

# Task 4. Timers and interrupts

AMM embedded course

# Links

- stm32\_reference\_manual.pdf - Part 12, Interrupts and events, Part 18, General-purpose timers
- <https://habrahabr.ru/post/218825/>
- <http://visualgdb.com/tutorials/arm/stm32/timers/>

# Nested vectored interrupt controller (NVIC)

- 82 maskable interrupt channels for STM32F405xx/07xx and STM32F415xx/17xx, and up to 91 maskable interrupt channels for STM32F42xxx and STM32F43xxx (not including the 16 interrupt lines of Cortex<sup>®</sup>-M4 with FPU)
- 16 programmable priority levels (4 bits of interrupt priority are used)
- low-latency exception and interrupt handling
- power management control
- implementation of system control registers

# Hardware interrupt selection

To configure the 23 lines as interrupt sources, use the following procedure:

- Configure the mask bits of the 23 interrupt lines (EXTI\_IMR)
- Configure the Trigger selection bits of the interrupt lines (EXTI\_RTSR and EXTI\_FTSR)
- Configure the enable and mask bits that control the NVIC IRQ channel mapped to the external interrupt controller (EXTI) so that an interrupt coming from one of the 23 lines can be correctly acknowledged

# General-purpose timers (TIM2 to TIM5)

- 16-bit (TIM3 and TIM4) or 32-bit (TIM2 and TIM5) up, down, up/down auto-reload counter.
- 16-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65536.
- Synchronization circuit to control the timer with external signals and to interconnect several timers.
- Interrupt/DMA generation on the following events:
  - Update: counter overflow/underflow, counter initialization (by software or internal/external trigger)
  - Trigger event (counter start, stop, initialization or count by internal/external trigger)
  - Input capture
  - Output compare

## Task 4. Rewrite main cycle via timer

- Init general-purpose timer with interrupt (NVIC\_Init(), TIM\_TimeBaseInit(), TIM\_ITConfig(), TIM\_Cmd())
- In timer interrupt handler - move leds to next iteration

## Task 4. Rewrite button press via interrupt

- Init interrupt pin (it is button pin), connect it to interrupt line with `SYSCFG_EXTILineConfig()`
- Configure interrupt line (`EXTI_Init()`)
- Enable the interrupt with lowest priority (`NVIC_Init()`)
- In interrupt handler – switch button logic