Rechnerpraktikum zu Kapitel 1

Wiederholung C-Programmierung

Sicherheitsunterweisung

Klausur Ingenieurinformatik 1, SoSe 2016

Export/Import von Qt-Creator-Projekten

Projekte mit mehreren Quelldateien

Beispiel: Elektronik, gedämpfte Schwingung

P1.2. Sicherheitsunterweisung

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN MÜNCHEN

Sicherheitsunterweisung:

- Es gilt die Laborordnung
- In den Laboren nicht Rauchen, Essen und Trinken
- Fluchtwege aus dem Labor auf den Flur ins Treppenhaus
- Grüne Fluchtwegemarkierungen an der Flurdecke
- Feuerlöscher auf dem Flur, Feuermelder in beiden Treppenhäusern
- Im Brandfall keinen Aufzug benutzen (möglicher Stromausfall)
- Im Brandfall die Fenster geschlossen halten
- Informationen an den Türen:
  Verhalten im Brandfall, Rufnummern für den Notfall, erste Hilfe

Ein Unfall – was ist zu tun?

- Verbandskästen in den Räumen B362 , B372 , B0055 (Sekretariat)
- Notausschalter sind in allen KCA-Laboren vorhanden

Bei Hard- und Softwareproblemen (Login nicht möglich, Fragen zu Netzwerklaufwerken/Backups usw.) hilft Herr Schneider, Raum B350

Rechnerpraktikum zu Kapitel 1

Wiederholung C-Programmierung

Sicherheitsunterweisung

Klausur Ingenieurinformatik 1, SoSe 2016

Export/Import von Qt-Creator-Projekten

Projekte mit mehreren Quelldateien

Beispiel: Elektronik, gedämpfte Schwingung

P1.4. Klausur Ingenieurinformatik 1, SoSe 2016

Programmieren Sie die drei Aufgaben der Ingenieurinformatik-1-Klausur aus dem SoSe 2016. Sie finden die Klausur hier: http://kuepper.userweb.mwn.de/informatik/pruef-ss16fa2.pdf

Aufgabe 1: (ca. 21 Punkte)

1.1. Programmieren Sie eine Funktion **int ggT(int a, int b)** zur Berechnung des größte Teilers (ggT) von zwei Integer-Werten a und b. Zur Berechnung ist der „euklidisch zu verwenden:

Solange b≠0 ist...
Hilfsvariable tmp = a modulo b
a = b
b = tmp
Der gesuchte ggT befindet sich nun in der Variablen a

1.2. Erstellen Sie ein Hauptprogramm **int main(void)**, welches die folgenden Aufgabe

Rechnerpraktikum zu Kapitel 1

Wiederholung C-Programmierung

Sicherheitsunterweisung

Klausur Ingenieurinformatik 1, SoSe 2016

Export/Import von Qt-Creator-Projekten

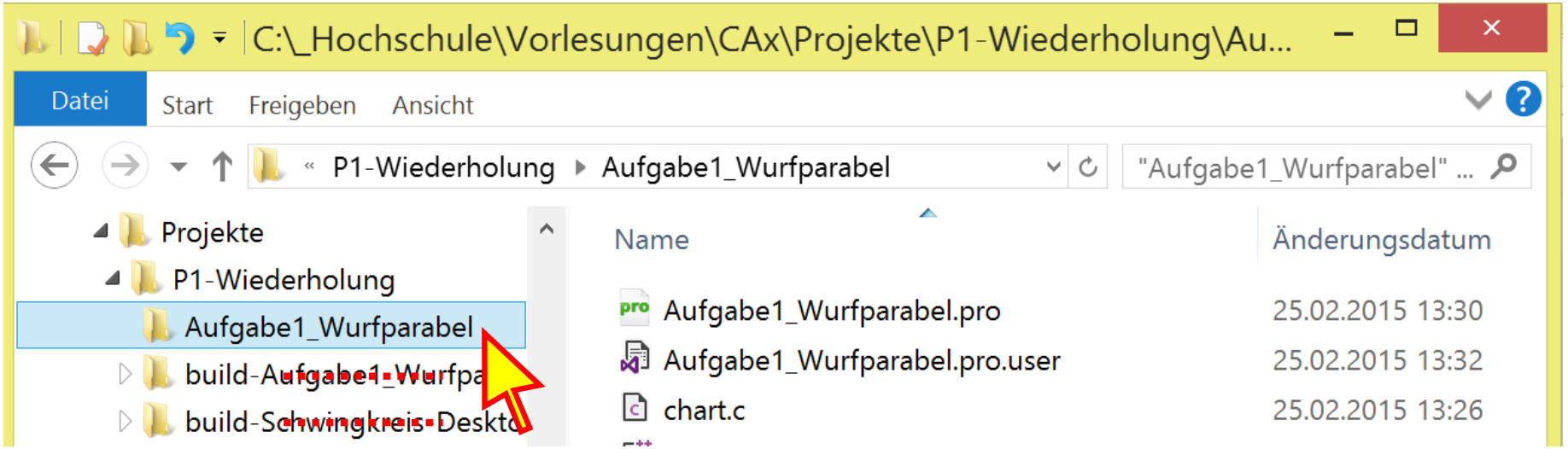
Projekte mit mehreren Quelldateien

Beispiel: Elektronik, gedämpfte Schwingung

P1.6. Export/Import von Qt-Creator-Projekten

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN MÜNCHEN

Frage: Wie kopiere ich ein C++-Projekt auf den USB-Stick, um es zur Weiterverarbeitung mit nach Hause zu nehmen?



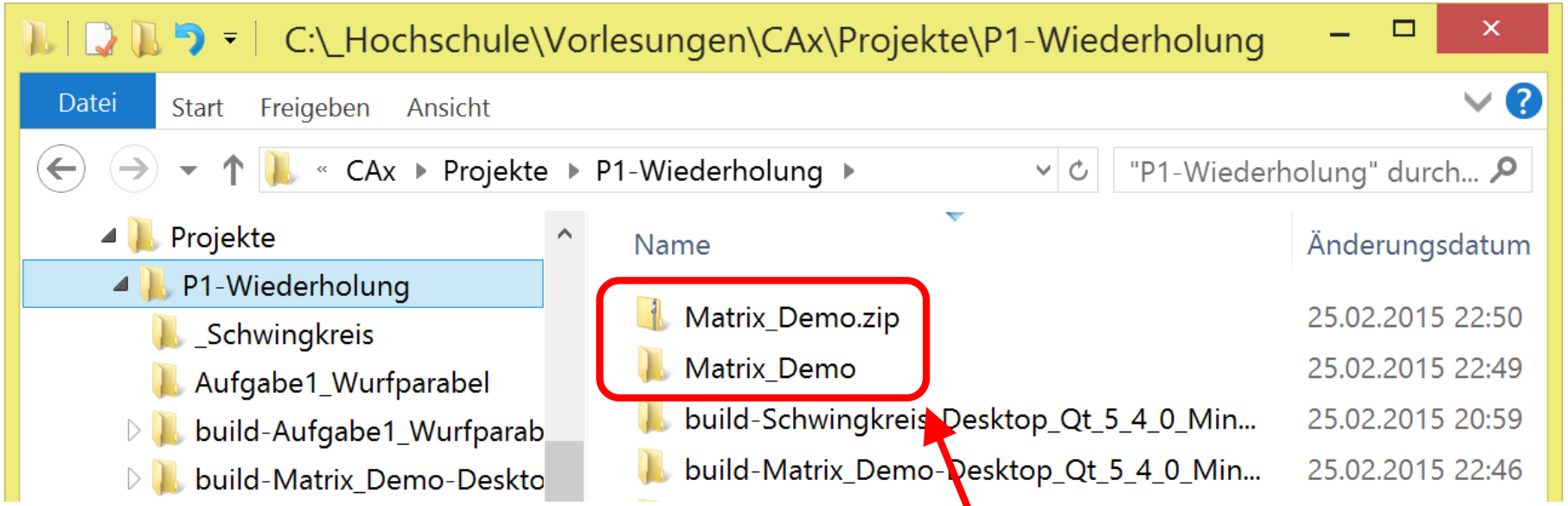
a) Qt Creator legt für jedes C-Projekt ein eigenes Unterverzeichnis an. Schließen Sie zuerst den Qt Creator.

b) Nun öffnen Sie den Windows Explorer und kopieren das gewünschte Projektverzeichnis mit dem kompletten Inhalt auf Ihren USB-Stick (eventuell vorher ZIP-Archiv erstellen).

c) Das „build-xxx-Verzeichnis“ muss nicht kopiert werden!

P1.7. Export/Import von Qt-Creator-Projekten

Frage: Wie kopiere ich ein C++-Projekt vom USB-Stick auf meinen Rechner, um es dort mit Qt Creator weiter zu bearbeiten?

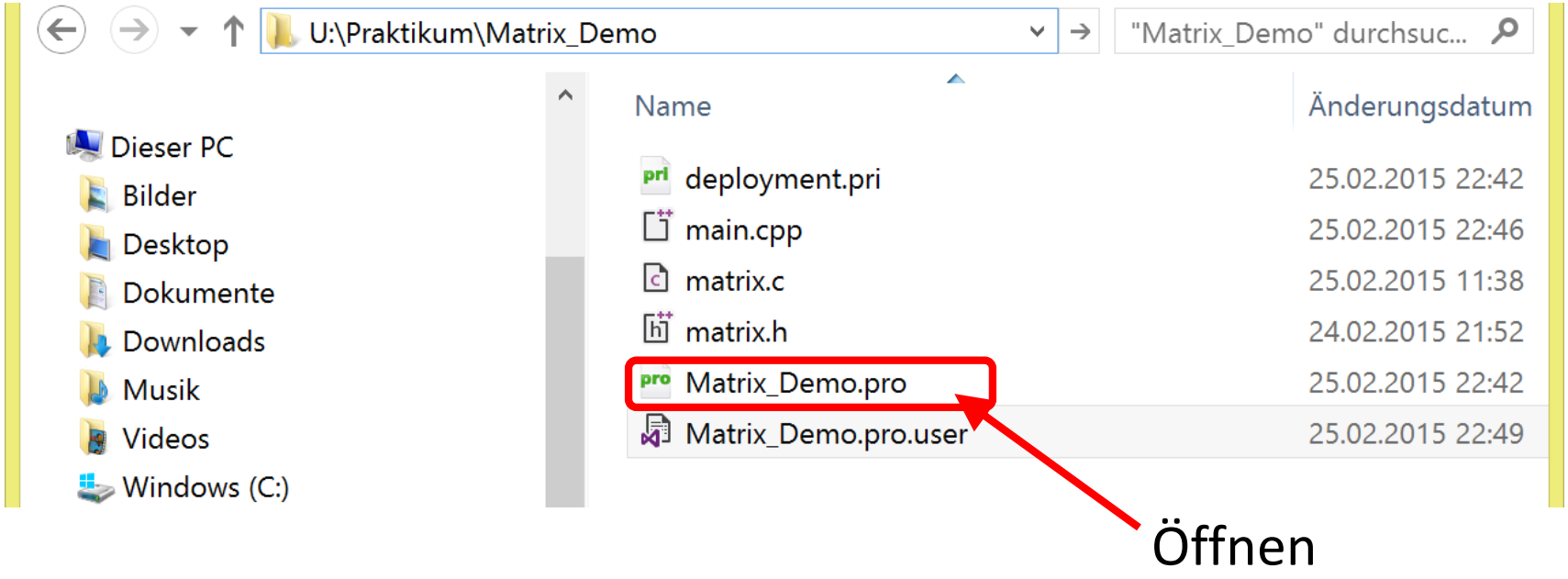


Konkretes Beispiel:

Kopieren Sie das Beispielprojekt „Matrix_Demo.zip“ vom Transfer-laufwerk ins C++-Projektverzeichnis auf Ihrem Rechner (zum Beispiel U:\Praktikum). Entpacken Sie dort die ZIP-Datei.

P1.8. Export/Import von Qt-Creator-Projekten

- Wechseln Sie nun mit dem Windows Explorer ins Projektverzeichnis (zum Beispiel U:\Praktikum\Matrix_Demo).
- Öffnen Sie die Projektdatei (Endung .pro) mit dem Qt Creator.
- **Bitte <u>nicht</u> die .cpp-Datei per Doppelklick öffnen!**



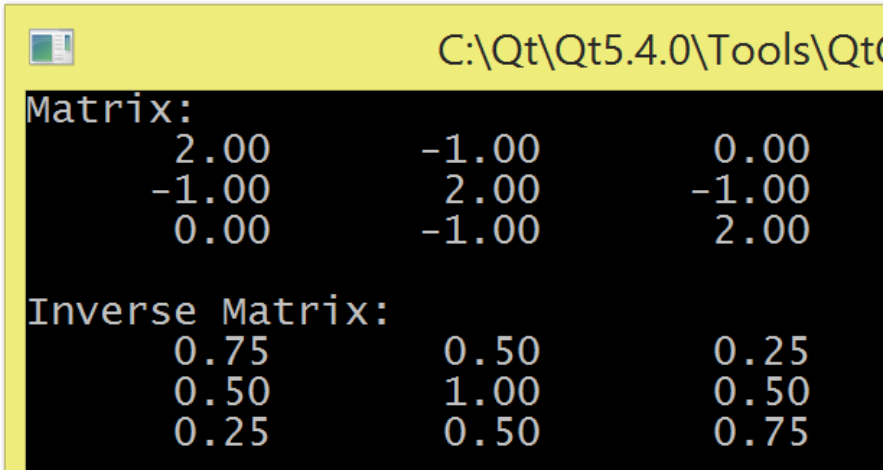
P1.9. Export/Import von Qt-Creator-Projekten

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN MÜNCHEN

Aufgabe:

Übersetzen und starten Sie das Programm **Matrix_Demo**. Versuchen Sie, den Quelltext des Programms zu verstehen und überprüfen Sie das Berechnungsergebnis mit MATLAB.

```
1  #include <stdio.h>
2  #include "matrix.h"
3
4  #define DIM 3
5
6  int main(void)
7  {
8      double mat[DIM][DIM] =
9      {
10         {  2, -1,  0 },
11         { -1,  2, -1 },
12         {  0, -1,  2 }
13     };
14
15     printf("Matrix:\n");
16     m_print(DIM, mat);
17
18     m_invert(DIM, mat);
19
20     printf("Inverse Matrix:\n");
21     m_print(DIM, mat);
```



Zusatzaufgabe:

Berechnen Sie das Produkt $mat \times mat^{-1}$ und geben Sie es auf dem Bildschirm aus.

Rechnerpraktikum zu Kapitel 1

Wiederholung C-Programmierung

Sicherheitsunterweisung
Klausur Ingenieurinformatik 1, SoSe 2016
Export/Import von Qt-Creator-Projekten
Projekte mit mehreren Quelldateien
Beispiel: Elektronik, gedämpfte Schwingung

P1.11. Elektronik, gedämpfte Schwingung

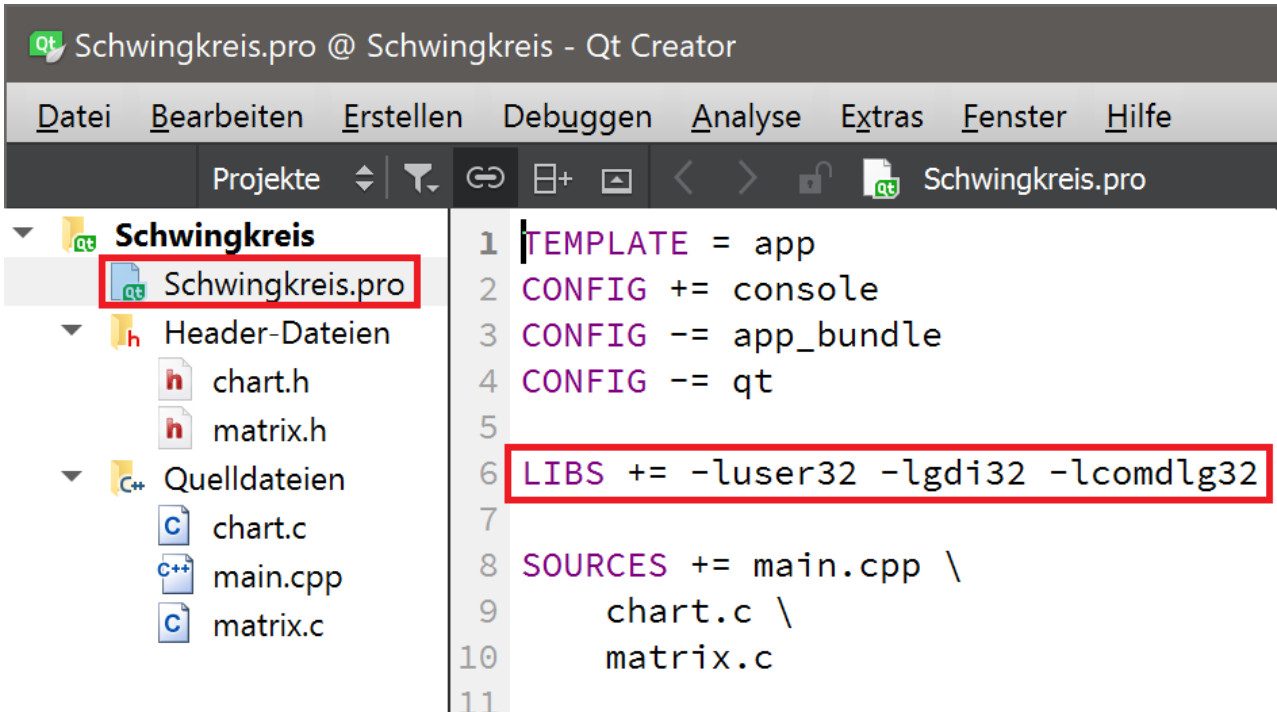
Vorbereitung:

- Erstellen Sie ein neues „reines“ C++-Projekt (ohne Qt).
- Kopieren Sie die Dateien **matrix.c**, **matrix.h**, **chart.c** und **chart.h** ins Projektverzeichnis (wo auch die Datei **main.cpp** liegt).
- Fügen Sie die folgende Zeile zur Projektdatei (.pro-Datei) hinzu:

  `LIBS += -luser32 -lgdi32 -lcomdlg32`

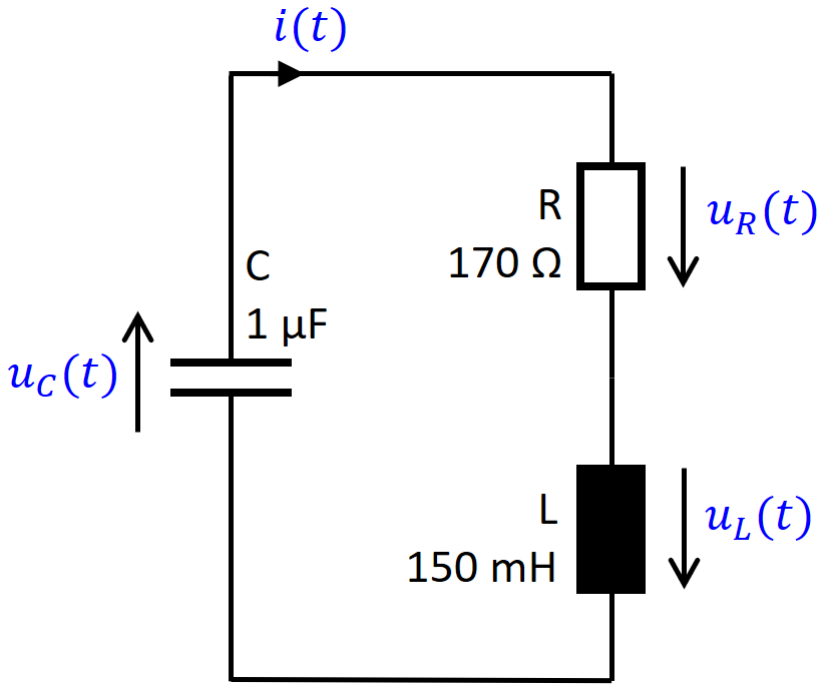
- Fügen Sie die oben genannten Quelltextdateien zum Qt-Projekt hinzu.

  Menüpunkt: „Existierende Datei hinzufügen“



Aufgabe:

Der Stromverlauf $i(t)$ in einem elektrischen Schwingkreis soll simuliert werden. Zum Zeitpunkt $t = 0$ ist (von außen) ein Strom $i(0) = 30\ \text{mA}$ eingeprägt und der Kondensator auf $u_C(0) = -4{,}9\ \text{V}$ geladen. Danach wird der Schwingkreis sich selbst überlassen.



$$u_C + u_R + u_L = 0$$

$$\frac{q}{C} + R \cdot i + L \cdot \frac{\mathrm{d}i}{\mathrm{d}t} = 0 \ \text{ mit } \ i = \frac{\mathrm{d}q}{\mathrm{d}t}$$

$$\frac{q}{C} + R \cdot \dot{q} + L \cdot \ddot{q} = 0$$

$$\boxed{\ddot{q} = -\frac{1}{LC} \cdot q - \frac{R}{L} \cdot \dot{q}}$$

Anfangsbeding.: $q(0) = -4{,}9\ \text{V} \cdot 1\ \mu\text{F}$ und $\dot{q}(0) = i(0) = 30\ \text{mA}$

P1.13. Elektronik, gedämpfte Schwingung

Mit den Zustandsvariablen $y_1 = q$ und $y_2 = \dot{q}$ wird die DGL zweiter Ordnung in ein DGL-System erster Ordnung umgewandelt:

$$\dot{y}_1 = y_2 \qquad y_1(0) = q(0) = -4{,}9\ \text{V} \cdot 1\ \mu\text{F}$$

$$\dot{y}_2 = -\frac{1}{LC} \cdot y_1 - \frac{R}{L} \cdot y_2 \qquad y_2(0) = i(0) = 30\ \text{mA}$$

In vektorieller Schreibweise mit $\vec{y} = [y_1; y_2]$ folgt schließlich:

$$\dot{\vec{y}} = \begin{pmatrix} 0 & 1 \\ -1/(LC) & -R/L \end{pmatrix} \cdot \vec{y} \qquad \vec{y}(0) = \begin{pmatrix} -4{,}9\ \text{V} \cdot 1\ \mu\text{F} \\ 30\ \text{mA} \end{pmatrix}$$

Erstellen Sie ein C-Programm zur numerischen Lösung dieses DGL-Systems mit dem einfachen Euler-Verfahren!

P1.14. Elektronik, gedämpfte Schwingung


```c
/* Schwingkreis */
#include "chart.h"
#include "matrix.h"

#define L 0.15
#define R 170.
#define C 1e-6
#define U 4.9
#define I 0.03

#define DELTA_T 1e-6
#define STEPS 10000

int main(void)
{
    double M[2][2] = { { 0., 1 }, { -1./(L*C), -R/L } };
    double y[2] = { -U*C , I }, dy_dt[2];
```

- 2x2-Matrix M definieren und mit Werten belegen, Vektoren $\vec{y} = [y_1; y_2]$ und $\dot{\vec{y}} = [\dot{y}_1; \dot{y}_2]$ definieren, Integer-Variable i definieren
- Vektor $\vec{y} = [y_1; y_2]$ mit Anfangswerten belegen
- Für alle i von 0 bis (Simulationsschritte – 1)…
  - Ableitungen mittels $\dot{\vec{y}} = \mathrm{M} \cdot \vec{y}$ berechnen
  - $\vec{y} \longleftarrow \vec{y} + \Delta t \cdot \dot{\vec{y}}$ berechnen (Euler-Verfahren)
  - Aktuellen y_2-Wert (= Strom) in Chart darstellen
- Chart auf dem Bildschirm anzeigen

P1.15. Elektronik, gedämpfte Schwingung