Sistemas Empotrados

Práctica 1: Adquisición y transmisión de datos

Objetivos:

El objetivo de esta práctica es conocer el entorno de desarrollo ESP-IDF proporcionado por Espressif para sus SoC de la familia ESP32 y llegar a realizar un sistema sencillo de adquisición y transmisión de datos. Se analizarán aspectos como:

1. Estructura de un proyecto en ESP-IDF.
2. Configuración del proyecto mediante menuconfig.
3. Adquisición de variables digitales y analógicas.
4. Transmisión de datos mediante Bluetooth.

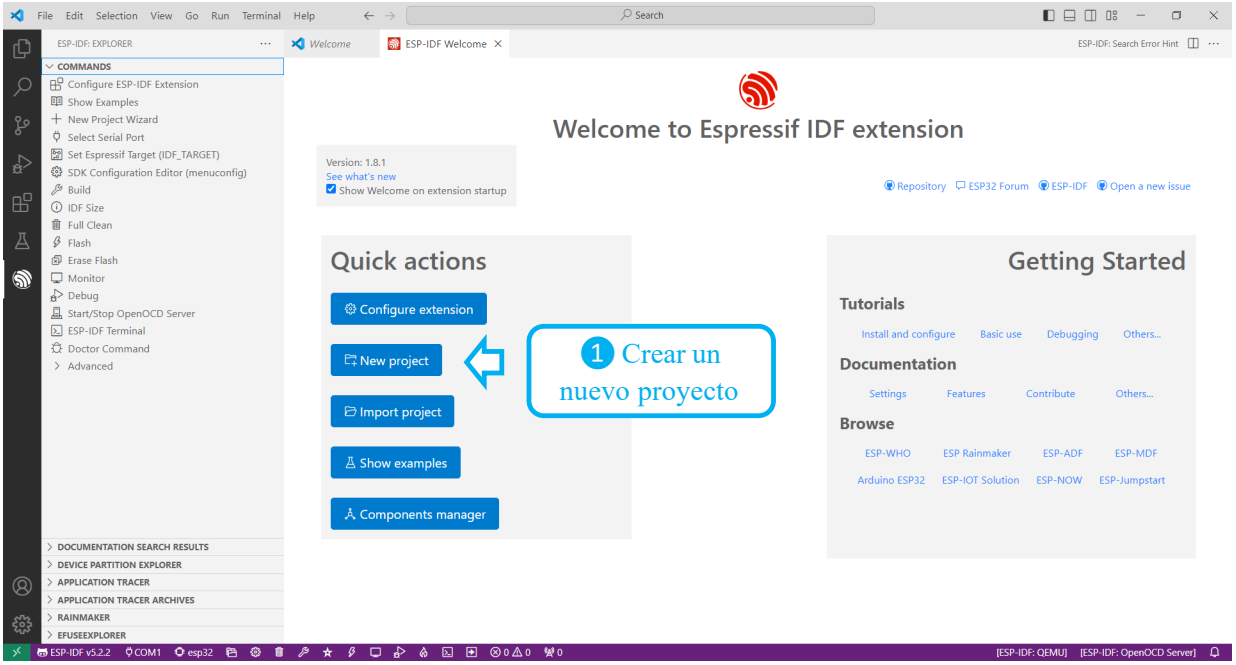
Recursos hardware:

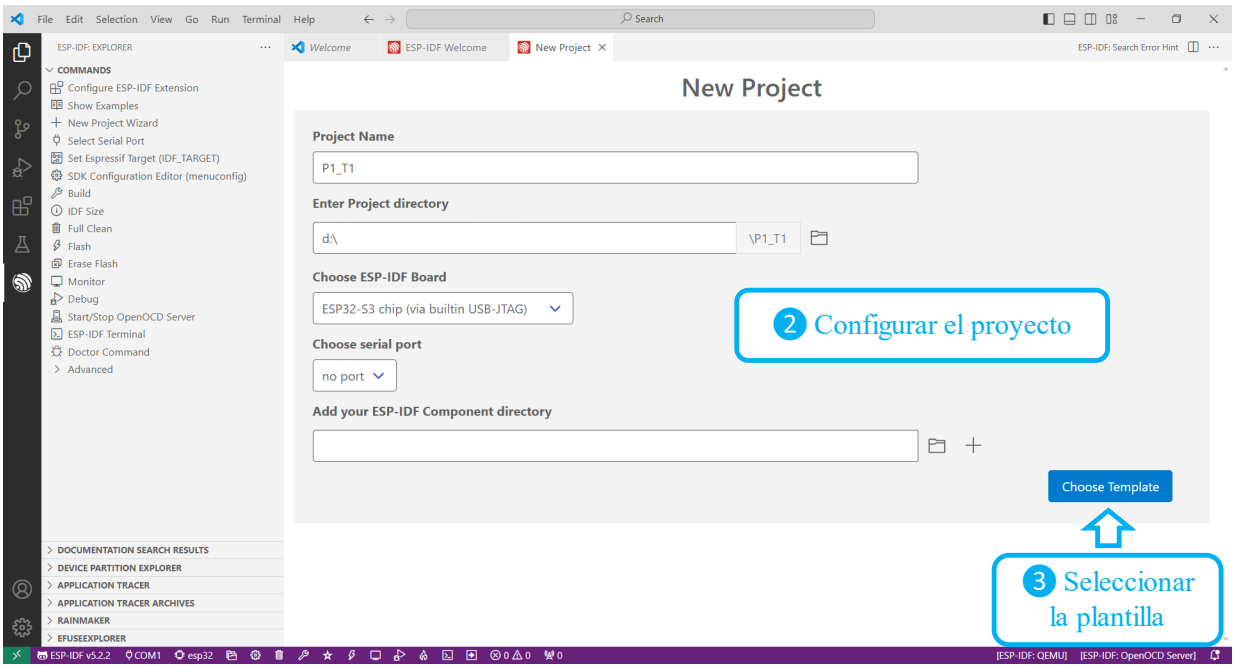
Para realizar la práctica se necesitarán los siguientes elementos:

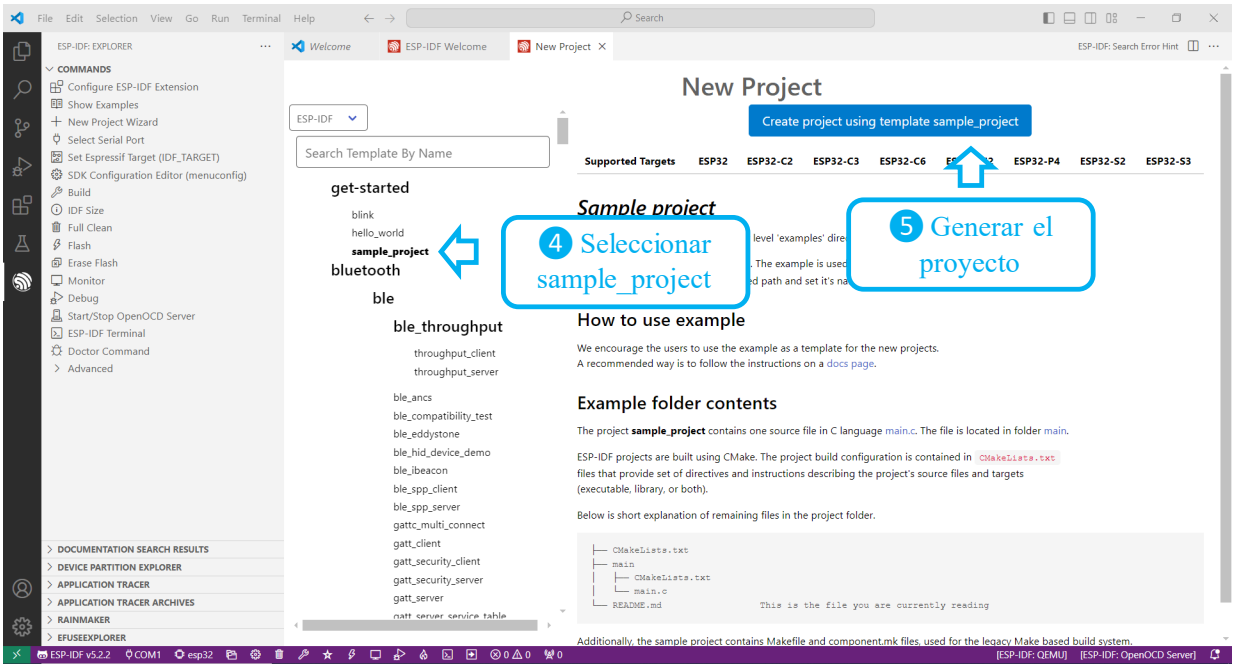
1. Placa de desarrollo Firebeetle 2 ESP32-S3
2. LDR
3. Resistencia de 10 kΩ.

Definición de un nuevo proyecto:

Para cada tarea de la práctica se debe crear un nuevo proyecto. En todos los casos se tomará como referencia la plantilla *sample_project*. Recuerde que los pasos para crear un nuevo proyecto son:







Tarea 1:

- Esta actividad se realiza de forma guiada por el profesor/a. **No es necesario entregar esta tarea.**
- Se analiza:
  - Cómo crear un proyecto.
  - Cuál es la estructura de un proyecto.
  - Dónde se ubican las librearías de ESP_IDF y cómo incluirlas en al proyecto.
- Se trabaja con la librería GPIO y las funciones básicas que permiten:
  - Configurar un terminal de E/S digital: *gpio_reset_pin()* y *gpio_set_direction()*.
  - Leer un terminal de E/S digital: *gpio_get_level()*.
  - Escribir en un terminal de E/S digital: *gpio_set_level()*.

Diseñar un programa que realice las siguientes funciones:

- Lee el estado de un pulsador.
- Si el pulsador está presionado enciende un LED y en caso contrario lo apaga.
- Imprime en pantalla (modo TERMINAL de ESP-IDF) el estado del LED: "LED encendido" / "LED apagado".
- Repite las acciones anteriores cada 2 segundos.

No será necesario conectar ningún elemento externo a la placa, sino que se utilizarán el pulsador y el LED disponibles en la misma.

Antes de iniciar la programación, analice la documentación general de la placa y su esquema de conexiones, y determine a qué terminales (GPIOs) del ESP32 están conectadas ambos dispositivos:

LED → GPIO___ y PULSADOR → GPIO___

Tarea 2:

- Esta actividad se realiza de forma guiada por el profesor/a. **No es necesario entregar esta tarea.**
- Se trabaja con la librería ADC y las funciones básicas que permiten:
  - Configurar un terminal de entrada analógico: *adc1_config_channel_atten()* y *adc1_config_width()*.
  - Leer un terminal de entrada analógico: *adc1_get_raw ()*.

Esta tarea se corresponde con una de los ejercicios realizados en la práctica 1 de la materia *Dispositivos IoT*. Se considera que el alumno sabe cómo conectar la LDR y su resistencia de acondicionamiento. Para cualquier duda se deben remitir a la documentación de dicha práctica.

En este caso, la LDR se conectará al terminar 8 (GPIO8) de la placa FireBeetle2, que corresponde a la entrada analógica A3 (canal 7 del ADC1). Antes de iniciar las tareas de programación verifique estos datos en la documentación de la placa.

Diseñar un programa que realice las siguientes funciones:

- Lee el valor de tensión de la LDR y lo imprime en pantalla (modo TERMINAL de ESP-IDF).
- Repite la acción anterior cada 2 segundos.

Tarea 3:

- Esta actividad la realiza el alumno/a de forma autónoma. **Es necesario entregar esta tarea.**

A partir del proyecto de la tarea 2, diseñar un nuevo proyecto que realice las siguientes funciones:

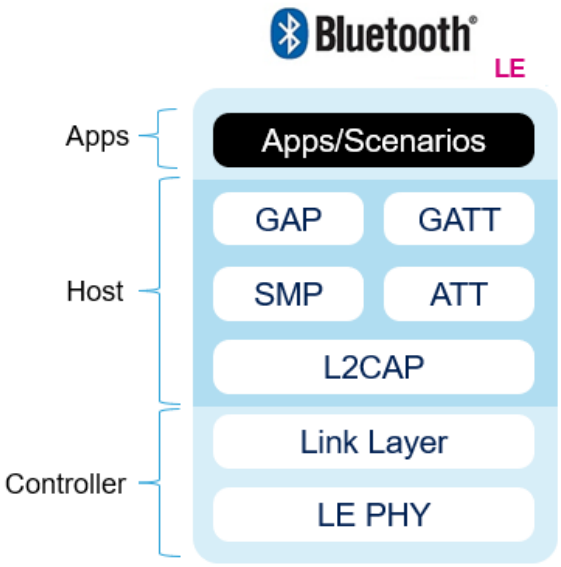
- Lee el valor de tensión de la LDR y lo imprime en pantalla (modo TERMINAL de ESP-IDF).
- Si dicho valor es superior a un determinado umbral (el valor de umbral se decide durante la práctica en función del nivel de luz ambiente) se debe encender el LED incorporado en la placa ESP32 y en caso contrario se debe apagar.
- Repite las acciones anteriores cada 2 segundos.

Tarea 4:

- Esta actividad se realiza de forma guiada por el profesor/a. **No es necesario entregar esta tarea.**
- Se analiza cómo añadir librerías de terceros a un diseño propio mediante el manejo de los ficheros CMakeLists.txt
- Se realiza una aplicación básica de comunicación por Bluetooth (BLE – Bluetooth Low Energy) configurando el ESP32 en modo servidor GATT. No se pretende estudiar la tecnología Bluetooth, sino sólo disponer de una plataforma genérica para el envío de datos a una app en un dispositivo móvil. Todo lo referente a las tecnologías de comunicación inalámbricas se estudia en la materia *Redes de Comunicaciones en IoT*.
- Para esta tarea se ha tomado como base el diseño propuesto por Vedat Ozan Oner (disponible en https://github.com/PacktPublishing/Internet-of-Things-with-ESP32/tree/main/ch8/gatt_server_ex) y se han realizado los cambios necesarios para adaptarlo a nuestra aplicación.

Conceptos básicos de BLE:

La arquitectura básica del BLE corresponde con el siguiente esquema:



Se divide en dos secciones principales: controlador y host. El controlador administra la radio y el host implementa la pila (capas de protocolo) que proporciona la interfaz para las aplicaciones. ESP32 utiliza Bluedroid como host Bluetooth predeterminado. Bluedroid es una implementación de código abierto del estándar Bluetooth para dispositivos Android importados a ESP-IDF.

En el nivel más alto del host se encuentran las capas GAP y GATT. La primera, GAP (Generic Access Profile), permite que un dispositivo sea visible para otros y determina el modo en el que puede interactuar un dispositivo con otro. Los modos pueden ser: sin conexión (*connectionless*), u orientado a la conexión (*connection-oriented*). En el primer caso sólo puede realizar tareas de difusión (*broadcaster*) o búsqueda (*observer*) de mensajes y alertas, mientras que en el segundo puede enviar datos (*peripheral*) o solicitarlos (*central*).

Una vez establecida la comunicación entre dos dispositivos (tarea controlada por la GAT), la capa GATT (Generic Attribute Profile) define cómo se realiza la transferencia de la información. Es decir, define la interfaz lógica con los clientes mediante servicios y características. Un servicio puede contener una o más características.

Respecto al Attribute Profile (ATT), describe las características básicas de un dispositivo BLE y define dos roles: servidor y cliente. Todas las características de un dispositivo BLE se describen como una lista de atributos. Cada atributo tiene los siguientes campos:

• Identificador (*Handle*): el identificador único del atributo
• Tipo (Type): el tipo de datos del atributo
• Valor (*Value*): el valor del atributo
• Permisos (*Permissions*): permisos de lectura y escritura para el campo Valor

Finalmente, el Security Manager Protocol (SMP) administra todas las operaciones de emparejamiento, autenticación y generación de claves.

ESP-IDF dispone de una API que incluye todas las funciones necesarias para configurar el módulo Bluetooth en las diferentes configuraciones que soporta.

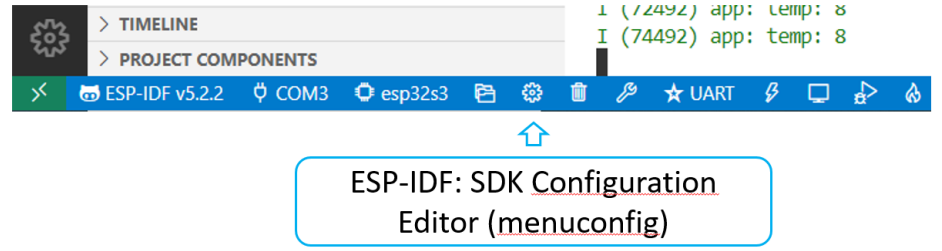
En esta tarea se pide diseñar un programa que realice las siguientes funciones:

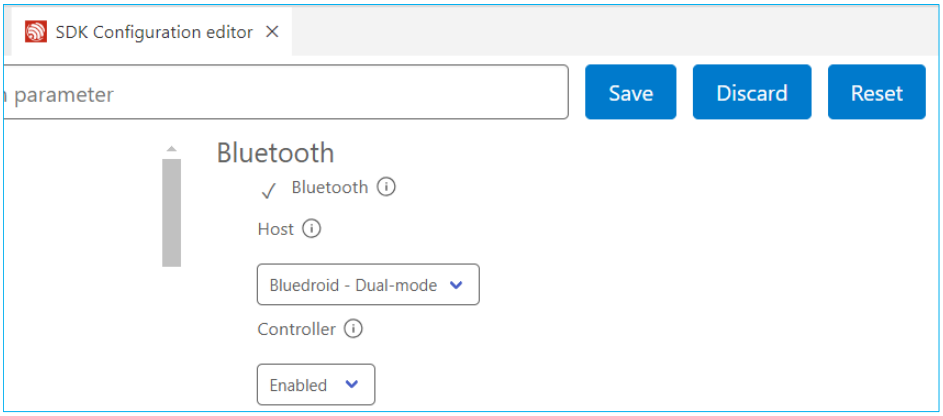
- Configurar el módulo Bluetooth en modo servidor.
- Iniciar una comunicación.
- Cada 2 segundos:
  - Enviar el valor decimal '8' por Bluetooth.
  - Escribir en pantalla un mensaje con dicho valor (modo TERMINAL de ESP-IDF).

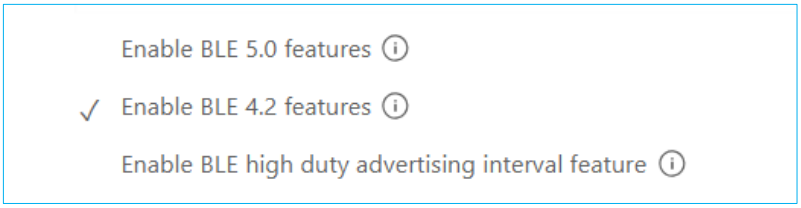
Para verificar el funcionamiento se utilizará la app *nRF Connect* de Nordic Semiconductor.

Para llevar a cabo la tarea 4 se deben realizar las siguientes acciones:

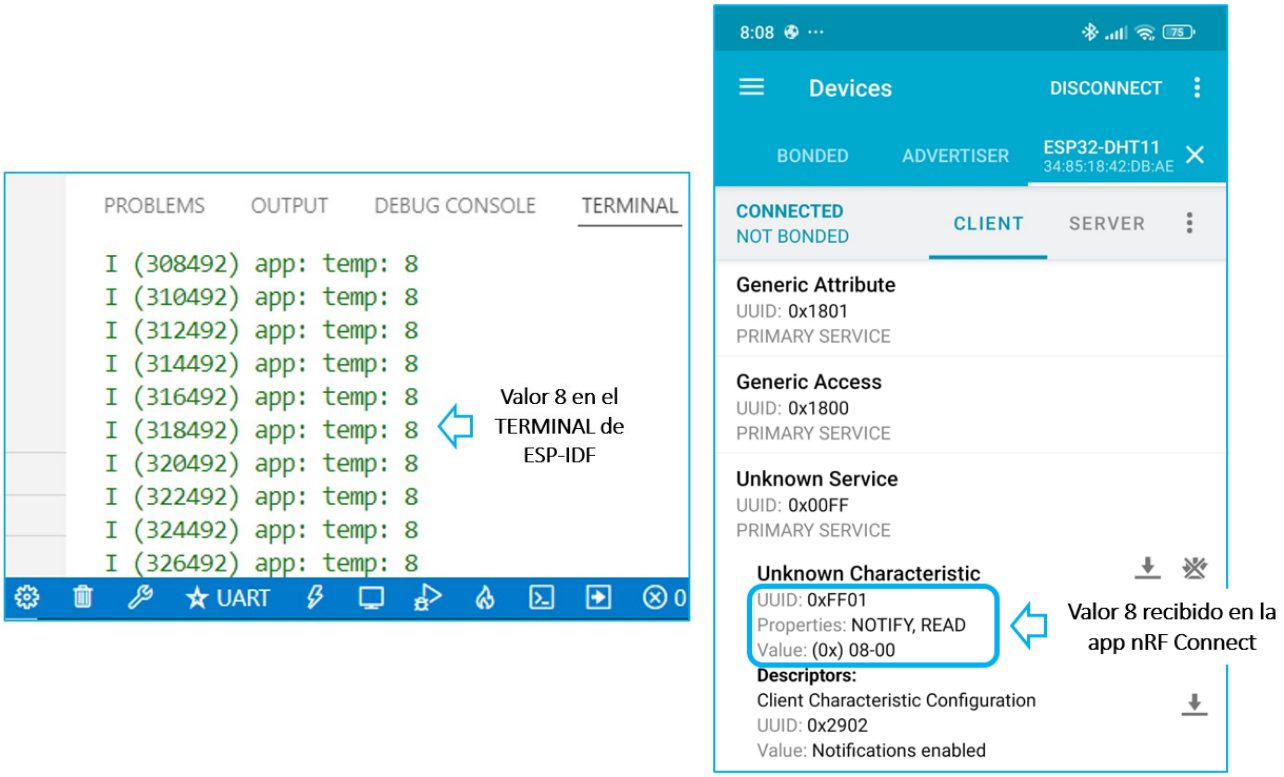
1. Crear un nuevo proyecto.
2. Copiar en la carpeta *main* del proyecto los archivos disponibles en Práctica 1 → Recursos → Tarea 4 (website de la asignatura). Los archivos que se deben copiar son app.c, app.h y main.c. Este último sustituye al que crea el proyecto por defecto.
3. Modificar el fichero *CMakeLists.txt* de la carpeta *main* para incluir app.c en la lista de ficheros a compilar. De esta forma queda correctamente configurado el componente *main*.
4. Mediante la interfaz de edición de configuración (menuconfig) habilitar el Bluetooth y escoger la opción BLE 4.2.







5. Compilar el programa y programar el ESP32.
6. Verificar la recepción del dato en pantalla (opción TERMINAL)
7. Verificar el funcionamiento de Bluetooth mediante la app nRF Connect.



Tarea 5:

- Esta actividad la realiza el alumno/a de forma autónoma. Es necesario entregar esta tarea.

A partir del proyecto de la tarea 4, diseñar un nuevo proyecto que realice las siguientes funciones:

- Configura e inicia una conexión Bluetooth.
- Cada 2 segundos lee el valor de tensión de la LDR, lo imprime en pantalla (modo TERMINAL de ESP-IDF) y lo envía por Bluetooth.