

Manipulando Reset e Interrupção

Jadsonlee da Silva Sá

Jadsonlee.sa@univasf.edu.br

www.univasf.edu.br/~jadsonlee.sa

Reset e Interrupções

- ✓ Há 25 diferentes fontes de interrupção mais o reset.
- ✓ Essas interrupções e o reset possuem um vetor de programa separado no espaço da memória de programa.
- ✓ Cada vetor de programa ocupa duas posições da memória de programa.

Interrupções

Vector No	Program Address ⁽²⁾	Source	Interrupts definition
1	0x0000 ⁽¹⁾	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 0
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/Counter1 Compare Match B
14	0x001A	TIMER1_OVF	Timer/Counter1 Overflow
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 (I ² C)
26	0x0032	SPM_READY	Store Program Memory Ready

O fuse BOOTRST e o bit IVSEL de MCUCR indicam, respectivamente, o endereço do vetor de reset e o de início do vetor de interrupções.

Note:

1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self- Programming"
2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

Vetores de Interrupção - ATmega328P

- ✓ Configuração mais geral para os endereços do reset e vetor de interrupção.

Address	Labels	Code	Comments
0x0000		jmp RESET	; Reset
0x0002		jmp INTO	; IRQ0
0x0004		jmp INT1	; IRQ1
0x0006		jmp PCINT0	; PCINT0
0x0008		jmp PCINT1	; PCINT1
0x000A		jmp PCINT2	; PCINT2
0x000C		jmp WDT	; Watchdog Timeout
0x000E		jmp TIM2_COMPA	; Timer2 CompareA
0x0010		jmp TIM2_COMPB	; Timer2 CompareB
0x0012		jmp TIM2_OVF	; Timer2 Overflow
0x0014		jmp TIM1_CAPT	; Timer1 Capture
0x0016		jmp TIM1_COMPA	; Timer1 CompareA
0x0018		jmp TIM1_COMPB	; Timer1 CompareB
0x001A		jmp TIM1_OVF	; Timer1 Overflow
0x001C		jmp TIM0_COMPA	; Timer0 CompareA
0x001E		jmp TIM0_COMPB	; Timer0 CompareB
0x0020		jmp TIM0_OVF	; Timer0 Overflow
0x0022		jmp SPI_STC	; SPI Transfer Complete
0x0024		jmp USART_RXC	; USART RX Complete
0x0026		jmp USART_UDRE	; USART UDR Empty
0x0028		jmp USART_TXC	; USART TX Complete
0x002A		jmp ADC	; ADC Conversion Complete
0x002C		jmp EE_RDY	; EEPROM Ready
0x002E		jmp ANA_COMP	; Analog Comparator
0x0030		jmp TWI	; 2-wire Serial
0x0032		jmp SPM_RDY	; SPM Ready
;			
0x0034	RESET:	ldi r16,high(RAMEND)	; Main program start
0x0035		out SPH,r16	; Set Stack Pointer to top of RAM
0x0036		ldi r16,low(RAMEND)	
0x0037		out SPL,r16	
0x0038		sei	; Enable interrupts
0x0039		<instr> xxx	
...	...	...	...

Interrupções

- ✓ Todas as interrupções possuem bits separados que devem ser setados '1' juntamente com o bit I de STATUS, que habilita a interrupção global, de modo a habilitar a interrupção.

Name: SREG

Offset: 0x5F

Reset: 0x00

Property: When addressing as I/O Register: address offset is 0x3F

Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – I: Global Interrupt Enable

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

Interrupções

- ✓ Possuem níveis de prioridade determinadas: quanto menor o endereço, maior é a prioridade.
- ✓ Reset possui a maior prioridade, em seguida, INTO (External Interrupt Request 0), e assim por diante.

Interrupções

- ✓ Quando uma interrupção ocorre, o bit I é zerado e todas as interrupções são desabilitadas.
 - O software do usuário pode fazer $I=1$ para habilitar interrupções aninhadas. Então, todas as interrupções habilitadas podem interromper a rotina da interrupção atual.
- ✓ O bit I é automaticamente setado quando a instrução RETI for executada.

Interrupções

- ✓ Quando o AVR sai de uma interrupção, ele sempre retorna ao programa principal e executa uma ou mais instruções antes de uma interrupção pendente ser servida.
- ✓ STATUS não é automaticamente armazenado quando uma rotina de interrupção for servida, e nem restaurado quando retorna da interrupção. Isso deve ser feito por software.

Assembly Code Example

```
in r16, SREG ; store SREG value  
cli ; disable interrupts during timed sequence  
sbi EECR, EEMPE ; start EEPROM write  
sbi EECR, EEPE  
out SREG, r16 ; restore SREG value (I-bit)
```


Interrupções

- ✓ **Instrução CLI** - As interrupções são imediatamente desabilitadas quando CLI (desabilita interrupções) for executada. Nenhuma interrupção será executada após CLI, mesmo se ela ocorrer simultaneamente com a execução de CLI.
- ✓ **Instrução SEI** - Após a execução de SEI (habilita interrupções), a instrução seguinte será executada antes de qualquer interrupção pendente.

Assembly Code Example

```
sei ; set Global Interrupt Enable  
sleep ; enter sleep, waiting for interrupt  
; note: will enter sleep before any pending interrupt(s)
```

Interrupções

✓ Exemplo de código - Timer 0 no modo normal.

```
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 16000000UL
```

```
ISR(TIMERO0_OVF_vect)
```

```
{
    PORTB^=(1<<PB5);
}
```

 interrupt.h

 inttypes.h

 io.h

 iom328p.h

```
#define TIMERO0_OVF_vect_num 16
#define TIMERO0_OVF_vect _VECTOR(16) /* Timer/Counter0 Overflow */
```

```
int main(void)
```

```
{
```

```
    DDRB = 0x20; //
```

```
    TIFR0 = 0x07; ←
```

```
    TIMSK0 = 0x01; ←
```

```
    sei(); ←
```

```
    PORTB = 0x00; /
```

```
    TCNT0 = 0x00; /
```

```
    TCCR0A = 0x00;
```

```
    TCCR0B = 0x05;
```

```
    while (1)
```

```
    {
```

```
    }
```

Timer/Counter 0

Jadsonlee da Silva Sá

Jadsonlee.sa@univasf.edu.br

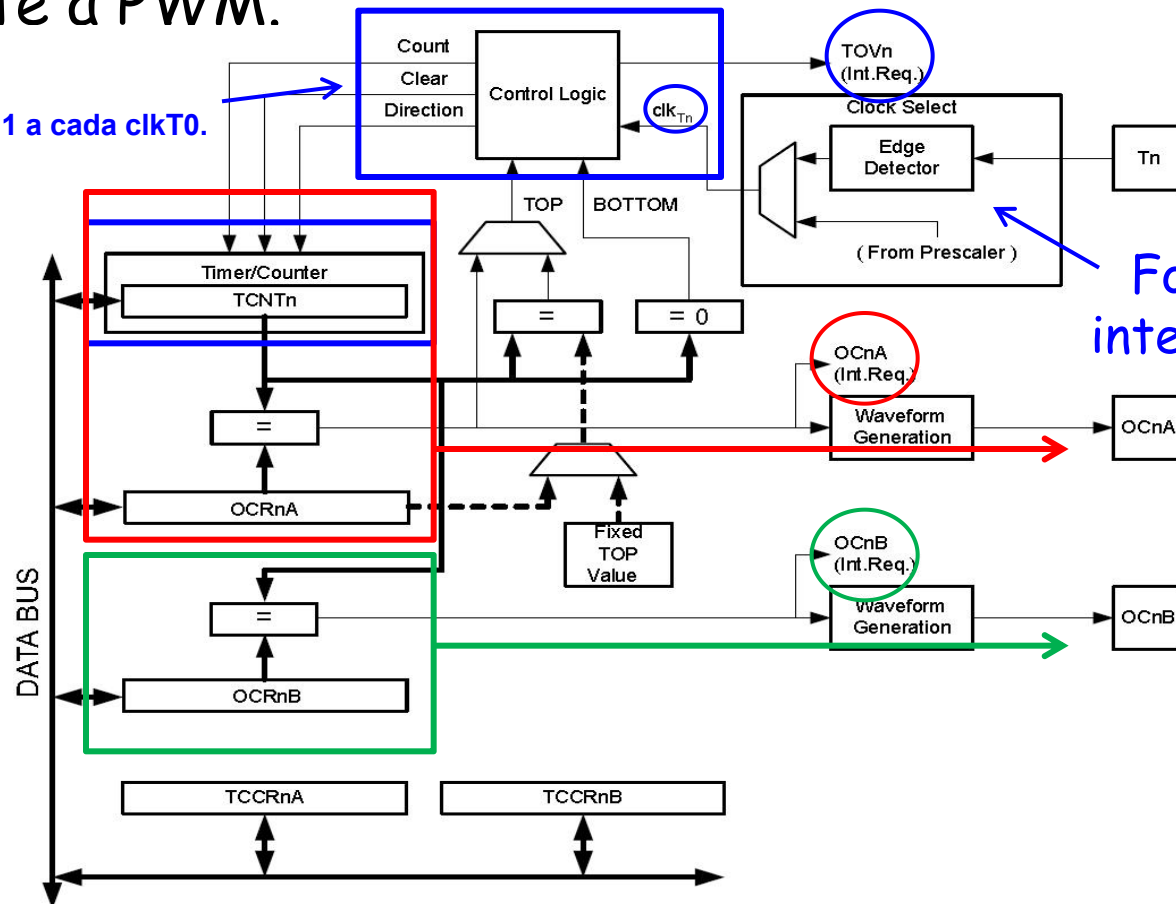
www.univasf.edu.br/~jadsonlee.sa

Introdução

- ✓ O Timer/Counter 0 (TC0) é um módulo temporizador/contador de 8 bits, que possui duas unidades de saída de comparação independentes e suporte a PWM.

Direction = inc. ou dec.
Count = inc. ou dec. por 1 a cada clkT0.
Clear = 0x00.

BOTTOM = 0x00
MAX = 0xFF
TOP = MAX fixado ou
o valor em OCR0A



Fonte de clock
interna ou externa

Introdução

✓ Pinos de E/S relacionados:

- T0 (PD4 ou pino 6), OCOA (PD6 ou pino 12) e OCOB (PD5 ou pino 11);

✓ Registradores relacionados ao TCO:

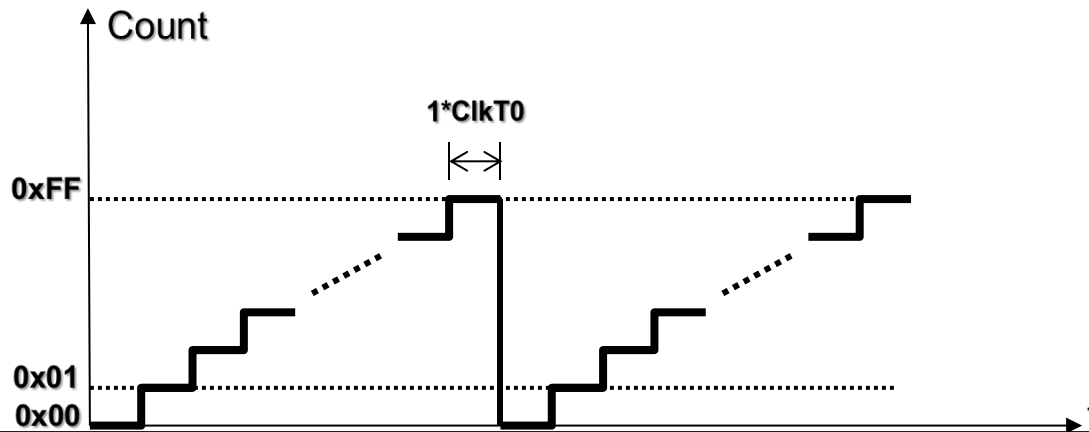
- TCCR0A - Registrador de controle A;
- TCCR0B - Registrador de controle B;
- TIMSK0 - Registrador de máscara de interrupção;
- TIFR0 - Registrador de flag de interrupção;
- TCNT0 - Registrador de contagem;
- OCR0A - Registrador de comparação de saída A;
- OCR0B - Registrador de comparação de saída B;

Modos de Operação - TCO

- ✓ Determina o comportamento do TCO e dos pinos de comparação de saída.
- ✓ Há 4 modos:
 - Normal;
 - CTC (Clear Time on Compare);
 - Fast PWM (Pulse Width Modulation);
 - Phase Correct PWM.

Modos de Operação Normal

- ✓ A contagem é sempre incrementada por 1 e não é possível zerar a contagem.
- ✓ A contagem é reiniciada (0x00 - BOTTOM) quando atinge 0xFF (TOP).
- ✓ O flag de overflow TOV0 (de TIFR0) vai para '1' quando TCNT0 for zerado.



Modos de Operação Normal

✓ Configuração dos registradores.

Name: TCCR0A

Offset: 0x44

Reset: 0x00

Property: When addressing as I/O Register: address offset is 0x24

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

- Bits 7 e 6 - *COM0A1* e *COM0A0*.

Table 19-3. Compare Output Mode, non-PWM

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.

Modos de Operação Normal

✓ Configuração dos registradores.

Name: TCCR0A

Offset: 0x44

Reset: 0x00

Property: When addressing as I/O Register: address offset is 0x24

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

- Bits 5 e 4 - *COM0B1* e *COM0B0*.

Table 19-6. Compare Output Mode, non-PWM

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match.
1	0	Clear OC0B on Compare Match.
1	1	Set OC0B on Compare Match.

Modos de Operação Normal

✓ Configuração dos registradores.

Name: TCCR0A

Offset: 0x44

Reset: 0x00

Property: When addressing as I/O Register: address offset is 0x24

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

- Bits 1 e 0 - WGM01 e WGM00.

Table 19-9. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCR0x at	TOV Flag Set on ⁽¹⁾⁽²⁾
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

Note:

1. MAX = 0xFF
2. BOTTOM = 0x00

Modos de Operação Normal

✓ Configuração dos registradores.

Name: TCCR0B

Offset: 0x45

Reset: 0x00

Property: When addressing as I/O Register, address offset is 0x25

Bit	7	6	5	4	3	2	1	0
	FOC0A	FOC0B			WGM02	CS0[2:0]		
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

- Bits 7 e 6 - Não utilizados nesse modo.
- Bit 3 - WGM02 = 0 - modo normal.

Modos de Operação Normal

✓ Configuração dos registradores.

Name: TCCR0B

Offset: 0x45

Reset: 0x00

Property: When addressing as I/O Register, address offset is 0x25

Bit	7	6	5	4	3	2	1	0
	FOC0A	FOC0B			WGM02	CS0[2:0]		
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

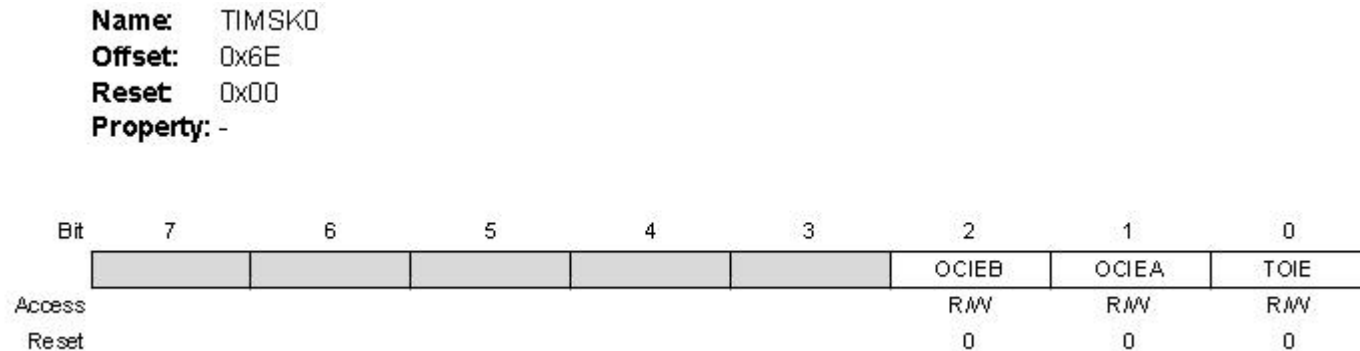
- Bits 2 a 0 - CS0[2:0] - Seleciona o clock e prescaler.

Table 19-10. Clock Select Bit Description

CA02	CA01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{\text{IO}}/1$ (No prescaling)
0	1	0	$\text{clk}_{\text{IO}}/8$ (From prescaler)
0	1	1	$\text{clk}_{\text{IO}}/64$ (From prescaler)
1	0	0	$\text{clk}_{\text{IO}}/256$ (From prescaler)
1	0	1	$\text{clk}_{\text{IO}}/1024$ (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.

Modos de Operação Normal

✓ Configuração dos registradores.



- Bits 2 e 1 - Não utilizados nesse modo.
- Bit 0 - TOIE - Habilita a interrupção no overflow de TCO.
 - Uma interrupção ocorrerá no overflow ($TOV0 = 1$), se $TOIE = 1$.

Modos de Operação Normal

✓ Configuração dos registradores.

Name: TIFR0

Offset: 0x35

Reset: 0x00

Property: When addressing as I/O Register: address offset is 0x15

Bit	7	6	5	4	3	2	1	0
						OCFB	OCFA	TOV
Access						R/W	R/W	R/W
Reset						0	0	0

- Bits 2 e 1 - Não utilizados nesse modo.
- Bit 0 - TOV - Indica um overflow em TCO (TOV=1).
 - Uma interrupção ocorrerá, se $TOIE = I = 1$.
 - Para zerar TOV, deve-se fazer $TOV = 1$.

Modos de Operação Normal

```
#include <avr/io.h>
#define F_CPU 16000000UL

int main(void)
{
    //No modo normal, o TC0 conta até 0xFF e reinicia a contagem. No overflow, TOV = 1;
    //clkSYS = 1/16MHz = 62,5 ns = clkIO = clkT0
    //Um overflow equivale a 256 * 62,5 ns ≈ 16 us - Sem prescaling
    DDRB = 0x20; //PB5 como saída.

    TIFR0 = 0x07; //Zera os flags OCFB, OCFA e TOV.
    PORTB = 0x00; //Zera PB5.
    TCNT0 = 0x00; //Zera timer 0.

    TCCR0A = 0x00; //COM0A1 = COM0A0 = 0 (operação da porta normal), COM0B1 = COM0B0 = 0 (operação da porta normal).
    //WGM01 = WGM00 = 0 = WGM02(TCCR0B) (modo normal, TOP = 0xFF, atualização de OCR0x imediata e flag
    //TOV setado em MAX).
    TCCR0B = 0x01; //FOC0A = FOC0B = 0, WGM02 = 0, CS0[2:0] = 001 (clkIO/1 - sem prescaling).
    //clkIO/1 = 16 us; clkIO/8 = 128 us; clkIO/64 = 1024 us; clkIO/256 = 4096 us; clkIO/1024 = 16384 us

    while (1)
    {
        while((TIFR0 & 0x01) != 0x01){} //Verifica se houve overflow, se não houve, não faz nada.
        TIFR0 = 0x01; //Escrevo 1 no bit 0 (TOV) para zerá-lo.
        PORTB^=(1<<PB5);
    }
}
```

Modos de Operação Normal

✓ Exercícios.

1. Execute o código TCO/Normal sem interrupção.
 - Verifique no osciloscópio (veja a forma de onda no pino PB5) se os atrasos correspondem ao calculado teoricamente.
 - Modifique o prescaler (8, 64, 256 e 1024) e veja o que ocorre com as formas de onda.
2. Modifique o código de modo que o LED acenda/apague a cada 800 ms.

Modos de Operação Normal

✓ Exemplo - TCO/Normal com interrupção.

```
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 16000000UL

ISR(TIMER0_OVF_vect)
{
    PORTB^=(1<<PB5);
}

int main(void)
{
    DDRB = 0x20; //PB5 como saída.

    TIFR0 = 0x07; //Zera os flags OCFB, OCFA e TOV.
    TIMSK0 = 0x01; //Habilita interrupção timer 0.
    sei();

    PORTB = 0x00; //Zera PB5.
    TCNT0 = 0x00; //Zera timer 0.

    TCCR0A = 0x00; //COM0A1 = COM0A0 = 0 (operação da porta normal), COM0B1 = COM0B0 = 0 (operação da porta normal).
    //WGM01 = WGM00 = 0 = WGM02(TCCR0B) (modo normal, TOP = 0xFF, atualização de OCR0x imediata e flag
    //TOV setado em MAX).
    TCCR0B = 0x05; //FOC0A = FOC0B = 0, WGM02 = 0, CS0[2:0] = 001 (clkIO/1 - sem prescaling).
    //clkIO/1 = 16 us; clkIO/8 = 128 us; clkIO/64 = 1024 us; clkIO/256 = 4096 us; clkIO/1024 = 16384 us

    while (1)
    {
    }
```

Modos de Operação Normal

✓ Exercícios.

1. Execute o código TCO/Normal com interrupção.
 - Verifique no osciloscópio (veja a forma de onda no pino PB5) se os atrasos correspondem ao calculado teoricamente.
 - Modifique o prescaler (8, 64, 256 e 1024) e veja o que ocorre com as formas de onda.
2. Modifique o código de modo que o LED acenda/apague a cada 800 ms.