

# Temporizadores e Contadores

## DCA0119 - Sistemas Digitais

Mário Sérgio Cavalcante<sup>1</sup>

<sup>1</sup>Departamento de Engenharia de Computação e Automação  
mariocavalcante@dca.ufrn.br

Natal - Rio Grande do Norte

# Sumário

- 1 Introdução
- 2 Temporizador/Contador 0: Modo Normal
  - TC0 - Normal
- 3 TC0 - Temporizador/Contador 0: Modo CTC
  - Exemplo 1
  - Exemplo 2
  - Exemplo 3
- 4 TC0 - Modo PWM Rápido
  - Introdução
  - Registradores
  - Exemplo 1

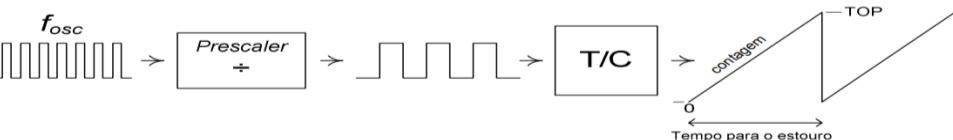
## Introdução

## Temporizadores e Contadores (TCs)

- Utilizados para geração de sinais periódicos e eventos;
- Contam pulso de clock interno ou externo;
- Um importante exemplo de aplicação é o sinal PWM;
- O Atmega3289p possui 3 contadores independentes:
  - TC0 e TC2 de 8 bits;
  - TC1 de 16 bits.

# Temporizadores e Contadores

- Contador de 8 bits conta de 0 a 255;
- Contador de 16 bits conta de 0 a 65.535;
- Um "**estouro**" ocorre quando o contador atinge seu valor máximo permitido para contagem (TOP) e retorna para o valor inicial (normalmente 0).



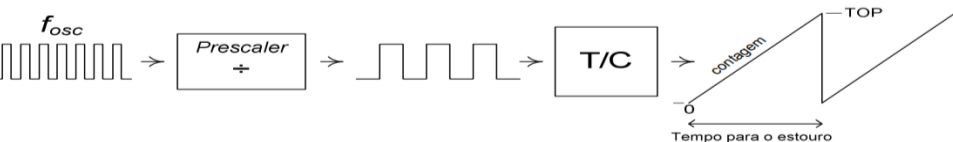
$$t_{\text{estouro}} = \frac{(TOP + 1) \cdot \text{prescaler}}{f_{osc}}$$

Top - Valor máximo;  
 $f_{osc}$  - Frequência do Clock (Oscilador);  
prescaler - Divisor da frequência

## Temporizadores e Contadores

- Exemplo:** Ao utilizar TC0 (8 bits) trabalhando com frequência do oscilador RC interno do Atmega328p (16 MHz) e sem divisão de frequência. Qual o tempo de estouro do contador?

$$t_{\text{estouro}} = \frac{(255 + 1) \cdot 1}{16 \cdot 10^6} = 16 \mu\text{s}$$



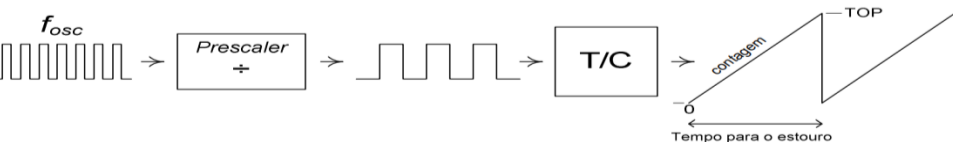
$$t_{\text{estouro}} = \frac{(TOP + 1) \cdot \text{prescaler}}{f_{\text{osc}}}$$

Top - Valor máximo;  
 $f_{\text{osc}}$  - Frequência do Clock (Oscilador);  
prescaler - Divisor da frequência

## Temporizadores e Contadores

- **Exemplo:** Ao utilizar TC1 (16 bits) trabalhando com frequência do cristal utilizado no arduino (16 MHz) e com divisão de frequência igual à 64. Qual o tempo de estouro do contador?

$$t_{\text{estouro}} = \frac{(65535 + 1) \cdot 64}{16 \cdot 10^6} = 262,1 \text{ ms}$$



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

Top - Valor máximo;  
 $f_{osc}$  - Frequência do Clock (Oscilador);  
prescaler - Divisor da frequência





## Temporizador/Contador 0: Modo Normal

## Temporizador/Contador - TC0

- Contador simples de 8 bits;
- Contador de eventos externos;
- Divisor do clock (prescaler) de até 10 bits;
- Gerador para 2 sinais PWM (pinos OC0A e OC0B);
- Gerador de onda quadrada em diversas frequências;
- 3 fontes independentes de interrupção (estouro/comparação);

## Temporizador/Contador - TC0

Endereço	Nome	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x35	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0
0x44	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00
0x45	TCCR0B	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00
0x46	TCNT0	Temporizador/Contador 0 (8 bits) - Registrador de contagem							
0x47	OCR0A	Registrador de comparação de saída A do Temporizador/Contador 0							
0x48	OCR0B	Registrador de comparação de saída B do Temporizador/Contador 0							
0x6E	TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0

## Registradores do TC0

### ■ TCNT0:

Timer/Counter 0 Register

- Registrador de 8 bits onde é realizada a contagem do TC0 pode ser lido ou escrito a qualquer tempo.

### ■ OCR0A e OCR0B:

Output Compare 0 Register A e Output Compare 0 Register B

- Registrador de comparação A ou B de 8 bits, possui o valor que é continuamente comparado com o valor do contador (TCNT0).
- A igualdade pode ser utilizada para gerar uma interrupção ou uma forma de onda no pino OC0A ou OC0B.

# Temporizadores/Contadores - TC0

Bit	7	6	5	4	3	2	1	0
<b>TCCR0A</b>	<b>COM0A1</b>	<b>COM0A0</b>	<b>COM0B1</b>	<b>COM0B0</b>	-	-	<b>WGM01</b>	<b>WGM00</b>
Lê/Escreve	L/E	L/E	L/E	L/E	L	L	L/E	L/E
Valor Inicial	0	0	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 Corrigido	OCR0A	Top (OCR0A)	0xFF
6	1	1	0	Reservado	-	-	-
7	1	1	1	PWM Rápido	OCR0A	0x00	0xFF

# Temporizadores/Contadores - TC0

Bit	7	6	5	4	3	2	1	0
TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00
Lê/Escreve	L/E	L/E	L/E	L/E	L	L	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

## ■ Modo CTC

COM0A1	COM0A0	Descrição
0	0	Operação normal do pino, OC0A desconectado.
0	1	Mudança do estado de OC0A na igualdade de comparação.
1	0	OC0A é limpo na igualdade de comparação.
1	1	OC0A é ativo na igualdade de comparação.

# Temporizadores/Contadores - TC0

Bit	7	6	5	4	3	2	1	0
<b>TCCR0A</b>	<b>COM0A1</b>	<b>COM0A0</b>	<b>COM0B1</b>	<b>COM0B0</b>	-	-	<b>WGM01</b>	<b>WGM00</b>
Lê/Escreve	L/E	L/E	L/E	L/E	L	L	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

## ■ Modo PWM

COM0A1	COM0A0	Descrição
0	0	Operação normal do pino, OC0A desconectado.
0	1	WGM02 = 0: operação normal do pino, OC0A desconectado. WGM02 = 1: troca de estado do OC0A na igualdade de comparação.
1	0	OC0A é limpo na igualdade de comparação, OC0A é ativo no valor do TC mínimo (modo não invertido).
1	1	OC0A é ativo na igualdade de comparação, OC0A é limpo no valor do TC mínimo (modo invertido).

## Temporizadores/Contadores - TC0

Bit	7	6	5	4	3	2	1	0
<b>TCCR0B</b>	<b>FOC0A</b>	<b>FOC0B</b>	<b>-</b>	<b>-</b>	<b>WGM02</b>	<b>CS02</b>	<b>CS01</b>	<b>CS00</b>
Lê/Escreve	E	E	L	L	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

CS02	CS01	CS00	Descrição
0	0	0	Sem fonte de clock (TC0 parado)
0	0	1	Clock/1 - Sem prescaler
0	1	0	Clock/8 (Prescaler de 8)
0	1	1	Clock/64 (Prescaler de 64)
1	0	0	Clock/256 (Prescaler de 256)
1	0	1	Clock/1024 (Prescaler de 1024)
1	1	0	Clock externo no Pino T0. Contagem de borda de descida
1	1	1	Clock externo no Pino T0. Contagem na borda de subida



## Temporizadores/Contadores - TC0

Bit	7	6	5	4	3	2	1	0
TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
Lê/Escreve	L	L	L	L	L	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

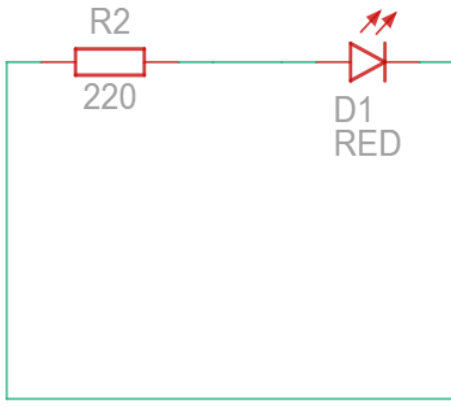
- **Bit 2 - OCIE0B** - Timer/Counter 0 Output Compare Match B Interrupt Enable
  - A escrita 1 neste bit ativa a interrupção do TC0 na igualdade de comparação com o registrador OCR0B
- **Bit 1 - OCIE0A** - Timer/Counter 0 Output Compare Match A Interrupt Enable
  - A escrita 1 neste bit ativa a interrupção do TC0 na igualdade de comparação com o registrador OCR0A
- **Bit 0 - TOIE0** - Timer/Counter 0 Overflow Interrupt Enable
  - A escrita 1 neste bit ativa a interrupção por estouro do TC0

## TC0 - Normal

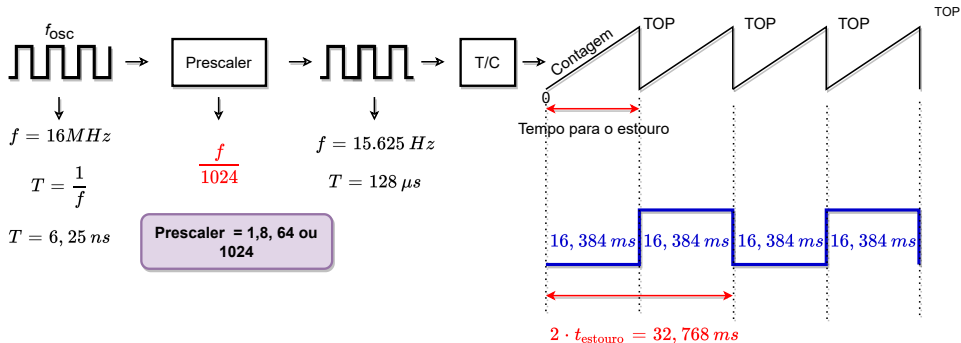
- É o modo mais simples de operação;
- Contagem crescente no registrador **TCNT0** com valor da contagem podendo ser alterado por programação;
- Como é um contador de 8 bits sua contagem se dá de 0 a 255 retornando a 0 e fechando um ciclo de forma contínua;
- Ocorre **estouro** quando retorna para 0 e o bit sinalizador de estouro (TOV0) é colocado em 1;
- Quando habilitada, a interrupção é gerada no estouro.

## 2.1: Exemplo Simples

O estado do LED (PB5) deve mudar de estado a cada estouro do TC0;



## TC0 - Modo Normal - Exemplo (Temporizador)



### 2.2: Nota:

Onde lê-se:

$T = 128\text{ }\mu\text{s}$  deveria ser:

$$T = \frac{1}{f} = \frac{1}{15.625} = 64\text{ }\mu\text{s}$$

$$\begin{aligned} t_{estouro} &= 256 \cdot T \\ t_{estouro} &= 256 \cdot 64 \cdot 10^{-6} \\ t_{estouro} &= 16,384\text{ ms} \end{aligned}$$

## TC0 - Modo Normal - Exemplo (Configurações)

### ■ Configurações:

- Para o modo normal, bits **WGM02:0** = 000.
- Vamos setar os prescaler: 1024 → Precisamos configurar o registrador **TCCR0B**, mais precisamente Bits **CS02:0** = 101;
- Os bits **COM0A1:0** e **COM0B1:0** = 00 no registrador **TCCR0A** para desconectarem os pinos **OC0A** e **OC0B** do TC0.
- Bit **TOIE0** = 1 no registrador **TIMSK0** para habilitar a interrupção no estouro.

## Code Exemplo

```
1  #include <avr/io.h>
2  #include <avr/interrupt.h>
3  #define F_CPU 16000000UL
4  #define LED PB5           //LED em PB5
5  int main(){
6      DDRB |= (1<<LED);    //Pino do LED como saída
7      PORTB &= ~(1<<LED);  //Apaga LED
8      TCCR0B = (1 << CS02) | (1 << CS00); //TC0 em prescaler de 1024
9      TCCR0A = 0; //Configurando o modo normal, OCR0A e OCR0B desconectado
10     TIMSK0 = 1 << TOIE0; //Habilita interrupção do TC0
11     sei(); //Habilita interrupção geral
12     while(1); //Laço Infinito
13 }
14 //Tratamento da interrupção do estouro de TC0
15 ISR(TIMER0_OVF_vect){ //Para F_CPU em 16 MHz, estouro em 16,384 ms
16     PORTB ^= 1<<LED; //Troca estado do LED
17 }
```

## TC0 - Temporizador/Contador 0: Modo CTC

## Registadores do TC0 - Modo CTC

### ■ TCNT0:

Timer/Counter 0 Register

- Registrador de 8 bits onde é realizada a contagem do TC0 pode ser lido ou escrito a qualquer tempo.

### ■ OCR0A e OCR0B:

Output Compare 0 Register A e Output Compare 0 Register B

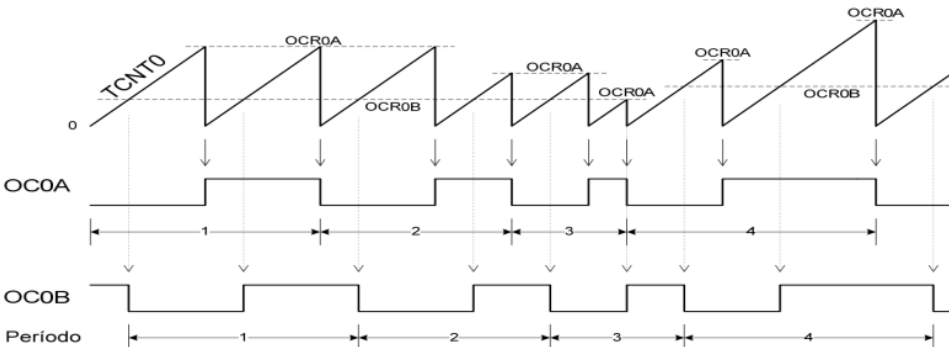
- Registrador de comparação A ou B de 8 bits, possui o valor que é continuamente comparado com o valor do contador (TCNT0).
- A igualdade pode ser utilizada para gerar uma interrupção ou uma forma de onda no pino OC0A ou OC0B.



## TC00 - Modo CTC

- CTC (Clear Time on Compare): Limpeza do contador na igualdade de comparação;
- O contador **TCNT0** é zerado quando é igual ao valor de **OCR0A** (Top);
- O registrador **OCR0A** é usado para manipular a resolução do TC0;
- Permite configura os pinos **OC0A** e **OC0B** para gerar ondas quadradas;
- Uma interrupção pode ser gerada cada vez que o contador atinge o valor de comparação (**OCR0A** ou **OCR0B**).

## TC0 - Modo CTC - Cálculo de frequência

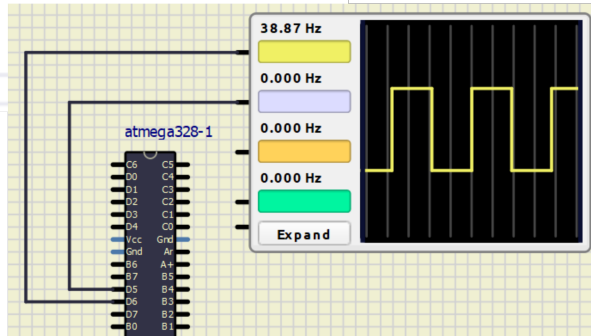


$$f_{OCnx} = \frac{f_{osc}}{2 \cdot (OCR0A + 1) \cdot \text{prescaler}}$$
$$OCR0A = \frac{f_{osc}}{2 \cdot \text{prescaler} \cdot f_{OCnx}} - 1$$

$f_{OCnx}$  - Frequência Gerada;  
 $f_{osc}$  - Frequência do Clock (Oscilador);  
prescaler - Divisor da frequência  
OCR0A - Valor do registrador

## TC0 - CTC - Exemplo 1: Pino OC0A

```
1  #include <avr/io.h>
2  int main(){
3      DDRD = 0b01100000;
4      PORTD = 0b10011111;
5      TCCR0A = (1 << COM0A0) | (1 << WGM01);
6      TCCR0B = (1 << CS02) | (1 << CS00);
7      OCR0A = 200;
8
9      while(1);
10 }
```





## TC0 - CTC - Exemplo 2: Pinos OC0A e OC0B

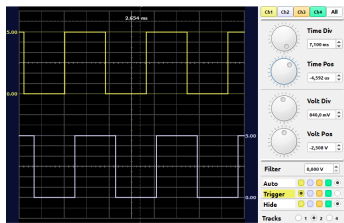
```

1  #include <avr/io.h>
2  int main(){
3      DDRD = 0b01100000; // OC0A e OC0B = (PD5 e PD6) como saída
4      PORTD = 0b10011111; // OC0A e OC0B = 0, Hab. pull-ups dos demais pinos
5      TCCR0A = (1 << COM0A0) | (1 << COM0B0) | (1 << WGM01); // OC0A troca de estado
6      TCCR0B = (1 << CS02) | (1 << CS00); //TC0 com prescaler = 1024
7      OCR0A = 200; //Máximo valor de contagem do registrador TCNT0
8      OCR0B = 50; // Deslocamento de OC0B em relação a OC0A
9      while(1); // Laço Infinito
10 }
```

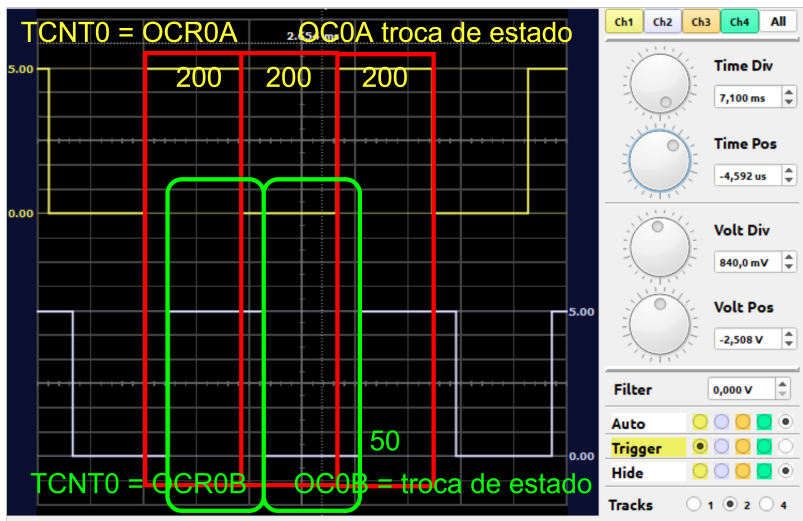
TCCR0A

TCCR0B

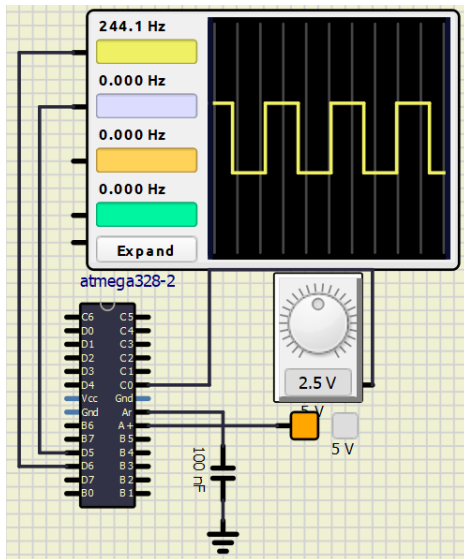
- TCNT0 estoura e retorna para zero quando igual a **OC0A**. **OC0A** troca de estado.
- **OC0B** troca de estado quando igual a **OCR0B**.



## TC0 - Exemplo 2: Pinos OC0A e OC0B



## Exemplo 3 - CTC: Variando Tempo de Estouro



## Exemplo 3 - CTC: Variando Tempo de Estouro

```
279 int main(void)
280 {
281     unsigned char ValAdc; //Armazena o valor do ADC - (apenas 8 bits)
282     DDRD = 0b01100000; //OC0B e OC0A (PD5 e PD6) como saída
283     PORTD = 0b10011111; //OC0B e OC0A = 0. Hab. pull-ups dos demais pinos
284     TCCR0A = (1<<COM0A0) | (1<<WGM01); //OC0A troca estado na igualdade de comparação
285     TCCR0B = (1<<CS02) | (1<<CS02); //TC0 com prescaler = 1024
286     OCR0A = 0; //Valor inicial do registrador OC0A
287     adcBegin(AVCC,0x01); // Inicia ADC
288     adcChannel(ADC0); // Seleciona canal do ADC
289     UART_Init(); // Inicia a transmissão (USART)
290     while(1){
291         ValAdc = (adcRead() >> 2); //Realiza conversão e utiliza 8 bits mais significativos
292         OCR0A = ValAdc; //Registrador OCR0A recebe o valor
293         uartDec1B(ValAdc); //Ler e envia valor do A/D
294         uartString("\r\n"); //e
295     }
```

### 3.1: Exemplo:

- O Registrador **OCR0A** (Top da contagem) recebe os oito bits mais significativos do conversor A/D. O período de estouro e, conseqüentemente, a frequência do sinal gerado OC0A mudam conforme se movimenta o potenciômetro e altera a tensão aplicada ao A/D.

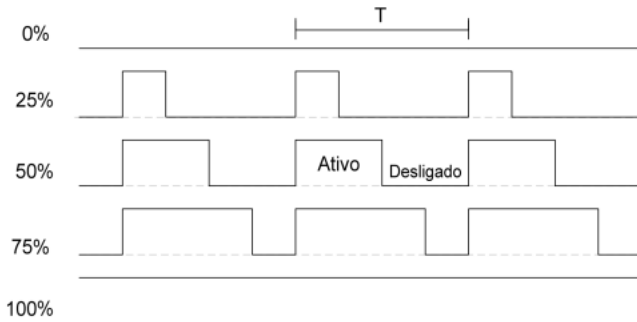


## TC0 - Modo PWM Rápido

## PWM - Modulação por largura de pulso

- A geração de sinais PWM é outra função importante dos Temporizadores/Contadores.
- O uso desses sinais é baseado no conceito de valor médio de uma onda periódica.
- O tempo em que se encontra em nível lógico alto é chamado de ciclo ativo (Duty Cycle);
- O valor médio de um sinal PWM pode ser encontrado utilizando o valor máximo da tensão e a razão entre período que se encontra ativo e o período total do sinal.

## PWM - Modulação por largura de pulso



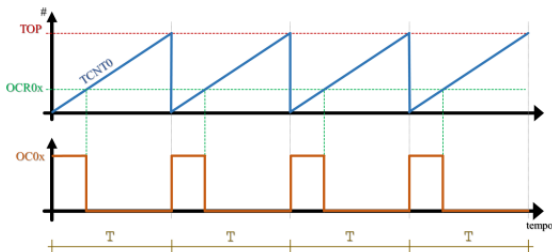
$$V_{\text{médio}} = \frac{T_{\text{ativo}}}{T} \cdot V_{\text{MAX}}$$

**Exemplo:** se o sinal digital tem variação de 0 a 5 V, um ciclo ativo de 50 % corresponde a um valor médio de 2,5 V, enquanto um ciclo ativo de 75 % corresponderia a 3,75 V.

## PWM - Modulação por largura de pulso

- O contador (Registrador TNT0) conta de zero até o valor máximo e volta a zero.
- No modo invertido, o pino OC0x é posto em 1 no valor mínimo do contador e posto em 0 na igualdade entre TCNT0 e OCR0x. O processo é inverso quando em saída invertida;
- O registrador OCR0A determina o ciclo ativo do sinal PWM no pino OC0A, quando habilitado;
- O registrador OCR0B determina o ciclo ativo do sinal PWM no pino OC0B, quando habilitado;

# PWM - Modulação por largura de pulso



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

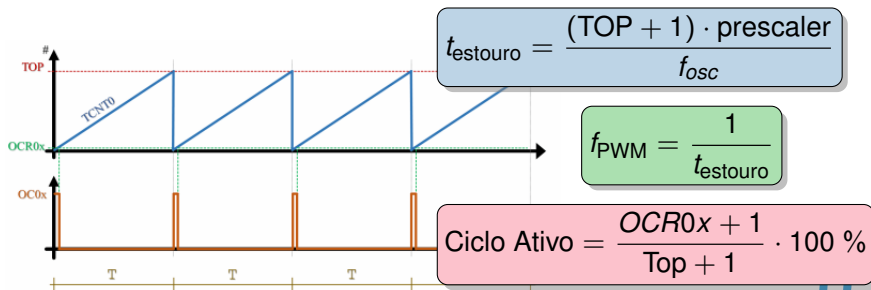
$$f_{\text{PWM}} = \frac{1}{t_{\text{estouro}}}$$

$$f_{\text{PWM}} = \frac{f_{osc}}{(TOP + 1) \cdot \text{prescaler}}$$

$$\text{Ciclo Ativo} = \frac{OCR0x + 1}{Top + 1} \cdot 100 \%$$

## TC0 - Modo PWM Rápido

- O valor de comparação OCR0x pode ser alterado, gerando um PWM variável.
- O valor zero para OCR0x produz um curto ciclo ativo e o valor máximo (255) deixa o sinal PWM em 0 ou 1, conforme foi habilitada a saída, invertida ou não;



## TC0 - Modo PWM rápido

Bit	7	6	5	4	3	2	1	0
TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00
Lê/Escreve	L/E	L/E	L/E	L/E	L	L	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

COM0A1	COM0A0	Descrição
0	0	Operação normal do pino, OC0A desconectado.
0	1	WGM02 = 0: operação normal do pino, OC0A desconectado. WGM02 = 1: troca de estado do OC0A na igualdade de comparação.
1	0	OC0A é limpo na igualdade de comparação, OC0A é ativo no valor do TC mínimo (modo não invertido).
1	1	OC0A é ativo na igualdade de comparação, OC0A é limpo no valor do TC mínimo (modo invertido).

## TC0 - Modo PWM rápido

Bit	7	6	5	4	3	2	1	0
TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00
Lê/Escreve	L/E	L/E	L/E	L/E	L	L	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
TCCR0B	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00
Lê/Escreve	E	E	L	L	L/E	L/E	L/E	L/E
Valor Inicial	0	0	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 Corrigido	OCR0A	Top (OCR0A)	0xFF
6	1	1	0	Reservado	-	-	-
7	1	1	1	PWM Rápido	OCR0A	0x00	0xFF



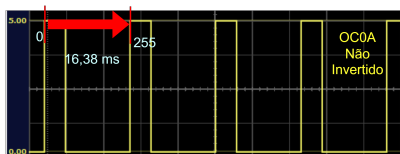
## TC0 - Modo PWM rápido

Bit	7	6	5	4	3	2	1	0
<b>TCCR0B</b>	<b>FOC0A</b>	<b>FOC0B</b>	<b>-</b>	<b>-</b>	<b>WGM02</b>	<b>CS02</b>	<b>CS01</b>	<b>CS00</b>
Lê/Escreve	E	E	L	L	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

CS02	CS01	CS00	Descrição
0	0	0	Sem fonte de clock (TC0 parado)
0	0	1	Clock/1 - Sem prescaler
0	1	0	Clock/8 (Prescaler de 8)
0	1	1	Clock/64 (Prescaler de 64)
1	0	0	Clock/256 (Prescaler de 256)
1	0	1	Clock/1024 (Prescaler de 1024)
1	1	0	Clock externo no Pino T0. Contagem de borda de descida
1	1	1	Clock externo no Pino T0. Contagem na borda de subida

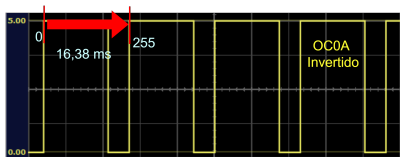
## TC0 - Modo PWM rápido

### ■ Modo Normal - Não Invertido



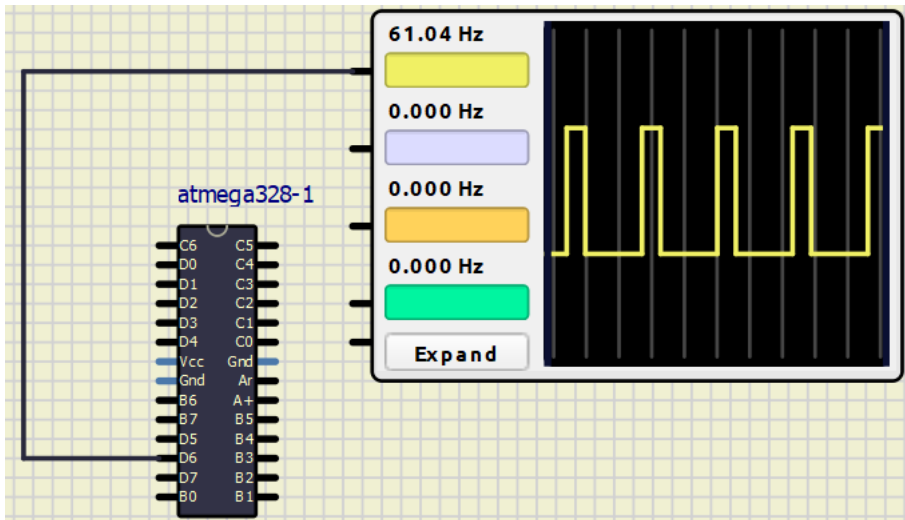
```
TCCR0A = (1 << COM0A1) | (1 << WGM01) | (1 << WGM00); // OC0A troca de estado  
TCCR0B = (1 << CS02) | (1 << CS00); //TC0 com prescaler = 1024  
OCR0A = 63; //Máximo valor de contagem do registrador TCNT0
```

### ■ Modo Invertido



```
TCCR0A = (1 << COM0A1) | (1 << COM0A0) | (1 << WGM01) | (1 << WGM00);  
TCCR0B = (1 << CS02) | (1 << CS00);  
OCR0A = 63;
```

## TC0 - Modo PWM rápido



## TC0 - Modo PWM rápido

```

1  #include <avr/io.h>
2  int main(){
3      DDRD = 0b01100000; // OC0A e OC0B = (PD5 e PD6) como saída
4      PORTD = 0b10011111; // OC0A e OC0B = 0, Hab. pull-ups dos demais pinos
5      TCCR0A = (1 << COM0A1) | (1 << WGM01) | (1 << WGM00);
6      TCCR0B = (1 << CS02) | (1 << CS00);
7      OCR0A = 63;
8      while(1); // Laço Infinito
9  }

```

$$t_{\text{estouro}} = \frac{(TOP + 1) \cdot \text{prescaler}}{f_{\text{osc}}} = \frac{256 \cdot 1.024}{16.000.000} = 16,384 \text{ ms}$$

$$f_{\text{PWM}} = \frac{1}{t_{\text{estouro}}} = \frac{1}{16,384 \cdot 10^{-3}} = 61,035 \text{ Hz}$$

$$\text{Ciclo Ativo} = \frac{OCR0A + 1}{Top + 1} \cdot 100 \% = \frac{63 + 1}{255 + 1} = \frac{64}{256} = 25 \% \Rightarrow 4,09 \text{ ms}$$

$$V_{\text{max}} = 0,25 \cdot 5 \text{ V} = 1,25 \text{ V}$$