Universidad Tecnológica de Pereira

FI013 – Sistemas Embebidos

Nombre del programa académico	**Ingeniería Electrónica**
Nombre completo de la asignatura	**Sistemas Embebidos**
Área académica o categoría	**Ingeniería Electrónica**
Semestre y año de actualización	**2024-I**
Semestre y año en que se imparte	**2024-I**
Tipo de asignatura	**[x] Obligatoria [ ] Electiva**
Número de créditos	**3**
Director o contacto del programa	**Arley Bejarano Martínez**
Coordinador o contacto de la asignatura	**Arley Bejarano Martínez**

Descripción y contenidos

1. Breve descripción

Materia teórica y práctica cuya finalidad es estudiar los sistemas embebidos y sus principios fundamentales. A lo largo del curso, se profundiza en la arquitectura interna de los microcontroladores y minicomputadoras, abordando aspectos como la unidad central de procesamiento (CPU), la memoria (RAM y ROM), y la programación en lenguajes de bajo nivel como C y ensamblador. Se enfatiza la interfaz con periféricos, incluyendo puertos de entrada/salida (GPIO), convertidores analógico-digitales (ADC), y comunicación serie (UART). Además, se estudian protocolos de comunicación como I2C, SPI y RS485, y se exploran diversas plataformas de hardware como PIC, STM32, ESP32, y minicomputadoras como Raspberry Pi y BeagleBone. La programación se realiza en C, C++, MicroPython y Python.

2. Objetivos

- Estudiar la arquitectura interna de microcontroladores y minicomputadoras.
- Implementar protocolos de comunicación como I2C, SPI, UART y RS485 para la interacción eficiente entre dispositivos en sistemas embebidos, destacando su implementación y uso en aplicaciones reales.
- Implementar y depurar proyectos en plataformas como PIC, STM32, ESP32, Raspberry Pi y BeagleBone, utilizando lenguajes de programación como C, C++, MicroPython y Python para resolver problemas específicos y optimizar el rendimiento de los sistemas embebidos.

3. Resultados de aprendizaje

- RAA1: Presenta conocimiento sobre la arquitectura y el funcionamiento interno de sistemas embebidos, incluyendo microcontroladores y minicomputadoras. Identifica y comprende los diversos componentes, la memoria, los registros, y los buses. RAP 2, 3

- RAA2: Desarrolla firmware y software para sistemas embebidos utilizando lenguajes como C, C++, MicroPython y Python, optimizando los recursos que presenta el hardware y garantizando la eficiencia y robustez del sistema. RAP 1, 10

- RAA3: Capacidad para interactuar de manera efectiva con diversos periféricos y módulos de comunicación en sistemas embebidos, como puertos de entrada/salida (GPIO), convertidores analógico-digitales (ADC) y protocolos de comunicación serie como I2C, SPI, UART y RS485. RAP 1, 2

4. Contenido

- Introducción a los sistemas embebidos (4H)

- Componentes internos, CPU, memoria (RAM y ROM), registros, buses (4H)
- Protocolos I2C, SPI, UART, RS485(8H)
- Manejo de interfaces de programación para microcontroladores (20H)
- Manejo ADC (2H).
- Interrupciones y manejo de pila (5H).
- Desarrollo de bibliotecas(10H).
- Envió de datos a internet(2H).
- Manejos de Linux(15H).
- Practicas variadas(10H).

5. Requisitos: Microcontroladores

6. Recursos

- Diapositivas de clase disponibles en la página web del curso.
- Software IDE del microcontrolador seleccionado.
- Zamarreño, C.R. (2021) Programación de Microcontroladores Paso a Paso: Ejemplos prácticos desarrollados en la nube. Barcelona: Marcombo.
- Bates, M. (no date) Programming 8-bit PIC microcontrollers in C Electronic Resource: With Interactive Hardware Simulation. Elsevier/Newnes
- Ward, Hubert. (2020). C Programming for the PIC Microcontroller: Demystify Coding with Embedded Programming. 10.1007/978-1-4842-5525-4.
- Marwedel, P. (2021). Embedded system design: embedded systems foundations of cyber-physical systems, and the internet of things (p. 433). Springer Nature.

7. Herramientas técnicas de soporte para la enseñanza

- Clase magistral.
- Estudio y análisis de casos prácticos.
- Realización de trabajos individuales.
- Talleres para solución en grupo sobre cada contenido de la materia.
- Consultas utilizando la bibliografía del curso.
- Proyecto integrador de asignaturas desarrollando software para solucionar problemas de ingeniería.
- Uso de videotutoriales para apoyar los conceptos vistos en clase.
- Creación de sitio web o aula virtual para la comunicación con estudiantes y almacenamiento de material de clase.

- **Trabajos en laboratorio y proyectos**

- En esta materia, se contemplan diferentes notas donde el estudiante deberá realizar simulaciones, montajes de circuitos y programación de firmware por medio de software especializado, dándole solución a problemas planteados en clase

- **Métodos de aprendizaje**
  - El enfoque principal de este curso será la interacción directa entre estudiantes y profesores, con el propósito de explicar los conceptos fundamentales de la teoría para el desarrollo de firmware. Se respaldará este proceso mediante la asignación de tareas extra-clase y la realización de talleres prácticos durante las clases.
  - Como una herramienta de refuerzo de contenidos, se realizará una asignación material complementario, para lograr el entendimiento del contenido, así como el refuerzo de los resultados de aprendizaje.
  - Se les presentará a los estudiantes un problema que deberán abordar mediante el desarrollo de firmware y la manipulación de periféricos en microcontroladores para encontrar una solución efectiva.

- **Métodos de evaluación**

- Evaluación Parcial 1 (Porcentaje de la materia: 20%): Examen escrito con preguntas teóricas sobre la arquitectura de sistemas embebidos, incluyendo microcontroladores y minicomputadoras, pidiendo a los estudiantes que identifiquen y describan componentes específicos como la CPU, memoria, registros y buses.RAA-1.
- Practicas (Porcentaje de la materia: 20%): Prácticas, talleres, exposiciones y trabajos escritos puesto en clase. RAA-1, RAA-2 y RAA-3.
- Evaluación Parcial 2 (Porcentaje de la materia: 25%): Examen con problemas prácticos donde los estudiantes deben desarrollar firmware utilizando lenguajes como C y C++, optimizando los recursos del hardware para dar solución a problemas reales.: RAA-2.
- Evaluación Final (Porcentaje de la materia 35%): Presentación de un proyecto final desarrollado utilizando firmware en lenguajes como C, C++ o Python, donde se debe resolver un problema real integrando diversos periféricos del sistema embebido, como GPIO, ADC, y módulos de comunicación serie (UART, I2C, SPI, RS485). Se evalúa el resultado de aprendizaje: RAA-3.

Universidad Tecnológica de Pereira