

EXP-UNC 0058325/2019

Anexo de la RCD FAMAFA 413/2019, página 15 de 33

<b>TÍTULO:</b> Microcontroladores - Sistemas embebidos			
<b>AÑO:</b> 2020	<b>CUATRIMESTRE:</b> 1°	<b>N° DE CRÉDITOS:</b> 3	<b>VIGENCIA:</b> 3 años
<b>CARGA HORARIA:</b> 60 horas de teoría, 60 horas de práctica			
<b>CARRERA/S:</b> Doctorado en Física, Doctorado en Ciencias de la Computación			

<p><b>FUNDAMENTOS</b></p> <p>Los avances en tecnologías tales como la fabricación de circuitos integrados CMOS, sistemas micro-electromecánicos (MEMs), y sistemas de comunicación por radio frecuencia (RF), han revolucionado la concepción tradicional que tenían los sistemas computacionales y las redes de información. La extraordinaria reducción en el tamaño y el consumo de los circuitos electrónicos actuales, permite que dispositivos con capacidades de procesamiento y comunicación encuentren todos los días nuevas áreas de aplicación. Este contexto ha dado lugar a la formación de nuevas áreas de investigación, conocidas bajo el nombre de “computación ubicua”, y más recientemente rebautizadas bajo el nombre de “Internet de las cosas” (IoT, Internet of Things). En el terreno práctico, la IoT es entendida como la integración de la informática en el entorno de la persona y los procesos, de forma que los sistemas computacionales no se perciban como objetos diferenciados. Consiste en la creación de una serie de objetos de uso cotidiano con cualidades interactivas no invasivas, y su objetivo básico es el dotar a estos objetos de capacidades de adquisición de información (tanto del entorno físico como del estado actual del objeto), procesamiento y comunicación, de tal forma que puedan comunicarse entre ellos y ofrecer nuevos servicios a sus usuarios.</p> <p>Las plataformas más utilizadas en el desarrollo de este tipo de dispositivos son las basadas en sistemas embebidos (SE): dispositivos electrónicos compactos y autónomos, con capacidad de cómputo, que realizan procesamiento de datos y/o control sobre variables físicas externas. La mayor diferencia que presenta un SE, respecto de una computadora personal (PC), es que un SE está dedicado a una función particular para la cual fue desarrollado, mientras que un PC está concebido para usos múltiples. Además, los recursos de hardware que dispone un SE son generalmente más reducidos y deben afrontar importantes restricciones de consumo.</p> <p>Los sistemas embebidos forman parte activa en la solución de problemas reales, en campos tan diversos como la industria, la robótica, la medicina, las telecomunicaciones, etc. El mundo físico se basa en variables análogas las cuales se deben convertir en forma digital para ser procesadas y poder controlar diferentes tipos de procesos. En este sentido la instrumentación científica no está exenta de transformaciones ante estos cambios de paradigma. Es importante que el profesional científico y técnico esté familiarizado con este tipo de tecnología, ya que le abre un campo nuevo para dar soluciones a medida en su entorno de trabajo, mediante sistemas electrónicos de bajo costo, con capacidad de adquisición y generación de señales de gran exactitud y precisión.</p>
--

<p><b>OBJETIVOS</b></p> <p><b>OBJETIVO GENERAL</b></p> <p>Que el alumno sea capaz de abordar el diseño e implementación del software específico de sistemas embebidos de tiempo real sobre una plataforma de microcontroladores, orientados al desarrollo de sistemas de control de procesos e instrumentación científica.</p>
--

#### OBJETIVOS ESPECÍFICOS

Conocer arquitecturas actuales de microcontroladores y el funcionamiento de los principales bloques de periféricos e interface.

Utilizar lenguajes estructurados en el desarrollo de aplicaciones de sistemas embebidos, atendiendo criterios de reutilización y portabilidad de código.

Reconocer la necesidad de implementación de sistemas de tiempo real.

Introducir al alumno en la utilización de sistemas operativos en tiempo real para el desarrollo de aplicaciones de mediana/alta complejidad.

Realizar prácticas de laboratorio y proyectos de control e instrumentación sobre plataformas reales, a fin de afianzar los conocimientos adquiridos.

#### PROGRAMA

##### **Unidad 1: Introducción a los sistemas embebidos.**

Definición. Importancia. Términos y límites. Áreas de aplicación. Características. Requerimientos. Fases de diseño.

##### **Unidad 2: Plataformas de sistemas embebidos.**

Arquitecturas de microcontroladores de 16 y 32 bits. Arquitectura ARM Cortex M (32 bits). Concepto de Systems on-chip (SoC). Mapeo de memoria. Bloques generadores de sincronismos (CLKs). Reset e interrupciones. Buses de periféricos. DMA. Modos de bajo consumo.

##### **Unidad 3: Desarrollo de firmware.**

Repaso de lenguaje C. Recursos del lenguaje C para sistemas embebidos. Compiladores. El preprocesador y directivas del compilador. Uso de manejo de bits mediante máscaras. Tipos de datos. Tipos de variables. Constantes. Arreglos, estructuras y uniones. Conceptos avanzados sobre funciones, parámetros por referencia y punteros a funciones. Implementación de rutinas de servicio de interrupciones (ISR).

##### **Unidad 4: Manejo de módulos periféricos.**

Puertos de entrada/salida (GIPO): estructura básica, registros asociados, uso mediante “pooling” e interrupción. Utilización del TIMER como generador de base de tiempos. Módulos de comparación y captura. Modulación PWM. Conversores A/D: estructura básica, configuración y modos de uso. Comunicación serial: UART, I2C, SPI.

##### **Unidad 5: Organización del código.**

Organización del código. Desarrollo mediante capas de abstracción: App, BSP, HAL. Modelado de librerías de manejo de hardware (HAL).

##### **Unidad 6: Sistemas de tiempo real.**

Conceptos de tiempo real. Definición. Multitarea en pequeños sistemas embebidos. Recursos de hardware asociados. Soluciones de compromiso. Terminología. Ejemplos de aplicación. Restricciones de tiempo: tiempo límite, tiempo de ejecución, tiempo de respuesta, jitter, etc.

**Unidad 7: Sistemas operativos de tiempo real (RTOS).**

Tarea, algoritmos de planificación (cooperativos / preemptivos), kernel, TBC, servicios. Gestión de memoria heap: asignación dinámica de memoria. Gestión de tareas: estado de tareas, prioridades, la tarea idle. FreeRTOS: servicios API para gestión de tareas. Delays. Colas (Queue): compartiendo datos entre tareas. Acceso (bloqueante / no bloqueante) desde múltiples tareas. Servicios API para la gestión de colas. Software timers (SWT): usos, atributos, estados y eventos. Funciones Callback. Gestión de interrupciones: servicios del RTOS para una ISR. Semáforos: tipos (binarios y contadores), usos, servicios.

**PRÁCTICAS**

El curso es fundamentalmente práctico y se orienta a una metodología de trabajos de laboratorio. El propósito es colocar a los estudiantes en un ambiente real de diseño de sistemas embebidos. Para ello se hará uso de placas de desarrollo basadas en microcontroladores ARM Cortex M con módulos periféricos de adquisición de datos, comunicación y de interfaz, entre otros.

El curso se estructura alrededor de las prácticas de laboratorio y los proyectos que serán utilizados para evaluación de los alumnos. Estas prácticas buscan aplicar conceptos específicos sobre sistemas embebidos además de generar librerías de código que puedan ser utilizados en posteriores desarrollos.

**CLASES PRÁCTICAS:**

Cuatro horas semanales. Corresponden a prácticas de laboratorio, donde se presentarán consignas y soluciones a problemas típicos de control e instrumentación, implementados mediante sistemas embebidos, haciendo uso de plataformas e instrumental disponibles en el laboratorio de electrónica. Además se coordinará la realización de un proyecto integrador hacia el final del curso, a desarrollarse en grupos de un número reducido de estudiantes.

**BIBLIOGRAFÍA**

“Embedded Systems with Arm Cortex-M Microcontrollers in Assembly Language and C: Third Edition”, Yifeng Zhu. Ed. E-Man Press LLC, 2017. ISBN: 0982692668.

“Mastering the FreeRTOS™ Real Time Kernel: A Hands-On Tutorial Guide”, Richard Barry. Pre-release 161204 Edition, Real Time Engineers Ltd, 2018.

**MODALIDAD DE EVALUACIÓN**

Los criterios de evaluación serán:

- a) La calidad de los conocimientos teóricos y prácticos adquiridos por el alumno.
- b) La Integración de conocimientos.
- c) El desarrollo de capacidades, habilidades y destrezas para el planteo y solución de problemas.
- d) Asistencia a clases y participación activa.



EXP-UNC 0058325/2019

Anexo de la RCD FAMAF 413/2019, página 18 de 33

#### FORMAS DE EVALUACIÓN

Los exámenes parciales serán instrumentados mediante la presentación de los trabajos prácticos y proyectos especiales de laboratorio, junto a sus informes correspondientes, realizados durante el desarrollo de la materia en las horas prácticas.

El examen final constará de la evaluación de un Trabajo Integrador, que consiste en el desarrollo de un sistema embebido de mediana complejidad, de aplicación práctica, implementado mediante la utilización de un RTOS, que resuelva un problema real. Se prevé la evaluación de su informe escrito correspondiente y una defensa oral del trabajo realizado.

#### REQUERIMIENTOS PARA EL CURSADO

Conocimientos básicos de programación en lenguaje estructurado, preferentemente "C"