



Grado en Ingeniería Electrónica Industrial, Doble Grado en Ingeniería Electrónica Industrial e Ingeniería Mecánica

DATOS DE LA ASIGNATURA

Nombre:

Informática Industrial II

Denominación en inglés:

Industrial Applications of Computing Science II

Código:

606610220, 609017228

Carácter:

Obligatorio

Horas:

| | Totales | Presenciales | No presenciales |
|--------------------------|---------|--------------|-----------------|
| Trabajo estimado: | 150 | 60 | 90 |

Créditos:

| Grupos reducidos | | | | |
|------------------|---------------|-------------|--------------------|---------------------|
| Grupos grandes | Aula estándar | Laboratorio | Prácticas de campo | Aula de informática |
| 4.14 | 0 | 1.86 | 0 | 0 |

Departamentos:

Áreas de Conocimiento:

| | |
|--|-------------------------------------|
| Ingeniería Electrónica, Sistemas Informáticos y Automática | Ingeniería de Sistemas y Automática |
| Ingeniería Electrónica, Sistemas Informáticos y Automática | Tecnología Electrónica |

Curso:

3º - Tercero

Cuatrimestre:

Segundo cuatrimestre

DATOS DE LOS PROFESORES

Nombre:

*Sánchez Raya, Manuel

E-Mail:

mrraya@uhu.es

Teléfono:

959217661

Despacho:

8 de Torreumbria

*Profesor coordinador de la asignatura

Consultar los horarios de la asignatura

DATOS ESPECÍFICOS DE LA ASIGNATURA

1. Descripción de contenidos

1.1. Breve descripción (en castellano):

Programación avanzada de periféricos conectados a un microcontrolador. Introducción a los Sistemas en Tiempo Real. Sistemas Industriales basados en Unix. Técnicas de programación. Programación de aplicaciones TCP-IP.

1.2. Breve descripción (en inglés):

Industrial computer systems. Open Systems. Real-Time Systems. Advanced programming techniques. Industrial communications applications. TCP/IP Application Programming.

2. Situación de la asignatura

2.1. Contexto dentro de la titulación:

Esta asignatura ubicada en el segundo cuatrimestre de tercer curso de la titulación. Esta asignatura supone una continuación de Informática Industrial I, en la que se apoya. La asignatura aborda temas avanzados de programación de sistemas empotrados y de sistemas abiertos en entornos industriales.

2.2. Recomendaciones:

Resulta aconsejable disponer de conocimientos adquiridos en asignaturas cursadas anteriormente. Entre estas asignaturas podemos citar las siguientes.

- Fundamentos de Informática (Primer curso, 2º C).
- Sistemas Digitales I (Segundo curso, 1º C).
- Sistemas Digitales II (Segundo curso, 2º C).
- Informática Industrial I (Tercer curso, 1º C).

Aunque puede ser de utilidad, la materia estudiada en esta asignatura no se considera imprescindible para el estudio de ninguna asignatura posterior de la titulación.

3. Objetivos (Expresados como resultados del aprendizaje):

Los objetivos a cubrir por la asignatura pueden resumirse en los siguientes:

- Conocimiento aplicado de informática industrial y comunicaciones.
- Conocer que se entiende por sistema empotrado, cuáles son sus principales características y sus áreas de aplicación.
- Conocer las diferentes herramientas existentes en el mercado para el desarrollo de sistemas empotrados y cuáles son las prestaciones de cada una.
- Proporcionar los conceptos fundamentales, tanto desde el punto de vista del Hardware como del Software, para la implementación de sistemas empotrados.
- Conocer los elementos más comunes que proporcionan señales de entrada a los sistemas empotrados y la forma de conectarlos a éstos.
- Conocer los elementos de salida más usuales sobre los que los sistemas empotrados deben actuar y la forma de controlarlos.
- Conseguir un cierto nivel de dominio de la arquitectura, programación y herramientas de desarrollo de una familia de microcontroladores en particular (AVR), con objeto de que el alumno pueda asentar en las clases de prácticas los conocimientos adquiridos en las clases de teoría y sea capaz de diseñar y poner a punto sistemas de complejidad media-baja.
- Conocer los elementos constitutivos de un sistema empotrado basado en sistema Linux.
- Ser capaz de poner en marcha un sistema empotrado basado en software libre y realizar su programación básica.

4. Competencias a adquirir por los estudiantes

4.1. Competencias específicas:

4.2. Competencias básicas, generales o transversales:

- **CB2:** Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de, su área de estudio
- **CB4:** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado
- **G04:** Capacidad de aplicar los conocimientos en la práctica
- **G05:** Capacidad para trabajar en equipo
- **G07:** Capacidad de análisis y síntesis
- **T01:** Uso y dominio de una segunda lengua.

5. Actividades Formativas y Metodologías Docentes

5.1. Actividades formativas:

- Sesiones de Teoría sobre los contenidos del Programa.
- Sesiones de Resolución de Problemas.
- Sesiones Prácticas en Laboratorios Especializados o en Aulas de Informática.
- Actividades Académicamente Dirigidas por el Profesorado: seminarios, conferencias, desarrollo de trabajos, debates, tutorías colectivas, actividades de evaluación y autoevaluación.

5.2. Metodologías docentes:

- Clase Magistral Participativa.
- Desarrollo de Prácticas en Laboratorios Especializados o Aulas de Informática en grupos reducidos.
- Resolución de Problemas y Ejercicios Prácticos.
- Evaluaciones y Exámenes.

5.3. Desarrollo y justificación:

Parte Teórica

En las 38,4 horas destinadas a clases de teoría se impartirán a la totalidad del grupo clases magistrales, donde se expondrán los conceptos teóricos fundamentales que el alumno debe adquirir y se intercalarán los ejemplos y problemas que se estime necesario.

Para impartir los conceptos teóricos el profesor usará transparencias principalmente, de las cuales se aportará a los alumnos una copia con la suficiente antelación para que puedan tomar notas sobre las mismas. Por el contrario, para la resolución de ejemplos y problemas el profesor hará uso de la pizarra, con objeto de que los alumnos puedan seguir su desarrollo con mayor facilidad.

El esquema de exposición a seguir en este tipo de clases será el siguiente:

- Presentación del tema, situándolo en su contexto y relacionándolo con los restantes temas de la asignatura.
- Desarrollo de los diferentes apartados que definen dicho tema, motivando la comprensión del alumno con el uso de cuestiones cortas y ejemplos.
- Íntesis de lo expuesto, así como conclusiones y formulación de críticas.
- Relación de la bibliografía relativa a lo expuesto, así como de aquella que puede ser usada por los alumnos que estén interesados en profundizar en el tema.

Prácticas de laboratorio

Las 18,6 horas de clases prácticas se desarrollarán en el laboratorio, donde cada alumno dispondrá tanto de un ordenador, como de las herramientas de desarrollo y tarjetas de aplicación necesarias.

Los grupos de prácticas tendrán un máximo de 20 alumnos, que podrán trabajar en parejas o de forma individual.

Para estas sesiones se plantearán un conjunto de cuestiones sobre diseño de sistemas empotrados que ayuden a asimilar los conceptos estudiados en las clases de teoría. Las cuestiones correspondientes a cada una de las prácticas se facilitarán a los alumnos con la suficiente antelación, con objeto de que puedan trabajar en su resolución antes de asistir a las clases. Los alumnos deberán presentar al profesor el correcto funcionamiento de la solución obtenida para las diferentes prácticas, así como entregar una memoria escrita para cada una de ellas y responder adecuadamente a las preguntas que éste les formule acerca de las mismas.

Para la realización de las prácticas de la asignatura, donde se aplicarán los conocimientos adquiridos en las clases de teoría, se hará uso de microcontroladores comerciales pertenecientes a la familia AVR de Atmel: ATMEGA328P y el empleo de sistemas Linux empotrados con microcontroladores ARM: Raspberry Pi. Con objeto de que los alumnos adquieran los conocimientos básicos para el empleo de estos microcontroladores, el profesor impartirá dos grupos de seminarios a lo largo del cuatrimestre.

Un grupo de seminarios irá encaminado a estudiar la arquitectura y los recursos principales de dichos microcontroladores, y estará compuesto al menos por los 5 seminarios siguientes, todos ellos de una hora de duración:

- El puerto serie del AVR.
- El convertidor A/D y memoria EEPROM.
- Programación de la Shell
- Programación de procesos
- Programación TCP/IP
- Programación cliente/servidor

El grado de asimilación por parte de los alumnos de los contenidos expuestos en estos seminarios será evaluado mediante un conjunto de cuestiones incluidas en el test del examen de teoría. El enfoque en las sesiones de práctica estará encaminado al empleo de las librerías y técnicas de programación fundamentalmente para conseguir los objetivos marcados. No se abordarán los detalles no relacionados con estos objetivos.

El otro grupo de seminarios irá encaminado a iniciar a los alumnos en el uso de las herramientas de desarrollo que se deberán emplear para la realización de las prácticas y constará de los dos siguientes:

- Introducción al sistema en tiempo real FreeRTOS: manejo de la librería de programación.
- Introducción al desarrollo de sistemas empotrados en Linux (1 hora): manejo de ficheros.
- Introducción la programación en Linux (1 hora): gnu y make.

Debido a la gran importancia de los conceptos proporcionados en estas sesiones para la posterior realización de las prácticas, la asistencia a las mismas será tenida en cuenta.

En el laboratorio de prácticas, los alumnos dispondrán de 30 minutos con el profesor para plantear sus dudas con respecto a la forma en que deben realizar las memorias de las prácticas.

6. Temario desarrollado:

Tema 0: Repaso Interrupciones y Temporizadores.

- Interrupciones en la familia AVR.
 - Tipos de interrupciones.
 - Definición de rutinas de servicio de interrupción en C para AVR.
- Temporizadores/contadores internos.
 - Temporizadores y contadores en la familia AVR.

Unidad didáctica I: Diseño y Programación Avanzada de Sistemas Empotrados.

Tema 1: Entrada/salida serie.

- Comunicación serie asíncrona.
 - Introducción.
 - La comunicación serie en los AVR.
 - Formateo y conversión a la salida.
 - Gestión del puerto serie por interrupciones.
- Interfaces Serie entre dispositivos.
 - Bus I2C.
 - Bus SPI.
 - Expansión empleando registros de desplazamiento.

Tema 2: Convertidores A/D y D/A.

- Convertidores A/D.
 - Introducción.
 - Convertidor A/D interno del PIC
 - Convertidores A/D externos - MCP3425.
- Conversión D/A.
 - Conversión D/A mediante PWM.
 - Conversión D/A externa. AD5722.
 - Potenciómetros digitales.
 - Multiplexores analógicos.

Tema 3: Sistemas en Tiempo Real.

- Introducción a la multitarea y los sistemas de Tiempo Real.
 - Las ideas básicas. Como implementar la multitarea y el tiempo real.
 - Multitarea - tareas, prioridades y plazos.
 - ¿Qué es "tiempo real"?
- Como conseguir multitarea mediante simple programación secuencial.
 - Evaluación del "bucle principal".
 - Tareas disparadas por tiempo y disparadas por eventos.
 - Empleo de las interrupciones para la fijación de prioridades, la estructura primer plano / trasfondo.
 - Introducción de un "tick de reloj" para sincronizar la actividad del programa.
 - Un "sistema operativo" de propósito general.
 - Los límites de la programación secuencial para hacer multitarea.
- El sistema operativo de tiempo real (RTOS).
- La planificación y el planificador.
 - Planificación cíclica.
 - Planificación Round Robin y cambio de contexto.
 - Estados de las Tareas.
 - Planificación pre-emptiva con prioridades.
 - Planificación Cooperativa.
 - El papel de las interrupciones en la planificación.
- El desarrollo de las tareas.
 - La definición de las tareas.
 - Escritura de las tareas y establecimiento de prioridades.
- La protección de los Datos y recursos - el semáforo.
- Introducción a FreeRTOS.
- Ejemplos en FreeRTOS.

Unidad didáctica II: Diseño y Programación de sistemas empotrados en Linux.

Tema 4: Introducción a los sistemas empotrados en Linux.

- Introducción.
- Descripción del sistema de desarrollo.
- Manejo básico del shell de Unix.
 - Introducción.
 - Manejo de directorios.
 - Manejo de archivos.
- Acceso a dispositivos.
- Visualización de ficheros.
- Búsqueda avanzada en ficheros.
 - Caracteres especiales
 - Expresiones regulares.
- Compresión de archivos.
- Redirecciones y tuberías.
 - Redirecciones
 - Tuberías
 - Bifurcaciones.
 - Redirección de E/S estándar.
- Ejecución de programas.
 - Kill, nice y nohup.
 - Time y top
 - sleep
- Comandos para trabajo en red.
 - Protocolos Internet.
 - DNS.
 - Telnet y ftp.

Tema 5: Programación BASH y C.

- Programación de Scripts de Shell.
 - Introducción.
 - Hola Mundo.
 - Variables y operaciones.
 - Sentencias de control de programa.
 - Funciones y librerías
 - Señales
- Compilado de programas en Linux.
- Programación con las librerías estándar de manejo de ficheros.

Tema 6: Programación de procesos y comunicación.

- Introducción.
- Estudio de los procesos.
 - Identificador de proceso.
 - Listado de procesos.
 - Matar un proceso.
- Creación de procesos.
 - system
 - fork y exec.
 - Planificación de procesos.
- Señales.
- Terminación de procesos.
 - Wait.
 - Procesos zombi.
- Comunicación entre procesos.
- Tuberías.
 - Creación de tuberías.
 - Comunicación entre proceso padre e hijo.
 - Redireccionado de E/S estándar.
 - Popen y pclose.
- Sockets.
 - Concepto de socket.
 - Llamadas al sistema.
 - Servidores.
 - Sockets locales.
 - Sockets de dominio internet.
 - Socketpair.

7. Bibliografía

7.1. Bibliografía básica:

- "Make: AVR Programming: Learning to Write Software for Hardware".
Autor: Elliot Williams,
Editorial: Maker Media, Inc
Año: 2014
- Building Embedded Linux Systems.
Autor: Karim Yaghmour, Jon Masters, Gilad Ben-Yossef, Philippe Gerum.
Editorial: O'Reilly Media
Año: 2008.

7.2. Bibliografía complementaria:

- Embedded C
Autor: Michael J. Pont.
Editorial: Adison-Wesley.
Año: 2002.
- Embedded Systems Design.
Autor: Heath, S.
Editorial: Newmes.
Año: 2003.
- Programming embedded systems.
Autor: Barr, M.
Editorial: O'Reilly.
Año: 1998.
- Embedded Linux Primer: A Practical Real-World Approach.
Autor: Christopher Hallinan.
Editorial: Prentice Hall
Año: 2006.
- Debugging Embedded Linux.
Autor: Christopher Hallinan.
Editorial: Prentice Hall
Año: 2007.
- Embedded Linux: Hardware, Software, and Interfacing.
Autor: Craig Hollabaugh.
Editorial: Addison-Wesley Professional
Año: 2002.
- Essential Linux Device Drivers.
Autor: Sreekrishnan Venkateswaran.
Editorial: Prentice Hall
Año: 2008.
- Embedded Linux System Design and Development.
Autor: P. Raghavan, Amol Lad, Sriram Neelakandan.
Editorial: Auerbach Publications
Año: 2005.
- Linux for Embedded and Real-Time Applications (Embedded Technology).
Autor: Doug Abbott.
Editorial: Newnes
Año: 2003.

8. Sistemas y criterios de evaluación.

8.1. Sistemas de evaluación:

- Examen de teoría/problemas
- Defensa de Prácticas
- Seguimiento Individual del Estudiante
- Examen de prácticas

8.2. Criterios de evaluación y calificación:

Criterios Generales

La nota final será la suma de la evaluación de la parte Teórica y de la parte Práctica **siempre que la nota de la parte Teórica sea mayor o igual a 5 puntos sobre 10 y la nota de la parte práctica sea mayor o igual a 5 puntos sobre 10.** También se realizará un seguimiento del aprendizaje de la asignatura mediante la realización de un test de preguntas al final de cada tema impartido que supondrá un punto añadido a la nota.

Por tanto, es necesario aprobar Examen y Prácticas de forma independiente. La nota de la evaluación de Test se sumará al total obtenido del examen y las prácticas solo si se han aprobado Teoría y Prácticas al mismo tiempo. Si se ha aprobado cada parte por separado la nota final se obtendrá según el siguiente criterio:

Nota final: 65% Ex. Escrito + 30% Prácticas + 5% Nota TEST.

Si solo se ha superado la parte Teórica o la parte Práctica, la calificación de la asignatura será de Suspenso. La nota de la parte aprobada se guardará hasta la convocatoria de Septiembre. No se guarda ninguna nota de un curso para otro

Examen Teórico-Práctico escrito

El examen Teórico-Práctico consistirá en una prueba escrita compuesta por un conjunto de preguntas cortas de teoría y dos o tres problemas prácticos. La puntuación de esta prueba tendrá el 70% del total, repartida entre preguntas cortas y problemas prácticos.

Examen: máximo 6.5 puntos (obligatorio).

Evaluación Periódica de Adquisición de conocimientos

Se realizarán test periódicos para autoevaluación del alumno al final de cada tema, realizados de forma individual. La evaluación del test tendrá un valor máximo de 0.5 puntos. Se hará la media de los test realizados. Si se realizan menos de tres test, la nota se contabilizará como cero puntos.

TEST: máximo 0.5 puntos (obligatorio).

Evaluación de prácticas de laboratorio

La calificación de las prácticas de laboratorio supondrá un 30% de la nota final, obtenida mediante cuestiones formuladas a los alumnos de manera individual sobre las prácticas realizadas, y mediante la calificación de las memorias entregadas.

Prácticas: máximo 3 puntos (obligatorio).

En el caso que el alumno no haya podido asistir a las prácticas de laboratorio que son obligatorias y desee presentarse a la convocatoria de Septiembre sin haber superado la parte de las prácticas de la asignatura, con el fin de evaluar al alumno en estos casos, éste podrá realizar las prácticas por su cuenta, siguiendo el material de la asignatura y planteando las dudas que le surjan, mediante el software empleado en el curso.

Deberá entregar una memoria completa de prácticas al menos dos semanas antes de la fecha del examen correspondiente.

Para la evaluación de estas prácticas además de la calidad de la memoria presentada, también se tendrán en cuenta las resupuestas a varias cuestiones por escrito que se plantearán durante el examen de la convocatoria correspondiente.

9. Organización docente semanal orientativa:

| | Semanas | Grupos Grandes | Grupos Reducidos Aula Estándar | Grupos Reducidos Aula de Informática | Grupos Reducidos Laboratorio | Grupos Reducidos prácticas de campo | Pruebas y/o actividades evaluables | Contenido desarrollado |
|-----|---------|----------------|-----------------------------------|---|---------------------------------|--|---------------------------------------|------------------------|
| #1 | 0 | 0 | 0 | 0 | 0 | | | Presentación |
| #2 | 2.4 | 0 | 0 | 0 | 0 | | | Tema 0 |
| #3 | 3 | 0 | 0 | 0.6 | 0 | | | Tema 1 |
| #4 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 1 |
| #5 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 2 |
| #6 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 2 |
| #7 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 3 |
| #8 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 3 |
| #9 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 4 |
| #10 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 4 |
| #11 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 5 |
| #12 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 5 |
| #13 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 6 |
| #14 | 3 | 0 | 0 | 1.5 | 0 | | | Tema 6 |
| #15 | 3 | 0 | 0 | 1.5 | 0 | Entrega de Trabajos/Prácticas | | |
| | 41.4 | 0 | 0 | 18.6 | 0 | | | |