

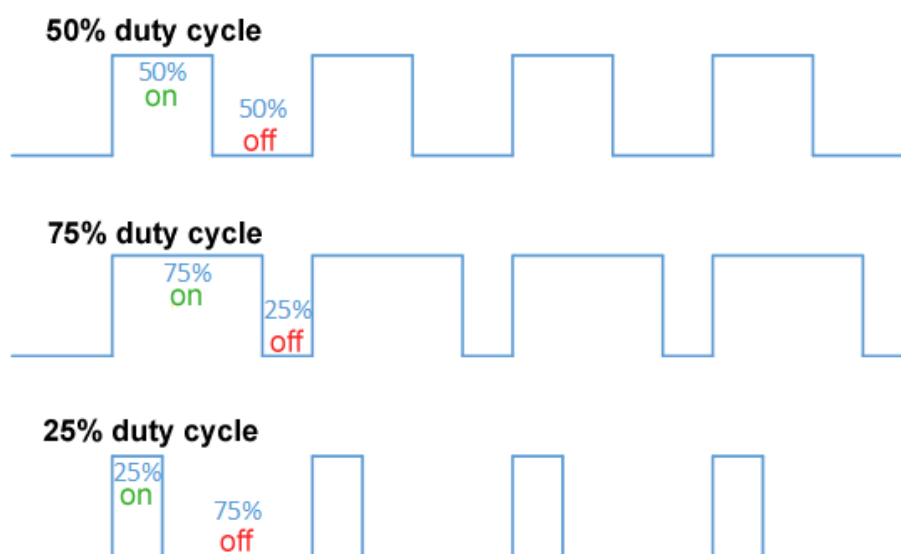
Experimento 4 – Temporizador em modo PWM e conversor A/D

Fundamentação

Temporizadores em modo PWM

O temporizador / contador 1 é um hardware mais complexo que os anteriores, apresentando vários modos de operação. Além dos modos clássicos, como temporizador e contador, este hardware possui alguns modos de operação destinados a modulação por largura de pulsos (PWM em inglês).

Os sinais PWM permitem emular uma saída analógica utilizando uma saída digital. Assim é possível entregar uma tensão média na carga diferente dos valores digitais tradicionais. Observe a figura a seguir.



Nesta figura é possível observar três sinais PWM, o primeiro com uma tensão média de 50%, o segundo com 75% e o terceiro com 25%.

Apesar de ser possível criar sinais PWM utilizando o software do microcontrolador a forma mais apropriada de fazê-lo é utilizar os contadores / temporizadores dedicados a esse fim.

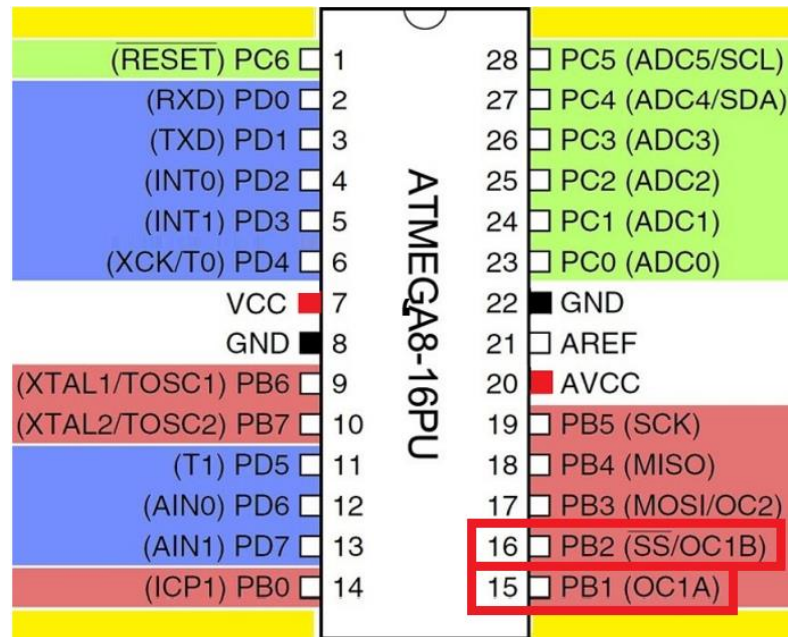
Características do temporizador / contador 1:

O temporizador / contador 1 possui as seguintes características:

- O contador utilizado pelo temporizador / contador 1 é de 16 bits.
- Possui duas unidades de comparação independentes
- Possui uma unidade de captura externa
- Filtro de ruído na entrada
- Recarga automática
- Função PWM
- Quatro Fontes de interrupção independentes

Para ajustar o temporizador / contador 1 em modo PWM será necessário configurar os valores corretos em seus registradores de controle. No caso do ATmega8, existem alguns pinos dedicados a função PWM, e conectados diretamente aos temporizadores.

A figura a seguir destaca os pinos PB1 e PB2, que desempenham as funções OC1A e OC1B e estão conectados ao temporizador / contador 1.



Os registradores de controle do temporizador / contador 1 são os seguintes.

Registrador TCCR1A	
Bit	Significado
0	WGM10 Modo de operação
1	WGM11 Modo de operação
2	FOC1B Força comparação no canal B
3	FOC1A Força comparação no canal A
4	COM1B0 Modo de comparação B
5	COM1B1 Modo de comparação B
6	COM1A0 Modo de comparação A
7	COM1A1 Modo de comparação A

Modo de comparação não PWM.

COM1A1/ COM1B1	COM1A0/ COM1B0	Significado
0	0	Modo normal de operação dos pinos, OC1A e OC1B (desconectados)
0	1	Inverte os sinais dos pinos OC1A e OC1B na comparação
1	0	Envia zero aos pinos OC1A e OC1B na comparação
1	1	Envia um aos pinos OC1A e OC1B na comparação

Modo de comparação PWM rápido.

COM1A1/ COM1B1	COM1A0/ COM1B0	Significado
0	0	Modo normal de operação dos pinos, OC1A e OC1B (desconectados)
0	1	WGM13:0 = 15: inverte OC1A na comparação, OC1B fica desconectado (opera normalmente como I/O). Para qualquer outra combinação de WGM1 os sinais OC1A/OC1B são desconectados (operam normalmente como I/O).
1	0	Zera OC1A/OC1B na comparação, seta OC1A/OC1B quando a contagem chega a zero.
1	1	Seta OC1A/OC1B na comparação, zera OC1A/OC1B quando a contagem chega a zero.

Nos modos de comparação e saída, PWM de fase correta e PWM de frequência correta.

COM1A1/ COM1B1	COM1A0/ COM1B0	Significado
0	0	Modo normal de operação dos pinos, OC1A e OC1B (desconectados)
0	1	WGM13:0 = 9 ou 14: Alterna OC1A na comparação, OC1B desconectado (operando como I/O). Para todas as outras combinações de WGM1, OC1A/OC1B desconectados (operando como I/O).
1	0	Zera OC1A/OC1B na comparação quando contando de forma crescente e seta OC1A/OC1B na comparação quando contando de forma decrescente.
1	1	Seta OC1A/OC1B na comparação quando contando de forma crescente e zera OC1A/OC1B na comparação quando contando de forma decrescente.

O segundo registrador de controle do temporizador / contador 1 é o registrador TCCR1B, a tabela a seguir mostra a função de cada um dos seus bits.

Registrador TCCR1B	
Bit	Significado
0	CS10 Seleção da entrada de clock
1	CS11 Seleção da entrada de clock
2	CS12 Seleção da entrada de clock
3	WGM12 Modo de operação
4	WGM13 Modo de operação
5	-
6	ICES1 seletor de borda da entrada de captura
7	ICNC1 Redutor de ruídos da entrada de captura

A tabela a seguir apresenta a função dos bits dos registradores TCCRA e TCCR1B, responsáveis pela definição do modo de operação.

Modo	WGM13	WGM12	WGM11	WGM10	Modo de operação	Topo	Atualização de OCR1x	Liga o bit TOV1
0	0	0	0	0	Normal	0xFFFF	Imediato	Máximo
1	0	0	0	1	PWM fase correta 8 bits	0x00FF	Topo	Base
2	0	0	1	0	PWM fase correta 9 bits	0x01FF	Topo	Base
3	0	0	1	1	PWM fase correta 10 bits	0x03FF	Topo	Base
4	0	1	0	0	CTC	OCR1A	Imediato	Máximo
5	0	1	0	1	PWM rápido 8 bits	0x00FF	Base	Topo
6	0	1	1	0	PWM rápido 9 bits	0x01FF	Base	Topo
7	0	1	1	1	PWM rápido 10 bits	0x03FF	Base	Topo
8	1	0	0	0	PWM fase e frequência correta	ICR1	Base	Base
9	1	0	0	1	PWM fase e frequência correta	OCR1A	Base	Base
10	1	0	1	0	PWM fase correta	ICR1	Topo	Base
11	1	0	1	1	PWM fase correta	OCR1A	Topo	Base
12	1	1	0	0	CTC	ICR1	Imediato	Máximo
13	1	1	0	1	(Reservado)	-	-	-
14	1	1	1	0	PWM rápido	ICR1	Base	Topo
15	1	1	1	1	PWM rápido	OCR1A	Base	Topo

Os bits 0, 1 e 2 do registrador TCCR1B são responsáveis por definir a fonte do sinal de entrada do contador conforme a tabela a seguir.

CS12	CS11	CS10	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 T1, na borda de descida
1	1	1	Pulsos do pino externo T1, na borda de subida

Assim como os outros periféricos deste microcontrolador, o temporizador / contador 1 utiliza interrupções para chamar sub-rotinas de software que ajudam em seu funcionamento. A tabela a seguir descreve o registrador TIMSK, destacando o funcionamento dos bits utilizados no temporizador / contador 1.

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	TOIE2 – Interrupção de estouro do temporizador 2
7	OCIE2 – Interrupção do comparador do temporizador 2

Exemplo de programa:

O exemplo de programa seguir configura o temporizador 1 para gerar um sinal PWM de 1 KHz com razão cíclica de 50%. Este exemplo serve como base para os experimentos a seguir.

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

```
int main(void)
```

```
{
```

```
    DDRB |= (1<<PB1); // Configura o pino OC1A (PB1) como saída
```

```
    ICR1 = 999; // Divide o clock por 1000 (ICR1+1) para um PWM de 1 KHz
```

```
    OCR1A=500; // Ajusta o PWM inicial para 50%
```

```
    TCCR1A = (1 << WGM11) | (1 << COM1A1); // Ajusta para o modo PWM Rápido
```

```
    TCCR1B = (1 << WGM12) | (1 << WGM13) | (1 << CS10); // Ajusta o prescaler para 1
```

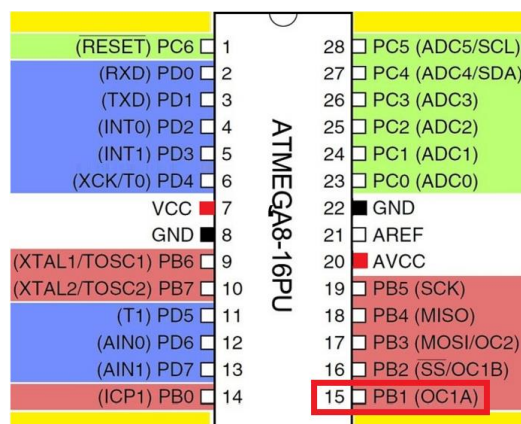
```
    while (1)
```

```
    {
```

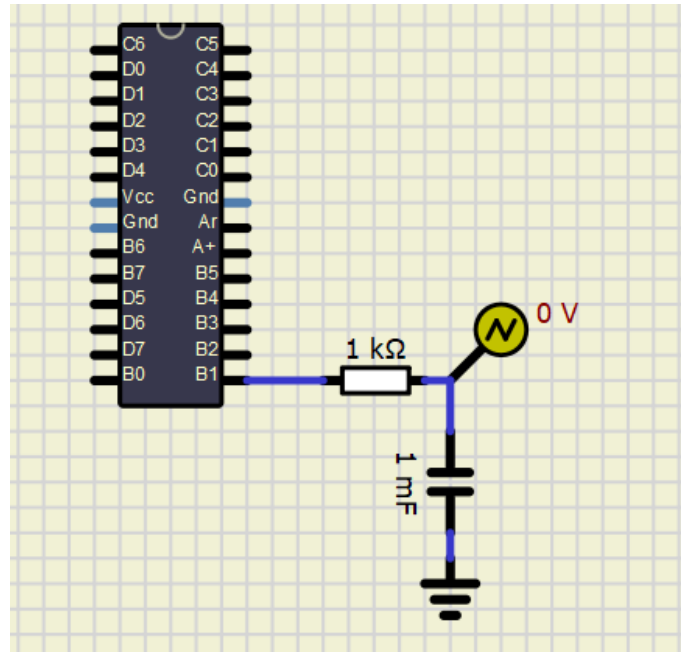
```
    }
```

```
}
```

Este exemplo utiliza o modo PWM rápido onde a saída é enviada para o pino OC1A (PB1) do microcontrolador. Veja a figura a seguir.



Um exemplo de circuito que utiliza um resistor e um capacitor é apresentado a seguir. Neste circuito é possível observar que o capacitor se carrega com uma tensão proporcional a razão cíclica do PWM.



Conversor analógico digital

A família de microcontroladores AVR apresenta internamente um conversor analógico digital, capaz de converter sinais analógicos de tensão em uma palavra digital de 10 bits. Este conversor apresenta as seguintes características:

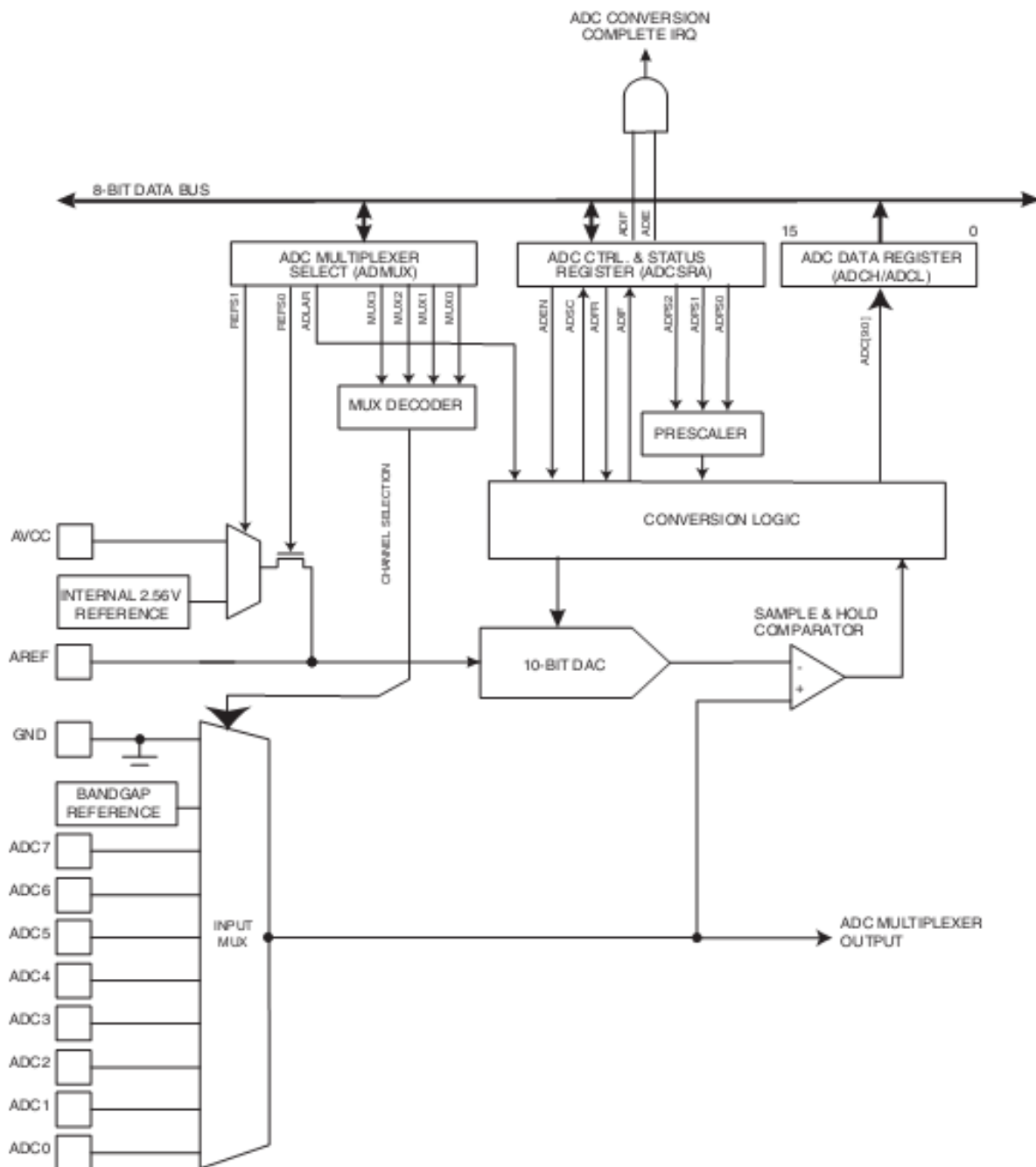
- 10 bits de resolução
- Linearidade de 0,5 LSB
- Precisão absoluta de ± 2 LSB
- Tempo de conversão de 13 - 260 μ s
- Até 15K amostras por segundo em resolução máxima
- 6 canais multiplexados
- Opção de trabalhar em 8 bits
- Tensão de entrada de 0 - VCC
- Referência interna opcional de 2,56V
- Interrupção de final de conversão

O conversor A/D da família AVR tem sua alimentação separada do resto do microcontrolador, assim para utilizarmos o conversor A/D devemos conectar o pino AVCC a alimentação positiva e o pino AGND a alimentação negativa. A alimentação do conversor A/D é separada para que possamos fornecer uma alimentação limpa de ruídos, aumentando assim a qualidade das medidas realizadas.

Este conversor A/D pode utilizar três diferentes referências de tensão, que são:

- Referência externa conectada no pino AREF
- Referência conectada a alimentação do conversor A/D (Pino AVCC)
- Referência interna de 2,56V

A figura a seguir apresenta um diagrama de blocos deste conversor analógico digital.



É possível escolher qual a referência para a conversão e qual o canal de entrada através dos registradores de controle do A/D. É também necessário ajustar a frequência de operação (clock) do A/D através da configuração de seu prescaler. É importante salientar que em resolução máxima (10 bits), o conversor A/D não pode operar em uma frequência superior a 200KHz. Os registradores que permitem estes ajustes serão abordados mais a frente.

O resultado da conversão é armazenado em um registrador de 10 bits, e seu valor depende de dois fatores, da tensão na entrada e da referência utilizada no conversor A/D. A fórmula a seguir mostra como calcular o valor de saída do conversor A/D.

$$ADC = \frac{V_{entrada} \times 1024}{V_{ref}}$$

No programa o resultado da conversão pode ser acessado através do registrador ADCW.

O restante dos ajustes do conversor A/D é feito através dos registradores ADMUX e ADCSRA, que serão vistos a seguir. As tabelas a seguir apresentam as funções dos bits do registrador ADCMUX.

Registrador ADCMUX	
Bit	Significado
0	MUX0 – bit 0 do multiplexador
1	MUX1 – bit 1 do multiplexador
2	MUX2 – bit 2 do multiplexador
3	MUX3 – bit 3 do multiplexador
4	–
5	ADLAR – Ajuste da saída do A/D
6	REFS0 – bit 0 da seleção de referência
7	REFS1 – bit 1 da seleção de referência

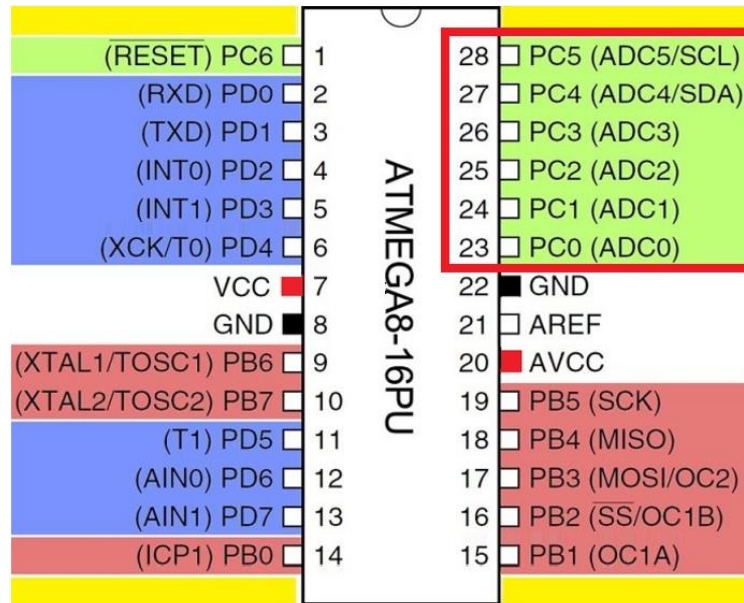
Os primeiros 4 bits são responsáveis por selecionar a entrada analógica, a tabela a seguir detalha seu funcionamento.

MUX3	MUX2	MUX1	MUX0	Significado
0	0	0	0	ADC0
0	0	0	1	ADC1
0	0	1	0	ADC2
0	0	1	1	ADC3
0	1	0	0	ADC4
0	1	0	1	ADC5
0	1	1	0	ADC6
0	1	1	1	ADC7
1	0	0	0	-
1	0	0	1	-
1	0	1	0	-
1	0	1	1	-
1	1	0	0	-
1	1	0	1	-
1	1	1	0	1,3V interno
1	1	1	1	0V interno

O bit 5 diz respeito a forma que o resultado é armazenado, e se for necessário o manual deve ser consultado. Já os bits 6 e 7 são responsáveis pela escolha da entrada referência do conversor A/D. A tabela a seguir mostra o significado de cada bit.

REFS1	REFS0	Significado
0	0	Referência externa no pino AREF
0	1	Referência externa no pino AVCC (Utilizar filtro em AREF)
1	0	-
1	1	Referência interna de 2,56 (Utilizar filtro em AREF)

Os pinos responsáveis pelas entradas analógicas no ATmega8 estão destacados na figura a seguir.



Outro registrador envolvido no controle do conversor A/D é o registrador ADCSRA. A tabela a seguir mostra o significado de cada um dos seus bits.

Registrador ADCSRA	
Bit	Significado
0	ADPS0 – bit 0 do prescaler do A/D
1	ADPS1 – bit 1 do prescaler do A/D
2	ADPS2 – bit 2 do prescaler do A/D
3	ADIE – habilita a interrupção do A/D
4	ADCIF – Indicador de interrupção
5	ADFR – Modo de conversão
6	ADSC – Inicia a conversão
7	ADEN – habilita o conversor A/D

Os bits 0, 1 e 2 são responsáveis pela escolha do prescaler, conforma tabela a seguir.

ADPS2	ADPS1	ADPS0	Significado
0	0	0	Divide o clock por 2
0	0	1	Divide o clock por 2
0	1	0	Divide o clock por 4
0	1	1	Divide o clock por 8
1	0	0	Divide o clock por 16
1	0	1	Divide o clock por 32
1	1	0	Divide o clock por 64
1	1	1	Divide o clock por 128

O bit 3 habilita a interrupção de final de conversão do conversor A/D. O bit 4 indica quando a conversão acabou e o valor já é válido. O bit 5 serve para escolhermos se as conversões devem acontecer sucessivamente, ou se devemos iniciar cada conversão manualmente. O bit 6 inicia a conversão. E finalmente o bit 7 habilita o conversor A/D.

Exemplo de programa

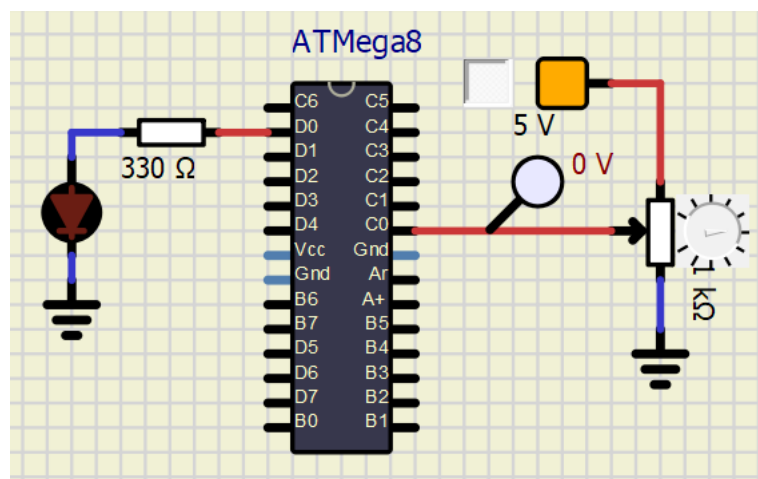
A seguir temos um exemplo de programa que utiliza o conversor A/D do microcontrolador.

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

ISR(ADC_vect)
{
    if(ADCW>512)
    {
        PORTD |= 0b00000001;
    }
    else
    {
        PORTD &= 0b11111110;
    }
    ADCSRA |= 0b01000000; // inicia nova conversão
}

int main()
{
    DDRD = 0b00000001; // define PD0 como saída
    ADMUX = 0b01000000; // escolhe a entrada analógica 0, com a referência no pino AVCC
    ADCSRA = 0b11001111; // habilita o A/D com prescaler de 128, habilitando sua interrupção.
    sei(); // ativa todas as interrupções
    while(1)
    {
    }
}
```

Um exemplo de circuito para este programa é apresentado na figura a seguir.



O programa do exemplo habilita o conversor A/D e sua interrupção, faz sua leitura e compara o valor lido do conversor A/D, que varia entre 0 e 1023, com o número 512, se o valor for maior que 512 o LED no pino PD0 é ligado, caso contrário é desligado.

Também é possível utilizar o conversor A/D sem as interrupções, porém nestes casos o programa deve ficar aguardando o final da conversão em um laço de repetição. A seguir temos um exemplo deste método.

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

```
int leAD(char canal) // lê o conversor AD sem usar interrupção
```

```
{
    ADMUX=0x40|(canal & 0x0f);    //seleciona o canal e a referência no AVCC
    ADCSRA = 0b11000111;        //Liga o AD e inicia a conversão
    while((ADCSRA&(1<<ADIF))==0); // Aguarda o final da conversão
    return(ADCW);                // retorna o resultado
}
```

```
int main(void)
```

```
{
    DDRB = (1<<PB0);
    int entrada;

    while (1)
    {
        entrada = leAD(0); // Armazena o valor da entrada analógica 0 em entrada.
        if(entrada>100)
        {
            PORTB|=(1<<PB0); // liga PB0
        }
        else
        {
            PORTB&=~(1<<PB0); //desliga PB0
        }
    }
}
```

Parte experimental

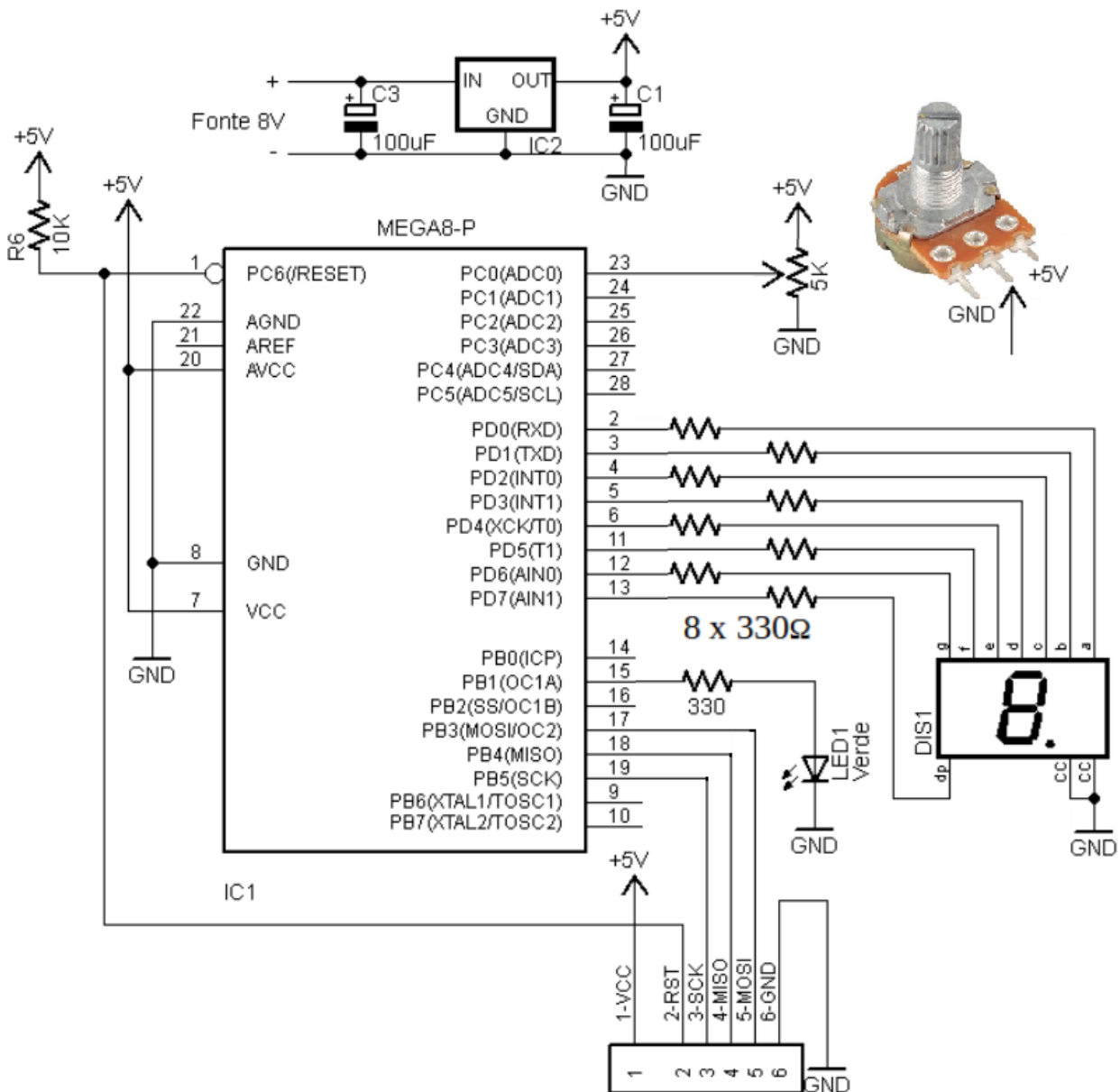
Introdução

O objetivo deste experimento é desenvolver práticas utilizando temporizadores para gerar sinais com modulação PWM, bem como desenvolver experimentos que utilizem o conversor analógico digital. Assim é possível entender como um microcontrolador pode processar entradas e saídas analógicas.

Experimento

Para o circuito da figura a seguir faça um programa que lê a tensão na entrada ADC0 do conversor A/D e ajusta o PWM para que a tensão média no LED verde seja equivalente a tensão na saída do potenciômetro.

Conforme o potenciômetro é girado o brilho do LED deve aumentar ou diminuir, indo de 0 a 100%.



O display de 7 segmentos conectado à porta D deve apresentar o valor da tensão de entrada (0 a 5V). O valor apresentado deve seguir a tabela a seguir.

Tensão	Display
0,0-0,5V	0
0,5-1,5V	1
1,5-2,5V	2
2,5-3,5V	3
3,5-4,5V	4
4,5-5,0V	5

Observação: utilize o osciloscópio para visualizar o sinal do PWM.

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