# Lab 7 – Interrupts and Event Programming – Prelab Report

Sunday, March 16th, 2025

Joey McIntyre (400520473), Devin Gao (400508489)

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is our own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario.

1. **When referring to interrupt triggering via GPIO, explain the difference between falling edge, rising edge, low level, and high level. (Hint - see textbook section 9.5) [5 marks]**

Interrupts can be triggered in different ways based on signal changes in the GPIO pin. For a falling edge, the interrupt is triggered when the signal transitions from HIGH to LOW. A rising edge, the interrupt is triggered when the signal transitions from LOW to HIGH. For low level, the interrupt is triggered as long as the signal remains LOW. For high level, the interrupt is triggered as long as the signal remains HIGH.

Edge-triggered interrupts only occur when the signal transitions, where level-triggered interrupts occur and last as long as the signal is at the required level.

2. **What five conditions must be true for an interrupt to occur? [5 marks]**

The five conditions that must be met for an interrupt to occur include:

i. The interrupt source must be configured and armed to generate an interrupt.
ii. The interrupt must be explicitly enabled in the Nested Vectored Interrupt Controller (NVIC).
iii. Global interrupts must be enabled to allow the processor to handle them.
iv. The interrupt must be assigned an appropriate priority level to ensure correct execution order.
v. An Interrupt Service Routine (ISR) must be written and linked to handle the interrupt event properly.

3. **How do you enable interrupts? [5 marks]**

The steps to enable interrupts are as follows:

i. The processor state registers (PSR, PC, LR, R12-R0) are pushed onto the stack to save the current execution state.
ii. The Program Counter (PC) is set to the address of the interrupt vector corresponding to the triggered interrupt.

iii. The Interrupt Program Status Register (IPSR) is updated with the interrupt number (e.g., set to 18).

iv. The Link Register (LR) is configured, with its top 24 bits set to 0xFFFFFFF and the bottom 8 bits set to 0xF9, indicating return to Thread Mode upon ISR completion.

v. The Main Stack Pointer (MSP) is used as the active stack pointer for handling the interrupt.

**4. What are the steps that occur when an interrupt is processed? [5 marks]**

When an interrupt occurs, the following steps occur.

i. The processor completes the current instruction, unless a higher-priority interrupt requires immediate handling.

ii. The current execution state is saved to ensure seamless resumption after interrupt handling.

iii. The Interrupt Service Routine (ISR) is executed to handle the interrupt event.

iv. The processor identifies the cause of the interrupt (e.g., GPIO event, timer overflow, or external signal).

v. The ISR executes its defined task, such as clearing interrupt flags or handling device input.

vi. Once completed, the processor restores the previous execution state and resumes normal program execution from where it left off.