



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area II: Sistemas de Computacion

(Programa del año 2026)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
SISTEMAS EMBEBIDOS	ING. EN COMPUT.	28/12	2026	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
GROSSO, ALEJANDRO LEONARDO	Prof. Responsable	P.Asoc Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
1 Hs	0 Hs	0 Hs	7 Hs	8 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
11/03/2026	23/06/2026	15	120

IV - Fundamentación

Los sistemas incrustados o incorporados en otros sistemas (embedded systems) están presentes en la mayoría de los dispositivos con los que interactuamos: electrodomésticos, automóviles, aviones, enrutadores, impresoras, etc.. En la mayoría de estos sistemas la ausencia de fallas y un tiempo de respuesta preciso son requerimientos indispensables para su funcionamiento. Por lo tanto es necesario utilizar, y en algunos casos desarrollar, herramientas adecuadas para la especificación, diseño e implementación del sistema incrustado en otro systema (embedded system). En este curso introduciremos algunas herramientas que ayudan al diseño y desarrollo de los sistemas incrustados (embedded systems).

V - Objetivos / Resultados de Aprendizaje

- Introducir métodos para el diseño de sistemas incrustados (embedded systems). Introducir métodos formales para el desarrollo de sistemas incrustados (embedded systems).
- Utilizar los métodos anteriores para desarrollar sistemas concretos sobre plataformas concretas y comprobar su funcionamiento.
- Conocer distintas plataformas utilizadas en el desarrollo de sistemas incrustados (embedded systems).

Durante el dictado de la materia se abordan los siguientes ejes transversales:

- Concepción, diseño y desarrollo de proyectos de ingeniería en computación.
- Utilización de técnicas y herramientas de aplicación en la ingeniería en computación.
- Fundamentos para el desempeño en equipos de trabajo.
- Fundamentos para una comunicación efectiva.
- Fundamentos para una actuación profesional ética y responsable.

-Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local.

VI - Contenidos

Contenidos mínimos:

Modelaje, diseño y realización de sistemas incrustados (embedded systems), diferencia metodológica entre programación reactiva y transformacional, interfaces de comunicación serial, utilización de sensores y actuadores. Características del lenguaje de programación necesarias para los sistemas incrustados. Realización del modelaje con statecharts en lenguajes de programación imperativos. Conceptos de sistemas de tiempo real, tareas, prioridades y planificación.

Programa Analítico

Unidad 1: Sensores y Actuadores

Sensores: Rango, rango dinámico, cuantización, ruido, muestreo, sensores comunes: medición de inclinación y aceleración, medición de posición y velocidad, otros sensores. Actuadores: diodos emisores de luz, parlantes, relés.

Unidad 2: Procesadores.

Introducción Cortex-M4 de ARM: elementos básicos, registros, mapa de memoria, instrucciones, interrupciones.

Introducción a Cortex-M4. Procesadores multicore simétricos y asimétricos.

Análisis de documentación de fabricantes. Plataforma de desarrollo. Introducción a un entorno de desarrollo para microcontroladores de 32 bits. Análisis del hardware empleado en el curso.

Unidad 3

Interfaces del microcontrolador utilizadas para la adquisición y almacenamiento de datos, y, para control y comunicación con otros microcontroladores y periféricos.

Unidad 4: Modelos reactivos y transformacionales.

Diferencia entre sistemas transformacionales y reactivos.

Introducción a los StateCharts (Diagramas de estados): estados, eventos, jerarquía de máquinas, superestados, subestados, estados OR, estados AND. Modelado utilizando StateCharts. Introducción de herramientas que generan código C a partir de una especificación que utiliza StateCharts.

Unidad 5: Sistemas operativos para sistemas incrustados (embedded systems)

Predicción de los tiempos de ejecución. Planificación en sistemas de tiempo real. Clasificación de los algoritmos de planificación. Sistemas Operativos incrustados (embedded operating systems).

Unidad 6: Programación de microcontroladores de 32 bits en C.

Análisis en bajo nivel de la asignación de memoria. Estructura de programas reactivos sin sistemas operativos de tiempo real. Especificación de drivers. Generación de eventos. Estructura de sistemas incrustados (embedded systems) que usan Sistemas Operativos de Tiempo Real.

VII - Plan de Trabajos Prácticos

Para la resolución de los prácticos propuestos, los alumnos utilizan conocimientos de: programación, estructuras de datos y algoritmos, sistemas operativos y procesamiento de señales introducidos en las correspondientes áreas curriculares de la carrera de Ingeniería en Computación. Para el diseño de los algoritmos el alumno se apoya en la metodología de statecharts que utiliza tanto refinamiento vertical como horizontal que forman parte de los ejes transversales. En la mayoría de los ejercicios los alumnos deben utilizar puertos de entrada-salida (eje transversal con arquitectura del procesador) y las operaciones booleanas del lenguaje C (eje trans. programación I). La materia es intensiva en prácticos de laboratorios y para ello se utilizan las placas disponibles en el laboratorio de sistemas incrustados (embedded systems).

Para la comunicación efectiva los alumnos presentan los diseños utilizando las herramientas formales de diseño introducidas en la asignatura.

Para el trabajo en equipo los alumnos desarrollan diferentes programas controladores de dispositivos (drivers) que luego intercambian con sus pares.

Para una actuación profesional ética y responsable se les exige soluciones personales de los prácticos y cumplimiento de los plazos de entrega. Además se hace responsable de las correcciones y mantenimiento del código compartido con sus pares.

Los docentes evalúan los prácticos entregados por los alumnos para verificar el funcionamiento efectivo e individual de la solución. Todos los prácticos deben ser aprobados con el 70% o más.

Los prácticos más avanzados además de servir como ejemplos de desarrollos tecnológicos en el área también contribuyen a la toma de conciencia del aumento de productividad que produce la incorporación de tecnología, su impacto social en el trabajo y en la competencia de las empresas o emprendedores.

Práctico N°1:

Titilar un led con demora ocupada. Implementar el diseño en lenguaje C utilizando la plataforma EDU-CIAA. Con este práctico evaluamos entrada-salida programada y el uso de operaciones booleanas del lenguaje C vistas en Arq. del Procesador y en Programación.

Práctico N°2:

Titilar un led definiendo el período. Utilizando la metodología de statecharts, implementar el diseño en lenguaje C utilizando demoras no bloqueantes en la plataforma EDU-CIAA.

Con este práctico evaluamos el refinamiento vertical y horizontal cuando usan los statecharts para modelar utilizan conocimientos de: programación I y II, arquitectura del procesador y sistemas operativos para implementar una demora no bloqueante.

Práctico N°3:

Interactuar con los periféricos a través de las distintas interfaces provistas por el microcontrolador NXP LPC4335 (ARM Cortex-M4/M0) de la placa EDU-CIAA. Aquí evaluamos e-s programada y por interrupciones, conceptos vistos en arq. del procesador y en sistemas operativos, como así también la realización en C, lenguaje visto en programación I.

Práctico N°4:

Asociar un GPIO a una interrupción del microcontrolador NXP LPC4335 (ARM Cortex-M4/M0) de la placa EDU-CIAA. La evaluación es similar al práctico N°3.

En los prácticos que siguen los modelos deben ser realizados utilizando la metodología de los statecharts e implementados en lenguaje C.

Se evalúan conocimientos de: programación, estructuras de datos y algoritmos, sistemas operativos y procesamiento de señales introducidos en las correspondientes áreas curriculares de la carrera de Ingeniería en Computación. Para el diseño de los algoritmos el alumno se apoya en la metodología de statecharts que utiliza tanto refinamiento vertical como horizontal que forman parte de los ejes transversales del área de programación y metodologías.

Práctico N°5:

Utilizando el DAC (convertor analógico-digital) generar un bitono utilizando E/S programada, E/S utilizando interrupciones y E/S con DMA y buffers. En este práctico también se evalúan conceptos vistos en señales y sistemas; y electrónica analógica.

Práctico N°6:

Idem al práctico N°5 pero utilizando un TIMER para programar el tiempo de conversión del DAC. El TIMER debe estar conectado a las interrupciones.

Práctico N°7:

Led con atenuación utilizando PWM (pulse-width modulation) a partir de la intensidad definida por un potenciómetro. Programación de Timers y Timers en modo captura. En este práctico también se evalúan conceptos de electrónica analógica y digital.

Práctico N°8:

Antirrebotes de llaves (debouncing). Utilizando la plataforma de la EDU-CIAA.

Práctico N°9:

Implementar un proceso de pesado de precisión.

Práctico N°10:

Implementar un sistema de control de ascensor.

Práctico N°11:

Comunicaciones en bus serie: SPI e I2C. Utilización de un magnetómetro y un servo motor angular para implementar una brújula. Aquí también evaluamos conceptos introducidos en el área de comunicaciones.

VIII - Regimen de Aprobación

La asignatura es promocional.

- a) El alumno debe asistir al 80% de las clases teóricas y prácticas, aprobar todos los laboratorios con al menos el 70%.
- b) El alumno promociona si cumple el punto a) y aprueba el parcial o alguna de sus dos recuperaciones con al menos el 70%.
- c) El alumno regulariza la asignatura si cumple el punto a) y aprueba el parcial o alguna de sus recuperaciones con una nota entre 4 (cuatro) y 6,99 (seis c/99 centésimas).
- d) En caso de no cumplir con los requisitos del punto b) o c) el alumno no regulariza la materia.

IX - Bibliografía Básica

- [1] HOHL, W. and HINDS, C. ARM Assembly Language 2nd. edition. ED. CRC Press; 2015.
- [2] P. Marwedel; Embedded System Design; Springer; 2006
- [3] D.E. Simon; An Embedded Software Primer; Addison-Wesley; 1999
- [4] Q. Li, C. Yao; Real-Time Concepts for Embedded Systems; CMP Books; 2003
- [5] J. Yiu. The Definitive Guide to the ARM® Cortex-M3 (Second Edition). Newnes. 2010.
- [6] E. Lee, S. Seshia. Introduction to Embedded Systems A Cyber-Physical Systems Approach Second Edition. MIT Press, 2017.
- [7] D. HAREL. STATECHARTS: A VISUAL FORMALISM FOR COMPLEX SYSTEMS. Science of Computer Programming 8 (1987) 231-274. North-Holland

X - Bibliografía Complementaria

- [1] S.C. McConnell; Code Complete: A Practical Handbook of Software Construction; Microsoft Press, 2nd ed., 2004
- [2] T. Noergaard; Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers; Newnes; 2005
- [3] J. Labrosse et. al.; Embedded Software: Know It All; Newnes; 2008
- [4] M. Samek. Practical UML Statecharts in C/C++ Second Edition. Newnes. 2009.

XI - Resumen de Objetivos

Introducir métodos para el diseño de sistemas incrustados (embedded systems).

Conocer distintas plataformas utilizadas en el desarrollo de sistemas incrustados (embedded systems).

XII - Resumen del Programa

Sensores y actuadores. Procesadores. Modelos reactivos. Sistemas operativos para sistemas incrustados (embedded systems).

Programación de microcontroladores de 32 bits.

XIII - Imprevistos

Comunicarse con la cátedra. (agrosso@email.unsl.edu.ar)

Sistemas Embebidos.

Departamento de Informática.

Of. 25. Bloque II. 1er. Piso.

Facultad de Cs. Fisico, Matemáticas y Naturales.

Universidad Nacional de San Luis.

Ejercito de los Andes 950. CP 5700.

XIV - Otros