

Technische Universität Chemnitz  
Fakultät für Elektrotechnik und Informationstechnik  
Professur für Mess- und Sensortechnik

**Vorbereitung und Protokoll zum Praktikum Elektrische Messtechnik**

**Versuch: LabVIEW Version 1.1**

Versuchstag: 15.01.2019

Laborgruppe: BREEM12

Han, Bing

Weirauch, Julius

Email: [julius.weirauch@s2013.tu-chemnitz.de](mailto:julius.weirauch@s2013.tu-chemnitz.de)

Signum / Bemerkung.....

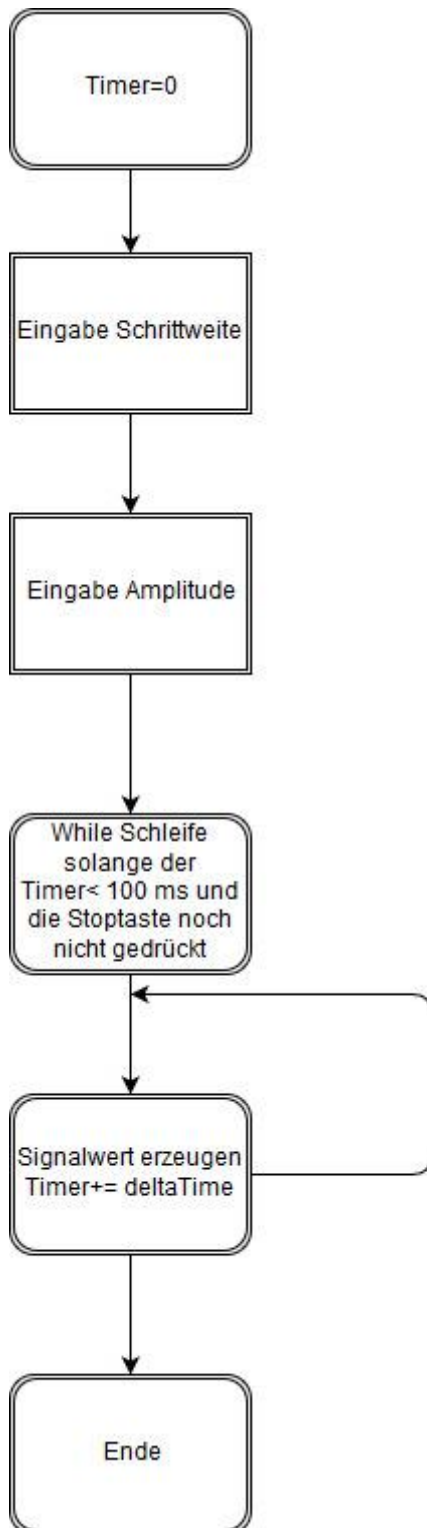
## 2. Grundlagen / Versuchsvorbereitung

### 1. Was ist ein Programmablaufplan/Flussdiagramm?

„Es ist eine grafische Darstellung zur Umsetzung eines Algorithmus in einem Programm und beschreibt die Folge von Operationen zur Lösung einer Aufgabe.“

Quelle: <https://de.wikipedia.org/wiki/Programmablaufplan>

### 2. Erstellen Sie ein Programmablaufdiagramm zur ersten Praktikumsaufgabe!



3. Was besagt das Abtasttheorem nach Shannon?

Um die Eingangsspannung korrekt abzutasten sollte nach dem Nyquist-Kriterium (Abtasttheorem nach Shannon) Abtastfrequenz mindestens doppelt so groß sein wie der Frequenz des Eingangssignals.

4. Erläutern Sie die Unterschiede zwischen differentieller und massebezogener Messung!

differentielle Messung: Die Potentialdifferenz zwischen 2 Knoten der zu messenden Schaltung wird gemessen.

massebezogene Messung: Messschaltung und Messgerät werden an dieselbe Masse angeschlossen. Dadurch wird das Knotenpotential gegenüber der Masse gemessen.

5. Welche Bedeutung haben Gestaltungsart und Farbe der Verbindungslinien im Blockdiagramm? Nennen Sie 5 konkrete Beispiele!



## 6. *Welchen Zweck haben Sequenzstrukturen?*

„Sequenzstrukturen sind dadurch gekennzeichnet, dass die darin enthaltenen Rahmen in sequenzieller Reihenfolge ausgeführt werden. Sie können jedoch auch nur aus einem Unterdiagramm bestehen. Die Ausführungsreihenfolge wird innerhalb der einzelnen Rahmen einer Sequenzstruktur wie im Rest des Blockdiagramms durch die Datenabhängigkeit bestimmt.

Es gibt zwei Arten von Sequenzstrukturen: flache Sequenzstrukturen und gestapelte Sequenzstrukturen. Sequenzstrukturen sollten sparsam verwendet werden, da sie Teile des Blockdiagramms ausblenden. Im Allgemeinen ist es besser, wenn die Ausführungsreihenfolge durch die Datenabhängigkeit und nicht durch Sequenzstrukturen bestimmt wird. Das liegt daran, dass bei jeder Nutzung einer lokalen Sequenzvariablen der in LabVIEW erwünschte Datenfluss von links nach rechts umgangen wird.

**Tipp** Eine bessere Vorgabe des Blockdiagrammablaufs lässt sich z. B. durch Fehler-Cluster erzielen. Wenn keine Durchflussparameter verfügbar sind und Sie eine Sequenzstruktur benötigen, sollten Sie mit der flachen Sequenzstruktur arbeiten.

Anders als bei Case-Strukturen kann es bei Sequenzstrukturen zu jedem Ausgangstunnel immer nur eine Datenquelle geben. Die Datenausgabe kann grundsätzlich von jedem Rahmen erfolgen. Wie bei Case-Strukturen stehen die Daten an den Eingangstunneln für alle Rahmen von flachen und gestapelten Sequenzstrukturen zur Verfügung.

### Flache Sequenzstruktur

Die Rahmen einer flachen Sequenzstruktur werden von links nach rechts ausgeführt, wenn alle erforderlichen Daten am Rahmen anliegen (siehe Abbildung). Die Daten verlassen jeden Rahmen im Anschluss an seine Ausführung. Das heißt, der Ausgangswert eines Rahmens kann als Eingangswert für den nächsten dienen.

Wenn Sie bei einer flachen Sequenzstruktur einen Rahmen hinzufügen oder einen Rahmen entfernen, passt sich die Größe der Struktur automatisch an.

Wenn Sie eine flache in eine gestapelte Sequenzstruktur und anschließend zurück in eine flache Sequenzstruktur umwandeln, werden alle Eingangsanschlüsse an den ersten Rahmen der Sequenz verschoben. Die wiederhergestellte flache Sequenzstruktur funktioniert genauso wie die gestapelte Sequenzstruktur. Nachdem Sie die gestapelte in eine flache Sequenzstruktur umgewandelt haben, können Sie Verbindungen wieder dorthin verschieben, wo sie sich in der ursprünglichen flachen Sequenzstruktur befanden.

Gestapelte Sequenzstruktur:

Bei einer gestapelten Sequenzstruktur (siehe Abbildung) werden die Rahmen übereinander dargestellt, so dass jeweils immer nur ein Rahmen sichtbar ist. Die Rahmen werden der Reihe nach beginnend bei Rahmen 0 abgearbeitet.

Die Daten werden immer erst nach Beendigung des letzten Rahmens ausgegeben. Die gestapelte Sequenzstruktur empfiehlt sich, um Platz im Blockdiagramm zu sparen.

Anders als beim Weitergeben von Daten zwischen Rahmen der flachen Sequenzstruktur benötigen Sie bei der gestapelten Sequenzstruktur lokale Sequenzvariablen zum Weiterleiten von Daten an einen anderen Rahmen.

In der Beschriftung am oberen Rand der gestapelten Sequenzstruktur (siehe Abbildung) wird die Nummer des aktuellen Rahmens und die Gesamtzahl der Rahmen angezeigt.

Hier können Sie zwischen den Rahmen hin- und herspringen und Rahmen neu anordnen. Die Rahmenbeschriftung am oberen Rand der Sequenzstruktur ähnelt der Selektorkennung in einer Case-Struktur. Die Rahmenbeschriftung enthält in der Mitte die Rahmennummer und an den Außenkanten Pfeile zur Rahmenauswahl.

Es können keine Werte in die Beschriftung eingegeben werden. Wenn Rahmen hinzugefügt, entfernt oder in ihrer Anordnung verändert werden, passt sich die Nummerierung automatisch an.“

Quelle: [http://zone.ni.com/reference/de-XX/help/371361R-0113/lvconcepts/sequence\\_structures\\_concepts/](http://zone.ni.com/reference/de-XX/help/371361R-0113/lvconcepts/sequence_structures_concepts/)

7. *Welche zwei grundlegenden Arten von Schleifen kennen Sie? Worin unterscheiden sich diese?*

While- und For-Schleifen:

While-Schleifen werden solange wiederholt bis eine Abbruchsbedingung erfüllt wird.

For-Schleifen werden solange wiederholt, bis ein Zähler einen bestimmten Wert überschreitet.

8. *Wie übergebe ich einen Wert an die nächste Iteration einer Schleife?*

Dafür werden Schieberegister verwendet. Man verbindet den Wert mit dem Rahmen der Schleife und kann dann mit Rechtsklick ein Schieberegister auswählen.

9. *Wie lässt sich der Datenfluss im Blockdiagramm überprüfen?*

Dafür nutzt man die Highlight Execution-Funktion. Dadurch wird der gesamte Datenfluss visualisiert und eventuelle Fehler sind leicht zu finden.

10. *Was sind Cluster?*

Cluster sind Datenstrukturen, die verschiedene Datentypen in ein Objekt zusammenfassen.

11. *Wie schreibe ich Daten in ein Array?*

Entweder ich schreibe die Daten direkt in das Array oder ich nutze eine For-Schleife und schreibe jeweils einen Wert pro Loop ins Array.

12. *Was ist der Unterschied zwischen einem Signalverlaufdiagramm und einem Signalverlaufsgraphen in LabVIEW?*

Beim Signalverlaufdiagramm wird der zeitliche Verlauf eines Wertes dargestellt, kontinuierlich bis die max. Anzahl Zeitpunkten (meist 1024) erreicht ist.

Beim Signalverlaufsgraphen wird ein Array an den Graphen übergeben und dann dargestellt.

13. *Was sind Express – VIs?*

„Ein [Express-VI](#) besteht aus folgenden Komponenten:

- [Konfigurationsdialogfeld-VI](#)—Ermöglicht es dem Benutzer, Einstellungen für die Ausführung des Express-VIs festzulegen.
- [Quell-VI](#)—Enthält den Programmcode des Express-VIs. Das Quell-VI enthält auch eine Verknüpfung zum Konfigurationsdialogfeld-VI.

Wenn Sie ein Express-VI in ein Blockdiagramm einfügen, wird ein Konfigurationsdialogfeld angezeigt. Das Konfigurationsdialogfeld-VI definiert das Erscheinungsbild und die Funktionsweise des Dialogfelds. Nach der Konfiguration des Express-VIs und Schließen des Dialogfelds, wird das VI als [erweiterbarer Knoten](#) im Blockdiagramm angezeigt. Sie können dann die Ein- und Ausgänge des Knotens festlegen. Es ist aber nicht möglich, den Knoten durch Doppelklick zu öffnen und Frontpanel sowie Blockdiagramm anzuzeigen, wie das bei SubVIs der Fall ist. Zur Anzeige des Programmcodes ist eine [Umwandlung des Express-VIs in ein SubVI](#) erforderlich. Der Programmcode von Express-VIs wird durch ein Quell-VI

definiert. Die Umwandlung von Express-VIs in SubVIs ist endgültig und der Zugriff auf das Konfigurationsdialogfeld ist nicht mehr möglich.

Um ein Express-VI zu erstellen, sind sowohl Konfigurationsdialogfeld-VI als auch Quell-VI notwendig.

Folgende Entwicklungsschritte müssen für die Erstellung und Verteilung von Express-VIs ausgeführt werden:

- [Entwicklung des Quell-VIs](#) für das Express-VI.
- [Entwicklung des Konfigurationsdialogfeld-VIs](#) einschließlich Benutzeroberfläche.
- [Zuweisen von Hilfedateien](#) zum Express-VI.
- Anordnung des Express-VIs in der LabVIEW-[Verzeichnisstruktur](#).
- Hinzufügen des Express-VIs zu einer [neuen oder bestehenden Palette](#).

Wenn Sie ein Express-VI dem Verzeichnis labview\user.lib\\_express hinzufügen, wird es auf der Palette [Eigene Bibliotheken \(Express\)](#) angezeigt. Für LLBs oder Verzeichnisse in labview\user.lib\\_express, die nicht mit einem Unterstrich beginnen, werden **Palettenmenü**-Unterpaletten angezeigt. Daher werden sowohl das Express-VI als auch das SubVI des Quell-VIs auf der Palette angezeigt.“

Quelle: [http://zone.ni.com/reference/de-XX/help/371361P-0113/lvconcepts/creating\\_express\\_vis/](http://zone.ni.com/reference/de-XX/help/371361P-0113/lvconcepts/creating_express_vis/)

#### *14. Welche Vorteile bringt das Anlegen von Sub – VIs?*

Das Programm wird übersichtlicher, Redundanzen wird vorgebeugt, einzelne mehrfach benötigte Programmteile können schnell wiederverwendet werden.

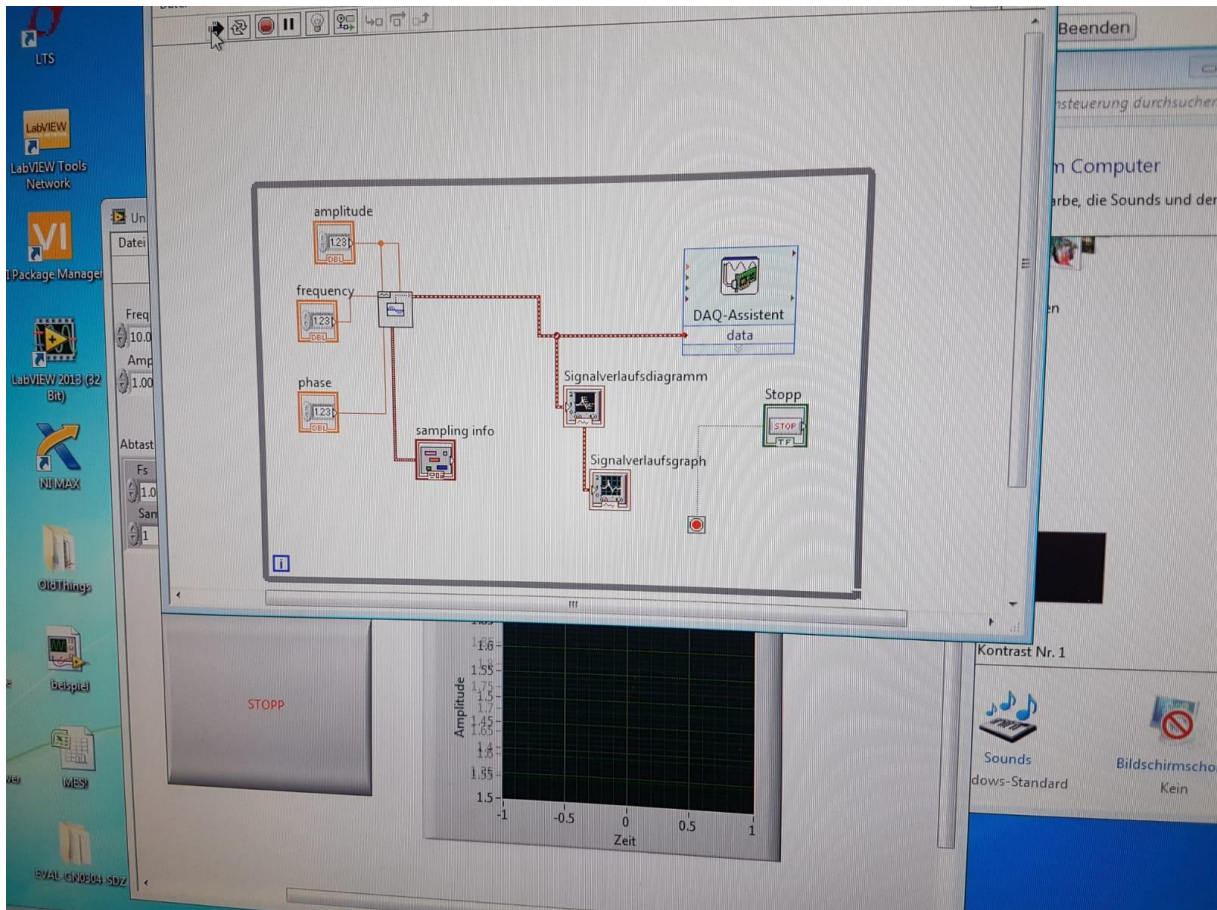


Bild 1: LabVIEW-Interface

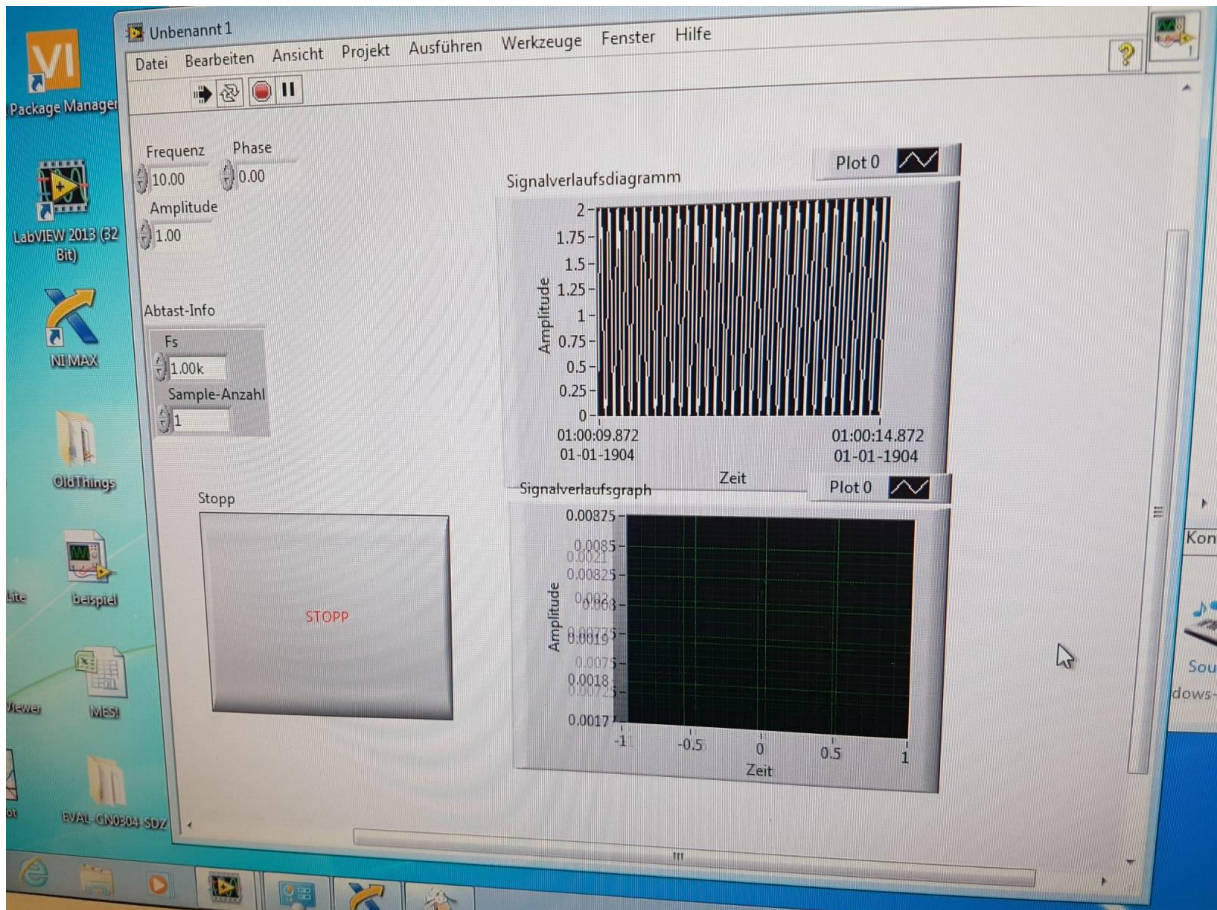


Bild 2: LabVIEW-Nutzer-Interface

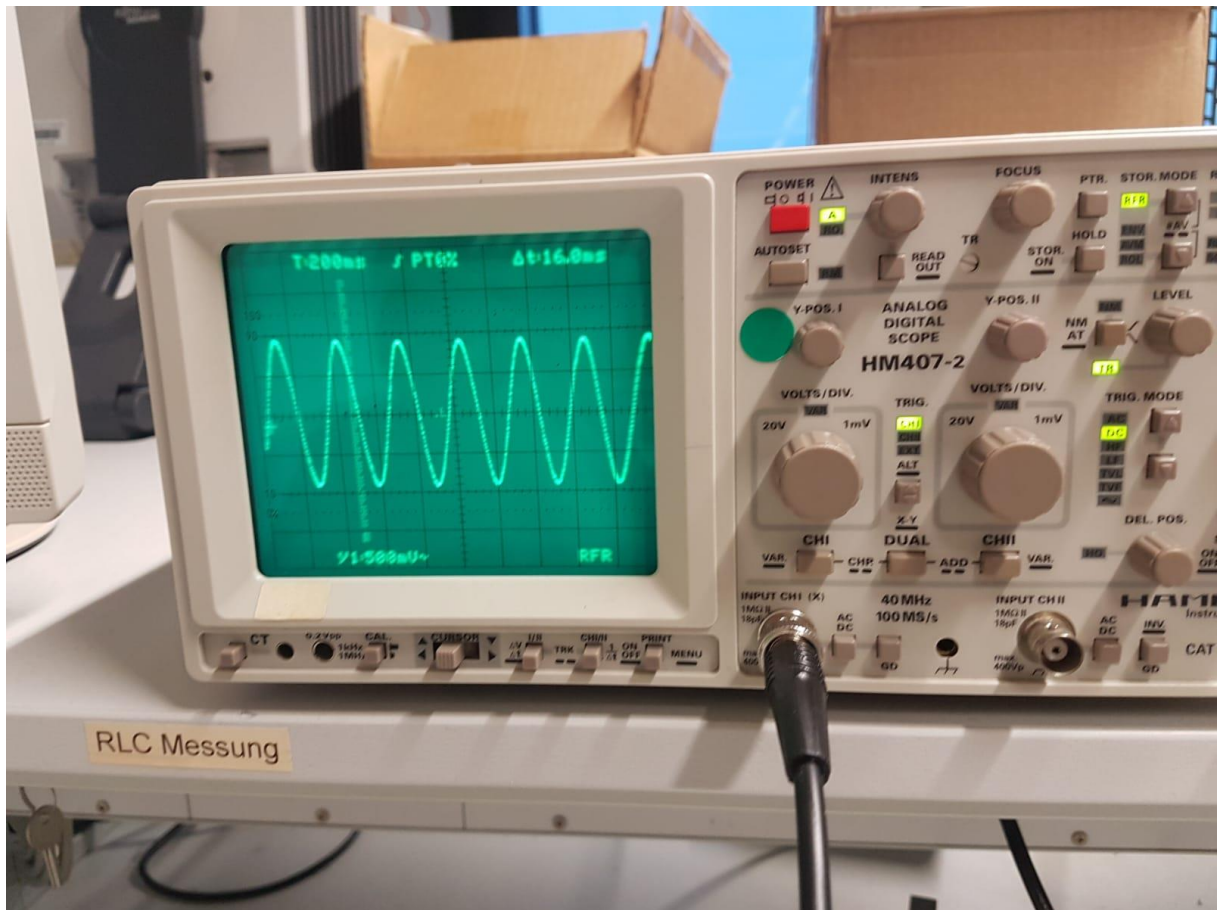


Bild 3: Oszi-Bild

**Weitere Quellen:**

[https://www.tu-chemnitz.de/mb/sgt/Download/wiss\\_Arbeit/Richtlinien\\_wiss\\_Arbeiten.pdf](https://www.tu-chemnitz.de/mb/sgt/Download/wiss_Arbeit/Richtlinien_wiss_Arbeiten.pdf)