Práctica 1,2 & 3. Instalación, conociendo el entorno y creación de procesos

Antecedentes

* Dentro de la sección de antecedentes deberá realizar una investigación donde de respuesta a lo siguiente:
  + ¿Qué es un sistema operativo en tiempo real?
  + ¿Cuál es la diferencia entre un sistema operativo en tiempo real contra un sistema operativo convencional?
  + Ventajas y desventajas de los sistemas operativos en tiempo real.
  + Tipos de sistemas operativos en tiempo real que hay en el mercado.
  + Arquitectura de microcontroladores que soportan un RTOS.

Material necesario

* Tarjeta de desarrollo ESP32.
* Osciloscopio.
* Arduino Ide.
* Resistencias y LED’s.
* Push button.

Procedimiento

1. Estructura de programación.
   1. Genere un código en el entorno de Arduino que considere los siguiente:
      * Toggle de dos salidas digitales (LEDs).
      * La primera salida digital con un periodo de 500 ms.
      * La segunda salida digital con un periodo de 323 ms.
   2. Genere un código en el entorno de Arduino que considere los siguiente:
      1. Mismas características al código anterior, pero utilizando FreeRTOS.
      2. Considere ambas tareas con la misma prioridad.
   3. Realice las siguientes modificaciones a los códigos.
      1. Integre un botón, el cual tendrá como función modificar el periodo. Al presionar el botón, la primera salida cambiará su periodo a 1 segundo y la segunda salida cambiará su periodo a 500 ms. Revise en el osciloscopio ambas señales.
      2. ¿Es posible diferenciar cual de las dos señales se habilita primero?
      3. ¿El periodo de trabajo corresponde al deseado?
      4. Modifique las prioridades y analice las últimas dos preguntas nuevamente.
2. Generación de tareas.
   1. Utilizando FreeRTOS, genere 7 tareas:
      1. La primera tarea enviará por puerto UART el mensaje: **Unidad Profesional Interdisciplinaria de Ingenieria Campus Zacatecas IPN**. Considere una velocidad de transmisión de 300 baudios (el más lento que permite el entorno Arduino IDE). Durante el envío considere que una salida digital esté en alto.
      2. La segunda tarea enviará por puerto UART el mensaje: **1**. Considere una velocidad de transmisión de 300 baudios (el más lento que permite el entorno Arduino IDE). Durante el envío considere que una salida digital esté en alto.
      3. La tercera tarea enviará por puerto UART el mensaje: **2.** Considere una velocidad de transmisión de 300 baudios (el más lento que permite el entorno Arduino IDE). Durante el envío considere que una salida digital esté en alto.
      4. La cuarta tarea enviará por puerto UART el mensaje: **3.** Considere una velocidad de transmisión de 300 baudios (el más lento que permite el entorno Arduino IDE). Durante el envío considere que una salida digital esté en alto.
      5. La quinta tarea enviará por puerto UART el mensaje: **4.** Considere una velocidad de transmisión de 300 baudios (el más lento que permite el entorno Arduino IDE). Durante el envío considere que una salida digital esté en alto.
      6. La sexta tarea enviará por puerto UART el mensaje: **5**. Considere una velocidad de transmisión de 300 baudios (el más lento que permite el entorno Arduino IDE). Durante el envío considere que una salida digital esté en alto.
      7. La séptima tarea enviará por puerto UART el mensaje: **6**. Considere una velocidad de transmisión de 300 baudios (el más lento que permite el entorno Arduino IDE). Durante el envío considere que una salida digital esté en alto.
   2. Considere las mismas prioridades todas las tareas y analice lo siguiente:
      1. ¿El mensaje por puerto UART tiene alguna consistencia u orden?
      2. Utilizando el osciloscopio, tome como base la salida digital de la tarea 2. Compare esta salida con las tareas 3, 4, 5, 6 y 7. Analice los resultados.
   3. Considere al menos 4 cambios en las prioridades y analice lo siguiente:
      1. ¿El mensaje por puerto UART tiene alguna consistencia u orden?
      2. Para una de las 4 configuraciones utilizadas, tome como base la salida digital de la tarea 2. Compare esta salida con las tareas 3, 4, 5, 6 y 7. Analice los resultados.
3. Trabajo con los estados de las tareas.
   1. Utilizando FreeRTOS, genere 3 tareas:
      1. Tarea 1, Envío de mensaje por UART, mensaje y velocidad libre.
      2. Tarea 2, Envío de mensaje por UART, mensaje y velocidad libre.
      3. Tarea 3, Envío de mensaje por UART, mensaje y velocidad libre.
   2. La tarea 1, se deberá ejecutar 1 sola vez y después será eliminada.
   3. La tarea 2, se deberá suspender por 2 segundos y después habilitarse nuevamente.
   4. La tarea 3, se ejecuta indefinidamente. Deberá tener máxima prioridad.
   5. Reporte una captura de los mensajes obtenidos por consola.
4. Dentro de las conclusiones considere lo siguiente:
   * 1. Ventajas y desventajas de utilizar RTOS.
     2. En qué condiciones considera que se debería implementar un RTOS.

Entregables

* Todos los ejercicios dentro de la sección de procedimiento, deberá tomar capturas donde muestre claramente los resultados de sus simulaciones principalmente de lo que se solicita.
* Repositorio en github con los códigos de cada uno de los ejercicios.
* Reporte digital.
  + Deberá contener lo solicitado.
  + En la sección de procedimiento donde se solicita documentar o analizar deberá realizar el análisis en equipo.
  + No olvide utilizar el formato del reporte solicitado.
  + El reporte se deberá subir por todos los integrantes del equipo a la asignación de Teams.