

Circuito RC:

Arduino+Python+Excel

Este é um projeto que envolver hardware e software.

1º foi montado num protoboard um circuito composto por um capacitor, um resistor conectado a uma fonte de tensão contínua.

2º o circuito foi conectado ao hardware Arduino para que este fizesse a leitura dos dados gerados no circuito.

3º os dados foram sendo coletados e enviados para a porta serial do Arduino.

4º Um script em python teve acesso a porta serial e capturou os dados, gerando um arquivo com extensão .csv.

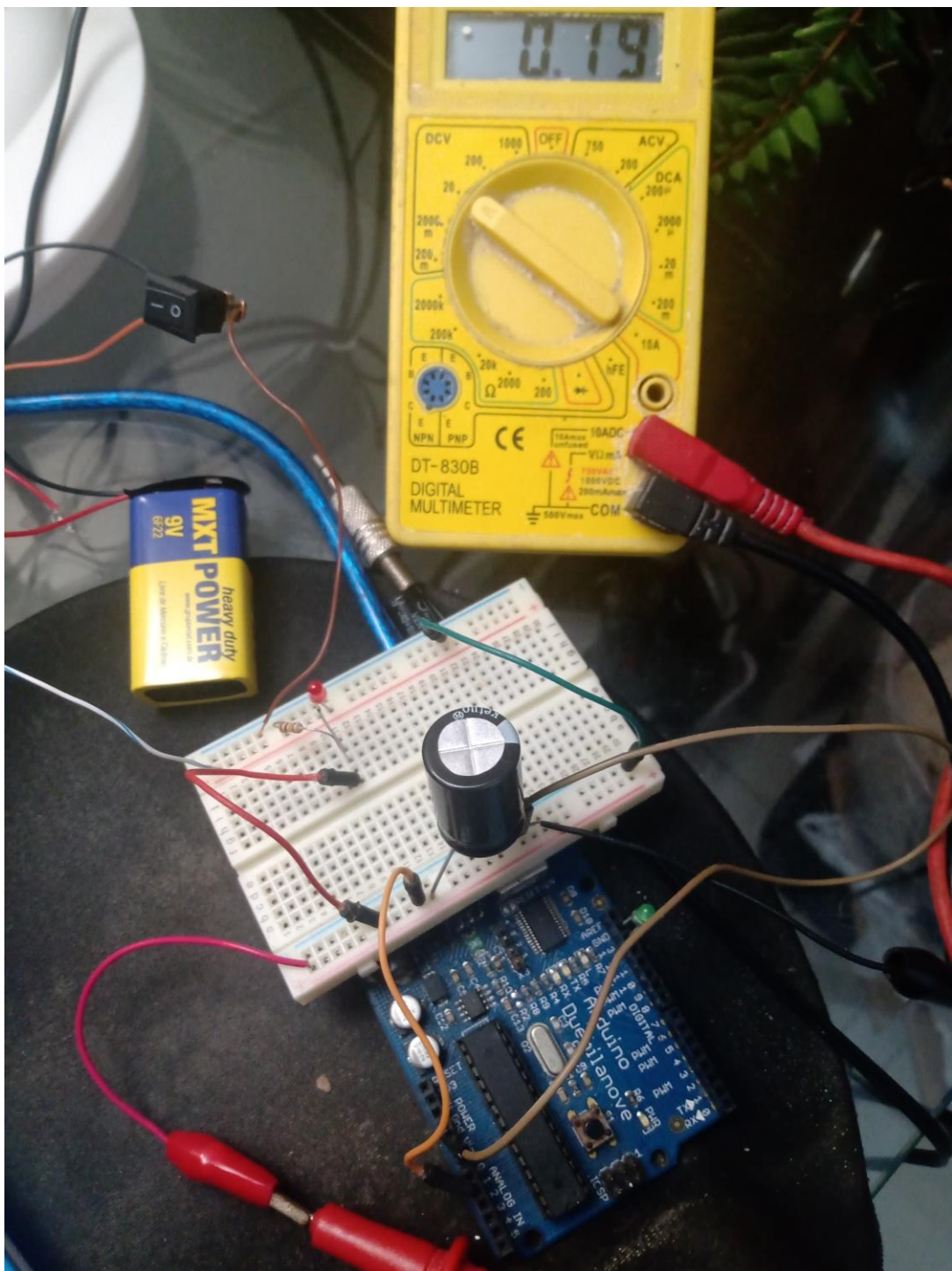
5º o arquivo foi aberto no excel e gerado um gráfico mostrando o comportamento das variáveis tempo e tensão no circuito RC.

6º a função solver no excel foi usada para determinar o 'ajuste' dos pontos a curva teórica.

7º um script em python usando o módulo scipy foi implementado determinando também o gráfico do ajuste da função teórica.

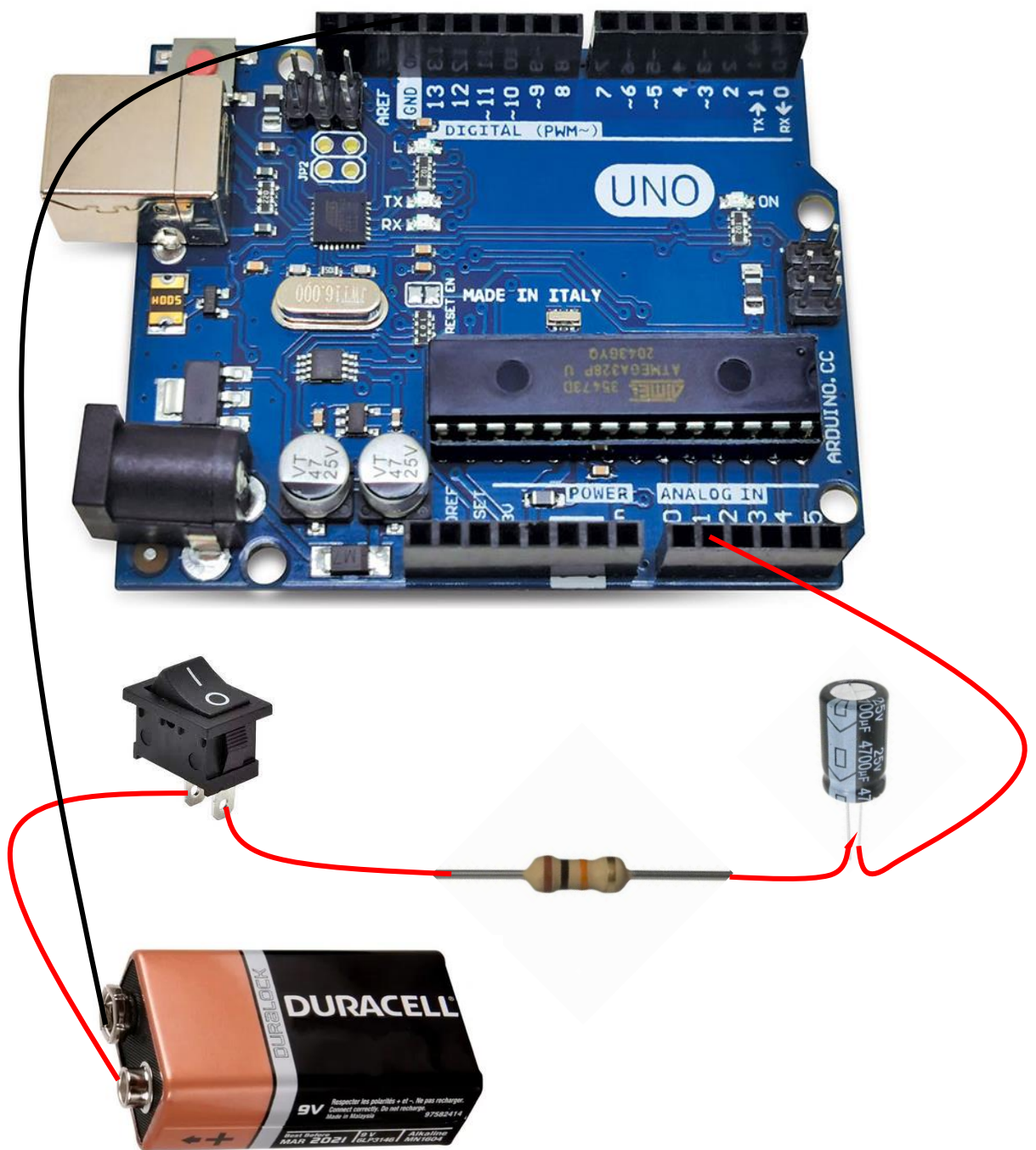
Prof. Dourival Júnior

O circuito real montado:



O circuito RC com fonte de tensão

- Este foi o circuito montado:



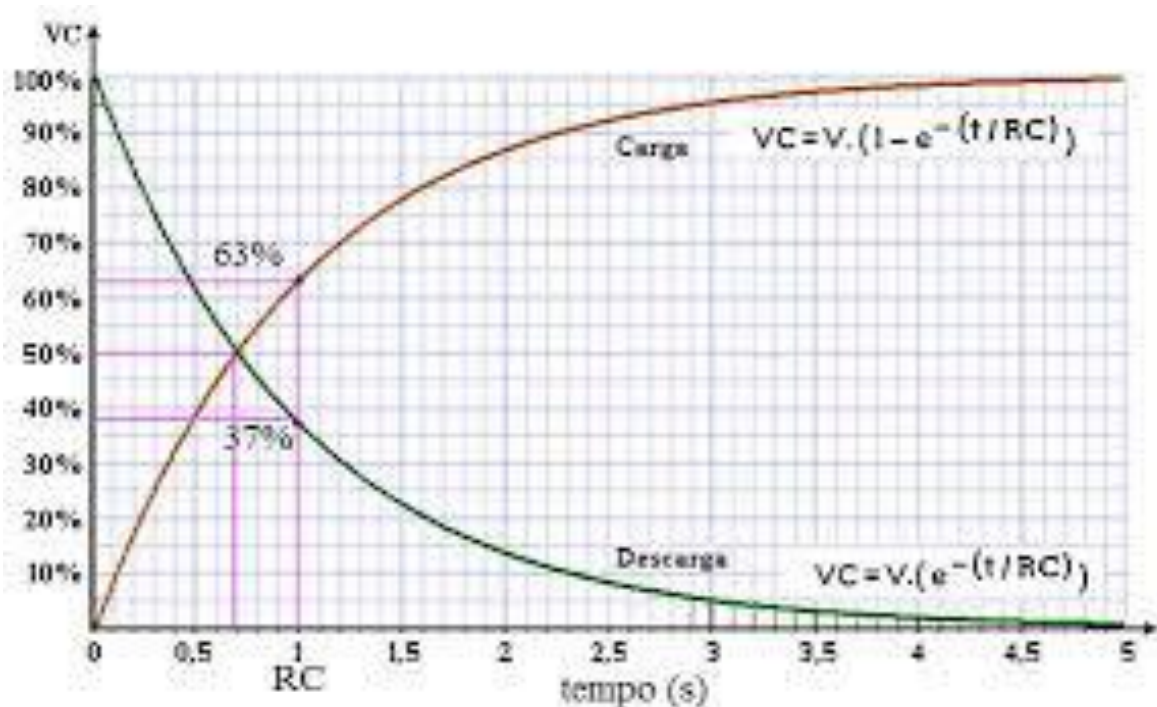
Explicando o Problema:

O circuito RC (resistor, capacitor, fonte de tensão) é um circuito não-linear. O software excel só tem algumas poucas funções instaladas na caixa 'formatar linha de tendência'. Assim, torna-se necessário o uso do solver no excel para o ajuste da função do problema.

- Caixa 'formatar linha de tendência no excel'



- O gráfico com as variáveis tempo e tensão no capacitor não pode ser ajustado por nenhuma das funções da caixa do excel.



Script no Arduino:

Este código faz com que as tensões no capacitor sejam impressas na porta serial COM3 a cada 5s.

// código do arduino para coletar os dados de tempo e tensão elétrica no capacitor de um circuito RC em função do tempo.

```
const int leitura = A1; // pino de entrada para o terminal + do capacitor
float Valor_Lido;
float vsaida;
unsigned long tempo;
float tempo_sec;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  //marca o tempo em milisegundos
  tempo = millis();
  tempo_sec=(tempo/1000.0);
  Valor_Lido = analogRead(leitura); // lê os valores do pino analógico entre 0 e 1023.
  //transforma número em tensão elétrica.
  vsaida=(float(Valor_Lido)*5.620/1024.0);
  Serial.print(tempo_sec),Serial.print(','),Serial.println(vsaida);
  delay(5000);
}
```


Script no Python:

Esse script faz a comunicação serial da IDE do Python com a IDE do Arduino. Depois ele cria um arquivo .csv com os dados gerados do Arduino.

```
import serial
#####script para capturar os dados da porta serial da IDE do arduino,
# abrir um arquivo .csv , criar uma tabela com os dados que serão
usados para cálculos e geração de gráficos #####
porta='COM3'
baund=9600
arquivo='logger1.csv'
ser=serial.Serial(porta,baund)
ser.flushInput()
print('lendo dados da porta serial')

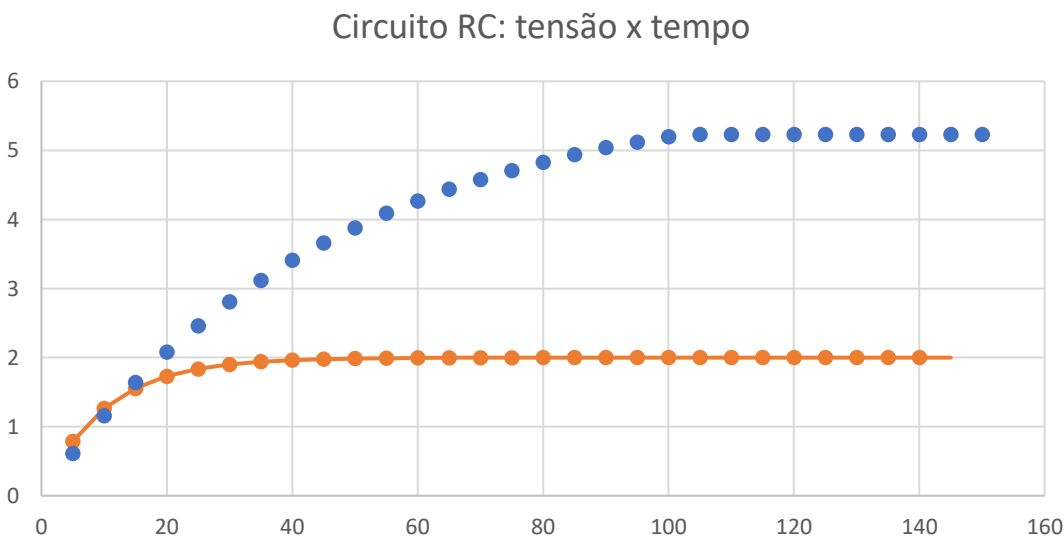
quantidade_de_dados=30
linha=0
while linha<=quantidade_de_dados:
    data=str(ser.readline().decode('utf-8'))
    file = open(arquivo, 'a')
    file.write(data)
    linha=linha+1
print('fim')
file.close()
ser.close()
```

Estudo no Excel:
Com os dados coletados e o
arquivo gerado. Abre-se o excel
para o estudo.

- Tabela com uma parte dos dados coletados de tempo e tensão.

| | |
|-------|------|
| 5 | 0,61 |
| 10 | 1,16 |
| 15 | 1,64 |
| 20 | 2,08 |
| 25 | 2,46 |
| 30 | 2,81 |
| 35,01 | 3,12 |
| 40,01 | 3,41 |
| 45,01 | 3,66 |
| 50,01 | 3,88 |
| 55,01 | 4,09 |
| 60,01 | 4,27 |
| 65,01 | 4,44 |

- Gráficos no excel.
- Pontos em azul são os dados coletados na prática.
- Curva em laranja são os valores calculados a partir de uma função não-linear de ajuste.



Função de Ajuste não-linear no Solver do Excel:

A função para o problema é do tipo:

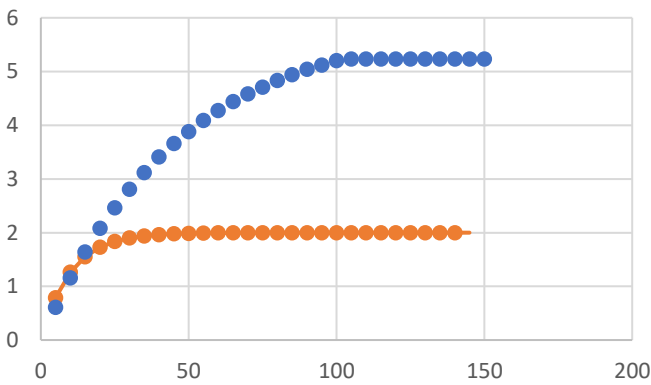
Função de ajuste: $V=A.(1-e^{(-B.t)})$

- Os parâmetros A e B foram inicialmente ‘tomados’ como A=2 e B=0,1 o que resulta na curva teórica com pontos na cor laranja.

Parâmetros de ajuste da função:

| | | |
|----|-----|--|
| A= | 2 | |
| B= | 0,1 | |

Circuito RC: tensão x tempo

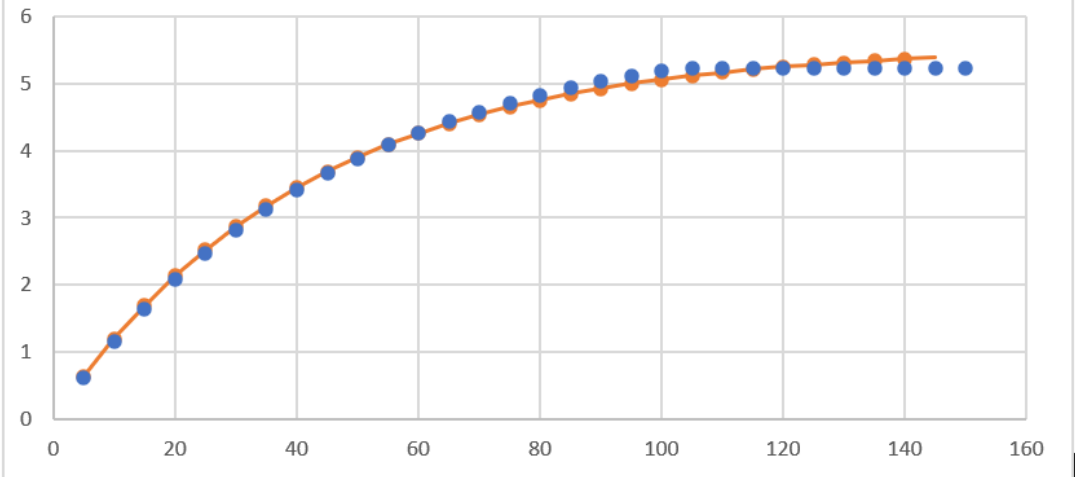


- Após aplicação do solver no excel, encontra-se os valores ajustados dos parâmetros A e B e o ajuste da curva teórica aos dados experimentais coletados.

Parâmetros de ajuste da função:

| | | |
|----|----------|--|
| A= | 5,554675 | |
| B= | 0,024254 | |

Circuito RC: tensão x tempo



Análise dos resultados:

- A tensão na fonte tinha um valor de 5,62V e no ajuste da função tensão foi encontrado o valor 5,55V. Isto dá um erro relativo percentual de 1,2%.
- O valor da constante de tempo teórica do circuito é de 47s e no experimento encontrou-se o valor de 41,2s. Isto dá um erro relativo percentual de 12,3%.
- Para melhorar os resultados torna-se necessário o uso de um resistor com tolerância de 1%, um acionamento mais preciso do script em python para o acesso a porta serial.

Usando o ajuste com um script em Python:

- No python temos que instalar o pacote scipy para fazermos o ajuste da curva aos pontos.
- O Script em python é esse:

```
import scipy
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

df=pd.read_csv('logger2.csv')
df.head()
x=df['tempo'].values
y=df['tensao'].values
def funcao(x,a,b):
    return a*(1-np.exp(-b*x))
popt,pocv=curve_fit(funcao,x,y)
a,b=popt
print(a,b)

#fazendo o gráfico da curva
ajustada
x_new_value = np.arange(min(x),
max(x), 5)
y_new_value =
funcao(x_new_value, a, b)
plt.scatter(x,y,color="green")
plt.plot(x_new_value,y_new_valu
e,color="red")
plt.xlabel('tempo')
plt.ylabel('tensão')
plt.show()
```

Resultado do script no python:

a= 5.821685930912305 b= 0.02728746631379206

