

A Literature Review on Embedded Systems

L. De Micco, *Senior Member, IEEE*, F. Vargas, *Senior Member, IEEE*, and P. Fierens, *Senior Member, IEEE*

Abstract—Embedded system design is crucial for the development of industry, technology, and science, and it is an area that has significantly grown in recent years throughout Latin America, both in academia and in industry. Embedded System (ES) refers to electronic equipment with a computing core which, unlike a personal computer, is designed to meet a specific function and is usually optimized to satisfy strict requirements of processing time, reliability, power consumption, size, and cost. With the advancement of research on the domains of Internet of Things (IoT), Computing Edge (CE) and Cloud Computing (CC) and their endless application possibilities, ESs have gained new roles and assumed unquestionable importance in our daily lives. In this article, we make a brief review of the relevant literature in the area of embedded systems in the Latin America region during recent years.

Index Terms—Embedded systems, Computer Architecture, FPGA, ASIC, HDL, DSP, RTOS, Embedded Software, Digital Signal Processing, Wireless Sensor Networks, Communications, Cryptography, Industry 4.0, Robotics, IoT, Agrotechnology, Bio-engineering.

I. INTRODUCCIÓN

EL término sistema embebido (SE) se refiere a un equipo electrónico que incluye procesamiento de datos, a veces en tiempo real, y que está diseñado para realizar una función dedicada, a diferencia de un ordenador de propósito general que cubre un amplio rango de necesidades [1], [2]. Hoy en día, en nuestra vida cotidiana estamos rodeados de muchos SEs, tales como un reproductor de MP3, un teléfono celular o hasta el sistema de control de una planta nuclear.

El cerebro de un SE es típicamente un microcontrolador o microprocesador, aunque los datos también pueden ser procesados por una matriz programable de compuertas lógicas (FPGA, *Field Programmable Gate Array*), un circuito que integra ambos (SoC, *System on Chip*), un procesador digital de señales (DSP, *Digital Signal Processor*), un circuito integrado para aplicaciones específicas tanto full, como semi-custom (ASIC, *Application-Specific Integrated Circuit*), en cualquier caso se combinan millones de transistores en un solo chip para su implementación (VLSI, *Very large-scale integration*). Se caracterizan, además, por tener una gran interacción con el entorno, a través de sensores y actuadores. Su diseño está optimizado para satisfacer requerimientos estrictos de tiempo de procesamiento, fiabilidad, consumo de potencia, tamaño y costo. En la Fig. 1 se muestra un diagrama en bloques simplificado de un sistema embebido. Central a un SE es la interacción con el medio físico, la cual tiene lugar a través

Luciana De Micco is at Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE). National University of Mar del Plata and the National Scientific and Technical Research Council (CONICET), Argentina.

Fabian Vargas is at the Catholic University (PUCRS) in Porto Alegre, Brazil.

Pablo I. Fierens is at the Instituto Tecnológico de Buenos Aires (ITBA) and CONICET, Argentina.

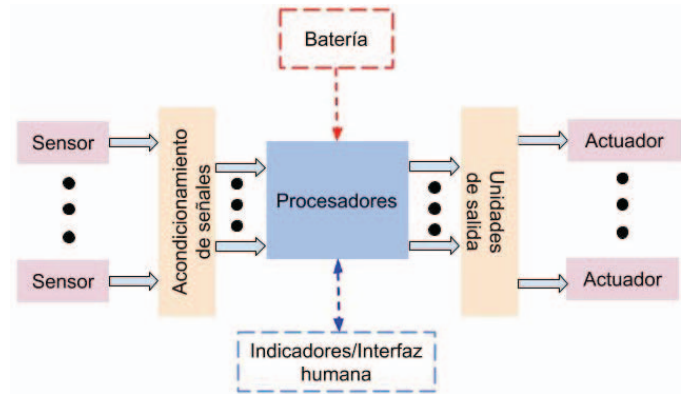


Fig. 1. Diagrama en bloques simplificado de un sistema embebido. Uno o varios procesadores ejercen acciones sobre el medio físico a través de actuadores, basando sus decisiones en variables medidas por sensores. Es común que los SEs sean sistemas autónomos alimentados por baterías y, en ocasiones, pueden incorporar una interfaz humana.

de sensores que miden variables del mismo y actuadores que accionan sobre él. Para poder interpretar las variables sensadas y tomar decisiones sobre cómo actuar, un sistema embebido necesita de uno o varios procesadores que pueden ser implementados físicamente, por ejemplo, en microcontroladores, FPGAs o ASICs. En muchas situaciones, los SEs son sistemas autónomos que se encuentran alimentados por baterías, las cuales a su vez pueden ser recargadas por energías renovables. No siempre se encuentra presente en un sistema embebido una interfaz computador-humano, viéndose muchas veces reducida a una mínima expresión como puede ser un conjunto de indicadores lumínicos.

La descripción de sistemas embebidos esbozada en los dos párrafos anteriores no es única. De hecho, una búsqueda de “¿Qué es un sistema embebido?” en cualquier buscador de Internet dará una gran cantidad de resultados con respuestas de lo más variadas. Muchas respuestas incluyen un relevamiento histórico sobre los orígenes de los SEs. Como hito fundamental se suele ubicar a la computadora de navegación en las misiones Apollo (véase, por ej., la referencia en Wikipedia [3] o el video *The History of Embedded Systems* [4] que fue parte de una charla plenaria en la *2008 Embedded Systems Conference*). Ciertamente, un sistema como este cumple con los términos de la definición dada más arriba.

En el análisis histórico, es difícil localizar el origen de la expresión inglesa *embedded system* para los sistemas que tratamos en este trabajo. Sin embargo, creemos que ha sido anterior la expresión *embedded computer system*. En relación a la misma, una definición de mediados de los '70 [5] (véase también la definición de la misma época en [6])

“An embedded computer system is a computer system that is integral to an electromechanical system

such as a combat weapons system, tactical system, aircraft, ship, missile, spacecraft, certain command and control systems, civilian systems such as a rapid transit system, and the like. [...] The key attributes of an embedded computer system are:

- It is a computer system that is physically incorporated into a larger system whose primary function is not data processing.
- It is integral to such a larger system from a design, procurement or operations- viewpoint.
- Its outputs generally include information, computer control signals and—computer data.”

Esta definición va al núcleo del origen de la expresión: un sistema embebido está físicamente incorporado en y es parte integral de (*embedded in*) un sistema mayor. Han pasado casi 50 años desde esta cita y el uso de SEs ha mutado en ese lapso. Aún así, el hecho fundamental que un SE no es una computadora genérica, sino un sistema específico sigue siendo válido.

Continuando con el análisis histórico, proponemos como métrica indirecta, ciertamente discutible, de la relevancia del área de sistemas embebidos al número de artículos de conferencia y revistas en la IEEE que contienen la expresión “*embedded system*”, ya sea en el título, el resumen o entre las palabras claves. La Fig. 2 muestra la evolución de dicho número. El primer artículo aparece hace casi 40 años, en 1981, y el número de artículos anuales no ha bajado de los 500 desde hace más de 15 años. Nótese que la elección de las palabras claves “*embedded system*” para la búsqueda es arbitraria. Hoy día, existen muchos sistemas embebidos que no son llamados por este nombre. Por ejemplo: las aplicaciones de Internet de las Cosas (IoT, según la sigla en inglés) están basadas en sistemas embebidos; los teléfonos móviles inteligentes y los televisores inteligentes son sistemas embebidos. Sin embargo, no hemos incluido “IoT”, “*smartphone*” o “*smart TV*” en la búsqueda. Es posible, por tanto, que el decrecimiento a partir del año 2010 observado en la Fig. 2 no sea producto de una disminución del trabajo en el área de sistemas embebidos, sino de un cambio en qué palabras se utilizan para hacer referencia a dicha área.

Entre las tantas definiciones no técnicas, y ciertamente parciales, de un sistema embebido que es posible encontrar está la siguiente [7]: “*An embedded system is one you don’t even know is there... until it stops working.*” Esta definición apunta a la ubicuidad de los SEs. Ciertamente, los sistemas embebidos integran conceptos de muchas áreas distintas y tienen aplicaciones en multitud de problemas tecnológicos.

En el presente artículo hacemos un breve repaso de varias áreas involucradas y las aplicaciones de SEs. Como referencia, utilizamos un indicador similar al anterior. En particular, en la Fig. 3 se contaron todos los artículos (publicados en la IEEE hasta el momento, Octubre 2019) que tratan de SEs y diversos temas que se mencionan a lo largo de este trabajo.

Entonces: ¿qué es un sistema embebido? Más allá de la definición dada al inicio de esa sección y de la breve discusión previa, esperamos que al llegar al final de este artículo, el lector pueda contestar esa pregunta por él mismo. En este trabajo hacemos un breve repaso de la literatura sobre

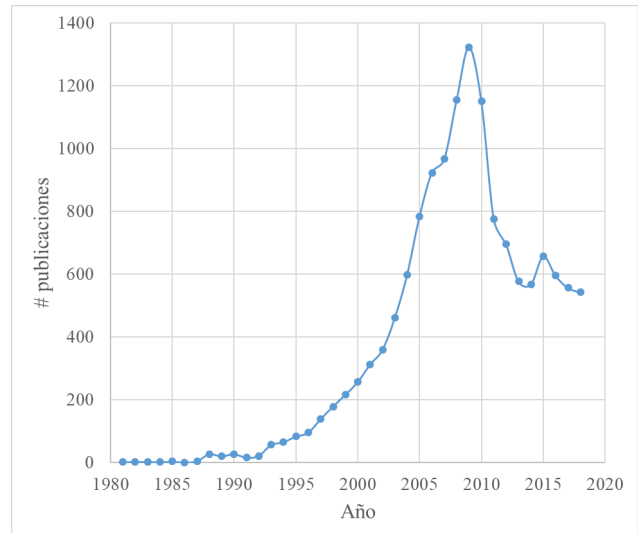


Fig. 2. Número de publicaciones sobre sistemas embebidos en la IEEE en función del año.

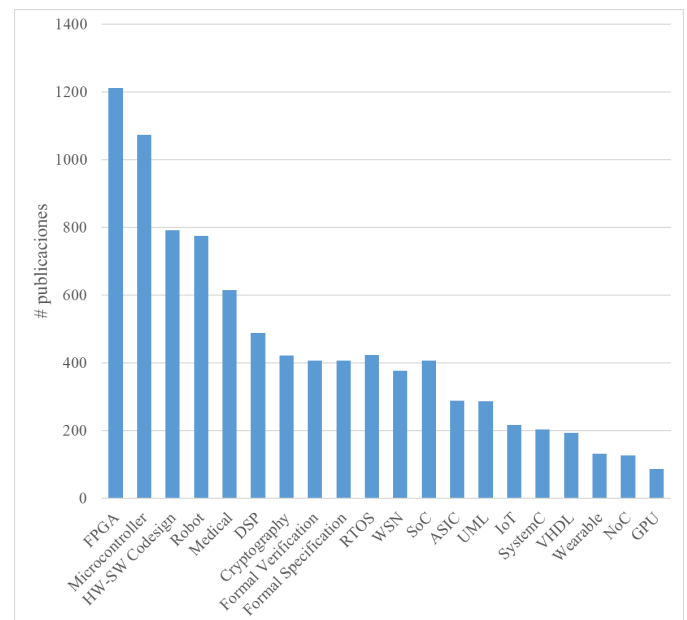


Fig. 3. Número de publicaciones sobre sistemas embebidos en la IEEE hasta Octubre de 2019, agrupadas por área.

sistemas embebidos en los últimos años. En la Sección II citamos algunos de los trabajos más relevantes sobre hardware específico a los SEs. El diseño de software para SEs también ocupa un lugar relevante en la literatura y lo repasamos en la Sección III. El abanico de aplicaciones de los sistemas embebidos es muy amplio y hacemos un breve resumen de algunas de las áreas cubiertas en la Sección IV. Cerramos este breve resumen con unas conclusiones y proyecciones a futuro del área en la Sección V.

II. HARDWARE PARA SISTEMAS EMBEBIDOS

Los sistemas embebidos son diseñados para realizar tareas dedicadas. Esto implica que, si bien existen plataformas para

el desarrollo de los SEs, el diseño del hardware asociado es siempre una parte importante del mismo.

Existe una gran literatura en el desarrollo de arquitecturas para SEs [1], [8]–[14], que van desde el uso de dispositivos genéricos como microcontroladores, *System on Chips* (SoCs) [15], [16], FPGAs, arquitecturas multinúcleo [17], hasta el desarrollo de arquitecturas específicas para optimizar la ejecución y dar soporte al software que se ejecutará en ellos [10], [18]. Tal es el caso de los sistemas en tiempo real que requieren plataformas con alta previsibilidad para permitir el análisis estático del peor tiempo de ejecución [19]–[21].

El núcleo de un sistema embebido suele ser un microcontrolador [22]–[24]. Estos procesadores generalmente contienen periféricos y sistemas específicos para tareas en las que serán utilizados. La especificidad de las tareas a realizar por un SE, hace de la lógica programable, como ser las FPGAs, un elemento frecuente de los mismos [25]. Cuando el volumen de producción o los requerimientos de la aplicación lo justifican, el desarrollo de *Application-Specific Integrated Circuits* (ASICs) es necesario [26]. Tradicionalmente, se suele considerar la eficiencia de los ASICs superior a la de las FPGAs, tanto en área utilizada como en consumo de energía. Sin embargo, el gap para ciertas aplicaciones se ha reducido en los últimos años [27].

Las herramientas de automatización de diseño electrónico (EDA, Electronic Design Automation) asistido por computadora (CAD, Computer-Aided Design) juegan un papel crucial e innegable en prácticamente todos los aspectos del diseño de los circuitos a medida que éstos se vuelven más complejos. Éstas herramientas abarcan desde el proyecto de circuitos integrados hasta el desarrollo de placas de circuito impreso.

Para el desarrollo de sistemas específicos, es habitual el uso de *Hardware Description Languages* (HDLs) como Verilog, VHDL (Very High Speed Integrated Circuit Hardware Description Language) [28]–[30] y SystemC [31]–[36].

En muchas aplicaciones, los sistemas embebidos deben realizar un intenso procesamiento digital de señales. En estos casos, los DSPs [37]–[39] y los *Graphics Processing Units* (GPUs) son alternativas comunes [40]–[43] ya que están optimizados para cálculos con valores en punto flotante. Sin embargo, cuando se requiere aumentar la velocidad de procesamiento e incrementar el *throughput* del sistema, una posibilidad es incorporar más de un núcleo en un mismo chip. Las plataformas *Multiprocessor Systems-on-Chips* (MPSoCs) usualmente integran procesadores de propósito general, como por ejemplo un core ARM, con núcleos de propósito especial, como DSP o GPU [44], de forma que el chip en su conjunto maximice el rendimiento. Por lo tanto, hay un fuerte énfasis en las cargas de trabajo paralelas y multiprogramadas, que tienen aspectos de computación y comunicación [45], [46]. En este nuevo escenario, las interconexiones internas entre los núcleos del chip toman especial relevancia ya que son el principal cuello de botella en el rendimiento de estos sistemas. Las interconexiones dedicadas y los buses compartidos presentan grandes problemas a medida que se incrementa el número de elementos integrados en un chip. De allí surge la arquitectura de comunicación *Network on chip* (NoC) [47]–[49], la cual comunica procesadores, memorias y otros diseños

personalizados mediante paquetes de conmutación en lugar de intercambiar mensajes o palabras, consiguiendo incrementar el rendimiento y el ancho de banda. Sin embargo, esta arquitectura de conmutación de paquetes para la comunicación interna del chip demanda recursos adicionales, tales como buffers de cola y nodos de procesamiento de paquetes, que aumentan el consumo de energía y espacio. Para abordar este problema se han propuesto soluciones para reemplazar los enrutadores electrónicos con conmutadores de paquetes ópticos o conmutadores de paquetes fotónicos [50]–[52].

En ocasiones, los SEs funcionan como sistemas autónomos alimentados por baterías, por ello el desarrollo de sistemas de bajo consumo es de gran relevancia [53]–[55]. Por otro lado, la literatura ha explorado la utilización de fuentes alternativas de energía para su alimentación [56], como ser energía solar [57], [58] o radiofrecuencia [59].

Los SEs se caracterizan por procesar información sensada y por actuar sobre diversos sistemas. En este sentido, el uso de sensores y actuadores forma parte integral de los sistemas embebidos. Generalmente los periféricos con los que interactúan son especializados, como por ejemplo dispositivos mecánicos y químicos, por lo que requieren de actuadores y sensores específicos para cada aplicación. Los sensores empleados van desde sensores de luz, de temperatura, acelerómetros, etc., hasta los nuevos desarrollos de sensores inteligentes específicos para aplicaciones como redes inteligentes, ciudades inteligentes o monitoreo de salud estructural, que se enfocan en el bajo consumo de potencia y son capaces de recolectar información y comunicarse entre sí [60]–[62].

Los conversores Analógico-Digital (ADC) y Digital-Analógico (DAC) juegan un papel primordial en este tipo de sistemas, ya que son indispensables para poder procesar las señales digitalmente como así también para actuar sobre el medio. La conversión analógico-digital implica primero un muestreo de la señal en tiempo y luego una cuantificación en amplitud, donde los requerimientos en frecuencia de muestreo y resolución son los principales determinantes del tipo de conversor a emplear. Para conversiones de alta resolución se emplean ADCs sigma-delta, ya que la modulación sigma-delta de la señal sobremuestreada actúa como un filtro pasa-altos para el ruido de cuantización, logrando disminuir significativamente la potencia de ruido en la banda de frecuencias de interés. El resultado es un mayor número efectivo de bits (ENOB - *Effective Number of Bits*) [63]. Actualmente, los conversores sigma-delta pueden tener resoluciones de 16 a 24 bits, pero para señales de frecuencias menores a 1 MHz, aproximadamente. Para aplicaciones que requieren alta velocidad, a menudo se usan conversores Flash. Si bien este tipo de ADCs es el más rápido, tiene baja resolución (típicamente hasta 8 bits) y alto consumo de potencia debido a la gran cantidad de componentes que requiere. Un tema de estudio es el desarrollo de arquitecturas de conversores Flash que reduzcan el uso de componentes y el consumo de potencia [64], [65]. Cuando un bajo consumo de potencia es importante, una opción es el conversor de aproximación sucesiva (SAR - *Successive Approximation Register*), el cual presenta bajo consumo de potencia gracias a la simplicidad de su estructura. Estos conversores van de media a alta resolución (entre 8 y

18 bits) y proporcionan una velocidad de muestreo de hasta 5 Msps. Gracias a ser casi en su totalidad digitales, escalan de acuerdo al avance de la tecnología de la microelectrónica. Por este motivo, entre sus parámetros generalmente se especifica la tecnología empleada en nanómetros. La reducción del tamaño de los transistores permite disminuir el área del circuito integrado digital, aumentar la velocidad y bajar el consumo de potencia. En la actualidad se pueden encontrar longitudes de canal menores a los 8 nm. La disminución del canal es un tema de estudio ya que acarrea problemas tecnológicos como el uso de dieléctricos de alta constante dieléctrica, el control de la degradación debido a los Efectos de Canal Corto (SCE - *Short Channel Effect*), así como la dificultad de lograr una alta reproducibilidad [66]–[68].

El diseño de SEs implica la integración de hardware heterogéneo ya que sus componentes provienen en general de una amplia variedad de dominios, donde cada dominio es compatible con un lenguaje específico, formalismo de modelado y marco de simulación. Las plataformas virtuales son un soporte poderoso para el desarrollo y la validación de hardware embebido, principalmente cuando el hardware empleado es heterogéneo [69], [70].

III. SOFTWARE PARA SISTEMAS EMBEBIDOS

El avance en estos sistemas ha generado una demanda en el desarrollo de arquitecturas de conjunto de instrucciones (ISA, *Instruction Set Architecture*) libres, en 1994 surge SPARC V8, en 2000 OpenRISC y en 2010 RISC V [71]. Esta última ha tenido respaldo de las empresas de semiconductores y en la actualidad se han desarrollado varios cores que lo emplean [72].

A diferencia de un software de propósito general, la función principal del software embebido es permitir la interacción del sistema embebido con el mundo físico, por esto, debe vincularse con el mundo y considerar cuestiones como sincronización, fiabilidad, robustez y consumo de energía. El desarrollo de software específico para su empleo en SEs es una tarea compleja. Por un lado los sistemas embebidos deben presentar prestaciones sofisticadas y fiables, por el otro se requieren costos bajos y con tiempos de lanzamiento al mercado muy acotados [73].

Una de las alternativas frecuentemente empleada para el desarrollo de software embebido son los modelos. Los modelos representan una abstracción del software embebido y el entorno en el que opera, capturando simultáneamente los requisitos y el diseño de la aplicación [74]–[77]. Algunos lenguajes de modelado incluyen diagramas de estado, como por ejemplo StateCharts de *Unified Modeling Language* (UML) [78], o la extensión Stateflow de Simulink Matlab [79].

Dentro de aplicaciones de un mismo tipo, generalmente el código de estos SEs es muy similar, de este hecho surgen los conocidos como *Frameworks*. Un *Framework* es una herramienta de programación especializada cuyo fin es el de facilitar el desarrollo de las aplicaciones, y de esta forma reducir el tiempo y los costos de diseño. Un *Framework* generalmente está por encima del sistema operativo, y lo constituyen un conjunto de funcionalidades organizadas en bibliotecas de código específicas [80].

En ciertas áreas de aplicación, los sistemas embebidos tienen requerimientos de fiabilidad, disponibilidad, mantenibilidad y seguridad muy altos como es el caso de los software para aviones comerciales. Existe una gran literatura sobre el desarrollo de software que satisfaga dichos requerimientos [81], [82] y la verificación formal del código [83]–[88].

En ocasiones, los SEs son diseñados como “sistemas de tiempo real” [89], [90]. En particular, en muchas ocasiones los núcleos computacionales utilizan *Real Time Operating Systems* (RTOS) [91], [92], tales como QNX [93], WinCE [94], Vxworks [95] y FreeRTOS [96], entre muchos otros.

La lógica de ciertas aplicaciones de los SEs, hace natural el empleo de la programación dirigida por eventos o *event-driven programming* (véase, por ej., [97], [98]). En estos casos, la ejecución de la aplicación sucede al recibir un evento, permaneciendo en reposo mientras no reciba un estímulo asíncrono que lo convierte en el *objeto activo*.

Si bien existen procesadores cada vez más potentes, en muchas aplicaciones de los SEs existen limitaciones a la capacidad de procesamiento de sus núcleos computacionales. Esto lleva a que existan diversas estrategias para adaptar algoritmos existentes para su uso en SEs. En particular, se destacan desarrollos específicos de algoritmos de procesamiento de señales [99] y aprendizaje automático [90], [100], [101].

La complejidad de un sistema embebido diseñado para una tarea específica obliga a considerar simultáneamente aspectos de software y hardware durante la etapa de desarrollo. El codiseño [102]–[104] y la cosimulación de hardware y software son enfoque establecidos en el área [105], [106].

IV. ALGUNAS APLICACIONES

Los sistemas embebidos tienen aplicaciones en sistemas de comunicación. Más aún, existen sistemas y protocolos particulares que son diseñados específicamente para la comunicación entre sistemas embebidos.

Los SEs se encuentran en la base de muchos sistemas de comunicaciones. En particular, muchos de los diseños de radios definidas por software (SDR, *Software Defined Radio*) pueden considerarse sistemas embebidos [107].

Las redes de sensores inalámbricas (WSN, *Wireless Sensor Networks*) se encuentran formadas por nodos que son en sí mismos sistema embebidos. El desarrollo de protocolos de comunicación y de enrutamiento eficiente en este tipo de redes tiene una muy amplia literatura que no es posible abarcar completamente en este breve resumen [108], [109].

La seguridad en los sistemas de comunicaciones es imprescindible [110]–[112]. Los algoritmos criptográficos suelen ser computacionalmente muy costosos, por lo que su adaptación a los sistemas embebidos ha ocupado el trabajo de muchos investigadores del área [113]–[115].

Los avances en sistemas embebidos aplicados a productos de consumo masivo han cambiado significativamente la forma en que nos conectamos con el mundo que nos rodea y ha mejorado nuestra calidad de vida. Dispositivos tales como teléfonos, tabletas portátiles y relojes inteligentes permiten mantener un acceso permanente a la información [116]. También han revolucionado dispositivos de uso cotidiano tales como televisores, sistemas de audio y electrodomésticos.

Los automóviles modernos usan varios sistemas embebidos para el control de diferentes características, como ser el sistema de aire acondicionado, el tablero, la transmisión, el motor, y diversos aspectos de la seguridad [117]. La complejidad es tal que la comunicación interna de estos SEs es en sí misma un área de estudio [118]. Los sistemas embebidos también tienen aplicación en otros tipos de transporte, como ser en los sistemas ferroviarios [119]. En los últimos años, ha cobrado mayor relevancia su uso en vehículos autónomos de diferentes tipos, automóviles [120], barcos y submarinos [121] y vehículos aéreos [122].

Los SEs ocupan un lugar de gran relevancia en la industria, especialmente en el control de producción [123], [124]. Los sistemas embebidos también forman parte esencial de los sistemas robóticos industriales [123], [125]. Más importante aún es el rol de los SEs en la así llamada cuarta revolución industrial o Industria 4.0 [126]. En la base de este paradigma se encuentra una gran cantidad de sistemas cyber-físicos interconectados entre sí. El concepto de *Cyber-Physical System* (CPS), una abstracción de la integración entre la computación y los sistemas físicos, es una evolución natural de la idea de sistema embebido, pero más orientada a la conexión en red [127]–[129].

Una idea relacionada es la de la Internet de las Cosas (*Internet of Things*, IoT), a tal punto que CPS e IoT son usados indistintamente por algunos autores [130]. IoT trae la promesa de un mundo con aún mayor conexión y control de cada una de las características de nuestros hogares, automóviles, oficinas, etc. La esperada revolución de IoT no sería posible sin los sistemas embebidos sobre la cual se sustenta [131], [132]. Esto implica que los sistemas embebidos integren capacidades de red, capacidades de control de acceso basadas en roles, capacidades de control remoto o capacidades relacionadas.

Una de las aplicaciones más importantes de los sistemas embebidos, especialmente en Latinoamérica, es en la agroindustria, sea en el monitoreo remoto y control de plantaciones como en procesos posteriores de elaboración [133], [134].

Las aplicaciones de los sistemas embebidos a la bioingeniería son comunes: sistemas de electrocardiografía (ECG) portables [135]–[138], sistemas de electroencefalograma (EEG) e interfaz computadora-cerebro (*Brain-Computer Interface*, BCI) [139]–[146] y gran variedad de sensores portables embebidos (*wearable sensors*) [147]–[150]. En particular, los SEs tienen un lugar importante en la salud de las personas, por ejemplo como sistemas de sensores implantables [151]. Uno de los ejemplos más destacados lo constituyen los marcapasos cardíacos [77], [152], [153]. En los últimos años, se ha prestado mayor atención a la seguridad de estos sistemas embebidos ante ataques [128], [154], [155].

El avance en el desarrollo de los SEs está fuertemente ligado al avance de la tecnología, ya que las principales características de un sistema embebido son el bajo costo y bajo consumo de potencia. Por lo tanto, las innovaciones en estas direcciones son las que más afectan a estos sistemas. Sin embargo, existe un amplio abanico de áreas de la electrónica que también influye, tanto los dispositivos que son parte de su estructura básica como microcontroladores, microprocesadores, FPGAs y DSPs [156], hasta los avances en, por ejemplo, criptografía

ya que la seguridad es imprescindible en estos sistemas [113], [157]–[166].

V. CONCLUSIONES

En este artículo, hemos hecho un breve repaso de algunos de los tópicos habitualmente relacionados con sistemas embebidos. El lector puede quedarse con la impresión de que “Sistemas Embebidos” es una expresión utilizada como título general que cubre demasiadas áreas. La realidad es que, los SEs forman parte de multitud de sistemas y están presentes en gran parte de los desarrollos tecnológicos. Es por eso que hemos nombrado áreas como arquitectura de computadoras, comunicaciones, verificación de código, sistemas de tiempo real, robótica, agrotecnología, bioingeniería, productos de consumo masivo, IoT, etc. Al hacer esta lista, no queremos decir que todas estas áreas son sólo una parte del gran área de SEs. Por el contrario, sólo deseamos explicar que los sistemas embebidos forman parte integral de un gran número de desarrollos científicos y tecnológicos. Por este motivo, el diseño de SEs es un campo que en los últimos años ha crecido notablemente en toda Latinoamérica, tanto en la academia como en la industria.

Se puede discutir, por tanto, la necesidad de hablar de sistemas embebidos *per se*. Nosotros creemos que sí es necesario. Así como los circuitos electrónicos forman parte de todas esas aplicaciones de las cuales mencionamos y tiene sentido hablar del diseño de circuitos electrónicos en sí mismo, como un área bien establecida, también es menester hablar del desarrollo de sistemas embebidos como un área bien definida. Existen numerosos criterios de diseño de hardware, software, protocolos de comunicación y estrategias de implementación de SEs que forman un cuerpo de conocimiento gran relevancia. Dicha relevancia se observa, no sólo en las citas en este artículo, sino también en la realización periódica de importantes conferencias nacionales e internacionales, como ser la International Conference on Embedded Systems and Applications (EMSA), la IEEE International Conference on Embedded Software and Systems (ICESS), la ACM SIGBED International Conference on Embedded Software (EMSOFT), el Congreso Argentino de Sistemas Embebidos (CASE), el Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC) y el IEEE Latin American Test Symposium (LATS), entre otras.

Posiblemente, en los próximos años el impulso del área de SEs provenga de la Industria 4.0, en particular como parte de los conceptos de Sistema cyber-físico e Internet de las Cosas. Estas nuevas abstracciones orientan aún más a los sistemas embebidos hacia la conectividad y los ubica, como en la vieja definición de 1975 citada en la Sección I, como partes “*physically incorporated into a larger system*” cuyo objetivo no es el mero procesamiento de datos, sino el funcionamiento de una sociedad y sus medios de producción.

AGRADECIMIENTOS

Deseamos agradecer especialmente el aporte y la discusión con los miembros del comité editorial invitado de este número especial sobre sistemas embebidos.

REFERENCIAS

- [1] T. Noergaard, *Embedded systems architecture: a comprehensive guide for engineers and programmers*. Newnes, 2012.
- [2] P. Marwedel, *Embedded system design*. Springer, 2006, vol. 1.
- [3] Wikipedia, “Embedded system,” https://en.wikipedia.org/wiki/Embedded_system, accessed: 2019-10-09.
- [4] E. Times, “History of embedded system,” <https://www.youtube.com/watch?v=1hX4urTFXr0>, accessed: 2019-10-09.
- [5] A. Kossiakoff, T. Sleight, E. Prettyman, J. Park, and P. Hazan, “Dod weapon systems software management study,” Johns Hopkins University, Applied Physics Laboratory, Tech. Rep., 1975.
- [6] D. L. Parnas, “Use of abstract interfaces in the development of software for embedded computer systems,” Naval Research Laboratory, Tech. Rep., 1977.
- [7] M. Maxfield, “What is an embedded system?” https://www.eetimes.com/author.asp?section_id=216&doc_id=1322671, accessed: 2019-10-09.
- [8] B. P. Dave, G. Lakshminarayana, and N. K. Jha, “Cosyn: Hardware-software co-synthesis of embedded systems,” in *Proceedings of the 34th annual Design Automation Conference*. ACM, 1997, pp. 703–708.
- [9] T. S. Hall and J. O. Hamblen, “Using system-on-a-programmable-chip technology to design embedded systems,” *International Journal of Computer Architectures*, vol. 13, no. 6, pp. 142–152, 2006.
- [10] M. Schoeberl, “A java processor architecture for embedded real-time systems,” *Journal of Systems Architecture*, vol. 54, no. 1-2, pp. 265–286, 2008.
- [11] P. Garcia, K. Compton, M. Schulte, E. Blem, and W. Fu, “An overview of reconfigurable hardware in embedded systems,” *EURASIP Journal on Embedded Systems*, vol. 2006, no. 1, p. 056320, 2006.
- [12] R. Kamal, *Embedded systems: architecture, programming and design*. Tata McGraw-Hill Education, 2011.
- [13] J. G. Tong, I. D. Anderson, and M. A. Khalid, “Soft-core processors for embedded systems,” in *2006 International Conference on Micro-electronics*. IEEE, 2006, pp. 170–173.
- [14] S. S. Bhattacharyya, E. F. Deprettere, R. Leupers, and J. Takala, *Handbook of signal processing systems*. Springer, 2018.
- [15] J. Teich, “Hardware/software codesign: The past, the present, and predicting the future,” *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1411–1430, 2012.
- [16] N. Hou, X. Yan, and F. He, “A survey on partitioning models, solution algorithms and algorithm parallelization for hardware/software co-design,” *Design Automation for Embedded Systems*, pp. 1–21, 2019.
- [17] G. Kornaros, *Multi-core embedded systems*. CRC Press, 2010.
- [18] J. Balfour, W. Dally, D. Black-Schaffer, V. Parikh, and J. Park, “An energy-efficient processor architecture for embedded systems,” *IEEE Computer Architecture Letters*, vol. 7, no. 1, pp. 29–32, 2008.
- [19] L. Thiele and R. Wilhelm, “Design for timing predictability,” *Real-Time Systems*, vol. 28, no. 2-3, pp. 157–177, 2004.
- [20] M. Schoeberl, S. Abbaspour, B. Akesson, N. Audsley, R. Capasso, J. Garside, K. Goossens, S. Goossens, S. Hansen, R. Heckmann et al., “T-crest: Time-predictable multi-core architecture for embedded systems,” *Journal of Systems Architecture*, vol. 61, no. 9, pp. 449–471, 2015.
- [21] C. Cullmann, C. Ferdinand, G. Gebhard, D. Grund, C. Maiza, J. Reineke, B. Triquet, and R. Wilhelm, “Predictability considerations in the design of multi-core embedded systems,” *Proceedings of Embedded Real Time Software and Systems*, pp. 36–42, 2010.
- [22] S. Heath, *Embedded systems design*. Elsevier, 2002.
- [23] R. Toulson and T. Wilmshurst, *Fast and effective embedded systems design: applying the ARM mbed*. Newnes, 2016.
- [24] J. K. Peckol, *Embedded systems: a contemporary design tool*. John Wiley & Sons, 2019.
- [25] R. Sass and A. G. Schmidt, *Embedded systems design with platform FPGAs: principles and practices*. Morgan Kaufmann, 2010.
- [26] W. J. Dally, J. Balfour, D. Black-Schaffer, J. Chen, R. C. Harting, V. Parikh, J. Park, and D. Sheffield, “Efficient embedded computing,” *Computer*, vol. 41, no. 7, pp. 27–32, 2008.
- [27] E. Nurvitadhi, J. Sim, D. Sheffield, A. Mishra, S. Krishnan, and D. Marr, “Accelerating recurrent neural networks in analytics servers: Comparison of fpga, cpu, gpu, and asic,” in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1–4.
- [28] F. Vahid, S. Narayan, and D. D. Gajski, “Speccharts: A vhdl front-end for embedded systems,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 6, pp. 694–706, 1995.
- [29] W. E. McUmbler and B. H. Cheng, “Uml-based analysis of embedded systems using a mapping to vhdl,” in *Proceedings 4th IEEE International Symposium on High-Assurance Systems Engineering*. IEEE, 1999, pp. 56–63.
- [30] T. G. Moreira, M. A. Wehrmeister, C. E. Pereira, J.-F. Petin, and E. Levrat, “Automatic code generation for embedded systems: From uml specifications to vhdl code,” in *2010 8th IEEE International Conference on Industrial Informatics*. IEEE, 2010, pp. 1085–1090.
- [31] A. Fin, F. Fummi, M. Martignano, and M. Signoreto, “Systemc: a homogenous environment to test embedded systems,” in *Proceedings of the ninth international symposium on Hardware/software codesign*. ACM, 2001, pp. 17–22.
- [32] F. Herrera, H. Posadas, P. Sánchez, and E. Villar, “Systematic embedded software generation from systemc,” in *Embedded software for SoC*. Springer, 2003, pp. 83–93.
- [33] S. Rigo, G. Araujo, M. Bartholomeu, and R. Azevedo, “Archc: A systemc-based architecture description language,” in *16th Symposium on Computer Architecture and High Performance Computing*. IEEE, 2004, pp. 66–73.
- [34] H. Posadas, J. Adamez, E. Villar, F. Blasco, and F. Escuder, “Rtos modeling in systemc for real-time embedded sw simulation: A posix model,” *Design Automation for Embedded Systems*, vol. 10, no. 4, pp. 209–227, 2005.
- [35] F. Herrera and E. Villar, “A framework for embedded system specification under different models of computation in systemc,” in *2006 43rd ACM/IEEE Design Automation Conference*. IEEE, 2006, pp. 911–914.
- [36] F. Fummi, D. Quaglia, and F. Stefanni, “A systemc-based framework for modeling and simulation of networked embedded systems,” in *2008 Forum on Specification, Verification and Design Languages*. IEEE, 2008, pp. 49–54.
- [37] M. Bramberger, J. Brunner, B. Rinner, and H. Schwabach, “Real-time video analysis on an embedded smart camera for traffic surveillance,” in *Proceedings. RTAS 2004. 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004*. IEEE, 2004, pp. 174–181.
- [38] M. Litzenberger, C. Posch, D. Bauer, A. N. Belbachir, P. Schon, B. Kohn, and H. Garn, “Embedded vision system for real-time object tracking using an asynchronous transient vision sensor,” in *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*. IEEE, 2006, pp. 173–178.
- [39] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, “Distributed embedded smart cameras for surveillance applications,” *Computer*, vol. 39, no. 2, pp. 68–75, 2006.
- [40] A. Sudarsanam, *Code optimization libraries for retargetable compilation for embedded digital signal processors*. Princeton University, 1998.
- [41] J. Chase, B. Nelson, J. Bodily, Z. Wei, and D.-J. Lee, “Real-time optical flow calculations on fpga and gpu architectures: a comparison study,” in *2008 16th International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2008, pp. 173–182.
- [42] A. P. D. Binotto, D. Doering, T. Stetzelberger, P. McVittie, S. Zimmermann, and C. E. Pereira, “A cpu, gpu, fpga system for x-ray image processing using high-speed scientific cameras,” in *2013 25th International Symposium on Computer Architecture and High Performance Computing*. IEEE, 2013, pp. 113–119.
- [43] D. Hernandez-Juarez, A. Chacón, A. Espinosa, D. Vázquez, J. C. Moure, and A. M. López, “Embedded real-time stereo estimation via semi-global matching on the gpu,” *Procedia Computer Science*, vol. 80, pp. 143–153, 2016.
- [44] K. Smiri, S. Bekri, and H. Smei, “Fault-tolerant in embedded systems (mpsoc): Performance estimation and dynamic migration tasks,” in *2016 11th International Design & Test Symposium (IDT)*. IEEE, 2016, pp. 1–6.
- [45] O. Terzo, K. Djemame, A. Scionti, and C. Pezuela, *Heterogeneous Computing Architectures: Challenges and Vision*. CRC Press, 2019.
- [46] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown et al., “Tile64-processor: A 64-core soc with mesh interconnect,” in *2008 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*. IEEE, 2008, pp. 88–598.
- [47] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, “A network on chip architecture and design methodology,” in *Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*. IEEE, 2002, pp. 117–124.

- [48] N. Venkataraman and R. Kumar, "Design and analysis of application specific network on chip for reliable custom topology," *Computer Networks*, vol. 158, pp. 69–76, 2019.
- [49] P. G. Raponi, E. Norige, and S. Kumar, "Generation of network-on-chip layout based on user specified topological constraints," Feb. 26 2019, uS Patent App. 10/218,581.
- [50] D. Zydek, N. Shlayan, E. Regentova, and H. Selvaraj, "Review of packet switching technologies for future noc," in *2008 19th International Conference on Systems Engineering*. IEEE, 2008, pp. 306–311.
- [51] J. Bashir, E. Peter, and S. R. Sarangi, "A survey of on-chip optical interconnects," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, p. 115, 2019.
- [52] T. Alexoudi, N. Terzenidis, S. Pitris, M. Moralis-Pegios, P. Maniotis, C. Vagionas, C. Mitsolidou, G. Mourgias-Alexandris, G. T. Kanellos, A. Miliou *et al.*, "Optics in computing: from photonic network-on-chip to chip-to-chip interconnects and disintegrated architectures," *Journal of Lightwave Technology*, vol. 37, no. 2, pp. 363–379, 2019.
- [53] Y. Shin, K. Choi, T. Sakurai, T. Sakurai, and T. Sakurai, "Power optimization of real-time embedded systems on variable speed processors," in *Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design*. IEEE Press, 2000, pp. 365–368.
- [54] S. Segars, "Low power design techniques for microprocessors," in *International Solid-State Circuits Conference Tutorial*, 2001, pp. 1–67.
- [55] C. Zhang, F. Vahid, and W. Najjar, "A highly configurable cache architecture for embedded systems," in *30th Annual International Symposium on Computer Architecture, 2003. Proceedings*. IEEE, 2003, pp. 136–146.
- [56] S. Chalasani and J. M. Conrad, "A survey of energy harvesting sources for embedded systems," in *IEEE SoutheastCon 2008*. IEEE, 2008, pp. 442–447.
- [57] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 64.
- [58] V. Raghunathan and P. H. Chou, "Design and power management of energy harvesting embedded systems," in *Proceedings of the 2006 international symposium on Low power electronics and design*. ACM, 2006, pp. 369–374.
- [59] T. Soyata, L. Copeland, and W. Heinzelman, "Rf energy harvesting for embedded systems: A survey of tradeoffs and methodology," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 22–57, 2016.
- [60] B. Spencer Jr, M. E. Ruiz-Sandoval, and N. Kurata, "Smart sensing technology: opportunities and challenges," *Structural Control and Health Monitoring*, vol. 11, no. 4, pp. 349–368, 2004.
- [61] J. Marulanda, P. Thomson, J. Marulanda *et al.*, "Monitoreo de salud estructural," *Ingeniería y Competitividad*, vol. 2, no. 2, pp. 40–46, 2011.
- [62] Y. Fu, K. Mechtov, T. Hoang, J. R. Kim, D. H. Lee, and B. F. Spencer Jr, "Development and full-scale validation of high-fidelity data acquisition on a next-generation wireless smart sensor platform," *Advances in Structural Engineering*, p. 1369433219866093, 2019.
- [63] M. Ortmanns and F. Gerfers, *Continuous-time sigma-delta A/D conversion: fundamentals, performance limits and robust implementations*. Springer, 2006.
- [64] M. S. Njinowa, H. T. Bui, and F.-R. Boyer, "Design of low power 4-bit flash adc based on standard cells," in *2013 IEEE 11th International New Circuits and Systems Conference (NEWCAS)*. IEEE, 2013, pp. 1–4.
- [65] F. Begum, S. Mishra, and A. Dandapat, "Low power 10-bit flash adc with divide and collate subranging conversion scheme," *Scientia Iranica*, 2019.
- [66] H. Wong and H. Iwai, "On the scaling issues and high- κ replacement of ultrathin gate dielectrics for nanoscale mos transistors," *Microelectronic Engineering*, vol. 83, no. 10, pp. 1867–1904, 2006.
- [67] H. Iwai, "Roadmap for 22 nm and beyond," *Microelectronic Engineering*, vol. 86, no. 7-9, pp. 1520–1528, 2009.
- [68] P. Wang, "Device characteristics of short-channel and narrow-width mosfet's," *IEEE Transactions on Electron Devices*, vol. 25, no. 7, pp. 779–786, 1978.
- [69] K. Keutzer, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: orthogonalization of concerns and platform-based design," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 19, no. 12, pp. 1523–1543, 2000.
- [70] M. Lora, S. Vinco, and F. Fummi, "Translation, abstraction and integration for effective smart system design," *IEEE Transactions on Computers*, 2019.
- [71] K. Asanović and D. A. Patterson, "Instruction sets should be free: The case for risc-v," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146*, 2014.
- [72] Y. Lee, A. Waterman, R. Avizienis, H. Cook, C. Sun, V. Stojanović, and K. Asanović, "A 45nm 1.3 ghz 16.7 double-precision gflops/w risc-v processor with vector accelerators," in *ESSCIRC 2014-40th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2014, pp. 199–202.
- [73] C. Ebert and C. Jones, "Embedded software: Facts, figures, and future," *Computer*, vol. 42, no. 4, pp. 42–52, 2009.
- [74] J. Sztipanovits and G. Karsai, "Model-integrated computing," *Computer*, vol. 30, no. 4, pp. 110–111, 1997.
- [75] G. Karsai, J. Sztipanovits, A. Ledeczki, and T. Bapty, "Model-integrated development of embedded software," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 145–164, 2003.
- [76] H. Shokry and M. Hinchey, "Model-based verification of embedded software," *Computer*, 2009.
- [77] P. Liggesmeyer and M. Trapp, "Trends in embedded software engineering," *IEEE software*, vol. 26, no. 3, pp. 19–25, 2009.
- [78] B. P. Douglass, "Uml statecharts," *Embedded systems programming*, vol. 12, no. 1, pp. 22–42, 1999.
- [79] Matlab, "Stateflow," [urlhttps://es.mathworks.com/products/stateflow.html](https://es.mathworks.com/products/stateflow.html).
- [80] P. Johnston. (2018) C++ embedded frameworks. [Online]. Available: <https://embeddedartistry.com/newsletter-archive/2018/5/7/may-2018-c-embedded-frameworks>
- [81] Y. Ait-Ameur, G. Bel, F. Boniol, S. Pairault, and V. Wiels, "Robustness analysis of avionics embedded systems," *ACM SIGPLAN Notices*, vol. 38, no. 7, pp. 123–132, 2003.
- [82] Y. Moy, E. Ledinot, H. Delseny, V. Wiels, and B. Monate, "Testing or formal verification: Do-178c alternatives and industrial experience," *IEEE software*, vol. 30, no. 3, pp. 50–57, 2013.
- [83] F. Balarin, Y. Watanabe, H. Hsieh, L. Lavagno, C. Passerone, and A. Sangiovanni-Vincentelli, "Metropolis: An integrated electronic system design environment," *Computer*, vol. 36, no. 4, pp. 45–52, 2003.
- [84] C. L. Heitmeyer, M. Archer, E. I. Leonard, and J. McLean, "Formal specification and verification of data separation in a separation kernel for an embedded system," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 346–355.
- [85] D. Große, U. Kühne, and R. Drechsler, "Hw/sw co-verification of embedded systems using bounded model checking," in *Proceedings of the 16th ACM Great Lakes symposium on VLSI*. ACM, 2006, pp. 43–48.
- [86] A. Fehnker, R. Huuck, F. Rauch, and S. Seefried, "Some assembly required-program analysis of embedded system code," in *2008 Eighth IEEE International Working Conference on Source Code Analysis and Manipulation*. IEEE, 2008, pp. 15–24.
- [87] G. Klein, "Operating system verification—an overview," *Sadhana*, vol. 34, no. 1, pp. 27–69, 2009.
- [88] Y. M. Perez, H. A. P. Marin, and A. E. Bedoya, "A review on verification and validation for embedded software," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2339–2347, May 2016.
- [89] Q. Li and C. Yao, *Real-time concepts for embedded systems*. CRC Press, 2003.
- [90] R. Oshana and M. Kraeling, *Software engineering for embedded systems: Methods, practical techniques, and applications*. Newnes, 2019.
- [91] Q. Jabeen, F. Khan, M. N. Hayat, H. Khan, S. R. Jan, and F. Ullah, "A survey: Embedded systems supporting by different operating systems," *arXiv preprint arXiv:1610.07899*, 2016.
- [92] A. S. Berger, *Embedded systems design: an introduction to processes, tools, and techniques*. CRC Press, 2001.
- [93] D. Hildebrand, "An architectural overview of qnx," in *USENIX Workshop on Microkernels and Other Kernel Architectures*, 1992, pp. 113–126.
- [94] Microsoft, "Windows ce," <http://www.microsoft.com/windowsce/embedded/>, 2008.
- [95] A. Gerstlauer, H. Yu, and D. D. Gajski, "Rtos modeling for system level design," in *Embedded Software for SoC*. Springer, 2003, pp. 55–68.
- [96] F. org. (2018) Quality rtos & embedded software. [Online]. Available: <https://www.freertos.org/a00114.html>
- [97] F. Dabek, N. Zeldovich, F. Kaashoek, D. Mazières, and R. Morris, "Event-driven programming for robust software," in *Proceedings of the 10th workshop on ACM SIGOPS European workshop*. ACM, 2002, pp. 186–189.
- [98] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Protothreads: Simplifying event-driven programming of memory-constrained embedded

- systems,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*. Acm, 2006, pp. 29–42.
- [99] K. Dewald and D. Jacoby, “Signal processing in embedded systems,” *IEEE Latin America Transactions*, vol. 11, no. 1, pp. 664–667, Feb 2013.
- [100] L. P. Kaelbling, *Learning in embedded systems*. MIT press, 1993.
- [101] B. Taylor, V. S. Marco, W. Wolff, Y. Elkhatib, and Z. Wang, “Adaptive deep learning model selection on embedded systems,” in *ACM SIGPLAN Notices*, vol. 53, no. 6. ACM, 2018, pp. 31–43.
- [102] R. Ernst, “Codesign of embedded systems: Status and trends,” *IEEE Design & Test of Computers*, vol. 15, no. 2, pp. 45–54, 1998.
- [103] W. Wolf, “A decade of hardware/software codesign,” *Computer*, no. 4, pp. 38–43, 2003.
- [104] J. Vidal, F. De Lamotte, G. Gogniat, P. Soulard, and J.-P. Diguët, “A co-design approach for embedded system modeling and code generation with uml and marte,” in *2009 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 2009, pp. 226–231.
- [105] K. Hines and G. Borriello, “Dynamic communication models in embedded system co-simulation,” in *Proceedings of the 34th annual Design Automation Conference*. ACM, 1997, pp. 395–400.
- [106] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, “Co-simulation: a survey,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, p. 49, 2018.
- [107] S. Sun, M. Kadoch, L. Gong, and B. Rong, “Integrating network function virtualization with sdr and sdn for 4g/5g networks,” *IEEE Network*, vol. 29, no. 3, pp. 54–59, 2015.
- [108] M. A. M. Vieira, C. N. Coelho, D. Da Silva, and J. M. da Mata, “Survey on wireless sensor network devices,” in *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No. 03TH8696)*, vol. 1. IEEE, 2003, pp. 537–544.
- [109] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [110] D. E. Robling Denning, *Cryptography and data security*. Addison-Wesley Longman Publishing Co., Inc., 1982.
- [111] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, “Security for industrial communication systems,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1152–1177, 2005.
- [112] W. Stallings, *Cryptography and Network Security, 4/E*. Pearson Education India, 2006.
- [113] T. Wollinger, J. Guajardo, and C. Paar, “Cryptography in embedded systems: An overview,” *Proc. Embedded World*, pp. 735–744, 2003.
- [114] O. Hyncica, P. Kucera, P. Honzik, and P. Fiedler, “Performance evaluation of symmetric cryptography in embedded systems,” in *Proceedings of the 6th IEEE international conference on intelligent data acquisition and advanced computing systems*, vol. 1. IEEE, 2011, pp. 277–282.
- [115] S. Marzougui and J. Krämer, “Post-quantum cryptography in embedded systems,” in *Proceedings of the 14th International Conference on Availability, Reliability and Security*. ACM, 2019, p. 48.
- [116] A. Ahmad, M. Anisetti, E. Damiani, and G. Jeon, “Special issue on real-time image and video processing in mobile embedded systems,” *Journal of Real-Time Image Processing*, vol. 16, no. 1, pp. 1–4, 2019.
- [117] A. Sangiovanni-Vincentelli and M. Di Natale, “Embedded system design for automotive applications,” *Computer*, vol. 40, no. 10, pp. 42–51, 2007.
- [118] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, “Trends in automotive communication systems,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, 2005.
- [119] E. Irrazábal, I. Sambrana, and C. Pinto Luft, “Proceso para el diseño de la arquitectura software en un sistema crítico ferroviario según la norma en-50128,” in *XIX Simposio Argentino de Ingeniería de Software (ASSE)-JAIHO 47 (CABA, 2018)*, 2018.
- [120] L. L. Bello, S. Mubeen, S. Saponara, R. Mariani, and U. D. Bordoloi, “Guest editorial embedded and networked systems for intelligent vehicles and robots,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1035–1037, 2019.
- [121] J. A. Monroy, E. Campos, and J. A. Torres, “Attitude control of a micro auv through an embedded system,” *IEEE Latin America Transactions*, vol. 15, no. 4, pp. 603–612, April 2017.
- [122] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, “Visually guided landing of an unmanned aerial vehicle,” *IEEE transactions on robotics and automation*, vol. 19, no. 3, pp. 371–380, 2003.
- [123] A. Malinowski and H. Yu, “Comparison of embedded system design for industrial applications,” *IEEE transactions on industrial informatics*, vol. 7, no. 2, pp. 244–254, 2011.
- [124] E. Cabal-Yepez, A. G. Garcia-Ramirez, R. J. Romero-Troncoso, A. Garcia-Perez, and R. A. Osornio-Rios, “Reconfigurable monitoring system for time-frequency analysis on industrial equipment through stft and dwt,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 760–771, 2012.
- [125] J. Kern, C. Urrea, R. Méndez, and G. González, “Development of an embedded control system by means of dspic applied in a 4 dof robot,” *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2099–2106, May 2016.
- [126] M. Rübmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch, “Industry 4.0: The future of productivity and growth in manufacturing industries,” *Boston Consulting Group*, vol. 9, no. 1, pp. 54–89, 2015.
- [127] E. A. Lee, “Cyber physical systems: Design challenges,” in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 363–369.
- [128] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, A. Roederer et al., “Challenges and research directions in medical cyber-physical systems,” *Proceedings of the IEEE*, vol. 100, no. 1, pp. 75–90, 2011.
- [129] D. Serpanos, “The cyber-physical systems revolution,” *Computer*, vol. 51, no. 3, pp. 70–73, 2018.
- [130] E. R. Griffor, C. Greer, D. A. Wollman, and M. J. Burns, “Framework for cyber-physical systems: Volume 1, overview,” NIST, Tech. Rep., 2017.
- [131] F. Xia, L. T. Yang, L. Wang, and A. Vinel, “Internet of things,” *International Journal of Communication Systems*, vol. 25, no. 9, p. 1101, 2012.
- [132] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [133] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta, “Review of iot applications in agro-industrial and environmental fields,” *Computers and Electronics in Agriculture*, vol. 142, pp. 283–297, 2017.
- [134] I. A. Lakhari, G. Jianmin, T. N. Syed, F. A. Chandio, N. A. Buttari, and W. A. Qureshi, “Monitoring and control systems in agriculture using intelligent sensor techniques: A review of the aeroponic system,” *Journal of Sensors*, vol. 2018, 2018.
- [135] J. J. Segura-Juárez, D. Cuesta-Frau, L. Samblas-Pena, and M. Aboy, “A microcontroller-based portable electrocardiograph recorder,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 9, pp. 1686–1690, 2004.
- [136] D. Lucani, G. Cataldo, J. Cruz, G. Villegas, and S. Wong, “A portable eeg monitoring device with bluetooth and holter capabilities for telemedicine applications,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2006, pp. 5244–5247.
- [137] S. Borromeo, C. Rodriguez-Sanchez, F. Machado, J. A. Hernandez-Tamames, and R. de la Prieta, “A reconfigurable, wearable, wireless eeg system,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 1659–1662.
- [138] R. Gupta, J. Bera, and M. Mitra, “Development of an embedded system and matlab-based gui for online acquisition and analysis of eeg signal,” *Measurement*, vol. 43, no. 9, pp. 1119–1126, 2010.
- [139] W.-C. Huang, S.-H. Hung, J.-F. Chung, M.-H. Chang, L.-D. Van, and C.-T. Lin, “Fpga implementation of 4-channel ica for on-line eeg signal separation,” in *2008 IEEE Biomedical Circuits and Systems Conference*. IEEE, 2008, pp. 65–68.
- [140] C.-T. Lin, Y.-C. Chen, T.-Y. Huang, T.-T. Chiu, L.-W. Ko, S.-F. Liang, H.-Y. Hsieh, S.-H. Hsu, and J.-R. Duann, “Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver’s drowsiness detection and warning,” *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 5, pp. 1582–1591, 2008.
- [141] C.-T. Lin, L.-W. Ko, J.-C. Chiou, J.-R. Duann, R.-S. Huang, S.-F. Liang, T.-W. Chiu, and T.-P. Jung, “Noninvasive neural prostheses using mobile and wireless eeg,” *Proceedings of the IEEE*, vol. 96, no. 7, pp. 1167–1183, 2008.
- [142] C.-T. Lin, C.-J. Chang, B.-S. Lin, S.-H. Hung, C.-F. Chao, and I.-J. Wang, “A real-time wireless brain-computer interface system for drowsiness detection,” *IEEE transactions on biomedical circuits and systems*, vol. 4, no. 4, pp. 214–222, 2010.
- [143] L. Brown, J. van de Molengraft, R. F. Yazicioglu, T. Torfs, J. Penders, and C. Van Hoof, “A low-power, wireless, 8-channel eeg monitoring headset,” in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 2010, pp. 4197–4200.
- [144] A. Page, C. Sagedy, E. Smith, N. Attaran, T. Oates, and T. Mohsenin, “A flexible multichannel eeg feature extractor and classifier for seizure

- detection,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 2, pp. 109–113, 2014.
- [145] B.-G. Lee, B.-L. Lee, and W.-Y. Chung, “Mobile healthcare for automatic driving sleep-onset detection using wavelet-based eeg and respiration signals,” *Sensors*, vol. 14, no. 10, pp. 17915–17936, 2014.
- [146] J. A. Mercado, J. Herrera, A. d. J. Pansza, and J. Gutierrez, “Embedded eeg recording module with active electrodes for motor imagery brain-computer interface,” *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 503–510, Feb 2016.
- [147] T. Gao and D. White, “A next generation electronic triage to aid mass casualty emergency medical response,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2006, pp. 6501–6504.
- [148] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, “A review of wearable sensors and systems with application in rehabilitation,” *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 21, 2012.
- [149] S. Majumder, T. Mondal, and M. Deen, “Wearable sensors for remote health monitoring,” *Sensors*, vol. 17, no. 1, p. 130, 2017.
- [150] A. H. Kronbauer, H. C. da Luz, and J. Campos, “Mobile security monitor: A wearable computing platform to detect and notify falls,” *IEEE Latin America Transactions*, vol. 16, no. 3, pp. 957–965, March 2018.
- [151] A. Darwish and A. E. Hassanien, “Wearable and implantable wireless sensor network solutions for healthcare monitoring,” *Sensors*, vol. 11, no. 6, pp. 5561–5595, 2011.
- [152] L. S. Wong, S. Hossain, A. Ta, J. Edvinsson, D. H. Rivas, and H. Naas, “A very low-power cmos mixed-signal ic for implantable pacemaker applications,” *IEEE Journal of solid-state circuits*, vol. 39, no. 12, pp. 2446–2456, 2004.
- [153] S. D. Chede and K. D. Kulat, “Design overview of processor based implantable pacemaker,” *Journal of Computers*, vol. 3, no. 8, pp. 49–57, 2008.
- [154] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, “Cyber-physical systems: A new frontier,” in *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)*. IEEE, 2008, pp. 1–9.
- [155] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, “Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses,” in *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 2008, pp. 129–142.
- [156] L. Kristen. (2019) Delivering next-generation ai experiences for the 5g world. [Online]. Available: <https://www.arm.com/company/news/2019/05/delivering-next-generation-ai-experiences-for-the-5g-world>
- [157] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, S. Moderator-Ravi, and S. Moderator-Ravi, “Security as a new dimension in embedded system design,” in *Proceedings of the 41st annual Design Automation Conference*. ACM, 2004, pp. 753–760.
- [158] S. Ravi, A. Raghunathan, and S. Chakradhar, “Tamper resistance mechanisms for secure embedded systems,” in *17th International Conference on VLSI Design. Proceedings*. IEEE, 2004, pp. 605–611.
- [159] S. Ravi, S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, “Security in embedded systems: Design challenges,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 3, pp. 461–491, 2004.
- [160] D. D. Hwang, P. Schaumont, K. Tiri, and I. Verbauwhede, “Securing embedded systems,” *IEEE Security & Privacy*, no. 2, pp. 40–49, 2006.
- [161] T. Huffmire, B. Brotherton, T. Sherwood, R. Kastner, T. Levin, T. D. Nguyen, and C. Irvine, “Managing security in fpga-based embedded systems,” *IEEE Design & Test of Computers*, vol. 25, no. 6, pp. 590–598, 2008.
- [162] A. Ukil, J. Sen, and S. Koilakonda, “Embedded security for internet of things,” in *2011 2nd National Conference on Emerging Trends and Applications in Computer Science*. IEEE, 2011, pp. 1–6.
- [163] D. Papp, Z. Ma, and L. Buttyan, “Embedded systems security: Threats, vulnerabilities, and attack taxonomy,” in *2015 13th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2015, pp. 145–152.
- [164] M.-K. Yoon, S. Mohan, J. Choi, M. Christodorescu, and L. Sha, “Learning execution contexts from system call distribution for anomaly detection in smart embedded system,” in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM, 2017, pp. 191–196.
- [165] A. Damien, M. Fumey, E. Alata, M. Kaâniche, and V. Nicomette, “Anomaly based intrusion detection for an avionic embedded system,” SAE Technical Paper, Tech. Rep., 2018.
- [166] A. Flores-Vergara, E. Garcia-Guerrero, E. Inzunza-González, O. López-Bonilla, E. Rodríguez-Orozco, J. Cardenas-Valdez, and E. Tlelo-Cuautle, “Implementing a chaotic cryptosystem in a 64-bit embedded system by using multiple-precision arithmetic,” *Nonlinear Dynamics*, vol. 96, no. 1, pp. 497–516, 2019.

A Literature Review on Embedded Systems

L. De Micco, *Senior Member, IEEE*, F. Vargas, *Senior Member, IEEE*, and P. Fierens, *Senior Member, IEEE*

Abstract—Embedded system design is crucial for the development of industry, technology, and science, and it is an area that has significantly grown in recent years throughout Latin America, both in academia and in industry. Embedded System (ES) refers to electronic equipment with a computing core which, unlike a personal computer, is designed to meet a specific function and is usually optimized to satisfy strict requirements of processing time, reliability, power consumption, size, and cost. With the advancement of research on the domains of Internet of Things (IoT), Computing Edge (CE) and Cloud Computing (CC) and their endless application possibilities, ESs have gained new roles and assumed unquestionable importance in our daily lives. In this article, we make a brief review of the relevant literature in the area of embedded systems in the Latin America region during recent years.

Index Terms—Embedded systems, Computer Architecture, FPGA, ASIC, HDL, DSP, RTOS, Embedded Software, Digital Signal Processing, Wireless Sensor Networks, Communications, Cryptography, Industry 4.0, Robotics, IoT, Agrotechnology, Bio-engineering.

I. INTRODUÇÃO

O termo sistemas embarcados (SEs) refere-se a equipamentos eletrônicos que incluem processamento de dados, às vezes em tempo real, e projetados para desempenhar uma função dedicada, diferentemente dos computadores de uso geral que cobrem uma ampla gama de necessidades [1], [2]. Hoje, em nossa vida cotidiana, estamos cercados por muitos SEs, como um MP3 player, um telefone celular ou mesmo o sistema de controle de uma usina nuclear.

O cérebro de um SE é tipicamente um microcontrolador ou um microprocessador, embora os dados também possam ser processados por uma matriz programável de portas lógicas (FPGA, Field Programmable Gate Array), um processador de sinal digital (DSP, Digital Signal Processor), um circuito integrado para aplicações específicas (ASIC, Application-Specific Integrated Circuit), em qualquer caso, milhões de transistores são combinados em um único chip para implementação (VLSI, Very large-scale integration). Eles também são caracterizados por ter uma ótima interação com o meio ambiente através de sensores e atuadores. Seu projeto é otimizado para atender a requisitos rigorosos de tempo de processamento, confiabilidade, consumo de energia, tamanho e custo.

Um diagrama de blocos simplificado de um sistema incorporado é mostrado na Fig. 1. O ponto central de um SE é a interação com o ambiente físico, que ocorre através de sensores que medem suas variáveis e atuadores que atuam nele. Para interpretar variáveis sensíveis e tomar decisões sobre como agir, um sistema embarcado precisa de um ou mais processadores que possam ser fisicamente implementados, por exemplo, em microcontroladores, FPGAs ou ASICs. Em muitas situações, os SEs são sistemas autônomos alimentados por baterias, que por sua vez podem ser recarregadas por

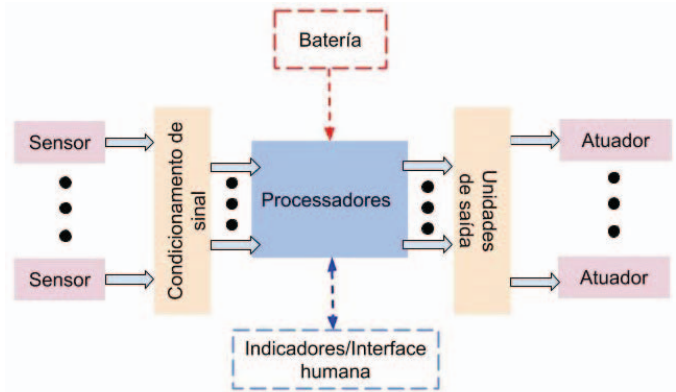


Fig. 1. Diagrama de blocos simplificado de um sistema incorporado. Um ou mais processadores exercem ações no ambiente físico por meio de atuadores, baseando suas decisões em variáveis medidas por sensores. É comum que os SEs sejam sistemas autônomos, alimentados por bateria e, às vezes, podem incorporar uma interface humana.

energia renovável. Uma interface homem-computador nem sempre está presente em um sistema embarcado, muitas vezes sendo reduzida a uma expressão mínima, como um conjunto de indicadores luminosos.

A descrição dos sistemas embarcados esboçada nos dois parágrafos anteriores não é exclusiva. De fato, uma pesquisa por “O que é um sistema embarcado?” em qualquer mecanismo de pesquisa na Internet, haverá um grande número de resultados com as mais variadas respostas. Muitas respostas incluem um levantamento histórico das origens dos SEs. Como marco fundamental, pode-se citar o computador de navegação localizado nas missões Apollo (veja, por exemplo, a referência na Wikipedia [3] ou o vídeo *The History of Embedded Systems* [4], que fazia parte de uma palestra plenária na Conferência de Sistemas Embarcados de 2008). Certamente, um sistema como esse atende aos termos da definição dada acima.

Na análise histórica, é difícil localizar a origem da expressão inglesa “embedded systems” para os sistemas que estamos discutindo neste trabalho. No entanto, acreditamos que a expressão sistema de computador embarcado tenha sido anterior. Em relação a isso, uma definição de meados da década de 1970 [5] (veja também a definição do mesmo período em [6])

“An embedded computer system is a computer system that is integral to an electromechanical system such as a combat weapons system, tactical system, aircraft, ship, missile, spacecraft, certain command and control systems, civilian systems such as a rapid transit system, and the like. [...] The key attributes of an embedded computer system are:

a. It is a computer system that is physically incor-

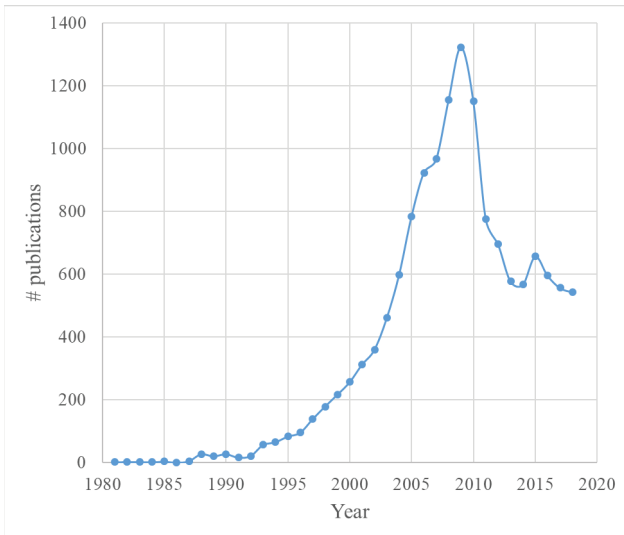


Fig. 2. Número de publicações em sistemas embarcados no IEEE em função do ano.

porated into a larger system whose primary function is not data processing.

b. It is integral to such a larger system from a design, procurement or operations- viewpoint.

c. Its outputs generally include information, computer control signals and-computer data."

Essa definição vai ao cerne da origem da expressão: um sistema embarcado é fisicamente incorporado e é parte integrante de um sistema maior. Quase 50 anos se passaram desde a nomeação e o uso do SE evoluiu nesse período. Mesmo assim, o fato fundamental de que um SE não é um computador genérico, mas um sistema específico permanece válido.

Continuando com a análise histórica, propomos como métrica indireta, certamente discutível, da relevância da área de sistemas embarcados para o número de artigos e periódicos de conferências no IEEE que contenham a expressão "sistema embarcado", seja no título, no resumo ou entre as palavras-chave. A figura 1 mostra a evolução do referido número. O primeiro artigo aparece há quase 40 anos, em 1981, e o número de artigos anuais não cai abaixo de 500 há mais de 15 anos.

Observe que a escolha das palavras-chave "sistemas embarcados" para a pesquisa é arbitrária. Hoje, existem muitos sistemas embarcados que não são chamados por esse nome. Por exemplo: aplicativos da Internet das Coisas (IoT) são baseados em sistemas incorporados; Telefones celulares inteligentes e TVs inteligentes são sistemas embarcados. No entanto, não incluímos "IoT", "smartphone" ou "smart TV" na pesquisa. É possível, portanto, que a redução em relação a 2010 observada na Fig. 1 não seja o resultado de uma diminuição no trabalho na área de sistemas embarcados, mas de uma mudança em quais palavras são usadas para se referir a essa área.

Entre as muitas definições não técnicas, e certamente parciais, de um sistema embarcado que podem ser encontradas existe a seguinte [7]: "An embedded system is one you don't even know is there... until it stops working." . Essa definição aponta para a onipresença do SE. Certamente, os sistemas embarcados integram conceitos de várias áreas diferentes e

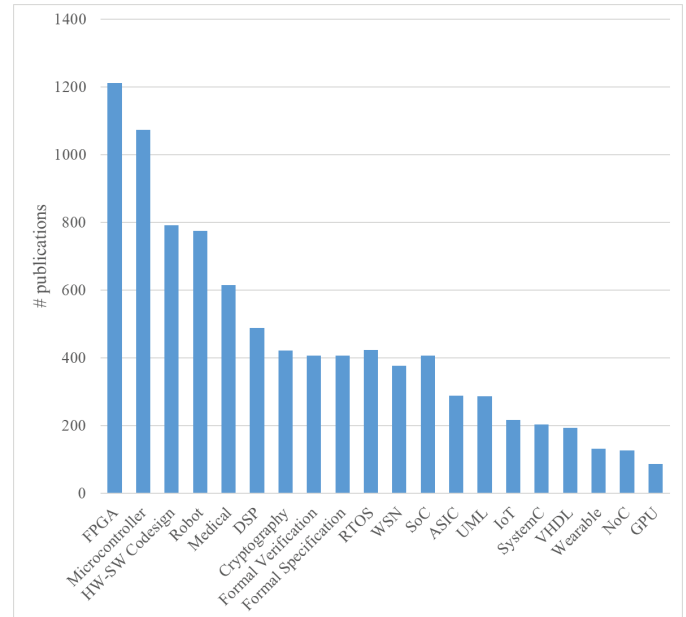


Fig. 3. Número de publicações sobre sistemas embarcados no IEEE até outubro de 2019, agrupadas por área.

têm aplicações em muitos problemas tecnológicos.

Neste artigo, revisamos brevemente várias áreas envolvidas e as aplicações de SE. Como referência, usamos um indicador semelhante ao anterior. Em particular, na Figura 3, contamos todos os artigos (publicado no IEEE até o momento da redação do presente, outubro de 2019) que tratam de SE e de vários tópicos mencionados ao longo deste trabalho.

Então: o que é um sistema embarcado? Além da definição dada no início dessa seção e da breve discussão anterior, esperamos que, ao chegar ao final deste artigo, o leitor possa responder a essa pergunta por si mesmo. Neste trabalho, revisamos brevemente a literatura sobre sistemas embarcados nos últimos anos. Na Seção II, citamos alguns dos trabalhos mais relevantes em hardware específico para SEs. O projeto de software para SE também ocupa um lugar relevante na literatura e o revisamos na Seção III. Os sistemas embarcados têm aplicações em sistemas de comunicação. Além disso, existem sistemas e protocolos dedicados projetados especificamente para a comunicação entre sistemas incorporados. O relacionamento dos SEs com as comunicações são relatadas na Seção IV. A gama de aplicações de sistemas embarcados é muito ampla e fazemos um breve resumo de algumas das áreas cobertas na Seção V. Fechamos este resumo com algumas conclusões e projeções futuras para área, na Seção VI.

II. HARDWARE PARA SISTEMAS EMBARCADOS

Os sistemas embarcados são projetados para executar tarefas dedicadas. Isso implica que, embora exista plataformas para o desenvolvimento de SEs, o projeto de hardware associado é sempre uma parte importante dele.

Existe uma vasta literatura sobre o desenvolvimento de arquiteturas para SE [1], [8]–[14], variando desde o uso de dispositivos genéricos como microcontroladores, System on Chip (SoC) [15], [16], FPGA, arquiteturas multicore [17], até

o desenvolvimento de arquiteturas específicas para otimizar a execução e dar suporte ao software que será executado nelas [10], [18]. É o caso de sistemas em tempo real que requerem plataformas com alta previsibilidade para permitir a análise estática do pior tempo de execução [19]–[21].

O núcleo de um sistema embarcado é geralmente um microcontrolador [22]–[24]. Esses processadores geralmente contêm periféricos e sistemas específicos para tarefas nas quais serão usados. A especificidade das tarefas a serem executadas por um SE torna a lógica programável, como os FPGAs, um elemento frequente deles [25]. Quando o volume de produção ou os requisitos da aplicação o justificam, é necessário o desenvolvimento de *Application-Specific Integrated Circuits* (ASICs) se faz necessário [26]. Tradicionalmente, a eficiência do ASIC é geralmente considerada superior à dos FPGAs, tanto na área utilizada quanto no consumo de energia. No entanto, estas diferenças para determinadas aplicações foi consideravelmente reduzida nos últimos anos [27].

As ferramentas de Desenho Assistido por Computador (CAD) desempenham um papel crucial e inegável em praticamente todos os aspectos do projeto de circuitos, à medida que os circuitos se tornam mais complexos. Essas ferramentas variam desde o design de circuitos integrados ao desenvolvimento de placas de circuito impresso.

Para o desenvolvimento de sistemas específicos, é comum o uso de *Hardware Description Languages* (HDLs) como Verilog, VHDL (Very High Speed Integrated Circuit Hardware Description Language) [28]–[30] e SystemC [31]–[36].

Em muitas aplicações, os sistemas embarcados devem executar intenso processamento de sinal digital. Nesses casos, o uso de processadores de sinais digitais (DSP – *Digital Signal Processor*) [37]–[39] e as unidades de processamento gráfico (GPU – *Graphics Processing Units*) são alternativas comuns [40]–[43], pois são otimizadas para cálculos com valores de ponto flutuante. No entanto, quando é necessário aumentar a velocidade de processamento e aumentar a taxa de transferência do sistema, uma possibilidade é incorporar mais de um núcleo no mesmo chip. As plataformas MPSoC (Multiprocessor Systems-on-Chip) geralmente integram processadores de uso geral, como um núcleo ARM, com núcleos de uso específico, como DSP ou GPU [44], de modo que o chip como um todo maximize o desempenho. Portanto, há uma forte tendência no uso de processamento paralelo, que possui aspectos de computação e comunicação [45], [46]. Nesse novo cenário, as interconexões internas entre os núcleos dos chips assumem uma relevância extremamente importante, pois são o principal gargalo no desempenho desses sistemas. Interconexões dedicadas e barramentos compartilhados apresentam grandes problemas à medida que o número de elementos integrados em um chip aumenta. É aqui que a arquitetura de comunicação *Network on chip* (NoC) [47]–[49] surge, visando comunicar processadores, memórias e outros dispositivos personalizados por meio de pacotes de comutação em vez de trocar mensagens ou palavras, aumentando o desempenho e a largura da banda. No entanto, essa arquitetura de comutação de pacotes para comunicação interna de chips exige recursos adicionais, como buffers de fila e nós de processamento de pacotes, que aumentam o consumo de energia e espaço.

Para resolver esse problema, foram propostas soluções para substituir roteadores eletrônicos por comutadores de pacotes ópticos ou comutadores de pacotes fotônicos [50]–[52].

Às vezes, os SEs geralmente funcionam como sistemas autônomos alimentados por baterias, o desenvolvimento de sistemas de baixa potência é de grande relevância [53]–[55]. Por outro lado, a literatura explorou o uso de fontes alternativas de energia para alimentar-los [56], , como energia solar [57], [58] ou radiofrequência [59].

SEs se caracterizam por processar informações sensíveis e agir em diversos sistemas. Nesse sentido, o uso de sensores e atuadores é parte integrante de sistemas embarcados. Geralmente os periféricos com os quais eles interagem são especializados, como dispositivos mecânicos e químicos, por isso requerem atuadores e sensores específicos para cada aplicação. Os sensores usados variam de sensores de luz, sensores de temperatura, acelerômetros, etc., até o desenvolvimento de novos sensores inteligentes específicos para aplicações como redes inteligentes, cidades inteligentes ou monitoramento estrutural da saúde, que se concentram no baixo consumo de energia e são capazes de coletar informações e se comunicar [60]–[62].

Os conversores analógico-digitais (ADC) e digital-analógicos (DAC) desempenham um papel fundamental nesse tipo de sistema, pois são indispensáveis para processar os sinais digitalmente e atuar no meio. A conversão analógico-digital implica primeiro uma amostragem do sinal no tempo e depois uma quantificação em amplitude, onde os requisitos em frequência e resolução da amostragem são os principais determinantes do tipo de conversor a ser usado. Para conversões de alta resolução, é utilizado o ADC sigma-delta, uma vez que a modulação sigma-delta do sinal subamostrado atua como um filtro passa-alto para quantificação de ruído, reduzindo significativamente a potência de ruído na faixa de frequência de interesse. O resultado é um maior número efetivo de bits (ENOB – *Effective Number of Bits*) [63]. Atualmente, os conversores sigma-delta podem ter resoluções de 16 a 24 bits, mas para sinais de frequências abaixo de 1 MHz, aproximadamente. Para aplicativos que exigem alta velocidade, os conversores Flash são frequentemente usados. Embora esse tipo de ADC seja o mais rápido, ele possui baixa resolução (geralmente até 8 bits) e alto consumo de energia devido ao grande número de componentes necessários. Um tópico de estudo é o desenvolvimento de arquiteturas de conversoras Flash que reduzem o uso de componentes e o consumo de energia [64], [65]. Quando um baixo consumo de energia é importante, uma opção é o sucessivo conversor de aproximação (SAR – *Successive Approximation Register*), que possui baixo consumo de energia graças à simplicidade de sua estrutura. Esses conversores variam de média a alta resolução (entre 8 e 18 bits) e fornecem uma taxa de amostragem de até 5 Msps. Graças ao fato de serem quase inteiramente digitais, elas são dimensionadas de acordo com o avanço da tecnologia microeletrônica. Por esse motivo, entre seus parâmetros, a tecnologia usada em nanômetros é geralmente especificada. A redução tecnológica dos transistores também permite reduzir o tamanho do circuito integrado, aumentar a velocidade e diminuir o consumo de energia. Atualmente, comprimentos

de canal inferiores a 8 nm podem ser encontrados. A diminuição do canal é objeto de estudo, pois provoca problemas tecnológicos, como o uso de dielétricos de alta constante dielétrica, o controle da degradação por efeitos de canal curto (SCE - *Short Channel Effect*) e a dificuldade de alcançar alta reprodutibilidade [66]–[68].

O projeto de um SE implica a integração de hardware heterogêneo, já que seus componentes geralmente vêm de uma ampla variedade de fornecedores, onde cada fornecedor é compatível com uma linguagem específica, formalismo de modelagem e estrutura de simulação. As plataformas virtuais são um poderoso suporte para o desenvolvimento e validação de hardware incorporado, principalmente quando o hardware usado é heterogêneo [69], [70].

III. SOFTWARE PARA SISTEMAS EMBARCADOS

Os avanços nesses sistemas geraram uma demanda pelo desenvolvimento da arquitetura livre do conjunto de instruções (ISA, *Instruction Set Architecture*), em 1994 surgiu o *SPARC V8*, no 2000 *OpenRISC* e no 2010 *RISC V* [71]. Este último teve o apoio de empresas de semicondutores e atualmente foram desenvolvidos vários núcleos que o utilizam [72]. Diferentemente do software de uso geral, a principal função do software incorporado é permitir a interação o sistemas embarcados com o mundo físico, portanto ele deve estar vinculado ao mundo e considerar questões como sincronização, confiabilidade, robustez e consumo de energia. O desenvolvimento de software específico para uso EM SE é uma tarefa complexa. Por um lado, os sistemas embarcados devem apresentar características sofisticadas e confiáveis, e por outro, apresentar baixo custo e reduzido tempo de desenvolvimento para comercialização [73].

Uma das alternativas mais frequentemente usadas para o desenvolvimento de software embarcado são os modelos. Os modelos representam uma abstração do software incorporado e do ambiente em que opera, capturando simultaneamente os requisitos e o projeto do aplicativo [74]–[77]. Algumas linguagens de modelagem incluem diagramas de estado, como *StateCharts* de *Unified Modeling Language* (UML) [78], ou a extensão *Stateflow* do *Simulink Matlab* [79].

Dentro de aplicações de mesmo tipo, geralmente o código desses SE é muito semelhante, desse fato surgem conhecidos como *Frameworks*. Um *Framework* é uma ferramenta de programação especializada cujo objetivo é facilitar o desenvolvimento de aplicações e, assim, reduzir o tempo e os custos de projeto. Um *Framework* geralmente está acima do sistema operacional e é um conjunto de funcionalidades organizadas em bibliotecas específicas [80].

Em certas áreas de aplicação, sistemas embarcados têm requisitos de confiabilidade muito altos, como software para aeronaves comerciais. Existe uma vasta literatura sobre desenvolvimento de software que atende a esses requisitos [81], [82] e à verificação formal do código [83]–[88].

Ocasionalmente, SEs são projetados com o requisito de “sistemas de tempo real” [89], [90]. Em particular, os núcleos computacionais costumam usar sistemas operacionais de tempo real (RTOS - *Real Time Operating Systems*) [91],

[92], tais como QNX [93], WinCE [94], Vxworks [95] e FreeRTOS [96], entre muitos outros.

A lógica de certas aplicações de SE torna natural o uso de programação orientada a eventos ou *event-driven programming* (ver, por exemplo, [97], [98]). Nesses casos, a execução do aplicativo ocorre ao receber um evento, permanecendo em repouso desde que ele não receba um estímulo assíncrono que o torne o objeto ativo.

Embora exista processadores cada vez mais potentes, existe limitações à capacidade de processamento de seus núcleos computacionais. Isso leva à existência de várias estratégias para adaptar os algoritmos existentes para uso em SEs. Em particular, desenvolvimentos específicos de algoritmos de processamento de sinal [99] e aprendizado de máquina [90], [100], [101] são destacados.

A complexidade de um sistema embarcado projetado para uma tarefa específica nos obriga a considerar simultaneamente aspectos de software e hardware durante o estágio de desenvolvimento. Assim sendo, co-design [102]–[104] e a co-simulação de hardware e software são estabelecidos na área [105], [106].

IV. COMUNICAÇÕES

SEs estão na base de muitos sistemas de comunicação. Em particular, muitos dos projetos de rádio definidos por software (SDR, *Software Defined Radio*) podem ser considerados como sistemas embarcados [107].

As redes de sensores sem fio (WSN, *Wireless Sensor Networks*) são formadas por nós que são sistemas embarcados. O desenvolvimento de protocolos de comunicação e roteamento eficiente nesse tipo de rede possui uma literatura muito ampla, que não pode ser totalmente abordada neste breve resumo [108], [109].

A segurança nos sistemas de comunicação é essencial [110]–[112]. Os algoritmos criptográficos geralmente são computacionalmente muito caros; portanto, sua adaptação a sistemas embarcados tem chamado a atenção de muitos pesquisadores na área [113]–[115].

V. ALGUMAS APLICAÇÕES

Os avanços nos sistemas embarcados aplicados aos produtos de consumo em massa mudaram significativamente a maneira como nos conectamos com o mundo ao nosso redor e melhoramos a qualidade de vida. Dispositivos como telefones, tablets portáteis e relógios inteligentes permitem acesso permanente a informações [116]. Eles também revolucionaram dispositivos do cotidiano, como televisores, sistemas de áudio e eletrodomésticos.

Os carros modernos usam vários sistemas embarcados para controlar diferentes características, como sistema de ar condicionado, painel de instrumentos, transmissão, motor e vários aspectos de segurança [117]. A complexidade é tal que a comunicação interna desses SEs é, em si mesma, uma área de estudo [118]. Os sistemas embarcados também têm aplicação em outros tipos de transporte, como em sistemas ferroviários [119]. Nos últimos anos, seu uso em veículos autônomos de diferentes tipos, como carros [120], navios

e submarinos [121] e veículos aéreos [122] tornou-se mais relevante.

SEs ocupam lugar de grande relevância na indústria, especialmente no controle da produção [123], [124]. Os sistemas incorporados também são uma parte essencial dos sistemas robóticos industriais [123], [125]. Mais importante é o papel do SE na chamada quarta revolução industrial ou Indústria 4.0 [126]. Na base desse paradigma, há um grande número de sistemas ciber-físicos interconectados. O conceito de sistema ciber-físico (CPS, *Cyber-Physical System*), uma abstração da integração entre sistemas físicos e de computação, é uma evolução natural da idéia de sistema embarcado, mas mais orientada para a conexão de rede [127]–[129].

Uma ideia relacionada é a da Internet das Coisas (IoT, *Internet of Things*), ao ponto de o CPS e a IoT serem usados de forma intercambiável por alguns autores [130]. A IoT traz a promessa de um mundo com conexão e controle ainda maiores de cada uma das características de nossas casas, carros, escritórios, etc. A revolução esperada da IoT não seria possível sem os sistemas embarcados nos quais se baseia [131], [132]. Isso implica que os sistemas embarcados integram recursos de rede, recursos de controle de acesso baseado em função, recursos de controle remoto ou recursos relacionados.

Uma das aplicações mais importantes de sistemas embarcados, especialmente na América Latina, é o agronegócio, tanto no monitoramento remoto quanto no controle de plantações e nos processos de processamento subsequentes [133], [134].

A aplicação de sistemas embarcados à bioengenharia é comum: sistemas de eletrocardiografia portátil (ECG) [135]–[138], sistemas de eletrocefalograma (EEG) e interface computador-cérebro (BCI, *Brain-Computer Interface*) [139]–[146] e uma variedade de sensores portáteis incorporados (sensores vestíveis ou *wearable sensors*) [147]–[150]. Em particular, os SEs têm um lugar importante na saúde das pessoas, por exemplo, como sistemas de sensores implantáveis [151]. Um dos exemplos mais importantes são os marcapassos cardíacos [77], [152], [153]. Nos últimos anos, uma maior atenção tem sido dada à segurança desses sistemas embarcados contra ataques [128], [154], [155].

O avanço no desenvolvimento de SEs está fortemente ligado ao avanço da tecnologia, uma vez que as principais características de um sistema embarcado são o baixo custo e o baixo consumo de energia. Assim, as inovações nessas direções são as que mais afetam estes sistemas. No entanto, existe uma ampla gama de áreas da eletrônica que também influencia, tanto os dispositivos que fazem parte de sua estrutura básica quanto os microcontroladores, microprocessadores, FPGA e DSP [156], para avanços, por exemplo, em criptografia, uma vez que a segurança é essencial nesse tipo de sistema [113], [157]–[166].

VI. CONCLUSÕES

Neste artigo, revisamos brevemente alguns dos tópicos comumente relacionados a sistemas embarcados. O leitor pode ficar com a impressão de que "Sistemas embarcados" é uma expressão usada como um título geral que cobre muitas áreas. A realidade é que os SEs fazem parte de uma infinidade de

sistemas e estão presentes em grande parte dos desenvolvimentos tecnológicos. Por isso, nomeamos neste artigo áreas como arquitetura de computadores, comunicações, verificação de código, sistemas de tempo real, robótica, agrotecnologia, bioengenharia, produtos de consumo em massa, IoT, etc. Ao fazer essa lista, não queremos dizer que todas essas áreas sejam apenas parte da grande área de SEs. Ao contrário, queremos apenas explicar que os sistemas embarcados são parte integrante de um grande número de desenvolvimentos científicos e tecnológicos. Por esse motivo, o projeto de um SE é um campo que nos últimos anos cresceu notavelmente em toda a América Latina, tanto na academia quanto na indústria. Portanto, a necessidade de falar sobre sistemas embarcados em si pode ser discutida. Acreditamos que é necessário. Assim como os circuitos eletrônicos fazem parte de todas as aplicações mencionadas e faz sentido falar do projeto de circuitos eletrônicos em si, como uma área bem estabelecida, também é necessário falar do desenvolvimento de sistemas embarcados como uma área bem definida. Existem inúmeros critérios para o projeto de hardware, software, protocolos de comunicação e estratégias de implementação de SEs que formam um corpo de conhecimento de grande relevância. Essa relevância é observada, não apenas nas citações deste artigo, mas também na realização periódica de importantes conferências nacionais e internacionais, como a International Conference on Embedded Systems and Applications (EMSA), a IEEE International Conference on Embedded Software and Systems (ICCESS), a ACM SIGBED International Conference on Embedded Software (EMSOFT), o Congreso Argentino de Sistemas Embebidos (CASE), o Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC) e o IEEE Latin American Test Symposium (LATS), entre outras. Possivelmente, nos próximos anos, o impulso no desenvolvimento da área de SEs virá da Indústria 4.0, em particular como parte dos conceitos de Sistema Ciber-físico e Internet das Coisas. Essas novas abstrações orientam ainda mais os sistemas embarcados em relação à conectividade e os colocam, como na antiga definição de 1975 citada na Seção I, como partes "physically incorporated into a larger system" cujo objetivo não é o mero processamento de dados, mas sim o funcionamento de uma sociedade e seus meios de produção.

AGRADECIMENTO

Deseamos agradecer especialmente el aporte y la discusión con los miembros del comité editorial invitado de este número especial sobre sistemas embebidos.

REFERÊNCIAS

- [1] T. Noergaard, *Embedded systems architecture: a comprehensive guide for engineers and programmers*. Newnes, 2012.
- [2] P. Marwedel, *Embedded system design*. Springer, 2006, vol. 1.
- [3] Wikipedia, "Embedded system," https://en.wikipedia.org/wiki/Embedded_system, accessed: 2019-10-09.
- [4] E. Times, "History of embedded system," <https://www.youtube.com/watch?v=1hX4urTFXr0>, accessed: 2019-10-09.

- [5] A. Kossiakoff, T. Sleight, E. Prettyman, J. Park, and P. Hazan, "Dod weapon systems software management study," Johns Hopkins University, Applied Physics Laboratory, Tech. Rep., 1975.
- [6] D. L. Parnas, "Use of abstract interfaces in the development of software for embedded computer systems," Naval Research Laboratory, Tech. Rep., 1977.
- [7] M. Maxfield, "What is an embedded system?" https://www.eetimes.com/author.asp?section_id=216&doc_id=1322671, accessed: 2019-10-09.
- [8] B. P. Dave, G. Lakshminarayana, and N. K. Jha, "Cosyn: Hardware-software co-synthesis of embedded systems," in *Proceedings of the 34th annual Design Automation Conference*. ACM, 1997, pp. 703–708.
- [9] T. S. Hall and J. O. Hamblen, "Using system-on-a-programmable-chip technology to design embedded systems," *International Journal of Computer Architectures*, vol. 13, no. 6, pp. 142–152, 2006.
- [10] M. Schoeberl, "A java processor architecture for embedded real-time systems," *Journal of Systems Architecture*, vol. 54, no. 1-2, pp. 265–286, 2008.
- [11] P. Garcia, K. Compton, M. Schulte, E. Blem, and W. Fu, "An overview of reconfigurable hardware in embedded systems," *EURASIP Journal on Embedded Systems*, vol. 2006, no. 1, p. 056320, 2006.
- [12] R. Kamal, *Embedded systems: architecture, programming and design*. Tata McGraw-Hill Education, 2011.
- [13] J. G. Tong, I. D. Anderson, and M. A. Khalid, "Soft-core processors for embedded systems," in *2006 International Conference on Microelectronics*. IEEE, 2006, pp. 170–173.
- [14] S. S. Bhattacharyya, E. F. Deprettere, R. Leupers, and J. Takala, *Handbook of signal processing systems*. Springer, 2018.
- [15] J. Teich, "Hardware/software codesign: The past, the present, and predicting the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1411–1430, 2012.
- [16] N. Hou, X. Yan, and F. He, "A survey on partitioning models, solution algorithms and algorithm parallelization for hardware/software co-design," *Design Automation for Embedded Systems*, pp. 1–21, 2019.
- [17] G. Kornaros, *Multi-core embedded systems*. CRC Press, 2010.
- [18] J. Balfour, W. Dally, D. Black-Schaffer, V. Parikh, and J. Park, "An energy-efficient processor architecture for embedded systems," *IEEE Computer Architecture Letters*, vol. 7, no. 1, pp. 29–32, 2008.
- [19] L. Thiele and R. Wilhelm, "Design for timing predictability," *Real-Time Systems*, vol. 28, no. 2-3, pp. 157–177, 2004.
- [20] M. Schoeberl, S. Abbaspour, B. Akesson, N. Audsley, R. Capasso, J. Garside, K. Goossens, S. Goossens, S. Hansen, R. Heckmann *et al.*, "T-crest: Time-predictable multi-core architecture for embedded systems," *Journal of Systems Architecture*, vol. 61, no. 9, pp. 449–471, 2015.
- [21] C. Cullmann, C. Ferdinand, G. Gebhard, D. Grund, C. Maiza, J. Reineke, B. Triquet, and R. Wilhelm, "Predictability considerations in the design of multi-core embedded systems," *Proceedings of Embedded Real Time Software and Systems*, pp. 36–42, 2010.
- [22] S. Heath, *Embedded systems design*. Elsevier, 2002.
- [23] R. Toulson and T. Wilmshurst, *Fast and effective embedded systems design: applying the ARM mbed*. Newnes, 2016.
- [24] J. K. Peckol, *Embedded systems: a contemporary design tool*. John Wiley & Sons, 2019.
- [25] R. Sass and A. G. Schmidt, *Embedded systems design with platform FPGAs: principles and practices*. Morgan Kaufmann, 2010.
- [26] W. J. Dally, J. Balfour, D. Black-Schaffer, J. Chen, R. C. Harting, V. Parikh, J. Park, and D. Sheffield, "Efficient embedded computing," *Computer*, vol. 41, no. 7, pp. 27–32, 2008.
- [27] E. Nurvitadhi, J. Sim, D. Sheffield, A. Mishra, S. Krishnan, and D. Marr, "Accelerating recurrent neural networks in analytics servers: Comparison of fpga, cpu, gpu, and asic," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1–4.
- [28] F. Vahid, S. Narayan, and D. D. Gajski, "Speccharts: A vhdl front-end for embedded systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 6, pp. 694–706, 1995.
- [29] W. E. McUmber and B. H. Cheng, "Uml-based analysis of embedded systems using a mapping to vhdl," in *Proceedings 4th IEEE International Symposium on High-Assurance Systems Engineering*. IEEE, 1999, pp. 56–63.
- [30] T. G. Moreira, M. A. Wehrmeister, C. E. Pereira, J.-F. Petin, and E. Levrat, "Automatic code generation for embedded systems: From uml specifications to vhdl code," in *2010 8th IEEE International Conference on Industrial Informatics*. IEEE, 2010, pp. 1085–1090.
- [31] A. Fin, F. Fummi, M. Martignano, and M. Signoretto, "Systemc: a homogenous environment to test embedded systems," in *Proceedings of the ninth international symposium on Hardware/software codesign*. ACM, 2001, pp. 17–22.
- [32] F. Herrera, H. Posadas, P. Sánchez, and E. Villar, "Systematic embedded software generation from systemc," in *Embedded software for SoC*. Springer, 2003, pp. 83–93.
- [33] S. Rigo, G. Araujo, M. Bartholomeu, and R. Azevedo, "Archc: A systemc-based architecture description language," in *16th Symposium on Computer Architecture and High Performance Computing*. IEEE, 2004, pp. 66–73.
- [34] H. Posadas, J. Adamez, E. Villar, F. Blasco, and F. Escuder, "Rtos modeling in systemc for real-time embedded sw simulation: A posix model," *Design Automation for Embedded Systems*, vol. 10, no. 4, pp. 209–227, 2005.
- [35] F. Herrera and E. Villar, "A framework for embedded system specification under different models of computation in systemc," in *2006 43rd ACM/IEEE Design Automation Conference*. IEEE, 2006, pp. 911–914.
- [36] F. Fummi, D. Quaglia, and F. Stefanni, "A systemc-based framework for modeling and simulation of networked embedded systems," in *2008 Forum on Specification, Verification and Design Languages*. IEEE, 2008, pp. 49–54.
- [37] M. Bramberger, J. Brunner, B. Rinner, and H. Schwabach, "Real-time video analysis on an embedded smart camera for traffic surveillance," in *Proceedings. RTAS 2004. 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004*. IEEE, 2004, pp. 174–181.
- [38] M. Litzberger, C. Posch, D. Bauer, A. N. Belbachir, P. Schon, B. Kohn, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*. IEEE, 2006, pp. 173–178.
- [39] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed embedded smart cameras for surveillance applications," *Computer*, vol. 39, no. 2, pp. 68–75, 2006.
- [40] A. Sudarsanam, *Code optimization libraries for retargetable compilation for embedded digital signal processors*. Princeton University, 1998.
- [41] J. Chase, B. Nelson, J. Bodily, Z. Wei, and D.-J. Lee, "Real-time optical flow calculations on fpga and gpu architectures: a comparison study," in *2008 16th International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2008, pp. 173–182.
- [42] A. P. D. Binotto, D. Doering, T. Stetzelberger, P. McVittie, S. Zimmermann, and C. E. Pereira, "A cpu, gpu, fpga system for x-ray image processing using high-speed scientific cameras," in *2013 25th International Symposium on Computer Architecture and High Performance Computing*. IEEE, 2013, pp. 113–119.
- [43] D. Hernandez-Juarez, A. Chacón, A. Espinosa, D. Vázquez, J. C. Moure, and A. M. López, "Embedded real-time stereo estimation via semi-global matching on the gpu," *Procedia Computer Science*, vol. 80, pp. 143–153, 2016.
- [44] K. Smiri, S. Bekri, and H. Smei, "Fault-tolerant in embedded systems (mpsoc): Performance estimation and dynamic migration tasks," in *2016 11th International Design & Test Symposium (IDT)*. IEEE, 2016, pp. 1–6.
- [45] O. Terzo, K. Djemame, A. Scionti, and C. Pezuela, *Heterogeneous Computing Architectures: Challenges and Vision*. CRC Press, 2019.
- [46] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown *et al.*, "Tile64-processor: A 64-core soc with mesh interconnect," in *2008 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*. IEEE, 2008, pp. 88–598.
- [47] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A network on chip architecture and design methodology," in *Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002*. IEEE, 2002, pp. 117–124.
- [48] N. Venkataraman and R. Kumar, "Design and analysis of application specific network on chip for reliable custom topology," *Computer Networks*, vol. 158, pp. 69–76, 2019.
- [49] P. G. Raponi, E. Norige, and S. Kumar, "Generation of network-on-chip layout based on user specified topological constraints," Feb. 26 2019, uS Patent App. 10/218,581.
- [50] D. Zydek, N. Shlayan, E. Regentova, and H. Selvaraj, "Review of packet switching technologies for future noc," in *2008 19th International Conference on Systems Engineering*. IEEE, 2008, pp. 306–311.

- [51] J. Bashir, E. Peter, and S. R. Sarangi, "A survey of on-chip optical interconnects," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, p. 115, 2019.
- [52] T. Alexoudi, N. Terzenidis, S. Pitris, M. Moralis-Pegios, P. Maniotis, C. Vagionas, C. Mitsolidou, G. Mourgias-Alexandris, G. T. Kanellos, A. Miliou *et al.*, "Optics in computing: from photonic network-on-chip to chip-to-chip interconnects and disintegrated architectures," *Journal of Lightwave Technology*, vol. 37, no. 2, pp. 363–379, 2019.
- [53] Y. Shin, K. Choi, T. Sakurai, T. Sakurai, and T. Sakurai, "Power optimization of real-time embedded systems on variable speed processors," in *Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design*. IEEE Press, 2000, pp. 365–368.
- [54] S. Segars, "Low power design techniques for microprocessors," in *International Solid-State Circuits Conference Tutorial*, 2001, pp. 1–67.
- [55] C. Zhang, F. Vahid, and W. Najjar, "A highly configurable cache architecture for embedded systems," in *30th Annual International Symposium on Computer Architecture, 2003. Proceedings*. IEEE, 2003, pp. 136–146.
- [56] S. Chalasani and J. M. Conrad, "A survey of energy harvesting sources for embedded systems," in *IEEE SoutheastCon 2008*. IEEE, 2008, pp. 442–447.
- [57] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 2005, p. 64.
- [58] V. Raghunathan and P. H. Chou, "Design and power management of energy harvesting embedded systems," in *Proceedings of the 2006 international symposium on Low power electronics and design*. ACM, 2006, pp. 369–374.
- [59] T. Soyata, L. Copeland, and W. Heinzelman, "Rf energy harvesting for embedded systems: A survey of tradeoffs and methodology," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 22–57, 2016.
- [60] B. Spencer Jr, M. E. Ruiz-Sandoval, and N. Kurata, "Smart sensing technology: opportunities and challenges," *Structural Control and Health Monitoring*, vol. 11, no. 4, pp. 349–368, 2004.
- [61] J. Marulanda, P. Thomson, J. Marulanda *et al.*, "Monitoreo de salud estructural," *Ingeniería y Competitividad*, vol. 2, no. 2, pp. 40–46, 2011.
- [62] Y. Fu, K. Mechitov, T. Hoang, J. R. Kim, D. H. Lee, and B. F. Spencer Jr, "Development and full-scale validation of high-fidelity data acquisition on a next-generation wireless smart sensor platform," *Advances in Structural Engineering*, p. 1369433219866093, 2019.
- [63] M. Ortmanns and F. Gerfers, *Continuous-time sigma-delta A/D conversion: fundamentals, performance limits and robust implementations*. Springer, 2006.
- [64] M. S. Njinowa, H. T. Bui, and F.-R. Boyer, "Design of low power 4-bit flash adc based on standard cells," in *2013 IEEE 11th International New Circuits and Systems Conference (NEWCAS)*. IEEE, 2013, pp. 1–4.
- [65] F. Begum, S. Mishra, and A. Dandapat, "Low power 10-bit flash adc with divide and collate subranging conversion scheme," *Scientia Iranica*, 2019.
- [66] H. Wong and H. Iwai, "On the scaling issues and high- κ replacement of ultrathin gate dielectrics for nanoscale mos transistors," *Microelectronic Engineering*, vol. 83, no. 10, pp. 1867–1904, 2006.
- [67] H. Iwai, "Roadmap for 22 nm and beyond," *Microelectronic Engineering*, vol. 86, no. 7–9, pp. 1520–1528, 2009.
- [68] P. Wang, "Device characteristics of short-channel and narrow-width mosfet's," *IEEE Transactions on Electron Devices*, vol. 25, no. 7, pp. 779–786, 1978.
- [69] K. Keutzer, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: orthogonalization of concerns and platform-based design," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 19, no. 12, pp. 1523–1543, 2000.
- [70] M. Lora, S. Vinco, and F. Fummi, "Translation, abstraction and integration for effective smart system design," *IEEE Transactions on Computers*, 2019.
- [71] K. Asanović and D. A. Patterson, "Instruction sets should be free: The case for risc-v," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146*, 2014.
- [72] Y. Lee, A. Waterman, R. Avizienis, H. Cook, C. Sun, V. Stojanović, and K. Asanović, "A 45nm 1.3 ghz 16.7 double-precision gflops/w risc-v processor with vector accelerators," in *ESSCIRC 2014-40th European Solid State Circuits Conference (ESSCIRC)*. IEEE, 2014, pp. 199–202.
- [73] C. Ebert and C. Jones, "Embedded software: Facts, figures, and future," *Computer*, vol. 42, no. 4, pp. 42–52, 2009.
- [74] J. Sztipanovits and G. Karsai, "Model-integrated computing," *Computer*, vol. 30, no. 4, pp. 110–111, 1997.
- [75] G. Karsai, J. Sztipanovits, A. Ledeczki, and T. Bapty, "Model-integrated development of embedded software," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 145–164, 2003.
- [76] H. Shokry and M. Hinchey, "Model-based verification of embedded software," *Computer*, 2009.
- [77] P. Liggesmeyer and M. Trapp, "Trends in embedded software engineering," *IEEE software*, vol. 26, no. 3, pp. 19–25, 2009.
- [78] B. P. Douglass, "Uml statecharts," *Embedded systems programming*, vol. 12, no. 1, pp. 22–42, 1999.
- [79] Matlab, "Stateflow," [urlhttps://es.mathworks.com/products/stateflow.html](https://es.mathworks.com/products/stateflow.html).
- [80] P. Johnston. (2018) C++ embedded frameworks. [Online]. Available: <https://embeddedartistry.com/newsletter-archive/2018/5/7/may-2018-c-embedded-frameworks>
- [81] Y. Ait-Ameur, G. Bel, F. Boniol, S. Pairault, and V. Wiels, "Robustness analysis of avionics embedded systems," *ACM SIGPLAN Notices*, vol. 38, no. 7, pp. 123–132, 2003.
- [82] Y. Moy, E. Ledinot, H. Delseny, V. Wiels, and B. Monate, "Testing or formal verification: Do-178c alternatives and industrial experience," *IEEE software*, vol. 30, no. 3, pp. 50–57, 2013.
- [83] F. Balarin, Y. Watanabe, H. Hsieh, L. Lavagno, C. Passerone, and A. Sangiovanni-Vincentelli, "Metropolis: An integrated electronic system design environment," *Computer*, vol. 36, no. 4, pp. 45–52, 2003.
- [84] C. L. Heitmeyer, M. Archer, E. I. Leonard, and J. McLean, "Formal specification and verification of data separation in a separation kernel for an embedded system," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 346–355.
- [85] D. Große, U. Kühne, and R. Drechsler, "Hw/sw co-verification of embedded systems using bounded model checking," in *Proceedings of the 16th ACM Great Lakes symposium on VLSI*. ACM, 2006, pp. 43–48.
- [86] A. Fehnker, R. Huuck, F. Rauch, and S. Seefried, "Some assembly required-program analysis of embedded system code," in *2008 Eighth IEEE International Working Conference on Source Code Analysis and Manipulation*. IEEE, 2008, pp. 15–24.
- [87] G. Klein, "Operating system verification—an overview," *Sadhana*, vol. 34, no. 1, pp. 27–69, 2009.
- [88] Y. M. Perez, H. A. P. Marin, and A. E. Bedoya, "A review on verification and validation for embedded software," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2339–2347, May 2016.
- [89] Q. Li and C. Yao, *Real-time concepts for embedded systems*. CRC Press, 2003.
- [90] R. Oshana and M. Kraeling, *Software engineering for embedded systems: Methods, practical techniques, and applications*. Newnes, 2019.
- [91] Q. Jabeen, F. Khan, M. N. Hayat, H. Khan, S. R. Jan, and F. Ullah, "A survey: Embedded systems supporting by different operating systems," *arXiv preprint arXiv:1610.07899*, 2016.
- [92] A. S. Berger, *Embedded systems design: an introduction to processes, tools, and techniques*. CRC Press, 2001.
- [93] D. Hildebrand, "An architectural overview of qnx," in *USENIX Workshop on Microkernels and Other Kernel Architectures*, 1992, pp. 113–126.
- [94] Microsoft, "Windows ce," <http://www.microsoft.com/windowsce/embedded/>, 2008.
- [95] A. Gerstlauer, H. Yu, and D. D. Gajski, "Rtos modeling for system level design," in *Embedded Software for SoC*. Springer, 2003, pp. 55–68.
- [96] F. org. (2018) Quality rtos & embedded software. [Online]. Available: <https://www.freertos.org/a00114.html>
- [97] F. Dabek, N. Zeldovich, F. Kaashoek, D. Mazières, and R. Morris, "Event-driven programming for robust software," in *Proceedings of the 10th workshop on ACM SIGOPS European workshop*. ACM, 2002, pp. 186–189.
- [98] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Prototreads: Simplifying event-driven programming of memory-constrained embedded systems," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. Acm, 2006, pp. 29–42.
- [99] K. Dewald and D. Jacoby, "Signal processing in embedded systems," *IEEE Latin America Transactions*, vol. 11, no. 1, pp. 664–667, Feb 2013.
- [100] L. P. Kaelbling, *Learning in embedded systems*. MIT press, 1993.
- [101] B. Taylor, V. S. Marco, W. Wolff, Y. Elkhatib, and Z. Wang, "Adaptive deep learning model selection on embedded systems," in *ACM SIGPLAN Notices*, vol. 53, no. 6. ACM, 2018, pp. 31–43.

- [102] R. Ernst, "Codesign of embedded systems: Status and trends," *IEEE Design & Test of Computers*, vol. 15, no. 2, pp. 45–54, 1998.
- [103] W. Wolf, "A decade of hardware/software codesign," *Computer*, no. 4, pp. 38–43, 2003.
- [104] J. Vidal, F. De Lamotte, G. Gogniat, P. Soulard, and J.-P. Digué, "A co-design approach for embedded system modeling and code generation with uml and marte," in *2009 Design, Automation & Test in Europe Conference & Exhibition*. IEEE, 2009, pp. 226–231.
- [105] K. Hines and G. Borriello, "Dynamic communication models in embedded system co-simulation," in *Proceedings of the 34th annual Design Automation Conference*. ACM, 1997, pp. 395–400.
- [106] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, p. 49, 2018.
- [107] S. Sun, M. Kadoch, L. Gong, and B. Rong, "Integrating network function virtualization with sdr and sdn for 4g/5g networks," *IEEE Network*, vol. 29, no. 3, pp. 54–59, 2015.
- [108] M. A. M. Vieira, C. N. Coelho, D. Da Silva, and J. M. da Mata, "Survey on wireless sensor network devices," in *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No. 03TH8696)*, vol. 1. IEEE, 2003, pp. 537–544.
- [109] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [110] D. E. Robling Denning, *Cryptography and data security*. Addison-Wesley Longman Publishing Co., Inc., 1982.
- [111] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, "Security for industrial communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1152–1177, 2005.
- [112] W. Stallings, *Cryptography and Network Security, 4/E*. Pearson Education India, 2006.
- [113] T. Wollinger, J. Guajardo, and C. Paar, "Cryptography in embedded systems: An overview," *Proc. Embedded World*, pp. 735–744, 2003.
- [114] O. Hyncica, P. Kucera, P. Honzik, and P. Fiedler, "Performance evaluation of symmetric cryptography in embedded systems," in *Proceedings of the 6th IEEE international conference on intelligent data acquisition and advanced computing systems*, vol. 1. IEEE, 2011, pp. 277–282.
- [115] S. Marzougui and J. Krämer, "Post-quantum cryptography in embedded systems," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*. ACM, 2019, p. 48.
- [116] A. Ahmad, M. Anisetti, E. Damiani, and G. Jeon, "Special issue on real-time image and video processing in mobile embedded systems," *Journal of Real-Time Image Processing*, vol. 16, no. 1, pp. 1–4, 2019.
- [117] A. Sangiovanni-Vincentelli and M. Di Natale, "Embedded system design for automotive applications," *Computer*, vol. 40, no. 10, pp. 42–51, 2007.
- [118] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, 2005.
- [119] E. Irrazábal, I. Sambrana, and C. Pinto Luft, "Proceso para el diseño de la arquitectura software en un sistema crítico ferroviario según la norma en-50128," in *XIX Simposio Argentino de Ingeniería de Software (ASSE)-JAIIO 47 (CABA, 2018)*, 2018.
- [120] L. L. Bello, S. Mubeen, S. Saponara, R. Mariani, and U. D. Bordoloi, "Guest editorial embedded and networked systems for intelligent vehicles and robots," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1035–1037, 2019.
- [121] J. A. Monroy, E. Campos, and J. A. Torres, "Attitude control of a micro auv through an embedded system," *IEEE Latin America Transactions*, vol. 15, no. 4, pp. 603–612, April 2017.
- [122] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE transactions on robotics and automation*, vol. 19, no. 3, pp. 371–380, 2003.
- [123] A. Malinowski and H. Yu, "Comparison of embedded system design for industrial applications," *IEEE transactions on industrial informatics*, vol. 7, no. 2, pp. 244–254, 2011.
- [124] E. Cabal-Yepez, A. G. Garcia-Ramirez, R. J. Romero-Troncoso, A. Garcia-Perez, and R. A. Osornio-Rios, "Reconfigurable monitoring system for time-frequency analysis on industrial equipment through stft and dwt," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 760–771, 2012.
- [125] J. Kern, C. Urrea, R. Méndez, and G. González, "Development of an embedded control system by means of dspic applied in a 4 dof robot," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2099–2106, May 2016.
- [126] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch, "Industry 4.0: The future of productivity and growth in manufacturing industries," *Boston Consulting Group*, vol. 9, no. 1, pp. 54–89, 2015.
- [127] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 363–369.
- [128] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, A. Roederer et al., "Challenges and research directions in medical cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 75–90, 2011.
- [129] D. Serpanos, "The cyber-physical systems revolution," *Computer*, vol. 51, no. 3, pp. 70–73, 2018.
- [130] E. R. Griffor, C. Greer, D. A. Wollman, and M. J. Burns, "Framework for cyber-physical systems: Volume 1, overview," NIST, Tech. Rep., 2017.
- [131] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," *International Journal of Communication Systems*, vol. 25, no. 9, p. 1101, 2012.
- [132] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [133] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta, "Review of iot applications in agro-industrial and environmental fields," *Computers and Electronics in Agriculture*, vol. 142, pp. 283–297, 2017.
- [134] I. A. Lakhari, G. Jianmin, T. N. Syed, F. A. Chandio, N. A. Buttar, and W. A. Qureshi, "Monitoring and control systems in agriculture using intelligent sensor techniques: A review of the aeroponic system," *Journal of Sensors*, vol. 2018, 2018.
- [135] J. J. Segura-Juárez, D. Cuesta-Frau, L. Samblas-Pena, and M. Aboy, "A microcontroller-based portable electrocardiograph recorder," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 9, pp. 1686–1690, 2004.
- [136] D. Lucani, G. Cataldo, J. Cruz, G. Villegas, and S. Wong, "A portable eeg monitoring device with bluetooth and holter capabilities for telemedicine applications," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2006, pp. 5244–5247.
- [137] S. Borromeo, C. Rodriguez-Sanchez, F. Machado, J. A. Hernandez-Tamames, and R. de la Prieta, "A reconfigurable, wearable, wireless eeg system," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 1659–1662.
- [138] R. Gupta, J. Bera, and M. Mitra, "Development of an embedded system and matlab-based gui for online acquisition and analysis of eeg signal," *Measurement*, vol. 43, no. 9, pp. 1119–1126, 2010.
- [139] W.-C. Huang, S.-H. Hung, J.-F. Chung, M.-H. Chang, L.-D. Van, and C.-T. Lin, "Fpga implementation of 4-channel ica for on-line eeg signal separation," in *2008 IEEE Biomedical Circuits and Systems Conference*. IEEE, 2008, pp. 65–68.
- [140] C.-T. Lin, Y.-C. Chen, T.-Y. Huang, T.-T. Chiu, L.-W. Ko, S.-F. Liang, H.-Y. Hsieh, S.-H. Hsu, and J.-R. Duann, "Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver's drowsiness detection and warning," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 5, pp. 1582–1591, 2008.
- [141] C.-T. Lin, L.-W. Ko, J.-C. Chiou, J.-R. Duann, R.-S. Huang, S.-F. Liang, T.-W. Chiu, and T.-P. Jung, "Noninvasive neural prostheses using mobile and wireless eeg," *Proceedings of the IEEE*, vol. 96, no. 7, pp. 1167–1183, 2008.
- [142] C.-T. Lin, C.-J. Chang, B.-S. Lin, S.-H. Hung, C.-F. Chao, and I.-J. Wang, "A real-time wireless brain-computer interface system for drowsiness detection," *IEEE transactions on biomedical circuits and systems*, vol. 4, no. 4, pp. 214–222, 2010.
- [143] L. Brown, J. van de Molengraft, R. F. Yazicioglu, T. Torfs, J. Penders, and C. Van Hoof, "A low-power, wireless, 8-channel eeg monitoring headset," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 2010, pp. 4197–4200.
- [144] A. Page, C. Sagedy, E. Smith, N. Attaran, T. Oates, and T. Mohsenin, "A flexible multichannel eeg feature extractor and classifier for seizure detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 2, pp. 109–113, 2014.
- [145] B.-G. Lee, B.-L. Lee, and W.-Y. Chung, "Mobile healthcare for automatic driving sleep-onset detection using wavelet-based eeg and respiration signals," *Sensors*, vol. 14, no. 10, pp. 17915–17936, 2014.
- [146] J. A. Mercado, J. Herrera, A. d. J. Pansza, and J. Gutierrez, "Embedded eeg recording module with active electrodes for motor imagery brain-

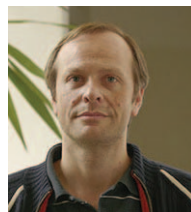
- computer interface," *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 503–510, Feb 2016.
- [147] T. Gao and D. White, "A next generation electronic triage to aid mass casualty emergency medical response," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2006, pp. 6501–6504.
- [148] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 21, 2012.
- [149] S. Majumder, T. Mondal, and M. Deen, "Wearable sensors for remote health monitoring," *Sensors*, vol. 17, no. 1, p. 130, 2017.
- [150] A. H. Kronbauer, H. C. da Luz, and J. Campos, "Mobile security monitor: A wearable computing platform to detect and notify falls," *IEEE Latin America Transactions*, vol. 16, no. 3, pp. 957–965, March 2018.
- [151] A. Darwish and A. E. Hassanien, "Wearable and implantable wireless sensor network solutions for healthcare monitoring," *Sensors*, vol. 11, no. 6, pp. 5561–5595, 2011.
- [152] L. S. Wong, S. Hossain, A. Ta, J. Edvinsson, D. H. Rivas, and H. Naas, "A very low-power cmos mixed-signal ic for implantable pacemaker applications," *IEEE Journal of solid-state circuits*, vol. 39, no. 12, pp. 2446–2456, 2004.
- [153] S. D. Chede and K. D. Kulat, "Design overview of processor based implantable pacemaker," *Journal of Computers*, vol. 3, no. 8, pp. 49–57, 2008.
- [154] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (suc 2008)*. IEEE, 2008, pp. 1–9.
- [155] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 2008, pp. 129–142.
- [156] L. Kristen. (2019) Delivering next-generation ai experiences for the 5g world. [Online]. Available: <https://www.arm.com/company/news/2019/05/delivering-next-generation-ai-experiences-for-the-5g-world>
- [157] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, S. Moderator-Ravi, and S. Moderator-Ravi, "Security as a new dimension in embedded system design," in *Proceedings of the 41st annual Design Automation Conference*. ACM, 2004, pp. 753–760.
- [158] S. Ravi, A. Raghunathan, and S. Chakradhar, "Tamper resistance mechanisms for secure embedded systems," in *17th International Conference on VLSI Design. Proceedings*. IEEE, 2004, pp. 605–611.
- [159] S. Ravi, S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 3, pp. 461–491, 2004.
- [160] D. D. Hwang, P. Schaumont, K. Tiri, and I. Verbauwhede, "Securing embedded systems," *IEEE Security & Privacy*, no. 2, pp. 40–49, 2006.
- [161] T. Huffmire, B. Brotherton, T. Sherwood, R. Kastner, T. Levin, T. D. Nguyen, and C. Irvine, "Managing security in fpga-based embedded systems," *IEEE Design & Test of Computers*, vol. 25, no. 6, pp. 590–598, 2008.
- [162] A. Ukil, J. Sen, and S. Koilakonda, "Embedded security for internet of things," in *2011 2nd National Conference on Emerging Trends and Applications in Computer Science*. IEEE, 2011, pp. 1–6.
- [163] D. Papp, Z. Ma, and L. Buttyan, "Embedded systems security: Threats, vulnerabilities, and attack taxonomy," in *2015 13th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2015, pp. 145–152.
- [164] M.-K. Yoon, S. Mohan, J. Choi, M. Christodorescu, and L. Sha, "Learning execution contexts from system call distribution for anomaly detection in smart embedded system," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM, 2017, pp. 191–196.
- [165] A. Damien, M. Fumey, E. Alata, M. Kaâniche, and V. Nicomette, "Anomaly based intrusion detection for an avionic embedded system," SAE Technical Paper, Tech. Rep., 2018.
- [166] A. Flores-Vergara, E. Garcia-Guerrero, E. Inzunza-González, O. López-Bonilla, E. Rodríguez-Orozco, J. Cardenas-Valdez, and E. Tlelo-Cuautle, "Implementing a chaotic cryptosystem in a 64-bit embedded system by using multiple-precision arithmetic," *Nonlinear Dynamics*, vol. 96, no. 1, pp. 497–516, 2019.



Luciana De Micco Electronics Engineer (2004) and Engineering Doctor with Mention in Electronics (2009), Facultad de Ingeniería, Universidad Nacional de Mar Del Plata - UNMDP, Argentina. De Micco is director of the Laboratory of Chaotic Systems at the Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE). She is Adjunct Researcher at the National Scientific and Technical Research Council (CONICET), Argentina. She is Category 3 Research Professor, according to the Comisión Nacional de Categorización, Ministry of Education, Argentina. Luciana is also a Junior Associate Researcher at the Abdus Salam International Centre for Theoretical Physics (ICTP). She is Adjunct Professor at the Universidad Nacional de Mar del Plata. De Micco is interested in chaotic systems, its digitalization and implementation in digital devices.



Fabian Luis Vargas Fabian Vargas obtained his Ph.D. Degree in Microelectronics from the Institut National Polytechnique de Grenoble (INPG), France, in 1995. At present, he is Full Professor at the Catholic University (PUCRS) in Porto Alegre, Brazil. He works in the area of computer systems architecture focusing on robustness and security for critical applications. Prof. Vargas is "Level-1" researcher of the Brazilian National Science Foundation (CNPq) since 1996. He co-founded the IEEE-Computer Society Latin American Test Technology Technical Council (LA-TTTC) in 1997 and the IEEE Latin American Test Symposium - LATS (former Latin American Test Workshop - LATW) in 2000. Prof. Vargas has acted as Guest Editor of several journal, such as the Journal of Electronic Testing: Theory and Applications - JETTA (Springer), IEEE Design and Test of Computers Magazine and the Journal of Low-Power Electronics - JOLPE (American Scientific Publishers). He is currently member of the Editorial Board of JETTA. Prof. Vargas received for several times the Meritorious Service Award of the IEEE Computer Society for providing significant services as chair of the IEEE Latin American Regional TTTC Group and the LATS. Prof. Vargas is Senior Member of IEEE and Golden Core Member of the IEEE Computer Society.



Pablo I. Fierens Electronics Engineer (1997), Instituto Tecnológico de Buenos Aires - ITBA, Argentina. MSc (2000) and PhD (2003), Cornell University, USA Fierens is Full Professor at ITBA. He is director of the Optoelectronics Lab at ITBA. Independent Researcher at CONICET. Associate Editor of IEEE Latin America Transactions. IEEE Senior Member. Pablo is interested in communications, nonlinear optics, and noise in nonlinear systems.