

# Práctica de desarrollo de interfaces hardware/software para la monitorización del estado de un PC

Juan P. Dominguez-Morales, A. Jimenez-Fernandez, A. Rios-Navarro, R. Tapiador-Morales,  
D. Gutierrez-Galan, Manuel J. Dominguez-Morales, Elena Cerezuela-Escudero  
Departamento de Arquitectura y Tecnología de Computadores  
Universidad de Sevilla  
Sevilla

jpdominguez@atc.us.es, ajimenez@atc.us.es, arios@atc.us.es, ricardo@atc.us.es,  
dgutierrez@atc.us.es, mdominguez@atc.us.es, ecerezuela@atc.us.es

## Resumen

Este artículo presenta una práctica laboratorio impartida mediante una metodología de aprendizaje basado en proyectos (ABP) [1] para dotar de la capacidad de diseñar y desarrollar un monitor del estado de un ordenador, integrado en un sistema empotrado que se comunica con una aplicación de escritorio, a nuestros alumnos de la asignatura de Diseño de Microcontroladores (DM) en el contexto del Máster en Ingeniería de Computadores y Redes.

Esta práctica abarca la comunicación Hardware/Software entre un microcontrolador con un núcleo Cortex-M4 y una aplicación software escrita en lenguaje C# usando el entorno Visual Studio Community 2015 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 PC con un sistema operativo de la familia Windows, en el que se construye una aplicación visual mediante Windows Forms, la cual obtiene información del sistema de forma periódica y la envía al microcontrolador mediante comandos usando el puerto serie (USB o comunicación Bluetooth). Por otro lado tenemos un microcontrolador de la familia STM32 que dispone de un display LCD ejecutando una plataforma completamente libre, .NET Micro Framework, la cual recibe a través del puerto serie la información obtenida gracias a la aplicación software del PC y la muestra en la pantalla, obteniendo así una herramienta de monitorización del PC sin tener que estar conectado físicamente a éste.

El desarrollo de este tipo de proyectos se añade la dificultad de 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 firmware para un microcontrolador usando la plataforma .NET MF y de su comunicación con el PC por medio de una aplicación visual.

## Abstract

This manuscript presents a practical laboratory session imparted using a project-based learning methodology (PBL) to provide the capacity of designing and developing a computer status monitoring device, integrated in an embedded system that communicates with a desktop software tool, to our students in the Computer Engineering Master's Degree.

This practice session encompasses Hardware/Software communication between a microcontroller with a Cortex-M4 kernel and a desktop software application through virtual COM ports (VCP) written in C# using Visual Studio Community 2015. This lab session is focused as a project that students must be making from scratch by achieving and completing some milestones to obtain a final functional system. The project is divided into two different parts. First, we have a Windows PC where a visual software application that gathers information from the system and sends it periodically to the microcontroller (USB or Bluetooth) has to be built using Windows Forms. On the other hand, we have a microcontroller from the STM32 family that has a 2.4" LCD display executing .NET Micro Framework that receives the information obtained from the PC through the serial port and displays it in the screen. This way, students create a computer status monitoring tool that does not need to be connected physically to it to receive the information.

The development of this project is added to the need of using different tools for firmware and software development, 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 firmware development for a microcontroller using the .NET MF platform as well as the communication

between the PC and the microcontroller using a visual software application.

## Palabras clave

Aprendizaje basado en proyectos, microcontrolador, STM32, Visual Studio, C#, .NET Micro Framework, Virtual COM Port, Windows Forms.

## 1. Motivación

La asignatura DM se imparte dentro del plan de estudios del Máster en Ingeniería de Computadores y Redes. El principal objetivo de esta asignatura es reforzar el conocimiento de los estudiantes en el ámbito de la programación para microcontroladores e introducirles en el desarrollo de firmware para diversos sistemas operativos de tiempo real. A lo largo de esta asignatura se realizan prácticas incrementales en cuanto a conocimiento que refuerzan lo aprendido en sesiones anteriores al mismo tiempo que plantean nuevos retos con diferente tipo de dispositivos externos como actuadores, sensores, etc. Desde las primeras clases de esta asignatura se utiliza un microcontrolador de la familia STM32, estudiando desde su configuración y funcionamiento a bajo nivel como la comunicación con una serie de periféricos externos como motores, sensores, manejo de buses I2C y SPI, y módulos para la comunicación inalámbrica usando el estándar 802.15.4.

Utilizar un microcontrolador permite acercar a los usuarios a la programación a bajo nivel. Sin embargo, en otras asignaturas de los grados relacionados con informática no se contempla la integración entre sistemas de este tipo con aplicaciones de escritorio de alto nivel. La comunicación entre estos dos sistemas permite aprovechar la potencia y reducido que ofrecen los microcontroladores, al mismo tiempo que se presenta al usuario de forma cómoda la información o resultados obtenidos. Es por ello que se ha pretendido realizar una práctica de laboratorio dividida en dos sesiones de dos horas cada una con la que los alumnos puedan experimentar con el material y aprender a realizar una aplicación específica, que les motive a aprovechar el tiempo durante las sesiones de laboratorio y a seguir experimentando por su cuenta para conseguir desarrollar nuevas aplicaciones gracias a que todo el entorno software utilizado es gratuito.

## 2. Objetivos

Debido a que el alumnado pertenece al Máster relacionado con la rama de informática más cercana al hardware y que es una continuación de asignaturas del Grado en Ingeniería de Computadores como Sistemas Empotrados en Tiempo Real, se parten de unos conocimientos básicos por parte de éste sobre programación de microcontroladores, lenguajes de programación orientada a objetos (POO) y comunica-

ción por medio de puertos serie. Bajo esta premisa, se fijaron los objetivos que se detallan a continuación.

En primer lugar, se desarrollará una herramienta haciendo uso del lenguaje C# desde el entorno de desarrollo Visual Studio. Esta herramienta obtendrá información del PC sobre el que se ejecuta mediante el uso de la clase System.Management y se realizará dentro de una aplicación de usuario llevada a cabo con Windows Forms, que es un conjunto de librerías que permiten realizar interfaces gráficas de forma sencilla. Además, debe ser capaz de enviar la información de forma empaquetada a través del puerto serie, para que posteriormente esos datos puedan ser recibidos por el microcontrolador.

En segundo lugar, y por primera vez para los alumnos del Máster, se hace uso de un lenguaje de alto nivel (C#) para programar un microcontrolador. Para ello se usa el framework de Microsoft .NET MF, que permite compilar y ejecutar código en el microcontrolador a través de Visual Studio. Se desarrollará el firmware apropiado para recibir paquetes del puerto serie, tratarlos y mostrarlos al usuario de forma cómoda a través del display LCD en el que, mediante un menú táctil, podrá elegir a qué tipo de información desea acceder.

En la Figura 1 se aprecia un diagrama de los distintos componentes con los que se trabajarán en la sesión práctica y la interconexión entre ellos.

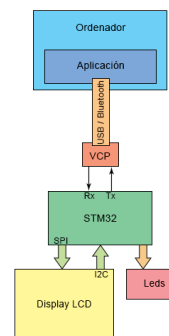


Figura 1: Diagrama de componentes del sistema

## 3. Material utilizado

Se describen a continuación cada uno de los elementos utilizados tanto hardware como software (programas) en el desarrollo de la práctica que se divide en dos sesiones de dos horas cada una:

- Placa de desarrollo STM32F429 Discovery. Se trata de una placa de bajo coste que consta de un microcontrolador de la familia STM32<sup>1</sup> de 32 bits, a 180 MHz y con 2 Mbytes de memoria flash. Además, posee una unidad de punto flotante (FPU), varios leds (actuadores), botones (sensores) y un display QVGA TFT LCD táctil de 2.4 pulgadas. Puede verse en la Figura. 2.

<sup>1</sup> STMicroelectronics, "STM32." [Online]. Disponible: <http://www.st.com/web/en/catalog/mmc/FM141/SC1169?sc=stm32>



Figura 2: Placa de evaluación STM32F429 Discovery

- Adaptador Bluetooth USB. Se conecta en un puerto USB del PC y transmite información por medio de un VCP. Se muestra en la Figura 3.
- Módulo Bluetooth HC-06. Se conecta a los pines de la placa y recibe la información que envía el PC a través del adaptador Bluetooth USB mencionado anteriormente. Mostrado en la Figura 3.



Figura 3: BT-USB (izq.) y módulo HC-06 (der)

- Conversor de TTL serie a USB (Crius FTDI). Conecta pines serie de la placa a un puerto USB del PC, generando un puerto COM virtual a través del cual enviar y recibir información. Usado en el caso de que se prefiera una conexión cableada en vez de la solución inalámbrica con Bluetooth. Se muestra en la Figura 4.



Figura 4: Crius FTDI

- Visual Studio 2015 Community. Versión gratuita del famoso entorno de desarrollo de Microsoft. Permite el desarrollo de aplicaciones con interfaz de usuario visual de forma simple y cómoda mediante la programación con lenguaje C#. Lo utilizaremos para programar tanto la aplicación que se ejecutará en el PC como el firmware que se escribe en la memoria del microcontrolador.
- .NET Micro Framework. Plataforma open source que permite escribir aplicaciones para sistemas empuados en C# utilizando Visual Studio. Utilizado para programar el firmware de la placa.

#### 4. Desarrollo de la sesión práctica de laboratorio

Utilizando los materiales descritos en la sección anterior se detallan a continuación los distintos hitos y problemas propuestos a los alumnos para resolver durante las dos sesiones que forman esta práctica de laboratorio de DM. Además, se resuelven de forma parcial cada uno de ellos.

#### 4.1. Creación de la aplicación en Visual Studio Community 2015

La práctica podría comenzar tanto por la creación de la aplicación de usuario como el firmware del microcontrolador, ya que ambos se realizan en lenguaje C# a través del entorno Visual Studio. Sin embargo, se ha decidido empezar con la primera por su mayor facilidad de desarrollo y por tratarse de algo gráfico, lo cual resulta más llamativo y motiva más al alumno por el constante feedback que obtiene tras cada cambio en el código.

Debido a que C# es un lenguaje que se ha trabajado con poca frecuencia en asignaturas precedentes, se le introduce al alumno la creación y la configuración de un proyecto básico a partir del cual pueden comenzar a incluir funcionalidades propias de la práctica.

#### 4.2. Obtener información del PC

Para esta tarea se hace uso de un software gratuito de Microsoft llamado Windows Management Instrumentation (WMI) Code Creator<sup>2</sup>. Dicha aplicación permite obtener cualquier información accesible del sistema (y de cada uno de sus componentes) y proporciona el código en lenguajes Visual Basic y C# que se necesita escribir para obtenerlo. Además, permite la ejecución del código en la propia aplicación para ver qué resultado nos proporcionaría. La Figura 5 muestra la ventana principal de WMI Code Creator en la que se aprecian distintas propiedades posibles, el código necesario para obtener el nombre del procesador y el resultado.

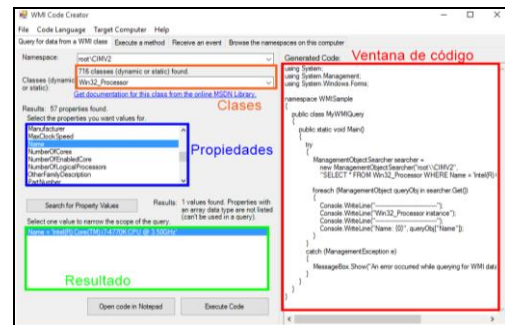


Figura 5: WMI Code Creator

#### 4.3. Envío de paquetes a través del puerto serie

Una vez obtenida la información del sistema hay que enviarla desde la aplicación del PC al microcontrolador por medio del puerto serie. Antes de realizar esto es necesario empaquetar los datos en una estructura conocida para que posteriormente, al recibirla en el otro lado, se pueda desempaquetar siguiendo el orden establecido al enviarla.

<sup>2</sup> Microsoft, "WMI Code Creator." [Online]. Disponible: <https://www.microsoft.com/en-us/download/details.aspx?id=8572>

La Figura 6 muestra cada uno de los campos de la estructura de paquete propuesta. El inicio y fin de trama son 0x7E y 0x0D (valores utilizados por norma general para éste tipo de transmisiones), respectivamente. La longitud del paquete indica la cantidad de bytes de datos que contiene la trama, mientras que el byte asignado para especificar el comando indica al receptor qué tipo de información se le está enviando.

Desplazamiento	Descripción	Dato
+0	Inicio de trama	0x7E
+1	Longitud del paquete	0x03
+2	Comando	0x0C
+3	Dato [0]	0x44
+4	Dato [1]	0xB2
+5	Dato [2]	0x15
+6	Fin de trama	0x0D

Figura 6: Estructura del paquete de datos

#### 4.4. Instalación de Microsoft .NET MF

Para poder programar el microcontrolador usando código escrito en C#, es necesario escribirle un firmware concreto, además de preparar y configurar el entorno de desarrollo con una serie de herramientas concretas. Todo este proceso está recogido en un manual de usuario [2] que facilita STM, empresa que fabrica el microcontrolador, y que consta de una guía paso a paso para la correcta instalación.

#### 4.5. Creación de la interfaz de usuario

El contenido descargable disponible en el manual de instalación y configuración de .NET Micro Framework ofrece una serie de ejemplos de código que hacen uso de los distintos tipos de funciones que nos permiten trabajar con la pantalla táctil de nuestra placa de evaluación. El alumno deberá hacer uso de dichas funciones para capturar los eventos de pulsación en la pantalla y, según en qué sección se haya presionado, muestre la información que corresponda.

#### 4.6. Recepción de información

Finalmente, para poder recibir en el microcontrolador la información enviada desde la aplicación software del PC que se desarrolló en los primeros hitos de la sesión práctica es necesario realizar una máquina de estados para organizar la recepción de paquetes y poder actuar ante las tramas que llegan.

La máquina de estados espera hasta recibir el inicio de trama (0x7E) y, hecho esto, se atiende al siguiente byte (longitud de los datos) para ir recogiendo la información hasta que se llega al fin de cadena (0x0D). Con los datos ya recogidos, se comprueba a qué comando (primer byte después de longitud de la trama) hace referencia la información obtenida y se modifica la cadena de texto correspondiente que se representa en el display.

Terminado este hito se comprueba el desarrollo completo de la sesión práctica, probando que se

recibe y representa de forma correcta en el display del microcontrolador la información del PC al que está conectado.

## 5. Resultados

Para tener un feedback sobre el grado de aceptación del alumnado ante el desarrollo de la práctica que se presenta se entregó una encuesta anónima tras finalizar la segunda sesión, invitando a los alumnos a participar, en la que se pedía qué valor de satisfacción habían obtenido en diferentes aspectos de la práctica. La media de satisfacción es de 4.78 (de un máximo de 5), siendo un resultado muy favorable.

Además de los resultados obtenidos, más de un 60% de los alumnos que asistieron a esta práctica realizaron el Trabajo Fin de Máster con el departamento que imparte la asignatura, mostrando un gran interés por seguir desarrollando aplicaciones relacionadas con la comunicación entre PC y microcontrolador y, más concretamente, por seguir trabajando con la familia de microcontroladores usados durante las sesiones de laboratorio.

## 6. Conclusiones

En el presente trabajo se ha presentado una metodología de aprendizaje basada en proyectos e hitos aplicados a una práctica de desarrollo hardware. Para realizarla en su totalidad, se ha dividido el contenido en dos sesiones de dos horas cada una, de forma que el alumno pueda experimentar con el material durante la semana entre ambas sesiones, fomentando así el trabajo de investigación.

Además de haber sentado las bases para numerosos Trabajos de Fin de Máster, las encuestas anónimas repartidas a los alumnos reflejan un muy alto grado de motivación y de implicación con la metodología utilizada para el desarrollo de la práctica.

## 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.

## Referencias

- [1] G. Solomon, "Project-based learning: A primer," *Technol. Learn.*, vol. 23, no. 6, pp. 20–30, 2003.
- [2] STMicroelectronics, "Getting started with .NET Micro Framework on the STM32F429 Discovery kit." [Online]. Disponible: [http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user\\_manual/DM00096190.pdf](http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user_manual/DM00096190.pdf).