- CO3009 -

- Working with Time: Interrupts, Counters, and Timers -

Timer in Stm32F103

- Advanced-control timers (TIM1 and TIM8)
- General-purpose timers (TIM2 to TIM5)
- Basic timers (TIM6 and TIM7)
- General-purpose timers (TIM9 to TIM14)
- Real-time clock (RTC)
- Independent watchdog (IWDG)
- Window watchdog (WWDG)

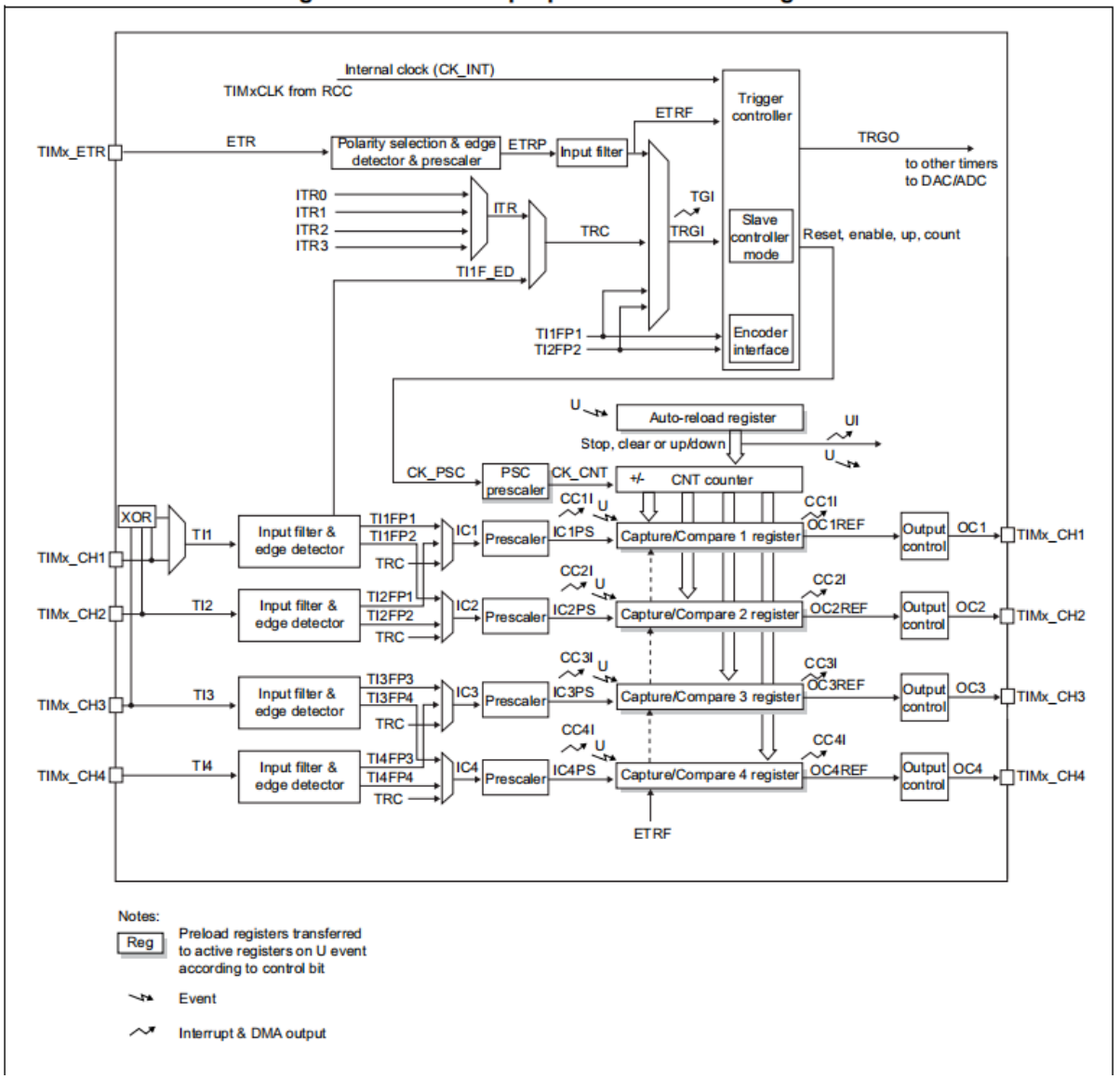
General-purpose Timers (TIM2 to TIM5)

- The general-purpose timers consist of a 16-bit auto-reload counter driven by a programmable prescaler.
- They may be used for a variety of purposes, including:
  - measuring the pulse lengths of input signals (input capture) or
  - generating output waveforms (output compare and PWM).
- Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the RCC clock controller prescalers.
- The timers are completely independent, and do not share any resources.

General-purpose TIMx Timer Features

- 16-bit 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.
- Up to 4 independent channels for:
  - Input capture
  - Output compare
  - PWM generation (Edge- and Center-aligned modes)
  - One-pulse mode output
- 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
- Supports incremental (quadrature) encoder and hall-sensor circuitry for positioning purposes
- Trigger input for external clock or cycle-by-cycle current management

Figure 100. General-purpose timer block diagram



TIMx Functional Description

- Time-base unit
  - Prescaler description
- Counter modes
  - Upcounting mode
  - Downcounting mode
  - Center-aligned mode (up/down counting)

https://www.youtube.com/watch?v=VfbW6nfG4kw&t=637s

https://drive.google.com/file/d/1IPCnbnfwLIAEaNPWvPxO6MSJ1Psvruva/view?usp=sharing

Figure 101. Counter timing diagram with prescaler division change from 1 to 2

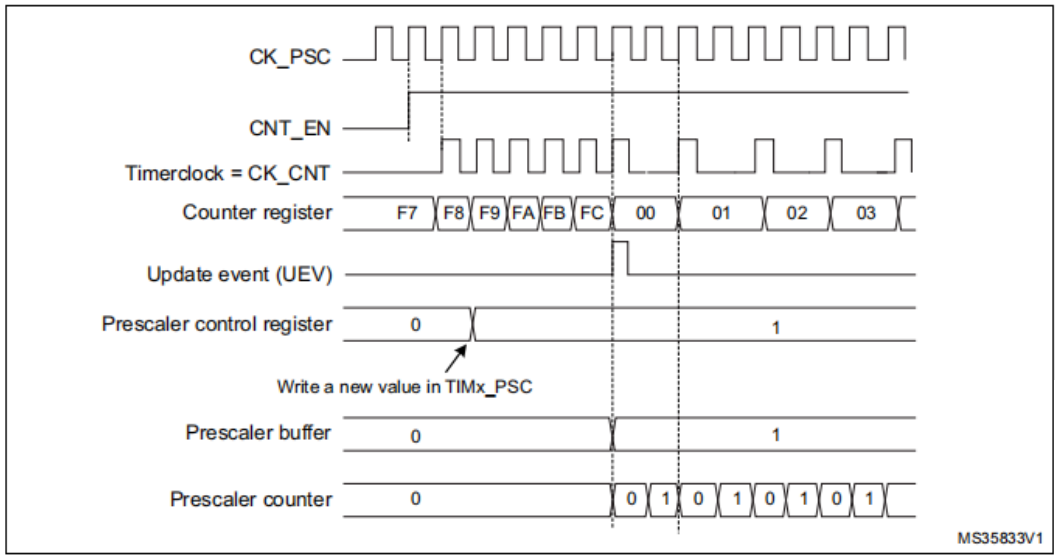


Figure 102. Counter timing diagram with prescaler division change from 1 to 4

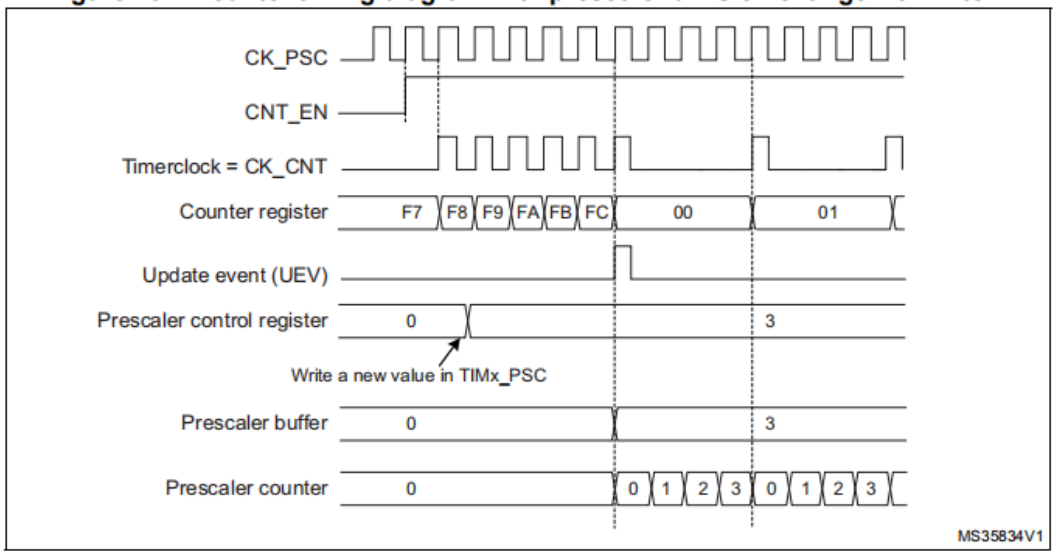


Figure 103. Counter timing diagram, internal clock divided by 1

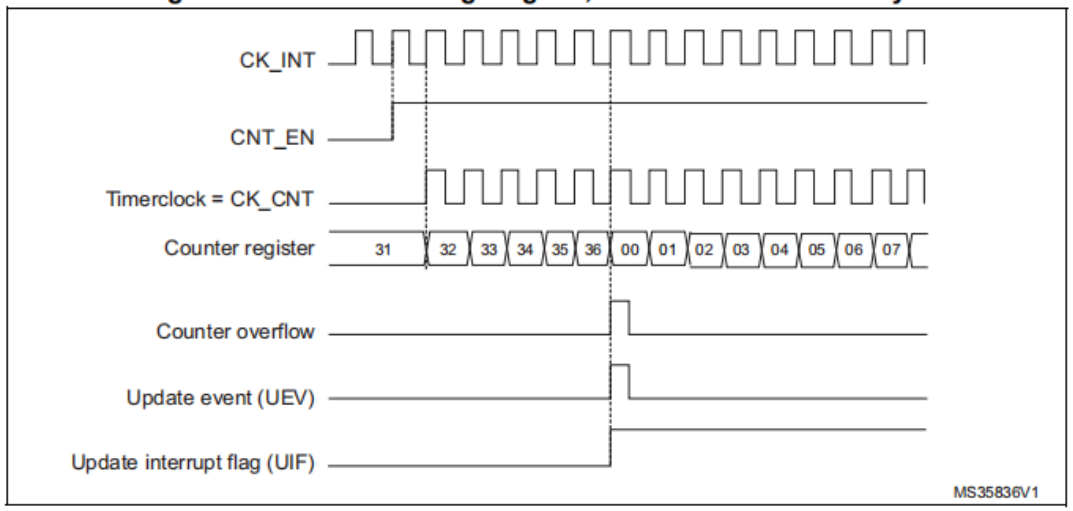
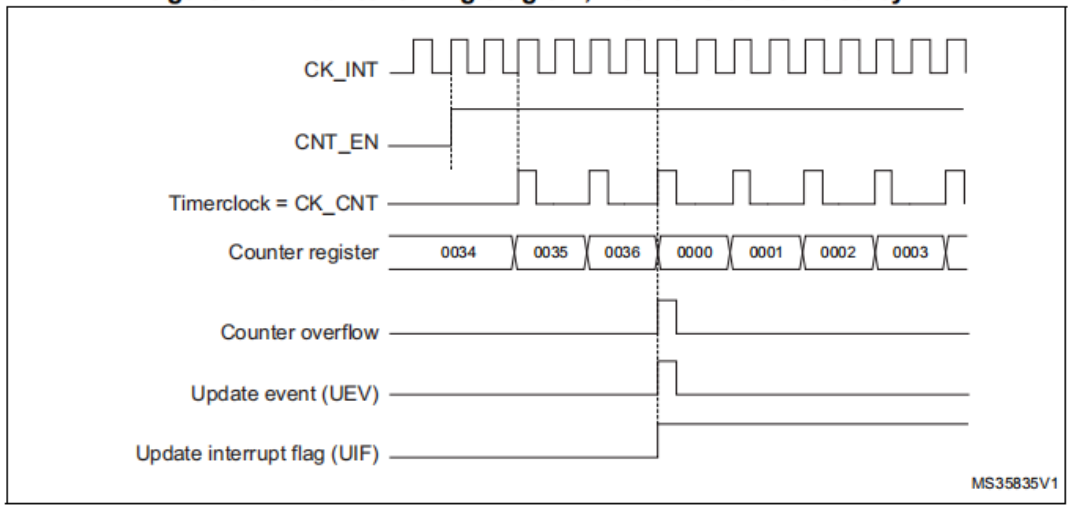


Figure 104. Counter timing diagram, internal clock divided by 2



Clock Selection

- The counter clock can be provided by the following clock sources:
  - Internal clock (CK_INT)
  - External clock mode1: external input pin (TIx)
  - External clock mode2: external trigger input (ETR).
  - Internal trigger inputs (ITRx): using one timer as prescaler for another timer, for example, Timer1 can be configured to act as a prescaler for Timer 2.

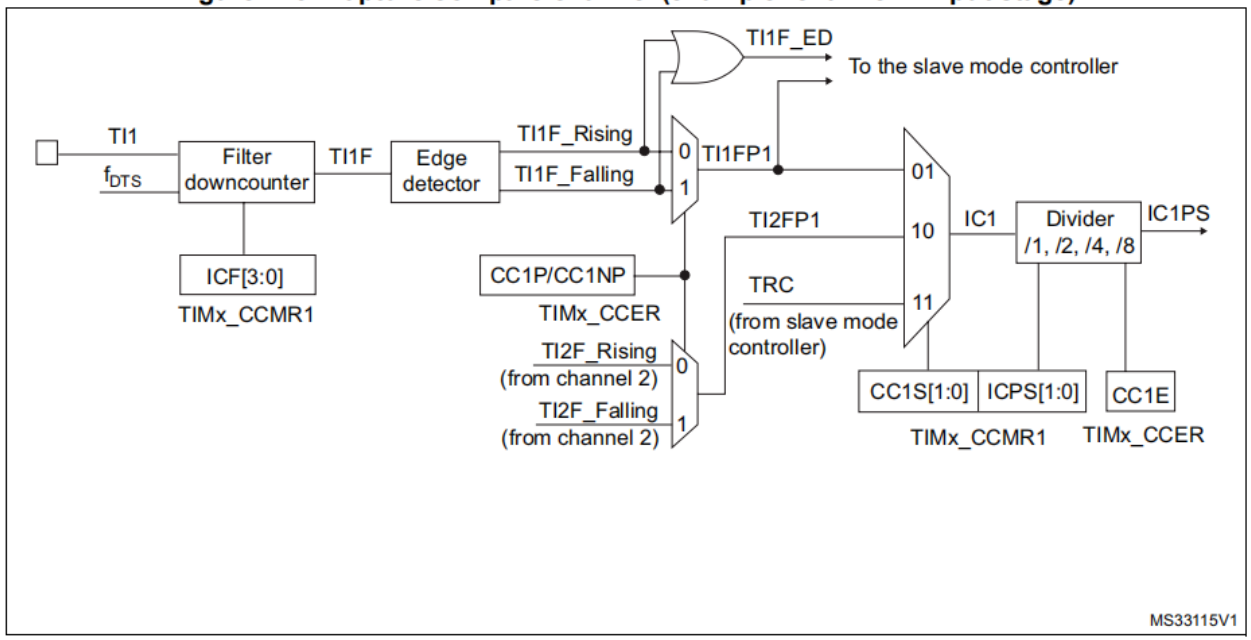
Questions?

- How to configure the upcounter to count in response to a rising edge on the TI2 input?
- How to configure the upcounter to count each 2 rising edges on ETR?

Capture/Compare Channels

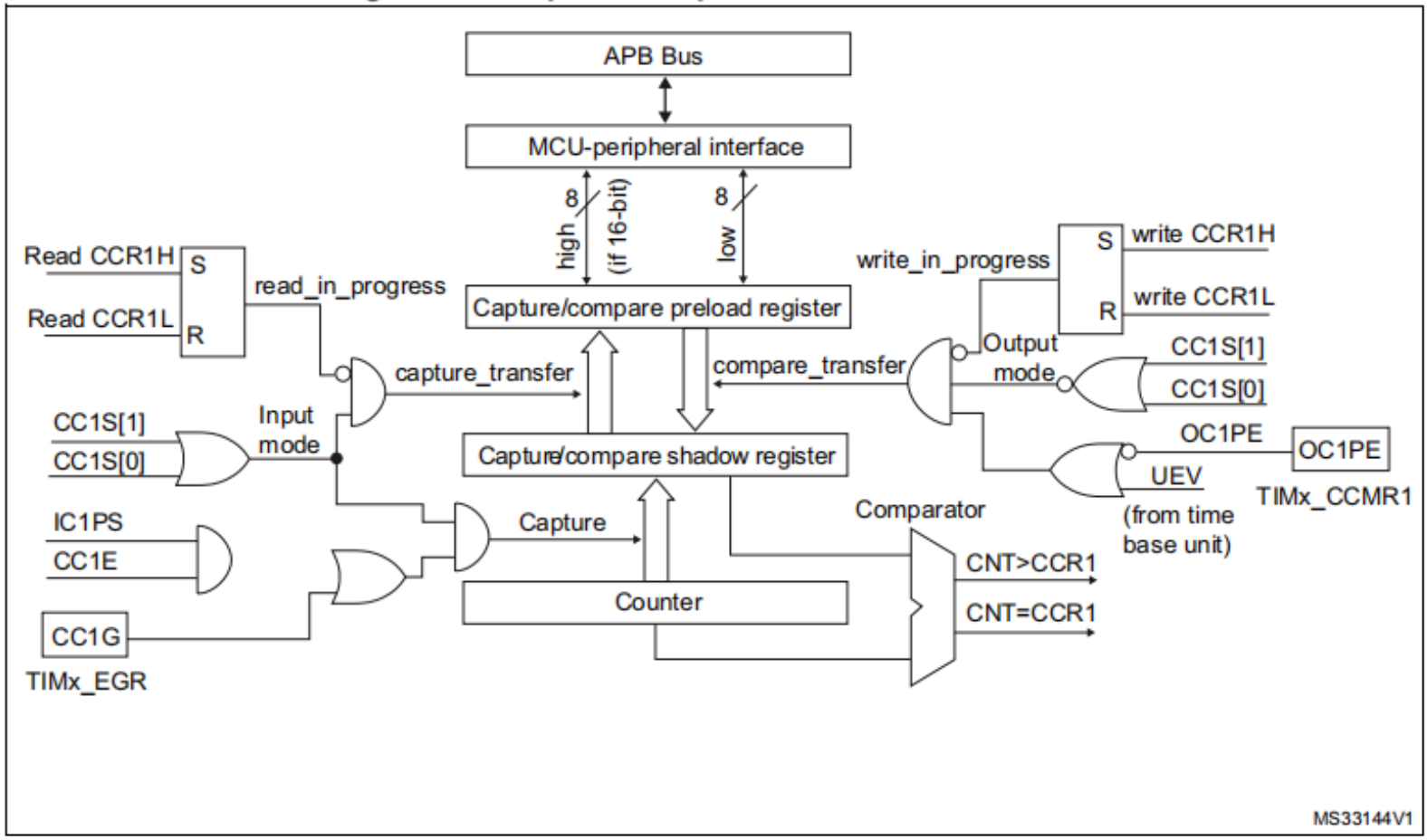
- Each Capture/Compare channel is built around a capture/compare register (including a shadow register),
- An input stage for capture (with digital filter, multiplexing and prescaler) and an output stage (with comparator and output control)

Figure 125. Capture/compare channel (example: channel 1 input stage)



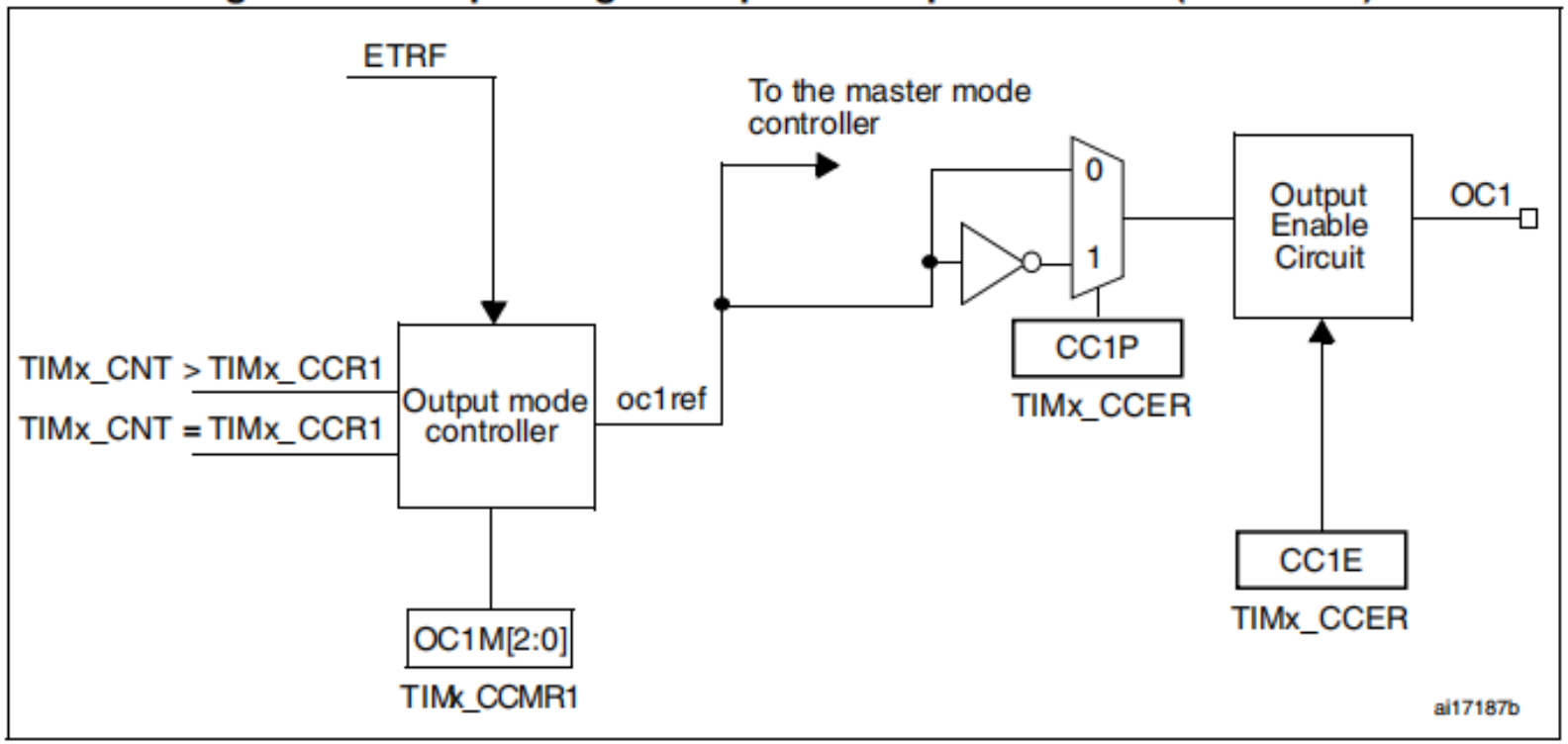
Capture/Compare Channels

Figure 126. Capture/compare channel 1 main circuit



Capture/Compare Channels

Figure 127. Output stage of capture/compare channel (channel 1)



Input/Output Capture/Compare Mode

- Input Capture Mode
  - How to capture the counter value in TIMx_CCR1 when TI1 input rises?
  - What happens when an input capture occurs?

https://www.youtube.com/watch?v=RZ0uszK2PMo

- Output Compare Mode
  - This function is used to control an output waveform or indicating when a period of time has elapsed.

https://drive.google.com/file/d/1F50A8NR4AR2zPZc7wvRDJZjfTO18WId1/view?usp=sharing

PWM Input Mode

- This mode is a particular case of input capture mode. The procedure is the same except:
  - Two ICx signals are mapped on the same TIx input.
  - These 2 ICx signals are active on edges with opposite polarity.
  - One of the two TIxFP signals is selected as trigger input and the slave mode controller is configured in reset mode.

Questions

- How to measure the period and the duty cycle of a PWM signal?
- How to measure the period and the duty cycle of the PWM applied on TI1?

https://drive.google.com/file/d/17RvIlt_PaihrqqBR-7AkB0DLXMltwL_b/view?usp=sharing

PWM Mode

- Pulse width modulation mode allows generating a signal with:
  - A frequency determined by the value of the TIMx_ARR register and
  - A duty cycle determined by the value of the TIMx_CCRx register.

- The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing 110 (PWM mode 1) or ‘111 (PWM mode 2) in the OCxM bits in the TIMx_CCMRx register.

- The user must enable the corresponding preload register by setting the OCxPE bit in the TIMx_CCMRx register, and eventually the auto-reload preload register by setting the ARPE bit in the TIMx_CR1 register.

https://www.youtube.com/watch?v=92_CWBWXPw0

Exercises

- Write a program to generate a square wave on pin PD5 (1 kHz)
  - Using **HAL_Delay(uint32_t ms)** function
- Write a program using Timer3 to generate a square wave on pin PD5 (1 kHz)
- Generate square waves on pins PD1 (10 Hz) and PD7 (100 Hz) respectively.

Exercises (cont)

- Use Timer2 and Timer3 interrupts to generate square waves on pins PD1 (10 Hz) and PD7 (100 Hz) respectively.
- Use Timer3 interrupts to generate square waves on pins PD1 (10 Hz) and PD7 (100 Hz) respectively.
- Write a program to count up PD[0-7] every 1 second using TIMER3 interrupt.