

## Plano Analítico: Sistemas Embarcados

### 1. Identificação da Unidade Curricular

- **Curso:** Engenharia de Redes e Telecomunicações (ERT)
  - **Ano:** 3º | **Semestre:** 2º
  - **Créditos:** 8.0 UC
  - **Carga Horária Total:** 120 Horas
  - **Distribuição:**
    - **Teóricas (T):** 30h
    - **Teórico-Práticas (TP):** 30h
    - **Práticas/Laboratório (P):** 30h
    - **Trabalho Autónomo (TA):** 22h
    - **Orientação e Tutoria (OT):** 4h
    - **Avaliação (AV):** 4h
- 

### 1. Fundamentação

Os sistemas embarcados são computadores de função dedicada que operam dentro de outros dispositivos. No contexto de ERT, eles são a base dos terminais de rede, sensores inteligentes e sistemas de controlo remoto. Esta disciplina fundamenta-se na necessidade de otimizar código para recursos limitados (memória e energia) e garantir o tempo de resposta em tempo real, competências vitais para a implementação de redes de sensores (WSN) e infraestruturas de telecomunicações.

### 2. Objectivos Instrutivos e Educativos

- **Instrutivos:** Compreender a arquitetura de microcontroladores modernos (ARM, AVR ou ESP32); dominar o desenvolvimento de firmware em C/C++ e Assembly; aprender a interfacear sensores, atuadores e módulos de comunicação; compreender sistemas operativos de tempo real (RTOS).
- **Educativos:** Fomentar a precisão no desenvolvimento de hardware/software; desenvolver a capacidade de criar soluções tecnológicas compactas e eficientes; promover a consciência sobre o consumo energético em dispositivos móveis.

### 3. Resultado de Aprendizagem

O estudante será capaz de:

- Projetar e programar sistemas baseados em microcontroladores para tarefas específicas de rede.
- Implementar protocolos de comunicação série (I2C, SPI, UART) para integração de componentes.
- Utilizar interrupções e temporizadores (Timers) para controlo preciso de processos.
- Desenvolver aplicações de IoT (Internet das Coisas) com conectividade sem fios (Wi-Fi/Bluetooth).
- Realizar o depuramento (*debugging*) de hardware utilizando osciloscópios e analisadores lógicos.

### 4. Planeamento Temático (8 UC)

Tema	Horas (T+TP+P)	Conteúdo Programático
<b>I. Arquitetura de Microcontroladores</b>	20h	Organização interna (CPU, Memória, I/O); Ciclo de instrução; Famílias de processadores (foco em ARM Cortex-M).
<b>II. Periféricos e Interfaces</b>	25h	GPIO (Entradas/Saídas); Conversores ADC/DAC; PWM; Protocolos série (UART, SPI, I2C).
<b>III. Programação de Baixo Nível</b>	25h	Gestão de Interrupções (ISR); Timers e Counters; Modos de baixo consumo (Sleep modes); Watchdog Timer.
<b>IV. Conectividade e Protocolos</b>	25h	Pilhas de protocolo para sistemas embebidos; Wi-Fi, Bluetooth/BLE e LoRa; Integração com Cloud/MQTT.
<b>V. RTOS e Sistemas Avançados</b>	25h	Introdução aos Sistemas Operativos de Tempo Real (FreeRTOS); Escalonamento de tarefas; Semáforos e Filas.

### 5. Recomendações Metodológicas

- **Laboratório de Projetos (30h):** Uso intensivo de plataformas como **Arduino, ESP32 ou STM32**. Os alunos devem montar circuitos reais em protoboard e placas de desenvolvimento.
- **Ferramentas de Desenvolvimento:** Utilização de IDEs profissionais como **STM32CubeIDE, VS Code com PlatformIO ou Keil uVision**.
- **Projeto de IoT Integrado:** Criação de um dispositivo que recolha dados de sensores e os envie para um servidor web através de um protocolo leve (MQTT).

## 6. Sistema de Avaliação

Conforme a alocação de **4h para AV:**

- **Avaliação Contínua (50%):** Projetos práticos intermédios (ex: controlo de temperatura ou interface de teclado) (20%) e um Projeto Final Integrador (30%).
- **Avaliação Formal (50%):** Exame escrito focado na arquitetura, programação e temporização de sistemas embebidos.

## 7. Bibliografia Principal Indicada

1. **MAZIDI, Muhammad Ali.** *The STM32 Arm Cortex-M Assembly Programming and Interface.*
2. **WOLF, Marilyn.** *Computers as Components: Principles of Embedded Computing System Design.* Morgan Kaufmann.
3. **VALVANO, Jonathan.** *Embedded Systems: Introduction to Arm Cortex-M Microcontrollers.*