Testprotokollierung

Programmtest mit Standardwerten

Um zu prüfen, ob die Hauptfunktionalität des Programms (Berechnung der Werte und Darstellung des Diagramms) korrekt funktioniert, haben wir 2 Testdateien mit vordefinierten Kondensatoren angelegt. Wenn die Werte der Kondensatoren noch richtig berechnet werden und die Darstellung korrekt aussieht, können wir annehmen, dass der - eventuell geänderte - Code ordnungsgemäß funktioniert.

Die Formel zur Berechnung der Zeitkonstante T:

$$T = R * C$$

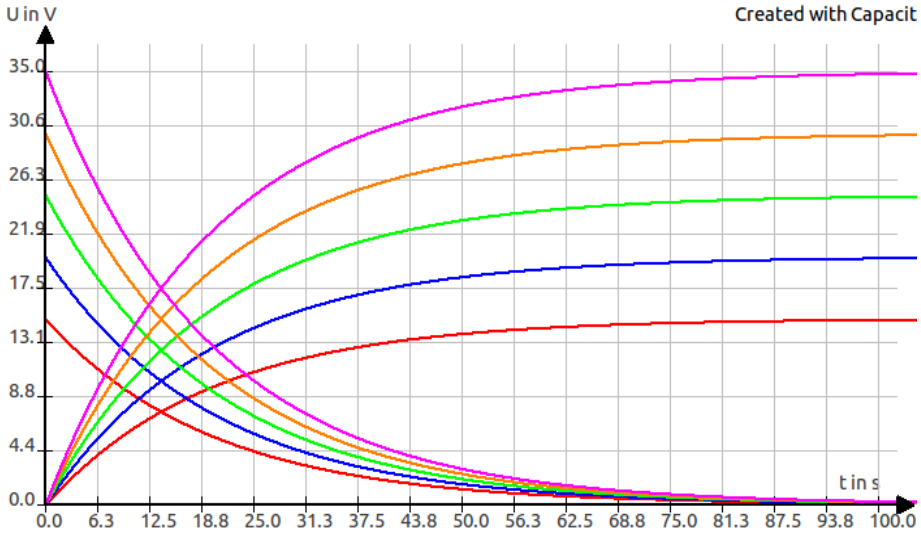
für Beispiel-Kondensator 1:

$$T1 = 1\ M\Omega * 20\ \mu F$$
$$= 1*(10^6)\Omega * 2 * (10^-5)F$$
$$= 2 * 10^1 s$$
$$= 20s$$

Die Aufladezeit beträgt 5T, also 5 * 20s = 100s. Die Werte in der Tabelle bestätigen dies.

Name	Kapazität C	Widerstand Rc	Spannung U	Zeitkonst. T	Aufladezeit
Rot	20 µF	1 MΩ	15 V	20 s	100 s
Blau	20 µF	1 MΩ	20 V	20 s	100 s
Grün	20 µF	1 MΩ	25 V	20 s	100 s
Orange	20 µF	1 MΩ	30 V	20 s	100 s
Lila	20 µF	1 MΩ	35 V	20 s	100 s

Bei 1T beträgt die Spannung im Verlauf etwa 63% des Maximalwerts. Das Diagramm zeigt auch hier die korrekte Funktionsweise des Programms: Maximaldauer 100s, Maximalwert 35V und bei etwa 1T liegt der Wert bei besagten 63%. Wenn das Programm richtig funktioniert, müssten sich außerdem beide Kurven in der Mitte treffen und gespiegelt verlaufen.

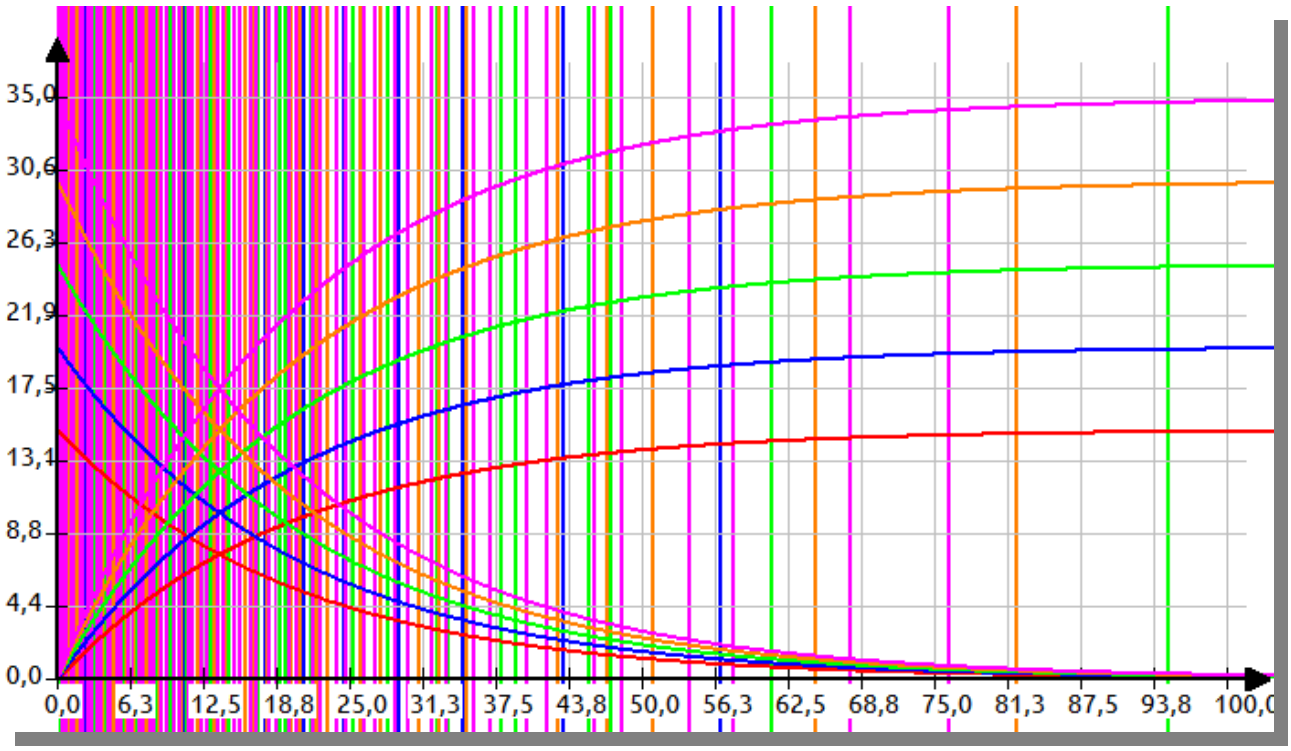


Wir können also annehmen, dass das Programm ordnungsgemäß funktioniert und schließen einen Fehler bei der grundsätzlichen Funktionalität dadurch aus.

Fehlerfälle im Betrieb:

FF1(Version 1.1.3.8):

Wenn man gewisse Werte verändert, eine Datei neulädt oder die Anzeige ändert, so ändert/aktualisiert sich auch gleichzeitig die Anzeige des Diagramms. So zumindest die Theorie. Tatsächlich passierte es auf auf Windows als auch auf Linux immer mal wieder, dass der alte Inhalt nicht gelöscht wurde, bevor der neue gezeichnet wurde. Dieser Fehler tritt auf. obwohl wir in unserem Programm bei jeder Änderung ein Refresh-Event aufrufen, welches den vorherigen Inhalt löscht. Mögliche Ursache hierfür wäre ein Buffer-Fehler der genutzten Komponente. Dieser Fehler tritt häufig nur 1 mal und nach der ersten Nutzereingabe, die die Grafik verändert, auf (z.b. auch beim Zoomen).



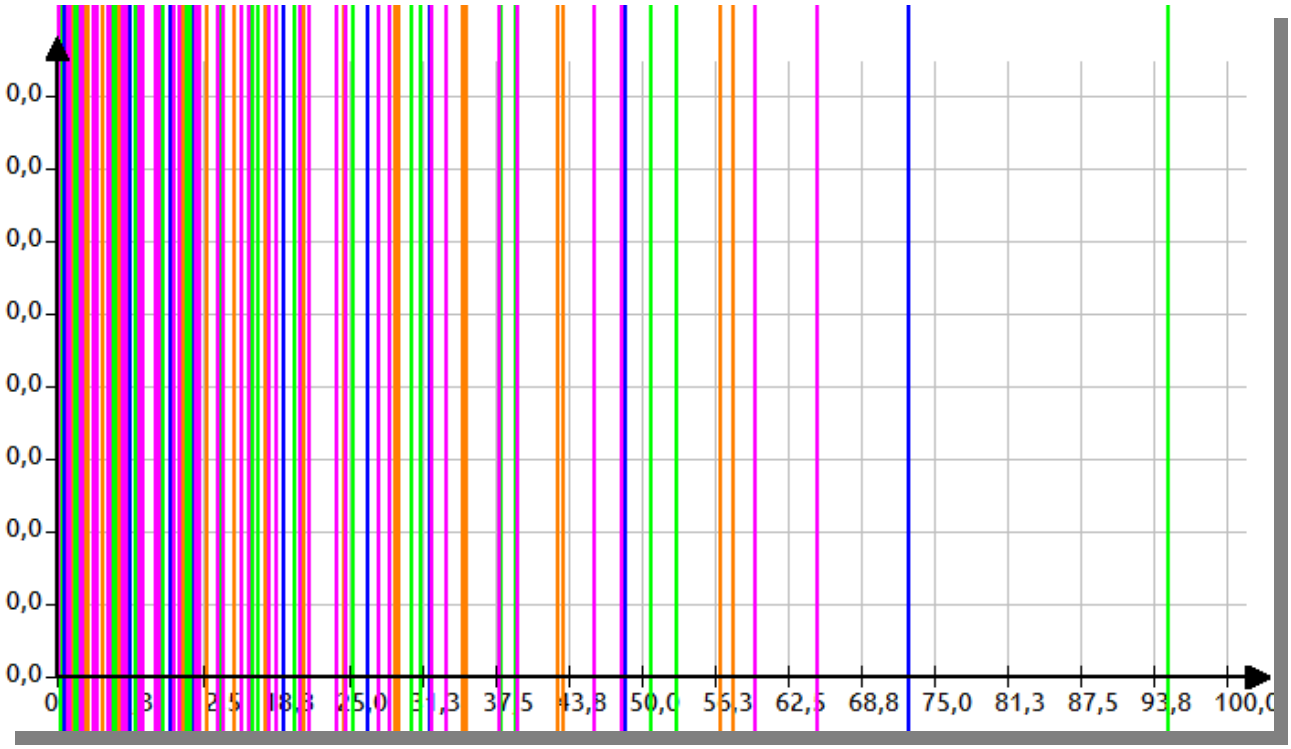
(altes und neues Bild überlagern sich durch fehlerhaften Refresh)

FF2 (Version 1.1.4.6 Windows):

Ein weiterer Fehler ist beim Testen auf Windows aufgetreten: Wenn man schnell auf die Häkchen in der Checkliste klickt, um einen Kondensator aus- / oder abzuwählen, dann springt die Auswahl manchmal schnell nach oben und statt dem eigentlichen Eintrag, den man anklicken wollte, klickt man den ersten an. Dadurch klickt man häufig auf den falschen Eintrag. Unter Linux mit GTK+ kam dies nicht vor. Deshalb ist vermutlich die Windows-API für dieses „Fehlverhalten“ verantwortlich. Versuche, das Verhalten auf
Windows zu kontrollieren, sind gescheitert.

FF3 & FF4 (Version 1.1.3.8):

In der genannten Version ist es möglich, den Zoom-Bereich so weit zu verkleinern, also ranzuzoomen, dass das Programm irgendwann komplett falsche Diagramme zeigt, die sogar immer mehr werden, je dichter man ranzoomt. Grund dafür ist ein Variable-**Underflow**, also ein Unterlauf eines Wertes, der mit der Darstellung zu tun hat. Es gibt nämlich eine Variable in diesem Programm, die angibt, welchen Wert in V oder s ein einzelner Pixel repräsentiert. Wird dieser Wert zu klein, entsteht irgendwann ein Underflow und der Wert und damit die Darstellung, die auf diesem beruht, sind unbrauchbar. Das Ergebnis einer solchen Berechnung ist dann also so klein geworden, dass es nicht mehr darstellbar ist und der Wert damit nicht fast 0, sondern gleich 0 wird.



(Fehlerhaftes Diagramm)

Ähnliches kann bei einem **Überlauf** passieren (je nach Architektur). Wenn wir z.b. einen 8 Bit-Wert 1111 1110 = 254 haben und wir inkrementieren diesen, dann ergibt sich erstmal der Wert 1111 1111 = 255. Wenn wir auf diesen jetzt noch einmal inkrementieren, erhalten wir (**1**) 0000 0000. Der Übertrag von 1 fällt also nicht mehr in das Byte, in dem unser Wert im Speicher eigentlich steht. Aus 255 wird plötzlich 0. Auch dies kann zu merkwürdigen Ergebnissen führen, wenn sie nicht bekannt sind oder nicht abgefangen werden.
Wenn man Auf der X-Achse ganz weit rauszoomt, passiert eben dieser Effekt.

Eine Lösung für dieses Problem könnte sein, komplett auf absolute Zahlenbeträge zu verzichten. Zahlen könnten dann einfach z.b. 1,56 * 10^3, also im wissenschaftlichen Format, dargestellt werden. Die Zahl besteht dann aus einem Gleitkommawert und einem Exponenten. Allerdings geht unter Umständen die Genauigkeit verloren und ein Zahlenwert beansprucht mehr Platz im Speicher.

Alle weiteren Fehler, z.b. durch Eingaben des Nutzers, werden möglichst durch das Programm selbst abgefangen und führen zu keinem Absturz. Während der Tests ist das Programm bisher nicht abgestürzt durch eine Fehlermeldung, die nicht behandeln wird.