

# Estudio de Investigación en Educación sobre Microcontroladores

Dimosthenis E. Bolanakis

**Resumen**—Los Microcontroladores constituyen una tecnología popular de nuestra era. Mientras que algunas décadas atrás parecía extremadamente difícil enseñar a personas que no eran ingenieros eléctricos como programar un sistema computacional de un solo chip, hoy en día, diseñadores sin experiencia comparten prototipos basados en microcontroladores en internet. El surgimiento de la cultura actual de hacerlo tú mismo (DIY) ha introducido la permanencia de esta tecnología e hizo que el desarrollo de los sistemas informáticos integrados sea más fácil que nunca. Esta comunicación identifica dos eras fundamentales en el avance de la programación de microcontroladores y el desarrollo de aplicaciones, y distingue el "pasado" con las tendencias de "actualidad" y los riesgos consecuentes en la educación. Luego, se presta atención a la clasificación y el progreso en la investigación educativa durante un período de veinte años, junto con una predicción de la dirección prospectiva en la educación de microcontroladores en la próxima década. En reconocimiento de la creciente naturaleza interdisciplinaria de la tecnología de microcontroladores, la información resumida en este documento podría servir como una guía de referencia para estudios de investigación adicionales en diversas disciplinas.

**Índice de Términos**— Educación de Ciencias de la Computación, Educación de Ingeniería Eléctrica, Educación de Ingeniería Electrónica, Educación de Ingeniería, microcomputadores, microcontroladores.

## I. INTRODUCCIÓN

LA tecnología de Microcontroladores pertenece a la categoría de sistemas informáticos integrados que siguen las reglas del método de programación convencional (secuencial) y no ejecutan un *sistema operativo (SO)*. El último atributo diferencia a los microcontroladores de la tecnología aliada de microprocesadores. El efecto de "hacerlo uno mismo" en la tecnología de microprocesadores ha dado forma a las computadoras de una sola placa de hoy (como la placa Raspberry Pi), donde el sistema operativo generalmente se carga en una tarjeta *digital segura (SD) integrada*. Si bien la programación y el desarrollo de la aplicación de una computadora de una sola placa se pueden realizar de manera similar a la de las computadoras personales (PC), el procedimiento correspondiente para un sistema basado en microcontroladores conlleva desafíos particulares. Los desafíos están de acuerdo con el control de hardware de bajo nivel que el sistema operativo oculta regularmente en una PC o computadora de una sola placa. Debido a esta diferenciación

en el desarrollo de aplicaciones con microcontroladores, junto con la posibilidad de implementar abundantes aplicaciones fascinantes, la formación de microcontroladores ha ganado terreno últimamente y continúa atrayendo cada vez más atención.

Esta comunicación se enfoca en los desafíos de investigación de la tecnología de microcontroladores e identifica dos eras fundamentales en el avance de la programación de microcontroladores y el desarrollo de aplicaciones, a la vez que distingue el "en aquel entonces" con las tendencias "actuales" y los riesgos consecuentes en la educación. Luego, se presta atención a la clasificación y el progreso en la investigación educativa durante un período de veinte años, junto con una predicción de la posible dirección en la educación de microcontroladores en la próxima década. En reconocimiento de la naturaleza interdisciplinaria de la tecnología de microcontroladores, la información resumida en este documento podría servir como guía de referencia para estudios de investigación adicionales en diversas disciplinas y diversos niveles de educación.

## II. EL AVANCE DE LA PROGRAMACIÓN DE MICROCONTROLADORES Y DESARROLLO DE APLICACIONES

El avance de la tecnología y la programación de los microcontroladores se puede separar en dos eras principales, es decir, a) la era del desarrollo de ciclo largo y la era del desarrollo de ciclo corto. La diferenciación recomendada está de acuerdo con el tiempo necesario para aprender, programar y desarrollar una aplicación / sistema basado en microcontroladores, mientras que el paso de la primera a la última era fue el resultado del avance del software y del dominio del hardware. Para aclarar el progreso significativo de ambos aspectos del avance, es aconsejable hacer una referencia a la estructura interna básica y el funcionamiento de un dispositivo de microcontrolador.

Una unidad de microcontrolador constituye un circuito digital síncrono que incorpora una unidad central de procesamiento (CPU) y una memoria para contener el código de firmware (es decir, la memoria del programa), así como los datos generados durante la ejecución del código (es decir, la memoria de datos). La tecnología de memoria anterior que se usó para mantener el código de firmware del microcontrolador fue una memoria de solo lectura programable y borrable (EPROM), donde el borrado de los datos internos se realizó exponiendo el dispositivo del microcontrolador a luz

ultravioleta (UV) y, por lo tanto, este tipo de memoria también se conocía como UVPROM. El último tipo de memoria fue una de las causas principales del proceso de desarrollo de ciclo largo en la era anterior de la programación de microcontroladores, ya que el borrado de la memoria del microcontrolador requería típicamente 30 minutos (lo que resultaba en una situación de prueba de la funcionalidad de un firmware en desarrollo, 2 a 3 veces por hora).

Debido al largo tiempo necesario para reprogramar el microcontrolador, el diseñador tuvo que hacer un esfuerzo considerable en las simulaciones antes de cargar el firmware actualizado en el dispositivo (para verificar su funcionalidad). La simulación soportada por el método de programación estándar del microcontrolador de esa era, que estaba en línea con un bajo nivel de programación; Es decir, la programación en lenguaje ensamblador. La simulación permitió la inspección de los registros internos del microcontrolador en cada ejecución mnemónica de ensamblaje.

Debido a las numerosas ventanas separadas y la afluencia de información mantenida por un programa simulador al mismo tiempo, varios esfuerzos de investigación de la era anterior se orientaron a herramientas de simulación diseñadas a medida con fines educativos [1]. Otros instructores estaban orientados a la implementación de plataformas educativas diseñadas a medida, ya que el largo proceso para construir una aplicación de microcontrolador en el tablero, a menudo centraba la atención de los estudiantes lejos del propósito del curso [2].

En ambas direcciones, el propósito del curso fue centrar la atención de los estudiantes en las técnicas de programación de ensamblaje no estructuradas, y difíciles de seguir, para adquirir el control total sobre las operaciones de la máquina. Las metodologías de enseñanza hacia la paralelización entre la programación de bajo y alto nivel (construida alrededor de un conjunto de palabras clave y símbolos familiares para los humanos), también fueron un estudio de caso de esa era [3]. Por lo tanto, se puede concluir que los riesgos "en aquel entonces" en la educación eran principalmente una preocupación del método de programación del microcontrolador (es decir, el aprendizaje en lenguaje ensamblador). Además, las herramientas de desarrollo disponibles para los microcontroladores no eran prácticas, eran inflexibles y tenían un costo considerable y, por lo tanto, afectaban negativamente el proceso de aprendizaje del lenguaje ensamblador.

#### A. Avances relacionados al hardware

Una reducción significativa del tiempo requerido entre actualizaciones sucesivas en el firmware del microcontrolador, resultó del avance del hardware de la tecnología de memoria. El tipo de memoria Flash de los dispositivos de microcontroladores de hoy, junto con la capacidad de programación en el sistema (PIS), hace posible la actualización de un firmware en segundos (seg), sin necesidad de mover el dispositivo entre el sistema de la placa principal y un programador independiente. Las placas de desarrollo modernas están precargadas con una parte separada del código

en la memoria del microcontrolador, que permite cargar el nuevo firmware al dispositivo del microcontrolador. Además, el puerto RS-232 de las computadoras personales ha sido reemplazado por el puerto USB familiar que proporciona una rápida reprogramación del microcontrolador, así como la posibilidad de suministrar 5 V de alimentación a la placa.

Además, el avance del hardware del paquete de chips desde el paquete dual en línea (DIP) hasta la tecnología de montaje en superficie (SMT) está fuertemente asociado con la revolución del diseño y la fabricación de la placa de circuito impreso (PCB), y ha sido líder gradualmente a la disponibilidad de la colección moderna y lista para usar de sistemas de placas de microcontroladores. Las modernas herramientas de desarrollo de hardware para microcontroladores (como la conocida placa Arduino Uno) se entregan con un código de cargador de arranque y también se alimentan, conectan y programan a través de un conector USB común. Tales sistemas de placa incorporan pasadores de cabecera que se colocan en una posición específica y admiten una gran cantidad de tarjetas intermedias. Las herramientas de hardware de bajo costo de hoy, junto con las bibliotecas de software listas para usar disponibles para los usuarios, hacen posible la implementación de sistemas avanzados de microcontroladores por parte de usuarios inexpertos en cuestión de minutos; Esbozando así la cultura del bricolaje de hoy.

#### B. Avances relacionados al software

A pesar del avance en el hardware de la tecnología de microcontroladores, también se han logrado avances significativos en los métodos de programación. El nivel de ensamblaje se ha reemplazado con un nivel de programación más alto, donde el código C incorporado contiene el método de programación principal y dominante del microcontrolador, en la actualidad. Otros métodos de programación modernos y populares incluyen, entre otros, Arduino, MicroPython y Scratch for Arduino (S4A), que aceleran el proceso de "inicio con los microcontroladores".

Si bien las herramientas de desarrollo modernas de hardware para microcontroladores han madurado considerablemente, las diversas herramientas de desarrollo de software dan lugar a una próspera Torre de Babel, cuando se trata de la educación sobre microcontroladores. Incluso el ejemplo más simple de un diodo emisor de luz (LED) parpadeante con un retardo de 1 segundo entre los estados "encendido" y "apagado", revela una inconsistencia en las reglas de sintaxis entre diversos compiladores C. Vale la pena señalar que los desarrolladores de código que usan programación de bajo nivel estaban luchando para hacer un pasaje entre diferentes lenguajes ensambladores, y el hecho de que este desafío aún existe en los métodos de programación modernos para microcontroladores [4], es notable en un sentido negativo.

### III. ESTUDIOS DE INVESTIGACIÓN Y PROGRESO DE LA EDUCACIÓN EN MICROCONTROLADORES EN LOS ÚLTIMOS VEINTE AÑOS

Esta sección intenta una clasificación de los estudios de

investigación y también explora el progreso en la educación de microcontroladores durante un período de veinte años, es decir, de 1998 a 2017. Para respaldar la consistencia en la clasificación, la información obtenida exclusivamente de la base de datos IEEE Xplore, utilizando el Búsqueda avanzada de frases 'educación de microcontroladores' o 'microcontroladores de enseñanza' dentro de los términos del índice. Un examen preliminar de los 476 registros devueltos mostró que solo 170 entradas (aproximadamente el 35%) dirigen las sesiones de tutoría hacia el aprendizaje del microcontrolador. El resto de las entradas aprovechan el dispositivo del microcontrolador como medio para integrar una aplicación, de modo que los estudiantes obtengan una visión integral de un tema educativo en particular (que no sea el aprendizaje del microcontrolador). Aquí se señala que aproximadamente el 10% de los artículos excluidos de un análisis más detallado, ya que fueron marcados por IEEE por estar en violación de los Principios de publicación, o fueron irrelevantes (por ejemplo, en referencia a la enseñanza de un modelo de microprocesador).

#### A. Investigación educativa en aprender programación de microcontroladores y desarrollo relevante de aplicaciones

Una examinación exhaustiva de los 170 documentos educativos que se aplican en esta área de estudio reveló 5 métodos de investigación diferentes, los últimos 20 años, donde uno de ellos aparece solo los últimos 10 años (es decir, de 2008 a 2017). El gráfico circular de la Fig. 1a) representa los cuatro métodos de investigación (comunes en los últimos 20 años), así como el porcentaje dentro de cada categoría durante el período 1998-2007. Los 5 métodos del período 2008-2017 se dan en la Fig. 1b), donde el método exclusivo (es decir, la migración tecnológica) de ese período se separa del pastel para proporcionar una visualización clara. Se debe tener en cuenta que los documentos que incorporan dos (o más) métodos de investigación (por ejemplo, un enfoque de enseñanza innovador y una herramienta de enseñanza de diseño personalizado) se consideraron como parte de una sola categoría, de acuerdo con el mayor peso de contribución del Método de investigación correspondiente.

La Tabla I presenta el número de artículos publicados de cada método de investigación (en cada período), mientras que el gráfico de barras de la Fig. 1c) muestra el número de artículos publicados para cada año (2018 también se incluye en el gráfico). Cabe señalar que hay un aumento del 430% en los artículos publicados en educación sobre microcontroladores en el período 2008-2017 (en comparación con el período 1998-2007). A continuación, se presenta una descripción detallada del tipo de artículos que se aplican a cada método de investigación en particular.

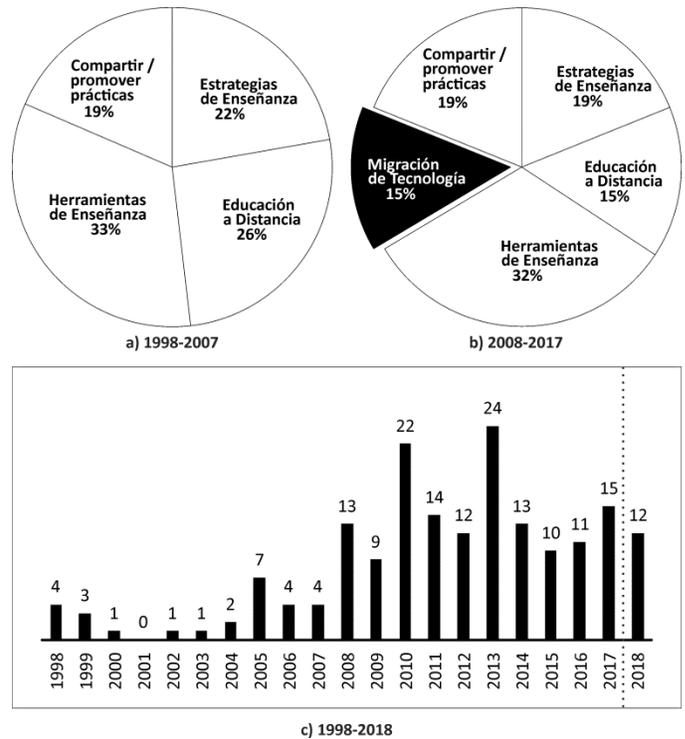


Fig. 1. Métodos de Investigación (a,b) y documentos (c) en enseñanza de microcontroladores.

#### 1) Estrategias de Enseñanza

Los artículos asociados a esta categoría se aplican a un enfoque / estrategia de enseñanza tal como lo revelan los ejemplos indicativos que se presentan a continuación. Tuma et al. [5] deja que los estudiantes tengan una mala experiencia de programación antes de volver a comenzar con el enfoque adecuado. Kramer et al. [6] por otro lado, comienza la tutoría desde el nivel de un sistema operativo y avanza gradualmente hacia el acceso directo al hardware. Los instructores en [7] discuten sobre los arreglos que podrían atraer a más alumnas en los cursos de programación de microcontroladores, mientras que las estrategias de enseñanza en la programación en C incorporada, con respecto a un código fuente generado que es fácilmente adaptable por diversos compiladores y / o modelos de procesadores, son dado en [8].

Debido a que la unidad de microcontrolador constituye el "cerebro" de varias aplicaciones diferentes, aproximadamente la mitad de los artículos en esta categoría abordan un enfoque de aprendizaje basado en proyectos (PBL). Es decir, una pedagogía centrada en el estudiante que dirige a los alumnos a una competencia de aprendizaje más profunda, a través de la exploración activa de un problema de diseño importante y realista [9]. Por ejemplo, el diseño progresivo de los estudiantes en [10] motiva sus procesos de aprendizaje a través de un proyecto basado en microcontroladores, y los ayuda a desarrollar las habilidades requeridas por una futura carrera profesional. Un concurso de carreras con robots de seguimiento de líneas (con el apoyo de la sucursal local de Microchip Corporation) motiva a los estudiantes a aprender todas las habilidades que se incluyen en el laboratorio de diseño de sistemas integrados [11]. Además, una gran cantidad de enfoques de PBL intentan capacitar a los estudiantes para el

desarrollo de sistemas de aplicación particulares, así como para la utilización de tecnologías modernas; por ejemplo, los sistemas de control de supervisión y adquisición de datos (SCADA) [12], las tecnologías de detección táctil [13], etc.

Vale la pena mencionar que la gran cantidad de documentos que se aplican al enfoque de PBL aparece en el período 2008-2017. Este alto porcentaje está de acuerdo con el avance de la tecnología de microcontroladores junto con la prevalencia de un método de programación de alto nivel para microcontroladores en la era del desarrollo de ciclo corto. La organización de un curso basado en PBL en microcontroladores no fue trivial en el desarrollo de ciclo largo. Las herramientas de hardware poco prácticas, así como las dificultades en el aprendizaje de la programación en lenguaje ensamblador, se oponían a la organización de la tutoría en un proyecto grande (y exigente) basado en microcontroladores.

## 2) Educación a distancia

La mayoría de los artículos en esta categoría se aplican a experimentos o laboratorios a los que se accede de forma remota, que se observan en la literatura en una base regular (aproximadamente) desde 2005. Por ejemplo, Gomes y Costa [14] proponen una configuración basada en un microcontrolador 8031 para respaldar un conjunto de experimentos, para aliviar la ocupación física del laboratorio. Zaleski [15] presenta un laboratorio web para la enseñanza (entre otros temas sobre sistemas integrados), desarrollo de software remoto para microcontroladores. El control remoto de un sistema modular basado en un microcontrolador en [16] se proporciona junto con información visual a través de una cámara web. En [17] se describe una descripción general de

las arquitecturas disponibles para laboratorios remotos basados en microcomputadoras, junto con una propuesta para el soporte adicional de una red de sensores inalámbricos (WSN). Los experimentos remotos para capacitación industrial (en mecatrónica y tecnologías alternativas) a través de tres cursos diferentes basados en microcontroladores se dan en [18]. Además, los educadores en [19] señalan otra razón frecuente para implementar experimentos remotos, es decir, la necesidad de proporcionar a los estudiantes acceso a costosos equipos de laboratorio.

Debido a la "distancia" que existe entre los estudiantes y el equipo físico en el método de experimentación de laboratorio remoto, el enfoque de laboratorio virtual también se ha considerado como parte de esta categoría (ya que se opone a los experimentos prácticos, también). En [20-23] se dan ejemplos particulares de este método indirecto de experimentación (por medio de un compromiso mental para completar los experimentos).

El gráfico circular de la Fig. 1 revela una disminución del porcentaje del método de educación a distancia en el período 2008-2017 (es decir, 15%) en comparación con el período 1998-2007 (es decir, 26%). Esta es una consecuencia lógica que surge del mérito de enseñar microcontroladores con experimentos prácticos de hardware. Una de las razones principales para organizar experimentos virtuales o de acceso remoto se basa en una demanda para reducir el costo del aparato de laboratorio. Debido a la disponibilidad de abundantes plataformas de desarrollo de bajo costo para microcontroladores, hoy en día no existe tal motivación con respecto al costo del laboratorio.

TABLA I  
DISTRIBUCIÓN DE MÉTODOS DE INVESTIGACIÓN (1998-2007 Y 2008-2017)

Método de Investigación	Docu- mentos 1998- 2007	Docu- mentos 2008- 2017	Porcentage de Cambio	Temas secundarios de cada método de investigación
Estrategias de Enseñanza	6	27	350%	<ul style="list-style-type: none"> <li>El 46% de los artículos están dedicados al aprendizaje basado en proyectos (PBL).</li> <li>El 36% de los artículos se organizan en torno a enfoques conocidos (es decir, dirigidos a apoyar el trabajo en equipo, el diseño progresivo, el diseño jerárquico del software, la creatividad del diseño, la toma de decisiones, la educación orientada a la acción, la educación innovadora, el aprendizaje activo, las técnicas de cuestionamiento, el aula invertida).</li> <li>El 15% de los artículos se aplica a una estrategia intuitiva / empírica (es decir, permite a los estudiantes tener una mala experiencia de programación antes de volver a comenzar con el enfoque adecuado; examinar las similitudes / diferencias de programación entre arquitecturas de 8 y 32 bits. ; a partir del nivel del sistema operativo (SO) y migrando gradualmente al acceso directo al hardware; desarrollando un código de firmware para soportar la portabilidad entre diferentes plataformas de desarrollo de software y hardware).</li> <li>El 3% de los artículos se dirigen para hacer que el material sea atractivo para una audiencia más amplia (es decir, discutir la posibilidad de aumentar el número de alumnas en el curso)</li> </ul>
Educación a Distancia	7	22	214%	<ul style="list-style-type: none"> <li>El 59% de los artículos se aplican a experimentos de acceso remoto.</li> <li>El 24% de los trabajos se aplican a experimentos virtuales.</li> <li>El 17% de los documentos hace un uso extensivo de la World Wide Web (WWW) para apoyar la educación basada en la web (por ejemplo, abordar un conjunto de diseños de referencia, clips de película, narraciones, etc.).</li> </ul>
Herramientas de Enseñanza	9	46	411%	<ul style="list-style-type: none"> <li>El 75% de los artículos están dedicados a herramientas de enseñanza de hardware.</li> <li>El 25% de los artículos están dedicados a herramientas de enseñanza de software.</li> </ul>
Compartir / promover prácticas.	5	27	440%	<ul style="list-style-type: none"> <li>El 59% de los documentos comparten / promueven una experiencia de enseñanza orientada a prácticas de laboratorio y de "Hazlo tu mismo" (DIY), resolución de problemas del mundo real (así como mini proyectos), tecnologías actualizadas y avanzadas. , Etcétera.</li> <li>El 22% de los documentos comparten / promueven prácticas con un enfoque general en la</li> </ul>

				<p>universidad / currículo (por ejemplo, colaboración entre diferentes universidades, colaboración entre la industria y la universidad y cursos diseñados para satisfacer las necesidades específicas del currículo)</p> <ul style="list-style-type: none"> <li>• El 19% de los documentos comparten / promueven una experiencia en cursos mixtos relacionados con la enseñanza de la arquitectura y organización de microcontroladores (es decir, organizados en torno al desarrollo de aplicaciones basadas en microcontroladores y FGPA / CPLD)</li> </ul>
Migración de Tecnología	0	21	Indefinido	<ul style="list-style-type: none"> <li>• El 62% de los artículos se tratan sobre temas de enseñanza relacionados con la disciplina de la computación (es decir, arquitectura de computadoras y programación de computadoras, con la necesidad de desmitificar el hardware involucrado en los procesos de computación, así como presentar a los estudiantes el desarrollo de aplicaciones con sistemas informáticos integrados).</li> <li>• El 19% de los artículos están dirigidos a educar a los estudiantes en ciencias (por ejemplo, Física), así como en disciplinas de ingeniería distintas a la ingeniería eléctrica / electrónica o informática.</li> <li>• El 19% de los trabajos se aplica a la educación de los estudiantes de secundaria.</li> </ul>
TOTAL	27	143	430%	

### 3) Herramientas de Enseñanza

También hay dos subcategorías disponibles en este grupo. El primer método se aplica a los kits de hardware para apoyar la experiencia práctica, mientras que el último se aplica a las herramientas de software. La superioridad numérica del primer enfoque es claramente mayor que la segunda.

En consecuencia, hay propuestas sobre sistemas de placas diseñados a medida que se aplican a un núcleo particular de microcontroladores, como el Freescale HC11 [24] (donde algunos modelos de esta familia están actualmente soportados por NPX), el Freescale HC08 [25] (actualmente extendido a la familia NPX S08), el AVR ATmega32 [26] (recientemente comprado por Microchip Inc), el INTEL 8051 [27] proporcionado por varios fabricantes de terceros (como Silicon Labs), etc. Otros esfuerzos de investigación en kits de hardware hacen hincapié en la evaluación de núcleos de microcontroladores, ya sea desarrollando kits de hardware universales [28-30] o explotando kits comerciales [31]. Algunas pedagogías se centran en el desarrollo de kits para llevar a casa [32, 33], u ocasionalmente motivan a los estudiantes a comprar sus propios paquetes comerciales para que puedan trabajar con ellos cuando lo deseen [34]. Tableros adicionales diseñados a medida se aplican a la enseñanza de temas específicos; por ejemplo, sistemas integrados móviles [35] e inalámbricos [35, 36], sistemas basados en sensores [37], etc.

Las herramientas de enseñanza de software se aplican regularmente a los simuladores diseñados a medida, que se dirigen para ayudar a los estudiantes a visualizar el funcionamiento interno de un dispositivo de microcontrolador [1, 38] o de varios microcontroladores con coincidencia [39] o arquitectura diferente [40]. Vale la pena señalar que los simuladores prosperaron durante la era de la programación de microcontroladores de bajo nivel, mientras que comenzaron a eliminarse cuando surgieron los métodos de programación de alto nivel. Las proposiciones sobre tecnología educativa en armonía con las tendencias de la era del desarrollo de ciclo corto, apuntan a reducir el esfuerzo del desarrollador. En detalle, Bai et al. [41] proporcionan una generación automatizada del código de inicialización del microcontrolador (basado en la configuración del usuario), mientras que Kato et al. [42], habilita un esquema de

programación visual basado en íconos (apropiado también para estudiantes no especialistas).

De acuerdo con la Tabla I, la superioridad de los documentos dedicados a las herramientas de enseñanza de hardware (es decir, 75%) es bastante clara en comparación con los documentos correspondientes dedicados a las herramientas de enseñanza de software (es decir, 25%). Por un lado, esta ocurrencia se debe al hecho de que el lenguaje ensamblador se ha reemplazado con un nivel más alto de programación y, por lo tanto, se ha eliminado la demanda de simuladores personalizados. Por otro lado, el interés particular de los educadores en las herramientas de enseñanza de hardware revela el requisito de la sociedad de una plataforma educativa estandarizada, que pueda ser utilizada por varias disciplinas diversas y varios niveles de educación. A menudo se requiere un esfuerzo considerable solo para rastrear e identificar (de una gran cantidad de fabricantes que entregan dispositivos electrónicos aliados) el módulo de hardware y / o el chip electrónico adecuados para un experimento personalizado. Además, la incompatibilidad entre diferentes herramientas de hardware para los microcontroladores motiva las proposiciones sobre la disposición de una conducción suave de los procesos de experimentación [31]. Para este fin, vale la pena señalar que esta área de investigación constituye el único método (común en los últimos 20 años) que no ha sido sometido a una disminución considerable.

### 4) Compartir / promover prácticas.

Unos cuantos investigadores comparten y / o promueven su experiencia en la enseñanza de microcontroladores. Esto puede incluir una iniciativa para ofrecer un curso profesional sobre tecnologías actualizadas y avanzadas. Los ejemplos indicativos están en línea con la organización de un curso sobre tecnologías subyacentes de bajo consumo [43] y estándar de la industria [44], o un curso que se aplica a las nuevas culturas del desarrollo de aplicaciones (como la cultura del bricolaje) [45, 46], etc. La investigación aliada en esta categoría está en línea con una promoción de las experiencias educativas que deben abordarse en el currículo [47, 48].

Otros esfuerzos que se encuentran dentro de esta área, comparten una experiencia en un curso mixto relacionado con la enseñanza de la arquitectura y organización de los microcontroladores. El término arquitectura se refiere a las

características del microordenador que son "visibles" para el programador durante el proceso de desarrollo del código. El término organización está asociado al diseño de la estructura interna del microcontrolador. El proceso de diseño generalmente se lleva a cabo con una tecnología de dispositivo lógico programable complejo (CPLD) [49] o Field Programmable Gate Array (FPGA) [50], y se realiza comúnmente en un lenguaje de descripción de hardware (HDL), generalmente VHDL o Verilog. Esos enfoques mixtos en HDL y programación secuencial (por ejemplo, incrustado en C), amplían aún más la perspectiva del alumno sobre los sistemas informáticos integrados.

Debido a la naturaleza experiencial de la educación con microcontroladores, compartir / promover prácticas puede considerarse el área más típica de la investigación educativa con muchos aspectos diferentes. Los instructores pueden compartir una experiencia educativa derivada de a) la aplicación de una o más pedagogías en el sistema de tutoría, b) la aplicación de tecnologías de hardware particulares en los procesos experimentales, c) el impacto del lenguaje de programación empleado (por ejemplo, Python, C, Arduino) a los resultados obtenidos por los alumnos, y mucho más.

#### 5) *Migración de Tecnología*

Desde hace muchos años, la tecnología del microcontrolador migra periódicamente para satisfacer las necesidades de la educación en disciplinas de ingeniería distintas a la Ingeniería electrónica (por ejemplo, Bioingeniería [51], Ingeniería mecánica [51, 52], Ingeniería aeroespacial [53], etc.) Sin embargo, La aparición de la cultura DIY actual y la consiguiente formación de la industria en la dirección del desarrollo de hardware listo para usar para microcontroladores, también ha introducido la omnipresencia de esta tecnología a los no ingenieros. Los cursos basados en hardware alrededor de unidades de microcontroladores se ofrecen hoy en día a estudiantes de ciencias de la computación (en programas de estudio orientados al software) [54-56], mientras que los microcontroladores (y / o máquinas similares a microcontroladores [57]) se dirigen a la enseñanza de Arquitectura de Computadores y Cursos introductorios de programación [57-59].

Debido a que los microcontroladores constituyen sistemas informáticos de un solo chip, esta tecnología está fuertemente asociada con los procesos de programación informática. Sin embargo, la relación mutua entre el firmware y el hardware, junto con la intensa participación del diseñador en ambos lados del desarrollo, dejó a los estudiantes orientados hacia el software por mucho tiempo fuera del juego. El progreso en la tecnología de hardware y el efecto de bricolaje en la construcción de sistemas basados en microcontroladores con tableros adicionales en una configuración de apilamiento, permitió la entrada de desarrolladores de software al interesante y desafiante mundo de los microcomputadores. Este hecho se revela por el surgimiento del método de 'Tecnología de migración' (durante los últimos 10 años) y la participación equivalente que se mantiene en el gráfico circular de la Fig. 1. También se evidencia por el incremento (aproximadamente 430%) de los artículos publicados (durante

el mismo lapso) en educación de microcontroladores (Fig. 1).

Según la Tabla I, la mayor cantidad de artículos publicados en esta área (es decir, el 62%) está fuertemente asociada con temas relacionados con la disciplina de la computación. Esta ocurrencia se basa en el progreso de los sistemas informáticos integrados que, hoy en día, se pueden encontrar en casi cualquier lugar, e influyen / dan forma al panorama educativo actual. Los desafíos futuros y la dirección prospectiva en la educación de microcontroladores, la próxima década, se discuten a continuación.

#### IV. PROSPECTOS A FUTURO Y RETOS

La programación es una de las áreas más populares en la disciplina de la computación. Uno de los desafíos recientes de la disciplina es el cambio de la programación tradicional de PC hacia la programación de sistemas informáticos integrados (ECS). El hecho de que los estudiantes de ingeniería de software puedan trabajar en la programación de ECS en su carrera de perspectiva (ya desde la década anterior) ha sido discutido en la literatura [60]. Hoy en día, algunos instructores intentan introducir el desarrollo de software integrado en la educación K12 [61, 62]. Debido al avance tecnológico en los sistemas informáticos integrados, se anticipa un aumento considerable (los próximos diez años) en el número de trabajos de investigación en perspectiva en educación de microcontroladores. Esto probablemente será el resultado de la absorción de la tecnología del microcontrolador en la educación K12, de manera similar, el lenguaje de programación Scratch [63] ha sido absorbido en todos los niveles educativos. Se podría decir que es una demanda del movimiento de creadores de hoy en la educación (es decir, el cambio del conocimiento ya hecho a un aula que involucra a los estudiantes en creatividad, innovación y colaboración a través de la construcción, en lugar de simplemente usar cosas [64] ), donde una gran parte de los fabricantes incorpora tecnología de microcontroladores en diversas áreas de innovación (por ejemplo, en ciencia, arte, música, etc.). Por supuesto, esto no será fácil, ya que la programación del microordenador y el desarrollo de la aplicación difieren mucho del procedimiento correspondiente en una computadora personal convencional, principalmente debido a la "entrada / salida (IO) no estándar que interactúa con el mundo exterior" junto con el hecho de que la tecnología de microcontrolador convencional no ejecuta un sistema operativo (SO).

Vale la pena señalar que los procesos de abstracción son más efectivos y claros cuando se aplican a una computadora personal en lugar de a un microcontrolador sin sistema operativo. Los módulos periféricos adicionales para los microcontroladores de hoy, como el conocido Arduino, se entregan con bibliotecas compartibles que ocultan la configuración compuesta y los procesos de lectura, y proporcionan a los diseñadores una interfaz simplificada para el mundo exterior. Sin embargo, la tecnología de microcontrolador sin sistema operativo no ofrece la posibilidad de reutilizar las bibliotecas prototipo en diversas plataformas de desarrollo de software y hardware [4, 8].

Además, algunos autores han destacado la pérdida de capacidad para diseñar sistemas basados en Arduino por parte de los estudiantes debido a la gran cantidad de información generada por el movimiento de creadores [65]. Martínez-Santos et al. [65] identificó 4 categorías en la plataforma Arduino al intentar resolver un problema de diseño de una manera más eficiente (por ejemplo, menor consumo de energía, mayor tasa de ejecución de instrucciones, etc.). Por lo tanto, se puede concluir que la contribución del movimiento de creadores y de la cultura del bricolaje ciertamente ha traído la revolución e introducido el desarrollo de aplicaciones con microcontroladores a una audiencia más amplia, pero no es una panacea para los problemas educativos. Muchos temas aún deben ser explorados y resueltos para facilitar la absorción de la tecnología del microcontrolador en varias disciplinas, así como los niveles de educación.

## V. CONCLUSIÓN

Esta comunicación identificó dos eras fundamentales en el avance de la programación de microcontroladores y el desarrollo de aplicaciones (es decir, la era de desarrollo de ciclo corto y ciclo largo) y discutió las tendencias y riesgos correspondientes en educación. Luego, una encuesta de investigación en educación de microcontroladores (más de dos décadas) reveló cinco categorías de investigación elementales: (a) Estrategias de enseñanza, (b) Educación a distancia, (c) Herramientas de enseñanza, (d) Prácticas de participación / promoción, y (e) La migración tecnológica (con el enfoque final surgido en la última década).

En lo que respecta al área de investigación (a), el avance de la tecnología junto con la prevalencia de un método de programación de alto nivel para microcontroladores (en la era del desarrollo de ciclo corto), promovió la organización de la tutoría en torno a un aprendizaje basado en proyectos (PBL) (encontrado como la pedagogía dominante en la educación de microcontroladores en la última década).

El mérito de aprender microcontroladores con experimentos prácticos de hardware, junto con la amplia difusión de plataformas de desarrollo de hardware de bajo costo para microcontroladores, causó una disminución (el período 2008-2017) al número de artículos publicados del área de investigación (b).

En consideración al área de investigación (c), el reemplazo del método de programación en lenguaje ensamblador con un nivel más alto de programación redujo el plan de investigación para simuladores diseñados a medida (que constituirían las herramientas fundamentales de enseñanza de software de la era del desarrollo de ciclo largo). Por otro lado, el mayor interés de los educadores en las herramientas de enseñanza de hardware revela el requisito de la sociedad de una plataforma educativa estandarizada para los microcontroladores. Este requisito mantiene el porcentaje de artículos publicados en los períodos 1998-2007 y 2008-2017, que se encuentran dentro de esta área de investigación.

Debido a la naturaleza experiencial de la educación con microcontroladores, el área de investigación (d) puede considerarse como el área más común de la investigación

educativa con muchos aspectos diferentes. Por ejemplo, un intercambio de experiencias puede basarse en la aplicación de una o más pedagogías en el sistema de tutoría, la aplicación de tecnologías de hardware particulares en los procesos experimentales, etc.

El método de investigación (e), presente exclusivamente en los años 2008-2017, revela la maduración de la tecnología de microcontroladores que hace posible su migración a diversas disciplinas científicas y de ingeniería. Debido al progreso en los sistemas informáticos integrados en la actualidad (que se puede encontrar en casi cualquier lugar, y por lo tanto, influyen / dan forma al panorama educativo actual), el 62% de los artículos publicados en esta área están asociados con temas de la disciplina informática. Esta eventualidad está relacionada con la alta probabilidad de que los estudiantes de ingeniería de software trabajen en la programación de sistemas informáticos integrados en su carrera de perspectiva.

La programación es una de las áreas más populares en la disciplina de la computación. El reciente cambio de la programación tradicional de computadoras personales hacia la programación de sistemas informáticos integrados pronostica un aumento considerable (los próximos años) en el número de trabajos de investigación en perspectiva sobre educación de microcontroladores. Este hecho se evidencia en el incremento de aproximadamente 430% de los artículos publicados en general durante el período 2008-2017, así como en la nueva área de investigación (denominada Migración de tecnología) que surgió durante el mismo período. Sin embargo, el mayor interés de los educadores en las herramientas de enseñanza de hardware (a pesar de la abundante disponibilidad de modernas plataformas de capacitación para microcontroladores) revela la limitación actual de un modelo de capacitación estandarizado en microcontroladores, que puede ser utilizado por diversas disciplinas y diversos niveles de educación. .

## REFERENCIAS

- [1] A. del Rio and J.J.R. Andina, "UV151: a simulation tool for teaching/learning the 8051 microcontroller," in *Proc. Frontiers in Education Conference (FIE)*, Kansas City, MO, USA, 2000, pp. F4E:11-16.
- [2] D. E. Bolanakis, E. Glavas, and G. A. Evangelakis, "An integrated microcontroller-based tutoring system for computer architecture laboratory course," *Int. J. Eng. Educ.*, vol. 23, no. 4, pp. 785-798, Jul. 2007.
- [3] D. E. Bolanakis, G. A. Evangelakis, and E. Glavas, "A teaching approach for bridging the gap between low-level and high-level programming using assembly language learning for small microcontrollers," *Comput. Appl. Eng. Educ.*, vol. 19, issue 3, pp. 525-537, Apr. 2009, 10.1002/cae.20333.
- [4] D. E. Bolanakis, "Microcontroller Education: Do It Yourself, Reinvent the Wheel, Code to Learn," in *Synthesis Digital Library of Engineering and Computer Science – Synthesis Lectures on Mechanical Engineering (Collection Nine)*, USA: Morgan & Claypool, 2017, 193 pp.
- [5] T. Tuma, I. Faifar, and J. Puhan, "The principle of vaccination in teaching," in *Proc. Frontiers in Education Conference (FIE)*, San Juan, Puerto Rico, USA, 1999, pp. 11B4:16-20.
- [6] M. Kramer, R. Bener, and P. Brejcha, "Reducing the entrance hurdle in embedded systems engineering courses," in *Proc. IEEE EUROCON*, Lisbon, Portugal, 2011, pp. 1-4.
- [7] P. Brejcha, M. Kramer, and R. Bener, "Increasing the attractiveness of embedded systems courses," in *Proc. IEEE EDUCON*, Marrakech, Morocco, 2012, pp. 1-9.

- [8] A. K. Rachioti, D. E. Bolanakis, and E. Glavas, "Teaching strategies for the development of adaptable (compiler, vendor/processor independent) embedded C code," in *Proc. 15th International Conference on Information Technology Based Higher Education and Training (ITHET)*, Istanbul, Turkey, 2016, pp. 1-7.
- [9] M. Hedley, "An undergraduate microcontroller systems laboratory," *IEEE Trans. Educ.*, vol. 41, no. 4, p. 345, Nov. 1998, 10.1109/13.728278.
- [10] J. W. Bruce, J. C. Harden, and R. B. Reese, "Cooperative and progressive design experience for embedded systems," *IEEE Trans. Educ.*, vol. 47, issue 1, pp. 83-92, Feb. 2004, 10.1109/TE.2003.817618.
- [11] C-S Lee, J-Huei Su, K-E Lin, J-H Chang, and G-H Lin, "A project-based laboratory for learning embedded system design with industry support," *IEEE Trans. Educ.*, vol. 53, issue 2, pp. 173-181, May 2010, 10.1109/TE.2008.2010990.
- [12] A. K. R. Segund, J. A. N. Cocota, R. Q. Hilário, V. de O. Gomide, and D. V. M. Ferreira, "Low cost SCADA system for education," in *Proc. IEEE EDUCON*, Tallinn, Estonia, 2015, pp. 536-542.
- [13] N He, H-w. Huang, and Y. Qian, "Teaching touch sensing technologies through project-based learning," in *Proc. Frontiers in Education Conference (FIE)*, Erie, PA, USA, 2016, pp. 1-7.
- [14] L. Gomes and A. Costa, "Remote laboratory support for an introductory microprocessor course," in *Proc. IEEE International Conference on Microelectronic Systems Education*, Anaheim, CA, USA, 2005, pp. 21-22.
- [15] J. Zalewski, "A comprehensive embedded systems lab for teaching web-based remote software development," in *Proc. IEEE Conference on Software Engineering Education and Training*, Pittsburgh, PA, USA, 2010, pp. 113-120.
- [16] J. V. Martínez, R. M. Bejarano, R. S. Reillo, and R. M. Bolea, "Virtualab, a teaching/learning system for 8 and 32 bits microcontrollers," in *Proc. IEEE EDUCON*, Madrid, Spain, 2010, pp. 1419-1423.
- [17] D. Fudurić, D. Cmk, P. Knežević, and S. Mrvoš, "Assimilation of Wireless Sensor Networks and remote microcomputer laboratories," in *Proc. 34th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2011, pp. 1075-1083.
- [18] O. Dziabenko, A. Rojko, I. Angulo, J. Garcia-Zubia, and P. Orduna, "Training of microcontrollers using remote experiments," in *Proc. 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, Bilbao, Spain, 2012, pp. 1-6.
- [19] H. Mostefaoui and A. Benachenhou, "Design of a remote electronic laboratory," in *Proc. International Conference on Interactive Mobile Communication Technologies and Learning (IMCL)*, Thessaloniki, Greece, 2015, pp. 160-162.
- [20] J. Richardson, N. Adamo-Villani, E. Carpenter, and G. Moore, "Designing and implementing a virtual 3D microcontroller laboratory environment," in *Proc. Frontiers in Education Conference (FIE)*, San Diego, CA, USA, 2006, pp. 1-5.
- [21] W. Xinhuan, Z. Hongwei, G. Qinghua, and Z. Wei, "The construction of single-chip microcomputer virtual experiment platform based on Proteus," in *Proc. 5th International Conference on Computer Science and Education (ICCSE)*, Hefei, China, 2010, pp. 609-611.
- [22] R. Mohanty and A. Routray, "Advanced virtual embedded system laboratory," in *Proc. 2nd Interdisciplinary Engineering Design Education Conference (IEDEC)*, Sanata Clara, CA, USA, 2012, pp. 92-95.
- [23] J. V. V. Sánchez, J. A. M. Zamora, and F. J. Álvarez Garcia, "Virtual lab for digital systems," in *Proc. Technologies Applied to Electronics Teaching (TAE)*, Seville, Spain, 2016, pp. 1-7.
- [24] B.R. Horowitz, R.J. Recltenwald, P.A. Kute, T.G. Cleaver, "Updating an instructional laboratory for a computer interfacing course: a flexible configuration based on the M68HC11 microcontroller," in *Proc. Frontiers in Education Conference (FIE)*, San Juan, Puerto Rico, USA, 1999, pp. 13C6:1-6.
- [25] D. E. Bolanakis, E. Glavas, and G. A. Evangelakis, "A multidisciplinary educational board system for microcontrollers: considerations in design for technically accurate custom-made platforms," in *Proc. 1st International Symposium on Information Technologies and Applications in Education*, Kunming, P. R. China, 2007, pp. 391-395.
- [26] G. Mostafa, "Development of an 8-bit RISC microcontroller learning kit using atmel ATmega32 architecture," in *Proc. International Conference on Advances in Electrical Engineering (ICAEE)*, Dhaka, Bangladesh, 2013, pp. 154-159.
- [27] M. U. Kamaluddin, S. Shahbudin, N. M. Isa, and H. Z. Abidin, "Teaching the Intel 8051 microcontroller with hands-on hardware experiments," in *Proc. 7th International Conference on Engineering Education (ICEED)*, Kanazawa, Japan, 2015, pp. 100-105.
- [28] A. Drumea, P. Svasta, and A. Vasile, "Universal development board for education in embedded systems," in *Proc. International Spring Seminar on Electronics Technology*, Budapest, Hungary, 2008, pp. 575-578.
- [29] R. S. K. Selvakumar, K. H. Ghazali, N. M. K. Nik Yusoff, and H. AbdulAziz, "Learning embedded systems with multi-microcontroller evaluation tool," in *Proc. International Conference on Electrical, Control and Computer Engineering (INECCE)*, Pahang, Malaysia, 2011, pp. 288-292.
- [30] V. Kh. Abdrakhmanov, N. N. Bikbaev, and R. B. Salikhov, "Development of low-cost electronic training boards based on universal microcontroller," in *Proc. 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, Novosibirsk, Russia, 2016, pp. 319-325.
- [31] D. E. Bolanakis, A. K. Rachioti, and E. Glavas, "Nowadays trends in microcontroller education: do we educate engineers or electronic hobbyists? Recommendation on a Multi-platform Method and System for Lab Training Activities," in *Proc. IEEE EDUCON*, Athens, Greece, 2017, pp. 73-77.
- [32] C. Ma, Q. Li, Z. Liu, and Y. Jin, "Low cost AVR microcontroller development kit for undergraduate laboratory and take-home pedagogies," in *Proc. 2nd International Conference on Education Technology and Computer (ICETC)*, Shanghai, China, 2010, pp. V1/35-V1/38.
- [33] J. Sarik and I. Kymissis, "Lab kits using the Arduino prototyping platform," in *Proc. Frontiers in Education Conference (FIE)*, Washington, DC, USA, 2010, pp. T3C:1-5.
- [34] O. H. Graven and J. Björk, "The use of an Arduino pocket lab to increase motivation in Electrical engineering students for programming," in *Proc. International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Bangkok, Thailand, 2016, pp. 239-243.
- [35] K. Kosturik and P. Weissar, "Application of microcontrollers in education based on mobile platform," in *Proc. IEEE EUROCON*, Belgrade, Serbia, 2005, pp. 902-905.
- [36] J-S. Chenard, Z. Zilic, and M. Prokic, "A laboratory setup and teaching methodology for wireless and mobile embedded systems," *IEEE Trans. Educ.*, vol. 51, issue 3, pp. 378-384, Aug. 2008, 10.1109/TE.2008.919690.
- [37] A. Kommu, N. K. Uttarkar, and R. R. Kanchi, "Design and development of sensor-based mini projects for embedded system laboratory using ARM Cortex-M3(LPC1768)," in *Proc. International Conference on Information Communication and Embedded Systems (ICICES)*, Chennai, India, 2014, pp. 1-6.
- [38] G. Malhotra, N. Atri, and S. R. Sarangi, "emuARM: a tool for teaching the ARM assembly language," in *Proc. 2nd International Conference on e-Learning and e-Technologies in Education (ICEEE)*, Lodz, Poland, 2013, pp. 115-120.
- [39] A. del Rio, J.J. Rodriguez-Andina, and A.A. Nogueiras-Melendez, "Learning microcontrollers with a CAI oriented multi-micro simulation environment," *IEEE Trans. Educ.*, vol. 44, issue 2, p. 197, May. 2001, 10.1109/13.925846.
- [40] B. Deaky, N. B. Lupulescu, and D. Ursutiu, "Extended educational use of the microcontroller student learning kit (MCU SLK)," in *Proc. IEEE EDUCON*, Amman, Jordan, 2011, pp. 913-916.
- [41] G. Y. Bai, E. S. Chng, and G. P. Bhanu, "An MCU description methodology for initialization code generation software," in *Proc. International Conference on Parallel and Distributed Systems*, Hsinchu, Taiwan, 2007, pp. 1-7.
- [42] Y. Kato, "Splish: a visual programming environment for Arduino to accelerate physical computing experiences," in *Proc. 8th International Conference on Creating Connecting and Collaborating through Computing (C5)*, La Jolla, CA, USA, 2010, pp. 3-10.
- [43] D. Ursutiu, C. Samoilă, V. Nascov, and M. Moga, "Microcontroller technologies in low power applications," in *Proc. 15th International*

- Conference on Interactive Collaborative Learning (ICL)*, Villach, Austria, 2012, pp. 1-5.
- [44] N. Alaraje, A. Sergeyev, J. Reutter, C. Kief, B. Matar, and D. Hata, "Technology driven university and community college collaboration: faculty training on ARM microcontrollers," in *Proc. Portland International Conference on Management of Engineering and Technology (PICMET)*, Honolulu, HI, USA, 2016, pp. 1681-1687.
- [45] P. J. Sotorrío Ruiz, F. D. Trujillo Aguilera, F. J. Sánchez Pacheco, and A. P. Ruz, "Towards the implementation of a personal learning environment in microprocessors by DIY method," in *Proc. Technologies Applied to Electronics Teaching (TAEE)*, Bilbao, Spain, 2014, pp. 1-7.
- [46] A. A. Galadima, "Arduino as a learning tool", in *Proc. 11<sup>th</sup> International Conference on Electronics, Computer and Computation (ICECCO)*, Abuja, Nigeria, 2014, pp. 1-4.
- [47] M. Bezdek, D. Helvick, R. Mercado, D. Rover, A. Tyagi, and Z. Zhang, "Developing and teaching an integrated series of courses in embedded computer systems," in *Proc. Frontiers in Education Conference (FIE)*, San Diego, CA, USA, 2006, pp. 19-24.
- [48] V. F. de Lucena, J. P. de Queiroz-Neto, I. B. Benchimol, A. P. Mendonca, V. R. da Silva, and M. F. Filho, "Teaching software engineering for embedded systems: an experience report from the manus research and development pole," in *Proc. Frontiers in Education Conference (FIE)*, Milwaukee, WI, USA, 2007, pp. S4C:3-8.
- [49] N. Bencheva, N. Kostadinov, and Y. Ruseva, "Interdisciplinary projects in teaching "Microprocessor Systems," in *Proc. 20th European Association for Education in Electrical and Information Engineering (EAEEIE)*, Valencia, Spain, 2009, pp. 1-4.
- [50] G. Morison and P. Barrie, "An integrated approach to teaching embedded systems early in the curriculum — flip flops to PONG," in *Proc. 5th European DSP Education and Research Conference (EDERC)*, Valencia, Spain, 2012, pp. 169-172.
- [51] S. Su, J. Kerwin, S. Crowe, J. Cavallaro, and G. L. Woods, "Teaching embedded programming to electrical engineers, BioEngineers, and mechanical engineers via the escape platform," in *Proc. 3<sup>rd</sup> Interdisciplinary Engineering Design Education Conference (IEDEC)*, Santa Clara, CA, USA, 2013, pp. 87-92.
- [52] R. Grepl, "Real-Time Control Prototyping in MATLAB/Simulink: Review of tools for research and education in mechatronics," in *Proc. International Conference on Mechatronics (ICM)*, Istanbul, Turkey, 2011, pp. 881-886.
- [53] L. Gil-Sánchez, R. Masot, and M. Alcañiz, "Teaching electronics to Aeronautical Engineering students by developing projects," *IEEE RITA Educ.*, vol. 10, no. 4, pp. 282-289, Nov. 2015, 10.1109/RITA.2015.2486398.
- [54] S. Ristov, M. Stolič, and N. Ackovska, "Awakening curiosity — Hardware education for computer science students," in *Proc. 34th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2011, pp. 1275-1280.
- [55] N. Ackovska and S. Ristov, "Hands-on improvements for efficient teaching computer science students about hardware," in *Proc. IEEE EDUCON*, Berlin, Germany, 2013, pp. 295-302.
- [56] N. Ackovska and S. Ristov, "OER approach for specific student groups in hardware-based courses," *IEEE Trans. Educ.*, vol. 57, issue. 4, pp. 242-247, Nov. 2014, 10.1109/TE.2014.2327007.
- [57] N. L. Ribas-Xirgo, "Yet another simple processor (YASP) for introductory courses on Computer Architecture," *IEEE Trans. Ind. Electron.*, vol. 57, issue. 10, pp. 3317-3323, Oct. 2010, 10.1109/TIE.2010.2044118.
- [58] D. C-T Lo, K. Qian, and L. Hong, "The use of low cost portable microcontrollers in teaching undergraduate Computer Architecture and Organization," in *Proc. 2<sup>nd</sup> Integrated STEM Education Conference (ISEC)*, Ewing, NJ, USA, 2012, pp. 1-4.
- [59] J. O. Hamblen, Z. C. Smith, and W. W. Woo, "Introducing embedded systems in the first C/C++ programming class," in *Proc. International Conference on Microelectronic Systems Education (MSE)*, Austin, TX, USA, 2013, pp. 1-4.
- [60] M. Anguita and F. J. Fernandez-Baldomero, "Software optimization for improving student motivation in a Computer Architecture course," *IEEE Trans. Educ.*, vol. 50, issue 4, pp. 373-378, Nov. 2007, 10.1109/TE.2007.906603.
- [61] M. Gasser, Y-H Lu, and C-K Koh, "Outreach project introducing computer engineering to high school students," in *Proc. Frontiers in Education Conference (FIE)*, Washington, DC, USA, 2010, pp. F2E:1-5.
- [62] F. Geth, et al, "Development of an open-source smart energy house for K-12 education," in *Proc. Power and Energy Society General Meeting (PES)*, Vancouver, BC, Canada, 2013, pp. 1-5.
- [63] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: programming for all," *Commun ACM.*, vol. 52, issue 11, pp. 60-67, Nov. 2009, 10.1145/1592761.1592779.
- [64] T. Roffey, C. Sverko, and J. Therien, "The Making of a Makerspace: Pedagogical and Physical Transformations of Teaching and Learning (Curriculum Guide)," University of British Columbia, 2016. 40 pp.
- [65] J. C. Martínez-Santos, O. Acevedo-Patino, and S. H. Contreras-Ortiz, "Influence of Arduino on the development of advanced microcontrollers courses," *IEEE RITA*, vol. 12, no. 4, pp. 208-217, Nov. 2017, 10.1109/RITA.2017.2776444.

**Dimosthenis E. Bolanakis** recibió el B.S. licenciado en ingeniería electrónica por el Instituto de Educación Tecnológica Alexander de Thessaloniki, Grecia, en 2001, el M.S. Licenciado en tecnologías electrónicas modernas, por la Universidad de Ioannina (Departamento de Física), Grecia, en 2004, y el Ph.D. Licenciado en ciencias de la educación, en la Universidad de Ioannina (Departamento de Educación Primaria), Grecia, en 2016.

De 2003 a 2009, fue asociado de laboratorio en el Instituto de Tecnología Educativa Epirus para la enseñanza de la arquitectura de microcomputadoras, y también participó en proyectos de investigación para Reforzar la Educación Informática y la Ampliación de la Educación Superior. Trabajó en la industria desde 2010 hasta 2014, donde el período 2012-2014 se unió a European System Sensors SA. Desde 2014, ha estado ocupado como Especial de Laboratorio y Personal Docente en la Academia Helénica de la Fuerza Aérea, Grecia. Es (co) autor de más de 30 artículos, principalmente sobre Investigación en Educación en Ingeniería.