



**Datos Generales de la asignatura**

<b>Nombre de la asignatura:</b>	Sistemas Operativos Embebidos.
<b>Clave de la asignatura:</b>	
<b>SATCA<sup>1</sup>:</b>	(3 - 2 - 5)
<b>Carrera:</b>	Ingeniería Electrónica, Ingeniería Biomédica, Ingeniería Mecatrónica

**2. Presentación**

**Caracterización de la asignatura**

Esta asignatura aporta los conocimientos básicos sobre desarrollo de software, sistemas operativos y redes de comunicaciones necesarios para el desarrollo de sistemas embebidos, tales como multitareas cooperativas y apropiativas, comunicación entre tareas y requerimientos de tiempo real de un sistema. Así mismo, la materia aporta las bases para la aplicación de sistemas operativos de propósito general, tales como línea de comandos y guiones, llamadas al sistema y manejo de archivos, así como el acceso a la infraestructura de comunicaciones.

En el aspecto personal la materia incentiva la solución de problemas utilizando las herramientas disponibles, comunicarse con efectividad en forma oral y escrita en el ámbito profesional y crear una disciplina de auto aprendizaje que le permita buscar soluciones a los problemas.

La asignatura se relaciona con las materias de programación estructurada y visual, diseño digital y microcontroladores de la carrera genérica y con programación móvil de la especialidad de automatización y sistemas embebidos. En esta materia se integran los conocimientos de las materias previas y los nuevos conocimientos en el desarrollo de sistemas más complejos.

Desde el punto de vista profesional, la asignatura proporciona habilidades que son requeridas en sectores como la industria automotriz, aeronáutica, de instrumentos médicos, tecnologías de la información y el sector manufacturero entre otras.

<sup>1</sup> Sistema de Asignación y Transferencia de Créditos Académicos





**Intención didáctica**

En las actividades prácticas sugeridas, es recomendable que el profesor busque sólo guiar a sus alumnos para que ellos aprendan a desarrollar la parte de creatividad que les corresponde y que se necesita para planificar y realizar las metas y objetivos.

**3. Participantes en la actualización, el diseño, consolidación y/o seguimiento curricular del programa**

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Morelia, Morelia Mich., 3-7 Junio 2019.	Miguelangel Fraga Aguilar. David Infante Sánchez Sergio Armando Galván Chavéz Tarsicio Alfaro García Arturo Méndez Patiño	Curso para la definición de las Especialidades de Ingeniería Electrónica e Ingeniería Biomedica.

**4. Competencia(s) a desarrollar**

**Competencia(s) específica(s) de la asignatura**

Desarrollar las competencias siguientes para aprender los conceptos y manejo básico de los sistemas operativos embebidos.

- Comprender las características de los sistemas operativos que permiten el desarrollo de sistemas de software complejo y su aplicación en los sistemas embebidos.
- Comprender las características de los sistemas embebido, así como los requisitos con los que deben cumplir.
- Aprender a manejar los periféricos de los sistemas embebidos dentro del entorno de un sistema operativo.
- Aprender a comunicar los sistemas embebidos con otros sistemas.
- Aplicar las capacidades de los sistemas embebidos como servidores para proporcionar interfaces web a los sistemas embebidos.
- Enfocar los conocimientos adquiridos a aplicaciones industriales, hogar, medicina, etc.

Desarrollar las competencias instrumentales siguientes para aplicarlas en proyectos de innovación multidisciplinarios o en el ámbito laboral.

- Capacidad de organizar ideas y planificar objetivos.
- Capacidad para buscar y analizar información proveniente de fuentes diversas.
- Habilidad para la comunicación oral y escrita en español.
- Habilidad para la comunicación oral y escrita en inglés.





- Habilidad para solucionar problemas y tomar decisiones.
- Aplicar los conocimientos básicos de la carrera.
- Aplicación de tecnologías de la información.
- Integrar nuevas tendencias tecnológicas de conectividad al desarrollo de nuevos productos.
- Tener alto conocimiento de lenguajes de programación para poder integrar soluciones innovadoras relacionando al hardware y software.
- Tener capacidad de aprendizaje continuo de nuevas tecnologías para su aplicación a los desarrollos de nuevos productos.
- Hacer trabajo estructurado y documentado para su compartición con integrantes de los equipos de trabajo con los que colabora.
- Tener conocimiento de sistemas operativos abiertos.
- Tener Conocimiento de arquitecturas de procesadores para seleccionar el más adecuado a problemas de conectividad y poder de procesamiento.

Desarrollar las competencias interpersonales siguientes para mejorar la comunicación y relación para establecer vínculos entre dos o más personas.

- Tener capacidad crítica y autocrítica.
- Aprender a trabajar en equipo.
- Desarrollar habilidades interpersonales.
- Adaptación a trabajar en otros lugares dentro y fuera del país.
- Respeto y entendimiento para adaptarse a diferentes culturales.

Desarrollar las competencias sistémicas siguientes.

- Capacidad de trabajar en equipo.
- Capacidad de liderazgo.
- Tener la capacidad de aplicar los conocimientos en la práctica.
- Desarrollar habilidades de investigación.
- Tener capacidad de aprender.
- Tener la capacidad de generar nuevas ideas (creatividad).
- Desarrollar la habilidad para trabajar en forma autónoma.
- Tener la capacidad para diseñar y gestionar proyectos.
- Tener Iniciativa y espíritu emprendedor.
- Mantener preocupación por la calidad.
- Habilidad para la búsqueda de logro.

### 5. Competencias previas

Conoce y aplica la **Programación Estructurada** para el desarrollo de software.

Analiza y diseña **Sistemas Digitales**.

Comprende el funcionamiento de programas en Lenguaje **C/C++** y **ensamblador**.

Conoce la arquitectura y aplicación de **Microcontroladores**.

### 6. Temario





No.	Temas	Subtemas
1	Metodología para el Desarrollo de software embebido.	1.1 Ciclo de desarrollo de Software 1.2 Sistemas de control de versiones (GIT) 1.3 Metodologías para lograr concurrencia sin un sistema operativo 1.3.1 Maquinas de estado y ciclo pausado 1.3.2 Aquitectura Foreground / Background (interrupciones / Ciclo principal)
2	Sistemas operativos de tiempo real	2.1 Requisitos de tiempo real 2.2 Multitareas cooperativas y apropiativas. 2.3 Administración de tareas. 2.3.1 Creación de tareas 2.3.2 Bloqueo y Suspensión de tareas 2.4 Mecanismos de comunicación y sincronización entre tareas. 2.4.1 Colas 2.4.1 Filas 2.4.2 Sincronización con rutinas de atención de Interupciones
3	Linux en sistemas embebidos	3.1 Línea de comandos 3.2 Guiones de interprete de comandos (shell) 3.3 Manejo de Archivos 3.4 Acceso a periféricos en línea de comandos / C/C++ / Python
4	Servicios de Red	4.1 Servidor WEB 4.2 Servidores de bases de datos 4.3 Internet de las cosas





## 7. Actividades de aprendizaje de los temas

Unidad 1 - Metodología para el Desarrollo de software embebido	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> El alumno conocerá el proceso de desarrollo de software embebido, así como algunas herramientas comúnmente usadas. El alumno identificara los casos en los que es conveniente usar un sistema operativo y conocerá algunas metodologías alternativas al uso de un sistema operativo.</p> <p><b>Genéricas:</b> Capacidad de aplicar los conocimientos en la práctica. Capacidad para generar nuevas ideas. Habilidad para trabajar en forma autónoma. Destreza de comunicarse en forma oral y escrita en idioma inglés. Habilidades básicas de manejo de la computadora. Toma de decisiones.</p>	<p>El alumno usara un servidor publico (GitHub, GitLab, BitBucket) para almacenar el codigo fuente de las practicas que realice durante el semestre y colaborar en linea con los miembros de su equipo. El alumno implementara una practica en que un microcontrolador realice dos o mas funciones de forma concurrente usando la arquitectura frontend / backend o la de maquinas de estado.</p>
Sistemas operativos de tiempo real	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> El alumno conocerá las los tipos de requerimientos de tiempo de un sistema embebido y como diseñar sistemas que los satisfagan El alumno empleara la multitareas para simplificar el diseño de sistemas de software complejos dividiendo el funcionamiento en tareas más simples que trabajan de manera coordinada.</p> <p><b>Genéricas:</b> Capacidad de aplicar los conocimientos en la práctica. Capacidad para generar nuevas ideas. Habilidad para trabajar en forma</p>	<p>El Alumno usara temporizadores e interrupciones para general señales que cumplan con requisitos de tiempo real El alumno implementara una practica en que un microcontrolador realice dos o mas funciones de forma concurrente usando un sistema operativo para crear dos o más tareas. El alumno realizara una practica en que comunique dos o más tareas para sincronizar su funcionamiento. El alumno realizara una practica en que comunique al menos una tarea con una rutina de atención a interrupción para</p>





<p>autónoma. Destreza de comunicarse en forma oral y escrita en idioma inglés. Toma de decisiones.</p>	<p>diferir el procesamiento de la interrupción.</p>
Linux en sistemas embebidos	
Competencias	Actividades de aprendizaje
<p>Específica(s): El alumno aprenderá a utilizar una computadora con Linux a través de la línea de comandos tanto de forma interactiva como por medio de guiones de comandos El alumno conocerá y aplicará los mecanismos de acceso a periféricos de Linux Genéricas: Capacidad de aplicar los conocimientos en la práctica. Capacidad para generar nuevas ideas. Habilidad para trabajar en forma autónoma. Destreza de comunicarse en forma oral y escrita en idioma inglés.</p>	<p>El alumno instalará una distribución de Linux en su computadora ya sea en modo de arranque dual o en una máquina virtual. Alternativamente podrá usar una computadora de una sola tarjeta (<b>SBC</b>) como la RaspberryPi o la Beagle Bone El alumno escribirá un programa en donde lea datos desde un archivo, los procese y almacene el resultado en otro archivo El alumno realizará una práctica en donde utilice un periférico de una computadora de una sola tarjeta (<b>SBC</b>) o un puerto serie en una PC</p>
Servicios de Red	
Competencias	Actividades de aprendizaje
<p>Específica(s): El alumno obtendrá las nociones básicas de redes de telecomunicaciones para usarla en aplicaciones embebidas. El alumno conocerá los fundamentos de las tecnologías web y su uso como interfaces para sistemas embebidos. El alumno aprenderá las características de la Genéricas: Capacidad de aplicar los conocimientos en la práctica. Capacidad para generar nuevas ideas. Habilidad para trabajar en forma autónoma. Destreza de comunicarse en forma oral y</p>	<p>El alumno conocerá los principales protocolos de la pila TCP/IP y su función El alumno estudiará los fundamentos del lenguaje de marcas HTML para crear páginas web El alumno instalará un servidor web (Apache, nginx), así como los módulos para ejecutar un lenguaje de servidor (PHP) El alumno usará una página web como interfaz para un sistema embebido</p>





escrita en idioma inglés.

## 8. Práctica(s)

- 1- Realizar un pequeño programa en c y almacenarlo en un deposito publico. Usar el deposito publico para que los miembros del equipo desarrollen el programa colaborando en linea.
- 2- Implementar en un microcontrolador dos o mas funciones de forma concurrente usando la arquitectura frontend / backend o la de maquinas de estado .
- 3- Implementar en un microcontrolador dos o mas funciones de forma concurrente usando un sistema operativo para crear dos o más tareas.
- 4- Realizar un programa en el microcontrolador donde dos o más tareas se comuniquen por medio de colas o semaforos
- 5- Realizar un programa en el microcontrolador donde una ruitina de atención de interrupciones se e comuniquen por medio de colas o semaforos con una tarea, implementaando el procesamiento diferido de la interrupción.
- 6- Elaborar scripts de Linux que manipulen archivos y directorios u obtengan información de archivos o escriban información a archivos.
- 7- Escribir programas en C en Linux que lean o escriban en archivos
- 8- Escribir programas en C en linux que accedan a perifericos tales como UART, SPI, I2C, GPIO
- 9- Escribir una pagina WEB que presente información obtenida de un sistema embebido o que reciba información y que la transmita a un sistema embebido.
- 10- Desarrollar un sistema embebido que envíe datos a un servidor de bases de datos para su almacenamiento

*(La elaboración y desarrollo de prácticas es un ingrediente indispensable que vincula y fortalece el aprendizaje del saber con el saber hacer, estas prácticas deben propiciar el desarrollo de las competencias genéricas a través de las competencias específicas. Las prácticas permitirán una formación más sólida, y una adecuada integración de las competencias profesionales). Es importante que el estudiante realice al menos una práctica durante el semestre y que corresponda a los contenidos educativos.*





## 9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.







## 10. Evaluación por competencias

Reportes escritos de las prácticas desarrolladas, con base al formato establecido.

- Reporte escrito de las investigaciones documentales solicitadas.
- Reporte escrito de la implementación y prueba de programas.
- Descripción de otras experiencias concretas que podrían realizarse adicionalmente.
- Reportes escritos con las conclusiones de las aplicaciones generadas.
- Exámenes escritos para comprobar el manejo de aspectos teóricos.
- Participación en clase considerando las actividades de trabajo en equipo y la exposición de temas, así como presentación de proyectos individuales o multidisciplinarios.

## 11. Fuentes de información

1. Barry, Richard. Mastering the FreeRTOS Real Time Kernel. Disponible en línea en [www.freertos.org](http://www.freertos.org)
2. Derek Molloy, Exploring Raspberry Pi: Interfacing to the real world with embedded linux, Wiley, 2016.
3. Yifeng Zhu, Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C, E-Man Press LLC, 2014.
4. Samuel Greengard, The Internet of Things, The MIT Press, 2015.
5. Philip Koopman, Better Embedded System Software, Drumndrochit Education, 2010.
6. Christopher Hallinan, Embedded Linux Primer: A Practical Real-World Approach, Prentice Hall, 2010.
7. Karim Yaghmour, Jon Masters, Gilad Ben-Yossef and Philippe Gerum, Building Embedded Linux Systems, O'Reilly Media, 2008.
8. Katalin Popovici, Frédéric Rousseau, Ahmed Jerraya, and Marilyn Wolf, Embedded Software Design and Programming of Multiprocessor System-on-Chip: Simulink and System C Case Studies, Springer, 2010.
9. Marisa Lee, Raspberry Pi 2: Beginner's Step-by-Step Guide to Using Your Raspberry Pi 2, 2015.
10. Yankee Bush Software LLC, How to Build Your Own Real-Time Operating System: Advanced Embedded Kernel Programming in C & Assembly, 2014.
11. Jim Cooling, Real-time Operating Systems, Lindentree Associates, 2013.
12. Simon Monk, Programming the BeagleBone Black: Getting Started with JavaScript and Bone Script, Mc. Graw Hill, 2014.

