

Программирование микроконтроллеров STM32

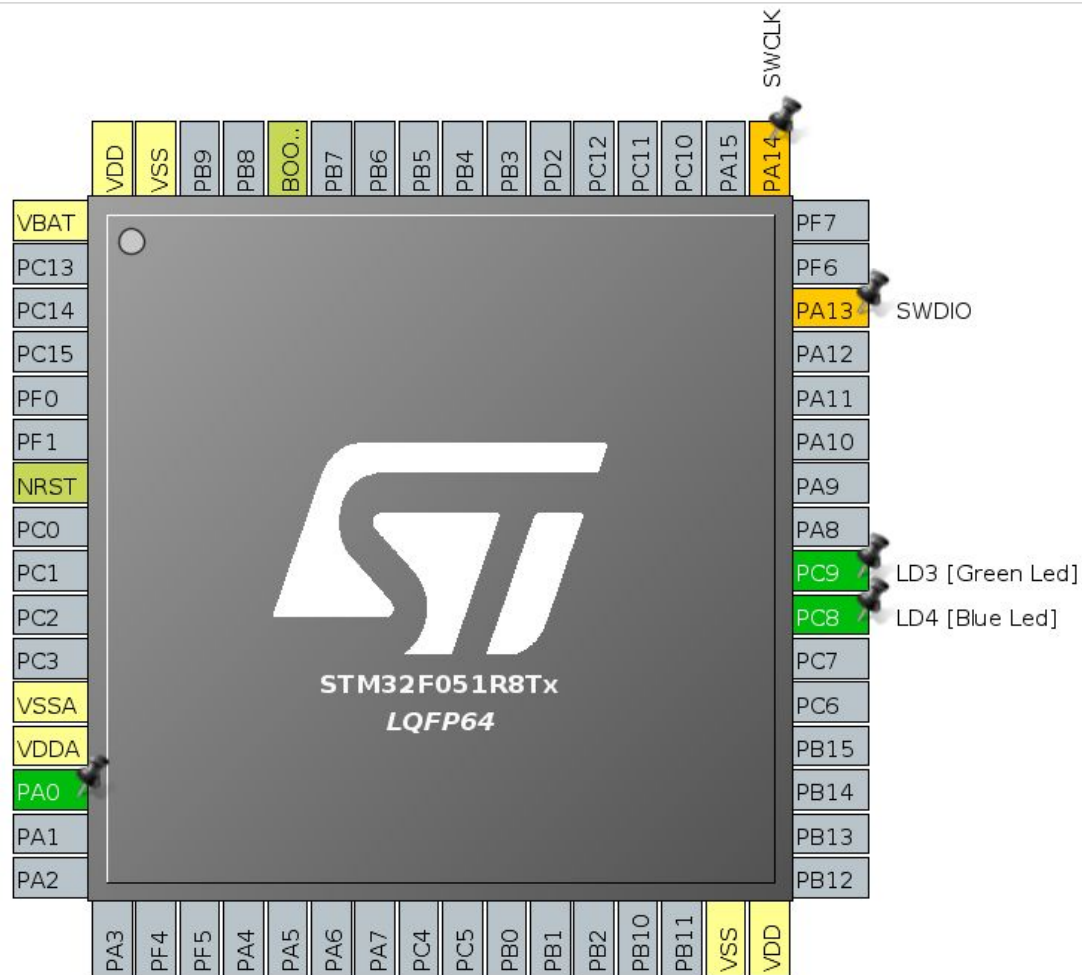
*Внешние прерывания EXTI.
Системный таймер.*

Предыдущий пример

- Постоянный опрос кнопки
- Примитивная задержка

Контроллер внешних прерываний

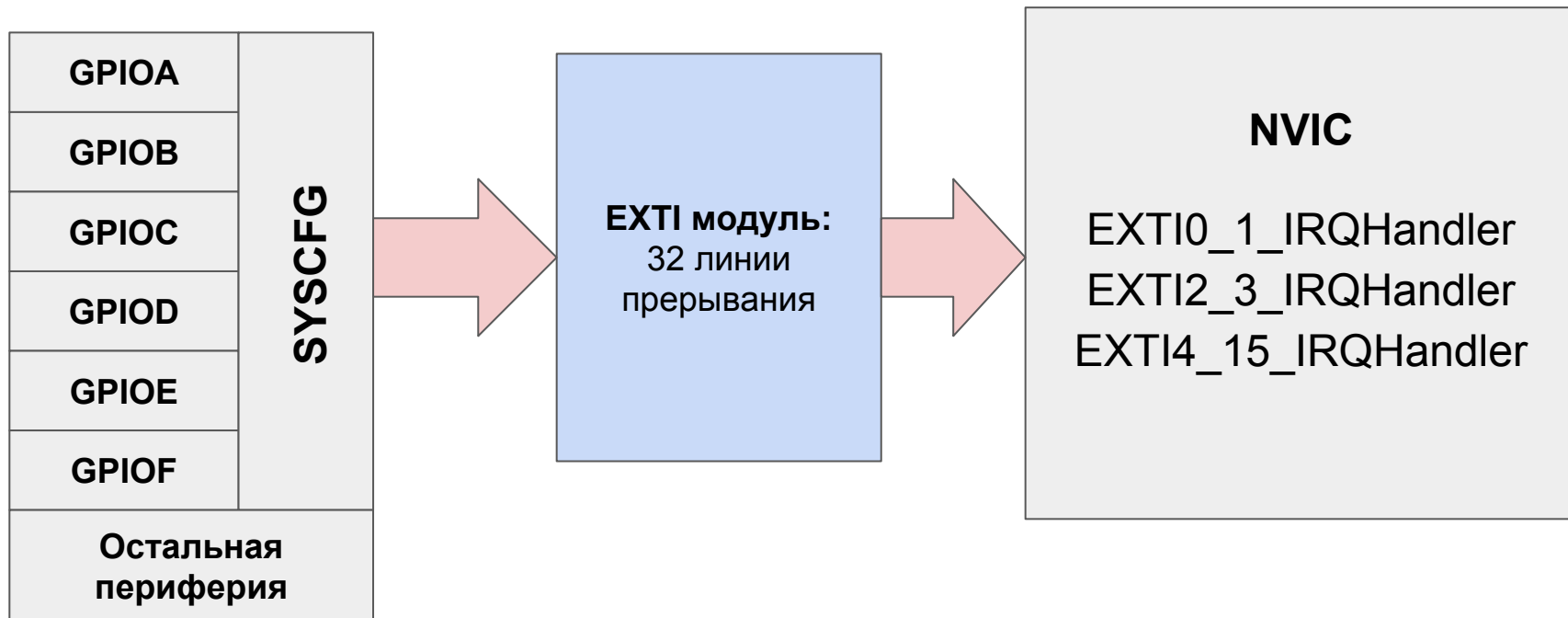
B1 [Blue PushButton]



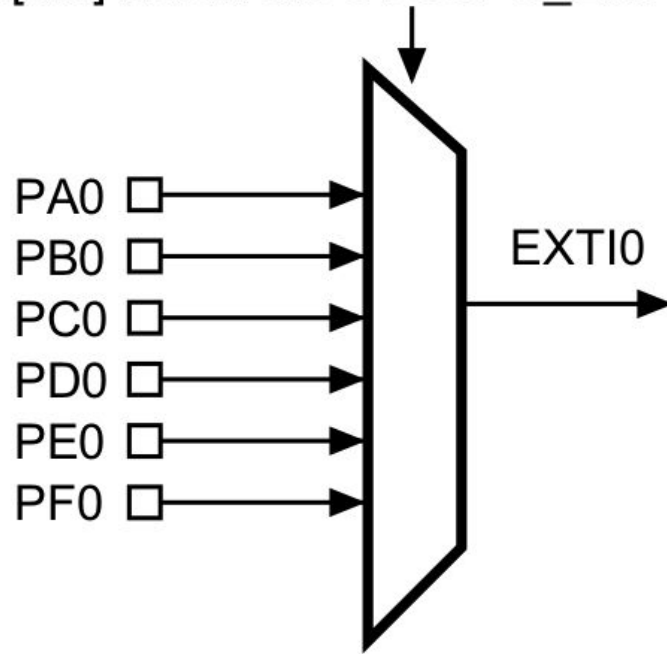
Контроллер внешних прерываний

- Поддержка 32 внешних прерывания/событий (50/50)
- Возможность конфигурирования каждого прерывания по отдельности
- Возможность детектирования переднего/заднего фронтов
- Статус регистр для каждой линии прерывания
- Поддержка эмулирования каждой линии прерывания

Связь EXTI, NVIC и SYSCFG



EXTI0[3:0] bits in the SYSCFG_EXTICR1 register

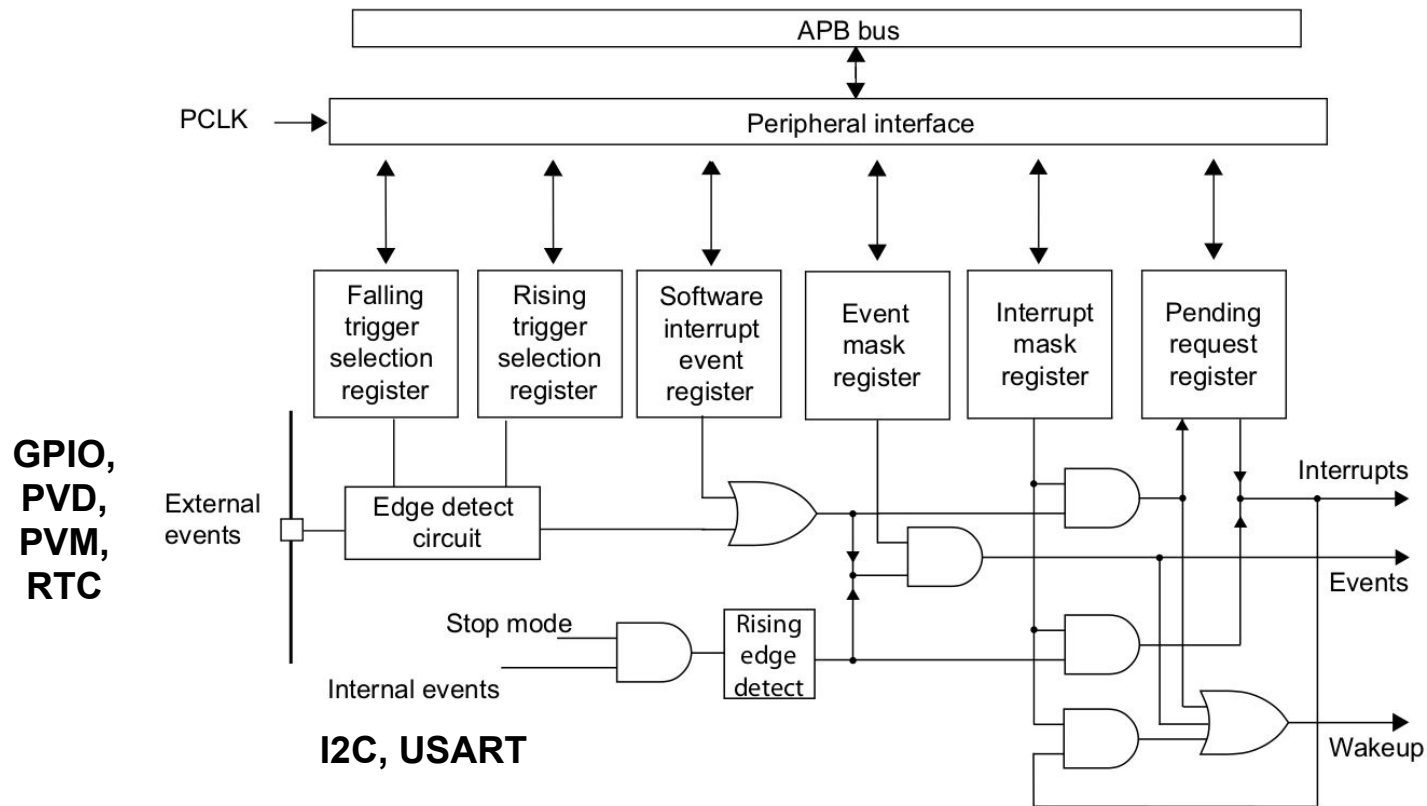


LL_SYSCFG_SetEXTISource (Только один пин!)

1) `LL_SYSCFG_EXTI_PORTx` -> `LL_SYSCFG_EXTI_PORTA`

2) `LL_SYSCFG_EXTI_LINEx` -> `LL_SYSCFG_EXTI_LINE0`

Внутренняя структура EXTI



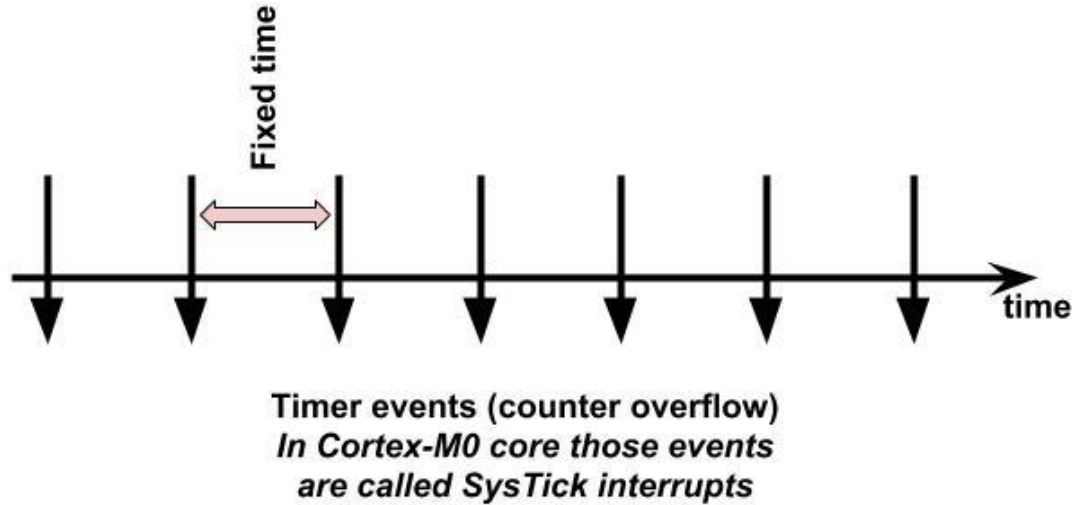
Настраиваем EXTI

- *LL_EXTI_EnableIT_0_31* -> Включить прерывание
 - *LL_EXTI_LINE_x*
- *LL_EXTI_EnableFallingTrig_0_31* -> Задний фронт
 - *LL_EXTI_LINE_x*
- *LL_EXTI_EnableRisingTrig_0_31* -> Передний фронт
 - *LL_EXTI_LINE_x*
- *LL_EXTI_IsActiveFlag_0_31*
 - *LL_EXTI_LINE_x*
- *LL_EXTI_ClearFlag_0_31*
 - *LL_EXTI_LINE_x*

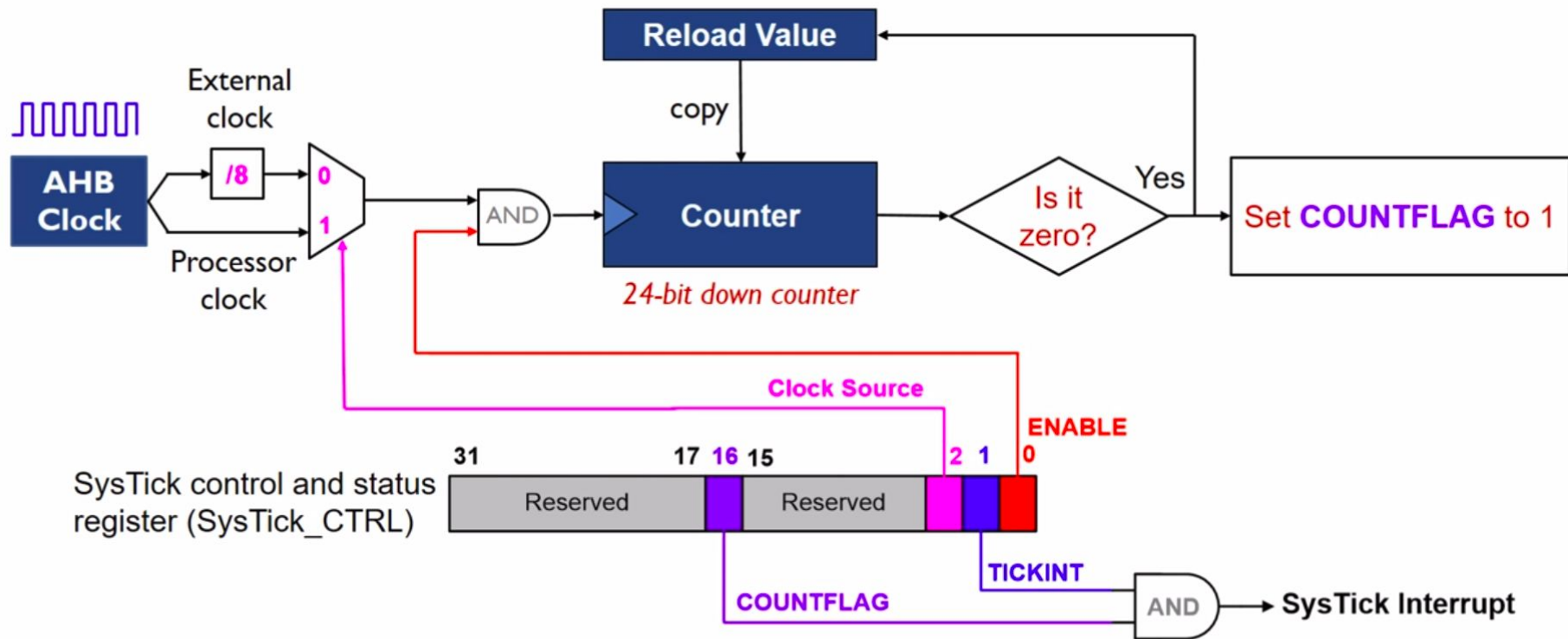
Настраиваем NVIC

- *NVIC_EnableIRQ* -> Включить прерывание
 - IRQn_Type IRQn)
- *NVIC_SetPriority* -> Настроить приоритет
 - IRQn_Type IRQn
 - uint32_t priority

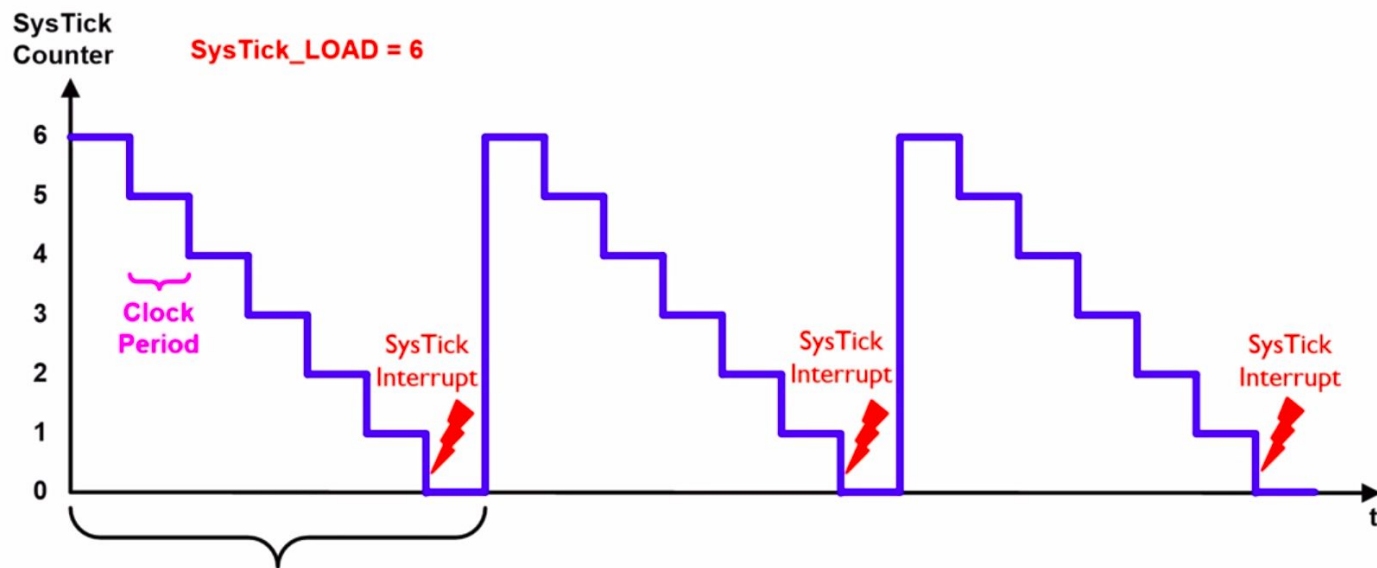
Системный таймер



Системный таймер



Системный таймер



$\text{SysTick Interrupt Time Period} = (\text{SysTick_LOAD} + 1) \times \text{Clock Period} = 7 \times \text{Clock Period}$

Системный таймер. Инициализация

- *SysTick_Config* (количество тактов)
- Имплементация `void SysTick_Handler(void)`

Репозиторий

https://github.com/edosedgar/stm32f0_ARM