

SISTEMAS EMBEBIDOS

PRÁCTICA 1: Entorno y simulación

Objetivos

- Familiarizarse con el flujo de trabajo en Sistemas Embebidos (Edit -> Compile -> Flash -> Monitor).
- Configurar el entorno de desarrollo local (VSCode + PlatformIO/ESP-IDF).
- Utilizar simuladores online (Wokwi) para prototipado rápido.
- Obtener datos mediante el primer programa en hardware real ESP32.

Descripción

En esta práctica utilizaremos placas de desarrollo basadas en el ecosistema ESP32, una plataforma potente y versátil diseñada específicamente para el Internet de las Cosas (IoT). Dependiendo de la configuración de tu laboratorio, utilizaremos uno de los siguientes modelos:

Hardware Principal

Característica	ESP32-DevKitV1	ESP32-C3
Arquitectura	Xtensa® Dual-Core LX6	RISC-V Single-Core
Rendimiento	Hasta 240 MHz	Hasta 160 MHz
Conectividad	WiFi + Bluetooth (Classic/BLE)	WiFi + Bluetooth 5 (LE)
Enfoque	Propósito general y potencia	Bajo consumo y seguridad

A diferencia de los microcontroladores tradicionales, estos dispositivos permiten una gestión avanzada de la conectividad sin sacrificar la simplicidad. Aunque internamente pueden ejecutar sistemas operativos en tiempo real (como FreeRTOS), para esta práctica se programarán de forma directa mediante el entorno de Arduino, facilitando la carga de código y la respuesta inmediata del hardware.

Para la adquisición y visualización de datos, el sistema se integra con los siguientes componentes:

- **Sensor de Pulsaciones MAX30102:** Es un sensor óptico que integra un oxímetro de pulso y un monitor de frecuencia cardíaca. Utiliza tecnología de detección mediante LEDs e infrarrojos para medir la reflectancia en la piel. Al analizar cómo la luz atraviesa el tejido sanguíneo, el sensor permite calcular:
 - La saturación de oxígeno en sangre (**SpO2**).
 - La frecuencia cardíaca (pulsaciones por minuto).
- **Pantalla OLED de 0.91 pulgadas:** Para la interfaz de usuario, utilizaremos una pantalla compacta con tecnología OLED, que ofrece un alto contraste y bajo consumo de energía. Se comunica habitualmente mediante el protocolo **I2C**, lo que permite una conexión sencilla utilizando solo dos cables de datos (SDA y SCL) hacia nuestro ESP32.

- **Led RGB:** para visualizar información de forma rápida.

Parte A: Simulación con Wokwi

- Configuración del Escenario:**
 - Accede a [Wokwi.com](#) y selecciona un proyecto de **ESP32** (o ESP32-C3).
 - Añade los componentes: un LED externo (con resistencia de 220Ω), un sensor **MAX30105** (Utilizaremos un potenciómetro) y una pantalla **OLED SSD1306**.
- Prueba de Concepto (Blink):**
 - Programa un parpadeo en el pin GPIO asignado.
 - Modifica el código para crear una secuencia de 5 parpadeos rápidos (100ms) seguida de una pausa de 1 segundo.
- Lógica del Sensor y Pantalla:**
 - Utiliza el Monitor Serial para visualizar datos simulados del sensor.
 - Diseña la interfaz en la pantalla OLED para mostrar el mensaje "Iniciando..." y luego los valores de "BPM" (Pulsaciones por minuto).
- Integración Virtual:**
 - Utiliza el potenciómetro para que simule las pulsaciones que se muestran por pantalla.

```
int lectura = analogRead(PIN); int bpm = map(lectura, 0, 4095, 40, 180);
```

Parte B: Instalación del Entorno y Hardware Real

1. Preparación del Entorno Local

- **IDE de Arduino:** Descarga e instala la última versión de [Arduino IDE](#).
- O Utiliza VS Code con la extensión PlatformIO
- **Gestor de Tarjetas:** * Ve a Preferencias y añade la URL de gestión de ESP32 de Espressif.
 - Desde el Board Manager, busca e instala el paquete **esp32** para habilitar el **ESP32-C3** y el **ESP32 Dev Module**.
- **Drivers:** Si el ordenador no reconoce la placa, instala el driver **CP210x** o **CH340** (dependiendo del chip de comunicación de tu placa).

2. Gestión de Librerías

Instala desde el Gestor de Librerías del IDE:

1. [SparkFun MAX3010x Pulse and Proximity Sensor Library](#)
2. [Adafruit SSD1306 y Adafruit GFX Library](#)

3. Montaje Físico e Interconexión

Al haber utilizado un potenciómetro en el simulador, nuestro montaje real no coincide con el simulado. Antes de proceder con el montaje, añade en Wokwi un custom chip con el mismo número de pines que tu sensor físico.

Reemplaza el potenciómetro por este custom chip y guarda tu esquema para la entrega. Estos dispositivos (sensor y pantalla) comparten el bus de comunicación **I2C**. Consulta el pinout de tu MCU para encontrar los pines SDA y SCL.

4. Flujo de Trabajo en Hardware

- **Paso 1: Diagnóstico I2C.** Carga un código "I2C Scanner" para verificar que la placa detecta las direcciones hexadecimales del sensor y la pantalla.
- **Paso 2: Probando el Sensor.** Carga el ejemplo de la librería SparkFun para leer el sensor MAX30105. Abre el **Serial Plotter** para observar la onda de pulso en tiempo real.
- **Paso 2: Probando el Sensor.** Carga el ejemplo de la librería para la prueba de la pantalla y observa el resultado. Borra todo lo referente de la prueba o carga el ejemplo vacío y muestra el mensaje BPM = 70.
- **Paso 3: Feedback Visual.** En un nuevo script, programa el LED externo para que realice ciclos como si fuese un semáforo, 5 segundos rojo, 3 segundos verde y 1 segundo de parpadeos amarillos de 0,1. Crea un nuevo script que realice un "destello" coincidiendo con cada latido detectado por el sensor.
- **Paso 4: Despliegue Final.** Integra el código para que la pantalla OLED muestre el valor numérico de las pulsaciones (BPM). Si no se detecta el dedo, muestra por pantalla falta dedo. Añade el destello de pulsaciones siguiendo esté comportamiento de colores
 - < de 70 pulsaciones = color Verde
 - <= 70 y <110 = color Amarillo
 - <=110 = color Rojo

Importante: Al trabajar con el **ESP32-C3**, asegúrate de seleccionar la placa correcta en tu IDE, ya que su arquitectura **RISC-V** difiere del ESP32 estándar. Ambos operan a **3.3V**, por lo que se debe tener especial cuidado de no alimentar los sensores con voltajes superiores para evitar daños permanentes.

Normas de entrega:

- La realización del trabajo es individual.
- La fecha límite de entrega, **4 de marzo de 2026, antes de 23:55**.
- La entrega se realizará a través de la herramienta de entrega de trabajos de Moodle.
- Se debe entregar un PDF con el informe escrito por el alumno, junto con el código de los diferentes scripts solicitados. Añade capturas, fotos y comentarios de tu código, prototipo físico y esquemas creados.