TAPiT

KHÓA HỌC
LẬP TRÌNH VI ĐIỀU KHIỂN

Giảng viên
NGUYỄN HUỲNH NHẬT THƯƠNG

LỊCH HỌC:
19h30 - 22h00 thứ 2 và thứ 6

ĐỊA ĐIỂM:
Online qua nền tảng Zoom / Google Meet

MODULE 4

- TIMER - TIME BASE
- THỰC HÀNH, DEBUG TIME BASE
- PULSE WIDTH MODULATION

0981001119 tapitlrs@gmail.com https://tapit.vn

TYPES OF TIMERS

Advanced timers (TIM1, TIM8):

- Three timers dedicated to motor control
- The four independent channels can be used for:

- Input capture
- Output compare
- PWM generation
- One-pulse mode output

TYPES OF TIMERS

General-purpose timers (TIM2, TIM3, TIM4, TIM15, TIM16, TIM17):

- TIM2, 3, and TIM4 (full-featured general-purpose timers)

- TIM2 has a **32-bit** auto-reload up/down **counter** and 32-bit **prescaler**

- TIM3 and 4 have 16-bit auto-reload up/down **counters** and 16-bit **prescalers.**

- 4 independent channels for input capture/output compare, PWM or one-pulse mode output.

- TIM15, 16 and 17 (general-purpose timers)

- 16-bit auto-reload upcounters and 16-bit prescalers.

- TIM15 has 2 channels and 1 complementary channel

- TIM16 and TIM17 have 1 channel and 1 complementary channel

- 4 independent channels for input capture/output compare, PWM or one-pulse mode output.

- **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

TYPES OF TIMERS

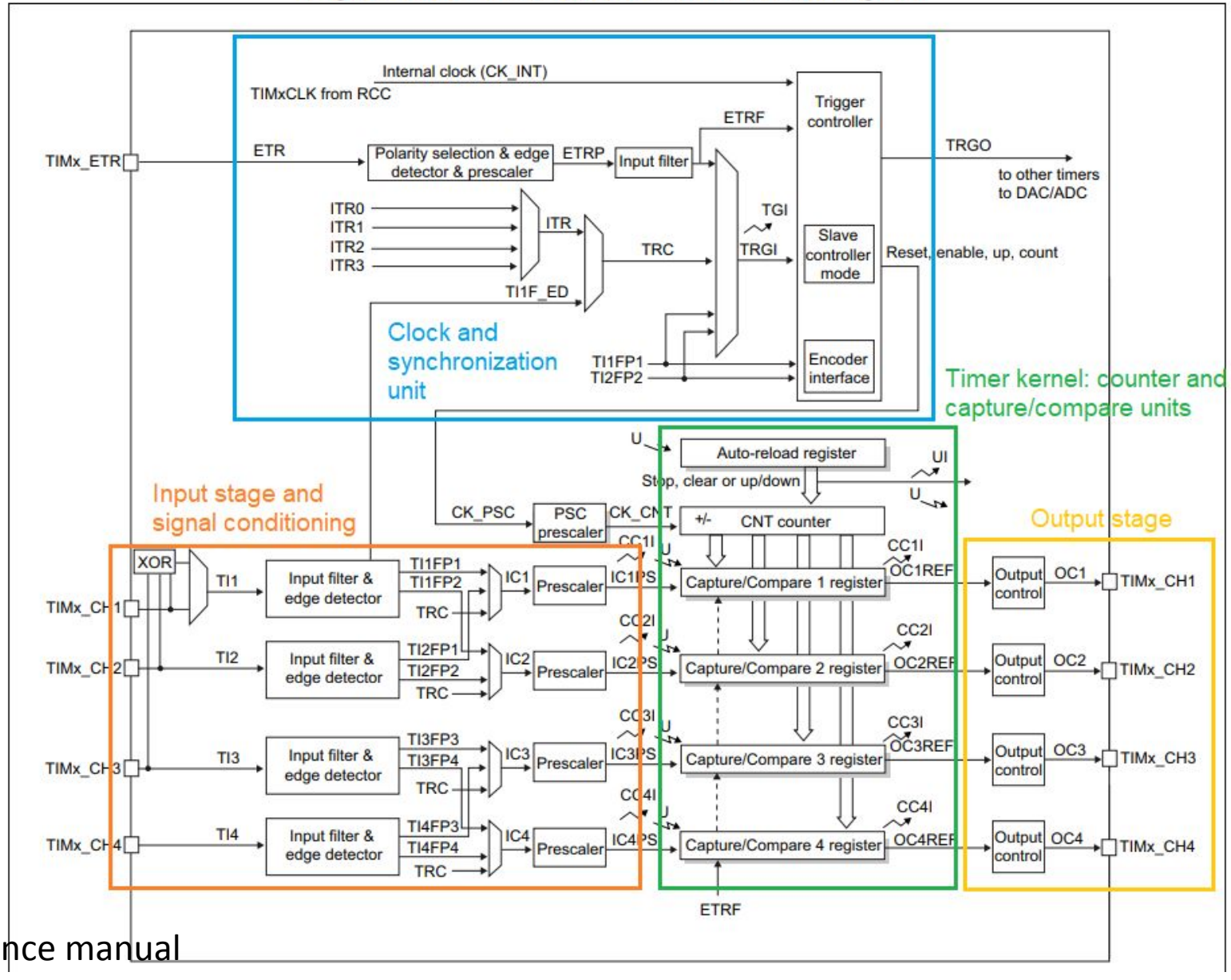
Basic timers (TIM6, TIM7):

These timers are mainly used for DAC trigger generation. They can also be used as a generic 16-bit time base.

- 16-bit auto-reload upcounter
- 16-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65535
- Interrupt/DMA generation on the update event: counter overflow

GENERAL-PURPOSE TIMERS

Figure 197. General-purpose timer block diagram



Reference manual

TIME-BASE UNIT

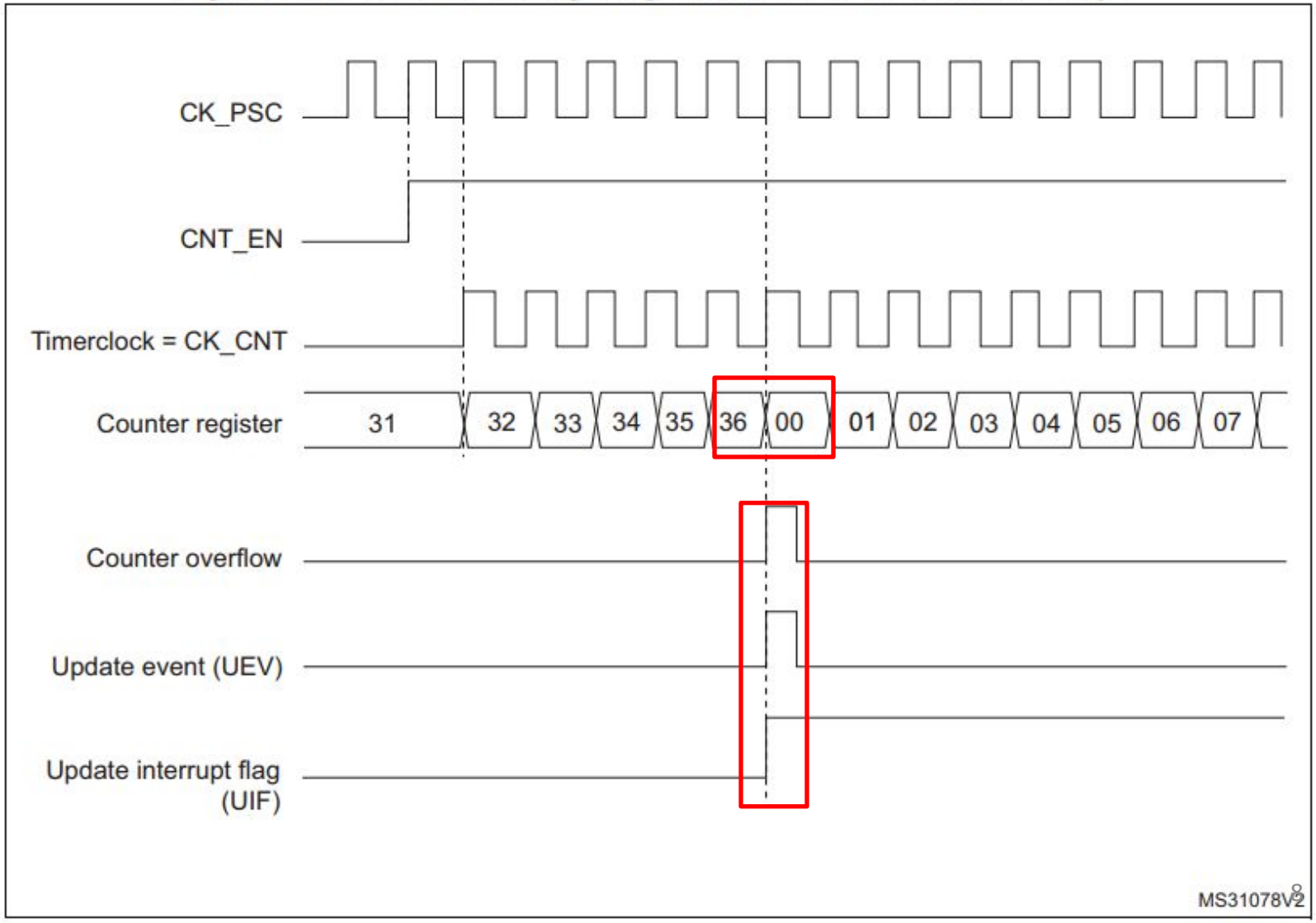
- The main block of the programmable timer is a 16-bit/32-bit counter (CNT) with its related autoreload register (ARR). The counter can count up, down or both up and down but also down or both up and down. The counter clock can be divided by a prescaler.
- The counter, the auto-reload register and the prescaler register can be written or read by software. This is true even when the counter is running.
- The time-base unit includes:
  - Counter Register (TIMx_CNT)
  - Prescaler Register (TIMx_PSC)
  - Auto-Reload Register (TIMx_ARR)

COUNTER MODES

- **Upcounting mode**
- Downcounting mode
- Center-aligned mode (up/down counting)

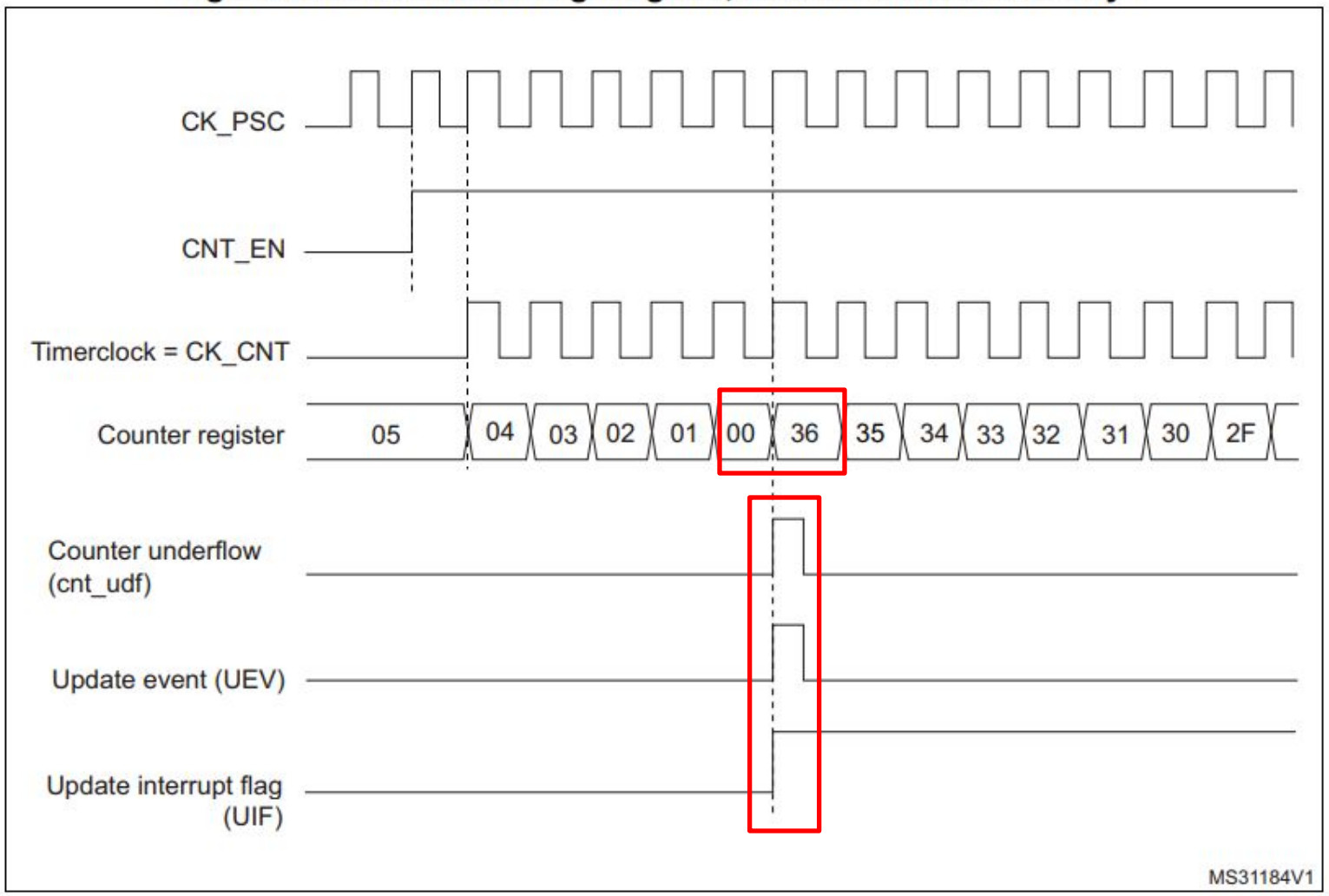
UPCOUNTING MODE

Figure 200. Counter timing diagram, internal clock divided by 1



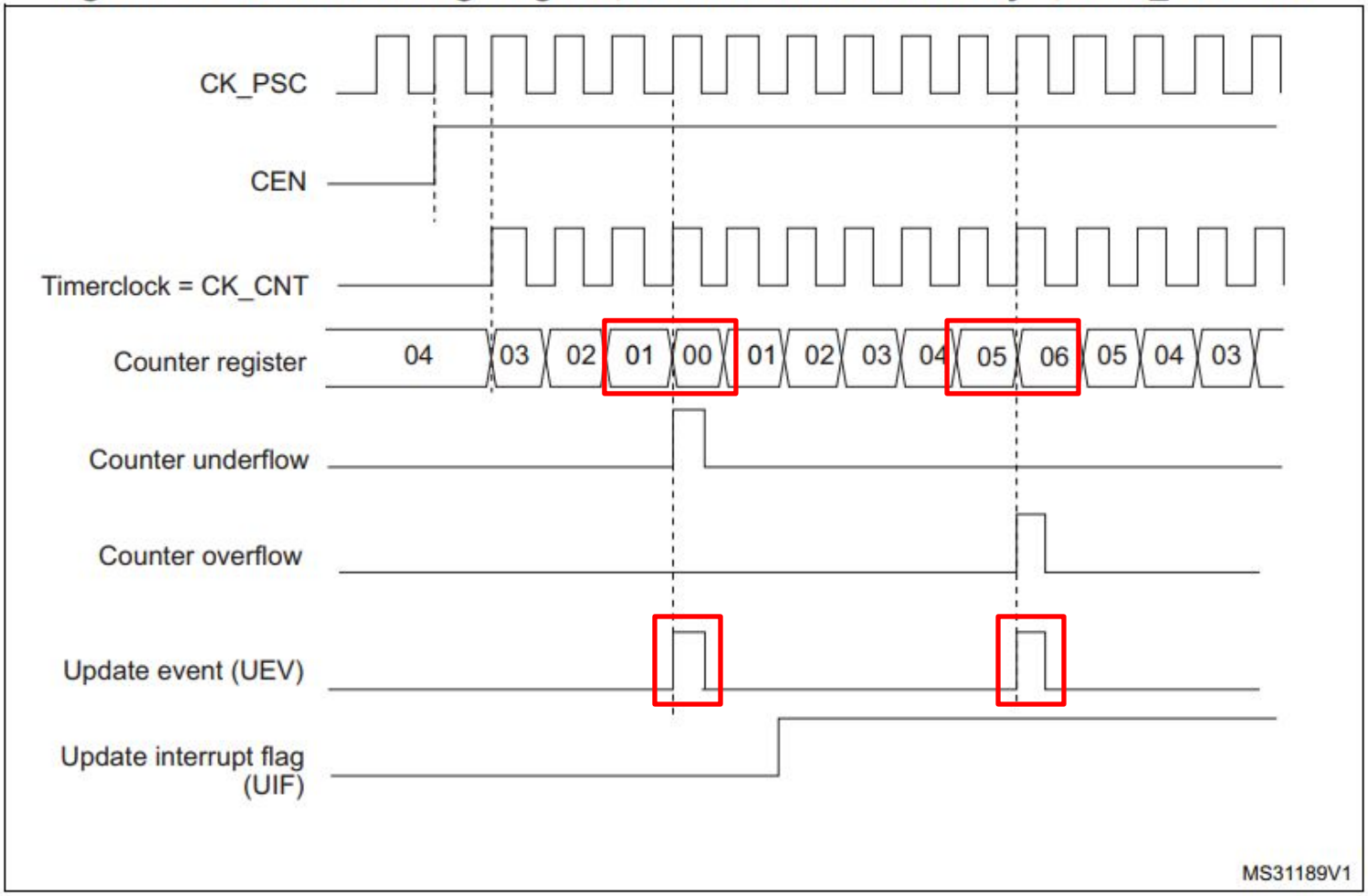
DOWNCOUNTING MODE

Figure 206. Counter timing diagram, internal clock divided by 1



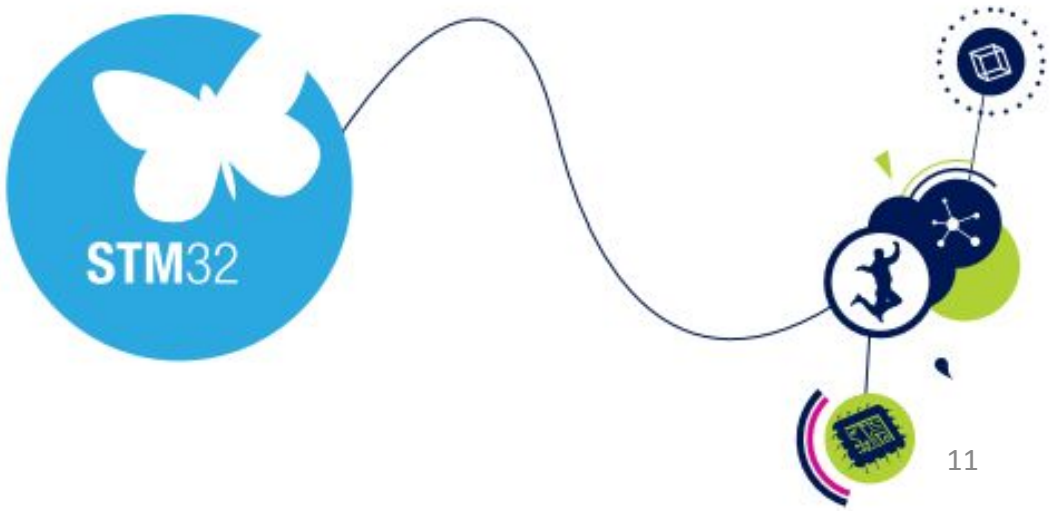
CENTER-ALIGNED MODE

Figure 211. Counter timing diagram, internal clock divided by 1, TIMx_ARR=0x6



MODULE 4

- THỰC HÀNH, DEBUG TIME BASE

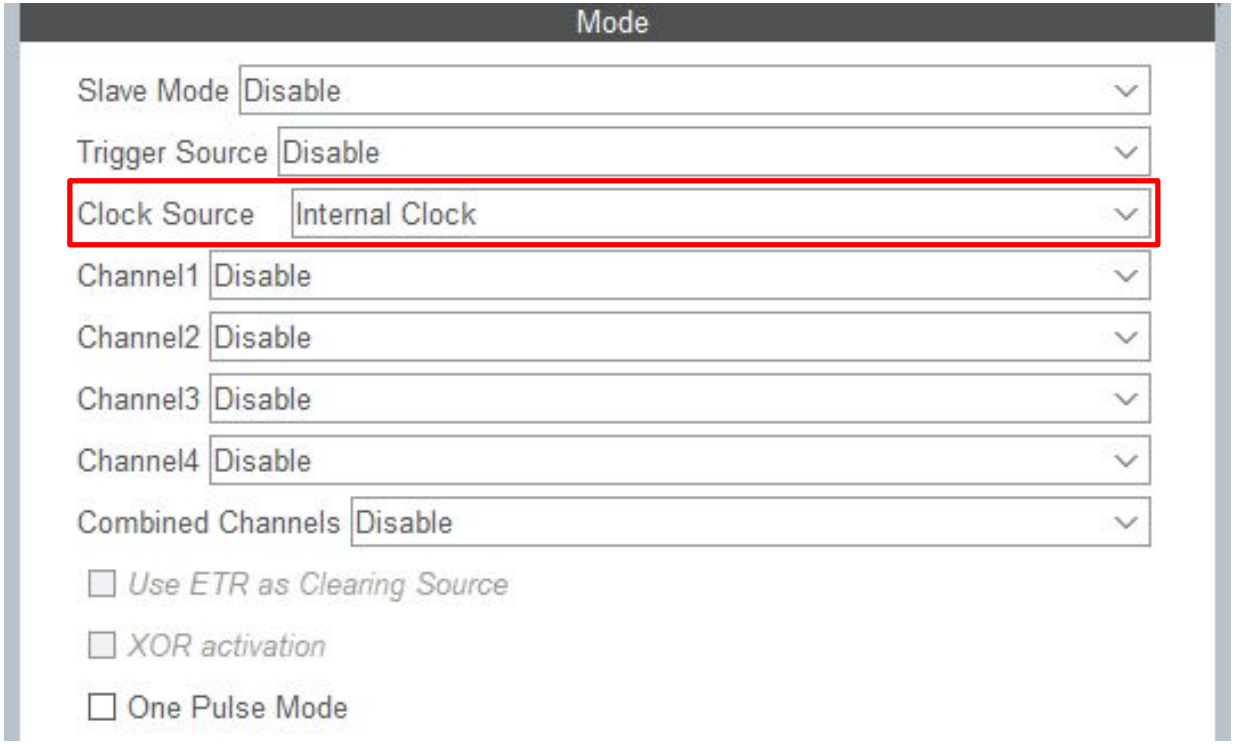


CLOCK SELECTION

The counter clock can be provided by the following clock sources:

- **Internal clock (CK_INT)** – RCC
- **External clock mode1**: external input pin (TIx) - The counter can count at each rising or falling edge on a selected input.
- **External clock mode2:** external trigger input (ETR) - The counter can count at each rising or falling edge on the external trigger input ETR.
- **Internal trigger inputs (ITRx):** using one timer as prescaler for another timer, for example, you can configure Timer 13 to act as a prescaler for Timer 2.

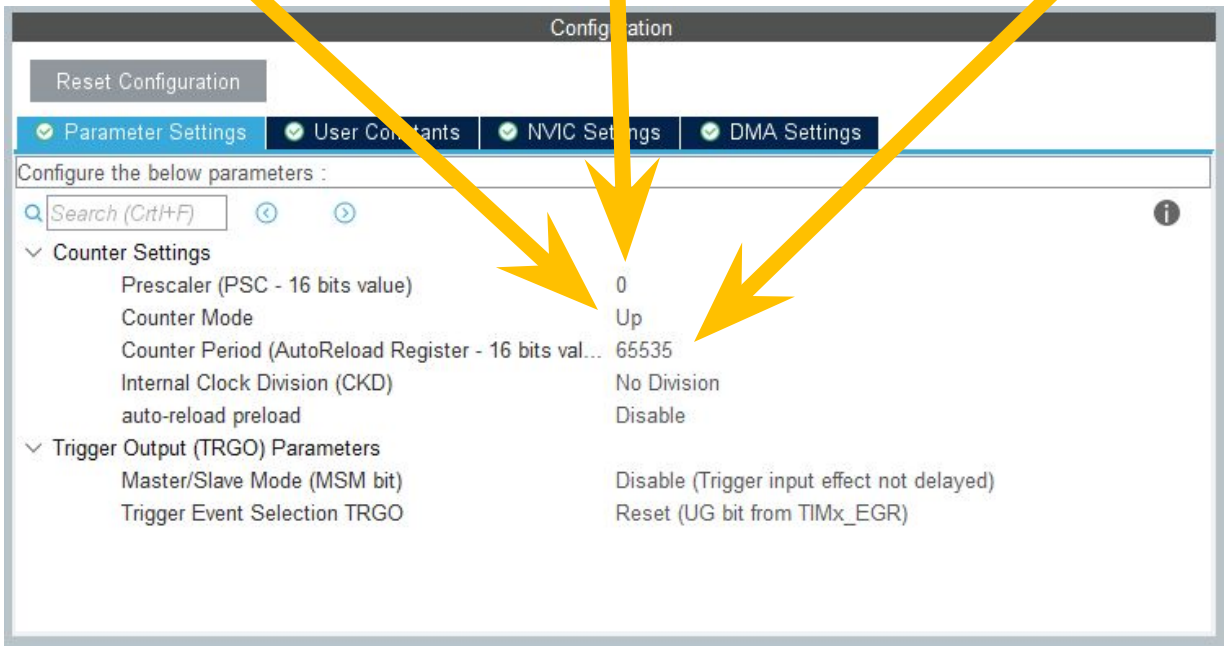
TIME-BASE UNIT



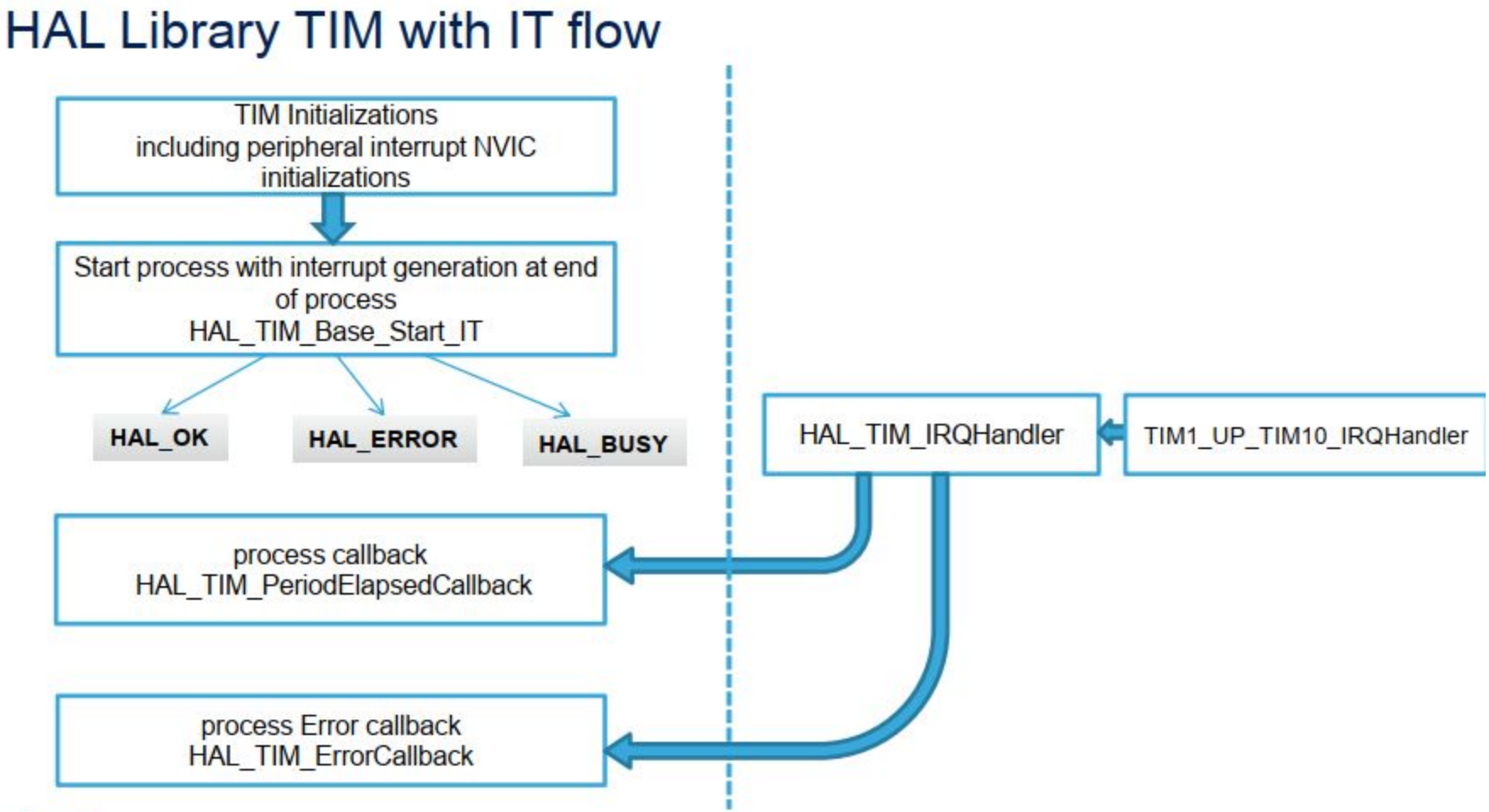
TIME-BASE UNIT

The time-base unit includes:

TIMx_CNT **TIMx_PSC** **TIMx_ARR**



Use TIM with interrupt



Use TIM with interrupt

For TIM start use function

•HAL_TIM_Base_Start_IT(TIM_HandleTypeDef *htim)

For TIM stop use function

•HAL_TIM_Base_Stop_IT(TIM_HandleTypeDef *htim)

TIM callback

- HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)

```
//TIM start
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(&htim1);
/* USER CODE END 2 */

//Callback handling
/* USER CODE BEGIN 4 */
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
      //do something…
}
/* USER CODE END 4 */
```

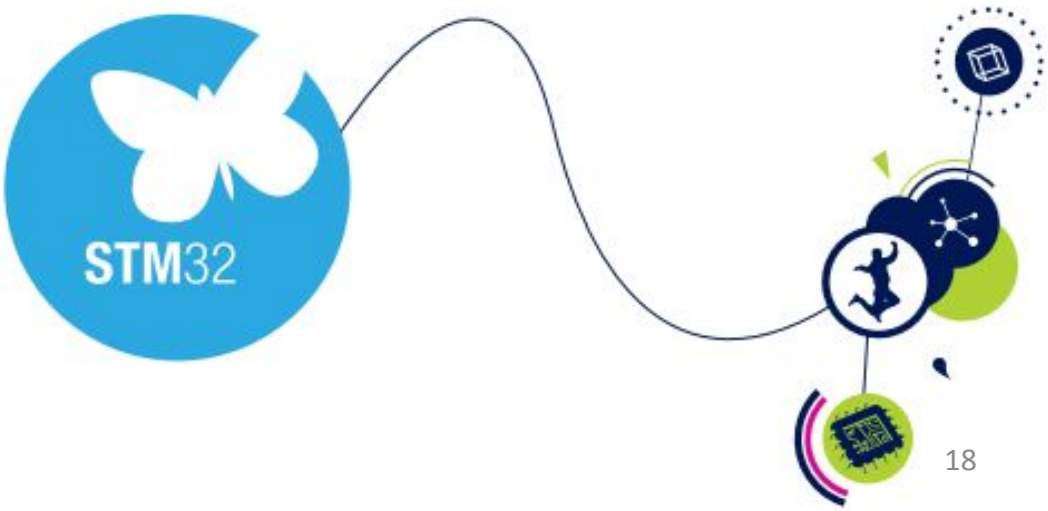
Lưu ý

MX_TIM2_Init(); // Khi gọi hàm INIT thì các giá trị thanh ghi sẽ được update -> cờ update bật lên.

```
  /* USER CODE BEGIN 2 */
  __HAL_TIM_CLEAR_IT(&htim2, TIM_IT_UPDATE); // Xóa cờ update đi.
  HAL_TIM_Base_Start_IT(&htim2);    // UIE , cho phép ngắt theo sự kiện udpate
  /* USER CODE END 2 */
```

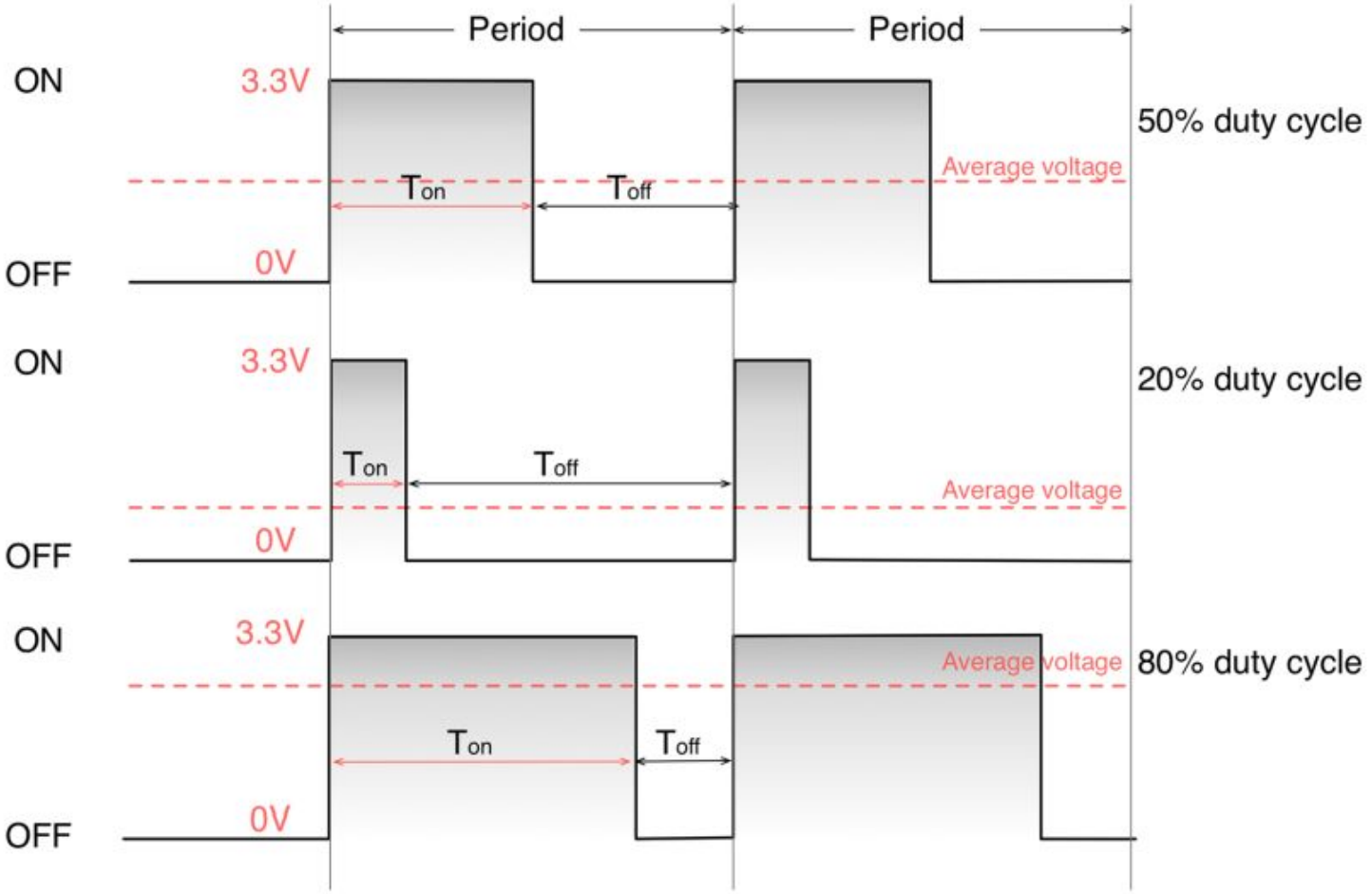
MODULE 5

- PULSE WIDTH MODULATION



Duty Cycle

A duty cycle is expressed as:



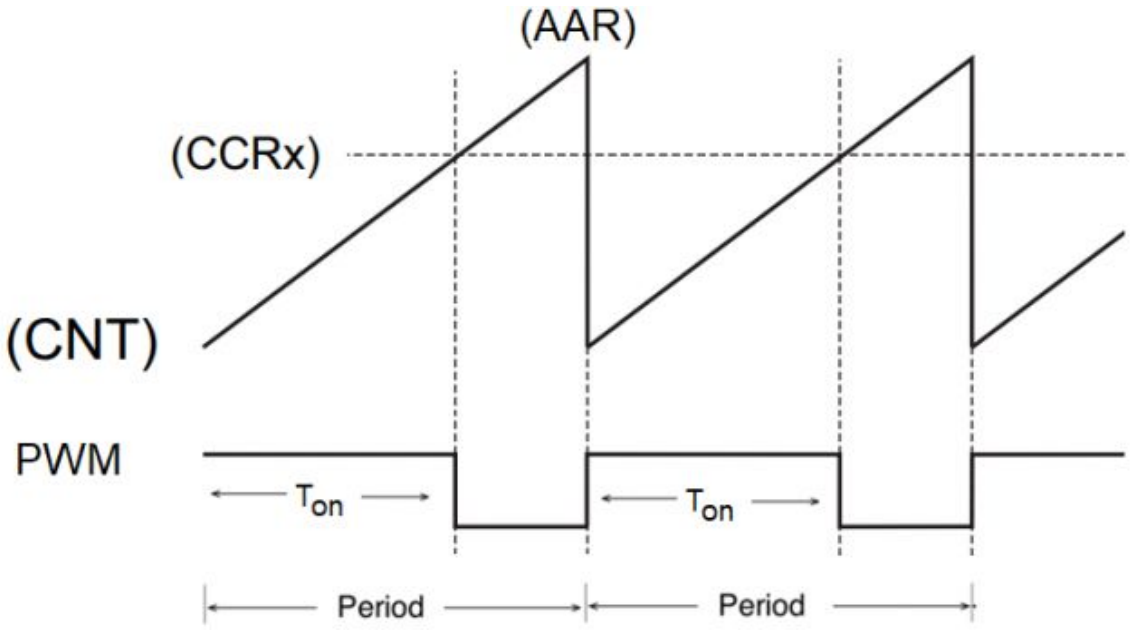
where D is the duty cycle, TON is the time the signal is active

Điều chế độ rộng xung PWM là làm những gì?

- Cài đặt tần số xung (chu kỳ xung) ARR, clock counter
- Cài đặt duty cycles -> CCR so với ARR
- Thay đổi tần số xung (chu kỳ xung), ARR, clock counter
- Thay đổi duty cycles -> CCR

PWM mode

Pulse width modulation mode allows you to generate a signal with a frequency (period) determined by the value of the TIMx_ARR register and a duty cycle determined by the value of the TIMx_CCRx register.



$$D = \frac{T_{ON}}{Period} \times 100\%$$

PWM applications

PWM has many applications in digital electronics, but all of them can be grouped in two main categories:

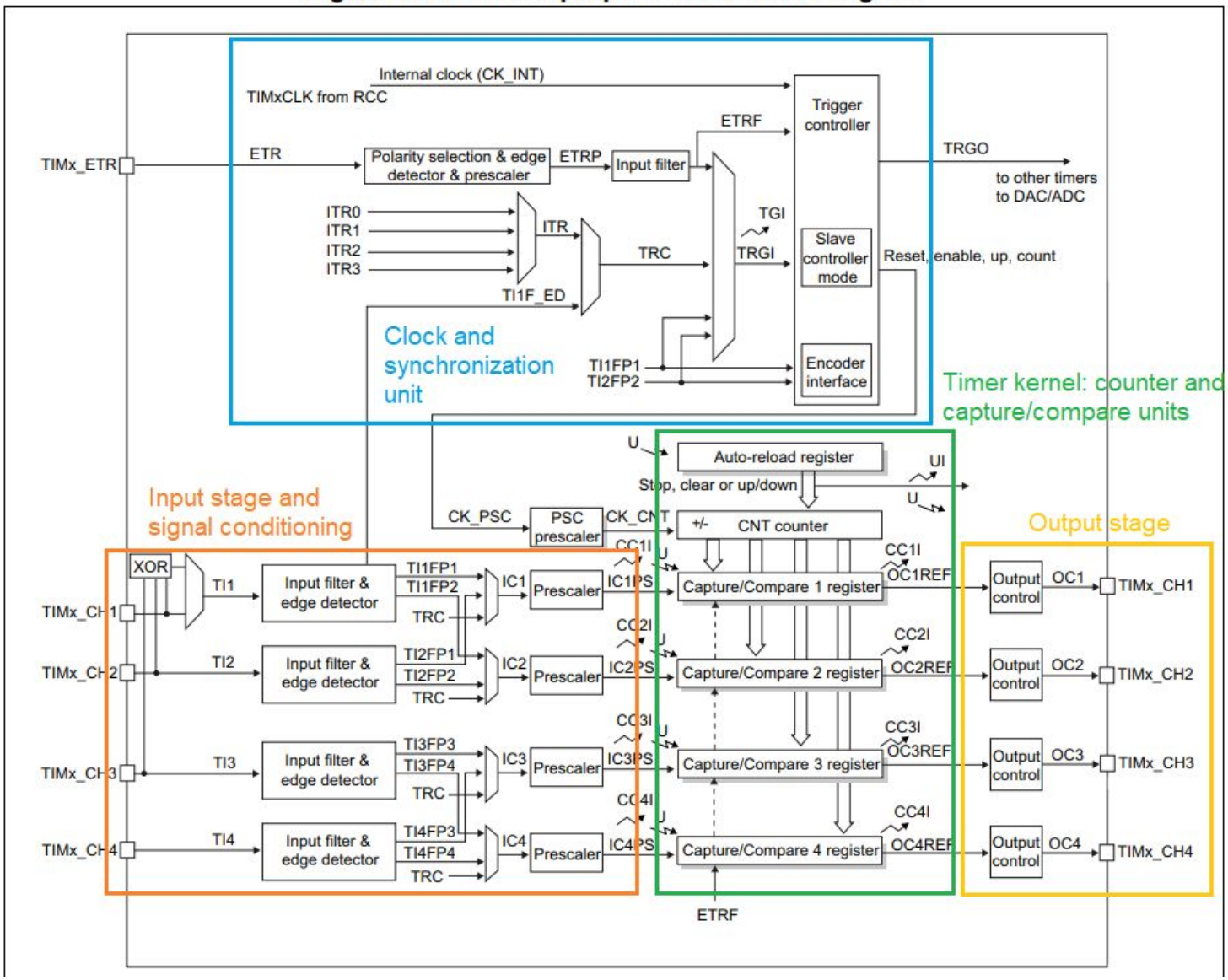
- control the output voltage (and hence the current);
- encoding (that is, modulate) a message (that is, a series of bytes in digital electronics) on a carrier wave (which runs at a given frequency).

Several applications on the control of the output voltage :

- Generation of an output voltage ranging from 0V up to VDD (that is, the maximum allowed voltage for an I/O, which in an STM32 is 3.3V);
  - dimming of LEDs;
  - DC motor control;
  - power conversion;
- Generation of an output wave running at a given frequency (sine wave, triangle, square, and so on);
- Sound output;

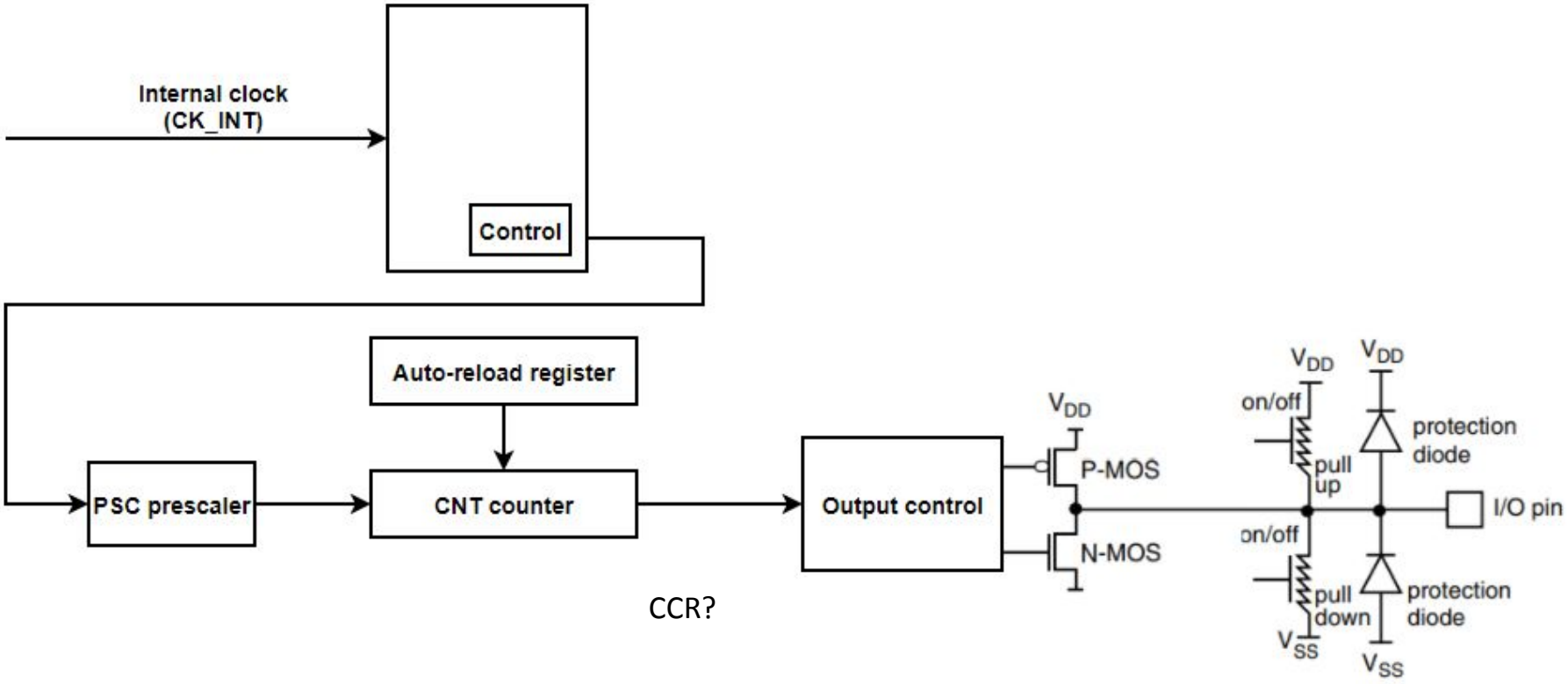
GENERAL-PURPOSE TIMERS

Figure 197. General-purpose timer block diagram



Ref Datasheet: https://www.st.com/resource/en/datasheet/stm32f303re.pdf

GENERAL-PURPOSE TIMERS



CCR?

PWM modes

There are two PWM modes avialable: PWM mode 1 and 2

- PWM mode 1:
  - In upcounting, the channel is active (1) as long as Period (CNT) < Pulse (CCR),
    else inactive.
  - In downcounting, the channel is inactive as long as Period > Pulse, else active.

- PWM mode 2:
  - In upcounting, channel 1 is inactive as long as Period < Pulse, else active.
  - In downcounting, channel 1 is active as long as Period > Pule, else inactive.

BÀI TOÁN:

PWM với tần số 10KHz. D = 50%

Sử dụng Timer2 với thanh ghi PSC, ARR, CCR, CNT là 16bit.
Tần số ngõ vào Timer là 8MHz.

Bai 2:

PWM với tần số 20KHz. ⇒ PSC, ARR

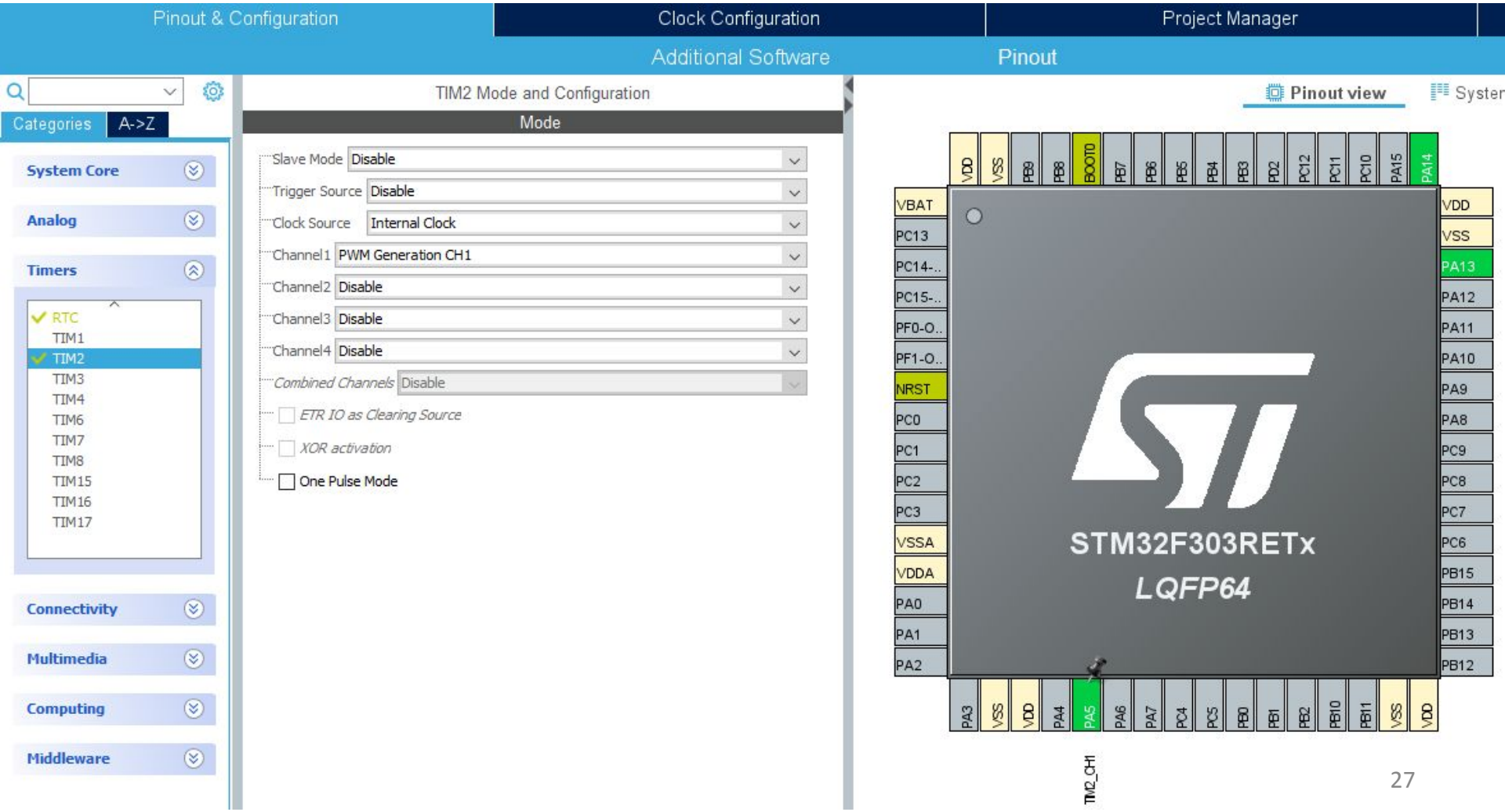
D = 20% => CCR

Sử dụng Timer2 với thanh ghi PSC, ARR, CCR, CNT là 16bit.
Tần số ngõ vào Timer là 8MHz.

15 phut:

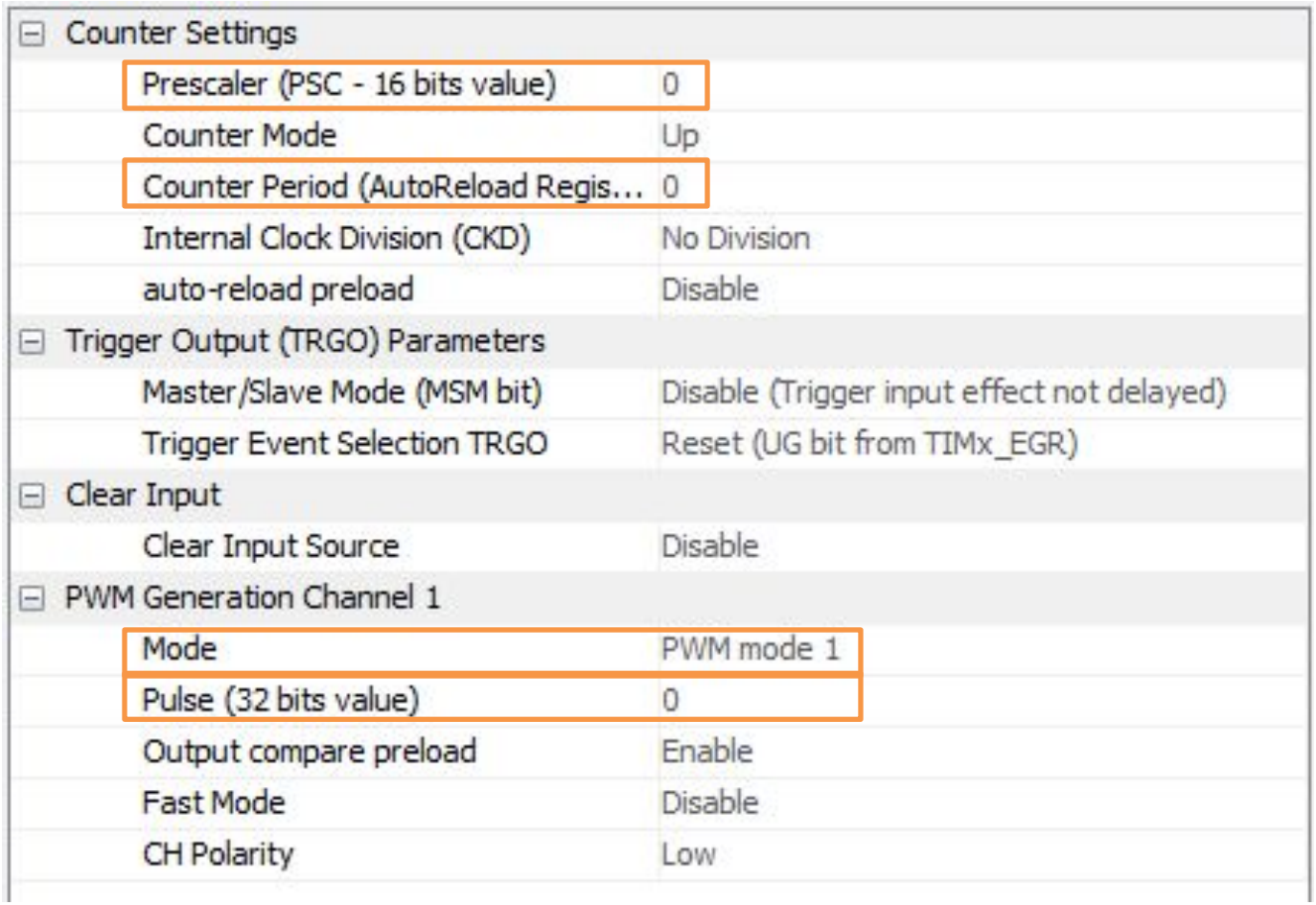
Using CubeMX to Configure the PWM Mode

The first step is to select the PWM Generation CHx mode for the desired channel



Using CubeMX to Configure the PWM Parameters

Next, from the TIMx configuration view, it is possible to configure the other PWM settings (PWM mode 1 or 2, Pulse, and so on).



Pulse - CCR value

How to change CCR value?

__HAL_TIM_SET_COMPARE (&htim2, TIM_CHANNEL_1, value);

or: htim2.Instance->CCR1 = value;

```
while (1)
  {
    for(pwmValue = 0; pwmValue < arrValue; pwmValue++)
      {
        htim2.Instance->CCR1 = pwmValue;
        HAL_Delay(1000/arrValue);
      }
    for(pwm = arrValue; pwm > 0 ; pwm--)
      {
        htim2.Instance->CCR1 = pwmValue;
        HAL_Delay(1000/arrValue);
      }
  }
```

Tìm hiểu thêm

https://www.st.com/resource/en/application_note/an4776-generalpurpose-timer-cookbook-for-stm32-microcontrollers-stmicroelectronics.pdf

https://www.st.com/resource/en/application_note/an4013-stm32-crossseries-timer-overview-stmicroelectronics.pdf

Instructor

Eng. Nguyen Huynh Nhat Thuong