

9.5 TEMPORIZADOR/CONTADOR 1

O TC1 é um contador de 16 bits que permite grande precisão na geração de formas de onda, temporizações e medição de largura de pulsos. Interrupções podem ser habilitadas para os eventos de estouro ou igualdade de comparação.

A contagem é feita no par de registradores TCNT1H e TCNT1L (contador TCNT1). O TC1 é incrementado via *clock* interno, com ou sem *prescaler*, ou por *clock* externo, ligado ao pino T1. Possuindo dois registradores de controle: TCCR1A e TCCR1B. Os registradores de comparação de saída OCR1A e OCR1B são constantemente comparados com o valor de contagem. O resultado da comparação pode ser usado para gerar sinais PWMs ou com frequência variável nos pinos de saída de comparação (OC1A e OC1B). O valor máximo de contagem (TOP) para o TC1 pode ser definido em alguns modos de operação por OCR1A, ICR1 ou por um conjunto fixo de valores. O registrador de captura de entrada pode capturar o valor do contador quando ocorrer um evento externo no pino ICP1 ou nos pinos do comparador analógico.

MODO NORMAL

É o modo mais simples de operação do contador. Neste modo, a contagem é crescente e contínua, o contador conta de 0 até 65535 (0xFFFF) e volta a zero, sinalizando um estouro e solicitando um pedido de interrupção (se habilitada). O registrador de contagem TCNT1 pode ser escrito ou lido a qualquer momento pelo programa e tem prioridade sobre a contagem ou a limpeza do contador.

MODO DE CAPTURA DE ENTRADA

O modo captura de entrada permite medir o período de um sinal digital externo. Para isso, deve-se ler o valor do TC1 quando ocorrer a interrupção; após duas interrupções, é possível calcular o tempo decorrido entre elas. O

programa deve lidar com os estouros do TC1, caso ocorram. No modo de captura de entrada, escolhe-se qual o tipo de borda do sinal gerará a interrupção (descida ou subida). Para medir o tempo em que um sinal permanece em nível alto ou baixo, após cada captura, é preciso alterar o tipo de borda para a captura.

MODOS CTC

No modo CTC (*Clear Timer on Compare*) os registradores OCR1A ou ICR1 são utilizados para mudar a resolução do contador (valor de topo). Neste modo, o contador é limpo (zerado) quando o valor do TCNT1 é igual a OCR1A ou igual a ICR1. Este modo permite grande controle na comparação para estabelecer a frequência de saída e também simplifica a operação de contagem de eventos externos. Um exemplo de diagrama temporal para o modo CTC é apresentado na fig. 9.9; os pinos OC1A e OC1B foram habilitados.

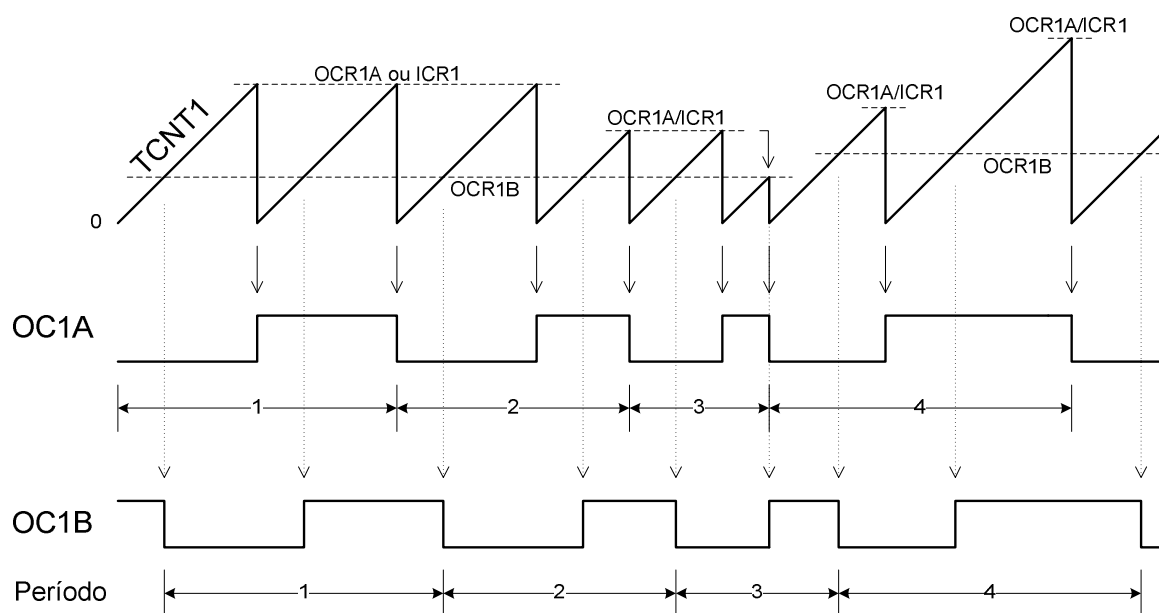


Fig. 9.9 – Diagrama de tempo para o modo CTC.

Para gerar uma onda de saída no modo CTC, os pinos OC1A e OC1B devem ser ajustados para trocar de nível a cada igualdade de comparação. Os valores de OC1A e OC1B não serão visíveis nos seus respectivos pinos, a

não ser que estejam configurados como saída. A máxima frequência que a forma de onda pode alcançar é a metade da frequência de trabalho da CPU. Essa frequência é definida por:

$$f_{OC1A} = \frac{f_{osc}}{2N(1+TOP)} \quad [\text{Hz}] \quad (9.7)$$

onde f_{osc} é a frequência de operação do microcontrolador, TOP tem um valor entre 0 e 65535, e N é o fator do *prescaler* (1, 8, 64, 256 ou 1024). Para calcular o valor TOP em função da frequência desejada, basta isolá-lo na equação acima, o que resulta em:

$$TOP = \frac{f_{osc}}{2N \cdot f_{OC1A}} - 1 \quad (9.8)$$

MODO PWM RÁPIDO

O modo PWM rápido permite a geração de um sinal PWM de alta frequência. O contador conta de zero até o valor máximo e volta a zero. No modo de comparação com saída não-invertida, o pino OC1A é zerado na igualdade entre TCNT1 e OCR1A e colocado em 1 no valor mínimo do contador. No modo de comparação com saída invertida, o processo é inverso: OC1A é ativo na igualdade e zerado no valor mínimo. A comparação para o pino OC1B é feita com OCR1B. O valor de contagem para o TC1 pode ser definido pelos valores fixos 0x0FF (255), 0x1FF (511) ou 0x3FF (1023), por OCR1A ou ICR1; caso OCR1A seja empregado, o pino OC1A não pode gerar uma sinal PWM, apenas uma onda quadrada.

A resolução para o PWM rápido pode ser fixada em 8, 9, 10 bits (0x0FF, 0x1FF ou 0x3FF) ou definida pelos registradores OCR1A ou ICR1. A resolução mínima é de 2 bits (OCR1A ou ICR1 = 3) e a máxima é de 16 bits (ICR1 ou OCR1A = 0xFFFF). A resolução do PWM é calculada com:

$$R_{PWM_Rápido} = \frac{\log(TOP + 1)}{\log(2)} \quad [\text{bits}] \quad (9.9)$$

Neste modo, o contador é incrementado até encontrar um dos valores fixos 0x0FF, 0x1FF ou 0x3FF, o valor de OCR1A ou o valor de ICR1. Um diagrama exemplo para o PWM rápido com OC1A e OC1B como saídas não invertidas é apresentado na fig. 9.10.

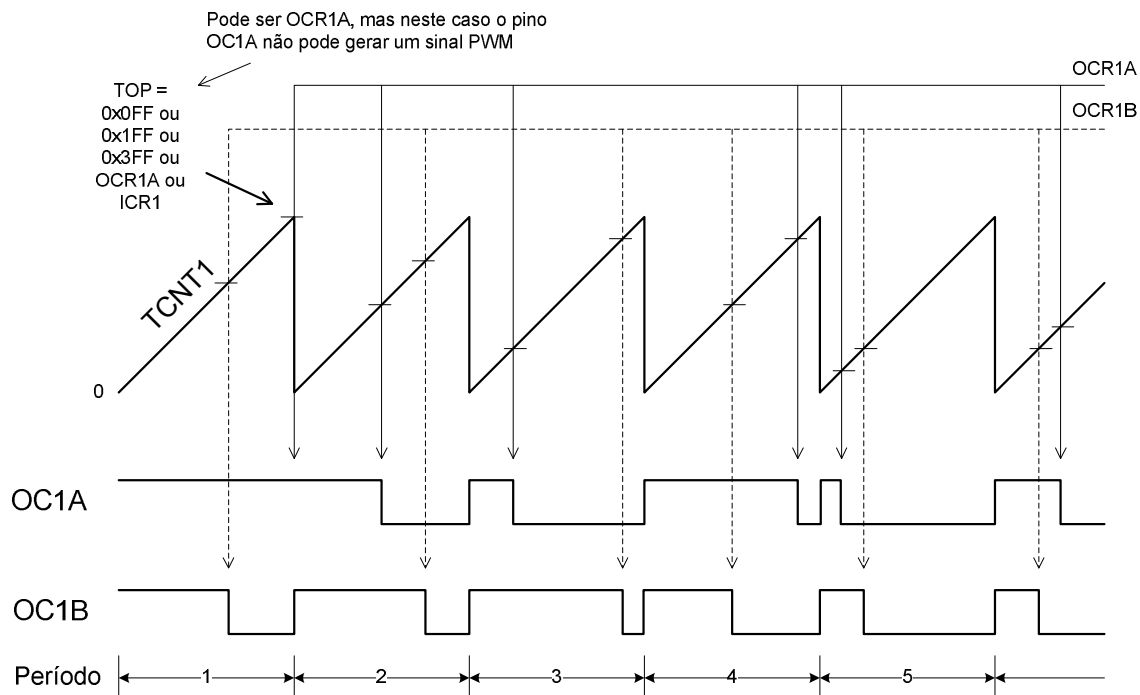


Fig. 9.10 – Diagrama temporal para o modo PWM rápido.

A frequência de saída do PWM rápido é calculada com:

$$f_{OC1x_PWM} = \frac{f_{osc}}{N.(1+TOP)} \quad [Hz] \quad (9.10)$$

onde f_{osc} é a frequência de operação do microcontrolador e N é o fator do *prescaler* (1, 8, 64, 256 ou 1024). Um sinal com ciclo ativo de 50% pode ser obtido ajustando-se OC1A para mudar de nível lógico a cada igualdade de comparação (onda quadrada). Isso se aplica apenas quando OCR1A é usado para definir o valor TOP. A frequência máxima gerada será a metade da frequência de trabalho da CPU quando OCR1A = 0. Quando se controla o ciclo ativo do sinal PWM, o valor zero para OCR1A produz um pequeno ruído, e o valor máximo (TOP) vai deixar o sinal PWM em 0 ou 1, conforme foi habilitada a saída, invertida ou não.

MODO PWM COM FASE CORRIGIDA

O modo PWM com fase corrigida permite ajustar a fase do PWM, isto é, o início e fim do ciclo ativo do sinal PWM. É baseado na contagem crescente e decrescente do TCNT1, que conta repetidamente de zero até o valor máximo (TOP) e vice-versa (permanece um ciclo de *clock* no valor TOP), o que torna a saída PWM simétrica dentro de um período do PWM. Além disso, o valor de comparação que determina o razão cíclica da saída PWM é armazenado em registradores, sendo atualizado apenas quando o contador atinge o valor TOP, o qual marca o início e o fim de um ciclo PWM. Assim, se o valor TOP não for alterado com o temporizador em operação, o pulso gerado será sempre simétrico em relação ao ponto médio do período (TCNT1 = 0x00), qualquer que seja a razão cíclica.

Para o sinal PWM não invertido, o pino OC1A é zerado na igualdade entre TCNT1 e OCR1A, quando a contagem é crescente, e colocado em 1 quando a contagem é decrescente. No modo de comparação com saída invertida, o processo é o contrário. A comparação para o pino OC1B é feita com o registrador OCR1B. O valor máximo de contagem pode ser definido como 0x0FF, 0x1FF, 0x3FF, pelo valor de OCR1A ou por ICR1, como no modo PWM rápido. Se OCR1A for utilizado para determinar o valor TOP, o pino OC1A não poderá gerar um sinal PWM, somente uma onda quadrada. A resolução do PWM é dada pela mesma equação do modo PWM rápido (eq. 9.9). Um diagrama de tempo exemplo, para o PWM com correção de fase, é mostrado na fig. 9.11 para OC1A como saída invertida e OC1B como saída não-invertida.

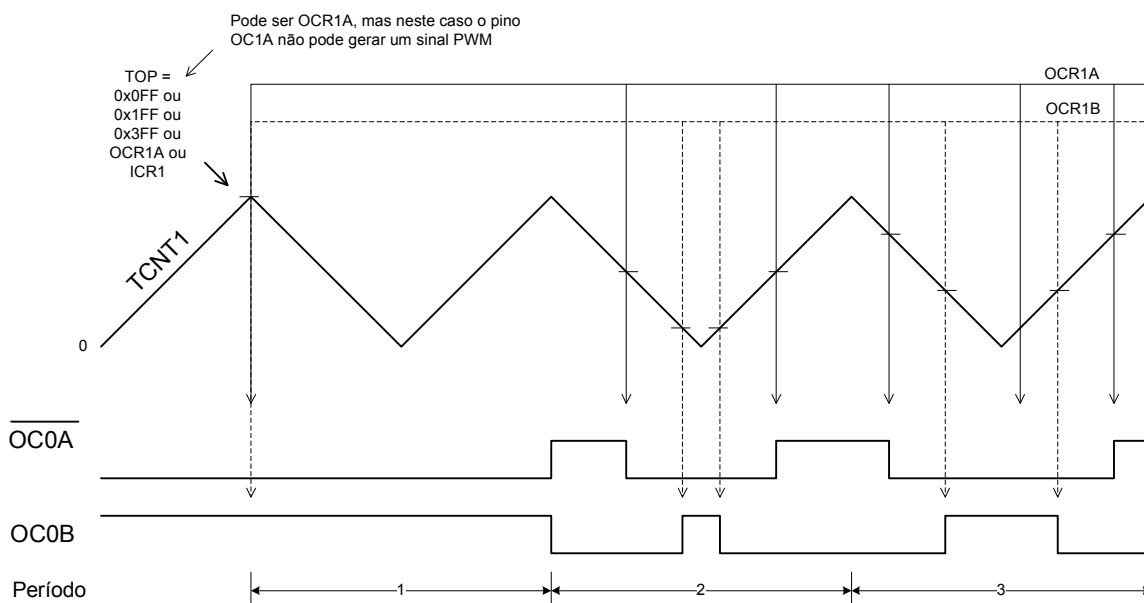


Fig. 9.11 – Diagrama de tempo para o PWM com fase corrigida, saída OC1A invertida e OC1B não invertida.

A frequência de saída do PWM com fase corrigida é calculada com:

$$f_{OC1A_PWM} = \frac{f_{osc}}{2N \cdot TOP} \quad [Hz] \quad (9.11)$$

onde f_{osc} é a frequência de operação do microcontrolador e N é o fator do *prescaler* (1, 8, 64, 256 ou 1024). Se OCR1A definir o valor TOP, a saída OC1A terá um ciclo ativo de 50%.

MODO PWM COM FASE E FREQUÊNCIA CORRIGIDAS

Este modo é igual ao PWM com fase corrigida, a diferença principal é que o pulso gerado será sempre simétrico em relação ao ponto médio do período (neste modo PWM, o TOP), mesmo quando o valor de TOP é alterado com o TC1 em operação. Isso ocorre porque os registradores de comparação são atualizados quando $TCNT1 = 0x00$, permitindo que o momento da comparação nas inclinações de subida e de descida seja sempre o mesmo (para mais detalhes consultar o manual do ATmega328).

9.5.1 REGISTRADORES DO TC1

O controle do modo de operação do TC1 é feito nos registradores TCCR1A e TCCR1B (*Timer/Counter 1 Control Registers*).

Bit	7	6	5	4	3	2	1	0
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10
Lê/Escr.	L/E	L/E	L/E	L/E	L	L	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bits 7:6 – COM1A1:0 – Compare Output Mode for channel A

Bits 5:4 – COM1B1:0 – Compare Output Mode for channel B

COM1A1:0 e COM1B1:0 controlam o comportamento dos pinos OC1A e OC1B, respectivamente. Se o valor 1 for escrito nesses bits a funcionalidade dos pinos é alterada (os bits no registrador DDRx correspondentes a OC1A e OC1B devem ser colocados em 1 para habilitar cada *driver* de saída). Quando OC1A e OC1B forem empregados, a funcionalidade dos bits COM1x1:0 dependerá do ajuste dos bits WGM13:0. Nas tabs. 9.17-19, são apresentadas as configurações dos bits COM1x1:0 para os diferentes modos de operação do TC1.

Tab. 9.17– Modo não PWM (normal e CTC).

COM1A1/COM1B1	COM1A0/COM1B0	Descrição
0	0	Operação normal dos pinos, OC1A/OC1B desconectados.
0	1	Mudança de OC1A/OC1B na igualdade de comparação.
1	0	Limpeza de OC1A/OC1B na igualdade de comparação (saída em nível lógico baixo).
1	1	OC1A/OC1B ativos na igualdade de comparação (saída em nível lógico alto).

Tab. 9.18 – Modo PWM rápido.

COM1A1/COM1B1	COM1A0/COM1B0	Descrição
0	0	Operação normal dos pinos, OC1A/OC1B desconectados.
0	1	WGM13:0 = 14 ou 15: Mudança de OC1A na igualdade de comparação, OC1B desconectado (operação normal do pino). Para os demais valores de WGM1, OC1A/OC1B estarão desconectados (operação normal dos pinos).
1	0	Limpeza de OC1A/OC1B na igualdade de comparação, ativos no valor mínimo de comparação (modo não invertido).
1	1	OC1A/OC1B ativos na igualdade de comparação, limpos no valor mínimo de comparação (modo invertido).

Tab. 9.19 – Modo PWM com correção de fase e correção de fase e frequência.

COM1A1/COM1B1	COM1A0/COM1B0	Descrição
0	0	Operação normal dos pinos, OC1A/OC1B desconectados.
0	1	WGM13:0 = 9 ou 11: Mudança de OC1A na igualdade de comparação, OC1B desconectado (operação normal do pino). Para os demais valores de WGM1, OC1A/OC1B estarão desconectados (operação normal dos pinos).
1	0	Limpeza de OC1A/OC1B na igualdade de comparação quando a contagem é crescente, ativos no valor mínimo de comparação quando a contagem é decrescente.
1	1	OC1A/OC1B ativos na igualdade de comparação quando a contagem é crescente, limpos no valor mínimo de comparação quando a contagem é decrescente.

Bits 1:0 – WGM11:0 – *Waveform Generation Mode*

Combinados com os bits WGM13:2 do registrador TCCR1B, esses bits controlam a forma de contagem do contador, a fonte para o valor máximo (TOP) e qual tipo de forma de onda gerada será empregada (tab. 9.20).

Tab. 9.20 – Descrição dos bits para os modos de geração de formas de onda.

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

TCCR1B - Timer/Counter 1 Control Register B

Bit	7	6	5	4	3	2	1	0
TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
Lê/Escr.	L/E	L/E	L	L/E	L/E	L/E	L/E	L/E
Valor	0	0	0	0	0	0	0	0
Inicial								

Bit 7 – ICNC1 – Input Capture Noise Canceler

Colocando este bit em 1, o filtro de ruído do pino de captura ICP1 é habilitado. Esse filtro requer 4 amostras sucessivas iguais para o ICP1 mudar sua saída. Assim, a captura de entrada é atrasada por 4 ciclos de *clock*.

Bit 6 – ICES1 – Input Capture Edge Select

Este bit seleciona qual borda no pino de entrada de captura (ICP1) será usada para disparar o evento de captura (ICES1=0 na transição de 1 para 0, ICES1=1 na transição de 0 para 1). Quando uma captura ocorre, o valor do contador é copiado no registrador ICR1.

Bit 4:3 – WGM13:2 – Waveform Generation Mode

Ver a tab. 9.20.

Bit 2:0 – CS12:0 – Clock Select

Existem 3 bits para a escolha da fonte de *clock* para o TC1 (tab. 9.21).

Tab. 9.21 – Descrição dos bits para seleção do *clock* para o TC1.

CS12	CS11	CS10	Descrição
0	0	0	Sem fonte de <i>clock</i> (TC1 parado).
0	0	1	<i>clock</i> /1 (prescaler = 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).

TCCR1C - Timer/Counter 1 Control Register C

Bit	7	6	5	4	3	2	1	0
TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-
Lê/Escr.	L/E	L/E	L	L	L	L	L	L
Valor	0	0	0	0	0	0	0	0
Inicial								

Bits 7:6 – FOC2A:B – Force Output Compare A e B

Estes bits são ativos somente para os modos não-PWM. Quando em 1, obrigam uma comparação no módulo gerador de forma de onda. O efeito nas saídas dependerá da configuração dado aos bits COM1A1:0 e COM1B1:0.

TCNTH e TCNTL (TCNT1) – Timer/Counter 1 Register

São os dois registrador de 8 bits onde é realizada a contagem do TC1, H (*high*) L (*Low*), podem ser lidos ou escritos a qualquer tempo.

OCR1AH e OCR1AL (OCR1A) – Output Compare 1 Register A

Registradores de comparação A de 8 bits cada, possui o valor que é continuamente comparado com o valor do contador (TCNT1). A igualdade pode ser utilizada para gerar uma interrupção ou uma forma de onda no pino OC1A.

OCR1BH e OCR1BL (OCR1B) – Output Compare 1 Register B

Registradores de comparação B de 8 bits cada, possui o valor que é continuamente comparado com o valor do contador (TCNT1). A igualdade pode ser utilizada para gerar uma interrupção ou uma forma de onda na pino OC1B.

ICR1H e ICR1L (ICR1) – Input Capture Register 1

Esses registradores são atualizados com o valor do TCNT1 cada vez que um evento ocorre no pino ICP1 (ou opcionalmente nos pinos do comparador analógico). Também são empregados para definir o valor máximo de contagem (TOP).

TIMSK1 – Timer/Counter 1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0
TIMSK1	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1
Lê/Escreve	L	L	L/E	L	L	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit 5 – ICIE1 – Timer/Counter 1, Input Capture Interrupt Enable

A escrita de 1 neste bit ativa a interrupção por captura da entrada. Quando ocorre uma mudança no pino ICP1 ou nos pinos do comparador analógico, o valor do TCNT1 é salvo no registrador ICR1.

Bit 2 – OCIE1B – Timer/Counter 1 Output Compare Match B Interrupt Enable

A escrita de 1 neste bit ativa a interrupção do TC1 na igualdade de comparação com o registrador OCR1B.

Bit 1 – OCIE1A – Timer/Counter 1 Output Compare Match A Interrupt Enable

A escrita de 1 neste bit ativa a interrupção do TC1 na igualdade de comparação com o registrador OCR1A.

Bit 0 – TOIE1 – Timer/Counter 1 Overflow Interrupt Enable

A escrita de 1 neste bit ativa a interrupção por estouro do TC1.

As interrupções individuais dependem da habilitação das interrupções globais pelo bit I do registrador SREG.

TIFR1 – Timer/Counter 1 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0
	TIFR2							
	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
Lê/Escreve	L	L	L/E	L	L	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit 5 – ICF1 – Timer/Counter 1 Input Capture Flag

Este bit é colocado em 1 quando um evento relacionado ao pino ICP1 ocorre.

Bit 2 – OCF1B – Timer/Counter 1 Output Compare B Match Flag

Este bit é colocado em 1 quando o valor da contagem (TCNT1) é igual ao valor do registrador de comparação de saída B (OCR1B) do TC1.

Bit 1 – OCF1A – Timer/Counter 1 Output Compare A Match Flag

Este bit é colocado em 1 quando o valor da contagem (TCNT1) é igual ao valor do registrador de comparação de saída A (OCR1A) do TC1.

Bit 0 – TOV1 – Timer/Counter 1 Overflow Flag

Este bit é colocado em 1 quando um estouro do TC1 ocorre.

9.5.2 CÓDIGOS EXEMPLO

MODO NORMAL – GERANDO UM PEDIDO DE INTERRUPÇÃO

O código abaixo é utilizado para gerar um evento periódico de interrupção a cada 4,19 s. Neste caso, um LED ligado ao pino PB5 troca de estado (liga ou desliga) a cada interrupção. Foi empregado o maior divisor de *clock* possível, 1024. Em relação ao exemplo apresentado na seção 9.3.2 (TC0), percebe-se o aumento considerável no tempo de temporização, pois, desta vez, o TC1 conta 65536 vezes ao invés de apenas 256.

TC1_estouro.c

```
//===== //
//      HABILITANDO A INTERRUPÇÃO POR ESTOURO DO TC1      //
//===== //
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#define cpl_bit(y,bit) (y^=(1<<bit))//troca o estado lógico do bit x da variável Y
#define LED      PB5
//-----
ISR(TIMR1_OVF_vect)          //interrupção do TC1
{
    cpl_bit(PORTB,LED);
}
//-----
```

```

int main()
{
    DDRB = 0b00100000; //somente pino do LED como saída
    PORTB = 0b11011111; //apaga LED e habilita pull-ups nos pinos não utilizados

    TCCR1B = (1<<CS12) | (1<<CS10); /*TC1 com prescaler de 1024, a 16 MHz gera uma
                                     interrupção a cada 4,19 s*/

    TIMSK1 = 1<<TOIE1; //habilita a interrupção do TC1
    sei()                //habilita interrupções globais

    while(1)
    {
        /*Aqui vai o código, a cada estouro do TC1 o programa desvia para ISR(TIMER1_OVF_vect)*/
    }
}
//=====

```

MODO DE CAPTURA DE ENTRADA

O programa abaixo utiliza um botão para gerar uma interrupção por captura no pino ICP1. Não foi feito o tratamento do ruído do botão e cada vez que o botão é pressionado o pino PB5 troca de estado. O valor do registrador ICR1 é utilizado para ler o valor do TC1.

TC1_captura.c

```

//===== //
//      HABILITANDO A INTERRUPÇÃO POR CAPTURA      //
//===== //
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#define cpl_bit(y,bit) (y^=(1<<bit))//troca o estado lógico do bit x da variável Y
#define LED          PB5
//-----
/*interrupção do TC1, toda vez que ocorrer um evento no pino ICP1 (PB0) ICR1 terá o
valor de contagem do TCNT1*/
//-----
ISR(TIMER1_CAPT_vect)
{
    cpl_bit(PORTB,LED); //troca estado do pino PB5
}
//-----
int main()
{
    DDRB = 0b00100000; //somente pino do LED como saída, botão no PB0
    PORTB = 0b11011111; //apaga LED e habilita pull-ups nos outros pinos
    TCCR1B = 1<<CS10; //TC1 com prescaler = 1, captura na borda de descida
    TIMSK1 = 1<<ICIE1; //habilita a interrupção por captura
    sei(); //habilita interrupções globais

    while(1)
    {
        /*Aqui vai o código, a cada evento no pino ICP1 o programa desvia para
                                     ISR(TIMER1_CAPT_vect)*/
    }
}
//=====

```

MOD0 CTC

TC1_PWMs.c

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

int main(void)
{
    DDRB = 0b00000110; //pinos OC1B e OC1A (PB2 e PB1) como saída
    PORTB = 0b11111001; //zera saídas e habilita pull-ups nos pinos não utilizados

    //MOD0 CTC - TOP = ICR1
    TCCR1A = 0b01010000; /*habilita OC1A e OC1B para trocar de estado na igualdade de
                                                                    comparação*/
    TCCR1B = 0b00011011; //liga TC1 com prescaler = 64.
    ICR1 = 10000;          //valor máximo de contagem

    while(1)
    {
        //o programa principal vai aqui
    }
}
```

MOD0 PWM RÁPIDO

```
...
//fast PWM, TOP = ICR1, OC1A e OC1B habilitados
TCCR1A = 0b10100010; //PWM não invertido nos pinos OC1A e OC1B
TCCR1B = 0b00011001; //liga TC1, prescaler = 1
ICR1 = 35000;          //valor máximo para contagem
OCR1A = 2000;          //controle do ciclo ativo do PWM OC1A
OCR1B = 100;           //controle do ciclo ativo do PWM OC1B

...
```

MOD0 PWM COM FASE E FREQUÊNCIA CORRIGIDAS

```
...
//phase and frequency correct PWM, TOP = OCR1A
TCCR1A = 0b00100011; // OC1B habilitado, modo não invertido
TCCR1B = 0b00011001; //liga TC1, prescaler = 1 e habilita
OCR1A = 300;          //máximo valor para contagem
OCR0B = 100;          //controle do ciclo ativo do PWM OC1B

...
```

MOD0 CTC – CONTANDO EVENTOS EXTERNOS

Abaixo, é apresentado um programa para gerar uma interrupção a cada 1 s baseado num sinal de 60 Hz externo ao microcontrolador. Esse tipo de sinal, por exemplo, é utilizado em rádios-relógio para gerar a base

de tempo. Na fig. 9.12, é apresentado o circuito para o programa. O sinal de 60 Hz é digital, sendo, usualmente obtido da rede elétrica após um tratamento adequado.

TC1_externo.c

```
//===== //
//          TC1 estouro na igualdade de comparação - sinal externo          //
//          Pisca LED a cada 1 segundo                                     //
//===== //
#include <avr/io.h>
#include <avr/interrupt.h>

#define cpl_bit(Y,bit_x) (Y^=(1<<bit_x))
#define LED             PB5
//-----
ISR(TIMER1_COMPA_vect)    //sub-rotina de interrupção por igualdade de comparação
{
    cpl_bit(PORTB,PB5); //troca o estado do LED do pino PB5
}
//-----
int main()
{
    DDRD = 0x00;          //PORTD será a entrada do sinal de clock para o TC1 (PD3)
    DDRB = 1<<PB5;        //pino PB5 é a saída para o LED      de sinalização

    TIMSK1 = 1<<OCIE1A; //habilita a interrupção do TC1 por igualdade de comparação
    TCCR1B = (1<<WGM12)|(1<<CS12) | (1<<CS11) |(1<<CS10); /*clock externo contagem na
                                                           borda de subida - modo CTC*/
    OCR1A = 59;          /*valor para a contagem máxima do TC1 (conta 60 vezes) - valor de
                           comparação. Como o sinal de clock externo é de 60 Hz, é gerada
                           uma interrupção a cada 1 s*/

    sei();                //liga a interrupção

    while(1){}
}
//=====
```

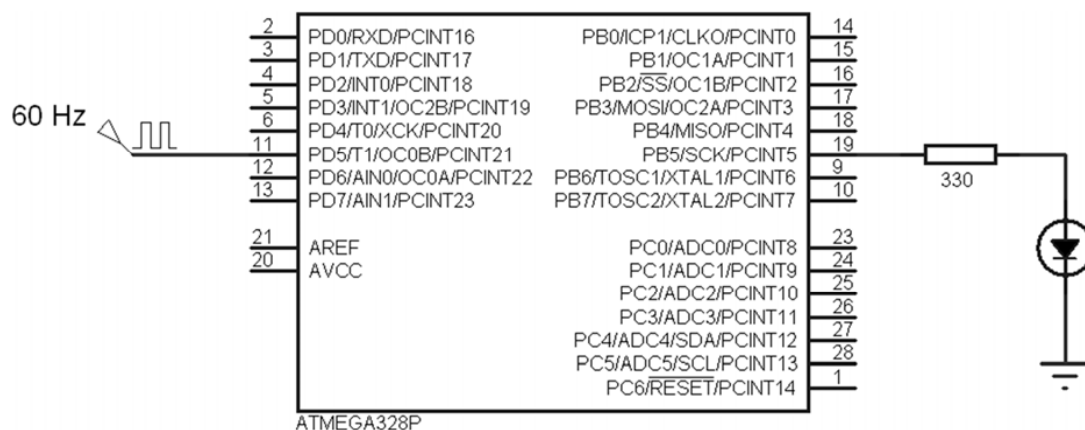


Fig. 9.12 – Sinal de 60 Hz para ligar um LED a cada 1 s.