

***eUCCvm*: Una Herramienta Educativa Integral para la Programación de Sistemas Embebidos**

Cristian F. Perez-Monte¹, Gustavo Mercado¹, Carlos Taffernaberry¹, Ana Laura Diedrichs¹, Fabiana Piccoli^{2,3}, Mario Sebastian Tobar¹, Marcelo Ledda¹, Raúl Moralejo¹, and Rodrigo Gonzalez¹

¹ GridTICs, Universidad Tecnológica Nacional, Facultad Regional Mendoza, Argentina

² Universidad Nacional de San Luis, Ejército de los Andes 950, San Luis Argentina

³ Universidad Autónoma de Entre Ríos, 25 de Mayo 385, Concepción del Uruguay, Entre Ríos, Argentina

{cristian.perez.gustavo.mercado,carlos.taffernaberry,sebastian.tobar,rodrigo.gonzalez}@gridtics.frm.utn.edu.ar mpiccoli@unsl.edu.ar

{ana.diedrichs,marcelo.ledda,raul.moralejo}@docentes.frm.utn.edu.ar
<http://www.gridtics.frm.utn.edu.ar>

Abstract. Replantar nuestras prácticas y formas de enseñanza, ya sea por acontecimientos mundiales como fue la pandemia o por particularidades de nuestra realidad: condiciones sociales, económicas, etc. es una necesidad. En este trabajo presentamos una herramienta de código abierto y compilación cruzada universal para la programación de Sistemas Embebidos: *eUCCvm*. Mediante su uso es posible desarrollar actividades de programación de aplicaciones para Sistemas Embebidos sobre múltiples arquitecturas. Puede ser utilizada en el ámbito académico como laboral, y en cualquier modalidad: virtual o presencial. Finalmente mostramos una breve encuesta sobre su inclusión en algunas cátedras de la Universidad Tecnológica Nacional, Regional Mendoza.

Keywords: *Sistemas Embebidos, Programación de Arquitecturas, Software Educativo, Código Abierto*

1. Introducción

eUCCvm (*Embedded use Universal Cross Compiler Virtual Machine*) es un sistema basado en una máquina virtual para desarrollo, compilación y programación de sistemas embebidos de uso gratuito, usada como herramienta para el desarrollo de las clases prácticas en diferentes cátedras de la Universidad Tecnológica Nacional, Facultad Regional Mendoza. Inicialmente fue desarrollada en el año 2017 para la programación específica de EDU-CIAA [4,8]. En 2018 y 2019 se extendió su uso a otras arquitecturas de microcontroladores y microprocesadores [26], aquellas utilizadas principalmente en la cátedra de Técnicas Digitales II [24] y III [25]. A partir del año 2020, con el inicio de la pandemia, se inició un proceso para la adaptación de todo el sistema educativo y la currícula por la virtualidad. De igual forma, en este mismo sentido se hicieron una serie de mejoras sobre *eUCCvm* a fin de adaptarlo, teniendo en cuenta las nuevas necesidades de alumnos y docentes para la realización de actividades de la cátedra derivadas del ASPO (Aislamiento Social Preventivo y Obligatorio dispuesto por Decreto 297/2020). En los años siguientes continuamos agregando arquitecturas de microcontroladores y microprocesadores, incluidas las usadas en computadoras hogareñas y consolas de

video juegos. Finalmente a partir de este año se agregaron los microcontroladores con arquitectura de hardware abierto RISCv [21], los cuales son centrales para el futuro del dictado de las cátedras de técnicas digitales. Además de lo expuesto, como *eUCCvm* es una herramienta pública y de uso educativo, es posible descargarla desde [23].

Este trabajo está organizado de la siguiente manera: en la sección 2 describimos su impacto en el ámbito académico. Las secciones 3 y 4 detallan las características generales y específicas respectivamente de *eUCCvm*. La sección 5 muestra resultados en relación al impacto de su aplicación. Finalmente, en la sección 6 presentamos las conclusiones y trabajos futuros.

2. *eUCCvm* y su Aplicación en el Ámbito Académico

eUCCvm está desarrollado para permitir aprender a programar sistemas embebidos en un entorno seguro y controlado, sin tener que preocuparse por dañar su propio sistema operativo.

Para el diseño y desarrollo incorporamos la perspectiva CTS (Ciencia, Tecnología y Sociedad) [20,6,19], la misma consiste en tener en cuenta los aspectos ideológicos, culturales y la experiencia personal (valores, sentimientos, motivaciones, entre otras), además de los aspectos técnicos y operativos.

eUCCvm permite utilizar computadoras no necesariamente con muchos recursos ya que demanda pocos requerimientos para su utilización por los alumnos, e ideal para en países subdesarrollados o en vías de desarrollo. Además, permite resolver problemas como los impuestos durante la virtualidad: el acceso intermitente, deficiente o nulo a Internet, constituyendo una importante barrera para la realización de actividades. En este contexto, *eUCCvm* brinda una herramienta integral donde tiene todo incluido y está lista para ser usada sin la necesidad de acceso permanente a Internet en la realización de actividades virtuales de los alumnos. Así es posible realizar actividades indistintamente en la virtualidad desde sus casas sin necesidad de recurrir a recursos computacionales avanzados o acceso a Internet, como también desde el ambiente académico, maximizando en este caso el tiempo de clase dedicado a la programación y desarrollo de proyectos con microcontroladores. En virtud de ello, para adaptarlo a las condiciones impuestas por la virtualidad, *eUCCvm* cuenta no solo con la capacidad de programar, compilar y grabar sobre el hardware de muchas arquitecturas sino que dispone de documentación de soporte y actividades desafíos para la competencia individual o entre grupos.

Otra ventaja que ofrece es su naturaleza de código abierto, la utilización de una herramienta con este tipo de código fuente abre la posibilidad a los estudiantes de modificar la herramienta para potenciarla para sus desarrollos propios, y a los docentes de adaptar sus prácticas. Asimismo se le suma la posibilidad de utilizar hardware abierto, como la arquitectura RISCv y los beneficios de su uso [12].

eUCCvm brinda un amplio abanico de plataformas de hardware de microcontroladores, no solo para fortalecer las habilidades de programación en cada uno de ellas sino también para fortalecer las capacidades de aprendizaje de nuevas arquitecturas que puedan surgir en el futuro. En otras palabras se refuerzan las capacidades de "aprender a aprender" nuevas arquitecturas y/o lenguajes de

programación para los mismos. Además, al contar con arquitecturas de diferentes niveles de complejidad, le posibilita a los estudiantes aprender a su ritmo y aplicar sus conocimientos a medida que avanza en el curso, permitiendo desarrollar proyectos completos sobre hardware real ya sea en forma individual o grupalmente.

Otra de sus características y que consideramos un incentivo adicional, es la inclusión de arquitecturas utilizadas en la industria de los videojuegos, a fin de conocerlas y poder programarlas, no solo para ampliar el conocimiento respecto a nuevas arquitecturas, sino también de focalizarse en aquellas que los estudiantes están familiarizados por su utilización previa. Ya es ampliamente conocido y debatido que la integración de los videojuegos en el ambiente educativo es muy beneficioso [7] [11].

Las potenciales habilidades y beneficios a lograr por la utilización de *eUCCvm* en los estudiantes que realizan materias relacionadas a la programación de Sistemas Embebidos son:

- Potenciar la programación en los lenguajes ensamblador y C para diferentes microcontroladores.
- Al permitir no solo la programación, sino también la compilación y grabación del programa sobre el hardware real, se permiten desarrollar prácticos completos, con lo cual los estudiantes pueden ver cómo se aplican los conceptos teóricos en situaciones del mundo real.
- Al ofrecer la posibilidad de trabajar sobre una variedad de arquitecturas de hardware de microcontroladores, no solo fortalece las habilidades de programación en cada uno de ellas, sino también potencia las destrezas en el aprendizaje de nuevas arquitecturas que puedan surgir en el futuro.
- Impulsar habilidades en la programación de microcontroladores para el manejo de sistemas de almacenamiento, visualización, sensado y actuadores.
- Potenciar habilidades en el diseño y programación de interfases de comunicaciones para la interconexión entre microcontroladores o entre microcontroladores y otros dispositivos como sensores, actuadores o de entrada salida.
- Favorecer el trabajo colaborativo mediante la realización de actividades desafío para potenciar el desarrollo de proyectos grupales o individuales.
- Reforzar habilidades en la utilización de lenguajes de descripción de hardware.
- Discernir sobre la arquitectura más adecuada para proyectos a desarrollar.
- Discernir sobre qué lenguaje de programación es más adecuado para determinadas arquitecturas y proyectos a desarrollar.
- Reforzar habilidades en el uso de línea de comandos para el manejo avanzado de un sistema operativo Linux.
- Reforzar habilidades en la programación, compilación, versionado y utilización de repositorios mediante Linux con línea de comandos.
- Fortalecer el conocimiento avanzado teórico y de programación en arquitecturas usadas para el entretenimiento en las cuales el alumno ya está previamente familiarizado por su uso en videojuegos.

Por todo lo expuesto, se puede determinar que *eUCCvm* posee muchas propiedades que lo hacen un buen candidato a la hora de considerar un software para el desarrollo

de prácticas de programación en Sistemas Embebidos cualquiera sea su modalidad y sin mayores requerimientos de recursos.

3. Características Generales de *eUCCvm*

eUCCvm está desarrollado sobre una máquina virtual [10] basada en el sistema operativo Linux Ubuntu 18.04 LTS. Puede utilizarse en distintos sistemas operativos de máquinas reales gracias a VirtualBox [17].

Para su instalación, se usó la versión de 32 bits a fin de permitir su virtualización incluso en computadoras sin características de virtualización por hardware o instrucciones AMD64/EMT64. Además se utilizó la instalación de servidor para obtener un sistema operativo liviano y sin entorno gráfico para la virtualización en máquinas con solo 256 MB de RAM tales como las netbooks disponibles del Plan Conectar Igualdad [30].

La máquina virtual posee diversos compiladores instalados desde los repositorios oficiales de Ubuntu. Además incluye otros compilados desde el código fuente oficial de sus desarrolladores y varios SDKs de desarrollo para distintas arquitecturas.

Adicionalmente se incluye la documentación básica para el soporte de desarrollo de hardware y la programación de cada una de las arquitecturas.

A continuación se especifican con mayor precisión algunas de las características de *eUCCvm*.

4. Especificaciones Detalladas

El sistema desarrollado agrega las siguientes capacidades al sistema operativo:

- Escritura de código. Permite la generación de código en diferentes lenguajes de programación, especialmente lenguaje ensamblador [13] y C [28].
- Modificación de códigos existentes. Interpretación del funcionamiento de códigos de ejemplos existentes y su modificación para adaptarlos a la solución de nuevos problemas propuestos.
- Compilación. Generación del código máquina para diferentes arquitecturas de hardware.
- Grabación de los programas compilados en dispositivos embebidos reales.
- Simulación de microcontroladores descritos en Verilog (NEORV32, arquitectura RISCv32).
- Acceso remoto seguro (ssh) para permitir la utilización de múltiples consolas y acceso a todo el sistema de archivos para su edición dentro o fuera de la máquina virtual.
- Documentación de soporte.
- Entornos de desarrollo con múltiples ejemplos didácticos.
- Actividades de desafíos prácticos para resolver en forma individual o grupal.

El sistema está limitado al uso de herramientas de entorno de línea de comandos (CLI) para permitir un entorno liviano y transportable, impidiendo la ejecución de herramientas que requieran de modo gráfico (GUI). Para la escritura y modificación de código se utilizan editores estándares disponibles para Linux.

4.1 Arquitecturas, Módulos y otras Tecnologías

En la actualidad, se cuenta con el soporte de varias arquitecturas, permitiendo una fácil actualización e inclusión de otras futuras.

Las arquitecturas de sistemas embebidos consideradas para compilación, tanto en ejecución *bare metal* como para su ejecución en GNU/Linux (para el caso de las más potentes), son:

- Microchip PIC [29].
- AVR (usados en Arduino. [9] NANO, UNO y MEGA).
- Xtensa LX106 (usados por ESP8266 [27] y ESP32).
- ARM Cortex M3 [5] [32] (utilizados en Blue pill y Arduino DUE).
- ARM Cortex M4 (utilizados en Black Pill y EDU CIAA) [32] – RISC RV32IMC (usados en módulos como ESP32-CX y ESP32-H2) [15] – ARM 11 de 32 bits [1] (utilizados en Raspberry PI 1).
- ARM Cortex A, A7 de 32 bits [2] [31] (utilizados en Raspberry PI 2) y A53 de 64 bits [3] [14] (utilizados en Raspberry PI 3).

Además de las arquitecturas más avanzadas para su ejecución en Linux, tales como:

- x86 estándar y x86 (amd64/emt64).
- MIPS.
- SPARC.
- ALPHA.
- 68k.
- POWERPC.
- RISC V RV64

Finalmente, y para el caso de las computadoras hogareñas y consolas de video juegos, las arquitecturas incluidas son:

- Ricoh 2A03 (Nintendo Entertainment System)
- 6507 (ATARI 2600)
- 68K (ATARI ST - AMIGA OCS - SEGA GENESIS)
- ARM (GameBoy Advance)
- Ricoh 5A22 (Super Nintendo)
- Zilog Z80 (Sega Master System)
- MOS 6502 (Atari400/800)
- Sharp LR35902 (GameBoy)
- CPU 32/64 MIPS6 III NEC VR 4300i + GPU RCP de 64 bits (Nintendo64)
- 68K+Z80 (Sega MegaDrive)

Además, agrega ejemplos para la utilización de módulos, que ofrecen funcionalidades a las arquitecturas antes mencionadas, tales como:

- Módulos de GPIO, sensado de variables analógicas y visualización numérica de 7 segmentos.
- Pantallas LCD y OLED de alta resolución.
- Módulos de conectividad: Ethernet, de radio GPRS, WiFi o Bluetooth.
- Módulos de CCDs.
- Almacenamiento con microSD.
- Módulos de posicionamiento global.

El sistema dispone también de ejemplos sencillos para la implementación de diferentes tecnologías electrónicas de entrada-salida y comunicaciones tales como:

- Uso de entradas salidas digitales de propósito general.
- Uso de entradas y salidas analógicas.
- Protocolo de comunicaciones SPI [16] e I2C. – Protocolos de comunicaciones paralelos.

Dentro de las herramientas integradas tiene las correspondiente a la grabación de la compilación en diversas arquitecturas para sistemas embebidos como:

- Grabación por puerto USB de AVR Arduino Uno y Mega.
- ARM Blue Pill, ARM Black Pill, Arduino Due y EDU-CIAA.
- ESP8266 y ESP32.
- Microchip PIC con PICkit 2 [18].

Todos los ejemplos están disponibles dentro del sistema, ordenados en carpetas, arquitectura y módulos usados.

5. Resultados Obtenidos

Actualmente el sistema está siendo usado en las cátedras de Técnicas Digitales III, desde el año 2017, y de Técnicas Digitales II, a partir del año 2018 hasta el presente año. En la tabla 1 se muestran los temas desarrollados utilizando *eUCCvm* en el área práctica de Técnicas Digitales II, año por año. Por un lado, mediante el uso de actividades guiadas durante el desarrollo de la cátedra, se indican el hardware (módulos, consolas o sistemas computacionales) con sus respectivas arquitecturas aplicadas a la cátedra. Por otro lado, además, mediante la aplicación de actividades especiales, por desafíos grupales o individuales, se muestran también actividades específicas de integración de tecnologías o módulos de funcionalidad a diferentes arquitecturas. Se puede observar como la cantidad de actividades integradas en la cátedra realizadas con *eUCCvm* se fue incrementando año a año, siendo mucho más creciente a partir del 2020 y de su adaptación a la virtualidad.

Table 1. Áreas temáticas aplicadas

Año	Arquitecturas	Funcionalidad	# Activ.
2018	EDU-CIAA (ARM CORTEX M4)	Ej de uso de firmware v2	1
2019	Microcontroladores PIC con pickit 2 Módulo PIC DM164140(16F18855) Arduino UNO(AVR Atmega328) Arduino MEGA(AVR Atmega2560) Arduino DUE (ARM CORTEX M3) Bluepill (ARM CORTEX M3) ESP8266 (Tensilica Xtensa L106) ESP32 (Tensilica Xtensa LX6) CIAA-Z3R0 (ARM CORTEX M0+)	Ejemplos de uso de firmware v3 Desafío de uso de puertos de comunicaciones con ARM	12
2020	Blackpill (ARM CORTEX M4) Nintendo (MOS 6502 /2A03) Super Nintendo(MOS 65C816/5A22) Sega Genesis (68K + Z80) Nintendo64(MIPS6 III NEC VR 4300i + GPU RCP de 64 bits)	-	23
2021	-	Desafío de aplicación de puerto de comunicaciones y sistemas de almacenamiento con sistemas AVR	25
2022	-	Desafío de aplicación de uso de comunicaciones y conversores analógico digital con ARM	27
2023	NEORV32 y ESP32-C3 (RISCV)	En desarrollo	30(...)

Respecto al uso académico, en la tabla 2 indicamos la cantidad de alumnos que usaron *eUCCvm* en cada año de cursado, la cantidad de descargas desde el servidor de la Facultad Regional Mendoza de UTN y la cantidad de versiones que se publicaron para la incorporación de mejoras. Como se puede observar, el número de descargas está vinculada a la cantidad de alumnos y versiones publicadas.

Table 2. Impacto académico

Año	Cantidad de alumnos	Cantidad de descargas	Cantidad de versiones de <i>eUCCvm</i>
2019	44	440	31
2020	43	400	9
2021	43	405	2
2022	33	187	2
2023	16	131	4 (...)

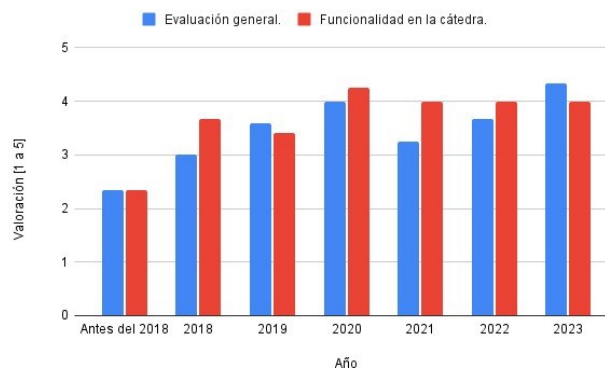
El proyecto está siendo usado, además del ambiente académico universitario, en otros ámbitos. Desde el año 2019 está compartido en forma pública por diversos medios, uno de ellos [22], independiente de las cátedras de UTN. Desde dicho sitio, cientos de descargas muestran el creciente interés, más allá del uso académico.

Para desarrollar un análisis del impacto de *eUCCvm* en los alumnos, se realizó una encuesta a alumnos que usaron la herramienta, con distintas modalidades, durante el

cursado de algunas de las materias, y a otros que no la usaron por no estar disponible en dicho momento. La muestra utilizada fue de 40 alumnos del año 2018 hasta el 2023 de una población aproximada de 220 y de otros 20 alumnos, quienes cursaron antes del 2018 y no la utilizaron, solo le describimos para la encuesta, las características generales de *eUCCvm*.

La encuesta consistía en los siguientes dos tipos de consultas:

- Consultas generales sobre las herramientas de software usadas en la cátedra.
- Consultas específicas sobre *eUCCvm*.

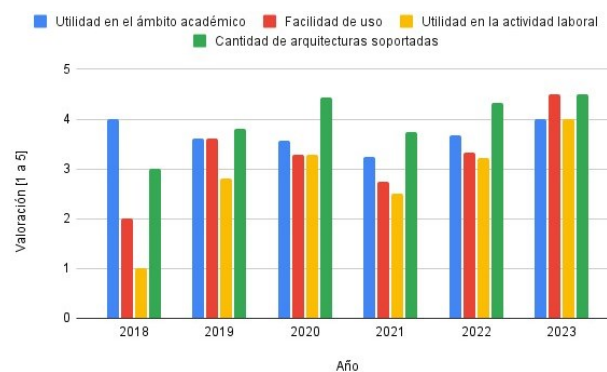


h

Fig.1. Valoraciones a las Consultas Generales de *eUCCvm*.

Respecto a las consultas generales, permitió realizar una evaluación de las herramientas por parte de los alumnos, tanto de *eUCCvm* como de otras. Esto nos permitió realizar una evaluación, incluyendo años de cursado en los cuales aún no se aplicaba *eUCCvm*.

Los resultados de las consultas generales pueden observarse en la Figura 1.



!ht

Fig.2. Consultas específicas sobre *eUCCvm*.

De las consultas específicas sobre *eUCCvm* solo pudo realizarse en forma acotada y con limitaciones a partir de alumnos del 2018 y con la mayor parte de las

funcionalidades disponibles en el año 2019. Los resultados de las consultas específicas se observan en la Figura 2.

La primera versión, 2018, sólo consideraba una placa: la edu-ciaa, a partir del 2019 se agregaron todas las detalladas en la tabla 1. En la Figura 2 se observa como los alumnos del 2018 tuvieron una valoración menor respecto a los años subsiguientes en lo que respecta a las arquitecturas. Con las constantes actualizaciones, todos los demás parámetros también crecieron.

6. Conclusiones y Trabajos futuros

Con el desarrollo e inclusión de *eUCCvm* en diferentes cátedras de carreras de la UTN regional Mendoza a partir de 2018, la herramienta ha mostrado resultados satisfactorios tanto para la modalidad presencial como en la virtualidad. Los resultados fueron más allá del ámbito académico y la resolución de prácticos, también permite su aplicación en ambientes laborales y de investigación, resaltando en todos ellos su facilidad de uso y cantidad de arquitecturas soportadas. Es bueno destacar el rol fundamental adquirido durante el año 2020, con su utilización para la modalidad virtual.

Resulta importante resaltar que con *eUCCvm*, los estudiantes cuentan con una herramienta "*lista para utilizar*", permitiendo optimizar el tiempo de clase al realizar actividades sin necesidad de descargar o instalación de programas adicionales. Además, la constante evolución de la plataforma, ya sea con el incremento año a año de la inclusión de nuevas arquitecturas y de actividades implementadas; o por su propiedad de ser software abierto y la inclusión de alguna mejora.

Como línea de trabajo futuro, existen varios aspectos a considerar, uno de ellos son llevar a cabo diferentes mejoras, entre ellas, se destaca la actualización de la máquina virtual a la última versión LTS de Ubuntu, así como la integración nativa de soporte para sistemas de 64 bits. Además, se planea incorporar la posibilidad de realizar actividades directamente en la plataforma mediante su integración web, lo que ofrecerá una experiencia más fluida y accesible para los usuarios.

Agradecimientos

Agradecemos el soporte de grupo GridTICs perteneciente a UTN - FRM y a los docentes de las cátedras Técnicas Digitales II y III por incluir *eUCCvm* o colaborar en su desarrollo. El resto del hardware necesario para el desarrollo de la plataforma ha sido obtenido por financiamiento particular de los autores del presente trabajo.

Referencias

1. BCM2835 chipset, Raspberry Pi documentation, <http://bit.ly/2woaRKW>
2. BCM2836 chipset, Raspberry Pi documentation, <http://bit.ly/2HDmLH9>
2. BCM2837 chipset, Raspberry Pi documentation, <http://bit.ly/2K76gET>

3. EDU-CIAA-NXP (Marzo 2017), <http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-ciaa:edu-ciaa-nxp>
4. Blem, E., Menon, J., Sankaralingam, K.: A detailed analysis of contemporary ARM and x86 architectures. UW-Madison Technical Report (2013)
5. Cabo, J.M., Moralejo, R.O.: Desarrollo de instrumentos de evaluación educativa hacia tecnologías específicas desde la perspectiva ciencia, tecnología y sociedad: El caso de la calidad del software. In: II Congreso de Tecnología en Educación y Educación en Tecnología (2007)
6. Coller, B.D., Scott, M.J.: Effectiveness of using a video game to teach a course in mechanical engineering. *Computers & Education* **53**(3), 900–912 (2009)
7. Dell’Oso, M., Lanzarini, L.C., Ridolfi, P.: Prototipo funcional de un sistema de detección de caídas basado en la plataforma CIAA. In: XXII Congreso Argentino de Ciencias de la Computación (CACiC 2016). (2016)
8. D’Ausilio, A.: Arduino: A low-cost multipurpose lab equipment. *Behavior research methods* **44**(2), 305–313 (2012)
9. Goldberg, R.P.: Survey of virtual machine research. *Computer* **7**(6), 34–45 (June 1974). <https://doi.org/10.1109/MC.1974.6323581>
10. Griffiths, M.D.: The educational benefits of videogames. *Education and health* **20**(3), 47–51 (2002)
11. Hannig, F., Teich, J.: Open source hardware. *Computer* **54**(10), 111–115 (2021). <https://doi.org/10.1109/MC.2021.3099046>
12. Hyde, R.: The art of assembly language. No Starch Press (2003)
13. Ivković, J., Radulović, B.: The advantages of using Raspberry Pi 3 compared to Raspberry Pi 2. SoC computers for sensor system support. In: International Conference on Applied Internet and Information Technologies. pp. 88–94 (2016)
14. Kuo, Y.M., Garcia-Herrero, F., de la Cuerda, J.A.M.: Design, implementation, and characterization of custom risc-v soft-core processors for future communication networks
15. Leens, F.: An introduction to I2C and SPI protocols. *IEEE Instrumentation & Measurement Magazine* **12**(1), 8–13 (2009)
16. Li, P.: Selecting and using virtualization solutions: our experiences with VMware and VirtualBox. *Journal of Computing Sciences in Colleges* **25**(3), 11–17 (2010)
17. Meriac, M.: Heart of darkness-exploring the uncharted backwaters of hid iclass (tm) security. In: 27th Chaos Communication Congress (2010)
18. Moralejo, R.O., Cabo, J.M.: Visión de la tecnología en estudiantes de ingeniería en sistemas de información en mendoza-argentina. In: IX Workshop de Investigadores en Ciencias de la Computación (2007)
19. Pacey, A.: La cultura de la tecnología, México, fce. Enfoques sobre la Tecnología en: <http://www.cneq.unam.mx/> [Consulta: 21 de octubre de 2014] (1990)
20. Patterson, D., Waterman, A.: The RISC-V Reader: an open architecture Atlas. Strawberry Canyon (2017)
21. Perez-Monte, C.F., Diedrichs, A.L.: eUCCvm: Plataforma educativa de programación y compilación cruzada universal de sistemas embebidos (May 2019). <https://doi.org/10.6084/m9.figshare.8185169.v1>, <https://doi.org/10.6084/m9.figshare.8185169.v1>
22. Perez-Monte, C.F., Diedrichs, A.L.: eUCCvm: Plataforma educativa de programación y compilación cruzada universal de sistemas embebidos (7 2023). <https://doi.org/10.6084/m9.figshare.8185169.v10>, <https://doi.org/10.6084/m9.figshare.8185169>
23. Perez Monte, C.F., Diedrichs, A.L.: Programatecnicasdigitalesii-2023.pdf (Aug 2023). <https://doi.org/10.6084/m9.figshare.23895969>, https://figshare.com/articles/preprint/ProgramaTecnicaDigitalesII-2023_pdf/23895969/1

24. Perez Monte, C.F., Diedrichs, A.L.: Programatecnicasdigitalesiii-2023.pdf (Aug2023). <https://doi.org/10.6084/m9.figshare.23895963>, https://figshare.com/articles/book/ProgramaTecnicaDigitalesIII-2023_pdf/23895963/1
25. Perez-Monte, C.F., Mercado, G., Taffernaberry, C., Diedrichs, A.L., Tobar, M.S., Crespillo, E., Moralejo, R., Caballero, G., Gonzalez, R.: eucevm: Plataforma educativa de programación y compilación cruzada universal de sistemas embebidos. CASE 2019 (2019)
26. Prachchhak, G., Bhatt, C., Thik, J.: Data logging and visualization using bolt IoT. In: International Conference on Advanced Computing Networking and Informatics. pp. 155–164. Springer (2019)
27. Ritchie, D.M.: The development of the C language. ACM Sigplan Notices **28**(3), 201–208 (1993)
28. Sanchez, J., Canton, M.P.: Microcontroller programming: the microchip PIC. CRCpress (2006)
29. Silvia Lago Martínez, Ana Marotias, S.A.: Inclusión digital en la educación república argentina. el programa conectar igualdad. Revista Educación y Pedagogía **24**(62), 205–218 (2013), <http://bit.ly/2wm5Ke3>
30. Soper, M.E.: Raspberry Pi system anatomy. In: Expanding Your Raspberry Pi, pp. 1–15. Springer (2017)
31. Yiu, J.: The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors. Newnes (2013)