

Lenguajes de alto nivel relevantes para mecatrónica: C y Python

Ingeniería | Ingeniería mecatrónica

Descripción del Curso

Este curso de Ingeniería Mecatrónica está diseñado para estudiantes con interés en la integración de técnicas de programación, simulación y fundamentos de hardware, con énfasis en la aplicación práctica a sistemas mecatrónicos. La experiencia de aprendizaje se desarrolla en cuatro unidades que desarrollan desde habilidades de programación básica hasta conceptos de simulación y optimización de rendimiento. La población objetivo incluye personas a partir de 17 años sin restricción de edad específica y orientadas a adquirir competencias para diseñar, analizar y justificar decisiones en contextos reales de mecatrónica. Unidad 1: Exploración de sintaxis y librerías. Se realizan ejercicios prácticos para escribir scripts en Python que lean datos simulados, manipulen arreglos y generen gráficos sencillos. Puntos clave: estructuras de control y datos, importación de librerías, manejo de datos y visualización básica. Unidad 2: Procesamiento de datos y visualización. Se trabajan técnicas de limpieza de datos, cálculo de estadísticas básicas y creación de gráficos que faciliten la interpretación de resultados. Aprendizajes centrales: selección de tipos de gráficos adecuados y claridad en la comunicación de hallazgos. Unidad 3: Simulación de un sistema mecatrónico. Se modela un sistema dinámico, se realizan simulaciones en Python y se analiza la respuesta ante distintos escenarios. Aprendizajes: evaluación de parámetros del modelo y validación de resultados frente a criterios de diseño. Unidad 4: Introducción a C para mecatrónica. Se crea un programa simple en C, se compila y se compara su rendimiento frente a una implementación equivalente en Python para una tarea básica. Aprendizajes: comprender límites de rendimiento y considerar hooks para interacción con hardware. Objetivo general y evaluación. La evaluación está orientada a verificar el logro del objetivo general y de los objetivos específicos mediante actividades prácticas, evaluaciones formativas y un proyecto integrador. Se contemplan quizzes y tareas cortas al finalizar cada tema, desarrollo de un script Python que procese datos, realice una simulación y genere visualizaciones, interpretación de resultados y justificación de decisiones de diseño, además de una rúbrica de desempeño con criterios y retroalimentación detallada. Duración: 4 semanas.

Competencias

- Capacidad para diseñar, implementar y analizar soluciones de software que integren procesamiento de datos, visualización y simulación de sistemas mecatrónicos, aplicando buenas prácticas de programación y comunicación técnica. - Habilidad para convertir requerimientos de ingeniería en modelos computacionales, realizar pruebas de rendimiento y validar resultados frente a escenarios reales o simulados. - Competencia para comparar herramientas y enfoques en Python y C, evaluar trade-offs de rendimiento y proponer soluciones adecuadas a hardware y restricciones de tiempo real. - Desarrollo de pensamiento crítico, trabajo colaborativo y toma de decisiones fundamentadas para comunicar hallazgos, justificaciones de diseño y recomendaciones a audiencias técnicas.

Requerimientos

- Computadora con sistema operativo moderno y conectividad a internet. - Python 3.x instalado (incluyendo paquetes numpy, pandas, matplotlib) y un entorno de desarrollo (por ejemplo, VS Code, PyCharm) configurado para proyectos de Python. - Compilador de C (por ejemplo, gcc) y un entorno compatible para compilar programas en C. - Conocimientos básicos de programación (lógica, estructuras de datos) y conceptos elementales de álgebra y física aplicados a sistemas dinámicos. - Acceso a herramientas de documentación y repositorios para gestión de código (opcional, pero recomendable). - Disponibilidad para actividades de 4 semanas y entrega de un proyecto integrador al final del curso.

Unidades del Curso

Unidad 1: Unidad 1: Lenguajes de alto nivel relevantes para mecatrónica - C y Python

Objetivos de Aprendizaje

- Identificar y utilizar librerías de Python relevantes para el procesamiento de datos, como NumPy y SciPy, así como herramientas de visualización como Matplotlib para representar resultados de manera interpretable.
- Diseñar y escribir scripts en Python para recoger, limpiar y procesar datos provenientes de sensores o simulaciones y extraer información útil para la toma de decisiones.
- Modelar y simular dinámicas simples de sistemas mecatrónicos (p. ej., respuesta de un actuador o control de un motor) utilizando Python y bibliotecas adecuadas, interpretando la salida de las simulaciones.
- Introducir conceptos básicos de C aplicados a mecatrónica, incluyendo fundamentos de programación y compilación, para comprender las limitaciones y el rendimiento en sistemas embebidos.

Contenidos Temáticos

Tema 1: Fundamentos de Python y herramientas esenciales

1. Descripción corta: sintaxis básica, estructuras de control, funciones, manejo de archivos y primeros pasos con librerías clave para ciencia de datos e ingeniería.
2. Entornos de desarrollo recomendados, estilo de código y buenas prácticas de escritura de scripts para proyectos mecatrónicos.
3. Ejemplos de flujo de trabajo en mecatrónica: lectura de datos simulados, procesamiento y generación de gráficos simples.