

addsign,allowlitunits,allowoptarg,allowzeroexp,alsoload,angformat,anglesep,astroang,closeerr,closrfr

# 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

15. September 2018

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Grundlegendes Prinzip von LabVIEW</b>	<b>2</b>
2.1	Produkt und Fakultät . . . . .	2
2.2	Programm zum Sinus . . . . .	3
2.3	Programm zu den Lissajous-Figuren . . . . .	4
<b>3</b>	<b>Messstruktur für Analog-Digital-Wandler</b>	<b>5</b>
3.1	DAQ-Assistent . . . . .	5
3.2	Programmieren einer eigenen Messstruktur . . . . .	6
3.3	Abtastung mit der Messstruktur . . . . .	11
<b>4</b>	<b>Modulation</b>	<b>17</b>
4.1	Amplitudenmodulation . . . . .	17
4.2	Erweiterung der Messstruktur . . . . .	21
<b>5</b>	<b>Anhang</b>	<b>39</b>

## 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 Grundlegendes Prinzip von LabVIEW

Das Erstellen eines Programmes in LabVIEW findet über eine grafische Oberfläche statt, welche in zwei Bereiche aufgeteilt ist. Das **Frontpanel** beinhaltet Steuerelemente zur Interaktion mit dem Nutzer des Programmes. Beispiel für solche Steuerelemente sind Schalter, Regler oder Diagramme. Das **Blockdiagramm** beinhaltet Funktionsblöcke sowie Repräsentationen der auf dem Frontpanel befindlichen Steuerelemente, die mit Verbindungslinien verbunden werden, welche den Datenfluss steuern. Die in LabVIEW erstellten Programme liegen in Form sogenannter VIs (virtual instruments) vor. Jedes Programm ist eigenständig lauffähig, kann aber auch als Unterprogramm in anderen VIs verwendet werden.

### 2.1 Produkt und Fakultät

Zur Verinnerlichung dieses Prinzips wurde ein Programm mit vergleichsweise einfachem Aufbau konstruiert, mit welchem das Produkt zweier Zahlen, und die Fakultät einer Zahl berechnet werden konnte. Das Blockdiagramm und das Frontpanel sind in der folgenden Abbildung 1 dargestellt.

**Hier Bilder zu Produkt und Fakultät einfügen.** Bildbeschreibung: Diese Grafik zeigt das Blockdiagramm (links) und das Frontpanel (rechts) des besagten Programmes zur Berechnung von Produkten und Fakultäten. Es ist der Output des Programmes für drei beispielhafte Zahlen zu sehen.

Im unteren Teil des Blockdiagrammes (Abb. 1 u.r.) ist der (grafische) Code zur Berechnung des Produktes zweier Zahlen enthalten. Die beiden Datenquellen (Variablen des Typs Double mit den Bezeichnungen Faktor 1 und Faktor 2) sind mit einem Funktionsblock verbunden, welcher die mathematische Operation der Multiplikation durchführt. Der Output des Funktionsblockes ist mit einem Indikator (Datensenke) verbunden. Die Datenquellen und -senken sind Bedien- und Anzeigeelementen auf dem Frontpanel verknüpft (Abb. 1 u.l.).

Im oberen Teil des Blockdiagrammes (Abb. 1 o.l.) ist der Code zur Berechnung der Fakultät der Variable n (Typ: Double) enthalten. Die Datenquelle ist mit einer for-Schleife

(umrandetes Rechteck) verbunden und gibt die Zahl der Wiederholungen an. Die for-Schleife beinhaltet ein Schieberegister (Startwert: 1, bestimmt durch die Konstante links außerhalb der Schleife), das Inkrement (Startwert: 0), und einen Multiplikationsoperator.

In jedem Schleifendurchlauf wird das Inkrement mit dem aktuellen Inhalt des Schieberegisters multipliziert, und das Ergebnis der Multiplikation als neuer Wert des Registers übernommen. Um eine Multiplikation mit Null zu vermeiden wird das Inkrement durch den „+1“-Block zusätzlich um 1 erhöht. Der Inhalt des Schieberegisters wird nach Durchlauf der Schleife an eine Datensenke weitergereicht. Wie auch beim Programmteil zur Berechnung des Produktes sind Datenquelle und -senke mit entsprechenden Elementen auf dem Frontpanel verbunden.

## 2.2 Programm zum Sinus

Nach Fertigstellung des VIs zur Berechnung von Produkt und Fakultät wurde ein Programm geschrieben, welches ein Array mit Funktionswerten einer Sinusfunktion füllte und grafisch ausgab. Genauer wurde eine Sinusfunktion des Typs  $A \cdot \sin(\omega t + \Phi)$  realisiert, mit variabler Amplitude  $A$ , Winkelgeschwindigkeit  $\omega = 2\pi f$  und Phase  $\Phi$ . Blockdiagramm und Frontpanel des Programms sind in der folgenden Abbildung 2 dargestellt.

### Blockdiagramm und Frontpanel zum Sinus einfügen!

Zur Berechnung der Zeitlichen Komponente  $t$  der Funktion wurde eine for-Schleife verwendet (vgl. Abb. 2 o.l.). Diese erhält als Input einen Wert  $N$ , und gibt in jeden Durchlauf den Wert  $i/N$  aus, so dass die Zeitreihe  $0/N, 1/N, \dots, (N-1)/N$  entsteht. Diese wird als Definitionsmenge („x-Wert“) an das Array weitergereicht. Zur Berechnung des Wertebereichs wird die mit der for-Schleife berechnete Zeitkomponente mit der Konstante  $2\pi$  und der variablen Frequenz  $f$  multipliziert. Zu dem Produkt wird anschließend die variable Phase  $\phi$  addiert. Mittels einem von LabVIEW zur Verfügung gestellten Funktionsblock wird der Sinus der Summe berechnet, welcher anschließend mit der Amplitude  $A$  multipliziert wird. Das gefüllte Array wird an den Funktionsblock „XY Graph 2“ weitergereicht, mit dessen Hilfe die Daten auf dem Frontpanel grafisch dargestellt werden. Um einzelne Werte des Arrays überprüfen zu können, wird der Wertebereich zusätzlich mit einem Indicator verbunden („sin(t<sub>i</sub>)“, vgl. Abb. 2 unten). Auf dem

Frontpanel (vgl. Abb. 3) sind die mit den Variablen  $N$ ,  $f$ ,  $\phi$  und  $A$  verknüpften Steuerelemente, der Graph und der Indicator zu sehen.

## 2.3 Programm zu den Lissajous-Figuren

Um die Funktion von VIs als Unterprogramme zu zeigen, wurde ein Programm erstellt, welches mit Hilfe des vorher erstellten Sinus.vi Lissajous-Figuren zeichnen kann. Als Lissajous-Figuren werden Überlagerungskurven zweier orthogonal aufeinander stehender harmonischer Schwingungen bezeichnet. Diese lassen sich mathematisch als Abbildungen folgender Form beschreiben:

(Hier dieses Vektording einfügen)

Zum Zeichnen der Figuren wurde ein s.g. Express.vi verwendet, welches aus einem „X Input“ und einem „Y Input“ in dynamischem Format ohne weitere Einstellungen einen „XY Graph“ erstellt. Für die beiden Inputs wurde jeweils das vorher erstellte Sinus-Programm verwendet. Letzteres wurde so eingestellt, dass es Amplitude, Frequenz und Phase als Input entgegennimmt, und das mit den Funktionswerten gefüllte Array ausgibt.

Das Express.vi samt In- und Output wurde in einer While-Schleife platziert, damit die Parameter während der Ausführung des Programmes geändert werden können, und die Änderungen direkt sichtbar werden. Man beachte den „stop“-Button (vgl. Abb. u.r.), mit dem die Schleife unterbrochen wird, und die Clock (vgl. Abb. o.r.), welche die Wartezeit zwischen zwei Schleifendurchläufen bestimmt. Letztere wurde auf einen Wert von 100 ms eingestellt, welcher gering genug ist um das Programm subjektiv flüssig darzustellen, aber hoch genug um die CPU-Auslastung merklich zu reduzieren.

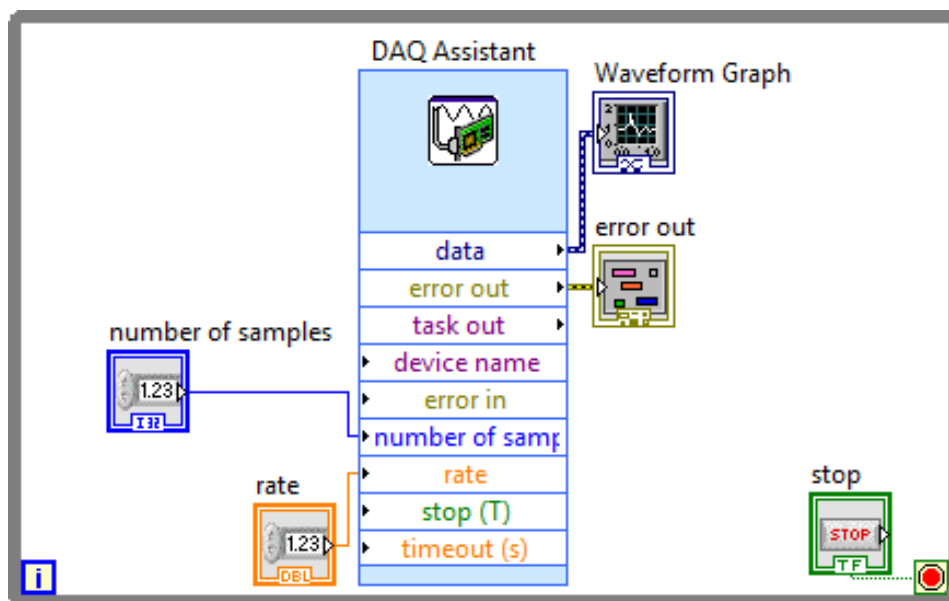
**Hier die Bilder zum Lissajous-Programm einfügen.**

### 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 National Instruments handelte, erkannte LabVIEW diesen sofort.

#### 3.1 DAQ-Assistant

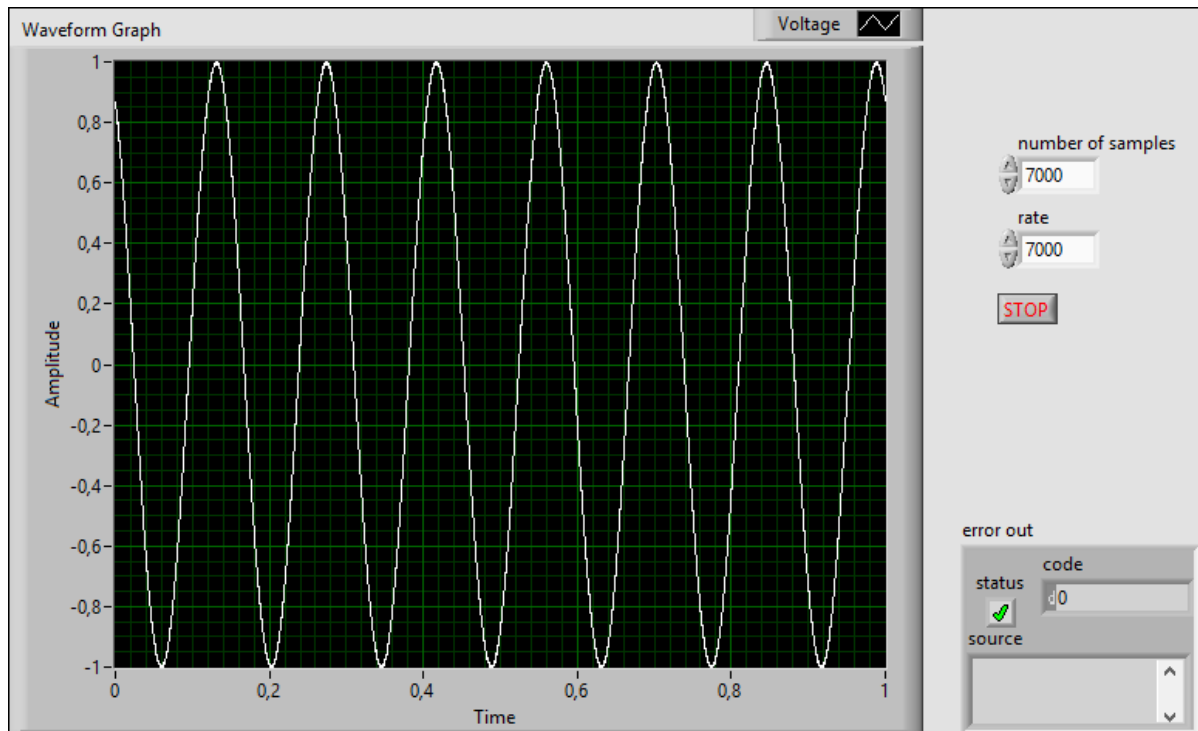
Mit dem vorprogrammierten Express-VI, dem DAQ-Assistant (siehe Abb. 1), konnte gemessen werden. Lediglich die Anzahl der Datenpunkte und die Messrate mussten dem



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

DAQ-Assistant 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.





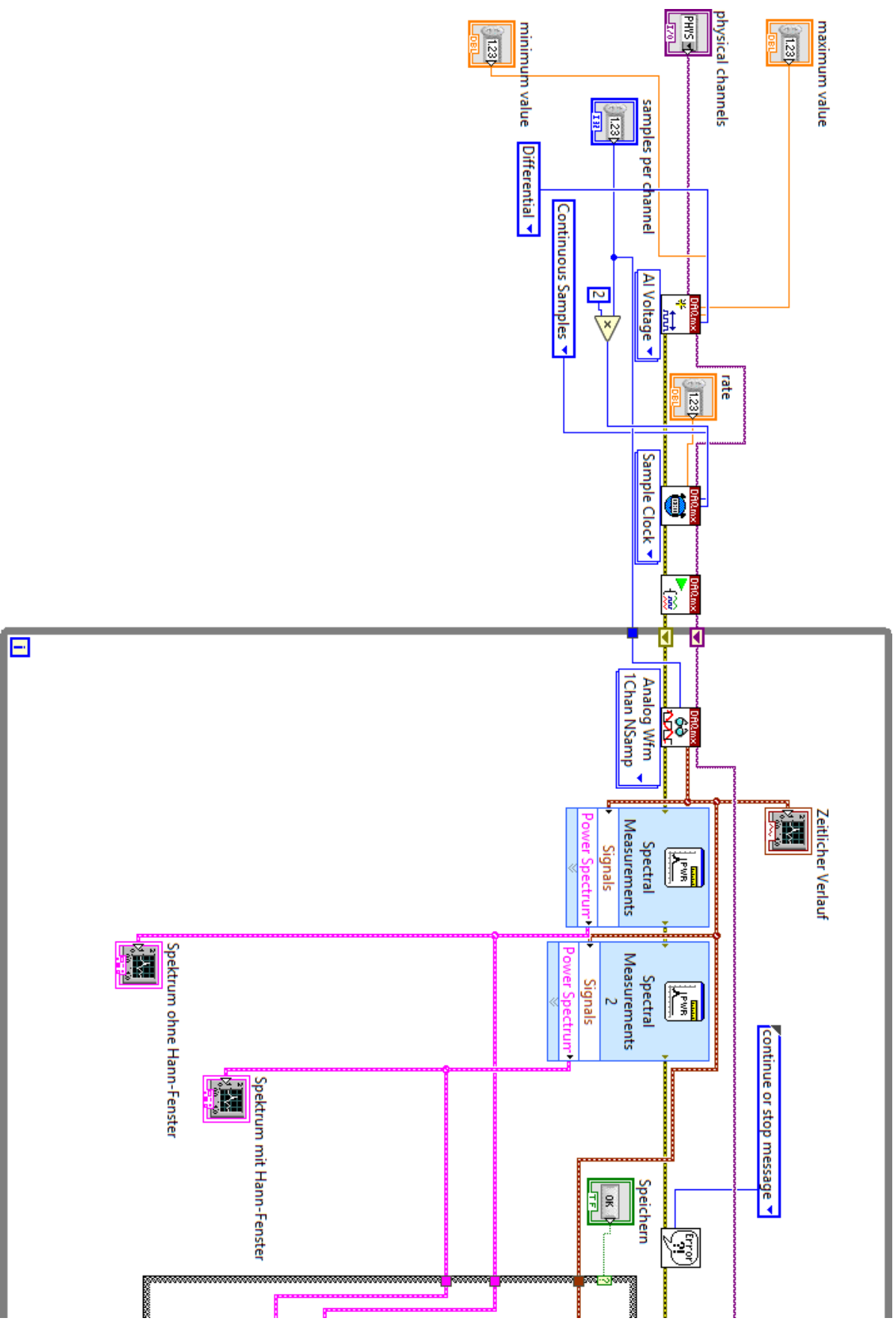
**Abbildung 2:** Diese Abbildung stellt das Frontpanel eines VIs mit dem DAQ-Assistent und Controls für Messrate und Anzahl der Datenpunkte dar.

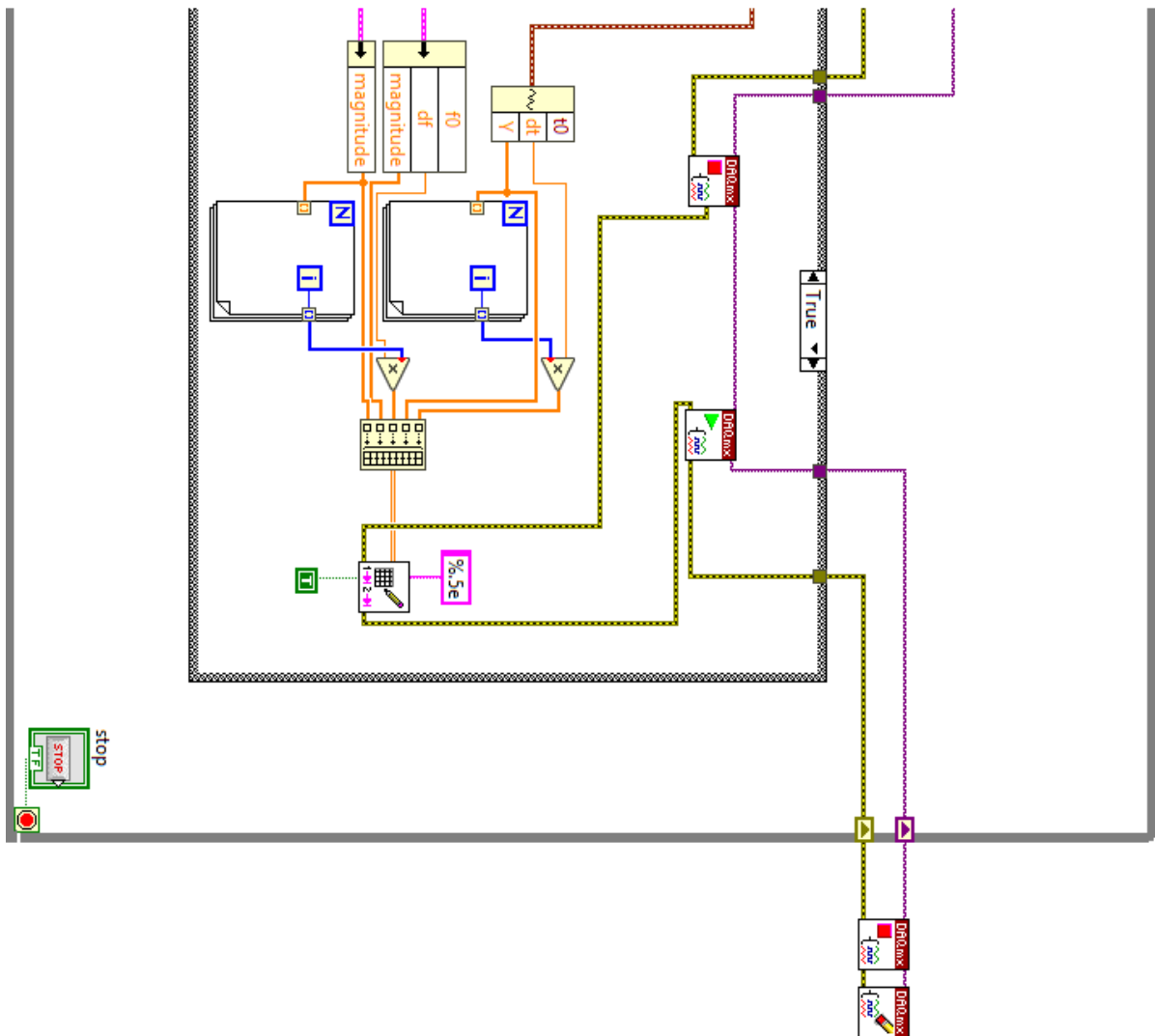
### 3.2 Programmieren einer eigenen Messstruktur

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.





**Abbildung 3:** LabVIEW VI zur Messung, Spektralanalyse und Speicherung eines eingehenden Signals.

Ähnlich zu dem DAQ-Assistent sind hier die Eingänge für Anzahl der Datenpunkte und Messrate vertreten. Die VIs, welche die Funktionen des DAQ-Assistants ersetzen sollen sind die DAQmx-VIs, welche in Reihe geschaltet werden. Als Eingang in das erste DAQmx-VI wird der Kanal übergeben von dem das Signal empfangen werden soll, welcher hier „physical channel“ heißt. Bei dem verwendeten Computer handelte es sich bei diesem Kanal um „Dev3/ai0“. Da Spannungen im differentiellen Modus gemessen

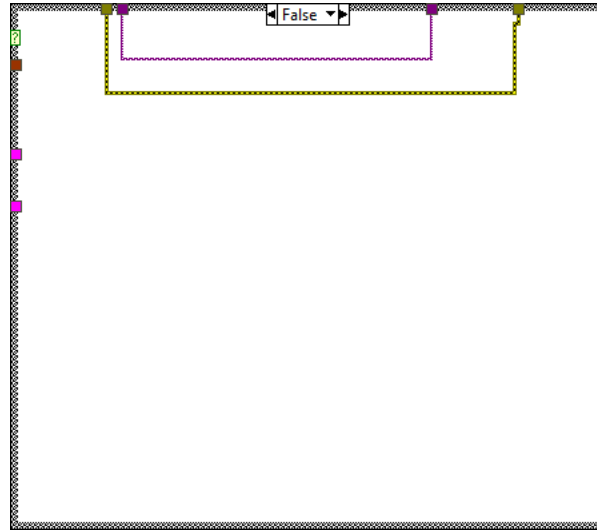
werden sollten wurden „AI Voltage“ und „Differential“ als Modus bzw. konstanter Eingang an das DAQmx-VI gelegt. Um die gemessene Spannungen variabel zu begrenzen wurden zusätzlich Controls angebracht. Die Benennung dieser mit „minimum value“ und „maximum value“ entstammt der voreingestellten Namen der 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.

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.

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

### 3.3 Abtastung mit der Messstruktur

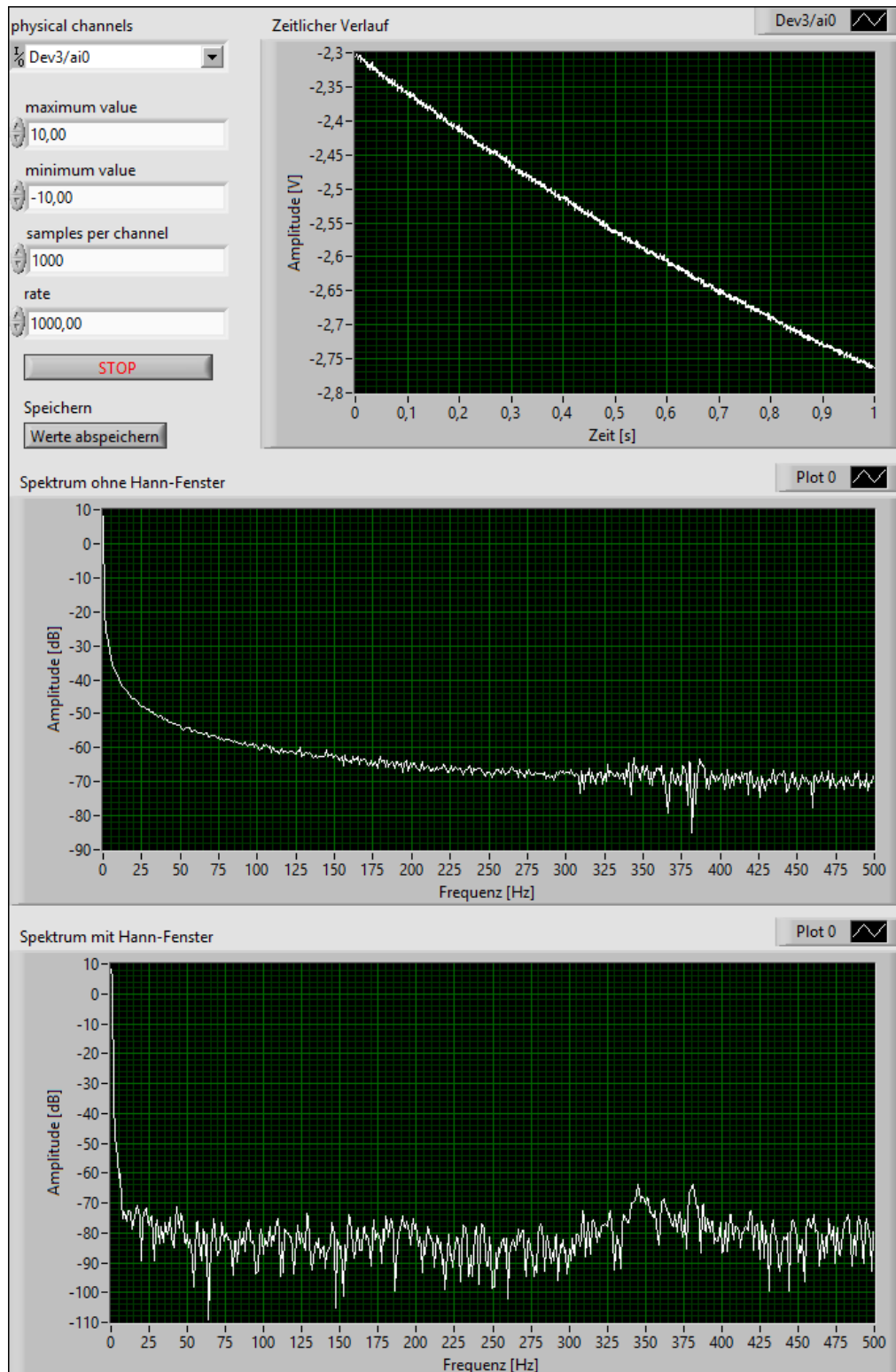
Mit der vorerst fertigen Messstruktur sollte nun abgetastet werden. Die Abbildungen 5 bis 8 stellen verschiedene Fälle dar. Der Verlauf in Abb. 5 stellt den Fall der Unterabtastung dar. Hierbei entspricht die Frequenz der Messung mit der Signalfrequenz des Funktionsgenerator überein. Um das Abtasttheorem 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

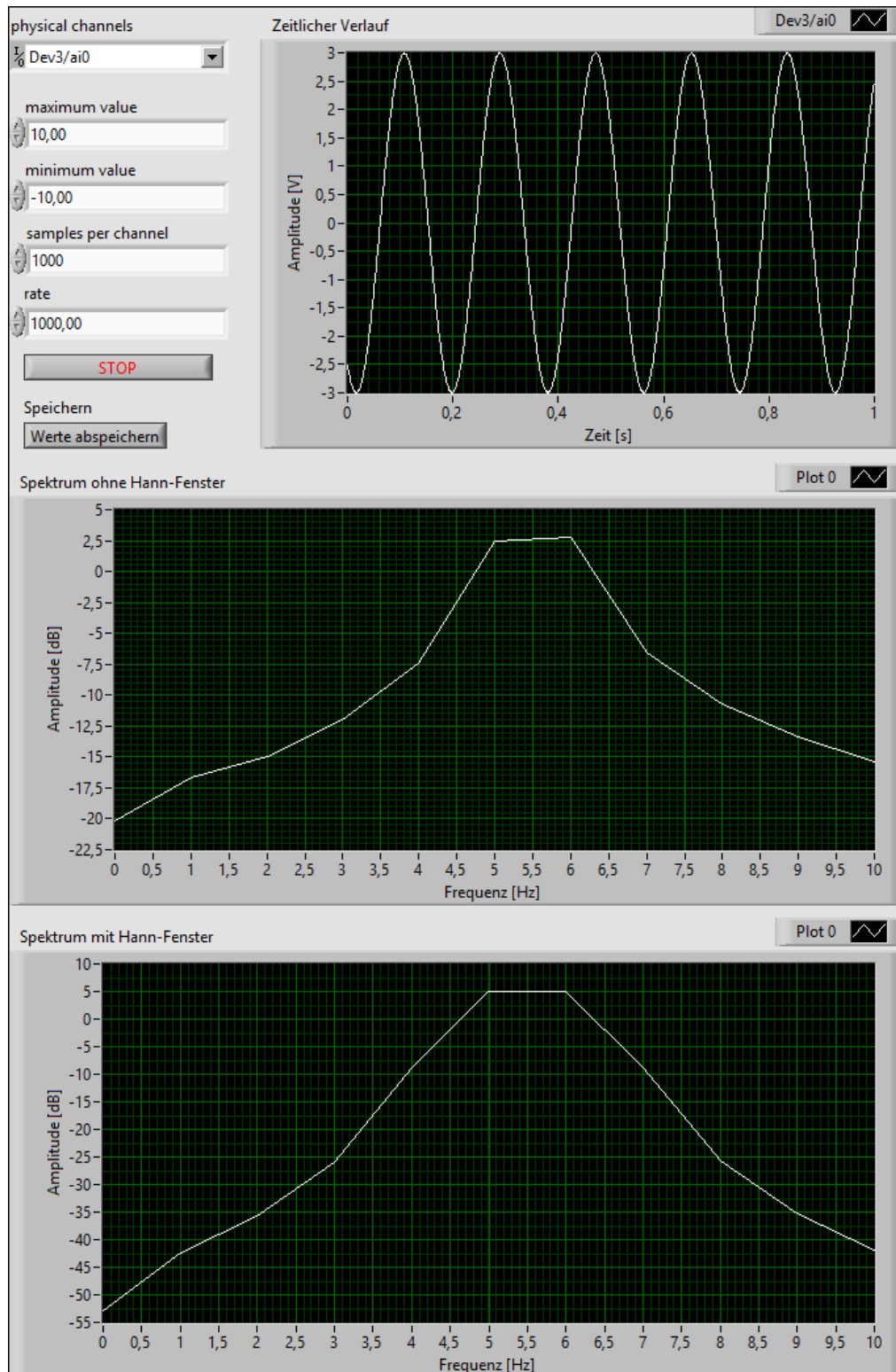
in Abb. 7 kaum zu erkennen, jedoch lässt sich anhand der Amplituden ein Muster erkennen, welches einem einfachen Sinus deutlich mehr ähnelt als das in Abb. 8.

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.

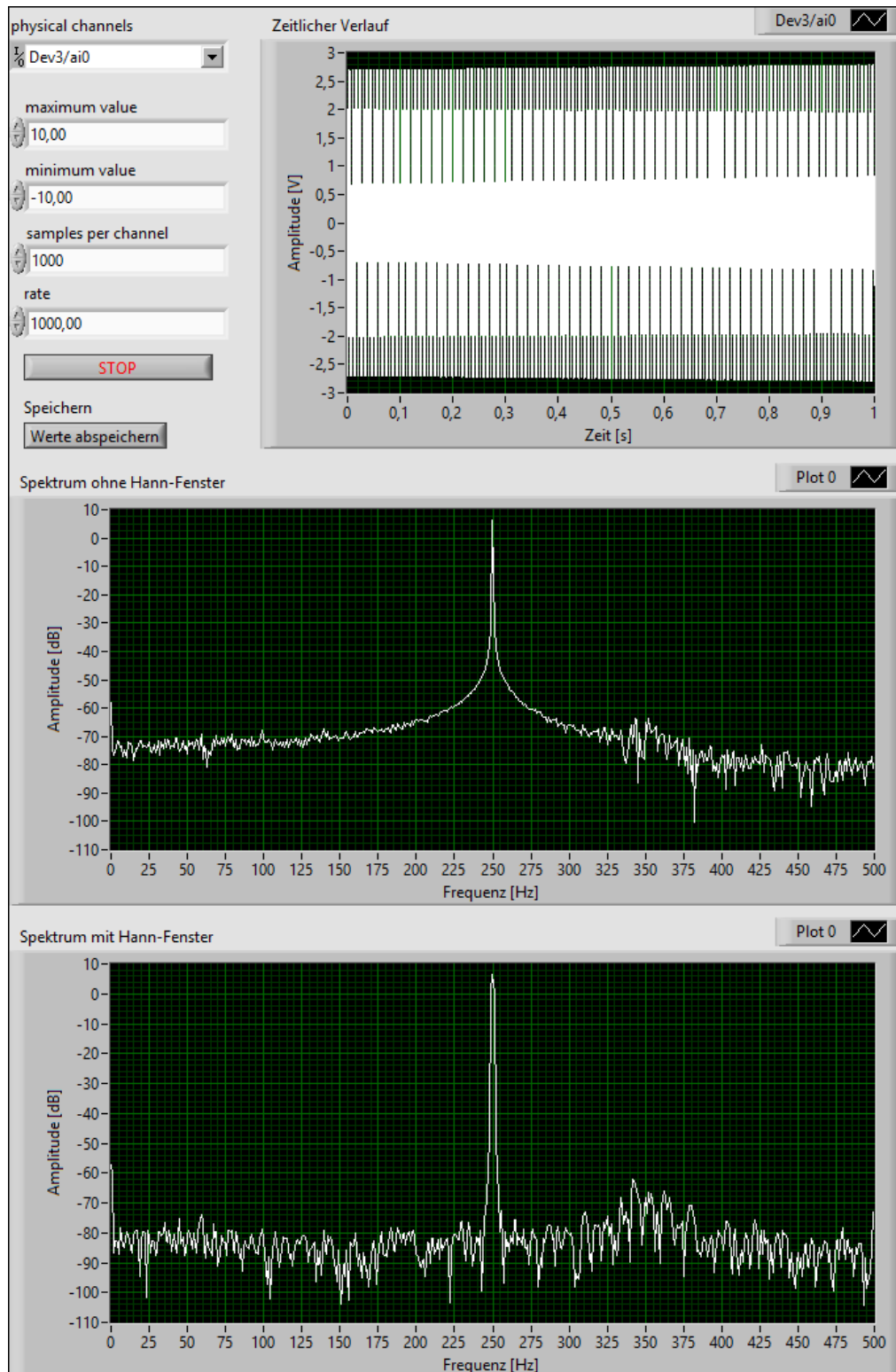


**Abbildung 5:** Frontpanel der Messstruktur bei eingehendem Sinussignal mit Frequenz 1 kHz und voreingestellten Parametern für Messrate etc.

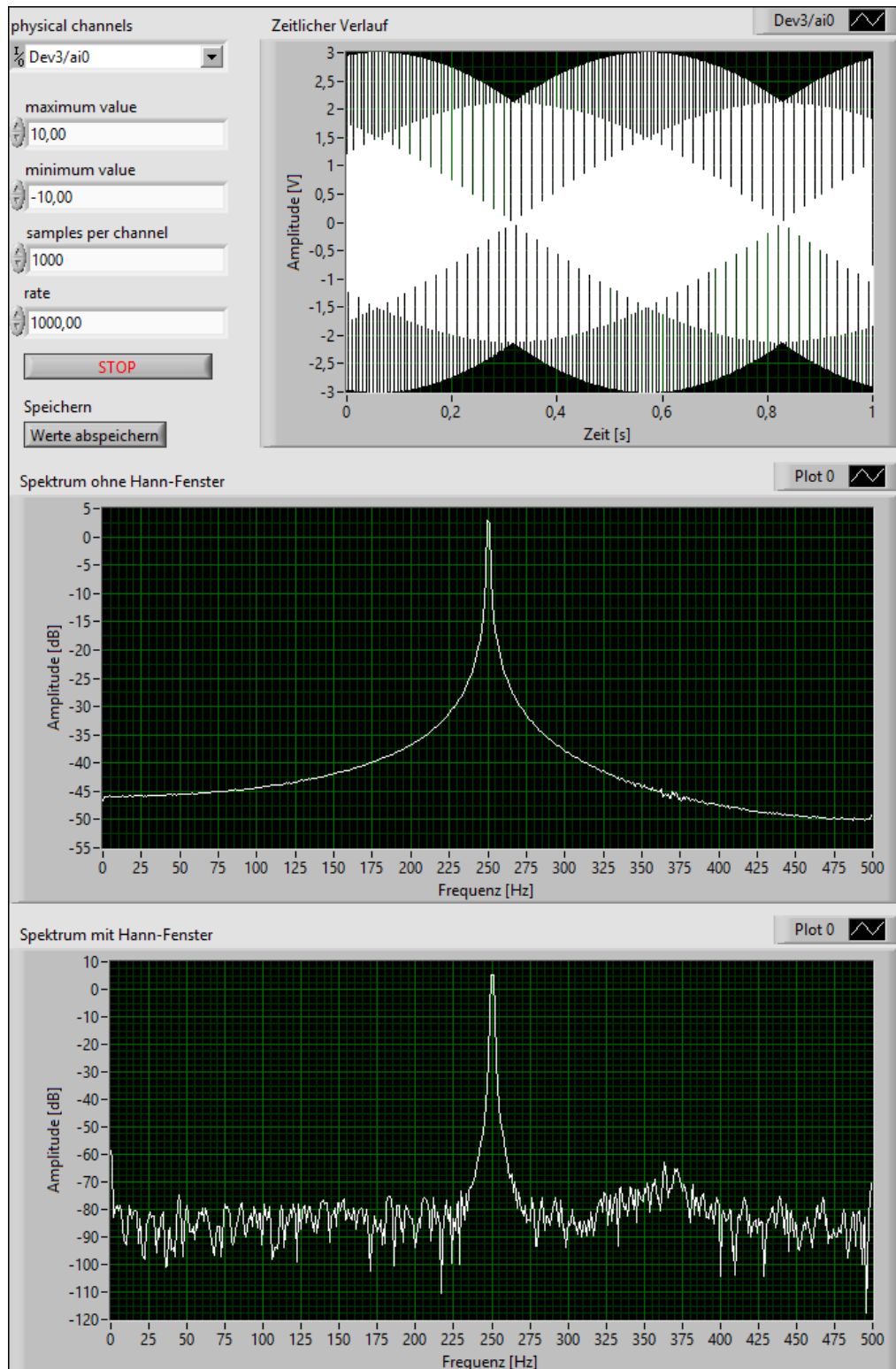




**Abbildung 6:** Frontpanel der Messstruktur bei eingehendem Sinussignal mit Frequenz 5 Hz und voreingestellten Parametern für Messrate etc.



**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

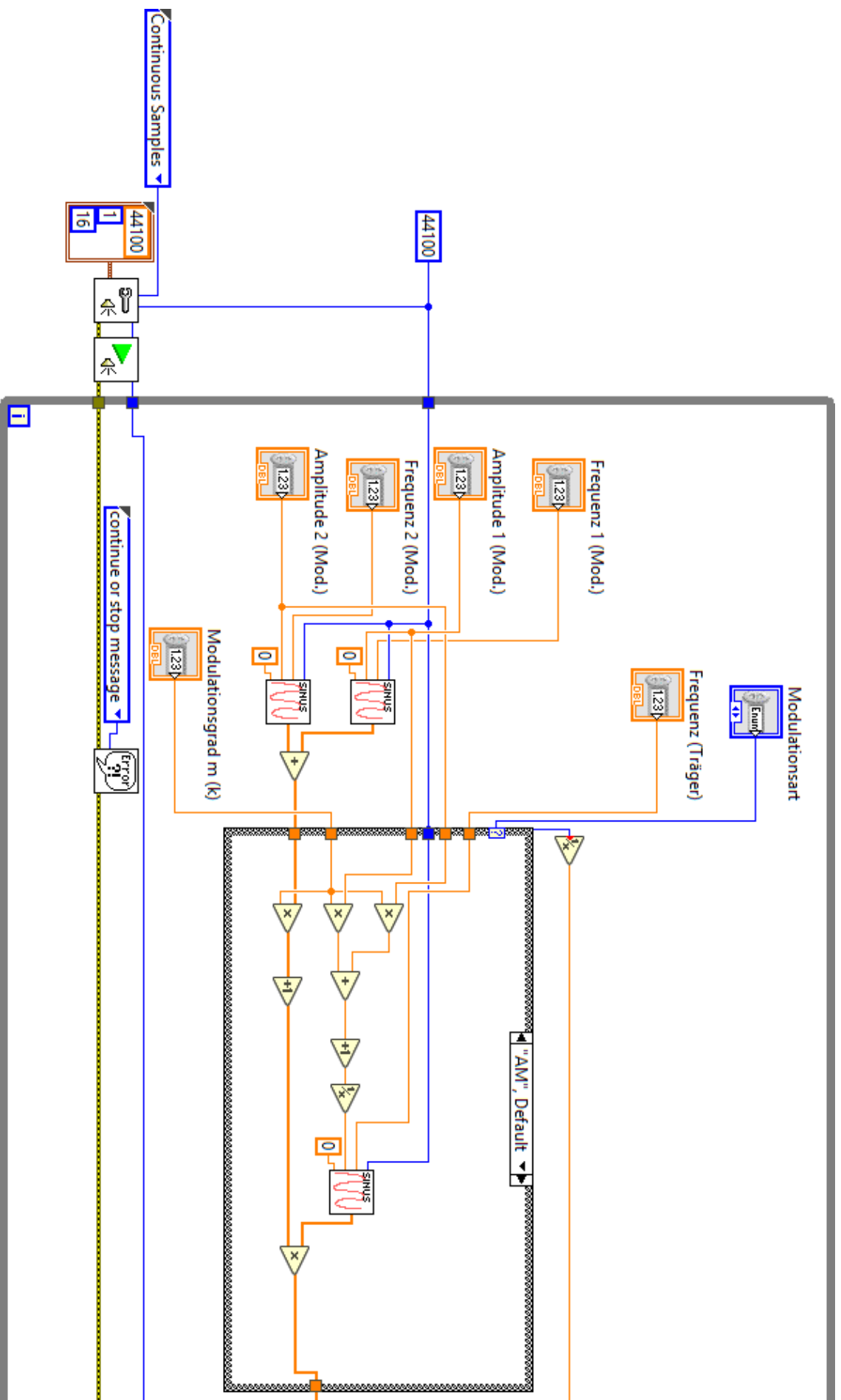
Das VI, welches ein amplitudenmoduliertes Signal erzeugt und von der Soundkarte ausgeben lässt ist in Abb. 9 dargestellt. Die VIs, welche für die Soundkartenausgabe zuständig sind, sind mit einem Lautsprechersymbol gekennzeichnet. Wie auch bei der Messtruktur verläuft von dem ersten dieser VIs eine Fehlerleitung, 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:

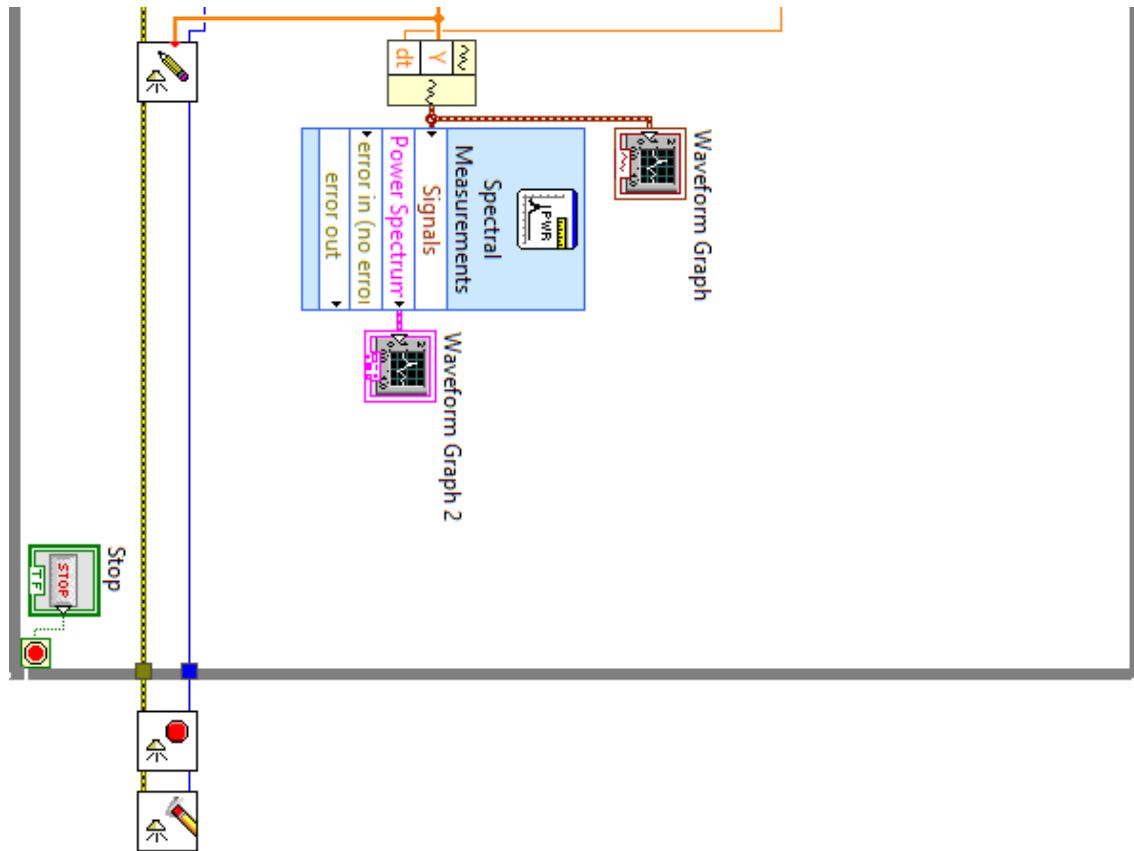
$$S_{AM}(t) = (m \cdot S(t) + 1) \cdot T(t) \quad (4.1)$$

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

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

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.





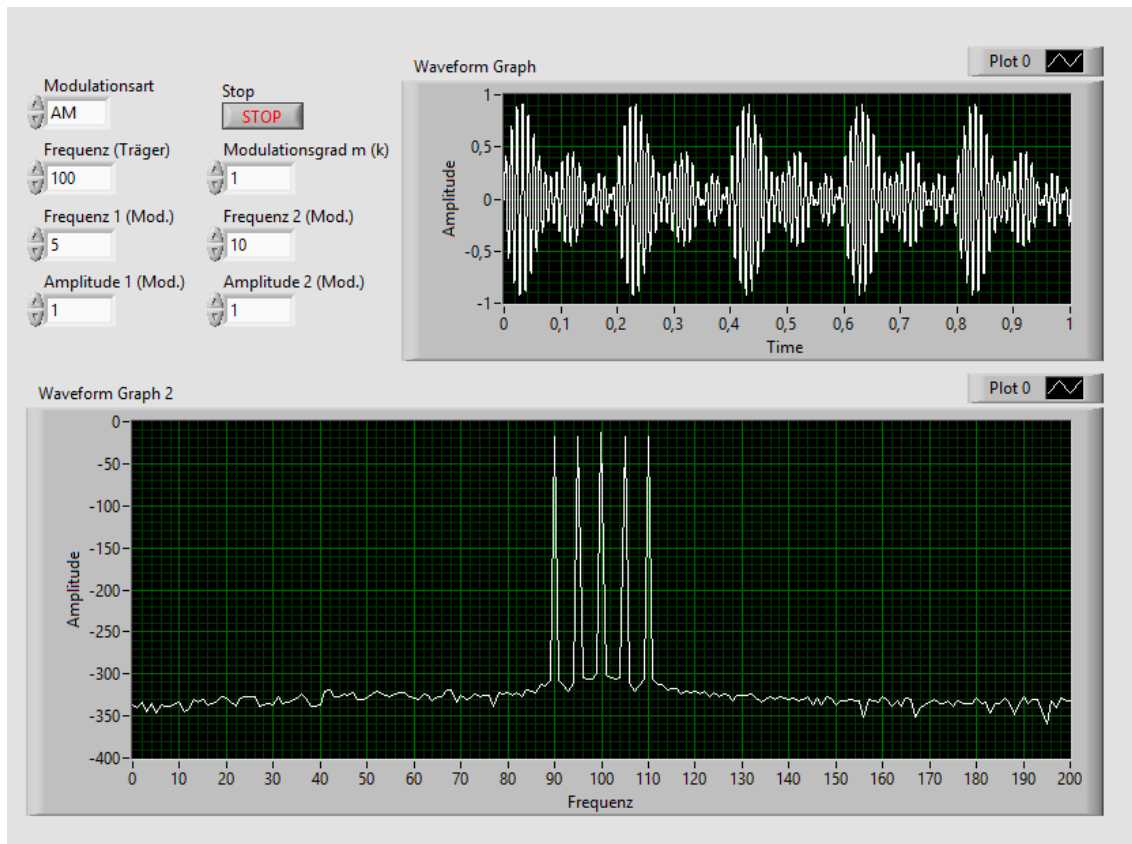
**Abbildung 9:** LabVIEW VI zur Erstellung eines amplitudenmodulierten Signals, welches von der Soundkarte ausgegeben und auf dem Frontpanel einstellbar und graphisch dargestellt wird.

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.

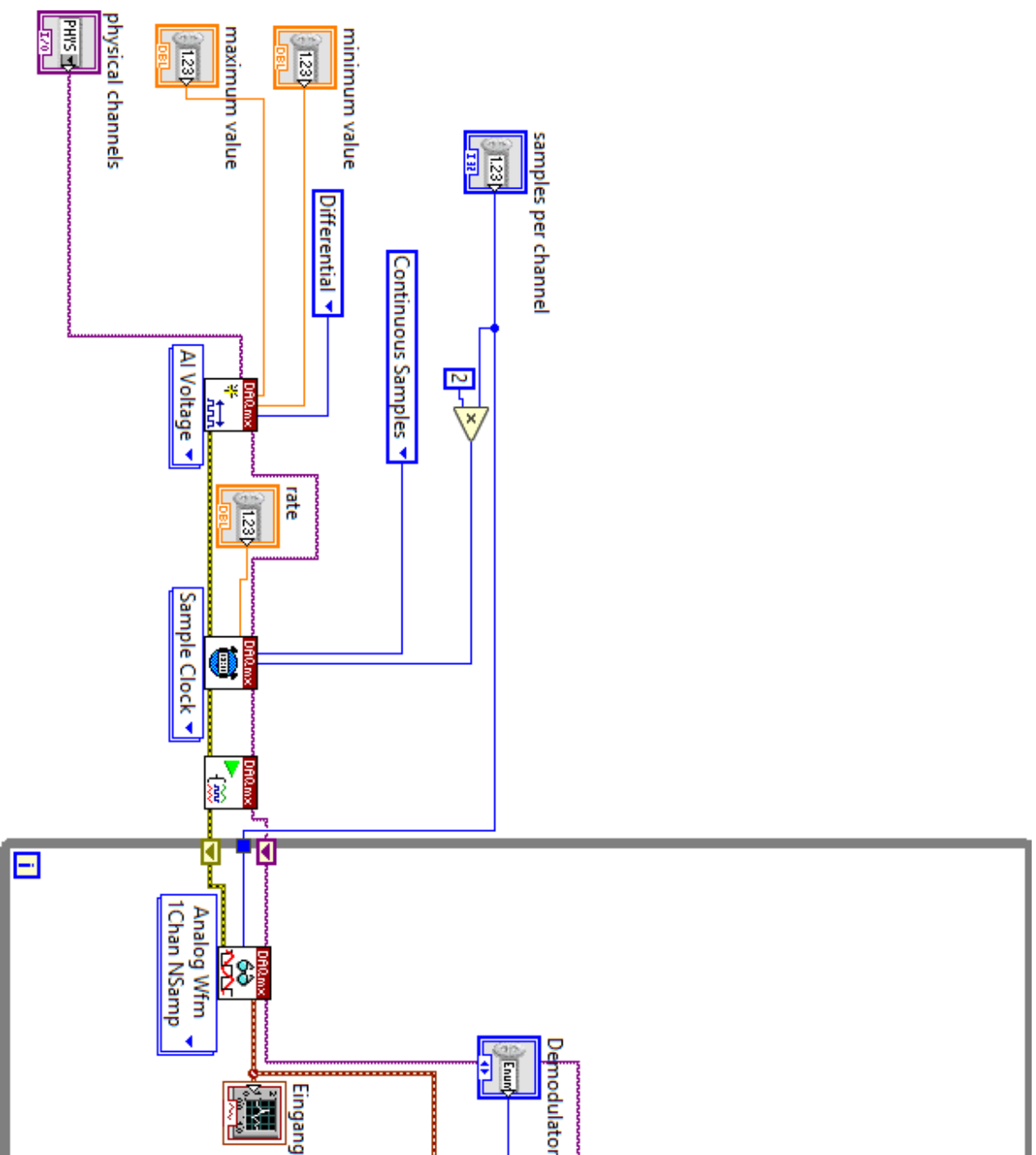
## 4.2 Erweiterung der Messstruktur

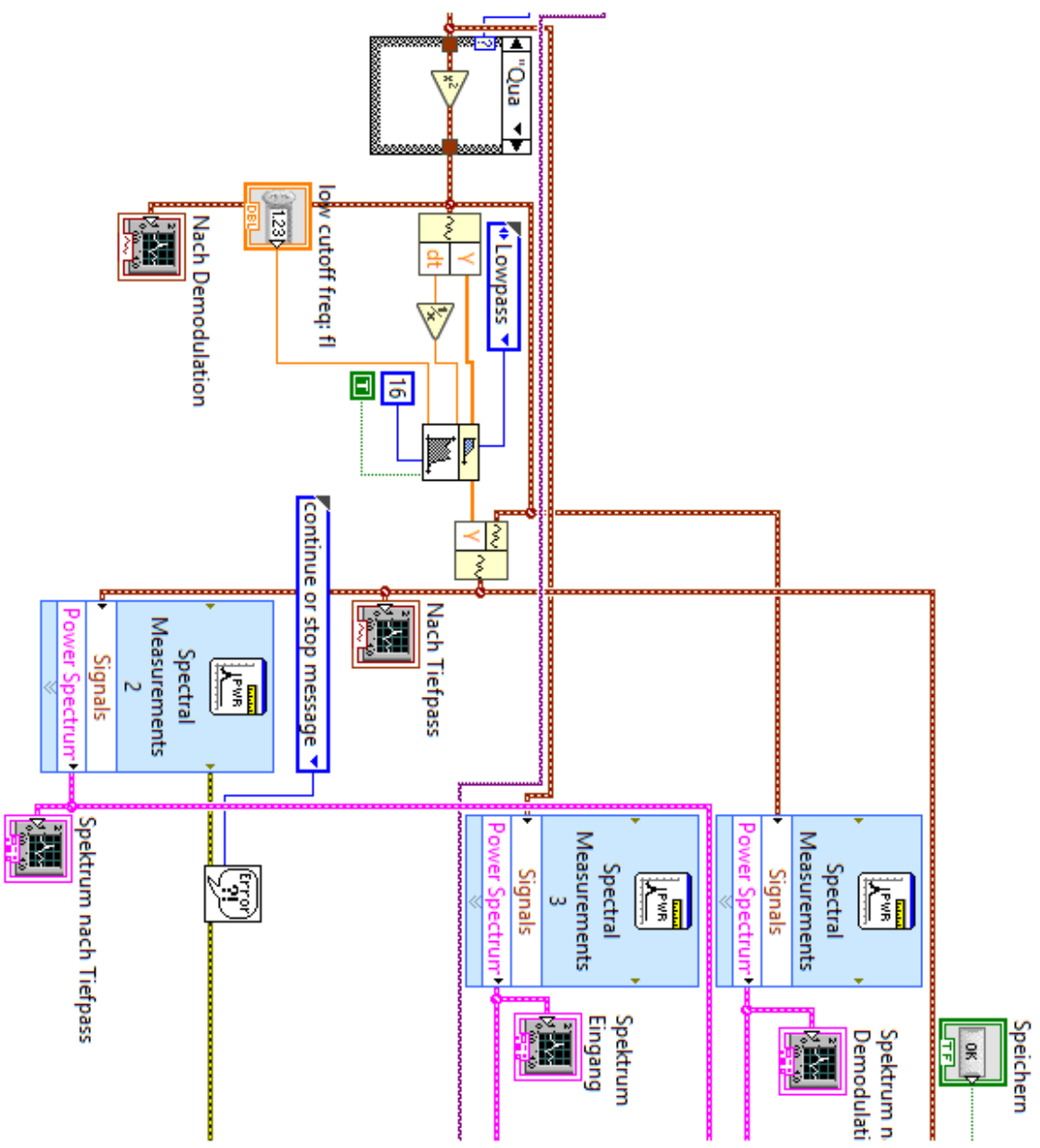
Damit das nun modulierte Signal auch wieder demoduliert werden kann, musste die Messstruktur erweitert werden. Für die Amplitudenmodulation wurden hier drei verschiedene Arten der Demodulation integriert, von denen eine ausgewählt werden kann. Die Erweiterung von Abb. 3 ist in Abb. 11 dargestellt.



Bei den verschiedenen Demodulationsverfahren handelt es sich um die Quadratur, die Betragsbildung und die Multiplikation mit dem Träger, welche jedoch erst im Nachhinein am Morgen des vierten Tages hinzugefügt wurde. Um auch hier zu unterscheiden wurde eine Case-Struktur angelegt, bei der auch „keine Demodulation“ eine Option ist. Der Fall der Quadratur ist bereits in Abb. 11 dargestellt. Das Signal wird dabei einfach quadriert. Analog dazu wird bei der Betragsbildung einfach ein VI für den Betrag eingeschoben und bei keiner Demodulation die Kabel ohne weiteres durch den Case gezogen (vgl. Abb. 12). Da nach den Rechenoperationen auch noch Frequenzen im Bereich doppelter Trägerfrequenz auftauchen, müssen diese gefiltert werden. Dies lässt sich leicht mit einem Tiefpass verwirklichen, solange das Verhältnis zwischen Trägerfrequenz und der des Ursprungssignals  $f_T \gg f_{1/2}$  gilt.

Vor und nach dem Tiefpass wurden weitere Graphen für Zeit- und Frequenzbild eingefügt, um die einzelnen Schritte der Signalverarbeitung auch visuell auf dem Frontpanel darzustellen. Ansonsten ist das VI im wesentlichen das gleiche wie in Abb. 3, nur die Speicherfunktion wurde noch um die zwei weiteren Datensätze nach Demodulation und nach Tiefpass erweitert.





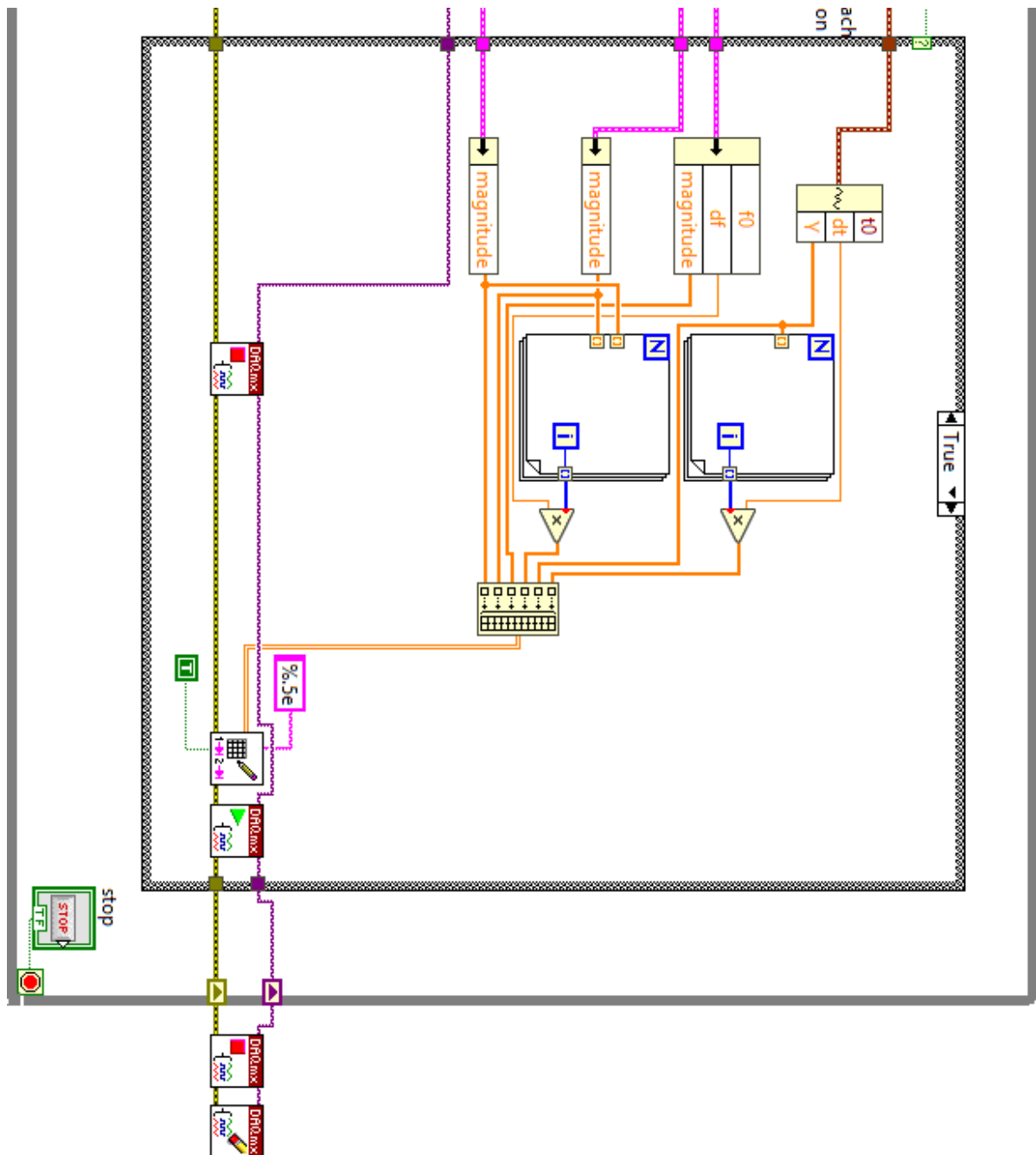
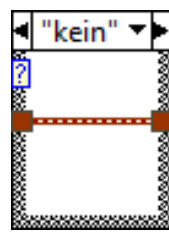
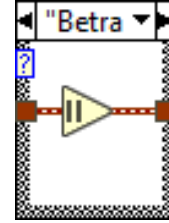


Abbildung 11: Frontpanel des Modulation-VIs bei AM mit Beispielsignal und einstellbaren Parametern.



(a) Case für den Fall ohne Demodulation.

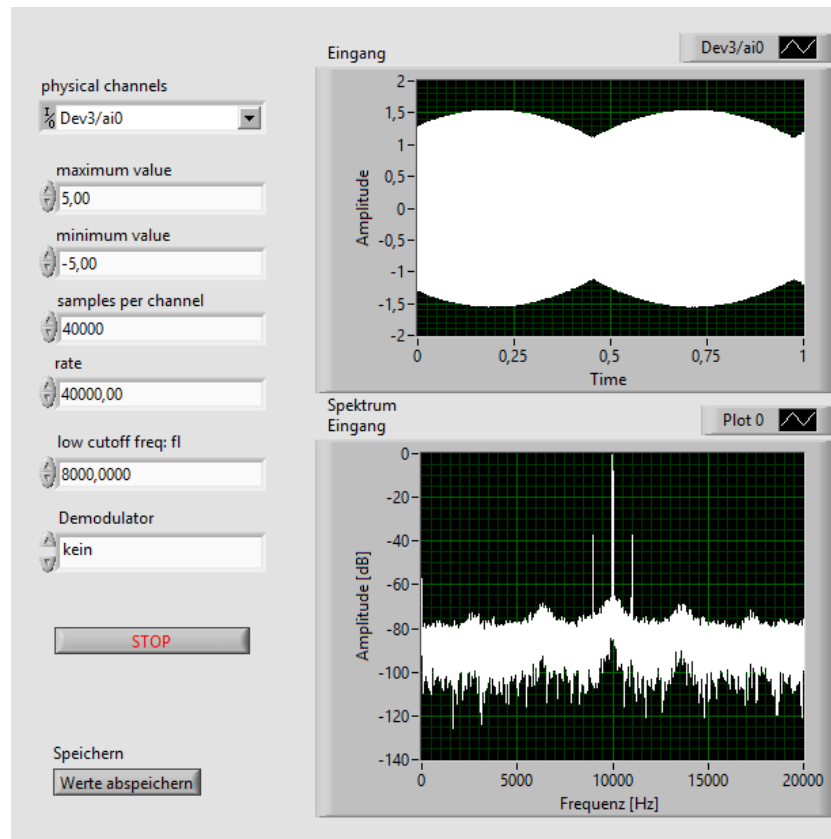


(b) Case für den Fall von Demodulation über Betragsbildung.

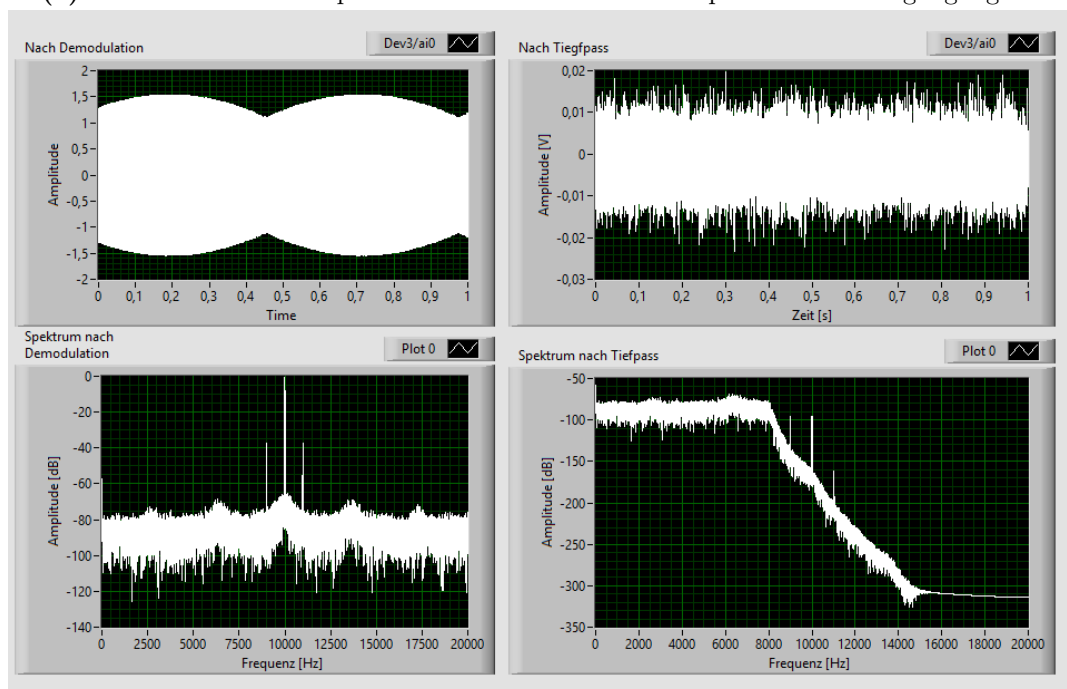
**Abbildung 12:** Alternative Fälle für den Case für die Demodulationsart in Abb. 11.

Wie bereits zuvor erwähnt, war das Ende der Modulations-VI zunächst falsch verkabelt. Die Darstellungen für das Frontpanel der erweiterten Messstruktur wurden deshalb mit Hilfe eines von dem Funktionsgenerator ausgehenden Signals aufgenommen. Diese sind in den Abbildungen 13 bis 15 jeweils einmal pro Demodulationsart aufgefasst. Für alle drei Abbildungen wurde das gleiche Signal von dem Funktionsgenerator verwendet mit Modulationsfrequenz  $f_1 = 1 \text{ kHz}$ ,  $f_2 = 0 \text{ Hz}$ , Trägerfrequenz  $f_T 1 \text{ kHz}$  und Modulationsgrad  $m = 0,03$ . Bei Abb. 13 entsprechen die Graphen nach der Demodulation selbstverständlich denen des Eingangssignals, da dieses nicht weiter verarbeitet wurde. Der Tiefpass erreicht an dieser Stelle nichts. Bei den Abbildungen 14 und 15 hingegen kann man den Frequenzbildern nach der jeweiligen Demodulation entnehmen, dass nur die Frequenz des zu modulierenden Signals  $f_1$ , die Trägerfrequenz  $f_T$  und das doppelte der Trägerfrequenz mit Seitenbändern vorliegt. An dieser Stelle erfüllt der Tiefpass seinen Zweck und in den Frequenzbildern ist die einzige Frequenz, die nicht in dem Rauschen bei  $\leq -50 \text{ dB}$  untergeht gerade die gesuchte Frequenzen des zu modulierenden Signals  $f_1$ .

Weitere Beispiele sind auch für diese im Anhang (Abschnitt 5) vorzufinden.

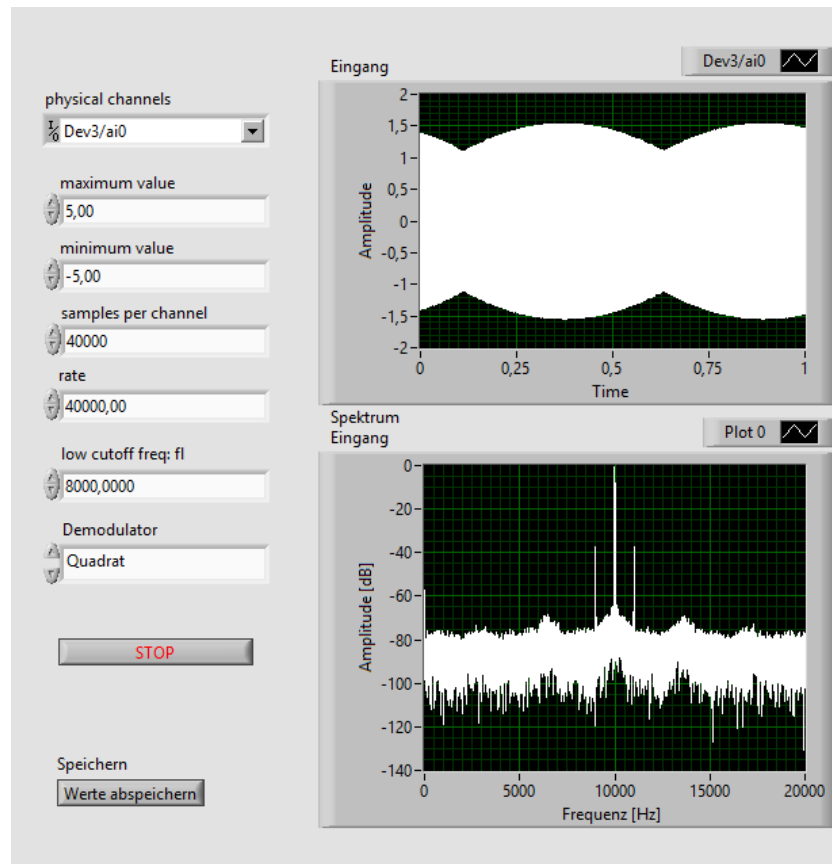


(a) Linke Seite des Frontpanels mit Parametern und Graphen für das Eingangssignal.

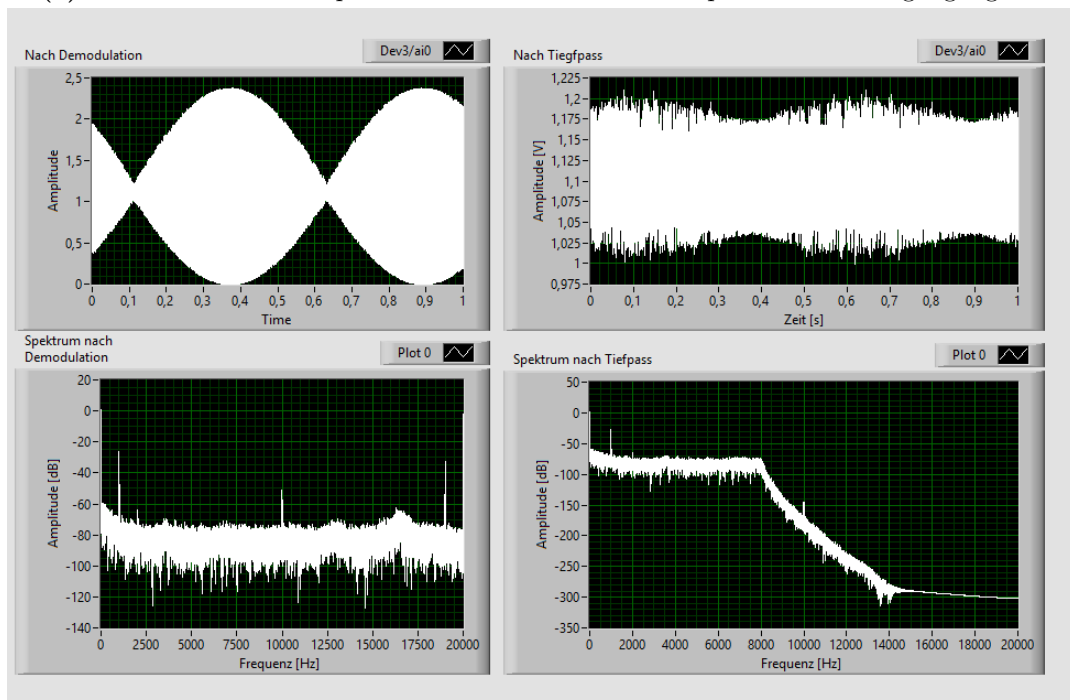


(b) Rechte Seite des Frontpanels mit Graphen für das demodulierte Signal vor und nach Tiefpass.

**Abbildung 13:** Frontpanel der erweiterten Messstruktur bei eingehendem AM-Signal vom Funktionsgenerator mit Modulationsfrequenz  $f_1 = 1 \text{ kHz}$ ,  $f_2 = 0 \text{ Hz}$ , Trägerfrequenz  $f_T 1 \text{ kHz}$  und Modulationsgrad  $m = 0,03$  ohne Demodulation.

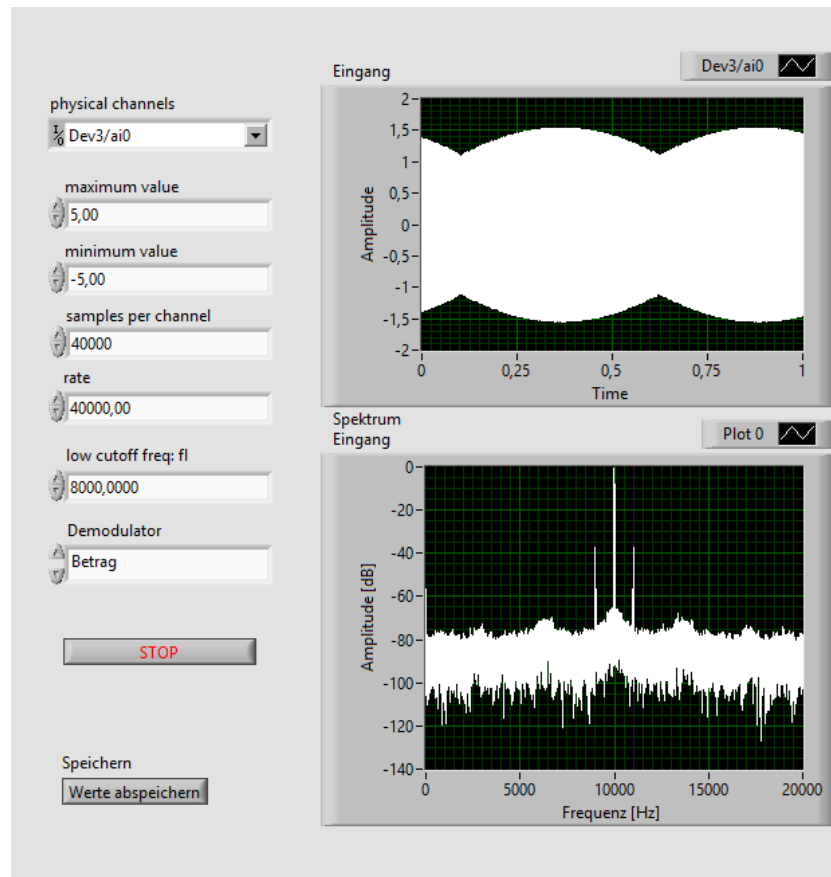


(a) Linke Seite des Frontpanels mit Parametern und Graphen für das Eingangssignal.

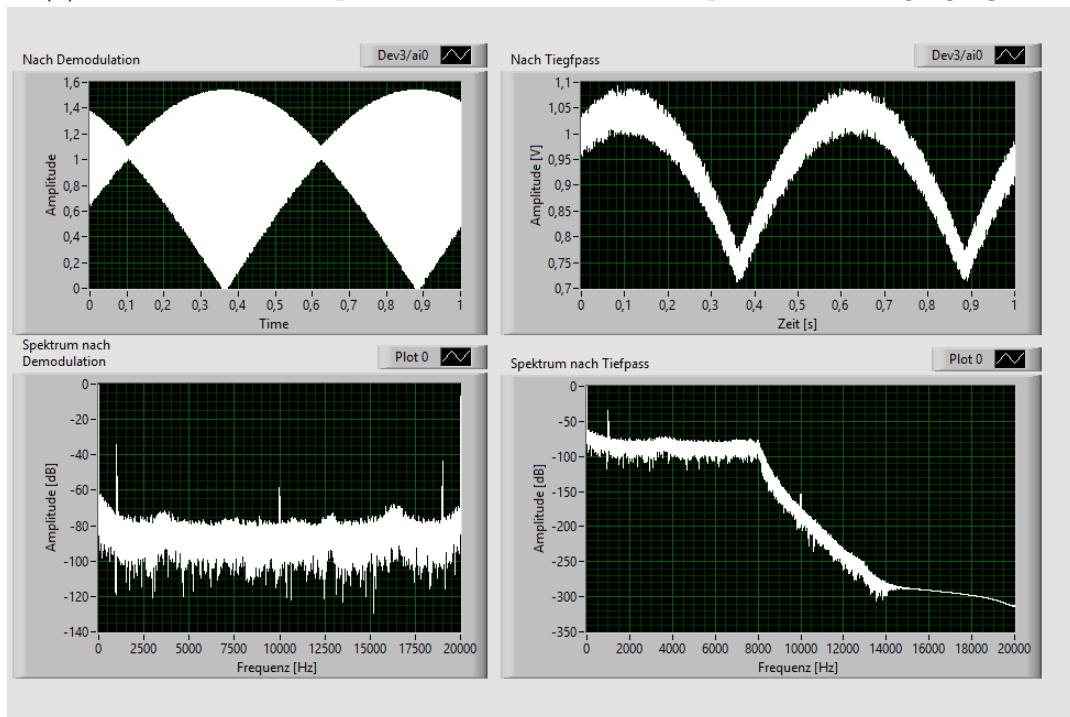


(b) Rechte Seite des Frontpanels mit Graphen für das demodulierte Signal vor und nach Tiefpass.

**Abbildung 14:** Frontpanel der erweiterten Messstruktur bei eingehendem AM-Signal vom Funktionsgenerator mit Modulationsfrequenz  $f_1 = 1 \text{ kHz}$ ,  $f_2 = 0 \text{ Hz}$ , Trägerfrequenz  $f_T 1 \text{ kHz}$  und Modulationsgrad  $m = 0,03$  ohne Demodulation.



(a) Linke Seite des Frontpanels mit Parametern und Graphen für das Eingangssignal.

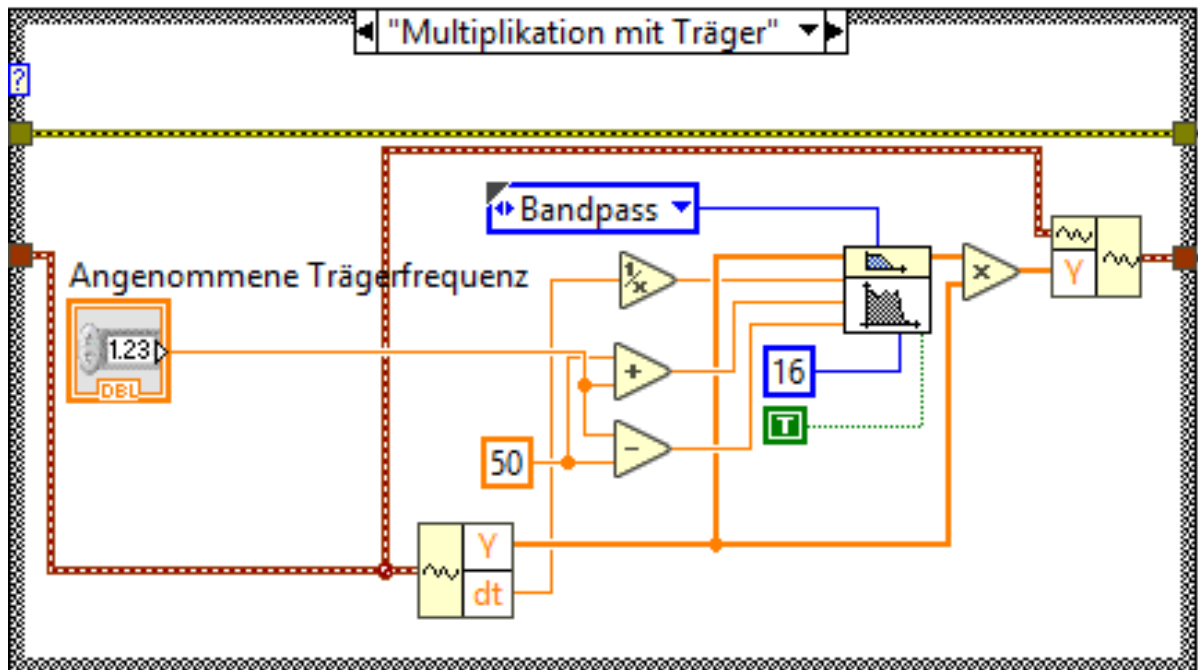


(b) Rechte Seite des Frontpanels mit Graphen für das demodulierte Signal vor und nach Tiefpass.

**Abbildung 15:** Frontpanel der erweiterten Messstruktur bei eingehendem AM-Signal vom Funktionsgenerator mit Modulationsfrequenz  $f_1 = 1 \text{ kHz}$ ,  $f_2 = 0 \text{ Hz}$ , Trägerfrequenz  $f_T 1 \text{ kHz}$  und Modulationsgrad  $m = 0,03$  ohne Demodulation.



Zu Beginn des vierten Tages sollte die Messstruktur nun um die dritte Art der Demodulation, der Multiplikation mit dem Träger, erweitert werden. Dieser neue Case ist in Abb. 16 dargestellt.



**Abbildung 16:** Case für den Fall der Multiplikation mit dem Träger als Demodulationsart.

In diesem Falle muss eine geschätzte Trägerfrequenz angegeben werden, da diese nicht einfach dem eingehenden Signal zu extrahieren ist. Zudem muss, um das Auftreten ungewollter Seitenbänder zu unterdrücken, das Trägersignal vorher mit einem Bandpass so verändert werden, dass die Trägerfrequenz möglichst die einzige verbliebene Frequenz in dem Signal ist, sodass dann bei der Multiplikation mit dem moduliertem Signal die Trägerfrequenz wegfällt und nur die Frequenzen des Ursprungssignals  $S(t)$  überbleibt. Vollständig verschwindet die Trägerfrequenz so jedoch nicht, da auch Frequenzanteile mit doppelter Trägerfrequenz auftreten. Diese lassen sich aber wie zuvor durch den Tiefpass hinter dem Case filtern. Um sich gleichzeitig auch noch um den Gleichanteil, zu kümmern wird dieser Tiefpass hinter dem Case durch einen Bandpass ersetzt, was in Abb. 19 in der vollständigen Messstruktur zu sehen ist.

Bevor es jedoch zu Vervollständigung der Messstruktur durch Erweiterung der Demodulation von frequenz- und phasenmodulierten Signalen kommt, war die nächste Aufgabe zunächst einmal das Modulations-VI um die beiden fehlenden Arten der Modulation zu erweitern. An dieser Stelle ließ sich dann auch der Fehler in der Verkabelung feststellen und die Soundkarte konnte nun die Position des Funktionsgenerators ersetzen. Dabei wurde der Audio-Ausgang des Rechners mit dem ADC dort verbunden, wo vorher der Funktionsgenerator angeschlossen war.

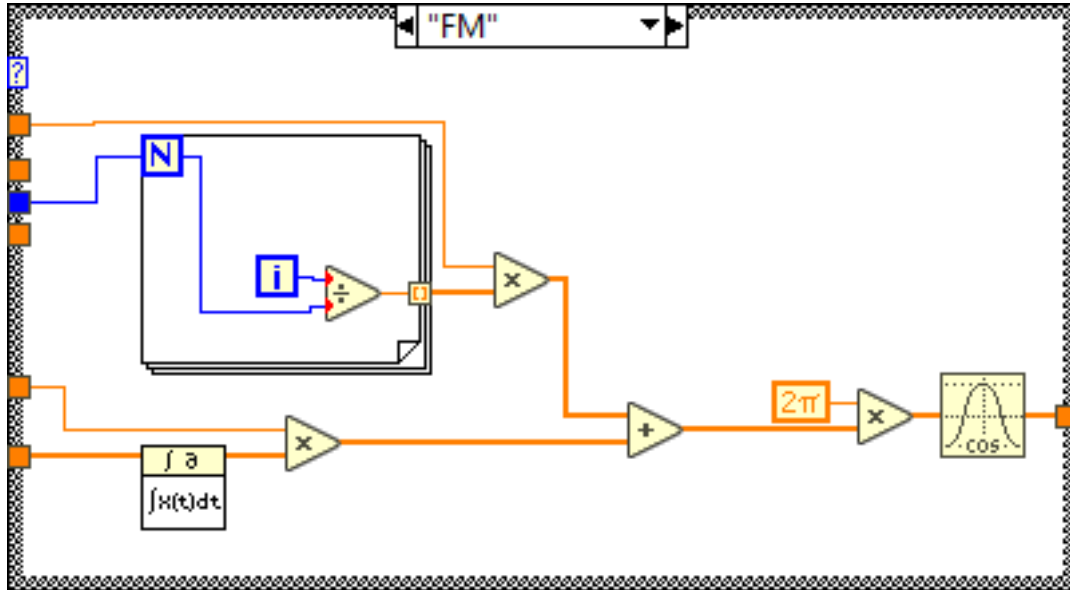
Die Erweiterung des Modulations-VI um Frequenz- und Phasenmodulation belief sich auf die in Abb. 9 nicht dargestellten Cases. Diese sind in der Abbildung 17 dargestellt. Für diese wurden folgende Formeln herangezogen und in LabVIEW modelliert:

$$S_{\text{FM}}(t) = \cos(2\pi[f_{\text{T}}t + k \cdot \int_{-\infty}^t S(\tau) d\tau]) \quad (4.4)$$

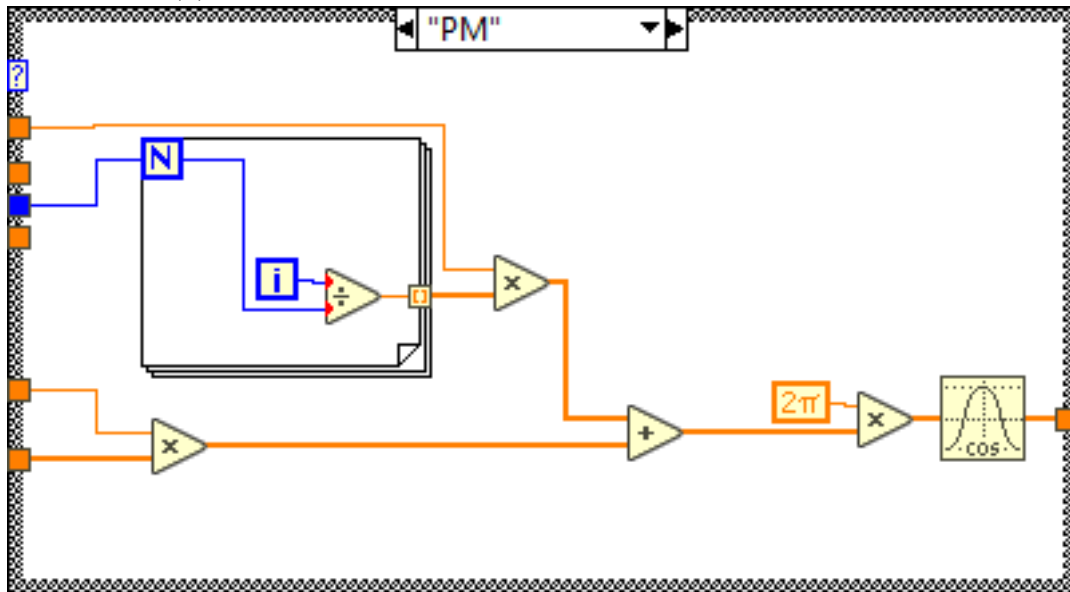
$$S_{\text{PM}}(t) = \cos(2\pi[f_{\text{T}}t + k \cdot S(t)]). \quad (4.5)$$

Wie auch bei dem Abspeichern wird eine for-Schleife verwendet um die Zeit  $t_i$  über den Kehrwert der Messfrequenz in  $i$  Schritten zu konstruieren. Der Modulationsgrad  $m$  aus der Amplitudenmodulation entspricht hier dem  $k$  aus der Formel. Im Wesentlichen unterscheiden sich der Fall der Frequenzmodulation von der Phasenmodulation nur dadurch, dass das zu modulierende Signal vor weiteren Rechenoperationen noch integriert wird. Da die Amplitude für das ausgehende Signal für den Modulationsprozess hier keine große Rolle spielt, solange sie ungleich null ist, wird diese hier auf eins gesetzt. Durch einfache Rechenoperationen wird das Argument des Kosinus aufgestellt und dann in die Funktion gegeben.

Jeweils ein Beispiel des Frontpanels für die beiden neuen Modulationsverfahren ist in Abb. 18a (Frequenzmodulation) bzw. Abb. 18b (Phasenmodulation) dargestellt. Auffällig ist die charakteristische Menge der auftretenden Seitenbänder verglichen mit der Amplitudenmodulation. Auch hierfür sind weitere Beispiele im Anhang (Abschnitt 5) vorzufinden.

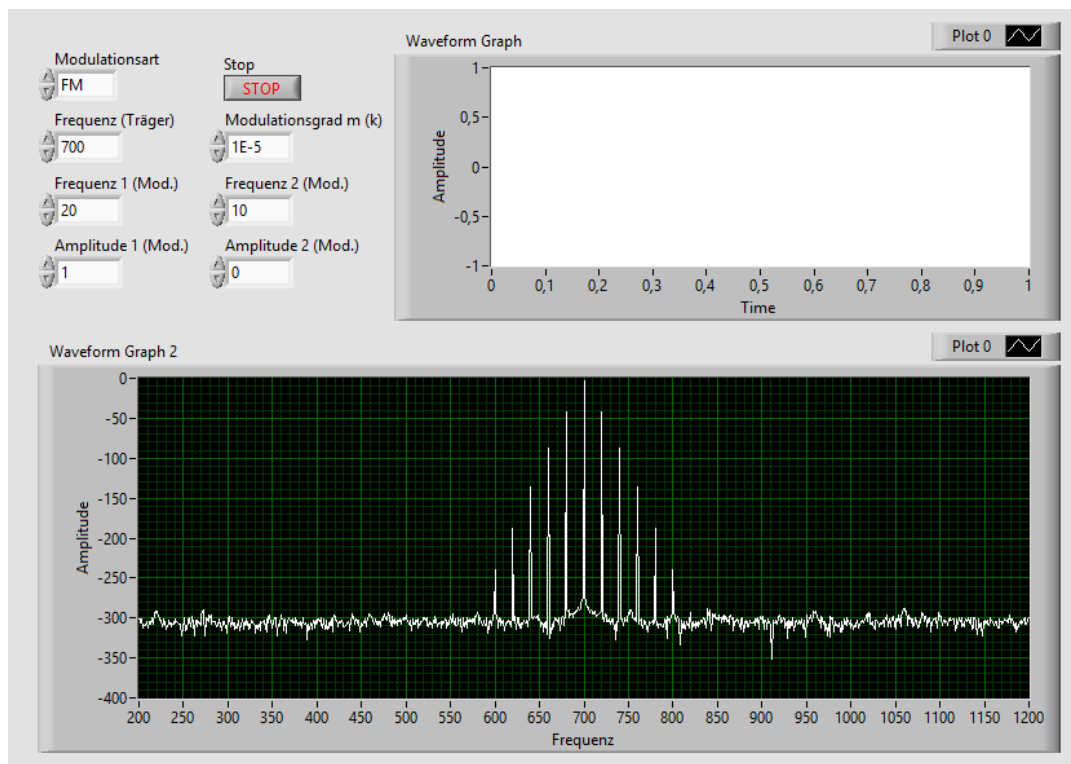


(a) Case für die Berechnung eines frequenzmodulierten Signals.

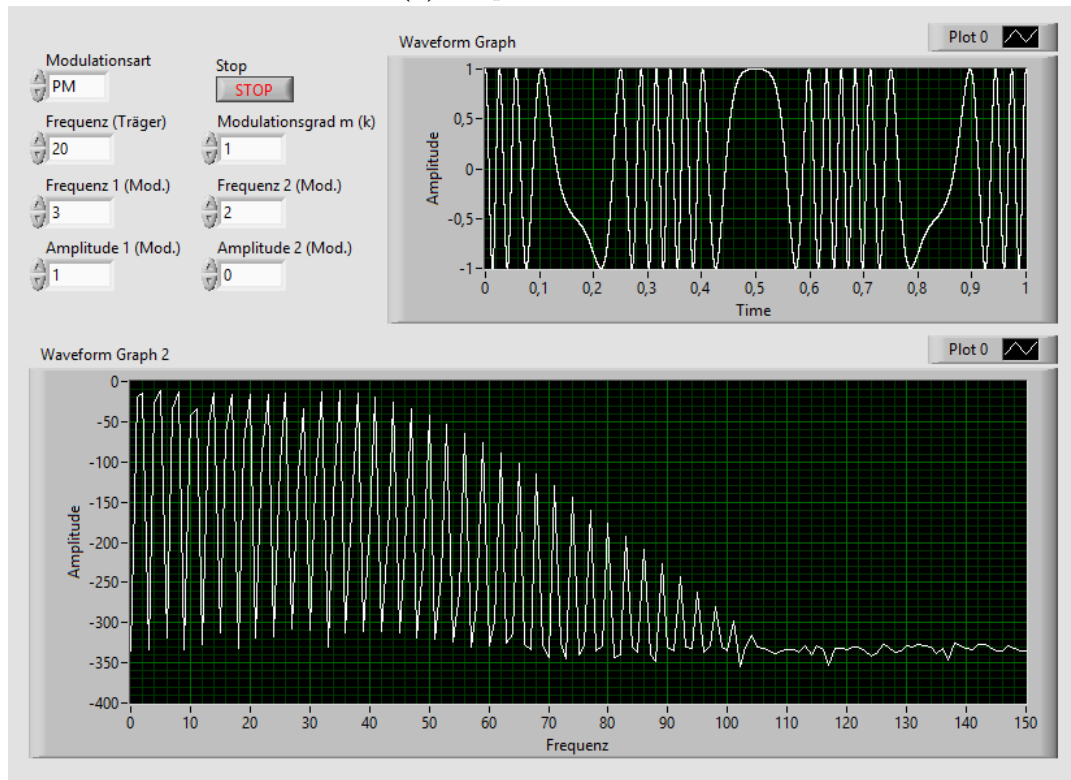


(b) Case für die Berechnung eines phasenmodulierten Signals.

**Abbildung 17:** Alternative Fälle für den Case für die Modulationsart in Abb. 9.



(a) Frequenzmodulation



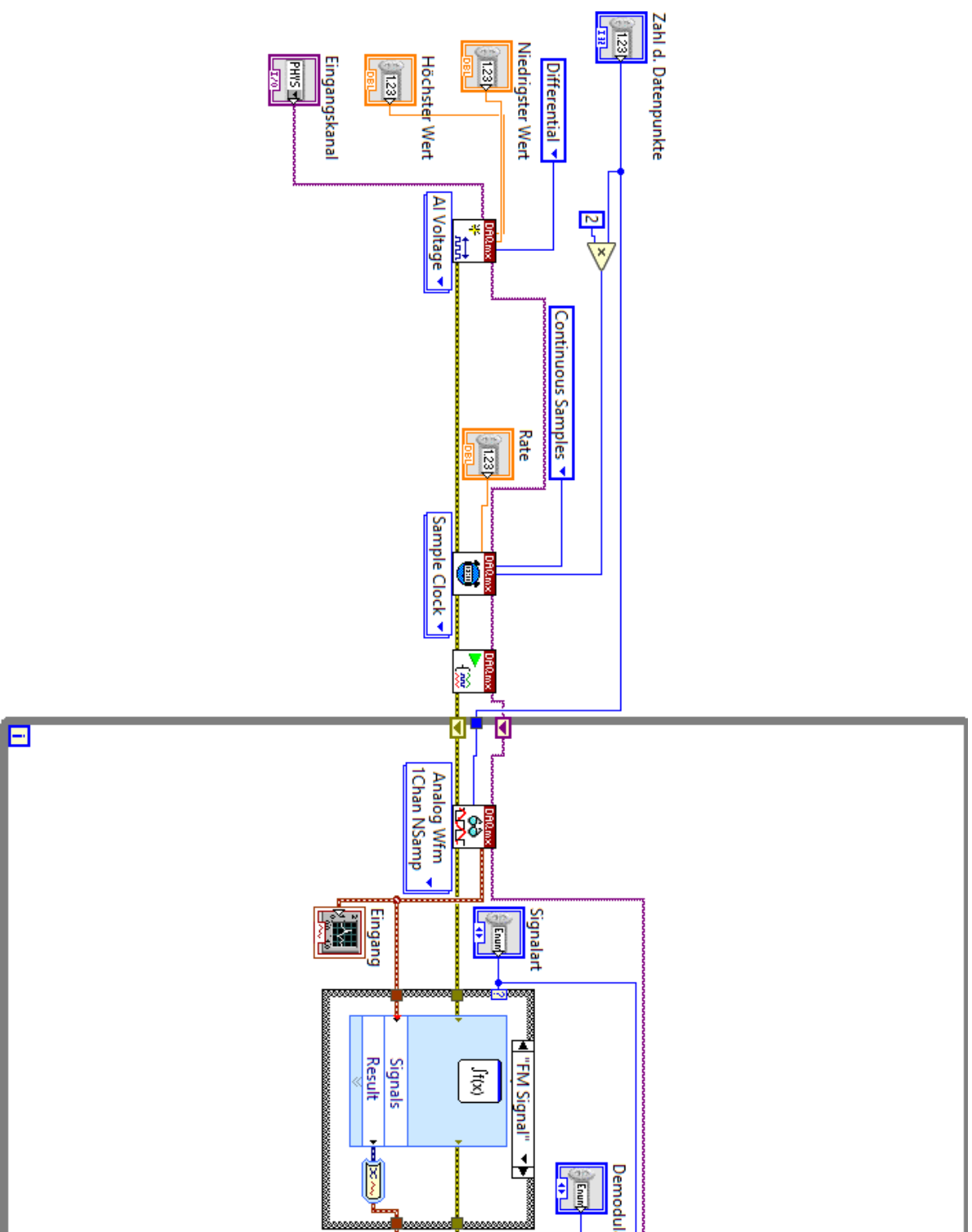
(b) Phasenmodulation

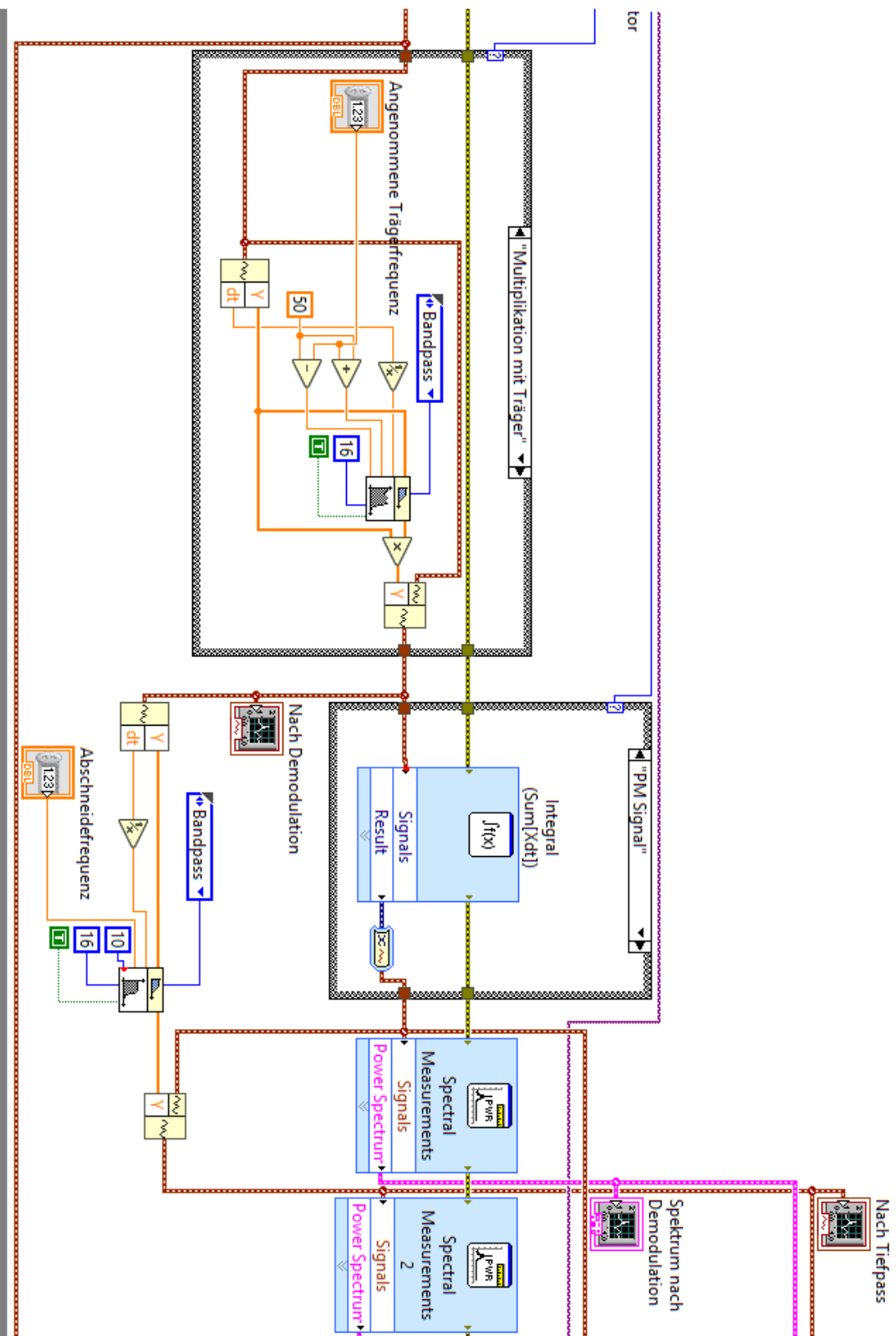
Abbildung 18: Frontpanel des Modulations-VIs bei frequenz- und phasenmoduliertem Signal.

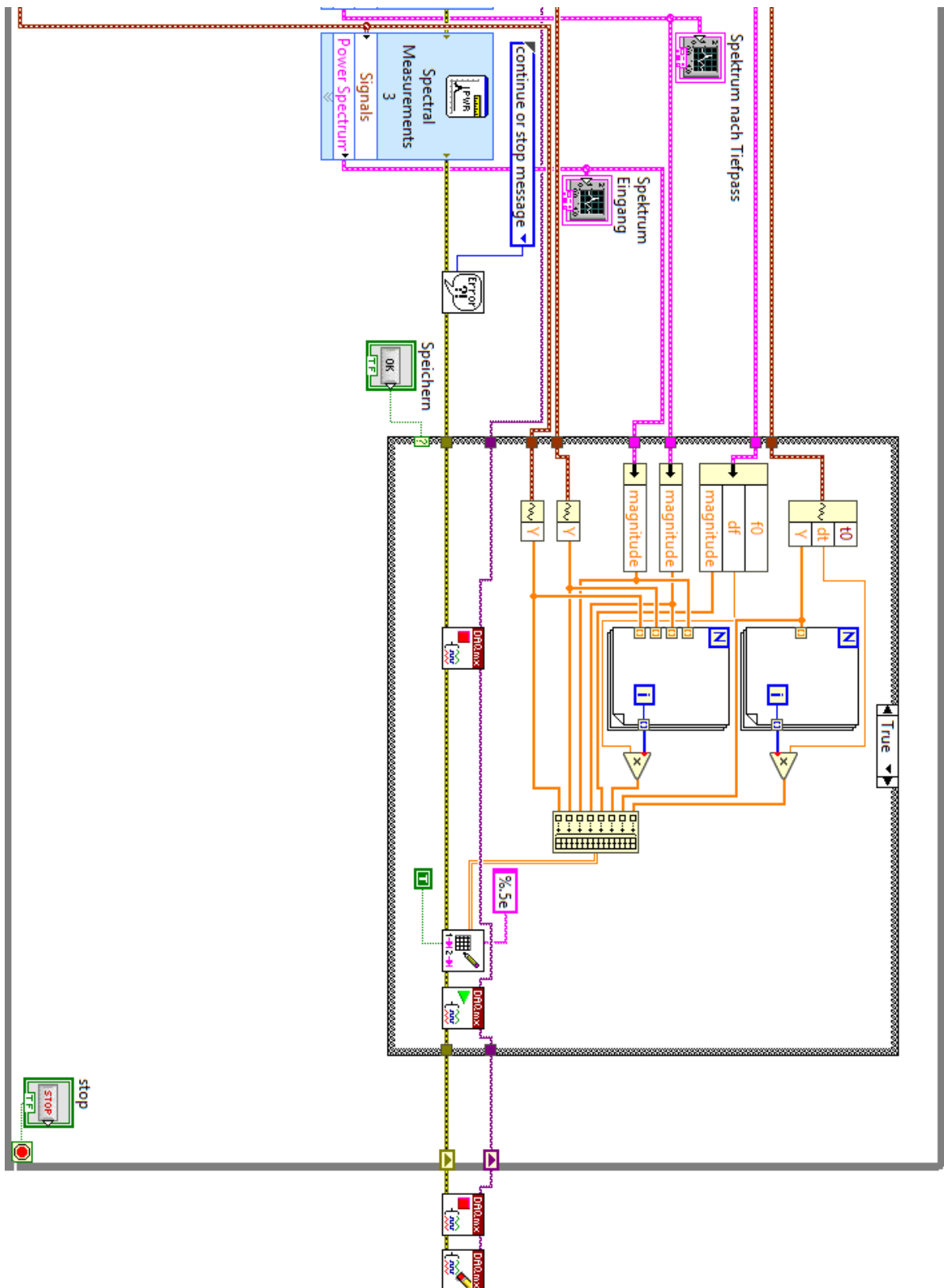
Mit dem nun vollständigen Modulations-VI konnten nun auch Signale erzeugt werden, die von der Messstruktur nicht demoduliert werden können. Die letzte Aufgabe war es die Messstruktur so zu vervollständigen, dass auch frequenz- und phasenmodulierte Signale demoduliert werden können. Für diese letzte Erweiterung mussten nur zwei weitere Cases eingebaut werden. Durch ableiten des Signals, wird das zu modulierende Signal ein Teil der Amplitude, weswegen die beiden anderen Modulationstypen mit den gleichen Verfahren demoduliert werden können wie bei der AM. Abbildung 19 stellt das vollständige Messstruktur-VI dar. Dieses unterscheidet sich von der Abbildung 11 nur in den zwei weiteren Cases, der Erweiterung der Speicherfunktion um die Amplituden in den Zeitbildern nach Demodulation/Tiefpass bzw. Bandpass, welcher den Tiefpass ersetzt um ebenfalls den Gleichanteil zu filtern.

Der erste Case liegt vor der Demodulationsoperation und ist für FM und PM gleich. Hier wird nur das Signal über ein Express-VI abgeleitet und nach der Konvertierung in diesem VI wieder in eine Waveform zurück konvertiert. Im Falle der AM geschieht nichts in dem Case, die Kabel gehen also einfach durch diesen durch. Da nach dem Ableiten bei dem frequenzmodulierten Signal  $S_{\text{FM}}(t)$  das zu modulierende Signal  $S(t)$  nun in der Amplitude steht, kann dieses wie ein amplitudenmoduliertes Signal behandelt werden.

In dem zweiten neuen Case geschieht im Falle der AM und FM also nichts, auch hier verlaufen die Kabel direkt durch den Case. Bei einem phasenmodulierten Signal befindet sich jedoch nicht das zu modulierende Signal  $S(t)$ , sondern dessen Ableitung in der Amplitude, weswegen der zweite Case hier genutzt wird, um das Signal einmal zu integrieren, analog zu der Differentiation in dem ersten Case. Das Frontpanel für ein Beispielsignal ist in Abb. 20 dargestellt. Wie auch bei den anderen Beispielen sind weitere im Anhang (Abschnitt 5) vorzufinden.

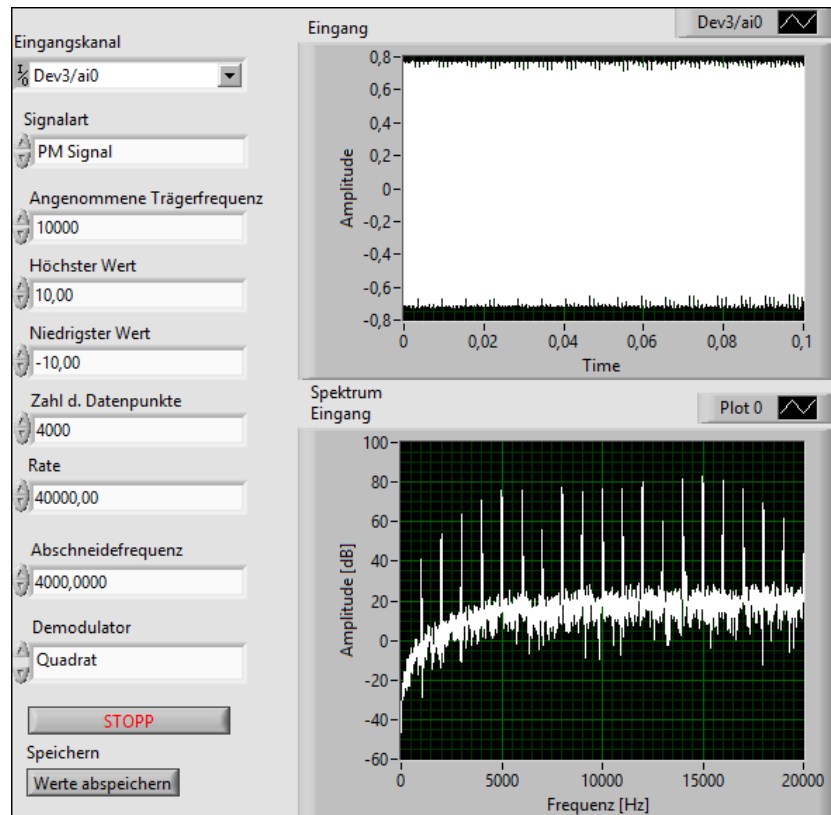




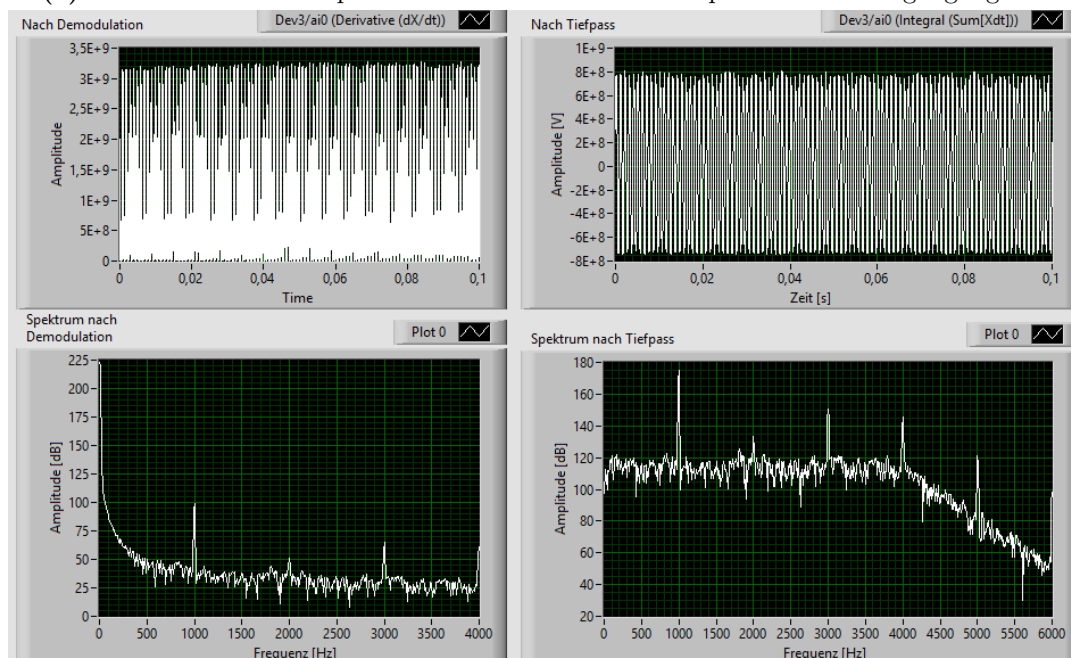


**Abbildung 19:** LabVIEW VI der vollständigen Messstruktur mit Demodulationsoptionen für AM, FM und PM, sowie Speicherfunktion.





(a) Linke Seite des Frontpanels mit Parametern und Graphen für das Eingangssignal.



(b) Rechte Seite des Frontpanels mit Graphen für das demodulierte Signal vor und nach Bandpass.

**Abbildung 20:** Frontpanel der vollständigen Messstruktur bei eingehendem PM-Signal von dem Modulations-VI.

## 5 Anhang