



MCO60408

MICROCONTROLADORES

Aula 08 – Temporizadores/Contadores

SUMÁRIO

- Temporizadores/Contadores;
- TIMER0/TIMER1/TIMER2;
 - Modo Normal;
 - Modo CTC;
 - Modo PWM Rápido;
 - Modo PWM com Fase Corrigida;
 - Modo PWM com Fase e Frequência Corrigidas;
- Registradores;
- Configuração dos Temporizadores;
- Bibliotecas LS / R2R;
- Exercício;
- Leitura Recomendada.



TEMPORIZADORES/CONTADORES



TEMPORIZADORES/CONTADORES

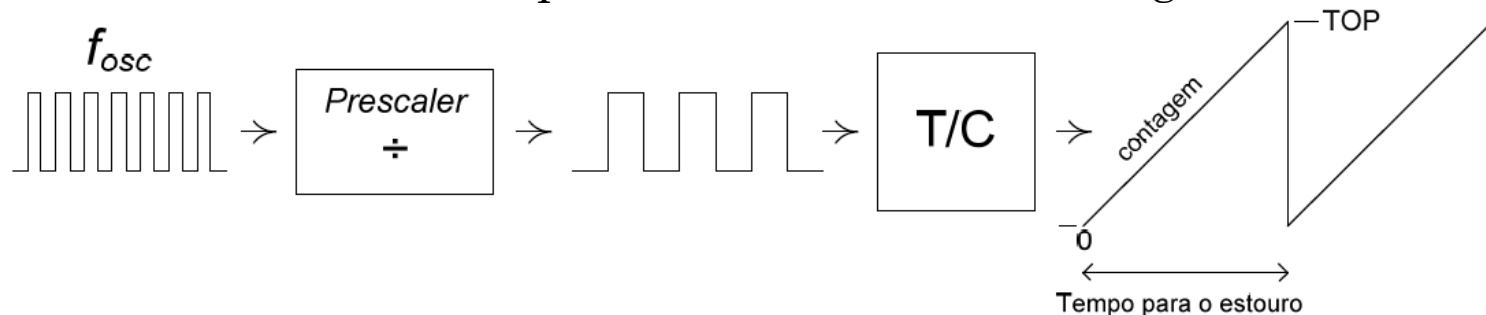
- São baseados em um circuito contador que incrementa um registrador a cada número específico de ciclos de relógio;
- São periféricos especializados em marcar a passagem do tempo, sem carregar a CPU;
- Quando ocorre o estouro do contador (*overflow*), uma interrupção pode ser ativada;
 - Permite a execução de código a intervalos regulares de tempo.
- O ATmega328 possui três temporizadores, sendo dois contadores de 8 bits (TIMER0 e TIMER2) e um contador de 16 bits (TIMER1).



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

TEMPORIZADORES/CONTADORES

Funcionamento de um temporizador/contador do Atmega.



Supondo um TC de 8 bits (Conta até o valor TOP 255), trabalhando com uma frequência de 1 MHz, sem divisor de frequência, o tempo para seu estouro é:

$$t_{estouro} = \frac{(255 + 1) \times 1}{1 \text{ MHz}} = 256\mu s$$

Supondo um TC de 16 bits (Conta até o valor TOP 65535), trabalhando com uma frequência de 16 MHz, com divisor de frequência de 64, o tempo para seu estouro é de:

$$t_{estouro} = \frac{(65535 + 1) \times 64}{16 \text{ MHz}} = 0,262s$$



MODULAÇÃO POR LARGURA DE PULSO – PWM

O uso desses sinais é baseado no conceito de valor médio de uma forma de onda periódica.

Dispositivos eletrônicos podem ser controladores com sinal de PWM:

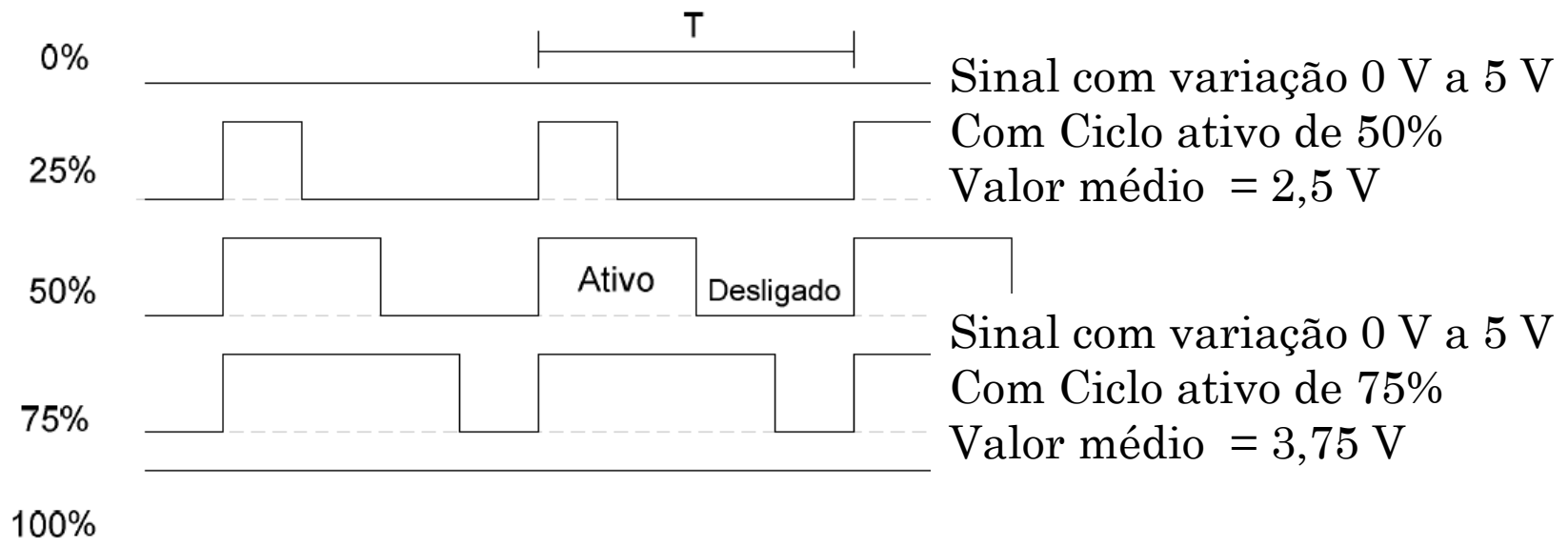
- Motores;
- Lâmpadas;
- LEDs;
- fontes chaveadas;
- circuitos inversores, etc.

O valor médio de uma forma de onda é controlado pelo tempo em que o sinal fica em nível lógico alto durante um determinado intervalo tempo. No PWM é chamado de Ciclo ativo (*Duty Cycle*).



MODULAÇÃO POR LARGURA DE PULSO – PWM

PWM com período T e ciclo ativo de 0%, 25%, 50%, 75% e 100%.



O calculo do valor médio de um sinal digital e dado por:

$$V_{\text{médio}} = \frac{\text{Amplitude Máxima}}{\text{Período}} \times (\text{Tempo Ativo no Período})$$



TIMER0/TIMER1/TIMER2



TIMER0/TIMER1/TIMER2

- O TIMER0 e o TIMER2 são temporizadores/contadores de uso geral de 8 bits;
 - Possuem duas saídas de comparação independentes (OCnA, OCnB);
 - Suportam geração de sinal PWM (*Pulse Width Modulation*).
- O TIMER1 é um temporizador/contador de uso geral de 16 bits;
 - Possui duas saídas de comparação independentes (OC1A, OC1B);
 - Suportam geração de sinal PWM (*Pulse Width Modulation*);
 - Suporta medição de duração de sinal.





O contador de 16 bits atinge **TOP** quando seu valor se torna igual ao maior valor de contagem. Pode ser igual a **0x00FF**, **0x01FF**, **0x03FF** ou igual ao valor de **OCR1A** ou de **ICR1**, dependendo do modo de operação.

DIAGRAMA EM BLOCOS DOS TEMPORIZADORES DO ATMEGA328

TIMER0/TIMER1/TIMER2

○ Modos de operação:

- O TIMER0 e o TIMER2 permitem o funcionamento em quatro modos de operação:
 - Modo normal;
 - Modo CTC (*Clear Timer on Compare match*);
 - Modo PWM rápido (*fast PWM*);
 - Modo PWM com fase corrigida (*phase correct PWM*).
- O TIMER1 permite o funcionamento em cinco modos de operação:
 - Modo normal;
 - Modo CTC (*Clear Timer on Compare match*);
 - Modo PWM rápido (*fast PWM*);
 - Modo PWM com fase corrigida (*phase correct PWM*);
 - Modo PWM com fase e frequência corrigidas (*phase and frequency correct PWM*).



MODO NORMAL



MODO NORMAL

- Modo normal;
 - Modo de operação mais simples;
 - A direção de contagem é sempre crescente e o contador não é limpo automaticamente na comparação;
 - O valor de TCNTn conta de BOTTOM até MAX e reinicia em BOTTOM;
 - A *flag* TOVn (*Timer/Counter Overflow Flag*) é setada no mesmo ciclo de relógio em que TCNTn se torna igual a zero;
 - A *flag* TOVn é setada por *hardware* e será limpa por *hardware* após o atendimento da interrupção;
 - O valor de TCNTn pode ser modificado por *software* a qualquer momento;



MODO NORMAL

○ Modo normal;

- As saídas de comparação podem ser utilizadas para gerar interrupções a cada intervalo de tempo;
- Utilizar as saídas de comparação para gerar sinais PWM em modo normal não é recomendado, uma vez que ocupará muito processamento da CPU;
- A frequência de estouro do contador é dada pela equação a seguir, com N sendo o *prescaler*.

$$f_{timer} = \frac{f_{clk_{I/O}}}{N \cdot (MAX + 1)}$$



Modo CTC

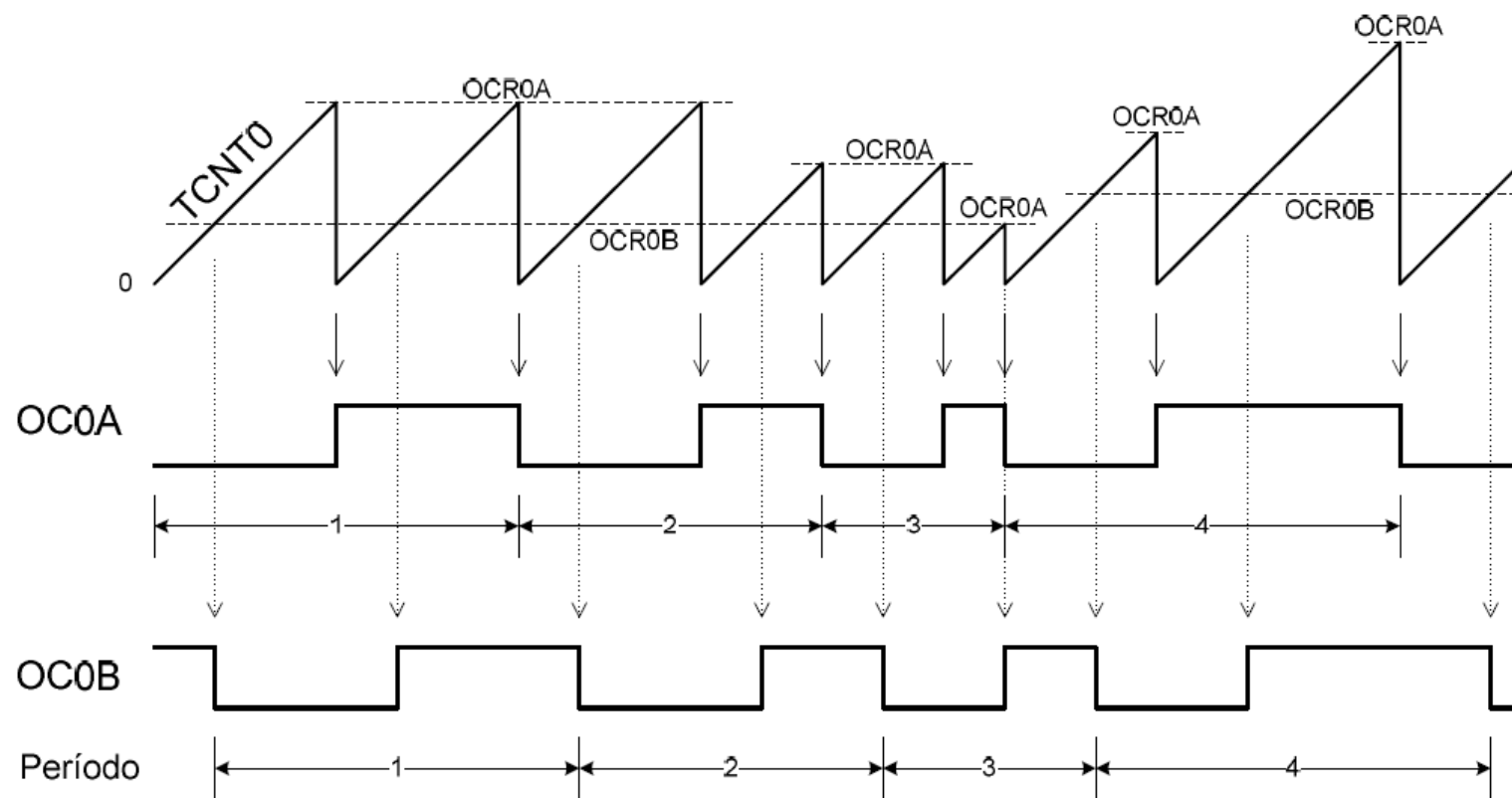


Modo CTC

- Modo CTC (*Clear Timer on Compare match*) ou (limpeza do contador na igualdade de comparação);
 - A resolução do contador é manipulada através de um registrador de comparação;
 - OCRnA (8 bits);
 - OCR1A ou ICR1 (16 bits).
 - O valor do contador (TCNTn) é zerado quando ele atinge o valor gravado no registrador de comparação;
 - O registrador de comparação define o valor de topo para o contador e, conseqüentemente, sua resolução;
 - Este modo permite controle maior da frequência de trabalho do contador.



Modo CTC



MOD0 PWM RÁPIDO



MODO PWM RÁPIDO

- Modo PWM Rápido (*Fast PWM*);
 - Provê geração de onda em alta frequência.
 - Atuar em rampa simples, contando de BOTTOM a TOP e reiniciando de BOTTOM.
 - No modo de saída de comparação não inversora, OCnx é limpo na comparação entre TCNTn e OCRnx, e setado em BOTTOM.
 - No modo de saída de comparação inversora, OCnx é setado na comparação entre TCNTn e OCRnx, e limpo em BOTTOM.
 - O valor de OCnx não estará visível no *port* a menos que o bit de direção do *port* esteja configurado como saída.

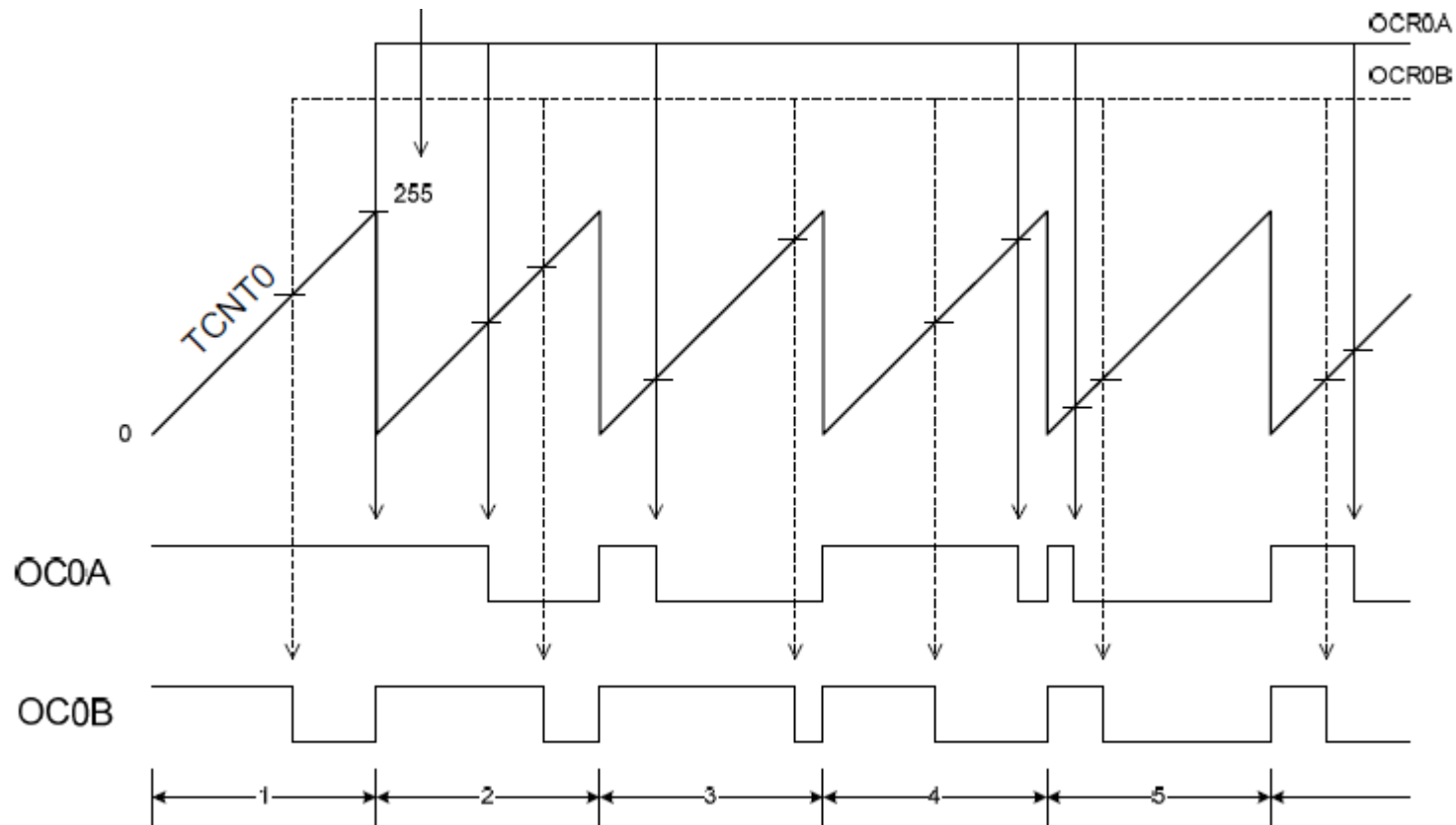


MODO PWM RÁPIDO

- Modo PWM Rápido (*Fast PWM*);
 - Devido à operação em rampa simples, a frequência do PWM pode ser o dobro dos modos PWM com correção de fase que operam em duas rampas;
 - Modo ideal para controle de operações de retificação, regulação de energia e conversão D/A.
 - O aumento da frequência diminui o tamanho dos componentes externos, reduzindo o tamanho e custo total.
 - A *flag* TOVn é setada cada vez que o contador atinge TOP. Se a interrupção estiver habilitada, a rotina de interrupção pode ser utilizada para modificar o valor de comparação.



MODULO PWM RÁPIDO



$$f_{OCn \times PWM} = \frac{f_{clk_{I/O}}}{N \cdot (TOP + 1)}$$



MODULO PWM COM FASE CORRIGIDA

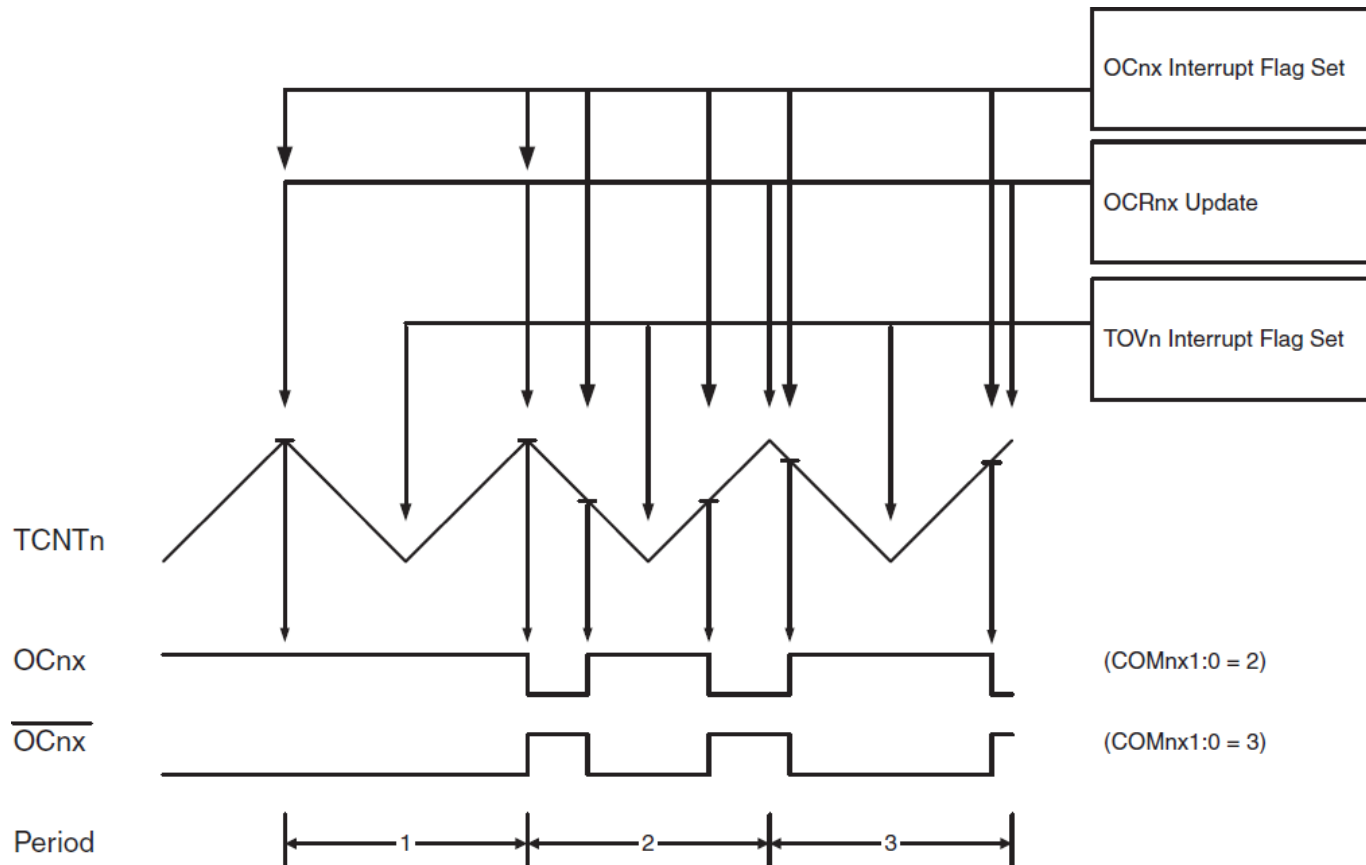


MODO PWM COM FASE CORRIGIDA

- Provê um sinal PWM de alta resolução e fase corrigida.
 - O modo opera em rampa dupla, ou seja, o contador conta de BOTTOM a TOP e depois retorna de TOP a BOTTOM. Ao final da rampa crescente, ao atingir TOP, o contador permanece um ciclo de relógio em TOP antes de inverter a direção da contagem.
- Em modo não inversor, OCnx é limpo na comparação na rampa crescente e setado na comparação na rampa decrescente. No modo inversor, a operação é o oposto do modo não inversor.
- A frequência de operação do PWM é menor que no modo PWM rápido, porém, devido à simetria da onda gerada, este modo é ideal para aplicações de controle de motores.



MODO PWM COM FASE CORRIGIDA



$$f_{OCnxPWM} = \frac{f_{clk_{I/O}}}{2 \cdot N \cdot TOP}$$



MODOS PWM COM FASE E FREQUÊNCIA CORRIGIDAS

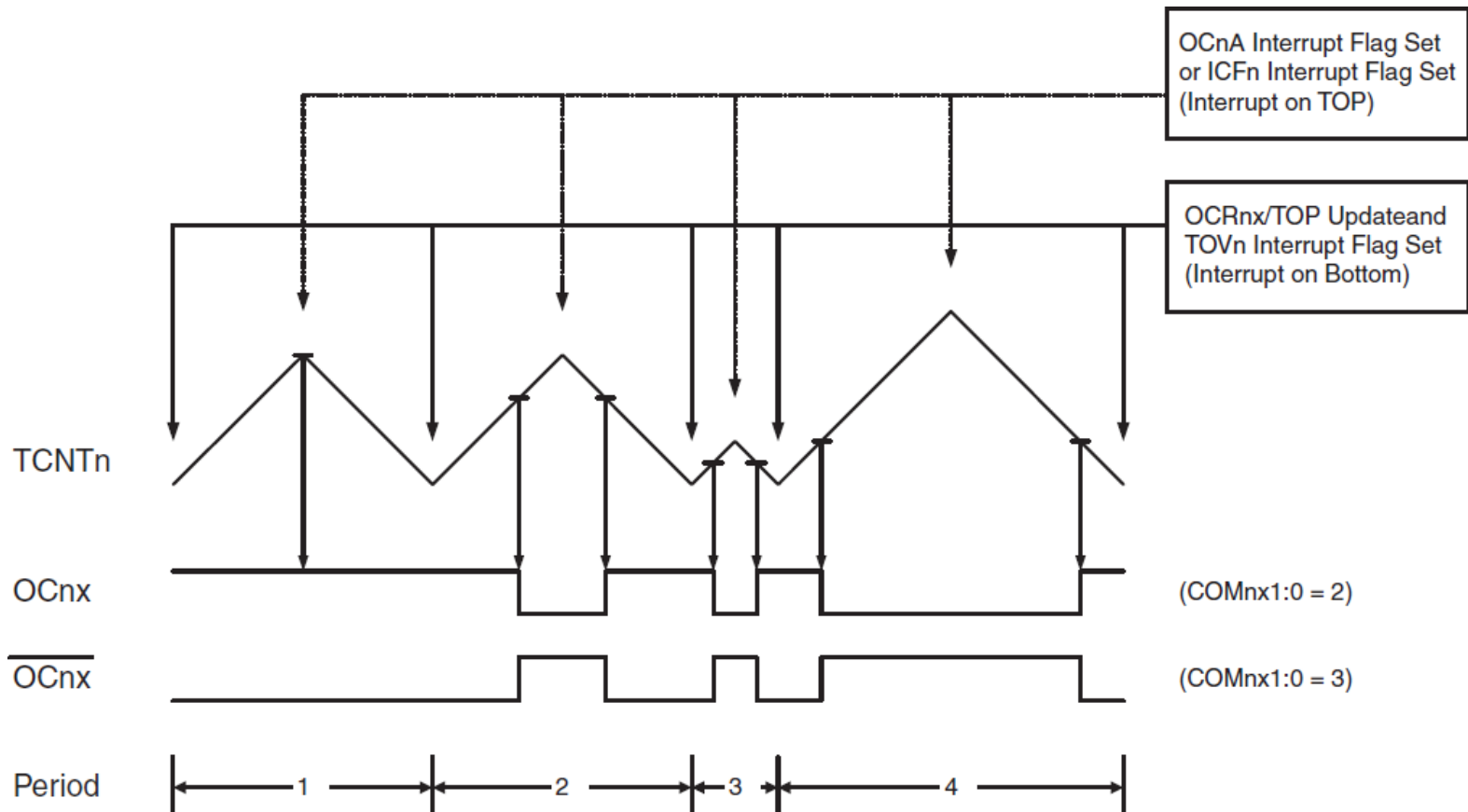


MODOS PWM COM FASE E FREQUÊNCIA CORRIGIDAS

- Provê um sinal PWM de alta resolução e fase e frequência corrigidas.
 - O modo opera em rampa dupla com dupla comparação, ou seja, o contador conta de BOTTOM a TOP e depois retorna de TOP a BOTTOM. Ao final da rampa crescente, ao atingir TOP, o contador permanece um ciclo de relógio em TOP antes de inverter a direção da contagem.
- Em modo não inversor, OCnx é limpo na comparação na rampa crescente e setado na comparação na rampa decrescente. No modo inversor, a operação é o oposto do modo não inversor.
- A frequência de operação do PWM é menor que no modo PWM rápido, porém, devido à simetria da onda gerada, este modo é ideal para aplicações de controle de motores.



MODOS PWM COM FASE E FREQUÊNCIA CORRIGIDAS



$$f_{OCnxPWM} = \frac{f_{clkI/O}}{2 \cdot N \cdot TOP}$$



REGISTRADORES



REGISTRADORES

- O funcionamento dos temporizadores é mediado por registradores de controle e de valores:
 - *Timer/Counter Control Register A.*

TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCCR2A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
(0xB0)	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	TCCR2A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



REGISTRADORES

- *Timer/Counter Control Register B.*

TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

TCCR2B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0
(0xB1)	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0



REGISTRADORES

- *Timer/Counter Control Register C.*
 - Disponível apenas para o TIMER1.

TCCR1C – Timer/Counter1 Control Register C

Bit	7	6	5	4	3	2	1	0
(0x82)	FOC1A	FOC1B	–	–	–	–	–	–
Read/Write	R/W	R/W	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0



REGISTRADORES

- *Timer/Counter Register.*
 - Armazena o valor do contador, é incrementado a cada ciclo do relógio do temporizador.

TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCNT1H and TCNT1L – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	TCNT1[7:0]								TCNT1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCNT2 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
(0xB2)	TCNT2[7:0]								TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



REGISTRADORES

- *Output Compare Register A.*
 - Armazena o valor de comparação A.

OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

OCR1AH and OCR1AL – Output Compare Register 1 A

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

OCR2A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
(0xB3)	OCR2A[7:0]								OCR2A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



REGISTRADORES

- *Output Compare Register B.*
 - Armazena o valor de comparação B.

OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x28 (0x48)	OCR0B[7:0]								OCR0B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

OCR1BH and OCR1BL – Output Compare Register 1 B

Bit	7	6	5	4	3	2	1	0	
(0x8B)	OCR1B[15:8]								OCR1BH
(0x8A)	OCR1B[7:0]								OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

OCR2B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
(0xB4)	OCR2B[7:0]								OCR2B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



REGISTRADORES

- *Input Capture Register.*
 - Pode ser utilizado para contar o tempo entre pulsos no pino externo ICP.

ICR1H and ICR1L – Input Capture Register 1

Bit	7	6	5	4	3	2	1	0	
(0x87)	ICR1[15:8]								ICR1H
(0x86)	ICR1[7:0]								ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



REGISTRADORES

- *Timer/Counter Interrupt Mask Register.*
 - Máscara de interrupções dos temporizadores.

TIMSK0 – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIMSK1 – Timer/Counter1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIMSK2 – Timer/Counter2 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x70)	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	TIMSK2
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



REGISTRADORES

- *Timer/Counter Interrupt Flag Register.*
 - Sinaliza as interrupções dos temporizadores.

TIFR0 – Timer/Counter 0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x15 (0x35)	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIFR1 – Timer/Counter1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TIFR2 – Timer/Counter2 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	–	–	–	–	–	OCF2B	OCF2A	TOV2	TIFR2
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



REGISTRADORES

- *Asynchronous Status Register.*
 - Registrador de controle e estado do temporizador assíncrono.

ASSR – Asynchronous Status Register

Bit	7	6	5	4	3	2	1	0	
(0xB6)	–	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	ASSR
Read/Write	R	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- *General Timer/Counter Control Register.*
 - Registrador de controle geral dos temporizadores.

GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	TSM	–	–	–	–	–	PSRASY	PSRSYNC	GTCCR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



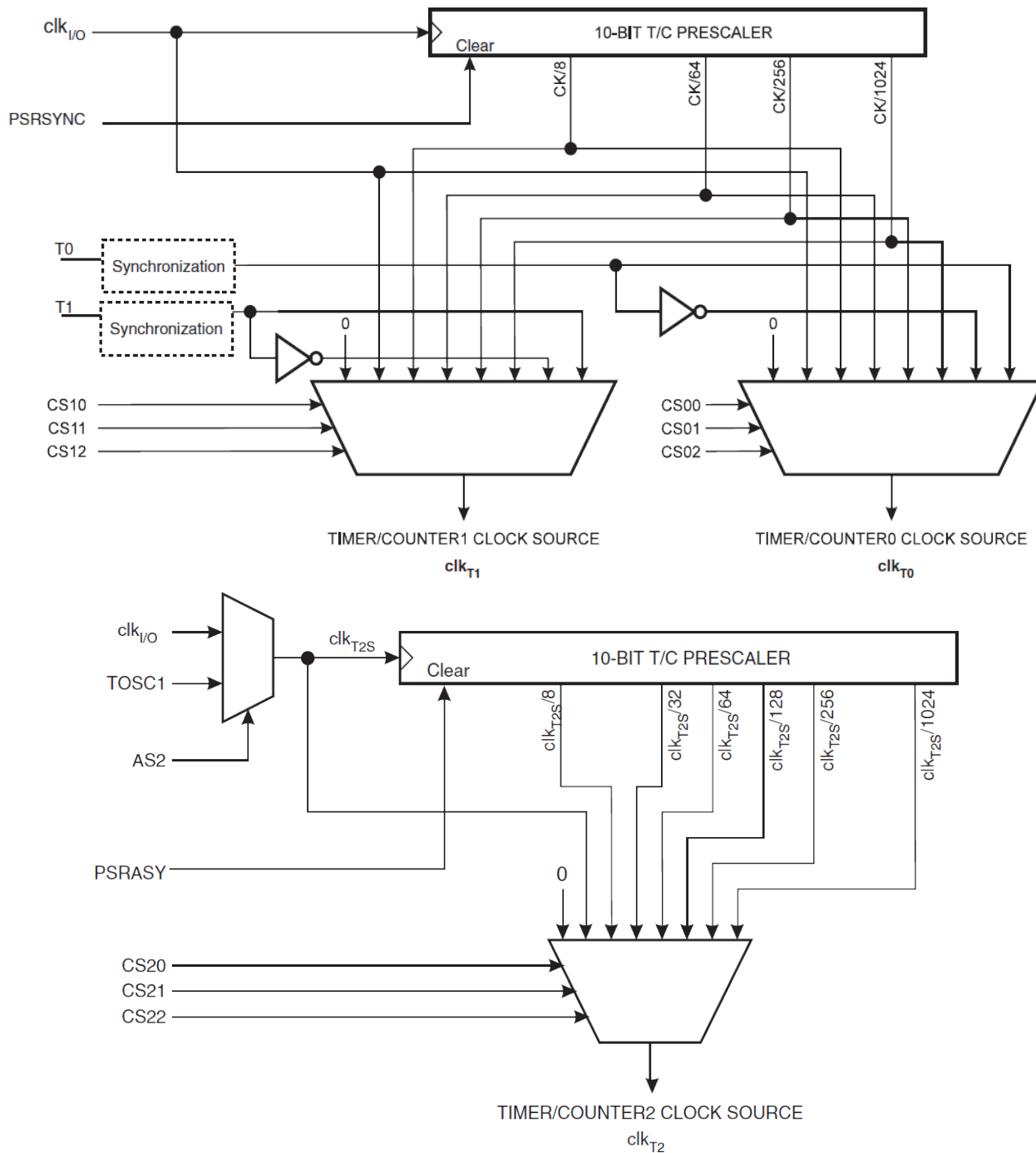
CONFIGURAÇÃO DOS TEMPORIZADORES



CONFIGURAÇÃO DOS TEMPORIZADORES

- Existem diversas configurações distintas para os temporizadores do ATmega328;
- Para facilitar o entendimento, as configurações foram subdivididas nos seguintes tópicos:
 - *Prescaler* do relógio;
 - Modo de funcionamento;
 - Comportamento das saídas de comparação A e B;
 - Interrupções;
 - Captura de eventos;
 - Forçar comparação;
 - Configuração de valores e níveis.





PRESCALER DO RELÓGIO – TIMER0 E TIMER1

Todos os temporizadores do Atmega328 possuem seleção individual, tanto para a fonte do relógio quanto para seu *prescaler*.

TIMER0 e TIMER1

Os temporizadores 0 e 1 permitem oito configurações de relógio: desligado, relógio principal sem *prescaler*, relógio principal com *prescaler* de 8, 64, 256 e 1024 e pino externo (T0 e T1) na borda de subida e de descida.

TIMER2

O temporizador 2 também permite oito configurações de relógio: desligado, relógio principal sem *prescaler* e relógio principal com *prescaler* de 8, 32, 64, 128, 256 e 1024. Este temporizador não permite fonte de relógio baseado em pino externo.

CS12	CS11	CS10	Descrição
0	0	0	Sem fonte de <i>clock</i> (TC1 parado).
0	0	1	<i>clock</i> /1 (<i>prescaler</i> = 1) sem <i>prescaler</i> .
0	1	0	<i>clock</i> /8 (<i>prescaler</i> = 8) .
0	1	1	<i>clock</i> /64 (<i>prescaler</i> = 64).
1	0	0	<i>clock</i> /256 (<i>prescaler</i> = 256).
1	0	1	<i>clock</i> /1024(<i>prescaler</i> = 1024).
1	1	0	Fonte de <i>clock</i> externa no pino T1 (contagem na borda de descida).
1	1	1	Fonte de <i>clock</i> externa no pino T1 (contagem na borda de subida).

CS22	CS21	CS20	Descrição
0	0	0	Sem fonte de <i>clock</i> (TC2 parado)
0	0	1	<i>clock</i> /1 (<i>prescaler</i> =1) - sem <i>prescaler</i>
0	1	0	<i>clock</i> /8 (<i>prescaler</i> = 8)
0	1	1	<i>clock</i> /32 (<i>prescaler</i> = 32)
1	0	0	<i>clock</i> /64 (<i>prescaler</i> = 64)
1	0	1	<i>clock</i> /128 (<i>prescaler</i> = 128)
1	1	0	<i>clock</i> /256 (<i>prescaler</i> = 256)
1	1	1	<i>clock</i> /1024 (<i>prescaler</i> = 1024)

TIMER0 e TIMER1

O *prescaler* do relógio dos temporizadores 0 e 1 é configurado através dos bits 2, 1 e 0 do registrador de controle TCCR0B e TCCR1B, respectivamente.

TIMER2

O *prescaler* do relógio do temporizador 2 é configurado através dos bits 2, 1 e 0 do registrador de controle TCCR2B.



TIMER0 e TIMER2

A seleção do modo de funcionamento dos temporizadores 0 e 2 é realizada pela configuração do pino 3 do registrador de controle B (TCCR0B e TCCR2B, respectivamente) e dos pinos 1 e 0 do registrador de controle A (TCCR0A e TCCR2A, respectivamente).

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

MODO DE FUNCIONAMENTO

TIMER1

A seleção do modo de funcionamento do temporizador 1 é realizada pela configuração dos pinos 4 e 3 do registrador de controle B (TCCR1B) e dos pinos 1 e 0 do registrador de controle A (TCCR1A).



Mo do	WGM 13	WGM 12	WGM 11	WGM 10	Modo de operação do TC1	Valor TOP	Atualiz. OCR1x no valor	Bit TOV1 ativo no valor:
0	0	0	0	0	Normal	0xFFFF	Imediata	0xFFFF
1	0	0	0	1	PWM com fase corrigida, 8 bits	0x00FF	0x00FF	0
2	0	0	1	0	PWM com fase corrigida, 9 bits	0x01FF	0x01FF	0
3	0	0	1	1	PWM com fase corrigida, 10 bits	0x03FF	0x03FF	0
4	0	1	0	0	CTC	OCR1A	Imediata	0xFFFF
5	0	1	0	1	PWM rápido, 8 bits	0x00FF	0	0x00FF
6	0	1	1	0	PWM rápido, 9 bits	0x01FF	0	0x01FF
7	0	1	1	1	PWM rápido, 10 bits	0x03FF	0	0x03FF
8	1	0	0	0	PWM com fase e freq. corrigidas	ICR1	0	0
9	1	0	0	1	PWM com fase e freq. corrigidas	OCR1A	0	0
10	1	0	1	0	PWM com fase corrigida	ICR1	ICR1	0
11	1	0	1	1	PWM com fase corrigida	OCR1A	OCR1A	0
12	1	1	0	0	CTC	ICR1	Imediata	0xFFFF
13	1	1	0	1	Reservado	-	-	-
14	1	1	1	0	PWM rápido	ICR1	0	ICR1
15	1	1	1	1	PWM rápido	OCR1A	0	OCR1A

COMPORTAMENTO DAS SAÍDAS DE COMPARAÇÃO A E B

As saídas de comparação A e B são configuradas de individualmente. As funções implementadas em cada pino externo (OCnA e OCnB) dependem do modo de funcionamento.

A função dos pinos é configurada nos bits 7 e 6 (saída A) e 5 e 4 (saída B) do registrador de controle A dos temporizadores 0, 1 e 2 (TCCR0A, TCCR1A e TCCR2A).

Modo não PWM

Nos modos não PWM (Normal e CTC), as saídas A e B podem ser configuradas como *ports* normais, ou para inverterem, limparem ou setarem durante a comparação.



COMnA1	COMnA0	Description
0	0	Normal port operation, OCnA disconnected.
0	1	Toggle OCnA on Compare Match
1	0	Clear OCnA on Compare Match
1	1	Set OCnA on Compare Match

COMnB1	COMnB0	Description
0	0	Normal port operation, OCnB disconnected.
0	1	Toggle OCnB on Compare Match
1	0	Clear OCnB on Compare Match
1	1	Set OCnB on Compare Match

COMPORTAMENTO DAS SAÍDAS DE COMPARAÇÃO A E B

Modo PWM Rápido

No modo PWM rápido, as saídas podem ser configuradas para operação normal do *port*, inversão na comparação com A, ou nos modos não inversor (limpa na comparação e seta em BOTTOM) e inversor (seta na comparação e limpa em BOTTOM) .



COMnA1	COMnA0	Description
0	0	Normal port operation, OCnA disconnected.
0	1	WGMn2 = 0: Normal Port Operation, OCnA Disconnected. WGMn2 = 1: Toggle OCnA on Compare Match.
1	0	Clear OCnA on Compare Match, set OCnA at BOTTOM, (non-inverting mode).
1	1	Set OCnA on Compare Match, clear OCnA at BOTTOM, (inverting mode).

COMnB1	COMnB0	Description
0	0	Normal port operation, OCnB disconnected.
0	1	Reserved
1	0	Clear OCnB on Compare Match, set OCnB at BOTTOM, (non-inverting mode).
1	1	Set OCnB on Compare Match, clear OCnB at BOTTOM, (inverting mode).

COM1A1/ COM1B1	COM1A0/ COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match, set OC1A/OC1B at BOTTOM (non-inverting mode)
1	1	Set OC1A/OC1B on Compare Match, clear OC1A/OC1B at BOTTOM (inverting mode)

COMPORTAMENTO DAS SAÍDAS DE COMPARAÇÃO A E B

Modo PWM com Fase Corrigida / PWM com Fase e Frequência Corrigidas

Nos modos PWM com Fase Corrigida e PWM com Fase e Frequência Corrigidas, as saídas podem ser configuradas para operação normal do *port*, inversão na comparação com A, ou nos modos não inversor (limpa na comparação durante a contagem crescente e seta na comparação durante a contagem decrescente) e inversor (seta na comparação durante a contagem crescente e limpa na comparação durante a contagem decrescente).




COMnA1	COMnA0	Description
0	0	Normal port operation, OCnA disconnected.
0	1	WGM02 = 0: Normal Port Operation, OCnA Disconnected. WGM02 = 1: Toggle OCnA on Compare Match.
1	0	Clear OCnA on Compare Match when up-counting. Set OCnA on Compare Match when down-counting.
1	1	Set OCnA on Compare Match when up-counting. Clear OCnA on Compare Match when down-counting.

COMnB1	COMnB0	Description
0	0	Normal port operation, OCnB disconnected.
0	1	Reserved
1	0	Clear OCnB on Compare Match when up-counting. Set OCnB on Compare Match when down-counting.
1	1	Set OCnB on Compare Match when up-counting. Clear OCnB on Compare Match when down-counting.

COM1A1/ COM1B1	COM1A0/ COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	WGM13:0 = 9 or 11: Toggle OC1A on Compare Match, OC1B disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B disconnected.
1	0	Clear OC1A/OC1B on Compare Match when up-counting. Set OC1A/OC1B on Compare Match when downcounting.
1	1	Set OC1A/OC1B on Compare Match when up-counting. Clear OC1A/OC1B on Compare Match when downcounting.

INTERRUPÇÕES


- Bit **OCIE_nB** do registrador **TIMSK_n**;
 - Habilita a interrupção por comparação com **OCR_nB**, ou seja, quando o bit **OCF_nB** no registrador **TIFR_n** for setado.
 - Bit **OCIE_nA** do registrador **TIMSK_n**;
 - Habilita a interrupção por comparação com **OCR_nA**, ou seja, quando o bit **OCF_nA** no registrador **TIFR_n** for setado.
 - Bit **TOIE_n** do registrador **TIMSK_n**;
 - Habilita a interrupção por estouro do contador, ou seja, quando o bit **TOV_n** no registrador **TIFR_n** for setado.
 - Bit **ICIE1** do registrador **TIMSK1**;
 - Habilita a interrupção por captura do contador, ou seja, quando o bit **ICF1** no registrador **TIFR1** for setado.
- 

CAPTURA DE EVENTOS

○ Bit **ICNC1** do registrador **TCCR1B**;

- Setar este bit ativa o cancelador de ruído da captura de eventos. Quando ativo, o pino de captura de eventos (ICP1) é filtrado. A função filtro requer quatro amostras sucessivas do pino ICP1, logo a captura de eventos é atrasada quatro ciclos de relógio quando o cancelador de ruídos estiver ativo.

○ Bit **ICES1** do registrador **TCCR1B**;

- Seleciona a borda de subida (setado) ou de descida (limpo) utilizada para disparar a captura de eventos.
 - Quando o registrador **ICR1** é utilizado como valor TOP, o pino ICP1 é desconectado e, conseqüentemente, a função de captura de eventos é desativada.
- 

FORÇAR COMPARAÇÃO

- Bit **FOCnx** do registrador **TCCRnB/TCCR1C**;
 - O bit somente estará ativo nos modos não-PWM. Setar este pino irá forçar a comparação com x, atuando em OCnx, porém não gera interrupção, nem limpa o timer no modo CTC.
 - Para compatibilidade com dispositivos futuros, deve ser escrito em zero quando atualizar o registrador TCCRnB/TCCR1C em modo PWM.



CONFIGURAR VALORES E NÍVEIS

- Registrador do contador **TCNTn**;
 - Registrador de 8/16 bits que armazena o valor do contador. Incrementa a cada número definido de ciclos de relógio.
- Registrador de comparação com x **OCRnx**;
 - Registrador de 8/16 bits que armazena o valor para comparação com x. Pode ser utilizado para gerar formas de onda ou para gerar interrupções.
- Registrador do contador **ICR1**;
 - Registrador de 16 bits que é atualizado com o valor do contador (TCNT1) a cada vez que um evento ocorre no pino ICP1 (ou na saída do Comparador Analógico para o TIMER1). O registrador pode ser usado para definir o valor TOP do contador.

EXEMPLOS

○ HABILITANDO A INTERRUPÇÃO POR ESTOURO DO TCO

```
7  #define F_CPU 16000000UL
8  #include <avr/io.h>
9  #include <avr/interrupt.h>
10
11  #define cpl_bit(y,bit)  (y^=(1<<bit))  //troca o estado lógico do bit x da variável Y
12  #define LED            PB5
13
14  //-----
15  ISR(TIMERO_OVF_vect)                //interrupção do TCO
16  {
17      cpl_bit(PORTB,LED);
18  }
19  //-----
20  int main()
21  {
22      DDRB = 0b00100000;                //somente pino do LED como saída
23      PORTB = 0b11011111;                //apaga LED e habilita pull-ups nos pinos não utilizados
24
25      TCCR0B = (1<<CS02) | (1<<CS00);    //TC0 com prescaler de 1024, a 16 MHz gera uma interrupção a cada 16,384 ms
26      TIMSK0 = 1<<TOIE0;                //habilita a interrupção do TCO
27      sei();                             //habilita interrupções globais
28
29      while(1)
30      {
31          //Aqui vai o código, a cada estouro do TCO o programa desvia para ISR(TIMERO_OVF_vect)
32      }
33  }
34  //=====
```

EXEMPLOS

○ HABILITANDO PWM

```
5  #define F_CPU 16000000UL
6  #include <avr/io.h>
7
8  int main(void)
9  {
10     DDRD = 0b01100000;    //pinos OC0B e OC0A (PD5 e PD6) como saída
11     PORTD = 0b10011111;   //zera saídas e habilita pull-ups nos pinos não utilizados
12
13     //MODO CTC
14     TCCR0A = 0b01010010;  //habilita OC0A e OC0B para trocar de estado na igualdade de comparação
15     TCCR0B = 0b00000001;  //liga TC0 com prescaler = 1.
16     OCR0A = 200;          //maximo valor de contagem
17     OCR0B = 100;         //deslocamento de OC0B em relação a OC0A
18
19     while(1)
20     {
21         //O programa principal vai aqui
22     }
23 }
24 //-----
```



EXEMPLOS

○ HABILITANDO INTERRUPÇÃO POR TC1

```
8  #define F_CPU 16000000UL
9  #include <avr/io.h>
10 #include <avr/interrupt.h>
11
12 #define cpl_bit(y,bit)  (y^=(1<<bit))  //troca o estado lógico do bit x da variável Y
13 #define LED            PB5
14
15 //-----
16 ISR(TIMER1_OVF_vect)                //interrupção do TC1
17 {
18     cpl_bit(PORTB,LED);
19 }
20 //-----
21 int main()
22 {
23     DDRB = 0b00100000;                //somente pino do LED como saída
24     PORTB = 0b11011111;                //apaga LED e habilita pull-ups nos pinos não utilizados
25
26     TCCR1B = (1<<CS12) | (1<<CS10);    //TC1 com prescaler de 1024, a 16 MHz gera uma interrupção a cada 4,19 s
27     TIMSK1 = 1<<TOIE1;                //habilita a interrupção do TC1
28     sei();                             //habilita interrupções globais
29
30     while(1)
31     {
32         //Aqui vai o código, a cada estouro do TC1 o programa desvia para ISR(TIMER1_OVF_vect)
33     }
34 }
35 //=====
```

EXEMPLOS TC2 COM CRISTAL EXTERNO DE 32,768 kHz

ROTINA PARA IMPLEMENTAR UM RELÓGIO E PISCAR UM LED

```
7 //===== //
8
9 #define F_CPU 1000000UL //frequência de operação de 1 MHz
10 #include <avr/io.h>
11 #include <avr/interrupt.h>
12
13 #define cpl_bit(Y,bit_x) (Y^=(1<<bit_x)) //complementa bit
14 #define LED PB5
15
16 volatile unsigned char segundos, minutos, horas;
17
18 //-----
19 ISR(TIMER2_OVF_vect) //entrada aqui a cada 1 segundo
20 { //rotina para contagem das horas, minutos e segundos
21     cpl_bit(PORTB,LED); //pisca LED
22
23     segundos++;
24
25     if(segundos == 60)
26     {
27         segundos = 0;
28         minutos++;
29
30         if (minutos == 60)
31         {
32             minutos = 0;
33             horas++;
34
35             if (horas == 24)
36                 horas = 0;
37         }
38     }
39 }
40 //-----
```

EXEMPLOS TC2 COM CRISTAL EXTERNO DE 32,768 kHz

ROTINA PARA IMPLEMENTAR UM RELÓGIO E PISCAR UM LED

```
41 int main()
42 {
43     DDRB = 0b00100000; //cristal ext., não importa a config. os pinos TOSC1 e TOSC2
44     PORTB= 0b11011111; //pull-ups habilitados nos pinos não utilizados
45
46     ASSR = 1<<AS2; //habilita o cristal externo para o contador de tempo real
47     TCCR2B = (1<<CS22) | (1<<CS20); /*prescaler = 128, freq. p/ o contador -> 32.768/128 = 256 Hz.
48                                     Como o contador é de 8 bits, ele conta 256 vezes,
49                                     resultando em um estouro preciso a cada 1 segundo*/
50     TIMSK2 = 1<<TOIE2; //habilita interrupção do TC2
51
52     sei(); //habilita interrupção global
53
54     while(1)
55     {} //código principal (display, ajuste de hora, minutos, etc..)
56 }
57 //=====
```

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

ATmega328 DIP-28

BIBLIOTECAS LS / R2R



BIBLIOTECAS LS / R2R

- As bibliotecas LS são formadas por três camadas de arquivos:
 - Arquivo de configurações do projeto;
 - Arquivo LS_defines.h, que deve ser modificado de acordo com as configurações de hardware do projeto em questão. Linhas devem ser comentadas e valores modificados, porém nada pode ser excluído do arquivo.
 - Arquivos de configuração de microcontrolador;
 - Arquivos H e C do microcontrolador. Contém as configurações dos registradores e dos periféricos do microcontrolador. Não deve ser alterado pelo usuário.
 - Arquivos de módulos externos;
 - Arquivos H e C de módulos específicos (LCD, teclado, etc.). Não deve ser alterado pelo usuário.



BIBLIOTECAS LS / R2R

○ Licença:

- As bibliotecas LS fazem parte de um projeto mantido pela R2R Tecnologia – Empresa Júnior. Empresa júnior do Departamento Acadêmico de Eletrônica do *campus* Florianópolis do IFSC.
- As bibliotecas LS podem ser utilizadas em projetos acadêmicos ou comerciais, contanto que não sejam modificadas. Se houver a necessidade de alteração das bibliotecas, entrar em contato com a R2R Tecnologia. Se utilizadas em projetos acadêmicos, devem ser citados e referenciados em relatórios, TCCs, monografias, dissertações, teses, artigos e outros documentos técnico-científicos.



BIBLIOTECAS LS / R2R

○ LS_ATmega328.h / LS_ATmega328.c

- Download:

- <http://pastebin.com/GYE107Yc>
- <http://pastebin.com/VcqvrBkw>

- *Prescaler* do relógio do Timer0/ Timer1;

- timer0ClockDisable() timer1ClockDisable()
- timer0ClockPrescallerOff() timer1ClockPrescallerOff()
- timer0ClockPrescaller8() timer1ClockPrescaller8()
- timer0ClockPrescaller64() timer1ClockPrescaller64()
- timer0ClockPrescaller256() timer1ClockPrescaller256()
- timer0ClockPrescaller1024() timer1ClockPrescaller1024()
- timer0ClockT0FallingEdge() timer1ClockT1FallingEdge()
- timer0ClockT0RisingEdge() timer1ClockT1RisingEdge()



BIBLIOTECAS LS / R2R

- *Prescaler* do relógio do Timer1;
 - timer2ClockDisable()
 - timer2ClockPrescallerOff()
 - timer2ClockPrescaller8()
 - timer2ClockPrescaller32()
 - timer2ClockPrescaller64()
 - timer2ClockPrescaller128()
 - timer2ClockPrescaller256()
 - timer2ClockPrescaller1024()
- Modo de funcionamento do Timer0/Timer1/Timer2;
 - timer0NormalMode()
 - timer1NormalMode()
 - timer2NormalMode()
 - timer1CTCICR1Mode()



BIBLIOTECAS LS / R2R

- timer0CTCMode()
- timer1CTCMode()
- timer2CTCMode()
- timer0FastPWMMaxMode()
- timer1FastPWM8bitsMode()
- timer1FastPWM9bitsMode()
- timer1FastPWM10bitsMode()
- timer2FastPWMMaxMode()
- timer0FastPWMOCR0AMode()
- timer1FastPWMOCR1AMode()
- timer2FastPWMOCR2AMode()
- timer1FastPWMICR1Mode()
- timer0PhaseCorrectPWMMaxMode()
- timer1PhaseCorrectPWM8bitsMode()
- timer1PhaseCorrectPWM9bitsMode()



BIBLIOTECAS LS / R2R

- timer1PhaseCorrectPWM10bitsMode()
 - timer2PhaseCorrectPWMMaxMode()
 - timer0PhaseCorrectPWMOCR0AMode()
 - timer1PhaseCorrectPWMOCR1AMode()
 - timer2PhaseCorrectPWMOCR2AMode()
 - timer1PhaseCorrectPWMICR1Mode()
 - timer1PhaseFrequencyCorrectPWMICR1Mode()
 - timer1PhaseFrequencyCorrectPWMOCR1AMode()
-
- Comportamento das saídas de comparação A
 - timer0OC0AOff()
 - timer1OC1AOff()
 - timer2OC2AOff()



BIBLIOTECAS LS / R2R

- timer0OC0AToggleOnCompare()
- timer1OC1AToggleOnCompare()
- timer2OC2AToggleOnCompare()
- timer0OC0AClearOnCompare()
- timer1OC1AClearOnCompare()
- timer2OC2AClearOnCompare()
- timer0OC0ASetOnCompare()
- timer1OC1ASetOnCompare()
- timer2OC2ASetOnCompare()
- timer0OC0ANonInvertedMode()
- timer1OC1ANonInvertedMode()
- timer2OC2ANonInvertedMode()
- timer0OC0AInvertedMode()
- timer1OC1AInvertedMode()
- timer2OC2AInvertedMode()



BIBLIOTECAS LS / R2R

- Comportamento das saídas de comparação B
 - timer0OC0BOff()
 - timer1OC1BOff()
 - timer2OC2BOff()
 - timer0OC0BToggleOnCompare()
 - timer1OC1BToggleOnCompare()
 - timer2OC2BToggleOnCompare()
 - timer0OC0BClearOnCompare()
 - timer1OC1BClearOnCompare()
 - timer2OC2BClearOnCompare()
 - timer0OC0BSetOnCompare()
 - timer1OC1BSetOnCompare()
 - timer2OC2BSetOnCompare()
 - timer0OC0BNonInvertedMode()
 - timer1OC1BNonInvertedMode()



BIBLIOTECAS LS / R2R

- timer2OC2BNonInvertedMode()
- timer0OC0BInvertedMode()
- timer1OC1BInvertedMode()
- timer2OC2BInvertedMode()
- Captura de eventos
 - timer1EnableInputCaptureNoiseCanceler()
 - timer1DisableInputCaptureNoiseCanceler()
 - timer1InputCaptureOnFallingEdge()
 - timer1InputCaptureOnRisingEdge()
- Interrupção de estouro
 - timer0ActivateOverflowInterrupt()
 - timer1ActivateOverflowInterrupt()
 - timer2ActivateOverflowInterrupt()



BIBLIOTECAS LS / R2R

- timer0DeactivateOverflowInterrupt()
- timer1DeactivateOverflowInterrupt()
- timer2DeactivateOverflowInterrupt()
- timer0ClearOverflowInterruptRequest()
- timer1ClearOverflowInterruptRequest()
- timer2ClearOverflowInterruptRequest()
- Interrupção de comparação com A
 - timer0ActivateCompareAInterrupt()
 - timer1ActivateCompareAInterrupt()
 - timer2ActivateCompareAInterrupt()
 - timer0DeactivateCompareAInterrupt()
 - timer1DeactivateCompareAInterrupt()
 - timer2DeactivateCompareAInterrupt()



BIBLIOTECAS LS / R2R

- timer0ClearCompareInterruptRequest()
- timer1ClearCompareInterruptRequest()
- timer2ClearCompareInterruptRequest()
- Interrupção de comparação com B
 - timer0ActivateCompareBInterrupt()
 - timer1ActivateCompareBInterrupt()
 - timer2ActivateCompareBInterrupt()
 - timer0DeactivateCompareBInterrupt()
 - timer1DeactivateCompareBInterrupt()
 - timer2DeactivateCompareBInterrupt()
 - timer0ClearCompareBInterruptRequest()
 - timer1ClearCompareBInterruptRequest()
 - timer2ClearCompareBInterruptRequest()



BIBLIOTECAS LS / R2R

- Interrupção de captura de eventos
 - timer1ActivateInputCaptureInterrupt()
 - timer1DeactivateInputCaptureInterrupt()
 - timer1ClearInputCaptureInterruptRequest()
- Forçar comparação
 - timer0ForceCompareA()
 - timer1ForceCompareA()
 - timer2ForceCompareA()
 - timer0ForceCompareB()
 - timer1ForceCompareB()
 - timer2ForceCompareB()



BIBLIOTECAS LS / R2R

- Valor do contador
 - timer0SetCounterValue(value)
 - timer1SetCounterValue(value)
 - timer2SetCounterValue(value)
 - timer0GetCounterValue()
 - timer1GetCounterValue()
 - timer2GetCounterValue()
- Valor do comparador A
 - timer0SetCompareAValue(value)
 - timer1SetCompareAValue(value)
 - timer2SetCompareAValue(value)
 - timer0GetCompareAValue()
 - timer1GetCompareAValue()
 - timer2GetCompareAValue()



BIBLIOTECAS LS / R2R

- Valor do comparador B
 - timer0SetCompareBValue(value)
 - timer1SetCompareBValue(value)
 - timer2SetCompareBValue(value)
 - timer0GetCompareBValue()
 - timer1GetCompareBValue()
 - timer2GetCompareBValue()
- Valor da captura de eventos
 - timer1SetInputCaptureValue(value)
 - timer1GetInputCaptureValue()

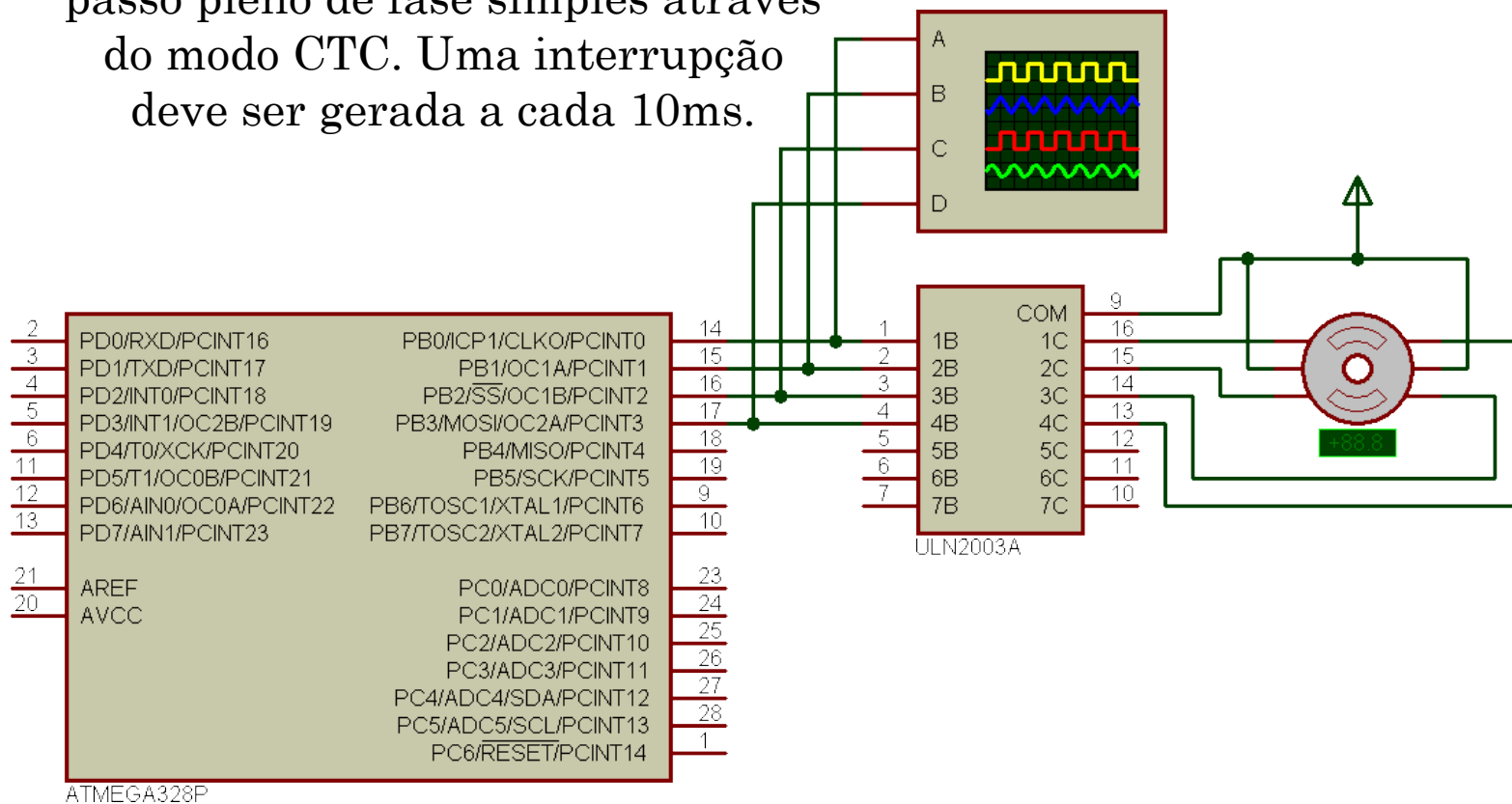


EXERCÍCIO



EXERCÍCIO 01

Acionar um motor de passos em passo pleno de fase simples através do modo CTC. Uma interrupção deve ser gerada a cada 10ms.



LEITURA RECOMENDADA



LEITURA RECOMENDADA

- Leitura obrigatória:
 - LIMA, VILLAÇA – Cap 9;
- Leitura recomendada;
 - Datasheet – Cap 15, 16, 17 e 18.





MCO60408

MICROCONTROLADORES

Aula 08 – Temporizadores/Contadores

Leandro Schwarz