

Análisis comparativo de distintas plataformas para la enseñanza de Sistemas Electrónicos Digitales

Manuel A. Perales, Federico J. Barrero, Sergio L. Toral
Departamento de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Camino de los descubrimientos s/n, 41092 Sevilla
mperales@us.es

Abstract— A la hora de enfocar la docencia de Sistemas Electrónicos Digitales, ha ido ganando sitio el uso de microcontroladores de bajo coste, entre otros factores gracias a su facilidad de uso y rapidez en obtener resultados de interés para los alumnos. Hay en el mercado numerosas plataformas de desarrollo rápido de sistemas basados en microcontrolador, lo que hace una tarea importante la búsqueda de la mejor opción. En esta comunicación se presentan tres de las más populares y se ofrecen datos comparativos para la selección de la plataforma adecuada.

Keywords— *Sistemas Electrónicos Digitales, Microcontroladores, Sistemas de desarrollo.*

I. INTRODUCCIÓN

Ante la profusión de sistemas de desarrollo asequibles (por debajo de 30€), el trabajo de selección de una plataforma adecuada se convierte en una tarea importante, que debe tomarse en cuenta a la hora de diseñar la asignatura. En este estudio se parte de la situación de una asignatura nueva, en la que se tenga absoluta libertad a la hora de elegir un sistema microprocesador u otro. Esta situación no siempre se da, debido a la inercia de los docentes a reutilizar elementos ya existentes, y por otro lado al gasto inherente a la hora de montar unas prácticas nuevas basadas en un sistema nuevo. Supuesto este marco de referencia, el siguiente paso es elegir una plataforma (software y hardware) con la que desarrollar la docencia.

En el presente trabajo se exponen una serie de reflexiones en torno a la necesidad de dedicar un tiempo a seleccionar cuidadosamente la herramienta que se vaya a usar en la docencia, no dejándose guiar por la inercia o por la popularidad de algunos sistemas. El fruto de estas consideraciones ha sido la selección de una serie de sistemas para varias asignaturas de electrónica digital en la Escuela Técnica Superior de Ingenieros de Sevilla, en concreto:

- Electrónica General, de 2º curso del Grado de Ingeniería en Tecnologías Industriales,

- Sistemas Electrónicos Digitales, de tercer curso de la misma titulación,
- Sistemas Electrónicos, de tercer curso del Grado de Ingeniería Electrónica, Robótica y Mecatrónica,
- Sistemas Electrónicos para la Automatización, de cuarto curso de la titulación anterior.

Las conclusiones a las que se llegan no son, en ningún caso, categóricas ni válidas para cualquier circunstancia, pero el método para la selección del sistema sí puede ser de utilidad.

II. CONSIDERACIONES PARA LA SELECCIÓN

La decisión de preparar una asignatura o un conjunto de ellas en torno a un sistema microcontrolador depende de muchos factores. Algunos aspectos que se deben tener en cuenta para ello serían:

- Precio de los sistemas de desarrollo, entendiéndose como tal el sistema de programación del microcontrolador, y el software para ello. Si se desea que los alumnos puedan manejar el microcontrolador, o incluso comprarse un sistema de desarrollo, éste debe estar por debajo de los 20€, por fijar un límite razonable.
- Entorno de programación cómodo, con posibilidad de combinar lenguajes de alto nivel de abstracción con lenguajes de menor nivel. Actualmente, prácticamente todos los microcontroladores ofrecen compiladores en C, C++, C#, siendo éstos los más usuales para programarlos. La gran calidad de los compiladores hace que no sea necesario, ni conveniente en la mayoría de los casos, programar en lenguaje ensamblador.
- Posibilidad de depuración en tiempo real. Para empezar con un sistema nuevo, es muy conveniente poder depurar los programas paso a paso, con visualización de valores de variables, registros internos, etc.
- Robustez de la plataforma hardware. Para un sistema que va a ser usado en prácticas, interesa que el diseño sea tolerante a cortocircuitos, inversiones de polaridad, etc. Aunque normalmente son condiciones que cumplen

los microcontroladores, es necesario comprobar dicho requisito, para evitar tener un alto coste de reposición.

- Ciclo de vida del producto. A la hora de elegir un nuevo sistema, hay que intentar evitar productos que lleven mucho tiempo en el mercado, por la posibilidad de que queden obsoletos pronto y no haya soporte en el futuro. Por otro lado, también hay que evitar productos demasiado recientes, de los que no se sabe a priori la longevidad que tendrán.
- Posibilidad de crecimiento. Es frecuente que, tras una primera asignatura introductoria de sistemas electrónicos digitales venga luego alguna otra más avanzada. O simplemente, es posible que interese mostrar soluciones más básicas y simultáneamente sistemas de mayor envergadura. Para ello, es conveniente que el sistema elegido tenga otros que presenten compatibilidad software o hardware (o ambas), que permitan reutilizar conocimientos o equipos.
- Base de conocimientos previos. Actualmente, existen multitud de foros, wikis, páginas específicas y repositorios de todo tipo en los que se pueden encontrar información y ejemplos de programación para los más diversos sistemas microcontroladores. Esta base de conocimientos es fundamental, tanto para el profesorado, a la hora de preparar una asignatura como para los alumnos.
- Posibilidad de ampliaciones con hardware prediseñado. Es frecuente que los fabricantes, o terceras empresas, comercialicen equipos para ampliar las plataformas de programación, y que faciliten de una u otra manera el desarrollo de éstos.
- Facilidad para diseñar hardware específico con el microcontrolador elegido. Resulta interesante que algún elemento de la familia del microcontrolador esté disponible en formato DIP. Aunque sea éste un formato bastante obsoleto, es fundamental a la hora de realizar prototipos de manera cómoda y barata.

Sin ser exhaustiva, esta lista contiene los aspectos que nos parecen más importantes a la hora de elegir una determinada plataforma. Dependiendo del caso concreto, se le dará más importancia a un ítem que a otro. Por ejemplo, si se van a montar más de 50 puestos de prácticas, conviene que el sistema hardware sea muy barato, y que el software de programación sea gratuito, para poder mantener los puestos en funcionamiento.

III. PLATAFORMAS ANALIZADAS

En el trabajo se presentan y se analizan tres plataformas hardware con sus diversas herramientas software aparejadas. No son las únicas existentes en el mercado, pero sí abarcan distintos campos de interés, siendo un conjunto suficiente para comenzar la búsqueda. Las plataformas analizadas son:

- Plataforma Arduino[1], basada en la placa Arduino UNO y en el entorno de desarrollo propio de Arduino, basado en Processing.

- Plataforma Launchpad de Texas Instruments[2], centrada en el Launchpad MSP430 Value Line y sus extensiones (MSP432, Tiva...) y programada con el entorno de desarrollo suministrado por el fabricante, Code Composer Studio
- Plataforma Freedom de NXP[3] (anteriormente Freescale), usando los elementos FRDM-KL25Z y FRD-K64F como sistema hardware, y el entorno de programación m-bed.

Se ha dejado fuera de la selección, de manera consciente, a la familia PIC de Microchip. Esto es debido a que es una familia de microcontroladores que, si bien ha sido muy importante y muy estudiada durante muchos años, está en clara desventaja con las otras plataformas estudiadas, en cuanto a capacidades, precio y otras opciones.

En todos los casos, se presentarán las plataformas hardware, comentando las similitudes y diferencias, tras lo que se compararán las posibilidades de programación y depuración en los distintos sistemas, así como la base de conocimientos disponible en Internet, tanto en las páginas oficiales de los sistemas analizados como en páginas independientes o foros.

A. Plataforma Arduino

Sin lugar a dudas, es actualmente la más popular entre los aficionados a la electrónica, por su facilidad de programación y por la ingente base de datos disponible de ejemplos en Internet[4] así como posibilidad de ampliar con hardware específico[5]. Igualmente, es muy utilizada por docentes, sobre todo de cursos básicos de electrónica o robótica [6]. Atendiendo a sus versiones oficiales (han surgido multitud de sistemas clónicos), existen cinco placas:

- Arduino UNO. Es la primera versión, e incluye un microcontrolador ATMEL de 8 bits a 16MHz. Es la única que tiene el microcontrolador en formato DIP.
- Arduino Micro: Mucho más pequeña que la anterior, usa un microcontrolador en formato SMD, pero la placa en sí tiene el factor de forma de un DIP de 34 pines.
- Arduino Mega: similar en forma a la UNO, pero con un microcontrolador más grande en cuanto a memoria y periféricos, aunque sigue siendo de 8 bits y 16MHz
- Arduino ZERO: Versión mucho más potente de la plataforma, con un microcontrolador basado en ARM Cortex M0+ a 48MHz.
- Arduino 101: en conjunción con Intel, incorpora un microcontrolador Intel Curie, a 32MHz, y algunos periféricos incluidos como un módulo Bluetooth BLE.

Esta plataforma fue la primera en popularizar el concepto de Open Hardware, ofreciendo los esquemas y planos de diseño de la misma, así como un sistema de programación muy sencillo. El lenguaje de programación usado, basado en el lenguaje Processing[7], usa librerías de abstracción hardware, que consiguen que el programador no tenga que dominar la configuración de los periféricos particulares del microcontrolador, centrándose en el proceso en sí.

El lenguaje de programación es realmente sencillo, aunque en aras de esa sencillez se pierde mucha de la capacidad del microcontrolador, que tampoco está nativamente muy dotado. Este es uno de los principales inconvenientes de esta plataforma a la hora de usarla como método para enseñar diseño de sistemas electrónicos digitales: la superposición de capas de programa hacen invisible el hardware, no ofreciendo la posibilidad, al menos al usuario no experto, de manejar los registros de configuración de los periféricos del microcontrolador.

Otra característica distintiva es el uso de un monitor de puerto serie como único método posible de depuración. Esto, sobre todo para usuarios poco expertos en programación, puede ser exasperante, por la dificultad para encontrar fallos en un programa sin poder ejecutarlo paso a paso. Algunas soluciones se han propuesto a esta carencia[8], aunque todavía no se cuenta con una solución estable.

De las plataformas analizadas, ésta es la única que sigue usando 5V como tensión de alimentación y de interfaz digital. Esta característica facilita la conexión de periféricos antiguos, pero dificulta mucho el uso de nuevos periféricos, que suelen ir alimentados y con conexión de datos de 3V

Atendiendo a sus capacidades hardware, si nos centramos en la placa original (Arduino UNO) se puede destacar que tiene, en total:

- 14 pines de E/S de los cuales 6 pueden ser pwm
- 6 entradas analógicas
- Conexión por puerto USB para programación
- Conector de alimentación independiente

El microcontrolador que implementa esta primera versión

de la plataforma Arduino es de Atmel, en concreto el ATMEGA328P, que cuenta entre sus características principales:

- Procesador de 8 bits a 16MHz
- 32KBytes de flash para programa
- 2Kbytes de Ram
- 1K de EEPROM
- Formato DIP28

Para concluir con la descripción de la plataforma, hay que comentar el gran número de placas auxiliares, *shields*, que existen en el mercado, compatibles con este sistema[5]

Por último, cabe destacar que el precio de estas placas está ahora mismo sensiblemente por encima de sus competidoras más directas, lo cual va en detrimento de su uso a largo plazo en asignaturas muy numerosas.

En la Fig. 1. se muestran dos de los sistemas (Arduino UNO y Arduino MEGA), en los que se aprecian algunas características distintivas, como sus conectores hembra a ambos lados, o el microcontrolador en formato DIP de la versión UNO.

B. Plataforma Freedom de NXP

La plataforma Freedom es la apuesta de Freescale (actualmente NXP) para entrar en el segmento de los sistemas de desarrollo de bajo coste. Frente a los otros dos sistemas analizados, en este caso no tenemos ningún miembro de la familia que use un microcontrolador en formato DIP. Esto en principio puede ser muy limitante, a la hora de desarrollar sistemas autónomos, dado que nos obligará a usar siempre el sistema de desarrollo como parte del equipo final. La

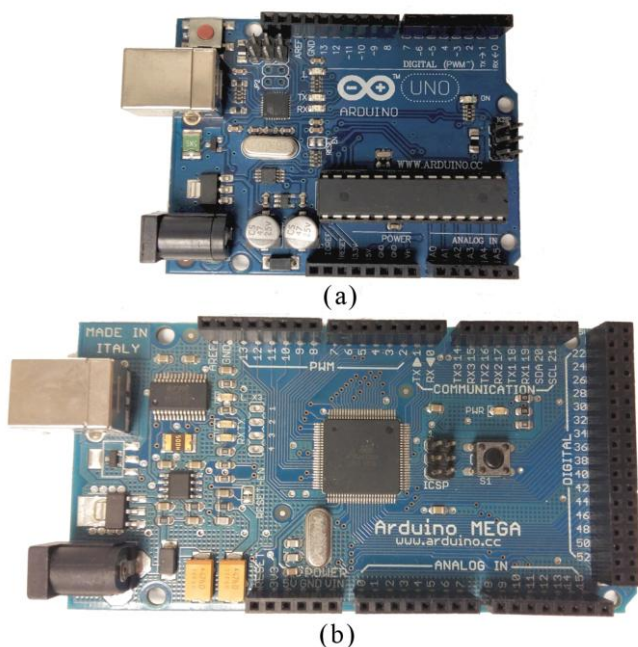


Fig. 1. Placas de Arduino UNO (a) y Arduino MEGA (b)

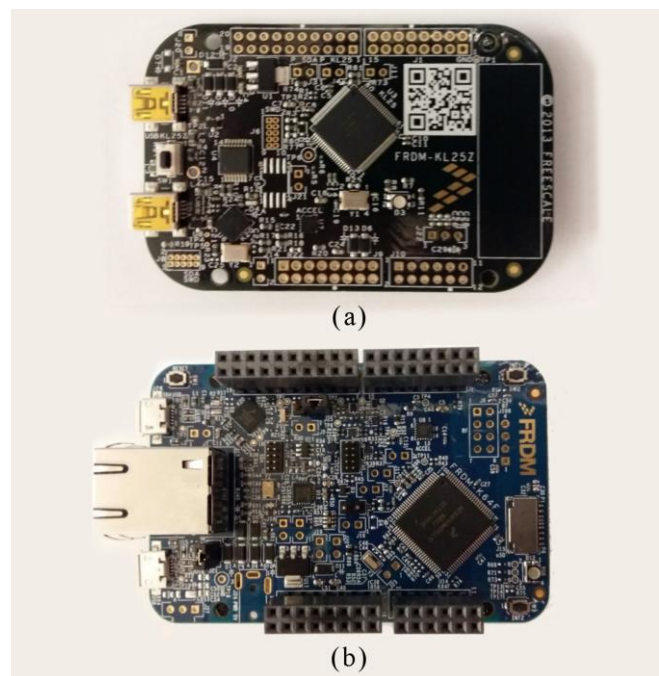


Fig. 2. Placas Freedom (a) KL25Z y (b) K64F

plataforma Freedom contiene gran cantidad de sistemas de desarrollo, de los cuales analizaremos dos de ellos en concreto, ambos con soporte mbed:

- **FRDM-KL25Z:** Es uno de los más básicos, contando con un microcontrolador ARM Cortex-M0+ a 48MHz, 128K de Flash y 16 K de SRAM. Aparte del microcontrolador, la tarjeta incorpora un acelerómetro de tres ejes con salida digital y un led tricolor, aparte de un par de sensores táctiles capacitivos.
- **FRDM-K64F:** Es el tope de la gama, con un procesador ARM Cortex M4F a 120MHz, con 1MB de flash para programas y 256Kb de sram. La placa incluye además de los periféricos de la anterior, conectores usb, Ethernet, y para tarjeta micro-sd.

Ambos sistemas están soportados por la extendida plataforma de desarrollo online MBED [9]. Esta herramienta permite una rápida transición entre familias de microcontroladores incluso de distintos fabricantes (NXP, Renesas, ST, Nordic...) con un mismo interfaz de programación online, que no requiere de instalación y funciona en prácticamente cualquier plataforma haciéndola muy cómoda para su uso en el aula[10].

Aparte de esta posibilidad, el fabricante ofrece un completo kit de desarrollo, formado por un IDE y un DSK con librerías y ejemplos de programación de los diferentes periféricos, que permite la depuración en tiempo real.

Otra característica interesante de esta plataforma es que la disposición de los pines es compatible con los sistemas de desarrollo de Arduino, siendo posible conectarle a estas placas los *shields* desarrollados para Arduino.

El inconveniente más destacable es que no tengan ningún elemento en la familia en formato DIP. Aunque a estas alturas está bastante en desuso, sigue siendo el estándar a la hora de montar prototipos o de fabricar sistemas con pocos medios.

En la Fig. 2. se pueden ver ambos sistemas, que tienen tamaño muy similar y una disposición idéntica de pines, si bien el K64F dispone además de conector Ethernet.

C. Plataforma Launchpad

La plataforma Launchpad ha sido la respuesta de Texas Instruments a la necesidad de tener un sistema de bajo coste, ampliable y que atienda a la filosofía iniciada por Arduino[11]. El primer elemento de la familia en aparecer fue el Launchpad de la familia MSP430G2, al que han seguido multitud de ellos, con microcontroladores cada vez más potentes y mayor cantidad de periféricos. Destacaremos aquí sólo dos de ellos, para centrarnos en el campo de microcontroladores de bajo coste, y que se muestran en la Fig. 3:

- **MSP-EXP430G2:** Sistema de muy bajo coste, con un microcontrolador de 16 bits y capacidades básicas. Viene a ser el más directo competidor, por segmento, del Arduino UNO.
- **MSP-EXP432P401R:** Incorpora un núcleo ARM Cortex M4F, pero con un juego de periféricos muy similares

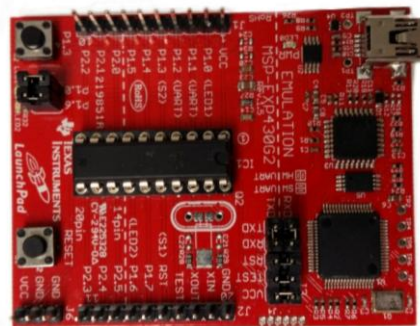
(aunque en mayor número y capacidad) a los de la familia MSP430.

Aparte de estos dos citados, en total la familia cuenta con más de 20 referencias, contando algunas de ellas con microcontroladores de gamas bajas, medias o altas, o con características especiales (wifi, memoria FRAM...)

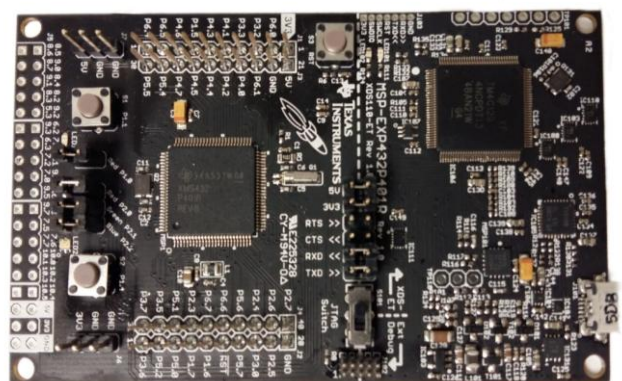
Centrándonos en la primera de ellas, resulta muy atractiva para el propósito deseado, por tener una serie de características distintivas:

- Bajo precio, en relación a sus competidores
- Posibilidad de programar en bajo nivel (lenguaje C) o con un entorno idéntico al de Arduino (llamado Energía)
- Microcontrolador en formato DIP que puede ser extraído de la placa y montado en prototipos independientes
- Gran riqueza de periféricos, incluyendo sensores capacitivos en sus pines.

El sistema se alimenta a través del puerto USB, aunque también se puede alimentar externamente con una fuente de no más de 3.6V, siendo ésta su tensión de alimentación máxima. El microcontrolador es de bajo consumo llegando a consumir por debajo de 1µA en modo de reposo. La placa es suministrada con dos microcontroladores distintos, de la misma familia, el MS430G2452 y el MSP43G2553.



(a)



(b)

Fig. 3. Placas Launchpad (a)MSP-EXP432G2 y (b)MSP-EXP432P401R

Algunas características del microcontrolador MSP430G2553 son:

- CPU Risc de 16 bits a 16MHz
- 16K de memoria Flash para programas y 512 bytes de ram.
- Gran riqueza de periféricos incluyendo entre otros un convertidor A/D de 10 bits y 500ksps, comparador analógico, 2 timers de 16 bits con 3 comparadores cada uno, 2 puertos serie configurables como spi, i2c o uart, 16 pines de e/s.

Por sus características técnicas, este sistema no ofrece mucha capacidad para manejar sistemas de control complejos, pero resulta muy sencillo de programar y tiene una estructura interna muy accesible, en comparación con otros microcontroladores estudiados. Por otro lado, en caso de necesitar más potencia de cálculo o más capacidad de memoria, el resto de los elementos de la familia pueden cumplir con creces con las expectativas. Así, por ejemplo, la otra placa referenciada incorpora un procesador MSP432P401R, que tiene como características básicas:

- Núcleo ARM Cortex-M4F, a 48MHz
- 256Kb de flash para programas, y 64Kb de Ram.
- Periféricos de alta resolución, como convertidor A/D de 14 bits y 1MSPS o timers de 32 bits
- Modos de bajo consumo tanto en funcionamiento como en reposo.

Un aspecto a destacar de la plataforma Launchpad es la existencia, al igual que en el caso de Arduino, de una gran cantidad de hardware específico diseñado para ser compatible con estas placas, que permiten una rápida evaluación tanto de periféricos externos como de las capacidades del microcontrolador elegido, que en este caso reciben el nombre de Boosterpacks [12]. Al igual que en el caso de Arduino, normalmente todos estos sistemas están disponibles con licencias Creative Commons o similares, que permiten disponer de los archivos de diseño del hardware, permitiendo replicarlos o adaptarlos a necesidades específicas.

Todos los microcontroladores de Texas Instruments pueden ser programados usando el entorno de desarrollo suministrado por el fabricante, Code Composer Studio, que permite entre otras cosas la programación, depuración y ejecución paso a paso. Además, para los microcontroladores de baja gama, entre los que se encuentra el montado en la plataforma Launchpad, se puede usar una herramienta de generación de código de inicialización de periféricos, llamada Grace, que facilita mucho el arranque con estos sistemas.

Además, se ha desarrollado de manera paralela otro entorno de desarrollo, basado en la misma filosofía que Arduino y compatible en código con éste, llamado Energia[13], que puede ser de interés para la docencia de asignaturas más centradas en el uso del sistema que en la comprensión del mismo internamente.

IV. COMPARACIÓN

Una vez presentadas las distintas plataformas, cabría hacer una serie de consideraciones globales, tras lo cual se presentará una tabla con una serie de ítems para comparar las distintas plataformas. De ello surgirá la selección hecha para las asignaturas comentadas en la introducción.

Es justo reconocer que no es posible hacer la comparación de manera totalmente objetiva, aunque se intentará en lo posible ceñirse a aquellos aspectos cuantificables.

De manera general, se observa que las plataformas de Texas Instruments, aunque han subido de precio en el último año, son las más económicas de las tres vistas, y en general son probablemente las más económicas del mercado, dentro de los grandes fabricantes de semiconductores. Por otro lado, no cabe duda que Arduino, al ser la primera en conseguir establecerse como estándar en este segmento, dispone de la mayor base de programas y desarrollos en la web. También cabe destacar que la plataforma Freedom ofrece la mayor potencia de cálculo y conectividad de las tres analizadas, siendo sistemas de una gran versatilidad, aunque no dispongan de microcontroladores en formato DIP.

Las tres soluciones analizadas pueden ser programadas al menos de dos maneras distintas: una de manera más o menos fácil, con un lenguaje de alto nivel y librerías de ayuda, y otra usando el compilador propio y descendiendo al nivel de programación de registros internos.

En la Tabla I se resumen los aspectos relativos al hardware que resultan de mayor interés a la hora de realizar la selección, como son:

- Precio: se incluye el precio oficial en la página del fabricante. No obstante, se pueden encontrar muchas veces otros precios más ventajosos, dependiendo de los distribuidores.
- Capacidad del microcontrolador. Se detallan las características básicas, y se concede una puntuación global de su capacidad de cálculo
- Formato DIP: se detalla si el microcontrolador incorporado está en formato DIP
- Periféricos internos y número de pines accesibles
- Existencia de placas conectables
- Extras incluidos en las placas

Se puede comprobar que las placas de la plataforma Launchpad son las más económicas de todas, aunque la primera de ellas es también la más modesta en cuanto a recursos hardware disponibles. También resulta destacable la gran cantidad de extras de las placas de NXP, disponiendo de varios periféricos en el propio sistema de desarrollo. Todas ellas permiten el uso de hardware adicional. En cuanto a las capacidades de cálculo de los microcontroladores implementados, es llamativo que Arduino es la única que usa sistemas de 8 bits, estando las otras centradas en sistemas de 16 ó 32 bits. Es justo reseñar, sin embargo, que Arduino tiene también un miembro de la familia que incorpora un núcleo ARM cortex M0+, de 32 bits.

TABLE I. RESUMEN DE CARACTERÍSTICAS HARDWARE

Características principales	Plataforma Arduino		Plataforma Freedom		Plataforma Launchpad	
	Arduino UNO	Arduino MEGA	Freedom KL25Z	Freedom K64F	Launchpad EXP430G2	Launchpad EXP432P401R
Precio	21,94\$	38,40\$	15\$	35\$	9,99\$	12,99\$
Microcontrolador	ATMEGA328P	ATMEGA2560	KL25Z	K63F	MSP430G2553	MSP432P401R
Núcleo	AVR	AVR	ARM Cortex-M0+	ARM Cortex-M4F	MSP430	Cortex-M4F
Resolución-Vel.	8bit-16MHz	8bit-16MHz	32bit-48MHz	32bit-120MHz	16bit-16MHz	32bit-48MHz
Flash/ram	32k/2k	256k/8k	128k/16K	1M/256K	16k/0.5k	256k/64k
Pines totales	32	86	64	64	20	78
ADC	6	16	6	6S.E. + 4 Diff.	8	24
E/S digital	14	54	48	40	16	69
PWM	6		24	12	7	24
Formato	DIP	SMD	SMD	SMD	DIP	SMD
Ampliaciones	Shields	Shields	Shields	Shields	Boosterpacks	Boosterpacks
Extras	1 led	1 led	Acelerómetro Led tricolor Slider capacitivo usb	Acelerómetro/ magnetómetro Led tricolor Usb Ethernet	2 leds 1 botón	Led tricolor 2 botones

De todas ellas, sólo la placa Arduino UNO y la placa Launchpad EXP430G2 presentan microcontroladores en formato DIP, siendo por tanto las únicas opciones si deseamos realizar después prototipos hardware que usen estos microcontroladores.

Otro aspecto que debería ser tenido en cuenta es la capacidad de depurar los programas paso a paso y la facilidad para acceder a los registros de configuración de cada uno de los periféricos internos de los microcontroladores. En este sentido, tanto la plataforma Arduino como Mbed presentan debilidades, dado que no permiten una cómoda depuración de los programas hechos. Ciertamente, ambas tienen sistemas de desarrollo de *bajo nivel*, que permiten programar en C los microcontroladores, (atmel Studio y Kinetis Design Studio respectivamente), aunque en ese caso se pierde el atractivo inicial de ambas soluciones, en cuanto a la portabilidad de software y simplicidad, que son las grandes bazas de estas dos posibilidades.

V. ESTUDIO DE UN CASO PARTICULAR: *BLINKY*

Una manera de poder comparar las diferentes plataformas puede ser ejecutar el mismo código en todas ellas, para ver el desempeño de cada una. Como ejemplo, se ha tomado el que suele ser el primer programa que se desarrolla en un microcontrolador: el parpadeo de un led. Para comprobar las características de los sistemas de desarrollo al máximo, se va a probar la máxima frecuencia a la que dicho parpadeo se puede ejecutar.

En la Fig. 4 se muestran los códigos en los distintos sistemas de desarrollo. Se puede comprobar que son muy similares en forma pero claramente distintos en filosofía de programación. Mientras que el programa para Arduino usa funciones de alto nivel, el programa mbed usa clases, y el programa para el Launchpad está escrito en C, accediendo directamente al registro de salida del puerto.

A efectos didácticos, la tercera opción presenta la ventaja de tener un contacto directo con los registros de configuración del microcontrolador, aunque el código resultante sea menos general.

Resulta interesante también comprobar el aprovechamiento que cada plataforma hace de los recursos presentes en el microcontrolador, o el nivel de optimización del código generado. Para ello, se midió con el osciloscopio la salida del pin conmutando a máxima frecuencia, obteniendo resultados bastante dispares, como se muestran en la Fig. 5.

Lo primero que se observa es que, en el caso de usar la plataforma Arduino hay momentos en que el microcontrolador no responde, o está atendiendo a tareas relativas al sistema, produciendo una frecuencia no continua. Esto podría ser crítico si se intentase usar en un sistema en tiempo real, aunque claramente no es la aplicación para la que está diseñado

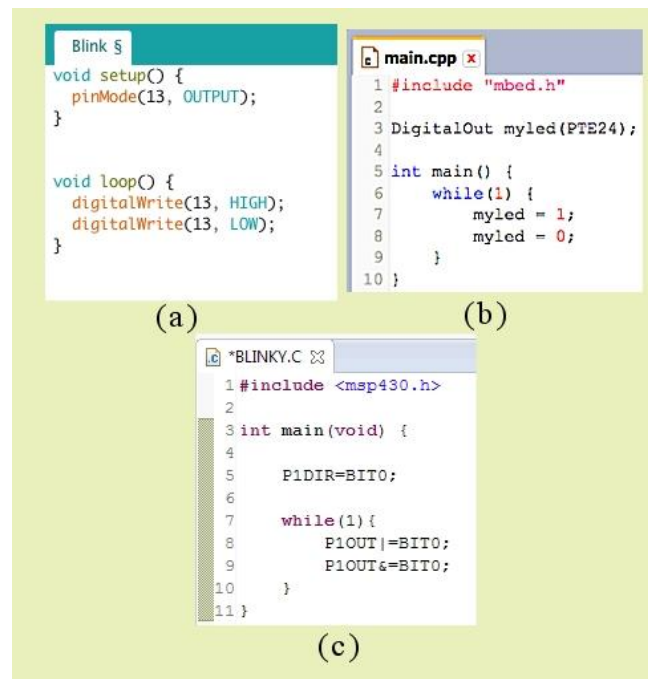
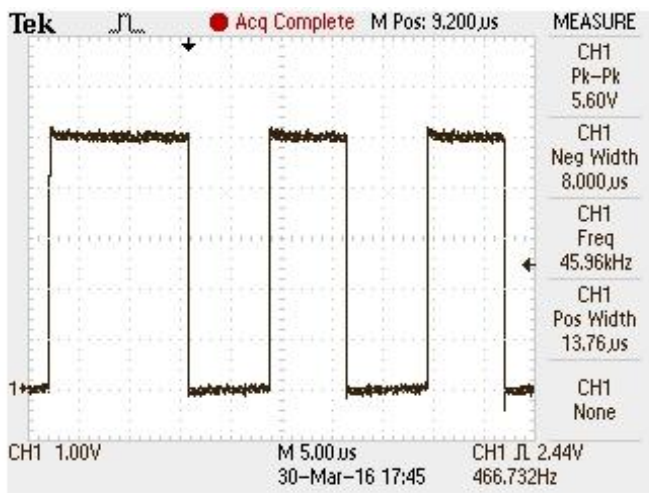
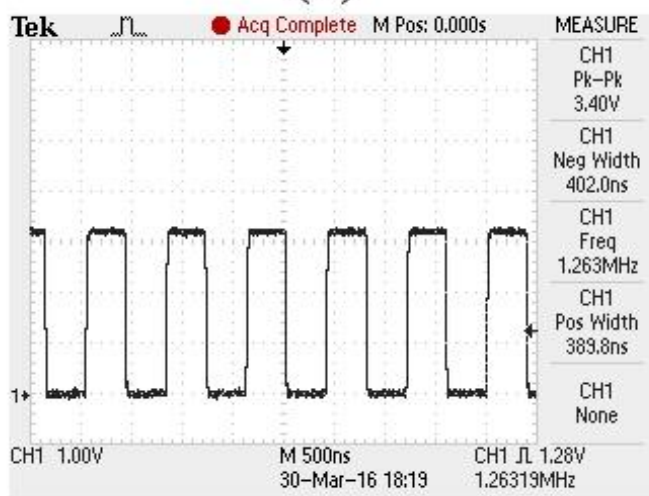


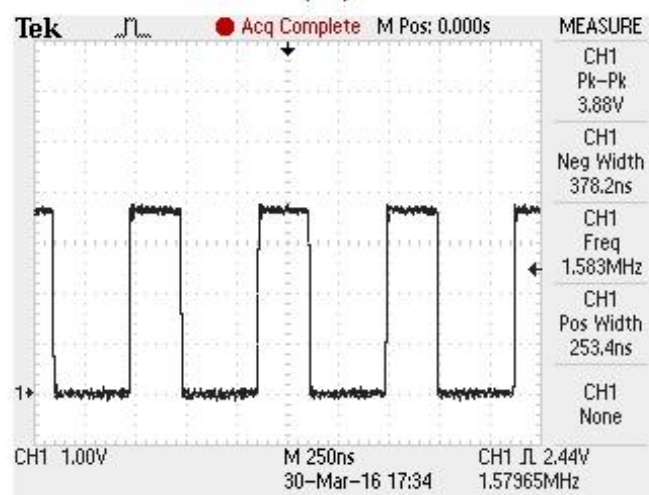
Fig. 4. Códigos para el programa 'blink': (a) para Arduino, (b) para la placa K64F en Mbed y (c) para el MSP430G2553



(a)



(b)



(c)

Fig. 5. Códigos para el programa 'blinky': (a) para Arduino, (b) para la placa K64F en Mbed y (c) para el MSP430G2553

Por otro lado, el tiempo mínimo para conmutar un pin vienen a ser 8µs, lo que equivale a 128 ciclos de reloj a 16MHz.

En el caso de usar la plataforma Mbed, el resultado es mejor, logrando en este caso un ciclo simétrico y continuo en el tiempo, lo que demuestra que el código no es interrumpido por labores de mantenimiento de la plataforma. La frecuencia obtenida es, no obstante, algo baja, dado que el microcontrolador tiene un reloj de 120MHz, por lo que el tiempo medio viene a ser de 46 ciclos de reloj.

Por último, en el caso de la plataforma Launchpad, de nuevo la frecuencia que se obtiene es constante, sin pausas debido a interrupciones no controladas, y el valor en ciclos de reloj es el mejor de los tres casos: 4 ciclos de reloj de 16MHz. Este resultado no es sorprendente, dado que la plataforma Launchpad es la única que está siendo programada en C, con un compilador nativo. De esta manera es sencillo mostrar las ventajas de una programación adaptada al microcontrolador usado en vez de una programación genérica, aunque lleve aparejado un mayor trabajo a la hora de aprender el manejo concreto del microcontrolador a usar.

VI. CONCLUSIÓN

A la vista de las comparaciones hechas, el sistema elegido para las clases de las asignaturas en cuestión fue el Launchpad MSP-EXP430G2, atendiendo fundamentalmente al precio de los sistemas, la facilidad para programar y depurar en bajo nivel, lo que resulta muy atractivo cuando se está explicando una arquitectura nueva, y la posibilidad de expandir luego los sistemas, con un mínimo esfuerzo, contando con elementos compatibles de gran potencia y versatilidad.

Usando esta plataforma, se han diseñado en total 4 asignaturas de grados distintos, y una del Master de Ingeniería Industrial. Para las asignaturas en las que se utiliza como herramienta, se ha programado en Energía, haciendo más hincapié en el sistema completo que en el microcontrolador

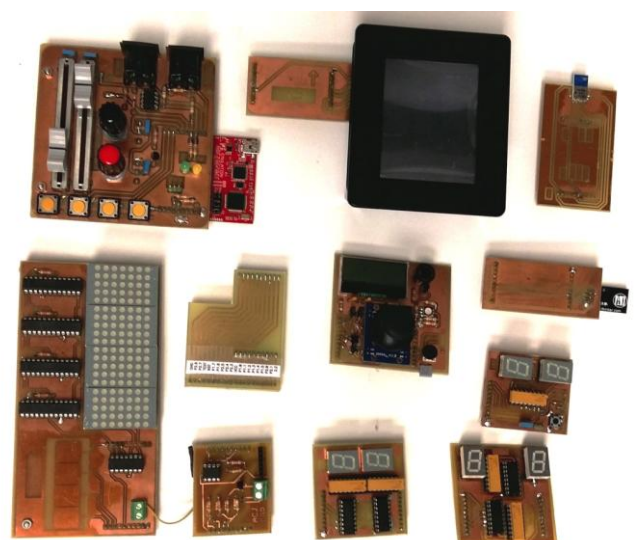


Fig. 6. Algunos de los Boosterpacks desarrollados

concreto. Para aquellas otras en las que el estudio del microcontrolador era importante, se ha programado con Code Composer Studio, pudiendo analizar el funcionamiento del sistema paso a paso, viendo los registros internos o las variables de los programas en tiempo real.

Igualmente se han diseñado más de 10 Boosterpacks propios (Fig. 6), para el manejo de distintos periféricos externos como sistemas de comunicación por radio o pantallas táctiles, y para la realización de pruebas de los periféricos internos como los puertos spi, i2c, etc. Para todos ellos se han desarrollado códigos de ejemplo, disponibles para los alumnos.

REFERENCIAS

- [1] <https://www.arduino.cc/> [consultado 10/02/2016]
- [2] <http://www.ti.com/ww/en/launchpad/launchpads-msp.html#tabs> [consultado 10/02/2016]
- [3] <http://www.nxp.com/products/software-and-tools/hardware-development-tools/freedom-development-boards:FREDEVPLA> [consultado 10/02/2016]
- [4] <https://www.arduino.cc/en/Tutorial/BuiltInExamples> [consultado 10/02/2016]
- [5] <http://playground.arduino.cc/Main/SimilarBoards#goShie> [consultado 10/02/2016]
- [6] L. M. Herger and M. Bodarky, "Engaging students with open source technologies and Arduino," Integrated STEM Education Conference (ISEC), 2015 IEEE, Princeton, NJ, 2015, pp. 27-32.
- [7] <https://processing.org/> [consultado 10/02/2016]
- [8] Y. Torroja, A. López, J. Portilla and T. Riesgo, "A serial port based debugging tool to improve learning with arduino," Design of Circuits and Integrated Systems (DCIS), 2015 Conference on, Estoril, 2015, pp. 1-4.
- [9] <https://developer.mbed.org/platforms/> [consultado 10/02/2016]
- [10] M. M. Macías, J. E. Agudo, C. J. G. Orellana, H. M. G. Velasco and A. G. Manso, "The "mbed" platform for teaching electronics applied to product design," Tecnologías Aplicadas a la Enseñanza de la Electrónica (Technologies Applied to Electronics Teaching) (TAEE), 2014 XI, Bilbao, 2014, pp. 1-6.
- [11] A. fernandez, D. Dang, "Getting Started with the MSP430 Launchpad", ed. Elsevier, 2013, ISBN: 978-0-12-411588-0.
- [12] <http://www.ti.com/ww/en/launchpad/boosterpacks.html> [consultado 10/02/2016]
- [13] <http://energia.nu/> [consultado 10/02/2016]