

Embedded Firmware Assignment

STM32L0 — Bare-metal

Overview

This short assignment (target: 1–2 days) evaluates core embedded skills: professional project setup, portable software structure, interrupt-driven sensor sampling, I²C communication, and DAC control. The hardware schematics have changed: the previous nRF5340 + Zephyr setup is replaced by the STM32L0 MCU and the system should be reimplemented as a bare-metal firmware (using vendor drivers such as STM32Cube HAL or LL is allowed). External DACs were removed — candidates should use the MCU’s internal DAC(s) as shown in the provided schematics.

Important: Use the provided schematic to determine exact pin assignments and power/voltage rails. Document any assumptions you make about the schematic in the repo README.

Task 1. Create a new git repository (public or private — provide access). Use meaningful, focused commits.

Task 2. Create a new firmware project with a clean, professional structure.

Task 3. Make the design portable: separate platform-specific HAL/board code from portable drivers and application logic so the project can be retargeted to another MCU with minimal changes.

Task 4. Pressure sensor sampling: read the pressure sensor at \approx 2 ms intervals (interrupt-based). You may use the existing pressure sensor driver code that was attached, adapted to STM32L0 HAL/LL. Sampling must be triggered by a hardware timer.

Task 5. I²C communication: implement I²C slave functionality to receive a 32-bit value from an external MCU (master) and to reply with a 32-bit value on request.(use different i2c bus then pressure sensor)

Task 6. DAC control: use the STM32L0 internal DAC(s) to set two output voltages. Provide API(s) that accept engineering units (e.g. volts) and convert/clip to DAC codes.