**CG2271 Real Time Operating Systems**

**Lab 2 – Interrupt Programming**

**Answer Book**

**Due date: Friday 12 September 2359 hours**

**Fill in all your teammates’ names here. Submit only ONE copy per team.**

|  | **Name** | **Student ID (Axxxxx)** |
| --- | --- | --- |
| 1. | John Woo Yi Kai | A0272561E |
| 2. | Rishi Moorthy | A0273618X |
| 3. | Wang Yanjie | A0276202M |
| 4. | Kenneth Wong Cun Wi | A0303203A |
| 5. | Kuek Yeau Hao Jonathan | A0258485M |

Question 1 (3 marks)

Answer:

When you press and hold SW3, the LED **doesn't** change color while you're holding it. It changes color only when you **release** the switch.

This behavior is determined by the PORT\_PCR\_IRQC setting in the initInterrupt function. The line PORTA->PCR[SWITCH\_PIN]|= PORT\_PCR\_IRQC(0b1001); configures the external interrupt. The value 0b1001 corresponds to **rising-edge triggering**. A rising edge occurs when a signal transitions from a low voltage (logic 0) to a high voltage (logic 1).

The circuit for the switch on the FRDM board uses a **pull-up resistor**. This means that when the switch isn't pressed, the input pin (PTA4) is pulled **high** (logic 1). When you press the switch, you connect the pin to **ground** (logic 0). Releasing the switch disconnects it from ground, and the pull-up resistor returns the pin's voltage to high. Therefore, the interrupt is only triggered on the **rising edge** which is when you **release** the switch, to changes the LED color, not when pressing down the switch.

Question 2 (3 marks)

Answer:

After changing the line to

PORTA->PCR[SWITCH\_PIN]|= PORT\_PCR\_IRQC(0b1010);

, the LED changes color when you **press down** on the switch. It does not change color when you release it.

The new value 0b1010 for PORT\_PCR\_IRQC configures the interrupt to trigger on a **falling edge**. A falling edge is a transition from a high voltage (logic 1) to a low voltage (logic 0).

Because the switch's circuit uses a pull-up resistor, the input pin is normally high. When you press the switch, the pin is pulled low, creating a **falling edge**. This is the specific event that the code is now configured to detect, so the interrupt is triggered, and the LED color changes.

Question 3 (4 marks)

Answer:

**void** **initSW2Interrupt**() {

// Disable interrupts before configuring

NVIC\_DisableIRQ(*PORTC\_PORTD\_IRQn*);

// Enable clock for PORTC

SIM->SCGC5 |= SIM\_SCGC5\_PORTC\_MASK;

// Set PTC3 to GPIO

PORTC->PCR[SW2] &= ~PORT\_PCR\_MUX\_MASK;

PORTC->PCR[SW2] |= PORT\_PCR\_MUX(1);

// Enable pull-up resistor (PS bit) and pull enable (PE bit)

PORTC->PCR[SW2] &= ~PORT\_PCR\_PS\_MASK;

PORTC->PCR[SW2] |= PORT\_PCR\_PS(1);

// Set as input

GPIOC->PDDR &= ~(1 << SW2);

PORTC->PCR[SW2] &= ~PORT\_PCR\_PE\_MASK;

PORTC->PCR[SW2] |= PORT\_PCR\_PE(1);

// Configure interrupt for falling edge

PORTC->PCR[SW2] &= ~PORT\_PCR\_IRQC\_MASK;

PORTC->PCR[SW2] |= PORT\_PCR\_IRQC(0b1010);

// Set NVIC priority to the lowest (192)

NVIC\_SetPriority(*PORTC\_PORTD\_IRQn*, 192);

// Clear pending interrupts

NVIC\_ClearPendingIRQ(*PORTC\_PORTD\_IRQn*);

// Enable interrupts

NVIC\_EnableIRQ(*PORTC\_PORTD\_IRQn*);

}

Question 4 (2 mark)

Answer:

We always set the **PS (Pull Select)** bit before the **PE (Pull Enable)** bit because the PS bit determines the *type* of resistor (pull-up or pull-down) to be enabled. The PE bit simply turns the pull resistor on or off. By setting PS first, we ensure that when PE is set, the correct resistor is enabled. Setting PE first then PS may result in undesired behaviours such as a trigger of an interrupt since we enable first before configuring correctly. Hence, the order is: first, specify what you want (pull-up), then, enable it.

Question 5 (5 marks)

Answer:

// SW3 handler. SW3 toggles the mode

**void** **PORTA\_IRQHandler**() {

// Clear pending IRQ for PORTA

NVIC\_ClearPendingIRQ(*PORTA\_IRQn*);

// Check that SW3 was triggered

**if**(PORTA->ISFR & (1 << SW3)) {

// mode cycles from 0 to 1 to 2 to 0, etc.

mode = (mode + 1) % 3;

}

// Write a 1 to clear the ISFR bit

PORTA->ISFR |= (1 << SW3);

}

**Report: \_\_\_\_\_\_\_\_\_\_\_/17**

**Demo : \_\_\_\_\_\_\_\_\_\_ / 3**

**Total: \_\_\_\_\_\_\_\_\_\_\_\_ / 20**