Conversão A/D – D/A com microcontrolador e filtro

Programação de microcontrolador para conversão AD e modulação PWM

Introdução

Os microcontroladores consistem em um único circuito integrado que apresenta um núcleo de processador, memórias (voláteis e não voláteis) e determinados periféricos de entrada e de saída de informações. Dentre as capacidades do microcontrolador se encontra, devido a presença de um conversor analógico-digital, a possibilidade de converter sinal analógico em digital.

Uma ferramenta extremamente importante, processar e converter informações é essencial, sobretudo quando se verifica a gama de variáveis analógicas e a capacidade de processamento digital dos computadores atuais. Dessa forma, uma seleta parte desse trabalho busca evidenciar essa característica dos microcontroladores.

Ademais, utilizando o ATMega 328, busca-se, através dessa breve pesquisa, explorar a capacidade de gerar um sinal PWM. O PWM (Pulse Width Modulation) ou Modulação de Largura de Pulso é empregado quando se almeja simular um sinal analógico a partir de um sinal digital. Esse mecanismo é possível quando modula-se a largura do pulso de um sinal digital, ou seja, quando se controla quanto tempo a cada período o sinal digital vai ficar em nível alto e por quanto tempo vai permanecer em nível baixo.

Avaliação da conversão AD

Com o objetivo de estudar sobre a conversão A/D, buscou-se, a priori, realizar uma leitura analógica através da função analogRead, por meio da porta analógica A0. Usando como suporte o código a seguir, realizou-se a digitalização da entrada V_{in} promovendo uma varredura de 0 a 5V com um passo de 0.5V

Código para realizar a leitura analógica

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
}
```

Como mecanismo de alterar a entrada analógica optou-se pela utilização de um potenciômetro que representa uma variável analógica que pode assumir diversos valores. Portanto, ao variar o potenciômetro (0 a 100%), consegue-se realizar uma varredura de 0.5 V. O circuito completo pode ser visualizado na figura 1 a seguir:

Esquemático do Circuito

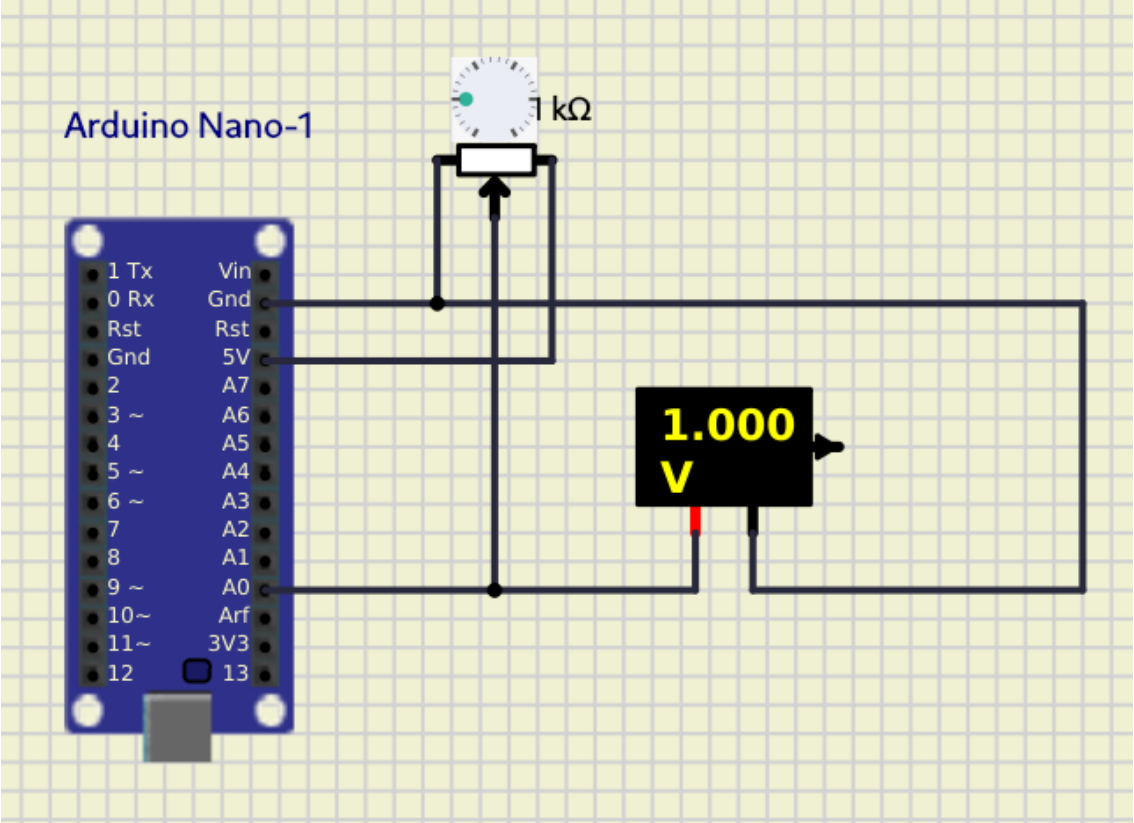


Figura 1 - Esquemático para avaliação da conversão AD

Após realizar o procedimento descrito anteriormente, é imperioso observar o resultado da digitalização da porta serial. Dessa forma, enviou-se o valor decimal da conversão para cada valor correspondente, podendo-se, inclusive realizar um gráfico V_{in} x Valor Decimal. Na figura 02 a seguir evidencia-se como foi o processo de coleta de dados por meio do monitor serial:

Saída do monitor serial

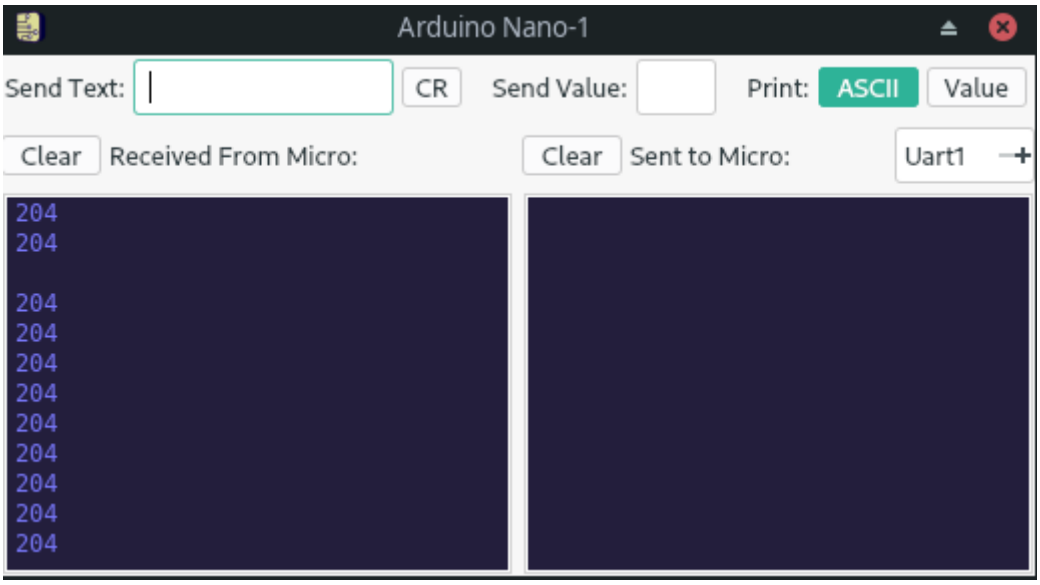


Figura 2 - Saída serial para entrada de 1.0V

Após a coleta de dados, montou-se a tabela 1 que correlaciona a entrada e o valor decimal:

Tensão de entrada $V_{in,DC}$ (V)	**Valor decimal**
0.5	102
1.0	204
1.5	306
2.0	408
2.5	511
3.0	613
3.5	715
4.0	818
4.5	920
5.0	1022

Tabela 1 - Varredura da tensão de entrada

O resultado do gráfico é elucidado na figura a seguir e como se observa, há um comportamento que se assemelha a uma função linear, ou seja, incrementos de tensão de entrada representam o acréscimo proporcional no valor decimal. Contudo, da teoria, observa-se que esse comportamento, se não fosse pela escala preterida de 0.5V, formaria um evidente gráfico digital e discretizado:

Gráfico $V_{in,DC} \times Valor\ decimal$

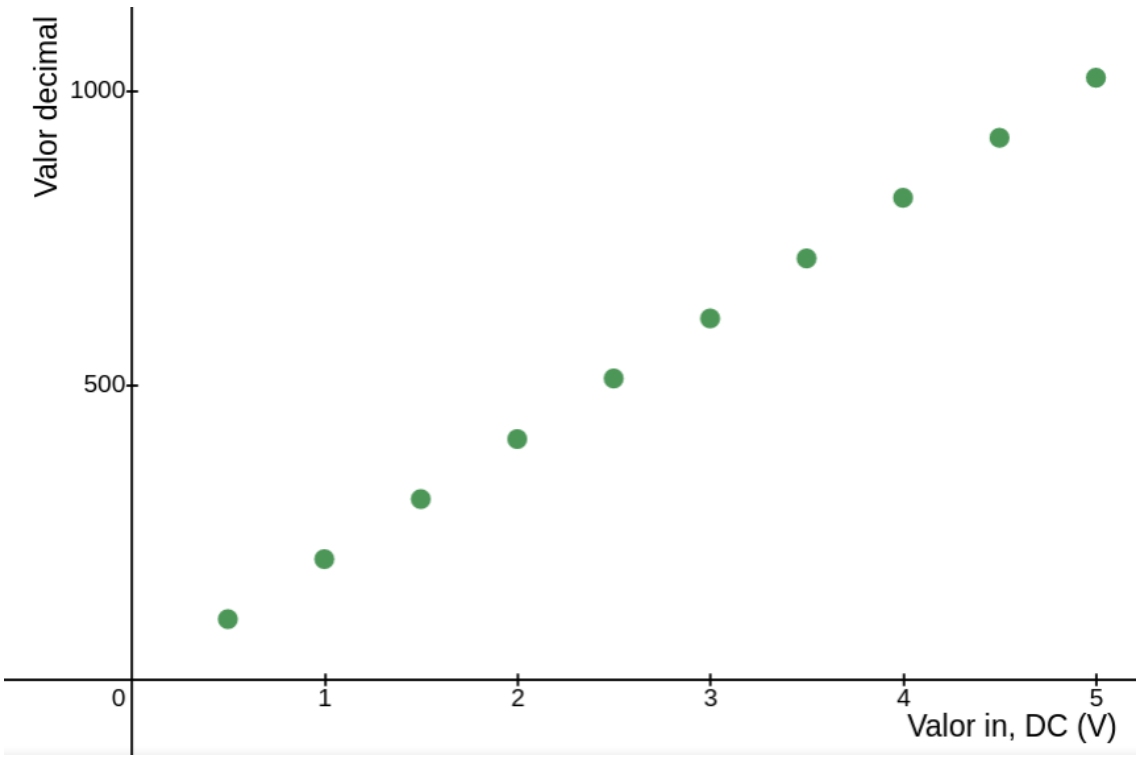


Figura 3 - Gráfico comparativo tensão de entrada e valor serial decimal

É imperioso entender ainda que, da literatura, sabe-se que a equação que relaciona o valor decimal com o valor de entrada é a equação **(1)** apresentada abaixo:

$$D = \frac{V_{IN}}{V_{REF}} 2^N \tag{1}$$

Como o arduino apresenta um conversor analógico-digital 10 bits significa que este irá mapear tensões entre 0 e a tensão operacional 5V para valores inteiros entre 0 e 1023. No Arduino UNO, por exemplo, isso permite uma resolução entre leituras de: 5 volts / 1024 unidades, ou 0,0049 volts (4,9 mV) por unidade

Avaliação da operação do PWM

O próximo passo do trabalho, após realizar a conversão AD foi, implementar a modulação por largura de pulso (PWM) através da função analogWrite, responsável por acionar uma onda PWM num pino qualquer. Além de gerar um sinal PWM, buscou-se promover uma varredura da variaűel da razão cíclica 0 a 255 com passo de 51.

Detalhe: A razão de ser de 0 a 255 está relacionada com o fato da quantidade de bits para gerar o PWM ser igual a 8 bits. A seguir verifica-se o código implementado para realizar a avaliação:

```
void setup() {
  pinMode(5,OUTPUT);
}

void loop() {
  analogWrite(5, 0); // Segundo parâmetro
                     // variado (D)
}
```

Além do código, montou-se o seguinte circuito, elucidado na figura 4, que possibilita realizar a varredura da razão cíclica com um passo de 51:

Esquemático do circuito para avaliação do PWM

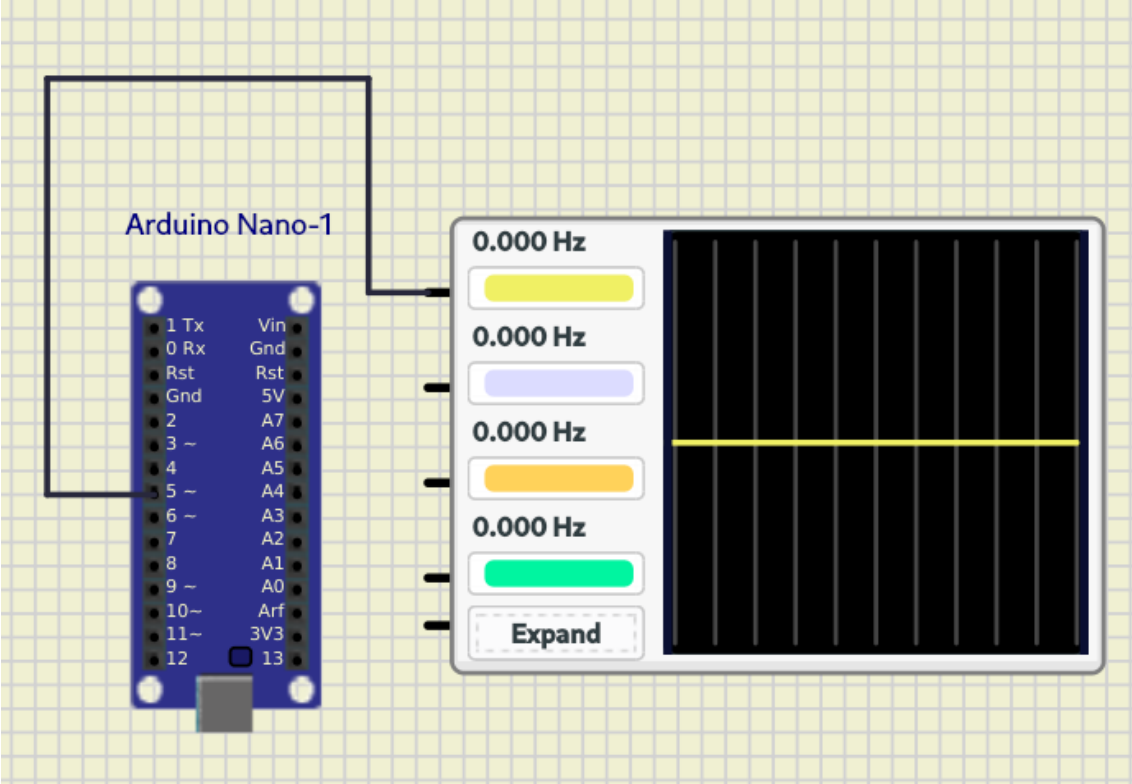


Figura 4 - Esquemático do circuito para avaliação do PWM

Resultados da simulação

Como se visualiza a seguir, com a varredura da variável que representa a razão cíclica, com D igual a 0, percebe-se que o valor de tensão é permamentente 0 ou nível baixo, enquanto que, conforme aumenta-se a variável da razão cíclica, intensifica também, a porcentagem de tempo em que a variável se encontra com um nível lógico igual a 1, até que se atinja 255 e o nível lógico é permanentemente 1. A figura 5 a seguir demonstra o efeito da varredura:

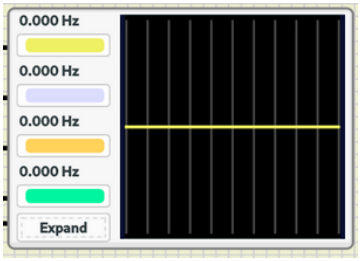
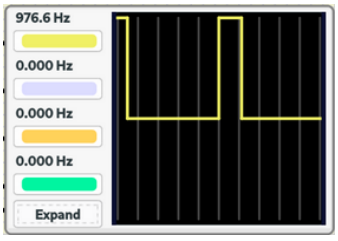
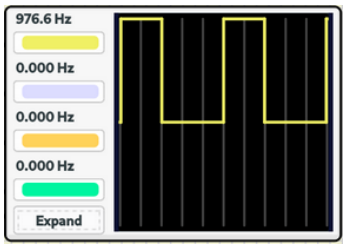
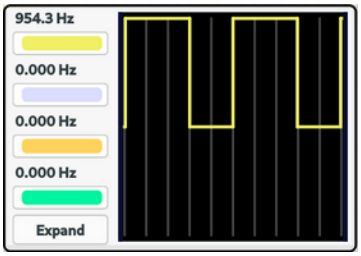
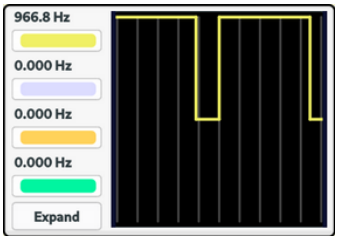
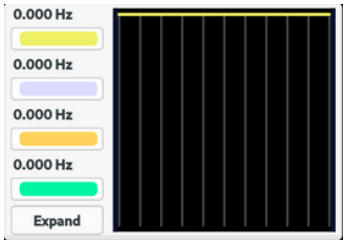
D=0	D=51	D=102
		
D=153	**D=204**	**D=255**
		

Figura 5 - Avaliação dos resultados para cada ciclo de trabalho

Para além das observações feitas acima, é fundamental realizar o cálculo do ciclo de tarefa relacionado para cada valor testado acima. Com uma frequência e período definidos, calculou-se o ciclo de tarefa relacionado:

$$\begin{cases} f = 976.7Hz \\ T = 1.02ms \end{cases} \quad (2)$$

A seguir, na tabela 2, para cada valor testado durante a varredura, determinou-se o ciclo de tarefa relacionado e, como já previsto, aumenta-se o tempo em que a variável se encontra em nível lógico alto, conforme eleva-se a varredura:

D	$t(\mu s)$	$D_{\%}$
0	0	0%
51	210.9	20.67%
102	421.9	41.36%
153	616.6	60.45%
204	820.7	80.46%
255	1020.3	100.00%

Tabela 2 - Cálculo do ciclo de trabalho

Modulação PWM de um entrada senoidal

Para realizar a modulação PWM de uma entrada senoidal, fez-se uso da função analogRead() para digitalizar uma entrada senoidal com características específicas senoidal com f = 10 Hz, Vpp = 5 V e Voffset = 2,5 V. Ademais, utilizou-se também analogWrite() para gerar um sinal PWM por meio da sinal senoidal digitalizado.

O código responsável por exercer essa ação está elucidado a seguir. Para implementar a tarefa fez-se uso da função map, esta última faz com que a leitura analógica da função senoidal (que varia de 0 a 1023) seja recebida pela variável output numa escala de 0 a 255:

```
void setup() {
  pinMode(5,1);
}

void loop() {
  int input = analogRead(A0);
  int output = map(output, 0, 1023, 0, 255);
  analogWrite(5, output);
}
```

O circuito responsável por realizar a modulação PWM de uma entrada senoidal está elucidado na figura 6 a seguir:

Esquemático do circuito para modulação PWM

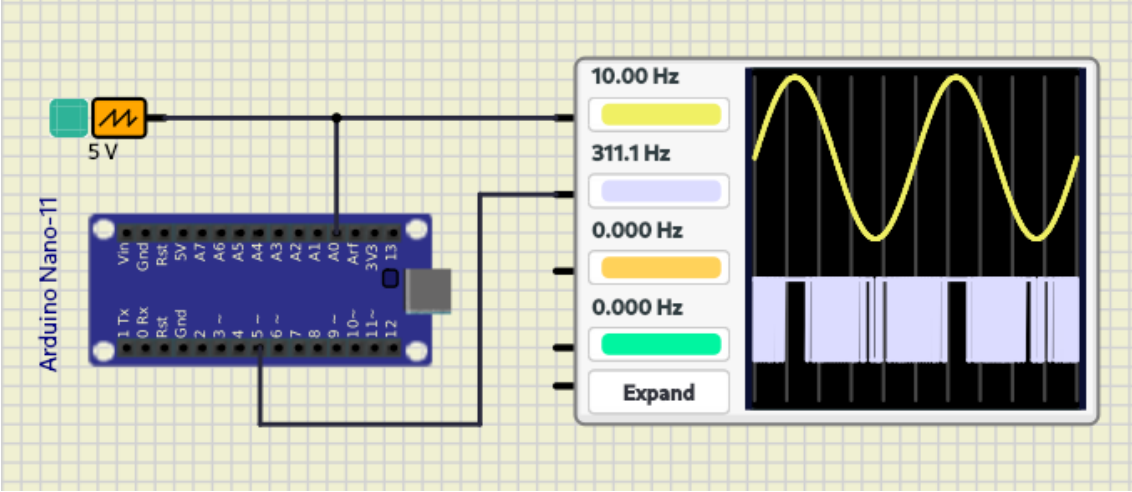


Figura 6 - Esquemático do circuito para modulação PWM de uma entrada senoidal

Após a montagem do circuito, variou-se a frequência da onda senoidal, fazendo a assumir = 10 Hz, 100 Hz, 500 Hz e 1 kHz. O resultado pode ser visualizado na figura 7 a seguir:

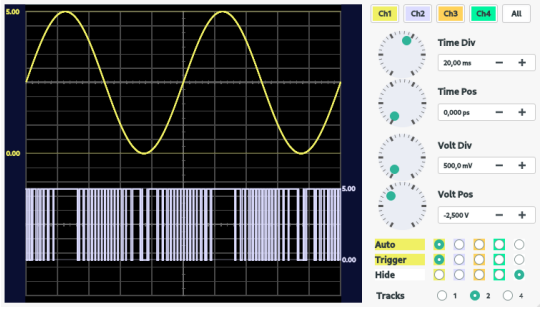
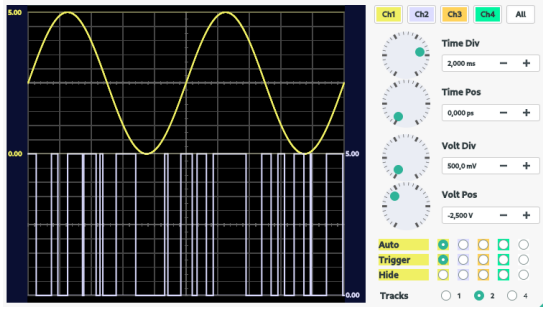
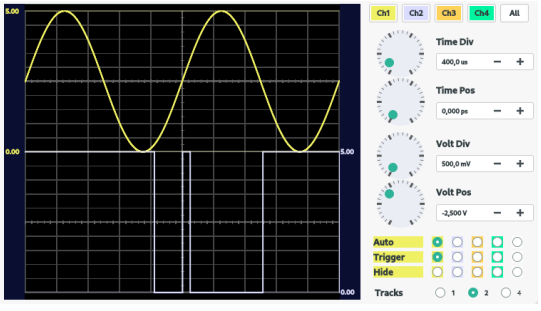
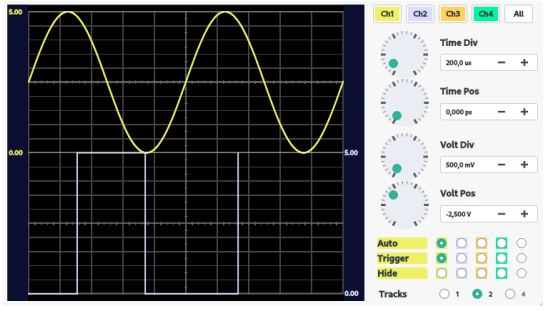
$f = 10Hz$	$f = 100Hz$
	
$f = 500Hz$	$f = 1000Hz$
	

Figura 7 - Avaliação da modulação da senoide para diversas frequências

Como se pode observar, a medida que se aumenta a frequência da onda senoidal, o PWM, por conta da quantidade de amostras/tempo, não consegue modular o sinal senoidal de frequências elevadas. Como uma possível solução para o problema, avalia-se o aumento da frequência do PWM. Dessa forma, posterior ao passo descrito, buscou-se analisar os seguintes registradores, almejando-se aumentar a frequência do PWM.

Análise dos registradores TCCR0B e ADCSRA

	Reg.	Type	Dec	Value
122	ADCSRA	u8	199	1100 0111
69	TCCR0B	u8	3	0000 0011

Figura 8 - Valores dos registradores TCCR0B e ADCSRA do ATMega328P

Configuração dos bits do registrador TCCR0B

Para se reconfigurar os bits do registrador TCC0B para obter a maior frequência possível de PWM primeiramente deve-se consultar o *data sheet* fornecido pelo fabricante.

Descrição do registrador TCC0B fornecido pelo fabricante

TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 9 - Descrição dos bits internos do registrador TCC0B

De acordo com a figura 8, sabemos que o valor do registrador TCC0B é de **0b00000011**. Analisando especialmente os bits **CS02**, **CS01**, **CS00**, nota-se pela figura 9 e 10 que eles são responsáveis pelo prescaler do clock para o **Timer/Counter1**, este Timer como sabemos, é utilizado para o circuito do PWM.

Esquemático do Prescaler

Figure 16-2. Prescaler for Timer/Counter0 and Timer/Counter1(1)

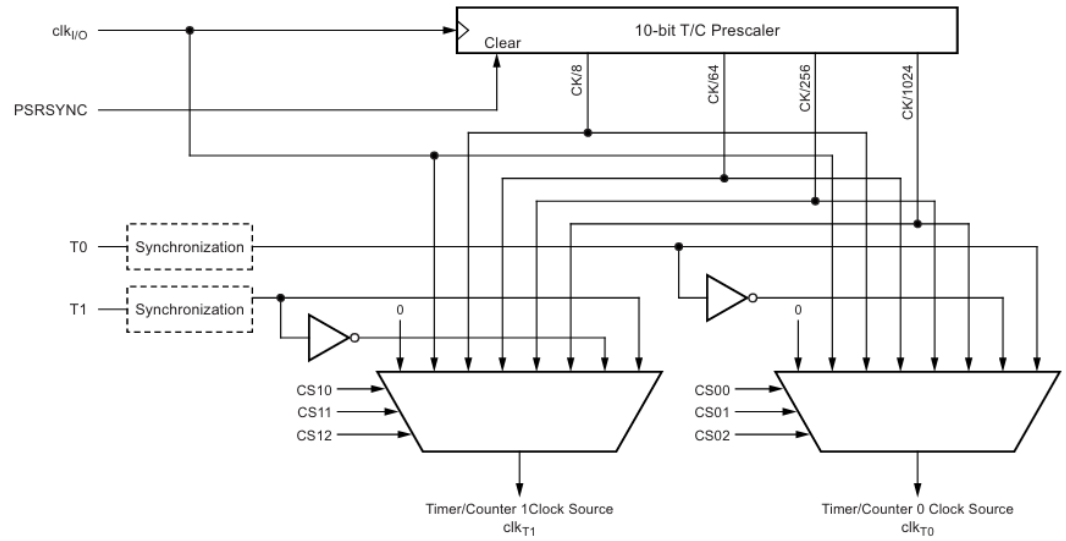


Figura 10 - Funcionamento interno do divisor de frequência

Analisando também a figura 11, e com o valor do registrador verificado na figura 8, é possível calcular a frequência atual do PWM. Sabendo que a frequência fornecido pelo circuito de cristal externo é de $16MHz$, com o valor de **0b011** nos bits **CS02**, **CS01**, **CS00**, temos um prescaler de 64. Considerando também que o PWM é de 8 bits, para se determinar a frequência do PWM atual

podemos implementar o que tá descrito em **(3)**:

$$clk_{PWM} = \frac{clk_{IO}}{64} \cdot \frac{1}{2^8} = \frac{16000000}{64 \cdot 256} = 976.5625Hz \quad (3)$$

Esta frequência também foi obtida via simulação, representada na figura 5 e definida na equação **(2)**.

Divisor de frequência

Table 14-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(no prescaling)
0	1	0	$clk_{I/O}$/8 (from prescaler)
0	1	1	$clk_{I/O}$/64 (from prescaler)
1	0	0	$clk_{I/O}$/256 (from prescaler)
1	0	1	$clk_{I/O}$/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.

Figura 11 - Tabela 14.9 do datasheet, configuração dos registradores para prescaler

Para se obter a maior frequência possível de PWM, deve-se ajustar esses mesmos bits para **0b001**, correspondente a opção no prescaling.

Como sabemos o valor atual do registrador, essa operação será realizada a seguir alterando somente os bits necessários, por isso essa configuração (instrução em código) poderia ser diferente se nos deparássemos com um outro microcontrolador em uma situação prática. O código final pode ser visto abaixo:

```
void setup() {
  pinMode(5,1);
  TCCR0B&=~(1<<CS01);
}

void loop() {
  int input = analogRead(A0);
  int output = map(output, 0, 1023, 0, 255);
  analogWrite(5, output);
}
```

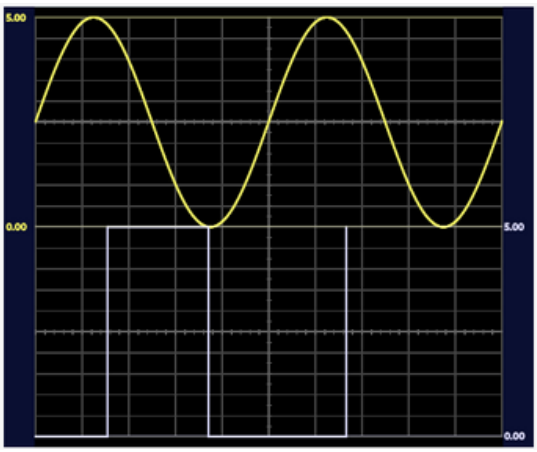
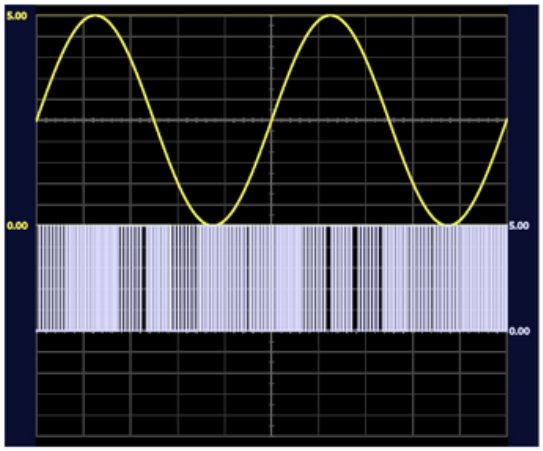
Resultados anteriores	Resultados após ajuste
	

Figura 12 - Comparação dos resultados anteriores e após ajuste no registrador TCC0B

É possível perceber pela figura 12 que o sinal após o ajuste consegue representar melhor a senoide. Esta senoidel utilizada possui frequência $f = 1kHz$.

Análise dos registradores TCCR0B e ADCSRA

	Reg.	Type	Dec	Value
122	ADCSRA	u8	196	1100 0100
69	TCCR0B	u8	1	0000 0001

Figura 13 - Valores dos registradores TCCR0B e ADCSRA do ATMega328P

Analisando os registradores, podemos ver a mudança realizada.

Configuração dos bits do registrador ADCSRA para fs = 76.9KHz

Para definir uma taxa de amostragem, primeiramente, deve-se conhecer o número de ciclos de clock necessários para uma conversão no ADC.

Diagrama temporal do ADC

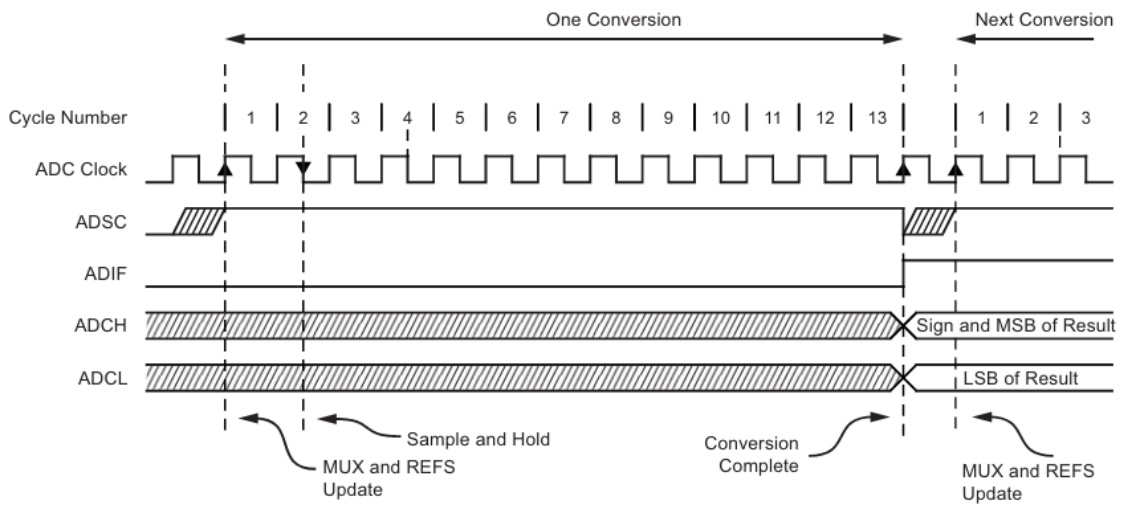


Figura 14 - Diagrama temporal do ADC para a obtenção de uma única amostra

Analisando a figura 14, obtida pelos dados do fabricante, pode-se verificar que são necessários exatos 13 ciclos de clock para uma única amostra. Com isso podemos determinar a frequência do sinal de clock para o circuito do ADC, sabendo a taxa de amostra do ADC desejado de $f_s = 76.9Hz$.

$$clk_{ADC} = n \cdot f_s = 13 \cdot 76.9kHz = 0.9997MHz \quad (4)$$

Com o clock do ADC requerido, consultando os dados do fabricante de uma forma análoga à etapa anterior e consultando a figura 15, conclue-se que para obter este clock, deve-se realizar um "prescaler" de 16.

Bits para prescaler do clock do ADC

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Figura 15 - Bits ADSP2, ADSP1, ADSP0 para prescaler do clock do ADC

Estes bits se encontram nos espaços de memória dos bits menos significativos do registrador ADCSRA.

Registrador de controle ADCSRA
ADCSRA – ADC Control and Status Register A Bit 7 6 5 4 3 2 1 0 (0x7A) ADEN ADSC ADATE ADIF ADIE ADPS2 ADPS1 ADPS0 ADCSRA Read/Write R/W R/W R/W R/W R/W R/W R/W R/W Initial Value 0 0 0 0 0 0 0 0

Figura 16 - Bits ADSP2, ADSP1, ADSP0 como LSB do ADCSRA

Assim, para obter um fator de divisão de 16, deve-se configurar os bits com os valores **0b100**, como é de conhecimento o valor anterior deste registrador, o código para mudar os bits para os desejados (para esta ocasião) está descrito abaixo:

```
void setup() {
  TCCR0B&=~(1<<CS01);
  ADCSRA&=~(1<<ADPS1);
  ADCSRA&=~(1<<ADPS0);
  pinMode(5,1);
}

void loop() {
  const int output = map(analogRead(A0), 0, 1023, 0, 255);
  analogWrite(5, output);
}
```

Para se avalizar o resultado, tomamos um intervalo de $\frac{\lambda}{4}$ como mostra o gráfico abaixo:

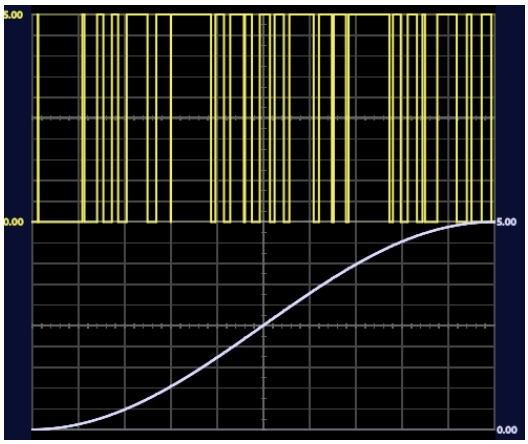
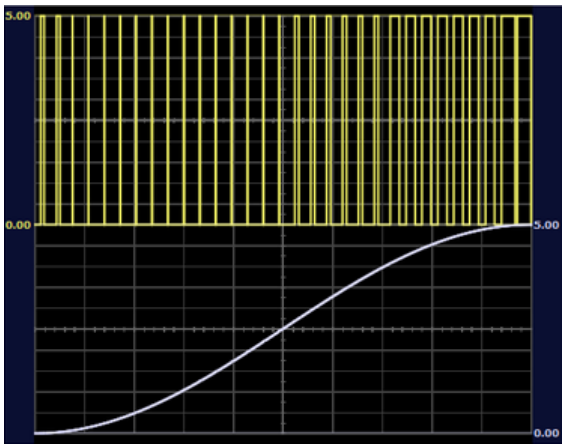
Resultado anterior	Resultado após ajuste em ADCSRA
	

Figura 17 - Comparação dos resultados após ajuste no registrador ADCSRA

Neste intervalo, foi possível verificar que após o ajuste, obtivemos uma modulação mais suave, não há mudanças tão repentinas nos ciclos de trabalho, isso decorre de uma maior amostragem do sinal.

Conclusões

A partir do presente trabalho foi possível compreender diversas funcionalidades de um microcontrolador, entre elas, a sua capacidade de converter sinais analógicos em digitais com uma precisão de 10 bits, no caso do microcontrolador utilizado.

Além disso, pode-se compreender conceitos importantes como modulação por largura de pulso (PWM) de um sinal senoidal e a possibilidade de alterar registradores de um microcontrolador com o objetivo de aumentar a frequência do PWM e a taxa de amostragem.

Projeto de um filtro e conversão DA a partir do PWM

Introdução

Sabe-se que os diferentes sinais podem ser escritos como cossenos e senos ou ainda, por suas somas, que se diferenciam com relação a fase, amplitude e frequência. Em aplicações de Engenharia Elétrica são incontáveis as situações em que se almeja realizar uma alteração na amplitudes relativas de algumas frequências, ou mesmo eliminar determinadas faixas de frequência, a esse processo denominamos o nome de filtragem. Os filtros são os responsáveis por realizar a operação de atenuação ou eliminar uma determinada faixa, enquanto permite a passagem de componentes com outras faixas de frequência. Entre as aplicações, pode-se citar: recortar uma faixa de frequência para que se observe o comportamento de um determinado sistema, ou ainda, melhorar
a resposta de um determinado sistema após a retirada de componentes específicos de frequência.

Por meio do teorema da amostragem um sinal de tempo contínuo pode ser completamente representado por suas amostras uniformemente espaçadas no tempo, em um período T , isto é, a informação contida no sinal em tempo contínuo amostrado é equivalente ao do sinal discreto e de forma a minimizar ou eliminar o erro atrelado a essa operação, deve-se manter uma alta taxa de amostragem para a aplicação em questão. Entre as aplicações, pode-se citar a gravação de um áudio de voz por um celular, é óbvio que o sinal de áudio não está em tempo contínuo, contudo a taxa de amostragem está suficientemente alta ao ponto dos erros não serem percebidos pelo ouvido humano.

Dessa forma, após compreender sobre filtros e conversão analógica digital (AD), a presente etapa do trabalho busca realizar o projeto de dois filtros de ordens diferentes para promover conversão D/A a partir de um sinal PWM.

Projeto dos filtros

O procedimento inicial para o trabalho foi o estudo, estruturação e desenvolvimento dos dois filtros solicitados conforme os requisitos a seguir:

- Arquiteturas: filtros RC passivo e RC-Ativo
- Seletividade: passa-baixas e passa banda
- Frequência central (de corte): $f_0 = 1kHz$
- RC-Ativo: $N = 2$, $K_{\omega_0} = 1$, $Q = 10$.

Projeto do Filtro RC

O primeiro filtro implementado foi o de primeira ordem, também conhecido como filtro RC (Resistor - Capacitor) passivo, com frequência central de 1kHz. A seguir vê-se o procedimento realizado para o dimensionamento dos componentes:

$$
\begin{aligned}
&f = 1000Hz \\
&w_0 = 1000 \cdot 2\pi \\
&RC = \frac{1}{w_0} = 0.0001592
\end{aligned}
\tag{5}
$$

Assumindo $C = 100nF$ obtemos $R = 1591.549\Omega$

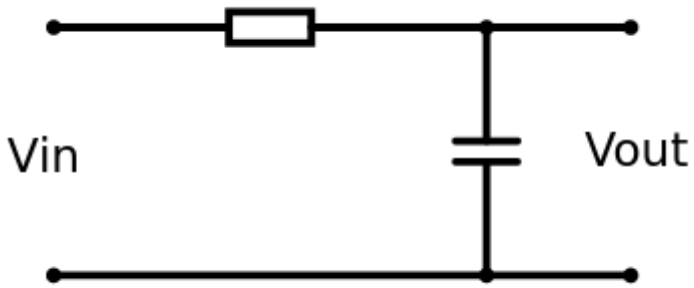


Figura 18 - Filtro RC simples passa baixa

Por fim verifica-se na figura 18 a topologia e os valores dos componentes assumidos para montar esse circuito.

Projeto do Filtro Ativo

Para o projeto do filtro passa faixa de segunda ordem, utilizamos a equação de transferência:

$$
T(s) = \frac{K_{max}\frac{w_0}{Q}s}{s^2 + \frac{w_0}{Q}s + w_0^2} \tag{6}
$$

De posse das espeficicações apresentadas, pode-se determinar os coeficientes numéricos da função de transferência do filtro.

$$
\begin{aligned}
T(s) &= \frac{200\pi\ s}{s^2 + 200\pi s + 4000000\pi^2} \\
T(s) &= \frac{628.3185\ s}{s^2 + 628.3185s + 39478417.60}
\end{aligned}
\tag{7}
$$

A topologia utilizada para implementação deste filtro será a **Sallen-&-Key**.

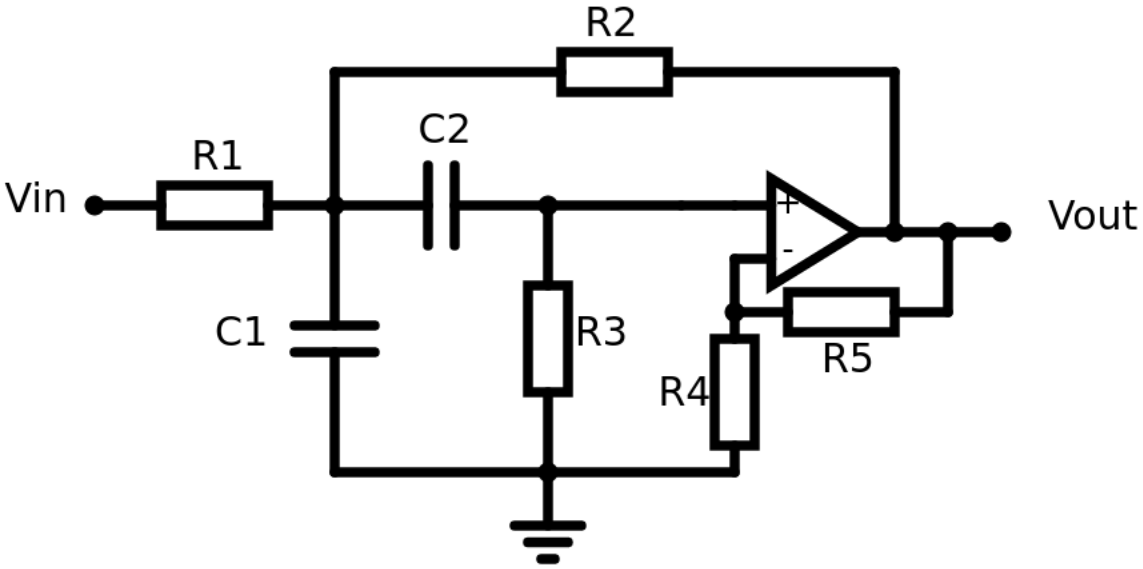


Figura 19 - Filtro Ativo passa faixa

A função de transferência deste filtro está representada abaixo:

$$T(s) = \frac{ks(R_1C_1)}{s^2 + s\left(\frac{1}{R_1C_1} + \frac{1}{R_3C_2} + \frac{1}{R_3C_1} + \frac{1-k}{R_2C_1}\right) + \frac{1}{(R_1//R_2)R_3C_1C_2}} \tag{8}$$

Analisando a equação de transferência, podemos implementar o filtro conforme a equação 7, assim temos:

$$\omega_0 = \sqrt{\frac{1}{(R_1//R_2)\,R_3C_1C_2}} \tag{9}$$

Fazendo $C_1 = C_2 = 1$ e $R_1 = R_2 = R_3 = R$:

$$w_0 = \sqrt{\frac{1}{\left(\frac{R}{2}\right)R \cdot 1 \cdot 1}} \tag{10}$$

$$R = \frac{\sqrt{2}}{w_0} = 0.000225079079039\ \Omega$$

Como na função transferência, o valor de R aparece sempre em par com C, podemos aumentar um, na mesma proporção de diminuição de outro valor mantendo o produto RC constante.

Utilizando o mesmo valor de capacitor do filtro passivo $C = 100nF$, representando uma redução de 10^7, aumentando R obtido no mesmo fator, temos $R = 2250.79\ \Omega$

E assim, para determinar o k:

$$\frac{w_0}{Q} = \frac{3}{R_iC_i} + \frac{1-k}{R_iC_i} = 200\pi \tag{11}$$
$$k = 3.85857864376$$

E com o valor de k, podemos atribuir um valor para R_4 e determinar R_5:

$$k_2 = 1 + \frac{R_5}{R_4} \tag{12}$$
$$R_4 = 1000\Omega$$
$$R_5 = 2858.57\Omega$$

De posse desses valores, podemos determinar o ganho para frequencia zero.

$$K_{max} = \frac{k}{4-k} = 27.28 \tag{13}$$

Por isso utilizamos resistores em divisor de tensão na entrada para diminuir o ganho por este mesmo fator:

$$V_1 = V_i \cdot \alpha = V_i \frac{R_{11}}{R_{11} + R_{12}} \tag{14}$$

$$\alpha = \frac{1}{K_{max}}$$

$$R_{12} = \frac{R_2}{1-\alpha} = 2336.42\Omega$$

$$R_{11} = R_{12} \cdot \frac{1-\alpha}{\alpha} = 61411.18\Omega$$

Descrição das simulações solicitadas e análise dos resultados

O resultado do filtro passa baixa implementado analiticamente pode ser observado a seguir na figura 19:

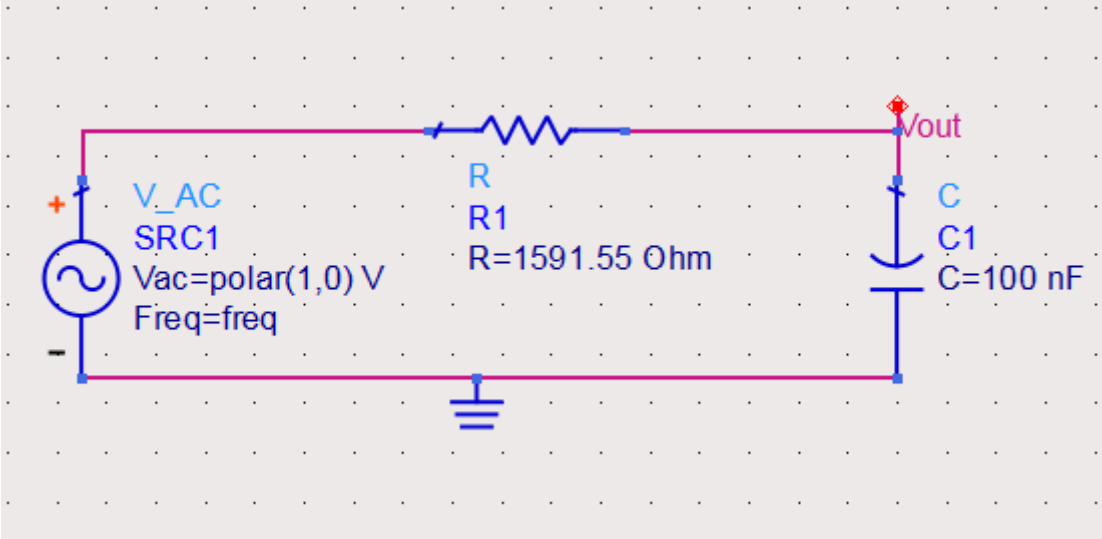


Figura 19 - Filtro passa baixa

Na figura 20, a seguir, é possível também, verificar a topologia do filtro passa faixa, sem, contudo, realizar a alteração promovida para definir um K máx igual a 1:

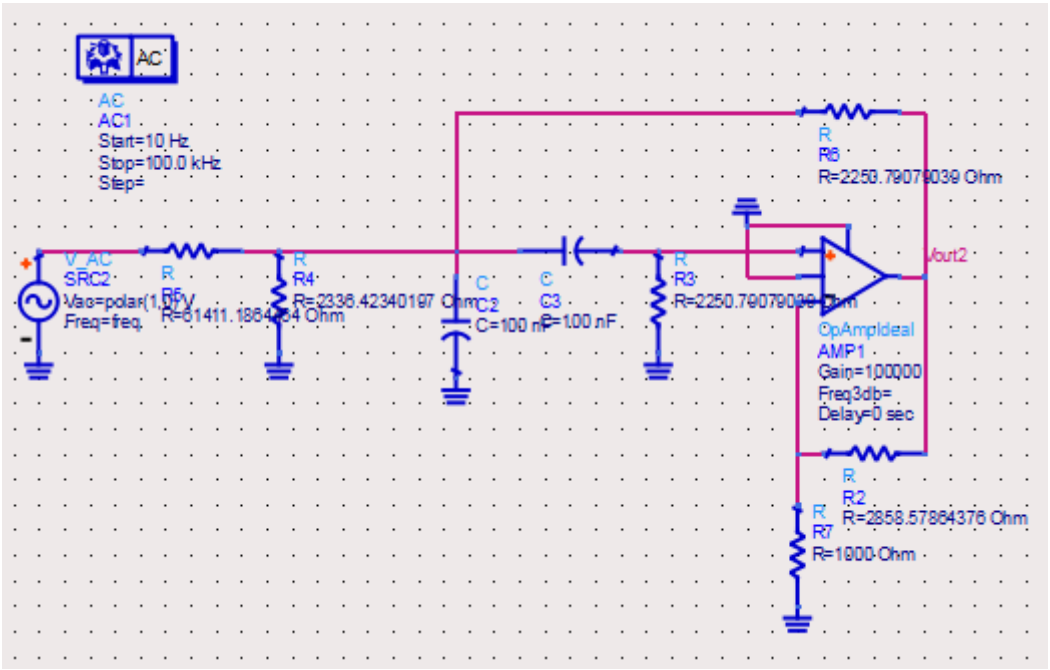


Figura 20 - Filtro Ativo passa faixa

Nas figura 21 e 22 abaixo pode-se comparar o comportamento dos dois filtros, o de primeira ordem com o seu comportamento característico e o passa faixa centralizado em 1KHz.

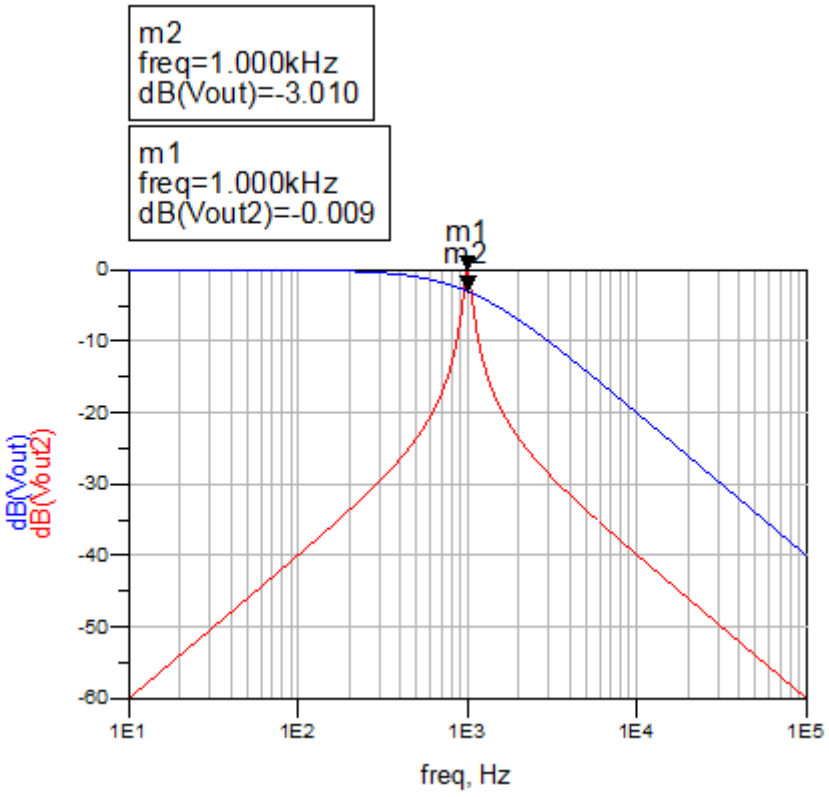


Figura 21 - Comparação da magnitude dos filtros

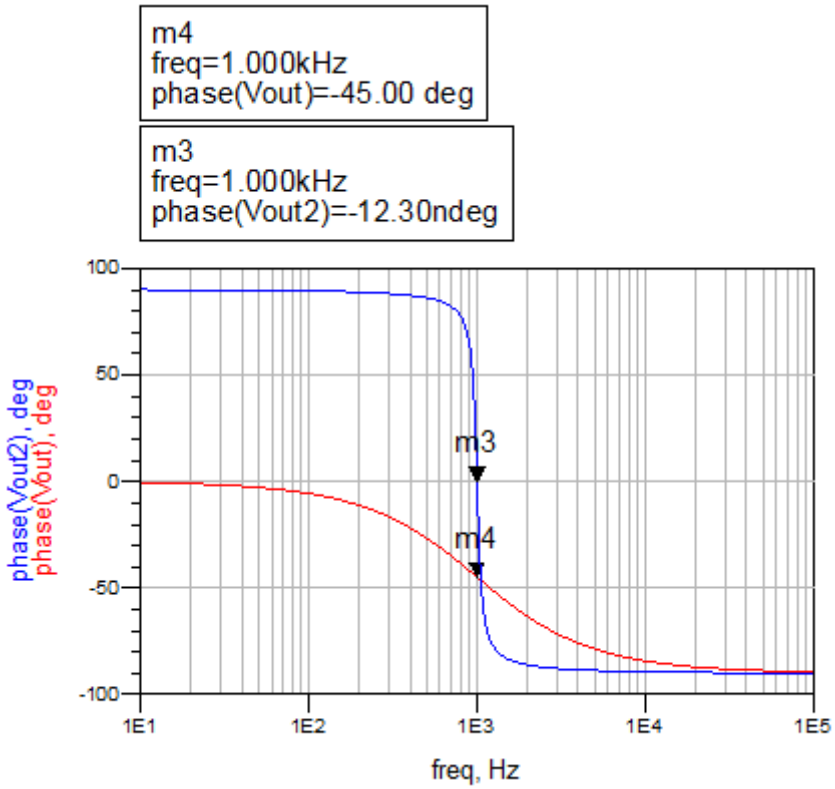


Figura 22 - Comparação da fase dos filtros

Filtro RC

O resultado da implementação do filtro RC dimensionado no ADS em conexão com o osciloscópio pode ser visualizado a seguir:

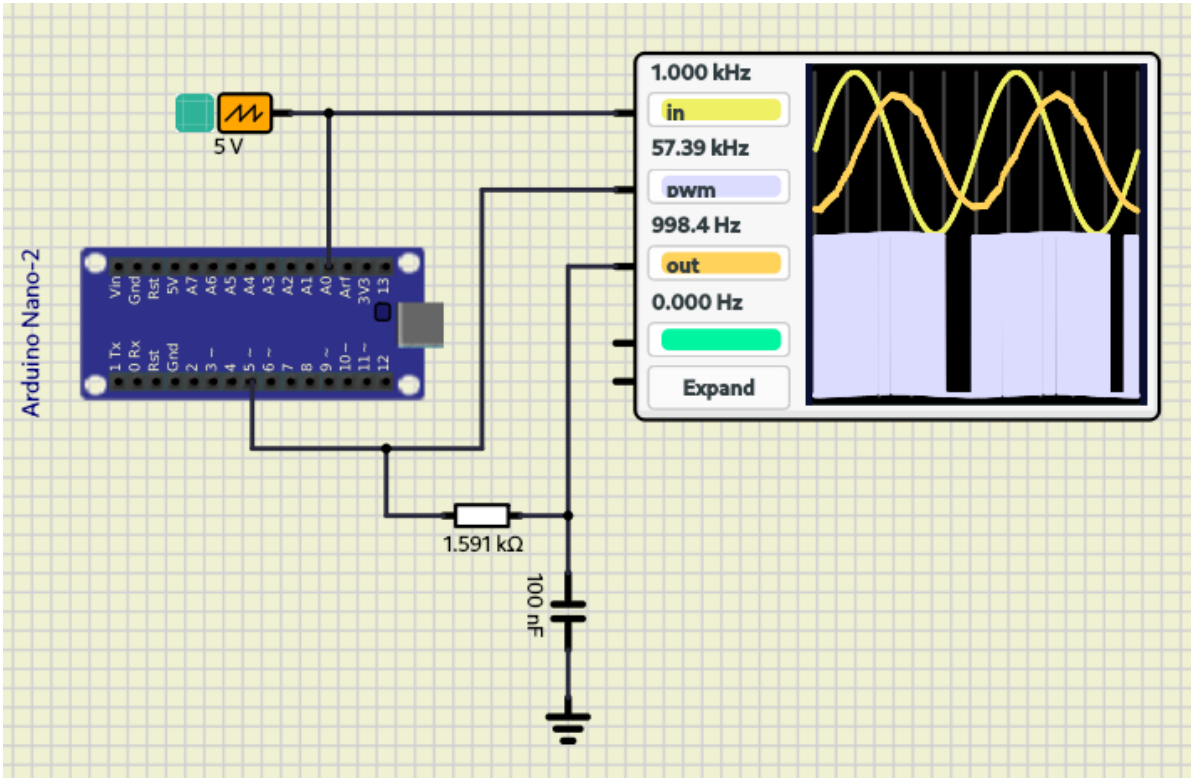


Figura 23 - Simulação do filtro passivo

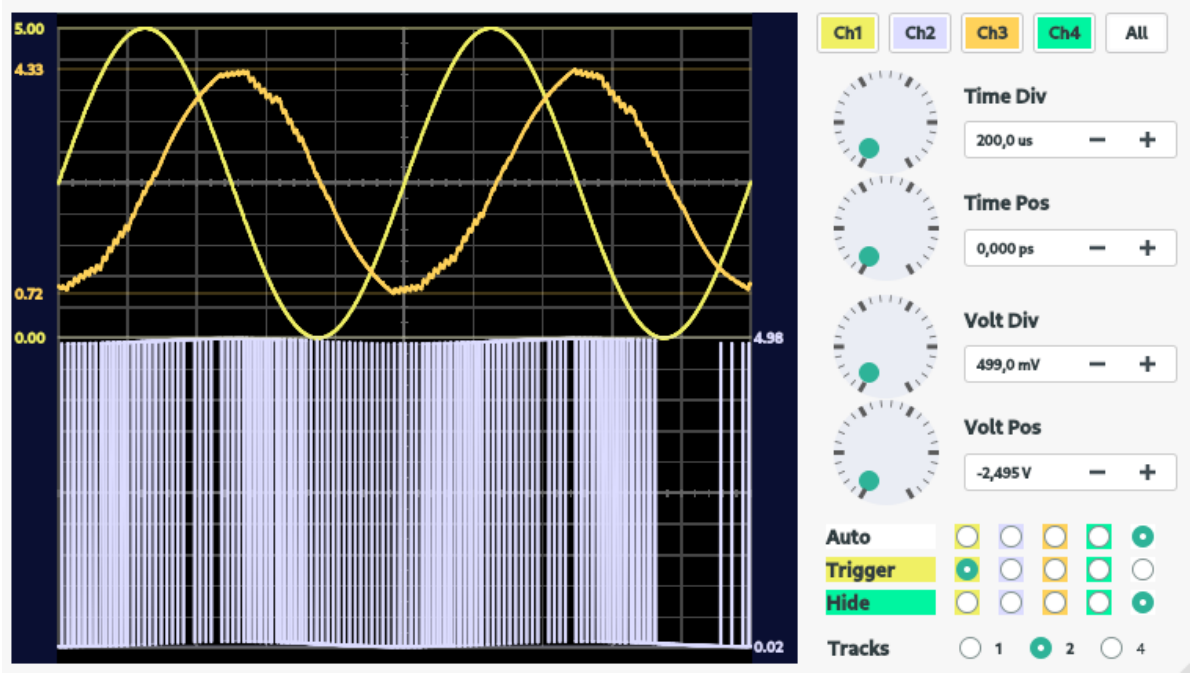


Figura 24 - Osciloscópio do filtro passivo

O que se observa é a reconstrução do sinal senoidal por meio da atuação do filtro no sinal PWM PWM. Verifica-se ainda, como um comportamento inerente do filtro RC, um defasamento de 45° entre saída e entrada.

Filtro ativo passa faixa

O resultado da implementação do filtro RC dimensionado no ADS em conexão com o osciloscópio pode ser visualizado a seguir:

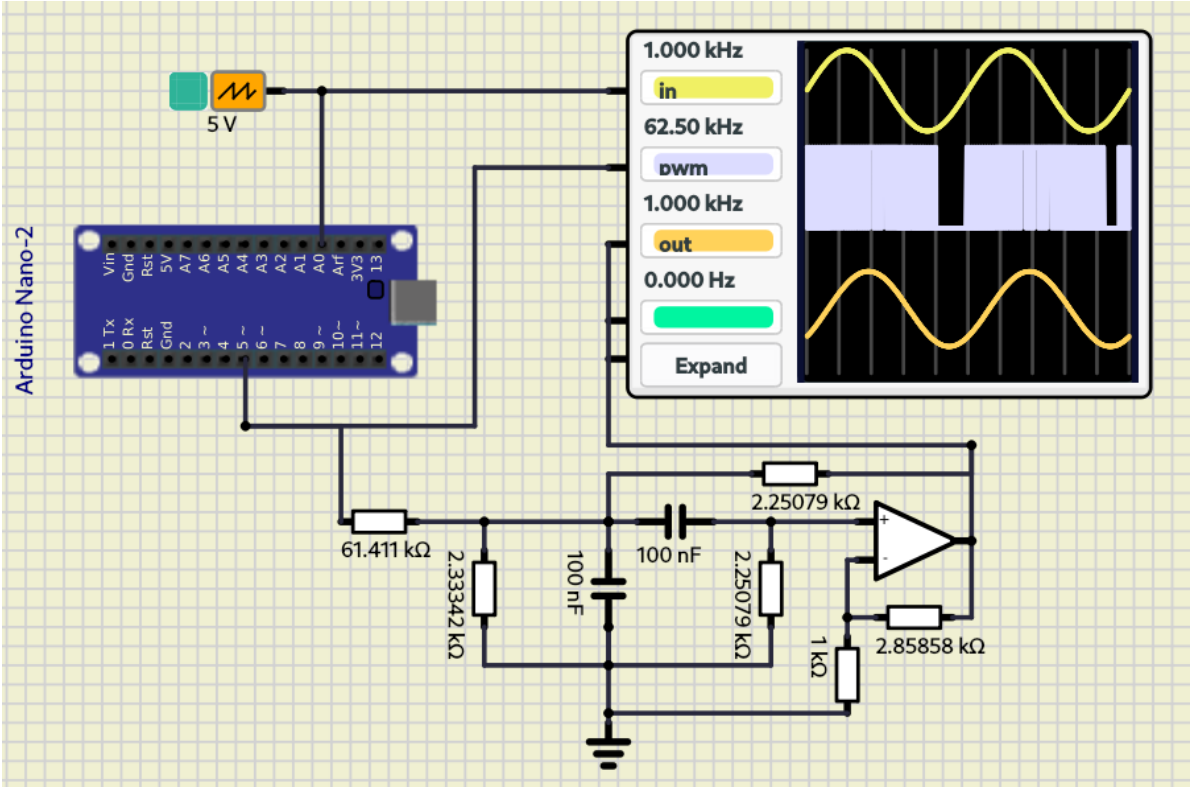


Figura 25 - Simulação do filtro ativo

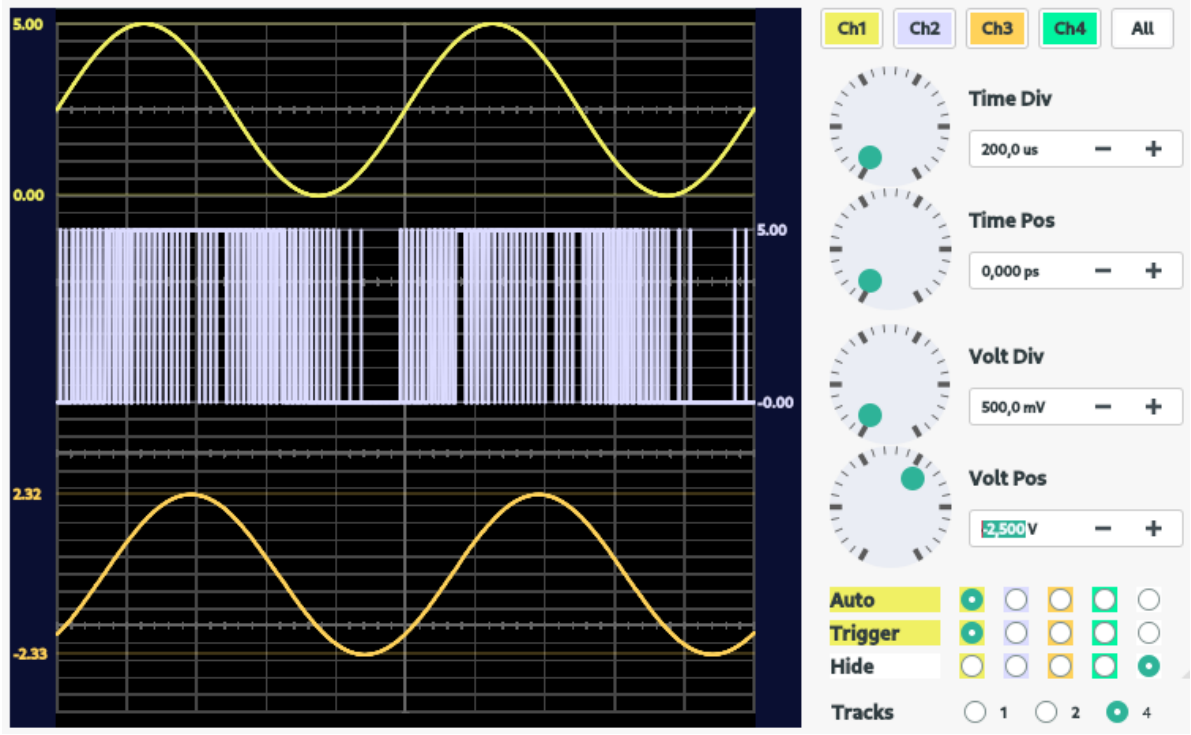


Figura 26 - Osciloscópio do filtro ativo

O resultado da atuação do circuito ativo é a satisfatória reconstrução do sinal senoidal com frequência 1kHz, sem, contudo, provocar uma defasagem entre os sinais.

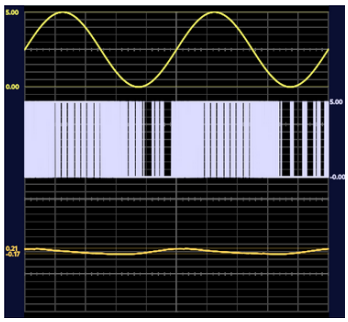
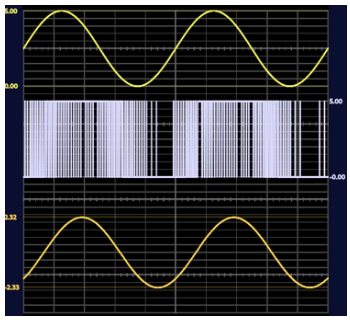
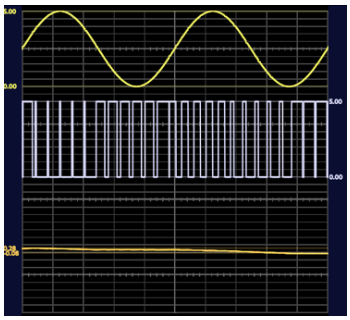
$f = 500Hz$	$f = 1000Hz$	$f = 5000Hz$
		

Figura 27 - Atuação do filtro ativo passa faixa para diversas frequências

Na figura 27, pode-se analisar a atuação do filtro passa faixa, atenuando o sinal para componentes do sinal que se encontram fora da faixa de frequência para o qual o filtro foi projetado.

Conclusões

Através da pesquisa pode-se compreender a atuação de dois tipos de filtros diferentes: o passa baixa RC e o filtro ativo passa faixa de segunda ordem. Como descrito anteriormente, o filtro passa baixa causou uma defasagem de 45° no sinal de saída com relação ao sinal de entrada PWM, por outro lado, o filtro RC ativo não causa defasagem entre a saída e a entrada.

Por fim, pode-se concluir, não somente a importância dos filtros, mas também a capacidade de se utilizá-los em conjunto com microcontroladores para filtrar sinais e auxiliar na obtenção de dados desejados.

Referências

Noceti-Filho, Sidnei, “Filtros Seletores de Sinais,” Editora da UFSC, Florianópolis, 2003.

Arduino Language Reference https://www.arduino.cc/reference/en/.

LATHI, B. P. Sinais e Sistemas Lineares. 2ª Ed., Porto Alegre, Editora Bookman, 2006. 856p.