OSHW Screening Task: Build and Demonstrate ESP32 Emulation from Scratch using QEMU

Objective

Your task is to:

1. **Set up a complete ESP32 emulation environment** using **QEMU** and **ESP-IDF**.
2. **Demonstrate two working examples** running on the emulator:
    - **Blink an LED**
    - **Read and print temperature (simulated)**
3. Document your steps and learnings clearly — this will help you (and others) contribute to building an **automated ESP32 code evaluation platform** integrated with **Yaksh**.

Background

ESP32 is a microcontroller commonly used for IoT projects.
In this project, you will **not use real hardware** — instead, you'll emulate the ESP32 using **QEMU (Quick Emulator)**, an open-source virtualization platform.

This work lays the foundation for creating a **virtual testing environment** where student code submissions can be compiled, run, and evaluated automatically inside Yaksh.

Deliverables

By the end of this task, you must:

1. Successfully set up the QEMU-based ESP32 Emulator environment locally.
2. Build and run two example applications:
    - LED Blink

    - Temperature Reading (or sensor simulation)

3. Submit a **Markdown report** documenting the full process, with screenshots and explanations.

Task Steps

Step 1: Understand the Goal

Read about what QEMU does and why it's used in embedded systems development.
Reference:

- QEMU Official Documentation
- Espressif Blog – Emulating ESP32 on QEMU
- Yaksh: Python-based Online Evaluation Platform

Step 2: Install Prerequisites

You'll need:

- Linux or WSL environment (Ubuntu 20.04 or later recommended)
- `git`, `python3`, `pip`, `cmake`, `make`, `gcc`, `g++`.
- Internet access for cloning repositories

Learn about setting up a development environment for ESP32:
ESP-IDF Get Started Guide

Step 3: Build QEMU for ESP32

You will compile QEMU from the Espressif fork that supports ESP32 emulation.

Reference:

- Espressif QEMU GitHub Repository
- Espressif QEMU Wiki and Build Instructions

Steps to learn:

- Clone the repository
- Configure for the `xtensa-softmmu` target
- Compile and test installation

Step 4: Set Up ESP-IDF

Install the **ESP-IDF SDK**, which provides the compiler, libraries, and tools for building ESP32 applications.

Reference:

- ESP-IDF Installation
- ESP-IDF Command Line Tools

After installation, verify using:

idf.py --version

Step 5: Create Your First ESP-IDF Project

Using the `idf.py` command-line tool, create a new project (e.g., `blink`) and explore its folder structure.

Learn from:

- ESP-IDF Example Projects
- ESP-IDF GPIO Guide

Step 6: Build and Run in QEMU

Compile your ESP32 project using `idf.py build`, then run it inside QEMU using:

qemu-system-xtensa -nographic -machine esp32 -drive file=build/<your_project>.elf,if=mtd,format=raw

You should see console logs that confirm your code is running.

Reference:

- QEMU ESP32 Command Line Options

Step 7: Demonstrate Two Programs

Once your environment works, demonstrate:

1. **Blink LED**
    - Emulate toggling a GPIO pin.
    - Print messages like “LED ON” / “LED OFF” to the console.
2. **Temperature Reading (Simulated)**
    - Simulate a temperature sensor using random or predefined values.
    - Print readings periodically on the console.

Take screenshots of both running on QEMU.

Step 8: Documentation and Submission

Create a file `report.md` including:

1. **System Information** — OS, QEMU version, ESP-IDF version
2. **Setup Steps** — detailed commands used
3. **Challenges & Fixes** — document any issues you faced
4. **Demonstration Screenshots** — both programs running
5. **Reflection** — how this setup can be used for automated code evaluation on Yaksh

Reference Links Summary

Topic	Link
QEMU Official	https://www.qemu.org/docs/master/
Espressif QEMU Repo	https://github.com/espressif/qemu
ESP-IDF Get Started	https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html
ESP-IDF GPIO Docs	https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference//gpio.html
Yaksh Platform	https://github.cperipheralsom/FOSSEE/online_test
Example ESP-IDF Projects	https://github.com/espressif/esp-idf/tree/master/examples/get-started