Trabajo Práctico 4

Sistemas Operativos en Tiempo Real

Introducción

En el siguiente informe se va a explicar el la implementacion de un sensado de temperatura usando un microcontrolador LM3S811. El objetivo del siguiente trabajo es aprender a diseñar, crear y entender la implementacion de un sistema operativo de tiempo real.

Main.c

En este archivo se realiza la inicializacion del proyecto. Se inicializan las colas de los valores sensados y los valores filtrados, se inicializa el hardware necesario para las tareas implementadas, se crean dichas tareas y se inicializa el scheduler, el cual es el encargado de orquestrar o definir y decidir las tareas que se van a ejecutar.

UartHandler.c

Se define las funciones del UART, este debe ser capaz de recibir el valor que se utilizara para el filtro (explicado mas adelante). Si recibe un valor menor o igual 0, se setea el valor del calculo de las mediciones a 10 (valor por defecto), de lo contrario, si se setea con un valor mayor a cero, se setea el valor de N al valor ingresado.

Sensor.c

Se definen las funciones del sensor, en este unicamente lo que se hace es generar valores aleatorios en un rango, definidos por TEMP_MIN y TEMP_MAX (12 y 40 respectivamente). Dichos valores se envian a la cola xSensorValueQueue. Dicha cola luego es accedida por el la tarea del filtro.

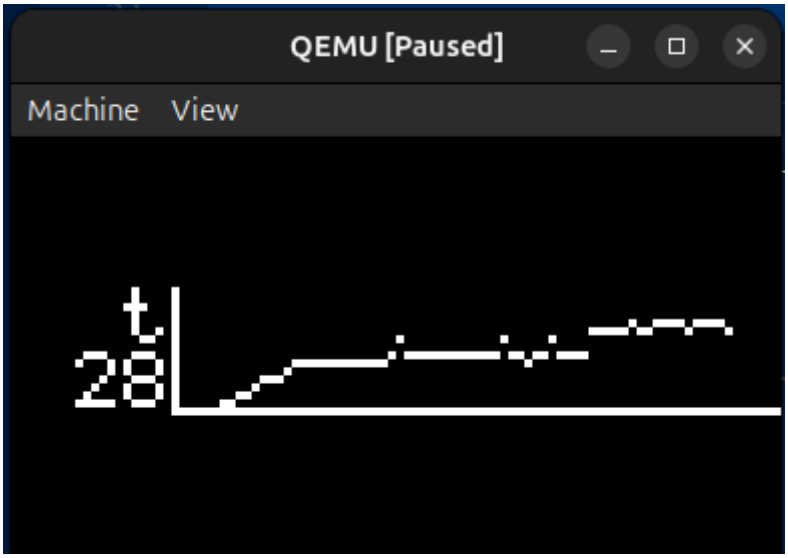
Filter.c

Recibe los valores ingresados en xSensorValueQueue, realiza el calculo correspondiente para el tipo del filtro diseñado, y se gaurda en la cola xFilteredValueQueue. El tipo de filtro implementado es un **filtro pasa bajo de media movil exponencial simple,** el cual suprime variaciones en la señal de entrada (no se notara tanto en el tp ya que esto se considera cuando la señal toma valores aproximadamente parecidos, como por ejemplo 24, 26, 29, 27, etc). Cuanto mayor sea el valor de N

mas lenta sera la respuesta del filtro y mayor suavizacion de la onda. Si N es pequeño, el filtro responde mas rapido a cambios de la señal.

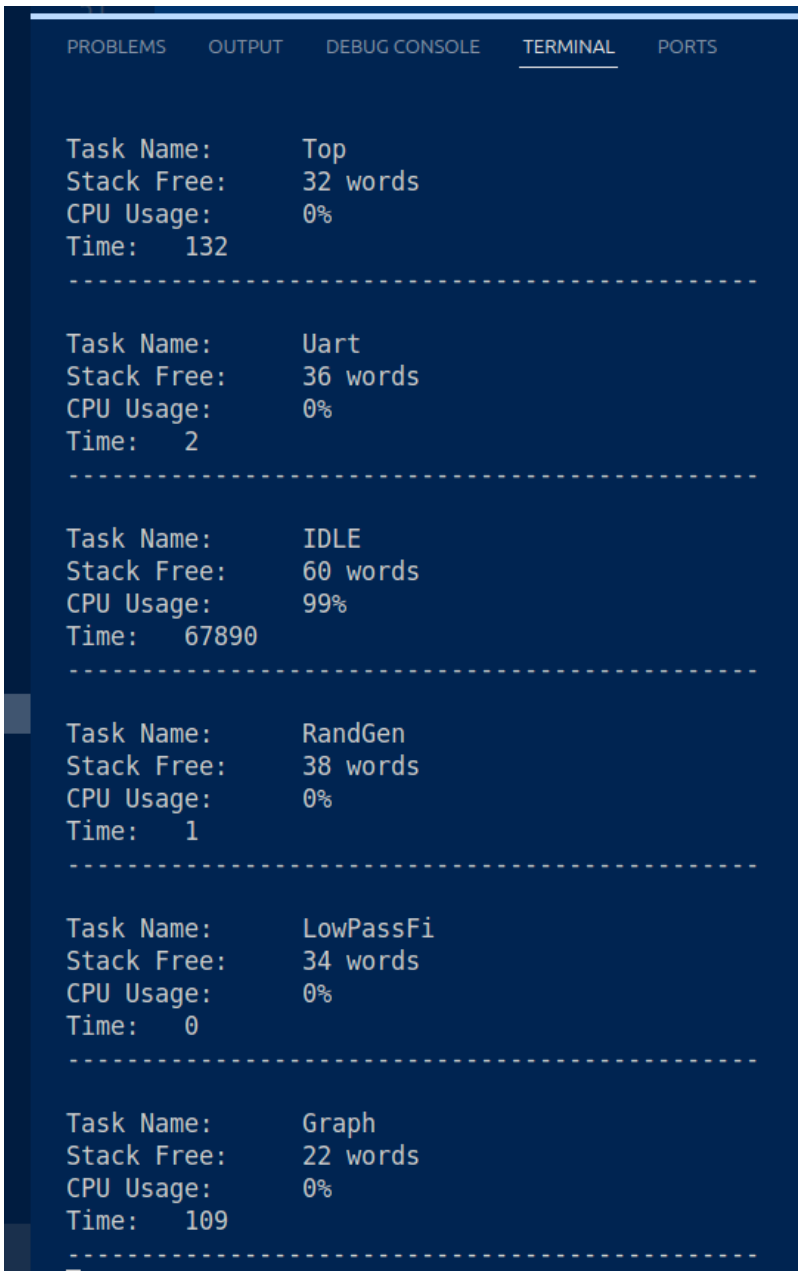
Graph.c

Este archivo contiene las funciones necesarias para la tarea de graficacion de la señal de salida filtrada. Este imprime una “t” (de temperatura) al lado del grafico, junto con los ejes cartesianos caracteristicos.



Top.c

Este archivo se encarga de imprimir de manera periodica informacion sobre las tareas ejecutadas, como el nombre de la CPU, el tiempo de ejecucion, stack libre de cada tarea (calculado con la funcion *uxTaskGetStackHighWaterMark*)


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Task Name: Top
Stack Free: 32 words
CPU Usage: 0%
Time: 132
Task Name: Uart
Stack Free: 36 words
CPU Usage: 0%
Time: 2
Task Name: IDLE
Stack Free: 60 words
CPU Usage: 99%
Time: 67890
Task Name: RandGen
Stack Free: 38 words
CPU Usage: 0%
Time: 1
Task Name: LowPassFi
Stack Free: 34 words
CPU Usage: 0%
Time: 0
Task Name: Graph
Stack Free: 22 words
CPU Usage: 0%
Time: 109