ECE 278 – Assignment 3: Interval Timer Experiment

Overview

Use the Interval Timer peripheral to perform precise time measurements, detect timeouts, and control logic based on elapsed time by using low-level HAL API functions.

Objectives

- Understand the operation of the Interval Timer.
- Measure real-time intervals at microsecond precision.
- Detect user inactivity using timeout logic.
- Use polling to implement repeatable timing behavior.

Part A – Stopwatch Mode

Build a stopwatch that measures how long KEY2 is held down:

1. When KEY2 is pressed, record the current timer count.
2. When KEY2 is released, capture the new timer count.
3. Compute the difference in ticks and convert it into microseconds based on the timer clock frequency.
4. Display the least significant bits of the result on LEDR.
5. Show the elapsed time over seven segment display.

Part B – Timeout Detection

Simulates a basic 'watchdog' function, built with polling logic and the timer. Detect when the user has been inactive for 5 seconds:

1. Each time KEY3 is pressed, reset a 'last activity' timer.
2. Continuously monitor the current timer value.
3. If 5 seconds pass without a KEY3 press, light up LEDR9.
4. Pressing KEY3 again should clear the LED and restart the idle timer.

Part C – Period Measurement and Repeat

Design a two-press timer that repeats automatically every 5 seconds:

1. On the first press of KEY1, store the timer count.
2. On the second press of KEY1, capture the new count and compute the difference.
3. Display the measured duration (in ticks or converted to time).
4. After each full two-press measurement, repeat this timing cycle every 5 seconds automatically.
5. Blinking LEDR7 or seven segment to show cycle repetitions.