

Projeto Final de Lab de Eletrônica 1

Henrique da Silva
henrique.pedro@ufpe.br

29 de setembro de 2023

Sumário

1 Introdução

2 Análise preliminar

2.1	O circuito	
2.2	Análise simbólica	
2.2.1	Resolução	
2.2.2	Resistores	

3 Medições em laboratório

3.1	Componentes	
3.2	Configuração Senoidal	
3.3	Configuração Triangular	
3.4	Tensão de saída do Arduino	

4 Análise dos resultados

5 Conclusões

6 Apêndice

1 Introdução

Neste relatório, discutiremos e analisaremos o design de um Conversor Digital-Analógico (DAC) que será implementado com o uso de três amplificadores operacionais em diferentes configurações.

Todos arquivos utilizados para criar este relatório, e o relatório em si estão em: https://github.com/Shapis/ufpe_ee/tree/main/6thsemester/Eletronica1/

O código utilizado para a análise numérica também se encontra no anexo ao final do relatório.

2 Análise preliminar

Utilizarei a biblioteca *sympy* em *Python* para fazer a análise simbólica e numérica do circuito antes de montá-lo fisicamente.

Após terminar as análises compararei os resultados obtidos nas análises numéricas e em laboratório para verificar sua coerência.

Utilizaremos valores de $n = [2, 2]$ para o nosso projeto. Isso vem de operações feitas com os CPFs da dupla.

E teremos todas entradas com corrente limitada a $0.75mA$, e uma tensão máxima de $5V$.

2.1 O circuito

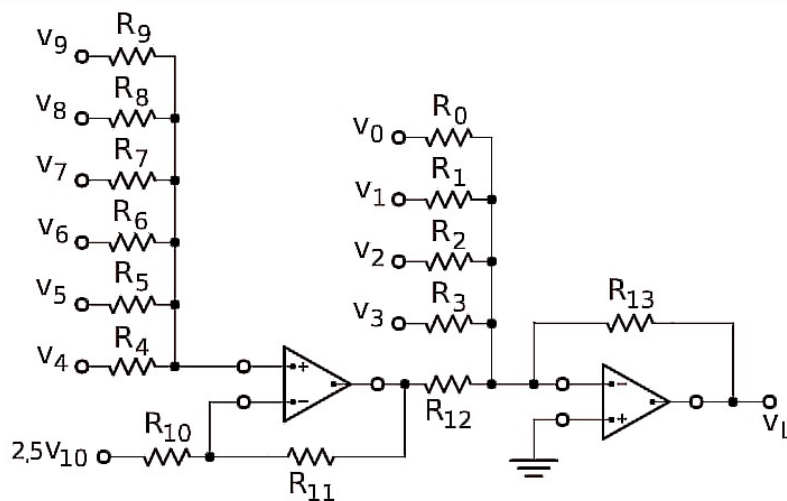


Figura 1: Planta esquemática do DAC de 11-bits.

2.2 Análise simbólica

Podemos realizar a análise do circuito utilizando análise nodal e princípio da superposição. Com isso, obtém-se a seguinte equação que rege a saída V_L do circuito.

$$V_L = - \left(R_{13} \sum_{i=0}^3 \frac{V_i}{R_i} + \frac{R_{13}}{R_{12}} \left(1 + \frac{R_{11}}{R_{10}} \right) R_{eq} \sum_{i=4}^9 \frac{V_i}{R_i} - \frac{R_{11}R_{13}}{R_{10}R_{12}} 2.5V_{10} \right) \quad (1)$$

Comparamos este V_L com uma saída M do DAC.

$$M = r \left(\sum_{i=0}^{n-1} d_i 2^i - d_n 2^n \right) \quad (2)$$

2.2.1 Resolução

Onde r é a resolução do DAC, d_i é o valor do bit i e n é o número de bits do DAC.

Para o nosso projeto, escolheremos uma tensão máxima de $V_L = M = 7V$.

Com as equações 1 e 2, analisaremos o comportamento das equações para diferentes configurações de bits d_i ligados.

Para obter a resolução r , analisamos o valor máximo possível de 2^n para 10 bits, e consideraremos que tanto o valor máximo positivo quanto negativo serão os mesmos, que serão 1024.

Então, com 2, tem-se:

$$r = \frac{7}{2^{10}} \quad (3)$$

2.2.2 Resistores

Analisaremos o comportamento de 1 e 2 para diferentes configurações de bits d_i ligados.

Como queremos ter uma limitação de corrente de $0.75mA$ em todas as entradas, podemos aplicar a lei de Ohm com apenas o d_3 ligado e encontrar o valor de R_3 .

$$\begin{aligned} V_3 &= V_m d_3 = IR_3 \\ I &= 0.75mA > \frac{V_m}{R_3} \\ R_3 &> \frac{5V}{0.75mA} = \frac{20}{3}k\Omega \end{aligned} \quad (4)$$

No caso, escolhemos $R_3 = 100k\Omega$, o que atende à condição de limitação de corrente.

Para manter os pesos vistos na equação eq : M nos resistores deste amplificador operacional, fazemos a mesma lógica para R_2 , R_1 e R_0 , ou seja, dobramos o valor de cada um deles considerando o anterior, e obtemos:

$$\begin{aligned} R_2 &= 2R_3 = 200k\Omega \\ R_1 &= 2R_2 = 400k\Omega \\ R_0 &= 2R_1 = 800k\Omega \end{aligned} \quad (5)$$

Com estes quatro determinados, agora buscamos o R_{13} , vemos que em 1, se zerarmos todas as entradas, exceto uma entrada entre 0 e 3, conseguimos novamente utilizar a relação $V_L = M$ e obter o R_{13} .

$$\begin{aligned} \frac{R_{13}V_m}{R_i r} &= 2^i \\ R_{13} &= 1093.75\Omega \end{aligned} \quad (6)$$

Agora fazemos as restrições de corrente para as entradas de 4 a 9. Para isso, precisamos saber a tensão mínima possível para V_a . Isolamos o V_a pela resolução das equações da análise nodal no *SymPy*, que se encontra no apêndice. E obtemos:

$$V_a = \sum_{i=4}^9 \frac{V_i R_{eq}}{R_i} \quad (7)$$

Os três termos da equação nunca serão negativos, e o único que pode ser zero é o V_i , logo, o valor mínimo de V_a é $0V$.

Daqui podemos fazer a análise com d_i com apenas um ligado entre 4 e 9 e obter os resistores R_4 a R_9 .

Com a mesma lógica que obtemos resistores de 0 a 3, vamos encontrar o R_9 que é o mais significativo, e a partir dele, obteremos os outros 5.

$$R_9 > \frac{20}{3} \Omega \quad (8)$$

Nós escolhemos um valor de $R_9 = 22k\Omega$ que atende à restrição.

Para calcular o R_{10} , R_{11} e R_{12} , utilizaremos as seguintes relações:

$$\begin{aligned} X &= \frac{R_{11}}{R_{10}} \\ Y &= \frac{R_{13}}{R_{12}} \end{aligned} \quad (9)$$

Então, desligando todas as entradas, exceto d_{10} , a partir de 1 e 2, e com as substituições 9, tem-se:

$$\begin{aligned} XY &= 2.5V_0 = 2.5V_m d_{10} \\ XY &= \frac{7}{12.5} = 0.56 \end{aligned} \quad (10)$$

Nas equações 1, 2 e fazendo as substituições 9, fazendo apenas a entrada d_9 estar ligada, e atentando-se que o R_{eq} é a resistência equivalente entre o R_4 até o R_9 e vale 11175Ω , tem-se:

$$\begin{aligned} Y(1 + X)R_{eq}\frac{V_9}{R_9} &= \frac{7}{2} \\ Y(1 + X) &= 1.38 \end{aligned} \quad (11)$$

Resolvendo o sistema

$$\begin{aligned} Y(1 + X) &= 1.38 \\ XY &= \frac{7}{12.5} = 0.56 \end{aligned} \quad (12)$$

Obtém-se: $X = 0.684$ e $Y = 0.818$.

Convertendo de volta o X e Y , tem-se:

$$\begin{aligned} \frac{R_{11}}{R_{10}} &= 0.684 \\ \frac{R_{13}}{R_{12}} &= 0.818 \end{aligned} \quad (13)$$

Como já havíamos obtido o R_{13} previamente, obtemos o valor exato do R_{12} , que é $R_{12} = 1337\Omega$.

Já o R_{11} e R_{10} têm apenas uma proporção entre os dois. Então, escolhemos valores comerciais de $R_{11} = 15k\Omega$ e $R_{12} = 22k\Omega$.

E com isso, determina-se os 15 resistores do projeto.

3 Medições em laboratório

Montaremos os dois circuitos discutidos anteriormente em laboratório e mediremos a tensão de saída em seis configurações de formato de onda e frequências diferentes. Esses parâmetros serão definidos por um dispositivo Arduino com 11 saídas, cada uma com tensão esperada de 5V.

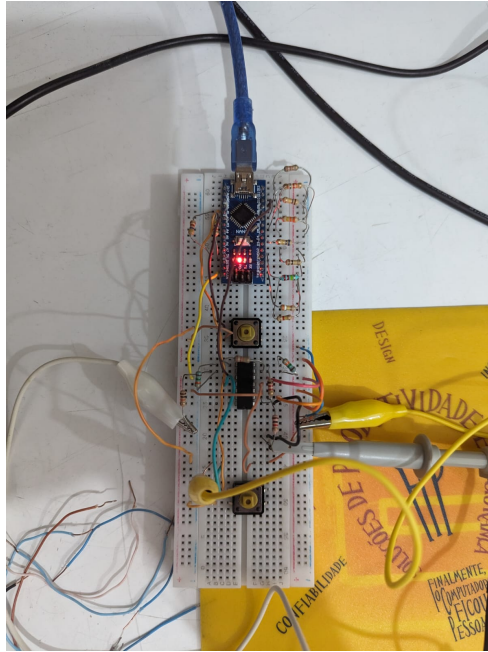


Figura 2: Vista superior do circuito montado em laboratório.

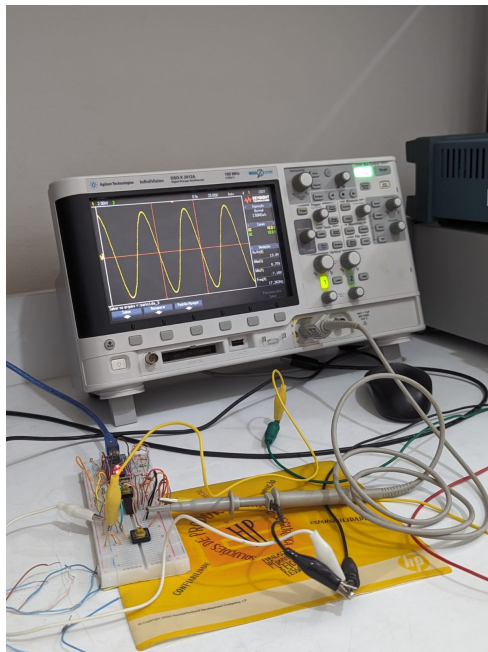


Figura 3: Circuito montado em laboratório e osciloscópio.

3.1 Componentes

Com os valores calculados na secao anterior, tenta-se utilizar componentes comerciais com valores compativeis com os calculados. Os componentes utilizados sao:

resistores	ideais	comerciais	reais
R0	800k	820k	818k
R1	400k	380k	384.5k
R2	200k	220k	221.5k
R3	100k	100k	99.4k
R4	700k	680k	675k
R5	352k	330k	351k
R6	176k	180k	178.2k
R7	88k	82k	80.6k
R8	44k	47k	46.3k
R9	22k	22k	21.89k
R10	22k	22k	22.1k
R11	15k	15k	14.86k
R12	1337	1200	1179
R13	1094	1k	990
R14	10k	10k	9.88k
R15	15k	15k	14.84k

Figura 4: Tabela de componentes utilizados.

3.2 Configuração Senoidal

Medimos três frequências diferentes de configuração Senoidal, e são estas as seguintes:

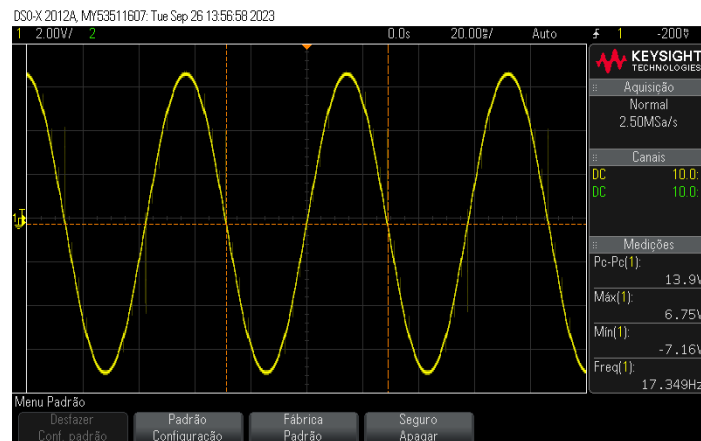


Figura 5: Onda de saída senoidal com frequência de $17.35Hz$.

$$V_{pp} = 13.9V$$

$$V_{max} = 6.75V$$

$$V_{min} = -7.16V$$

$$Freq = 17.35Hz$$

(14)

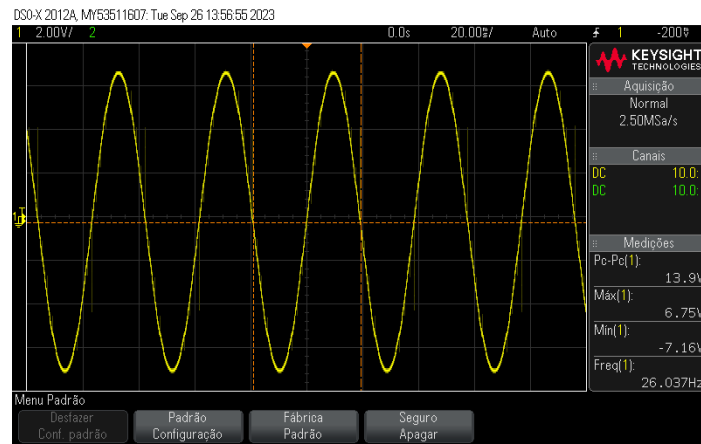


Figura 6: Onda de saída senoidal com frequência de $26.03Hz$.

$$\begin{aligned}
 V_{pp} &= 13.9V \\
 V_{max} &= 6.75V \\
 V_{min} &= -7.16V \\
 Freq &= 26.037Hz
 \end{aligned}
 \tag{15}$$

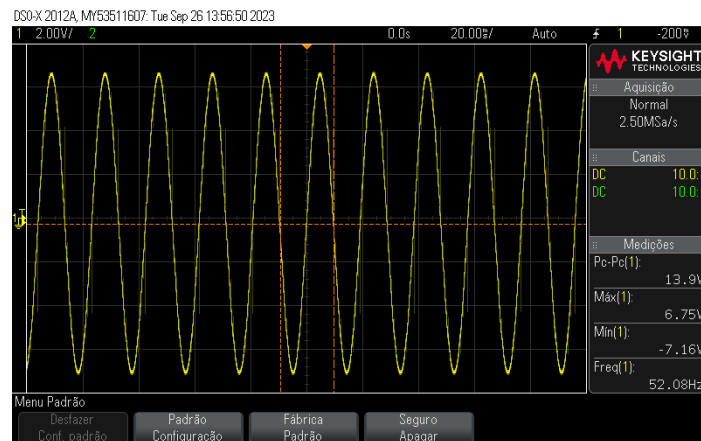


Figura 7: Onda de saída senoidal com frequência de $52.08Hz$.

$$\begin{aligned}
 V_{pp} &= 13.9V \\
 V_{max} &= 6.75V \\
 V_{min} &= -7.16V \\
 Freq &= 52.08Hz
 \end{aligned}
 \tag{16}$$

3.3 Configuração Triangular

Medimos três frequências diferentes de configuração Triangular, e são estas as seguintes:

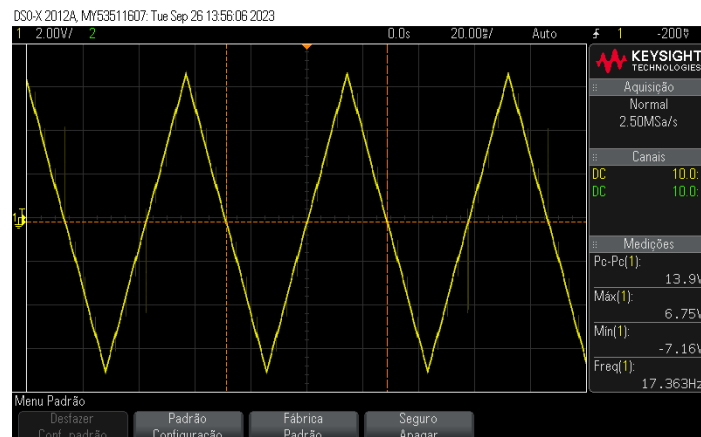


Figura 8: Onda de saída triangular com frequência de $17.363Hz$.

$$\begin{aligned}V_{pp} &= 13.9V \\V_{max} &= 6.75V \\V_{min} &= -7.16V \\Freq &= 17.35Hz\end{aligned}\tag{17}$$

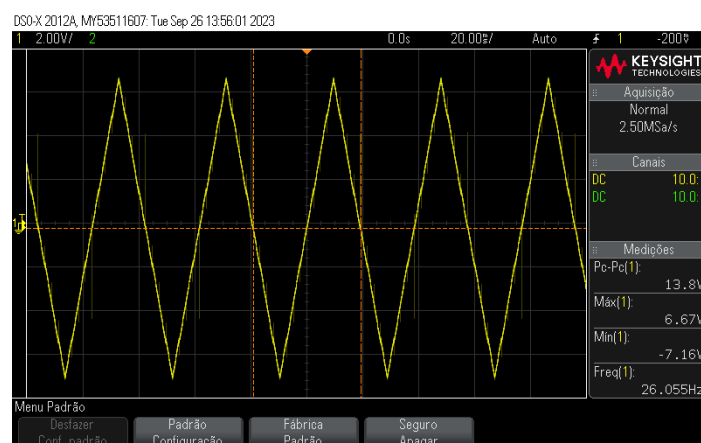


Figura 9: Onda de saída triangular com frequência de $26.055Hz$.

$$\begin{aligned}V_{pp} &= 13.8V \\V_{max} &= 6.67V \\V_{min} &= -7.16V \\Freq &= 26.055Hz\end{aligned}\tag{18}$$

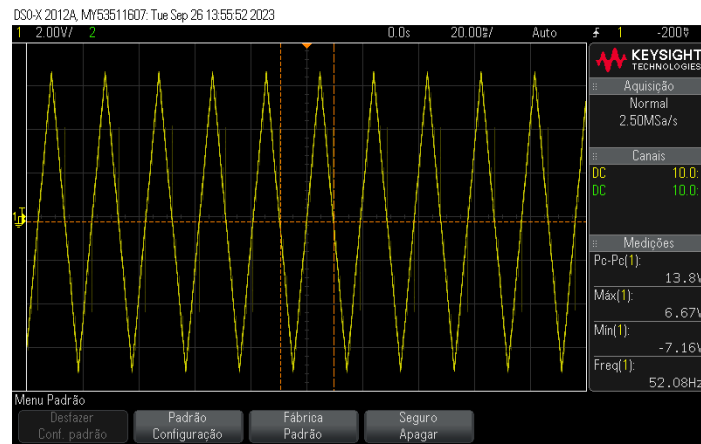


Figura 10: Onda de saída triangular com frequência de $52.08Hz$.

$$\begin{aligned}
 V_{pp} &= 13.8V \\
 V_{max} &= 6.67V \\
 V_{min} &= -7.16V \\
 Freq &= 52.08Hz
 \end{aligned}
 \tag{19}$$

3.4 Tensão de saída do Arduino

Nesta imagem, analisamos o V_m que o Arduino pode fornecer em cada um dos seus pinos de saída, que na fundamentação teórica assumimos que seria de $5V$.

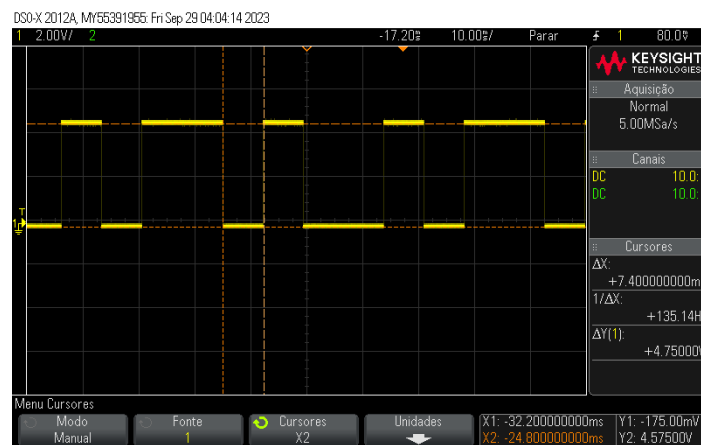


Figura 11: Tensão do pino de saída do Arduino.

Observa-se que o pino de saída do Arduino tem tensão de $4,75V$, e não $5V$ como o esperado.

4 Análise dos resultados

Analisamos a saída V_L do conversor que havíamos projetado com o intuito de estar no intervalo entre $-7V$ e $7V$. Observamos que as saídas medidas reais estão entre $-7.16V$ e $6.75V$. Aqui fazemos a análise de como esses valores se comparam com os valores esperados.

Lembra-se de que as saídas calculadas na fundamentação teórica esperavam um valor de $5V$ na saída do pino do Arduino. O que medimos na realidade foi $4.75V$. Levaremos isso em consideração na análise.

Observando a equação 1, vemos que seu máximo ocorre quando d_{10} está ligado, e todos os outros *bits* estão desligados. Neste caso, a tensão máxima de saída é dada pela seguinte equação:

$$V_{L_{max}} = \frac{2.5 R_{11} R_{13}}{R_{10} R_{12}} V_{10} = \frac{2.5 \cdot 14.86k \cdot 990}{22.1k \cdot 1179} \cdot 4.75 = 6.7V \quad (20)$$

Já a tensão mínima é alcançada quando se ligam os bits entre d_0 e d_9 e se desliga d_{10} .

E obtém-se da seguinte forma:

$$V_{L_{min}} = -R_{13} \sum_{i=0}^3 \frac{V_i}{R_i} - \frac{R_{13}}{R_{12}} \left(1 + \frac{R_{11}}{R_{10}} \right) R_{eq} \sum_{i=4}^9 \frac{V_i}{R_i} = -6.757V \quad (21)$$

Disto, podemos fazer a comparação entre os valores esperados e os valores medidos, e obtemos a seguinte tabela:

	Teórico	Real
$V_{L_{max}}$	$6.7V$	$6.75V$
$V_{L_{min}}$	$-6.757V$	$-7.16V$

Observa-se que os valores estão próximos. Um dos requisitos era que a saída real estivesse com erro de no máximo 1% em relação à saída teórica, e obtivemos os seguintes erros: para $V_{L_{min}}$ o erro foi de 6% e para $V_{L_{max}}$, o erro foi de 0.8%.

5 Conclusões

O projeto foi conduzido com sucesso, e o circuito montado em laboratório apresentou resultados satisfatórios. A saída do circuito foi medida em seis configurações distintas, e os resultados obtidos foram comparados com os valores esperados. Os erros apurados na tensão mínima totalizaram 6%, enquanto na tensão máxima foram de apenas 0.8%. Isso revela que, infelizmente, o requisito de manter o erro abaixo de 1% não foi integralmente cumprido na tensão mínima.

Acredito que este desvio possa ser atribuído a um erro no ganho do circuito de entrada do V_{10} . O fato de o erro ter sido mais pronunciado na tensão mínima é consistente com os valores negativos de tensão de saída decorrentes da decodificação do V_{10} .

Após a correção deste erro específico, possivelmente estaria em conformidade com os requisitos de precisão estabelecidos no projeto.

Os requisitos de corrente máxima nas saídas dos pinos do Arduino foram todos atendidos integralmente, e o circuito não apresentou nenhum comportamento anômalo durante os testes destes.

6 Apêndice

Abaixo se encontra o código utilizado para a análise simbólica e numérica do circuito.

```
#from google.colab import drive
#drive.mount('/content/drive')

# Updating sympy to version 1.12 for faster inverse laplace transform
# You need to restart the environment for the changes to take action
# You need to run it before every session, or else the version will
# be 1.11.1

from google.colab import files

%pip install -q --upgrade sympy

import sympy
sympy.__version__

import matplotlib.pyplot as plt
from sympy import *
from IPython.core.interactiveshell import InteractiveShell

# Allows multiple latex formatted lines
InteractiveShell.ast_node_interactivity = 'all'

# init_session prints the result in latex format, and has some useful presets,
# more information at: https://docs.sympy.org/latest/modules/interactive.html
init_session(quiet=True)
# Allows the use of unicode characters
init_printing(use_unicode=True)
```

```
import matplotlib.pyplot as plt
from sympy import *

# init_session prints the result in latex format, and has some useful presets,
# more information at: https://docs.sympy.org/latest/modules/interactive.html
init_session(quiet=True)
# Allows the use of unicode characters
init_printing(use_unicode=True)

R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13 = \
symbols('R0 R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13')

Va0, Va1, Va2, Va3, Va4, Va5, Va6, Va7, Va8, Va9, Va10 = \
symbols('Va0 Va1 Va2 Va3 Va4 Va5 Va6 Va7 Va8 Va9 Va10')

V0, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10 = \
symbols('V0 V1 V2 V3 V4 V5 V6 V7 V8 V9 V10')

Vo0, Vo1, Vo2, Vo3, Vo4, Vo5, Vo6, Vo7, Vo8, Vo9, Vo10 = \
symbols('Vo0 Vo1 Vo2 Vo3 Vo4 Vo5 Vo6 Vo7 Vo8 Vo9 Vo10')

V10, V11, V12, V13, V14, V15, V16, V17, V18, V19, V110 = \
symbols('V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V110')

array_resistores = [R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13]

Va = [Va0, Va1, Va2, Va3, Va4, Va5, Va6, Va7, Va8, Va9, Va10]

V = [V0, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10]

Vo = [Vo0, Vo1, Vo2, Vo3, Vo4, Vo5, Vo6, Vo7, Vo8, Vo9, Vo10]
```

```
Vl = [Vl0, Vl1, Vl2, Vl3, Vl4, Vl5, Vl6, Vl7, Vl8, Vl9, Vl10]
```

```
# Recebe um array de resistores e calcula a resistencia equivalente
def paralelo(array_resistor, jump):
    sum = 0
    for i in range(len(array_resistor)):
        if i != jump:
            sum += 1/array_resistor[i]

    return 1/sum

# Recebe um vetor de tensoes de entrada e um vetor de resistores ambos de mesmo
# tamanho e retorna o vetor com as tensoes de saida

def divisorTensao(Vin, Vout ,resistores):
    for i in range(len(Vout)):
        Req = paralelo(resistores, i)
        Vout[i] = Vin[i]*Req/(resistores[i] + Req)
    return Vout
```

```
Va[0:4]=[0]*4
Va[4:10] = divisorTensao(V[4:10], Va[4:10], array_resistores[4:10])
Va[10] = 0
Va

for i in range(len(Vo)):
    if i<4:
        Vo[i]=0
    elif i<10:
        Vo[i] = Va[i]*(1 + array_resistores[11]/array_resistores[10])
    else:
        Vo[i] = -2.5*V[i]*array_resistores[11]/array_resistores[10]
Vo

for i in range(len(Vl)):
    if i<4:
        Vl[i] = -V[i]*array_resistores[13]/array_resistores[i]
    else:
        Vl[i] = -Vo[i]*(array_resistores[13]/array_resistores[12])
Vl

#K1 = R13
#K2 = (R13/R12)*(1+(R11/R10))
#S10 = (2.5*R11*R13)/(R10*R12)
```