Курс по STM32

Лекция #7:

- Выдача ДЗ №4.
- Внешние прерывания.
- Таймеры в STM32F051:
  устройство, предзагрузка, режимы по сравнению и по захвату.

18.03.2023 / 20.03.2023

ДЗ №4: 03_systick

03_systick: как настраивать исключения?

Настройка системного таймера:

1) Реализация обработчика.
2) Задание вектора прерываний.
3) Настройка таймера на нужную частоту.

```
void systick_handler(void)
{
    static int handler_ticks = 1U;

    handler_ticks += 1U;

    if (handler_ticks == 10000U)
    {
        handler_ticks = 0U;

        uint32_t reg_gpio_odr = *GPIOC_ODR;
        *GPIOC_ODR = (reg_gpio_odr & ~0x0100U) | (~reg_gpio
    }
}
```

```
.section .vector_table
.word __stack_start
.word __reset_handler
.word __exc_handler
.word __exc_handler
.fill 7, 4, 0x00
.word __exc_handler
.fill 2, 4, 0x00
.word __exc_handler
.word systick_handler
```

03_systick: секции .data и .bss

Куда ложатся статические переменные?

```
static int ticks = 1U;

❯ arm-...-objdump -x systick.elf
SYMBOL TABLE:
...
20000000 l     O .data 4 ticks.4023
...
```

```
static int ticks = 0U;

❯ arm-...-objdump -x systick.elf
SYMBOL TABLE:
...
20000000 l     .bss 4 ticks.4023
...
```

Вся прошивка записывается на Flash-память. Flash - почти что Read-Only.

Как быть?

03_systick: перенос секции .data

```
Sections:
Idx  Name    VMA       LMA
...
  3  .data   20000000  00000394
```

Перенос .data из LMA по VMA!

```
.data :
{
    . = ALIGN(4);
    __data_start_vma = .;

    *(.data)

    . = ALIGN(4);
    __data_end_vma = .;
} >SRAM AT >FLASH

__data_start_lma = LOADADDR(.data);
```

```
__reset_handler:
    // Copy .data section to SRAM:
    ldr r0, __data_start_lma_val
    ldr r1, __data_start_vma_val
    ldr r2, __data_end_vma_val

__loop_copy_data_section:
    cmp r1, r2
    beq __loop_copy_data_section_end

    ldr r4, [r0, #0]
    str r4, [r1, #0]

    adds r0, r0, #4
    adds r1, r1, #4
    b __loop_copy_data_section
__loop_copy_data_section_end:
```

03_systick: гонка с исключением

Пример гонки с исключением:

```
main: *GPIOC_ODR |= 0x0200U;
24a:  ldr   r3, [pc, #32]  ; r3 = GPIOC_ODR
24c:  ldr   r2, [r3, #0]   ; r2 = *r3
24e:  ldr   r3, [pc, #28]  ; r3 = GPIOC_ODR   Исключение!
250:  movs  r1, #128       ; r1 = 128         Запись в GPIOC_ODR
252:  lsls  r1, r1, #2     ; r1 = r1 << 2
254:  orrs  r2, r1         ; r2 |= r1
256:  str   r2, [r3, #0]   ; *r3 = r2         Потеря данных :(
```

Как быть?

03_systick: гонка с исключением

```
; Запрет конфигурируемых исключений (PRIMASK = 1):
__asm__ volatile(“cpsid i”);
; Включение исключений (PRIMASK = 0):
__asm__ volatile(“cpsie i”);
```

Table 2-7 PRIMASK register bit assignments

Bits	Name	Function
[31:1]	-	Reserved
[0]	PRIMASK	0 = no effect 1 = prevents the activation of all exceptions with configurable priority.

Требования к ДЗ №4

```
[1] Отрефакторить код 03_systick и разметить все регистры:
    [ ] Регистры используются только по их именам.
    [ ] Используются биты регистров только по их именам.
    [ ] Произвести перенос секции .bss в SRAM-память.
    [ ] Избавиться от гонки с диодами, используя GPIO_BSRR.
[2] Переписать игру "пальчики" и/или пример из 02_gpio.
    [ ] Запрещается использовать функцию delay.
    [ ] Разрешается использовать исключения.
    [ ] В итоговой программе не должно быть гонок.
```

Внешние прерывания

Внешние прерывания и события

У контроллера EXTI 32 линии:

- Прерывания/события по состоянию пинов GPIO (x16).
- Прерывания/события по специфическим причинам (x16).

Внешние прерывания:

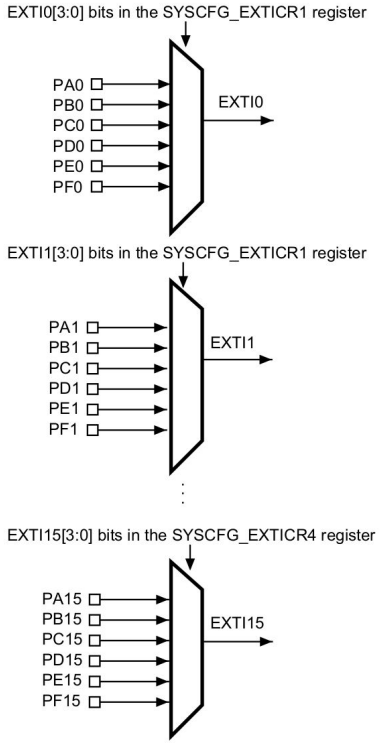
- По спадающему и/или нарастающему фронту (falling/rising edge).
- Бывают замаскированы.
- Требуют сброса флага Pending.

Каждая линия EXTI:

- Может быть настроена как прерывание/событие
  (для работы в running/stop mode)

Линии прерываний EXTI

Figure 24. External interrupt/event GPIO mapping

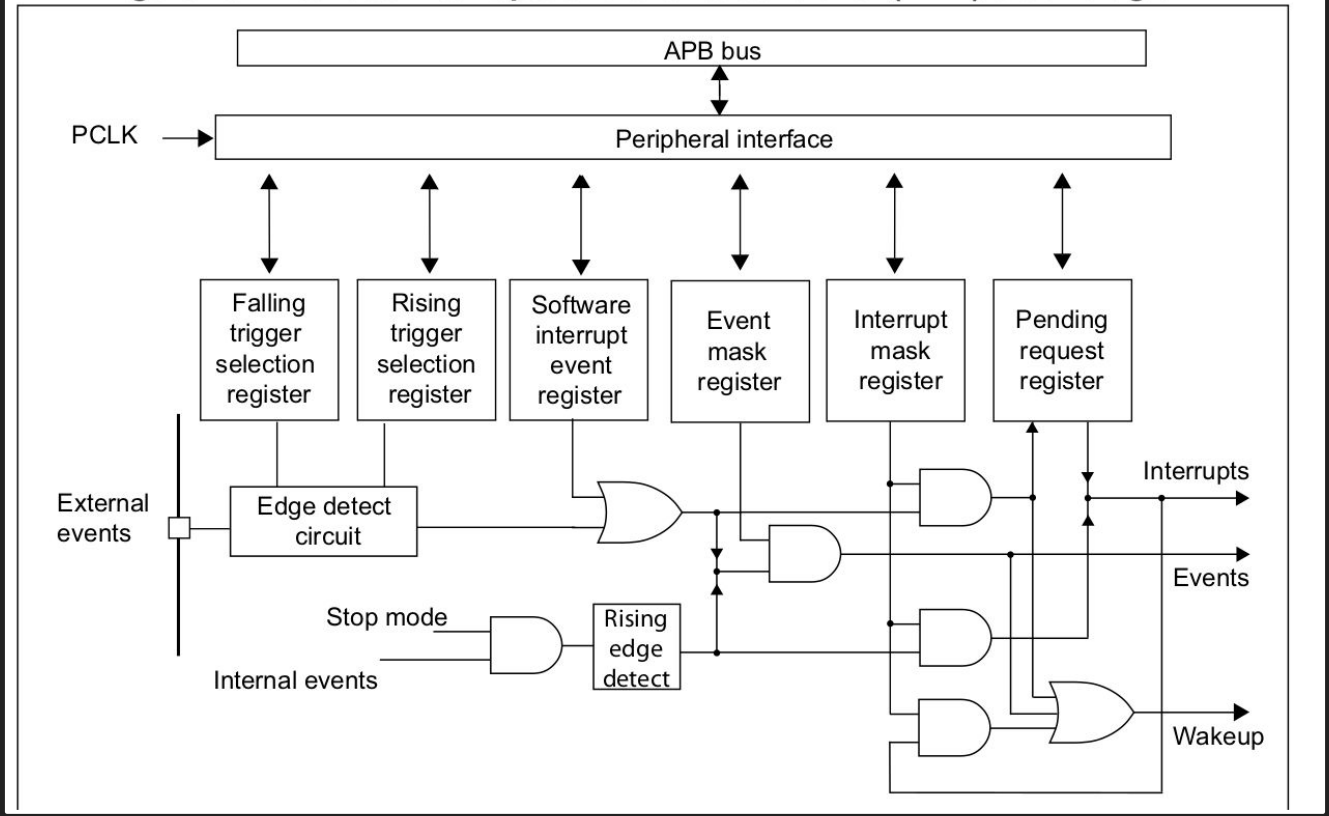


The remaining lines are connected as follow:

- EXTI line 16 is connected to the PVD output
- EXTI line 17 is connected to the RTC Alarm event
- EXTI line 18 is connected to the internal USB wakeup event
- EXTI line 19 is connected to the RTC Tamper and TimeStamp events
- EXTI line 20 is connected to the RTC Wakeup event (available only on STM32F07x and STM32F09x devices)
- EXTI line 21 is connected to the Comparator 1 output
- EXTI line 22 is connected to the Comparator 2 output
- EXTI line 23 is connected to the internal I2C1 wakeup event
- EXTI line 24 is reserved (internally held low)
- EXTI line 25 is connected to the internal USART1 wakeup event
- EXTI line 26 is connected to the internal USART2 wakeup event (available only on STM32F07x and STM32F09x devices)
- EXTI line 27 is connected to the internal CEC wakeup event
- EXTI line 28 is connected to the internal USART3 wakeup event (available only on STM32F09x devices)
- EXTI line 29 is reserved (internally held low)
- EXTI line 30 is reserved (internally held low)
- EXTI line 31 is connected to the V_{DDIO2} supply comparator output (available only on STM32F04x, STM32F07x and STM32F09x devices

Контроллер внешних прерываний

Figure 23. Extended interrupts and events controller (EXTI) block diagram



Внешние прерывания: настройка EXTI

Пример из репозитория Эдгара (labs/06_exti_systick):

```
static void exti_config(void)
{
    LL_APB1_GRP2_EnableClock(LL_APB1_GRP2_PERIPH_SYSCFG);

    LL_SYSCFG_SetEXTISource(LL_SYSCFG_EXTI_PORTA, LL_SYSCFG_EXTI_LINE1);
    LL_SYSCFG_SetEXTISource(LL_SYSCFG_EXTI_PORTA, LL_SYSCFG_EXTI_LINE0);
    LL_EXTI_EnableIT_0_31(LL_EXTI_LINE_1);
    LL_EXTI_EnableIT_0_31(LL_EXTI_LINE_0);

    LL_EXTI_EnableFallingTrig_0_31(LL_EXTI_LINE_1);
    LL_EXTI_EnableRisingTrig_0_31(LL_EXTI_LINE_1);

    LL_EXTI_EnableFallingTrig_0_31(LL_EXTI_LINE_0);
    LL_EXTI_EnableRisingTrig_0_31(LL_EXTI_LINE_0);
    /*
     * Setting interrupts
     */
    NVIC_EnableIRQ(EXTI0_1_IRQn);
    NVIC_SetPriority(EXTI0_1_IRQn, 0);
}
```

Вектор исключений и сброс флагов

```
.word   FLASH_IRQHandler                /* FLASH
.word   RCC_CRS_IRQHandler              /* RCC and CRS
.word   EXTI0_1_IRQHandler              /* EXTI Line 0 and 1
.word   EXTI2_3_IRQHandler              /* EXTI Line 2 and 3
.word   EXTI4_15_IRQHandler             /* EXTI Line 4 to 15
.word   TSC_IRQHandler                  /* TSC
.word   DMA1_Channel1_IRQHandler        /* DMA1 Channel 1
```

```
/*
 * don't forget to reset flags
 */
LL_EXTI_ClearFlag_0_31(LL_EXTI_LINE_1);
LL_EXTI_ClearFlag_0_31(LL_EXTI_LINE_0);
```

Таймеры в STM32F051

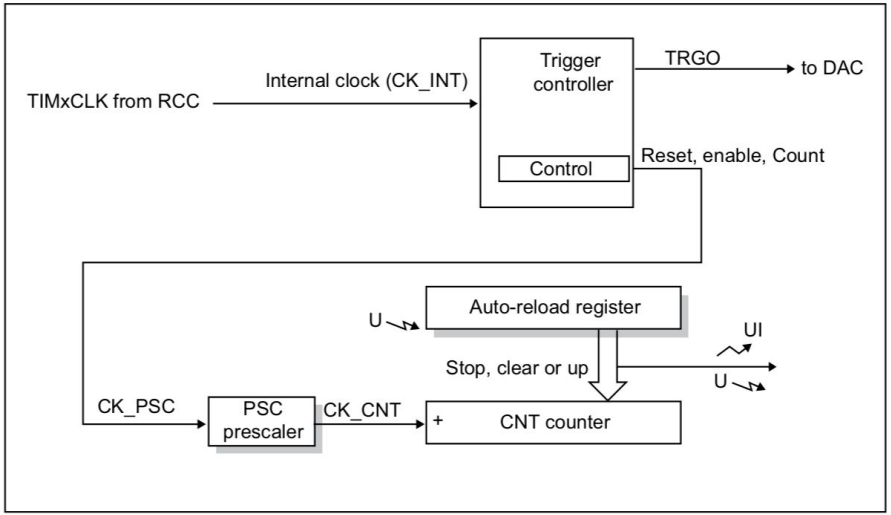
Таймеры в STM32F051

Table 7. Timer feature comparison

Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/compare channels	Complementary outputs
Advanced control	TIM1	16-bit	Up, down, up/down	integer from 1 to 65536	Yes	4	3
General purpose	TIM2	32-bit	Up, down, up/down	integer from 1 to 65536	Yes	4	-
	TIM3	16-bit	Up, down, up/down	integer from 1 to 65536	Yes	4	-
	TIM14	16-bit	Up	integer from 1 to 65536	No	1	-
	TIM15	16-bit	Up	integer from 1 to 65536	Yes	2	1
	TIM16 TIM17	16-bit	Up	integer from 1 to 65536	Yes	1	1
Basic	TIM6	16-bit	Up	integer from 1 to 65536	Yes	-	-

Устройство TIM6: уровень Basic

Figure 197. Basic timer block diagram



Notes:

Reg — Preload registers transferred to active registers on U event according to control bit

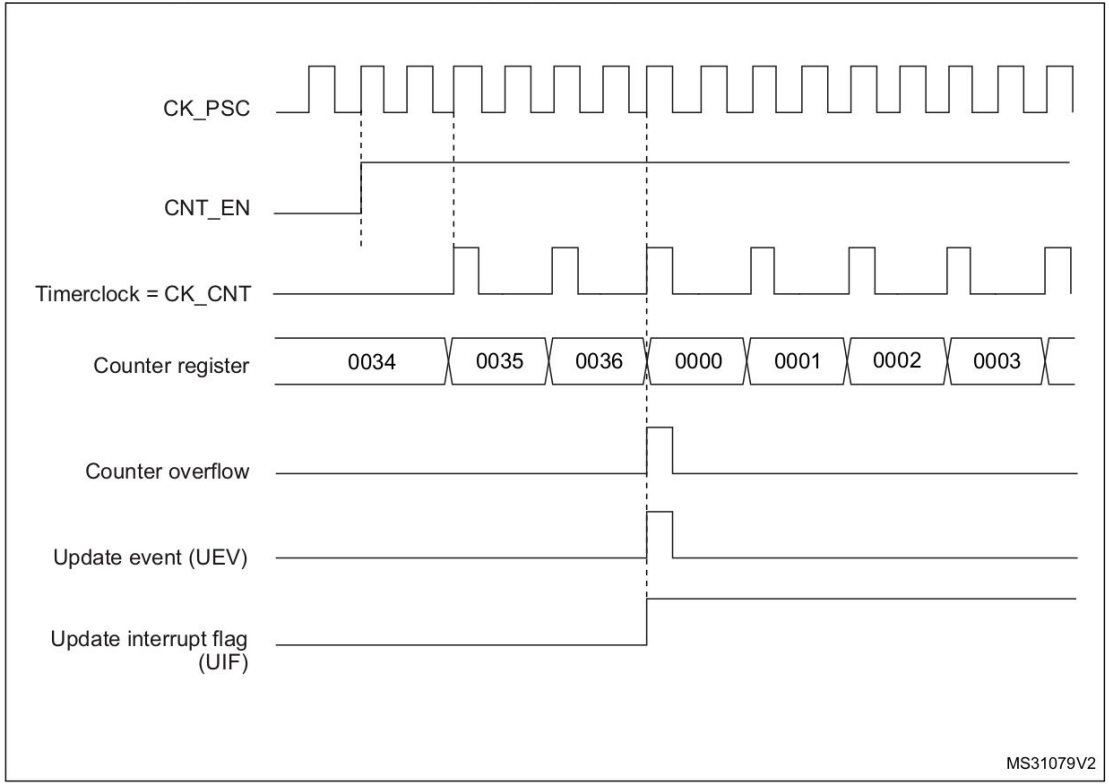
Event

Interrupt & DMA output

MS33142V1

Устройство TIM6: работа счётчика

Figure 201. Counter timing diagram, internal clock divided by 2



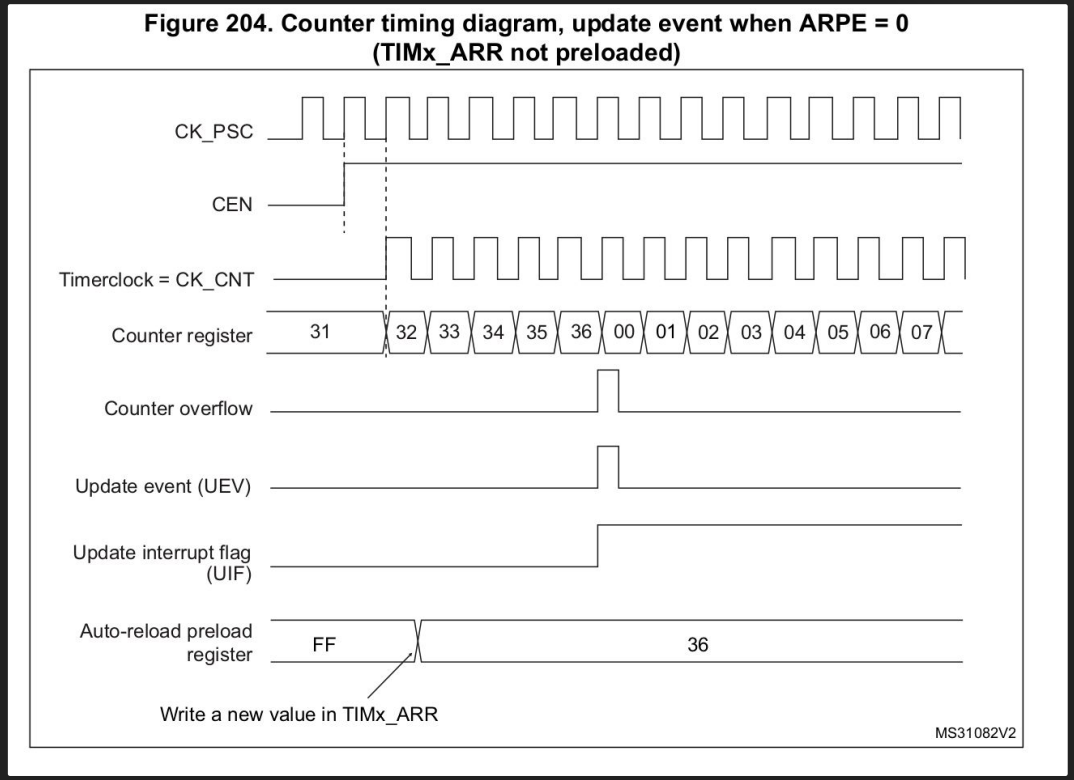
Устройство TIM6: без предзагрузки ARR

Счётчик можно “сломать”!

Пример: обновление периода сигнала на более низкий:

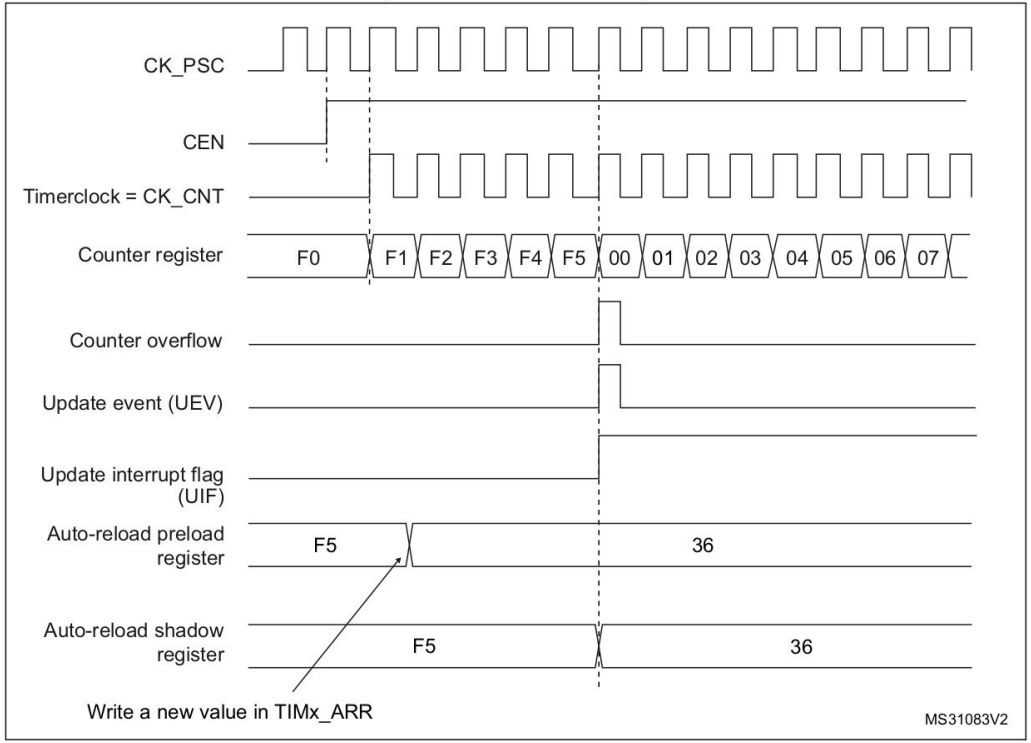
`CNT = 240 -> CNT = 241`

`ARR = 250 -> ARR = 230`



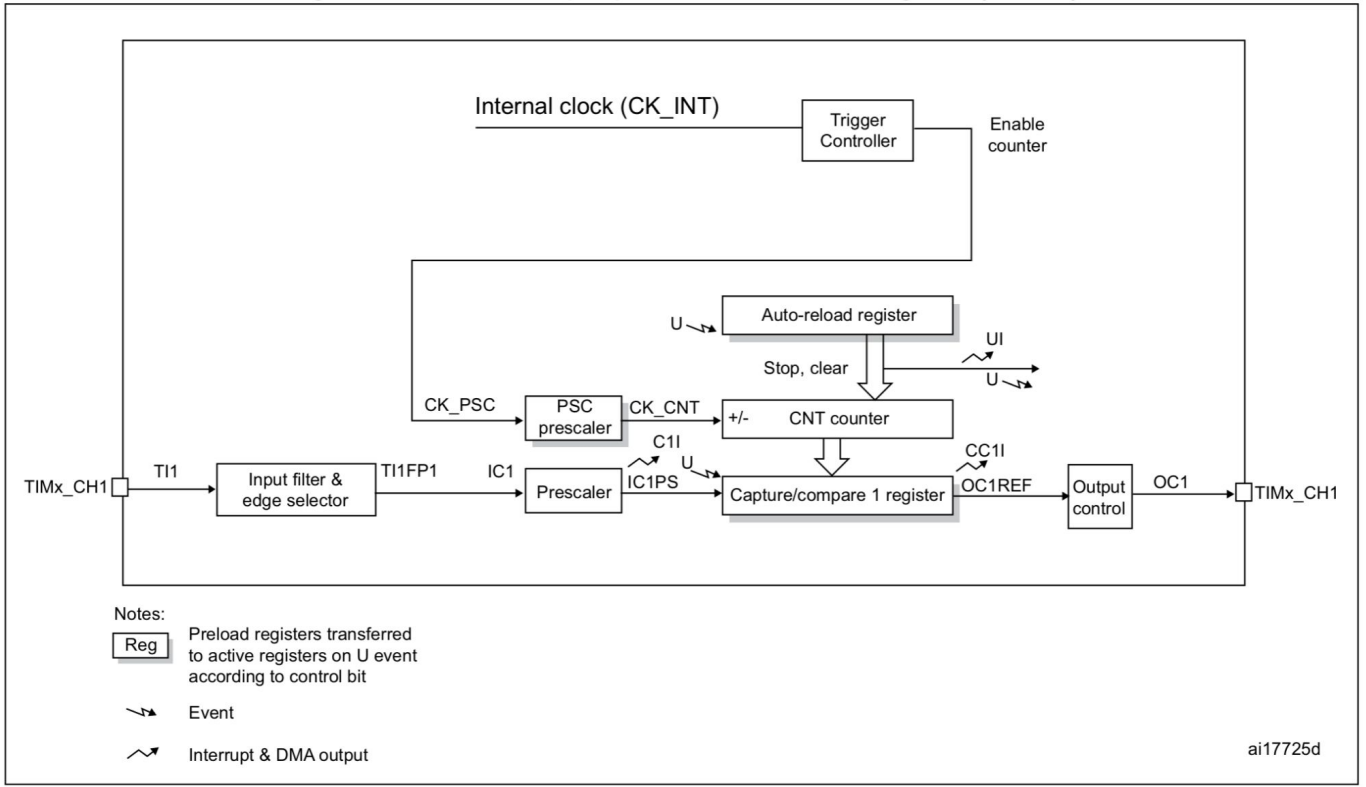
Устройство TIM6: предзагрузка ARR

Figure 205. Counter timing diagram, update event when ARPE=1 (TIMx_ARR preloaded)



Устройство TIM14: уровень General Purpose

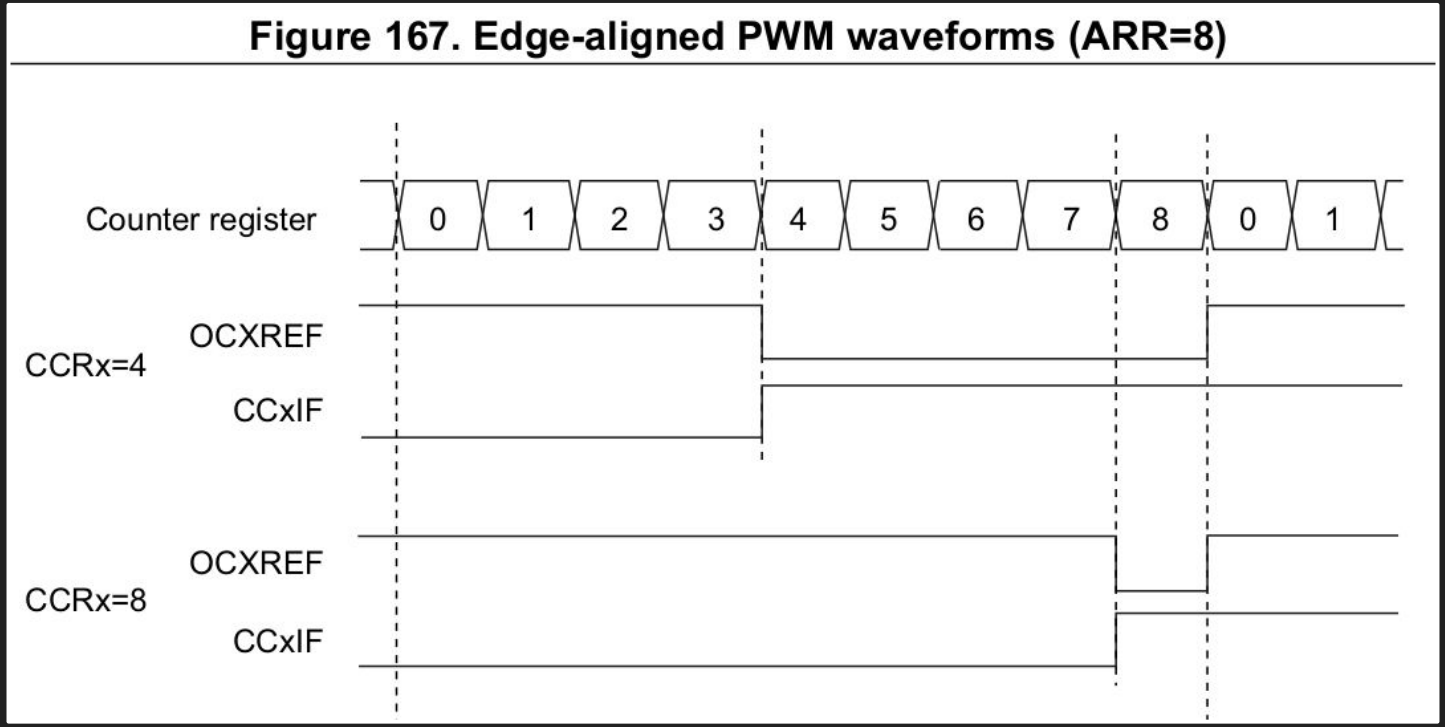
Figure 153. General-purpose timer block diagram (TIM14)



Устройство TIM14: режим по сравнению

Генерация ШИМ-сигнала!

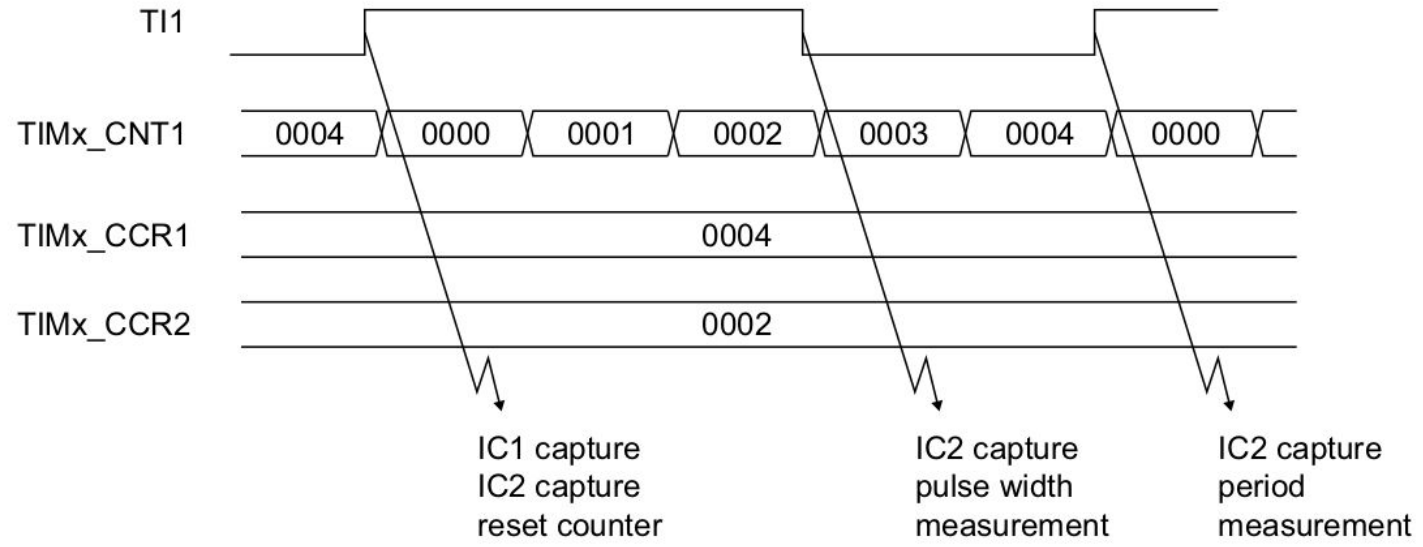
Figure 167. Edge-aligned PWM waveforms (ARR=8)



Устройство TIM14: режим по захвату

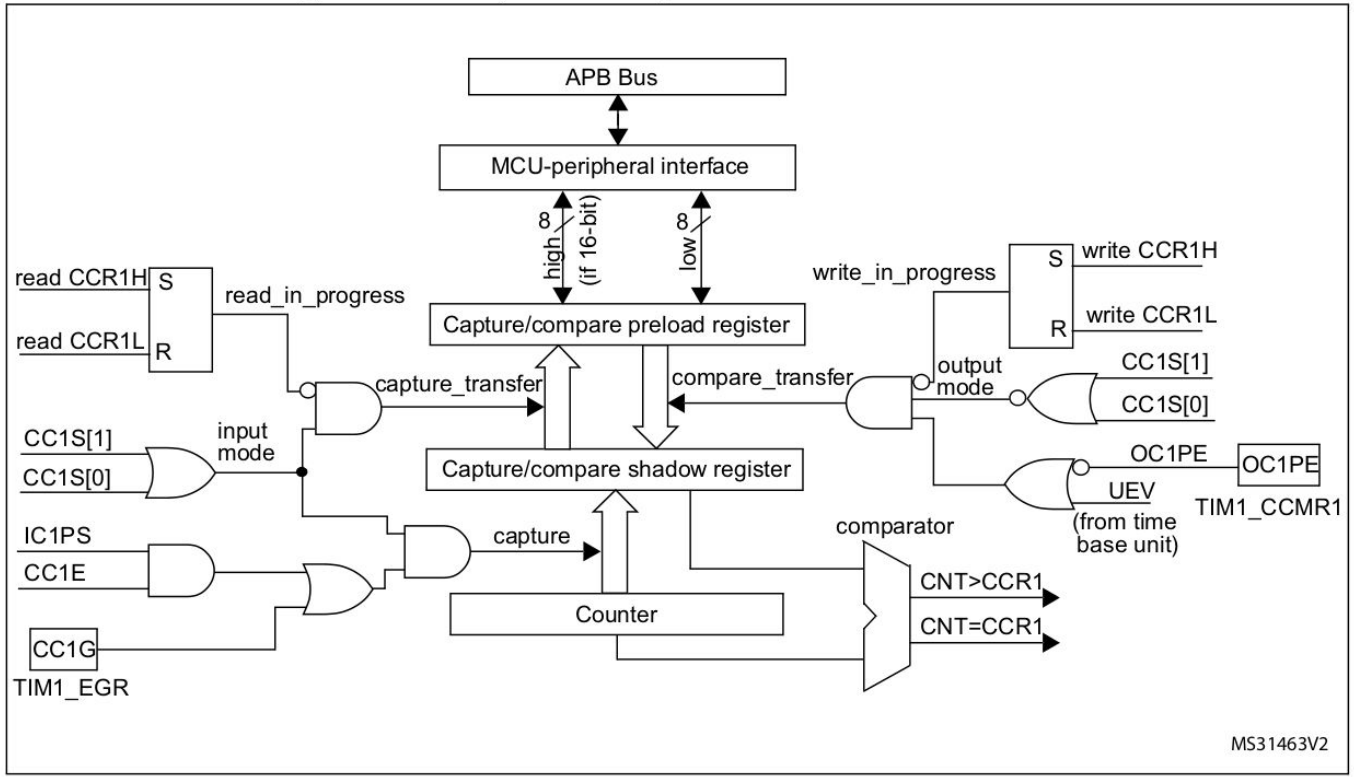
Измерение длительности импульса и периода сигнала!

Figure 186. PWM input mode timing



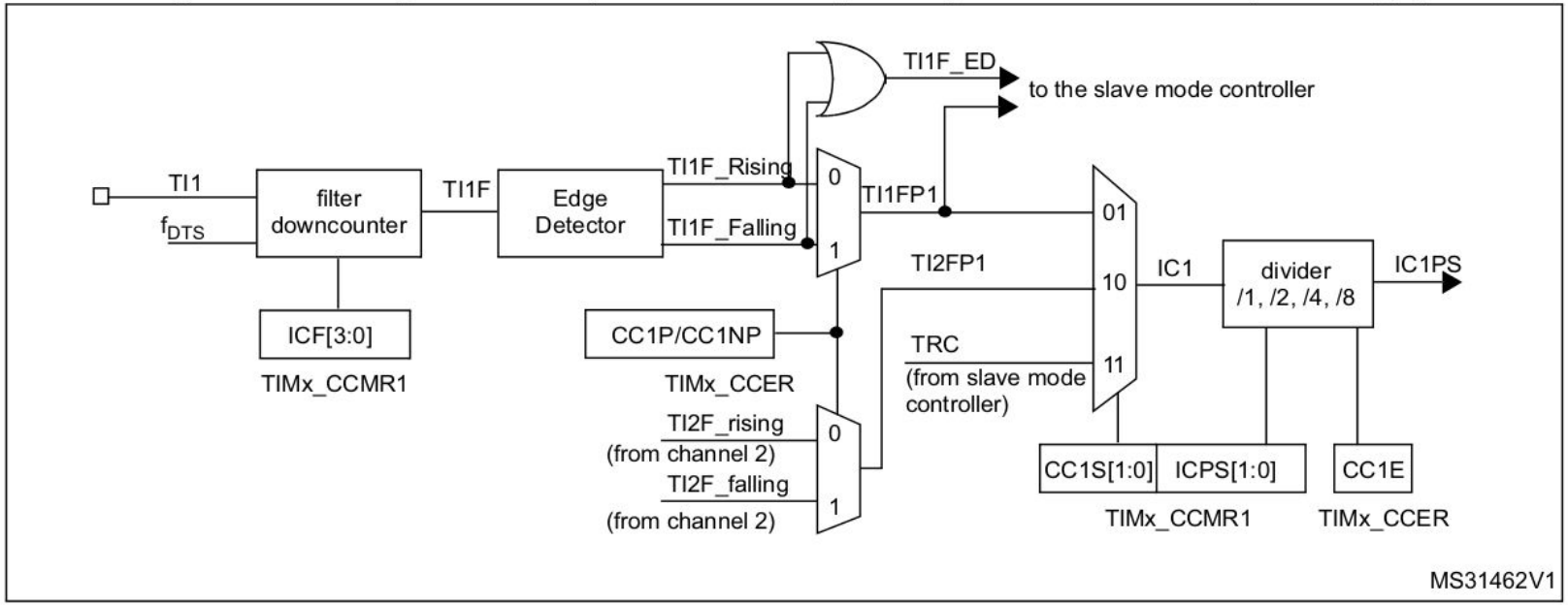
Устройство TIM14: схема захвата/сравнения

Figure 164. Capture/compare channel 1 main circuit



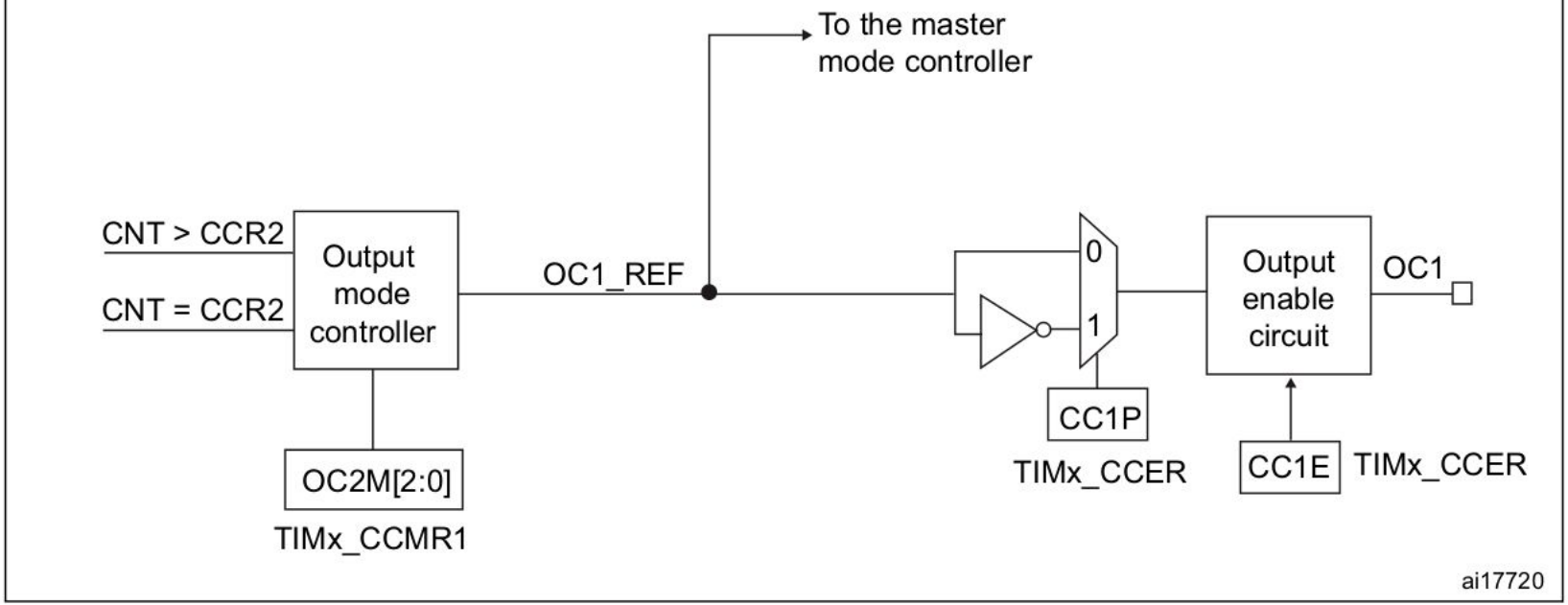
Устройство TIM14: схема захвата

Figure 163. Capture/compare channel (example: channel 1 input stage)



Устройство TIM14: схема сравнения

Figure 165. Output stage of capture/compare channel (channel 1)

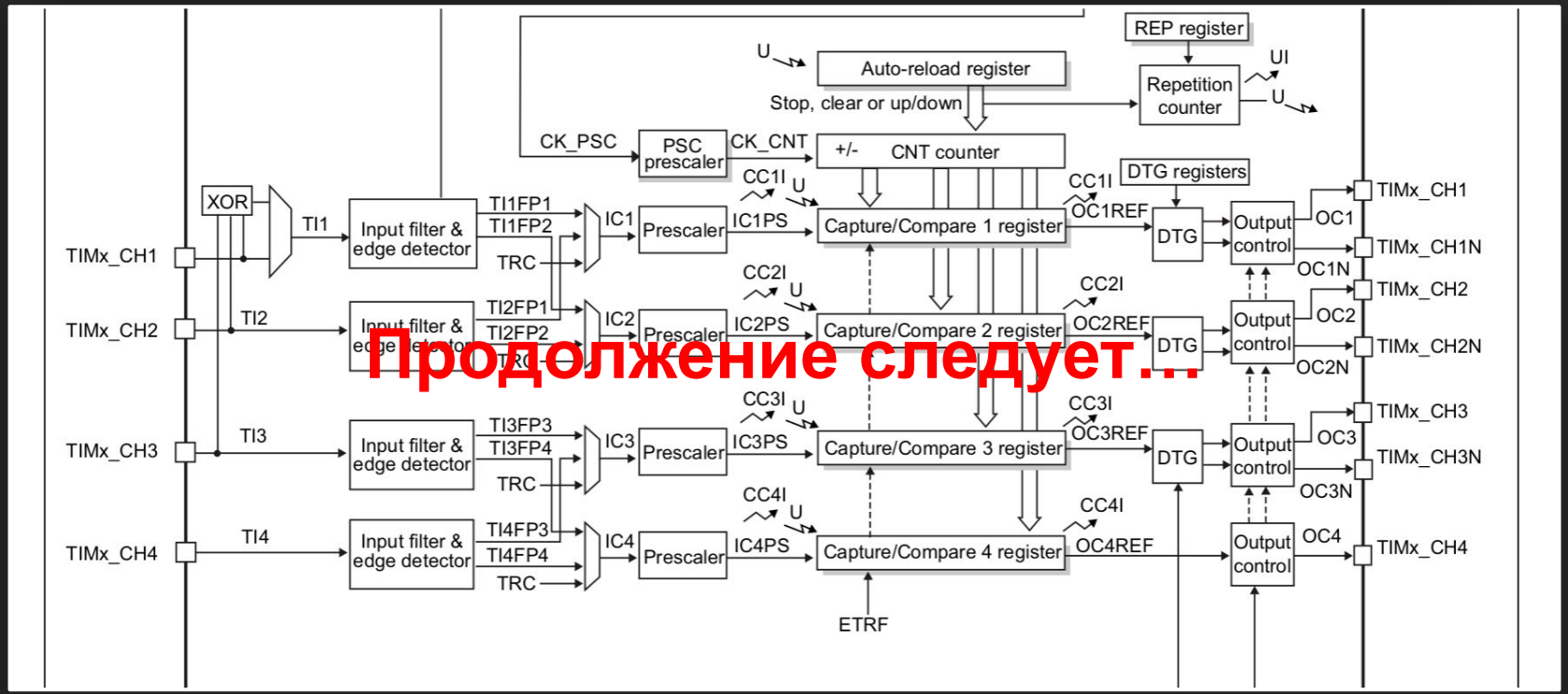


Устройство TIM1: уровень Advanced

Figure 59. Advanced-control timer block diagram

Продолжение следует…

Устройство TIM1: уровень Advanced



Таймеры в STM32F051: примеры кода

Настройка TIM2 в репо Эдгара Казиахмедова (labs/07_timers_counter):

```
static void timers_config(void)
{
    /*
     * Setup timer
     */
    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM2);
    LL_TIM_SetPrescaler(TIM2, 47999);
    LL_TIM_SetAutoReload(TIM2, 999);
    LL_TIM_SetCounterMode(TIM2, LL_TIM_COUNTERMODE_UP);
    LL_TIM_EnableIT_UPDATE(TIM2);
    LL_TIM_EnableCounter(TIM2);
    /*
     * Setup NVIC
     */
    NVIC_EnableIRQ(TIM2_IRQn);
    NVIC_SetPriority(TIM2_IRQn, 0);
    return;
}
```

```
void TIM2_IRQHandler(void)
{
    LL_GPIO_TogglePin(GPIOC, LL_GPIO_PIN_8);
    LL_TIM_ClearFlag_UPDATE(TIM2);
}
```

```
.word   ADC1_COMP_IRQHandler
.word   TIM1_BRK_UP_TRG_COM_IRQHandler
.word   TIM1_CC_IRQHandler
.word   TIM2_IRQHandler
.word   TIM3_IRQHandler
.word   TIM6_DAC_IRQHandler
```

См. также - TIM2 в режиме по сравнению (labs/07_timers_inp_capture).

Спасибо за внимание!

Красивые иконки взяты с сайта handdrawngoods.com