

PROTOKOLL ZU DER VERANSTALTUNG

EINFÜHRUNG IN DAS RECHNERGESTÜTZTE EXPERIMENTIEREN

David Koke(d_koke01@uni-muenster.de)

Alex Oster(a_oste16@uni-muenster.de)

im Zeitraum vom 03. bis 06.09.2018

betreut von Jürgen Berkemeier

11. September 2018

Inhaltsverzeichnis

1	Einführung	1
2	EIRE: Tag 1	2
3	Messstruktur für Analog-Digital-Wandler	3
3.1	DAQ-Assistent	3
3.2	Programmieren einer eigenen Messstruktur	4
3.3	Abtastung mit der Messstruktur	9
4	Modulation	15
4.1	Amplitudenmodulation	15
4.2	Erweiterung der Messstruktur	19
5	Anhang	20

1 Einführung

Dieser Bericht dient der Auswertung und Erläuterung der Versuchsergebnisse des Praktikums zum Thema „Einführung in das rechnergestützte Experimentieren“, durchgeführt im Zeitraum vom 03. bis 06.09.2018, betreut von Dr. Jürgen Berkemeier.

Ziel des Praktikums war die Vermittlung elementarer Kenntnisse zur rechnergestützten Erfassung und Verarbeitung von Messwerten. Dazu wurde das grafische Programmiersystem LabVIEW (Entwickelt von National Instruments, Akronym für „Laboratory Virtual Instrumentation Engineering Workbench“ (**Quelle noch einfügen**)) verwendet, welches wegen der leichten Erlernbarkeit ideal für einen Einstieg in die Thematik geeignet ist.

Das Kernelement des Praktikums bestand darin, mit einem Funktionsgenerator erstellte Signale zu digitalisieren und mit Hilfe von LabVIEW in Echtzeit zu verarbeiten. Dazu wurden verschiedene Programme erstellt.

Zum Erlernen der grundlegenden Funktionsweise der Programmierungsumgebung wurde ein Programm erstellt, mit welchem das Produkt zweier Zahlen sowie die Fakultät einer Zahl errechnet werden konnte. Anschließend wurde ein Programm zur Ausgabe eines sinusförmigen Signals geschrieben, welches für spätere Versuchsteile als Unterprogramm verwendet wurde. Im weiteren Verlauf des Praktikums wurde ein Programm erstellt, das mit dem Funktionsgenerator kreierte und mittels einer ADC/DAC-Box erfassten Signale grafisch darstellte und speicherte. Weiterhin wurden Programme zur Fouriertransformation, Modulation und Demodulation der Signale (Verwendung von Amplituden-, Phasen- und Frequenzmodulationsverfahren) erstellt.

2 EIRE: Tag 1

3 Messstruktur für Analog-Digital-Wandler

Nach der Einführung in die grundlegenden Funktionen von LabView sollte an dem zweiten Tag eine Messstruktur programmiert werden. Dazu wurde den Studenten jeweils ein Funktionsgenerator und ein Analog-Digital-Wandler zur Verfügung gestellt. Über diesen Wandler ließ sich der Funktionsgenerator an den Computer schließen. Da es sich hierbei um einen ADC von NationalInstruments handelte, erkannte LabView diesen sofort.

3.1 DAQ-Assistent

Mit dem vorprogrammierten Express-VI, dem DAQ-Assistent (siehe Abb. 1), konnte gemessen werden. Lediglich die Anzahl der Datenpunkte und die Messrate mussten dem

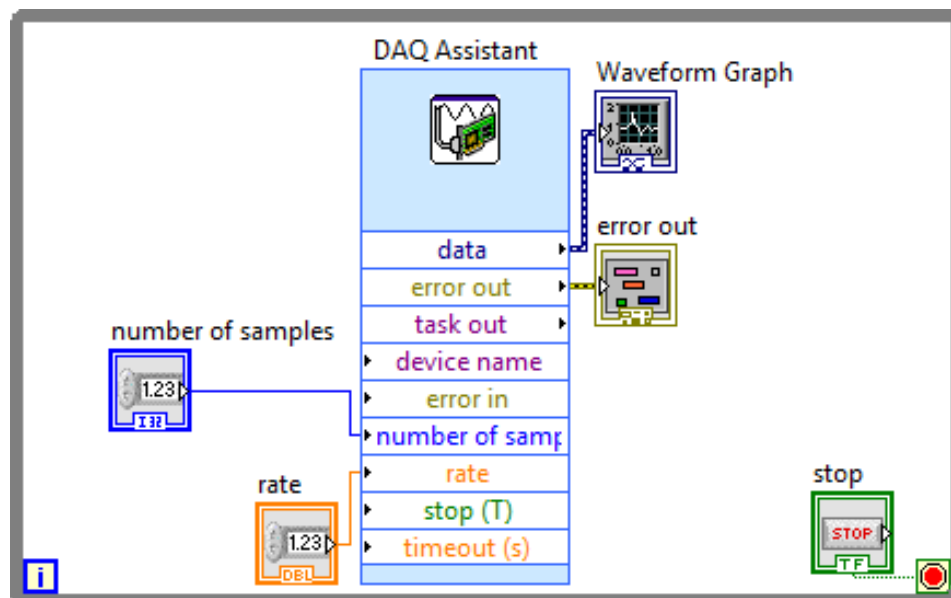


Abbildung 1: Diese Abbildung stellt das Express-VI „DAQ-Assistent“ dar. Darauf sind die verschiedenen Ein- und Ausgänge zu sehen.

DAQ-Assistent geliefert werden damit ein Waveform-Graph aus dem Array, welches der „data“-Ausgang ausgab, das Bild in Abb. 2 erzeugt werden konnte. Das aufgezeichnete Signal stimmte mit dem überein, was an dem Funktionsgenerator eingestellt wurde.

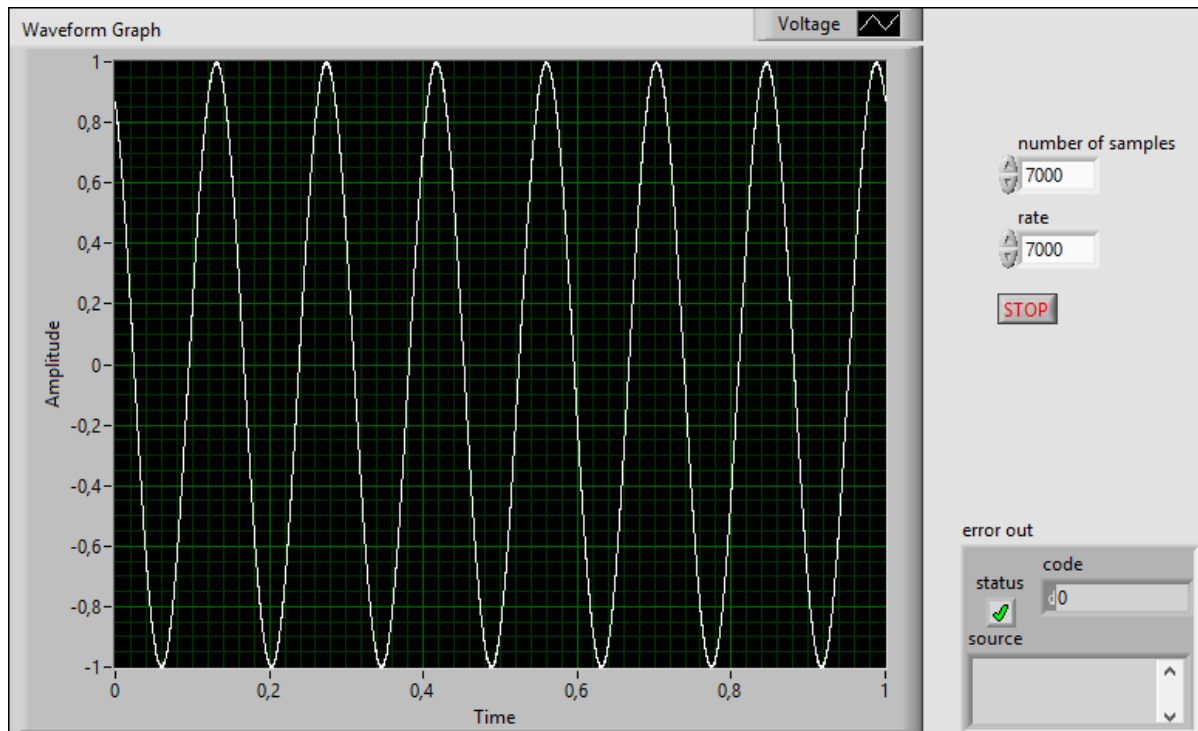


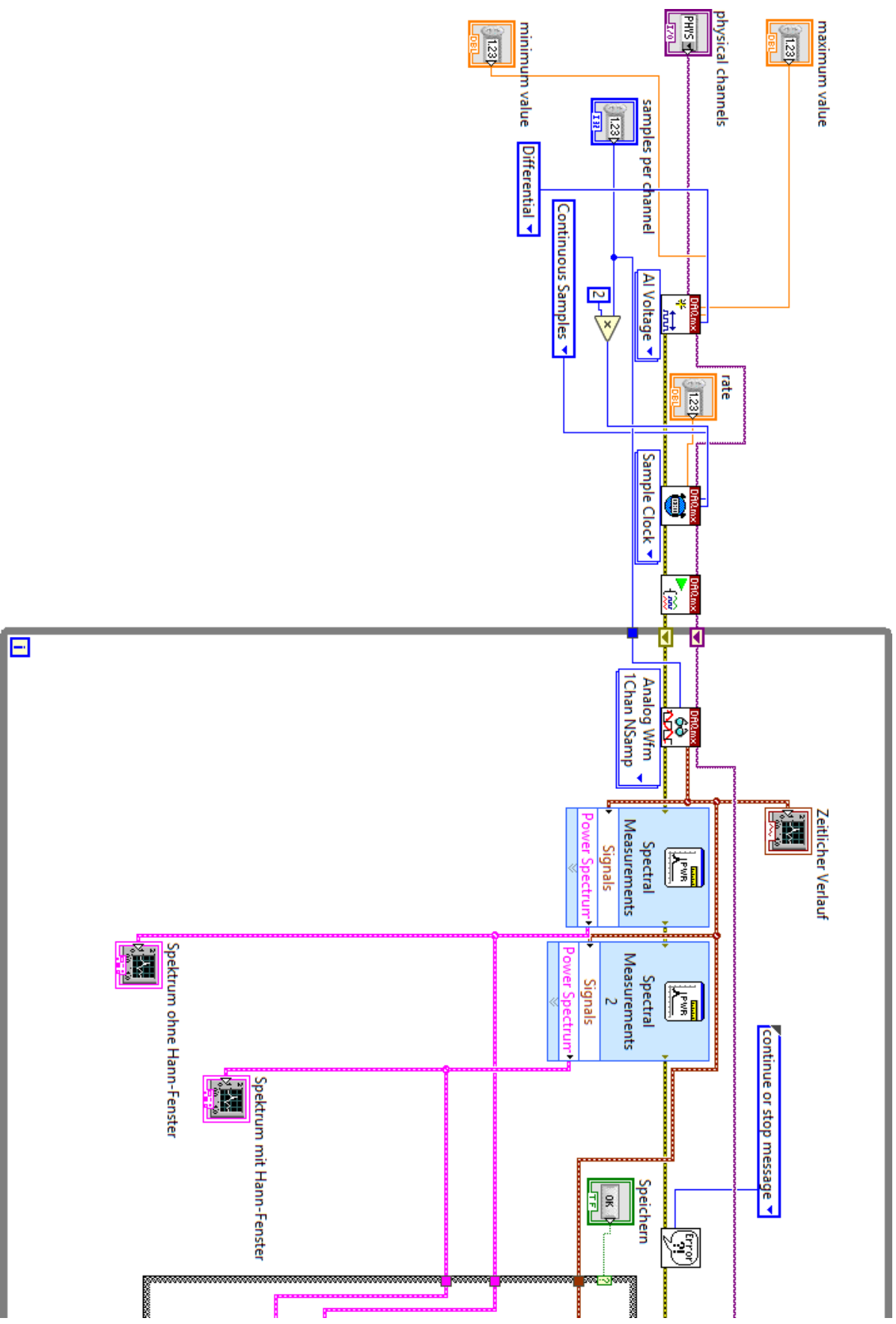
Abbildung 2: Diese Abbildung stellt das Frontpanel eines VIs mit dem DAQ-Assistent und Controls für Messrate und Anzahl der Datenpunkte dar.

3.2 Programmieren einer eigenen Messstruktur

Da lediglich eine Anzeige des gemessenen Signals oft nicht reicht, musste eine Messstruktur programmiert werden, die den Umfang des DAQ-Assistants übersteigt und weiter manuell angepasst werden kann.

Aufgrund von Problemen bei Wartungsarbeiten ließen sich die Computer jedoch gegen Mittag nicht mehr bedienen, weswegen das Fertigstellen des Programms auf den dritten Tag verschoben wurde. Zur Vorbereitung für das Weiterarbeiten wurden thematisch noch die Fouriertransformation zur Spektralanalyse, so wie das Abtasttheorem für eine richtige Messung angesprochen.

Zu Beginn des dritten Tages wurde die Arbeit an der Messstruktur fortgesetzt. Abb. 3 stellt das Endprodukt für diesen Aufgabenteil dar.



werden sollten wurden „AI Voltage“ und „Differential“ als Modus bzw. konstanter Eingang an das DAQmx-VI gelegt. Um die gemessene Spannungen variabel zu begrenzen wurden zusätzlich Controls angebracht. Die Benennung dieser mit „minimum value“ und „maximum value“ entstammt der voreingestellten Namen der Eingänge des DAQmx-VIs. In einer späteren Version wurden die englischen Namen durch deutsche ersetzt. Von den Ausgängen dieses VIs gehen nur der Signalkanal und ein Fehlerkanal durch, welche sich von hier aus bis ans Ende der Messstruktur durch alle DAQmx-VIs ziehen.

Das zweite DAQmx-VI dient als Taktgeber für die Aufnahme des Signals, deswegen der Modus „Sample Clock“. Ihm werden einerseits der Signalkanal und der Fehler übergeben, andererseits auch die Messrate „rate“ und die doppelte Anzahl der Datenpunkte „samples per channel“, damit der CPU weniger ausgelastet wird. Mit dem letzten und konstanten Eingang „Continuous Samples“ wird festgelegt, wie die Daten aufgenommen werden. Da der Funktionsgenerator kontinuierlich ein Signal ausgibt, soll auch so lange getaktet werden, bis der Nutzer das Programm beendet.

Das nächste DAQmx-VI ist nur für den Start des Messvorgangs zuständig. Von dort aus geht es in eine while-Schleife mit Stopp-Knopf, damit das Programm jederzeit von dem Benutzer beendet werden kann, ohne dass ein Stopp von Labview erzwungen werden muss. In dieser while-Schleife ist das erste Element ein weiteres DAQmx-VI. Dieses ist für das Lesen des Signals mit den vorgegebenen Parametern des Taktgebers verantwortlich. Es gibt ein dynamisches Array der Messdaten aus, welches mit Hilfe eines Waveform-Graphen auf der Frontplatte dargestellt werden kann. Des Weiteren wird die Ausgabe für eine Spektralanalyse über zwei verschiedene Express-VIs genutzt. Dabei wird bei einem mit dem sogenannten Hann-Fenster gearbeitet und bei dem anderen ohne. Das Hann-Fenster dient zur Manipulation der Messdaten, um die vorkommenden nicht ganzzahligen Frequenzen schärfer im Frequenzbild darzustellen, um den Leakage-Effekt zu unterdrücken, auf den später noch genauer eingegangen wird.

Neben den Daten verläuft auch das Fehlerkabel durch die Express-VIs. Falls ein Fehler in dem Programm auftritt dann wird aufgrund des VIs mit der Sprechblase eine Fehlermeldung ausgegeben und der Nutzer wird gefragt, ob das Programm weiterlaufen oder gestoppt werden soll.

Um die ermittelten Daten für das Zeitbild und das Frequenzbild zu speichern wird die Case-Struktur nach der Spektralanalyse eingebaut. Wird auf den Speicherknopf auf dem Frontpanel gedrückt, so wird der in Abb. 3 dargestellte Fall durchgeführt. Andernfalls

der in Abb. 4 dargestellte Fall, in dem das Programm einfach weiterläuft.

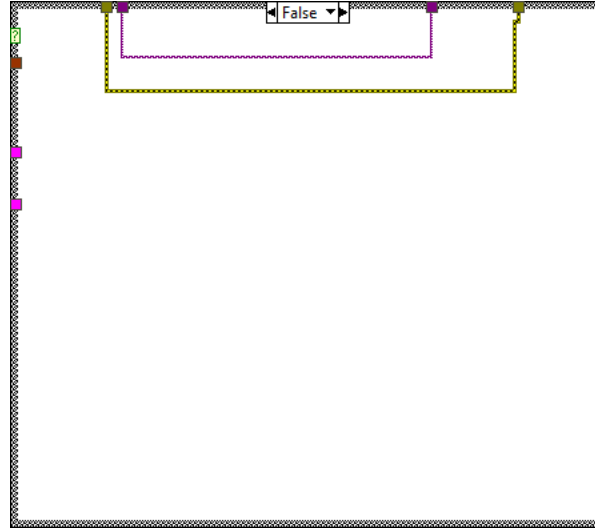


Abbildung 4: Case der Messstruktur VI für den Fall, dass nicht gespeichert werden soll.

Im Falle des Speicherns hingegen wird das Lesen der Daten über ein DAQmx-VI angehalten, damit überhaupt ein fester Datensatz gespeichert werden kann. Um die Daten zu erhalten werden das Signal des Zeitbildes und der Frequenzbilder in Waveform-Builder gegeben um die einzelnen Parameter zu erhalten. Damit die Zeit t_i und die Frequenz f_i dargestellt werden können, werden die Differentiale dt und ds über for-Schleifen i mal aufaddiert. Aus diesen Zeiten und Frequenzen werden mit ihren jeweiligen Amplituden (V im Zeitbild und dB im Frequenzbild, mit und ohne Hann-fenster) in ein Array gegeben und in ein VI gegeben, welches eine Datei ausgibt, in dem das Array mit i Zeilen und den fünf Spalten für die einzelnen Parametern ausgegeben wird. An diesem VI lässt sich auch das Format der Daten darstellen, hier wurde „%.5e“ gewählt für eine Ausgabe wie in Tab. 1 für die exponentielle Schreibweise und fünf Nachkommastellen. Eine automatische Benennung der Spalten ist hierbei nicht möglich, weswegen verfolgt werden muss welcher Wert an welcher Stelle in das Array eingefügt wird.

Danach wird die Schleife wiederholt bis sie durch Betätigen des Stopp-Knopfes verlassen wird. Nach zwei weiteren DAQmx-VIs wird das Programm dann beendet.

0,00000E+0	0,00000E+0	1,21707E+0	-4,15539E+1	-5,78080E+1
1,00000E-3	1,00000E+0	-6,34577E-1	-3,99954E+1	-6,09291E+1
2,00000E-3	2,00000E+0	-2,24044E+0	-4,00174E+1	-9,43314E+1
3,00000E-3	3,00000E+0	-2,98161E+0	-4,00395E+1	-8,78620E+1
:				
:				

Tabelle 1: Beispielausgabe nach Nutzung der Speicherfunktion der Messstruktur.

3.3 Abtastung mit der Messstruktur

Mit der vorerst fertigen Messstruktur sollte nun abgetastet werden. Die Abbildungen 5 bis 8 stellen verschiedene Fälle dar. Der Verlauf in Abb. 5 stellt den Fall der Unterabtastung dar. Hierbei entspricht die Frequenz der Messung mit der Signalfrequenz des Funktionsgenerator überein. Um das Abtasttheorem zu erfüllen, müsste jedoch mit mindestens der doppelten Frequenz abgetastet werden, weswegen hierbei keine Schwingung sondern nur ein annähernd linearer Verlauf im Zeitbild zu erkennen ist. Im Falle einer Messung mit exakt gleicher Frequenz wäre jedoch ein konstanter Verlauf zu erwarten, dies schließt auf einen Unterschied zwischen der Frequenz des Funktionsgenerators und der Messrate, obwohl beide auf den selben Wert eingestellt waren.

Abb. 6 verdeutlicht die Ungenauigkeit bei der Messung weiter. Hier war der Funktionsgenerator auf 5 Hz eingestellt, gemessen wurde jedoch ein breites Spektrum an Frequenzen, bei denen die am stärksten vorkommenden Frequenzen zwischen 5 Hz bis 6 Hz liegen.

Insbesondere die Darstellung im Frequenzbild von Signalen mit nicht ganzzahligen Frequenzen stellt bei vorliegender Messtechnik ein Problem dar. Vergleicht man die Abbildungen 7 und 8, so ist bereits im Zeitbild zu erkennen, dass im nicht ganzzahligen Fall, bei einer Frequenz, die lediglich 0,5 Hz über der anderen liegt ein deutlich anderes Bild zu sehen ist, obwohl es sich bei beiden um einfache Sinusschwingungen mit jeweils nur eine Frequenz handeln sollte. Aufgrund der groß gewählten Frequenzen ist die Sinusform in Abb. 7 kaum zu erkennen, jedoch lässt sich anhand der Amplituden ein Muster erkennen, welches einem einfachen Sinus deutlich mehr ähnelt als das in Abb. 8.

Betrachtet man nun die Frequenzbilder, so fällt zunächst im Fall ohne Hann-Fenster auf, dass bei dem 250 Hz-Signal eine Frequenz noch recht deutlich heraus sticht. Bei dem 250,5 Hz-Signal weitet sich dieser Frequenzpeak jedoch auf und die Frequenz des

Signals lässt sich somit deutlich schlechter erfassen. Dies nennt man Leakage-Effekt. Um dieses „Ausschmieren“ zu verhindern, kann das Signal mit Hilfe einer sogenannten Fensterfunktion manipuliert werden. In den Abbildungen werden hier Hann-Fenster verwendet. Vergleicht man die Frequenzbilder mit Hann-Fenster in den Abbildungen 7 und 8 so fällt auf, dass der Leakage-Effekt deutlich geschwächt auftritt, dafür aber verstärktes Rauschen in den umliegenden Frequenzbereichen zu erkennen ist. Dieses Rauschen jedoch liegt weit unter dem gewünschten Frequenzpeak bei ca. -80 dB und ist demnach zu vernachlässigen.

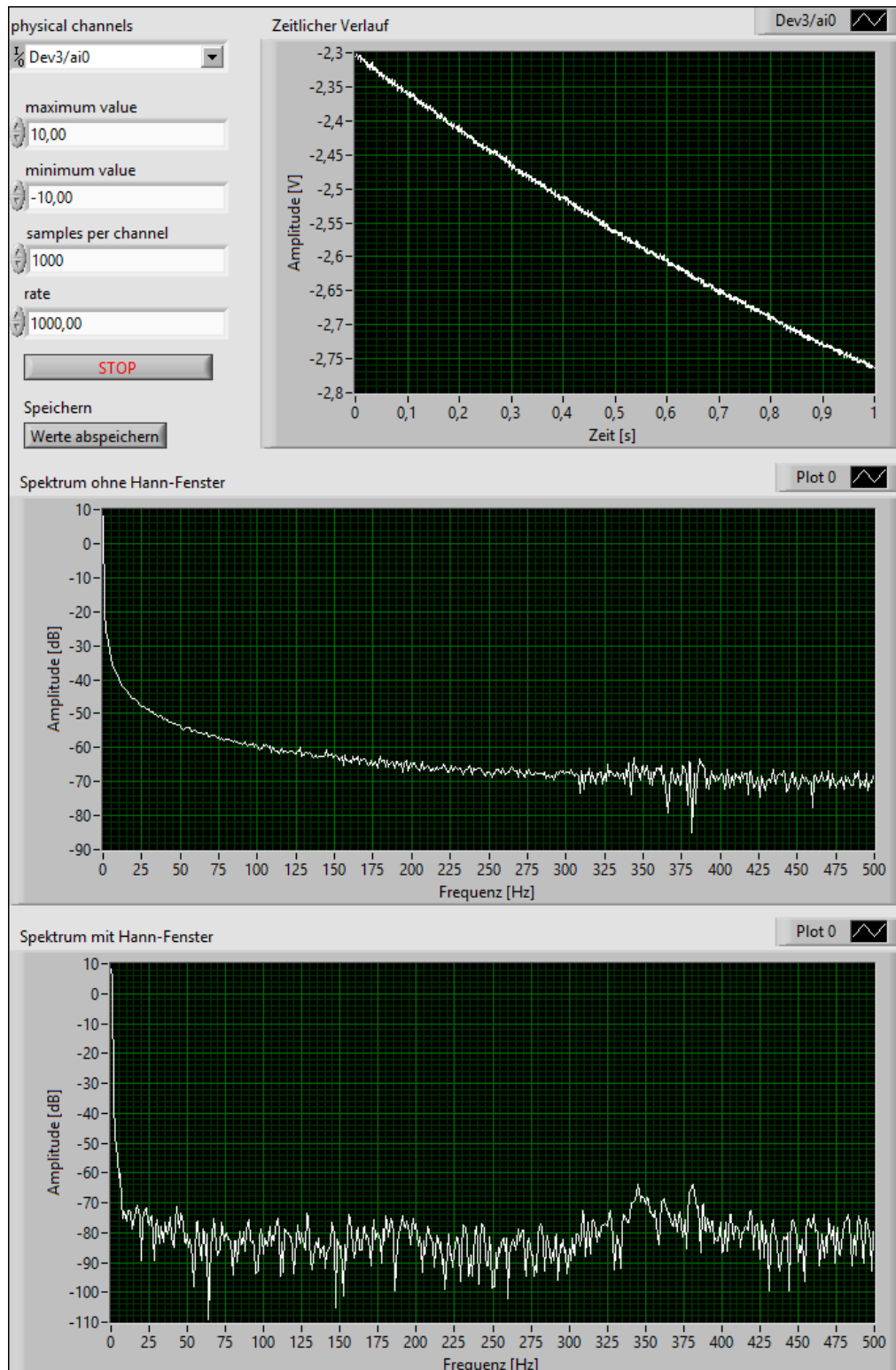


Abbildung 5: Frontpanel der Messstruktur bei eingehendem Sinussignal mit Frequenz 1 kHz und voreingestellten Parametern für Messrate etc.

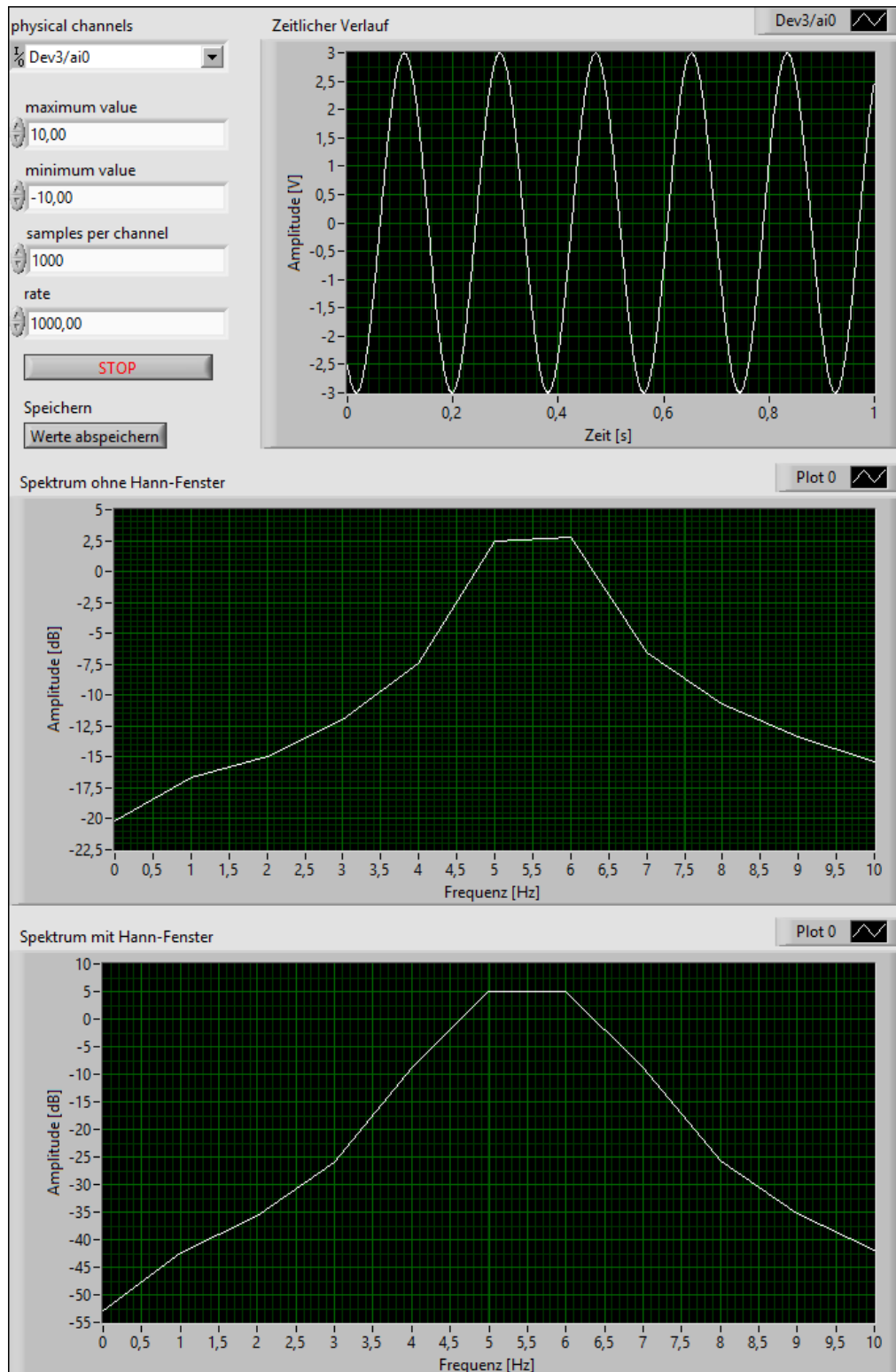


Abbildung 6: Frontpanel der Messstruktur bei eingehendem Sinussignal mit Frequenz 5 Hz und voreingestellten Parametern für Messrate etc.

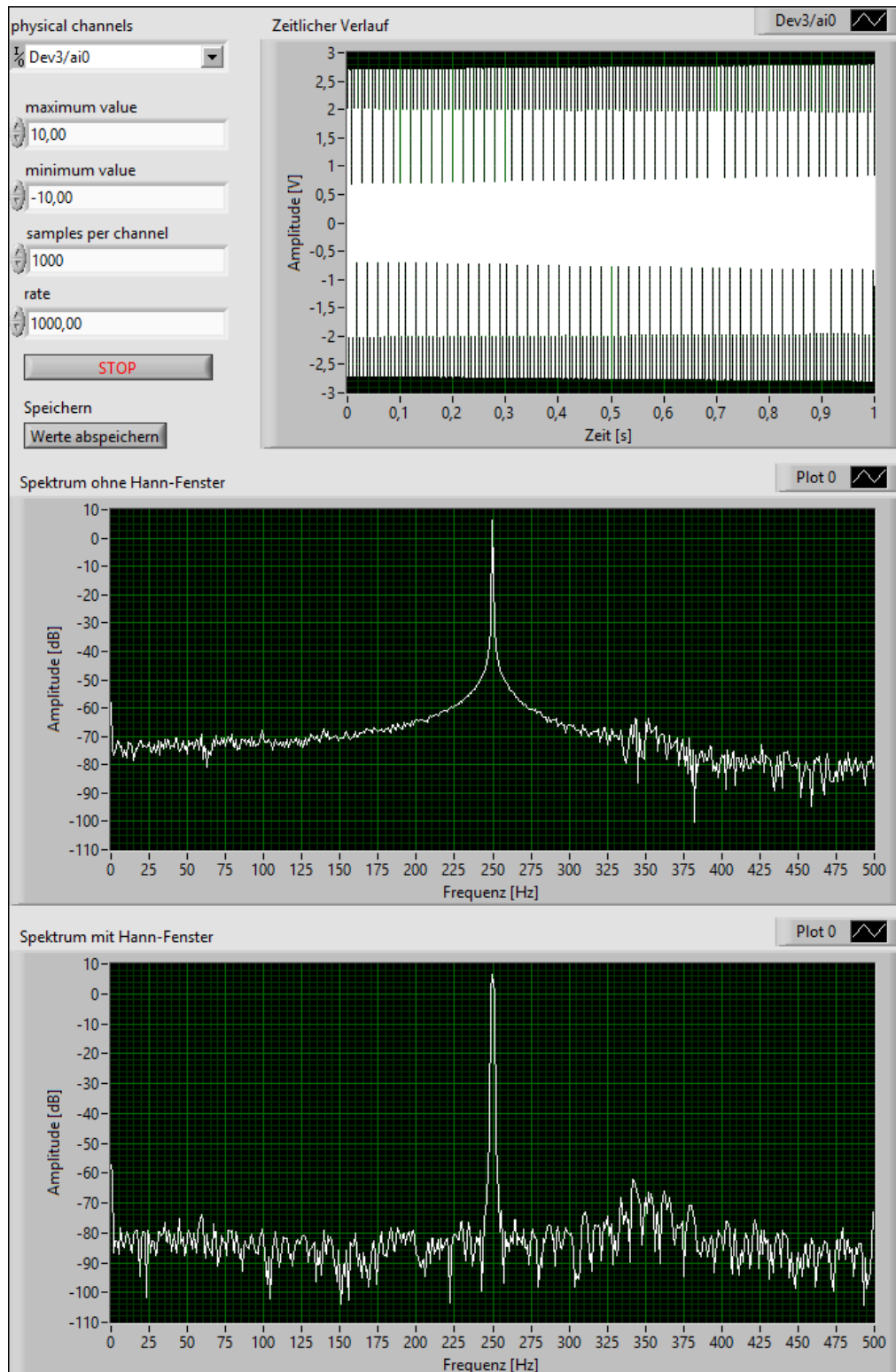


Abbildung 7: Frontpanel der Messstruktur bei eingehendem Sinussignal mit Frequenz 250 Hz und voreingestellten Parametern für Messrate etc.

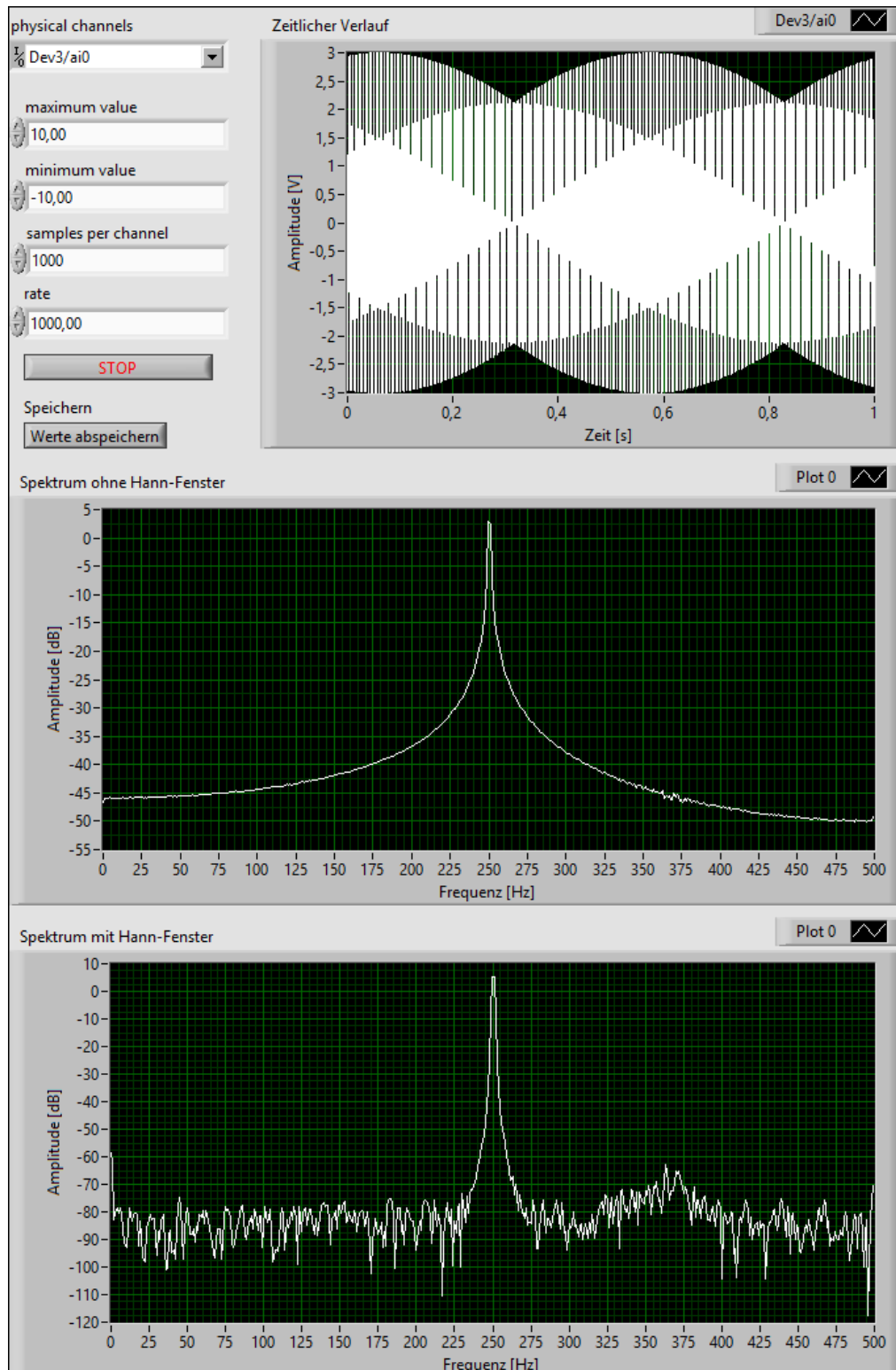


Abbildung 8: Frontpanel der Messstruktur bei eingehendem Sinussignal mit Frequenz 250,5 Hz und voreingestellten Parametern für Messrate etc.

4 Modulation

Als nächstes sollte der Funktionsgenerator durch die computereigene Soundkarte ersetzt werden. Um einen Schritt weiter zu gehen, sollte das Signal, welches der Soundkarte über ein Labview VI übergeben wird, amplitudenmoduliert sein.

4.1 Amplitudenmodulation

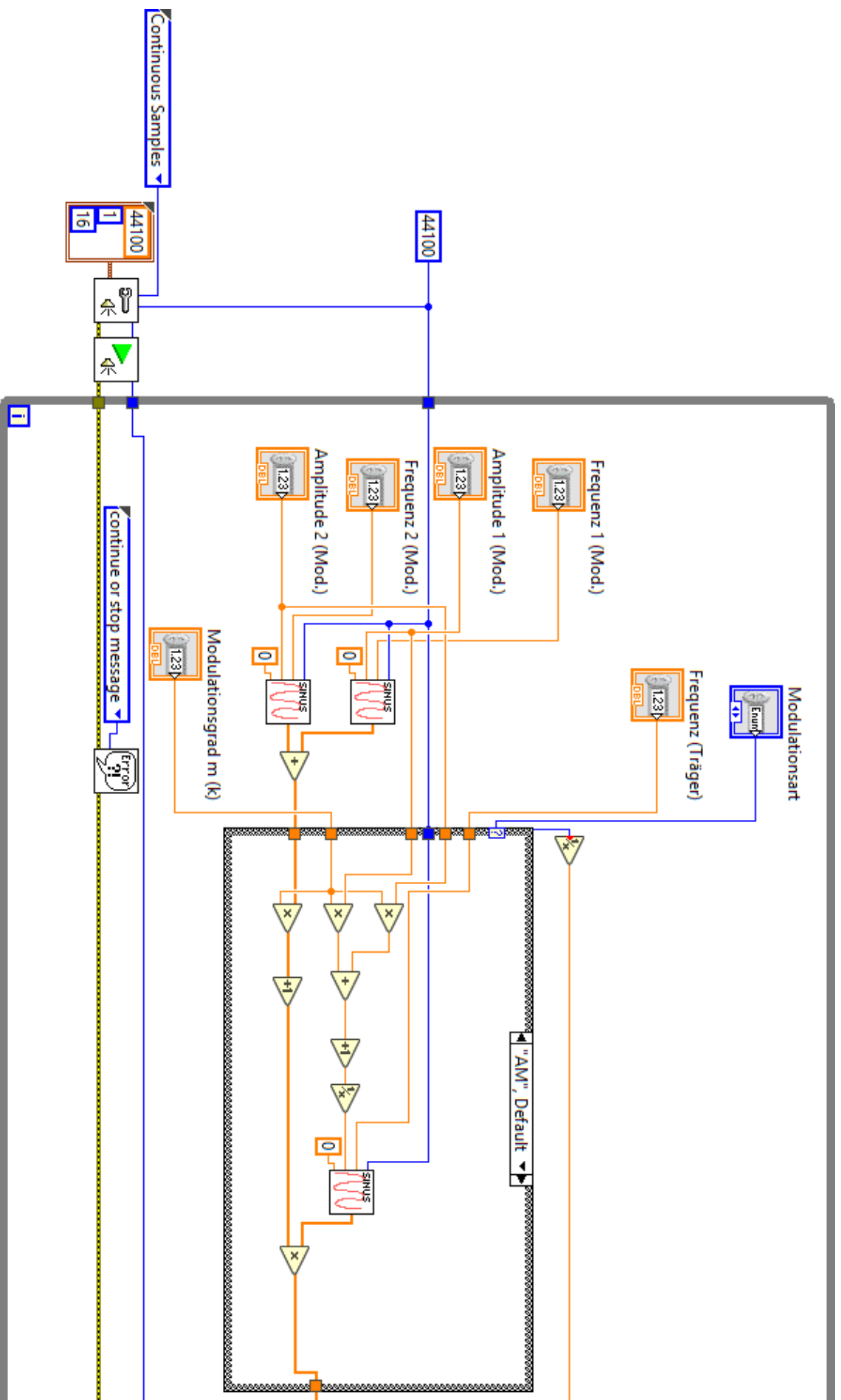
Das VI, welches ein amplitudenmoduliertes Signal erzeugt und von der Soundkarte ausgeben lässt ist in Abb. 9 dargestellt. Die VIs, welche für die Soundkartenausgabe zuständig sind, sind mit einem Lautsprechersymbol gekennzeichnet. Wie auch bei der Messtruktur verläuft von dem ersten dieser VIs eine Fehlerleitung, damit eine Fehlermeldung ausgegeben werden kann, falls in dem Programm ein Fehler auftritt. Da auch hier ein kontinuierliches Signal gesendet werden soll, wird der Modus „Continuous Samples“ für die Ausgabe festgelegt. Bei den drei Konstanten welche neben dem Modus in das Sound-VI gegeben werden handelt es sich um Frequenz, Anzahl der Kanäle und Bits. Mit den gewählten Werten von 44 100 Hz, einem Kanal und 16 Bits wird der Ton im Mono-Format und mit CD-Qualität ausgegeben, welche die eingebaute Soundkarte unterstützt. Das Sound-VI nach der Übergabe der Parameter startet die Ausgabe und von dort wird die Schleife betreten, in der das Signal entsteht. Zu erkennen sind bereits Controls für das zu modulierende Signal, wie auch für den Träger. Hierzu liegt dem Aufbau folgende Formel zu Grunde:

$$S_{AM}(t) = (m \cdot S(t) + 1) \cdot T(t) \quad (4.1)$$

$$\text{mit dem Signal } S(t) = A_1 \cdot \sin(2\pi f_1 t) + A_2 \cdot \sin(2\pi f_2 t) \quad (4.2)$$

$$\text{und dem Träger } T(t) = s_T \cdot \sin(2\pi f_T t) \quad (4.3)$$

Aufgrund der Beschränkung der Soundkarte auf Signale mit maximaler Amplitude von einem Volt, wird der Modulationsgrad m variabel gehalten, aber s_T auf $\frac{1}{mA_1 + mA_2 + 1}$ normiert. $A_{1/2}$ sind dabei die Amplituden der beiden Sinusschwingungen aus denen das zu modulierende Signal besteht und $f_{1/2}$ die jeweiligen Frequenzen, analog f_T die Frequenz für den Träger.



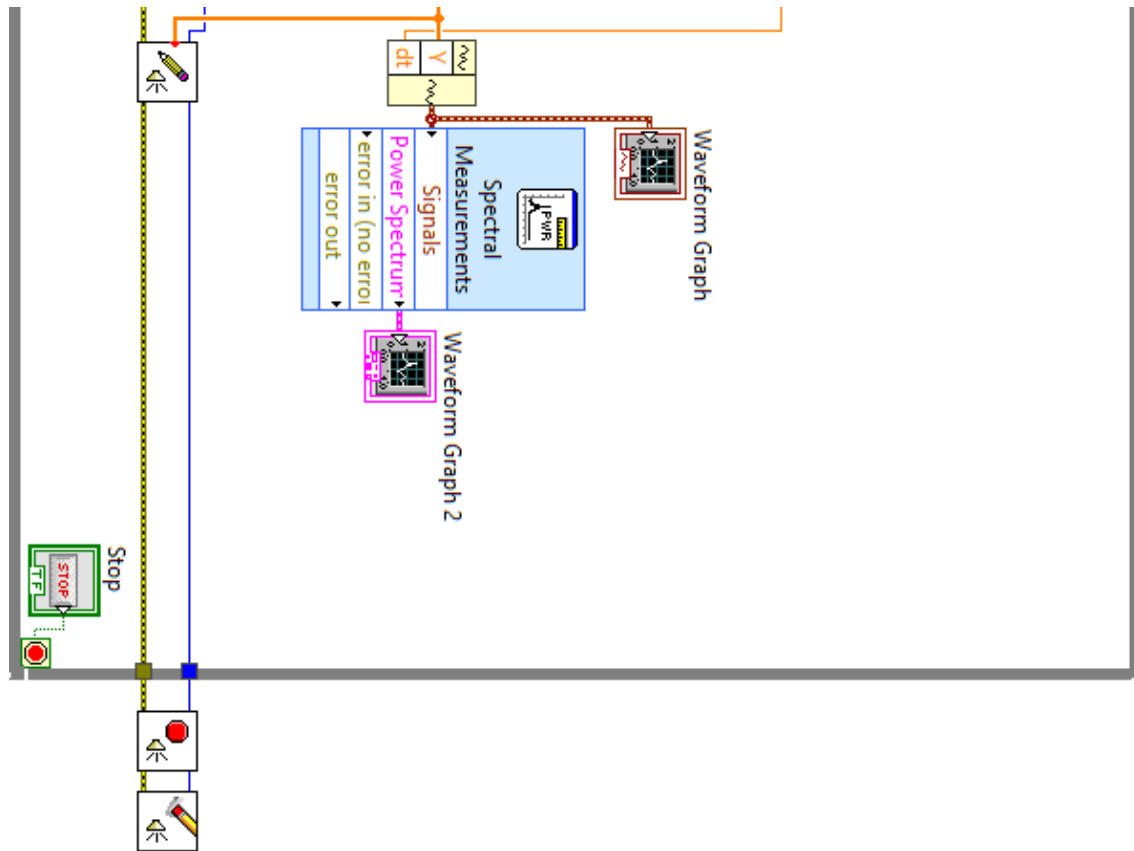


Abbildung 9: Labview VI zur Erstellung eines amplitudenmodulierten Signals, welches von der Soundkarte ausgegeben und auf dem Frontpanel einstellbar und graphisch dargestellt wird.

Das Signal $S(t)$ wird wie in Abb. 9 zu erkennen durch zwei der zuvor geschriebenen Sinus VIs verwirklicht. Zur besseren Aufnahme werden an dieser Stelle genau so viele Datenpunkte gewählt wie auch für die Sound VIs. Analog zu der Frequenz hier 44100 Datenpunkte. Wie in der obigen Formel zu erkennen, werden die Phasen der Signale vernachlässigt und einfach gleich null gesetzt. Die eigentliche Modulation, also der Übergang von $S(t)$ zu $S_{AM}(t)$ findet in der Case-Struktur statt. Bei der Abbildung handelt es sich um das Endprodukt mit zudem eingebauter Frequenz- und Phasenmodulation zwischen denen hier unterschieden werden kann. Auf die anderen Cases wird später eingegangen, da diese erst im Nachhinein in das Modulations-VI hinzugefügt wurden. Zurück zu dem AM-Case: Hier sind die Rechnungen aus den Gleichungen 4.1 und 4.3 mit normierter Amplitude nachgestellt. Das Sinus-VI in dem Case stellt hierbei das

Trägersignal dar, welches ebenfalls die Zahl der Datenpunkte von 44100 und Phase null besitzt. Zur Darstellung des modulierten Signals sind zudem ein Waveform-Builder und zwei damit verbundene Graphen hinzugefügt worden, damit man dieses im Zeit- und Frequenzbild auf dem Frontpanel betrachten kann. Natürlich verläuft das Signal nicht nur in den Waveform-Builder sondern auch in das Sound-VI, welches für die Ausgabe des eingehenden Signals über die Soundkarte führt. An dieser Stelle jedoch wurde der Ausgang zunächst falsch verkabelt, weswegen nicht das modulierte Signal ausgegeben wurde. Dieses Problem wurde erst am vierten und letzten Tag bei Erweiterung von Frequenz- und Phasenmodulation behoben.

Wie auch bei der Messstruktur wird der Vorgang solange wiederholt bis der Stopp-Knopf auf dem Frontpanel gedrückt wird. In Abbildung 10 ist dieses Frontpanel für ein Beispielsignal dargestellt.

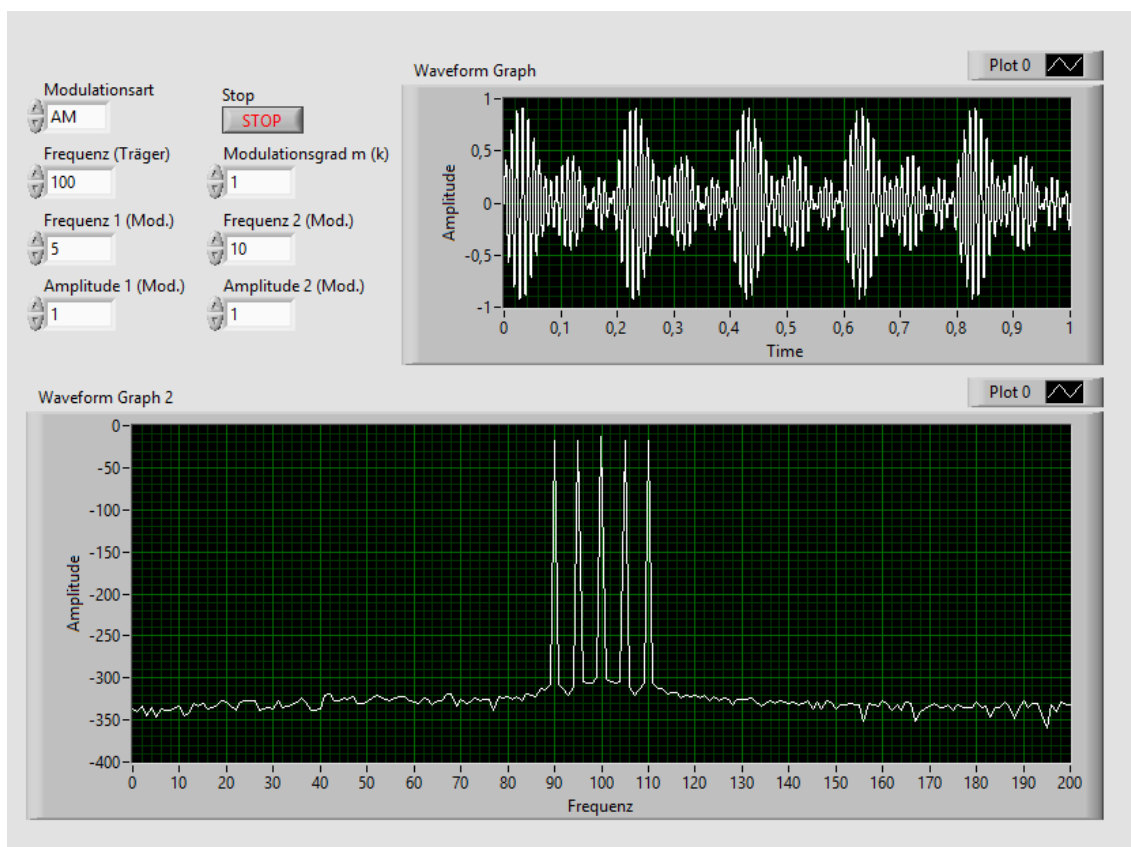


Abbildung 10: Frontpanel des Modulation-VIs bei AM mit Beispielsignal und einstellbaren Parametern.

Zu erkennen sind hier im Frequenzbild die Trägerfrequenz und die von der Amplitudenmodulation üblichen Seitenbänder links und rechts von der Trägerfrequenz pro Frequenz die im zu modulierenden Signal auftritt, sowie auch der schwebungsähnliche Verlauf im Zeitbild. Im Anhang (Abschnitt 5) sind weitere Beispiele verzeichnet.

4.2 Erweiterung der Messstruktur

Damit das nun modulierte Signal auch wieder demoduliert werden kann, musste die Messstruktur erweitert werden

5 Anhang