UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

COORDINACIÓN GENERAL DE FORMACIÓN BÁSICA
COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA
PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

1. **Unidad Académica**: Facultad de Ingeniería, Arquitectura y Diseño, Ensenada; Facultad de Ciencias Químicas e Ingeniería, Tijuana y Facultad de Ingeniería, Mexicali.

2. **Programa Educativo:** Ingeniero en Electrónica

3. **Plan de Estudios:** 2020-1

4. **Nombre de la Unidad de Aprendizaje:** Sistemas Embebidos

5. **Clave**: 36170

6. **HC:** 01 **HL:** 04 **HT:** 00 **HPC:** 00 **HCL:** 00 **HE:** 01 **CR:** 06

7. **Etapa de Formación a la que Pertenece**: Terminal

8. **Carácter de la Unidad de Aprendizaje**: Obligatoria

9. **Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
REGISTRADO
27 MAR 2019
REGISTRADO
COORDINACIÓN GENERAL DE FORMACIÓN BÁSICA

Equipo de diseño de PUA	**Firma**
Everardo Inzunza González	
Jorge Edson Loya Hernández	
Guillermo Galaviz Yáñez	
Marco Antonio Pinto Ramos	

Fecha: 19 de febrero de 2019

Vo.Bo. de Subdirectores de Unidades Académicas	**Firma**
Humberto Cervantes de Ávila	
Rocío Alejandra Chávez Santoscoy	
Alejandro Mungaray Moctezuma	

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

La finalidad de esta unidad de aprendizaje es brindar la bases y herramientas para el desarrollo e implementación de sistemas embebidos de alto rendimiento computacional que permitirán al estudiante integrar tareas de monitoreo y control remoto vía internet en las diferentes aplicaciones de acuerdo con las demandas y necesidades del contexto laboral.

Se imparte en la etapa terminal con carácter obligatorio y corresponde al área de conocimiento de Diseño en Ingeniería.

III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE

Diseñar e implementar sistemas embebidos de alto rendimiento computacional, mediante el uso de herramientas y técnicas de programación de sistemas comerciales que operan en tiempo real, para la solución de problemas de ingeniería electrónica, de forma analítica, eficiente, ordenada y con respeto al medio ambiente.

IV. EVIDENCIA(S) DE DESEMPEÑO

Diseña e implementa un sistema embebido de alto rendimiento computacional, utilizando una plataforma de tiempo real con arquitectura de 32 o 64 bits, para emitir una solución de un problema real de instrumentación o control remoto vía internet, así como el uso de periféricos, tales como teclados, pantallas LCD/Táctil, cámaras digitales, sensores digitales, sensores inteligentes, periféricos en general y tecnologías modernas de comunicaciones aplicadas a IoT.

V. DESARROLLO POR UNIDADES

UNIDAD I. Sistemas embebidos

Competencia:
Analizar los componentes de sistemas embebidos, a través del estudio de sus características y arquitecturas, para reconocer su aplicación dentro de un sistema electrónico, con ahínco.

Contenido: **Duración:** 3 horas

1.1. Conceptos básicos de sistemas embebidos
1.2. Componentes de un sistema embebido
1.3. Características de los sistemas embebidos
1.4. Lenguajes de programación para sistemas embebidos
1.5. Sistemas embebidos comerciales
1.6. Evolución de los microprocesadores ARM
1.7. Arquitecturas System on Chip
1.8. Familias de microprocesadores y microcontroladores ARM
    1.8.1. Cortex M
    1.8.2. Cortex R
    1.8.3. Cortex A
    1.8.4. Selección del microcontrolador/microprocesador ARM
1.9. Aplicaciones de sistemas embebidos

UNIDAD II. Sistemas operativos embebidos

Competencia:
Utilizar un sistema operativo embebido, para la configuración del sistema, gestión de procesos y hardware, a través de comandos básicos de forma local y remota, de manera eficiente y responsable.

Contenido: **Duración:** 3 horas

2.1. Tipos de sistemas operativos comerciales para sistemas embebidos
2.2. Principales comandos del sistema operativo embebido
- 2.2.1. Comandos básicos
- 2.2.2. Gestión de archivos
- 2.2.3. Gestión de hardware
- 2.2.4. Trabajo en red
- 2.2.5. Instalación y actualización de software
- 2.2.6. Configuración y acceso a hardware

2.3. Sistema operativo en tiempo real
- 2.3.1. Historia y propósito
- 2.3.2. El planificador (The scheduler)
- 2.3.3. Tareas (Tasks)
- 2.3.4. Sincronización de tareas (Task sinchronization)
- 2.3.5. Comunicación entre tareas (Message passing)
- 2.3.6. Otras funcionalidades
- 2.3.7. Manejo de interrupciones (Interrupt handling)
- 2.3.8. Características del tiempo real
- 2.3.9. Recursos adicionales
- 2.3.10. Control de recursos compartidos
- 2.3.11. Gestión y uso de memoria
- 2.3.12. Monitoreo y uso de recursos
- 2.3.13. Sistemas con multiprocesadores embebidos

UNIDAD III. Programación en Python para sistemas embebidos

Competencia:
Utilizar el lenguaje de programación Python, con apego a la estructura e instrucciones propias, para programar un sistema embebido, con actitud entusiasta y propositiva.

Contenido: **Duración:** 4 horas

3.1. Python versus C/C++
3.2. Introducción a Python
- 3.2.1. Tipos de datos
- 3.2.2. Operadores aritméticos
- 3.2.3. Operadores a nivel de bit
- 3.2.4. Operadores booleanos
- 3.2.5. Control de flujo
- 3.2.6. Ciclos
- 3.2.7. Funciones
- 3.2.8. Listas
- 3.2.9. Tuplas
- 3.2.10. Diccionarios
- 3.2.11. Vectores y matrices
- 3.2.12. Funciones numéricas

3.3. Bibliotecas firmware
- 3.3.1. Matemáticas
- 3.3.2. Numéricas
- 3.3.3. Puertos de propósito general (GPIO)

3.4. Sensores digitales
- 3.4.1. Sensores inteligentes

3.5. Comunicación con periféricos
- 3.5.1. USB
- 3.5.2. Bluetooth
- 3.5.3. Otros (CAN, OBD II, SSP, GPRS, etc.)
- 3.5.4. Bibliotecas firmware para comunicación serial

3.6. Comunicaciones y redes
- 3.6.1. Ethernet
- 3.6.2. WiFi

3.7. Programación multiproceso (Multithreaded Programming)
    3.7.1. Programación multihilos (Multithreading programming)
    3.7.2. Programación en tiempo real
    3.7.3. Bibliotecas firmware para multiprocesamiento
3.8. Técnicas avanzadas de depuración y optimización
    3.8.1. Incrementando la eficiencia del código
    3.8.2. Drecrementando el tamaño del código
    3.8.3. Problemas de optimización
    3.8.4. Reduciendo el uso de memoria RAM
    3.8.5. Técnicas de ahorro de energía

UNIDAD IV. Aplicaciones con sistemas embebidos

Competencia:
Desarrollar aplicaciones con sistemas embebidos, mediante la aplicación de técnicas de programación y el uso óptimo de recursos del sistema, para el desarrollo de tareas de tiempo real con monitoreo remoto vía internet, adquisición de datos o control, con actitud innovadora y responsabilidad social.

Contenido: **Duración:** 6 horas

4.1. Procesamiento de imágenes
- 4.1.1. Bibliotecas firmware para procesamiento digital de imágenes
  - 4.1.1.1. Operaciones con imágenes
  - 4.1.1.2. Histogramas y ecualización
  - 4.1.1.3. Filtros digitales para imágenes
  - 4.1.1.4. Procesamiento de video en tiempo real

4.2. Procesamiento de señales
- 4.2.1. Bibliotecas firmware para procesamiento de señales
- 4.2.2. Procesamiento de señales de audio
- 4.2.3. Filtros digitales para audio

4.3. Internet de las cosas (IoT) con sistemas embebidos
- 4.3.1. Introducción a IoT
- 4.3.2. Tecnologías (IoT)
- 4.3.3. Bibliotecas firmware para IoT
- 4.3.4. Monitoreo remoto vía internet
- 4.3.5. Control remoto vía internet

4.4. Temas selectos de sistemas embebidos

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO				
No. de Práctica	**Competencia**	**Descripción**	**Material de Apoyo**	**Duración**
UNIDAD II				
1	Instalar y manipular un sistema operativo de tiempo real en el sistema embebido, siguiendo el procedimiento correspondiente, para la realización de aplicaciones, con disciplina.	1. Descarga el software del sistema operativo. 2. Identifica requerimientos de hardware. 3. Formatea la unidad de almacenamiento. 4. Instala el sistema operativo. 5. Prueba la ejecución del sistema operativo para descartar errores. 6. Usa comandos del sistema operativo de tiempo real. 7. Elabora el reporte de laboratorio.	Computadora, software IDE, procedimiento de instalación, red de Internet, sistema embebido y monitor.	6 horas
UNIDAD III				
2	Desarrollar programas en lenguaje de Python, mediante el uso de su herramienta IDE, para la configuración y uso de puertos GPIO del sistema embebido, con responsabilidad.	1. Interconecta el sistema embebido. 2. Diseña y escribe un programa en lenguaje Python. 3. Ejecuta el código Python en el sistema embebido. 4. Prueba el funcionamiento del sistema. 5. En caso de fallas, depura el programa o el circuito. 6. Elabora el reporte de laboratorio.	Computadora, software IDE, red de Internet, sistema embebido, botones, LEDS, resistencias y monitor.	4 horas
3	Desarrollar programas en lenguaje de Python, mediante el uso de su herramienta IDE, para la configuración y uso sensores digitales e inteligentes, con orden.	1. Interconecta el sistema embebido y sensores digitales. 2. Diseña y escribe un programa en lenguaje Python. 3. Ejecuta el código Python en el	Computadora, software IDE, red de Internet, sistema embebido, botones, LEDS, resistencias y monitor.	6 horas

		sistema embebido. 4. Prueba el funcionamiento del sistema. 5. En caso de fallas, depura el programa o el circuito. 6. Elabora el reporte de laboratorio.		
4		1. Interconecta el sistema embebido y sensores inteligentes. 2. Diseña y escribe un programa en lenguaje Python. 3. Ejecuta el código Python en el sistema embebido. 4. Prueba el funcionamiento del sistema. 5. En caso de fallas, depura el programa o el circuito. 6. Elabora el reporte de laboratorio.	Computadora, software IDE, red de Internet, sistema embebido, botones, LEDS, resistencias y monitor.	4 horas
5	Desarrollar programas en lenguaje de Python, mediante el uso de su herramienta IDE, para la configuración y uso de periféricos externos, con honestidad.	1. Interconecta el sistema embebido con periféricos externos. 2. Diseña y escribe un programa en lenguaje Python. 3. Ejecuta el código Python en el sistema embebido. 4. Prueba el funcionamiento del sistema. 5. En caso de fallas, depura el programa o el circuito. 6. Elabora el reporte de laboratorio.	Computadora, software IDE, red de Internet, sistema embebido, botones, LEDS, resistencias, periféricos externos (teclados, pantallas LCD/Táctil, cámaras digitales, sensores digitales, sensores inteligentes, entre otros) y monitor.	10 horas
6	Desarrollar programas en lenguaje de Python, mediante el uso de su herramienta IDE, para la configuración y uso de redes y conexión a internet, con creatividad.	1. Interconecta el sistema embebido a Internet. 2. Diseña y escribe un programa en lenguaje Python. 3. Ejecuta el código Python en el sistema embebido.	Computadora, software IDE, red de Internet, sistema embebido, botones, LEDS, resistencias, punto de acceso y monitor.	4 horas

		4. Prueba el funcionamiento del sistema. 5. En caso de fallas, depura el programa o el circuito. 6. Elabora el reporte de laboratorio.		
7	Desarrollar programas en lenguaje de Python, para la realización de tareas de multiprocesamiento, empleando técnicas de cómputo paralelo, de forma eficiente.	1. Configura el sistema embebido para multiprocesamiento. 2. Diseña y escribe un programa en lenguaje Python. 3. Ejecuta el código Python en el sistema embebido. 4. Prueba el funcionamiento del sistema. 5. En caso de fallas, depura el programa o el circuito. 6. Elabora el reporte de laboratorio.	Computadora, software IDE, red de Internet, sistema embebido, botones, LEDS, resistencias y monitor.	12 horas
UNIDAD IV				
8	Desarrollar programas en lenguaje de Python, para el procesamiento de imágenes digitales, utilizando bibliotecas de funciones, con actitud innovadora.	1. Diseña y escribe un programa de aplicación en lenguaje Python para el procesamiento de imágenes digitales. 2. Ejecuta el código Python en el sistema embebido. 3. Prueba el funcionamiento del sistema. 4. En caso de fallas, depura el programa o el circuito. 5. Elabora el reporte de laboratorio.	Computadora, software IDE, red de Internet, sistema embebido, teclado, monitor y cámara digital.	4 horas
9	Desarrollar programas en lenguaje de Python, para el procesamiento de señales de audio, utilizando bibliotecas de funciones, con actitud propositiva.	1. Diseña y escribe un programa de aplicación en lenguaje Python para el procesamiento de señales de audio. 2. Ejecuta el código Python en el sistema embebido.	Computadora, software IDE, red de Internet, sistema embebido, teclado, monitor, micrófonos y bocinas con amplificador.	4 horas

		3. Prueba el funcionamiento del sistema. 4. En caso de fallas, depura el programa o el circuito. 5. Elabora el reporte de laboratorio.		
10	Desarrollar programas en lenguaje de Python, para el desarrollo de aplicaciones del Internet de las cosas, mediante el uso de bibliotecas firmware, con responsabilidad social.	1. Desarrolla e implementa aplicaciones en lenguaje Python para el Internet de las cosas. 2. Ejecuta el código Python en el sistema embebido. 3. Prueba el funcionamiento del sistema. 4. En caso de fallas, depura el programa o el circuito. 5. Elabora el reporte de laboratorio.	Computadora, software IDE, red de Internet, sistema embebido, teclado, monitor, puntos de acceso y periféricos externos (teclados, pantallas LCD/Táctil, cámaras digitales, sensores digitales, sensores inteligentes, entre otros).	10 horas

VII. MÉTODO DE TRABAJO

Encuadre: El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

Estrategia de enseñanza (docente)

- Exposición.
- Interpretación de hojas de datos.
- Resolución de ejercicios de programación.
- Instrucción guiada.
- Estudios de caso.
- Demostraciones.
- Simulaciones.

Estrategia de aprendizaje (alumno)

- Investigación documental.
- Resolución de ejercicios.
- Síntesis.
- Cuestionarios.
- Diseño de programas y circuitos.
- Trabajo en equipo.
- Exposición.
- Discusión.
- Análisis de programas.
- Elaboración de reportes de prácticas.

VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

Criterios de acreditación

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

Criterios de evaluación

- Evaluaciones…….…. 30%
- Prácticas de laboratorio…..…….. 40%
- Tareas…….. 10%
- Evidencia de desempeño..... 20%
  (Sistema embebido de alto rendimiento computacional)

Total…….. 100%

IX. REFERENCIAS

Básicas

ARM Limited. (2018). *ARM Architecting a secure foundation, realize true business value with IoT - From chip to cloud*. Recuperado el 19 de septiembre de 2018, de ARM Sitio web: https://www.arm.com/

Bakos, J.D. (2015). *Embedded Systems: ARM Programming and Optimization*. USA: Morgan Kaufmann.

Barr, M. & Massa, A. (2009). *Programming Embedded Systems: With C and GNU Development Tools.* USA: O'Reilly Media. [clásica]

Bell, C. (2017). *MicroPython for the Internet of Things.* USA: *Apress.*

Catsoulis, J. (2005). *Designing Embedded Hardware: Create New Computers and Devices*. USA: O'Reilly Media. [clásica]

Kurniawan, A. (2016). *Smart Internet of Things Projects*. USA: Packt Publishing

Lacamera, D. (2018). *Embedded Systems Architecture: Explore architectural concepts, pragmatic design patterns, and best practices to produce robust systems.* USA: Packt Publishing.

Tollervey, N.H. (2017). *Programming with MicroPython: Embedded Programming with Microcontrollers and Python.* USA: O'Reilly Media.

Valvano, J. W. (2014). *Real-Time Operating Systems for ARM® Cortex™-M microcontrollers*. USA: Jonathan W. Valvano.

Complementarias

Barr, M. (2018). *Embedded C Coding Standard*. Recuperado el 13 de agosto de 2018 de https://barrgroup.com/Embedded-Systems/Books/Embedded-C-Coding-Standard

Burns, A. & Wellings, A. (2009). *Real-Time Systems and Programming Languages.* Recuperado el 26 de septiembre de 2018 de http://www.cs.york.ac.uk/rts/books/RTSBookFourthEdition.html [clásica]

Embedded Linux Wiki (n.d.) Recuperado el 13 de agosto de 2018 de https://elinux.org/Main_Page

FreeRTOS Documentation PDF files. (n.d.). Recuperado el 13 de agosto de 2018 de https://www.freertos.org/Documentation/RTOS_book.html

Gay, W. (2018). *BEGINNING STM32: Developing with freertos, libopencm3 and gcc*. USA: Apress.

Hillar, G.C. (2016). *Internet of Things with Python.* USA: Packt Publishing.

Jim, C. (2018). *Real-time Operating Systems Book 1: The Foundations (The engineering of real-time embedded systems).* USA: Independently published.

Labrosse, J. J. (1999). *Embedded systems building blocks: Complete and ready-to-use modules in C.* USA: CMP Books. [clásica]

Labrosse, J. J. (2007). *Micro C/OS-II: The real-time kernel.* USA: CMP Books. [clásica]

Valvano, J. W. (2015). *Embedded systems: Real-time interfacing to ARM® Cortex™-M microcontrollers.* USA: Jonathan W. Valvano.

Valvano, J. W. (2017). *Embedded systems: Introduction to ARM® Cortex™-M microcontrollers*. USA: Jonathan W. Valvano.

Xiao, P. (2018). *Designing Embedded Systems and the Internet of Things (IoT) with the ARM mbed.* USA: Wiley.

Norris, D. (2016). *Python for Microcontrollers: Getting Started with MicroPython.* USA: McGraw-Hill Education TAB.

Real-Time Operating System: API and RTX Reference Implementation. (n.d.). Recuperado el 13 de agosto de 2018 de http://www.keil.com/pack/doc/CMSIS/RTOS/html/modules.html l

White, E. (2011). *Making Embedded Systems: Design Patterns for Great Softwar*. USA: O'Reilly Media. [clásica]

X. PERFIL DEL DOCENTE

El docente de esta asignatura debe poseer formación inicial en Ingeniería Electrónica/Computación o área afín, preferentemente con grado de maestría o doctorado en ciencias o ingeniería. Se sugiere experiencia docente o profesional de al menos tres años en el campo de electrónica o computación, actualización en su formación y práctica docente. Además, debe dominar el uso de instrumentos de laboratorio, así como tecnologías de la información y las comunicaciones, lenguajes de programación compilados e interpretados. Es deseable que domine información técnica en inglés y que sea capaz de comunicarse efectivamente, facilitar la colaboración y propiciar el trabajo en equipo. Ser una persona proactiva, innovadora, analítica, responsable, con un alto sentido de la ética y capaz de plantear soluciones metódicas a un problema dado, con vocación de servicio a la enseñanza.