30-Day-FreeRTOS-Course-for-ESP32-Using-ESP-IDF

A complete 30-day hands-on course to master FreeRTOS on ESP32 with ESP-IDF. Learn task creation, synchronization, multicore design, debugging, and build real-world IoT projects. Perfect for embedded developers ready to level up with real-time systems.

📅 Week 1: Foundation – Getting Started with FreeRTOS

Day	Topic
Day 1	**Introduction to RTOS and FreeRTOS**
Understand what an RTOS is, why use FreeRTOS, and how it differs from bare-metal programming.	
Day 2	**Setting Up ESP-IDF for FreeRTOS Development**
Install and configure ESP-IDF, create your first ESP32 project.	
Day 3	**FreeRTOS Architecture on ESP32 (SMP)**
Learn how FreeRTOS runs on dual-core ESP32, including task scheduling and core affinity.	
Day 4	**Creating and Deleting Tasks**
Write your first task-based program. Learn how to create, delete, and manage tasks.	
Day 5	**Task States and Priorities**
Explore task lifecycle, priorities, and task switching with practical examples.	
Day 6	**Using vTaskDelay() and vTaskDelayUntil()**
Learn how FreeRTOS handles time and how to delay tasks accurately.	
Day 7	**Hands-on Mini Project: Blinking Two LEDs with Two Tasks**
Solidify task basics by building a multi-task LED application.	

📅 Week 2: Communication and Synchronization

Day	Topic
Day 8	**Queues – Theory and Practice** Use queues to send data between tasks.
Day 9	**Using Queues with ISR** Learn how to safely send data from ISRs using xQueueSendFromISR().
Day 10	**Binary Semaphores** Synchronize between ISR and tasks.
Day 11	**Counting Semaphores** Use them for managing shared resources like buffer pools.
Day 12	**Mutexes and Recursive Mutexes** Understand resource locking and avoid race conditions.
Day 13	**Avoiding Priority Inversion** Learn how FreeRTOS handles this issue with priority inheritance.
Day 14	**Hands-on Mini Project: UART Logger Task Using Mutex** Create a thread-safe UART logger shared by multiple tasks.

📆 Week 3: Advanced Concepts and Peripheral Integration

Day	Topic
Day 15	**Event Groups**
Create multi-flag synchronization logic using events.	
Day 16	**Task Notifications – Lightweight Alternatives to Semaphores**
Use notifications for fast ISR-to-task signaling.	
Day 17	**Stream Buffers and Message Buffers**
Transmit streams of data between tasks efficiently.	
Day 18	**Static vs Dynamic Memory Allocation**
Explore heap management (heap_4, heap_caps) and memory planning.	
Day 19	**Software Timers**
Set up periodic or one-shot timers for non-blocking operations.	
Day 20	**Using FreeRTOS with GPIO, UART, I2C, SPI**
Practical interfacing using tasks + semaphores.	
Day 21	**Mini Project: Sensor Polling with Notifications**
Create a periodic task to read a sensor and update a display.	

📆 Week 4: Optimization, Debugging, and Multicore Design

Day	Topic
Day 22	**Multicore Task Placement (Core Affinity)** Pin tasks to cores and design load-balanced systems.
Day 23	**FreeRTOS Hook Functions and Customization** Use Idle, Tick, and MallocFailed hooks for control.
Day 24	**Tickless Idle and Low Power FreeRTOS** Implement tickless idle for power savings in ESP32.
Day 25	**Handling Interrupts with FreeRTOS** Write ISR-friendly code and synchronize with tasks.
Day 26	**Stack Overflow and Watchdog Timers** Detect, handle, and prevent stack overflows.
Day 27	**Runtime Statistics and Trace Tools** Use ESP-IDF logging, system view, and tracealyzer.
Day 28	**Mini Project: Dual-Core Task Distribution (Wi-Fi + Display)** Design a system where networking and UI run on different cores.
Day 29	**Unit Testing FreeRTOS Code** Integrate Unity test framework to test task logic.
Day 30	**Final Project: Real-Time Sensor Hub with Display and Logger** Build a complete multitasking system to read multiple sensors, log data, and display it over UART or OLED.

30-Day-FreeRTOS-Course-for-ESP32-Using-ESP-IDF

📦 What You'll Need

- ESP32 DevKit (any variant)
- ESP-IDF installed and working
- USB-to-UART driver
- Terminal (minicom, PuTTY, or ESP-IDF Monitor)
- Optional peripherals: LEDs, DHT22, OLED, pushbuttons

🧠 Learning Outcome

By the end of this 30-day course, you'll:

- Understand and apply core FreeRTOS concepts
- Efficiently synchronize and manage concurrent tasks
- Interface ESP32 peripherals using tasks and ISRs
- Design real-time multitasking systems
- Debug and test RTOS-based embedded code