Programmable timing functions
Part 2: Timer operating modes

Textbook: Chapter 8, Section 8.6 (pulse-width modulation)
Chapter 9, Sections 9.6, 9.7 (SysTick and Timer interrupts)

STM32F4xx Technical Reference Manual:
Chapter 17 – Basic timers (TIM6)
Chapter 15 – General-purpose timers (TIM4)
Chapter 10 - Interrupt vectors (for TIM4/TIM6 interrupts)

Timer operating modes

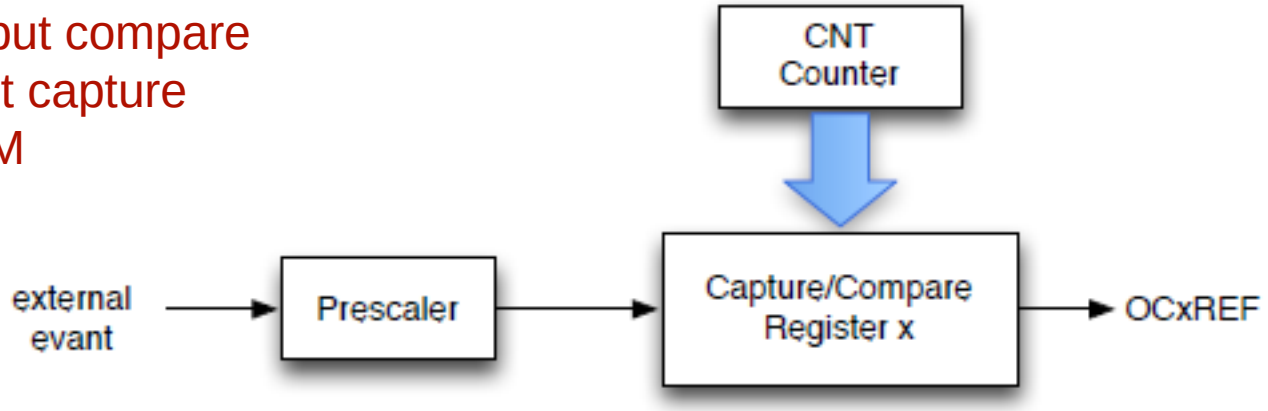
Timers provide operating modes other than periodic interrupts

- **Input capture mode**
  - Connect a GPIO pin to timer input TIMx_CHy
  - Capture CNT value at time of an event on the pin
    - CNT captured in Capture/Compare Register CCRy
  - Use to measure time between events, tachometer signal periods, etc.
- **Output compare mode**
  - Connect timer output TIMx_CHy to a GPIO pin
  - Compare CNT to value in Capture/Compare register CCRy
  - Change output pin when CNT = CCRx
    - Creates a signal change/waveform/pulse/etc.
- **One pulse mode**
  - Setup similar to output compare mode
  - Disable the counter when the event occurs
- **Generate pulse-width modulated (PWM) waveforms**
  - Setup similar to output compare mode
  - Output pin active while CNT < CCRy
  - Output pin inactive while CCRy < CNT < ARR

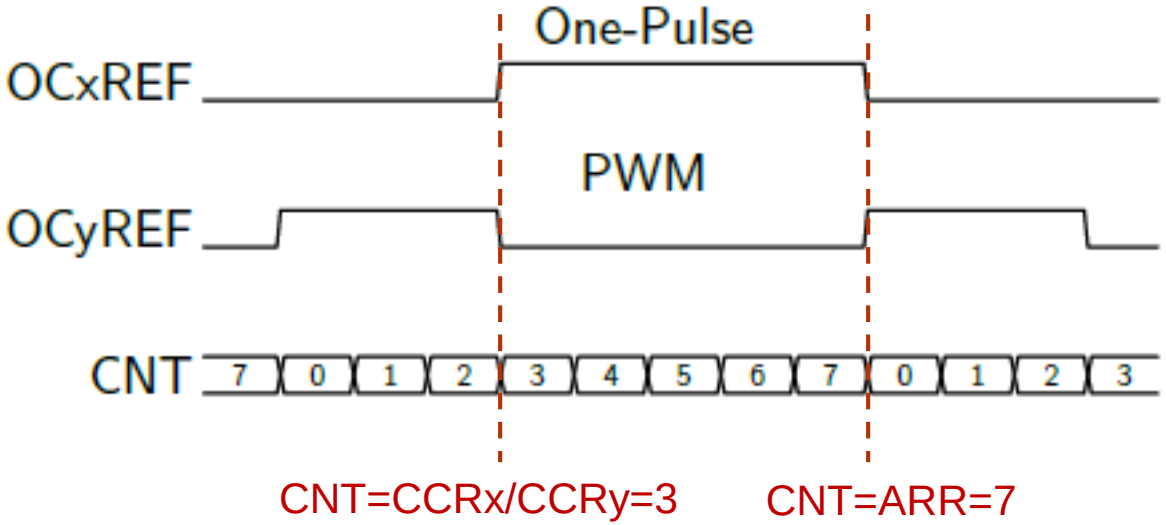
Timer channel hardware

Operating modes:

- Output compare
- Input capture
- PWM

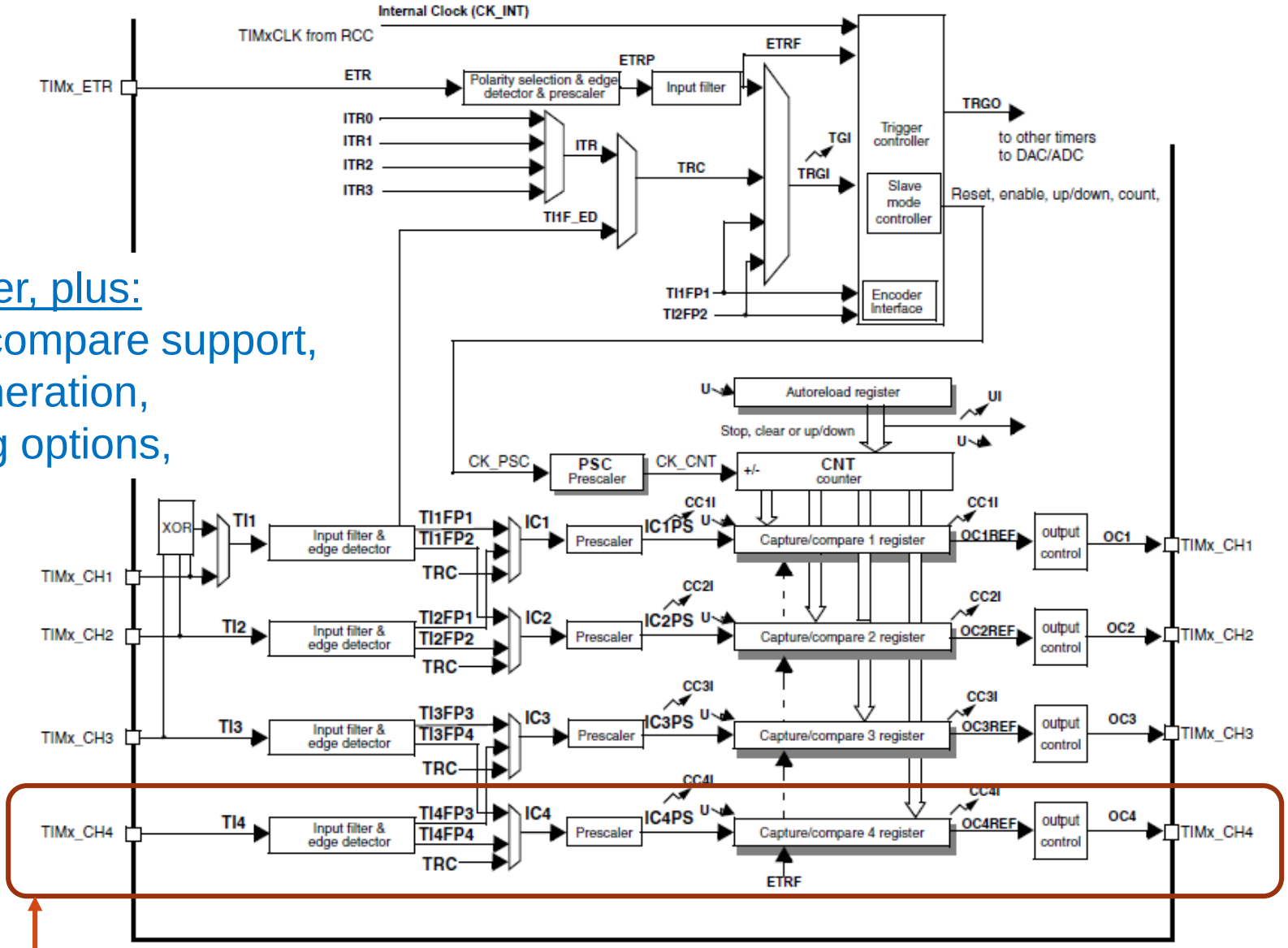


Pulse-width modulation



General-purpose timers TIM2 – TIM5

Basic timer, plus:
Capture/compare support,
PWM generation,
Triggering options,



One “channel” – general-purpose timers have 1, 2, or 4 channels

Capture/Compare Channels

- Different channels but same blocks
  - Capture mode can be used to measure the pulse width or frequency
  - Input stage includes digital filter, multiplexing and prescaler
  - Output stage includes comparator and output control
  - A capture register (with shadow register)

- **Input Stage Example**
  - Input signal->filter->edge detector->slave mode controller or capture command

Capture/Compare Channels

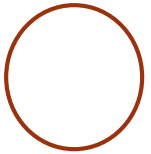
- Main Circuit

- The block is made of one preload register and a shadow register.
  - In capture mode, captures are done in shadow register than copied into preload register
  - In compare mode, the content of the preload register is copied into the shadow register which is compared to the counter

Capture/Compare Channels

- Output stage

To output pin



Comparator
outputs

Configure the GPIO - AF

- Refer to the user manual to determine which pin is able to connect to TIMx channels (table of pin functions)
- Configure the GPIO pin as AF mode, be careful with the pull up or down setting since it should match the setting of edge detection
- Configure the GPIO AF register to select the TIMx channel for the pin

Alternate functions for pins PD12-13-14-15

From STM32F407 Data Sheet – Table 6

Pin number						Pin name (function after reset)[1]	Pin type	I / O structure	Notes	Alternate functions	Additional functions
LQFP64	WLCSP90	LQFP100	LQFP144	UFBGA176	LQFP176						
-	-	60	82	M15	101	PD13	I/O	FT		FSMC_A18/TIM4_CH2/ EVENTOUT	
-	-	-	83	-	102	V_{SS}	S				
-	-	-	84	J13	103	V_{DD}	S				
-	F2	61	85	M14	104	PD14	I/O	FT		FSMC_D0/TIM4_CH3/ EVENTOUT/ EVENTOUT	
-	F1	62	86	L14	105	PD15	I/O	FT		FSMC_D1/TIM4_CH4/ EVENTOUT	
-	G2	59	81	N13	100	PD12	I/O	FT		FSMC_A17/TIM4_CH1 / USART3_RTS/ EVENTOUT	

TIM4 can connect to GPIO pins (alt. function):

PD12 = TIM4_CH1
PD13 = TIM4_CH2
PD14 = TIM4_CH3
PD15 = TIM4_CH4

Discovery board LEDs driven by PD12-PD15.

TIM4 inputs for input capture mode.

TIM4 outputs for output capture/PWM/one-pulse mode.

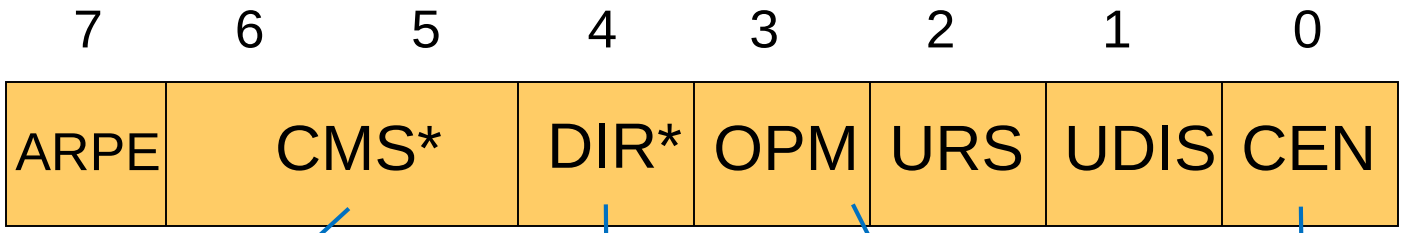
TIMx capture/compare registers

- **TIMx_CCRy** = TIMx capture/compare register, channel y
  - TIMx_CCR1 – address offset 0x34
  - TIMx_CCR2 – address offset 0x38
  - TIMx_CCR3 – address offset 0x3C
  - TIMx_CCR4 – address offset 0x40
  - Register width (16/32 bits) same as CNT/ARR registers
  - TIMx may have 0, 1, 2, or 4 channels (see manual)

- **Output compare mode**: TIMx_CCRy compared to CNT, with match signaled on OCy output
- **Input capture mode**: CNT captured in TIMx_CCRy when designated input signal event occurs

Timer System Control Register 1

TIMx_CR1 address offset 0x00 (default = all 0's)



Center mode select

00 = edge-aligned
-count in one direction
Others: center aligned
-count in both directions

Direction

0 = count up
1 = count down

Counter Enable

0 = disable
1 = enable

One Pulse Mode

1 = counter stops at update event
0 = counter continues at UE

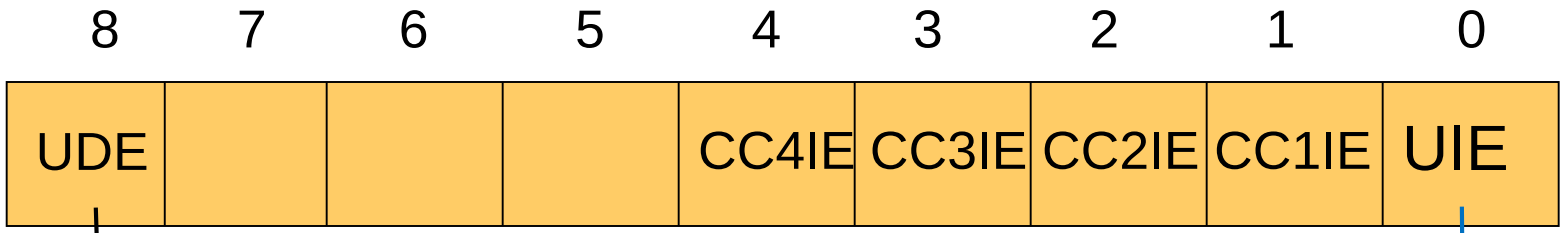
* Not in TIM6/TIM7

Advanced Options:

ARPE = 1 enables ARR buffer (transferred to ARR on update event)
URS = 0 allows multiple events to generate update interrupt
1 restricts update interrupt to counter overflow/underflow
UDIS = 0 enables update event to be generated

Timer DMA/Interrupt Control Register

TIMx_DIER address offset 0x0C (default = all 0's)



Update DMA request enable

1 = enable

0 = disable

Update interrupt enable

1 = enable

0 = disable

Capture/Compare interrupt enable

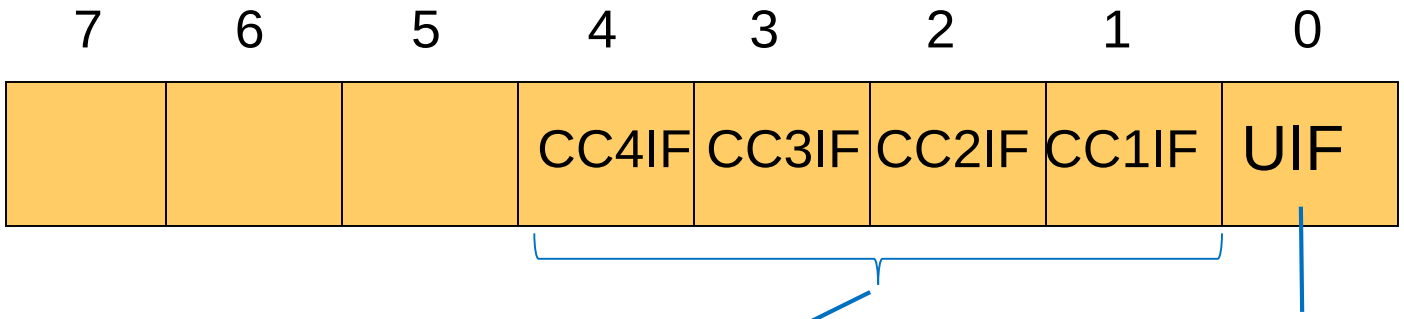
1 = CCx interrupt enabled

0 = disabled

TIMx interrupt on capture/compare event

Timer Status Register

TIMx_SR address offset 0x10 (reset value = all 0's)



Capture/compare interrupt flags

1 = update interrupt pending
0 = no update occurred

Set by hardware on capture/comp event
Cleared by software
(reset CCxIF bit to 0)

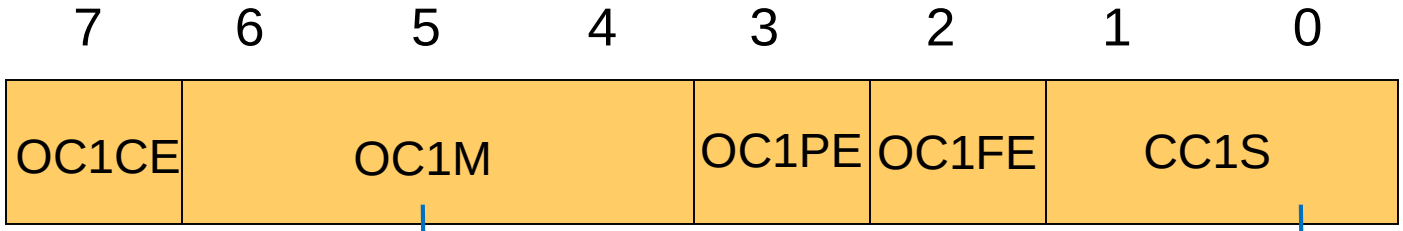
Update interrupt flag

1 = update interrupt pending
0 = no update occurred

Set by hardware on update event
Cleared by software
(reset UIF bit to 0)

Capture/compare mode register 1/2

TIMx_CCMR1 address offset 0x18 (reset value = all 0's)
TIMx_CCMR2 offset 0x1C



Output compare 1 mode

000 = frozen (no events)
001 = Set CH1 active on match
010 = Set CH1 inactive on match
011 = Toggle CH1 on match
100 = Force CH1 to inactive (immediate)
101 = Force CH1 to active (immediate)
110 =PWM mode 1 (active to inactive)
111 = PWM mode 2 (inactive to active)

Capture/compare 1 select

00 = output
01 = input: IC1 = TI1
10 = input: IC1 = TI2
11 = input: IC1 = TRC

Bits 15-8 configure Channel 2 (same order)

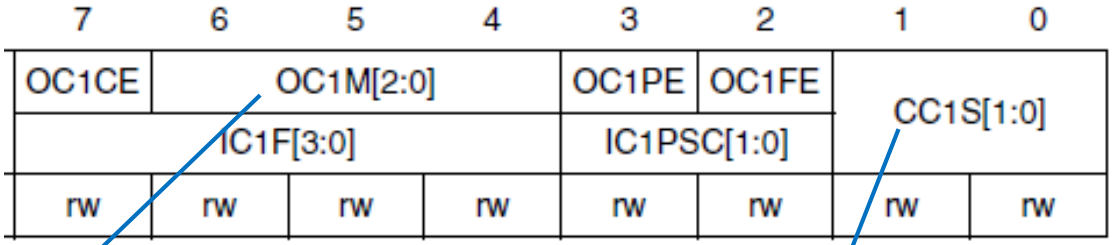
CCMR2 configures Channels 3/4

OC1xxx = function if CC1S selects "output"
IC1xxx = functions if CC1S selects "input"
Active/inactive level selected in TIMx_CCER register

Capture/compare mode register 1/2

TIMx_CCMR1 address offset 0x18 (reset value = all 0's)
TIMx_CCMR2 offset 0x1C

Bits 15-8 configure Channel 2 (same order)
CCMR2 configures Channels 3/4



Bits shown for output mode. (Input mode next slide)

Output compare 1 mode

000 = frozen (no events)
001 = Set CH1 active* on match
010 = Set CH1 inactive* on match
011 = Toggle CH1 on match
100 = Force CH1 to inactive* (immediate)
101 = Force CH1 to active* (immediate)
110 = PWM mode 1 (active* to inactive*)
111 = PWM mode 2 (inactive* to active*)

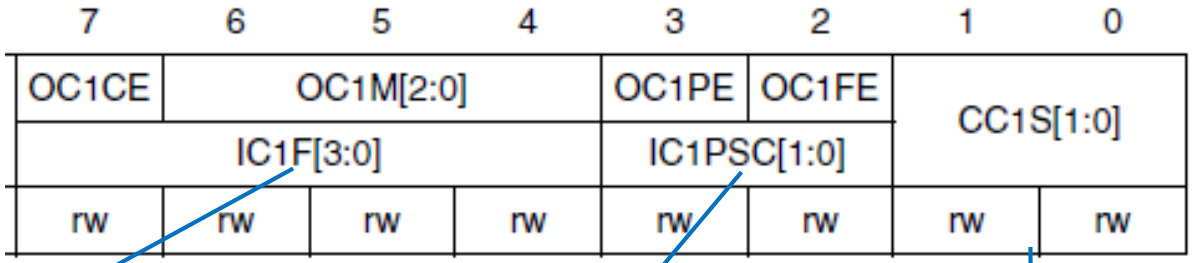
Capture/compare 1 select

00 = output
01 = input: IC1 = TI1
10 = input: IC1 = TI2
11 = input: IC1 = TRC

* Active/inactive level selected in TIMx_CCER register

Capture/compare mode register ½

(Input capture mode)



Input Capture 1 Filter

Defines frequency used to sample TI1 input and length of digital filter applied to TI1

Input Capture 1 Prescaler

00: capture on every event
01: capture on every 2nd event
10: capture on every 4th event
11: capture on every 8th event

Capture/Compare 1 Select

00 = output
01 = input: IC1 = TI1
10 = input: IC1 = TI2
11 = input: IC1 = TRC

Bits 15-8 configure Channel 2 (same order)

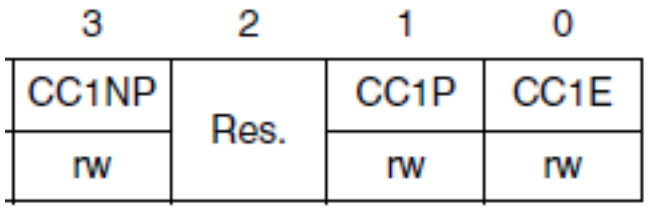
CCMR2 configures Channels 3/4

OC1xxx = function if CC1S selects “output”
IC1xxx = functions if CC1S selects “input”
Active/inactive level selected in TIMx_CCER register

Capture/compare enable register

TIMx_CCER address offset 0x20 (reset value = all 0's)

CC4: bits 15-12
CC3: bits 11-8
CC2: bits 7-4
(same order as CC1)



CC1 Polarity
If CC1 output, CC1P selects:
0 = OC1 active high
1 = OC1 active low

If CC1 input:
CC1NP/CC1P select capture trigger:
00: falling edge of input
01: rising edge of input
11: both edges of input

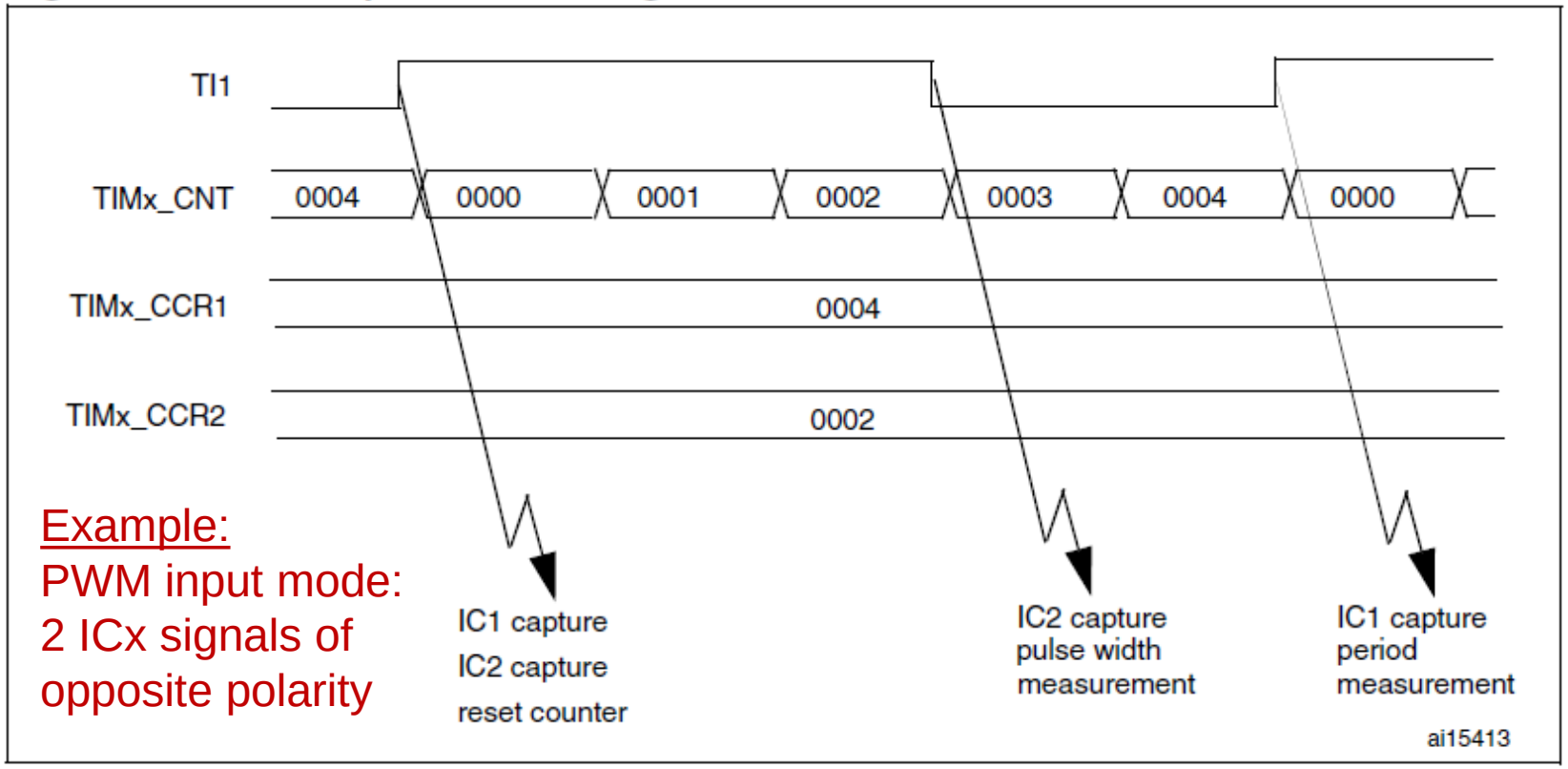
CC1 Enable
If CC1 output:
1 = On: OC1 driven to output pin
0 = Off: OC1 not on output

If CC1 input:
1 = Capture enabled
0 = Capture disabled

Input capture mode

TIMx_CCRx latches TIMx_CNT value when transition detected

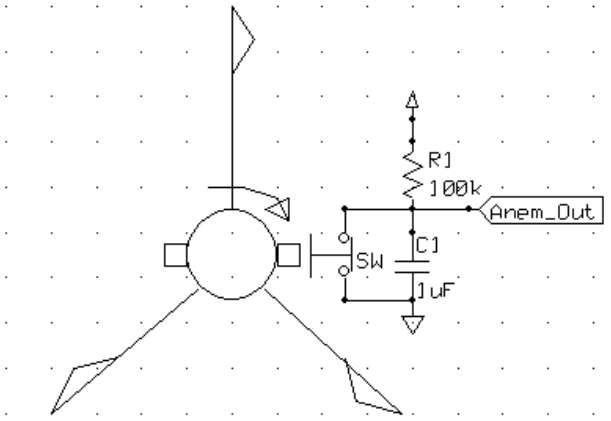
- CCxIF flag sets and interrupt generated, if enabled
- Signal edge programmable (rising, falling, both)

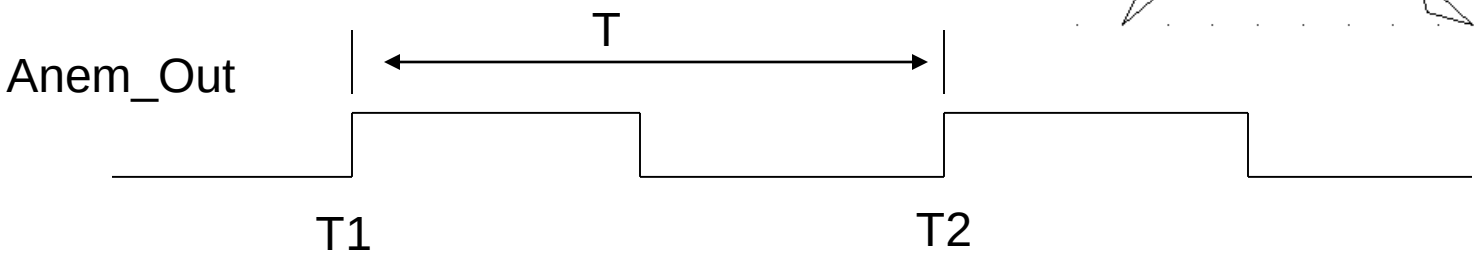


Example:
PWM input mode:
2 ICx signals of opposite polarity

Wind Speed Indicator (Anemometer)

- Rotational speed (and pulse frequency) is proportional to wind velocity
- Two measurement options:
  - Frequency (best for high speeds)
  - Width (best for low speeds)
- Can solve for wind velocity v
- How can we use the Timer for this?
  - Use Input Capture Mode to measure period of input signal





Input Capture Mode for Anemometer

- Operation: Repeat
  - First capture - on rising edge
    - Reconfigure channel for input capture on falling edge
    - Clear counter, start new counting
  - Second Capture - on falling edge
    - Read capture value, save for later use in wind speed calculation
    - Reconfigure channel for input capture on rising edge
    - Clear counter, start new counting
- Solve the wind speed
  - $V_{wind} = K \div (C_{falling} - C_{rising}) \times Freq$

Set up for Anemometer measurement

- Apply ***Anem_Out*** signal to pin PD15
  - TIM4_CH4 is an alternate function for PD15 (from data sheet)
  - Configure PD15 as alternate function in GPIOD_MODER
  - Select alternate function TIM4_CH4 for PD15 in GPIOD_AFRH
- Configure **TIM4_PSC** and **TIM4_ARR** for TIM4 counting period
  - Best if counting period > time to be measured
  - Reset **TIM4_CNT** after each capture
- **TIM4_CCMR2** Capture/Compare mode register 2 (Channels 3 and 4)
  - Set CC4S to map IC4 on TI4
  - Set IC4F, IC4PSC to defaults (no filter or prescale)
- **TIM4_CCER** Capture/compare enable register
  - Set CC4E to select "input" mode
  - Set CC4N:CC4P = 00 to select rising-edge (01 for falling edge)
- **TIMx_DIER** DMA/interrupt enable register
  - Set CC4IE to enable interrupt on input capture event *(to read captured value)*
- **TIM4_CR1** Control register: Set CEN to enable the counter
- **TIM4_SR** Status register: CC1IF indicates input event occurred *(clear by software)*
- **TIM4_CCR4** Capture/Compare register = captured value of TIM4_CNT

Output Compare Mode

- Control an output waveform or indicating when a period of time has elapsed
- When a Match occurs

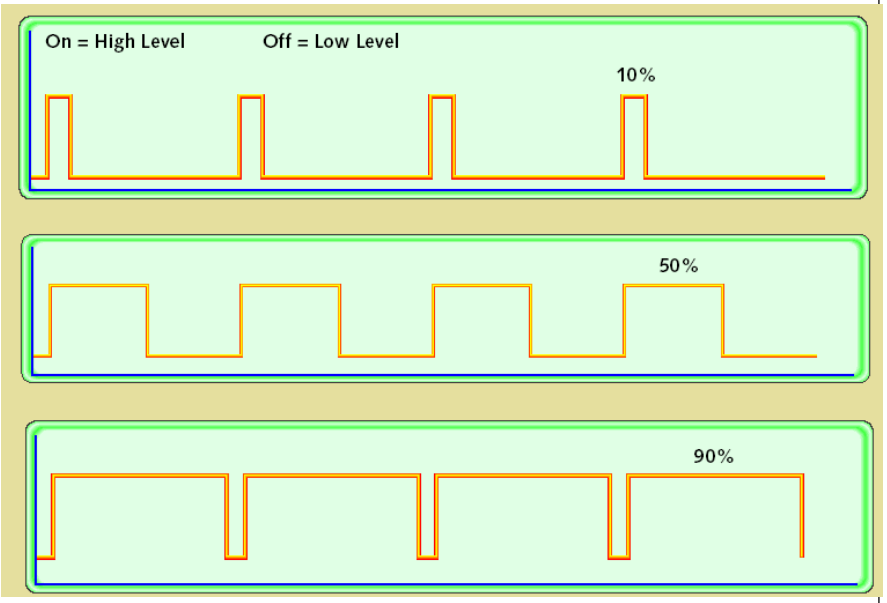
  (CCRx=CNT)

  - Generate specific output on corresponding pin
  - Set the CCxIF(Interrupt status) bit in the SR
  - Generate Interrupt if configured
  - Generate DMA request if configured

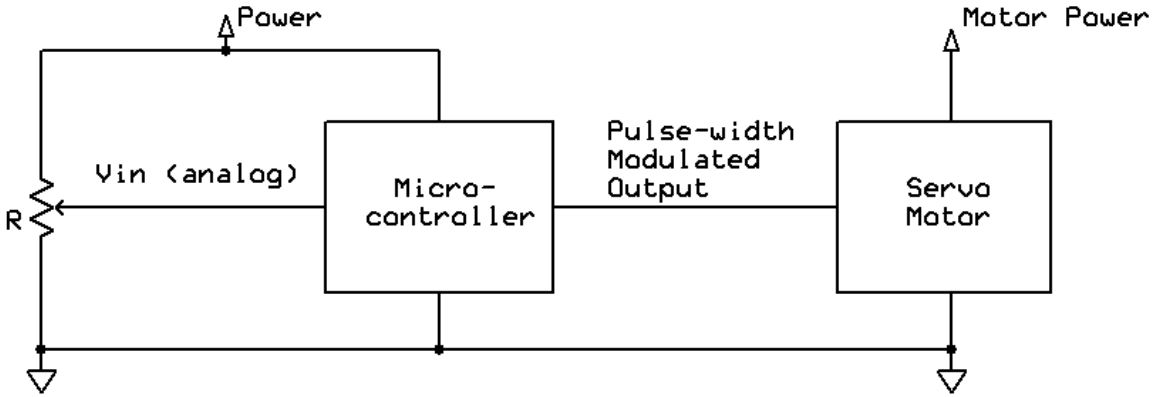
- **Configure steps**
  - Select the counter clock
  - Write the desired data in ARR and CCR registers
  - Enable Interrupt or DMA request if needed
  - Select the output mode
  - Enable the counter

Pulse-Width Modulation

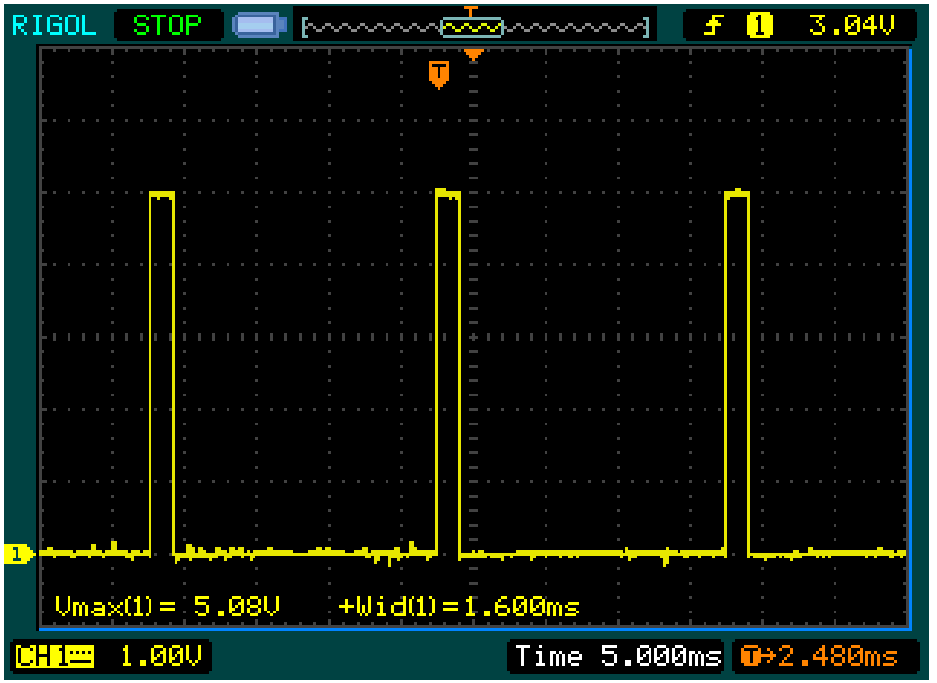
- Uses of PWM
  - **Digital power amplifiers** are more efficient and less expensive than analog power amplifiers
    - Applications: motor speed control, light dimmer, switch-mode power conversion
    - Load (motor, light, etc.) responds slowly, averages PWM signal
  - **Digital communication** is less sensitive to noise than analog methods
    - PWM provides a *digital encoding* of an *analog* value
    - Much less vulnerable to noise
- PWM signal characteristics
  - Modulation frequency – how many pulses occur per second (fixed)
  - Period – 1/(modulation frequency)
  - On-time – amount of time that each pulse is on (asserted)
  - Duty-cycle – on-time/period
  - Adjust *on-time* (hence *duty cycle*) to represent the analog value



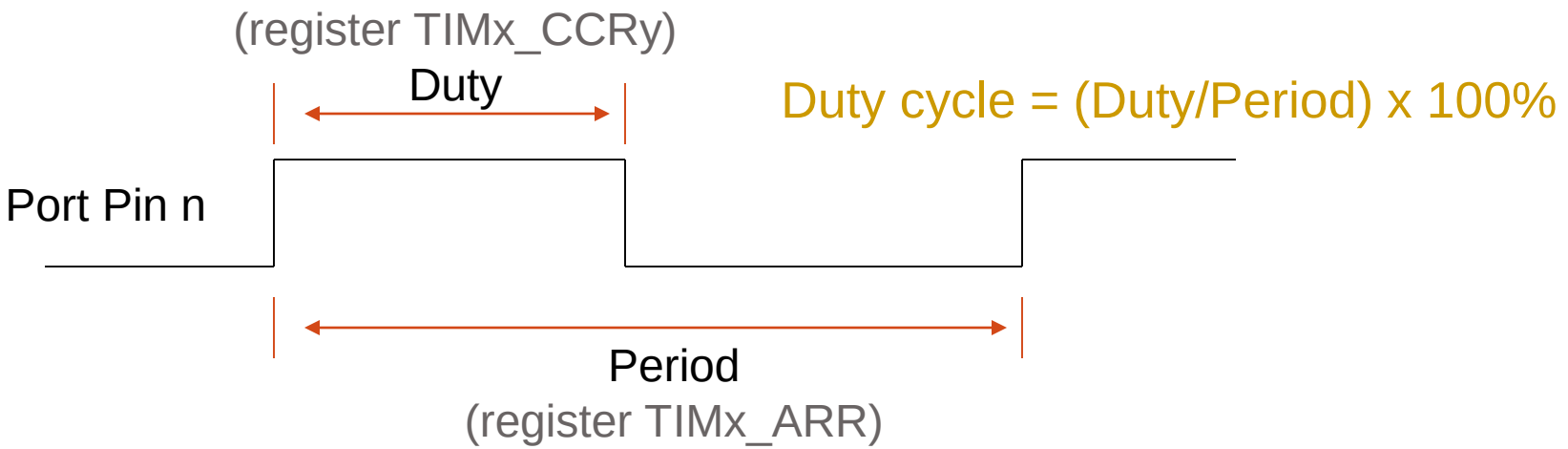
PWM to Drive Servo Motor



- Servo PWM signal
  - 20 ms period
  - 1 to 2 ms pulse width



Pulse-Width Modulator (PWM)



- Configure the GPIO AF for the pin to be driven
- PWM is done by comparing TIMx_CCRy and TIMx_CNT
  - Set **TIMx_ARR** = period
  - Set **TIMx_CCRy** = duty cycle time
- **TIMx_CCMRn** (capture/compare configuration)
  - Set bit CCxE = 1 to configure the channel as output
  - Set bits OCxM = 110 (PWM mode 1) – active if CNT < CCRy, inactive otherwise
    OCxM = 111 (PWM Mode 2) - inactive if CNT < CCRy , active otherwise
- **TIMx_CCER**:
  - Set CCxP bit to select output polarity of Ocx (active level)
  - Set CCxE bit to enable the output OCx

Edge-Aligned

- CMS bits in TIMx_CR1 are 00
- As long as TIMx_CNT<TIMx_CCRx then the OCXREF is high
- Select down-counting or up-counting using DIR bit in TIMx_CR1

Example – 20KHz PWM signal with 10% duty cycle on pin PD12

- Configure TIM4, channel 1
- Assume timer clock = 84MHz
  - Period = 84MHz/20KHz = 4200 = TIM4_ARR
  - Duty = 4200 x 10% = 420 = TIM4_CCR1
- Program TIM4_CCMR1 bits:
  - CC1E = 0 (make channel 1 is output)
  - CC1M = 110 (PWM mode 1: active-to-inactive)
- Program TIM4_CCER bits:
  - CC1P = 0 to define OC1 as active high
  - CC1E = 1 to enable output OC1
- Configure PD12 as TIM4_CH1
  - Select AF mode for PD12 in MODER
  - Select TIM4_CH1 for PD12 in AFRH