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.

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

```
1  void setup() {
2    Serial.begin(9600);
3  }
4
5  void loop() {
6    int sensorValue = analogRead(A0);
7    Serial.println(sensorValue);
8  }
```

Como mecanismo de alterar a entrada analógica optou-se pela utilização de um potenciômetro que representa uma variável analógica que pode assumir diversos valores.

Esquemático do Circuito

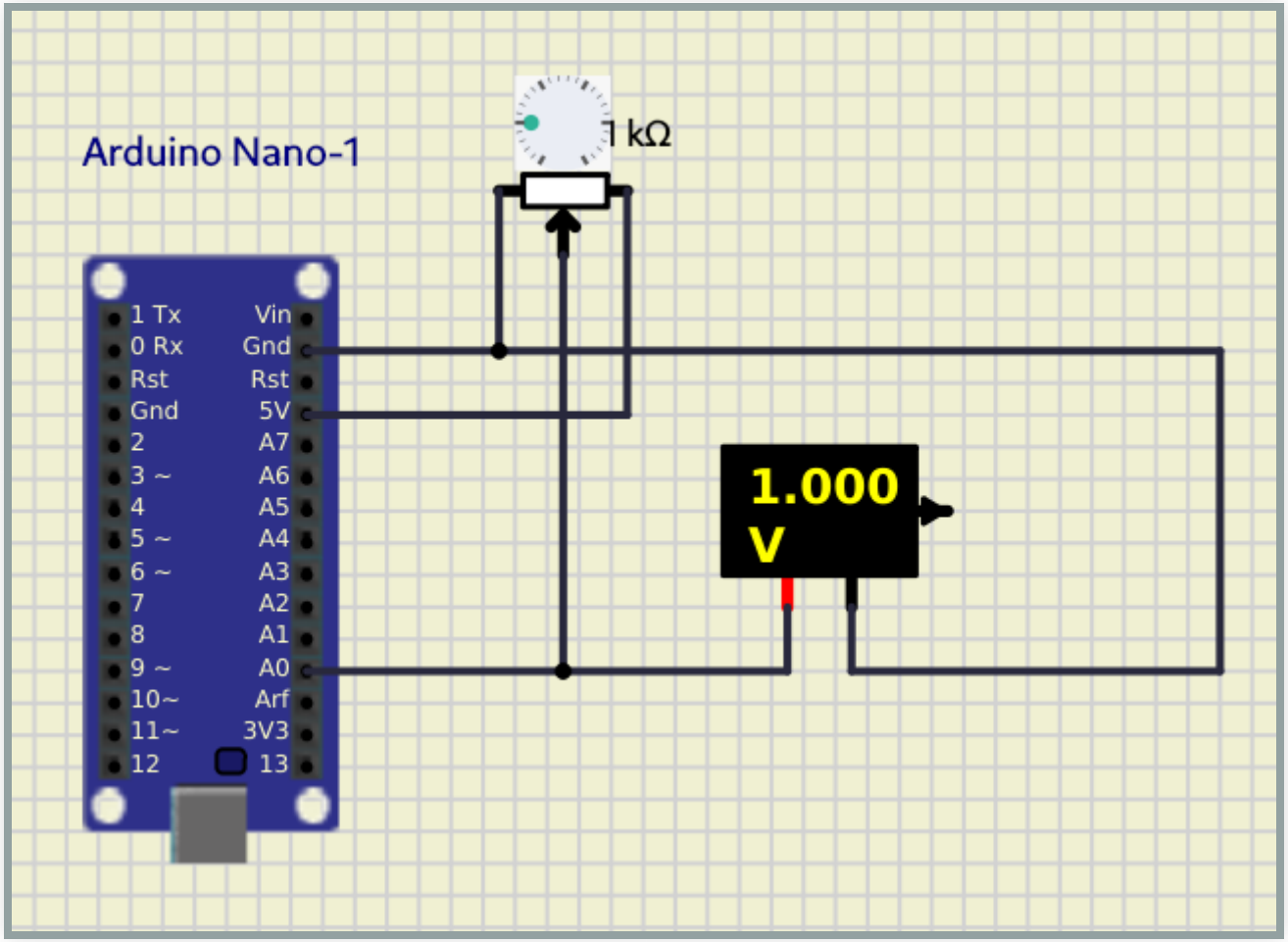


Figura 1 - Esquemático para avaliação da conversão AD

Saída do monitor serial

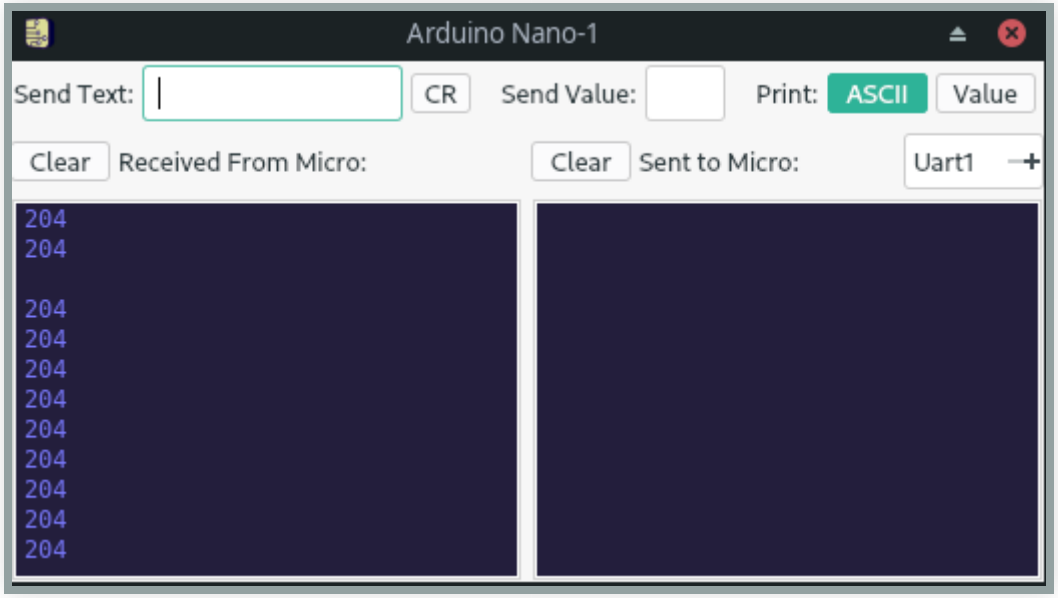


Figura 2 - Saída serial para entrada de 1.0V

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

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

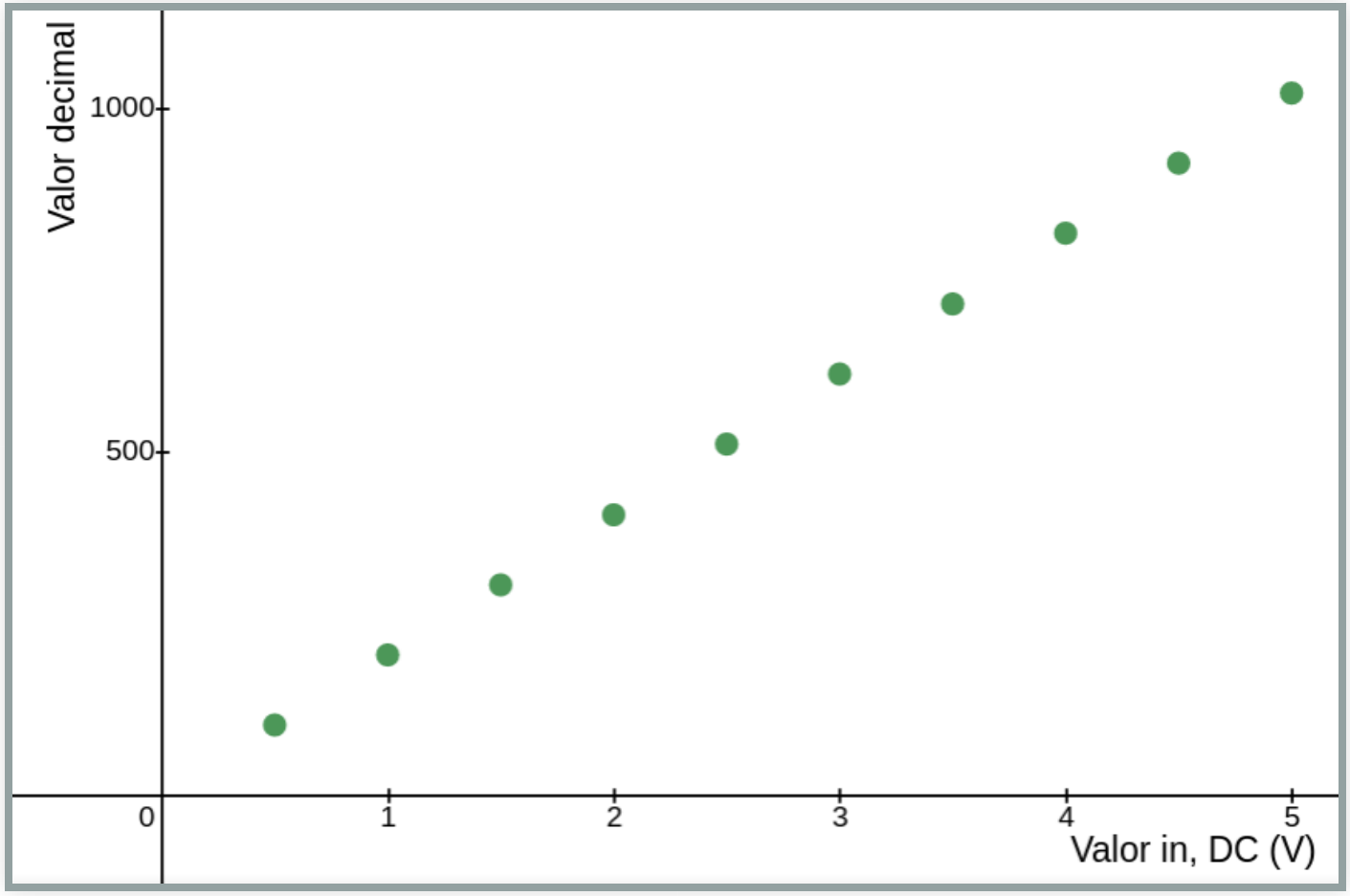


Figura 3 - Gráfico comparativo tensão de entrada e valor serial decimal

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.

```
1  void setup() {
2    pinMode(5,OUTPUT);
3  }
4
5  void loop() {
6    analogWrite(5, 0); // Segundo parâmetro
7                       // variado (D)
8  }
```

Esquemático do circuito para avaliação do PWM

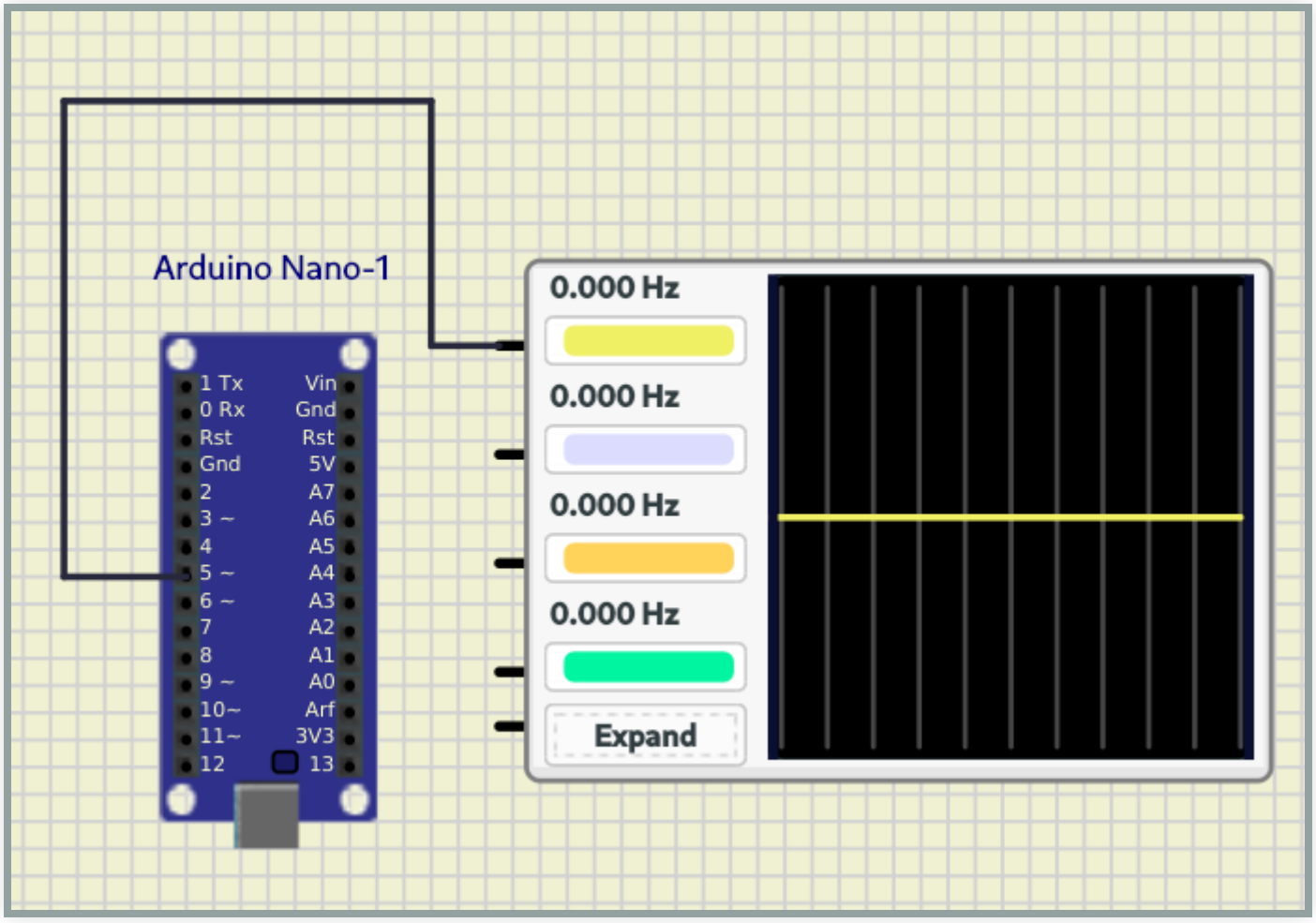
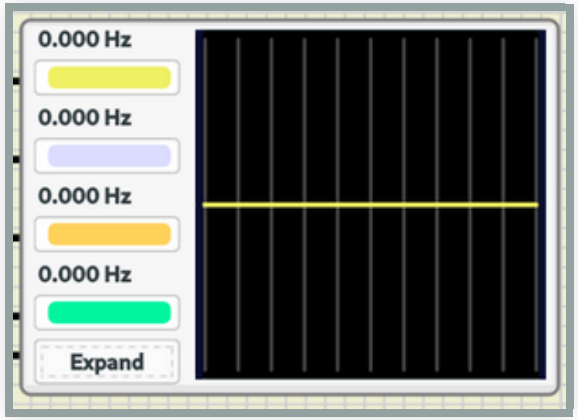


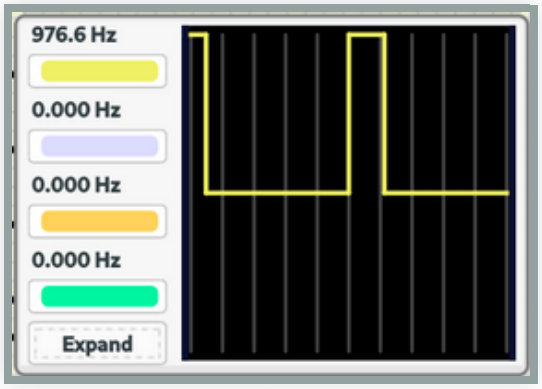
Figura 4 - Esquemático do circuito para avaliação do PWM

Resultados da simulação

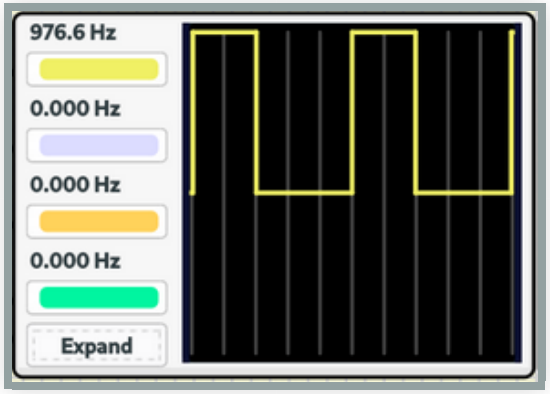
D=0



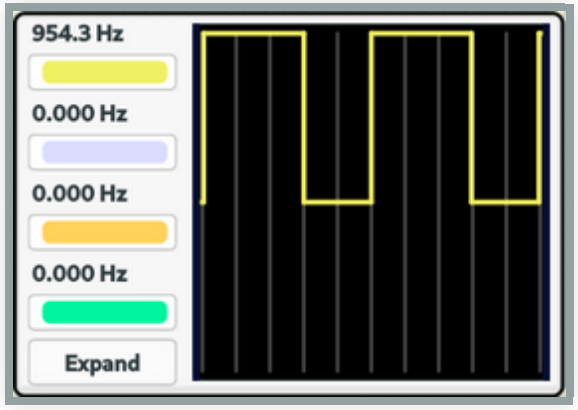
D=51



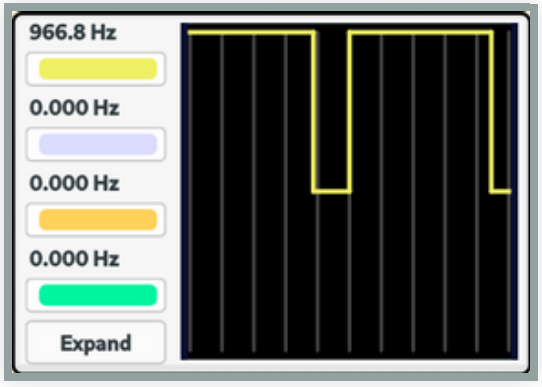
D=102



D=153



D=204



D=255

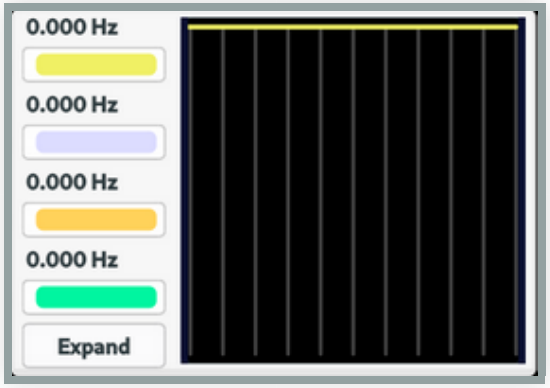


Figura 5 - Avaliação dos resultados para cada ciclo de trabalho

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:

$$\begin{cases} f = 976.7Hz \\ T = 1.02ms \end{cases} \tag{2}$$

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

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:

```
1  void setup() {
2    pinMode(5,1);
3  }
4
5  void loop() {
6    int input = analogRead(A0);
7    int output = map(output, 0, 1023, 0, 255);
8    analogWrite(5, output);
9  }
```

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

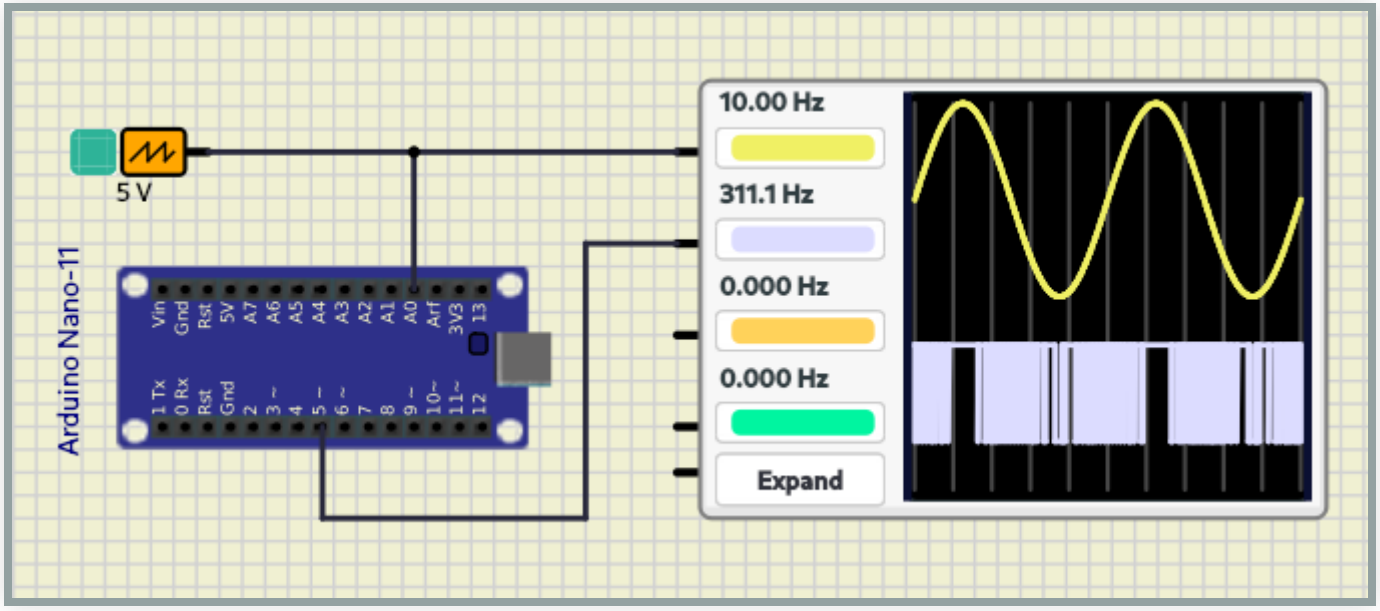


Figura 6 - Esquemático do circuito para modulação PWM de uma entrada senoidal

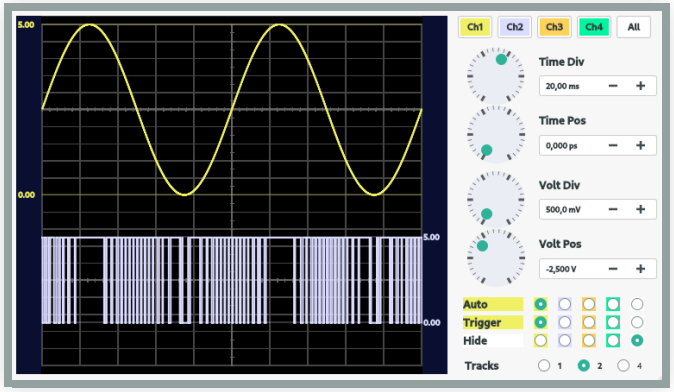
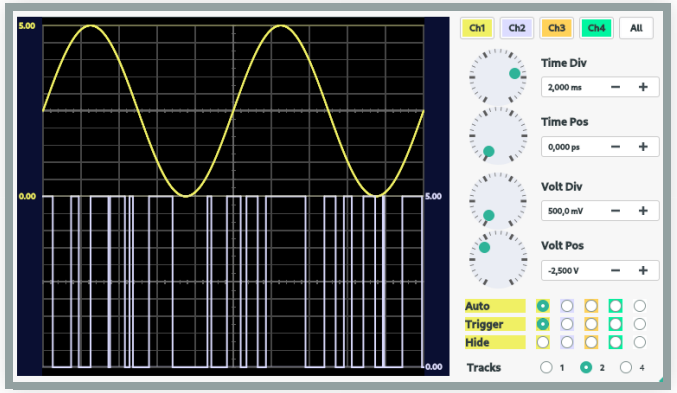
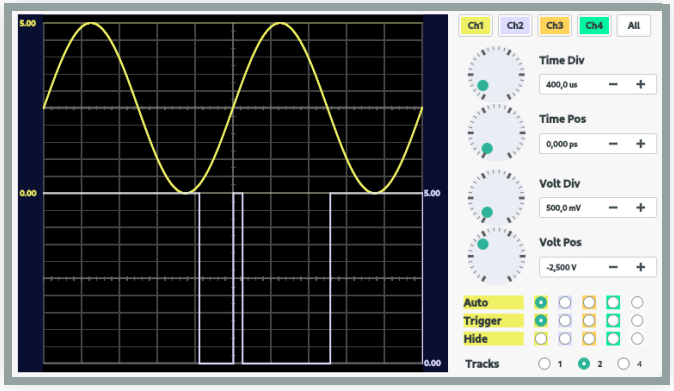
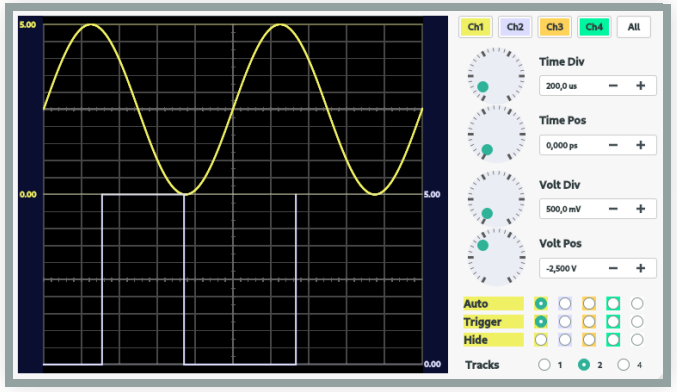
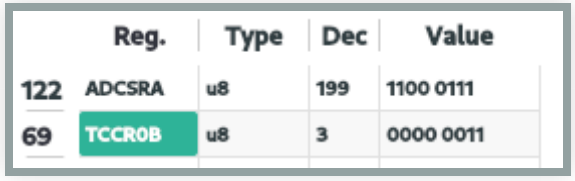
$f = 10Hz$	$f = 100Hz$
	
$f = 500Hz$	$f = 1000Hz$
	

Figura 7 - Avaliação da modulação da senoide para diversas frequências

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

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

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:

```
1  void setup() {
2    pinMode(5,1);
3    TCCR0B&=~(1<<CS01);
4  }
5
6  void loop() {
7    int input = analogRead(A0);
8    int output = map(output, 0, 1023, 0, 255);
9    analogWrite(5, output);
10 }
```

Resultados anteriores Resultados após ajuste

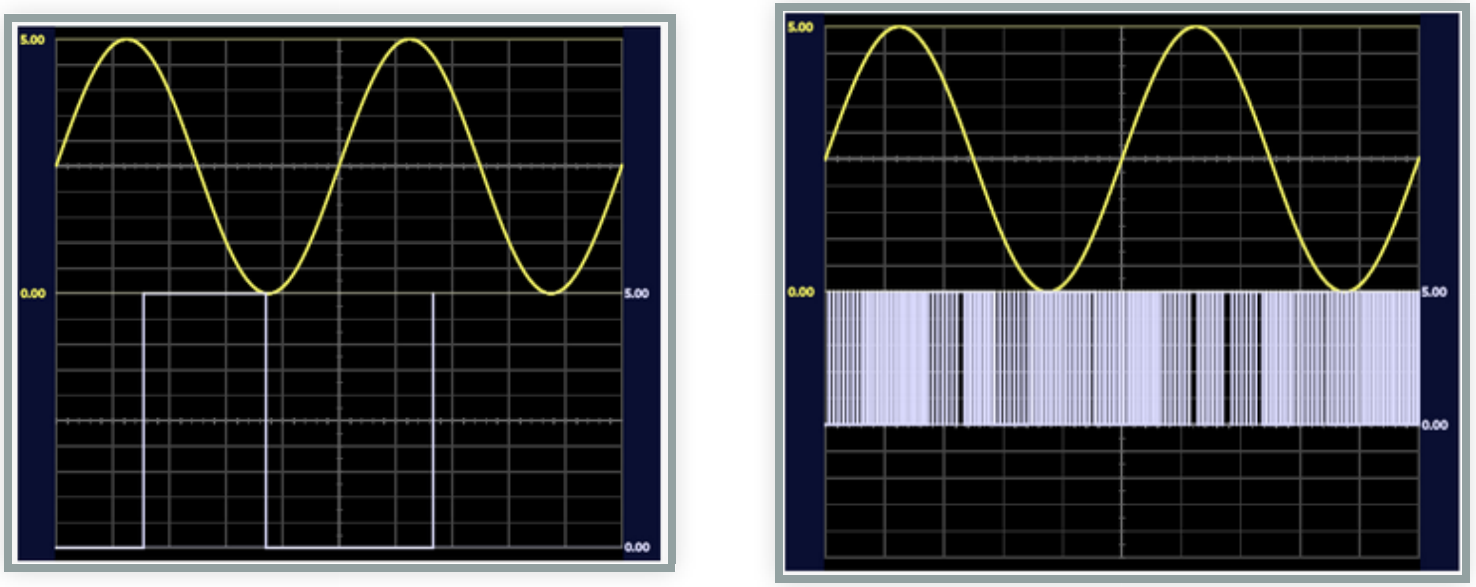
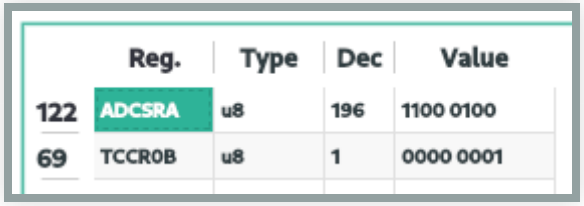


Figura 12 - Comparação dos resultados anteriores e após ajuste no registrador TCC0B

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

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.

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:

```
1  void setup() {
2    TCCR0B&=~(1<<CS01);
3    ADCSRA&=~(1<<ADPS1);
4    ADCSRA&=~(1<<ADPS0);
5    pinMode(5,1);
6  }
7
8  void loop() {
9    const int output = map(analogRead(A0), 0, 1023, 0, 255);
10   analogWrite(5, output);
11 }
```

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

Resultado anterior **Resultado após ajuste em ADCSRA**

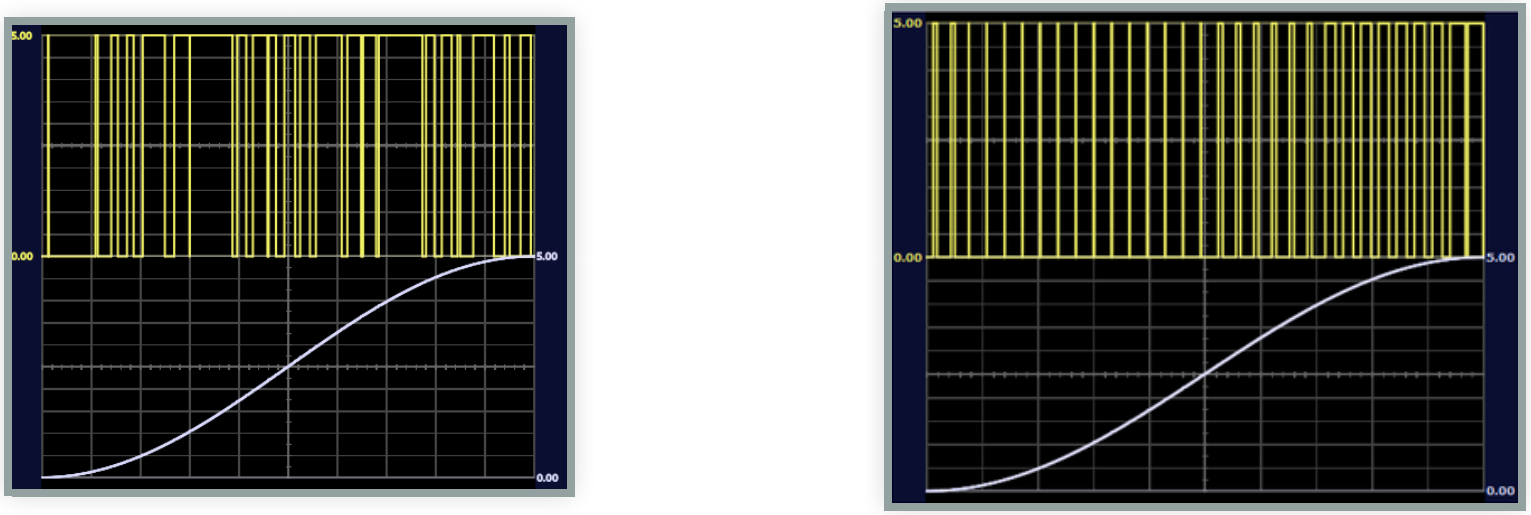


Figura 17 - Comparação dos resultados após ajuste no registrador ADCSRA

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

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.

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:

$$f = 1000Hz \tag{5}$$

$$w_0 = 1000 \cdot 2\pi$$

$$RC = \frac{1}{w_0} = 0.0001592$$

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

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 especificações apresentadas, pode-se determinar os coeficientes numéricos da função de transferência do filtro.

$$T(s) = \frac{200\pi\ s}{s^2 + 200\pi s + 4000000\pi^2} \tag{7}$$

$$T(s) = \frac{628.3185\ s}{s^2 + 628.3185s + 39478417.60}$$

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

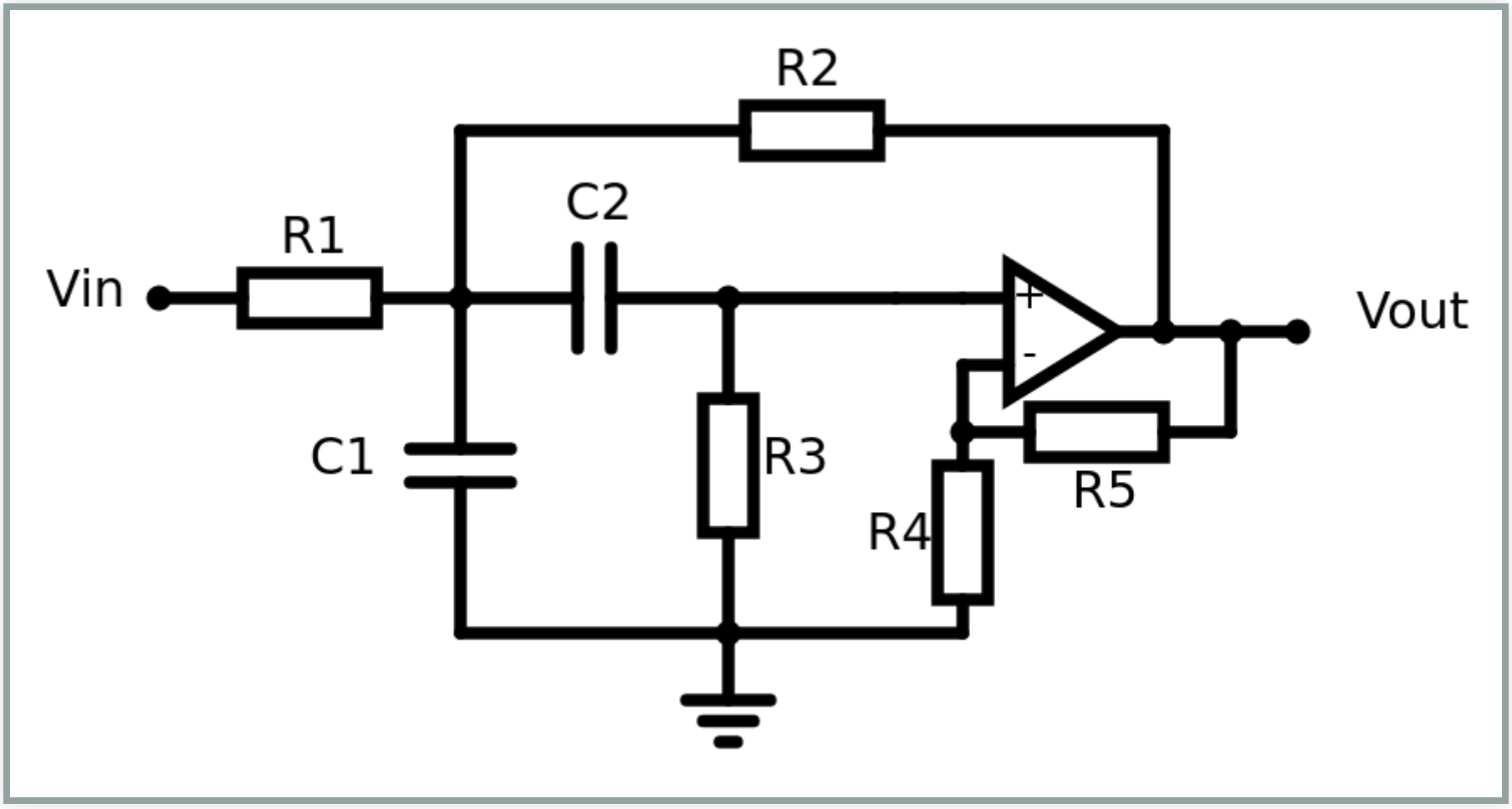


Figura 19 - Filtro Ativo passa faixa

A função de transferência deste filtro está representada abaixo:

$$T(s) = \frac{ks\left(R_1 C_1\right)}{s^2 + s\left(\frac{1}{R_1 C_1} + \frac{1}{R_3 C_2} + \frac{1}{R_3 C_1} + \frac{1-k}{R_2 C_1}\right) + \frac{1}{(R_1 // R_2) R_3}$$

A solução encontrada foi:

- $C = 100nF$
- $R_1 = R_2 = R_3 = 2250.79\ \Omega$
- $k = 3.85857864376$
- $R_4 = 1000\Omega$
- $R_5 = 2858.57\Omega$
- $K_{max} = \frac{k}{4-k} = 27.28$
- $R_{12} = \frac{R_2}{1-\alpha} = 2336.42\Omega$
- $R_{11} = R_{12} \cdot \frac{1-\alpha}{\alpha} = 61411.18\Omega$

Descrição das simulações solicitadas e análise dos resultados

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

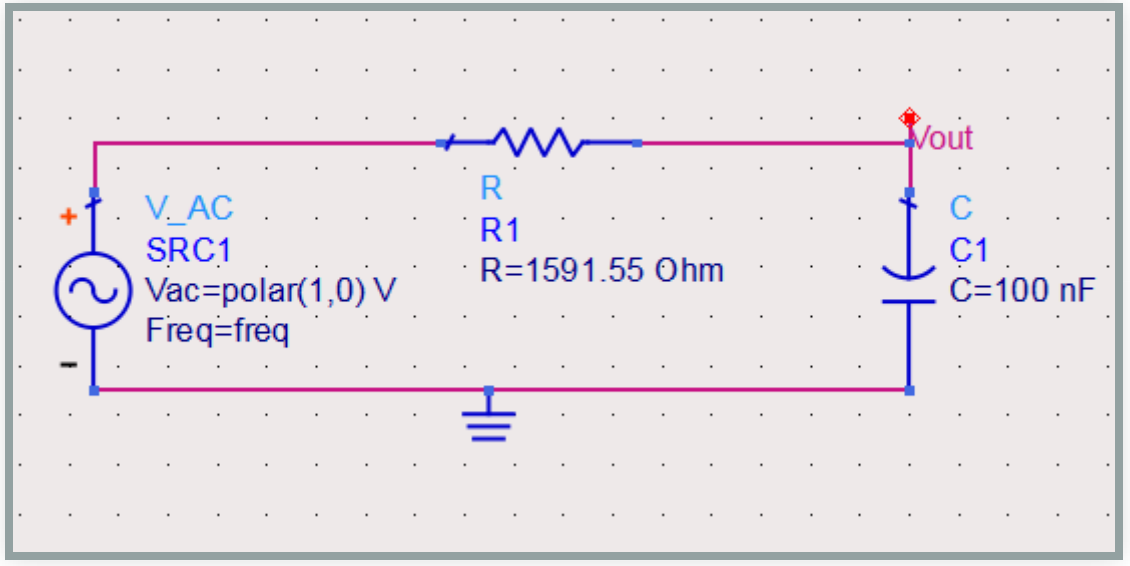


Figura 19 - Filtro passa baixa

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:

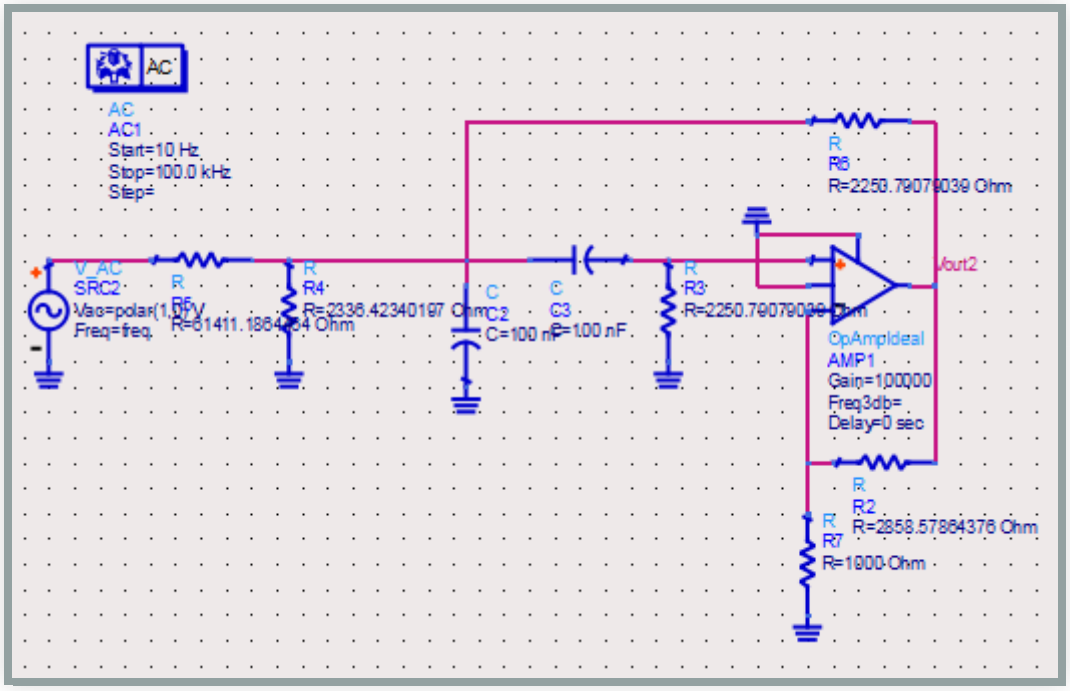


Figura 20 - Filtro Ativo passa faixa

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.

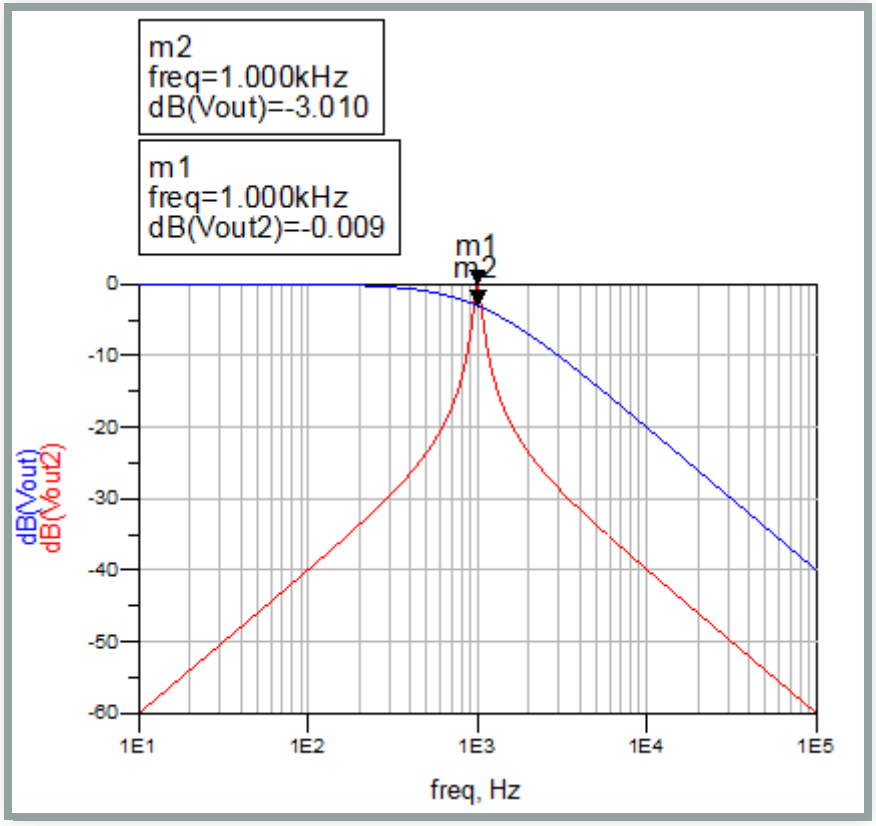


Figura 21 - Comparação da magnitude dos filtros

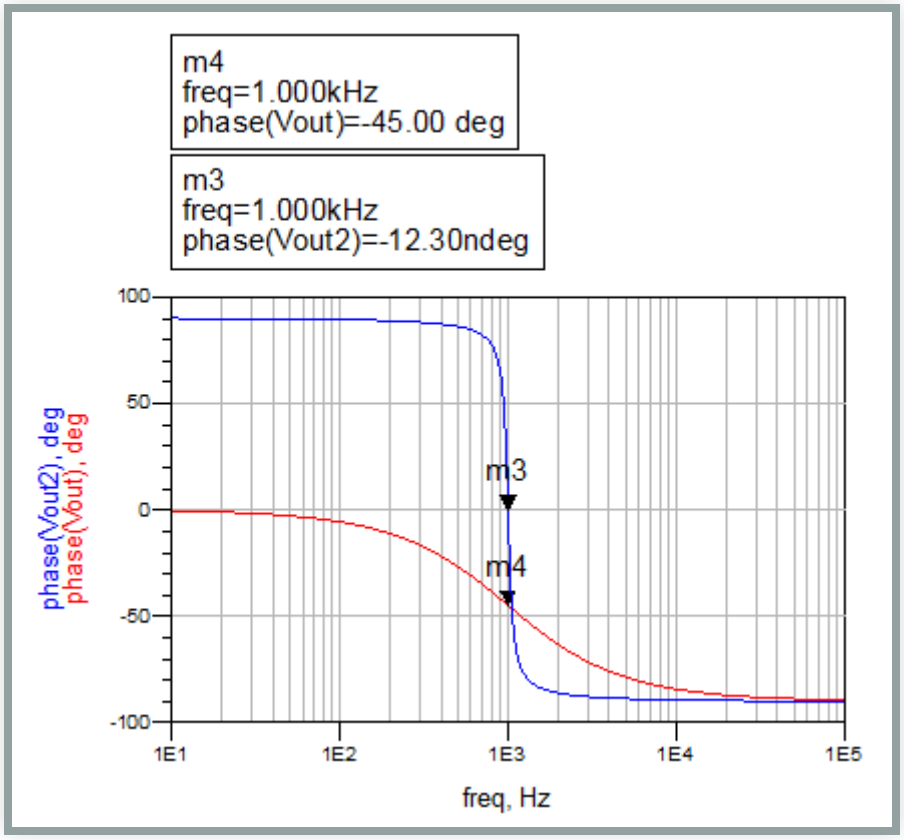


Figura 22 - Comparação da fase dos filtros

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

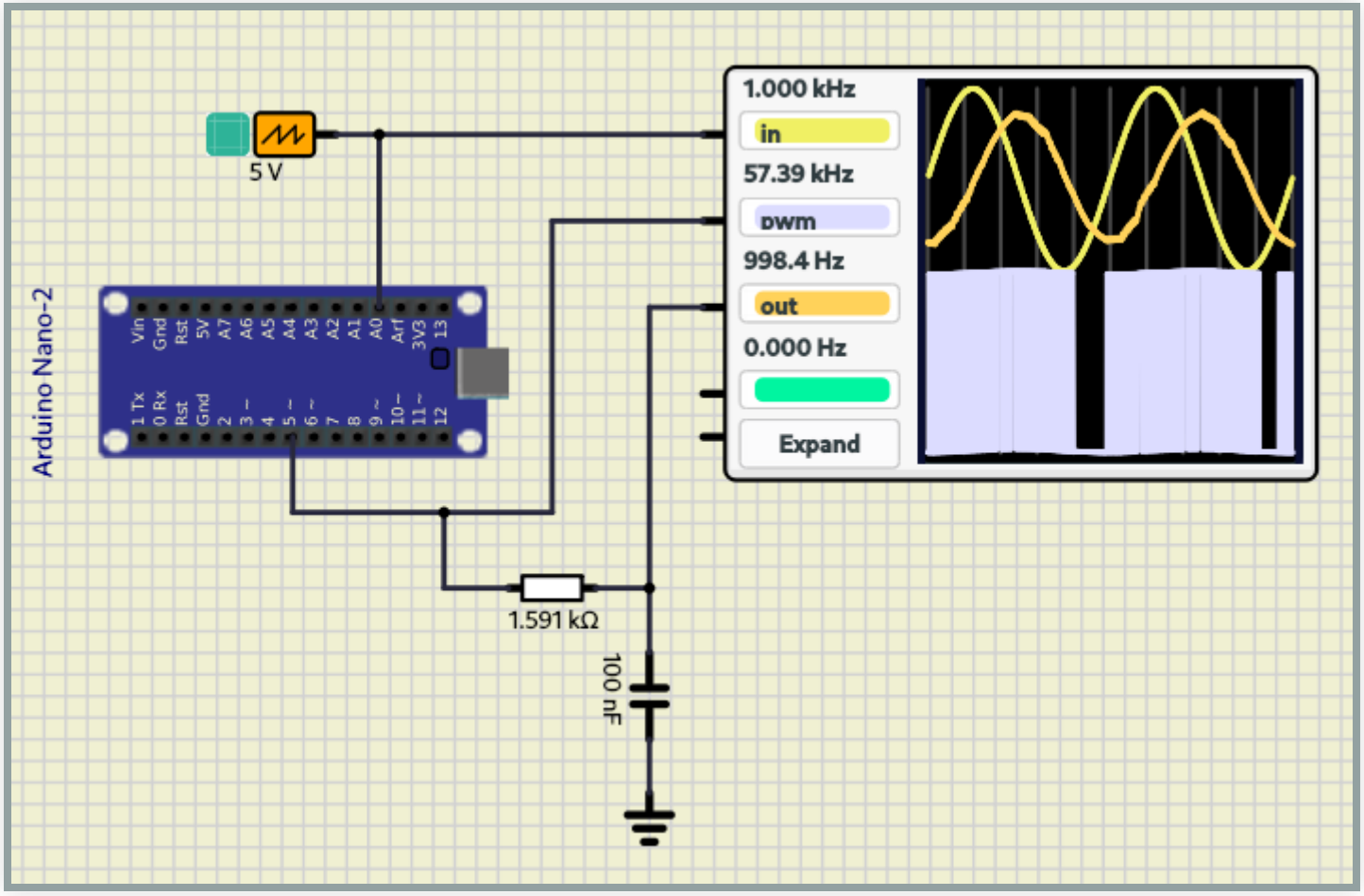


Figura 23 - Simulação do filtro passivo

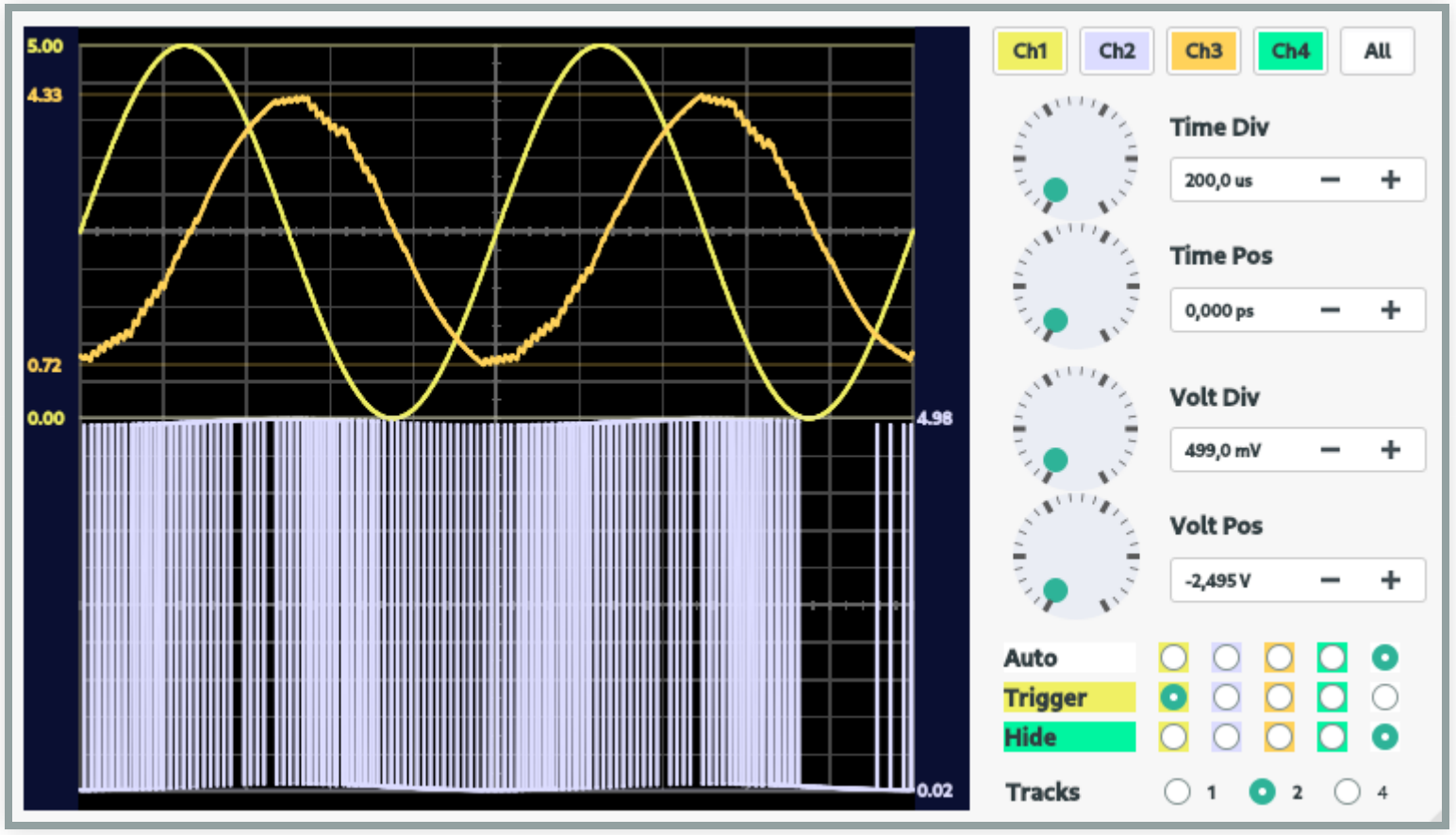


Figura 24 - Osciloscópio do filtro passivo

Filtro ativo passa faixa

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

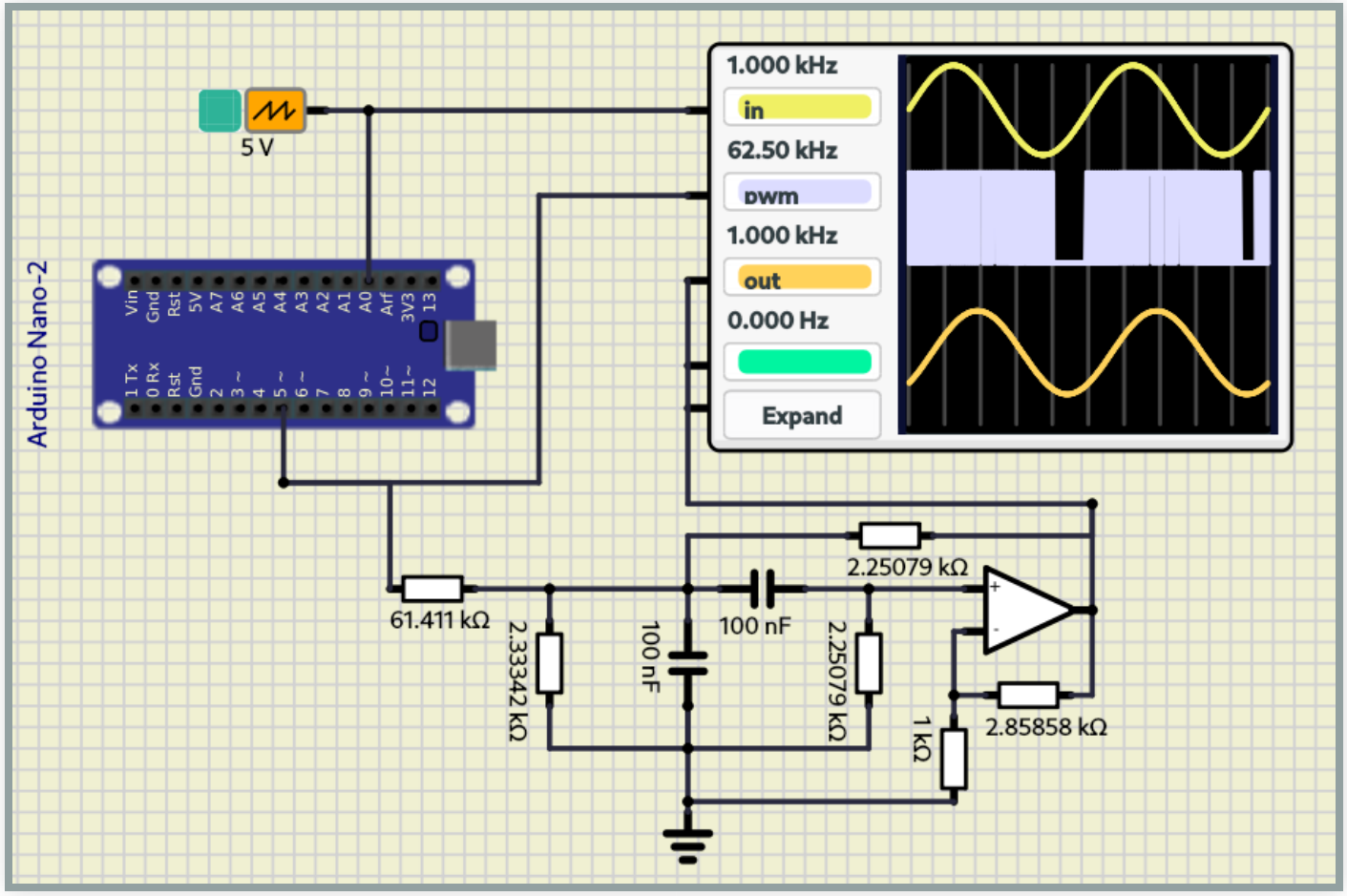


Figura 25 - Simulação do filtro ativo

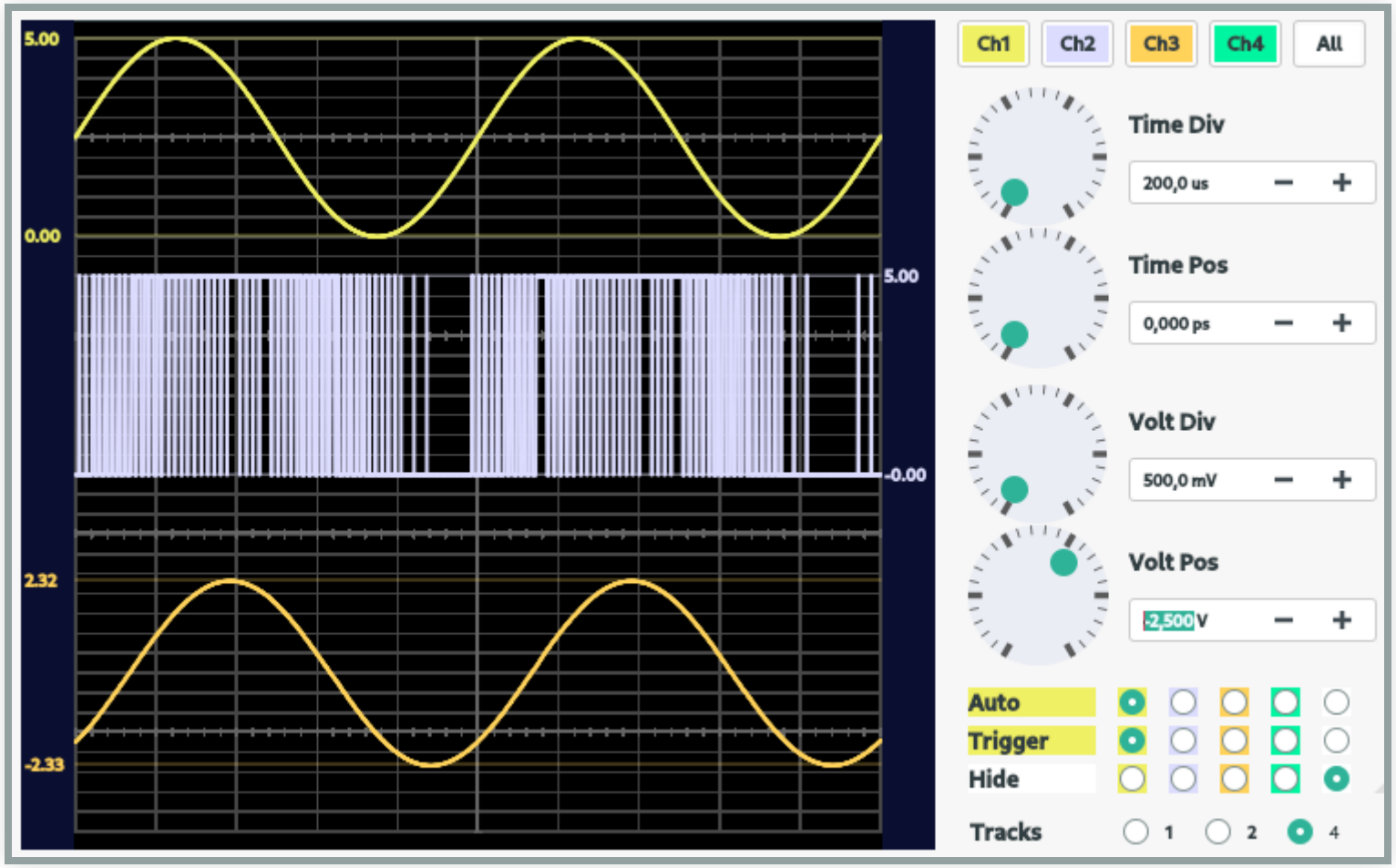


Figura 26 - Osciloscópio do filtro ativo

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.

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

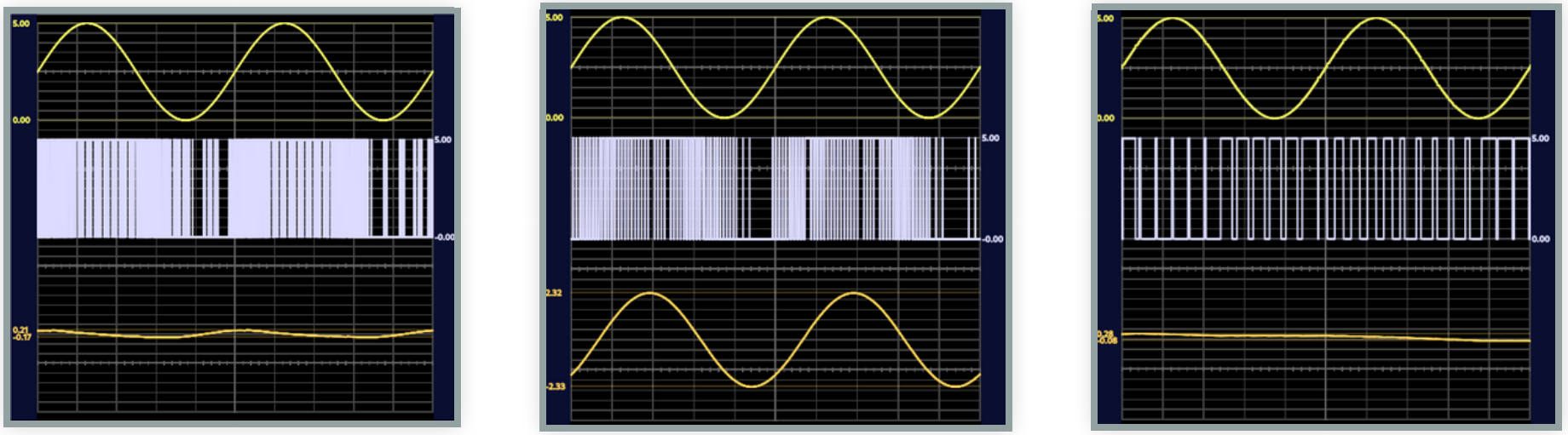


Figura 27 - Atuação do filtro ativo passa faixa para diversas frequências

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.