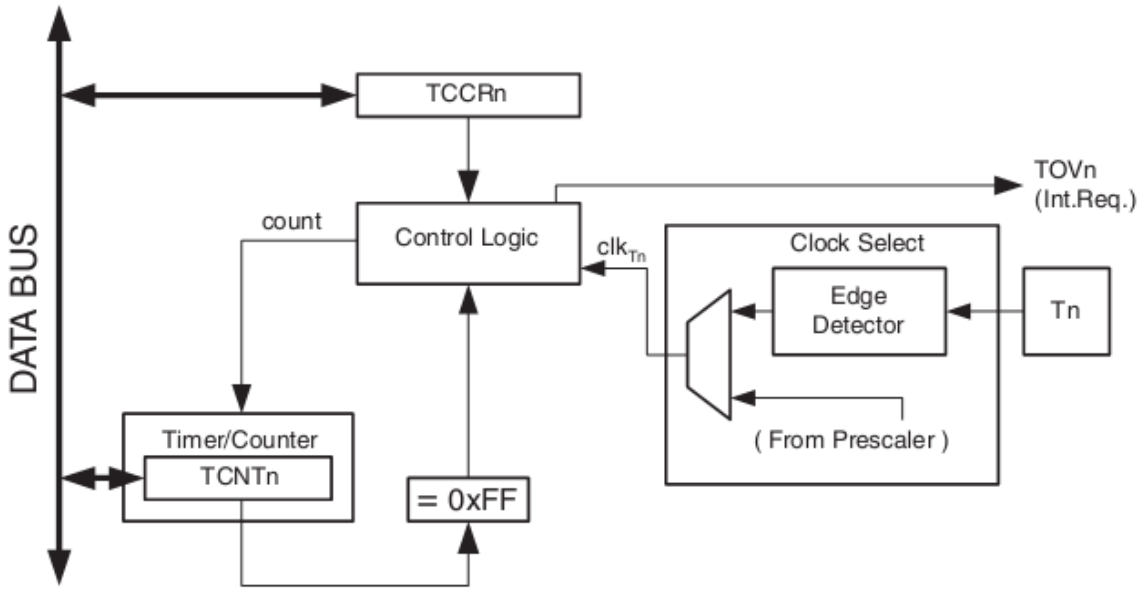
Experimento 3 – Temporizadores e contadores

Fundamentação

Os temporizadores ou contadores são circuitos do microcontrolador responsáveis por contar eventos. Estes eventos podem ter origem em pinos externos, caracterizando assim o circuito como contador, ou podem ter origem no circuito de *clock* do próprio microcontrolador, caracterizando assim o circuito como temporizador. Assim sendo um temporizador nada mais é do que um contador que conta o tempo. O microcontrolador ATmega8 possui três temporizadores / contadores, cada um com características únicas. Neste roteiro será utilizado o temporizador / contador 0, que é composto por um contador de 8 bits. O temporizador / contador 0 possui também um divisor de frequência, que pode ser utilizado para dividir o *clock* por um determinado valor entes que este sinal seja aplicado ao contador. Este circuito é conhecido como *prescaler*. A figura a seguir apresenta um diagrama de blocos do temporizador / contador 0.



O temporizador / contador 0 possui um registrador de controle chamado TCCR0, onde são realizados os ajustes deste circuito. A tabela a seguir apresenta a função de cada um dos bits deste registrador.

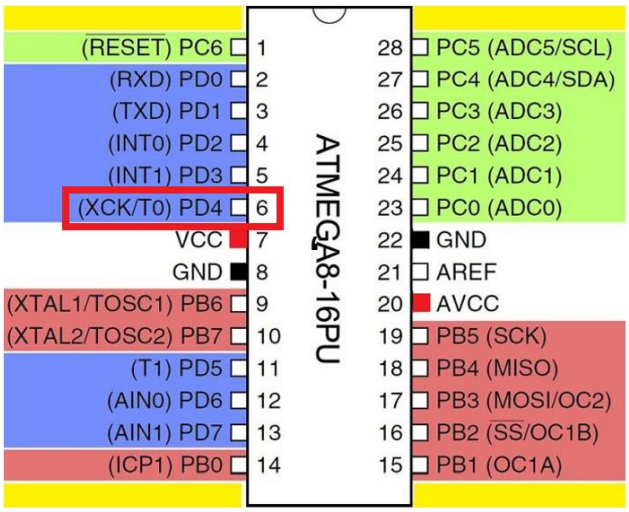
Registrador TCCR0	
Bit	**Significado**
0	CS00 – Bit de seleção de clock 0
1	CS01 – Bit de seleção de clock 1
2	CS02 – Bit de seleção de clock 2
3	–
4	–
5	–
6	–
7	–

Observe que apenas três bits deste registrador são utilizados. Com estes três bits é possível ajustar o temporizador / contador 0 para operar como temporizador, utilizando como fonte de sinal o *clock* do sistema, ou como contador, utilizando a entrada T0. Colocando estes três bits de controle em 0 o contador / temporizador 0 permanece desligado. O sinal de *clock* pode ser recebido diretamente do sistema ou derivado de um divisor (prescaler). O sinal externo, obtido do pino T0, pode ativar a contagem na borda de subida ou descida. A tabela a seguir mostra os possíveis modos de operação deste circuito.

CS02	CS01	CS00	Significado
0	0	0	Sem pulsos de contagem (Contador desligado)
0	0	1	Pulsos direto do clock do microcontrolador
0	1	0	Pulsos do clock do microcontrolador dividido por 8
0	1	1	Pulsos do clock do microcontrolador dividido por 64
1	0	0	Pulsos do clock do microcontrolador dividido por 256
1	0	1	Pulsos do clock do microcontrolador dividido por 1024
1	1	0	Pulsos do pino externo T0, na borda de descida
1	1	1	Pulsos do pino externo T0, na borda de subida

É importante lembrar que o *clock* do sistema pode variar conforma as características da aplicação, mas o valor padrão de fábrica é 1 MHz.

Quando operando como contador, a contagem é feita com base em um sinal externo aplicado ao pino T0. A figura a seguir destaca esta entrada.



O temporizador/contador 0 pode gerar uma interrupção quando chega ao final da contagem, para isso, basta configurar o registrador TIMSK, colocando em 1 o bit destacado na tabela a seguir.

Registrador TIMSK	
Bit	**Significado**
0	TOIE0 – Interrupção de estouro do temporizador 0
1	-
2	TOIE1 – Interrupção de estouro do temporizador 1
3	OCIE1B – Interrupção do comparador B do temporizador 1

4	OCIE1A – Interrupção do comparador B do temporizador 1
5	TICIE1 – Interrupção de captura externa do temporizador 1
6	TOIE1 – Interrupção de estouro do temporizador 2
7	OCIE2 – Interrupção do comparador do temporizador 2

O vetor de interrupção utilizado com o contador / temporizador 0 se chama **TIMER0_OVF_vect**.

Como se trata de um contador / temporizador de 8 bits, a contagem está limitada ao intervalo de 0 a 255. O registrador que armazena o valor desta contagem se chama TCNT0.

O circuito do contador / temporizador 0 foi projetado para executar a interrupção sempre que o valor da contagem ultrapassar o limite máximo de 255, ou seja, na passagem de 255 para 0. Desta forma, se desejarmos contar um número de eventos menor que 256, devemos carregar um valor inicial apropriado no registrador TCNT0, como apresentado na expressão a seguir.

$$TCNT0 = 256 - N_{Pulsos}$$

Onde N_{Pulsos} é o número de eventos desejado.

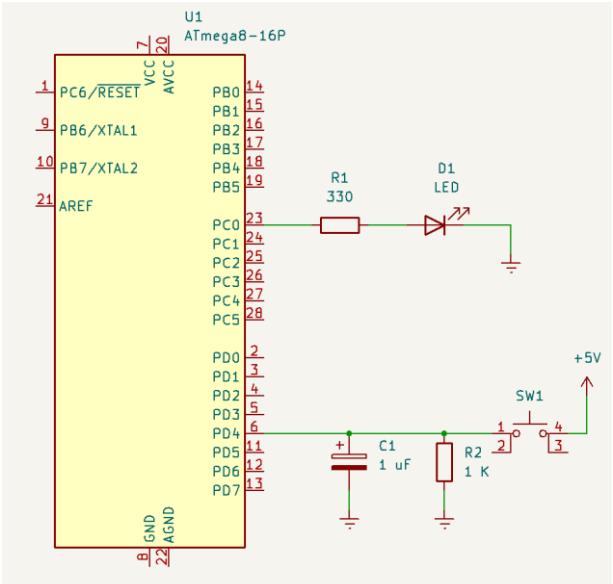
Já quando usamos o temporizador / contador 0 no modo de temporização a expressão para o valor de recarga é:

$$TCNT0 = 256 - \frac{T_{segundos} \times clock}{prescaler}$$

Onde:

T = tempo em segundos desejado.
clock = frequencia de operação do microcontrolador.
prescaler = valor do divisor de clock.

Para ilustrar o funcionamento do temporizador / contador 0 serão apresentados a seguir dois exemplos de programas, um contador e um temporizador. A figura a seguir mostra as ligações utilizadas nos exemplos.



É importante ressaltar que no circuito da figura foram desprezados os circuitos necessários para a alimentação e programação do microcontrolador, necessários para seu funcionamento. Também é importante destacar a inserção de um capacitor de 1 uF junto a entrada do sinal da chave. Este capacitor é necessário, pois como a entrada do contador é muito rápida qualquer ruido neste sinal será considerado na contagem.

Como primeiro exemplo, para o circuito da figura deseja-se um programa que acenda o LED após 10 pulsos no botão. O programa a seguir realiza esta tarefa.

```
#include <avr/io.h>
#include <avr/interrupt.h>

ISR(TIMER0_OVF_vect )
{
	PORTC|=0b00000001;
	TCCR0=0b00000000; // desabilita o temporizador / contador 0
}

int main()
{
	DDRC=0b00000001; // define PC0 como saída
	TCCR0=0b00000111; // Habilita o contador para pulsos no pino T0
	TCNT0=246; // inicia a contagem em 246 para contar 10 pulsos
	TIMSK|=1; // Habilita a interrupção do timer 0
	sei(); // ativa todas as interrupções
	// T0 é a segunda função do pino PD4
	while(1)
	{
	}
}
```

Como segundo exemplo, para o mesmo circuito deseja-se fazer um programa que pisque o led a 5 Hz. O programa a seguir faz isso.

```
#include <avr/io.h>
#include <avr/interrupt.h>

ISR(TIMER0_OVF_vect )
{
	TCNT0=158; // reinicia a contagem em 158
	if((PINC&0b00000001)==0)
	{
		PORTC|=0b00000001;
	}
	else
	{
		PORTC&=0b11111110;
	}
}

int main()
{
	DDRC=0b00000001; // define PC0 como saída
	TCCR0=0b00000101; // Habilita o timer 0 com prescaler 1024
	TCNT0=158; // inicia a contagem em 158
	TIMSK|=1; // Habilita a interrupção do timer 0
	sei(); // ativa todas as interrupções

	while(1)
	{
	}
}
```

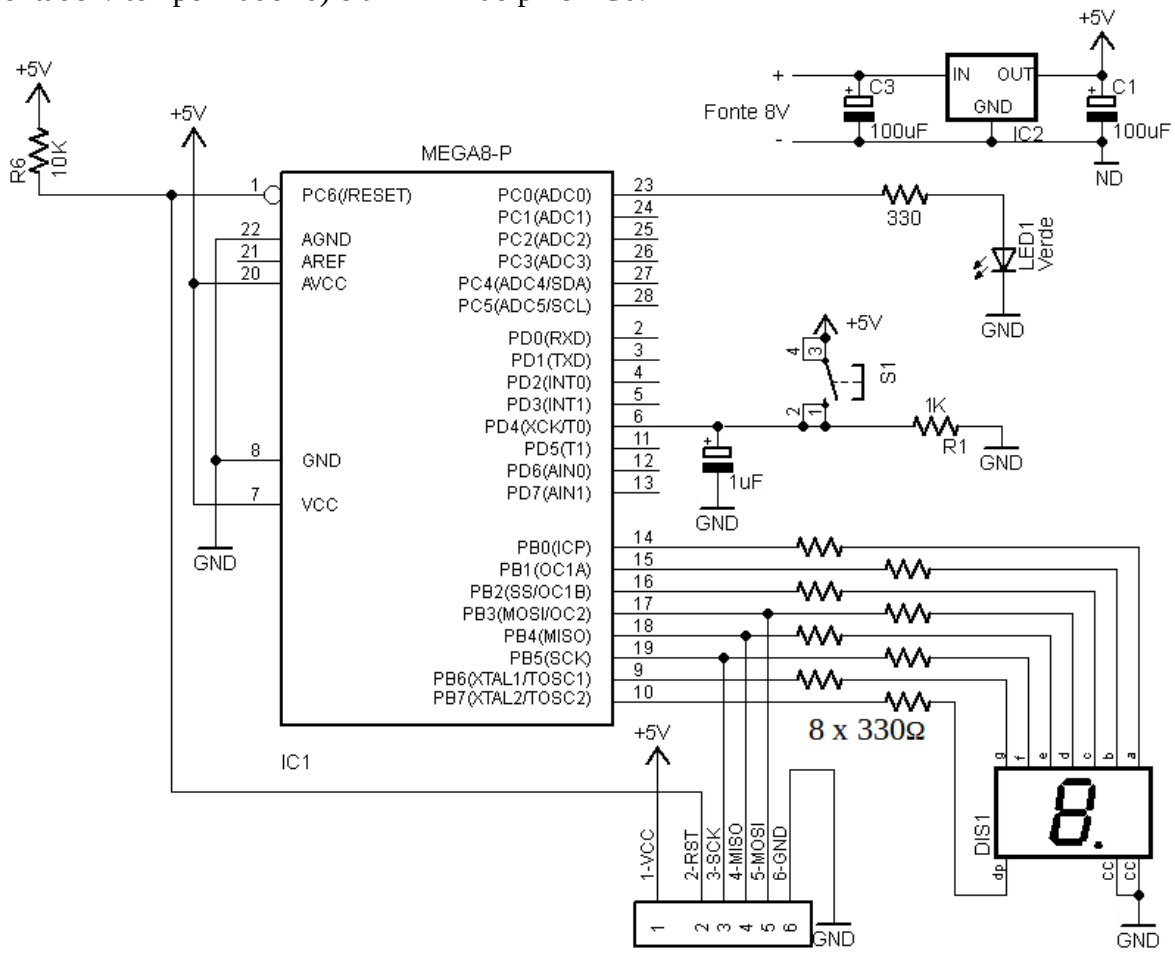
Parte experimental

Introdução

O objetivo deste experimento é exercitar com os alunos a montagem de circuitos microcontrolados que utilizem temporizadores e contadores, bem como a programação relacionada e estes periféricos. Leia atentamente todo o roteiro antes de realizar os experimentos.

Experimento Parte 1

Construa o circuito conforme diagrama a seguir, conectando um botão ao pino PD4 (entrada do contador / temporizador 0) e um LED ao pino PC0.



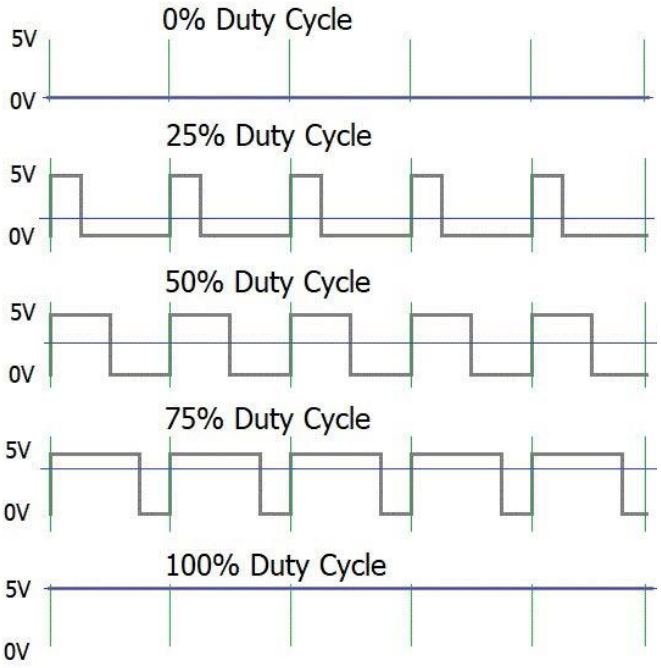
Para este circuito faça as seguintes implementações:

A. Faça um programa utilizando o temporizador / contador 0 que conta 15 pulsos na entrada T0 e quando termina acende um LED na porta PC0. Monte o circuito e transfira o programa para o microcontrolador, comprovando seu funcionamento.
B. Faça um programa utilizando o contador / temporizador 0 que pisca um led na porta C0 em uma frequência de 3 Hz e teste seu funcionamento.
C. Mude o programa para que o led pisque em 1 Hz e repita o teste. Será necessário usar mais de uma interrupção do timer para chegar a este período.
D. Utilizando o display de 7 segmentos faça um programa para que o número mostrado no display seja incrementado de 1 em 1 segundos (de 0 a 9) utilizando o temporizador para contar este tempo. Comprove seu funcionamento.

Experimento Parte 2

Os temporizadores têm várias aplicações nos microcontroladores, uma delas é gerar sinais modulados por largura de pulso, do inglês PWM.

Sinais PWM são sinais digitais oscilantes no tempo em que a relação entre o tempo ligado e desligado obedece a uma proporção pré-determinada. Veja a figura a seguir.

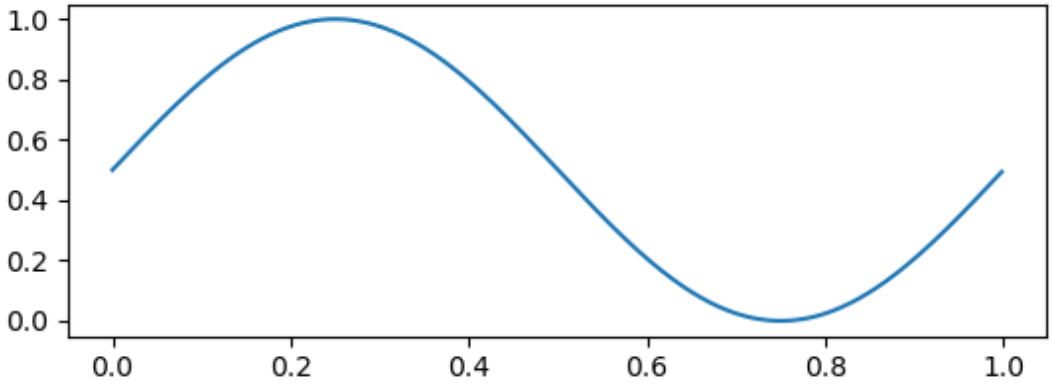


Com a variação da relação entre o tempo ligado e desligado é possível obter uma tensão média ajustável, simulando assim uma saída analógica. Este tipo de sinal é utilizado por exemplo para variar o brilho de uma lâmpada, o calor de um aquecedor, a rotação de um motor etc.

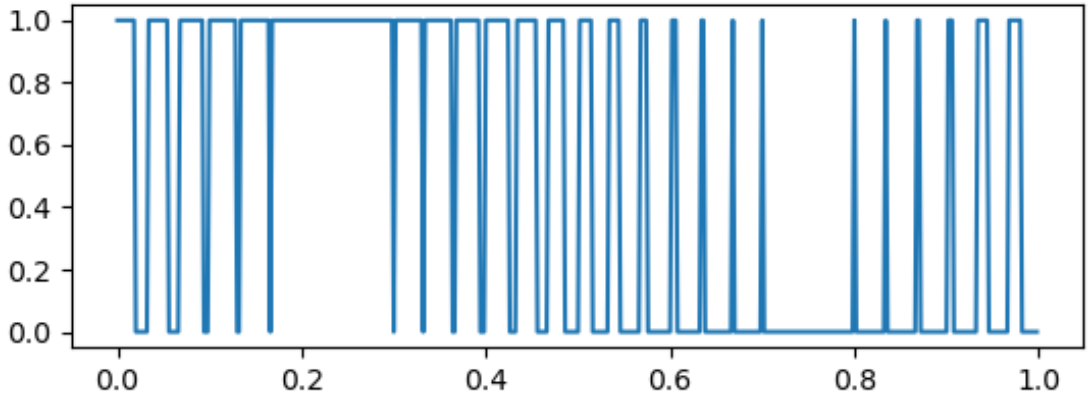
Os contadores / temporizadores dos microcontroladores podem ser utilizados para gerar este tipo de sinais.

Utilizando o contador / temporizador 0 do microcontrolador no circuito da parte experimental 1 deste roteiro o brilho do LED pode ser ajustado simplesmente variando rapidamente o tempo ligado e desligado do pino PC0.

É possível elaborar um programa que varia o brilho do LED senoidalmente, obedecendo o gráfico da figura a seguir.



Para elaborar este programa o contador / temporizador deve ser programado como temporizador, com um intervalo de tempo variável, de forma a gerar um sinal PWM. A figura a seguir mostra como se parece o sinal gerado pelo temporizador.



Observe que a saída fica mais tempo ligada na parte superior da senoide, e mais tempo desligada na parte inferior da senoide.

Para que o brilho do LED varie senoidalmente com uma frequência de 1 Hz, deve-se ajustar os tempos do temporizador de forma a gerar pulsos com uma frequência muito mais alta. A lista a seguir apresenta estes valores.

- Frequência da senoide: 1 Hz.
- Frequência base do sinal PWM 100 Hz.

Se você leu o roteiro até aqui faça o seguinte: implemente apenas a parte 1 do experimento, a parte 2 é apenas ilustrativa, não deve ser implementada, serve apenas como um teste da atenção dos alunos. Não comente sobre este parágrafo com ninguém.

Relatório

Após a realização dos experimentos deve ser elaborado um relatório seguindo o modelo disponibilizado. Este relatório deve ser submetido via sistema SIGAA para avaliação, em formato PDF, até a data estipulada em aula.

O modelo do relatório pode ser encontrado em: https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/microcontroladores-experimental/

Serão avaliados os seguintes itens no relatório:

- Introdução
- Objetivo
- Fundamentação teórica
- Desenvolvimento
- Componentes utilizados
- Circuito eletrônico
- Código fonte do programa
- Resultados e discussões
- Conclusão