

eUCCvm 2 - Development of an Educational Tool for Cross-Compilation, Documentation Access, and Embedded Systems Programming

Helvio Hild¹ , Cristian F. Perez-Monte¹ , María Fabiana Piccoli^{2,3} ,
Gustavo Mercado¹ , Ana Laura Diedrichs¹ , Mario Sebastián Tobar¹ ,
Rodrigo González¹ , and Raúl Moralejo¹ 

¹ GridTICs, Universidad Tecnológica Nacional, Facultad Regional Mendoza, Argentina <http://www.gridtics.frm.utn.edu.ar> helviorhild@gmail.com,
{cristian.perez, raul.moralejo}@docentes.frm.utn.edu.ar, {gmercado, mstobar, ana.diedrichs, rodrigo.gonzalez}@frm.utn.edu.ar

² Universidad Nacional de San Luis, Ejército de los Andes 950, San Luis Argentina
<http://www.unsl.edu.ar> mpiccoli@unsl.edu.ar

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

Abstract. In this work, we present *eUCCvm-2*, the evolution of an open-source universal cross-compilation and documentation tool for guided educational programming of Embedded Systems: *eUCCvm*. Through its use, it is possible to develop application programming activities for Embedded Systems across multiple architectures. It can be used in both academic and professional environments, and in any modality: virtual or in-person.

Keywords: Embedded Systems, Architecture Programming, Educational Software, Open Source

eUCCvm 2 - Desarrollo de una Herramienta Educativa de Compilación Cruzada, Acceso a la Documentación y Programación de Sistemas Embebidos

Abstract. En este trabajo presentamos *eUCCvm-2*, la evolución de una herramienta de código abierto de compilación cruzada universal y acceso a documentación confiable para la programación guiada educativa 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.

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

1 Introducción

eUCCvm-2 (*Educational/Embedded/Electronic/Entertainment use Universal Cross Compiler Virtual Machine 2*) es la segunda versión de un sistema basado en una máquina virtual para el desarrollo, compilación y programación de diferentes arquitecturas Ryzhyk, 2006 Shanley, 2010 Waterman et al., 2011 Bossard, 2023 Price, 1995 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. El sistema *eUCCvm-2* está basado en trabajos previos Pérez Monte et al., 2024 que fueron desarrollados bajo el nombre de *eUCCvm*. *eUCCvm* es una máquina virtual lista para usar basada en sistema operativa Linux que incluye compiladores de múltiples arquitecturas, software para transferir o implementar los binarios sobre dichas arquitecturas, documentación y ejemplos de implementación. La herramienta puede ser usada mediante shell de linux, tanto para la programación como para el acceso a la documentación adicional. Actualmente ha sido implementado en una cantidad elevada de arquitecturas incluida x86, x86-64, arm, avr, mips, pic, risc-v, xtensa, 68k, entre otras. Se ha priorizado la implementación de arquitecturas de microcontroladores para su uso en cátedras relacionadas a ingeniería electrónica y arquitecturas vinculadas con el entretenimiento debido a la diversidad de arquitecturas que resultan abarcativas para usarse en cátedras de arquitecturas de microprocesadores. Además la existencia de herramientas de simulación de dichas arquitecturas permite no solo el desarrollo sino la implementación completa y ejecución del software. En el siguiente resumen detallamos los avances realizados sobre *eUCCvm-2*, realizando un análisis comparativo con la versión previa.

2 Características de *eUCCvm-2*

eUCCvm-2 al igual que su predecesor es una herramienta educativa integral, basada en una máquina virtual con todo incluido y lista para ser usada por alumnos en el desarrollo de diferentes actividades, sin la necesidad del acceso a internet ni software adicional. Estas características permiten su uso tanto en la presencialidad del ámbito académico como en la virtualidad. *eUCCvm-2* integra una zona visual o área de trabajo, distribuida en varias ventanas de trabajo donde se realizan las actividades mediante acceso web local utilizando Compiler Explorer Godbolt, 2019. El mismo integra en el área de trabajo seis características principales compuesta de un asistente de configuración (setup wizard) y cinco ventanas de trabajo. Estas características son:

- Asistente de configuración para la elección de ejemplos para la realización de las actividades de acuerdo a los siguientes patrones:
 - Arquitectura de micro-instrucciones.
 - Microcontrolador.
 - Placa de implementación del microcontrolador.

- Lenguaje de programación.
 - Selección de ejemplo de características o de rendimiento.
 - Elección del ejemplo.
- Ventana de trabajo del código del ejemplo elegido.
En dicha área, el alumno visualiza y/o modifica el ejemplo elegido, sin necesidad de requerir un entorno de desarrollo adicional. En dicha área, los comentarios no solo brindarán información textual sobre el ejemplo sino que pueden estar hipervinculados a documentación adicional, la cual puede visualizarse en la ventana dinámica.
 - Ventana de trabajo del código máquina desensamblado.
Aquí, el alumno puede observar como el código escrito en la ventana anterior, al ser compilado sobre la arquitectura elegida, produce un código máquina y su respectivo código ensamblador para comprender su funcionamiento a bajo nivel.
 - Ventana de acceso al hardware.
En esta ventana se visualizará la comunicación con el hardware para permitir transferir el código compilado al hardware.
 - Ventana estática de la documentación de la actividad vinculada al ejemplo.
En esta zona, el alumno tiene un explicativo detallado de la actividad con el procedimiento completo necesario para poder ejecutar el ejemplo sobre una arquitectura determinada. Desde dicho documento también tiene acceso hipervinculado a toda la documentación técnica adicional, set de instrucciones de la arquitectura, microcontrolador, módulo sobre el cual está implementado el microcontrolador entre otros que podrán ser visualizados desde la ventana dinámica.
 - Ventana dinámica de acceso a toda la documentación.
En esta ventana se visualiza toda la documentación hipervinculada, ya sea en el área estática de la documentación principal de la actividad o en comentarios en el código del ejemplo.

A través de estas 5 ventanas el alumno puede desarrollar las actividades de forma autocontenida, con toda la información necesaria sin requerir acceso a Internet. En la imagen 1 pueden observarse las 5 ventanas correspondientes.

3 Ejemplos de Código Disponibles

En el desarrollo de *eUCCvm-2* por cada arquitectura, procesador, placa de implementación y lenguaje de programación se presentan ejemplos de código, los cuales, por razones pedagógicas se dividen en dos tipos:

- Ejemplos de características: Con ellos se busca que el alumno domine una característica específica del hardware, por ejemplo, comunicación serial, uso de conversores adc o dac o uso de gpio.
- Ejemplos de rendimiento: Con ellos se pretende que el alumno haga uso óptimo o eficiente de determinada característica específica del hardware, por ejemplo, transmisión serial a la máxima velocidad del baudrate.

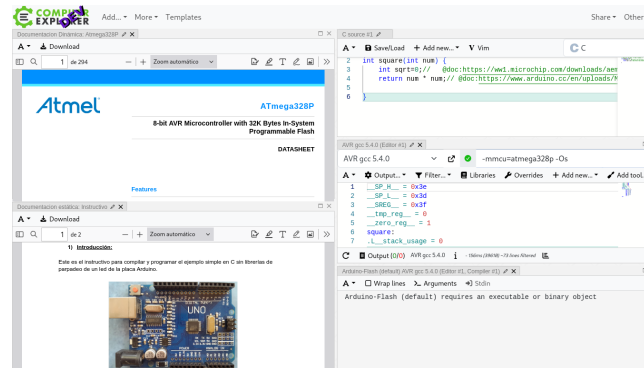


Fig. 1. Espacio de trabajo de *eUCCvm-2*

La utilización de estos dos tipos de ejemplo permiten que el alumno comprenda las características del hardware y como usarlas por un lado, y que pueda extraerles el máximo rendimiento por otro.

4 Documentación en *eUCCvm-2*

A la hora de realizar desarrollos, el acceso a la documentación confiable es fundamental Azevedo and Hadwin, 2005. Para el alumno pueden haber impedimentos a su acceso:

- Falta de conocimiento a la hora de buscar y determinar la fuente de información más confiable.
- Falta de acceso a Internet.
- Información inaccesible en internet por ser hardware obsoleto o de empresas ya no existentes.

En desarrollos de ingeniería no existe una única fuente de información confiable sino que deben utilizarse varias fuentes confiables simultáneamente, algunas de las cuales son:

- Hojas de datos de microcontroladores o microprocesadores publicados por el fabricante
- Esquemáticos u hojas de datos del módulo en el cual está implementado el microprocesador
- Especificaciones de las ISA (es el fabricante Microchip Technology Inc., n.d.Espressif Systems, n.d., la empresa licenciataria de la arquitectura Arm Ltd., n.d. o la organización encargada de publicar sus especificaciones RISC-V International, n.d.
- Especificaciones de señalización eléctrica, protocolos de red, estándares de comunicaciones, etc, publicados por diferentes organizaciones Joint Electron Device Engineering Council, n.d.Institute of Electrical and Electronics Engineers, n.d.Internet Engineering Task Force, n.d.

- Bibliografía especializada para la resolución de problemas específicos.

A diferencia de *eUCCvm*, en donde la documentación se almacenaba junto a la carpeta de los ejemplos de cada arquitectura, en *eUCCvm-2* el acceso a la documentación es estructurada e integrada al área de trabajo para una mayor disponibilidad del alumno.

En *eUCCvm-2* existen dos tipos de documentación:

- Documentación de la actividad: Es el centro cognitivo estable Cognition and at Vanderbilt, 1990 Azevedo and Hadwin, 2005 o punto de partida a utilizar por el alumno, con un paso a paso detallado a la hora de compilar, programar y probar el ejemplo seleccionado. Esta documentación está siempre disponible en una ventana del área de trabajo en el entorno web. Además es una base de organización para acceder a toda la información adicional mediante hipervínculos, de forma que el alumno comprenda mejor el funcionamiento de la implementación del ejemplo.
- Documentación confiable accesoria: Es la expansión contextual o documentación adicional, accesible mediante hipervínculos desde la documentación de la actividad. Si bien puede estar disponible en Internet, se almacena localmente en *eUCCvm-2* para independizarse del acceso a Internet. Debido a la cantidad de documentos que pueden existir por cada actividad, solo se visualiza desde la ventana de documentación dinámica la última seleccionada desde los hipervínculos de la documentación de la actividad o de los comentarios del código del ejemplo. Esta documentación se clasifica en:
 - Documentación de la ISA, por ejemplo set de instrucciones de RISC-V Waterman et al., 2011.
 - Hoja de datos del hardware del procesador, por ejemplo la hoja de datos del microcontrolador de raspberry pi pico 2 Raspberry Pi Ltd., 2024b.
 - Esquemático o documentación de la placa o módulo donde se implementa el procesador, por ejemplo esquemático del raspberry pi pico 2 Raspberry Pi Ltd., 2024a.
 - Manual de programación, por ejemplo manual de programación de raspberry pi pico 2 Raspberry Pi Ltd., 2024c.
 - Estándares de la industria, por ejemplo estándar de wifi presentado por la IEEE, 802.11ax IEEE, 2021.

De esta forma la documentación cumple las tres características requeridas en el sistema educativo: que sea confiable, accesible e inter-relacionada con el resto de las herramientas del entorno.

5 Resultados: Análisis Comparativo de *eUCCvm-2* con *eUCCvm*

La tabla 1 muestra una comparación entre *eUCCvm-2* y *eUCCvm*. La comparación se realizó considerando las características principales y los objetivos planteados para esta herramienta.

Table 1. *eUCCvm* vs *eUCCvm-2*, tabla Comparativa

Características	eUCCvm	eUCCvm-2
Forma de trabajo	Línea de comandos (CLI)	Acceso gráfico (entorno web)
Dificultad de uso	Elevada	Reducida
Dificultad de mantenimiento	Elevada	Reducida
Organización de las actividades y ejemplos	Estructura de carpetas	Acceso gráfico
Acceso a la Documentación	Archivos de documentación por carpeta	Acceso estructurado por nivel hipervinculados
Documentación de la actividad	externa	incluida
Recursos de hardware necesarios de RAM	Muy bajos (> a 230MB de RAM)	Moderados (> a 2GB de RAM)
Recursos de hardware necesarios de Disco	Muy Altos (>32 GB)	Altos (>8 GB)
Software de Virtualización	VirtualBox	Qemu o VirtualBox
Sistema operativo virtual	Linux Ubuntu 18.04 LTS	Linux Ubuntu 24.04 LTS
Arquitectura virtual	x86 de 32 bits	x86-64 de 64 bits
Estado de implementación	Desarrollada	En desarrollo
Arquitecturas implementadas	Mas de 20 arquitecturas	Arquitectura AVR

6 Conclusiones

eUCCvm-2 fue planteado como una extensión, no como reemplazo de *eUCCvm*. Su característica principal es su adaptabilidad, incorporación de nuevas tecnologías y facilidad de uso. Ambos minimizan el recambio constante de hardware. Si bien *eUCCvm-2* requiere un mayor consumo de memoria RAM, su superioridad pedagógica impulsa su adopción en todas las cátedras.

References

- Arm Ltd. (n.d.). *Home*. Retrieved April 17, 2026, from <https://www.arm.com>
- Azevedo, R., & Hadwin, A. F. (2005). Scaffolding self-regulated learning and metacognition—implications for the design of computer-based scaffolds. *Instructional science*, 33(5/6), 367–379.
- Bossard, A. (2023). *Understanding microcontrollers: A gentle introduction to an avr architecture*. Ohmsha Co., Ltd.
- Cognition & at Vanderbilt, T. G. (1990). Anchored instruction and its relationship to situated cognition. *Educational Researcher*, 19(6), 2–10.
- Espressif Systems. (n.d.). *Home*. Retrieved April 17, 2026, from <https://www.espressif.com>
- Godbolt, M. (2019). Compiler explorer: Behind the scenes [Conference presentation]. *Proceedings of CppCon 2019*. <https://www.youtube.com/watch?v=kIoZDUd5DKw>

- IEEE. (2021). *IEEE Std 802.11ax-2021: High Efficiency WLAN (Wi-Fi 6)*. Retrieved April 17, 2026, from https://standards.ieee.org/standard/802_11ax-2021.html
- Institute of Electrical and Electronics Engineers. (n.d.). *Home*. Retrieved April 17, 2026, from <https://www.ieee.org>
- Internet Engineering Task Force. (n.d.). *Home*. Retrieved April 17, 2026, from <https://www.ietf.org>
- Joint Electron Device Engineering Council. (n.d.). *Home*. Retrieved June 26, 2026, from <https://www.jedec.org>
- Microchip Technology Inc. (n.d.). *Home*. Retrieved April 17, 2026, from <https://www.microchip.com>
- Pérez Monte, C., Mercado, G., Taffernaberry, J. C., Diedrichs, A., Piccoli, M. F., Tobar, S., Ledda, M., Moralejo, R. O., & Gonzalez, R. (2024). Euccvm: Una herramienta educativa integral para la programación de sistemas embebidos. *XXIX Congreso Argentino de Ciencias de la Computación (CACIC)(Luján, 9 al 12 de octubre de 2023)*.
- Price, C. (1995). Mips iv instruction set.
- Raspberry Pi Ltd. (2024a). *Raspberry Pi Pico 2: Schematic* [Accessed: 2026-04-17]. <https://datasheets.raspberrypi.com/pico/pico-2-schematic.pdf>
- Raspberry Pi Ltd. (2024b). *RP2350 Datasheet*. Retrieved April 17, 2026, from <https://datasheets.raspberrypi.com/rp2350/rp2350-datasheet.pdf>
- Raspberry Pi Ltd. (2024c). *RP2350 Microcontroller: Programming Guide* [Accessed: 2026-04-17]. <https://datasheets.raspberrypi.com/rp2350/rp2350-programming-guide.pdf>
- RISC-V International. (n.d.). *Home*. Retrieved April 17, 2026, from <https://riscv.org>
- Ryzhyk, L. (2006). The arm architecture. *Chicago University, Illinois, EUA, 1*.
- Shanley, T. (2010). *X86 instruction set architecture*. MindShare press.
- Waterman, A., Lee, Y., Patterson, D. A., & Asanovic, K. (2011). The risc-v instruction set manual, volume i: Base user-level isa. *EECS Department, UC Berkeley, Tech. Rep. UCB/EECS-2011-62, 116*(2011), 1–32.