

# CLAD Vorbereitungsleitfaden mit LabVIEW 2017

Dieser Vorbereitungsleitfaden bereitet Sie auf die CLAD-Prüfung mit LabVIEW 2017 vor. Wenn Sie die CLAD-Prüfung mit LabVIEW NXG ablegen möchten, lesen Sie bitte den *Vorbereitungsleitfaden für die CLAD-Prüfung mit LabVIEW NXG*.

## Inhalt

Einführung: Umfang und Grundlage der CLAD-Prüfung.....	2
Quellen für die Vorbereitung .....	2
Die Logistik der CLAD-Prüfung .....	2
Format.....	2
Bewertung.....	2
Planung und Durchführung der Prüfung.....	3
KSAs: Kenntnisse, Fähigkeiten und Fertigkeiten (KSAs), die von CLAD getestet werden .....	3
Themenliste.....	5
Hardware (10 % der Prüfungsfragen) .....	5
LabVIEW-Programmierumgebung (25 % der Prüfungsfragen) .....	5
Grundlagen zum Programmieren mit LabVIEW (50 % der Prüfungsfragen).....	5
Best Practices für das Programmieren (15 % der Prüfungsfragen) .....	7
Beispielfragen und Informationsquellen .....	7
Hinweis zur Verwendung dieser Beispielfragen.....	7
Hardware.....	7
LabVIEW-Programmierumgebung.....	15
LabVIEW-Programmiergrundlagen.....	20
Best Practices für das Programmieren.....	30
ANHANG I: SIMULIERTER DAQ FÜR DIE VORBEREITUNG AUF CLAD.....	34
Simulierte Hardware erstellen .....	34
Die DAQmx-VIs sind in der CLAD-Prüfung enthalten.....	35
ANHANG II: RICHTIGE ANTWORTEN AUF BEISPIELFRAGEN.....	36

## Einführung: Umfang und Grundlage der CLAD-Prüfung

Der Certified LabVIEW Associated Developer (CLAD) ist die erste Stufe der professionellen Zertifizierung für das Programmieren mit LabVIEW. Die Zertifizierung ist in drei allgemeinen Bereichen verankert, in denen LabVIEW im Einsatz ist:

- Automatisierter Test
- Datenerfassung mit hoher Kanalzahl
- Messung und Datenprotokollierung für Domain-Experten

CLAD stellt die Kompetenz für LabVIEW dar, mit der Personen mit geringem Kenntnisstand LabVIEW für Folgendes verwenden können:

- Daten erfassen und interpretieren
- Kleine VIs erstellen
- Mittlere VIs bearbeiten
- Elemente zu großen VIs und Projekten beitragen

## Quellen für die Vorbereitung

Die beste Vorbereitung auf diese Prüfung ist Erfahrung im Programmieren mit LabVIEW unter Anwendung der in den Kursen LabVIEW Core1 und Core2 vorgestellten Konzepte. Die Teilnahme am Kurs allein reicht nicht aus. Typische CLAD-Kandidaten haben die Kurse LabVIEW Grundlagen 1 und Grundlagen 2 (oder gleichwertig) von NI besucht und verfügen über mindestens 6 bis 9 Monate Erfahrung mit LabVIEW.

Dieser Vorbereitungsleitfaden gibt einen Überblick über die Prüfung, einschließlich der Prüfungslogistik und Prüfungsabdeckung. Die im Rahmen der CLAD-Prüfung geprüften relevanten Kenntnisse, Fähigkeiten und Fertigkeiten (KSAs) sind im Abschnitt KSAs: Kenntnisse, Fähigkeiten und Fertigkeiten (KSAs), die von CLAD getestet werden aufgeführt. Eine detaillierte Themenlisten-Tabelle folgt den KSA-Beschreibungen.

Dieser Vorbereitungsleitfaden enthält auch Beispielfragen. **Dies ist keine Beispielprüfung.** Die Fragen sind als Studienhilfe enthalten. Sie spiegeln den Umfang der CLAD-Prüfung wider und machen den Prüfungsteilnehmer mit dem Ansatz von NI zur Prüfung der KSAs vertraut, die für einen Certified LabVIEW Associate Developer relevant sind.

## Die Logistik der CLAD-Prüfung

### Format

Die CLAD-Prüfung ist eine Multiple-Choice-Prüfung mit 40 Fragen. Für jede Frage wählen Sie nur eine der vier aufgeführten Optionen aus. Wenn mehr als eine Antwort richtig erscheint, wählen Sie die beste aus. Also die Antwort, die entweder einen zusätzlichen Nutzen bringt oder Probleme verhindert, die andere Antworten verursachen könnten.

### Bewertung

Für das Bestehen der Prüfung ist eine Punktzahl von 70 % oder mehr erforderlich.

## Planung und Durchführung der Prüfung

Einzelpersonen können die CLAD-Prüfung in jedem PearsonVUE-Prüfungszentrum ablegen. Dazu müssen sie die Prüfung unter <http://www.pearsonvue.com/ni/> planen. Wenn Sie die Prüfung planen, geben Sie an, ob Sie die Prüfung mit LabVIEW NXG oder LabVIEW 20xx (wobei xx das aktuelle Jahr angibt) ablegen möchten. Die Bilder und die Terminologie der Prüfung spiegeln den von Ihnen gewählten LabVIEW-Editor wider, der Prüfungsinhalt ist aber ansonsten identisch. Wählen Sie einfach den LabVIEW-Editor, mit dem Sie sich am wohlsten fühlen. Das Bestehen der CLAD-Prüfung mit einem der beiden Editoren führt zur gleichen CLAD-Zertifizierung.

Während der Prüfung ist die Verwendung von LabVIEW oder anderen Hilfsmitteln nicht gestattet. Die CLAD-Prüfung enthält Screenshots aus der LabVIEW-Umgebung und der LabVIEW-Hilfe, falls zutreffend. Beispiele finden Sie im Abschnitt Beispielfragen und Informationsquellen.

Um die Prüfung zu absolvieren, müssen Sie einer Geheimhaltungsvereinbarung (NDA) zustimmen. Die NDA erklärt, dass Sie keinen Teil der Prüfung in irgendeiner Form, einschließlich mündlich oder elektronisch, kopieren, reproduzieren oder kommunizieren werden. Die Nichteinhaltung der NDA führt zu Strafen, die von einer nicht bestandenen Prüfung bis hin zum lebenslangen Verbot der LabVIEW-Zertifizierung reichen.

## KSAs: Kenntnisse, Fähigkeiten und Fertigkeiten (KSAs), die von CLAD getestet werden

Die CLAD-Prüfung konzentriert sich auf Aufgaben, die in professionellen Umgebungen programmiert werden müssen, einschließlich automatisiertem Test, Datenerfassung mit hoher Kanalzahl oder Domain-Support. Die meisten dieser Aufgaben beinhalten das Sammeln und/oder Verarbeiten von Daten in Form von Signalen von Sensoren. Zur Überprüfung der relevanten Kenntnisse, Fähigkeiten und Fertigkeiten verwendet CLAD ein NI-DAQmx-System als repräsentative Hardware. NI-DAQmx wurde gewählt, weil es auf allen Versionen von LabVIEW simuliert werden kann. Wenn Sie Zugriff auf LabVIEW haben, haben Sie die Informationsquellen, um sich auf die Prüfung vorzubereiten. Sie müssen keine spezielle Hardware kaufen.

Anhang I des Dokuments enthält Anweisungen zum Einrichten von simulierter Hardware, die für die Prüfungsvorbereitung ausreichend ist. Der Anhang enthält auch eine Liste der DAQmx-Funktionen, die möglicherweise in der Prüfung verwendet werden. Die Prüfung testet keine für die DAQmx-Funktionen spezifischen Einstellungen. In den Fragen wird NI-DAQmx verwendet, um Kenntnisse und Fähigkeiten für gängige Datenerfassungsaufgaben zu testen, wie z. B. die Berechnung der Abstraten, die Bestimmung der korrekten Reihenfolge der Operationen und die Programmierung grundlegender Datei-I/O.

Eine Person, die LabVIEW auf dem Level eines Associated Developer verwendet, ist zu Folgendem in der Lage:

- Softwarearchitekturen von einem einzelnen VI bis hin zu einem einfachen Zustandsautomaten oder ereignisgesteuerten UI-Handler zu verwenden.
- Daten von Sensoren mit NI-Hardware sammeln.
- Array-Funktionen umfassend zu verwenden, um einen einzelnen Datenkanal aus mehrkanaligen Daten zu extrahieren und zu manipulieren, die durch ein 1D-Signalform-Array oder ein 2D-Numerik-Array dargestellt werden.

- Schleifen zu verwenden, um einen Test über eine bestimmte Anzahl von Durchläufen auszuführen oder bis eine Bedingung erfüllt ist, um eine Spannungsrampe zu erstellen oder andere wiederholte Aufgaben durchzuführen.
- SubVIs, Cluster und Typendefinitionen zu erstellen und zu ändern, um ihren Code zu vereinfachen und zu größeren Projekten beizutragen.

## Themenliste

Ein zertifizierter LabVIEW Associate Developer sollte mit den folgenden Themen vertraut sein.

### Hardware (10 % der Prüfungsfragen)

- Angeschlossene Hardware: Sensoren, Messgeräte, Prüflinge (DUT).
- Ein Signal erfassen und validieren
- Signale verarbeiten
- Geeignete Abtastraten verwenden

### LabVIEW-Programmierumgebung (25 % der Prüfungsfragen)

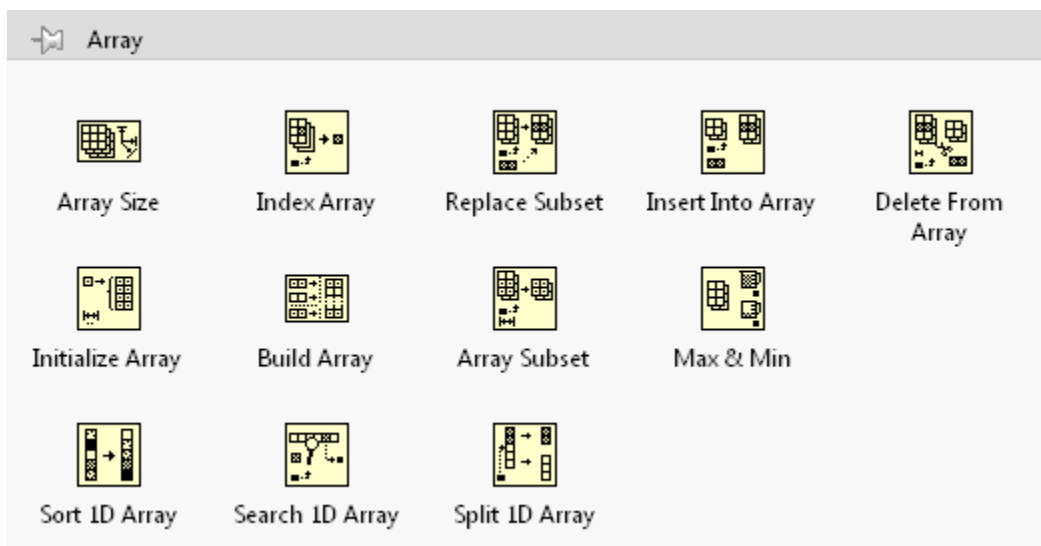
- Ein LabVIEW-Projekt einrichten und verwenden, um:
  - Elemente hinzuzufügen, zu löschen und zu verschieben
  - Bibliotheken und geeignete Ordnertypen zu verwenden
  - Cross-Linking zu vermeiden
- Datentypen:
  - Datentypen auf dem Frontpanel erkennen
  - Datentypen im Blockschaltbild von Anschlüssen und Leitungen erkennen
  - Geeignete Bedien- und Anzeigeelemente, Datentypen und Funktionen für ein bestimmtes Szenario wählen
- Ausführungsreihenfolge und Verhalten vorhersagen von
  - Einer nicht Schleifen-VI
  - Einem einfachen Zustandsautomaten
  - Einem ereignisgesteuerten UI-Handler
  - Parallele Schleifen (ohne Warteschlangen)
- Grundfunktionen zur Erstellung einer einfachen Erfassen-Analysieren-Visualisieren-Anwendung verwenden
- Fehlersuche durch Identifizierung und Behebung der Ursache für eine unterbrochene Schaltfläche oder falsche Daten
- Fehlerbehandlung mit Hilfe von Fehlerclustern und Zusammenführen von Fehlerfunktionen, um eine gute Fehlerbehandlung zu gewährleisten
- In LabVIEW navigieren, um mehr Informationen über Eingänge, Ausgänge und Funktionen zu erhalten

### Grundlagen zum Programmieren mit LabVIEW (50 % der Prüfungsfragen)

- Schleifen
  - Eine kontinuierliche HW-Erfassungs- oder Erzeugungsschleife erstellen, indem Sie ein Öffnen-Konfigurieren-Durchführen-Operation-Schließen-Modell anwenden.
  - Daten in Schieberegistern speichern
  - Ein- und Ausgangsanschlüsse effektiv verwenden, einschließlich:
    - Die letzte Wertausgabe ermitteln
    - Ein- und Ausgangsanschlüsse indizieren
    - Ausgang verknüpfen
    - Bedingten Ausgang verwenden
    - Schieberegister verwenden, sowohl initialisiert als auch nicht initialisiert

- Timing der Schleifen richtig verwenden, einschließlich:
  - Software-Timing
  - Hardware-Timing
- For- und While-Schleifen richtig verwenden
- Arrays
  - Daten von einem n-Kanal-HW-Erfassung-VI (mit DAQmx Lesen-VI) unter Verwendung eines Signalformgraphen, eines Signalform-Diagramms oder einer numerischen/Signalform-Array-Anzeige anzeigen.
  - Einen einzelnen Datenkanal (Signalform oder 1D-Array) extrahieren von einem:
    - 1D-Signalform-Array, das die erfassten Daten von mehreren Kanälen darstellt
    - 2D numerisches Array, das die erfassten Daten von mehreren Kanälen darstellt
    - 1D numerisches Array, das eine Einzelmessung von mehreren Kanälen darstellt
  - Eine For-Schleife mit automatischer Indexierung und bedingten Tunneln verwenden für:
    - Durch ein Array iterieren
    - Verarbeitungscode auf jedem Datenkanal in einem 1D-Signalform-Array iterieren
    - Ein Daten-Array erstellen, das die erforderlichen Bedingungen erfüllt
  - Nach Ansicht identifizieren und das Verhalten der folgenden Array-Funktionen und VIs verwenden und vorhersagen:
 

<ul style="list-style-type: none"> <li>▪ Array-Größe</li> <li>▪ Array indizieren</li> <li>▪ Abschnitt ersetzen</li> <li>▪ In Array einfügen</li> <li>▪ Aus Array entfernen</li> <li>▪ Array initialisieren</li> </ul>	<ul style="list-style-type: none"> <li>▪ Array erstellen</li> <li>▪ Teil-Array</li> <li>▪ Max &amp; Min</li> <li>▪ 1D-Array sortieren</li> <li>▪ 1D-Array durchsuchen</li> <li>▪ 1D-Array teilen</li> </ul>
---	---



- Bedingten Code schreiben, um eine Aktion basierend auf dem Wert einer Benutzereingabe oder einem Messergebnis auszuführen.
- Daten lesen in Dateien schreiben
  - Das Modell Öffnen/Handeln/Schließen für die Datei-I/O verwenden
  - Daten mithilfe von High-Level-Datei-I/O-Funktionen in eine Textdatei schreiben
  - Daten kontinuierlich in eine Textdatei oder eine TDMS-Datei streamen
  - Daten an eine bestehende Datendatei anhängen
  - Daten mit einfachen VIs protokollieren
  - Daten aus DAQmx-Funktionen erfassen
  - Daten in einem Graphen anzeigen
  - Daten in eine CSV-Datei speichern
  - Konfigurationen für Einzelmessungen/mehrere Kanäle und Einzelkanal/Mehrfachmessungen richtig wählen

### Best Practices für das Programmieren (15 % der Prüfungsfragen)

- SubVIs – Wiederverwendung von Programmcode
  - SubVIs erstellen, um die Lesbarkeit und Skalierbarkeit von VIs zu verbessern
  - Das subVI-Anschlussfeld unter Verwendung von Best Practices konfigurieren
  - Geeigneten Programmcode als SubVI-Quelle auswählen
- Clusters – Gruppierung von Daten gemischter Datentypen
  - Clusterdaten in gängigen Szenarien erstellen, bearbeiten, analysieren und verwenden
  - Zusammengehörige Daten durch Erstellen eines Clusters gruppieren, um die Datenorganisation und die VI-Lesbarkeit zu verbessern
- Typendefinitionen – Datentypänderungen propagieren
  - Typendefinitionen erstellen und Typendefinitionen an mehreren Stellen verwenden
  - Typendefinitionen aktualisieren, um Änderungen an alle Instanzen der Typendefinition zu propagieren

### Beispielfragen und Informationsquellen

#### Hinweis zur Verwendung dieser Beispielfragen

Dieser Abschnitt ist keine Beispielprüfung. Es handelt sich um eine Reihe von Fragen, die als Leitfaden zur Verfügung gestellt werden. Diese Fragen dienen hauptsächlich für Folgendes:

1. Erwägen Sie die Frage und notieren Sie, was Ihrer Meinung nach die richtige Antwort ist.
2. Erstellen Sie alle VIs, die in einer Frage dargestellt sind.
3. Bestätigen Sie mit dem VI Ihre Antwort.
4. Ändern Sie Parameter und andere Aspekte der VIs, um sicherzustellen, dass Sie das Thema und den Sachverhalt der Frage verstehen.

### Hardware

#### Unterthemen

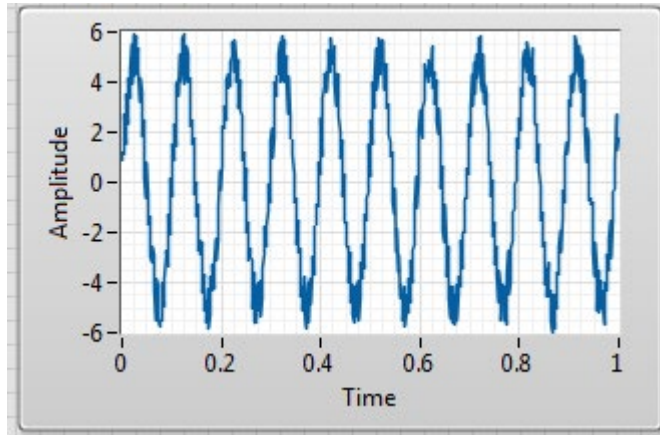
- Angeschlossene Hardware: Sensoren, Messgeräte, Prüflinge (DUT).
- Ein Signal erfassen und validieren
- Signale verarbeiten
- Geeignete Abstraten verwenden

#### Informationsquellen

- LabVIEW-Hilfe -- Messungen durchführen

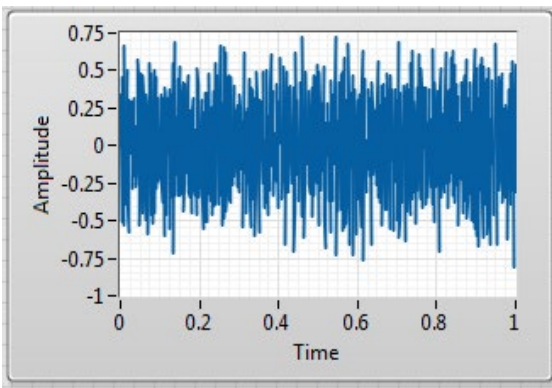
- LabVIEW-Hilfe -- *Aliasing*
- Hilfe zu NI-DAQmx
  - *NI-DAQmx-Grundlagen*
  - *Grundlagen der Messtechnik*
- NI Whitepaper zu Abtastraten und Aliasing
- NI Whitepaper „[So richten Sie ein akademisches Labor für die Datenerfassung ein](#)“

*Beispielfragen*

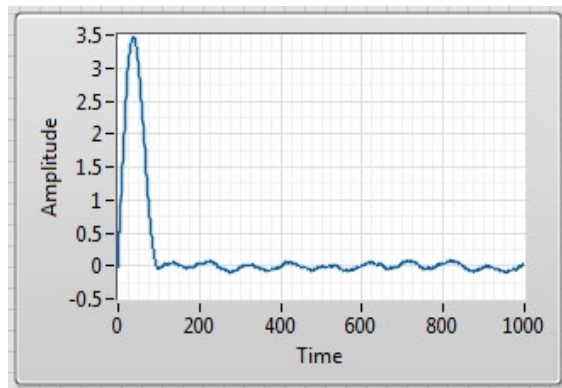


F1: Welcher Graph stellt angesichts dieser Eingangsdaten am besten einen Tiefpassfilter dar?

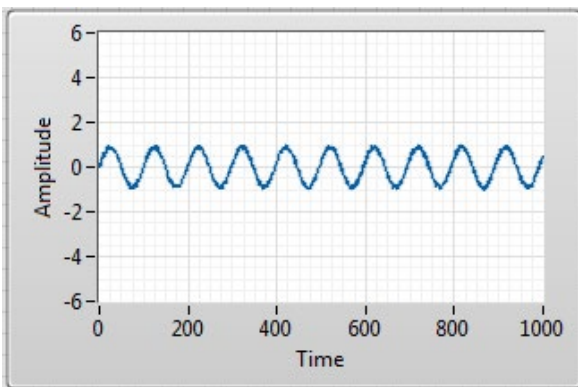
A:



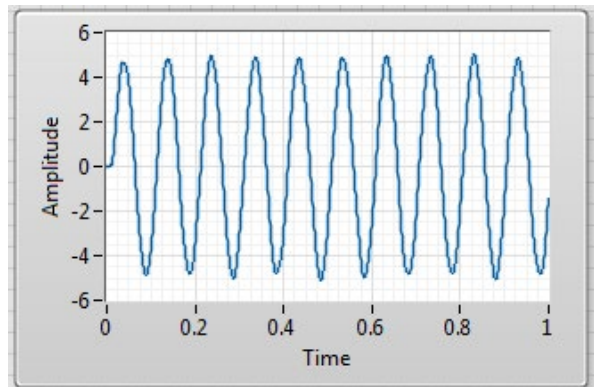
C:



B:

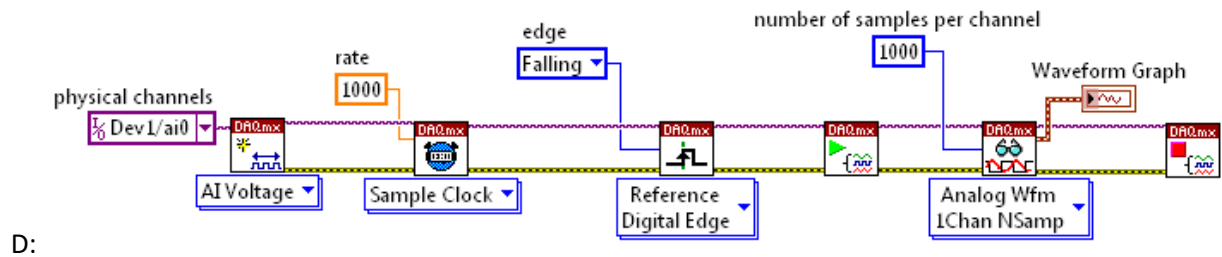
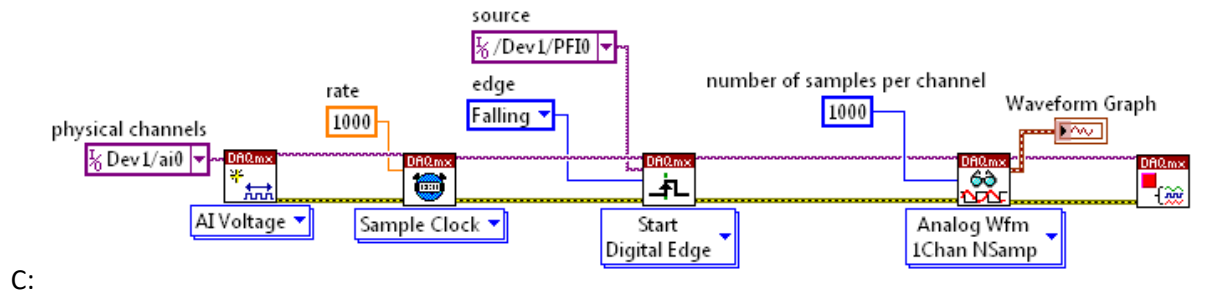
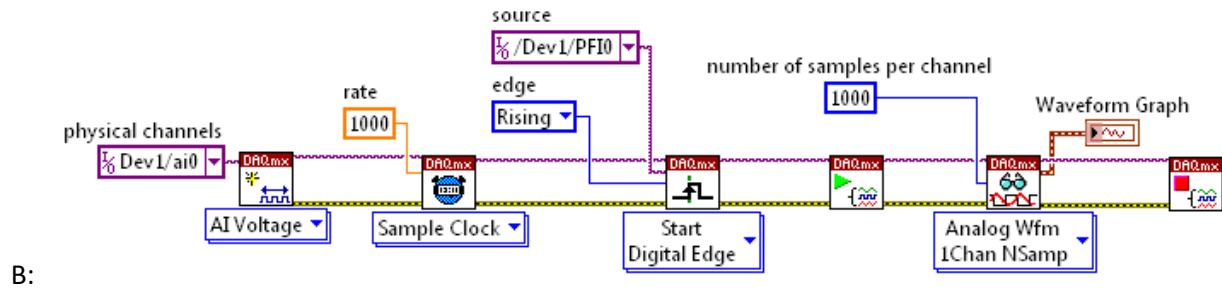
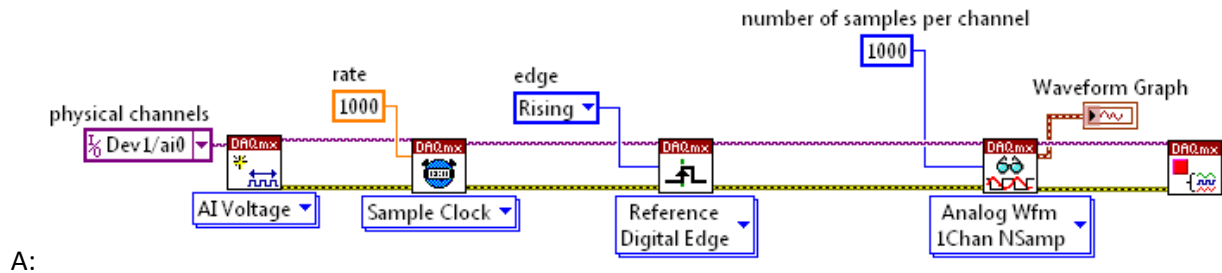


D:





F2: Welches VI erfasst 1.000 Abtastwerte unmittelbar nach dem PFI0-Übergang von niedrig zu hoch?

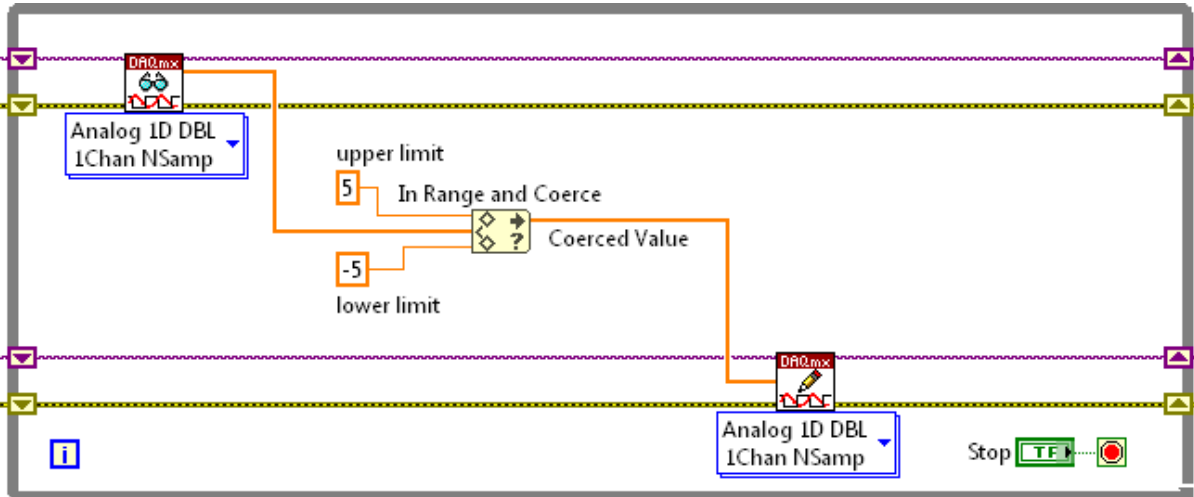


F3: Die DAQmx-Lesen-Aufgabe ist so konfiguriert, dass sie kontinuierlich zwischen -10 V und +10 V liest.

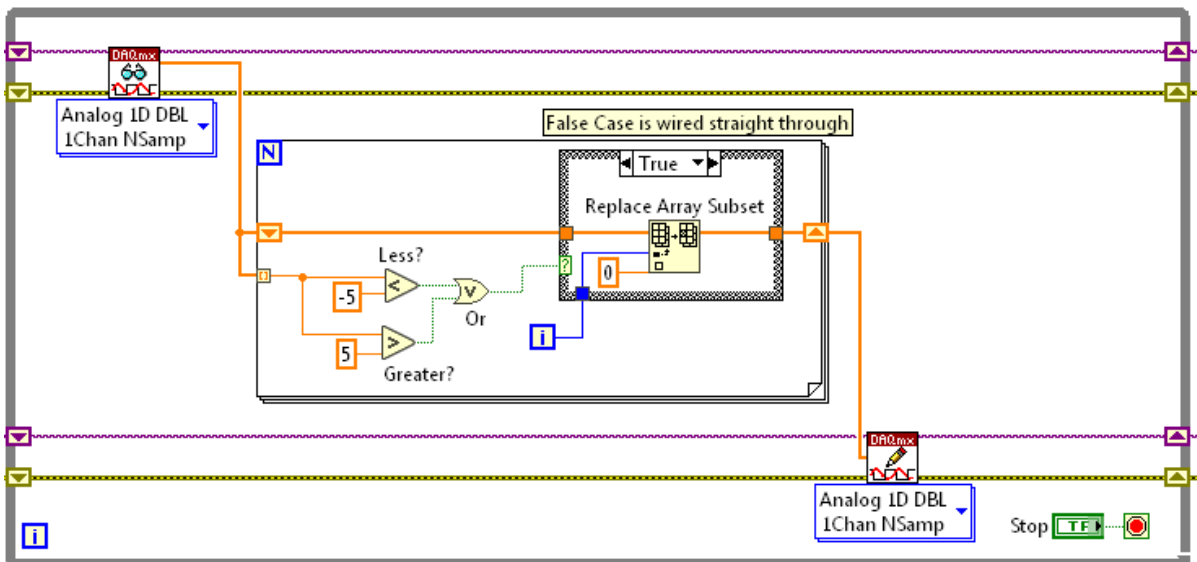
Die DAQmx-Schreiben-Aufgabe ist so konfiguriert, dass sie kontinuierlich zwischen -5 V und +5 V ohne Regeneration schreibt.

Sowohl die Lesen- als auch Schreiben-Aufgabe erfassen mit 1 KHz, und die Regeneration ist auf „keine“ eingestellt.

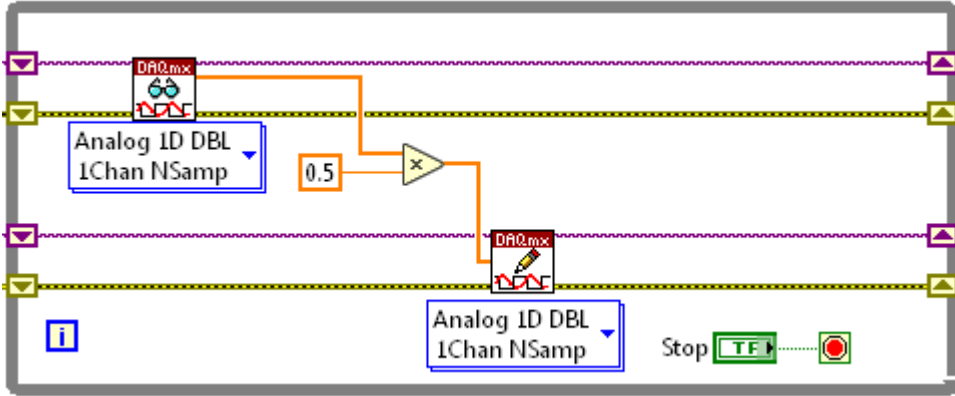
Wenn das gelesene Signal Abtastwerte über den gesamten +/-10 V-Bereich aufweist, welcher VI-Ausschnitt verursacht schließlich einen Pufferunterlauffehler?



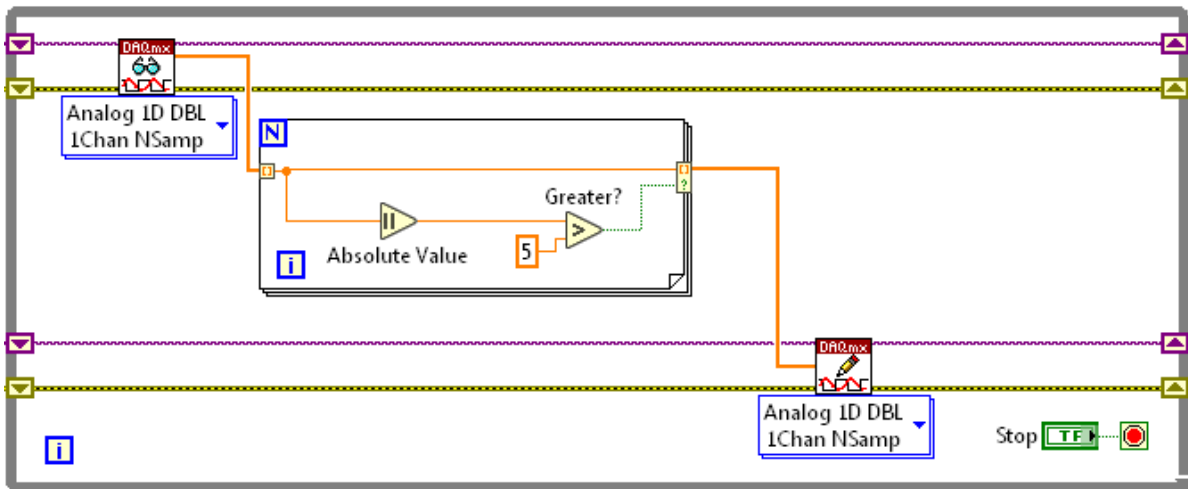
A:



B:

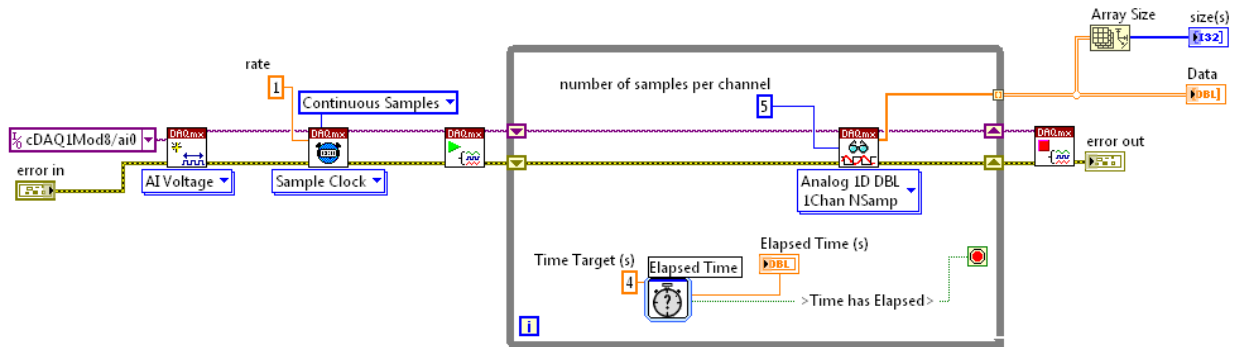


C:



D:

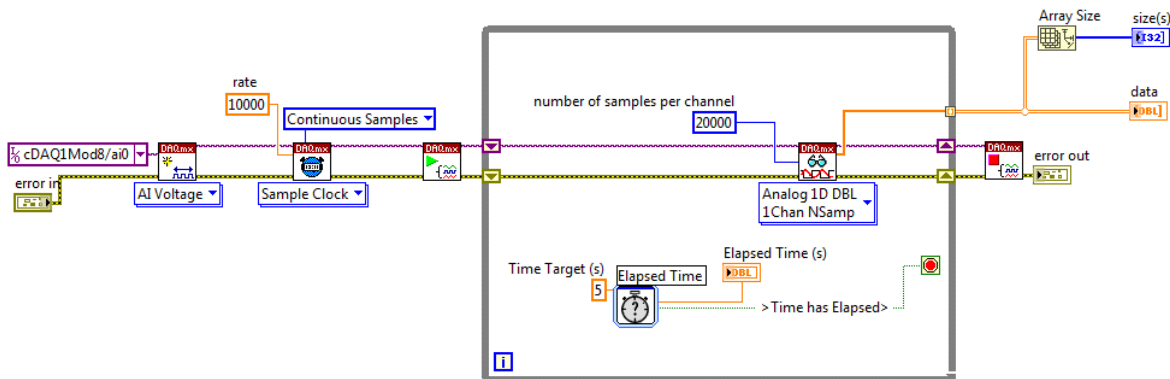
F4: Nachdem dieses VI ausgeführt wurde, wie groß ist das Array in der Datenanzeige?



- A: 5 Zeilen und 4 Spalten.
- B: 2 Zeilen und 5 Spalten.
- C: 4 Zeilen und 5 Spalten.
- D: 1 Zeile und 5 Spalten.

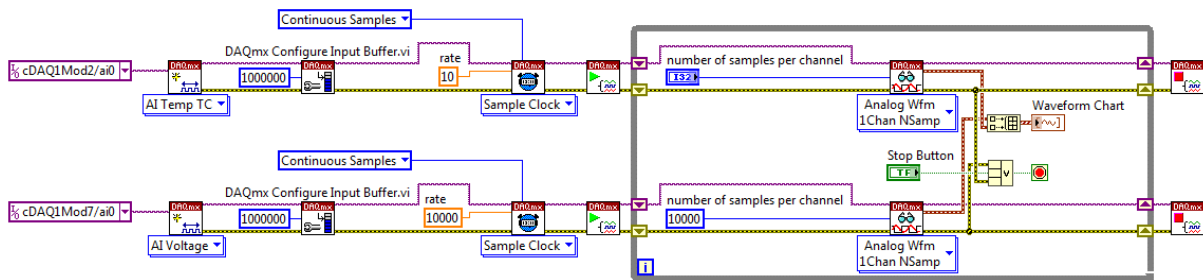
Hinweis: Versuchen Sie verschiedene Wertekombinationen für **Rate**, **Anzahl der Abtastwerte pro Kanal** und **Zeitziel**.

F5: Nachdem dieses VI ausgeführt wurde, wie groß ist das Datenanzeige-Array?



- A: 2 Zeilen, 20.000 Spalten
- B: 3 Zeilen, 20.000 Spalten
- C: 4 Zeilen, 20.000 Spalten
- D: 5 Zeilen, 20.000 Spalten

F6: Welcher Wert muss in das Bedienelement **Anzahl der Abtastwerte pro Kanal** eingegeben werden, damit dieses VI kontinuierlich ausgeführt wird, bis die **Schaltfläche Stopp** gedrückt wird?



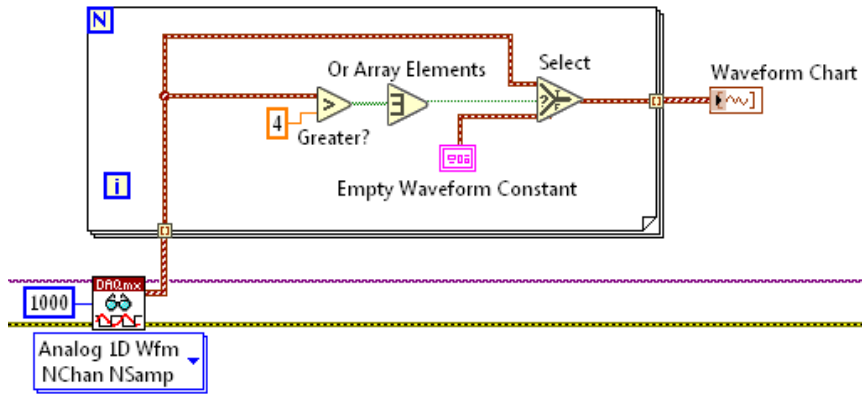
A: 10

B: 1.000

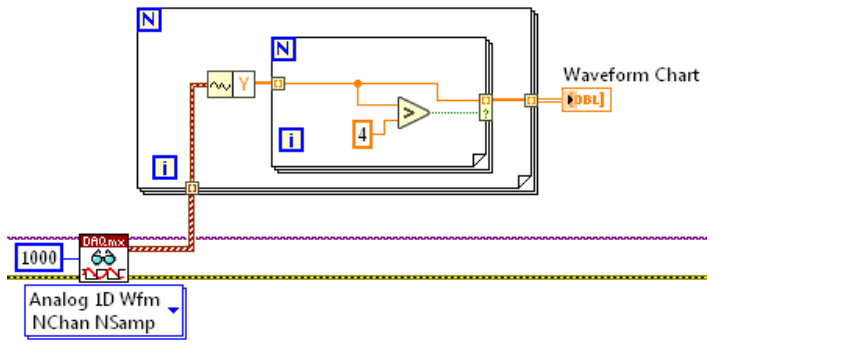
C: 10.000

D: 1.000.000

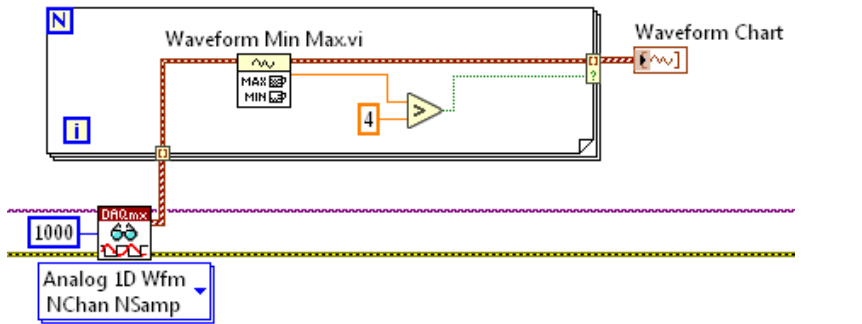
F7: Welches VI gibt nur alle Kanäle aus, die einen Wert größer als 4 haben?



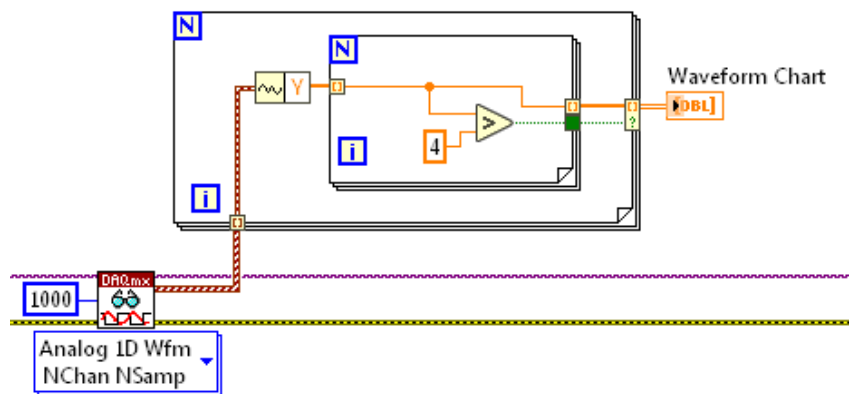
A:



B:



C:



D:

## LabVIEW-Programmierungsumgebung

### Themen

- Ein LabVIEW-Projekt einrichten und verwenden
- Datentypen
- Ausführungsreihenfolge und Verhalten eines VI vorhersagen
- Grundfunktionen zur Erstellung einer einfachen Erfassen-Analysieren-Visualisieren-Anwendung verwenden
- Fehlersuche durch Identifizierung und Behebung der Ursache für eine unterbrochene Schaltfläche oder falsche Daten
- Fehlerbehandlung mit Hilfe von Fehlerclustern und Zusammenführen von Fehlerfunktionen, um eine gute Fehlerbehandlung zu gewährleisten
- In LabVIEW navigieren, um mehr Informationen über Eingänge, Ausgänge und Funktionen zu erhalten

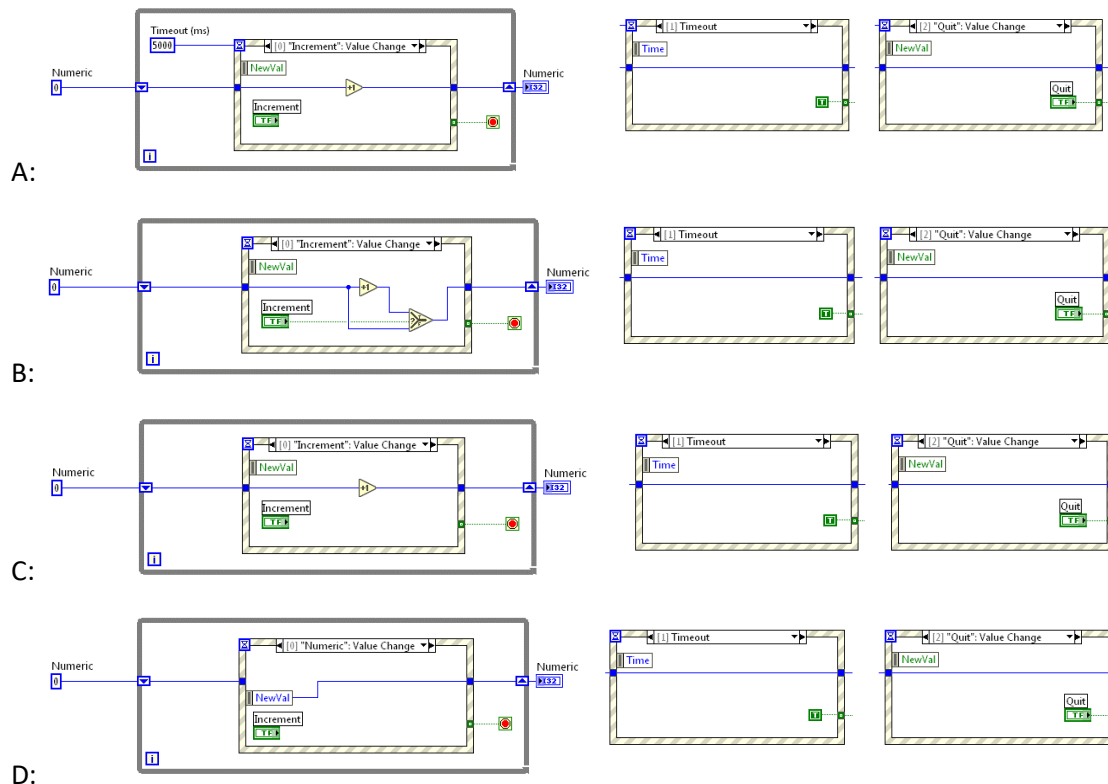
### Informationsquellen

- LabVIEW-Hilfe
  - *Übersicht über die numerischen Datentypen*
  - *Boolesche Bedien- und Anzeigeelemente*
  - *Blockdiagramm-Datenfluss*
  - *Verschiedene Typen von Graphen und Diagrammen*

### Beispielfragen

F1: Ihr VI muss zählen, wie oft die Inkrement-Schaltfläche gedrückt wird, wenn die Ausführen-Schaltfläche und dann die Abbrechen-Schaltfläche gedrückt werden.

Alle Ereignisfälle sind angezeigt. Welcher Code erfüllt diese Anforderungen am besten?



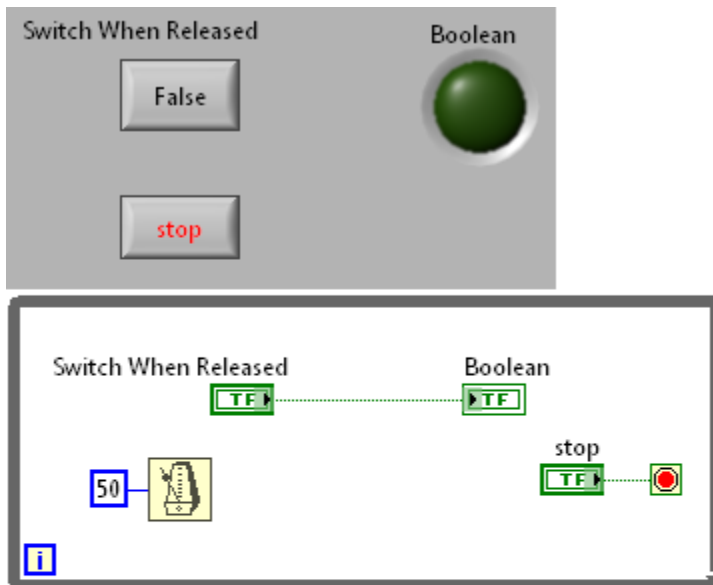
F2: Welcher Datentyp wäre geeignet, um den Winkel in Grad von -180 bis +180 Grad gerundet auf die nächste ganze Zahl anzuzeigen?

- A:  B:  C:  D: 

F3: Bevor dieses VI ausgeführt wird, sind **Umschalten beim Loslassen** und **Boolesch** „False“.

Sehen Sie sich das VI und die Benutzereingriffe an.

Welcher Wert wird in der Anzeige **Boolesch** bei 12 Sekunden und dann bei 18 Sekunden angezeigt?



Time in Seconds	User Action
0	Clicks <b>Run Arrow</b>
5	Clicks and holds mouse on the <b>Switch When Released</b> button
8	Moves mouse away from the <b>Switch When Released</b> button
11	Releases mouse
15	Clicks and holds mouse on the <b>Switch When Released</b> button
17	Releases mouse while still over the <b>Switch When Released</b> button
20	Clicks the <b>Stop</b> button

A: TRUE, TRUE

C: FALSE, TRUE

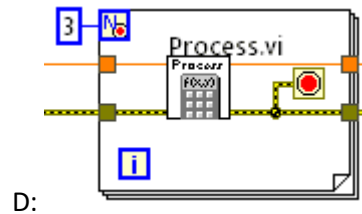
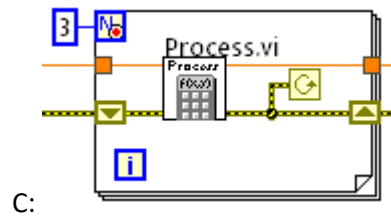
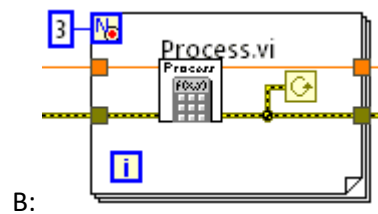
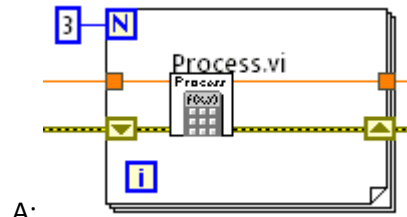
B: TRUE, FALSE

D: FALSE, FALSE



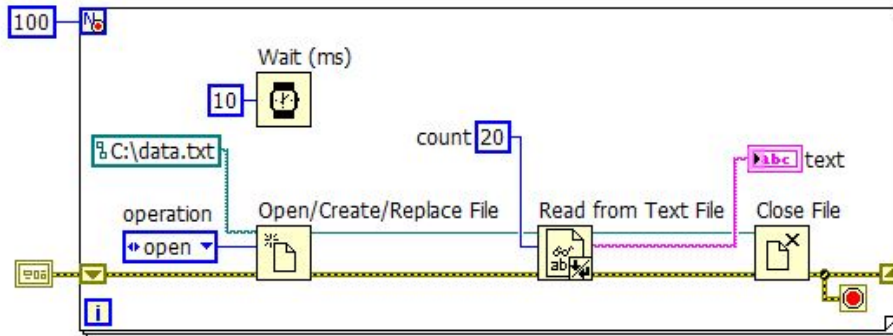
F4: Wenn „SubVI Process.vi“ einen Fehler erzeugt, sollte Ihr VI den Fehler beheben und versuchen, SubVI noch zweimal auszuführen, um den Vorgang erfolgreich abzuschließen.

Welches der folgenden VIs erfüllt diese Anforderungen am besten?

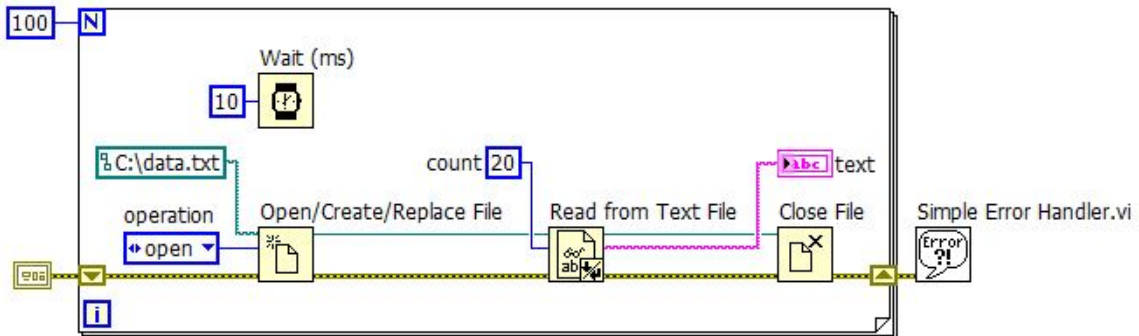


F5: Die Datei C:\data.txt existiert nicht. Die automatische Fehlerbehandlung ist deaktiviert.

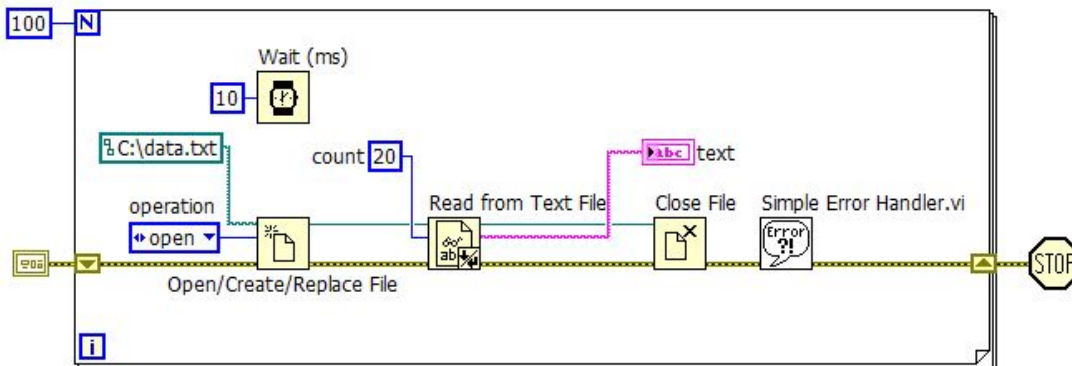
Welches Diagramm meldet den Fehler und stoppt innerhalb von 100 ms?



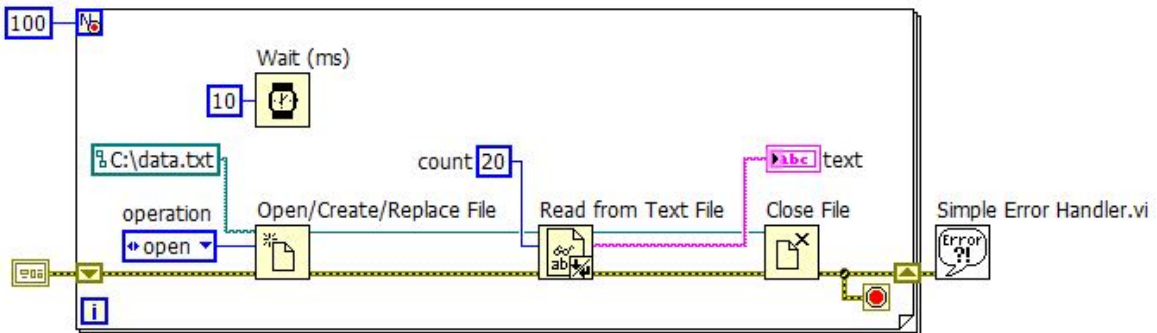
A:



B:

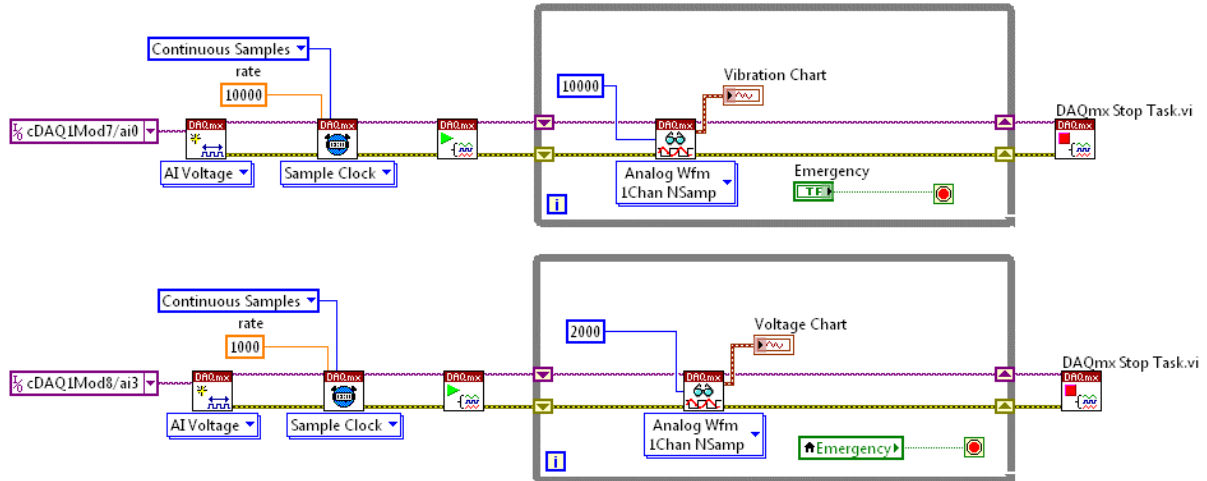


C:



D:

F6: Nachdem das Diagramm mit der Aktualisierung begonnen hat, wird die Notfalltaste gedrückt. Das VI:



- A: Stoppt nach maximal 1 Sekunde.
- B: Stoppt nach maximal 2 Sekunden.
- C: Stoppt nach maximal 4 Sekunden.
- D: Reagiert nicht mehr, bis der Vorgang abgebrochen wird.

# LabVIEW-Programmiergrundlagen

## Themen

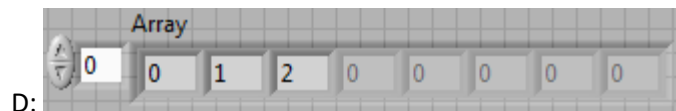
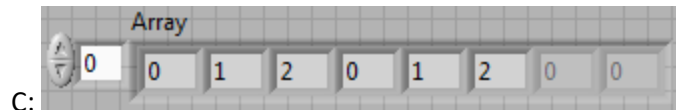
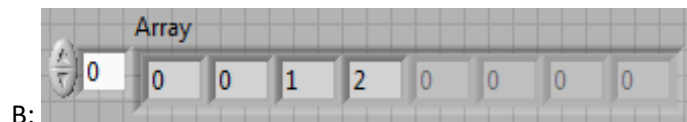
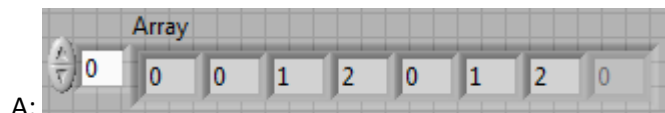
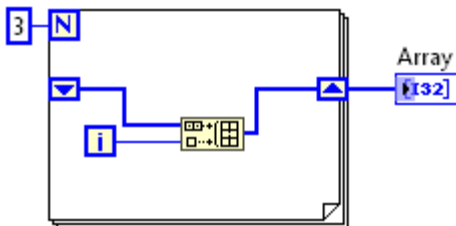
- Schleifen
- Arrays
- Bedingter Code
- Daten lesen/schreiben (Datei-I/O)

## Informationsquellen

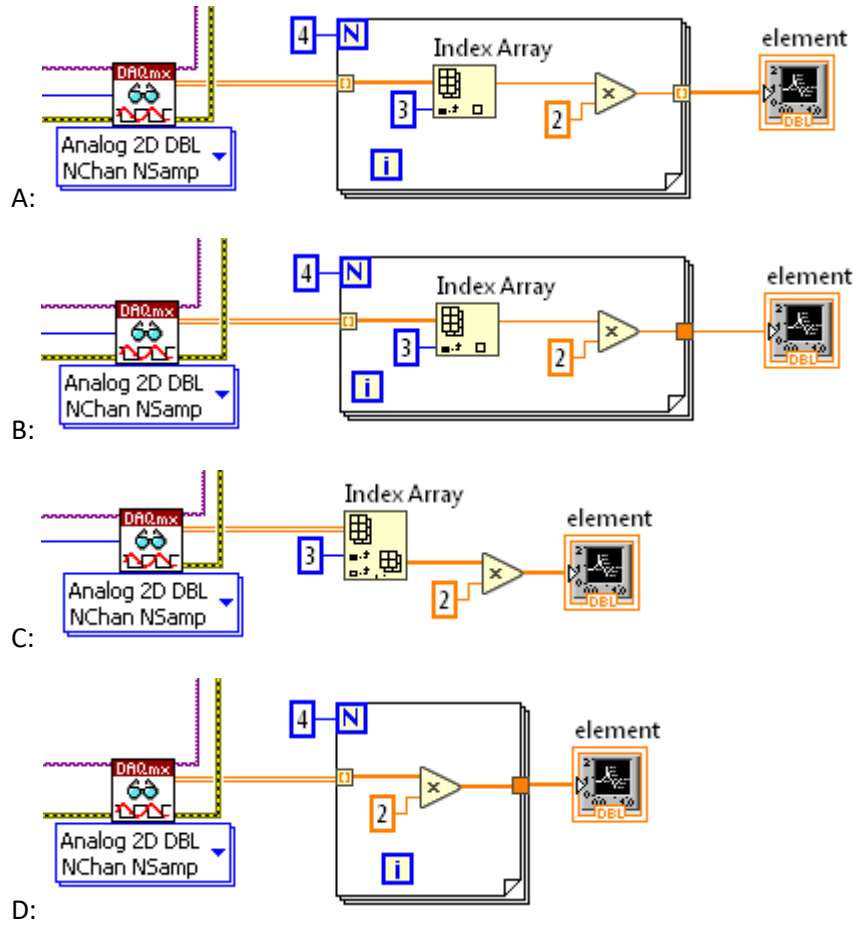
- LabVIEW-Grundlagen 1
- LabVIEW-Hilfe
  - Datei-I/O
  - Binärdateien
  - Tabellenkalkulationsdateien
  - TDM/TDMS-Dateien
  - Textdateien
  - Signalformen
  - For-Schleife
  - While-Schleife
  - Verarbeiten von einzelnen Elementen in einem Array mit einer Schleife

## Beispielfragen

F1: Wie sieht die Anzeige **Array** aus, nachdem dieses VI zweimal ausgeführt wurde?



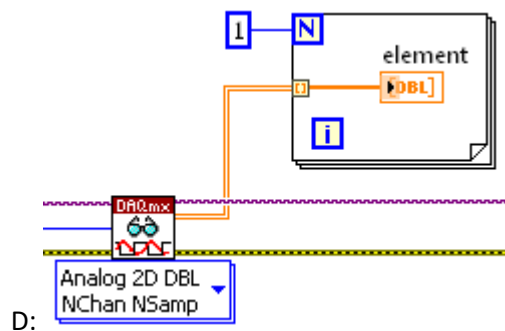
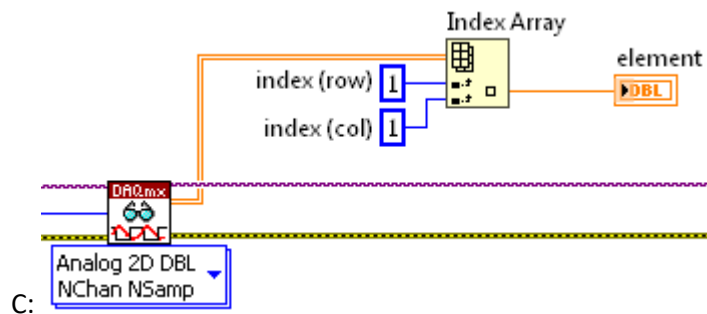
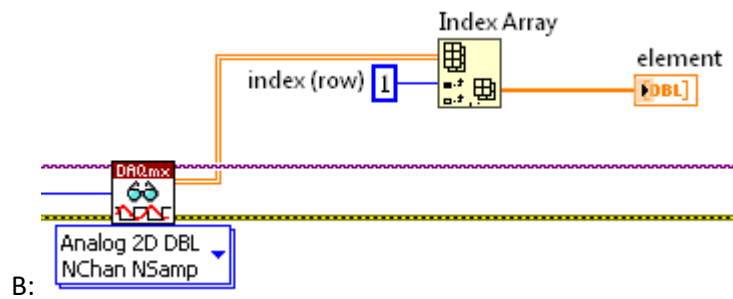
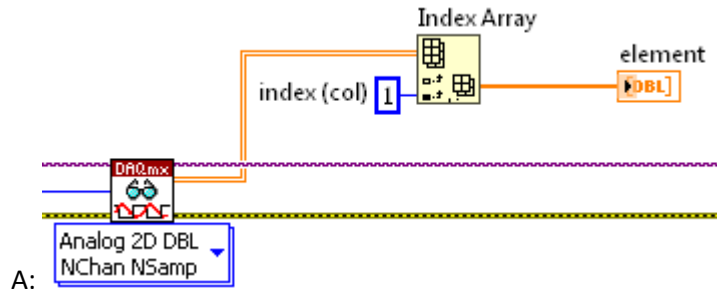
F2: Welches VI kann den vierten Kanalausgang der DAQmx-Lesen-Funktion extrahieren und jeden Wert verdoppeln oder ein leeres Array zurückgeben, wenn es weniger als 4 Kanäle gibt?

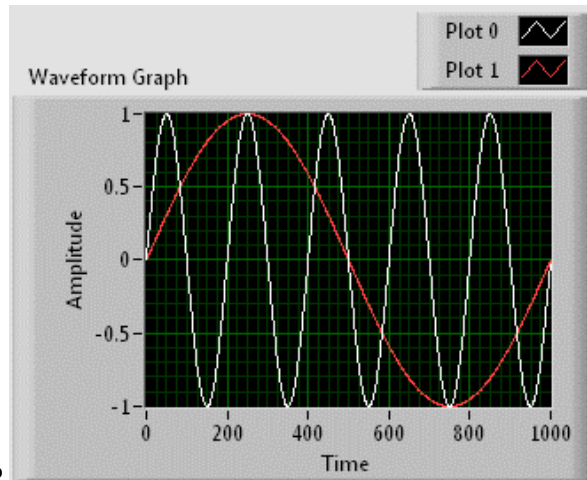


F3: Das DAQmx-Lesen-VI ist für das Lesen dieser Kanäle eingerichtet:

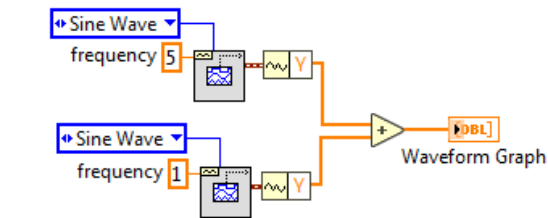


Welches VI extrahiert und zeigt korrekt alle Daten von Kanal ai2 an?

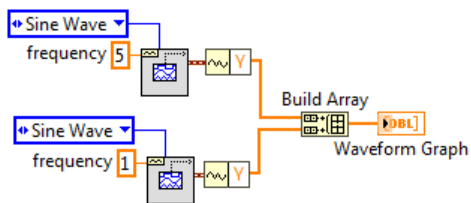




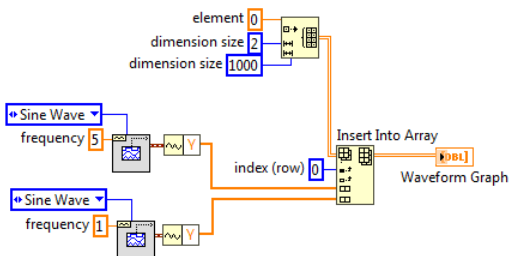
F4: Welches VI erzeugt diese Ausgabe?



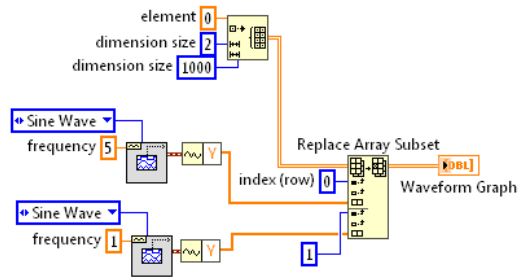
A:



B:

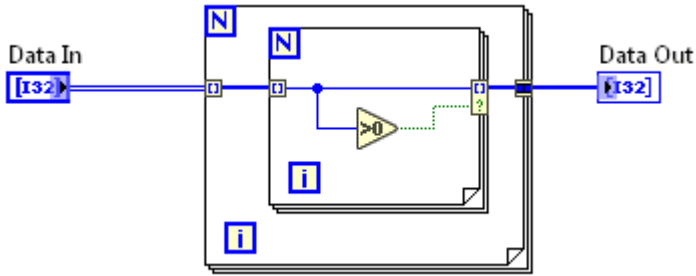


C:

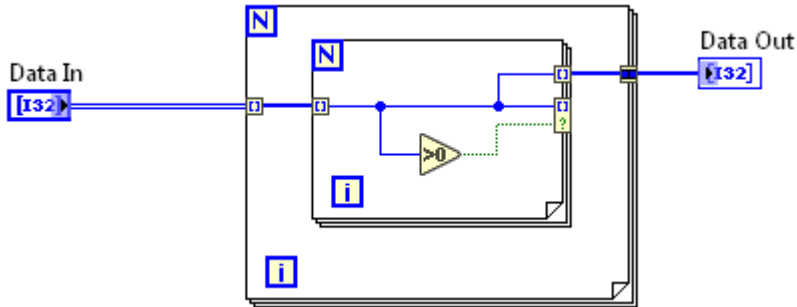


D:

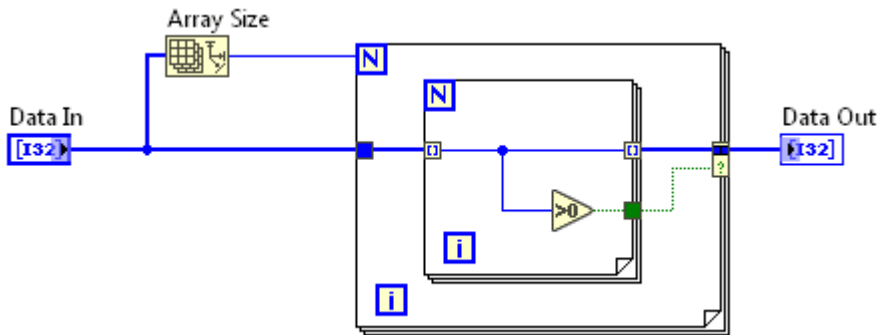
F5: Sie werden aufgefordert, ein VI zu schreiben, das ein 2D-Array verwendet und alle positiven Werte in einem 1D-Array ausgibt. Welches VI erfüllt diese Anforderungen?



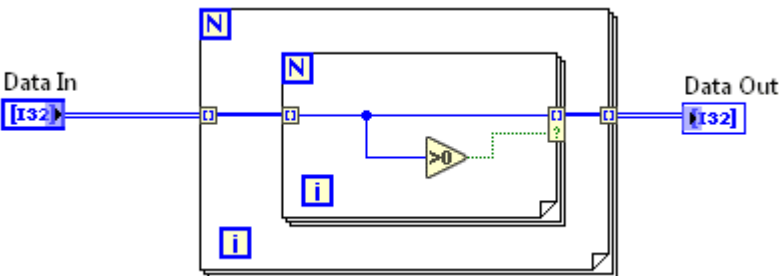
A:



B:



C:

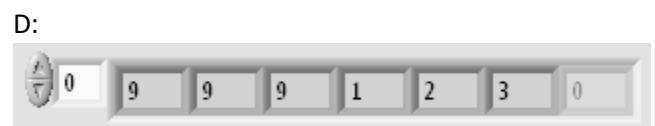
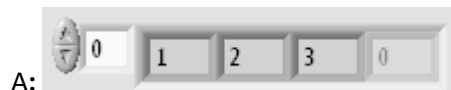
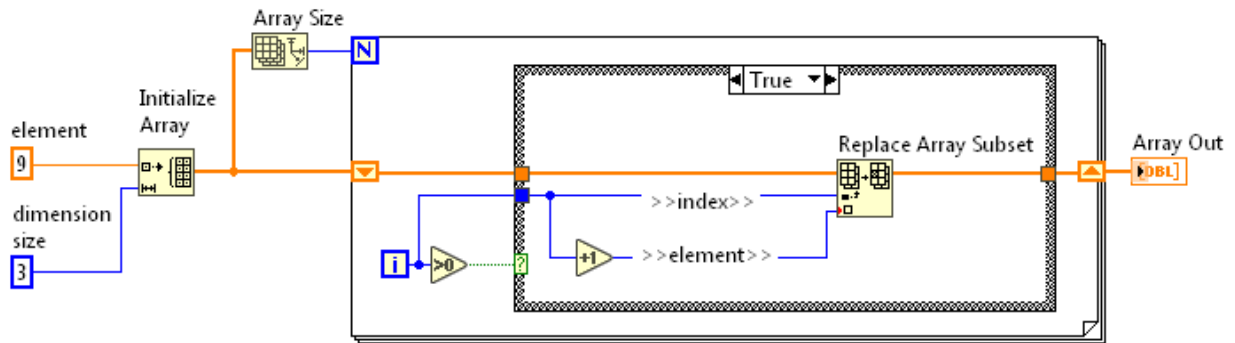


D:

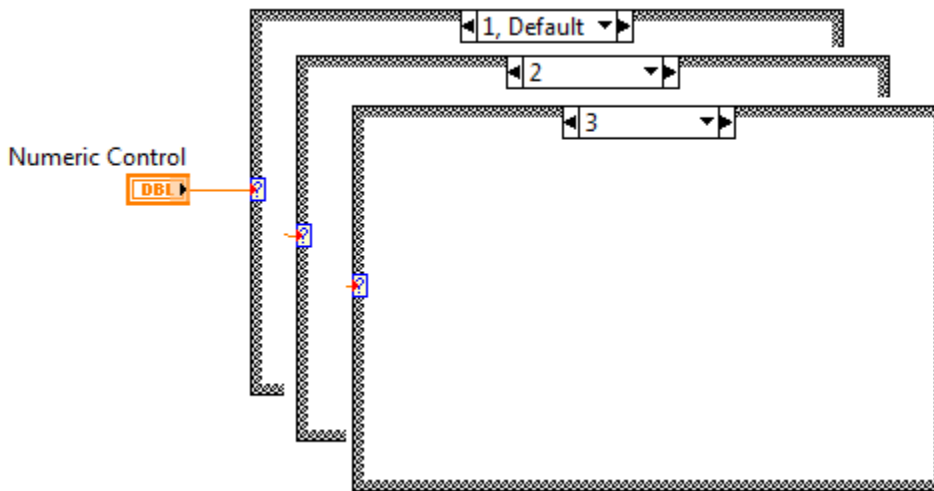


F6: Wie sieht die Anzeige **Array (Ausgang)** aus, nachdem das VI die Ausführung abgeschlossen hat?

Der Fall „False“ ist direkt verbunden.



F7: Das Bedienelement „Numerisch“ ist auf 2.5 gesetzt. Welcher Fall wird ausgeführt?



A: 1, Voreinstellung

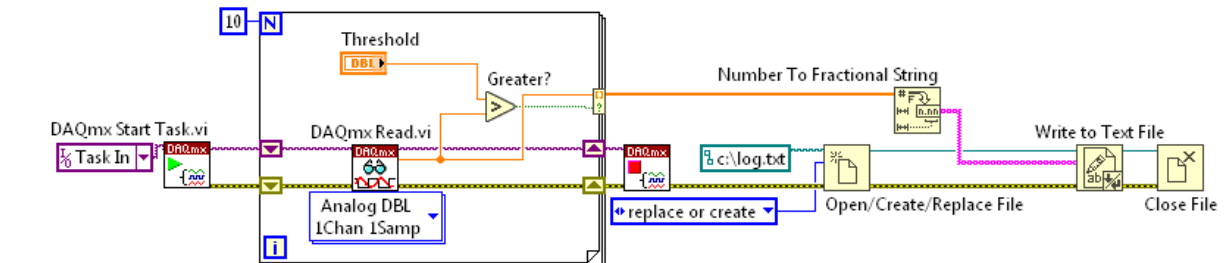
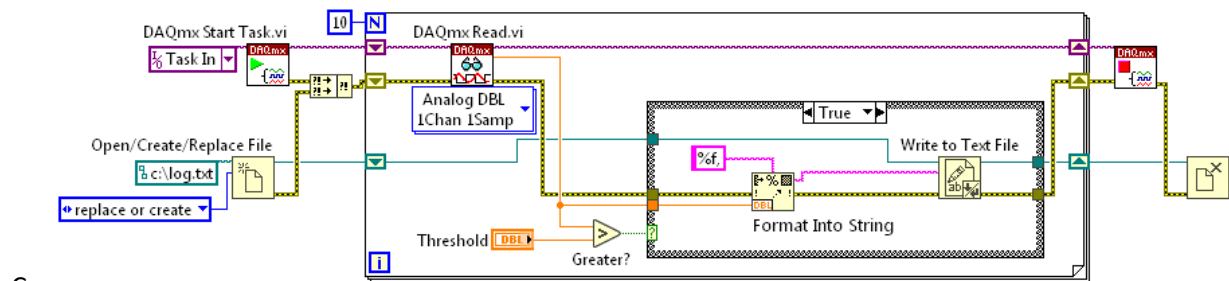
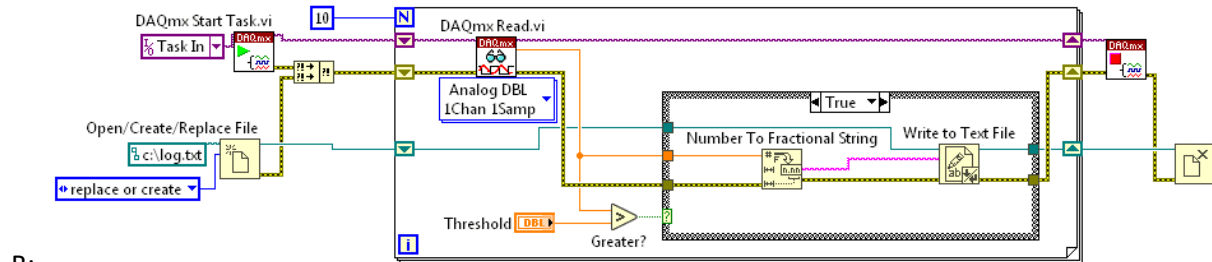
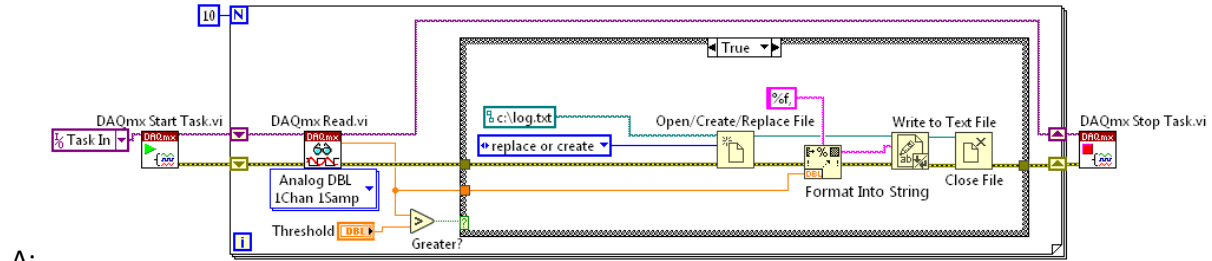
B: 2

C: 3

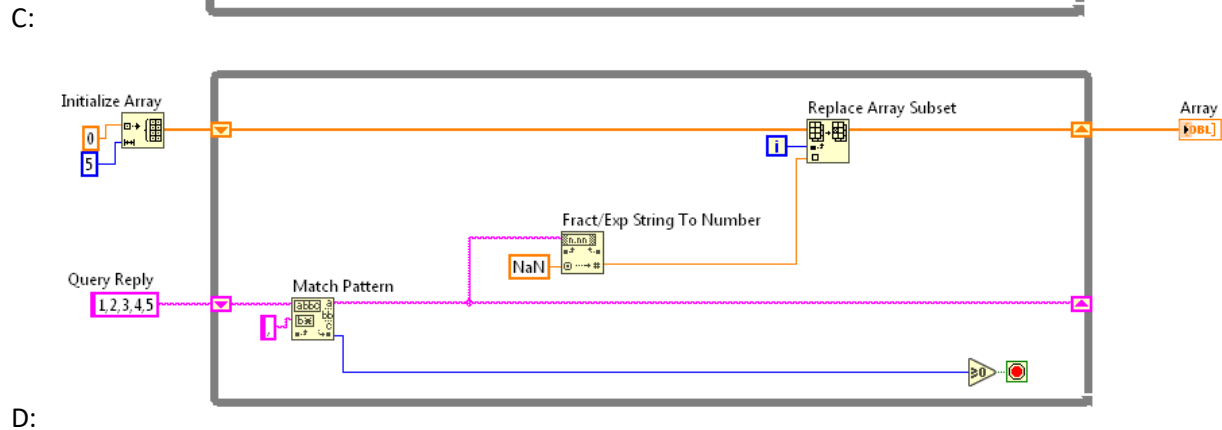
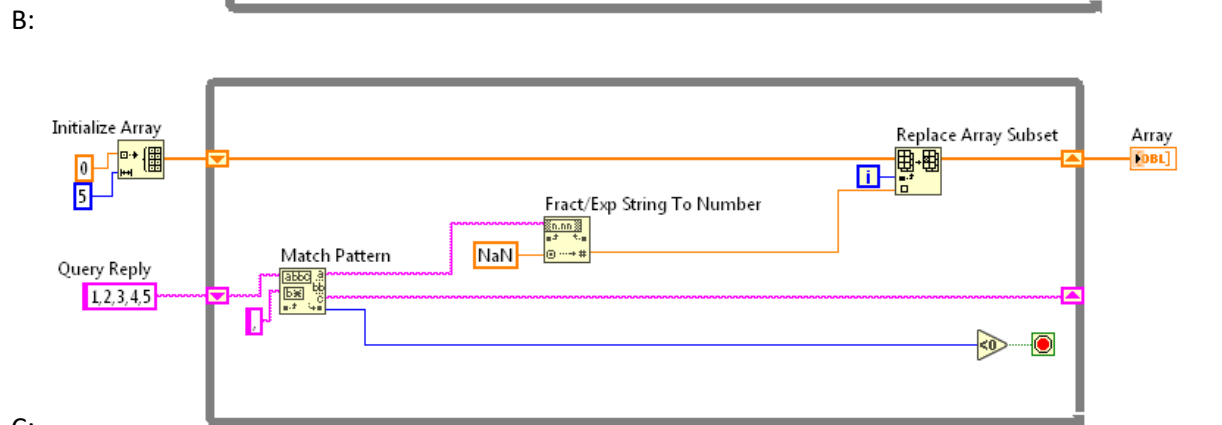
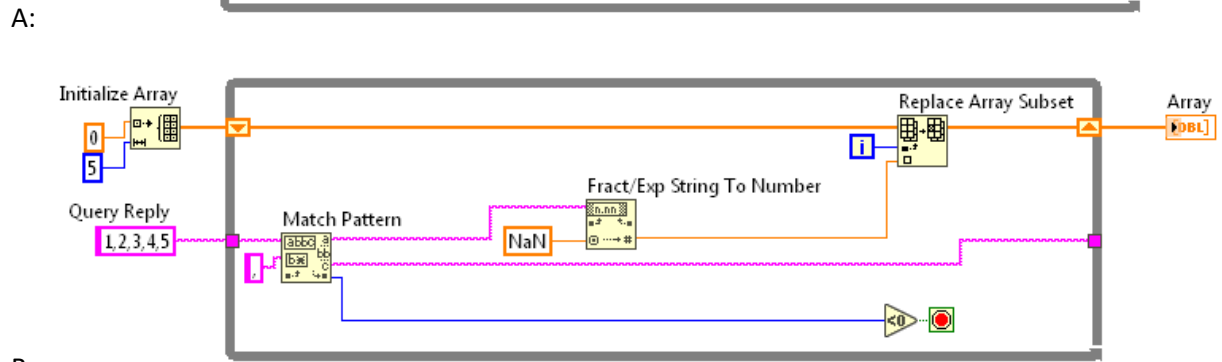
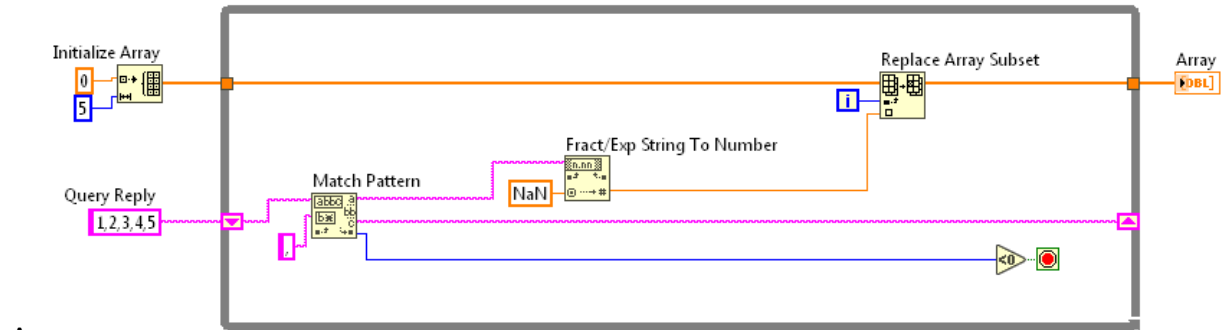
D: Unbestimmt

F8: Welches VI protokolliert jeden Abtastwert über dem Schwellenwert in einer vom Benutzer lesbaren Datei?

Alle nicht dargestellten Fälle sind direkt verbunden.



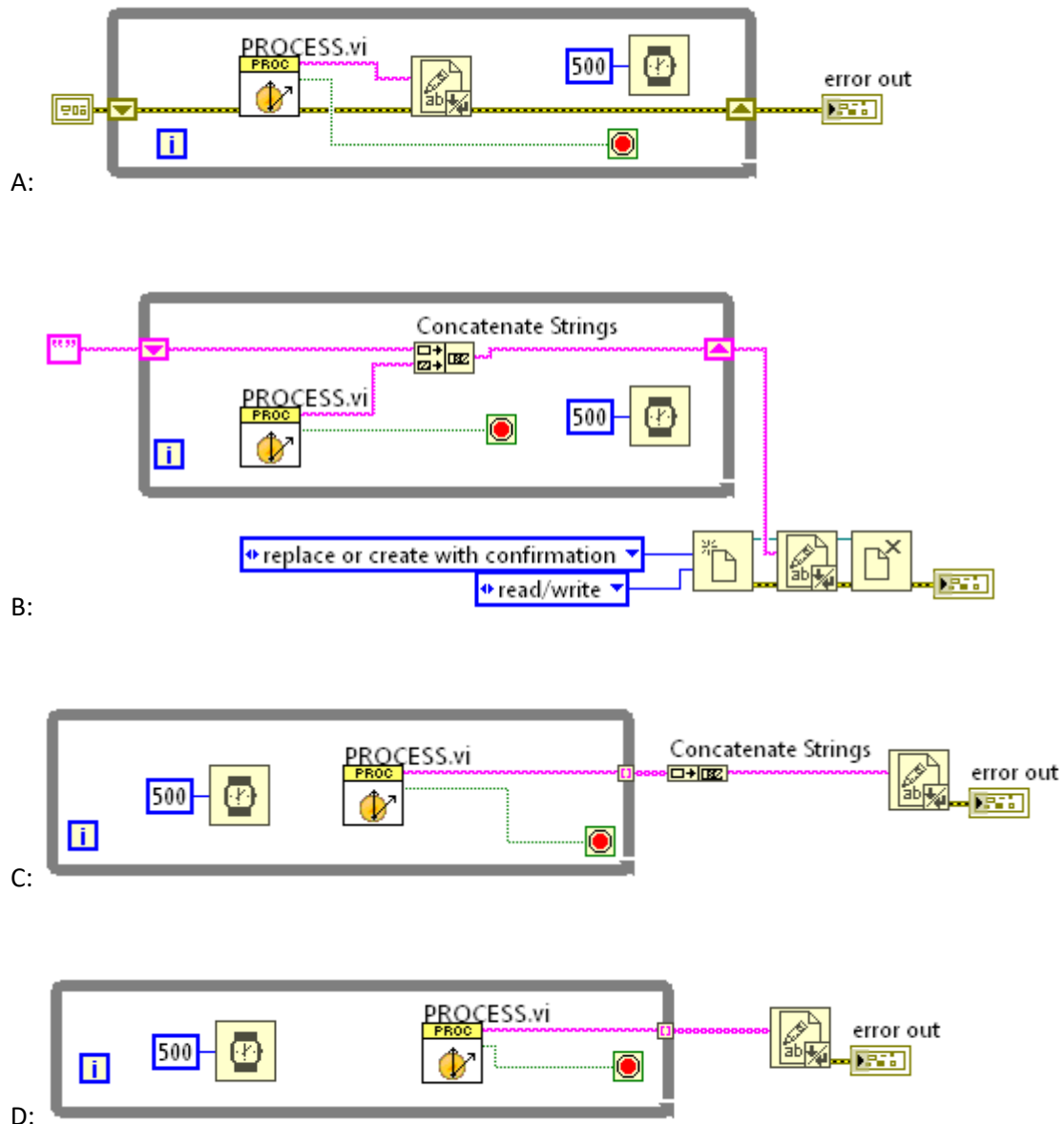
F9: Die Antwort auf eine Geräteabfrage erzeugt ein String von bis zu 5 kommagetrennten Parametern. Welches VI analysiert die Antwort und ordnet die Werte in einem numerischen Array an?



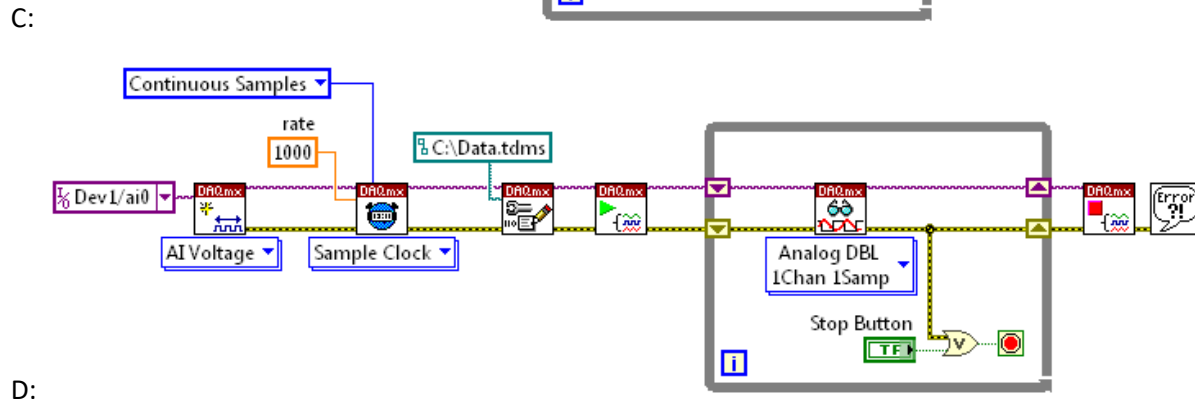
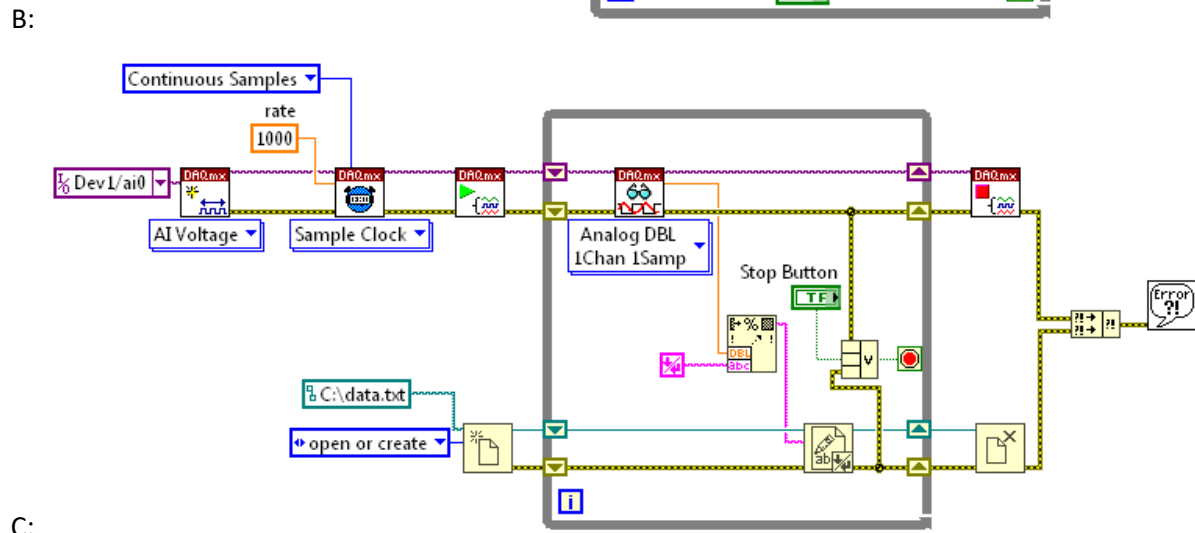
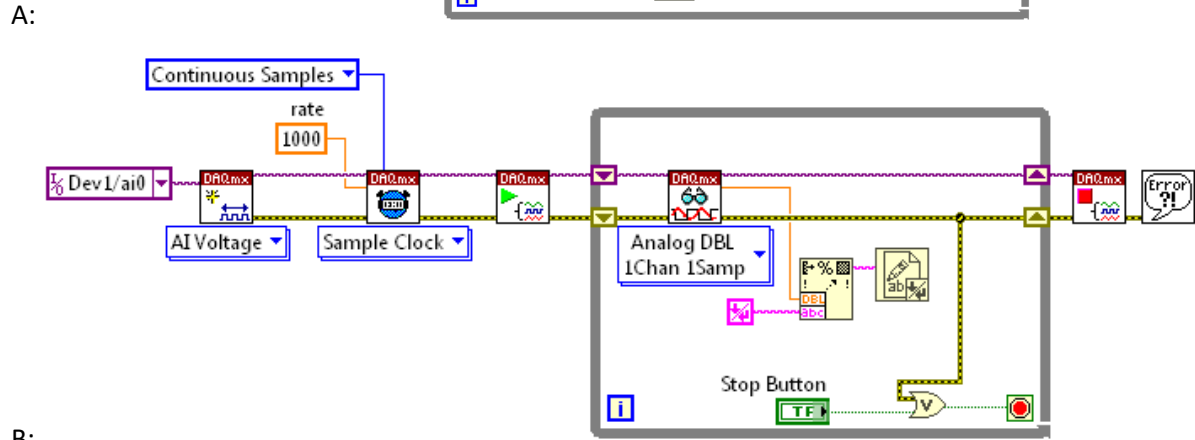
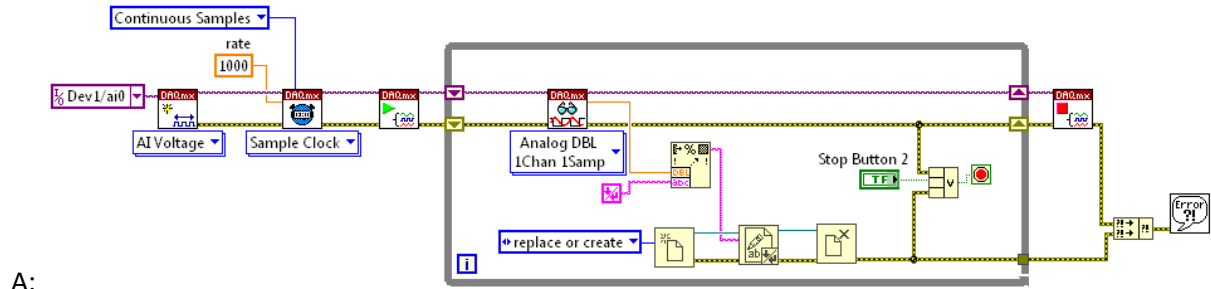
F10: Der SubVI PROCESS stellt einen Test dar, der bei jedem Testlauf einen String ausgibt, ohne die Konstante „Zeilenumbruch“ oder „Zeilenende“ im String.

Ihr VI muss den SubVI PROCESS wiederholen, bis der Benutzer die Schaltfläche „Stopp“ drückt, dann muss der Benutzer den Ort für das Protokoll auswählen können. Das VI muss alle Daten protokollieren und die Daten aus jedem Durchlauf von PROCESS in eine separate Zeile schreiben.

Welches der folgenden VIs erfüllt diese Anforderungen?



F11: Welches VI erfasst und protokolliert kontinuierlich Daten in einer für den Benutzer lesbaren Textdatei?



D:

## Best Practices für das Programmieren

### Unterthemen

- SubVIs – Wiederverwendung von Programmcode
- Clusters – Gruppierung von Daten gemischter Datentypen
- Typdefinitionen – Datentypänderungen propagieren

### Informationsquellen

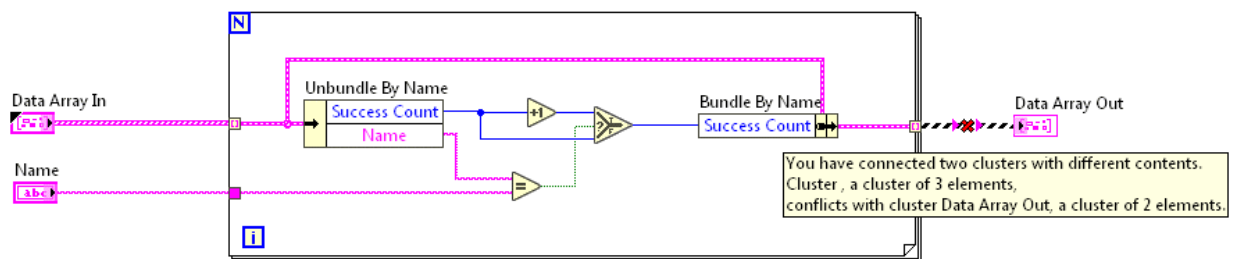
#### LabVIEW-Hilfe

- Erstellen benutzerdefinierter Bedien- und Anzeigeelemente und Typdefinitionen
- Erstellen von modularem Programmcode

### Beispielfragen

F1: Wenn sich der Cursor über der unterbrochenen Schaltfläche befindet, wird die unten dargestellte Fehlermeldung angezeigt.

Datenarray (Ausgang) ist der Cluster mit dem richtigen Inhalt. Wie könnte dieser Fehler behoben werden?



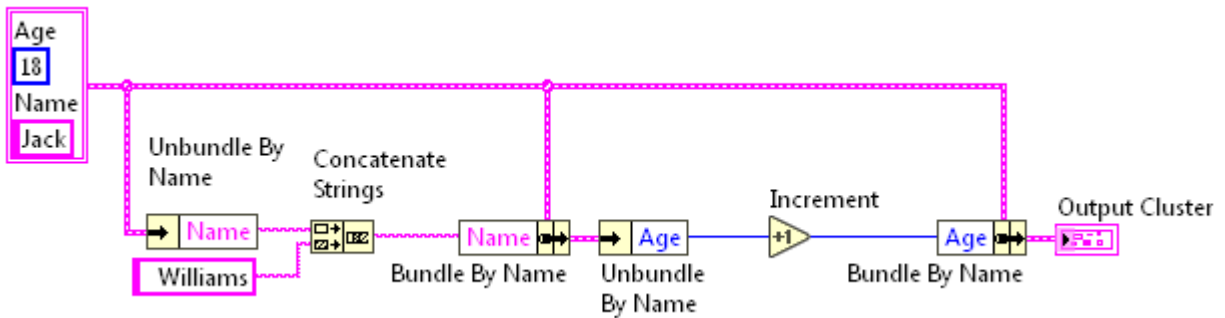
A: Datenarray (Ausgang) zu einer Typendefinition umwandeln, eine Kopie von Datenarray (Ausgang) erstellen, die Kopie in ein Bedienelement ändern und Datenarray (Eingang) durch das neue Bedienelement ersetzen.

B: Das zusätzliche Element aus dem Datenarray (Eingang) löschen und dann Datenarray (Ausgang) als eine Typendefinition speichern.

C: Mit der rechten Maustaste auf Datenarray (Eingang) klicken, „Typendefinition öffnen“ wählen, dann „Änderungen übernehmen“ im Dateimenü der Typendefinition wählen.

D: Datenarray (Eingang) löschen, STR-B drücken, um die unterbrochene Schaltfläche zu löschen, dann mit der rechten Maustaste auf das Schleifeneingabeterminal klicken und „Erstellen -- Bedienelement“ wählen.

F2: Welchen Wert gibt die Anzeige Cluster (Ausgang) nach der Ausführung des VIs aus?  
 Stringkonstanten sind auf '/' Codeanzeige gesetzt.



A: Output Cluster

Age
18
Name
Jack

B: Output Cluster

Age
19
Name
Jack

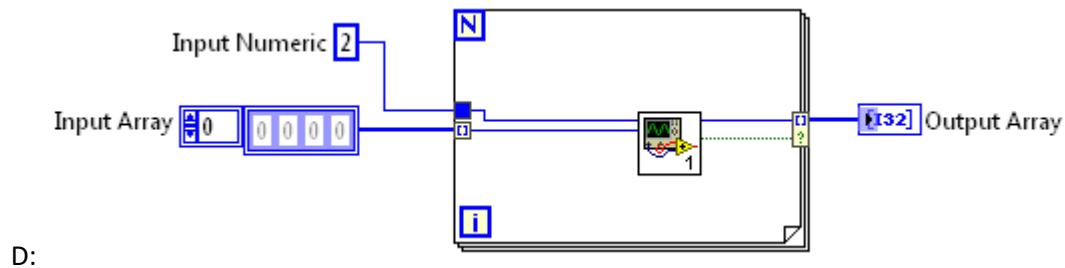
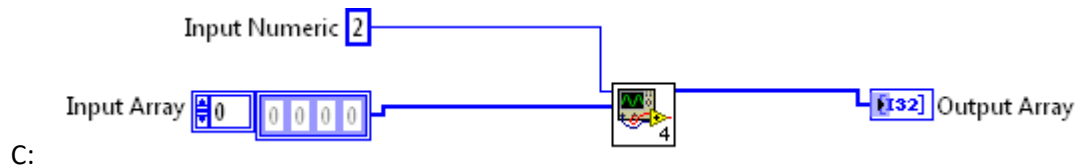
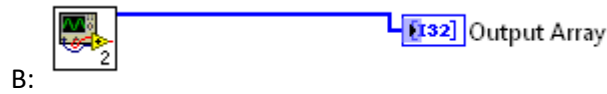
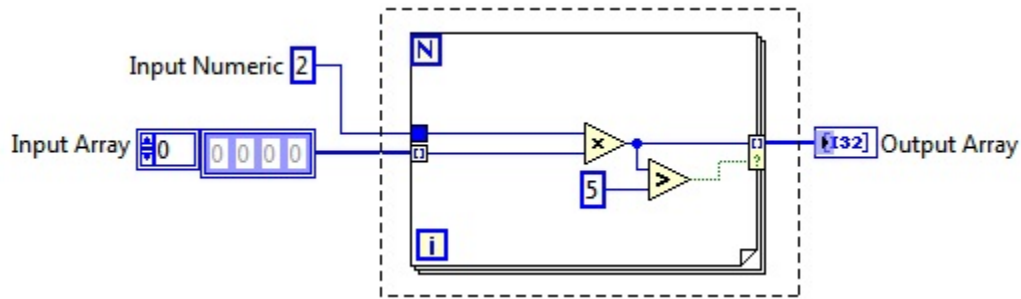
C: Output Cluster

Age
18
Name
Jack Williams

D: Output Cluster

Age
19
Name
Jack Williams

F3: Sie wählen nur die For-Schleife aus, die durch die gestrichelte Linie im Bild dargestellt ist und wählen dann **Bearbeiten»SubVI erstellen** im Bearbeitungs-menü. Wie sieht der Code im Blockdiagramm unmittelbar nach Erstellen von SubVI aus?

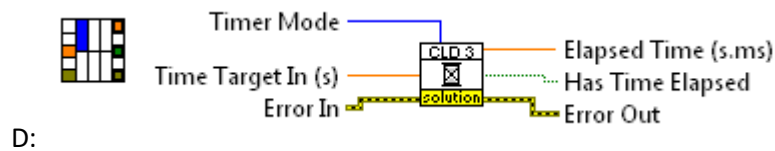
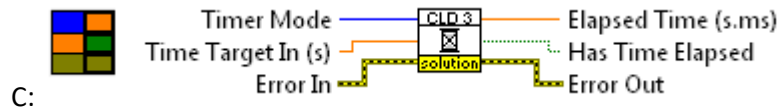
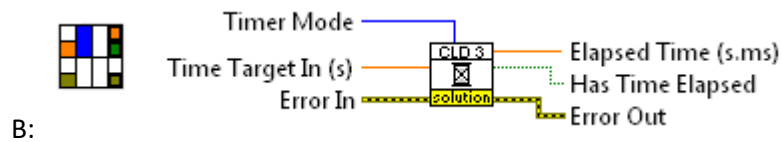
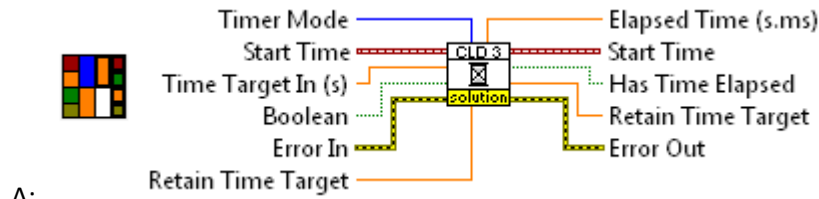
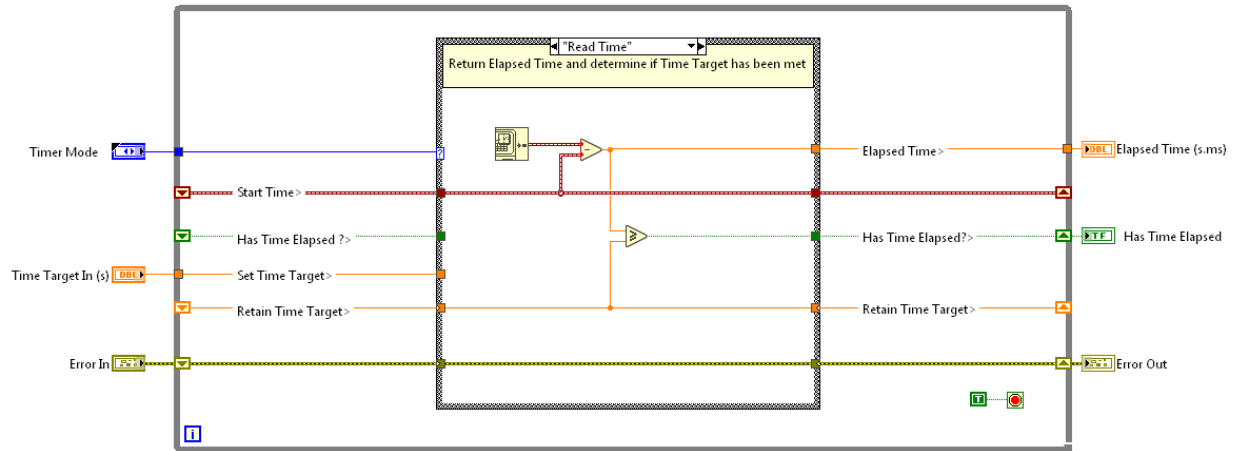


F4: Welcher der folgenden Schritte wird nicht automatisch durchgeführt, wenn Sie ein SubVI aus einem Codeabschnitt mit „Bearbeiten»SubVI erstellen“ erstellen?

- A: Zuordnung von Anschlüssen an die entsprechenden Anzeigen und Bedienelemente.
- B: Anschluss der Bedienelemente und Anzeigen auf dem Frontpanel an Anschlüsse auf dem Anschlussfeld.
- C: Konfigurieren des Anschlussfelds, um ausreichend Anschlüsse für alle Ein- und Ausgänge zur Verfügung zu stellen.
- D: Konfiguration von obligatorischen Eingängen nach Bedarf.



F5: Welche Konfiguration des Anschlussfelds entspricht bei dieser funktionalen globalen Variablen, die als Timer verwendet wird, am besten der Best Practice?



## ANHANG I: SIMULIERTER DAQ FÜR DIE VORBEREITUNG AUF CLAD

Verwenden Sie NI MAX, um die folgende simulierte DAQ-Hardware zu erstellen. Die Konfiguration dieses simulierten cDAQ-Geräts entspricht der Konfiguration des physischen cDAQ-Demo-Kits, das in einigen Kursen verwendet wird. Mit dieser Konfiguration können Sie viele der VIs erstellen und ausführen, die in den HW-bezogenen Fragen der CLAD-Prüfung und den Vorbereitungsmaterialien verwendet werden.

### Simulierte Hardware erstellen

1. Öffnen Sie MAX.
2. Klicken Sie mit der rechten Maustaste auf **Geräte und Schnittstellen** und wählen Sie **Neu**.
3. Wählen Sie im Dialogfeld Neu erstellen den Punkt **Simuliertes NI-DAQmx-Gerät oder modulares Instrument** aus und klicken Sie auf die Schaltfläche **Abschließen**.
4. Erweitern Sie das **CompactDAQ-Gehäuse**, wählen Sie **NI cDAQ-9178** und klicken Sie auf **OK**.
  - a. Das Gerät wird in der Liste **Geräte und Schnittstellen** als **NI cDAQ-9173 „cDAQx“** angezeigt, wobei „x“ 1 ist, es sei denn Sie haben weitere cDAQ-9178-Geräte (echt oder simuliert).
5. Klicken Sie mit der rechten Maustaste auf den Gerätenamen (NI cDAQ-9178 „CLAD“) in der Liste „Geräte und Schnittstellen“, und wählen Sie **Simuliertes cDAQ-Gehäuse simulieren** aus.
  - a. Erstellen Sie simulierte Instrumente für jeden Slot wie in Tabelle 2-1 beschrieben. Hinweis: Sie geben nicht die Informationen aus der Spalte „Beschreibung“ ein. Diese Informationen dienen als Referenz und beschreiben den Modultyp.

Slot	Modul/Modell	Beschreibung
1	NI 9236	Dehnungs-/Brückenmessung
2	NI 9213	Thermoelementeingang
3	NI 9472	Digitalausgabe
4	NI 9263	Spannungsausgabe
5	Leer	-
6	Leer	-
7	NI 9234	Schall- und Schwingungseingang
8	NI 9215	Spannungseingang

**Tabelle 2-1.**  
Simuliertes CompactDAQ-Gerät für LabVIEW Core1/Core2 und CLAD-Prüfung

## Die DAQmx-VIs sind in der CLAD-Prüfung enthalten

Der Kandidat muss die Funktion der folgenden NI-DAQmx-Konstanten und -VIs erkennen und kennen, die in der Palette „Funktionen » Messung-I/O » DAQmx“ enthalten sind. Die Aufgaben Erfassung/Analyse/Visualisierung und Protokollierung verwenden ähnliche VIs, unabhängig von der tatsächlich verwendeten Hardware. Die Beispielfragen in diesem Vorbereitungsleitfaden sollen den erforderlichen Kenntnisstand verdeutlichen.

- Konstanten: DAQmx physischer Kanal, Name der DAQmx-Aufgabe
- Konfiguration-VIs: DAQmx virtuellen Kanal erstellen, DAQmx-Timing, DAQmx-Trigger
- Start- und I/O-VIs: DAQmx Start, DAQmx Lesen, DAQmx Schreiben
- Ende-VIs: DAQmx Warten bis zum Abschluss, DAQmx Stopp-Aufgabe, DAQmx Löschen-Aufgabe

DAQmx Physical Channel



DAQmx Task Name



DAQmx Create Virtual Channel.vi



AI Voltage

DAQmx Timing.vi



Sample Clock

DAQmx Trigger.vi



Start  
None

DAQmx Start Task.vi



DAQmx Read.vi



Analog DBL  
1Chan 1Samp

DAQmx Write.vi



Analog DBL  
1Chan 1Samp

DAQmx Wait Until Done.vi



DAQmx Stop Task.vi



DAQmx Clear Task.vi



## ANHANG II: RICHTIGE ANTWORTEN AUF BEISPIELFRAGEN

### Hardware

F1: D

F2: B

F3: D

F4: B

F5: C

F6: A

F7: C

### LabVIEW-Programmierungsumgebung

F1: C

F2: D

F3: C

F4: B

F5: D

F6: C

### LabVIEW-Programmiergrundlagen

F1: C

F2: C

F3: B

F4: D

F5: A

F6: B

F7: B

F8: C

F9: C

F10: D

F11: C

Best Practices für das Programmieren

F1: A

F2: B

F3: C

F4: D

F5: B

ENDE DES VORBEREITUNGSLEITFADENS

Rev. 11. Jan. 2019