

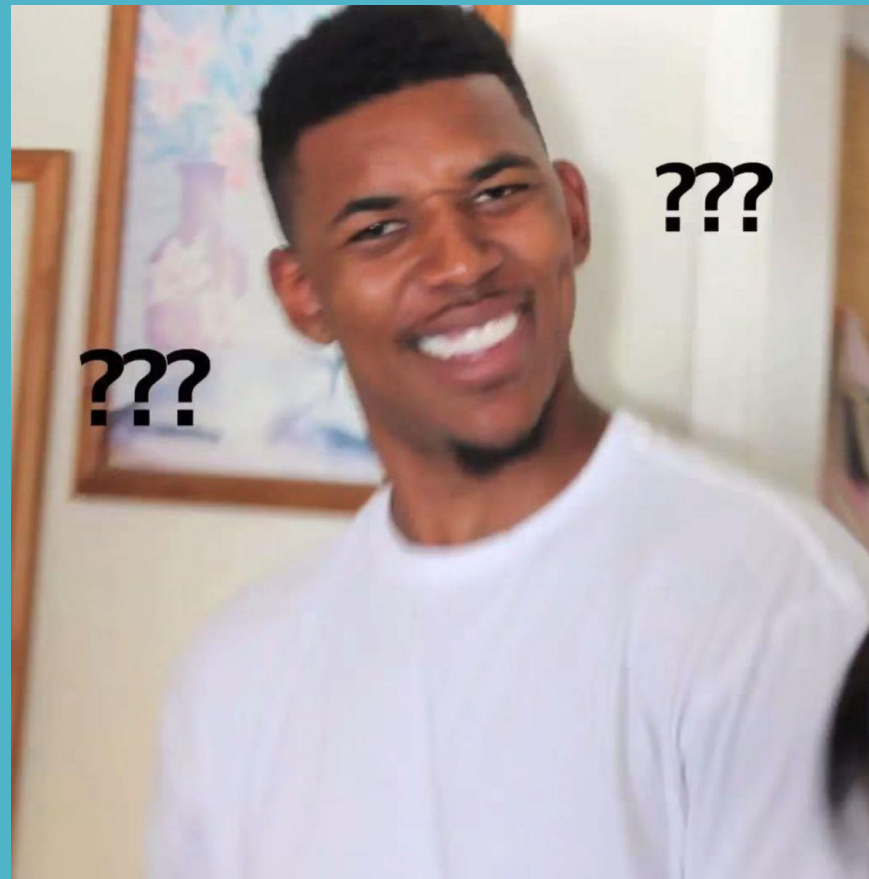
Microcontroladores

TIMER

Timer

Contagem de Tempo

Para que serve a contagem de tempo?



Timer no ATmega328P

- O microcontrolador ATmega328P possui três unidades de Timer/Contador;
 - TIMER 0: 8 bits;
 - TIMER 1: 16 bits;
 - TIMER 2: 8 bits;



Modos de Operação

Overflow

Como Funciona?

- A contagem é feita de forma crescente;
- No caso do TIMER 0, quando a contagem passa de 255->0, ocorre o estouro (OVERFLOW). Dessa forma, o BIT TOV0, presente no registrador TIFR, é setado. Caso a interrupção esteja habilitada, ela é gerada.;

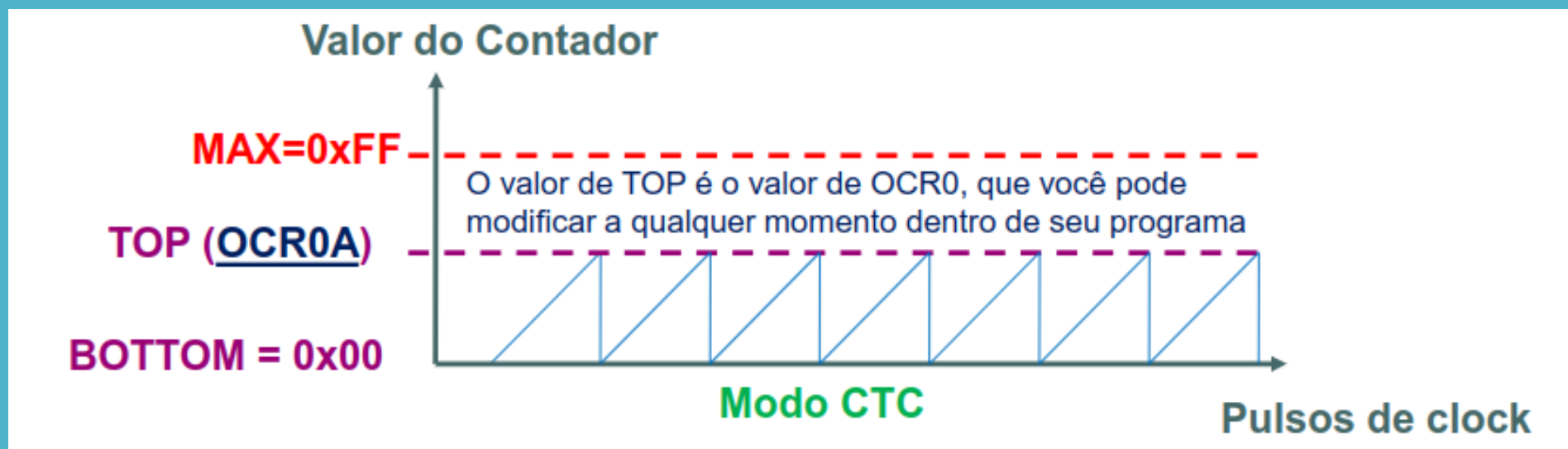
$$t_{estouro} = \frac{(TOP+1) \times prescaler}{f_{osc}}$$

TOP é o valor máximo de contagem, f_{osc} é a frequência do *clock* e o *prescaler* é o divisor dessa frequência.

Comparação (CTC)

Como Funciona?

- No modo CTC (Clear timer on compare - Limpeza do contador na igualdade de comparação), o registrador OCRnA é utilizado como referência (TOP) na contagem;
- Dessa forma, quando o registrador de contagem:
 $TCNTn = OCRnA$ ou $OCRnB$, a contagem é reiniciada.



Registradores – TIMER 0

Timer/Counter Control Register (TCCR0A E TCCR0B)

- Registrador de Controle;
- Permite a configuração de clock + prescaler;

	7	6	5	4	3	2	1	0
<u>TCCR0B</u>	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00
	E	E	L	L	L/E	L/E	L/E	L/E
	0	0	0	0	0	0	0	0

- Os bits CS00, CS01 e CS02 fazem a seleção da fonte de clock;
- O bit WGM02, junto com os bits WGM00 e WGM01 (PRESENTES NO TCCR0A) definem o modo de operação do TIMER;

CS02	CS01	CS00	Descrição
0	0	0	Sem fonte de <i>clock</i> (TC0 parado).
0	0	1	<i>clock/1</i> (<i>prescaler=1</i>) - sem <i>prescaler</i> .
0	1	0	<i>clock/8</i> (<i>prescaler</i> = 8).
0	1	1	<i>clock/64</i> (<i>prescaler</i> = 64).
1	0	0	<i>clock/256</i> (<i>prescaler</i> = 256).
1	0	1	<i>clock/1024</i> (<i>prescaler</i> = 1024).
1	1	0	clock externo no pino T0. Contagem na borda de descida.
1	1	1	clock externo no pino T0. Contagem na borda de subida.

Bit	7	6	5	4	3	2	1	0
	COM0A1	COM0A0	COM0B1	COM0B0			WGM01	WGM00
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Modo	WGM02	WGM01	WGM00	Modo de Operação TC	TOP	Atualização de OCR0A no valor:	Sinalização do bit TOV0 no valor:
0	0	0	0	Normal	0xFF	Imediata	0xFF
1	0	0	1	PWM com fase corrigida	0xFF	0xFF	0x00
2	0	1	0	CTC	OCR0A	Imediata	0xFF
3	0	1	1	PWM rápido	0xFF	0x00	0xFF
4	1	0	0	Reservado	-	-	-
5	1	0	1	PWM com fase corrigida	OCR0A	OCR0A	0x00
6	1	1	0	Reservado	-	-	-
7	1	1	1	PWM rápido	OCR0A	0x00	OCR0A

TC0 Interrupt Mask Register (TIMSK0)

- Registrador responsável por habilitar as interrupções;

	7	6	5	4	3	2	1	0
TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
	L	L	L	L	L	L/E	L/E	L/E
	0	0	0	0	0	0	0	0

- A escrita '1' no bit OCIE0B, habilita a interrupção do TC0 na igualdade de comparação com registrador OCR0B;
- A escrita '1' no bit OCIE0A, habilita a interrupção do TC0 na igualdade de comparação com registrador OCR0A;
- A escrita '1' no bit TOIE0, habilita a interrupção do TC0 por OVERFLOW(255->0);

Configuração dos registradores para trabalhar com comparação

- Configurar o modo de operação do TIMER e o divisor do clock: TCCR0A e TCCR0B;
- Configurar o valor máximo de contagem: OCR0A;
- Habilitar a interrupção do comparador desejado: TIMSK0;
- Habilitar a interrupção global do microcontrolador: sei();

Rotina de Interrupção (ISR)

Para o OVERFLOW:

ISR(TIMERO_OVF_vect)

{

 //Faz algo

}

Para o CTC(A):

ISR(TIMERO_COMPA_vect)

{

 //Faz algo

}

Para o CTC(B):

ISR(TIMERO_COMPB_vect)

{

 //Faz algo

}

Obrigado!