, y seis bits para el transceptor *NRF24L01* en modo recepción (cinco del puerto C: RC5, RC4, RC3, RC2 y RC1 y uno del puerto B: RB0 para la interrupción). El microcontrolador empleado en el entrenador para el control del vehículo, usa dos entradas analógicas para el joystick (RA1 y RA0) y los mismos bits descritos anteriormente para el correspondiente transceptor en modo emisión.

Para generar la frecuencia general de trabajo del microcontrolador (4MHz), se ha empleado un oscilador integrado a cristal de cuarzo (también llamado *Canned Oscillator*) encapsulado en formato DIP8 y conectado al pin 9.

#### 3.2 Vehículo

En un chasis de metacrilato se han colocado dos motores de corriente continua (Figura 4) con una relación reductora de 1:42 y que pueden funcionar a una tensión de 3 a 6V, con un consumo de 120-140mA y una velocidad de 200rpm a 5V.



Figura 4: Motores

Con este consumo, hay que emplear un circuito amplificador en intensidad basado en el circuito integrado *L293B*, junto a dos inversores *Trigger Schmitt (74HC14N)* para cambiar el sentido de giro de los mismos. En la Figura 5 se muestra el circuito completo.

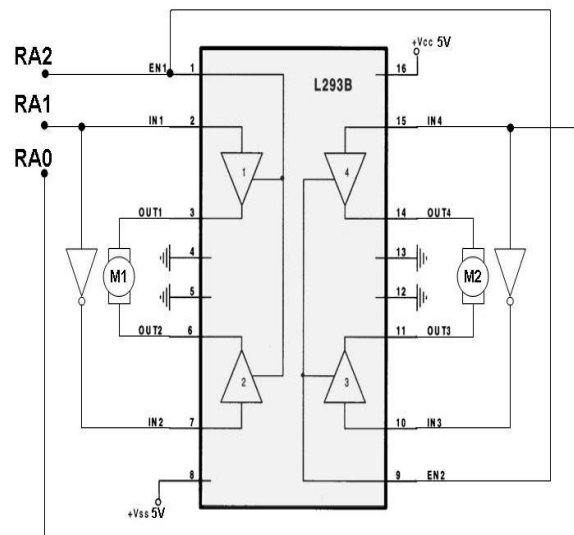


Figura 5: Circuito de control de los motores

La siguiente tabla muestra el modo de funcionamiento del vehículo, de acuerdo con el circuito anterior. Se observa que para que el vehículo esté parado, bastará con poner en el puerto A del microcontrolador cualquier número menor de 4 y para que gire a la izquierda, por ejemplo, se deberá poner el número 5 en ese puerto.



RA2	RA1	RA0	n°	MODO
0	x	x	< 4	STOP
1	1	1	7	ADELANTE
1	0	1	5	IZQUIERDA
1	1	0	6	DERECHA
1	0	0	4	ATRÁS

Tabla 1: Modos de funcionamiento

Es necesario tener en cuenta que se deben colocar condensadores en los terminales de los motores con objeto de filtrar el ruido eléctrico que generan los mismos y que puede interferir en el funcionamiento del microcontrolador y, sobre todo, del transceptor. También es recomendable utilizar cables de suficiente sección y poca longitud.

### 3.3 Alimentación

La alimentación se ha realizado mediante una batería tipo 6LR61 de 9V. Para obtener los 5V para el microcontrolador y 3,3V para el transceptor se ha empleado un módulo de alimentación para placas de prototipos MB102 basada en los circuitos AMS1117-3.3/5.0 (Figura 6). En el entrenador se ha empleado otro AMS1117-3.3, ya que es el regulador con el que se han obtenido los mejores resultados.

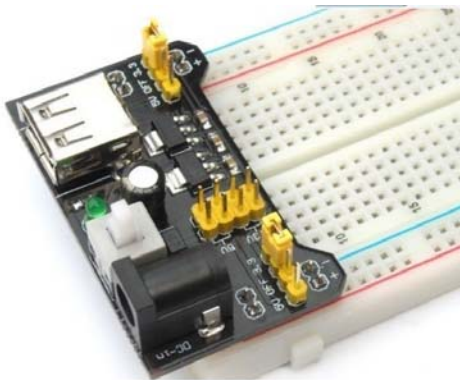


Figura 6: Módulo de alimentación

### 3.4 Joystick

El joystick (Figura 7) empleado para gobernar el vehículo tiene una alimentación de 5V y proporciona dos señales analógicas (una para el eje X y otra para el Y) en el rango 0-5V lineal (2,5V en la posición intermedia). También tiene una salida digital para el pulsador.



Figura 7: Joystick de gobierno del vehículo

En el entrenador *USB-PIC School* se conecta la salida del eje X a la entrada RA0 del microcontrolador y la del eje Y a RA1.

### 3.5 Módulo de comunicaciones

El módulo de comunicaciones empleado (uno en el vehículo para la recepción y el otro en el entrenador para la transmisión) está basado en el transceptor NRF24L01 de *Nordic Semiconductors* (<http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01>), e incluye toda la electrónica necesaria para su funcionamiento y una antena tipo microstrip (Figura 8). Su comunicación con el microcontrolador es a través de un bus SPI, funciona en la banda ISM a 2,4GHz con 128 canales de transmisión posibles, su alimentación típica es 3,3V, tiene una velocidad máxima de transmisión de 2Mbps y un radio de alcance de hasta 40 m a 1Mbps en el mejor de los casos.

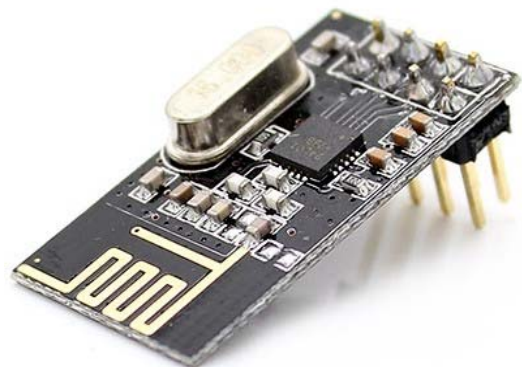


Figura 8: Transceptor NRF24L01

Este transceptor emplea seis líneas de E/S del microcontrolador, tal y como se muestra en la Figura 9.



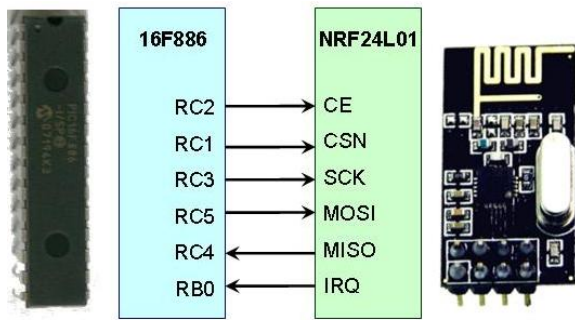


Figura 9: Conexión microcontrolador-transceptor

A la hora de elegir uno de los 128 canales de comunicación, hay que tener en cuenta que en el laboratorio hay varios grupos de prácticas trabajando simultáneamente y, por tanto, va a existir más de una comunicación entre módulos en diferentes canales (entre un canal y el anterior o posterior hay una diferencia de 1 MHz). Por ello, y para evitar interferencias, hay que dejar un espaciamiento de 2 canales entre los utilizables quedando de este modo un total de 32 canales útiles. También hay que tener en cuenta las posibles interferencias que puedan causar otros dispositivos que trabajan en la banda ISM 2.4GHZ, como WIFI o Bluetooth.

## 4 Programación

Para la programación del microcontrolador se ha empleado el software *Niple* v5.7.3 [6,7,8] que permite la programación de microcontroladores de una forma visual, basada en diagramas de flujo.

Como se ha indicado anteriormente, el ámbito de esta práctica se engloba en un Grado en Radioelectrónica, y el alumno no tiene por qué tener conocimientos de lenguajes de alto nivel como, por ejemplo, *C* o *BASIC*. El *Niple* es también un lenguaje de alto nivel que permite programar microcontroladores PIC de forma visual y, por tanto, no es necesario conocer otro lenguaje de alto nivel aunque, obviamente, si son necesarios tener conocimientos básicos de microcontroladores a nivel de ensamblador (bajo nivel).

### 4.1 Configuración del microcontrolador

En el microcontrolador para la transmisión de datos, habrá que configurar dos bits (RA0 y RA1) como entradas A/D con una resolución de 8 bits para los dos ejes del joystick. Salvo los bits RB0 (interrupción) y RC4 (*MISO* del bus SPI) que son entradas, los restantes serán salidas. Para el microcontrolador del vehículo, sólo RB0 y RC4 serán entradas. Los restantes bits serán salidas (el control de los motores es a través del Puerto A). Todo esto se observa en la Figura 10.

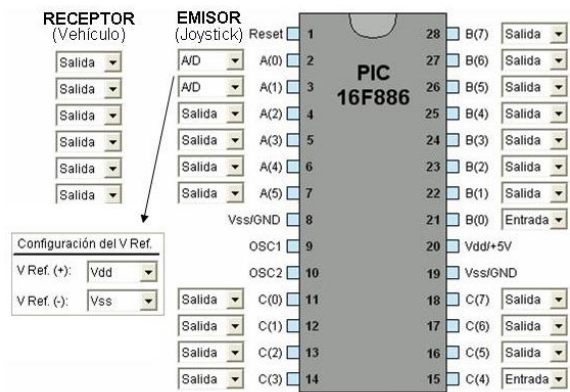


Figura 10: Configuración de los microcontroladores

### 4.2 Módulo de comunicaciones

Para su programación se ha empleado la librería desarrollada por la empresa *Bizintek Innovas S.L* [9]. Es de tipo Open Source, pudiéndose, por tanto, distribuir y/o modificar bajo los términos de la licencia GNU. La librería en lenguaje ensamblador se ha integrado en el software *Niple* como varias subrutinas de usuario insertando las correspondientes instrucciones.

En total son 14 las subrutinas empleadas, siendo 5 las principales (las 9 restantes son, en general, subrutinas de las 5 anteriores). Éstas son: *R\_CONFIG\_SPI*, *RF\_CONFIG*, *RF\_ON*, *RF\_SEND* y *RF\_RECEIVE*.

La subrutina *RF\_CONFIG\_SPI* (Figura 11) se emplea para configurar los diferentes parámetros del bus SPI y los pines del microcontrolador, tal y como se detalla en el apartado 3.5.

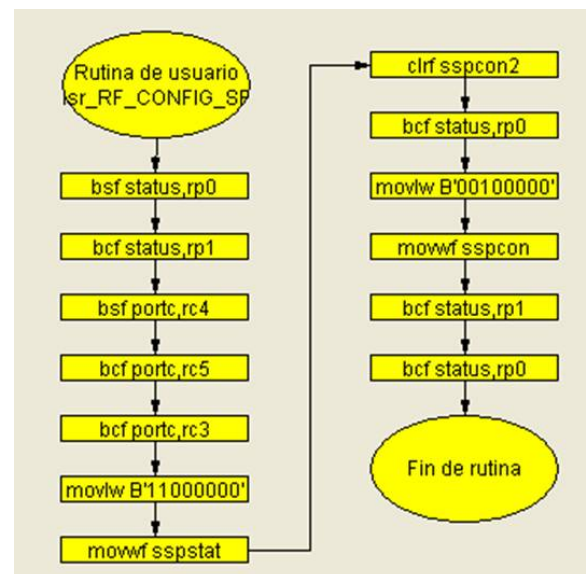


Figura 11: Subrutina *RF\_CONFIG\_SPI*



# Tercer Congreso Virtual, Microcontroladores y sus Aplicaciones

La subrutina *RF\_CONFIG* (Figura 12) configura la dirección del dispositivo (ID) en la variable *RF\_DIR* (un valor entre 0x01 y 0xFE), el canal a utilizar en la comunicación con la variable *RF\_CHN* (valor comprendido entre 0x00 y 0x7F, es decir, entre 0 y 127), velocidad, potencia de transmisión y otros parámetros.

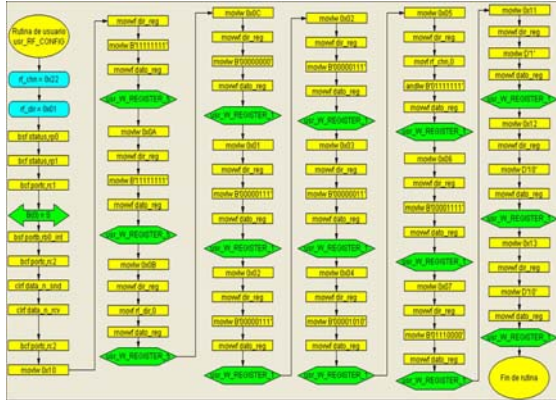


Figura 12: Subrutina *RF\_CONFIG*

Para activar el transceptor para la transmisión/recepción de datos se emplea la subrutina *RF\_ON* (Figura 13). Tras llamar a esta subrutina hay que esperar unos 3ms a que el transceptor esté listo para funcionar.

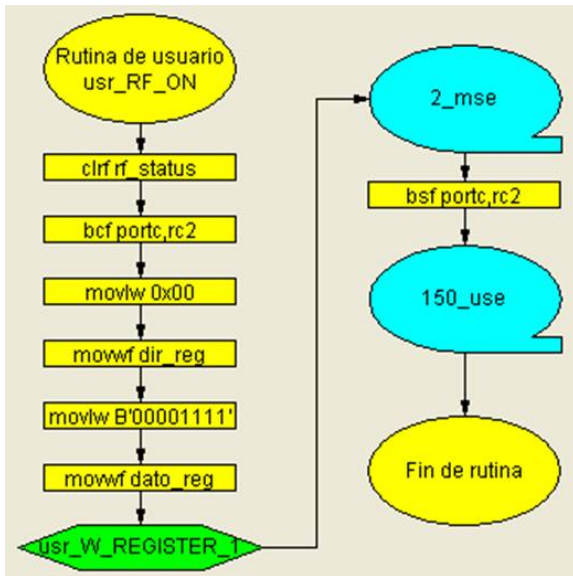


Figura 13: Subrutina *RF\_ON*

Los datos se transmiten en paquetes de 8 bytes mediante la subrutina *RF\_SEND*. En la librería original [9], los datos transmitidos y recibidos se asocian a una misma variable, denominada *RF\_DATA<sub>n</sub>*, siendo *n=0,1,...,7* (Figura 14). En este trabajo se ha decidido definir las por separado, de forma que para esta subrutina, los datos que se quieren transmitir se asocian a las ocho variables denominadas *RFO\_DATA<sub>n</sub>*. También hay que indicar

en la subrutina, la dirección del equipo al que se quiere enviar los datos mediante la variable *RF\_DIR*.

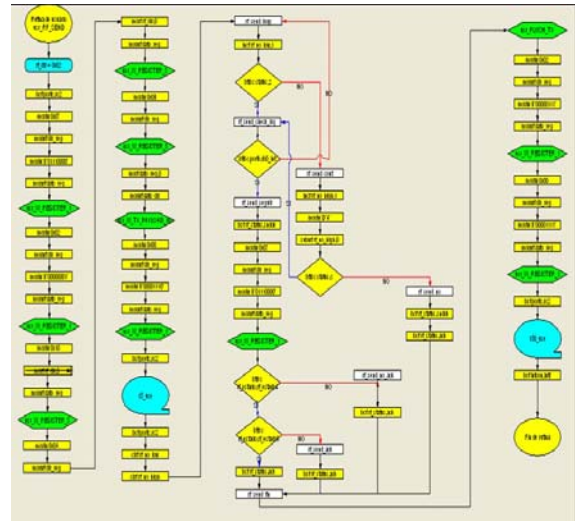


Figura 14: Subrutina *RF\_SEND*

Esta subrutina proporciona dos bits (o flags) a través de la variable *RF\_STATUS* que son de gran ayuda a la hora de verificar la transmisión de los datos: *SNDOK* y *ACK*. El primero indica si el transceptor ha realizado el envío de los datos (1) o no ha podido hacerlo (0). El segundo indica si se ha recibido la confirmación de que el receptor recibió los datos correctamente (1) o no (0).

La última subrutina principal es *RF\_RECEIVE* (Figura 15), que comprueba si se ha producido una recepción de datos y, en caso afirmativo, los almacena en la ocho variables denominadas *RFI\_DATA<sub>n</sub>*.

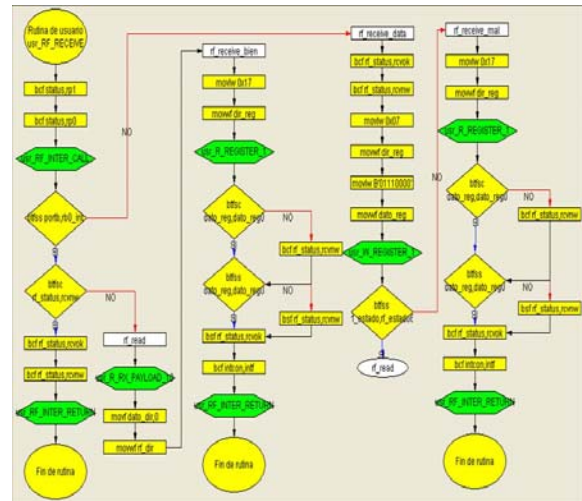


Figura 15: Subrutina *RF\_RECEIVE*

De forma similar a la anterior subrutina, en ésta se ven afectados los bits *RCVOK* y *RCVNW* del registro *RF\_STATUS*. El primero informa que la recepción es correcta (1) o no (0) y el segundo indica si la pila del transceptor está vacía y no hay mensajes en espera







### 4.5 Recepción de datos con interrupción

Cuando se necesita que la transmisión de datos sea bidireccional, hay que trabajar activando las interrupciones. La librería descrita anteriormente [9] incluyen subrutinas para estos casos, pero se ha preferido trabajar las interrupciones directamente con el programa *Niple* (Figura 19), dado su facilidad y comodidad en el manejo de las mismas.

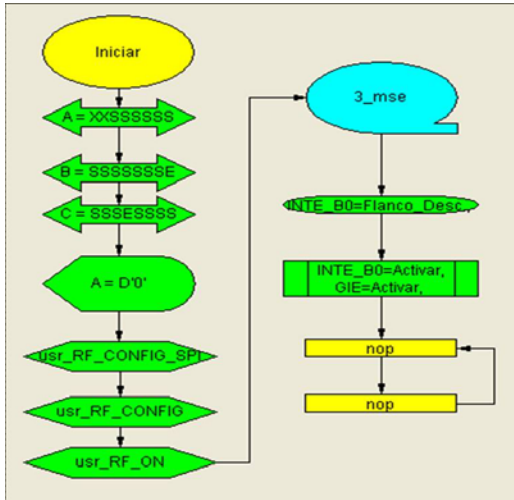


Figura 19: Programa del microcontrolador del vehículo con interrupción

Hay que tener en cuenta que la señal de interrupción generada por el transceptor es complementada y, por tanto, hay que configurar la interrupción en RB0 por flanco descendente. Una vez que se ha producido la interrupción, *Niple* pasa a una subrutina de *Interrupción por RB0*, en la que primero se desactiva (para que puedan procesarse los datos sin problemas), luego se leen los datos de forma similar a la del apartado anterior, para finalmente, volver a activarla y salir de la subrutina (Figura 20).

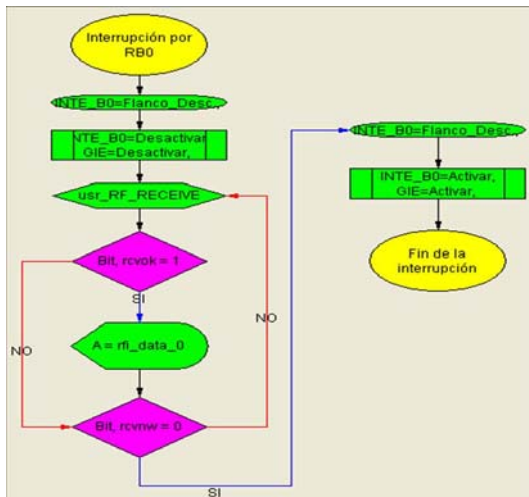


Figura 20: Subrutina de Interrupción por RB0

## 5 Aplicaciones navales

A continuación se presentan varias aplicaciones prácticas de ámbito naval desarrolladas a partir del sistema descrito en este trabajo.

### 5.1 Autopiloto naval

Un autopiloto naval es un equipo que se emplea en un buque para mantener el rumbo elegido, actuando sobre el control del timón. En este trabajo, el buque se puede sustituir por el vehículo descrito anteriormente y fijar una trayectoria. Para determinar el rumbo real se emplea un compás digital *CMPS03* de la empresa *Devantech Ltd.* (Figura 21), que mide su posición en grados respecto del norte magnético.

La lectura del compás se puede realizar de dos modos: a partir de la salida PWM, cuya anchura es proporcional a la posición, o bien a través del bus I2C. En [5] se desarrolla la programación con *Niple* para la lectura del compás por el primer método. A partir de la versión 5.7, *Niple* incluye la implementación de los buses I2C y SPI de forma directa. De este modo y para este dispositivo, en primer lugar hay que solicitar el rumbo actual escribiendo la dirección del registro 0x01 (rumbo en un solo byte) en la dirección 0xC0 del módulo (modo escritura). Luego habrá que leer el byte solicitado en la dirección 0xC1 del módulo (modo lectura) y asociarla a un registro del microcontrolador, para posteriormente escalarlo en dos registros (de 0-255 a 0-359°).

Fijando el punto de consigna (rumbo deseado) y conocido el rumbo real, se puede calcular la señal de error (diferencia entre los dos). Según sea el signo de la señal de error, se podrá controlar adecuadamente la dirección del vehículo.

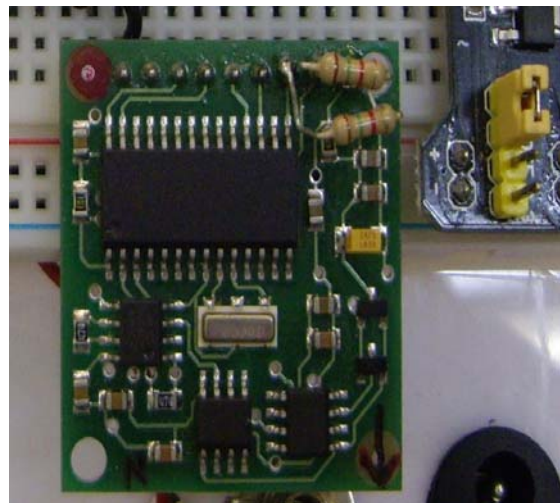


Figura 21: Compás digital *CMPS03*

## 5.2 Radiobaliza de emergencia

El sistema internacional de satélites *COSPAS-SARSAT* y *GEOSAR* (estacionarios) capta las señales de emergencia emitidas por radiobalizas que se activan de forma automática al sumergirse en el agua, o bien de forma manual. Actualmente emiten en la frecuencia de 406MHz y transmiten un código de identificación único que personaliza a la embarcación propietaria, y algunos modelos incluyen un GPS para indicar la posición exacta de la emergencia.

Para simular la radiobaliza, se añade al vehículo un módulo receptor GPS 28500 de la firma *Polstar* [5] (Figura 22). La transmisión de los datos se realiza por RS-232 (a 4800 baudios, 8 bits de datos, un bit de parada y sin paridad) según el protocolo *NMEA 0183* [10]. De los 47 primeros bytes de la trama *RMC* del GPS, del 17 al 40 se corresponden con la latitud, orientación (N/S), longitud y orientación de la misma (E/W). Por tanto, para conocer la posición exacta se necesitan 24 bytes, es decir, el transceptor deberá transmitir 3 paquetes de 8 bytes cada uno, más uno que incluya su identificación o ID. La lectura del GPS se describe en [5].



Figura 22: GPS 28500 de *Polstar*

En el entrenador *USB-PIC School* que se emplea para controlar el vehículo, se conecta la pantalla LCD al Puerto B para mostrar la posición exacta en la que se encuentra el vehículo. Dado que la comunicación es bidireccional, ambos microcontroladores habrá que programarlos con las interrupciones.

## 5.3 Estiba de contenedores

En los puertos de Hamburgo y Rotérdam, se ha automatizado la estiba (carga/descarga) de contenedores mediante el uso de vehículos guiados automáticamente, también conocidos como *AGVs*. Son transportadores de contenedores que los mueven desde la zona de descarga hasta la de almacenamiento de forma automática (Figura 23). Por tanto, otra aplicación del vehículo diseñado en este trabajo sería asociar su movimiento a una

trayectoria concreta, por ejemplo, una línea oscura en el suelo.



Figura 23: *AGVs* en la estiba de contenedores

Para ello, se ha instalado en el frontal del vehículo (Figura 2) un doble sensor de reflexión *MSE-110.2* de luz IR, tal y como se observa en la Figura 1. Cuando la luz es dispersada o absorbida por una superficie oscura genera una salida lógica de nivel "1", y cuando la luz es reflejada por una superficie clara, nivel "0". También se puede añadir el sensor de distancias del apartado 5.5, para evitar el choque entre vehículos con la misma trayectoria.

## 5.4 Posicionamiento de antenas

Una antena parabólica se puede posicionar para seguir a un satélite mediante dos servomotores tal y como se muestra en la Figura 24.

Un sistema similar con dos servos se puede colocar en el vehículo de forma que, por ejemplo, si se adapta un pequeño panel solar, siga la posición de una fuente de luz midiendo la tensión generada en la placa y enviándola al entrenador *USB\_PIC School*. Este valor lo procesará el microcontrolador y el correspondiente movimiento de los servos se enviará como 2 bytes al transceptor del vehículo. Una vez recibidos esos valores, el microcontrolador del vehículo los procesará mediante PWM para realizar el posicionamiento del panel solar.



Figura 24: Posicionamiento de antenas





## 5.5 Sónar

Un sónar se emplea para navegar, comunicar o detectar objetos sumergidos en el agua, mediante la propagación de sonido en la misma. Tomando esto como base, y empleando un medidor de distancias ultrasónico HC-SRF04 se puede desarrollar una práctica. Este medidor se puede colocar en el sistema de posicionamiento de antenas descrito anteriormente (Figura 23) y medir distancias, así como enviar el valor medido mediante el transceptor.

Para medir distancias con este dispositivo, hay que aplicar un pulso de disparo de  $10\mu\text{s}$  en la correspondiente entrada y luego medir (con el *TIMER* del microcontrolador) el tiempo que está en alto la salida de *ECO*, que es proporcional a la distancia entre el dispositivo y el objeto detectado. En la referencia [8] se muestra la programación completa del mismo, aunque *Niple* también incluye el módulo correspondiente.

## 5.6 Estación meteorológica

A partir de un sensor de temperatura/humedad (*SHT11*), un sensor de presión atmosférica (*MPX4115AP*) y un anemómetro, se puede instalar en el vehículo una estación meteorológica. El anemómetro se puede simular mediante un encoder adaptado a una de las ruedas del vehículo para medir su velocidad (ambos funcionan con el mismo principio) (Figura 25). El sensor *SHT11*, que funciona a través de un bus I2C, está incluido como dispositivo en *Niple* y el sensor *MPX4115AP* tiene una salida analógica (0,2-4,8V) en el rango 15-15KPa. Los datos se almacenan en varias variables de 1 byte que son transmitidas por el transceptor hacia una estación base instalada en el entrenador *USB-PIC'School*.

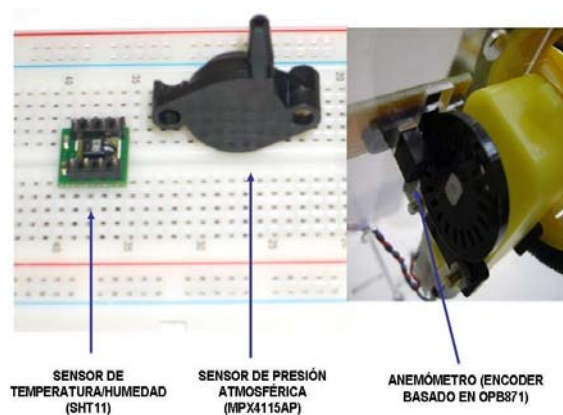


Figura 25: Elementos de la estación meteorológica

## 6 Conclusiones

Este trabajo permite conocer y ver en funcionamiento aplicaciones reales de microcontroladores PIC, facilitando la integración de conocimientos de distintas asignaturas del Grado en Ingeniería Radioelectrónica. Así mismo, aumenta el grado de auto implicación del alumno, su creatividad, y facilita la motivación y experiencia, haciendo que el alumno encuentren sentido a lo que estudia ya que aprende haciendo y descubriendo. El punto de partida es el diseño del vehículo y su sistema de control, y posteriormente se desarrolla una serie de aplicaciones prácticas navales. Además, el material empleado es fácil de obtener, tiene coste reducido y existe amplia bibliografía sobre el mismo, lo que hace que sea muy accesible para el alumno.

Dada la orientación de la titulación, en la que la programación informática no tiene mucho peso, la forma de programar de *Niple*, basada en diagramas de flujo, es idónea para este tipo de estudios. Además, el alumno no va a necesitar grandes conocimientos de lenguaje ensamblador para realizar multitud de casos prácticos.

## Referencias

- [1] Terrón J., Corrales C., López M. J., Romero E., Moreno F., Priam M., "Aplicación de la diagramación cognitiva integrada en la formación mediante uso del software comercial de simulación y diseño técnico", *Actas del XVI Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas*, Cádiz, España, (2008).
- [2] Terrón J., "Metodologías de aprendizaje basadas en el conocimiento experto y la hipermedia", *Tesis Doctoral*, Universidad de Cádiz, España, (1995).
- [3] Terrón J., Corrales C., Ortega M. J., Machado P., Milla M. E., García J., "Metodologías para la integración del conocimiento y aprendizaje en diferentes proyectos de ingeniería mediante informática técnica y multimedia", *Actas del IV Congreso Internacional de Ingeniería de Proyectos*, Córdoba, España, (1998).
- [4] Corrales C., Terrón J., López M.J., "Modelado de procesos virtuales controlados por microcontrolador programado con lenguaje visual", *Actas de las XXXI Jornadas de Automática*. Jaén, España, (2010).

- [5] Corrales C., Terrón J., López M.J., "Diseño de un autopiloto naval para formación y aprendizaje integrado", *Actas del Segundo Congreso Virtual: Microcontroladores y sus Aplicaciones*, Universidad Tecnológica Nacional, Facultad Regional Paraná, Argentina, (2010).
- [6] Corrales C., Terrón J., López M.J., "Modelado de procesos virtuales controlados por microcontrolador programado con lenguaje visual", *Actas de las XXXI Jornadas de Automática*. Jaén, España, (2010).
- [7] Cano J., "Niple Software, Programación visual de microcontroladores", *Primer Congreso virtual de microcontroladores y sus aplicaciones*, Universidad Tecnológica Nacional, Facultad Regional Paraná, Argentina, (2009).
- [8] González M., Corrales C., "Medidas de distancias empleando un medidor ultrasónico con indicación numérica, luminosa y sonora de una distancia fijada", *Actas del Segundo Congreso Virtual: Microcontroladores y sus Aplicaciones*, Universidad Tecnológica Nacional, Facultad Regional Paraná, Argentina, (2010).
- [9] Bizintek Innova S.L., Manual Módulo BZIRF2GH4, (Febrero 2007).
- [10] Cicuéndez, D., Cagigal, N., Velasco, F. J., Casas, A., "El Standard NMEA 0183 en la integración de la instrumentación del buque", *I Congreso Nacional de Investigación e Innovaciones Tecnológicas en el Ámbito Marítimo*, Barcelona, España, (1997).



## Tercer Congreso Virtual, Microcontroladores y sus Aplicaciones

“Copyright ©2014. “Carlos Corrales Alba”: El autor delega a la Organización del Tercer Congreso Virtual de Microcontroladores la licencia para reproducir este documento para los fines del Congreso ya sea que este artículo se publique en el sitio web del congreso, en un CD o en un documento impreso de las ponencias del Segundo Congreso Virtual de Microcontroladores.