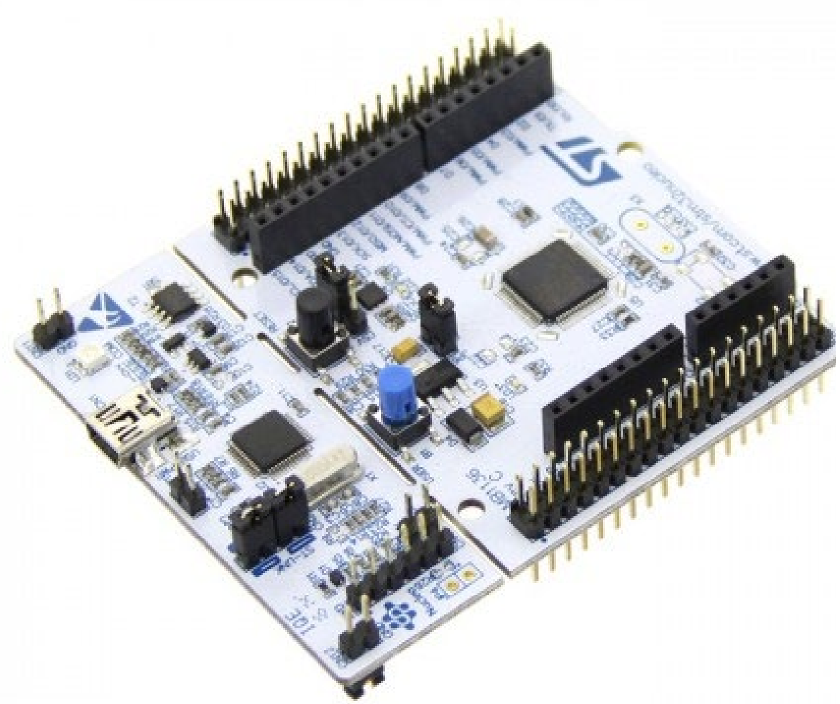
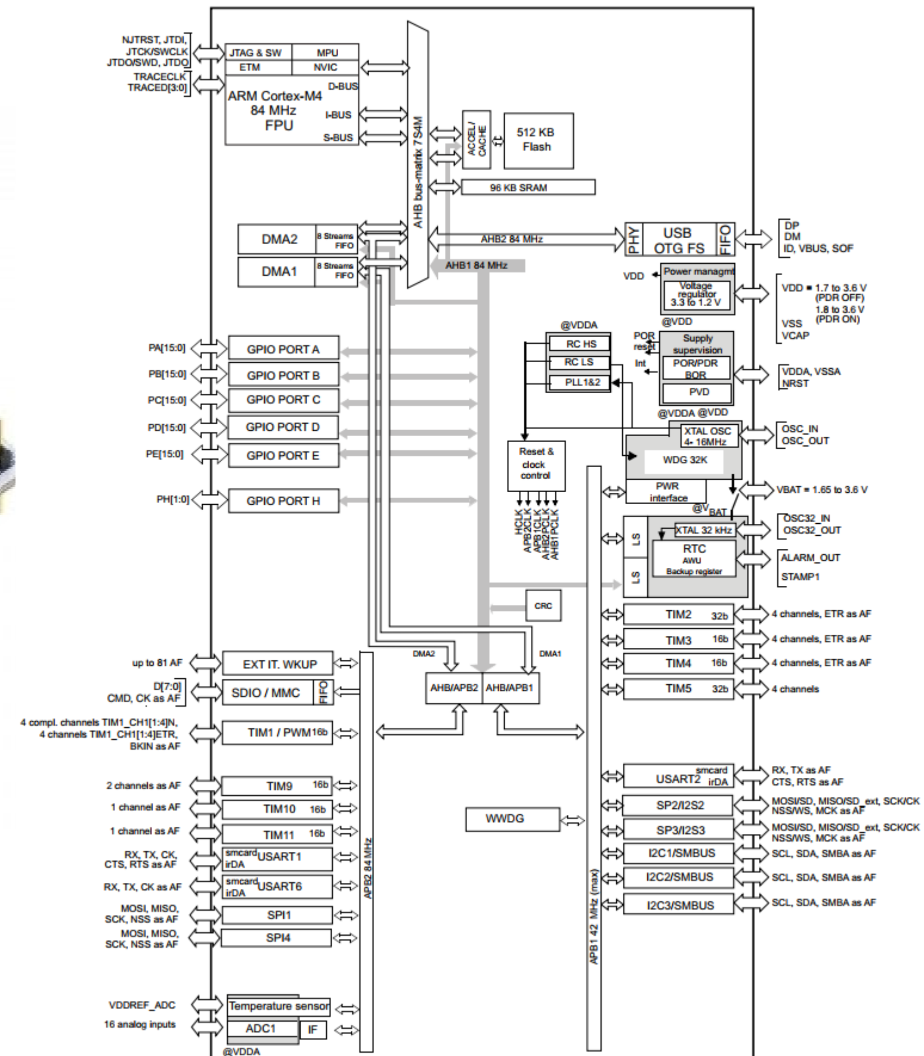
Microcontroller Engineering
TMIK13
Lecture 8

TIMERS, PWM

ANDREAS AXELSSON (ANDREAS.AXELSSON@JU.SE)

Nucleo-64 STM32F401RE



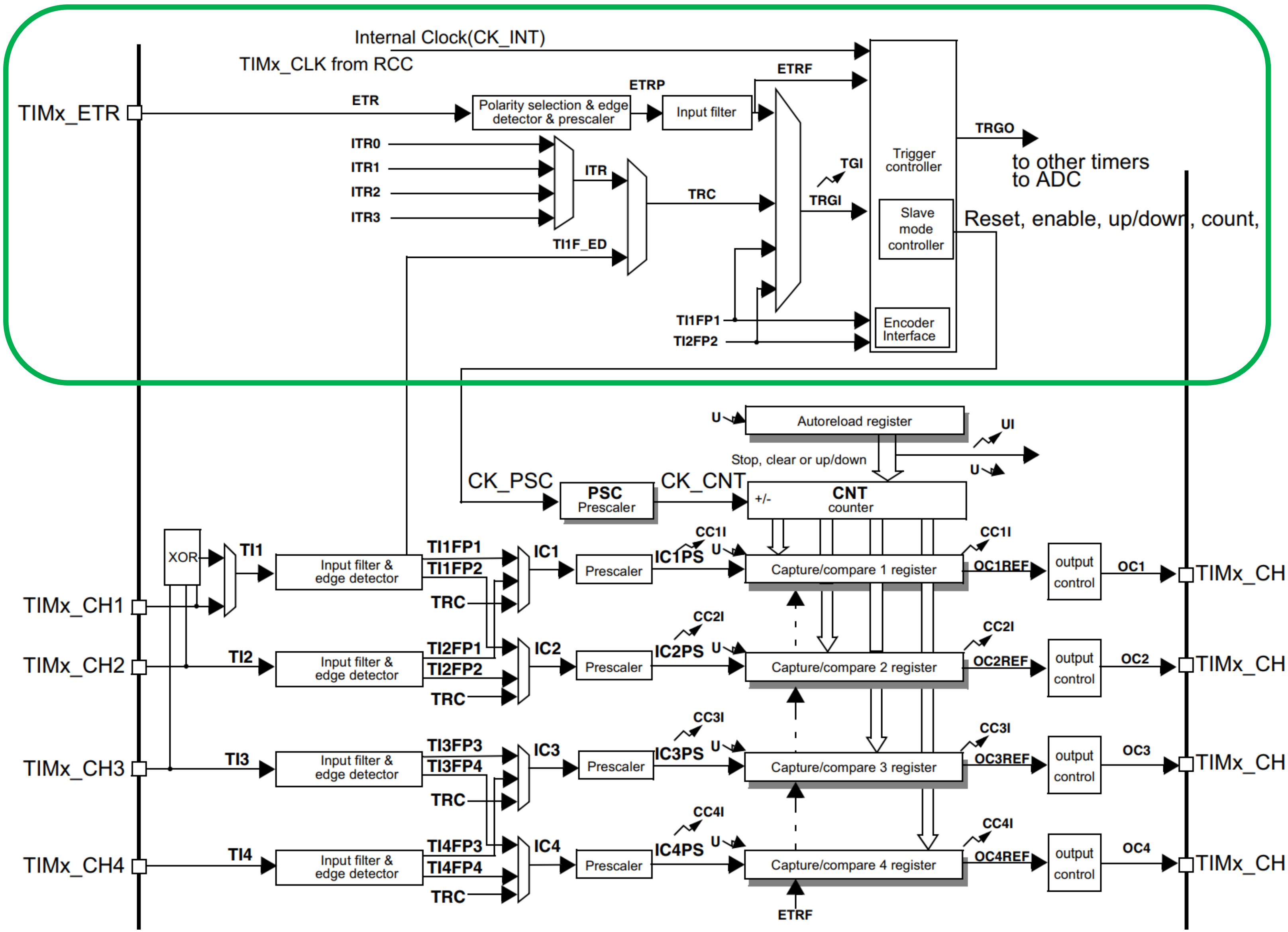


Timers – STM32F401RE

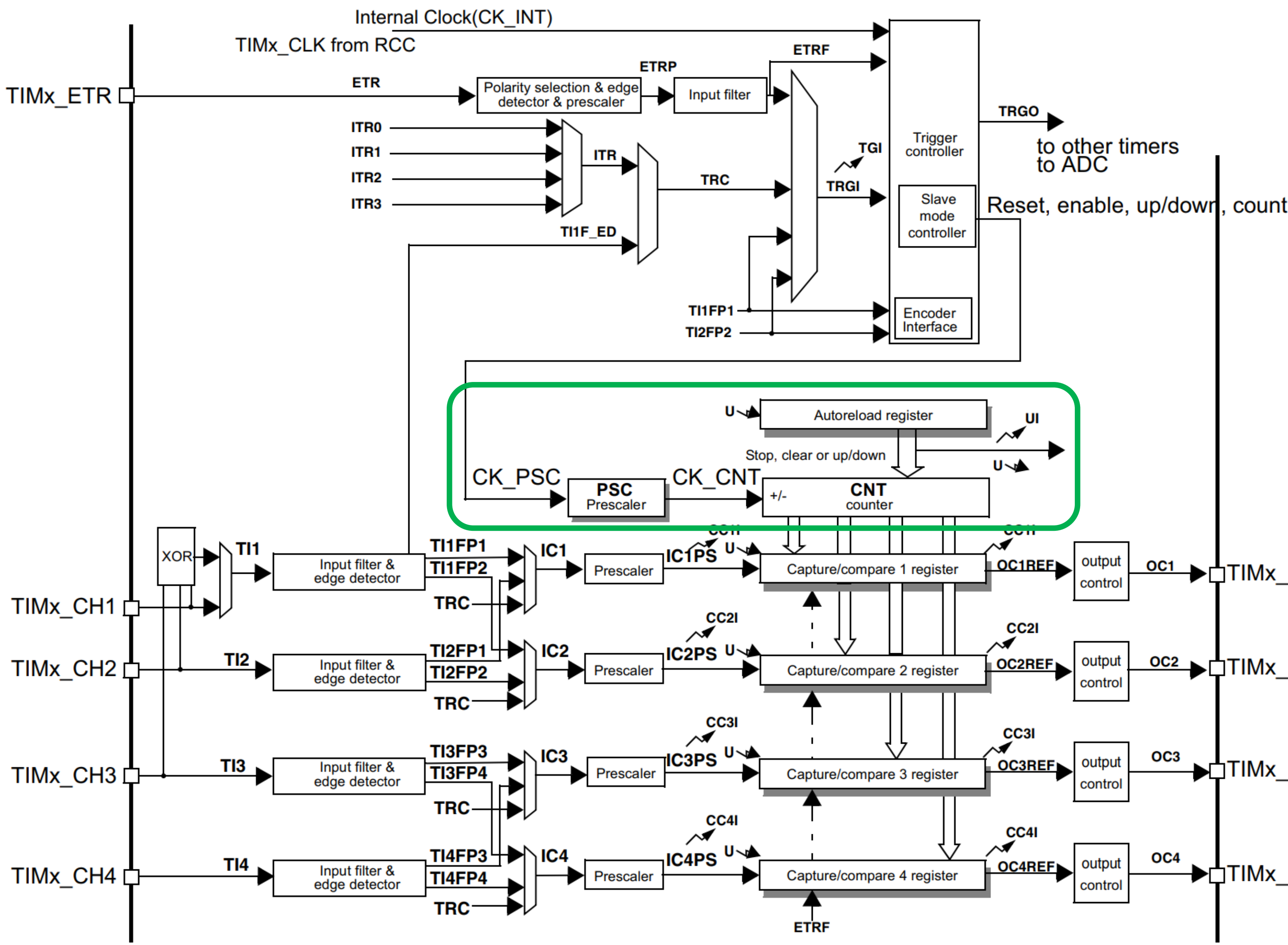
Table 4. Timer feature comparison

Timer type	Timer	Counter resolution	Counter type	Prescaler factor	DMA request generation	Capture/ compare channels	Complementary 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

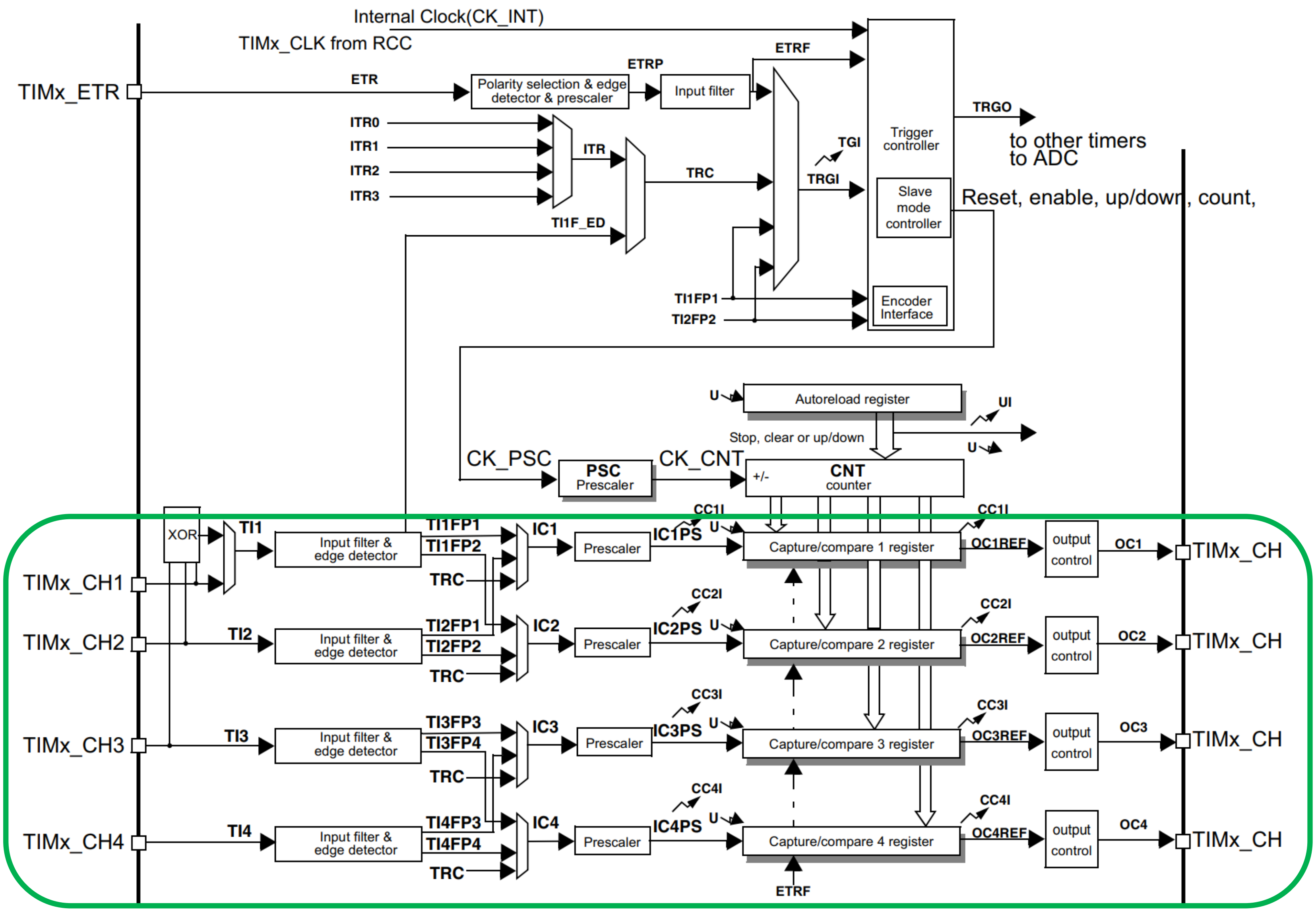
Timer – Block Schematics



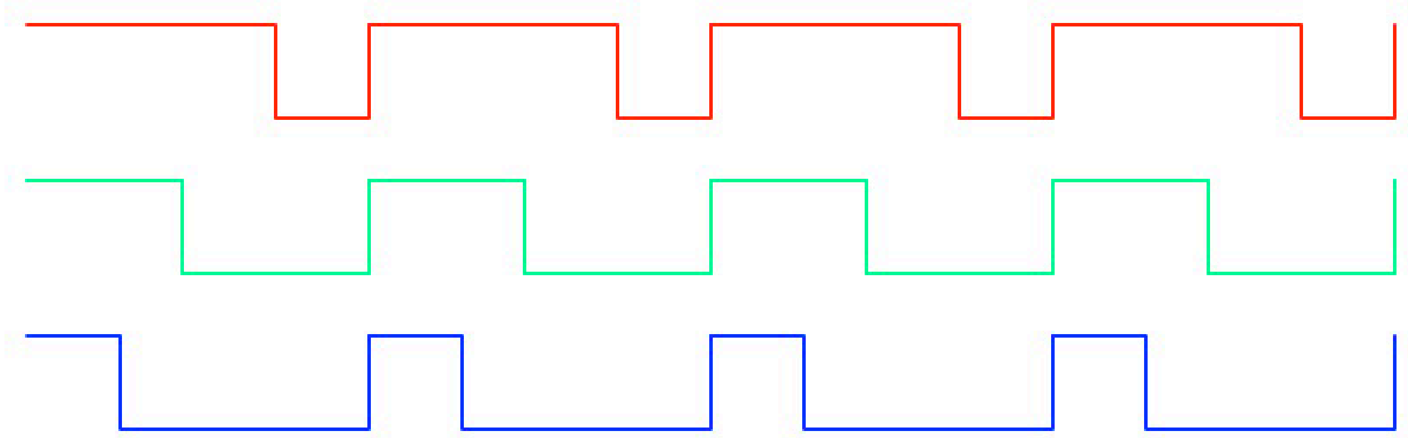
Timer – Block Schematics



Timer – Block Schematics



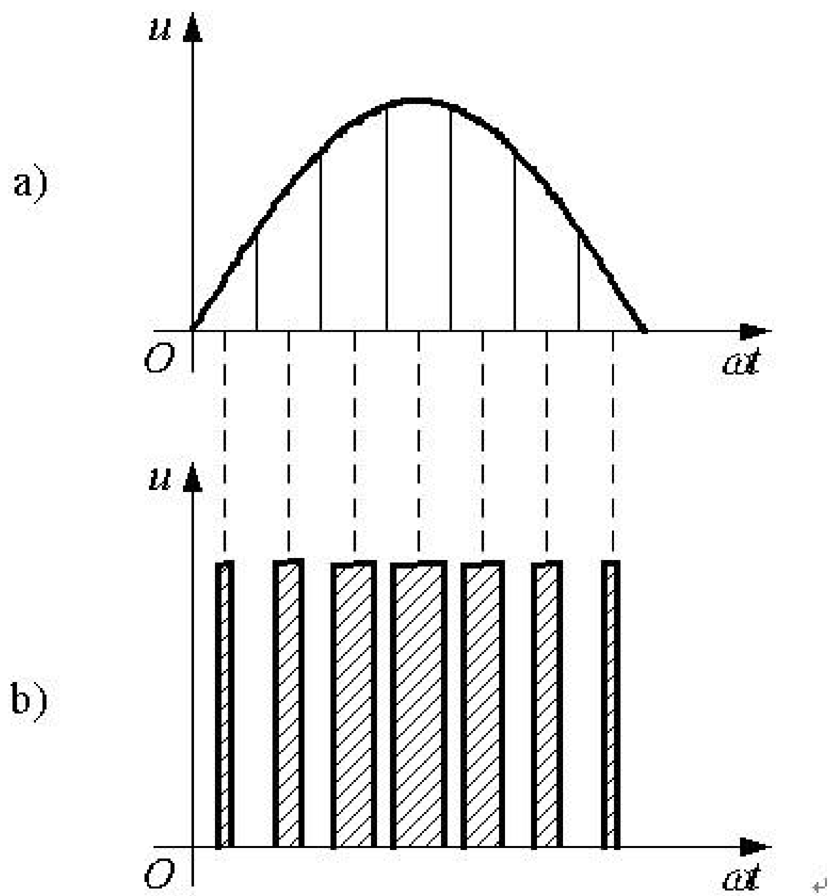
Timer – Pulse Width Modulation



Some usages of PWM:

1) Motor control. Speed is proportional to duty cycle
2) Intensity modulating LEDs. The eye only sees the average intensity
3) Audio generation. Filter to create an analog signal from PWM

Timer – Pulse Width Modulation



https://www.waveshare.com/wiki/STM32CubeMX_Tutorial_Series:_PWM

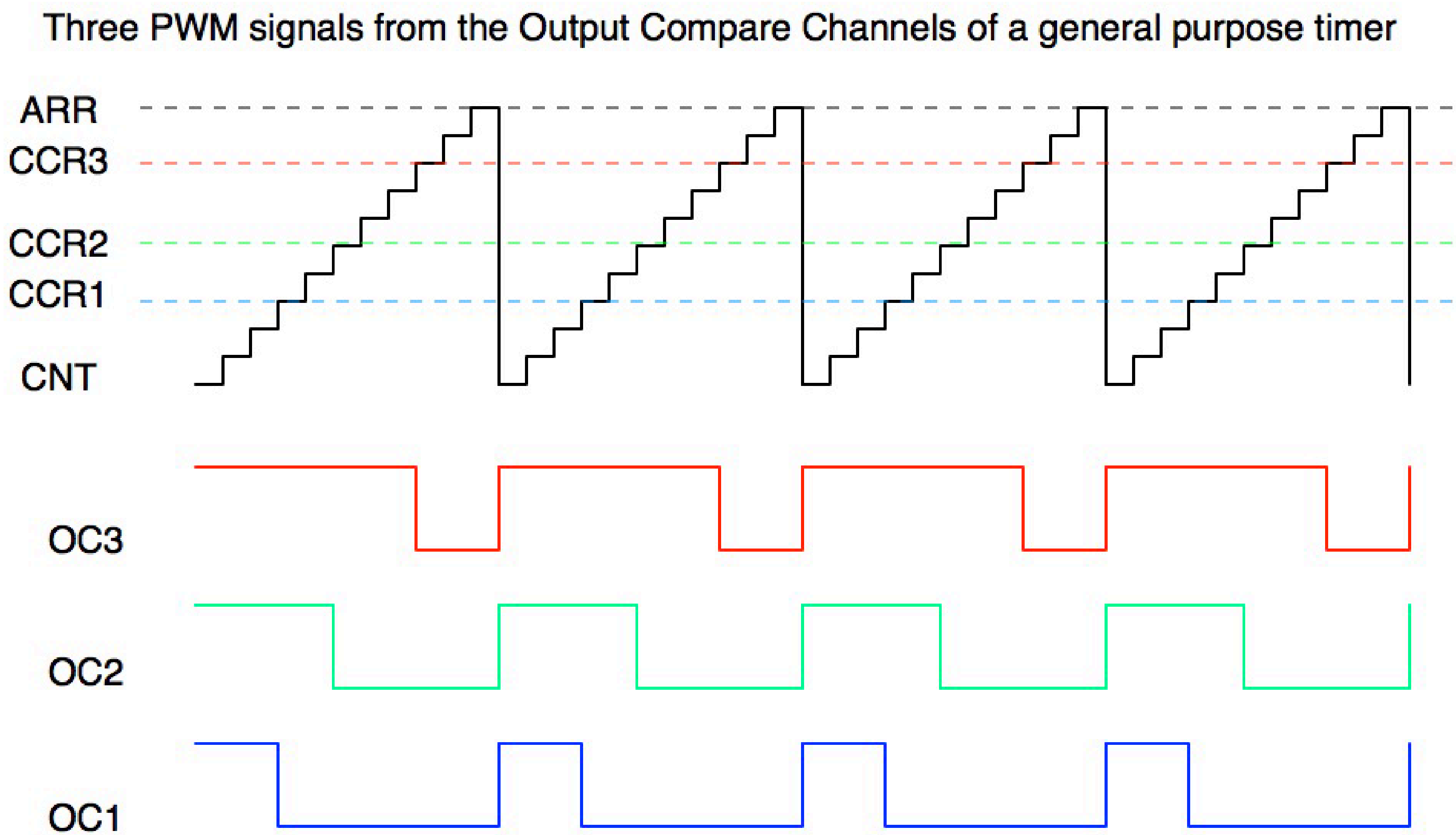
Timer – Pulse Width Modulation

Two ways to generate PWM:

1) Via software:
   a) Put the output to high
   b) Wait for required time for signal being high
   c) Put the output to low
   d) Wait for required time for signal being low
   e) Repeat from beginning

2) Hardware timer:
   Use the output compare functionality to generate the output.
   a) Program the timer period time (reload register)
   b) Set the width of the signal being high
   c) Activate the timer
   d) Sit back and enjoy!

Timer – Pulse Width Modulation



Each timer can generate up to 4 PWM output waveforms

Timer – Pulse Width Modulation

Reference manual:

12.3.10 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 corresponding preload register must be enabled by setting the OCxPE bit in the TIMx_CCMRx register, and eventually the auto-reload preload register (in upcounting or center-aligned modes) by setting the ARPE bit in the TIMx_CR1 register.

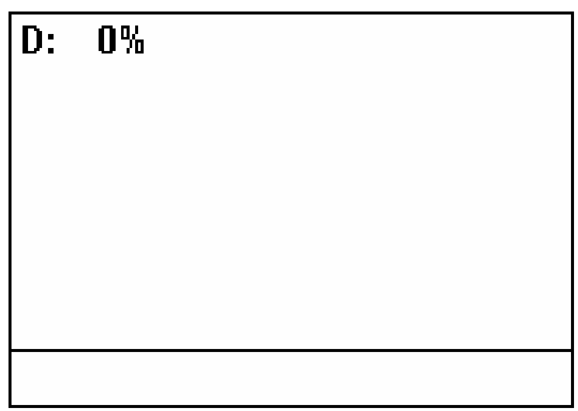
As the preload registers are transferred to the shadow registers only when an update event occurs, before starting the counter, the user must initialize all the registers by setting the UG bit in the TIMx_EGR register.

OCx polarity is software programmable using the CCxP bit in the TIMx_CCER register. It can be programmed as active high or active low. OCx output is enabled by a combination of the CCxE, CCxNE, MOE, OSSI and OSSR bits (TIMx_CCER and TIMx_BDTR registers). Refer to the TIMx_CCER register description for more details.

Timer – Pulse Width Modulation

Duty Cycle

$$Duty\ Cycle = \frac{Pulse\ Width}{Period\ Time} = \frac{Duration\ of\ pulse\ high}{Total\ duration}$$



Timer – Pulse Width Modulation

How to choose frequency (period time)?

Motor

- Motor time constant, $\tau \approx 10$ms
- Choose period time $\leq 0.1*\tau$, thus 1ms thus frequency $\geq$ 1kHz
    - Preferably higher frequency as it can be heard if in audible range.

LED (Light Emitting Diode)

- The eye can see flickering if frequency is less than approx. 25 Hz
- Switching a LED on/off doesn't create a sound.

$$PWM\ Frequency = \frac{TIM\_CLK}{(PSC+1)*(ARR+1)}$$

Timer – Pulse Width Modulation

How to choose duty cycle? What is reasonable step size

Motor

- Steps can be quite coarse
- A change of 1% or even 10% in some cases can be reasonable
  Thus an 8-bit resolution is often enough
- Motors often can't go from 0-100% in one step.
  Some sort of "soft" starting and stopping must be used.

LED (Light Emitting Diode)

- The eye is very sensitive to small changes in intensity
- At least 1000 levels should be used if high fidelity in change is required
- This corresponds to 10-bit resolution
- A LED can go from 0-100% in one step without breaking

Timers – HAL Driver

Timers – STM32CubeIDE Demo

Lets go!!!

Microcontroller Engineering

Questions?

Contact information

Andreas Axelsson

Email: andreas.axelsson@ju.se

Mobile: 0709-467760