

# Prekinitve & NVIC

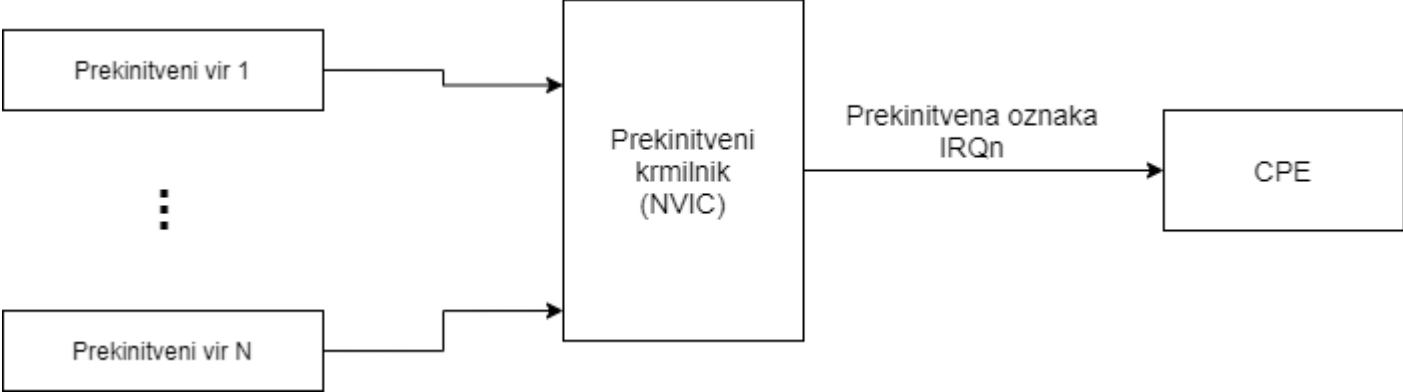
Vgrajeni sistemi

Rok Češnovar

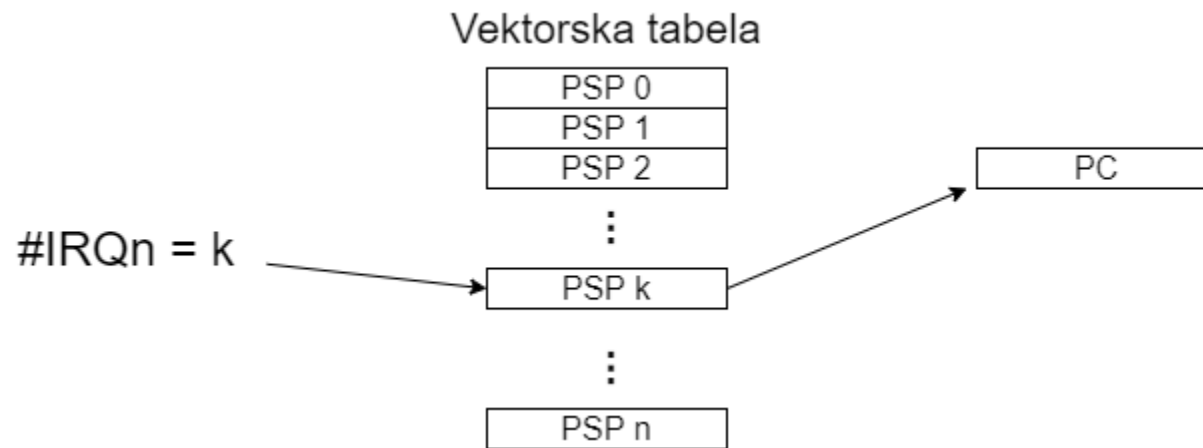
# Prekinitve

- dogodek, ki povzroči, da procesor prekine izvajanje trenutno izvajajočega se programa
- vsaka prekinitev/izjema ima določen svoj prekinitven-servisni program (PSP)
  - PSP je funkcija, ki obravnava prekinitve

# Prekinitve



# Prekinitve



# Prekinitveni krmilnik

- Prekinitve obravnava prekinitveni krmilnik NVIC (Nested vectored interrupt controller)
  - omogoča priklop do 240 prekinitvenih izvorov
  - vsakemu izvoru lahko določimo prioriteto
    - dve prioritetni oznaki po 16 prioritetnih nivojev
      - (0,0), (0,1), .... (15, 14), (15,15)
- za vsak izvor je vnaprej določena oznaka (IRQn), seznam najdemo v

`Drivers/CMSIS/Device/ST/STM32F4xx/Include/stm32f769xx.h`

- Procesor prejme le oznako prekinitve

# Vektorska tabela

- **definirana v** Core/Startup/startup\_stm32f769nihx.s.

g\_pfnVectors:

```
.word  _estack
.word  Reset_Handler
.word  NMI_Handler
.word  HardFault_Handler
.word  MemManage_Handler
.word  BusFault_Handler
.word  UsageFault_Handler
...
.word  FLASH_IRQHandler
.word  RCC_IRQHandler
.word  EXTI0_IRQHandler
.word  EXTI1_IRQHandler
.word  EXTI2_IRQHandler
...
.word  TIM3_IRQHandler
```

- funkcija, ki bo PSP prekinitve, mora imeti enako ime kot je zapisano v vektorski tabeli

# Vklop prekinitev

- NVIC
  - poiščemo IRQn za napravo, za katero nastavljamo prekinitev
  - določimo prioriteto
    - v knjižnici je prioriteta dvonivojska (16 x 16)

```
HAL_NVIC_SetPriority(OZNAKA_PREKINITVE_IRQn, 1, 2);  
HAL_NVIC_EnableIRQ(OZNAKA_PREKINITVE_IRQn);
```

- Naprava
  - proženje prekinitev omogočimo tudi na napravi
  - kako? glej ref. manual ali knjižnico naprave

# PSP

- v `Src/stm32f7xx_it.c` poiščemo funkcijo, ki služi kot PSP
  - če ne obstaja, jo dodamo (POZOR: Ime se mora ujemati z imenom v vektorski tabeli)
- 1. če ima naprava več virov prekinitev , preverimo če je prekinitev prožil željen vir
- 2. opravimo željeno nalogo PSP
  - npr. prižgi/ugasni LED, beri senzor, ...
- 3. brišemo prekinitveno zahtevo (zastavico)



# Prekinitve GPIO - EXTI

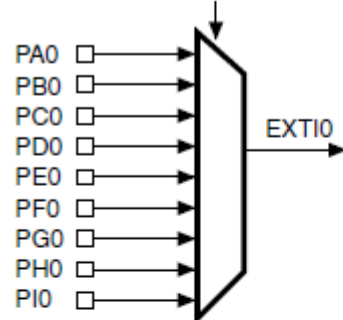
Vgrajeni sistemi

Rok Češnovar

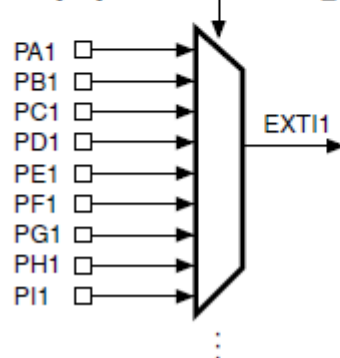
# EXTI

- External Interrupt (zunanje prekinitve)
- EXTI krmilnik omogoča priklop GPIO naprav na NVIC

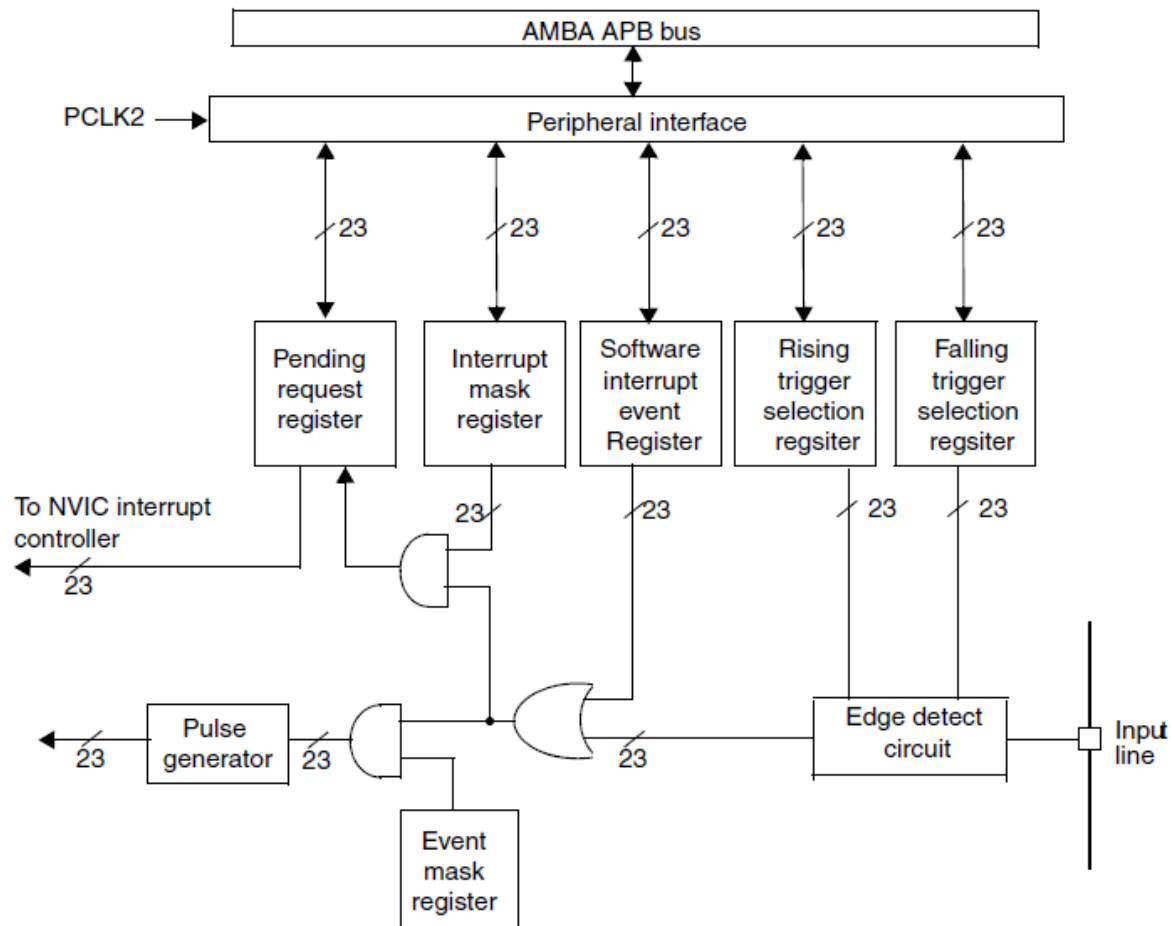
EXTI0[3:0] bits in the SYSCFG\_EXTICR1 register



EXTI1[3:0] bits in the SYSCFG\_EXTICR1 register



# Bloční diagram EXTI



# EXTI

- EXTI krmilnik skrbi le za preverjanje nastopa morebitnih front in posredovanje prekinitve NVIC
- S knjižnico HAL vklopimo uporabo EXTI enostavno tako, da izberemo način delovanja GPIO pina
  - za nastavljanje EXTI naprave poskrbi HAL

# Primer inicializacije EXTI

```
__HAL_RCC_GPIOE_CLK_ENABLE();
```

```
GPIO_InitTypeDef init_structure;
```

```
init_structure.Pin = GPIO_PIN_3;
```

```
init_structure.Mode = GPIO_MODE_IT_RISING;
```

```
init_structure.Pull = GPIO_NOPULL;
```

```
init_structure.Speed = GPIO_SPEED_FREQ_LOW;
```

```
HAL_GPIO_Init(GPIOE, &init_structure);
```

- **Možnosti za Mode** so `GPIO_MODE_IT_RISING`, `GPIO_MODE_IT_FALLING` ter `GPIO_MODE_IT_RISING_FALLING`
- Ob zaznani fronti se bo aktiviral PSP `EXTI3_IRQHandler`

# Prekinitev EXTI

- Branje prekinitev

```
__HAL_GPIO_EXTI_GET_IT(GPIO_PIN_3)
```

- Brisanje prekinitev

```
__HAL_GPIO_EXTI_CLEAR_IT(GPIO_PIN_3)
```

# Prekinitev SysTick

- Proži se ob konstantnem intervalu
  - običajno 1ms
- Prioriteta prekinitve je (15, 0)
- Ob prekinitvi se požene SysTick\_Handler()

# Naloga za vajo

- Ob pritisku gumba naj se proži prekinitev
- Ob kratkem pritisku naj se LED prižgejo
- Ob dolgem pritisku ( $\geq 1s$ )