

Desde CoOs hasta .NetFramework: Comunicación hardware/software mediante VCP

Juan Pedro Domínguez-Morales, Alberto Vázquez-Baeza, Elena Cerezuela-Escudero, Manuel Domínguez-Morales, Ángel Jiménez-Fernández, Antón Civit-Balcells

Dpto. Arquitectura y Tecnología de Computadores
Universidad de Sevilla
Sevilla

{jpdominguez, avazquez, ecerezuela, mdominguez, ajimenez, civit}@atc.us.es

Resumen

En este artículo se describe una práctica de laboratorio para la docencia de sistemas operativos en tiempo real (RTOS) para microcontroladores de 32bits en el ámbito de los grados en Ingeniería de Computadores, mediante una metodología de aprendizaje basado en proyectos (ABP) [1]. En concreto esta práctica abarca la comunicación Hardware/Software entre un RTOS en un microcontrolador con un núcleo vectorial Cortex-M4 y una aplicación software escrita en C# usando Visual Studio 2010 a través de puertos series virtuales (VCP).

Esta práctica está enfocada como un proyecto que los alumnos han de ir realizando desde cero, avanzando mediante la consecución de hitos, hasta conseguir obtener un sistema final. El sistema a desarrollar se divide en dos partes, por un lado tenemos un microcontrolador de la familia STM32 ejecutando un RTOS completamente libre, CoOs, el cual recibe comandos para mover un motor y un servo, y simultáneamente envía la información de sensores de distancia y contacto usando el puerto serie. Por otro lado tenemos un PC con un sistema operativo de la familia Windows, en el que se construye una aplicación visual, la cual muestra el valor de los sensores y permite enviar comandos de movimiento y posición al microcontrolador.

El desarrollo de este tipo de proyectos se añade a la necesidad del uso de diferentes herramientas para el desarrollo del firmware y del software en paralelo, de manera incremental, y enfocadas para ámbitos de uso muy distintos.

Esta práctica ha tenido una gran acogida por parte de los alumnos, ya que les ha servido de ejemplo del desarrollo de periféricos empotrados y tiempo real, con una aplicación directa en el campo de la robótica, instrumentación o control automático.

Abstract

In this article a practical laboratory session for teaching operating systems in real time (RTOS) with 32-bit microcontrollers is described. This session is

thought to be imparted in the Computer Engineering degree, through a methodology of project-based learning (PBL) [1]. Specifically, this practice session encompasses Hardware/Software RTOS communication between a microcontroller with Cortex-M4 core vector and a software application written in C# using Visual Studio 2010 via virtual serial ports (VCP). This practice is focused as a project that students must be making from scratch, by achieving several milestones, to obtain a final system. The system to develop is divided into two parts. First, we have a STM32 microcontroller family running a completely free RTOS, CoOS, which receives commands to move motors, and simultaneously sends information about distance sensors and using the serial port. On the other hand we have a PC with a Windows system operating family, where a visual application, which shows the value of the sensors and allows to send commands to the microcontroller motion and position is constructed.

The development of this type of project is added to the need to use different development tools for firmware and software in parallel, incrementally, and focused to very different fields of use.

This practice has been well received by the students, because it has served as an example of the development of a real-time system with embedded peripherals; and it has a direct application in the field of robotics, instrumentation and automatic control.

Palabras clave

Sistema operativo en tiempo real, microcontrolador STM32, CoOS, servomotor, C#, Windows Form.

1. Motivación

La asignatura de Sistemas Empotrados y de Tiempo Real se imparte dentro del plan de estudios del Grado de Ingeniería Informática, especialidad en Ingeniería de Computadores, en la Escuela Técnica Superior en Ingeniería Informática de la Universidad

de Sevilla. Esta asignatura posee, como principal objetivo, introducir al alumnado en el estudio, programación y desarrollo de sistemas basados en microcontrolador [6, 7, 8] sin y con sistema operativo incorporado. Así pues, se realiza un estudio progresivo de los microcontroladores y una serie de sensores y actuadores sobre los que trabajar con el microcontrolador.

Partiendo de un microcontrolador tipo i8051 hasta terminar en un microcontrolador de la familia STMicroelectronics de 32 bits (a partir de ahora STM32) se estudia la configuración interna y la comunicación con una serie de periféricos. En cuanto a los periféricos se abarca una amplia gama de sensores y actuadores: desde simples sensores de temperatura hasta sensores de movimiento y aceleración; igualmente con actuadores que varían desde servomotores a motores de corriente continua. De igual forma se introduce en la asignatura en la comunicación inalámbrica de ámbito industrial, como son las redes inalámbricas que implementan el estándar 802.15.4 (Zigbee) [10], para su uso con un microcontrolador.

Aunque esta asignatura suscita un alto grado de motivación por parte del alumnado, se ha comprobado que por término general se suele asociar el uso de un microcontrolador a un sistema autónomo con poca o nula intervención por parte del usuario, quedando ilusoriamente distanciado de un sistema de propósito general o de algoritmos de procesamiento de alto nivel.

Es por ello que, para intentar evitar esta visión secundaria de los diferentes tipos de computadores, se ha pretendido realizar una práctica de laboratorio que realice una integración de un sistema basado en microcontrolador dentro de una aplicación de usuario de alto nivel, programada en un ordenador de propósito general. Esta práctica intenta solventar las dudas en la comunicación entre estos dos sistemas, aparentemente opuestos, pero que pueden llegar a complementarse para obtener un procesamiento o aplicaciones de mucho más alto nivel de las que el alumno cabría esperar.

Así pues, en este trabajo se presenta la estructura, motivación y desarrollo de una práctica de laboratorio, dividida en dos sesiones de dos horas cada una, que presenta un sistema de comunicación inalámbrica bluetooth entre una PCB con microcontrolador STM32 y un computador de propósito general corriendo una aplicación en C# con Windows Forms.

A continuación se detallan los objetivos marcados para el alumnado que asista a estas sesiones prácticas.

2. Objetivos

Dentro del ámbito en el que se encuentra focalizada esta sesión práctica y teniendo en cuenta que el conjunto del alumnado posee un perfil de Ingeniería

de Computadores, cien por cien relacionada con el aspecto hardware de la informática, se fijaron los objetivos indicados a continuación.

Por primera vez en su carrera, se hace uso de una comunicación serie asíncrona desde un ordenador de propósito general. Para ello se hace uso del lenguaje C#, orientado a objetos, desde el entorno de programación Visual Studio.

Asimismo, completando la parte más software del proyecto, se realiza una aplicación de usuario sobre Windows Forms, que integra componentes gráficos. Esta aplicación deberá tener en cuenta dos aspectos contrapuestos: por un lado deberá ser capaz de mostrar la información recibida por parte de los sensores situados en la placa del microcontrolador y, por el otro, deberá permitir el envío de comandos por el puerto serie. Es crítica una buena organización del programa, ya que el puerto serie actúa de cuello de botella para ambos propósitos y no cabe la posibilidad de utilizarlo de forma bidireccional.

En cuanto al apartado más relacionado con la parte hardware, es necesario indicar que el alumnado que asista a esta sesión práctica ya posee conocimientos acerca de la programación del microcontrolador en cuestión y el uso del entorno correspondiente. Sin embargo, hay determinados aspectos novedosos con los que trabajarán en esta sesión práctica.

Entre los aspectos novedosos, cabe destacar el control de servomotores y motores de corriente continua con el microcontrolador, ya que los conocimientos previos sobre la modulación PWM (modulación por ancho de pulso) han sido recibidos a estas alturas en sesiones previas. Por otro lado se integran en el microcontrolador una serie de sensores y actuadores adicionales tales como sensor de aceleración, de temperatura, de movimiento (joystick) y componentes de bajo perfil tales como diodos led, display, etc.

Todo ello, teniendo en cuenta que la comunicación por bluetooth debe ser implementada a ambos lados. Para ello el alumno aprenderá la manera implementar puertos series virtuales, tanto desde una aplicación de usuario en un ordenador de propósito general como desde un computador empujado.

Para evitar las colisiones que pudieran ocasionarse entre paquetes enviados y recibidos, el alumno aprenderá conceptos muy utilizados en sistemas operativos para hacer uso de recursos compartidos, como son los semáforos.

En la Figura 1 puede apreciarse un diagrama de las partes en las que se divide la sesión práctica:

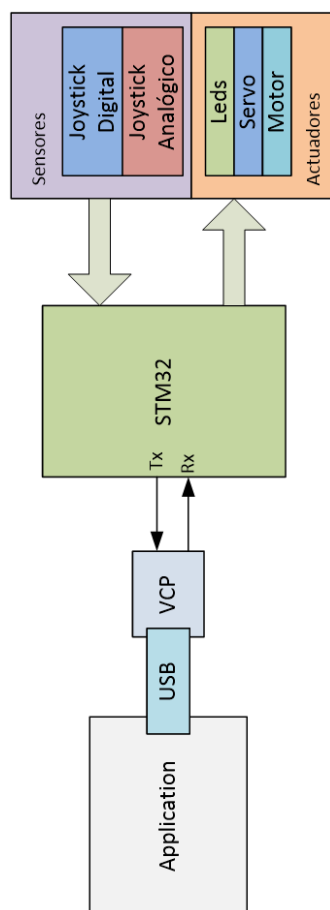


Figura 1. Visión a nivel de sistema

3. Metodología

En cuanto a la metodología utilizada en esta sesión práctica, se hace hincapié en una serie de aspectos importantes que se citarán a continuación.

Por un lado, esta práctica está pensada para realizarla en dos sesiones de dos horas de duración cada una, lo cual permite una mayor flexibilidad en cuanto a los conocimientos requeridos y complejidad de la práctica; permitiendo al alumno poder trabajar con ella en el periodo comprendido entre ambas sesiones.

Se sigue una metodología basada en problemas (ABP) para la satisfacción de los requisitos exigidos al alumno al final de la práctica completa. Esta metodología permite apoyarse en la consecución de diversos hitos, o pequeños proyectos, para resolver un problema mayor a posteriori y ha sido probada en numerosos trabajos [2,3,4,5].

Estos pequeños hitos comienzan con el desarrollo de una pequeña aplicación de usuario de muestreo de información, pasando por el establecimiento de una comunicación inalámbrica, y pasando con la incorporación al sistema completo. Es importante indicar que cada hito está claramente diferenciado del anterior y requiere de un test final para comprobar el funcionamiento de dicho módulo. Uno a uno, estos módulos

se van desarrollando y testeando para, finalmente, realizar la labor de integración en el proyecto completo. El uso de este tipo de metodología va más allá de la sesión práctica en sí; la asignatura anteriormente citada posee otra serie de sesiones prácticas que preceden a la expuesta en este trabajo, de forma que algunos hitos necesarios en esta práctica ya han sido completados en sesiones previas y, más aun, habrá algunos hitos de sesiones posteriores que se verán beneficiados del trabajo del alumno en esta sesión.

Esta metodología supone el aprovechamiento máximo del tiempo del alumno en cada una de las sesiones, aumentando la motivación del alumnado, que se ve compensado en su calificación por el trabajo realizado a lo largo del curso.

Para permitir a los más rezagados llegar a completar todos los hitos de la sesión, se ha habilitado un laboratorio especial en el departamento que posee control de acceso mediante tarjeta. A este laboratorio se permite acceder a alumnos de trabajo fin de carrera, trabajo fin de grado, trabajo fin de máster y, en el caso que aquí atañe, a alumnos de asignaturas concretas que requieran del uso de material de laboratorio fuera de horario de clase [9,11]. Además, el laboratorio posee una serie de armarios, también controlados por tarjeta, en los que se puede dejar el material para su uso por parte del alumnado. De esta forma se consiguen solventar dos problemas: por un lado se permite a los alumnos terminar aquellos hitos de las sesiones prácticas que no hayan sido capaces de completar y, por otro lado, se mantiene un control del material de laboratorio para evitar el mal uso del mismo y el extravío.

En cuanto a la evaluación utilizada en esta sesión (y en las restantes) para la obtención de la calificación práctica del alumnado, viene detallada por la asistencia y aprovechamiento en clase, la completitud de los hitos marcados y, por último, mediante la entrega de una memoria final de prácticas.

Seguidamente se detallará una descripción del material utilizado.

4. Descripción del material utilizado

A continuación se describe el material utilizado en la realización de la presente sesión práctica, desde material software hasta hardware.

- Placa de desarrollo de STM32 [8]: ARM Cortex-M4 32-bit RISC, a 168MHz, con 512kB de memoria flash y 192kB de SRAM. El kit de desarrollo viene acompañado de multitud de sensores y displays. Mostrada en la Figura 2.



Figura 2. Placa de evaluación para ARM32.

- Sensores de aceleración, temperatura y movimiento, como el que por ejemplo encontramos en la Figura 3.



Figura 3. Sensor de movimiento analógico (joystick).

- Módulos bluetooth para comunicación entre el ordenador de sobremesa y el microcontrolador STM32.
- Entorno de desarrollo Visual Studio 2010 con el paquete de programación en C#.

5. Desarrollo de la sesión práctica

Se irá detallando paso a paso los hitos propuestos al alumno, así como una resolución parcial de los mismos. Para ello se hará uso de capturas y esquemas para comprender mejor los objetivos a cumplir.

5.1. Obtención de datos de sensores

Modificar el código para realizar una lectura cada 100 milisegundos de los valores de los sensores, almacenando el contenido a variables para poder ser mandados a posteriori.

Inicialmente para este hito se solicita obtener los valores y comprobar, mediante depuración, que se están realizando correctamente las lecturas: hay que comprender que algunos sensores requieren de una transformación de la información analógica recibida antes de poder interpretarlos; y este es el punto de mayor complejidad que se encuentra el alumno en la práctica, ya que deberá manejarse con los datasheet de los diversos sensores e interpretarlos.

5.2. Comunicación serie desde el microcontrolador.

El alumno hace uso de mecanismos de empaquetado de la información, así como implementación de

tareas periódicas del sistema operativo del microcontrolador para transmitir cadenas de texto por el puerto serie. Para ello se utilizarán dos tareas, una de transmisión y otra de recepción, ambas dependientes de la interrupción del puerto serie, tal y como vemos en la Figura 4.

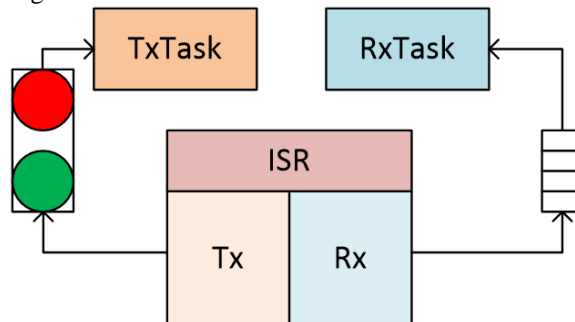


Figura 4. Esquema de comunicación serie

Off-set	Descripción	Dato
+0	Inicio de trama	0x7E
+1	Longitud de la trama	0x05
+2	Dato [0]	0x01
+3	Dato [1]	0x55
+4	Dato [2]	0x00
+5	Dato [3]	0xAA
+6	Dato [4]	0x04
+8	Fin de trama	0x0D

Figura 5. Empaquetado de la información

Se modifica la tarea de comunicación serie periódica para que realice el empaquetado de la información y la mande acorde con la estructura de datos de la Figura 5, y siguiendo el esquema proporcionado en la Figura 6.

Para el testeo de este hito se hace uso de un terminal virtual para un ordenador de propósito general denominado X-CTU, que permite visualizar los paquetes mandados y servirnos meramente de chivato de la comunicación.

5.3. Creación de una aplicación gráfica en Visual Studio.

En primer lugar es importante destacar que, a la altura de los alumnos que cursan esta asignatura, aún no han tenido la oportunidad de necesitar aplicaciones gráficas en las precedentes asignaturas. Es por ello que, para muchos de los alumnos, esta es la primera toma de contacto que tienen con un entorno de programación para aplicaciones gráficas.

Por ello, se hace especial hincapié en la creación y configuración del proyecto, siendo un primer hito la programación de determinadas herramientas dentro de la aplicación gráfica, tales como botones, cuadros de texto, y manejo de eventos de ratón y similares.

```

void serialTxTask(void * parg){
    int i;
    for(;;){
        //Actualizar estado
        x = getAnalogJoy(0);
        //...

        SerialSendByte(0x7E);
        SerialSendByte(5);
        //SerialSendByte(...)
        //SerialSendByte(...)
        // ..
        //SerialSendByte(..)
        SerialSendByte(0x0D);

        CoTimeDelay(0,0,0,100);
    }
}
    
```

Figura 6. Tarea de envío de información vía serie

5.4. Comunicación serie desde la aplicación de Visual Studio.

En este hito se agrega un puerto serie virtual a la aplicación gráfica comenzada en el hito anterior. A este puerto serie creado se le asocia un evento de envío de información, que se realiza periódicamente a una determinada frecuencia y velocidad (en baudios).

El alumno debe ser capaz de inicializar el puerto serie y de mandar paquetes de información, correspondiente a la introducida en un cuadro de texto de la aplicación gráfica.

El testeo del funcionamiento de este hito se realiza mediante un microcontrolador STM32 con un display LCD y un puerto serie conectado al ordenador. Una vez se compruebe la correcta recepción de la información transmitida desde el ordenador se puede dar por concluido de forma satisfactoria el presente hito.

Incluido dentro de este apartado, en la comunicación serie, se da paso a la recepción de información por el puerto serie. Para ello no basta una simple tarea repetitiva o una función activada mediante algún evento concreto, sino que es necesaria la implementación de una máquina de estados para llevar a cabo correctamente la comunicación.

Para testear el funcionamiento de esta máquina de estado, se hará uso del hito implementado en primera instancia acerca de la recogida de información de los determinados sensores en la placa del STM32. LA única diferencia radicará en que, en este punto, la información será remitida a la aplicación del alumno, muestreando los resultados en un cuadro de texto.

Cabe destacar que la información recibida debe pasar un proceso de decodificación, en el que se realizará el proceso inverso al utilizado a la hora de empaquetar la información en el STM32.

5.5. Transmisión de comandos al STM32 y actuación.

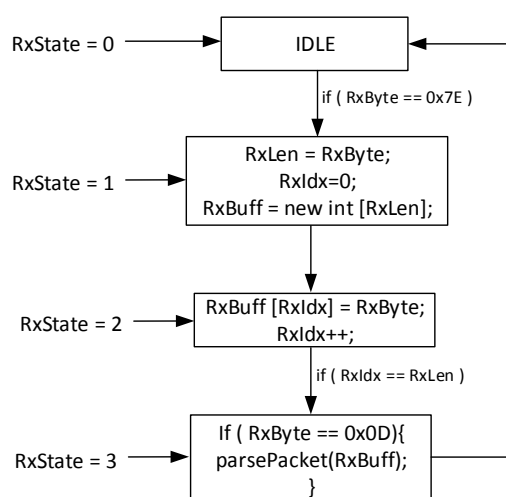


Figura 7. Máquina de estados para recepción serie.

Por último únicamente quedaría implementar una máquina de estados en el código del STM32 para recibir los datos proporcionados por la aplicación de Visual Studio y actuar sobre los servomotores. Dicha máquina de estados se implementará según la Figura 7.

Para simplificar la comunicación se implementa una codificación basada en caracteres para diferenciar los tipos de comandos a mandar. De esa forma puede diferenciarse entre el comando para encender los leds y el mandado para mover un determinado servomotor. Para el caso de los leds, se le aporta a continuación un byte de información diferenciando, bit a bit, el estado de cada uno de los 8 diodos led disponibles en la placa.

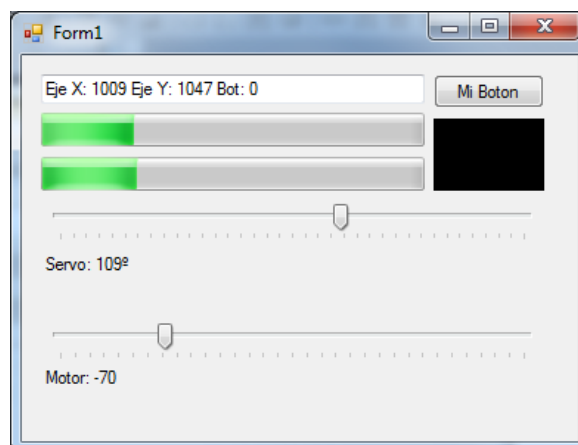


Figura 8. Aplicación de usuario final.

Para los servomotores, se hace uso de dos bytes: el primero especifica de forma numérica el servomotor sobre el que actuar (pudiendo diferenciar hasta 256

motores diferentes, aunque únicamente se haga uso de dos), y el segundo byte especifica el valor de la señal de PWM utilizada para actuar sobre el servomotor (previamente se hace una traducción del valor en grados del servomotor al valor del PWM con resolución de 8 bits).

Tras este hito, se comprueba el funcionamiento global del proyecto, para ello se hace uso de la aplicación desarrollada en Visual Studio, cuyo resultado vemos en la Figura 8.

Una vez indicado, hito a hito, el desarrollo de la sesión práctica, se pasará a comentar los resultados obtenidos tras la impartición de esta práctica.

6. Resultados obtenidos

En primer lugar, cabe destacar que se expondrán los resultados en dos aspectos diferenciados: por un lado se mostrarán los resultados académicos relacionados con el presente trabajo y se detallará la influencia que ha tenido esta metodología. Por otro lado, se mostrarán los resultados de una encuesta anónima realizada al alumnado antes de la puesta de calificaciones, en la que se les preguntaba por diversos aspectos relacionados con las prácticas.

En el aspecto que atañe a los resultados académicos cabe destacar que, de entre los 65 matriculados del presente curso académico, 60 de ellos han seguido al día la asignatura, habiendo un pequeño porcentaje que abandonó la asignatura sin terminar de cursarla.

De los alumnos indicados, la totalidad de los mismos ha asistido a más del 90% de prácticas. Y, de todos ellos, todos los alumnos excepto uno han aprobado la asignatura.

Por otro lado, con respecto a la encuesta anónima, se entregó un cuestionario antes de publicar las notas, en el que se solicitaba la opinión del alumno acerca de la práctica que aquí se presenta, siendo la presente práctica la única novedad con respecto al pasado curso académico.

Todos los alumnos tuvieron comentarios positivos con respecto a la práctica. Este hecho se ha traducido en 3 alumnos, de entre los que cursaron esta práctica, que se encuentran en la actualidad realizando el trabajo fin de grado dentro del departamento relacionado con la comunicación bluetooth entre PC y microcontrolador, partiendo de lo aprendido en esta práctica.

Este hecho motiva en gran medida la labor docente de los profesores implicados en esta asignatura, por ello se encuentran satisfactorios de los resultados obtenidos.

7. Conclusiones

En el presente trabajo se ha expuesto una metodología de aprendizaje basada en problemas, que parte

de un objetivo final a completarse en una serie de hitos propuestos al alumnado.

Estos hitos se estipulan en un principio y se dividen en dos sesiones prácticas de dos horas de duración cada una.

Además de esta metodología, se permite el uso de material por parte del alumnado en aulas especializadas y con control de acceso, fomentando el trabajo fuera de clase y motivando al alumnado.

Debido a la falta de conocimiento del alumnado en el ámbito de las aplicaciones gráficas y en la integración de un sistema empotrado con un pc de sobremesa, esta práctica ha sido el punto de partida de numerosos proyectos de alumnos internos y trabajos fin de grado y máster que parten de la base expuesta en el presente trabajo.

Los resultados obtenidos tras la impartición de la asignatura, muestran un alto grado de implicación por parte del alumnado, además de un gran índice de aprobados; a pesar de la complejidad del temario impartido, tanto en la asignatura en sí como en la práctica detallada. Estos hechos prueban la eficiencia de hacer uso de metodologías de aprendizaje progresivo mediante hitos, de forma que el alumno se vea recompensado al final por el trabajo que ha ido realizando a lo largo de una serie de sesiones.

Agradecimientos

Este trabajo ha sido financiado por el proyecto BIOSENSE (TEC2012-37868-C04-02) del Plan Nacional de Investigación del Ministerio de Economía y Competitividad con soporte del Fondo Europeo de Desarrollo Regional (FEDER).

Referencias

- [1] Solomon, G., 2003. Project-Based Learning: a Primer. *Technology and Learning*, 23(6), 20-30.
- [2] Hedley, M., Barrie, S., 1998. An undergraduate microcontroller systems laboratory. *IEEE Transactions on Education*, vol. 41, issue 4.
- [3] Redes inalámbricas Zigbee (802.15.4): <http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf>
- [4] Labra, J.E., Fernández, D., Calvo, J., Cernuda, A. 2006. Una Experiencia de aprendizaje basado en proyectos utilizando herramientas colaborativas de desarrollo de software libre. *XII Jornadas de Enseñanza Universitaria de Informática (JENUI)*, Bilbao, 12-14 Junio.
- [5] Markham, T., 2003. Project Based Learning, a guide to Standard-focused Project based learning for middle and high school teachers. Buck Institute for Education,

- [6] Solomon, G., 2003. Project Based Learning: a Primer, Technology a Learning. Vol. 23, num. 6.
- [7] Dolmans, H, et al., 2005. Problem-based learning: future challenges for educational practice and research. Blackwell Publishing Ltd. Medical Education, 39.
- [8] Brown, G. y Atkins, M., 1988. Effective teaching in Higher Education. Ed. Routledge. Londres.
- [9] Fuentes, P. y otros, 1997. Técnicas de trabajo individual y de grupo en el aula. De la teoría a la práctica. Ed. Pirámide.
- [10] Colén, M.T., Giné, N. y Imbernon, F., 2006. La carpeta de aprendizaje del alumnado universitario: la autonomía del estudiante en el proceso de aprendizaje.
- [11] Datasheet del microcontrolador STM32: http://www.st.com/web/en/resource/sales_and_marketing/promotional_material/brochure/brstm32.pdf