

# 5197 - Sistema Digitais

Bacharelado de Informática

UEM – DIN - Prof. Elvio

2017

# Roteiro

- ATmega328 (Interrupções)
- ATmega328 (Interrupções Externas)
- ATmega328 (Portas de E/S)
- ATmega328 (Temporizador/Contador 0)
- ATmega328 (Temporizador/Contador 1)
- ATmega328 (Temporizador/Contador 2)

# ATmega328 (Interrupções)

<b>VectorNo.</b>	<b>Program Address<sup>(2)</sup></b>	<b>Source</b>	<b>Interrupt Definition</b>
1	0x0000 <sup>(1)</sup>	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Coutner1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow

# ATmega328 (Interrupções)

VectorNo.	Program Address <sup>(2)</sup>	Source	Interrupt Definition
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready

# ATmega328 (Interrupções)

Endereço	Código	Comentários
	.ORG 0x00	;diretiva do assembly para gravar ;o código abaixo no end. 0x00
0x00	RJMP INICIO	;desvia para o início do programa
0x01	RJMP EXT_INT0	;interrup. externa 0
0x02	RJMP EXT_INT1	;interrup. externa 1
0x03	RJMP PCINT0	;interrup. 0 por mudança de pino
0x04	RJMP PCINT1	;interrup. 1 por mudança de pino
0x05	RJMP PCINT2	;interrup. 2 por mudança de pino
0x06	RJMP WDT	;estouro do temporizador Watchdog
0x07	RJMP TIM2_COMPA	;igualdade de comparação A do TC2
0x08	RJMP TIM2_COMPB	;igualdade de comparação B do TC2
0x09	RJMP TIM2_OVF	;estouro do TC2
0x0A	RJMP TIM1_CAPT	;evento de captura do TC1
0x0B	RJMP TIM1_COMPA	;igualdade de comparação A do TC1
0x0C	RJMP TIM1_COMPB	;igualdade de comparação B do TC1
0x0D	RJMP TIM1_OVF	;estouro do TC1
0x0E	RJMP TIM0_COMPA	;igualdade de comparação A do TC0
0x0F	RJMP TIM0_COMPB	;igualdade de comparação B do TC0
0x10	RJMP TIM0_OVF	;estouro do TC0
...		
	INICIO:           ...	;código para o início, programa principal
	EXT_INT0:       ...	;código para a interrup. externa 0
	...	

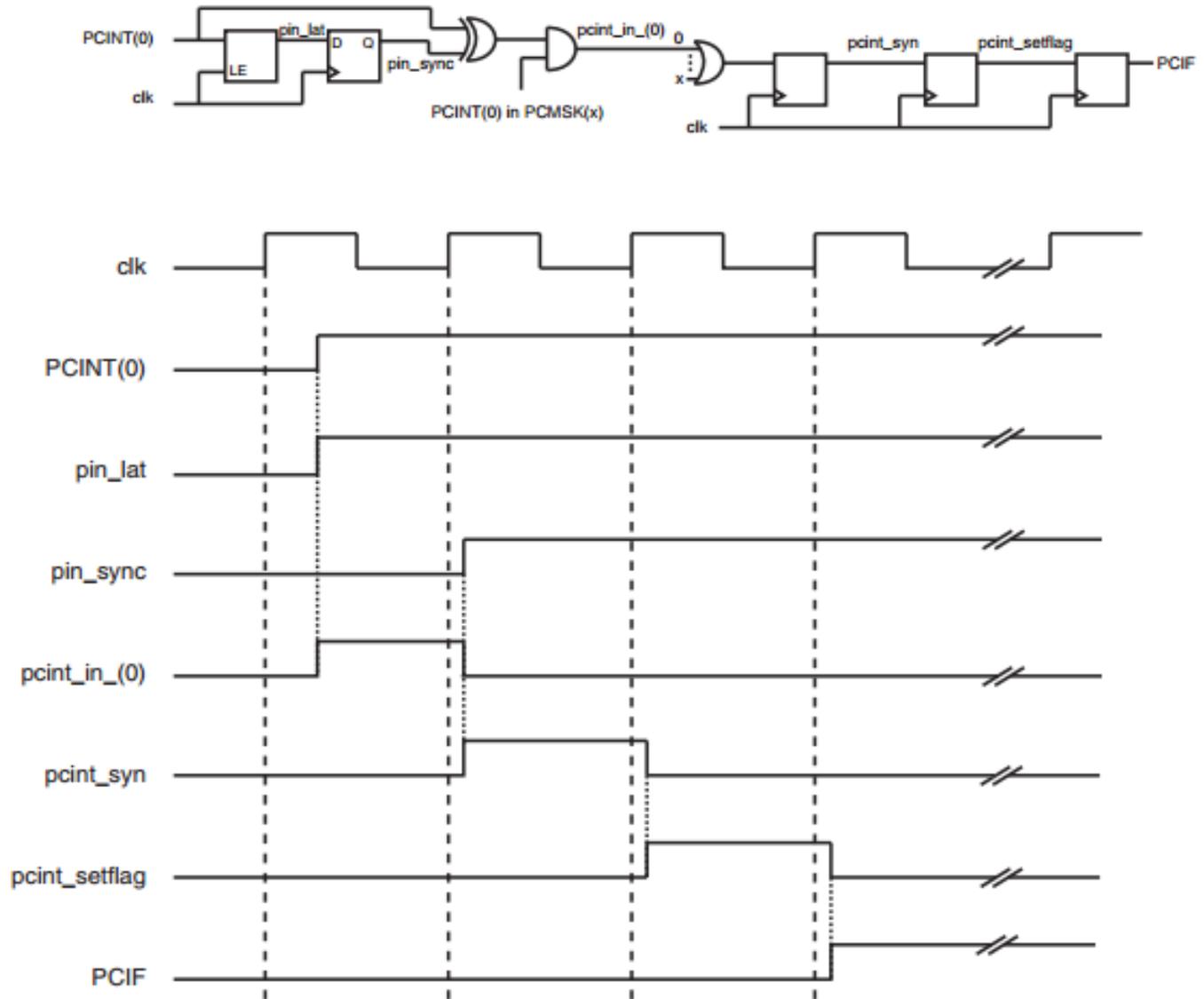
# ATmega328 (Interrupções)

Código (função)

```
ISR(INT0_vect)           { //interrupção externa 0 }
ISR(INT1_vect)           { //interrupção externa 1 }
ISR(PCINT0_vect)         { //interrupção 0 por mudança de pino }
ISR(PCINT1_vect)         { //interrupção 1 por mudança de pino }
ISR(PCINT2_vect)         { //interrupção 2 por mudança de pino }
ISR(WDT_vect)            { //estouro do temporizador Watchdog }
ISR(TIMER2_COMPA_vect)   { //igualdade de comparação A do TC2 }
ISR(TIMER2_COMPB_vect)   { //igualdade de comparação B do TC2 }
ISR(TIMER2_OVF_vect)     { //estouro do TC2 }
ISR(TIMER1_CAPT_vect)    { //evento de captura do TC1 }
ISR(TIMER1_COMPA_vect)   { //igualdade de comparação A do TC1 }
ISR(TIMER1_COMPB_vect)   { //igualdade de comparação B do TC1 }
ISR(TIMER1_OVF_vect)     { //estouro do TC1 }
ISR(TIMER0_COMPA_vect)   { //igualdade de comparação A do TC0 }
ISR(TIMER0_COMPB_vect)   { //igualdade de comparação B do TC0 }
ISR(TIMER0_OVF_vect)     { //estouro do TC0 }
ISR(SPI_STC_vect)        { //transferência serial completa - SPI }
ISR(USART_RX_vect)       { //USART, recepção completa }
ISR(USART_UDRE_vect)     { //USART, limpeza do registrador de dados }
ISR(USART_TX_vect)       { //USART, transmissão completa }
ISR(ADC_vect)            { //conversão do ADC completa }
ISR(EE_READY_vect)       { //EEPROM pronta }
ISR(ANALOG_COMP_vect)    { //comparador analógico }
ISR(TWI_vect)            { //interface serial TWI }
ISR(SPM_READY_vect)      { //armazenagem na memória de programa pronta }
```

# ATmega328 (Interrupções Externas)

- Timing



# ATmega328 (Interrupções Externas)

- Reg. External Interrupt Control A (EICRA)

Bit	7	6	5	4	3	2	1	0	
(0x69)	-	-	-	-	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

– ISC<sub>x</sub>[1-0]: Interrupt Sense Control x

ISC <sub>x</sub> 1	ISC <sub>x</sub> 0	Description
0	0	The low level of INT <sub>x</sub> generates an interrupt request.
0	1	Any logical change on INT <sub>x</sub> generates an interrupt request.
1	0	The falling edge of INT <sub>x</sub> generates an interrupt request.
1	1	The rising edge of INT <sub>x</sub> generates an interrupt request.

# ATmega328 (Interrupções Externas)

- Reg. External Interrupt Mask (EIMSK )

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	-	-	-	-	-	-	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

– INT<sub>x</sub>: External Interrupt Request x Enable

- Reg. External Interrupt Flag (EIFR)

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	-	-	-	-	-	-	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

– INTF<sub>x</sub>: External Interrupt Flag x: fica em 1 em caso de interrupção pendente



# ATmega328 (Portas de E/S)

- Todas as portas de E/S podem ser configuradas individualmente como entrada ou saída
- Todas as portas operando como entrada podem acrescentar ou não um resistor de *pull-up*
- Portas operando como saída tem capacidade de alimentar um LED, com mesma capacidade de corrente para níveis 0 ou 1
- Cada bit pode ser configurado individualmente utilizando as instruções SBI (Set Bit in I/O Register) e CBI (Clear Bit in I/O Register)

# ATmega328 (Portas de E/S)

- Cada porta tem três registradores:
  - DDxn: seleciona a direção da porta x, pino n (0 para entrada, 1 para saída)
  - PORTxn:
    - Entrada: habilita (com 1 e PUD – Pull up Disable em 0) resistor de pull-up
    - Saída: escreve valor lógico
  - PINxn: lê conteúdo das portas

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

# ATmega328 (Portas de E/S)

## Assembly Code Example<sup>(1)</sup>

```
...
; Define pull-ups and set outputs high
; Define directions for port pins
ldi r16, (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)
ldi r17, (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0)
out PORTB, r16
out DDRB, r17
; Insert nop for synchronization
nop
; Read port pins
in r16, PINB
...
```

### LDI - Load Immediate

Carrega constante de 8 bits em registrador R16 a R31.

## C Code Example

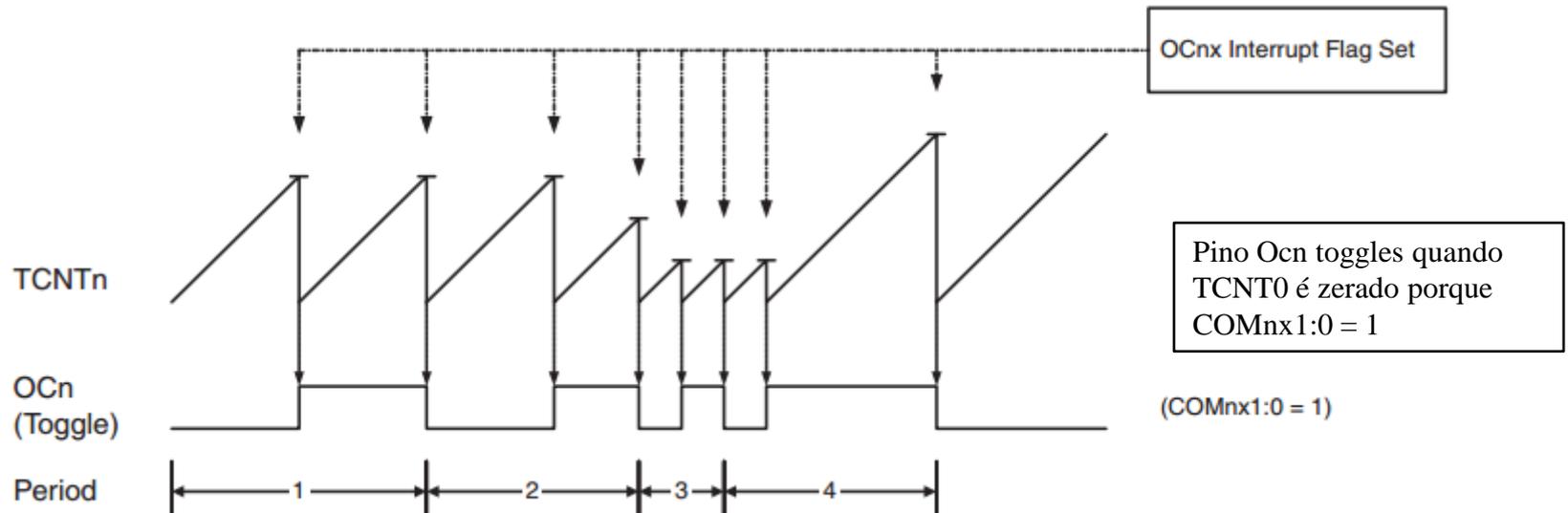
```
unsigned char i;
...
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0);
DDRB = (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
/* Insert nop for synchronization*/
__no_operation();
/* Read port pins */
i = PINB;
...
```

# Temporizador/Contador 0 (TC 0)

- Contador de 8 bits, incrementado com pulsos de relógio interno ou externo
- Modos de operação: normal, CTC (*Clear Timer on Compare*), PWM (*Pulse Width Modulation*) rápida e com fase corrigida
- Modo Normal
  - Modo mais simples, onde TC0 conta de modo crescente
  - A contagem se dá de 0 até 255 voltando a 0, num ciclo contínuo
  - Quando a contagem passa de 255 para 0 o bit sinalizador de estouro (TOV0) é colocado em 1 (gera interrupção se habilitada)
  - O valor da contagem pode ser escrito a qualquer momento no *Timer/Counter Register* (TCNT0)

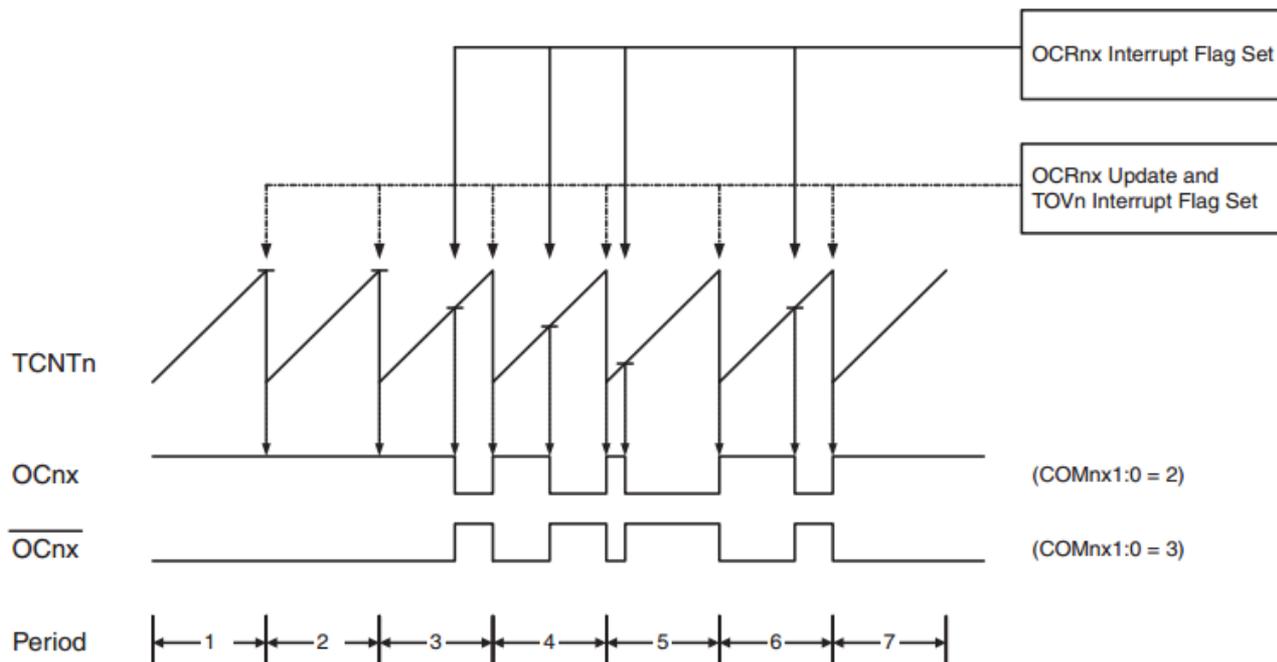
# Temporizador/Contador 0 (TC 0)

- Modo CTC (*Clear Timer on Compare*)
  - Valor do contador (TCNT0) é zerado sempre que a contagem iguala o valor contido em Output Compare Register A (OCR0A)



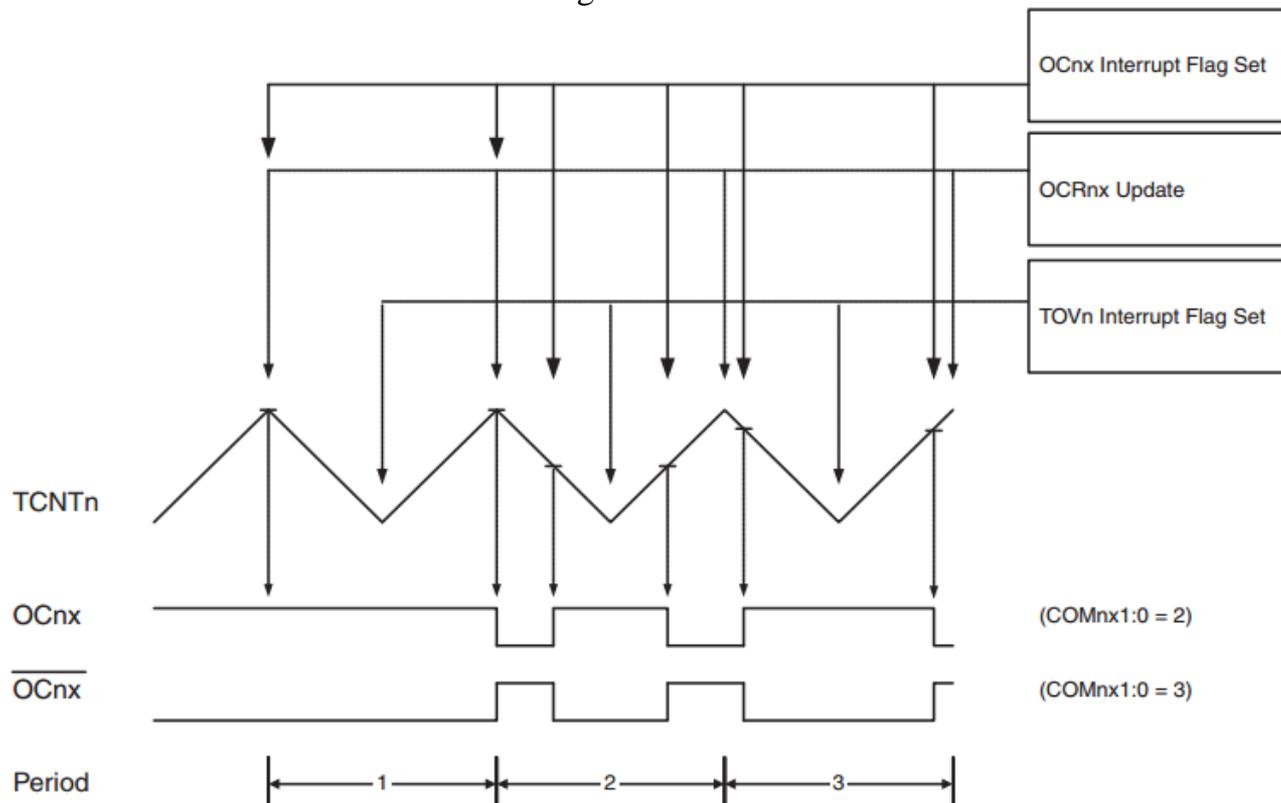
# Temporizador/Contador 0 (TC 0)

- Modo PWM Rápida
  - Opera em rampa única e permite geração de ondas PWM de alta frequência
  - Contador conta do valor mínimo até o máximo e recomeça do mínimo
  - Valor máximo é 0xFF (se WGM2:0 = 3) ou OCR0A (se WGM2:0 = 7)
    - Se valor máximo é OCR0A não gera sinal PWM



# Temporizador/Contador 0 (TC 0)

- Modo PWM com Fase Corrigida
  - Opera em rampa dupla
  - Contador conta do valor mínimo até o máximo e do máximo de volta ao mínimo
  - Valor máximo é 0xFF (se WGM2:0 = 1) ou OCR0A (se WGM2:0 = 5)
    - Se valor máximo é OCR0A não gera sinal PWM





# Temporizador/Contador 0 (TC 0)

- Regs. Timer/Counter Ctrl A e B (TCCR0[A,B])

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on <sup>(1)(2)</sup>
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

# Temporizador/Contador 0 (TC 0)

- Regs. Timer/Counter Ctrl A e B (TCCR0[A,B])
  - COM0A[1-0]: Compare Match Output A Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

# Temporizador/Contador 0 (TC 0)

- Regs. Timer/Counter Ctrl A e B (TCCR0[A,B])
  - COM0A[1-0]: Compare Match Output A Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM, (inverting mode).

# Temporizador/Contador 0 (TC 0)

- Regs. Timer/Counter Ctrl A e B (TCCR0[A,B])
  - COM0A[1-0]: Compare Match Output A Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match when up-counting. Set OC0A on Compare Match when down-counting.
1	1	Set OC0A on Compare Match when up-counting. Clear OC0A on Compare Match when down-counting.

# Temporizador/Contador 0 (TC 0)

- Regs. Timer/Counter Ctrl A e B (TCCR0[A,B])

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$\text{clk}_{I/O}$ /(No prescaling)
0	1	0	$\text{clk}_{I/O}/8$ (From prescaler)
0	1	1	$\text{clk}_{I/O}/64$ (From prescaler)
1	0	0	$\text{clk}_{I/O}/256$ (From prescaler)
1	0	1	$\text{clk}_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

# Temporizador/Contador 0 (TC 0)

- Regs. Timer/Counter Ctrl A e B (TCCR0[A,B])
  - FOC0A: Force Output Compare A
    - Para modo não PWM
    - Força comparação no módulo gerador de onda, com efeitos determinado por COM0A1:0



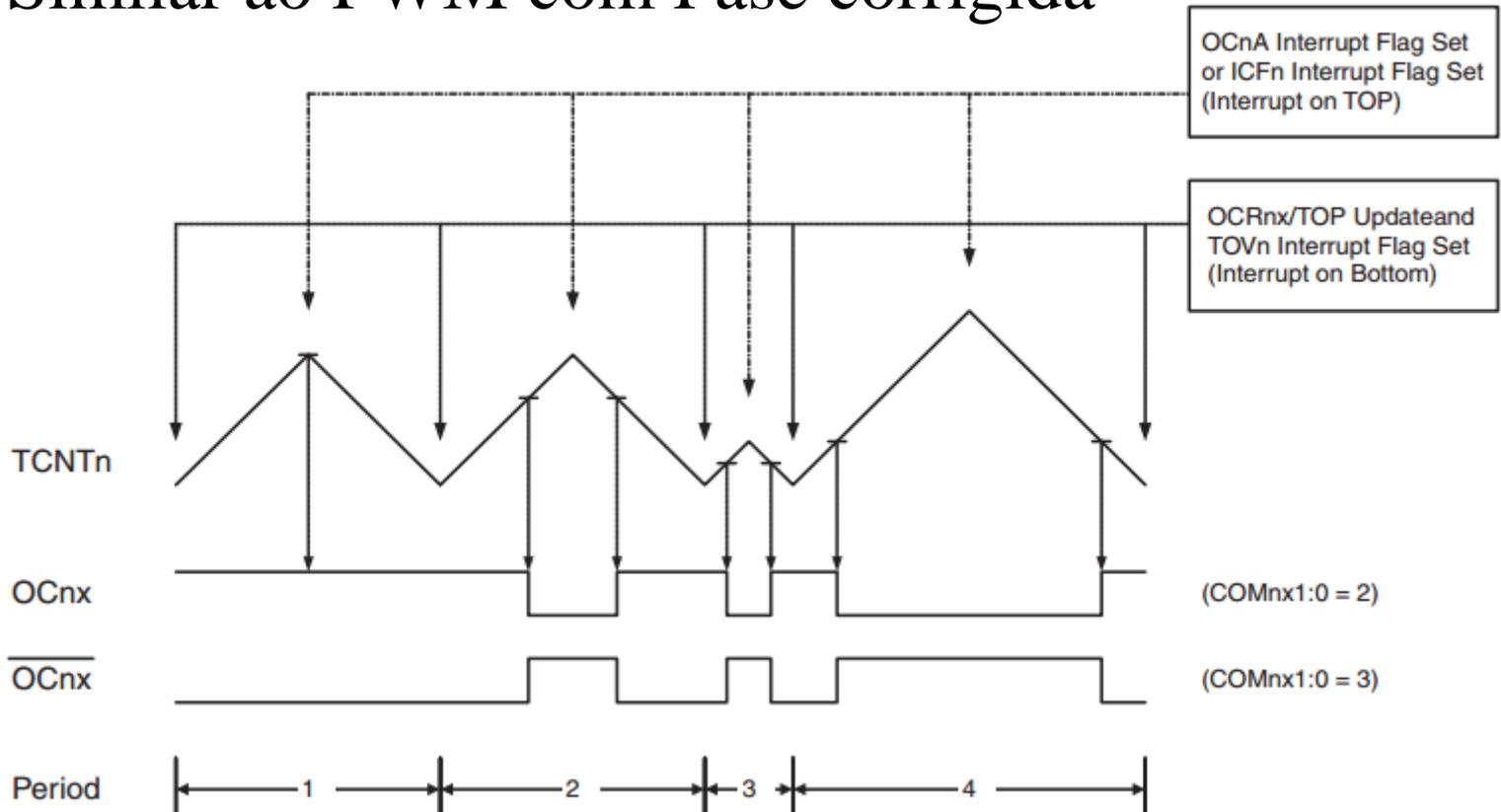


# Temporizador/Contador 1 (TC 1)

- Contador de 16 bits, incrementado com pulsos de relógio interno ou externo, selecionado pelos bits *Clock Select* (CS0[2-0]) do *Timer/Counter Control Register* (TCCR1A/B)
- Modos de operação
  - Similares ao TC0: normal, CTC (*Clear Timer on Compare*), PWM (*Pulse Width Modulation*) rápida e com fase corrigida
  - Novo: PWM com fase e frequência corrigidas

# Temporizador/Contador 1 (TC 1)

- Modo PWM com Frequência e Fase Corrigidas
  - Similar ao PWM com Fase corrigida



# Temporizador/Contador 2 (TC 2)

- Contador de 8 bits, incrementado com pulsos de relógio interno ou externo, com operação similar ao TC 0
- Modos de operação: normal, CTC (*Clear Timer on Compare*), PWM (*Pulse Width Modulation*) rápida e com fase corrigida
- Permite uso do relógio externo de 32,768 kHz (denominado modo assíncrono)

# Temporizadores no Arduino Uno R3

- Todos os temporizadores são usados pela biblioteca do Arduino (Arduino IDE)
  - Deve-se, portanto, usar cautela ao alterar a configuração dos temporizadores sob o risco da funcionalidade correspondente ser quebrada
- Temporizador 0: usado para funções de temporização, como `delay()`, `millis()` e `micros()`
- Temporizador 1: usado para funções de servo controle (biblioteca `servo`, em particular para o modo PWM)
- Temporizador 2: usado para geração de sons através da função `tone()`