



**UNIVERSIDAD AUTÓNOMA DE SINALOA**  
**FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS**  
**CARRERA: INGENIERÍA EN ELECTRÓNICA**



**PROGRAMA DE ESTUDIOS**

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE	SISTEMAS EMBEBIDOS		
<b>Clave:</b>	1461		
<b>Semestre:</b>	IV		
<b>Eje Curricular:</b>	<input type="checkbox"/> Básica <input checked="" type="checkbox"/> Profesionalizante		
<b>Área:</b>	<input type="checkbox"/> Física-Matemática <input type="checkbox"/> Cs. Sociales y Humanidades <input type="checkbox"/> Idiomas <input type="checkbox"/> Básico Profesional <input checked="" type="checkbox"/> Profesional		
<b>Horas y créditos:</b>	<b>Teóricas: 40</b>	<b>Prácticas: 40</b>	<b>Estudio Independiente: 16</b>
	<b>Total de horas: 96</b>		<b>Créditos: 6</b>
<b>Tipo de curso:</b>	<b>Teórico ( )</b>	<b>Teórico-práctico (X)</b>	<b>Práctico ( )</b>
<b>Competencias del perfil de egreso a la que aporta</b>	<p>Diseña sistemas electrónicos analógicos y digitales para resolver problemas del entorno haciendo uso de diversas tecnologías atendiendo las normas y reglamentos para su uso.</p> <p>Identifica fallas en sistemas electrónicos para aplicar un correcto mantenimiento de acuerdo a las normas establecidas.</p> <p>Administra proyectos para la implementación de tecnología electrónica con pertinencia, responsabilidad y calidad.</p> <p>Identifica y evalúa tecnologías electrónicas emergentes para ser consideradas en futuros diseños de forma continua y oportuna.</p> <p>Desarrolla telecomunicaciones, instrumentación y control para resolver problemas del sector industrial de forma eficaz y atendiendo los criterios de calidad necesarios.</p>		
<b>Componentes</b>	<ul style="list-style-type: none"> <li>- Aplicación de TIC en el área de la electrónica para el manejo de la información proveniente de procesos de instrumentación y control.</li> <li>- Determinar si una tecnología es conveniente para un nuevo diseño electrónico.</li> <li>- Capacitarse de forma autodidacta en el uso de nuevas tecnologías de la electrónica.</li> </ul>		
<b>Unidades de aprendizaje relacionadas</b>	Tecnologías de la información y la comunicación, Lenguaje de programación, Programación orientada a objetos.		
<b>Responsables de elaborar y/o actualizar el programa:</b>	<b>Dr. José Vidal Jiménez Ramírez</b> <b>Dr. Carlos Duarte Galván</b>		
<b>Fecha de:</b>	<b>Elaboración: julio de 2017</b>		<b>Actualización: Enero 2021</b>

## 2. PROPÓSITO

El objetivo de la asignatura sistemas embebidos es que los estudiantes apliquen los conocimientos adquiridos en las materias de **programación** y **programación orientada a objetos** para programar una microcomputadora Raspberry Pi y realizar aplicaciones de ingeniería. Se busca que pueden diseñar software que pueda interactuar con hardware a través de los puertos de comunicación y el puerto GPIO de la Raspberry, programar servidores web par almacenar información y controlar de forma remota su hardware.

## 3. SABERES

<b>Teóricos:</b>	<ul style="list-style-type: none"><li>– Entender los conceptos de sistemas embebidos, sus aplicaciones, ventajas y desventajas respecto a las computadoras personales.</li><li>– Conocer la diferencia de sistemas analógicos y digitales.</li><li>– Diseñar sistemas electrónicos analógicos y digitales básicos</li><li>– Diseñar sistemas embebidos tomando en cuenta el problema a resolver y realizar una correcta selección de los componentes del sistema.</li></ul>
<b>Prácticos:</b>	<ul style="list-style-type: none"><li>– Instalar e configurar sistemas operativos en microcomputadoras</li><li>– Manejar los recursos de hardware de una microcomputadora desde Python.</li><li>– Diseñar y construir circuitos analógicos básicos.</li><li>– Diseñar y construir circuitos digitales básicos.</li></ul>
<b>Actitudinales:</b>	<ul style="list-style-type: none"><li>– Valorar el papel de la ciencia en el entendimiento de la naturaleza.</li><li>– Demostrar rigor científico en el planteamiento y solución de problemas.</li><li>– Actitud de trabajo en equipo en la solución de ejercicios.</li><li>– Desarrollará habilidades para trabajar en los laboratorios de manera organizada y estandarizada.</li><li>– Desarrollar habilidades autodidactas.</li><li>– Desarrollar habilidad para la lectura de textos científicos.</li></ul>

## 4. CONTENIDO TEMÁTICO

1. Introducción a sistemas embebidos
  - 1.1. Sistemas embebidos
  - 1.2. Sistemas numéricos (binario, decimal, hexadecimal), conversiones entre sistemas
  - 1.3. Electrónica digital básica (compuertas lógicas, multiplexores, contadores)
  - 1.4. Electrónica analógica básica (diodo rectificador, diodo Zener, transistores BJT)
  - 1.5. Arquitectura de microcontroladores
  - 1.6. Perifericos de microcontroladores, ADC, UART, CCP, PWM, TIMERS, SPI, I2C.
  - 1.7. Programación de microcontroladores en lenguaje C, Arduino, MSP430
  - 1.8. Optoacopladores y etapas de potencia básicas (SCR, MOC, TRIAC)
2. Introduccion a Python
  - 2.1. Numpy y Scipy

- 2.2. Matplotlib
- 2.3. Gráficos en Python y diseño de interfaces gráficas de usuario (GUI), usando librerías como Tkinter
- 3. Aplicaciones de Python con Raspberry pi
  - 3.1. Acceso a los puertos GPIO de la Raspberry pi
    - 3.1.1. Lectura de sensores digitales
    - 3.1.2. Uso de PWM
  - 3.2. Comunicación UART utilizando Raspberry pi (lectura y escritura)
    - 3.2.1. Pyserial
    - 3.2.2. Comunicación entre Raspberri pi y Arduino
    - 3.2.3. Leer datos por el puerto y guardarlos en archivos txt.
- 4. Internet of Things
- 5. Bases de datos (MySQL)
- 6. Servidores WEB (Node.js)

### **5. ACCIONES ESTRATÉGICAS PARA EL APRENDIZAJE**

Durante la clase se abordarán dos temas importantes, por un lado es aprender a programar en lenguaje Python y ejecutar los programas en una microcomputadora Raspberry Pi y aprender a usar librerías como Numpy, Scipy, matplotlib, Pyserial, entre otras. Se recomienda ampliamente realizar programas que utilicen los puertos de la Raspberry, específicamente el puerto de pines digitales (GPIO, General Purpose Input/Output).

La otra parte del curso involucra los sistemas electrónicos que van a interactuar con la Raspberry a través de los GPIO, se sugiere que el estudiante diseñe programas en Python para escribir y leer señales digitales en dicho puerto, también utilizar los protocolos UART, SPI e I2C, etc. Para esto se deben ver temas relacionados con circuitos electrónicos analógicos y digitales que el estudiante tiene que entender y construir.

### **6. EVALUACIÓN DEL APRENDIZAJE**

<b>6.1. Evidencias de aprendizaje</b>	<b>6.2. Criterios de desempeño</b>	<b>6.3. Calificación y acreditación</b>
Exámenes por unidad.  Reportes de investigación.  Exposiciones en clase.  Tareas.  Entrega de prácticas.	Exámenes por unidad: Explicación clara y concreta de los conceptos relacionados con la materia. Solución correcta de problemas de ingeniería propuestos.  Entrega de prácticas: 70% por funcionalidad del circuito electrónico, 30% por el reporte impreso con la descripción del hardware de la práctica.  En lo que respecta a los demás criterios de evaluación, se asignará 30% al formato, 40% al contenido y 30% a las conclusiones que el alumno presente.	70% exámenes.  30% Prácticas y demás trabajos.

## 7. FUENTES DE INFORMACIÓN

### Fuentes de Información Básica:

1. Hablemos Embebido: Guía para diseñar sistemas embebidos. **Marco Antonio Aceves Fernández**. Editorial Asociación Mexicana de Software Embebido, 2019.
2. Las guías para aprender Python se pueden consultar en <https://www.python.org/doc/>
3. Las guías para aprender a usar la Raspberry Pi se pueden consultar en <https://learn.adafruit.com/> y <https://www.raspberrypi.org/>
4. Programación de sistemas embebidos en C. **Gustavo Galeano**, Editorial México 2009.
5. Todo sobre sistemas embebidos: arquitectura, programación y diseño de aplicaciones con el PIC18F. **Sergio Salas Arriarán**, 2017. Editorial Universidad Peruana de Ciencias Aplicadas.
6. Dispositivos electrónicos y teoría de circuitos, **Boylestad y Nashelsky**, 2013. Editorial Bukupedia.
7. Protesus: Simulación de circuitos electrónicos y microcontroladores a través de ejemplos. **Germán Tojeiro Calaza**, Editorial Marcombo 2008.

### Fuentes de Información Complementaria:

1. Microcontroladores PIC, 1ª parte, 4ta edición. **José María Angulo Usategui**. Editorial McGraw-Hill. 2007.
2. Compilador C CCS y simulador Proteus para microcontroladores PIC, **Eduardo García Brejo**. Editorial Marcombo 2009.
3. Proteus VSM, Victor Rossano. Editorial USERSHOP 2013.

## 8. PERFIL DEL PROFESOR:

Experiencia programando lenguajes de programación como ensamblador, Python, C, C++  
Experiencia programando sistemas microcontroladores Arduino, MSP, PIC, ARM, etc.  
Experiencia diseñando circuitos electrónicos analógicos y digitales  
Habilidades para establecer analogías entre sistemas.  
Habilidades didácticas de enseñanza y evaluación del aprendizaje.