

Universidad Autónoma de Baja California
Facultad de Ciencias químicas e ingeniería



Asignatura: Sistemas embebidos

Práctica de Laboratorio #4

Alumno:

Rodríguez Contreras, Raul Arturo
1261510

Docente: Lara Camacho, Evangelina

Objetivo

El alumno se familiarizará con el uso del periférico I2C usando el sistema embebido ESP32 Devkit 1 para desarrollar aplicaciones para sistemas basados en microcontrolador para aplicarlos en la resolución de problemas de cómputo, de una manera eficaz y responsable.

Equipo

Computadora personal con conexión a internet.

Teoría

Revisar los cambios en el código de I2C en la última versión del SDK (las nuevas funciones, estructuras, etc.) en:

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/migration-guides/release5.x/5.2/peripherals.html>

Y actualizar los ejemplos vistos en clase a los cambios en el código de I2C.

Sintaxis y Parámetros:

Ejemplo de uso

Implemente en dos ESP32 ESP-IDF y un sensor I2C una comunicación privada haciendo uso de I2C y tareas. La implementación debe ser eficiente en el uso de recursos de cómputo (procesador, memoria y periféricos).

Un ESP32 es el master y el otro ESP32 es el slave. El slave tiene conectado un sensor I2C. El master solicita al slave cada 2 segundos la medición actual del sensor. Cuando el slave recibe la petición, lee el sensor usando I2C y envía la medición al master por I2C. El master despliega en la terminal la medición por medio de UART.

Si el master no recibe del slave la medición en los siguientes 500 ms después de hacer la solicitud hace máximo dos intentos más de comunicación (es decir, envía la solicitud nuevamente y espera 500 ms por la respuesta). Si aún con los intentos no recibe la respuesta, el master despliega en la terminal "Comunicación terminada, el periférico no responde", y se termina la comunicación (en este estado el master puede ejecutar un ciclo infinito sin ninguna operación).

El formato de la solicitud del master es la siguiente:

Header = 0x27	Cmd = 0x9F
---------------	------------

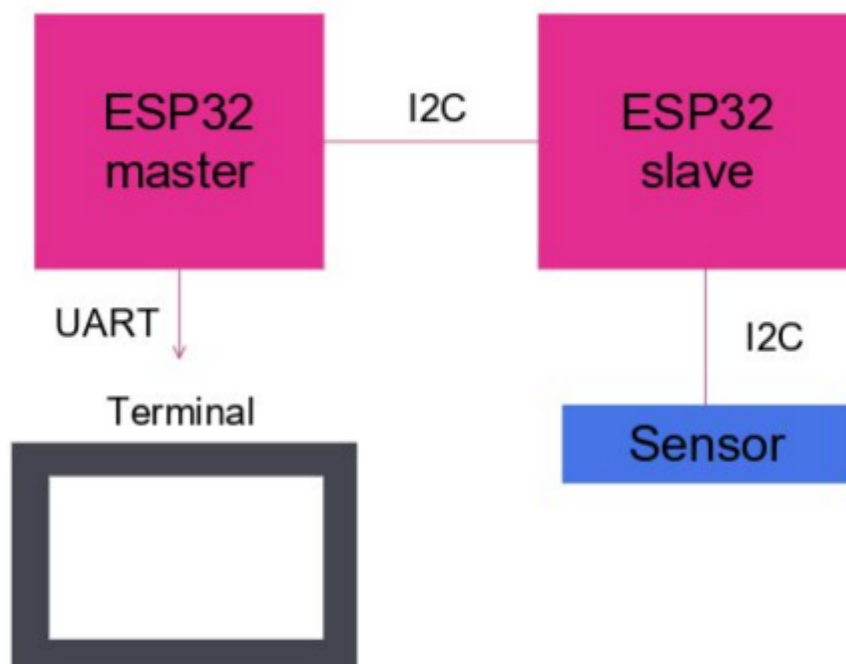
Y el formato de la respuesta del slave es la siguiente:

Header = 0x27	Cmd = 0x9F	Medición (n bytes)
---------------	------------	--------------------

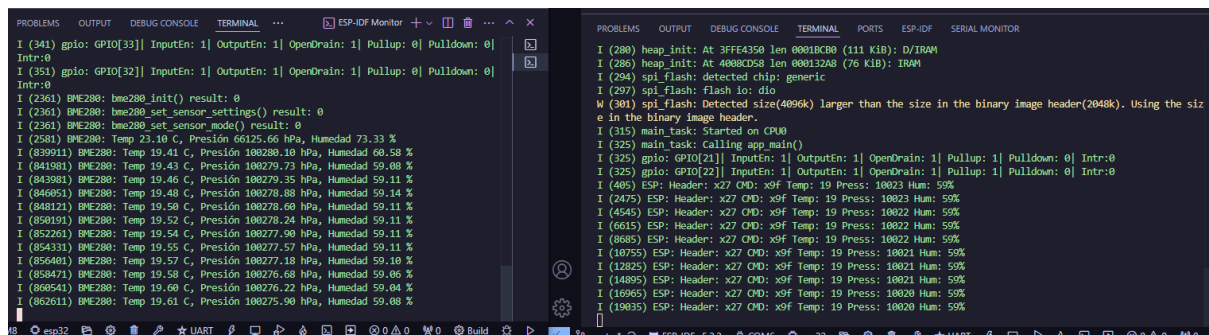
Teoría

Describe a detalle la función `xEventGroupSync` la cual activa bits dentro de un grupo de eventos y luego espera a que se active una combinación de bits dentro del mismo grupo de eventos. Incluye un ejemplo de uso diferente al de la documentación. La función `xEventGroupSync` es utilizada para sincronizar múltiples tareas en un sistema de tiempo real mediante la activación y espera de ciertos bits dentro de un grupo de eventos. Esto es útil cuando varias tareas necesitan alcanzar un estado común antes de continuar su ejecución, logrando una coordinación precisa.

Fig. 1. Diagrama a bloques.



En la siguiente imagen, se muestra a la izquierda el ESP32 slave, que obtiene la información del sensor, y la derecha muestra el ESP32 Master, que solicita la información y la despliega.



```
I (341) gpio: GPIO[33] InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 0| Pulldown: 0|
Intr:0
I (351) gpio: GPIO[32] InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 0| Pulldown: 0|
Intr:0
I (2361) BME280: bme280_init() result: 0
I (2361) BME280: bme280_set_sensor_settings() result: 0
I (2361) BME280: bme280_set_sensor_mode() result: 0
I (2581) BME280: Temp 23.10 C, Presión 66125.66 hPa, Humedad 73.33 %
I (839911) BME280: Temp 19.41 C, Presión 100280.10 hPa, Humedad 60.58 %
I (841803) BME280: Temp 19.43 C, Presión 100279.73 hPa, Humedad 59.08 %
I (843881) BME280: Temp 19.46 C, Presión 100279.35 hPa, Humedad 59.11 %
I (846951) BME280: Temp 19.48 C, Presión 100278.88 hPa, Humedad 59.14 %
I (848121) BME280: Temp 19.50 C, Presión 100278.60 hPa, Humedad 59.11 %
I (850191) BME280: Temp 19.52 C, Presión 100278.24 hPa, Humedad 59.11 %
I (852261) BME280: Temp 19.54 C, Presión 100277.90 hPa, Humedad 59.11 %
I (854331) BME280: Temp 19.55 C, Presión 100277.57 hPa, Humedad 59.11 %
I (856401) BME280: Temp 19.57 C, Presión 100277.10 hPa, Humedad 59.10 %
I (858471) BME280: Temp 19.58 C, Presión 100276.68 hPa, Humedad 59.06 %
I (860541) BME280: Temp 19.60 C, Presión 100276.22 hPa, Humedad 59.04 %
I (862611) BME280: Temp 19.61 C, Presión 100275.90 hPa, Humedad 59.08 %

I (280) heap_init: At 3FFE4350 len 00018CB0 (111 KiB): D/IRAM
I (286) heap_init: At 4008CD58 len 000132A8 (76 KiB): IRAM
I (294) spi_flash: detected chip: generic
I (297) spi_flash: Flash io: dio
W (301) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
I (315) main task: Started on CPU0
I (325) main task: Calling app.main()
I (325) gpio: GPIO[21] InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (325) gpio: GPIO[22] InputEn: 1| OutputEn: 1| OpenDrain: 1| Pullup: 1| Pulldown: 0| Intr:0
I (405) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10023 Hum: 59%
I (2475) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10023 Hum: 59%
I (4245) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10022 Hum: 59%
I (6615) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10022 Hum: 59%
I (8685) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10022 Hum: 59%
I (10755) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10021 Hum: 59%
I (12825) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10021 Hum: 59%
I (14895) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10021 Hum: 59%
I (16965) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10020 Hum: 59%
I (19035) ESP: Header: x27 CMD: x9f Temp: 19 Press: 10020 Hum: 59%
```

Conclusiones y comentarios

El uso de las tareas en esta ocasión es muy sencillo, considerando que se tiene un solo cursor para escribir, se debe considerar que no se pueden crear tantas tareas que lo utilicen, pues para imprimir se requiere tiempo, y esta limitación de un cursor hace que sea más eficiente tener solo un par de tareas, y de ser posible, que solo una haga uso frecuente del cursor. Entonces la misma tarea que hace los cálculos de los recursos, funciona como render de esos mismos recursos. Por lo que siempre están sincronizados.

Dificultades de desarrollo

Las bibliotecas nuevas, con la documentación disponible no hicieron el trabajo de implementación sencillo, sin embargo después de extensiva revisión de los documentos y de el código fuente de las bibliotecas, fue posible completar la práctica, pero con diferentes problemas, ya que el comportamiento asíncrono de la biblioteca causaba comportamientos no esperados. Pero al final se logró.