



# Práctica 5

---

## Objetivo

El alumno se familiarizará con el periférico SPI usando el sistema embebido ESP32 DevKit v1 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.  
Visual Studio Code con ESP-IDF Development Framework.

## Teoría

Realice un mapa mental donde compare las características de I2C y SPI y muestre ejemplos donde es más conveniente usar un periférico que el otro.

## Desarrollo

1. Realice las modificaciones necesarias al código de la Práctica 4 de manera que la comunicación entre el ESP32 y el sensor sea por medio de **SPI**. Básicamente se deben modificar las siguientes dos funciones para la configuración del sensor y lectura de las mediciones con SPI.

```
/**
 * @brief Lee un registro del sensor
 */
esp_err_t device_register_read(uint8_t reg_addr, uint8_t *data,
size_t len)
```

donde:

**reg\_addr**: Es el registro a leer.

**data**: Buffer para almacenar los datos leídos.

**len**: Longitud de los datos a leer.

```
/**
 * @brief Escribe un byte a un registro del sensor
 */
esp_err_t device_register_write_byte(uint8_t reg_addr, uint8_t data)
```

donde:

**reg\_addr**: Es el registro a escribir.

**data**: Dato a escribir.

Similar que en la Práctica 4, en la función principal **app\_main**, hago uso de **device\_register\_write\_byte** para configurar el sensor en el modo de operación que permita el menor consumo de energía y la mayor resolución en el valor sensado.

Después de configurar el sensor, incluya un ciclo infinito donde cada tres segundos lea el registro de sanidad y los registros con la medición. Haga uso de **device\_register\_read** para leer los registros.

**Conclusiones y comentarios**  
**Dificultades en el desarrollo**  
**Referencias**