Embedded Systems II — Lab Plan (No Hardware)

Course Syllabus and Software-Based Laboratory Exercises

Course Details

Course: Embedded Systems II

Instructor: Morris Kaburu

Institution: Dedan Kimathi University of Technology

Target: https://8051-simulator.vercel.app/

Overview

This course does not require hardware for laboratory exercises. Instead, all labs are conducted using software simulators for the 8051 microcontroller, which are either free or widely accessible. Each laboratory session is designed to align with the core topics of the course outline, encompassing 8051 architecture, input/output operations, addressing modes, timer and interrupt management, assembly language programming, and the use of related tools and peripherals.

For each lab, students will submit the following deliverables:

- Source code
- Simulator waveforms or screenshots
- A brief interpretive report
- A reflection on data integrity and reproducibility

Lab 1: 8051 Assembly Fundamentals in a Simulator

Objectives

- Explore the 8051 microcontroller's architecture, registers, and instruction set
- Write and simulate basic assembly routines covering data movement, looping, and I/O emulation

Tools

Students may use online 8051 simulators such as emu8051 and Online 8051 IDEs, or free desktop simulators like Sim8051 and MIDE/Keil demo versions.

Tasks

- Inspect the 8051-block diagram and register banks within the simulator
- Write an assembly program that toggles P1.0 at approximately 1 Hz
- Implement a loop that increments register R0 and halts after 1000 iterations
- Add a subroutine (using CALL/RET) to blink two LEDs sequentially
- Execute the program, observe I/O states and waveforms, and capture outputs as screenshots

Deliverables

- Assembly source listing (ASM)
- Screenshot of waveforms showing P1.0 toggling and the LED sequence
- Brief reflection on instruction timings and cycles

Lab 2: Basic I/O and Port Addressing (8051)

Objectives

- Practice port input/output operations, bit-addressable I/O, and dual-port behaviour

Tools

Same 8051 simulator as used in Lab 1.

Tasks

- Configure Port 0 as input and Port 1 as output, or use the simulator's equivalent functionality
- Implement a debouncing routine for a button in assembly language
- Create a simple keyboard-like interface where pressing keys lights the corresponding LEDs
- Demonstrate bit-addressable operations using MOV instructions with bit-addressable SFRs

Deliverables

- Assembly source code
- Explanation of the debouncing routine
- Screenshot or video showing LED patterns

Lab 3: Analog-Digital Interaction (Software Simulation)

Objectives

- Conceptualise ADC and DAC interfaces and simulate analog sensing in software

Tools

8051 simulator equipped with pseudo-ADC blocks or a mock ADC implemented via a testbench in the simulator.

Tasks

- Create a project where a simulated ADC channel reads a virtual potentiometer
- Map the ADC result to control LED brightness or display a numeric value
- Implement a simple low-pass RC-like filter in software to smooth the readings
- Log ADC values to the console or a log window

Deliverables

- Source code (C or assembly)
- Log of sample ADC readings