

Fachhochschule Köln  
University of Applied Sciences Cologne

07

Fakultät für Informations-, Medien- und  
Elektrotechnik, Studiengang Elektrotechnik

Institut für elektrische Energietechnik

# Bachelorarbeit

## **Thema:**

Software für eine automatische Messdatenerfassung eines Synchrongenerator  
Messstandes

## **Mustu, Sertac**

Fliederweg 2

51429 Bergisch Gladbach

Matr. Nr. 11075224

Referent:

Prof. Dr. Eberhard Waffenschmidt

Korreferent:

Prof. Dr.-Ing. Christof Humpert

Abgabedatum:

15. Oktober 2013



## **i. Eidesstattliche Erklärung**

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen Quellen wörtlich oder sinngemäß entnommen wurden, sind kenntlich gemacht. Die Arbeit war in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung.

---

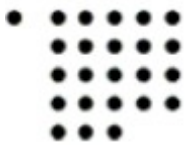
Ort, Datum

Unterschrift: Mustu, Sertac



## ii. Danksagung

Hiermit möchte ich mich bei meinem Referenten Herr Prof. Dr. Eberhard Waffenschmidt bedanken, dafür das er mir diese Arbeit ermöglichen konnte, mir immer Hilfestellungen gegeben hat und den „Diplomarbeit“-Raum zur Verfügung stellen konnte. Weiterhin bedanke ich mich bei meinem guten Freund und Kommilitonen Herr Taregh Ain Haddadi der für die Hardware der Arbeit zuständig war , er war immer als Ansprechpartner an meiner Seite, hat mir bei meiner Arbeit sehr geholfen und mich ständig motivieren. Desweiteren bedanke ich mich bei Prof. Dr.-Ing. Christof Humpert, der die Zustimmung für die Funktion als Korreferenten abgeben hat und sich um meine Anmeldung gekümmert hat. Ein weiterer Dank geht an den wissenschaftlichen Mitarbeiter Herrn Jochen Reichert , der mir die Materialien zur Verfügung stellen konnte. Auch bedanke ich mich bei meiner Familie die mich dabei unterstützt haben.



### iii. Inhaltsverzeichnis

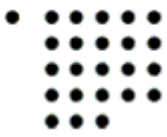
<u>i.Eidesstattliche Erklärung.....</u>	<u>i</u>
<u>ii.Danksagung.....</u>	<u>ii</u>
<u>iii. Inhaltsverzeichnis.....</u>	<u>iii</u>
<u>1 Einleitung.....</u>	<u>1</u>
<u>2 Aufgabenstellung und Ziel.....</u>	<u>2</u>
<u>3 Materialliste für das Projekt.....</u>	<u>3</u>
<u>4 Theorie.....</u>	<u>4</u>
<u>4.1 Motivation.....</u>	<u>4</u>
<u>4.2 Erläuterung der Schaltung.....</u>	<u>4</u>
<u>4.2.1 Aufbau.....</u>	<u>4</u>
<u>4.2.2 Die Messinstrumente.....</u>	<u>5</u>
<u>4.2.2.1 Fazit.....</u>	<u>5</u>
<u>4.2.3 Die Aronschaltung.....</u>	<u>6</u>
<u>4.2.3.1 Fazit.....</u>	<u>7</u>
<u>4.2.4 Der Leonardumformer.....</u>	<u>8</u>
<u>4.2.4.1 Der Gleichstrommotor in Bezug auf den Leonardumformer.....</u>	<u>8</u>
<u>4.2.4.2 Fazit.....</u>	<u>9</u>
<u>4.2.5 Die Dunkelschaltung.....</u>	<u>9</u>
<u>4.2.5.1 Fazit.....</u>	<u>10</u>
<u>4.3 Begründung der Materialliste.....</u>	<u>10</u>
<u>4.3.1 Nationalinstruments LabVIEW.....</u>	<u>10</u>
<u>4.3.2 Funktionsgenerator(Testgenerator).....</u>	<u>11</u>
<u>4.3.3 Echtzeitbetriebssystem.....</u>	<u>11</u>
<u>4.3.3.1 Fazit.....</u>	<u>12</u>
<u>4.3.4 BNC-Buchse,PCI-Karte und Kabel.....</u>	<u>12</u>
<u>4.4 Nyquist-Theorem.....</u>	<u>13</u>
<u>4.4.1 Theorie zur Auflösung der Oszilloskopbilder.....</u>	<u>13</u>
<u>4.5 Überlegung zur Berechnung der Leistungen(S,P,Q) in Labview.....</u>	<u>15</u>
<u>4.5.1 Wirkleistung P.....</u>	<u>16</u>
<u>4.5.2 Scheinleistung S.....</u>	<u>17</u>
<u>4.5.3 Blindleistung Q.....</u>	<u>18</u>
<u>4.5.4 Gesamtwirkleistung(Blindleistung,Wirkleistung und Scheinleistung).....</u>	<u>18</u>
<u>4.6 Überlegung zur Berechnung der Effektivwerte (Strom und Spannung) in Labview.....</u>	<u>18</u>
<u>4.6.1 Effektivwert Spannung.....</u>	<u>19</u>
<u>4.6.2 Effektivwert Strom.....</u>	<u>19</u>
<u>4.7 Messprogramm.....</u>	<u>19</u>
<u>4.7.1 Überarbeitete Messschaltung.....</u>	<u>19</u>
<u>4.7.2 Vorwort.....</u>	<u>20</u>
<u>4.7.3 Zu erfassende Messsignale und Kanaluweisung.....</u>	<u>21</u>
<u>4.7.3.1 Messsignale.....</u>	<u>21</u>
<u>4.7.3.2 Dynamische Spezifikation für alle Signale.....</u>	<u>22</u>
<u>4.7.4 Von Messpunkt zu berechnen.....</u>	<u>23</u>
<u>4.7.4.1 Für jeden Messpunkt zu berechnen:.....</u>	<u>23</u>
<u>4.7.4.2 Für jedes Messintervall zu berechnen.....</u>	<u>24</u>
<u>4.7.5 Speichern der Daten.....</u>	<u>27</u>
<u>4.7.5.1 Name und Format.....</u>	<u>27</u>



4.7.5.1.1	Metadaten	27
4.7.5.1.2	Dateninhalt	27
4.7.5.1.3	Muster	28
4.7.5.2	Oszilloskopkurvenaufzeichnung	28
4.7.5.2.1	Automatisierung Oszilloskopkurvenaufzeichnung	28
4.7.5.3	Kenndaten aus Oszilloskopkurvenaufzeichnung	28
4.7.5.4	Namensauflösung der Datei	29
4.7.6	Benutzeroberfläche	29
4.7.6.1	Fenster „Bedienung“	29
4.7.6.2	Fenster „Einstellungen“	30
4.7.6.3	Fenster „Kenndaten“	30
4.7.6.3.1	Zeigerinstrumente	30
4.7.6.3.2	Dunkelschaltung	30
4.7.6.4	Fenster „Oszillogramme“	31
4.7.6.5	Fenster „Schreiberkurven“	31
4.7.6.6	Formatierung von Diagrammen	31
5	Programmierung in Labview	32
5.1	Auflistung benutzter Express-VI's und VI Funktionen	32
5.2	Quellcode	36
5.2.1	Sub-VI	36
5.2.1.1	Sub-VI zur Bestimmung der Leistungen und Anzahl der Perioden	36
5.2.1.1.1	Prüfen der Sub-VI (Bestimmung der Leistungen und Anzahl der Perioden)	40
5.2.1.2	Sub-VI zur Bestimmung des Effektivwertes	41
5.2.1.2.1	Prüfen der Sub-VI Effektivwert	42
5.2.1.3	Sub-VI Minuswert(Instantan)	42
5.2.2	Erläuterung zum Quellcode der Benutzeroberfläche	44
5.2.2.1	Vorwort	44
5.2.2.2	Inputs und Outputs der Benutzeroberfläche	44
5.2.2.3	Das Messprogramm(Hauptprogramm)	46
5.2.2.3.1	Die Messsignale	46
5.2.2.3.2	Messpunkt- und Messintervallberechnung	47
5.2.2.3.2.1	Leistungen	47
5.2.2.3.2.2	Effektivwerte	48
5.2.2.3.2.3	Phasendifferenz berechnen	48
5.2.2.3.2.4	Differenzspannung	49
5.2.2.4	Speichern der Daten	49
5.2.2.4.1	Metadaten(Oszilloskop-Bilder)	50
5.2.2.4.2	Metadaten (Kenndaten\Schreiberkurve)	52
5.2.2.4.3	Dateninhalt (Oszilloskop-Bilder)	52
5.2.2.4.4	Dateninhalt (Schreiberkuve/Kenndaten)	54
5.2.2.4.5	Dateinamen	56
5.2.2.4.6	Vorwort Fenster	58
5.2.2.4.7	Fenster Bedienung	60
5.2.2.4.7.1	Laufendes Programm	60
5.2.2.4.7.2	Kenndatenspeicherung	60
5.2.2.4.7.3	Speichervorgang Oszilloskop-Aufzeichnung	61
5.2.2.4.7.4	Taster „Rücksetzen der Kennwerte“	62
5.2.2.4.7.5	Start und Stopp der Oszilloskop-Aufzeichnung	62

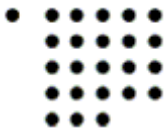


5.2.2.4.7.6 Start und Stopp der Kenndaten-Aufzeichnung .....	63
5.2.2.4.8 Fenster „Einstellungen“ .....	63
5.2.2.4.8.1 Angaben Messintervall .....	63
5.2.2.4.8.2 Angaben Datenerfassung .....	64
5.2.2.4.9 Fenster Kenndaten .....	65
5.2.2.4.10 Fenster „Oszillosgramme“ .....	66
5.2.2.4.11 Fenster „Schreiberkurven“ .....	67
5.2.2.5 Ergebnisse und Fazit .....	68
5.3 Bedienungsanleitung für Programmbenutzer .....	71
6 Gesamtfazit .....	73
7 Anhang .....	75
7.1 A1 Spezifikation .....	75
7.2 A2 Aufbau und Installation (Materialliste) .....	85
7.3 A3 Registerkartenfenster .....	90
7.4 A4 Frequenzmessungstest .....	98
7.4.1 Rauschen .....	98
7.4.2 Mittelwert und Standardabweichung .....	99
7.5 A5 Erläuterung zum Quellcode „Test für die Frequenzmessung“ .....	102
7.5.1.1 Sub-VI Mittelwert .....	102
7.5.1.1.1 Prüfen der Sub-VI AV .....	103
7.5.1.2 Sub-VI Standardabweichung .....	104
7.5.2 Die Folge: „Test für die Frequenzmessung“ .....	110
I. Abbildungsverzeichnis .....	112
II. Tabellenverzeichnis .....	115
III. Literaturverzeichnis .....	117



## 1 Einleitung

Bestehend aus einem fremderregten Synchrongenerator, wird ein Maschinensatz mit einem Messstand von einem fremderregten Gleichstrommotor angetrieben. Die variable Betriebsspannung des fremderregten Synchrongenerators wird über einen fremderregten Gleichstromgenerator geregelt. Dieses ist gekoppelt mit einem Asynchronmotor welches zum Antrieb zuständig ist und an einem dreiphasigen Wechselspannungsnetz anliegt, das ganze nennt sich dann Leonardsatz ohne den Synchrongenerator. Die Synchronmaschine kann im Inselbetrieb oder nach Synchronisation am Dreiphasen-Stromnetz betrieben werden. Die Messdaten werden hardwaretechnisch über spezielle Sensoren und galvanisch getrennten Bauteile erfasst, für die ein weiterer Bachelorarbeiter zuständig ist. Solche und ähnliche Einrichtungen befinden sich heute noch in Laboren und anderen wissenschaftlichen Einrichtungen. Leider besitzen einige Einrichtungen nicht die wissenschaftlichen Messinstrumente, die technisch auf dem aktuellen Stand sind. Auf Grund dieses Problems sind Experimente, wie man zum Beispiel ein Inselnetz mit einem größeren Netz synchronisiert um diese zusammenzuschalten und Problemstellungen fast gar nicht Vorstellbar. Die Software für eine automatische Messdatenerfassung eines Synchrongenerator Messstandes erleichtert nicht nur Experimente oder Problemstellungen, es ist auch benutzerfreundlich und erspart auch viele Messinstrumente. Der Experimentierende hat somit die Möglichkeit, eine lukrativere Messung durchzuführen und dabei die fremderregte Synchrongeneratorspannung/-frequenz zu variieren.

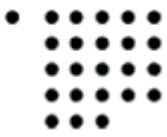


## 2 Aufgabenstellung und Ziel

Für einen Maschinensatz mit einem Messstand soll eine Software erstellt werden, welches Messdaten erfasst:

Datenerfassung für alle 3-Phasenströme, alle 3-Phasenspannungen, Erregerspannung und -strom von Gleichstrommotor, Gleichstromgenerator, Synchrongenerator und Gleichstrom und -spannung der Gleichstrommaschine. Ziel ist es die Messdaten zu erfassen, Fitfunktionen (wie Effektivwert) für diese zu erstellen, aus diesen die Leistungen zu berechnen, diese in Schreiberkurven, Oszilloskopbildern und Zeigerinstrumenten darzustellen und über eine Automatisierung die Abspeicherung mit einer entsprechenden Kopfzeile und Dateinamen abzuspeichern. Dabei kann man bei der Automatisierung die Kennwerte zusätzlich noch zurücksetzen und die Kurven- und Oszilloskopaufzeichnung stoppen.





### 3 Materialliste für das Projekt




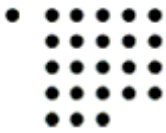
Bezeichnung der Geräte	Gerätetyp	Anzahl	Technische Daten
NI PCI-6259 16 bit, 1 MS/s (Mehrkanal), 1,25 MS/s (Einkanal), 32 Analogeingänge  [1]	A/D - Karte	1	<ul style="list-style-type: none"> <li>4 Analogausgänge mit 16-bit-Auflösung (2,8 MS/s), 48 Digital-I/O-Kanäle, Counter/Timer mit 32-bit-Auflösung</li> <li>Mehr als 70 Möglichkeiten der Signalkonditionierung, NIST-konformer Kalibrierschein</li> <li>Kalibriertechnologie NI-MCal für erhöhte Messgenauigkeit</li> <li>Synchronisierte Digital-I/O-Kanäle (32 getaktete Kanäle, 10 MHz), analoge und digitale Trigger</li> <li>Optimierte Messgenauigkeit, -auflösung und -empfindlichkeit mit den hochpräzisen Modulen der M-Serie</li> <li>Enthält Treibersoftware NI-DAQmx und die interaktive Datenprotokollierungssoftware NI LabVIEW SignalExpress LE</li> </ul>
National Instruments LabVIEW 2012	CD(mit der Software)	1	
NI BNC-2111 Geschirmter Anschlussblock für Geräte der X- und M-Serie  [2]	BNC-Anschlüsse für I/O-Anbindung	2(AC/DC)	<ul style="list-style-type: none"> <li>Anschlussblock für Single-ended-Analogueingänge, Analogausgänge und Digital-I/O</li> <li>Geschirmtes Gehäuse zur störungsfreien Messung</li> </ul>
NI SHC68-68-EPM  Geschirmtes Kabel mit erweiterter Funktionalität für Geräte der X- und M-Serie[3]	Verbindet 68-Pin-Geräte der X- und M-Serie mit 68-Pin-Zubehör (Kabel)	2	Verfügbare Kabellängen: 0,5, 1, 2, 5 und 10 m <ul style="list-style-type: none"> <li>Einzel geschirmte, analoge, verdrehte Kabelpaare für vermindertes Übersprechen zwischen Hochgeschwindigkeitsmodulen</li> <li>RoHS-konform</li> <li>Geschirmtes SHC68-68-Kabel mit erweiterter Funktionalität</li> </ul>
PROGRAMMABLE FUNCTION GENERATOR HM8150[4]	Funktionsgenerator	1	<ul style="list-style-type: none"> <li>- Frequenzbereich 10 MHz ... 12,5 MHz</li> <li>- Ausgangsspannung 10 mV ... 10 Vpp (an 50 Ω)</li> <li>- Signalformen: Sinus, Rechteck, Dreieck, Puls, Sägezahn,</li> <li>- Aufstieg und Fall Zeit &lt;10 ns</li> <li>- Pulsdauer Einstellung: 100 ns ... 80 s</li> <li>- Arbiträrgenerator 40 MSa / s</li> <li>- Burst, Gating, externe Triggerung, Fegen -Kostenlos</li> <li>Software für Remote Control und für die Erzeugung von Arbiträrsignale</li> <li>- Externe Amplitudenmodulation (Bandbreite 20 kHz)</li> <li>- Intuitive Bedienung mit einem Knopfdruck, schnelleres Wechseln der Signale</li> </ul>
Prozessor: Intel(R) Core(TM)2 Duo CPU E4800 @ 3.00GHz 3.00GHz RAM: 2,00GB Systemtyp: 64-Bit-Betriebssystem	Rechner mit Betriebssystem	1	kompatibel mit der LabVIEW-Software

Tabelle 3.1: Materialliste



## 4 Theorie

### 4.1 Motivation

Der Synchrongenerator soll eine Art Inselnetz zur Verfügung stellen dessen Frequenz und Spannung variabel ist. Die Messeinrichtung ermöglicht es unterschiedliche Experimente zu erstellen und diese zu untersuchen. Ein Fall wäre zum Beispiel zu untersuchen bzw. zu zeigen, wie man ein Inselnetz mit einem größeren Netz synchronisiert um diese zusammenzuschalten (dabei müssten beide Netze auf einer Phase sein). Außerdem kann man beobachten was passiert, wenn zum Beispiel bei einem Blackout (Stromausfall) in einem Kraftwerk die Stromnetze hochgefahren werden müssen. Weil dann die Kraftwerke zunächst einzeln für sich starten und dann Stück für Stück sich miteinander synchronisieren müssen. Ein weiterer Fall ist zum Beispiel eine dezentrale Einspeise einer Photovoltaik Anlage in einem Inselnetz bzw. Netz als auch die Reaktion eines Wechselrichters zu untersuchen. Auch bei Frequenz- und Netzschwankungen ist eine Untersuchung realisierbar. Dabei kann man den Messstand verwenden und die Frequenz erhöhen und vermindern. Hierfür werden dann Messdaten erstellt, die über die Software z.B. pro Sekunde erfasst und in einer Datei abgespeichert werden. Diese Simulationen sind mit der Messdatenerfassung realisierbar.

### 4.2 Erläuterung der Schaltung

#### 4.2.1 Aufbau

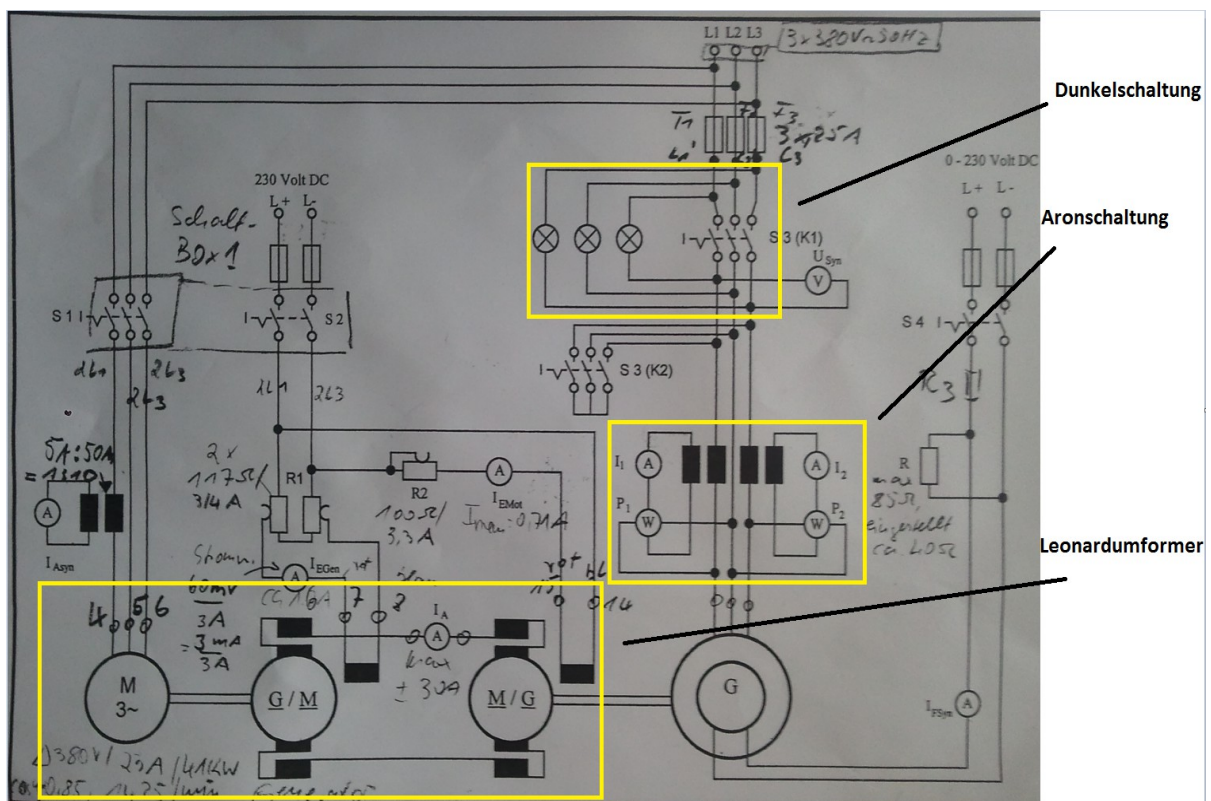


Abbildung 4.1: Synchrongenerator mit Leonardsatz

Die in der Einleitung erläuterte Messeinrichtung mit dem Synchrongenerator (Abb 4.1 unten rechts) und Leonardsatz (Abb 4.1 unten links) befindet sich in der Fh-Köln HW1-36/37 und hat folgende Schaltung vermerkt:

Wie man in Abb 4.1 sieht, wurde hier schon markiert welche weiteren Spannungen und Ströme noch zusätzlich ermittelt werden muss. Am Synchrongenerator befinden sich zwei Wattmeter, welche die Aron-Schaltung darstellt. Wenn die Lampen aus sind zeigen sie, dass die Spannung des Synchrongenerators in Phase mit der Netzspannung ist. Weitere Spannungsmessung wie die Erregerspannung der Synchronmaschine, Gleichstromgenerator Gleichstrommotor und die Gleichstrom Zwischenkreisspannung sind hinzuzufügen. S1, S2, S3 und S4 sind Lasttrennschalter, die als Schutz einrichtung gesetzt werden. Für die Strommessungen ist eine Schaltung zu entwickeln, die den Stromwert als Spannungswert entnimmt. Für die Aufgabe ist der für die Hardware-Aufgabe verpflichtende Bachelorarbeiter zuständig. Der auch für die Kanalvergabe zuständig ist.

## 4.2.2 Die Messinstrumente



Abbildung 4.2: Zeigerinstrumente

Die Messinstrumente (in Abb. 4.2) bestehen hauptsächlich aus Zeigerinstrumente. Diese stehen schon seit einer Weile im Labor, haben eine bestimmte Lebensdauer und funktionieren auch nicht mehr zu 100%. Wie man sieht wurden schon auf einigen Zeigerinstrumente die Skalierung mit einem Marker bzw. Filzstift markiert. Da es wahrscheinlich einen Mangel an Zeigerinstrumenten mit der passenden Skalierung existierte oder die richtigen Stromwandler hierfür nicht bereit gestellt wurden. LabVIEW kann verschiedene Formen von Instrumenten darstellen, außerdem kann die Skalierung digital eingestellt werden.

### 4.2.2.1 Fazit

LabVIEW hat nicht nur den Vorteil Messinstrumente zu sparen, es kann sogar die instantanen Werte auf Oszilloskopbilder und Schreiberkurven für Messintervalle aufzeichnen. (Dazu aber mehr in Kapitel 4.3.1)

### 4.2.3 Die Aronschtaltung

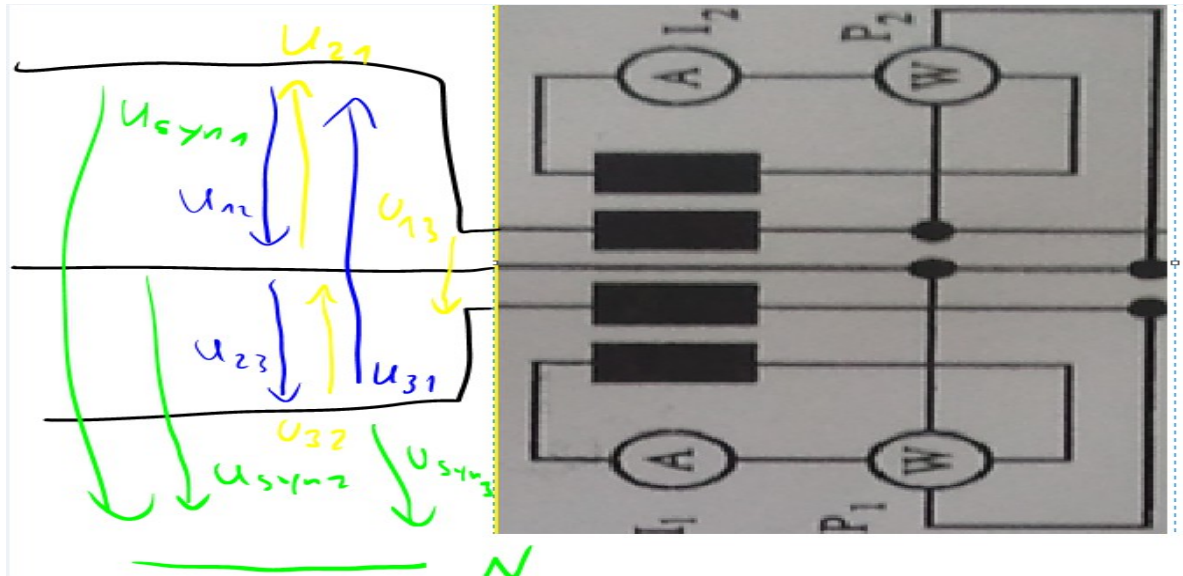


Abbildung 4.3: Aaronschtaltung

Wie schon in Kapitel 4.2.1 erklärt ist die Aronschtaltung an dem Synchrongenerator verschaltet. Die Aronschtaltung besteht aus zwei Wattmetern. Die Spannung wird durch das Anlegen der Wattmeter für die Außenleiterspannung, der Strom über Stromwandler bestimmt und berechnet. Die Summe der Wattmetern ergibt die Gesamtleistung des Synchrongenerators. Zeigt ein Zeigerinstrument als Wattmeter ein negativen Wert, ist die Umpolung vertauscht. Die Berechnung, mit Hilfe des Augenblickwertes, zeigt ganz klar den Beweis für die Aronschtaltung:

$$P_{\text{synges}}(t) = p_{\text{syn1}}(t) + p_{\text{syn2}}(t) + p_{\text{syn3}}(t) = u(t)_{\text{syn1}} \cdot i(t)_{\text{syn1}} + u(t)_{\text{syn2}} \cdot i(t)_{\text{syn2}} + u(t)_{\text{syn3}} \cdot i(t)_{\text{syn3}} \quad (4.2.3.1)$$

mit:

$P_{\text{synges}}(t)$  – Augenblickswert Gesamtwirkleistung

$P_{\text{syn1}}(t), P_{\text{syn2}}(t), P_{\text{syn3}}(t)$  – Augenblickswert 1,2,3 Phasenwirkleistung

$I_{\text{syn1}}(t), I_{\text{syn2}}(t), I_{\text{syn3}}(t)$  – Augenblickswert 1,2,3 Phasenstrom

$U_{\text{syn1}}(t), U_{\text{syn2}}(t), U_{\text{syn3}}(t)$  – Augenblickswert 1,2,3 Phasenspannung

da „symmetrisch“ gilt:

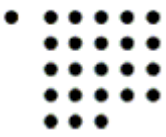
$$i(t)_{\text{syn1}} + i(t)_{\text{syn2}} + i(t)_{\text{syn3}} = 0$$

$$\Rightarrow p(t) = u(t)_{\text{syn1}} \cdot i(t)_{\text{syn1}} - u(t)_{\text{syn2}} \cdot (i(t)_{\text{syn1}} + i(t)_{\text{syn3}}) + u(t)_{\text{syn3}} \cdot i(t)_{\text{syn3}}$$

$$\Leftrightarrow p(t) = i(t)_{\text{syn1}} (u(t)_{\text{syn1}} - u(t)_{\text{syn2}}) + i(t)_{\text{syn3}} (u(t)_{\text{syn3}} - u(t)_{\text{syn2}}) = i(t)_{\text{syn1}} \cdot u(t)_{\text{syn12}} + i(t)_{\text{syn3}} \cdot u(t)_{\text{syn32}}$$

Integrieren wir nun  $p(t)$  über eine Periode und mitteln es über Diese erhalten wir die Wirkleistung:

$\Rightarrow$



$$P = \bar{p}(t) = \frac{1}{T} \int_{t_0}^{T+t_0} (i(t)_{syn1} \cdot u(t)_{syn12} + i(t)_{syn2} \cdot u(t)_{syn32}) dt = i(t)_{syn1} \cdot \bar{u}(t)_{syn12} + i(t)_{syn2} \cdot \bar{u}(t)_{syn32} \quad (4.2.3.2)$$

mit

$P$  – Wirkleistung

$T$  – Periodendauer

$u(t)_{syn23}, u(t)_{syn13}$  – Zeitabhängige außenleiter Spannung (zwischen 2 und 3, 1 und 3)

$t_0$  – Anfangszeitpunkt der Periode

Für die Berechnung wird elektronisch verfahren. Mit dieser Rechnung sehen wir, dass die Mittelung und das Integral über eine Periode, mit Multiplikation zwischen Phasenstrom und Außenleiterspannung der betroffenen Aronschaltung in Abb. 4.1, zu einer Gesamtwirkleistung führt.

#### 4.2.3.1 Fazit

Sinnvoll wäre es hier die drei Ströme und Spannungen der Phasen zu messen, um mit diesen Daten die Berechnung für die Leistungen, über das Programm, durchführen zu können. Die Aronschaltung ist nicht nur komplex aufgebaut, sie behindert auch mit dem Strommessgerät die Leistungsmessgeräte. Ein weiterer Vorteil bei der Entnahme der Aronschaltung ist, dass man 2 Wattmeter spart. Bei der Gesamtwirkleistung und weiteren Leistungen würde dann das Programm die Berechnung folgenderweise durchführen:

$$P_{synges} = \frac{1}{T} \int_{t_0}^{T+t_0} p_{syn1}(t) dt + \frac{1}{T} \int_{t_0}^{T+t_0} p_{syn2}(t) dt + \frac{1}{T} \int_{t_0}^{T+t_0} p_{syn3}(t) dt = \frac{1}{T} \int_{t_0}^{T+t_0} u(t)_{syn1} \cdot i(t)_{syn1} dt \quad (4.2.3.3)$$

$$+ \frac{1}{T} \int_{t_0}^{T+t_0} u(t)_{syn2} \cdot i(t)_{syn2} dt + \frac{1}{T} \int_{t_0}^{T+t_0} u(t)_{syn3} \cdot i(t)_{syn3} dt$$

#### 4.2.4 Der Leonardumformer

Der Leonardumformer ist eine Alternative zum Vier-Quadranten-Betrieb. Wie in Abb. 4.1 besteht er aus einem Drehstrommotor wie zu meist Asynchronmotor(Gleichstrom- und Synchronmotor auch möglich ) der mit einem fremderregten Gleichstromgenerator über eine Welle angeschossen ist. Die Ankeranschlüsse des Gleichstromgenerators sind mit einem fremderregten Gleichstrommotor verbunden, der diesen auch einspeist.

##### 4.2.4.1 Der Gleichstrommotor in Bezug auf den Leonardumformer

In Abb. 4.1 sieht man, dass die Fremderregung über eine externe Spannungsquelle versorgt wird. Diese kann man noch zusätzlich mit einem Stellwiderstand regeln. Wie schon erläutert, ist es möglich den Synchrongenerator eine variable Betriebsspannung zu vergeben. Das liegt daran, dass der Synchrongenerator als Last an der Welle angeschossen ist. Der Synchrongenerator wiederum speist Leistung ins Netz. Der Gleichstromgenerator bzw. Motor spielt hier eine besonders große Rolle.

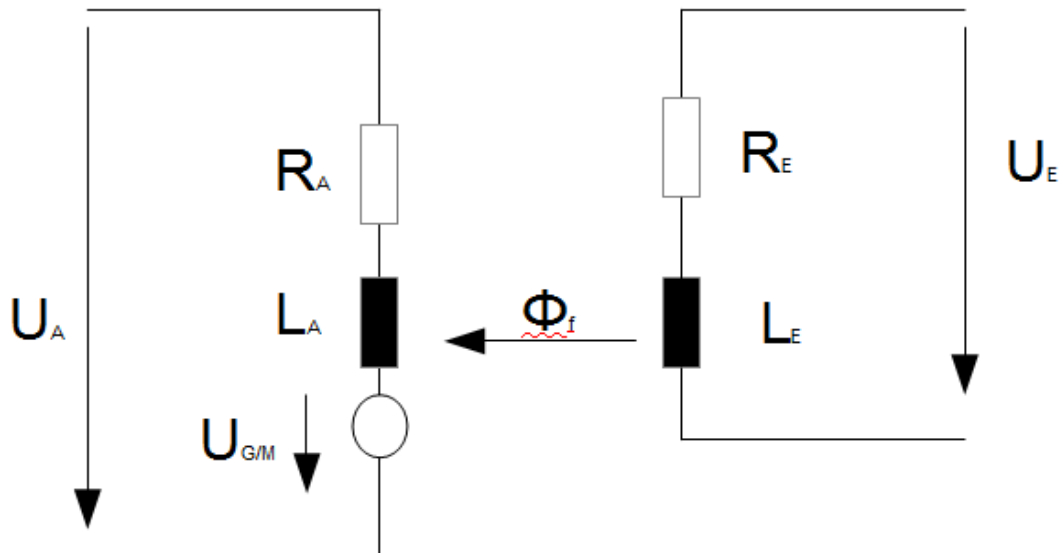


Abbildung 4.4: Gleichstrommaschine

Die Formel für die Ankerspannung der Gleichstrommaschine lautet allgemein:

$$U_A = C_e \cdot \Phi_f \cdot n + R_A \cdot I_A \quad (4.2.4.1)$$

mit  $U_{hG}$  bzw  $U_{hM} = C_e \cdot \Phi_f \cdot n$

$U_A$  – Ankerspannung

$C_e$  – Maschinenkonstante

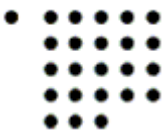
$\Phi_f$  – magnetischer Fluss der Fremderregung

$n$  – Drehzahl der Maschine

$R_A$  – Ankerwiderstand

$I_A$  – Ankerstrom

Durch den Stellwiderstand(in Abb. 4.1) kann man eine niedrigere Ankerspannung justieren, da



$I_f = f(\Phi_f)$  und somit  $I_f \propto \Phi_f$  ist. Somit hätte man die Möglichkeit, die Ankerspannung  $U_A$  zwischen minimale Ankernennspannung  $-U_{AN}$  und maximale Ankernennspannung  $+U_{AN}$  zu variieren. Bei der Umstellung der Formel 4.2.4.1 für den Ankerstrom ergibt sich dann

$$I_A = \frac{U_A - U_{hG/hM}}{R_A} \quad (4.2.4.2)$$

Angenommen die Erregerspannung  $U_E$  vom Gleichstromgenerator verringert sich, wonach auch  $U_A$  sich verringert, würde sich dabei der Ankerstrom im Ankerkreis kehren, da die Fremderregung des Gleichstrommotors konstant bleibt. Diesen Vorgang nennt man auch „Bremsvorgang“. Sprich der eigentliche Gleichstromgenerator ist als Motor für den Asynchronmotorantrieb zuständig. Aber gleichzeitig wird Leistung ins Netz eingespeist über den Asynchronmotor. Das heißt über die Fremderregung des Gleichstromgenerators kann man die Drehzahl des Gleichstrommotors verstellen. Somit auch die Leistung für die Einspeisung in das Netz.

#### 4.2.4.2 Fazit

Nachteile hierbei sind, dass der Leonardumformer an sich eine große Fläche an Raum beansprucht, der Gesamtwirkungsgrad nicht gut genug ist und unangenehme Geräusche von sich gibt. Dafür ist die Technik sehr leicht anwendbar, die Netzströme bleiben sinusförmig und es hat eine gute Regelung.

### 4.2.5 Die Dunkelschaltung

Beim Anschließen einer Synchronmaschine in das Netz ist zu nächst einmal die Synchronisation erforderlich. Diese Synchronisation ist realisierbar wenn Phasenlage, Phasenfolge, Betrag und die Frequenz der Spannung bei Netz als auch Synchronmaschine übereinstimmen. Um so etwas zu beobachten wurde die Dunkelschaltung (wie in Abb. 4.1) als Hilfsmittel erstellt. Vergrößert man die Schaltung und vergibt man den Phasen der Synchronmaschine Namen wie u, v und w, sieht die Schaltung wie folgt aus:

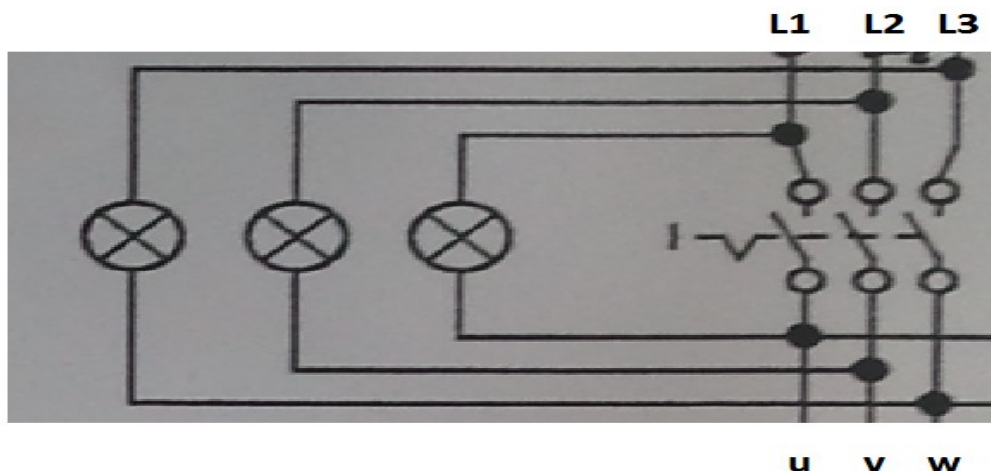
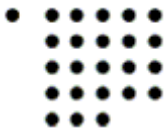


Abbildung 4.5: Dunkelschaltung



Diese besteht aus drei Leuchtlampen und überbrückt den Hauptschalter. Dabei wird über Phase L1 - u, L2 - v und L3 - w überbrückt. Flackern die Leuchtlampen so ist die gleiche Phasenlage nicht erreicht. Verlangsamt sich das Flackern, so nimmt der Wert der Phasendifferenz ab. Existiert kein Flackern und Licht, so existiert keine Phasendifferenz, die Synchronisation ist nun möglich und der Hauptschalter bzw. Schutzschalter (Abb.4.5, Rechts) darf betätigt werden. In einem Zeigerdiagramm sieht das ganze dann folgenderweise aus :

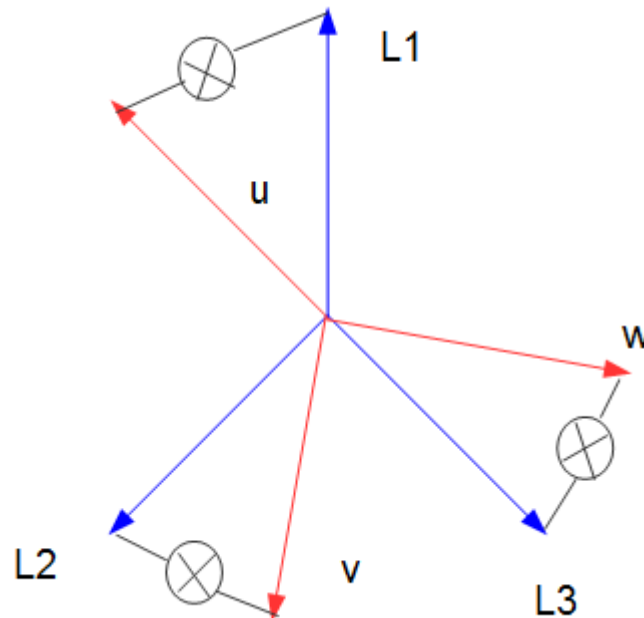


Abbildung 4.6: Zeigerdiagramm

#### 4.2.5.1 Fazit

Die Dunkelschaltung ist eine sehr gute Methode um die Synchronisation festzustellen. Aber auch Leuchtlampen haben eine bestimmte Lebensdauer. LabVIEW bietet hierbei LED-Anzeigen an, der Vorteil hierbei ist, dass dieser verschiedene Farben als Anzeige annehmen kann. Die gelbe Farbe die man mit dem Auge wahrnimmt, ist zu meist nicht ausreichend.

### 4.3 Begründung der Materialliste

#### 4.3.1 Nationalinstruments LabVIEW

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) ist eine Programmiersprache, die als Hilfsmittel zur einfachen Erstellung von Messanwendungen entwickelt wurde. Labview hat die Eigenschaft einer Datenflussprogrammierung, wie im Blockdiagramm zu sehen ist (Abb 4.8). Es kann technische Prozesse und Instrumente so abbilden, wie sie in der Realität auch üblich sind (siehe Frontpanel Abb. 4.7). Das bedeutet LabVIEW ist eine Entwicklungsumgebung und gleichzeitig eine graphische Programmiersprache. Positiv ist die leichte Erlernbarkeit des Programmes gegenüber Andere, da diese unübersichtlich, kompliziert sind und meistens unerwartete Fehler anzeigen.



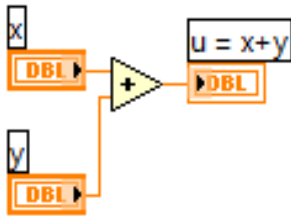
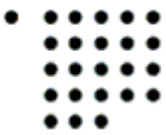


Abbildung 4.8:  
Blockdiagramm

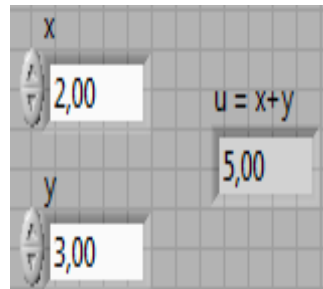


Abbildung 4.7: Frontpanel

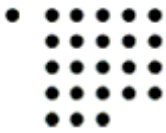
Ein Beispiel in Abb 4.7, ein DBL-Wert (Fließkommazahl)  $x = 2$  wird addiert mit dem DBL-Wert (Fließkommazahl)  $y = 3$ . Aus der Operation resultiert die neue Variable „ $u = x + y$ “, welche den Wert 5 bekommt. Wie man beobachten kann, werden die Operationen durch Funktionsblöcke erstellt, auch VIs (Virtual Instruments) genannt. Oft benutzte Funktionen sind auch Express-VI's, diese können z.B. von einem VI mehrere Attribute haben, so dass man nicht aus einer Bibliothek mit ganz vielen VIs der gleichen Funktion eine raus suchen muss, sondern diese einstellen kann. Zusätzlich zu der Software werden hardwaretechnische PCI-Karten mit den Treibern und den dazugehörigen Messbuchsen von NI angeboten. Einige haben A/D-Wandler integriert, um analoge Werte aufzunehmen und diese in digitale umzuwandeln (wie in der Tabelle 3.1). Aufgrund dieser Eigenschaften wurde LabVIEW verwendet mit der PCI-6259. Auch der BNC-2111 Aufsatz mit den Kabeln. Wie in der Materialliste (Tabelle 3.1) schon erklärt, kann die PCI-Karte über die BNC-2 Buchsen 32 Kanäle einlesen. Da insgesamt 12 AC-Messsignale und 8 DC-Messsignale zu erfassen sind, die 2 Buchsen nicht nur geeignet, man kann sogar die AC und DC Signale jeweils für eine Buchse belegen. Da die beiden Buchsen sich auf eine Masse (Singel-Ended) beziehen und somit sich nicht gegenseitig stören können.

### 4.3.2 Funktionsgenerator (Testgenerator)

Wichtig ist hierbei, dass eine einfache Wechselspannung generiert wird, um zu testen, ob die Messdatenerfassung - VI den Ansprüchen angemessen ist. Da die hardwaretechnische Bearbeitung in Laufen war.

### 4.3.3 Echtzeitbetriebssystem

Echtzeitbetriebssystem ist zu nächst einmal ein herkömmliches Standardbetriebssystem. Mit dem Unterschied, dass bei dem Echtzeitbetriebssystem bestimmte Dienste und Anforderungen erfüllt werden müssen. Auch RTOS (Real-Time Operating System) hat das Betriebssystem mit zusätzlicher Echtzeitfunktion, Zeitbedingungen und die Vorhersagbarkeit des Systemverhaltens einzuhalten. In der Regel sind Echtzeitbetriebssysteme deutlich kompakter als Standardbetriebssysteme und kommen häufig in eingebetteten Systemen zum Einsatz.



Der RTOS hat folgende Anforderungen gegenüber ein herkömmliches Standardbetriebssystem:

- Er besitzt ein effizienten Prozesswechsel (Benutzerdefiniert nach Prozesspriorität)
- Interruptbehandlung
- minimaler Speicherbedarf(je größer der Speicher umso Höher der Aufwand )
- eingestellte Reaktionszeit
- Zeitdienste(absolute, relative Uhren, Timeouts, Weckdienste, etc.)
- einfaches Speichermanagement
- Unterstützung für Ein-/Ausgabe (z.B. Zugriff auf Hardware-Register)

Betriebssystem	Ziele eines Standard-OS:	Ziele eines RTOS:
Behandlung Prozesse	Gleichbehandlung aller Prozesse(Batchprozess)	Behandlung nach Priorität ,unwichtige Prozesse werden nicht wie wichtige behandelt
Gesamtbehandlung	Optimierung auf den Gesamtumsatz	kritische Prozesse verdrängen weniger wichtige
Gesamtbehandlung		vorhersagbare Laufzeiten sind wichtiger als optimaler Gesamtdurchsatz

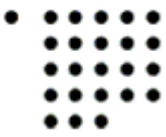
Tabelle 4.1: Echtzeitbetriebssystem

#### 4.3.3.1 Fazit

Ein Betriebssystem wie RTLinux wäre geeigneter für die Schreiberkurven gewesen. Da bei Windows 7 Latenzzeiten entstehen. Diese sorgen für eine Verzögerung der Messaufnahme. Da leider hierfür der Aufwand für das eigentliche Projekt viel zu riskant war, kann man trotz dessen im Nachhinein das Programm auf ein besseres Betriebssystem durchlaufen lassen.

#### 4.3.4 BNC-Buchse,PCI-Karte und Kabel

Die BNC-2111, welches Messsignale aufnimmt ist über das NI SHC68-68-EPM Kabel an die PCI-6259(AD-Wandler Karte) angeschlossen. Die PCI-6259 ist wiederum an ein PC-Slot im Rechner angeschlossen. Über ein LabVIEW-kompatibles Betriebssystem wird die Software von Labview installiert. Hierbei sind Messaufnahmen und Verarbeitungen möglich. Wie schon in Kapitel 3 erläutert existieren 2 BNC- Buchsen. Eine für AC und eine für DC, da die Signale auf eine Masse geerdet sind ist eine Störung beider Signale fast unmöglich. Weiterhin ist die Signalzuordnung sinnvoll, weil es die Verarbeitung beim Programmieren erleichtert.(Weitere Informationen bezüglich der Installation im Anhang Kapitel 7.2 )



## 4.4 Nyquist-Theorem

Um ein Signal auf ein Digitaloszilloskop sichtbar zu machen ist die Theorie in der Signalverarbeitung erforderlich. Da die A/D-Karte eine Reihe von Abtastungen über eine Abtastfrequenz durchführt.

### 4.4.1 Theorie zur Auflösung der Oszilloskopbilder

„Jede periodische Funktion  $t \rightarrow f(t)$  mit der kleinsten Periode  $T$ , die aus einer Summe von endlich vielen Sinus- und Kosinusfunktionen mit  $\omega = \frac{2\pi}{T}$  besteht und nach der Formel

$$f(t) = \sum_{k=-n}^n c_k e^{jk\omega t} \quad (4.4.1.1)$$

gebildet ist (d.h., die höchste auftretende Frequenz ist  $f_{max} = \frac{\omega n}{2\pi} = \frac{n}{T}$ ), kann exakt rekonstruiert werden, wenn man die Funktionswerte an mehr als  $2n$  äquidistanten Stellen innerhalb einer Periode  $T$  erfasst (abgetastet) hat. Als Formel:

$$f_{ab} > 2 \cdot f_{max} \quad [5]$$

mit

$f_{ab}$  - Abtastfrequenz

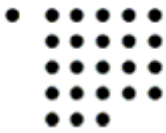
$f_{max}$  - tatsächliche maximale Frequenz

Bemerkung:

- Das die Abtastfrequenz doppelt so groß wie die tatsächliche maximale Frequenz sein soll ist falsch. Sie muss größer als das Doppelte der tatsächlichen maximalen Frequenz sein. Bsp:  
 $f_{max} = 1\text{kHz} \Rightarrow f_{ab} > 2\text{kHz}$  also beispielsweise 2,001kHz.
- Um eine Funktion im Zeitbereich bei Labview zu erstellen sollte man wesentlich höhere Abtastfrequenzen nehmen wie z.B.  $f_{maxlv} = 10 f_{max}$ . Da Labview nicht mit Sinuslinien sondern mit einfachen geraden Stücken arbeitet ist das hier sinnvoll.

Beispiel:

Hier hat das Eingangssignal 1Hz und tastet mit 5 Hz ab, das Bild ähnelt zwar einem Sinus ist aber dennoch ungenau:



Frequenz des Eingangssignals - 1 Hz  
Abtastrate = 5 Hz (5 Werte/Periode: verbunden)

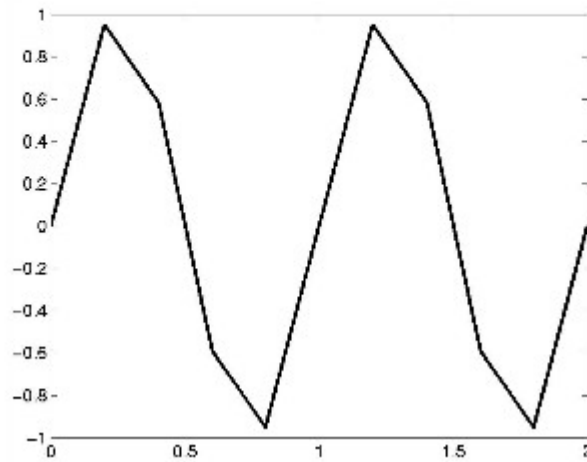


Abbildung 4.9: Eingangssignal mit 1Hz und Abtastfrequenz 5 Hz

Hier das 10fache des Eingangssignales viel genauer ist:

Frequenz des Eingangssignals - 1 Hz  
Abtastrate = 10 Hz (10 Werte/Periode: verbunden)

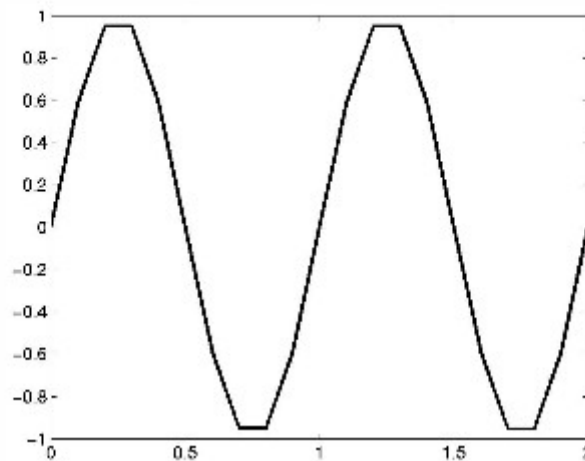
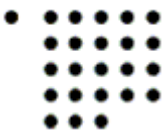


Abbildung 4.10: Eingangssignal mit 1Hz und Abtastfrequenz 10 Hz

[6]

- Bei einer Abtastfrequenz die nicht den Fall  $f_{ab} > 2 \cdot f_{max}$  erfüllt ist das Programm gezwungen abzurechnen und eine Fehlermeldung zu berichten.



Die Frequenzauflösung auf dem Digitaloszilloskop wird dann mit folgender Formel berechnet:

$$f_{\text{Auf}} = \frac{f_{\text{ab}}}{N} \quad (4.4.1.2)$$

Wobei hier  $f_{\text{auf}}$  die Frequenzauflösung ist,  $f_{\text{ab}}$  die Abtastfrequenz und  $N$  die Anzahl der Abtastungen ist. Auch notwendig ist bei der Frequenzdedektion, die Zahl der Abtastungen, die für eine Periode berechnet wird:

$$N_T = \frac{f_{\text{ab}}}{f_T} \quad (4.4.1.3)$$

hier  $N_T$  - Anzahl der Abtastung für eine Periode

Mit der Anzahl der Abtastung für eine Periode ist es möglich Berechnungen wie quadratischer Mittelwert, Mittelwert, Phasenverschiebung usw. durchzuführen. Wenn man nun wissen will wie viele Perioden auf ein Digitaloszilloskop aufgelöst wird, teilt man nun  $N$  durch  $N_T$ :

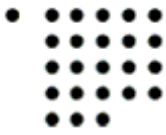
$$N_A = \frac{N}{N_T} \quad (4.4.1.4)$$

mit  $N_A$  - Anzahl der Perioden

## 4.5 Überlegung zur Berechnung der Leistungen (S, P, Q) in Labview

Als Prioritäten müssen die Wechselgrößen symmetrische Verhalten aufweisen, ist das nicht der Fall sollten sie umgerechnet werden, damit eine Berechnung der Leistungen überhaupt möglich ist. Berechnet werden sollen die zugeführte, abgeführte oder erregte Schein-, Wirk- aber auch Blindleistung des Asynchronmotors, Synchrongenerators bzw. der Gleichstrommaschinen.

Ein triviales rechnen der Wechselgrößen, die schon mit den Integralen berechnet wurden, über  $S = U \cdot I$ ,  $P = U \cdot I \cdot \cos(\phi)$  oder  $Q = U \cdot I \cdot \sin(\phi)$  ist zwar auch möglich (Auch bei Gleichstrom:  $P = U \cdot I$ ). Aber da das Ziel sich auf eine Messdatenerfassung richtet, damit die Messdatenerfassung so genau wie möglich und präzise sein soll., ist hier ein besonderes Verfahren der Darstellung der Leistungen erforderlich.



### 4.5.1 Wirkleistung P

Allgemein gilt für die Wirkleistung P :

$$P = \bar{p} = \frac{1}{T} \int_{t_0}^{T+t_0} P(t) dt = \frac{1}{T} \int_{t_0}^{T+t_0} U(t) \cdot I(t) dt \quad [7][8] \quad (4.5.1.1)$$

Mit

*P – Wirkleistung*  
*P(t) – zeitabhängige Wirkleistung*  
*T – Periodendauer*  
*U(t) – Zeitabhängige Spannung*  
*I(t) – Zeitabhängige Strom*  
*t<sub>0</sub> – Anfangszeitpunkt der Periode*

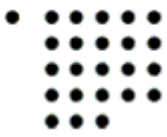
Da wir aber hier mit Abtastfrequenzen und einer bestimmten Zahl an Abtastungen zu programmieren haben, ist eine triviale Integration zwar auch möglich, aber dennoch, aufgrund der ungenauen Periode als auch Zeit, immer noch ungenau. Deshalb ist hier numerisch mit den instantanen Werten zu rechnen. Für jeden Zeitpunkt wird Strom bzw. Spannung über die Sensoren und galvanisch getrennten Bauteile erfasst, verarbeitet und von der AD- Karte eingelesen. Diese instantanen Werte kann man zum Beispiel in einen 2D - Array abspeichern und je nach Größe (hier Spannung und Strom) abschneiden und auswerten. Diese Verarbeitung sieht dann, mit  $i = 0$  bis  $n \cdot N_T - 1$  Messwerten, folgender weise aus:

$$P = \frac{1}{n \cdot N_T} \sum_{i=0}^{n \cdot N_T - 1} U(i) \cdot i(i) \quad (4.5.1.2)$$

Mit

*P – Wirkleistung*  
*N<sub>T</sub> – Anzahl der Abtastung pro Periode*  
*U(i) – 1D – Arrayabhängige momentane Spannung*  
*I(i) – 1D – Arrayabhängige momentane Strom*  
*n – Anzahl der Perioden*

In Formel 4.6.1.2 wird um 1, in dem großen Sigma, dekrementiert weil das Array an 0ter Stelle als aufgenommenes Wert mitgezählt wird und beim Array-Wert  $n \cdot N_T - 1$  keine weitere nachträglich Periode folgt. Zu jedem  $i$  (im Integral: Stück  $dt$ ) wird ein  $U(i) \cdot i(i)$  auf die Summe dazu addiert (im Integral:  $U(t) \cdot i(t)$ ) und wird durch die Anzahl der Abtastung  $n$ -Periode  $n \cdot N_T$  (im Integral Periode:  $T$ , wenn  $n=1$ ) gemittelt. Da man beim Mittelwert über mehrere Perioden „ $n$ “ einen besseren bzw. genaueren Mittelwert bekommt, ist eine Mittelwertbildung über mehrere Perioden hier effektiv.



## 4.5.2 Scheinleistung S

Allgemein gilt für die Scheinleistung S :

$$S = \sqrt{\frac{1}{T} \int_{t_0}^{T+t_0} U(t)^2 dt} \cdot \sqrt{\frac{1}{T} \int_{t_0}^{T+t_0} I(t)^2 dt} = \frac{1}{T} \sqrt{\int_{t_0}^{T+t_0} U(t)^2 dt \cdot \int_{t_0}^{T+t_0} I(t)^2 dt} \quad [9] [8] \quad (4.5.2.1)$$

Mit

*S – Scheinleistung*

*T – Periodendauer*

*U(t) – Zeitabhängige Spannung*

*I(t) – Zeitabhängige Strom*

*t<sub>0</sub> – Anfangszeitpunkt der Periode*

Da wir aber hier mit Abtastfrequenzen und einer bestimmten Zahl an Abtastungen zu programmieren haben, ist eine triviale Integration zwar auch möglich, aber dennoch, aufgrund der ungenauen Periode als auch Zeit, immer noch ungenau. Deshalb ist hier numerisch mit den instantanen Werten zu rechnen. Für jeden Zeitpunkt wird Strom bzw. Spannung über die Sensoren und galvanisch getrennten Bauteile erfasst, verarbeitet und von der AD- Karte eingelesen. Diese instantanen Werte kann man zum Beispiel in einen 2D - Array abspeichern und je nach Größe (hier Spannung und Strom) abschneiden und auswerten. Diese Verarbeitung sieht dann, mit  $i = 0$  bis  $n \cdot N_T - 1$  Messwerten, folgender Weise aus:

$$S = \sqrt{\frac{1}{n \cdot N_T} \sum_{i=0}^{n \cdot N_T - 1} U(i)^2} \cdot \sqrt{\frac{1}{n \cdot N_T} \sum_{i=0}^{n \cdot N_T - 1} I(i)^2} = \frac{1}{n \cdot N_T} \sqrt{\sum_{i=0}^{n \cdot N_T - 1} U(i)^2 \cdot \sum_{i=0}^{n \cdot N_T - 1} I(i)^2} \quad (4.5.2.2)$$

*S – Scheinleistung*

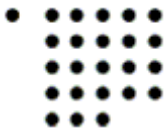
*N<sub>T</sub> – Anzahl der Abtastung pro Periode*

*U(i) – 1D – Arrayabhängige momentane Spannung*

*I(i) – 1D – Arrayabhängige momentane Strom*

*n – Anzahl der Perioden*

In Formel 4.6.2.2 wird um 1, in dem großen Sigma, dekrementiert weil das Array an 0ter Stelle als aufgenommenes Wert mitgezählt wird und beim Array-Wert  $n \cdot N_T - 1$  keine weitere nachträglich Periode folgt. Zu jedem  $i$  (im Integral: Stück  $dt$ ) wird ein  $U(i)^2$  und  $I(i)^2$  auf die Summe dazu addiert (im Integral:  $U(t)^2$  und  $I(t)^2$ ), am Ende wird die Wurzel gezogen und durch die Anzahl der Abtastung n-Periode  $n \cdot N_T$  (im Integral Periode: T, bei  $n=1$ ) gemittelt. Auch hier ist eine Effektivwertbildung über n-Perioden effektiver.



### 4.5.3 Blindleistung Q

Nun kann man ganz trivial die Blindleistung berechnen ,mit:

$$Q = \sqrt{S^2 - P^2} \quad (4.5.3.1)$$

### 4.5.4 Gesamtwirkleistung(Blindleistung,Wirkleistung und Scheinleistung)

Die Überlegungen für die Leistungen gelten nur für Phasenleistungen. Da hier im drei Phasensystem hantiert wird, ist hierbei die Berechnung für die Gesamtleistung (Wirk-, Blind- sowie Scheinleistung) des Synchrongenerators und Asynchronmotors auch ein wichtiger Faktor. Die Berechnung erfolgt dann mit folgender trivialen Formeln:

Für die Blindleistung gilt:

$$Q_{ges} = Q_1 + Q_2 + Q_3 \quad (4.5.4.1)$$

Für die Wirkleistung gilt:

$$P_{ges} = P_1 + P_2 + P_3 \quad (4.5.4.2)$$

und Scheinleistung:

$$S_{ges} = S_1 + S_2 + S_3 \quad (4.5.4.3)$$

[10]

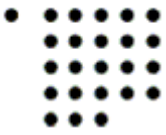
## 4.6 Überlegung zur Berechnung der Effektivwerte (Strom und Spannung) in Labview

Für Wechsignale haben einige Messgeräte die Aufgabe den Effektivwert von Strom und Spannung zu berechnen. Ebenfalls ist hier numerisch mit den instantanen Werten zu rechnen. Für jeden Zeitpunkt wird Strom bzw. Spannung über die Sensoren bzw. galvanisch getrennten Bauteile erfasst,verarbeitet und von der AD- Karte eingelesen. Effektivwert, Mittelwert und Spitzenwert sind bei Gleichspannung alle gleich, deshalb spielt es keine Rolle ob man dieses Verfahren nimmt. Hingegen ist es in der Spezifikation(Kapitel 4.7.4.2 )gefordert. Ähnlich wie in Kapitel 4.5.2 ,wird hier die gleiche Methode für Strom und Spannung benutzt:

$$A_{eff} = \sqrt{\frac{1}{N_T} \sum_{i=0}^{N_T-1} A(i)^2 dt} \quad (4.6.1)$$

$A_{eff}$  - Ist hier der Effektivwert für  $I_{eff}$  oder  $U_{eff}$





$A(i)$  - Ist hier der instantane Wert für  $I(i)$  oder  $U(i)$

$N_T$  – Anzahl der Abtastung pro Periode

#### 4.6.1 Effektivwert Spannung $U_{eff}$

Allgemein gilt für die Effektivwert Spannung:

$$U_{eff} = \sqrt{\frac{1}{T} \int_0^T U(t)^2 dt} \quad [11] \quad (4.6.1.1)$$

#### 4.6.2 Effektivwert Strom $I_{eff}$

Allgemein gilt für den Effektivwert Strom:

$$I_{eff} = \sqrt{\frac{1}{T} \int_0^T I(t)^2 dt} \quad [12] \quad (4.6.2.1)$$

### 4.7 Messprogramm

#### 4.7.1 Überarbeitete Messschaltung

Wie schon in Kapitel 4.1 erläutert ist die Messschaltung zu überarbeiten von einem weiteren Bachelorarbeiter. Dabei ist eine Kanaluweisung hier wichtig. Die überarbeitete Schaltung sieht dann folgenderweise aus

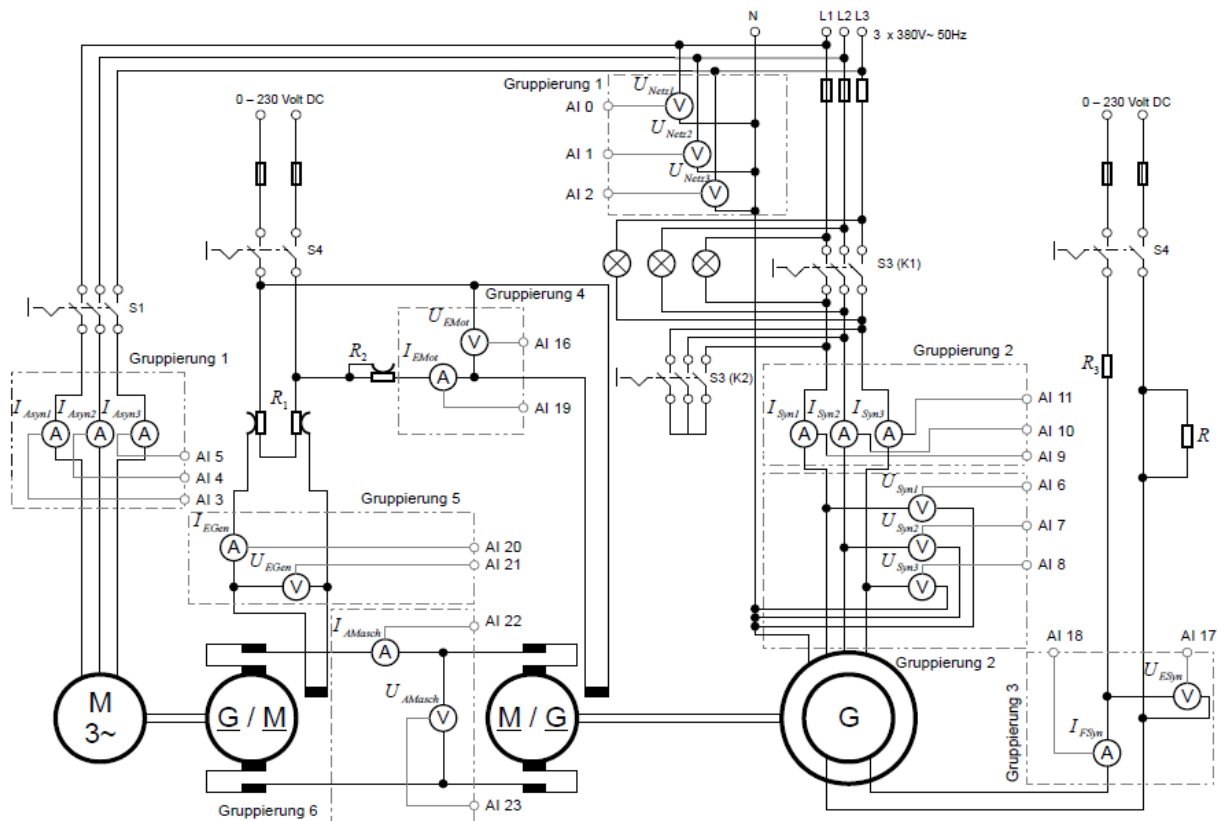
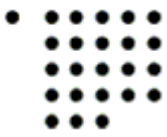


Abbildung 4.11: Überarbeitete\_Schaltung

Das besondere bei den Wechselspannungsmessung ist, dass sie noch einen zusätzlichen Null-Leiter bekommen haben. Damit wäre die Masse für die drei Phasenspannung geregelt. Die Lampen kann man bei Bedarf auch entfernen bzw. sich sparen. Gespart wird auch bei den Zeigerinstrumenten da diese mit den Sensoren und galvanisch getrennten Bauteilen ersetzt werden und mit dem LabVIEW-Programm realisiert werden.

#### 4.7.2 Vorwort

Um das Programm zu erstellen muss ein Gerüst erstellt werden um die Funktionen, die Eingabe- und Ausgabeverarbeitung des LabVIEW Programms zu verdeutlichen. Hierfür wurde eine Spezifikation vom Referenten erstellt und überarbeitet (Original im Anhang Kapitel 7.1), nach dem man sich richten soll, um die Haupteigenschaften des Programms zu verdeutlichen. Notfalls können Erweiterungen im Nachhinein bearbeitet werden.



## 4.7.3 Zu erfassende Messsignale und Kanalzuweisung

### 4.7.3.1 Messsignale

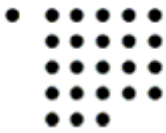
Vorgaben nach Elektrischen Schaltplan und Leistungsschilder der Maschinen							
Messgröße	Messgrößen Beschreibung	I Max/Min	U Max/Min	Frequenz	AD-Kanal /AI	Auflösung/Bit	Genauigkeit/%
U Netz1	Strangspannung L1	-	+/-400 V (AC)	50 Hz	0	>=12	<0.1
U Netz2	Strangspannung L2	-	+/-400 V (AC)	50 Hz	1	>=12	<0.1
U Netz3	Strangspannung L3	-	+/-400 V (AC)	50 Hz	2	>=12	<0.1
I Asyn1	Phasenstrom L1	+/-40 A (AC)	-	50 Hz	3	>=10	<0.5
I Asyn2	Phasenstrom L2	+/-40 A (AC)	-	50 Hz	4	>=10	<0.5
I Asyn3	Phasenstrom L3	+/-40 A (AC)	-	50 Hz	5	>=10	<0.5
U Syn1	Strangspannung L1	-	+/-400 V (AC)	50 Hz	6	>=12	<0.1
U Syn2	Strangspannung L2	-	+/-400 V (AC)	50 Hz	7	>=12	<0.1
U Syn3	Strangspannung L3	-	+/-400 V (AC)	50 Hz	8	>=12	<0.1
I Syn1	Phasenstrom L1	+/-40 A (AC)	-	50 Hz	9	>=10	<0.5
I Syn2	Phasenstrom L2	+/-40 A (AC)	-	50 Hz	10	>=10	<0.5
I Syn3	Phasenstrom L3	+/-40 A (AC)	-	50 Hz	11	>=10	<0.5
U Emot	Erregerspannung	-	230 V (DC)	50 Hz	16	>=8	<1
U Esyn	Erregerspannung	-	230 V (DC)	50 Hz	17	>=8	<1
I Fsyn	Erregerstrom	20 A (DC)	-	50 Hz	18	>=8	<1
I Emot	Erregerstrom	1,5 A (DC)	-	50 Hz	19	>=8	<1
I Egen	Erregerstrom	4 A – 2 A (DC)	-	50 Hz	20	>=8	<1
U Egen	Erregerspannung	-	230 V (DC)	50 Hz	21	>=8	<1
I Amasch	Ankerstrom	+/- 50 A (DC)	-	50 Hz	22	>=8	<1
U Amasch	Ankerspannung	-	148 – 400 V (DC)	50 Hz	23	>=8	<1

t Zeit seit Beginn Messintervall 0.. 86400s(1 Tag)

<10µs

Abbildung 4.12: Messsignale

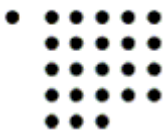
' - Anlaufstrom darf nicht zerstört werden



#### 4.7.3.2 Dynamische Spezifikation für alle Signale

Messgrößenbeschreibung	Symbol	Wert
Messintervall	$T_{\text{mess}}$	100 ms
Messpunkte pro Messintervall	$N_{\text{mess}}$	$\geq 1000$
Samplerate	$R_s$	$\geq 10 \text{ kS/s}$
Bandbreite der Sensoren	$F_g$	$\geq 20 \text{ kHz}$

Tabelle 4.2: dynamische\_Spez

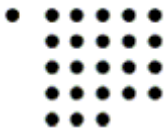


## 4.7.4 Von Messpunkt zu berechnen

### 4.7.4.1 Für jeden Messpunkt zu berechnen:

Messgrößenbeschreibung		Symbol	Berechnung
Instantane Leistungen der Synchronmaschine	Phase 1	$P_{\text{syn1inst}}$	$= U_{\text{syn10}} \cdot I_{\text{syn1}}$
	Phase 2	$P_{\text{syn2inst}}$	$= U_{\text{syn20}} \cdot I_{\text{syn2}}$
	Phase 3	$P_{\text{syn3inst}}$	$= U_{\text{syn30}} \cdot I_{\text{syn3}}$
Instantane Leistungen der Asynchronmaschine	Phase 1	$P_{\text{asy1inst}}$	$= U_{10} \cdot I_{\text{asy1}}$
	Phase 2	$P_{\text{asy2inst}}$	$= U_{20} \cdot I_{\text{asy2}}$
	Phase 3	$P_{\text{asy3inst}}$	$= U_{30} \cdot I_{\text{asy3}}$
Instantane Leistung der Gleichstrommaschinen		$P_{\text{amaschinst}}$	$= U_{\text{DC}} \cdot I_{\text{DC}}$
Instantane Leistung des Erregerkreis Synchrongenerator		$P_{\text{esyncinst}}$	$= U_{\text{esyn}} \cdot I_{\text{esyn}}$
Instantane Leistung des Erregerkreis Gleichstromgenerator		$P_{\text{egeninst}}$	$= U_{\text{egen}} \cdot I_{\text{egen}}$
Instantane Leistung des Erregerkreis Gleichstrommotors		$P_{\text{emotinst}}$	$= U_{\text{emot}} \cdot I_{\text{emot}}$
Spannung zwischen Netz und Synchrongenerator	Phase 1	$U_{1\text{dif}}$	$= U_{10} - U_{\text{syn10}}$
	Phase 2	$U_{2\text{dif}}$	$= U_{20} - U_{\text{syn20}}$
	Phase 3	$U_{3\text{dif}}$	$= U_{30} - U_{\text{syn30}}$

Tabelle 4.3: Berechnung Messpunkt

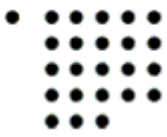


#### 4.7.4.2 Für jedes Messintervall zu berechnen

Effektivwerte

Messgrößenbeschreibung		Symbol	Berechnung aus
Netzspannungen	Phase 1	$U_{10rms}$	$U_{10}$
	Phase 2		
	Phase 3	$U_{20rms}$	$U_{20}$
Phasenströme Asynchronmaschine	Phase 1	$I_{asy1rms}$	$I_{asy1}$
	Phase 2		
	Phase 3	$I_{asy2rms}$	$I_{asy2}$
Phasenspannungen Synchrongenerator	Phase 1	$U_{syn10rms}$	$U_{syn10}$
	Phase 2		
	Phase 3	$U_{syn20rms}$	$U_{syn20}$
Phasenströme Synchrongenerator	Phase 1	$I_{syn1rms}$	$I_{syn1}$
	Phase 2		
	Phase 3	$I_{syn2rms}$	$I_{syn2}$
Spannung zwischen Netz und Synchrongenerator	Phase 1	$U_{1difrms}$	$U_{1difi}$
	Phase 2		
	Phase 3	$U_{2difrms}$	$U_{2dif}$
Zwischenkreisspannung Gleichstrommaschinen		$U_{amaschrms}$	$U_{amasch}$
Zwischenstrom Gleichstrommaschinen		$I_{amaschrms}$	$I_{amasch}$
Erregerspannung Synchronmaschine		$U_{esynrms}$	$U_{esyn}$
Erregerstrom Synchronmaschine		$I_{esynrms}$	$I_{esyn}$
Erregerspannung Gleichstromgenerator		$U_{egenrms}$	$U_{egen}$
Erregerstrom Gleichstromgenerator		$I_{egenrms}$	$I_{egen}$
Erregerspannung Gleichstrommotor		$U_{emotrms}$	$U_{emot}$
Erregerstrom Gleichstrommotor		$I_{emotrms}$	$I_{emot}$

Tabelle 4.4: Effektivwerte



### Wirkleistungen

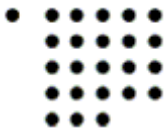
Messgrößenbeschreibung		Symbol	Berechnung aus
Synchronmaschine	Phase 1	$P_{syn1}$	$P_{syn1inst}$
	Phase 2	$P_{syn2}$	$P_{syn2inst}$
	Phase 3	$P_{syn3}$	$P_{syn3inst}$
	Gesamt	$P_{syn}$	$= P_{syn1} + P_{syn2} + P_{syn3}$
Asynchronmaschine	Phase 1	$P_{asy1}$	$P_{asy1inst}$
	Phase 2	$P_{asy2}$	$P_{asy2inst}$
	Phase 3	$P_{asy3}$	$P_{asy3inst}$
	Gesamt	$P_{asy}$	$= P_{asy1} + P_{asy2} + P_{asy3}$
Gleichstrommaschinen		$P_{amasch}$	$P_{amaschinst}$
Erregerwicklung Synchrongenerator		$P_{esyn}$	$P_{esyninst}$
Erregerwicklung Gleichstromgenerator		$P_{egen}$	$P_{egeninst}$
Erregerwicklung Gleichstrommotor		$P_{emot}$	$P_{emotinst}$

Tabelle 4.5: Wirkleistungen

### Scheinleistungen

Messgrößenbeschreibung		Symbol	Berechnung aus
Synchronmaschine	Phase 1	$S_{syn1}$	$= U_{syn10rms} \cdot I_{syn1rms}$
	Phase 2	$S_{syn2}$	$= U_{syn20rms} \cdot I_{syn2rms}$
	Phase 3	$S_{syn3}$	$= U_{syn30rms} \cdot I_{syn3rms}$
	Gesamt	$S_{syn}$	$= S_{syn1} + S_{syn2} + S_{syn3}$
Asynchronmaschine	Phase 1	$S_{asy1}$	$= U_{10rms} \cdot I_{asy1rms}$
	Phase 2	$S_{asy2}$	$= U_{20rms} \cdot I_{asy2rms}$
	Phase 3	$S_{asy3}$	$= U_{30rms} \cdot I_{asy3rms}$
	Gesamt	$S_{asy}$	$= S_{asy1} + S_{asy2} + S_{asy3}$

Tabelle 4.6: Scheinleistungen



### Blindleistungen

Messgrößenbeschreibung		Symbol	Berechnung aus
Synchronmaschine	Phase 1	$Q_{syn1}$	$= \text{sqrt} [S_{syn1}^2 - P_{syn1}^2]$
	Phase 2	$Q_{syn2}$	$= \text{sqrt} [S_{syn2}^2 - P_{syn2}^2]$
	Phase 3	$Q_{syn3}$	$= \text{sqrt} [S_{syn3}^2 - P_{syn3}^2]$
	Gesamt	$Q_{syn}$	$= \sqrt{[S_{syn}^2 - P_{syn}^2]}$
Asynchronmaschine	Phase 1	$Q_{asy1}$	$= \text{sqrt} [S_{asy1}^2 - P_{asy1}^2]$
	Phase 2	$Q_{asy2}$	$= \text{sqrt} [S_{asy2}^2 - P_{asy2}^2]$
	Phase 3	$Q_{asy3}$	$= \text{sqrt} [S_{asy3}^2 - P_{asy3}^2]$
	Gesamt	$Q_{asy}$	$= \sqrt{[S_{asy}^2 - P_{asy}^2]}$

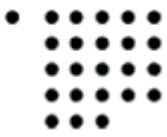
Tabelle 4.7: Blindleistungen

### Zeitabhängige Größen

Messgrößenbeschreibung	Symbol	Berechnung aus	Genauigkeit
Netzfrequenz	$f_N$	Fitfunktion aus $U_{10}, U_{20}, U_{30}$	1 mHz
Frequenz Synchrongenerator	$f_{syn}$	Fitfunktion aus $U_{syn10}, U_{syn20}, U_{syn30}$	1 mHz
Phasendifferenz zwischen Netzspannung und Synchrongeneratorspannung	$\varphi_{Nsyn}$	Fitfunktion aus $U_{10}, U_{20}, U_{30}, U_{syn10}, U_{syn20}, U_{syn30}$ ,	1°
Datum und Uhrzeit der Messung	$t_{int}$	Format: JJJJ-MM-TT HH:MM:SS.ss	10 ms

Tabelle 4.8: Zeitabhängige





## 4.7.5 Speichern der Daten

In dem folgenden Text wird die Art der Speicherung in Stichpunkte erläutert. Jede Datei hat Metadaten und einen Datenkörper. Metadaten sind Zusatzinformationen, welche die Verarbeitung der Daten beschreiben(hier im Kopf), diese sind Titel oder Kenngrößen. Mit dem Dateninhalt, worin die tatsächlichen Messdaten enthalten sind, werden die Messdaten als Tabellenanordnung in ASCII-Dateien abgespeichert.

### 4.7.5.1 Name und Format

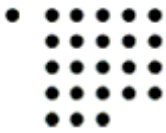
- Für Kommazahlen werden die Kommata mit Punkte ersetzt(globale Darstellung)
- Datumsformate werden in der Norm ISO 8601 als vollständige Darstellung in YYYY-MM-TT angegeben (mit YYYY -Jahr wie 1990,MM Monat wie 11 und TT Tag wie 12 )
- Uhrzeitformate werden in DIN5008 HH:MM:SS(mit „,sss“ notfalls) angezeigt (mit HH-Stunde wie 22, MM-Minute wie 30, SS-Sekunde wie 59 und sss-Millisekunde wie 100)
- Spalten sind über „Tab“s( [→] ) aneinander gereiht
- Zeilenvorschübe werden am Zeilenende angehängt
- Indizes für Variablenbezeichnung werden nach der Variable mit einem Unterstrich [ \_ ] angehängt

#### 4.7.5.1.1 Metadaten

- Die Metadaten sind Mehr- oder Zweispaltig und können viele Zeilen haben
- Von diesen Spalten sollte Eine bzw. Einige die Kenngrößenbezeichnung haben, die Andere bzw. Anderen den Kenngrößenwert
- Beginnend wird die Zeile mit der Kenngrößenbezeichnung Titel und seinem Wert „Titelnummer\_1“, anschließend folgen in der Nächsten „Date\_Time“ mit dem Zeitstempel und danach User mit dem „Benutzernamen“
- Die Metadaten sind vom Dateninhalt mit Zeilenvorschübe zu trennen

#### 4.7.5.1.2 Dateninhalt

- der Dateninhalt besteht aus einer Tabelle
- Für jede Spalte ist an der 0ten Zeile „Messdatensname/Messdatengröße“ zu versehen
- Die Spaltenanzahl der Messdatensnamen, müssen in der nächsten Zeile, mit der darauffolgenden Spaltenanzahl der Messdaten übereinstimmen
- Jede Zeile ist ein Messpunkt mit den angegebenen Messdaten
- Je nach Aufzeichnung werden Messdatenpunkte aufgezeichnet
- Nach der Tabelle folgt nichts mehr



#### 4.7.5.1.3 Muster

In der folgenden Aufzeichnung wurde ein Musterbeispiel für die Formatierung der Datei versehen

```
[→]Titel           [→]Meine erste Datei[←↓]
[→]Date_Time      [→]2013-08-27 [→] 10:29:10,333 [←↓]
[→]User           [→]Frau Max Mustermann[←↓]
[→]Urms1/V        [→]33.333333 [←↓]

[←↓]
Zeit/s            [→]Usyn1/V      [→]Usyn2/V      [→]Usyn3/V [←↓]
0.000            [→]0.123        [→]0.235        [→]10.356789 [←↓]
0.011            [→]0.123        [→]110.235      [→]9.345 [←↓]
0.025            [→]0.123        [→]0.235        [→]10.345678 [←↓]
```

#### 4.7.5.2 Oszilloskopkurvenaufzeichnung

- Die Oszilloskopkurven können aufgezeichnet und abgespeichert werden
- Alle gemessenen(Kapitel 4.7.4.1) und berechneten Werte(Kapitel 4.7.4.2) sind hierbei aufzuzeichnen
- Die berechneten Kenndaten( Tabelle 4.4,4.5,4.6 und 4.7)sind hierbei als Parameter abzuspeichern

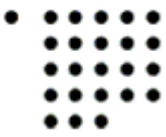
##### 4.7.5.2.1 Automatisierung Oszilloskopkurvenaufzeichnung

Gespeichert wird dann wie folgt:

- Mit dem Tastendruck des Users werden die aktuellen Oszilloskopkurven-Aufzeichnungen gespeichert
- Über eine benutzerdefinierte Zeit hat der Benutzer die Möglichkeit über Zeitabständen Oszilloskopkurven-Aufzeichnungen

##### 4.7.5.3 Kenndaten aus Oszilloskopkurvenaufzeichnung

- Alle Kennwerte(Kapitel 4.7.4.2) der Oszilloskopkurven werden in einer Datei ,unmittelbar nach einem Tastendruck, aufgezeichnet und mit einem anderen Tastendruck abgespeichert, somit auch die Aufzeichnung beendet
- Stürzt der Rechner bei einer Dateiaufzeichnung ab, sollte diese Datei trotz dessen abgespeichert in dem vorgegebenen Pfad sich befinden



#### 4.7.5.4 Namensauflösung der Datei

1. Alle abgespeicherten Dateien bekommen einen Zeitstempel für den Datum, wie in Kapitel 4.7.5.1 (YYYY-MM-HH)
2. Danach folgt ein Unterstrich der nachträglich eine 4-stellige Ziffer besitzt, für die Anzahl der Messung
3. Abgetrennt wird das ganze mit einem Bindestrich
4. Handelt es sich um eine Kenndatenaufzeichnung wird ein „SyncGen“ an dem Bindestrich angehängt
5. Bei „SyncGenOsc“ ist es die aktuelle Aufzeichnung der Oszilloskopbilder
6. Optional kann der Benutzer die Datei benennen, welches mit einem weiteren Bindestrich, an den Dateinamen angehängt wird
7. Die Datei wird als „.dat“ ganz hinten nochmal angehängt und abgespeichert

Die Abspeicherung kann dann so aussehen:

```
2013-08-25_0002-SyncGenOsc.DAT
2013-08-25_0013-SyncGen.DAT
2013-08-25_0001-SyncGen-Praktikumsversuch1.DAT
2013-08-25_0011-SyncGenOsc-Masterpraktikum_v3.DAT
```

### 4.7.6 Benutzeroberfläche

Die Benutzeroberfläche besteht aus 5 Fenstern die zum Beispiel in einer Art „Registerkarte“ rausgesucht und angezeigt werden kann. Für das Programm wird ein Breitbildmonitor mit der Auflösung 1920 pt x 1080 pt verwendet.

Folgende Fenster stehen zur Verfügung:

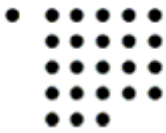
- Bedienung
- Einstellungen
- Kenndaten
- Oszillogramme
- Schreiberkurven

#### 4.7.6.1 Fenster „Bedienung“

Das Fenster Bedienung sollte, wie der Name schon sagt, immer extern neben der Registerkarte sein. Um die Bedienung neben der Ausführung sichtbar zu haben, ist es sinnvoll, dass Es einen schmalen Grad des Bildschirmes ausnutzt.

In der Bedienung befinden sich Schaltflächen als Taster und LEDs mit denen man operieren kann Diese sind:

- Programm beenden (Taster der das Programm beendet)
- Aktivieren der Speicherung von Kenndaten(Taster)
- Deaktivieren der Speicherung von Kenndaten(Taster)
- Speichern der aktuellen Oszilloskop-Aufzeichnung(Taster)
- Rücksetzen der Kennwerte(Taster)
- Start der Oszilloskop-Aufzeichnung(Taster)
- Stop der Oszilloskop-Aufzeichnung(Taster)
- Start der Kenndatenaufzeichnung(Taster)



- Stopp der Kenndatenaufzeichnung(Taster)
- Programm läuft(LED bei leuchten: Programm läuft , leuchtet es nicht: Programm beendet)
- Kenndatenspeicherung(LED bei leuchten: Programm zeichnet Werte in die Datei, leuchtet es nicht: zeichnet das Programm keine Werte mehr auf)
- Speichervorgang Oszilloskop-Aufzeichnung(LED)
- Eingabe und Anzeige für den Titel der aktuellen Kenndaten/Oszilloskop - Messung(String)
- Eingabe und Anzeige für den Namen der aktuellen Kenndaten/Oszilloskop - Messung(String)

#### **4.7.6.2 Fenster „Einstellungen“**

Ein- und Ausgabe der Oszilloskop-Aufzeichnung:

- Einstellen von Messintervall (Ausgabe)  $T_{mess}$  in ms über „Abtastfrequenz/f\_ab“ und „Anzahl der Messpunkte pro Messintervall“
- Anzahl der Messpunkte pro Messintervall(Eingabe)
- Abtastfrequenz/f\_ab(Eingabe)
- momentane Latenzzeit zwischen 2 Messintervallen(Ausgabe)

Eigenschaften der Messdatenerfassung:

- Ein- und Ausgabe von Name, Pfad und Beschriftung der Datei
- Für jeden Kanal ist eine Zuordnung mit dem Eingangsspannungsbereich erforderlich
- Ersatzkanäle(damit bei einem Kanalausfall, ein Weiterer zur Verfügung steht)
- Jedem Messsignal kann ein Skalierungsfaktor und Offset zugeordnet werden

#### **4.7.6.3 Fenster „Kenndaten“**

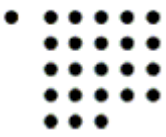
Alle Kenndaten sind als Zahlenanzeige und in Zeigerinstrumenten darzustellen.

##### **4.7.6.3.1 Zeigerinstrumente**

Alle Kenndaten werden in Zeigerinstrumenten im Zahlenanzeige dargestellt. Die Anzeige der Instrumente sollten so einen hohen Anzeigewert haben, dass der maximale Wert noch angezeigt werden kann.

##### **4.7.6.3.2 Dunkelschaltung**

Wie schon in Kapitel 4.2.5 erwähnt wird die Dunkelschaltung über 3 LED-Anzeigen im Programm ersetzt. Die Differenzspannung ist dann mit der Farbe „gelb“ zu erkennen. Das heißt es werden keine Zeigerinstrumente hier angewendet. Aber natürlich besitzt jede Differenzspannung eine zusätzliche Zahlendarstellung.



#### **4.7.6.4 Fenster „Oszillogramme“**

Diese Registerkarte besteht aus 9 XY-Diagrammen, die jeweils 4 Oszillosgramm-Messsignale darstellen können.

Die X-Achse enthält die Zeit der Messwerte, wobei die Y-Achse die Amplitude darstellt.

Alle Momentanwerte(Kapitel 4.7.4) und berechneten Momentanwert(Kapitel 4.7.3.1) sind hierbei auf den XY-Diagrammen darzustellen. Der Benutzer hat die Möglichkeit, im Programmablauf, von den 4 Messsignalen bei allen Diagrammen zu entscheiden, welchen Plot Sie/Er angezeigt haben möchte und welchen nicht.

#### **4.7.6.5 Fenster „Schreiberkurven“**

Diese Registerkarte besteht aus 146XY-Diagrammen, die jeweils 4 Schreiberkurven darstellen können. Alle Diagramme nutzen verteilt die Registerkarte.

Die X-Achse enthält die Zeit des in dem Zeitpunkt aufgenommenen Messwert, wobei die Y-Achse die Amplitude darstellt. Den Start der Zeit bestimmt das Rücksetzen oder das Starten des Programmes. Alle Kenndaten(Kapitel 4.7.4.2) sind hierbei auf den XY-Diagrammen darzustellen. Der Benutzer hat die Möglichkeit, im Programmablauf, von den 4 Messsignalen bei allen Diagrammen zu entscheiden, welchen Plot Sie/Er angezeigt haben möchte und welchen nicht.

#### **4.7.6.6 Formatierung von Diagrammen**

Die Diagramme sind klar und übersichtlich zu formatieren, so dass man diese in Textdokumenten oder Powerpointfolien mit einbinden kann.

Geeignet für die Diagramme sind:

- Weißer Hintergrund (RGB = 255, 255, 255)
- Einen schwarzen Rahmen und Achsenbeschriftungen (RGB = 0, 0, 0)
- Hellgraue Gitternetzlinien (RGB = 193, 193, 193)

Die Schriftgröße ist minimal 16 und die Schriftart sollte leserlich wie z.b. in Arial geschrieben sein. Alle Diagramme haben eine Überschrift versehen mit weißem Hintergrund und schwarze Schrift.

Die Überschrift sollte nahe über dem Diagramm stehen

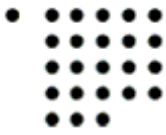
Jedes Plot erhält seine eigene Farbe und werden in der Weise bevorzugt:

Plot1: Orange (RGB = 255, 127, 0)

Plot2: Dunkelblau (RGB = 0, 0, 127)

Plot3: Grau (RGB = 127, 127, 127)

Plot4: Dunkel-Türkis (RGB = 0, 127, 127)



## 5 Programmierung in Labview

### ***5.1 Auflistung benutzter Express-VI's und VI Funktionen***

In diesem Kapitel befindet sich eine Tabellenauflistung aller benutzten Express-VIs und VI Funktionen. Die Tabelle soll bei unverständlichen VIs noch einmal zur Aufklärung helfen.

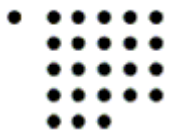



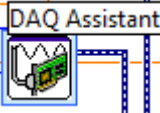






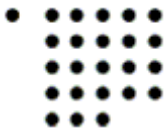
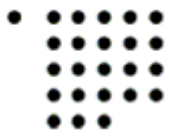


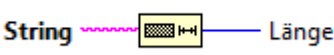
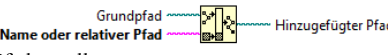
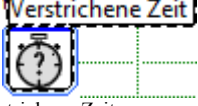
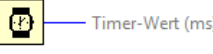
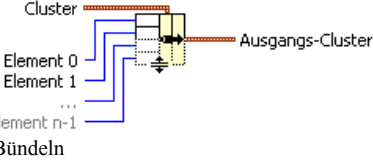
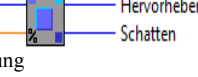
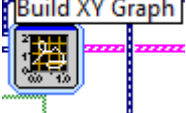


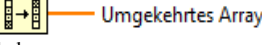
Bild und Name	Beschreibung	Input	Output	Typ
 <b>2D-Array</b> 2D-Array transponieren	Transponiert ein 2D-Array von [m,n] nach [n,m]	-2D-Array (2D-Array)	- <b>Transponiertes Array</b>	Funktion
 Aus Array entfernen	Entnimmt aus einem Array das Element (bei 1D) oder ein Array und gibt den gelöschten Teil und den Array ,ohne das Element bzw. Array, wieder aus	- <b>n-dimens. Array</b> - <b>Länge</b> - <b>Index 0..n-1</b>	- <b>Ausgangs-Array</b> - <b>Gelöschter Teil</b>	Funktion
 Messung von Frequenzkomponenten (Express-VI)	Dieses VI sucht sich den Spitzenwert der Schwingung (bsp. Dynamisches Signal) aus und gibt die Frequenz der Einzelfrequenzmessung mit der darauffolgenden Amplitude aus	- <b>Signale</b> - <b>Fehler (Eingang)</b>	- <b>Amplitude</b> - <b>Frequenz</b> - <b>Phase</b> - <b>Fehler (Ausgang)</b>	Express-VI
 DAQ-Assistant	Erstellt virtuelle Kanäle um Signale aufzunehmen. Dabei kann man die Anzahl der Abtastung, die Messsignale, Abtastfrequenz, das stoppen der Signalaufnahme, Fehlermeldung und die zeitliche Messaufnahme variieren	- <b>data</b> - <b>error in</b> - <b>number of samples</b> - <b>samplerates</b> - <b>stop</b> - <b>time out</b>	- <b>data</b> - <b>error out</b> - <b>stop</b> - <b>time out</b>	Express-VI
 Signal Trennen	Trennt ein kombiniertes Signal (bsp. dynamisches Signal), nach Signal 1,2,....	- <b>Kombiniertes Signal</b>	- <b>Signal 1</b> - <b>Signal 2</b> - <b>Signal x</b> ..	Express-VI
 Signale zusammenfassen	Kombiniert mehrere Signale 1,2,.... (bsp. dynamisches Signal), nach kombiniertes Signal	- <b>Signal 1</b> - <b>Signal 2</b> - <b>Signal x</b> ..	- <b>Kombiniertes Signal</b>	Express-VI
 Von dynamischen Daten konvertieren (Express-VI)	Wandelt dynamische Signale in Array, Zahlen oder boolesche Wert um	- <b>dynamisches Signal</b>	- <b>Array,Zahl, boolesches Wert</b>	Express-VI
 Datum/Zeit in Sekunden ermitteln	Gibt die Systemzeit als Zeitstempelformat von Labview aus, die nach dem [01-01-1904 00:00:00] verstrichen wurde		- <b>Aktuelle Zeit (Zeitstempel)</b>	Funktion
 Datum/Zeit formatieren	Formatiert ein Zeitstempelformat in String um, dabei kann über ein Zeitformatierungsstring (Input) bestimmt werden, wie das Zeitformat als String auszusehen hat	- <b>Zeitformatierungs-String</b> - <b>Zeitstempel</b> - <b>Weltuhr(boolesch)</b>	- <b>Datum/Zeit (String)</b>	Funktion
 In String formatieren	Formatiert Strings, Pfade, Enums, boolesche Werte, Zahlen und Zeitstempel in Textformate	- <b>Format-String</b> - <b>Eingangs-String</b> - <b>Fehler(Eingang)</b> - <b>Eingang1(0)...n(0)</b>	- <b>Ausgabe-String</b> - <b>Fehler(Ausgang)</b>	Funktion



<p>String 0 String 1 ... String n-1</p> <p>String verknüpfen</p>	<p>Verknüpft mehrere Strings zu einem</p>	<p>-String1 -String 2 .....</p>	<p>-verknüpfter String</p>	<p>Funktion</p>
<p>Cluster</p> <p>Element 0 Element 1 ... Element n-1</p> <p>Aufschlüsseln</p>	<p>Wird ein Cluster als Eingang angelegt, so entschlüsselt die Funktion den Cluster nach seinen individuellen Komponenten</p>	<p>-Cluster</p>	<p>-Element 0 -Element 1 -Element 2 ...</p>	<p>Funktion</p>
<p>Mehrzeilig? (F) Schreibweise ignorieren? (F) Alle ersetzen? (F) Eingangs-String Such-String String ersetzen(“”) Offset (0) Fehler (Eingang, kein Fehler)</p> <p>Ergebnis-String Anzahl der Ersetzungen Offset nach Ersetzung Fehler (Ausgang)</p> <p>String suchen und ersetzen</p>	<p>Ersetzt den ganzen String oder ein Teilstring des gesuchten Strings mit dem Ersetzstring</p>	<p>-Mehrzeilig?(boolesch,(f) std.-Wert) -Schreibweise ignorieren?(boolesch,(f) std.-Wert) -Alle ersetzen(boolesch,(f) std.-Wert) -Eingangs-String -Such-String -String ersetzen(„“) -Offset(0) -Fehler</p>	<p>-Ergebnis-String -Anzahl der Ersetzungen(Zahl ,long) -Offset nach Ersetzung -Fehler(Ausgang)</p>	<p>Funktion</p>
<p>Eingabeaufforderung (Dateip... Datei (Dialog verwenden) Text Fehler (Eingang) Referenz (Ausgang) Abgebrochen Fehler (Ausgang)</p> <p>In Textdatei schreiben</p>	<p>Schreibt ein String oder Array in eine Textdatei</p>	<p>-Eingabeaufforderung (String) -Datei(Pfad) -Text(String) -Fehler</p>	<p>-Referenz -Abgebrochen (boolesch) _fehler</p>	<p>Funktion</p>
<p>Format (%.3f) Dateipfad (Dialog, wenn leer) 2D-Daten 1D-Daten An Datei anhängen? (Neue Da... Transponieren? (nein:F) Trennzeichen (\t)</p> <p>Neuer Dateipfad (Kein Pfad,...</p> <p>In Tabellenkalkulationsdatei schreiben</p>	<p>Diese VI verarbeitet 1D-Arrays oder 2d-arrays in Textstring als Tabellenkalkulationsdatei</p>	<p>-Format(String) -Dialog(Pfad) -2D-Daten(2D-Array) -1D-Daten(1D-Array) -An Datei Anhängen(boolesch) -Transponieren (boolesch) -Trennzeichen (String)</p>	<p>-Neuer Dateipfad(Pfad)</p>	<p>Funktion</p>
<p>Array Element Element Element</p> <p>Angehängtes Array</p> <p>Array erstellen</p>	<p>Verbindet mehrere Arrays zu einem oder verarbeitet Elemente zu einem Array</p>	<p>-Array -Array -Array... oder -Element -Element...</p>	<p>-Angehängtes Array</p>	<p>Funktion</p>
<p>t s f</p> <p>s? t:f</p> <p>Auswählen</p>	<p>Gibt für s = „true“ den Wert von „f“ aus und umgekehrt „t“ (t und f müssen den gleichen Datentyp haben)</p>	<p>-t -s -f</p>	<p>-s?t:f</p>	<p>Funktion</p>
<p>Aktivieren Initialisieren</p> <p>Zähler Überlauf</p> <p>Inkrement(Punkt für Punkt)</p>	<p>Inkrementiert bei jedem Aufruf, mit der „true“-Bedingung („Aktivieren“) den Zähler, hat Initialisieren den Wert True wird eine 1 bei Zähler angezeigt aber es wird nicht inkrementiert</p>	<p>-Aktivieren(boolesch) -Initialisieren(boolesch)</p>	<p>-Zähler(Long) -Überlauf (Zeigt „true“ an bei Überlauf, boolesch)</p>	<p>Funktion</p>
<p>Wert Breite (-)</p> <p>Dezimal-Integer-String</p> <p>Zahl nach String(Dezimaldarstellung)</p>	<p>Wandelt den Wert mit der Breite in ein String. Dabei ist die Breite einstellbar. Ist der Wert eine Fließkommazahl so wird dieser in ein 64-bit-Integer umgewandelt</p>	<p>-Wert (Integerwert,Fließkomma zahl) -Breite(Integerwert)</p>	<p>- Dezimal-Integer-String</p>	<p>Funktion</p>





 <p>String-Länge</p>	<p>Von der Zeichenkette wird die Anzahl der Zeichen im Ausgang Länge ausgegeben</p>	<p>- String</p>	<p>-Länge(long)</p>	<p>Funktion</p>
 <p>Pfad erstellen</p>	<p>Über den Eingang „Name oder relativer Pfad“ als String wird ein Pfad als „Hinzugefügter Pfad“ erstellt und bei einer „Gruppfad“-Eingabe der String angehängt</p>	<p>-Grundpfad(Pfad) -Name oder relativer Pfad(String)</p>	<p>-Hinzugefügter Pfad(Pfad)</p>	<p>Funktion</p>
 <p>Vestrichene Zeit</p>	<p>Zeigt die Zeit an nach Zählbefehl</p>	<p>-Startzeit festlegen(DBL) -Zielzeit(s)(DBL) -Zurücksetzen(boolesch) -Automatisch zurücksetzen(boolesch) -Fehler(Eingang)</p>	<p>-Aktuell(s)(DBL) -Aktuell(Text)(String) -Verstrichene Zeit(Text)(String) -Zeit ist verstrichen(boolesch) -Verstrichene Zeit(Text)(String) -Fehler(Ausgang)(String) -Startzeit lesen(DBL)</p>	<p>Express-VI</p>
 <p>Warte(ms)</p>	<p>Wartet eine angegebene Zeit in Millisekunden und gibt den Endwert des Millisekunden-Timers aus</p>	<p>-Zu wartende ms</p>	<p>-Timer-Wert(ms)</p>	<p>Funktion</p>
 <p>Bündeln</p>	<p>Verbindet Elemente zu einem Cluster</p>	<p>-Element 0 -Element 1 .... -Cluster</p>	<p>-Ausgangs-Cluster</p>	<p>Funktion</p>
 <p>Farbe für Hervorhebung</p>	<p>Gibt eine Hervorhebung oder Schattierung der Eingangsfarbe aus. Die Hervorhebung bzw. Schattierung erfolgt über den Prozentualen Anteil</p>	<p>-Farbe -Prozentualer Anteil (30%(standart), ohne Wertzuweisung)</p>	<p>-Hervorheben -Schatten</p>	<p>Funktion</p>
 <p>XY Graph erstellen</p>	<p>Erstellt aus dem Eingang X und Y als dynamisches Signal den Graphen</p>	<p>-Aktivieren -X-Eingang -Y-Eingang -Zurücksetzen -Fehler(Eingang)</p>	<p>-XY-Graph -Fehler(Ausgang)</p>	<p>Express-VI</p>
 <p>In dynamischen Daten konvertieren</p>	<p>Wandelt Zahlen,boolesche Werte und Array in dynamischen Daten um</p>	<p>-Array,Zahl, boolesches Wert</p>	<p>-dynamisches Signal</p>	<p>Express-VI</p>
 <p>Cluster nach Array</p>	<p>Wandelt ein Cluster mit den beinhalteten Datenelementen in Array</p>	<p>-Cluster</p>	<p>-Array</p>	<p>Funktion</p>
 <p>1D-Array umkehren</p>	<p>Tauscht die Anordnung der Elemente im Array umgekehrt um</p>	<p>-Array</p>	<p>-Umgekehrtes Array</p>	<p>Funktion</p>

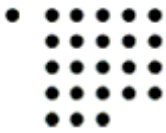


Tabelle 5.1: Auflistung benutzter Express-VI's und VI Funktionen

## 5.2 Quellcode

### 5.2.1 Sub-VI

Um die Überlegungen in den einzelnen Kapitel realisieren zu können sind Sub-VIs sehr Vorteilhaft. Diese Sub-VIs sind klein,kompakt (z.b. in Form eines Rechtecks) und haben nur Inputs und Outputs. Außerdem ordnen sie und verkleinern das Programm um Unübersichtlichkeiten zu vermeiden. Dadurch erleichtert es die Benutzung beim Programmieren.

#### 5.2.1.1 Sub-VI zur Bestimmung der Leistungen und Anzahl der Perioden

Die Überlegung für die Berechnungen der Leistung als auch „Anzahl der Perioden“ wurde in Kapitel 4.5 schon dargestellt. Um es in Labview ausführbar zu machen wurde hier extra eine Sub-VI erstellt , welches dann folgendermaßen aussieht:

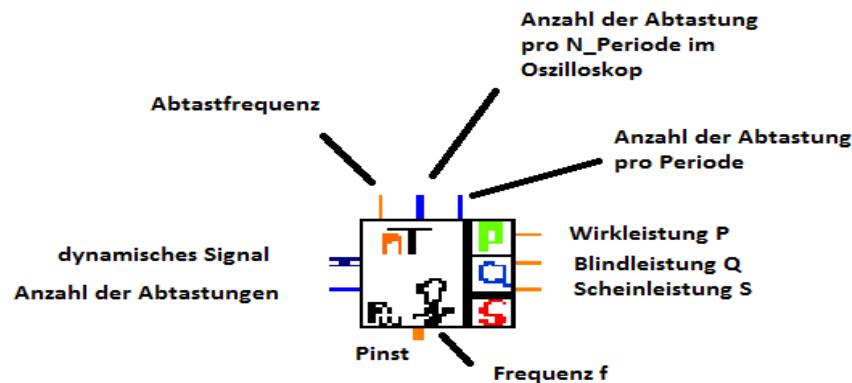
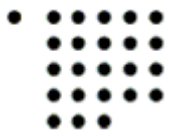


Abbildung 5.1: Sub-VI(Leistungsermittler,Anzahl\_Abtastung\_N\_Periode)\_Schaltbild

Die Inputs und Outputs haben dann folgende Daten:

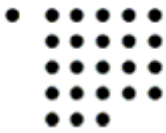


Bezeichnung im Sub-VI	Art des Gates	Datentyp	Anzeigeformat
Abtastfrequenz	Input	Doppelte Genauigkeit	Fließkomma
Dynamisches Signal	Input	Dynamisch	
Anzahl der Abtastungen	Input	Long	Fließkomma
Anzahl der Abtastungen pro N_Periode im Oszilloskop	Output	Long	Fließkomma
Pinst	Output	Doppelte Genauigkeit	Fließkomma
Anzahl der Abtastungen pro Periode	Output	Long	Fließkomma
Wirkleistung P	Output	Doppelte Genauigkeit	Fließkomma
Blindleistung Q	Output	Doppelte Genauigkeit	Fließkomma
Scheinleistung S	Output	Doppelte Genauigkeit	Fließkomma
Frequenz f	Output	Doppelte Genauigkeit	Fließkomma

Tabelle 5.2: Sub-VI(Leistungsermittler,Anzahl\_Abtastung\_p\_Periode)\_Tabelle

Für die Leistungen und Frequenz wurde Doppelte Genauigkeit beansprucht, weil man das Signal genau messen will mit mehr Nachkommastellen. Sinnvoll ist für die Kanäle mit „Anzahl“ den Longdatentyp zu verwenden, weil zum Beispiel der A/D-Wandler keine halbe Abtastung macht. Trotz dennoch kann man im Programm den Datentyp der Gatevergabe bei Restriktionen ändern. Allgemein:

Durch die Eingänge von „Abtastfrequenz“(Doppelte Genauigkeit), „Dynamisches Signal“(Dynamisch ) mit Strom und Spannung , „Anzahl der Abtastungen“ (Long) werden „Anzahl der Abtastungen pro N\_Periode im Oszilloskop“(Long), „Anzahl der Abtastungen pro Periode“(Long), „Wirkleistung P“(Doppelte Genauigkeit), „Blindleistung Q“(Doppelte Genauigkeit), „Scheinleistung S“(Doppelte Genauigkeit) ,und „Frequenz f“ (Doppelte Genauigkeit)ausgegeben.



In diesem Sub-VI befindet sich ein zusätzlicher Programmcode im Blockdiagramm von Labview, welches die Eingänge aufnimmt, diese verarbeitet und im Ausgang wieder ausgibt:

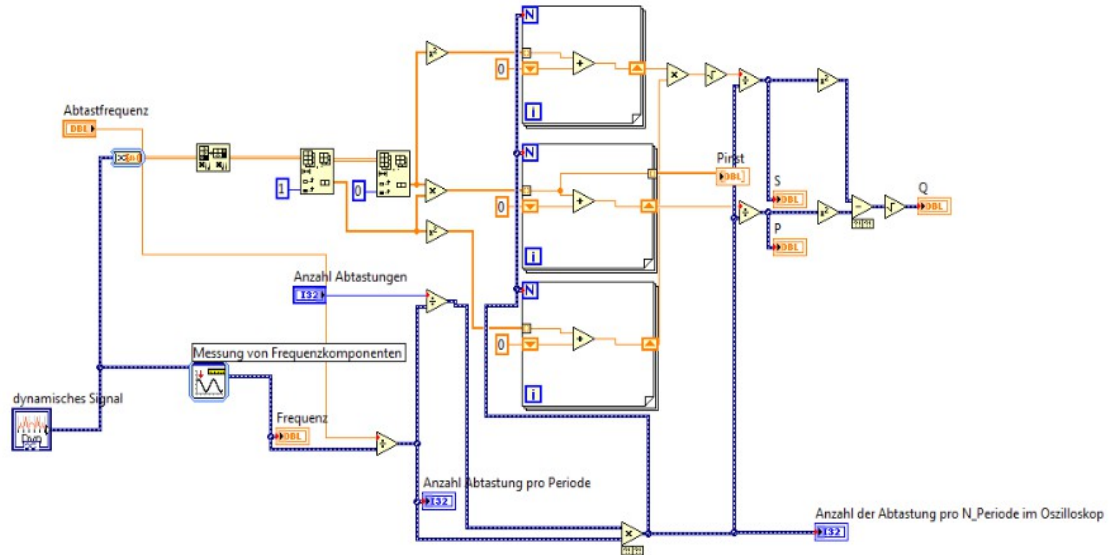


Abbildung 5.2: Sub-VI(Leistungsermittler,Anzahl\_Abtastung\_p\_Periode)\_Blockdiagramm

Am Eingang erwartet das Sub-VI eine Abtastfrequenz, ein dynamisches Signal und eine Anzahl von Abtastungen. Von dem dynamischen Signal wird die Frequenz über das Express-VI „Messung von Frequenzkomponenten (Express-VI)“ (in Abb. 5.2 das große hell-blau Rechteck) bestimmt. Dieses VI sucht sich den Spitzenwert der Schwingung aus und gibt die Frequenz der Einzelfrequenzmessung mit der darauffolgenden Amplitude aus. Somit kann man die erfasste Frequenz im Sub-VI über den Ausgang ausgeben. Das dynamische Signal wird gleichzeitig auch transponiert, diese Transponierung ist notwendig damit man mit der FOR-Schleife arbeiten kann. Die Transponierung erfolgt mit der Funktion „2D-Array transponieren (Funktion)“ (in Abb. 5.2 links oben das Block mit  $x_{j,i}$ ). Bei der Transponierung erfolgt folgendes:

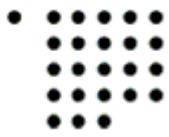
I	Wert1	Wert2	Wert3	.....
U	Wert1	Wert2	Wert3	.....

Tabelle 5.3: Sub-VI(Leistungsermittler,Anzahl\_Abtastung\_p\_Periode)\_Transponieren1

↓ Wird transponiert zu

I	U
Wert1	Wert1
Wert2	Wert2
Wert3	Wert3
.....	.....

Tabelle 5.4: Sub-VI(Leistungsermittler,Anzahl\_Abtastung\_p\_Periode)\_Transponieren2



Davor waren die I und U Werte (in der Tabelle 5.3) von Links nach Rechts verteilt .Nun sind die I und U Werte übersichtlich von oben nach unten verteilt (in der Tabelle 5.4) . Dies hat den Vorteil das die Funktion „Aus Array entfernen (Funktion)“(in Abb. 5.2 in der Mitte links oben) das 2-Dimensionale in 1D-Arrays ,mit den dazugehörigen Werten,ausschneiden kann. Dies sieht dann so aus:

U
Wert1
Wert2
Wert3
.....

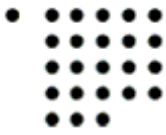
Tabelle 5.5: Sub-VI(Leistungsermittler,Anzahl\_Abtastung\_p\_Periode)\_Arraycut1

Und einmal

I
Wert1
Wert2
Wert3
.....

Tabelle 5.6: Sub-VI(Leistungsermittler,Anzahl\_Abtastung\_p\_Periode)\_Arraycut2

Beide Vektoren werden einmal quadriert , einmal miteinander multipliziert(in Abb. 5.2 mit dem Eingang in die zweite FOR-Schleife) , woraus ein neuer Vektor entsteht und jeweils in eine FOR-Schleife(in Abb. 5.2 die Rechtecke mit den Eselsohren) eingeführt (in Abb. 5.2 in der Mitte). Die FOR-Schleifen laufen so lange, bis das Array von Strom und Spannung n-Periode erreicht hat .Die Anzahl der Abtastung pro n-Periode erhält man durch die Division von Abtastfrequenz und Frequenz multipliziert mit der Anzahl der Perioden im Oszilloskop. Dieser fängt automatisch von 0 an zu zählen. In den FOR-Schleifen werden alle instantanen Werte jeweils, über den Schieberegister, zusammen addiert. Nach dem Durchlauf der Schleife , wird der Ausgang der ersten Schleife(in Abb. 5.2 von oben nach unten ) mit dem der dritten Schleife multipliziert und davon die Quadratwurzel gezogen. Dieser wird noch einmal durch die Anzahl der Abtastung pro n-Periode geteilt ,so dass daraus die Scheinleistung S resultiert. Teilen wir den Ausgang der mittleren FOR-Schleife durch die Anzahl der Abtastungen pro n-Periode , erhalten wir die Wirkleistung P. Quadriert man nun S als auch P ,subtrahiert das quadrierte P von dem quadrierten S und zieht von dem Wert die Quadratwurzel erhält man die Blindleistung Q. Die instantanen Wirkleistungen in der mittleren FOR-Schleife werden zusätzlich noch einmal ausgegeben, da man diese ,laut Kapitel



4.7.4.2 , braucht.

### 5.2.1.1.1 Prüfen der Sub-VI (Bestimmung der Leistungen und Anzahl der Perioden)

Um das Sub-VI nun zu prüfen wurde mit einem Funktionsgenerator(hier: HM8150) eine Sinusfunktion für Spannung und Strom generiert. Dabei hat die Spannung keine Phasenverschiebung hingegen der Strom(als Spannungswert) um  $180^\circ$ . Mit einer Frequenz von 50 Hz und einem Spitzen-Spitzen-Wert von 10 V wurde der Funktionsgenerator eingestellt. Trotz allem wäre der Strom als 10 A nicht Phasenverschoben ,dieses Problem wurde umgangen in dem man ganz trivial den Strom im Programm negiert hat. Berechnet man die Leistungen mit den trigonometrischen Funktionen , erhält man

$$P = \hat{U} \cdot \hat{I} \frac{1}{2} \cos(180) = \frac{1}{2} \hat{U} \cdot \hat{I} \cdot (-1) \quad (5.2.1.1)$$

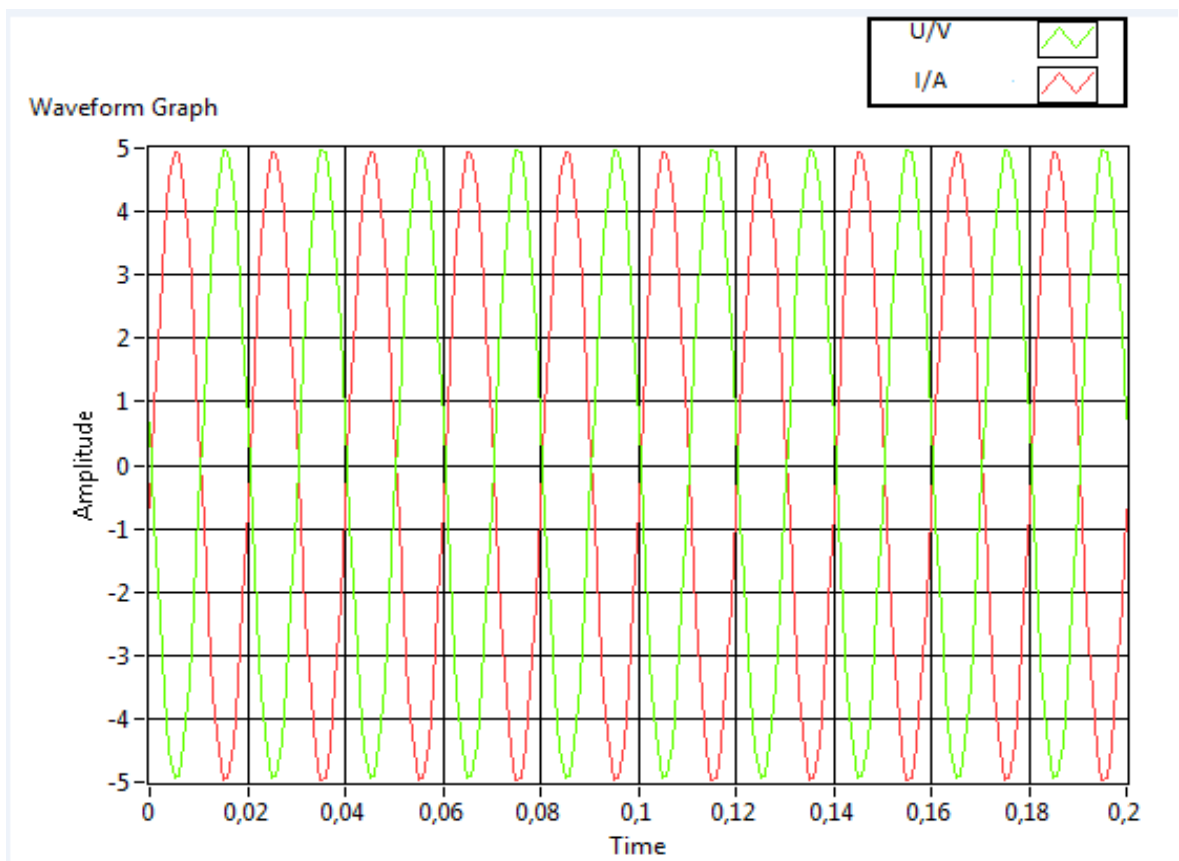
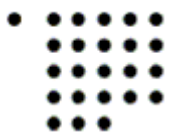


Abbildung 5.3: Test(Sub-VI Leistungsmessung)

Beim Digitaloszilloskop sehen die Werte dann wie folgt in Labview aus:

Man sieht , trotz 5V Amplitude, immer noch eine minimale Abweichung. Für die Wirkleistung P würde sich ein Wert von -12,25 W ergeben, hingegen die Scheinleistung S 12,25 VA und die Blindleistung damit 0 VA. Das Frontpanel mit dem Design gibt folgende Werte aus :



Da die Messung und Eingabe nicht zu 100% genau ist, entstehen Ungenauigkeiten der Messung. Trotz dennoch ist dieses Ergebnis für die Auswertung der Messung von Leistungen ausreichend. Eine alternative wären Express-VI's gewesen. Besser ist es aber die eigene VI zu erstellen, um auch das eigene Basiswissen über die Sub-VI zu haben.

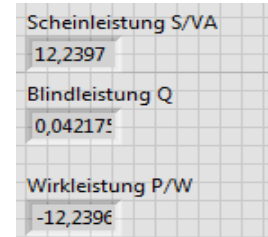


Abbildung 5.4:  
Test(Sub-  
VI\_Leistung)

### 5.2.1.2 Sub-VI zur Bestimmung des Effektivwertes

Für Kapitel 4.6 wurde dieses Design für das Schaltbild erstellt:

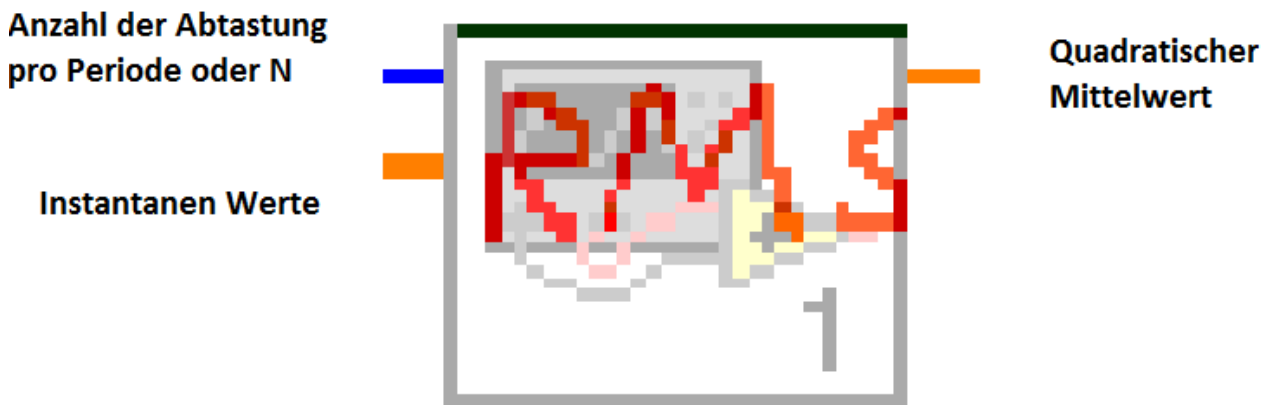
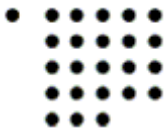


Abbildung 5.5: Sub-VI(Effektivwert)\_Schaltbild

Mit den dazugehörigen Kanälen:

Bezeichnung im Sub-VI	Art des Gates	Datentyp	Anzeigeformat
Anzahl der Abtastung pro Periode oder N	Input	Doppelte Genauigkeit	Fließkomma
Instantanen Werte	Input	Doppelte Genauigkeit	Automatisch Formatieren
Quadratischer Mittelwert	Output	Long	Fließkomma

Tabelle 5.7: Sub-VI(Effektivwert)\_Tabelle



Durch den Eingang der „Anzahl der Abtastung pro Periode oder N“ und „Instantanen Werte“ erhält man im Ausgang den quadratischen Mittelwert („Quadratischer Mittelwert“).

In diesem Sub-VI befindet sich ein zusätzlicher Programmcode im Blockdiagramm von Labview, welches die Eingänge aufnimmt, diese verarbeitet und im Ausgang wieder ausgibt:

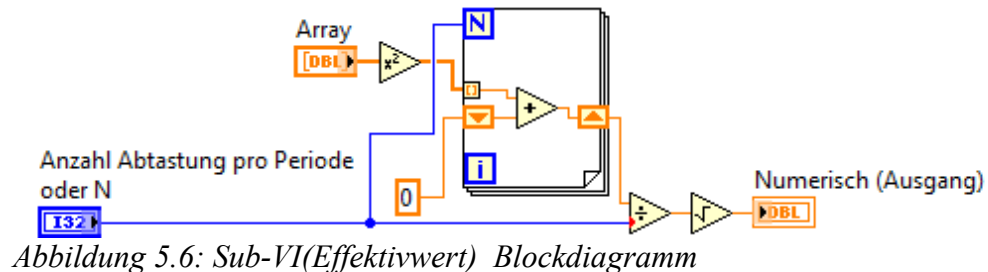


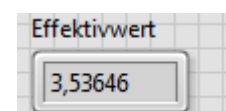
Abbildung 5.6: Sub-VI(Effektivwert)\_Blockdiagramm

Man gibt ein 1D-Array Messreihe in den Gate „Instantanen Werte“ ein. Diese (Zahlen im Array) werden quadriert und nacheinander nach i'ter-Stelle in der FOR-Schleife, bis zur vorletzten Zahl des Eingangs „Anzahl der Abtastung pro Periode oder N“ da das Array von 0 anfängt zu zählen, zusammen addiert. Danach wird geteilt durch „Anzahl der Abtastung pro Periode oder N“ und daraus die Wurzel gezogen. Nun erhält man den Quadratischen Mittelwert über den Ausgang „Quadratischer Mittelwert“. Diesen kann man für Strom, Spannung und andere Signale verwenden.

#### 5.2.1.2.1 Prüfen der Sub-VI Effektivwert

Um das Sub-VI nun zu prüfen wurde mit einem Funktionsgenerator(hier: HM8150) eine Sinusfunktion für Spannung generiert. Dabei hat die Spannung keine Phasenverschiebung. Mit einer Frequenz von 50 Hz und einem Spitzen-Spitzen-Wert von 10 V wurde der Funktionsgenerator eingestellt.

Der Effektivwert hat dann folgenden Wert



Da bei einer integrierten Sinusspannung gilt  $U_{eff} = \frac{\hat{u}}{\sqrt{2}}$ . Trotz nicht hundertprozentiger Gewährleistung ist die Prüfung ausreichend.

#### 5.2.1.3 Sub-VI Minuswert(Instantan)

Aus dem Kapitel 5.2.1.2 kann man den quadratischen Mittelwert hierbei als theoretische Grundlagen nehmen, wichtig ist hierbei, dass die Funktion 4.6.1.1 im Integral statt U(t) mit  $U_x(t) - U_y(t)$  ersetzt werden sollte. Im Schaltbild sieht dies dann so aus:



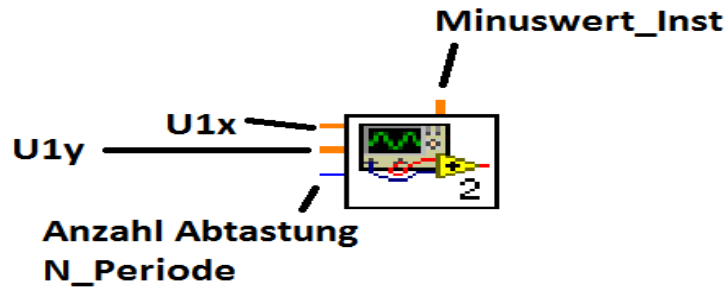
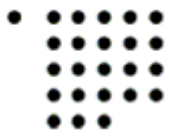


Abbildung 5.7: SubVI(Schaltbild)\_Minuswert

Mit den dazugehörigen Kanälen:

Bezeichnung im Sub-VI	Art des Gates	Datentyp	Anzeigeformat
U1x	Input	Doppelte Genauigkeit	Fließkomma
U1y	Input	Doppelte Genauigkeit	Fließkomma
Anzahl Abtastung N_Periode	Input	Long	Fließkomma
Minuswert_Inst	Output	Doppelte Genauigkeit	Fließkomma

Das VI gibt den instantanen Spannungsdifferenzwert(Minuswert\_Inst) über die Verknüpfung U1x,U1y und Anzahl Abtastung N\_Periode wieder.In diesem Sub-VI befindet sich ein zusätzliches Programmcode im Blockdiagramm von Labview, welches die Eingänge aufnimmt , diese verarbeitet und im Ausgang wieder ausgibt:

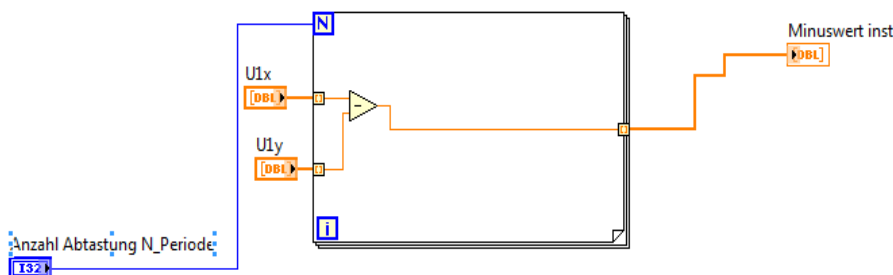
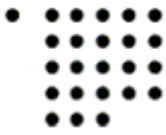


Abbildung 5.8: Minuswert\_Betrag

Die FOR-Schleife läuft von 0 bis N-1 dabei werden aus der Differenz aller instantanen „U1x“ und „U1y“-Werte, die instantane Differenzspannung berechnet und dem „Minuswert inst“ instanan wiedergegeben.

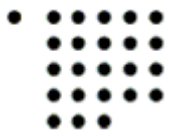


## **5.2.2 Erläuterung zum Quellcode der Benutzeroberfläche**

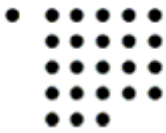
### **5.2.2.1 Vorwort**

Wie schon in Kapitel 4.7.3.1 erläutert sind die Daten alle dynamisch und haben alle eine Kanalzuweisung um die Kanäle zu erfassen. Auf Grund des zu großen Programms ist es unmöglich das ganze Programm in die Arbeit als Bild mit anzuhängen. Deshalb wird jetzt nur auf die wesentlichen und wichtige Programmteile eingegangen. Die VI wird digital mit einem Stick/CD im Anhang eingefügt. Es hat sich empfohlen die Arbeit nach dem Programmgrundgerüst wie in Kapitel 4.7 alles abzuarbeiten.

### **5.2.2.2 Inputs und Outputs der Benutzeroberfläche**



Bezeichnung im VI	Art des Gates	Datentyp/ Anzeigeformat	Typ	Anzahl	Beschreibung
Schreiberkurven(S1-S14 für Kenndaten in Kapitel 4.7.4.2 ohne zeitabhängige Größen)	Output		XY-Graph	14	Schreibt den aufgenommenen RMS-Wert für jeden zeitlichen Messintervall
S1 Plot – S16 Plot	Input		Cluster aus 4 booleschen Variablen	16	Zeigt an welche Plots in der Schreiberkurve aktiv/inaktiv sind
Oszilloskopkurven(Osz1 – Osz9 für Momentanwerte in Kapitel 4.7.3.1 und 4.7.4.1 )	Output		XY-Graph	9	Oszilloskopbilder
Osz 1 Plot – Osz 2 Plot	Input		Cluster aus 4 booleschen Variablen	9	Zeigt an welche Plots in den Oszilloskopbildern aktiv/inaktiv sind
Zeigerinstrumente(für Kenndaten in Kapitel 4.7.4.2 )	Output	Doppelte Genauigkeit	Drehspulinstrument	20	Zeigt die Kenndaten im Zeigerinstrument mit seinem Anzeigeelement an
LED-Anzeigen(U_dif1-U_dif3)	Output	Long	Rundes Farbfeld	3	Stellt die Dunkelschaltung dar
U_dif1-U_dif3	Output	Doppelte Genauigkeit	Numerisches Anzeigeelement	3	Stellt die Zahlendarstellung für die Dunkelschaltung dar
LED1-LED3(Skalierung)	Input	Doppelte Genauigkeit	Numerisches Bedienelement	3	Skaliert die max. Zahl auf 100%
Phasendifferenz(U_dif1-U_dif3)	Output	Doppelte Genauigkeit	Numerisches Anzeigeelement	3	Zeigt die Phasenverschiebung zwischen Asynchronmotor und Synchrongenerator an
Kanalumschalter (für Kapitel 4.7.3.1 von AI16-AI23 zu AI24-AI31)	Input	Boolesch	Umschalter Horizontal	8	Schaltet auf Ersatzkanäle um
Kanalwähler AI(für Kapitel 4.7.3.1 von AI0-AI11 zu AI12-AI15 auswählen)	Input	Doppelte Genauigkeit	Numerisches Bedienelement	12	Schaltet um, auf einen der 3 freien Kanäle
Dynamische Spezifikation(Kapitel 4.7.3.2 )	Input	Doppelte Genauigkeit	Numerisches Bedienelement	2	Messpunkte pro Messintervall und Smpelrate Rs
Dynamische Spezifikation(Kapitel 4.7.3.2 )	Output	Doppelte Genauigkeit	Numerisches Anzeigeelement	1	Messintervall Tmess
Instantane Latenz- und Abtastzeit	Output	Doppelte Genauigkeit	Numerisches Anzeigeelement	2	Latenzzeit(Zeigt die Momentane Zeit die gebraucht wurde um den darauffolgenden Messintervall zu lesen)



Speicherung "Messintervall" in Sekunde	Input	Doppelte Genauigkeit	Numerisches Bedienelement	1	Eingabe für das Speichern des Messintervalles
Name/Beschriftung/Pfad (Datei)	Input	String	String Bedienelement	4	Eingabe für den Dateinamen bzw. Dateibeschriftung und Pfad
Name/Beschriftung/Pfad (Datei)	Output	String	String Anzeigeelement	2	Anzeige für den Dateinamen bzw. Dateibeschriftung und Pfad
Taster	Input	Boolesch	Schaltfläche	9	Stellen die Taster im Kapitel 4.7.6.1 dar
LED-Anzeigen	Output	Boolesch	LED	3	Stellen die Taster im Kapitel 4.7.6.1 dar
Name/Beschriftung (Metadaten)	Input	String	String Bedienelement	4	Stellen die Eingabe-Strings im Kapitel 4.7.6.1 dar
Name/Beschriftung (Metadaten)	Output	String	String Anzeigeelement	4	Stellen die Ausgabe-Strings im Kapitel 4.7.6.1 dar

### 5.2.2.3 Das Messprogramm(Hauptprogramm)

Das Programm besteht aus zwei Whileschleifen(eine Innere und eine in der Äußeren ), die äußere Whileschleife beinhaltet die Zeitbearbeitung der Oszilloskop-Aufzeichnung, weiterhin wird hierüber das Programmende gesteuert.Außerdem befinden sich hierbei Schieberegister die für das Erhalten der Kenndaten(Kapitel 4.7.4.2 ) zuständig sind. Der Rest befindet sich in der inneren Whileschleife . Die Whileschleifen sorgen dafür, dass das Programm nicht abbricht. Auf die Abbruchbedienung wird noch eingegangen.

#### 5.2.2.3.1 Die Messsignale

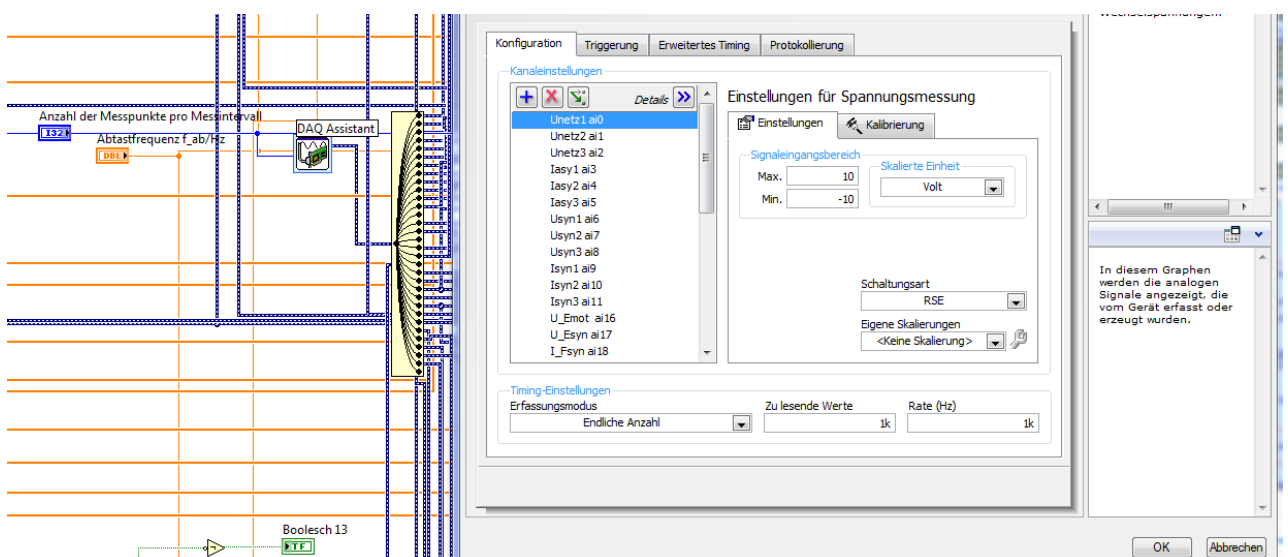
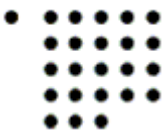


Abbildung 5.9: Signalaufnahme



Wie man in Abb. 5.9 sehen kann, ist der DAQ- Assistent einstellbar. In dem Reiter „Konfiguration“, kann man über den blauen Kreuz Button virtuelle Kanäle erstellen mit dem dazugehörigen physikalischen Kanal. Die Kanäle kann man auch bezeichnen, um die Übersicht der Kanäle beizubehalten wurde nach Kapitel 4.7.3.1 die Kanalbeschriftung mit dem Messsignal angegeben. Diese wurden von oben nach unten erstellt und abgearbeitet. Die Spannungsbandbreite(hier max. -10V bis 10V) ist veränderlich, empfehlenswert ist zu nächst einmal die maximale Bandbreite auszunutzen, damit der Hardware- Bachelorarbeiter die maximale Bandbreite zur Verfügung hat. Die Skalierte Einheit ist in Volt darzustellen, da der Stromwert ja auch in Volt angezeigt wird. Die Schaltungsart wird in RSE(ein gemeinsamer Bezugspunkt für gegen Masse ) betrieben, damit alle Analog Input(AI)- Eingänge ausgenutzt werden können. Weiterhin wurden Ersatzkanäle erstellt, auf die im Nachhinein eine bessere Erklärung folgen. Die Abtastfrequenz und Anzahl der Messpunkte pro Messintervall werden extern eingestellt(links oben Abb. 5.9 ). Abtastrate und Abtastzahl werden eingestellt über die Variable „Abtastfrequenz f<sub>ab</sub>/Hz“(als DBL) und „Anzahl der Messpunkte pro Messintervall“(als Long). Als nächstes wird dass Signal aus dem DAQ- Assistent entnommen und die Signale getrennt, damit die Verarbeitung jeder einzelnen Signale möglich sind. Die Express-VI mit der getrennt wird heißt „Signal Trennen“-VI, dieser trennt das kombinierte Signal.

### 5.2.2.3.2 Messpunkt- und Messintervallberechnung

#### 5.2.2.3.2.1 Leistungen

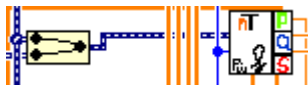


Abbildung 5.10:

Messpunkte\_berechnen

Jeder Phasenspannungsmesssignal wird kombiniert mit der dazugehörigen Phasenstrommesssignal(bsp.  $U_{netz1}$  mit  $I_{asyl}$  ). Die Kombination ist über das Express-VI (Funktion)„Signale zusammenfassen“ möglich. Anschließend wird aus dem kombinierten Signal mit der Anzahl der Abtastungen, über das Sub-VI zur „Bestimmung der Leistungen und Anzahl der Perioden“ (siehe Kapitel 5.2.1.1 für die Sub-VI) instantane

Leistung, Leistung, Blindleistung und Scheinleistung berechnet. Hierbei handelt es sich um die Leistungswerte in Kapitel 4.7.4.1(Berechnung für jeden Messpunkt) und 4.7.4.2 „Kennwerte“. Die Frequenz wurde nur beim Asynchronmotor(Netzfrequenz) und Synchrongenerator entnommen. Für die Gleichstromanteile macht die Frequenzaufnahme und Schein-/Blindleistung, auf Grund der Frequenzunabhängigkeit, kein Sinn. Alle anderen Ausgänge beim dem Sub-VI sind zu nächst zu vernachlässigen.

### 5.2.2.3.2 Effektivwerte

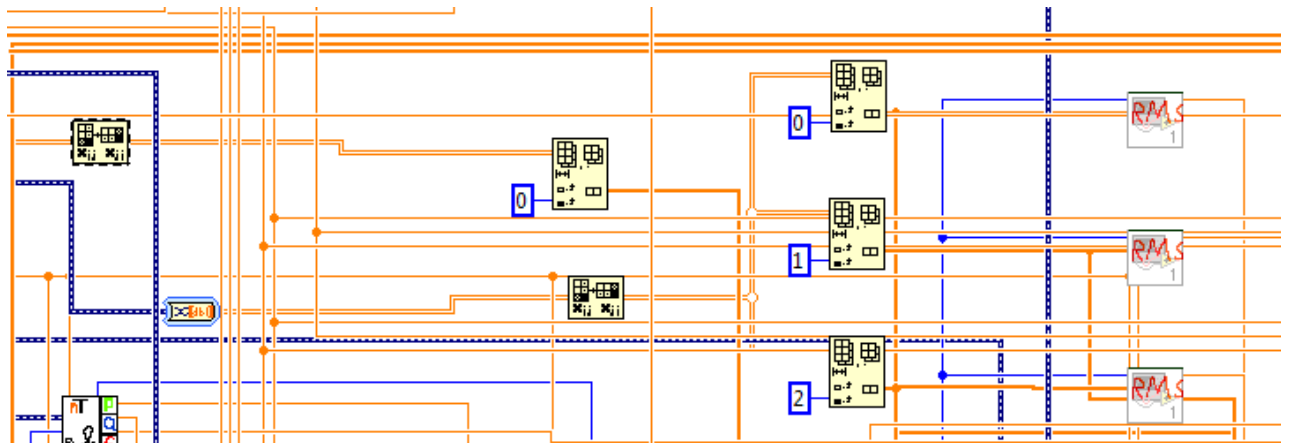


Abbildung 5.11: Effektivwerte

Wie man in Abb. 5.11 sehen kann, wurden alle drei Netzspannungssignale kombiniert, in DBL konvertiert (links in Abb. 5.11), transponiert und nach Spalte(0,1,2) mit der Funktion „Array löschen“ gefiltert. Die Kombination ist hierbei nicht notwendig, wie man es bei dem Gleichstromsignal (Abb. 5.11 links oben mit der Transponierenfunktion) mit nur einem „Array löschen“ und „2D-Array transponieren“ Funktion sieht. Sie wurde nur gemacht um die Bereiche wie Netzspannung oder Synchronstrom als kombiniertes Signal zuzuordnen. Wichtig ist, dass von allen Strom- und Spannungssignalen die Effektivwerte über das Sub-VI „RMS“ (siehe Kapitel 5.2.1.2.1) wie in Abb. 5.11 (ganz rechts) berechnet wird.

### 5.2.2.3.3 Phasendifferenz berechnen

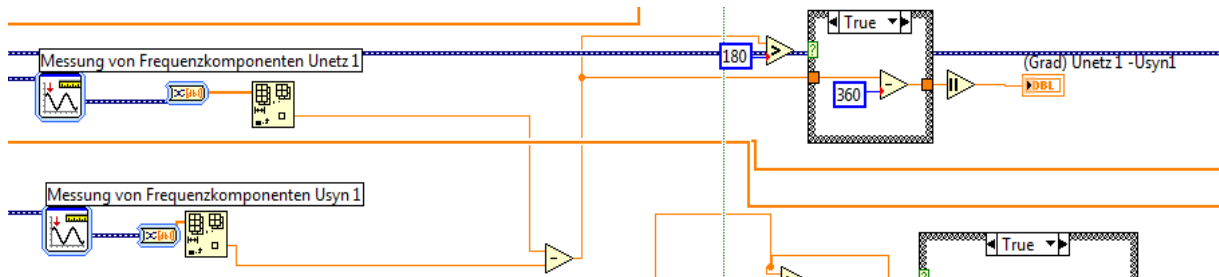
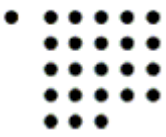


Abbildung 5.12: Phasendif\_berechnen

Um die Phasen für alle 3  $U_{netz}$  - und  $U_{synx}$  - Signale zu berechnen, wurde das Express-VI „Messung von Frequenzkomponenten“ verwendet (links in Abb. 5.12). Um aus dem dynamischen Signal einen Array zu erhalten wurde die VI „Von dynamischen Daten konvertieren“ verwendet. Um das Element zu filtern, wurde die VI „Aus Array löschen“ verwendet (Das VI hat als Standard für den Eingang Spalte 0). Aus diesen Werten wird die Differenz berechnet. Ist der Differenzwert kleiner als 180 Grad, so bekommt die Case-Struktur den Wert „False“ und das Signal kann die Case-Struktur sorglos passieren. Ist der Differenzwert größer als 180 Grad, so bekommt die Case-Struktur den Wert „True“ und der Wert wird um 360 Grad dekrementiert (da die Definitionsmenge einer harmonischen Schwingung in der Regel  $-180^\circ < \varphi \leq 180^\circ$  hat). Trotz dennoch wäre die Rechnung im Programm falsch weil der Wert „-180 Grad“ hier existieren würde. Aber dadurch dass nur die Phasendifferenz, die man auch als Betrag darstellen kann, wichtig ist wird der Fehler durch den Betrag aufgehoben. Anschließend werden alle Phasendifferenz in DBL als Anzeige dargestellt.



#### 5.2.2.3.2.4 Differenzspannung



Der Effektivwert der Differenzspannung wurde in Abb. 5.13 folgenderweise bestimmt:

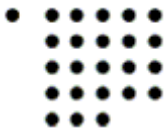
Abbildung 5.13:  
*Diff\_eff*

Um den Effektivwert der Differenzspannung (Kapitel 4.7.4.2, Tabelle 4.4, siehe „Spannung zwischen Netz und Synchrongenerator“) zu ermitteln wurde Signal2 von Signal1 subtrahiert (bsp.  $U_{netz1}(t) - U_{syn1}(t)$ ). Das

subtrahierte Signal ist dann in ein DBL-Wert über die Funktion „Von dynamischen Daten konvertieren“ umzuwandeln. Danach wird der DBL-Wert in die Sub-VI als Eingang „Instantane Wert“ eingeführt, weiterhin wird der Eingang „Anzahl der Abtastung pro Periode oder N“ der Sub-VI mit „Anzahl der Abtastung pro N\_Perioden“ verbunden. Als Ausgang erhält man dann den Effektivwert der Differenzspannung, der in eine DBL-Anzeigeelement auszugeben ist. Für die instantanen Differenzspannungen, werden die U<sub>synx</sub> und U<sub>asyx</sub> Arrays (z.b. U<sub>syn1</sub> und U<sub>netz1</sub>) in Kapitel 5.2.2.3.2.2 vor dem „RMS“-Sub-VI als Verbindung entnommen und in die Sub-VI „Minuswert(Instantan)“ (Kapitel 5.2.1.3) eingeführt. Daraus wird dann, wie schon in Kapitel 5.2.1.3 erläutert, die instantanen Differenzspannungen bestimmt.

#### 5.2.2.4 Speichern der Daten

Eine klare (Tabellen-)Struktur ist ein wichtiger Bestandteil für die Speicherung von Daten. Einige Express-VI's haben die Möglichkeit Daten zu speichern. Trotz dennoch sind Express VI's schwer, meist komplex zu verstehen und für die Datensicherung ungeeignet. Deshalb wurde die eigene (Tabellen-)Struktur quasi „gebastelt“.



5.2.2.4.1 Metadaten(Oszilloskop-Bilder)

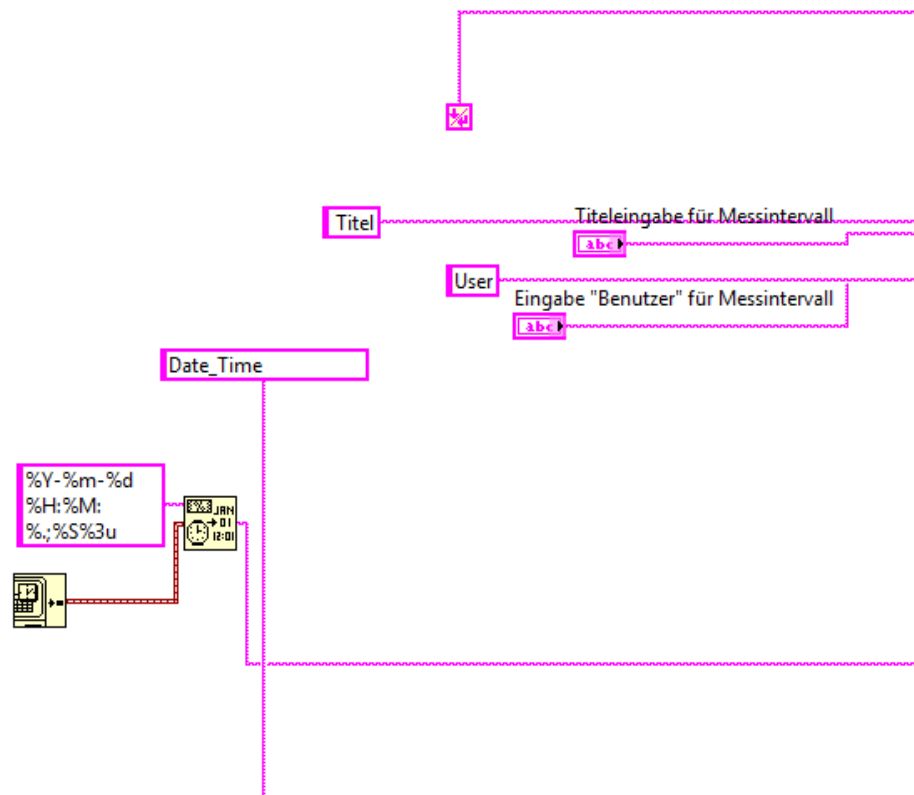


Abbildung 5.14: Metadaten

Die Tabs wurden ganz trivial bei den Stringkonstanten ganz hinten ggf. vorne bei den Parameterüberschriften angehängt(Abb.5.14). Ganz oben befindet sich ein Symbol in einem pinken Kasten mit [ ↓ / ← ] dies ist ein Zeilenvorschub. Nun kann man die gewünschten Metadaten basteln. Zu nächst folgt die String- Konstante „ Titel“(zweites Element von oben nach unten Abb.5.14), anschließend die „Titeleingabe für Messdaten“(drittes Element von oben nach unten Abb.5.14) usw. bis zu dem letzten Parameter in Kapitel (siehe Musterbeispiel in Kapitel 4.7.5.1.3). Die Parameter werden entweder über lokale

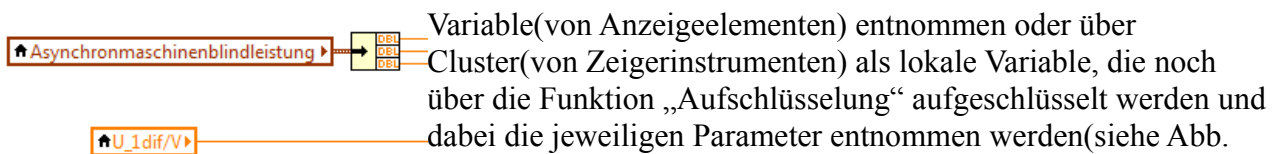
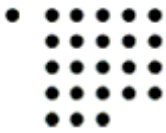


Abbildung 5.15: Parameter\_Osz 5.15).

Für die Zeit wurde die Funktion „Datum/Zeit in Sekunden ermitteln“ verwendet, dieser bestimmt die aktuelle Systemzeit(in Abb. 5.14 Links, das Rechteck mit der Uhrzeit). Um diesen in ein String zu formatieren, wurde „Datum/Zeit formatieren“ als Funktion genommen. Um das Zeitstempel auf die Zeitformatierung wie in Kapitel 4.7.5.1 zu verwirklichen, wurde in der Funktion-VI „Datum/Zeit formatieren“ „-“ für die Trennung, „%Y“ für das Jahr, %m Monat und %d für den Tag angesetzt. Für die Uhrzeit wurde ein Tab hinzugefügt und wie üblich als Trennzeichen genommen, hierbei „%H“ als Stunde, „%M“ als Minute, „%.;“ damit die Kommastelle eine Punkt







#### 5.2.2.4.2 Metadaten (Kenndaten\Schreiberkurve)



Da hier keine Parameter in Metadaten sich befinden sollen, sind diese wegzulassen, bis auf die Grunddaten, die auch wie in Abb. 5.14(Titel,User und Date\_Time) dargestellt wurden. Die Metadaten werden aber nicht mit der VI-Funktion „In String formatieren“ sondern mit der VI-Funktion „Strings verknüpfen“ verknüpft. Dieser fungiert ebenfalls wie die VI-Funktion „In String formatieren“, bloß mit dem Unterschied, dass dieser keine Zahlenwerte (boolesch etc.) annimmt, sondern nur Strings. Die nachfolgenden Eingänge enthalten die Spaltenüberschrift des Dateninhaltes, auf die gleich in den darauffolgenden Kapiteln eingegangen wird. Weiterhin wurden hierbei Tabs [ → ] verwendet. Die in Pink dargestellt ist (→). Auch hier spielt es keine Rolle wie man etwas „zusammenbastelt“. Wichtig ist dass die Kriterien für das Musterbeispiel ( in Kapitel 4.7.5.1.3), erfüllt werden.

Abbildung  
5.17: I\_S\_F

#### 5.2.2.4.3 Dateninhalt (Oszilloskop-Bilder)

Wie schon in 5.2.2.4.1 erklärt enthält der letzte String der Funktion-VI „In String formatieren“, die Zeit/s, die Werte der Messpunkte und berechneten Messpunkte(Kapitel 4.7.4.1 und 4.7.3.1). Dies sieht dann so aus

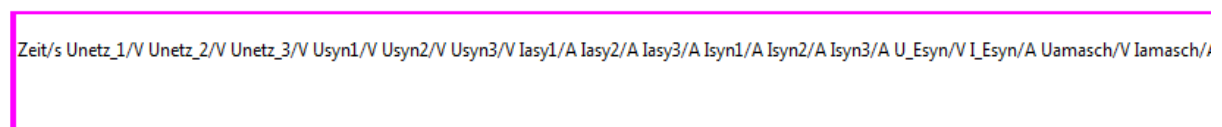


Abbildung 5.18: Überschrift\_Zeile[0]

Wie man in Abb. 5.18 sieht wurden die Spaltenüberschriften mit einem Tab getrennt. Nun haben wir die Metadaten und Spaltenüberschriften als Ausgangsstring. Leider sind die Kommata der Parameter(hier in lokalen Variablen) immer noch nicht weg. Deswegen werden diese über die Funktion-VI „String suchen und ersetzen“ gesucht und ersetzt mit einem Punkt. Die „.“-Stringkonstante ist hierbei verbunden mit dem Eingang „Such-String“ und „.“ als Stringkonstante ist verbunden mit dem Eingang „String ersetzen“, wie in Abb.5.19(links unten). Dabei werden wegen der „True“- Konstante ‚beim Eingang „Alle ersetzen“, alle Kommata ersetzt. Mit dem Ausgangsstring folgt die Einschreibung in eine Textdatei, die mit der VI-Funktion „In Textdatei schreiben“ über den Eingang „Text“ verwirklicht wird(Abb.5.19, ganz links in der Mitte). Der Funktion wird auch ein Pfad zugewiesen, als „Datei“- Eingang, auf die in den nächsten Kapiteln eingegangen wird.Von der Funktion wird die Textdatei als „Referenz“-Ausgang der „In Textdatei schreiben“- VI als Pfad konvertiert und in die „In Tabellenkalkulation schreiben“ als „Dateipfad“-Eingang eingeführt(Abb.5.19, links oben, der Block mit dem Stift und der Tabelle), damit dieser als Tabellenkalkulation anerkannt wird. Die VI bekommt zur Sicherheit beim Eingang

„Datei anhängen“ den Wert „True“ damit nichts überschrieben wird. Weiter rechts in Abb.5.19 befindet sich eine weitere „In Tabellenkalkulation schreiben“ VI dieser bekommt das 2D-Array(auf die noch eingegangen wird) als „2D-Daten“-Eingang, welches die Messpunkte(Kapitel 4.7.4.1 und 4.7.3.1) beinhaltet. Dieses wird durch das „True“ transponiert und angehängt zu dem Pfad wo sich die Metadaten mit der Spaltenüberschrift sich befindet. Wichtig ist hierbei, dass das „Format“ als Eingang den Wert „%;%.6f“ als Stringkonstante bekommt, dieser bedeutet, dass das Komma im Array mit mit einem Punkt ersetzt werden soll und es 6 Nachkommastellen „;%.6f“ als Float existieren soll. Der Pfad wird als Ausgabe noch gebraucht. Das ganze befindet sich auch im Case von Abb.5.16.

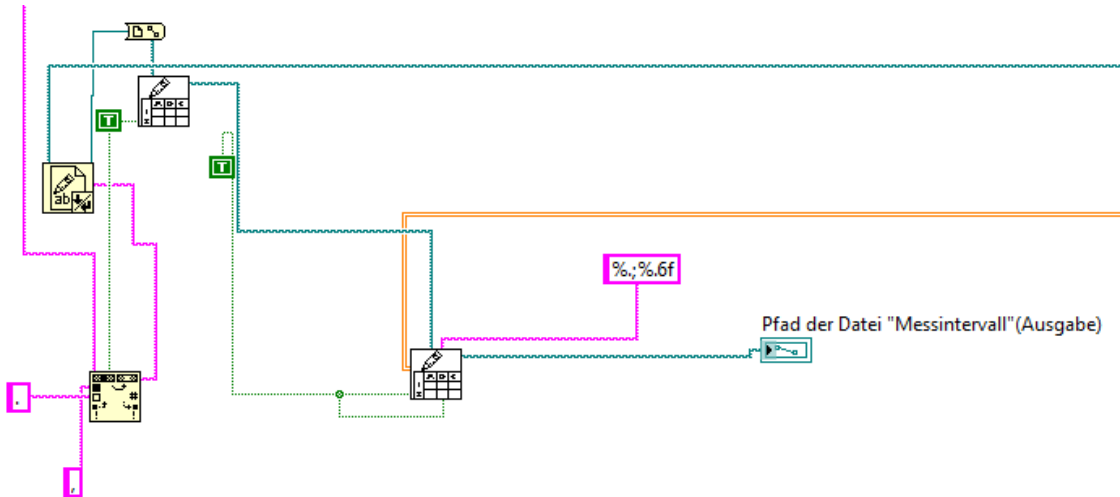


Abbildung 5.19: Verarbeitung der Textdaten

Für die Verarbeitung des Arrays wird, wie in Abb. 5.11 zu sehen ist, die Arrays der jeweiligen Messpunkte genommen, die durch das Auslösen der VI-Funktion „Array löschen“ erstellt wurde. Die Zeit wird in der äußeren Whileschleife verarbeitet und in die Innere eingeführt. Dieses sieht wie folgt aus:

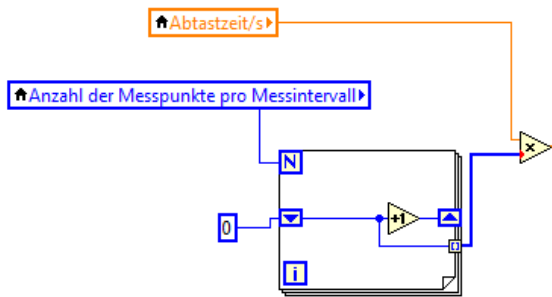
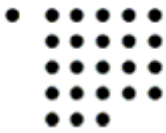


Abbildung 5.20: Zeitverarbeitung\_Osz

Über den Wert der lokalen Variable „Anzahl der Messpunkte pro Messintervall“ läuft eine FOR-Schleife durch(Abb.5.20). Zu jedem folgenden Element wird die letzte Zahl einer 1 dazu addiert. So dass die 0te Stelle den Wert 0, die 1te den Werte 1, die 2te Stelle den Wert 2 .. usw. hat. Am Ende werden alle Elemente des Arrays mit der Abtastzeit multipliziert. So dass jeder Punkt für die Elemente aus den Array-Messsignalen die entsprechende Zeit zugeordnet bekommt.



Alle Messpunkte(Kapitel 4.7.4.1 und 4.7.3.1) als Array-Messsignale(Abb. 5.11 ) mit dem Zeitarray werden über die Funktion „Array erstellen“(in Abb.5.21 ) verknüpft. In Abb.5.21 muss man jetzt auf die Spaltenüberschriften achten. Die erste Spaltenüberschrift ist Zeit/s, also bekommt die VI-Funktion „Array erstellen“ als ersten Eingang das Zeitarray, die nächste wäre U10/V also bekommt der nächste Eingang die erste Netzspannung usw. . Nach dem alle Eingänge verknüpft wurden, erstellt die VI ein Array aus den Messpunkten(Abb.5.21 ), dieses Array wir nun über den Case in Abb.5.16 und somit in Abb.5.19 in den rechten VI „In Tabellenkalkulation schreiben“ als „2D-Daten“-Eingang eingeführt, hier transponiert mit der „True“-Konstante am Eingang „Transponieren?“ und zu den Metadaten mit den Spaltenüberschriften angehängt. Nun erhaltet man die gewünschte Tabelle. Der Ausgangspfad zeigt nochmal an in welchem Pfad sich die Datei befindet. Auf den Dateinamen wird noch eingegangen.

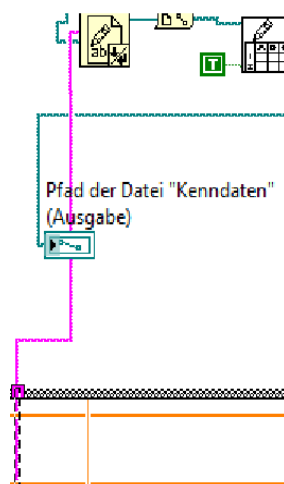
ung

5.21:

Array\_

Osz

#### 5.2.2.4.4 Dateninhalt (Schreiberkuve/Kenndaten)



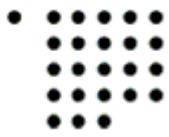
In Abb. 5.17 werden nun nach den Metadaten (die ersten Verbindungen), die Kenndaten(Kapitel 4.7.4.2) als Spaltenüberschrift eingeführt. Wie in Kapitel 5.2.2.4.3 werden die Metadaten und Spaltenüberschriften als verknüpften String über in Case (auf die noch eingegangen wird) in die Funktion-VI „In Textdatei schreiben VI“ als „Text“-Eingang eingeführt mit dem Pfad als „Datei“-Eingang(auf diesen Nachfolgend eingegangen wird). Die Datei wird als Pfad konvertiert und in die VI „In Tabellenkalkulation schreiben“ (Eingang: Dateipfad) eingeführt, transponiert durch die „True“-Konstante und als Tabellenkalkulation verarbeitet.

Abbildung 5.22:

Kenndaten\_Speicheru

ng

Für den Kennwert „Zeit/s“ ist ein besonderes Verfahren durchzuführen. Eine Zeit-VI „Datum/Zeit in Sekunden ermitteln“ wird hinter der äußeren Whileschleife gestartet(Abb.5.23). Eingeführt in die äußere Whileschleife wird dieser in ein Zahlenwert umgewandelt und nochmal in die innere Whileschleife eingeführt. Über die „Auswählen“- Funktion-VI wird abgefragt(Abb.5.23, Rechts). In der inneren Whileschleife befindet sich eine weitere Zeit-VI(Abb.5.23, Rechts 2, die Rechte von beiden), die in eine Zahl(DBL) konvertiert wird. Hat „Rücksetzen der Kennwerte“(als Taster-Schaltenelement) den Wert „False“(Taste nicht gedrückt), so hat die Verbindung zum VI „Inkrementieren(Punkt für Punkt)“-Eingang(Abb.5.23, ganz rechts, der Linke von Beiden) als „Aktivieren“- Eingang den Wert „False“. Somit bleibt beim „Inkrementieren(Punkt für Punkt)“-Ausgang „Zahl“ der Wert bei Null. Ist die „Zahl“ gleich Null(welches dies gerade entspricht) so bekommt die „Auswählen“- Funktion-VI den Wert „False“(da die Abfrage: Ist der Wert ungleich 0 lautet). Dabei wird der Ausgang von der Zeit-VI, außerhalb der Whileschleifen(Abb.5.23, ganz Links), übernommen. Der Ausgang der „Auswählen“- Funktion-VI wird von der Zeit-VI subtrahiert (von der ganz rechten Zeit-VI in Abb. der inneren Whileschleife), als Endergebnis erhält man die



Messzeit der Schreiberkurve. Die Zeit-VI fängt von [01-01-1904 00:00:00] als Zeit in Sekunden an zu zählen, beim Datum [01-01-2013 00:00:00] hätte man den DBL-Wert 3352665600. Bekommt aber „Rücksetzen der Kennwerte“ den Wert „True“(Taster gedrückt), erhöht sich die „Zahl“ als „Inkrementieren(Punkt für Punkt)“- Ausgang um eins, somit wird der „True“-Teil beim „Auswählen“- Funktion-VI übernommen. Der „True“-Teil erfolgt über eine weitere Zeit-VI(Abb., Rechts 2, die Linke von beiden) und einem leerem Array verknüpft in der Funktion-VI „Array erstellen“ welches in einem Case(Abb.5.23,Rechts) stattfindet, um in das leere Array(verknüpft als Konstante auch in der Funktion-VI „Array erstellen“) Elemente hinzuzufügen. Allgemein wird das 1D-Array außerhalb der äußeren Whileschleife als Konstante mit einem leeren 1D-Array initialisiert (Abb.5.23, links). Dieses wird an der äußeren als auch inneren Whileschleife als Schieberegister verknüpft. Da das Array die Aufgabe hat bei jedem Durchlauf der inneren Whileschleife eine neue Messzeit der Schreiberkurve als Element zum leeren Array, welches nun befüllt wird, hinzuzufügen. Einer weiteren „Inkrementieren(Punkt für Punkt)“-VI(Abb.5.23, rechts , der Rechte von beiden) wurde eine „True“ - Konstante an den „Initialisierung“-Eingang zugewiesen. Das bedeutet das VI initialisiert ‚bei einem „True“ - Wert im Eingang „Aktivieren“, eine 1 für „Zahl“-Ausgang(es finden keine weiteren Inkremente statt, im nächsten Durchlauf der inneren Whileschleife hat „Zahl“ den Wert Null bei „True“ wieder die Eins). Hat die „Zahl“ den Wert 1 ist dieser nicht gleich Null, somit bekommt sowohl das Endergebnis als auch die Case-Struktur den Wert „False“. Dabei wird ein neues Array erzeugt mit dem linken Zeit-VI über die Funktion-VI „Array erstellen“ im Case der inneren Whileschleife. Von diesem Array wird immer das nullte Element aus dem Case entnommen über die „Array löschen“-VI entnommen, da dieser die Zeit darstellt nachdem ein Rücksetzen der Kennwerte stattgefunden hat. Beim nächsten Durchlauf wird nicht mehr initialisiert(wenn „Rücksetzen der Kennwerte“ den Wert „False“ hat, kein Tastendruck praktisch), somit hätte praktisch „Inkrementieren(Punkt für Punkt)“-VI(der Rechte von Beiden) als „Zahl“- Ausgang den Wert Null und der Case den Wert „True“.

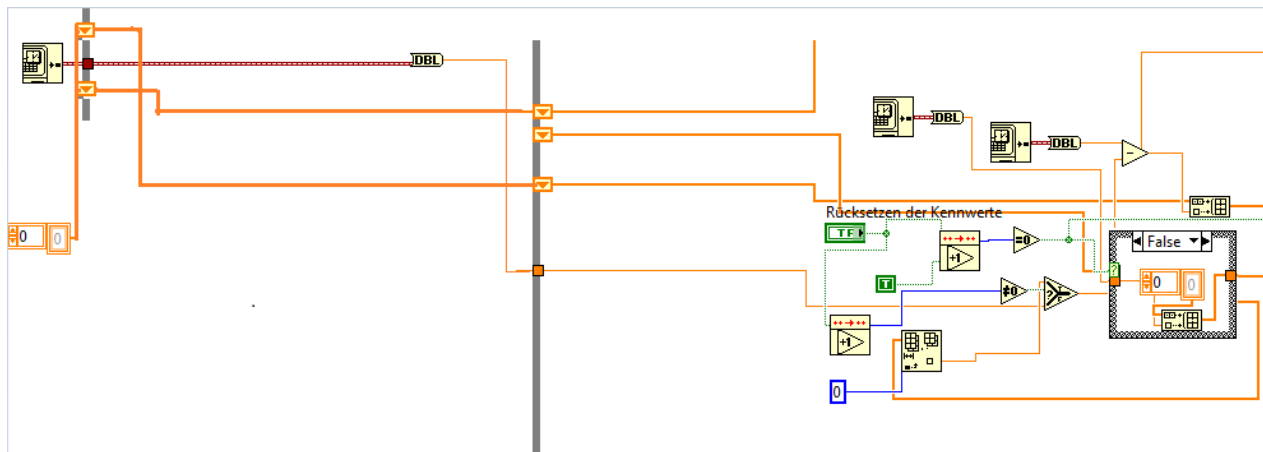


Abbildung 5.23: Zeit-Kenndaten

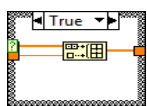
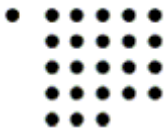


Abbildung 5.24: True

Bei „True“ (Abb.5.24) nimmt das Array die darauffolgenden Werte an . Wichtig ist hierbei, dass nur das nullte Element entnommen wird, weil nur die Differenz mit der Ursprungszeit gebildet werden darf. Das Endergebnis der Messzeit-Schreiberkurve(Abb.5.23,rechts oben Ausgang am Subtrahierblock) wird wiederum über einen Case in ein Array eingeführt(Abb. 5.25 Als „True“ und 5.26 als „False“).

Das boolesches Schaltverhalten wird über „Rücksetzen der Kennwerte“(in Abb.5.23) bestimmt , da dieser als Auswahl mit dem Case verknüpft ist. Hat „Rücksetzen der Kennwerte“ den Wert „False“(Taster nicht gedrückt) werden neue Elemente in das 1D-Array geschrieben. Das 1D-Array wird außerhalb der äußeren Whileschleife als Konstante mit einem



leeren 1D-Array initialisiert. Dieses wird an der äußeren als auch inneren Whileschleife als Schieberegister verknüpft. Das Array hat die Aufgabe bei jedem Durchlauf der inneren Whileschleife eine neue hreiberkurve als Element zum leeren Array, welches nun befüllt wird, hinzuzufügen. Hat „Rücksetzen der Kennwerte“ den Wert „True“(Taster gedrückt) wird das 1D-Array geleert, die neue Messzeit-Schreiberkurve wird reingeschrieben und beim nächsten Durchlauf(wenn kein weiteres mal der Taster „Rücksetzen der Kennwerte“ betätigt wird, was automatisch zum „False“-Fall führt) werden neue Elemente wieder in das 1D-Array geschrieben .

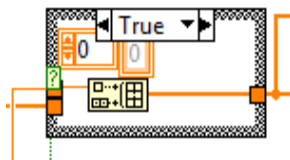


Abbildung 5.25:  
TRUE\_SK-Zeit

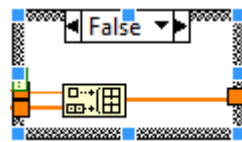


Abbildung 5.26:  
FALSE\_SK-Zeit

Alle restlichen Kennwerte aus Kapitel 4.7.4.2 haben das gleiche Case-Verfahren mit den Schieberegistern wie der Kennwert „Zeit/s“. Auch die Abfrage des Cases ist verbunden mit der booleschen Variable „Rücksetzen der Kennwerte“. Nun werden alle Kennwerte aus Kapitel 4.7.4.2 inklusive der Zeit in ein Array gespeichert(Abb.5.27). Dafür werden alle Kenndaten-Elemente in Kapitel 5.2.2.3.2.2 entnommen . Weiterhin die Frequenzen und Leistungen in Kapitel 5.2.2.3.2.1 (nicht die instantane Leistung), die Phasendifferenz in Kapitel 5.2.2.3.2.3 als auch Differenzspannung in Kapitel 5.2.2.3.2.4 .

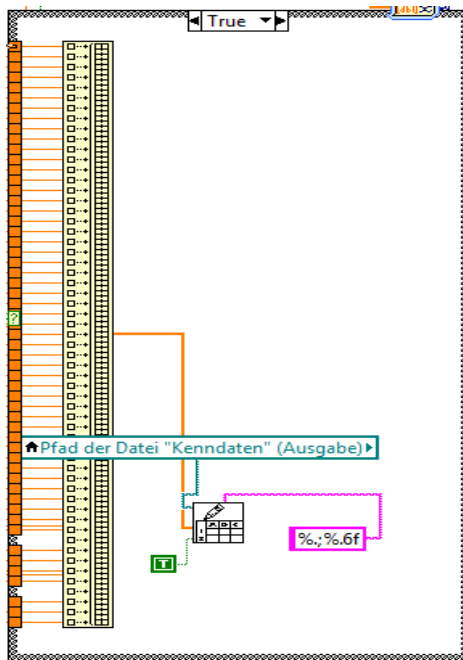


Abbildung 5.27:  
Speichern\_Kenndaten\_Dateninhalt

Das ganze sieht dann so aus wie in Abb. 5.27. Die Kenndaten werden so strukturiert, dass die Spaltenüberschrift mit dem Array übereinstimmt. Das angehängte Array wird nun transponiert über die Funktion-VI „In Textdatei schreiben VI“(dies passiert mit der „True“ - Konstante an Eingang „Datei anhängen?“) und abgespeichert in dem Pfad(auf die Nachfolgend eingegangen wird), in welcher sich die Metadaten mit den Spaltenüberschriften befinden. Natürlich werden die Kommata mit einem Punkt ersetzt und die Zahlen haben 6 Nachkommastellen(über den Format-Eingang).Auf den Case wird noch eingegangen.

#### 5.2.2.4.5 Dateinamen

Um die Namensauflösung der Datei in Kapitel 4.7.5.4 zu bestimmen wurde ein besonderer Prozess erstellt wie man in Abb.5.28 sehen kann. Der Prozess in Abb.5.28 findet in dem Case(in

Abb.5.16) statt. Rechts oben in Abb.5.28 sieht man die Funktion-VI „Strings verknüpfen“, mit dieser VI bastelt man nun den eigenen Namen für die Namensauflösung der Datei. Wie üblich folgt zu nächst einmal der Dateipfad als Stringeingabeaufforderung(Bedienelement), welches die erste Verbindung zum VI „String verknüpfen“ darstellt, um den Aufenthaltsort der Datei zu bestimmen. Als nächste Verbindung folgt die Zeit-VI „Datum/Zeit formatieren“(Abb.5.28 , links oben), die als erstes über „Datum/Zeit formatieren“ in der Form „YYYY-MM-TT“(Kapitel 4.7.5.4) mit der String- Konstante „%Y-%m-%d“ an Eingang „Zeitformatierungs-String“ formatiert wird und mit der VI-Funktion „In String formatieren“ als Textstring formatiert wird. Angehängt wird dazu die String- Konstant „\_“ als weitere Verbindung. Die lokale Variable „Anzahl der Messung“ „Messintervall““(ganz Links in der Mitte in Abb.5.28) wird über eine automatische Steuerung geregelt auf die noch in den nächsten Kapiteln spezifischer eingegangen wird. Bei jedem „True“(Tastendruck) oder jeder abgelaufene Zeit wird „Anzahl der Messung“ „Messintervall““ um eine 1 inkrementiert. Verbunden mit Eingang „Wert“ wird die Zahl anschließend 4-ziffrig als String dargestellt über die VI „Zahl nach String“ und durch den Eingang „Breite“ mit der Long- Konstante „4“. Nun würde die Zahl als Zeichenkette folgenderweise aussehen „ 1“ als „Dezimal-Integer-String“- Ausgang. Das Problem ist jetzt, dass die ersten 3 Ziffern keine Null haben sondern Leerzeichen. Das String „ 1“ wird verbunden mit dem „Eingangsstring“ über die Funktion-VI „String suchen und ersetzen“. Die Leerzeichen werden nun über die Funktion-VI „String suchen und ersetzen“ gesucht(durch das Anlegen der Zahl-String in den Such-String Eingang) und mit einer „0“ als String- Konstante am „String ersetzen“- Eingang ersetzt. Am Ausgang bekommt man als „Ergebnis-String“ den Wert „0001“, dieses ist dann die nächste Verbindung in der VI „String verknüpfen“. Die darauffolgende Verbindung ist eine „\_“ Zeichenkette als String- Konstante. Die nächste Verbindung zum VI „String verknüpfen“ ist die Bezeichnung der Messung, hier „SyncGenOsc“ als Oszilloskopbild- Aufzeichnung als String- Konstante. Nun folgt der interessante Teil der Verbindungen. Der Benutzer hat die Möglichkeit über das Bedienelement „Name der Datei“ „Messintervall““ einen Dateinamen anzugeben.

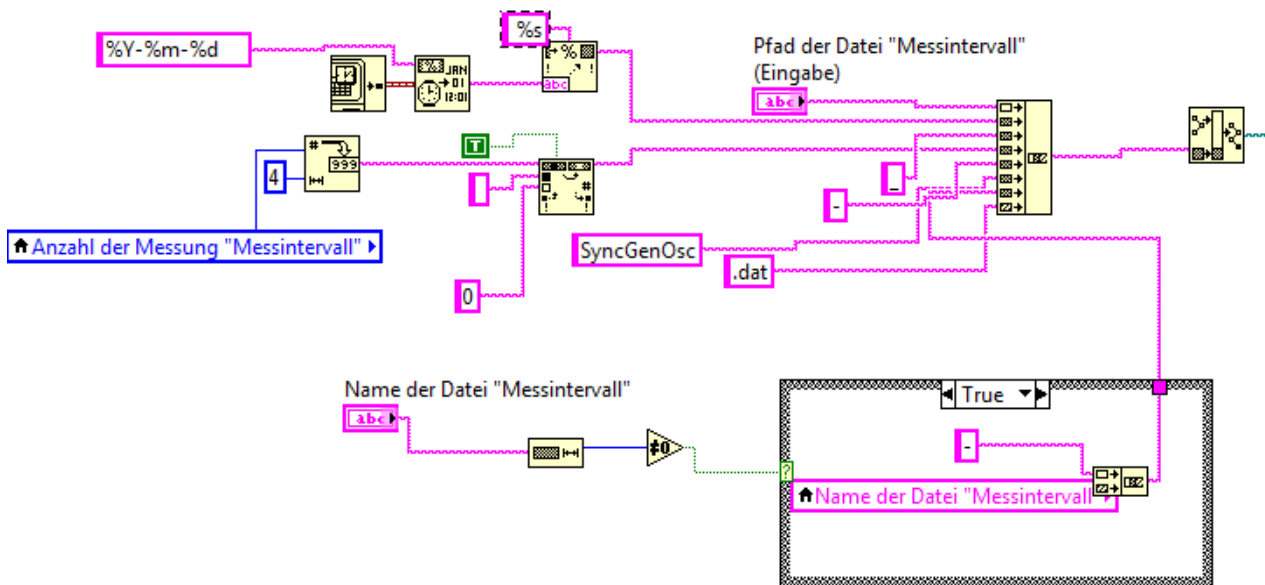
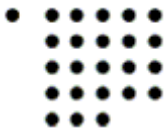
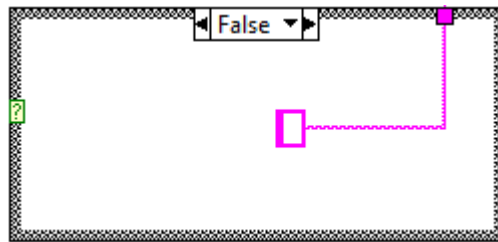


Abbildung 5.28: Dateinamen(Oszilloskop-Bilder)

Wird ein Dateinamen als „String“ angegeben so ist über die „String-VI“ der Wert der „Länge“ größer als Null und somit ungleich Null. Der Case hat den Wert „True“. In dem Case(Rechts unten, Abb.5.28) wird die String- Konstante „\_“ mit der lokalen Variable „Name der Datei



„Messintervall“ in einer „String verknüpfen“ VI verknüpft und aus dem Case verbunden mit der VI „String verknüpfen“(Rechts oben, Abb.5.28).



Hat „Name der Datei „Messintervall““ keinen Namen, so ist die String- „Länge“ Null und somit nicht ungleich 0. Der Case bekommt den Wert „False“, eine Stringkonstante ohne ein Wert wird ausgegeben und aus dem Case verbunden mit „String verknüpfen“(Rechts oben, Abb.5.28). Beendet wird das ganze mit der Verbindung „.dat“ als String- Konstante.

Abbildung 5.29: Case-False\_Dateiname

Das ganze wird als verknüpfter String an den Eingang „Pfad erstellen“ VI eingeführt und als Ausgang „hinzugefügter Pfad“ ausgegeben. Dieses wird in Abb.5.19 an den Eingang „Datei(Dialog verwenden)“ der VI „In Text schreiben“ angehängt. Sodass dieser Pfad, den Dateinamen mit dem dazugehörigen Pfad der Oszilloskop-Aufzeichnung bestimmt. Abb.5.28 ist die Speicherung für die Oszilloskop- Aufzeichnung, die Speicherung der Schreiberkurve hat eine identischen Prozess. Der Unterschied ist nur hierbei die Automatisierung und un bei der Kenndatenspeicherung gibt es eine weitere Funktion-VI „Inkrementieren(Punkt für Punkt)“ auf diese wird im folgenden Kapitel eingegangen(Abb.5.34, rechts).

#### 5.2.2.4.6 Vorwort Fenster

Nun wird die Programmierung nach Kapitel 4.7.6 erläutert, zunächst einmal bekommt die „Bedienung“ eine eigene Registerkarte die dann wie in Anhang 3(A3 Registerkarte, Abb.7.6 ) aussieht. Die „Bedienung“ hat eine eigene Registerkarte bekommen, da sie immer parat zum Programm sein soll. Für den Rest gilt die übliche Registerkarte mit den dazugehörigen Reitern. Dieses sieht dann so aus:

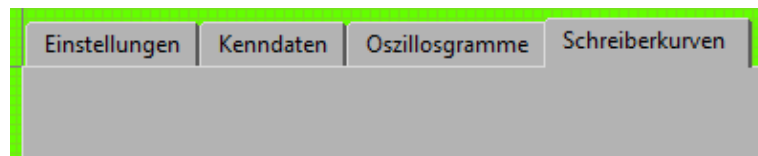


Abbildung 5.30: Registerkarte

Für die Automatisierung wurde ein besonderes VI verwendet welches sich als „SR-Flipflop“ ausgibt. Dieses ist eine Sub-VI der wie folgt funktioniert:

Set(Input/Bool)	Reset(Input/Bool)	Q(Output/Bool)	$\bar{Q}$ (Output/Bool)	Zustand
0	0	-	-	Speicherzustand
0	1	0	1	Rücksetzen
1	0	1	0	Setzen
1	1	x	x	Unerlaubter Zustand(oder gesetzt je nach Dominieren)

Tabelle 5.8: S-R\_flipflop

Dazu das Schaltbild(links) und die Funktion(rechts) im Sub-VI:



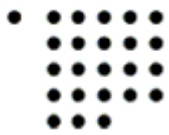


Abbildung 5.32:  
srff

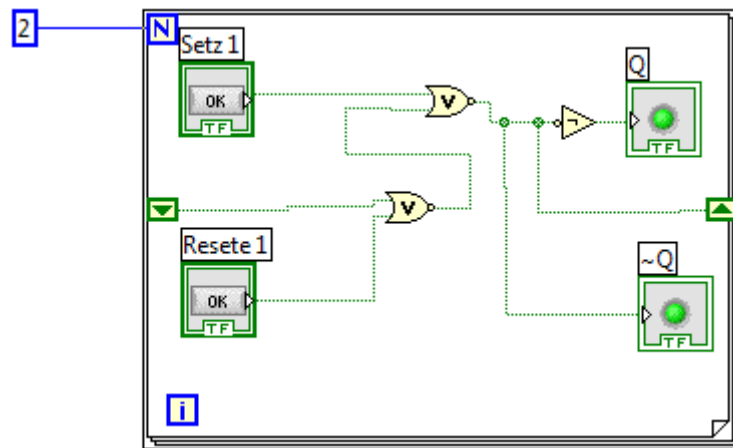
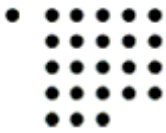


Abbildung 5.31: srff-Funktion

Den unerlaubten Zustand zu erreichen ist unmöglich. Da die Wahrscheinlichkeit sehr gering ist, dass der Benutzer die Setzen - und Rücksetzentaste in dem gleichen Moment drückt. Deshalb kann man diesen Zustand nahezu vernachlässigen. Ein weiterer wichtiger Zustand ist der Anfangszustand. Da diese S-Dominant ist, wird als Anfangszustand das Setzen auserwählt. Dies liegt an der Funktion in Abb.5.31. Startet man die VI, so hat der Schieberegister in der FOR-Schleife den Anfangswert „False“ (da kein Einfluß existiert) in dem 0ten Durchgang. Die Laufbedingung für die FOR-Schleife lautet „ $i < 2$ “. Das Signal wird mit der „Resete 1“-Taste über ein NOR(Nicht-Oder) – Gatter verknüpft. Ein NOR-Gatter vergleicht zwei boolesche Werte, sind beide Werte „False“ wird ein „True“ als Ausgang ausgegeben sonst immer „False“. In dem ersten Falle hat die „Resete 1“-Taste den Wert „False“, also gibt das NOR-Gatter ein „True“ aus. Nun wird dieser Ausgang wieder über ein NOR-Gatter verknüpft mit der „Setz 1“-Taste. Der Wert dieser Taste spielt keine Rolle, da das andere verknüpfte Signal den Wert „True“ hat wird automatisch ein „False“ am zweiten NOR-Gatter (Abb.5.31) ausgegeben. Negiert wird das Signal für den Ausgang „Q“ also erhält man „True“ für „Q“. „ $\bar{Q}$ “ wird nicht negiert also „False“, der Schieberegister bekommt den Wert „False“, so dass die gleiche Abfolge im nächsten Durchlauf passiert (Speicherzustand). Wird z.B. jetzt die „Resete 1“-Taste betätigt (bei einem ständig durchlaufenden Programm), so hat „Resete 1“ den Wert „True“ also wird im linken NOR-Gatter (Abb.5.31) der Wert „False“ ausgelöst, „Setz 1“ hat auch den Wert „False“ also hat der rechte Gatter den Wert „True“ somit ist „Q“ = „False“ und „ $\bar{Q}$ “ = „True“, der Schieberegister nimmt den Wert mit und die gleiche Abfolge passiert (mit dem Unterschied, dass „Resete 1“ den Wert „False“ und der Schieberegister „True“ hat). Wird jetzt in einem durchlaufenden Programm „Setz 1“ gesetzt und der Schieberegister hat den Wert „True“, so gibt der linke Gatter ein „False“ aus (Abb.5.31), der rechte hat auf Grund „Setz 1“ den Wert „False“ als Ausgang und „Q“ hat auf Grund der Negierung einen „True“- Wert der andere Ausgang „False“. Beim nächsten Durchgang ist „Resete 1“ = „False“ als auch der Schieberegister. Das heisst der linke Gatter hat ein „True“ als Ausgang und das gleiche wiederholt sich nochmal. Sonst funktioniert der SR-Flipflop als VI wie nach Tabelle 5.8. NI übernimmt keine Haftung für dieses VI. Da der Verfasser der VI, keiner vom NI-Unternehmen ist (für weitere Informationen siehe [13])



#### 5.2.2.4.7 Fenster Bedienung

##### 5.2.2.4.7.1 Laufendes Programm

Für den Programmabbruch wurde ein boolesches Schaltelement als Taster namens „Programm beenden“ und ein boolesches LED-Anzeigeelement mit dem Namen „Programm läuft“ erstellt (Abb. 5.33).

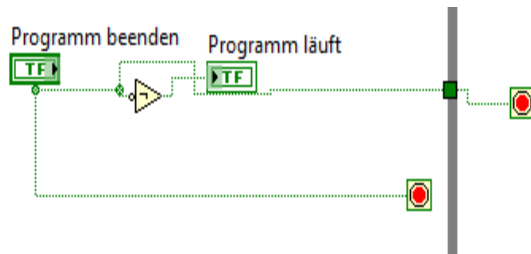


Abbildung 5.33: Zustand-Programm

Hat „Programm beenden“ den Wert „False“ (also nicht gedrückt). So wird der Wert negiert und als LED-Anzeigeelement erhält man den Wert „True“, sprich die LED leuchtet (Abb. 5.33). Hat „Programm beenden“ den Wert „True“ (also gedrückt). So wird der Wert negiert und als LED-Anzeigeelement erhält man den Wert „False“, sprich die LED leuchtet nicht, die innere Whileschleife wird beendet und der letzte Wert von „Programm beenden“ beendet somit

auch die äußere Whileschleife und somit das ganze Programm. Das ganze kann man dann auch im Frontpanel im Anhang sehen (Anhang 7.3 A3 Registerkartenfenster, Abb. 7.6, ganz oben im Bild mit der Anzeige und dem Schaltelement).

##### 5.2.2.4.7.2 Kenndatenspeicherung

Wie schon in Kapitel 5.2.2.4.5 und 5.2.2.4.4 erläutert, ist Abb. 5.34 die Automatisierung für den Case als auch der Kenndatenspeicherung. Die Kenndatenspeicherung erfolgt hingegen mit zwei Schaltelementen als Taster und einem LED-Anzeigeelement (Abb. 5.34, ganz links). Die Taster-Schaltelement sind „Aktivieren der Speicherung von Kenndaten“ verbunden am „Setz“-Eingang und „Deaktivieren der Speicherung von Kenndaten“ verbunden am „Rücksetz“-Eingang der Sub-VI „SR-Flipflop“. Hingegen ist das LED-Anzeigeelement die „Kenndatenspeicherung“, dieser zeigt an ob die Kenndatenspeicherung in Betrieb ist. Bekommt die Kenndatenspeicherung ein „True“ (Tastendruck) wird am Ausgang „Q“ im SR-Flipflop gesetzt. Die LED-Anzeige wird somit auch auf „True“ gesetzt und leuchtet. Der Ausgang „Q“ wird UND verknüpft mit dem booleschen Wert, welches über die Abfrage „Anzahl der Messung „Kenndaten““ gleich Null sich ergibt. Dieser ist anfangs „True“ weil in „Anzahl der Messung „Kenndaten““ nichts reingeschrieben wurde. Somit hätte der Ausgang der UND-Verknüpfung „True“. Jetzt kann die Kenndatenspeicherung in Abb. 5.28 und 5.27 als „True“-Wert im Case ausgeführt werden. Wie schon in Kapitel 5.2.2.4.5 den Unterschied zur Kenndatenspeicherung wird der „True“-Wert vom Case (Abb. 5.34, rechts) übernommen und in ein „Inkrementieren (Punkt für Punkt)“ als „Aktivieren“ Eingang eingeführt. Somit bekommt der Case in Abb. 5.27 den Wert „True“ und wird auch in der Art und Weise, wie in Kapitel 5.2.2.4.4 erklärt, ausgeführt mit den Dateinamen (wie in Kapitel 5.2.2.4.4).

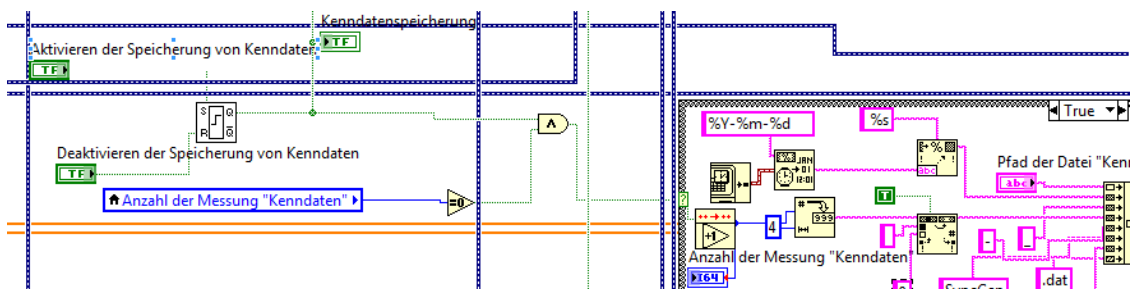


Abbildung 5.34: Kenndatenspeicherung

Wird aber nun „Deaktivieren der Speicherung von Kenndaten“ aktiviert, so hat „Q“ als Ausgang „False“, die LED-Anzeige wird zurückgesetzt. Der Case (Abb.5.34, rechts) hat den Wert „False“, dieser hat einen leeren Inhalt, sprich es findet keine Aktion statt (keine Namensauflösung für die Datei). Hingegen hat der Case in Abb. 5.27 auch ein „False“, es werden keine Werte mehr gespeichert, hingegen bekommt „Anzahl der Messung „Kenndaten““ als lokale Variable den Wert „0“ (Abb.5.35). Wird das Programm ganz am Anfang geöffnet und gestartet, so wird direkt gesetzt (Da wie schon in Kapitel 5.2.2.4.6 erklärt, der Anfangszustand „Setzen“ ist). Im Frontpanel befindet sich die Oberfläche dazu in Abb.7.6 von Anhang 7.3 A3 Registerkartenfenster, ganz oben im Bild in der zweiten Reihe (von oben nach unten gezählt) mit der Anzeige und den Schaltelementen.

Abbildung 5.35: False-Kennnd

### 5.2.2.4.7.3 Speichervorgang Oszilloskop-Aufzeichnung

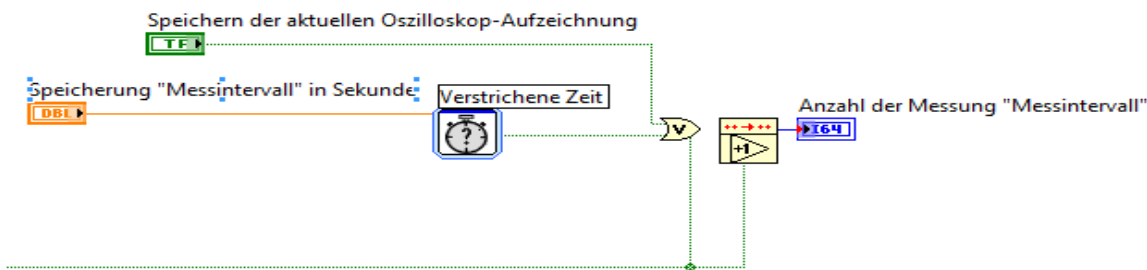


Abbildung 5.36: Speicherung\_Osz

Über die Schaltfläche „Speichern der aktuellen Oszilloskop-Aufzeichnung“ als Taster-Schaltelement und „Speicherung „Messintervall“ in Sekunden“ (im Express VI „Verstrichene Zeit“ als „Zielzeit(s)“ Abb. 5.36) wird die Speicherung der Oszilloskop-Aufzeichnung bestimmt. Nach dem die „Zielzeit(s)“ erreicht ist, wird am Ausgang „Zeit ist verstrichen“ dieses Express-VI's (Abb. 5.36) ein „True“, ausgegeben sonst immer „False“. Bei der Schaltfläche wird als Tastendruck ein „True“ am Taster ausgegeben, sonst immer „False“. Der Ausgang „Zeit ist verstrichen“ ist ODER verknüpft mit der Schaltfläche. Wenn am ODER-Ausgang ein „True“ entsteht (dieser entsteht nur dann nicht, wenn beide Verknüpfungen einen „False“ haben), bekommt die VI „Inkrementieren (Punkt für Punkt)“ am Eingang „Aktivieren“ den Wert „True“, wird der Zähler um 1 erhöht. Die Speicherung erfolgt dann im Case von Abb.5.16, in Abb. 5.37 nochmal dargestellt. In Abb. 5.37 kann man auch sehen, dass die Flache-Sequenzstruktur in dem Case verwendet wurde. Dieser macht einen Ablauf nach Reihe. Zu nächst einmal folgt in der Struktur ein „True“ an dem LED-Anzeigeelement „Speichervorgang Oszilloskop-Aufzeichnung“, über die Funktion-VI „Warten(ms)“ dauert die erste Sequenz um 1 ms an, das heißt die LED leuchtet für 1 ms um den User zu benachrichtigen, dass die Speicherung ausgeführt wird. Anschließend folgt mit der nächsten Sequenz das Rücksetzen der LED über die „False“-Konstante (Abb.5.37). Die „False“-Konstante ist verbunden mit dem Eigenschaftsknoten „Wert“ von „Speichervorgang Oszilloskop-Aufzeichnung“.

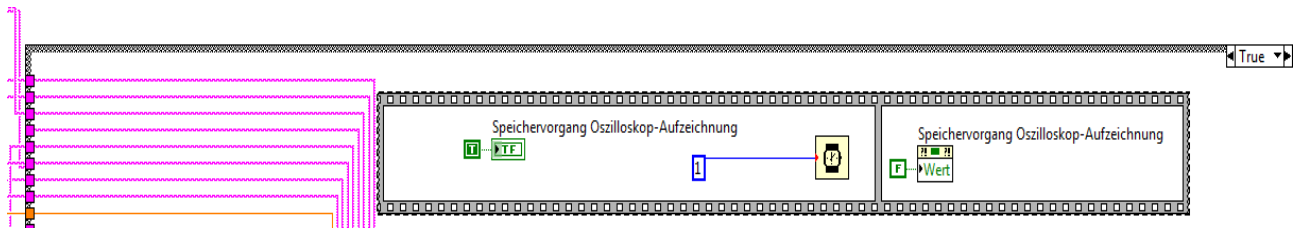
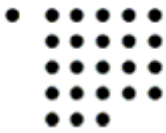


Abbildung 5.37: Osz\_LED

Im Frontpanel befindet sich dieses in Abb.7.6 von Anhang 7.3 A3 Registerkartenfenster, ganz oben im Bild in der dritten Reihe(von oben nach unten gezählt) mit der Anzeige und dem Schaltelement.

#### 5.2.2.4.7.4 Taster „Rücksetzen der Kennwerte“

Für die Erläuterung siehe Kapitel 5.2.2.4.4. Im Frontpanel befindet sich die Oberfläche dazu in Abb.7.6 von Anhang 7.3 A3 Registerkartenfenster, ganz oben im Bild in der vierten Reihe (von oben nach unten gezählt) mit dem Schaltelement.

#### 5.2.2.4.7.5 Start und Stopp der Oszilloskop-Aufzeichnung

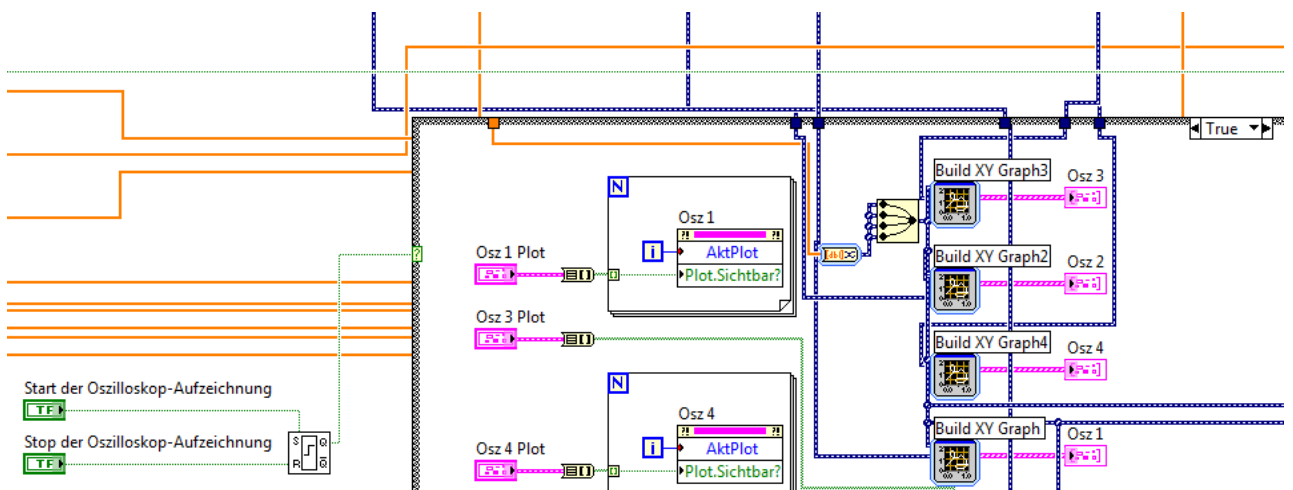


Abbildung 5.38: ST\_SP\_Osz

Für das Starten und Stoppen der Oszilloskop-Aufzeichnung wurden Schaltelemente erstellt(hier: „Start der Oszilloskop-Aufzeichnung“ und „Stop der Oszilloskop-Aufzeichnung“, Abb.5.38). Der Anfangswert der Sub-VI „SR-Flipflop“ beim Ausgang „Q“ hat, wie schon in Kapitel 5.2.2.4.6 erwähnt, den Standardwert „True“. Bei „True“ erfolgt die Oszilloskop-Aufzeichnung(auf die noch eingegangen wird) in dem Case auf die in den nächsten Kapitel eingegangen wird. Wird „Stop der Oszilloskop-Aufzeichnung“, welcher am „Rücksetz“-Eingang anliegt, betätigt(Tastendruck), so wird der Ausgang „Q“ rückgesetzt. Der Inhalt des Cases(in Abb.5.38) ist bei „False“ leer, das heißt es passiert nichts, somit werden die Oszilloskope gestoppt. Bei einem Tastendruck am „Setz“-Eingang bekommt man am „Q“-Ausgang wieder ein „True“. Im Frontpanel befindet sich die Oberfläche dazu in Abb.7.6 von Anhang 7.3 A3 Registerkartenfenster, ganz oben im Bild in der vierten Reihe (von oben nach unten gezählt) mit den beiden Schaltelementen.

### 5.2.2.4.7.6 Start und Stopp der Kenndaten-Aufzeichnung

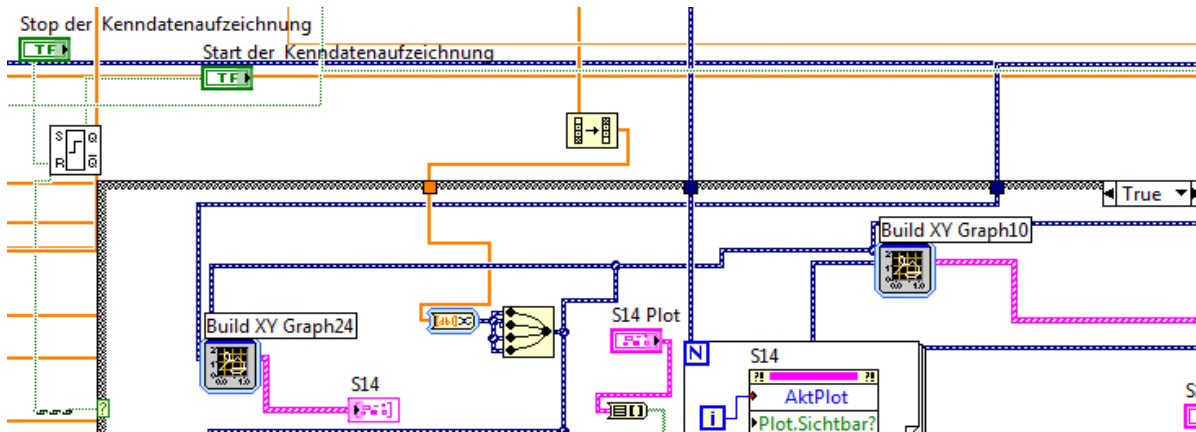


Abbildung 5.39: Kenndaten-Aufzeichnung

Hier wurde das gleiche Prinzip wie in Kapitel 5.2.2.4.7.5 durchgeführt. Auf den „True“-Teil wird noch eingegangen, beim „False“-Fall passiert nichts, sprich der Inhalt ist leer. Im Frontpanel befindet sich die Oberfläche dazu in Abb.7.6 von Anhang 7.3 A3 Registerkartenfenster, ganz oben im Bild in der fünften Reihe (von oben nach unten gezählt) mit den beiden Schaltelementen. Metadaten wie Titel und Username(als Bedienelement) für die Schreiberkurve und Oszilloskop-Aufzeichnung befinden sich ebenfalls im Reiter Bedienung(auch im Anhang 7.3 A3 Registerkartenfenster zu sehen, die letzten Reihen).

### 5.2.2.4.8 Fenster „Einstellungen“

#### 5.2.2.4.8.1 Angaben Messintervall

Wie schon in Kapitel 5.2.2.3.1 erläutert, werden die Abtastzeit und Abtastfrequenz über Variable eingestellt. Demnach kann man über die Formel 4.4.1.2 in Kapitel 4.4.1, die Frequenzauflösung ermitteln. Nimmt man nun den Kehrwert, so erhält man die Messzeit der Oszilloskopauflösung, dies sieht dann wie folgt aus:

$$T_{mess} = \frac{1}{f_{Auf}} = \frac{N}{f_{ab}} \quad (5.2.2.1)$$

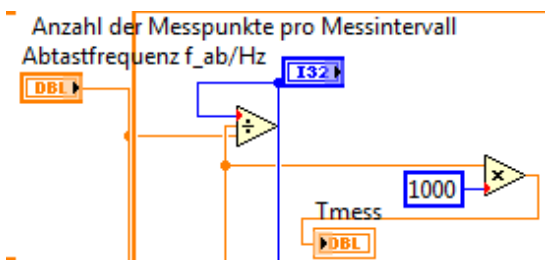


Abbildung 5.40:  $T_{mess}$

Um die Formel in Labview(Abb. 5.40) zu realisieren wurde „Anzahl der Abtastung pro Messintervall“ durch die „Abtastfrequenz  $f_{ab}/\text{Hz}$ “ geteilt. Daraus erhält man die Messzeit. Nun war aber die Messzeit der Oszilloskopaufzeichnung in der Einheit ms, laut Spezifikation, gefordert deswegen wird noch einmal um 1000 multipliziert. Ausgegeben wird das ganze dann in

ein Anzeigeelement als eine DBL-Variable „Tmess“ ausgegeben. Im Frontpanel befindet sich die Anzeige dazu in Abb.7.7 von Anhang 7.3 A3 Registerkartenfenster, in dem Bereich „Eigenschaften der Oszilloskop und Kenndaten-Aufzeichnung“(rechts im Abbild). Hier befindet sich die Abtastfrequenz,Anzahl der Messpunkte pro Messintervall und die Messzeit(Anzeigeelemente, rechts wo ein Bruchstrich ist).

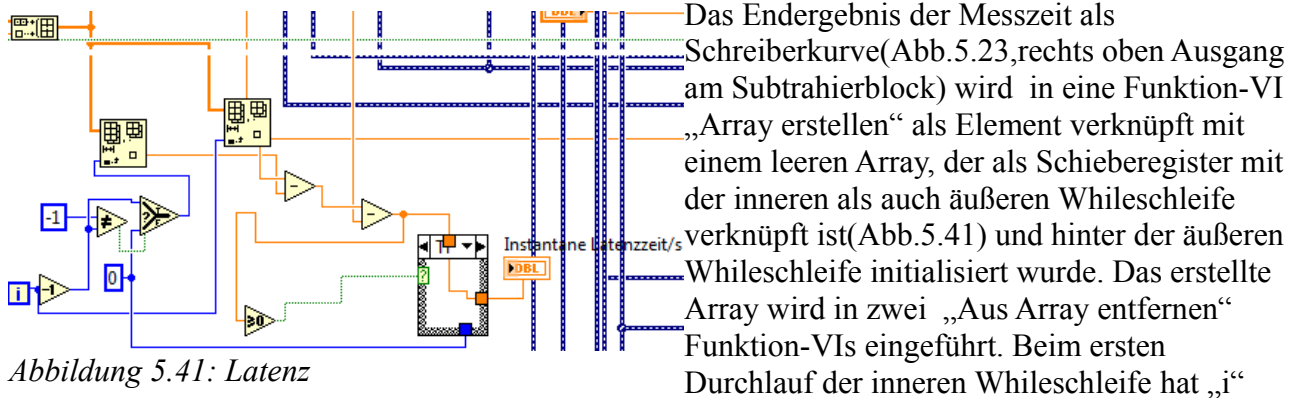
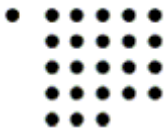


Abbildung 5.41: Latenz

Das Endergebnis der Messzeit als Schreiberkurve (Abb. 5.23, rechts oben Ausgang am Subtrahierblock) wird in eine Funktion-VI „Array erstellen“ als Element verknüpft mit einem leeren Array, der als Schieberegister mit der inneren als auch äußeren Whileschleife verknüpft ist (Abb. 5.41) und hinter der äußeren Whileschleife initialisiert wurde. Das erstellte Array wird in zwei „Aus Array entfernen“ Funktion-VIs eingeführt. Beim ersten Durchlauf der inneren Whileschleife hat „i“ den Wert 0. 0 wird um 1 dekrementiert, somit würde man „-1“ bekommen, dabei hätte die „Auswählen“-VI den Wert „False“ (Abb. 5.41, links unten, weil -1 nicht ungleich -1 ist). Der „False“-Teil würde die 0te Zeile entfernen und für die erste Funktion-VI „Aus Array entfernen“ in Abb. 5.41 (links von beiden) als Element ausgeben. Die zweite Funktion-VI „Aus Array entfernen“ würde auch somit die 0te Zeile ausgeben, die beiden Elemente werden nun subtrahiert und ein weiteres mal mit der Oszilloskop-Messzeit subtrahiert. Ist dieser Wert nun größer gleich „0“, so hat der Case (in Abb. 5.41) ein „True“ und der subtrahierte Wert wird an das Anzeigeelement (DBL) „Instantane Latenzzeit/s“ weitergeleitet. Bei einem „False“ (Abb. 5.42, also nicht größer gleich 0), wird eine 0 als Konstante übernommen. Dies ist bei dem ersten Durchlauf der Fall, weil erst Anfangszeit keine Latenzzeit hat.

Abbildung 5.42: False\_Lat

Beim 2ten Durchlauf ist  $i = 1$  dekrementiert, „0“ also ungleich -1, somit wäre die „Abfrage“-VI in Abb. 5.41 „True“. Die erste Funktion-VI „Array entfernen“ in Abb. 5.41 (links von beiden) würde somit die Zeit an der 0ten Zeile entfernen und ausgeben, die zweite hingegen an der 1ten, da direkt mit „i“ verbunden ist. Die Zeit der Arrays an der 1ten Zeile liegt in der Zukunft, das heißt der DBL-Wert an dieser Stelle ist höher als das Element in der 0ten Zeile. Von dem Arrayelement an der 0ten Zeile wird das Arrayelement an der 1ten Zeile subtrahiert. Dieser wird nochmal von der Messzeit subtrahiert. Die Endzahl wäre dann größer gleich 0, somit würde die Zahl den Case als „True“ passieren und den Wert als Anzeigeelement „Instantane Zeit/s“ ausgeben. Im Frontpanel befindet sich die Anzeige dazu in Abb. 7.7 von Anhang 7.3 A3 Registerkartenfenster, in dem Bereich „Eigenschaften der Oszilloskop und Kenndaten-Aufzeichnung“ (ganz unten rechts im Abbild).

### 5.2.2.4.8.2 Angaben Datenerfassung

Jedes Kanal hat die Möglichkeit mit einem anderen Kanal verbunden zu werden.

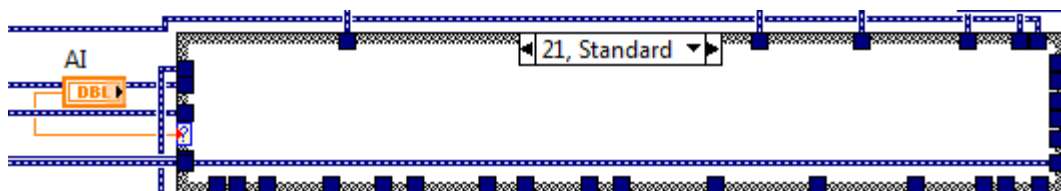
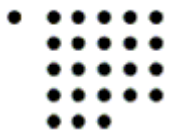


Abbildung 5.43: Ersatzkanal(AI21 U\_Egen)

Deshalb wurden für die Ersatzkanäle Fallabfragen (Case) erstellt wie in Abb. 5.43 (hier nur für AI21 U\_Egen). Als Standardkanal wird der zugeordnete Kanal in der Abfrage bevorzugt. Über eine DBL-Variable als Bedienelement, kann der Benutzer dann zwischen AI0 und AI31 den gewünschten Kanal einstellen. Wie man in Abb. 5.43 sehen kann, wird der Kanal AI21 (da er hier der Standardkanal ist) bevorzugt und durchgelassen. Wenn der Benutzer nun U\_Egen (als BNC-Anschluss) an den Kanal AI0 anschließen will, sollte er in das DBL-Bedienelement „AI“ eine „0“



reinschreiben.

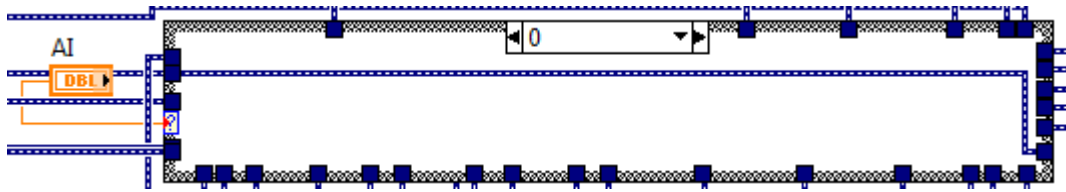


Abbildung 5.44: AI0 Wechsel

Somit würde die Fallabfrage auf 0 zutreffen und AI0 wird durchgelassen. Im Frontpanel befindet sich die Anzeige dazu in Abb.7.8 von Anhang 7.3 A3 Registerkartenfenster, da leider der andere Bachelorarbeiter der für die Hardware zuständig ist, nicht fertig geworden ist wurde [10V;-10V] als Eingangsspannungsbereich angenommen(diesen kann man im Programm ändern).

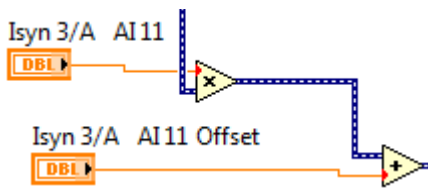


Abbildung 5.45: Skalierung und Offset

Jedes Kanal braucht einen zusätzlichen Skalierungsfaktor und Offset. Der Skalierungsfaktor(in Abb.5.45, hier nur für Isyn 3) „Isyn3/A AI11“ wird als DBL-Bedienelement mit dem messenden Signal „Isyn3/A AI11“ multipliziert. Nach der Multiplikation erfolgt eine Addition mit dem DBL-Bedienelement „Isyn 3/A AI11 Offset“. In die Bedienelemente(Abb.5.45) kann der Benutzer dann selber reinschreiben. Im Frontpanel befindet sich die Anzeige dazu in

Abb.7.7 von Anhang 7.3 A3 Registerkartenfenster, in dem Bereich „Skalierungsfaktor und Offset“(ganz oben im Abbild). Außerdem kann der Benutzer den Dateinamen zusätzlich in Abb.7.7 im Bereich „Name/Beschriftung/Pfad(Datei)“ einstellen.

#### 5.2.2.4.9 Fenster Kenndaten

Nachdem alle Kennwerte aus Kapitel 4.7.4.2 berechnet wurden, sind diese nun in Zeigerinstrumente mit Zahlenanzeige darzustellen. Insgesamt existieren 57 Kenndaten. Aufgrund der hohen Anzahl ist es sinnvoll, sparsam und übersichtlich ein Zeigerinstrument mit 3 Zeigern zu verwenden. Dabei ist für die Zuordnung hilfreich z.B. [Unetz1,Unetz2,Unetz3] als drei Nadeln für ein Zeigerinstrument zu verwenden.



Abbildung 5.46: Cluster Bündel Zeiger

Ein Beispiel zeigt Abb., hier wurde zuerst einmal die Scheinleistung 1,2,3 des Synchrongenerators bestimmt. Danach in die VI „Bündeln“ nach der Reihe gebündelt und in dem Zeigerinstrument „Sphase-Synchronmaschinenleistung“ ausgegeben. Für die Differenzspannung wurden Farbfelder in Form von LEDs verwendet. Diese Farbfelder haben die Aufgabe, Abweichungen in Prozent zu bestimmen. Nebenbei wurden extra Anzeigeelemente für die Differenzspannungen und Phasendifferenzen erstellt(wie schon in Kapitel und erläutert). In Abb.5.13 wird das Array in Kapitel 5.2.2.3.2.4 welches von der Funktions-VI „Von dynamischen Daten konvertieren“ konvertiert wurde durch

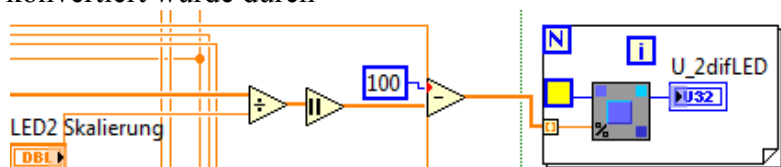
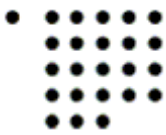


Abbildung 5.47: Farbelement

die DBL-Variable „LED2 Skalierung“ geteilt(Abb. 5.47). Diese Skalierung ist notwendig für die Umrechnung in Prozent. Bei einer Annahme:



$U_{netz2} = 1V \sin(\omega t)$  wäre die maximale Differenzspannung, mit der gleichen Effektivspannung vom Synchrongenerator, bei einer Phasenverschiebung von  $\frac{T}{2}$  :  $U_{syn2} = 1V \sin(\omega t + \frac{T}{2})$  .

Der maximale instantane Spannungsdifferenzwert wäre hierbei bei  $t = \frac{T}{4}$  mit dem Wert

$U_{diffmax2} = \sqrt{2}V + \sqrt{2}V = 2,82V$  . Um diesen Wert in Prozent umzurechnen teilt man über die DBL-Variable „LED2 Skalierung“ als Bedienelement durch den Wert  $\frac{2,82}{100\%}$  oder  $\frac{0,0282}{\%}$

(Diesen Wert sollte man auch eintragen in die DBL-Variable „LED2 Skalierung“ ). Weil bei  $t = \frac{T}{4}$  , 100% der instantanen Differenzspannung erreicht ist. Davon wird der Betrag genommen (da es in dem Fall keine negativen Prozentzahlen existieren) und von der Zahl 100 subtrahiert (Alle Arraywerte, Abb. 5.47). Nun wird das Array mit einer FOR-Schleife verknüpft, dabei wird das „N“ automatisch bestimmt, alle Elemente im Array passieren nun nach dem Zähler „i“ in die FOR-Schleife. Die Elemente sind verbunden mit dem Eingang „Prozentualen Anteil“ in der FOR-Schleife an dem VI „Farbe für Hervorhebung“. An dem „Farbe“-Eingang der VI „Farbe für Hervorhebung“ wurde die gelbe Farbfeldkonstante bevorzugt, wie es auch in der Spezifikation der Fall ist. Nun wird der Ausgang „Hervorheben“ weitergeleitet an das Farbfeld „U\_2difLED“. Das Farbfeld zeigt bei einer Synchronisation ein sehr schwaches flackerndes Gelb oder gar kein Gelb an. Hingegen beim Gegenteil ein sehr starkes flackerndes Gelb an. Im Frontpanel befindet sich die Anzeige dazu in Abb.7.9 von Anhang 7.3 A3 Registerkartenfenster.

#### 5.2.2.4.10 Fenster „Oszillosgramme“

Wie schon in Kapitel 5.2.2.4.3 anhand Abb. 5.20 erläutert, wird die Oszilloskop-Messezeit in der äußeren Whileschleife erstellt. Von diesem führt eine Verbindung in die innere Whileschleife.

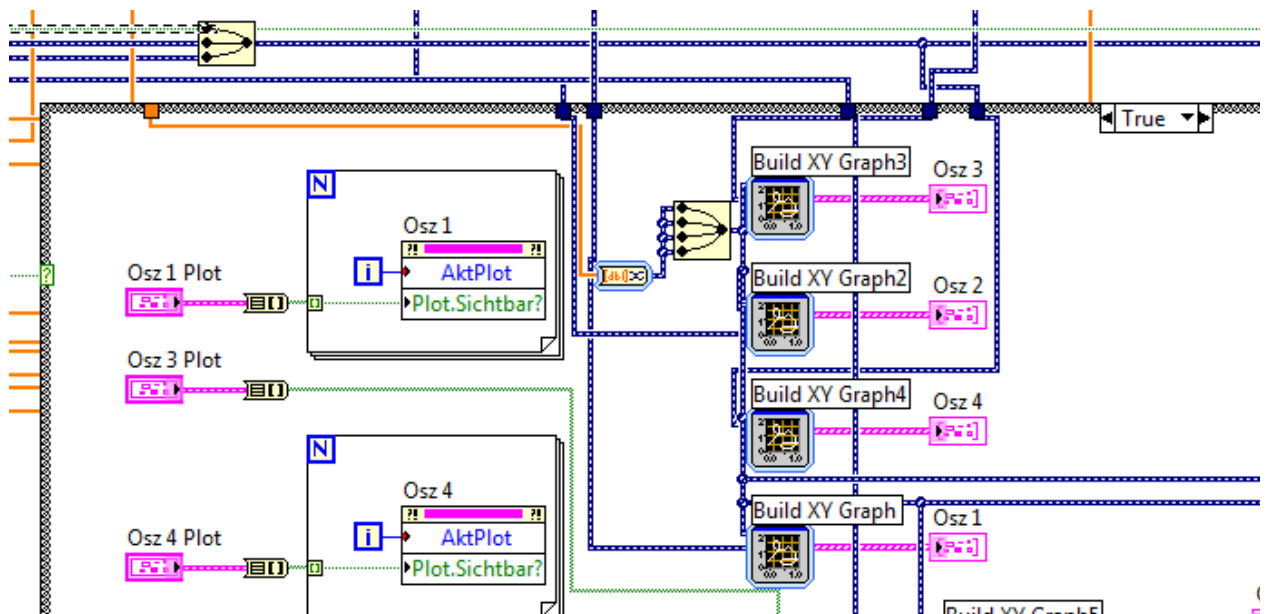
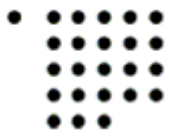


Abbildung 5.48: Oszgramme

Danach in den Case wo sich die XY-Diagramme befinden. Dieser wird in ein dynamischen Signal mit der Express-VI „In dynamische Daten konvertieren“ konvertiert (Abb.5.48, in der Mitte). Um jetzt die Zeit in der X-Achse für 4 Signale der Y-Achse darzustellen ist es notwendig, das





konvertierte Signal 4mal mit der Funktion-VI „Signale zusammenfassen“ zu verbinden und als kombinierten Signal in alle, die sich in dem Case befindenden, „XY-Graphen erstellen“ Express-VI als „X-Eingang“ einzuführen. Die Y-Signale sind die berechneten Messpunkte aus Kapitel 4.7.4.1 und Messsignale aus Kapitel 4.7.3.1. Um weniger XY-Diagramme zu verwenden bzw. Diagramme zu sparen, wurden die Signale wie in Abb.5.48 (oben links) über die Funktion-VI „Signale zusammenfassen“ als ein Signal kombiniert und in ein „XY-Graphen erstellen“ Express-VI als „Y-Eingang“ eingeführt (in der Form z.B. [Unetz1,Unetz2,Unetz3]). Im Ausgang „XY-Graph“ der Express-VI „XY-Graphen erstellen“ wird das ganze dann mit einem XY-Diagramm verknüpft (Abb.5.48 , über „Build XY Graph4“, Osz4 ). Für die Plots wurde ein spezieller Prozess erstellt:

Ein Cluster(hier: Osz 4 Plot) welches 4 boolesche Auswahlfelder enthält ist verknüpft mit einem „Cluster nach Array“-VI. Dieser konvertiert den Cluster in ein 1D-Array und verknüpft mit einer FOR-Schleife. Die FOR-Schleife bekommt jetzt automatisch das „N“ über die Anzhl der Elemente im Array(hier 4). Also wird in der Zählerschleife viermal die Plotsichtbarkeit mit dem aktiven Plot als Eigenschaftsknoten des XY-Diagramm „Osz 4“(in Abb.5.48, links)eingelesen. Angenommen der Benutzer vergibt Plot 1 und Plot 4 als Auswahlfeld den Wert „True“. So läuft die Zählerschleife, bei „i = 0“ wäre der aktive Plot „Plot 1“ dieser hat den Wert „True“ und das Plot ist sichtbar, bei „i = 1“ wäre der aktive Plot „Plot 2“ dieser ist „False“ und somit als Plot deaktiviert(unsichtbar), bei „i = 2“ wäre der aktive Plot „Plot 3“ dieser ist „False“ und somit als Plot deaktiviert, bei „i = 3“ wäre der aktive Plot „Plot 4“ dieser ist „True“ und das Plot ist sichtbar und beim nächsten Durchlauf wäre die Abbruchbedingung der Schleife erfüllt. In dieser Art und Weise kann man die Plots regeln. Wie schon erläutert haben alle berechneten Messpunkte aus Kapitel 4.7.4.1 und Messsignale aus Kapitel 4.7.3.1 über 3 bis 4 zusammengefasste Signale einen „XY-Diagramm“ zugewiesen bekommen. Im Frontpanel befindet sich die Anzeige dazu in Abb.7.10 und 7.11 von Anhang 7.3 A3 Registerkartenfenster. Die Steuerung des Cases ist in Kapitel 5.2.2.4.7.5 erläutert.

#### 5.2.2.4.11 Fenster „Schreiberkurven“

Einen ähnlichen Aufbau hat die Aufzeichnung der Schreiberkurvenbilder.

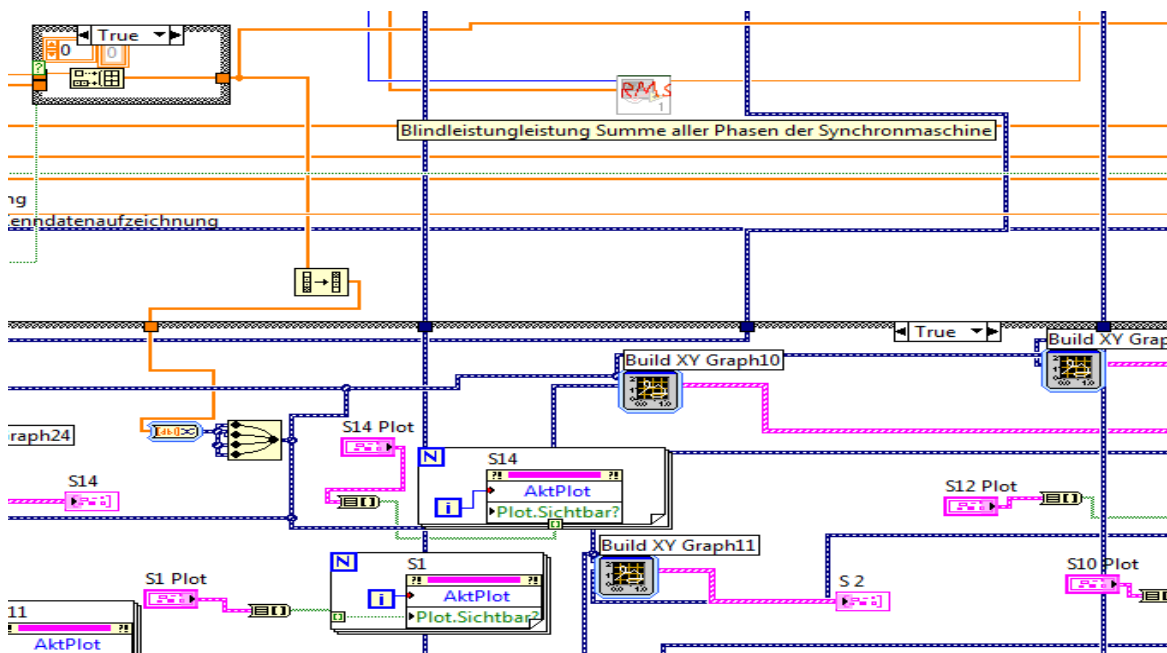
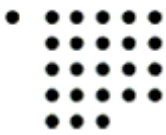


Abbildung 5.49: S\_Kurve



Die Verarbeitung des Messzeit-Arrays für die Schreiberkurve wurde in Abb 5.49(links oben der Case) anhand Abb.5.25 und 5.26 in Kapitel 5.2.2.4.4 erläutert. Nun wird das Array umgekehrt mit der VI „Array umkehren“, damit die Express-VI „XY-Graphen erstellen“ die Zeit nicht von rechts nach links sondern von links nach rechts aufzeichnet dieser wird wie in Kapitel 5.2.2.4.10 verarbeitet und als „X-Eingang“ eingeführt. Die Kenndaten von Kapitel 4.7.4.2 werden ebenfalls gebündelt und als „Y-Eingang“ eingeführt(Abb 5.49) . Das Plot hat ebenfalls die gleicher Verarbeitung wie in Kapitel 5.2.2.4.10 . Im Frontpanel befindet sich die Anzeige dazu in Abb.7.12,7.13 und 7.14von Anhang 7.3 A3 Registerkartenfenster. Die Steuerung des Cases ist in Kapitel 5.2.2.4.7.6 erläutert.

### 5.2.2.5 Ergebnisse und Fazit

Die Dateinamen funktionieren wie gefordert :

2013-09-30_0001-SyncGen-test1	30.09.2013 14:33	DAT-Datei	4 KB
2013-09-30_0002-SyncGenOsc-test1	30.09.2013 14:33	DAT-Datei	341 KB
2013-09-30_0001-SyncGen	30.09.2013 14:33	DAT-Datei	33 KB
2013-09-30_0001-SyncGenOsc	30.09.2013 14:33	DAT-Datei	341 KB

Abbildung 5.50: Dateiname\_er

Die gespeicherten Oszilloskop-Aufzeichnung und Kenndaten sehen folgendermaßen aus(nur teilweise):

Titel	Test1	
Date_Time	2013-09-30	14:33:12.108
User	Frau	Muster
Unetz_1rms/V	10.548641	
Unetz_2rms/V	10.548641	
Unetz_3rms/V	10.548641	
lasy1rms/A	10.548641	
lasy2rms/A	10.548641	

Abbildung 5.51: Osziauf\_teil1

Zeit/s	Unetz_1/V	Unetz_2/V
0.000000	10.548641	10.548641
0.000100	10.548641	10.548641
0.000200	10.548641	10.548641
0.000300	10.548641	10.548641
0.000400	10.548641	10.548641

Abbildung 5.52: Osziauf\_teil2

Titel	Test1		
Date_Time	2013-09-30	14:33:23.995	
User	Frau	Muster	
Zeit/s	Unetz_1rms/V	Unetz_2rms/V	Unetz_3rms/V
14.853516	10.548641	10.548641	10.548641
14.995117	10.548641	10.548641	10.548641
15.134766	10.548641	10.548641	10.548641
15.275391	10.548641	10.548641	10.548641
15.416992	10.548641	10.548641	10.548641

Abbildung 5.53: S\_Kurveauf

Bei den Einstellungen wird nach Eingabe der Abtastfrequenz und Anzahl der Abtastungen , die instantane Latenz-,Mess- und Abtastzeit ausgegeben:

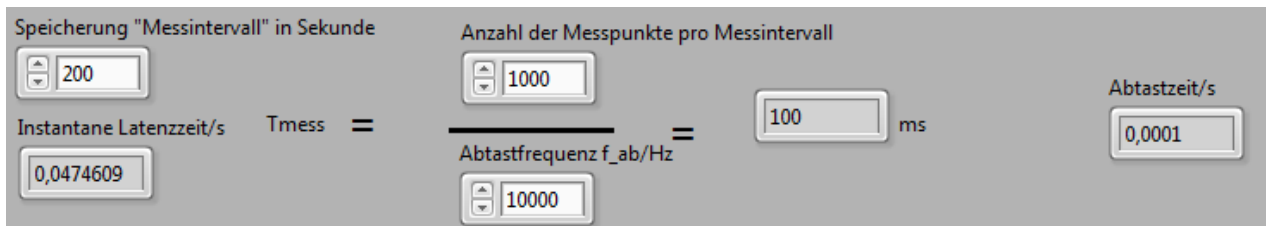
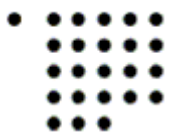


Abbildung 5.54: Ausgabe\_Einstellung

Setzt man nun bei einer Wechselnetzspannung von  $u(t) = 3,53 V \sin(\omega t)$  an Kanal AI0 ( $u_{max} = 5V$ ) ein Skalierungsfaktor von 2 und einen Offset von 5V erhält man folgenden Wert beim Oszilloskopbild:

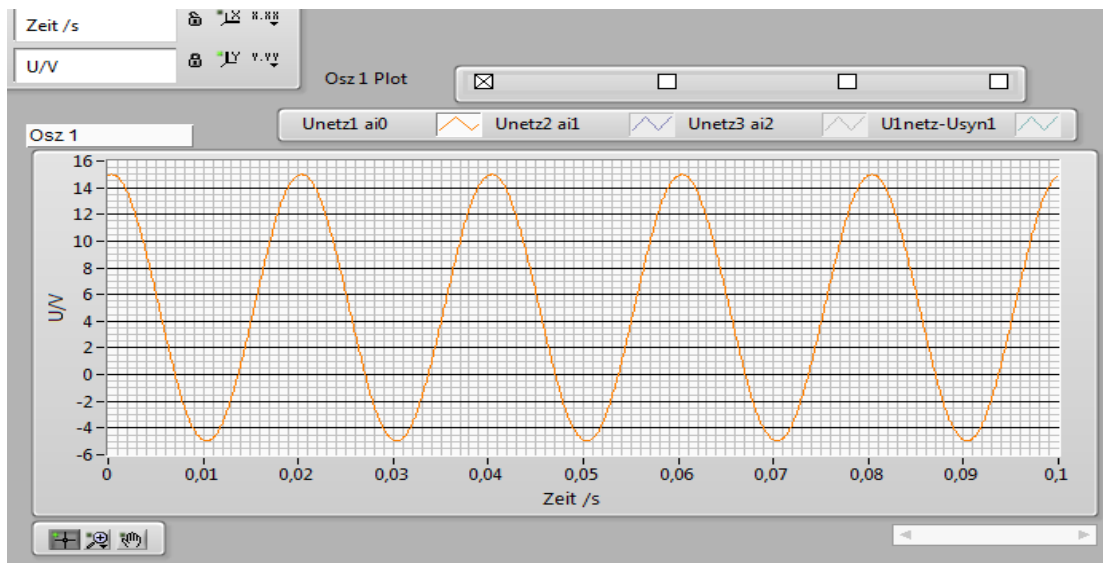


Abbildung 5.55: Osz\_Bild

Das heißt als maximalen Wert hätte die Funktion  $u_{maxneu} = u_{max} \cdot 2 + 5V = 15V$  und als minimalen Wert  $u_{minneu} = -u_{max} \cdot 2 + 5V = -5V$  und somit bestätigt Abb.5.55 die These. Außerdem sieht man in Abb.5.55 den Plot der nur für Unetz ai0 vergeben wurde und für den Rest nicht. Geht man weiterhin von einer Wechselnetzspannung  $u(t) = 3,53 V \sin(\omega t)$  an AI0 und einem Strom von  $i(t) = 3,53 A \sin(\omega t)$  an AI3 mit einem Skalierungsfaktor von 1 und keinem Offset aus. So erhält man bei den Kenndaten:

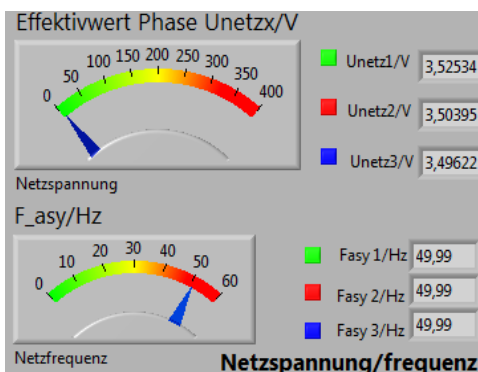


Abbildung 5.56: Kenndatenwertefundu

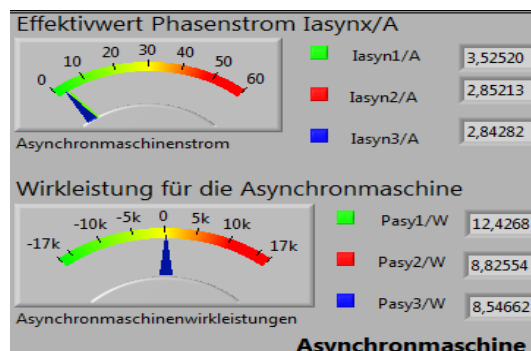
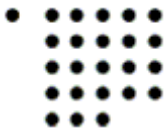


Abbildung 5.57: KenndatenwertelundP

Wie man sehen kann, erfüllen auch die Zeigerinstrumente ihre Aufgaben. Angezeigt Werden



$I_{eff} = 3,5253 A$  ,  $U_{eff} = 3,5253 V$  ,  $f_{asy1} = 50Hz$  und  $P_{asy1} = 12,42 W$ .  $P=S$  und  $Q$  annähernd 0:

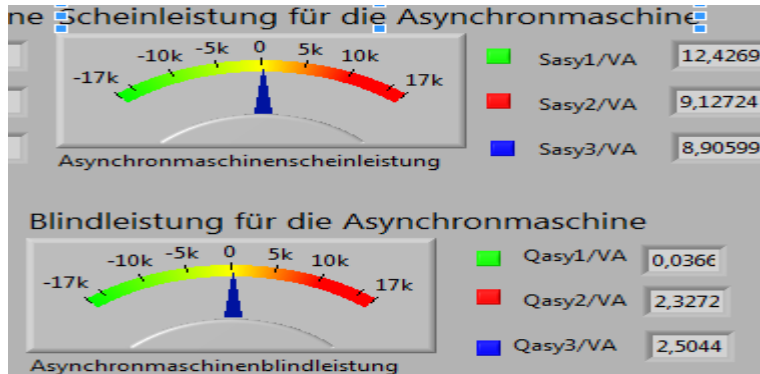


Abbildung 5.58: SP-Kenndaten

Die Kenndaten-Aufzeichnung würde folgendes ausgeben:

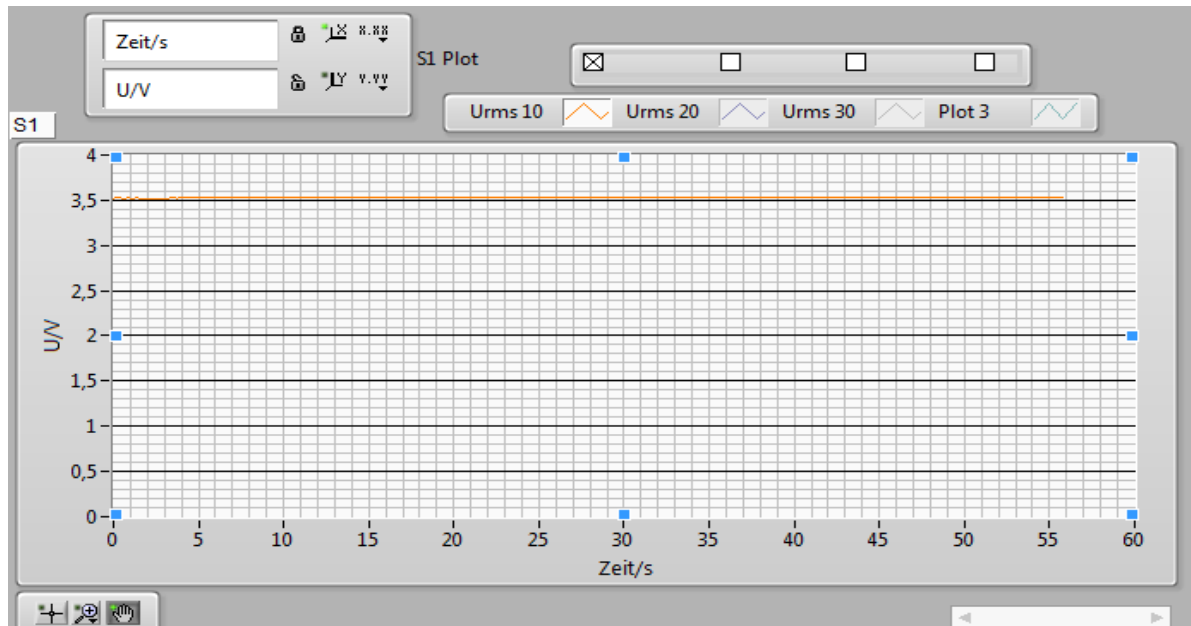


Abbildung 5.59: Kenndatenaufzeichnung

Auch hier sieht man die Plotauswahl, die hier funktioniert. Beim Testen der Dunkelschaltung kann man z.B. bei einer Netzspannung  $u(t) = 3,53 V \sin(\omega t)$  an AI0, die gleiche Netzspannung als Negiertes (eingestellt über den Skalierungsfaktor) an AI6 einführen. AI6 hätte dann den Spannungssignal  $-u(t) = -3,53 V \sin(\omega t) = 3,53 V \sin(\omega t + 180)$  also die maximalste Verschiebung.

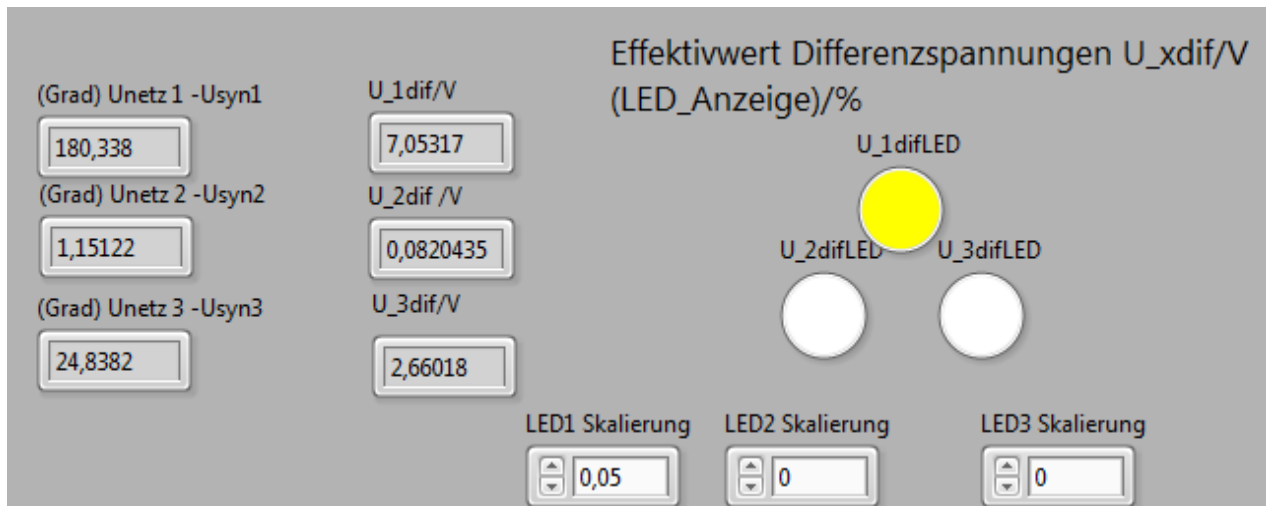
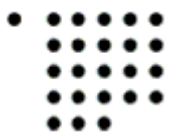


Abbildung 5.60: Dunkelprog

Die „LED1 Skalierung“ hätte dann den Wert 0,05(wegen 5V), „U-1dif/V“ die maximale Verschiebung 7,053V(weil bei 90° der Wert 10V ist und somit der Effektivwert der Differenz 7,053V) und der maximale Verschiebungswinkel 180°. Die Lampe flackert und somit funktioniert auch die Dunkelschaltung.

### 5.3 Bedienungsanleitung für Programm benutzer

Die Bedienungsanleitung soll den Benutzer die Anwendung des Programms erleichtern.

Schritt 1 Reiter „Einstellungen“:

Abb.7.7:

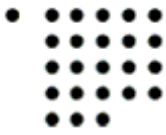
Bitte geben sie Name und Pfad der Datei vor dem Programmablauf ein, die erstellt werden soll. Die erstellte Datei wird in dem gewünschten Pfad erstellt und in der Form z.b. als „2013-07-15\_0001-SyncGen-Praktikumsversuch1.DAT“ versehen. Um die Oszilloskop-Messzeit einzustellen(vor dem Programmablauf), bitte regeln sie das über die „Abtastfrequenz f<sub>ab</sub>/Hz“ und „Anzahl der Messpunkte pro Messintervall“. Zum Beispiel:

$$\frac{\text{Anzahl der Abtastung}}{\text{Abtastfrequenz}} = \frac{1000}{10000 \text{ Hz}} \text{ daraus ergibt sich dann die Messzeit } T_{\text{mess}} = 100\text{ms} \text{ beim}$$

Ausführen des Programms, außerdem wird die Abtastzeit und instantane Latenzzeit angezeigt. Weiterhin kann man die Zeit der Oszilloskop-Aufzeichnung „Speicherung "Messintervall" in Sekunde“ einstellen. Um Skalierungsfaktoren einzustellen sollte man als erstes das Verhältnis berechnen. Wenn zum Beispiel an der primären Spannung 400V Effektivwert anliegt und als Sekundäre 3V so sollte man 400V durch 3V teilen und als Skalierungsfaktor 133,3 eintragen. Ist die sekundäre Spannung mit einem Offset verschoben, so kann man ihn zum Beispiel durch einen weiteren Offset am Ursprung wieder richten.

Abb.7.8:

Die Kanäle haben alle einen Standardkanal, ist nun aber ein Kanal defekt oder will man einen Kanal umtauschen sollte man in eines der AI mit dem Standardkanal einen anderen Kanal zuweisen. Zum Beispiel der Kanal AI 0 ist defekt oder zeigt gar nichts am Oszilloskop an . Der Benutzer hat jetzt die Möglichkeit Unetz1/V zum Beispiel einen unbenutzten Kanal AI 15 zu benutzen und dem Programm zuzuweisen. Bitte schreiben sie anstatt 0 eine 15 in den Eingang AI rein, entnehmen sie



vor der Ausführung des Programms den BNC-Anschluss Unetz1/V von dem Kanal AI 0 und stöpseln sie ihn in AI 15 rein.

Schritt 2 Reiter „Kenndaten“:

Abb 7.9:

Da die Lampen als Differenzspannung mit Prozentwerten arbeiten ist auch hier eine Skalierung notwendig. Hat man zum Beispiel ein zwei Sinus-Signale mit der gleichen Spannung Bsp. 400V könnte hierbei die maximale Phasenverschiebung bei 180°(Worst Case ) liegen. Somit wäre die maximale Differenzspannung bei 90° mit einem instantanen Wert von 800V. Deshalb sollte man immer den Worst Case Wert nehmen, dieser wäre hierbei  $\frac{800}{100\%} = \frac{8}{\%}$  , den Wert 8 trägt man nun in alle DBL-Variable „LED-Skalieren“ ein.

Schritt 3 Reiter „Oszillosgramme“ und „Schreiberkurve“:

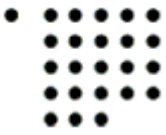
Abb.7.10 bis 7.14

Um die Plots sichtbar bzw. unsichtbar zu machen kann man diese über die Plotlegende als Auswahlfeld deklarieren. Hat das Auswahlfeld ein Kreuz z.b. an Plot 1 so ist dieser sichtbar und umgekehrt unsichtbar. Navigieren der Bilder kann man zusätzlich mit der Graph-Palette oder X-Bildaufleiste.

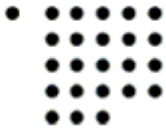
Abb.7.6

Schritt 4 Fenster „Bedienung“:

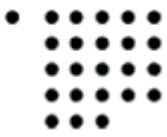
Bei einem Programmablauf blinkt eine grüne LED mit dem Namen „Programm läuft“. Das Programm kann man beenden mit der Taste „Programm beenden“, dann leuchtet die LED „Programm läuft“ nicht mehr. Die Kenndatenspeicherung wird automatisch beim ersten Start des Programms aktiviert. Dies kann man an der grünen LED „Kenndatenspeicherung“, die bei Aktivierung leuchtet und bei Deaktivierung nicht leuchtet, erkennen. Will man beim ersten Start keine Kenndatenspeicherung so sollte man die Taste „Deaktivieren der Speicherung von Kenndaten“ drücken. Wird das Programm beendet speichert die Kenndatenspeicherung den letzten gedrückten Wert. Will man die Kenndatenspeicherung beim Programmablauf aktivieren, so drückt man „Aktivieren der Speicherung von Kenndaten“ bei einer Deaktivierung „Deaktivieren der Speicherung von Kenndaten“. Will man die aktuelle Oszilloskop-Aufzeichnung speichern so drückt man „Speicherung der aktuellen Oszilloskop-Aufzeichnung“ und die grüne LED „Speichervorgang Oszilloskop-Aufzeichnung“ blinkt ganz kurz bei der Aufzeichnung. Will man die Oszilloskop-Aufzeichnung deaktivieren so drückt man den Knopf „Stop der Oszilloskop-Aufzeichnung“ bei einer Aktivierung „Start der Oszilloskop-Aufzeichnung“. Als Anfangswert ist die Oszilloskop-Aufzeichnung aktiviert sonst wird immer über den zuletzt gedrückten Wert entschieden. Die Funktion der Taster Kenndaten-Aufzeichnung ist die selber wie bei der Oszilloskop-Aufzeichnung.



## 6 Gesamtfazit







## 7 Anhang

### 7.1 A1 Spezifikation

#### Spezifikation für Messsoftware zum Synchrongeneratormessstand

E. Waffenschmidt, FH-Köln, 5.7.2013

#### Einleitung

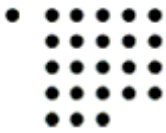
Für den existierenden Messstand eines Maschinensatzes soll eine neue Messdatenerfassungssoftware erstellt werden.

Der Maschinensatz besteht aus einem fremderregten Synchrongenerator, der von einem fremderregten Gleichstrommotor angetrieben wird. Dessen variable Betriebsspannung wird von einem fremderregten Gleichstromgenerator erzeugt, der von einer ans Dreiphasen-Stromnetz angeschlossenen Asynchronmaschine angetrieben wird. Die Synchronmaschine kann im Inselbetrieb oder nach Synchronisation am Dreiphasen-Stromnetz betrieben werden.

#### Zu erfassende Messdaten

Messsignale:

Bezeichnung	Sym- bol	Max.. Wertebereich	Typ. Signal- Frequenz	Genau- igkeit	Auflösung
Phasenspannungen des Netzes gegen Null-Leiter	$U_{10}$ , $U_{20}$ , $U_{30}$	-400 V ... +400V	50 Hz	<0.1%	>=12 bit
Phasenströme der Asynchronmaschine	$I_{asy1}$ , $I_{asy2}$ , $I_{asy3}$	-40 A ... +40 A Anlaufstrom ist noch höher, Schaltung darf nicht zerstört werden	50 Hz	<0.5%	>=10 bit
Phasenspannungen der Synchronmaschine gegen Null-Leiter	$U_{syn10}$ , $U_{syn20}$ , $U_{syn30}$	-400 V ... +400 V	50 Hz	<0.1%	>=12 bit
Phasenströme der Synchronmaschine	$I_{syn1}$ , $I_{syn2}$ , $I_{syn3}$	-40 A ... +40 A	50 Hz	<0.5%	>=10 bit
Erregerspannung der Synchronmaschine	$U_{excyn}$ $n$	0 ... ??V	DC	<1%	>=8 bit
Erregerstrom der Synchronmaschine	$I_{excyn}$	0 ... ??A	DC	<1%	>=8 bit
Gleichstrom	$U_{DC}$	0 ... ??V	DC	<1%	>=8 bit



Zwischenkreisspannung					
Gleichstrom Zwischenkreisstrom	$I_{DC}$	-50A ... +50A	DC	<1%	>=8 bit
Erregerspannung des Gleichstromgenerators	$U_{excgen}$	0 ... ??V	DC	<1%	>=8 bit
Erregersstromdes Gleichstromgenerators	$I_{excgen}$	0 ... ??V	DC	<1%	>=8 bit
Erregerspannung des Gleichstrommotors	$U_{excmo}$	0 ... ??V	DC	<1%	>=8 bit
Erregersstromdes Gleichstrommotors	$I_{excmot}$	0 ... ??V	DC	<1%	>=8 bit
Zeit seit Beginn des Messintervalls	t	0... 86400s (1 Tag)		<10 $\mu$ s	

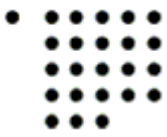
Dynamische Spezifikation für alle Signale

Bezeichnung	Symbol	Wert
Messintervall	$T_{mess}$	100 ms
Messpunkte pro Messintervall	$N_{mess}$	>= 1000
Samplerate	$R_s$	>= 10 kS/s
Bandbreite der Sensoren	$F_g$	>= 20 kHz

Aus Messdaten zu berechnende Daten

Für jeden Messpunkt zu berechnen:

Bezeichnung		Symbol	Berechnung
Instantane Leistungen der Synchronmaschine	Phase 1	$P_{syn1inst}$	$= U_{syn10} \cdot I_{syn1}$
	Phase 2	$P_{syn2inst}$	$= U_{syn20} \cdot I_{syn2}$
	Phase 3	$P_{syn3inst}$	$= U_{syn30} \cdot I_{syn3}$
Instantane Leistungen der Asynchronmaschine	Phase 1	$P_{asy1inst}$	$= U_{10} \cdot I_{asy1}$
	Phase 2	$P_{asy2inst}$	$= U_{20} \cdot I_{asy2}$
	Phase 3	$P_{asy3inst}$	$= U_{30} \cdot I_{asy3}$
Instantane Leistung der Gleichstrommaschinen		$P_{DCinst}$	$= U_{DC} \cdot I_{DC}$
Instantane Leistung des Erregerkreis Synchrongenerator		$P_{excsyncinst}$	$= U_{excyn} \cdot I_{excyn}$

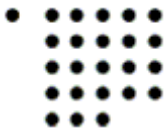


Instantane Leistung des Erregerkreis Gleichstromgenerator		$P_{excgeninst}$	$= U_{excgen} \cdot I_{excgen}$
Instantane Leistung des Erregerkreis Gleichstrommotors		$P_{excmotinst}$	$= U_{excmot} \cdot I_{excmot}$
Spannung zwischen Netz und Synchrongenerator	Phase 1	$U_{1dif}$	$= U_{10} - U_{syn10}$
	Phase 2	$U_{2dif}$	$= U_{20} - U_{syn20}$
	Phase 3	$U_{3dif}$	$= U_{30} - U_{syn30}$

Für jedes Messintervall zu berechnen:

Effektivwerte

Bezeichnung		Symbol	Berechnung aus
		y	$= \text{sqrt}(1/N_{mess} \cdot \sum x^2)$ x = gemessene Größe im Zeitintervall:
Netzspannungen	Phase 1	$U_{10rms}$	$U_{10}$
	Phase 2	$U_{20rms}$	$U_{20}$
	Phase 3	$U_{30rms}$	$U_{30}$
Phasenströme Asynchronmaschine	Phase 1	$I_{asy1rms}$	$I_{asy1}$
	Phase 2	$I_{asy2rms}$	$I_{asy2}$
	Phase 3	$I_{asy3rms}$	$I_{asy3}$
Phasenspannungen Synchrongenerator	Phase 1	$U_{syn10rms}$	$U_{syn10}$
	Phase 2	$U_{syn20rms}$	$U_{syn20}$
	Phase 3	$U_{syn30rms}$	$U_{syn30}$
Phasenströme Synchrongenerator	Phase 1	$I_{syn1rms}$	$I_{syn1}$
	Phase 2	$I_{syn2rms}$	$I_{syn2}$
	Phase 3	$I_{syn3rms}$	$I_{syn3}$
Spannung zwischen Netz und Synchrongenerator	Phase 1	$U_{1difrms}$	$U_{1dif}$
	Phase 2	$U_{2difrms}$	$U_{2dif}$
	Phase 3	$U_{3difrms}$	$U_{3dif}$
Zwischenkreisspannung Gleichstrommaschinen		$U_{DCrms}$	$U_{DC}$
Zwischenstrom Gleichstrommaschinen		$I_{DCrms}$	$I_{DC}$
Erregerspannung Synchronmaschine		$U_{excsynrms}$	$U_{excsyn}$



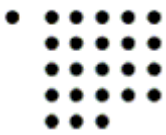
Erregerstrom Synchronmaschine		$I_{exc\,syn\,rms}$	$I_{exc\,syn}$
Erregerspannung Gleichstromgenerator		$U_{exc\,gen\,rms}$	$U_{exc\,gen}$
Erregerspannung Gleichstromgenerator		$U_{exc\,gen\,rms}$	$U_{exc\,gen}$
Erregerspannung Gleichstrommotor		$U_{exc\,mot\,rms}$	$U_{exc\,mot}$
Erregerspannung Gleichstrommotor e		$U_{exc\,mot\,rms}$	$U_{exc\,mot}$

### Wirkleistungen

Bezeichnung		Symbol	Berechnung aus
		y	$= 1/N_{mess} \cdot \sum x$ $x =$ gemessene Größe im Zeitintervall:
Synchronmaschine	Phase 1	$P_{syn1}$	$P_{syn1\,inst}$
	Phase 2	$P_{syn2}$	$P_{syn2\,inst}$
	Phase 3	$P_{syn3}$	$P_{syn3\,inst}$
	Gesamt	$P_{syn}$	$= P_{syn1} + P_{syn2} + P_{syn3}$
Asynchronmaschine	Phase 1	$P_{asy1}$	$P_{asy1\,inst}$
	Phase 2	$P_{asy2}$	$P_{asy2\,inst}$
	Phase 3	$P_{asy3}$	$P_{asy3\,inst}$
	Gesamt	$P_{asy}$	$= P_{asy1} + P_{asy2} + P_{asy3}$
Gleichstrommaschinen		$P_{DC}$	$P_{DC\,inst}$
Erregerwicklung Synchrongenerator		$P_{exc\,syn}$	$P_{exc\,syn\,inst}$
Erregerwicklung Gleichstromgenerator		$P_{exc\,gen}$	$P_{exc\,gen\,inst}$
Erregerwicklung Gleichstrommotor		$P_{exc\,mot}$	$P_{exc\,mot\,inst}$

### Scheinleistungen

Bezeichnung		Symbol	Berechnung aus
Synchronmaschine	Phase 1	$S_{syn1}$	$= U_{syn1\,rms} \cdot I_{syn1\,rms}$
	Phase 2	$S_{syn2}$	$= U_{syn2\,rms} \cdot I_{syn2\,rms}$
	Phase 3	$S_{syn3}$	$= U_{syn3\,rms} \cdot I_{syn3\,rms}$
	Gesamt	$S_{syn}$	$= \text{srqt} [P_{syn}^2 + Q_{syn}^2]$
Asynchronmaschine	Phase 1	$S_{asy1}$	$= U_{1\,rms} \cdot I_{asy1\,rms}$
	Phase 2	$S_{asy2}$	$= U_{2\,rms} \cdot I_{asy2\,rms}$



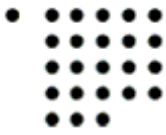
	Phase 3	$S_{asy3}$	$= U_{30rms} \cdot I_{asy3rms}$
	Gesamt	$S_{asy}$	$= \text{srqt} [P_{asy}^2 + Q_{asy}^2]$

### Blindleistungen

Bezeichnung		Symbol	Berechnung aus
Synchronmaschine	Phase 1	$Q_{syn1}$	$= \text{srqt} [S_{syn1}^2 - P_{syn1}^2]$
	Phase 2	$Q_{syn2}$	$= \text{srqt} [S_{syn2}^2 - P_{syn2}^2]$
	Phase 3	$Q_{syn3}$	$= \text{srqt} [S_{syn3}^2 - P_{syn3}^2]$
	Gesamt	$Q_{syn}$	$= Q_{syn1} + Q_{syn2} + Q_{syn3}$
Asynchronmaschine	Phase 1	$Q_{asy1}$	$= \text{srqt} [S_{asy1}^2 - P_{asy1}^2]$
	Phase 2	$Q_{asy2}$	$= \text{srqt} [S_{asy2}^2 - P_{asy2}^2]$
	Phase 3	$Q_{asy3}$	$= \text{srqt} [S_{asy3}^2 - P_{asy3}^2]$
	Gesamt	$Q_{asy}$	$= Q_{asy1} + Q_{asy2} + Q_{asy3}$

### Zeitabhängige Größen

Bezeichnung		Symbol	Berechnung aus	Genauigkeit
Netzfrequenz		$f_N$	Fitfunktion aus $U_{10}, U_{20}, U_{30}$	1 mHz
Frequenz Synchrongenerator		$f_{syn}$	Fitfunktion aus $U_{syn10}, U_{syn20}, U_{syn30}$	1 mHz
Phasendifferenz zwischen Netzspannung und Synchrongeneratorspannung		$\varphi_{Nsyn}$	Fitfunktion aus $U_{10}, U_{20}, U_{30}, U_{syn10}, U_{syn20}, U_{syn30}$	1°
Datum und Uhrzeit der Messung		$t_{int}$	Format: JJJJ-MM-TT HH:MM:SS.ss	10 ms



## **Datensicherung**

Die Messdaten werden in ASCII-Dateien in Tabellenform abgespeichert.

Datenformat und Dateinamen

Die Daten werden im Klartext lesbar als ASCII-Datei gespeichert.

Zahlen können als Fließkomma oder in exponentialschreibweise (mit „E“) geschrieben werden.

Als Dezimalpunkt wird – wie international üblich – ein Punkt verwendet.

Vor dem Dezimalpunkt braucht keine 0 zu stehen, darf aber.

Datum und Uhrzeit werden in folgendem Format angeben: YYYY-MM-TT HH:MM:SS.sss

Dabei sind YYYY das Jahr in 4-stelligen Ziffern, MM der Monat mit zwei Ziffern, TT der Tag des Monats mit zwei Ziffern, HH die Stunde im 2-stelligen 24h-Format, MM die Minuten im 2-stelligen Format, SS die Sekunden im zweistelligen Format und ss die Nachkommastellen der Sekunden, sofern notwendig.

Spalten werden getrennt durch „Tab“ ([→], ASCII-Code = 9, Hex 09).

Zeilen werden beendet mit „Carriage Return“ + „Line Feed“ ([←↓], ASCII-Codes = 13 + 10, Hex 0D + 0A).

Texteinträge wie Spaltenüberschriften, Variablenbezeichnungen oder Einheiten brauchen nicht in Anführungsstriche gesetzt werden. Es empfiehlt sich, Teile von Variablenamen nicht mit Space [ ] sondern stattdessen mit Unterstrich [ \_ ] zu trennen.

Die Datei besteht aus einem Datenkopf und dem Datenkörper.

Der Datenkopf enthält Angaben zu den Messdaten wie Titel und Parameter

Der Datenkörper enthält die eigentlichen Messdaten.

### **Datenkopf**

Der Datenkopf ist zweispaltig.

Der Datenkopf kann beliebig viele Zeilen haben.

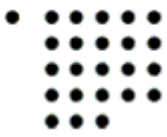
Die erste Spalte enthält den Namen des Parameters. Die zweite und ggf. folgende Spalten den Inhalt des Parameters

Die erste Zeile enthält den Titel der Messung. Daraus gehen Art und Umstände der Messdaten hervor. Name des Parameters: „Titel“

Die zweite Zeile enthält das Datum und die Uhrzeit, zu der die Datei angelegt wurde. Name des Parameters „Date\_Time“

Die dritte Zeile enthält den Namen des Benutzers. Name des Parameters „User“

Die weiteren Parameter sind zur freien Verfügung



## Datenkörper

Datenkopf und Datenkörper sind durch eine Leerzeile getrennt.

Die erste Zeile des Datenkörpers enthält die Spaltenüberschriften. Typischerweise enthält sie die Variablennamen und die dazugehörige Einheit.

Es müssen mindestens genauso viele Spaltenüberschriften existieren wie in den folgenden Zeilen Daten pro Messpunkt abgespeichert werden.

Jede Zeile entspricht einem Messpunkt. Sie kann Daten von beliebig vielen unterschiedlichen Variablen enthalten.

Es können beliebig viele Zeilen (=Datenpunkte) abgespeichert werden.

Der Datensatz endet mit dem Ende der Datei und hat keine extra Zeile am Ende.

## Beispiel

```
Titel      [→]Beispiel für eine ASCII-Datendatei[←↓]
Date_Time [→]2013-07-15 10:21:00[←↓]
User      [→]E. Waffenschmidt[←↓]
Temperatur/°C [→]25.0[←↓]
[←↓]
Zeit/s     [→]P1/W [→]P2/W [→]P3/W[←↓]
0.000     [→].123 [→]0.235 [→]10.356789[←↓]
0.011     [→]1.23E-01 [→]110.235 [→]9.345E02[←↓]
0.025     [→].123 [→]0.235 [→]10.345678[←↓]
```

## Messintervall

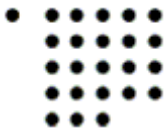
Die Messdaten eines Messintervalls („Oszilloskop-Kurven“) können als Datendatei abgespeichert werden. Für jedes Messintervall wird eine Datei angelegt. Es werden alle gemessenen (Kap. Messsignale:) und berechneten Messdaten (Kap. Für jeden Messpunkt zu berechnen:) abgespeichert. Die aus dem Messintervall berechneten Kenndaten (Kap. Für jedes Messintervall zu berechnen: ff) werden als Parameter abgespeichert.

Die Speicherung wird ausgelöst durch:

- Tastendruck des Benutzers: Speichert das zuletzt dargestellte Messintervall
- Timer: Speicher Messintervalle in vorgegeben Zeitabständen. Die Zeitabstände können vom Benutzer vorgegeben werden.

## Kennwerte aus Messintervall

Die Kennwerte eines Messintervalls (Kap. Für jedes Messintervall zu berechnen: ff) werden fortlaufend in einer Datendatei gespeichert. Dabei werden alle Kennwerte abgespeichert unmittelbar nachdem sie ermittelt wurden und bevor das nächste Messintervall aufgenommen wird. Bei einem Absturz des Computers sollen die gemessenen Daten auf der Festplatte erhalten bleiben.



Die Speicherung wird durch den Benutzer per Tastendruck aktiviert und durch eine andere Taste deaktiviert.

### **Dateinamen**

Die Datendateien bekommen die Endung „.DAT“.

Der Dateiname beginnt immer mit dem aktuellen Datum in der Form YYYY-MM-TT

Als nächstes folgt mit Unterstrich getrennt eine immer 4-stellige fortlaufende Nummer.

Sodann folgt durch Bindestrich „-“ getrennt eine nähere Bezeichnung der Messung. Die Bezeichnung lautet:

Für Messdaten eines Messintervalls: SyncGenOsc

Für fortlaufend gespeicherte Kennwerte: SyncGen

Beim Abspeichern schlägt das Computerprogramm automatisch den nächsten freien Dateinamen vor, der dann nur noch bestätigt werden braucht.

Optional kann der Dateiname mit Bindestrich „-“ getrennt um Details zur Messung erweitert werden.

### **Beispiele für Dateinamen**

2013-07-15\_0002-SyncGenOsc.DAT

2013-07-15\_0013-SyncGen.DAT

2013-07-15\_0001-SyncGen-Praktikumsversuch1.DAT

### **Benutzeroberfläche**

Die Benutzeroberfläche ist in mehrere Fenster aufgeteilt. Jedes Fenster nutzt den gesamten Bildschirm. Es ist ausgelegt für einen Breitbildmonitor mit der Auflösung 1920 pt x 1080 pt.

Die Fenster können „nebeneinander“ angeordnet sein, sodass man mit der Maus den Bildschirm verschieben kann, um ein anderes Fenster zu sehen.

Alternativ können die Fenster „gestapelt“ angeordnet sein, z.B. in Form von Karteikarten oder durch Auswahl in einer Menu-Struktur.

Optional wird jedes Fenster in einem eigenen Windows-Fenster dargestellt wird. Das ermöglicht es, Ausschnitte von Fenstern zu kombinieren.

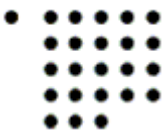
Es gibt folgende Fenster:

- Bedienung
- Einstellungen
- Kenndaten
- Oszillogramme
- Schreiberkurven

### **Fenster „Bedienung“**

Für das Fenster „Bedienung“ gelten abweichende Spezifikationen: Es nutzt nur einen schmalen Rand des Bildschirms und ist (nach Möglichkeit) immer sichtbar.





Es enthält die Bedienelemente zur Ablaufsteuerung. Dies sind:

- Start der Aufzeichnung der Kenndaten
- Stopp der Aufzeichnung der Kenndaten
- Start der Oszilloskop-Aufzeichnung
- Stopp der Oszilloskop-Aufzeichnung
- Rücksetzen der Kennwerte-Darstellung (Löschen der Diagramme und Rücksetzen der Zeit)
- Aktivieren der Speicherung von Kenndaten
- Deaktivieren der Speicherung von Kenndaten
- Speichern der aktuellen Oszilloskop-Aufzeichnung
- Eingabe und Anzeige für den Titel der aktuellen Messung
- Eingabe und Anzeige für den Namen des aktuellen Benutzers
- Anzeige, die blinkt, solange das Programm läuft.
- Anzeige, ob Kenndaten gespeichert werden.
- Anzeige, dass gerade eine Oszilloskop-Aufzeichnung gespeichert wird.

#### **Fenster „Einstellungen“**

Angaben zum Messintervall:

- Länge des Messintervalls in ms
- Anzahl der Messpunkte pro Messintervall
- Daraus resultiert die Programmierung der A/D-Wandler Karte im Zeitbereich
- Zusätzliche Latenzzeit zwischen zwei Messintervallen

*Angaben zur Datenerfassung*

- Für jeden Datenkanal der A/D-Wandler-Karte:
  - Zuordnung zum physikalischen Anschluss
  - Eingangsspannungsbereich
- Für jedes analoge Messsignal:
  - Skalierungsfaktor
  - Offset

Dabei gilt:

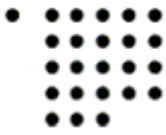
$$y = a \cdot x + b$$

mit  $a$  = Skalierungsfaktor in V oder A

$b$  = Offset in V oder A

$x$  = Digitaler Messwert der A/D-Wandler-Karte in dimensionsloser Größe

$y$  = Angezeigter Messwert in V oder A



- Aktuelle Dateinamen
- Ggf. Auswahl von Berechnungsalgorithmen

### **Fenster „Kenndaten“**

Es werden alle Kenndaten dargestellt.

Die Darstellung erfolgt als Zahlenwert und als „Zeigerinstrument“.

Darstellung als Zeigerinstrumente

Es können Kenndaten in einem Zeigerinstrument mit mehreren Zeigern zusammengefasst dargestellt werden. Jeder Kennwert muss aber einzeln als Zahlenwert dargestellt werden. Die Gruppierung erfolgt in Absprache.

Dunkelschaltung

Die Spannungen zwischen Synchronmaschine und Netz werden nicht als Zeigerinstrument dargestellt. Hier erfolgt eine spezielle Anzeige, die eine sogenannte Dunkelschaltung mit Glühlampen nachbildet: Jeder der drei Spannungen wird eine runde Farbfläche („LED-Anzeige“) zugeordnet. Die Helligkeit der Farbfläche ist proportional zum Spannungswert. Zusätzlich wird der Zahlenwert angezeigt.

Fenster „Oszillogramme“

Dieses Fenster enthält 6 (ggf. nach Absprache auch andere Anzahl) X-Y-Diagramme zum Darstellen der Messwerte eines Messintervalls.

Die horizontale Achse („X-Achse“) stellt die Zeit der Messwerte dar (nicht die Nummer des Datenpunktes!).

Jedes Diagramm kann bis zu vier Kurven gleichzeitig darstellen. In jedem Diagramm kann jede Kurve jeden beliebigen Messwert (Kap. Messsignale:) oder berechneten Wert (Kap. Für jeden Messpunkt zu berechnen:) darstellen. Die Auswahl kann während des Programmablaufs erfolgen und wird dann sofort wirksam. Die Auswahl erfolgt im selben Fenster wie die Darstellung. Die Legende im Diagramm wird entsprechend angepasst.

**Fenster „Schreiberkurven“**

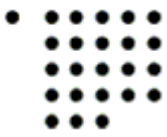
Dieses Fenster enthält 8 (ggf. nach Absprache auch andere Anzahl) X-Y-Diagramme zum Darstellen der Kennwerte der Messintervalle. Sie werden in Form einer „Schreiberkurve“ fortlaufend dargestellt. Jedes dieser Diagramme nutzt die gesamte Fensterbreite.

Die horizontale Achse („X-Achse“) stellt die Zeit dar, zu dem der jeweilige Kennwert ermittelt wurde (nicht die Nummer des Datenpunktes!). Die Zeit berechnet sich seit dem Start der Messungen oder seit dem Zeitpunkt der letzten manuellen Rücksetzung.

Jedes Diagramm kann bis zu vier Kurven gleichzeitig darstellen. In jedem Diagramm kann jede Kurve jeden beliebigen Kennwert (Kap. Für jedes Messintervall zu berechnen:) darstellen. Die Auswahl kann während des Programmablaufs erfolgen und wird dann sofort wirksam. Die Auswahl erfolgt im selben Fenster wie die Darstellung. Die Legende im Diagramm wird entsprechend angepasst.

**Formatierung von Diagrammen**

Die Formatierung der Diagramme soll so geeignet sein, dass ein Diagramm als Kopie vom Bildschirm problemlos, optisch ansprechend und gut lesbar in ein Textdokument oder Präsentation



einzuügen ist. Als Vorschlag dienen folgende Parameter:

Die Diagramme haben einen weißen Hintergrund, (RGB = 255, 255, 255), einen schwarzen Rahmen, schwarze Achsenbeschriftungen (RGB = 0, 0, 0) und hellgraue Gitternetzlinien (RGB = 193, 193, 193). Die Schriftart zur Achsenbeschriftung ist Arial oder ähnlich, die Schriftgröße mindestens 16 pt. Jedes Diagramm hat eine Überschrift, ebenfalls schwarze Schrift auf weißem Hintergrund dicht über dem Diagramm.

Die Kurven werden farbig dargestellt. Bevorzugt sind folgende Farben:

Orange (RGB = 255, 127, 0)

Dunkelblau (RGB = 0, 0, 127)

Grau (RGB = 127, 127, 127)

Dunkel-Türkis (RGB = 0, 127, 127)

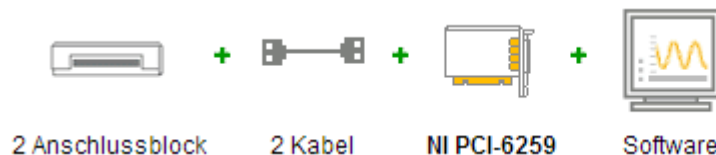
## 7.2 A2 Aufbau und Installation (Materialliste)

Aufbau

Benötigt wird zu nächst einmal das Komplettpaket für den NI PCI 6259, bestehend aus :

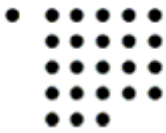
- 1 Anschlussbox
- 2 Kabel
- NI PCI – 6259
- Software von NI PCI 6259

Jedes Element muss hintereinander verschaltet werden , dies sieht nach "sine.ni.com" folgenderweise aus:



Dazu die Begründung und Information zu jedem Bauteil:

2 Anschlussblock:



Last Revised: 2013-07-23 10:45:15.0

### DAQ Accessories Selection Guide



- Shielded and unshielded cables available in many lengths
- Custom cable connector kits and mounting accessories
- Direct BNC connectivity for most products



- Shielded, unshielded, and DIN mountable terminal blocks
- Custom connectivity

### Overview

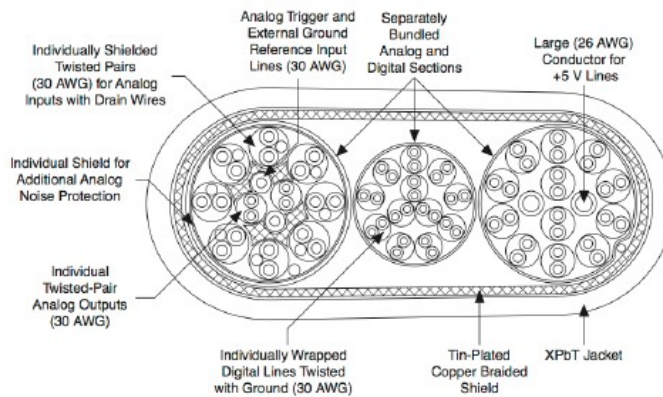
National Instruments accessories offer the flexibility you need in creating your measurement system. From shielded cables and terminal blocks to RTSI and ribbon cables, NI provides a variety of connectivity solutions including screw terminals and BNC connectors. NI shielded connector block (SCB) products feature rugged, very low-noise signal termination, and the NI terminal block (TB) and connector block (CB) families offer unshielded, low-cost alternatives.

[Back to Top](#)

### Application and Technology

#### NI Cable Design Advantages

National Instruments provides many different high-performance shielded cables that have been designed for specific NI multifunction DAQ devices. Each type of cable features improved signal integrity using the following technologies:



### Multifunction DAQ Accessories

Abbildung 7.1: Datenblatt\_1

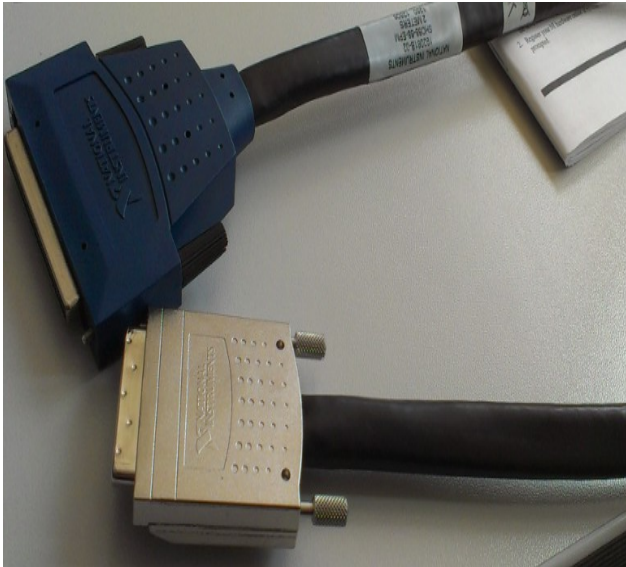
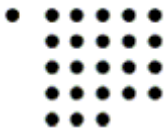


Abbildung 7.2: BNC-Buchse(Anhang)

Für die Anschlussblock wurde die BNC-2111 (siehe Abb) ausgewählt. Dieser hat 16-Single-ended-Analogeingänge. Passend zu unserem A/D-Wandler ,davon wurden 2 natürlich bestellt .  
2 wegen 32 Single-ended-Analogeingänge. Der BNC-2111 ist demnach auch kompatibel mit Labview , da dieser ebenfalls von der Firma National Instruments ist. Dieser hat einen 68-Pin-Anschluss Schnittstelle vom Typ D ,welches auch der NI PCI – 6259 besitzt. Ein weiterer Grund den BNC-2111 ist , dass dieser für die M-Serie Geräte geeignet sind. Der NI-PCI – 6259 gehört zu den M-Serie.(weiter Informationen siehe Datenblatt)

2 Kabel:

Ein Kabel wird benötigt, um Signale zwischen dem Gerät und den Signalanschlüssen des Anschlussblocks zu übertragen.



Dieses Kabel ist ein NI SHC68-68-EPM . Es besitzt eine Schnittstelle zum Anschlussblock (hier BNC-2111) und einen zur A/D-Karte(hier NI PCI 6259). Es hat die Funktion die analogen Signale vom BNC-2111 Anschlussblock auf die NI PCI 6259 A/D-Wandler-Karte zu transportieren.(weiter Informationen siehe Datenblatt)

Abbildung 7.3: NI SHC\_Kabel(Anhang)

NI PCI-6259 die AD-Karte (siehe Begründung)

Software:

Software ist erforderlich, um mit der Hardware zu kommunizieren und Daten zu sammeln, zu analysieren, darzustellen und zu speichern. Diese Karte kann mit verschiedenen Programmiersprachen eingesetzt werden, darunter LabVIEW, C/C++, Visual Basic und .NET. LabVIEW ermöglicht dabei die einfache Integration in andere NI-Hardware und wahrt somit Ihre bereits getätigten Investitionen.

"Installation der Software "NI-DAQ mx 9.5.5"":

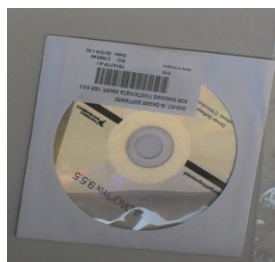
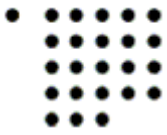


Abbildung 7.4:  
LV\_Software(Anhang)

Die Software auf der CD ist zu installieren, hierbei muss man sich zusätzlich auf der Homepage von National Instruments anmelden. Die Software bietet das "Signalverlaufserzeugung Labview



2012" ,es bietet Treiber zum Anerkennen der A/D-Karte .

"Montieren der A/D-Karte":

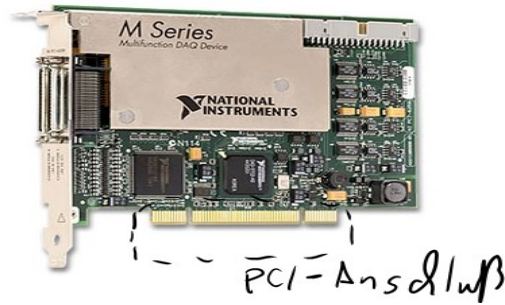
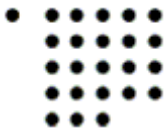


Abbildung 7.5: PCI(Anhang)

In diesem Schritt sollte der Rechner abgeschaltet sein.

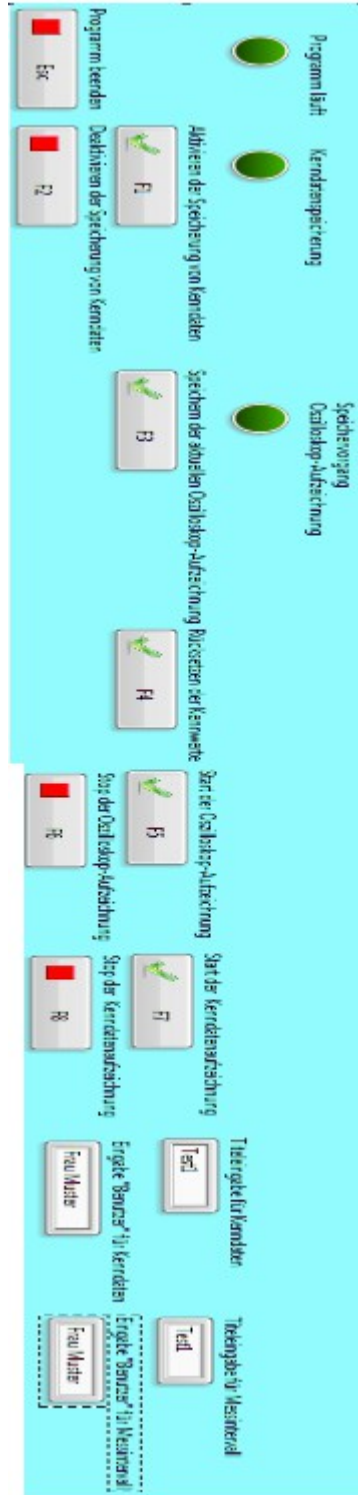
Die A/D-Karte "NI PCI-6259 " hat einen PCI-Anschluss , dieser Anschluss ist auf ein PCI-Slot im PC einzusetzen und festzuschrauben. Nachdem Einschalten ist alles installiert und montiert und man kann LabView Messsignale nun über Labview empfangen.



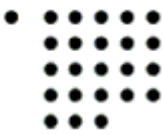
### 7.3 A3 Registerkartenfenster

#### Fenster Bedienung

Abbildung 7.6: Fenster Bedienung







Fenster Einstellungen

### Name/Beschriftung/Pfad(Datei)

**Kenndaten**

Name der Datei "Kenndaten"

Pfad der Datei "Kenndaten" (Eingabe)

C:\Users\Mustu\Desktop\Bachelorarbeit Sertac\Bachelorarbeit\Projekt in Labview\Labview\_2012\Test zur Speicherung für Ascii.dat

Pfad der Datei "Kenndaten" (Ausgabe)

C:\Users\Mustu\Desktop\Bachelorarbeit Sertac\Bachelorarbeit\Projekt in Labview\Labview\_2012\Test zur Speicherung für Ascii.dat

**Messintervall**

Name der Datei "Messintervall"

Pfad der Datei "Messintervall" (Eingabe)

C:\Users\Mustu\Desktop\Bachelorarbeit Sertac\Bachelorarbeit\Projekt in Labview\Labview\_2012\Test zur Speicherung für Ascii.dat

Pfad der Datei "Messintervall" (Ausgabe)

C:\Users\Mustu\Desktop\Bachelorarbeit Sertac\Bachelorarbeit\Projekt in Labview\Labview\_2012\Test zur Speicherung für Ascii.dat

### Eigenschaften der Oszilloskop und Kenndaten-Aufzeichnung

Speicherung "Messintervall" in Sekunde:

Instantane Latenzzeit/s:

Tmess =  /

Abtastzeit/s:

### Skalierungsfaktor und Offset

#### AC AI 0 - AI 15

Unetz 1/V AI0  Offset

Unetz 2/V AI1  Offset

Unetz 3/V AI2  Offset

Iasyn 1/A AI3  Offset

Iasyn 2/A AI4  Offset

Iasyn 3/A AI5  Offset

Usyn 1/V AI6  Offset

Usyn 2/V AI7  Offset

Usyn 3/V AI8  Offset

Isyn 1/A AI9  Offset

Isyn 2/A AI10  Offset

Isyn 3/A AI11  Offset

#### DC AI 16 - AI 31

U\_Emot/V AI16  Offset

U\_Esyn/V AI17  Offset

L\_Esyn/A AI18  Offset

L\_Emot/A AI19  Offset

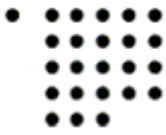
L\_Egen/V AI20  Offset

U\_Egen/V AI21  Offset

L\_amasch/A AI22  Offset

U\_amasch/V AI23  Offset

Abbildung 7.7: Fenster\_Einstellung\_Teill

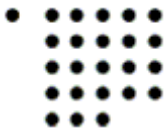


<b>Kanalzuordnung mit Eingangsspannungsbereich</b>		
<b>AC AI 0 - AI 15</b>	<b>DC AI 16 - AI 31</b>	
AI 0	Unetz 1/V AI0 [- 10V;10V]	AI 16
AI 1	Unetz 2/V AI1 [- 10V;10V]	AI 17
AI 2	Unetz 3/V AI2 [- 10V;10V]	AI 18
AI 3	Iasyn 1/A AI3 [- 10V;10V]	AI 19
AI 4	Iasyn 2/A AI4 [- 10V;10V]	AI 20
AI 5	Iasyn 3/A AI5 [- 10V;10V]	AI 21
AI 6	Usyn 1/V AI6 [- 10V;10V]	AI 22
AI 7	Usyn 2/V AI7 [- 10V;10V]	AI 23
AI 8	Usyn3/V AI8 [- 10V;10V]	
AI 9	Isyn 1/A AI9 [- 10V;10V]	
AI 10	Isyn 2/A AI10 [- 10V;10V]	
AI 11	Isyn 3/A AI11 [- 10V;10V]	

Abbildung 7.8: Einstellungen\_Fenster\_Teil2

**Fenster Kenndaten**





### Fenster Oszillosgramme

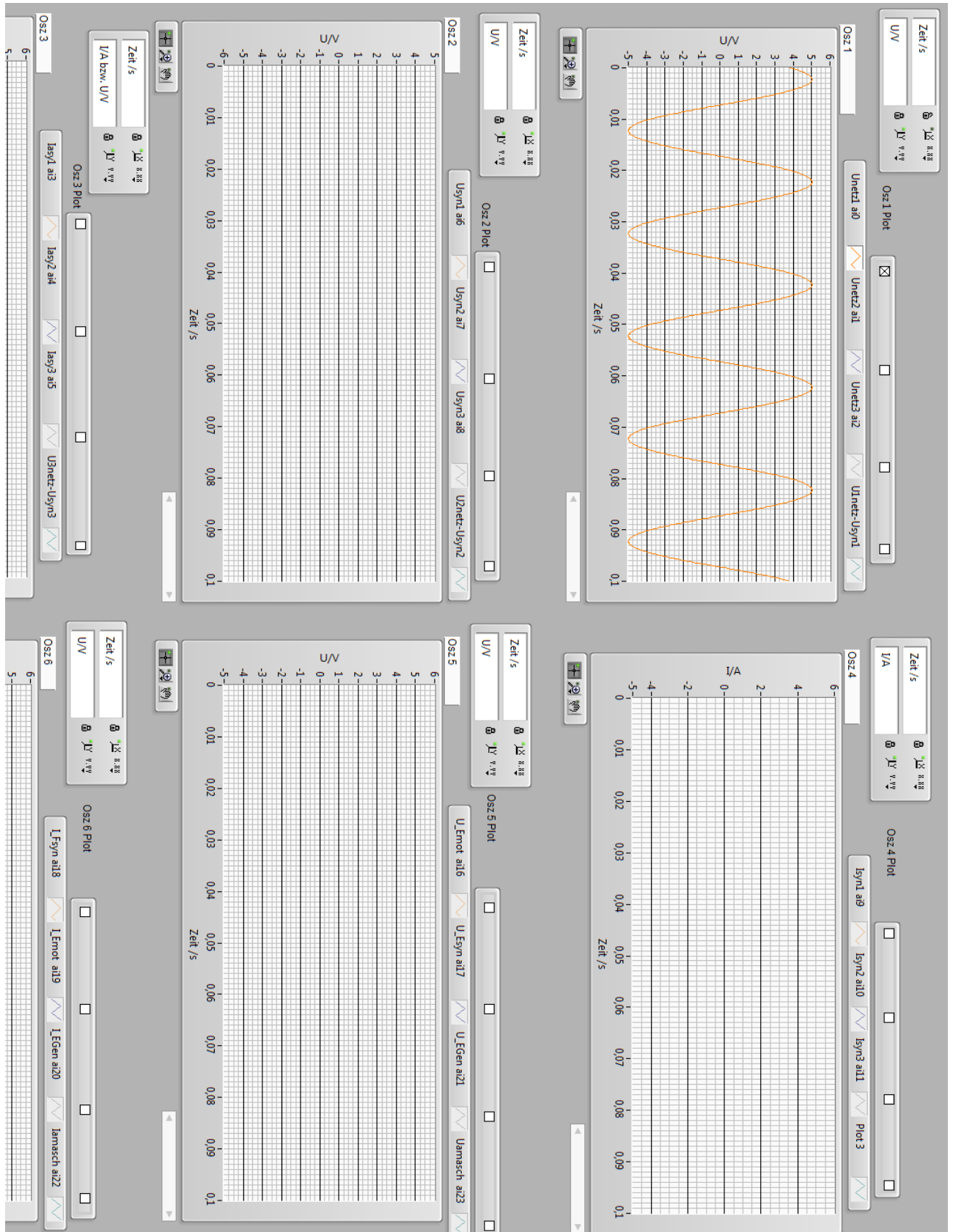


Abbildung 7.10: Osz\_Teill

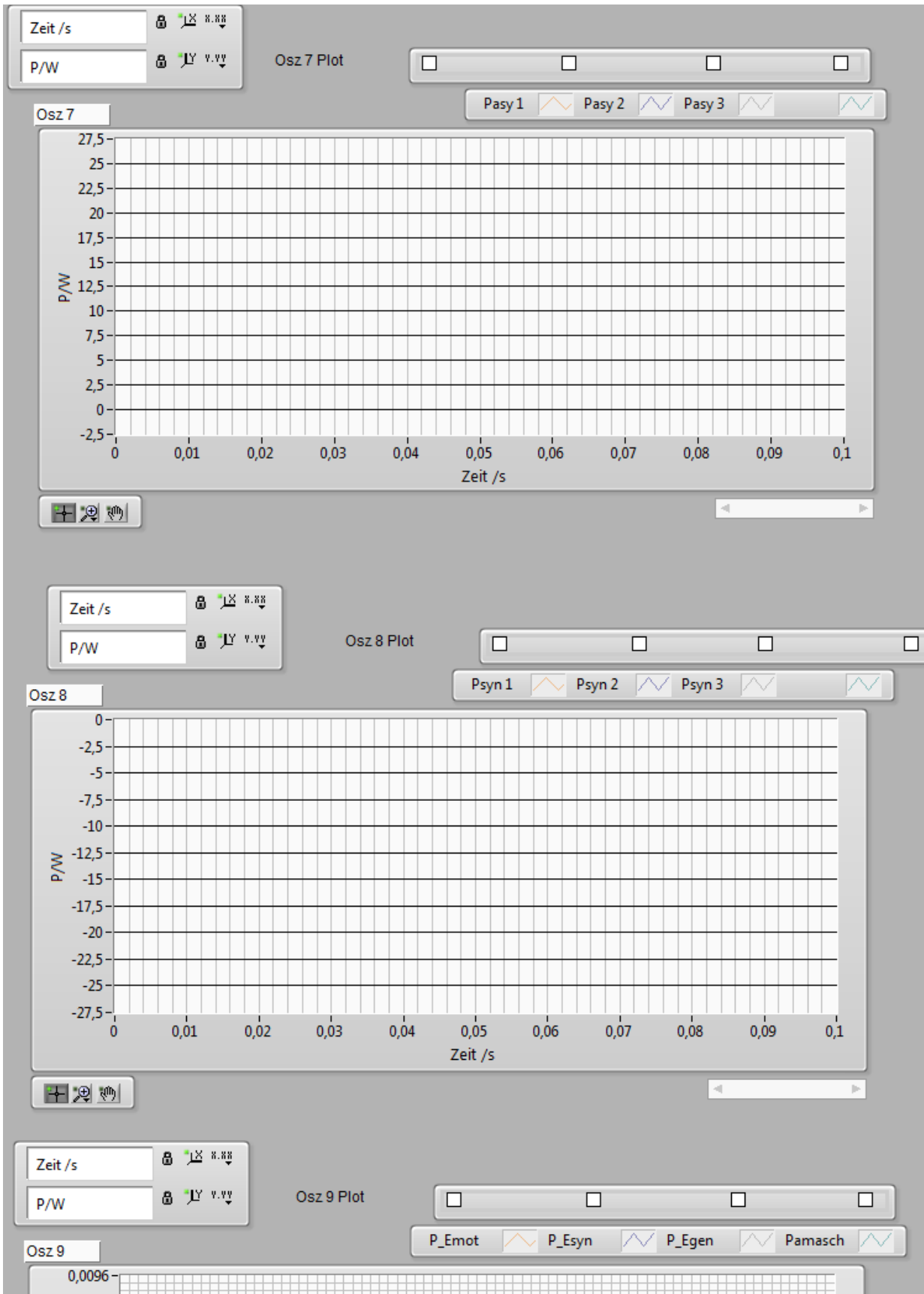
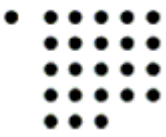
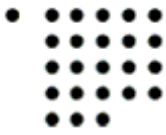


Abbildung 7.11: Osz\_Teil2



### Fenster Schreiberkurve

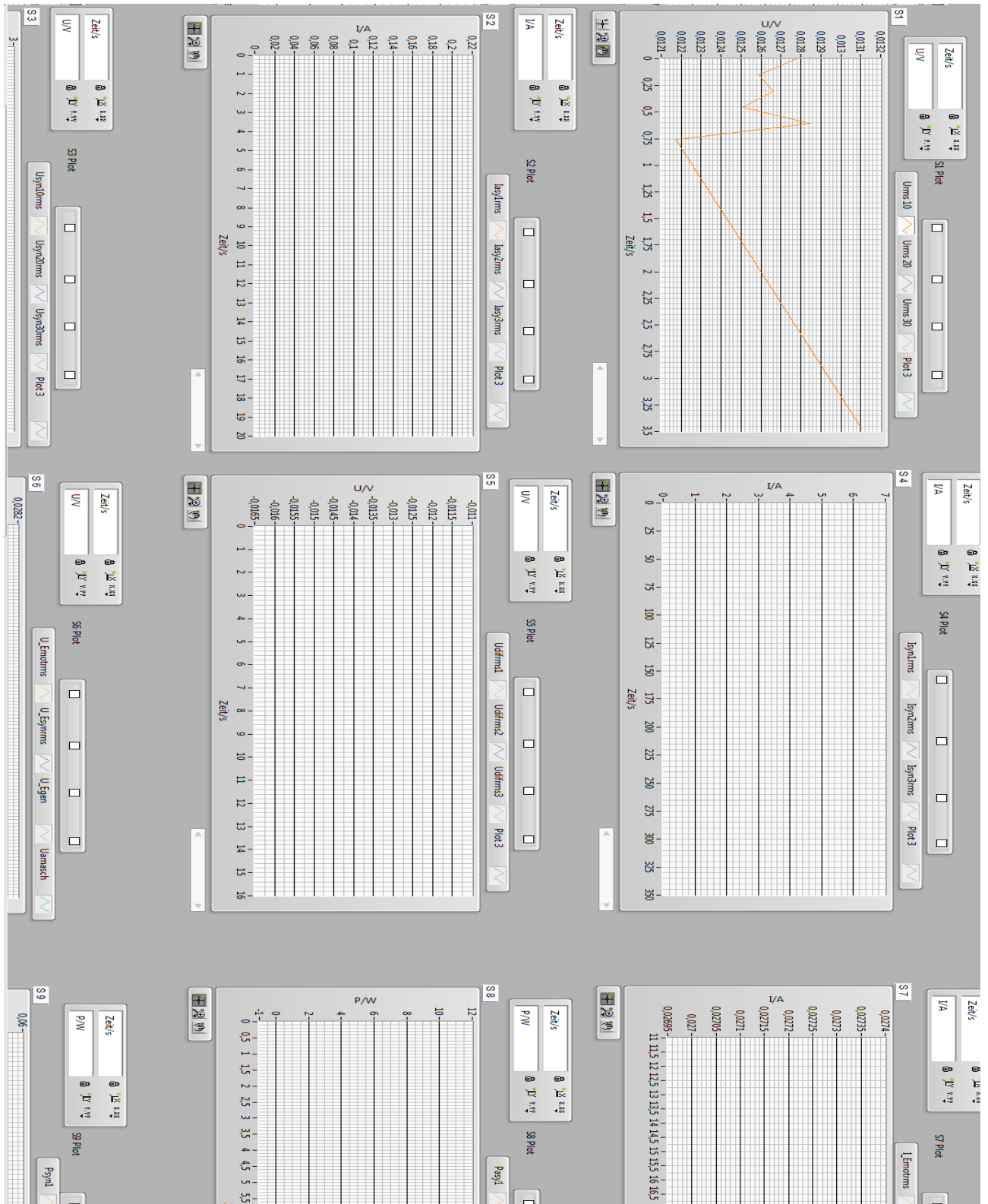


Abbildung 7.12: S-Plot Teil1

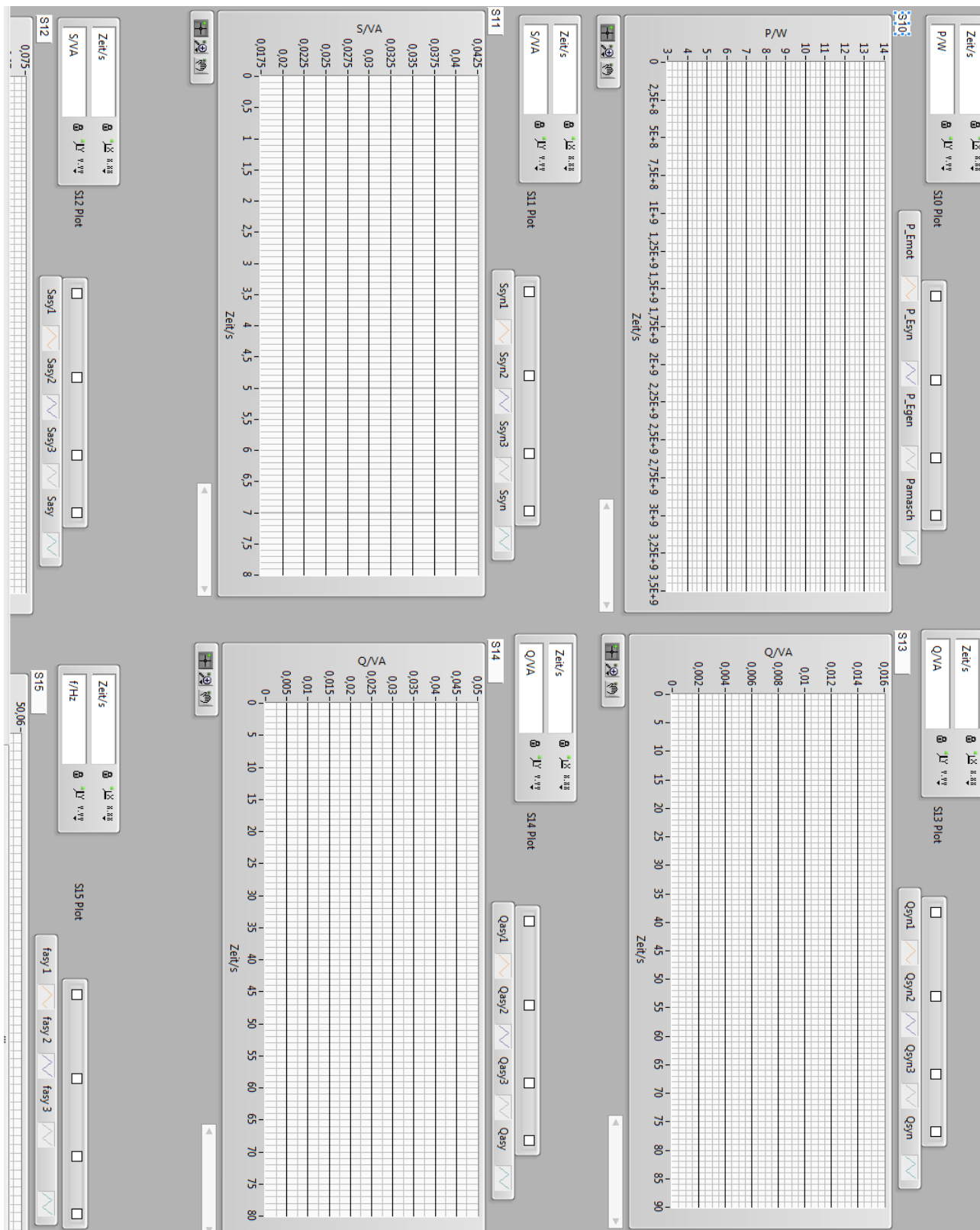


Abbildung 7.13: S-Plot Teil2

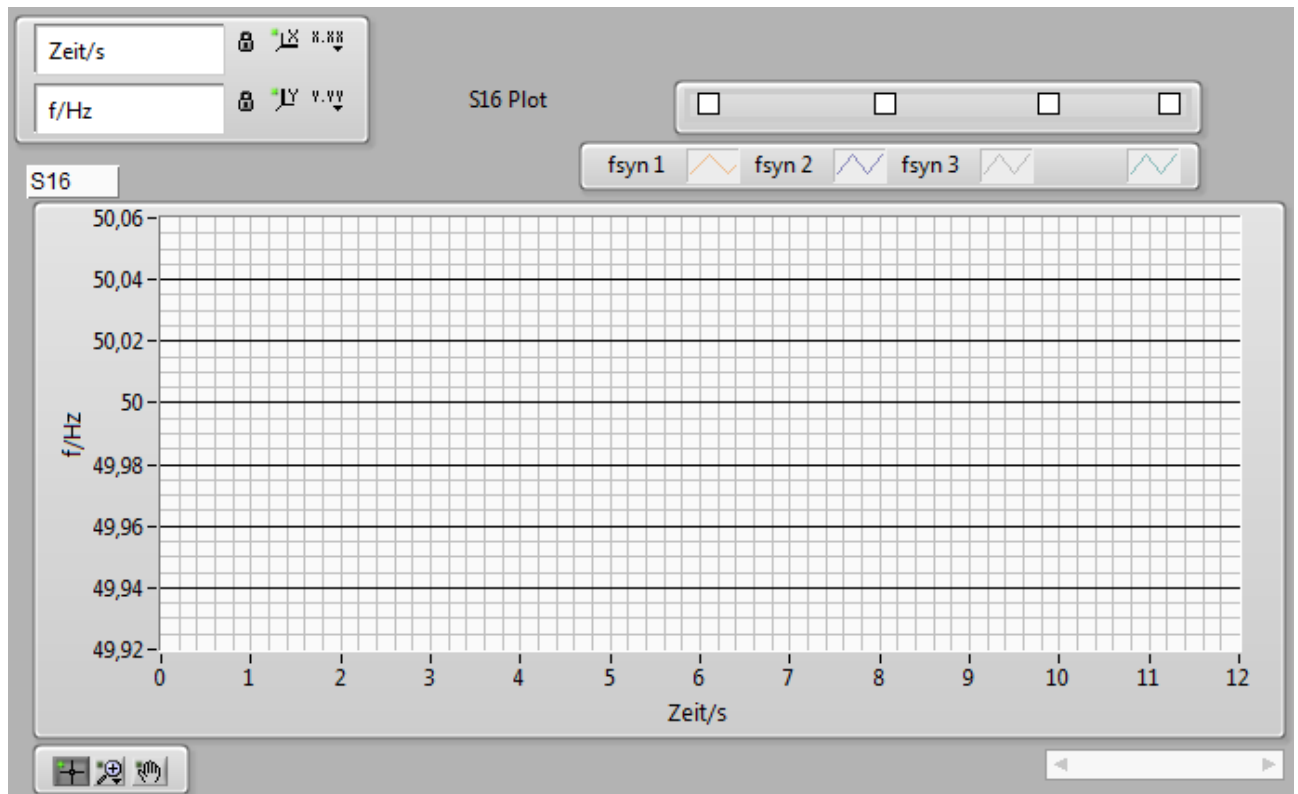
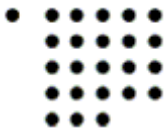


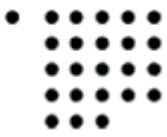
Abbildung 7.14: S-Plot Teil3

## 7.4 A4 Frequenzmessungstest

### 7.4.1 Rauschen

Ein Rauschen ist ein oder mehrere Signale mit unterschiedlichen Frequenzen, Amplituden sowie Perioden, die ein harmonisches (sauberes) Signal überlagern und damit zu einer Störgröße beim harmonischen (sauberen) Signal führen können. Diese Störsignale können äußerliche als auch innerliche Quellen haben. Äußere wie Kosmisches oder Innerliche wie Wärmerauschen, die Wärme z.B. über Leiter aufnehmen. Es ist sehr wahrscheinlich, dass bei den Generatoren und Motoren Wärmerauschfälle und andere Rauschfälle sich ereignen werden. Da diese zum Beispiel aus vielen Widerständen und induktive Leitungsbeläge besteht Hierfür wurde erst einmal ein Simulationsprogramm in Labview erstellt, um zu prüfen wie genau Labview ,trotz Rauschen, die





Grundfrequenz einer harmonischen Schwingung bestimmen kann.

## 7.4.2 Mittelwert und Standardabweichung

Um die Genauigkeit der Messung zu prüfen, ist eine Darstellung des Mittelwertes und der Standardabweichung in Abhängigkeit zur Rauschamplitude geeignet. Der Mittelwert ist eine Anzahl von Messungen, zusammenaddiert und geteilt durch die Anzahl der Messungen.

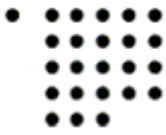
Der Mittelwert lässt sich mit folgenden Formel für die Frequenz berechnen:

$$\bar{f} = \frac{\sum_{i=0}^N f(i)}{N} \quad (7.4.2.1)$$

Mit:

$\bar{f}$  – Frequenzmittelwert

$f(i)$  – Momentanfrequenzwert



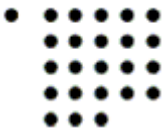
$N$  – Anzahl der Frequenzen

Die Standardabweichung ist ein Bereich ,der zufälligen Fehler, für die Streuung einzelner Werte auf einen bestimmten Mittelwert. Die Standardabweichung lässt sich mit der Formel berechnen:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=0}^N (\bar{f} - f(i))^2} \quad [14] \quad (7.4.2.2)$$

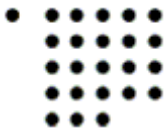
Mit:

$s$  – Standardabweichung



$\bar{f}$  – *Frequenzmittelwert*

$f(i)$  – *Momentanfrequenzwert*



$N$  – Anzahl der Frequenzen

Dieses Verfahren ist nun für bestimmte Frequenzen in Labview zu programmieren und in einem Diagramm „primäre y - Achse „Frequenzmittelwert“, Sekundäre „Standartabweichung“ und primäre x-Achse „Rauschamplitude in %“ auszugeben .

## 7.5 A5 Erläuterung zum Quellcode „Test für die Frequenzmessung“

### 7.5.1.1 Sub-VI Mittelwert

Um die Bestimmung des Mittelwertes zu realisieren wurde hier ein extra Sub-VI erstellt. Wie schon in Kapitel 7.4.2 mit der Formel 7.4.2.2 sieht die Sub-VI dann optisch so aus:

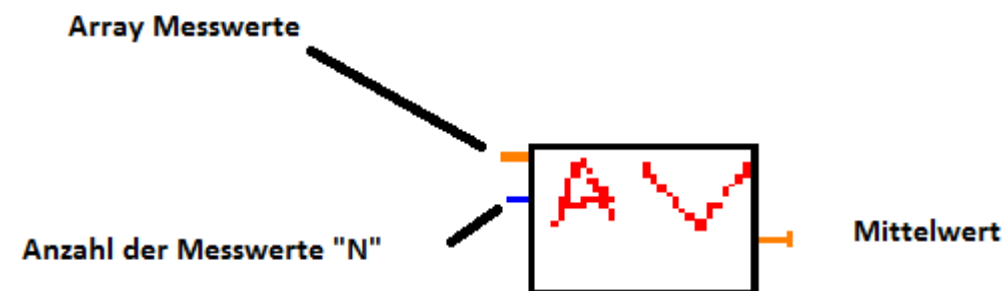


Abbildung 7.15: Sub-VI(Mittelwert)

Die zusätzliche Gatevergabe ist dann laut Tabelle:

Bezeichnung im Sub-VI	Art des Gates	Datentyp	Anzeigeformat
Mittelwert	Output	Doppelte Genauigkeit	Automatisch Formatieren
Array Messwerte	Input	Doppelte Genauigkeit	Automatisch Formatieren
Anzahl der Messungen "N"	Input	Long	Fließkomma(keine Nachkommastelle)

Für die „Anzahl der Messungen „N““ (Input) wurde der Longdatentyp verwendet, weil eine ungerade Anzahl hier nicht vorkommen wird. Bei „Array Messwerte“ (Input) bzw. „Mittelwert“ (Output) hingegen dies gut möglich ist und man auf die Genauigkeit achtet.

Allgemein:

Durch die Eingabe von „Anzahl der Messungen "N"“ (Input), „Array Messwerte“ (Input) berechnet das Sub-VI den Mittelwert über „Mittelwert“ (Output) aus.

Inhaltlich sieht das Sub-VI, welches die Werte verarbeitet, folgendermaßen aus:

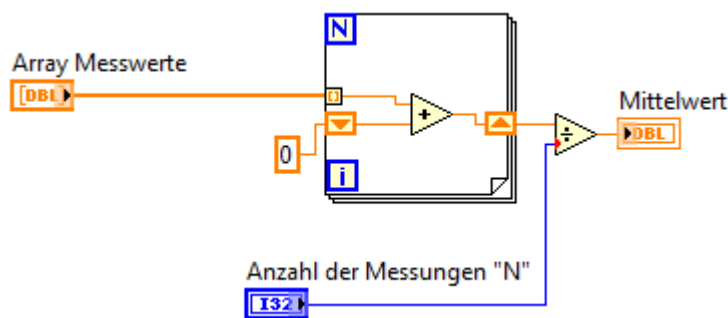


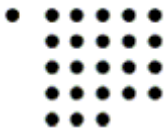
Abbildung 7.16: Sub-VI\_Blockdiagramm(Mittelwert)

Über die Eingabe „Array Messwerte“ wird „N“ in der FOR-Schleife des Arrays bestimmt durch den „Autoidentifizierten Tunnel“. Hierbei ist „N“, bei der FOR-Schleife, Anzahl der Messwerte, weiterhin ist  $i$  der instantane Wert des Arrays (der natürlich von 0 beginnt zu zählen). Dem Schieberegister wird anfangs der Wert 0 zugeführt, so dass beim ersten Durchlauf die 0, danach der letzte abgespeicherte Wert von „Array Messwerte“ und danach der nächste operierende instantane Wert zu dem letzten addiert wird usw. Die Operation erfolgt dann in der FOR-Schleife folgenderweise:

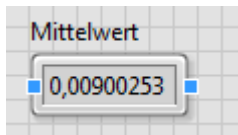
Das Array an der  $i$ 'ten-Stelle wird zusammenaddiert mit dem Inhalt des Schieberegisters (links) und als neuer Wert in den Schieberegister (rechts) eingeführt. Dies wird so lange gemacht bis alle Messwerte durchgelaufen sind. Danach wird die Summe bzw. die Ausgabe der FOR-Schleife durch die „Anzahl der Messungen „N““ geteilt. Letzten Endes erhält man die „Mittelwert“ als Output.

#### 7.5.1.1.1 Prüfen der Sub-VI AV

Um das Sub-VI nun zu prüfen wurde mit einem Funktionsgenerator (hier: HM8150) eine Sinusfunktion für Spannung generiert. Dabei hat die Spannung keine Phasenverschiebung. Mit einer



Frequenz von 50 Hz und einem Spitzen-Spitzen-Wert von 10 V wurde der Funktionsgenerator eingestellt.



Da die Messung und Eingabe nicht zu 100% genau ist, entstehen Ungenauigkeiten der Messung. Trotz dennoch ist dieses Ergebnis für die Auswertung der Messung von Leistungen ausreichend.

#### **7.5.1.2 Sub-VI Standardabweichung**

Um die Bestimmung der Standardabweichung zu realisieren wurde hier ein extra Sub-VI erstellt. Wie schon in Kapitel 7.4.2 mit der Formel 7.4.2.2 sieht die Sub-VI dann optisch so aus:

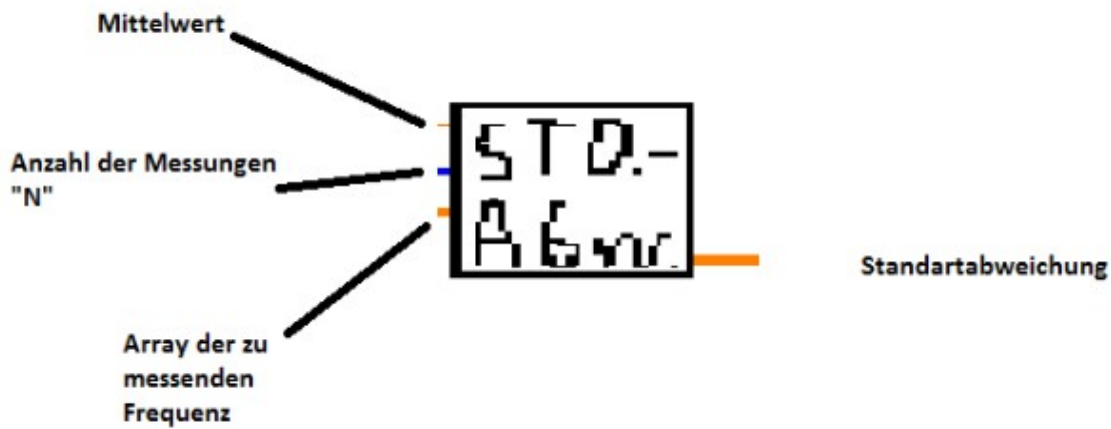
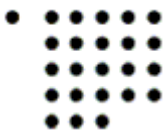


Abbildung 7.17: Sub-VI(Std-Abw.)

Die zusätzliche Gatevergabe ist dann laut Tabelle:

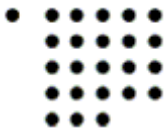
Bezeichnung im Sub-VI	Art des Gates	Datentyp	Anzeigeformat
Mittelwert	Input	Doppelte Genauigkeit	Automatisch Formatieren
Array der zu messenden Frequenz	Input	Doppelte Genauigkeit	Automatisch Formatieren
Anzahl der Messungen "N"	Input	Long	Fließkomma(keine Nachkommastele)
Std-Abweichung	Output	Doppelte Genauigkeit	Automatisch Formatieren

Tabelle 7.1: Std-Abw.

Da die „Anzahl der Messungen "N"“ keine ungenaue Anzahl hat wurde der Datentyp Long für den Gate gewählt. Der Rest bekommt die doppelte Genauigkeit für eine genauere Messung.

Allgemein:

Durch die Eingabe von „Mittelwert“(Input), „Array der zu messenden Frequenz“(Input), „Anzahl der Messungen "N"“(Input) berechnet das Sub-VI die Standartabweichung über „Std-Abweichung“(Output) aus.



In diesem Sub-VI gibt es ein zusätzliches Programm ,welches die Werte aufnimmt verarbeitet und wieder ausgibt, dies sieht dann so aus:

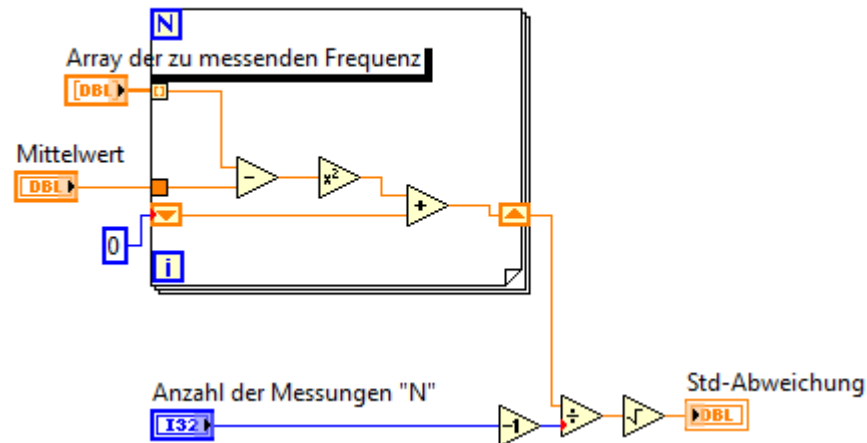


Abbildung 7.18: Sub-VI\_Blockdiagramm(std-Abw.)

Über die Eingabe „Array der zu messenden Frequenz“ wird „N“ in der FOR-Schleife des Arrays bestimmt durch den „Autoidentifizierten Tunnel“. Hierbei ist „N“ ,bei der FOR-Schleife, Anzahl der Messwerte , weiterhin ist i der instantane Wert des Arrays(der natürlich von 0 beginnt zu zählen). Dem Schieberegister wird anfangs der Wert 0 zugeführt, so dass beim ersten Durchlauf die 0, danach der letzte abgespeicherte Wert bzw. die erste Operation von „Mittelwert“ und „Array der zu messenden Frequenz“ hat und danach der nächste operierende instantane Wert zu dem letzten addiert wird usw. . Die Operation erfolgt dann in der FOR-Schleife folgenderweise:

Das Array an der i'ten-Stelle wird subtrahiert mit dem Mittelwert, quadriert und zusammenaddiert mit dem Inhalt des Schieberegisters(links) und als neuer Wert in den Schieberegister(rechts) eingeführt. Dies wird so lange gemacht bis alle Messwerte durchgelaufen sind. Danach wird die Summe bzw. die Ausgabe der FOR-Schleife durch die einmal dekrementierte „Anzahl der Messungen „N““ geteilt und davon die Wurzel gezogen. Letzten Endes erhält man die „Std-Abweichung“ als Output.



Um den Frequenzmessungstest zu verwirklichen wurde mit Hilfe der Sub-VI's in 5.1.1.2 und 5.1.1.1 ein Labviewprogramm erstellt. Diese Labview Programm sieht folgenderweise aus:

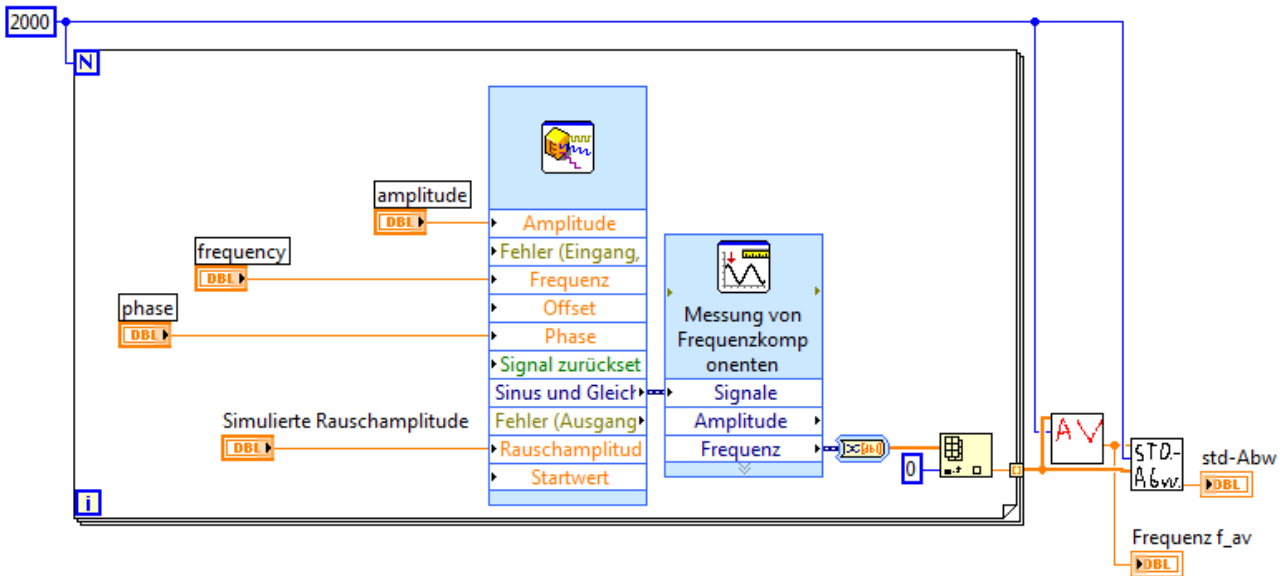


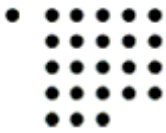
Abbildung 7.19: Frequenzmessungstest\_VI

Gatevergabe:

Bezeichnung im Sub-VI	Art des Gates	Datentyp	Anzeigeformat
amplitude	Input	Doppelte Genauigkeit	SI-Schreibweise
frequency	Input	Doppelte Genauigkeit	SI-Schreibweise
Simulierte Rauschamplitude	Input	Doppelte Genauigkeit	SI-Schreibweise
phase	Input	Doppelte Genauigkeit	SI-Schreibweise
std-Abw	Output	Doppelte Genauigkeit	Automatisch Formatieren
Frequenz f_av	Output	Doppelte Genauigkeit	Automatisch Formatieren

Tabelle 7.2: Frequenzmessungstest\_VI

Hier wurde die doppelte Genauigkeit für alle Werte genommen, wegen der genauen Messung. Für die Spannung („amplitude“) wurde 10 V ausgewählt, die Frequenz („frequency“) hatte ausnahmsweise eine ungerade Frequenz von  $f = 50,1$  Hz und Phase 0. Die Rauschamplitude wurde von 0,001 V (0,01 in Prozent) bis 10V (100 %) (je nach Abtastfrequenz) auf die harmonische Schwingung simuliert. Bei der Anzahl der Abtastung wurde 1000 im „SignalExpress“ eingetragen. Insgesamt wurden 2000 Werte für die Frequenz aufgenommen in der FOR-Schleife, bei jedem Signaldurchlauf über das „SignalExpress“-VI wurde die Frequenz mit der VI „Messung von Frequenzkomponenten“ gemessen. Anschließend wurde das erste Element (welches auch die Frequenz ist) über die Funktion „Array indizieren (Funktion)“ gefiltert und nacheinander in die Sub-VI Mittelwert (siehe Kapitel 7.5.1.1) eingeführt, auch die Anzahl der Messung (hier 2000 als



konstante) wurde eingeführt, dadurch wird der Mittelwert ermittelt und über den Mittelwert wird die Standardabweichung durch die Sub-VI Standardabweichung(siehe Kapitel 7.5.1.2) bestimmt. Dieses Programm wurde für 3 Abtastfrequenzen ausgeführt. Einmal für 10 kHz, für 33,3 kHz und 50 kHz, für die jeweiligen Abtastfrequenzen wurde dann ein Diagramm erstellt. Die Diagramme sind auf der primären y - Achse als „Frequenzmittelwert“, Sekundären „Standardabweichung“ und auf der primären x-Achse als „Rauschamplitude in %“ deklariert(Wie schon in Kapitel 7.4.2 hingewiesen).

Diese sehen dann so aus beim Prüfen :

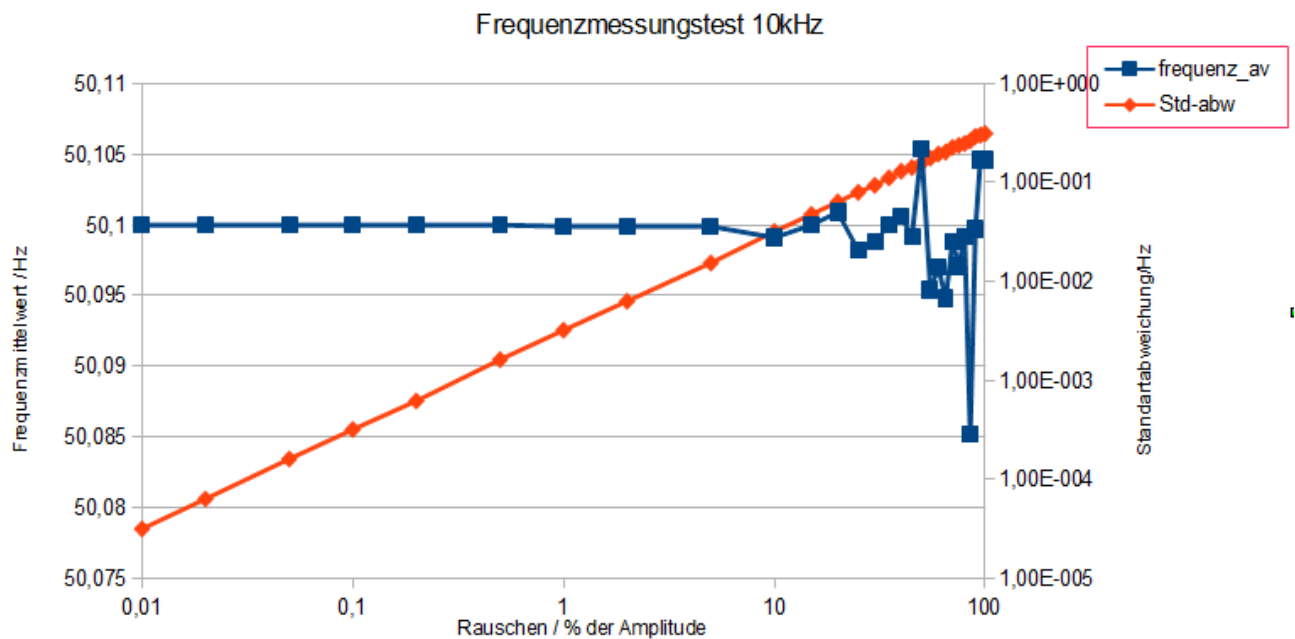


Abbildung 7.20: Diagramm\_10KHz(Abtastfrequenz)

Bei 10kHz Abtastfrequenz sieht man das bis zu 5% Rauschamplitude der Mittelwert sich wenig ändert hingegen, danach pendelt der Frequenzmittelwert zwischen 50,105 Hz und 50,085 Hz, die Standardabweichung steigt linear bis auf 0,5 Hz.

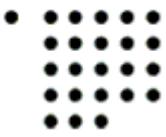
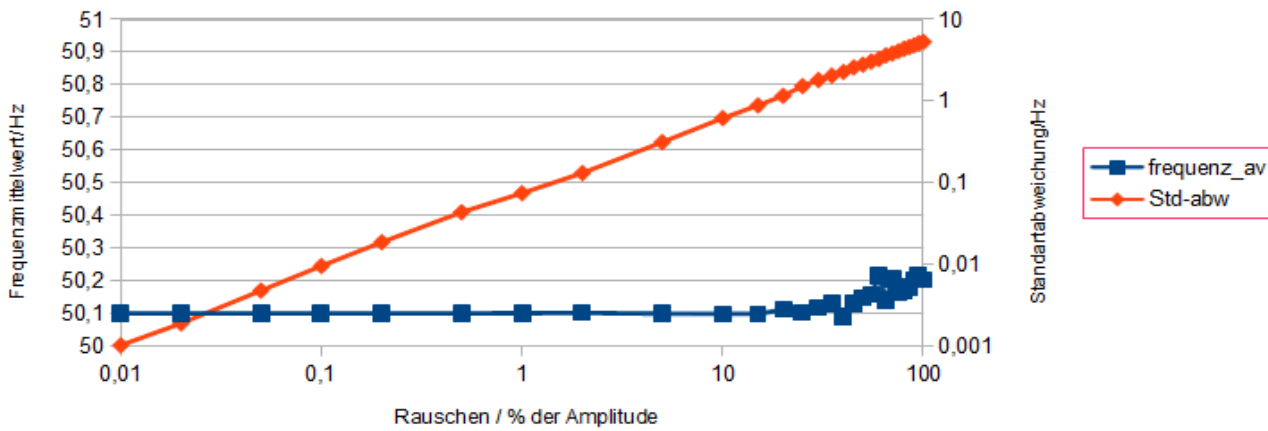


Abbildung 7.21: Diagramm\_33,3KHz(Abstastfrequenz)

Frequenzmessungstest 33,3KHz



Bis zu 10% Rauschamplitude ändert sich fast nichts an dem Frequenzmittelwert. Danach steigt der Mittelwert bis zu 50,2 Hz. Hingegen nimmt die Standardabweichung linear zu bis zum Endwert 9%.

Frequenzmessungstest 50KHz

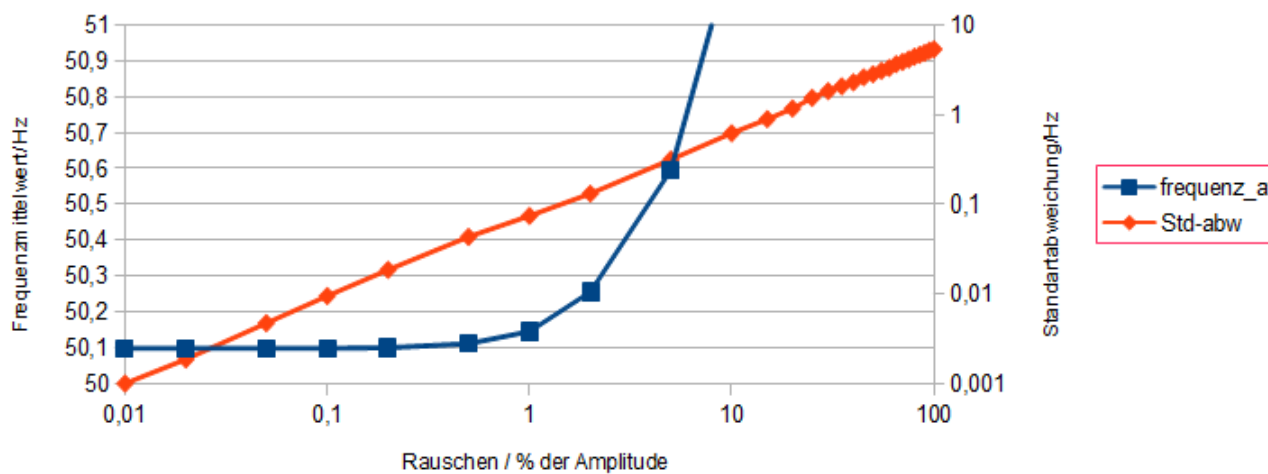
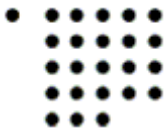


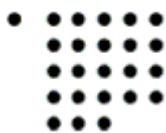
Abbildung 7.22: Diagramm\_50KHz(Abstastfrequenz)

Auch bei diesem Diagramm sieht man einen linearen Anstieg der Standardabweichung bis zu 9%. Der Frequenzmittelwert bleibt bis zu einer Rauschamplitude von 0,7 % konstant und steigt danach exponentiell, bei 10% schon über 51 Hz.



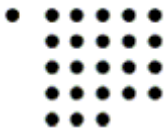
### 7.5.2 Die Folge: „Test für die Frequenzmessung“

Je mehr die Abtastfrequenz zunimmt, umso ungenauer wird die Frequenz. Da die Abtastfrequenz bei 50 KHz am Genauesten misst von den Drei. Für die Frequenzmessung sollte eine VI mit einem Algorithmus entwickelt werden, die dafür sorgt, dass trotz Rauschamplitude die genaue Frequenz mit der Rauschamplitude ermittelt wird. Leider war der Umfang des Zeitrahmens für die Arbeit nicht ausreichend. Trotz dessen ist es bedeutsam dieses Kapitel mit seinem Inhalt hier zu erwähnen. Da, wie schon in Kapitel 7.4.2 erwähnt, am Messstand Bauteile die Messwerte gegenseitig manipulieren.

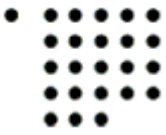


## I. Abbildungsverzeichnis

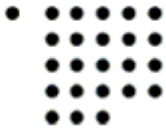
• Abbildung 4.1: Synchrongenerator mit Leonardsatz.....	4
• Abbildung 4.2: Zeigerinstrumente.....	5
• Abbildung 4.3: Aaronschaltung.....	6
• Abbildung 4.4: Gleichstrommaschine.....	8
• Abbildung 4.5: Dunkelschaltung.....	9
• Abbildung 4.6: Zeigerdiagramm.....	10
• Abbildung 4.7: Frontpanel.....	11
• Abbildung 4.8: Blockdiagramm.....	11
• Abbildung 4.9: Eingangssignal mit 1Hz und Abtastfrequenz 5 Hz.....	14
• Abbildung 4.10: Eingangssignal mit 1Hz und Abtastfrequenz 10 Hz.....	14
• Abbildung 4.11: Überarbeitete_Schaltung.....	20
• Abbildung 4.12: Messsignale.....	21
• Abbildung 5.1: Sub-VI(Leistungsermittler,Anzahl_Abtastung_N_Periode)_Schaltbild.....	36
• Abbildung 5.2: Sub-VI(Leistungsermittler,Anzahl_Abtastung_p_Periode)_Blockdiagramm.....	38
• Abbildung 5.3: Test(Sub-VI_Leistungsmessung).....	40
• Abbildung 5.4: Test(Sub-VI_Leistung).....	41
• Abbildung 5.5: Sub-VI(Effektivwert)_Schaltbild.....	41
• Abbildung 5.6: Sub-VI(Effektivwert)_Blockdiagramm.....	42
• Abbildung 5.7: SubVI(Schaltbild)_Minuswert.....	43
• Abbildung 5.8: Minuswert_Betrag.....	43
• Abbildung 5.9: Signalaufnahme.....	46
• Abbildung 5.10: Messpunkte_berechnen.....	47
• Abbildung 5.11: Effektivwerte.....	48
• Abbildung 5.12: Phasendif_berechnen.....	48
• Abbildung 5.13: Diff_eff.....	49
• Abbildung 5.14: Metadaten.....	50
• Abbildung 5.15: Parameter_Osz.....	50
• Abbildung 5.16: In String formatieren.....	51
• Abbildung 5.17: I_S_F.....	52
• Abbildung 5.18: Überschrift_Zeile[0].....	52
• Abbildung 5.19: Verarbeitung der Textdaten.....	53
• Abbildung 5.20: Zeitverarbeitung_Osz.....	53
• Abbildung 5.21: Array_Osz.....	54
• Abbildung 5.22: Kenndaten_Speicherung.....	54
• Abbildung 5.23: Zeit-Kenndaten.....	55
• Abbildung 5.24: True.....	55
• Abbildung 5.25: TRUE_SK-Zeit.....	56
• Abbildung 5.26: FALSE_SK-Zeit.....	56
• Abbildung 5.27: Speichern_Kenndaten_Dateninhalt.....	56
• Abbildung 5.28: Dateinamen(Oszilloskop-Bilder).....	57
• Abbildung 5.29: Case-False_Dateiname.....	58
• Abbildung 5.30: Registerkarte.....	58
• Abbildung 5.31: srff-Funktion.....	59



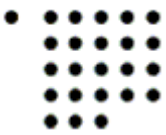
• Abbildung 5.32: srff.....	59
• Abbildung 5.33: Zustand-Programm.....	60
• Abbildung 5.34: Kenndatenspeicherung.....	60
• Abbildung 5.35: False-Kennd.....	61
• Abbildung 5.36: Speicherung_Osz.....	61
• Abbildung 5.37: Osz_LED.....	62
• Abbildung 5.38: ST_SP_Osz.....	62
• Abbildung 5.39: Kenndaten-Aufzeichnung.....	63
• Abbildung 5.40: T_mess.....	63
• Abbildung 5.41: Latenz.....	64
• Abbildung 5.42: False_Lat.....	64
• Abbildung 5.43: Ersatzkanal(AI21 U_Egen).....	64
• Abbildung 5.44: AI0_Wechsel.....	65
• Abbildung 5.45: Skalierung und Offset.....	65
• Abbildung 5.46: Cluster_Bündel_Zeiger.....	65
• Abbildung 5.47: Farbelement.....	65
• Abbildung 5.48: Oszgramme.....	66
• Abbildung 5.49: S_Kurve.....	67
• Abbildung 5.50: Dateiname_er.....	68
• Abbildung 5.51: Osziauf_teil1.....	68
• Abbildung 5.52: Osziauf_teil2.....	68
• Abbildung 5.53: S_Kurveauf.....	68
• Abbildung 5.54: Ausgabe_Einstellung.....	69
• Abbildung 5.55: Osz_Bild.....	69
• Abbildung 5.56: Kenndatenwertefundu.....	69
• Abbildung 5.57: KenndatenwertelundP.....	69
• Abbildung 5.58: SP-Kenndaten.....	70
• Abbildung 5.59: Kenndatenaufzeichnung.....	70
• Abbildung 5.60: Dunkelprog.....	71
• Abbildung 7.1: Datenblatt_1.....	86
• Abbildung 7.2: BNC-Buchse(Anhang).....	87
• Abbildung 7.3: NI SHC_Kabel(Anhang).....	88
• Abbildung 7.4: LV_Software(Anhang).....	88
• Abbildung 7.5: PCI(Anhang).....	89
• Abbildung 7.6: Fenster Bedienung.....	90
• Abbildung 7.7: Fenster_Einstellung_Teil1.....	91
• Abbildung 7.8: Einstellungen_Fenster_Teil2.....	92
• Abbildung 7.9: Fenster_Kenndaten.....	93
• Abbildung 7.10: Osz_Teil1.....	94
• Abbildung 7.11: Osz_Teil2.....	95
• Abbildung 7.12: S-Plot_Teil1.....	96
• Abbildung 7.13: S-Plot_Teil2.....	97
• Abbildung 7.14: S-Plot_Teil3.....	98
• Abbildung 7.15: Sub-VI(Mittelwert).....	102
• Abbildung 7.16: Sub-VI_Blockdiagramm(Mittelwert).....	103



• Abbildung 7.17: Sub-VI(Std-Abw.).....	105
• Abbildung 7.18: Sub-VI_Blockdiagramm(std-Abw.).....	106
• Abbildung 7.19: Frequenzmessungstest_VI.....	107
• Abbildung 7.20: Diagramm_10KHz(Abtastfrequenz).....	109
• Abbildung 7.21: Diagramm_33,3KHz(Abtastfrequenz).....	109
• Abbildung 7.22: Diagramm_50KHz(Abtastfrequenz).....	110

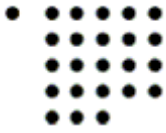


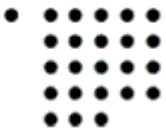




## II. Tabellenverzeichnis

Tabelle 3.1: Materialliste.....	3
Tabelle 4.1: Echtzeitbetriebssystem.....	12
Tabelle 4.2: dynamische_Spez.....	22
Tabelle 4.3: Berechnung Messpunkt.....	23
Tabelle 4.4: Effektivwerte.....	24
Tabelle 4.5: Wirkleistungen.....	25
Tabelle 4.6: Scheinleistungen.....	25
Tabelle 4.7: Blindleistungen.....	26
Tabelle 4.8: Zeitabhängige.....	26
Tabelle 5.1: Auflistung benutzter Express-VI's und VI Funktionen.....	36
Tabelle 5.2: Sub-VI(Leistungsermittler,Anzahl_Abtastung_p_Periode)_Tabelle.....	37
Tabelle 5.3: Sub-VI(Leistungsermittler,Anzahl_Abtastung_p_Periode)_Transponieren1.....	38
Tabelle 5.4: Sub-VI(Leistungsermittler,Anzahl_Abtastung_p_Periode)_Transponieren2.....	38
Tabelle 5.5: Sub-VI(Leistungsermittler,Anzahl_Abtastung_p_Periode)_Arraycut1.....	39
Tabelle 5.6: Sub-VI(Leistungsermittler,Anzahl_Abtastung_p_Periode)_Arraycut2.....	39
Tabelle 5.7: Sub-VI(Effektivwert)_Tabelle.....	41
Tabelle 5.8: S-R_flipflop.....	58
Tabelle 7.1: Std-Abw.....	105
Tabelle 7.2: Frequenzmessungstest_VI.....	107





### III. Literaturverzeichnis

- [1] National Instruments,NI-PCI 6259,<http://sine.ni.com/nips/cds/view/p/lang/de/nid/14128>,S.1,  
(22.08.2013)
- [2] National Instruments,NI BNC-2111,<http://sine.ni.com/nips/cds/view/p/lang/de/nid/201731>,S.1,  
(22.08.2013)
- [3] National Instruments,NI SHC68-68-EPM,National Instruments,S.1,(22.08.2013)
- [4] HAMEG,PROGRAMMABLE FUNCTION GENERATOR  
HM8150,<http://www.hameg.com/0.327.0.html>,S.1,(22.08.2013)
- [5] Wolfgang Georgi,Ergun Metin, Einführung in LabVIEW,2. Auflage,2007,S.197
- [6] Otto Von Guericke Universität Magdeburg,Abtastfrequenz,[http://www.uni-magdeburg.de/isut/LSS/Lehre/Skripte\\_Messtechnik/8\\_DG\\_SV.pdf](http://www.uni-magdeburg.de/isut/LSS/Lehre/Skripte_Messtechnik/8_DG_SV.pdf),S.10-12,(19.8.2013)
- [7] Heinrich Frohne,Karl-Heinz Löcherer,Hans Müller,Thomas Harriehausen,Dieter Schwarzenau,  
Moeller Grundlagen der Elektrotechnik,22. Auflage,2011,S.324
- [8] Uwe Siart,Wirkleistung, Blindleistung und  
Effektivwert,<http://www.siart.de/lehre/leistung.pdf>,S.5,(19.08.2013)
- [9] Heinrich Frohne,Karl-Heinz Löcherer,Hans Müller,Thomas Harriehausen,Dieter Schwarzenau,  
Moeller Grundlagen der Elektrotechnik,22. Auflage,2011,S.325
- [10] Dieter, Zastrow, The OpenOffice.org 2 Guidebook,16. Auflage,2006,S.345-346
- [11] Gert Hagmann, Aufgabensammlung zu den Grundlagen der Elektrotechnik,15.  
Auflage,2012,S.205-207
- [12] Gert Hagmann, Aufgabensammlung zu den Grundlagen der Elektrotechnik,15.  
Auflage,2012,S.205-207
- [13] Layland Trucks,S-  
R\_Flipflop,<http://zone.ni.com/devzone/cda/epd/p/id/2726#0requirements>,S.1,(07.09.2013)
- [14] Prof. Dr.-Ing. Martin Bantel, Messgeräte-Praxis: Funktion und Einsatz moderner  
Messgeräte,1. Auflage,2004,S.24-25