Task Description

This project demonstrates the use of **external interrupts** on the ATmega32 microcontroller. Two different functionalities are implemented:

1. **Toggle LED using EXTI0 (INT0):**
    - When the push button connected to **INT0 (PD2)** is pressed, the LED connected to **PD7** toggles its state (ON/OFF).
2. **Reset Seven-Segment Counter using EXTI1 (INT1):**
    - A seven-segment display is connected to **PORTC (PC0–PC6)**.
    - The display continuously counts from **1 to 9**.
    - When the push button connected to **INT1 (PD3)** is pressed, the counter is reset back to **1**.

Hardware Connections

- **INT0 (PD2):** Push button for toggling LED.
- **INT1 (PD3):** Push button for resetting seven-segment counter.
- **LED:** Connected to **PD7**.
- **Seven-Segment Display:** Connected to **PC0–PC6** (common cathode).
- **VCC & GND:** Properly connected to the ATmega32 and components.

Software Implementation

- **Drivers Developed:**
    1. EXTI0.c / EXTI0.h / EXTI0_register.h → Custom driver for external interrupt 0.
    2. EXTI1.c / EXTI1.h / EXTI1_register.h → Custom driver for external interrupt 1.
- **Main Program Logic:**
    1. Initialize ports and configure external interrupts.
    2. Start infinite loop:
        - Display numbers from 1 to 9 on the seven-segment display.
        - Delay between numbers for visibility.

3. **Interrupt Service Routines (ISR):**
    - **ISR(INT0):** Toggles LED.
    - **ISR(INT1):** Resets counter to 1.

Simulation Behavior

1. At startup, the seven-segment display begins counting from **1 to 9** repeatedly.
2. Pressing the **INT0 button** (PD2) → LED on **PD7** toggles.
3. Pressing the **INT1 button** (PD3) → Seven-segment counter resets to **1** immediately.