

Klausur Programmieren 1

HAW-Hamburg, Fakultät Technik und Informatik, Department Informations- und Elektrotechnik

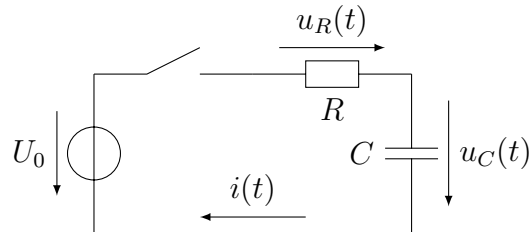
Prof. Dr. Robert Heß, 7.7.2015, Bearbeitungsdauer: 180 Min.

Erlaubte Hilfsmittel: Vorlesungsunterlagen, Lösungen aus dem Praktikum und C/C++ Einführungsbücher.

Ergebnis: von 100 Punkten

Note: Punkte.

1 Einleitung



Ein Kondensator C wird über einen Widerstand R auf die angelegte Spannung U_0 aufgeladen. Es sollen die Spannung am Kondensator $u_C(t)$, die Spannung am Widerstand $u_R(t)$ und der Strom durch den Stromkreis $i(t)$ über einen betrachteten Zeitraum t_{Dauer} berechnet und ausgegeben werden. Die benötigten Formeln lauten:

$$u_R(t) = U_0 e^{-t/\tau} \quad u_C(t) = U_0 (1 - e^{-t/\tau}) \quad i(t) = \frac{U_0}{R} e^{-t/\tau} \quad \text{mit } \tau = RC$$

Zu Beginn werden vom Benutzer die Kapazität C in Farad, der Widerstand R in Ohm, die angelegte Spannung U_0 in Volt und das Zeitintervall für die Berechnung t_{Dauer} in Sekunden abgefragt. Die Berechnungen sollen in 100 Schritten erfolgen, was zu 101 Zeitpunkten führt:

$$t_k = k t_{\text{Dauer}}/100 \quad \text{für } k = 0, 1, 2, \dots, 100$$

Die Ergebnisse sollen tabellarisch in einen zweidimensionalen Vektor gespeichert und später ausgegeben werden. In jede Zeile sollen die Werte für t , $u_R(t)$, $u_C(t)$ und $i(t)$ gespeichert werden.

2 Programmieraufgaben

Aufgabe 1 (5 Punkte)

Legen Sie im Hauptprogramm alle benötigten Variablen an. Vermeiden Sie unnötige Variablen und Redundanzen.

Aufgabe 2 (10 Punkte)

Fragen Sie im Hauptprogramm die Werte für R , C , U_0 und t_{Dauer} vom Benutzer ab. Verwenden Sie eine geeignete Funktion, um eine sichere Abfrage zu gewährleisten. Folgende Wertebereiche sind erlaubt: C : 1 pF – 10 F, R : 1 Ω – 1 G Ω , U_0 : 1 mV – 1 kV, t_{Dauer} : 1 μ s – 1000 s.

Aufgabe 3 (15 Punkte)

Erstellen Sie eine Funktion mit Namen `TableBerechnen(...)` zum Berechnen aller Werte in der Tabelle.

Aufgabe 4 (15 Punkte)

Erstellen Sie eine Funktion mit Namen `TabelleAusgeben(...)`, welche die Tabelle auf dem Bildschirm ausgibt. Achten Sie auf eine ordentliche Ausrichtung der Spalten.

Aufgabe 5 (20 Punkte)

Fügen Sie die Programmstücke zu einem lauffähigen Programm zusammen:



Achten Sie auf einen guten Programmierstil (Vermeidung globaler Variablen, sinnvolle Variablennamen, Quellcode einrücken und kommentieren, keine absoluten Sprünge mit `goto`, keine Warnungen vom Compiler etc.).

Aufgabe 6 (10 Punkte)

Erstellen Sie für die Funktion `TableBerechnen(...)` aus Aufgabe 3 ein Aktivitätsdiagramm. Besonders Schleifen sollen dargestellt werden.

3 Verständnisfragen

Aufgabe 7 (3 Punkte)

Sie wiederholen einen Vorgang mindestens einmal, bis der Benutzer die Taste `ESC` betätigt. Welche Schleife ist zu bevorzugen? `for()` ... `while()` ... `do ...`

Aufgabe 8 (5 Punkte)

Warum sollten globale Variablen vermieden werden?

Aufgabe 9 (5 Punkte)

Was bedeutet ein Rechteck mit abgerundeten Ecken in einem Aktivitätsdiagramm?

Aufgabe 10 (6 Punkte)

Welchen Wert haben folgende Zahlen im Dezimalsystem?

a) $10011010_2 \Rightarrow$

b) $432_8 \Rightarrow$

c) $1AF_{16} \Rightarrow$

Aufgabe 11 (6 Punkte)

Welche Datentypen haben folgende Ausdrücke?

7-3f	
2*1.2	
2/3	

0xefa	
a>b?1.1:1e3	
3&5	