



Guía docente

230707 - AES - Sistemas Incrustados para Automoción

Última modificación: 29/04/2020

Unidad responsable: Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona
Unidad que imparte: 744 - ENTEL - Departamento de Ingeniería Telemática.

Titulación: MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN (Plan 2013). (Asignatura optativa).
MÁSTER UNIVERSITARIO EN INGENIERÍA ELECTRÓNICA (Plan 2013). (Asignatura optativa).
MÁSTER UNIVERSITARIO EN TECNOLOGÍAS AVANZADAS DE TELECOMUNICACIÓN (Plan 2019). (Asignatura optativa).

Curso: 2020 **Créditos ECTS:** 5.0 **Idiomas:** Inglés

PROFESORADO

Profesorado responsable: de la Cruz Llopis, Luis J.

Otros: Moreno Arostegui, J. Manuel
Madrenas Boadas, Jordi
Franch Gutiérrez, Xavier
Abella Ferrer, Jaume

METODOLOGÍAS DOCENTES

Clases expositivas
Clases de aplicación
Clases laboratorio
Trabajo individual (no presencial)
Trabajo en grupo (no presencial)
Prácticas de laboratorio
Pruebas de respuesta corta (Control)
Pruebas de respuesta corta (Test)
Pruebas de respuesta larga (Examen Final)

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

En la actualidad es evidente una evolución en el sector de la automoción hacia vehículos que hacen un uso exhaustivo de tecnologías electrónicas y de las comunicaciones. Vehículos con todo tipo de sensores y actuadores (temperatura, proximidad, cámaras, asistencia en conducción, aparcamiento, ...), tecnologías de comunicaciones de largo y corto alcance (4G, 5G, RFID/NFC, WiFi,...) y sus servicios asociados (acceso a Internet, infotainment, teleoperación, ...) hacen cada día más cómoda, fiable y segura la conducción. Para dotar a los vehículos con todas estas posibilidades, son numerosas las compañías, tanto nacionales como internacionales, que centran su esfuerzo en la producción de sistemas que permitan a las fábricas de vehículos estar a la vanguardia en la oferta que hacen a sus posibles clientes. Dichas compañías constituyen en la actualidad un atractivo destino en el que desempeñar su labor para ingenieros de telecomunicación e informáticos con especial interés en el desarrollo y programación de sistemas embedded.

Esta asignatura surge de la necesidad de ofrecer una primera especialización para los ingenieros que deseen orientar su carrera profesional hacia la contribución en el desarrollo de dichos sistemas electrónicos de comunicaciones. Ha sido diseñada en colaboración entre grupos de investigación de distintos departamentos de la UPC y grupos de trabajo de compañías punteras en el sector de la electrónica para la automoción. Entre sus contenidos se incluyen los aspectos relacionados con desarrollo software para este tipo de sistemas, sistemas operativos en tiempo real, buses de comunicaciones y arquitecturas de referencia. Además, se estudian los procesos relacionados con la evaluación, verificación, validación y seguridad funcional del software desarrollado. Como resultado, se ofrece una interesante formación previa que permita a los titulados entrar con garantías de éxito en esta apasionante industria.



HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo grande	26,0	20.80
Horas grupo pequeño	13,0	10.40
Horas aprendizaje autónomo	86,0	68.80

Dedicación total: 125 h

CONTENIDOS

1. Introducción

Descripción:

- 1. Introducción.
 - 1.1. Apertura.
 - 1.2. El modelo V.
 - 1.3. Conceptos emergentes.
 - 1.3.1. Software incrustado y telemática.
 - 1.3.2. Seguridad funcional, arquitectura de software y testing.
 - 1.3.3. Evaluación de procesos.
 - 1.4. Estructura del curso. Información práctica.
-
- 1.3.2. Seguridad funcional, arquitecturas de software y testing.
 - 1.3.3. Evaluación de procesos
 - 1.4. Estructura del curso. Información práctica.

Dedicación: 0h 45m

Grupo grande/Teoría: 0h 45m



2. Software incrustado.

Descripción:

- 2.1. Principios de diseño de software incrustado.
 - 2.1.1. Prácticas de diseño y codificación de algoritmos.
 - 2.1.2. Técnicas avanzadas de E/S.
 - 2.1.2.1. E/S gestionada por DMA.
 - 2.1.2.2. E/S manejada por interrupción.
 - 2.1.3. Reglas de diseño y buenas prácticas MISRA-C.
- 2.2. RTOS.
 - 2.2.1. Introducción.
 - 2.2.2. Núcleo.
 - 2.2.3. Tareas, multitarea y multihilo.
 - 2.2.4. Programador.
 - 2.2.5. Comunicación entre procesos.
- 2.3. El protocolo de comunicación CAN.
 - 2.3.1. Introducción.
 - 2.3.2. Topología del bus.
 - 2.3.3. Mensajes CAN.
 - 2.3.4. Capa física.
 - 2.3.5. Bit Timing.
 - 2.3.6. Manejo de errores.
 - 2.3.7. Versiones de protocolo (2.0A, 2.0B, Open).
- 2.4. Sesiones de laboratorio.
 - 2.4.1. Introducción al laboratorio y las herramientas de diseño.
 - 2.4.2. Diseño de una aplicación de software independiente.
 - 2.4.3. Diseño de una aplicación de software basada en un RTOS.

Dedicación: 8h 30m

Grupo grande/Teoría: 8h 30m

3. Autosar.

Descripción:

- 3.1. Arquitecturas de referencia y su papel en los sistemas de software.
- 3.2. AUTOSAR: una arquitectura de referencia de software para la industria automotriz.
 - 3.2.1. Metas.
 - 3.2.2. Cronología. Releases.
 - 3.2.3. Partnership.
- 3.3. Aspectos básicos.
 - 3.3.1. Protocolos de comunicación en autopmoción: CAN, LIN, Flexray.
 - 3.3.2. Diagnosticos. UDS ISO 14229. Adaptación de UDS a CAN.
- 3.4. Elementos constitutivos de AUTOSAR.
 - 3.4.1. Las capas.
 - 3.4.1.1. Software básico. Dependencias.
 - 3.4.1.2. Runtime Environment y su configuración.
 - 3.4.1.3. Capa de aplicación.
 - 3.4.2. El bus funcional virtual.
 - 3.4.3. Interfaces.
- 3.5. Metodología AUTOSAR.
 - 3.5.1. Arquitectura.
 - 3.5.2. Procesos de desarrollo.
 - 3.5.3. Producción de software. Generación de código (basado en modelo).
 - 3.5.4. Intercambio de datos.
 - 3.5.5. Soporte de herramientas.
- 3.6. Conclusiones.

Dedicación: 6h 15m

Grupo grande/Teoría: 6h 15m

4. Telemática.

Descripción:

- 4.1. Comunicaciones V2X.
 - 4.1.1. Sistemas inteligentes de transporte (ITS).
 - 4.1.1.1. Arquitectura ETSI.
 - 4.1.1.1.1. Capa de aplicación.
 - 4.1.1.1.2. Capa de facilities.
 - 4.1.1.1.2.1. Servicios básicos y mensajes.
 - 4.1.1.1.3. Capa de red y transporte.
 - 4.1.1.1.3.1. Protocolo de transporte básico (BTP).
 - 4.1.1.1.3.2. GeoNetworking.
 - 4.1.1.1.4. Capa de acceso.
 - 4.1.1.1.4.1. IEEE 802.11p / ITS-G5.
 - 4.1.1.1.4.2. Cellular V2X.
 - 4.1.2. Linux incrustado en automoción.
 - 4.2.1. Arquitectura del kernel de Linux: puntos esenciales para adaptar el kernel a una plataforma incrustada personalizada.
 - 4.2.2. Técnicas de dimensionado para el kernel.
 - 4.2.3. Yocto Distribution: Entorno de desarrollo cruzado para proyectos integrados.
 - 4.2.4. Bootloaders. U-Boot y Android Fastboot.
 - 4.2.5. Almacenamiento Flash y sistemas de archivos.
 - 4.2.6. Desarrollo y depuración de aplicaciones sobre Linux para sistemas incrustados.
- 4.3. Sesiones de laboratorio.
 - 4.3.1. Desarrollo de una aplicación Linux para iniciar e interactuar con una TCU basada en módem Qualcomm.

Dedicación: 7h 45m

Grupo grande/Teoría: 7h 45m

5. Verificación y validación.

Descripción:

- 5.1. Introducción.
 - 5.1.1. Definición e importancia de Software Quality Assurance & Testing.
 - 5.1.2. Gestión de riesgos.
 - 5.1.3. Pruebas en Agile y DevOps.
 - 5.1.4. Software testing economics.
- 5.2. Niveles de prueba (pruebas unitarias, pruebas del sistema, pruebas de integración, ...).
 - 5.2.1. Formas de probar el software.
 - 5.2.2. Los siete principios de prueba.
 - 5.2.3. Niveles y responsabilidades de prueba de software.
 - 5.2.4. Tipos de prueba de software.
- 5.3. Métodos de prueba (black box, white box, grey box, ...).
 - 5.3.1. Modelo V y métodos de prueba.
 - 5.3.2. El ciclo de vida de prueba.
 - 5.3.3. Gestión de pruebas.
 - 5.3.4. Gestión de defectos.
 - 5.3.5. Diseño de casos de prueba.
 - 5.3.6. Practicando el ciclo de vida de prueba (diseño de caso de prueba, ejecución de caso de prueba e informe de defectos).
- 5.4. Automatización de pruebas.
 - 5.4.1. Introducción a la automatización de pruebas.
 - 5.4.2. Hacia un marco de automatización de pruebas de aceptación.
 - 5.4.3. Grabación vs automatización en capas.
 - 5.4.4. Conceptos básicos para la automatización de JUnit + Selenium.
 - 5.4.5. Pruebas automatizadas para APIs.
 - 5.4.6. Práctica de automatización de casos de prueba (el desafío de 10 niveles).
 - 5.4.7. Pruebas automatizadas en dispositivos móviles (demostración).
- 5.5. Desarrollo basado en pruebas.
 - 5.5.1. El concepto de desarrollo basado en pruebas.
 - 5.5.2. Test-First & Acceptance Test-Driven Development.
 - 5.5.3. Practicando TDD a través de un pequeño ejemplo.
- 5.6. Conclusiones.
 - 5.6.1. Conclusiones.

Dedicación: 6h 15m

Grupo grande/Teoría: 6h 15m

6. Seguridad funcional.

Descripción:

- 6.1. Introducción.
 - 6.1.1. ¿Qué es la seguridad funcional?
 - 6.1.2. Seguridad funcional y seguridad del producto / ciberseguridad.
 - 6.1.3. Estándares de seguridad funcional y definiciones.
 - 6.1.4. Visión general de ISO26262.
- 6.2. Conceptos de seguridad.
 - 6.2.1. Análisis de peligros y riesgos y determinación de ASIL.
 - 6.2.2. Arquitecturas y ejemplos a nivel de sistema.
- 6.3. Seguridad del software.
 - 6.3.1. Requerimientos de seguridad.
 - 6.3.2. Arquitecturas de software para la seguridad funcional.
 - 6.3.3. Patrones en arquitecturas software, E-Gas concept.
 - 6.3.4. Conceptos sobre "Freedom from Interference".
 - 6.3.5. Análisis de seguridad a nivel software.
 - 6.3.6. Autosar y seguridad funcional.
 - 6.3.7. Visión general del proceso de seguridad del software.

Dedicación: 5h 30m

Grupo grande/Teoría: 5h 30m

7. Metodología SPICE.

Descripción:

- 7.1. Introducción.
- 7.2. Modelos de madurez de proceso. CMM. ESPECIA.
- 7.3. SPICE en automoción.
 - 7.3.1. Grupos de procesos.
 - 7.3.2. Work Products.
 - 7.3.3. Niveles de madurez.
- 7.4. Conclusiones.

Dedicación: 4h

Grupo grande/Teoría: 4h

SISTEMA DE CALIFICACIÓN

La evaluación se realiza a través de la realización de trabajos individuales, correspondientes a cada uno de los diferentes módulos que forman la asignatura.



BIBLIOGRAFÍA

Básica:

- ETSI EN. EN 302 636-3-V.1.1.2 Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network Architecture [en línea]. 2014 [Consulta: 16/05/2018]. Disponible a: <http://www.etsi.org/standards>.
- AUTOSAR [en línea]. Disponible a: www.autosar.org.
- SPICE. Automotive SPICE® Process Reference and Assessment Model [en línea]. RELEASE 3.1. 2017 [Consulta: 16/05/2018]. Disponible a: <http://www.automotivespice.com/download/>.
- Skiena, Steven S. The Algorithm design manual. 2nd ed. London: Springer, 2012. ISBN 9781848000698.
- Motor Industry Software Reliability Association. Guidelines for the use of the C Language in critical systems [en línea]. Misra, 2012 [Consulta: 16/05/2018]. Disponible a: <https://cds.cern.ch/record/2017046>. ISBN 978-1-906400-11-8.
- International Organization for Standardization. ISO/DIS 26262. Road Vehicles - Functional Safety [en línea]. 2009 [Consulta: 16/05/2018]. Disponible a: <https://www.iso.org/obp/ui/#iso:std:iso:26262:-2:dis:ed-2:v1:en>.
- Koomen, Tim; Van der Aalst, Leo; Broekman, Bart; Vroon, Mitchiel. TMap Next, for result-driven testing. Tutein Nolthenius, Uitgeverij, 2007. ISBN 9789072194800.