RTOS & MCU Fast-Track Project Roadmap

This roadmap is designed to help you build practical experience with **RTOS** and **microcontrollers (MCUs)** in under three weeks. It emphasizes hands-on projects, progressively building from bare-metal basics to a polished capstone demo. The final output includes working code repositories and clear documentation.

Week 1 – Setup & Bare Metal MCU Basics

Goal: Get comfortable with your hardware and toolchain.

Hardware Options (choose one): - ESP32 DevKit (built-in FreeRTOS, Wi-Fi/Bluetooth, very affordable) - STM32 Nucleo or Discovery board (ARM Cortex-M, widely used)

Tasks: 1. Install toolchain (ESP-IDF for ESP32 or STM32CubeIDE for STM32). 2. Run a **bare-metal LED blink** program. 3. Run a **UART Hello World** program.

Deliverables: - Repo folder: `week1_basics/` - `blinky/` – basic LED blink - `uart_hello/` – UART print - Document setup instructions in README.

Week 2 – Core RTOS Features

Goal: Learn FreeRTOS primitives (tasks, queues, semaphores, interrupts).

Daily Mini-Projects: 1. **Two-task LED blinker** – two LEDs at different frequencies. 2. **Button interrupt + task** – button press signals a task via queue. 3. **UART Logger Task** – multiple tasks send messages through a queue. 4. **Semaphore demo** – two tasks safely share a resource.

Deliverables: - Repo folder: `week2_rtos/` - `multitask_blinky/` - `button_queue/` - `uart_logger/` - `semaphore_demo/` - README explains each FreeRTOS feature used.

Week 3 – Capstone Project

Goal: Integrate peripherals and RTOS features into one demo.

Option A: ESP32 IoT Demo - Task A: Reads sensor data (I2C temperature sensor). - Task B: Publishes data via Wi-Fi/MQTT. - Task C: Heartbeat LED blink.

Option B: STM32 Sensor + LCD Demo - Task A: Reads temperature sensor via I2C. - Task B: Updates LCD once per second. - Task C: Logs readings via UART.

Deliverables: - Repo folder: `week3_capstone/` - `src/` – source code - `docs/` – wiring diagram + notes - `README.md` – overview, setup steps, usage

---

GitHub Repo Structure (Template)

```
rtos-mcu-fasttrack/
│
├── week1_basics/
│   ├── blinky/
│   └── uart_hello/
│
├── week2_rtos/
│   ├── multitask_blinky/
│   ├── button_queue/
│   ├── uart_logger/
│   └── semaphore_demo/
│
├── week3_capstone/
│   ├── src/
│   ├── docs/
│   │   └── wiring_diagram.png
│   └── README.md
│
└── README.md (top-level overview)
```

README Template (for each project)

```
# Project Title

## Overview
Short description of what the project does.

## Features
- Task A: description
- Task B: description
- Task C: description (if applicable)

## Hardware
- Board: ESP32 DevKit or STM32 Nucleo
- Peripherals: e.g., I2C temp sensor, LEDs, push button

## How to Build & Run
```

```
1. Clone repo
2. Open in IDE (ESP-IDF or STM32CubeIDE)
3. Flash to board

## Demo
(Add serial output sample, diagram, or photo)
```

Suggested Timeline (Aggressive, 3 Weeks)

- **Week 1:** Hardware setup + bare-metal blinky + UART hello.
- **Week 2:** One FreeRTOS mini-project per day (4 total).
- **Week 3:** Capstone project + documentation.

By the end of this roadmap, you will have: - **6–7 mini demos** showcasing RTOS concepts. - **1 polished capstone project** integrating MCU peripherals. - A **GitHub portfolio** with structured repos and clear READMEs.

This provides practical, demonstrable experience with RTOS and MCUs in less than three weeks.