

# Experiencia docente en el desarrollo de aplicaciones empotradas con MarteOS

Silvia Terrasa, Patricia Balbastre, Alfons Crespo

Dept. de Informática de Sistemas y Computadores

Universidad Politécnica de Valencia

46022 Valencia

e-mail: {sterrasa, patricia, alfons}@disca.upv.es

## Resumen

Este trabajo presenta la experiencia de realizar trabajos basados en sistemas empotrados, para la asignaturas relacionadas con sistemas de tiempo real. Las herramientas software utilizadas para el desarrollo de los trabajos son el lenguaje de programación Ada'95 y el *microkernel* de tiempo real MarteOS desarrollado en la Universidad de Cantabria.

## 1. Introducción y motivación.

Actualmente, uno de los aspectos que deben cubrir las enseñanzas universitarias es el de proporcionar al alumno escenarios de trabajo lo más cercanos posible a la realidad. La utilización de entornos reales, no obstante, no debe ser precipitada, ya que no hay que olvidar que para que los alumnos puedan *exprimir* al máximo estas experiencias, primero hay que formarlos con una base sólida. Una vez claros los conocimientos básicos, ya se pueden acabar consolidar con ejemplos reales.

Este es el planteamiento de la mayoría de los planes de estudios que hay actualmente en vigor en las universidades españolas, los cuales se dedican los primeros semestres de las titulaciones a impartir asignaturas básicas con fuerte contenido teórico para finalmente dedicar los últimos semestres a asignaturas optativas con un fuerte carácter práctico. En el marco de las asignaturas prácticas es dónde tiene más sentido ofrecer ejemplos o situaciones reales.

Con esta idea en la cabeza, los profesores implicados en las asignaturas de sistemas de tiempo real (las cuales son optativas de último curso), estamos siempre dispuestos a introducir elementos "reales" en la elaboración tanto de las prácticas como de los trabajos de asignatura [1][2][3].

Este artículo presenta la última experiencia que hemos tenido en esa búsqueda de escenarios y ejemplos reales. Se trata básicamente de la incorporación a la asignatura de aspectos de programación de sistemas empotrados. Para poder llevar a cabo esta labor ha sido indispensable la utilización de herramientas de soporte, en concreto hemos utilizado un *microkernel* de tiempo real llamado MarteOS, el cual ha sido desarrollado por el departamento de Electrónica y Computadores de la Universidad de Cantabria [4]. Por otra parte también ha sido necesario utilizar el lenguaje de programación de tiempo real Ada'95. Finalmente y para hacer más atractivos los trabajos, éstos tenían como objetivo el realizar el control procesos físicos reales.

El resto del artículo está esquematizado de la siguiente manera, en la sección 2 se verá con más detalle el marco y los objetivos de las asignaturas implicadas. En la sección 3 se describirán las herramientas utilizadas en el desarrollo de los trabajos para pasar en la sección 4 a describir los trabajos así como su planificación en términos de objetivos a desarrollar por los alumnos. Finalmente en la sección 5 concluiremos y presentaremos una serie de retos futuros.

## 2. Marco y objetivos de STR<sup>1</sup>.

La asignatura objeto del presente artículo es la de Sistemas de tiempo real (STR). Actualmente esta asignatura se imparte en cuatro titulaciones distintas: Ingeniero Industrial, Ingeniero en Automática y Electrónica Industrial, Ingeniero en Informática, Ingeniero Técnico en Informática (en la especialidad de sistemas). En todas ellas se imparte como asignatura optativa de último curso excepto en el caso de Ingeniero en Automática y Electrónica que es una asignatura obligatoria también de último curso.

A pesar de la diversidad de las titulaciones donde se imparte, en todas ellas la podemos enmarcar dentro de lo que se denomina Ingeniería de Control y los objetivos más relevantes de ésta son saber diseñar e implementar sistemas de control. Las asignaturas de control, que se ven en todas las titulaciones, aportan el conocimiento y la habilidad para comprender y analizar los procesos industriales y los métodos y técnicas para diseñar e implementar el sistema de control. Los ingenieros deben especializarse en muchos aspectos como:

- Teoría de Control.
- Modelización.
- Simulación.
- Técnicas avanzadas de control.
- Programación de propósito general.
- Planificación.

A parte de los objetivos generales de las asignaturas que se engloban en la llamada Ingeniería de Control, la asignatura de STR tiene una serie de objetivos concretos que podemos resumir en:

- Entender las restricciones de tiempo y los elementos que intervienen en el desarrollo de aplicaciones controladas por computadora.
- Diseñar e implementar aplicaciones de control por computadora.
- Conocer las restricciones que imponen los sistemas empuotrados.

- Conocer los lenguajes de alto nivel y sus abstracciones para implementar la concurrencia, comunicación y control del tiempo (Ada'95).
- Conocer la influencia del sistema operativo y los algoritmos de planificación.
- Conocer y asimilar las técnicas de análisis de planificabilidad.

Otra característica u objetivo común que tienen esta asignatura en las diferentes titulaciones es el deseo, por parte tanto del profesorado como del alumnado, de estar en contacto con entornos de desarrollo y procesos reales. Tal y como se planteaba en [1] nos resulta especialmente interesante que el alumno pueda enfrentarse con situaciones reales tales como el control de un depósito de agua o de un robot móvil. Por otra parte estamos interesados en ofrecerle al alumno herramientas que le permitan realizar sus propias pruebas [2]. En curso actual hemos decidido introducir un elemento más a la cadena y permitir que desarrollen aplicaciones para ser ejecutadas en un computador empuotrado. Para ello hemos utilizado el *microkernel* de tiempo real llamado MarteOS [4] que en el siguiente apartado pasamos a detallar.

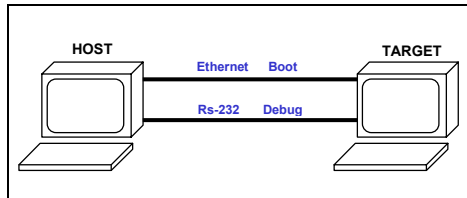
## 3. Entorno de desarrollo de MarteOS.

MarteOS es un *microkernel* de tiempo real desarrollado en el Departamento de Electrónica y Computadores de la Universidad de Cantabria. Este *microkernel* sigue el subconjunto mínimo POSIX de tiempo real (POSIX.13) el cual ofrece una interfaz estándar para aplicaciones escritas tanto en Ada como en C. En este *microkernel* la mayoría del código está escrito en Ada, aunque también hay partes realizadas en C.

MarteOS trabaja en un entorno de desarrollo cruzado formado por un PC con Linux actuando como *host* y una máquina desnuda basada en la arquitectura del 386 que actúa como *target*, como muestra la figura 1 ambos sistemas se conectan mediante una red Ethernet, para realizar el arranque de la máquina, y mediante una línea RS-232 para las tareas de depuración remota.

---

<sup>1</sup> STR son las siglas de Sistemas de Tiempo Real.



**Figura 1.- Entorno de desarrollo cruzado de MarteOS**

El ciclo de desarrollo de las aplicaciones sigue los siguientes pasos:

- Se arranca el PC *target* con el disquete de arranque generado en la instalación de MarteOS.
- La aplicación se compila y se enlaza en el PC *host*.
- En el PC *target* el programa de arranque se “baja” la aplicación a través de la conexión de red y la pone en ejecución.
- La aplicación puede ser ejecutada normalmente o puede ejecutarse en modo depuración remota.
- Tan pronto como la aplicación termina el programa de arranque del PC *target* se queda esperando para “bajarse” y ejecutar la siguiente aplicación.

El desarrollo cruzado da la oportunidad al alumno de implementar sus aplicaciones en un entorno conocido para luego probarlo en el entorno real.

Respecto a otros entornos de programación reales utilizados anteriormente en la realización de los trabajos, como RT-Linux [2], la ventaja que ofrece MarteOS es fundamentalmente la posibilidad de desarrollar todas las aplicaciones en Ada'95, que es el lenguaje que ven los alumnos durante todo el curso.

En el siguiente apartado se describen los trabajos planteados el presente curso.

#### **4. Planteamiento de los trabajos.**

Como se comentaba en [1], para realizar la evaluación de la asignatura de Sistemas de

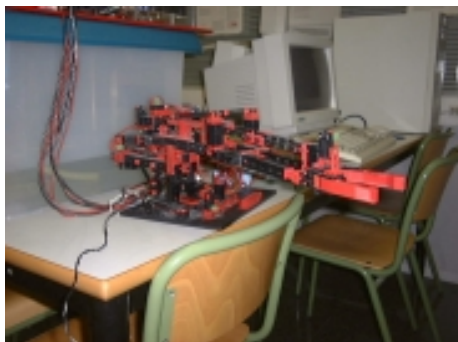
Tiempo Real se le ofrece al alumno la posibilidad de realizar un trabajo de asignatura en vez de realizar un examen escrito al final de la misma. Para que esta opción les resulte viable, se debe proporcionar un horario de laboratorio suficiente para que puedan desarrollar los trabajos durante el tiempo de laboratorio.

Actualmente la asignatura dispone de 6 créditos, 4,5 de teoría (que incluye 1.5 créditos de problemas de aula) y 1.5 créditos de prácticas de laboratorio. Esta distribución de créditos proporciona tres horas de teoría a la semana (repartidas en dos clases de una hora y media) y dos horas de prácticas cada quince días. Obviamente con dos horas cada quince días no se pueden realizar las prácticas y trabajo de asignatura, por tanto, lo que se hace es destinar ese crédito y medio de problemas de aula a sesiones de laboratorio, de manera que puedan tener dos horas a la semana más de laboratorio. Con este cambio y reduciendo al mínimo el número de prácticas (actualmente sólo realizan cuatro) conseguimos tener el tiempo suficiente para que los alumnos realicen los trabajos antes del periodo de exámenes.

Respecto a las tareas a realizar en los trabajos, normalmente se debe realizar el control de un proceso físico, como robots móviles, depósitos de agua, brazos robot, etc. A continuación se detallan un par de los trabajos que se han planteado el presente curso.

##### **4.1. Control de un brazo robot con MarteOS.**

Uno de los trabajos propuestos durante el presente curso fue realizar el control de movimiento de un brazo robot de cuatro grados de libertad (Figura2). El robot dispone de cuatro motores, uno por articulación y también lleva incorporados los finales de carrera. Todos las entradas y las salidas del brazo robot son digitales.



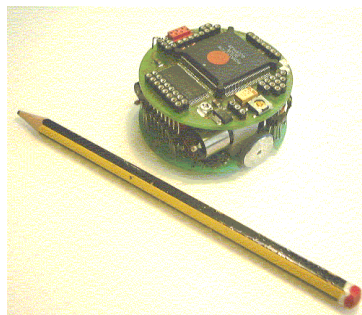
**Figura 2.- Brazo robot.**

Para realizar el control desde el PC se conectaron las salidas y entradas digitales del robot a una tarjeta de adquisición de datos para que fueran accesibles desde el PC.

Debido a que muchos de los alumnos no disponen de conocimientos de robótica, no se les puede decir que realicen control de movimiento del brazo robot por cinemática inversa (dado un punto en el sistema de coordenadas global hacer que la pinza del brazo robot se posicione en el). No obstante, aún se les puede pedir que muevan las distintas articulaciones teniendo en cuenta los finales de carrera de cada una de ellas.

#### **4.2. Control de un robot móvil con MarteOS.**

Otro de los trabajos que propuesto este curso es el control de un robot móvil (Figura 3). El robot en cuestión mide 27 mm de radio y 2 cm de espesor. Está provisto de dos ruedas, controladas por dos motores de corriente continua. Además, estas ruedas tienen cada una un encoder incremental que proporciona información de odometría. El sistema de sensores lo forman ocho sensores de infrarrojos con un alcance de 60 mm, situados a lo largo del perímetro del robot.



**Figura 3 Robot móvil Khepera.**

El trabajo consiste en que el robot debe seguir un haz de luz (proporcionado por una linterna) para ello deben analizar la información de los sensores de infrarrojos para determinar si hay luz y que deben hacer para encararse a ella y avanzar, siguiéndola.

Previamente se había utilizado el mismo robot [1] pero con otro tipo de control, en este caso lo que debía hacer el robot era alcanzar objetivos (introducidos por el usuario) evitando todos los obstáculos que se encontrara por el camino. Esta es otra alternativa para realizar trabajos, aunque en el caso de MarteOS las aplicaciones sin E/S funcionan mucho mejor debido al tiempo que se pierde realizando este tipo de operaciones. En el caso del seguimiento de la luz, el robot es mucho más autónomo y las acciones de usuario no implican ninguna operación de E/S, únicamente han de mover la linterna para que el robot reaccione.

#### **5. Conclusiones y trabajo futuro.**

La iniciativa de introducir la programación de sistemas empotrados en los trabajos de asignatura ha sido un éxito entre el alumnado. Este tipo de trabajos prácticos tienen como objetivo que el alumno se enfrente con problemas reales. La experiencia ha sido totalmente satisfactoria, ahora no sólo disponen de procesos reales con los que interactuar, sino que el medio a través del que interactúan también está mucho más cercano al mundo real.

Como trabajo futuro se va a plantear la posibilidad de desarrollar todas las aplicaciones de la asignatura en este entorno cruzado de programación. La idea es que el alumno desde su primera práctica (un simple *Hola Mundo*), utilice el entorno de desarrollo de MarteOS, para que su adaptación sea más paulatina. Esto permitirá que trabajen durante todo el curso con un entorno de tiempo real.

### Referencias

- [1] S. Terrasa, P. Balbastre, A. Crespo. “La importancia del uso de procesos físicos reales en la enseñanza universitaria en la ingeniería”. JENUI’2000, pp.304-310
- [2] S. Terrasa, P. Balbastre, A. Crespo. “Time Logger: una herramienta software de soporte a la docencia de los sistemas operativos de tiempo real”, JENUI’2001, pp 173-178.
- [3] S. Terrasa, P. Balbastre, A. Crespo. “Estudio de técnicas de Planificación en Sistemas de Tiempo Real de Laboratorio”, Jornadas de Automática 2000, ISBN 84-699-3163-6
- [4] MarteOS : <http://marte.unican.es>
- [5] A. Crespo, J. Vila, F. Blanes, I. Ripoll. “Real-Time Education in a Control Engineering Curriculum”. Real-Time Systems Education. IEEE Computer Society Press.