



PROGRAMA DE ASIGNATURA

Nombre del curso o actividad	DISEÑO DE <i>HARDWARE</i> Y <i>SOFTWARE</i> PARA ROBOTS
Nombre del programa	Doctorado en Ciencias de la Ingeniería mención Automática
Descripción	<i>Código: 11315, Nivel: 2 o 3, Créditos SCT-Chile: 10, Tipo: Electiva, Plan de Estudios: Resolución N°1118 del 03/04/20.</i> Desarrollar competencias en el diseño e implementación de sistemas embebidos para aplicaciones robóticas, integrando conocimientos de arquitecturas de <i>hardware</i>, programación de <i>firmware</i>, sistemas operativos en tiempo real, y diseño de interfaces para sensores y actuadores, con énfasis en metodologías de co-diseño <i>hardware-software</i> que permitan crear soluciones robóticas eficientes, confiables y escalables.
Objetivos	Objetivo General: Desarrollar competencias avanzadas en el diseño e implementación de sistemas embebidos para robótica, integrando arquitecturas de <i>hardware</i>, programación de <i>firmware</i> y sistemas operativos en tiempo real, para crear soluciones tecnológicas innovadoras que respondan a las demandas actuales de la automatización industrial. Objetivos Específicos: • Analizar arquitecturas de microcontroladores y procesadores para sistemas robóticos embebidos. • Diseñar e implementar <i>firmware</i> para control de robots utilizando lenguajes de bajo nivel. • Integrar sensores y actuadores mediante interfaces estándar (I2C, SPI, UART, CAN). • Implementar Sistemas Operativos en Tiempo Real (RTOS) para gestión de tareas concurrentes. • Aplicar técnicas de co-diseño <i>hardware-software</i> para optimización de recursos. • Desarrollar prototipos funcionales de sistemas robóticos con arquitecturas embebidas.
Contenidos	Unidad 1: Arquitecturas de Sistemas Embebidos • Introducción a sistemas embebidos: características y aplicaciones. • Arquitectura de microcontroladores ARM Cortex-M. • Sistemas embebidos basados en Linux (Raspberry Pi, BeagleBone). • Sistemas de desarrollo y herramientas de programación. Unidad 2: Programación de <i>Firmware</i> • Programación en C/C++ para sistemas embebidos. • Gestión de memoria y optimización de código. • Programación de periféricos: GPIO, PWM, ADC, DAC. • Manejo de interrupciones y <i>timers</i>. Unidad 3: Integración de Sensores y Actuadores • Protocolos de comunicación: I2C, SPI, UART, CAN. • Integración de sensores: IMU, <i>encoders</i>, sensores de distancia. • Control de actuadores: motores DC, servomotores, motores paso a paso. • Drivers y bibliotecas de <i>software</i> para periféricos. Unidad 4: Sistemas Operativos en Tiempo Real



	• Conceptos de sistemas en tiempo real. • FreeRTOS: arquitectura y componentes. • Gestión de tareas, semáforos y colas de mensajes. • Planificación de tareas y análisis de tiempos de respuesta. Unidad 5: Co-diseño <i>Hardware-Software</i> • Metodología de co-diseño HW/SW. • Particionamiento <i>hardware-software</i>. • Aceleradores hardware y optimización de desempeño. • Diseño de PCB para sistemas robóticos.
Modalidad de evaluación	Estrategias Metodológicas: • Clases expositivas sobre arquitecturas de microcontroladores, protocolos de comunicación y metodologías de co-diseño <i>hardware-software</i>. • Laboratorios de programación de firmware en C/C++ para plataformas embebidas (STM32, ESP32, Raspberry Pi). • Talleres de integración de sensores y actuadores mediante protocolos I2C, SPI, UART y CAN. • Desarrollo de aplicaciones con FreeRTOS para gestión de tareas concurrentes. • Proyectos de prototipado de sistemas embebidos robóticos con metodologías ágiles. • Trabajo autónomo: estudio de documentación técnica, programación y desarrollo de proyectos. Evaluación: • Prácticas programadas: Actividades de programación de <i>firmware</i>, integración de periféricos e implementación de RTOS. Entrega vía repositorio Git. • Proyecto de desarrollo: Sistema embebido robótico completo incluyendo diseño, programación, integración de sensores/actuadores, RTOS y documentación técnica. • Demostración y defensa: Presentación oral con demostración en vivo del prototipo funcional.
Resultados de aprendizajes esperados	Resultado de Aprendizaje General: Al finalizar el curso, el(la) estudiante será capaz de diseñar e implementar sistemas embebidos completos para robótica, integrando arquitecturas de <i>hardware</i>, programación de <i>firmware</i>, sistemas operativos en tiempo real y periféricos, demostrando competencia en el co-diseño <i>hardware-software</i> para crear soluciones robóticas eficientes, confiables y escalables. Resultados de Aprendizaje Específicos: • Seleccionar y configurar arquitecturas de microcontroladores (ARM Cortex-M, ESP32) y procesadores embebidos (Raspberry Pi, BeagleBone) adecuadas para aplicaciones robóticas específicas, evaluando criterios de desempeño, consumo energético y disponibilidad de periféricos. • Programar <i>firmware</i> eficiente en C/C++ con gestión óptima de memoria, interrupciones y recursos computacionales, aplicando técnicas de optimización de código para sistemas embebidos. • Integrar sensores (IMU, <i>encoders</i>, sensores de distancia) y actuadores (motores DC, servos, <i>stepper</i>) mediante protocolos de comunicación estándar I2C, SPI, UART y CAN, implementando drivers y bibliotecas de software. • Implementar sistemas operativos en tiempo real (FreeRTOS) para gestión de tareas concurrentes, sincronización mediante semáforos y comunicación por colas de mensajes, garantizando determinismo temporal.



	• Aplicar metodologías de co-diseño <i>hardware-software</i> para particionar funcionalidades, optimizar recursos y desarrollar prototipos funcionales de sistemas embebidos robóticos validados experimentalmente.
Bibliografía	Básica: • E. White, <i>Making Embedded Systems: Design Patterns for Great Software</i>. Sebastopol, CA, USA: O'Reilly Media, 2011. • R. Barry, <i>Mastering the FreeRTOS Real Time Kernel</i>. Real Time Engineers Ltd., 2016. • T. Bräunl, <i>Embedded Robotics: From Mobile Robots to Autonomous Vehicles with Raspberry Pi and Arduino</i>, 4th ed. Singapore: Springer, 2021. • A. J. Moshayedi et al., <i>Unity in Embedded System Design and Robotics: A Step-by-Step Guide</i>. Boca Raton, FL, USA: CRC Press, 2023. • Artículos científicos relacionados con el tema de tesis del(de la) estudiante. Recomendada: • M. Barr and A. Massa, <i>Programming Embedded Systems: With C and GNU Development Tools</i>, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2006. • N. Kakoty, R. Goswami, and R. Vinjamuri, <i>Introduction to Embedded Systems and Robotics: A Practical Guide</i>. Cham, Switzerland: Springer, 2024.