



Emulación de dispositivos I2C para formación a distancia en sistemas empotrados basados en microcontroladores.

José Manuel Cano García, José Borja Castillo Sánchez y Eva González Parada^a

^a Todos los autores pertenecen al Dpto. de Tecnología Electrónica - E.T.S.I. de Telecomunicación - Universidad de Málaga. Los e-mails son, respectivamente, (jcgarcia@uma.es) (ioscassan@uma.es) y (gonzalez@uma.es)

Abstract

This paper proposes the implementation of emulated I2C slave devices, which can be connected to and controlled from a microcontroller acting as the I2C master. These emulated I2C devices are used as a tool to enhance both the student learning and evaluation processes in online courses on embedded microcontroller-based systems with an experimental approach.

Resumen

En este trabajo se propone la implementación de dispositivos I2C esclavos emulados mediante *firmware*, que pueden controlarse desde un dispositivo microcontrolador configurado como maestro I2C, como una herramienta para facilitar tanto el aprendizaje de los estudiantes como la evaluación de los mismos en el ámbito de asignaturas relacionadas con sistemas empotrados basados en microcontroladores con docencia a distancia pero con un enfoque práctico y experimental.

Palabras clave: Sistemas empotrados, Microcontroladores, IoT, Lab-in-a-box

1. Introducción y objetivos

El presente trabajo recoge una propuesta para la implementación de algunas actividades prácticas para la docencia relacionada con sistemas empotrados basados en microcontrolador en el ámbito del Máster Oficial en Sistemas Electrónicos para Entornos Inteligentes (MSEEI) de la Universidad de Málaga [1], que es una titulación oficial de nivel de máster con enseñanza a distancia, pero con un carácter eminentemente práctico en su desarrollo. El máster tiene una extensión de 60 ECTS y su temática se centra en el estudio y aplicación de las diferentes plataformas de computación para la Internet de las cosas (sistemas empotrados basados en diversas plataformas como microcontroladores, microprocesadores y FPGAs), tecnologías de interconexión (comunicaciones inalámbricas de bajo consumo, middleware), así como tecnologías para el procesamiento de información, actuación y control en entornos inteligentes (inteligencia artificial, visión artificial, interacción e interfaces de usuario).

Para mantener un enfoque experimental en la mayoría de las asignaturas del máster, es importante facilitar a los estudiantes el acceso a herramientas de desarrollo hardware y software profesionales y basadas en dispositivos reales. Para ello a los estudiantes de nuevo ingreso se les hace llegar al comienzo del master, y en régimen de préstamo, un kit con diversas plataformas de desarrollo hardware para microcontroladores, microprocesadores y FPGA (por ejemplo en el curso 22/23 se han utilizado las plataformas ESP32 Wrover kit, MSP430F5529 launchpad, Raspberry Pi 3B, Digilent Basys 3,...) así como componentes

discretos que pueden conectar a los mismos. Todas estas plataformas cuentan con entornos de desarrollo gratuitos que se ponen a disposición de los estudiantes en el campus virtual, de forma que pueden utilizarlos para poder ejecutar y probar los ejemplos suministrados y realizar los desarrollos que se le pidan en cada asignatura. Esta metodología de trabajo, que también se conoce con el nombre de *Lab-in-a-box*, está siendo cada vez más utilizada por instituciones internacionales de reconocido prestigio [2]–[4].

Las actividades prácticas propuestas en este trabajo se han aplicado en las asignaturas relacionadas con los sistemas empotrados basados en microcontroladores (aunque también podrían extenderse en un futuro a otras plataformas IoT utilizadas en el máster). La fig. 1 muestra las arquitecturas *hardware* y *software* típicas de las actuales plataformas IoT basadas en microcontroladores. A lo largo de diferentes asignaturas del máster se deben abordar tanto aspectos relacionados con el funcionamiento del hardware de estas plataformas (arquitectura de la CPU, periféricos de entrada/salida, interfaces de interconexión, etc.) como del desarrollo de aplicaciones para (utilización de bibliotecas de drivers de periféricos, sistemas operativos de tiempo real, *stacks* de comunicaciones y *middleware*).

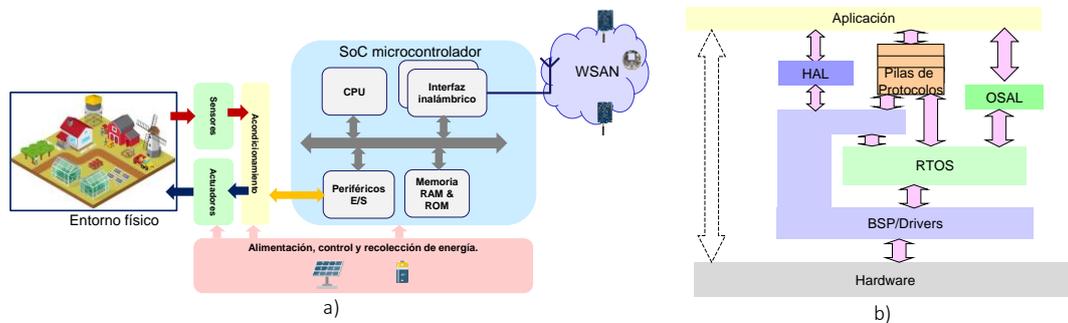


Fig. 1 Arquitectura hardware (a) y software (b) de una plataforma para IoT basada en microcontrolador.

En el caso de las plataformas IoT utilizadas en entornos inteligentes, es importante tanto la capacidad de interactuar con el entorno (es decir, recoger información mediante sensores y la actuación sobre el mismo) como la capacidad de comunicación con el resto del sistema inteligente (mediante los interfaces apropiados). Por este motivo, en las asignaturas del MSEEI se presta especial atención a la conexión de sensores y actuadores tanto mediante periféricos analógicos (ADC y DAC) del microcontrolador, como mediante interfaces serie síncronos como I2C o SPI. En lo que respecta al *software/firmware* de los dispositivos, la mayoría de plataformas IoT actuales integran un sistema operativo de tiempo real (RTOS) para permitir una multitarea básica en sistemas con importantes restricciones de memoria y capacidad de procesamiento como son los microcontroladores. Esta multitarea es necesaria a su vez para permitir la ejecución concurrente de los diferentes procesos que corresponden a la aplicación y a la gestión de las comunicaciones y facilita el particionado e implementación modular del *software* que ejecuta en el microcontrolador.

En la actualidad existen muchas herramientas y plataformas de desarrollo de aplicaciones para IoT que ocultan al desarrollador la complejidad tanto del hardware como del software de los dispositivos, ofreciendo una interfaz de programación (API) de alto nivel que permite la lectura de datos de los sensores, su procesamiento y su envío hacia el resto del sistema o la actuación sobre el entorno. Sin embargo, este no es el enfoque que se pretende seguir en el MSEEI, ya que está orientado a postgraduados de titulaciones del ámbito de la ingeniería electrónica, que persiguen una especialización profesional en el desarrollo de este tipo de sistemas, y a los que se les supone ya una formación previa en la materia. Por ello, como objetivo de las

diferentes asignaturas se plantea no sólo que el estudiante aborde el estudio del funcionamiento de los diferentes periféricos y aprenda a utilizar los drivers suministrados por los fabricantes para gestionarlos, así como los mecanismos del RTOS para la gestión de la concurrencia, sino que también se pretende que el estudiante sea capaz de extraer información de la documentación técnica de los dispositivos (hojas de características, manuales del desarrollador) y utilizarla para desarrollar sus propios drivers en lugar de utilizar los proporcionados por terceros. Los drivers desarrollados por los estudiantes deberán además estar adecuadamente encapsulados en una biblioteca que pueda ser reutilizada por terceros, con un interfaz de programación (API) clara y bien estructurada y deberán utilizar correctamente los recursos del RTOS para permitir una ejecución eficiente de las aplicaciones que los utilicen, facilitando su integración con el resto de componentes del software del dispositivo. Se persigue con ello reforzar en los estudiantes las competencias relacionadas con la interpretación de la documentación técnica y con el diseño y producción de módulos software que cumplan con los requisitos de calidad especificados.

La utilización de dispositivos I2C en el contexto de estas actividades previstas en diversas asignaturas del máster resulta especialmente interesante porque permite desarrollar las competencias previstas en un entorno realista. El interfaz I2C es un bus serie síncrono que permite expandir un sistema basado en microcontrolador (maestro del bus) conectándolo con uno o más dispositivos esclavos que implementan diversas funciones específicas. En el mercado existen numerosos dispositivos comerciales con este interfaz con numerosas aplicaciones, que van desde el propósito general, como la conversión digital/analógica de señales, expansión de puertos, control PWM; a tareas muy especializadas como medidas fisiológicas o de magnitudes climáticas. Dada la inmensa variedad de aplicaciones, la gran mayoría de los microcontroladores actuales integran dentro de su juego de periféricos uno o varios maestros I2C que permiten su conexión con estos dispositivos, lo que hace posible utilizarlos para desarrollar las competencias aquí planteadas.

La incorporación de actividades prácticas que implican la conexión de uno o varios dispositivos esclavos I2C a un microcontrolador que actúa como maestro, permite introducir los principios de funcionamiento del bus I2C a nivel electrónico y la organización interna de los dispositivos esclavos I2C más habitualmente utilizada por los dispositivos comerciales (un banco de registros de 8 bits propósito específico accesibles mediante operaciones I2C de escritura o de escritura-lectura, cuya funcionalidad es definida por el fabricante de cada dispositivo), instruir al estudiante sobre el funcionamiento del periférico maestro I2C integrado en el microcontrolador y su programación mediante los drivers proporcionados por el fabricante como parte del software de soporte, familiarizar al estudiante con la consulta e interpretación de la documentación proporcionadas por los fabricantes de los dispositivos I2C sobre su funcionamiento (operación utilizando los registros internos), y ejercitarle en la implementación de drivers propios para la gestión de estos dispositivos en base a la información extraída de dicha documentación.

En la siguiente sección se proporcionan más detalles acerca de las actividades que se pide realizar a los estudiantes y sobre el papel que juega la emulación de algunos dispositivos I2C en el proceso de formación y evaluación de las asignaturas implicadas.

2. Metodología

En las diferentes asignaturas del máster relacionadas con los sistemas empotrados basados en microcontrolador se sigue un enfoque orientado a proyectos en el que, por una parte, se

proporciona al estudiante información, acompañada por ejemplos sencillos, sobre aspectos como los principios de funcionamiento de los diferentes periféricos del microcontrolador, o como el uso de los RTOS para la gestión eficiente de las tareas y de su interacción con las rutinas de interrupción que atienden los eventos relacionados con los periféricos hardware; y por otra parte, se les pide a los estudiantes completar un proyecto de aplicación IoT de complejidad media/alta que se les da implementado parcialmente y al que tienen que ir añadiendo nuevas funcionalidades durante el desarrollo de las diferentes asignaturas. Esta aplicación IoT que debe ampliar el estudiante implica la captura de datos del entorno a través de diversos interfaces del microcontrolador y de sensores conectados al mismo y su envío hacia el resto del sistema inteligente mediante diferentes protocolos y tecnologías de comunicación (TCP, MQTT, o interfaces inalámbricas como BLE). Entre las nuevas funcionalidades que el estudiante debe incorporar a la aplicación se encuentra la medida de nuevas magnitudes del entorno mediante la conexión de sensores adicionales al microcontrolador a través de sus interfaces, entre ellos el bus I2C.

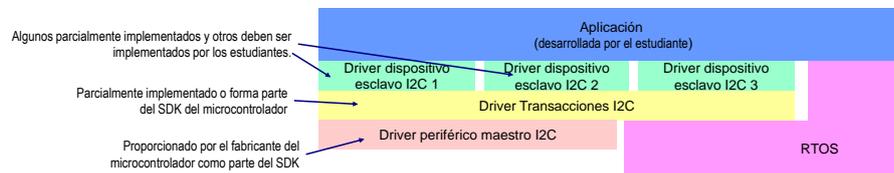


Fig. 2 Componentes del software del microcontrolador involucrados en la actividad.

Como ya se ha mencionado, la incorporación de sensores I2C al sistema obliga al estudiante no sólo a comprender el funcionamiento del bus I2C, sino también a estudiar el periférico maestro I2C del microcontrolador (y su correspondiente driver de soporte), a consultar la documentación de los de los sensores I2C proporcionada por sus fabricantes y a desarrollar las correspondientes bibliotecas de soporte (drivers) para su gestión desde la aplicación de una forma eficiente y que permita aprovechar los recursos multitarea que proporciona el RTOS. En la fig. 2 se muestran los diferentes componentes software del sistema que se verían involucrados en la actividad, indicando si son componentes del software del microcontrolador proporcionados por terceros que el estudiante debería aprender a utilizar (como el RTOS o los drivers del periférico maestro I2C del microcontrolador) o si por el contrario son componentes que debería desarrollar o completar el propio estudiante (como los drivers de los dispositivos esclavos y/o la propia aplicación). Para añadir las nuevas funcionalidades requeridas el estudiante cuenta (además de con el código de la aplicación básica de partida) con ejemplos suministrados por los profesores que ilustran el funcionamiento del periférico maestro I2C, y con ejemplos de implementación (parcial) del *driver* de algún sensor I2C concreto.

Dentro de los sensores I2C que el estudiante debe agregar al sistema se encuentran algunos sensores comerciales (de humedad, temperatura y presión; inerciales) que se le envían al estudiante junto con el resto de sistemas de desarrollo y componentes que forman parte del "kit *Lab-in-a-box* del máster" descrito en la sección 1, junto con otros sensores I2C que son emulados y no corresponden con dispositivos comerciales reales, aunque sí se comportan de forma muy similar a dispositivos disponibles comercialmente. Estos dispositivos esclavos I2C emulados, han sido desarrollados por los autores de este trabajo para facilitar tanto el aprendizaje como la evaluación de los estudiantes. Para cada dispositivo se han implementado dos variantes diferentes de emulación:

- Dispositivos I2C emulados "internos": Se implementan utilizando los periféricos I2C esclavo del microcontrolador que el estudiante está utilizando para realizar el desarrollo

de su aplicación, junto con otros recursos hardware del micro. El estudiante tiene que añadir una biblioteca de emulación a su proyecto e invocar a una función de inicialización. A partir de ese momento el dispositivo emulado puede ser accedido a través del bus I2C como si se tratase de un dispositivo I2C real. Los periféricos del microcontrolador utilizado en la emulación quedan reservados para ésta y no pueden ser accedidos directamente por el estudiante, ni utilizado para la realización de otras funcionalidades.

- Dispositivos I2C emulados “externos”: En este caso el dispositivo se implementa configurando como un microcontrolador diferente a aquel que el estudiante está utilizando para su aplicación. El estudiante debe grabar un firmware (suministrado por los profesores) en este micro externo, que realiza las funciones del dispositivo emulado utilizando sus propios periféricos. El micro externo debe ser conectado al bus I2C del microcontrolador utilizado por el estudiante como maestro para poder controlarlo. Este tipo de emulación no “consume” ningún recurso hardware del micro maestro, por lo que todos sus periféricos pueden seguir siendo utilizados libremente por el estudiante.

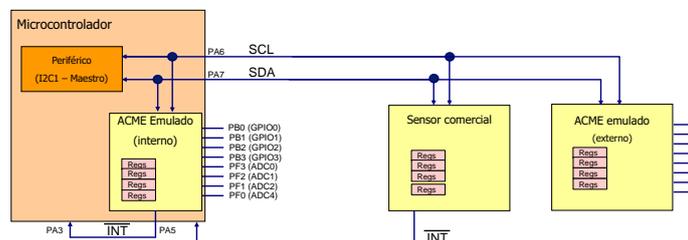


Fig. 3 Topología del bus I2C con sensores emulados externa e internamente.

Al microcontrolador maestro se le podrán conectar simultáneamente múltiples sensores a través del bus I2C, algunos de ellos dispositivos comerciales reales, y otros, dispositivos emulados externos o internos, tal y como se ilustra en la figura 3.

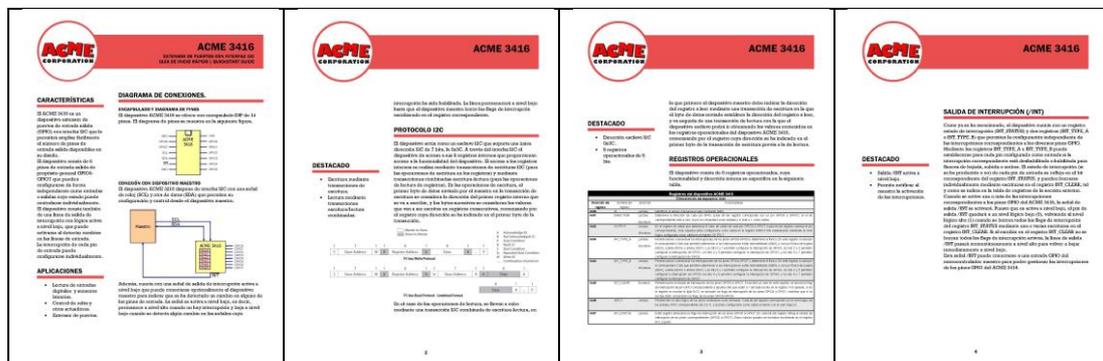


Fig. 4 Ejemplo de la “guía rápida del desarrollador” de uno de los dispositivos emulados

Además de la implementación de diversos modelos de dispositivos I2C emulados con diferentes funcionalidades (por ejemplo, extensor de puertos GPIO, controlador PWM, conversores ADC, DAC y comparador digital, etc.), en sus variantes “interna” y “externa”, también se ha generado una documentación técnica asociada, en la que se describe el funcionamiento de cada dispositivo emulado de una forma similar a la que lo hacen las hojas de características y/o las guías de desarrollo de los dispositivos comerciales. Estas “guías del desarrollador” de los dispositivos emulados, no son tutoriales o “apuntes” sobre la utilización de estos dispositivos, sino que siguen un formato similar (aunque simplificado en algunos puntos) al que suelen tener los correspondientes documentos de dispositivos comerciales reales. La fig. 4 muestra un ejemplo de uno de estos documentos, que deberán ser analizados

y consultados por los estudiantes para el desarrollo de los módulos que deben realizar para completar la implementación de su aplicación IoT.

3. Resultados y Conclusiones

La introducción de los sensores I2C emulados supone una herramienta que facilita tanto el aprendizaje como la evaluación de los estudiantes en varios aspectos:

- Permite ampliar el catálogo de dispositivos I2C con los que el estudiante va a trabajar sin coste añadido, proporcionando algunos ejemplos adicionales de dispositivos I2C que el estudiante debe estudiar e incorporar a su proyecto.
- Las guías del desarrollador simplificadas generadas para acompañar a los dispositivos implementados suponen una introducción más amigable para los estudiantes a este tipo de documentación, ya que en muchos casos la documentación de los dispositivos comerciales es ambigua, incompleta o más compleja. En cualquier caso, el estudiante también deberá consultar la documentación de dispositivos comerciales reales, sirviendo la documentación de los dispositivos emulados como una primera toma de contacto.
- Los dispositivos emulados “internos” no requieren la realización de ningún conexionado, lo que elimina una fuente de errores potenciales que a veces hacen perder tiempo al estudiante. Por ello su utilización es más recomendable en las primeras actividades (o durante la evaluación). Una vez que el estudiante esté más familiarizado con el uso del bus y haya probado sus desarrollos puede continuar con la incorporación de dispositivos externos adicionales.
- La posibilidad de agregar nuevos dispositivos emulados “internos” en cualquier momento y sin conexionado supone también una herramienta para la evaluación del estudiante, ya que permite generar y añadir fácilmente al sistema nuevos dispositivos con características y organización interna diferente que tendrían que ser programados por el estudiante en base a su documentación durante las pruebas de evaluación, permitiendo comprobar así si ha adquirido las habilidades esperadas.

El software desarrollado para la emulación de sensores I2C, junto con documentación adicional para su puesta en marcha se hará público en el repositorio <http://u.uma.es/d1v/>.

4. Agradecimientos

El presente trabajo ha sido financiado en parte por el Proyecto de Innovación Educativa PIE22-163 y por el I Plan propio Integral de Docencia, ambos de la Universidad de Málaga.

5. Referencias

- [1] “máster SEEI - Master SEEI.” <https://www.masterseeiuma.es/> (accessed May 15, 2022).
- [2] Jay Weitzen, “‘Lab In A Box’ Gives Students Hands-On Experience,” *Electronic Design*, Sep. 2013. [Online]. Available: <https://www.electronicdesign.com/technologies/embedded/article/21798380/lab-in-a-box-gives-students-handson-experience> (accessed Jun 15, 2023)
- [3] F. A. Mujica *et al.*, “Teaching digital signal processing with Stanford’s Lab-in-a-Box,” in *2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*, 2015, pp. 307–312. doi: 10.1109/DSP-SPE.2015.7369571.
- [4] E. Nixon, “Lab-in-a-box and remote experiments: Education news from Imperial | Imperial News | Imperial College London.” <https://www.imperial.ac.uk/news/209193/lab-in-a-box-remote-experiments-education-news-from/> (accessed May 15, 2022).