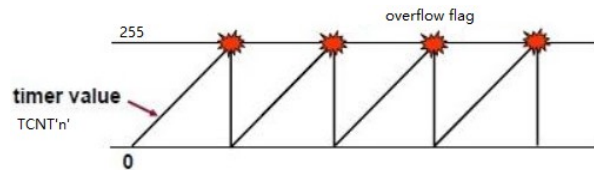


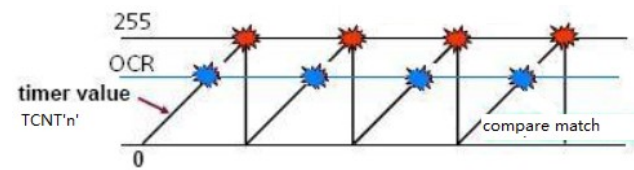
## AVR timer

An AVR timer in simplest terms is a register, usually 8 bits wide. Timer register increases/decreases automatically at a predefined rate (supplied by user). AVR timer runs asynchronous to (independent of) the main program, interacting via control and count registers, and timer interrupts. This means, timer can be used to measure time accurately and with help of Interrupt Service Routine – ISR can make events execution precise. Timer can use two types of interrupts – overflow and compare match.

*Overflow interrupt is triggered whenever the timer register overflows (reaches its max value, like 255)*



*Compare match interrupt is triggered whenever timer value becomes equal to certain value (this value is stored in OCR)*



### **My signage for registers:**

'n' – timer No. (0, 1, 2)

'b' – bit No.

'x' – output pin (A, B)

'p' – MCU port (B, C, D)

7:6:5:4:3:2:1:0: - bit position in register

### **Register and their corresponding bits signage:**

TCCR'n' – Timer/Counter Control Register (there is two registers (A, B) with different parameters)

WGM'n''b'	- Waveform Generation Mode	1:0:	bit within TCCR'n'A
		2:	bit within TCCR'n'B
COM'n''x''b'	- Compare Output Mode	7:6:5:4:	bit within TCCR'n'A
CS'n''b'	- Clock Select	2:1:0:	bit within TCCR'n'B
FOC'n''x'	- Force Output Compare	7:6:	bit within TCCR'n'B

TCNT'n' – Timer/Counter Register (8bit value)

OCR'n''x' – Output Compare Register (8bit value which gets compared to TCNT'n')

DDR'p' – Data Direction Register (1 – configured as output, 0 – as input, each bit of register corresponds to some physical pin)

PORT'p' – Data Register (determines the state of outputs /if pin is set as output/, 1 – high, 0 – low )

PIN'p' – Input Pins Register (it will read all of the digital input pins at the same time)

ICR'n' – Input Capture Register

OC'n''x' – Signage for output pin