MASARYK
UNIVERSITY
Czech Republic

PV198 – One-chip Controllers

Interrupts

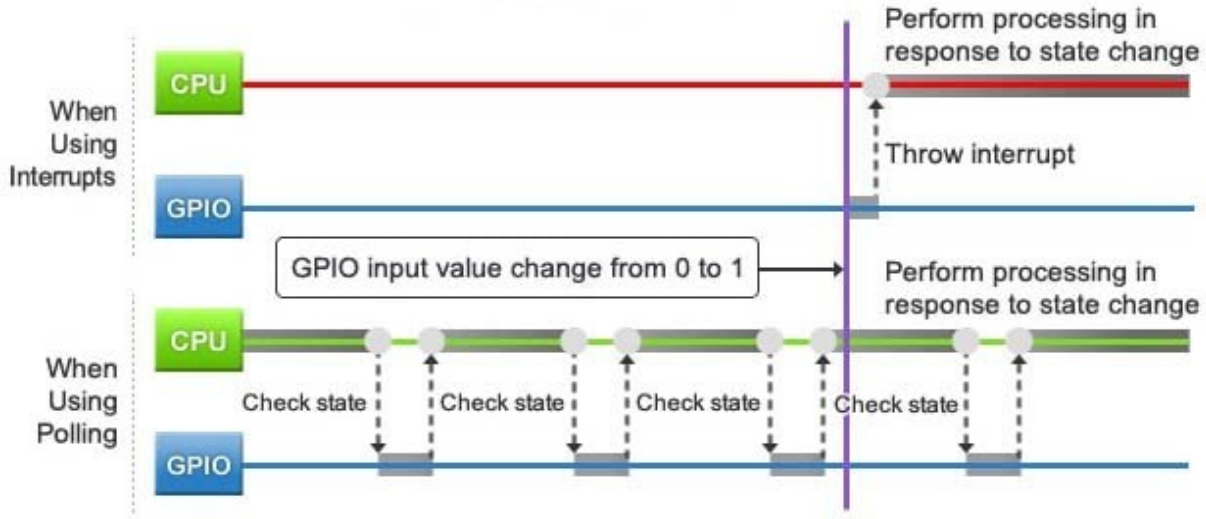
MASARYK UNIVERSITY
Czech Republic

Content

1. What is Interrupt
2. How does it work
3. NVIC – **N**ested **V**ector **I**nterrupt **C**ontroller
4. Code
5. Application

Interrupts – What it is

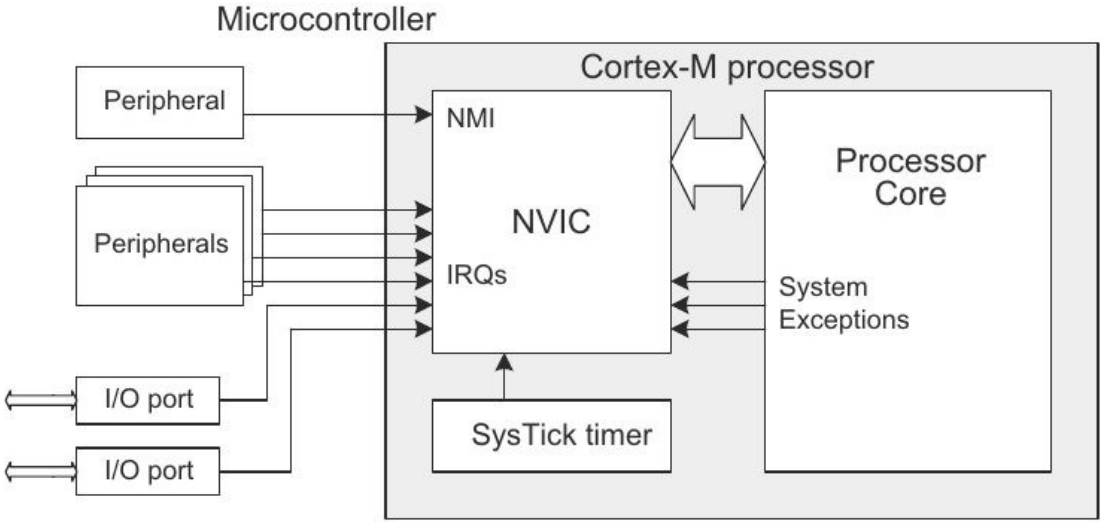
- Signal to the processor indicating an event



https://www.motioncontroltips.com/what-is-nested-vector-interrupt-control-nvic/

Interrupts – Types

- Level / Edge trigger
- HW / SW interrupt



https://www.motioncontroltips.com/what-is-nested-vector-interrupt-control-nvic/

MASARYK UNIVERSITY
Czech Republic

Interrupts – How does it work

- Interrupt is triggered
- MCU finish execution of current instruction
- Save current state (PC, flags, registers)
- Get address of an **ISR** (**I**nterrupt **S**ervice **R**outine) from Interrupt Vector Table
- Execute ISR
- Load previously saved state and continue

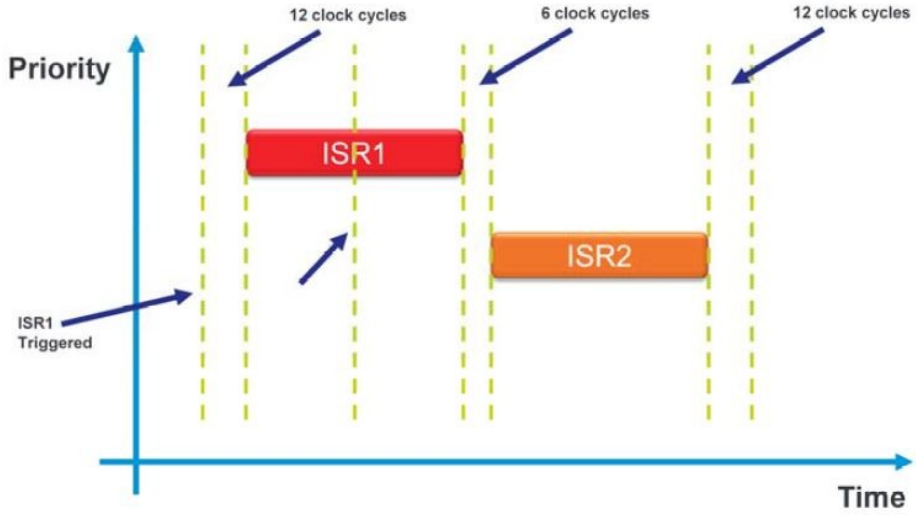
MASARYK UNIVERSITY
Czech Republic

Interrupts – NVIC (FRDM-K66F)

- **NVIC – N**ested **V**ector **I**nterrupt **C**ontroller
- Up to 120 interrupt sources
- Up to 16 priority levels
- Nested interrupts
- 12 clock cycles to enter/exit an **ISR** (**I**nterrupt **S**ervice **R**outine)
- 6 clock cycles when switching from one ISR to another

Interrupts

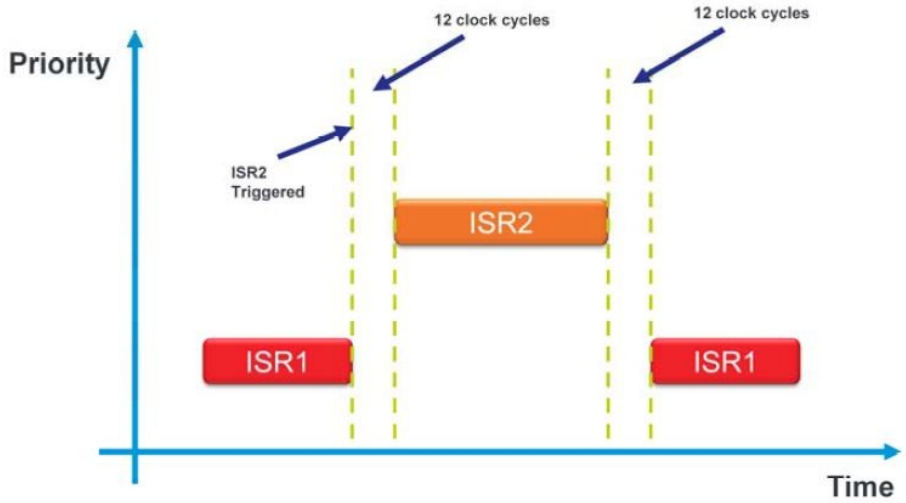
- Tail Chaining



http://cache.freescale.com/files/training/Nested-Vector-Interrupt-Controller-Training.pdf

Interrupts

- Preemption



http://cache.freescale.com/files/training/Nested-Vector-Interrupt-Controller-Training.pdf

MASARYK UNIVERSITY
Czech Republic

Interrupts – Code (GPIO button)

1. Configure interrupt for a pin
2. Enable interrupts for a port
3. Write interrupt handler

Interrupts – Code (GPIO button)

- Configure interrupt for a pin (button) – *can be done in Pins tool*

```
/* Interrupt configuration on PORTA10 (pin M9): Interrupt on falling edge */
    PORT_SetPinInterruptConfig(BOARD_SW3_PORT, BOARD_SW3_PIN,
kPORT_InterruptFallingEdge);
```

Interrupts – Code (GPIO button)

- Enable interrupts for a port (GPIOA for SW3) – *can be done in Peripherals tool*
- (Optionally) Define your handler name – *Peripherals tool*

```
/* GPIO_1 interrupt handler identifier. */
#define SW3_BUTTON_PRESSED_IRQ PORTA_IRQHandler

/* Enable interrupt PORTA_IRQn request in the NVIC */
  EnableIRQ(PORTA_IRQn);
```

MASARYK UNIVERSITY
Czech Republic

Interrupts – Code (GPIO button)

- Write interrupt handler
- Don’t forget to clear a flag that triggers interrupt

```
/**
 * Interrupt handler for button SW3.
 * Set flag when SW3 button is pressed.
 */
void SW3_BUTTON_PRESSED_IRQ(void) {
    /* Clear external interrupt flag. */
    GPIO_PortClearInterruptFlags(BOARD_SW3_GPIO, 1U << BOARD_SW3_GPIO_PIN);
    /* Change state of button. */
    g_ButtonSw3Press = true;
}
```

MASARYK UNIVERSITY
Czech Republic

Application

Goal:

- Write an application that prints text to a console when SW3 button is pressed. Use interrupts.

- You can use the application from the previous lesson and modify it to use interrupt instead of polling