ETH zürich

MOTORS and PWM

Credits:
STMicroelectronics

Project-based Learning Center | ETH Zurich
Tommaso Polonelli tommaso.polonelli@pbl.ee.ethz.ch
Michele magno michele.magno@pbl.ee.ethz.ch
Vlad Niculescu vladn@iis.ee.ethz.ch

Safety considerations

NEVER USE PROPELLES IN THE CLASSROOM

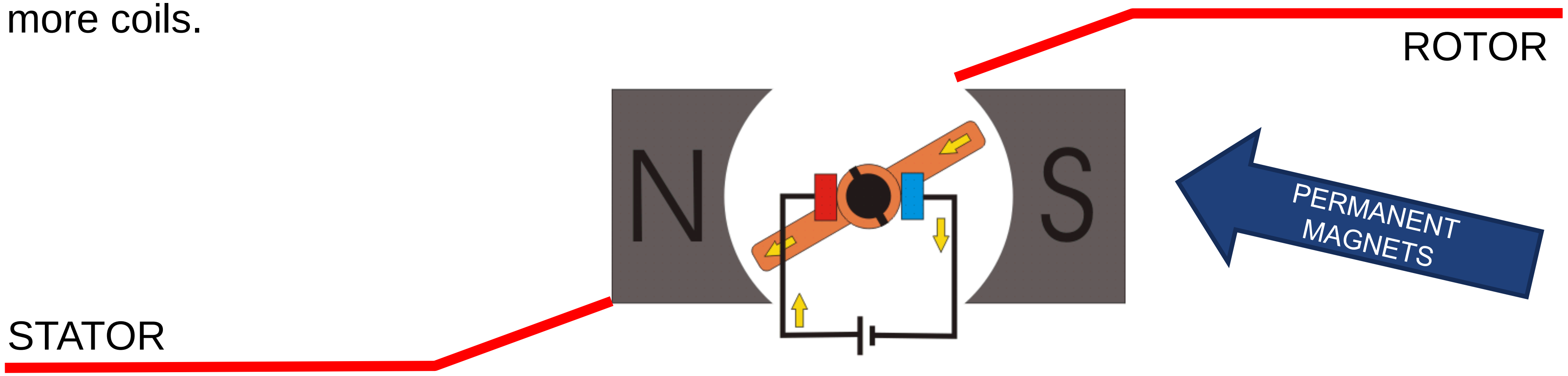
Safety considerations

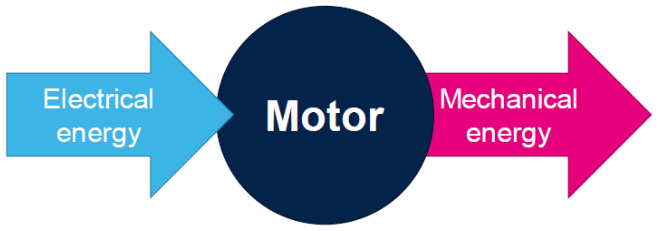
BATTERY

⚠ **LiPO batteries can be damaged and even explode if they are short-circuited or overcharged.**

Basic Principle

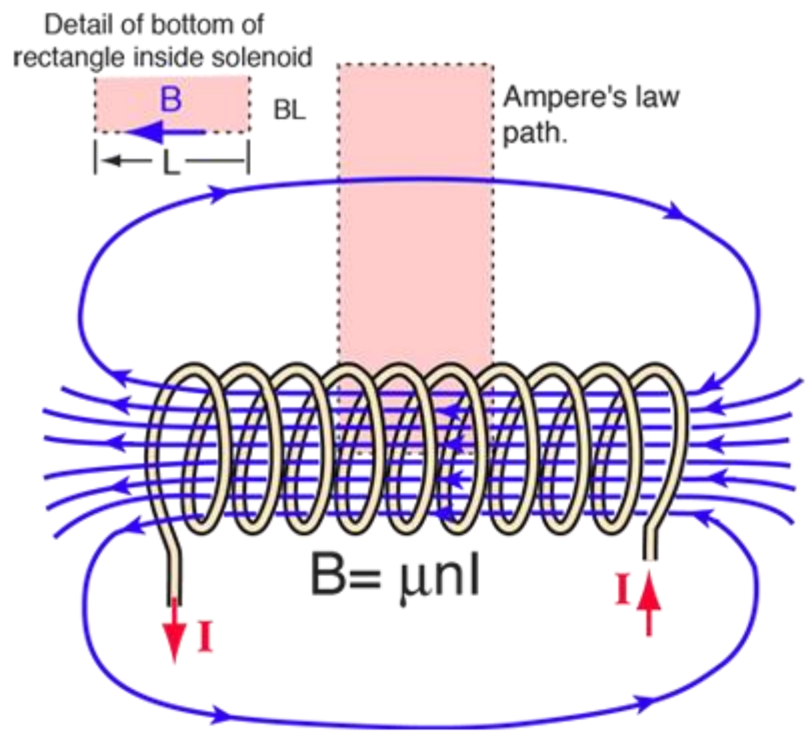
An electric motor is a device converting electrical energy into mechanical energy (generally a torque). This conversion is usually obtained through the generation of a magnetic field by means of a current flowing into one or more coils.





Magnetic field generation

The relation between electrical energy (current) and magnetic field generated by a solenoid (coil) is obtained through the following formula:



magnetic field

motor characteristics

supply current

magnetic permeability

# of coil turns expressed as coil density [turns/cm]

A long straight coil of wire can be used to generate a nearly uniform magnetic field similar to that of a bar magnet. Such coils, called solenoids, have an enormous number of practical applications.

ETH zürich

Torque and load angle

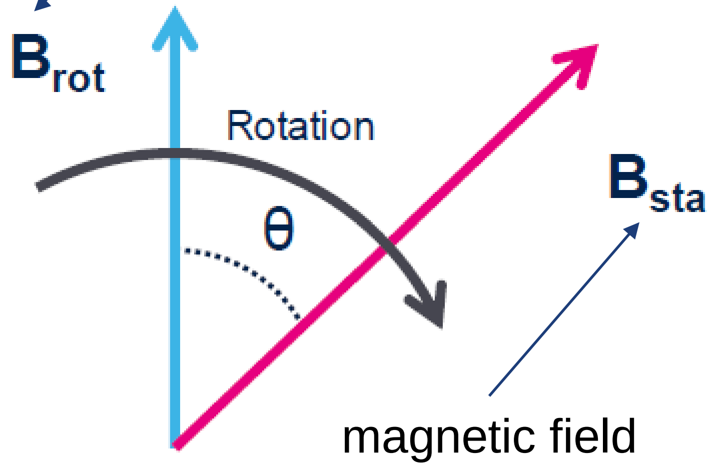
The **output torque (T)** of an electrical motor depends on the intensity of the rotor and stator magnetic fields and on their phase relation.

Load angle
The angle between two magnetic field

motor torque

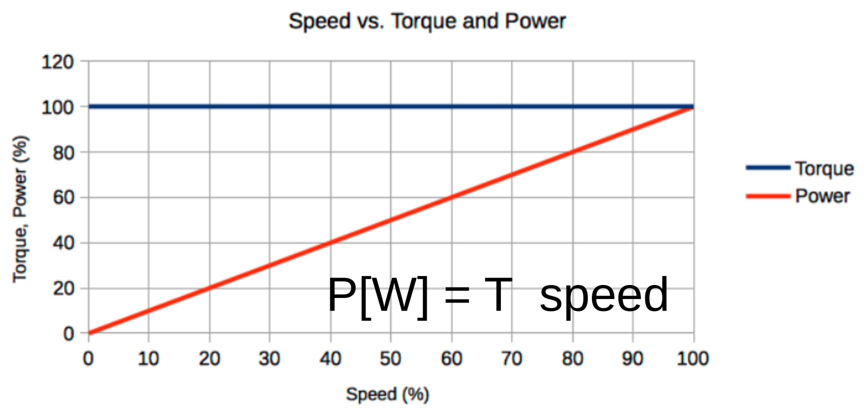
T

magnetic field of the rotor



magnetic field of the stator

The maximum output torque, and then the **maximum efficiency**, is obtained when the load angle is **90°**



P[W] = T speed

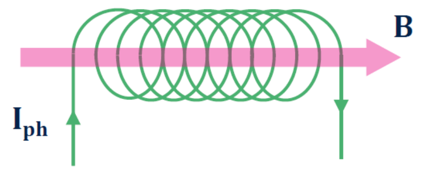
The driving force of an electric motor is torque - not horsepower.

The torque is the twisting force that makes the motor running and the torque is active from 0% to 100% operating speed.

Basic principle

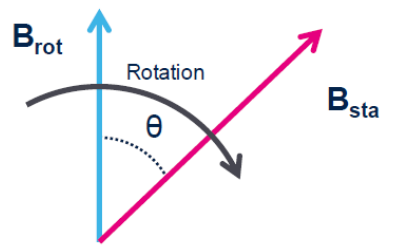
The electric motor operation is based on the following points:

- At least one of the two magnetic field is generated by a solenoid carrying a current.

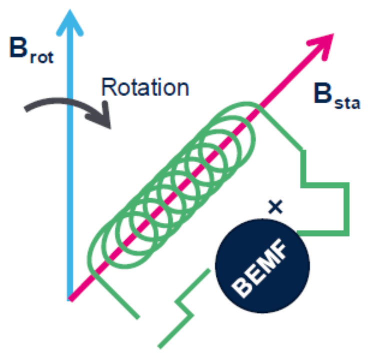


- Phase relation between the rotor and stator magnetic field (i.e. the load angle) must be always greater than 0° in order to keep the motor in motion (negative angles reverse the rotation).

- Output **torque** depends to both solenoid **current** and **load angle**



- Motor rotation causes a **back electro-motive** force opposing the motion itself.

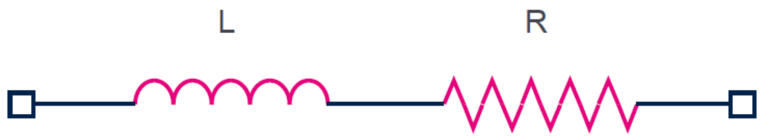


ETH zürich

Basic principle: inductive load

Where:
- V is in Volts
- R is in Ohms
- L is in Henries
- t is in Seconds

Current in an LR Series Circuit

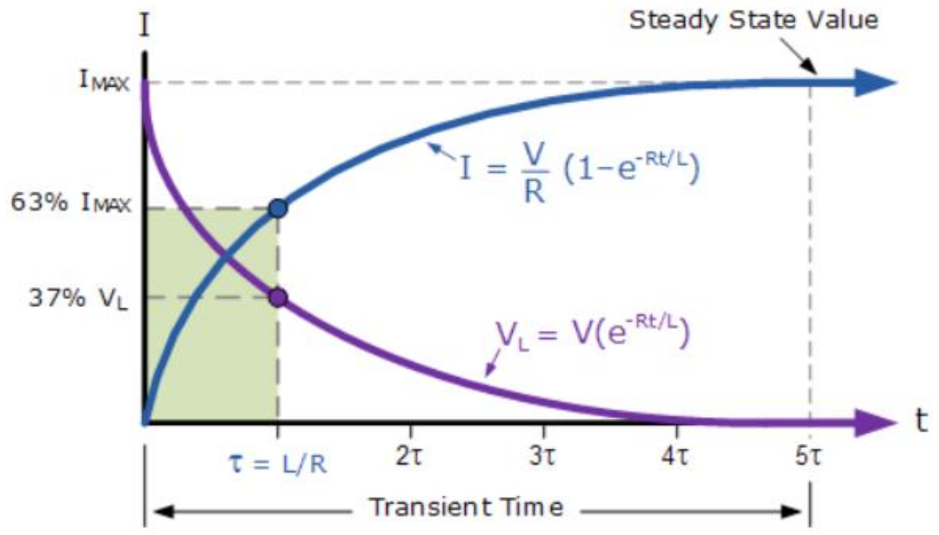


An inductive load (motor phases included) can be represented as and **LR** series which stores energy in the form of current.

Applying a voltage to the load it is possible to change the amount of current stored into the inductance.

Basic principle: inductive load

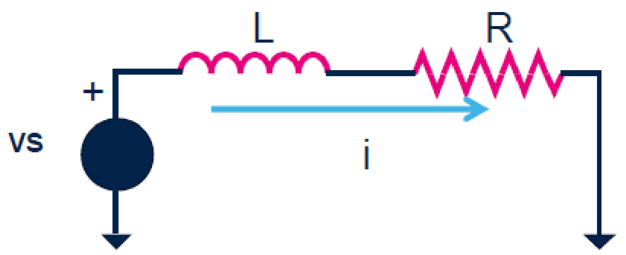
Current in an LR Series Circuit



The Time Constant, (τ) of the LR series circuit is given as L/R and in which V/R represents the **final steady state** current value after five time constant values. Once the current reaches this maximum steady state value at ~ 5τ, the inductance of the coil has reduced to zero acting more like a short circuit and effectively removing it from the circuit.

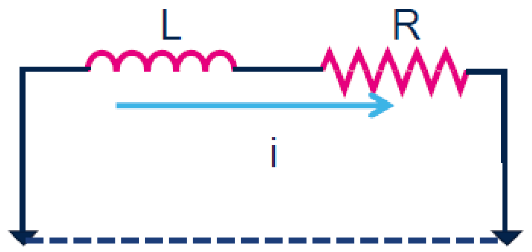
Therefore the current flowing through the coil is limited only by the resistive element in Ohms of the coils windings.

Charge and discharge of an inductive load



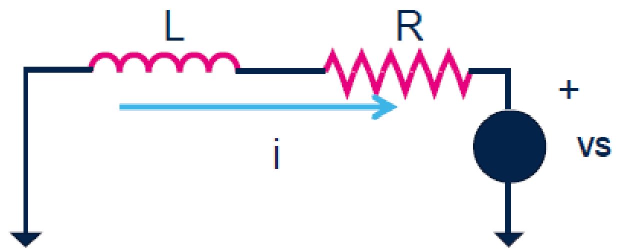
Scenario 1 (ON time) accelerate
inductance is charged applying a voltage

i



Scenario 2 (slow decay)
Inductance is discharged shorting the leads

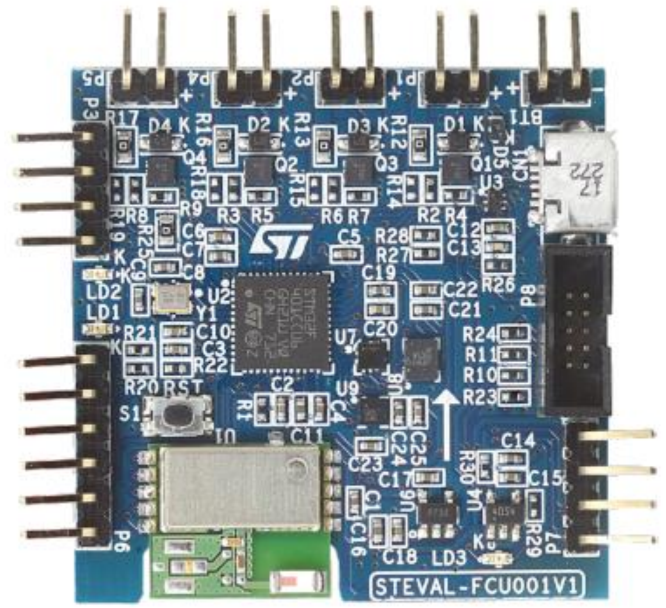
i

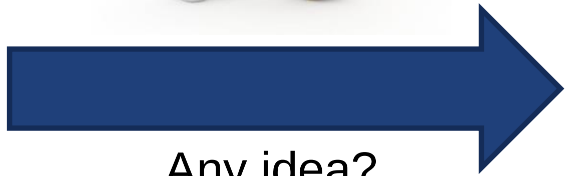


Scenario 3 (fast decay) break
Inductance is discharged applying a voltage

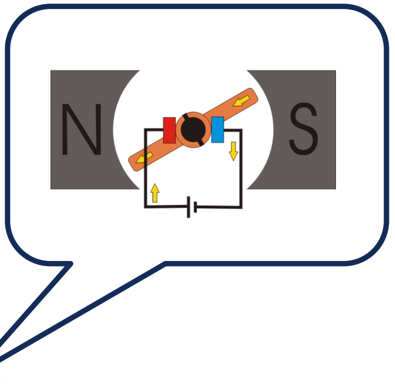
i

How to control motor speed/torque/current?





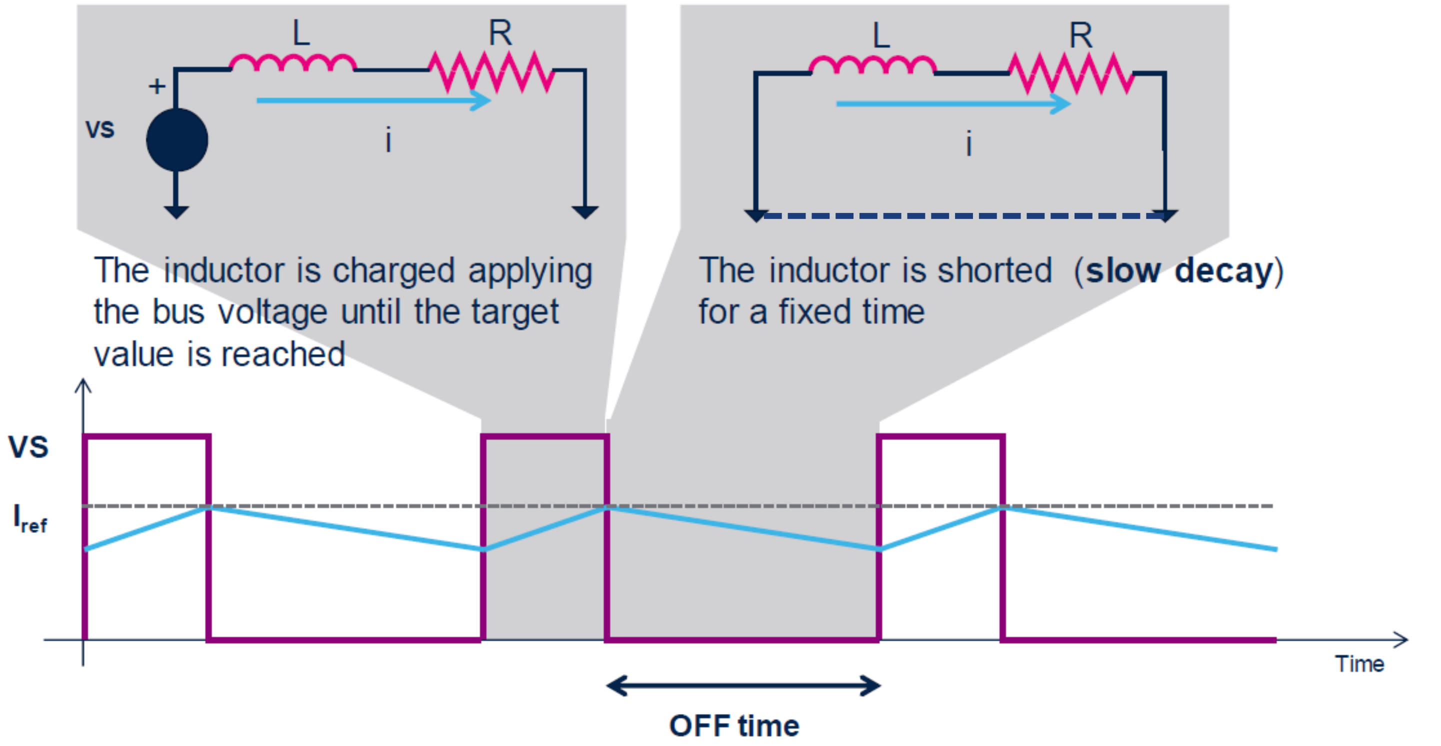
Any idea?



PWM current control basics

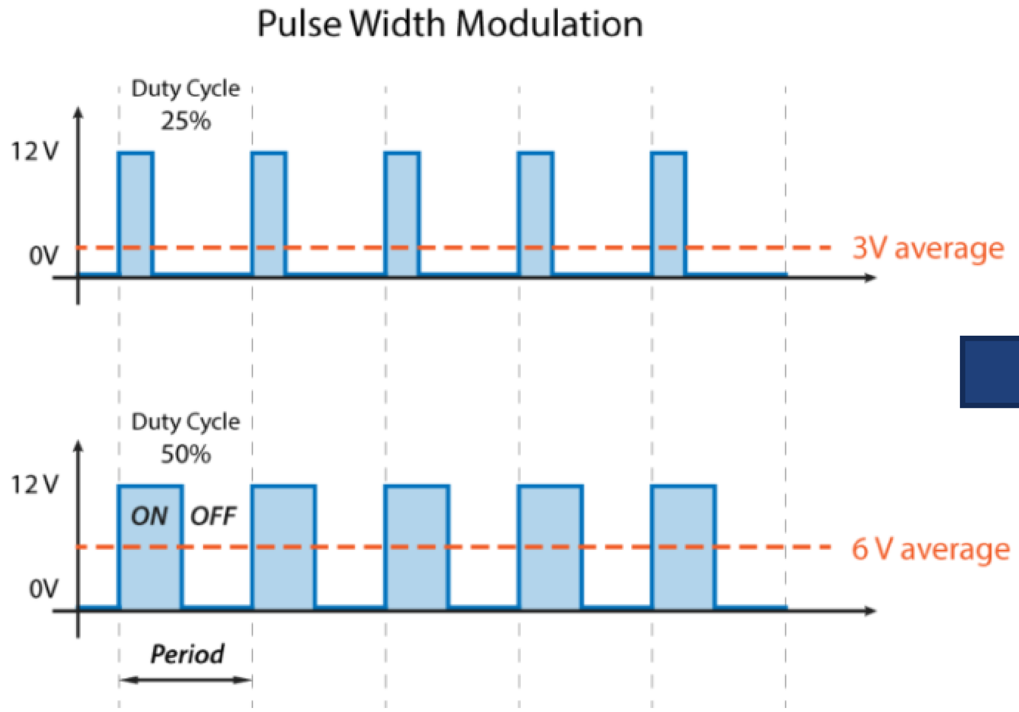
The most common method to control the current is the **Pulse Width Modulation** method.
The duty-cycle changes according to the target current and boundary conditions.

It is a digital modulation → MCU! (our STM32F4 natively supports PWM)

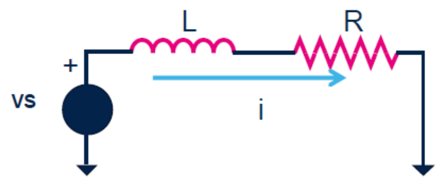

L
R
+
VS
i
The inductor is charged applying the bus voltage until the target value is reached
L
R
i
The inductor is shorted (**slow decay**) for a fixed time
VS
I_{ref}
Time
OFF time

PWM current control basics

PWM is a technique that allows us to adjust the average value of the voltage.



Low pass filter



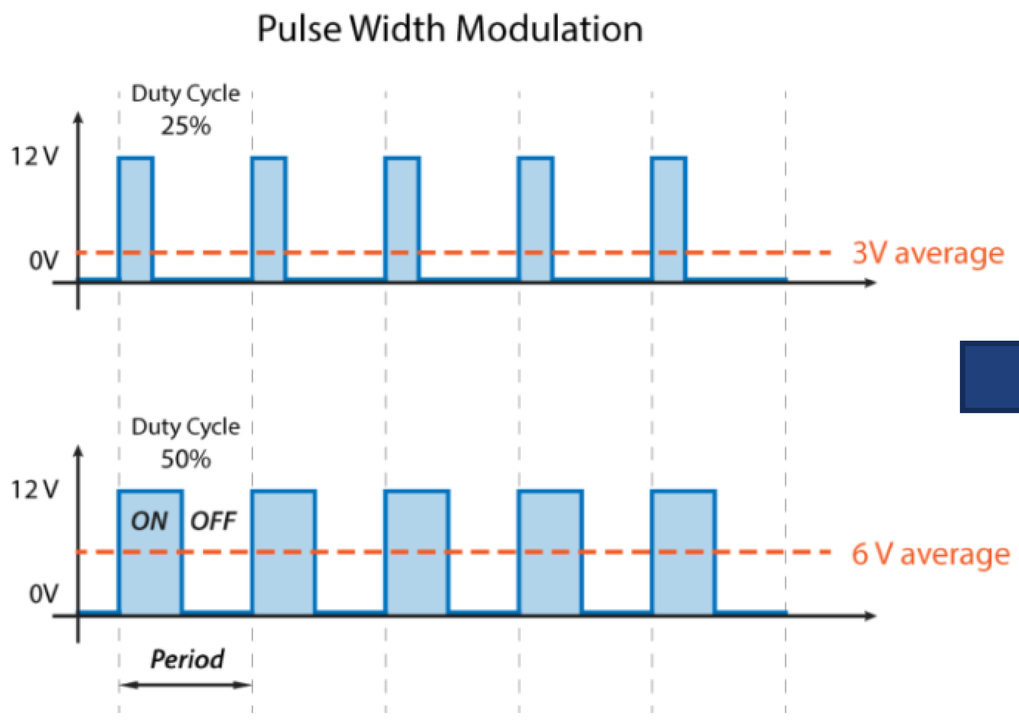
i

working condition

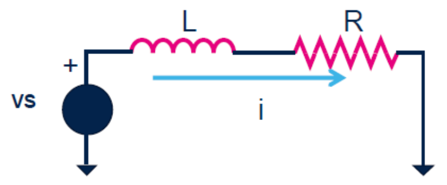
PWM period << Time Constant (τ)

PWM current control basics

PWM is a technique that allows us to adjust the average value of the voltage.



Low pass filter



i

working condition

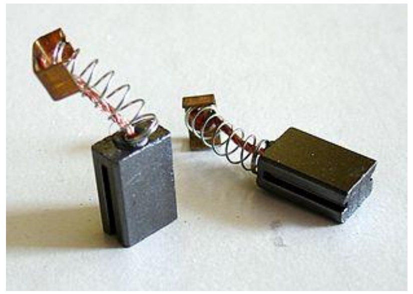
PWM period << Time Constant (τ)

Brushed DC motors

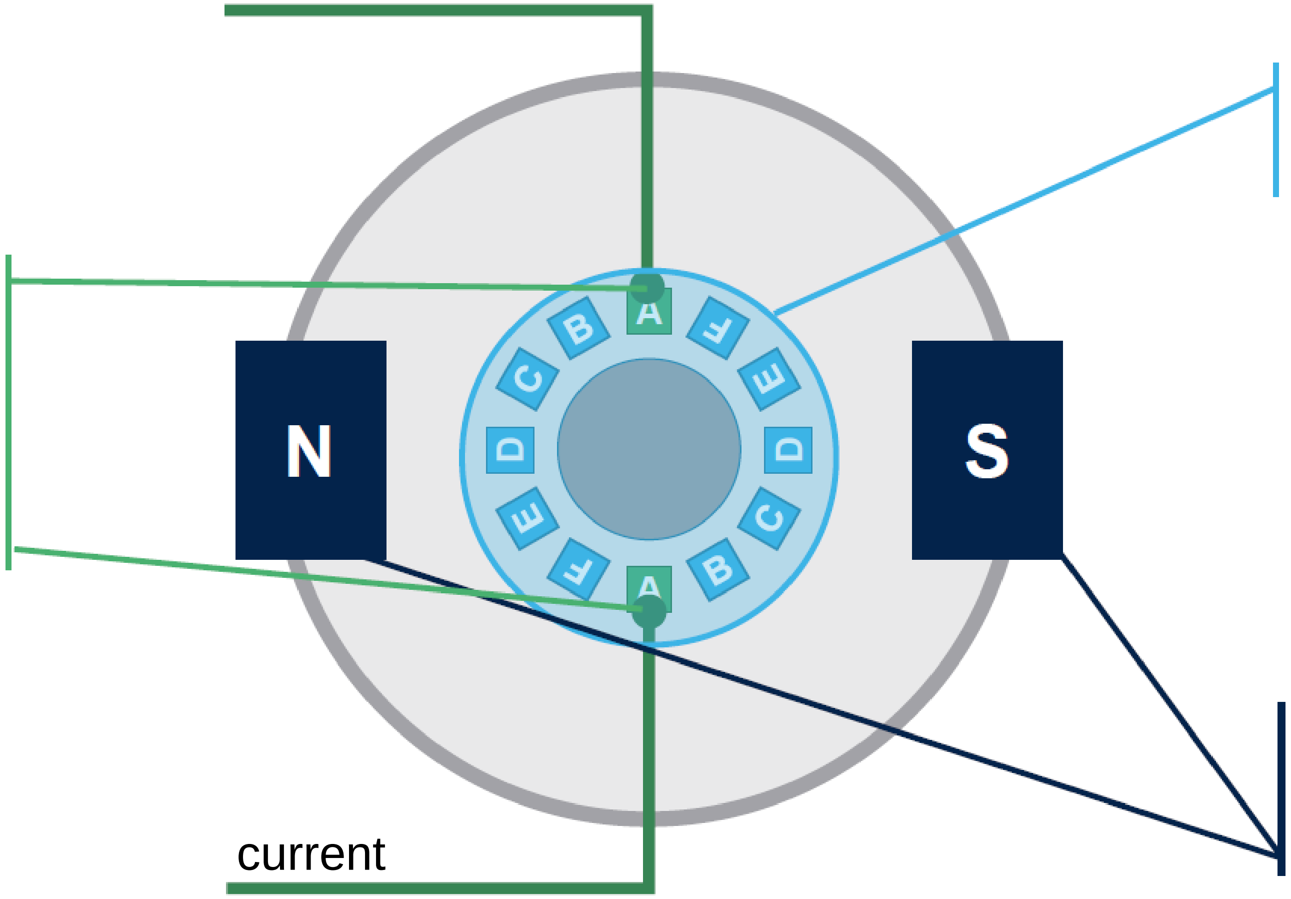
Credits:
STMicroelectronics

Project-based Learning Center | ETH Zurich
Tommaso Polonelli tommaso.polonelli@pbl.ee.ethz.ch
Michele Magno michele.magno@pbl.ee.ethz.ch
Vlad Niculescu vladn@iis.ee.ethz.ch

Basics - mechanical



The rotor coils are sequentially connected to the motor leads through mechanical switches (brushes)

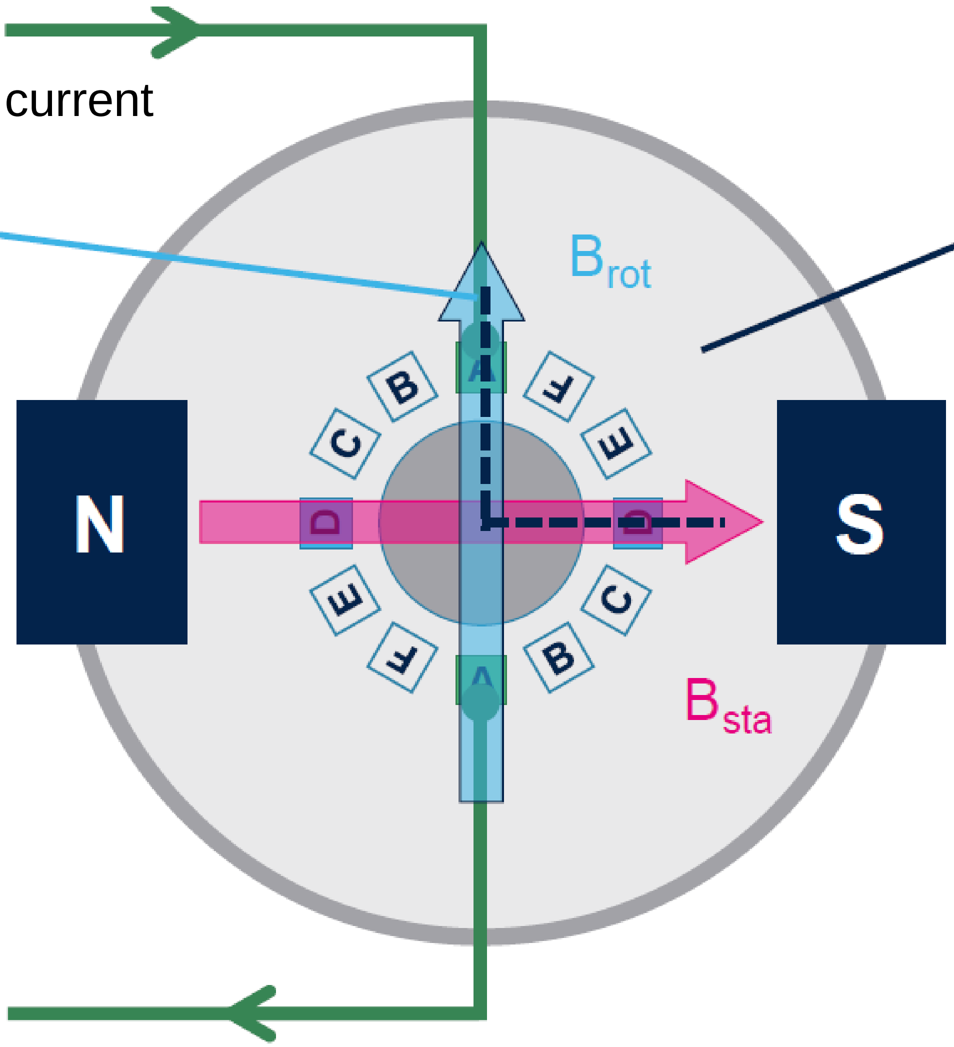


The rotor is composed by a group of coils

The stator magnetic field is generated by a permanent magnet

Basics – magnetic fields and load angle

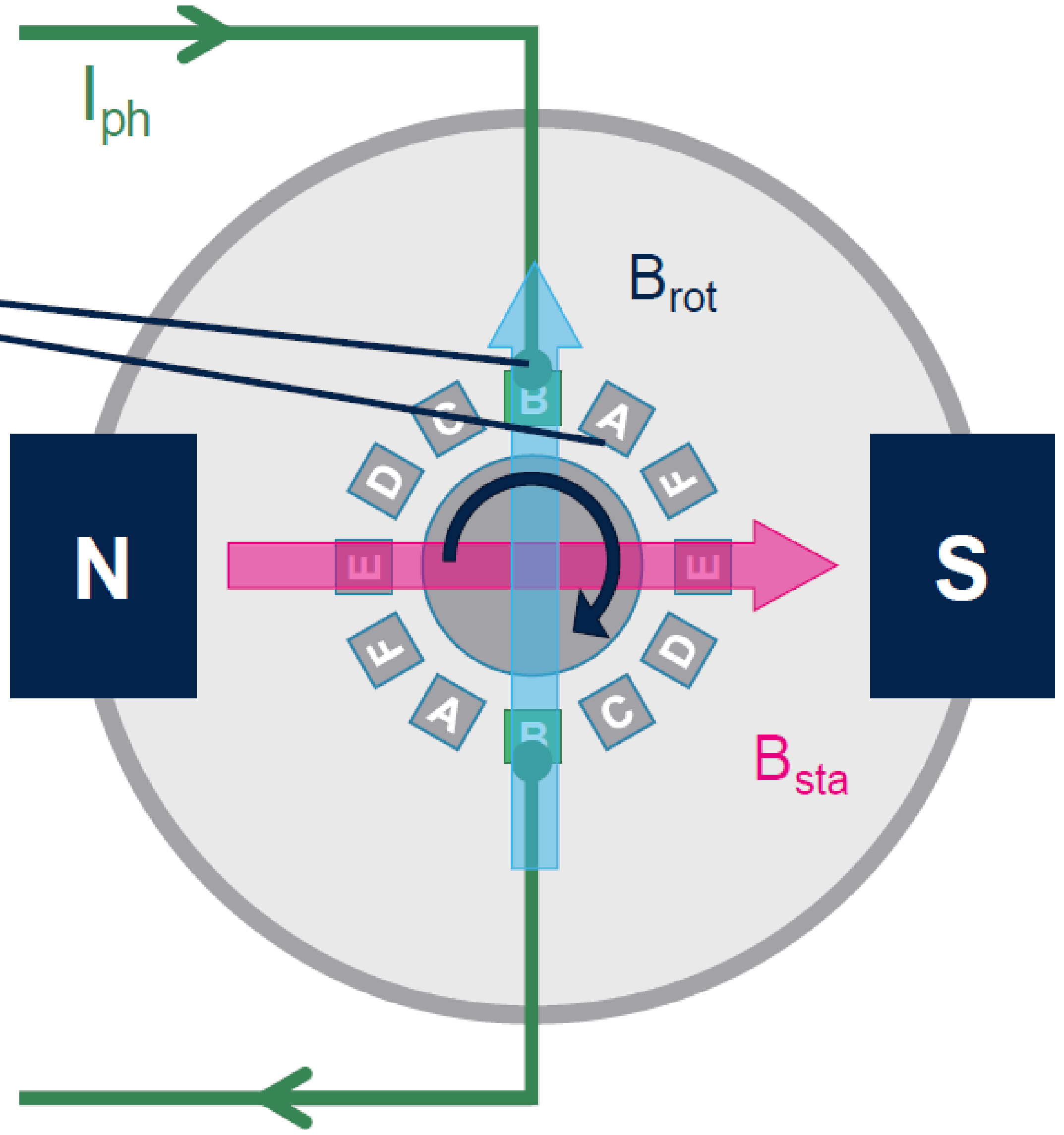
Forcing a current in the motor leads the rotor magnetic field is generated



The torque applied to the rotor is the highest possible because the load angle (θ) is about 90°

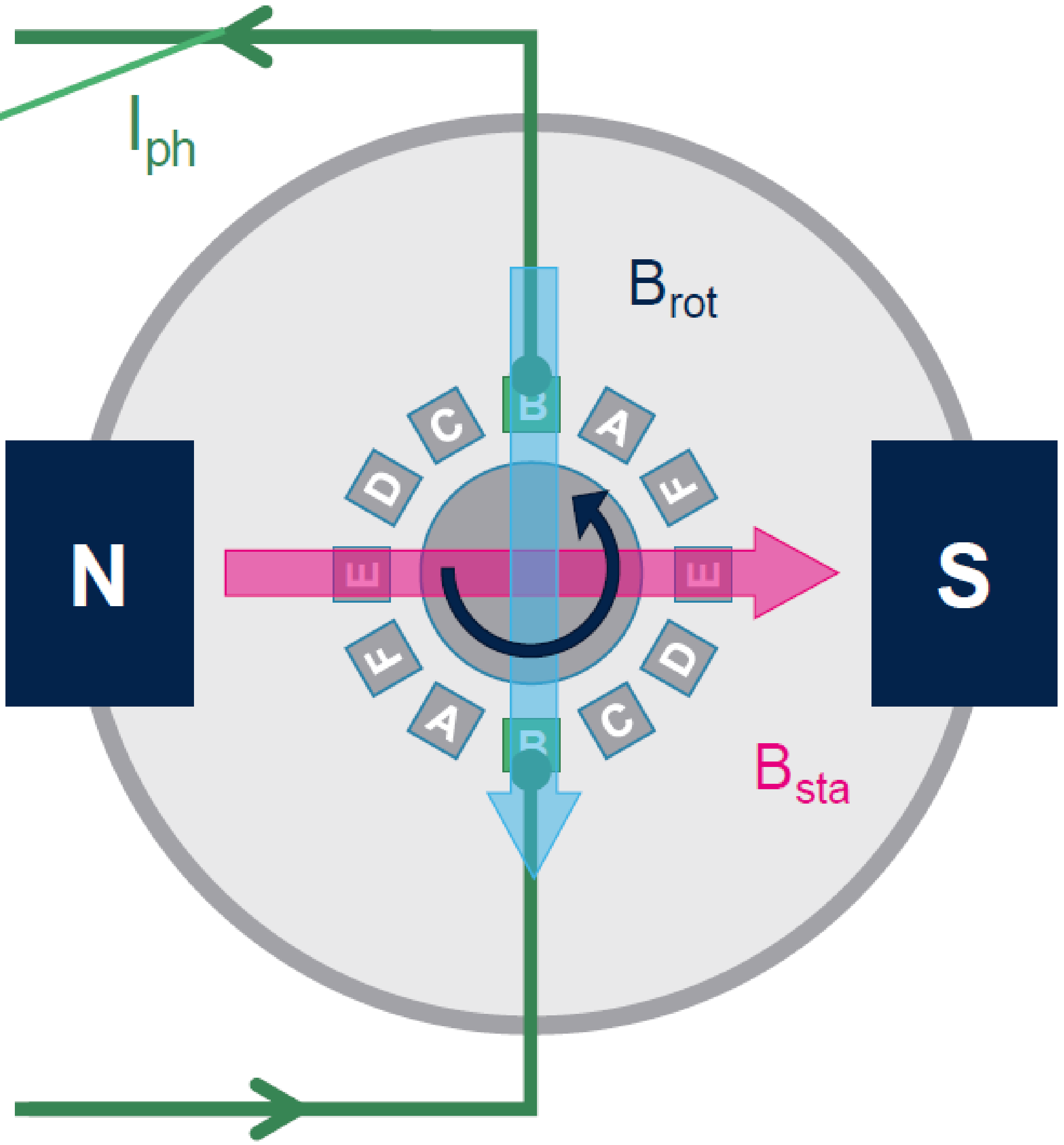
Basics – brushes and rotation

The brushes connect the motor leads to the next coil **(B)** keeping the load angle almost equal to 90° during rotation



Basics – brushes and rotation

Changing the current direction the motor rotation is reversed



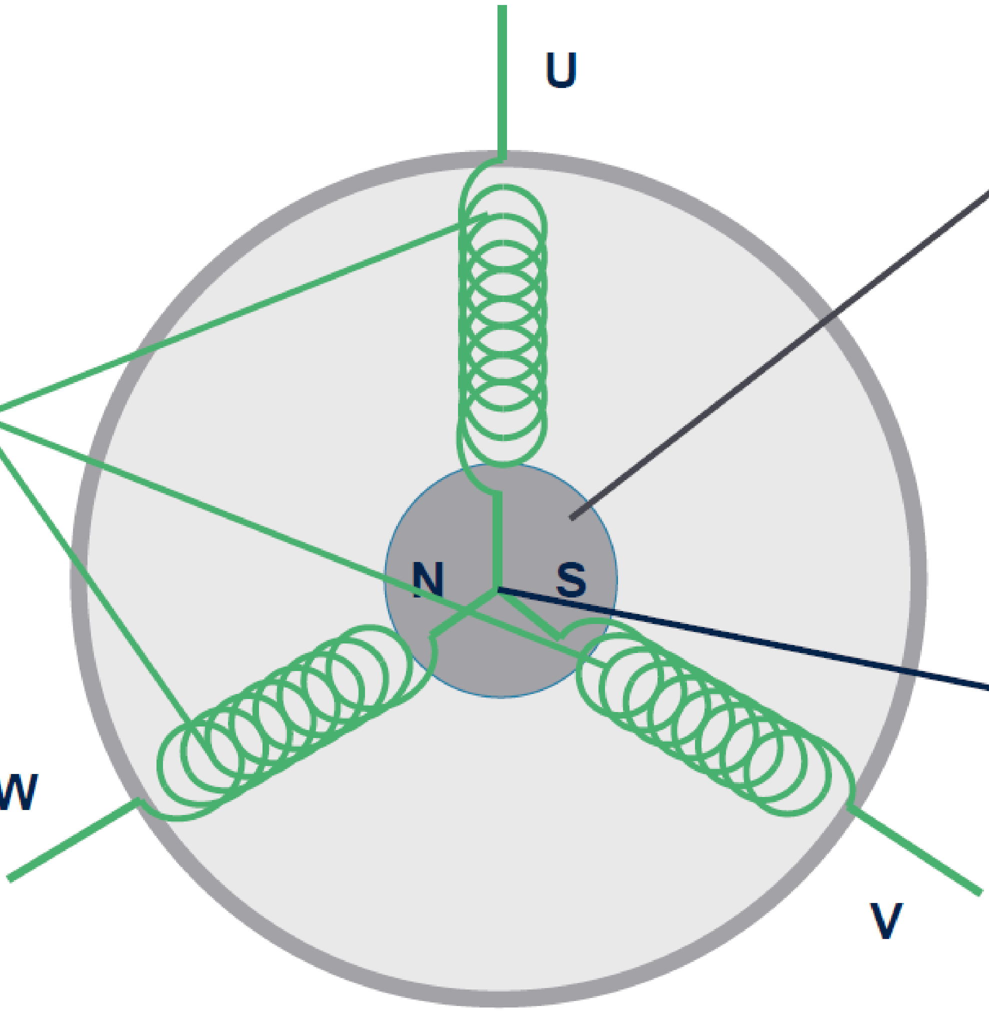
Brush DC motor summary

The electric motor operation is based on the following points:

- The magnetic field intensity is proportional to the current forced into the motor leads.
- The magnetic field rotation is automatically obtained commutating the active coil through mechanical switches (brushes).
- The load angle is almost constant, and it is about 90° allowing the maximum efficiency (current vs. torque proportion).
- The motor is controlled applying a voltage on the motor leads. The higher the voltage, the higher the speed. The direction is changed reversing the polarity on the leads.
- The **maximum torque** is limited by the current rating of the motor and it is obtained at zero speed (start-up).
- The **maximum speed** is limited by the supply voltage and it is obtained when no load torque is present.

Three-phase brushless DC motor

The stator is composed by three coils, named **phases**, positioned at 120° from each other



A permanent magnet generates the magnetic field of the rotor

The windings are connected by one of the sides.
The sum of the currents is zero

life.augmented

ETH zürich

Timers

Credits:
STMicroelectronics

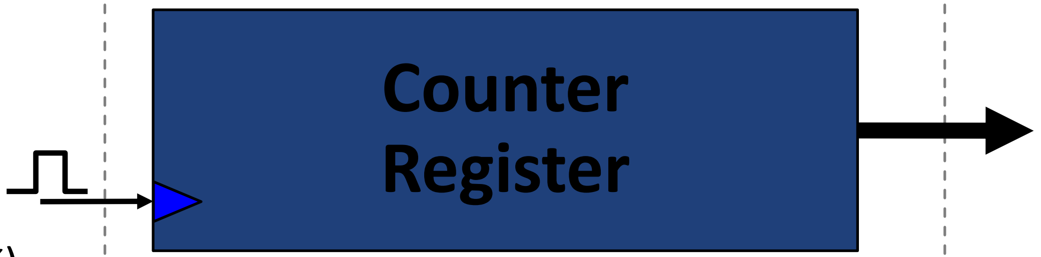
Project-based Learning Center | ETH Zurich
Tommaso Polonelli tommaso.polonelli@pbl.ee.ethz.ch
Michele Magno michele.magno@pbl.ee.ethz.ch
Vlad Niculescu vladn@iis.ee.ethz.ch

Timers

- Correct system timing is a fundamental requirement for the proper operation of a real-time application;
  - If the timing is incorrect, the input data may be processed after the output was updated
- The timers may be driven from an internal or external clock;
- Usually timers include multiple independent capture and compare blocks, with interrupt capabilities;
- Main applications:
  - Generate events of fixed-time period;
  - Allow periodic wake-up from sleep;
  - Count external signals/events;
  - Signal generation (Pulse Width Modulation – PWM);
  - Replacing delay loops with timer calls allows the CPU to sleep between operations, thus consuming less power.

Timer/Counter Basics

Clock Input
- Clock
- GPIO Pin (CLK)



Counter
Overflow Action
- Interrupt

Input

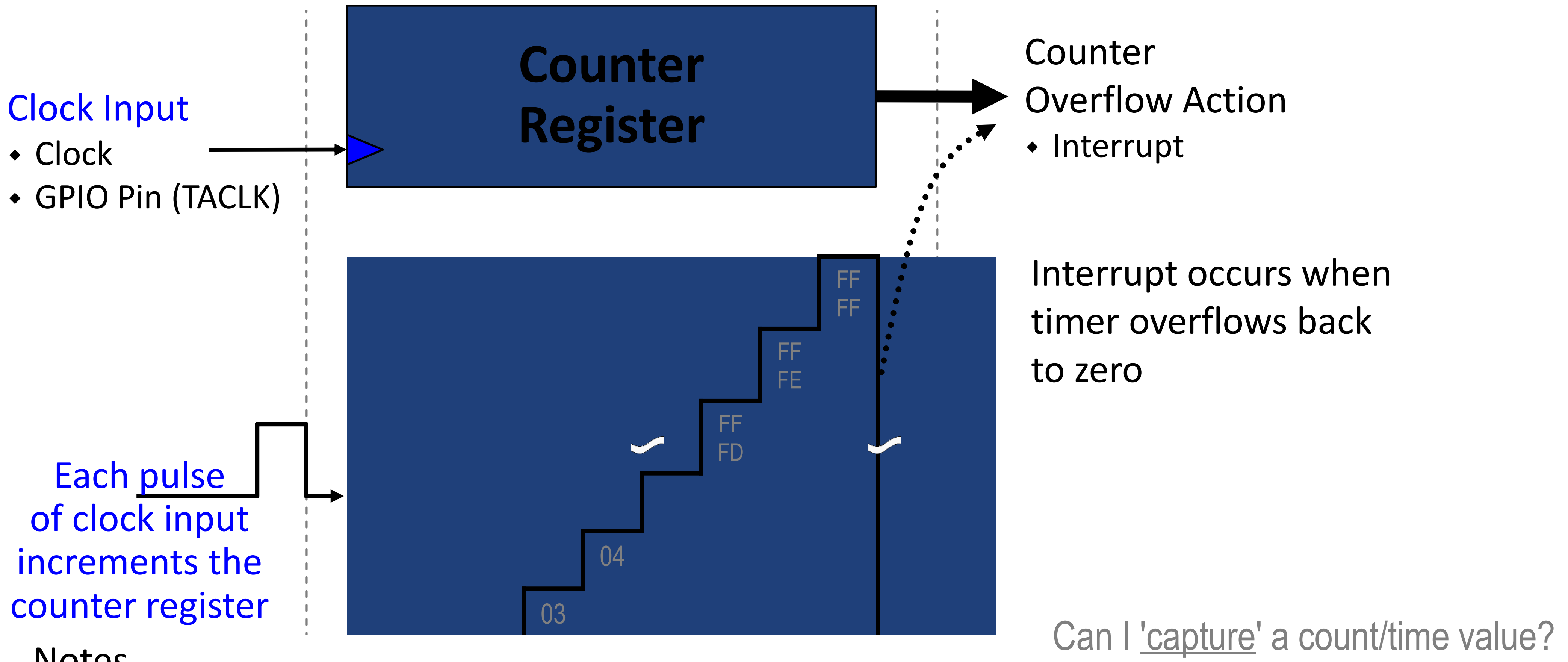
Counter

Action

Notes
- Timers are often called “Timer/Counters” as a counter is the essential element
- “Timing” is based on counting inputs from a known clock rate

What happens on each clock input?

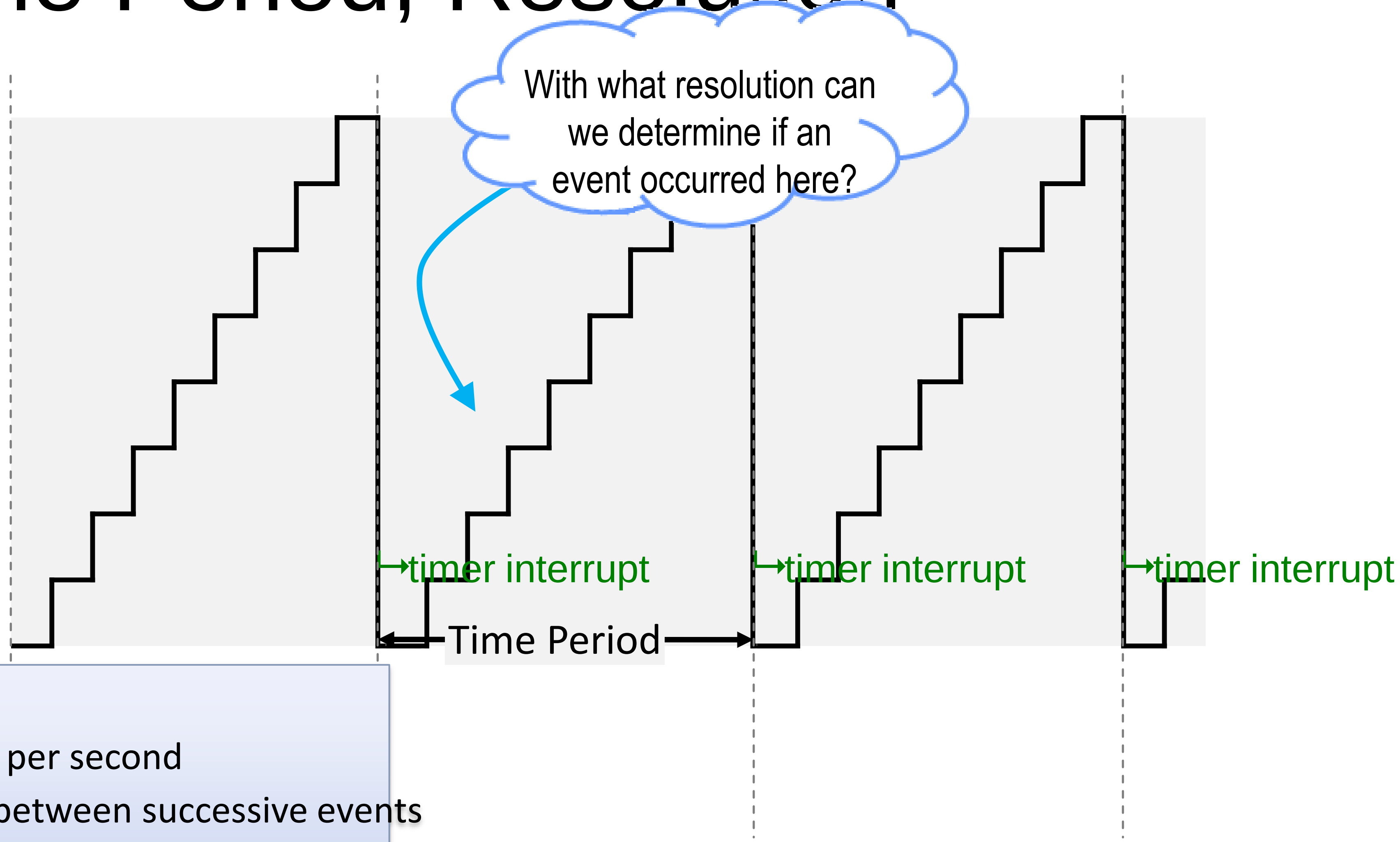
Timer/Counter Basics


Counter Register
Clock Input
◆ Clock
◆ GPIO Pin (TACLK)
Counter Overflow Action
◆ Interrupt
Interrupt occurs when timer overflows back to zero
Each pulse of clock input increments the counter register
03
04
FF FD
FF FE
FF FF

Can I 'capture' a count/time value?

Notes
- Timers are often called "Timer/Counters" as a counter is the essential element
- "Timing" is based on counting inputs from a known clock rate
- Actions don't occur when writing value to counter

Frequency, Time Period, Resolution

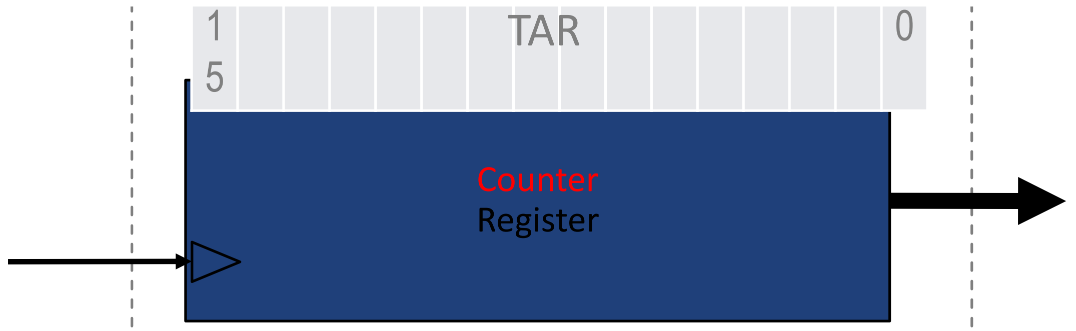

With what resolution can we determine if an event occurred here?
timer interrupt
timer interrupt
timer interrupt
Time Period

Definitions

Frequency:	How many times per second
Time Period:	Amount of time between successive events
Resolution:	Granularity in determining system events

Capture Basics

Clock Input
- Clock
- GPIO Pin

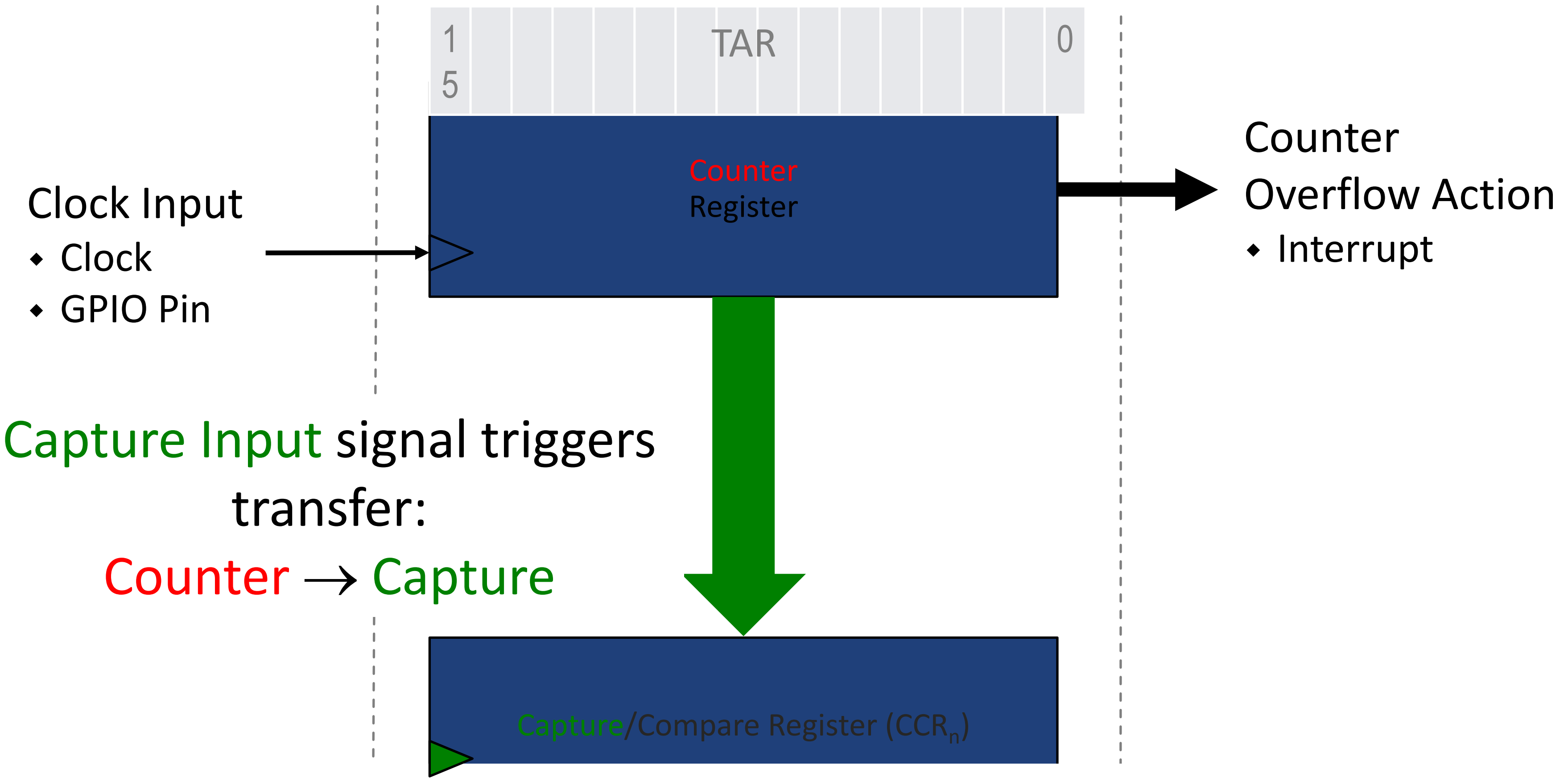


Counter
Overflow Action
- Interrupt (

Alternatively, use CCR for compare...

Notes
- Capture time (i.e. count value) when Capture Input signal occurs

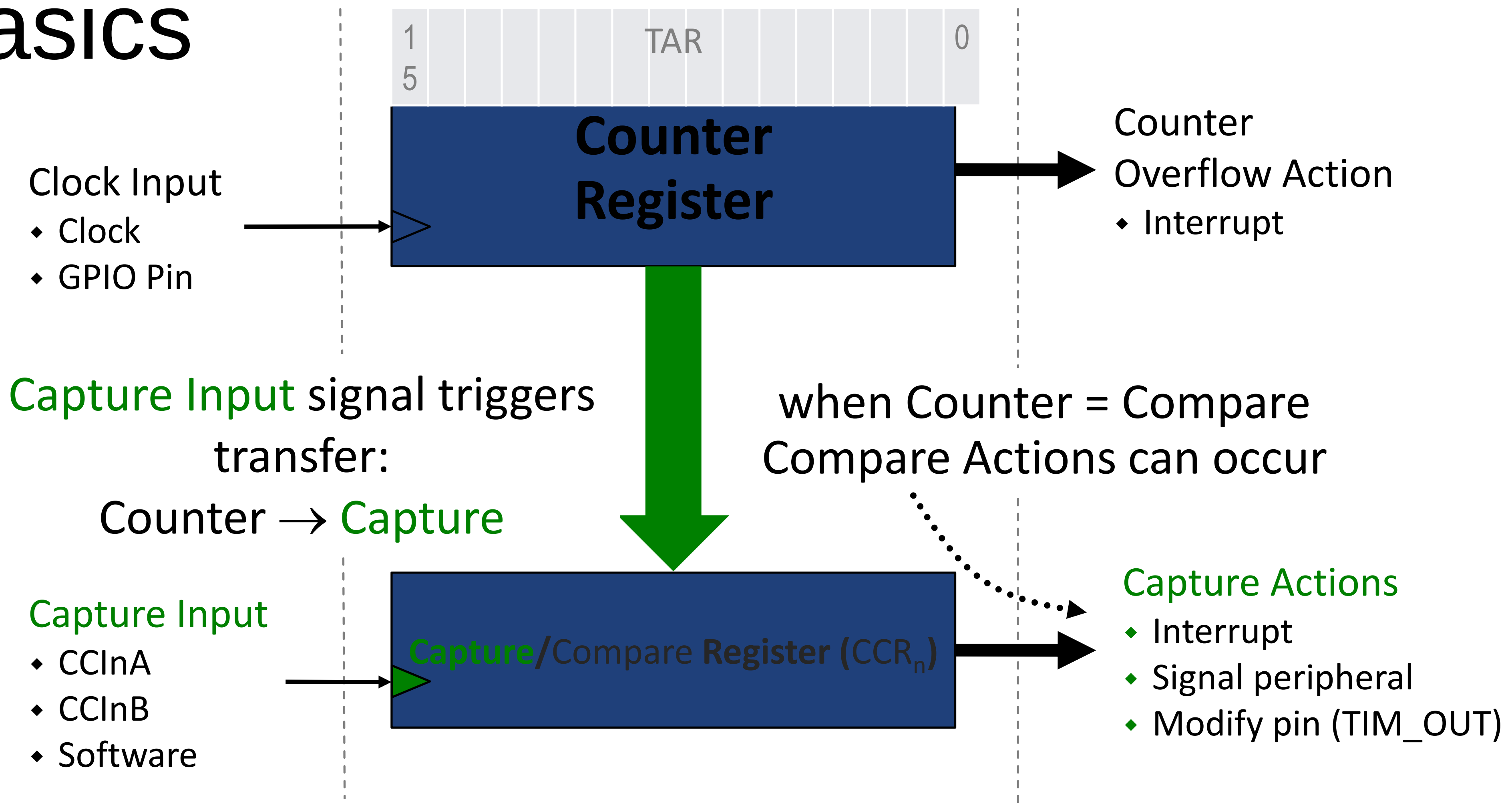
Capture Basics



Notes

- Capture time (i.e. count value) when Capture Input signal occurs
- When capture is triggered, count value is placed in CCR and an interrupt is generated

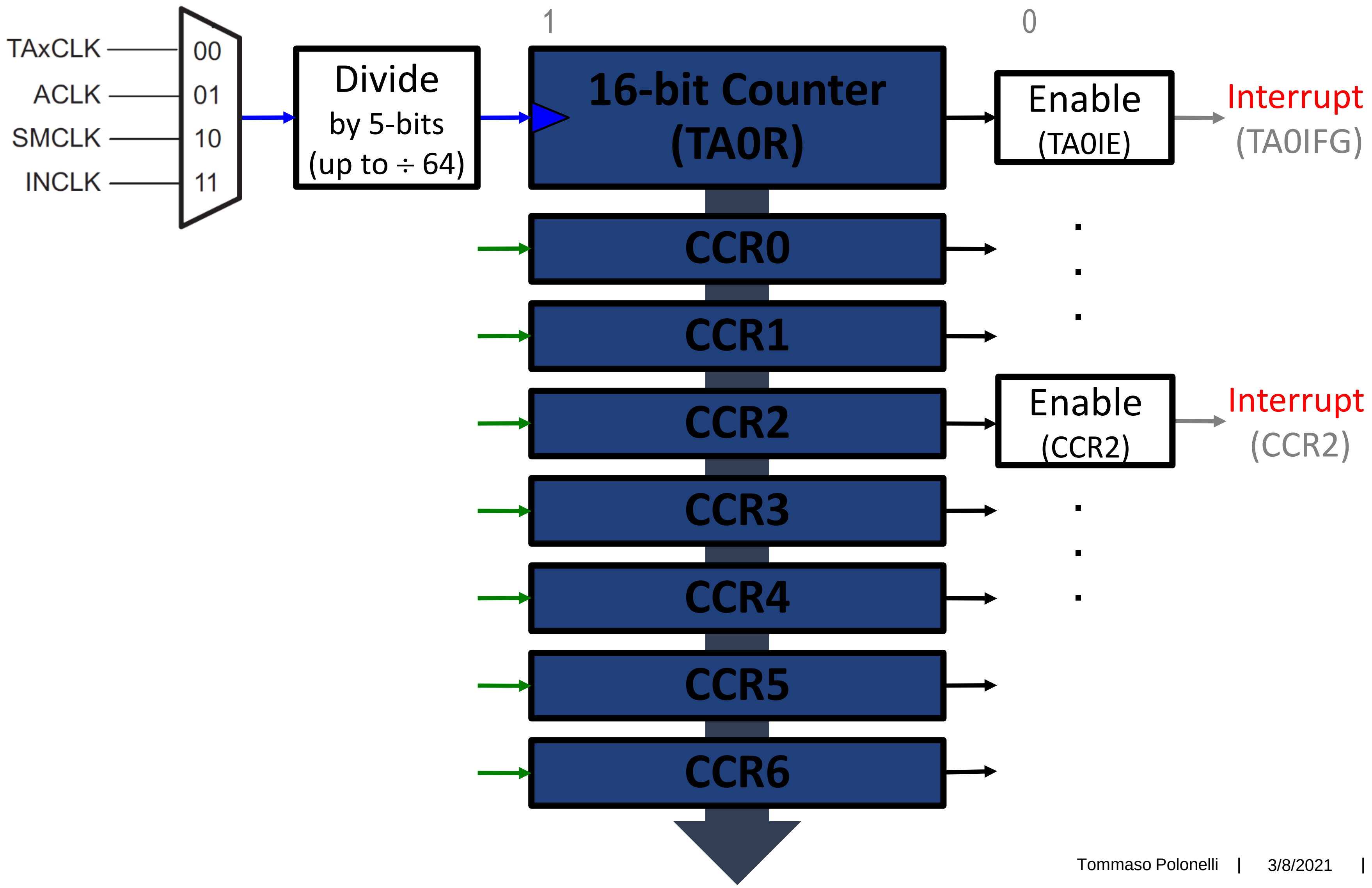
Capture Basics



Notes

- Capture time (i.e. count value) when Capture Input signal occurs
- When capture is triggered, count value is placed in CCR and an interrupt is generated
- Capture Overflow (COV): indicates 2nd capture to CCR before 1st was read

Example.


TAxCLK
ACLK
SMCLK
INCLK
00
01
10
11
Divide by 5-bits (up to ÷ 64)
1
0
16-bit Counter (TA0R)
Enable (TA0IE)
Interrupt (TA0IFG)
CCR0
CCR1
CCR2
CCR3
CCR4
CCR5
CCR6
Enable (CCR2)
Interrupt (CCR2)

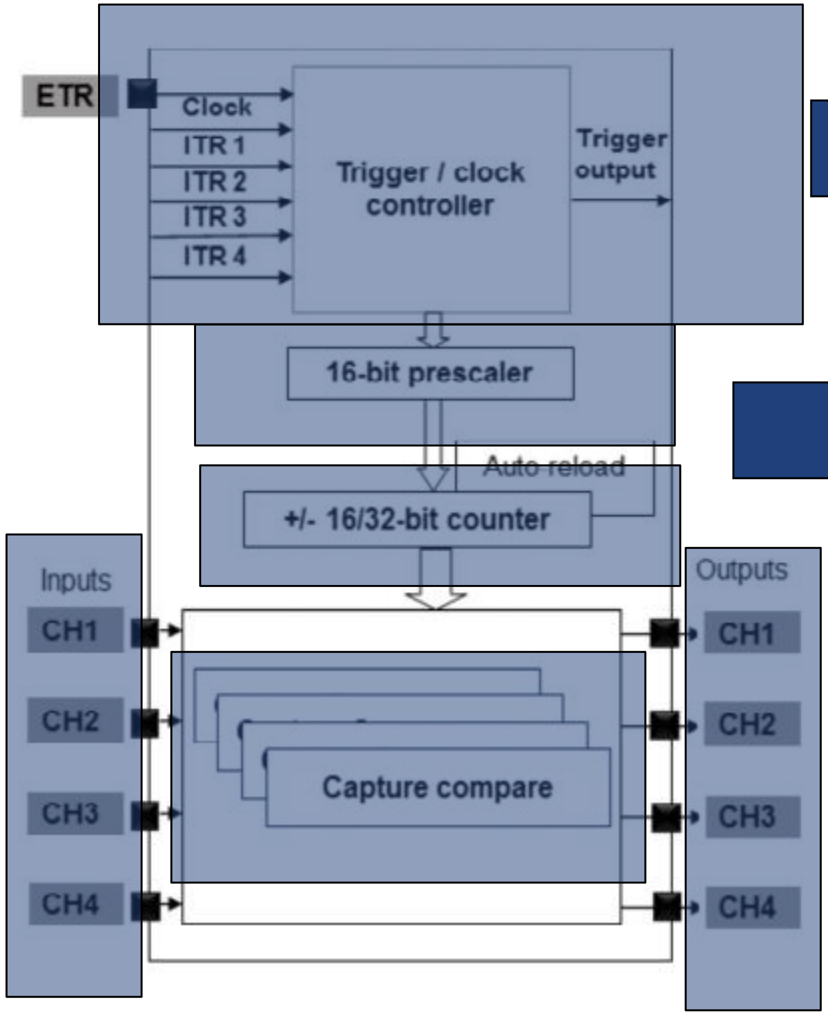
Timers – STM32

- The general-purpose timers consist of a **16-bit (or 32bits) 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.

- General-purpose TIMx timer features include:
  - 16/32-bit up, down, up/down auto-reload counter.
  - 16/32-bit programmable prescaler used to divide (also “on the fly”) the counter clock frequency by any factor between 1 and 65535.
  - **Up to 4 independent channels** for:
    - Input capture
    - Output compare
    - PWM generation (Edge- and Center-aligned modes) / One-pulse mode output

Timers – Basic architecture STM32.



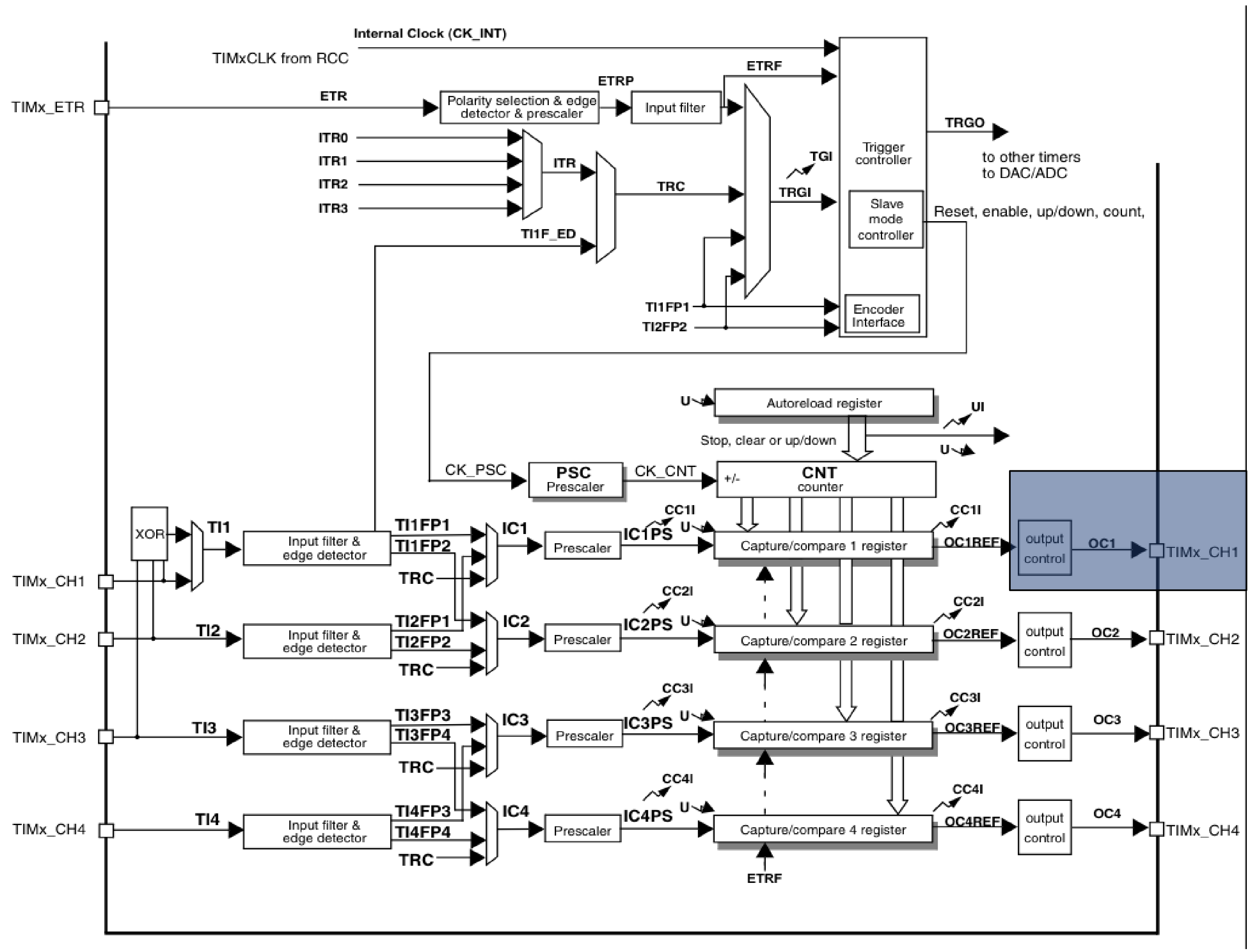
CLOCKS

CLOCKS / N

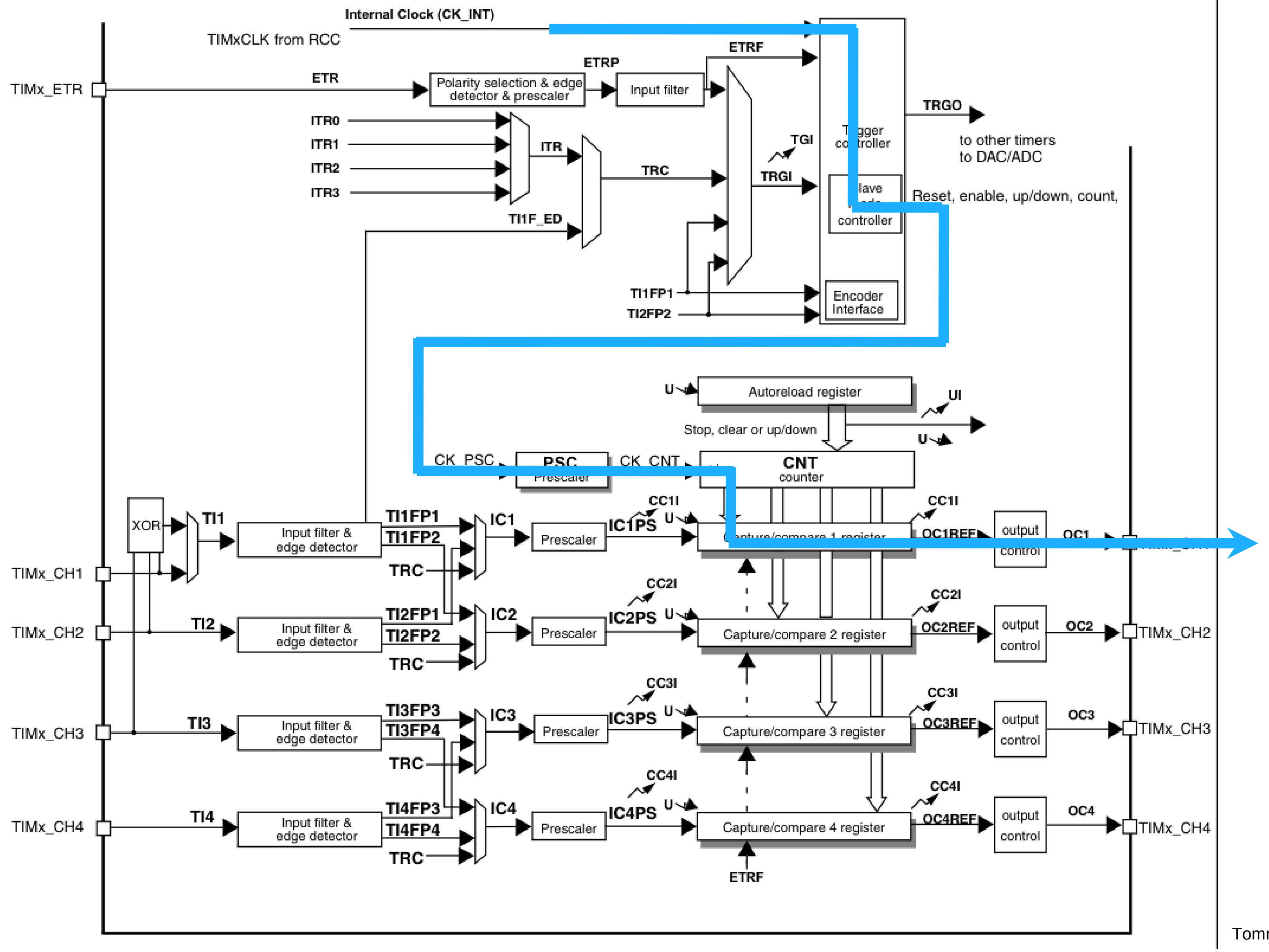
Counter/Auto-reload

Capture/compare

General-purpose timer block diagram

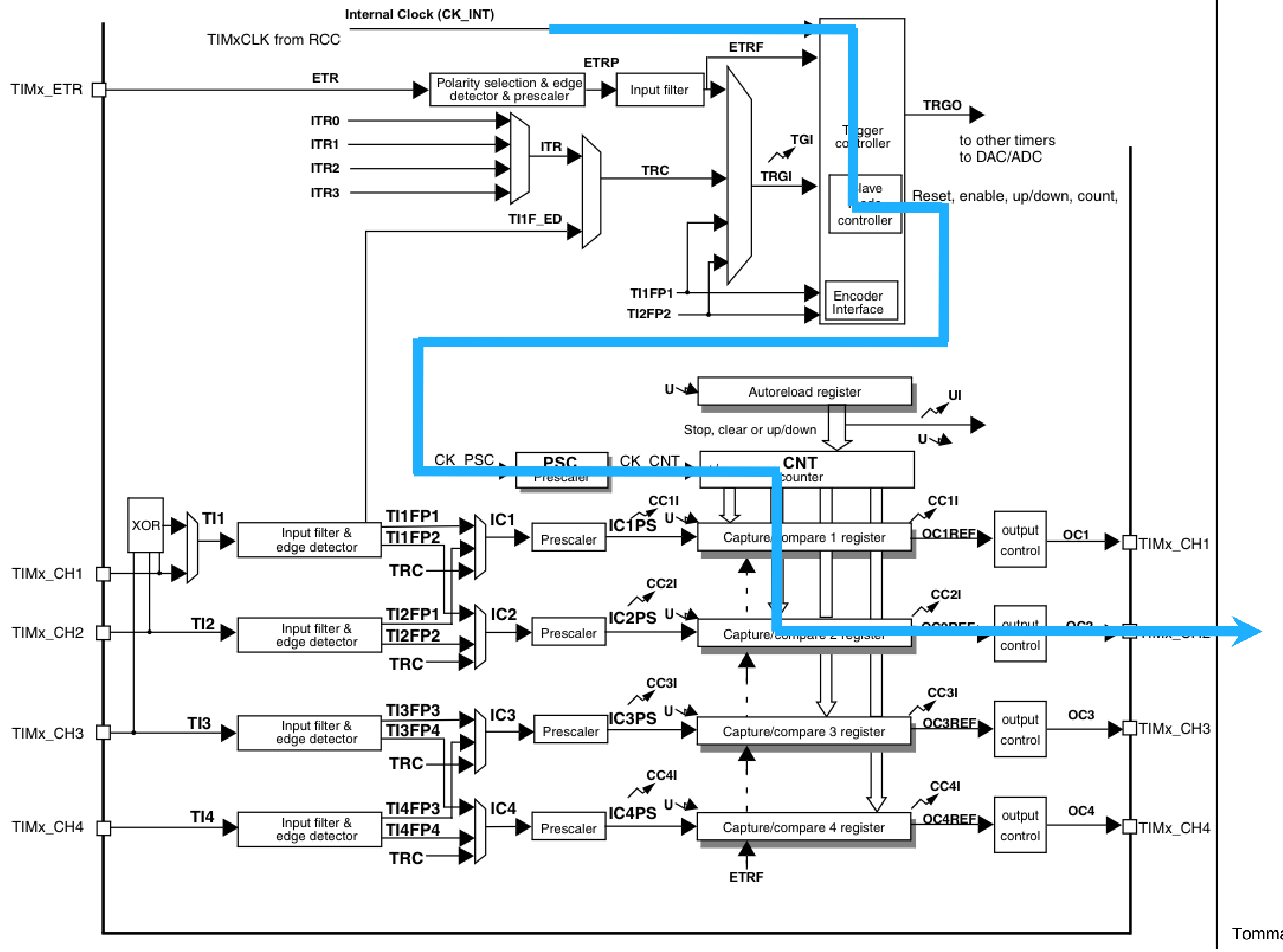


Timers



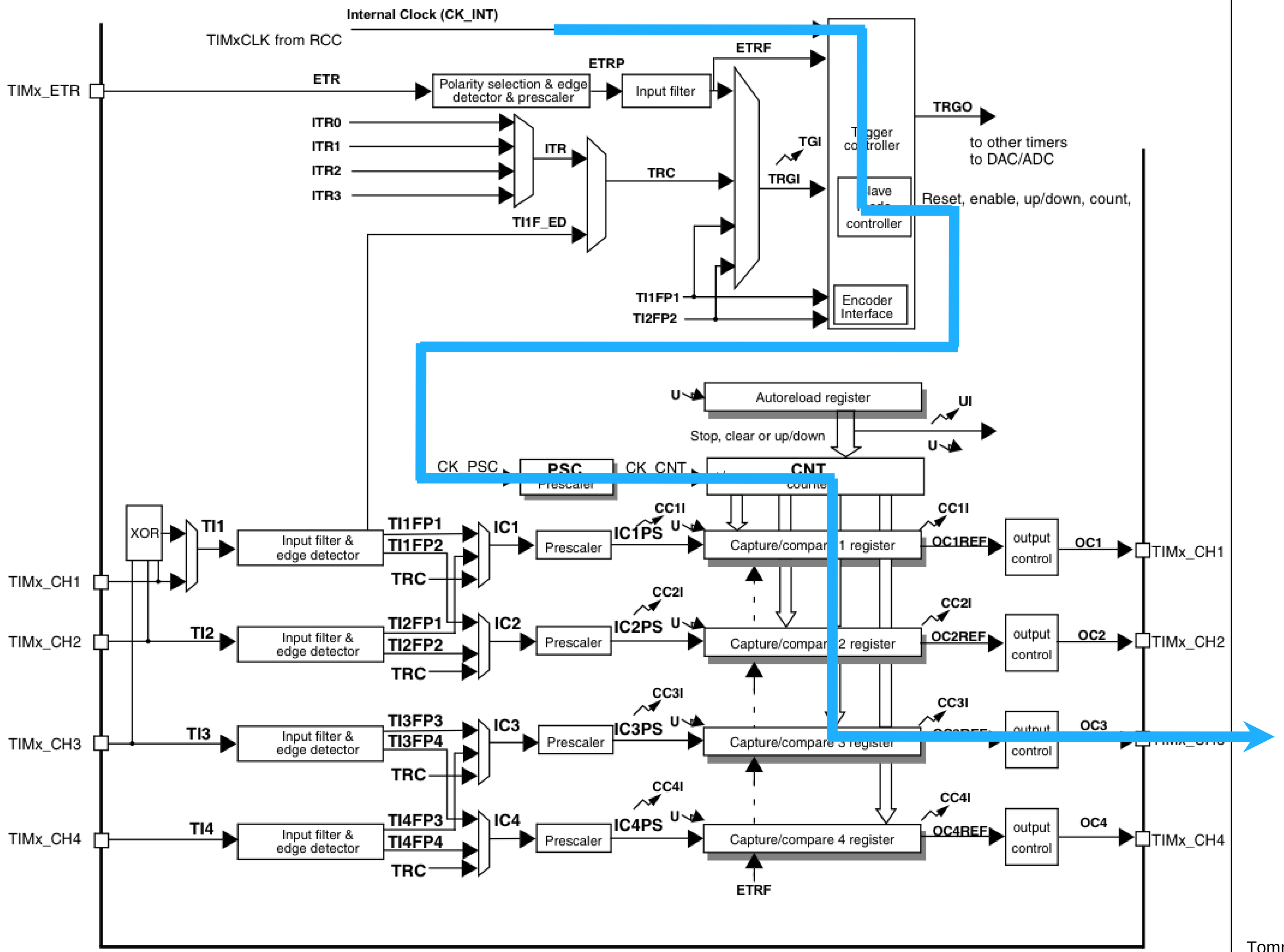
PATH: TimerX - Channel 1 - output compare

Timers



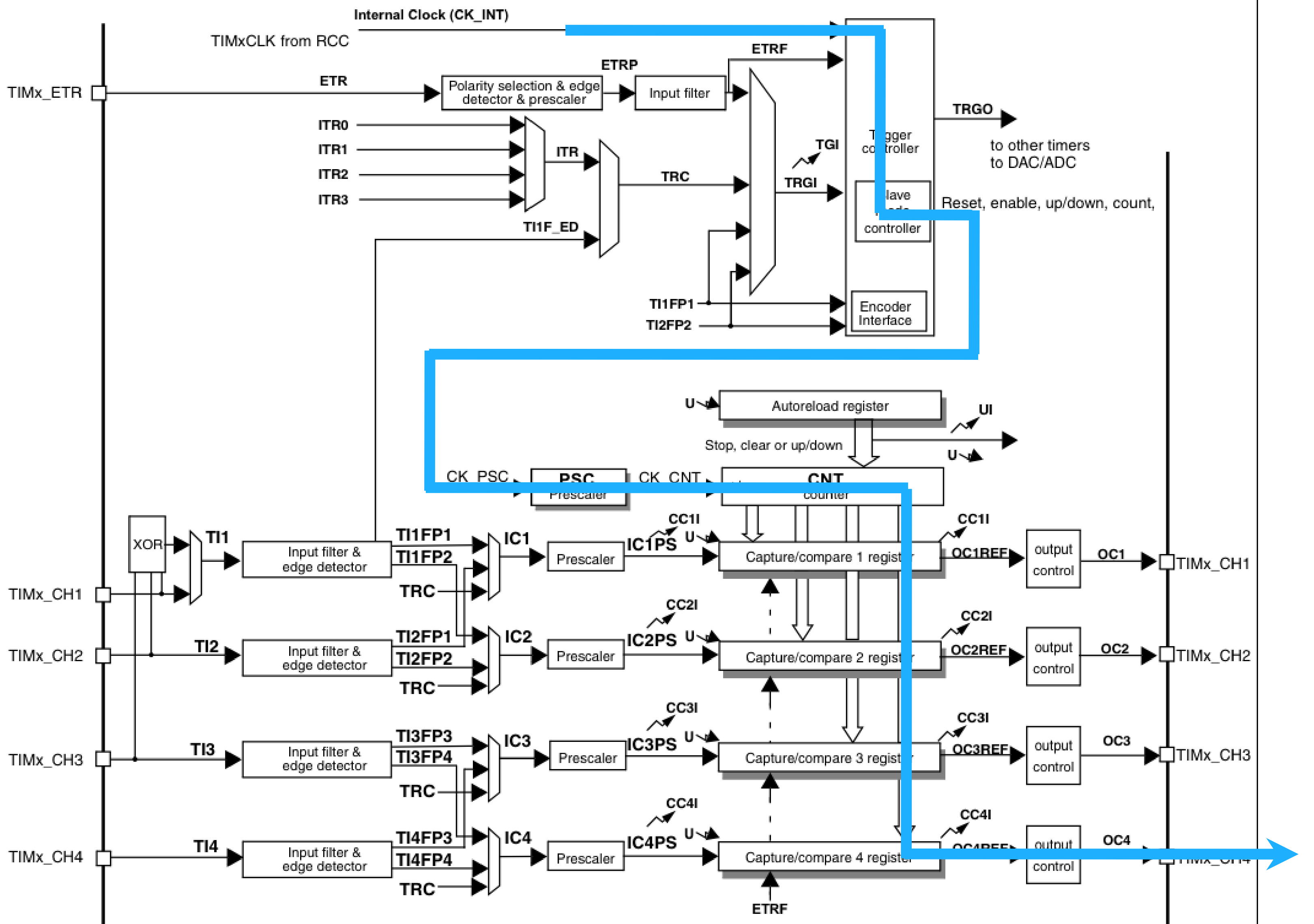
PATH: TimerX - Channel 2 - output compare

Timers



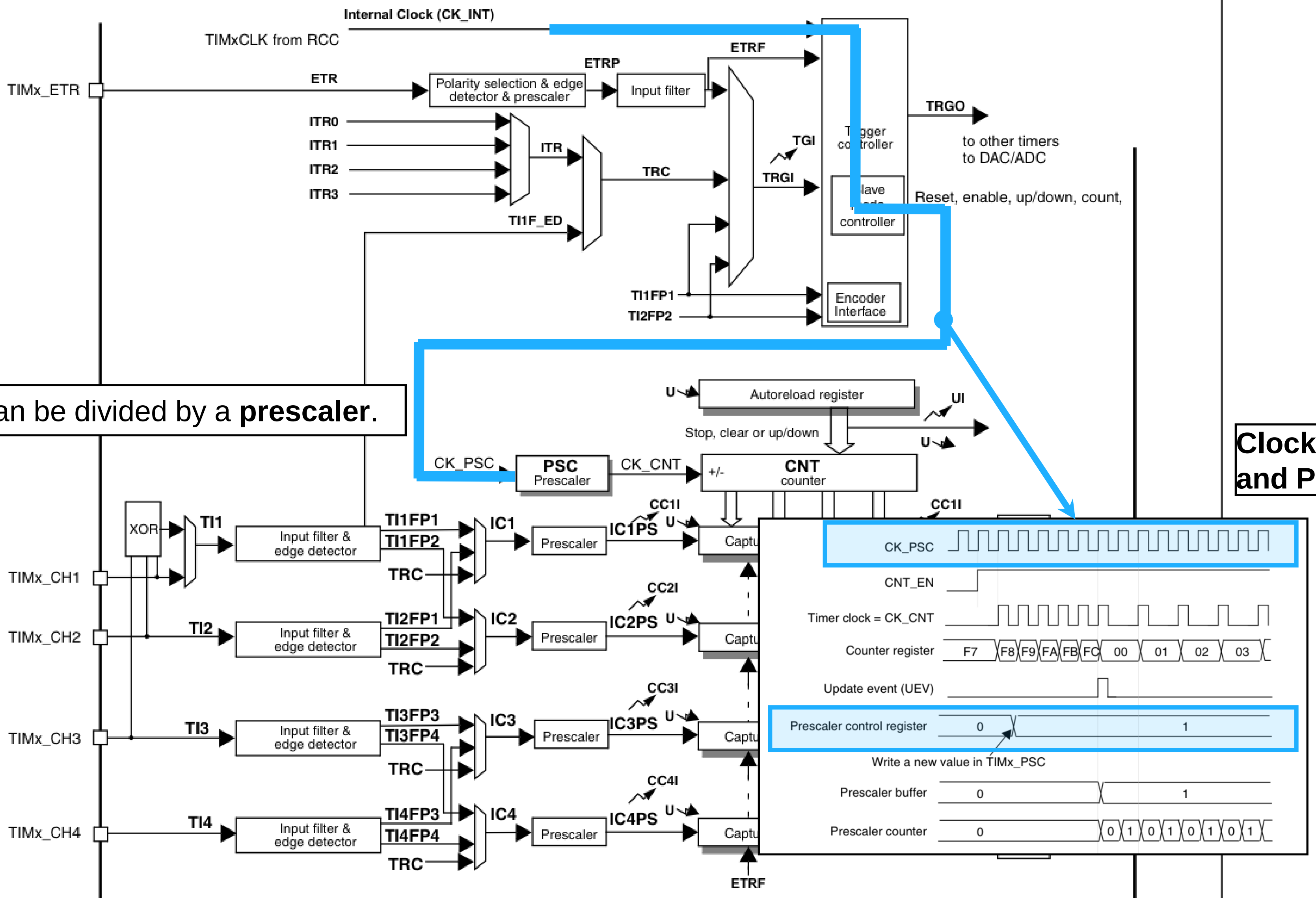
PATH: TimerX - Channel 3 - output compare

Timers



PATH: TimerX - Channel 4 - output compare

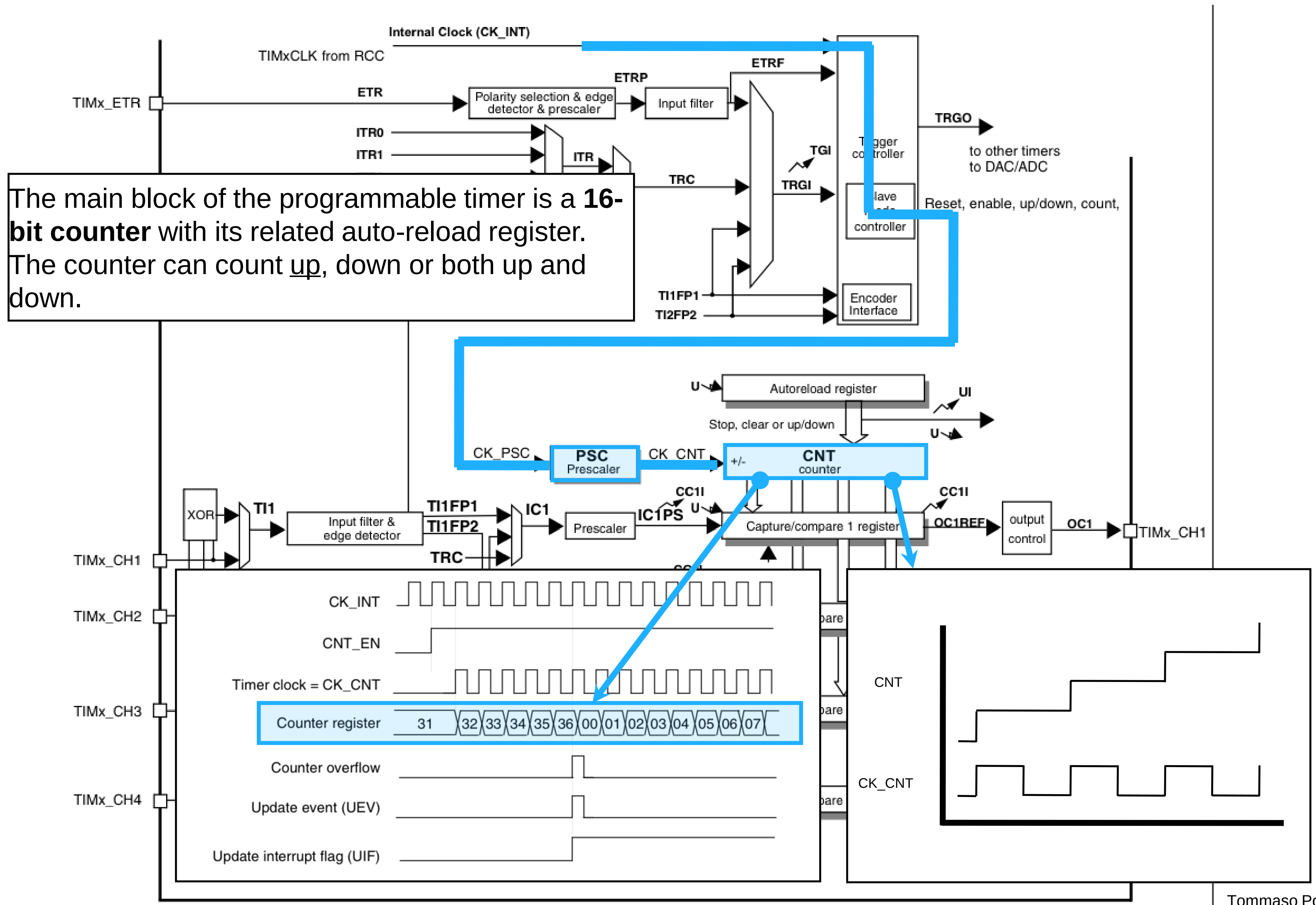
Timers



The counter clock can be divided by a **prescaler**.

Clock selection and Prescaler

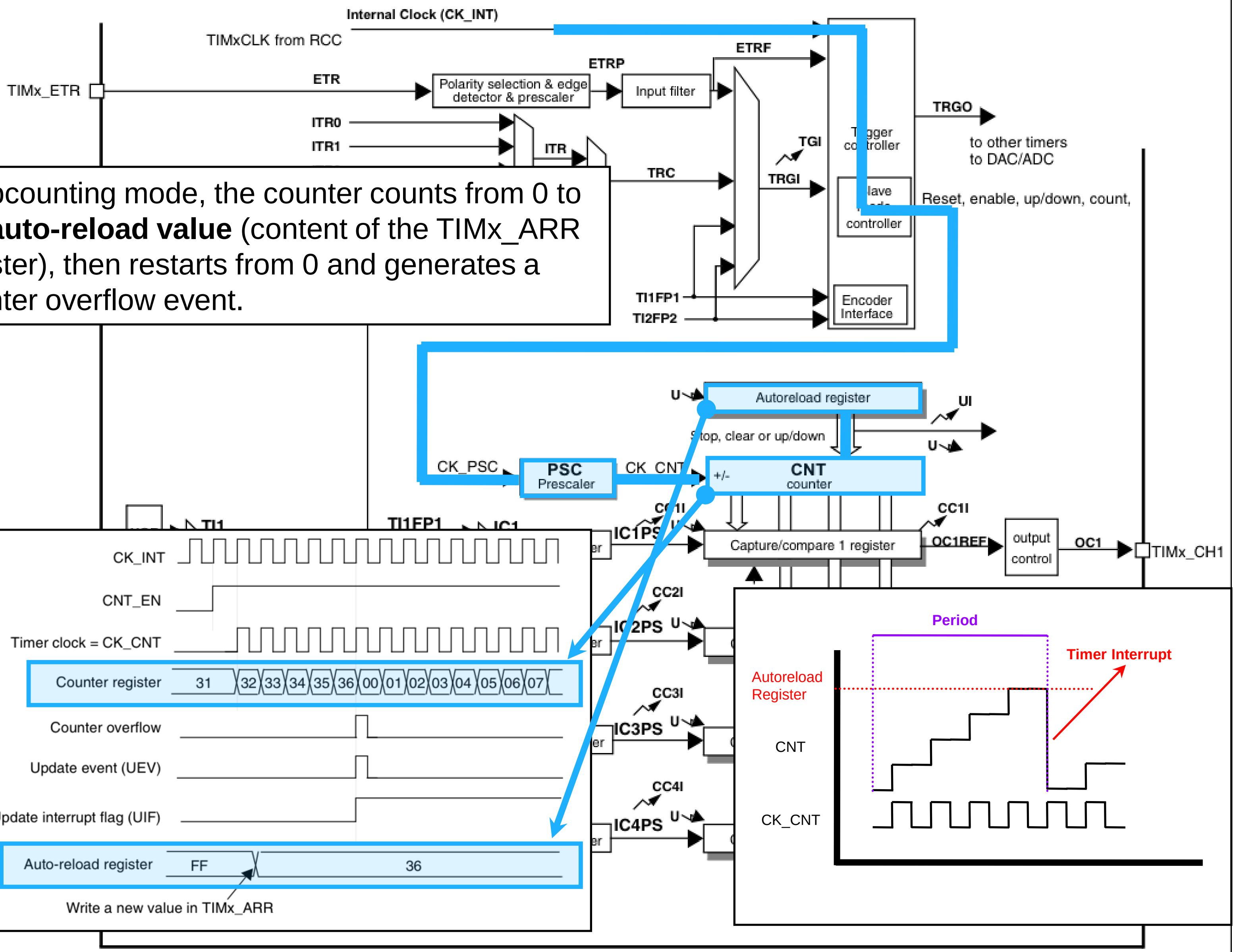
Timers



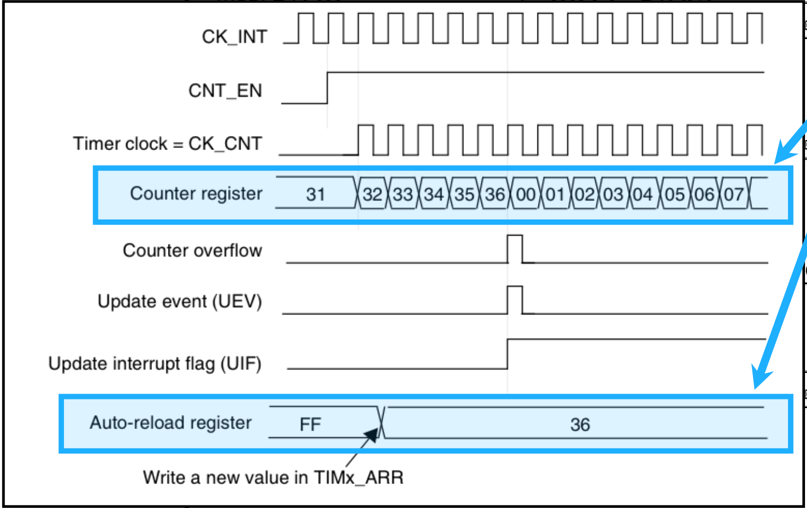
The main block of the programmable timer is a **16-bit counter** with its related auto-reload register. The counter can count up, down or both up and down.

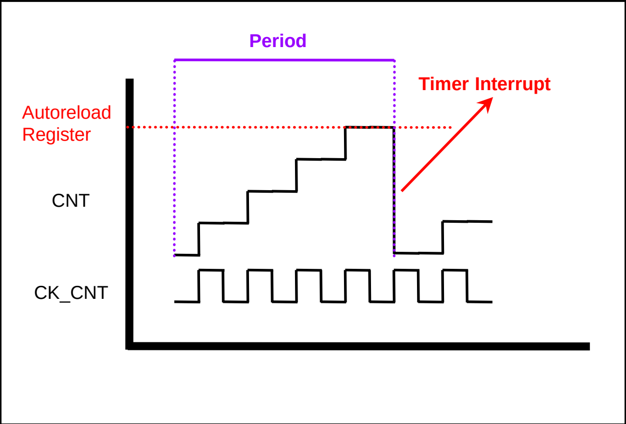
ETH zürich

Timers



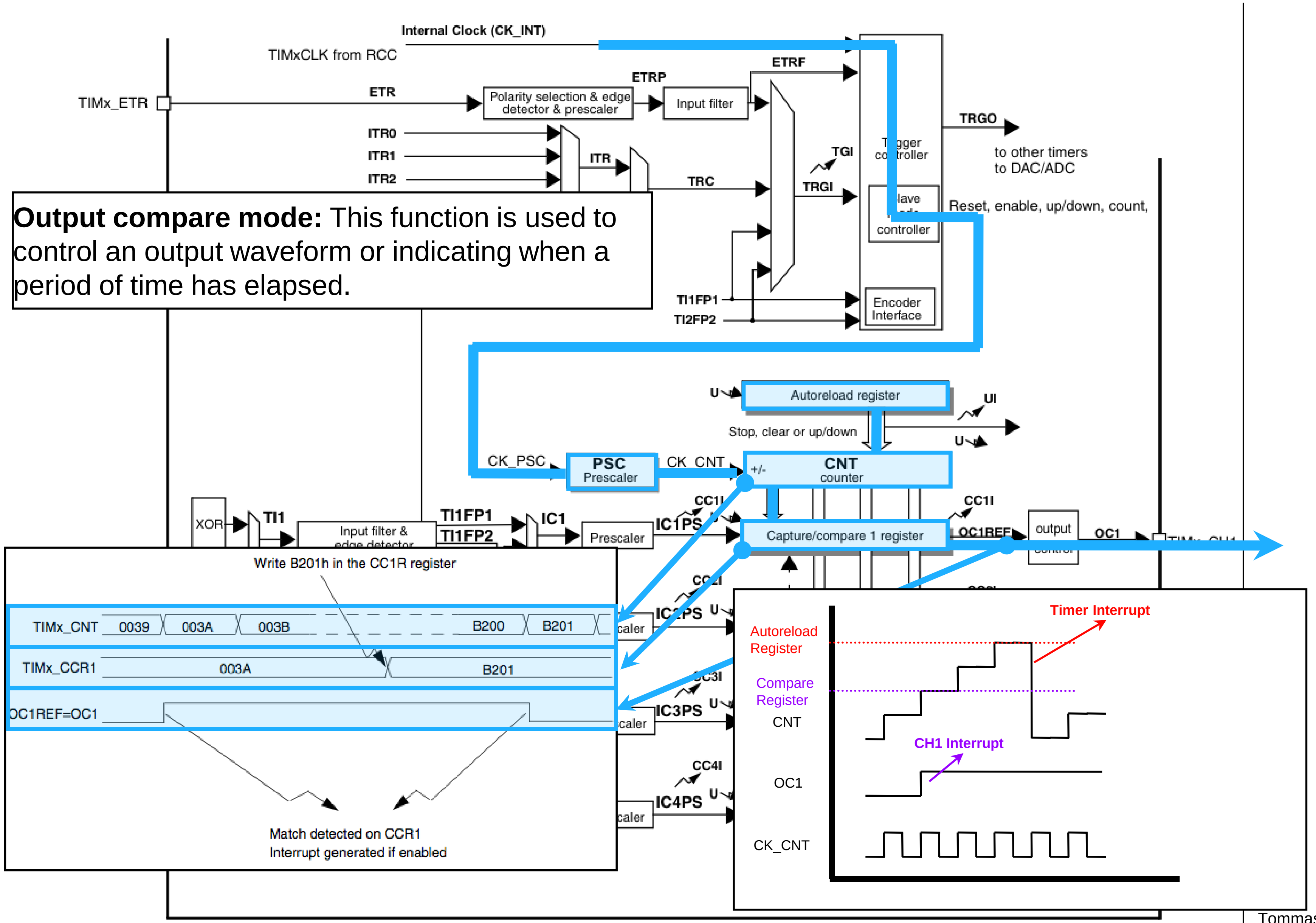
In upcounting mode, the counter counts from 0 to the **auto-reload value** (content of the TIMx_ARR register), then restarts from 0 and generates a counter overflow event.





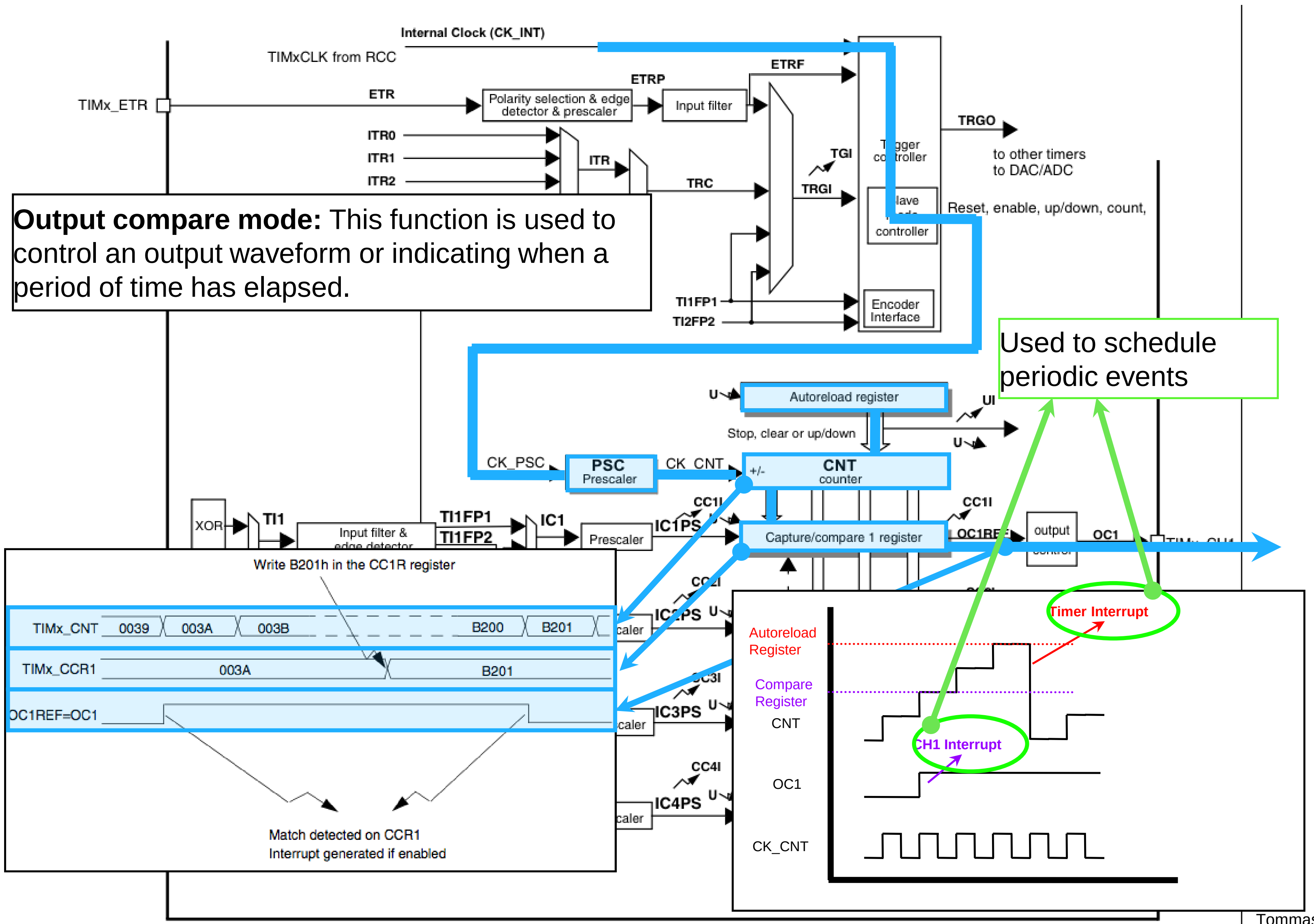
Timers

Output compare mode: This function is used to control an output waveform or indicating when a period of time has elapsed.



ETH zürich

Timers

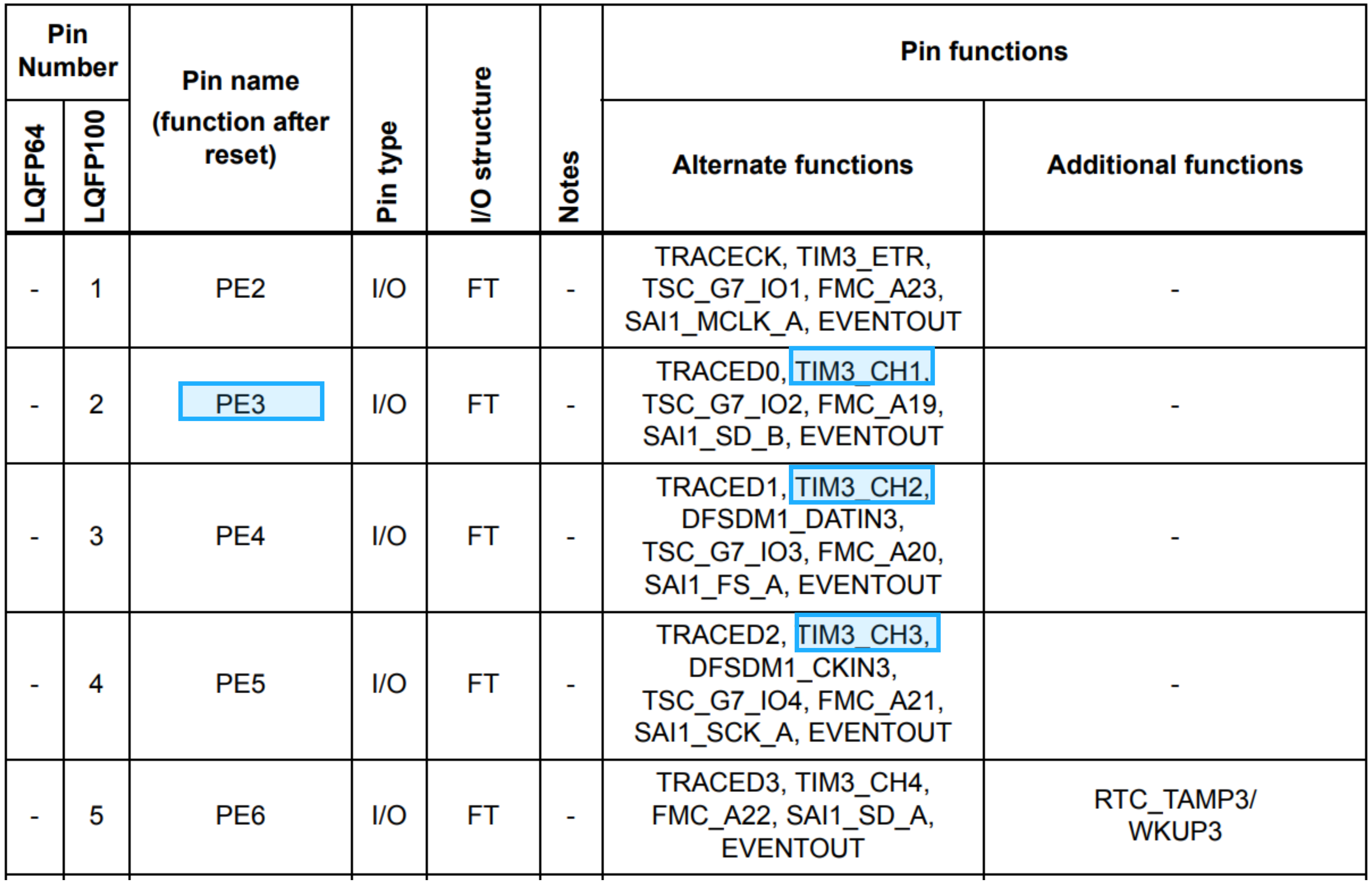


Output compare mode: This function is used to control an output waveform or indicating when a period of time has elapsed.

Used to schedule periodic events

STM32F4 Timers and Output Channels on GPIOs

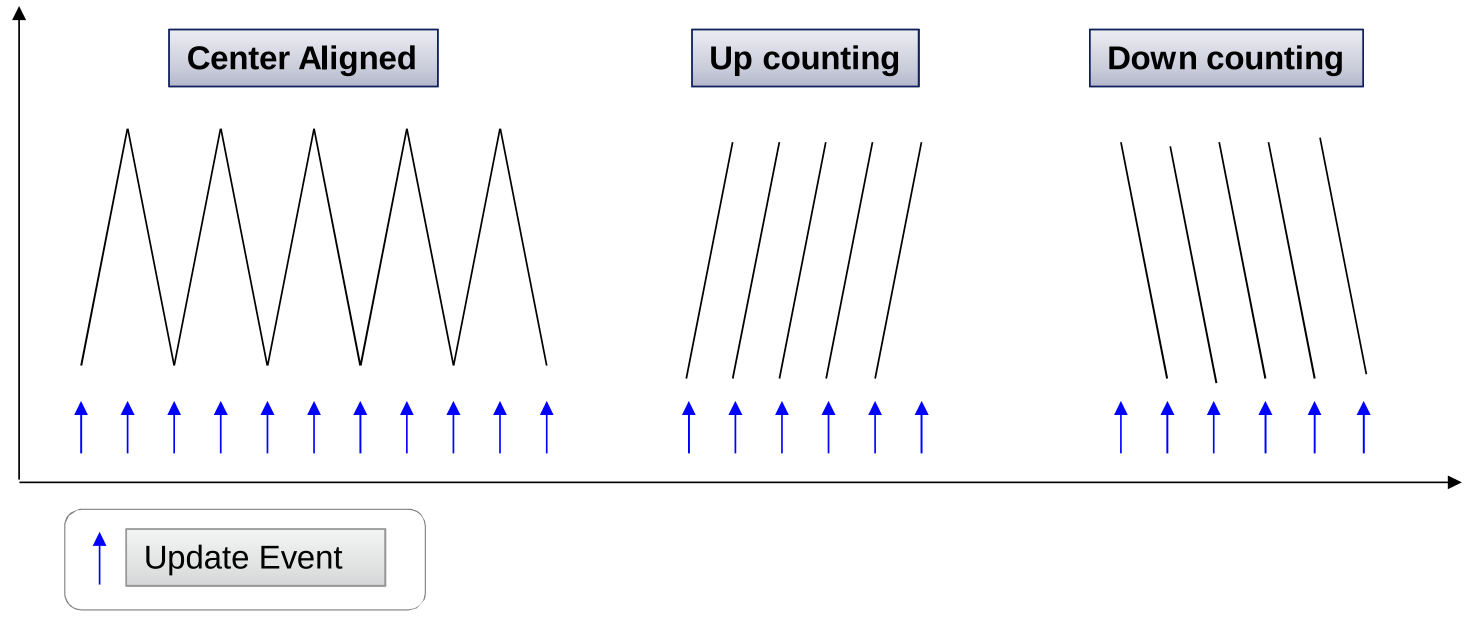
Table 16. STM32L475xx pin definitions



Pin Number		Pin name (function after reset)	Pin type	I/O structure	Notes	Pin functions	
LQFP64	LQFP100					Alternate functions	Additional functions
-	1	PE2	I/O	FT	-	TRACECK, TIM3_ETR, TSC_G7_IO1, FMC_A23, SAI1_MCLK_A, EVENTOUT	-
-	2	PE3	I/O	FT	-	TRACED0, TIM3_CH1, TSC_G7_IO2, FMC_A19, SAI1_SD_B, EVENTOUT	-
-	3	PE4	I/O	FT	-	TRACED1, TIM3_CH2, DFSDM1_DATIN3, TSC_G7_IO3, FMC_A20, SAI1_FS_A, EVENTOUT	-
-	4	PE5	I/O	FT	-	TRACED2, TIM3_CH3, DFSDM1_CKIN3, TSC_G7_IO4, FMC_A21, SAI1_SCK_A, EVENTOUT	-
-	5	PE6	I/O	FT	-	TRACED3, TIM3_CH4, FMC_A22, SAI1_SD_A, EVENTOUT	RTC_TAMP3/ WKUP3

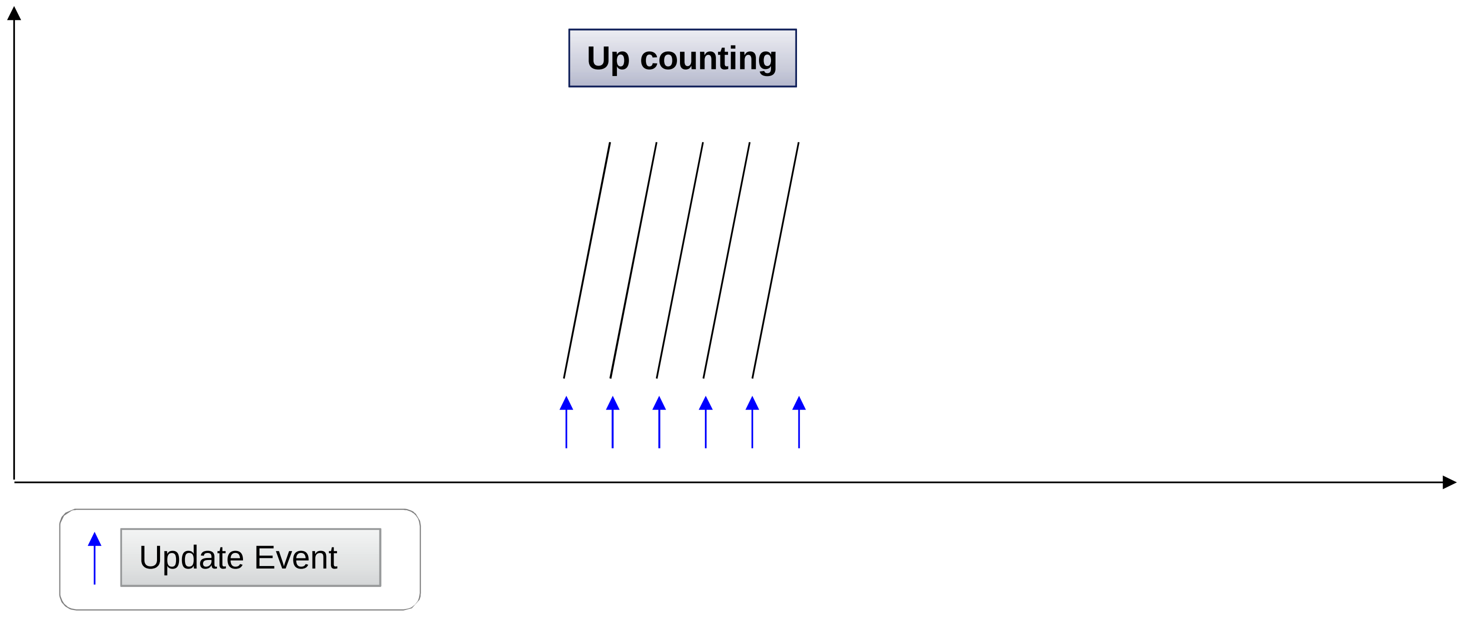
Counting Modes (1/2)

- Some timers have three counter modes:
  - Up counting mode
  - Down counting mode
  - Center-aligned mode



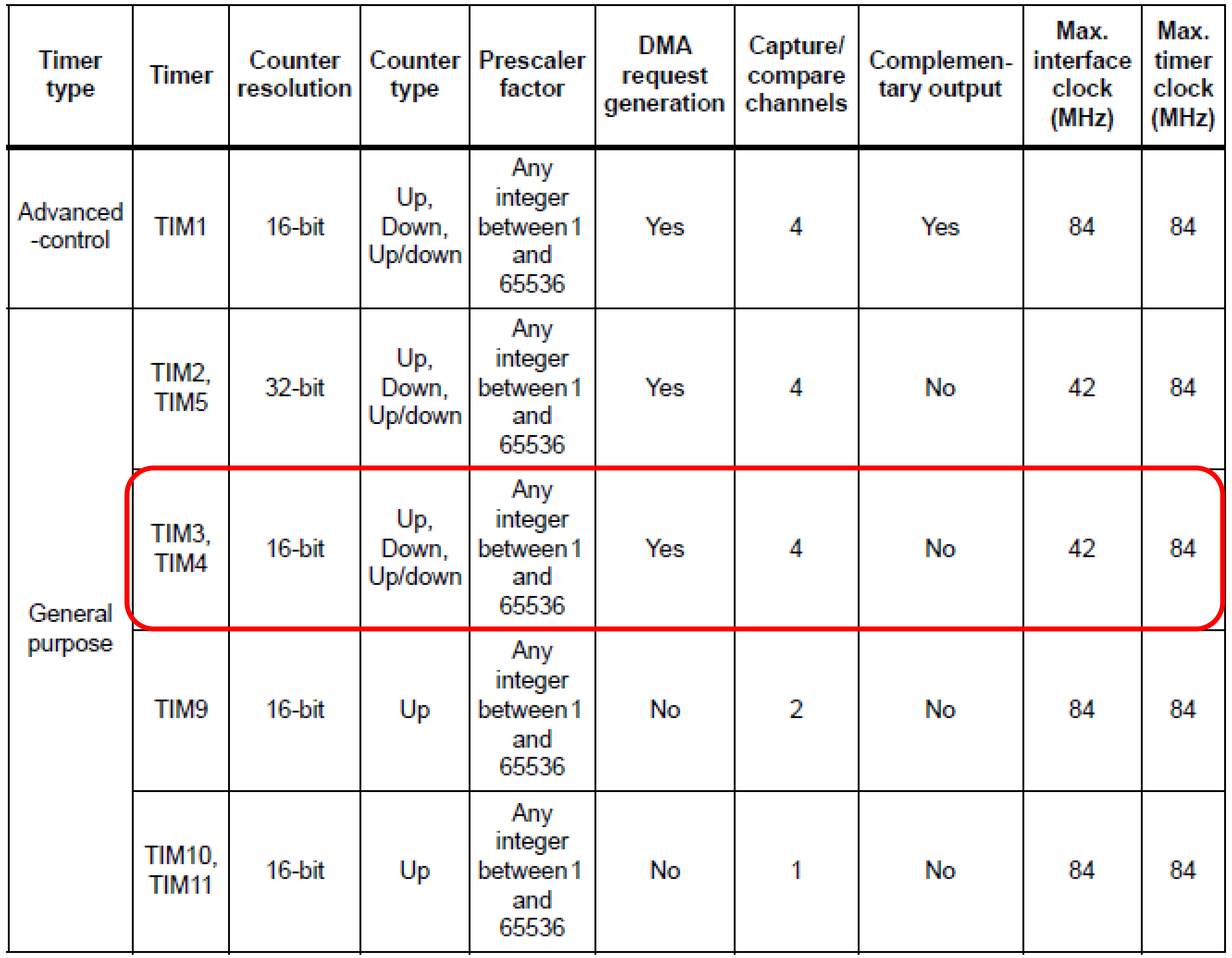
Counting Modes (2/2)

- One counting mode only for timers with less than 4 channels:
  - Up counting mode



ETH zürich

STM32F401 Timers Overview



Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/ compare channels	Complemen-tary output	Max. interface clock (MHz)	Max. timer clock (MHz)
Advanced -control	TIM1	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	Yes	84	84
General purpose	TIM2, TIM5	32-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
	TIM3, TIM4	16-bit	Up, Down, Up/down	Any integer between 1 and 65536	Yes	4	No	42	84
	TIM9	16-bit	Up	Any integer between 1 and 65536	No	2	No	84	84
	TIM10, TIM11	16-bit	Up	Any integer between 1 and 65536	No	1	No	84	84