

Assignment 3 – Timer & EXTI Task

by: Thomas Fischer

Objective

In this assignment, you will compare two approaches for detecting a button press (falling edge) using the STM32L432KC Nucleo development board together with the UAS Technikum Wien Click-Shield.

- The built-in button on the Click-Shield serves as the input.
- The onboard RGB LED on the Click-Shield is used to visualize the result.

Refer to the Click-Shield schematics on Moodle to understand the exact pin assignments and electrical connections.

Your personal parameters (assigned timer and LED color) are provided in Moodle via the test: “Get assignment 3 parameters”.

Part One – Timer-Based Polling Detection (3.0 of the 7.5 points)

Task

- Use your assigned timer (e.g., TIM1) to generate a periodic interrupt with a 10 ms interval.
- In the Timer IRQ Handler, detect if the Click-Shield button has been pressed (falling edge).
- If yes, toggle the assigned RGB LED color (e.g., Red).
- Do the same check in the HAL Timer Period Elapsed Callback, but toggle a different LED color (e.g., Green).

This should compare the latency between the actual Timer IRQ Handler – and the overhead of the Callback.

LED color assignments (Red, Green, Blue) are individual and specified in Moodle.

Measurement

- Use a logic analyzer to monitor the LED output pins.
- Measure the latency between the actual button press and LED toggle.
- Repeat the experiment 3 times, document your results with screenshots and a latency table.

Part Two – External Interrupt (EXTI) Detection (3.0 of the 7.5 points)

Task

- Configure the Click-Shield button to generate an external interrupt (EXTI) on falling edge.
- In the EXTI IRQ Handler, toggle the assigned LED color (e.g., Red).
- In the HAL EXTI Callback, toggle a different LED color (e.g., Blue).

Measurement

- Again, use the logic analyzer to capture the toggle timing.
- Perform 3 repetitions and document the latency behavior as before.

Verification (1.5 of the 7.5 points)

- Summarize your measurements from both Part One and Part Two in a single table.
- Highlight the shortest and longest latency values across all runs.
- Write a short paragraph reflecting on the timing behavior of polling vs. interrupt-driven approaches.

Implementation Notes

- Configure the output GPIOs (RGB LED) with maximum speed to reduce latency overhead
- You have to use the Logic-Analyzer for the measurements/evaluation
- Follow a layered code structure. Avoid implementing everything in main.c

Grading Criteria

The grading criteria are functionality, code quality, modularity, and structuring.