9. TEMPORIZADORES/CONTADORES

Uma necessidade importante no trabalho com circuitos microcontrolados é a geração de sinais periódicos e eventos. Essas funcionalidades são realizadas pelos contadores/temporizadores (TCs) e muitos projetos só podem ser executados com o uso deles. Logo, entender o seu funcionamento é indispensável, permitindo a escrita de programas eficientes e projetos bem feitos. Neste capítulo, são apresentados os conceitos essenciais para o trabalho com os TCs do ATmega e um resumo para o trabalho com o seus registradores.

No ATmega328, existem dois temporizadores/contadores de 8 bits (TC0 e TC2) e um de 16 bits (TC1). Todos independentes e com características próprias.

O Temporizador/Contador 0 (TC0) emprega 8 bits (permite contagens de 0 até 255). Suas principais características são:

- Contador simples (baseado no *clock* da CPU).
- Contador de eventos externos.
- Divisor do *clock* para o contador de até 10 bits - *prescaler*.
- Gerador para 2 sinais PWM (pinos OC0A e OC0B).
- Gerador de frequência (onda quadrada).
- 3 fontes independentes de interrupção (por estouro e igualdades de comparação).

O Temporizador/Contador 2 (TC2) também emprega 8 bits e suas características são similares ao TC0. Entretanto, apresenta uma função especial para a contagem precisa de 1 s, permitindo o uso de um cristal externo independente para o seu *clock* (32,768 kHz). Pode gerar dois sinais PWM nos pinos OC2A e OC2B.

O Temporizador/Contador 1 (TC1) possui 16 bits e pode contar até 65535. Permite a execução precisa de temporizações, geração de formas de onda e medição de períodos de tempo. Suas principais características são:

- Contador simples (baseado no *clock* da CPU).
- Contador de eventos externos.
- Divisor do *clock* para o contador de até 10 bits - *prescaler*.
- Gerador para 2 sinais PWM (pinos OC1A e OC1B) com inúmeras possibilidades de configuração.
- Gerador de frequência (onda quadrada).
- 4 fontes independentes de interrupção (por estouro e igualdades de comparação).

9.1 TEMPORIZANDO E CONTANDO

Uma das principais função dos TCs é a contagem de pulsos de *clock*, os quais podem ser externos ou internos ao microcontrolador. Dessa forma, permitem a geração de eventos periódicos para uso do programador ou para a determinação de um número de contagens.

Um estouro do contador ocorre quando ele passa do valor máximo permitido para a contagem para o valor zero. Assim, se o TC for de 8 bits, ele contará de 0 até 255 resultando em 256 contagens; se for de 16 bits, contará de 0 até 65535, resultando em 65536 contagens. Assim, o tempo que um TC leva para estourar é dado por:

$$t_{estouro} = \frac{(TOP+1)\times prescaler}{f_{osc}} \qquad (9.1)$$

onde *TOP* é o valor máximo de contagem, f_{osc} é a frequência do *clock* empregado (oscilador) e o *prescaler* é o divisor dessa frequência.

Exemplos:

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

$$t_{estouro} = 256 \times \frac{1}{1\ \text{MHz}} \times 1 = 256\ \mu\text{s}$$

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

$$t_{estouro} = 65536 \times \frac{1}{16\ \text{MHz}} \times 64 = 0{,}262\ \text{s}$$

Na fig. 9.1, o funcionamento de um TC para o ATmega é ilustrado.

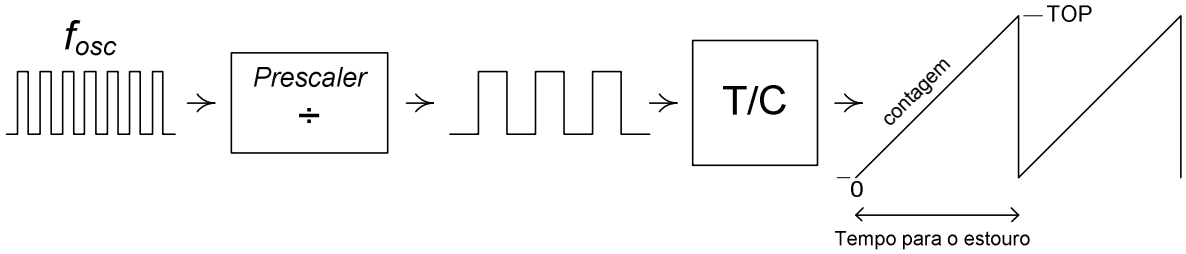

Fig. 9.1 – Funcionamento de um temporizador/contador do ATmega.

9.2 MODULAÇÃO POR LARGURA DE PULSO – PWM

A geração de sinais PWM é outra função importante dos TCs. O uso desses sinais é baseado no conceito de valor médio de uma forma de onda periódica. Baseado no valor médio de um sinal PWM, muitos dispositivos eletrônicos podem ser controlados, como por exemplo: motores, lâmpadas, LEDs, fontes chaveadas e circuitos inversores.

Digitalmente, 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 de tempo. No PWM, esse tempo é chamado de ciclo ativo (*Duty Cycle*). Na fig. 9.2, são apresentadas formas de onda PWM em que a largura do pulso (*Duty Cycle*) é variável de 0 até 100%, com incrementos de 25%.

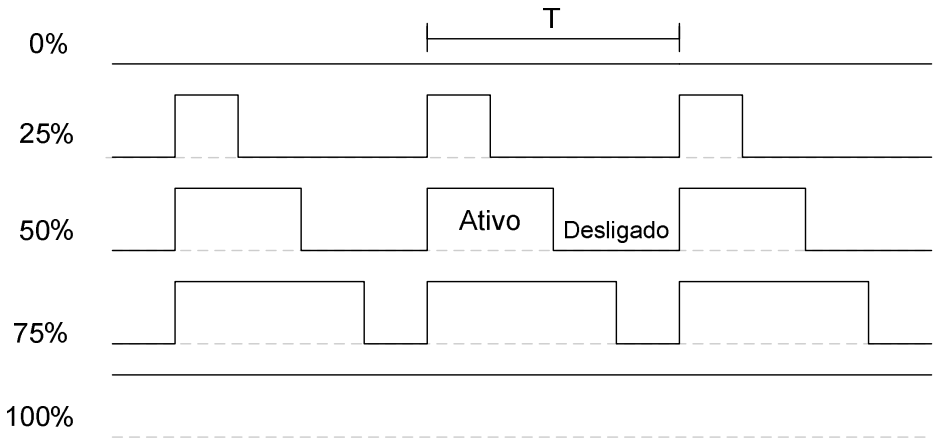

Fig. 9.2 – PWM com período T e ciclo ativo de 0%, 25%, 50%, 75% e 100%.

O cálculo do valor médio de um sinal digital é dado por:

$$V_{médio} = \frac{Amplitude\ Máxima}{Período} \times (Tempo\ Ativo\ no\ Periodo) \qquad (9.2)$$

Assim, 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. Logo, ao se alterar o ciclo útil do sinal PWM, altera-se o seu valor médio. Na fig. 9.3, é ilustrada a variação do ciclo ativo de um sinal PWM de 0 a 100% do seu ciclo útil (período) com o passar do tempo.

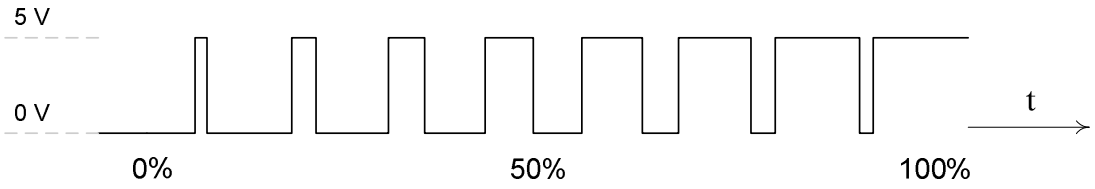

Fig. 9.3 – Variação do ciclo ativo de um sinal PWM em um intervalo de tempo.

É importante notar que o período do sinal PWM não se altera, e sim sua largura de ciclo ativo. Desta forma, quando se gera um sinal PWM, deve-se em primeiro lugar determinar sua frequência de trabalho.

A resolução do PWM em um microcontrolador é dada pelo seu número de bits e indica quantos diferentes ciclos ativos podem ser gerados. Por exemplo, um PWM de 10 bits pode gerar 1024 diferentes níveis, começando com o nível de tensão 0 e terminando com 100 % do ciclo ativo. Como será visto posteriormente, no ATmega o ciclo ativo de um sinal PWM é controlado pela escrita em registradores específicos e sua resolução vai depender do TC empregado.

Além de permitir a geração de sinais PWM, os TCs do ATmega permitem gerar a forma de onda quadrada. Sempre que se desejar gerar tais sinais, os pinos correspondentes do microcontrolador devem ser configurados como pinos de saída, caso contrário o sinal não aparecerá no pino.

9.3 TEMPORIZADOR/CONTADOR 0

O TC0 é um contador de 8 bits que é incrementado com pulsos de *clock,* os quais podem ser obtidos da fonte interna de *clock* do microcontrolador ou de uma fonte externa. Existem vários modos de operação: normal, CTC, PWM rápido e PWM com fase corrigida; permitindo desde simples contagens até a geração de diferentes tipos de sinais PWM.

MODO NORMAL

É o modo mais simples de operação, onde o TC0 conta continuamente de forma crescente. A contagem se dá de 0 até 255 voltando a 0, num ciclo contínuo. Quando a contagem passa de 255 para 0, ocorre o estouro e então, o bit sinalizador de estouro (TOV0) é colocado em 1. Se habilitada, uma interrupção é gerada.

A contagem é feita no registrador TCNT0 e um novo valor de contagem pode ser escrito a qualquer momento, permitindo-se alterar o número de contagens via programação.

MODO CTC

No modo CTC (*Clear Timer on Compare* - limpeza do contador na igualdade de comparação), o registrador OCR0A é usado para manipular a resolução do TC0. Neste modo, o contador é zerado quando o valor do contador (TCNT0) é igual ao valor de OCR0A (o valor TOP da contagem), ou seja, o contador conta de 0 até o valor de OCR0A. Isso permite um controle mais fino da frequência de operação e da resolução do TC0. O modo CTC permite configurar 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). Um exemplo de diagrama de tempo para o modo CTC é mostrado na fig. 9.4, onde os pinos OC0A e OC0B foram configurados para trocar de estado quando ocorre a igualdade entre o valor do registrador TCNT0 com OCR0A e OCR0B.

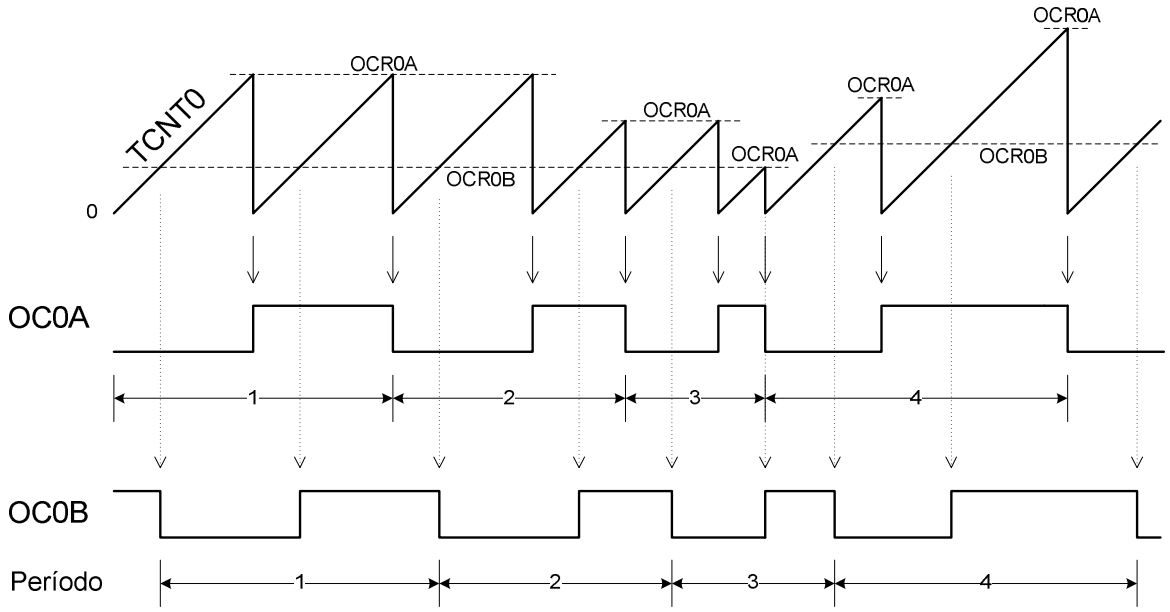

Fig. 9.4 – Modo CTC para a geração de ondas quadradas.

No exemplo, para o pino OC0B, o registrador OCR0B é empregado para deslocar a forma de onda em relação a OC0A e deve ter valor menor ou igual a OCR0A. Para gerar uma onda de saída no modo CTC, os pinos OC0A e/ou OC0B devem ser configurados como saídas e ajustados para

trocar de estado em cada igualdade de comparação através dos bits do modo de comparação.

A frequência no modo CTC é definida por:

$$f_{OC0A} = \frac{f_{osc}}{2N(1+OCR0A)} \qquad \text{[Hz]} \tag{9.3}$$

onde f_{osc} é a frequência de operação do microcontrolador, OCR0A assume valores entre 0 e 255 e N é o fator do *prescaler* (1, 8, 64, 256 ou 1024). Para calcular o valor de OCR0A baseado na frequência desejada, basta isolá-lo na equação acima, o que resulta em:

$$\text{OCR0A} = \frac{f_{osc}}{2N.f_{OC0A}} - 1 \tag{9.4}$$

O modo CTC não necessita ser empregado para a geração de formas de onda, pode ser utilizado para temporizações e contagens externas. Ele permite flexibilizar o número de contagens do TC0 e gerar interrupções por igualdade de comparação do TCNT0 com os registradores OCR0A e OCR0B.

MODO PWM RÁPIDO

O modo de modulação por largura de pulso 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 OC0A é zerado na igualdade entre TCNT0 e OCR0A, e colocado em 1 no valor mínimo do contador. No modo de comparação com saída invertida, o processo é inverso: OC0A é ativo na igualdade e zerado no valor mínimo. É o registrador OCR0A que determina o ciclo ativo do sinal PWM. Também é possível habilitar o pino OC0B para gerar um sinal PWM, onde o ciclo ativo do PWM será controlado pelo valor de OCR0B. A contagem do TC0 pode ser ajustada para 255 ou pelo valor dado por OCR0A; se for utilizado o registrador OCR0A, o pino OC0A não poderá gerar um sinal PWM, apenas uma onda quadrada. Quando se controla o ciclo ativo do sinal

PWM, o valor zero para OCR0A produz um pequeno ruído e o valor máximo (255) vai deixar o sinal PWM em 0 ou 1, conforme foi habilitada a saída, invertida ou não.

A alta frequência do PWM rápido o torna adequado para aplicações de regulação de potência, retificação e outras usando conversores DAs. Na fig. 9.5, é apresentado o exemplo do diagrama temporal para o modo PWM rápido para as saídas não invertidas e contagem máxima do TC de 255.

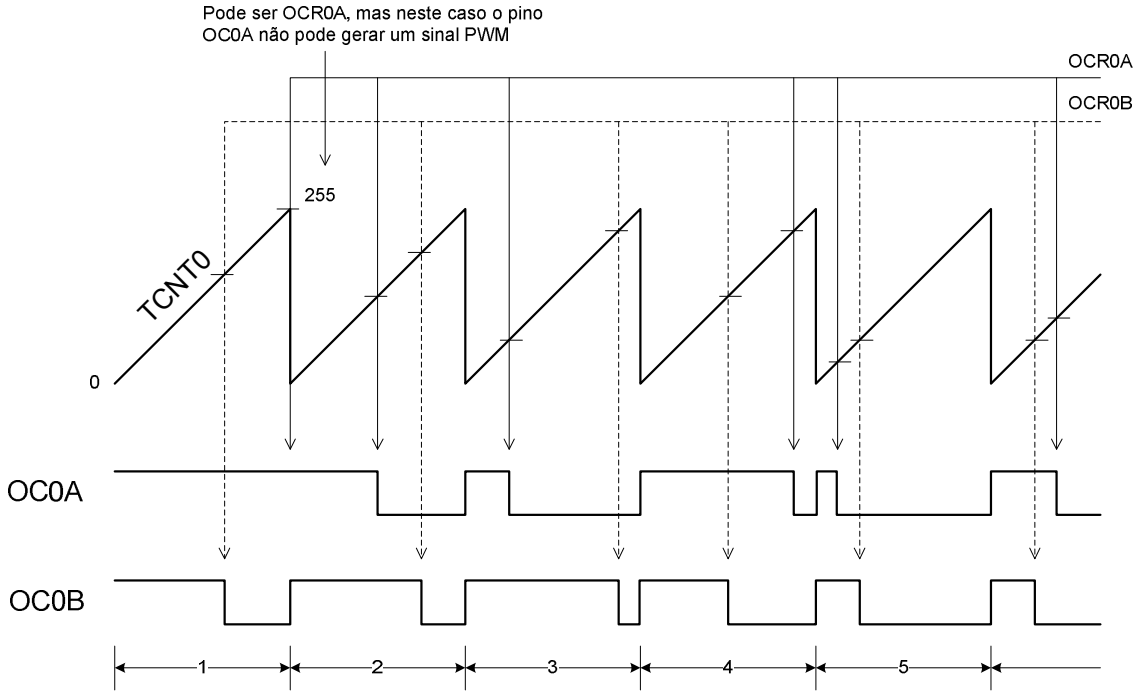

Fig. 9.5 – Modo PWM rápido, saídas não invertidas.

O valor de comparação OCR0A pode ser atualizado na interrupção por estouro do contador cada vez que o contador chegar no valor máximo de contagem, gerando um sinal PWM variável.

A frequência de saída do PWM é calculada como:

$$f_{OC0A\_PWM} = \frac{f_{osc}}{N(1+TOP)} \qquad \text{[Hz]} \qquad (9.5)$$

onde f_{osc} é a frequência de operação do microcontrolador, N é o fator do *prescaler* (1, 8, 64, 256 ou 1024) e TOP assume valores conforme tab. 9.7.

Um sinal com ciclo ativo do PWM de 50% pode ser obtido ajustando-se OC0A para mudar de estado a cada igualdade de comparação. A máxima frequência é obtida quando OCR0A é zero ($f_{OC2\_PWM} = f_{osc}/2$).

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 TCNT0, 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. 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 (TCNT0 = 0x00), qualquer que seja a razão cíclica.

Por ser um sinal que permite um maior controle do ciclo ativo do sinal PWM, é adequado para o controle de motores, sendo mais lento e preciso que o modo PWM rápido.

Para o sinal PWM não invertido, o pino OC0A é zerado na igualdade entre TCNT0 e OCR0A, quando a contagem é crescente, e colocado em 1 quando a contagem é decrescente. No modo de comparação com saída invertida, o processo é contrário. A comparação para o pino OC0B é feita com o registrador OCR0B. O valor máximo de contagem pode ser definido como 255 ou pelo valor de OCR0A, como no modo PWM rápido; se OCR0A for utilizado para determinar o valor TOP, o pino OC0A não poderá gerar um sinal PWM, somente uma onda quadrada.

Na fig. 9.6, é exemplificado um diagrama de tempo do modo PWM com fase corrigida com a saída OC0A não invertida e a saída OC0B invertida.

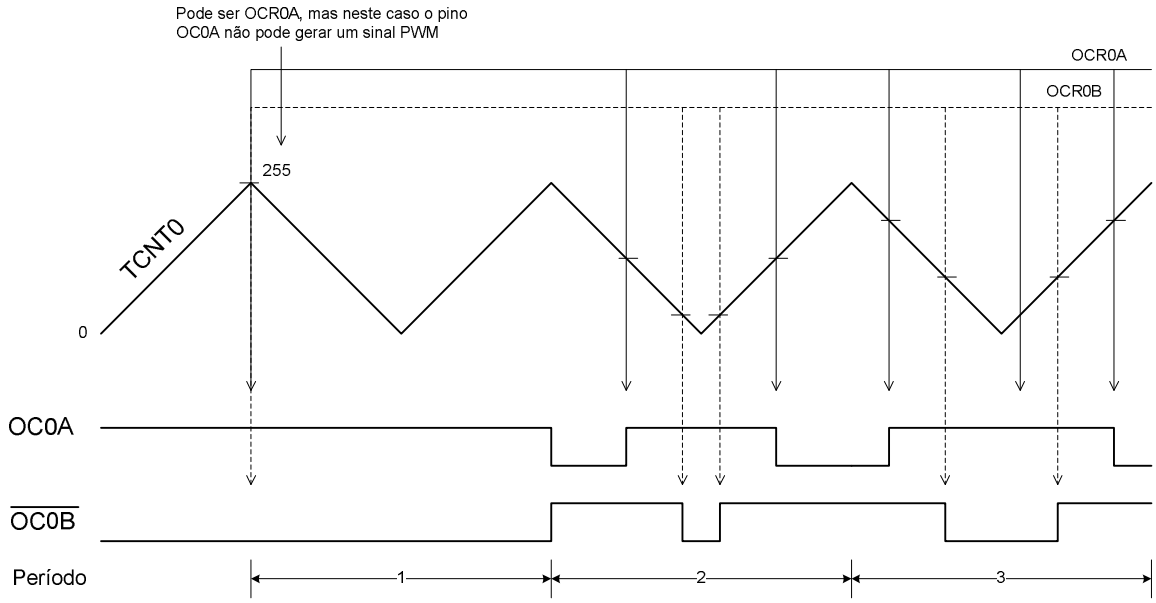

Fig. 9.6 – Modo PWM com correção de fase , OC0A não invertida e OC0B invertida.

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

$$f_{OC0A\_PWM} = \frac{f_{osc}}{2N(1+\mathrm{TOP})} \qquad \text{[Hz]} \qquad (9.6)$$

onde f_{osc} é a frequência de operação do microcontrolador, N é o fator do *prescaler* (1, 8, 64, 256 ou 1024) e TOP assume valores conforme tab. 9.7.

9.3.1 REGISTRADORES DO TC0

O controle do modo de operação do TC0 é feito nos registradores TCCR0A e TCCR0B (*Timer/Counter Control* 0 *Register A e B*).

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

Bits 7:6 – COM0A1:0 – *Compare Match Output A Mode*
Estes bits controlam o comportamento do pino OC0A (*Output Compare* 0A). Se um ou ambos os bits forem colocados em 1, a funcionalidade normal do pino é alterada. Entretanto, o bit do registrador DDRx correspondente ao pino OC0A deve estar ajustado para habilitar o *driver* de saída. A funcionalidade dos bits COM0A1:0 depende do ajuste dos bits WGM02:0. Suas possíveis configurações são apresentadas nas tabs. 9.1-3.

Tab. 9.1 – Modo CTC (não PWM).

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.

Tab. 9.2 – Modo PWM rápido.

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 ativo no valor do TC mínimo (modo não invertido).
1	1	OC0A é ativo na igualdade de comparação e limpo no valor do TC mínimo (modo invertido).

Tab. 9.3 – Modo PWM com fase corrigida.

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 é na igualdade de comparação quando a contagem é crescente, e ativo na igualdade de comparação quando a contagem é decrescente.
1	1	OC0A é ativo na igualdade de comparação quando a contagem é crescente, e limpo na igualdade de comparação quando a contagem é decrescente.

Bits 5:4 – COM0B1:0 – *Compare Match Output B Mode*
Estes bits controlam o comportamento do pino OC0B (*Output Compare* 0B). Se um ou ambos os bits forem colocados em 1, a funcionalidade normal do pino é alterada. Entretanto, o bit do registrador DDRx correspondente ao pino OC0B deve estar ajustado para habilitar o *driver* de saída. A funcionalidade dos bits COM0B1:0 depende do ajuste dos bits WGM02:0. Suas possíveis configurações são apresentadas nas tabs. 9.4-6.

Tab. 9.4 – Modo CTC (não PWM).

COM0B1	COM0B0	Descrição
0	0	Operação normal do pino, OC0B desconectado.
0	1	Mudança do estado de OC0B na igualdade de comparação.
1	0	OC0B é limpo na igualdade de comparação.
1	1	OC0B é ativo na igualdade de comparação.

Tab. 9.5 – Modo PWM rápido.

COM0B1	COM0B0	Descrição
0	0	Operação normal do pino, OC0B desconectado.
0	1	Reservado.
1	0	OC0B é limpo na igualdade de comparação, OC0B ativo no valor do TC mínimo (modo não invertido).
1	1	OC0B é ativo na igualdade de comparação e limpo no valor do TC mínimo (modo invertido).

Tab. 9.6 – Modo PWM com fase corrigida.

COM0B1	COM0B0	Descrição
0	0	Operação normal do pino, OC0B desconectado.
0	1	Reservado.
1	0	OC0B é limpo é na igualdade de comparação quando a contagem é crescente, e ativo na igualdade de comparação quando a contagem é decrescente.
1	1	OC0B é ativo na igualdade de comparação quando a contagem é crescente, e limpo na igualdade de comparação quando a contagem é decrescente.

Bits 1:0 – WGM01:0 – *Wave Form Generation Mode*
Combinados com o bit WGM02 do registrador TCCR0B, estes bits controlam a sequência de contagem do contador, a fonte do valor máximo para contagem (TOP) e o tipo de forma de onda a ser gerada, conforme tab. 9.7.

Tab. 9.7 – Bits para configurar o modo de operação do TC0.

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

TCCR0B – *Timer/Counter* 0 *Control Register B*

Bit		7	6	5	4	3	2	1	0
	TCCR0B	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00
Lê/Escr.		E	E	L	L	L/E	L/E	L/E	L/E
Valor Inic.		0	0	0	0	0	0	0	0

Bits 7:6 – FOC0A:B – ***Force Output Compare A e B***
Estes bits são ativos somente para os modos não-PWM. Quando em 1, uma comparação é forçada no módulo gerador de onda. O efeito nas saídas dependerá da configuração dada aos bits COM0A1:0 e COM0B1:0.

Bit 3 – WGM02 – ***Wave Form Generation Mode***
Função descrita na tab. 9.7.

Bits 2:0 – CS02:0 – ***Clock Select***
Bits para seleção da fonte de *clock* para o TC0, conforme tab. 9.8.

Tab. 9.8 – Seleção do *clock* para o TC0.

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

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 – ***Output Compare* 0 *Register A***
Registrador de comparação A 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.

OCR0B – ***Output Compare* 0 *Register B***
Registrador de comparação 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 OC0B.

TIMSK0 – *Timer/Counter 0 Interrupt Mask Register*

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.

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

TIFR0 – *Timer/Counter 0 Interrupt Flag Register*

		7	6	5	4	3	2	1	0
Bit									
	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0
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 – OCF0B – *Timer/Counter 0 Output Compare B Match Flag*
Este bit é colocado em 1 quando o valor da contagem (TCNT0) é igual ao valor do registrador de comparação de saída B (OCR0B) do TC0.

Bit 1 – OCF0A – *Timer/Counter 0 Output Compare A Match Flag*
Este bit é colocado em 1 quando o valor da contagem (TCNT0) é igual ao valor do registrador de comparação de saída A (OCR0A) do TC0.

Bit 0 – TOV0 – *Timer/Counter 0 Overflow Flag*
Este bit é colocado em 1 quando um estouro do TC0 ocorre.

9.3.2 CÓDIGOS EXEMPLO

MODO NORMAL – GERANDO UM PEDIDO DE INTERRUPÇÃO

O código abaixo é utilizado para gerar um evento periódico de interrupção a cada 16,384 ms. 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, e não há possibilidade de gerar um intervalo de tempo maior sem a diminuição da frequência de trabalho do microcontrolador.

TC0_estouro.c

```
//============================================================================== //
//       HABILITANDO A INTERRUPÇÃO POR ESTOURO DO TC0                            //
//============================================================================== //
#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(TIMER0_OVF_vect) //interrupção do TC0
{
     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

     TCCR0B = (1<<CS02) | (1<<CS00);/*TC0 com prescaler de 1024, a 16 MHz gera uma
                                                    interrupção a cada 16,384 ms*/
     TIMSK0 = 1<<TOIE0;   //habilita a interrupção do TC0
     sei();               //habilita a chave de interrupção global

     while(1)
     {
          /*Aqui vai o código, a cada estouro do TC0 o programa desvia para
                                                              ISR(TIMER0_OVF_vect)*/
     }
}
//==============================================================================
```

MODO CTC

Neste exemplo, os pinos OC0A e OC0B são configurados para trocar de estado na igualdade de comparação do registrador TCNT0 com o OCR0A e OCR0B, respectivamente.

TC0_PWMs.c

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

int main(void)
{
     DDRD  = 0b01100000;  //pinos OC0B e OC0A (PD5 e PD6) como saída
     PORTD = 0b10011111;  //zera saídas e habilita pull-ups nos pinos não utilizados

     //MODO CTC
     TCCR0A = 0b01010010; /*habilita OC0A e OC0B para trocar de estado na igualdade de
                                                                          comparação*/
     TCCR0B = 0b00000001; //liga TC0 com prescaler = 1.
     OCR0A  = 200;        //máximo valor de contagem
     OCR0B  = 100;        //deslocamento de OC0B em relação a OC0A

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

MODO PWM RÁPIDO

Abaixo são apresentadas algumas configurações, considerando que os pinos OC0A e OC0B estejam configurados como saída.

```
...
//fast PWM, TOP = 0xFF, OC0A e OC0B habilitados
TCCR0A = 0b10100011; //PWM não invertido nos pinos OC0A e OC0B
TCCR0B = 0b00000011; //liga TC0, prescaler = 64
OCR0A = 200;         //controle do ciclo ativo do PWM 0C0A
OCR0B = 50;          //controle do ciclo ativo do PWM OC0B

...
/*fast PWM, TOP = OCR0A com OC0A gerando uma onda quadrada e OC0B um sinal PWM não
                                                                          Invertido*/
TCCR0A = 0b01100011; //TOP = OCR0A, OC0A e OC0B habilitados
TCCR0B = 0b00001001; //liga TC0, prescaler = 1 e ajusta modo para comparação com OCR0A
OCR0A  = 100;        //controle da período do sinal no pino OC0A
OCR0B  = 30;         //controle do ciclo ativo do PWM 0C0B
```

MODO PWM COM FASE CORRIGIDA

Abaixo são apresentadas algumas configurações, considerando que os pinos OC0A e OC0B estejam configurados como saída.

```
...
//phase correct PWM, TOP = 0xFF
TCCR0A = 0b10100001; //OC0A e OC0B habilitados
TCCR0B = 0b00000001; //liga TC0, prescaler = 1
OCR0A = 100;         //controle da período do sinal no pino OC0A
OCR0B = 50;          //controle do ciclo ativo do PWM 0B

...
//phase correct PWM, TOP = OCR0A
TCCR0A = 0b01100011; //OC0A e OC0B habilitado
TCCR0B = 0b00001001; /*liga TC0, prescaler = 1 e habilita o pino OC0A para gerar um
                                                                    onda quadrada*/
OCR0A = 200;         //controle do período do sinal no pino OC0A
OCR0B = 10;          //controle do ciclo ativo do PWM OC0B
```

Exercícios:

9.1 – Considerando o TC0 trabalhando a 8 MHz com um *prescaler* de 256, quanto tempo leva para o TC0 gerar um estouro de contagem?

9.2 – Considerando-se o PWM rápido configurado para que o valor máximo de contagem do TC0 seja dado pelo valor de OCR0A e que o pino OC0B esteja configurado para gerar o sinal PWM, determine:

- Qual o período do sinal PWM se o ATmega328 estiver rodando a 12 MHz e o valor de OCR0A for 99?
- Supondo que OCR0A é 200, qual deve ser o valor de OCR0B para que o ciclo ativo do sinal PWM (OC0B) seja de 75 %?

9.3 – Verifique a configuração dada ao TC0 para os exemplos acima. Analise os bits dos registradores envolvidos.

9.4 – Com o emprego de um osciloscópio, teste os sinais gerados pelos códigos exemplo dados previamente, gravando-os no Arduino. Altere os valores dos registradores OCR0A e OCR0B para analisar o comportamento do microcontrolador. Para a análise, também pode ser empregado o software de simulação Proteus (ISIS).

9.5 – Elaborar um programa para ler 6 teclas utilizando a interrupção externa INT0 do ATmega328 (ver o capítulo 6). O programa principal não deve ficar responsável pela varredura do teclado (ver o capítulo 8); a interrupção do TC0 deve ser empregada para esse fim. O tratamento do

teclado será transparente ao programa principal, devendo ser feito nas interrupções. Assim, quando uma tecla for pressionada será requisitado um pedido de interrupção e o LED correspondente deve ser ligado. O circuito abaixo ilustra o circuito necessário.

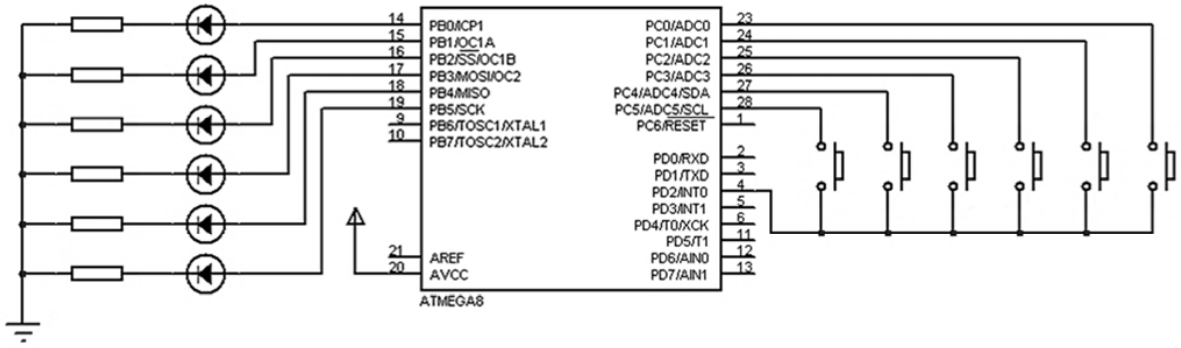

Fig. 9.7 – Seis teclas gerando um pedido de interrupção com o emprego de apenas uma interrupção externa (obs.: a pinagem do ATmega8 é igual a do ATmega328).

9.4 TEMPORIZADOR/CONTADOR 2

O TC2 possui características similares a do TC0, é um contador de 8 bits que permite os modos normal, CTC, PWM rápido e PWM com fase corrigida. Possui dois pinos de saída para os sinais gerados, OC2A (PB3) e OC2B (PD3), ligados aos PORTB e PORTD, respectivamente. Não possui entrada para contagem de sinais de *clock* externos como o TC0. Entretanto, possui mais possibilidades para a divisão do *clock* para a contagem (*prescaler*) e uma característica peculiar: permite o uso de um cristal externo de 32,768 kHz para o seu circuito de *clock*. Essa característica é denominada operação assíncrona.

Para o TC2, os registradores de comparação são o OCR2A e OCR2B e as fórmulas de cálculo para as frequência de trabalho do PWM são similares as do TC0, ver as eqs. 9.3-6. O registrador principal é o OCR2A, que pode ser configurado para determinar o valor máximo de contagem. O *prescaler* pode assumir os seguintes valores: 1, 8, 32, 64, 128, 256 ou 1024.