

Práctica 3

Entrada/Salida con ESP32 (ESP-IDF)

3. Entrada/Salida con ESP32 (ESP-IDF)	1
3.1. Objetivos	1
3.2. Material de consulta	1
3.3. Desarrollo de la práctica	2
3.3.1. Modificación (OPCIONAL)	2
3.4. Instrucciones de entrega	3

3.1. Objetivos

El objetivo de esta práctica es conocer los mecanismos de entrada/salida ofrecidos por ESP-IDF para interactuar con dispositivos usando ESP32.

Trabajaremos los siguientes aspectos del API de ESP-IDF:

- *High Resolution Timers*.
- Conceptos de muestro e interrupciones.
- Configuración y uso del controlador GPIO.
- Uso de sensores (*TouchSensor* y sensor de efecto Hall).

3.2. 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/esp_timer.html. Documentación del API de *Timers* propios de ESP-IDF.
- <https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/peripherals/gpio.html>. Documentación sobre el API para el uso del GPIO del ESP32.

- *ESP32 Technical Reference Manual*. De consulta obligada para cada aspecto del ESP32. El capítulo 2 es de interés en lo referente al mapa de memoria. Se puede descargar en https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.
- https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/peripherals/touch_pad.html. Documentación y API para el uso del *Touch Sensor*.
- <https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/peripherals/adc.html#api-reference>. Incluye información de uso del sensor de efecto *Hall*.
- <https://github.com/espressif/esp-idf/tree/release/v4.1/examples/peripherals>. Ejemplos de uso de los dispositivos utilizados en la práctica.

3.3. Desarrollo de la práctica

Se deberá implementar un cronómetro con la funcionalidad vista en clase. En concreto:

- Contará únicamente minutos y segundos.
- Dispondrá de una entrada *Start/Stop* que comienza/para el cronómetro.
 - Se implementará usando el *TouchSensor* configurado por interrupciones.
- Dispondrá de una entrada *Reset* que pondrá a 0 la cuenta y parará el cronómetro.
 - Se implementará muestreando el sensor de efecto *Hall*. Cuando supere un umbral, se considerará activo.
- Se configurará un *timer* (alta resolución) para llevar la cuenta de los segundos transcurridos
 - A cada vencimiento del *timer*, se modificará el valor del tiempo transcurrido si el cronómetro está en marcha.
- Se imprimirá la cuenta (minutos:segundos) cada segundo por el puerto serie, independientemente de que el cronómetro esté activo o no.

3.3.1. Modificación (OPCIONAL)

Para conseguir la máxima en la puntuación en la práctica, se podrá realizar la siguiente modificación en el uso del *timer*:

- Se configurará un *timer* (alta resolución) para llevar la cuenta de los segundos transcurridos
 - A cada vencimiento del *timer*, se modificará el valor de un pin configurado como salida.
 - Dicho pin se conectará a otro pin configurado como entrada, por interrupciones por ambos flancos.

La ISR de dicha interrupción será la que haga el trabajo que, en la versión original, se encargaba al *callback* del *timer*.

3.4. 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 en el que se especifique el diseño de la aplicación: tareas, elementos de comunicación/sincronización, uso de *timers* o eventos.... Se recomienda usar un esquema gráfico que muestre cada elemento y su interacción con el resto.