

Baliza Autónoma de Prevención y Alerta de Incendios (B.A.P.A.I)

Identificador de grupo: SBC25M10

Sitio web: <https://upm365.sharepoint.com/sites/SBC25M10>

Repositorio de código en Github:

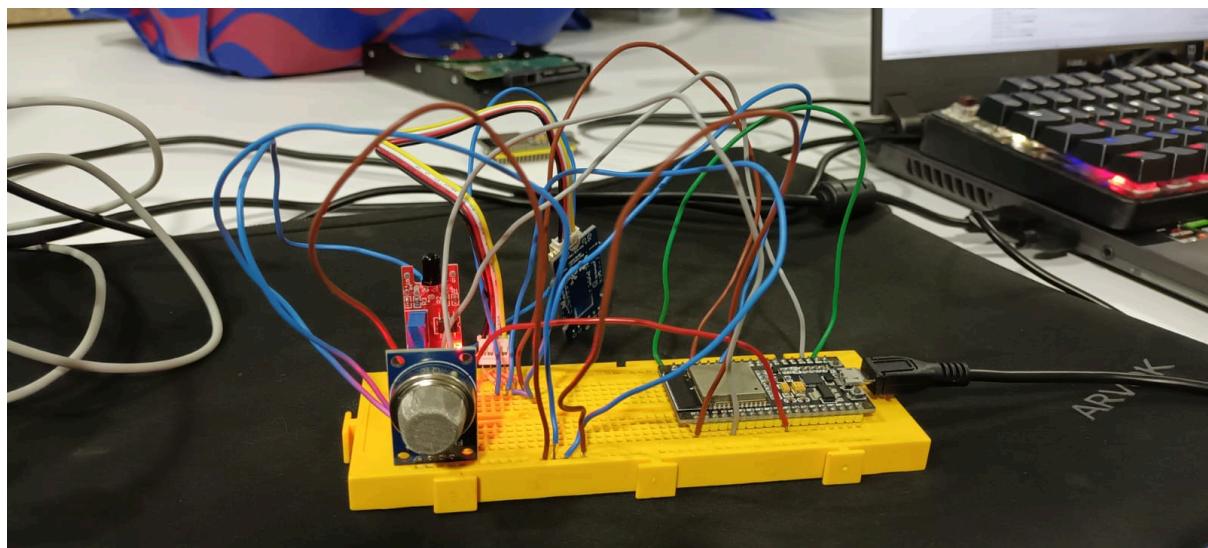
https://github.com/Valent1nus/Proyectos_Universidad/tree/main/practica-sistemas-basados-en-computador/BAPAI

Video demostrativo: [VID_20251215_180331.mp4](#)

Miembros del grupo / email:

1. GUILLERMO LUQUE HERNANDEZ / guillermo.luque@alumnos.upm.es
2. HUGO GARCIA MARTIN / hugo.garmartin@alumnos.upm.es
3. CARLOS GOMEZ DE ENTERRIA LOPEZ / c.gomezdeenterria@alumnos.upm.es
4. VALENTIN MANUEL EXPÓSITO SONO / valentin.exposito@alumnos.upm.es

Foto del sistema:



Organización y planificación

Meta

Nuestra meta era diseñar una baliza autónoma que transmite telemetría sobre las condiciones del ambiente para anticipar y detectar incendios. Está orientado hacia entidades como guardas forestales o bomberos.

Objetivos

Crear un dispositivo:

- De bajo coste
- Autonomo
- Escalable
- Adaptable

Además de utilizar todos los conocimientos aprendidos en clase posibles.

Prioridad

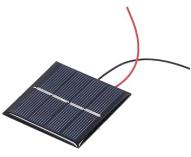
1. Utilizar los sensores propuestos en el anteproyecto
2. Implementar envío de telemetría
3. Implementar actualizaciones OTA
4. Conseguir autonomía mediante placas fotovoltaicas
5. Carcasa 3D

Actividades

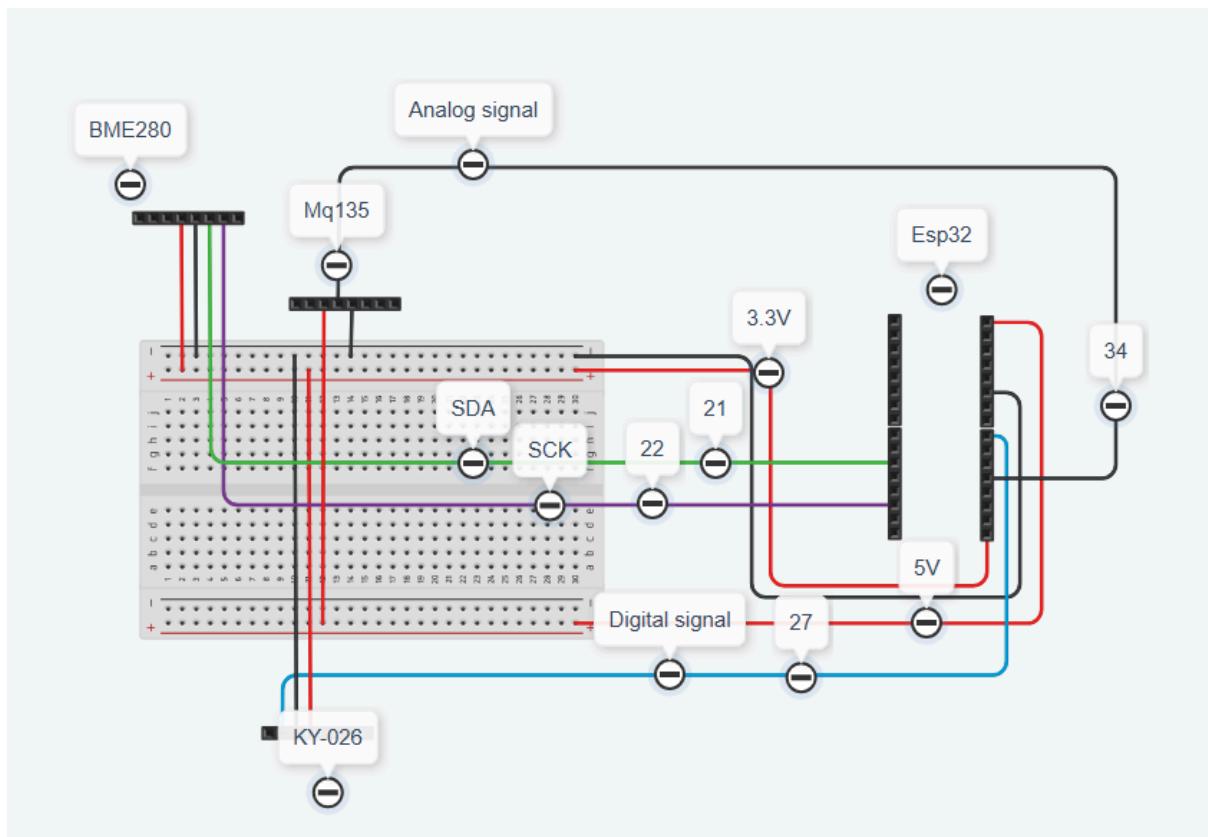
1. Asimilación de funcionamiento de cada sensor y calibración del mismo
2. Despliegue de un dispositivo y tablero en Thingsboard, así como de un bot de telegram y la rule chain correspondiente a su comportamiento
3. Implementación del servicio de actualizaciones OTA de Thingsboard.
4. Soldar y hacer pruebas de rendimiento con las placas fotovoltaicas, así como su compatibilidad con la powerbank
5. Diseño y modelado de la carcasa en 3D del dispositivo

Listado de componentes hardware

En la siguiente tabla mostramos los sensores, actuadores, microcontroladores y capacitores utilizados en el desarrollo de la práctica.

Nombre	Imagen	Descripción	Datasheet
ESP32		Microcontrolador de bajo consumo con conectividad Wi-Fi y Bluetooth integrada. Dispone de doble núcleo, múltiples GPIO, ADC, DAC, UART, SPI e I ² C	Datasheet
BME - 280		Sensor ambiental digital que mide temperatura, humedad relativa y presión atmosférica. Utiliza comunicación I ² C o SPI y se caracteriza por su alta precisión y bajo consumo	Datasheet
MQ - 135		Sensor analógico de calidad del aire capaz de detectar gases como CO ₂ , NH ₃ , NOx, alcohol, benceno y humo	Datasheet
KY - 026		Sensor infrarrojo diseñado para la detección de llamas o fuego. Proporciona salida analógica y digital	Datasheet
Batería portátil Varta-PowerPack Type 57960		Power Bank de 5V de salida y 1,0 A. Almacenaje de 6000 mAh	Inventario
Paneles solares		Dispositivo fotovoltaico que convierte la radiación solar en energía eléctrica. Se utilizan como fuente de energía renovable	Inventario

Esquema electrónico



Hemos usado tinkercad para realizar el esquema electrónico.

Los cables negros simbolizan tierra así como los rojos VCC, siendo el del inferior de la motherboard de 5V y el de la parte superior de 3,3V y hemos tratado desibolizar todos los pines a los que es relevante la conexión de la ESP así como cada entrada en los sensores, (Digital signal, SDA, SCK y Analog signal).

Diagrama de flujo del programa

Gráfico: [Anexo 1: Diagrama de flujo BAPAI](#)

- **Inicializa NVS** lee los valores relevantes de wifi (SSID y contraseña)
 - o si no hay valores o son incorrectos levanta un acces point
 - o Si los valores existen y son correctos se conecta a la red wifi.
- **Acces point** levanta un punto de acceso con una página web en la primera IP disponible de su wifi en la que se encuentra un cuestionario para introducir las credenciales del wifi. Una vez introducidas la ESP32 comprueba si son correctas .
 - o De ser correctas guarda las credenciales en NVS e inicia el programa
 - o De ser incorrectas el Acces point seguirá levantado.
- **Comprueba la versión actual y la remota(OTA)**. Realiza una petición http a la API de Things Board de cual es la última versión disponible del código y la compara contra la versión actual que tiene instalada.
 - o Si es la misma version el codigo continuará con normalidad
 - o Si la versión remota es más reciente, se lanza la OTA y se descarga dicha versión.
- **Configura ADC6** (conectado a un sensor de calidad de aire en el pin 34), el **SDA y el SCL** (conectado a un sensor de humedad temperatura y presión en los pines 21 y 22) y **GPIO16** como entrada digital (conectado a un sensor de llama en el pin 27).
- **Se suscribe por MQTT** a un *topic* especificado en Things Board utilizando el *token* de dispositivo correspondiente, y se inicia un bucle infinito donde se leen y publican datos.
- En cada iteración del bucle infinito:
 - **Lee Bme280 (humedad temperatura y presión)**. Recoge los datos en un JSON, para poder enviarlos posteriormente por mqtt.
 - **Lee mq135 (Calidad de aire)**. Recoge los datos en un JSON, para poder enviarlos posteriormente por mqtt.
 - **Lee Pir (Sensor de llama)**. Recoge los datos en un JSON, para poder enviarlos posteriormente por mqtt.
 - **Envía los datos** por mqtt.
 - **Comprueba** si han pasado 24h la última versión del código en thingsboard.

Diagrama de dependencias

El firmware del sistema B.A.P.A.I se apoya en distintas librerías y componentes del ESP-IDF, que permiten gestionar la conectividad, la adquisición de datos de sensores y la actualización remota del código.

COMPONENTES	LINK	DESCRIPCIÓN
BME280		Librería para comunicación I2C. Se utiliza para obtener medidas de temperatura, humedad y presión que se envían a ThingsBoard vía MQTT.
AP	GIT	Basado en el ejemplo “softap_sta” de ESP-IDF. Permite crear un punto de acceso para configurar credenciales WiFi cuando no existen y luego conectarse como estación.
OTA	GIT	Implementado mediante el ejemplo “simple_ota_example”. Permite al sistema verificar si existe una versión más reciente del firmware en ThingsBoard e instalarla automáticamente.
ONESHOT_READ	GIT	Para la lectura del sensor MQ-135 se utiliza el módulo “oneshot_read” del ESP-IDF, que permite realizar lecturas puntuales del ADC. La configuración se realiza una única vez y las lecturas se ejecutan dentro del bucle principal, optimizando el consumo energético.

Paneles de datos e interacción

- Pantalla de comprobación de conectividad HTTP en ThingsBoard usando un access token: [Anexo 2](#).
- Pantalla de configuración de punto de acceso WiFi desde un móvil, mostrando red, contraseña y banda activa para el softAP: [Anexo 3](#).
- Interfaz web de configuración WiFi del ESP32 mediante modo SoftAP: [Anexo 4](#).
- Dashboard IoT con visualización de sensores (temperatura, humedad, presión, CO₂ y PIR) y estado de alarmas: [Anexo 5](#).
- Flujo de trabajo lógico (rule chain) y alertas de Telegram: [Anexo 6](#).
- Configuración de suscriptores en el canal de Telegram: [Anexo 7](#).
- Chat de Telegram recibiendo múltiples notificaciones automáticas de nuestro proyecto: [Anexo 8](#).
- Repositorio con versiones de firmware OTA, incluyendo binarios y control de versiones: [Anexo 9](#).