

Programa de

Sistemas Embebidos Avanzados

Código:



Identificación y características de la Actividad Curricular

Carrera/s:	Ingeniería Electrónica		
Plan de Estudios:	2014	Carácter:	Optativa
Bloque:	TECNOLOGÍAS APLICADAS	Área:	SISTEMAS DIGITALES
Régimen de cursado:	Cuatrimestral		
Cuatrimestre:	9		
Carga horaria:	96 hs	Formato curricular:	Espacio Curricular Electivo
Escuela:	Electrónica	Departamento:	Sistemas e Informática
Docente responsable:	Javier G. Belmonte		

Programa Sintético

Estructura de software para Sistemas Embebidos de alto desempeño. Administración de los recursos de hardware. Los sistemas operativos y el desarrollo de sistemas embebidos. Construcción del software. Herramientas de versionado. Implantación *bare metal*. Implantación de sistemas embebidos sobre Linux, FreeRTOS y Android. Diseño y programación del sistema embebido. Herramientas de desarrollo. El paradigma Internet de las Cosas (IoT). Tecnologías actuales. Desarrollo de sistemas embebidos para IoT.

Actividades Curriculares Relacionadas

Previas Aprobadas: Sistemas Digitales II

Simultáneas Recomendadas:

Posteriores:

Aprobado por Resolución CD N°:

Características Generales

Esta actividad curricular se encuadra como materia Optativa (optativa II) dentro del Espacio Curricular Electivo del plan de estudios de la carrera de Ingeniería Electrónica. Tiene por fin profundizar en los conocimientos necesarios para el desarrollo de soluciones digitales avanzadas basadas en microcontroladores. Su desarrollo contempla el aprendizaje y la utilización de las tecnologías de software y hardware que se requieren para abordar la construcción de sistemas embebidos de alto desempeño, abarcando desde el análisis de las arquitecturas necesarias para soportar la constitución de estos sistemas y el estudio de los conceptos de diseño más actuales, hasta la efectiva aplicación de la tecnología hoy disponible mediante el uso de aquellos.

La asignatura avanza sobre su contenido a partir del planteo de las estructuras de software requeridas para constituir sistemas embebidos avanzados basados en sistemas operativos subyacentes, para seguir luego con la implantación embebida de los sistemas operativos hoy utilizados, para diferentes paradigmas de uso, y finalmente encarar el desarrollo de la solución sobre ellos, siempre apelando a criterios de diseño que permitan la creación de un sistema digital sólido, eficiente y alineado con el fin o aplicación pretendida.

Para terminar: las tendencias actuales respecto de conectividad a todo nivel han impactado fuertemente en el ambiente de los sistemas embebidos. Las aplicaciones tecnológicas en donde los sistemas digitales conversan entre sí en forma directa o vía Internet están en un punto de máxima expansión. Atendiendo a esta realidad en la que los sistemas embebidos avanzados están inmersos y cumplen un rol fundamental, la asignatura aborda la comunicación entre estos sistemas en forma ejemplificada a través del paradigma más actual, esto es: Internet de las Cosas (del inglés, IoT). Este abordaje se hace primero estudiando los protocolos y tecnologías que dan base a IoT para luego pasar a la implementación efectiva del sistema embebido mediante el uso de la oferta de productos y servicios, como insumos, existentes en la actualidad, cerrando de esta forma un contenido valioso y a la vez atractivo para el alumno que pretende expandir sus conocimientos dentro del Área Digital.

Objetivos

Que los alumnos sean capaces de:

Entender los conceptos fundamentales relativos a la arquitectura del software necesario para constituir sistemas embebidos de altas prestaciones.

Conocer y utilizar herramientas de desarrollo colaborativo y versionado.

Utilizar las herramientas actuales para implantar eficientemente sistemas operativos aptos para soluciones embebidas.

Diseñar e implementar sistemas embebidos basados en diferentes sistemas operativos acorde a los requerimientos de prestaciones a cumplir, incluyendo aplicaciones de tiempo real.

Conocer las tendencias actuales en cuanto a conectividad de sistema embebidos.

Dados ciertos requerimientos o especificaciones, seleccionar componentes de hardware, protocolos y servicios para constituir soluciones digitales basadas en sistemas embebidos interconectados que los satisfagan eficientemente.

Contenido Temático

1. UNIDAD 1: ESTRUCTURA DE SOFTWARE EN SISTEMAS EMBEBIDOS

1.1. Introducción. El software en un SE. Arquitectura. Portabilidad y eficiencia. Desarrollo *bare metal* vs. soportado bajo un S.O.

1.2. Administración y control de los recursos de hardware: Administración de dispositivos de entrada/salida. Administración de la memoria. Niveles de abstracción. Uso y desarrollo de Interfaces de software: Hardware Abstraction Layer, Librerías, Software development Kits, Device Drivers, Application Program Interfaces.

- 1.3. Sistemas operativos: conceptos generales. Tipos de S.O. Estructura de un Sistema Operativo. Elementos constituyentes. Gestión de recursos, planificación y tiempo. APIs del S.O. Criterios de selección y utilización de SO para SE.
- 1.4. Diseño y construcción de software en SE: Modelos de desarrollo. Herramientas de programación, entornos integrados de desarrollo y "toolchains". Herramientas de versionado. Desarrollo colaborativo. Repositorios.

2. UNIDAD 2: DISEÑOS EMBEBIDOS SOPORTADOS EN SISTEMAS OPERATIVOS

2.1. Implantación de SE con Linux embebido.

Requisitos de hardware. Arquitectura del SE con Linux. El kernel de Linux. Bootloaders. Filesystems. Imágenes de FS: herramientas de generación. Sistemas de archivos para memorias flash. Diseño y programación del SE. Depuración.

2.2. Implantación de SE bajo FreeRTOS.

Sistemas operativos de tiempo real. Caracterización y Tipos. Requisitos del hardware. *Scheduller*. Las tareas y sus prioridades. Mecanismos de acceso compartido a recursos. Manejo de interrupciones. Las APIs de FreeRTOS. Diseño y programación del SE.

2.3. Implantaciones bajo Android.

La arquitectura de Android. Estructura de los proyectos. Componentes de las aplicaciones. Interfaz del usuario. Gestión de recursos. Notificaciones. Servicios en background. Uso de C/C++ con NDK. Conectividad e Internet.

3. UNIDAD 3: SISTEMAS EMBEBIDOS E INTERNET: INTERNET DE LA COSAS (IoT)

3.1. Conceptos fundamentales sobre IoT. Evolución de los tipos de conexiones en Internet. Arquitectura de IoT. Areas actuales de aplicación. Las capas de comunicaciones en IoT. Protocolos de enlace, red y transporte utilizados en IoT. Seguridad y calidad de servicio. Tecnologías aplicadas a IoT. Dispositivos ad-hoc para IoT.

3.2. Presentación de la Información y montaje de APIs. APIs locales y remotas. Manejo de servidores y bases de datos orientados a aplicaciones multiplataforma. Node-Red, NodeJS, webservices, dashboards, respuesta a eventos. Oferta actual de Servicios.

3.3. Plataformas de desarrollo para IoT. La oferta actual del mercado. Plataformas *bare metal* y basadas en sistemas operativos embebidos. Herramientas de desarrollo.

3.4. Desarrollo de la aplicación. Definición conceptual. Criterios para selección de la tecnología. Selección de hardware y protocolos. Diseño de la presentación de los datos.

Modalidades de enseñanza-aprendizaje

Las clases son de tipo teórico-práctico. Se utilizan diversas metodologías de enseñanza-aprendizaje:

- Exposición de núcleos teóricos fuertes
- Planteo de situaciones tecnológicamente problemáticas y su resolución/implementación
- Descripción de códigos ejemplificadores.
- Uso práctico de herramientas de diseño, configuración y desarrollo
- Desarrollo de programas para microcontrolador sobre sistemas operativos.
- Realización de implementaciones de ejemplo
- Realización de Trabajos Prácticos utilizando diferentes kits de desarrollo (diferentes plataformas)
- Consultas individuales

En las clases se integran los aspectos conceptuales, teóricos e informativos del contenido temático de la materia con las actividades de índole práctico, como son el planteo de requerimientos y la posterior formulación de una solución, la producción de sistemas embebidos sobre kits de desarrollo utilizando diferentes entornos y herramientas, y la realización de trabajos prácticos sobre los diferentes tópicos y plataformas que demandan los contenidos temáticos. Estas actividades de índole práctico se realizan, según el caso, en forma individual o grupal, contando para el desarrollo de las mismas con kits de desarrollo profesionales. Los trabajos prácticos se realizan en grupos de dos estudiantes. Tanto en el desarrollo de las actividades de carácter teórico como las de índole práctica se intenta siempre acercar los planteos a su implicancia o impacto en la aplicación o implementación tecnológica final, priorizando y promoviendo el desarrollo de la creatividad, planteando sistemáticamente la resolución de situaciones

problemáticas reales y actuales.

Se busca entonces darle a la asignatura un enfoque orientado al diseño creativo y la aplicación práctica del conocimiento.

A los tiempos previstos para la carga horaria destinada a la materia dentro del plan, debe agregarse la disponibilidad de no menos de 4 horas semanales de consulta que los docentes en conjunto brindan a los estudiantes fueran del horario regular de cursado y que están destinadas a la clarificación de conceptos, explicaciones puntuales o soporte para la resoluciones de las prácticas planteadas.

Actividades de Formación Práctica

Las actividades prácticas se dividen entre la realización de actividades individuales y trabajos grupales de laboratorio, según la organización que sigue.

Práctica 1: realización de ejercicios de implementación de sistemas embebidos bajo Linux

Práctica 2: realización de ejercicios de implementación de sistemas embebidos bajo FreeRTOS

Práctica 3: realización de aplicaciones bajo Android

Práctica 4: constituida por ejercicios de implementación de soluciones para IoT.

Trabajo practico 1: configuración e implantación de Linux

Trabajo práctico 2: configuración e implantación de FreeRTOS

Trabajo practico 3: implementación de una solución para IoT

Evaluación:

El proceso de evaluación de esta asignatura incluye la realización de las siguientes actividades dentro de la carga horaria y horarios asignados:

- Evaluación parcial 1: incluye los contenidos temáticos de los puntos 1 y 2.1
- Evaluación parcial 2: incluye los contenidos temáticos de los puntos 2.2 y 2.3
- Evaluación parcial 3: integración y contenidos temáticos de la unidad 3
- Actividad de recuperación: es una instancia de recuperación para los alumnos que no hayan aprobado algunas de las evaluaciones parciales.
- Ejecución satisfactoria de los trabajos prácticos

Para obtener la promoción en la asignatura los alumnos deberán aprobar con nota mayor o igual a 6 (seis) estas evaluaciones y prácticos durante el cursado de la materia.

Distribución de la carga horaria:

Presenciales	Horas	
Teóricas	30	
Prácticas:	Experimental de Laboratorio	9
	Experimental de Campo	0
	Resolución de Problemas y Ejercicios	15
	Problemas Abiertos de Ingeniería	9
	Actividades de Proyecto y Diseño	24
	Práctica Profesional Supervisada	0
Evaluaciones	9	
Total	96	

Dedicadas por el alumno fuera de clase:	Horas
Preparación Teórica	8
Preparación Práctica	26
Elaboración y redacción de informes, trabajos, presentaciones, etc.	6
Total	40

Cronograma de actividades

SEMANA	UNIDAD	ACTIVIDAD	TEMA
01	1	Exposición	Los de la unidad
02	1 y 2.1	Ejercitación, Exposición	Los de las unidades
03	2.1	Exposición, ejercitación e implementaciones	Los de las unidad
04	2.1	Ejercitación e implementaciones	Los de las unidad
05	2.1	Implementaciones	Los de las unidad
06	2.1 y 2.2	Presentación de Trabajo Práctico. Exposición	Trabajo practico 1. Resto: los de la unidad 2.2
07	1, 2.1 y 2.2	Evaluación Parcial 1. Exposición, Ejercitación	Evaluación parcial unidades 1 y 2.1. Resto: los de la unidad 2.2
08	2.2	Exposición, ejercitación e implementaciones	Los de las unidad
09	2.2	Implementaciones	Los de las unidad
10	2.2 y 2.3	Presentación de Trabajo Práctico. Exposición	Trabajo practico 2. Resto: los de la unidad 2.3
11	2.3	Exposición, ejercitación y desarrollos	Los de las unidad
12	2.3	Exposición, ejercitación y desarrollos	Los de las unidad
13	2.2, 2.3 y 3	Evaluación parcial 2. Exposición	Evaluación parcial unidades 2.2 y 2.3. Resto: los de la unidad 3
14	3	Exposición, implementaciones	Los de las unidad
15	3	Implementaciones. Presentación de Trabajo Práctico.	Trabajo practico 3. Resto: los de la unidad 3
16	1, 2, 3	Evaluación parcial 3. Recuperación	Los de todas la unidades

Bibliografía básica

Título	Autor(es)	Editorial	Año de edición	Ejemplares disponibles
Sistemas Operativos Modernos. ISBN: 9702603153	Andrew S. Tanenbaum –	Pearson	-2004	
Linux Device Drivers (3ªEd, online)				http://www.free-electrons.com/doc/books/ldd3.pdf
Embedded Linux Primer (2da edición) ISBN 978-0-13-701783-6	Christopher Hallinan -	Prentice Hall	- 2010	
Using The FreeRTOS Real Time Kernel ISBN 9781446171080	Richard Barry -	Real Time Engineers Ltd.	- 2011	
Sistemas de Tiempo Real y Lenguajes de Programación (3ª Edición). ISBN: 8478290583	Alan BURNS y Andy WELLINGS	ADDISON WESLEY	2005	
Desarrollo con Microcontroladores ARM ISBN 9789872872007	Sergio Caprile -		2012	
Networking Protocols and Standards for Internet of Things				https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot.pdf
Android Programming Succinctly				http://www.onlineprogrammingbooks.com/android-programming-succinctly/

Bibliografía complementaria

Título	Autor(es)	Editorial	Año de edición	Ejemplares disponibles
Fundamentos de Sistemas Operativos (7a edición) ISBN: 8448146417	Galvin, Silverschatz y Gagne	McGraw-Hill Interamericana	2006	
Ingeniería de Software - ISBN:9786073206037	Ian Sommerville	PEARSON EDUCACION	2011	
Exploring Beaglebone. ISBN 978-1-118-93512-5	Derek Molloy	Wile	2014	
Android Developers				https://developer.android.com/training/index.html?hl=es-419
Node-RED				https://nodered.org/docs/
Node.js Tutorial				https://www.w3schools.com/nodejs/
An Application Builder for the Internet of Things				https://ubidots.com/
Kolban's book on ESP32				https://leanpub.com/kolban-ESP32
Particle Docs				https://docs.particle.io/guide/getting-started/intro/photon/

Recursos web y otros recursos

- Apuntes de cátedra disponibles en www.dsi.fceia.unr.edu.ar.
- Kits de desarrollo para ejercitación sobre Linux embebido
- Kits de desarrollo para ejercitación sobre FreeRTOS
- Kits de desarrollo para ejercitación de Android
- Kits de desarrollo para implementaciones IoT