

Desarrollo de aplicaciones con programación visual de microcontroladores para control de procesos

Carlos Corrales Alba^a, Manuel Jesús López Sánchez^b y Agustín Conseglere Castilla^c

^a*Escuela de Ingenierías Marina, Náutica y Radioelectrónica, Universidad de Cádiz, Polígono Río San Pedro s/n, 11510 Puerto Real (Cádiz), carlos.corrales@uca.es*

^b*Escuela Superior de Ingeniería, Universidad de Cádiz, Avda. Universidad de Cádiz 10, 11519 Puerto Real, Cádiz, manueljesus.lopez@uca.es*

^c*Escuela Superior de Ingeniería, Universidad de Cádiz, Avda. Universidad de Cádiz 10, 11519 Puerto Real, Cádiz, agustin.conseglere@uca.es*

Abstract

The aim of this paper is to present several applications used in Microcontrollers practical sessions of the Degree in Radioelectronics Engineering of the University of Cadiz. Block diagrams prove to be a suitable visual tool in programming especially in academic degrees where computer programming is not provided with a relevant specific weight. This is applicable to transducers, actuators, communication networks and industrial processes. The material used is easy to acquire, has reduced cost and there is extensive literature on it, being readily accessible for the students to carry out both course projects and the final degree project.

Keywords: *technical education teaching and training, microcontrollers, application, programming.*

Resumen

El objetivo de este trabajo es presentar varias aplicaciones utilizadas en las clases prácticas de microcontroladores del Grado en Ingeniería Radioelectrónica de la Universidad de Cádiz. La programación es visual con diagramas de bloque, siendo idónea para titulaciones en las que la programación informática no tiene mucho peso. Se aplica a una serie de transductores, actuadores, redes de comunicaciones y procesos industriales. El material empleado es fácil de obtener, tiene coste reducido y existe amplia bibliografía sobre el mismo, siendo muy accesible para la realización de proyectos y trabajos fin de grado de los alumnos.

Palabras clave: *docencia, microcontroladores, aplicaciones, programación.*

Introducción

En 3^{er} curso del Grado en Ingeniería Radioelectrónica de la Universidad de Cádiz está la asignatura “*Microprocesadores y Microcontroladores*”, cuyo objetivo final es la integración de trabajos y conocimientos de diferentes disciplinas de forma que el alumno obtenga una visión global de casos reales. Con este fin, la metodología docente aplicada en esta práctica busca inculcar al alumno una serie de principios educativos (Terrón, 2008; Corrales, 2010), tales como: aprendizaje significativo, aprendizaje cognitivo, aprendizaje por descubrimiento, aprendizaje constructivista y la creación de un ambiente de motivación, reflexión, creatividad, trabajo en grupo, estudio de casos y discusión en grupo, de forma que se favorezca el razonamiento lógico.

Para la docencia de prácticas de microcontroladores se pueden desarrollar multitud de aplicaciones, pero hay que tener en cuenta dos factores: el entorno de la asignatura y la funcionalidad del material de prácticas. En primer lugar, en este Grado predomina el carácter electrónico y práctico frente a la programación informática. Por este motivo, para la programación de los microcontroladores se va a emplear un software basado en la programación visual por diagramas de bloques, realizando una programación didáctica y visual del sistema. En segundo lugar, las aplicaciones que se han desarrollado, tienen un enfoque desde un punto de vista naval y tecnológico.

Descripción del sistema

El tipo de programación empleada hace que el presente trabajo se realice con microcontroladores PIC de *Microchip*, en lugar de trabajar con la plataforma *Arduino*, que es la que actualmente predomina en el mundo de los microcontroladores. Se va a trabajar fundamen-

talmente con el modelo *16F886*, aunque también se utilizará el modelo *18F4550* cuando se necesite un mayor número de entradas y salidas.

Como plataforma de desarrollo y programación se ha empleado el laboratorio *USB-PIC'School* (Ingeniería de Microsistemas Programados S.L) (Figura 1). En esta plataforma, se pueden conectar y asociar los periféricos a las distintas entradas y salidas del microcontrolador mediante un sencillo cableado sin tener que soldar. Por otra, integra el hardware necesario para la depuración y/o grabación del microcontrolador a través del puerto USB del ordenador, siendo compatible con el *PICKIT2* de *Microchip*.

Figura 1 Plataforma de desarrollo USB-PIC'School.

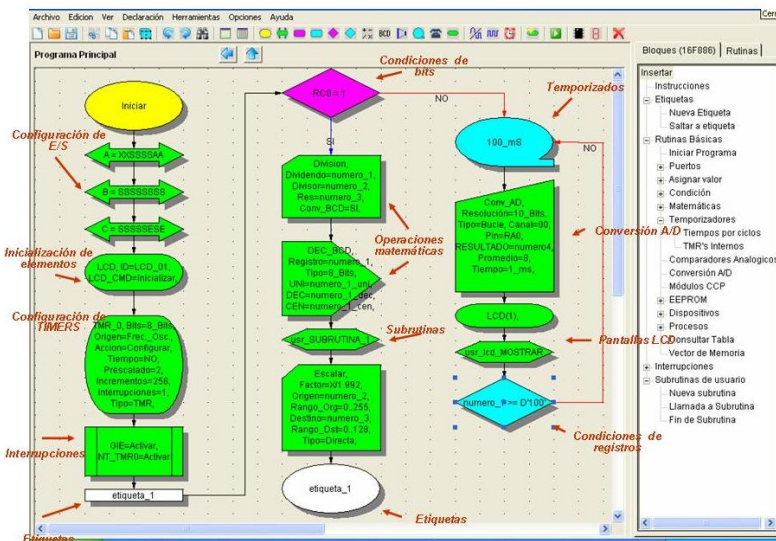


Actualmente en el mercado hay dos programas orientados a la programación visual con microcontroladores PIC: *Niple* y *Flow Code*. Por sencillez y economía se va a emplear el primero en su versión 6.2. Hay que tener en cuenta que, aunque se programa con diagramas de bloques, es necesario tener conocimientos de la estructura (registros de trabajo) y recursos (tipos de E/S, temporizadores e interrupciones) del microcontrolador.

Niple (Cano, 2009; Corrales, 2009), incluye algunos transductores y actuadores integrados como subrutinas (similar a la forma de trabajo con *Arduino*), pero por motivos didácticos, se ha preferido programarlos basándonos exclusivamente en su modo de funcionamiento. Los bloques más utilizados (Figura 2) son los correspondientes a las operaciones matemáticas, temporizados, timers e interrupciones. Precisamente el punto débil de *Niple* es el co-

respondiente a las operaciones matemáticas, ya que no permite programar en coma flotante, reduciéndose al trabajo con números de 1, 2, 3 y 4 Bytes.

Figura 2 Software de programación Niple.



Transductores

Como transductores, se ha elegido dos tipos característicos: el sensor ultrasónico de distancias *SRF04* y el colorímetro *TCS3200*.

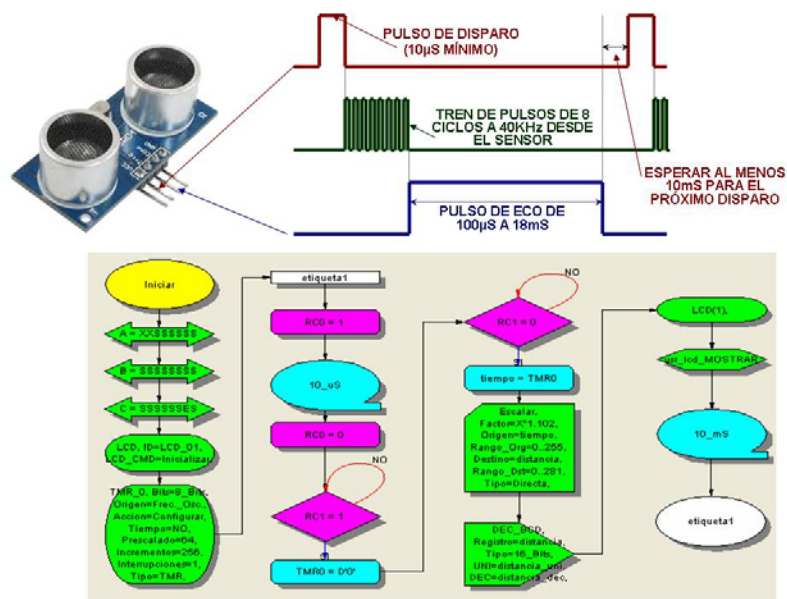
Medidor de distancias *SRF04*

Este transductor es un medidor ultrasónico de distancias cuya principal característica es su bajo coste y facilidad de funcionamiento, el rango de medida es aproximadamente de 2 a 400cm y dispone de un pin de disparo (*TRIG*) y uno de eco (*ECHO*). Al aplicar un pulso TTL de disparo en el pin *TRIG* (pin *RC0* en el microcontrolador *16F886*) de 10µs de duración, la cápsula transmisora del módulo (altavoz) transmite un tren de pulsos de 8 ciclos a 40KHz. En ese momento, el pin *ECHO* (pin *RC1*) pasa a nivel "1". Cuando la cápsula receptora del módulo (micrófono) recibe la señal que ha transmitido como consecuencia de haber rebotado en un objeto, éste pin pasa a nivel "0". Midiendo el tiempo en que la señal *ECHO* se mantiene a "1", se puede obtener la distancia entre el módulo y el objeto sin más que escalar ese tiempo. Para ello, se tiene en cuenta que la velocidad del sonido en el aire es de 343 m/s a 20°C y que el pulso del eco varía desde 100µs (para una distancia de 1,7cm según los cálculos correspondientes) hasta 18 ms (para 308,7cm) (González, 2009).

En la Figura 3 se muestra el desarrollo de la programación. La salida *TRIG* se conecta al pin *RC0* (configurado como entrada en el microcontrolador) y la entrada *ECHO* a *RC1* (salida). En primer lugar, dado que se va a medir un tiempo, hay que configurar el *TIMER* del microcontrolador. Se puede utilizar el *TMR0* (8 bits) o el *TMR1* (16 bits), dependiendo de la resolución con que se quiera trabajar. Por simplicidad, se va a emplear el *TMR0*, de modo que se divide la frecuencia (1MHz) por 64 y se obtiene un tiempo máximo de medida de 16384µs (equivalente a 281cm).

A continuación, se genera la señal de disparo en *RC0* (se pone a “1” y se espera 10µs para bajarla a “0”). Luego se espera que la señal de eco llegue (*RC1*=1) y en ese momento, se pone a cero el *TMR0* para empezar a contar de cero. Cuando *RC1*=0, se “captura” el valor del *TMR0* en un registro de un byte denominado *tiempo* y se escala para obtener la distancia (en centímetros) en un registro denominado *distancia* que será de 2 bytes, para finalmente pasarlo a BCD con objeto de poder visualizar la distancia en una pantalla LCD. Para poder volver a realizar otra medida, habrá que esperar al menos 10ms.

Figura 3 Transductor SRF04 de distancias y programación.



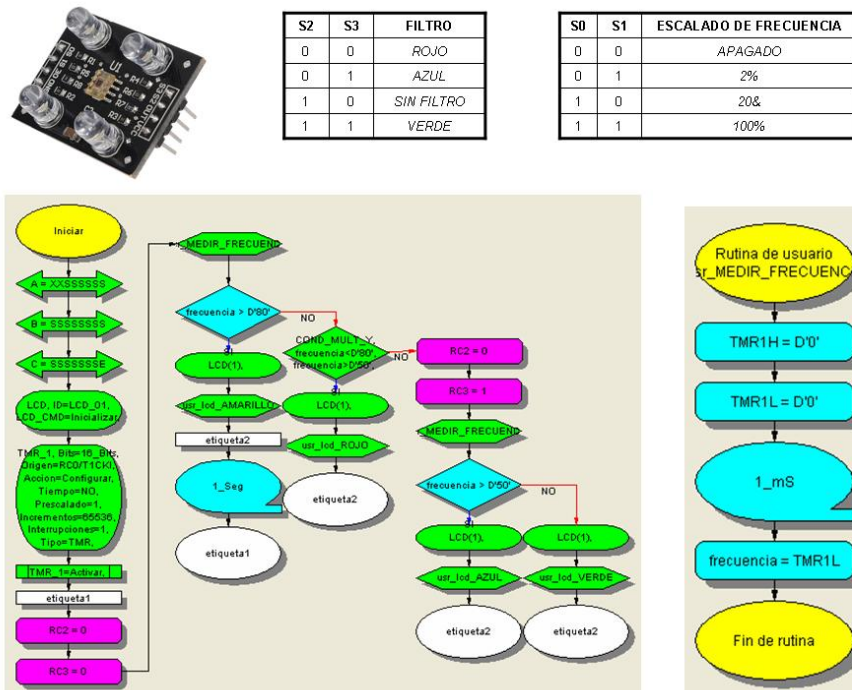
Colorímetro TCS3200

El módulo integra cuatro diodos LED blancos con el sensor de color *TCS3200* (Texas Advanced Optoelectronic Solutions) que detecta la intensidad de luz reflejada, convirtiéndola

en una onda cuadrada de frecuencia variable. Como cada color refleja una intensidad distinta, su identificación es posible asociarla a la medida de frecuencias. Hay que tener en cuenta que el transductor es de bajo coste y no tiene gran precisión pero para detectar los colores amarillo, rojo, verde y azul es más que suficiente. Tiene una salida *OUT* para la señal cuadrada y dos pines, denominados *S2* y *S3*, que permiten seleccionar el filtro con el que se toma la muestra de la frecuencia y así se obtiene el valor de cada uno de los filtros que componen el sistema RGB.

Para los colores deseados, los valores de frecuencia que se miden rondan desde los 10KHz hasta los 150KHz, con una precisión de ± 4 KHz por color. Por este motivo y para simplificar los cálculos y agilizar las medidas, se medirá la frecuencia (número de flancos de la señal) cada 1ms (KHz), y el resultado se podrá almacenar en un solo registro de un byte. Sólo faltará medir las frecuencias (con el *TIMER1* a través del pin *RC0*) e identificar el correspondiente color mediante su frecuencia asociada, obtenida por una muestra patrón. Los pines *RC2* y *RC3* se emplean para controlar los filtros a través de *S2* y *S3*, respectivamente. En la Figura 4 se muestra su programación visual.

Figura 4 Transductor *TCS3200* de color y programación.



Actuadores

Servomotor de continua *Futaba S3003*

El servomotor elegido tiene un ángulo de giro de 0° a 180°, una alimentación a 5V, velocidad de 0° a 180° de 0,60s y un par de 3,2kg-cm. El control del ángulo de giro se realiza por modulación del ancho de pulso (PWM), con un período de 20ms y con unos valores de duración del pulso de 0,3 y 2,1ms para 0° y 180° respectivamente. Tomando los valores descritos anteriormente y con objeto de poder emplear sólo registros de 8 bits (t_{alto} , t_{bajo} y $angulo$), se va a trabajar con unidades de tiempo de $\times 10\mu s$. De este modo, se puede llegar a las siguientes ecuaciones que generan la señal de control:

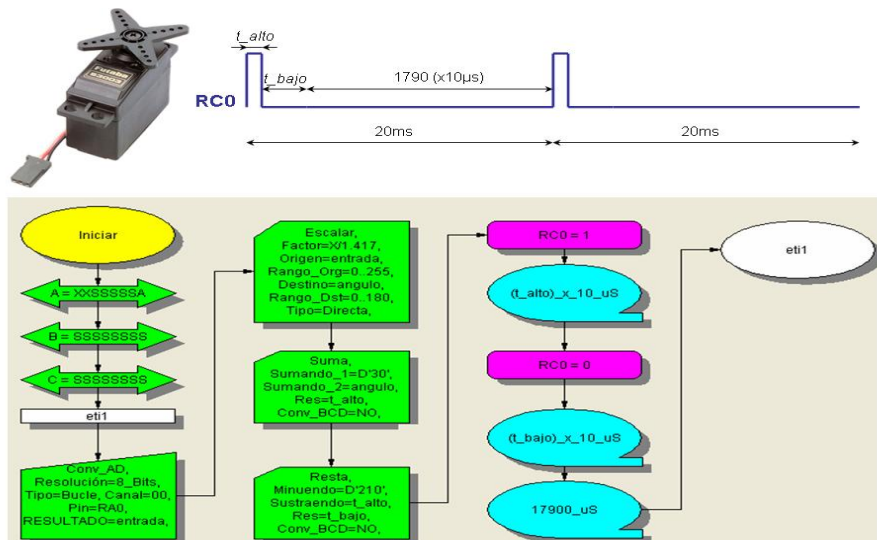
$$t_{alto} (\times 10\mu s) = 30 + angulo$$

$$t_{bajo} (\times 10\mu s) = 210 - t_{alto}$$

donde $2000 (\times 10\mu s) = t_{alto} + t_{bajo} + 1790$

La introducción del ángulo del servomotor se realiza mediante un potenciómetro conectado al pin del microcontrolador *RA0* (configurado como entrada analógica con resolución de 8 bits) y la señal de control se genera a través del pin *RC0*. En la figura 5 se muestra la correspondiente programación visual.

Figura 5 Servomotor S3003 y programación.



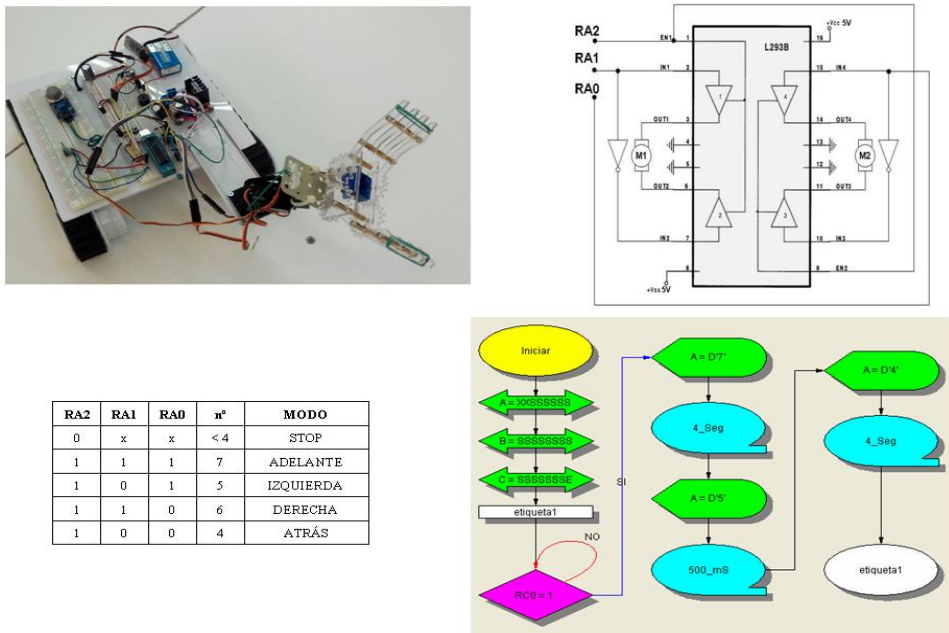
En primer lugar, se lee el valor analógico introducido por *RA0* y se almacena en un registro de 8 bit denominado *entrada*. Seguidamente se escala, desde 0-255 hasta 0-180, en otro registro denominado *angulo* y, a partir de éste, ya se pueden calcular t_{alto} y t_{bajo} . Finalmente, se emplea el bloque temporizador por bucles con tiempo variable, para mantener la señal en *RC0* en alto y bajo el tiempo correspondiente.

Motores de continua

El siguiente montaje se ha realizado para controlar el movimiento de un vehículo empleando dos motores CC (Corrales, 2014). Se basa en el circuito amplificador en intensidad *L293B*, mostrándose en la Figura 6 su disposición. Para el movimiento del vehículo se emplean los pines *RA0*, *RA1* y *RA2* del microcontrolador, mostrándose en la Figura 6 los valores que habría que escribir en el *PORTA* (no empleando los restantes pines de ese puerto) para realizar el movimiento deseado del vehículo.

El programa realizado, hace sencillamente que el vehículo se ponga en marcha si el pin *RC0*=1. En ese caso, marchará 4s hacia delante, luego girará a la izquierda para, finalmente, ir hacia atrás 2s. Si *RC0* sigue en alto, volverá a realizar de nuevo toda la secuencia.

Figura 6 Control y programación de un vehículo.



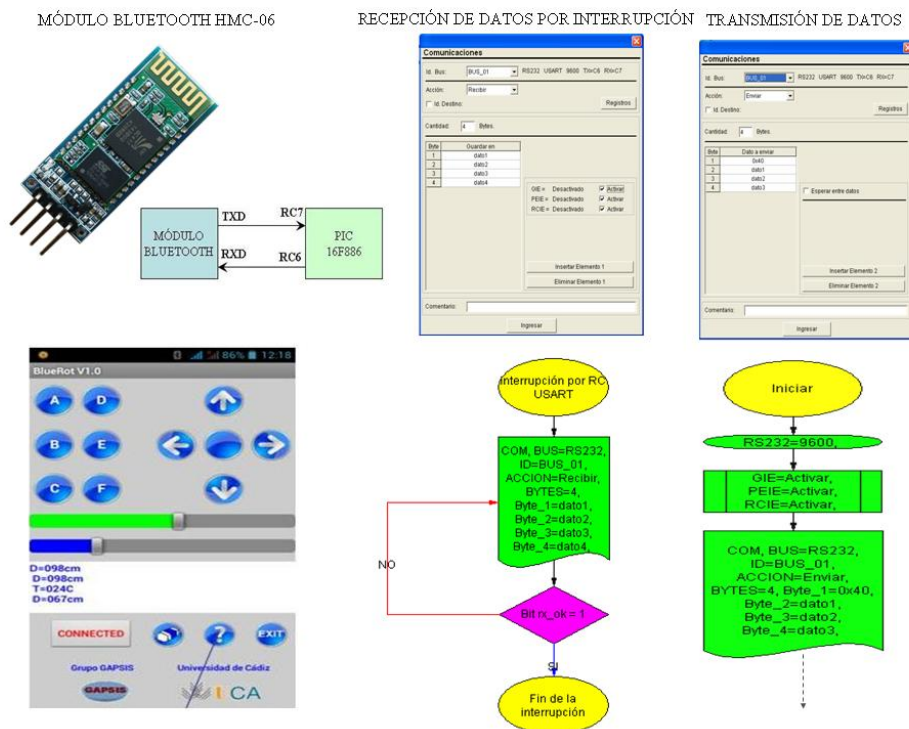
Comunicaciones

Bluetooth

La comunicación entre el microcontrolador y un sistema de control (teléfono móvil o tableta, por ejemplo) se puede realizar vía Bluetooth a través de un módulo esclavo *HC-06*. Las ventajas de este módulo son su economía, sencillez, pequeño tamaño, bajo consumo y amplia bibliografía de soporte. Por sus pines, denominados *TXD* y *RXD*, se envía y recibe datos con el microcontrolador (pines *RC7* y *RC6* respectivamente) en formato serie pero con niveles TTL. De fábrica viene con un nombre de usuario (*linvor*), clave (*1234*), velocidad de transmisión (9600 baudios), bits de datos (8), bit de paridad (no) y bit de parada (1) prefijados, pero se pueden cambiar, simplemente, mediante comandos AT.

La programación con *Niple* es bastante sencilla. Para transmitir datos, sólo hay que decirle cuántos bytes se quieren y el tipo (registro, número o ASCII). La recepción es a través de la correspondiente interrupción asociada, indicando en qué registros se quieren almacenar los datos recibidos (Figura 7). La supervisión y control de los datos se pueden realizar, por ejemplo, mediante la aplicación *BlueRot* desarrollada para *Android* (Corrales, 2014).

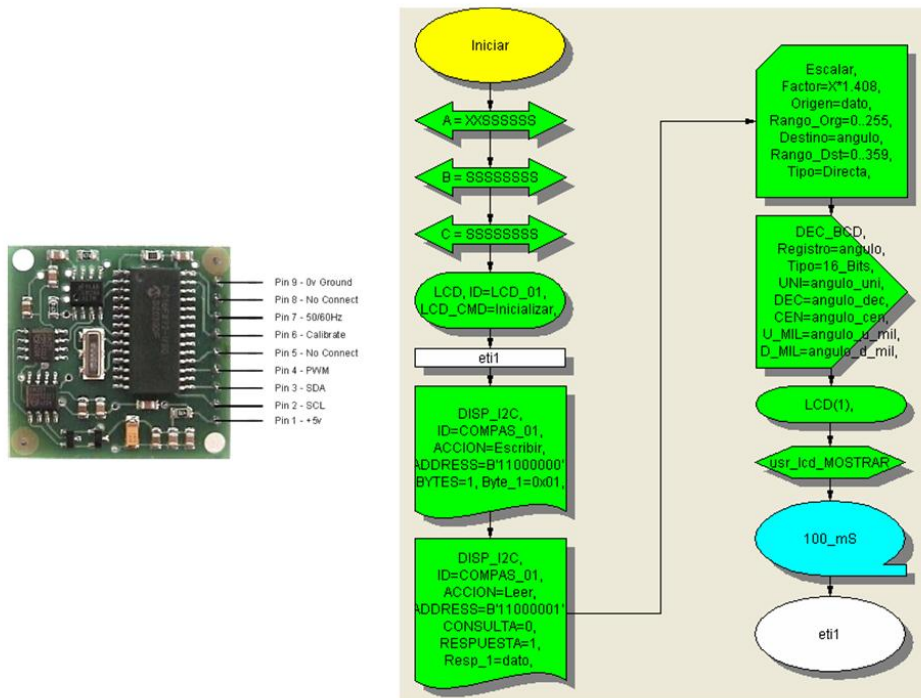
Figura 7 Comunicación Bluetooth y programación.



Bus I2C

Este bus, desarrollado por *Philips*, se emplea principalmente para el intercambio de información, tipo serie a través de dos líneas denominadas *SDA* (datos) y *SCL* (reloj), entre dispositivos periféricos (sensores, memorias, etc.) y microcontroladores. Cada dispositivo periférico tiene una dirección asignada por el fabricante (es posible cambiarla en algunos casos) y, de forma muy resumida, su funcionamiento es el siguiente: el microcontrolador se comunica con el dispositivo, a través de su dirección asignada, y le envía el registro interno del dispositivo que quiere obtener, y luego le pide que le envíe ese dato a través de la dirección del dispositivo incrementada en una unidad. Se ha aplicado a un compás magnético digital *CMPS03*, que tiene asignada la dirección *0xC0*, obteniéndose el valor del ángulo respecto del norte magnético, a través de la posición *0x01* de sus registros internos. Para ello, hay que leer en la dirección *0xC1*. Finalmente, habrá que escalar el valor del registro de 8 bit hasta 359° , para obtener el valor del ángulo con una resolución de $\pm 1,4^\circ$.

Figura 8 Bus I2C y programación.

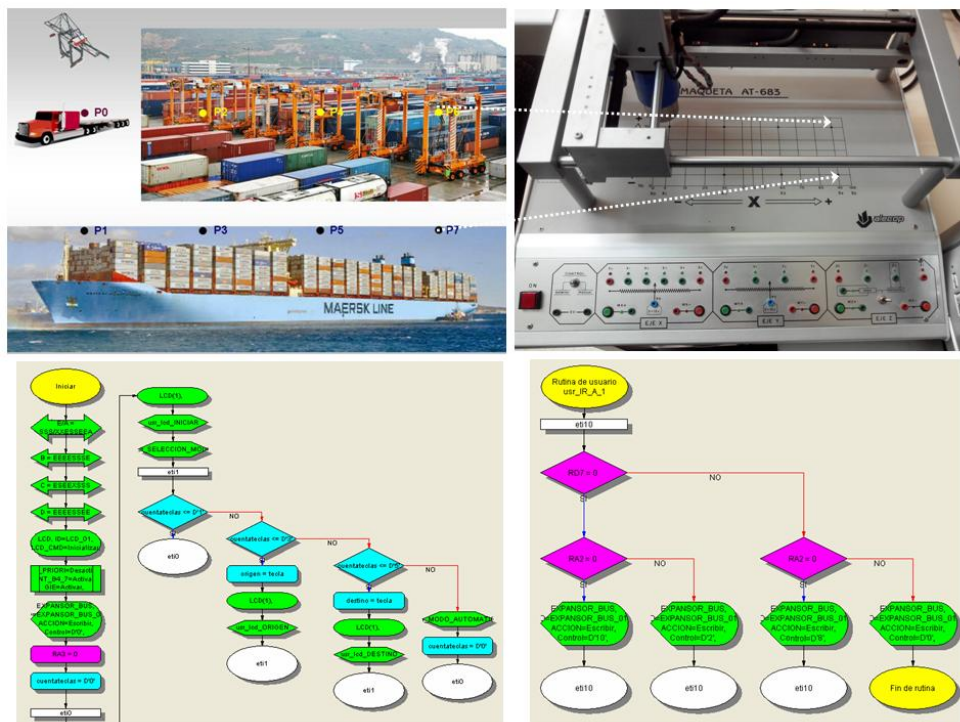


Control de procesos

Se toma como punto de partida una maqueta puente-grúa AT-683 realizada por ALECOP. Esta maqueta tiene la posibilidad de controlar el movimiento de sus ejes de forma digital o manual, pudiéndose emplear, por tanto, un PLC, PC o microcontrolador para realizar su automatización. Con esta maqueta se pueden simular distintos procesos, como por ejemplo, la estiba de contenedores (Figura 9).

Dispone de varios finales de carrera para posicionar el electroimán en 8 posiciones determinadas, así como dos potenciómetros para elegir cualquier posición de la misma. Dado el elevado número de entradas y salidas necesarias para realizar la automatización del proyecto (maqueta, pantalla LCD y teclado matricial), se ha elegido el microcontrolador 18F4550 de Microchip, así como un módulo expansor PCF8574 con otras 8 salidas, empleadas para el control de los motores.

Figura 9 Control de procesos.



Mediante un teclado matricial se introduce la posición inicial donde coger la pieza (contenedor) y la final donde depositarla. La programación se reduce al empleo de subrutinas que llevan la grúa a las distintas posiciones leyendo el estado de los finales de carrera y potenciómetros.

Conclusiones

En este trabajo se han desarrollado varias aplicaciones de microcontroladores para conseguir el objetivo de aumentar el grado de auto-implicación del alumno, su creatividad, y facilitar la motivación y experiencia, haciendo que encuentre un sentido más a la práctica ya que aprende haciendo y descubriendo. Dada la orientación del Grado en Ingeniería Radioelectrónica, 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 ni de lenguaje de alto nivel, como el C, para realizar multitud de casos prácticos. Finalmente indicar que 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 y para los laboratorios de la Universidad.

Referencias

Cano J., "Niple Software, Programación visual de microcontroladores", *Primer Congreso virtual de microcontroladores y sus aplicaciones*, Universidad Tecnológica Nacional, Facultad Regional Paraná, Argentina, (2009).

Corrales C., "*BlueRo!*", Google Play, (2014).

Corrales C., Terrón J., López M.J., "Diseño de un autopiloto naval para formación y aprendizaje integrado", *Actas del Primer Congreso virtual de microcontroladores y sus aplicaciones*, Universidad Tecnológica Nacional, Facultad Regional Paraná, Argentina, (2009).

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).

Corrales C., López M. J., Terrón J., Consigliere A., "Sistema integrado para aprendizaje de microcontroladores", *Actas del XVII Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas*, Almadén (Ciudad Real), España, (2014).

Carlos Corrales Alba, Manuel Jesús López Sánchez y Agustín Consegliere Castilla

González M., Corrales C., "Medida de distancias empleando un medidor ultrasónico con indicación numérica, luminosa y sonora de una distancia fijada", *Actas del Primer Congreso virtual de micro-controladores y sus aplicaciones*, Universidad Tecnológica Nacional, Facultad Regional Paraná, Argentina, (2009).

Terrón J., Corrales C., López M. J., Romero E., Moreno F., Prian 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).

USB-PIC´School Guía de Usuario, *Ingeniería de Microsistemas Programados S.L.*, Bilbao, España, (2010).