Imagen que contiene Texto

Descripción generada automáticamente

**Título curso:**

Sistemas Operativos en Tiempo Real 2

**Descripción corta:**

El estudiante deberá de familiarizarse con los conceptos generales de un sistema operativo en tiempo real, así como sus implicaciones dentro de un sistema embebido.

Este módulo se pretende llevar a cabo en 40 horas desde el punto de vista teórico y práctico para preparar al estudiante para los siguientes módulos en el track de sistemas embebidos.

**Datos importantes del curso**:

Fecha de inicio: TBD

Duración: 6 semanas

Dedicación: 3 horas por semana

Horario: TBD

Costo: Gratuito

Modalidad: Online vía Teams

Certificación: IA. Center & SEP

Hardware, favor de revisar la sección de “Pre-requisitos”

**Condiciones especiales del curso**

Contar con la tarjeta de evaluación sugerida

Cuenta de estudiante de GitHub, gratuita

**Objetivo del curso**

La intención de este curso es dar una base teórica y práctica para que el participante de este curso pueda comprender que es un sistema operativo en tiempo real, sus aplicaciones, y su implementación en un sistema embebido.

Los temas principales para desarrollar durante este módulo serán la creación y manejo de tareas, manejo de recursos y buenas prácticas en el diseño de una aplicación con un sistema operativo de tiempo real.

**Dirigido a**

Estudiantes o recién egresados de ingeniería, así como recién egresados de carrera técnica de electrónica, mecatrónica, sistemas computacionales con especialidad en hardware o carreras afines. Personas que quieran practicar los conocimientos adquiridos previamente de programación y electrónica, aplicados específicamente en un sistema embebido.

**Perfil de ingreso**

La persona que desee ingresar debe de tener acceso a un equipo de cómputo, así como acceso de administrador al mismo, así como conexión a internet. Es necesario tener una cuenta de GitHub.

También deben de cumplirse los pre-requisitos listados en secciones posteriores.

**Perfil de egreso**

La persona que egrese tendrá un entendimiento de las implicaciones de utilizar un sistema operativo en tiempo real y podrá utilizar el conocimiento adquirido para desarrollar aplicaciones que incluyan un microcontrolador.

**Prerrequisitos**

*Teóricos:*

* Clase de: Programación en C, Herramientas de desarrollo de software, Proceso de desarrollo de software, Micros 2 y RTOS1.
* Conocimiento básico de electrónica digital
* Entendimiento de conceptos básicos de microcontroladores.
* Inglés intermedio

*Hardware e infraestructura:*

Se espera que este hardware e infraestructura sea el que se utilice durante toda la serie de módulos del track de sistemas embebidos.

* Computadora para ejecutar las herramientas de desarrollo (Windows o Linux).
* Cuenta GitHub
* Tarjeta de evaluación sugerida: ESP32-C3-DevKitM-1
* Cable de USB micro.

**Reseña del instructor**

Nombre: MI César Rodríguez Esqueda

César cuenta con una ingeniería en mecatrónica del Instituto Tecnológico de Estudios Superiores de Monterrey, Campus Chihuahua y con una maestría en ingeniería mecatrónica del Instituto Tecnológico de México, Campus Chihuahua.

Actualmente se encuentra laborando en el área de Firmware en Resideo Centro de Diseño, Chihuahua con experiencia en desarrollo de producto por 6 años, todos ellos en desarrollo de software embebido para electrónica de consumo.

Durante su experiencia laboral César ha trabajado en distintos proyectos que hacen uso del sistema operativo “FreeRTOS”, ha trabajado con creación y manejo de tareas y ha realizado análisis de aplicaciones que hacen uso de este sistema operativo en tiempo real.

Auxiliar: MI José Miguel Díaz Arriaga

**Duración**

Número de horas: 40 horas

1. Asignación de memoria. 2 hrs.
   1. Stack de threads y stack normal.
   2. Stack / heap
   3. Mapa de memoria sin RTOS vs Mapa de memoria con RTOS
2. Aplicación y entendimiento de los siguientes componentes en FreeRTOS. 35 hrs.
   1. Multitarea.
      1. Manejo del tiempo de procesamiento.
      2. Implementación de tareas
         1. Prioridades
         2. Stack borrar tareas
      3. Tarea “Idle”
      4. Tarea “Tmr Svc”
      5. Politicas de “Scheduler” (Preemption vs cooperative)
      6. Estados de una tarea
      7. Cambio de Contexto
      8. Borrar tareas
   2. “Tick” del RTOS.
   3. Delays.
   4. Notificaciones
   5. Queue.
   6. Manejo de Interrupciones.
   7. Manejo de Recursos
      1. Mutex
      2. Semáforos
      3. Dead lock.
3. Análisis de stack. 1 hr.
   1. FreeRTOS StackWatermark