

## Práctica 2

# Gestion de tareas en ESP-IDF

<b>2. Gestion de tareas en ESP-IDF</b>	<b>1</b>
2.1. Objetivos	1
2.2. Material de consulta	1
2.3. Desarrollo de la práctica	2
2.4. Instrucciones de entrega	2

### 2.1. Objetivos

El objetivo de esta práctica es conocer los mecanismos para la gestión de tareas que ofrece FreeRTOS, concretamente en su porting ESP-IDF (con alguna particularidad por el hecho de estar adaptado a tener 2 cores).

Trabajaremos los siguientes aspectos del API de ESP-IDF:

- Creación y destrucción de tareas.
- Uso de *delays* para tareas periódicas (veremos mejores opciones en el futuro).
- Comunicación y sincronización de tareas mediante colas.

### 2.2. Material de consulta

Para ver los detalles de cada aspecto de esta práctica se recomienda la lectura de los siguientes enlaces:

- <https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/system/freertos.html#>. Documentación del API ofrecida por ESP-IDF.
- <https://www.freertos.org/features.html>. Documentación oficial de FreeRTOS.
- *Mastering de FreeRTOS Real Time Kernel*, de Richard Barry. Se puede descargar en [https://www.freertos.org/fr-content-src/uploads/2018/07/161204\\_Mastering\\_the\\_FreeRTOS\\_Real\\_Time\\_Kernel-A\\_Hands-On\\_Tutorial\\_Guide.pdf](https://www.freertos.org/fr-content-src/uploads/2018/07/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf)

## 2.3. Desarrollo de la práctica

Deberás desarrollar un sistema con tres tipos de tareas:

- Tarea *sensor*. Esta tarea emulará las lecturas de un sensor de forma periódica. Para ello, generará un número entero de forma aleatoria cada  $n$ ms (siendo  $n$  un parámetro que recibirá la tarea en su creación). Para realizar el *muestreo periódico* se pasará a la tarea a modo bloqueado el tiempo necesario usando la función `vTaskDelay()`.
- Tarea *filtro*. Esta tarea recibirá las muestras de la tarea anterior. Realizará una media ponderada de las últimas cinco muestra usando los coeficientes (0.05, 0.10, 0.15, 0.25, 0.45)<sup>1</sup>. Durante las primeras cuatro muestras no realizará ningún cómputo.
- Tarea *controlador*. Esta tarea emula el control de nuestra aplicación. Quedará a la espera de notificaciones por parte de tareas tipo *filtro*. Cuando le llegue alguna, imprimirá por el puerto serie el nombre de la tarea que le envía el dato, el valor recibido y un *timestamp* del instante en que se recogió la muestra por parte de la correspondiente tara *sensor*.<sup>2</sup>

El sistema final contará con 2 tareas tipo *sensor*, con su correspondientes 2 tareas tipo *filtro* y una única tarea tipo *controlador*. La figura 2.1 ilustra la estructura del proyecto. Los argumentos para las dos tareas tipo *sensor* se definirán a través de *menuconfig*.

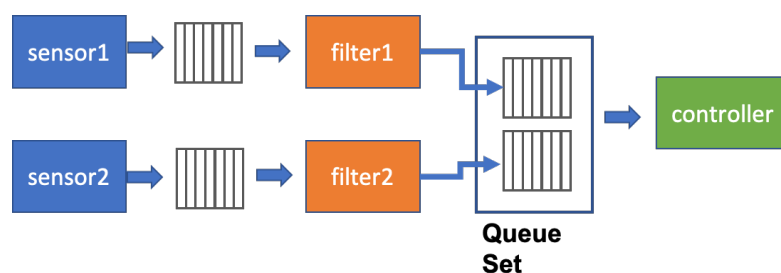


Figura 2.1: Diagrama de tareas del sistema

Asimismo, la tarea *controlador* imprimirá periódicamente (cada 10 segundos) una estadística de las tareas del sistema haciendo uso de la llamada `vTaskGetRunTimeStats()`. Para poder usar dicha función deberás activar una opción en *menuconfig*. Puedes consultar qué debe configurarse en [https://github.com/espressif/esp-idf/tree/release/v4.1/examples/system/freertos/real\\_time\\_stats](https://github.com/espressif/esp-idf/tree/release/v4.1/examples/system/freertos/real_time_stats).

## 2.4. Instrucciones de entrega

La entrega se realizará por el Campus Virtual. Deberá constar del código fuente (sólo ficheros `.c`, `.h` y `Kconfig.projbuild`) en un fichero comprimido. Asimismo, se entregará un PDF que incluya también el código fuente desarrollado y en el que se respondan las siguientes cuestiones.

<sup>1</sup>El coeficiente 0.45 se usará para la muestra más reciente

<sup>2</sup>Consulta la documentación de ESP-IDF sobre el *System Timer* ([https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/system/system\\_time.html](https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/system/system_time.html))

**Cuestiones**

- ¿Qué prioridad tiene, por defecto, la tarea que se crea en el arranque para ejecutar `app_main()`?
- ¿Qué prioridades has asignado a cada uno de las tareas que has creado?
- ¿Por qué es imprescindible asignar un *core* fijo a las tareas de tipo *filtro* ?